

Package ‘xpose4’

May 8, 2026

Title Diagnostics for Nonlinear Mixed-Effect Models

Version 4.7.4

Depends R (>= 2.2.0), lattice

Imports Hmisc, survival, dplyr, tibble, methods, lazyeval, gam,
splines, grid, readr

Suggests testthat

Description A model building aid for nonlinear mixed-effects
(population) model analysis using NONMEM, facilitating data set
checkout, exploration and visualization, model diagnostics, candidate
covariate identification and model comparison. The methods are described
in Keizer et al. (2013) <[doi:10.1038/psp.2013.24](https://doi.org/10.1038/psp.2013.24)>, and Jonsson et al. (1999)
<[doi:10.1016/s0169-2607\(98\)00067-4](https://doi.org/10.1016/s0169-2607(98)00067-4)>.

LazyLoad yes

LazyData yes

ByteCompile true

License LGPL (>= 3)

URL <https://uupharmacometrics.github.io/xpose4/>,
<https://github.com/UUPharmacometrics/xpose4>

BugReports <https://github.com/UUPharmacometrics/xpose4/issues>

Copyright 2009-2025 Andrew C. Hooker, E. Niclas Jonsson and Mats O.
Karlsson. 2005-2008 Andrew C. Hooker, Justin J. Wilkins, Mats
O. Karlsson and E. Niclas Jonsson. 1998-2004 E. Niclas Jonsson
and Mats Karlsson.

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Andrew C. Hooker [aut, cre, cph],
Mats O. Karlsson [aut, cph],
Justin J. Wilkins [aut],
E. Niclas Jonsson [aut, trl, cph],
Ron Keizer [ctb] (functionality for bootstrap of GAM and SCM)

Maintainer Andrew C. Hooker <andrew.hooker@uu.se>

Repository CRAN

Date/Publication 2025-09-03 10:50:02 UTC

Contents

xpose4-package	5
absval.cwres.vs.cov.bw	7
absval.cwres.vs.pred	9
absval.cwres.vs.pred.by.cov	11
absval.iwres.cwres.vs.ipred.pred	13
absval.iwres.vs.cov.bw	15
absval.iwres.vs.idv	16
absval.iwres.vs.ipred	17
absval.iwres.vs.ipred.by.cov	19
absval.iwres.vs.pred	21
absval.wres.vs.cov.bw	22
absval.wres.vs.idv	24
absval.wres.vs.pred	26
absval.wres.vs.pred.by.cov	28
absval_delta_vs_cov_model_comp	30
add.grid.table	33
add.model.comp	35
addit.gof	37
add_transformed_columns	39
autocorr.cwres	40
autocorr.iwres	42
autocorr.wres	44
basic.gof	46
basic.model.comp	47
boot.hist	49
bootgam.print	51
bootscm.import	52
cat.dv.vs.idv.sb	53
cat.pc	55
change.parm	57
change.var.name	58
change.xlabel	59
change.xvardef	60
change_graphical_parameters	62
change_misc_parameters	64
compute.cwres	67
cov.splom	75
create.xpose.plot.classes	78
createXposeClasses	78
cwres.dist.hist	79
cwres.dist.qq	80

cwres.vs.cov	81
cwres.vs.idv	83
cwres.vs.idv.bw	84
cwres.vs.pred	86
cwres.vs.pred.bw	87
cwres.wres.vs.idv	89
cwres.wres.vs.pred	90
data.checkout	92
data_extract_or_assign	94
db.names	96
dOFV.vs.cov	97
dOFV.vs.id	99
dOFV1.vs.dOFV2	101
dv.preds.vs.idv	102
dv.vs.idv	104
dv.vs.ipred	105
dv.vs.ipred.by.cov	107
dv.vs.ipred.by.idv	108
dv.vs.pred	110
dv.vs.pred.by.cov	111
dv.vs.pred.by.idv	113
dv.vs.pred.ipred	114
export.graph.par	115
export.variable.definitions	117
GAM_summary_and_plot	118
gof	120
import.graph.par	123
import.variable.definitions	124
ind.plots	126
ind.plots.cwres.hist	129
ind.plots.cwres.qq	132
ipred.vs.idv	134
iwres.dist.hist	136
iwres.dist.qq	137
iwres.vs.idv	138
kaplan.plot	140
make.sb.data	144
npc.coverage	145
nsim	147
parm.vs.cov	148
parm.vs.parm	150
par_cov_hist	151
par_cov qq	153
par_cov_summary	155
pred.vs.idv	157
print.xpose.multiple.plot	158
randtest.hist	159
ranpar.vs.cov	160

read.nm.tables	162
read.npc.vpc.results	164
read.TTE.sim.data	165
read.vpctab	166
read_NM_output	168
read_nm_table	169
reset.graph.par	170
runsum	171
simpraz.xpdb	173
simprazExample	173
tabulate.parameters	174
wres.dist.hist	175
wres.dist.qq	176
wres.vs.cov	178
wres.vs.idv	179
wres.vs.idv.bw	181
wres.vs.pred	182
wres.vs.pred.bw	183
xlabel	185
xp.boot.par.est	186
xp.boot.par.est.corr	188
xp.daic.npar.plot	189
xp.distr.mod.size	190
xp.dofv.npar.plot	191
xp.dofv.plot	192
xp.get.disp	192
xp.inc.cond.stab.cov	193
xp.inc.ind.cond.stab.cov	194
xp.inc.prob	195
xp.inc.prob.comb.2	196
xp.inc.stab.cov	197
xp.incl.index.cov	198
xp.incl.index.cov.comp	199
xp.incl.index.cov.ind	200
xp.scope3	201
xpose.bootgam	202
xpose.data	204
xpose.data-class	207
xpose.gam	207
xpose.license.citation	210
xpose.logTicks	210
xpose.multiple.plot	212
xpose.multiple.plot-class	213
xpose.multiple.plot.default	213
xpose.panel.bw	215
xpose.panel.default	218
xpose.panel.histogram	230
xpose.panel.qq	232

xpose.panel.splom	233
xpose.plot.bw	236
xpose.plot.default	239
xpose.plot.histogram	244
xpose.plot.qq	247
xpose.plot.splom	249
xpose.prefs-class	252
xpose.print	253
xpose.string.print	254
xpose.VPC	254
xpose.VPC.both	260
xpose.VPC.categorical	262
xpose4	265
xsubset	266
xvardef	267

Index**269**

xpose4-package	<i>The Xpose Package</i>
----------------	--------------------------

Description

Xpose is an R-based model building aid for population analysis using NONMEM. It facilitates data set checkout, exploration and visualization, model diagnostics, candidate covariate identification and model comparison.

Details

Xpose takes output from NONMEM output and/or PsN output and generates graphs or other analyses. It is assumed that each NONMEM run can be uniquely identified by a run number (see section below for how to generate the appropriate input to Xpose). Xpose is implemented using the lattice graphics library.

The Xpose package can be divided up into six subsections (functions associated with each of the different subsections are linked in the "See Also" section):

Data Functions Functions for managing the input data and manipulating the Xpose database.

Generic Functions Generic wrapper functions around the lattice functions. These functions can be invoked by the user but require quite detailed instructions to generate the desired output.

Specific Functions These functions are single purpose functions that generate specific output given only the Xpose database as input. The behavior can, to some extent, be influenced by the user.

Classic Functions Xpose has a text based menu interface to make it simple for the user to invoke the Xpose specific functions. This interface is called Xpose Classic. Given the limitations a text based interface imposes, Xpose Classic is not very flexible but may be useful for quick assessment of a model and for learning to use Xpose.

PsN Functions These functions are the interface between Xpose and PsN, i.e. they do not post-process NONMEM output but rather PsN output.

GAM Functions Functions take an Xpose object and performs a generalized additive model (GAM) stepwise search for influential covariates on a single model parameter.

How to make NONMEM generate input to Xpose

Xpose recognizes NONMEM runs, and files associated to a particular run, though the run number. This is a number that is used in the name of NONMEM model files, output files and table files. The fundamental input to Xpose is one or more NONMEM table files. These table files should be named as below followed by the run number, for example xptab1 for run number 1. Xpose looks for files according to the following pattern, where * is your run number:

sdtab* Standard table file, containing ID, IDV, DV, PRED, IPRED, WRES, IWRES, RES, IRES, etc.

patab* Parameter table, containing model parameters - THETAs, ETAs and EPSes

catab* Categorical covariates, e.g. SEX, RACE

cotab* Continuous covariates, e.g. WT, AGE

extra*, **mutab***, **mytab***, **xptab***, **cwtab*** Other variables you might need to have available to Xpose

run*.mod Model specification file

run*.lst NONMEM output

Strictly, only one table file is needed for xpose (for example sdtab* or xptab*). However, using patab*, cotab*, catab* will influence the way that Xpose interprets the data and are recommended to get full benefit from Xpose.

You can use code in NONMEM similar to the following to generate the tables you need. NONMEM automatically appends DV, PRED, WRES and RES unless NOAPPEND is specified. Don't forget to leave at least one blank line at the end of the NONMEM model specification file.

```
$TABLE ID TIME IPRED IWRES EVID MDV NOPRINT ONEHEADER FILE=sdtab1 $TABLE ID CL V2 KA K SLP
KENZ NOPRINT ONEHEADER FILE=patab1 $TABLE ID WT HT AGE BMI PKG NOPRINT ONEHEADER FILE=cotab1
$TABLE ID SEX SMOK ALC NOPRINT ONEHEADER FILE=catab1
```

Author(s)

E. Niclas Jonsson, Mats Karlsson, Justin Wilkins and Andrew Hooker

References

PsN

See Also

Useful links:

- <https://uopharmacometrics.github.io/xpose4/>
- <https://github.com/UUPharmacometrics/xpose4>
- Report bugs at <https://github.com/UUPharmacometrics/xpose4/issues>

Other data functions: `add_transformed_columns`, `change_graphical_parameters`, `change_misc_parameters`, `compute.cwres()`, `data.checkout()`, `data_extract_or_assign`, `db.names()`, `export.graph.par()`, `export.variable.definitions()`, `import.graph.par()`, `import.variable.definitions()`, `make.sb.data()`, `nsim()`, `par_cov_summary`, `read.TTE.sim.data()`, `read.nm.tables()`, `read_NM_output`, `read_nm_table()`, `simprazExample()`, `tabulate.parameters()`, `xlabel()`, `xpose.data`, `xpose.print()`, `xsubset()`

Other generic functions: `gof()`, `xpose.multiple.plot`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `dOFV.vs.cov()`, `dOFV.vs.id()`, `dOFV1.vs.dOFV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`

Other classic functions: `xpose4()`

Other PsN functions: `boot.hist()`, `bootscm.import()`, `npc.coverage()`, `randtest.hist()`, `read.npc.vpc.results()`, `read.vpctab()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`

Other GAM functions: `GAM_summary_and_plot`, `xp.get.disp()`, `xp.scope3()`, `xpose.bootgam()`, `xpose.gam()`

Examples

```
## Not run:
# run the classic interface
library(xpose4)
xpose4()

# command line interface
library(xpose4)
xpdb <- xpose.data(5)
basic.gof(xpdb)

## End(Not run)
```

`absval.cwres.vs.cov.bw`

Absolute conditional weighted residuals vs covariates for Xpose 4

Description

This creates a stack of box and whisker plot of absolute population conditional weighted residuals (ICWRES) vs covariates, and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the [codexpose.plot.bw](#) function. Most of the options take their default values from xpose.data object but may be overridden by supplying them as arguments.

Usage

```
absval.cwres.vs.cov.bw(object, xlb = "|CWRES|", main = "Default", ...)
```

Arguments

object	An xpose.data object.
xlb	A string giving the label for the x-axis. NULL if none.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to xpose.plot.bw .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling box-and-whisker plots are available. See [xpose.plot.bw](#) for details.

Value

Returns a stack of box-and-whisker plots of |CWRES| vs covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.bw](#), [xpose.panel.bw](#), [compute.cwres](#), [bwplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#),

```
dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(),
ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(), iwres.dist.qq(), iwres.vs.idv(),
kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package
```

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

absval.cwres.vs.cov.bw(xpdb)
```

absval.cwres.vs.pred *Absolute population conditional weighted residuals vs population predictions for Xpose 4*

Description

This is a plot of absolute population conditional weighted residuals (ICWRESI) vs population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.cwres.vs.pred(object, idsdir = "up", type = "p", smooth = TRUE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>idsdir</code>	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
<code>type</code>	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns an xyplot of |CWRES| vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.bw\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov.bw\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.cwres.vs.pred(xpdb)

## A conditioning plot
absval.cwres.vs.pred(xpdb, by="HCTZ")

## Custom heading and axis labels
absval.cwres.vs.pred(xpdb, main="My conditioning plot", ylb="|CWRES|", xlb="PRED")

## Custom colours and symbols, no IDs
```

```
absval.cwres.vs.pred(xpdb, cex=0.6, pch=3, col=1, ids=FALSE)
```

```
absval.cwres.vs.pred.by.cov
```

Absolute value of the conditional weighted residuals vs. population predictions, conditioned on covariates, for Xpose 4

Description

This is a plot of absolute population conditional weighted residuals (ICWRESI) vs population predictions (PRED) conditioned by covariates, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.cwres.vs.pred.by.cov(
  object,
  covs = "Default",
  ylb = "|CWRESI|",
  type = "p",
  smooth = TRUE,
  idsdir = "up",
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>covs</code>	A vector of covariates to use in the plot. If "Default" the the covariates defined in <code>object@Prefs@Xvardef\$Covariates</code> are used.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>type</code>	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>idsdir</code>	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

The main argument is not supported owing to the multiple plots generated by the function.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See `compute.cwres` for details.

A wide array of extra options controlling xyplots are available. See `xpose.plot.default` for details.

Value

Returns a stack of xyplots of |CWRES| vs PRED, conditioned on covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`absval.cwres.vs.pred`, `xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `compute.cwres`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.iwres.cwres.vs.ipred.pred`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `dOFV.vs.cov()`, `dOFV.vs.id()`, `dOFV1.vs.dOFV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
absval.cwres.vs.pred.by.cov(simpraz.xpdb, covs=c("HCTZ", "WT"), max.plots.per.page=2)
```

```
absval.iwres.cwres.vs.ipred.pred
```

Absolute population weighted residuals vs population predictions, and absolute individual weighted residuals vs individual predictions, for Xpose 4

Description

This is a matrix plot of absolute population weighted residuals (ICWRESI) vs population predictions (PRED) and absolute individual weighted residuals (IIWRESI) vs individual predictions (IPRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `absval.cwres.vs.pred` and `absval.iwres.vs.ipred` functions.

Usage

```
absval.iwres.cwres.vs.ipred.pred(object, main = "Default", ...)
```

```
absval.iwres.wres.vs.ipred.pred(object, main = "Default", ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

The plots created by the `absval.wres.vs.pred` and `absval.iwres.vs.ipred` functions are presented side by side for comparison.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns a compound plot.

Functions

- `absval.iwres.wres.vs.ipred.pred()`: absolute population weighted residuals (IWRESI) vs population predictions (PRED) and absolute individual weighted residuals (IIWRESI) vs individual predictions (IPRED)

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[absval.wres.vs.pred](#), [absval.iwres.vs.ipred](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## A vanilla plot
absval.iwres.wres.vs.ipred.pred(xpdb)
absval.iwres.cwres.vs.ipred.pred(xpdb)

## Custom colours and symbols
absval.iwres.cwres.vs.ipred.pred(xpdb, cex=0.6, pch=8, col=1)
```

 absval.iwres.vs.cov.bw

box and whisker plots of the absolute value of the individual weighted residuals vs. covariates

Description

box and whisker plots of the absolute value of the individual weighted residuals vs. covariates

Usage

```
absval.iwres.vs.cov.bw(object, xlb = "|iWRES|", main = "Default", ...)
```

Arguments

object	An "xpose.data" object.
xlb	A string giving the label for the x-axis. NULL if none.
main	A string giving the plot title or NULL if none.
...	Other arguments passed to <code>xpose.panel.default</code> .

Value

An `xpose.multiple.plot` object

See Also

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

absval.iwres.vs.idv *absolute value of the individual weighted residuals vs. the independent variable*

Description

absolute value of the individual weighted residuals vs. the independent variable

Usage

```
absval.iwres.vs.idv(
  object,
  ylb = "|iWRES|",
  smooth = TRUE,
  idsdir = "up",
  type = "p",
  ...
)
```

Arguments

object	An "xpose.data" object.
ylb	A string giving the label for the y-axis. NULL if none.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
idsdir	a string indicating the directions of the extremes to include in labelling. Possible values are "up", "down" and "both".
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c" for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
...	Other arguments passed to <code>xpose.panel.default</code> .

Value

A lattice object

See Also

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`,

[cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#),
[cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#),
[dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#),
[dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#),
[ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#),
[iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#),
[parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#),
[wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#),
[xpose.VPC.categorical\(\)](#), [xpose4-package](#)

absval.iwres.vs.ipred *Absolute individual weighted residuals vs individual predictions for Xpose 4*

Description

This is a plot of absolute individual weighted residuals (IWRESI) vs individual predictions (IPRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.iwres.vs.ipred(
  object,
  ylb = "|IWRES|",
  type = "p",
  ids = FALSE,
  idsdir = "up",
  smooth = TRUE,
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>type</code>	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
<code>ids</code>	Should id values be displayed?
<code>idsdir</code>	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns an xyplot of |IWRES| vs IPRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#), [runsum](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.idv\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.iwres.vs.ipred(xpdb)

## A conditioning plot
absval.iwres.vs.ipred(xpdb, by="HCTZ")

## Custom heading and axis labels
absval.iwres.vs.ipred(xpdb, main="My conditioning plot", ylb="|IWRES|", xlb="IPRED")

## Custom colours and symbols, no IDs
absval.iwres.vs.ipred(xpdb, cex=0.6, pch=3, col=1, ids=FALSE)
```

```
absval.iwres.vs.ipred.by.cov
```

Absolute individual weighted residuals vs individual predictions, conditioned on covariates, for Xpose 4

Description

This is a plot of absolute individual weighted residuals (IWRESI) vs individual predictions (IPRED) conditioned by covariates, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.iwres.vs.ipred.by.cov(
  object,
  ylb = "|IWRESI|",
  idsdir = "up",
  type = "p",
  smooth = TRUE,
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>idsdir</code>	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
<code>type</code>	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns a stack of xyplots of |IWRES| vs IPRED, conditioned by covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[absval.iwres.vs.ipred](#), [xpose.plot.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.idv\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.iwres.vs.ipred.by.cov(xpdb)

## Custom axis labels
absval.iwres.vs.ipred.by.cov(xpdb, ylb="|IWRES|", xlb="IPRED")

## Custom colours and symbols, no IDs
absval.iwres.vs.ipred.by.cov(xpdb, cex=0.6, pch=3, col=1, ids=FALSE)

## End(Not run)
```

absval.iwres.vs.pred *Absolute individual weighted residuals vs population predictions or independent variable for Xpose 4*

Description

This is a plot of absolute individual weighted residuals (IIWRESI) vs individual predictions (PRED) or independent variable (IDV), specific functions in Xpose 4. These functions are wrappers encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.iwres.vs.pred(
  object,
  ylb = "|IWRES|",
  smooth = TRUE,
  idsdir = "up",
  type = "p",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>idsdir</code>	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
<code>type</code>	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns an xyplot of `|IWRESI|` vs `PRED` or `|IWRESI|` vs `IDV`.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.iwres.vs.pred(xpdb)

## A conditioning plot
absval.iwres.vs.pred(xpdb, by="HCTZ")

## Custom heading and axis labels
absval.iwres.vs.pred(xpdb, main="My conditioning plot", ylb="|IWRES|", xlb="PRED")

## Custom colours and symbols, no IDs
absval.iwres.vs.pred(xpdb, cex=0.6, pch=3, col=1, ids=FALSE)
```

Description

This creates a stack of box and whisker plot of absolute population weighted residuals (IWRESI or liWRESI) vs covariates. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from the `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.wres.vs.cov.bw(object, xlb = "|WRESI", main = "Default", ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>xlb</code>	A string giving the label for the x-axis. NULL if none.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>xpose.plot.bw</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

A wide array of extra options controlling box-and-whisker plots are available. See `xpose.plot.bw` for details.

Value

Returns a stack of box-and-whisker plots of |WRESI| vs covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.bw`, `xpose.panel.bw`, `bwplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`,

```

parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(),
wres.vs.idv(), wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(),
xpose.VPC.categorical(), xpose4-package

```

Examples

```

## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.wres.vs.cov.bw(xpdb)

## A custom plot
absval.wres.vs.cov.bw(xpdb, bwdotcol="white",
  bwdotpch=15,
  bwreccol="red",
  bwrecfill="red",
  bwumbcol="red",
  bwoutpch=5,
  bwoutcol="black")

## A vanilla plot using IWRES
absval.iwres.vs.cov.bw(xpdb)

## End(Not run)

```

absval.wres.vs.idv *Absolute value of (C)WRES vs. independent variable plot in Xpose4.*

Description

This is a plot of the absolute value of the CWRES (default, other residuals as an option) vs independent variable, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the [xpose.plot.default](#) function. Most of the options take their default values from the `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```

absval.wres.vs.idv(
  object,
  idv = "idv",
  wres = "Default",

```

```

    ylb = "Default",
    smooth = TRUE,
    idsdir = "up",
    type = "p",
    ...
)

```

Arguments

object	An xpose.data object.
idv	the independent variable.
wres	Which weighted residual to use. "Default" is the CWRES.
ylb	Y-axis label.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
idsdir	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
type	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns an xyplot of $|CWRES|$ vs `idv` (often `TIME`, defined by `xvardef`).

Author(s)

Andrew Hooker

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class help](#), ~~~

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.idv\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#),

```
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(),
parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(),
wres.vs.idv(), wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(),
xpose.VPC.categorical(), xpose4-package
```

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.wres.vs.idv(xpdb)

## A conditioning plot
absval.wres.vs.idv(xpdb, by="HCTZ")

## Custom heading and axis labels
absval.wres.vs.idv(xpdb, main="Hello World", ylb="|CWRES|", xlb="IDV")

## Custom colours and symbols
absval.wres.vs.idv(xpdb, cex=0.6, pch=3, col=1)

## using the NPDEs instead of CWRES
absval.wres.vs.idv(xpdb, wres="NPDE")

## subsets
absval.wres.vs.idv(xpdb, subset="TIME<10")
```

absval.wres.vs.pred	<i>Absolute population weighted residuals vs population predictions for Xpose 4</i>
---------------------	---

Description

This is a plot of absolute population weighted residuals (IWRESI) vs population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.wres.vs.pred(
  object,
  ylb = "|WRES|",
  idsdir = "up",
  type = "p",
  smooth = TRUE,
  ...
)
```

Arguments

object	An xpose.data object.
ylb	A string giving the label for the y-axis. NULL if none.
idsdir	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
type	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns an xyplot of |WRES| vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#),

```
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(),
parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(),
wres.vs.idv(), wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(),
xpose.VPC.categorical(), xpose4-package
```

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.wres.vs.pred(xpdb)

## A conditioning plot
absval.wres.vs.pred(xpdb, by="HCTZ")

## Custom heading and axis labels
absval.wres.vs.pred(xpdb, main="My conditioning plot",
  ylb="|WRES|", xlb="PRED")

## Custom colours and symbols
absval.wres.vs.pred(xpdb, cex=0.6, pch=19, col=1,
  smcol="blue", smlty=2)
```

```
absval.wres.vs.pred.by.cov
```

*Absolute population weighted residuals vs population predictions,
conditioned on covariates, for Xpose 4*

Description

This is a plot of absolute population weighted residuals (IWRESI) vs population predictions (PRED) conditioned by covariates, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
absval.wres.vs.pred.by.cov(
  object,
  ylb = "|WRES|",
  type = "p",
  smooth = TRUE,
  ids = FALSE,
  idsdir = "up",
  main = "Default",
  ...
)
```

Arguments

object	An xpose.data object.
ylb	A string giving the label for the y-axis. NULL if none.
type	Type of plot. The default is points only ("p"), but lines ("l") and both ("b") are also available.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
ids	Logical. Should id labels on points be shown?
idsdir	Direction for displaying point labels. The default is "up", since we are displaying absolute values.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns a stack of xyplots of |WRES| vs PRED, conditioned on covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[absval.wres.vs.pred](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval_delta_vs_cov_model_comp](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
absval.wres.vs.pred.by.cov(xpdb)

## Custom axis labels
absval.wres.vs.pred.by.cov(xpdb, ylb="|CWRES|", xlb="PRED")

## Custom colours and symbols, IDs
absval.wres.vs.pred.by.cov(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

## End(Not run)
```

absval_delta_vs_cov_model_comp

Model comparison plots, of absolute differences in goodness-of-fit predictors against covariates, for Xpose 4

Description

These functions plot absolute differences in PRED, IPRED, WRES, CWRES and IWRES against covariates for two specified model fits.

Usage

```
absval.dcwres.vs.cov.model.comp(  
  object,  
  object.ref = NULL,  
  type = NULL,  
  ylb = expression(paste("|", Delta, "CWRES|")),  
  main = "Default",  
  ...  
)
```

```
absval.dipred.vs.cov.model.comp(  
  object,  
  object.ref = NULL,  
  type = NULL,  
  ylb = expression(paste("|", Delta, "IPRED|")),  
  main = "Default",  
  ...  
)
```

```
absval.diwres.vs.cov.model.comp(  
  object,  
  object.ref = NULL,  
  type = NULL,  
  ylb = expression(paste("|", Delta, "IWRES|")),  
  main = "Default",  
  ...  
)
```

```
absval.dpred.vs.cov.model.comp(  
  object,  
  object.ref = NULL,  
  type = NULL,  
  ylb = expression(paste("|", Delta, "PRED|")),  
  main = "Default",  
  ...  
)
```

```
absval.dwres.vs.cov.model.comp(  
  object,  
  object.ref = NULL,  
  type = NULL,  
  ylb = expression(paste("|", Delta, "WRES|")),  
  main = "Default",
```

```
    ...
  )
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>object.ref</code>	An <code>xpose.data</code> object. If not supplied, the user will be prompted.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Conditional weighted residuals (CWRES) may require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Value

Returns a stack of plots comprising comparisons of PRED, IPRED, WRES (or CWRES) and IWRES for the two specified runs.

Functions

- `absval.dcwres.vs.cov.model.comp()`: The absolute differences in individual predictions against covariates for two specified model fits.
- `absval.dipred.vs.cov.model.comp()`: The absolute differences in individual predictions against covariates for two specified model fits.
- `absval.diwres.vs.cov.model.comp()`: The absolute differences in individual weighted residuals against covariates for two specified model fits.
- `absval.dpred.vs.cov.model.comp()`: The absolute differences in population predictions against covariates for two specified model fits.
- `absval.dwres.vs.cov.model.comp()`: The absolute differences in population weighted residuals against covariates for two specified model fits.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.panel.default`, `xyplot`, `compute.cwres`, `xpose.prefs-class`,
`xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`,
`absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`,
`absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`,
`absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `addit.gof()`,
`autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`,
`cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`,
`cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`,
`cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`,
`dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`,
`dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`,
`ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`,
`iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`,
`parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`,
`wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`,
`xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for runs
## 5 and 6 in the current working directory
xpdb5 <- xpose.data(5)
xpdb6 <- xpose.data(6)

## A basic dWRES plot, without prompts
absval.dwres.vs.cov.model.comp(xpdb5, xpdb6)

## Custom colours and symbols, no user IDs
absval.dpred.vs.cov.model.comp(xpdb5, xpdb6, cex=0.6, pch=8, col=1, ids=NULL)

## End(Not run)
```

add.grid.table

Print tables or text in a grid object

Description

These functions take an array of values and labels or an array of text and add it to one or many grid viewports in an orderly fashion.

Usage

```

add.grid.table(
  txt,
  col.nams = NULL,
  ystart,
  xstart = unit(0, "npc"),
  start.pt = 1,
  vp,
  vp.num = 1,
  minrow = 5,
  cell.padding = 0.5,
  mult.col.padding = 1,
  col.optimize = TRUE,
  equal.widths = FALSE,
  space.before.table = 1,
  center.table = FALSE,
  use.rect = FALSE,
  fill.type = NULL,
  fill.col = "grey",
  cell.lines.lty = 0,
  ...
)

```

Arguments

<code>txt</code>	The text or table values to add to the grid object.
<code>col.nams</code>	the column names of the table values
<code>ystart</code>	The y location to start printing in the grid viewport
<code>xstart</code>	The x location to start printing in the grid viewport
<code>start.pt</code>	The start point (row) in the table array to start printing
<code>vp</code>	The viewport(s) to add the table or text to
<code>vp.num</code>	the viewport number in <code>vp</code> to start printing to
<code>minrow</code>	The minimum rows before printing more columns to use in the table
<code>cell.padding</code>	padding between cells in the table
<code>mult.col.padding</code>	padding between multiple columns in the table
<code>col.optimize</code>	should we column optimize (TRUE) or row optimize (FALSE)
<code>equal.widths</code>	Should all columns have equal widths
<code>space.before.table</code>	Should there be a space before the table
<code>center.table</code>	should we center the table in the viewport?
<code>use.rect</code>	Should we make rectangles with background color around the table entries TRUE or FALSE

fill.type	Which rectangles should be filled. Allowed values are "all", "top", "side", "both" and NULL.
fill.col	The color of the filled rectangles
cell.lines.lty	The line-type for the lines between the cells, using the same values as lty.
...	Other arguments passed to the various functions.

Value

A List is returned with the following components

ystart	new starting point for new text
stop.pt	null if everything gets printed
vp.num	the viewport needed for next text printed
xpose.table	A grob object that can be plotted.

Author(s)

Andrew Hooker

See Also

[runsum](#), [grid.text](#)

add.model.comp

Additional model comparison plots, for Xpose 4

Description

This creates a stack of four plots, comparing absolute values of PRED, absolute values of IPRED, delta CWRES (or WRES) and delta IWRES estimates for the two specified model fits.

Usage

```
add.model.comp(
  object,
  object.ref = NULL,
  onlyfirst = FALSE,
  inclZeroWRES = FALSE,
  subset = xsubset(object),
  main = "Default",
  force.wres = FALSE,
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>object.ref</code>	An <code>xpose.data</code> object. If not supplied, the user will be prompted.
<code>onlyfirst</code>	Logical value indicating whether only the first row per individual is included in the plot.
<code>inclZeroWRES</code>	Logical value indicating whether rows with <code>WRES=0</code> is included in the plot. The default is <code>TRUE</code> .
<code>subset</code>	A string giving the subset expression to be applied to the data before plotting. See xsubset .
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or <code>NULL</code> for no plot title.
<code>force.wres</code>	Should we use the <code>WRES</code> in the plots instead of <code>CWRES</code> (logical <code>TRUE</code> or <code>FALSE</code>)
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Four model comparison plots are displayed in sequence.

Conditional weighted residuals (`CWRES`) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling `xyplots` are available. See [xpose.plot.default](#) for details.

Value

Returns a stack of plots comprising comparisons of absolute values of `PRED`, absolute values of `IPRED`, absolute differences in `CWRES` (or `WRES`) and absolute differences in `IWRES` for the two specified runs.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [compute.cwres](#), [xpose.prefs-class](#), [xpose.data-class](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for runs
## 5 and 6 in the current working directory
xpdb5 <- xpose.data(5)
xpdb6 <- xpose.data(6)

## A vanilla plot, without prompts
add.model.comp(xpdb5, xpdb6, prompt = FALSE)
```

```
## Custom colours and symbols, no user IDs
add.model.comp(xpdb5, xpdb6, cex=0.6, pch=8, col=1, ids=NULL)

## End(Not run)
```

 addit.gof

Additional goodness-of-fit plots, for Xpose 4

Description

This is a compound plot consisting of plots of weighted population residuals (WRES) vs population predictions (PRED), absolute individual weighted residuals (IWRESI) vs independent variable (IDV), WRES vs IDV, and weighted population residuals vs log(IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `wres.vs.pred`, `iwres.vs.idv` and `wres.vs.idv` functions.

Usage

```
addit.gof(
  object,
  type = "p",
  title.size = 0.02,
  title.just = c("center", "top"),
  main = "Default",
  force.wres = FALSE,
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see <code>'plot'</code> : <code>"p"</code> for points, <code>"l"</code> for lines, <code>"o"</code> for over-plotted points and lines, <code>"b"</code> , <code>"c"</code> for (empty if <code>"c"</code>) points joined by lines, <code>"s"</code> and <code>"S"</code> for stair steps and <code>"h"</code> for histogram-like vertical lines. Finally, <code>"n"</code> does not produce any points or lines.
<code>title.size</code>	Amount, in a range of 0-1, of how much space the title should take up in the plot)
<code>title.just</code>	how the title should be justified
<code>main</code>	The title of the plot. If <code>"Default"</code> then a default title is plotted. Otherwise the value should be a string like <code>"my title"</code> or <code>NULL</code> for no plot title.
<code>force.wres</code>	Plot the WRES even if other residuals are available.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Four additional goodness-of-fit plots are presented side by side for comparison.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.multiple.plot.default](#) for details.

Value

Returns a compound plot comprising plots of weighted population residuals (WRES) vs population predictions (PRED), absolute individual weighted residuals (IWRES) vs independent variable (IDV), WRES vs IDV, and weighted population residuals vs log(IDV).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[wres.vs.pred](#), [iwres.vs.idv](#), [wres.vs.idv](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## A vanilla plot
addit.gof(xpdb)
```

`add_transformed_columns`*Column-transformation functions for Xpose 4*

Description

These functions transform existing Xpose 4 data columns, adding new columns.

Usage

```
add.absval(object, listall = TRUE, classic = FALSE)
```

```
add.dichot(object, listall = TRUE, classic = FALSE)
```

```
add.exp(object, listall = TRUE, classic = FALSE)
```

```
add.log(object, listall = TRUE, classic = FALSE)
```

```
add.tad(object, classic = FALSE)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>listall</code>	A logical operator specifying whether the items in the database should be listed.
<code>classic</code>	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

These functions may be used to create new data columns within the Xpose data object by transforming existing ones.

Value

An `xpose.data` object (`classic == FALSE`) or null (`classic == TRUE`).

Functions

- `add.absval()`: Create a column containing the absolute values of data in another column.
- `add.dichot()`: Create a categorical data column based on a continuous data column
- `add.exp()`: Create an exponentiated version of an existing variable
- `add.log()`: Create a log transformation of an existing variable
- `add.tad()`: Create a time-after-dose (TAD) data item based upon the dose and time variables in the dataset.

Author(s)

Niclas Jonsson, Justin Wilkins and Andrew Hooker

See Also

[xpose.data](#)

Other data functions: [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Create a column containing the absolute values of data in another
## column
add.absval(xpdb5)

## Create a categorical data column based on a continuous data column,
## and do not list variables
add.dichot(xpdb5, listall = FALSE)

## Create a column containing the exponentiated values of data in
## another column
add.exp(xpdb5)

## Create a column containing log-transformations of data in another
## column
add.log(xpdb5)

## Create a time-after-dose column
add.tad(xpdb5)

## End(Not run)
```

autocorr.cwres

Autocorrelation of conditional weighted residuals for Xpose 4

Description

This is an autocorrelation plot of conditional weighted residuals, a specific function in Xpose 4. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
autocorr.cwres(
  object,
  type = "p",
  smooth = TRUE,
  ids = F,
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see plot : "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
<code>smooth</code>	Logical value indicating whether a smooth should be superimposed.
<code>ids</code>	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the <code>idlab</code> <code>xpose</code> data variable).
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

Value

Returns an autocorrelation plot for conditional weighted population residuals (CWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xyplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_co](#)

```

addit.gof(), autocorr.iwres(), autocorr.wres(), basic.gof(), basic.model.comp(), cat.dv.vs.idv.sb(),
cat.pc(), cov.splom(), cwres.dist.hist(), cwres.dist.qq(), cwres.vs.cov(), cwres.vs.idv(),
cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(), cwres.wres.vs.pred(),
dOFV.vs.cov(), dOFV.vs.id(), dOFV1.vs.dOFV2(), data.checkout(), dv.preds.vs.idv(),
dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(), dv.vs.pred(),
dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(),
ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(), iwres.dist.qq(), iwres.vs.idv(),
kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
autocorr.cwres(xpdb)

## A conditioning plot
autocorr.cwres(xpdb, dilution=TRUE)

## Custom heading and axis labels
autocorr.cwres(xpdb, main="My conditioning plot", ylb="|CWRES|", xlb="PRED")

## Custom colours and symbols, IDs
autocorr.cwres(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

```

autocorr.iwres

autocorrelation of the individual weighted residuals

Description

autocorrelation of the individual weighted residuals

Usage

```
autocorr.iwres(
  object,
  type = "p",
  smooth = TRUE,
  ids = F,
  main = "Default",
  ...
)
```

Arguments

object	An "xpose.data" object.
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
ids	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the idlab xpose data variable).
main	A string giving the plot title or NULL if none.
...	Other arguments passed to <code>xpose.panel.default</code> .

Value

A Lattice object

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_cov\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

autocorr.wres

Autocorrelation of weighted residuals for Xpose 4

Description

This is an autocorrelation plot of weighted residuals. Most of the options take their default values from the `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
autocorr.wres(
  object,
  type = "p",
  smooth = TRUE,
  ids = F,
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see plot : "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
<code>smooth</code>	Logical value indicating whether a smooth should be superimposed.
<code>ids</code>	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the <code>idlab</code> <code>xpose</code> data variable).
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling `xyplots` are available. See [xpose.plot.default](#) for details.

Value

Returns an autocorrelation plot for weighted population residuals (WRES) or individual weighted residuals (IWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Justin Wilkins & Andrew Hooker

See Also

[xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## End(Not run)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
autocorr.wres(xpdb)

## A conditioning plot
autocorr.wres(xpdb, dilution=TRUE)

## Custom heading and axis labels
autocorr.wres(xpdb, main="My conditioning plot", ylb="|CWRES|", xlb="PRED")

## Custom colours and symbols, IDs
autocorr.wres(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

## A vanilla plot with IWRES
autocorr.iwres(xpdb)
```

 basic.gof

Basic goodness-of-fit plots, for Xpose 4

Description

This is a compound plot consisting of plots of observations (DV) vs population predictions (PRED), observations (DV) vs individual predictions (IPRED), absolute individual weighted residuals (IIWRESI) vs IPRED, and weighted population residuals (CWRES) vs independent variable (IDV), a specific function in Xpose 4. WRES are also supported. It is a wrapper encapsulating arguments to the `dv.vs.pred`, `dv.vs.ipred`, `absval.iwres.vs.ipred` and `wres.vs.idv` functions.

Usage

```
basic.gof(object, force.wres = FALSE, main = "Default", use.log = FALSE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>force.wres</code>	Should the plots use WRES? Values can be TRUE/FALSE. Otherwise the CWRES are used if present.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>use.log</code>	Should we use log transformations in the plots?
<code>...</code>	Other arguments passed to xpose.plot.default .

Details

Four basic goodness-of-fit plots are presented side by side for comparison.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) for details.

`basic.gof.cwres` is just a wrapper for `basic.gof` with `use.cwres=TRUE`.

Value

Returns a compound plot comprising plots of observations (DV) vs population predictions (PRED), DV vs individual predictions (IPRED), absolute individual weighted residuals (IIWRESI) vs IPRED, and weighted populations residuals (WRES) vs the independent variable (IDV).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[dv.vs.pred](#), [dv.vs.ipred](#), [absval.iwres.vs.ipred](#), [wres.vs.idv](#), [cwres.vs.idv](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [compute.cwres](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
basic.gof(simpraz.xpdb)
```

```
basic.model.comp
```

Basic model comparison plots, for Xpose 4

Description

This creates a stack of four plots, comparing PRED, IPRED, WRES (or CWRES), and IWRES estimates for the two specified model fits.

Usage

```
basic.model.comp(
  object,
  object.ref = NULL,
  onlyfirst = FALSE,
  inclZeroWRES = FALSE,
  subset = xsubset(object),
  main = "Default",
  force.wres = FALSE,
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>object.ref</code>	An <code>xpose.data</code> object. If not supplied, the user will be prompted.
<code>onlyfirst</code>	Logical value indicating whether only the first row per individual is included in the plot.
<code>inclZeroWRES</code>	Logical value indicating whether rows with <code>WRES=0</code> is included in the plot. The default is <code>TRUE</code> .
<code>subset</code>	A string giving the subset expression to be applied to the data before plotting. See xsubset .
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or <code>NULL</code> for no plot title.
<code>force.wres</code>	Force function to use <code>WRES</code> ?
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Four basic model comparison plots are displayed in sequence.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling `xyplots` are available. See [xpose.plot.default](#) for details.

Value

Returns a stack of plots comprising comparisons of `PRED`, `IPRED`, `WRES` (or `CWRES`) and `IWRES` for the two specified runs.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [compute.cwres](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_compare.addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#),

```

kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Not run:
## We expect to find the required NONMEM run and table files for runs
## 5 and 6 in the current working directory
xpdb5 <- xpose.data(5)
xpdb6 <- xpose.data(6)

## A vanilla plot, without prompts
basic.model.comp(xpdb5, xpdb6, prompt = FALSE)

## Custom colours and symbols, no user IDs
basic.model.comp.cwres(xpdb5, xpdb6, cex=0.6, pch=8, col=1, ids=NULL)

## End(Not run)

```

boot.hist	<i>Function to create histograms of results from the bootstrap tool in PsN</i>
-----------	--

Description

Reads results from the bootstrap tool in [PsN](https://uupharmacometrics.github.io/PsN/PsN) and then creates histograms.

Usage

```

boot.hist(
  results.file = "raw_results_run1.csv",
  incl.ids.file = "included_individuals1.csv",
  min.failed = FALSE,
  cov.failed = FALSE,
  cov.warnings = FALSE,
  boundary = FALSE,
  showOriginal = TRUE,
  showMean = FALSE,
  showMedian = FALSE,
  showPCTS = FALSE,
  PCTS = c(0.025, 0.975),
  excl.id = c(),
  layout = NULL,
  sort.plots = TRUE,

```

```

    main = "Default",
    ...
)

```

Arguments

results.file	The location of the results file from the bootstrap tool in PsN
incl.ids.file	The location of the included ids file from the bootstrap tool in PsN
min.failed	Should NONMEM runs that had failed minimization be skipped? TRUE or FALSE
cov.failed	Should NONMEM runs that had a failed covariance step be skipped? TRUE or FALSE
cov.warnings	Should NONMEM runs that had covariance step warnings be skipped? TRUE or FALSE
boundary	Should NONMEM runs that had boundary warnings be skipped? TRUE or FALSE
showOriginal	Should we show the value from the original NONMEM run in the histograms? TRUE or FALSE
showMean	Should we show the mean of the histogram data? TRUE or FALSE
showMedian	Should we show the median of the histogram data? TRUE or FALSE
showPCTS	Should we show the percentiles of the histogram data? TRUE or FALSE
PCTS	the percentiles to show. Can be a vector of any length. For example, <code>c(0.05, 0.2, 0.5, 0.7)</code>
excl.id	Vector of id numbers to exclude.
layout	Layout of plots. A vector of number of rows and columns in each plot. <code>c(3, 3)</code> for example.
sort.plots	Should the plots be sorted based on type of parameter. Sorting on parameters, standard errors, shrinkage and eigenvalues.
main	The title of the plot.
...	Additional arguments that can be passed to xpose.plot.histogram , xpose.panel.histogram , histogram and other lattice-package functions.

Value

A lattice object

Author(s)

Andrew Hooker

References

[PsN](#)

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#) and other [lattice-package](#) functions.

Other [PsN](#) functions: [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
boot.hist(results.file="./boot1/raw_results_run1.csv",
          incl.ids.file="./boot1/included_individuals1.csv")

## End(Not run)
```

bootgam.print	<i>Print summary information for a bootgam or bootscm</i>
---------------	---

Description

This functions prints some summary information for a bootgam performed in Xpose, or for a bootscm performed in PsN.

Usage

```
bootgam.print(bootgam.obj = NULL)
```

Arguments

bootgam.obj The bootgam or bootscm object.

Value

No value returned

Author(s)

Ron Keizer

Examples

```
## Not run:
bootgam.print(current.bootgam) # Print summary for the current Xpose bootgam object
bootgam.print(current.bootscm) # Print summary for the current Xpose bootscm object

## End(Not run)
```

bootscm.import	<i>Import bootscm data into R/Xpose</i>
----------------	---

Description

This function imports data generated by the PsN `boot_scm` function into the Xpose / R environment.

Usage

```
bootscm.import(
  scm.folder = NULL,
  silent = FALSE,
  n.bs = NULL,
  cov.recoding = NULL,
  group.by.cov = NULL,
  skip.par.est.import = FALSE,
  dofv.forward = 3.84,
  dofv.backward = 6.64,
  runno = NULL,
  return.obj = FALSE
)
```

Arguments

<code>scm.folder</code>	The folder in which the PsN-generated bootscm data are.
<code>silent</code>	Don't output any progress report. Default is FALSE.
<code>n.bs</code>	The number of bootstraps performed. Defaults to 100.
<code>cov.recoding</code>	For categorical covariates that are recoded to dichotomous covariates within the bootscm configuration file, a list can be specified containing data frames for recoding. See the example below for details.
<code>group.by.cov</code>	Group inclusion frequencies by covariate, instead of calculating them per parameter-covariates relationship. Default is NULL, which means that the user will be asked to make a choice.
<code>skip.par.est.import</code>	Skip the import of all parameter estimates (in each final model in all scm's, as well as parameter estimates in first step of each scm). These data are required to make plot that show inclusion bias and correlation in parameter estimates. Importing these data takes a bit of time (may take a minute or so), so if you don't intend to make these plots anyhow this step can be skipped. Default is FALSE.
<code>dofv.forward</code>	dOFV value used in forward step of scm.
<code>dofv.backward</code>	dOFV value used in backward step of scm.
<code>runno</code>	The run-number of the base model for this bootSCM.
<code>return.obj</code>	Should the bootscm object be returned by the function?

Author(s)

Ron Keizer

See Also

Other bootscm: [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

Other PsN functions: [boot.hist\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

cat.dv.vs.idv.sb

*Categorical observations vs. independent variable using stacked bars.***Description**

Categorical observations vs. independent variable using stacked bars.

Usage

```
cat.dv.vs.idv.sb(
  object,
  dv = xvardef("dv", object),
  idv = xvardef("idv", object),
  by = NULL,
  groups = dv,
  force.by.factor = FALSE,
  recur = F,
  xlb = idv,
  ylb = "Proportion",
  subset = NULL,
  vary.width = T,
  level.to.plot = NULL,
  refactor.levels = TRUE,
  main = xpose.create.title.text(idv, dv, "Proportions of", object, subset = subset, ...),
  stack = TRUE,
  horizontal = FALSE,
  strip = function(...) strip.default(..., strip.names = c(TRUE, TRUE)),
  scales = list(),
  inclZeroWRES = TRUE,
  onlyfirst = FALSE,
  samp = NULL,
  aspect = object@Prefs@Graph.prefs$aspect,
  auto.key = "Default",
  mirror = FALSE,
  mirror.aspect = "fill",
  pass.plot.list = FALSE,
```

```

    x.cex = NULL,
    y.cex = NULL,
    main.cex = NULL,
    mirror.internal = list(strip.missing = missing(strip)),
    ...
)

```

Arguments

<code>object</code>	Xpose data object.
<code>dv</code>	The dependent variable (e.g. "DV" or "CP".)
<code>idv</code>	The independent variable (e.g. "TIME".)
<code>by</code>	Conditioning variable
<code>groups</code>	How we should group values in each conditional plot.
<code>force.by.factor</code>	Should we force the data to be treated as factors?
<code>recur</code>	Not used.
<code>xlb</code>	A string giving the label for the x-axis. NULL if none.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>subset</code>	Subset of data.
<code>vary.width</code>	Should we vary the width of the bars to match amount of information?
<code>level.to.plot</code>	Which levels of the DV to plot.
<code>refactor.levels</code>	Should we refactor the levels?
<code>main</code>	The title of the plot.
<code>stack</code>	Should we stack the bars?
<code>horizontal</code>	Should the bars be horizontal?
<code>strip</code>	Defining how the strips should appear in the conditioning plots.
<code>scales</code>	Scales argument to xyplot .
<code>inclZeroWRES</code>	Include rows with WRES=0?
<code>onlyfirst</code>	Only include first data point for each individual?
<code>samp</code>	Sample to use in mirror plot (a number).
<code>aspect</code>	Aspect argument to xyplot .
<code>auto.key</code>	Make a legend.
<code>mirror</code>	Mirror can be FALSE, TRUE, 1 or 3.
<code>mirror.aspect</code>	Aspect for mirror.
<code>pass.plot.list</code>	Should the plot list be passed back to user?
<code>x.cex</code>	Size of x axis label.
<code>y.cex</code>	Size of Y axis label.
<code>main.cex</code>	Size of Title.
<code>mirror.internal</code>	Internal stuff.
<code>...</code>	Other arguments passed to function.

Author(s)

Andrew Hooker

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## read in table files
runno <- 45
xpdb <- xpose.data(runno)

## make some stacked bar plots
cat.dv.vs.idv.sb(xpdb, idv=NULL, stack=F)
cat.dv.vs.idv.sb(xpdb, idv=NULL, stack=F, by="DOSE")
cat.dv.vs.idv.sb(xpdb, idv="DOSE")
cat.dv.vs.idv.sb(xpdb, idv=NULL, stack=F, by="TIME")
cat.dv.vs.idv.sb(xpdb, idv="TIME")
cat.dv.vs.idv.sb(xpdb, idv="CAVH")
cat.dv.vs.idv.sb(xpdb, idv="TIME", by="DOSE", scales=list(x=list(rot=45)))

## make some mirror plots
cat.dv.vs.idv.sb(xpdb, idv="DOSE", mirror=1)
cat.dv.vs.idv.sb(xpdb, idv="CAVH", mirror=1, auto.key=F)

## End(Not run)
```

Description

Categorical (visual) predictive check plots.

Usage

```
cat.pc(
  object,
  dv = xvardef("dv", object),
  idv = xvardef("idv", object),
  level.to.plot = NULL,
  subset = NULL,
  histo = T,
  median.line = F,
  PI.lines = F,
  xlb = if (histo) {
    paste("Proportion of ", dv)
  } else {
    paste(idv)
  },
  ylb = if (histo) {
    paste("Percent of Total")
  } else {
    paste("Proportion of Total")
  },
  main = xpose.create.title.text(NULL, dv, "Predictive check of", object, subset =
    subset, ...),
  strip = "Default",
  ...
)
```

Arguments

object	Xpose data object.
dv	The dependent variable (e.g. "DV" or "CP".)
idv	The independent variable (e.g. "TIME".)
level.to.plot	The levels to plot.
subset	Subset of data.
histo	If FALSE then a VPC is created, given that idv is defined.
median.line	Make a median line?
PI.lines	Make prediction interval lines?
xlb	Label for x axis.
ylb	label for y axis.
main	Main title.
strip	Defining how the strips should appear in the conditioning plots.
...	Extra arguments passed to the function.

Author(s)

Andrew C. Hooker

See Also

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `dOFV.vs.cov()`, `dOFV.vs.idv()`, `dOFV1.vs.dOFV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Not run:
## read in table files
runno <- 45
xpdb <- xpose.data(runno)

## create proportion (visual) predictive check
cat.pc(xpdb, idv=NULL)
cat.pc(xpdb, idv="DOSE")
cat.pc(xpdb, idv="DOSE", histo=F)
cat.pc(xpdb, idv="TIME", histo=T, level.to.plot=1)

## End(Not run)
```

change.parm

Change parameter scope.

Description

Function to change the parameter scope.

Usage

```
change.parm(object, listall = TRUE, classic = FALSE)
```

Arguments

object	The xpose data object.
listall	whether we should list all the current parameters.
classic	true if used in the classic menu system (for internal use).

Value

If classic then return nothing. Otherwise return the new data object.

Author(s)

Andrew C. Hooker

change.var.name	<i>Changes the name of an Xpose data item</i>
-----------------	---

Description

This function allows the names of data items in the Xpose database to be changed.

Usage

```
change.var.name(object, classic = FALSE)
```

Arguments

object	An xpose.data object.
classic	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

This function facilitates the changing of data item names in the object@Data slot.

Value

An [xpose.data](#) object.

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

[Data](#), [SData](#), [xpose.data](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

xpdb5 <- change.var.name(xpdb5)

## End(Not run)
```

change.xlabel	<i>Changes the label of an Xpose data item</i>
---------------	--

Description

This function allows the labels of data items in the Xpose database to be changed.

Usage

```
change.xlabel(object, listall = TRUE, classic = FALSE)
```

Arguments

object	An <code>xpose.data</code> object.
listall	A logical operator specifying whether the items in the database should be listed.
classic	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

This function facilitates the changing of data item labels in the `object@Prefs@Labels` slot.

Value

An `xpose.data` object.

Author(s)

Justin Wilkins

See Also

[Data](#), [SData](#), [xpose.data](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

xpdb5 <- change.xlabel(xpdb5)

## End(Not run)
```

change.xvardef	<i>Change Xpose variable definitions.</i>
----------------	---

Description

These functions allow for the changing of Xpose variable definitions like "idv" and "dv". These variable definitions are used to refer to columns of the observed data in a generic way, so that generic plotting functions can be created.

Usage

```
change.xvardef(
  object,
  var = ".ask",
  def = ".ask",
  listall = TRUE,
  classic = FALSE,
  check.var = FALSE,
  ...
)

change.xvardef(
  object,
  var,
  listall = FALSE,
  classic = FALSE,
  check.var = FALSE,
  ...
) <- value
```

Arguments

object	An xpose.data object.
var	The Xpose variable you would like to change or add to the current object. A one-element character vector (e.g. "idv"). If ".ask" the user will be prompted to input a value.

<code>def</code>	A vector of column names from NONMEM table files (<code>names(object@Data)</code>) that should be associated with this variable (e.g. <code>c("TIME")</code>). Multiple values are allowed. If <code>".ask"</code> the user will be prompted to input values.
<code>listall</code>	Should the function list the database values?
<code>classic</code>	Is the function being used from the classic interface. This is an internal option.
<code>check.var</code>	Should the variables be checked against the current variables in the object?
<code>...</code>	Items passed to functions within this function.
<code>value</code>	a vector of values

Value

If called from the the command line then this function returns an xpose database. If called from the classic interface this function updates the current xpose database (`.cur.db`).

Functions

- `change.xvardef(object, var, listall = FALSE, classic = FALSE, check.var = FALSE, ...)` <- value: Change the covariate scope of the xpose database object

Additional arguments

The default xpose variables are:

id Individual identifier column in dataset

idlab values used for plotting ID values on data points in plots

occ The occasion variable

dv The dv variable

pred The pred variable

ipred The ipred variable

wres The wres variable

cwres The cwres variable

res The res variable

parms The parameters in the database

covariates The covariates in the database

ranpar The random parameters in the database

Author(s)

Andrew Hooker

See Also

[xvardef](#), [xpose.data](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

# Change the "id" variable to point to "PRED" in the xpose object
xpdb <- change.xvardef(xpdb,var="id",def="PRED")

# Check the value of the "id" variable
xvardef("id",xpdb)

# Change the "idv" variable
change.xvardef(xpdb,var="idv") <- "TIME"

# Change the covariate scope
change.xvardef(xpdb,var="covariates") <- c("SEX","AGE","WT")

## Not run:
# Use the interactive capabilities of the function
xpdb <- change.xvardef(xpdb)

## End(Not run)
```

change_graphical_parameters

Functions changing variable definitions in Xpose 4

Description

These functions allow customization of Xpose's graphics settings.

Usage

```
change.ab.graph.par(object, classic = FALSE)

change.bw.graph.par(object, classic = FALSE)

change.cond.graph.par(object, classic = FALSE)

change.dil.graph.par(object, classic = FALSE)

change.label.par(object, classic = FALSE)

change.lm.graph.par(object, classic = FALSE)

change.misc.graph.par(object, classic = FALSE)

change.pi.graph.par(object, classic = FALSE)
```

```
change.smooth.graph.par(object, classic = FALSE)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>classic</code>	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

Settings can be saved and loaded using `export.graph.par` and `import.graph.par`, respectively.

Value

An `xpose.data` object (`classic == FALSE`) or null (`classic == TRUE`).

Functions

- `change.ab.graph.par()`: change settings for the line of identity.
- `change.bw.graph.par()`: sets preferences for box-and-whisker plots
- `change.cond.graph.par()`: sets preferences for conditioning
- `change.dil.graph.par()`: responsible for dilution preferences
- `change.label.par()`: responsible for labelling preferences
- `change.lm.graph.par()`: responsible for linear regression lines.
- `change.misc.graph.par()`: sets basic graphics parameters, including plot type, point type and size, colour, line type, and line width.
- `change.pi.graph.par()`: responsible for prediction interval plotting preferences
- `change.smooth.graph.par()`: sets preferences for loess smooths.

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.panel.default`, `xpose.plot.bw`, `xpose.panel.bw`, `xpose.plot.default`, `import.graph.p`, `export.graph.par`, `plot.default`, `par`, `import.graph.par`, `panel.abline`, `panel.lmline`, `lm`, `panel.loess`, `loess.smooth`, `loess`, `panel.bwplot`, `shingle`, `reorder.factor`

Other data functions: `add_transformed_columns`, `change_misc_parameters`, `compute.cwres()`, `data.checkout()`, `data_extract_or_assign`, `db.names()`, `export.graph.par()`, `export.variable.definitions()`, `import.graph.par()`, `import.variable.definitions()`, `make.sb.data()`, `nsim()`, `par_cov_summary`, `read.TTE.sim.data()`, `read.nm.tables()`, `read_NM_output`, `read_nm_table()`, `simprazExample()`, `tabulate.parameters()`, `xlabel()`, `xpose.data`, `xpose.print()`, `xpose4-package`, `xsubset()`

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Change default miscellaneous graphic preferences
xpdb5 <- change.misc.graph.par(xpdb5)

## Change default linear regression line preferences, creating a new
## object
xpdb5.a <- change.lm.graph.par(xpdb5)

## Change conditioning preferences
xpdb5 <- change.cond.graph.par(xpdb5)

## End(Not run)
```

change_misc_parameters

Functions changing miscellaneous parameter settings in Xpose 4

Description

These functions allow viewing and changing of settings relating to subsets, categorical threshold values, documentation and numbers indicating missing data values.

Usage

```
change.cat.cont(
  object,
  listall = TRUE,
  classic = FALSE,
  to.cat.vec = NULL,
  to.cont.vec = NULL,
  change.type.vec = NULL,
  ...
)
```

```
change.cat.cont(
  object,
  listall = TRUE,
  classic = FALSE,
  to.cat.vec = NULL,
  to.cont.vec = NULL,
```

```

...
) <- value

change.cat.levels(object, classic = FALSE, cat.limit = NULL, ...)

change.cat.levels(object, classic = FALSE, ...) <- value

change.dv.cat.levels(object, classic = FALSE, dv.cat.limit = NULL, ...)

change.dv.cat.levels(object, classic = FALSE, ...) <- value

change.miss(object, classic = FALSE)

change.subset(object, classic = FALSE)

get.doc(object, classic = FALSE)

set.doc(object, classic = FALSE)

```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>listall</code>	A logical operator specifying whether the items in the database should be listed.
<code>classic</code>	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.
<code>to.cat.vec</code>	A vector of strings specifying the names of the categorical variables that should be transformed to continuous.
<code>to.cont.vec</code>	A vector of strings specifying the names of the continuous variables that should be transformed to categorical.
<code>change.type.vec</code>	A vector of strings specifying the names of the variables that should be transformed to/from continuous/categorical.
<code>...</code>	arguments passed to other functions.
<code>value</code>	This is the value that will be replaced in the <code>xpose</code> data object <code>object</code> . <code>value</code> is used in the “replacement function” version of these functions. That is the form where we have <code>function.name(object) <- value</code> . If <code>value</code> is <code>NULL</code> then the functions prompt the user for a value. For <code>change.cat.levels</code> , <code>value</code> is the categorical limit <code>cat.limit</code> . For <code>change.dv.cat.levels</code> , <code>value</code> is the DV categorical limit <code>dv.cat.limit</code> . For <code>change.cat.cont</code> , <code>value</code> is the <code>change.type.vec</code> . See the examples below.
<code>cat.limit</code>	The limit for which we treat a list of values as categorical. If there are <code>cat.limit</code> or less unique values then the list is treated as categorical.
<code>dv.cat.limit</code>	The limit for which we treat DV as categorical. If there are <code>dv.cat.limit</code> or less unique <code>dv</code> values then <code>dv</code> is treated as categorical.

Value

An `xpose.data` object, except `get.doc`, which returns the value of `object@Doc`.

Functions

- `change.cat.cont()`: allows interchange between categorical and continuous data formats within the Xpose database. This in turn affects how plots are drawn.
- `change.cat.cont(object, listall = TRUE, classic = FALSE, to.cat.vec = NULL, to.cont.vec = NULL, ...) <- value`: allows interchange between categorical and continuous data formats within the Xpose database. This in turn affects how plots are drawn.
- `change.cat.levels()`: change settings for the number of unique data values required in a variable in order to define it as continuous for ordinary variables.
- `change.cat.levels(object, classic = FALSE, ...) <- value`: change settings for the number of unique data values required in a variable in order to define it as continuous for ordinary variables.
- `change.dv.cat.levels()`: change settings for the number of unique data values required in a variable in order to define it as continuous for the dependent variable.
- `change.dv.cat.levels(object, classic = FALSE, ...) <- value`: change settings for the number of unique data values required in a variable in order to define it as continuous for the dependent variable.
- `change.miss()`: change the value to use as 'missing'.

- `change.subset()`: is used for setting the data item's subset field. To specify a subset of the data to process, you use the variable names and the regular R selection operators. To combine a subset over two or more variables, the selection expressions for the two variables are combined using R's unary logical operators.

The variable names are those that are specified in the NONMEM table files (e.g. PRED, TIME, SEX).

The selection operators are: `==` (equal) `!=` (not equal) `||` (or) `>` (greater than) `<` (less than)

For example, to specify that TIME less than 24 should be processed, you type the expression: `TIME < 24`.

The unary logical operators are: `&` (and) `|` (or)

For example, to specify TIME less than 24 and males (SEX equal to 1), you type the expression: `TIME < 24 & SEX == 1`

This subset selection scheme works on all variables, including ID numbers.

The subset selection is not entirely stable. For example, there is no check that the user enters a valid expression, nor that the user specifies existing variable names. An erroneous expression will not become evident until a plot is attempted and the expression takes effect.

- `get.doc()`: get the documentation field in the Xpose data object.
- `set.doc()`: set the documentation field in the Xpose data object.

Author(s)

Andrew Hooker, Niclas Jonsson & Justin Wilkins

See Also

[Data](#), [SData](#), [subset](#), [xpose.data](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Change default subset
xpdb5 <- change.subset(xpdb5)

## Set documentation field
xpdb5 <- set.doc(xpdb5)
## View it
view.doc(xpdb5)

## change the categorical limit for the dv variable
change.dv.cat.levels(xpdb5) <- 10

## change the categorical limit for non DV variables
change.cat.levels(xpdb5) <- 2
## or
xpdb5 <- change.cat.levels(xpdb5,cat.levels=2)

## change variables from categorical to continuous
xpdb5 <- change.cat.cont(xpdb5,to.cat.vec=c("AGE"),to.cont.vec=c("SEX"))
xpdb5 <- change.cat.cont(xpdb5,change.type.vec=c("AGE","SEX"))
change.cat.cont(xpdb5) <- c("AGE","SEX")

## End(Not run)
```

compute.cwres

Compute the Conditional Weighted Residuals

Description

This function computes the conditional weighted residuals (CWRES) from a NONMEM run. CWRES are an extension of the weighted residuals (WRES), but are calculated based on the first-order with conditional estimation (FOCE) method of linearizing a pharmacometric model (WRES are calculated based on the first-order (FO) method). The function requires a NONMEM table file and an extra output file that must be explicitly asked for when running NONMEM, see details below.

Usage

```

compute.cwres(
  run.number,
  tab.prefix = "cwtab",
  sim.suffix = "",
  est.tab.suffix = ".est",
  deriv.tab.suffix = ".deriv",
  old.file.convention = FALSE,
  id = "ALL",
  printToOutfile = TRUE,
  onlyNonZero = TRUE,
  ...
)

xpose.calculate.cwres(
  object,
  cwres.table.prefix = "cwtab",
  tab.suffix = "",
  sim.suffix = "sim",
  est.tab.suffix = ".est",
  deriv.tab.suffix = ".deriv",
  old.file.convention = FALSE,
  id = "ALL",
  printToOutfile = TRUE,
  onlyNonZero = FALSE,
  classic = FALSE,
  ...
)

```

Arguments

<code>run.number</code>	The run number of the NONMEM from which the CWRES are to be calculated.
<code>tab.prefix</code>	The prefix to two NONMEM file containing the needed values for the computation of the CWRES, described in the details section.
<code>sim.suffix</code>	The suffix ,before the ".", of the NONMEM file containing the needed values for the computation of the CWRES, described in the details section. For example, the table files might be named <code>cwtab1sim.est</code> and <code>cwtab1sim.deriv</code> , in which case <code>sim.suffix="sim"</code> .
<code>est.tab.suffix</code>	The suffix, after the ".", of the NONMEM file containing the estimated parameter values needed for the CWRES calculation.
<code>deriv.tab.suffix</code>	The suffix, after the ".", of the NONMEM file containing the derivatives of the model with respect to the random parameters needed for the CWRES calculation.
<code>old.file.convention</code>	For backwards compatibility. Use this if you are using the previous file convention for CWRES (table files named <code>cwtab1</code> , <code>cwtab1.50</code> , <code>cwtab1.51</code> , ... , <code>cwtab.58</code> for example).

id	Can be either "ALL" or a number matching an ID label in the datasetname. Value is fixed to "ALL" for xpose.calculate.cwres.
printToOutfile	Logical (TRUE/FALSE) indicating whether the CWRES values calculated should be appended to a copy of the datasetname. Only works if id="ALL". If chosen the resulting output file will be datasetname.cwres. Value is fixed to TRUE for xpose.calculate.cwres.
onlyNonZero	Logical (TRUE/FALSE) indicating if the return value (the CWRES values) of compute.cwres should include the zero values associated with non-measurement lines in a NONMEM data file.
...	Other arguments passed to basic functions in code.
object	An xpose.data object.
cwres.table.prefix	The prefix to the NONMEM table file containing the derivative of the model with respect to the etas and epsilons, described in the details section.
tab.suffix	The suffix to the NONMEM table file containing the derivative of the model with respect to the etas and epsilons, described in the details section.
classic	Indicates if the function is to be used in the classic menu system.

Details

The function reads in the following two files:

```
paste(tab.prefix,run.number,sim.suffix,est.tab.suffix,sep="")
```

```
paste(tab.prefix,run.number,sim.suffix,deriv.tab.suffix,sep="")
```

Which might be for example:

```
cwtab1.est cwtab1.deriv
```

and (depending on the input values to the function) returns the CWRES in vector form as well as creating a new table file named:

```
paste(tab.prefix,run.number,sim.suffix,sep="")
```

Which might be for example:

```
cwtab1
```

Value

compute.cwres Returns a vector containing the values of the CWRES.

xpose.calculate.cwres Returns an Xpose data object that contains the CWRES. If simulated data is present, then the CWRES will also be calculated for that data.

Functions

- `xpose.calculate.cwres()`: This function is a wrapper around the function `compute.cwres`. It computes the CWRES for the model file associated with the Xpose data object input to the function. If possible it also computes the CWRES for any simulated data associated with the current Xpose data object. If you have problems with this function try using `compute.cwres` and then rereading your dataset into Xpose.

Setting up the NONMEM model file

In order for this function to calculate the CWRES, NONMEM must be run while requesting certain tables and files to be created. How these files are created differs depending on if you are using \$PRED or ADVAN as well as the version of NONMEM you are using. These procedures are known to work for NONMEM VI but may be different for NONMEM V and NONMEM VII. We have attempted to indicate where NONMEM V may be different, but this has not been extensively tested! For NONMEM VII the CWRES are calculated internally so this function is rarely needed.

This procedure can be done automatically using Perl Speaks NONMEM (PsN) and we highly recommend using PsN for this purpose. After installing PsN just type 'execute [modelName]-cwres'. See <https://uupharmacometrics.github.io/PsN/> for more details.

There are five main insertions needed in your NONMEM control file:

1. \$ABB COMRES=X.

Insert this line directly after your \$DATA line. The value of X is the number of ETA() terms plus the number of EPS() terms in your model. For example for a model with three ETA() terms and two EPS() terms the code would look like this:

```
$DATA temp.csv IGNORE=@
$ABB COMRES=5
$INPUT ID TIME DV MDV AMT EVID
$SUB ADVAN2 TRANS2
```

2. Verbatim code.

- Using ADVAN.

If you are using ADVAN routines in your model, then Verbatim code should be inserted directly after the \$ERROR section of your model file. The length of the code depends again on the number of ETA() terms and EPS() terms in your model. For each ETA(y) in your model there is a corresponding term G(y,1) that you must assign to a COM() variable. For each EPS(y) in your model, there is a corresponding HH(y,1) term that you must assign to a COM() variable.

For example for a model using ADVAN routines with three ETA() terms and two EPS() terms the code would look like this:

```
"LAST
" COM(1)=G(1,1)
" COM(2)=G(2,1)
" COM(3)=G(3,1)
" COM(4)=HH(1,1)
" COM(5)=HH(2,1)
```

- Using PRED.

If you are using \$PRED, the verbatim code should be inserted directly after the \$PRED section of your model file. For each ETA(y) in your model there is a corresponding term G(y,1) that you must assign to a COM() variable. For each EPS(y) in your model, there is a corresponding H(y,1) term that you must assign to a COM() variable. The code would look like this for three ETA() terms and two EPS() terms:

```
"LAST
" COM(1)=G(1,1)
" COM(2)=G(2,1)
```

```

" COM(3)=G(3,1)
" COM(4)=H(1,1)
" COM(5)=H(2,1)

```

3. INFN routine.

- Using ADVAN with NONMEM VI and higher.

If you are using ADVAN routines in your model, then an \$INFN section should be placed directly after the \$PK section using the following code. In this example we are assuming that the model file is named something like 'run1.mod', thus the prefix to these file names 'cwtab' has the same run number attached to it (i.e. 'cwtab1'). This should be changed for each new run number.

```

$INFN
IF (ICALL.EQ.3) THEN
  OPEN(50,FILE='cwtab1.est')
  WRITE(50,*) 'ETAS'
  DO WHILE(DATA)
    IF (NEWIND.LE.1) WRITE (50,*) ETA
  ENDDO
  WRITE(50,*) 'THETAS'
  WRITE(50,*) THETA
  WRITE(50,*) 'OMEGAS'
  WRITE(50,*) OMEGA(BLOCK)
  WRITE(50,*) 'SIGMAS'
  WRITE(50,*) SIGMA(BLOCK)
ENDIF

```

- Using ADVAN with NONMEM V.

If you are using ADVAN routines in your model, then you need to use an INFN subroutine. If we call the INFN subroutine 'myinfn.for' then the \$SUBS line of your model file should include the INFN option. That is, if we are using ADVAN2 and TRANS2 in our model file then the \$SUBS line would look like:

```
$SUB ADVAN2 TRANS2 INFN=myinfn.for
```

The 'myinfn.for' routine for 4 thetas, 3 etas and 1 epsilon is shown below. If your model has different numbers of thetas, etas and epsilons then the values of NTH, NETA, and NEPS, should be changed respectively. These vales are found in the DATA statement of the subroutine. additionally, in this example we are assuming that the model file is named something like 'run1.mod', thus the prefix to the output file names ('cwtab') in this subroutine has the same run number attached to it (i.e. 'cwtab1'). This number should be changed for each new run number (see the line beginning with 'OPEN').

```

SUBROUTINE INFN(ICALL, THETA, DATREC, INDXS, NEWIND)
DIMENSION THETA(*), DATREC(*), INDXS(*)
DOUBLE PRECISION THETA
COMMON /ROCM6/ THETAF(40), OMEGAF(30,30), SIGMAF(30,30)
COMMON /ROCM7/ SETH(40), SEOM(30,30), SESIG(30,30)
COMMON /ROCM8/ OBJECT
COMMON /ROCM9/ IERE, IERC
DOUBLE PRECISION THETAF, OMEGAF, SIGMAF
DOUBLE PRECISION OBJECT
REAL SETH, SEOM, SESIG

```

```

      DOUBLE PRECISION ETA(10)
      INTEGER J,I
      INTEGER IERE,IERC
      INTEGER MODE
      INTEGER NTH,NETA,NEPS
      DATA NTH,NETA,NEPS/4,3,1/
      IF (ICALL.EQ.0) THEN
C      open files here, if necessary
      OPEN(50,FILE='cwtab1.est')
      ENDIF
      IF (ICALL.EQ.3) THEN
      MODE=0
      CALL PASS(MODE)
      MODE=1
      WRITE(50,*) 'ETAS'
20     CALL PASS(MODE)
      IF (MODE.EQ.0) GO TO 30
      IF (NEWIND.NE.2) THEN
      CALL GETETA(ETA)
      WRITE (50,97) (ETA(I),I=1,NETA)
      ENDIF
      GO TO 20
30     CONTINUE
      WRITE (50,*) 'THETAS'
      WRITE (50,99) (THETAF(J),J=1,NTH)
      WRITE(50,*) 'OMEGAS'
      DO 7000 I=1,NETA
7000    WRITE (50,99) (OMEGAF(I,J),J=1,NETA)
      WRITE(50,*) 'SIGMAS'
      DO 7999 I=1,NEPS
7999    WRITE (50,99) (SIGMAF(I,J),J=1,NEPS)
      ENDIF
99     FORMAT (20E15.7)
98     FORMAT (2I8)
97     FORMAT (10E15.7)
      RETURN
      END

```

- Using \$PRED with NONMEM VI and higher.

If you are using \$PRED, then the following code should be placed at the end of the \$PRED section of the model file (together with the verbatim code). In this example we are assuming that the model file is named something like 'run1.mod', thus the prefix to these file names 'cwtab' has the same run number attached to it (i.e. 'cwtab1'). This should be changed for each new run number.

```

IF (ICALL.EQ.3) THEN
  OPEN(50,FILE='cwtab1.est')
  WRITE(50,*) 'ETAS'
  DO WHILE(DATA)
    IF (NEWIND.LE.1) WRITE (50,*) ETA

```

```

ENDDO
WRITE(50,*) 'THETAS'
WRITE(50,*) THETA
WRITE(50,*) 'OMEGAS'
WRITE(50,*) OMEGA(BLOCK)
WRITE(50,*) 'SIGMAS'
WRITE(50,*) SIGMA(BLOCK)
ENDIF

```

- Using \$PRED with NONMEM V.

If you are using \$PRED with NONMEM V, then you need to add verbatim code immediately after the \$PRED command. In this example we assume 4 thetas, 3 etas and 1 epsilon. If your model has different numbers of thetas, etas and epsilons then the values of NTH, NETA, and NEPS, should be changed respectively. These vales are found in the DATA statement below.

```

$PRED
"FIRST
"   COMMON /ROCM6/ THETAF(40), OMEGAF(30,30), SIGMAF(30,30)
"   COMMON /ROCM7/ SETH(40), SEOM(30,30), SESIG(30,30)
"   COMMON /ROCM8/ OBJECT
"   DOUBLE PRECISION THETAF, OMEGAF, SIGMAF
"   DOUBLE PRECISION OBJECT
"   REAL SETH, SEOM, SESIG
"   INTEGER J,I
"   INTEGER MODE
"   INTEGER NTH,NETA,NEPS
"   DATA NTH,NETA,NEPS/4,3,1/

```

After this verbatim code you add all of the abbreviated code needed for the \$PRED routine in your model file. After the abbreviated code more verbatim code is needed. This verbatim code should be added before the verbatim code discussed above under point 2. In the example below we are assuming that the model file is named something like 'run1.mod', thus the prefix to the output file names ('cwtab') has the same run number attached to it (i.e. 'cwtab1'). This number should be changed for each new run number (see the line beginning with 'OPEN').

```

"   IF (ICALL.EQ.0) THEN
"C   open files here, if necessary
"   OPEN(50,FILE='cwtab1.est')
"   ENDIF
"   IF (ICALL.EQ.3) THEN
"   MODE=0
"   CALL PASS(MODE)
"   MODE=1
"   WRITE(50,*) 'ETAS'
"20  CALL PASS(MODE)
"   IF (MODE.EQ.0) GO TO 30
"   IF (NEWIND.NE.2) THEN
"   CALL GETETA(ETA)
"   WRITE (50,97) (ETA(I),I=1,NETA)
"   ENDIF

```

```

"      GO TO 20
"30    CONTINUE
"      WRITE (50,*) 'THETAS'
"      WRITE (50,99) (THETA(J),J=1,NTH)
"      WRITE (50,*) 'OMEGAS'
"      DO 7000 I=1,NETA
"7000  WRITE (50,99) (OMEGA(I,J),J=1,NETA)
"      WRITE (50,*) 'SIGMAS'
"      DO 7999 I=1,NEPS
"7999  WRITE (50,99) (SIGMA(I,J),J=1,NEPS)
"      ENDIF
"99    FORMAT (20E15.7)
"98    FORMAT (2I8)
"97    FORMAT (10E15.7)

```

4. cwtab*.deriv table file.

A special table file needs to be created to print out the values contained in the COMRES variables. In addition the ID, IPRED, MDV, DV, PRED and RES data items are needed for the computation of the CWRES. The following code should be added to the NONMEM model file. In this example we continue to assume that we are using a model with three ETA() terms and two EPS() terms, extra terms should be added for new ETA() and EPS() terms in the model file. We also assume the model file is named something like 'run1.mod', thus the prefix to these file names 'cwtab' has the same run number attached to it (i.e. 'cwtab1'). This should be changed for each new run number.

```

$TABLE ID COM(1)=G11 COM(2)=G21 COM(3)=G31 COM(4)=H11 COM(5)=H21
        IPRED MDV NOPRINT ONEHEADER FILE=cwtab1.deriv

```

5. \$ESTIMATION.

To compute the CWRES, the NONMEM model file must use (at least) the FO method with the POSTHOC step. If the FO method is used and the POSTHOC step is not included then the CWRES values will be equivalent to the WRES. The CWRES calculations are based on the FOCE approximation, and consequently give an idea of the ability of the FOCE method to fit the model to the data. If you are using another method of parameter estimation (e.g. FOCE with interaction), the CWRES will not be calculated based on the same model linearization procedure.

Author(s)

Andrew Hooker

References

Hooker AC, Staatz CE, Karlsson MO. *Conditional weighted residuals, an improved model diagnostic for the FO/FOCE methods*. PAGE 15 (2006) Abstr 1001 [<http://www.page-meeting.org/?abstract=1001>].

Hooker AC, Staatz CE and Karlsson MO, Conditional weighted residuals (CWRES): a model diagnostic for the FOCE method, Pharm Res, 24(12): p. 2187-97, 2007, [[doi:10.1007/s110950079361-x](https://doi.org/10.1007/s110950079361-x)].

See Also

Other data functions: `add_transformed_columns`, `change_graphical_parameters`, `change_misc_parameters`, `data.checkout()`, `data_extract_or_assign`, `db.names()`, `export.graph.par()`, `export.variable.definitions()`, `import.graph.par()`, `import.variable.definitions()`, `make.sb.data()`, `nsim()`, `par_cov_summary`, `read.TTE.sim.data()`, `read.nm.tables()`, `read_NM_output`, `read_nm_table()`, `simprazExample()`, `tabulate.parameters()`, `xlabel()`, `xpose.data`, `xpose.print()`, `xpose4-package`, `xsubset()`

Examples

```
## Not run:
## Capture CWRES from cwtab5.est and cwtab5.deriv
cwres <- compute.cwres(5)
mean(cwres)
var(cwres)

## Capture CWRES from cwtab1.est and cwtab1.deriv, do not print out, allow zeroes
cwres <- compute.cwres("1", printToFile = FALSE,
  onlyNonZero = FALSE)

## Capture CWRES for ID==1
cwres.1 <- compute.cwres("1", id=1)

## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Compare WRES, CWRES
xpdb5 <- xpose.calculate.cwres(xpdb5)
cwres.wres.vs.idv(xpdb5)

## End(Not run)
```

 cov.splom

Plot scatterplot matrices of parameters, random parameters or covariates

Description

These functions plot scatterplot matrices of parameters, random parameters and covariates.

Usage

```
cov.splom(
  object,
  main = xpose.multiple.plot.title(object = object, plot.text =
    "Scatterplot matrix of covariates", ...),
```

```

varnames = NULL,
onlyfirst = TRUE,
smooth = TRUE,
lmline = NULL,
...
)

parm.splom(
  object,
  main = xpose.multiple.plot.title(object = object, plot.text =
    "Scatterplot matrix of parameters", ...),
  varnames = NULL,
  onlyfirst = TRUE,
  smooth = TRUE,
  lmline = NULL,
  ...
)

ranpar.splom(
  object,
  main = xpose.multiple.plot.title(object = object, plot.text =
    "Scatterplot matrix of random parameters", ...),
  varnames = NULL,
  onlyfirst = TRUE,
  smooth = TRUE,
  lmline = NULL,
  ...
)

```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>main</code>	A string giving the plot title or NULL if none.
<code>varnames</code>	A vector of strings containing labels for the variables in the scatterplot matrix.
<code>onlyfirst</code>	Logical value indicating if only the first row per individual is included in the plot.
<code>smooth</code>	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
<code>lmline</code>	logical variable specifying whether a linear regression line should be superimposed over an <code>xyplot</code> . NULL ~ FALSE. ($y \sim x$)
<code>...</code>	Other arguments passed to <code>xpose.plot.histogram</code> .

Details

The parameters or covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$params`, `object@Prefs@Xvardef$ranpar` or `object@Prefs@Xvardef$covariates`, are plotted together as scatterplot matrices.

A wide array of extra options controlling scatterplot matrices are available. See [xpose.plot.splom](#) for details.

To control the appearance of the labels and names in the scatterplot matrix plots you can try `varname.cex=0.5` and `axis.text.cex=0.5` (this changes the tick labels and the variable names to be half as large as normal).

Value

Delivers a scatterplot matrix.

Functions

- `cov.splom()`: A scatterplot matrix of covariates
- `parm.splom()`: A scatterplot matrix of parameters
- `ranpar.splom()`: A scatterplot matrix of random parameters

Author(s)

Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.splom](#), [xpose.panel.splom](#), [splom](#), [xpose.data-class](#), [xpose.prefs-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## A scatterplot matrix of parameters, grouped by sex
parm.splom(xpdb, groups="SEX")

## A scatterplot matrix of ETAs, grouped by sex
ranpar.splom(xpdb, groups="SEX")
```

```
## Covariate scatterplots, with text customization
cov.splom(xpdb, varname.cex=0.4, axis.text.cex=0.4, smooth=NULL, cex=0.4)
```

```
create.xpose.plot.classes
```

Create xpose.multiple.plot class.

Description

Creates a class for viewing and plotting xpose plots with multiple plots on the same page or multiple pages.

Usage

```
create.xpose.plot.classes()
```

Author(s)

Niclas Jonsson and Andrew C. Hooker

```
createXposeClasses    This function creates the Xpose data classes ("xpose.data" and
                       "xpose.prefs")
```

Description

This function defines and sets the Xpose data classes.

Usage

```
createXposeClasses(nm7 = F)
```

Arguments

nm7 FALSE if not using NONMEM 7.

Note

All the default settings are defined in this function.

Author(s)

Niclas Jonsson and Andrew C. Hooker

See Also

[xpose.data-class](#), [xpose.prefs-class](#)

cwres.dist.hist *Histogram of conditional weighted residuals (CWRES), for Xpose 4*

Description

This is a histogram of the distribution of conditional weighted residuals (CWRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.histogram` function.

Usage

```
cwres.dist.hist(object, ...)
```

Arguments

`object` An `xpose.data` object.
`...` Other arguments passed to `xpose.plot.histogram`.

Details

Displays a histogram of the conditional weighted residuals (CWRES).

Value

Returns a histogram of conditional weighted residuals (CWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#),

```
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),  
xpose4-package
```

Examples

```
## Here we load the example xpose database  
xpdb <- simpraz.xpdb  
  
## A vanilla plot  
cwres.dist.hist(xpdb)
```

cwres.dist.qq	<i>Quantile-quantile plot of conditional weighted residuals (CWRES), for Xpose 4</i>
---------------	--

Description

This is a QQ plot of the distribution of conditional weighted residuals (CWRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.qq` function.

Usage

```
cwres.dist.qq(object, ...)
```

Arguments

object	An xpose.data object.
...	Other arguments passed to <code>link{xpose.plot.qq}</code> .

Details

Displays a QQ plot of the conditional weighted residuals (CWRES).

Value

Returns a QQ plot of conditional weighted residuals (CWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.qq](#), [xpose.panel.qq](#), [qqmath](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
cwres.dist.qq(simpraz.xpdb)
```

`cwres.vs.cov`

Conditional Weighted residuals (CWRES) plotted against covariates, for Xpose 4

Description

This creates a stack of plots of conditional weighted residuals (CWRES) plotted against covariates, and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` and `xpose.plot.histogram` functions. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
cwres.vs.cov(
  object,
  ylb = "CWRES",
  smooth = TRUE,
  type = "p",
  main = "Default",
  ...
)
```

Arguments

object	An xpose.data object.
ylb	A string giving the label for the y-axis. NULL if none.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c" (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to <code>link{xpose.plot.default}</code> or <code>link{xpose.plot.histogram}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See `compute.cwres` for details.

A wide array of extra options controlling xyplots and histograms are available. See `xpose.plot.default` and `xpose.plot.histogram` for details.

Value

Returns a stack of xyplots and histograms of CWRES versus covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.plot.histogram`, `xyplot`, `histogram`, `xpose.prefs-class`, `compute.cwres`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp.addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`,

[kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

cwres.vs.cov(xpdb)
```

cwres.vs.idv	<i>Population conditional weighted residuals (CWRES) plotted against the independent variable (IDV) for Xpose 4</i>
--------------	---

Description

This is a plot of population conditional weighted residuals (CWRES) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
cwres.vs.idv(object, abline = c(0, 0), smooth = TRUE, ...)
```

Arguments

object	An <code>xpose.data</code> object.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Conditional weighted residuals (CWRES) are plotted against the independent variable, as specified in `object@Prefs@Xvardef$idv`.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of CWRES vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [compute.cwres](#),
[xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#),
[absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#),
[absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#),
[absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#),
[addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#),
[cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#),
[cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#),
[d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#),
[dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#),
[dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#),
[ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#),
[kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#),
[ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#),
[wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#),
[xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## A vanilla plot
cwres.vs.idv(xpdb)

## A conditioning plot
cwres.vs.idv(xpdb, by="HCTZ")
```

`cwres.vs.idv.bw`

Box-and-whisker plot of conditional weighted residuals vs the independent variable for Xpose 4

Description

This creates a box and whisker plot of conditional weighted residuals (CWRES) vs the independent variable (IDV), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
cwres.vs.idv.bw(object, ...)
```

Arguments

`object` An `xpose.data` object.

`...` Other arguments passed to `link{xpose.plot.bw}`.

Details

This creates a box and whisker plot of conditional weighted residuals (CWRES) vs the independent variable (IDV), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the [xpose.plot.bw](#) function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling bwplots are available. See [xpose.plot.bw](#) and [xpose.panel.bw](#) for details.

Value

Returns a stack of box-and-whisker plots of CWRES vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.bw](#), [xpose.panel.bw](#), [bwplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

cwres.vs.idv.bw(xpdb)
```

cwres.vs.pred	<i>Population conditional weighted residuals (CWRES) plotted against population predictions (PRED) for Xpose 4</i>
---------------	--

Description

This is a plot of population conditional weighted residuals (cwres) vs population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
cwres.vs.pred(object, abline = c(0, 0), smooth = TRUE, ...)
```

Arguments

object	An <code>xpose.data</code> object.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of CWRES vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xyplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

cwres.vs.pred(xpdb)

## A conditioning plot
cwres.vs.pred(xpdb, by="HCTZ")
```

cwres.vs.pred.bw	<i>Box-and-whisker plot of conditional weighted residuals vs population predictions for Xpose 4</i>
------------------	---

Description

This creates a box and whisker plot of conditional weighted residuals (CWRES) vs population predictions (PRED), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
cwres.vs.pred.bw(object, ...)
```

Arguments

object	An <code>xpose.data</code> object.
...	Other arguments passed to <code>link{xpose.plot.bw}</code> .

Details

This creates a box and whisker plot of conditional weighted residuals (CWRES) vs population predictions (PRED), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Conditional weighted residuals (CWRES) require some extra steps to calculate. See [compute.cwres](#) for details.

A wide array of extra options controlling bwplots are available. See [xpose.plot.bw](#) and [xpose.panel.bw](#) for details.

Value

Returns a box-and-whisker plot of CWRES vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.bw](#), [xpose.panel.bw](#), [bwplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

cwres.vs.pred.bw(xpdb)
```

cwres.wres.vs.idv	<i>Weighted residuals (WRES) and conditional WRES (CWRES) plotted against the independent variable (IDV)</i>
-------------------	--

Description

A graphical comparison between the WRES and CWRES as plotted against the independent variable. Conditional weighted residuals (CWRES) require some extra steps to calculate. Either add CWRES to your NONMEM table files or compute them using the information provided in [compute.cwres](#). A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Usage

```

cwres.wres.vs.idv(
  object,
  ylb = "Residuals",
  abline = c(0, 0),
  smooth = TRUE,
  scales = list(),
  ...
)

```

Arguments

object	An xpose.data object.
ylb	A string giving the label for the y-axis. NULL if none.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
scales	scales is passed to xpose.plot.default .
...	Other arguments passed to xpose.plot.default .

Value

A compound xyplot.

Author(s)

Niclas Jonsson & Andrew Hooker

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#), [compute.cwres](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.idv\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
cwres.wres.vs.idv(simpraz.xpdb)
```

<code>cwres.wres.vs.pred</code>	<i>Weighted residuals (WRES) and conditional WRES (CWRES) plotted against the population predictions (PRED)</i>
---------------------------------	---

Description

Graphically compares WRES and CWRES as plotted against the population predictions. Conditional weighted residuals (CWRES) require some extra steps to calculate. Either add CWRES to your NONMEM table files or compute them using the information provided in [compute.cwres](#). A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Usage

```

cwres.wres.vs.pred(
  object,
  ylb = "Residuals",
  abline = c(0, 0),
  smooth = TRUE,
  scales = list(),
  ...
)

```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>abline</code>	Vector of arguments to the <code>panel.abline</code> function. No abline is drawn if NULL.
<code>smooth</code>	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
<code>scales</code>	<code>scales</code> is passed to <code>xpose.plot.default</code>
<code>...</code>	Other arguments passed to <code>xpose.plot.default</code> .

Value

A compound xyplot.

Author(s)

Niclas Jonsson & Andrew Hooker

See Also

`xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `xpose.data-class`, `compute.cwres`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `dOFV.vs.cov()`, `dOFV.vs.id()`, `dOFV1.vs.dOFV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
cwres.wres.vs.pred(simpraz.xpdb)
```

data.checkout	<i>Check through the source dataset to detect problems</i>
---------------	--

Description

This function graphically "checks out" the dataset to identify errors or inconsistencies.

Usage

```
data.checkout(
  obj = NULL,
  datafile = ".ask.",
  hlin = -99,
  dotcol = "black",
  dotpch = 16,
  dotcex = 1,
  idlab = "ID",
  csv = NULL,
  main = "Default",
  ...
)
```

Arguments

obj	NULL or an xpose.data object.
datafile	A data file, suitable for import by read.table .
hlin	An integer, specifying the line number on which the column headers appear.
dotcol	Colour for the dots in the dotplot. If obj is an xpose data object then the default is to use the same value as defined for box-and-whisker plots.
dotpch	Plotting character for the dots in the dotplot. If obj is an xpose data object then the default is to use the same value as defined for box-and-whisker plots.
dotcex	Relative scaling for the dots in the dotplot. If obj is an xpose data object then the default is to use the same value as defined for box-and-whisker plots.
idlab	The ID column label in the dataset. Input as a text string.
csv	Is the data file in CSV format (comma separated values)? If the value is NULL then the user is asked at the command line. If supplied to the function the value can be TRUE/FALSE.
main	The title to the plot. "default" means that Xpose creates a title.
...	Other arguments passed to <code>link[lattice]{dotplot}</code> .

Details

This function creates a series of dotplots, one for each variable in the dataset, against individual ID. Outliers and clusters may easily be detected in this manner.

Value

A stack of dotplots.

Author(s)

Niclas Jonsson, Andrew Hooker & Justin Wilkins

See Also

[dotplot](#), [xpose.prefs-class](#), [read.table](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run, table and data files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

data.checkout(xpdb5, datafile = "mydata.dta")
data.checkout(datafile = "mydata.dta")

## End(Not run)
```

```
data_extract_or_assign
```

Extract or assign data from an xpose.data object.

Description

Extracts or assigns the data from the Data or SData slots in an "xpose.data" object.

Usage

```
Data(object, inclZeroWRES = FALSE, onlyfirst = FALSE, subset = NULL)
```

```
Data(object, quiet = TRUE, keep.structure = F) <- value
```

```
SData(
  object,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  subset = NULL,
  samp = NULL
)
```

```
SData(object) <- value
```

Arguments

object	An "xpose.data" object
inclZeroWRES	Logical value indicating whether rows with WRES==0 should be included in the extracted data.
onlyfirst	Logical value indicating whether only the first line per individual should be included in the extracted data.
subset	Expression with which the extracted data should be subset (see xsubset)
quiet	TRUE or FALSE if FALSE then some more information is printed out when adding data to an Xpose object.
keep.structure	TRUE or FALSE iff FALSE then values are converted to continuous or categorical according to the rules set up by xpose using object@Prefs@Cat.levels, object@Prefs@DV.cat.levels and the values in the "catab" file.
value	An R data.frame.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.

Details

When using `Data` to assign a `data.frame` to the `Data` slot in the "xpose.data" object a number of things happen:

Each column in the `data.frame` is checked and set to factor if the number of unique values are less than the value of `Cat.levels` (see [xpose.prefs-class](#)).

It is checked which of the predefined xpose data variables that exists in the `data.frame`. The variable definitions that does not exist are set to `NULL`.

The column identified by the `dv` xpose variable definition, is checked and set to factor if the number of unique values are less than or equal to the `DV.Cat.levels` (see [xpose.prefs-class](#)).

Finally, each column name in the `data.frame` is checked for a label (see [xpose.prefs-class](#)). If it is non-existent, the label is set to the column name.

When `SData` is used to assign a `data.frame` to the `SData` slot it is first checked that the number of rows in the `SData` `data.frame` is an even multiple of the number of rown in `Data`. Next, each column in the `SData` `data.frame` is assigned the same class as the corresponding column in the `Data` `data.frame` (it is required that the columns are the same in `Data` and `SData`). Finally, an extra column, "iter", is added to `SData`, which indicates the iteration number that each row belongs to. At the same time, the `Nsim` slot of the "xpose.data" object is set to the number of iterations (see [nsim](#)).

Value

Returns a `data.frame` from the `Data` or `SData` slots, excluding rows as indicated by the arguments.

Functions

- `Data()`: Extract data
- `Data(object, quiet = TRUE, keep.structure = F) <- value`: assign data
- `SData()`: extract simulated data
- `SData(object) <- value`: assign simulated data

Author(s)

Niclas Jonsson

See Also

[xpose.data-class](#), [xpose.prefs-class](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
xpdb <- simpraz.xpdb

## Extract data
my.dataframe <- Data(xpdb)

## Assign data
Data(xpdb) <- my.dataframe

## Extract simulated data
my.simulated.dataframe <- SData(xpdb)

## Assign simulated data
SData(xpdb) <- my.simulated.dataframe
```

db.names	<i>Prints the contents of an Xpose data object</i>
----------	--

Description

These functions print a summary of the specified Xpose object to the R console.

Usage

```
db.names(object)
```

Arguments

object An xpose.data object.

Details

These functions return a detailed summary of the contents of a specified [xpose.data](#) object.

Value

A detailed summary of the contents of a specified [xpose.data](#) object.

Author(s)

Niclas Jonsson & Justin Wilkins

See Also[xpose.data](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [export.graph.par\(\)](#), [export.variable.definition](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
db.names(simpraz.xpdb)
```

dOFV.vs.cov

Change in individual objective function value vs. covariate value.

Description

Change in individual objective function value vs. covariate value.

Usage

```
dOFV.vs.cov(
  xpdb1,
  xpdb2,
  covariates = xvardef("covariates", xpdb1),
  ylb = expression(paste(Delta, OFV[i])),
  main = "Default",
  smooth = TRUE,
  abline = c(0, 0),
  ablcol = "grey",
  abllwd = 2,
  abllty = "dashed",
  max.plots.per.page = 1,
  ...
)
```

Arguments

xpdb1	Xpose data object for first NONMEM run
xpdb2	Xpose data object for second NONMEM run
covariates	Covariates to plot against
ylb	Label for Y axis.
main	Title of plot.
smooth	Should we have a smooth?

abline	abline description.
ablcol	color of abline
ablwd	line width of abline
abllty	type of abline
max.plots.per.page	Plots per page.
...	additional arguments to function

Author(s)

Andrew C. Hooker

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## read in table files
xpdb8 <- xpose.data(8)
xpdb11 <- xpose.data(11)

## Make some plots
dOFV.vs.cov(xpdb8, xpdb11, "AGE")
dOFV.vs.cov(xpdb8, xpdb11, c("AGE", "SECR"))

## End(Not run)
```

dOFV.vs.id

*Change in Objective function value vs. removal of individuals.***Description**

A plot showing the most and least influential individuals in determining a drop in OFV between two models.

Usage

```
dOFV.vs.id(
  xpdb1,
  xpdb2,
  sig.drop = -3.84,
  decrease.label.number = 3,
  increase.label.number = 3,
  id.lab.cex = 0.6,
  id.lab.pos = 2,
  type = "o",
  xlb = "Number of subjects removed",
  ylb = expression(paste(Delta, "OFV")),
  main = "Default",
  sig.line.col = "red",
  sig.line.lty = "dotted",
  tot.line.col = "grey",
  tot.line.lty = "dashed",
  key = list(columns = 1, lines = list(pch = c(super.sym$pch[1:2], NA, NA), type =
    list("o", "o", "1", "1"), col = c(super.sym$col[1:2], sig.line.col, tot.line.col),
    lty = c(super.sym$lty[1:2], sig.line.lty, tot.line.lty)), text =
    list(c(expression(paste(Delta, OFV[i] < 0)), expression(paste(Delta, OFV[i] > 0)),
    expression(paste("Significant ", Delta, OFV)), expression(paste("Total ", Delta,
    OFV)))), corner = c(0.95, 0.5), border = T),
  ...
)
```

Arguments

xpdb1	Xpose data object for first NONMEM run ("new" run)
xpdb2	Xpose data object for Second NONMEM run ("reference" run)
sig.drop	What is a significant drop of OFV?
decrease.label.number	How many points should be labeled with ID values for those IDs with a drop in iOFV?
increase.label.number	How many points should be labeled with ID values for those IDs with an increase in iOFV?

id.lab.cex	Size of ID labels.
id.lab.pos	ID label position.
type	Type of lines.
xlb	X-axis label.
ylb	Y-axis label.
main	Title of plot.
sig.line.col	Significant OFV drop line color.
sig.line.lty	Significant OFV drop line type.
tot.line.col	Total OFV drop line color.
tot.line.lty	Total OFV drop line type.
key	Legend for plot.
...	Additional arguments to function.

Author(s)

Andrew C. Hooker

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_cov\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
library(xpose4)

## first make sure that the iofv values are read into xpose
cur.dir <- getwd()
setwd(paste(cur.dir, "/LAG_TIME", sep=""))
xpdb1 <- xpose.data(1)
setwd(paste(cur.dir, "/TRANSIT_MODEL", sep=""))
xpdb2 <- xpose.data(1)
setwd(cur.dir)
```

```
## then make the plot
dOFV.vs.id(xpdb1,xpdb2)

## End(Not run)
```

dOFV1.vs.dOFV2	<i>Change in individual objective function value 1 vs. individual objective function value 2.</i>
----------------	---

Description

Change in individual objective function value 1 vs. individual objective

Usage

```
dOFV1.vs.dOFV2(
  xpdb1,
  xpdb2,
  xpdb3,
  ylb = expression(paste(Delta, OFV1[i])),
  xlb = expression(paste(Delta, OFV2[i])),
  main = "Default",
  smooth = NULL,
  abline = c(0, 1),
  ablcol = "grey",
  abllwd = 2,
  abllty = "dashed",
  lmline = TRUE,
  ...
)
```

Arguments

xpdb1	Xpose data object for first NONMEM run
xpdb2	Xpose data object for second NONMEM run
xpdb3	Xpose data object for third NONMEM run
ylb	Label for Y axis.
xlb	Label for X axis.
main	Title of plot.
smooth	Should we have a smooth?
abline	abline description.
ablcol	color of abline
abllwd	line width of abline

abllty type of abline
 lmline Linear regression line?
 ... Additional arguments to function.

Author(s)

Andrew C. Hooker

See Also

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Not run:
## read in table files
xpdb8 <- xpose.data(8)
xpdb9 <- xpose.data(9)
xpdb11 <- xpose.data(11)

## Make the plot
d0FV.vs.cov(xpdb8,xpdb9,xpdb11)

## End(Not run)
```

dv.preds.vs.idv *Observations (DV), individual predictions (IPRED) and population predictions (IPRED) plotted against the independent variable (IDV), for Xpose 4*

Description

This is a compound plot consisting of plots of observations (DV), individual predictions (IPRED), and population predictions (PRED) against the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function.

Usage

```
dv.preds.vs.idv(
  object,
  ylb = "Observations/Predictions",
  layout = c(3, 1),
  smooth = TRUE,
  scales = list(),
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>layout</code>	A list controlling the number of columns and rows in a compound plot. The default is 2 columns and 1 row.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>scales</code>	A list to be used for the <code>scales</code> argument in <code>xyplot</code> .
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling `xyplots` are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a compound plot comprising plots of observations (DV), individual predictions (IPRED), and population predictions (PRED) against the independent variable (IDV).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[dv.vs.idv](#), [ipred.vs.idv](#), [pred.vs.idv](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#),

```

absval.iwres.vs.ipred(), absval.iwres.vs.ipred.by.cov(), absval.iwres.vs.pred(), absval.wres.vs.cov.bw(),
absval.wres.vs.idv(), absval.wres.vs.pred(), absval.wres.vs.pred.by.cov(), absval_delta_vs_cov_model_co
addit.gof(), autocorr.cwres(), autocorr.iwres(), autocorr.wres(), basic.gof(), basic.model.comp(),
cat.dv.vs.idv.sb(), cat.pc(), cov.splom(), cwres.dist.hist(), cwres.dist.qq(), cwres.vs.cov(),
cwres.vs.idv(), cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(),
cwres.wres.vs.pred(), d0FV.vs.cov(), d0FV.vs.id(), d0FV1.vs.d0FV2(), data.checkout(),
dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(), dv.vs.pred(),
dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(),
ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(), iwres.dist.qq(), iwres.vs.idv(),
kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Here we load the example xpose database
xpdb <- simpraz.xpdb

dv.preds.vs.idv(xpdb)

```

dv.vs.idv	<i>Observations (DV) plotted against the independent variable (IDV) for Xpose 4</i>
-----------	---

Description

This is a plot of observations (DV) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.idv(object, smooth = TRUE, ...)
```

Arguments

object	An <code>xpose.data</code> object.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling `xyplot` are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns an xyplot of DV vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

dv.vs.idv(xpdb)

## A conditioning plot
dv.vs.idv(xpdb, by="HCTZ")

## Logarithmic Y-axis
dv.vs.idv(xpdb, logy=TRUE)
```

dv.vs.ipred

Observations (DV) plotted against individual predictions (IPRED) for Xpose 4

Description

This is a plot of observations (DV) vs individual predictions (IPRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.ipred(object, abline = c(0, 1), smooth = TRUE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>abline</code>	Vector of arguments to the <code>panel.abline</code> function. No abline is drawn if NULL.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling `xyplot` are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns an `xyplot` of DV vs IPRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

dv.vs.ipred(xpdb)

## A conditioning plot
dv.vs.ipred(xpdb, by="HCTZ")
```

dv.vs.ipred.by.cov *Dependent variable vs individual predictions, conditioned on covariates, for Xpose 4*

Description

This is a plot of dependent variable (DV) vs individual predictions (IPRED) conditioned by covariates, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.ipred.by.cov(
  object,
  covs = "Default",
  abline = c(0, 1),
  smooth = TRUE,
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>covs</code>	A vector of covariates to use in the plot. If "Default" the the covariates defined in <code>object@Prefs@Xvardef\$Covariates</code> are used.
<code>abline</code>	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

A wide array of extra options controlling `xyplot` are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns a stack of `xyplots` of DV vs IPRED, conditioned on covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`dv.vs.ipred`, `xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
dv.vs.ipred.by.cov(simpraz.xpdb, covs=c("HCTZ", "WT"), max.plots.per.page=2)
```

<code>dv.vs.ipred.by.idv</code>	<i>Dependent variable vs individual predictions, conditioned on independent variable, for Xpose 4</i>
---------------------------------	---

Description

This is a plot of the dependent variable (DV) vs individual predictions (IPRED) conditioned by the independent variable, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.ipred.by.idv(object, abline = c(0, 1), smooth = TRUE, ...)
```

Arguments

object	An xpose.data object.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplot are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a stack of xyplots of DV vs IPRED, conditioned on the independent variable.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[dv.vs.ipred](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
dv.vs.ipred.by.idv(simpraz.xpdb)
```

dv.vs.pred	<i>Observations (DV) plotted against population predictions (PRED) for Xpose 4</i>
------------	--

Description

This is a plot of observations (DV) vs population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.pred(object, abline = c(0, 1), smooth = TRUE, ...)
```

Arguments

object	An <code>xpose.data</code> object.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of DV vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#),

```

cwres.wres.vs.pred(), d0FV.vs.cov(), d0FV.vs.id(), d0FV1.vs.d0FV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(),
ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(), iwres.dist.qq(), iwres.vs.idv(),
kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Here we load the example xpose database
xpdb <- simpraz.xpdb

## A vanilla plot
dv.vs.pred(xpdb)

## A conditioning plot
dv.vs.pred(xpdb, by="HCTZ")

```

dv.vs.pred.by.cov	<i>Dependent variable vs population predictions, conditioned on covariates, for Xpose 4</i>
-------------------	---

Description

This is a plot of the dependent variable (DV) vs population predictions (PRED) conditioned by covariates, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```

dv.vs.pred.by.cov(
  object,
  covs = "Default",
  abline = c(0, 1),
  smooth = TRUE,
  main = "Default",
  ...
)

```

Arguments

object	An <code>xpose.data</code> object.
covs	A vector of covariates to use in the plot. If "Default" the the covariates defined in <code>object@Prefs@Xvardef\$Covariates</code> are used.

abline	Vector of arguments to the <code>panel.abline</code> function. No abline is drawn if NULL.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$Covariates`, is evaluated in turn, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns a stack of xyplots of DV vs PRED, conditioned on covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`dv.vs.pred`, `xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
dv.vs.pred.by.cov(simpraz.xpdb, covs=c("HCTZ", "WT"), max.plots.per.page=2)
```

dv.vs.pred.by.idv	<i>Dependent variable vs population predictions, conditioned on independent variable, for Xpose 4</i>
-------------------	---

Description

This is a plot of the dependent variable (DV) vs population predictions (PRED) conditioned by the independent variable, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
dv.vs.pred.by.idv(object, abline = c(0, 1), smooth = TRUE, ...)
```

Arguments

object	An <code>xpose.data</code> object.
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a stack of xyplots of DV vs PRED, conditioned on the independent variable.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[dv.vs.pred](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#),

```

cwres.wres.vs.pred(), d0FV.vs.cov(), d0FV.vs.idv(), d0FV1.vs.d0FV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(),
ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(), iwres.dist.qq(), iwres.vs.idv(),
kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(),
ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(),
wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```
dv.vs.pred.by.idv(simpraz.xpdb)
```

dv.vs.pred.ipred	<i>Observations (DV) are plotted against individual predictions (IPRED) and population predictions (PRED), for Xpose 4</i>
------------------	--

Description

This is a compound plot consisting of plots of observations (DV) against individual predictions (IPRED) and population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function.

Usage

```

dv.vs.pred.ipred(
  object,
  xlb = "Predictions",
  layout = c(2, 1),
  abline = c(0, 1),
  lmline = TRUE,
  smooth = NULL,
  scales = list(),
  ...
)

```

Arguments

object	An <code>xpose.data</code> object.
xlb	A string giving the label for the x-axis. NULL if none.
layout	A list giving the layout of the graphs on the plot, in columns and rows.
abline	Vector of arguments to the <code>panel.abline</code> function. No abline is drawn if NULL.
lmline	logical variable specifying whether a linear regression line should be superimposed over an <code>xyplot</code> . NULL ~ FALSE. (y~x)
smooth	NULL or TRUE value indicating whether an x-y smooth should be superimposed.
scales	A list to be used for the <code>scales</code> argument in <code>xyplot</code> .
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Plots of DV vs PRED and IPRED are presented side by side for comparison.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a compound plot comprising plots of observations (DV) against individual predictions (IPRED) and population predictions (PRED).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[dv.vs.pred](#), [dv.vs.ipred](#), [xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp.addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
dv.vs.pred.ipred(simpraz.xpdb)
```

export.graph.par

Exports Xpose graphics settings to a file.

Description

This function exports graphics settings for a specified Xpose data object to a file.

Usage

```
export.graph.par(object)

xpose.write(object, file = "xpose.ini")
```

Arguments

object	An xpose.data object.
file	The file to contain exported Xpose settings.

Details

This function exports the graphics settings (contents of object@Prefs@Graph.prefs) for a given xpose.data object to a file, typically 'xpose.ini'. It is a wrapper for xpose.write. Note that the file format is not the same as is used in [import.variable.definitions](#) and [export.variable.definitions](#).

Value

Null.

Functions

- xpose.write(): export graphics settings for a specified Xpose data object to a file.

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

[import.graph.par](#), [xpose.prefs-class](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data.extract_or_assign](#), [db.names\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## For a filename prompt
export.graph.par(xpdb5)

## Command-line driven
xpose.write(xpdb5, "c:/XposeSettings/mytheme.ini")
```

```
## End(Not run)
```

```
export.variable.definitions
```

Exports Xpose variable definitions to a file from an Xpose data object.

Description

This function exports variable definitions for a specified Xpose data object to a file.

Usage

```
export.variable.definitions(object, file = "")
```

Arguments

object	An xpose.data object.
file	A file name as a string.

Details

This function exports variable definitions (contents of `object@Prefs@Xvardef`) for a given xpose.data object to a file, typically 'xpose.vardefs.ini'. Note that file format is not the same as used for graphics settings. It is a wrapper for the R function [dput](#).

Value

Null.

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

[import.variable.definitions](#), [xpose.prefs-class dput](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
od = setwd(tempdir()) # move to a temp directory
(cur.files <- dir()) # current files in temp directory

export.variable.definitions(simpraz.xpdb,file="xpose.vardefs.ini")
(new.files <- dir()[!(dir() %in% cur.files)]) # what files are new here?

file.remove(new.files) # remove this file
setwd(od) # restore working directory
```

GAM_summary_and_plot *GAM functions for Xpose 4*

Description

These are functions for summarizing and plotting the results of the generalized additive model within Xpose

Usage

```
xp.akaike.plot(
  gamobj = NULL,
  title = "Default",
  xlb = "Akaike value",
  ylb = "Models",
  ...
)
```

```
xp.cook(gam.object)
```

```
xp.ind.inf.fit(
  gamobj = NULL,
  plot.ids = TRUE,
  idscex = 0.7,
  ptscex = 0.7,
  title = "Default",
  recur = FALSE,
  xlb = NULL,
  ylb = NULL,
  ...
)
```

```
xp.ind.inf.terms(
  gamobj = NULL,
  xlb = NULL,
```

```

    ylb = NULL,
    plot.ids = TRUE,
    idsceex = 0.7,
    ptsceex = 0.7,
    prompt = TRUE,
    ...
)

xp.ind.stud.res(
  gamobj = NULL,
  title = "Default",
  recur = FALSE,
  xlb = NULL,
  ylb = NULL
)

xp.plot(
  gamobj = NULL,
  plot.ids = TRUE,
  idsceex = 0.7,
  ptsceex = 0.7,
  prompt = TRUE,
  ...
)

xp.summary(gamobj = NULL)

```

Arguments

gamobj	A GAM object to use in the plot. IF null then the user is asked to choose from a list of GAM objects in memory.
title	A text string indicating plot title. If NULL, left blank.
xlb	A text string indicating x-axis legend. If NULL, left blank.
ylb	A text string indicating y-axis legend. If NULL, left blank.
...	Other arguments passed to the GAM functions.
gam.object	A GAM object (see gam).
plot.ids	Logical, specifies whether or not ID numbers should be displayed.
idsceex	ID label size.
ptsceex	Point size.
recur	If dispersion should be used in the GAM object.
prompt	Specifies whether or not the user should be prompted to press RETURN between plot pages. Default is TRUE.
object	An xpose.data object.

Value

Plots or summaries.

Functions

- `xp.akaike.plot()`: An Akaike plot of the results.
- `xp.cook()`: Individual parameters to GAM fit.
- `xp.ind.inf.fit()`: Individual influence on GAM fit.
- `xp.ind.inf.terms()`: Individual influence on GAM terms.
- `xp.ind.stud.res()`: Studentized residuals.
- `xp.plot()`: GAM residuals of base model vs. covariates.
- `xp.summary()`: Summarize GAM.

Author(s)

Niclas Jonsson & Andrew Hooker

See Also

[gam](#), [dotplot](#)

Other GAM functions: [xp.get.disp\(\)](#), [xp.scope3\(\)](#), [xpose.bootgam\(\)](#), [xpose.gam\(\)](#), [xpose4-package](#)

gof

Structured goodness of fit diagnostics.

Description

This is a template function for creating structured goodness of fit diagnostics using the functions in the Xpose specific library.

Usage

```
gof(  
  runno = NULL,  
  save = FALSE,  
  onefile = FALSE,  
  saveType = "pdf",  
  pageWidth = 7.6,  
  pageHeight = 4.9,  
  structural = TRUE,  
  residual = TRUE,  
  covariate = FALSE,  
  iiv = FALSE,  
  iov = FALSE,  
  all = FALSE,  
  myTrace = xpPage  
)
```

Arguments

runno	The run number fo Xpose to identify the appropriate files to read. In addition runno is used to construct the file name to save plots in. runno can also be NULL for cases in which the function is used for non-Xpose based code.
save	Logical. TRUE if the plot(s) is to be saved in a file. FALSE if the plot(s) is to be displayed on screen. The plot(s) will be saved in a file named with the function name followed by the word 'run', the run number, an order number followed by a file name extension appropriate for the selected saveType. For example 'gofrun1-01.pdf' for the first plot file created by a script called gof based on output from run 1 and saveType='pdf'.
onefile	Logical. TRUE if plots are to be save in a single file and FALSE if each plot should be saved as a separate file. In the latter case, each file will be have an incremented order number (01-99).
saveType	The type of graphics file to produce if save=TRUE. Allowed values are 'pdf' (default), 'wmf' (only Windows) and 'png'.
pageWidth	The width of the graphics device in inches.
pageHeight	The height of the graphics device in inches.
structural	Logical. TRUE if the code in the structural model section (see below) should be executed and FALSE if not.
residual	Logical. TRUE if the code in the residual model section (see below) should be executed and FALSE if not.
covariate	Logical. TRUE if the code in the covariate model section (see below) should be executed and FALSE if not.
iiv	Logical. TRUE if the code in the IIV model section (see below) should be executed and FALSE if not.
ioy	Logical. TRUE if the code in the IOV model section (see below) should be executed and FALSE if not.
all	Logical. TRUE if the code in all sections (see below) should be executed.
myTrace	NULL or the name of a function. The value of myTrace can used with the lattice page= argument to annotate plots for traceability.

Details

The gof function is provided as a template to facilitate the (structured) use of the functions in the Xpose specific library. Xpose specific is extensively described in the 'Xpose Bestiary'.

The function can be renamed so that multiple scripts can be used in parallel.

The function is set up to make it easy to display plots on screen as well as to save them in files. In the latter case, plots are save in a sub-directory called 'Plots'.

The arguments structural, residual, covariate, iiv, ioy and all are just "switches" to different parts of the code (if-blocks). These blocks can be removed or the default values of the arguments changed to better suit the needs of the user.

It is also possible to add tracing information to the produced plots. This is done via the myTrace argument. A non-NULL value should be a function that returns a panel.text object. The default

is the `xpPage` function that will put a string concatenated from the device name, function name, working directory and date, in small, faint grey, font at the bottom of each graph page. Note that the user need to add `page=myTrace` as an argument to the `Xpose` functions for this to have an effect.

The function calls a support function called `gofSetup`, which is responsible for setting up the graphics device and determining the file names for saved graphs.

Value

Does not return anything unless the user specify a return value.

Author(s)

E. Niclas Jonsson, Mats Karlsson and Andrew Hooker

See Also

[xpose4-package](#)

Other generic functions: [xpose.multiple.plot](#), [xpose4-package](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## This is an example of how the function may be setup by a user.

library(xpose4)
mygof <- gof
fix(mygof)

myggof <- function (runno = NULL, save = FALSE, onefile = FALSE, saveType = "pdf",
  pageWidth = 7.6, pageHeight = 4.9, structural = TRUE, residual = TRUE,
  covariate = FALSE, iiv = FALSE, iov = FALSE, all = FALSE, myTrace=xpPage) {

  gofSetup(runno, save, onefile, saveType, pageWidth, pageHeight)
  xpdb <- xpose.data(runno)
```

```

    if (structural || all) {
      xplot <- dv.vs.pred.ipred(xpdb, page = myPage)
      print(xplot)
    }
    if (residual || all) {
      xplot <- absval.wres.vs.pred(xpdb, page = myPage)
      print(xplot)
    }
    if (covariate || all) {
    }
    if (iiv || all) {
    }
    if (ioy || all) {
    }
    if (save) dev.off()
invisible()
}

## The function can then be execute, e.g.:
mygof(1)

## End(Not run)

```

import.graph.par	<i>Imports Xpose graphics settings from a file to an Xpose data object.</i>
------------------	---

Description

This function imports graphics settings for a specified Xpose data object from a file.

Usage

```
import.graph.par(object, classic = FALSE)
```

Arguments

object	An xpose.data object.
classic	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

This function imports graphics settings (contents of object@Prefs@Graph.prefs) for a given xpose.data object from a file, typically 'xpose.ini'. It is a wrapper for xpose.read. It returns an xpose.data object. Note that the file format is not the same as is used in [import.variable.definitions](#) and [export.variable.definitions](#).

Value

An `xpose.data` object (classic = FALSE) or null (classic = TRUE).

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

`export.graph.par`, `xpose.prefs-class`

Other data functions: `add_transformed_columns`, `change_graphical_parameters`, `change_misc_parameters`, `compute.cwres()`, `data.checkout()`, `data_extract_or_assign`, `db.names()`, `export.graph.par()`, `export.variable.definitions()`, `import.variable.definitions()`, `make.sb.data()`, `nsim()`, `par_cov_summary`, `read.TTE.sim.data()`, `read.nm.tables()`, `read_NM_output`, `read_nm_table()`, `simprazExample()`, `tabulate.parameters()`, `xlabel()`, `xpose.data`, `xpose.print()`, `xpose4-package`, `xsubset()`

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Import graphics preferences you saved earlier using export.graph.par
xpdb5 <- import.graph.par(xpdb5)

## Command-line driven
xpdb5 <- xpose.read(xpdb5, "c:/XposeSettings/mytheme.ini")

## End(Not run)
```

```
import.variable.definitions
```

Imports Xpose variable definitions from a file to an Xpose data object.

Description

This function imports variable definitions for a specified Xpose data object from a file.

Usage

```
import.variable.definitions(object, classic = FALSE)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>classic</code>	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

This function imports variable definitions (contents of `object@Prefs@Xvardef`) for a given `xpose.data` object from a file, typically `'xpose.vardefs.ini'`. It returns an `xpose.data` object. Note that file format is not the same as used for graphics settings. It is a wrapper for the R function [dget](#).

Value

An `xpose.data` object (`classic == FALSE`) or null (`classic == TRUE`).

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

[export.variable.definitions](#), [xpose.prefs-class](#) [dget](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), `xpose.data`, [xpose.print\(\)](#), `xpose4-package`, [xsubset\(\)](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

xpdb5 <- import.variable.definitions(xpdb5)

## End(Not run)
```

ind.plots	<i>Observations (DV), individual predictions (IPRED) and population predictions (PRED) are plotted against the independent variable for every individual in the dataset, for Xpose 4</i>
-----------	--

Description

This is a compound plot consisting of plots of observations (DV), individual predictions (IPRED) and population predictions (PRED) against the independent variable for every individual in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function.

Usage

```
ind.plots(
  object,
  y.vals = c(xvardef("dv", new.obj), xvardef("ipred", new.obj), xvardef("pred", new.obj)),
  x.vals = xvardef("idv", new.obj),
  id.vals = xvardef("id", new.obj),
  key.text = y.vals,
  main = "Default",
  key = "Default",
  xlb = xlabel(xvardef("idv", object), object),
  ylb = NULL,
  layout = c(4, 4),
  inclZeroWRES = FALSE,
  subset = xsubset(object),
  type = "o",
  grid = FALSE,
  col = c(1, 2, 4),
  lty = c(0, 1, 3),
  lwd = c(1, 1, 1),
  pch = c(21, 32, 32),
  cex = c(0.7, 0.7, 0.7),
  fill = c("lightgrey", 0, 0),
  prompt = FALSE,
  mirror = NULL,
  main.cex = 0.9,
  max.plots.per.page = 1,
  pch.ip.sp = c(21, 19, 18),
  cex.ip.sp = c(0.7, 0.4, 0.4),
  y.vals.subset = NULL,
  ...
)
```

Arguments

`object` An `xpose.data` object.

y.vals	The Y values to use.
x.vals	The X values to use.
id.vals	The ID values to use.
key.text	The text in the legend to use.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
key	Create a legend.
xlb	A string giving the label for the x-axis. NULL if none.
y1b	A string giving the label for the y-axis. NULL if none.
layout	A list giving the layout of the graphs on the plot, in columns and rows. The default is 4x4.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot. The default is TRUE.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
type	1-character string giving the type of plot desired. The default is "o", for over-plotted points and lines. See xpose.plot.default .
grid	Should the plots have a grid in each plot?
col	A list of three elements, giving plotting characters for observations, individual predictions, and population predictions, in that order. The default is black for DV, red for individual predictions, and blue for population predictions.
lty	A list of three elements, giving line types for observations, individual predictions, and population predictions, in that order.
lwd	A list of three elements, giving line widths for observations, individual predictions, and population predictions, in that order.
pch	A list of three elements, giving plotting characters for observations, individual predictions, and population predictions, in that order.
cex	A list of three elements, giving relative point size for observations, individual predictions, and population predictions, in that order. The default is c(0.7,0.7,0.7).
fill	Fill the circles in the points?
prompt	Specifies whether or not the user should be prompted to press RETURN between plot pages. Default is TRUE.
mirror	Mirror plots are not yet implemented in this function and this argument must contain a value of NULL
main.cex	The size of the title.
max.plots.per.page	Maximum number of plots per page.
pch.ip.sp	If there is a panel with just one observation then this specifies the type of points for the DV, IPRED and PRED respectively.
cex.ip.sp	If there is a panel with just one observation then this specifies the size of the points for the DV, IPRED and PRED respectively.

`y.vals.subset` Used to subset on the DV, IPRED and PRED variables separately. Either NULL or a vector of three strings, corresponding to the subset of DV, IPRED and PRED respectively. See examples below.

... Other arguments passed to `link{xpose.plot.default}`.

Details

Matrices of individual plots are presented for comparison and closer inspection.

Value

Returns a stack of plots observations (DV) against individual predictions (IPRED) and population predictions (PRED).

A wide array of extra options controlling xyplots are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.panel.default`, `xyplot`, `strip.default`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## Monochrome, suitable for manuscript or report
ind.plots(xpdb,
  subset="ID>40 & ID<57",
  col=c(1,1,1),
```

```

lty=c(0,2,3),
strip=function(..., bg)
  strip.default(..., bg="grey"))

## Not run:
## IF we simulate in NONMEM using a dense grid of time points
## and all non-observed DV items are equal to zero.
ind.plots(xpdb,inclZeroWRES=TRUE,y.vals.subset=c("DV!=0","NULL","NULL"))

# to plot individual plots of multiple variables
ind.plots(xpdb,subset="FLAG==1")
ind.plots(xpdb,subset="FLAG==2")

## End(Not run)

```

ind.plots.cwres.hist *Histograms of weighted residuals for each individual in an Xpose data object, for Xpose 4*

Description

This is a compound plot consisting of histograms of the distribution of weighted residuals (any weighted residual available from NONMEM) for every individual in the dataset. It is a wrapper encapsulating arguments to the [xpose.plot.histogram](#) function.

Usage

```
ind.plots.cwres.hist(object, wres = "cwres", ...)
```

```

ind.plots.wres.hist(
  object,
  main = "Default",
  wres = "wres",
  ylb = NULL,
  layout = c(4, 4),
  inclZeroWRES = FALSE,
  subset = xsubset(object),
  scales = list(cex = 0.7, tck = 0.5),
  aspect = "fill",
  force.by.factor = TRUE,
  ids = F,
  as.table = TRUE,
  hicol = object@Prefs@Graph.prefs$hicol,
  hilty = object@Prefs@Graph.prefs$hilty,
  hilwd = object@Prefs@Graph.prefs$hilwd,
  hidcol = object@Prefs@Graph.prefs$hidcol,

```

```

hidlty = object@Prefs@Graph.prefs$hidlty,
hidlwd = object@Prefs@Graph.prefs$hidlwd,
hiborder = object@Prefs@Graph.prefs$hiborder,
prompt = FALSE,
mirror = NULL,
main.cex = 0.9,
max.plots.per.page = 1,
...
)

```

Arguments

object	An xpose.data object.
wres	Which weighted residual should we plot? Defaults to the WRES.
...	Other arguments passed to xpose.plot.histogram .
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
ylb	A string giving the label for the y-axis. NULL if none.
layout	A list giving the layout of the graphs on the plot, in columns and rows. The default is 4x4.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot. The default is FALSE.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
scales	see xpose.plot.histogram
aspect	see xpose.plot.histogram
force.by.factor	see xpose.plot.histogram
ids	see xpose.plot.histogram
as.table	see xpose.plot.histogram
hicol	the fill colour of the histogram - an integer or string. The default is blue (see histogram).
hilty	the border line type of the histogram - an integer. The default is 1 (see histogram).
hilwd	the border line width of the histogram - an integer. The default is 1 (see histogram).
hidcol	the fill colour of the density line - an integer or string. The default is black (see histogram).
hidlty	the border line type of the density line - an integer. The default is 1 (see histogram).
hidlwd	the border line width of the density line - an integer. The default is 1 (see histogram).
hiborder	the border colour of the histogram - an integer or string. The default is black (see histogram).

prompt	Specifies whether or not the user should be prompted to press RETURN between plot pages. Default is FALSE.
mirror	Mirror plots are not yet implemented in this function and this argument must contain a value of NULL
main.cex	The size of the title.
max.plots.per.page	Maximum number of plots per page

Details

Matrices of histograms of weighted residuals in each included individual are displayed. `ind.plots.cwres.hist` is just a wrapper for `ind.plots.wres.hist(object, wres="cwres")`.

Value

Returns a compound plot comprising histograms of weighted residual conditioned on individual.

Functions

- `ind.plots.cwres.hist()`: Histograms of conditional weighted residuals for each individual

Author(s)

E. Niclas Jonsson, Mats Karlsson, Justin Wilkins & Andrew Hooker

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb
```

```
## A plot of the first 16 individuals
ind.plots.cwres.hist(xpdb, subset="ID<18")
```

ind.plots.cwres.qq	<i>Quantile-quantile plots of weighted residuals for each individual in an Xpose data object, for Xpose 4</i>
--------------------	---

Description

This is a compound plot consisting of QQ plots of the distribution of weighted residuals (any weighted residual produced by NONMEM) for every individual in the dataset. The function is a wrapper encapsulating arguments to the [xpose.plot.qq](#) function.

Usage

```
ind.plots.cwres.qq(object, wres = "cwres", ...)
```

```
ind.plots.wres.qq(
  object,
  main = "Default",
  wres = "wres",
  layout = c(4, 4),
  inclZeroWRES = FALSE,
  subset = xsubset(object),
  scales = list(cex = 0.7, tck = 0.5),
  aspect = "fill",
  force.by.factor = TRUE,
  ids = F,
  as.table = TRUE,
  type = "o",
  pch = object@Prefs@Graph.prefs$pch,
  col = object@Prefs@Graph.prefs$col,
  cex = object@Prefs@Graph.prefs$cex,
  abllty = object@Prefs@Graph.prefs$abllty,
  abllwd = object@Prefs@Graph.prefs$abllwd,
  ablcol = object@Prefs@Graph.prefs$ablcol,
  prompt = FALSE,
  main.cex = 0.9,
  mirror = NULL,
  max.plots.per.page = 1,
  ...
)
```

Arguments

object	An xpose.data object.
--------	-----------------------

wres	Which weighted residual should we plot? Defaults to the WRES.
...	Other arguments passed to <code>link{xpose.plot.qq}</code> .
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
layout	A list giving the layout of the graphs on the plot, in columns and rows. The default is 4x4.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot. The default is FALSE.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
scales	See xpose.plot.qq .
aspect	See xpose.plot.qq .
force.by.factor	See xpose.plot.qq .
ids	See xpose.plot.qq .
as.table	See xpose.plot.qq .
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c" for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
pch	The plotting character, or symbol, to use. Specified as an integer. See R help on points . The default is an open circle.
col	The color for lines and points. Specified as an integer or a text string. A full list is obtained by the R command <code>colours()</code> . The default is blue (<code>col=4</code>).
cex	The amount by which plotting text and symbols should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'.
abllty	Line type of the line of identity.
abllwd	Line width of the line of identity.
ablcol	Line colour of the line of identity.
prompt	Specifies whether or not the user should be prompted to press RETURN between plot pages. Default is FALSE.
main.cex	The size of the title.
mirror	Mirror plots are not yet implemented in this function and this argument must contain a value of NULL
max.plots.per.page	Maximum number of plots per page

Details

Matrices of Q-Q plots of weighted residuals in each included individual are displayed.

A wide array of extra options controlling Q-Q plots are available. See [xpose.plot.qq](#) for details.

Value

Returns a compound plot comprising QQ plots of weighted residuals conditioned on individual.

Functions

- `ind.plots.cwres.qq()`: Q-Q plots of conditional weighted residuals for each individual

Author(s)

E. Niclas Jonsson, Mats Karlsson, Justin Wilkins & Andrew Hooker

See Also

[xpose.plot.qq](#), [xpose.panel.qq](#), [qqplot](#), [qqmath](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
ind.plots.cwres.qq(simpraz.xpdb, subset="ID<18")
```

`ipred.vs.idv`

Individual predictions (IPRED) plotted against the independent variable (IDV) for Xpose 4

Description

This is a plot of Individual predictions (IPRED) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
ipred.vs.idv(object, smooth = TRUE, ...)
```

Arguments

object	An xpose.data object.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of IPRED vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

ipred.vs.idv(xpdb)
```

```
## A conditioning plot
ipred.vs.idv(xpdb, by="HCTZ")

## Logarithmic Y-axis
ipred.vs.idv(xpdb, logy=TRUE)

## Custom colours and symbols, IDs
ipred.vs.idv(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)
```

iwres.dist.hist *Histogram of individual weighted residuals (IWRES), for Xpose 4*

Description

This is a histogram of the distribution of individual weighted residuals (IWRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.histogram` function.

Usage

```
iwres.dist.hist(object, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>...</code>	Other arguments passed to <code>xpose.plot.histogram</code> .

Details

Displays a histogram of the individual weighted residuals (IWRES).

Value

Returns a histogram of individual weighted residuals (IWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#),

cat.dv.vs.idv.sb(), cat.pc(), cov.splom(), cwres.dist.hist(), cwres.dist.qq(), cwres.vs.cov(), cwres.vs.idv(), cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(), cwres.wres.vs.pred(), d0FV.vs.cov(), d0FV.vs.id(), d0FV1.vs.d0FV2(), data.checkout(), dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(), dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(), ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(), parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(), wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(), xpose4-package

Examples

```
iwres.dist.hist(simpraz.xpdb)
```

iwres.dist.qq	<i>Quantile-quantile plot of individual weighted residuals (IWRES), for Xpose 4</i>
---------------	---

Description

This is a QQ plot of the distribution of individual weighted residuals (IWRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.qq` function.

Usage

```
iwres.dist.qq(object, ...)
```

Arguments

object	An <code>xpose.data</code> object.
...	Other arguments passed to <code>link{xpose.plot.qq}</code> .

Details

Displays a QQ plot of the individual weighted residuals (IWRES).

Value

Returns a QQ plot of individual weighted residuals (IWRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.qq`, `xpose.panel.qq`, `qqmath`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
iwres.dist.qq(simpraz.xpdb)
```

`iwres.vs.idv`

Individual weighted residuals (IWRES) plotted against the independent variable (IDV) for Xpose 4

Description

This is a plot of individual weighted residuals (IWRES) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
iwres.vs.idv(object, abline = c(0, 0), smooth = TRUE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>abline</code>	Vector of arguments to the <code>panel.abline</code> function. No abline is drawn if NULL. Here, the default is <code>c(0,0)</code> , specifying a horizontal line at $y=0$.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of IWRES vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp.addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

iwres.vs.idv(xpdb)

## A conditioning plot
iwres.vs.idv(xpdb, by="HCTZ")
```

`kaplan.plot`*Kaplan-Meier plots of (repeated) time-to-event data*

Description

Kaplan-Meier plots of (repeated) time-to-event data. Includes VPCs.

Usage

```
kaplan.plot(  
  x = "TIME",  
  y = "DV",  
  id = "ID",  
  data = NULL,  
  evid = "EVID",  
  by = NULL,  
  xlab = "Time",  
  ylab = "Default",  
  object = NULL,  
  events.to.plot = "All",  
  sim.data = NULL,  
  sim.zip.file = NULL,  
  VPC = FALSE,  
  nsim.lab = "simNumber",  
  sim.evct.lab = "counter",  
  probs = c(0.025, 0.975),  
  add.baseline = T,  
  add.last.area = T,  
  subset = NULL,  
  main = "Default",  
  main.sub = "Default",  
  main.sub.cex = 0.8,  
  nbins = NULL,  
  real.type = "1",  
  real.lty = 1,  
  real.lwd = 1,  
  real.col = "blue",  
  real.se = if (!is.null(sim.data)) F else T,  
  real.se.type = "1",  
  real.se.lty = 2,  
  real.se.lwd = 0.5,  
  real.se.col = "red",  
  cens.type = "1",  
  cens.lty = 1,  
  cens.col = "black",  
  cens.lwd = 1,  
  cens.rll = 0.02,
```

```

inclZeroWRES = TRUE,
onlyfirst = FALSE,
samp = NULL,
poly.alpha = 1,
poly.fill = "lightgreen",
poly.line.col = "darkgreen",
poly.lty = 2,
censor.lines = TRUE,
ylim = c(-5, 105),
cov = NULL,
cov.fun = "mean",
...
)

```

Arguments

x	The independent variable.
y	The dependent variable. event (>0) or no event (0).
id	The ID variable in the dataset.
data	A dataset can be used instead of the data in an Xpose object. Must have the same form as an xpose data object xpdb@Data.
evid	The EVID data item. If not present then all rows are considered events (can be censored or an event). Otherwise, EVID!=0 are dropped from the data set.
by	A vector of conditioning variables.
xlab	X-axis label
ylab	Y-axis label
object	An Xpose object. Needed if no data is supplied.
events.to.plot	Vector of events to be plotted. "All" means that all events are plotted.
sim.data	The simulated data file. Should be a table file with one header row and have, at least, columns with headers corresponding to x, y, id, by (if used), nsim.lab and sim.evct.lab.
sim.zip.file	The sim.data can be in \.zip format and xpose will unzip the file before reading in the data. Must have the same structure as described above in sim.data.
VPC	TRUE or FALSE. If TRUE then Xpose will search for a zipped file with name paste("simtab",object@Runno, ".zip", sep=""), for example "simtab42.zip".
nsim.lab	The column header for sim.data that contains the simulation number for that row in the data.
sim.evct.lab	The column header for sim.data that contains the individual event counter information. For each individual the event counter should increase by one for each event (or censored event) that occurs.
probs	The probabilities (non-parametric percentiles) to use in computation of the prediction intervals for the simulated data.
add.baseline	Should a (x=0,y=1) baseline measurement be added to each individual in the dataset. Otherwise each plot will begin at the first event in the dataset.

<code>add.last.area</code>	Should an area be added to the VPC extending the last PI?
<code>subset</code>	The subset of the data and <code>sim.data</code> to use.
<code>main</code>	The title of the plot. Can also be NULL or "Default".
<code>main.sub</code>	The title of the subplots. Must be a list, the same length as the number of subplots (actual graphs), or NULL or "Default".
<code>main.sub.cex</code>	The size of the title of the subplots.
<code>nbins</code>	The number of bins to use in the VPC. If NULL, the the number of unique <code>x</code> values in <code>sim.data</code> is used.
<code>real.type</code>	Type for the real data.
<code>real.lty</code>	Line type (<code>lty</code>) for the curve of the original (or real) data.
<code>real.lwd</code>	Line width (<code>lwd</code>) for the real data.
<code>real.col</code>	Color for the curve of the original (or real) data.
<code>real.se</code>	Should the standard errors of the real (non simulated) data be plotted? Calculated using survfit .
<code>real.se.type</code>	Type for the standard errors.
<code>real.se.lty</code>	Line type (<code>lty</code>) for the standard error lines.
<code>real.se.lwd</code>	Line width (<code>lwd</code>) for the standard error lines.
<code>real.se.col</code>	Color for the standard error lines.
<code>cens.type</code>	Type for the censored lines.
<code>cens.lty</code>	Line type (<code>lty</code>) for the censored lines.
<code>cens.col</code>	Color for the censored lines.
<code>cens.lwd</code>	Line width for the censored lines.
<code>cens.rll</code>	The relative line length of the censored line compared to the limits of the y-axis.
<code>inclZeroWRES</code>	Include <code>WRES=0</code> rows from the real data set in the plots?
<code>onlyfirst</code>	Include only the first measurement for the real data in the plots?
<code>samp</code>	Simulated data in the <code>xpose</code> data object can be used as the "real" data. <code>samp</code> is a number selecting which simulated data set to use.
<code>poly.alpha</code>	The transparency of the VPC shaded region.
<code>poly.fill</code>	The fill color of the VPC shaded region.
<code>poly.line.col</code>	The line colors for the VPC region.
<code>poly.lty</code>	The line type for the VPC region.
<code>sensor.lines</code>	Should censored observations be marked on the plot?
<code>ylim</code>	Limits for the y-axes
<code>cov</code>	The covariate in the dataset to plot instead of the survival curve.
<code>cov.fun</code>	The summary function for the covariate in the dataset to plot instead of the survival curve.
<code>...</code>	Additional arguments passed to the function.

Value

returns an object of class "xpose.multiple.plot".

Author(s)

Andrew C. Hooker

See Also

[survfit](#), [Surv](#), [xpose.multiple.plot](#).

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
library(xpose4)

## Read in the data
runno <- "57"
xpdb <- xpose.data(runno)

#####
# here are the real data plots
#####

kaplan.plot(x="TIME",y="DV",object=xpdb)
kaplan.plot(x="TIME",y="DV",object=xpdb,
            events.to.plot=c(1,2),
            by=c("DOSE==0","DOSE!=0"))
kaplan.plot(x="TIME",y="DV",object=xpdb,
            events.to.plot=c(1,2),
            by=c("DOSE==0","DOSE==10",
                "DOSE==50","DOSE==200"))

## make a PDF of the plots
pdf(file=paste("run",runno,"_kaplan.pdf",sep=""))
```

```

kaplan.plot(x="TIME",y="DV",object=xpdb,
            by=c("DOSE==0","DOSE==10",
                "DOSE==50","DOSE==200"))
dev.off()

#####
## VPC plots
#####

kaplan.plot(x="TIME",y="DV",object=xpdb,VPC=T,events.to.plot=c(1))
kaplan.plot(x="TIME",y="DV",object=xpdb,VPC=T,
            events.to.plot=c(1,2,3),
            by=c("DOSE==0","DOSE!=0"))
kaplan.plot(x="TIME",y="DV",object=xpdb,VPC=T,
            events.to.plot=c(1),
            by=c("DOSE==0","DOSE==10","DOSE==50","DOSE==200"))

## make a PDF of all plots
pdf(file=paste("run",runno,"_kaplan.pdf",sep=""))
kaplan.plot(x="TIME",y="DV",object=xpdb,VPC=T,
            by=c("DOSE==0","DOSE==10","DOSE==50","DOSE==200"))
dev.off()

## End(Not run)

```

make.sb.data

Make stacked bar data set.

Description

Function to make stacked bar data set for categorical data plots.

Usage

```
make.sb.data(data, idv, dv, nbins = 6, by = NULL, by.nbins = 6, ...)
```

Arguments

data	Data set to transform.
idv	the independent variable.
dv	the dependent variable.
nbins	the number of bins.
by	Conditioning variable.
by.nbins	by.nbins.
...	additional arguments.

Author(s)

The Xpose team.

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

 npc.coverage

Function to plot the coverage of the Numerical Predictive Check

Description

This function takes the output from the npc command in Perl Speaks NONMEM (PsN) and makes a coverage plot. A coverage plot for the NPC looks at different prediction intervals (PIs) for each data point and calculates the total number of data points in the data set lying outside of these PIs. The plot shows the relative amount of data points outside of their PI compared to the expected amount at that PI. In addition a confidence interval around these values are computed based on the simulated data.

Usage

```
npc.coverage(
  npc.info = "npc_results.csv",
  main = "Default",
  main.sub = NULL,
  main.sub.cex = 0.85,
  ...
)
```

Arguments

npc.info	The results file from the npc command in PsN. For example, 'npc_results.csv', or if the file is in a separate directory './npc_dir1/npc_results.csv'.
main	A string giving the plot title or NULL if none. "Default" creates a default title.
main.sub	Used for names above each plot when using multiple plots. Should be a vector c("Group 1", "Group 2")
main.sub.cex	The size of the main.sub titles.
...	Other arguments passed to xpose.multiple.plot.default , xyplot and others. Please see these functions (and below) for more descriptions of what you can do.

Value

A list of plots

Additional arguments for the NPC coverage plots**Additional plot features**

CI Specifies whether confidence intervals (as lines, a shaded area or both) should be added to the plot. Allowed values are: "area", "lines", "both", or NULL.

mark.outside.data Should the points outside the CI be marked in a different color to identify them. Allowed values are TRUE or FALSE.

abline Should there be a line to mark the value of $y=1$? Possible values are TRUE, FALSE and NULL.

Line and area control. See [plot](#), [grid.polygon](#) and [xyplot](#) for more details.

CI.area.col Color of the area for the CI. Defaults to "blue"

CI.area.alpha Transparency of the CI.area.col. Defaults to 0.3.

ab.lwd The width of the abline. Default is 1.

ab.lty Line type of the abline. Default is "dashed"

CI.upper.lty Line type of the line at the upper edge of the CI.

CI.upper.col Color of the line at the upper edge of the CI.

CI.upper.lwd The line width of the line at the upper edge of the CI.

CI.lower.lty The line type at the lower edge of the CI.

CI.lower.col The color of the line at the lower edge of the CI.

CI.lower.lwd The line width of the line at the lower edge of the CI.

obs.col The color of the observed values.

obs.pch The type of point to use for the observed values.

obs.lty The type of line to use for the observed values.

obs.type The combination of lines and points to use for the observed values. Default is "b" for both.

obs.cex The size of the points to use for the observed values.

obs.lwd The line width to use for the observed values.

out.col The color of the observed values that lie outside of the CI. Only used if mark.outside.data=TRUE.

out.pch The type of point to use for the observed values that lie outside of the CI. Only used if mark.outside.data = TRUE.

out.cex The size of the points of the observed values that lie outside of the CI. Only used if mark.outside.data = TRUE.

out.lwd The line width of the observed values that lie outside of the CI. Only used if mark.outside.data = TRUE.

Author(s)

Andrew Hooker

See Also

[read.npc.vpc.results](#) [xpose.multiple.plot.default](#) [xyplot](#)

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
library(xpose4)

npc.coverage()

## to read files in a directory different than the current working directory
npc.file <- "../another_directory/npc_results.csv"
npc.coverage(npc.info=npc.file)

## End(Not run)
```

nsim

Extract or set the value of the Nsim slot.

Description

Extract or set the value of the Nsim slot of an "xpose.data" object.

Usage

```
nsim(object)
```

Arguments

object An "xpose.data" object.

Author(s)

Niclas Jonsson

See Also

[xpose.data-class](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Report number of simulations
nsim(xpdb5)

## End(Not run)
```

 parm.vs.cov

Parameters plotted against covariates, for Xpose 4

Description

This creates a stack of plots of Bayesian parameter estimates plotted against covariates, and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
parm.vs.cov(
  object,
  onlyfirst = TRUE,
  smooth = TRUE,
  type = "p",
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>onlyfirst</code>	Logical value indicating whether only the first row per individual is included in the plot.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>type</code>	The plot type - defaults to points only.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the parameters in the Xpose data object, as specified in `object@Prefs@Xvardef$params`, is plotted against each covariate present, as specified in `object@Prefs@Xvardef$covariates`, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns a stack of xyplots and histograms of parameters against covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.plot.histogram`, `xyplot`, `histogram`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov qq`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb <- xpose.data(5)

## A vanilla plot
parm.vs.cov(xpdb)

## Custom colours and symbols, IDs
parm.vs.cov(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

## End(Not run)
```

 parm.vs.parm

Plot parameters vs other parameters

Description

This function plots the parameter values stored in an Xpose data object versus each other in a series of graphs. The mirror functionality is available for this function.

Usage

```
parm.vs.parm(
  object,
  onlyfirst = TRUE,
  abline = FALSE,
  smooth = TRUE,
  type = "p",
  main = "Default",
  ...
)
```

Arguments

object	An xpose.data object.
onlyfirst	Logical value indicating whether only the first row per individual is included in the plot.
abline	Allows for a line of identity.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
type	The plot type - defaults to points only.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to xpose.plot.default.

Details

Each of the parameters in the Xpose data object, as specified in `object@Prefs@Xvardef$params`, is plotted against the rest, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a stack of xyplots and histograms of parameters against parameters.

Author(s)

Andrew Hooker

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb <- xpose.data(5)

parm.vs.parm(xpdb)

parm.vs.parm(xpdb,mirror=3)

## End(Not run)
```

par_cov_hist

*Plot the parameter or covariate distributions using a histogram***Description**

These functions plot the parameter or covariate values stored in an Xpose data object using histograms.

Usage

```
cov.hist(object, onlyfirst = TRUE, main = "Default", ...)

parm.hist(object, onlyfirst = TRUE, main = "Default", ...)

ranpar.hist(object, onlyfirst = TRUE, main = "Default", ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>onlyfirst</code>	Logical value indicating if only the first row per individual is included in the plot.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to xpose.plot.histogram .

Details

Each of the parameters or covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$parms`, `object@Prefs@Xvardef$covariates` or `object@Prefs@Xvardef$ranpar` is evaluated in turn, creating a stack of histograms.

A wide array of extra options controlling histograms are available. See [xpose.plot.histogram](#) for details.

Value

Delivers a stack of histograms.

Functions

- `cov.hist()`: Covariate distributions
- `parm.hist()`: parameter distributions
- `ranpar.hist()`: random parameter distributions

Author(s)

Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#), [xpose.data-class](#), [xpose.prefs-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#),

```

cwres.vs.idv(), cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(),
cwres.wres.vs.pred(), dOFV.vs.cov(), dOFV.vs.id(), dOFV1.vs.dOFV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_qq, parm.vs.cov(), parm.vs.parm(),
pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(),
wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(),
xpose.VPC.categorical(), xpose4-package

```

Examples

```

## Here we load the example xpose database
xpdb <- simpraz.xpdb

## Parameter histograms
parm.hist(xpdb)

## Covariate distribution, in green
cov.hist(xpdb, hicol=11, hidcol="DarkGreen", hiborder="White")

## Random parameter histograms
ranpar.hist(xpdb)

```

par_cov_qq	<i>Plot the parameter or covariate distributions using quantile-quantile (Q-Q) plots</i>
------------	--

Description

These functions plot the parameter or covariate values stored in an Xpose data object using Q-Q plots.

Usage

```

cov.qq(object, onlyfirst = TRUE, main = "Default", ...)

parm.qq(object, onlyfirst = TRUE, main = "Default", ...)

ranpar.qq(object, onlyfirst = TRUE, main = "Default", ...)

```

Arguments

object	An xpose.data object.
onlyfirst	Logical value indicating if only the first row per individual is included in the plot.

`main` The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.

`...` Other arguments passed to `xpose.plot.qq`.

Details

Each of the parameters or covariates in the Xpose data object, as specified in `object@Prefs@Xvardef$params`, `object@Prefs@Xvardef$ranpar` or `object@Prefs@Xvardef$covariates`, is evaluated in turn, creating a stack of Q-Q plots.

A wide array of extra options controlling Q-Q plots are available. See `xpose.plot.qq` for details.

Value

Delivers a stack of Q-Q plots.

Functions

- `cov.qq()`: Covariate distributions
- `parm.qq()`: parameter distributions
- `ranpar.qq()`: random parameter distributions

Author(s)

Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.qq`, `xpose.panel.qq`, `qqmath`, `xpose.data-class`, `xpose.prefs-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

## parameter histograms
parm.qq(xpdb)

## A stack of random parameter histograms
ranpar.qq(xpdb)

## Covariate distribution, in green with red line of identity
cov.qq(xpdb, col=11, ablcol=2)
```

par_cov_summary	<i>Summarize individual parameter values and covariates</i>
-----------------	---

Description

These functions produce tables, printed to the screen, summarizing the individual parameter values or covariates from a dataset in Xpose 4.

Usage

```
cov.summary(  
  object,  
  onlyfirst = TRUE,  
  subset = xsubset(object),  
  inclZeroWRES = FALSE,  
  out.file = ".screen",  
  main = "Default",  
  fill = "gray",  
  values.to.use = xvardef("covariates", object),  
  value.name = "Covariate",  
  ...  
)
```

```
parm.summary(  
  object,  
  onlyfirst = TRUE,  
  subset = xsubset(object),  
  inclZeroWRES = FALSE,  
  out.file = ".screen",  
  main = "Default",  
  fill = "gray",  
  values.to.use = xvardef("parms", object),  
  value.name = "Parameter",
```

```

    max.plots.per.page = 1,
    ...
)

```

Arguments

object	An xpose.data object.
onlyfirst	Logical value indicating if only the first row per individual is included in the plot.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
inclZeroWRES	Logical value indicating whether rows with WRES=0 are included in the plot. The default is FALSE.
out.file	Where the results should be output to. Can be ".screen", ".ask", ".graph" or a filename in quotes.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
fill	The color to fill the boxes in the table if the table is printed to ".graph"
values.to.use	Which values should be summarized
value.name	The name of the values
...	Other arguments passed to Data and SData.
max.plots.per.page	Maximum plots per page.

Value

Returned is the matrix of values from the table. `parm.summary` and `cov.summary` produce summaries of parameters and covariates, respectively. `parm.summary` produces less attractive output but supports mirror functionality.

`parm.summary` and `cov.summary` utilize [print.char.matrix](#) to print the information to the screen.

Functions

- `cov.summary()`: Covariate summary
- `parm.summary()`: Parameter summary

Author(s)

Andrew Hooker & Justin Wilkins

See Also

[Data](#), [SData](#), [xpose.data-class](#), [print.char.matrix](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data.extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#),

```
make.sb.data(), nsim(), read.TTE.sim.data(), read.nm.tables(), read_NM_output, read_nm_table(),  
simprazExample(), tabulate.parameters(), xlabel(), xpose.data, xpose.print(), xpose4-package,  
xsubset()
```

Examples

```
parm.summary(simpraz.xpdb)
```

pred.vs.idv	<i>Population predictions (PRED) plotted against the independent variable (IDV) for Xpose 4</i>
-------------	---

Description

This is a plot of population predictions (PRED) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
pred.vs.idv(object, smooth = TRUE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of PRED vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.panel.default](#), [xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

pred.vs.idv(xpdb)

## A conditioning plot
pred.vs.idv(xpdb, by="HCTZ")

## Logarithmic Y-axis
pred.vs.idv(xpdb, logy=TRUE)

## Custom colours and symbols, IDs
pred.vs.idv(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)
```

```
print.xpose.multiple.plot
```

Print an Xpose multiple plot object.

Description

Print an Xpose multiple plot object, which is the output from the function [xpose.multiple.plot](#).

Usage

```
## S3 method for class 'xpose.multiple.plot'
print(x, ...)
```

Arguments

x Output object from the function `xpose.multiple.plot`.
 ... Additional options passed to function.

Details

Print method for a plot class.

Author(s)

Niclas Jonsson and Andrew C. Hooker

See Also

`xpose.multiple.plot`.

randtest.hist	<i>Function to create a histogram of results from the randomization test tool (randtest) in R</i> hrefhttps://uupharmacometrics.github.io/PsN/PsN
---------------	--

Description

Reads results from the randtest tool in **PsN** and then creates a histogram.

Usage

```
randtest.hist(  
  results.file = "raw_results_run1.csv",  
  df = 1,  
  p.val = 0.05,  
  main = "Default",  
  xlim = NULL,  
  PCTSlcol = "black",  
  vlcol = c("red", "orange"),  
  ...  
)
```

Arguments

results.file The location of the results file from the randtest tool in **PsN**
 df The degrees of freedom between the full and reduced model used in the randomization test.
 p.val The p-value you would like to use.
 main The title of the plot.
 xlim The limits of the x-axis

PCTS1col	Color of the empirical line
v1col	Colors of the original and nominal line
...	Additional arguments that can be passed to xpose.plot.histogram , xpose.panel.histogram , histogram and other lattice-package functions.

Value

A lattice object

Author(s)

Andrew Hooker

References

[PsN](#)

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#) and other [lattice-package](#) functions.

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
randtest.hist(results.file="randtest_dir1/raw_results_run1.csv",df=2)

## End(Not run)
```

Description

This creates a stack of plots of Bayesian random parameter estimates plotted against covariates, and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```

ranpar.vs.cov(
  object,
  onlyfirst = TRUE,
  smooth = TRUE,
  type = "p",
  main = "Default",
  ...
)

```

Arguments

object	An xpose.data object.
onlyfirst	Logical value indicating whether only the first row per individual is included in the plot.
smooth	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
type	The plot type - defaults to points only.
main	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
...	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Each of the random parameters (ETAs) in the Xpose data object, as specified in `object@Prefs@Xvardef$ranpar`, is plotted against each covariate present, as specified in `object@Prefs@Xvardef$covariates`, creating a stack of plots.

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns a stack of xyplots and histograms of random parameters against covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.plot.histogram](#), [xyplot](#), [histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#),

```

cwres.vs.idv(), cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(),
cwres.wres.vs.pred(), dOFV.vs.cov(), dOFV.vs.id(), dOFV1.vs.dOFV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(),
parm.vs.parm(), pred.vs.idv(), runsum(), wres.dist.hist(), wres.dist.qq(), wres.vs.idv(),
wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(),
xpose.VPC.categorical(), xpose4-package

```

Examples

```

## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb <- xpose.data(5)

## A vanilla plot
ranpar.vs.cov(xpdb)

## Custom colours and symbols, IDs
ranpar.vs.cov(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

## End(Not run)

```

read.nm.tables

Reading NONMEM table files

Description

Reads one or more NONMEM table files, removes duplicated columns and merges the data into a data.frame.

Usage

```

read.nm.tables(
  table.files = NULL,
  runno = NULL,
  tab.suffix = "",
  table.names = c("sdtab", "mutab", "patab", "catab", "cotab", "mytab", "extra", "xptab"),
  cwres.name = c("cwtab"),
  cwres.suffix = "",
  quiet = FALSE,
  new_methods = TRUE,
  ...
)

```

Arguments

<code>table.files</code>	Exact names of table files to read. If not provided then the exact names are created using the other arguments to the function.
<code>runno</code>	Run-number to identify sets of table files.
<code>tab.suffix</code>	Table file name suffix.
<code>table.names</code>	Vector of template table file names to read.
<code>cwres.name</code>	Vector of CWRES table file names to read.
<code>cwres.suffix</code>	CWRES table file name suffix.
<code>quiet</code>	Logical value to indicate whether some warnings should be quiet or not.
<code>new_methods</code>	Should faster methods of reading tables be used (uses readr package)?
<code>...</code>	Additional arguments passed to this function

Details

Reads one or more table files, removes duplicate columns and merges the data. The function also checks to see if the table files are of the same length (required).

If there are header lines in the table files (for example if your data are simulated with `NSUB>1`), these are removed.

The table file names to read are constructed from the file name templates of `table.names`. The `runno` and `tab.suffix` are appended to the file name template before checking if the file is readable.

Xpose expects, by default, to find the following NONMEM tables in the working directory to be able to create an Xpose data object (using a run number of 5 as an example):

`sdtab5`: The 'standard' parameters, including IWRE, IPRE, TIME, and the NONMEM default items (DV, PRED, RES and WRES) that are added when NOAPPEND is not present in the \$TABLE record.

```
$TABLE ID TIME IPRE IWRE NOPRINT ONEHEADER FILE=sdtab5
```

`patab5`: The empirical Bayes estimates of individual model parameter values, or posthoc estimates. These are model parameters, such as CL, V2, ETA1, etc.

```
$TABLE ID CL V2 KA K F1 ETA1 ETA2 ETA3 NOPRINT NOAPPEND ONEHEADER FILE=patab5
```

`catab5`: Categorical covariates, e.g. SEX, RACE.

```
$TABLE ID SEX HIV GRP NOPRINT NOAPPEND ONEHEADER FILE=catab5
```

`cotab5`: Continuous covariates, e.g. WT, AGE.

```
$TABLE ID WT AGE BSA HT GGT HB NOPRINT NOAPPEND ONEHEADER FILE=cotab5
```

`mutab5`, `mytab5`, `extra5`, `xptab5`: Additional variables of any kind. These might be useful if there are more covariates than can be accommodated in the covariates tables, for example, or if you have other variables that should be added, e.g. CMAX, AUC.

Value

A dataframe.

Author(s)

Niclas Jonsson, Andrew Hooker

See Also

[xpose.data-class](#), [compute.cwres](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory, and that the table files have
## a suffix of '.dat', e.g. sdtab5.dat

my.dataframe <- read.nm.tables(5, tab.suffix = ".dat")

## End(Not run)
```

`read.npc.vpc.results` *Read the results file from a Numerical or Visual Predictive Check run in PsN*

Description

This function reads in a results file from running either the PsN command `vpc` or `npc`. The function then parses the file and passes the result to plotting functions.

Usage

```
read.npc.vpc.results(
  vpc.results = NULL,
  npc.results = NULL,
  verbose = FALSE,
  ...
)
```

Arguments

`vpc.results` The name of the results file from running the PsN command `vpc`. Often this is named `'vpc_results.csv'`. If the file is in a directory different than the working directory then you can define a relative or absolute path to the file by, for example, `'./vpc_strat_WT_4_mirror_5/vpc_results.csv'`.

<code>npc.results</code>	The name of the results file from running the PsN command <code>npc</code> . Often this is named <code>'npc_results.csv'</code> . relative or absolute paths to the file are allowed as for <code>vpc.results</code> .
<code>verbose</code>	Text messages passed to screen or not.
<code>...</code>	arguments passed to other functions.

Details

One of `vpc.results` or `npc.results` are necessary. If both or none are defined then the function does nothing and a NULL is returned from the function.

Value

A list of values is returned.

<code>model.file</code>	The model file that PsN ran either the <code>npc</code> or <code>vpc</code> with
<code>dv.var</code>	The dependent variable used in the calculations.
<code>idv.var</code>	The independent variable used in the calculations. NULL if <code>npc.results</code> is used.
<code>num.tables</code>	The number of separate tables in the results file.
<code>by.interval</code>	The conditioning interval for the stratification variable, only returned if <code>vpc.results</code> is used.
<code>result.tables</code>	The results tables from the results file. this is a list.

Author(s)

Andrew Hooker

See Also

[xpose.VPC.npc.coverage](#)

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

`read.TTE.sim.data` *Read (repeated) time-to-event simulation data files.*

Description

Read (repeated) time-to-event simulation data files.

Usage

```
read.TTE.sim.data(
  sim.file,
  subset = NULL,
  headers = c("REP", "ID", "DV", "TIME", "FLAG2", "DOSE"),
  xpose.table.file = FALSE,
  ...
)
```

Arguments

sim.file	Name of the simulated file.
subset	subset to extract.
headers	headers in file.
xpose.table.file	xpose table files.
...	Extra arguments passed to function.

Author(s)

Andrew C. Hooker

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

read.vpctab

Read the vpctab file from PsN into Xpose

Description

This function read in the vpctab file created from PsN and gathers the information needed to make a vpc plot.

Usage

```
read.vpctab(
  vpctab = NULL,
  object = NULL,
  vpc.name = "vpctab",
  vpc.suffix = "",
```

```

    tab.suffix = "",
    inclZeroWRES = FALSE,
    verbose = FALSE,
    ...
)

```

Arguments

vpctab	The vpctab file from a 'vpc' run in PsN.
object	An xpose data object. Created from xpose.data . One of object or vpctab is required. If both are present then the information from the vpctab will over-ride the xpose data object (i.e. the values from the vpctab will replace any matching values in the object@Data portion of the xpose data object). If only object is present then the function will look for a vpctab with the same run number as the one associated with the object.
vpc.name	The default name of the vpctab file. Used if only object is supplied.
vpc.suffix	The suffix of the vpctab file. Used if only object is supplied.
tab.suffix	The table suffix of the vpctab file. Used if only object is supplied. Final order of the file would be then paste(vpc.name, object@Runno, vpc.suffix, tab.suffix)
inclZeroWRES	If there are no zero valued weighted residuals in the object then this should be TRUE.
verbose	Text messages passed to screen or not.
...	Other arguments passed to other functions.

Value

Returned is an xpose data object with vpctab information included.

Author(s)

Andrew Hooker

See Also

[xpose.VPC](#)

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

read_NM_output	<i>Read NONMEM output files into Xpose 4</i>
----------------	--

Description

These are functions that read in a NONMEM output file (a '*.lst' file) and then format the input.

Usage

```
calc.npar(object)

create.parameter.list(listfile)

read.lst(filename)
```

Arguments

object	The return value of read.lst(filename)
listfile	A NONMEM output file.
filename	A NONMEM output file.

Value

lists of read values.

Functions

- `calc.npar()`: calculates the number and type of parameters included in a NONMEM output file
- `create.parameter.list()`: Reads parameters, uncertainty and termination messages included in a NONMEM output file
- `read.lst()`: parses information out of NONMEM output.

Author(s)

Niclas Jonsson, Andrew Hooker & Justin Wilkins

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data.extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

read_nm_table	<i>Read NONMEM table files produced from simulation.</i>
---------------	--

Description

The function reads in NONMEM table files produced from the \$SIM line in a NONMEM model file.

Usage

```
read_nm_table(
  nm_table,
  only_obs = FALSE,
  method = "default",
  quiet = TRUE,
  sim_num = FALSE,
  sim_name = "NSIM"
)
```

Arguments

nm_table	The NONMEM table file to read. A text string.
only_obs	Should the non-observation lines in the data set be removed? Currently filtered using the expected MDV column. TRUE or FALSE.
method	The methods to use for reading the tables, Can be "readr_1", "readr_2", "readr_3" or "slow".
quiet	Should the error message be verbose or not?
sim_num	Should a simulation number be added to simulation tables?
sim_name	What name should one use to name the column of the simulation number?

Details

Currently the function expects the \$TABLE to have a header for each new simulation. This means that the NOHEADER option or ONEHEADER option in the table file is not allowed.

Value

Returns a data frame of the simulated table with an added column for the simulation number. The data frame is given class `c("tbl_df", "tbl", "data.frame")` for easy use with [dplyr](#).

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#),

[simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

reset.graph.par	<i>Resets Xpose variable definitions to factory settings</i>
-----------------	--

Description

Function to reset Xpose's graphics parameters definitions to the default.

Usage

```
reset.graph.par(object, classic = FALSE)
```

Arguments

object	An <code>xpose.data</code> object.
classic	A logical operator specifying whether the function should assume the classic menu system. This is an internal option and need never be called from the command line.

Details

This functions is used to reset Xpose's graphic settings definitions to their default values. Graphical settings are read from the file 'xpose.ini' in the root of the 'xpose4' package.

Value

An `xpose.data` object (`classic == FALSE`) or null (`classic == TRUE`).

Author(s)

Niclas Jonsson & Justin Wilkins

See Also

[xpose.prefs-class](#), [import.graph.par](#), [change.xvardef](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Import graphics preferences you saved earlier using export.graph.par
xpdb5 <- import.graph.par(xpdb5)
```

```
## Reset to default values
xpdb5 <- reset.graph.par(xpdb5)

## Change WRES definition
xpdb5 <- change.wres(xpdb5)

## End(Not run)
```

runsum

Print run summary in Xpose 4

Description

Function to build Xpose run summaries.

Usage

```
runsum(
  object,
  dir = "",
  modfile = paste(dir, "run", object@Runno, ".mod", sep = ""),
  listfile = paste(dir, "run", object@Runno, ".lst", sep = ""),
  main = NULL,
  subset = xsubset(object),
  show.plots = TRUE,
  txt.cex = 0.7,
  txt.font = 1,
  show.ids = FALSE,
  param.table = TRUE,
  txt.columns = 2,
  force.wres = FALSE,
  ...
)
```

Arguments

object	An xpose.data object.
dir	The directory to look for the model and output file of a NONMEM run.
modfile	The name of the NONMEM control stream associated with the current run.
listfile	The name of the NONMEM output file associated with the current run.
main	A string giving the main heading. NULL if none.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
show.plots	Logical indicating if GOF plots should be shown in the run summary.
txt.cex	Number indicating the size of the txt in the run summary.

<code>txt.font</code>	Font of the text in the run summary.
<code>show.ids</code>	Logical indicating if IDs should be plotted in the plots for the run summary.
<code>param.table</code>	Logical indicating if the parameter table should be shown in the run summary.
<code>txt.columns</code>	The number of text columns in the run summary.
<code>force.wres</code>	Plot the WRES even if other residuals are available.
<code>...</code>	Other arguments passed to the various functions.

Value

A compound plot containing an Xpose run summary is created.

Author(s)

Niclas Jonsson and Andrew Hooker

See Also

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
od = setwd(tempdir()) # move to a temp directory
(cur.files <- dir()) # current files in temp directory

simprazExample(overwrite=TRUE) # write files
(new.files <- dir()[!(dir() %in% cur.files)]) # what files are new here?

xpdb <- xpose.data(1)
runsum(xpdb)

file.remove(new.files) # remove these files
setwd(od) # restore working directory
```

simpraz.xpdb	<i>Simulated prazosin Xpose database.</i>
--------------	---

Description

Xpose database from the NONMEM output of a model for prazosin using simulated data (and NONMEM 7.3).

Usage

```
simpraz.xpdb
```

Format

an xpose.data object

Details

The database can be used to test functions in Xpose 4. This database is slightly different than the database that is created when reading in the files created by [simprazExample](#) using `xpose.data`.

See Also

[simprazExample](#)

Examples

```
xpose.print(simpraz.xpdb)
Data(simpraz.xpdb)
str(simpraz.xpdb)
```

simprazExample	<i>Function to create files for the simulated prazosin example in Xpose</i>
----------------	---

Description

Creates NONMEM data, model and output files for a model of prazosin using simulated data.

Usage

```
simprazExample(overwrite = FALSE)
```

Arguments

overwrite	Logical. Should the function overwrite files with the same names already in the current working directory?
-----------	--

Details

Creates files in the current working directory named: run1.ext run1.lst run1.mod simpraz.dta xptab1

Author(s)

Niclas Jonsson and Andrew Hooker

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data.extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
od = setwd(tempdir()) # move to a temp directory
(cur.files <- dir()) # current files in temp directory

simprazExample(overwrite=TRUE) # write files

(new.files <- dir()[!(dir() %in% cur.files)]) # what files are new here?

file.remove(new.files) # remove these files
setwd(od) # restore working directory
```

tabulate.parameters *Tabulate the population parameter estimates*

Description

This function provides a summary of the model's parameter estimates and precision.

Usage

```
tabulate.parameters(object, prompt = FALSE, outfile = NULL, dir = "")
```

Arguments

object	An xpose.data object.
prompt	Ask before printing.
outfile	file to output to (NULL means screen).
dir	Which directory is the NONMEM output file located. "" means the current working directory getwd().

Value

A table summarizing the parameters and their precision.

Author(s)

Niclas Jonsson, Andrew Hooker & Justin Wilkins

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
od = setwd(tempdir()) # move to a temp directory
(cur.files <- dir()) # current files in temp directory

simprazExample(overwrite=TRUE) # write files
(new.files <- dir()[!(dir() %in% cur.files)]) # what files are new here?
xpdb <- xpose.data(1) # read in files to xpose database

tabulate.parameters(xpdb)

file.remove(new.files) # remove these files
setwd(od) # restore working directory
```

wres.dist.hist

Histogram of weighted residuals (WRES), for Xpose 4

Description

This is a histogram of the distribution of weighted residuals (WRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.histogram` function.

Usage

```
wres.dist.hist(object, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>...</code>	Other arguments passed to xpose.plot.histogram .

Details

Displays a histogram of the weighted residuals (WRES).

Value

Returns a histogram of weighted residuals (WRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.histogram](#), [xpose.panel.histogram](#), [histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.dist.hist(xpdb)
```

wres.dist.qq

Quantile-quantile plot of weighted residuals (WRES), for Xpose 4

Description

This is a QQ plot of the distribution of weighted residuals (WRES) in the dataset, a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.qq` function.

Usage

```
wres.dist.qq(object, ...)
```

Arguments

```
object      An xpose.data object.
...         Other arguments passed to link{xpose.plot.qq}.
```

Details

Displays a QQ plot of the weighted residuals (WRES).

Value

Returns a QQ plot of weighted residuals (WRES).

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.qq](#), [xpose.panel.qq](#), [qqmath](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [dOFV.vs.cov\(\)](#), [dOFV.vs.id\(\)](#), [dOFV1.vs.dOFV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.dist.qq(xpdb)
```

wres.vs.cov

*Weighted residuals (WRES) plotted against covariates, for Xpose 4***Description**

This creates a stack of plots of weighted residuals (WRES) plotted against covariates, and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` and `xpose.plot.histogram` functions. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
wres.vs.cov(
  object,
  ylb = "WRES",
  smooth = TRUE,
  type = "p",
  main = "Default",
  ...
)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>smooth</code>	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
<code>type</code>	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': 'p' for points, 'l' for lines, 'o' for over-plotted points and lines, 'b', 'c' for (empty if 'c') points joined by lines, 's' and 'S' for stair steps and 'h' for histogram-like vertical lines. Finally, 'n' does not produce any points or lines.
<code>main</code>	The title of the plot. If "Default" then a default title is plotted. Otherwise the value should be a string like "my title" or NULL for no plot title.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> or <code>link{xpose.plot.histogram}</code> .

Details

Weighted residuals (WRES) are plotted against each covariate present, as specified in `object@Prefs@Xvardef$covariates`, creating a stack of plots.

A wide array of extra options controlling xyplots and histograms are available. See [xpose.plot.default](#) and [xpose.plot.histogram](#) for details.

Value

Returns a stack of xyplots and histograms of CWRES versus covariates.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xpose.plot.histogram](#), [xyplot](#), [histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Examples

```
## Not run:
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Here we load the example xpose database
data(simpraz.xpdb)
xpdb <- simpraz.xpdb

## A vanilla plot
wres.vs.cov(xpdb)

## Custom colours and symbols, IDs
wres.vs.cov(xpdb, cex=0.6, pch=3, col=1, ids=TRUE)

## End(Not run)
```

wres.vs.idv	<i>Population weighted residuals (WRES) plotted against the independent variable (IDV) for Xpose 4</i>
-------------	--

Description

This is a plot of population weighted residuals (WRES) vs the independent variable (IDV), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
wres.vs.idv(object, abline = c(0, 0), smooth = TRUE, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>abline</code>	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
<code>smooth</code>	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

Weighted residuals (WRES) are plotted against the independent variable, as specified in `object@Prefs@Xvardef$idv`.

A wide array of extra options controlling xyplots are available. See `xpose.plot.default` and `xpose.panel.default` for details.

Value

Returns an xyplot of WRES vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.default`, `xpose.panel.default`, `xyplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `d0FV.vs.cov()`, `d0FV.vs.id()`, `d0FV1.vs.d0FV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `wres.vs.pred.bw()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.vs.idv(xpdb)

## A conditioning plot
wres.vs.idv(xpdb, by="HCTZ")
```

wres.vs.idv.bw	<i>Box-and-whisker plot of weighted residuals vs the independent variable for Xpose 4</i>
----------------	---

Description

This creates a box and whisker plot of weighted residuals (WRES) vs the independent variable (IDV), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
wres.vs.idv.bw(object, ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.bw}</code> .

Details

This creates a box and whisker plot of weighted residuals (WRES) vs the independent variable (IDV), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

A wide array of extra options controlling bwplots are available. See [xpose.plot.bw](#) and [xpose.panel.bw](#) for details.

Value

Returns a stack of box-and-whisker plots of WRES vs IDV.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.bw](#), [xpose.panel.bw](#), [bwplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#),

```

cwres.wres.vs.pred(), d0FV.vs.cov(), d0FV.vs.id(), d0FV1.vs.d0FV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(),
parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(),
wres.vs.idv(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.both(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.vs.idv.bw(xpdb)

```

wres.vs.pred	<i>Population weighted residuals (WRES) plotted against population predictions (PRED) for Xpose 4</i>
--------------	---

Description

This is a plot of population weighted residuals (WRES) vs population predictions (PRED), a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.default` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
wres.vs.pred(object, smooth = TRUE, abline = c(0, 0), ...)
```

Arguments

<code>object</code>	An <code>xpose.data</code> object.
<code>smooth</code>	Logical value indicating whether an x-y smooth should be superimposed. The default is TRUE.
<code>abline</code>	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
<code>...</code>	Other arguments passed to <code>link{xpose.plot.default}</code> .

Details

A wide array of extra options controlling xyplots are available. See [xpose.plot.default](#) and [xpose.panel.default](#) for details.

Value

Returns an xyplot of WRES vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.default](#), [xyplot](#), [xpose.prefs-class](#), [compute.cwres](#), [xpose.data-class](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.idv\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.vs.pred(xpdb)

## A conditioning plot
wres.vs.pred(xpdb, by="HCTZ")
```

wres.vs.pred.bw	<i>Box-and-whisker plot of weighted residuals vs population predictions for Xpose 4</i>
-----------------	---

Description

This creates a box and whisker plot of weighted residuals (WRES) vs population predictions (PRED), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

Usage

```
wres.vs.pred.bw(object, ...)
```

Arguments

object An xpose.data object.
 ... Other arguments passed to `link{xpose.plot.bw}`.

Details

This creates a box and whisker plot of weighted residuals (WRES) vs population predictions (PRED), and is a specific function in Xpose 4. It is a wrapper encapsulating arguments to the `xpose.plot.bw` function. Most of the options take their default values from `xpose.data` object but may be overridden by supplying them as arguments.

A wide array of extra options controlling bwplots are available. See `xpose.plot.bw` and `xpose.panel.bw` for details.

Value

Returns a box-and-whisker plot of WRES vs PRED.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

`xpose.plot.bw`, `xpose.panel.bw`, `bwplot`, `xpose.prefs-class`, `xpose.data-class`

Other specific functions: `absval.cwres.vs.cov.bw()`, `absval.cwres.vs.pred()`, `absval.cwres.vs.pred.by.cov()`, `absval.iwres.cwres.vs.ipred.pred()`, `absval.iwres.vs.cov.bw()`, `absval.iwres.vs.idv()`, `absval.iwres.vs.ipred()`, `absval.iwres.vs.ipred.by.cov()`, `absval.iwres.vs.pred()`, `absval.wres.vs.cov.bw()`, `absval.wres.vs.idv()`, `absval.wres.vs.pred()`, `absval.wres.vs.pred.by.cov()`, `absval_delta_vs_cov_model_comp()`, `addit.gof()`, `autocorr.cwres()`, `autocorr.iwres()`, `autocorr.wres()`, `basic.gof()`, `basic.model.comp()`, `cat.dv.vs.idv.sb()`, `cat.pc()`, `cov.splom()`, `cwres.dist.hist()`, `cwres.dist.qq()`, `cwres.vs.cov()`, `cwres.vs.idv()`, `cwres.vs.idv.bw()`, `cwres.vs.pred()`, `cwres.vs.pred.bw()`, `cwres.wres.vs.idv()`, `cwres.wres.vs.pred()`, `dOFV.vs.cov()`, `dOFV.vs.id()`, `dOFV1.vs.dOFV2()`, `data.checkout()`, `dv.preds.vs.idv()`, `dv.vs.idv()`, `dv.vs.ipred()`, `dv.vs.ipred.by.cov()`, `dv.vs.ipred.by.idv()`, `dv.vs.pred()`, `dv.vs.pred.by.cov()`, `dv.vs.pred.by.idv()`, `dv.vs.pred.ipred()`, `gof()`, `ind.plots()`, `ind.plots.cwres.hist()`, `ind.plots.cwres.qq()`, `ipred.vs.idv()`, `iwres.dist.hist()`, `iwres.dist.qq()`, `iwres.vs.idv()`, `kaplan.plot()`, `par_cov_hist`, `par_cov_qq`, `parm.vs.cov()`, `parm.vs.parm()`, `pred.vs.idv()`, `ranpar.vs.cov()`, `runsum()`, `wres.dist.hist()`, `wres.dist.qq()`, `wres.vs.idv()`, `wres.vs.idv.bw()`, `wres.vs.pred()`, `xpose.VPC()`, `xpose.VPC.both()`, `xpose.VPC.categorical()`, `xpose4-package`

Examples

```
## Here we load the example xpose database
xpdb <- simpraz.xpdb

wres.vs.pred.bw(xpdb)
```

xlabel	<i>Extract and set labels for Xpose data items.</i>
--------	---

Description

This function extracts and sets label definitions in Xpose data objects.

Usage

```
xlabel(x, object)
```

```
xlabel(object) <- value
```

Arguments

x	Name of the variable to assign a label to.
object	An <code>xpose.data</code> object.
value	A two element vector of which the first element is the name of the variable and the second the label

Details

x should be a string exactly matching the name of a column in the data.frame in the Data slot of an `xpose.data` object. The name of columns defined through `xpose` variable definitions (see [xpose.data](#)) can be extracted using the `xvardef` function and to be used in the `xlabel` function, e.g. `xlabel(xvardef("dv", object), object)`, which would give the label for the `dv` variable.

Value

The label of the specified column.

Functions

- `xlabel(object) <- value`: sets label definitions in Xpose data objects. assigned value should be a two-element vector of which the first element is the name of the variable and the second the label

Author(s)

Niclas Jonsson

See Also

[xpose.prefs-class](#), [xvardef](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
xpdb <- simpraz.xpdb

## Display label for dependent variable in the Xpose data object
xlabel("DV", xpdb)

## Set label for dependent variable
xlabel(xpdb) <- c("DV", "Concentration (mg/L)")
xlabel("DV", xpdb) # how has this chnaged?
```

xp.boot.par.est

Compare parameter estimates for covariate coefficients

Description

This function creates a plot of the estimates for covariate coefficients, obtained from the first step (univariate testing) in each scm performed in the bootscm. When normalized for their standard deviation, these plots can be used to compare the strength of the covariate relationship. Coloring is based on the covariate being included in the final model (blue) not being included (red).

Usage

```
xp.boot.par.est(
  bootgam.obj = NULL,
  sd.norm = TRUE,
  by.cov.type = FALSE,
  abs.values = FALSE,
  show.data = TRUE,
  show.means = TRUE,
  show.bias = TRUE,
  dotpch = c(1, 19),
  labels = NULL,
  pch.mean = "|",
  xlab = NULL,
  ylab = NULL,
  col = c(rgb(0.8, 0.5, 0.5), rgb(0.2, 0.2, 0.7), rgb(0.2, 0.2, 0.7), rgb(0.6, 0.6, 0.6)),
  ...
)
```

Arguments

bootgam.obj	The object created using bootscm.import(), which hold the data for plotting.
sd.norm	Perform normalization of the covariate coefficients (default is TRUE). When TRUE, the estimated covariate coefficients will be multiplied by the standard deviation of the specific covariate (both for continuous and categorical covariates).
by.cov.type	Split the plot for continuous and dichotomous covariates. Default is FALSE.
abs.values	Show the covariate coefficient in absolute values. Default is FALSE.
show.data	Show the actual covariate coefficients in the plot. Default is TRUE.
show.means	Show the means of included covariates (blue) and all covariates (grey) in the plot. Default is TRUE.
show.bias	Show estimated bias as text in the plot. Default is TRUE.
dotpch	The character used for plotting.
labels	Custom labels for the parameter-covariate relationships, (character vector)
pch.mean	The character used for plotting the mean.
xlab	Custom x-axis label
ylab	Custom y-axis label
col	The color scheme.
...	Additional plotting arguments may be passed to this function.

Details

Optionally, estimated bias is plotted in the graph (as text). Bias is also shown by the difference in mean of parameter estimates when the covariate is included (blue diamond), as opposed to the mean of all parameter estimates (grey diamond)

Note: For dichotomous covariates, the default PsN implementation is to use the most common covariate value as base, while the effect of the other value, is estimated by a theta. Xpose (bootscm.import) however recalculates the estimated parameters, to the parametrization in which the lowest value of the dichotomous covariate is the base (e.g. 0), and the estimated THETA denotes the proportional change, when the covariate has the other value (e.g. 1).

Value

No value returned.

Author(s)

Ron Keizer

Examples

```
xp.boot.par.est()
```

xp.boot.par.est.corr *Correlations between covariate coefficients*

Description

This function creates a plot showing the correlations in estimates for covariate coefficients, obtained from the first step (univariate testing) in each scm performed in the bootscm.

Usage

```
xp.boot.par.est.corr(
  bootgam.obj = NULL,
  sd.norm = TRUE,
  by.cov.type = FALSE,
  cov.plot = NULL,
  ask.covs = FALSE,
  dotpch = 19,
  col = rgb(0.2, 0.2, 0.9, 0.75),
  ...
)
```

Arguments

bootgam.obj	The object created using bootscm.import(), which hold the data for plotting.
sd.norm	Perform normalization of the covariate coefficients (default is TRUE). When TRUE, the estimated covariate coefficients will be multiplied by the standard deviation of the specific covariate (both for continuous and categorical covariates).
by.cov.type	Split the plot for continuous and dichotomous covariates. Default is FALSE.
cov.plot	A character vector which lists the covariates to include in the plot. If none are specified (NULL), all covariate coefficients will be included in the plot.
ask.covs	Ask the user which covariates to include in the plot. Default is FALSE.
dotpch	The character used for plotting.
col	The colors used for plotting.
...	Additional plotting arguments may be passed to this function.

Value

No value returned.

Author(s)

Ron Keizer

Examples

```
## Not run:
xp.boot.par.est.corr(current.bootscm, sd.norm = TRUE,
                     cov.plot = c("CLSEX", "VSEX", "CLWT"))

## End(Not run)
```

xp.daic.npar.plot *Distribution of difference in AIC*

Description

Distribution of difference in AIC

Usage

```
xp.daic.npar.plot(
  bootscm.obj = NULL,
  main = NULL,
  xlb = "Difference in AIC",
  ylb = "Density",
  ...
)
```

Arguments

bootscm.obj	a bootscm object.
main	The title of the plot
xlb	The x-label of the plot
ylb	The y-label of the plot
...	Additional parameters passed to panel.xyplot and xyplot.

Value

A lattice plot object.

See Also

Other bootgam: [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

Other bootscm: [bootscm.import\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

xp.distr.mod.size	<i>Plot of model size distribution for a bootgam or bootscm</i>
-------------------	---

Description

This function creates a kernel smoothed plot of the number of covariates included in the final model in each gam/scm in the bootgam/bootscm procedure.

Usage

```
xp.distr.mod.size(  
  bootgam.obj = NULL,  
  boot.type = NULL,  
  main = NULL,  
  bw = 0.5,  
  xlb = NULL,  
  ...  
)
```

Arguments

bootgam.obj	The bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	Plot title.
bw	The smoothing bandwidth to be used for the kernel.
xlb	The x-axis label.
...	Additional plotting parameter may be passed to this function.

Value

A lattice plot object will be returned.

Author(s)

Ron Keizer

xp.dofv.npar.plot	<i>Distribution of difference in OFV</i>
-------------------	--

Description

Distribution of difference in OFV

Usage

```
xp.dofv.npar.plot(  
  bootscm.obj = NULL,  
  main = NULL,  
  xlb = "Difference in OFV",  
  ylb = "Density",  
  ...  
)
```

Arguments

bootscm.obj	a bootscm object.
main	The title of the plot
xlb	The x-label of the plot
ylb	The y-label of the plot
...	Additional parameters passed to <code>panel.xyplot</code> and <code>xyplot</code> .

Value

A lattice plot object.

See Also

Other bootgam: [xp.daic.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

Other bootscm: [bootscm.import\(\)](#), [xp.daic.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

xp.dofv.plot *OFV difference (optimism) plot.*

Description

A plot of the difference in OFV between final bootscm models and the reference final scm model.

Usage

```
xp.dofv.plot(
  bootscm.obj = NULL,
  main = NULL,
  xlb = "Difference in OFV",
  ylb = "Density",
  ...
)
```

Arguments

bootscm.obj	The bootgam or bootscm object.
main	Plot title.
xlb	Label for x-axis.
ylb	Label for y-axis.
...	Additional plotting parameters.

Value

A lattice plot object is returned.

Author(s)

Ron Keizer

xp.get.disp *Default function for calculating dispersion in [xpose.gam](#).*

Description

Default function for calculating dispersion in [xpose.gam](#).

Usage

```
xp.get.disp(gamdata, parnam, covnams, family = "gaussian", ...)
```

Arguments

gamdata	the data used for a GAM
parnam	ONE (and only one) model parameter name.
covnam	Covariate names to test on parameter.
family	Assumption for the parameter distribution.
...	Used to pass arguments to more basic functions.

Value

a list including the dispersion

See Also

Other GAM functions: [GAM_summary_and_plot](#), [xp.scope3\(\)](#), [xpose.bootgam\(\)](#), [xpose.gam\(\)](#), [xpose4-package](#)

xp.inc.cond.stab.cov *Trace plots for conditional indices*

Description

Trace plots for conditional indices

Usage

```

xp.inc.cond.stab.cov(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  xlb = "Bootstrap replicate number",
  ylb = "Conditional inclusion frequency",
  normalize = TRUE,
  split.plots = FALSE,
  ...
)

```

Arguments

bootgam.obj	The bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	The title of the plot
xlb	The x-label of the plot
ylb	The y-label of the plot
normalize	Should one normalize?
split.plots	Should the plots be split?
...	Additional parameters passed to <code>panel.xyplot</code> and <code>xyplot</code> .

Value

A lattice plot object.

See Also

Other bootgam: `xp.daic.npar.plot()`, `xp.dofv.npar.plot()`, `xp.inc.ind.cond.stab.cov()`, `xp.inc.stab.cov()`, `xp.incl.index.cov()`, `xp.incl.index.cov.ind()`

Other bootscm: `bootscm.import()`, `xp.daic.npar.plot()`, `xp.dofv.npar.plot()`, `xp.inc.ind.cond.stab.cov()`, `xp.inc.stab.cov()`, `xp.incl.index.cov()`, `xp.incl.index.cov.ind()`

`xp.inc.ind.cond.stab.cov`

Trace plots for conditional indices rper replicate number

Description

Trace plots for conditional indices rper replicate number

Usage

```
xp.inc.ind.cond.stab.cov(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  xlb = "Bootstrap replicate number",
  ylb = "Conditional inclusion frequency",
  limits = c(0.2, 0.8),
  normalize = TRUE,
  split.plots = FALSE,
  start = 25,
  ...
)
```

Arguments

<code>bootgam.obj</code>	The bootgam or bootscm object.
<code>boot.type</code>	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
<code>main</code>	The title of the plot
<code>xlb</code>	The x-label of the plot
<code>ylb</code>	The y-label of the plot
<code>limits</code>	Limits for the inclusion index.
<code>normalize</code>	Should one normalize?
<code>split.plots</code>	Should the plots be split?
<code>start</code>	When to start.
<code>...</code>	Arguments passed to other functions.

Value

A lattice plot object.

See Also

Other bootgam: `xp.daic.npar.plot()`, `xp.dofv.npar.plot()`, `xp.inc.cond.stab.cov()`, `xp.inc.stab.cov()`, `xp.incl.index.cov()`, `xp.incl.index.cov.ind()`

Other bootscm: `bootscm.import()`, `xp.daic.npar.plot()`, `xp.dofv.npar.plot()`, `xp.inc.cond.stab.cov()`, `xp.inc.stab.cov()`, `xp.incl.index.cov()`, `xp.incl.index.cov.ind()`

<code>xp.inc.prob</code>	<i>Inclusion frequency plot</i>
--------------------------	---------------------------------

Description

Plot the inclusion frequencies of covariates in the final models obtained in a bootgam or bootscm. Covariates are ordered by inclusion frequency.

Usage

```
xp.inc.prob(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  col = "#6495ED",
  xlb = NULL,
  ylb = "Covariate",
  ...
)
```

Arguments

<code>bootgam.obj</code>	The bootgam or bootscm object.
<code>boot.type</code>	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
<code>main</code>	Plot title
<code>col</code>	Color used for the plot.
<code>xlb</code>	Label for x-axis.
<code>ylb</code>	Label for y-axis.
<code>...</code>	Additional plotting parameters.

Value

A lattice plot object will be returned.

Author(s)

Ron Keizer

xp.inc.prob.comb.2 *Inclusion frequency plot for combination of covariates.*

Description

Plot the inclusion frequency of the most common 2-covariate combinations.

Usage

```
xp.inc.prob.comb.2(  
  bootgam.obj = NULL,  
  boot.type = NULL,  
  main = NULL,  
  col = "#6495ED",  
  xlb = NULL,  
  ylb = "Covariate combination",  
  ...  
)
```

Arguments

bootgam.obj	The bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	Plot title
col	Color used for plot.
xlb	Label for x-axis.
ylb	Label for y-axis.
...	Additional plotting parameters.

Value

A lattice plot object will be returned.

Author(s)

Ron Keizer

xp.inc.stab.cov	<i>Inclusion stability plot A plot of the inclusion frequency of covariates vs bootgam/bootscm iteration number. This plot can be used to evaluate whether sufficient iterations have been performed.</i>
-----------------	---

Description

Inclusion stability plot

A plot of the inclusion frequency of covariates vs bootgam/bootscm iteration number. This plot can be used to evaluate whether sufficient iterations have been performed.

Usage

```

xp.inc.stab.cov(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  normalize = TRUE,
  split.plots = FALSE,
  xlb = "Bootstrap replicate number",
  ylb = "Difference of estimate with final",
  ...
)

```

Arguments

bootgam.obj	The bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	Plot title
normalize	Should the plot be normalized?
split.plots	Should the plots be split?
xlb	The label for the x-axis.
ylb	The label for the y-axis.
...	Additional plotting parameters

Value

A lattice plot object is returned.

Author(s)

Ron Keizer

See Also

Other bootgam: [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

Other bootscm: [bootscm.import\(\)](#), [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

xp.incl.index.cov *Plot of inclusion index of covariates.*

Description

Covariate inclusion indices show the correlation in inclusion of a covariate in the final model in a bootgam or bootscm.

Usage

```
xp.incl.index.cov(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  xlb = "Index",
  ylb = "Covariate",
  add.ci = FALSE,
  incl.range = NULL,
  return_plot = TRUE,
  results.tab = NULL,
  ...
)
```

Arguments

bootgam.obj	The bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	Plot title.
xlb	Label for the x-axis.
ylb	Label for the y-axis.
add.ci	Add a confidence interval to the plotted data.
incl.range	Included range
return_plot	Should the function return a plot?
results.tab	Specify your own results table.
...	Additional plotting information.

Value

A lattice plot object is returned.

Author(s)

Ron Keizer

See Also

Other bootgam: [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

Other bootscm: [bootscm.import\(\)](#), [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov.ind\(\)](#)

xp.incl.index.cov.comp

Inclusion index individuals, compare between covariates.

Description

A plot showing the range of inclusion indices for individuals for all covariates. This plot can be used to evaluate whether there were covariates which were more influenced by the constituency of the bootstrapped dataset than others.

Usage

```
xp.incl.index.cov.comp(
  bootgam.obj = NULL,
  boot.type = NULL,
  main = NULL,
  xlb = "Individual inclusion index",
  ylb = "ID",
  ...
)
```

Arguments

bootgam.obj	A bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
main	The title of the plot.
xlb	The label for the x-axis.
ylb	The label for the y-axis.
...	Additional plotting parameters.

Value

A lattice plot object is returned.

Author(s)

Ron Keizer

xp.incl.index.cov.ind *Individual inclusion index*

Description

This function will generate a plot of individual inclusion indexes for a specific covariate, which can be used to identify influential individuals for inclusion of that covariate. The index for an individual is calculated as the observed number of inclusions of that individual when the specific covariate was included minus the expected number of inclusions (based on the total bootstrap inclusions), divided by expected.

Usage

```
xp.incl.index.cov.ind(
  bootgam.obj = NULL,
  boot.type = NULL,
  cov.name = NULL,
  main = NULL,
  ylb = "ID",
  xlb = "Individual inclusion index",
  return_plot = TRUE,
  results.tab = NULL,
  ...
)
```

Arguments

bootgam.obj	A bootgam or bootscm object.
boot.type	Either "bootgam" or "bootscm". Default is NULL, which means the user will be asked to make a choice.
cov.name	The name of the covariate for which to create the plot.
main	The title of the plot.
ylb	The label for the x-axis.
xlb	The label for the y-axis.
return_plot	Should a plot object be returned?
results.tab	Supply your own results table.
...	Additional plotting parameters.

Value

A lattice plot object is returned.

Author(s)

Ron Keizer

See Also

Other bootgam: [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#)

Other bootscm: [bootscm.import\(\)](#), [xp.daic.npar.plot\(\)](#), [xp.dofv.npar.plot\(\)](#), [xp.inc.cond.stab.cov\(\)](#), [xp.inc.ind.cond.stab.cov\(\)](#), [xp.inc.stab.cov\(\)](#), [xp.incl.index.cov\(\)](#)

xp.scope3	<i>Define a scope for the gam. Used as default input to the scope argument in xpose.gam</i>
-----------	---

Description

Define a scope for the gam. Used as default input to the scope argument in xpose.gam

Usage

```

xp.scope3(
  object,
  covnam = xvardef("covariates", object),
  nmods = 3,
  smoother1 = 0,
  arg1 = NULL,
  smoother2 = 1,
  arg2 = NULL,
  smoother3 = "ns",
  arg3 = "df=2",
  smoother4 = "ns",
  arg4 = "df=3",
  excl1 = NULL,
  excl2 = NULL,
  excl3 = NULL,
  excl4 = NULL,
  extra = NULL,
  subset = xsubset(object),
  ...
)

```

Arguments

object	An xpose.data object.
covnam	Covariate names to test.
nmods	Number of models to examine.
smoother1	Smoother for each model.
arg1	Argument for model 1.
smoother2	Smoother for each model.
arg2	Argument for model 2.
smoother3	Smoother for each model.
arg3	Argument for model 3.
smoother4	Smoother for each model.
arg4	Argument for model 4.
excl1	Covariate exclusion from model 1.
excl2	Covariate exclusion from model 2.
excl3	Covariate exclusion from model 3.
excl4	Covariate exclusion from model 4.
extra	Extra exclusion criteria.
subset	Subset on data.
...	Used to pass arguments to more basic functions.

See Also

Other GAM functions: [GAM_summary_and_plot](#), [xp.get.disp\(\)](#), [xpose.bootgam\(\)](#), [xpose.gam\(\)](#), [xpose4-package](#)

Examples

```
xp.scope3(simpraz.xpdb)
```

xpose.bootgam

Title

Description

Title

Usage

```
xpose.bootgam(
  object,
  n = n,
  id = object@Prefs@Xvardef$id,
  oid = "OID",
  seed = NULL,
  parnam = xvardef("parms", object)[1],
  covnams = xvardef("covariates", object),
  conv.value = object@Prefs@Bootgam.prefs$conv.value,
  check.interval = as.numeric(object@Prefs@Bootgam.prefs$check.interval),
  start.check = as.numeric(object@Prefs@Bootgam.prefs$start.check),
  algo = object@Prefs@Bootgam.prefs$algo,
  start.mod = object@Prefs@Bootgam.prefs$start.mod,
  liif = as.numeric(object@Prefs@Bootgam.prefs$liif),
  ljif.conv = as.numeric(object@Prefs@Bootgam.prefs$ljif.conv),
  excluded.ids = as.numeric(object@Prefs@Bootgam.prefs$excluded.ids),
  ...
)
```

Arguments

object	An xpose.data object.
n	number of bootstrap iterations
id	column name of id
oid	create a new column with the original ID data
seed	random seed
parnam	ONE (and only one) model parameter name.
covnams	Covariate names to test on parameter.
conv.value	Convergence value
check.interval	How often to check the convergence
start.check	When to start checking
algo	Which algorithm to use
start.mod	which start model
liif	The liif value
ljif.conv	The convergence value for the liif
excluded.ids	ID values to exclude.
...	Used to pass arguments to more basic functions.

Value

a list of results from the bootstrap of the GAM.

See Also

Other GAM functions: [GAM_summary_and_plot](#), [xp.get.disp\(\)](#), [xp.scope3\(\)](#), [xpose.gam\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
## filter out occasion as a covariate as only one value
all_covs <- xvardef("covariates",simpraz.xpdb)
some_covs <- all_covs[!(all_covs %in% "OCC") ]

## here only running n=5 replicates to see that things work
## use something like n=100 for resonable results
boot_gam_obj <- xpose.bootgam(simpraz.xpdb,5,parnam="KA",covnams=some_covs,seed=1234)

## End(Not run)
```

xpose.data

Create an Xpose data object

Description

Creates an xpose.data object.

Usage

```
xpose.data(
  runno,
  tab.suffix = "",
  sim.suffix = "sim",
  cwres.suffix = "",
  directory = ".",
  quiet = TRUE,
  table.names = c("sdtab", "mutab", "patab", "catab", "cotab", "mytab", "extra", "xptab",
    "cwtab"),
  cwres.name = c("cwtab"),
  mod.prefix = "run",
  mod.suffix = ".mod",
  phi.suffix = ".phi",
  phi.file = NULL,
  nm7 = NULL,
  ...
)
```

Arguments

runno	Run number of the table files to read.
tab.suffix	Suffix to be appended to the table file names for the "real" data.
sim.suffix	Suffix to be appended to the table file names for any simulated data.
cwres.suffix	Suffix to be appended to the table file names for any CWRES data.
directory	Where the files are located.
quiet	A logical value indicating if more diagnostic messages should be printed when running this function.
table.names	Default text that Xpose looks for when searching for table files.
cwres.name	default text that xpose looks for when searching for CWRES table files.
mod.prefix	Start of model file name.
mod.suffix	End of model file name.
phi.suffix	End of .phi file name.
phi.file	The name of the .phi file. If not NULL then supersedes paste(mod.prefix,runno,phi.suffix,sep="").
nm7	T/F if table files are for NONMEM 7/6, NULL for undefined.
...	Extra arguments passed to function.

Details

Xpose expects, by default, to find at least one the the following NONMEM tables in the working directory to be able to create an Xpose data object (using a run number of '5' as an example):

sdtab5: The 'standard' parameters, including IWRE, IPRE, TIME, and the NONMEM default items (DV, PRED, RES and WRES) that are added when NOAPPEND is not present in the \$TABLE record.

```
$TABLE ID TIME IPRE IWRE NOPRINT ONEHEADER FILE=sdtab5
```

patab5: The empirical Bayes estimates of individual model parameter values, or posthoc estimates. These are model parameters, such as CL, V2, ETA1, etc.

```
$TABLE ID CL V2 KA K F1 ETA1 ETA2 ETA3 NOPRINT NOAPPEND ONEHEADER FILE=patab5
```

catab5: Categorical covariates, e.g. SEX, RACE.

```
$TABLE ID SEX HIV GRP NOPRINT NOAPPEND ONEHEADER FILE=catab5
```

cotab5: Continuous covariates, e.g. WT, AGE.

```
$TABLE ID WT AGE BSA HT GGT HB NOPRINT NOAPPEND ONEHEADER FILE=cotab5
```

mutab5, mytab5, extra5, xptab5: Additional variables of any kind. These might be useful if there are more covariates than can be accommodated in the covariates tables, for example, or if you have other variables that should be added, e.g. CMAX, AUC.

The default names for table files can be changed by changing the default values to the function. The files that Xpose looks for by default are:

```
paste(table.names, runno, tab.suffix, sep="")
```

The default CWRES table file name is called:

```
paste(cwres.name,runno,cwres.suffix,tab.suffix,sep="")
```

If there are simulation files present then Xpose looks for the files to be named:

```
paste(table.names, runno, sim.suffix, tab.suffix, sep="") paste(cwres.name, runno, sim.suffix, cwres.suff
```

This is basically a wrapper function for the `read.nm.tables`, `Data` and `SData` functions. See them for further information.

Also reads in the `.phi` file associated with the run (Individual OFVs, parameters, and variances of those parameters.)

Value

An `xpose.data` object. Default values for this object are created from a file called `'xpose.ini'`. This file can be found in the root directory of the `'xpose4'` package:

```
system.file("xpose.ini", package="xpose4").
```

It can be modified to fit the users wants and placed in the home folder of the user or the working directory, to override default settings.

Author(s)

Niclas Jonsson, Andrew Hooker

See Also

[xpose.data-class](#), [Data](#), [SData](#), [read.nm.tables](#), [compute.cwres](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data.extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.print\(\)](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
# Here we create files from an example NONMEM run

od = setwd(tempdir()) # move to a temp directory
(cur.files <- dir()) # current files in temp directory

simprazExample(overwrite=TRUE) # write files
(new.files <- dir()[!(dir() %in% cur.files)]) # what files are new here?

xpdb <- xpose.data(1)

file.remove(new.files) # remove these files
setwd(od) # restore working directory

## Not run:

# We expect to find the required NONMEM run and table files for run
```

```
# 5 in the current working directory, and that the table files have
# a suffix of '.dat', e.g. sdtab5.dat
xpdb5 <- xpose.data(5, tab.suffix = ".dat")

## End(Not run)
```

xpose.data-class *Class xpose.data*

Description

The xpose.data class is the fundamental data object in Xpose 4. It contains the data and preferences used in the creation of the Xpose plots and analyses.

Objects from the Class

Objects are most easily created by the xpose.data function, which reads the appropriate NON-MEM table files and populates the slots of the object.

Author(s)

Niclas Jonsson and Andrew Hooker

See Also

[xpose.data](#), [Data](#), [SData](#) [read.nm.tables](#), [xpose.prefs-class](#)

xpose.gam *Stepwise GAM search for covariates on a parameter (Xpose 4)*

Description

Function takes an Xpose object and performs a generalized additive model (GAM) stepwise search for influential covariates on a single model parameter.

Usage

```
xpose.gam(
  object,
  parnam = xvardef("parms", object)[1],
  covnams = xvardef("covariates", object),
  trace = TRUE,
  scope = NULL,
  disp = object@Prefs@Gam.prefs$disp,
  start.mod = object@Prefs@Gam.prefs$start.mod,
```

```

family = "gaussian",
wts.data = object@Data.firstonly,
wts.col = NULL,
steppit = object@Prefs@Gam.prefs$steppit,
subset = xsubset(object),
onlyfirst = object@Prefs@Gam.prefs$onlyfirst,
medianNorm = object@Prefs@Gam.prefs$medianNorm,
nmods = object@Prefs@Gam.prefs$nmods,
smoother1 = object@Prefs@Gam.prefs$smoother1,
smoother2 = object@Prefs@Gam.prefs$smoother2,
smoother3 = object@Prefs@Gam.prefs$smoother3,
smoother4 = object@Prefs@Gam.prefs$smoother4,
arg1 = object@Prefs@Gam.prefs$arg1,
arg2 = object@Prefs@Gam.prefs$arg2,
arg3 = object@Prefs@Gam.prefs$arg3,
arg4 = object@Prefs@Gam.prefs$arg4,
excl1 = object@Prefs@Gam.prefs$excl1,
excl2 = object@Prefs@Gam.prefs$excl2,
excl3 = object@Prefs@Gam.prefs$excl3,
excl4 = object@Prefs@Gam.prefs$excl4,
extra = object@Prefs@Gam.prefs$extra,
...
)

```

Arguments

object	An xpose.data object.
parnam	ONE (and only one) model parameter name.
covnam	Covariate names to test on parameter.
trace	TRUE if you want GAM output to screen.
scope	Scope of the GAM search.
disp	If dispersion should be used in the GAM object.
start.mod	Starting model.
family	Assumption for the parameter distribution.
wts.data	Weights on the least squares fitting of parameter vs. covariate. Often one can use the variances of the individual parameter values as weights. This data frame must have column with name ID and any subset variable as well as the variable defined by the wts.col.
wts.col	Which column in the wts.data to use.
steppit	TRUE for stepwise search, false for no search.
subset	Subset on data.
onlyfirst	TRUE if only the first row of each individual's data is to be used.
medianNorm	Normalize to the median of parameter and covariates.
nmods	Number of models to examine.

smoother1	Smoother for each model.
smoother2	Smoother for each model.
smoother3	Smoother for each model.
smoother4	Smoother for each model.
arg1	Argument for model 1.
arg2	Argument for model 2.
arg3	Argument for model 3.
arg4	Argument for model 4.
excl1	Covariate exclusion from model 1.
excl2	Covariate exclusion from model 2.
excl3	Covariate exclusion from model 3.
excl4	Covariate exclusion from model 4.
extra	Extra exclusion criteria.
...	Used to pass arguments to more basic functions.

Value

Returned is a [step.Gam](#) object. In this object the step-wise-selected model is returned, with up to two additional components. There is an "anova" component corresponding to the steps taken in the search, as well as a "keep" component if the "keep=" argument was supplied in the call.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[step.gam](#)

Other GAM functions: [GAM_summary_and_plot](#), [xp.get.disp\(\)](#), [xp.scope3\(\)](#), [xpose.bootgam\(\)](#), [xpose4-package](#)

Examples

```
## Run a GAM using the example xpose database
gam_ka <- xpose.gam(simpraz.xpdb, parnam="KA")

## Summarize GAM
xp.summary(gam_ka)

## GAM residuals of base model vs. covariates
xp.plot(gam_ka)

## An Akaike plot of the results
xp.akaike.plot(gam_ka)

## Studentized residuals
```

```
xp.ind.stud.res(gam_ka)

## Individual influence on GAM fit
xp.ind.inf.fit(gam_ka)

## Individual influence on GAM terms
xp.ind.inf.terms(gam_ka)

## Individual parameters to GAM fit
xp.cook(gam_ka)
```

```
xpose.license.citation
```

Displays the Xpose license and citation information

Description

This function displays a copy of Xpose's end user license agreement (EULA).

Usage

```
xpose.license.citation()
```

Value

The EULA.

Author(s)

Andrew Hooker

Examples

```
xpose.license.citation()
```

```
xpose.logTicks
```

Functions to create nice looking axes when using Log scales.

Description

The functions are used to create standard tic marks and axis labels when the axes are on the log scale.

Usage

```
xpose.logTicks(lim, loc = c(1, 5))  
  
xpose.yscale.components.log10(lim, ...)  
  
xpose.xscale.components.log10(lim, ...)
```

Arguments

lim	Limits
loc	Locations
...	Additional arguments passed to the function.

Details

These functions create log scales that look like they should (not the default R scales). These functions are used as input to the [xscale.components](#) argument in a lattice plot.

Functions

- `xpose.logTicks()`: Make log tic marks
- `xpose.xscale.components.log10()`: Make log scale on x-axis

Author(s)

Andrew Hooker

See Also

[xpose.plot.default](#) [xscale.components](#)

Examples

```
## Not run:  
xpdb5 <- xpose.data(5)  
xpose.plot.default("PRED", "DV", xpdb, logy=T, logx=T)  
xpose.plot.default("PRED", "DV", xpdb, logy=T, logx=T,  
                   yscale.components = xpose.yscale.components.log10,  
                   xscale.components = xpose.xscale.components.log10)  
  
## both give the same result  
  
## End(Not run)
```

xpose.multiple.plot *Create and object with class "xpose.multiple.plot".*

Description

Create and object with class "xpose.multiple.plot".

Usage

```
xpose.multiple.plot(
  plotList,
  plotTitle = NULL,
  nm7 = TRUE,
  prompt = FALSE,
  new.first.window = FALSE,
  max.plots.per.page = 4,
  title = list(title.x = unit(0.5, "npc"), title.y = unit(0.5, "npc"), title.gp =
    gpar(cex = 1.2, fontface = "bold"), title.just = c("center", "center")),
  mirror = FALSE,
  bql.layout = FALSE,
  ...
)
```

Arguments

plotList	A list of lattice plots.
plotTitle	Main title for plots.
nm7	TRUE if we are using NONMEM 7
prompt	When printing should we prompt for each new page in plot?
new.first.window	TRUE or FALSE.
max.plots.per.page	A number. Max value is 9.
title	Title properties.
mirror	Are there mirror plots in plot list?
bql.layout	Should we use layout optimized for plots with BQL (below limit of quantification) measurements?
...	Additional options passed to function.

Value

An object of class "xpose.multiple.plot".

Author(s)

Niclas Jonsson and Andrew C. Hooker

See Also

[print.xpose.multiple.plot](#), [xpose.multiple.plot.default](#)

Other generic functions: [gof\(\)](#), [xpose4-package](#)

xpose.multiple.plot-class

Class for creating multiple plots in xpose

Description

Class for creating multiple plots in xpose

Slots

`plotList` A list of lattice plots
`plotTitle` The plot title
`prompt` Should prompts be used
`new.first.window` Create a new first window?
`max.plots.per.page` How many plots per page?
`title` The title
`mirror` Are there mirror plots to create
`bql.layout` Should we use bql.layout

xpose.multiple.plot.default

Xpose 4 generic function for plotting multiple lattice objects on one page

Description

Function takes a list of **lattice** plot objects and prints them in a multiple plot layout with a title.

Usage

```
xpose.multiple.plot.default(
  plotList,
  plotTitle = NULL,
  prompt = FALSE,
  new.first.window = FALSE,
  max.plots.per.page = 4,
  title = list(title.x = unit(0.5, "npc"), title.y = unit(0.5, "npc"), title.gp =
    gpar(cex = 1.2, fontface = "bold"), title.just = c("center", "center")),
```

```

    mirror = FALSE,
    bql.layout = FALSE,
    page.numbers = TRUE,
    ...
)

```

Arguments

<code>plotList</code>	A list of lattice plot objects such that plot object <i>i</i> can be called with <code>plotList[[i]]</code>
<code>plotTitle</code>	The title used for the multiple plot layout
<code>prompt</code>	If more than one page is needed do you want a prompt at the command line before the next page is printed
<code>new.first.window</code>	Should the first page of this plot be in the already opened window or should a new window be created
<code>max.plots.per.page</code>	Maximum number of plots per page in the multiple layout
<code>title</code>	Look of title using grid .
<code>mirror</code>	if the list contains mirror plots
<code>bql.layout</code>	should we use layout optimized for BQL measurements?
<code>page.numbers</code>	Should we add page numbers to multiple page plots?
<code>...</code>	Other arguments passed to the code in this function

Details

Additional arguments:

title.x Where the title should be placed in the title **grid** region

title.y Where the title should be placed in the title **grid** region

title.just how the title should be justified

title.gp The par parameters for the title (see **grid**)

Value

returns nothing

Author(s)

Andrew Hooker

See Also

[grid](#), [basic.gof](#), [parm.vs.parm](#), [parm.vs.cov](#),

xpose.panel.bw	<i>Default box-and-whisker panel function for Xpose 4</i>
----------------	---

Description

This is the box-and-whisker panel function for Xpose 4. This is not intended to be used outside the `xpose.plot.bw` function. Most of the arguments take their default values from `xpose.data` object but this can be overridden by supplying them as arguments to `xpose.plot.bw`.

Usage

```
xpose.panel.bw(
  x,
  y,
  object,
  subscripts,
  groups = NULL,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  xvarnam = NULL,
  yvarnam = NULL,
  type = object@Prefs@Graph.prefs$type,
  col = object@Prefs@Graph.prefs$col,
  pch = object@Prefs@Graph.prefs$pch,
  cex = object@Prefs@Graph.prefs$cex,
  lty = object@Prefs@Graph.prefs$lty,
  fill = object@Prefs@Graph.prefs$col,
  ids = NULL,
  idsmode = object@Prefs@Graph.prefs$idsmode,
  idsext = object@Prefs@Graph.prefs$idsext,
  idsce = object@Prefs@Graph.prefs$idsce,
  idsdir = object@Prefs@Graph.prefs$idsdir,
  bwhoriz = object@Prefs@Graph.prefs$bwhoriz,
  bwratio = object@Prefs@Graph.prefs$bwratio,
  bwvarwid = object@Prefs@Graph.prefs$bwvarwid,
  bwdotpch = object@Prefs@Graph.prefs$bwdotpch,
  bwdotcol = object@Prefs@Graph.prefs$bwdotcol,
  bwdotcex = object@Prefs@Graph.prefs$bwdotcex,
  bwreccol = object@Prefs@Graph.prefs$bwreccol,
  bwrecfill = object@Prefs@Graph.prefs$bwrecfill,
  bwreclty = object@Prefs@Graph.prefs$bwreclty,
  bwreclwd = object@Prefs@Graph.prefs$bwreclwd,
  bwumbcol = object@Prefs@Graph.prefs$bwumbcol,
  bwumbly = object@Prefs@Graph.prefs$bwumbly,
  bwumbld = object@Prefs@Graph.prefs$bwumbld,
  bwoutcol = object@Prefs@Graph.prefs$bwoutcol,
```

```

bwoutcex = object@Prefs@Graph.prefs$bwoutcex,
bwoutpch = object@Prefs@Graph.prefs$bwoutpch,
grid = object@Prefs@Graph.prefs$grid,
logy = FALSE,
logx = FALSE,
force.x.continuous = TRUE,
binvar = NULL,
bins = 10,
...
)

```

Arguments

x	Name(s) of the x-variable.
y	Name(s) of the y-variable.
object	An xpose.data object.
subscripts	The standard Trellis subscripts argument (see xyplot).
groups	Name of the variable used for superpose plots.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot.
onlyfirst	Logical value indicating whether only the first row per individual is included in the plot.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.
xvarnam	Character string with the name of the x-variable.
yvarnam	Character string with the name of the y-variable.
type	Character value indicating the type of display to use: "l"=lines, "p"=points, "b"=both points and lines.
col	Colour of lines and plot symbols.
pch	Plot character to use.
cex	Size of the plot characters.
lty	Line type.
fill	Fill colour.
ids	Character value with the name of the variable to label data points with.
idsmode	Determines the way text labels are added to plots. NULL means that only extreme points are labelled. Non-NULL means all data points are labelled. (See link{xpose.plot.default})
idsext	See link{xpose.plot.bw}
idsces	Size of text labels.
idsdir	A value of "both" (the default) means that both high and low extreme points are labelled while "up" and "down" labels the high and low extreme points respectively. See xpose.plot.bw
bwhoriz	logical value indicating whether box and whiskers should be horizontal or not. The default is FALSE.

bwratio	Ratio of box height to inter-box space. The default is 1.5. An argument for panel.bwplot .
bwvarwid	Logical. If TRUE, widths of boxplots are proportional to the number of points used in creating it. The default is FALSE. An argument for panel.bwplot .
bwdotpch	Graphical parameter controlling the dot plotting character 'bwdotpch=""' is treated specially, by replacing the dot with a line. The default is 16. An argument for panel.bwplot .
bwdotcol	Graphical parameter controlling the dot colour - an integer or string. See 'col'. The default is black. An argument for panel.bwplot .
bwdotcex	The amount by which plotting text and symbols should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'. An argument for panel.bwplot .
bwreccol	The colour to use for the box rectangle - an integer or string. The default is blue. See trellis.par.get and "box.rectangle".
bwrecfill	The colour to use for filling the box rectangle - an integer or string. The default is transparent (none). See trellis.par.get and "box.rectangle".
bwreclty	The line type for the box rectangle - an integer or string. The default is solid. See trellis.par.get and "box.rectangle".
bwreclwd	The width of the lines for the box rectangle - an integer. The default is 1. See trellis.par.get and "box.rectangle".
bwumbcol	The colour to use for the umbrellas - an integer or string. The default is blue. See trellis.par.get and "box.umbrella".
bwumbly	The line type for the umbrellas - an integer or string. The default is solid. See trellis.par.get and "box.umbrella".
bwumbld	the width of the lines for the umbrellas - an integer. The default is 1. See trellis.par.get and "box.umbrella".
bwoutcol	The colour to use for the outliers - an integer or string. The default is blue. See trellis.par.get and "box.symbol".
bwoutcex	The amount by which outlier points should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'. The default is 0.8. See trellis.par.get and "box.symbol".
bwoutpch	The plotting character, or symbol, to use for outlier points. Specified as an integer. See R help on 'points'. The default is an open circle. See trellis.par.get and "box.symbol".
grid	logical value indicating whether a visual reference grid should be added to the graph. (Could use arguments for line type, color etc).
logy	Logical value indicating whether the y-axis should be logarithmic.
logx	Logical value indicating whether the x-axis should be logarithmic.
force.x.continuous	Logical value indicating whether x-values should be taken as continuous, even if categorical.
binvar	Variable to be used for binning.
bins	The number of bins to be used. The default is 10.
...	Other arguments that may be needed in the function.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.data-class](#), Cross-references above.

xpose.panel.default *Default panel function for Xpose 4*

Description

This is the panel function for Xpose 4. This is not intended to be used outside the `xpose.plot.default` function. Most of the arguments take their default values from `xpose.data` object but this can be overridden by supplying them as argument to `xpose.plot.default`.

Usage

```
xpose.panel.default(
  x,
  y,
  object,
  subscripts,
  groups = object@Prefs@Xvardef$id,
  grp.col = NULL,
  iplot = NULL,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  xvarnam = NULL,
  yvarnam = NULL,
  PI = NULL,
  PI.subset = NULL,
  PI.bin.table = NULL,
  PI.real = NULL,
  PI.mirror = NULL,
  PI.ci = NULL,
  PPI = NULL,
  PI.mean = FALSE,
  PI.delta.mean = FALSE,
  PI.x.median = TRUE,
  PI.rug = "Default",
  PI.rug.col = "orange",
  PI.rug.lwd = 3,
  PI.identify.outliers = TRUE,
  PI.outliers.col = "red",
  PI.outliers.pch = 8,
```

```
PI.outliers.cex = 1,  
PI.limits = c(0.025, 0.975),  
PI.arcol = "lightgreen",  
PI.up.lty = 2,  
PI.up.type = "l",  
PI.up.col = "black",  
PI.up.lwd = 2,  
PI.down.lty = 2,  
PI.down.type = "l",  
PI.down.col = "black",  
PI.down.lwd = 2,  
PI.med.lty = 1,  
PI.med.type = "l",  
PI.med.col = "black",  
PI.med.lwd = 2,  
PI.mean.lty = 3,  
PI.mean.type = "l",  
PI.mean.col = "black",  
PI.mean.lwd = 2,  
PI.delta.mean.lty = 3,  
PI.delta.mean.type = "l",  
PI.delta.mean.col = "black",  
PI.delta.mean.lwd = 2,  
PI.real.up.lty = 2,  
PI.real.up.type = "l",  
PI.real.up.col = "red",  
PI.real.up.lwd = 2,  
PI.real.down.lty = 2,  
PI.real.down.type = "l",  
PI.real.down.col = "red",  
PI.real.down.lwd = 2,  
PI.real.med.lty = 1,  
PI.real.med.type = "l",  
PI.real.med.col = "red",  
PI.real.med.lwd = 2,  
PI.real.mean.lty = 3,  
PI.real.mean.type = "l",  
PI.real.mean.col = "red",  
PI.real.mean.lwd = 2,  
PI.real.delta.mean.lty = 3,  
PI.real.delta.mean.type = "l",  
PI.real.delta.mean.col = "red",  
PI.real.delta.mean.lwd = 2,  
PI.mirror.up.lty = 2,  
PI.mirror.up.type = "l",  
PI.mirror.up.col = "darkgreen",  
PI.mirror.up.lwd = 1,  
PI.mirror.down.lty = 2,
```

```
PI.mirror.down.type = "1",
PI.mirror.down.col = "darkgreen",
PI.mirror.down.lwd = 1,
PI.mirror.med.lty = 1,
PI.mirror.med.type = "1",
PI.mirror.med.col = "darkgreen",
PI.mirror.med.lwd = 1,
PI.mirror.mean.lty = 3,
PI.mirror.mean.type = "1",
PI.mirror.mean.col = "darkgreen",
PI.mirror.mean.lwd = 1,
PI.mirror.delta.mean.lty = 3,
PI.mirror.delta.mean.type = "1",
PI.mirror.delta.mean.col = "darkgreen",
PI.mirror.delta.mean.lwd = 1,
PI.ci.up.arcol = "blue",
PI.ci.up.lty = 3,
PI.ci.up.type = "1",
PI.ci.up.col = "darkorange",
PI.ci.up.lwd = 2,
PI.ci.down.arcol = "blue",
PI.ci.down.lty = 3,
PI.ci.down.type = "1",
PI.ci.down.col = "darkorange",
PI.ci.down.lwd = 2,
PI.ci.med.arcol = "red",
PI.ci.med.lty = 4,
PI.ci.med.type = "1",
PI.ci.med.col = "darkorange",
PI.ci.med.lwd = 2,
PI.ci.mean.arcol = "purple",
PI.ci.mean.lty = 4,
PI.ci.mean.type = "1",
PI.ci.mean.col = "darkorange",
PI.ci.mean.lwd = 2,
PI.ci.delta.mean.arcol = "purple",
PI.ci.delta.mean.lty = 4,
PI.ci.delta.mean.type = "1",
PI.ci.delta.mean.col = "darkorange",
PI.ci.delta.mean.lwd = 2,
PI.ci.area.smooth = FALSE,
type = object@Prefs@Graph.prefs$type,
col = object@Prefs@Graph.prefs$col,
pch = object@Prefs@Graph.prefs$pch,
cex = object@Prefs@Graph.prefs$cex,
lty = object@Prefs@Graph.prefs$lty,
lwd = object@Prefs@Graph.prefs$lwd,
fill = object@Prefs@Graph.prefs$fill,
```

```
ids = NULL,  
idsmode = object@Prefs@Graph.prefs$idsmode,  
idsext = object@Prefs@Graph.prefs$idsext,  
idscecx = object@Prefs@Graph.prefs$idscecx,  
idsdir = object@Prefs@Graph.prefs$idsdir,  
abline = object@Prefs@Graph.prefs$abline,  
abllwd = object@Prefs@Graph.prefs$abllwd,  
ablly = object@Prefs@Graph.prefs$ablly,  
ablcol = object@Prefs@Graph.prefs$ablcol,  
smooth = object@Prefs@Graph.prefs$smooth,  
smlwd = object@Prefs@Graph.prefs$smlwd,  
smlty = object@Prefs@Graph.prefs$smlty,  
smcol = object@Prefs@Graph.prefs$smcol,  
smspan = object@Prefs@Graph.prefs$smspan,  
smdegr = object@Prefs@Graph.prefs$smdegr,  
smooth.for.groups = NULL,  
lmline = object@Prefs@Graph.prefs$lmline,  
lmlwd = object@Prefs@Graph.prefs$lmlwd,  
lmlty = object@Prefs@Graph.prefs$lmlty,  
lmcol = object@Prefs@Graph.prefs$lmcol,  
suline = object@Prefs@Graph.prefs$suline,  
sulwd = object@Prefs@Graph.prefs$sulwd,  
sulxy = object@Prefs@Graph.prefs$sulxy,  
sucol = object@Prefs@Graph.prefs$sucol,  
suspan = object@Prefs@Graph.prefs$suspan,  
sudegr = object@Prefs@Graph.prefs$sudegr,  
grid = object@Prefs@Graph.prefs$grid,  
logy = FALSE,  
logx = FALSE,  
force.x.continuous = FALSE,  
bwhoriz = object@Prefs@Graph.prefs$bwhoriz,  
bwratio = object@Prefs@Graph.prefs$bwratio,  
bwvarwid = object@Prefs@Graph.prefs$bwvarwid,  
bwdotpch = object@Prefs@Graph.prefs$bwdotpch,  
bwdotcol = object@Prefs@Graph.prefs$bwdotcol,  
bwdotcex = object@Prefs@Graph.prefs$bwdotcex,  
bwreccol = object@Prefs@Graph.prefs$bwreccol,  
bwrecfill = object@Prefs@Graph.prefs$bwrecfill,  
bwreclty = object@Prefs@Graph.prefs$bwreclty,  
bwreclwd = object@Prefs@Graph.prefs$bwreclwd,  
bwumbcol = object@Prefs@Graph.prefs$bwumbcol,  
bwumbly = object@Prefs@Graph.prefs$bwumbly,  
bwumbld = object@Prefs@Graph.prefs$bwumbld,  
bwoutcol = object@Prefs@Graph.prefs$bwoutcol,  
bwoutcex = object@Prefs@Graph.prefs$bwoutcex,  
bwoutpch = object@Prefs@Graph.prefs$bwoutpch,  
autocorr = FALSE,  
vline = NULL,
```

```

vllwd = 3,
vllty = 2,
vlcol = "grey",
hline = NULL,
hllwd = 3,
hllty = 1,
hlcol = "grey",
pch.ip.sp = pch,
cex.ip.sp = cex,
...
)

```

Arguments

x	Name(s) of the x-variable.
y	Name(s) of the y-variable.
object	An xpose.data object.
subscripts	The standard Trellis subscripts argument (see xyplot)
groups	Name of the variable used for superpose plots.
grp.col	Logical value indicating whether or not to use colour highlighting when groups are specified. NULL means no highlighting, while TRUE will identify group members by colour.
iplot	Is this an individual plots matrix? Internal use only.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot.
onlyfirst	Logical value indicating whether only the first row per individual is included in the plot.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.
xvarnam	Character string with the name of the x-variable.
yvarnam	Character string with the name of the y-variable.
PI	Either "lines", "area" or "both" specifying whether prediction intervals (as lines, as a shaded area or both) should be computed from the data in SData and added to the display. NULL means no prediction interval.
PI.subset	The subset to be used for the PI.
PI.bin.table	The table used to create VPC plots. Has a specific format created by read.npc.vpc.results
PI.real	Plot the percentiles of the real data in the various bins. values can be NULL or TRUE. Note that for a bin with few actual observations the percentiles will be approximate. For example, the 95th percentile of 4 data points will always be the largest of the 4 data points.
PI.mirror	Plot the percentiles of one simulated data set in each bin. values allowed are NULL, TRUE or AN.INTEGER.VALUE. TRUE takes the first mirror from PI.bin.table and AN.INTEGER.VALUE can be 1, 2, ... { } n where n is the number of mirror's output in the PI.bin.table. Used mainly by xpose.VPC .

PI.ci	Plot the prediction interval of the simulated data's percentiles for each bin. Values can be "both", "area" or "lines" This can be thought of as a prediction interval about the PI.real or a confidence interval about the PI. However, note that with increasing number of simulations the CI will not go towards zero because the interval is also dependent on the size of the data set.
PPI	The plot prediction interval. Has a specific format that must be followed. See setup.PPI .
PI.mean	Should the mean be plotted in the VPCs? TRUE or FALSE.
PI.delta.mean	Should the delta mean be plotted in the VPCs? TRUE or FALSE.
PI.x.median	Should the x-location of percentile lines in a bin be marked at the median of the x-values? (TRUE or FALSE)
PI.rug	Should there be markings on the plot showing where the binning intervals for the VPC are (or the locations of the independent variable used for each VPC calculation if binning is not used)?
PI.rug.col	Color of the PI.rug.
PI.rug.lwd	Linw width of the PI.rug.
PI.identify.outliers	Should outlying percentiles of the real data be highlighted? (TRUE of FALSE)
PI.outliers.col	Color of PI.identify.outliers points
PI.outliers.pch	pch of PI.identify.outliers points
PI.outliers.cex	cex of PI.identify.outliers points
PI.limits	A vector of two values that describe the limits of the prediction interval that should be displayed. For example $c(0.025, 0.975)$. These limits should be found in the 'PI.bin.table' table. These limits are also used as the percentages for the PI.real, PI.mirror and PI.ci. However, the confidence interval in PI.ci is always the one defined in the PI.bin.table.
PI.arcol	The color of the PI area
PI.up.lty	The upper line type. can be "dotted" or "dashed", etc.
PI.up.type	The upper type used for plotting. Defaults to a line.
PI.up.col	The upper line color
PI.up.lwd	The upper line width
PI.down.lty	The lower line type. can be "dotted" or "dashed", etc.
PI.down.type	The lower type used for plotting. Defaults to a line.
PI.down.col	The lower line color
PI.down.lwd	The lower line width
PI.med.lty	The median line type. can be "dotted" or "dashed", etc.
PI.med.type	The median type used for plotting. Defaults to a line.
PI.med.col	The median line color

PI.med.lwd	The median line width
PI.mean.lty	The mean line type. can be "dotted" or "dashed", etc.
PI.mean.type	The mean type used for plotting. Defaults to a line.
PI.mean.col	The mean line color
PI.mean.lwd	The mean line width
PI.delta.mean.lty	The delta.mean line type. can be "dotted" or "dashed", etc.
PI.delta.mean.type	The delta.mean type used for plotting. Defaults to a line.
PI.delta.mean.col	The delta.mean line color
PI.delta.mean.lwd	The delta.mean line width
PI.real.up.lty	The upper line type. can be "dotted" or "dashed", etc.
PI.real.up.type	The upper type used for plotting. Defaults to a line.
PI.real.up.col	The upper line color
PI.real.up.lwd	The upper line width
PI.real.down.lty	The lower line type. can be "dotted" or "dashed", etc.
PI.real.down.type	The lower type used for plotting. Defaults to a line.
PI.real.down.col	The lower line color
PI.real.down.lwd	The lower line width
PI.real.med.lty	The median line type. can be "dotted" or "dashed", etc.
PI.real.med.type	The median type used for plotting. Defaults to a line.
PI.real.med.col	The median line color
PI.real.med.lwd	The median line width
PI.real.mean.lty	The mean line type. can be "dotted" or "dashed", etc.
PI.real.mean.type	The mean type used for plotting. Defaults to a line.
PI.real.mean.col	The mean line color
PI.real.mean.lwd	The mean line width
PI.real.delta.mean.lty	The delta.mean line type. can be "dotted" or "dashed", etc.

PI.real.delta.mean.type
The delta.mean type used for plotting. Defaults to a line.

PI.real.delta.mean.col
The delta.mean line color

PI.real.delta.mean.lwd
The delta.mean line width

PI.mirror.up.lty
The upper line type. can be "dotted" or "dashed", etc.

PI.mirror.up.type
The upper type used for plotting. Defaults to a line.

PI.mirror.up.col
The upper line color

PI.mirror.up.lwd
The upper line width

PI.mirror.down.lty
The lower line type. can be "dotted" or "dashed", etc.

PI.mirror.down.type
The lower type used for plotting. Defaults to a line.

PI.mirror.down.col
The lower line color

PI.mirror.down.lwd
The lower line width

PI.mirror.med.lty
The median line type. can be "dotted" or "dashed", etc.

PI.mirror.med.type
The median type used for plotting. Defaults to a line.

PI.mirror.med.col
The median line color

PI.mirror.med.lwd
The median line width

PI.mirror.mean.lty
The mean line type. can be "dotted" or "dashed", etc.

PI.mirror.mean.type
The mean type used for plotting. Defaults to a line.

PI.mirror.mean.col
The mean line color

PI.mirror.mean.lwd
The mean line width

PI.mirror.delta.mean.lty
The delta.mean line type. can be "dotted" or "dashed", etc.

PI.mirror.delta.mean.type
The delta.mean type used for plotting. Defaults to a line.

PI.mirror.delta.mean.col
The delta.mean line color

PI.mirror.delta.mean.lwd
The delta.mean line width

PI.ci.up.arcol	The color of the upper PI.ci.
PI.ci.up.lty	The upper line type. can be "dotted" or "dashed", etc.
PI.ci.up.type	The upper type used for plotting. Defaults to a line.
PI.ci.up.col	The upper line color
PI.ci.up.lwd	The upper line width
PI.ci.down.arcol	The color of the lower PI.ci.
PI.ci.down.lty	The lower line type. can be "dotted" or "dashed", etc.
PI.ci.down.type	The lower type used for plotting. Defaults to a line.
PI.ci.down.col	The lower line color
PI.ci.down.lwd	The lower line width
PI.ci.med.arcol	The color of the median PI.ci.
PI.ci.med.lty	The median line type. can be "dotted" or "dashed", etc.
PI.ci.med.type	The median type used for plotting. Defaults to a line.
PI.ci.med.col	The median line color
PI.ci.med.lwd	The median line width
PI.ci.mean.arcol	The color of the mean PI.ci.
PI.ci.mean.lty	The mean line type. can be "dotted" or "dashed", etc.
PI.ci.mean.type	The mean type used for plotting. Defaults to a line.
PI.ci.mean.col	The mean line color
PI.ci.mean.lwd	The mean line width
PI.ci.delta.mean.arcol	The color of the delta.mean PI.ci.
PI.ci.delta.mean.lty	The delta.mean line type. can be "dotted" or "dashed", etc.
PI.ci.delta.mean.type	The delta.mean type used for plotting. Defaults to a line.
PI.ci.delta.mean.col	The delta.mean line color
PI.ci.delta.mean.lwd	The delta.mean line width
PI.ci.area.smooth	Should the "area" for PI.ci be smoothed to match the "lines" argument? Allowed values are TRUE/FALSE. The "area" is set by default to show the bins used in the PI.ci computation. By smoothing, information is lost and, in general, the confidence intervals will be smaller than they are in reality.
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.

col	The color for lines and points. Specified as an integer or a text string. A full list is obtained by the R command <code>colours()</code> . The default is blue (<code>col=4</code>).
pch	The plotting character, or symbol, to use. Specified as an integer. See R help on points . The default is an open circle.
cex	The amount by which plotting text and symbols should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'.
lty	The line type. Line types can either be specified as an integer (0=blank, 1=solid, 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., doesn't draw them).
lwd	the width for lines. Specified as an integer. The default is 1.
fill	fill for areas in plot
ids	Logical value specifying whether to label data points.
idsmode	Determines the way text labels are added to plots. NULL means that only extreme points are labelled. Non-NULL means all data points are labelled. (See <code>link{xpose.plot.default}</code>)
idsext	specifies the extent of the extremes to be used in labelling points. The default is 0.05 (only the most extreme 5% of points are labelled).
idsdex	the amount by which labels should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'.
idsdir	a string indicating the directions of the extremes to include in labelling. Possible values are "up", "down" and "both".
abline	Vector of arguments to the panel.abline function. No abline is drawn if NULL.
ablwd	Line width of any abline.
ablty	Line type of any abline.
abcol	Line colour of any abline.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
smlwd	Line width of the x-y smooth.
smlty	Line type of the x-y smooth.
smcol	Line color of the x-y smooth.
smspan	The smoothness parameter for the x-y smooth. The default is 0.667. An argument to panel.loess .
smdegr	The degree of the polynomials to be used for the x-y smooth, up to 2. The default is 1. An argument to panel.loess .
smooth.for.groups	Should a smooth for each group be drawn?
lmline	logical variable specifying whether a linear regression line should be superimposed over an xyplot . NULL ~ FALSE. ($y \sim x$)
lmlwd	Line width of the lmline.

lmlty	Line type of the lmline.
lmcol	Line colour of the lmline.
suline	A NULL value indicates that no superposed line should be added to the graph. If non-NULL then this should be the vector (the same length as y) of data points to be used for the smoothed superposed line.
sulwd	Line width of the superposed smooth.
sulty	Line type of the superposed smooth.
sucol	Line color of the superposed smooth.
suspan	The smoothness parameter. The default is 0.667. An argument to panel.loess .
sudegr	The degree of the polynomials to be used, up to 2. The default is 1. An argument to panel.loess .
grid	logical value indicating whether a visual reference grid should be added to the graph. (Could use arguments for line type, color etc).
logy	Logical value indicating whether the y-axis should be logarithmic.
logx	Logical value indicating whether the y-axis should be logarithmic.
force.x.continuous	Logical value indicating whether x-values should be taken as continuous, even if categorical.
bwhoriz	logical value indicating whether box and whiskers should be horizontal or not. The default is FALSE.
bwratio	Ratio of box height to inter-box space. The default is 1.5. An argument for panel.bwplot .
bwvarwid	Logical. If TRUE, widths of boxplots are proportional to the number of points used in creating it. The default is FALSE. An argument for panel.bwplot .
bwdotpch	Graphical parameter controlling the dot plotting character in boxplots. 'bwdotpch=" "' is treated specially, by replacing the dot with a line. The default is 16. An argument for panel.bwplot .
bwdotcol	Graphical parameter controlling the dot colour in boxplots - an integer or string. See 'col'. The default is black. An argument for panel.bwplot .
bwdotcex	The amount by which plotting text and symbols should be scaled relative to the default in boxplots. 'NULL' and 'NA' are equivalent to '1.0'. An argument for panel.bwplot .
bwreccol	The colour to use for the box rectangle in boxplots - an integer or string. The default is blue. See trellis.par.get and "box.rectangle".
bwrecfill	The colour to use for filling the box rectangle in boxplots - an integer or string. The default is transparent (none). See trellis.par.get and "box.rectangle".
bwrectly	The line type for the box rectangle in boxplots - an integer or string. The default is solid. See trellis.par.get and "box.rectangle".
bwreclwd	The width of the lines for the box rectangle in boxplots - an integer. The default is 1. See trellis.par.get and "box.rectangle".
bwumbcol	The colour to use for the umbrellas in boxplots - an integer or string. The default is blue. See trellis.par.get and "box.umbrella".

bwumbly	The line type for the umbrellas in boxplots - an integer or string. The default is solid. See trellis.par.get and "box.umbrella".
bwumbld	the width of the lines for the umbrellas in boxplots - an integer. The default is 1. See trellis.par.get and "box.umbrella".
bwoutcol	The colour to use for the outliers in boxplots - an integer or string. The default is blue. See trellis.par.get and "box.symbol".
bwoutcex	The amount by which outlier points should be scaled relative to the default in boxplots. 'NULL' and 'NA' are equivalent to '1.0'. The default is 0.8. See trellis.par.get and "box.symbol".
bwoutpch	The plotting character, or symbol, to use for outlier points in boxplots. Specified as an integer. See R help on 'points'. The default is an open circle. See trellis.par.get and "box.symbol".
autocorr	Is this an autocorrelation plot? Values can be TRUE/FALSE.
vline	Add a vertical line to the plot at the values specified.
vllwd	Width (lwd) of vertical line
vllty	Line type (lty) for vertical line
v1col	Color (col) of vertical line
hline	Add a horizontal line to the plot at the values specified.
hllwd	Width (lwd) of horizontal line
hllty	Line type (lty) for horizontal line
h1col	Color (col) of horizontal line
pch.ip.sp	If there is a panel with just one observation then this specifies the type of points for the DV, IPRED and PRED respectively.
cex.ip.sp	If there is a panel with just one observation then this specifies the size of the points for the DV, IPRED and PRED respectively.
...	Other arguments that may be needed in the function.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Justin Wilkins and Andrew Hooker

See Also

xpose.data-class, Cross-references above.

xpose.panel.histogram *Default histogram panel function for Xpose 4*

Description

This is the histogram panel function for Xpose 4. This is not intended to be used outside the `xpose.plot.histogram` function. Most of the arguments take their default values from `xpose.data` object but this can be overridden by supplying them as argument to `xpose.plot.histogram`.

Usage

```
xpose.panel.histogram(
  x,
  object,
  breaks = NULL,
  dens = TRUE,
  hidlty = object@Prefs@Graph.prefs$hidlty,
  hidcol = object@Prefs@Graph.prefs$hidcol,
  hidlwd = object@Prefs@Graph.prefs$hidlwd,
  hiborder = object@Prefs@Graph.prefs$hiborder,
  hilty = object@Prefs@Graph.prefs$hilty,
  hicol = object@Prefs@Graph.prefs$hicol,
  hilwd = object@Prefs@Graph.prefs$hilwd,
  math.dens = NULL,
  vline = NULL,
  vllwd = 3,
  vllty = 1,
  vlcol = "grey",
  hline = NULL,
  hllwd = 3,
  hllty = 1,
  hlcol = "grey",
  bins.per.panel.equal = TRUE,
  showMean = FALSE,
  meanllwd = 3,
  meanllty = 1,
  meanlcol = "orange",
  showMedian = FALSE,
  medianllwd = 3,
  medianllty = 1,
  medianlcol = "black",
  showPCTS = FALSE,
  PCTS = c(0.025, 0.975),
  PCTSllwd = 2,
  PCTSllty = hidlty,
  PCTSlcol = "black",
  vdline = NULL,
```

```

    vdllwd = 3,
    vdllty = 1,
    vdlcol = "red",
    ...,
    groups
)

```

Arguments

x	Name(s) of the x-variable.
object	An xpose.data object.
breaks	The breakpoints for the histogram.
dens	Density plot on top of histogram?
hidlty	Density line type.
hidcol	Color of density line.
hidlwd	Width of density line.
hiborder	Colour of the bar borders.
hilty	Line type for the bar borders.
hicol	Fill colour for the bars.
hilwd	Width for the bar borders.
math.dens	Should a density line be drawn. Values are NULL or TRUE.
vline	NULL or a vector of locations for the vertical lines to be drawn. For example, <code>vline=c(50,60)</code> will draw two vertical lines. The function <code>panel.abline</code> is used.
vllwd	Line width of the vertical lines defined with <code>vline</code> . Can be a vector or a single value, for example <code>vllwd=2</code> or <code>vllwd=c(2,3)</code> .
vllty	Line type of the vertical lines defined with <code>vline</code> . Can be a vector or a single value, for example <code>vllty=1</code> or <code>vllty=c(1,2)</code> .
vlcol	Line color of the vertical lines defined with <code>vline</code> . Can be a vector or a single value, for example <code>vlcol="red"</code> or <code>vllty=c("red","blue")</code> .
hline	NULL or a vector of locations for the horizontal lines to be drawn. For example, <code>hline=c(50,60)</code> will draw two horizontal lines. The function <code>panel.abline</code> is used.
hllwd	Line width of the horizontal lines defined with <code>hline</code> . Can be a vector or a single value, for example <code>hllwd=2</code> or <code>hllwd=c(2,3)</code> .
hllty	Line type of the horizontal lines defined with <code>hline</code> . Can be a vector or a single value, for example <code>hllty=1</code> or <code>hllty=c(1,2)</code> .
hlcol	Line color of the horizontal lines defined with <code>hline</code> . Can be a vector or a single value, for example <code>hlcol="red"</code> or <code>hllty=c("red","blue")</code> .
bins.per.panel.equal	Allow for different bins in different panels for continuous data? TRUE or FALSE.

showMean	Should the mean of the data in the histogram be shown?
meanllwd	Line width of mean line.
meanllty	The line type for the mean
meanlcol	Color for the mean line
showMedian	Should the median of the data for the histogram be shown as a vertical line?
medianllwd	line width of median line.
medianllty	line type of median line.
medianlcol	color of median line.
showPCTS	Should percentiles of the data for the histogram be shown?
PCTS	A vector of percentiles to show. Can be any length.
PCTSllwd	line width of percentiles. Can be a vector of same length as PCTS.
PCTSllty	Line type of the percentiles. Can be a vector of same length as PCTS.
PCTSlcol	Color of the percentiles. Can be a vector of same length as PCTS.
vdline	vertical line different for each histogram. Must be a vector.
vdllwd	line widths
vdllty	line types
vdicol	line colors
...	Other arguments that may be needed in the function.
groups	used to pass the conditioning variable into this function.

Author(s)

Andrew Hooker, Mats Karlsson, Justin Wilkins & E. Niclas Jonsson

See Also

xpose.data-class, Cross-references above.

xpose.panel.qq

Default QQ panel function for Xpose 4

Description

This is the QQ panel function for Xpose 4. This is not intended to be used outside the xpose.plot.qq function. Most of the arguments take their default values from xpose.data object but this can be overridden by supplying them as argument to xpose.plot.qq.

Usage

```
xpose.panel.qq(
  x,
  object,
  pch = object@Prefs@Graph.prefs$pch,
  col = object@Prefs@Graph.prefs$col,
  cex = object@Prefs@Graph.prefs$cex,
  abllty = object@Prefs@Graph.prefs$abllty,
  abllwd = object@Prefs@Graph.prefs$abllwd,
  ablcol = object@Prefs@Graph.prefs$ablcol,
  grid = object@Prefs@Graph.prefs$grid,
  ...
)
```

Arguments

x	Name(s) of the x-variable.
object	An xpose.data object.
pch	Plot character to use.
col	Colour of lines and plot symbols.
cex	Amount to scale the plotting character by.
abllty	Line type.
abllwd	Line width.
ablcol	Line colour.
grid	logical value indicating whether a visual reference grid should be added to the graph. (Could use arguments for line type, color etc).
...	Other arguments that may be needed in the function.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.qq](#), [qqmath](#), [panel.qqmathline](#), [xpose.data-class](#)

xpose.panel.splom

Scatterplot matrix panel function for Xpose 4

Description

This is the scatterplot matrix panel function for Xpose 4. This is not intended to be used outside the `xpose.plot.splom` function. Most of the arguments take their default values from `xpose.data` object but this can be overridden by supplying them as argument to `xpose.plot.splom`.

Usage

```
xpose.panel.splom(
  x,
  y,
  object,
  subscripts,
  onlyfirst = TRUE,
  inclZeroWRES = FALSE,
  type = "p",
  col = object@Prefs@Graph.prefs$col,
  pch = object@Prefs@Graph.prefs$pch,
  cex = object@Prefs@Graph.prefs$cex,
  lty = object@Prefs@Graph.prefs$lty,
  lwd = object@Prefs@Graph.prefs$lwd,
  smooth = TRUE,
  smlwd = object@Prefs@Graph.prefs$smlwd,
  smlty = object@Prefs@Graph.prefs$smlty,
  smcol = object@Prefs@Graph.prefs$smcol,
  smspan = object@Prefs@Graph.prefs$smspan,
  smdegr = object@Prefs@Graph.prefs$smdegr,
  lmline = NULL,
  lmlwd = object@Prefs@Graph.prefs$lmlwd,
  lmlty = object@Prefs@Graph.prefs$lmlty,
  lmcol = object@Prefs@Graph.prefs$lmcol,
  grid = object@Prefs@Graph.prefs$grid,
  groups = NULL,
  ...
)
```

Arguments

x	Name(s) of the x-variable.
y	Name(s) of the y-variable.
object	An xpose.data object.
subscripts	The standard Trellis subscripts argument (see xyplot)
onlyfirst	Logical value indicating whether only the first row per individual is included in the plot.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot.
type	1-character string giving the type of plot desired. The following values are possible, for details, see 'plot': "p" for points, "l" for lines, "o" for over-plotted points and lines, "b", "c") for (empty if "c") points joined by lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
col	The color for lines and points. Specified as an integer or a text string. A full list is obtained by the R command <code>colours()</code> . The default is blue (<code>col=4</code>).

pch	The plotting character, or symbol, to use. Specified as an integer. See R help on points . The default is an open circle.
cex	The amount by which plotting text and symbols should be scaled relative to the default. 'NULL' and 'NA' are equivalent to '1.0'.
lty	The line type. Line types can either be specified as an integer (0=blank, 1=solid, 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "blank", "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash", where "blank" uses 'invisible lines' (i.e., doesn't draw them).
lwd	the width for lines. Specified as an integer. The default is 1.
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
smlwd	Line width of the x-y smooth.
smlty	Line type of the x-y smooth.
smcol	Line color of the x-y smooth.
smspan	The smoothness parameter for the x-y smooth. The default is 0.667. An argument to panel.loess .
smdegr	The degree of the polynomials to be used for the x-y smooth, up to 2. The default is 1. An argument to panel.loess .
lmline	logical variable specifying whether a linear regression line should be superimposed over an xyplot . NULL ~ FALSE. (y~x)
lmlwd	Line width of the lmline.
lmlty	Line type of the lmline.
lmcol	Line colour of the lmline.
grid	logical value indicating whether a visual reference grid should be added to the graph. (Could use arguments for line type, color etc).
groups	Name of the variable used for superpose plots.
...	Other arguments that may be needed in the function.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.plot.splom](#), [xpose.data-class](#), [xyplot splom](#), [panel.splom](#), [panel.pairs](#)

Description

This is a wrapper function for the lattice `bwplot` function.

Usage

```
xpose.plot.bw(
  x,
  y,
  object,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  panel = xpose.panel.bw,
  groups = NULL,
  ids = FALSE,
  logy = FALSE,
  logx = FALSE,
  aspect = object@Prefs@Graph.prefs$aspect,
  funy = NULL,
  funx = NULL,
  PI = FALSE,
  by = object@Prefs@Graph.prefs$condvar,
  force.by.factor = FALSE,
  ordby = object@Prefs@Graph.prefs$ordby,
  byordfun = object@Prefs@Graph.prefs$byordfun,
  shingnum = object@Prefs@Graph.prefs$shingnum,
  shingol = object@Prefs@Graph.prefs$shingol,
  strip = function(...) strip.default(..., strip.names = c(TRUE, TRUE)),
  subset = xsubset(object),
  main = xpose.create.title(x, y, object, subset, funx, funy, ...),
  xlb = xpose.create.label(x, object, funx, logx, ...),
  ylb = xpose.create.label(y, object, funy, logy, ...),
  scales = list(),
  suline = object@Prefs@Graph.prefs$suline,
  binvar = NULL,
  bins = 10,
  mirror = FALSE,
  max.plots.per.page = 4,
  mirror.aspect = "fill",
  pass.plot.list = FALSE,
  x.cex = NULL,
  y.cex = NULL,
  main.cex = NULL,
```

```

    mirror.internal = list(strip.missing = missing(strip)),
    ...
)

```

Arguments

x	Name(s) of the x-variable.
y	Name(s) of the y-variable.
object	An xpose.data object.
inclZeroWRES	A logical value indicating whether rows with WRES=0 should be plotted.
onlyfirst	A logical value indicating whether only the first row per individual should be included in the plot.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.
panel	The name of the panel function to use. This should in most cases be left as xpose.panel.bw .
groups	A string with the name of any grouping variable (used as the groups argument to panel.xyplot).
ids	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the idlab Xpose data variable).
logy	Logical value indicating whether the y-axis should be logarithmic.
logx	Logical value indicating whether the x-axis should be logarithmic.
aspect	The aspect ratio of the display (see bwplot).
funy	String with the name of a function to apply to the y-variable before plotting, e.g. "abs".
funx	String with the name of a function to apply to the x-variable before plotting, e.g. "abs".
PI	Either "lines", "area" or "both" specifying whether prediction intervals (as lines, as a shaded area or both) should be computed from the data in SData and added to the display. NULL means no prediction interval.
by	A string or a vector of strings with the name(s) of the conditioning variables.
force.by.factor	Logical value. If TRUE, and by is not NULL, the variable specified by by is taken as categorical.
ordby	A string with the name of a variable to be used to reorder any factor conditioning variables (by). The variable is used in a call to the reorder function.
byordfun	The name of the function to be used when reordering a factor conditioning variable (see argument ordby).
shingnum	The number of shingles ("parts") a continuous conditioning variable should be divided into.
shingol	The amount of overlap between adjacent shingles (see argument shingnum)

strip	The name of the function to be used as the strip argument to the bwplot .
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
main	A string giving the plot title or NULL if none.
xlbl	A string giving the label for the x-axis. NULL if none.
ylbl	A string giving the label for the y-axis. NULL if none.
scales	A list to be used for the scales argument in bwplot .
suline	A string giving the variable to be used to construct a smooth to superpose on the display. NULL if none. This argument is used if you want to add a superpose line of a variable not present in the y list of variables.
binvar	Variable to be used for binning.
bins	The number of bins to be used. The default is 10.
mirror	Should we create mirror plots from simulation data? Value can be FALSE, TRUE or 1 for one mirror plot, or 3 for three mirror plots.
max.plots.per.page	The maximum number of plots per page that can be created with the mirror plots.
mirror.aspect	The aspect ratio of the plots used for mirror functionality.
pass.plot.list	Should we pass the list of plots created with mirror or should we print them directly. Values can be TRUE/FALSE.
x.cex	The size of the x-axis label.
y.cex	The size of the y-axis label.
main.cex	The size of the title.
mirror.internal	an internal mirror argument used in create.mirror . Checks if the strip argument from bwplot has been used.
...	Other arguments passed to xpose.panel.bw .

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.data-class](#), Cross-references above.

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## Box & whisker plot of WRES vs PRED
```

```
xpose.plot.bw("WRES", "PRED", xpdb5, binvar="PRED")

## End(Not run)
```

xpose.plot.default *The Xpose 4 generic functions for continuous y-variables.*

Description

This function is a wrapper for the lattice xyplot function.

Usage

```
xpose.plot.default(
  x,
  y,
  object,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  panel = xpose.panel.default,
  groups = object@Prefs@Xvardef$id,
  ids = object@Prefs@Graph.prefs$ids,
  logy = FALSE,
  logx = FALSE,
  yscale.components = "default",
  xscale.components = "default",
  aspect = object@Prefs@Graph.prefs$aspect,
  funx = NULL,
  funy = NULL,
  iplot = NULL,
  PI = NULL,
  by = object@Prefs@Graph.prefs$condvar,
  force.by.factor = FALSE,
  ordby = object@Prefs@Graph.prefs$ordby,
  byordfun = object@Prefs@Graph.prefs$byordfun,
  shingnum = object@Prefs@Graph.prefs$shingnum,
  shingol = object@Prefs@Graph.prefs$shingol,
  by.interval = NULL,
  strip = function(...) {
    strip.default(..., strip.names = c(TRUE, TRUE))
  },
  use.xpose.factor.strip.names = TRUE,
  subset = xsubset(object),
  autocorr = FALSE,
  main = xpose.create.title(x, y, object, subset, funx, funy, ...),
```

```

xlb = xpose.create.label(x, object, funx, logx, autocorr.x = autocorr, ...),
ylb = xpose.create.label(y, object, funy, logy, autocorr.y = autocorr, ...),
scales = list(),
suline = object@Prefs@Graph.prefs$suline,
bwhoriz = object@Prefs@Graph.prefs$bwhoriz,
dilution = FALSE,
dilfrac = object@Prefs@Graph.prefs$dilfrac,
diltype = object@Prefs@Graph.prefs$diltype,
dilci = object@Prefs@Graph.prefs$dilci,
seed = NULL,
mirror = FALSE,
max.plots.per.page = 4,
mirror.aspect = "fill",
pass.plot.list = FALSE,
x.cex = NULL,
y.cex = NULL,
main.cex = NULL,
mirror.internal = list(strip.missing = missing(strip)),
...
)

```

Arguments

<code>x</code>	A string or a vector of strings with the name(s) of the x-variable(s).
<code>y</code>	A string or a vector of strings with the name(s) of the y-variable(s).
<code>object</code>	An "xpose.data" object.
<code>inclZeroWRES</code>	A logical value indicating whether rows with WRES=0 should be plotted.
<code>onlyfirst</code>	A logical value indicating whether only the first row per individual should be included in the plot.
<code>samp</code>	An integer between 1 and <code>object@Nsim</code> (see xpose.data-class) specifying which of the simulated data sets to extract from <code>SData</code> .
<code>panel</code>	The name of the panel function to use.
<code>groups</code>	A string with the name of any grouping variable (used as the <code>groups</code> argument to <code>panel.xyplot</code>).
<code>ids</code>	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the <code>idlab</code> xpose data variable).
<code>logy</code>	Logical value indicating whether the y-axis should be logarithmic.
<code>logx</code>	Logical value indicating whether the x-axis should be logarithmic.
<code>yscale.components</code>	Used to change the way the axis look if <code>logy</code> is used. Can be a user defined function or <code>link{xpose.yscale.components.log10}</code> . If the axes are not log transformed then <code>yscale.components.default</code> is used.
<code>xscale.components</code>	Used to change the way the axis look if <code>logx</code> is used. Can be a user defined function or <code>link{xpose.xscale.components.log10}</code> . If the axes are not log transformed then <code>xscale.components.default</code> is used.

aspect	The aspect ratio of the display (see xyplot).
funx	String with the name of a function to apply to the x-variable before plotting, e.g. "abs".
funy	String with the name of a function to apply to the y-variable before plotting, e.g. "abs".
iplot	Is this an individual plots matrix? Internal use only.
PI	Either "lines", "area" or "both" specifying whether prediction intervals (as lines, as a shaded area or both) should be computed from the data in SData and added to the display. NULL means no prediction interval.
by	A string or a vector of strings with the name(s) of the conditioning variables.
force.by.factor	Logical value. If TRUE, and by is not NULL, the variable specified by by is taken as categorical.
ordby	A string with the name of a variable to be used to reorder any factor conditioning variables (by). The variable is used in a call to the <code>reorder.factor</code> function.
byordfun	The name of the function to be used when reordering a factor conditioning variable (see argument <code>ordby</code>)
shingnum	The number of shingles ("parts") a continuous conditioning variable should be divided into.
shingol	The amount of overlap between adjacent shingles (see argument <code>shingnum</code>)
by.interval	The intervals to use for conditioning on a continuous variable with <code>by</code> .
strip	The name of the function to be used as the <code>strip</code> argument to the xyplot . An easy way to change the strip appearance is to use strip.custom . For example, if you want to change the text in the strips you can use <code>strip=strip.custom(factor.levels=c("Hi", "There</code> if the <code>by</code> variable is a factor and <code>strip=strip.custom(var.name=c("New Name"))</code> if the <code>by</code> variable is continuous.
use.xpose.factor.strip.names	Use factor names in strips of conditioning plots..
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
autocorr	Is this an autocorrelation plot? Values can be TRUE/FALSE.
main	A string giving the plot title or NULL if none.
xlb	A string giving the label for the x-axis. NULL if none.
ylb	A string giving the label for the y-axis. NULL if none.
scales	A list to be used for the <code>scales</code> argument in xyplot .
suline	A string giving the variable to be used to construct a smooth to superpose on the display. NULL if none. This argument is used if you want to add a superpose line of a variable not present in the <code>y</code> list of variables.
bwhoriz	A logical value indicating if box and whiskers bars should be plotted horizontally or not. Used when the x-variable(s) is categorical.
dilution	Logical value indicating whether data dilution should be used.

<code>dilfrac</code>	Dilution fraction indicating the expected fraction of individuals to display in the plots. The exact meaning depends on the type of dilution (see below).
<code>diltype</code>	Indicating what type of dilution to apply. NULL means random dilution without stratification. A nonNULL value means stratified dilution.
<code>dilci</code>	A number between 0 and 1 giving the range eligible for dilution in a stratified dilution (see below).
<code>seed</code>	Seed number used for random dilution. NULL means no seed.
<code>mirror</code>	Should we create mirror plots from simulation data? Value can be FALSE, TRUE or 1 for one mirror plot, or 3 for three mirror plots.
<code>max.plots.per.page</code>	The maximum number of plots per page that can be created with the mirror plots.
<code>mirror.aspect</code>	The aspect ratio of the plots used for mirror functionality.
<code>pass.plot.list</code>	Should we pass the list of plots created with mirror or should we print them directly. Values can be TRUE/FALSE.
<code>x.cex</code>	The size of the x-axis label.
<code>y.cex</code>	The size of the y-axis label.
<code>main.cex</code>	The size of the title.
<code>mirror.internal</code>	an internal mirror argument used in <code>create.mirror</code> . Checks if the <code>strip</code> argument from <code>xyplot</code> has been used.
<code>...</code>	Other arguments passed to <code>xpose.panel.default</code> .

Details

`y` must be numeric (continuous) while `x` can be either numeric or factor. If `x` is numeric then a regular xy-plot is drawn. If `x` is a factor, on the other hand, a box and whiskers plot is constructed.

`x` and `y` can be either single valued strings or vector of strings. `x` and `y` can not both be vectors in the same call to the function.

If `ids` is TRUE, text labels are added to the plotting symbols. The labels are taken from the `idlab` `xpose` data variable. The way the text labels are plotted is governed by the `idsmode` argument (passed down to the `panel` function). `idsmode=NULL` (the default) means that only extreme data points are labelled while a non-NULL value adds labels to all data points (the default in `Xpose 3`). `xpose.panel.default` identifies extreme data points by fitting a loess smooth ($y \sim x$) and looking at the residuals from that fit. Points that are associated with the highest/lowest residuals are labelled. "High" and "low" are judged by the `panel` function parameter `idsext`, which gives the fraction of the total number of data points that are to be judged extreme in the "up" and "down" direction. The default value for `idsext` is 0.05 (see `xpose.prefs-class`). There is also a possibility to label only the high or low extreme points. This is done through the `idsdir` argument to `xpose.panel.default`. A value of "both" (the default) means that both high and low extreme points are labelled while "up" and "down" labels the high and low extreme points respectively.

Data dilution is useful in situations when there is an excessive amount of data. `xpose.plot.default` can dilute data in two different ways. The first is a completely random dilution in which all individuals are eligible for exclusion from the plot. In this case the argument `dilfrac` determines the

fraction of individuals that are excluded from the plot. The second type of dilution uses stratification to make sure that none of the extreme individuals are omitted from the plot. Extreme individuals are identified in a similar manner as extreme data points are identified for text labelling. A smooth is fitted to the data and the extreme residuals from that fit is used to inform about extremeness. What is judged as extreme is determined by the argument `dilci`, which defaults to 0.95 (Note that the meaning of this is the opposite to `idsext`). `dilci` give the confidence level of the interval around the fitted curve outside of which points are deemed to be extreme. Extreme individuals are those that have at least one point in the "extremeness" interval. Individuals that do not have any extreme points are eligible for dilution and `dilfrac` give the number of these that should be omitted from the graph. This means that `dilfrac` should usually be greater for stratified dilution than in completely random dilution. Any smooths added to a diluted plot is based on undiluted data.

More graphical parameters may be passed to [xpose.panel.default](#).

Value

Returns a xyplot graph object.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.panel.default](#), [xyplot](#), [panel.xyplot](#), [xpose.prefs-class](#), [xpose.data-class](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## A spaghetti plot of DV vs TIME
xpose.plot.default("TIME", "DV", xpdb5)

## A conditioning plot
xpose.plot.default("TIME", "DV", xpdb5, by = "SEX")

## Multiple x-variables
xpose.plot.default(c("WT", "SEX"), "CL", xpdb5)

## Multiple y-variables
xpose.plot.default("WT", c("CL", "V"), xpdb5)
xpose.plot.default("WT", c("CL", "V"), xpdb5, by=c("SEX", "HCTZ"))

## determining the interval for the conditioning variable
wt.ints <- matrix(c(50,60,60,70,70,80,80,90,90,100,100,150),nrow=6,ncol=2,byrow=T)
xpose.plot.default("TIME", "DV", xpdb5, by="WT", by.interval=wt.ints)

## End(Not run)
```

xpose.plot.histogram *The Xpose 4 generic functions for continuous y-variables.*

Description

This function is a wrapper for the lattice xyplot function.

Usage

```
xpose.plot.histogram(
  x,
  object,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  type = "density",
  aspect = object@Prefs@Graph.prefs$aspect,
  scales = list(),
  by = object@Prefs@Graph.prefs$condvar,
  force.by.factor = FALSE,
  ordby = object@Prefs@Graph.prefs$ordby,
  byordfun = object@Prefs@Graph.prefs$byordfun,
  shingnum = object@Prefs@Graph.prefs$shingnum,
  shingol = object@Prefs@Graph.prefs$shingol,
  strip = function(...) strip.default(..., strip.names = c(TRUE, TRUE)),
  subset = xsubset(object),
  main = xpose.create.title.hist(x, object, subset, ...),
  xlb = NULL,
  ylb = "Density",
  hicol = object@Prefs@Graph.prefs$hicol,
  hilty = object@Prefs@Graph.prefs$hilty,
  hilwd = object@Prefs@Graph.prefs$hilwd,
  hidcol = object@Prefs@Graph.prefs$hidcol,
  hidlty = object@Prefs@Graph.prefs$hidlty,
  hidlwd = object@Prefs@Graph.prefs$hidlwd,
  hiborder = object@Prefs@Graph.prefs$hiborder,
  mirror = FALSE,
  max.plots.per.page = 4,
  mirror.aspect = "fill",
  pass.plot.list = FALSE,
  x.cex = NULL,
  y.cex = NULL,
  main.cex = NULL,
  mirror.internal = list(strip.missing = missing(strip)),
```

```
    ...
  )
```

Arguments

x	A string or a vector of strings with the name(s) of the x-variable(s).
object	An "xpose.data" object.
inclZeroWRES	A logical value indicating whether rows with WRES=0 should be plotted.
onlyfirst	A logical value indicating whether only the first row per individual should be included in the plot.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.
type	The type of histogram to make. See histogram .
aspect	The aspect ratio of the display (see histogram).
scales	A list to be used for the scales argument in histogram .
by	A string or a vector of strings with the name(s) of the conditioning variables.
force.by.factor	Logical value. If TRUE, and by is not NULL, the variable specified by by is taken as categorical.
ordby	A string with the name of a variable to be used to reorder any factor conditioning variables (by). The variable is used in a call to the <code>reorder.factor</code> function.
byordfun	The name of the function to be used when reordering a factor conditioning variable (see argument <code>ordby</code>)
shingnum	The number of shingles ("parts") a continuous conditioning variable should be divided into.
shingol	The amount of overlap between adjacent shingles (see argument <code>shingnum</code>)
strip	The name of the function to be used as the strip argument to the xyplot .
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
main	A string giving the plot title or NULL if none.
xlb	A string giving the label for the x-axis. NULL if none.
ylb	A string giving the label for the y-axis. NULL if none.
hicol	the fill colour of the histogram - an integer or string. The default is blue (see histogram).
hilty	the border line type of the histogram - an integer. The default is 1 (see histogram).
hilwd	the border line width of the histogram - an integer. The default is 1 (see histogram).
hidcol	the fill colour of the density line - an integer or string. The default is black (see histogram).
hidlty	the border line type of the density line - an integer. The default is 1 (see histogram).
hidlwd	the border line width of the density line - an integer. The default is 1 (see histogram).

<code>hiborder</code>	the border colour of the histogram - an integer or string. The default is black (see histogram).
<code>mirror</code>	Should we create mirror plots from simulation data? Value can be FALSE, TRUE or 1 for one mirror plot, or 3 for three mirror plots.
<code>max.plots.per.page</code>	The maximum number of plots per page that can be created with the mirror plots.
<code>mirror.aspect</code>	The aspect ratio of the plots used for mirror functionality.
<code>pass.plot.list</code>	Should we pass the list of plots created with mirror or should we print them directly. Values can be TRUE/FALSE.
<code>x.cex</code>	The size of the x-axis label.
<code>y.cex</code>	The size of the y-axis label.
<code>main.cex</code>	The size of the title.
<code>mirror.internal</code>	an internal mirror argument used in create.mirror . Checks if the <code>strip</code> argument from xyplot has been used.
<code>...</code>	Other arguments passed to xpose.plot.histogram .

Details

`x` can be either numeric or factor, and can be either single valued strings or vectors of strings.

Value

Returns a histogram.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.panel.histogram](#), [histogram](#), [panel.histogram](#), [xpose.prefs-class](#), [xpose.data-class](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

xpose.plot.histogram("AGE", xpdb5, onlyfirst = TRUE)
xpose.plot.histogram(c("SEX", "AGE"), xpdb5, onlyfirst = TRUE)

## End(Not run)
```

Description

This is a wrapper function for the lattice [qqmath](#) function.

Usage

```
xpose.plot.qq(
  x,
  object,
  inclZeroWRES = FALSE,
  onlyfirst = FALSE,
  samp = NULL,
  aspect = object@Prefs@Graph.prefs$aspect,
  scales = list(),
  by = object@Prefs@Graph.prefs$condvar,
  force.by.factor = FALSE,
  ordby = object@Prefs@Graph.prefs$ordby,
  byordfun = object@Prefs@Graph.prefs$byordfun,
  shingnum = object@Prefs@Graph.prefs$shingnum,
  shingol = object@Prefs@Graph.prefs$shingol,
  strip = function(...) strip.default(..., strip.names = c(TRUE, TRUE)),
  subset = xsubset(object),
  main = xpose.create.title.hist(x, object, subset, ...),
  xlb = "Quantiles of Normal",
  ylb = paste("Quantiles of ", xlabel(x, object), sep = ""),
  pch = object@Prefs@Graph.prefs$pch,
  col = object@Prefs@Graph.prefs$col,
  cex = object@Prefs@Graph.prefs$cex,
  abllty = object@Prefs@Graph.prefs$abllty,
  abllwd = object@Prefs@Graph.prefs$abllwd,
  ablcol = object@Prefs@Graph.prefs$ablcol,
  mirror = FALSE,
  max.plots.per.page = 4,
  mirror.aspect = "fill",
  pass.plot.list = FALSE,
  x.cex = NULL,
  y.cex = NULL,
  main.cex = NULL,
  mirror.internal = list(strip.missing = missing(strip)),
  ...
)
```

Arguments

x A string or a vector of strings with the name(s) of the x-variable(s).

object	An "xpose.data" object.
inclZeroWRES	A logical value indicating whether rows with WRES=0 should be plotted.
onlyfirst	A logical value indicating whether only the first row per individual should be included in the plot.
samp	An integer between 1 and object@Nsim (see xpose.data-class) specifying which of the simulated data sets to extract from SData.
aspect	The aspect ratio of the display (see qqmath).
scales	A list to be used for the scales argument in qqmath .
by	A string or a vector of strings with the name(s) of the conditioning variables.
force.by.factor	Logical value. If TRUE, and by is not NULL, the variable specified by by is taken as categorical.
ordby	A string with the name of a variable to be used to reorder any factor conditioning variables (by). The variable is used in a call to the reorder function.
byordfun	The name of the function to be used when reordering a factor conditioning variable (see argument ordby).
shingnum	The number of shingles ("parts") a continuous conditioning variable should be divided into.
shingol	The amount of overlap between adjacent shingles (see argument shingnum).
strip	The name of the function to be used as the strip argument to the xyplot .
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
main	A string giving the plot title or NULL if none.
xlb	A string giving the label for the x-axis. NULL if none.
ylb	A string giving the label for the y-axis. NULL if none.
pch	Plotting symbol.
col	Color of plotting symbol.
cex	Amount to scale the plotting character by.
abl1ty	Line type for qqline.
abl1wd	Line width for qqline.
ablcol	Color for qqline.
mirror	Should we create mirror plots from simulation data? Value can be FALSE, TRUE or 1 for one mirror plot, or 3 for three mirror plots.
max.plots.per.page	The maximum number of plots per page that can be created with the mirror plots.
mirror.aspect	The aspect ratio of the plots used for mirror functionality.
pass.plot.list	Should we pass the list of plots created with mirror or should we print them directly. Values can be TRUE/FALSE.
x.cex	The size of the x-axis label.

`y.cex` The size of the y-axis label.
`main.cex` The size of the title.
`mirror.internal` an internal mirror argument used in `create.mirror`. Checks if the strip argument from `qqmath` has been used.
`...` Other arguments passed to `xpose.plot.qq`.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.panel.qq](#), [qqmath](#), [panel.qqmathline](#), [xpose.data-class](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## A QQ plot of WRES
xpose.plot.qq("WRES", xpdb5)

## End(Not run)
```

`xpose.plot.splom` *The Xpose 4 generic functions for scatterplot matrices.*

Description

This function is a wrapper for the lattice `splom` function.

Usage

```
xpose.plot.splom(
  plist,
  object,
  varnames = NULL,
  main = "Scatterplot Matrix",
  xlb = NULL,
  ylb = NULL,
  scales = list(),
  onlyfirst = TRUE,
```

```

inclZeroWRES = FALSE,
subset = xsubset(object),
by = object@Prefs@Graph.prefs$condvar,
force.by.factor = FALSE,
include.cat.vars = FALSE,
ordby = NULL,
byordfun = object@Prefs@Graph.prefs$byordfun,
shingnum = object@Prefs@Graph.prefs$shingnum,
shingol = object@Prefs@Graph.prefs$shingol,
strip = function(...) strip.default(..., strip.names = c(TRUE, TRUE)),
groups = NULL,
ids = object@Prefs@Graph.prefs$ids,
smooth = TRUE,
lmline = NULL,
panel = xpose.panel.splom,
aspect = object@Prefs@Graph.prefs$aspect,
samp = NULL,
max.plots.per.page = 4,
mirror = FALSE,
mirror.aspect = "fill",
pass.plot.list = FALSE,
x.cex = NULL,
y.cex = NULL,
main.cex = NULL,
mirror.internal = list(strip.missing = missing(strip)),
...
)

```

Arguments

<code>plist</code>	A vector of strings containing variable names for the scatterplot matrix.
<code>object</code>	An "xpose.data" object.
<code>varnames</code>	A vector of strings containing labels for the variables in the scatterplot matrix.
<code>main</code>	A string giving the plot title or NULL if none.
<code>xlb</code>	A string giving the label for the x-axis. NULL if none.
<code>ylb</code>	A string giving the label for the y-axis. NULL if none.
<code>scales</code>	A list to be used for the scales argument in <code>xyplot</code> .
<code>onlyfirst</code>	A logical value indicating whether only the first row per individual should be included in the plot.
<code>inclZeroWRES</code>	A logical value indicating whether rows with <code>WRES=0</code> should be plotted.
<code>subset</code>	A string giving the subset expression to be applied to the data before plotting. See xsubset .
<code>by</code>	A string or a vector of strings with the name(s) of the conditioning variables.
<code>force.by.factor</code>	Logical value. If TRUE, and <code>by</code> is not NULL, the variable specified by <code>by</code> is taken as categorical.

include.cat.vars	Logical value.
ordby	A string with the name of a variable to be used to reorder any factor conditioning variables (by). The variable is used in a call to the <code>reorder.factor</code> function.
byordfun	The name of the function to be used when reordering a factor conditioning variable (see argument <code>ordby</code>)
shingnum	The number of shingles ("parts") a continuous conditioning variable should be divided into.
shingol	The amount of overlap between adjacent shingles (see argument <code>shingnum</code>)
strip	The name of the function to be used as the <code>strip</code> argument to the <code>xyplot</code> .
groups	A string with the name of any grouping variable (used as the <code>groups</code> argument to <code>panel.xyplot</code>).
ids	A logical value indicating whether text labels should be used as plotting symbols (the variable used for these symbols indicated by the <code>idlab</code> xpose data variable).
smooth	A NULL value indicates that no superposed line should be added to the graph. If TRUE then a smooth of the data will be superimposed.
lmline	logical variable specifying whether a linear regression line should be superimposed over an <code>xyplot</code> . NULL ~ FALSE. ($y \sim x$)
panel	The name of the panel function to use.
aspect	The aspect ratio of the display (see <code>xyplot</code>).
samp	An integer between 1 and <code>object@Nsim</code> (see <code>xpose.data-class</code>) specifying which of the simulated data sets to extract from <code>SData</code> .
max.plots.per.page	The maximum number of plots per page that can be created with the mirror plots.
mirror	Should we create mirror plots from simulation data? Value can be FALSE, TRUE or 1 for one mirror plot, or 3 for three mirror plots.
mirror.aspect	The aspect ratio of the plots used for mirror functionality.
pass.plot.list	Should we pass the list of plots created with mirror or should we print them directly. Values can be TRUE/FALSE.
x.cex	The size of the x-axis label.
y.cex	The size of the y-axis label.
main.cex	The size of the title.
mirror.internal	an internal mirror argument used in <code>create.mirror</code> . Checks if the <code>strip</code> argument from <code>qqmath</code> has been used.
...	Other arguments passed to <code>xpose.panel.default</code> .

Details

If `ids` is TRUE, text labels are added to the plotting symbols. The labels are taken from the `idlab` xpose data variable. The way the text labels are plotted is governed by the `idsmode` argument (passed down to the panel function). `idsmode=NULL` (the default) means that only extreme data

points are labelled while a non-NULL value adds labels to all data points (the default in Xpose 3). `xpose.panel.default` identifies extreme data points by fitting a loess smooth ($y \sim x$) and looking at the residuals from that fit. Points that are associated with the highest/lowest residuals are labelled. "High" and "low" are judged by the panel function parameter `idsext`, which gives the fraction of the total number of data points that are to be judged extreme in the "up" and "down" direction. The default value for `idsext` is 0.05 (see [link{xpose.prefs-class}](#)). There is also a possibility to label only the high or low extreme points. This is done through the `idsdir` argument to `xpose.panel.default`. A value of "both" (the default) means that both high and low extreme points are labelled while "up" and "down" labels the high and low extreme points respectively.

More graphical parameters may be passed to `xpose.panel.splom`. for example, if you want to adjust the size of the varnames and axis tick labels you can use the parameters `varname.cex=0.5` and `axis.text.cex=0.5`.

Value

Returns a scatterplot matrix graph object.

Author(s)

E. Niclas Jonsson, Mats Karlsson, Andrew Hooker & Justin Wilkins

See Also

[xpose.panel.splom](#), [splom](#), [panel.splom](#), [xpose.prefs-class](#), [xpose.data-class](#)

Examples

```
## Not run:
## xpdb5 is an Xpose data object
## We expect to find the required NONMEM run and table files for run
## 5 in the current working directory
xpdb5 <- xpose.data(5)

## CL, WT, HT, SEX with a regression line
xpose.plot.splom(c("CL", "WT", "HT", "SEX"), xpdb5, lmline = TRUE)

## End(Not run)
```

xpose.prefs-class *Class "xpose.prefs"*

Description

An object of the "xpose.prefs" class holds information about all the variable and graphical preferences for a particular "xpose.data" object.

Objects from the Class

Objects can be created by calls of the form `new("xpose.prefs", ...)` but this is usually not necessary since the "xpose.prefs" object is created at the same time as the "xpose.data" object.

Author(s)

Niclas Jonsson & Andrew Hooker

See Also

[xvardef](#), [xlabel](#), [xsubset](#), [Data](#), [SData](#), [xpose.data](#), [read.nm.tables](#), [xpose.data-class](#), [xpose.gam](#)

xpose.print	<i>Summarize an xpose database</i>
-------------	------------------------------------

Description

Summarize an xpose database

Usage

```
xpose.print(object, long = TRUE)
```

Arguments

object	An xpose data object
long	long format or not.

Value

""

See Also

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose4-package](#), [xsubset\(\)](#)

Examples

```
xpose.print(simpraz.xpdb)
```

```
xpose.string.print      Print a pretty string.
```

Description

Print a string with a certain number of characters per row.

Usage

```
xpose.string.print(value, fill = 60, file = "")
```

Arguments

value	The text to print.
fill	How wide should the text be per row.
file	Where to print. "" means to the screen.

Author(s)

Niclas Jonsson and Andrew C. Hooker

```
xpose.VPC              Visual Predictive Check (VPC) using XPOSE
```

Description

This Function is used to create a VPC in xpose using the output from the vpc command in Pearl Speaks NONMEM (PsN). The function reads in the output files created by PsN and creates a plot from the data. The dependent variable, independent variable and conditioning variable are automatically determined from the PsN files.

Usage

```
xpose.VPC(
  vpc.info = "vpc_results.csv",
  vpctab = dir(pattern = "^vpctab")[1],
  object = NULL,
  ids = FALSE,
  type = "p",
  by = NULL,
  PI = NULL,
  PI.ci = "area",
  PI.ci.area.smooth = FALSE,
  PI.real = TRUE,
  subset = NULL,
```

```

    main = "Default",
    main.sub = NULL,
    main.sub.cex = 0.85,
    inclZeroWRES = FALSE,
    force.x.continuous = FALSE,
    funy = NULL,
    logy = FALSE,
    ylb = "Default",
    verbose = FALSE,
    PI.x.median = TRUE,
    PI.rug = "Default",
    PI.identify.outliers = TRUE,
    ...
)

```

Arguments

<code>vpc.info</code>	The results file from the <code>vpc</code> command in PsN. for example <code>'vpc_results.csv'</code> , or if the file is in a separate directory <code>'./vpc_dir1/vpc_results.csv'</code> .
<code>vpctab</code>	The <code>'vpctab'</code> from the <code>vpc</code> command in PsN. For example <code>'vpctab5'</code> , or if the file is in a separate directory <code>'./vpc_dir1/vpctab5'</code> . Can be <code>NULL</code> . The default looks in the current working directory and takes the first file that starts with <code>'vpctab'</code> that it finds. Note that this default can result in the wrong files being read if there are multiple <code>'vpctab'</code> files in the directory. One of <code>object</code> or <code>vpctab</code> is required. If both are present then the information from the <code>vpctab</code> will over-ride the <code>xpose</code> data object <code>object</code> (i.e. the values from the <code>vpctab</code> will replace any matching values in the <code>object@Data</code> portion of the <code>xpose</code> data object).
<code>object</code>	An <code>xpose</code> data object. Created from <code>xpose.data</code> . One of <code>object</code> or <code>vpctab</code> is required. If both are present then the information from the <code>vpctab</code> will over-ride the <code>xpose</code> data object <code>object</code> (i.e. the values from the <code>vpctab</code> will replace any matching values in the <code>object@Data</code> portion of the <code>xpose</code> data object).
<code>ids</code>	A logical value indicating whether text ID labels should be used as plotting symbols (the variable used for these symbols indicated by the <code>idlab</code> <code>xpose</code> data variable). Can be <code>FALSE</code> or <code>TRUE</code> .
<code>type</code>	Character string describing the way the points in the plot will be displayed. For more details, see plot . Use <code>type="n"</code> if you don't want observations in the plot.
<code>by</code>	A string or a vector of strings with the name(s) of the conditioning variables. For example <code>by = c("SEX", "WT")</code> . Because the function automatically determines the conditioning variable from the PsN input file specified in <code>vpc.info</code> , the <code>by</code> command can control if separate plots are created for each condition (<code>by=NULL</code>), or if a conditioning plot should be created (<code>by="WT"</code> for example). If the <code>vpc.info</code> file has a conditioning variable then <code>by</code> must match that variable. If there is no conditioning variable in <code>vpc.info</code> then the PI for each conditioned plot will be the PI for the entire data set (not only for the conditioning subset).
<code>PI</code>	Either <code>"lines"</code> , <code>"area"</code> or <code>"both"</code> specifying whether prediction intervals (as lines, a shaded area or both) should be added to the plot. <code>NULL</code> means no prediction interval.

PI.ci	Plot the confidence interval for the simulated data's percentiles for each bin (for each simulated data set compute the percentiles for each bin, then, from all of the percentiles from all of the simulated datasets compute the 95% CI of these percentiles). Values can be "both", "area" or "lines". These CIs can be used to asses the PI.real values for model misspecification. Note that with few observations per bin the CIs will be approximate because the percentiles in each bin will be approximate. For example, the 95th percentile of 4 data points will always be the largest of the 4 data points.
PI.ci.area.smooth	Should the "area" for PI.ci be smoothed to match the "lines" argument? Allowed values are TRUE/FALSE. The "area" is set by default to show the bins used in the PI.ci computation. By smoothing, information is lost and, in general, the confidence intervals will be smaller than they are in reality.
PI.real	Plot the percentiles of the real data in the various bins. values can be NULL or TRUE. Note that for a bin with few actual observations the percentiles will be approximate. For example, the 95th percentile of 4 data points will always be the largest of the 4 data points.
subset	A string giving the subset expression to be applied to the data before plotting. See xsubset .
main	A string giving the plot title or NULL if none. "Default" creates a default title.
main.sub	Used for names above each plot when using multiple plots. Should be a vector c("Group 1", "Group 2")
main.sub.cex	The size of the main.sub titles.
inclZeroWRES	Logical value indicating whether rows with WRES=0 is included in the plot.
force.x.continuous	Logical value indicating whether x-values should be converted to continuous variables, even if they are defined as factors.
funy	String of function to apply to Y data. For example "abs"
logy	Logical value indicating whether the y-axis should be logarithmic, base 10.
ylb	Label for the y-axis
verbose	Should warning messages and other diagnostic information be passed to screen? (TRUE or FALSE)
PI.x.median	Should the x-location of percentile lines in a bin be marked at the median of the x-values? (TRUE or FALSE)
PI.rug	Should there be markings on the plot showing where the binning intervals for the VPC are (or the locations of the independent variable used for each VPC calculation if binning is not used)?
PI.identify.outliers	Should outlying percentiles of the real data be highlighted? (TRUE of FALSE)
...	Other arguments passed to xpose.panel.default , xpose.plot.default and others. Please see these functions for more descriptions of what you can do.

Value

A plot or a list of plots.

Additional arguments

Below are some of the additional arguments that can control the look and feel of the VPC. See [xpose.panel.default](#) for all potential options.

Additional graphical elements available in the VPC plots.

PI.mirror = NULL, TRUE or AN.INTEGER.VALUE Plot the percentiles of one simulated data set in each bin. TRUE takes the first mirror from 'vpc_results.csv' and AN.INTEGER.VALUE can be 1, 2, ... { } n where n is the number of mirror's output in the 'vpc_results.csv' file.

PI.limits = c(0.025, 0.975) A vector of two values that describe the limits of the prediction interval that should be displayed. These limits should be found in the 'vpc_results.csv' file. These limits are also used as the percentages for the PI.real, PI.mirror and PI.ci. However, the confidence interval in PI.ci is always the one defined in the 'vpc_results.csv' file.

Additional options to control the look and feel of the PI. See [grid.polygon](#) and [plot](#) for more details.

PI.arcol The color of the PI area

PI.up.lty The upper line type. can be "dotted" or "dashed", etc.

PI.up.type The upper type used for plotting. Defaults to a line.

PI.up.col The upper line color

PI.up.lwd The upper line width

PI.down.lty The lower line type. can be "dotted" or "dashed", etc.

PI.down.type The lower type used for plotting. Defaults to a line.

PI.down.col The lower line color

PI.down.lwd The lower line width

PI.med.lty The median line type. can be "dotted" or "dashed", etc.

PI.med.type The median type used for plotting. Defaults to a line.

PI.med.col The median line color

PI.med.lwd The median line width

Additional options to control the look and feel of the PI.ci. See [grid.polygon](#) and [plot](#) for more details.

PI.ci.up.arcol The color of the upper PI.ci.

PI.ci.med.arcol The color of the median PI.ci.

PI.ci.down.arcol The color of the lower PI.ci.

PI.ci.up.lty The upper line type. can be "dotted" or "dashed", etc.

PI.ci.up.type The upper type used for plotting. Defaults to a line.

PI.ci.up.col The upper line color

PI.ci.up.lwd The upper line width

PI.ci.down.lty The lower line type. can be "dotted" or "dashed", etc.
PI.ci.down.type The lower type used for plotting. Defaults to a line.
PI.ci.down.col The lower line color
PI.ci.down.lwd The lower line width
PI.ci.med.lty The median line type. can be "dotted" or "dashed", etc.
PI.ci.med.type The median type used for plotting. Defaults to a line.
PI.ci.med.col The median line color
PI.ci.med.lwd The median line width
PI.ci.area.smooth Should the "area" for `PI.ci` be smoothed to match the "lines" argument? Allowed values are TRUE/FALSE. The "area" is set by default to show the bins used in the `PI.ci` computation. By smoothing, information is lost and, in general, the confidence intervals will be smaller than they are in reality.

Additional options to control the look and feel of the `PI.real`. See See [grid.polygon](#) and [plot](#) for more details.

PI.real.up.lty The upper line type. can be "dotted" or "dashed", etc.
PI.real.up.type The upper type used for plotting. Defaults to a line.
PI.real.up.col The upper line color
PI.real.up.lwd The upper line width
PI.real.down.lty The lower line type. can be "dotted" or "dashed", etc.
PI.real.down.type The lower type used for plotting. Defaults to a line.
PI.real.down.col The lower line color
PI.real.down.lwd The lower line width
PI.real.med.lty The median line type. can be "dotted" or "dashed", etc.
PI.real.med.type The median type used for plotting. Defaults to a line.
PI.real.med.col The median line color
PI.real.med.lwd The median line width

Additional options to control the look and feel of the `PI.mirror`. See See [plot](#) for more details.

PI.mirror.up.lty The upper line type. can be "dotted" or "dashed", etc.
PI.mirror.up.type The upper type used for plotting. Defaults to a line.
PI.mirror.up.col The upper line color
PI.mirror.up.lwd The upper line width
PI.mirror.down.lty The lower line type. can be "dotted" or "dashed", etc.
PI.mirror.down.type The lower type used for plotting. Defaults to a line.
PI.mirror.down.col The lower line color
PI.mirror.down.lwd The lower line width
PI.mirror.med.lty The median line type. can be "dotted" or "dashed", etc.
PI.mirror.med.type The median type used for plotting. Defaults to a line.
PI.mirror.med.col The median line color
PI.mirror.med.lwd The median line width

Author(s)

Andrew Hooker

See Also

[read.vpctab](#) [read.npc.vpc.results](#) [xpose.panel.default](#) [xpose.plot.default](#)

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC.both\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
library(xpose4)

xpose.VPC()

## to be more clear about which files should be read in
vpc.file <- "vpc_results.csv"
vpctab <- "vpctab5"
xpose.VPC(vpc.info=vpc.file, vpctab=vpctab)

## with lines and a shaded area for the prediction intervals
xpose.VPC(vpc.file, vpctab=vpctab, PI="both")

## with the percentages of the real data
xpose.VPC(vpc.file, vpctab=vpctab, PI.real=T)

## with mirrors (if supplied in 'vpc.file')
xpose.VPC(vpc.file, vpctab=vpctab, PI.real=T, PI.mirror=5)

## with CIs
xpose.VPC(vpc.file, vpctab=vpctab, PI.real=T, PI.ci="area")
xpose.VPC(vpc.file, vpctab=vpctab, PI.real=T, PI.ci="area", PI=NULL)

## stratification (if 'vpc.file' is stratified)
```

```

cond.var <- "WT"
xpose.VPC(vpc.file, vpctab=vpctab)
xpose.VPC(vpc.file, vpctab=vpctab, by=cond.var)
xpose.VPC(vpctab=vpctab, vpc.info=vpc.file, PI="both", by=cond.var, type="n")

## with no data points in the plot
xpose.VPC(vpc.file, vpctab=vpctab, by=cond.var, PI.real=T, PI.ci="area", PI=NULL, type="n")

## with different DV and IDV, just read in new files and plot
vpc.file <- "vpc_results.csv"
vpctab <- "vpctab5"
cond.var <- "WT"
xpose.VPC(vpctab=vpctab, vpc.info=vpc.file, PI="both", by=cond.var)
xpose.VPC(vpctab=vpctab, vpc.info=vpc.file, PI="both")

## to use an xpose data object instead of vpctab
##
## In this example
## we expect to find the required NONMEM run and table files for run
## 5 in the current working directory
runnumber <- 5
xpdb <- xpose.data(runnumber)
xpose.VPC(vpc.file, object=xpdb)

## to read files in a directory different than the current working directory
vpc.file <- "../vpc_strat_WT_4_mirror_5/vpc_results.csv"
vpctab <- "../vpc_strat_WT_4_mirror_5/vpctab5"
xpose.VPC(vpc.info=vpc.file, vpctab=vpctab)

## to rearrange order of factors in VPC plot
xpdb@Data$SEX <- factor(xpdb@Data$SEX, levels=c("2", "1"))
xpose.VPC(by="SEX", object=xpdb)

## End(Not run)

```

xpose.VPC.both	<i>Xpose Visual Predictive Check (VPC) for both continuous and Limit of Quantification data.</i>
----------------	--

Description

Xpose Visual Predictive Check (VPC) for both continuous and Below or Above Limit of Quantification (BLQ or ALQ) data.

Usage

```
xpose.VPC.both(
```

```

vpc.info = "vpc_results.csv",
vpctab = dir(pattern = "^vpctab")[1],
object = NULL,
subset = NULL,
main = "Default",
main.sub = NULL,
inclZeroWRES = FALSE,
cont.logy = F,
hline = "default",
add.args.cont = list(),
add.args.cat = list(),
...
)

```

Arguments

vpc.info	Name of PSN file to use. File will come from VPC command in PsN.
vpctab	Name of vpctab file produced from PsN.
object	Xpose data object.
subset	Subset of data to look at.
main	Title for plot.
main.sub	Used for names above each plot when using multiple plots. Should be a vector, e.g. c("title 1", "title 2").
inclZeroWRES	Include WRES=0 rows in the computations for these plots?
cont.logy	Should the continuous plot y-axis be on the log scale?
hline	Horizontal line marking the limits of quantification. If they are defined, they must be a vector of values.
add.args.cont	Additional arguments to the continuous plot. xpose.VPC .
add.args.cat	Additional arguments to the categorical plot. xpose.VPC.categorical .
...	Additional arguments to both plots.

Author(s)

Andrew C. Hooker

See Also

[xpose.VPC](#), [xpose.VPC.categorical](#).

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.categorical\(\)](#), [xpose4-package](#)

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_cov\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#),

```

cat.dv.vs.idv.sb(), cat.pc(), cov.splom(), cwres.dist.hist(), cwres.dist.qq(), cwres.vs.cov(),
cwres.vs.idv(), cwres.vs.idv.bw(), cwres.vs.pred(), cwres.vs.pred.bw(), cwres.wres.vs.idv(),
cwres.wres.vs.pred(), dOFV.vs.cov(), dOFV.vs.id(), dOFV1.vs.dOFV2(), data.checkout(),
dv.preds.vs.idv(), dv.vs.idv(), dv.vs.ipred(), dv.vs.ipred.by.cov(), dv.vs.ipred.by.idv(),
dv.vs.pred(), dv.vs.pred.by.cov(), dv.vs.pred.by.idv(), dv.vs.pred.ipred(), gof(),
ind.plots(), ind.plots.cwres.hist(), ind.plots.cwres.qq(), ipred.vs.idv(), iwres.dist.hist(),
iwres.dist.qq(), iwres.vs.idv(), kaplan.plot(), par_cov_hist, par_cov_qq, parm.vs.cov(),
parm.vs.parm(), pred.vs.idv(), ranpar.vs.cov(), runsum(), wres.dist.hist(), wres.dist.qq(),
wres.vs.idv(), wres.vs.idv.bw(), wres.vs.pred(), wres.vs.pred.bw(), xpose.VPC(), xpose.VPC.categorical(),
xpose4-package

```

Examples

```

## Not run:
library(xpose4)

## move to the directory where results from PsN
## are found
cur.dir <- getwd()
setwd(paste(cur.dir, "/vpc_cont_LL0Q/", sep=""))

xpose.VPC()
xpose.VPC.categorical(censored=T)

xpose.VPC.both()

xpose.VPC.both(subset="DV>1.75")

xpose.VPC.both(add.args.cont=list(ylim=c(0,80)))

xpose.VPC.both(add.args.cont = list(ylim = c(0.01, 80)), xlim = c(0,
  40), add.args.cat = list(ylim = c(0, 0.4)), cont.logy = T)

xpose.VPC.both(cont.logy=T)

## End(Not run)

```

xpose.VPC.categorical *Xpose visual predictive check for categorical data.*

Description

Xpose visual predictive check for categorical data (binary, ordered categorical and count data).

Usage

```

xpose.VPC.categorical(
  vpc.info = "vpc_results.csv",

```

```

vpctab = dir(pattern = "^vpctab")[1],
object = NULL,
subset = NULL,
main = "Default",
main.sub = "Default",
main.sub.cex = 0.85,
real.col = 4,
real.lty = "b",
real.cex = 1,
real.lwd = 1,
median.line = FALSE,
median.col = "darkgrey",
median.lty = 1,
ci.lines = FALSE,
ci.col = "blue",
ci.lines.col = "darkblue",
ci.lines.lty = 3,
xlb = "Default",
ylb = "Proportion of Total",
force.x.continuous = FALSE,
level.to.plot = NULL,
max.plots.per.page = 1,
rug = TRUE,
rug.col = "orange",
censored = FALSE,
...
)

```

Arguments

<code>vpctab</code>	Name of VPC file to use. File will come from VPC command in PsN.
<code>object</code>	Xpose data object.
<code>subset</code>	Subset of data to look at.
<code>main</code>	Title for plot.
<code>main.sub</code>	Used for names above each plot when using multiple plots. Should be a vector, e.g. <code>c("title 1", "title 2")</code> .
<code>main.sub.cex</code>	Size of <code>main.sub</code>
<code>real.col</code>	Color of real line.
<code>real.lty</code>	Real line type.
<code>real.cex</code>	Size of real line.
<code>real.lwd</code>	Width of real line.
<code>median.line</code>	Draw a median line?
<code>median.col</code>	Color of median line.
<code>median.lty</code>	median line type.

<code>ci.lines</code>	Lines marking confidence interval?
<code>ci.col</code>	Color of CI area.
<code>ci.lines.col</code>	Color of CI lines.
<code>ci.lines.lty</code>	Type of CI lines.
<code>xlb</code>	X-axis label. If other than "default" passed directly to <code>xyplot</code> .
<code>ylb</code>	Y-axis label. Passed directly to <code>xyplot</code> .
<code>force.x.continuous</code>	For the x variable to be continuous.
<code>level.to.plot</code>	Which levels of the variable to plot. Smallest level is 1, largest is <code>number_of_levels</code> . For example, with 4 levels, the largest level would be 4, the smallest would be 1.
<code>max.plots.per.page</code>	The number of plots per page.
<code>rug</code>	Should there be markings on the plot showing where the intervals for the VPC are?
<code>rug.col</code>	Color of the rug.
<code>censored</code>	Is this censored data? Censored data can be both below and above the limit of quantification.
<code>...</code>	Additional information passed to function.

Author(s)

Andrew C. Hooker

See Also

[xpose.VPC.both](#).

Other specific functions: [absval.cwres.vs.cov.bw\(\)](#), [absval.cwres.vs.pred\(\)](#), [absval.cwres.vs.pred.by.cov\(\)](#), [absval.iwres.cwres.vs.ipred.pred\(\)](#), [absval.iwres.vs.cov.bw\(\)](#), [absval.iwres.vs.idv\(\)](#), [absval.iwres.vs.ipred\(\)](#), [absval.iwres.vs.ipred.by.cov\(\)](#), [absval.iwres.vs.pred\(\)](#), [absval.wres.vs.cov.bw\(\)](#), [absval.wres.vs.idv\(\)](#), [absval.wres.vs.pred\(\)](#), [absval.wres.vs.pred.by.cov\(\)](#), [absval_delta_vs_cov_model_comp\(\)](#), [addit.gof\(\)](#), [autocorr.cwres\(\)](#), [autocorr.iwres\(\)](#), [autocorr.wres\(\)](#), [basic.gof\(\)](#), [basic.model.comp\(\)](#), [cat.dv.vs.idv.sb\(\)](#), [cat.pc\(\)](#), [cov.splom\(\)](#), [cwres.dist.hist\(\)](#), [cwres.dist.qq\(\)](#), [cwres.vs.cov\(\)](#), [cwres.vs.idv\(\)](#), [cwres.vs.idv.bw\(\)](#), [cwres.vs.pred\(\)](#), [cwres.vs.pred.bw\(\)](#), [cwres.wres.vs.idv\(\)](#), [cwres.wres.vs.pred\(\)](#), [d0FV.vs.cov\(\)](#), [d0FV.vs.id\(\)](#), [d0FV1.vs.d0FV2\(\)](#), [data.checkout\(\)](#), [dv.preds.vs.idv\(\)](#), [dv.vs.idv\(\)](#), [dv.vs.ipred\(\)](#), [dv.vs.ipred.by.cov\(\)](#), [dv.vs.ipred.by.idv\(\)](#), [dv.vs.pred\(\)](#), [dv.vs.pred.by.cov\(\)](#), [dv.vs.pred.by.idv\(\)](#), [dv.vs.pred.ipred\(\)](#), [gof\(\)](#), [ind.plots\(\)](#), [ind.plots.cwres.hist\(\)](#), [ind.plots.cwres.qq\(\)](#), [ipred.vs.idv\(\)](#), [iwres.dist.hist\(\)](#), [iwres.dist.qq\(\)](#), [iwres.vs.idv\(\)](#), [kaplan.plot\(\)](#), [par_cov_hist](#), [par_cov_qq](#), [parm.vs.cov\(\)](#), [parm.vs.parm\(\)](#), [pred.vs.idv\(\)](#), [ranpar.vs.cov\(\)](#), [runsum\(\)](#), [wres.dist.hist\(\)](#), [wres.dist.qq\(\)](#), [wres.vs.idv\(\)](#), [wres.vs.idv.bw\(\)](#), [wres.vs.pred\(\)](#), [wres.vs.pred.bw\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose4-package](#)

Other PsN functions: [boot.hist\(\)](#), [bootscm.import\(\)](#), [npc.coverage\(\)](#), [randtest.hist\(\)](#), [read.npc.vpc.results\(\)](#), [read.vpctab\(\)](#), [xpose.VPC\(\)](#), [xpose.VPC.both\(\)](#), [xpose4-package](#)

Examples

```
## Not run:
library(xpose4)

## move to the directory where results from PsN
## are found
cur.dir <- getwd()
setwd(paste(cur.dir, "/binary/vpc_36", sep=""))

xpose.VPC.categorical(level.to.plot=1,max.plots.per.page=4)
xpose.VPC.categorical(level.to.plot=1,max.plots.per.page=4,by="DOSE")

## ordered categorical plots
setwd(paste(cur.dir, "/ordered_cat/vpc_45", sep=""))
xpose.VPC.categorical()

## count
setwd(paste(cur.dir, "/count/vpc65b", sep=""))
xpose.VPC.categorical()

setwd(paste(cur.dir, "/count/vpc65a", sep=""))
xpose.VPC.categorical()

## End(Not run)
```

xpose4

Classic menu system for Xpose 4

Description

Classic menu system for Xpose 4

Usage

```
xpose4()
```

Author(s)

Andrew Hooker

See Also

Other classic functions: [xpose4-package](#)

Examples

```
## Not run:  
xpose4()  
  
## End(Not run)
```

xsubset

Extract or set the value of the Subset slot.

Description

Extract or set the value of the Subset slot of an "xpose.data" object.

Usage

```
xsubset(object)  
  
xsubset(object) <- value
```

Arguments

object	An "xpose.data" object.
value	A string with the subset expression.

Details

The subset string has the same syntax as the subset argument to, e.g. `panel.xyplot`. Note, however, that the "xpose.data" subset is not used as an argument to `panel.xyplot`. It is intended as the subset argument to the `Data` and `SData` functions.

Value

A string representing the subset expression.

Functions

- `xsubset(object) <- value`: assign value with a string representing the subset expression

Author(s)

Niclas Jonsson

See Also

[Data](#), [SData](#)

Other data functions: [add_transformed_columns](#), [change_graphical_parameters](#), [change_misc_parameters](#), [compute.cwres\(\)](#), [data.checkout\(\)](#), [data_extract_or_assign](#), [db.names\(\)](#), [export.graph.par\(\)](#), [export.variable.definitions\(\)](#), [import.graph.par\(\)](#), [import.variable.definitions\(\)](#), [make.sb.data\(\)](#), [nsim\(\)](#), [par_cov_summary](#), [read.TTE.sim.data\(\)](#), [read.nm.tables\(\)](#), [read_NM_output](#), [read_nm_table\(\)](#), [simprazExample\(\)](#), [tabulate.parameters\(\)](#), [xlabel\(\)](#), [xpose.data](#), [xpose.print\(\)](#), [xpose4-package](#)

Examples

```
xpdb <- simpraz.xpdb
xsubset(xpdb) <- "DV > 0"
xsubset(xpdb)
```

xvardef

Extract and set Xpose variable definitions.

Description

This function extracts and set Xpose variable definitions in "xpose.data" objects.

Usage

```
xvardef(x, object)

xvardef(object) <- value
```

Arguments

x	The name of an xpose variable (see below).
object	An xpose.data object.
value	A two element vector of which the first element is the name of the variable and the second the column name in the Data slot of the object.

Details

The Xpose variable definitions are used to map particular variable types to column names in the data.frame in the Data slot of the "xpose.data" object. The single-valued Xpose variable definitions are: id, idlab, idv, occ, dv, pred, ipred, iwres, res. The (potentially) vector-valued Xpose variable definitions are: parms, covariates, ranpar, tvparms (parameters, covariates, random effects parameters=etas, typical value parameters). The default values of these can be found in the createXposeClasses function.

Value

Returns a string with the name of the data variable defined as the Xpose data variable.

Functions

- `xvardef(object) <- value`: reset the which column the label dv points to in the Data slot of the xpose database object

Author(s)

Niclas Jonsson

See Also

[xpose.data-class](#), [xpose.prefs-class](#)

Examples

```
xpdb <- simpraz.xpdb

## get the column name in the Data slot of object xpdb
## corresponding to the label dv
xvardef("dv", xpdb)

## reset the which column the label dv points to in the Data slot of
## object xpdb
xvardef(xpdb) <- c("dv", "DVA")
```

Index

- * **GAM functions**
 - GAM_summary_and_plot, 118
 - xp.get.disp, 192
 - xp.scope3, 201
 - xpose.bootgam, 202
 - xpose.gam, 207
 - xpose4-package, 5
- * **PsN functions**
 - boot.hist, 49
 - bootscm.import, 52
 - npc.coverage, 145
 - randtest.hist, 159
 - read.npc.vpc.results, 164
 - read.vpctab, 166
 - xpose.VPC, 254
 - xpose.VPC.both, 260
 - xpose.VPC.categorical, 262
 - xpose4-package, 5
- * **~bootgam**
 - bootgam.print, 51
 - xp.distr.mod.size, 190
 - xp.dofv.plot, 192
 - xp.inc.prob, 195
 - xp.inc.prob.comb.2, 196
 - xp.incl.index.cov.comp, 199
- * **~bootscm**
 - bootgam.print, 51
 - xp.boot.par.est, 186
 - xp.boot.par.est.corr, 188
 - xp.distr.mod.size, 190
 - xp.dofv.plot, 192
 - xp.inc.prob, 195
 - xp.inc.prob.comb.2, 196
 - xp.incl.index.cov.comp, 199
- * **bootgam**
 - xp.daic.npar.plot, 189
 - xp.dofv.npar.plot, 191
 - xp.inc.cond.stab.cov, 193
 - xp.inc.ind.cond.stab.cov, 194
 - xp.inc.stab.cov, 197
 - xp.incl.index.cov, 198
 - xp.incl.index.cov.ind, 200
- * **bootscm**
 - bootscm.import, 52
 - xp.daic.npar.plot, 189
 - xp.dofv.npar.plot, 191
 - xp.inc.cond.stab.cov, 193
 - xp.inc.ind.cond.stab.cov, 194
 - xp.inc.stab.cov, 197
 - xp.incl.index.cov, 198
 - xp.incl.index.cov.ind, 200
- * **classes**
 - xpose.data-class, 207
 - xpose.prefs-class, 252
- * **classic functions**
 - xpose4, 265
 - xpose4-package, 5
- * **data functions**
 - add_transformed_columns, 39
 - change_graphical_parameters, 62
 - change_misc_parameters, 64
 - compute.cwres, 67
 - data.checkout, 92
 - data_extract_or_assign, 94
 - db.names, 96
 - export.graph.par, 115
 - export.variable.definitions, 117
 - import.graph.par, 123
 - import.variable.definitions, 124
 - make.sb.data, 144
 - nsim, 147
 - par_cov_summary, 155
 - read.nm.tables, 162
 - read.TTE.sim.data, 165
 - read_NM_output, 168
 - read_nm_table, 169
 - simprazExample, 173
 - tabulate.parameters, 174

- xlabel, 185
- xpose.data, 204
- xpose.print, 253
- xpose4-package, 5
- xsubset, 266
- * **datasets**
 - simpraz.xpdb, 173
- * **generic functions**
 - gof, 120
 - xpose.multiple.plot, 212
 - xpose4-package, 5
- * **methods**
 - absval.cwres.vs.cov.bw, 7
 - absval.cwres.vs.pred, 9
 - absval.cwres.vs.pred.by.cov, 11
 - absval.iwres.vs.ipred, 17
 - absval.iwres.vs.ipred.by.cov, 19
 - absval.iwres.vs.pred, 21
 - absval.wres.vs.cov.bw, 22
 - absval.wres.vs.idv, 24
 - absval.wres.vs.pred, 26
 - absval_delta_vs_cov_model_comp, 30
 - add.grid.table, 33
 - add.model.comp, 35
 - addit.gof, 37
 - autocorr.cwres, 40
 - autocorr.wres, 44
 - basic.model.comp, 47
 - boot.hist, 49
 - cat.dv.vs.idv.sb, 53
 - cat.pc, 55
 - change.parm, 57
 - change.var.name, 58
 - change.xlabel, 59
 - change.xvardef, 60
 - change_misc_parameters, 64
 - compute.cwres, 67
 - createXposeClasses, 78
 - cwres.dist.hist, 79
 - cwres.dist.qq, 80
 - cwres.vs.cov, 81
 - cwres.vs.idv, 83
 - cwres.vs.idv.bw, 84
 - cwres.vs.pred, 86
 - cwres.vs.pred.bw, 87
 - data.checkout, 92
 - data_extract_or_assign, 94
 - db.names, 96
 - dOFV.vs.cov, 97
 - dOFV.vs.id, 99
 - dOFV1.vs.dOFV2, 101
 - dv.preds.vs.idv, 102
 - dv.vs.idv, 104
 - dv.vs.ipred, 105
 - dv.vs.ipred.by.cov, 107
 - dv.vs.ipred.by.idv, 108
 - dv.vs.pred, 110
 - dv.vs.pred.by.cov, 111
 - dv.vs.pred.by.idv, 113
 - dv.vs.pred.ipred, 114
 - export.graph.par, 115
 - export.variable.definitions, 117
 - gof, 120
 - import.variable.definitions, 124
 - ind.plots, 126
 - ind.plots.cwres.qq, 132
 - ipred.vs.idv, 134
 - iwres.dist.hist, 136
 - iwres.dist.qq, 137
 - iwres.vs.idv, 138
 - make.sb.data, 144
 - npc.coverage, 145
 - nsim, 147
 - par_cov_hist, 151
 - par_cov qq, 153
 - par_cov_summary, 155
 - parm.vs.cov, 148
 - parm.vs.parm, 150
 - pred.vs.idv, 157
 - randtest.hist, 159
 - ranpar.vs.cov, 160
 - read.nm.tables, 162
 - read.npc.vpc.results, 164
 - read.vpctab, 166
 - reset.graph.par, 170
 - runsum, 171
 - simprazExample, 173
 - tabulate.parameters, 174
 - wres.dist.hist, 175
 - wres.dist.qq, 176
 - wres.vs.cov, 178
 - wres.vs.idv.bw, 181
 - wres.vs.pred, 182
 - wres.vs.pred.bw, 183
 - xlabel, 185
 - xpose.data, 204

- xpose.license.citation, 210
- xpose.logTicks, 210
- xpose.panel.bw, 215
- xpose.panel.default, 218
- xpose.panel.histogram, 230
- xpose.panel.qq, 232
- xpose.panel.splom, 233
- xpose.plot.bw, 236
- xpose.plot.default, 239
- xpose.plot.histogram, 244
- xpose.plot.qq, 247
- xpose.plot.splom, 249
- xpose.string.print, 254
- xpose.VPC, 254
- xpose.VPC.both, 260
- xpose.VPC.categorical, 262
- xpose4, 265
- xpose4-package, 5
- xsubset, 266
- * package**
 - xpose4-package, 5
- * specific functions**
 - absval.cwres.vs.cov.bw, 7
 - absval.cwres.vs.pred, 9
 - absval.cwres.vs.pred.by.cov, 11
 - absval.iwres.cwres.vs.ipred.pred, 13
 - absval.iwres.vs.cov.bw, 15
 - absval.iwres.vs.idv, 16
 - absval.iwres.vs.ipred, 17
 - absval.iwres.vs.ipred.by.cov, 19
 - absval.iwres.vs.pred, 21
 - absval.wres.vs.cov.bw, 22
 - absval.wres.vs.idv, 24
 - absval.wres.vs.pred, 26
 - absval.wres.vs.pred.by.cov, 28
 - absval_delta_vs_cov_model_comp, 30
 - addit.gof, 37
 - autocorr.cwres, 40
 - autocorr.iwres, 42
 - autocorr.wres, 44
 - basic.gof, 46
 - basic.model.comp, 47
 - cat.dv.vs.idv.sb, 53
 - cat.pc, 55
 - cov.splom, 75
 - cwres.dist.hist, 79
 - cwres.dist.qq, 80
 - cwres.vs.cov, 81
 - cwres.vs.idv, 83
 - cwres.vs.idv.bw, 84
 - cwres.vs.pred, 86
 - cwres.vs.pred.bw, 87
 - cwres.wres.vs.idv, 89
 - cwres.wres.vs.pred, 90
 - data.checkout, 92
 - dOFV.vs.cov, 97
 - dOFV.vs.id, 99
 - dOFV1.vs.dOFV2, 101
 - dv.preds.vs.idv, 102
 - dv.vs.idv, 104
 - dv.vs.ipred, 105
 - dv.vs.ipred.by.cov, 107
 - dv.vs.ipred.by.idv, 108
 - dv.vs.pred, 110
 - dv.vs.pred.by.cov, 111
 - dv.vs.pred.by.idv, 113
 - dv.vs.pred.ipred, 114
 - gof, 120
 - ind.plots, 126
 - ind.plots.cwres.hist, 129
 - ind.plots.cwres.qq, 132
 - ipred.vs.idv, 134
 - iwres.dist.hist, 136
 - iwres.dist.qq, 137
 - iwres.vs.idv, 138
 - kaplan.plot, 140
 - par_cov_hist, 151
 - par_cov_qq, 153
 - parm.vs.cov, 148
 - parm.vs.parm, 150
 - pred.vs.idv, 157
 - ranpar.vs.cov, 160
 - runsum, 171
 - wres.dist.hist, 175
 - wres.dist.qq, 176
 - wres.vs.idv, 179
 - wres.vs.idv.bw, 181
 - wres.vs.pred, 182
 - wres.vs.pred.bw, 183
 - xpose.VPC, 254
 - xpose.VPC.both, 260
 - xpose.VPC.categorical, 262
 - xpose4-package, 5
- absval.cwres.vs.cov.bw, 7, 7, 10, 12, 14–16, 18, 20, 22, 23, 25, 27, 30, 33,

- 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.cwres.vs.pred, 7, 8, 9, 12, 14–16, 18, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.cwres.vs.pred.by.cov, 7, 8, 10, 11, 14–16, 18, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.dcwres.vs.cov.model.comp
(absval_delta_vs_cov_model_comp),
30
- absval.dipred.vs.cov.model.comp
(absval_delta_vs_cov_model_comp),
30
- absval.diwres.vs.cov.model.comp
(absval_delta_vs_cov_model_comp),
30
- absval.dpred.vs.cov.model.comp
(absval_delta_vs_cov_model_comp),
30
- absval.dwres.vs.cov.model.comp
(absval_delta_vs_cov_model_comp),
30
- absval.iwres.cwres.vs.ipred.pred, 7, 8, 10, 12, 13, 15, 16, 18, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.vs.cov.bw, 7, 8, 10, 12, 14, 15, 16, 18, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.vs.idv, 7, 8, 10, 12, 14, 15, 16, 18, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 103, 105, 106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.vs.ipred, 7, 8, 10, 12, 14–16, 17, 20, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.vs.ipred.by.cov, 7, 8, 10, 12, 14–16, 18, 19, 22, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.vs.pred, 7, 8, 10, 12, 14–16, 18, 20, 21, 23, 25, 27, 30, 33, 38, 41, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108–110, 112, 113, 115, 122, 128, 131, 134–136, 138, 139, 143, 149, 151, 152, 154, 158, 161, 172, 176, 177, 180, 181, 183, 184, 259, 261, 264
- absval.iwres.wres.vs.ipred.pred

- (absval.iwres.cwres.vs.ipred.pred),
13
- absval.wres.vs.cov.bw, 7, 8, 10, 12, 14–16,
18, 20, 22, 22, 25, 27, 30, 33, 38, 41,
43, 45, 47, 48, 55, 57, 77, 79, 81, 82,
84, 85, 87, 88, 90, 91, 93, 98, 100,
102, 104–106, 108–110, 112, 113,
115, 122, 128, 131, 134–136, 138,
139, 143, 149, 151, 152, 154, 158,
161, 172, 176, 177, 180, 181, 183,
184, 259, 261, 264
- absval.wres.vs.idv, 7, 8, 10, 12, 14–16, 18,
20, 22, 23, 24, 27, 30, 33, 38, 41, 43,
45, 47, 48, 55, 57, 77, 79, 81, 82, 84,
85, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134–136, 138, 139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 261, 264
- absval.wres.vs.pred, 7, 8, 10, 12, 14–16,
18, 20, 22, 23, 25, 26, 30, 33, 38, 41,
43, 45, 47, 48, 55, 57, 77, 79, 81, 82,
84, 85, 87, 88, 90, 91, 93, 98, 100,
102, 104–106, 108–110, 112, 113,
115, 122, 128, 131, 134–136, 138,
139, 143, 149, 151, 152, 154, 158,
161, 172, 176, 177, 180, 181, 183,
184, 259, 261, 264
- absval.wres.vs.pred.by.cov, 7, 8, 10, 12,
14–16, 18, 20, 22, 23, 25, 27, 28, 33,
38, 41, 43, 45, 47, 48, 55, 57, 77, 79,
81, 82, 84, 85, 87, 88, 90, 91, 93, 98,
100, 102, 104–106, 108–110, 112,
113, 115, 122, 128, 131, 134–136,
138, 139, 143, 149, 151, 152, 154,
158, 161, 172, 176, 177, 180, 181,
183, 184, 259, 261, 264
- absval_delta_vs_cov_model_comp, 7, 8, 10,
12, 14–16, 18, 20, 22, 23, 25, 27, 30,
30, 38, 41, 43, 45, 47, 48, 55, 57, 77,
79, 81, 82, 84, 85, 87, 88, 90, 91, 93,
98, 100, 102, 104–106, 108–110,
112, 113, 115, 122, 128, 131,
134–136, 138, 139, 143, 149, 151,
152, 154, 158, 161, 172, 176, 177,
180, 181, 183, 184, 259, 261, 264
- add.absval (add_transformed_columns), 39
- add.dichot (add_transformed_columns), 39
- add.exp (add_transformed_columns), 39
- add.grid.table, 33
- add.grid.text (add.grid.table), 33
- add.log (add_transformed_columns), 39
- add.model.comp, 35
- add.tad (add_transformed_columns), 39
- add_transformed_columns, 7, 39, 63, 67, 75,
93, 95, 97, 116, 117, 124, 125, 145,
147, 156, 164, 166, 168, 169, 174,
175, 186, 206, 253, 267
- addit.gof, 7, 8, 10, 12, 14–16, 18, 20, 22, 23,
25, 27, 30, 33, 37, 42, 43, 45, 47, 48,
55, 57, 77, 79, 81, 82, 84, 85, 87, 88,
90, 91, 93, 98, 100, 102, 104–106,
108–110, 112, 113, 115, 122, 128,
131, 134–136, 138, 139, 143, 149,
151, 152, 154, 158, 161, 172, 176,
177, 180, 181, 183, 184, 259, 261,
264
- autocorr.cwres, 7, 8, 10, 12, 14–16, 18, 20,
22, 23, 25, 27, 30, 33, 38, 40, 43, 45,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134–136, 138, 139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 261, 264
- autocorr.iwres, 7, 8, 10, 12, 14–16, 18, 20,
22, 23, 25, 27, 30, 33, 38, 42, 42, 45,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134–136, 138, 139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 261, 264
- autocorr.wres, 7, 8, 10, 12, 14–16, 18, 20,
22, 23, 25, 27, 30, 33, 38, 42, 43, 44,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134–136, 138, 139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 261, 264
- basic.gof, 7, 8, 10, 12, 14–16, 18, 20, 22, 23,

- 25, 27, 30, 33, 38, 42, 43, 45, 46, 48,
55, 57, 77, 79, 81, 82, 84, 85, 87, 88,
90, 91, 93, 98, 100, 102, 104–106,
108–110, 112, 113, 115, 122, 128,
131, 134–136, 138, 139, 143, 149,
151, 152, 154, 158, 161, 172, 176,
177, 180, 181, 183, 184, 214, 259,
261, 264
- basic.model.comp, 7, 8, 10, 12, 14–16, 18,
20, 22, 23, 25, 27, 30, 33, 38, 42, 43,
45, 47, 47, 55, 57, 77, 79, 81, 82, 84,
85, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134–136, 138, 139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 261, 264
- boot.hist, 7, 49, 53, 147, 160, 165, 167, 259,
261, 264
- bootgam.print, 51
- bootscm.import, 7, 50, 52, 147, 160, 165,
167, 189, 191, 194, 195, 198, 199,
201, 259, 261, 264
- bwplot, 8, 23, 85, 88, 181, 184, 236–238
- calc.npar (read_NM_output), 168
- cat.dv.vs.idv.sb, 7, 8, 10, 12, 14–16, 18,
20, 22, 23, 25, 27, 30, 33, 38, 42, 43,
45, 47, 48, 53, 57, 77, 79, 81, 82, 84,
85, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cat.pc, 7, 8, 10, 12, 14–16, 18, 20, 22, 23, 25,
27, 30, 33, 38, 42, 43, 45, 47, 48, 55,
55, 77, 79, 81, 82, 84, 85, 87, 88, 90,
91, 93, 98, 100, 102, 104–106,
108–110, 112, 113, 115, 122, 128,
131, 134, 135, 137–139, 143, 149,
151, 152, 154, 158, 161, 172, 176,
177, 180, 181, 183, 184, 259, 262,
264
- change.ab.graph.par
(change_graphical_parameters),
62
- change.bw.graph.par
(change_graphical_parameters),
62
- change.cat.cont
(change_misc_parameters), 64
- change.cat.cont<-
(change_misc_parameters), 64
- change.cat.levels
(change_misc_parameters), 64
- change.cat.levels<-
(change_misc_parameters), 64
- change.cond.graph.par
(change_graphical_parameters),
62
- change.dil.graph.par
(change_graphical_parameters),
62
- change.dv.cat.levels
(change_misc_parameters), 64
- change.dv.cat.levels<-
(change_misc_parameters), 64
- change.label.par
(change_graphical_parameters),
62
- change.lm.graph.par
(change_graphical_parameters),
62
- change.misc.graph.par
(change_graphical_parameters),
62
- change.miss (change_misc_parameters), 64
- change.parm, 57
- change.pi.graph.par
(change_graphical_parameters),
62
- change.smooth.graph.par
(change_graphical_parameters),
62
- change.subset (change_misc_parameters),
64
- change.var.name, 58
- change.xlabel, 59
- change.xvardef, 60, 170
- change.xvardef<- (change.xvardef), 60
- change_graphical_parameters, 7, 40, 62,
67, 75, 93, 95, 97, 116, 117, 124,
125, 145, 147, 156, 164, 166, 168,
169, 174, 175, 186, 206, 253, 267
- change_misc_parameters, 7, 40, 63, 64, 75,
93, 95, 97, 116, 117, 124, 125, 145,

- 147, 156, 164, 166, 168, 169, 174,
175, 186, 206, 253, 267
- character_or_NULL-class
(xpose.multiple.plot-class),
213
- character_or_numeric-class
(xpose.prefs-class), 252
- compute.cwres, 7–10, 12, 25, 32, 33, 36, 40,
41, 46–48, 63, 67, 67, 79, 81–91, 93,
95, 97, 116, 117, 124, 125, 145, 147,
156, 164, 166, 168, 169, 174, 175,
177, 183, 186, 206, 253, 267
- cov.hist (par_cov_hist), 151
- cov.qq (par_cov_qq), 153
- cov.splom, 7, 8, 10, 12, 14–16, 18, 20, 22, 23,
25, 27, 30, 33, 38, 42, 43, 45, 47, 48,
55, 57, 75, 79, 81, 82, 84, 85, 87, 88,
90, 91, 93, 98, 100, 102, 104–106,
108–110, 112, 113, 115, 122, 128,
131, 134, 135, 137–139, 143, 149,
151, 152, 154, 158, 161, 172, 176,
177, 180, 181, 183, 184, 259, 262,
264
- cov.summary (par_cov_summary), 155
- create.mirror, 238, 242, 246, 249, 251
- create.parameter.list (read_NM_output),
168
- create.xpose.plot.classes, 78
- createXposeClasses, 78
- cwres.dist.hist, 7, 8, 10, 12, 14–16, 18, 20,
22, 23, 25, 27, 30, 33, 38, 42, 43, 45,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.dist.qq, 7, 8, 10, 12, 14–16, 18, 20,
22, 23, 25, 27, 30, 33, 38, 42, 43, 45,
47, 48, 55, 57, 77, 79, 80, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.vs.cov, 7, 8, 10, 12, 14–16, 18, 20, 22,
23, 25, 27, 30, 33, 38, 42, 43, 45, 47,
48, 55, 57, 77, 79, 81, 81, 84, 85, 87,
88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 152, 154, 158, 161,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.vs.idv, 7, 8, 10, 12, 14, 15, 17, 18, 20,
22, 23, 25, 27, 30, 33, 38, 42, 43, 45,
47, 48, 55, 57, 77, 79, 81, 82, 83, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 153, 154, 158, 162,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.vs.idv.bw, 7, 8, 10, 12, 14, 15, 17, 18,
20, 22, 23, 25, 27, 30, 33, 38, 42, 43,
45, 47, 48, 55, 57, 77, 79, 81, 82, 84,
84, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 153, 154, 158, 162,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.vs.pred, 7, 8, 10, 12, 14, 15, 17, 18,
20, 22, 23, 25, 27, 30, 33, 38, 42, 43,
45, 47, 48, 55, 57, 77, 79, 81, 82, 84,
85, 86, 88, 90, 91, 93, 98, 100, 102,
104–106, 108–110, 112, 113, 115,
122, 128, 131, 134, 135, 137–139,
143, 149, 151, 153, 154, 158, 162,
172, 176, 177, 180, 181, 183, 184,
259, 262, 264
- cwres.vs.pred.bw, 7, 8, 10, 12, 14, 15, 17,
18, 20, 22, 23, 25, 27, 30, 33, 38, 42,
43, 45, 47, 48, 55, 57, 77, 79, 81, 82,
84, 85, 87, 87, 90, 91, 93, 98, 100,
102, 104–106, 108–110, 112, 113,
115, 122, 128, 131, 134, 135,
137–139, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180, 181,
183, 184, 259, 262, 264
- cwres.wres.vs.idv, 7, 8, 10, 12, 14, 15, 17,
18, 20, 22, 23, 25, 27, 30, 33, 38, 42,
43, 45, 47, 48, 55, 57, 77, 79, 81, 82,
84, 85, 87, 88, 89, 91, 93, 98, 100,

- 102, 104–106, 108–110, 112, 113, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 181, 183, 184, 259, 262, 264
- `cwres.wres`.vs.`pred`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 90, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- Data, 58, 59, 67, 156, 206, 207, 253, 267
- Data(`data_extract_or_assign`), 94
- `data.checkout`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 40, 42, 43, 45, 47, 48, 55, 57, 63, 67, 75, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 92, 95, 97, 98, 100, 102, 104–106, 108, 109, 111, 112, 114–117, 122, 124, 125, 128, 131, 134, 135, 137–139, 143, 145, 147, 149, 151, 153, 154, 156, 158, 162, 164, 166, 168, 169, 172, 174–177, 180, 182–184, 186, 206, 253, 259, 262, 264, 267
- `data.frame_or_NULL`-class
(`xpose.data`-class), 207
- `Data<-`(`data_extract_or_assign`), 94
- `data_extract_or_assign`, 7, 40, 63, 67, 75, 93, 94, 97, 116, 117, 124, 125, 145, 147, 156, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- `db.names`, 7, 40, 63, 67, 75, 93, 95, 96, 116, 117, 124, 125, 145, 147, 156, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- `dget`, 125
- `dOFV`.vs.`cov`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 97, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `dOFV`.vs.`id`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 99, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `dOFV1`.vs.`dOFV2`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 101, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `dotplot`, 93, 120
- `dplyr`, 169
- `dput`, 117
- `dv.preds`.vs.`idv`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 102, 105, 106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `dv`.vs.`idv`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102–104, 104, 106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `dv`.vs.`ipred`, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104, 105, 105, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264

- dv.vs.ipred.by.cov*, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 107, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- dv.vs.ipred.by.idv*, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 25, 27, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 108, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- dv.vs.pred*, 7, 8, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 110, 112–115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- dv.vs.pred.by.cov*, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 111, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- dv.vs.pred.by.idv*, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 113, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- dv.vs.pred.ipred*, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 114, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- export.graph.par*, 7, 40, 63, 67, 75, 93, 95, 97, 115, 117, 124, 125, 145, 147, 156, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- export.variable.definitions*, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 123–125, 145, 147, 156, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- gam*, 119, 120
- GAM.summary.and.plot*, 7, 118, 193, 202, 204, 209
- get.doc* (*change.misc.parameters*), 64
- gof*, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 48, 55, 57, 77, 79, 81, 82, 84, 85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 120, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 213, 259, 262, 264
- gofSetup* (*gof*), 120
- grid.polygon*, 146, 257, 258
- grid.text*, 35
- help*, 25
- histogram*, 50, 79, 82, 130, 131, 136, 149, 152, 160, 161, 176, 179, 245, 246
- import.graph.par*, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 123, 125, 145, 147, 156, 164, 166, 168–170, 174, 175, 186, 206, 253, 267
- import.variable.definitions*, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 123, 124, 124, 145, 147, 156, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- ind.cwres* (*compute.cwres*), 67
- ind.plots*, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47,

- 48, 55, 57, 77, 79, 81, 82, 84, 85, 87,
88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 126, 131, 134, 135,
137–139, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `ind.plots.cwres.hist`, 7, 9, 10, 12, 14, 15,
17, 18, 20, 22, 23, 26, 28, 30, 33, 38,
42, 43, 45, 47, 48, 55, 57, 77, 79, 81,
82, 84, 85, 87, 88, 90, 91, 93, 98,
100, 102, 104–106, 108, 109, 111,
112, 114, 115, 122, 128, 129, 134,
135, 137–139, 143, 149, 151, 153,
154, 158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `ind.plots.cwres.qq`, 7, 9, 10, 12, 14, 15, 17,
18, 20, 22, 23, 26, 28, 30, 33, 38, 42,
43, 45, 47, 48, 55, 57, 77, 79, 81, 82,
84, 85, 87, 88, 90, 91, 93, 98, 100,
102, 104–106, 108, 109, 111, 112,
114, 115, 122, 128, 131, 132, 135,
137–139, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `ind.plots.wres.hist`
(`ind.plots.cwres.hist`), 129
- `ind.plots.wres.qq` (`ind.plots.cwres.qq`),
132
- `ipred.vs.idv`, 7, 9, 10, 12, 14, 15, 17, 18, 20,
22, 23, 26, 28, 30, 33, 38, 42, 43, 45,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102–106,
108, 109, 111, 112, 114, 115, 122,
128, 131, 134, 134, 137–139, 143,
149, 151, 153, 154, 158, 162, 172,
176, 177, 180, 182–184, 259, 262,
264
- `is.cwres.readable.file` (`compute.cwres`),
67
- `iwres.dist.hist`, 7, 9, 10, 12, 14, 15, 17, 18,
20, 22, 23, 26, 28, 30, 33, 38, 42, 43,
45, 47, 48, 55, 57, 77, 79, 81, 82, 84,
85, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 128, 131, 134, 135, 136,
138, 139, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `iwres.dist.qq`, 7, 9, 10, 12, 14, 15, 17, 18,
20, 22, 23, 26, 28, 30, 33, 38, 42, 43,
45, 47, 48, 55, 57, 77, 79, 81, 82, 84,
85, 87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 128, 131, 134, 135, 137,
137, 139, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `iwres.vs.idv`, 7, 9, 10, 12, 14, 15, 17, 18, 20,
22, 23, 26, 28, 30, 33, 38, 42, 43, 45,
47, 48, 55, 57, 77, 79, 81, 82, 84, 85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 128, 131, 134, 135, 137,
138, 138, 143, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `kaplan.plot`, 7, 9, 10, 12, 14, 15, 17, 18, 20,
22, 23, 26, 28, 30, 33, 38, 42, 43, 45,
47, 49, 55, 57, 77, 79, 81, 83–85, 87,
88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 128, 131, 134, 135,
137–139, 140, 149, 151, 153, 154,
158, 162, 172, 176, 177, 180,
182–184, 259, 262, 264
- `lattice-package`, 50, 160
- `list_or_NULL-class`
(`xpose.multiple.plot-class`),
213
- `lm`, 63
- `loess`, 63
- `loess.smooth`, 63
- `logical_or_numeric-class`
(`xpose.multiple.plot-class`),
213
- `make.sb.data`, 7, 40, 63, 67, 75, 93, 95, 97,
116, 117, 124, 125, 144, 147, 157,
164, 166, 168, 169, 174, 175, 186,
206, 253, 267
- `npc.coverage`, 7, 50, 53, 145, 160, 165, 167,
259, 261, 264
- `nsim`, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117,
124, 125, 145, 147, 157, 164, 166,

- 168, 169, 174, 175, 186, 206, 253, 267
- `nsim<-` (`nsim`), 147
- `numeric_or_NULL-class`
(`xpose.data-class`), 207
- `panel.abline`, 63, 83, 86, 89, 91, 106, 107, 109, 110, 112–114, 138, 179, 182, 227, 231
- `panel.bwplot`, 63, 217, 228
- `panel.histogram`, 246
- `panel.lmline`, 63
- `panel.loess`, 63, 227, 228, 235
- `panel.pairs`, 235
- `panel.qqmathline`, 233, 249
- `panel.splom`, 235, 252
- `panel.xyplot`, 237, 243
- `par`, 63
- `par_cov_hist`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 151, 154, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `par_cov_qq`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 153, 158, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `par_cov_splom` (`cov.splom`), 75
- `par_cov_summary`, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 155, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- `parm.hist` (`par_cov_hist`), 151
- `parm.qq` (`par_cov_qq`), 153
- `parm.splom` (`cov.splom`), 75
- `parm.summary` (`par_cov_summary`), 155
- `parm.vs.cov`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 23, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 148, 151, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 214, 259, 262, 264
- `parm.vs.parm`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 150, 153, 154, 158, 162, 172, 176, 177, 180, 182–184, 214, 259, 262, 264
- `plot`, 41, 44, 146, 255, 257, 258
- `plot.default`, 63
- `points`, 133, 227, 235
- `pred.vs.idv`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 157, 162, 172, 176, 177, 180, 182–184, 259, 262, 264
- `print,xpose.multiple.plot-method`
(`create.xpose.plot.classes`), 78
- `print.char.matrix`, 156
- `print.xpose.multiple.plot`, 158, 213
- `qqmath`, 81, 134, 138, 154, 177, 233, 247–249, 251
- `qqplot`, 134
- `randtest.hist`, 7, 50, 53, 147, 159, 165, 167, 259, 261, 264
- `ranpar.hist` (`par_cov_hist`), 151
- `ranpar.qq` (`par_cov_qq`), 153
- `ranpar.splom` (`cov.splom`), 75
- `ranpar.vs.cov`, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 160, 172, 176, 177, 180, 182–184, 259, 262, 264

- read.cwres.data (compute.cwres), 67
- read.lst (read_NM_output), 168
- read.nm.tables, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 162, 166, 168, 169, 174, 175, 186, 206, 207, 253, 267
- read.npc.vpc.results, 7, 50, 53, 147, 160, 164, 167, 222, 259, 261, 264
- read.table, 92, 93
- read.TTE.sim.data, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 164, 165, 168, 169, 174, 175, 186, 206, 253, 267
- read.vpctab, 7, 50, 53, 147, 160, 165, 166, 259, 261, 264
- read_NM_output, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- read_nm_table, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 164, 166, 168, 169, 174, 175, 186, 206, 253, 267
- reset.graph.par, 170
- runsum, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 35, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 171, 176, 177, 180, 182–184, 259, 262, 264
- SData, 58, 59, 67, 156, 206, 207, 253, 267
- SData (data_extract_or_assign), 94
- SData<- (data_extract_or_assign), 94
- set.doc (change_misc_parameters), 64
- setup.PPI, 223
- shingle, 63
- simpraz.xpdb, 173
- simprazExample, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 164, 166, 168, 170, 173, 175, 175, 186, 206, 253, 267
- splom, 77, 235, 252
- sqrtn (compute.cwres), 67
- step.Gam, 209
- step.gam, 209
- strip.custom, 241
- strip.default, 128
- subset, 67
- Surv, 143
- survfit, 142, 143
- tabulate.parameters, 7, 40, 63, 67, 75, 93, 95, 97, 116, 117, 124, 125, 145, 147, 157, 164, 166, 168, 170, 174, 174, 186, 206, 253, 267
- trellis.par.get, 217, 228, 229
- wres.dist.hist, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 175, 177, 180, 182–184, 259, 262, 264
- wres.dist.qq, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 176, 180, 182–184, 259, 262, 264
- wres.vs.cov, 178
- wres.vs.idv, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 179, 182–184, 259, 262, 264
- wres.vs.idv.bw, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22, 24, 26, 28, 30, 33, 38, 42, 43, 45, 47, 49, 55, 57, 77, 79, 81, 83–85, 87, 88, 90, 91, 93, 98, 100, 102, 104–106, 108, 109, 111, 112, 114, 115, 122, 128, 131, 134, 135, 137–139, 143, 149, 151, 153, 154, 158, 162, 172, 176, 177, 180, 181, 183, 184, 259, 262, 264

- wres.vs.pred, [7](#), [9](#), [10](#), [12](#), [14](#), [15](#), [17](#), [18](#), [20](#),
[22](#), [24](#), [26](#), [28](#), [30](#), [33](#), [38](#), [42](#), [43](#), [45](#),
[47](#), [49](#), [55](#), [57](#), [77](#), [80](#), [81](#), [83–85](#), [87](#),
[88](#), [90](#), [91](#), [93](#), [98](#), [100](#), [102](#),
[104–106](#), [108](#), [109](#), [111](#), [112](#), [114](#),
[115](#), [122](#), [128](#), [131](#), [134](#), [135](#),
[137–139](#), [143](#), [149](#), [151](#), [153](#), [154](#),
[158](#), [162](#), [172](#), [176](#), [177](#), [180](#), [182](#),
[182](#), [184](#), [259](#), [262](#), [264](#)
- wres.vs.pred.bw, [7](#), [9](#), [10](#), [12](#), [14](#), [15](#), [17](#), [18](#),
[20](#), [22](#), [24](#), [26](#), [28](#), [30](#), [33](#), [38](#), [42](#), [43](#),
[45](#), [47](#), [49](#), [55](#), [57](#), [77](#), [80](#), [81](#), [83–85](#),
[87](#), [88](#), [90](#), [91](#), [93](#), [98](#), [100](#), [102](#),
[104–106](#), [108](#), [109](#), [111](#), [112](#), [114](#),
[115](#), [122](#), [128](#), [131](#), [134](#), [135](#),
[137–139](#), [143](#), [149](#), [151](#), [153](#), [154](#),
[158](#), [162](#), [172](#), [176](#), [177](#), [180](#), [182](#),
[183](#), [183](#), [259](#), [262](#), [264](#)
- xlabel, [7](#), [40](#), [63](#), [67](#), [75](#), [93](#), [95](#), [97](#), [116](#), [117](#),
[124](#), [125](#), [145](#), [147](#), [157](#), [164](#), [166](#),
[168](#), [170](#), [174](#), [175](#), [185](#), [206](#), [253](#),
[267](#)
- xlabel<- (xlabel), [185](#)
- xp.akaike.plot (GAM_summary_and_plot),
[118](#)
- xp.boot.par.est, [186](#)
- xp.boot.par.est.corr, [188](#)
- xp.cook (GAM_summary_and_plot), [118](#)
- xp.daic.npar.plot, [53](#), [189](#), [191](#), [194](#), [195](#),
[198](#), [199](#), [201](#)
- xp.distr.mod.size, [190](#)
- xp.dofv.npar.plot, [53](#), [189](#), [191](#), [194](#), [195](#),
[198](#), [199](#), [201](#)
- xp.dofv.plot, [192](#)
- xp.get.disp, [7](#), [120](#), [192](#), [202](#), [204](#), [209](#)
- xp.inc.cond.stab.cov, [53](#), [189](#), [191](#), [193](#),
[195](#), [198](#), [199](#), [201](#)
- xp.inc.ind.cond.stab.cov, [53](#), [189](#), [191](#),
[194](#), [194](#), [198](#), [199](#), [201](#)
- xp.inc.prob, [195](#)
- xp.inc.prob.comb.2, [196](#)
- xp.inc.stab.cov, [53](#), [189](#), [191](#), [194](#), [195](#),
[197](#), [199](#), [201](#)
- xp.incl.index.cov, [53](#), [189](#), [191](#), [194](#), [195](#),
[198](#), [198](#), [201](#)
- xp.incl.index.cov.comp, [199](#)
- xp.incl.index.cov.ind, [53](#), [189](#), [191](#), [194](#),
[195](#), [198](#), [199](#), [200](#)
- xp.ind.inf.fit (GAM_summary_and_plot),
[118](#)
- xp.ind.inf.terms
(GAM_summary_and_plot), [118](#)
- xp.ind.stud.res (GAM_summary_and_plot),
[118](#)
- xp.plot (GAM_summary_and_plot), [118](#)
- xp.scope3, [7](#), [120](#), [193](#), [201](#), [204](#), [209](#)
- xp.summary (GAM_summary_and_plot), [118](#)
- xpose (xpose4-package), [5](#)
- xpose.bootgam, [7](#), [120](#), [193](#), [202](#), [202](#), [209](#)
- xpose.calculate.cwres (compute.cwres),
[67](#)
- xpose.data, [7](#), [39](#), [40](#), [58](#), [59](#), [61](#), [63](#), [66](#), [67](#),
[75](#), [93](#), [95–97](#), [116](#), [117](#), [124](#), [125](#),
[145](#), [147](#), [157](#), [164](#), [166–168](#), [170](#),
[173–175](#), [185](#), [186](#), [204](#), [207](#), [253](#),
[255](#), [267](#)
- xpose.data-class, [207](#)
- xpose.gam, [7](#), [120](#), [192](#), [193](#), [202](#), [204](#), [207](#),
[253](#)
- xpose.license.citation, [210](#)
- xpose.logTicks, [210](#)
- xpose.multiple.plot, [7](#), [122](#), [143](#), [158](#), [159](#),
[212](#)
- xpose.multiple.plot-class, [213](#)
- xpose.multiple.plot.default, [38](#), [145](#),
[147](#), [213](#), [213](#)
- xpose.panel.bw, [8](#), [23](#), [63](#), [85](#), [88](#), [181](#), [184](#),
[215](#), [237](#), [238](#)
- xpose.panel.default, [10](#), [12](#), [14–16](#), [18](#), [22](#),
[25](#), [27](#), [30](#), [33](#), [36](#), [38](#), [43](#), [47](#), [48](#), [63](#),
[83](#), [84](#), [86](#), [89–91](#), [103–106](#),
[108–110](#), [112](#), [113](#), [115](#), [128](#), [135](#),
[139](#), [149](#), [150](#), [157](#), [158](#), [161](#), [180](#),
[182](#), [218](#), [242](#), [243](#), [251](#), [256](#), [257](#),
[259](#)
- xpose.panel.histogram, [50](#), [79](#), [131](#), [136](#),
[152](#), [160](#), [176](#), [230](#), [246](#)
- xpose.panel.qq, [81](#), [134](#), [138](#), [154](#), [177](#), [232](#),
[249](#)
- xpose.panel.splom, [77](#), [233](#), [252](#)
- xpose.plot.bw, [8](#), [23](#), [63](#), [85](#), [88](#), [181](#), [184](#),
[216](#), [236](#)
- xpose.plot.default, [9](#), [10](#), [12–14](#), [18–22](#),
[24](#), [25](#), [27](#), [29](#), [30](#), [32](#), [33](#), [36](#), [38](#), [41](#),
[44](#), [46–48](#), [63](#), [82–84](#), [86](#), [87](#), [89–91](#),
[103–106](#), [108–110](#), [112](#), [113](#), [115](#),

- 127, 128, 135, 139, 149, 150, 157,
 - 158, 161, 178–180, 182, 183, 211,
 - 239, 256, 259
- xpose.plot.histogram, 50, 76, 79, 82,
129–131, 136, 149, 152, 160, 161,
175, 176, 178, 179, 244, 246
- xpose.plot.qq, 81, 132–134, 138, 154, 177,
233, 247, 249
- xpose.plot.splom, 77, 235, 249
- xpose.prefs-class, 252
- xpose.print, 7, 40, 63, 67, 75, 93, 95, 97,
116, 117, 124, 125, 145, 147, 157,
164, 166, 168, 170, 174, 175, 186,
206, 253, 267
- xpose.read(import.graph.par), 123
- xpose.string.print, 254
- xpose.VPC, 7, 9, 10, 12, 14, 15, 17, 18, 20, 22,
24, 26, 28, 30, 33, 38, 42, 43, 45, 47,
49, 50, 53, 55, 57, 77, 80, 81, 83–85,
87, 88, 90, 91, 93, 98, 100, 102,
104–106, 108, 109, 111, 112, 114,
115, 122, 128, 131, 134, 135,
137–139, 143, 147, 149, 151, 153,
154, 158, 160, 162, 165, 167, 172,
176, 177, 180, 182–184, 222, 254,
261, 262, 264
- xpose.VPC.both, 7, 9, 10, 12, 14, 15, 17, 18,
20, 22, 24, 26, 28, 30, 33, 38, 42, 43,
45, 47, 49, 50, 53, 55, 57, 77, 80, 81,
83–85, 87, 88, 90, 91, 93, 98, 100,
102, 104–106, 108, 109, 111, 112,
114, 115, 122, 128, 131, 134, 135,
137–139, 143, 147, 149, 151, 153,
154, 158, 160, 162, 165, 167, 172,
176, 177, 180, 182–184, 259, 260,
264
- xpose.VPC.categorical, 7, 9, 10, 12, 14, 15,
17, 18, 20, 22, 24, 26, 28, 30, 33, 38,
42, 43, 45, 47, 49, 50, 53, 55, 57, 77,
80, 81, 83–85, 87, 88, 90, 91, 93, 98,
100, 102, 104–106, 108, 109, 111,
112, 114, 115, 122, 128, 131, 134,
135, 137–139, 143, 147, 149, 151,
153, 154, 158, 160, 162, 165, 167,
172, 176, 177, 180, 182–184, 259,
261, 262, 262
- xpose.write(export.graph.par), 115
- xpose.xscale.components.log10
 - (xpose.logTicks), 210
- xpose.yscale.components.log10
 - (xpose.logTicks), 210
- xpose4, 7, 265
- xpose4-package, 5
- xpPage(gof), 120
- xscale.components, 211
- xscale.components.default, 240
- xsubset, 7, 36, 40, 48, 63, 67, 75, 93–95, 97,
116, 117, 124, 125, 127, 130, 133,
145, 147, 156, 157, 164, 166, 168,
170, 171, 174, 175, 186, 206, 238,
241, 245, 248, 250, 253, 256, 266
- xsubset<- (xsubset), 266
- xvardef, 25, 61, 186, 253, 267
- xvardef<- (xvardef), 267
- xyplot, 10, 12, 14, 18, 20, 22, 25, 27, 30, 33,
36, 38, 41, 45, 47, 48, 54, 76, 82, 84,
87, 90, 91, 103–106, 108–110,
112–115, 128, 135, 139, 145–147,
149, 158, 161, 179, 180, 183, 216,
222, 227, 234, 235, 241–243, 245,
246, 248, 251, 264
- yscale.components.default, 240