

Package ‘radiant.multivariate’

July 18, 2017

Type Package

Title Multivariate Menu for Radiant: Business Analytics using R and Shiny

Version 0.8.7.1

Date 2017-7-7

Description The Radiant Multivariate menu includes interfaces for perceptual mapping, factor analysis, cluster analysis, and conjoint analysis. The application extends the functionality in radiant.data.

Depends R (>= 3.3.0),
radiant.data (>= 0.8.7.1)

Imports radiant.model (>= 0.8.7.1),
shiny (>= 1.0.3),
dplyr (>= 0.7.1),
rlang (>= 0.1.1),
ggplot2 (>= 2.0.0),
gridExtra (>= 2.0.0),
scales (>= 0.4.0),
magrittr (>= 1.5),
psych (>= 1.6.6),
GPArotation (>= 2014.11-1),
car (>= 2.1.1),
MASS (>= 7.3),
wordcloud (>= 2.5),
import (>= 1.1.0),
Gmedian (>= 1.2.3),
pryr (>= 0.1.2),
methods

Suggests testthat (>= 1.0.0)

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

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

License AGPL-3 | file LICENSE

LazyData true

RoxygenNote 6.0.1

R topics documented:

carpet	3
city	3
city2	4
clean_loadings	4
computer	5
conjoint	5
full_factor	6
hclus	7
kclus	8
mds	9
movie	10
mp3	10
plot.conjoint	11
plot.full_factor	12
plot.hclus	13
plot.kclus	14
plot.mds	15
plot.pmap	15
plot.pre_factor	16
pmap	17
predict.conjoint	18
predict_conjoint_by	19
pre_factor	20
print.conjoint.predict	20
radiant.multivariate	21
radiant.multivariate-deprecated	21
retailers	22
shopping	22
store.conjoint	23
store.conjoint.predict	23
store.conjoint.predict.by	24
store.full_factor	24
store.kclus	25
summary.conjoint	26
summary.full_factor	27
summary.hclus	28
summary.kclus	28
summary.mds	29
summary.pmap	30
summary.pre_factor	31
the_table	31
toothpaste	32
tpbrands	33

`carpet`*Carpet cleaners*

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`*City distances*

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 https://radiant-rstats.github.io/docs/multivariate/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, int = "", by = "none", 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
int	Interaction terms to include in the model
by	Variable to group data by before analysis (e.g., a respondent id)
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 <https://radiant-rstats.github.io/docs/multivariate/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 https://radiant-rstats.github.io/docs/multivariate/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")
```

hclus

*Hierarchical cluster analysis***Description**

Hierarchical cluster analysis

Usage

```
hclus(dataset, vars, distance = "sq.euclidian", method = "ward.D",
       max_cases = 5000, 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 <https://radiant-rstats.github.io/docs/multivariate/hclus.html> for an example in Radiant

Value

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

See Also

[summary.hclus](#) to summarize results

[plot.hclus](#) to plot results

Examples

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

kclus

K-clustering

Description

K-clustering

Usage

```
kclus(dataset, vars, fun = "mean", 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
fun	Function to use: "mean" or "median"
hc_init	Use centers from hclus as the starting point
distance	Distance for hclus
method	Method for hclus
seed	Random see to use for k-clustering 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 <https://radiant-rstats.github.io/docs/multivariate/kclus.html> for an example in Radiant

Value

A list of all variables used in kclus as an object of class kclus

See Also

`summary.kclus` to summarize results

`plot.kclus` to plot results

`store.kclus` to add cluster membership to the selected dataset

Examples

```
result <- kclus("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., Data > View to filter the dataset in Radiant. The expression should be a string (e.g., "price > 10000")

Details

See <https://radiant-rstats.github.io/docs/multivariate/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", show = "", scale_plot = FALSE,
     shiny = FALSE, custom = FALSE, ...)
```

Arguments

x	Return value from conjoint
plots	Show either the part-worth ("pw") or importance-weights ("iw") plot
show	Level in by variable to analyse (e.g., a specific respondent)
scale_plot	Scale the axes of the part-worth plots to the same range
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org/ for options.
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/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, custom = FALSE, ...)
```

Arguments

x	Return value from full_factor
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org/ for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/multivariate/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.hclus

*Plot method for the hclus function***Description**

Plot method for the hclus function

Usage

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

Arguments

x	Return value from hclus
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
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org/ for options.
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/hclus.html> for an example in Radiant

See Also

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

Examples

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

plot.kclus

*Plot method for kclus***Description**

Plot method for kclus

Usage

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

Arguments

x	Return value from kclus
plots	One of "density", "bar", or "scatter")
shiny	Did the function call originate inside a shiny app
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org/ for options.
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/kclus.html> for an example in Radiant

See Also

[kclus](#) to generate results

[summary.kclus](#) to summarize results

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

Examples

```
result <- kclus("shopping", vars = c("v1:v6"))
plot(result)
shopping %>% kclus(, 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 <https://radiant-rstats.github.io/docs/multivariate/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 <https://radiant-rstats.github.io/docs/multivariate/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, custom = 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
custom	Logical (TRUE, FALSE) to indicate if ggplot object (or list of ggplot objects) should be returned. This option can be used to customize plots (e.g., add a title, change x and y labels, etc.). See examples and http://docs.ggplot2.org/ for options.
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/multivariate/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

Attribute based brand maps

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 <https://radiant-rstats.github.io/docs/multivariate/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")
```

predict.conjoint	<i>Predict method for the conjoint function</i>
------------------	---

Description

Predict method for the conjoint function

Usage

```
## S3 method for class 'conjoint'
predict(object, pred_data = "", pred_cmd = "",
        conf_lev = 0.95, se = FALSE, dec = 3, ...)
```

Arguments

object	Return value from conjoint
pred_data	Name of the dataset to use for prediction
pred_cmd	Command used to generate data for prediction
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/conjoint.html> for an example in Radiant

See Also

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

Examples

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

predict_conjoint_by *Predict method for the conjoint function when a by variables is used*

Description

Predict method for the conjoint function when a by variables is used

Usage

```
predict_conjoint_by(object, pfun, pred_data = "", pred_cmd = "",
  conf_lev = 0.95, se = FALSE, dec = 3, ...)
```

Arguments

object	Return value from conjoint
pfun	Function to use for prediction
pred_data	Name of the dataset to use for prediction
pred_cmd	Command used to generate data for prediction
conf_lev	Confidence level used to estimate confidence intervals (.95 is the default)
se	Logical that indicates if prediction standard errors should be calculated (default = FALSE)
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/conjoint.html> for an example in Radiant

See Also

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

```
pre_factor
```

Evaluate if data are appropriate for PCA / Factor analysis

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 https://radiant-rstats.github.io/docs/multivariate/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"))
```

```
print.conjoint.predict
```

Print method for predict.conjoint

Description

Print method for predict.conjoint

Usage

```
## S3 method for class 'conjoint.predict'
print(x, ..., n = 50)
```

Arguments

x	Return value from prediction method
...	further arguments passed to or from other methods
n	Number of lines of prediction results to print. Use -1 to print all lines

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

Description

radiant.multivariate
Launch Radiant in the default browser

Usage

```
radiant.multivariate()
```

Details

See <https://radiant-rstats.github.io/docs> for documentation and tutorials

radiant.multivariate-deprecated	<i>Deprecated function(s) in the radiant.multivariate package</i>
---------------------------------	---

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.kclus](#)
 kmeans_clus is now a synonym for [kclus](#)
 hier_clus is now a synonym for [hclus](#)

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

Shopping attitudes

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.conjoint	<i>Store method for the Multivariate > Conjoint tab</i>
----------------	--

Description

Store method for the Multivariate > Conjoint tab

Usage

```
## S3 method for class 'conjoint'
store(object, name = "PWs", type = "PW",
      envir = parent.frame(), ...)
```

Arguments

object	Return value from conjoint
name	Name of the dataset to store
type	Type of output to store
envir	Environment to assign 'new' dataset (optional). Used when an r_data list is not available
...	further arguments passed to or from other methods

Details

Store data frame with PWs or IWs in Radiant r_data list if available

store.conjoint.predict	<i>Store predicted values generated in predict.conjoint</i>
------------------------	---

Description

Store predicted values generated in predict.conjoint

Usage

```
## S3 method for class 'conjoint.predict'
store(object, ..., data = attr(object,
  "pred_data"), name = "prediction")
```

Arguments

object	Return value from model predict function
...	Additional arguments
data	Data or dataset name (e.g., data = mtcars or data = "mtcars")
name	Variable name(s) assigned to predicted values

Details

See <https://radiant-rstats.github.io/docs/multivariate/conjoint.html> for an example in Radiant

```
store.conjoint.predict.by
```

Store method for the Multivariate > Conjoint > Predict

Description

Store method for the Multivariate > Conjoint > Predict

Usage

```
## S3 method for class 'conjoint.predict.by'
store(object, name = "predict_by",
      envir = parent.frame(), ...)
```

Arguments

object	Return value from predict.conjoint
name	Name of the dataset to store
envir	Environment to assign 'new' dataset (optional). Used when an r_data list is not available
...	further arguments passed to or from other methods

Details

Store data frame with predictions in Radiant r_data list if available

```
store.full_factor
```

Store factor scores to active dataset

Description

Store factor scores to active dataset

Usage

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

Arguments

object	Return value from full_factor
...	Additional arguments
name	Name of factor score variables

Details

See https://radiant-rstats.github.io/docs/multivariate/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.kclus

Add a cluster membership variable to the active dataset

Description

Add a cluster membership variable to the active dataset

Usage

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

Arguments

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

Details

See <https://radiant-rstats.github.io/docs/multivariate/kclus.html> for an example in Radiant

See Also

[kclus](#) to generate results

[summary.kclus](#) to summarize results

[plot.kclus](#) to plot results

Examples

```
kclus(shopping, vars = "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, show = "", mc_diag = FALSE,
        additional = FALSE, dec = 3, ...)
```

Arguments

object	Return value from conjoint
show	Level in by variable to analyse (e.g., a specific respondent)
mc_diag	Shows multicollinearity diagnostics.
additional	Show additional regression results
dec	Number of decimals to show
...	further arguments passed to or from other methods

Details

See <https://radiant-rstats.github.io/docs/multivariate/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 https://radiant-rstats.github.io/docs/multivariate/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.hclus	Summary method for the hclus function
---------------	---------------------------------------

Description

Summary method for the hclus function

Usage

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

Arguments

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

Details

See <https://radiant-rstats.github.io/docs/multivariate/hclus.html> for an example in Radiant

See Also

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

Examples

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

summary.kclus	Summary method for kclus
---------------	--------------------------

Description

Summary method for kclus

Usage

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

Arguments

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

Details

See <https://radiant-rstats.github.io/docs/multivariate/kclus.html> for an example in Radiant

See Also

[kclus](#) to generate results

[plot.kclus](#) to plot results

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

Examples

```
result <- kclus("shopping", vars = c("v1:v6"))
summary(result)
shopping %>% kclus(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 <https://radiant-rstats.github.io/docs/multivariate/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 <https://radiant-rstats.github.io/docs/multivariate/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 <code>pre_factor</code>
dec	Rounding to use for output
...	further arguments passed to or from other methods

Details

See https://radiant-rstats.github.io/docs/multivariate/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	<i>Function to calculate the PW and IW table for conjoint</i>
-----------	---

Description

Function to calculate the PW and IW table for conjoint

Usage

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

Arguments

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

Details

See <https://radiant-rstats.github.io/docs/multivariate/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(tidy(result$model_list[[1]][["model"]]), result$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, [3](#)
- city, [3](#)
- city2, [4](#)
- computer, [5](#)
- movie, [10](#)
- mp3, [10](#)
- retailers, [22](#)
- shopping, [22](#)
- toothpaste, [32](#)
- tpbrands, [33](#)

- carpet, [3](#)
- city, [3](#)
- city2, [4](#)
- clean_loadings, [4](#)
- computer, [5](#)
- conjoint, [5](#), [11](#), [18](#), [19](#), [26](#), [32](#)
- full_factor, [6](#), [12](#), [24](#), [25](#), [27](#)
- hclus, [7](#), [13](#), [21](#), [28](#)
- hier_clus
 - (radiant.multivariate-deprecated), [21](#)
- kclus, [8](#), [14](#), [21](#), [25](#), [28](#), [29](#)
- kmeans_clus
 - (radiant.multivariate-deprecated), [21](#)
- mds, [9](#), [15](#), [29](#)
- movie, [10](#)
- mp3, [10](#)
- plot.conjoint, [6](#), [11](#), [19](#), [26](#), [32](#)
- plot.full_factor, [6](#), [12](#), [12](#), [25](#), [27](#)
- plot.hclus, [7](#), [13](#), [28](#)
- plot.kclus, [8](#), [14](#), [25](#), [29](#)
- plot.mds, [9](#), [15](#), [29](#)
- plot.pmap, [15](#), [18](#), [30](#)
- plot.pre_factor, [16](#), [20](#), [31](#)
- pmap, [16](#), [17](#), [30](#)
- pre_factor, [16](#), [17](#), [20](#), [31](#)
- predict.conjoint, [18](#)
- predict.conjoint_by, [19](#)
- print.conjoint.predict, [20](#)
- radiant.multivariate, [21](#)
- radiant.multivariate-deprecated, [21](#)
- radiant.multivariate-deprecated-package
 - (radiant.multivariate-deprecated), [21](#)
- radiant.multivariate-package
 - (radiant.multivariate), [21](#)
- retailers, [22](#)
- save_factors
 - (radiant.multivariate-deprecated), [21](#)
- save_membership
 - (radiant.multivariate-deprecated), [21](#)
- shopping, [22](#)
- store.conjoint, [23](#)
- store.conjoint.predict, [23](#)
- store.conjoint.predict.by, [24](#)
- store.full_factor, [21](#), [24](#)
- store.kclus, [8](#), [14](#), [21](#), [25](#), [29](#)
- summary.conjoint, [6](#), [11](#), [19](#), [26](#), [32](#)
- summary.full_factor, [6](#), [25](#), [27](#)
- summary.hclus, [7](#), [13](#), [28](#)
- summary.kclus, [8](#), [14](#), [25](#), [28](#)
- summary.mds, [9](#), [15](#), [29](#)
- summary.pmap, [16](#), [18](#), [30](#)
- summary.pre_factor, [17](#), [20](#), [31](#)
- the_table, [31](#)
- toothpaste, [32](#)
- tpbrands, [33](#)