

Package ‘asuR’

April 17, 2009

Type Package

Title Advanced statistics using R

Version 0.08-24

Date 2007-06-01

Author Thomas Fabbro <thomas.fabbro@unibas.ch>

Maintainer Thomas Fabbro <thomas.fabbro@unibas.ch>

Description Functions and data sets for a lecture in Advanced Statistics using R. Especially the functions `mancontr()` and `inspect()` may be of general interest. With `mancontr()`, short for manual contrasts, it is possible to specify your own contrasts and give them useful names. Something that is important in most projects of reasonable size. The function `inspect()` shows a wide range of inspection plots to validate model assumptions (currently for models fitted with `lm`, `glm`, and `lmer`). And do not forget to have a look at `norm.test()`, it is not only fun!

Depends MASS, lattice, grid, methods, lme4

Suggests car

License GPL (>= 2)

URL http://www.evolution.unibas.ch/teaching/r_course/asuR.htm

Repository CRAN

Date/Publication 2007-06-02 16:40:06

R topics documented:

| | |
|------------------------|---|
| asuR-package | 3 |
| BtheB | 3 |
| budworm | 4 |
| cathedral | 6 |
| colVars | 6 |
| dep | 7 |

| | |
|--------------------------------|----|
| dep-methods | 8 |
| dep.glm | 9 |
| flowers | 10 |
| gala | 11 |
| growth | 11 |
| hnp | 12 |
| hnp-methods | 13 |
| hnp.glm | 13 |
| houseflies | 14 |
| inspect | 15 |
| inspect-methods | 16 |
| inspect.glm | 16 |
| inspect.lmer | 18 |
| lep | 19 |
| lep-methods | 20 |
| lep.glm | 20 |
| mancontr | 21 |
| mytrees | 22 |
| nep | 23 |
| nep-methods | 24 |
| nep.lmer | 24 |
| norm.test | 25 |
| nrp | 25 |
| nrp-methods | 26 |
| nrp.lmer | 26 |
| oring | 27 |
| parseFormula | 28 |
| parseFormula | 29 |
| parseFormula-methods | 30 |
| parseFormula.glm | 31 |
| parseFormula.lmer | 31 |
| pea | 32 |
| plants | 33 |
| rfp | 33 |
| rfp-methods | 34 |
| rfp.lmer | 35 |
| rgp | 36 |
| rgp-methods | 36 |
| rgp.lmer | 37 |
| rpp | 38 |
| rpp-methods | 39 |
| rpp.glm | 39 |
| rpp.lm | 40 |
| schoolclass | 41 |
| unemployment | 41 |
| weight | 43 |
| wellplate | 44 |
| xyp | 45 |

| | |
|--------------|------------------------------------|
| asuR-package | <i>Advanced statistics using R</i> |
|--------------|------------------------------------|

Description

Functions and data sets used for a statistic course using R.

Details

Package: asuR
 Type: Package
 License: GPL version 2 or newer

Author(s)

thomas.fabbro@unibas.ch

| | |
|-------|----------------------------|
| BtheB | <i>Beat the Blues Data</i> |
|-------|----------------------------|

Description

Data from a clinical trial of an interactive multimedia program called ‘Beat the Blues’.

Usage

```
data(BtheB)
```

Format

A data frame with 100 observations of 100 patients on the following 8 variables.

drug did the patient take anti-depressant drugs (No or Yes).

length the length of the current episode of depression, a factor with levels <6m (less than six months) and >6m (more than six months).

treatment treatment group, a factor with levels TAU (treatment as usual) and BtheB (Beat the Blues)

bdi.pre Beck Depression Inventory II before treatment.

bdi.2m Beck Depression Inventory II after two months.

bdi.4m Beck Depression Inventory II after four months.

bdi.6m Beck Depression Inventory II after six months.

bdi.8m Beck Depression Inventory II after eight months.

Details

Longitudinal data from a clinical trial of an interactive, multimedia program known as "Beat the Blues" designed to deliver cognitive behavioural therapy to depressed patients via a computer terminal. Patients with depression recruited in primary care were randomised to either the Beating the Blues program, or to "Treatment as Usual (TAU)".

Note that the data are stored in the wide form, i.e., repeated measurements are represented by additional columns in the data frame.

Note

see BtheBlong for a long version of this dataset

Source

from package HSAUR: J. Proudfoot, D. Goldberg and A. Mann (2003). Computerised, interactive, multimedia CBT reduced anxiety and depression in general practice: A RCT. *Psychological Medicine*, **33**, 217–227.

References

```
library(HSAUR)
```

Examples

```
data(BtheB)
layout(matrix(1:2, nrow = 1))
ylim <- range(BtheB[,grep("bdi", names(BtheB))], na.rm = TRUE)
boxplot(subset(BtheB, treatment == "TAU")[,grep("bdi", names(BtheB))],
        main = "Treated as usual", ylab = "BDI",
        xlab = "Time (in months)", names = c(0, 2, 4, 6, 8), ylim = ylim)
boxplot(subset(BtheB, treatment == "BtheB")[,grep("bdi", names(BtheB))],
        main = "Beat the Blues", ylab = "BDI", xlab = "Time (in months)",
        names = c(0, 2, 4, 6, 8), ylim = ylim)
```

budworm

budworm data

Description

Data from a small experiment on the toxicity to the tobacco budworm

Usage

```
data(budworm)
```

Format

A data frame with 12 observations on the following 4 variables.

num.dead a numeric vector

num.alive a numeric vector

sex a factor with levels female male

dose a numeric vector

Details

Experiment on the toxicity to the tobacco budworm *Heliothis virescens* of doses of the pyrethroid trans-cypermethrin to which the moths were beginning to show resistance. Batches of 20 moths of each sex were exposed for three days to the pyrethroid and the number in each batch that were dead or knocked down was recorded.

Source

Collette(1991)

References

from MASS p.190; by Venables and Ripley (see also there for suggestions on how to analyse)

Examples

```
## usage:
data(budworm)

budworm.contr <- rbind("female-male"=c(1,-1))
b.glm <- glm(cbind(num.dead,num.alive) ~ sex*log2(dose),
             family=binomial,
             contrasts=list(sex=mycontr(contr=budworm.contr)),
             data=budworm)

b1.glm <- glm(cbind(num.dead,num.alive) ~ sex*I(log2(dose)-3),
             family=binomial,
             contrasts=list(sex=mycontr(contr=budworm.contr)),
             data=budworm)

b2.glm <- glm(cbind(num.dead,num.alive) ~ sex + I(log2(dose)-3),
             family=binomial,
             contrasts=list(sex=mycontr(contr=budworm.contr)),
             data=budworm)
```

`cathedral`*Medieval cathedrals in England*

Description

The length and height of medieval cathedrals in England and whether they are belonging to the Gothic or Romanesque style. Some cathedrals have parts in both styles and are listed twice.

Usage

```
data(cathedral)
```

Format

A data frame with 25 observations on the following 3 variables.

style a factor with levels `g`, Gothic, `r`, Romanesque

x a numeric vector, nave height

y a numeric vector, total length in feet

Source

From Julian Faraway

Examples

```
data(cathedral)
```

`colVars`*Column variances*

Description

Form column variances for numeric arrays

Usage

```
colVars(x, na.rm = FALSE, dims = 1, unbiased = TRUE, SumSquares = FALSE, twopass =
```

Arguments

| | |
|-------------------------|--|
| <code>x</code> | A numeric array (or a dataframe to convert to a matrix) |
| <code>na.rm</code> | Logical: Remove NA's (not available values) |
| <code>dims</code> | Number of dimensions to sum over [<code>colSums</code>] or leave alone [<code>rowSums</code>]. Only useful when <code>x</code> is a multidimensional array |
| <code>unbiased</code> | Logical: Use (N-1) in the denominator when calculating variance |
| <code>SumSquares</code> | Logical: If TRUE, <code>colVars</code> just returns sums of squares. |
| <code>twopass</code> | Logical: If TRUE, <code>colVars</code> uses the corrected two-pass algorithm of Chan Golub & LeVeque, which is slower but less subject to roundoff error. |

Value

A vector with the variance for every column.

Author(s)

`colVars`: Originally by Douglas Bates <bates@stat.wisc.edu> as package "MatUtils". Modified, expanded, and renamed by David Brahm <brahm@alum.mit.edu>, with help of course from the R-help gurus.

See Also

[colMeans](#)

Examples

```
mat <- cbind(rnorm(100, sd=sqrt(1)),
            rnorm(100, sd=sqrt(2)),
            rnorm(100, sd=sqrt(3)),
            rnorm(100, sd=sqrt(4)),
            rnorm(100, sd=sqrt(5)))
colVars(mat)
```

dep

deviance residuals vs. linear predictor plot

Description

There is a separate help page for every generic.

Usage

```
dep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>glm</code> , usually, a result of a call to the function <code>glm</code> . |
| <code>id</code> | a character string; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[dep.glm](#), [inspect](#)

Examples

```
##
```

dep-methods

Methods for Function dep in Package 'asuR'

Description

Creates a plot of deviance residuals vs. linear predictor (eta).

Methods

mymodel = "ANY" Generic function

mymodel = "glm" Generates a plot of an object of class "glm"

`dep.glm`*deviance residuals vs. linear predictor plot*

Description

Plots the deviance residuals versus the linear predictor (eta).

Usage

```
## S3 method for class 'glm':  
dep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>glm</code> , usually, a result of a call to the function <code>glm</code> . |
| <code>id</code> | a character string; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

***expected pattern:

random scatter in a horizontal band

***Questions:

A) is the relationship linear? otherwise:

1. change the choice of predictors
2. change the transformations of the predictor
3. change the link function (but there are only few choices...)
4. do not transform the response in `glm` since this would change the distribution of the response (you would do this in a `lm`)

B) is the variance constant? otherwise:

1. change the variance function
2. use weights if you identify some features of the data that suggest a suitable choice

Value

A plot and for the user identified points an invisible vector of row indices (corresponding to the original data).

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[inspect](#)

Examples

```
##
```

```
flowers
```

```
Flower
```

Description

A data set with the dry mass of all flowers and the dry mass of the total plant from 20 species growing at high and 20 species growing at low altitude.

Usage

```
data(flowers)
```

Format

A data frame with 40 observations on the following 3 variables.

alt a factor with levels high low

flower a numeric vector; dry mass of flowers in mg

total a numeric vector; dry mass of total plant in mg

Details

The data sets shows a random subsample of the original data set.

Source

Fabbro, T. & Koerner, Ch. (2004): *Altitudinal differences in flower traits and reproductive allocation*. FLORA 199, 70-81. Fabbro, Koerner ()

Examples

```
data(flowers)
## a model with two intercepts and two slopes
# m1 <- lm(log(flower) ~ alt/log(total) -1, data=flower)
#
## a model with two intercepts and one slope
# altdiff <- rbind("high-low"=c(1,-1))
# m2 <- lm(log(flower) ~ alt + log(total), data=flower,
#          contrasts=list(alt=mycontr(contr=altdiff)))
#
## are separate slopes needed?
# anova(m1, m2) # conclusion:
#
## no difference in slopes but difference in intercept
## for interpretation also test whether the slope is one!
```

`gala`*Species diversity on the Galapagos Islands*

Description

There are 30 Galapagos islands and 7 variables in the dataset. The relationship between the number of plant species and several geographic variables is of interest. The original dataset contained several missing values which have been filled for convenience.

Usage

```
data(gala)
```

Format

The dataset contains the following variables

Species the number of plant species found on the island

Endemics the number of endemic species

Area the area of the island (km²)

Elevation the highest elevation of the island (m)

Nearest the distance from the nearest island (km)

Scruz the distance from Santa Cruz island (km)

Adjacent the area of the adjacent island (square km)

Note

original from the package faraway, but there tortoise instead of plant species

Source

M. P. Johnson and P. H. Raven (1973) "Species number and endemism: The Galapagos Archipelago revisited" *Science*, 179, 893-895

`growth`*Weight Gain of two Species at different Nitrogen Concentrations*

Description

For 30 individuals of species A and B the weight gain during one week and the soil nitrogen concentration was measured. The aim of the study was to test whether this two species respond differently.

Usage

```
data(growth)
```

Format

A data frame with 60 observations on the following 3 variables.

gain a numeric vector; weight gain in mg

species a factor with levels sp_A sp_B

nitrogen a numeric vector; the soil nitrogen concentration

Source

thomas.fabbro@unibas.ch

Examples

```
data(growth)
```

hnp

half-normal quantiles vs. absolute stud. residuals plot

Description

There is a separate help page for every generic.

Usage

```
hnp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>glm</code> , usually, a result of a call to the function <code>glm</code> . |
| <code>id</code> | a character string or numeric value; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

<thomas.fabbro@unibas.ch>

See Also

[hnp.glm](#), [inspect](#)

Examples

```
##
```

| | |
|-------------|---|
| hnp-methods | <i>Methods for Function hnp in Package 'asuR'</i> |
|-------------|---|

Description

Methods for function hnp in Package 'asuR'

Methods

mymodel = "ANY" generic function

mymodel = "glm" Diagnostics plots for objects of class "glm"

| | |
|---------|--|
| hnp.glm | <i>half normal quantiles vs. absolute studentized residuals plot</i> |
|---------|--|

Description

In linear regression we inspect the normal quantile plot to check the normality of residuals. For generalized linear models we do not expect that the residuals are normally distributed. Nevertheless the half-normal plot can help us to find outliers. Outliers can be detected as points off the trend.

Usage

```
## S3 method for class 'glm':
hnp(mymodel, id= c("all", "none"), ...)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class glm, usually, a result of a call to the function glm. |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

***expected pattern:

all points are scattered along a line

***question: are there points of the trend? otherwise

1. Is the data point correct?

(keep attention that this is not a quantile–quantile plot where you check for normality!)

Value

A plot and a vector with identified values (corresponding to the row number in the original data).

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

`inspect`

Examples

```
#
```

houseflies

Housfly Development

Description

Developmental duration of three housefly strains for two different treatments.

Usage

```
data(houseflies)
```

Format

A data frame with 48 observations on the following 3 variables.

duration a numeric vector

treatment a factor with levels `contr` `treat`

strain a factor with levels `BE` `BW` `OL`

Source

thomas.fabbro@unibas.ch

Examples

```
data(houseflies)
```

inspect

*Plots to inspect fitted models***Description**

This functions can be used to validate some assumptions of linear regressions. The idea of this functions is to illustrate possibilities, not to provide fully functional code. The functions work for regression models with continuous variables only.

Usage

```
inspect(mymodel, which = c("select", "sequence", "all"), id = c("all", "none"), ...
irp(mymodel)
ryp(mymodel, id= c("none", "all"))
ilp(mymodel)
prp(mymodel)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class <code>glm</code> or <code>lm</code> , usually, a result of a call to the function <code>glm</code> or <code>lm</code> |
| which | a character string; one of the following: select select a diagnostic plot from a menu with all plots available (also repeatedly selecting the same plot is possible) sequence plot all available diagnostic plots in a sequence (with the possibility to skip a plot or quit) all plot all available diagnostic plots without user interaction (mainly useful for printing, see examples) id a character string or numeric value; in which panel should it be possible to interactively identify values ... further arguments |

Details

The function `inspect` calls other functions (depending on their class) to inspect the model assumptions.

Value

A list with identified values (row names), one slot for each inspection function.

Note

On some devices you are not allowed to resize the window before interactively selecting points

Author(s)

<thomas.fabbro@unibas.ch>

See Also

plot

Examples

```
data(mytrees)
model <- lm(log(Volume) ~ log(Girth) + log(Height), data=mytrees)
## inspect(model)

### for printing
##pdf(~temp/diagnostic.pdf)
##inspect(model)
##dev.off()
```

inspect-methods *Methods for Model Inspection in Package 'asuR'*

Description

Inspection of model objects using diagnostic plots

Methods

mymodel = "ANY" Generic function
mymodel = "lm" Diagnostic plots for objects of class "lm"
mymodel = "glm" Diagnostics plots for objects of class "glm"
mymodel = "lmer" Diagnostics plots for objects of class "lmer"

inspect.glm *Plots to Inspect fitted Models*

Description

This functions can be used to validate some assumptions of generalized linear models.

Usage

```
## S3 method for class 'glm':
inspect(mymodel, which = c("select", "sequence", "all"), id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|---|
| <code>mymodel</code> | an object of class <code>glm</code> , usually the result of a call to the function <code>glm</code> |
| <code>which</code> | a character string; do you want to <code>select</code> the plots from a list of all possible diagnostics plots available or do you want <code>all</code> of them, one after the other |
| <code>id</code> | a character string or numeric value; in which panel should it be possible to interactively identify values |
| <code>...</code> | further arguments |

Details

For `glm` available:

| description | short & | long function names |
|--|------------------|---|
| deviance residuals vs. linear predictor | <code>dep</code> | <code>devianceResidual.linearPredictor</code> |
| partial residual vs. each predictor | <code>rpp</code> | <code>partialResidual.eachPredictor</code> |
| linearized response vs. linear predictor | <code>lep</code> | <code>linearizedResponse.linearPredictor</code> |
| half-normal quantiles vs. absolute stud. residuals | <code>hnp</code> | <code>halfNormalQuantiles.absoluteStudentizedResiduals</code> |

Value

A list with identified values (row indices), one named slot (short name) for each inspection function and an additional slot (called `all`) with all values that were selected at least in one plot.

Note

This functions help to inspect `glm`'s with continuous predictors, sometimes they are anyway informative also for categorical predictors ...

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

`plot`

Examples

```
## data(gala)
## model <- glm(Species ~ log(Area) + log(Elevation) + log(Nearest) + log(Scruz+0.1) + log(
## inspect(model)
```

inspect.lmer

*Plots to Inspect fitted Models***Description**

This functions can be used to validate some assumptions of (generalized) linear mixed effects models fitted using the `lmer` function from package 'lme4'.

Usage

```
## S3 method for class 'lmer':
inspect(mymodel, which = c("select", "sequence", "all"), id = c("all", "none"), ...
```

Arguments

| | |
|----------------------|---|
| <code>mymodel</code> | an object of class <code>lmer</code> , usually the result of a call to the function <code>lmer</code> |
| <code>which</code> | a character string; do you want to <i>select</i> the plots from a list of all possible diagnostics plots available or do you want <i>all</i> of them, one after the other |
| <code>id</code> | a character string or numeric value; in which panel should it be possible to interactively identify values |
| <code>...</code> | further arguments |

Details

For `lmer` available:

description

| | | |
|---|-----|------------------------------------|
| Normal quantile quantile plot of residuals by levles of categorical fixed effects | nrp | short & long function names |
| Box- or dotplot of residuals by levels, for each random factor | rgp | residuals.by.groups |
| Residuals vs. fitted values for all categorical fixed effects | rfp | Residuals.Fitted.CategoricalFixedE |
| Normal quantile quantile plot of random effects, for each factor | nep | NormalQuantiles.RandomEffectsQ |

Value

A list with identified values (row indices), one named slot (short name) for each inspection function and an additional slot (called `all`) with all values that were selected at lest in one plot.

Note

identification of points is not implemented yet

Author(s)

<thomas.fabbro@unibas.ch>

See Also

`plot`

Examples

```
## data(wellplate)
## model <- lmer(int ~ trt + (1|row) + (1|col), data=wellplate)
## inspect(model)
```

lep

linearized response vs. linear predictor plot

Description

There is a separate help page for every generic

Usage

```
lep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class <code>glm</code> , usually, a result of a call to the function <code>glm</code> . |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

⟨thomas.fabbro@unibas.ch⟩

See Also

[lep.glm](#), [inspect](#)

Examples

```
##
```

 lep-methods

Methods for Function lep in Package 'asuR'

Description

Methods for function lep in Package 'asuR'

Methods

mymodel = "ANY" generic function

mymodel = "glm" Diagnostics plots for objects of class "glm"

 lep.glm

linearized response vs. linear predictor (eta) plot

Description

A plot to check the link assumption in a generalized linear model.

Usage

```
## S3 method for class 'glm':
lep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class glm, usually, a result of a call to the function glm. |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

after violations of simpler model assumptions have been eliminated.

***Question:

Is the relationship linear? otherwise

1. change the link function

Value

A plot and a vector with identified values (corresponding to the row number in the original data).

Author(s)

<thomas.fabbro@unibas.ch>

See Also[inspect](#)**Examples**

#

`mancontr`*Contrast Matrix Construction*

Description

Helps you to construct a contrast matrix and naming the contrasts

Usage

```
mancontr(contr = NULL, contr.names = NULL)
```

Arguments

`contr` a list of contrasts (given as vectors) or a matrix with contrasts given in (k-1) rows or (k-1) columns (k is the number of levels of a factor)

`contr.names` a list or vector of contrast names

Details

For factors with more than two levels, comparisons among different levels are usually of interest (also comparisons among means of different levels). Therefore you need to be able to set your contrasts manually. Because it is also important to give your contrasts meaningful names. This is especially easy with this function. You can either provide them as `row.names` directly in the contrast matrix or in separate vector or list.

Value

A (k times k-1) matrix

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also[contrasts](#), [C](#)

Examples

```

data(pea)
trt.contr <- rbind("control-sugar"=c(1, -1/4, -1/4, -1/4, -1/4),
                  "pure-mixed"=c(0, 1/3, 1/3, -1, 1/3),
                  "monosaccharides-disaccharides"=c(0,1/2,1/2,0,-1),
                  "gluc-fruc"=c(0,1,-1,0,0))
model <- aov(length ~ trt,
             contrasts=list(trt=mancontr(contr=trt.contr)), data=pea)

### ALTERNATIVE formulations:
## contrasts in a list:
# trt.contr <- list(c( 1, -1/4, -1/4, -1/4, -1/4),
#                  c( 0, 1/3, 1/3, -1, 1/3),
#                  c( 0, 1/2, 1/2, 0, -1),
#                  c( 0, 1, -1, 0, 0))
## contrasts in a matrix
# trt.contr <- rbind(c( 1, -1/4, -1/4,-1/4, -1/4),
#                  c( 0, 1/3, 1/3, -1, 1/3),
#                  c( 0, 1/2, 1/2, 0, -1),
#                  c( 0, 1, -1, 0, 0))
## names of contrasts in a list:
# trt.contr.names=list("control-sugar", "pure-mixed",
#                      "monosaccharides-disaccharides", "gluc-fruc")
## names of contrasts in a vector:
# trt.contr.names=c("control-sugar", "pure-mixed",
#                   "monosaccharides-disaccharides", "gluc-fruc")
#model <- aov(length ~ trt,
#             contrasts=list(trt=mancontr(contr=trt.contr, contr.names=trt.contr.names)),
#             data=pea)

```

mytrees

Simulated tree data

Description

Similar to the data frame called trees in the dataset package. But simulated data for learning regressions.

Usage

```
data(mytrees)
```

Format

A data frame with 100 observations on the following 3 variables.

Volume a numeric vector

Girth a numeric vector

Height a numeric vector

Examples

```
data(mytrees)
```

nep

Normal quantile quantile plot of random effects, for each factor

Description

There is a separate help page for every generic.

Usage

```
nep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>lmer</code> , usually, a result of a call to the function <code>lmer</code> . |
| <code>id</code> | a character string or numeric value; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[nep.lmer](#), [inspect](#)

Examples

```
##
```

 nep-methods

Methods for Function nep in Package 'asuR'

Description

Methods for function nep in Package 'asuR'

Methods

mymodel = "ANY" generic function

mymodel = "lmer" see the documentation of nep.lmer

 nep.lmer

Normal quantile quantile plot of random effects, for each factor

Description

A normal quantile quantile plot of the random effects.

Usage

```
## S3 method for class 'lmer':
nep(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|---------|---|
| mymodel | an object of class lmer, usually, a result of a call to the function lmer. |
| id | a character string; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

***Questions:

A) Do the points lie on a line?

Value

A plot

Author(s)

<thomas.fabbro@unibas.ch>

See Also

[inspect](#)

Examples

```
##
```

```
norm.test
```

Inspect your residuals after transformation

Description

This function shows you nine normal quantile-quantile plots. Eight are sampled from a normal distribution and one is showing your data. Do you recognise your data?

Usage

```
norm.test(x)
```

Arguments

`x` a numeric vector with your data

Value

Nine quantile-quantile plots on your active device.

Author(s)

<thomas.fabbro@unibas.ch>

Examples

```
#norm.test(rchisq(30,df=10))
```

```
nrp
```

Normal quantile quantile plot of residuals by levels of categorical fixed effects

Description

There is a separate help page for every generic.

Usage

```
nrp(mymodel, id = c("all", "none"), ...)
```

Arguments

`mymodel` an object of class `lmer`, usually, a result of a call to the function `lmer`.
`id` a character string or numeric value; in which panel should it be possible to interactively identify points
`...` further arguments

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

`nrp.lmer`, `inspect`

Examples

```
##
```

| | |
|--------------------------|---|
| <code>nrp-methods</code> | <i>Methods for Function nrp in Package 'asuR'</i> |
|--------------------------|---|

Description

Methods for function `nrp` in Package 'asuR'

Methods

`mymodel = "ANY"` generic function
`mymodel = "lmer"` see the documentation of `nrp.lmer`

| | |
|-----------------------|---|
| <code>nrp.lmer</code> | <i>Normal quantile quantile plot of random effects, for each factor</i> |
|-----------------------|---|

Description

A normal quantile quantile plot of the random effects.

Usage

```
## S3 method for class 'lmer':
nrp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>lmer</code> , usually, a result of a call to the function <code>lmer</code> . |
| <code>id</code> | a character string; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

***Questions:

A) Do the points lie on a line?

Value

A plot

Author(s)

<thomas.fabbro@unibas.ch>

See Also

[inspect](#)

Examples

```
##
```

oring

O-ring data

Description

Data of O-ring failures in different tests

Usage

```
data(oring)
```

Format

A data frame with 24 observations on the following 4 variables.

date a Date

temperature a numeric vector

failures a numeric vector

fail a factor with levels no yes

Details

Space shuttle Challenger exploded right at the beginning of its flight on January 28, 1986. This was one of the largest disasters in the American space program. The night before, an engineer had recommended to NASA (National Aeronautics and Space Administration) that the shuttle should not be launched in the cold weather. Forecast of temperature for the launch was 31 degrees Fahrenheit, the coldest launch ever. This suggestion was over-ruled. Inquiry Commission appointed by the President of the United States, wanted to see if enough evidence existed to predict serious trouble due to low temperature at the time of launch. Since the shuttle had, up to that time, not met with

any accident, the only evidence available was regarding damage to O-rings. These rubber rings fill the gaps between parts of the giant tube that makes the rocket. If there is even a minor leak, hot gases push through it and in milliseconds, large portion of the rocket fuel can come out to destroy the rocket. Hence damaged O-rings can be treated as signs of major trouble. Such instances had indeed been recorded in previous flights of the shuttle. Data are to be analyzed to check if statistical methods would have given the right guidance.

Source

from <http://wps.aw.com/wps/media/objects/15/15719/projects/>, without additional information about the source

Examples

```
data(oring)
model.glm <- glm(fail ~ temperature, data=oring, family=binomial)
```

| | |
|--------------|--|
| parseFormula | <i>providing names and data from model objects</i> |
|--------------|--|

Description

internal use only

Usage

```
## S3 method for class 'lm':
parseFormula(mymodel, ...)
```

Arguments

| | |
|---------|-----------------------|
| mymodel | a fitted model object |
| ... | further arguments |

Value

| | |
|--------------------------|---|
| response.var | the name of the response variable |
| response.term | the name with transformation of the response variable |
| predict.vars.numeric | the name of the numeric predictor variables |
| predict.terms.numeric | the name with transformation of the numeric predictor variables |
| index_coef.terms.numeric | the index of the numeric terms in the vector returned by coef() |
| my.data | the data used for model fitting |
| response.values | the values of the response variable |

```
intercept.logical
      logical flag: is there an intercept
```

Note

under construction

Author(s)

⟨thomas.fabbro@unibas.ch⟩

See Also

[inspect](#)

Examples

```
##
```

| | |
|---------------------------|--|
| <code>parseFormula</code> | <i>providing names and data from model objects</i> |
|---------------------------|--|

Description

internal use only

Usage

```
parseFormula(mymodel, ...)
```

Arguments

| | |
|----------------------|-----------------------|
| <code>mymodel</code> | a fitted model object |
| <code>...</code> | further arguments |

Details

none

Value

for `glm`, and if possible for all others:

| | |
|------------------------------------|---|
| <code>response.var</code> | the name of the response variable |
| <code>response.term</code> | the name with transformation of the response variable |
| <code>predict.vars.numeric</code> | the name of the numeric predictor variables |
| <code>predict.terms.numeric</code> | the name with transformation of the numeric predictor variables |

`index_coef.terms.numeric`
the index of the numeric terms in the vector returned by `coef()`

`my.data` the data used for model fitting

`response.values`
the values of the response variable

`intercept.logical`
logical flag: is there an intercept

Note

under construction

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[inspect](#)

Examples

```
##
```

parseFormula-methods

Methods for Function parseFormula in Package 'asuR'

Description

Methods for function `parseFormula` in Package 'asuR'

Methods

`mymodel = "ANY"` generic function

`mymodel = "lm"` see its man page

`mymodel = "glm"` see its man page

`mymodel = "lmer"` see its man page

parseFormula.glm *providing names and data from model objects*

Description

internal use only

Usage

```
## S3 method for class 'glm':  
parseFormula(mymodel, ...)
```

Arguments

| | |
|---------|-----------------------|
| mymodel | a fitted model object |
| ... | further arguments |

Note

under construction

Author(s)

<thomas.fabbro@unibas.ch>

See Also

[inspect](#)

Examples

```
##
```

parseFormula.lmer *providing names and data from model objects*

Description

internal use only

Usage

```
## S3 method for class 'lmer':  
parseFormula(mymodel, ...)
```

Arguments

| | |
|---------|-----------------------|
| mymodel | a fitted model object |
| ... | further arguments |

Note

under construction

Author(s)

<thomas.fabbro@unibas.ch>

See Also

[inspect](#)

Examples

```
##
```

pea

Pea data

Description

The effect of different sugars on length, in ocular units ($\times 0.114 = \text{mm}$), of pea sections grown in tissue culture with auxin present.

Usage

```
data(pea)
```

Format

A data frame with 50 observations on the following 2 variables.

trt a factor with levels `cont`: control, `fruc`: 2% fructose added, `gluc`: 2% glucose added, `glucfruc`: 1% glucose and 1% fructose added, `sucr`: 2% sucrose added

length a numeric vector

Source

Sokal, R. R. & Rohlf, J. F. (1995): *Biometry: the principles and practice of statistics in biological research*. Freeman, New York (p 218).

Examples

```
data(pea)
```

| | |
|--------|---------------------|
| plants | <i>Plant height</i> |
|--------|---------------------|

Description

The height of all species of Fabaceae and Rosaceae growing in Switzerland. As well as a factor indicating whether the species can grow to a shrub, tree, or stays herbaceous.

Usage

```
data(plants)
```

Format

A data frame with 460 observations on the following 3 variables.

family a factor with levels Fabaceae Rosaceae

height a numeric vector

type a factor with levels herb shrub tree

Source

thomas.fabbro@unibas.ch

Examples

```
data(plants)
tapply(plants$height, list(plants$family, plants$type), mean)
tapply(plants$height, list(plants$family, plants$type), summary)

tapply(plants$height, list(plants$family, plants$type), length)

table(plants$family, plants$type)
```

| | |
|-----|--|
| rfp | <i>Residuals vs. fitted values for all categorical fixed effects</i> |
|-----|--|

Description

There is a separate help page for every generic.

Usage

```
rfp(mymodel, id = c("all", "none"), ...)
```

Arguments

`mymodel` an object of class `lmer`, usually, a result of a call to the function `lmer`.
`id` a character string or numeric value; in which panel should it be possible to interactively identify points
`...` further arguments

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[rfp.lmer](#), [inspect](#)

Examples

```
##
```

rfp-methods

Methods for Function rfp in Package 'asuR'

Description

Methods for function `rfp` in Package 'asuR'

Methods

mymodel = "ANY" generic function

mymodel = "lmer" see the documentation of `rfp.lmer`

`rfp.lmer`*Residuals vs. fitted values for all categorical fixed effects*

Description

A scatterplot of residuals vs. fitted values for all categorical fixed effects. THE IDENTIFICATION IS NOT YET IMPLEMENTED!

Usage

```
## S3 method for class 'lmer':  
rfp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>lmer</code> , usually, a result of a call to the function <code>lmer</code> . |
| <code>id</code> | a character string; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

***Questions:
A)

Value

A plot

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[inspect](#)

Examples

```
##
```

 rgp

Box- or dotplot of residuals by groups, for each random factor

Description

There is a separate help page for every generic.

Usage

```
rgp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class <code>lmer</code> , usually, a result of a call to the function <code>lmer</code> . |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

⟨thomas.fabbro@unibas.ch⟩

See Also

[rgp.lmer](#), [inspect](#)

Examples

```
##
```

 rgp-methods

Methods for Function rgp in Package 'asuR'

Description

Methods for function `rgp` in Package 'asuR'

Methods

mymodel = "ANY" generic function
mymodel = "lmer" see the documentation of `rgp.lmer`

`rgp.lmer`*Box- or dotplot of residuals by groups for each random factor*

Description

A Boxplot (or if less than ten observations a dotplot) of the residuals at every level of each random factor. THE IDENTIFICATION IS NOT YET IMPLEMENTED!

Usage

```
## S3 method for class 'lmer':  
rgp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|--|
| <code>mymodel</code> | an object of class <code>lmer</code> , usually, a result of a call to the function <code>lmer</code> . |
| <code>id</code> | a character string; in which panel should it be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

***Questions:

A) Are the residuals centered at zero? otherwise:

- 1. B) is the variance constant? otherwise:

- a) if the variance changes among levels of a fixed factor

- - 1) use a heteroscedastic model (only in function `lme` from package `nlme`) - b)

Value

A plot

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[inspect](#)

Examples

```
##
```

`rpp`*partial residual vs. each predictor plot*

Description

There is a separate help page for every generic.

Usage

```
rpp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|----------------------|---|
| <code>mymodel</code> | an fitted model object |
| <code>id</code> | a character string or numeric value; indicating in which panel it should be possible to interactively identify points |
| <code>...</code> | further arguments |

Details

There is a separate help page for every generic.

Value

A plot and an invisible vector of row indices (in the original data) of identified values.

Author(s)

`<thomas.fabbro@unibas.ch>`

See Also

[rpp.glm](#), [inspect](#)

Examples

```
##
```

 rpp-methods

 Methods for Function rpp in Package 'asuR'

Description

Creates a plot of partial residuals against all predictors.

Methods

mymodel = "ANY" Generic function

mymodel = "lm" Generates a plot of an object of class "lm"

mymodel = "glm" Generates a plot of an object of class "glm"

 rpp.glm

 Partial Residual Plots

Description

Plots the partial residuals of a glm-model versus each predictor

Usage

```
## S3 method for class 'glm':
rpp(mymodel, id=c("all", "none"), ...)
```

Arguments

| | |
|---------|---|
| mymodel | an object of class glm, usually, a result of a call to the function glm. |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points; not yet available |
| ... | further arguments |

Details

A partial residual plot allows to study the effect of a focal predictor and taking the other predictors into account. This helps to find an appropriate transformation for the focal predictor.

***expected pattern:

linear scatter of points along the regression line

***Question:

Is the relationship linear? otherwise:

1. change the transformations of the predictor

Value

A plot and a vector with the row index of the identified values in the original data.

Author(s)

⟨thomas.fabbro@unibas.ch⟩

See Also

[inspect](#)

Examples

```
#
```

rpp.lm *partial residual vs. each predictor plot*

Description

Plots residuals of a linear model against each predictor.

Usage

```
## S3 method for class 'lm':  
rpp(mymodel, id = c("all", "none"), ...)
```

Arguments

| | |
|---------|--|
| mymodel | an object of class <code>lm</code> , usually, a result of a call to the function <code>lm</code> . |
| id | a character string or numeric value; in which panel should it be possible to interactively identify points |
| ... | further arguments |

Details

The shape of the scatter can help to find a appropriate transformation for the focal predictors.

Value

A plot and a vector with the row index of the identified values in the original data.

Author(s)

⟨thomas.fabbro@unibas.ch⟩

See Also

[inspect](#)

Examples

```
##
```

| | |
|-------------|----------------------------|
| schoolclass | <i>Schools and Classes</i> |
|-------------|----------------------------|

Description

A data set for studying nested random factors.

Usage

```
data(schoolclass)
```

Format

A data frame with 300 observations on the following 4 variables.

school a factor with 300 levels

class a factor with 300 levels

gender a factor with levels `female` `male`

score a numeric vector

Details

A simulated data set with the parameters: variance among schools (100), variance among classes (50), residual variance (10), factor gender: level female (200), level male (100).

Source

thomas.fabbro@unibas.ch

Examples

```
data(schoolclass)
str(schoolclass)
#
#with library(lme4)
# (m0 <- lmer(y ~ gender + (1|school/class), data=schoolclass))
```

| | |
|--------------|---|
| unemployment | <i>data on long / short term unemployment</i> |
|--------------|---|

Description

Data set with success (long term) unemployment against total number o unemployment.

Usage

```
data(unemployment)
```

Format

A data frame with 983 observations on the following 2 variables.

success a factor with levels <6m >6m

gender a factor with levels female male

Details

no

Source

GLM course notes by Gerhard Tutz (have a look at his nice book!)

Examples

```
data(unemployment)
contrast.matrix <- rbind("male-female"=c(1,-1))

u.glm <- glm(success~gender, data=unemployment, family=binomial, contrasts=list(gender=mycon

### for data in the grouped form
#####
### we make a data set in grouped form
table(unemployment$gender, unemployment$success)
unemployment.grouped <- data.frame(longterm=c(167,175), shortterm=c(403,238), gender=c("male

u.glm.grouped <- glm(cbind(longterm,shortterm) ~ gender, data=unemployment.grouped, family=b
summary(u.glm.grouped)
coefficients(u.glm.grouped)

### extracting the factors by which the odds change from male to female
factor <- exp(coefficients(u.glm.grouped)[2])

### the "same" calculations by hand:
attach(unemployment.grouped)
p <- longterm/(shortterm+longterm)
detach(unemployment.grouped)
odds <- p/(1-p)

odds[1]*factor
### should give the same as
odds[2]

### for those who like p-values
#####
anova(u.glm, test="Chisq")
### is the same
anova(u.glm.grouped, test="Chisq")
### the same p-value
pchisq(17.959,1,lower.tail=FALSE)
### and also almost the same p-value as with a chisq.test()...that you
```

```
### can try yourself!
```

weight

Weight Gain

Description

The data arise from an experiment to study the gain in weight of rats fed on four different diets, distinguished by amount of protein (low and high) and by source of protein (beef and cereal).

Usage

```
data(weight)
```

Format

A data frame with 40 observations on the following 3 variables.

source a factor with levels `Beef Cereal`; source of the protein fed

type a factor with levels `High Low`, amount of the protein fed

weightgain a numeric vector; weight gained in grams

Details

Ten rats are randomized to each of the four treatments. The question of interest is how diet affects weight gain.

Source

D. J. Hand, F. Daly, A. D. Lunn, K. J. McConway and E. Ostrowski (1994). A Handbook of Small Datasets, Chapman and Hall/CRC, London.

References

```
from library(HSAUR)
```

Examples

```
data(weight)
interaction.plot(weight$type, weight$source,
                 weight$weightgain)
```

`wellplate`*wellplate*

Description

An example data set with two random variables that are crossed (row and column). And one fixed variable a treatment.

Usage

```
data(wellplate)
```

Format

A data frame with 96 observations on the following 4 variables.

row a factor with 8 levels

col a factor with 12 levels

trt a factor with levels `contr` `treat`

int a numeric vector

Details

A simulated data set with the parameters: variance among rows (50), variance among columns (150), residual variance (10), factor trt: levels: "contr" (100), "treat" (200).

DATASETS with different column variance structure:

wellplate2: variance among columns ("treat" 150; "contr" 0; covariance 0)

wellplate3: variance among columns ("treat" 150; "contr" 150; covariance 0)

Source

simulated by thomas.fabbro@unibas.ch

Examples

```
data(wellplate)
str(wellplate)
# library(lme4)
# (model <- lmer(int ~ 1 + (1|row) + (1|col), data=wellplate))
# (model <- lmer(int ~ 1 + (1|row) + (1|col), data=wellplate))
# (model2 <- lmer(int ~ trt + (1|row) + (1|trt:col), data=wellplate2))
# (model3 <- lmer(int ~ trt + (1|row) + (trt-1|col), data=wellplate3))
```

`xxp`*Plot all explanatory variables against each other*

Description

A matrix with scatterplots and boxplots of all variables of a data set against each other.

Usage

```
xxp(mydata)
```

Arguments

`mydata` a data frame

Details

Similar to the function `pairs`, but all variables are only plotted once against each other. This is especially of advantage, if a data set has many variables. Depending on the class of the variables different plots are produced: scatterplots if both are numeric, boxplots if one is numeric and the other is categorical

Value

a plot

Author(s)

`<thomas.fabbro@unibas.ch>`

Index

*Topic **datasets**

- BtheB, 2
- budworm, 3
- cathedral, 5
- flowers, 9
- gala, 10
- growth, 10
- houseflies, 13
- mytrees, 21
- oring, 26
- pea, 31
- plants, 32
- schoolclass, 40
- unemployment, 40
- weight, 42
- wellplate, 43

*Topic **file**

- colVars, 5
- inspect, 14
- inspect.glm, 15
- inspect.lmer, 17
- mancontr, 20
- norm.test, 24
- xxp, 44

*Topic **methods**

- dep-methods, 7
- hnp-methods, 12
- inspect-methods, 15
- lep-methods, 19
- nep-methods, 23
- nrp-methods, 25
- parseFormula-methods, 29
- rfp-methods, 33
- rgp-methods, 35
- rpp-methods, 38

*Topic **models**

- dep, 6
- dep.glm, 8
- hnp, 11

- hnp.glm, 12
- lep, 18
- lep.glm, 19
- nep, 22
- nep.lmer, 23
- nrp, 24
- nrp.lmer, 25
- parseFormula, 27, 28
- parseFormula.glm, 30
- parseFormula.lmer, 30
- rfp, 32
- rfp.lmer, 34
- rgp, 35
- rgp.lmer, 36
- rpp, 37
- rpp.glm, 38
- rpp.lm, 39

*Topic **package**

- asuR-package, 2

- app (*inspect*), 14
- asuR (*asuR-package*), 2
- asuR-package, 2

- BtheB, 2
- BtheBlong (*BtheB*), 2
- budworm, 3

- C, 20
- cathedral, 5
- colMeans, 6
- colVars, 5
- contrasts, 20

- dep, 6
- dep, ANY-method (*dep-methods*), 7
- dep, glm-method (*dep-methods*), 7
- dep-methods, 7
- dep.glm, 7, 8
- devianceResidual.linearPredictor
(*dep*), 6

- flowers, 9
- gala, 10
- growth, 10
- halfNormalQuantiles.absoluteStudentizedResiduals
(*hnp*), 11
- hnp, 11
- hnp, ANY-method (*hnp-methods*), 12
- hnp, glm-method (*hnp-methods*), 12
- hnp-methods, 12
- hnp.glm, 11, 12
- houseflies, 13
- ihp (*inspect*), 14
- ilp (*inspect*), 14
- inspect, 7, 8, 11, 13, 14, 18, 20, 22, 23, 25,
26, 28–31, 33–37, 39
- inspect, ANY-method
(*inspect-methods*), 15
- inspect, glm-method
(*inspect-methods*), 15
- inspect, lm-method
(*inspect-methods*), 15
- inspect, lmer-method
(*inspect-methods*), 15
- inspect-methods, 15
- inspect.glm, 15
- inspect.lmer, 17
- irp (*inspect*), 14
- lep, 18
- lep, ANY-method (*lep-methods*), 19
- lep, glm-method (*lep-methods*), 19
- lep-methods, 19
- lep.glm, 18, 19
- linearizedResponse.linearPredictor
(*lep*), 18
- mancontr, 20
- mycontr (*mancontr*), 20
- mytrees, 21
- nep, 22
- nep, ANY-method (*nep-methods*), 23
- nep, lmer-method (*nep-methods*), 23
- nep-methods, 23
- nep.lmer, 22, 23
- norm (*norm.test*), 24
- norm.test, 24
- NormalQuantiles.RandomEffectsQuantiles
(*nep*), 22
- NormalQuantiles.Residuals.CategoricalFixedEffects
(*nrp*), 24
- nrp, 24
- nrp, ANY-method (*nrp-methods*), 25
- nrp, lmer-method (*nrp-methods*), 25
- nrp-methods, 25
- nrp.lmer, 25, 25
- oring, 26
- parseFormula, 27, 28
- parseFormula, ANY-method
(*parseFormula-methods*), 29
- parseFormula, glm-method
(*parseFormula-methods*), 29
- parseFormula, lm-method
(*parseFormula-methods*), 29
- parseFormula, lmer-method
(*parseFormula-methods*), 29
- parseFormula-methods, 29
- parseFormula.glm, 30
- parseFormula.lmer, 30
- partialResidual.eachPredictor
(*rpp*), 37
- pea, 31
- plants, 32
- prp (*inspect*), 14
- residuals.by.groups (*rgp*), 35
- Residuals.Fitted.CategoricalFixedEffects
(*rfp*), 32
- rfp, 32
- rfp, ANY-method (*rfp-methods*), 33
- rfp, lmer-method (*rfp-methods*), 33
- rfp-methods, 33
- rfp.lmer, 33, 34
- rgp, 35
- rgp, ANY-method (*rgp-methods*), 35
- rgp, lmer-method (*rgp-methods*), 35
- rgp-methods, 35
- rgp.lmer, 35, 36
- rpp, 37
- rpp, ANY-method (*rpp-methods*), 38
- rpp, glm-method (*rpp-methods*), 38
- rpp, lm-method (*rpp-methods*), 38
- rpp-methods, 38
- rpp.glm, 37, 38

rpp.lm, [39](#)
ryp(*inspect*), [14](#)

schoolclass, [40](#)

unemployment, [40](#)

weight, [42](#)
wellplate, [43](#)
wellplate2(*wellplate*), [43](#)
wellplate3(*wellplate*), [43](#)

xyp, [44](#)