

# Package: interca (via r-universe)

October 31, 2024

**Type** Package

**Title** Multiple Correspondence Analysis Based on Interpretive Coordinates

**Version** 0.1.2

**Date** 2023-01-19

**Description** Various functions and a Shiny app to enrich the results of Multiple Correspondence Analysis with interpretive axes and planes (see Moschidis, Markos, and Thanopoulos, 2022; <[doi:10.1108/ACI-07-2022-0191](https://doi.org/10.1108/ACI-07-2022-0191)>).

**License** GPL-3

**Depends** ggplot2, DT

**Imports** shiny, shinyFeedback, waiter, dplyr,ggrepel, magrittr, openxlsx, shinyWidgets,readr, readxl, FactoMineR, factoextra, rmarkdown

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Stratos Moschidis [aut, cre], Angelos Markos [aut]

**Maintainer** Stratos Moschidis <[smos@statistics.gr](mailto:smos@statistics.gr)>

**Date/Publication** 2023-01-23 07:40:02 UTC

**Repository** <https://intercapackage.r-universe.dev>

**RemoteUrl** <https://github.com/cran/interca>

**RemoteRef** HEAD

**RemoteSha** 197b1426810be061d8e1c8a0e452d6236443ba9a

## Contents

interca . . . . .	2
intercaShiny . . . . .	3
plot1d . . . . .	3
plot2d . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

interca	<i>interca</i>
---------	----------------

---

### Description

The function performs multiple correspondence analysis (MCA) on a given data set and a given number of categorical variables. The function returns for each point for each axis the coordinates, the interpretive coordinates, the contributions, and the quality of display.

### Usage

```
interca(data = data, numaxes = 10)
```

### Arguments

data	A data frame containing the data to be analyzed.
numaxes	The number of categorical variables that will be retained for the calculation of the elements referred to in the function description.

### Value

The function returns a list with the principal MCA coordinates (coords), the interpretive MCA coordinates (ecoords), the values of the CTR (ctr) and COR (cor), the labels of the variable categories (lbl) and the scree plot (plot).

### References

Moschidis, S., Markos, A., & Thanopoulos, A. C. (2022). "Automatic" interpretation of multiple correspondence analysis (MCA) results for nonexpert users, using R programming. *Applied Computing and Informatics*, (ahead-of-print).

### Examples

```
# Set the seed to ensure reproducibility
set.seed(123)
# Create three categorical variables
X1 <- sample(c("X1_1", "X1_2"), size = 200, replace = TRUE)
X2 <- sample(c("X2_1", "X2_2", "X2_3"), size = 200, replace = TRUE)
X3 <- sample(c("X3_1", "X3_2", "X3_3", "X3_4"), size = 200, replace = TRUE)
# the resulting data frame
df <- data.frame(cbind(X1,X2,X3))
# convert to factors
df$X1 <- factor(df$X1)
df$X2 <- factor(df$X2)
df$X3 <- factor(df$X3)

res <- interca(df, 5)
```

---

intercaShiny	<i>Runs the intercaShiny application</i>
--------------	--

---

**Description**

The function runs a Shiny app that allows the user to upload a data file and calculate the MCA output (tables, plots and ready-made reports) through a graphical interface. The data can be loaded in .xlsx or csv format and then the number of axes to be retained should be specified. The user can produce an interpretive axis or plane, filter points based on their interpretive coordinates, and download the plots and tables in .pdf (.xlsx) format, as well as a ready report with the interpretation of MPCA results.

**Usage**

```
intercaShiny()
```

**Value**

shiny application object

**Examples**

```
# --- Run the 'shiny' app  
## Not run: intercaShiny()
```

---

plot1d	<i>Creates interpretive axis</i>
--------	----------------------------------

---

**Description**

The function takes the result of the interca() function and the number of a factorial axis and returns the corresponding interpretive axis with the most significant points.

**Usage**

```
plot1d(res, dim = 1)
```

**Arguments**

res	The result of the execution of the interca() function.
dim	The number of the interpretive axis to be printed.

**Details**

The interca() function must be executed first in order to use the function.

**Value**

The function returns a ggplot2 scatterplot that corresponds to a single interpretive dimension obtained via `interca()`.

**See Also**

`interca`

**Examples**

```
# Set the seed to ensure reproducibility
set.seed(123)
# Create three categorical variables
X1 <- sample(c("X1_1", "X1_2"), size = 200, replace = TRUE)
X2 <- sample(c("X2_1", "X2_2", "X2_3"), size = 200, replace = TRUE)
X3 <- sample(c("X3_1", "X3_2", "X3_3", "X3_4"), size = 200, replace = TRUE)
# the resulting data frame
df <- data.frame(cbind(X1,X2,X3))
# convert to factors
df$X1 <- factor(df$X1)
df$X2 <- factor(df$X2)
df$X3 <- factor(df$X3)

res<-interca(df,5)
plot1d(res,1)
```

---

plot2d

*Interpretive 2D plane*

---

**Description**

The function takes as parameters the result of the `interca()` function, the interpretive axis that will form the x axis, and the interpretive axis that will form the y axis. The function returns the interpretive plane formed by the above.

**Usage**

```
plot2d(res, dim1 = 1, dim2 = 2)
```

**Arguments**

<code>res</code>	The result of the <code>interca()</code> function.
<code>dim1</code>	The number of the interpretive axis that will form the x axis of the interpretive plane.
<code>dim2</code>	The number of the interpretive axis that will form the y axis of the interpretive plane.

**Value**

The function returns a ggplot2 scatterplot that corresponds to a 2D interpretive plane obtained via `interca()`.

**See Also**

`interca`

**Examples**

```
#For the example, we will use the wg93 data frame from the ca package.
# Set the seed to ensure reproducibility
set.seed(123)
# Create three categorical variables
X1 <- sample(c("X1_1", "X1_2"), size = 200, replace = TRUE)
X2 <- sample(c("X2_1", "X2_2", "X2_3"), size = 200, replace = TRUE)
X3 <- sample(c("X3_1", "X3_2", "X3_3", "X3_4"), size = 200, replace = TRUE)
# the resulting data frame
df <- data.frame(cbind(X1,X2,X3))
# convert to factors
df$X1 <- factor(df$X1)
df$X2 <- factor(df$X2)
df$X3 <- factor(df$X3)

res <- interca(df,5)
plot2d(res,1,2)
```

# Index

interca, [2](#)  
intercaShiny, [3](#)

plot1d, [3](#)  
plot2d, [4](#)