Package 'factorplot'

October 13, 2022

Type Package

Title Graphical Presentation of Simple Contrasts

Version 1.1-2

Date 2016-06-22

Author Dave Armstrong (Department of Political Science, UW-Milwaukee)

Maintainer Dave Armstrong <dave@quantoid.net>

Imports multcomp, nnet

Description Methods to calculate, print, summarize and plot pairwise differences from GLMs, GLHT or Multinomial Logit models.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2016-06-23 00:57:36

R topics documented:

`factorplot-package` *factorplot*

Description

Factorplot is a way to summarize and plot information from categorical predictors from linear models and GLMs. It creates all simple contrasts and analytical standard errors for those contrasts.

Details

Package:	factorplot
Type:	Package
Version:	1.0
Date:	2010-07-20
License:	GPL (>=2)
LazyLoad:	yes

After a linear model or GLM has been estimated, the factorplot command creates all pairwise differences among the levels (including the reference category) of the indicated factor as well as their associated standard errors to facilitate hypothesis testing directly. The print method prints the pairwise difference, standard error, p-value and Bonferroni-corrected p-value. The summary method prints the number of significant positive/negative pairwise differences. The plot method makes something akin to an upper-triangular levelplot that indicates whether differences are positive/negative and statistically significant.

Author(s)

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

References

Armstrong, David A., II. 2013. factorplot: Improving Presentation of Simple Contrasts in Generalized Linear Models. *The R Journal* **5(2)**: 4–15.

`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

```
## S3 method for class 'glm'
factorplot(obj, adjust.method="none", order="natural",
    factor.variable=NULL, pval=.05, two.sided=TRUE, ...)
## S3 method for class 'lm'
factorplot(obj, adjust.method="none", order="natural",
    factor.variable=NULL, pval=.05, two.sided=TRUE, ...)
## S3 method for class 'glht'
factorplot(obj, adjust.method="none", pval=.05, ...)
## S3 method for class 'summary.glht'
factorplot(obj, ...)
## S3 method for class 'multinom'
factorplot(obj, adjust.method="none", order="natural",
    variable, pval = .05, two.sided=TRUE, ...)
## S3 method for class 'eff'
factorplot(obj, adjust.method="none", order="natural",
    pval=0.05, two.sided=TRUE, ordby = NULL, ...)
## Default S3 method:
factorplot(obj, adjust.method="none", order="natural",
    var, resdf, 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.
`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)
`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
`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
`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
`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.
`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.

`ordby` For objects of class `eff` with interactions, `ordby` is a string indicating the variable by which the plot should be ordered.

`...` Additional arguments to be passed to `summary.glht`, including, but not limited to `level` and `alternative`.

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

`p` The p-value specified in the command

Author(s)

Dave Armstrong (Department of Political Science, UW-Milwaukee)

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")
```

`france` *Example data for factorplot function*

Description

A subset of data from the 1994 Eurobarometer for France

Usage

```
data(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=10, polycol=NULL,
textcol=NULL, trans=NULL, print.sig.leg=TRUE, print.square.leg=TRUE,
scale.text=1, space.text=1, print.est = TRUE, print.se=TRUE)
```

Arguments

`x`	An object of class factorplot, produced by `factorplot`.
`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.
`polycol`	A vector of three colors indicating the colors of polygons when the difference is significant negative, insignificant, and significant positive, in that order. Defaults to c('gray80', 'white', 'gray40').
`textcol`	A vector of three colors indicating the text color for polygons that are significant negative, insignificant, and significant positive, in that order. Defaults to c('black', 'black', 'white')
`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`
`print.sig.leg`	logical indicating whether the legend identifying the meaning of the different colors should be included.
`print.square.leg`	logical indicating whether the legend identifying the meaning of the numbers in each square should be included.
`scale.text`	optional scale factor to be applied to text, numbers bigger than 1 make text bigger than default and numbers smaller than 1 do the opposite
`space.text`	optional text spacing factor, numbers bigger than 1 push text toward the extent of the boxes and numbers smaller than one bring text in toward the center
`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
`...`	Other arguments to be passed to plot, currently not implemented

Value

a graph

Author(s)

Dave Armstrong (Department of Political Science, UW-Milwaukee)

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.
digits	The number of digits to print in each column
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
sig	Logical indicating whether only significant differences should be printed.
...	Other arguments passed to print, currently not implemented

Author(s)

Dave Armstrong (Department of Political Science, UW-Milwaukee)

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 (UW-Milwaukee, Department of Political Science)

---

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

Author(s)

Dave Armstrong (Department of Political Science, UW-Milwaukee)

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