# Package ‘QuantPsyc’

February 19, 2015

**Type**  Package

**Title**  Quantitative Psychology Tools

**Version**  1.5

**Date**  2012-03-18

**Depends**  boot, MASS

**Author**  Thomas D. Fletcher

**Maintainer**  Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

**Description**  Contains functions useful for data screening, testing moderation, mediation and estimating power.

**License**  GPL (>= 2)

**Repository**  CRAN

**Date/Publication**  2012-03-20 05:52:48

**NeedsCompilation**  no

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>QuantPsyc-package</td>
<td>2</td>
</tr>
<tr>
<td>ClassLog</td>
<td>4</td>
</tr>
<tr>
<td>distal.med</td>
<td>5</td>
</tr>
<tr>
<td>distInd.ef</td>
<td>7</td>
</tr>
<tr>
<td>eda.uni</td>
<td>8</td>
</tr>
<tr>
<td>graph.mod</td>
<td>9</td>
</tr>
<tr>
<td>lm.beta</td>
<td>10</td>
</tr>
<tr>
<td>Make.Z</td>
<td>11</td>
</tr>
<tr>
<td>meanCenter</td>
<td>12</td>
</tr>
<tr>
<td>moderate.lm</td>
<td>13</td>
</tr>
<tr>
<td>mult.norm</td>
<td>14</td>
</tr>
<tr>
<td>norm</td>
<td>16</td>
</tr>
<tr>
<td>Normalize</td>
<td>17</td>
</tr>
<tr>
<td>plotNorm</td>
<td>18</td>
</tr>
<tr>
<td>powerF</td>
<td>19</td>
</tr>
<tr>
<td>proximal.med</td>
<td>20</td>
</tr>
</tbody>
</table>
Description

This package contains tools useful in screening univariate and multivariate data, testing simple moderating relationships, estimating indirect effects based on simple (proximal) and complex (distal) mediating relationships. A tool for computing power in a given F distribution is also included. These are basic operations covered in a multivariate course in most Doctoral level Psychology programs. These functions will also likely be useful in other domains (e.g., sociology, business management, medicine).

Details

Package: quantpsyc
Type: Package
Version: 1.4
Date: 2010-08-07
License: GPL (version 2 or later)

These functions can be grouped into 4 sets.
Data Screening
Moderation
Mediation
Power Calculation

Data Screening tools include both graphical and statistical methods for assessing the shape of the distributions as well as look for any outliers. Key functions include norm and mult.norm. Moderation functions are based on Aiken & West (1991) and Cohen, Cohen, West & Aiken (2003). Currently, only simple models are permissible (Y ~ X + Z + XZ). Key functions include moderate.lm, sim.slopes, and graph.mod. Mediation functions are largely based on MacKinnon et al (2002) and Fletcher (2006). Both simple (one mediator) and complex (chain of two mediators) relationships can be estimated. Key functions include proximal.med, and distal.med. Finally, powerF will calculate power based on an F distribution given percent variance accounted for (e.g. effect size) and degrees of freedom (e.g., model parameters and sample size). Some of the functions are original and others borrowed from numerous sources. I have taken care to reference appropriately.
Author(s)

Thomas D. Fletcher
Strategic Resources
State Farm Insurance Cos.

Maintainer: Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


Examples

# Data Screening

data(USJudgeRatings)
norm(USJudgeRatings$CONT)
mult.norm(USJudgeRatings[,1:4])

# Moderation

data(tra)
lm.mod1 <- moderate.lm(beliefs, values, attitudes, tra)
ss.mod1 <- sim.slopes(lm.mod1,meanCenter(tra$values))
summary(lm.mod1)
ss.mod1

# use mouse click to place legend in graph.mod
# graph.mod(ss.mod1,beliefs,attitudes,tra,"Interaction Example")

# Mediation

# create object with names x, m, y
# data(tra)
temp.tra <- tra
names(temp.tra) <- c("x", "z", "m", "y")
proximal.med(temp.tra)

ClassLog

Classification for Logistic Regression

Description

Provides a Classification analysis for a logistic regression model. Also provides McFadden’s Rsq.

Usage

ClassLog(MOD, resp, cut=.5)

Arguments

MOD        Model of class glm where family is bonomial
resp       response variable from data
cut        Arbitrary cut for the proportion deemed '1' in model

Value

A list containing:

- rawtab two-way table of classifications as frequencies
- classtab two-way table of classifications as percentages
- overall Overall percent classifications that are correct
- mcFadden McFaddens pseudoRsq: 1 - ModelDeviance / NullDeviance

Warning

This is a primative function. I have a long to do list. For example, it is not yet written to handle missing observations.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

See Also

glm
distal.med

Examples

# create some data
x <- rnorm(100)
y <- as.numeric(cut(.5*x + rnorm(100), breaks=2))-1
tdf <- data.frame(x=x, y=y)

# run a logistic regression
glm1 <- glm(y ~ x, data=tdf, family=binomial)

# Get typical summary of results
summary(glm1)

# Classification Analysis
ClassLog(glm1, tdf$y)

distal.med Distal Indirect Effect

Description

Computes the indirect effect (and all paths) in a 4 variable system, assuming all paths estimated.

Usage

distal.med(data)

Arguments

data data.frame containing the variables labeled 'x', 'm1', 'm2', and 'y' respectively.

Details

Computes the paths in the model system: /cr Y = t'X + fM1 + cM2
M2 = eX + bM1
M1 = aX
and the indirect effect a*b*c + a*f + e*c

Value

Returns a table with all the effects and decomposition of effects in the above 4 variable system including the standard errors and t-values.

a Effect of X on M1
b Effect of M1 on M2 controlling for X
c Effect of M2 on Y controlling for X and M1
e Effect of X on M2 controlling for M1
f  Effect of M1 on Y controlling for X and M2
abc 'Direct' Indirect Effect of X on Y
af  Indirect Effect of X on Y through M1 only
ef  Indirect Effect of M1 on Y though M2
ind.xy 'Total' Indirect effect of X on Y
t  Total Effect of X on Y
t'  Direct Effect of X on Y accounting for all mediators

Warning
This function is primitive in that it is based on a simplistic model AND forces the user to name the variables in the dataset x, m1, m2, and y.

Note
This function uses the following undocumented functions: se.indirect3

Author(s)
Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also
proximal.med, distInd.ef

Examples
cormat <- matrix (c(1,3,.15,.075,.3,1,.3,.15,.3,.15,.3,.15,.3,.15,.3,1,.15,.3,1,.15,.3,1,.15,.3,1,.15,.3,1,.15,.3),ncol=4)
require(MASS)
d200 <- data.frame(mvrnorm(200, mu=c(0,0,0,0), cormat))
names(d200) <- c("x","m1","m2","y")
distal.med(d200)
Computes the 'total indirect effect' from distal.med for use in boot

Usage

```r
distInd.ef(data, i)
```

Arguments

data: data.frame used in distal.med

i: i is a 'count' placeholder necessary for use in boot

Details

This function is not useful of itself. It is specifically created as an intermediate step in bootstrapping the indirect effect.

Value

indirect effect that is passed to boot for each bootstrap sample

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also

distal.med
Examples

```r
cormat <- matrix(c(1,3,15,.075,.3,1,.3,15,.3,1,.3,15,.3,1,.3,15,.3,1),ncol=4)
require(MASS)
d200 <- data.frame(mvrnorm(200, mu=c(0,0,0,0), cormat))
names(d200) <- c("x","m1","m2","y")

require(boot)
distmed.boot <- boot(d200, distIndef, R=999)
sort(distmed.boot$t)[c(25,975)] # 95% CI
plot(density(distmed.boot$t)) # Distribution of bootstrapped indirect effect
summary(distmed.boot$t)
```

eda.uni

**Plots for Exploratory Data Analysis**

Description

This function is a modified version of eda.shape found in the S+ Guide to Statistics, v1, p. 124. It is based on work by Tukey (1977) and each plot is described in more detail in Ch. 4 of Cohen et al. Creates 4 plots useful in assessing univariate distributions of data.

Usage

```r
eda.uni(x, title = "")
```

Arguments

- `x` A univariate data object such as column of variable from a data.frame()
- `title` Title printed above first plot in upper left corner

Details

Simply provides a histogram, smoothed histogram, qq-plot, and boxplot for x.

Value

A single graph object with 4 basic plots.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>
References


See Also

hist, plot.density, qnorm, boxplot

Examples

# create negatively skewed dat with 100 observations
xc <- rchisq(100,3)
eda.uni(xc)

data:

graph.mod

Moderation Graph

Description

x-y plot containing the simple slopes conditioned on z.

Usage

graph.mod(ssmod, x, y, data, title = "", xlab = "Centered X", ylab = "Raw Y", ylimit = 1.5, ...)

Arguments

ssmod Results of sim.slopes; Simple Slopes to be graphed
x Explanatory variable to be used in xypplot
y Outcome variable to be used in xypplot
data data.frame containing x, z, y
title Optional 'main' title for the plot
xlab x-axis label
ylab y-axis label
ylimit used as a multiple of SDy to define the limits of the y-axis...
... to be determined ...
Details

Given `moderate.lm` and `sim.slopes`, this function plots x,y and adds the simple slopes corresponding to arbitrary values of z defined in `sim.slopes`. Users must 'click' on an area to add the legend.

Value

A plot object.

Warning

This is based on a simple 3 variable moderation model with continuous variables. Users must modify the functions to accommodate other models (e.g., categorical moderators, covariates)

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also

`moderate.lm`, `sim.slopes`

Examples

data(tra)
ml.mod <- moderate.lm(beliefs, values, attitudes, tra)
ss.mod <- sim.slopes(ml.mod, tra$values)
## requires user interaction
## graph.mod(ss.mod, beliefs, attitudes, tra,"Interaction Example")

---

**lm.beta**

*Standardized Regression Coefficients*

Description

Computes the standardized regression coefficients (beta) from an object of class (lm)
Usage

lm.beta(MOD)

Arguments

MOD MOD is object from lm with the form y ~ x1 + x2 + ...

Value

A "numeric" representing each standardized coefficient from lm() model

Warning

This function does not produce 'correct' standardized coefficients when interaction terms are present

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

See Also

Make.Z.lm

Examples

us <- USJudgeRatings
names(us)
lm1 <- lm ( CONT ~ INTG + DMNR + DILG, us)
lm.beta(lm1)

# Standardized data (using Make.Z())

usz <- data.frame (Make.Z (us))
lm1.z <- lm ( CONT ~ INTG + DMNR + DILG, usz)

# compare standardized data versus lm.beta

summary(lm1.z)

Description

Converts data to standard normal (mean = 0; SD = 1) - i.e., z-scores.
meanCenter

Usage

Make.Z(x)

Arguments

x Any data object (especially useful for multiple columns of a data.frame).

Details

Takes the data (by columns if necessary) and subtracts out the mean and then divides by the standard deviation. The result is a standard normal z score.

Value

A numeric or matrix containing standardized data (i.e., z scores)

Warning

The result is a matrix. One may wish to convert to data.frame or use as.data.frame(Make.Z(x))

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

Examples

zUSJR <- Make.Z(USJudgeRatings) # creates new object containing z scores
dim(zUSJR) # shows that there are 43 observed z scores for 12 variables
zUSJR[,1] # to look at only the first column of z scores

meanCenter

Mean Center Variables

Description

This simple function subtracts the mean from a variable rendering mean-centered variables.

Usage

meanCenter(x)

Arguments

x variable or column of data to be centered
moderate.lm

Details
This is particularly useful in lm() with higher order terms as in moderation.

Value
a numeric; mean centered 'x'

Author(s)
Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

Examples
data(USJudgeRatings)
usCONT <- meanCenter(USJudgeRatings$CONT)
summary(usCONT)

moderate.lm

Simple Moderated Regression Model

Description
This function creates an object of class lm() specific to a moderated multiple regression involving 3 variables.

Usage
moderate.lm(x, z, y, data, mc = FALSE)

Arguments
x   focal explanatory variable
z   moderating variable
y   outcome variable
data data.frame containing the variables
mc   Logical specifying whether the data are already mean centered

Details
This model takes x and z and creates the interaction term x*z. If the data are not already mean centered, then x and z are mean centered by subtracting out the means. This is necessary for interpretation and to reduce multicolinearity. The lm() is then computed thusly: Y ~ X + Z + XZ.

Value
An object of class lm(). One can use summary(), coef() or any other function useful to lm(). This model is used by other moderator tools - see below.
mult.norm

Warning

This is a very simplistic model. If x or z are categorical, the results will not be accurate. The function can be modified by the user to deal with complications such as covariates, non-continuous variables, etc.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also

sim.slopes, graph.mod

Examples

data(trait)
mod1 <- moderate.lm(beliefs, values, attitudes, trait)
summary(lm.mod1)

```
```

Description

Returns tests for multivariate Skewness and kurtosis as well as Mahalanobis’ D-squared.

Usage

`mult.norm(x, s = var(x), chicrit = 0.005)`

Arguments

- `x`: A multivariate data object as in columns from a data.frame
- `s`: Covariance matrix of `x` (not necessary to specify)
- `chicrit`: p-value corresponding to critical value of chi-square distribution for detecting multivariate outliers
mult.norm

Details

Tests for multivariate skewness and kurtosis were adapted from SAS macros in Khatree & Naik (1999). They attribute the formula to Mardia (1970; 1974). Mahalanobis' Dsq is based on Mahalanobis (1936). Dsq is multivariate analogue to z scores, but based on the chi-sq distribution rather than normal distribution. Once can specify at what level one wishes to define multivariate outliers (e.g., .005, .001)

Value

A list containing the following:

- mult.test: Values for multivariate skewness and kurtosis and their significance
- Dsq: Mahalanobis' distances
- CriticalDsq: Critical value of chi-sq distribution based on df and specified critical level

Note

Mahalanobis is returned without regard to NAs (missing observations) and is useful only in detecting IF multivariate outliers are present. If one wishes to determine which cases are multivariate outliers and if one has missing observations, mahalanobis is perhaps a better choice.

These statistics are known to be susceptible to sample size (as in their univariate counterparts). One should always use graphical methods such as qqplot in addition to statistical.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also

mahalanobis, qqplot

Examples

```r
# assess the multivariate normality of variables 4,5,6 in USJudgeRatings
data(USJudgeRatings)
mn <- mult.norm(USJudgeRatings[,4:6], chicrit=.001)
mn

mn$Dsq > mn$CriticalDsq
```
**Description**

Computes Skewness and Kurtosis of data.

**Usage**

```r
norm(x)
Skew(x)
Kurt(x)
```

**Arguments**

- `x` A data object such as a numeric or column from a data.frame

**Value**

`norm` returns a table containing `Skew` & `Kurt`. Each contain the following elements:

- **Statistic** value for Skewness or Kurtosis respectively
- **SE** Standard error for Skewness or Kurtosis
- **t-val** t or z ratio - Statistic/SE
- **p** p value associated with z distribution

**Warning**

These statistics should be used with caution as they are influenced by sample size!

**Author(s)**

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

**Examples**

```r
# create negatively skewed dat with 100 observations
xc <- rchisq(100,3)
norm(xc)
```
Normalize

Normalize Data

Description

Convert data to Normal Scores with the same Mean and SD. This reshapes data to conform to a Normal Distribution. It is not converting to z-scores (i.e., it is not standardizing data)

Usage

Normalize(x)

Arguments

x

Data to be normalized. Should be vector of scores

Value

A numeric with the same Mean and SD as x, but without skew or kurtosis

Warning

This is a primitive function. I have a long to do list. For example, it is not yet written to handle missing observations.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


Snippets of code were borrowed and modified from: http://zoonek2.free.fr/UNIX/48_R/03.html

See Also

norm

Examples

summary(USJudgeRatings$CONT)
plot(density(USJudgeRatings$CONT))

ContN <- Normalize(USJudgeRatings$CONT)
summary(ContN)
lines(density(ContN), col=2)
plotNorm  Normal Density Plot

Description

Plots the density distribution of 'x' (e.g., smoothed histogram) with an overlaying normal density plot with the same mean and SD. This is useful for 'seeing' the degree of deviance from normality.

Usage

plotNormX(x)
plotNorm Xm(x, im)

Arguments

x       any data object such as a column(s) or variable(s) from a data.frame
im      number of items in x-multivariate to be plotted

Details

plotNormX is useful for single use (univariate) objects, but plotNorm Xm is more useful for creating multiple graphs (i.e., multivariate) as in sending graphs to a postscript or pdf device. See examples below.

Value

A graph of density of x.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

See Also

eda.uni

Examples

# plot.normX
data(USJudgeRatings)  # data packaged with R
plotNormX(USJudgeRatings$CONT)

# creates a pdf file that contains plots for all 12 variables in USJudgeRatings

# plot.norm Xm
data(USJudgeRatings)
pdf("Judge.pdf")  # writes file to working directory
plotNorm Xm(USJudgeRatings, 12)
dev.off()
powerF

Description

Computes power (1 - beta) to detect an effect with a given effect size, sample size (df) and specified alpha (significance) level.

Usage

```r
powerF(PV, df2, df1 = 1, alpha = 0.05)
```

Arguments

- `PV` Percent of variance accounted for by effect.
- `df2` Denominator Degrees of Freedom for a given model
- `df1` Numerator Degrees of Freedom for a given model
- `alpha` Significance level for desired effect

Details

Murphy & Myors (2004) detail the use of a similar function and the notion that most distributions can be converted to F. Therefore, they argue that the F distribution is the most versatile in computing power. Typically, alpha is set at .05 (default). Users will likely find conversions of various distributions to F corresponding to a df1=1 (default). Therefore, users can manipulate df2 based on their model to estimate sample size needs. Likewise, one may begin with a given sample size (i.e., df2) and manipulate PV (effect size) to iteratively determine what power their study is likely to detect. Conventions maintain that .80 is a sufficient target, and that no study should be designed with power = .5 or less.

Value

A numeric value representing the power to detect the effect

Warning

It is critical that the user correctly specify the model for which the effect is obtained. For instance, if a single coefficient from a regression model is the object of inquiry (e.g., interaction effect in moderation model), the DF should reflect that effect and not the overall model, which also contains the 'main effects'.

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>
References


Examples

```r
# Simulated TRA example
data(tra)
lm1 <- lm(attitudes ~ beliefs*values, tra)
summary(lm1)

# power to detect the interaction effect, where df1 = 1 and df2 = n-k-1 = 996
# PV = t^2/(t^2+df2) = .1863
power(.1863, 996)

# Estimate sample size needed to detect interaction effect with PV = .01 and power = .8
power(.01, 200)  # too low
power(.01, 1000)  # too high
power(.01, (800-3-1))  # just right: n=800 - k=3 - 1
```

proximal.med

**Simple Mediation Models**

Description

Computes the Indirect Effect for a simple 3 variable mediation model: X -> M -> Y assuming direct effect X -> Y

Usage

`proximal.med(data)`

Arguments

data

data.frame containing the variables labeled 'x', 'm', and 'y' respectively.

Details

This function computes all paths in the simple 3 variable system involving the following regressions:

- Y = t'X + bM, and
- M = aX

where t' + ab = t

The indirect effect is computed as the product of a*b. Several formula are used for the computation of the standard error for the indirect effect (see MacKinnon et al for a comprehensive review). As noted below, one can use this function to create the indirect effect and then utilize bootstrapping for a more accurate estimate of the standard error and model the distribution of the direct effect.
Value

Creates a table containing the following effects, their standard errors, and t-values:

a  Effect of X on M
b  Effect of M on Y controlling for X
t  Total effect of X on Y
t'  Direct effect of X on Y accounting for M
ab  Indirect effect of X on Y though M
Aroian  Standard error of ab using Aroian method
Goodman  Standard error of ab using Goodman method
Med. Ratio  Mediation Ratio: indirect effect / total effect

Warning

This function is primitive in that it is based on a simplistic model AND forces the user to name the variables in the dataset x, m, and y.

Note

This function uses the following undocumented functions: aroian.se.indirect2, goodman.se.indirect2, se.indirect2

Author(s)

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

References


See Also

distal.med, proxInd.ef

Examples

data(tra)
tmp.tra <- tra
names(tmp.tra) <- c('x','y','m','y')
data.frame(proximal.med(tmp.tra))  ## data.frame() simple makes the table 'pretty'
proxInd.ef  

---

**proxInd.ef**    **Simple Mediation for use in Bootstrapping**

**Description**

Calculates the indirect effect from `proximal.med` in a form useful to send to `boot`.

**Usage**

`proxInd.ef(data, i)`

**Arguments**

- `data`: data.frame used in `proximal.med`
- `i`: is a 'count' placeholder necessary for use in `boot`

**Details**

This function is not useful of itself. It is specifically created as an intermediate step in bootstrapping the indirect effect.

**Value**

Indirect effect that is passed to `boot` for each bootstrap sample.

**Author(s)**

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>

**References**


**See Also**

`proximal.med`
### sim.slopes

**Moderated Simple Slopes**

**Description**

Computes the simple slopes for a moderated regression model.

**Usage**

```r
sim.slopes(mod, z, zsd = 1, mcz = FALSE)
```

**Arguments**

- `mod`: linear model - usually constructed with `moderate.lm`
- `z`: moderating variable
- `zsd`: Multiple for SD of `z`; number of SDs from mean to construct simple slopes
- `mcz`: logical whether `z` is already centered or not in the original data

**Details**

Constructs the simple slopes for arbitrary values of `z` (e.g., +/- 1, 2, 3 standard deviations) involved in a moderated multiple regression equation.

**Value**

A table with the following values for `zHigh` (`Meanz + zsd*SDz`), `Mean(Meanz)`, and `zLow` (`Meanz - zsd*SDz`): 

- **INT**: Intercept of simple slope
- **Slope**: Slope of the simple slope
- **SE**: Standard Error of the slope
- **LCL**, **UCL**: Lower and Upper confidence limits for slope

**Author(s)**

Thomas D. Fletcher <tom.fletcher.mp7e@statefarm.com>
References


See Also

`moderate.lm`, `graph.mod`

Examples

```r
data(tra)
lm.mod1 <- moderate.lm(beliefs, values, attitudes, tra)
ss.mod1 <- sim.slopes(lm.mod1, tra$values)
ss.mod1
```

Simulated Theory of Reasoned Action Data

Description

Simulated data loosely based on the Theory of Reasoned Action from Social Psychology. Similar data are frequently used in publications involving moderation and mediation. These data were created for illustrative purposes.

Usage

`data(tra)`

Format

A data frame with 1000 observations on the following 4 variables.

- `beliefs` a numeric vector
- `values` a numeric vector
- `attitudes` a numeric vector
- `intentions` a numeric vector

Details

The data were constructed with the following model(s) in mind. Attitudes are a function of beliefs as moderated by values. Beliefs lead to intentions though their association with attitudes.
Examples

data(tra)
str(tra)
eda.uni(tra$intentions)
Index

*Topic datasets
  tra, 24
*Topic distribution
  eda.uni, 8
  mult.norm, 14
  Normalize, 17
  plotNorm, 18
  powerF, 19
*Topic models
  ClassLog, 4
  distal.med, 5
  distInd.ef, 7
  graph.mod, 9
  lm.beta, 10
  moderate.lm, 13
  proximal.med, 20
  proxInd.ef, 22
  sim.slopes, 23
*Topic multivariate
  Make.Z, 11
  mult.norm, 14
*Topic package
  QuantPsyc-package, 2
*Topic smooth
  plotNorm, 18
*Topic univar
  eda.uni, 8
  Make.Z, 11
  meanCenter, 12
  norm, 16
  Normalize, 17
  aroian.se.indirect2(proximal.med), 20
  boxplot, 9
  ClassLog, 4
  distal.med, 5, 7, 21
  distInd.ef, 6, 7
  eda.uni, 8, 18
  glm, 4
  goodman.se.indirect2(proximal.med), 20
  graph.mod, 9, 14, 24
  hist, 9
  Kurt(norm), 16
  lm, 11
  lm.beta, 10
  mahalanobis, 15
  Make.Z, 11, 11
  meanCenter, 12
  moderate.lm, 10, 13, 24
  mult.norm, 14
  norm, 16, 17
  Normalize, 17
  plot.density, 9
  plotNorm, 18
  plotNormX(plotNorm), 18
  plotNormXm(plotNorm), 18
  powerF, 19
  proximal.med, 6, 20, 22
  proxInd.ef, 21, 22
  qnorm, 9
  qplot, 15
  QuantPsyc (QuantPsyc-package), 2
  QuantPsyc-package, 2
  se.indirect2(proximal.med), 20
  se.indirect3(distal.med), 5
  sim.slopes, 10, 14, 23
  Skew(norm), 16
  tra, 24