Package ‘factorplot’

May 8, 2026

Type Package

Title Presenting Pairwise Comparisons

Version 1.3

Depends ggplot2, ggtext, multcomp

Suggests nnet, carData

Description The tools herein calculate, print, summarize and plot pairwise differences that result from generalized linear models, general linear hypothesis tests and multinomial logistic regression models. For more information, see Armstrong (2013) <doi:10.32614/RJ-2013-021>.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 7.3.2

Encoding UTF-8

NeedsCompilation no

Author Dave Armstrong [aut, cre]

Maintainer Dave Armstrong <`davearmstrong.ps@gmail.com`>

Repository CRAN

Date/Publication 2025-11-01 22:40:15 UTC

Contents

`factorplot` *Calculate Pairwise Differences*

Description

This function calculates all pairwise difference from the input data. The input data can be the result of a GLM (produced with `glm`), a multinomial logit model (produced with `multinom` from the **nnet** package), a general linear hypothesis test (produced with `glht` from the **multcomp** package), an object of class `eff` from the `effects` package or any vector of values and a corresponding variance-covariance matrix.

Usage

```
factorplot(obj, adjust.method = "none", ...)

## S3 method for class 'glm'
factorplot(
  obj,
  adjust.method = "none",
  order = "natural",
  factor.variable = NULL,
  pval = 0.05,
  two.sided = TRUE,
  ...
)

## S3 method for class 'lm'
factorplot(
  obj,
  adjust.method = "none",
  order = "natural",
  factor.variable = NULL,
  pval = 0.05,
  two.sided = TRUE,
  ...
)

## S3 method for class 'summary.glht'
factorplot(obj, ...)

## S3 method for class 'glht'
factorplot(obj, adjust.method = "none", pval = 0.05, ...)

## S3 method for class 'sims'
factorplot(obj, adjust.method = "none", order = "natural", pval = 0.05, ...)

## Default S3 method:
```

```
factorplot(
  obj,
  adjust.method = "none",
  order = "natural",
  var,
  resdf = Inf,
  pval = 0.05,
  two.sided = TRUE,
  ...
)

## S3 method for class 'eff'
factorplot(
  obj,
  adjust.method = "none",
  order = "natural",
  pval = 0.05,
  two.sided = TRUE,
  ordby = NULL,
  ...
)

## S3 method for class 'multinom'
factorplot(
  obj,
  adjust.method = "none",
  order = "natural",
  variable,
  pval = 0.05,
  two.sided = TRUE,
  ...
)
```

Arguments

`obj`	An object of class `glm` or `lm`, `glht`, `summary.glht`, `multinom` or a vector of values (of class `numeric`) for which pairwise differences will be calculated. Alternatively, an object of class `sims` which gives an Nsim x Nstimulus matrix of predictions from which differences will be calculated.
`adjust.method`	For objects of class `multinom` and `numeric` - one of the methods allowed by `p.adjust` in **stats** - ‘holm’, ‘hochberg’, ‘hommel’, ‘bonferroni’, ‘BH’, ‘BY’, ‘fdr’, ‘none’. See help for the `p.adjust` for more information on these different adjustment methods. For objects of class `glm`, `lm` or `glht`, additional arguments of ‘single-step’, ‘Shaffer’, ‘Westfall’ and ‘free’ are possible. See `glht` from the **multcomp** package for details.
`...`	Additional arguments to be passed to `summary.glht`, including, but not limited to `level` and `alternative`.
`order`	One of ‘natural’, ‘alph’, or ‘size’ indicating how the levels of the factor should

	be ordered for presentation. The 'natural' option (the default) leaves the levels as they are in the factor contrasts. 'alph' sorts the levels alphabetically and 'size' sorts the levels by size of coefficient.
`factor.variable`	String containing the name of the factor for which pairwise coefficient differences will be calculated (if a `glm` or `lm` class object is passed to the function)
`pval`	The (uncorrected) Type I error probability required, default = 0.05
`two.sided`	Logical argument indicating whether the hypothesis test should be against a two-sided alternative if TRUE (default) or a one-sided alternative if FALSE
`var`	Variance-covariance matrix to be used if `obj` is a numeric vector. This could also be a vector of quasi/floating variances from which a diagonal variance-covariance matrix will be produced
`resdf`	Residual degrees of freedom used as the degrees of freedom for the t-distribution from which p-values will be generated if `obj` is a numeric vector
`ordby`	For objects of class `eff` with interactions, `ordby` is a string indicating the variable by which the plot should be ordered.
`variable`	String containing the name of the column of the model matrix for which pairwise differences will be calculated if a `multinom` class object is passed to the function

Details

This function calculates pairwise differences that can be passed to a novel plotting method that does not suffer from some of the same problems as floating/quasi confidence intervals and is easier to apprehend immediately than a compact letter display.

While the factorplot function and its print and summary methods work equally well regardless of the number of levels in the `factor.variable`, the plot function automatically scales the resulting graph to the appropriate size, but will be less useful as the number of contrasts gets large (e.g., > 30). If more than one factor covariate is present and the `factor.variable` option is NULL, the function generates a text-based menu in the R GUI that will allow the users to pick the term for which they want to calculate the results.

Value

`b.diff`	An upper-triangular matrix of pairwise differences between row and column levels of the factor
`b.sd`	An upper-triangular matrix of standard errors of the pairwise differences represented in b.diff
`pval`	An upper-triangular matrix of uncorrected (one-sided) p-values corresponding to the entries of b.diff
`est`	The values of the estimates used in the calculations.
`p`	The p-value specified in the command

Author(s)

Dave Armstrong

References

Easton, D.F., J. Peto and G.A.G. Babiker. 1991. Floating absolute risk: An alternative to relative risk in survival and case control analysis avoiding an arbitrary reference group. *Statistics in Medicine* **10**: 1025–1035.
Firth, David and Renee X. de Menzes. 2004. Quasi-variances. *Biometrika* **91.1**: 65–80.
Plummer, M. 2004. Improved estimates of floating absolute risk. *Statistics in Medicine* **23**: 93–104.

Examples

```
## for lm/glm
x <- as.factor(round(runif(1000, .5,5.5)))
levels(x) <- paste("lab", 1:20, sep="")
X <- model.matrix(~x)
Y <- X %*% rnorm(ncol(X),0,4) + rnorm(1000)
mod <- lm(Y ~ x)
fp <- factorplot(mod, factor.variable="x",  pval = 0.05, order="alph")

## for glht
library(multcomp)
mod.glht <- glht(mod, linfct = mcp('x' = 'Tukey'))
fp2 <- factorplot(mod.glht, adjust.method='single-step')

## for vector of values
b <- c(0, mod$coef[-1])
v <- rbind(0, cbind(0, vcov(mod)[-1,-1]))
names(b) <- colnames(v) <- rownames(v) <- mod$xlevels[["x"]]
fp3 <- factorplot(b, var=v, resdf=mod$df.residual)

## for multinomial logit
data(france)
library(nnet)
multi.mod <- multinom(vote ~ retnat + lrself + male + age, data=france)
fp4 <- factorplot(multi.mod, variable="lrself")
```

`fp_to_df` *Convert factorplot output to data frame*

Description

Converts the output from factorplot to a data frame that is convenient for custom plotting. The plot method for factorplot objects from the factorplot package will produce a plot that is lightly customisable. However, for more control over the appearance of the plot, and to plot estimates on the diagnoal of the display, returning the data and making a plot is easier.

Usage

```
fp_to_df(obj, type = c("upper_tri", "both_tri"), ...)
```

Arguments

`obj`	An object of class ‘factorplot‘ produced by the ‘factorplot()‘ function from the ‘factorplot‘ package.
`type`	Indicates whether you want the resulting plot to have differences on the upper triangle and p-values on the lower triangle (if ‘"both_tri"‘) or both p-values and differences to be plotted in the upper triangle.
`...`	Other arguments, currently ignored.

Value

A data frame with columns - ‘row‘: The name of the parameter in the row - ‘col‘: The name of the parameter in the column - ‘estimate‘: The estimate for the parameter (on the diagonal) - ‘difference‘: Pairwise differences between parameters. - ‘p_value‘: The p-value of the pairwise difference.

Examples

```
library(factorplot)
library(ggplot2)
data(chickwts)
mod <- lm(weight ~ feed, data=chickwts)
fp <- factorplot(mod, factor.variable = "feed", order="size")
chick_df <- fp_to_df(fp, type="upper_tri")
ggplot(chick_df, aes(x=row, y=column)) +
  geom_tile(aes(fill=difference), color="black") +
  geom_text(aes(label = ifelse(p_value < .05, "*", "")), color="white", size=10) +
  scale_fill_viridis_c(option = "D", na.value = "transparent") +
  theme_classic() +
  geom_text(data=chick_df,
            aes(x=row, y=column, label=round(estimate, 2))) +
  labs(fill="Difference", x="", y="")
```

`france` *Example data for factorplot function*

Description

A subset of data from the 1994 Eurobarometer for France

Format

A data frame with 542 observations on the following 5 variables.

lrself respondent’s left-right self-placement on a 1(left)-10(right) scale

male a dummy variable coded 1 for males and 0 for females

age respondent’s age

vote a factor indicating vote choice with levels PCF, PS, Green, RPR and UDF

retnat a factor indicating the respondent’s retrospective national economic evaluation with levels Better, Same and Worse

References

Reif, Karlheinz and Eric Marlier. 1997. *Euro-barometer 42.0: The First Year of the New European Union, November-December 1994*. Inter-university Consortium for Political and Social Research (ICPSR) [distributor].

`plot.factorplot` *Plot method for objects of class factorplot*

Description

Creates a plot akin to an upper-triangular levelplot (though using `plot` rather than `levelplot`) where the coloring of the squares represents significance and text inside the squares represents the pairwise difference and its correspopnding standard error.

Usage

```
## S3 method for class 'factorplot'
plot(
  x,
  ...,
  abbrev.char = 100,
  print.est = TRUE,
  print.se = TRUE,
  text.nudge = 0.1,
  text.args = list(),
  remove.caption = FALSE
)
```

Arguments

`x`	An object of class factorplot, produced by `factorplot`.
`...`	Other arguments to be passed to plot, currently not implemented
`abbrev.char`	The number of characters that should be used to abbreviate the levels of the factor. Set to a large value for unabbreviated names.
`print.est`	logical argument indicating whether the estimates should be printed in the boxes
`print.se`	logical argument indicating whether the standard errors should be printed in the boxes
`text.nudge`	Scalar giving the value the estimate text will be moved in the positive direction on y and the standard error text will be moved in the negative direction on y.
`text.args`	A list of other arguments to be passed to ‘geom_text()‘, such as ‘size‘ or ‘color‘.
`remove.caption`	Logical indicating whether the caption should be removed from the plot.

Value

`a graph`	For m categories, the plot returns an m-1 x m-1 matrix where the nexus of the row and column values represent the pairwise differences between the row and column values along with the standard error of the difference on the linear scale (unless a transformation is performed).

Author(s)

Dave Armstrong

See Also

`factorplot`

Examples

```
est1 <- log(c(1.00,2.12,1.44,1.31,1.44,
     1.46,0.90))
var1 <- c(0.242,0.096,0.156,0.140,
     0.380,0.484,0.375)^2
names(est1) <- c(
     "Normal & superficial gastritis",
     "Chronic gastritis",
     "Chronic atrophic gastritits",
     "Intestinal metaplasia I",
     "Intestinal metaplasia II",
     "Intestinal metaplasia III",
     "Dysplasia")

plummer_fp1 <- factorplot(est1, var=var1, resdf=Inf)
plot(plummer_fp1, trans="exp", abbrev.char = 100)
```

`print.factorplot` *Print method for objects of class factorplot*

Description

Prints the output from an object of class `factorplot`. By default, the function prints all pairwise differences along with standard errors and p-values (optionally adjusted for multiple testing). Optionally, it can print only significant differences.

Usage

```
## S3 method for class 'factorplot'
print(x, ..., digits = 3, sig = FALSE, trans = NULL)
```

Arguments

`x`	An object of class `factorplot`.
`...`	Other arguments passed to print, currently not implemented
`digits`	The number of digits to print in each column
`sig`	Logical indicating whether only significant differences should be printed.
`trans`	A character string representing the post-hypothesis-testing transformation to be performed on the estimates. For example, if the estimates provided to the `factorplot` command are log-floating absolute risks, you could use the transformation 'exp'. The transformation is performed through a call to `do.call`

Value

`Printed output`	The printed output shows the difference between all pairs of stimuli (i.e., levels of the factor) along with their standard errors and (optionally adjusted) p-values. If a transformation is implemented, the difference is transformed accordingly, but the standard errors and other values are on the linear scale.

Author(s)

Dave Armstrong

See Also

`factorplot`

Examples

```
est1 <- log(c(1.00,2.12,1.44,1.31,1.44,
     1.46,0.90))
var1 <- c(0.242,0.096,0.156,0.140,
     0.380,0.484,0.375)^2
names(est1) <- c(
     "Normal & superficial gastritis",
     "Chronic gastritis",
     "Chronic atrophic gastritits",
     "Intestinal metaplasia I",
     "Intestinal metaplasia II",
     "Intestinal metaplasia III",
     "Dysplasia")
plummer_fp1 <- factorplot(est1, var=var1, resdf=Inf)
print(plummer_fp1, trans="exp")
```

`squares` *Auxiliary Function to Plot a Square*

Description

An auxiliary function to plot squares, used by the `plot.factorplot` function

Usage

```
squares(ll, width = 1, col)
```

Arguments

`ll`	The (x,y) coordinate of the lower-left corder of the square
`width`	a scalar indicating how wide the squares should be
`col`	a color with which the square will be filled in

Details

This is a function called by `plot.factorplot` and not intended to be directly used by the user; however, it is possible that this could be of more general use as a utility. The function is simply a wrapper to `polygon` that obviates the need to specify all (x,y) coordinates for the polygon.

Value

`square`	A square is printed on the graph, but nothing else is returned

Author(s)

Dave Armstrong

`summary.factorplot` *Summary method for objects of class factorplot*

Description

Summarizes the number of significant positive and negative differences for objects of class `factorplot`

Usage

```
## S3 method for class 'factorplot'
summary(object, ...)
```

Arguments

`object`	An object of class `factorplot`
`...`	Other arguments passed to summary, currently not implemented

Value

`Printed Output`	The printed output summarises the number of stimuli that are significantly higher or lower and not significantly different from each other.

Author(s)

Dave Armstrong

See Also

`factorplot`

Examples

```
x <- as.factor(round(runif(1000, .5,5.5)))
levels(x) <- paste("lab", 1:20, sep="")
X <- model.matrix(~x)
b <- rnorm(ncol(X),0,4)
Y.hat <- X %*% b
Y <- Y.hat  + rnorm(1000)
mod <- lm(Y ~ x)
fp <- factorplot(mod, factor.variable="x", pval=0.05, order="alph")
summary(fp)
```

Index