Package ‘vpc’

May 8, 2026

Title Create Visual Predictive Checks

Version 1.2.4

Date 2026-02-10

Maintainer Ron Keizer <ron@insight-rx.com>

Description Visual predictive checks are a commonly used diagnostic plot in pharmacometrics, showing how certain statistics (percentiles) for observed data compare to those same statistics for data simulated from a model. The package can generate VPCs for continuous, categorical, censored, and (repeated) time-to-event data.

Depends R (>= 3.1.0)

Imports classInt, dplyr, ggplot2, MASS, readr, rlang, stringr, survival, tidyr

License MIT + file LICENSE

LazyData true

URL https://github.com/ronkeizer/vpc

Suggests covr, knitr, testthat (>= 3.0.0)

RoxygenNote 7.3.3

Encoding UTF-8

Config/testthat/edition 3

NeedsCompilation no

Author Ron Keizer [aut, cre],
Bill Denney [aut] (ORCID: <https://orcid.org/0000-0002-5759-428X>)

Repository CRAN

Date/Publication 2026-02-17 07:20:02 UTC

Contents

add_noise 3
add_sim_index_number 3
add_stratification 4
as_vpcdb 4

auto_bin . . . 5
bin_data . . . 5
calc_pred_corr_continuous . . . 6
calc_vpc_continuous . . . 7
check_stratification_columns_available . . . 8
compute_kaplan . . . 8
compute_kmmc . . . 9
create_vpc_theme . . . 9
define_bins . . . 10
define_data_columns . . . 11
define_loq . . . 11
draw_params_mvr . . . 12
filter_dv . . . 12
format_vpc_input_data . . . 13
ggplot.vpcdb_continuous . . . 14
ggplot2_label_variable . . . 15
label_vpc_tte . . . 16
loq_frac . . . 16
new_vpc_theme . . . 17
pk_iv_1cmt . . . 18
pk_oral_1cmt . . . 19
plot_vpc . . . 20
pred_correction_core . . . 21
quantile_cens . . . 21
read_table_nm . . . 22
read_vpc . . . 23
replace_list_elements . . . 24
rtte_obs_nm . . . 24
rtte_sim_nm . . . 25
show_default . . . 25
simple_data . . . 25
standardize_column . . . 26
theme_empty . . . 27
theme_plain . . . 27
triangle_to_full . . . 27
vpc . . . 28
vpc_cat . . . 30
vpc_cens . . . 33
vpc_ggplot . . . 35
vpc_tte . . . 37

Index **40**

add_noise — *Add noise / residual error to data*

Description

Add noise / residual error to data

Usage

```
add_noise(x, ruv = list(proportional = 0, additive = 0, exponential = 0))
```

Arguments

x	data
ruv	list describing the magnitude of errors. List arguments: "proportional", "additive", "exponential".

Examples

```
library(dplyr)
ipred <- c(10, 8, 6, 4, 2, 0) %>% add_noise(ruv = list(proportional = 0.1, additive = 0.2))
```

add_sim_index_number — *Add sim index number*

Description

Add simulation index number to simulation when not present

Usage

```
add_sim_index_number(sim, id = "id", sim_label = "sim")
```

Arguments

sim	a data.frame containing the simulation data
id	character specifying the column name in the data.frame
sim_label	label to indicate simulation index (if available)

add_stratification *Adds stratification to data set*

Description

Adds stratification to data set

Usage

```
add_stratification(dat, stratify, verbose = FALSE)
```

Arguments

dat	An input data.frame or similar object
stratify	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
verbose	verbosity (‘TRUE‘ or ‘FALSE‘)

as_vpcdb *Create a vpcdb object, and standardize parameter checking*

Description

Create a vpcdb object, and standardize parameter checking

Usage

```
as_vpcdb(..., type = NULL, facet = NULL, scales = NULL, labeller = NULL)
```

Arguments

...	Extra parameters (not checked) added to the object
type	The type of vpc (e.g. "continuous", "categorical", "censored", or "time-to-event")
facet	either "wrap", "columns", or "rows"
scales	Are scales shared across all facets (the default, `"fixed"`), or do they vary across rows (`"free_x"`), columns (`"free_y"`), or both rows and columns (`"free"`)?
labeller	ggplot2 labeller function to be passed to underlying ggplot object

Value

A vpcdb object which is simply a named list with some of the values checked for correctness

auto_bin — *Calculate appropriate bin separators for vpc*

Description

This function calculates bin separators either using R's native binning approaches available in the classInt library such as 'kmeans', 'jenks', 'pretty' etc. Alternatively, a custom approach is available which is based on finding the nadirs in the density functions for the independent variable. Default approach is k-means clustering.

Usage

```
auto_bin(dat, type = "kmeans", n_bins = 8, verbose = FALSE, ...)

## S3 method for class 'numeric'
auto_bin(dat, type = "kmeans", n_bins = 8, verbose = FALSE, ...)

## S3 method for class 'data.frame'
auto_bin(dat, type = "kmeans", n_bins = 8, verbose = FALSE, ...)
```

Arguments

dat	data frame
type	auto-binning type: "density", "time", or "data"
n_bins	number of bins to use; either a positive integer or "auto". For "density" the function might not return a solution with the exact number of bins.
verbose	show debugging information (TRUE or FALSE)
...	arguments passed on to underlying binning functions

Value

A vector of bin separators

bin_data — *Function to bin data based on a vector of bin separators, e.g. for use in VPC*

Description

Function to bin data based on a vector of bin separators, e.g. for use in VPC

Usage

```
bin_data(x, bins = c(0, 3, 5, 7), idv = "time", labeled = F)
```

Arguments

`x`	data
`bins`	numeric vector specifying bin separators
`idv`	variable in the data specifies the independent variable (e.g. "time")
`labeled`	whether a labeled factor instead of integers should be returned

---

`calc_pred_corr_continuous` *Perform prediction-correction*

Description

Perform prediction-correction

Usage

```
calc_pred_corr_continuous(
  sim,
  obs,
  pred_corr,
  pred_corr_lower_bnd,
  cols,
  verbose
)
```

Arguments

`sim`	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
`obs`	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
`pred_corr`	perform prediction-correction?
`pred_corr_lower_bnd`	lower bound for the prediction-correction
`cols`	A length 2, named list with one element named "obs" and the other named "sim", each containing a sub-list with elements for mapping columns names in the data to expected column names for use.
`verbose`	show debugging information (TRUE or FALSE)

Value

A list with "sim" and "obs" (with `pred_corr` performed, if requested)

calc_vpc_continuous *Calculate aggregate statistics for simulated and observed VPC data*

Description

Calculate aggregate statistics for simulated and observed VPC data

Usage

```
calc_vpc_continuous(sim, obs, loq, pi, ci, stratify, bins, bin_mid, verbose)
```

Arguments

`sim`	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
`obs`	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
`loq`	The list output from `define_loq()`
`pi`	simulated prediction interval to plot. Default is c(0.05, 0.95),
`ci`	confidence interval to plot. Default is (0.05, 0.95)
`stratify`	character vector of stratification variables.
`bins`	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
`bin_mid`	either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
`verbose`	show debugging information (TRUE or FALSE)

Value

A list with "vpc_dat" and "aggr_obs"

check_stratification_columns_available *Check whether stratification columns are available*

Description

Check whether stratification columns are available

Usage

```
check_stratification_columns_available(data, stratify, type = "observation")
```

Arguments

data	‘data.frame‘ with observation or simulation data (or NULL to skip checking)
stratify	vector of stratification columns (or NULL to skip checking)
type	either ‘observation‘ or ‘simulation‘

Value

TRUE or raise an error about the missing columns

compute_kaplan *Compute Kaplan-Meier statistics*

Description

Compute Kaplan-Meier statistics

Usage

```
compute_kaplan(
  dat,
  strat = "strat",
  reverse_prob = FALSE,
  rtte_conditional = TRUE,
  ci = NULL
)
```

Arguments

`dat`	data.frame with events
`strat`	vector of stratification variables
`reverse_prob`	reverse the probability (i.e. return ‘1-probability‘)?
`rtte_conditional`	‘TRUE‘ (default) or ‘FALSE‘. Compute the probability for each event newly (‘TRUE‘), or calculate the absolute probability (‘FALSE‘, i.e. the "probability of a 1st, 2nd, 3rd event etc" rather than the "probability of an event happening").
`ci`	confidence interval to calculate, numeric vector of length 2

`compute_kmmc` *Compute KMMC statistics*

Description

Kaplan-Meier Mean Covariate plots are a simulation-based diagnostic to study the influence of covariates and identify potential model misspecification.

Usage

```
compute_kmmc(dat, strat = NULL, reverse_prob = FALSE, kmmc = "DOSE")
```

Arguments

`dat`	data.frame with events
`strat`	vector of stratification variables
`reverse_prob`	reverse the probability (i.e. return ‘1-probability‘)?
`kmmc`	variable to create the KMMC plot for.

`create_vpc_theme` *Create new vpc theme*

Description

Create new vpc theme

Usage

```
create_vpc_theme(...)
```

Arguments

`...`	pass arguments to ‘new_vpc_theme‘

Value

The vpc theme

`define_bins` *Define bins for many types of data*

Description

Define bins for many types of data

Usage

```
define_bins(obs, sim, bins, n_bins, verbose = FALSE)

define_bins_tte(obs, sim, bins, n_bins, kmmc, verbose = FALSE)
```

Arguments

`obs`	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
`sim`	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
`bins`	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
`n_bins`	when using the "auto" binning method, what number of bins to aim for
`verbose`	show debugging information (TRUE or FALSE)
`kmmc`	either NULL (for regular TTE vpc, default), or a variable name for a KMMC plot (e.g. "WT")

Value

A list with named elements: "bins", the bin separator values; "labeled", are the bins labeled?; "obs", binned observed data; "sim", binned simulated data. Additionally, "tmp_bins" is added for tte data.

Functions

- `define_bins_tte()`: Define bins for time-to-event data

define_data_columns *Define data column defaults for various softwares*

Description

Define data column defaults for various softwares

Usage

```
define_data_columns(sim, obs, sim_cols, obs_cols, software_type)
```

Arguments

sim	simulated data
obs	observed data
sim_cols	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
obs_cols	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
software_type	software type, one of ‘nonmem‘, ‘phoenix‘, ‘PKPDsim‘

define_loq *Check and standardize limits of quantification*

Description

Check and standardize limits of quantification

Usage

```
define_loq(
  lloq = NULL,
  uloq = NULL,
  pred_corr = FALSE,
  pred_corr_lower_bnd = 0,
  require_loq
)
```

Arguments

lloq	Number or NULL indicating lower limit of quantification. Default is NULL.
uloq	Number or NULL indicating upper limit of quantification. Default is NULL.
pred_corr	perform prediction-correction?
pred_corr_lower_bnd	lower bound for the prediction-correction
require_loq	Is at least one of lloq or uloq required?

Value

A list with elements "lloq", "uloq", "cens_limit", "pred_corr", "pred_corr_lower_bnd", and "cens_type". "cens_limit" indicates the range for censoring (`c(lloq, uloq)`, `NULL` if neither is given), "cens_type" indicates whether the data are "left" censored (low values are censored, only), "right" (high values are censored, only), "both" (low and high values are censored), or "neither" (no values are censored).

`draw_params_mvr` *Draw parameters from multivariate distribution*

Description

Draw parameters from multivariate distribution

Usage

```
draw_params_mvr(ids, n_sim, theta, omega_mat, par_names = NULL)
```

Arguments

`ids`	vector of id numbers
`n_sim`	number of simulations
`theta`	theta vector
`omega_mat`	omega matrix
`par_names`	parameter names vector

`filter_dv` *Remove values that are not observed values from data*

Description

Remove values that are not observed values from data

Usage

```
filter_dv(x, verbose = FALSE, ...)
```

Arguments

`x`	A data.frame or similar object
`verbose`	show debugging information (TRUE or FALSE)
`...`	Passed to software-specific filtering function

Value

`x` With non-observation rows removed

`format_vpc_input_data` *Prepare VPC data for future calculations by standardizing column names and modifying the input data based on the limits of quantification, stratification, and logarithmic values.*

Description

Prepare VPC data for future calculations by standardizing column names and modifying the input data based on the limits of quantification, stratification, and logarithmic values.

Usage

```
format_vpc_input_data(
  dat,
  cols,
  lloq,
  uloq,
  stratify,
  log_y,
  log_y_min,
  what = "observed",
  verbose = FALSE,
  pred_corr = FALSE
)

format_vpc_input_data_tte(
  dat,
  cols,
  stratify,
  rtte,
  rtte_calc_diff,
  what = "observed",
  verbose = FALSE
)
```

Arguments

`dat`	An input data.frame or similar object
`cols`	A list with an element for colname giving the name for colname in `dat`.
`lloq`	Number or NULL indicating lower limit of quantification. Default is NULL.
`uloq`	Number or NULL indicating upper limit of quantification. Default is NULL.
`stratify`	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
`log_y`	Boolean indicting whether y-axis should be shown as logarithmic. Default is FALSE.

`log_y_min`	minimal value when using log_y argument. Default is 1e-3.
`what`	The description of the data (typically "observed" or "simulated")
`verbose`	show debugging information (TRUE or FALSE)
`pred_corr`	perform prediction-correction?
`rtte`	repeated time-to-event data? Default is FALSE (treat as single-event TTE)
`rtte_calc_diff`	recalculate time (T/F)? When simulating in NONMEM, you will probably need to set this to TRUE to recalculate the TIME to relative times between events (unless you output the time difference between events and specify that as independent variable to the vpc_tte() function.

Value

`dat` modified based on other inputs.

A named list with "dat" modified, as required, and "stratify" with the stratification parameters

Functions

- `format_vpc_input_data_tte()`: Prepare VPC data for future calculations for time-to-event data

---

`ggplot.vpcdb_continuous`

Create a ggplot for each vpcdb type

Description

These functions are not to be called directly by users; they are for internal use. Users should call `plot_vpc()`.

Usage

```
ggplot.vpcdb_continuous(
  data = NULL,
  mapping = NULL,
  ...,
  environment = parent.frame()
)

ggplot.vpcdb_categorical(
  data = NULL,
  mapping = NULL,
  ...,
  environment = parent.frame()
)
```

```
ggplot.vpcdb_censored(
  data = NULL,
  mapping = NULL,
  ...,
  environment = parent.frame()
)

ggplot.vpcdb_time_to_event(
  data = NULL,
  mapping = NULL,
  ...,
  environment = parent.frame()
)
```

Arguments

data	Default dataset to use for plot. If not already a data.frame, will be converted to one by `fortify()`. If not specified, must be supplied in each layer added to the plot.
mapping	Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.
...	Other arguments passed on to methods. Not currently used.
environment	**[Deprecated]** Used prior to tidy evaluation.

---

ggplot2_label_variable

Function copied from ggplot2:::label_variable, with addition of ‘ Copied here since CRAN does not like use of ‘:::‘.

Description

Function copied from ggplot2:::label_variable, with addition of ‘ Copied here since CRAN does not like use of ‘:::‘.

Usage

```
ggplot2_label_variable(labels, multi_line = TRUE)
```

Arguments

labels	Data frame of labels. Usually contains only one element, but faceting over multiple factors entails multiple label variables.
multi_line	Whether to display the labels of multiple factors on separate lines.

label_vpc_tte — *Custom ggplot2 labeller function.*

Description

Slight rewrite of ggplot2::label_both, to make sure that labels for events are ordered appropriately when doing facet_wrap.

Usage

```
label_vpc_tte(labels, multi_line = TRUE, sep = ": ")
```

Arguments

`labels`	Data frame of labels. Usually contains only one element, but faceting over multiple factors entails multiple label variables.
`multi_line`	Whether to display the labels of multiple factors on separate lines.
`sep`	String separating variables and values.

loq_frac — *Calculate fraction of observations below lloq / above uloq*

Description

Calculate fraction of observations below lloq / above uloq

Usage

```
loq_frac(x, limit = 1, cens = c("left", "right", "neither", "both"))
```

Arguments

`x`	A numeric vector
`limit`	censoring limit (ignored if `cens="neither"`)
`cens`	censoring direction

Value

The fraction of observations (`NA` is counted as below/above)

`new_vpc_theme` *Create a customized VPC theme*

Description

Create a customized VPC theme

Usage

```
new_vpc_theme(update = NULL)
```

Arguments

`update`	list containing the plot elements to be updated. Run ‘new_vpc_theme()‘ with no arguments to show an overview of available plot elements.

Details

This function creates a theme that customizes how the VPC looks, i.e. colors, fills, transparencies, linetypes, sizes, etc. The following arguments can be specified in the input list:

obs_color color for observations points

obs_size size for observation points

obs_median_color color for median observation line

obs_median_linetype linetype for median observation line

obs_median_linewidth linewidth for median observation line

obs_ci_fill color for observation CI fill

obs_ci_color color for observation CI lines

obs_ci_linetype linetype for observation CI lines

obs_ci_linewidth linewidth for observations CI lines

sim_pi_fill fill color for simulated prediction interval areas

sim_pi_alpha transparency for simulated prediction interval areas

sim_pi_color color for simulated prediction interval lines

sim_pi_linetype linetype for simulated prediction interval lines

sim_pi_linewidth linewidth for simulated prediction interval lines

sim_median_fill fill color for simulated median area

sim_median_alpha transparency for simulated median area

sim_median_color color for simulated median line

sim_median_linetype linetype for simulated median line

sim_median_linewidth linewidth for simulated median line

bin_separators_color color for bin separator lines, NA for don’t plot

bin_separators_location where to plot bin separators ("t" for top, "b" for bottom)

loq_color color of line showing limit of quantification

Value

A list with vpc theme specifiers

Examples

```
theme1 <- new_vpc_theme(update = list(
  obs_color = "red",
  obs_ci_color = "#aa0000",
  obs_alpha = .3,
  sim_pi_fill = "#cc8833",
  sim_pi_linewidth = 2
))
vpc(simple_data$sim, simple_data$obs, vpc_theme = theme1)
```

---

pk_iv_1cmt *Simulate PK data from a 1-compartment iv model*

Description

Simulate PK data from a 1-compartment iv model

Usage

```
pk_iv_1cmt(
  t,
  t_inf = 1,
  tau = 24,
  dose = 120,
  CL = 0.345,
  Vc = 1.75,
  ruv = NULL
)
```

Arguments

t	Time after dose
t_inf	Infusion length
tau	Dosing interval
dose	Dose
CL	Clearance
Vc	Volume of distribution
ruv	Residual variability

Value

A vector of predicted values, with or without added residual variability

Examples

```
dat1 <- vpc:::pk_iv_1cmt(t = c(0:72), tau = 24, dose = 120,
                   CL = 5, Vc = 50)
dat2 <- vpc:::pk_iv_1cmt(t = c(0:72), tau = 24, dose = 120,
                   CL = 5, Vc = 50,
                   ruv = list(proportional = 0.1, additive = 0.1))
```

---

pk_oral_1cmt *Simulate PK data from a 1-compartment oral model*

Description

Simulate PK data from a 1-compartment oral model

Usage

```
pk_oral_1cmt(t, tau = 24, dose = 120, ka = 1, ke = 1, cl = 10, ruv = NULL)
```

Arguments

t	Time after dose
tau	Dosing interval
dose	Dose
ka	Absorption rate
ke	Elimination rate
cl	Clearance
ruv	Residual variability

Value

A vector of predicted values, with or without added residual variability

Examples

```
dat1 <- vpc:::pk_oral_1cmt(t = c(0:72), tau = 24, dose = 120,
                    ka = 1, ke = 1, cl = 10)
dat2 <- vpc:::pk_oral_1cmt(t = c(0:72), tau = 24, dose = 120,
                    ka = 1, ke = 1, cl = 10,
                    ruv = list(proportional = 0.1, additive = 0.1))
```

plot_vpc — *VPC plotting function*

Description

This function performs no parsing of data, it just plots the already calculated statistics generated using one of the ‘vpc‘ functions.

Usage

```
plot_vpc(
  db,
  show = NULL,
  vpc_theme = NULL,
  smooth = TRUE,
  log_x = FALSE,
  log_y = FALSE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  verbose = FALSE
)
```

Arguments

db	object created using the ‘vpc‘ function
show	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
log_x	Boolean indicting whether x-axis should be shown as logarithmic. Default is FALSE.
log_y	Boolean indicting whether y-axis should be shown as logarithmic. Default is FALSE.
xlab	label for x axis
ylab	label for y axis
title	title
verbose	show debugging information (TRUE or FALSE)

See Also

vpc_cens, vpc_tte, vpc_cat

Examples

```
## See vpc.ronkeizer.com for more documentation and examples

library(vpc)
vpc_db <- vpc(sim = simple_data$sim, obs = simple_data$obs, vpcdb = TRUE)
plot_vpc(vpc_db, title = "My new vpc", x = "Custom x label")
```

---

pred_correction_core *Core prediction correction function*

Description

Perform pred-correction for predictions that were non-zero and were not missing observation. The latter can happen e.g. when censored data is set to NA in ‘format_vpc_input_data()‘.

Usage

```
pred_correction_core(data, pred_col, pred_corr_lower_bnd)
```

Arguments

data	dataset, either ‘sim‘ or ‘obs‘ data.frame
pred_col	colsobspred
pred_corr_lower_bnd	lower bound for the prediction-correction

Value

data.frame

---

quantile_cens *Calculate quantiles respecting the censored data*

Description

Calculate quantiles respecting the censored data

Usage

```
quantile_cens(
  x,
  probs = 0.5,
  limit = 1,
  cens = c("left", "right", "neither", "both")
)
```

Arguments

x	numeric vector whose sample quantiles are wanted, or an object of a class for which a method has been defined (see also 'details'). NA and NaN values are not allowed in numeric vectors unless na.rm is TRUE.
probs	numeric vector of probabilities with values in $[0, 1]$. (Values up to '2e-14' outside that range are accepted and moved to the nearby endpoint.)
limit	censoring limit (ignored if cens="neither")
cens	censoring direction

Value

The quantile of x treating NA values as censored

---

read_table_nm *NONMEM output table import function*

Description

Quickly import NONMEM output tables into R. Function taken from 'modelviz' package by Benjamin Guiastrennec. When both skip and header are NULL, read_nmtab will automatically detect the optimal settings to import the tables. When more than one files are provided for a same NONMEM run, they will be combined into a single data.frame.

Usage

```
read_table_nm(
  file = NULL,
  skip = NULL,
  header = NULL,
  rm_duplicates = FALSE,
  nonmem_tab = TRUE
)
```

Arguments

file	full file name
skip	number of lines to skip before reading data
header	logical value indicating whether the file contains the names of the variables as its first line
rm_duplicates	logical value indicating whether duplicated columns should be removed
nonmem_tab	logical value indicating to the function whether the file is a table or a nonmem additional output file.

Value

A data.frame

Examples

```
## Not run:
data <- read_table_nm(file = '../models/pk/sdtab101')

## End(Not run)
```

---

read_vpc *Read in VPC data*

Description

Read in VPC data

Usage

```
read_vpc(sim, obs, psn_folder, software, sim_cols, obs_cols, verbose = FALSE)
```

Arguments

sim	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
obs	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
psn_folder	instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
software	name of software platform using (e.g. nonmem, phoenix)
sim_cols	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
obs_cols	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
verbose	show debugging information (TRUE or FALSE)

Value

A list with names of "sim", "obs", "software", and "cols"

replace_list_elements *Replace list elements by name*

Description

Replace list elements by name

Usage

```
replace_list_elements(list, replacement)
```

Arguments

list	original list
replacement	replacement list

Details

Finds and replaces list elements by name and throws an error if an element is not available in the original list. This is a local duplicate of the PKPDmisc copy for the VPC package to reduce dependency on PKPDmisc at this time.

Examples

```
## Not run:
list <- list(ipred = "ipred", dv = "dv", idv = "idv", "pred" = "pred")
replacement <- list(dv = "conc", idv = "time")
list <- replace_list_elements(list, replacement)

## End(Not run)
```

rtte_obs_nm *Simulated RTTE data (1x)*

Description

An example dataset with simulated repeated time-to-event data

Usage

```
rtte_obs_nm
```

Format

An object of class `data.frame` with 573 rows and 6 columns.

rtte_sim_nm — *Simulated RTTE data (100x)*

Description

An example dataset with simulated repeated time-to-event data (100 simulations)

Usage

```
rtte_sim_nm
```

Format

An object of class `data.frame` with 2000000 rows and 7 columns.

show_default — *Defaults for show argument with various endpoint types*

Description

Defaults for show argument with various endpoint types

Usage

```
show_default
```

Format

An object of class `list` of length 4.

simple_data — *A small rich dataset*

Description

A small rich dataset

Usage

```
simple_data
```

Format

An object of class `list` of length 2.

Details

a list containing the obs and sim data for an example dataset to run a simple vpc.

Examples

```
## Not run:
vpc(simple_data$sim, simple_data$obs)

## End(Not run)
```

---

standardize_column — *Confirm that a column is in the data, and rename data to prepare that column for later use.*

Description

Confirm that a column is in the data, and rename data to prepare that column for later use.

Usage

```
standardize_column(dat, cols, colname, coldesc, what, default)
```

Arguments

dat	An input data.frame or similar object
cols	A list with an element for colname giving the name for colname in dat.
colname	The name of the column (character scalar)
coldesc	The description of the column (character scalar)
what	The description of the data (typically "observed" or "simulated")
default	A default value (scalar or vector) to use if the column is not found.

Value

If colname is already named colname in dat, dat unchanged. If not, check if dat has that column name already, and if so, name the existing dat[[colname]] to dat[[paste0(colname, ".old")]] and then rename cols[[colname]] to colname.

theme_empty — *Empty ggplot2 theme*

Description

Empty ggplot2 theme

Usage

```
theme_empty()
```

Examples

```
vpc(simple_data$sim, simple_data$obs) + theme_empty()
```

theme_plain — *A nicer default theme for ggplot2*

Description

A nicer default theme for ggplot2

Usage

```
theme_plain()
```

Examples

```
vpc(simple_data$sim, simple_data$obs) + theme_plain()
```

triangle_to_full — *Lower to full triangle*

Description

Convert the lower triangle of a covariance matrix to a full matrix object

Usage

```
triangle_to_full(vect)
```

Arguments

vect	the lower triangle of a covariance matrix

vpc *VPC function*

Description

Creates a VPC plot from observed and simulation data

Usage

```
vpc(sim, ...)

## Default S3 method:
vpc(sim, ...)

vpc_vpc(
  sim = NULL,
  obs = NULL,
  psn_folder = NULL,
  bins = "jenks",
  n_bins = "auto",
  bin_mid = "mean",
  obs_cols = NULL,
  sim_cols = NULL,
  software = "auto",
  show = NULL,
  stratify = NULL,
  pred_corr = FALSE,
  pred_corr_lower_bnd = 0,
  pi = c(0.05, 0.95),
  ci = c(0.05, 0.95),
  uloq = NULL,
  lloq = NULL,
  log_y = FALSE,
  log_y_min = 0.001,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  smooth = TRUE,
  vpc_theme = NULL,
  facet = "wrap",
  scales = "fixed",
  labeller = NULL,
  vpcdb = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

`sim`	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
`...`	Other arguments sent to other methods (like xpose or nlmixr); Note these arguments are not used in the default vpc and are ignored by the default method.
`obs`	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
`psn_folder`	instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
`bins`	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
`n_bins`	when using the "auto" binning method, what number of bins to aim for
`bin_mid`	either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
`obs_cols`	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
`sim_cols`	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
`software`	name of software platform using (e.g. nonmem, phoenix)
`show`	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
`stratify`	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
`pred_corr`	perform prediction-correction?
`pred_corr_lower_bnd`	lower bound for the prediction-correction
`pi`	simulated prediction interval to plot. Default is c(0.05, 0.95),
`ci`	confidence interval to plot. Default is (0.05, 0.95)
`uloq`	Number or NULL indicating upper limit of quantification. Default is NULL.
`lloq`	Number or NULL indicating lower limit of quantification. Default is NULL.
`log_y`	Boolean indicting whether y-axis should be shown as logarithmic. Default is FALSE.
`log_y_min`	minimal value when using log_y argument. Default is 1e-3.
`xlab`	label for x axis
`ylab`	label for y axis
`title`	title
`smooth`	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.

vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet	either "wrap", "columns", or "rows"
scales	Are scales shared across all facets (the default, "fixed"), or do they vary across rows ("free_x"), columns ("free_y"), or both rows and columns ("free")?
labeller	ggplot2 labeller function to be passed to underlying ggplot object
vpcdb	Boolean whether to return the underlying vpcdb rather than the plot
verbose	show debugging information (TRUE or FALSE)

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

vpc_cens, vpc_tte, vpc_cat

Examples

```
## See vpc.ronkeizer.com for more documentation and examples
library(vpc)

# Basic commands:
vpc(sim = simple_data$sim, obs = simple_data$obs)
vpc(sim = simple_data$sim, obs = simple_data$obs, lloq = 20)
```

---

vpc_cat *VPC function for categorical*

Description

Creates a VPC plot from observed and simulation data for categorical variables.

Usage

```
vpc_cat(
  sim = NULL,
  obs = NULL,
  psn_folder = NULL,
  bins = "jenks",
  n_bins = "auto",
  bin_mid = "mean",
  obs_cols = NULL,
  sim_cols = NULL,
  software = "auto",
  show = NULL,
```

```
  ci = c(0.05, 0.95),
  uloq = NULL,
  lloq = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  smooth = TRUE,
  vpc_theme = NULL,
  facet = "wrap",
  labeller = NULL,
  vpcdb = FALSE,
  verbose = FALSE
)
```

Arguments

sim	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
obs	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
psn_folder	instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
bins	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
n_bins	when using the "auto" binning method, what number of bins to aim for
bin_mid	either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
obs_cols	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
sim_cols	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
software	name of software platform using (e.g. nonmem, phoenix)
show	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
ci	confidence interval to plot. Default is (0.05, 0.95)
uloq	Number or NULL indicating upper limit of quantification. Default is NULL.
lloq	Number or NULL indicating lower limit of quantification. Default is NULL.
xlab	label for x axis
ylab	label for y axis
title	title

smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet	either "wrap", "columns", or "rows"
labeller	ggplot2 labeller function to be passed to underlying ggplot object
vpcdb	boolean whether to return the underlying vpcdb rather than the plot
verbose	show debugging information (TRUE or FALSE)

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

vpc, vpc_tte, vpc_cens

Examples

```
## See vpc.ronkeizer.com for more documentation and examples
library(vpc)

# simple function to simulate categorical data for single individual
sim_id <- function(id = 1) {
  n <- 10
  logit <- function(x) exp(x) / (1+exp(x))
  data.frame(id = id, time = seq(1, n, length.out = n),
             dv = round(logit((1:n) - n/2 + rnorm(n, 0, 1.5))) )
}
## simple function to simulate categorical data for a trial
sim_trial <- function(i = 1, n = 20) { # function to simulate categorical data for a trial
  data.frame(sim = i, do.call("rbind", lapply(1:n, sim_id)))
}

## simulate single trial for 20 individuals
obs <- sim_trial(n = 20)

## simulate 200 trials of 20 individuals
sim <- do.call("rbind", lapply(1:200, sim_trial, n = 20))

## Plot categorical VPC
vpc_cat(sim = sim, obs = obs)
```

vpc_cens *VPC function for left- or right-censored data (e.g. BLOQ data)*

Description

Creates a VPC plot from observed and simulation data for censored data. Function can handle both left- (below lower limit of quantification) and right-censored (above upper limit of quantification) data.

Usage

```
vpc_cens(
  sim = NULL,
  obs = NULL,
  psn_folder = NULL,
  bins = "jenks",
  n_bins = 8,
  bin_mid = "mean",
  obs_cols = NULL,
  sim_cols = NULL,
  software = "auto",
  show = NULL,
  stratify = NULL,
  stratify_color = NULL,
  ci = c(0.05, 0.95),
  uloq = NULL,
  lloq = NULL,
  xlab = "Time",
  ylab = "Probability of <LOQ",
  title = NULL,
  smooth = TRUE,
  vpc_theme = NULL,
  facet = "wrap",
  labeller = NULL,
  vpcdb = FALSE,
  verbose = FALSE
)
```

Arguments

sim	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
obs	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm

psn_folder	instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
bins	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
n_bins	when using the "auto" binning method, what number of bins to aim for
bin_mid	either "mean" for the mean of all timepoints (default) or "middle" to use the average of the bin boundaries.
obs_cols	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
sim_cols	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
software	name of software platform using (e.g. nonmem, phoenix)
show	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
stratify	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
stratify_color	variable to stratify and color lines for observed data. Only 1 stratification variables can be supplied.
ci	confidence interval to plot. Default is (0.05, 0.95)
uloq	Number or NULL indicating upper limit of quantification. Default is NULL.
lloq	Number or NULL indicating lower limit of quantification. Default is NULL.
xlab	label for x axis
ylab	label for y axis
title	title
smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet	either "wrap", "columns", or "rows"
labeller	ggplot2 labeller function to be passed to underlying ggplot object
vpcdb	boolean whether to return the underlying vpcdb rather than the plot
verbose	show debugging information (TRUE or FALSE)

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

vpc, vpc_tte, vpc_cat

Examples

```
## See vpc.ronkeizer.com for more documentation and examples
library(vpc)

vpc_cens(sim = simple_data$sim, obs = simple_data$obs, lloq = 30)
vpc_cens(sim = simple_data$sim, obs = simple_data$obs, uloq = 120)
```

---

vpc_ggplot *A collection of internal ggplot helpers for VPC plotting*

Description

A collection of internal ggplot helpers for VPC plotting

Usage

```
geom_bin_sep(bins, show, vpc_theme)

geom_hline_loq(data, vpc_theme)

geom_obs_cens_dat_tte(data)

geom_obs_ci_continuous(data, show, vpc_theme)

geom_obs_ci_tte(data, show, vpc_theme)

geom_obs_dv_continuous(data, show, vpc_theme)

geom_obs_km(data)

geom_obs_median_continuous(data, show, vpc_theme)

geom_sim_km(data, show)

geom_sim_median_continuous(data, show, vpc_theme)

geom_sim_median_tte(data, show, smooth)

geom_sim_median_ci_continuous(data, show, smooth, vpc_theme)

geom_sim_pi_as_area_continuous(data, show, smooth, vpc_theme)

geom_sim_pi_as_area_tte(data, show, smooth, vpc_theme)

geom_sim_pi_continuous(data, show, vpc_theme)
```

```
geom_sim_pi_ci_continuous(data, show, smooth, vpc_theme, what = c("q5", "q95"))

facet_continuous(data)

facet_tte(data)

guides_stratify_color(data)

scale_x_log10_vpc(data, show)

scale_y_log10_vpc(show)
```

Arguments

bins	Numeric vector of bin separators (if logical value, geom_blank is returned)
show	Should the geom be shown? (TRUE/FALSE)
vpc_theme	The theme to use
data	The vpcdb object
smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
what	Which interval should be shown (low = "q5" or high = "q95")

Functions

- `geom_bin_sep()`: Show rug plot of bin separators
- `geom_hline_loq()`: Generate hlines for the lloq/uloq
- `geom_obs_cens_dat_tte()`: Show the cens_dat for time-to-events
- `geom_obs_ci_continuous()`: Show confidence interval for observed, continuous data
- `geom_obs_ci_tte()`: Show confidence interval for observed, time-to-event data
- `geom_obs_dv_continuous()`: Show observed data points
- `geom_obs_km()`: Show observed Kaplan-Meier line
- `geom_obs_median_continuous()`: Show observed median line for continuous data
- `geom_sim_km()`: Show simulated Kaplan-Meier curves for time-to-event data
- `geom_sim_median_continuous()`: Show simulated median line for continuous data
- `geom_sim_median_tte()`: Show simulated median line (or steps) for time-to-event data
- `geom_sim_median_ci_continuous()`: Show simulated median confidence interval for continuous data
- `geom_sim_pi_as_area_continuous()`: Show simulated prediction interval area for continuous data
- `geom_sim_pi_as_area_tte()`: Show simulated prediction interval area for time-to-event data
- `geom_sim_pi_continuous()`: Show simulated prediction interval lines for continuous data

- `geom_sim_pi_ci_continuous()`: Show simulated prediction interval confidence interval areas for continuous data
- `facet_continuous()`: Facet continuous data by stratification factors
- `facet_tte()`: Facet time-to-event data by stratification factors
- `guides_stratify_color()`: Generate guides for stratification fill and colour
- `scale_x_log10_vpc()`: Optionally show log-x scale
- `scale_y_log10_vpc()`: Optionally show log-y scale

`vpc_tte` *VPC function for time-to-event (survival) data*

Description

This function can be used for either single time-to-event (TTE) or repeated time-to-event (RTTE) data.

Usage

```
vpc_tte(
  sim = NULL,
  obs = NULL,
  psn_folder = NULL,
  rtte = FALSE,
  rtte_calc_diff = TRUE,
  rtte_conditional = TRUE,
  events = NULL,
  bins = FALSE,
  n_bins = 10,
  software = "auto",
  obs_cols = NULL,
  sim_cols = NULL,
  kmmc = NULL,
  reverse_prob = FALSE,
  stratify = NULL,
  stratify_color = NULL,
  ci = c(0.05, 0.95),
  xlab = "Time",
  ylab = "Survival (%)",
  show = NULL,
  as_percentage = TRUE,
  title = NULL,
  smooth = FALSE,
  vpc_theme = NULL,
  facet = "wrap",
  labeller = NULL,
```

```
  verbose = FALSE,
  vpcdb = FALSE
)
```

Arguments

sim	this is usually a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm. However it can also be an object like a nlmixr or xpose object
obs	a data.frame with observed data, containing the independent and dependent variable, a column indicating the individual, and possibly covariates. E.g. load in from NONMEM using read_table_nm
psn_folder	instead of specifying "sim" and "obs", specify a PsN-generated VPC-folder
rtte	repeated time-to-event data? Default is FALSE (treat as single-event TTE)
rtte_calc_diff	recalculate time (T/F)? When simulating in NONMEM, you will probably need to set this to TRUE to recalculate the TIME to relative times between events (unless you output the time difference between events and specify that as independent variable to the vpc_tte() function.
rtte_conditional	‘TRUE‘ (default) or ‘FALSE‘. Compute the probability for each event newly (‘TRUE‘), or calculate the absolute probability (‘FALSE‘, i.e. the "probability of a 1st, 2nd, 3rd event etc" rather than the "probability of an event happening").
events	numeric vector describing which events to show a VPC for when repeated TTE data, e.g. c(1:4). Default is NULL, which shows all events.
bins	either "density", "time", or "data", "none", or one of the approaches available in classInterval() such as "jenks" (default) or "pretty", or a numeric vector specifying the bin separators.
n_bins	when using the "auto" binning method, what number of bins to aim for
software	name of software platform using (e.g. nonmem, phoenix)
obs_cols	list for mapping observation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
sim_cols	list for mapping simulation data columns, e.g. ‘list(dv = "DV", id = "ID", idv = "TIME", pred="PRED")‘
kmmc	either NULL (for regular TTE vpc, default), or a variable name for a KMMC plot (e.g. "WT")
reverse_prob	reverse the probability scale (i.e. plot 1-probability)
stratify	character vector of stratification variables. Only 1 or 2 stratification variables can be supplied.
stratify_color	character vector of stratification variables. Only 1 stratification variable can be supplied, cannot be used in conjunction with ‘stratify‘.
ci	confidence interval to plot. Default is (0.05, 0.95)
xlab	label for x axis

ylab	label for y axis
show	what to show in VPC (obs_dv, obs_ci, pi, pi_as_area, pi_ci, obs_median, sim_median, sim_median_ci)
as_percentage	Show y-scale from 0-100 percent? TRUE by default, if FALSE then scale from 0-1.
title	title
smooth	"smooth" the VPC (connect bin midpoints) or show bins as rectangular boxes. Default is TRUE.
vpc_theme	theme to be used in VPC. Expects list of class vpc_theme created with function vpc_theme()
facet	either "wrap", "columns", or "rows"
labeller	ggplot2 labeller function to be passed to underlying ggplot object
verbose	show debugging information (TRUE or FALSE)
vpcdb	Boolean whether to return the underlying vpcdb rather than the plot

Details

Creates a VPC plot from observed and simulation survival data

Value

a list containing calculated VPC information (when vpcdb=TRUE), or a ggplot2 object (default)

See Also

vpc, vpc_tte, vpc_cens

Examples

```
## See vpc-docs.ronkeizer.com for more documentation and examples.

## Example for repeated) time-to-event data
## with NONMEM-like data (e.g. simulated using a dense grid)

data(rtte_obs_nm)
data(rtte_sim_nm)

# treat RTTE as TTE, no stratification
vpc_tte(sim = rtte_sim_nm[rtte_sim_nm$sim <= 20,],
       obs = rtte_obs_nm,
       rtte = FALSE,
       sim_cols=list(dv = "dv", idv = "t"), obs_cols=list(idv = "t"))
```

Index

∗ **datasets**
rtte_obs_nm, 24
rtte_sim_nm, 25
show_default, 25
simple_data, 25

add_noise, 3
add_sim_index_number, 3
add_stratification, 4
as_vpcdb, 4
auto_bin, 5

bin_data, 5

calc_pred_corr_continuous, 6
calc_vpc_continuous, 7
check_stratification_columns_available, 8
compute_kaplan, 8
compute_kmmc, 9
create_vpc_theme, 9

define_bins, 10
define_bins_tte *(define_bins)*, 10
define_data_columns, 11
define_loq, 11
draw_params_mvr, 12

facet_continuous *(vpc_ggplot)*, 35
facet_tte *(vpc_ggplot)*, 35
filter_dv, 12
format_vpc_input_data, 13
format_vpc_input_data_tte *(format_vpc_input_data)*, 13
fortify(), *15*

geom_bin_sep *(vpc_ggplot)*, 35
geom_hline_loq *(vpc_ggplot)*, 35
geom_obs_cens_dat_tte *(vpc_ggplot)*, 35
geom_obs_ci_continuous *(vpc_ggplot)*, 35
geom_obs_ci_tte *(vpc_ggplot)*, 35
geom_obs_dv_continuous *(vpc_ggplot)*, 35
geom_obs_km *(vpc_ggplot)*, 35
geom_obs_median_continuous *(vpc_ggplot)*, 35
geom_sim_km *(vpc_ggplot)*, 35
geom_sim_median_ci_continuous *(vpc_ggplot)*, 35
geom_sim_median_continuous *(vpc_ggplot)*, 35
geom_sim_median_tte *(vpc_ggplot)*, 35
geom_sim_pi_as_area_continuous *(vpc_ggplot)*, 35
geom_sim_pi_as_area_tte *(vpc_ggplot)*, 35
geom_sim_pi_ci_continuous *(vpc_ggplot)*, 35
geom_sim_pi_continuous *(vpc_ggplot)*, 35
ggplot.vpcdb_categorical *(ggplot.vpcdb_continuous)*, 14
ggplot.vpcdb_censored *(ggplot.vpcdb_continuous)*, 14
ggplot.vpcdb_continuous, 14
ggplot.vpcdb_time_to_event *(ggplot.vpcdb_continuous)*, 14
ggplot2_label_variable, 15
guides_stratify_color *(vpc_ggplot)*, 35

label_vpc_tte, 16
loq_frac, 16

NA, *22*
new_vpc_theme, 17

pk_iv_1cmt, 18
pk_oral_1cmt, 19
plot_vpc, 20
pred_correction_core, 21

quantile_cens, 21

read_table_nm, *6, 7*, *10*, 22, *23*, *29*, *31*, *33*, *38*
read_vpc, 23

replace_list_elements, 24
rtte_obs_nm, 24
rtte_sim_nm, 25

scale_x_log10_vpc *(*vpc_ggplot*)*, 35
scale_y_log10_vpc *(*vpc_ggplot*)*, 35
show_default, 25
simple_data, 25
standardize_column, 26

theme_empty, 27
theme_plain, 27
triangle_to_full, 27

vpc, 28, *32*, *34*, *39*
vpc_cat, *20*, *30*, 30, *34*
vpc_cens, *20*, *30*, *32*, 33, *39*
vpc_ggplot, 35
vpc_tte, *20*, *30*, *32*, *34*, 37, *39*
vpc_vpc *(*vpc*)*, 28