Package ‘predictmeans’
August 23, 2018

Type Package
Title Calculate Predicted Means for Linear Models
Version 1.0.1
Date 2018-08-06
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 (>= 3.0.0), 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 2018-08-23 04:40:07 UTC
NeedsCompilation no

R topics documented:

predictmeans-package .................................................. 2
anovalmer ................................................................. 2
contrastmeans ............................................................ 3
CookD ................................................................. 5
covariatemmeans .......................................................... 6
Kmatrix ................................................................. 7
permanova.lmer .......................................................... 8
permindex ............................................................... 9
permlmer ................................................................. 10
PMplot ................................................................. 11
predictmeans ........................................................... 12
residplot ............................................................... 15
Index

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 Koolaard

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

References


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

anovalmer(model, DDF=NULL)
contrastmeans

Arguments

model  Model object returned by lmer.
Df    Model terms’ denominate degree of freedom specified by user.

Value

ANOVA table.

Author(s)

Dongwen Luo, Siva Ganesh and John Koolaard

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)
anovalmer(fm)

calculates contrasts for linear models obtained from functions aov, lm, glm, gls, lme, and 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 program 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".
contrastmeans

Df A denominator degree of freedom for model term. (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 Koolaard

References


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
CookD

`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", ctmatrix=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**

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

**Arguments**

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

**Examples**

```r
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)
```
covariatemans

**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**

```r
covariatemans(model, modelterm=NULL, covariate, as.is=FALSE, level=0.05, Df=NULL, 
trans=NULL, transoff=0, responsen=NULL, 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(s) (in "quotes") of the covariate variable(s) in the model.
- `as.is`: A logic value to specify whether or not using original covariate values, the default is FALSE.
- `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.
Kmatrix

Value
Predicted Means
A table of predicted means.

Author(s)
Dongwen Luo, Siva Ganesh and John Koolaard

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)
covariatemeans(fm, "Variety", covariate="nitro")

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, 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 covariate in the model.
as.is A logic value to specify whether or not using original covariate values, the default is FALSE.
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


Examples

```r
library(predictmeans)
# library(lme4)
# 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)
kmatrix(fm, "Variety", 0.5, prtnum=TRUE)
# kmatrix(fm, "Variety", "nitro")
```

---

### permanova.lmer

*Permutation ANOVA for lmer Model*

---

Description

This function provides permutation ANOVA for lmer model.

Usage

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

Arguments

- **model**
  - Model object returned by lmer.
- **perms**
  - 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.

Value

Permutation ANOVA table.

Author(s)

Dongwen Luo, Siva Ganesh and John Koolaard
permindex

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 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))

Description

This function obtains permutation index for a dataset.

Usage

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

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.

Value

A matrix has `nsim` columns of permuted index.

Author(s)

Dongwen Luo, Siva Ganesh and John Koolaard

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))
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, perms = 999, ncore=3, plot=TRUE)

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.
perms  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 TRUE.

Value

Permutation p-value.

Author(s)

Dongwen Luo, Siva Ganesh and John Koolaard

References


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)
# perm_lmer(fm0, fm)

## Description

Creates a plot of p-values of pairwise comparisons.

## Usage

```r
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 Koolaard

## Examples

```r
library(predictmeans)
set.seed(2013)
pvalues <- runif(28)
pmat <- matrix(0,8,8)
pumat[lower.tri(pmat)] <- pvalues
round(pmat, 4)
PMplot(pmat)
Oats$nitro <- factor(Oats$nitro)
fm <- lmer(yield ~ nitro*Variety+(1|Block/Variety), data=Oats)
```
predictmeans <- 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
Studer bar or LSD bar for parametric models such as aov, lm, glm, gls, lme, and lmer. The function
also performs 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)

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 cal-
culated. 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.
predictmeans

**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` is `NULL`, the option `df` will be non-functional. This is a key option for pairwise comparisons via permutation tests.

**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 Koolaard

References


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)
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

```r
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 Koolaard

Examples

```r
# Note that the order of levels of nested random effects is opposite 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"
```
Index

* Topic \textasciitilde kwd1
  PMplot, 11
* Topic \textasciitilde kwd2
  PMplot, 11
* Topic package
  predictmeans-package, 2

anova, 2
contrastmeans, 3
CookD, 5
covariatemmeans, 6
Kmatrix, 7
permanova.lmer, 8
permindex, 9
permmlmer, 10
PMplot, 11
predictmeans, 12
predictmeans-package, 2
residplot, 15