

# Package ‘processR’

March 7, 2019

**Type** Package

**Title** Implementation of the 'PROCESS' Macro

**Version** 0.1.0

**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)

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Imports** lavaan(>= 0.6.3), diagram, magrittr, dplyr, flextable,  
ggiraphExtra, ggplot2, jtools, mycor, officer, psych, purrr,  
rrtable, semTools, stringr, tidycselect, shiny, modelr,  
prediction, rlang, tidyr, TH.data, shinyWidgets

**RoxygenNote** 6.1.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

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

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

**Repository** CRAN

**Date/Publication** 2019-03-07 17:40:03 UTC

## R topics documented:

addArrows . . . . .	4
addCatVars . . . . .	4

addCovarEquation . . . . .	5
addLatentNodes . . . . .	6
addLine . . . . .	6
addNodes . . . . .	7
addPlus . . . . .	7
adjustNodes . . . . .	8
adjustPosNodes . . . . .	8
adjustxpos . . . . .	8
caskets . . . . .	9
catMediation . . . . .	9
centerPrint . . . . .	10
changeLabelName . . . . .	11
conceptDiagram . . . . .	11
conceptDiagram2 . . . . .	12
conditionalEffectPlot . . . . .	13
convertPvalue . . . . .	13
corPlot . . . . .	14
corTable . . . . .	14
corTable2 . . . . .	15
countM . . . . .	15
disaster . . . . .	16
discriminantValidityTable . . . . .	16
discriminantValidityTable2 . . . . .	17
drawArrows . . . . .	17
drawCovar . . . . .	18
drawStatDiagram . . . . .	18
drawtext . . . . .	19
education . . . . .	19
eq2df . . . . .	20
est2Arrows . . . . .	20
est2Nodes . . . . .	20
estimatesTable . . . . .	21
estimatesTable2 . . . . .	21
estress . . . . .	22
extractLatentVar . . . . .	22
extractLatentVarName . . . . .	23
extractX . . . . .	23
findName . . . . .	23
findNames . . . . .	24
fit2alpha . . . . .	24
fit2df2 . . . . .	25
fun2eq . . . . .	25
getAspectRatio . . . . .	25
getHelmert . . . . .	26
getInfo . . . . .	26
getRatioTable . . . . .	27
getYhat . . . . .	27
glbwarm . . . . .	28

interactStr . . . . .	29
makeCatEquation . . . . .	29
makeEquation . . . . .	30
makeEquation1 . . . . .	31
makeEquation2 . . . . .	31
makeEquation3 . . . . .	32
makeIndirectEquation . . . . .	32
makeIndirectEquationCat . . . . .	33
meanCentering . . . . .	34
modelFitGuideTable . . . . .	34
modelFitGuideTable2 . . . . .	34
modelFitTable . . . . .	35
modelFitTable2 . . . . .	35
modelsSummary . . . . .	36
modelsSummaryTable . . . . .	36
moderator2pos . . . . .	37
modmedEquation . . . . .	37
modmedSummary . . . . .	38
modmedSummaryTable . . . . .	39
myarrow . . . . .	39
myarrow2 . . . . .	40
myflatten . . . . .	40
myformat . . . . .	41
nodes . . . . .	41
p2asterisk . . . . .	42
parrows . . . . .	42
pastecolon . . . . .	43
pformat . . . . .	43
pmacro . . . . .	43
pmacroModel . . . . .	44
pmi . . . . .	45
print.modelSummary . . . . .	45
print.modmedSummary . . . . .	46
protest . . . . .	46
regEquation . . . . .	47
reliabilityTable . . . . .	48
reliabilityTable2 . . . . .	48
removeParentheses . . . . .	49
rightPrint . . . . .	49
seekGroup . . . . .	49
seekGroup1 . . . . .	50
seekGroup2 . . . . .	50
seekNameVars . . . . .	51
seekVar . . . . .	51
showModels . . . . .	52
statisticalDiagram . . . . .	52
str2vector . . . . .	53
strGrouping . . . . .	53

str_detect2 . . . . .	54
str_setdiff . . . . .	54
sumEquation . . . . .	55
teams . . . . .	55
treatModerator . . . . .	56
tripleEquation . . . . .	56
tripleInteraction . . . . .	57

<b>Index</b>	<b>59</b>
--------------	-----------

---

addArrows	<i>Add covariates to arrows</i>
-----------	---------------------------------

---

### Description

Add covariates to arrows

### Usage

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

### Arguments

arrows	A data.frame
covar	A list of covariates

---

addCatVars	<i>Add dummy vars to data.frame</i>
------------	-------------------------------------

---

### 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	<i>Add covariates to equation</i>
------------------	-----------------------------------

---

**Description**

Add covariates to equation

**Usage**

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

**Arguments**

equation	The equation
covar	A list
prefix	prefix
grouplabels	A list

**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)
```

---

addLatentNodes	<i>Add latent nodes information to nodes</i>
----------------	----------------------------------------------

---

**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	<i>Add line feed to string</i>
---------	--------------------------------

---

**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	<i>Add covariates to nodes</i>
----------	--------------------------------

---

**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	<i>Add '+' mark to string</i>
---------	-------------------------------

---

**Description**

Add '+' mark to string

**Usage**

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

**Arguments**

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

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)
```

**Arguments**

xpos            x position  
xmargin        horizontal margin of plot  
radx            horizontal radius of the box.

---

caskets	<i>CASKETS dataset</i>
---------	------------------------

---

**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	<i>Make Mediation Equation with one categorical variable</i>
--------------	--------------------------------------------------------------

---

**Description**

Make Mediation Equation with one categorical variable

**Usage**

```
catMediation(X, M = NULL, Y, 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
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**

```

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(X="cyl",M="am",Y="mpg",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	<i>Change Label Names</i>
-----------------	---------------------------

---

**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="skeptic",Mi="empathy",Y="intervention",W="frame")
x=c("skeptic","test","empathy","skeptic:frame","D1:frame")
changeLabelName(x,labels)
changeLabelName(x,labels,add=TRUE)
```

---

conceptDiagram	<i>Make conceptDiagram</i>
----------------	----------------------------

---

**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, 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
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(fit, values = NULL, data, mod = "skeptical")
```

**Arguments**

fit	An object of class lavaan
values	Optional value
data	A data.frame
mod	Name of moderator variable

---

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	<i>Draw correlation plot</i>
---------	------------------------------

---

**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	<i>Make a table with correlation</i>
----------	--------------------------------------

---

**Description**

Make a table with correlation

**Usage**

```
corTable(fit)
```

**Arguments**

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

---

corTable2	<i>Make a table with correlation</i>
-----------	--------------------------------------

---

**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	<i>Count the group names start with "M"</i>
--------	---------------------------------------------

---

**Description**

Count the group names start with "M"

**Usage**

```
countM(group)
```

**Arguments**

group	A string vectors
-------	------------------

---

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

---

 drawArrows

*Draw arrows*


---

**Description**

Draw arrows

**Usage**

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

**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.

---

drawCovar	<i>Draw covariate</i>
-----------	-----------------------

---

**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

---

drawStatDiagram	<i>draw StatDiagram</i>
-----------------	-------------------------

---

**Description**

draw StatDiagram

**Usage**

```
drawStatDiagram(no, arrows, nodes, labels, xmargin, radx, rady,
               fit = NULL)
```

**Arguments**

no	process macro model number
arrows	A data.frame
nodes	A data.frame
labels	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()

---

drawtext	<i>Draw node</i>
----------	------------------

---

**Description**

Draw node

**Usage**

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

**Arguments**

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

---

education	<i>Data Set for education and income</i>
-----------	------------------------------------------

---

**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** education duration

**income** income

---

eq2df	<i>Convert equation to data.frame</i>
-------	---------------------------------------

---

**Description**

Convert equation to data.frame

**Usage**

```
eq2df(eq)
```

**Arguments**

eq	equation seperated by linefeed
----	--------------------------------

---

est2Arrows	<i>Make arrows from estimatesTable</i>
------------	----------------------------------------

---

**Description**

Make arrows from estimatesTable

**Usage**

```
est2Arrows(res)
```

**Arguments**

res	A data.frame, result of estimatesTable
-----	----------------------------------------

---

est2Nodes	<i>Make nodes from estimatesTable</i>
-----------	---------------------------------------

---

**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 = FALSE, digits = 2, 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	<i>ESTRESS: Economic stress dataset</i>
---------	-----------------------------------------

---

**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>

---

extractLatentVar	<i>Extract Latent Variables Data</i>
------------------	--------------------------------------

---

**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()

---

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, name = "MiX", exact = FALSE)
```

**Arguments**

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

**Examples**

```
labels=list(X="wt",Mi="am",Y="mpg");name="MiX"
findName(labels,name)
```

---

findNames	<i>convert a vector of names with list</i>
-----------	--------------------------------------------

---

**Description**

convert a vector of names with list

**Usage**

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

**Arguments**

labels	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)
```

---

fit2alpha	<i>Make a Cronbach alpha table</i>
-----------	------------------------------------

---

**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	<i>Make a data.frame for conceptDiagram</i>
---------	---------------------------------------------

---

**Description**

Make a data.frame for conceptDiagram

**Usage**

```
fit2df2(fit)
```

**Arguments**

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

---

---

fun2eq	<i>Make equation from function</i>
--------	------------------------------------

---

**Description**

Make equation from function

**Usage**

```
fun2eq(fun)
```

**Arguments**

fun	A function
-----	------------

---

---

getAspectRatio	<i>Get aspect information of a ggplot</i>
----------------	-------------------------------------------

---

**Description**

Get aspect information of a ggplot

**Usage**

```
getAspectRatio(p)
```

**Arguments**

p	A ggplot object
---	-----------------

---

getHelmert	<i>Get Helmert Coding of column j of group with length of unique values (count-1)</i>
------------	---------------------------------------------------------------------------------------

---

**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	<i>Get information of a model</i>
---------	-----------------------------------

---

**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,data=mtcars)
getInfo(fit)
```

---

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)
```

---

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**

<code>fit</code>	Object of class "lm"
<code>group</code>	names of dummy variables in formula
<code>mode</code>	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)
```

---

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	<i>make interaction equation</i>
-------------	----------------------------------

---

**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)
```

---

makeCatEquation	<i>Make equation for sem and lm</i>
-----------------	-------------------------------------

---

**Description**

Make equation for sem and lm

**Usage**

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

**Arguments**

X	Name of independent variable
Y	Name of dependent variable
W	Name of moderators
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))

```

---

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))
```

### 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]

**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))

```

---

```
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

---

meanCentering	<i>Perform mean centering</i>
---------------	-------------------------------

---

**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])
```

---

modelFitGuideTable	<i>Model fit guide table</i>
--------------------	------------------------------

---

**Description**

Model fit guide table

**Usage**

```
modelFitGuideTable()
```

---

modelFitGuideTable2	<i>Model fit guide table</i>
---------------------	------------------------------

---

**Description**

Model fit guide table

**Usage**

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

**Arguments**

vanilla	Logical
---------	---------

---

modelFitTable	<i>Extract model fit measures to data.frame</i>
---------------	-------------------------------------------------

---

**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	<i>Extract model fit measures to flextable</i>
----------------	------------------------------------------------

---

**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	<i>Make Summary for Model Coefficients</i>
---------------	--------------------------------------------

---

**Description**

Make Summary for Model Coefficients

**Usage**

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

**Arguments**

fit	A list of objects of class lm
labels	optional list

**Value**

A data.frame

**Examples**

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

---

modelsSummaryTable	<i>Make Summary Table for Model Coefficients</i>
--------------------	--------------------------------------------------

---

**Description**

Make Summary Table for Model Coefficients

**Usage**

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

**Arguments**

x	An object of class modelSummary
vanilla	A logical

**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)
x=modelsSummary(list(fit1))
modelsSummaryTable(x)
x=modelsSummary(list(fit1,fit2))
modelsSummaryTable(x,vanilla=FALSE)
x=modelsSummary(list(fit1,fit2,fit3))
modelsSummaryTable(x)

```

---

moderator2pos	<i>get position from moderator</i>
---------------	------------------------------------

---

**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	<i>Make moderated mediation equation</i>
----------------	------------------------------------------

---

**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(fit, mod = "skeptical", values = NULL,
  boot.ci.type = "bca.simple")

```

**Arguments**

fit	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")

**Value**

A data.frame and an object of class modmedSummary

---

modmedSummaryTable	<i>Make a table summarizing the moderated mediation</i>
--------------------	---------------------------------------------------------

---

**Description**

Make a table summarizing the moderated mediation

**Usage**

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

**Arguments**

x	An object of class modmedSummary
vanilla	A logical

---

myarrow	<i>Draw arrow</i>
---------	-------------------

---

**Description**

Draw arrow

**Usage**

```
myarrow(from, to, lwd = 1, adjust = 1, label = "", label.pos = 0.5,
        arr.pos = NULL, ...)
```

**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
...	Further argument to be passed to straightarrow()

---

myarrow2	<i>Draw arrow with adjustment of a position</i>
----------	-------------------------------------------------

---

**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, ...)
```

**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
...	Further argument to be passed to straightarrow()

---

myflatten	<i>flatten string</i>
-----------	-----------------------

---

**Description**

flatten string

**Usage**

```
myflatten(x)
```

**Arguments**

x	character to flatten
---	----------------------

---

myformat	<i>Format a numeric vector</i>
----------	--------------------------------

---

**Description**

Format a numeric vector

**Usage**

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

**Arguments**

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

---

nodes	<i>Node Data Set for drawing statistical diagram of process macro model</i>
-------	-----------------------------------------------------------------------------

---

**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

---

p2asterisk	<i>Convert p values to asterisk</i>
------------	-------------------------------------

---

**Description**

Convert p values to asterisk

**Usage**

p2asterisk(x)

**Arguments**

x                    a numeric vector or matrix

---

parrows	<i>Arrow Data Set for drawing statistical diagram of process macro model</i>
---------	------------------------------------------------------------------------------

---

**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	<i>paste two character with colon</i>
------------	---------------------------------------

---

**Description**

paste two character with colon

**Usage**

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

**Arguments**

temp	a character
x	a character

---

pformat	<i>Make p value format</i>
---------	----------------------------

---

**Description**

Make p value format

**Usage**

```
pformat(x)
```

**Arguments**

x	A numeric vector
---	------------------

---

pmacro	<i>Data Set for process macro model</i>
--------	-----------------------------------------

---

**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)
```

**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

**Examples**

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

---

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.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.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

---

```
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 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>

---

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)
```

---

reliabilityTable	<i>make reliability Table</i>
------------------	-------------------------------

---

**Description**

make reliability Table

**Usage**

```
reliabilityTable(fit)
```

**Arguments**

fit	An object of a class lavaan
-----	-----------------------------

---

reliabilityTable2	<i>make reliability Table in flextable format</i>
-------------------	---------------------------------------------------

---

**Description**

make reliability Table in flextable format

**Usage**

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

**Arguments**

fit	An object of a class lavaan
vanilla	Logical

---

removeParentheses	<i>Remove parentheses</i>
-------------------	---------------------------

---

**Description**

Remove parentheses

**Usage**

```
removeParentheses(string)
```

**Arguments**

string	A character vector
--------	--------------------

---

rightPrint	<i>Print a string in center</i>
------------	---------------------------------

---

**Description**

Print a string in center

**Usage**

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

**Arguments**

string	A string
width	A numeric

---

seekGroup	<i>Find group with variable name</i>
-----------	--------------------------------------

---

**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	<i>Find group with variable name</i>
------------	--------------------------------------

---

**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	<i>Find group with variable name</i>
------------	--------------------------------------

---

**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	<i>select names of variables from list var</i>
--------------	------------------------------------------------

---

**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	<i>Seek var form covariates</i>
---------	---------------------------------

---

**Description**

Seek var form covariates

**Usage**

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

**Examples**

```

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

```

---

showModels	<i>Run process macro shiny app</i>
------------	------------------------------------

---

**Description**

Run process macro shiny app

**Usage**

```
showModels()
```

---

statisticalDiagram	<i>Draw statistical diagram</i>
--------------------	---------------------------------

---

**Description**

Draw statistical diagram

**Usage**

```

statisticalDiagram(no = 1, radx = 0.1, rady = 0.04, xmargin = 0.01,
  arrowlabel = TRUE, labels = list(), whatLabel = "name",
  fit = NULL, estimateTable = NULL, covar = list(),
  includeLatentVars = FALSE)

```

**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
labels	A character list
whatLabel	What should the edge labels indicate in the path diagram? Choices are c("est","std","name")
fit	An object of class lavaan. Result of lavaan::sem()
estimateTable	A data.frame
covar	Optional list of covariates
includeLatentVars	A logical

**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("Mi", "Y"), c("Mi", "Y"), c("Mi", "Y")))
statisticalDiagram(no=4, covar=covar)
statisticalDiagram(no=8, covar=covar)

```

---

str2vector	<i>Make character vector from string</i>
------------	------------------------------------------

---

**Description**

Make character vector from string

**Usage**

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

**Arguments**

string	string
--------	--------

---

strGrouping	<i>Make Grouping equation</i>
-------------	-------------------------------

---

**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	<i>summation of equations</i>
-------------	-------------------------------

---

**Description**

summation of equations

**Usage**

sumEquation(eq1, eq2)

**Arguments**

eq1	A equation
eq2	A equation

---

teams	<i>Teams data set</i>
-------	-----------------------

---

**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>

---

treatModerator	<i>Treat moderator name with mean value</i>
----------------	---------------------------------------------

---

**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)
```

---

tripleEquation	<i>Make equation with triple interaction</i>
----------------	----------------------------------------------

---

**Description**

Make equation with triple interaction

**Usage**

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

**Arguments**

X	Name of independent variable
M	Name of mediator
Y	Name of dependent variable
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]

**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("c")))
cat(tripleEquation(X=X,Y=Y,moderator=moderator))
covar=list(name=c("C1","C2","C3"),label=c("ese","sex","tenure"),site=list(c("M","Y"),"Y","Y"))
cat(tripleEquation(X=X,M=M,Y=Y,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	<i>Make triple interaction equation</i>
-------------------	-----------------------------------------

---

**Description**

Make triple interaction equation

**Usage**

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

**Arguments**

<code>vars</code>	variable names to be interact
<code>prefix</code>	A character
<code>suffix</code>	A number
<code>mode</code>	A number
<code>addPrefix</code>	A logical

**Examples**

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

# Index

## \*Topic **datasets**

- caskets, 9
  - disaster, 16
  - education, 19
  - estress, 22
  - glbwarm, 28
  - nodes, 41
  - parrows, 42
  - pmacro, 43
  - pmi, 45
  - protest, 46
  - teams, 55
- 
- addArrows, 4
  - addCatVars, 4
  - addCovarEquation, 5
  - addLatentNodes, 6
  - addLine, 6
  - addNodes, 7
  - addPlus, 7
  - adjustNodes, 8
  - adjustPosNodes, 8
  - adjustxpos, 8
- 
- caskets, 9
  - catMediation, 9
  - centerPrint, 10
  - changeLabelName, 11
  - conceptDiagram, 11
  - conceptDiagram2, 12
  - conditionalEffectPlot, 13
  - convertPvalue, 13
  - corPlot, 14
  - corTable, 14
  - corTable2, 15
  - countM, 15
- 
- disaster, 16
  - discriminantValidityTable, 16
  - discriminantValidityTable2, 17
- 
- drawArrows, 17
  - drawCovar, 18
  - drawStatDiagram, 18
  - drawtext, 19
- 
- education, 19
  - eq2df, 20
  - est2Arrows, 20
  - est2Nodes, 20
  - estimatesTable, 21
  - estimatesTable2, 21
  - estress, 22
  - extractLatentVar, 22
  - extractLatentVarName, 23
  - extractX, 23
- 
- findName, 23
  - findNames, 24
  - fit2alpha, 24
  - fit2df2, 25
  - fun2eq, 25
- 
- getAspectRatio, 25
  - getHelmert, 26
  - getInfo, 26
  - getRatioTable, 27
  - getYhat, 27
  - glbwarm, 28
- 
- interactStr, 29
- 
- makeCatEquation, 29
  - makeEquation, 30
  - makeEquation1, 31
  - makeEquation2, 31
  - makeEquation3, 32
  - makeIndirectEquation, 32
  - makeIndirectEquationCat, 33
  - meanCentering, 34
  - modelFitGuideTable, 34
  - modelFitGuideTable2, 34

modelFitTable, [35](#)  
modelFitTable2, [35](#)  
modelsSummary, [36](#)  
modelsSummaryTable, [36](#)  
moderator2pos, [37](#)  
modmedEquation, [37](#)  
modmedSummary, [38](#)  
modmedSummaryTable, [39](#)  
myarrow, [39](#)  
myarrow2, [40](#)  
myflatten, [40](#)  
myformat, [41](#)

nodes, [41](#)

p2asterisk, [42](#)  
parrows, [42](#)  
pastecolon, [43](#)  
pformat, [43](#)  
pmacro, [43](#)  
pmacroModel, [44](#)  
pmi, [45](#)  
print.modelSummary, [45](#)  
print.modmedSummary, [46](#)  
protest, [46](#)

regEquation, [47](#)  
reliabilityTable, [48](#)  
reliabilityTable2, [48](#)  
removeParentheses, [49](#)  
rightPrint, [49](#)

seekGroup, [49](#)  
seekGroup1, [50](#)  
seekGroup2, [50](#)  
seekNameVars, [51](#)  
seekVar, [51](#)  
showModels, [52](#)  
statisticalDiagram, [52](#)  
str2vector, [53](#)  
str\_detect2, [54](#)  
str\_setdiff, [54](#)  
strGrouping, [53](#)  
sumEquation, [55](#)

teams, [55](#)  
treatModerator, [56](#)  
tripleEquation, [56](#)  
tripleInteraction, [57](#)