Package ‘iNZightRegression’

May 8, 2026

Type Package

Title Tools for Exploring Regression Models with 'iNZight'

Version 1.3.5

Depends R (>= 4.0)

Imports car, dplyr, GGally, ggplot2, ggrepel, ggtext, graphics, grDevices, grid, iNZightPlots (>= 2.13), multcomp, patchwork, stats, stats4, utils

Suggests covr, broom.helpers, iNZightTools (>= 1.9), survey, survival, testthat

Description Provides a suite of functions to use with regression models, including summaries, residual plots, and factor comparisons. Used as part of the Model Fitting module of 'iNZight', a graphical user interface providing easy exploration and visualisation of data for students of statistics, available in both desktop and online versions.

BugReports https://github.com/iNZightVIT/iNZightRegression/issues

Contact inzight_support@stat.auckland.ac.nz

URL https://inzight.nz

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

Language en-GB

Config/Needs/dependencies tmelliott/surveyspec@develop, iNZightVIT/iNZightTools@1.9, iNZightVIT/iNZightPlots@2.13, iNZightVIT/iNZightMR@2.2.7, rcmdcheck, curl

Config/Needs/coverage tmelliott/surveyspec@develop, iNZightVIT/iNZightTools@1.9, iNZightVIT/iNZightPlots@2.13, iNZightVIT/iNZightMR@2.2.7, covr

NeedsCompilation no

Author Tom Elliott [aut, cre] (ORCID: <https://orcid.org/0000-0002-7815-6318>),
Simon Potter [aut],
David Banks [aut],
Danny Chang [ctb]

Maintainer Tom Elliott <`tom.elliott@auckland.ac.nz`>

Repository CRAN

Date/Publication 2025-07-29 04:30:09 UTC

Contents

`compare_models` *Compare regression models using AIC and BIC.*

Description

Obtain a quick model comparison matrix for a selection of models

Usage

```
compare_models(x, ...)

## Default S3 method:
compare_models(x, ...)

## S3 method for class 'svyglm'
compare_models(x, ...)
```

Arguments

`x`	a regression model (lm, glm, svyglm, ...)
`...`	other models

Value

an ‘inzmodelcomp‘ object containing model comparison statistics

Methods (by class)

- `compare_models(default)`: default method
- `compare_models(svyglm)`: method for survey GLMs

Author(s)

Tom Elliott

Examples

```
m0 <- lm(Sepal.Length ~ 1, data = iris)
m1 <- lm(Sepal.Length ~ Sepal.Width, data = iris)
m2 <- lm(Sepal.Length ~ Sepal.Width + Species, data = iris)
compare_models(m0, m1, m2)
```

`factorComp` *Compare factor levels*

Description

Computes confidence intervals for the pairwise differences between levels of a factor, based off of `stats::TukeyHSD`.

Usage

```
factorComp(fit, factor)

## S3 method for class 'inzfactorcomp'
print(x, ...)
```

Arguments

`fit`	a lm/glm/svyglm object
`factor`	the name of the factor to compare
`x`	an `inzfactorcomp` object
`...`	extra arguments for print (ignored)

Value

a factor level comparison object with estimates, CIs, and (adjusted) p-values

Functions

- `print(inzfactorcomp)`: print method for object of class `inzfactorcomp`

Author(s)

Tom Elliott

Examples

```
f <- lm(Sepal.Length ~ Sepal.Width + Species, data = iris)
factorComp(f, "Species")
```

`histogramArray` *Histogram Array*

Description

Produces an array of histograms to compare against the histogram of residuals for a fitted linear model.

Usage

```
histogramArray(x, n = 7, env = parent.frame())
```

Arguments

`x`	an `lm` or `svyglm` object.
`n`	the number of additional histograms to plot alongside the original.
`env`	environment for finding data to bootstrap

Details

The histogram of the model `x` appears in the top-left position. For each of the other histograms, the fitted values of `x` are taken and normal random errors are added to these. The normal residual standard errors have standard error equal to the estimated residual standard error of `x`. A model is then fitted to this altered data and a histogram is produced.

Value

No return value, called to generate plot.

Author(s)

David Banks, Tom Elliott

See Also

`iNZightQQplot`

Examples

```
histogramArray(lm(Sepal.Length ~ Sepal.Width + Species, data = iris))
```

`iNZightQQplot` *iNZight QQ Plot*

Description

Produces a sample of QQ-plots based on the fitted values, overlaid by a QQ-plot of the original data.

Usage

```
iNZightQQplot(x, n = 5, env = parent.frame())
```

Arguments

`x`	an `lm` or `svyglm` object (with `family = "Gaussian"`.
`n`	the number of sampled QQ plots to produce beneath the QQ plot of `x`.
`env`	environment for finding data to bootstrap

Details

Multiple bootstrap models are generated from the fitted values of the model, each with different random normal errors with standard error equal to the estimated residual standard error from the original model. These are plotted, and then overlaid by the QQ plot from the original data.

This plot can be used to assess the assumption of normality in the residuals for a linear regression model.

Value

No return value, called to produce plot.

Author(s)

David Banks, Tom Elliott

See Also

`histogramArray`

Examples

```
fit <- lm(Volume ~ Height + Girth, data = trees)
iNZightQQplot(fit)
```

`iNZightSummary` *Informative Summary Information for Regression Models*

Description

The iNZight summary improves upon the base R summary output for fitted regression models. More information is provided and displayed in a more intuitive format. This function both creates and returns a summary object, as well as printing it.

Usage

```
iNZightSummary(
  x,
  method = "standard",
  reorder.factors = FALSE,
  digits = max(3, getOption("digits") - 3),
  symbolic.cor = x$symbolic.cor,
  signif.stars = getOption("show.signif.stars"),
  exclude = NULL,
  exponentiate.ci = FALSE,
  ...
)
```

Arguments

`x`	an object of class `"lm"`, `"glm"` or `"svyglm"`, usually the result of a call to the corresponding function.
`method`	one of either `"standard"` or `"bootstrap"`. If `"bootstrap"`, then bootstrapped estimates and standard errors are calculated; otherwise, uses the standard estimates.
`reorder.factors`	logical, if `TRUE`, and there are factors present in the model, then the most common level of the factor is set to be the baseline.
`digits`	the number of significant digits to use when printing.
`symbolic.cor`	logical, if `TRUE`, print the correlations in a symbolic form (see `symnum`), rather than as numbers.
`signif.stars`	logical, if `TRUE`, 'significance stars' are printed for each coefficient.
`exclude`	a character vector of names of variables to be excluded from the summary output (i.e., confounding variables).
`exponentiate.ci`	logical, if `TRUE`, the exponential of the confidence intervals will be printed if appropriate (log/logit link or log transformed response)
`...`	further arguments passed to and from other methods.

Details

This summary function provides more information in the following ways:

Factor headers are now given. The base level for a factor is also listed with an estimate of 0. This is to make it clear what the base level of a factor is, rather than attempting to work out by deduction from what has already been printed.

The p-value of a factor is now given; this is the output from `Anova`, which calculates the p-value based off of Type III sums of squares, rather than sequentially as done by `anova`.

Each level of a factor is indented by 2 characters for its label and its p-value to distinguish between a factor, and levels of a factor.

The labels for each level of an interaction are now just the levels of the factor (separated by a `.`), rather than being prepended with the factor name also.

Value

An object of class `summary.lm`, `summary.glm`, or `summary.svyglm`.

Note

If any level is not observed in a factor, no p-values will be printed on all factors. This is because we cannot calculate Type III sums of squares when this is the case.

The fitted model currently requires that the data are stored in a dataframe, which is pointed at by the `data` argument to `lm` (or equivalent).

Author(s)

Simon Potter, Tom Elliott.

See Also

The model fitting functions `lm`, `glm`, and `summary`.

The 'survey' package: https://cran.r-project.org/package=survey.

Function `coef` will extract the matrix of coefficients with standard errors, t-statistics and p-values.

To calculate p-values for factors, use `Anova` with type III sums of squares.

Examples

```
m <- lm(Sepal.Length ~ ., data = iris)
iNZightSummary(m)

# exclude confounding variables for which you don't
# need to know about their coefficients:
iNZightSummary(m, exclude = "Sepal.Width")
```

`inzplot` *inzplot method*

Description

inzplot method

Diagnostic Plots for Regression Models

Usage

```
## S3 method for class 'glm'
inzplot(x, ..., env = parent.frame())

## S3 method for class 'lm'
inzplot(
  x,
  which = c("residual", "scale", "leverage", "cooks", "normal", "hist"),
  show.bootstraps = nrow(x$model) < 1e+05,
  label.id = 3L,
  col.smooth = "orangered",
  col.bs = "lightgreen",
  cook.levels = c(0.5, 1),
  col.cook = "pink",
  ...,
  bs.fits = NULL,
  env = parent.frame()
)
```

Arguments

`x`	a regression model
`...`	additional arguments
`env`	the environment for evaluating things (e.g., bootstraps)
`which`	the type of plot to draw
`show.bootstraps`	logical, if `TRUE` bootstrap smoothers will be shown (defaults to `TRUE` if fewer than 100,000 observations)
`label.id`	integer for the number of extreme points to label (with row id)
`col.smooth`	the colour of smoothers
`col.bs`	the colour of bootstrap (smoothers)
`cook.levels`	levels of the Cook's distance at which to draw contours.
`col.cook`	the colour of Cook's distance contours
`bs.fits`	a list of bootstrapped datasets

Value

A ggplot object with a plot method that will show the plot in the graphics device

Functions

- `inzplot(glm)`: Method for GLMs

Plot types

There are several plot types available:

- residual versus fitted
- scale-location
- residual versus leverage
- Cook's distance
- normal Q-Q
- histogram array
- forest plot

Author(s)

Tom Elliott

Examples

```
iris_fit <- lm(Sepal.Width ~ Sepal.Length, data = iris)
inzplot(iris_fit)
inzplot(iris_fit, which = "residual", show.bootstraps = FALSE)
```

`inzsummary` *inzsummary method*

Description

inzsummary method

Summary method for linear models

Usage

```
## S3 method for class 'lm'
inzsummary(x, ..., env = parent.frame())
```

Arguments

`x`	an `lm`, `glm`, or `svyglm` object
`...`	additional arguments passed to `iNZightSummary`
`env`	the environment for evaluating things (e.g., bootstraps)

Value

An object of class `summary.lm`, `summary.glm`, or `summary.svyglm`.

See Also

iNZightSummary

`partialResPlot` *Partial residual plot of continuous variable*

Description

This function draws partial residual plots for a continuous explanatory variables in a given model.

Usage

```
partialResPlot(
  fit,
  varname,
  showBootstraps = nrow(fit$model) >= 30 & nrow(fit$model) < 4000,
  use.inzightplots = FALSE,
  env = parent.frame()
)

allPartialResPlots(fit, ...)
```

Arguments

`fit`	an `lm`, `glm` or `svyglm` object.
`varname`	character, the name of an explanatory variable in the model
`showBootstraps`	logical, if `TRUE`, bootstrap smoothers will overlay the graph. By default this is `TRUE` if there are between 30 and 4000 observations in the model, otherwise it is `FALSE`.
`use.inzightplots`	logical, if `TRUE`, the iNZightPlots package will be used for plotting.
`env`	environment where the data is stored for bootstrapping
`...`	additional arguments passed to ‘partialResPlot‘

Value

No return value, called for side-effect of producing a plot.

Functions

- `allPartialResPlots()`: Cycle through all partial residual plots

Author(s)

David Banks, Tom Elliott.

Examples

```
m <- lm(Sepal.Length ~ Sepal.Width + Petal.Width, data = iris)
partialResPlot(m, "Sepal.Width")


allPartialResPlots(lm(Sepal.Length ~ Sepal.Width + Petal.Width, data = iris))
```

`plotlm6` *Extended Plot Diagnostics for (g)lm Models*

Description

These plots are an extension of the original plots provided by `plot.lm`.

Six plots are currently available: residuals versus fitted, Scale-Location of $\sqrt{|residuals|}$ against fitted values, residuals against leverages, Cook's distance, Normal Q-Q plot and histogram of residuals.

Also provided is the summary plot which shows all diagnostic plots arranged in a 2 by 3 grid. By default, this is shown first, then each of the individual plots in turn.

Usage

```
plotlm6(
  x,
  which = 1:6,
  panel = if (add.smooth) panel.smooth else points,
  sub.caption = NULL,
  main = "",
  ask = prod(par("mfcol")) < length(which) && dev.interactive(),
  id.n = 3,
  labels.id = names(residuals(x)),
  cex.id = 0.75,
  qqline = TRUE,
  cook.levels = c(0.5, 1),
  add.smooth = getOption("add.smooth", TRUE),
  label.pos = c(4, 2),
  cex.caption = 1,
  showBootstraps = nrow(x$model) >= 30 && nrow(x$model) < 4000,
  use.inzightplots = FALSE,
  env = parent.frame(),
  ...
)
```

Arguments

`x`	an `lm` object, typically the result of `lm` or `glm`. Can also take `svyglm` objects.
`which`	numeric, if a subset of the plots is required, specify a subset of the numbers `1:6`. `7` will produce a summary plot showing all of the plots arranged in a a grid. `1:6` will show the summary plot followed by each of the single plots one by one (default).
`panel`	panel function. the useful alternative to `points`, `panel.smooth` can be chosen by `add.smooth = TRUE`.
`sub.caption`	common title. Above the figures if there are more than one; used as `sub` (s.`title`) otherwise. If `NULL`, as by default, a possible abbreviated version of `deparse(x$call)` is used.
`main`	title to each plot, in addition to `caption`.
`ask`	logical, if `TRUE`, the user is *ask*ed before each plot, see `par(ask=.)`. Ignored when only one plot is being shown.
`id.n`	number of points to be labelled in each plot, starting with the most extreme.
`labels.id`	vector of labels, from which the labels for extreme plots will be chosen. `NULL` uses observation numbers.
`cex.id`	magnification of point labels.
`qqline`	logical, if `TRUE`, a `qqline()` is added to the normal QQ plot.
`cook.levels`	levels of the Cook's distance at which to draw contours.
`add.smooth`	logical, if `TRUE`, a smoother is drawn to the appropriate plots; see also `panel` above.
`label.pos`	positioning of labels, for the left half and right half of the graph respectively, for plots 1–3.
`cex.caption`	controls the size of `caption`.
`showBootstraps`	logical, if `TRUE`, bootstrap loess smoothers are drawn in the first 4 plots. By default, only drawn for sample sizes of at least 30.
`use.inzightplots`	logical, if set to `TRUE`, the iNZightPlots package will be used for plotting, rather than base R graphics.
`env`	environment for performing bootstrap simulations (i.e., to find the dataset!)
`...`	other arguments to be passed to through to plotting functions.

Details

For the residuals versus fitted values plot, we add bootstrapped smoothers to illustrate variance. The smoother is also added to the Scale-Location plot.

The Normal Q-Q and histogram plots are taken from the `normcheck` function in the `s20x` package.

Value

No return value; called for the side-effect of producing a plot.

Author(s)

Simon Potter, David Banks, Tom Elliott.

See Also

`histogramArray`, `iNZightQQplot`

Examples

```
m <- lm(Sepal.Length ~ Sepal.Width + Petal.Width, data = iris)
plotlm6(m, which = 1)

# the summary grid:
plotlm6(m, which = 7)

# the default cycles through all 6 plots
plotlm6(m)
```

`Poly` *Polynomial Matrix*

Description

A modified ‘poly()‘ function that allows for missing values.

Usage

```
Poly(x, degree = 1, coefs = NULL, raw = FALSE, ...)
```

Arguments

`x`	variable to convert to matrix
`degree`	degree of polynomial
`coefs`	pass to poly() function
`raw`	pass to poly() function
`...`	more arguments for the poly() function

Details

Credit goes to whoever posted this online first (google search if you must find it!)

Value

a matrix, with NAs in the missing rows

Author(s)

Tom Elliott

Examples

```
Poly(rnorm(100), degree = 2L)

# handles missing values:
iris.na <- iris
iris.na$Sepal.Length[c(5, 10)] <- NA
lm(Sepal.Width ~ Poly(Sepal.Length, 2L), data = iris.na)

# stats::poly() produces an error in this case:
# lm(Sepal.Width ~ poly(Sepal.Length, 2L), data = iris.na)
```

Index