

# Package ‘predictmeans’

April 15, 2020

**Type** Package

**Title** Calculate Predicted Means for Linear Models

**Version** 1.0.4

**Date** 2020-04-14

**Author** Dongwen Luo, Siva Ganesh and John Koolaard

**Maintainer** Dongwen Luo <dongwen.luo@agresearch.co.nz>

**Description** Providing functions to diagnose and make inferences from various linear models, such as those obtained from 'aov', 'lm', 'glm', 'gls', 'lme', and 'lmer'. Inferences include predicted means and standard errors, contrasts, multiple comparisons, permutation tests and graphs.

**License** GPL (>= 2)

**Depends** R (>= 3.5.0), lme4, nlme, parallel

**Imports** ggplot2, graphics, grDevices, Matrix, methods, numDeriv, pbkrtest, plyr, stats, utils

**URL** <https://CRAN.R-project.org/package=predictmeans>

**Repository** CRAN

**Date/Publication** 2020-04-15 05:40:03 UTC

**NeedsCompilation** no

## R topics documented:

predictmeans-package . . . . .	2
anovalmer . . . . .	2
contrastmeans . . . . .	3
CookD . . . . .	5
covariatemeans . . . . .	6
Kmatrix . . . . .	7
permanova.lmer . . . . .	8
permindex . . . . .	9
permlmer . . . . .	10
permmodels . . . . .	11

PMplot . . . . .	13
predictmeans . . . . .	14
residplot . . . . .	17
varcomp.lmer . . . . .	18

<b>Index</b>	<b>19</b>
--------------	-----------

---

predictmeans-package    *Calculate Predicted Means for Linear Models*

---

## Description

This package provides functions to diagnose and make inferences from various linear models, such as those obtained from 'aov', 'lm', 'glm', 'gls', 'lme', and 'lmer'. Inferences include predicted means and standard errors, contrasts, multiple comparisons, permutation tests and graphs.

## Details

Package: predictmeans  
 Type: Package  
 Version: 1.0.1  
 Date: 2018-08-23  
 License: GPL (>= 2)

## Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

Maintainer: Dongwen Luo <dongwen.luo@agresearch.co.nz>

## References

Welham, S., Cullis, B., Gogel, B., Gilmour, A., & Thompson, R. (2004), *Prediction in linear mixed models*, Australian and New Zealand Journal of Statistics, 46(3), 325-347.

---

anovalmer                      *ANOVA of a Linear Mixed Effects Model produced by lmer function*

---

## Description

This function provides ANOVA of a linear mixed effects model for lmer object with model terms' denominate degree of freedom estimated by Kenward-Roger method.

**Usage**

```
anova1mer(model, DDF=NULL)
```

**Arguments**

`model` Model object returned by `lmer`.  
`DDf` Model terms' denominate degree of freedom specified by user.

**Value**

ANOVA table.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**Examples**

```
## Not run for simplifying process of submitting pkg to CRAN
library(predictmeans)
Oats$nitro <- factor(Oats$nitro)
fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
anova1mer(fm)
```

---

contrastmeans

*Linear Contrast Tests for a Linear Model*

---

**Description**

Performs t-tests (or permuted t-tests) of specified contrasts for linear models obtained from functions `aov`, `lm`, `glm`, `gls`, `lme`, or `lmer`.

**Usage**

```
contrastmeans(model, modelterm, ctrmatrix, ctrnames=NULL, adj="none", Df, permlist)
```

**Arguments**

`model` Model object returned by `aov`, `lm`, `glm`, `gls`, `lme`, and `lmer`.  
`modelterm` Name (in "quotes") for indicating which factor term's contrast to be calculated. The `modelterm` must be given exactly as it appears in the printed model, e.g. "A" or "A:B".  
`ctrmatrix` A specified contrast matrix. If `ctrmatrix` is missing, the programe will ask user to enter it.  
`ctrnames` Names of the specified contrasts, e.g. `c("A vs D", "C vs B", ...)`

adj	Name (in "quote") for indicating a method for adjusting p-values of pairwise comparisons. The choices are "none", "tukey", "holm", "hochberg", "hommel", "bonferroni", "BH", "BY" and "fdr". The default method is "none".
Df	A denominator degree of freedom for modelterm. (For glmer models the Df needs to be specified, while for the other models, Df is obtained from the fitted model automatically).
permlist	A model parameter list containing nsim parameters produced by the function permmodels. When permlist != NULL, the option Df will be non-functional. This is a key option for the permutation test.

### Value

There are two components in the output which are

Table	A table showing t-test results for the specified linear contrasts.
K	A contrast matrix.

### Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

### References

Torsten Hothorn, Frank Bretz and Peter Westfall (2008), *Simultaneous Inference in General Parametric Models*. *Biometrical*, Journal 50(3), 346–363.

### Examples

```
library(predictmeans)
# ftable(xtabs(yield ~ Block+Variety+nitro, data=Oats))
Oats$nitro <- factor(Oats$nitro)
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
# library(lme4)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)

## Not run:
## The contrast has a contrast matrix as follows:
#   0:Golden Rain 0:Marvellous 0:Victory
#[1,]          -1           0           1
#[2,]           0           0           1
#   0.2:Golden Rain 0.2:Marvellous 0.2:Victory
#[1,]           0           0           0
#[2,]           0           0           0
#   0.4:Golden Rain 0.4:Marvellous 0.4:Victory
#[1,]           0           0           0
#[2,]           0          -1           0
#   0.6:Golden Rain 0.6:Marvellous 0.6:Victory
#[1,]           0           0           0
#[2,]           0           0           0

# 1. Enter above contrast matrix into a pop up window, then close the window
```

```
# contrastmeans(fm, "nitro:Variety")

# 2. Construct the contrast matrix directly
cm <- rbind(c(-1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
            c(0, 0, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0))
contrastmeans(fm, "nitro:Variety", ctrmatrix=cm)
```

---

CookD

*Calculates and plots Cook's distances for a Linear (Mixed) Model*


---

### Description

This function produces Cook's distance plots for a linear model obtained from functions `aov`, `lm`, `glm`, `gls`, `lme`, or `lmer`.

### Usage

```
CookD(model, group=NULL, plot=TRUE, idn=3, newwd=TRUE)
```

### Arguments

<code>model</code>	Model object returned by <code>aov</code> , <code>lm</code> , <code>glm</code> , <code>gls</code> , <code>lme</code> , and <code>lmer</code> .
<code>group</code>	Name (in "quotes") for indicating how observations are deleted for Cook's distance calculation. If <code>group!=NULL</code> then deletions will be along levels of group variable, otherwise, will be along individual observations.
<code>plot</code>	A logical variable; if it is true, a plot of Cook's distance will be presented. The default is TRUE.
<code>idn</code>	An integer indicating the number of top Cook's distances to be labelled in the plot. The default value is 3.
<code>newwd</code>	A logical variable to indicate whether to print graph in a new window. The default value is TRUE.

### Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

### Examples

```
library(predictmeans)
Oats$nitro <- factor(Oats$nitro)
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
# library(lme4)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
CookD(fm)
```

---

covariatemmeans                      *Predicted Means of a Linear Model with Covariate Variable(s)*

---

### Description

This function obtains predicted means with graph for a new set of covariate values.

### Usage

```
covariatemmeans(model, modelterm=NULL, covariate, as.is=FALSE, covariateV=NULL,
  level=0.05, Df=NULL, trans=NULL, transOff=0, responsen=NULL, trellis=TRUE,
  plotord=NULL, mtitle=NULL, ci=TRUE, point=TRUE, jitterv=0, newwd=TRUE)
```

### Arguments

model	Model object returned by aov, lm, glm, gls, lme, and lmer.
modelterm	Name (in "quotes") for indicating which factor term's predicted mean to be calculated. The modelterm must be given exactly as it appears in the printed model, e.g. "A" or "A:B".
covariate	Name (in "quotes") of one the covariate variables in the model.
as.is	A logic value to specify wheather or not using original covariate values, the default is FALSE.
covariateV	A numeric vector when as.is is FALSE, then covariatemmeans will produce the result for covariate at value of covariateV.
level	A significant level for calculating confident interval. The default value is 0.05.
Df	A degree of freedom for calculating CI of predicted means (you can manually specified Df here). For the above models, Df is obtained from the function automatically.
trans	A function object for calculating the back transformed means, e.g. trans=exp.
transOff	When you use trans=exp(x+1), then transOff=1, the default is 0.
responsen	Name (in "quotes") of the back transformed response variable in the model.
trellis	A logical variable. If set to TRUE (default), a trellis plots of predicted means with CI will be drawn.
plotord	A numeric vector specifying the order of plotting for two or three way interaction (e.g. plotord = c(2,1,3) will put the second variable in modelterm on the X axis, the first variable as the grouping variable, and the third one as the panel variable). The defaults are c(1,2) and c(1,2,3) for two and three way interactions.
mtitle	The main title in the graph.
ci	A logical variable to indicate whether to print confidence interval. The default value is TRUE.
point	A logical variable to indicate whether to print raw data points. The default value is TRUE.

jitterv	A degree of jitter in x and y direction in the graph. The default is zero.
newwd	A logical variable to indicate whether to print graph in a new window. The default value is TRUE.

**Value**

Predicted Means	A table of predicted means.
-----------------	-----------------------------

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**Examples**

```
library(predictmeans)
data(Oats, package="nlme")
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
# library(lme4)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
covariatemmeans(fm, "Variety", covariate="nitro")
covariatemmeans(fm, "Variety", covariate="nitro", as.is=TRUE)$data
covariatemmeans(fm, "Variety", covariate="nitro", covariateV=seq(0, 0.6, 0.1))$data
```

---

Kmatrix

---

*Matrix of Coefficients in a Linear Model*


---

**Description**

This function obtains a matrix of coefficients for parametric models such as aov, lm, glm, gls, lme, and lmer.

**Usage**

```
Kmatrix(model, modelterm, covariate=NULL, as.is=FALSE, covariateV=NULL, prtnum=FALSE)
```

**Arguments**

model	Model object returned by aov, lm, glm, gls, lme, and lmer.
modelterm	Name (in "quotes") for indicating which model term's predicted mean to be calculated. The modelterm must be given exactly as it appears in the printed model, e.g. "A" or "A:B".
covariate	A numerical vector to specify values of covariates for calculating predicted means, default values are the means of the associated covariates. It also can be the name of one covariate in the model.
as.is	A logic value to specify wheather or not using original covariate values, the default is FALSE.

covariateV	A numeric vector when as.is is FALSE and covariate is name of one covariate in the model, then covariate means will produce the result for covariate at value of covariateV.
prtnum	An option for printing covariate info on the screen or not. The default is FALSE.

**Value**

K	Coefficients matrix
fctnames	A model frame contains factor(s) info in the model.
response	The name of response variable in the model.

**Author(s)**

This function heavily depends on the codes from package "lsmeans".

**References**

Welham, S., Cullis, B., Gogel, B., Gilmour, A., & Thompson, R. (2004), *Prediction in linear mixed models*, Australian and New Zealand Journal of Statistics, 46(3), 325-347.

**Examples**

```
library(predictmeans)
data(Oats, package="nlme")
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
Kmatrix(fm, "Variety", prtnum=TRUE)$K
Kmatrix(fm, "Variety", 0.5, prtnum=TRUE)$K
Kmatrix(fm, "Variety", "nitro")$K
Kmatrix(fm, "Variety", "nitro", as.is=TRUE)$K
Kmatrix(fm, "Variety", "nitro", covariateV=seq(0, 0.6, 0.1))$K
```

---

permanova.lmer

*Permutation ANOVA for lmer Model*


---

**Description**

This function provides permutation ANOVA for lmer model.

**Usage**

```
permanova.lmer(model, nperm = 999, ncore=3, drop=TRUE, ...)
```



**Arguments**

model	Model object returned by lmer.
nperm	Number of permutation, the default value is 999.
ncore	Number of core for parallel computing, the default value is 3.
drop	Drop or add model term sequentially, the default value is TRUE.
...	Use to setup option: seed – Specify a random number generator seed, for reproducible results.

**Value**

Permutation ANOVA table.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**Examples**

```
library(predictmeans)
Oats$nitro <- factor(Oats$nitro)
fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)

## Permutation Test for model terms
# permanova.lmer(fm)
# permanova.lmer(fm, drop=FALSE)
## Compare to F test
# fm0 <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
# anova(fm0)
```

---

permindex

*Permutation Index*

---

**Description**

This function obtains permutation index for a dataset.

**Usage**

```
permindex(data, block=NULL, group=NULL, nsim=4999, seed)
```

**Arguments**

data	Data object used in the model fitting.
block	Name (in "quotes") for the blocking factor in the data.
group	Name (in "quotes") for the group factor in the data.
nsim	The number of permutations. The default is 4999.
seed	Specify a random number generator seed, for reproducible results.

**Value**

A matrix has 'nsim' columns of permuted index.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**Examples**

```
library(predictmeans)
block <- rep(1:3, each=12)
group <- rep(rep(1:3, each=4), 3)
data <- data.frame(block, group)
cbind(data, permindex(data, block="block", group="group", nsim=5))
# Permute group as a whole within each block first,
# then permute obs within each group.
cbind(data, permindex(data, block="block", nsim=5))
# Permute obs within each block only.
cbind(data, permindex(data, group="group", nsim=5))
# Permute groups as a whole block first,
# then permute obs within each group.
cbind(data, permindex(data, nsim=5)) # Free permutation.
```

---

permlmer

---

*Permutation Test of random or fixed effects for lmer model.*


---

**Description**

This function provides permutation tests for the terms in a linear mixed model of lmer.

**Usage**

```
permlmer(lmer0, lmer1, nperm = 999, ncore=3, plot=FALSE, seed)
```

**Arguments**

lmer0	lmer model under H0, note that lmer0 model must nest within lmer1 model.
lmer1	lmer model under H1, note that lmer0 model must nest within lmer1 model.
nperm	Number of permutation, the default value is 999.
ncore	Number of core for parallel computing, the default value is 3.
plot	Plot permutation distribution or not, the default value is FALSE.
seed	Specify a random number generator seed, for reproducible results.

**Value**

Permutation p-value.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**References**

Oliver E. Lee and Thomas M. Braun (2012), *Permutation Tests for Random Effects in Linear Mixed Models*. *Biometrics*, Journal 68(2).

**Examples**

```
# library(predictmeans)
## Test random effects
# fm1 <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
# fm2 <- lmer(Reaction ~ Days + (Days || Subject), sleepstudy)
# fm3 <- update(fm1, . ~ . - (Days | Subject) + (1 | Subject))
# anova(fm1, fm2, fm3)
# permlmer(fm3, fm2)
# permlmer(fm2, fm1)

## Test fixed effects
# Oats$nitro <- factor(Oats$nitro)
# fm0 <- lmer(yield ~ nitro+Variety+(1|Block/Variety), data=Oats)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
# permlmer(fm0, fm)
```

---

permmodels

*Permutation Test of Linear Model*


---

**Description**

This function provides permutation t-tests for coefficients of (fixed) effects and permutation F-tests for the terms in a linear model such as aov, lm, glm, gls, lme, and lmer.

**Usage**

```
permmodels(model, data, block=NULL, group=NULL, covariate=NULL, nsim=4999, check=FALSE,
  exact=FALSE, fo=NULL, prt=TRUE, seed)
```

**Arguments**

model	Model object returned by aov, lm, glm, gls, lme, and lmer.
data	Data object used in the model fitting.
block	Name (in "quotes") for the blocking factor in the data.
group	Name (in "quotes") for the group factor in the data.
covariate	Name(s) (in "quotes") for the covariate variable(s) in the model.
nsim	The number of permutations. The default is 4999.

check	a logical variable to indicate whether or not you want to check permutation results, the default is FALSE.
exact	A logical variable to indicate whether or not exact no. of permutations will be used (applicable only to free the permutation case). The default is FALSE.
fo	A model formula used in the model; fo!=NULL when the formula is specified by function formula.
prt	A logical variable to indicate whether or not to print output on the screen. The default is TRUE.
seed	Specify a random number generator seed, for reproducible results.

### Value

The function produces permutation t-test table for coefficients of (fixed) effects, permutation ANOVA table for model terms and a model parameter list `permlist`, a list containing `nsim=4999` times permutation refitted model parameters which are used in functions `predictmeans` and `contrastmeans`.

### Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

### Examples

```
## Not run for simplifying process of submitting pkg to CRAN
#library(predictmeans)
#Oats$nitro <- factor(Oats$nitro)
#fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
## library(lme4)
## fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
#
## Permutation Test for model terms
#system.time(
#  permlme <- permmodels(model=fm, data=Oats, block="Block", group="Variety", nsim=999)
#)
#
## Permutation Test for multiple comparisons
#predictmeans(model=fm, modelterm="nitro:Variety", atvar="Variety", adj="BH",
#  permlist=permlme, plot=FALSE)
#
## Permutation Test for specified contrasts
#cm <- rbind(c(-1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
#  c(0, 0, 1, 0, 0, 0, 0, -1, 0, 0, 0, 0))
#contrastmeans(model=fm, modelterm="nitro:Variety", ctrmatrix=cm, permlist=permlme)
```

---

PMplot *Level Plot of a Matrix of p-values.*

---

### Description

Creates a plot of p-values of pairwise comparisons.

### Usage

```
PMplot(pmatrix, level=0.05, mtitle=NULL, xlabel=NULL, margin=5, legendx=0.73,  
newwd=TRUE)
```

### Arguments

pmatrix	A matrix with p-values from pairwise comparisons. (This is a lower triangle matrix.)
level	The level of p-value to be highlighted. Default is 0.05.
mtitle	The main title in the graph.
xlabel	The x and y labels in the graph.
margin	A value for specifying x and y margins in the graph. The default value is 5.
legendx	A value for specifying x coordinate of legend. The default value is 0.73.
newwd	A logical variable to indicate whether to print graph in a new window. The default is TRUE.

### Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

### Examples

```
library(predictmeans)  
set.seed(2013)  
pvalues <- runif(28)  
pmatrix <- matrix(0,8,8)  
pmatrix[lower.tri(pmatrix)] <- pvalues  
round(pmatrix, 4)  
PMplot(pmatrix)  
  
Oats$nitro <- factor(Oats$nitro)  
fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)  
predictout <- predictmeans(fm, "nitro:Variety", atvar="Variety", adj="BH", barplot=TRUE)  
PMplot(predictout$p_valueMatrix)
```

---

 predictmeans

*Predicted Means of a Linear Model*


---

### Description

This function obtains predicted means, SE of means, SED of means, LSDs and plots of means with Stder bar or LSD bar for parametric models such as aov, lm, glm, gls, lme, and lmer. The function also performrs pairwise comparisons and permutation tests.

### Usage

```
predictmeans(model, modelterm, pairwise=FALSE, atvar=NULL, adj="none", Df=NULL,
  level=0.05, covariate=NULL, letterdecr=TRUE, trans = NULL, transOff = 0,
  responsen=NULL, count=FALSE, plotord=NULL, plottitle=NULL, plotxlab=NULL,
  plotylab=NULL, mplot=TRUE, barplot=FALSE, pplot=TRUE, bkplot=TRUE, plot=TRUE,
  jitterv=0, basesz=12, prtnum=TRUE, newwd=TRUE, permlist=NULL, ndecimal=4)
```

### Arguments

model	Model object returned by aov, lm, glm, gls, lme, and lmer.
modelterm	Name (in "quotes") for indicating which factor term's predicted mean to be calculated. The modelterm must be given exactly as it appears in the printed model, e.g. "A" or "A:B".
pairwise	An option for showing pair-wise LSDs and p-values, or not. The default is FALSE.
atvar	When pairwise = TRUE, a quoted name indicating within levels of which variable in modelterm the multiple comparison will be performed.
adj	Name (in "quote") for indicating a method for adjusting p-values of pairwise comparisons. The choices are "none", "tukey", "holm", "hochberg", "hommel", "bonferroni", "BH", "BY" and "fdr". The default method is "none". Note that LSD can't be adjusted except for "bonferroni" method.
Df	A degree of freedom for calculating LSD. For the above models, Df is obtained from the function automatically.
level	A significant level for calculating LSD. The default value is 0.05.
covariate	A numerical vector to specify values of covariates for calculating predicted means. The default values are the means of the associated covariates.
letterdecr	A logical variable to indicate whether to print letters for multiple comparisons by decreasing order. The default is TRUE.
trans	A function object for calculating the back transformed means, e.g. trans=exp.
transOff	When you use trans=exp(x+1), then transOff=1, the default is 0.
responsen	Name (in "quotes") of the back transformed response variable in the model.
count	An option for indicating the back transformed mean values are counts or not. The default is FALSE.

plotord	A numeric vector specifying the order of plotting for two or three way interaction (e.g. plotord = c(2,1,3) will put the second variable in modelterm on the X axis, the first variable as the grouping variable, and the third one as the panel variable). The defaults are c(1,2) and c(1,2,3) for two and three way interactions.
plottitle	A character vector specifying the main title for plot(s). The default is NULL.
plotxlab	A character vector specifying the x label for plot(s). The default is NULL.
plotylab	A character vector specifying the y label for plot(s). The default is NULL.
mplot	An option for drawing a means plot, or not. The default is TRUE.
barplot	An option for drawing a bar chart, or not. The default is FALSE.
pplot	An option for drawing a p-values plot, or not when there are more than six p-values. The default is TRUE.
bkplot	An option for drawing back transformed plot, or not. The default is TRUE.
plot	An option for drawing plots, or not. The default is TRUE.
jitterv	A degree of jitter in x and y direction in the back transformed means graph. The default is zero.
basesz	The base font size. The default is 12.
prtnum	An option for printing covariate information on the screen, or not. The default is TRUE.
newwd	A logical variable to indicate whether to print graph in a new window. The default is TRUE.
permlist	A model parameter list produced by the function permmodels. When permlist != NULL, the option Df will be non-functional. This is a key option for pairwise comparisons via permutation tests.
ndecimal	An option for specifying number of decimal point to be print at predicted means table. The default is 4.

## Value

Predicted Means	A table of predicted means.
Standard Error of Means	A table of standard errors of predicted means.
Standard Error of Differences	Standard errors of differences between predicted means.
LSD	Least significant differences between predicted means.
Back Transformed Means	When trans!=NULL, a table of back transformed means with CIs are shown.
Pairwise p-value	A matrix with t-values above the diagonal and p-values below the diagonal, or matrix of pairwise comparison p-values for each level of atvar.
predictmeansPlot	ggplot of predicted means.

predictmeansBKPlot  
                   ggplot of back transformed means.  
 predictmeansBarPlot  
                   gg bar plot of predicted means.  
 p\_valueMatrix  p\_value matrix for pairwise comparison.

### Note

The predictmeans function becomes confused if a factor or covariate is changed to the other in a model formula. Consequently, formulae that include calls as `factor`, `factor`, or `numeric` (e.g. `as.factor(income)`) will cause errors. Instead, create the modified variables outside of the model formula (e.g., `fincome <- as.factor(income)`) and then use them in the model formula.

Factors cannot have colons in level names (e.g., "level:A"); the predictmeans function will confuse the colons with interactions; rename levels to avoid colons.

For predictmeans function, it is assumed that methods `coef`, `vcov`, `model.matrix`, `model.frame` and `terms` are available for `model`.

### Author(s)

Dongwen Luo, Siva Ganesh and John Koolgaard

### References

Torsten Hothorn, Frank Bretz and Peter Westfall (2008), *Simultaneous Inference in General Parametric Models*. *Biometrical*, Journal 50(3), 346–363.

Welham, S., Cullis, B., Gogel, B., Gilmore, A., & Thompson, R. (2004), *Prediction in linear mixed models*, Australian and New Zealand Journal of Statistics, 46(3), 325-347.

### Examples

```
library(predictmeans)
fdata(xtabs(yield ~ Block+Variety+nitro, data=Oats))
Oats$nitro <- factor(Oats$nitro)
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
# library(lme4)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
predictmeans(fm, "nitro", adj="BH")
predictmeans(fm, "nitro:Variety", atvar="Variety", adj="BH")
predictout <- predictmeans(fm, "nitro:Variety", atvar="Variety", adj="BH", barplot=TRUE)
names(predictout)
print(predictout$predictmeansPlot)
print(predictout$predictmeansBarPlot)
```



residplot

*Diagnostic Plots for a Linear (Mixed) Model***Description**

This function produces diagnostic plots for linear models including 'aov', 'lm', 'glm', 'gls', 'lme' and 'lmer'.

**Usage**

```
residplot(model, group = "none", level = 1, slope = FALSE, id = FALSE, newwd=TRUE,
          ask=FALSE)
```

**Arguments**

model	Model object returned by aov, lm, glm, gls, lme, and lmer.
group	Name (in "quotes") for indicating the variable used to show grouping in the residual vs predicted plot. If variable is a term in the model, then group will be a name of the variable such as group="A", otherwise group will be the actual variable such as group=data\$A.
level	An integer 1, 2, etc. used to specify a level of the random effect for plotting. The default value is 1.
slope	A logical variable. If set to TRUE, a Q-Q plot of random slope will be drawn.
id	A logical variable. If set to TRUE, outliers in the residual vs fitted plot can be identified interactively.
newwd	A logical variable to indicate whether to print graph in a new window. The default is TRUE.
ask	logical. If TRUE (and the R session is interactive) the user is asked for input, before a new figure is drawn.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolgaard

**Examples**

```
## Note that the order of levels of nested random effects is oposite
## between lme and lmer objects.

library(predictmeans)
Oats$nitro <- factor(Oats$nitro)
fm <- lme(yield ~ nitro*Variety, random=~1|Block/Variety, data=Oats)
residplot(fm, level=2) #lme: level=2 for random effect "Block:Variety"

# Not Run
# library(lme4)
# fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
# residplot(fm) # lmer: By default level=1 for random effect "Block:Variety"
```

---

`varcomp.lmer`*Calculate stder and CI of variance components for lmer Model*

---

**Description**

This function calculates stder and CI of variance components for lmer Model.

**Usage**

```
varcomp.lmer(model, level=0.95)
```

**Arguments**

<code>model</code>	Model object returned by lmer.
<code>level</code>	level of confidence of CI, the default value is 0.95.

**Value**

Variance components table.

**Author(s)**

Dongwen Luo, Siva Ganesh and John Koolaard

**Examples**

```
library(predictmeans)
Oats$nitro <- factor(Oats$nitro)
fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
varcomp.lmer(fm)
# Compare with simulation
confint(fm)^2
```

# Index

## \*Topic **package**

predictmeans-package, [2](#)

anova.lmer, [2](#)

contrastmeans, [3](#)

CookD, [5](#)

covariatemeans, [6](#)

Kmatrix, [7](#)

permanova.lmer, [8](#)

perindex, [9](#)

permlmer, [10](#)

permodels, [11](#)

PMplot, [13](#)

predictmeans, [14](#)

predictmeans-package, [2](#)

residplot, [17](#)

varcomp.lmer, [18](#)