Package ‘radiant.multivariate’

October 14, 2022

Type Package

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

Version 1.4.4

Date 2022-7-22

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 (>= 3.4.0), radiant.data (>= 1.4.4)

Imports radiant.model (>= 1.4.1), shiny (>= 1.7.1), dplyr (>= 1.0.7), rlang (>= 0.4.10), ggplot2 (>= 2.2.1), scales (>= 0.4.0), magrittr (>= 1.5), psych (>= 1.8.4), GPArotation (>= 2014.11-1), car (>= 2.1.1), MASS (>= 7.3), import (>= 1.1.0), ggrepel (>= 0.8), lubridate (>= 1.7.4), polycor (>= 0.7.10), gower (>= 0.2.1), clustMixType (>= 0.2.1), patchwork (>= 1.0.0)

Suggests testthat (>= 2.0.0), pkgdown (>= 1.1.0)


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

License AGPL-3 | file LICENSE

LazyData true

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation no

Author Vincent Nijs [aut, cre]

Maintainer Vincent Nijs <radiant@rady.ucsd.edu>

Repository CRAN

Date/Publication 2022-07-23 06:20:02 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>carpet</td>
<td>3</td>
</tr>
<tr>
<td>city</td>
<td>3</td>
</tr>
<tr>
<td>city2</td>
<td>4</td>
</tr>
<tr>
<td>clean_loadings</td>
<td>4</td>
</tr>
<tr>
<td>computer</td>
<td>5</td>
</tr>
<tr>
<td>conjoint</td>
<td>5</td>
</tr>
<tr>
<td>full_factor</td>
<td>6</td>
</tr>
<tr>
<td>hclus</td>
<td>8</td>
</tr>
<tr>
<td>kclus</td>
<td>9</td>
</tr>
<tr>
<td>mds</td>
<td>10</td>
</tr>
<tr>
<td>movie</td>
<td>11</td>
</tr>
<tr>
<td>mp3</td>
<td>12</td>
</tr>
<tr>
<td>plot.conjoint</td>
<td>12</td>
</tr>
<tr>
<td>plot.full_factor</td>
<td>13</td>
</tr>
<tr>
<td>plot.hclus</td>
<td>14</td>
</tr>
<tr>
<td>plot.kclus</td>
<td>15</td>
</tr>
<tr>
<td>plot.mds</td>
<td>16</td>
</tr>
<tr>
<td>plot.pre_factor</td>
<td>17</td>
</tr>
<tr>
<td>plot.prmap</td>
<td>18</td>
</tr>
<tr>
<td>predict.conjoint</td>
<td>20</td>
</tr>
<tr>
<td>predict_conjoint_by</td>
<td>21</td>
</tr>
<tr>
<td>pre_factor</td>
<td>22</td>
</tr>
<tr>
<td>print.conjoint.predict</td>
<td>23</td>
</tr>
<tr>
<td>prmap</td>
<td>23</td>
</tr>
<tr>
<td>radiant.multivariate</td>
<td>24</td>
</tr>
<tr>
<td>radiant.multivariate_viewer</td>
<td>25</td>
</tr>
<tr>
<td>radiant.multivariate_window</td>
<td>26</td>
</tr>
<tr>
<td>retailers</td>
<td>26</td>
</tr>
<tr>
<td>shopping</td>
<td>27</td>
</tr>
<tr>
<td>store.conjoint</td>
<td>27</td>
</tr>
<tr>
<td>store.conjoint.predict</td>
<td>28</td>
</tr>
<tr>
<td>store.full_factor</td>
<td>28</td>
</tr>
<tr>
<td>store.hclus</td>
<td>29</td>
</tr>
<tr>
<td>store.kclus</td>
<td>30</td>
</tr>
<tr>
<td>summary.conjoint</td>
<td>31</td>
</tr>
<tr>
<td>summary.full_factor</td>
<td>32</td>
</tr>
<tr>
<td>summary.hclus</td>
<td>33</td>
</tr>
<tr>
<td>summary.kclus</td>
<td>33</td>
</tr>
<tr>
<td>summary.mds</td>
<td>34</td>
</tr>
<tr>
<td>summary.pre_factor</td>
<td>35</td>
</tr>
<tr>
<td>summary.prmap</td>
<td>36</td>
</tr>
<tr>
<td>the_table</td>
<td>37</td>
</tr>
<tr>
<td>toothpaste</td>
<td>37</td>
</tr>
<tr>
<td>tpbrands</td>
<td>38</td>
</tr>
</tbody>
</table>

Index: 39
carpet

<table>
<thead>
<tr>
<th>carpet</th>
<th>Carpet cleaners</th>
</tr>
</thead>
</table>

**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

<table>
<thead>
<tr>
<th>city</th>
<th>City distances</th>
</tr>
</thead>
</table>

**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  

**City distances 2**

**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  

**Sort and clean loadings**

**Description**
Sort and clean loadings

**Usage**
clean_loadings(floadings, cutoff = 0, fsort = FALSE, dec = 8, repl = NA)

**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
- `repl`: Replace loadings below the cutoff by NA (or "")

**Details**
See [https://radiant-rstats.github.io/docs/multivariate/full_factor.html](https://radiant-rstats.github.io/docs/multivariate/full_factor.html) for an example in Radiant
Examples

```r
result <- full_factor(shopping, "v1:v6", nr_fact = 2)
clean_loadings(result$floadings, fsort = TRUE, cutoff = .5, dec = 2)
```

---

**computer**  
*Perceptions of computer (re)sellers*

---

**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**  
*Conjoint analysis*

---

**Description**

Conjoint analysis

**Usage**

```r
conjoint(
  dataset,  
rvar,  
evar,  
int = "",  
by = "none",  
reverse = FALSE,  
data_filter = "",  
envir = parent.frame()
)
```
Arguments

- **dataset**: Dataset
- **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")
- **envir**: Environment to extract data from

Details

See [https://radiant-rstats.github.io/docs/multivariate/conjoint.html](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

```r
conjoint(mp3, rvar = "Rating", evar = "Memory:Shape") %>% str()
```

---

### full_factor

**Factor analysis (PCA)**

**Description**

Factor analysis (PCA)
full_factor

Usage

full_factor(
  dataset,
  vars,
  method = "PCA",
  hcor = FALSE,
  nr_fact = 1,
  rotation = "varimax",
  data_filter = "",
  envir = parent.frame()
)

Arguments

dataset       Dataset
vars          Variables to include in the analysis
method        Factor extraction method to use
hcor          Use polycor::hetcor to calculate the correlation matrix
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")
envir         Environment to extract data from

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

full_factor(shopping, "v1:v6") %>% str()
Hierarchical cluster analysis

Usage

```r
hclus(
  dataset, vars,
  labels = "none",
  distance = "sq.euclidian",
  method = "ward.D",
  max_cases = 5000,
  standardize = TRUE,
  data_filter = "",
  envir = parent.frame()
)
```

Arguments

- **dataset**: Dataset
- **vars**: Vector of variables to include in the analysis
- **labels**: A vector of labels for the leaves of the tree
- **distance**: Distance
- **method**: Method
- **max_cases**: Maximum number of cases allowed (default is 1000). Set to avoid long-running analysis in the radiant web-interface
- **standardize**: Standardized data (TRUE or FALSE)
- **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")
- **envir**: Environment to extract data from

Details

See [https://radiant-rstats.github.io/docs/multivariate/hclus.html](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
kclus

See Also

summary.hclus to summarize results
plot.hclus to plot results

Examples

hclus(shopping, vars = "v1:v6") %>% str()

kclus

K-clustering

Usage

kclus(
  dataset,
  vars,
  fun = "kmeans",
  hc_init = TRUE,
  distance = "sq.euclidian",
  method = "ward.D",
  seed = 1234,
  nr_clus = 2,
  standardize = TRUE,
  lambda = NULL,
  data_filter = "",
  envir = parent.frame()
)

Arguments

dataset [Dataset]  Dataset
vars [Vector of variables]  Vector of variables to include in the analysis
fun [Use either "kmeans" or "kproto" for clustering]
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 hc_init is FALSE]
nr_clus [Number of clusters to extract]
standardize [Standardize data (TRUE or FALSE)]
Parameter > 0 to trade off between Euclidean distance of numeric variables and simple matching coefficient between categorical variables. Also a vector of variable specific factors is possible where the order must correspond to the order of the variables in the data. In this case all variables’ distances will be multiplied by their corresponding lambda value.

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")

envir
Environment to extract data from

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
kclus(shopping, c("v1:v6"), nr_clus = 3) %>% str()

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 = "",
  envir = parent.frame()
)
### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataset</td>
<td>Dataset</td>
</tr>
<tr>
<td>id1</td>
<td>A character variable or factor with unique entries</td>
</tr>
<tr>
<td>id2</td>
<td>A character variable or factor with unique entries</td>
</tr>
<tr>
<td>dis</td>
<td>A numeric measure of brand dissimilarity</td>
</tr>
<tr>
<td>method</td>
<td>Apply metric or non-metric MDS</td>
</tr>
<tr>
<td>nr_dim</td>
<td>Number of dimensions</td>
</tr>
<tr>
<td>seed</td>
<td>Random seed</td>
</tr>
<tr>
<td>data_filter</td>
<td>Expression entered in, e.g., Data &gt; View to filter the dataset in Radiant. The expression should be a string (e.g., &quot;price &gt; 10000&quot;)</td>
</tr>
<tr>
<td>envir</td>
<td>Environment to extract data from</td>
</tr>
</tbody>
</table>

### 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

```r
mds(city, "from", "to", "distance") %>% str()
mds(diamonds, "clarity", "cut", "price") %>% str()
```

---

**movie**  
Conjoint data for Movie theaters

### Description

Conjoint data for Movie theaters

### Usage

```r
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

Plot method for the conjoint function

Description

Plot method for the conjoint function

Usage

```r
## 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 analyze (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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- **...**: further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/multivariate/conjoint.html](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

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

---

**plot.full_factor**

*Plot method for the full_factor function*

Description

Plot method for the full_factor function

Usage

```r
## S3 method for class 'full_factor'
plot(x, plots = "attr", shiny = FALSE, custom = FALSE, ...)
```
Arguments

- `x`: Return value from `full_factor`
- `plots`: Include attribute ("attr"), respondents ("resp") or both in 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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- `...`: further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/multivariate/full_factor.html](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

```r
result <- full_factor(shopping , "v1:v6", nr_fact = 2)
plot(result)
```

---

**plot.hclus**  
*Plot method for the hclus function*

Description

Plot method for the hclus function

Usage

```r
## 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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- **...**: further arguments passed to or from other methods.

Details

See [https://radiant-rstats.github.io/docs/multivariate/hclus.html](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

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

---

plot.kclus  

*Plot method for kclus*

Description

Plot method for kclus

Usage

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

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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- **...**: further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/multivariate/kclus.html](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

```r
result <- kclus(shopping, vars = "v1:v6", nr_clus = 3)
plot(result)
```

---

plot.mds

**Plot method for the mds function**

Description

Plot method for the mds function

Usage

```r
## S3 method for class 'mds'
plot(x, rev_dim = NULL, fontsize = 5, shiny = FALSE, custom = FALSE, ...)
```

Arguments

- **x**: Return value from `mds`
- **rev_dim**: Flip the axes in plots
- **fontsize**: Font size to use in plots
- **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 https://ggplot2.tidyverse.org/ for options.

... 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, fontsize = 7)
plot(result, rev_dim = 1:2)

plot.pre_factor  Plot method for the pre_factor function

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

\textbf{x} \hspace{1cm} \text{Return value from \texttt{pre_factor}}

\textbf{plots} \hspace{1cm} \text{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}

\textbf{cutoff} \hspace{1cm} \text{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}

\textbf{shiny} \hspace{1cm} \text{Did the function call originate inside a shiny app}

\textbf{custom} \hspace{1cm} \text{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 \url{https://ggplot2.tidyverse.org/} for options.}

... \hspace{1cm} \text{further arguments passed to or from other methods}

Details

See \url{https://radiant-rstats.github.io/docs/multivariate/pre_factor.html} for an example in Radiant

See Also

\texttt{pre_factor} to calculate results

\texttt{summary.pre_factor} to summarize results

Examples

```r
result <- pre_factor(shopping, "v1:v6")
plot(result, plots = c("change", "scree"), cutoff = .05)
```

\_\_\_\_\_\_\_\_\_

\textit{plot.prmap} \hspace{1cm} \textit{Plot method for the prmap function}

\_\_\_\_\_\_\_\_\_

Description

Plot method for the prmap function

Usage

```r
## S3 method for class 'prmap'
plot(
x, 
plots = "", 
scaling = 2, 
fontsz = 5,
```
Arguments

- **x**: Return value from `prmap`
- **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
- **fontsize**: Font size to use in plots
- **seed**: Random seed
- **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 [https://ggplot2.tidyverse.org/](https://ggplot2.tidyverse.org/) for options.
- **...**: further arguments passed to or from other methods

Details

See [https://radiant-rstats.github.io/docs/multivariate/prmap.html](https://radiant-rstats.github.io/docs/multivariate/prmap.html) for an example in Radiant

See Also

- `prmap` to calculate results
- `summary.prmap` to plot results

Examples

```r
result <- prmap(computer, brand = "brand", attr = "high_end:business")
plot(result, plots = "brand")
plot(result, plots = c("brand", "attr"))
plot(result, scaling = 1, plots = c("brand", "attr"))
prmap(
  retailers, brand = "retailer",
  attr = "good_value:cluttered",
 pref = c("segment1", "segment2")
) %>% plot(plots = c("brand", "attr", "pref"))
```
predict.conjoint 

Predict method for the conjoint function

Description

Predict method for the conjoint function

Usage

## S3 method for class 'conjoint'
predict(
  object,
  pred_data = NULL,
  pred_cmd = "",
  conf_lev = 0.95,
  se = FALSE,
  interval = "confidence",
  dec = 3,
  envir = parent.frame(),
  ...
)

Arguments

object 
Return value from conjoint

pred_data 
Provide the dataframe to generate predictions. The dataset must contain all columns used in the estimation

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)

interval 
Type of interval calculation ("confidence" or "prediction"). Set to "none" if se is FALSE

dec 
Number of decimals to show

envir 
Environment to extract data from

... 
further arguments passed to or from other methods

Details

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

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 = NULL,
  pred_cmd = "",
  conf_lev = 0.95,
  se = FALSE,
  dec = 3,
  envir = parent.frame(),
  ...
)

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
envir Environment to extract data from
... 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

预处理函数

Evaluate if data are appropriate for PCA / Factor analysis

Description

Evaluate if data are appropriate for PCA / Factor analysis

Usage

pre_factor(
  dataset, vars, hcor = FALSE, data_filter = "", envir = parent.frame()
)

Arguments

dataset  Dataset
vars  Variables to include in the analysis
hcor  Use polycor::hetcor to calculate the correlation matrix
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")
envir  Environment to extract data from

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

pre_factor(shopping, "v1:v6") %>% str()
Arguments

dataset          Dataset
brand            A character variable with brand names
attr             Names of numeric variables
pref             Names of numeric brand preference measures
nr_dim           Number of dimensions
hcor             Use polycor::hetcor to calculate the correlation matrix
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")
envir            Environment to extract data from

Details

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

Value

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

See Also

summary.prmap to summarize results
plot.prmap to plot results

Examples

prmap(computer, brand = "brand", attr = "high_end:business") %>% str()
radiant.multivariate_viewer

Details

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

Examples

```r
## Not run:
radiant.multivariate()

## End(Not run)
```

---

radiant.multivariate_viewer

*Launch radiant.multivariate in the Rstudio viewer*

Description

Launch radiant.multivariate in the Rstudio viewer

Usage

```r
radiant.multivariate_viewer(state, ...)
```

Arguments

- `state` Path to state file to load
- `...` additional arguments to pass to shiny::runApp (e.g., port = 8080)

Details

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

Examples

```r
## Not run:
radiant.multivariate_viewer()

## End(Not run)
```
radiant.multivariate_window

Launch radiant.multivariate in an Rstudio window

Description
Launch radiant.multivariate in an Rstudio window

Usage
radiant.multivariate_window(state, ...)

Arguments
state 
Path to state file to load
... 
additional arguments to pass to shiny::runApp (e.g, port = 8080)

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

Examples

```r
## Not run:
radiant.multivariate_window()
## End(Not run)
```

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

<table>
<thead>
<tr>
<th>shopping</th>
<th>Shopping attitudes</th>
</tr>
</thead>
</table>

**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")

<table>
<thead>
<tr>
<th>store.conjoint</th>
<th>Store method for the Multivariate &gt; Conjoint tab</th>
</tr>
</thead>
</table>

**Description**

Store method for the Multivariate > Conjoint tab

**Usage**

```r
## S3 method for class 'conjoint'
store(dataset, object, name, ...)
```

**Arguments**

- `dataset`: Dataset
- `object`: Return value from conjoint
- `name`: Variable name(s) assigned to predicted values
- `...`: 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

*Store predicted values generated in predict.conjoint*

**Description**

Store predicted values generated in predict.conjoint

**Usage**

```r
## S3 method for class 'conjoint.predict'
store(dataset, object, name = "prediction", ...)
```

**Arguments**

- `dataset`: Dataset to add predictions to
- `object`: Return value from model predict function
- `name`: Variable name(s) assigned to predicted values
- `...`: Additional arguments

**Details**

See [https://radiant-rstats.github.io/docs/multivariate/conjoint.html](https://radiant-rstats.github.io/docs/multivariate/conjoint.html) for an example in Radiant

**Examples**

```r
conjoint(mp3, rvar = "Rating", evar = "Memory:Shape") %>%
predict(mp3) %>%
store(mp3, ., name = "pred_pref")
```

---

store.full_factor

*Store factor scores to active dataset*

**Description**

Store factor scores to active dataset

**Usage**

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

Arguments

- dataset: Dataset to append to factor scores to
- object: Return value from `full_factor`
- name: Name of factor score variables
- ... Additional arguments

Details

See [https://radiant-rstats.github.io/docs/multivariate/full_factor.html](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

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

---

**store.hclus**  
*Add a cluster membership variable to the active dataset*

Description

Add a cluster membership variable to the active dataset

Usage

```r
## S3 method for class 'hclus'
store(dataset, object, nr_clus = 2, name = "", ...)
```

Arguments

- dataset: Dataset to append to cluster membership variable to
- object: Return value from `hclus`
- nr_clus: Number of clusters to extract
- name: Name of cluster membership variable
- ... Additional arguments
Details

See [https://radiant-rstats.github.io/docs/multivariate/hclus.html](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
- `plot.hclus` to plot results

Examples

```r
hclus(shopping, vars = "v1:v6") %>%
  store(shopping, ., nr_clus = 3) %>%
  head()
```
Examples

kclus(shopping, vars = "v1:v6", nr_clus = 3) %>%
store(shopping, .) %>%
head()

summary.conjoint

Summary method for the conjoint function

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 analyze (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)
Summary method for the full_factor function

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

```r
result <- full_factor(shopping, "v1:v6", nr_fact = 2)
summary(result)
summary(result, cutoff = .5, fsort = TRUE)
```
**summary.hclus**

Summary method for the hclus function

**Description**

Summary method for the hclus function

**Usage**

```r
## 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](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**

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

---

**summary.kclus**

Summary method for kclus

**Description**

Summary method for kclus

**Usage**

```r
## 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](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

```r
result <- kclus(shopping, vars = "v1:v6", nr_clus = 3)
summary(result)
```

---

**summary.mds**

Summary method for the `mds` function

Description

Summary method for the `mds` function

Usage

```r
## 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](https://radiant-rstats.github.io/docs/multivariate/mds.html) for an example in Radiant
Summary method for the pre_factor function

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 [https://radiant-rstats.github.io/docs/multivariate/pre_factor.html](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

```r
result <- pre_factor(shopping, "v1:v6")
summary(result)
`pre_factor(computer, "high_end:business") %>% summary()
```
Summary method for the prmap function

Usage

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

Arguments

- **object**: Return value from `prmap`
- **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/prmap.html](https://radiant-rstats.github.io/docs/multivariate/prmap.html) for an example in Radiant

See Also

- `prmap` to calculate results
- `plot.prmap` to plot results

Examples

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

Function to calculate the PW and IW table for conjoint

Description
Function to calculate the PW and IW table for conjoint

Usage
doctors_table(model, dataset, evar)

Arguments
model: Tidied model results (broom) output from conjoint passed on by summary.conjoint
dataset: Conjoint data
evar: 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(mp3, rvar = "Rating", evar = "Memory:Shape")
the_table(tidy(result$model_list[[1]][["model"]]), result$dataset, result$evar)

toothpaste

Toothpaste attitudes

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")

data(tpbrands)

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

* datasets
  carpet, 3
  city, 3
  city2, 4
  computer, 5
  movie, 11
  mp3, 12
  retailers, 26
  shopping, 27
  toothpaste, 37
  tpbrands, 38

  carpet, 3
  city, 3
  city2, 4
  clean_loadings, 4
  computer, 5
  conjoint, 5, 13, 20–22, 31, 37

  full_factor, 6, 14, 29, 32
  hclus, 8, 15, 29, 30, 33
  kclus, 9, 16, 30, 34
  mds, 10, 16, 17, 34, 35
  movie, 11
  mp3, 12

  plot.conjoint, 6, 12, 21, 22, 31, 37
  plot.full_factor, 7, 13, 14, 29, 32
  plot.hclus, 9, 14, 30, 33
  plot.kclus, 10, 15, 30, 34
  plot.mds, 11, 16, 35
  plot.pre_factor, 17, 23, 33
  plot.prmap, 18, 24, 36
  pre_factor, 18, 22, 35
  predict.conjoint, 20
  predict_conjoint_by, 21
  print.conjoint.predict, 23
  prmap, 19, 23, 36

  radiant.mutivariate, 24
  radiant.mutivariate_viewer, 25
  radiant.mutivariate_window, 26
  retailers, 26
  shopping, 27
  store.conjoint, 27
  store.conjoint.predict, 28
  store.full_factor, 28
  store.hclus, 29
  store.kclus, 10, 16, 30, 34
  summary.conjoint, 6, 13, 21, 22, 31, 37
  summary.full_factor, 7, 29, 32
  summary.hclus, 9, 15, 30, 33
  summary.kclus, 10, 16, 30, 33
  summary.mds, 11, 17, 34
  summary.pre_factor, 18, 23, 35
  summary.prmap, 19, 24, 36

  the_table, 37
  toothpaste, 37
  tpbrands, 38