Package 'processR'

January 23, 2023

Type Package

Title Implementation of the 'PROCESS' Macro

Version 0.2.7

URL https://github.com/cardiomoon/processR

BugReports https://github.com/cardiomoon/processR/issues

Description Perform moderation, mediation, moderated mediation and moderated moderation. Inspired from famous 'PROCESS' macro for 'SPSS' and 'SAS' created by Andrew Hayes.

Depends R (>= 2.10)

Date 2023-1-20

License GPL-2

Encoding UTF-8

LazyData true

Imports lavaan, diagram, dplyr, flextable(>= 0.5.8), ggplot2(>= 3.1.1), ggrepel, officer, psych, purrr, rrtable, semTools, stringr, tidyselect, rlang, tidyr, predict3d(>= 0.1.3.3), interactions, ztable, rmarkdown

Suggests shiny, shinyWidgets, knitr, readr

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Keon-Woong Moon [aut, cre], Sokyoung Hong [ctb]

Maintainer Keon-Woong Moon <cardiomoon@gmail.com>

Repository CRAN

Date/Publication 2023-01-23 08:30:06 UTC

R topics documented:

`addArrows` *Add covariates to arrows*

Description

Add covariates to arrows

Usage

```
addArrows(arrows, covar)
```

Arguments

`arrows`	A data.frame
`covar`	A list of covariates

addCatVars *Add dummy vars to data.frame*

Description

Add dummy vars to data.frame

Usage

```
addCatVars(df, varnames, groupLetter = "D", mode = 1)
```

Arguments

df	A data.frame
varnames	Variable name to be converted as factor and add dummies
groupLetter	A character
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding

Examples

```
mtcars1=addCatVars(mtcars,c("cyl","carb"))
mtcars1[c(3:5),]
mtcars2=addCatVars(mtcars,c("cyl","carb"),mode=3)
mtcars2[c(3:5),]
protest1=addCatVars(protest,"protest")
head(protest1)
iris1=addCatVars(iris,c("Species"),mode=3)
(iris1[c(1,51,101),])
```

addCovarEquation *Add covariates to equation*

Description

Add covariates to equation

Usage

```
addCovarEquation(
  equation,
  covar = list(),
  prefix = "f",
  grouplabels = NULL,
  multipleMediator = FALSE
)
```

Arguments

equation	The equation
covar	A list
prefix	prefix
grouplabels	A list
multipleMediator	logical

Examples

```
equation="M ~ X*W\nY ~ a1*M + C1*X"
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M","Y"),"Y","Y"))
grouplabels=list(C1="e")
addCovarEquation(equation,covar=covar)
equation="M1 ~ a11*X\nM2 ~ a12*M"
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M1","Y"),"M2","M2"))
addCovarEquation(equation,covar=covar,multipleMediator=TRUE)
addCovarEquation(equation,covar=covar)
```

---

addLabels *add name to labels*

Description

add name to labels

Usage

```
addLabels(labels, id, name)
```

Arguments

labels	A list
id	label id
name	A character

Examples

```
labels=c(X="X",M="M",Y="Y")
addLabels(labels,"W","X")
addLabels(labels,"W","W")
```

addLatentNodes *Add latent nodes information to nodes*

Description

Add latent nodes information to nodes

Usage

```
addLatentNodes(nodes, fit, labels)
```

Arguments

nodes	A data.frame
fit	An object of class lavaan. Result of lavaan::sem()
labels	A list

addLine *Add line feed to string*

Description

Add line feed to string

Usage

```
addLine(x, ...)
```

Arguments

x	A string
...	one or more R objects, to be converted to character vectors.

addNodes *Add covariates to nodes*

Description

Add covariates to nodes

Usage

```
addNodes(nodes, covar, radx = 0.1, rady = 0.04, no = NULL)
```

Arguments

nodes	A data.frame
covar	A list of covariates
radx	horizontal radius of the box.
rady	vertical radius of the box.
no	A numeric

addPlus *Add '+' mark to string*

Description

Add '+' mark to string

Usage

```
addPlus(x, ...)
```

Arguments

x	A string
...	one or more R objects, to be converted to character vectors.

addTripleInteraction *Add triple interaction*

Description

Add triple interaction

Usage

```
addTripleInteraction(res, names, interactionNo = 0, mode = 1)
```

Arguments

res	A character vector
names	A character vector
interactionNo	A numeric
mode	a numeric

adjustNodes *Adjust y position of nodes*

Description

Adjust y position of nodes

Usage

```
adjustNodes(nodes)
```

Arguments

nodes	A data.frame

adjustPosNodes *Adjust position of nodes*

Description

Adjust position of nodes

Usage

```
adjustPosNodes(nodes)
```

Arguments

nodes	A data.frame

adjustxpos *Adjust x position*

Description

Adjust x position

Usage

```
adjustxpos(xpos, xmargin = 0.01, radx = 0.12, xspace = NULL, mode = 1)
```

Arguments

xpos	x position
xmargin	horizontal margin of plot
radx	horizontal radius of the box
xspace	numeric. horizontal interval
mode	integer adjust mode

adjustypos *Adjust y position*

Description

Adjust y position

Usage

```
adjustypos(
  ypos,
  ymargin = 0.02,
  rady = 0.06,
  maxypos = 0.6,
  minypos = 0,
  totalOnly = FALSE
)
```

Arguments

ypos	y position
ymargin	vertical margin of plot
rady	vertical radius of the box
maxypos	maximal y position of X or W variables
minypos	minimal y position of X or W variables
totalOnly	logical if TRUE, arrange ypos with center 0.5

Examples

```
ypos=c(0.5,0.9,1,1,2,3)
adjustypos(ypos)
adjustypos(ypos,totalOnly=TRUE)
```

---

appendLabels *Append labels from vars, moderator and covar*

Description

Append labels from vars, moderator and covar

Usage

```
appendLabels(labels, vars = list(), moderator = list(), covar = NULL)
```

Arguments

labels	A list
vars	A list
moderator	A list
covar	A list

---

bda.mediation.test *The Sobel mediation test*

Description

To compute statistics and p-values for the Sobel test. Results for three versions of "Sobel test" are provided: Sobel test, Aroian test and Goodman test.

Usage

```
bda.mediation.test(mv, iv, dv)
```

Arguments

mv	The mediator variable
iv	The independent variable
dv	The dependent variable

caskets *CASKETS dataset*

Description

CASKETS dataset

Usage

```
caskets
```

Format

A data.frame with 541 obs. of 7 variables

policy Given information about policy (0 = No information, 1 = Told About Policy)

interest Interest in viewing casket images

age Participant age

educ Participant education level, 1 = less than high school, 2 = high school, 3 = some college, 4 = associates or technical school, 5 = bachelor degree, 6 = some graduate school, 7 = graduate degree

male Participant sex (0 = female, 1 = male)

conserv Participant social conservatism

kerry Kerry or Bush supporter, 0 = bush supporter, 1 = kerry supporter

Source

Hayes, A. F., & Reineke, J. B. (2007). The effects of government censorship of war-related news coverage on interest in the censored coverage: A test of competing theories. Mass Communication and Society, 10, 423-438

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

catMediation *Make Mediation Equation with one categorical variable*

Description

Make Mediation Equation with one categorical variable

Usage

```
catMediation(
  X = NULL,
  M = NULL,
  Y = NULL,
  labels = list(),
  data,
  moderator = list(),
  covar = NULL,
  mode = 0,
  maxylev = 2,
  range = TRUE,
  rangemode = 1
)
```

Arguments

X	Name of independent variable
M	Name of mediator variable
Y	Name of dependent variable
labels	optional list
data	A data.frame
moderator	A list
covar	A list of covariates
mode	A numeric. 0: SEM equation, 1: regression equation
maxylev	maximal unique length of categorical variable
range	A logical
rangemode	range mode

Examples

```
labels=list(X="cyl",M="am",Y="mpg")
moderator=list(name=c("cyl","wt"),site=list(c("a","c"),c("c")))
covar=list(name=c("carb","disp"),label=c("carb","disp"),site=list(c("M","Y"),"Y","Y"))
cat(catMediation(labels=labels,data=mtcars))
cat(catMediation(X="am",Y="mpg",data=mtcars,moderator=moderator,covar=covar,maxylev=6))
cat(catMediation(X="am",Y="mpg",data=mtcars,moderator=moderator,covar=covar))
cat(catMediation(X="cyl",M="am",Y="mpg",data=mtcars))
cat(catMediation(X="cyl",M="am",Y="mpg",data=mtcars,moderator=moderator))
cat(catMediation(X="cyl",M="am",Y="mpg",data=mtcars,moderator=moderator))
cat(catMediation(X="am",M="hp",Y="mpg",data=mtcars,moderator=moderator,maxylev=6))
cat(catMediation(X="hp",M="am",Y="mpg",data=mtcars,maxylev=6))
cat(catMediation(X="am",M="hp",Y="mpg",data=mtcars,moderator=moderator,covar=covar))
```

centerPrint *Print a string in center*

Description

Print a string in center

Usage

```
centerPrint(string, width)
```

Arguments

string	A string
width	A numeric

changeLabelName *Change Label Names*

Description

Change Label Names

Usage

```
changeLabelName(x, labels, add = FALSE)
```

Arguments

x	A character vector
labels	A list
add	A logical

Examples

```
labels=list(X="frame:test",Mi="empathy",Y="intervention",W="frame",Z="test")
x=c("skeptic","test","empathy","skeptic:frame:test","D1:frame","frame:test")
changeLabelName(x,labels)
changeLabelName(x,labels,add=TRUE)
x=c("baby","milk","baby:milk")
labels=list(X="baby",M=c("wine","tent","sand"),Y="tile",W="milk")
changeLabelName(x,labels)
```

checkEquationVars *Check dependent variables in equations*

Description

Check dependent variables in equations

Usage

```
checkEquationVars(equation)
```

Arguments

equation	A string of regression formula

Examples

```
equation="M1~X*M*W+W*Z\nM2~X+M1+X"
checkEquationVars(equation)
```

checkEqVars *Check dependent variables in a equation*

Description

Check dependent variables in a equation

Usage

```
checkEqVars(eq)
```

Arguments

eq	A string of regression formula

Examples

```
eq="M2~X+M+X+X*M*W"
checkEqVars(eq)
eq="Y~M+W+M:W+X+W+X:W"
checkEqVars(eq)
```

compareMC *Compare effects of mean-centering and standardization of model*

Description

Compare effects of mean-centering and standardization of model

Usage

```
compareMC(fit, mode = 1)
```

Arguments

fit	An object of class 'lm'
mode	integer

Value

if mode is 1, an object of modelSummary2. Otherwise lift of models

Examples

```
fit=lm(govact~negemot*age,data=glbwarm)
compareMC(fit)
compareMC(fit,mode=2)
```

compareMCTable *Make table comparing effects of mean-centering and standardization of model*

Description

Make table comparing effects of mean-centering and standardization of model

Usage

```
compareMCTable(fit, vanilla = TRUE)
```

Arguments

fit	An object of class 'lm'
vanilla	logical.

compareVIF — *Compare correlation, tolerance, vif of mean-centered and standardized models*

Description

Compare correlation, tolerance, vif of mean-centered and standardized models

Usage

```
compareVIF(fit)
```

Arguments

fit	An object of class lm

Examples

```
fit=lm(govact~negemot*age,data=glbwarm)
compareVIF(fit)
```

compareVIFTable — *Make table comparing correlation, tolerance, vif of mean-centered and standardized models*

Description

Make table comparing correlation, tolerance, vif of mean-centered and standardized models

Usage

```
compareVIFTable(fit, vanilla = TRUE)
```

Arguments

fit	An object of class lm
vanilla	logical

Examples

```
fit=lm(govact~negemot*age,data=glbwarm)
compareVIFTable(fit)
compareVIFTable(fit,vanilla=FALSE)
```

conceptDiagram *Make conceptDiagram*

Description

Make conceptDiagram

Usage

```
conceptDiagram(fit, labels = NULL)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
labels	labels

conceptDiagram2 *Make concept Diagram*

Description

Make concept Diagram

Usage

```
conceptDiagram2(
  X = "X",
  M = "M",
  Y = "Y",
  latent = rep(FALSE, 3),
  xb = FALSE,
  mc = FALSE,
  radx = 0.06,
  rady = 0.06,
  xmargin = 0.03,
  yinterval = NULL,
  box.col = "white",
  xlim = NULL,
  ylim = NULL,
  moderator = list(),
  labels = list(),
  covar = list()
)
```

Arguments

X	character Name of independent variable
M	character Name of mediator variable
Y	character Name of dependent variable
latent	Logical. whether or not X,Y and Z are latent variables or not
xb	Logical. if positive draw line between X and (Y+Z)
mc	Logical. if positive draw line between M and (X+Y)
radx	horizontal radius of the box.
rady	vertical radius of the box.
xmargin	horizontal margin of plot
yinterval	vertical interval between box
box.col	fill color of box
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot
moderator	optional list of moderators
labels	optional labels of X,Y and Z variables
covar	covariate optional list of covariates

Examples

```
labels=list(X="Time Spent in\n Grad School", M="# of\n Publications", Y="# of Job Offers")
conceptDiagram2(xb=TRUE,labels=labels)
moderator=list(name="Z1",label="Time Spent\n with Alex",pos=3,
    site=list(c("a","b","c")),latent=FALSE)
conceptDiagram2(moderator=moderator,labels=labels)
moderator=list(name=c("Z1","Z2"),label=c("Time Spent\n with Alex","Z2label"),pos=c(3,3),
    site=list(c("a","b","c"),c("b","c")),latent=c(FALSE,FALSE))
conceptDiagram2(moderator=moderator,labels=labels,yinterval=0.4)
covar=list(name=c("C1","C2"),label=c("sex","tenure"),site=list(c("Y"),c("Y")))
conceptDiagram2(M=NULL,moderator=list(name="M",pos=4,site=list("c"),latent=FALSE),covar=covar)
conceptDiagram2(covar=covar)
```

---

conditionalEffectPlot *Make conditional effect plot*

Description

Make conditional effect plot

Usage

```
conditionalEffectPlot(
  semfit,
  values = NULL,
  data,
  no = 1,
  mod = NULL,
  color = c("black", "red"),
  lty = c(1, 3),
  linesize = 1
)
```

Arguments

semfit	An object of class lavaan
values	Optional value
data	A data.frame
no	Integer
mod	Name of moderator variable
color	character vector line color
lty	numeric line type
linesize	numeric linesize

---

condPlot *Draw conditional effect plot*

Description

Draw conditional effect plot

Usage

```
condPlot(
  fit,
  xmode = 1,
  pred = NULL,
  modx = NULL,
  pred.values = NULL,
  modx.values = NULL,
  labels = NULL,
  mode = 1,
  rangemode = 1,
  ypos = NULL,
  hjust = NULL,
```

```
  linecolor = "gray60",
  linetype = 2,
  linesize = 1,
  arrowsize = 1,
  digits = 3,
  depM = FALSE,
  ...
)
```

Arguments

fit	An object of class lm
xmode	integer. 1 or 2.
pred	name of predictor variable
modx	name of moderator variable
pred.values	Values of predictor variables
modx.values	Values of modifier variables
labels	labels of regression lines
mode	integer. one of 1:3.
rangemode	integer. 1 or 2
ypos	integer. label y position.
hjust	hjust of label
linecolor	name of color of vline and hline
linetype	linetype of arrow
linesize	size of regression line
arrowsize	size of arrow
digits	integer indicating the number of decimal places
depM	logical If true, label M instead of X
...	further arguments to be passed to add_lines

Examples

```
fit=lm(justify~frame*skeptic,data=disaster)
condPlot(fit,rangemode=2,xpos=0.7,labels=c("Climate change(X=1)","Natural causes(X=0)"))

condPlot(fit,mode=2,xpos=0.6)
condPlot(fit,mode=3,rangemode=2,xpos=0.5)
condPlot(fit,xmode=2)
condPlot(fit,xmode=2,mode=2)
condPlot(fit,xmode=2,mode=3)
fit=lm(mpg~vs*hp,data=mtcars)
condPlot(fit,rangemode=2,xpos=0.6)
condPlot(fit,mode=2,xpos=0.5)
condPlot(fit,mode=3,rangemode=2)
fit=lm(govact~negemot*age+posemot+ideology+sex,data=glbwarm)
```

```
condPlot(fit,xmode=2,hjust=c(-0.1,-0.1,1.1))
condPlot(fit,xmode=2,pred.values=c(30,70),hjust=c(-0.1,-0.1,1.1),xpos=0.5)
condPlot(fit,xmode=2,mode=2,pred.values=c(30,50,70),xpos=0.2)
condPlot(fit,xmode=2,mode=3,xpos=0.5,hjust=c(-0.1,-0.1,1.1))
condPlot(fit,xmode=2,modx.values=c(2,3,4),mode=3,xpos=0.6)
```

`condPlot2` *Draw conditional plot for moderated moderation*

Description

Draw conditional plot for moderated moderation

Usage

```
condPlot2(
  fit,
  pred = NULL,
  modx = NULL,
  mod2 = NULL,
  mod2.values = NULL,
  rangemode = 1,
  vjust = NULL,
  digits = 3,
  addlabel = TRUE,
  xvar = "Z",
  ...
)
```

Arguments

`fit`	An object of class lm
`pred`	name of the predictor variable
`modx`	name of the moderator variable
`mod2`	name of the second moderator variable
`mod2.values`	values of moderator variable
`rangemode`	integer. 1 or 2
`vjust`	integer
`digits`	integer indicating the number of decimal places
`addlabel`	logical
`xvar`	character. "Z" or "W"
`...`	Further arguments to be passed to predict3d::ggPredict()

Examples

```
fit=lm(govact~negemot*sex*age+posemot+ideology,data=glbwarm)
## Not run:
condPlot2(fit)
condPlot2(fit,mod2.values = c(30,50,70))
fit1=lm(govact~negemot*age*sex+posemot+ideology,data=glbwarm)
condPlot2(fit1,pred="negemot",modx="sex",mod2="age",mod2.values = c(30,50,70),xvar="W")

## End(Not run)
```

---

condPlotCat *Make conditional effect plot with data including a categorical variable*

Description

Make conditional effect plot with data including a categorical variable

Usage

```
condPlotCat(
  labels = list(),
  yvar = "Y",
  total = FALSE,
  data,
  addvars = TRUE,
  mode = 1,
  rangemode = 2,
  maxylev = 6,
  catlabels = NULL,
  add.slopelabel = FALSE,
  xpos = 0.5,
  add.point = TRUE,
  add.vlines = TRUE,
  add.vlines.text = TRUE,
  add.anova = TRUE,
  ypos = NULL,
  add.arrow = TRUE,
  xinterval = NULL,
  hjust1 = NULL,
  hjust2 = NULL,
  ypos2 = NULL,
  ypos3 = NULL,
  ceno = 1
)
```

Arguments

labels	Named list of variables
yvar	character. "Y"(default) or "M"
total	logical. If true, model include mediator variable.
data	A data.frame
addvars	logical
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding
rangemode	rangemode. 1 or 2.
maxylev	maximal unique length of categorical variable
catlabels	optional string of labels for the categorical variable
add.slopelabel	logical
xpos	numeric. x position of slope labels
add.point	logical. If true, add point to the plot
add.vlines	logical. If true, add vlines to the plot
add.vlines.text	logical. If true, add vlines.text to the plot
add.anova	logical. If true, add results of ANOVA to the plot
ypos	optional. Y position of anova results
add.arrow	logical. If true, add conditional effects to the plot
xinterval	Integer. Width of angled arrow
hjust1	optional. hjust of conditional effects 1
hjust2	optional. hjust of conditional effects 2
ypos2	optional. Y position of conditional effects 1
ypos3	optional. Y position of conditional effects 2
ceno	integer. 1 or 2

Examples

```
library(ggplot2)
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
catlabels=c("No protest","Individual protest","Collective protest")
catlabels2=c("No protest","Individual protest","Collective protest","Any protest")
condPlotCat(labels=labels,yvar="M",data=protest,mode=3,ypos=c(0.2,0.15,0.1))
condPlotCat(labels=labels,yvar="M",data=protest,mode=3,ceno=c(1,2),add.vlines.text=FALSE)
condPlotCat(labels=labels,catlabels=catlabels,yvar="M",data=protest,mode=3,
     add.arrow=FALSE,addvars=FALSE)
condPlotCat(labels=labels,yvar="M",data=protest,mode=3,catlabels=catlabels2,ceno=c(1,2))
condPlotCat(labels=labels,data=protest,catlabels=catlabels,add.slopelabel=TRUE,
    xpos=c(0.3,0.7,0.7),add.point=FALSE,add.vlines=FALSE,add.anova=FALSE,add.arrow=FALSE)
condPlotCat(labels=labels,data=protest,catlabels=catlabels,add.anova=FALSE,add.arrow=FALSE)
condPlotCat(labels=labels,data=protest,catlabels=catlabels,add.anova=FALSE)+xlim(c(3.5,6.5))
```

```
condPlotCat(labels=labels,data=protest,add.anova=TRUE,ypos=c(0.2,0.2,0.5),add.arrow=FALSE)
condPlotCat(labels=labels,data=protest,catlabels=catlabels,add.anova=FALSE,ceno=1)
condPlotCat(labels=labels,data=protest,catlabels=catlabels,add.anova=FALSE,ceno=2)
condPlotCat(labels=labels,data=protest,total=TRUE,catlabels=catlabels,ypos=0.1,
      add.arrow=FALSE)+xlim(c(4,6))
condPlotCat(labels=labels,data=protest,total=TRUE,catlabels=catlabels2,add.anova=FALSE,
      ceno=c(1,2),xinterval=0.05,hjust1=c(-0.05,-0.05,1.05),hjust2=c(-0.05,1.05,1.05),
      ypos2=c(0.5,0.1,0.3),ypos3=c(0.2,0.4,0.4),mode=3)+xlim(c(4,6))
```

`condPlotCat2` *Draw direct and indirect effect plot*

Description

Draw direct and indirect effect plot

Usage

```
condPlotCat2(
  labels = NULL,
  data = NULL,
  semfit,
  catlabels = NULL,
  digits = 3,
  add.point = FALSE,
  ...
)
```

Arguments

`labels`	list of variable labels
`data`	data.frame
`semfit`	An object of class lavaan
`catlabels`	labels for direct/indirect effects
`digits`	Integer indicating the number of decimal places
`add.point`	logical. Whether or not add points to the plot
`...`	further arguments to be passed to predict3d::add_lines()

Examples

```
library(lavaan)
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
moderator=list(name="sexism",site=list(c("a","c")))
data1=addCatVars(protest,"protest",mode=3)
catlabels=c("Indirect: Protest\n   vs. No Protest",
      "Indirect: Collective\n       vs. Individual",
```

```
      "Direct: Protest\n vs. No Protest",
      "Direct: Collective\n      vs. Individual")
model=catMediation(X="protest",M="respappr",Y="liking",moderator=moderator,
       data=data1,maxylev=6,rangemode = 2)
semfit=sem(model=model,data=data1)
condPlotCat2(labels=labels,data=data1,semfit=semfit,catlabels=catlabels,
       xpos=c(0.7,0.3,0.3,0.7),add.point=TRUE)
```

`convertPvalue` *convert vector of p values to string*

Description

convert vector of p values to string

Usage

```
convertPvalue(x)
```

Arguments

`x`	vector of p values

`corPlot` *Draw correlation plot*

Description

Draw correlation plot

Usage

```
corPlot(
  fit,
  label = 2,
  yreverse = TRUE,
  xangle = 45,
  seek = NULL,
  replace = NULL,
  ...
)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
label	if 0, no label(default), if 1, use r value as label, if 2, use r value with significant mark as label
yreverse	Logical. if true, reverse the order of y axis.
xangle	axis.x.text.angle
seek	string to look for
replace	A string of replacement
...	Further arguments to be passed on to geom_text

Value

A ggplot

---

corTable *Make a table with correlation*

Description

Make a table with correlation

Usage

```
corTable(fit)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan

---

corTable2 *Make a table with correlation*

Description

Make a table with correlation

Usage

```
corTable2(fit, vanilla = TRUE, addFooter = FALSE, seek = NULL, replace = NULL)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
vanilla	Logical. If true, vanilla.table is returned
addFooter	Logical. If true, footer added
seek	string to look for
replace	A string of replacement

---

countM *Count the group names start with "M"*

Description

Count the group names start with "M"

Usage

```
countM(group)
```

Arguments

group	A string vectors

---

covar2df *Make data.frame with covariates*

Description

Make data.frame with covariates

Usage

```
covar2df(covar = list(), df)
```

Arguments

covar	A list
df	A data.frame

deleteSingleNumber *remove coefficient number of equation*

Description

remove coefficient number of equation

Usage

```
deleteSingleNumber(equation)
```

Arguments

equation	string

densityPlot *Draw Smoothed Kernel density plot*

Description

Draw Smoothed Kernel density plot

Usage

```
densityPlot(
  x,
  sig = 0.05,
  digits = 3,
  xlab = "Indirect effect(ab)",
  ylab = NULL
)
```

Arguments

x	A numeric vector
sig	significant level. Default value is 0.05
digits	Integer indicating the number of decimal places
xlab	character. x axis label
ylab	character. y axis label

Examples

```
require(lavaan)
labels=list(X="cond",M="pmi",Y="reaction")
model=tripleEquation(labels=labels)

set.seed(1234)
semfit=sem(model,data=pmi,se="boot",bootstrap=100)
bootData=getBootData(semfit)
bootData$indirect=bootData$a*bootData$b
densityPlot(bootData$indirect)
```

---

disaster *Disaster dataset*

Description

Disaster dataset

Usage

```
disaster
```

Format

A data.frame with 211 obs. of 5 variables

id id

frame Experimental condition. 0 = naturally caused disaster, 1 = climate change caused disaster

donate Positive attitudes toward donating

justify Negative justifications

skeptic Climate change skepticism

Source

Chapman, D. A., & Little, B. (2016). Climate change and disasters: How framing affects justifications for giving or withholding aid to disaster victims. Social Psychological and Personality Science, 7, 13-20.

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

discriminantValidityTable *make discriminant Validity Table*

Description

make discriminant Validity Table

Usage

```
discriminantValidityTable(fit)
```

Arguments

fit	An object of a class lavaan

discriminantValidityTable2 *make discriminant Validity Table in flextable format*

Description

make discriminant Validity Table in flextable format

Usage

```
discriminantValidityTable2(fit, vanilla = FALSE)
```

Arguments

fit	An object of a class lavaan
vanilla	Logical

divideEquation *divide equation*

Description

divide equation

Usage

```
divideEquation(equation)
```

Arguments

equation	a string

Examples

```
equation="(a1+a3*W)*(b)"
divideEquation(equation)
```

drawArrows *Draw arrows*

Description

Draw arrows

Usage

```
drawArrows(
  arrows,
  nodes,
  xmargin = 0.01,
  radx = 0.1,
  rady = 0.04,
  addprime = TRUE
)
```

Arguments

arrows	A data.frame
nodes	A data.frame
xmargin	horizontal margin of plot
radx	horizontal radius of the box.
rady	vertical radius of the box.
addprime	logical Whether add prime to label "c"

drawCatModel *Draw statistical diagram including categorical X*

Description

Draw statistical diagram including categorical X

Usage

```
drawCatModel(
  xcount = 3,
  M = NULL,
  W = NULL,
  whatLabel = "name",
  addDots = TRUE,
  xmargin = 0.01,
  radx = 0.08,
  ymargin = 0.02,
  xlim = c(-0.2, 1.2),
  ylim = xlim,
  rady = 0.04,
  maxypos = 0.6,
  minypos = 0.2,
  ypos = c(1, 0.5),
  mpos = c(0.5, 0.9),
  xinterval = NULL,
  yinterval = NULL,
  box.col = "white",
  xspace = NULL,
  label.pos = list()
)
```

Arguments

xcount	integer length of categorical variables
M	character name of mediator variable
W	character name of moderator variable
whatLabel	What should the edge labels indicate in the path diagram? Choices are c("est","name")
addDots	logical.
xmargin	horizontal margin between nodes
radx	horizontal radius of the box.
ymargin	vertical margin between nodes
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot

rady	vertical radius of the box.
maxypos	maximal y position of X or W variables
minypos	minimal y position of X or W variables
ypos	The x and y position of Y node. Default value is c(1,0.5)
mpos	The x and y position of M node. Default value is c(0.5,0.9)
xinterval	numeric. Horizontal intervals among labels for nodes and nodes
yinterval	numeric. Vertical intervals among labels for nodes and nodes
box.col	fill color of the box
xspace	numeric. Horizontal distance between nodes
label.pos	Optional list of arrow label position

Examples

```
drawCatModel(xcount=4)
drawCatModel(M="M",box.col="yellow")
drawCatModel(W="W",xlim=c(-0.08,1),ylim=c(0.13,0.95),ypos=c(1,0.55))
drawCatModel(M="M",W="W",xlim=c(-0.08,1),ylim=c(0.13,0.95),ypos=c(1,0.55))
drawCatModel(xcount=4,M="M",W="W",xlim=c(-0.08,1),ylim=c(0.13,0.95),ypos=c(1,0.55))
```

---

drawConcept *Draw Concept Diagram*

Description

Draw Concept Diagram

Usage

```
drawConcept(
  labels,
  nodelabels = list(),
  vars = NULL,
  moderator = NULL,
  covar = NULL,
  nodemode = 1,
  xpos = c(0, 0.5),
  mpos = c(0.5, 0.9),
  ypos = c(1, 0.5),
  minypos = 0,
  maxypos = 0.6,
  node.pos = list(),
  serial = FALSE,
  parallel = FALSE,
  parallel2 = FALSE,
  parallel3 = FALSE,
```

```
  bmatrix = NULL,
  curved.arrow = NULL,
  segment.arrow = NULL,
  radx = 0.06,
  rady = 0.04,
  box.col = "white",
  palette = NULL,
  reverse = FALSE,
  xmargin = 0.02,
  ymargin = 0.02,
  showPos = FALSE,
  xinterval = NULL,
  yinterval = NULL,
  label.pos = 1,
  drawbox = FALSE
)
```

Arguments

labels	A list
nodelabels	A list
vars	A list of triple moderators
moderator	A list of modeators
covar	A list of covariates
nodemode	integer If 1, separate node name and node label
xpos	The x and y position of X node. Default value is c(0,0.5)
mpos	The x and y position of M node. Default value is c(0.5,0.9)
ypos	The x and y position of Y node. Default value is c(1,0.5)
minypos	minimal y position of X or W variables
maxypos	maximal y position of X or W variables
node.pos	A optional list of node position
serial	Logical. If TRUE, serial variables are added
parallel	logical If true, draw parallel multiple mediation model
parallel2	logical If true, draw parallel2 multiple mediation model
parallel3	logical If true, draw parallel3 multiple mediation model
bmatrix	integer specifying causal relations among mediators
curved.arrow	Optional numeric vector specifying curvedarrow
segment.arrow	Optional numeric vector specifying segmentarrow
radx	horizontal radius of the box.
rady	vertical radius of the box.
box.col	fill color of the box
palette	character. palette name

reverse	logical. Reverse palette or not.
xmargin	horizontal margin between nodes
ymargin	vertical margin between nodes
showPos	logical If true print node position
xinterval	numeric. Horizontal intervals among labels for nodes and nodes
yinterval	numeric. Vertical intervals among labels for nodes and nodes
label.pos	Integer Position of nodelabels. Choices are one of 1:2
drawbox	logical If true, draw rectangle

Examples

```
labels=list(X="estress",M="affect",Y="withdraw")
vars=list(name=list(c("tenure","age")),site=list(c("a","b")))
moderator=list(name=c("age","sex"),site=list(c("c"),c("b","c")),pos=c(1,2),
     arr.pos=list(c(0.3),c(0.3,0.7)))
drawConcept(labels=labels)
drawConcept(labels=labels,vars=vars,drawbox=TRUE)
drawConcept(labels=labels,moderator=moderator,drawbox=TRUE)
drawConcept(labels=labels,vars=vars,moderator=moderator,drawbox=TRUE)
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
drawConcept(labels=labels,serial=TRUE)
drawConcept(labels=labels,parallel=TRUE,bmatrix=c(1,1,0,1,0,0,1,1,1,1))
drawConcept(labels=labels,parallel2=TRUE,bmatrix=c(1,1,0,1,0,0,1,1,1,1))
labels=list(X="baby",M=c("wine","tent","sand"),Y="tile")
bmatrix=c(1,1,0,1,0,0,1,1,1,1)
drawConcept(labels=labels,parallel=TRUE,bmatrix=bmatrix)
moderator=list(name=c("milk","hair"),
  matrix=list(c(1,1,0,1,0,0,0,0,0,0),c(0,0,0,0,0,0,0,1,0,0)))
drawConcept(labels=labels,parallel=TRUE,bmatrix=bmatrix,moderator=moderator)
bmatrix=c(1,1,0,0,1,1,1,1,0,1)
moderator=list(name=c("milk","hair"),
            matrix=list(c(1,0,0,0,1,0,1,0,0,0),c(1,1,0,0,0,0,0,0,0,0)),
            pos=c(1,4))
node.pos=list(X=c(0,0.5),M1=c(0.3,0.9),M2=c(0.3,0.1),M3=c(0.7,0.9),
Y=c(1,0.5),W1=c(0.7,0.1),W2=c(0,0.9))
drawConcept(labels=labels,bmatrix=bmatrix,moderator=moderator,node.pos=node.pos)
labels=list(X="baby",M=c("wine","tent","sand"),Y="tile")
vars=list(name=list(c("milk","hair")),matrix=list(c(1,0,0,0,0,0,1,0,0,0)),pos=2)
bmatrix=c(1,1,0,1,0,0,1,1,1,1)
drawConcept(labels=labels,parallel=TRUE,bmatrix=bmatrix,vars=vars)
labels=list(X="X",M=c("M1","M2"),Y="Y")
vars=list(name=list(c("W","Z")),matrix=list(c(0,0,1,0,0,0)),pos=6)
bmatrix=c(1,1,1,1,1,1)
drawConcept(labels=labels,bmatrix=bmatrix,vars=vars,palette="Set3")
labels=list(X="X",M="M",Y="Y")
vars=list(name=list(c("W","Z")),site=list(c("a","c")),arr.pos=list(c(0.7,0.3)))
moderator=list(name=c("V","Q"),site=list(c("b","c"),c("c")),
   pos=c(2,5),arr.pos=list(c(0.3,0.7),0.5))
drawConcept(labels=labels,vars=vars,moderator=moderator,nodemode=2)
```

drawCovar *Draw covariate*

Description

Draw covariate

Usage

```
drawCovar(covar = list(), x, y, m, radx = 0.1, rady = 0.06, yinterval = 0.02)
```

Arguments

covar	A list
x	position of x
y	position of y
m	position of m
radx	horizontal radius of the box.
rady	vertical radius of the box.
yinterval	vertical interval between box

drawModel *Draw statistical diagram with an object of class lavaan or a list of class lm*

Description

Draw statistical diagram with an object of class lavaan or a list of class lm

Usage

```
drawModel(
  semfit = NULL,
  labels = NULL,
  equation = NULL,
  vars = list(),
  moderator = list(),
  covar = NULL,
  data = NULL,
  nodelabels = NULL,
  arrowslabels = NULL,
  whatLabel = "name",
  mode = 1,
  nodemode = 1,
```

```
  xmargin = 0.02,
  radx = NULL,
  ymargin = 0.02,
  xlim = NULL,
  ylim = NULL,
  box.col = "white",
  palette = NULL,
  reverse = FALSE,
  rady = 0.06,
  maxypos = NULL,
  minypos = 0,
  ypos = c(1, 0.5),
  mpos = c(0.5, 0.9),
  xinterval = NULL,
  yinterval = NULL,
  xspace = NULL,
  node.pos = list(),
  arrow.pos = list(),
  interactionFirst = FALSE,
  totalOnly = FALSE,
  parallel = FALSE,
  parallel2 = FALSE,
  parallel3 = FALSE,
  kmediator = FALSE,
  serial = FALSE,
  bmatrix = NULL,
  label.pos = 1,
  curved.arrow = list(),
  segment.arrow = list(),
  digits = 3,
  showPos = FALSE,
  drawCovar = TRUE,
  drawbox = FALSE
)
```

Arguments

semfit	An object of class lavaan or a list of class lm
labels	list of variable names
equation	Optional string contains equation
vars	A list
moderator	A list
covar	A list
data	A data.frame
nodelabels	list of nodes names
arrowslabels	list of arrows names

whatLabel	What should the edge labels indicate in the path diagram? Choices are c("est","name","estSE")
mode	integer If 1, models with categorical X
nodemode	integer If 1, separate node name and node label
xmargin	horizontal margin between nodes
radx	horizontal radius of the box.
ymargin	vertical margin between nodes
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot
box.col	fill color of the box
palette	character. palette name
reverse	logical. Reverse palette or not.
rady	vertical radius of the box.
maxypos	maximal y position of X or W variables
minypos	minimal y position of X or W variables
ypos	The x and y position of Y node. Default value is c(1,0.5)
mpos	The x and y position of M node. Default value is c(0.5,0.9)
xinterval	numeric. Horizontal intervals among labels for nodes and nodes
yinterval	numeric. Vertical intervals among labels for nodes and nodes
xspace	numeric. Horizontal distance between nodes
node.pos	Optional list of node position
arrow.pos	Optional list of arrow label position
interactionFirst	logical If true, place nodes with interaction first
totalOnly	logical If true, draw total effect model only
parallel	logical If true, draw parallel multiple mediation model
parallel2	logical If true, draw parallel2 multiple mediation model
parallel3	logical If true, draw parallel3 multiple mediation model
kmediator	logical If true, draw parallel multiple mediation model with k mediator
serial	Logical. If TRUE, serial variables are added
bmatrix	integer specifying causal relations among mediators
label.pos	Integer Position of nodelabels. Choices are one of 1:2
curved.arrow	Optional list of curved arrow
segment.arrow	Optional list of curved arrow
digits	integer indicating the number of decimal places
showPos	logical If true print node position
drawCovar	logical If true, draw covariates
drawbox	logical If true, draw rectangle

Examples

```
library(lavaan)
labels=list(X="frame",Y="donate")
drawModel(labels=labels)
drawModel(labels=labels,arrowslabels=list(c="c"))
labels=list(X="frame",W="skeptic",M="justify",Y="donate")
moderator=list(name="skeptic",site=list(c("a","c")))
model=tripleEquation(labels=labels,moderator=moderator,data=disaster)
semfit=sem(model=model,data=disaster)
drawModel(semfit,labels=labels,interactionFirst=TRUE)
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
moderator=list(name="sexism",site=list(c("a","c")))
data1=addCatVars(protest,"protest",mode=3)
model=catMediation(X="protest",M="respappr",Y="liking",moderator=moderator,data=protest,maxylev=6)
semfit=sem(model,data=data1)
nodelabels=list(D1="Ind.Protest",D2="Col.Protest",W="sexism",M="respappr",Y="liking")
drawModel(semfit,labels=labels,nodelabels=nodelabels,whatLabel="name",
       xlim=c(-0.4,1.3))
drawModel(semfit,labels=labels)
labels=list(X="cyl",M=c("am","wt","hp"),Y="mpg",W="vs")
moderator=list(name=c("vs"),site=list(c("a1","b1")))
model=multipleMediation(labels=labels,moderator=moderator,data=mtcars)
semfit=sem(model=model,data=mtcars)
drawModel(semfit,labels=labels,maxypos=0.5)
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
nodelabels=c(X="Intervention\n(vs.control)",
  M=c("Restrained\nEating","Emotional\nEating","Perceived\nBarriers to\nExercise"),Y="Weight Loss")
drawModel(labels=labels,nodelabels=nodelabels,whatLabel="none",parallel=TRUE,
ylim=c(-0.3,1.2),label.pos=2)
labels=list(X="X",M=c("M1","M2","Mk-1","Mk"),Y="Y")
drawModel(labels=labels,parallel=TRUE,kmediator=TRUE,nodemode=2,
    arrow.pos=list(c=0.4),serial=FALSE,radx=0.08)
labels=list(X="cond",M=c("import","pmi"),Y="reaction")
drawModel(labels=labels,parallel=TRUE)
drawModel(labels=labels,parallel=TRUE,serial=TRUE)
model=multipleMediation(labels=labels,data=pmi,serial=TRUE)
model=multipleMediation(labels=labels,data=pmi)
cat(model)
semfit=sem(model=model,data=pmi)
drawModel(semfit,labels=labels,parallel=TRUE)
drawModel(semfit,labels=labels,whatLabel="est",parallel=TRUE)
labels=list(X="X",M=c("M1","M2"),Y="Y")
drawModel(labels=labels,serial=TRUE,nodemode=4)
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
drawModel(labels=labels,serial=TRUE)
equation='M1~X
M2~X+M1
M3~X+M1
Y~X+M1+M2+M3'
node.pos=list(X=c(0,0.5),M1=c(0.5,0.5),M2=c(0.75,0.9),M3=c(0.75,0.1),Y=c(1,0.5))
curved.arrow=list(a2=-0.1,a3=0.1,c=-0.15)
drawModel(equation=equation,nodemode=2,node.pos=node.pos,curved.arrow=curved.arrow)
```

```
equation='M1~X
M2~X
M3~X
M4~X+M1+M2+M3
Y~X+M1+M2+M3+M4'
node.pos=list(X=c(0,0.5),M1=c(0.35,0.9),M2=c(0.35,0.5),M3=c(0.35,0.1),M4=c(0.7,0.5),Y=c(1,0.5))
curved.arrow=list(a4=0.15,b2=0.15)
segment.arrow=list(c=0.5)
drawModel(equation=equation,nodemode=2,node.pos=node.pos,radx=0.08,curved.arrow=curved.arrow,
segment.arrow=segment.arrow)
labels=list(X="baby",M="wine",Y="tile")
moderator=list(name=c("milk"),site=list("a"))
covar=list(name=c("milk","tent","sand"),site=list(c("Y"),c("M","Y"),c("M","Y")))
drawModel(labels=labels,moderator=moderator,covar=covar,palette="Set3")
```

---

drawStatDiagram *draw StatDiagram*

Description

draw StatDiagram

Usage

```
drawStatDiagram(
  no,
  arrows,
  nodes,
  labels,
  nodeslabels = list(),
  xmargin,
  radx,
  rady,
  fit = NULL,
  addprime = TRUE,
  box.col = "white",
  xlim = c(0, 1),
  ylim = c(0, 1)
)
```

Arguments

no	process macro model number
arrows	A data.frame
nodes	A data.frame
labels	A list
nodeslabels	A list

xmargin	horizontal margin of plot
radx	horizontal radius of the box.
rady	vertical radius of the box.
fit	An object of class lavaan. Result of lavaan::sem()
addprime	logical Whether add prime to label "c"
box.col	fill color of the box
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot

`drawtext` *Draw node*

Description

Draw node

Usage

```
drawtext(..., latent = TRUE)
```

Arguments

...	Further argument to be passed to textellipse() or textrect()
latent	Logical

`education` *Data Set for education and income*

Description

A dataset contains measures about the teacher's knowledge, empathy and intervention about attention-deficit hyperactivity disorder(ADHD).

Usage

```
education
```

Format

A data.frame with 850 rows and 4 variables:

age student age

number number of students per class

duration eduation duration

income income

eq2df *Convert equation to data.frame*

Description

Convert equation to data.frame

Usage

```
eq2df(eq)
```

Arguments

eq	equation separated by linefeed

eq2fit *Make a list of objects of class lm*

Description

Make a list of objects of class lm

Usage

```
eq2fit(equations, data)
```

Arguments

equations	equations for linear regression
data	A data.frame

Value

a list of objects of class lm

eq2var *make data.frame with equation*

Description

make data.frame with equation

Usage

```
eq2var(eq, labels = list())
```

Arguments

eq	equation
labels	A list

Examples

```
labels=list(X="frame",M="justify",Y="donate",W="skeptic")
eq="donate~justify+frame+skeptic+frame:skeptic"
eq2var(eq,labels=labels)
eq="Y~M+W+M:W+X+X:W"
labels=list(X="X",M="M",Y="Y")
eq2var(eq,labels=labels)
eq="wine~baby+milk+baby:milk"
labels=list(X="baby",M=c("wine","tent","sand"),Y="tile",W="milk")
eq2var(eq,labels=labels)
eq="M3~X"
eq="Y~M1+M2+X"
labels=list(X="X",M=c("M1","M2"),Y="Y")
eq2var(eq,labels=labels)
```

equations2var *make data.frame with equation*

Description

make data.frame with equation

Usage

```
equations2var(eq, labels = list())
```

Arguments

eq	equation
labels	A list

Examples

```
labels=list(X="frame",M="justify",Y="donate",W="skeptic")
moderator=list(name="skeptic",site=list(c("a","c")))
eq=multipleMediation(labels=labels,moderator=moderator,mode=1)
covar=list(name=c("C1","C2","C3"),site=list(c("M","Y"),c("M","Y"),c("M","Y")))
eq=multipleMediation(labels=labels,covar=covar,mode=1)
equations2var(eq,labels=labels)
```

`est2Arrows` *Make arrows from estimatesTable*

Description

Make arrows from estimatesTable

Usage

```
est2Arrows(res)
```

Arguments

`res`	A data.frame, result of estimatesTable

`est2Nodes` *Make nodes from estimatesTable*

Description

Make nodes from estimatesTable

Usage

```
est2Nodes(res, lastxno = 2)
```

Arguments

`res`	A data.frame, result of estimatesTable
`lastxno`	A numeric

estimatesTable *convert parameterEstimates to data.frame*

Description

convert parameterEstimates to data.frame

Usage

```
estimatesTable(
  fit,
  latent = TRUE,
  regression = TRUE,
  mediation = FALSE,
  covar = FALSE,
  ci = FALSE,
  standardized = TRUE,
  digits = 2
)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
latent	whether the latent variables be included in result
regression	whether the regressions be included in result
mediation	whether the mediation effects be included in result
covar	whether the covariances be included in result
ci	If TRUE, confidence intervals are added to the output
standardized	Logical. If TRUE, standardized estimates are added to the output
digits	integer indicating the number of decimal places to be used.

estimatesTable2 *convert parameterEstimates to flextable*

Description

convert parameterEstimates to flextable

Usage

```
estimatesTable2(
  fit,
  vanilla = TRUE,
  digits = 3,
  seek = NULL,
  replace = NULL,
  ...
)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
vanilla	Logical
digits	integer indicating the number of decimal places to be used.
seek	string to look for
replace	A string of replacement
...	Further arguments to be passed to estimatesTable()

---

estress *ESTRESS: Economic stress dataset*

Description

ESTRESS: Economic stress dataset

Usage

```
estress
```

Format

A data.frame with 262 obs. of 7 variables

tenure Company Tenure

estress Economic stress

affect Depressed affect

withdraw Withdrawal intentions

sex Male (1) or Female (0)

age age

ese Entrepreneurial self efficacy

Source

Pollack, J., VanEpps, E. M., & Hayes, A. F. (2012). The moderating role of social ties on entrepreneurs' depressed affect and withdrawal intentions in response to economic stress. Journal of Organizational Behavior, 33, 789-810.

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

---

extractIMM *extract index of moderated mediation from string*

Description

extract index of moderated mediation from string

Usage

```
extractIMM(string)
```

Arguments

string	A string

Examples

```
string="(a1+a3*age.mean)*(b1+b3*age.mean)"
string="(a1+a3*skeptic.mean)*(b1+b2*skeptic.mean+b4*Z.mean)"
string="(a1+a3*age.mean)*(b)"
string="(a1+a3*4.12)*(b)"
string="(a)*(b)"
extractIMM(string)
```

---

extractLatentVar *Extract Latent Variables Data*

Description

Extract Latent Variables Data

Usage

```
extractLatentVar(fit, labels)
```

Arguments

fit	An object of class lavaan. Result of lavaan::sem()
labels	A list

extractLatentVarName *Extract Latent Variables Names*

Description

Extract Latent Variables Names

Usage

```
extractLatentVarName(fit)
```

Arguments

fit	An object of class lavaan. Result of lavaan::sem()

extractModerator *Extract name of moderator from string*

Description

Extract name of moderator from string

Usage

```
extractModerator(string)
```

Arguments

string	A string

Examples

```
string="(a1+a3*age.mean)*(b1+b3*age.mean)"
string="(a1+a3*age.mean)*(b)"
string="(a1+a3*4.12)*(b)"
string="(a)*(b)"
extractNumber(string)
extractModerator(string)
```

extractNumber *extract number from string*

Description

extract number from string

Usage

```
extractNumber(x)
```

Arguments

x	a string

extractRange *Extract range from a data.frame*

Description

Extract range from a data.frame

Usage

```
extractRange(res, mod, what = "indirect")
```

Arguments

res	A data.frame
mod	Name of moderator
what	string

extractX *Extract group by string*

Description

Extract group by string

Usage

```
extractX(string, groupby = "X")
```

Arguments

string	character vector
groupby	name of groupby

findName *convert name with list*

Description

convert name with list

Usage

```
findName(labels, nodeslabels = list(), name = "MiX", exact = FALSE)
```

Arguments

labels	A named list
nodeslabels	A named list
name	A name to look for
exact	A logical

Examples

```
labels=list(X="wt",M="am",Y="mpg");name="MiX"
nodeslabels=list(X="weight",M="automatic",Y="milepergallon")
findName(labels=labels,nodeslabels=nodeslabels,name="MiX")
findName(labels=labels,name="MiX")
findName(labels=labels,nodeslabels=nodeslabels,name="M")
labels=list(X="GDPpp",M="Illit",Y="LifeEx")
nodeslabels=list(X="GDP\nper inhabitant",M="Illiteracy Rate",Y="Mean Life\nExpectation")
findName(labels=labels,name="Mi")
findName(labels=labels,nodeslabels=nodeslabels,name="Mi")
labels=list(X="GDPpp",Mi="Illit",Y="LifeEx")
nodeslabels=list(X="GDP\nper inhabitant",Mi="Illiteracy Rate",Y="Mean Life\nExpectation")
findName(labels=labels,name="M")
findName(labels=labels,nodeslabels=nodeslabels,name="M")
labels=list(X="cond",M=c("import","pmi"),Y="reaction")
findName(labels=labels,name="M1")
```

findNames *convert a vector of names with list*

Description

convert a vector of names with list

Usage

```
findNames(labels, nodeslabels = list(), names, exact = FALSE)
```

Arguments

labels	A named list
nodeslabels	A named list
names	A character vector to look for
exact	A logical

Examples

```
labels=list(X="wt",Mi="am",Y="mpg");names=c("X","MiX","Y")
findNames(labels,names=names)
```

---

fit2alpha *Make a Cronbach alpha table*

Description

Make a Cronbach alpha table

Usage

```
fit2alpha(fit, digits = 3)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
digits	integer indicating the number of decimal places to be used.

---

fit2df2 *Make a data.frame for conceptDiagram*

Description

Make a data.frame for conceptDiagram

Usage

```
fit2df2(fit)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan

fit2table *Make estimatesTable with a list of lm object*

Description

Make estimatesTable with a list of lm object

Usage

```
fit2table(fit, labels = labels, digits = 3)
```

Arguments

fit	A list of lm object
labels	A list
digits	integer indicating the number of decimal places

Examples

```
labels=list(X="frame",M="justify",Y="donate",W="skeptic")
moderator=list(name="skeptic",site=list(c("a","c")))
eq=tripleEquation(labels=labels,moderator=moderator,data=disaster,mode=1)
fit=eq2fit(eq,data=disaster)
fit2table(fit=fit,labels=labels)
```

fit2vif *Summarizing correlation, tolerance and variance inflation factors of a model*

Description

Summarizing correlation, tolerance and variance inflation factors of a model

Usage

```
fit2vif(fit, mode = 1, namemode = 1, digits = 3)
```

Arguments

fit	An object of class lm
mode	integer. one of 1:2
namemode	integer. One of 1:3
digits	logical

Examples

```
fit=lm(govact~negemot*age,data=glbwarm)
fit2vif(fit)
```

fun2eq *Make equation from function*

Description

Make equation from function

Usage

```
fun2eq(fun)
```

Arguments

fun	A function

get2ndIndirect *get2ndIndirect effect*

Description

get2ndIndirect effect

Usage

```
get2ndIndirect(X = NULL, M = NULL, Y = NULL, labels = list())
```

Arguments

X	Names of independent variable
M	Names of mediator variable
Y	Names of dependent variable
labels	A list

Examples

```
get2ndIndirect(X="X",M=c("M1","M2","M3"))
```

getArrows *Get arrows data with no*

Description

Get arrows data with no

Usage

```
getArrows(no = 25)
```

Arguments

no	model number

getAspectRatio *Get aspect information of a ggplot*

Description

Get aspect information of a ggplot

Usage

```
getAspectRatio(p)
```

Arguments

p	A ggplot object

getBootData *Get bootstrapped values*

Description

Get bootstrapped values

Usage

```
getBootData(semfit, what = "coef.boot", ...)
```

Arguments

semfit	An object of class lavaan
what	Character. What needs to be inspected/extracted?
...	Further argument to be passed to lavaan::lavTech()

Examples

```
require(lavaan)
labels=list(X="cond",M="pmi",Y="reaction")
model=tripleEquation(labels=labels)

set.seed(1234)
semfit=sem(model,data=pmi,se="boot",bootstrap=100)
getBootData(semfit)
```

---

getCatSlopeDf *Make data summarizing regression slopes and intercepts*

Description

Make data summarizing regression slopes and intercepts

Usage

```
getCatSlopeDf(
  labels = NULL,
  data,
  yvar = "Y",
  total = FALSE,
  addvars = TRUE,
  add.label = FALSE,
  maxylev = 6,
  mode = 1,
  rangemode = 2
)
```

Arguments

labels	Named list of variables
data	A data.frame
yvar	Label of the dependent variable. Either "Y"(default) or "M".
total	logical. If true, model include mediator variable.
addvars	logical. Whether or not add categorical variables to the data
add.label	logical

maxylev	maximal unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding
rangemode	rangemode. 1 or 2.

Examples

```
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
getCatSlopeDf(labels=labels,yvar="M",data=protest,mode=3)
getCatSlopeDf(labels=labels,yvar="M",data=protest,mode=1)
```

---

getCoef *Get coef summary table*

Description

Get coef summary table

Usage

```
getCoef(fit)
```

Arguments

fit	An object of class lm

Examples

```
fit=lm(mpg~hp*wt+am,data=mtcars)
getCoef(fit)
```

---

getEq2p *get slope and intercept with 2 points*

Description

get slope and intercept with 2 points

Usage

```
getEq2p(x1, y1, x2, y2)
```

Arguments

x1	x coordinate of the first point
y1	y coordinate of the first point
x2	x coordinate of the second point
y2	y coordinate of the second point

`getHelmert` *Get Helmert Coding of column j of group with length of unique values (count-1)*

Description

Get Helmert Coding of column j of group with length of unique values (count-1)

Usage

```
getHelmert(x, j, count = NULL)
```

Arguments

`x`	a numeric vector
`j`	column no
`count`	length unique values of group minus 1

Value

A numeric vector

Source

Andrew F. Hayes.(2018) Introduction to Mediation, Moderation and Conditional Process Analysis(2nd Ed.). New York, NY: The Guilford Press. p584

Examples

```
x=c(1:4,4:2,2,3,5)
getHelmert(x,1)
getHelmert(mtcars$cyl,1)
```

`getInfo` *Get information of a model*

Description

Get information of a model

Usage

```
getInfo(fit, digits = 3)
```

Arguments

fit	object of class lm
digits	integer indicating the number of decimal places

Examples

```
fit=lm(mpg~wt*hp,data=mtcars)
getInfo(fit)
```

---

getMeanSd *get mean and sd*

Description

get mean and sd

Usage

```
getMeanSd(data, X, Y, digits)
```

Arguments

data	A data.frame
X	Name of independent variable
Y	Name of dependent variable
digits	Integer indicating the number of decimal places

---

getNodes *Get nodes data with model no*

Description

Get nodes data with model no

Usage

```
getNodes(no = 25)
```

Arguments

no	model number

getRatioTable *Get coding table for dummy variables*

Description

Get coding table for dummy variables

Usage

```
getRatioTable(count = 3, mode = 1)
```

Arguments

count	number of unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding

Examples

```
getRatioTable(count=3)
getRatioTable(count=4,mode=3)
```

getRepValues *Get representative values*

Description

Get representative values

Usage

```
getRepValues(data, colname, rangemode = 2, maxylev = 6, digits = digits)
```

Arguments

data	A data.frame
colname	Name of column
rangemode	Integer. 1 or 2
maxylev	Integer. Maximum unique length of variable to be treated as a categorical variable
digits	integer indicating the number of decimal places

getYhat *Get predicted value from object of class "lm"*

Description

Get predicted value from object of class "lm"

Usage

```
getYhat(fit, group = "D", mode = 1)
```

Arguments

fit	Object of class "lm"
group	names of dummy variables in formula
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding

Examples

```
iris1=addCatVars(iris,c("Species"))
iris3=addCatVars(iris,c("Species"),mode=3)
fit1=lm(Sepal.Length~Sepal.Width+D1+D2,data=iris1)
getYhat(fit1)
fit1=lm(Sepal.Length~D2*Sepal.Width+Sepal.Width*D1+Petal.Width,data=iris1)
getYhat(fit1)
fit3=lm(Sepal.Length~D2*Sepal.Width+Sepal.Width*D1+Petal.Width*D1+Petal.Width*D2,data=iris3)
getYhat(fit3,mode=3)
```

getYhat1 *Get Yhat value from simple mediation*

Description

Get Yhat value from simple mediation

Usage

```
getYhat1(
  data,
  X = NULL,
  M = NULL,
  Y = NULL,
  labels,
  digits = 3,
  maxylev = 6,
  mode = 1
)
```

Arguments

data	A data.frame
X	Name of independent variable
M	Name of moderator variable
Y	Name of dependant variable
labels	optional list of labels
digits	Integer indicating the number of decimal places
maxylev	maximal unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding

Examples

```
data=protest
labels=list(X="protest",M="respappr",Y="liking")
getYhat1(data=protest,labels=labels)
```

---

ggCor *Draw a heatmap of correlation test*

Description

Draw a heatmap of correlation test

Usage

```
ggCor(
  data,
  what = 1,
  label = 0,
  colors = NULL,
  title = TRUE,
  mode = 2,
  digits = 2,
  yreverse = TRUE,
  xangle = 45,
  yangle = 0,
  use.label = FALSE
)
```

Arguments

data	A data.frame
what	if 1, correlation, if 2, partial correlation, if 3, semi-partial correlation
label	if 0, no label(default), if 1, use r value as label, if 2, use r value with significant mark as label
colors	colors for low, mid and high correlation values
title	if true, add title to the heatmap
mode	1 or 2
digits	The number of decimal place'
yreverse	If true, reverse y axis
xangle	x-axis text angle
yangle	y-axis text angle
use.label	Logical whether or not use label in case of labelled data

---

glbwarm *Global Warming dataset*

Description

Global Warming dataset

Usage

```
glbwarm
```

Format

A data.frame with 815 obs. of 7 variables

govact Support for government action

posemot Positive emotions about climate change

negemot Negative emotions about climate change

ideology Political ideology (conservatism), 1 = Very Liberal, 2 = Liberal, 3 = Somewhat Liberal, 4 = Moderate; Middle of the Road, 5 = Somewhat Conservative, 6 = Conservative, 7 = Very Conservative

age Respondent age at last birthday

sex female(0) or male(1)

partyid 1 = Democrat, 2 = Independent, 3= Republican

Source

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

interactStr *make interaction equation*

Description

make interaction equation

Usage

```
interactStr(x, prefix = "a", skip = FALSE, count = 1, addPrefix = TRUE)
```

Arguments

x	character vector
prefix	prefix
skip	whether or not skip
count	Numeric
addPrefix	A logical

Examples

```
interactStr(LETTERS[1])
interactStr(LETTERS[1:3])
interactStr(LETTERS[1:3],skip=TRUE)
```

jnPlot *Draw johnson_neyman plot*

Description

Draw johnson_neyman plot

Usage

```
jnPlot(
  fit,
  pred = NULL,
  modx = NULL,
  digits = 3,
  plot = FALSE,
  mode = 1,
  xvar = "Z",
  addEq = FALSE,
  ...
)
```

Arguments

fit	A regression model
pred	name of predictor variable
modx	name of moderator variable
digits	integer indicating the number of decimal places
plot	logical. Whether or not draw plot
mode	integer 1 or 2
xvar	Name of xvar
addEq	logical
...	Further arguments to be passed to interactions::johnson_neyman()

Examples

```
fit=lm(mpg~hp*wt,data=mtcars)
jnPlot(fit)

fit=lm(justify~frame*skeptic,data=disaster)
res=jnPlot(fit)
res$plot
fit=lm(govact~negemot*sex*age+posemot+ideology,data=glbwarm)
jnPlot(fit,pred="negemot:sex",modx="age",mode=2,addEq=TRUE)
```

---

label2name *Change label into name*

Description

Change label into name

Usage

```
label2name(label, labels)
```

Arguments

label	A string
labels	A named list

Examples

```
label="X:W:Z"
labels=list(X="dep",W="mod",Z="mod2")
label2name(label,labels)
```

`labels2table` *Make table with labels*

Description

Make table with labels

Usage

```
labels2table(
  labels = labels,
  vars = list(),
  moderator = list(),
  covar = NULL,
  serial = TRUE,
  bmatrix = NULL,
  eq = NULL
)
```

Arguments

`labels`	A list
`vars`	A list
`moderator`	A list
`covar`	A list
`serial`	A logical
`bmatrix`	integer specifying causal relations among mediators
`eq`	Optional string contains equation

Examples

```
labels=list(X="frame",M="justify",Y="donate",W="skeptic")
moderator=list(name="skeptic",site=list(c("a","c")))
covar=list(name=c("C1","C2","C3"),site=list(c("M","Y"),c("M","Y"),c("M","Y")))
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
labels=list(X="X",M=c("M1","M2"),Y="Y")
moderator=list();serial=FALSE;eq=NULL
labels2table(labels)
labels2table(labels,serial=FALSE)
labels2table(labels,covar=covar)
labels2table(labels,moderator=moderator)
labels=list(X="X",M="M",Y="Y")
moderator=list(name=c("W"),site=list(c("b","c")))
labels2table(labels,moderator=moderator)
labels=list(X="baby",M="wine",Y="tile")
moderator=list(name=c("milk"),site=list("a"))
covar=list(name=c("milk","tent","sand"),site=list(c("Y"),c("M","Y"),c("M","Y")))
labels2table(labels,moderator=moderator,covar=covar,serial=FALSE)
```

makeAnovaDf *Make data summarizing ANOVA results*

Description

Make data summarizing ANOVA results

Usage

```
makeAnovaDf(
  labels,
  data,
  yvar = "Y",
  total = FALSE,
  addvars = TRUE,
  maxylev = 6,
  mode = 1,
  rangemode = 2
)
```

Arguments

labels	Named list of variables
data	A data.frame
yvar	Label of the dependent variable. Either "Y"(default) or "M".
total	logical. If true, model include mediator variable.
addvars	logical. Whether or not add categorical variables to the data
maxylev	maximal unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding
rangemode	rangemode. 1 or 2.

Examples

```
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
makeAnovaDf(labels=labels,data=protest,total=TRUE,mode=3)
```

makeCatEquation *Make equation for sem and lm for categorical variables*

Description

Make equation for sem and lm for categorical variables

Usage

```
makeCatEquation(
  X = NULL,
  Y = NULL,
  W = NULL,
  labels = list(),
  data,
  prefix = "b",
  maxylev = 6,
  grouplabels = list(),
  mode = 0
)
```

Arguments

X	Name of independent variable
Y	Name of dependent variable
W	Name of moderators
labels	optional list
data	a data.frame
prefix	a character
maxylev	maximal unique length of categorical variable
grouplabels	A list
mode	A numeric

Examples

```
makeCatEquation(X="wt",Y="mpg",data=mtcars)
makeCatEquation(X="wt",Y="mpg",W="cyl",data=mtcars)
makeCatEquation(X="wt",Y="mpg",W=c("cyl","hp"),data=mtcars)
grouplabels=list(carb="f")
makeCatEquation(X="carb",Y="mpg",W=c("cyl","hp"),data=mtcars,maxylev=6)
makeCatEquation(X="carb",Y="mpg",W=c("cyl","hp"),data=mtcars)
cat(makeCatEquation(X="wt",Y="carb",W=c("am","hp"),data=mtcars,maxylev=6,grouplabels=grouplabels))
```

makeCatEquation2 *Make equation for sem and lm for multiple X or multiple Y*

Description

Make equation for sem and lm for multiple X or multiple Y

Usage

```
makeCatEquation2(
  X = NULL,
  Y = NULL,
  W = NULL,
  labels = list(),
  vars = list(),
  prefix = "b",
  mode = 0,
  pos = list(),
  serial = FALSE,
  depy = FALSE,
  depx = FALSE
)
```

Arguments

X	Names of independent variable
Y	Names of dependent variable
W	Names of moderators
labels	optional list
vars	optional list
prefix	a character
mode	A numeric
pos	Numeric moderator position
serial	logical If TRUE, serial variables are added
depy	logical
depx	logical

Examples

```
makeCatEquation2(X="wt",Y="mpg")
makeCatEquation2(X="wt",Y="mpg",W="cyl")
makeCatEquation2(X="wt",Y=c("cyl","am"),prefix="a")
makeCatEquation2(X="wt",Y=c("hp","vs"),W="cyl",prefix="a")
makeCatEquation2(X="wt",Y=c("hp","vs"),W=c("cyl","am"),prefix="a",pos=list(1,2))
```

```
makeCatEquation2(X="wt",Y=c("hp","vs"),W=c("cyl"),prefix="a",pos=list(1))
makeCatEquation2(X="wt",Y=c("hp","vs"),W=c("cyl"),prefix="a",pos=list(c(1,2)))
makeCatEquation2(X=c("hp","vs"),Y="mpg",W=c("cyl"),prefix="b",pos=list(c(1)))
makeCatEquation2(X=c("hp","vs"),Y="mpg",W=c("cyl"),prefix="b")
makeCatEquation2(X=c("hp","vs"),Y="mpg",W=c("cyl"),prefix="b",pos=list(c(1,2)))
cat(makeCatEquation2(X="wt",Y="carb",W=c("am","hp")))
cat(makeCatEquation2(X="X",Y=c("M1","M2","M3"),W=NULL,prefix="a",serial=TRUE))
cat(makeCatEquation2(X="X",Y=c("M1","M2","M3"),W=NULL,prefix="a"))
cat(makeCatEquation2(X="X",Y=c("M1","M2"),prefix="a",mode=1,serial=TRUE))
```

`makeCatEquation3` *Make equation for sem and lm for multiple X or multiple Y*

Description

Make equation for sem and lm for multiple X or multiple Y

Usage

```
makeCatEquation3(
  X = NULL,
  Y = NULL,
  W = NULL,
  labels = list(),
  prefix = "b",
  mode = 0,
  pos = list(),
  bmatrix = NULL,
  vars = list(),
  moderator = list(),
  depy = FALSE,
  depx = FALSE,
  interactionNo = 0
)
```

Arguments

`X`	Names of independent variable
`Y`	Names of dependent variable
`W`	Names of moderators
`labels`	optional list
`prefix`	a character
`mode`	A numeric
`pos`	Numeric moderator position
`bmatrix`	integer specifying causal relations among mediators

vars	A list of triple moderator
moderator	A list
depy	logical
depx	logical
interactionNo	numeric

Examples

```
cat(makeCatEquation3(X="X",Y=c("M1","M2","M3"),prefix="a",bmatrix=c(1,1,0,1,0,0,1,1,1,1)))
cat(makeCatEquation3(X="X",Y=c("M1","M2","M3"),prefix="a",bmatrix=c(1,1,0,1,0,1,1,1,1,1)))
cat(makeCatEquation3(X="X",Y=c("M1","M2","M3"),prefix="a",bmatrix=c(1,1,0,1,1,0,1,1,1,1)))
cat(makeCatEquation3(X="X",Y=c("M1","M2","M3"),prefix="a",bmatrix=c(1,1,1,1,1,1,1,1,1,1)))
cat(makeCatEquation3(X=c("M1","M2","M3"),Y="Y",prefix="a",bmatrix=c(1,1,1,1,1,1,1,1,1,1),depy=TRUE))
cat(makeCatEquation3(X="X",Y="Y",prefix="a",bmatrix=c(1,1,1,1,1,1,1,1,1,1),depy=TRUE,depx=TRUE))
cat(makeCatEquation3(X="X",Y="Y",prefix="a",bmatrix=c(1,1,1,1,1,1,0,1,1,1),depy=TRUE,depx=TRUE))
cat(makeCatEquation3(X=c("M1","M2"),Y="Y",prefix="a",bmatrix=c(1,1,1,1,0,1),depy=TRUE))
cat(makeCatEquation3(X=c("M1","M2"),Y="Y",prefix="a",bmatrix=c(1,1,1,1,1,0),depy=TRUE))
cat(makeCatEquation3(X="X",Y=c("M1","M2"),prefix="a",bmatrix=c(1,1,1,0,0,1),depy=FALSE))
cat(makeCatEquation3(X="X",Y=c("M1","M2"),W="W",prefix="a",bmatrix=c(1,1,1,1,1,1),depy=FALSE,
  moderator=list(name="W",matrix=list(c(0,0,1,0,0,0)))))
cat(makeCatEquation3(X=c("M1","M2"),Y="Y",prefix="a",bmatrix=c(1,1,1,1,0,1),depy=TRUE))
cat(makeCatEquation3(X=c("M1","M2"),Y="Y",W="W",pos=list(c(1,2)),prefix="a",
  bmatrix=c(1,1,1,1,0,1),depy=TRUE))
cat(makeCatEquation3(X=c("M1","M2"),Y="Y",W="W",
 moderator=list(name="W",matrix=list(c(0,0,0,0,1,1))),bmatrix=c(1,1,1,1,1,1),depy=TRUE))
vars=list(name=list(c("W","Z")),matrix=list(c(0,0,1,0,0,0)))
cat(makeCatEquation3(X="X",Y=c("M1","M2"),bmatrix=c(1,1,1,1,1,0),vars=vars,depy=FALSE,depx=TRUE))
```

`makeCatModel` *Make simple regression model with one categorical variable*

Description

Make simple regression model with one categorical variable

Usage

```
makeCatModel(
  labels = labels,
  data,
  yvar = "Y",
  total = FALSE,
  addvars = TRUE,
  maxylev = 6,
  mode = 1
)
```

Arguments

labels	Named list of variables
data	A data.frame
yvar	Label of the dependent variable. Either "Y"(default) or "M".
total	logical. If true, model include mediator variable.
addvars	logical. Whether or not add categorical variables to the data
maxylev	maximal unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding

Value

An object of class lm

Examples

```
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
data1=addCatVars(protest,"protest")
makeCatModel(labels=labels,data=data1)
```

---

makeCEDf *Make data summarizing conditional effects*

Description

Make data summarizing conditional effects

Usage

```
makeCEDf(
  labels = labels,
  data,
  yvar = "Y",
  total = FALSE,
  addvars = TRUE,
  maxylev = 6,
  mode = 1,
  rangemode = 2
)
```

Arguments

labels	Named list of variables
data	A data.frame
yvar	Label of the dependent variable. Either "Y"(default) or "M".
total	logical. If true, model include mediator variable.
addvars	logical. Whether or not add categorical variables to the data
maxylev	maximal unique length of categorical variable
mode	Numeric. One of 1:4. 1= simple indicator coding, 2= sequential coding, 3= Helmert coding, 4= effect coding
rangemode	rangemode. 1 or 2.

Examples

```
labels=list(X="protest",W="sexism",M="respappr",Y="liking")
data1=addCatVars(protest,varnames="protest",mode=1)
makeCEDf(labels=labels,data=protest,mode=1)
```

---

makeCoefLabel *Change regression coefficient name*

Description

Change regression coefficient name

Usage

```
makeCoefLabel(name, dep, labels, constant, prefix)
```

Arguments

name	string vector to change
dep	names of dependent variable
labels	optional list
constant	name of constant
prefix	name of prefix

makeEquation *Make mediation equations 3*

Description

Make mediation equations 3

Usage

```
makeEquation(X, M, Y, add2ndMediation = TRUE, covar = list())
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
Y	A character vectors indicating dependent variables
add2ndMediation	whether or not make a secondmediation equation
covar	Optional list of covariates

Examples

```
X="X";M=c("M1","M2","M3");Y=c("Y1","Y2");add2ndMediation=TRUE
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M1","Y1"),"Y2","Y2"))
cat(makeEquation(X,M,Y,covar=covar))
```

makeEquation1 *Make mediation equations 1*

Description

Make mediation equations 1

Usage

```
makeEquation1(X, M, stage = 1, start = 0, add2ndMediation = TRUE)
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
stage	An integer indicating the order
start	An integer
add2ndMediation	whether or not make a secondmediation equation

makeEquation2 *Make mediation equations 2*

Description

Make mediation equations 2

Usage

```
makeEquation2(X, M, Y)
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
Y	A character vectors indicating dependent variables

makeEquation3 *Make mediation equations 3*

Description

Make mediation equations 3

Usage

```
makeEquation3(X, M, Y, add2ndMediation = TRUE)
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
Y	A character vectors indicating dependent variables
add2ndMediation	whether or not make a secondmediation equation

`makeIndirectEquation` *Make indirect equation*

Description

Make indirect equation

Usage

```
makeIndirectEquation(
  X,
  M,
  temp1,
  temp2,
  temp3,
  moderatorNames,
  range = FALSE,
  data = NULL,
  rangemode = 1,
  probs = c(0.16, 0.5, 0.84),
  effectsize = FALSE,
  Y = NULL
)
```

Arguments

`X`	A character string
`M`	A character string
`temp1`	A character vector
`temp2`	A character vector
`temp3`	A character vector
`moderatorNames`	A character vector
`range`	A logical
`data`	A data.frame
`rangemode`	range mode
`probs`	numeric vector of probabilities with values in [0,1]
`effectsize`	logical If true, calculate effect size.
`Y`	Optional character string

Examples

```
X="negemot";M="ideology"
temp1=c("a1*negemot","a2*sex","a4*negemot*sex","a5*negemot*age","a6*sex*age")
temp2="b1*ideology"
temp3="c1*negemot"
moderatorNames=c("age","sex")
X= "hp";M= "am"
temp1= c("a1*hp","a2*wt","a3*hp:wt")
temp2= "b1*am"
temp3= c("c1*hp","c2*wt","c3*hp:wt")
#moderatorNames=c("wt")
#X= c("d1","d2");M="am"
#temp1=c("a1*d1","a2*d2","a3*wt","a4*d1:wt","a5*d2:wt")
#temp2="b1*am"
#temp3=c("c1*d1","c2*d2","c3*wt","c4*d1:wt","c5*d2:wt")
cat(makeIndirectEquation(X,M,temp1,temp2,temp3,moderatorNames))
cat(makeIndirectEquation(X,M,temp1,temp2,temp3,moderatorNames,range=TRUE))
X="wt";M=NULL;temp1=NULL;temp2=NULL;temp3=c("c1*wt","c2*hp","c3*wt:hp");
moderatorNames="hp";range=TRUE;rangemode=1;probs=c(0.16,0.5,0.84)
cat(makeIndirectEquation(X,M,temp1,temp2,temp3,moderatorNames,range=TRUE))
X="frame:skeptic"; M="justify";temp1="a1*frame:skeptic";
temp2="b1*justify";temp3="c1*frame:skeptic";moderatorNames=NULL
range=TRUE;rangemode=1
```

---

makeIndirectEquationCat

Make indirect equation for categorical variables

Description

Make indirect equation for categorical variables

Usage

```
makeIndirectEquationCat(
  X,
  M,
  temp1,
  temp2,
  temp3,
  moderatorNames,
  range = TRUE,
  data = NULL,
  rangemode = 1,
  probs = c(0.16, 0.5, 0.84),
  grouplabels = list()
)
```

Arguments

X	A character vector
M	A character vector
temp1	A character vector
temp2	A character vector
temp3	A character vector
moderatorNames	A character vector
range	A logical
data	A data.frame
rangemode	range mode
probs	numeric vector of probabilities with values in [0,1]
grouplabels	A list

---

makeIndirectEquationCat2

Make indirect equation for categorical variables

Description

Make indirect equation for categorical variables

Usage

```
makeIndirectEquationCat2(
  X,
  M,
  temp1,
  temp2,
  temp3,
  moderatorNames,
  range = TRUE,
  data = NULL,
  rangemode = 1,
  probs = c(0.16, 0.5, 0.84),
  serial = FALSE,
  contrast = 1
)
```

Arguments

X	A character vector
M	A character vector
temp1	A character vector
temp2	A character vector
temp3	A character vector
moderatorNames	A character vector
range	A logical
data	A data.frame
rangemode	range mode
probs	numeric vector of probabilities with values in [0,1]
serial	logical If TRUE, serial variables are added
contrast	integer If 2, absolute difference of contrasts are calculated

---

makeLabel *Make Labels*

Description

Make Labels

Usage

```
makeLabel(
  fit,
  pred,
  modx,
  mod2,
  mod2.values = c(30, 50, 70),
  xvar = "Z",
  digits = 3
)
```

Arguments

fit	An object of class lm
pred	name of the predictor variable
modx	name of the moderator variable
mod2	name of the second moderator variable
mod2.values	values of moderator variable
xvar	character. "Z" or "W"
digits	integer indicating the number of decimal places

makePPTx *make powerpoint presentation*

Description

make powerpoint presentation

Usage

```
makePPTx(
  data,
  preprocessing = "",
  filename = "report.pptx",
  rawDataName = NULL,
  rawDataFile = "rawData.RDS",
  rmdRemove = TRUE
)
```

Arguments

data	A data.frame with title and code
preprocessing	string preprocessing
filename	character name of output file
rawDataName	The name of the rawData
rawDataFile	The name of the rawData file which the data are to be read from
rmdRemove	A logical

matrix2df *Make data.frame with matrix*

Description

Make data.frame with matrix

Usage

```
matrix2df(matrix = c(1, 1, 1, 0, 0, 1), labels = NULL)
```

Arguments

matrix	a numeric vector
labels	Optional list of labels

Examples

```
labels=list(X="indep",M=c("med1","med2"),Y="dep")
matrix2df(c(1,1,1,0,0,1),labels=labels)
```

matrix2no *Calculate the dimension of matrix*

Description

Calculate the dimension of matrix

Usage

```
matrix2no(matrix = c(1, 1, 1, 0, 0, 1))
```

Arguments

matrix	a numeric vector

Examples

```
matrix2no(c(1,1,1,0,0,1))
```

matrixPlot *Draw matrix plot*

Description

Draw matrix plot

Usage

```
matrixPlot(
  matrix = c(1, 1, 1, 0, 0, 1),
  radx = 0.1,
  rady = 0.05,
  xlim = NULL,
  ylim = NULL,
  labels = NULL
)
```

Arguments

matrix	A numeric vector
radx	horizontal radius of the box.
rady	vertical radius of the box.
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot
labels	Optional list of labels

Examples

```
matrixPlot(c(1,1,1))
labels=list(X="X",M=c("M1","M2"),Y="Y")
bmatrix=c(1,1,1,0,0,1)
eq=multipleMediation(labels=labels,bmatrix=bmatrix,mode=1)
drawModel(equation=eq,labels=labels,nodemode=2)
matrixPlot(bmatrix)
bmatrix=c(1,1,0,1,0,0,1,1,1,1)
matrixPlot(c(1,1,0,1,0,0,1,1,1,1))
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
eq=multipleMediation(labels=labels,bmatrix=bmatrix,mode=1)
drawModel(equation=eq,labels=labels,parallel=TRUE,nodemode=2)
labels=list(X="indep",M=c("med1","med2"),Y="dep")
matrixPlot(c(1,1,1,0,0,1),labels=labels)
```

---

meanCentering *Perform mean centering*

Description

Perform mean centering

Usage

```
meanCentering(data, names)
```

Arguments

data	A data.frame
names	column names to mean centering

Examples

```
library(processR)
newData=meanCentering(education,colnames(education)[1:3])
```

---

meanSummary *Make mean summary table*

Description

Make mean summary table

Usage

```
meanSummary(
  data,
  X = NULL,
  Y = NULL,
  M = NULL,
  W = NULL,
  labels = labels,
  digits = 3,
  xlabels = NULL,
  maxylev = 6,
  mode = 1
)
```

Arguments

data	A data.frame
X	Name of independent variable
Y	Name of dependent variable
M	Name of mediator variable
W	Name of moderator variable
labels	A list of labels
digits	Integer indicating the number of decimal places
xlabels	Optional string vector
maxylev	maximal unique length of categorical variable
mode	integer

Examples

```
labels=list(X="cond",Y="reaction",M="pmi")
xlabels=c("Front Page","Interior Page")
meanSummary(data=pmi,labels=labels,xlabels=xlabels)
labels=list(X="frame",Y="justify",W="skeptic")
xlabels=c("Natural causes condition","Climate change condition")
meanSummary(data=disaster,labels=labels,xlabels=xlabels)
labels=list(X="protest",Y="liking",M="respappr")
meanSummary(data=protest,labels=labels)
meanSummary(data=protest,labels=labels,maxylev=2)
```

meanSummaryTable *Make mean summary table*

Description

Make mean summary table

Usage

```
meanSummaryTable(..., vanilla = TRUE)
```

Arguments

...	Further arguments to be passed to meanSummary
vanilla	logical

Examples

```
labels=list(X="cond",Y="reaction",M="pmi")
xlabels=c("Front Page","Interior Page")
meanSummaryTable(data=pmi,labels=labels,xlabels=xlabels)
labels=list(X="frame",Y="justify",W="skeptic")
xlabels=c("Natural causes condition","Climate change condition")
meanSummaryTable(data=disaster,labels=labels,xlabels=xlabels)
```

mediationBK *Perform mediation analysis by Baron and Kenny Method*

Description

Perform mediation analysis by Baron and Kenny Method

Usage

```
mediationBK(
  X = NULL,
  M = NULL,
  Y = NULL,
  labels = list(),
  data,
  silent = TRUE,
  indirect.test = TRUE,
  sig = 0.05
)
```

Arguments

X	name of independent variable
M	name of mediator variable
Y	name of dependent variable
labels	An optional list of variable names
data	A data.frame
silent	Logical. Whether or not show summary of regression tests
indirect.test	Logical. Whether or not show results of bda.mediation.test
sig	significant level. default value is 0.05

Examples

```
labels=list(X="cond",M="pmi",Y="reaction")
result=mediationBK(labels=labels,data=pmi,silent=FALSE)
result
```

---

medSummary *Summarize the mediation effects*

Description

Summarize the mediation effects

Usage

```
medSummary(semfit, boot.ci.type = "perc", effects = c("indirect", "direct"))
```

Arguments

semfit	An object of class lavaan
boot.ci.type	Type of bootstrapping interval. Choices are c("norm","basic","perc",bca.simple","all")
effects	Names of effects to be summarized

Value

A data.frame and an object of class medSummary

Examples

```
library(lavaan)
labels=list(X="cond",M="pmi",Y="reaction")
model=tripleEquation(labels=labels)

semfit=sem(model=model,data=pmi, se="boot", bootstrap=100)
medSummary(semfit)
medSummary(semfit,boot.ci.type="all")
```

medSummaryTable *Make a table summarizing the mediation effects*

Description

Make a table summarizing the mediation effects

Usage

```
medSummaryTable(x, vanilla = TRUE, ...)
```

Arguments

x	An object of class medSummary or medSummary2 or lavaan
vanilla	A logical
...	Further arguments to be passed to medSummary

medSummaryTable1 *Make a table summarizing the mediation effects*

Description

Make a table summarizing the mediation effects

Usage

```
medSummaryTable1(x, vanilla = TRUE, showP = FALSE)
```

Arguments

x	An object of class medSummary
vanilla	A logical
showP	A logical

medSummaryTable2 *Make a table summarizing the mediation effects*

Description

Make a table summarizing the mediation effects

Usage

```
medSummaryTable2(x, vanilla = TRUE)
```

Arguments

x	An object of class medSummary2
vanilla	A logical

modelFitGuideTable *Model fit guide table*

Description

Model fit guide table

Usage

```
modelFitGuideTable()
```

modelFitGuideTable2 *Model fit guide table*

Description

Model fit guide table

Usage

```
modelFitGuideTable2(vanilla = FALSE)
```

Arguments

vanilla	Logical

modelFitTable *Extract model fit measures to data.frame*

Description

Extract model fit measures to data.frame

Usage

```
modelFitTable(fit, digits = 2, names = NULL)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
digits	integer indicating the number of decimal places to be used.
names	names of statistic to be extracted

Value

A data.frame

modelFitTable2 *Extract model fit measures to flextable*

Description

Extract model fit measures to flextable

Usage

```
modelFitTable2(fit, vanilla = FALSE, ...)
```

Arguments

fit	An object of class lavaan. Result of sem function of package lavaan
vanilla	Logical
...	Further arguments to be passed to modelFitTable()

`modelsSummary` *Make Summary for Model Coefficients*

Description

Make Summary for Model Coefficients

Usage

```
modelsSummary(
  fit = NULL,
  labels = NULL,
  vars = NULL,
  moderator = NULL,
  covar = NULL,
  serial = FALSE,
  data = NULL,
  prefix = "b",
  constant = "iy",
  autoPrefix = TRUE
)
```

Arguments

`fit`	A list of objects of class lm
`labels`	optional list
`vars`	optional list
`moderator`	optional list
`covar`	optional list
`serial`	logical
`data`	optional data.frame
`prefix`	A character
`constant`	A string vector
`autoPrefix`	logical automatic numbering of prefix

Value

A data.frame

Examples

```
fit1=lm(mpg~wt,data=mtcars)
fit2=lm(mpg~wt*hp*am,data=mtcars)
fit=list(fit1,fit2)
labels=list(Y="mpg",X="wt",W="hp",Z="am")
modelsSummary(fit,labels=labels)
labels=list(Y="withdraw",M="affect",X="estress")
covar=list(name=c("ese","sex","age"),site=list(c("M","Y"),c("M","Y"),c("M","Y")))
modelsSummary(labels=labels,covar=covar,data=estress)
labels=list(X="dysfunc",M="negtone",W="negexp",Y="perform")
moderator=list(name="negexp",site=list(c("a","b","c")))
eq=tripleEquation(labels=labels,moderator=moderator,data=teams,mode=1)
fit=eq2fit(eq,data=teams)
modelsSummary(fit,labels=labels)
labels=list(X="cond",M="pmi",Y="reaction")
modelsSummary(labels=labels,data=pmi)
```

`modelsSummary2` *Make Summary for Model Coefficients*

Description

Make Summary for Model Coefficients

Usage

```
modelsSummary2(
  fit,
  labels = NULL,
  prefix = "b",
  constant = "iy",
  fitlabels = NULL,
  autoPrefix = TRUE
)
```

Arguments

`fit`	A list of objects of class lm
`labels`	optional list
`prefix`	A character
`constant`	A string vector
`fitlabels`	Optional. labels of models
`autoPrefix`	logical

Value

A data.frame

Examples

```
fit1=lm(mpg~wt,data=mtcars)
fit2=lm(mpg~wt*hp*vs+am,data=mtcars)
labels=list(Y="mpg",X="wt",W="hp",Z="vs")
fit=list(fit1,fit2)
modelsSummary2(fit,labels=labels)
modelsSummary2(fit,labels=labels,prefix=c("c","b"),autoPrefix=FALSE)
modelsSummary2(fit1)
```

---

modelsSummary2Table *Make Summary Table 2 for Model Coefficients*

Description

Make Summary Table 2 for Model Coefficients

Usage

```
modelsSummary2Table(x, vanilla = TRUE, mode = 1, ...)
```

Arguments

x	An object of class modelSummary2
vanilla	A logical
mode	An Integer One of 1:2
...	further arguments to be passed to modelsSummary2()

---

modelsSummaryTable *Make Summary Table for Model Coefficients*

Description

Make Summary Table for Model Coefficients

Usage

```
modelsSummaryTable(x = NULL, vanilla = TRUE, ...)
```

Arguments

x	An object of class modelSummary
vanilla	A logical
...	further arguments to be passed to modelsSummary()

Value

A flextable

Examples

```
fit1=lm(mpg~wt,data=mtcars)
fit2=lm(mpg~wt*hp,data=mtcars)
fit3=lm(mpg~wt*hp*am,data=mtcars)
labels=list(X="wt",W="hp",Y="mpg",Z="am")
x=modelsSummary(fit1,labels=labels)
modelsSummaryTable(x)
modelsSummaryTable(x,vanilla=FALSE)
x=modelsSummary(list(fit1,fit2),labels=labels)
modelsSummaryTable(x)
modelsSummaryTable(x,vanilla=FALSE)
x=modelsSummary(list(fit1,fit2,fit3),labels=labels)
modelsSummaryTable(x)
modelsSummaryTable(x,vanilla=FALSE)
```

---

moderator2df *Make data.frame from a list of moderator*

Description

Make data.frame from a list of moderator

Usage

```
moderator2df(moderator, mpos = c(0.5, 0.9), vars = NULL, df = NULL)
```

Arguments

moderator	A list
mpos	A numeric vector of length 2
vars	A list
df	A data.frame

Examples

```
moderator=list(name=c("milk","hair"),matrix=list(c(1,1,0,1,0,0,0,0,0,0)
   ,c(0,0,0,0,0,0,0,1,0,0)))
moderator2df(moderator)
```

moderator2pos *get position from moderator*

Description

get position from moderator

Usage

```
moderator2pos(moderator = list(), x, y, m)
```

Arguments

moderator	A list
x	position of x
y	position of y
m	position of m

modmedEquation *Make moderated mediation equation*

Description

Make moderated mediation equation

Usage

```
modmedEquation(
  X = "",
  M = NULL,
  Y = "",
  moderator = list(),
  labels = NULL,
  range = FALSE,
  covar = list()
)
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
Y	A character vectors indicating dependent variables
moderator	moderator
labels	labels
range	Whether or not add range equation
covar	Optional list of covariates

Examples

```
X="X";Y="Y"
moderator=list(name=c("Z"),site=list(c("a","c")))
cat(modmedEquation(X=X,Y=Y,moderator=moderator,range=TRUE))
X="X";M="M";Y="Y"
cat(modmedEquation(X=X,M=M,Y=Y,range=TRUE))
X="X";M="M";Y="Y"
moderator=list(name=c("Z"),site=list(c("a","c")))
cat(modmedEquation(X=X,M=M,Y=Y,moderator=moderator,range=TRUE))
X="X";M="M";Y="Y";labels=NULL;range=FALSE
moderator=list(name=c("X"),site=list(c("b")))
cat(modmedEquation(X=X,M=M,Y=Y,moderator=moderator,range=FALSE))
X="X";Y="Y"
moderator=list(name=c("Z"),site=list(c("c")))
cat(modmedEquation(X=X,Y=Y,moderator=moderator,range=FALSE))
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M","Y"),"Y","Y"))
cat(modmedEquation(X=X,M="M",Y=Y,moderator=moderator,range=FALSE,covar=covar))
```

`modmedSummary` *Summarize the moderated mediation*

Description

Summarize the moderated mediation

Usage

```
modmedSummary(
  semfit,
  mod = NULL,
  values = NULL,
  boot.ci.type = "perc",
  add.range = TRUE
)
```

Arguments

`semfit`	An object of class lavaan
`mod`	name of moderator
`values`	Optional. Numeric vector
`boot.ci.type`	Type of bootstrapping interval. Choices are c("norm","basic","perc",bca.simple")
`add.range`	logical Whether or not add range

Value

A data.frame and an object of class modmedSummary

Examples

```
require(lavaan)
labels=list(X="frame",M="justify",Y="donate",W="skeptic")
moderator=list(name="skeptic",site=list(c("a","c")))
model=tripleEquation(labels=labels,moderator=moderator)
cat(model)

semfit=sem(model,data=disaster,se="boot",bootstrap=100)
modmedSummary(semfit)
conditionalEffectPlot(semfit,data=disaster)
labels=list(X="dysfunc",M="negtone",Y="perform",W="negexp")
moderator=list(name="negexp",site=list("b"))
model=tripleEquation(labels=labels,moderator=moderator,data=teams,rangemode=2)
cat(model)
semfit=sem(model,data=teams,se="boot",bootstrap=100)
summary(semfit)
modmedSummary(semfit)
conditionalEffectPlot(semfit,data=teams)
```

---

modmedSummary2Table *make table summarizing moderated mediation effect*

Description

make table summarizing moderated mediation effect

Usage

```
modmedSummary2Table(x, vanilla = TRUE, ...)
```

Arguments

x	An object of class lavaan or modmedSummary2
vanilla	logical.
...	Further arguments to be passed to modmedSummary

---

modmedSummaryTable *Make a table summarizing the moderated mediation*

Description

Make a table summarizing the moderated mediation

Usage

```
modmedSummaryTable(x, vanilla = TRUE, showP = FALSE, ...)
```

Arguments

x	An object of class modmedSummary or class lavaan
vanilla	A logical
showP	logical
...	Further arguments to be passed to modmedSummary

---

modSummary *Make moderation effect summary*

Description

Make moderation effect summary

Usage

```
modSummary(semfit, mod = NULL, values = NULL, boot.ci.type = "bca.simple")
```

Arguments

semfit	An object of class lavaan
mod	name of moderator variable
values	optional values of moderator
boot.ci.type	Type of bootstrapping interval. Choices are c("norm","basic","perc",bca.simple")

Examples

```
require(lavaan)
labels=list(X="frame",W="skeptic",M="justify",Y="donate")
moderator=list(name='skeptic',site=list(c("a")))
model=tripleEquation(labels=labels,moderator=moderator,data=disaster,rangemode=2)
cat(model)
## Not run:
semfit=sem(model=model,data=disaster,se="boot",bootstrap=100)
modSummary(semfit)
modSummaryTable(semfit)
labels=list(X="dysfunc",M="negtone",Y="perform",W="negexp")
moderator=list(name="negexp",site=list("b"))
model=tripleEquation(labels=labels,moderator=moderator,data=teams,rangemode=2)
cat(model)
semfit=sem(model,data=teams,se="boot",bootstrap=100)
modmedSummary(semfit)
modSummaryTable(semfit)

## End(Not run)
```

modSummary2 *Make table summarizing moderation effect*

Description

Make table summarizing moderation effect

Usage

```
modSummary2(
  fit,
  rangemode = 2,
  pred.values = NULL,
  summarymode = 2,
  maxylev = 6,
  digits = 3,
  labels = NULL,
  ...
)
```

Arguments

fit	An object of class lm
rangemode	An integer. If 1, mean+c(-1,0,1)*sd used. If 2, 16th, 50th and 84th percentiles are used
pred.values	Values of predictor variables
summarymode	An integer. 1 or 2. Summarizing method of variables. If 1, typical values are used. If 2, mean values are used
maxylev	An integer. Maximum length of predictor variables to be treated as a categorical variable.
digits	An integer indicating the number of decimal places
labels	Optional list of labels of variables
...	Further arguments to be passed to predict3d::fit2newdata()

Examples

```
labels=list(X="negemot",W="sex",Z="age",Y="govact",C1="posemot",C2="ideology")
fit=lm(govact~negemot*sex+negemot*age+posemot+ideology,data=glbwarm)
modSummary2(fit,rangemode=2,mod2.values=c(30,50,70),summarymode=2)
modSummary2(fit,mod2.values=c(30,50,70),summarymode=1,labels=labels)
labels=list(X="frame",W="skeptic",Y="justify")
fit=lm(justify~frame*skeptic,data=disaster)
modSummary2(fit,labels=labels)
```

modSummary2Table *Make flextable summarizing moderation effect*

Description

Make flextable summarizing moderation effect

Usage

```
modSummary2Table(x, vanilla = TRUE, ...)
```

Arguments

x	An object
vanilla	logical
...	Further argument to be passed to modSummary3

Examples

```
fit=lm(govact~negemot*sex+negemot*age+posemot+ideology,data=glbwarm)
modSummary2Table(fit)
```

modSummary3 *Summary of moderation effect*

Description

Summary of moderation effect

Usage

```
modSummary3(
  fit,
  X = NULL,
  W = NULL,
  Z = NULL,
  labels = NULL,
  modx.values = NULL,
  mod2.values = NULL,
  rangemode = 2,
  maxylev = 6,
  digits = 3
)
```

Arguments

fit	An object of class lm
X	Name of independent variable
W	Name of the first moderator variable
Z	Name of the second moderator variable
labels	Optional list of variable names
modx.values	Values of moderator variable
mod2.values	Values of the second moderator variable
rangemode	Integer. 1 or 2.
maxylev	maximum unique length of variable to be treated as a categorical variable
digits	integer indicating the number of decimal places

Examples

```
fit=lm(govact~negemot*sex+negemot*age+posemot+ideology,data=glbwarm)
modSummary3(fit,mod2.values=c(30,50,70))
fit1=lm(govact~negemot*sex*age+posemot+ideology,data=glbwarm)
modSummary3(fit1,rangemode=1)
fit=lm(mpg~hp*wt,data=mtcars)
modSummary3(fit)
```

---

modSummary3Table *Make flextable summarizing moderation effect*

Description

Make flextable summarizing moderation effect

Usage

```
modSummary3Table(x, vanilla = TRUE, ...)
```

Arguments

x	An object
vanilla	logical
...	Further argument to be passed to modSummary3

Examples

```
fit=lm(govact~negemot*sex+negemot*age+posemot+ideology,data=glbwarm)
modSummary3Table(fit,mod2.values=c(30,50,70))
```

modSummaryTable *Make flextable summarizing moderation effect*

Description

Make flextable summarizing moderation effect

Usage

```
modSummaryTable(x, vanilla = TRUE, ...)
```

Arguments

x	An object
vanilla	logical
...	Further argument to be passed to modSummary

moreModels *more models data*

Description

more models data

Usage

```
moreModels
```

Format

A data.frame 2 variables

no process macro model number

no1 model number

`multipleMediation` *Make Mediation Equation with multiple X or multiple Y*

Description

Make Mediation Equation with multiple X or multiple Y

Usage

```
multipleMediation(
  X = NULL,
  M = NULL,
  Y = NULL,
  labels = list(),
  data = NULL,
  vars = list(),
  moderator = list(),
  covar = NULL,
  mode = 0,
  range = TRUE,
  rangemode = 1,
  serial = FALSE,
  contrast = 1,
  bmatrix = NULL
)
```

Arguments

`X`	Names of independent variable
`M`	Names of mediator variable
`Y`	Names of dependent variable
`labels`	optional list
`data`	A data.frame
`vars`	A list
`moderator`	A list
`covar`	A list of covariates
`mode`	A numeric. 0: SEM equation, 1: regression equation
`range`	A logical
`rangemode`	range mode
`serial`	logical If TRUE, serial variables are added
`contrast`	integer If 2, absolute difference of contrasts are calculated
`bmatrix`	integer specifying causal relations among mediators

Examples

```
labels=list(X="cyl",M="am",Y="mpg")
covar=list(name=c("carb","disp"),site=list(c("M","Y"),"Y","Y"))
cat(multipleMediation(labels=labels,covar=covar,data=mtcars))
labels=list(X=c("cyl","wt"),M="am",Y="mpg")
moderator=list(name=c("vs"),site=list(c("a1","b1")))
cat(multipleMediation(labels=labels,data=mtcars))
cat(multipleMediation(labels=labels,moderator=moderator,data=mtcars))
labels=list(X="wt",M=c("cyl","am"),Y="mpg")
moderator=list(name=c("vs"),site=list(c("b1","b2")))
cat(multipleMediation(labels=labels,data=mtcars,range=FALSE))
cat(multipleMediation(labels=labels,moderator=moderator,data=mtcars,range=FALSE))
eq=multipleMediation(labels=labels,moderator=moderator,data=mtcars,range=FALSE,serial=FALSE,mode=1)
drawModel(equation=eq,labels=labels)
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
labels=list(X="X",M=c("M1","M2"),Y="Y")
cat(multipleMediation(labels=labels))
cat(multipleMediation(labels=labels,serial=TRUE))
moderator=list(name=c("W"),site=list(c("a1","b1")))
cat(multipleMediation(labels=labels,moderator=moderator,range=FALSE))
cat(multipleMediation(labels=labels,moderator=moderator,data=mtcars,range=FALSE))
cat(multipleMediation(X="am",Y="mpg",data=mtcars,moderator=moderator,covar=covar))
labels=list(X="cond",M=c("import","pmi"),Y="reaction")
cat(multipleMediation(labels=labels,data=pmi,serial=TRUE))
cat(multipleMediation(labels=labels,data=pmi,contrast=2))
cat(multipleMediation(labels=labels,data=pmi,mode=1,serial=TRUE))
labels=list(X="X",M=c("M1","M2","M3"),Y="Y")
cat(multipleMediation(labels=labels,bmatrix=c(1,1,1,1,1,1,1,1,1,1)))
labels=list(X="X",M=c("M1","M2"),Y="Y",W="W")
cat(multipleMediation(labels=labels,bmatrix=c(1,1,1,1,1,0)))
cat(multipleMediation(labels=labels,bmatrix=c(1,1,1,1,0,0)))
moderator=list(name=c("W"),matrix=list(c(1,1,0,1,0,0)))
eq=multipleMediation(labels=labels,moderator=moderator,bmatrix=c(1,1,1,1,1,1),mode=1)
drawModel(equation=eq,labels=labels,nodemode=2)
labels=list(X="X",M=c("M1","M2","M3"),Y="Y",W="W")
cat(multipleMediation(labels=labels,bmatrix=c(1,1,0,0,1,1,1,1,0,1)))
labels=list(X="X",M=c("M1","M2"),Y="Y")
cat(multipleMediation(labels=labels,serial=TRUE,mode=1))
vars=list(name=list(c("W","Z")),matrix=list(c(0,0,1,0,0,0)))
cat(multipleMediation(labels=labels,bmatrix=c(1,1,1,1,1,0),vars=vars))
```

`myarrow` *Draw arrow*

Description

Draw arrow

Usage

```
myarrow(
  from,
  to,
  lwd = 1,
  adjust = 1,
  label = "",
  label.pos = 0.5,
  arr.pos = NULL,
  radx = 0.1,
  rady = 0.06,
  xadj = NULL,
  yadj = NULL,
  curve = 0,
  dd = 0,
  ...
)
```

Arguments

from	coordinates (x,y) of the point *from* which to draw arrow.
to	coordinates (x,y) of the point *to* which to draw arrow.
lwd	line width
adjust	adjust position
label	label
label.pos	label position
arr.pos	arrow position
radx	horizontal radius of the box.
rady	vertical radius of the box.
xadj	numeric x adjustment
yadj	numeric y adjustment
curve	integer relative size of curve (fraction of points distance)
dd	length of segment arm, directed away from endpoints.
...	Further argument to be passed to straightarrow()

---

myarrow2 *Draw arrow with adjustment of a position*

Description

Draw arrow with adjustment of a position

Usage

```
myarrow2(
  nodes,
  from,
  to,
  label = "",
  no,
  radx = 0.12,
  rady = 0.04,
  xmargin = 0.01,
  label.pos = 0.5,
  arr.pos = NULL,
  addprime = TRUE,
  xspace = NULL,
  mode = 1,
  ...
)
```

Arguments

nodes	A data.frame
from	coordinates (x,y) of the point *from* which to draw arrow.
to	coordinates (x,y) of the point *to* which to draw arrow.
label	label to display
no	process macro model number
radx	horizontal radius of the box.
rady	vertical radius of the box.
xmargin	horizontal margin of plot
label.pos	label position
arr.pos	arrow position
addprime	logical Whether add prime to label "c"
xspace	numeric horizontal space between nodes
mode	integer mode for adjustxpos
...	Further argument to be passed to straightarrow()

---

mycat *append something to file*

Description

append something to file

Usage

```
mycat(..., file = "report.Rmd")
```

Arguments

...	Further argument passed to the cat()
file	name of file

---

mycor *Perform correlation and linear regression for a data.frame*

Description

Perform correlation and linear regression for a data.frame

Usage

```
mycor(x, digits = 3)
```

Arguments

x	A data.frame
digits	integer indicating the number of decimal places

---

myflatten *flatten string*

Description

flatten string

Usage

```
myflatten(x)
```

Arguments

x	character to flatten

myformat *Format a numeric vector*

Description

Format a numeric vector

Usage

```
myformat(x, digits = 3)
```

Arguments

x	A numeric vector
digits	integer indicating the number of decimal places

mylm *Correlation and Fitting linear model function for function "mycor"*

Description

Correlation and Fitting linear model function for function "mycor"

Usage

```
mylm(y, x, digits = 3)
```

Arguments

y	numeric vectors of data values
x	numeric vectors of data values
digits	integer indicating the number of decimal places (round) or significant digits (signif) to be used.

Value

mylm returns a list of following components

out a list of class "htest" from `cor.test` between the last paired samples in a data.frame.

result a numeric vector of length 4, consist of r and p values from `cor.test`,slope and intercept values from `lm` between numeric vector y and x

nodes *Node Data Set for drawing statistical diagram of process macro model*

Description

Node Data Set for drawing statistical diagram of process macro model

Usage

```
nodes
```

Format

A data.frame with 327 rows and 4 variables

no process macro model number

name name of node

xpos x position

ypos y position

numberSubscript *Make number subscript*

Description

Make number subscript

Usage

```
numberSubscript(ft, label, vanilla)
```

Arguments

ft	An object of class flextable
label	string vector
vanilla	logical

p2asterisk *Convert p values to asterisk*

Description

Convert p values to asterisk

Usage

```
p2asterisk(x)
```

Arguments

x	a numeric vector or matrix

p2chr *Convert p values to character*

Description

Convert p values to character

Usage

```
p2chr(x)
```

Arguments

x	A vector

parallelMatrix *Make bmatrix for parallel multiple mediator model*

Description

Make bmatrix for parallel multiple mediator model

Usage

```
parallelMatrix(n = 2)
```

Arguments

n	integer number of mediator

Examples

```
parallelMatrix(3)
```

parrows — *Arrow Data Set for drawing statistical diagram of process macro model*

Description

Arrow Data Set for drawing statistical diagram of process macro model

Usage

```
parrows
```

Format

A data.frame with 392 rows and 6 variables

no process macro model number

name name of arrow

start start node

end end node

labelpos position of label

arrowpos position of arrow head

pastecolon — *paste two character with colon*

Description

paste two character with colon

Usage

```
pastecolon(temp, x)
```

Arguments

temp	a character
x	a character

pformat *Make p value format*

Description

Make p value format

Usage

```
pformat(x)
```

Arguments

x	A numeric vector

plot.mediationBK *S3 method for class mediationBK*

Description

S3 method for class mediationBK

Usage

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

Arguments

x	An object of class mediationBK
...	Further arguments to be passed to plot()

Examples

```
labels=list(X="cond",M="pmi",Y="reaction")
result=mediationBK(labels=labels,data=pmi)
plot(result,type=1)
plot(result)
plot(result,type=1,whatLabel="label",arrowslabels="c",addprime=FALSE)
plot(result,whatLabel="label",arrowslabels=c("a","b","c"))
```

plotCoef *Make Slopes Plot*

Description

Make Slopes Plot

Usage

```
plotCoef(ss, color = "deepskyblue2", size = 0.75, digits = 1)
```

Arguments

ss	An object of class sim_slopes
color	Name of color
size	size of pointrange
digits	An integer indicating the number of decimal places

Value

A ggplot

pmacro *Data Set for process macro model*

Description

Data Set for process macro model

Usage

```
pmacro
```

Format

A data.frame with 43 rows and 7 variables

no process macro model number

X name of independent variable

M names of mediator variables

Y name of dependent variable

modName names of moderator variables

modSite sites of moderators

pos position of moderators

pmacroModel *draw conceptual diagram of process macro model*

Description

draw conceptual diagram of process macro model

Usage

```
pmacroModel(
  no = 1,
  labels = list(),
  covar = list(),
  radx = 0.06,
  rady = 0.06,
  xmargin = 0.03,
  box.col = "white",
  xlim = NULL,
  ylim = NULL
)
```

Arguments

no	process macro model number
labels	A character list
covar	A optional list of covariates
radx	horizontal radius of the box.
rady	vertical radius of the box.
xmargin	horizontal margin of plot
box.col	fill color of box
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot

Examples

```
pmacroModel(1)
covar=list(name=c("C1","C2"),label=c("ese","sex","tenure"),site=list("Y","Y"))
pmacroModel(1,covar=covar)
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list("M",c("Mi","Y"),c("Y")))
pmacroModel(4,covar=covar)
```

pmi *PMI: Presumed Media Influence dataset*

Description

PMI: Presumed Media Influence dataset

Usage

```
pmi
```

Format

A data.frame with 123 obs. of 6 variables

cond front (1) or interior (0) page of the newspaper

pmi presumed media influence

import article is on an important topic

reaction sugar purchase

gender GENDER: female (0) or male (1)

age age

Source

Tal-Or, N., Cohen, J., Tsafati, Y., & Gunther, A. C. (2010). Testing causal direction in the influence of presumed media influence. Communication Research, 37, 801-824.

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

print.compareVIF *S3 method of class compareVIF*

Description

S3 method of class compareVIF

Usage

```
## S3 method for class 'compareVIF'
print(x, ...)
```

Arguments

x	An object of class compareVIF
...	Further arguments to be passed to print

print.meanSummary *S3 method of class meanSummary*

Description

S3 method of class meanSummary

Usage

```
## S3 method for class 'meanSummary'
print(x, ...)
```

Arguments

x	An object of class meanSummary
...	Further arguments to be passed to print()

print.mediationBK *S3 method for class mediationBK*

Description

S3 method for class mediationBK

Usage

```
## S3 method for class 'mediationBK'
print(x, ...)
```

Arguments

x	An object of class mediationBK
...	Further arguments to be passed to print()

print.medSummary *S3 method print for an object of class medSummary*

Description

S3 method print for an object of class medSummary

Usage

```
## S3 method for class 'medSummary'
print(x, ...)
```

Arguments

x	An object of class medSummary
...	additional arguments to pass to print.medSummary

print.medSummary2 *S3 method print for an object of class medSummary2*

Description

S3 method print for an object of class medSummary2

Usage

```
## S3 method for class 'medSummary2'
print(x, ...)
```

Arguments

x	An object of class medSummary
...	additional arguments to pass to print.medSummary

print.modelSummary *S3 method print for object modelSummary*

Description

S3 method print for object modelSummary

Usage

```
## S3 method for class 'modelSummary'
print(x, ...)
```

Arguments

x	Object of class modelSummary
...	additional arguments to pass to print.modelSummary

print.modelSummary2 *S3 method print for object modelSummary2*

Description

S3 method print for object modelSummary2

Usage

```
## S3 method for class 'modelSummary2'
print(x, ...)
```

Arguments

x	Object of class modelSummary
...	additional arguments to pass to print.modelSummary

print.modmedSummary *S3 method print for an object of class modmedSummary*

Description

S3 method print for an object of class modmedSummary

Usage

```
## S3 method for class 'modmedSummary'
print(x, ...)
```

Arguments

x	An object of class modmedSummary
...	additional arguments to pass to print.modmedSummary

print.modmedSummary2 *S3 method print for an object of class modmedSummary2*

Description

S3 method print for an object of class modmedSummary2

Usage

```
## S3 method for class 'modmedSummary2'
print(x, ...)
```

Arguments

x	An object of class modmedSummary2
...	additional arguments to pass to print.modmedSummary2

print.modSummary *S3 method of class modSummary*

Description

S3 method of class modSummary

Usage

```
## S3 method for class 'modSummary'
print(x, ...)
```

Arguments

x	An object of class modSummary
...	Further arguments to be passed to print

productEq *Make products of equations*

Description

Make products of equations

Usage

```
productEq(equation1, equation2)
```

Arguments

equation1	The first equation
equation2	The second equation

Examples

```
equation1=c("a1+b1*W")
equation2=c("a2+b2*W")
productEq(equation1,equation2)
```

protest *Protest dataset*

Description

Garcia, Schmitt, Branscombe, and Ellemers (2010) report data for 129 subjects on the effects of perceived sexism on anger and liking of women's reactions to ingroup members who protest discrimination. This data set is also used as the 'protest' data set by Hayes (2013 and 2018). It is a useful example of mediation and moderation in regression. It may also be used as an example of plotting interactions.

Usage

```
protest
```

Format

A data.frame with 129 rows and 6 variables

subnum subject number

protest experimental condition, 0 = no protest, 1 = individual protest, 2 = group protest

sexism perceived pervasiveness of sex discrimination. Means of an 8 item Modern Sexism Scale.

angry anger toward the attorney. "I feel angry towards Catherine".

liking liking of the attorney. Mean rating of 6 liking ratings of the target.

respappr appropriateness of response. Mean of four items of appropriateness of the target's response.

Details

The reaction of women to women who protest discriminatory treatment was examined in an experiment reported by Garcia et al. (2010). 129 women were given a description of sex discrimination in the workplace (a male lawyer was promoted over a clearly more qualified female lawyer). Subjects then read that the target lawyer felt that the decision was unfair. Subjects were then randomly assigned to three conditions: Control (no protest), Individual Protest ("They are treating me unfairly") , or Collective Protest ("The firm is is treating women unfairly"). Participants were then asked how much they liked the target (liking), how angry they were to the target (anger) and to evaluate the appropriateness of the target's response (respappr). Garcia et al(2010) report a number of interactions (moderation effects) as well as moderated-mediation effects.

Source

Garcia, D. M., Schmitt, M. T., Branscombe, N. R., & Ellemers, N. (2010). Women's reactions to ingroup members who protest discriminatory treatment: The importance of beliefs about inequality and response appropriateness. European Journal of Social Psychology, 40, 733-745.

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

qqPlot *Draw quantile-quantile plot*

Description

Draw quantile-quantile plot

Usage

```
qqPlot(x, linecolor = "red", xlab = NULL, ylab = NULL, title = NULL, ...)
```

Arguments

x	A numeric vector
linecolor	character line color
xlab	character label for x axis
ylab	character label for y axis
title	character label for plot title
...	Further arguments to be passed to geom_qq()

Examples

```
qqPlot(rnorm(200))
qqPlot(rt(200, df = 5))
```

r2diff *Calculate difference of R2 and adjusted R2*

Description

Calculate difference of R2 and adjusted R2

Usage

```
r2diff(fit, mode = 1, digits = 3)
```

Arguments

fit	An object of class lm
mode	Integer If 1, remove all interaction. If 2, remove variables one by one
digits	Integer indicating the number of decimal places

Examples

```
fit=lm(mpg~wt*hp,data=mtcars)
r2diff(fit)
r2diff(fit,mode=2)
```

r2pptx *Make powerpoint presentation from R file*

Description

Make powerpoint presentation from R file

Usage

```
r2pptx(
  file,
  filename = "report.pptx",
  keyword = c("Concept", "Diagram", "Model", "Plot", "plot", "Table", "summary"),
  rmdRemove = TRUE
)
```

Arguments

file	source file name
filename	destination file name
keyword	A string vector
rmdRemove	A logical

regEquation *Make regression equation*

Description

Make regression equation

Usage

```
regEquation(
  X = "X",
  M = NULL,
  Y = "Y",
  moderator = list(),
  covar = list(),
  secondIndirect = FALSE
)
```

Arguments

X	A character vectors indicating independent variables
M	A character vectors indicating mediators
Y	A character vectors indicating dependent variables
moderator	moderator
covar	covariates
secondIndirect	A logical

Examples

```
X="X";M=NULL;Y="Y"; moderator=list(name="W",site=list("c"))
regEquation(X,M,Y,moderator)
M=c("M1","M2")
regEquation(X,M,Y,moderator,secondIndirect=TRUE)
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M1","Y"),"Y","Y"))
regEquation(X,M,Y,moderator,covar=covar)
covar=list(name=c("ese","sex","tenure"),site=list(c("M","Y"),c("M","Y"),c("M","Y")))
regEquation(X="estress",M="affect",Y="withdraw",covar=covar)
```

---

reliabilityTable *make reliability Table*

Description

make reliability Table

Usage

```
reliabilityTable(fit)
```

Arguments

fit	An object of a class lavaan

reliabilityTable2 *make reliability Table in flextable format*

Description

make reliability Table in flextable format

Usage

```
reliabilityTable2(fit, vanilla = FALSE)
```

Arguments

fit	An object of a class lavaan
vanilla	Logical

removeParentheses *Remove parentheses*

Description

Remove parentheses

Usage

```
removeParentheses(string)
```

Arguments

string	A character vector

rightPrint *Print a string in right alignment*

Description

Print a string in right alignment

Usage

```
rightPrint(string, width)
```

Arguments

string	A string
width	A numeric

seekGroup *Find group with variable name*

Description

Find group with variable name

Usage

```
seekGroup(var, res, group)
```

Arguments

var	A string to seek
res	A data.frame. Result of parameterEstimates function of package lavaan or subset.
group	A string vector

seekGroup1 *Find group with variable name*

Description

Find group with variable name

Usage

```
seekGroup1(var, res)
```

Arguments

var	A string to seek
res	A data.frame. Result of parameterEstimates function of package lavaan or subset.

seekGroup2 *Find group with variable name*

Description

Find group with variable name

Usage

```
seekGroup2(var, res, group)
```

Arguments

var	A string to seek
res	A data.frame. Result of parameterEstimates function of package lavaan or subset.
group	A character vector

seekNameVars *select names of variables from list var*

Description

select names of variables from list var

Usage

```
seekNameVars(vars, site = "a")
```

Arguments

vars	A list
site	Site for look for

Examples

```
vars=list(name=list(c("W","Z"),c("V","Q")),site=list(c("a","c"),c("b","c")))
vars=list(name=list(c("W","Z")),site=list(c("a","c")))
seekNameVars(vars,"a")
seekNameVars(vars,"b")
seekNameVars(vars,"c")
```

seekVar *Seek var form covariates*

Description

Seek var form covariates

Usage

```
seekVar(
  covar = list(),
  var,
  prefix = "h",
  start = 1,
  grouplabels = NULL,
  suffix = NULL
)
```

Arguments

covar	A list of covariates
var	A name of variable to look for
prefix	A prefix
start	A start number
grouplabels	A list
suffix	A suffix

Examples

```
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M","Y"),"Y","Y"))
var="Y"
seekVar(covar,var,prefix="h")
```

separateEq *Separate equation*

Description

Separate equation

Usage

```
separateEq(equation)
```

Arguments

equation	string. Equations to separate

Examples

```
equation="( a1 + b1 * W )"
separateEq(equation)
```

---

setPositionNodes *Set Position of nodes*

Description

Set Position of nodes

Usage

```
setPositionNodes(
  nodes,
  arrows,
  radx = 0.08,
  rady = 0.06,
  xmargin = 0.02,
  ymargin = 0.02,
  xlim = c(-0.3, 1.35),
  ylim = c(-0.07, 1.05),
  parallel2 = FALSE,
  parallel3 = FALSE
)
```

Arguments

nodes	A data.frame of nodes
arrows	A data.frame of arrows
radx	horizontal radius of the box.
rady	vertical radius of the box.
xmargin	horizontal margin between nodes
ymargin	vertical margin between nodes
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot
parallel2	logical
parallel3	logical

showModels *Run process macro shiny app*

Description

Run process macro shiny app

Usage

```
showModels()
```

standardize *Standardize variable*

Description

Standardize variable

Usage

```
standardize(x)
```

Arguments

x	A numeric vector

standardizeDf *standardize data*

Description

standardize data

Usage

```
standardizeDf(df, names)
```

Arguments

df	A data.frame
names	column names to mean centering

`statisticalDiagram` *Draw statistical diagram*

Description

Draw statistical diagram

Usage

```
statisticalDiagram(
  no = 1,
  radx = 0.1,
  rady = 0.04,
  xmargin = 0.01,
  arrowlabel = TRUE,
  arrowslabels = NULL,
  arrowslty = NULL,
  labels = list(),
  nodeslabels = list(),
  whatLabel = "name",
  fit = NULL,
  estimatesTable = NULL,
  digits = 3,
  covar = list(),
  addCovar = TRUE,
  type = NULL,
  includeLatentVars = FALSE,
  addprime = TRUE,
  box.col = "white",
  xlim = c(0, 1),
  ylim = NULL
)
```

Arguments

`no`	process macro model number
`radx`	horizontal radius of the box.
`rady`	vertical radius of the box.
`xmargin`	horizontal margin of plot
`arrowlabel`	logical whether or not draw arrowlabel
`arrowslabels`	A character vector
`arrowslty`	linetype of arrows
`labels`	A list of character string
`nodeslabels`	A list of character string

whatLabel	What should the edge labels indicate in the path diagram? Choices are c("est","std","name","label")
fit	A list of class lm or an object of class lavaan
estimatesTable	A data.frame
digits	Integer indicating the number of decimal places
covar	Optional list of covariates
addCovar	Logical. Whether or not include covariates
type	An integer
includeLatentVars	A logical
addprime	logical. Whether or not add prime to label "c"
box.col	fill color of the box
xlim	the x limits (min,max) of the plot
ylim	the y limits (min,max) of the plot

Examples

```
statisticalDiagram(no=1)
covar=list(name=c("posemot","ideology","sex"),site=list(c("Y"),c("Y"),c("Y")))
statisticalDiagram(no=1,covar=covar)
covar=list(name=c("posemot","ideology","sex"),site=list(c("M","Y"),c("Mi","Y"),c("Mi","Y")))
covar=list(name=c("C1","C2"),site=list(c("M","Y"),"Y"))
statisticalDiagram(no=4,covar=covar)
statisticalDiagram(no=8,covar=covar)
labels=list(X="wintense",Mi="cogapp",Y="emotion")
nodeslabels=list(X="Work\nIntensity",Mi="Cognitive\nAppraisal",Y="Emotional\nExhaustion")
statisticalDiagram(4,labels=labels)
statisticalDiagram(4,labels=nodeslabels)
statisticalDiagram(4,labels=labels,nodeslabels=nodeslabels)
labels=list(X="GDP\nper inhabitant",M="Illiteracy Rate",Y="Mean Life\nExpectation")
statisticalDiagram(4,labels=labels)
statisticalDiagram(4,labels=labels,arrowslabels=c("e","f","g"),whatLabel="label")
```

---

str2vector *Make character vector from string*

Description

Make character vector from string

Usage

```
str2vector(string = "a,b,c")
```

Arguments

string	string

strGrouping *Make Grouping equation*

Description

Make Grouping equation

Usage

```
strGrouping(x, groupby = "X")
```

Arguments

x	character vector
groupby	name of groupby

str_detect2 *Extension of str_detect to list*

Description

Extension of str_detect to list

Usage

```
str_detect2(list, pattern)
```

Arguments

list	a list
pattern	pattern to look for

Examples

```
site=list(c("a","c"),c("a","b","c"))
str_detect2(site,"b")
```

str_setdiff *Remove matched pattern from string*

Description

Remove matched pattern from string

Usage

```
str_setdiff(string = "a,c", pattern = "a")
```

Arguments

string	string
pattern	pattern to look for

sumEquation *summation of equations*

Description

summation of equations

Usage

```
sumEquation(eq1, eq2)
```

Arguments

eq1	A equation
eq2	A equation

summary.mediationBK *S3 method for class mediationBK*

Description

S3 method for class mediationBK

Usage

```
## S3 method for class 'mediationBK'
summary(object, ...)
```

Arguments

object	An object of class mediationBK
...	Further arguments to be passed to summary()

teachers *Teacher Efficacy Data*

Description

A dataset containing teacher efficacy, social support, psychological wellbeing and job stress of 247 teachers working in kindergarten

Usage

```
teachers
```

Format

A data.frame with 247 rows and 7 variables:

age teacher's age. 1: 20-24, 2: 25-29, 3: 30-34, 4: 35-39, 5: 40-44, 6:45-49, 7: 50 or above

marriage Marital Status. 0: single, 2: married

children Parental Status. 0: no children, 1: one or more children

wellbeing Psychological Well-being

stress Job stress. A response syndrome of negative affects(such as anger or depression) resulting from the teacher's job

efficacy Teacher Efficacy. A judgement of his or her capabilities to bring about desired outcomes of student engagement and learning

support Social Support. Various resources provided by ones's interpersonal ties.

Source

Cohen, S., & Hoberman, H. M. (1983). Positive events and social support as butters of life change stress. Journal of Social Applied Psychology, 13, 99-125

Tschannen-Moran, M., & Hoy, A. W. (2001). Teacher efficacy: Capturing an elusive construct. Teaching and teacher education, 17(7), 783-805

Ryff, Carol D. (1989). Happiness Is Everything, or Is It? Explorationson the Meaning of Psychological Well-Being. Journal of-Personality and Social Psychology, 57, 1069-1081

Kyriacou, C., & Sutcliffe, J. (1978). Teacher stress: Prevalence, sources, and symptoms. British Journal of Educational Psychology, 55, 61-64

`teams` *Teams data set*

Description

Teams data set

Usage

```
teams
```

Format

A data.frame with 60 rows and 4 variables

dysfunc Dysfunctional team behavior

negtone Negative affective tone

negexp Negative expressivity

perform Team performance

Source

Cole, M. S., Walter, F., & Bruch, H. (2008). Affective mechanisms linking dysfunctional behavior to performance in work teams: A moderated mediation study. Journal of Applied Psychology, 93, 945-958.

http://www.afhayes.com/introduction-to-mediation-moderation-and-conditional-process-analysis.html

theme_clean2 *Clean theme for ggCor*

Description

Clean theme for ggCor

Usage

```
theme_clean2(base_size = 12, xangle = 45, yangle = 0)
```

Arguments

base_size	base font size
xangle	x-axis text angle
yangle	y-axis text angle

treatInteraction *unfold interaction*

Description

unfold interaction

Usage

```
treatInteraction(var)
```

Arguments

var	name of variables

Examples

```
var="X*M"
treatInteraction(var)
var="X*M*W"
treatInteraction(var)
```

treatModerator *Treat moderator name with mean value*

Description

Treat moderator name with mean value

Usage

```
treatModerator(
  ind,
  moderatorNames,
  data = NULL,
  rangemode = 1,
  probs = c(0.16, 0.5, 0.84)
)
```

Arguments

ind	An equation
moderatorNames	character vectors
data	A data.frame
rangemode	range mode
probs	numeric vector of probabilities with values in [0,1]

Examples

```
ind="(a1+a4*sex+a5*age)*(b1)"
moderatorNames=c("age","sex")
treatModerator(ind,moderatorNames)
ind="c1+c3*hp"
moderatorNames="hp"
treatModerator(ind,moderatorNames)
```

tripleEquation *Make equation with triple interaction*

Description

Make equation with triple interaction

Usage

```
tripleEquation(
  X = NULL,
  M = NULL,
  Y = NULL,
  labels = list(),
  vars = NULL,
  suffix = 0,
  moderator = list(),
  covar = NULL,
  range = TRUE,
  mode = 0,
  data = NULL,
  rangemode = 1,
  probs = c(0.16, 0.5, 0.84),
  effectsize = FALSE
)
```

Arguments

X	Name of independent variable
M	Name of mediator
Y	Name of dependent variable
labels	A list of variables
vars	A list of variables names and sites
suffix	A number
moderator	A list of moderators
covar	A list of covariates
range	A logical
mode	A number
data	A data.frame
rangemode	range mode
probs	numeric vector of probabilities with values in [0,1]
effectsize	logical If true, calculate effectsize

Examples

```
X="negemot";M="ideology";Y="govact";suffix=0
cat(tripleEquation(X=X,M=M,Y=Y))
vars=list(name=list(c("sex","age")),site=list(c("a","c")))
vars=list(name=list(c("W","Z"),c("V","Q")),site=list(c("a","b","c"),c("a","b","c")))
X="negemot";Y="govact";suffix=0
moderator=list(name=c("W"),site=list(c("b","c")))
cat(tripleEquation(X=X,Y=Y,moderator=moderator))
covar=list(name=c("C1","C2","C3"),site=list(c("M","Y"),c("Mi","Y"),"Y"))
```

```
labels=list(X="negemot",M="ideology",Y="govact")
cat(tripleEquation(labels=labels,moderator=moderator,covar=covar))
cat(tripleEquation(X=X,M=M,Y=Y,moderator=moderator,covar=covar,mode=1))
cat(tripleEquation(X=X,M=M,Y=Y,vars=vars))
cat(tripleEquation(X=X,M=M,Y=Y,vars=vars,moderator=moderator,covar=covar))
cat(tripleEquation(X=X,M=M,Y=Y,vars=vars,mode=1))
cat(tripleEquation(X=X,M=M,Y=Y,vars=vars,covar=covar,mode=1))
X="negemot";Y="govact";suffix=0
vars=list(name=list(c("sex","age")),site=list(c("c")))
cat(tripleEquation(X=X,Y=Y,vars=vars))
```

`tripleInteraction` *Make triple interaction equation*

Description

Make triple interaction equation

Usage

```
tripleInteraction(vars, prefix = "c", suffix = 0, mode = 0, addPrefix = TRUE)
```

Arguments

`vars`	variable names to be interact
`prefix`	A character
`suffix`	A number
`mode`	A number
`addPrefix`	A logical

Examples

```
vars=c("negemot","sex","age")
tripleInteraction(vars)
tripleInteraction(vars,mode=1)
```

unfold *Unfold equations*

Description

Unfold equations

Usage

```
unfold(string, var = "W", mode = -1)
```

Arguments

string	Character vectors with equation
var	name of variable
mode	integer. Default value is -1. If 0, get intercept, If 1, get slope

Examples

```
string=c("(a1+b1*W)*(a2+b2*W)*(a3+b3*W)","a1+b1*W")
unfold(string)
```

vars2df *Make data.frame from a list of vars*

Description

Make data.frame from a list of vars

Usage

```
vars2df(vars, mpos = c(0.5, 0.9), df = NULL)
```

Arguments

vars	A list
mpos	A numeric vector of length 2
df	A data.frame

Examples

```
vars=list(name=list(c("tenure","age")),site=list(c("a","b")))
vars2df(vars)
vars=list(name=list(c("milk","hair")),matrix=list(c(1,0,0,0,0,0,1,0,0,0)),pos=5)
vars2df(vars)
```

vif *Variance Inflation Factors Calculates variance-inflation and generalized variance-inflation factors for linear, generalized linear, and other models.*

Description

Variance Inflation Factors Calculates variance-inflation and generalized variance-inflation factors for linear, generalized linear, and other models.

Usage

```
vif(mod)
```

Arguments

mod	an object that responds to coef, vcov, and model.matrix, such as an lm or glm object.

Value

A vector of vifs, or a matrix containing one row for each term in the model, and columns for the GVIF, df.

ztable.compareMC *S3 method for class compareMC*

Description

S3 method for class compareMC

Usage

```
## S3 method for class 'compareMC'
ztable(x, digits = digits, ...)
```

Arguments

x	An object of class compareMC
digits	integer indicating the number of decimal places
...	further arguments to be passed to ztable

Examples

```
require(ztable)
fit=lm(govact~negemot*age,data=glbwarm)
res=compareMC(fit)
ztable(res)
```

`ztable.modelSummary` *S3 method for class 'modelSummary'*

Description

S3 method for class 'modelSummary'

Usage

```
## S3 method for class 'modelSummary'
ztable(x, digits = NULL, ...)
```

Arguments

`x`	An object of class modelSummary
`digits`	integer indicating the number of decimal places
`...`	Further argument to be passed to ztable

Index