

Package ‘radiant.multivariate’

July 24, 2016

Type Package

Title Multivariate analysis menu for Radiant. Builds on the radiant.data package

Version 0.5.5

Date 2016-7-17

Description Multivariate analysis menu for Radiant.

Depends R (>= 3.3.0),
radiant.data (>= 0.5.6)

Imports shiny (>= 0.13.2.9003),
dplyr (>= 0.5),
ggplot2 (>= 2.0.0),
gridExtra (>= 2.0.0),
scales (>= 0.4.0),
magrittr (>= 1.5),
psych (>= 1.5.8),
car (>= 2.1.1),
MASS (>= 7.3),
wordcloud (>= 2.5),
import (>= 1.1.0),
methods

Suggests testthat (>= 1.0.0),
covr (>= 1.2.0)

URL <https://github.com/radiant-rstats/radiant.multivariate>, <http://vnijs.github.io/radiant/>

BugReports <https://github.com/radiant-rstats/radiant.multivariate/issues>

License AGPL-3 | file LICENSE

LazyData true

RoxygenNote 5.0.1

R topics documented:

carpet	2
city	3
city2	3
clean_loadings	4
computer	4

conjoint	5
full_factor	6
hier_clus	7
kmeans_clus	8
mds	9
movie	10
mp3	10
plot.conjoint	11
plot.full_factor	11
plot.hier_clus	12
plot.kmeans_clus	13
plot.mds	14
plot.pmap	14
plot.pre_factor	15
pmap	16
pre_factor	17
radiant.multivariate	18
radiant.multivariate-deprecated	18
retailers	18
shopping	19
store.full_factor	19
store.kmeans_clus	20
summary.conjoint	21
summary.full_factor	21
summary.hier_clus	22
summary.kmeans_clus	23
summary.mds	24
summary.pmap	24
summary.pre_factor	25
the_table	26
toothpaste	27
tpbrands	27
Index	28

carpet	<i>Carpet cleaners</i>
--------	------------------------

Description

Carpet cleaners

Usage

data(carpet)

Format

A data frame with 18 rows and 5 variables

Details

Rankings reflect the evaluation of 18 alternative carpet cleaners by one respondent. Description provided in `attr(carpet,"description")`

city	<i>City distances</i>
------	-----------------------

Description

City distances

Usage

```
data(city)
```

Format

A data frame with 45 rows and 3 variables

Details

Distance in miles between nine cities in the USA. The dataset is used to illustrate multi-dimensional scaling (MDS). Description provided in `attr(city,"description")`

city2	<i>City distances 2</i>
-------	-------------------------

Description

City distances 2

Usage

```
data(city2)
```

Format

A data frame with 78 rows and 3 variables

Details

Distance in miles between 12 cities in the USA. The dataset is used to illustrate multi-dimensional scaling (MDS). Description provided in `attr(city2,"description")`

clean_loadings	<i>Sort and clean loadings</i>
----------------	--------------------------------

Description

Sort and clean loadings

Usage

```
clean_loadings(floadings, cutoff = 0, fsort = FALSE, dec = 8)
```

Arguments

floadings	Data frame with loadings
cutoff	Show only loadings with (absolute) values above cutoff (default = 0)
fsort	Sort factor loadings
dec	Number of decimals to show

Details

See http://vnijs.github.io/radiant/marketing/full_factor.html for an example in Radiant

Examples

```
result <- full_factor("diamonds",c("price","carat","table","x","y"))
clean_loadings(result$floadings, TRUE, .5, 2)
```

computer	<i>Perceptions of computer (re)sellers</i>
----------	--

Description

Perceptions of computer (re)sellers

Usage

```
data(computer)
```

Format

A data frame with 5 rows and 8 variables

Details

Perceptions of computer (re)sellers. The dataset is used to illustrate perceptual maps. Description provided in attr(computer,"description")

conjoint	<i>Conjoint analysis</i>
----------	--------------------------

Description

Conjoint analysis

Usage

```
conjoint(dataset, rvar, evar, reverse = FALSE, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
rvar	The response variable (e.g., profile ratings)
evar	Explanatory variables in the regression
reverse	Reverse the values of the response variable ('rvar')
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See <http://vnijs.github.io/radiant/marketing/conjoint.html> for an example in Radiant

Value

A list with all variables defined in the function as an object of class `conjoint`

See Also

[summary.conjoint](#) to summarize results

[plot.conjoint](#) to plot results

Examples

```
result <- conjoint("mp3", rvar = "Rating", evar = "Memory:Shape")
result <- mp3 %>% conjoint(rvar = "Rating", evar = "Memory:Shape")
```

full_factor	<i>Factor analysis (PCA)</i>
-------------	------------------------------

Description

Factor analysis (PCA)

Usage

```
full_factor(dataset, vars, method = "PCA", nr_fact = 1,
  rotation = "varimax", data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
vars	Variables to include in the analysis
method	Factor extraction method to use
nr_fact	Number of factors to extract
rotation	Apply varimax rotation or no rotation ("varimax" or "none")
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See http://vnijs.github.io/radiant/marketing/full_factor.html for an example in Radiant

Value

A list with all variables defined in the function as an object of class `full_factor`

See Also

`summary.full_factor` to summarize results

`plot.full_factor` to plot results

Examples

```
result <- full_factor("diamonds",c("price","carat","table","x","y"))
result <- full_factor("diamonds",c("price","carat","table","x","y"), method = "maxlik")
result <- diamonds %>% full_factor(c("price","carat","table","x","y"), method = "maxlik")
```

hier_clus*Hierarchical cluster analysis*

Description

Hierarchical cluster analysis

Usage

```
hier_clus(dataset, vars, distance = "sq.euclidian", method = "ward.D",  
  max_cases = 1000, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
vars	Vector of variables to include in the analysis
distance	Distance
method	Method
max_cases	Maximum number of cases allowed (default is 1000)
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See http://vnijs.github.io/radiant/marketing/hier_clus.html for an example in Radiant

Value

A list of all variables used in `hier_clus` as an object of class `hier_clus`

See Also

[summary.hier_clus](#) to summarize results

[plot.hier_clus](#) to plot results

Examples

```
result <- hier_clus("shopping", vars = "v1:v6")
```

kmeans_clus	<i>K-means cluster analysis</i>
-------------	---------------------------------

Description

K-means cluster analysis

Usage

```
kmeans_clus(dataset, vars, hc_init = TRUE, distance = "sq.euclidian",  
             method = "ward.D", seed = 1234, nr_clus = 2, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
vars	Vector of variables to include in the analysis
hc_init	Use centers from <code>hier_clus</code> as the starting point
distance	Distance for <code>hier_clus</code>
method	Method for <code>hier_clus</code>
seed	Random see to use for <code>kmeans</code> if <code>hc_init</code> is FALSE
nr_clus	Number of clusters to extract
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See http://vnijs.github.io/radiant/marketing/kmeans_clus.html for an example in Radiant

Value

A list of all variables used in `kmeans_clus` as an object of class `kmeans_clus`

See Also

[summary.kmeans_clus](#) to summarize results

[plot.kmeans_clus](#) to plot results

[store.kmeans_clus](#) to add cluster membership to the selected dataset

Examples

```
result <- kmeans_clus("shopping", c("v1:v6"))
```

mds	(Dis)similarity based brand maps (MDS)
-----	--

Description

(Dis)similarity based brand maps (MDS)

Usage

```
mds(dataset, id1, id2, dis, method = "metric", nr_dim = 2, seed = 1234,  
      data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
id1	A character variable or factor with unique entries
id2	A character variable or factor with unique entries
dis	A numeric measure of brand dissimilarity
method	Apply metric or non-metric MDS
nr_dim	Number of dimensions
seed	Random seed
data_filter	Expression entered in, e.g., <code>Data > View</code> to filter the dataset in Radiant. The expression should be a string (e.g., <code>"price > 10000"</code>)

Details

See <http://vnijs.github.io/radiant/marketing/mds.html> for an example in Radiant

Value

A list of all variables defined in the function as an object of class `mds`

See Also

[summary.mds](#) to summarize results

[plot.mds](#) to plot results

Examples

```
result <- mds("city", "from", "to", "distance")  
summary(result)  
result <- mds("diamonds", "clarity", "cut", "price")  
summary(result)
```

movie

Conjoint data for Movie theaters

Description

Conjoint data for Movie theaters

Usage

```
data(movie)
```

Format

A data frame with 18 rows and 6 variables

Details

Rankings reflect the evaluation of 18 alternative movie theaters by one respondent. Description provided in `attr(movie,"description")`

mp3

Conjoint data for MP3 players

Description

Conjoint data for MP3 players

Usage

```
data(mp3)
```

Format

A data frame with 18 rows and 6 variables

Details

Ratings reflect the evaluation of 18 alternative MP3 players by one respondent. Description provided in `attr(mp3,"description")`

plot.conjoint	<i>Plot method for the conjoint function</i>
---------------	--

Description

Plot method for the conjoint function

Usage

```
## S3 method for class 'conjoint'
plot(x, plots = "pw", scale_plot = FALSE,
      shiny = FALSE, ...)
```

Arguments

x	Return value from conjoint
plots	Show either the part-worth ("pw") or importance-weights ("iw") plot
scale_plot	Scale the axes of the part-worth plots to the same range
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/conjoint.html> for an example in Radiant

See Also

[conjoint](#) to generate results
[summary.conjoint](#) to summarize results

Examples

```
result <- conjoint(dataset = "mp3", rvar = "Rating", evar = "Memory:Shape")
plot(result, scale_plot = TRUE)
plot(result, plots = "iw")
```

plot.full_factor	<i>Plot method for the full_factor function</i>
------------------	---

Description

Plot method for the full_factor function

Usage

```
## S3 method for class 'full_factor'
plot(x, shiny = FALSE, ...)
```

Arguments

x	Return value from full_factor
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/full_factor.html for an example in Radiant

See Also

[full_factor](#) to calculate results
[plot.full_factor](#) to plot results

Examples

```
result <- full_factor("diamonds", c("price", "carat", "table"))
plot(result)
result <- full_factor("computer", "high_end:business")
summary(result)
```

plot.hier_clus	<i>Plot method for the hier_clus function</i>
----------------	---

Description

Plot method for the hier_clus function

Usage

```
## S3 method for class 'hier_clus'
plot(x, plots = c("scree", "change"), cutoff = 0.05,
     shiny = FALSE, ...)
```

Arguments

x	Return value from hier_clus
plots	Plots to return. "change" shows the percentage change in within-cluster heterogeneity as respondents are grouped into different number of clusters, "dendro" shows the dendrogram, "scree" shows a scree plot of within-cluster heterogeneity
cutoff	For large datasets plots can take time to render and become hard to interpret. By selection a cutoff point (e.g., 0.05 percent) the initial steps in hierarchical cluster analysis are removed from the plot
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/hier_clus.html for an example in Radiant

See Also

[hier_clus](#) to generate results

[summary.hier_clus](#) to summarize results

Examples

```
result <- hier_clus("shopping", vars = c("v1:v6"))
plot(result, plots = c("change", "scree"), cutoff = .05)
plot(result, plots = "dendro", cutoff = 0)
shopping %>% hier_clus(vars = c("v1:v6")) %>% plot
```

plot.kmeans_clus	<i>Plot method for kmeans_clus</i>
------------------	------------------------------------

Description

Plot method for kmeans_clus

Usage

```
## S3 method for class 'kmeans_clus'
plot(x, plots = "density", shiny = FALSE, ...)
```

Arguments

x	Return value from kmeans_clus
plots	One of "density", "bar", or "scatter")
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/kmeans_clus.html for an example in Radiant

See Also

[kmeans_clus](#) to generate results

[summary.kmeans_clus](#) to summarize results

[store.kmeans_clus](#) to add cluster membership to the selected dataset

Examples

```
result <- kmeans_clus("shopping", vars = c("v1:v6"))
plot(result)
shopping %>% kmeans_clus(, vars = c("v1:v6")) %>% plot
```

plot.mds	<i>Plot method for the mds function</i>
----------	---

Description

Plot method for the mds function

Usage

```
## S3 method for class 'mds'
plot(x, rev_dim = "", fontsz = 1.3, ...)
```

Arguments

x	Return value from mds
rev_dim	Flip the axes in plots
fontsz	Font size to use in plots
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/mds.html> for an example in Radiant

See Also

[mds](#) to calculate results
[summary.mds](#) to plot results

Examples

```
result <- mds("city","from","to","distance")
plot(result)
plot(result, rev_dim = 1:2)
plot(result, rev_dim = 1:2, fontsz = 2)
```

plot.pmap	<i>Plot method for the pmap function</i>
-----------	--

Description

Plot method for the pmap function

Usage

```
## S3 method for class 'pmap'
plot(x, plots = "", scaling = 2.1, fontsz = 1.3, ...)
```

Arguments

x	Return value from pmap
plots	Components to include in the plot ("brand", "attr"). If data on preferences is available use "pref" to add preference arrows to the plot
scaling	Arrow scaling in the brand map
fontsz	Font size to use in plots
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/pmap.html> for an example in Radiant

See Also

[pmap](#) to calculate results

[summary.pmap](#) to plot results

Examples

```
result <- pmap("computer", "brand", "high_end:business")
plot(result, plots = "brand")
plot(result, plots = c("brand", "attr"))
plot(result, plots = c("brand", "attr"))
plot(result, scaling = 1, plots = c("brand", "attr"))
result <- pmap("computer", "brand", "high_end:dated",
              pref = c("innovative", "business"))
plot(result, plots = c("brand", "attr", "pref"))
```

plot.pre_factor	<i>Plot method for the pre_factor function</i>
-----------------	--

Description

Plot method for the pre_factor function

Usage

```
## S3 method for class 'pre_factor'
plot(x, plots = c("scree", "change"), cutoff = 0.2,
     shiny = FALSE, ...)
```

Arguments

x	Return value from pre_factor
plots	Plots to return. "change" shows the change in eigenvalues as variables are grouped into different number of factors, "scree" shows a scree plot of eigenvalues

cutoff	For large datasets plots can take time to render and become hard to interpret. By selection a cutoff point (e.g., eigenvalues of .8 or higher) factors with the least explanatory power are removed from the plot
shiny	Did the function call originate inside a shiny app
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/pre_factor.html for an example in Radiant

See Also

[pre_factor](#) to calculate results

[summary.pre_factor](#) to summarize results

Examples

```
result <- pre_factor("diamonds", c("price", "carat", "table"))
plot(result)
plot(result, plots = c("change", "scree"), cutoff = .05)
```

pmap	<i>Attribute based brand maps</i>
------	-----------------------------------

Description

Attribute based brand maps

Usage

```
pmap(dataset, brand, attr, pref = "", nr_dim = 2, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
brand	A character variable with brand names
attr	Names of numeric variables
pref	Names of numeric brand preference measures
nr_dim	Number of dimensions
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See <http://vnijs.github.io/radiant/marketing/pmap.html> for an example in Radiant

Value

A list of all variables defined in the function as an object of class pmap

See Also

[summary.pmap](#) to summarize results

[plot.pmap](#) to plot results

Examples

```
result <- pmap("computer", "brand", "high_end:business")
```

pre_factor	<i>Evaluate if data are appropriate for PCA / Factor analysis</i>
------------	---

Description

Evaluate if data are appropriate for PCA / Factor analysis

Usage

```
pre_factor(dataset, vars, data_filter = "")
```

Arguments

dataset	Dataset name (string). This can be a dataframe in the global environment or an element in an <code>r_data</code> list from Radiant
vars	Variables to include in the analysis
data_filter	Expression entered in, e.g., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See http://vnijs.github.io/radiant/marketing/pre_factor.html for an example in Radiant

Value

A list with all variables defined in the function as an object of class pre_factor

See Also

[summary.pre_factor](#) to summarize results

[plot.pre_factor](#) to plot results

Examples

```
result <- pre_factor("diamonds", c("price", "carat", "table"))
```

radiant.multivariate	<i>radiant.multivariate</i>
----------------------	-----------------------------

Description

radiant.multivariate

Launch Radiant in the default browser

Usage

radiant.multivariate()

DetailsSee <http://vnijs.github.io/radiant> for documentation and tutorials

radiant.multivariate-deprecated

*Deprecated function(s) in the radiant.multivariate package***Description**

These functions are provided for compatibility with previous versions of radiant. They will eventually be removed.

Usage

save_factors(...)

Arguments

... Parameters to be passed to the updated functions

Details

save_factors is now a synonym for [store.full_factor](#)
 save_membership is now a synonym for [store.kmeans_clus](#)

retailers

*Perceptions of retailers***Description**

Perceptions of retailers

Usage

```
data(retailers)
```

Format

A data frame with 6 rows and 10 variables

Details

Consumer evaluations for a set of retailers in the Chicago area on 7 attributes. The dataset is used to illustrate perceptual maps. Description provided in `attr(retailers,"description")`

shopping	<i>Shopping attitudes</i>
----------	---------------------------

Description

Shopping attitudes

Usage

```
data(shopping)
```

Format

A data frame with 20 rows and 7 variables

Details

Attitudinal data on shopping for 20 consumers. Description provided in `attr(shopping,"description")`

store.full_factor	<i>Store factor scores to active dataset</i>
-------------------	--

Description

Store factor scores to active dataset

Usage

```
## S3 method for class 'full_factor'  
store(object, ..., name = "")
```

Arguments

object	Return value from <code>full_factor</code>
...	Additional arguments
name	Name of factor score variables

Details

See http://vnijs.github.io/radiant/marketing/full_factor.html for an example in Radiant

See Also

[full_factor](#) to generate results
[summary.full_factor](#) to summarize results
[plot.full_factor](#) to plot results

Examples

```
full_factor(shopping, "v1:v6", nr_fact = 3) %>%
  store %>%
  head
```

store.kmeans_clus	<i>Add a cluster membership variable to the active dataset</i>
-------------------	--

Description

Add a cluster membership variable to the active dataset

Usage

```
## S3 method for class 'kmeans_clus'
store(object, ..., name = "")
```

Arguments

object	Return value from kmeans_clus
...	Additional arguments
name	Name of cluster membership variable

Details

See http://vnijs.github.io/radiant/marketing/kmeans_clus.html for an example in Radiant

See Also

[kmeans_clus](#) to generate results
[summary.kmeans_clus](#) to summarize results
[plot.kmeans_clus](#) to plot results

Examples

```
kmeans_clus(shopping, vars = c("v1:v6")) %>% store %>% head
```

summary.conjoint	<i>Summary method for the conjoint function</i>
------------------	---

Description

Summary method for the conjoint function

Usage

```
## S3 method for class 'conjoint'
summary(object, mc_diag = FALSE, dec = 3, ...)
```

Arguments

object	Return value from conjoint
mc_diag	Shows multicollinearity diagnostics.
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/conjoint.html> for an example in Radiant

See Also

[conjoint](#) to generate results
[plot.conjoint](#) to plot results

Examples

```
result <- conjoint("mp3", rvar = "Rating", evar = "Memory:Shape")
summary(result, mc_diag = TRUE)
mp3 %>% conjoint(rvar = "Rating", evar = "Memory:Shape") %>% summary(., mc_diag = TRUE)
```

summary.full_factor	<i>Summary method for the full_factor function</i>
---------------------	--

Description

Summary method for the full_factor function

Usage

```
## S3 method for class 'full_factor'
summary(object, cutoff = 0, fsort = FALSE, dec = 2,
  ...)
```

Arguments

object	Return value from full_factor
cutoff	Show only loadings with (absolute) values above cutoff (default = 0)
fsort	Sort factor loadings
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/full_factor.html for an example in Radiant

See Also

[full_factor](#) to calculate results

[plot.full_factor](#) to plot results

Examples

```
result <- full_factor("diamonds",c("price","carat","depth","table","x"))
summary(result)
summary(result, cutoff = 0, fsort = FALSE)
summary(result, cutoff = 0, fsort = TRUE)
summary(result, cutoff = .5, fsort = TRUE)
diamonds %>% full_factor(c("price","carat","depth","table","x")) %>% summary
diamonds %>% full_factor(c("price","carat","depth","table","x")) %>% summary(cutoff = .5)
```

summary.hier_clus	<i>Summary method for the hier_clus function</i>
-------------------	--

Description

Summary method for the hier_clus function

Usage

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

Arguments

object	Return value from hier_clus
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/hier_clus.html for an example in Radiant

See Also

[hier_clus](#) to generate results
[plot.hier_clus](#) to plot results

Examples

```
result <- hier_clus("shopping", vars = c("v1:v6"))
summary(result)
```

summary.kmeans_clus	<i>Summary method for kmeans_clus</i>
---------------------	---------------------------------------

Description

Summary method for kmeans_clus

Usage

```
## S3 method for class 'kmeans_clus'
summary(object, dec = 2, ...)
```

Arguments

object	Return value from kmeans_clus
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/kmeans_clus.html for an example in Radiant

See Also

[kmeans_clus](#) to generate results
[plot.kmeans_clus](#) to plot results
[store.kmeans_clus](#) to add cluster membership to the selected dataset

Examples

```
result <- kmeans_clus("shopping", vars = c("v1:v6"))
summary(result)
shopping %>% kmeans_clus(vars = c("v1:v6"), nr_clus = 3) %>% summary
```

summary.mds*Summary method for the mds function*

Description

Summary method for the mds function

Usage

```
## S3 method for class 'mds'  
summary(object, dec = 2, ...)
```

Arguments

object	Return value from mds
dec	Rounding to use for output (default = 2). +1 used for stress measure
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/mds.html> for an example in Radiant

See Also

[mds](#) to calculate results
[plot.mds](#) to plot results

Examples

```
result <- mds("city", "from", "to", "distance")  
summary(result)  
summary(result, dec = 2)  
city %>% mds("from", "to", "distance") %>% summary
```

summary.pmap*Summary method for the pmap function*

Description

Summary method for the pmap function

Usage

```
## S3 method for class 'pmap'  
summary(object, cutoff = 0, dec = 2, ...)
```


Arguments

object	Return value from pmap
cutoff	Show only loadings with (absolute) values above cutoff (default = 0)
dec	Rounding to use for output
...	further arguments passed to or from other methods

Details

See <http://vnijs.github.io/radiant/marketing/pmap.html> for an example in Radiant

See Also

[pmap](#) to calculate results

[plot.pmap](#) to plot results

Examples

```
result <- pmap("computer", "brand", "high_end:business")
summary(result)
summary(result, cutoff = .3)
result <- pmap("computer", "brand", "high_end:dated", pref = c("innovative", "business"))
summary(result)
computer %>% pmap("brand", "high_end:dated", pref = c("innovative", "business")) %>%
  summary
```

summary.pre_factor	<i>Summary method for the pre_factor function</i>
--------------------	---

Description

Summary method for the pre_factor function

Usage

```
## S3 method for class 'pre_factor'
summary(object, dec = 2, ...)
```

Arguments

object	Return value from pre_factor
dec	Rounding to use for output
...	further arguments passed to or from other methods

Details

See http://vnijs.github.io/radiant/marketing/pre_factor.html for an example in Radiant

See Also

`pre_factor` to calculate results

`plot.pre_factor` to plot results

Examples

```
result <- pre_factor("diamonds", c("price", "carat", "table"))
summary(result)
diamonds %>% pre_factor(c("price", "carat", "table")) %>% summary
result <- pre_factor("computer", "high_end:business")
summary(result)
```

the_table

Function to calculate the PW and IW table for conjoint

Description

Function to calculate the PW and IW table for conjoint

Usage

```
the_table(model, dat, evar)
```

Arguments

model	Tidied model results (broom) output from <code>conjoint</code> passed on by <code>summary.conjoint</code>
dat	Conjoint data
evar	Explanatory variables used in the conjoint regression

Details

See <http://vnijs.github.io/radiant/marketing/conjoint.html> for an example in Radiant

See Also

`conjoint` to generate results

`summary.conjoint` to summarize results

`plot.conjoint` to plot results

Examples

```
result <- conjoint(dataset = "mp3", rvar = "Rating", evar = "Memory:Shape")
the_table(result$model, result$dat, result$evar)
```

toothpaste	<i>Toothpaste attitudes</i>
------------	-----------------------------

Description

Toothpaste attitudes

Usage

```
data(toothpaste)
```

Format

A data frame with 60 rows and 10 variables

Details

Attitudinal data on toothpaste for 60 consumers. Description provided in `attr(toothpaste,"description")`

tpbrands	<i>Toothpaste brands</i>
----------	--------------------------

Description

Toothpaste brands

Usage

```
data(tpbrands)
```

Format

A data frame with 45 rows and 4 variables

Details

Perceived (dis)similarity of a set of toothpaste brands. The dataset is used to illustrate multi-dimensional scaling (MDS). Description provided in `attr(tpbrands,"description")`

Index

*Topic **datasets**

- carpet, [2](#)
 - city, [3](#)
 - city2, [3](#)
 - computer, [4](#)
 - movie, [10](#)
 - mp3, [10](#)
 - retailers, [18](#)
 - shopping, [19](#)
 - toothpaste, [27](#)
 - tpbrands, [27](#)
- carpet, [2](#)
- city, [3](#)
- city2, [3](#)
- clean_loadings, [4](#)
- computer, [4](#)
- conjoint, [5](#), [11](#), [21](#), [26](#)
- full_factor, [6](#), [12](#), [19](#), [20](#), [22](#)
- hier_clus, [7](#), [12](#), [13](#), [22](#), [23](#)
- kmeans_clus, [8](#), [13](#), [20](#), [23](#)
- mds, [9](#), [14](#), [24](#)
- movie, [10](#)
- mp3, [10](#)
- plot.conjoint, [5](#), [11](#), [21](#), [26](#)
- plot.full_factor, [6](#), [11](#), [12](#), [20](#), [22](#)
- plot.hier_clus, [7](#), [12](#), [23](#)
- plot.kmeans_clus, [8](#), [13](#), [20](#), [23](#)
- plot.mds, [9](#), [14](#), [24](#)
- plot.pmap, [14](#), [17](#), [25](#)
- plot.pre_factor, [15](#), [17](#), [26](#)
- pmap, [15](#), [16](#), [25](#)
- pre_factor, [15](#), [16](#), [17](#), [25](#), [26](#)
- radiant.multivariate, [18](#)
- radiant.multivariate-deprecated, [18](#)
- radiant.multivariate-deprecated-package
(radiant.multivariate-deprecated),
[18](#)
- radiant.multivariate-package
(radiant.multivariate), [18](#)
- retailers, [18](#)
- save_factors
(radiant.multivariate-deprecated),
[18](#)
- save_membership
(radiant.multivariate-deprecated),
[18](#)
- shopping, [19](#)
- store.full_factor, [18](#), [19](#)
- store.kmeans_clus, [8](#), [13](#), [18](#), [20](#), [23](#)
- summary.conjoint, [5](#), [11](#), [21](#), [26](#)
- summary.full_factor, [6](#), [20](#), [21](#)
- summary.hier_clus, [7](#), [13](#), [22](#)
- summary.kmeans_clus, [8](#), [13](#), [20](#), [23](#)
- summary.mds, [9](#), [14](#), [24](#)
- summary.pmap, [15](#), [17](#), [24](#)
- summary.pre_factor, [16](#), [17](#), [25](#)
- the_table, [26](#)
- toothpaste, [27](#)
- tpbrands, [27](#)