



## Post, Search and Hire Faster

Be present at where the job candidates are. Enjoy more exposure when you post jobs with us

JobStreet Singapore

## Articles - Principal Component Methods in R: Practical Guide MCA - Multiple Correspondence Analysis in R: Essentials



Ads by Google

[Stop seeing this ad](#)

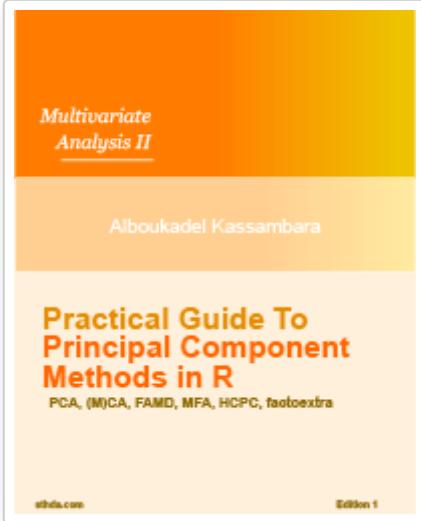
[Why this ad?](#)

- A group of individuals with similar profile in their answers to the questions
- The associations between variable categories

Previously, we described how to compute and interpret the simple correspondence analysis (chapter @ref(correspondence-analysis)). In the current chapter, we demonstrate how to compute and visualize multiple correspondence analysis in R software using *FactoMineR* (for the analysis) and *factoextra* (for data visualization). Additionally, we'll show how to reveal the most important variables that contribute the most in explaining the variations in the data set. We continue by explaining how to predict the results for supplementary individuals and variables. Finally, we'll demonstrate how to filter MCA results in order to keep only the most contributing variables.

Contents:

- Computation
  - R packages
  - Data format
  - Data summary
  - R code
- Visualization and interpretation
  - Eigenvalues / Variances
  - Biplot
  - Graph of variables
  - Graph of individuals
  - Color individuals by groups
  - Dimension description
- Supplementary elements
  - Definition and types
  - Specification in MCA
  - Results
  - Plots
- Filtering results
- Exporting results
  - Export plots to PDF/PNG files
  - Export results to txt/csv files
- ↖ • Summary

[Stop seeing this ad](#)[Why this ad?](#)

Practical Guide to Principal Component Methods in R

## Computation

### R packages

Several functions from different packages are available in the *R software* for computing multiple correspondence analysis. These functions/packages include:

- *MCA()* function [*FactoMineR* package]
- *dudi.mca()* function [*ade4* package]
- and *epMCA()* [*ExPosition* package]

No matter what function you decide to use, you can easily extract and visualize the MCA results using R functions provided in the *factoextra* R package.

[Stop seeing this ad](#)[Why this ad?](#)

Load the packages:

```
library("FactoMineR")
library("factoextra")
```

## Data format

We'll use the demo data sets `poison` available in *FactoMineR* package:

```
data(poison)
head(poison[, 1:7], 3)
```

```
##   Age Time   Sick Sex   Nausea Vomiting Abdominals
## 1   9   22 Sick_y   F Nausea_y  Vomit_n     Abdo_y
## 2   5    0 Sick_n   F Nausea_n  Vomit_n     Abdo_n
## 3   6   16 Sick_y   F Nausea_n  Vomit_y     Abdo_y
```

This data is a result from a survey carried out on children of primary school who suffered from food poisoning. They were asked about their symptoms and about what they ate.

The data contains 55 rows (individuals) and 15 columns (variables). We'll use only some of these individuals (children) and variables to perform the multiple correspondence analysis. The coordinates of the remaining individuals and variables on the factor map will be predicted from the previous MCA results.

In **MCA** terminology, our data contains :

- *Active individuals* (rows 1:55): Individuals that are used in the multiple correspondence analysis.
- *Active variables* (columns 5:15) : Variables that are used in the MCA.
- *Supplementary variables*: They don't participate to the MCA. The coordinates of these variables will be predicted.



Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

Subset only active individuals and variables for multiple correspondence analysis:

```
poison.active <- poison[1:55, 5:15]
head(poison.active[, 1:6], 3)
```

```
##      Nausea Vomiting Abdominals   Fever   Diarrhae   Potato
## 1 Nausea_y   Vomit_n     Abdo_y Fever_y Diarrhea_y Potato_y
## 2 Nausea_n   Vomit_n     Abdo_n Fever_n Diarrhea_n Potato_y
## 3 Nausea_n   Vomit_y     Abdo_y Fever_y Diarrhea_y Potato_y
```

## Data summary

The R base function `summary()` can be used to compute the frequency of variable categories. As the data table contains a large number of variables, we'll display only the results for the first 4 variables.

Statistical summaries:

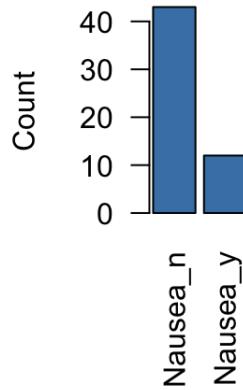
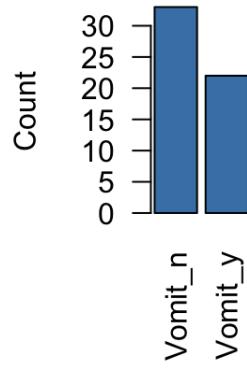
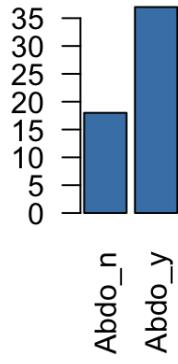
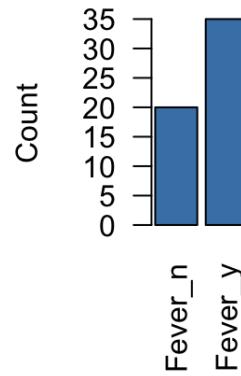
```
# Summary of the 4 first variables
summary(poison.active)[, 1:4]
```

```
##      Nausea      Vomiting    Abdominals      Fever
##  Nausea_n:43    Vomit_n:33    Abdo_n:18    Fever_n:20
##  Nausea_y:12    Vomit_y:22    Abdo_y:37    Fever_y:35
```

The `summary()` functions return the size of each variable category.

It's also possible to plot the frequency of variable categories. The R code below, plots the first 4 columns:

```
for (i in 1:4) {
  plot(poison.active[, i], main=colnames(poison.active)[i],
```

**Nausea****Vomiting****Abdominals****Fever**

The graphs above can be used to identify variable categories with a very low frequency. These types of variables can distort the analysis and should be removed.

## R code

The function *MCA()*[*FactoMiner* package] can be used. A simplified format is :

```
MCA(X, ncp = 5, graph = TRUE)
```

- **X**: a data frame with n rows (individuals) and p columns (categorical variables)
- **ncp**: number of dimensions kept in the final results.
- **graph**: a logical value. If TRUE a graph is displayed.

In the R code below, the MCA is performed only on the active individuals/variables :





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

```
print(res.mca)
```

```
## **Results of the Multiple Correspondence Analysis (MCA)**
## The analysis was performed on 55 individuals, described by 11 variables
## *The results are available in the following objects:
##
##      name            description
## 1  "$eig"          "eigenvalues"
## 2  "$var"           "results for the variables"
## 3  "$var$coord"    "coord. of the categories"
## 4  "$var$cos2"     "cos2 for the categories"
## 5  "$var$contrib"  "contributions of the categories"
## 6  "$var$v.test"   "v-test for the categories"
## 7  "$ind"           "results for the individuals"
## 8  "$ind$coord"    "coord. for the individuals"
## 9  "$ind$cos2"     "cos2 for the individuals"
## 10 "$ind$contrib"  "contributions of the individuals"
## 11 "$call"          "intermediate results"
## 12 "$call$marge.col" "weights of columns"
## 13 "$call$marge.li"  "weights of rows"
```

The object that is created using the function *MCA()* contains many information found in many different lists and matrices. These values are described in the next section.

## Visualization and interpretation

We'll use the *factoextra* R package to help in the interpretation and the visualization of the multiple correspondence analysis. No matter what function you decide to use [*FactoMiner::MCA()*, *ade4::dudi.mca()*], you can easily extract and visualize the results of multiple correspondence analysis using R functions provided in the *factoextra* R package.

These *factoextra* functions include:

[Stop seeing this ad](#)[Why this ad?](#)

In the next sections, we'll illustrate each of these functions.



Note that, the MCA results is interpreted as the results from a simple correspondence analysis (CA). Therefore, it is strongly recommended to read the interpretation of simple CA which has been comprehensively described in the Chapter @ref(correspondence-analysis).

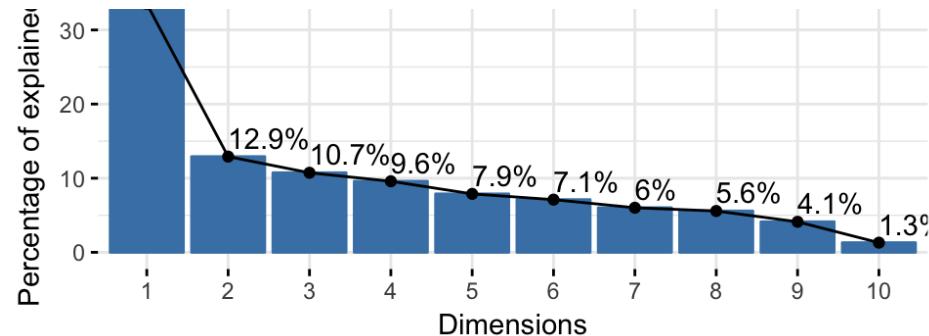
## Eigenvalues / Variances

The proportion of variances retained by the different dimensions (axes) can be extracted using the function `get_eigenvalue()` [*factoextra* package] as follow:

```
library("factoextra")
eig.val <- get_eigenvalue(res.mca)
# head(eig.val)
```

To visualize the percentages of inertia explained by each MCA dimensions, use the function `fviz_eig()` or `fviz_screeplot()` [*factoextra* package]:

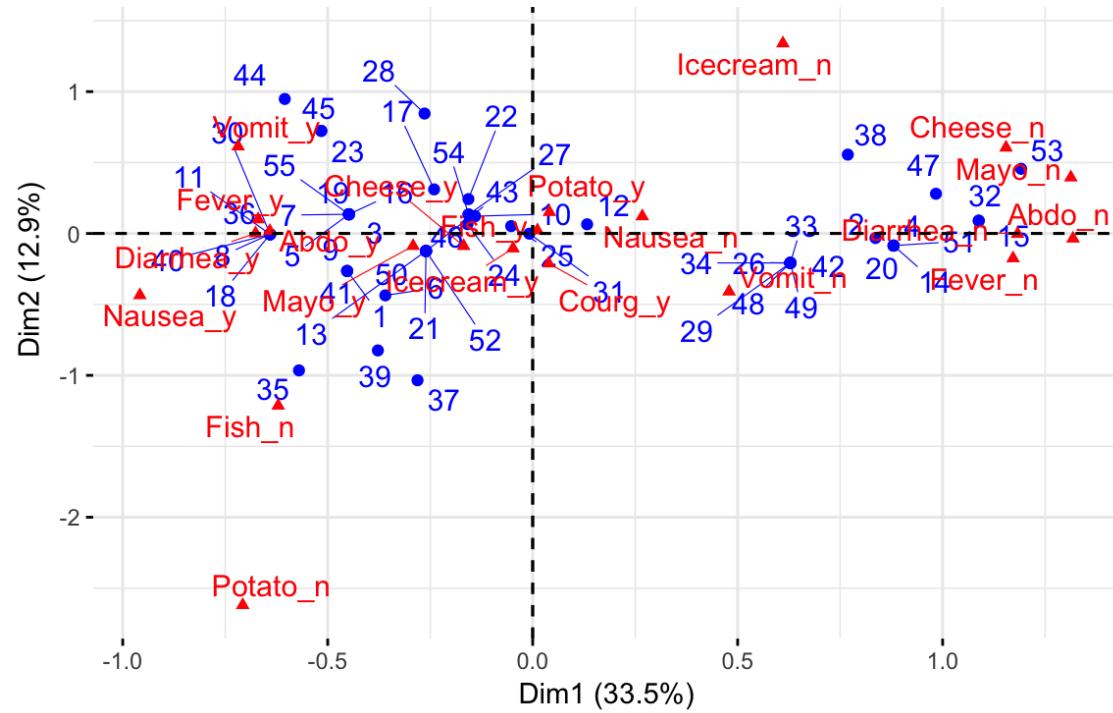
```
fviz_screeplot(res.mca, addlabels = TRUE, ylim = c(0, 45))
```

[Stop seeing this ad](#)[Why this ad?](#)

## Biplot

The function `fviz_mca_biplot()` [factoextra package] is used to draw the biplot of individuals and variable categories:

```
fviz_mca_biplot(res.mca,
  repel = TRUE, # Avoid text overlapping (slow if many point)
  ggtheme = theme_minimal())
```



The plot above shows a global pattern within the data. Rows (individuals) are represented by blue points and columns (variable categories) by red triangles.

The distance between any row points or column points gives a measure of their similarity (or dissimilarity). Row points with similar profile are closed on the factor map. The same holds true for column points.

## Graph of variables

## Results

The function `get_mca_var()` [in `factoextra`] is used to extract the results for variable categories. This function returns a list containing the coordinates, the cos2 and the contribution of variable categories:



[Stop seeing this ad](#)[Why this ad?](#)

```
## Multiple Correspondence Analysis Results for variables
## =====
## Name      Description
## 1 "$coord" "Coordinates for categories"
## 2 "$cos2"   "Cos2 for categories"
## 3 "$contrib" "contributions of categories"
```

The components of the `get_mca_var()` can be used in the plot of rows as follow:

- `var$coord`: coordinates of variables to create a scatter plot
- `var$cos2`: represents the quality of the representation for variables on the factor map.
- `var$contrib`: contains the contributions (in percentage) of the variables to the definition of the dimensions.

 Note that, it's possible to plot variable categories and to color them according to either i) their quality on the factor map (cos2) or ii) their contribution values to the definition of dimensions (contrib).

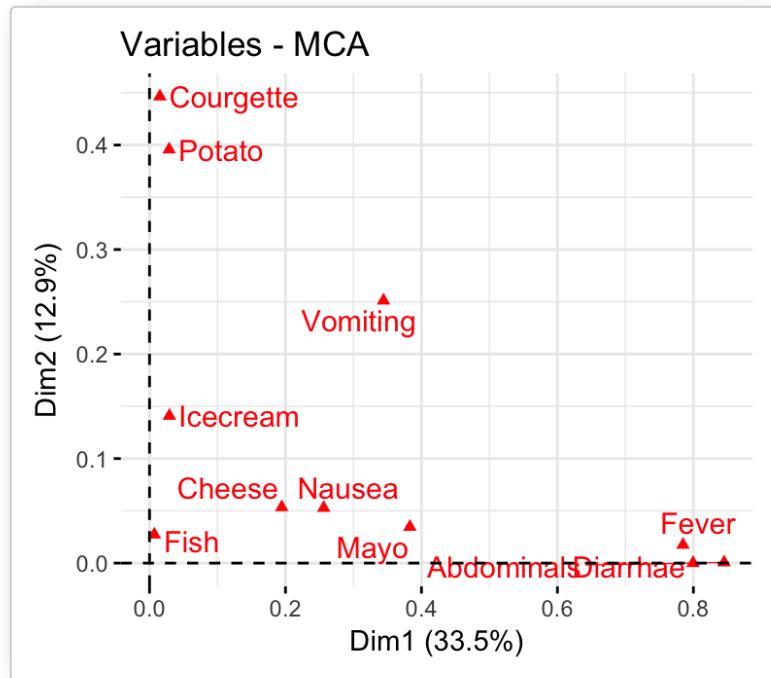
The different components can be accessed as follow:

```
# Coordinates
head(var$coord)
# Cos2: quality on the factor map
head(var$cos2)
# Contributions to the principal components
head(var$contrib)
```

In this section, we'll describe how to visualize variable categories only. Next, we'll highlight variable categories according to either i) their quality of representation on the factor map or ii) their contributions to the dimensions.

## Correlation between variables and principal dimensions

To visualize the correlation between variables and MCA principal dimensions, type this:

[Stop seeing this ad](#)[Why this ad?](#)

- The plot above helps to identify variables that are the most correlated with each dimension. The squared correlations between variables and the dimensions are used as coordinates.
- It can be seen that, the variables Diarrhoea, Abdominal and Fever are the most correlated with dimension 1. Similarly, the variables Courgette and Potato are the most correlated with dimension 2.

## Coordinates of variable categories

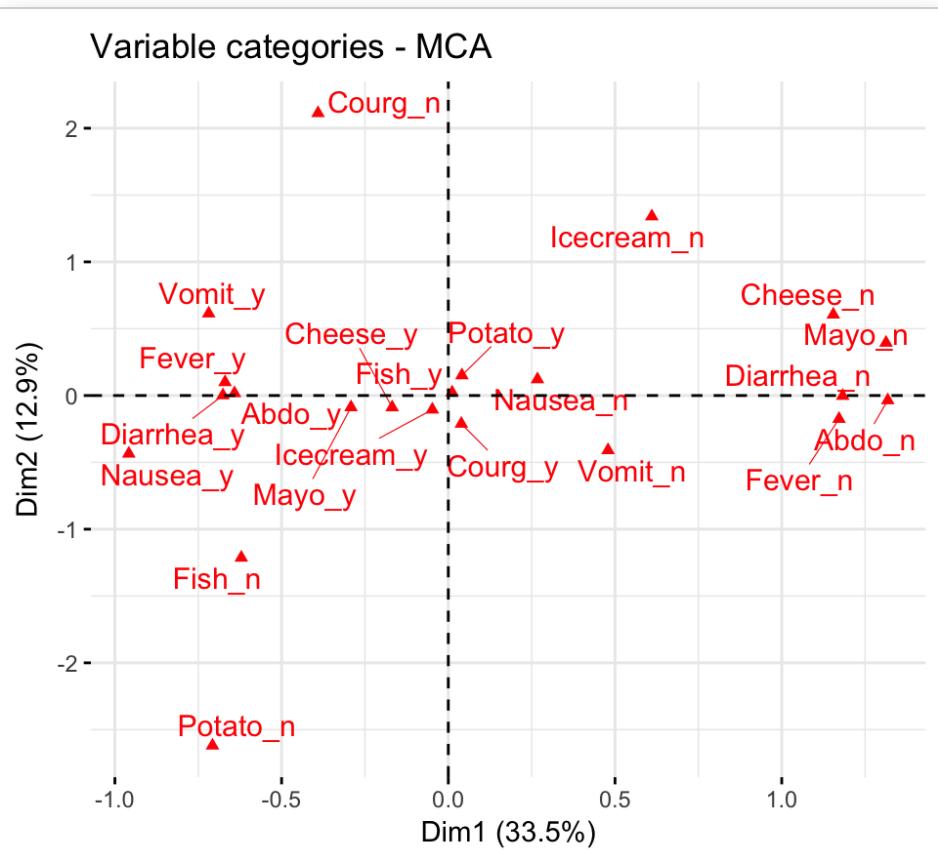
The R code below displays the coordinates of each variable categories in each dimension (1, 2 and 3):

```
head(round(var$coord, 2), 4)
```

```
## Vomit_y -0.72  0.61 -0.13 -0.41 -0.08
```

Use the function `fviz_mca_var()` [in `factoextra`] to visualize only variable categories:

```
fviz_mca_var(res.mca,
              repel = TRUE, # Avoid text overlapping (slow)
              ggtheme = theme_minimal())
```



[Stop seeing this ad](#)[Why this ad?](#)

The plot above shows the relationships between variable categories. It can be interpreted as follow:

- Variable categories with a similar profile are grouped together.
- Negatively correlated variable categories are positioned on opposite sides of the plot origin (opposed quadrants).
- The distance between category points and the origin measures the quality of the variable category on the factor map. Category points that are away from the origin are well represented on the factor map.

## Quality of representation of variable categories

The two dimensions 1 and 2 are sufficient to retain 46% of the total inertia (variation) contained in the data. Not all the points are equally well displayed in the two dimensions.

The quality of the representation is called the *squared cosine* (cos2), which measures the degree of association between variable categories and a particular axis. The cos2 of variable categories can be extracted as follow:

```
head(var$cos2, 4)
```

```
##           Dim 1   Dim 2   Dim 3   Dim 4   Dim 5
## Nausea_n  0.256  0.0528  0.2527  0.00408  0.01947
## Nausea_y  0.256  0.0528  0.2527  0.00408  0.01947
## Vomit_n   0.344  0.2512  0.0107  0.11229  0.00413
## Vomit_y   0.344  0.2512  0.0107  0.11229  0.00413
```

If a variable category is well represented by two dimensions, the sum of the cos2 is closed to one. For some of the row items, more than 2 dimensions are required to perfectly represent the data.

It's possible to color variable categories by their cos2 values using the argument `col.var = "cos2"`. This produces a gradient colors, which can be customized using the argument `gradient.cols`. For instance, `gradient.cols = c("white", "blue", "red")` means that:

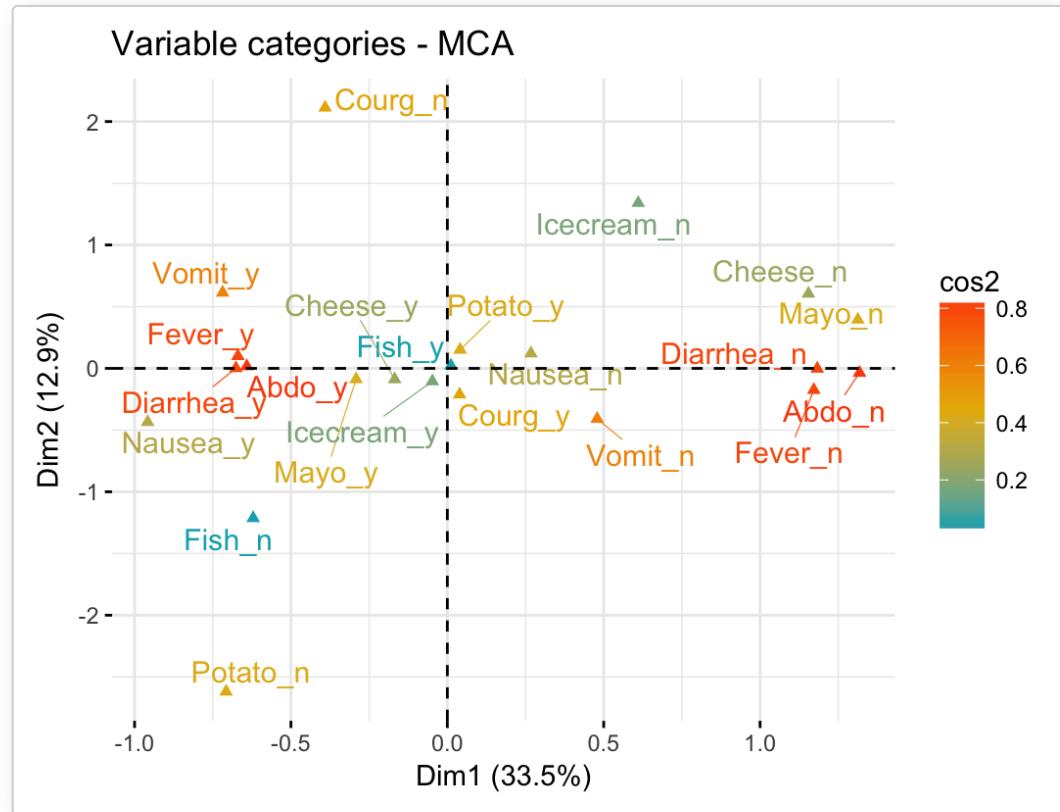
- variable categories with low cos2 values will be colored in "white"
- variable categories with mid cos2 values will be colored in "blue"



Ads by Google

[Stop seeing this ad](#) [Why this ad?](#)

```
gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
repel = TRUE, # Avoid text overlapping
ggtheme = theme_minimal()
```



Note that, it's also possible to change the transparency of the variable categories according to their cos2 values using the option `alpha.var = "cos2"`. For example, type this:

```
# Change the transparency by cos2 values
fviz_mca_var(res.mca, alpha.var="cos2",
^
```

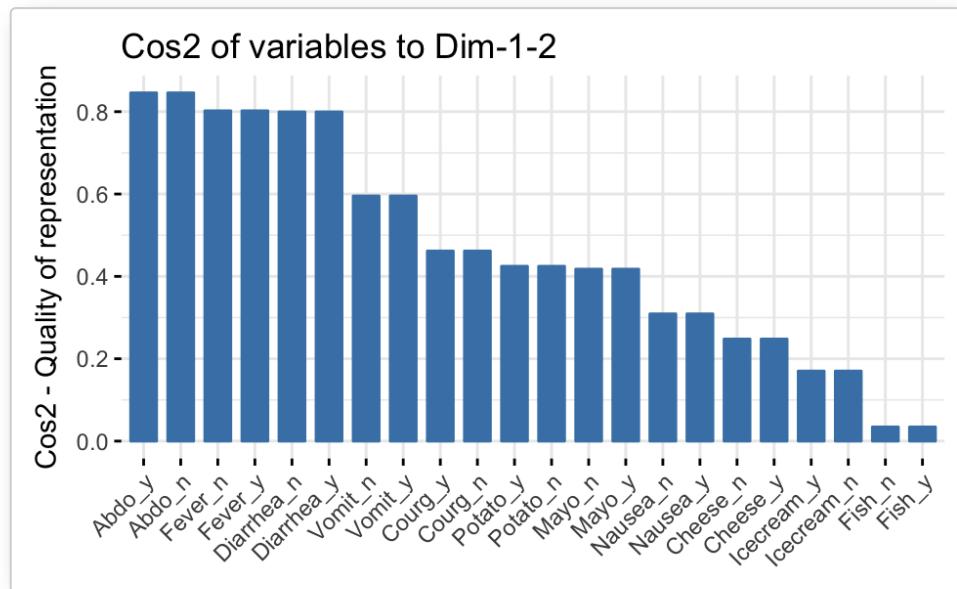
[Stop seeing this ad](#)[Why this ad?](#)

You can visualize the cos2 of row categories on all the dimensions using the corrplot package:

```
library("corrplot")
corrplot(var$cos2, is.corr=FALSE)
```

It's also possible to create a bar plot of variable cos2 using the function `fviz_cos2()` [in *factoextra*]:

```
# Cos2 of variable categories on Dim.1 and Dim.2
fviz_cos2(res.mca, choice = "var", axes = 1:2)
```



Note that, variable categories `Fish_n`, `Fish_y`, `Icecream_n` and `Icecream_y` are not very well represented by the first two dimensions. This implies that the position of the corresponding points on the scatter plot should be interpreted with some caution. A higher dimensional solution is probably necessary.

[Stop seeing this ad](#)[Why this ad?](#)

The contribution of the variable categories (in %) to the definition of the dimensions can be extracted as follow:

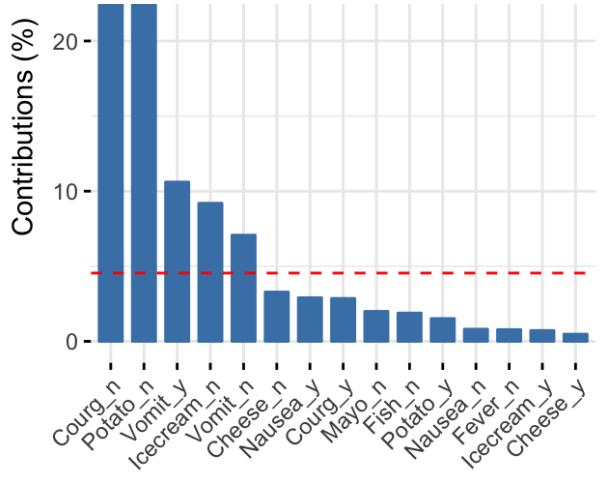
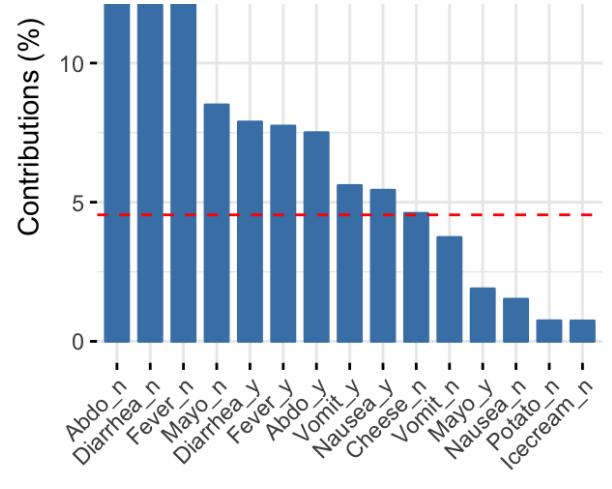
```
head(round(var$contrib, 2), 4)
```

```
##           Dim 1 Dim 2 Dim 3 Dim 4 Dim 5
## Nausea_n    1.52  0.81  4.67  0.08  0.49
## Nausea_y    5.43  2.91 16.73  0.30  1.76
## Vomit_n     3.73  7.07  0.36  4.26  0.19
## Vomit_y     5.60 10.61  0.54  6.39  0.29
```

 The variable categories with the larger value, contribute the most to the definition of the dimensions. Variable categories that contribute the most to Dim.1 and Dim.2 are the most important in explaining the variability in the data set.

The function `fviz_contrib()` [factoextra package] can be used to draw a bar plot of the contribution of variable categories. The R code below shows the top 15 variable categories contributing to the dimensions:

```
# Contributions of rows to dimension 1
fviz_contrib(res.mca, choice = "var", axes = 1, top = 15)
# Contributions of rows to dimension 2
fviz_contrib(res.mca, choice = "var", axes = 2, top = 15)
```

[Stop seeing this ad](#)[Why this ad?](#)

The total contributions to dimension 1 and 2 are obtained as follow:

```
# Total contribution to dimension 1 and 2
fviz_contrib(res.mca, choice = "var", axes = 1:2, top = 15)
```

The red dashed line on the graph above indicates the expected average value, If the contributions were uniform. The calculation of the expected contribution value, under null hypothesis, has been detailed in the principal component analysis chapter.

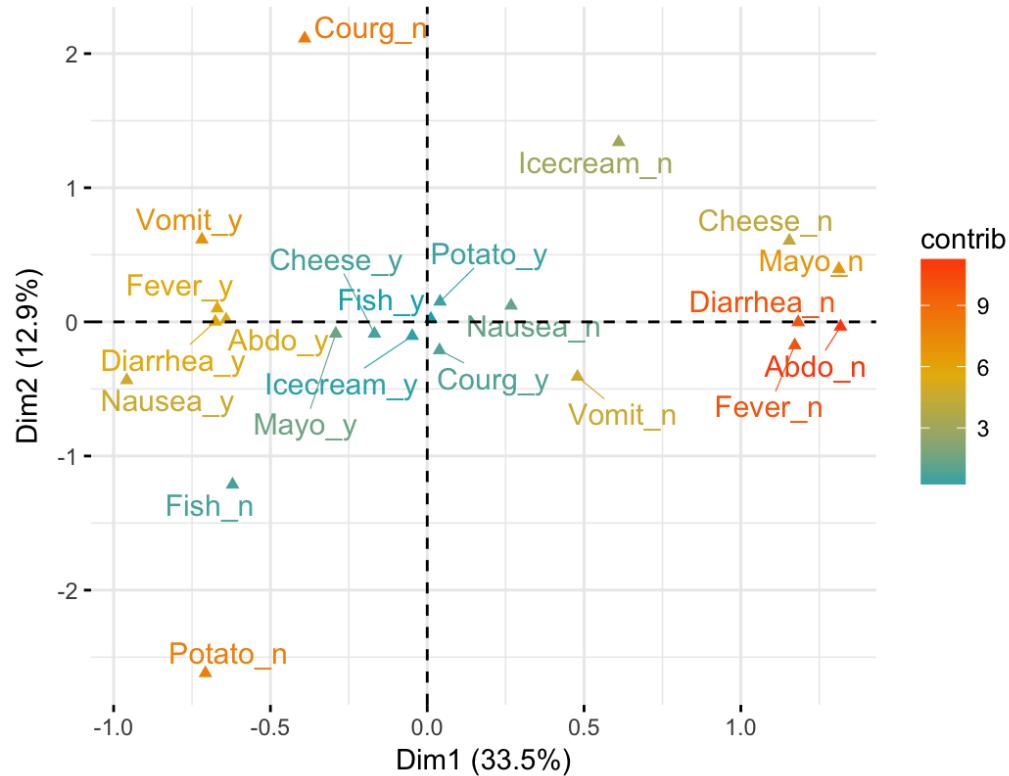
It can be seen that:

- the categories Abdo\_n, Diarrhea\_n, Fever\_n and Mayo\_n are the most important in the definition of the first dimension.
- The categories Cougn\_n, Potato\_n, Vomit\_y and Icecream\_n contribute the most to the dimension 2

The most important (or, contributing) variable categories can be highlighted on the scatter plot as follow:

```
fviz_mca_var(res.mca, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, # avoid text overlapping (slow)
```

## Variable categories - MCA



✓ The plot above gives an idea of what pole of the dimensions the categories are actually contributing to.

It is evident that the categories Abdo\_n, Diarrhea\_n, Fever\_n and Mayo\_n have an important contribution to the positive pole of the first dimension, while the categories Fever\_y and Diarrhea\_y have a major contribution to the negative pole of the first dimension; etc, ....

Note that, it's also possible to control the transparency of variable categories according to their contribution values using the option `alpha.var = "contrib"`. For example, type this:



Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

## Graph of individuals

### Results

The function `get_mca_ind()` [in `factoextra`] is used to extract the results for individuals. This function returns a list containing the coordinates, the cos2 and the contributions of individuals:

```
ind <- get_mca_ind(res.mca)
ind
```

```
## Multiple Correspondence Analysis Results for individuals
## =====
##   Name      Description
## 1 "$coord" "Coordinates for the individuals"
## 2 "$cos2"   "Cos2 for the individuals"
## 3 "$contrib" "contributions of the individuals"
```



The result for individuals gives the same information as described for variable categories. For this reason, I'll just displayed the result for individuals in this section without commenting.

To get access to the different components, use this:

```
# Coordinates of column points
head(ind$coord)
# Quality of representation
head(ind$cos2)
```



Ads by Google

[Stop seeing this ad](#)

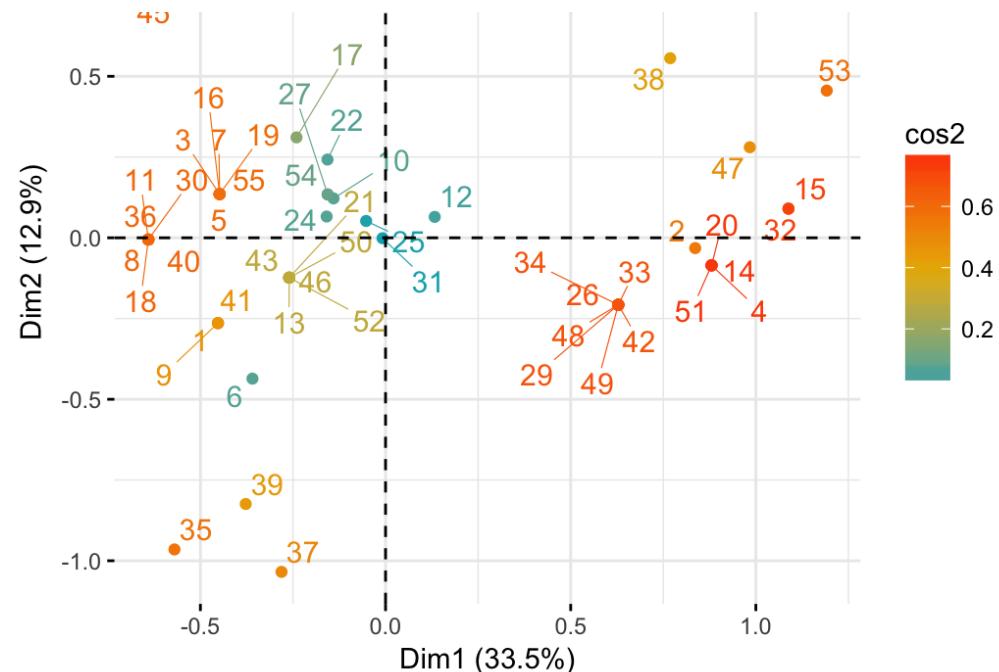
[Why this ad?](#)

## Plots: quality and contribution

The function `fviz_mca_ind()` [in *factoextra*] is used to visualize only individuals. Like variable categories, it's also possible to color individuals by their cos2 values:

```
fviz_mca_ind(res.mca, col.ind = "cos2",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE, # Avoid text overlapping (slow if many points)
              ggtheme = theme_minimal())
```





The R code below creates a bar plots of individuals cos2 and contributions:

```
# Cos2 of individuals
fviz_cos2(res.mca, choice = "ind", axes = 1:2, top = 20)
# Contribution of individuals to the dimensions
fviz_contrib(res.mca, choice = "ind", axes = 1:2, top = 20)
```

## Color individuals by groups



Note that, it's possible to color the individuals using any of the qualitative variables in the initial data table (poison)

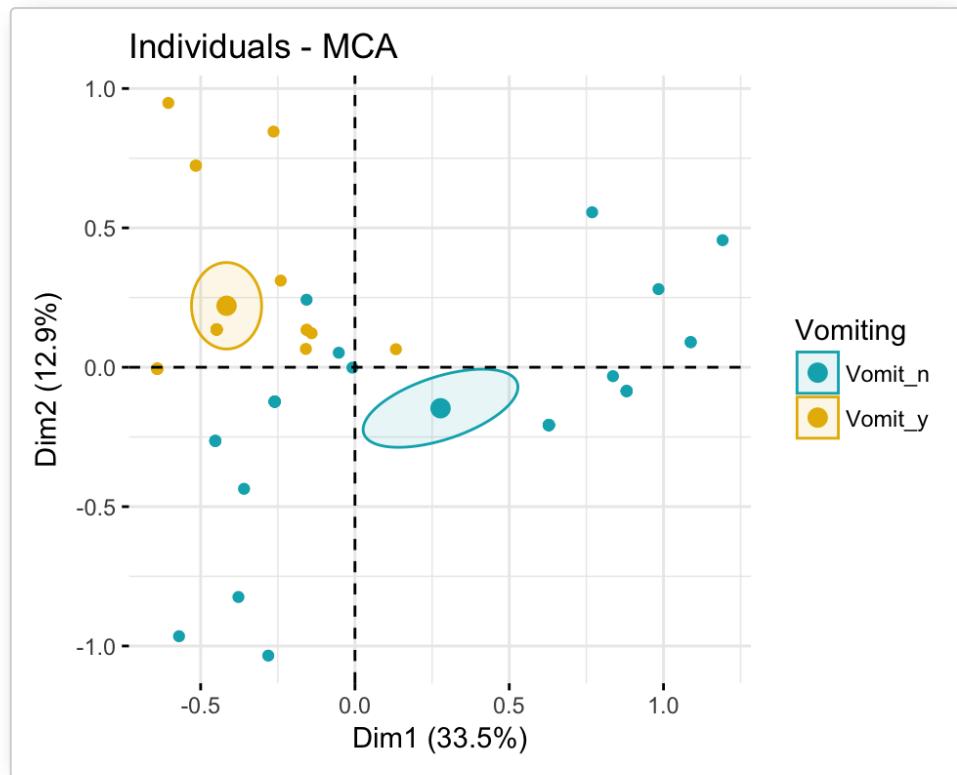




Ads by Google

[Stop seeing this ad](#)[Why this ad?](#)

```
fviz_mca_ind(res.mca,
  label = "none", # hide individual labels
  habillage = "Vomiting", # color by groups
  palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  ggtheme = theme_minimal())
```



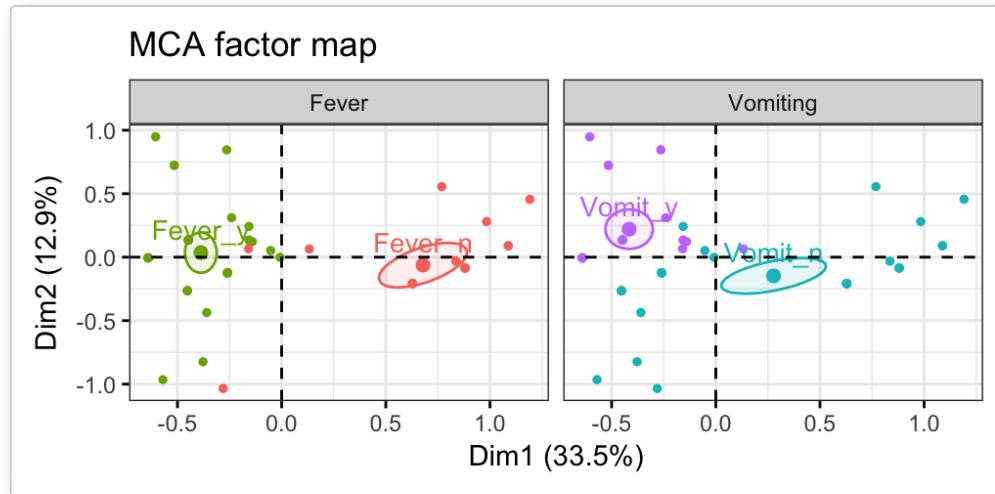
Note that, to specify the value of the argument `habillage`, it's also possible to use the index of the column as follow (`habillage = 2`). Additionally, you can provide an external grouping variable as follow: `habillage = poison$Vomiting`. For example:



[Stop seeing this ad](#)[Why this ad?](#)

If you want to color individuals using multiple categorical variables at the same time, use the function `fviz_ellipses()` [in *factoextra*] as follow:

```
fviz_ellipses(res.mca, c("Vomiting", "Fever"),
              geom = "point")
```



Alternatively, you can specify categorical variable indices:

```
fviz_ellipses(res.mca, 1:4, geom = "point")
```

## Dimension description

The function `dimdesc()` [in FactoMineR] can be used to identify the most correlated variables with a given dimension:



[Stop seeing this ad](#)[Why this ad?](#)

res.desc[[2]]

## Supplementary elements

### Definition and types

As described above (section @ref(mca-data-format)), the data set *poison* contains:

- *supplementary continuous variables* (quanti.sup = 1:2, columns 1 and 2 corresponding to the columns **age** and **time**, respectively)
- *supplementary qualitative variables* (quali.sup = 3:4, corresponding to the columns **sick** and **Sex**, respectively). This factor variables are used to color individuals by groups

The data doesn't contain *supplementary individuals*. However, for demonstration, we'll use the individuals 53:55 as supplementary individuals.



Supplementary variables and individuals are not used for the determination of the principal dimensions. Their coordinates are predicted using only the information provided by the performed multiple correspondence analysis on active variables/individuals.

### Specification in MCA

To specify supplementary individuals and variables, the function *MCA()* can be used as follow :

```
MCA(X, ind.sup = NULL, quanti.sup = NULL, quali.sup=NULL,
      graph = TRUE, axes = c(1,2))
```

- **X**: a data frame. Rows are individuals and columns are variables.
- **ind.sup**: a numeric vector specifying the indexes of the supplementary individuals.
- **quanti.sup, quali.sup**: a numeric vector specifying, respectively, the indexes of the quantitative and qualitative variables.
- **graph**: a logical value. If TRUE a graph is displayed.

```
quanti.sup = 1:2, quali.sup = 3:4, graph=FALSE)
```

## Results

The predicted results for supplementary individuals/variables can be extracted as follow:

```
# Supplementary qualitative variable categories  
res.mca$quali.sup  
# Supplementary quantitative variables  
res.mca$quanti  
# Supplementary individuals  
res.mca$ind.sup
```

## Plots

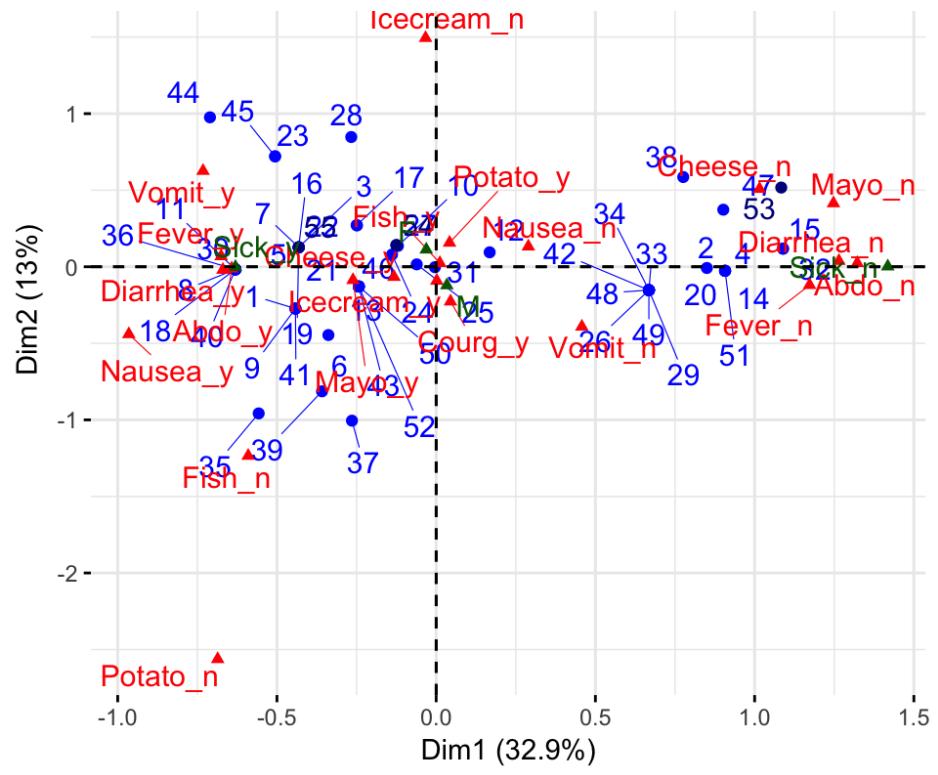
To make a biplot of individuals and variable categories, type this:

```
# Biplot of individuals and variable categories  
fviz_mca_biplot(res.mca, repel = TRUE,  
                  ggtheme = theme_minimal())
```



Stop seeing this ad

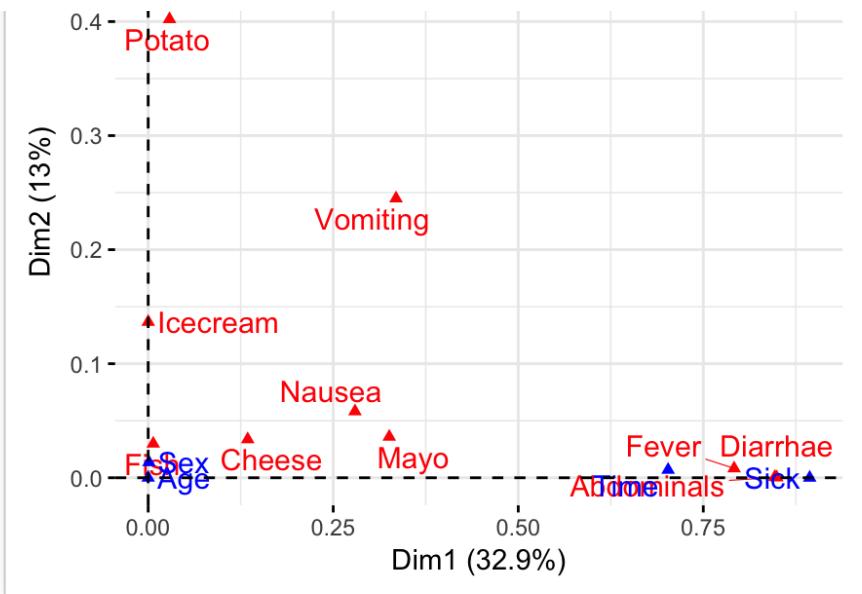
## Why this ad?



- Active individuals are in blue
  - Supplementary individuals are in darkblue
  - Active variable categories are in red
  - Supplementary variable categories are in darkgreen

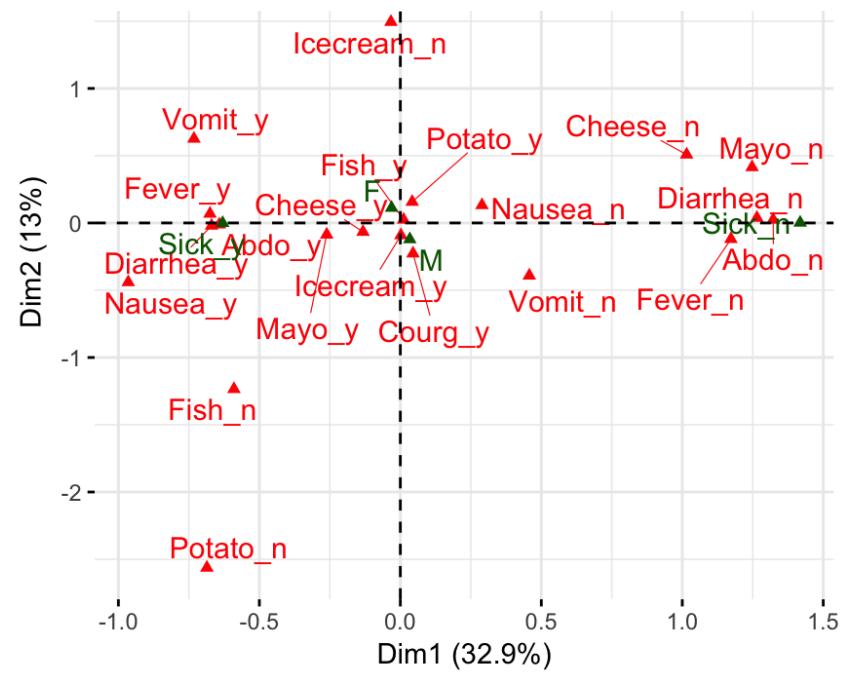
If you want to highlight the correlation between variables (active & supplementary) and dimensions, use the function `fviz_mca_var()` with the argument choice = "mca.cor":

```
fviz_mca_var(res.mca, choice = "mca.cor",  
             repel = TRUE)
```

[Stop seeing this ad](#)[Why this ad?](#)

The R code below plots qualitative variable categories (active & supplementary variables):

```
fviz_mca_var(res.mca, repel = TRUE,  
             ggtheme= theme_minimal())
```

[Stop seeing this ad](#)[Why this ad?](#)

For supplementary quantitative variables, type this:

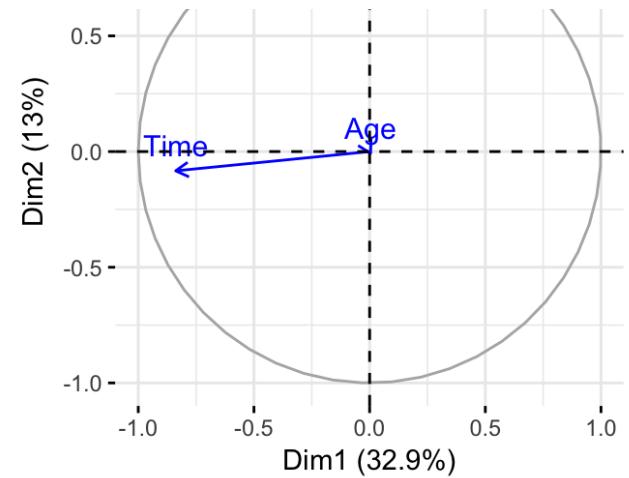
```
fviz_mca_var(res.mca, choice = "quanti.sup",
              ggtheme = theme_minimal())
```



Ads by Google

[Stop seeing this ad](#)

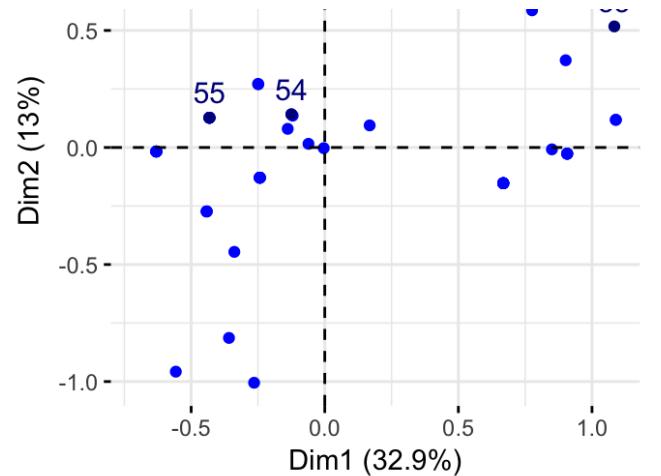
[Why this ad?](#)



To visualize supplementary individuals, type this:

```
fviz_mca_ind(res.mca,
              label = "ind.sup", #Show the label of ind.sup only
              ggtheme = theme_minimal())
```





## Filtering results

If you have many individuals/variable categories, it's possible to visualize only some of them using the arguments `select.ind` and `select.var`.

`select.ind, select.var`: a selection of individuals/variable categories to be drawn. Allowed values are `NULL` or a `list` containing the arguments `name`, `cos2` or `contrib`:

- `name`: is a character vector containing individuals/variable category names to be plotted
- `cos2`: if `cos2` is in  $[0, 1]$ , ex: 0.6, then individuals/variable categories with a  $\text{cos2} > 0.6$  are plotted
- `if cos2 > 1`, ex: 5, then the top 5 active individuals/variable categories and top 5 supplementary columns/rows with the highest `cos2` are plotted
- `contrib`: if `contrib > 1`, ex: 5, then the top 5 individuals/variable categories with the highest contributions are plotted

```
# Visualize variable categories with cos2 >= 0.4
fviz_mca_var(res.mca, select.var = list(cos2 = 0.4))
# Top 10 active variables with the highest cos2
fviz_mca_var(res.mca, select.var= list(cos2 = 10))
# Select by names
name <- list(name = c("Fever_n", "Abdo_y", "Diarrhea_n",
```

[Stop seeing this ad](#)[Why this ad?](#)

```
select.var = list(contrib = 0),  
ggtheme = theme_minimal())
```



When the selection is done according to the contribution values, supplementary individuals/variable categories are not shown because they don't contribute to the construction of the axes.

## Exporting results

### Export plots to PDF/PNG files

Two steps:

1. Create the plot of interest as an R object:

```
# Scree plot  
scree.plot <- fviz_eig(res.mca)  
# Biplot of row and column variables  
biplot.mca <- fviz_mca_biplot(res.mca)
```

2. Export the plots into a single pdf file as follow (one plot per page):

```
library(ggpubr)  
ggexport(plotlist = list(scree.plot, biplot.mca),  
        filename = "MCA.pdf")
```

[Stop seeing this ad](#)[Why this ad?](#)

```
# Export into a TXT file
write.infile(res.mca, "mca.txt", sep = "\t")
# Export into a CSV file
write.infile(res.mca, "mca.csv", sep = ";")
```

## Summary

In conclusion, we described how to perform and interpret multiple correspondence analysis (CA). We computed MCA using the *MCA()* function [FactoMineR package]. Next, we used the *factoextra* R package to produce ggplot2-based visualization of the CA results.

Other functions [packages] to compute MCA in R, include:

1. Using *dudi.acm()* [ade4]

```
library("ade4")
res.mca <- dudi.acm(poison.active, scannf = FALSE, nf = 5)
```

4. Using *epMCA()* [ExPosition]

```
library("ExPosition")
res.mca <- epMCA(poison.active, graph = FALSE, correction = "bg")
```

No matter what functions you decide to use, in the list above, the *factoextra* package can handle the output.

```
fviz_eig(res.mca)      # Scree plot
fviz_mca_biplot(res.mca) # Biplot of rows and columns
```





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

- Exploratory Multivariate Analysis by Example Using R (book) (Husson, Le, and Pagès 2017).
- Principal component analysis (article) (Abdi and Williams 2010). <https://goo.gl/1Vtwq1>.
- Correspondence analysis basics (blog post). <https://goo.gl/Xyk8KT>.

## References

Abdi, Hervé, and Lynne J. Williams. 2010. "Principal Component Analysis." *John Wiley and Sons, Inc. WIREs Comp Stat* 2: 433–59. <http://staff.ustc.edu.cn/~zwp/teach/MVA/abdi-awPCA2010.pdf>.

Husson, Francois, Sébastien Le, and Jérôme Pagès. 2017. *Exploratory Multivariate Analysis by Example Using R*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <http://factominer.free.fr/bookV2/index.html>.

Last update : 24/09/2017

1 Note



Enjoyed this article? Give us 5 stars (just above this text block)! Reader needs to be STHDA member for voting. I'd be very grateful if you'd help it spread by emailing it to a friend, or sharing it on Twitter, Facebook or Linked In.

Show me some love with the like buttons below... Thank you and please don't forget to share and comment below!!





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

Maximise Promotional  
bonus interest with  
Bonus\$aver Wealth  
Promotion. T&Cs apply.

Standard Chartered Bank

Share 69

Like 69

Tweet

[Share](#)

[Save](#)

Share

37





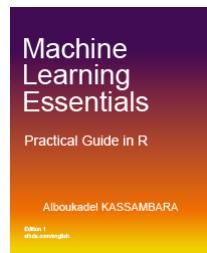
Ads by Google

[Stop seeing this ad](#)

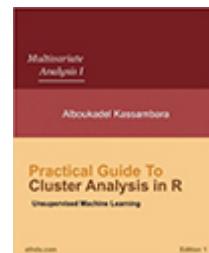
[Why this ad?](#)



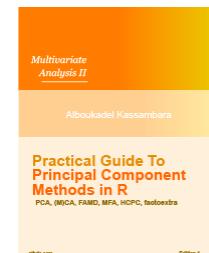
## Recommended for You!



[Machine Learning Essentials: Practical Guide in R](#)



[Practical Guide to Cluster Analysis in R](#)



[Practical Guide to Principal Component Methods in R](#)





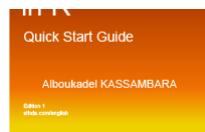
Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)



R Graphics Essentials for Great Data Visualization



Network Analysis and Visualization in R



More books on R and data science

## Recommended for you

This section contains best data science and self-development resources to help you on your path.

## Coursera - Online Courses and Specialization

### Data science

- Course: [Machine Learning: Master the Fundamentals](#) by Standford
- Specialization: [Data Science](#) by Johns Hopkins University
- Specialization: [Python for Everybody](#) by University of Michigan
- Courses: [Build Skills for a Top Job in any Industry](#) by Coursera
- Specialization: [Master Machine Learning Fundamentals](#) by University of Washington
- Specialization: [Statistics with R](#) by Duke University
- Specialization: [Software Development in R](#) by Johns Hopkins University
- Specialization: [Genomic Data Science](#) by Johns Hopkins University

### Popular Courses Launched in 2020

- [Google IT Automation with Python](#) by Google
- [AI for Medicine](#) by deeplearning.ai
- [Epidemiology in Public Health Practice](#) by Johns Hopkins University
- [AWS Fundamentals](#) by Amazon Web Services

### Trending Courses

- [The Science of Well-Being](#) by Yale University
- [Google IT Support Professional](#) by Google
- [Python for Everybody](#) by University of Michigan



Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

- [Graphic Design by Cal Arts](#)

## Books - Data Science

### Our Books

- [Practical Guide to Cluster Analysis in R](#) by A. Kassambara (Datanovia)
- [Practical Guide To Principal Component Methods in R](#) by A. Kassambara (Datanovia)
- [Machine Learning Essentials: Practical Guide in R](#) by A. Kassambara (Datanovia)
- [R Graphics Essentials for Great Data Visualization](#) by A. Kassambara (Datanovia)
- [GGPlot2 Essentials for Great Data Visualization in R](#) by A. Kassambara (Datanovia)
- [Network Analysis and Visualization in R](#) by A. Kassambara (Datanovia)
- [Practical Statistics in R for Comparing Groups: Numerical Variables](#) by A. Kassambara (Datanovia)
- [Inter-Rater Reliability Essentials: Practical Guide in R](#) by A. Kassambara (Datanovia)

### Others

- [R for Data Science: Import, Tidy, Transform, Visualize, and Model Data](#) by Hadley Wickham & Garrett Grolemund
- [Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems](#) by Aurelien Géron
- [Practical Statistics for Data Scientists: 50 Essential Concepts](#) by Peter Bruce & Andrew Bruce
- [Hands-On Programming with R: Write Your Own Functions And Simulations](#) by Garrett Grolemund & Hadley Wickham
- [An Introduction to Statistical Learning: with Applications in R](#) by Gareth James et al.
- [Deep Learning with R](#) by François Chollet & J.J. Allaire
- [Deep Learning with Python](#) by François Chollet



You are not authorized to post a comment



**Kimosh** 04/16/2021 at 17h15

Member

Loved this tutorial... very thorough!



#913



## Ads by Google

[Stop seeing this ad](#)[Why this ad?](#)

Thanks for the tutorial. However, I am wondering whether all the information on this page still accurate is. I have no problem with the basic results and plots but when I try to use col.var="contrib" or col.var="cos2", I get an error: invalid color name 'contrib'. I also get issues using select.var or gradient.cols: I get told that those are not graphical parameters. Am I missing something?

[#849](#)

Visitor 03/11/2019 at 15h59

Visitor

Hi

I really appreciate the nice article. I have a question. How can I get rid of the column name associated with my biplot for variables. My dataframe has only categorical variables.

For example the Column Country has levels c("Austria", "Germany", Switzerland") But in the MCA biplot I want Austria, Germany and Switzerland instead of Country\_Austria, Country\_Germany and so on.

thanks and best regards

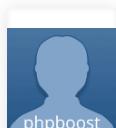
Dev

[#712](#)

Marcelo C Anjos 01/29/2019 at 21h04

Visitor

**Thank you so much for it Sr. !** This tutorial gave me a big help that I need to learn MCA using R. Your help has saved me many hours of work.

[#678](#)

Visitor 11/22/2018 at 12h25

Visitor





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)



Visitor 11/22/2018 at 12h23

Visitor

Thanks a lot very nice, easy and completed tutorial.

I have a trouble though, with the MCA function when I want to put some supplementary individuals that are 5 observations that correspond to one "Code" variable with the same modality "IDEAL". (Because I'd like to point the ideal profile in my plot). Notice that only those individuals have this particular "IDEAL" modality.

Well I got this error:

Error in plot.window(...) : need finite 'xlim' values

If I leave one of the individual of the category of modality it Works!!! But I don't want my model be missconstructed with this value.

Thanks for your help

Sophie

[#648](#)



Ding 10/19/2018 at 23h38

Visitor

Hi,

Thanks for the useful tutorials and tools. I tried to do the biplot of Dim1-Dim2 & Dim2-Dim3, each with the supplementary quantitative variables. One thing didn't make sense to me as the coordinates of vectors (quantitative variables) are identical on Dim1-Dim2 & Dim2-Dim3 biplots. It seems that the prediction of supplementary quantitative variables was done on Dim1-Dim2 only?

Thanks again for the help!

[#634](#)



Nira 09/18/2018 at 10h49

Visitor



Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

best regards,

Nira

[#609](#)



**kassambara** 05/19/2018 at 14h51

Administrator

Note that, numeric matrix is also accepted by the function MCA() [FactoMineR] package

[#480](#)



Se-Kang Kim 05/06/2018 at 13h36

Visitor

Hi there,

I've enjoyed your posted explanations about MCA, but the data format for MCA vs. mjca (from the "ca" package in R) seem different. Your data are in a text format. How can I analyze the numeric/indicator data matrix as done in "mjca" in your MCA paradigm?

Best,  
Se-Kang

[#454](#)

[Sign in](#)

[Login](#)





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

### Auto connect



[Sign in](#)

[Register](#)

[Forgotten password](#)

## Welcome!

Want to Learn More on R Programming and Data Science?

Follow us by Email

[Subscribe](#)

by [FeedBurner](#)

on Social Networks

[Click to see our collection of resources to help you on your path...](#)

## Course & Specialization

Recommended for You (on Coursera):

- [Course: Machine Learning: Master the Fundamentals](#)
- [Specialization: Data Science](#)



Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

- [Specialization: Software Development](#)
- Specialization: Genomic Data Science

[See More Resources](#)





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)

Ad veeam.com More ▼

 **factoextra**

 **survminer**

 **ggpubr**

 **ggcorrplot**

 **fastqcr**

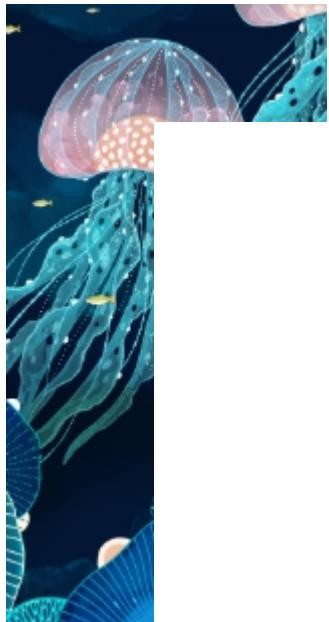




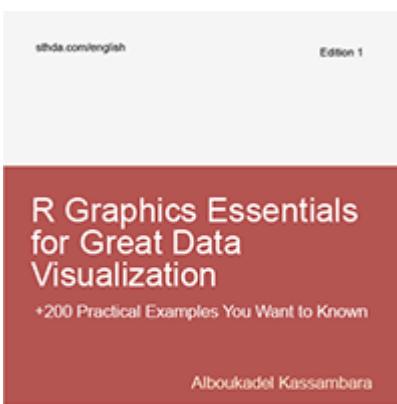
Ads by Google

Stop seeing this ad

Why this ad?



## Our Books

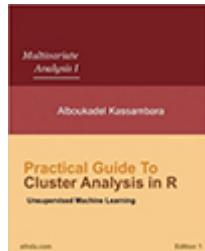




Ads by Google

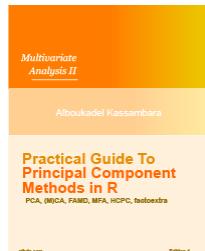
[Stop seeing this ad](#)

[Why this ad?](#)



Practical Guide to Cluster Analysis in R

## D Plots in R



Practical Guide to Principal Component Methods in R





Ads by Google

Stop seeing this ad

Why this ad?

sthda.com First edition





Ads by Google

[Stop seeing this ad](#)

[Why this ad?](#)



-  [Datanovia: Online Data Science Courses](#)
-  [R-Bloggers](#)

Newsletter

Email



Boosted by PHPBoost

