

Package ‘JWileymisc’

September 15, 2018

Type Package

Title Miscellaneous Utilities and Functions

Version 0.3.1

Date 2018-09-15

Author Joshua F. Wiley <jwiley.psych@gmail.com>

Maintainer Joshua F. Wiley <jwiley.psych@gmail.com>

URL <https://github.com/JWiley/JWileymisc>

BugReports <https://github.com/JWiley/JWileymisc/issues>

Description A collection of miscellaneous tools and functions, such as tools to generate descriptive statistics tables, format output, visualize relations among variables or check distributions.

License GPL (>= 3)

Depends R (>= 3.4.0), data.table (>= 1.10.0), ggplot2 (>= 2.2.0), cowplot, foreach

Imports stats, utils, MASS, multcompView, emmeans, devtools, graphics, ggthemes, mgcv, mice, methods, zoo, psych, rms, robustbase, quantreg, lavaan (>= 0.5-20), VGAM (>= 1.0-6), nlme, Matrix, lme4, lmerTest

Suggests scales, testthat

RoxygenNote 6.1.0

NeedsCompilation no

Repository CRAN

Date/Publication 2018-09-15 21:40:03 UTC

R topics documented:

.detailedTestsLMER	3
.detailedTestsVGLM	4
aces_daily	5

acfByID	6
APAStyler	8
APAStyler.lm	8
APAStyler.mira	9
APAStyler.SEMSummary	9
as.na	10
cd	11
compareIVs	12
compareLMER	13
cor2cov	14
corOK	14
corplot	15
detailedTests	17
egtable	18
empirical_pvalue	20
findSigRegions	20
formatHtest	21
formatLMER	22
formatMedIQR	23
formatPval	24
gglikert	25
iccMixed	26
intSigRegGraph	27
lagk	28
logicals	28
meanDecompose	30
meanDeviations	31
moments	32
mvqq	33
nEffective	34
param_summary	35
param_summary_format	36
plot.SEMSummary	37
plot.SEMSummary.list	37
plotDiagnosticsLM	38
plotDiagnosticsLMER	39
R2LMER	40
roundedfivenum	41
SEMSummary	41
SEMSummary.fit	43
star	44
testdistr	45
timeshift	47
TukeyHSDgg	48
updateInstall	49
VACConverter	50
VAObject-class	51
VASummaryObject-class	52

`.detailedTestsLMER` 3

`winsorizer` 53

Index 55

<code>.detailedTestsLMER</code>	<i>estimate detailed results per variable and effect sizes for both fixed and random effects from lmer models</i>
---------------------------------	---

Description

This function extends the current `drop1` method for `merMod` class objects from the `lme4` package. Where the default method to be able to drop both fixed and random effects at once.

Usage

```
.detailedTestsLMER(obj, method = c("Wald", "profile", "boot"), ...)
```

Arguments

- `obj` A `merMod` class object, the fitted result of `lmer`.
- `method` A character vector indicating the types of confidence intervals to calculate. One of "Wald", "profile", or "boot".
- `...` Additional arguments passed to `confint`

Details

At the moment, the function is aimed to `lmer` models and has very few features for `glmer` or `nLmer` models. The primary motivation was to provide a way to provide an overall test of whether a variable "matters". In multilevel data, a variable may be included in both the fixed and random effects. To provide an overall test of whether it matters requires jointly testing the fixed and random effects. This also is needed to provide an overall effect size.

The function works by generating a formula with one specific variable or "term" removed at all levels. A model is then fit on this reduced formula and compared to the full model passed in. This is a complex operation for mixed effects models for several reasons. Firstly, R has no default mechanism for dropping terms from both the fixed and random portions. Secondly, mixed effects models do not accomodate all types of models. For example, if a model includes only a random slope with no random intercept, if the random slope was dropped, there would be no more random effects, and at that point, `lmer` or `glmer` will not run the model. It is theoretically possible to instead fit the model using `lm` or `glm` but this becomes more complex for certain model comparisons and calculations and is not currently implemented. Marginal and conditional R2 values are calculated for each term, and these are used also to calculate something akin to an f-squared effect size.

This is a new function and it is important to carefully evaluate the results and check that they are accurate and that they are sensible. Check accuracy by viewing the model formulae for each reduced model and checking that those are indeed accurate. In terms of checking whether a result is sensible or not, there is a large literature on the difficulty interpreting main effect tests in the presence of interactions. As it is challenging to detect all interactions, especially ones that are made outside of R formulae, all terms are tested. However, it likely does not make sense to report results from dropping a main effect but keeping the interaction term, so present and interpret these with caution.

Examples

```
## Not run:
data(aces_daily)
m1 <- lme4::lmer(NegAff ~ STRESS + (1 + STRESS | UserID),
  data = aces_daily)
m2 <- lme4::lmer(NegAff ~ STRESS + I(STRESS^2) + (1 + STRESS | UserID),
  data = aces_daily)
testm1 <- .detailedTests(m1, method = "profile")
testm2 <- .detailedTests(m2, method = "profile")
testm2b <- .detailedTests(m2, method = "boot", nsim = 100)

## End(Not run)
```

<code>.detailedTestsVGLM</code>	<i>Calculates all pairwise contrasts and omnibus tests for multinomial regression</i>
---------------------------------	---

Description

TODO: make me!

Usage

```
.detailedTestsVGLM(obj, OR = TRUE, digits = 2L, pdigits = 3L)
```

Arguments

<code>obj</code>	A <code>vglm</code> class object, the fitted result of <code>vglm()</code> . At the moment only handles the multinomial family, although this may get expanded in the future.
<code>OR</code>	a logical value whether to report odds ratios and 95 percent confidence intervals, if <code>TRUE</code> , or regression coefficients on the logit scale with standard errors, if <code>FALSE</code> .
<code>digits</code>	An integer indicating the number of digits for coefficients, standard errors, and confidence intervals
<code>pdigits</code>	An integer indicating the number of digits for p-values.

Value

A list with two elements. `Results` contains a data table of the actual estimates. `Table` contains a nicely formatted character matrix.

Examples

```

mtcars$cyl <- factor(mtcars$cyl)
m <- VGAM::vglm(cyl ~ qsec,
  family = VGAM::multinomial(), data = mtcars)
.detailedTestsVGLM(m)

rm(m, mtcars)

## Not run:
mtcars$cyl <- factor(mtcars$cyl)
mtcars$am <- factor(mtcars$am)
m <- VGAM::vglm(cyl ~ qsec,
  family = VGAM::multinomial(), data = mtcars)
.detailedTestsVGLM(m)

.detailedTestsVGLM(m, digits = 4)$Table
.detailedTestsVGLM(m, OR = FALSE)
.detailedTestsVGLM(m, digits = 4, OR = FALSE)$Table

m <- VGAM::vglm(cyl ~ scale(qsec),
  family = VGAM::multinomial(), data = mtcars)
.detailedTestsVGLM(m)

m2 <- VGAM::vglm(cyl ~ factor(vs) * scale(qsec),
  family = VGAM::multinomial(), data = mtcars)
.detailedTestsVGLM(m2)

m <- VGAM::vglm(Species ~ Sepal.Length,
  family = VGAM::multinomial(), data = iris)
.detailedTestsVGLM(m)

## End(Not run)

```

aces_daily

Multilevel Daily Data Example

Description

A data frame drawn from a daily diary study, conducted at Monash University in 2017 where young adults old completed measures up to three times per day (morning, afternoon, and evening) for about 12 days. Thus each participant contributed about 36 observations to the dataset. To protect participant confidentiality and anonymity, the data used here were simulated from the original data, but in such a way as to preserve the relations among variables and most features of the raw data.

Usage

aces_daily

Format

A data frame containing 19 variables.

UserID A unique identifier for each individual

SurveyDay The date each observation occurred on

SurveyInteger The survey coded as an integer (1 = morning, 2 = afternoon, 3 = evening)

SurveyStartTime Survey start time, centered at time since 11am

Female A 0 or 1 variable, where 1 = female and 0 = male

Age Participant age in years, top coded at 25

BornAUS A 0 or 1 variable where 1 = born in Australia and 0 = born outside of Australia

SES_1 Participants subjective SES, bottom coded at 4 and top coded at 8

EDU Participants level of education (1 = university graduate or higher, 0 = less than university graduate)

SOLs Self-reported sleep onset latency in minutes, morning survey only

WASONS Self-reported number of awakenings after sleep onset, top coded at 4, morning survey only

STRESS Overall stress ratings on a 0–10 scale, repeated 3x daily

SUPPORT Overall social support ratings on a 0–10 scale, repeated 3x daily

PosAff Positive affect ratings on a 1–5 scale, repeated 3x daily

NegAff Negative affect ratings on a 1–5 scale, repeated 3x daily

COPEPrb Problem focused coping on a 1–4 scale, repeated 1x daily at the evening survey

COPEPrc Emotional processing coping on a 1–4 scale, repeated 1x daily at the evening survey

COPEExp Emotional expression coping on a 1–4 scale, repeated 1x daily at the evening survey

COPEDis Mental disengagement coping on a 1–4 scale, repeated 1x daily at the evening survey

 acfByID

Estimate the effective sample size from longitudinal data

Description

This function estimates the (approximate) effective sample size.

Usage

```
acfByID(xvar, timevar, idvar, data, lag.max = 10L,
        na.function = c("na.approx", "na.spline", "na.locf"), ...)
```

Arguments

xvar	A character string giving the variable name of the variable to calculate autocorrelations on.
timevar	A character string giving the variable name of the time variable.
idvar	A character string giving the variable name of the ID variable. Can be missing if only one time series provided, in which case one will be created.
data	A data.table containing the variables used in the formula. This is a required argument. If a data.frame, it will silently coerce to a data.table. If not a data.table or data.frame, it will attempt to coerce, with a message.
lag.max	An integer of the maximum lag to estimate. Must be equal to or greater than the number of observations for all IDs in the dataset.
na.function	A character string giving the name of the function to use to address any missing data. Functions come from the zoo package, and must be one of: “na.approx”, “na.spline”, “na.locf”.
...	Additional arguments passed to zoo.

Value

A data.table of the estimated autocorrelations by ID and lag

References

For details, see Campbell, M. K., Mollison, J., & Grimshaw, J. M. (2001). Cluster trials in implementation research: estimation of intracluster correlation coefficients and sample size. *Statistics in Medicine*, 20(3), 391-399.

Examples

```
## example 1
dat <- data.table(
  x = sin(1:30),
  time = 1:30,
  id = 1)
acfByID("x", "time", "id", data = dat)

## example 2
dat2 <- data.table(
  x = c(sin(1:30), sin((1:30)/10)),
  time = c(1:30, 1:30),
  id = rep(1:2, each = 30))
dat2$x[4] <- NA

res <- acfByID("x", "time", "id", data = dat2, na.function = "na.approx")

ggplot(res, aes(factor(Lag), AutoCorrelation)) +
  geom_boxplot()

## clean up
rm(dat, dat2, res)
```

 APASTyler

A generic function for pretty printing in (semi) APA Style

Description

A generic function for pretty printing in (semi) APA Style

Usage

```
APASTyler(object, ...)
```

Arguments

object	An object with a class matching one of the methods
...	Additional arguments passed on to methods.

 APASTyler.lm

A generic function for pretty printing in (semi) APA Style

Description

A generic function for pretty printing in (semi) APA Style

Usage

```
## S3 method for class 'lm'
APASTyler(object, digits = 2, pdigits, file, ...)
```

Arguments

object	lm object
digits	The number of digits to round results to. Defaults to 2.
pdigits	The number of digits to use for p values. Defaults to digits + 1 if missing.
file	An optional argument indicating whether the output should be written to a file.
...	Additional arguments passed on to write.table.

APASTyler.mira *A generic function for pretty printing in (semi) APA Style*

Description

A generic function for pretty printing in (semi) APA Style

Usage

```
## S3 method for class 'mira'
APASTyler(object, lobject, digits = 2, pdigits, file,
  ...)
```

Arguments

object	mira object
lobject	an lm object the degrees of freedom of which can be used for conservative F tests
digits	The number of digits to round results to. Defaults to 2.
pdigits	The number of digits to use for p values. Defaults to digits + 1 if missing.
file	An optional argument indicating whether the output should be written to a file.
...	Additional arguments passed on to write.table.

APASTyler.SEMSummary *A generic function for pretty printing in (semi) APA Style*

Description

A generic function for pretty printing in (semi) APA Style

Usage

```
## S3 method for class 'SEMSummary'
APASTyler(object, digits = 2, type = c("cov",
  "cor", "both"), stars = FALSE, file = ifelse(.Platform$OS.type ==
  "windows", "clipboard", FALSE), sep = "\t", ...)
```

Arguments

object	SEMSummary object
digits	The number of digits to round results to. Defaults to 2.
type	A character vector giving what to print. Defaults to 'cov', the covariances. Other options are 'cor' and 'both'.
stars	A logical value whether to include significance values as stars (*** p < .001, ** p < .01, * p < .05).
file	An optional argument indicating whether the output should be written to a file.
sep	Character what the separator for the table should be. Defaults to tabs.
...	Additional arguments passed on to write.table.

Examples

```
m <- SEMSummary(~., data = mtcars)
APAStyler(m, type = "cor", stars = FALSE, file = FALSE)
APAStyler(m, type = "cov", stars = FALSE, file = FALSE)
APAStyler(m, type = "both", stars = FALSE, file = FALSE)
APAStyler(m, type = "cor", stars = TRUE, file = FALSE)
APAStyler(m, type = "cov", stars = TRUE, file = FALSE)
APAStyler(m, type = "both", stars = TRUE, file = FALSE)
```

as.na

*Coerces vectors to missing***Description**

Given a vector, convert it to missing (NA) values, where the class of the missing matches the input class. Currently supports character, logical, integer, factor, numeric, times (from **chron**), Date, POSIXct, POSIXlt, and zoo (from **zoo**).

Usage

```
as.na(x)
```

Arguments

x A vector to convert to missing (NA)

Value

a vector the same length as the input with missing values of the same class

Examples

```
str(as.na(1L:5L))
str(as.na(rnorm(5)))
str(as.na(c(TRUE, FALSE)))
str(as.na(as.Date("2017-01-01")))
```

cd	<i>Change directory</i>
----	-------------------------

Description

The function takes a path and changes the current working directory to the path. If the directory specified in the path does not currently exist, it will be created.

Usage

```
cd(base, pre, num)
```

Arguments

base	a character string with the base path to the directory. This is required.
pre	an optional character string with the prefix to add to the base path. Non character strings will be coerced to character class.
num	an optional character string, prefixed by pre. Non character strings will be coerced to character class.

Details

The function has been designed to be platform independent, although it has had limited testing. Path creation is done using `file.path`, the existence of the directory is checked using `file.exists` and the directory created with `dir.create`. Only the first argument, is required. The other optional arguments are handy when one wants to create many similar directories with a common base.

Value

NULL, changes the current working directory

Examples

```
## Not run:
# an example just using the base
cd("~/testdir")

# an example using the optional arguments
base <- "~/testdir"
pre <- "test_"

cd(base, pre, 1)
cd(base, pre, 2)

## End(Not run)
```

compareIVs	<i>Compares the effects of various independent variables on dependent variables</i>
------------	---

Description

Utility to estimate the unadjusted, covariate adjusted, and multivariate adjusted unique contributions of one or more IVs on one or more DVs

Usage

```
compareIVs(dv, type, iv, covariates = character(), data,  
           multivariate = FALSE, ...)
```

Arguments

dv	A character string or vector of the dependent variable(s)
type	A character string or vector indicating the type of dependent variable(s)
iv	A character string or vector giving the IV(s)
covariates	A character string or vector giving the covariate(s)
data	The data to be used for analysis
multivariate	A logical value whether to have models with all IVs simultaneously.
...	Additional arguments passed on to the internal function, <code>.runIt</code> .

Value

A list with all the model results.

Examples

```
test1 <- compareIVs(  
  dv = c("mpg", "disp"),  
  type = c("normal", "normal"),  
  iv = c("hp", "qsec"),  
  covariates = "am",  
  data = mtcars, multivariate = TRUE)  
test1$OverallSummary  
rm(test1)
```

`compareLMER`*Compare two lmer models*

Description

This function provides fit statistics and effect sizes for model comparisons. The models must be nested.

Usage

```
compareLMER(m1, m2)
```

Arguments

`m1` A model estimated by lmer.
`m2` A model estimated by lmer.

Value

a data table with the fit indices for each model and comparing models to each other.

References

For estimating the marginal and conditional R-squared values, see: Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining R² from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142. as well as: Johnson, P. C. (2014). Extension of Nakagawa & Schielzeth's R²GLMM to random slopes models. *Methods in Ecology and Evolution*, 5(9), 944-946.

Examples

```
## Not run:  
data(aces_daily)  
m1 <- lme4::lmer(NegAff ~ STRESS + (1 + STRESS | UserID),  
  data = aces_daily)  
m2 <- lme4::lmer(NegAff ~ STRESS + (1 | UserID),  
  data = aces_daily)  
  
compareLMER(m1, m2)  
  
rm(m1, m2)  
  
## End(Not run)
```

cor2cov	<i>Convert a correlation matrix and standard deviations to a covariance matrix</i>
---------	--

Description

This is a simple function designed to convert a correlation matrix (standardized covariance matrix) back to a covariance matrix. It is the opposite of cov2cor.

Usage

```
cor2cov(V, sigma)
```

Arguments

V	an n x n correlation matrix. Should be numeric, square, and symmetric.
sigma	an n length vector of the standard deviations. The length of the vector must match the number of columns in the correlation matrix.

Value

an n x n covariance matrix

See Also

[cov2cor](#)

Examples

```
# using a built in dataset
cor2cov(cor(longley), sapply(longley, sd))

# should match the above covariance matrix
cov(longley)
all.equal(cov(longley), cor2cov(cor(longley), sapply(longley, sd)))
```

corOK	<i>Return a non-missing correlation matrix</i>
-------	--

Description

Given a square, symmetric matrix (such as a correlation matrix) this function tries to drop the fewest possible number of variables to return a (square, symmetric) matrix with no missing cells.

Usage

```
corOK(x, maxiter = 100)
```

Arguments

x	a square, symmetric matrix or object coercable to such (such as a data frame).
maxiter	a number indicating the maximum number of iterations, currently as a sanity check. See details.

Details

The assumption that x is square and symmetric comes because it is assumed that the number of missing cells for a given column are identical to that of the corresponding row. corOK finds the column with the most missing values, and drops that (and its corresponding row), and continues on in like manner until the matrix has no missing values. Although this was intended for a correlation matrix, it could be used on other types of matrices. Note that because corOK uses an iterative method, it can be slow when many columns/rows need to be removed. For the intended use (correlation matrices) there probably should not be many missing. As a sanity check and to prevent tediously long computations, the maximum number of iterations can be set.

Value

A list with two elements

x	The complete non missing matrix.
keep.indices	A vector of the columns and rows from the original matrix to be kept (i.e., that are nonmissing).

Examples

```
cormat <- cor(iris[, -5])
# set missing
cormat[cbind(c(1,2), c(2,1))] <- NA

# print
cormat

# return complete
corOK(cormat)

# using maximum iterations
corOK(cormat, maxiter=0)

# clean up
rm(cormat)
```

corplot

Heatmap of a Correlation Matrix

Description

This function creates a heatmap of a correlation matrix using **ggplot2**.

Usage

```
corplot(x, coverage, pvalues, plot = c("cor", "p", "coverage"),
        digits = 2, order = c("cluster", "asis"), ...,
        control.grobs = list())
```

Arguments

<code>x</code>	A correlation matrix or some other square symmetric matrix.
<code>coverage</code>	An (optional) matrix with the same dimensions as <code>x</code> giving the proportion of data present. Particularly useful when the correlation matrix is a pairwise present.
<code>pvalues</code>	An (optional) matrix with the same dimensions as <code>x</code> giving the p values for each correlation. To show, use <code>plot = "p"</code> .
<code>plot</code>	A character string indicating what to show on top of the heatmap. Can be 'coverage', in which case bubble points show coverage; 'p', in which case p values are shown, or 'cor', in which case correlations are shown. Only has an effect if a coverage (or pvalue) matrix is passed also. Defaults to <code>cor</code> .
<code>digits</code>	The number of digits to round to when printing the correlations on the heatmap. Text is suppressed when a coverage matrix is passed and <code>points = TRUE</code> .
<code>order</code>	A character string indicating how to order the resulting plot. Defaults to 'cluster' which uses hierarchical clustering to sensibly order the variables. The other option is 'asis' in which case the matrix is plotted in the order it is passed.
<code>...</code>	Additional arguments currently only passed to <code>hclust</code> and <code>corOK</code> .
<code>control.grobs</code>	A list of additional <code>quote()</code> d options to customize the <code>ggplot2</code> output.

Details

The actual plot is created using `ggplot2` and `geom_tile`. In addition to creating the plot, the variables are ordered based on a hierarchical clustering of the correlation matrix. Specifically, `1 - x` is used as the distance matrix. If `coverage` is passed, will also add a bubble plot with the area proportional to the proportion of data present for any given cell. Defaults for `ggplot2` are set, but it is possible to use a named list of `quote()`d `ggplot` calls to override all defaults. This is not expected for typical use. Particularly `main`, `points`, and `text` as these rely on internal variable names; however, `labels`, the gradient color, and area scaling can be adjusted more safely.

Value

Primarily called for the side effect of creating a plot. However, the `ggplot2` plot object is returned, so it can be saved, replotted, edited, etc.

Examples

```
# example plotting the correlation matrix from the
# mtcars dataset
corplot(cor(mtcars))

dat <- as.matrix(iris[, 1:4])
```



```

# randomly set 25% of the data to missing
set.seed(10)
dat[sample(length(dat), length(dat) * .25)] <- NA

# create a summary of the data (including coverage matrix)
sdat <- SEMSummary(~ ., data = dat)
# using the plot method for SEMSummary (which basically just calls corplot)
plot(sdat)

## getting p values instead of coverage
# plot(sdat, plot = "p")

## showing correlations instead of coverage
# plot(sdat, plot = "cor")

# use the control.grobs argument to adjust the coverage scaling
# to go from 0 to 1 rather than the range of coverage
corplot(x = sdat$sSigma, coverage = sdat$coverage,
        control.grobs = list(area = quote(scale_size_area(limits = c(0, 1))))
)

# also works with plot() on a SEMSummary
plot(x = sdat, control.grobs = list(area = quote(scale_size_area(limits = c(0, 1))))

rm(dat, sdat)

```

detailedTests

Detailed Comparisons and Tests on Models

Description

TODO: make me!

Usage

```
detailedTests(obj, ...)
```

Arguments

obj	A fitted model object, currently either a merMod or vglm class object.
...	Additional arguments passed to specific methods.

Value

Depends on the method dispatch, see `.detailedTestsLMER` and `.detailedTestsVGLM`.

Examples

```

mtcars$cyl <- factor(mtcars$cyl)
m <- VGAM::vglm(cyl ~ qsec,
  family = VGAM::multinomial(), data = mtcars)
detailedTests(m)
rm(m, mtcars)

## Not run:
mtcars$cyl <- factor(mtcars$cyl)
mtcars$am <- factor(mtcars$am)
m <- VGAM::vglm(cyl ~ qsec,
  family = VGAM::multinomial(), data = mtcars)
detailedTests(m)

detailedTests(m, digits = 4)$Table
detailedTests(m, OR = FALSE)
detailedTests(m, digits = 4, OR = FALSE)$Table

m <- VGAM::vglm(cyl ~ scale(qsec),
  family = VGAM::multinomial(), data = mtcars)
detailedTests(m)

m2 <- VGAM::vglm(cyl ~ factor(vs) * scale(qsec),
  family = VGAM::multinomial(), data = mtcars)
detailedTests(m2)

m <- VGAM::vglm(Species ~ Sepal.Length,
  family = VGAM::multinomial(), data = iris)
detailedTests(m)

data(aces_daily)
m1 <- lme4::lmer(NegAff ~ STRESS + (1 + STRESS | UserID),
  data = aces_daily)
m2 <- lme4::lmer(NegAff ~ STRESS + I(STRESS^2) + (1 + STRESS | UserID),
  data = aces_daily)
testm1 <- detailedTests(m1, method = "profile")
testm2 <- detailedTests(m2, method = "profile")
testm2b <- detailedTests(m2, method = "boot", nsim = 100)

## End(Not run)

```

egtable

Function makes nice tables

Description

Give a dataset and a list of variables, or just the data in the vars. For best results, convert categorical variables into factors. Provides a table of estimated descriptive statistics optionally by group levels.

Usage

```
egltable(vars, g, idvar, data, strict = TRUE, parametric = TRUE,
         simChisq = FALSE, sims = 1e+06)
```

Arguments

<code>vars</code>	Either an index (numeric or character) of variables to access from the data argument, or the data to be described itself.
<code>g</code>	A variable used to group/separate the data prior to calculating descriptive statistics.
<code>idvar</code>	A character string indicating the variable name of the ID variable. Not currently used, but will eventually support <code>egltable</code> supporting repeated measures data.
<code>data</code>	optional argument of the dataset containing the variables to be described.
<code>strict</code>	Logical, whether to strictly follow the type of each variable, or to assume categorical if the number of unique values is less than or equal to 3.
<code>parametric</code>	Logical whether to use parametric tests in the case of multiple groups to test for differences. Only applies to continuous variables. If TRUE, the default, uses one-way ANOVA, and a F test. If FALSE, uses the Kruskal-Wallis test.
<code>simChisq</code>	Logical whether to estimate p-values for chi-square test for categorical data when there are multiple groups, by simulation. Defaults to FALSE. Useful when there are small cells as will provide a more accurate test in extreme cases, similar to Fisher Exact Test but generalizing to large dimension of tables.
<code>sims</code>	Integer for the number of simulations to be used to estimate p-values for the chi-square tests for categorical variables when there are multiple groups.

Value

A data frame of the table.

Examples

```
egltable(iris)
egltable(colnames(iris)[1:4], "Species", iris)
egltable(iris, parametric = FALSE)
egltable(colnames(iris)[1:4], "Species", iris,
         parametric = FALSE)
egltable(colnames(iris)[1:4], "Species", iris,
         parametric = c(TRUE, TRUE, FALSE, FALSE))
egltable(colnames(iris)[1:4], "Species", iris,
         parametric = c(TRUE, TRUE, FALSE, FALSE), simChisq=TRUE)

diris <- as.data.table(iris)
egltable("Sepal.Length", g = "Species", data = diris)
```

empirical_pvalue	<i>Calculates an empirical p-value based on the data</i>
------------------	--

Description

This function takes a vector of statistics and calculates the empirical p-value, that is, how many fall on the other side of zero. It calculates a two-tailed p-value.

Usage

```
empirical_pvalue(x, na.rm = TRUE)
```

Arguments

x	a data vector to operate on
na.rm	Logical whether to remove NA values. Defaults to TRUE

Value

a named vector with the number of values falling at or below zero, above zero, and the empirical p-value.

Author(s)

Joshua F. Wiley <josh@elkhartgroup.com>

Examples

```
empirical_pvalue(rnorm(100))
```

findSigRegions	<i>Function to find significant regions from an interaction</i>
----------------	---

Description

This function uses the contrast function from **rms** to find the threshold for significance from interactions.

Usage

```
findSigRegions(object, l1, l2, name.vary, lower, upper, alpha = 0.05,  
starts = 50)
```

Arguments

object	A fitted rms object
l1	the first set of values to fix for the contrast function
l2	the second set of values to fix for the contrast function
name.vary	the name of the model parameter to vary values for to find the threshold. Note that this should not be included in l1 or l2 arguments.
lower	The lower bound to search for values for the varying value
upper	The upper bound to search for values for the varying value
alpha	The significance threshold, defaults to .05
starts	Number of starting values to try between the lower and upper bounds.

Value

A data table with notes if no convergence or significance thresholds (if any).

Examples

```
## make me
```

formatHtest	<i>Function to format the results of a hypothesis test as text</i>
-------------	--

Description

Function to format the results of a hypothesis test as text

Usage

```
formatHtest(x, type = c("t", "F", "chisq", "kw", "mh", "r_pearson",
  "r_kendall", "r_spearman"), ...)
```

Arguments

x	A htest class object
type	The type of htest. Currently one of: "t", "F", "chisq", "kw", "mh", "r_pearson", "r_kendall", or "r_spearman" for t-tests, F-tests, chi-square tests, kruskal-wallis tests, Mantel-Haenszel tests, pearson correlations, kendall tau correlation, and spearman rho correlation, respectively.
...	Arguments passed on to p-value formatting

Value

A character string with results

Examples

```
formatHtest(t.test(extra ~ group, data = sleep), type = "t")
formatHtest(anova(aov(mpg ~ factor(cyl), data = mtcars)), type = "F")
formatHtest(chisq.test(c(A = 20, B = 15, C = 25)), type = "chisq")
formatHtest(kruskal.test(Ozone ~ Month, data = airquality))
formatHtest(mantelhaen.test(UCBAdmissions), type = "mh")
formatHtest(cor.test(~ mpg + hp, data = mtcars, method = "pearson"), type = "r_pearson")
formatHtest(cor.test(~ mpg + hp, data = mtcars, method = "kendall"), type = "r_kendall")
formatHtest(cor.test(~ mpg + hp, data = mtcars, method = "spearman"), type = "r_spearman")
```

formatLMER

Format results from a linear mixed model

Description

Format results from a linear mixed model

Usage

```
formatLMER(list, modelnames, format = list(FixedEffects =
  c("%s%s [%s, %s]"), RandomEffects = c("%s", "%s [%s, %s]"),
  EffectSizes = c("%s/%s, %s")), digits = 2, pcontrol = list(digits
  = 3, stars = TRUE, includeP = FALSE, includeSign = FALSE, dropLeadingZero
  = TRUE), ...)
```

Arguments

<code>list</code>	A list of one (or more) models estimated from <code>lmer</code>
<code>modelnames</code>	An (optional) vector of names to use in the column headings for each model.
<code>format</code>	A list giving the formatting style to be used for the fixed effects, random effects, and effect sizes. For the random effects, must be two options, one for when the random effects do not have confidence intervals and one when the random effects do have confidence intervals.
<code>digits</code>	A numeric value indicating the number of digits to print. This is still in early implementation stages and currently does not change all parts of the output (which default to 2 decimals per APA style).
<code>pcontrol</code>	A list controlling how p values are formatted.
<code>...</code>	Additional arguments passed to <code>confint</code> . Notably <code>nsim</code> and <code>boot.type</code> if the bootstrap method is used.

Value

a data table of character data

Examples

```
## Not run:
data(sleepstudy)
m1 <- lme4::lmer(Reaction ~ Days + (1 + Days | Subject),
  data = sleepstudy)
m2 <- lme4::lmer(Reaction ~ Days + I(Days^2) + (1 + Days | Subject),
  data = sleepstudy)

testm1 <- detailedTests(m1, method = "profile")
testm2 <- detailedTests(m2, method = "profile")
formatLMER(list(testm1, testm2))
formatLMER(list(testm1, testm2),
  format = list(
    FixedEffects = "%s, %s (%s, %s)",
    RandomEffects = c("%s", "%s (%s, %s)"),
    EffectSizes = "%s, %s; %s"),
  pcontrol = list(digits = 3, stars = FALSE,
    includeP = TRUE, includeSign = TRUE,
    dropLeadingZero = TRUE))

## End(Not run)
```

formatMedIQR

*Function to format the median and IQR of a variable***Description**

Function to format the median and IQR of a variable

Usage

```
formatMedIQR(x, d = 2, na.rm = TRUE)
```

Arguments

x	the data to have the median and IQR calculated
d	How many digits to display. Defaults to 2.
na.rm	Logical whether to remove missing values. Defaults to TRUE.

Value

A character string with results

Examples

```
formatMedIQR(mtcars$mpg)
```

formatPval	<i>Function to simplify formatting p-values for easy viewing / publication</i>
------------	--

Description

Function to simplify formatting p-values for easy viewing / publication

Usage

```
formatPval(x, d = 3, sd, includeP = FALSE, includeSign = FALSE,
           dropLeadingZero = TRUE)
```

Arguments

x	p values to convert
d	number of digits
sd	number of scientific digits. Defaults to d if missing.
includeP	logical value whether to include the character “p” itself. Defaults to FALSE.
includeSign	logical value whether to include the character “=” or “<”. Defaults to FALSE and if includeP = TRUE it must be TRUE.
dropLeadingZero	logical value whether to drop leading zeros for p-values. Defaults to TRUE.

Value

A character string with stars

Examples

```
formatPval(c(.00052456, .000000124, .01035, .030489, .534946))
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 3, 3, FALSE, TRUE)
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 3, 3, TRUE, TRUE)
formatPval(c(.00052456, .000000124, .01035, .030489, .534946), 5)
formatPval(c(1, .15346, .085463, .05673, .04837, .015353462,
             .0089, .00164, .0006589, .000000053326), 3, 5)
formatPval(c(1, .15346, .085463, .05673, .04837, .015353462,
             .0089, .00164, .0006589, .000000053326), 3, 5, dropLeadingZero = FALSE)
```

gglikert	<i>Creates a plot for likert scale</i>
----------	--

Description

Creates a plot for likert scale

Usage

```
gglikert(x, y, leftLab, rightLab, colour, data, xlim, title, shape = 18,
         size = 7)
```

Arguments

x	Variable to plot on the x axis (the likert scale responses or averages)
y	The variable containing an index of the different items, should be integers
leftLab	The variable with anchors for the low end of the Likert scale
rightLab	The variable with anchors for the high end of the Likert scale
colour	A character string giving the name of a variable for colouring the data, like a grouping variable. Alternately the colour of points passed to geom_point
data	The data to use for plotting
xlim	A vector giving the lower an upper limit for the x axis. This should be the possible range of the Likert scale, not the actual range.
title	A character vector giving the title for the plot
shape	A number indicating the point shape, passed to geom_point
size	A number indicating the size of points, passed to geom_point

Examples

```
testdat <- data.frame(
  Var = 1:4,
  Mean = c(1.5, 3, 2.2, 4.6),
  Low = c("Happy", "Peaceful", "Excited", "Content"),
  High = c("Sad", "Angry", "Hopeless", "Anxious"),
  stringsAsFactors = FALSE)

gglikert("Mean", "Var", "Low", "High", data = testdat, xlim = c(1, 5),
         title = "Example Plot of Average Affect Ratings")

testdat <- rbind(
  cbind(testdat, Group = "Young"),
  cbind(testdat, Group = "Old"))
testdat$Mean[5:8] <- c(1.7, 2.6, 2.0, 4.4)

gglikert("Mean", "Var", "Low", "High", colour = "Group",
```

```

data = testdat, xlim = c(1, 5),
title = "Example Plot of Average Affect Ratings")

gglikert("Mean", "Var", "Low", "High", colour = "Group",
  data = testdat, xlim = c(1, 5),
  title = "Example Plot of Average Affect Ratings") +
scale_colour_manual(values = c("Young" = "grey50", "Old" = "black"))

## clean up
rm(testdat)

```

iccMixed

Intraclass Correlation Coefficient (ICC) from Mixed Models

Description

This function estimates the ICC from mixed effects models estimated using **lme4**.

Usage

```
iccMixed(dv, id, data, family = c("gaussian", "binomial"))
```

Arguments

dv	A character string giving the variable name of the dependent variable.
id	A character vector of length one or more giving the ID variable(s). Can be more than one.
data	A data.table containing the variables used in the formula. This is a required argument. If a data.frame, it will silently coerce to a data.table. If not a data.table or data.frame, it will attempt to coerce, with a message.
family	A character vector giving the family to use for the model. Currently only supports "gaussian" or "binomial".

Value

A data table of the ICCs

References

For details, see Campbell, M. K., Mollison, J., & Grimshaw, J. M. (2001). Cluster trials in implementation research: estimation of intraclass correlation coefficients and sample size. *Statistics in Medicine*, 20(3), 391-399.

Examples

```
iccMixed("mpg", "cyl", mtcars)
iccMixed("mpg", "cyl", as.data.table(mtcars))
iccMixed("mpg", "cyl", as.data.table(mtcars), family = "gaussian")
iccMixed("mpg", c("cyl", "am"), as.data.table(mtcars))
iccMixed("am", "cyl", as.data.table(mtcars), family = "binomial")
```

intSigRegGraph	<i>Function to find significant regions from an interaction</i>
----------------	---

Description

This function uses the contrast function from **rms** to find the threshold for significance from interactions.

Usage

```
intSigRegGraph(object, predList, contrastList, xvar, varyvar,
  varyvar.levels, xlab = xvar, ylab = "Predicted Values", ratio = 1,
  xlim, ylim, xbreaks, xlabels = xbreaks, scale.x = c(m = 0, s = 1),
  scale.y = c(m = 0, s = 1), starts = 50)
```

Arguments

object	A fitted rms object
predList	TODO
contrastList	TODO
xvar	TODO
varyvar	TODO
varyvar.levels	TODO
xlab	optional
ylab	TODO
ratio	TODO
xlim	TODO
ylim	TODO
xbreaks	TODO
xlabels	optional
scale.x	optional
scale.y	optional
starts	Number of starting values to try between the lower and upper bounds.

Value

A data table with notes if no convergence or significance thresholds (if any).

Examples

```
## make me
```

lagk	<i>Create a lagged variable</i>
------	---------------------------------

Description

Given a variable, create a k lagged version, optionally do it by a grouping factor, such as an ID.

Usage

```
lagk(x, k = 1, by)
```

Arguments

x	the variable to lag
k	the length to lag it
by	a variable to lag by. Must be sorted.

Value

a vector of the lagged values

Examples

```
lagk(1:4, 1)
```

logicals	<i>Several logical range comparison helpers</i>
----------	---

Description

Several logical range comparison helpers

Usage

```

e1 %gele% e2
e1 %gel% e2
e1 %gle% e2
e1 %gl% e2
e1 %sgele% e2
e1 %sgel% e2
e1 %sgle% e2
e1 %sgl% e2
e1 %sge% e2
e1 %sg% e2
e1 %sle% e2
e1 %sl% e2
e1 %nin% e2
e1 %sin% e2
e1 %snin% e2
e1 %flipIn% e2

```

Arguments

e1	A number of vector to be evaluated
e2	A vector of one or two numbers used to denote the limits for logical comparison.

Value

A logical vector of the same length as e1 or for those functions prefaced with “s” the subsetted vector.

Examples

```

1:5 %gele% c(2, 4)
1:5 %gele% c(4, 2) # order does not matter uses min / max

```

```

1:5 %gel% c(2, 4)
1:5 %gel% c(4, 2) # order does not matter uses min / max

1:5 %gle% c(2, 4)
1:5 %gle% c(4, 2) # order does not matter uses min / max

1:5 %gl% c(2, 4)
1:5 %gl% c(4, 2) # order does not matter uses min / max

1:5 %sgele% c(2, 4)
1:5 %sgele% c(4, 2) # order does not matter uses min / max

1:5 %sgel% c(2, 4)
1:5 %sgel% c(4, 2) # order does not matter uses min / max

1:5 %sgle% c(2, 4)
1:5 %sgle% c(4, 2) # order does not matter uses min / max

1:5 %sgl% c(2, 4)
1:5 %sgl% c(4, 2) # order does not matter uses min / max

1:5 %sge% 2
1:5 %sge% 4

1:5 %sg% 2
1:5 %sg% 4

1:5 %sle% 2
1:5 %sle% 4

1:5 %sl% 2
1:5 %sl% 4

1:5 %nin% c(2, 99)
c("jack", "jill", "john", "jane") %nin% c("jill", "jane", "bill")

1:5 %sin% c(2, 99)
c("jack", "jill", "john", "jane") %sin% c("jill", "jane", "bill")

1:5 %snin% c(2, 99)
c("jack", "jill", "john", "jane") %snin% c("jill", "jane", "bill")

```

meanDecompose

Mean decomposition of a variable by group(s)

Description

This function decomposes a variable in a long data set by grouping factors, such as by ID.

Usage

```
meanDecompose(formula, data)
```

Arguments

`formula` A formula of the variables to be used in the analysis. Should have the form: `variable ~ groupingfactors`.

`data` A data table or data frame containing the variables used in the formula. This is a required argument.

Value

A list of data tables with the means or residuals

Examples

```
meanDecompose(mpg ~ vs, data = mtcars)
meanDecompose(mpg ~ vs + cyl, data = mtcars)

## Example plotting the results
tmp <- meanDecompose(Sepal.Length ~ Species, data = iris)
do.call(plot_grid, c(lapply(names(tmp), function(x) {
  testdistr(tmp[[x]]$X, plot = FALSE, varlab = x)$Density
}), ncol = 1))

rm(tmp)
```

<code>meanDeviations</code>	<i>Function to calculate the mean and deviations from mean</i>
-----------------------------	--

Description

Tiny helper function to calculate the mean and deviations from the mean, both returned as a list. Works nicely with `data.table` to calculate a between and within variable.

Usage

```
meanDeviations(x, na.rm = TRUE)
```

Arguments

`x` A vector, appropriate for the mean function.

`na.rm` A logical, whether to remove missing or not. Defaults to `TRUE`.

Value

A list of the mean (first element) and deviations from the mean (second element).

Examples

```
## simple example showing what it does
meanDeviations(1:10)

## example use case, applied to a data.table
d <- as.data.table(iris)
d[, c("BSepal.Length", "WSepal.Length") := meanDeviations(Sepal.Length),
     by = Species]
str(d)
```

moments

Estimate the first and second moments

Description

This function relies on the **lavaan** package to use the Expectation Maximization (EM) algorithm to estimate the first and second moments (means and [co]variances) when there is missing data.

Usage

```
moments(data, ...)
```

Arguments

data	A data frame or an object coercible to a data frame. The means and covariances of all variables are estimated.
...	Additional arguments passed on to the <code>estimate.moments.EM</code> function in lavaan . Note this is not an exported function.

Value

A list containing the estimates from the EM algorithm.

mu	A named vector of the means.
sigma	The covariance matrix.

Author(s)

Suggested by Yves Rosseel author of the lavaan package on which this depends

See Also

[SEMSummary](#)

Examples

```

# sample data
Xmiss <- as.matrix(iris[, -5])
# make 25% missing completely at random
set.seed(10)
Xmiss[sample(length(Xmiss), length(Xmiss) * .25)] <- NA
Xmiss <- as.data.frame(Xmiss)

# true means and covariance
colMeans(iris[, -5])
# covariance with n - 1 divisor
cov(iris[, -5])

# means and covariance matrix using list wise deletion
colMeans(na.omit(Xmiss))
cov(na.omit(Xmiss))

# means and covariance matrix using EM
moments(Xmiss)
# clean up
rm(Xmiss)

```

mvqq

NOTE: this function is replaced and combined into the testdistr function.

Description

This is a simple plotting function designed to help examine multivariate normality using the (squared) Mahalanobis distance.

Usage

```

mvqq(dat, use = c("fiml", "pairwise.complete.obs", "complete.obs"),
      plot = TRUE)

```

Arguments

dat	A data frame or matrix of multivariate data to be plotted
use	A character vector indicating how the moments (means and covariance matrix) should be estimated in the presence of missing data. The default is to use full information maximum likelihood based on functions in lavaan .
plot	A logical argument whether to plot the results. Defaults to TRUE.

Value

An invisible list of the density plot, QQ plot, and the data containing quantiles from the chi-squared distribution. Can be useful to find and remove multivariate outliers.

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

```
testdistr(mtcars, "mvnormal")
```

nEffective

Estimate the effective sample size from longitudinal data

Description

This function estimates the (approximate) effective sample size.

Usage

```
nEffective(n, k, icc, dv, id, data, family = c("gaussian", "binomial"))
```

Arguments

n	The number of unique/independent units of observation
k	The (average) number of observations per unit
icc	The estimated ICC. If missing, will estimate (and requires that the family argument be correctly specified).
dv	A character string giving the variable name of the dependent variable.
id	A character vector of length one giving the ID variable.
data	A data.table containing the variables used in the formula. This is a required argument. If a data.frame, it will silently coerce to a data.table. If not a data.table or data.frame, it will attempt to coerce, with a message.
family	A character vector giving the family to use for the model. Currently only supports "gaussian" or "binomial".

Value

A data.table including the effective sample size.

References

For details, see Campbell, M. K., Mollison, J., & Grimshaw, J. M. (2001). Cluster trials in implementation research: estimation of intracluster correlation coefficients and sample size. *Statistics in Medicine*, 20(3), 391-399.

Examples

```
## example where n, k, and icc are estimated from the data
## provided, partly using iccMixed function
nEffective(dv = "mpg", id = "cyl", data = mtcars)

## example where n, k, and icc are known (or being 'set')
## useful for sensitivity analyses
nEffective(n = 60, k = 10, icc = .6)
```

param_summary	<i>Calculates summaries for a parameter</i>
---------------	---

Description

This function takes a vector of statistics and calculates several summaries: mean, median, 95 the empirical p-value, that is, how many fall on the other side of zero.

Usage

```
param_summary(x, trans = function(x) x, ..., na.rm = TRUE)
```

Arguments

x	a data vector to operate on
trans	A function to transform the data. Used for summaries, but not p-values. Defaults to the identity function.
...	Additional arguments passed to formatPval to control p-value printing.
na.rm	Logical whether to remove NA values. Defaults to TRUE

Value

A data frame of summary statistics

Author(s)

Joshua F. Wiley <josh@elkhartgroup.com>

Examples

```
param_summary(rnorm(100))
```

param_summary_format *Format a data frame of summary statistics*

Description

This functions nicely formats a data frame of parameter summary statistics and is designed to be used with the param_summary() function.

Usage

```
param_summary_format(d, digits = getOption("digits"), pretty = FALSE)
```

Arguments

d	A data frame of the parameter summary statistics
digits	Number of digits to round to for printing
pretty	Logical value whether prettified values should be returned. Defaults to FALSE.

Value

A formatted data frame of summary statistics or a formatted vector (if pretty = TRUE).

Author(s)

Joshua F. Wiley <josh@elkhartgroup.com>

Examples

```
set.seed(1234)
xsum <- do.call(rbind, apply(matrix(rnorm(100*10), ncol = 10),
  2, param_summary))
rownames(xsum) <- letters[1:10]
param_summary_format(xsum)
param_summary_format(xsum, pretty = TRUE)

rm(xsum)
```

plot.SEMSummary *Plots SEMSummary object*

Description

Plots SEMSummary object

Usage

```
## S3 method for class 'SEMSummary'  
plot(x, y, ...)
```

Arguments

x	An object of class SEMSummary.
y	Ignored
...	Additional arguments passed on to the real workhorse, corplot.

See Also

[corplot](#), [SEMSummary](#)

Examples

```
# default plot  
plot(SEMSummary(~ ., data = mtcars))  
  
# same as default  
plot(SEMSummary(~ ., data = mtcars), plot = "coverage")  
  
# shows p values  
plot(SEMSummary(~ ., data = mtcars), plot = "p")  
  
# shows correlations  
plot(SEMSummary(~ ., data = mtcars), plot = "cor")
```

plot.SEMSummary.list *Plots SEMSummary.list object*

Description

Plots SEMSummary.list object

Usage

```
## S3 method for class 'SEMSummary.list'  
plot(x, y, which, ...)
```

Arguments

x	An object of class SEMSummary.list.
y	Ignored
which	either a numeric vector based on the positions, or a character vector giving the names of the levels of the list to plot.
...	Additional arguments passed on to the real workhorse, corplot.

See Also

[corplot](#), [SEMSummary](#)

plotDiagnosticsLM *Plot Diagnostics for an lm model*

Description

This function creates a number of diagnostic plots from lm models. It relies heavily on the `testdistr` function.

Usage

```
plotDiagnosticsLM(object, plot = TRUE, ev.perc = 0.001, ask = TRUE,
  ncol = 1)
```

Arguments

object	A fitted model object from lm.
plot	A logical value whether or not to plot the results or simply return the graphical objects.
ev.perc	A real number between 0 and 1 indicating the proportion of the theoretical distribution beyond which values are considered extreme values (possible outliers). Defaults to .001.
ask	A logical whether to ask before changing plots. Only applies to interactive environments.
ncol	The number of columns to use for interactive plots Must be either 1 or 2. Defaults to 1.

Value

a list including plots of the residuals, residuals versus fitted values, and one list for plots of all random effects and finally a data table with any extreme values identified

Examples

```
# make me!
```

plotDiagnosticsLMER *Plot Diagnostics for an lmer model*

Description

This function creates a number of diagnostic plots from lmer models. It relies heavily on the `testdistr` function.

Usage

```
plotDiagnosticsLMER(object, plot = TRUE, ev.perc = 0.001, ask = TRUE,  
  ncol = 1)
```

Arguments

<code>object</code>	A fitted model object, either of class <code>merMod</code> from the <code>lme4</code> package or <code>merModLmerTest</code> from the <code>lmerTest</code> package.
<code>plot</code>	A logical value whether or not to plot the results or simply return the graphical objects.
<code>ev.perc</code>	A real number between 0 and 1 indicating the proportion of the theoretical distribution beyond which values are considered extreme values (possible outliers). Defaults to <code>.001</code> .
<code>ask</code>	A logical whether to ask before changing plots. Only applies to interactive environments.
<code>ncol</code>	The number of columns to use for interactive plots Must be either 1 or 2. Defaults to 1.

Value

a list including plots of the residuals, residuals versus fitted values, and one list for plots of all random effects and finally a data table with any extreme values identified

Examples

```
# make me!
```

R2LMER*Calculates the R2 from lmer models*

Description

For pseudo R2 by cluster, the squared correlation between observed and predicted values for each cluster unit is returned. For the overall model, the marginal and conditional R2 are calculated as described in the references.

Usage

```
R2LMER(model, modelsum, cluster = FALSE)
```

Arguments

<code>model</code>	A model estimated by lmer.
<code>modelsum</code>	The saved model summary (i.e., <code>summary(model)</code>).
<code>cluster</code>	A logical whether to calculate individual pseudo R2 values by cluster unit (if TRUE) or the marginal and conditional R2 for the overall model (if FALSE, the default).

Value

a named vector with the marginal and conditional R2 values, if `CLUSTER = FALSE`, otherwise, a data table with the pseudo R2 for each cluster unit.

References

For estimating the marginal and conditional R-squared values, see: Nakagawa, S. and Schielzeth, H. (2013). A general and simple method for obtaining R2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2), 133-142. as well as: Johnson, P. C. (2014). Extension of Nakagawa & Schielzeth's R2GLMM to random slopes models. *Methods in Ecology and Evolution*, 5(9), 944-946.

Examples

```
## Not run:
data(aces_daily)
m1 <- lme4::lmer(NegAff ~ STRESS + (1 + STRESS | UserID),
  data = aces_daily)

R2LMER(m1, summary(m2))

rm(m1)

## End(Not run)
```

roundedfivenum	<i>Calculate a rounded five number summary</i>
----------------	--

Description

Numbers are the minimum, 25th percentile, median, 75th percentile, and maximum, of the non missing data. Values returned are either the significant digits or rounded values, whichever ends up resulting in the fewest total digits.

Usage

```
roundedfivenum(x, round = 2, sig = 3)
```

Arguments

x	The data to have the summary calculated on
round	The number of digits to try rounding
sig	The number of significant digits to try

Value

The rounded or significant digit five number summary

Examples

```
JWileymisc:::roundedfivenum(rnorm(1000))
JWileymisc:::roundedfivenum(mtcars$hp)
```

SEMSummary	<i>Summary Statistics for a SEM Analysis</i>
------------	--

Description

This function is designed to calculate the descriptive statistics and summaries that are often reported on raw data when the main analyses use structural equation modelling.

Usage

```
SEMSummary(formula, data, use = c("fiml", "pairwise.complete.obs",
  "complete.obs"))
```

Arguments

formula	A formula of the variables to be used in the analysis. See the ‘details’ section for more information.
data	A data frame, matrix, or list containing the variables used in the formula. This is a required argument.
use	A character vector of how to handle missing data. Defaults to “fiml”.

Details

This function calculates a variety of relevant statistics on the raw data used in a SEM analysis. Because it is meant for SEM style data, for now it expects all variables to be numeric. In the future I may try to expand it to handle factor variables somehow.

Both the formula and data arguments are required. The formula should be the right hand side only. The most common way to use it would be with variable names separated by ‘+s’. For convenience, a ‘.’ is expanded to mean “all variables in the data set”. For a large number of variables or when whole datasets are being analyzed, this can be considerably easier to write. Also it facilitates column indexing by simply passing a subset of the data (e.g., `data[, 1:10]`) and using the ‘.’ expansion to analyze the first 10 columns. The examples section demonstrate this use.

Also noteworthy is that `SEMSummary` is not really meant to be used on its own. It is the computational workhorse, but it is meant to be used with a styling or printing method to produce simple output. `APASTyler` has methods for `SEMSummary` output.

There are several new ways to handle missing data now including listwise deletion, pairwise deletion, and using the EM algorithm, the default.

Value

A list with S3 class “SEMSummary”

names	A character vector containing the variable names.
n	An integer vector of the length of each variable used (this includes available and missing data).
nmissing	An integer vector of the number of missing values in each variable.
mu	A vector of the arithmetic means of each variable (on complete data).
stdev	A numeric vector of the standard deviations of each variable (on complete data).
Sigma	The numeric covariance matrix for all variables.
sSigma	The numeric correlation matrix for all variables.
coverage	A numeric matrix giving the percentage (technically decimal) of information available for each pairwise covariance/correlation.
pvalue	The two-sided p values for the correlation matrix. Pairwise present N used to calculate degrees of freedom.

See Also

[APASTyler](#)

Examples

```
## Example using the built in iris dataset
s <- SEMSummary(~ Sepal.Length + Sepal.Width + Petal.Length, data = iris)
s # show output ... not very nice

## Prettier output from SEMSummary
APAStyler(s)

#### Subset the dataset and use the . expansion ####

## summary for all variables in mtcars data set
## with 11 variables, this could be a pain to write out
SEMSummary(~ ., data = mtcars)

## . expansion is also useful when we know column positions
## but not necessarily names
SEMSummary(~ ., data = mtcars[, c(1, 2, 3, 9, 10, 11)])

## clean up
rm(s)

#' # sample data
Xmiss <- as.matrix(iris[, -5])
# make 25% missing completely at random
set.seed(10)
Xmiss[sample(length(Xmiss), length(Xmiss) * .25)] <- NA
Xmiss <- as.data.frame(Xmiss)

SEMSummary(~ ., data = Xmiss, use = "fiml")

## clean up
rm(Xmiss)
```

SEMSummary.fit

Summary Statistics for a SEM Analysis

Description

This is a low level fitting function, for SEMSummary.

Usage

```
SEMSummary.fit(formula, data, use = c("fiml", "pairwise.complete.obs",
  "complete.obs"))
```

Arguments

formula A formula of the variables to be used in the analysis. See the ‘details’ section for more information.

data	A data frame, matrix, or list containing the variables used in the formula. This is a required argument.
use	A character vector of how to handle missing data. Defaults to “fiml”.

Value

A list with S3 class “SEMSummary”

names	A character vector containing the variable names.
n	An integer vector of the length of each variable used (this includes available and missing data).
nmissing	An integer vector of the number of missing values in each variable.
mu	A vector of the arithmetic means of each variable (on complete data).
stdev	A numeric vector of the standard deviations of each variable (on complete data).
Sigma	The numeric covariance matrix for all variables.
sSigma	The numeric correlation matrix for all variables.
coverage	A numeric matrix giving the percentage (technically decimal) of information available for each pairwise covariance/correlation.
pvalue	The two-sided p values for the correlation matrix. Pairwise present N used to calculate degrees of freedom.

See Also

[SEMSummary](#)

star	<i>Function to simplify converting p-values to asterisks</i>
------	--

Description

Function to simplify converting p-values to asterisks

Usage

```
star(x, includeMarginal = FALSE)
```

Arguments

x	p values to convert to stars
includeMarginal	logical value whether to include a symbol for marginally significant >.05 but < .10 p-values. Defaults to FALSE.

Value

A character string with stars

Examples

```
star(c(.0005, .001, .005, .01, .02, .05, .08, .1, .5, 1))
```

testdistr	<i>Graphically compare the distribution of a variable against a specific distribution</i>
-----------	---

Description

This is a simple plotting function designed to help examine distributions. It also includes an option for assessing multivariate normality using the (squared) Mahalanobis distance.

Usage

```
testdistr(x, distr = c("normal", "beta", "chisq", "f", "gamma", "nbinom",
  "poisson", "mvnormal"), na.rm = TRUE, starts, xlim = NULL,
  varlab = "x", plot = TRUE, extremevalues = c("no", "theoretical",
  "empirical"), ev.perc = 0.005, use = c("complete.obs",
  "pairwise.complete.obs", "fiml"), robust = FALSE, rugthreshold = 500,
  seed = 1234, factor = 1, ...)
```

Arguments

x	The data as a single variable or vector to check the distribution unless the distribution is “mvnormal” in which case it should be a data frame or data table.
distr	A character string indicating the distribution to be tested. Currently one of: “normal”, “beta”, “chisq” (chi-squared), “f”, “gamma”, “nbinom” (negative binomial), “poisson”, or “mvnormal” for multivariate normal where Mahalanobis distances are calculated and compared against a Chi-squared distribution with degrees of freedom equal to the number of variables.
na.rm	A logical value whether to omit missing values. Defaults to TRUE.
starts	A named list of the starting values. Not required for all distributions. Passed on to <code>fitdistr</code> which fits the maximum likelihood estimates of the distribution parameters.
xlim	An optional vector to control the x limits for the theoretical distribution density line, useful when densities become extreme at boundary values to help keep the scale of the graph reasonable. Passed on to <code>stat_function</code> .
varlab	A character vector the label to use for the variable
plot	A logical vector whether to plot the graphs. Defaults to TRUE.
extremevalues	A character vector whether to indicate extreme values. Should be “no” to do nothing, “empirical” to show extreme values based on the observed data percentiles, or “theoretical” to show extreme values based on percentiles of the theoretical distribution.

ev.perc	Percentile to use for extreme values. For example if .01, then the lowest 1 percent and highest 1 percent will be labelled extreme values. Defaults to the lowest and highest 0.5 percent.
use	A character vector indicating how the moments (means and covariance matrix) should be estimated in the presence of missing data when <code>distr = mvnormal</code> . The default is to use complete observations, but full information maximum likelihood based on functions in lavaan is also available. See details.
robust	A logical whether to use robust estimation or not. Currently only applies to normally distributed data (univariate or multivariate). Also, when <code>robust = TRUE</code> , only complete observations are used (i.e., <code>use = "complete.obs"</code>). See details.
rugthreshold	Integer determining the number of observations beyond which no rug plot is added. Note that even if this threshold is exceeded, a rug plot will still be added for any extreme values (if extreme values are used and present).
seed	a random seed used to make the jitter added for Poisson and Negative Binomial distributions reproducible
factor	A scale factor fo the amount of jitter added to the QQ and Deviates plots for Poisson and Negative Binomial distributions. Defaults to 1. This results in 1 * smallest distance between points / 5 being used.
...	Additional arguments. If these include <code>mu</code> and <code>sigma</code> and the distribution is multivariate normal, then it will use the passed values instead of calculating the mean and covariances of the data.

Details

Note that for the `use` argument, several options are possible. By default it is “complete.obs”, which uses only cases with complete data on all variables. Another option is “pairwise.complete.obs”, which uses all available data for each variable individually to estimate the means and variances, and all pairwise complete observation pairs for each covariance. Because the same cases are not used for all estimates, it is possible to obtain a covariance matrix that is not positive definite (e.g., correlations $> +1$ or < -1).

Finally, the last option is “fiml”, which uses full information maximum likelihood estimates of the means and covariance matrix. Depending on the number of cases, missing data patterns, and variables, this may be quite slow and computationally demanding.

The `robust` argument determines whether to use robust estimates or not when calculating densities, etc. By default it is `FALSE`, but if `TRUE` and a univariate or multivariate normal distribution is tested, then robust estimates of the means and covariance matrix (a variance if univariate) will be used based on `covMcd` from the **robustbase** package.

Value

An invisible list with the `ggplot2` objects for graphs, as well as information about the distribution (parameter estimates, name, log likelihood (useful for comparing the fit of different distributions to the data), and a dataset with the sorted data and theoretical quantiles.#’

See Also

[SEMSummary](#)

Examples

```
## Not run:

## example data
set.seed(1234)
d <- data.frame(
  Ynorm = rnorm(200),
  Ybeta = rbeta(200, 1, 4),
  Ychisq = rchisq(200, 8),
  Yf = rf(200, 5, 10),
  Ygamma = rgamma(200, 2, 2),
  Ynbinom = rnbinom(200, mu = 4, size = 9),
  Ypois = rpois(200, 4))

## testing and graphing
testdistr(d$Ybeta, "beta", starts = list(shape1 = 1, shape2 = 4))
testdistr(d$Ychisq, "chisq", starts = list(df = 8))
testdistr(d$Yf, "f", starts = list(df1 = 5, df2 = 10))
testdistr(d$Ygamma, "gamma")
testdistr(d$Ynbinom, "poisson")
testdistr(d$Ynbinom, "nbinom")
testdistr(d$Ypois, "poisson")

## compare log likelihood of two different distributions
testdistr(d$Ygamma, "normal")$Distribution$LL
testdistr(d$Ygamma, "gamma")$Distribution$LL

testdistr(d$Ynorm, "normal")
testdistr(c(d$Ynorm, 10, 1000), "normal",
  extremevalues = "theoretical")
testdistr(c(d$Ynorm, 10, 1000), "normal",
  extremevalues = "theoretical", robust = TRUE)

testdistr(mtcars, "mvnormal")

rm(d) ## cleanup

## End(Not run)
```

timeshift

Shift a time variable to have a new center (zero point)

Description

Given a vector, shift the values to have a new center, but keeping the same minimum and maximum. Designed to work with time values where the minimum indicates the same time as the maximum (e.g., 24:00:00 is the same as 00:00:00).

Usage

```
timeshift(x, center = 0, min = 0, max = 1, inverse = FALSE)
```

Arguments

x	the time scores to shift
center	A value (between the minimum and maximum) to center the time scores. Defaults to 0, which has no effect.
min	The theoretical minimum of the time scores. Defaults to 0.
max	the theoretical maximum of the time scores. Defaults to 1.
inverse	A logical value, whether to ‘unshift’ the time scores. Defaults to FALSE.

Value

A vector of shifted time scores, recentered as specified.

Examples

```
## example showing centering at 11am (i.e., 11am becomes new 0)
plot((1:24)/24, timeshift((1:24)/24, 11/24))

## example showing the inverse, note that 24/24 becomes 0
plot((1:24)/24, timeshift(timeshift((1:24)/24, 11/24), 11/24, inverse = TRUE))
```

TukeyHSDgg

Tukey HSD Plot

Description

This calculates and displays means, confidence intervals as well as which groups are different based on Tukey’s HSD. Inspired by <http://stackoverflow.com/questions/18771516/is-there-a-function-to-add-aov-post-hoc-testing-results-to-ggplot2-boxplot>

Usage

```
TukeyHSDgg(x, y, d, ci = 0.95, idvar, ...)
```

Arguments

x	A categorical grouping variable name.
y	A continuous outcome variable name.
d	A dataset
ci	A numeric value indicating the coverage of the confidence interval to use. Defaults to 0.95.
idvar	An optional ID variable for multilevel data
...	Additional arguments passed on.

Value

A ggplot graph object.

Examples

```
## examples using it with single level data
## differences based on an ANOVA and follow up contrasts
TukeyHSDgg("cyl", "mpg", mtcars)

## Not run:
TukeyHSDgg("Species", "Sepal.Length", iris)

## example based on multilevel data
## differences based on model fit with lmer and follow up contrasts
TukeyHSDgg("treatment", "decrease", OrchardSprays, idvar = "colpos")

## End(Not run)
```

updateInstall

Update R and install my core package set

Description

Update R and install my core package set

Usage

```
updateInstall(x, repo)
```

Arguments

x	A character vector of any additional packages to be installed
repo	The repository to be used. Defaults to <code>getOption("repos")</code>

Value

NULL, called for its side effect.

Examples

```
# updateInstall()
```

 VAConverter

Visual Acuity Converter

Description

Converter character (string) input of Snellen fractions, Counting Fingers (CF), and Hand Motion (HM) to logMAR values for use in statistical models. Can handle linear interpolation if passed an appropriate chart or if the measures fit with the default chart.

Usage

```
VAConverter(OS, OD, chart.values = NULL, chart.nletters = NULL,
  datatype = c("snellen", "decimal", "logMAR"), zero = 3)
```

Arguments

OS	The values to be converted for the left eye (oculus sinister).
OD	The values to be converted for the right eye (oculus dexter)
chart.values	The Snellen fractions for the chart used (if interpolation is necessary and it is different from the default).
chart.nletters	The number of letters on each line of the chart that was used. Necessary for proper interpolation.
datatype	The type of data passed to OS and OD. One of "Snellen" (the default), "decimal", or "logMAR". Determines what transformations are needed to convert to logMAR values.
zero	The "zero" logMAR value. This is used as the zero point for visual acuity. For example, for light perception (LP), no light perception (NLP), etc. It defaults to 3 (which is equivalent to a Snellen value of 20/20000), but may also be NA. See details.

Details

VAConverter is primarily designed to take raw character data of various forms and convert them to logMAR values. Acceptable examples include: "20/20", "20/80 + 3", "20/20 - 4", "10/20", "CF 10", "HM 2", "CF 4", "NLP", "LP", "", "CF", "HM", etc. For Snellen values, both parts should be present, and there should be a space between components; e.g., between fraction, +/- and number or between CF and 10. Although I have attempted to make it as flexible and general as possible, there are still fairly rigid requirements so that it can parse a variety of text formats to numerical values. Optionally, it can also handle decimal values (i.e., the results of actually dividing a Snellen value $20/20 = 1$).

chart.values and chart.nletters must be the same length. These are used to interpolate values such as "20/20 + 3" which is interpreted as reading all of the letters on the "20/20" line and "3" of the letters on the next best line (typically "20/15" but this can be chart dependent). The functions goes $3/n$ of the distance between the logMAR values for each line. This is why it is important to know the values for the chart *that was actually used*.

If `datatype = "logMAR"`, the values passed to OS and OD are directly assigned to the `logMAROS` and `logMAROD` slots of a `"VAObject"` and an error is returned if that results in the creation of an invalid object (e.g., they are not numeric or not of equal length).

The `zero` argument is primarily included to facilitate calculating averages. For example, in some cases it may be nice to get a sense of an individual's "overall" or "average" logMAR value. Because on the logMAR scale, 0 is "20/20", an alternate number needs to be used. 3 was chosen as a rough default, but it is by no means necessarily the best choice. If you are not interested in computing an average between the left and right eyes within individuals, it makes sense to simply use NA rather than a crude "zero" approximation.

Value

An object of class `VAObject`. This includes the left and right eye logMAR values in slots `@logMAROS` and `@logMAROD` as well as additional information. More information can be found in the class documentation.

Examples

```
## sampdat <- c("HM 12", "20/20 + 3", "20/50", "CF", "HM",
##           "20/70 - 2", "LP", NA, "Prosthetic")
## tmp <- VAConverter(OS = sampdat, OD = rev(sampdat), datatype = "snellen")
```

VAObject-class	<i>An S4 class to hold visual acuity data</i>
----------------	---

Description

A class to hold Visual Acuity data for the oculus sinister (OS; left eye) and oculus dexter (OD; right eye)

Usage

```
## S4 method for signature 'VAObject,ANY,ANY,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'VAObject'
print(x, ...)

## S4 method for signature 'VAObject'
show(object)

## S4 method for signature 'VAObject'
summary(object, weightbest = TRUE, w = c(0.75,
0.25))
```

Arguments

x	the object to subset
i	the rows to subset (optional)
j	the columns to subset (optional)
...	Additional arguments passed to lower functions
drop	should be missing
object	A VAObject class object
weightbest	Logical whether to upweight the best seeing eye. Defaults to TRUE.
w	A numeric vector of the weights, first for the best seeing then the worst seeing eye. Defaults to <code>c(.75, .25)</code> .

Methods (by generic)

- `[]`: extract method
- `print`: print method
- `show`: show method
- `summary`: summary method

Slots

<code>originalOS</code>	the original visual acuity data for the left (ocular sinister) eye
<code>originalOD</code>	the original visual acuity data for the right (ocular dexter) eye
<code>logMAROS</code>	Logarithm of the minimum angle of resolution data for OS
<code>logMAROD</code>	Logarithm of the minimum angle of resolution data for OD
<code>chart.values</code>	the snellen values for each line of the chart used to measure visual acuity. Used for the linear interpolation in the case of partially correct line readings.
<code>chart.nletters</code>	the number of letters on each line of the chart used to measure visual acuity. Used for the linear interpolation in the case of partially correct line readings (+2 is 2/4 of the way to the next line if there are four letters, but only 2/6 if there are six, etc.)
<code>zero</code>	the logMAR value chosen to represent "zero" visual acuity when creating the combined logMAR values for both eyes or taking the arithmetic mean.

VASummaryObject-class *An S4 class to hold visual acuity summary data*

Description

A class designed to hold visual acuity summary data

Usage

```
## S4 method for signature 'VASummaryObject'
show(object)

## S4 method for signature 'VASummaryObject,missing'
plot(x, y, ...)
```

Arguments

object	The object to be shown
x	A VASummaryObject
y	Should be missing
...	Additional, unused arguments

Methods (by generic)

- show: show method
- plot: plot method

Slots

logMAR.combined Numeric values of the combined logarithm of the minimum angle of resolution data for both eyes

snellen.combined the snellen values back transformed from the combined logMAR values

mean.logMAR average of the logarithm of the minimum angle of resolution data

mean.snellen average of the combined Snellen data

winsorizer	<i>Winsorize at specified percentiles</i>
------------	---

Description

Simple function winsorizes data at the specified percentile.

Usage

```
winsorizer(d, percentile, values, na.rm = TRUE)
```

Arguments

d	A vector, matrix, data frame, or data table to be winsorized
percentile	The percentile bounded by [0, 1] to winsorize data at. If a data frame or matrix is provided for the data, this should have the same length as the number of columns, or it will be repeated for all.
values	If values are specified, use these instead of calculating by percentiles. Should be a data frame with columns named “low”, and “high”. If a data frame or matrix is provided for the data, there should be as many rows for values to winsorize at as there are columns in the data.
na.rm	A logical whether to remove NAs.

Value

winsorized data. Attributes are included to list the exact values (for each variable, if a data frame or matrix) used to winsorize at the lower and upper ends.

Examples

```
dev.new(width = 10, height = 5)
par(mfrow = c(1, 2))
hist(as.vector(eurodist), main = "Eurodist")
hist(winsorizer(as.vector(eurodist), .05), main = "Eurodist with lower and upper\n5% winsorized")

dat <- data.table(x = 1:5)
dat[, y := scale(1:5)]
winsorizer(dat$y, .01)

## make a copy of the data table
winsorizer(dat, .01)

winsorizer(mtcars, .01)

winsorizer(matrix(1:9, 3), .01)

rm(dat) # clean up
```

Index

- *Topic **datasets**
 - aces_daily, 5
- *Topic **hplot**
 - corplot, 15
 - testdistr, 45
- *Topic **misc**
 - formatHtest, 21
 - formatLMER, 22
 - formatMedIQR, 23
 - formatPval, 24
 - star, 44
- *Topic **multivariate**
 - acfByID, 6
 - iccMixed, 26
 - meanDecompose, 30
 - moments, 32
 - mvqq, 33
 - nEffective, 34
 - SEMSummary, 41
 - SEMSummary.fit, 43
 - testdistr, 45
- *Topic **plot**
 - plotDiagnosticsLM, 38
 - plotDiagnosticsLMER, 39
 - TukeyHSDgg, 48
- *Topic **utilities**
 - cd, 11
 - empirical_pvalue, 20
 - param_summary, 35
 - param_summary_format, 36
- *Topic **utils**
 - as.na, 10
 - compareLMER, 13
 - corOK, 14
 - eghtable, 18
 - lagk, 28
 - R2LMER, 40
 - updateInstall, 49
- .detailedTestsLMER, 3
- .detailedTestsVGLM, 4
- [, VAObject, ANY, ANY, ANY-method (VAObject-class), 51
- [, VAObject-method (VAObject-class), 51
- %flipIn% (logicals), 28
- %gel% (logicals), 28
- %gele% (logicals), 28
- %gl% (logicals), 28
- %gle% (logicals), 28
- %nin% (logicals), 28
- %sg% (logicals), 28
- %sge% (logicals), 28
- %sgel% (logicals), 28
- %sgele% (logicals), 28
- %sgl% (logicals), 28
- %sgle% (logicals), 28
- %sin% (logicals), 28
- %sl% (logicals), 28
- %sle% (logicals), 28
- %snin% (logicals), 28
- aces_daily, 5
- acfByID, 6
- APAStyler, 8, 42
- APAStyler.lm, 8
- APAStyler.mira, 9
- APAStyler.SEMSummary, 9
- as.na, 10
- cd, 11
- compareIVs, 12
- compareLMER, 13
- cor2cov, 14
- corOK, 14
- corplot, 15, 37, 38
- cov2cor, 14
- detailedTests, 17
- eghtable, 18

- empirical_pvalue, 20
- findSigRegions, 20
- formatHtest, 21
- formatLMER, 22
- formatMedIQR, 23
- formatPval, 24
- geom_point, 25
- gglikert, 25
- iccMixed, 26
- intSigRegGraph, 27
- lagk, 28
- logicals, 28
- meanDecompose, 30
- meanDeviations, 31
- moments, 32
- mvqq, 33
- nEffective, 34
- param_summary, 35
- param_summary_format, 36
- plot, VASummaryObject, missing-method
(VASummaryObject-class), 52
- plot.SEMSummary, 37
- plot.SEMSummary.list, 37
- plotDiagnosticsLM, 38
- plotDiagnosticsLMER, 39
- print, VAObject-method (VAObject-class),
51
- R2LMER, 40
- roundedfivenum, 41
- SEMSummary, 32, 34, 37, 38, 41, 44, 46
- SEMSummary.fit, 43
- show, VAObject-method (VAObject-class),
51
- show, VASummaryObject-method
(VASummaryObject-class), 52
- star, 44
- summary, VAObject-method
(VAObject-class), 51
- testdistr, 45
- timeshift, 47
- TukeyHSDgg, 48
- updateInstall, 49
- VAConverter, 50
- VAObject, 51
- VAObject-class, 51
- VASummaryObject-class, 52
- winsorizer, 53