AFD_heart

Oumar Kane

19/03/2022

FAD with heart attack dataset

We have recuperated a dataset named heart_attack which is very interesting for a debut

Let's import the needed libraries further

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(FactoMineR)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'T':
## please set environment variable 'TZ'
```

```
library(factoextra)
```

```
## Le chargement a nécessité le package : ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# the glm function will be use because we have binary categorical target
```

We can recuperate the dataset by indicating the path

```
heart_data = read.csv("E:/Oumar/Ordinateur Dell/oumar/documents/Cours/IA data forest/master semestre 2/A
```

Let's attach the dataset to manipulate fastly the columns

```
attach(heart_data)
```

We have to see the five first lines of the dataset and verify the types of the variables

```r
head(heart_data)
```

```
##   age sex cp trestbps chol fbs restecg thalach exang oldpeak slope ca thal
## 1  52   1  0      125  212   0       1     168     0     1.0     2  2    3
## 2  53   1  0      140  203   1       0     155     1     3.1     0  0    3
## 3  70   1  0      145  174   0       1     125     1     2.6     0  0    3
## 4  61   1  0      148  203   0       1     161     0     0.0     2  1    3
## 5  62   0  0      138  294   1       1     106     0     1.9     1  3    2
## 6  58   0  0      100  248   0       0     122     0     1.0     1  0    2
##   target
## 1      0
## 2      0
## 3      0
## 4      0
## 5      0
## 6      1
```

The data is already cleaned. We have some categorical variables but those variables are encoded. Let's profile deeper the dataset.

Let's see if the different variables are correlated together with a pairplot

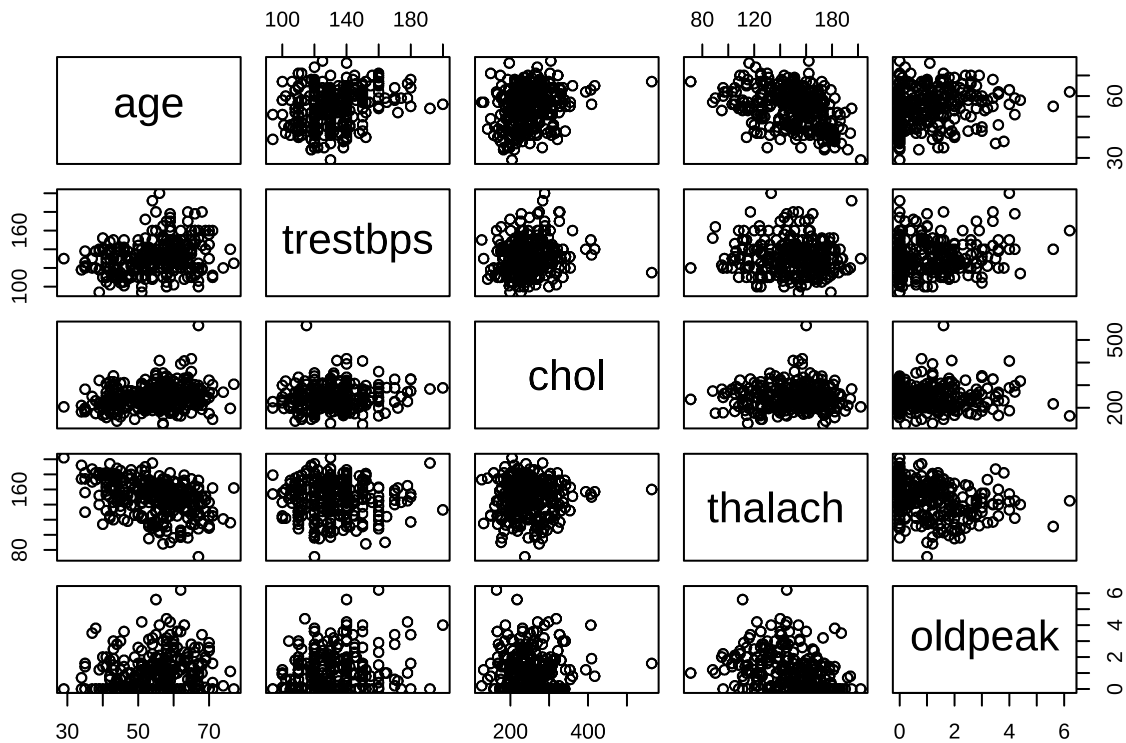
and a corrplot

We cannot describe the correlation plots with the categorical variables but only with the quantitative variables. Let's recuperate only the quantitative variables in a new data frame.

```r
quanti = heart_data[, c(1, 4, 5, 8, 10)]
```

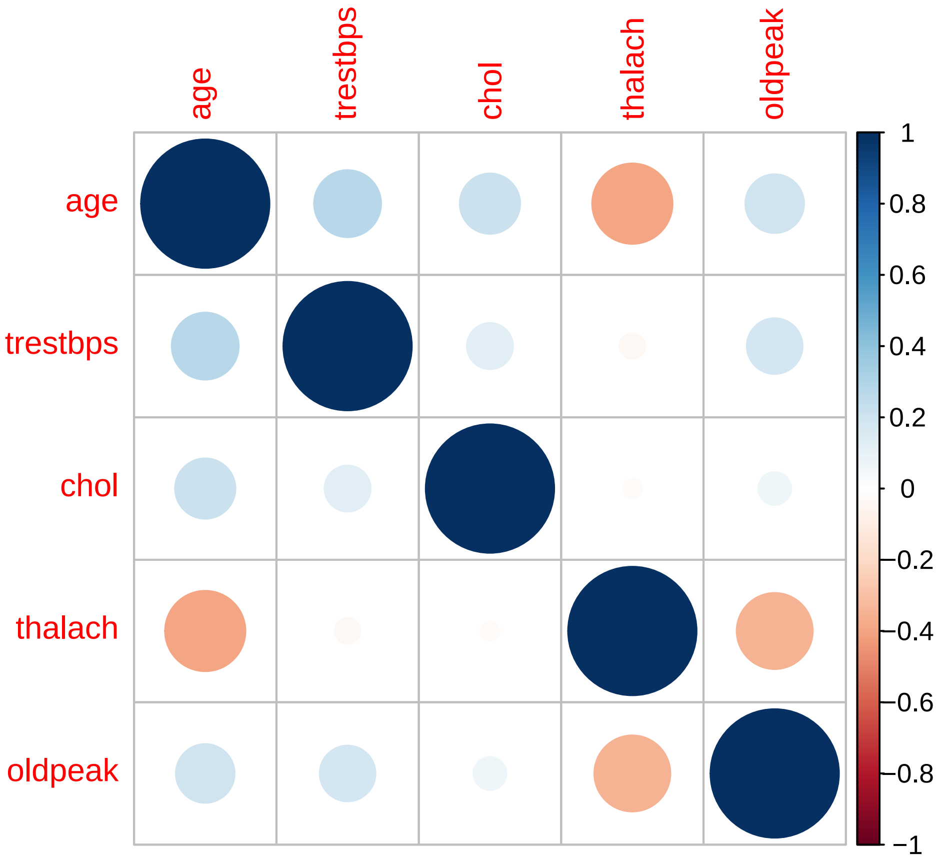
Let's trace a pairplot without categorical variables

```r
pairs(quanti)
```



We see that some variables like thalach and age or chol and age are correlated

```
# Let's trace the corrplot to see more clearly the variables interactions
corrplot(cor(quanti))
```



Conclusion : We can see that age is correlate with the other variables. Thalach and oldpeak are also correlated.

With the factorial discrimant analysis we cannot use all of the categorical variables but just one of them the target variable. We will do it with two categorical variables at another time The target variable contains the binary digits 1 and 0 which represent respectively that the patient may die from a heart attack and the patient will not get a heart attack

```
# We can recuperate the variables that we want to use for the next steps
heart_dis = heart_data[, c(1, 4, 5, 8, 10, 14)]
```

Let's change the type of the categorical variable to factor

```
heart_dis[, 6] = as.factor(heart_dis[, 6])
```

Let's begin the factorial discriminant analysis

We will use the FAMD function which can perform the analysis

```
# we will not visualize a graph for the moment
res.fad = FAMD(heart_dis, graph = FALSE)

# Let's see which attributes the result contains
print(res.fad)
```

```
## *The results are available in the following objects:
```

```
##
##   name          description
## 1 "$eig"        "eigenvalues and inertia"
## 2 "$var"        "Results for the variables"
## 3 "$ind"        "results for the individuals"
## 4 "$quali.var"  "Results for the qualitative variables"
## 5 "$quanti.var" "Results for the quantitative variables"
```

Interprate the results

Eigenvalues and inertia

The eigen values measure the variance percentage explained by each of the components or eigen vectors

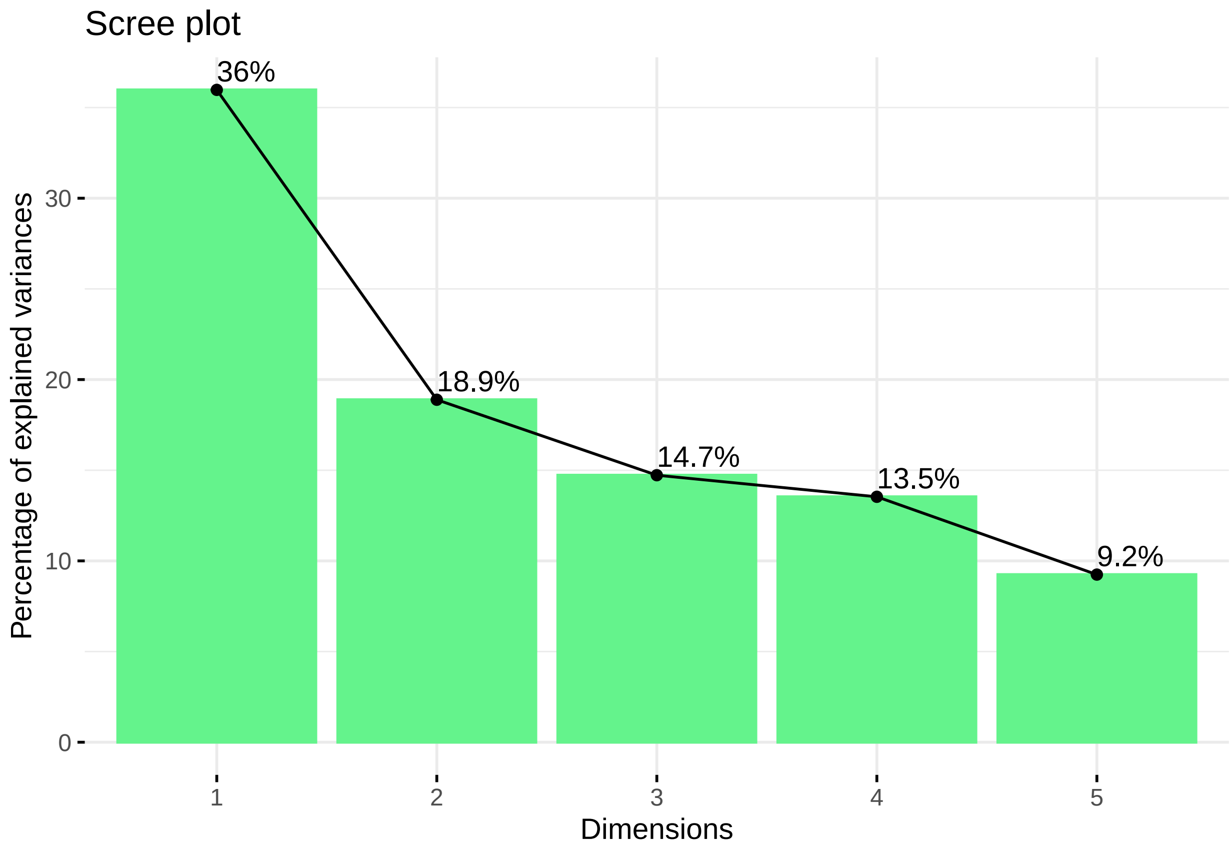
```r
# Let's recuperate the eigen values in a variable
eig_val = get_eigenvalue(res.fad)
eig_val
```

```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1  2.1584774        35.974624                    35.97462
## Dim.2  1.1332553        18.887588                    54.86221
## Dim.3  0.8835463        14.725772                    69.58798
## Dim.4  0.8121462        13.535769                    83.12375
## Dim.5  0.5546565         9.244275                    92.36803
```

The two first dimensions contain 54.80 % of the information and the three first variables explain 69.53 % of the information. To obtain more than 80 % of the information we must use as least the four first dimensions

Visualization of the eigen values on a screeplot

```r
# Let's use the fviz_eig to obtain the graphic
fviz_eig(res.fad, addlabels = TRUE, barfill = "#64f38c", barcolor = "#64f38c")
```



We obtained a curve elbow on the fourth dimension **conclusion: The four first dimensions are enough important for obtain most of the dataset information**

Visualization of the variable results :

For the moment we trace the plots with the target variable but the categorical variable will be exclude of the interpretations for the moment.

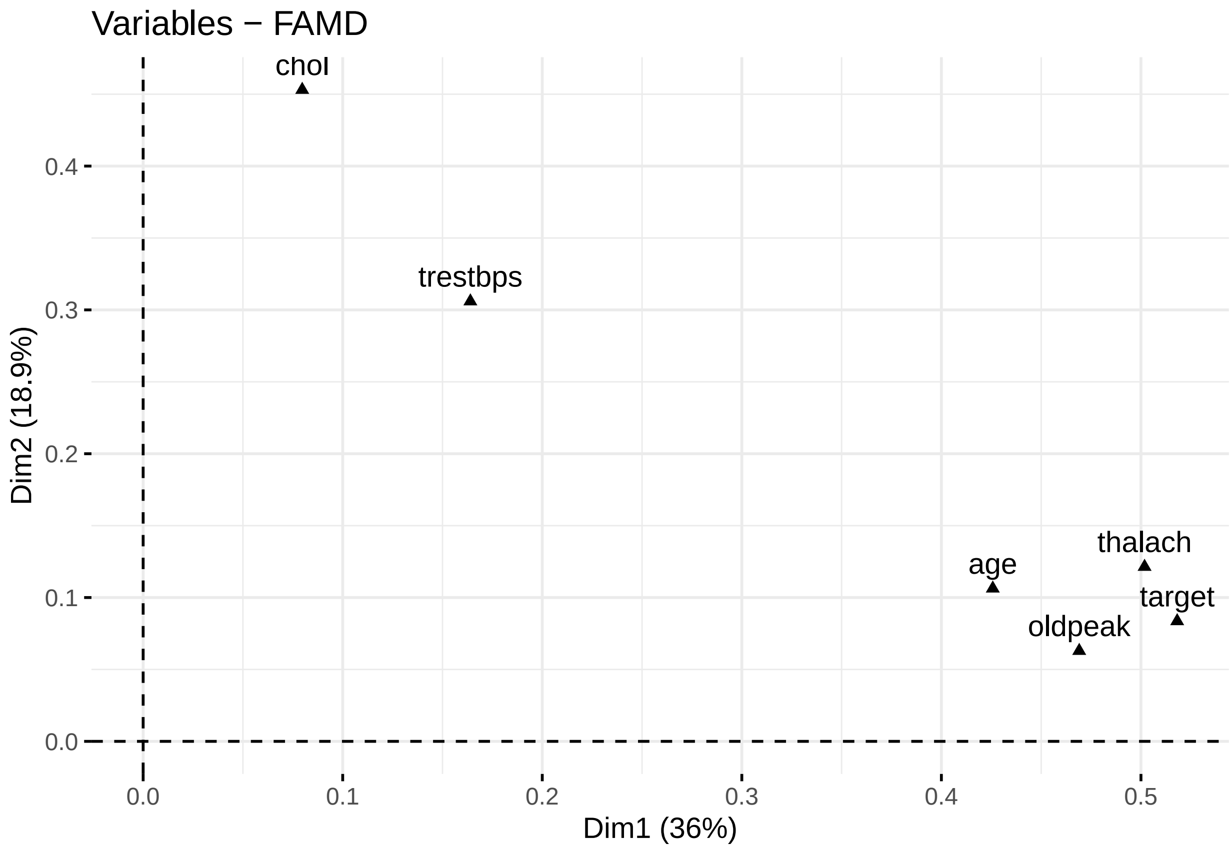
the get_afd_var() function will be used to get the variable results

```
var = get_famd_var(res.fad)
var
```

```
## FAMD results for variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates"
## 2 "$cos2"    "Cos2, quality of representation"
## 3 "$contrib" "Contributions"
```

Let's plot the results on a graph

```
fviz_famd_var(res.fad, col.var = "Black")
```

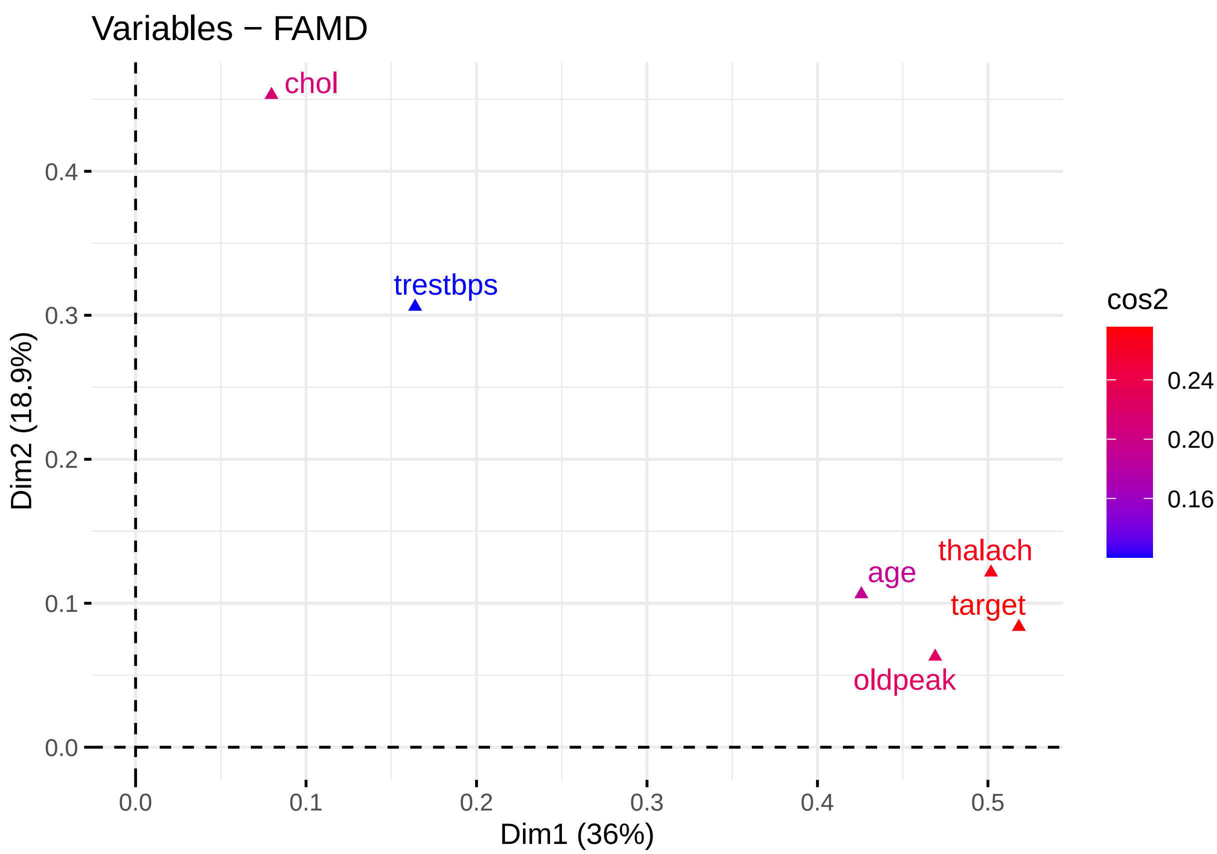


Conclusion: We see clearly that the variable which the second dimension represents the most is chol variable and for the first dimension we have age, thalach and oldpeak. trestbps variable isn't well represented by both of the dimensions.

Quality of representation (cos2)

Let's plot the same graph with, at this time, variables colored by quality of representation

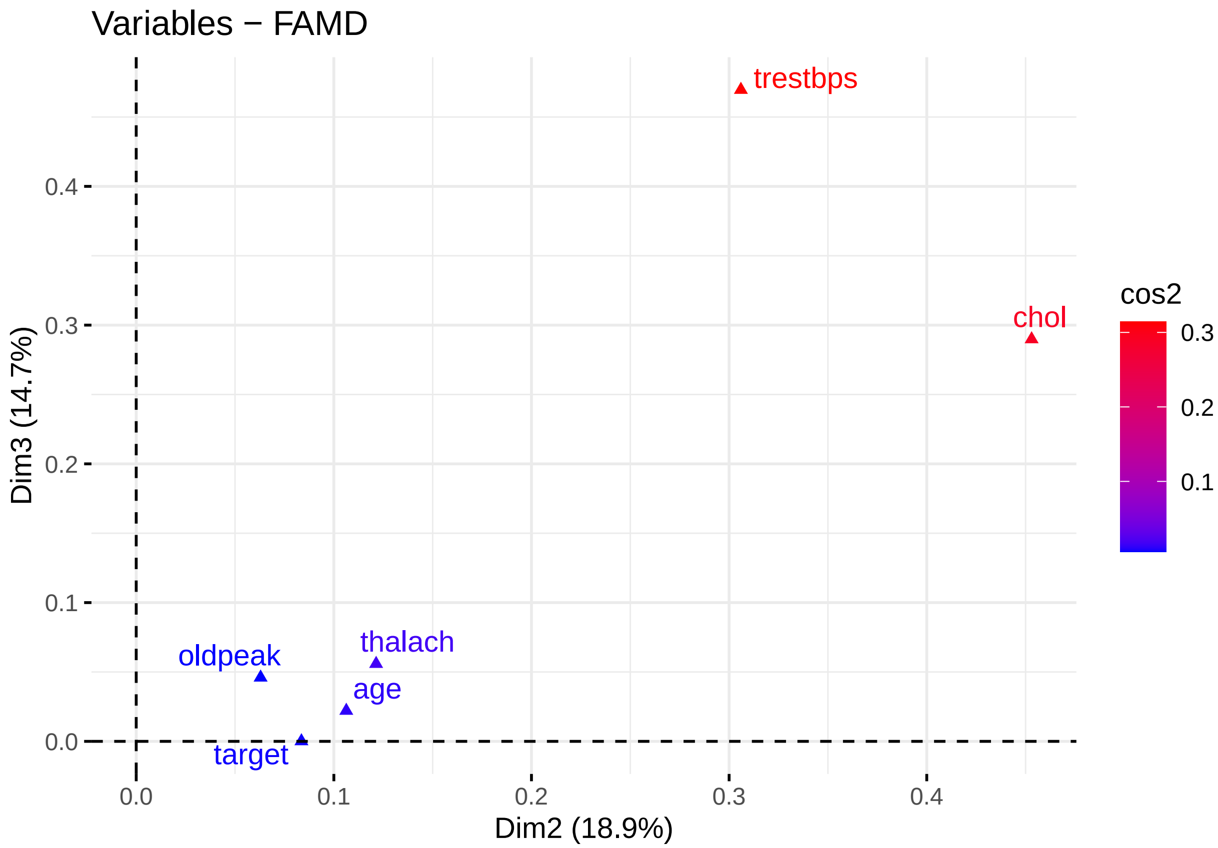
```
fviz_famd_var(res.fad, col.var = "cos2", gradient.cols = c("blue", "red"), repel = TRUE)
```



But we can see that the most high quality percentage is around 0.5, which is not very important So we can conclude that we have to choose more than 2 dimensions to have a good representation of the variables.

We can trace with the dimensions 3 and 4 and see what happens

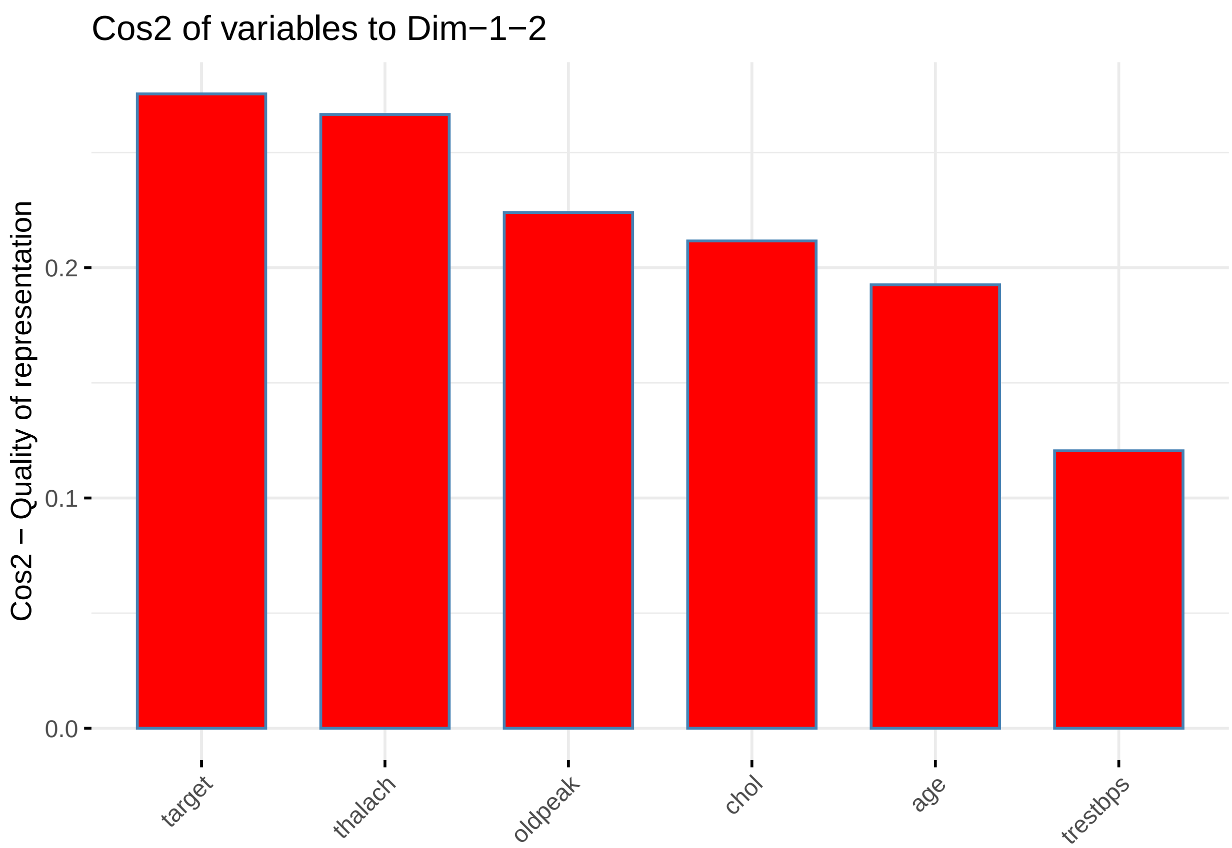
```
fviz_famd_var(res.fad, col.var = "cos2", gradient.cols = c("blue", "red"), repel = TRUE, axes = c(2, 3))
```



The trestbps is most well represented by the third dimension

Quality of representation to axes 1 and 2

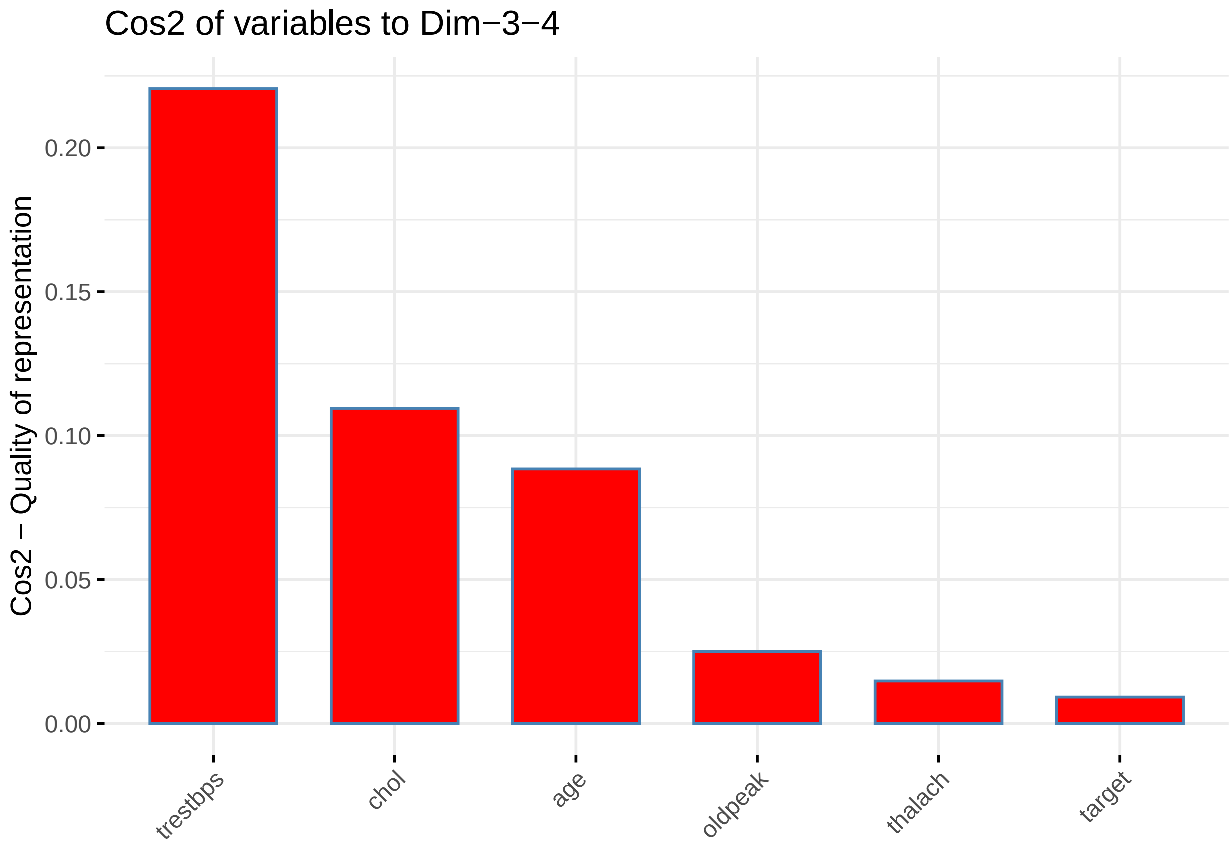
```
fviz_cos2(res.fad, choice = "var", fill = "red", axes = 1:2)
```



Every variables are represented by the first two axes

Quality of representation to axes 3 and 4

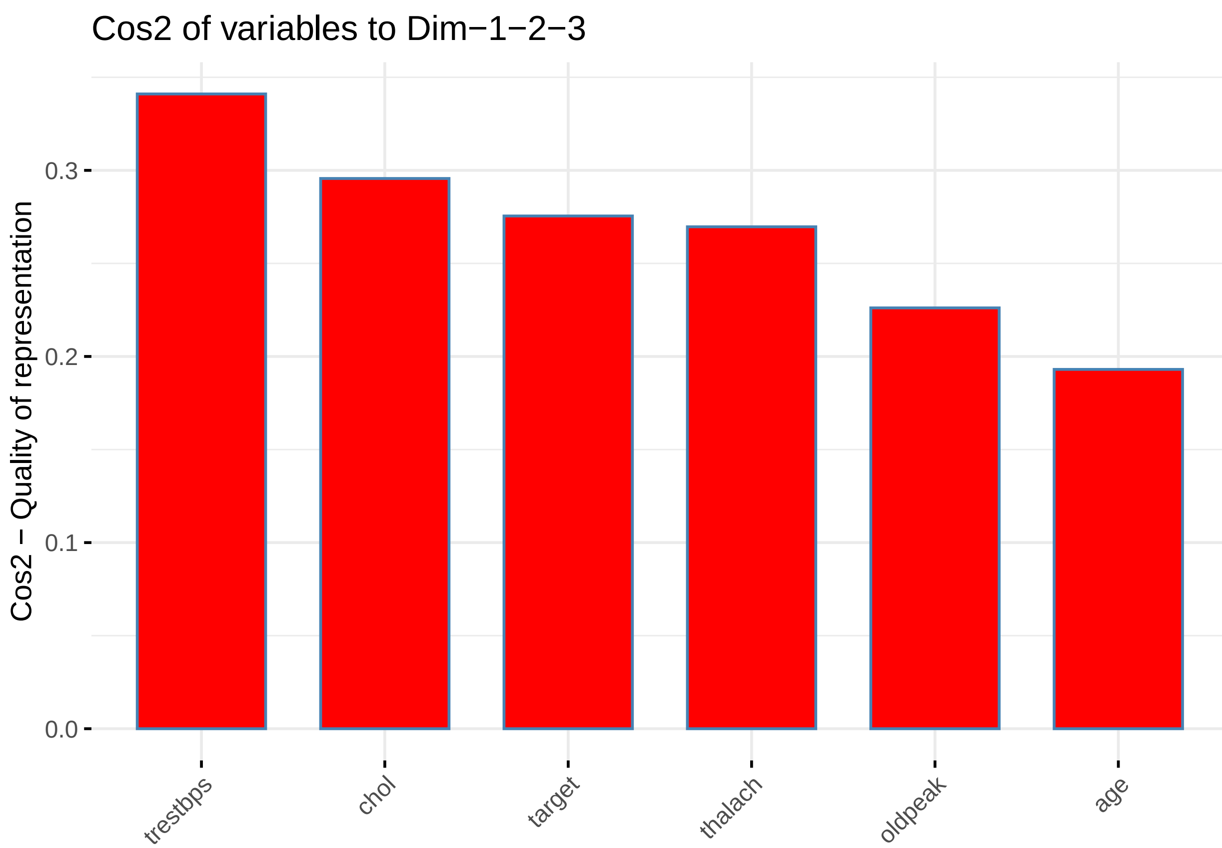
```
fviz_cos2(res.fad, choice = "var", fill = "red", axes = 3:4)
```



Some variables like trestbps and chol are more represented by the axes 3 and 4. We can plot with the first third axes to see if we obtain best analysis.

Quality of representation to axes 1 to 3

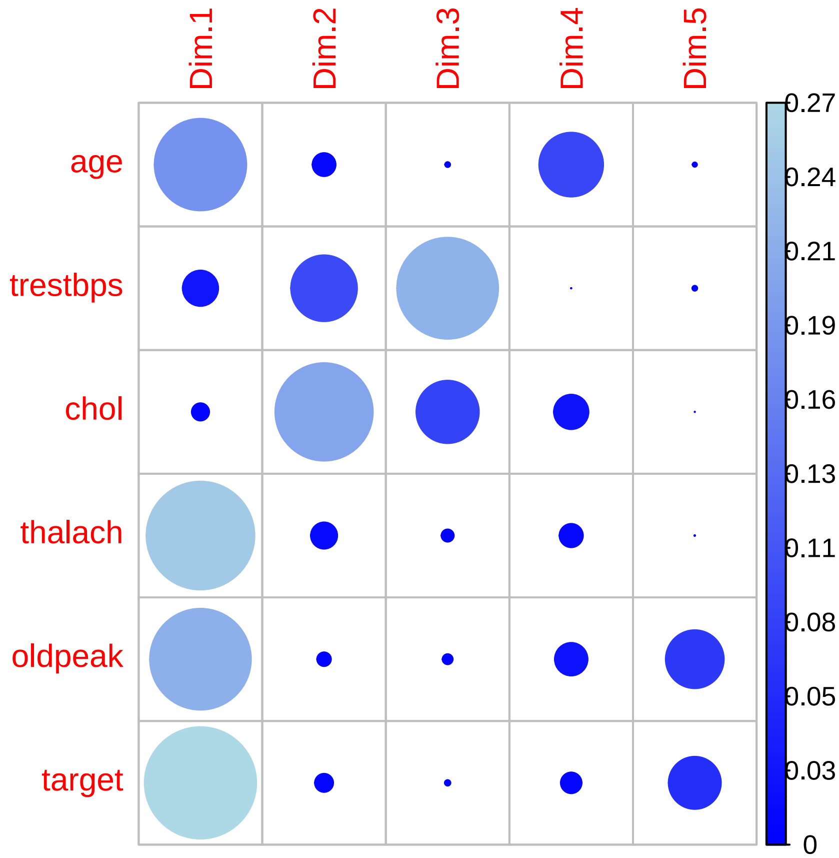
```
fviz_cos2(res.fad, choice = "var", fill = "red", axes = 1:3)
```



We obtain a greater quality for all the variables. The quality of representation of trestbps is the more important.

Let's see if we obtain a greater analysis with a corrplot

```
corrplot(var$cos2, is.corr = FALSE, col = colorRampPalette(c("blue", "lightblue"))(200))
```

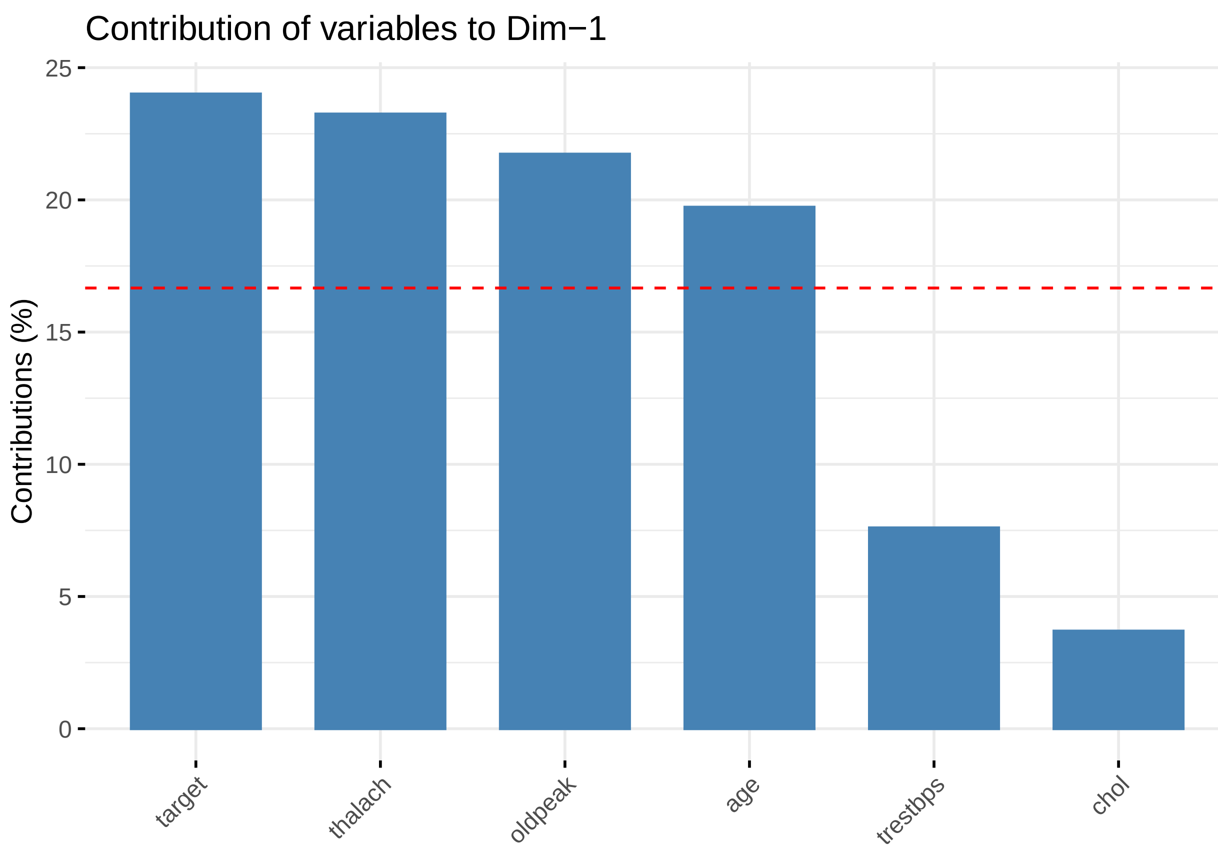


We see that the first three dimensions represent all of the variables if we combine them together

Contributions of the variables to the contruction of the dimension

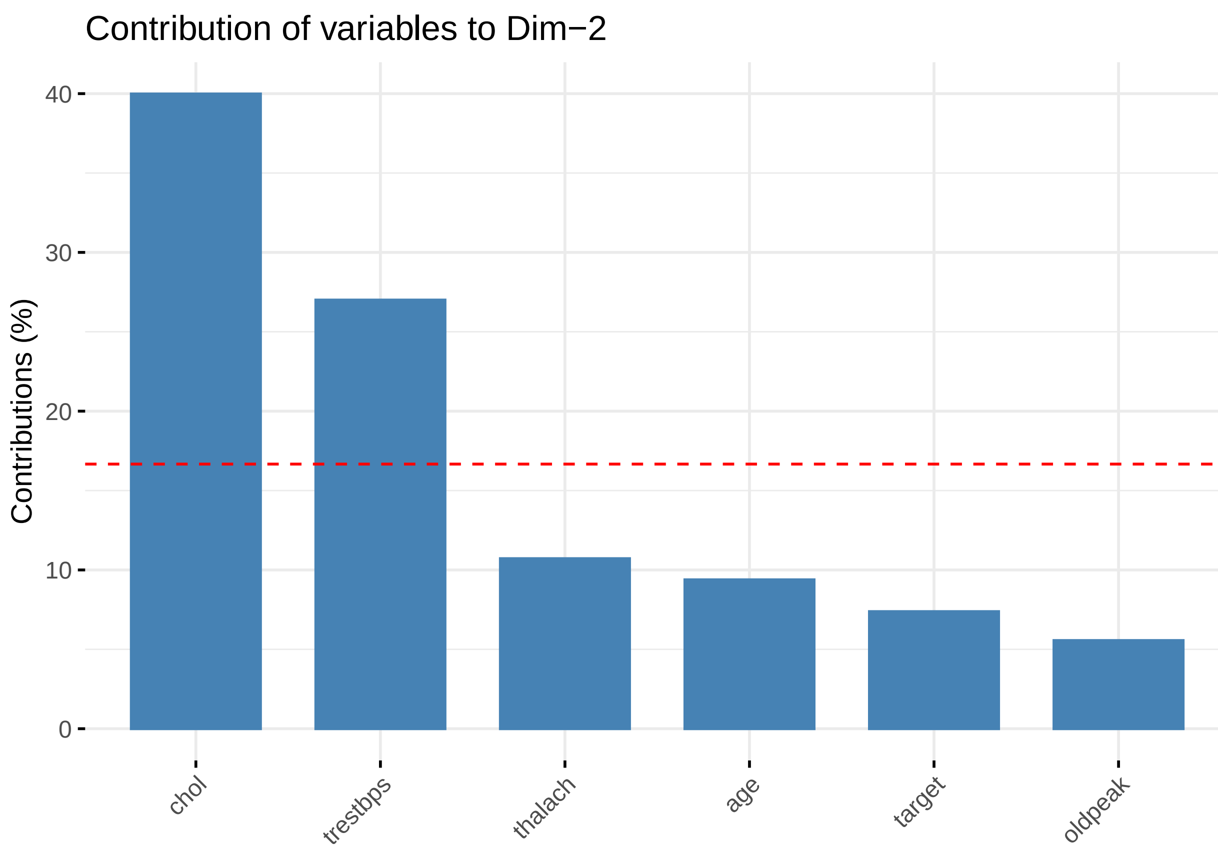
Let's plot the contributions of the variables to the axes

```
fviz_contrib(res.fad, choice = "var")
```



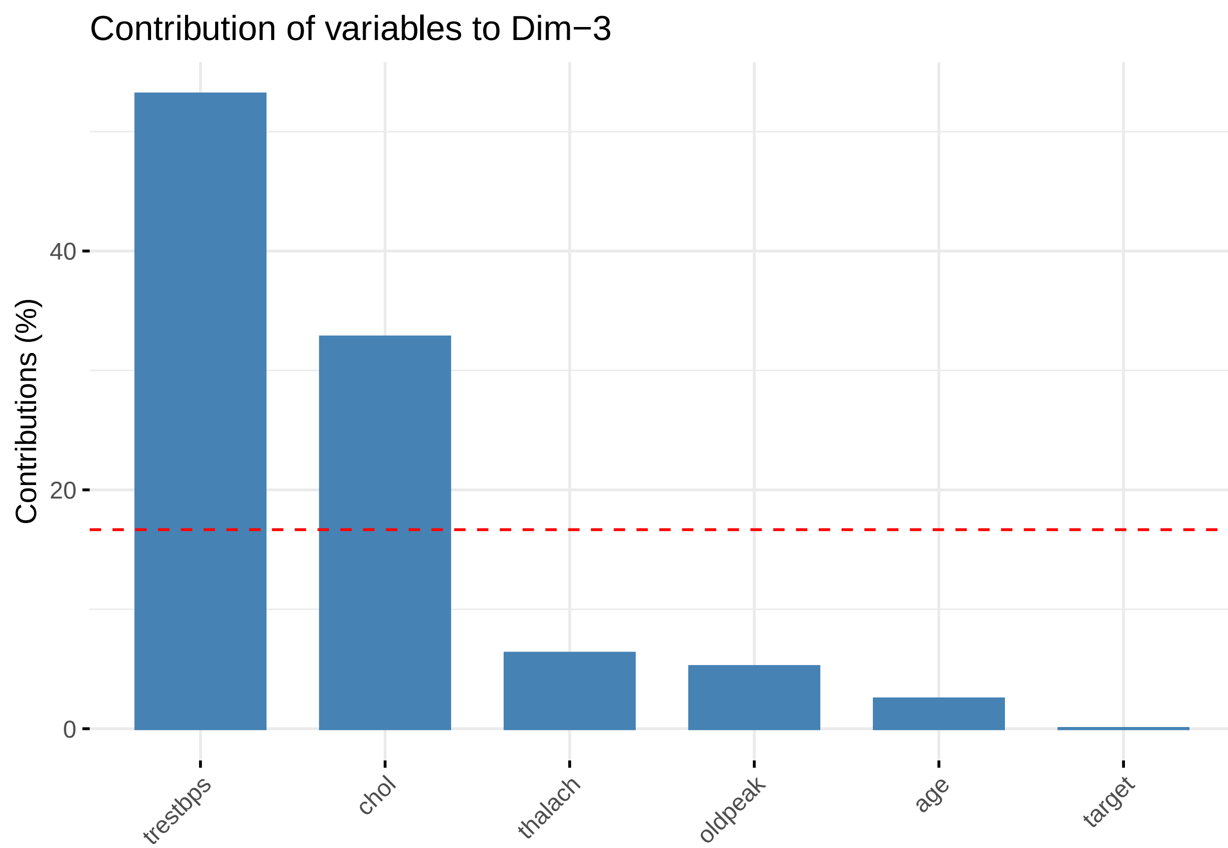
To axe 1
Automaticaly we see that thalach, oldpeak and age are the variable that contribute the most to the first dimension.

```
fviz_contrib(res.fad, choice = "var", axes = 2)
```



To axe 2
For the second axe, only chol and trestbps contribute greatly to his construction.

```
fviz_contrib(res.fad, choice = "var", axes = 3)
```

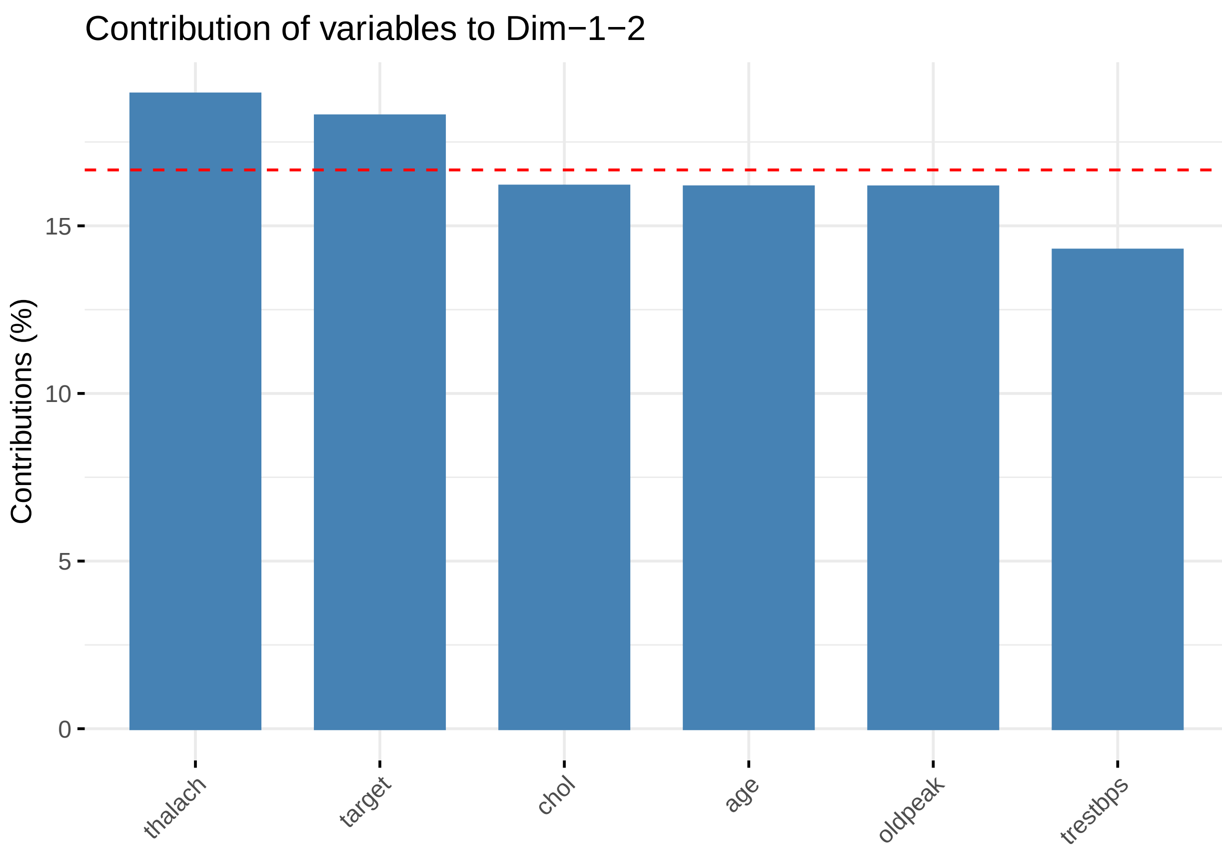


To axe 3

For the axe 3 we have trestbps and chol.

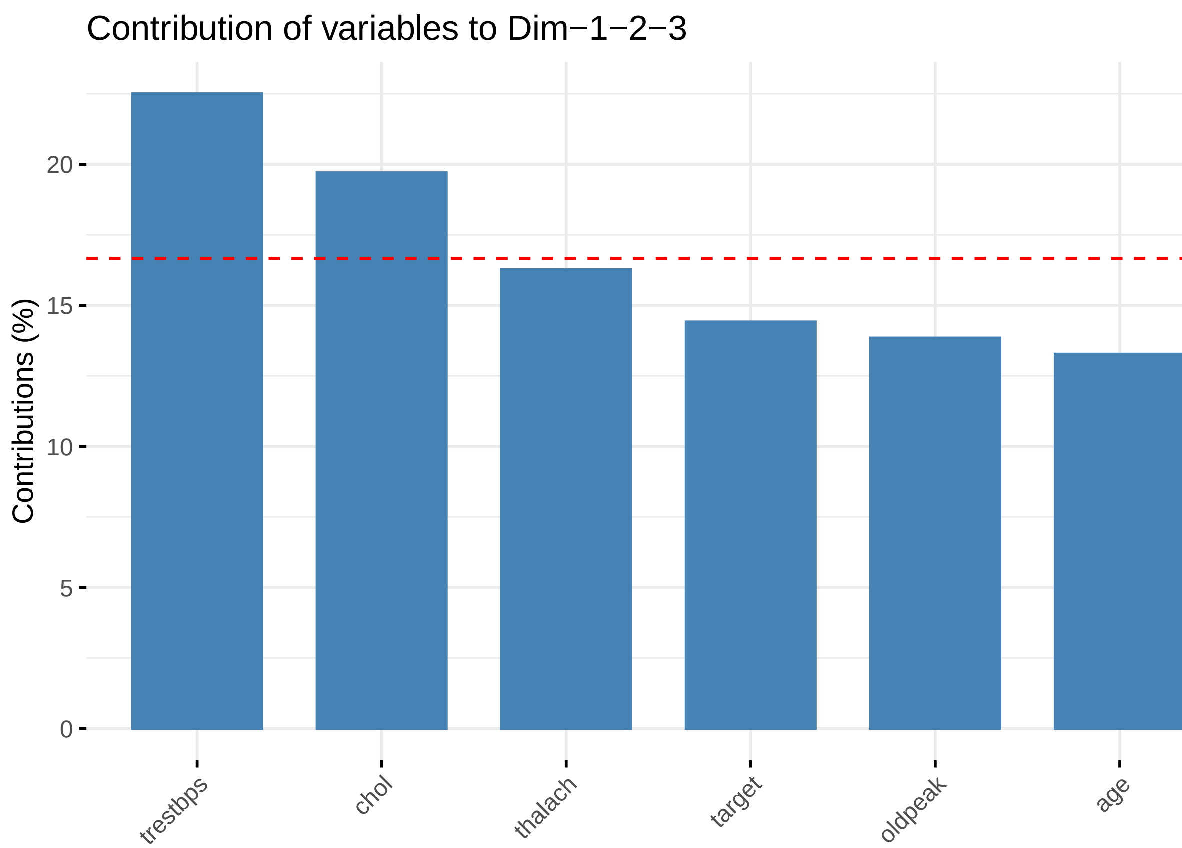
Let's plot now for multiple axes #### To axes 1 and 2

```
fviz_contrib(res.fad, choice = "var", axes = c(1, 2))
```



Great contribution are done by the variables : thalach, chol, age, oldpeak and trestbps.

```
fviz_contrib(res.fad, choice = "var", axes = 1:3)
```

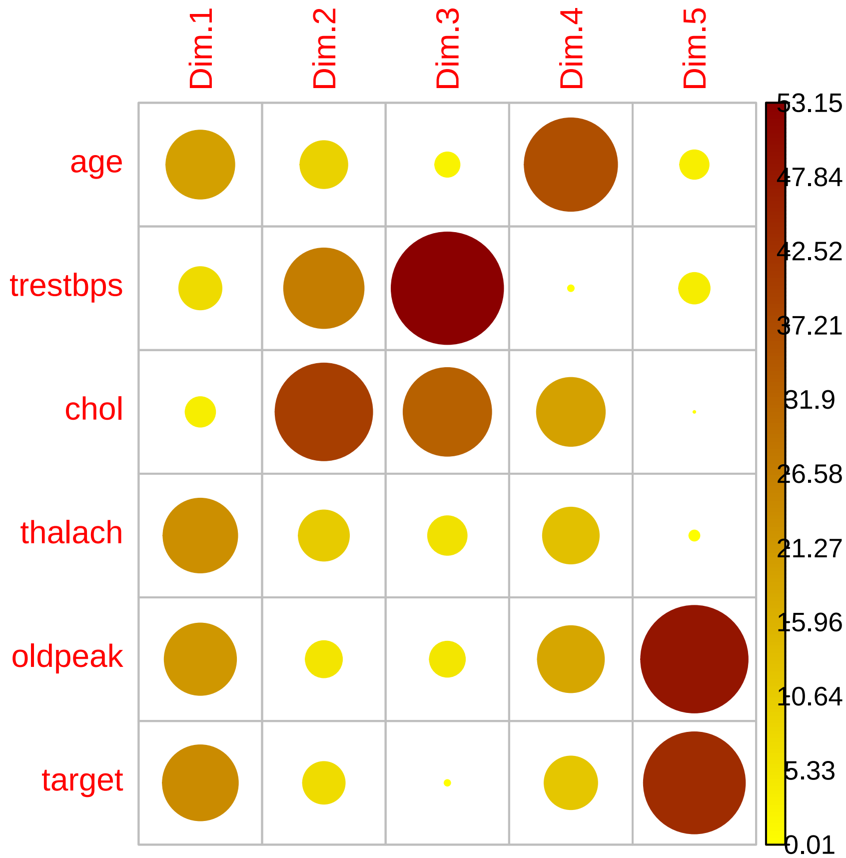


To axes 1, 2 and 3

With the axe 3, the contribution of trestbps is the most important.

Let's plot the contributions of the variables to the axes with a corrplot and see what happens

```
corrplot(var$contrib, is.corr = FALSE, col = colorRampPalette(c("yellow", "darkred"))(200))
```



Every variables contribute to the construction of the first three dimensions.

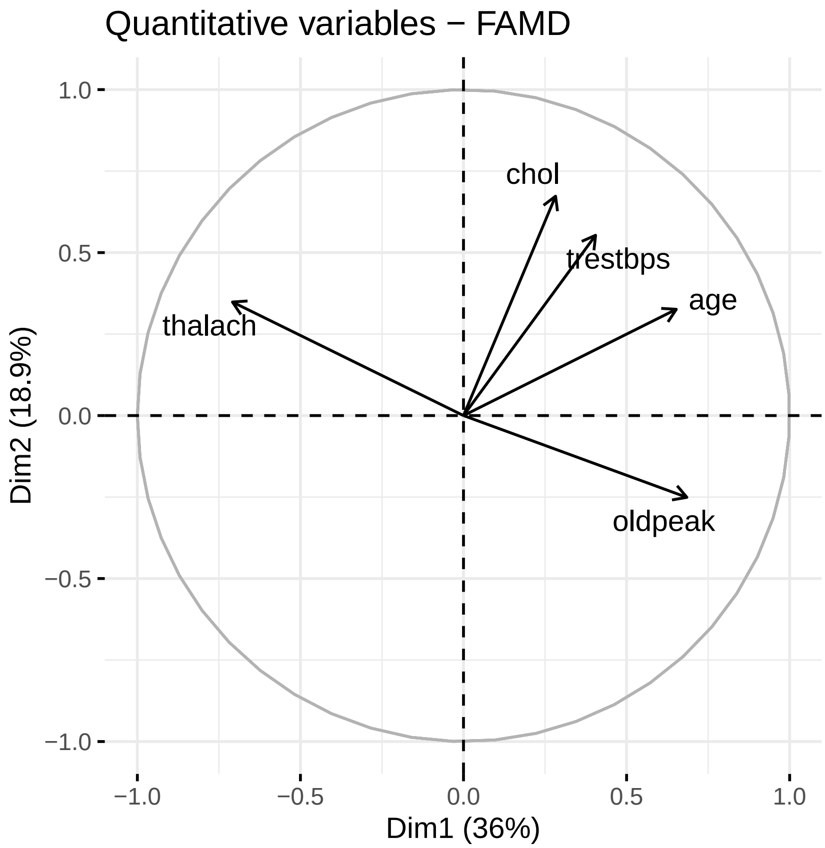
We can begin the analysis of the quantitative variables

Let's recuperate those variables, further.

```
quanti.var = get_famd_var(res.fad, element = "quanti.var")
quanti.var
```

```
## FAMD results for quantitative variables
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates"
## 2 "$cos2"    "Cos2, quality of representation"
## 3 "$contrib" "Contributions"
```

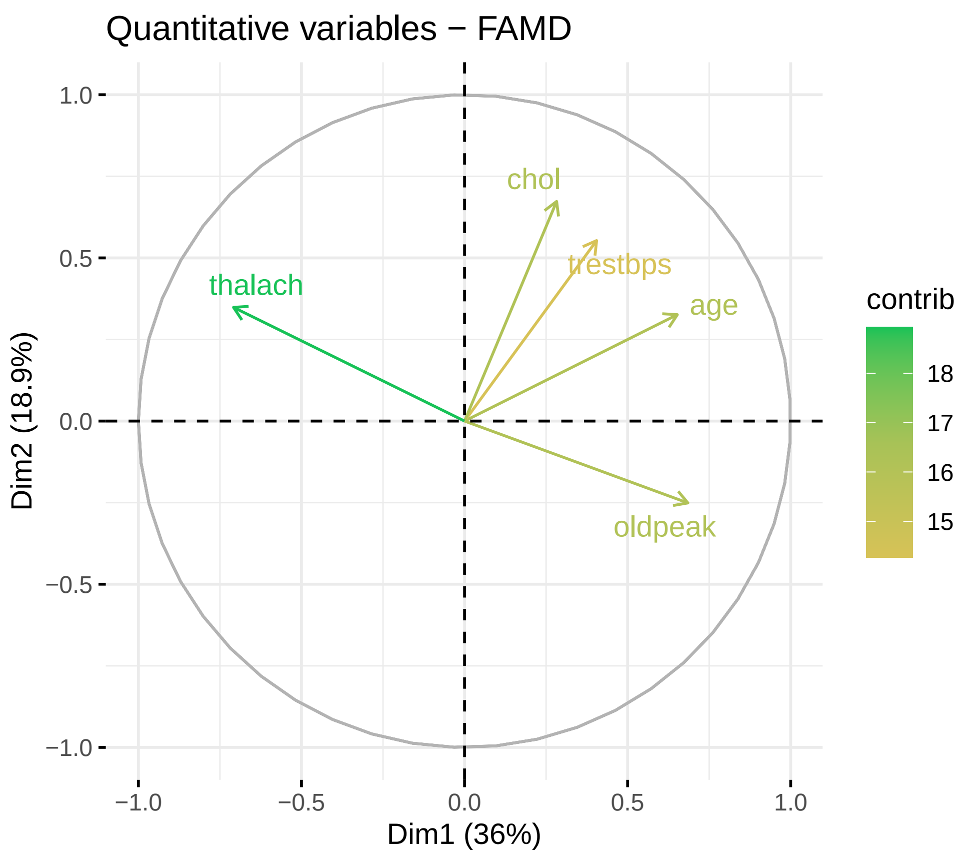
```
fviz_famd_var(res.fad, "quanti.var", repel = TRUE, col.var = "Black")
```



The thalach, age and oldpeak variables are more represented by the first dimension and the chol and trestbps are more represented by the second dimension. We can see clearly that the thalach and oldpeak variables are negatively correlated and chol, trestbps and age are posivetely correlated.

Let's color the variables by percentage of contributions

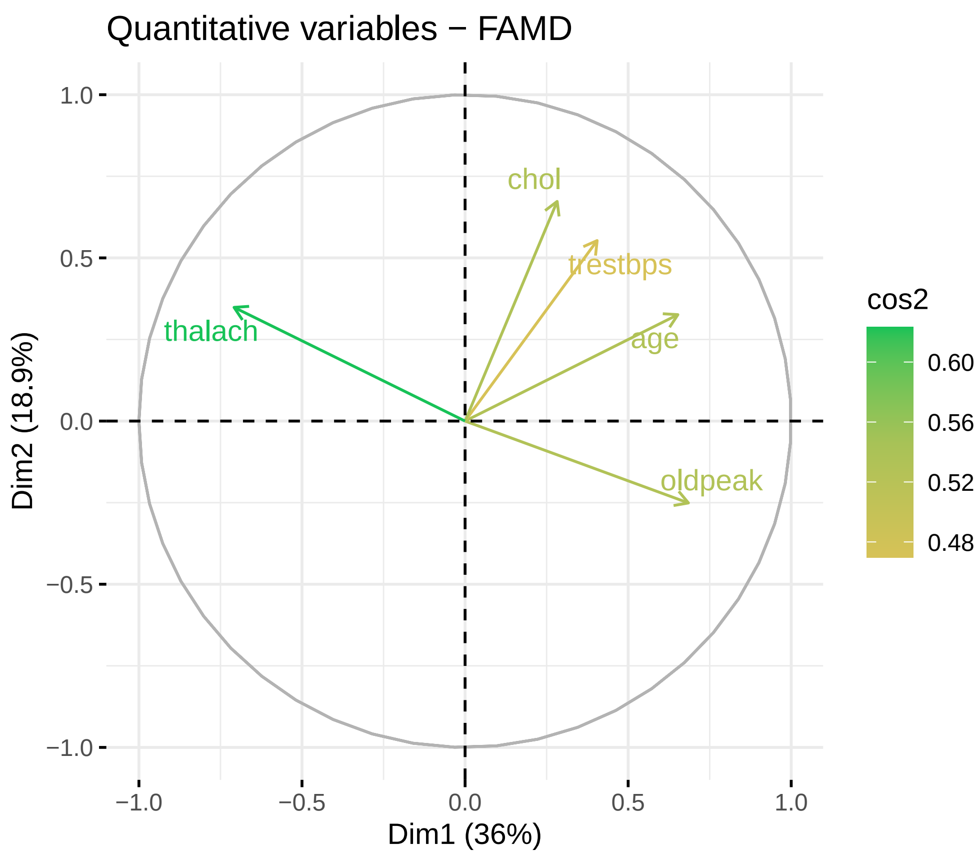
```
fviz_famd_var(res.fad, "quanti.var", col.var = "contrib", repel = TRUE, gradient.cols = c("#D7c257", "#A
```



We see that thalach are the variable that contributes the most to the construction of the first two dimensions.

Let's color the variables by percentage of representation quality

```
fviz_famd_var(res.fad, "quanti.var", col.var = "cos2", repel = TRUE, gradient.cols = c("#D7c257", "#A7c
```



The thalach variable is also the variable that have the best quality of representation.

Visualization of observations

Let's get the observation results and analyse them.

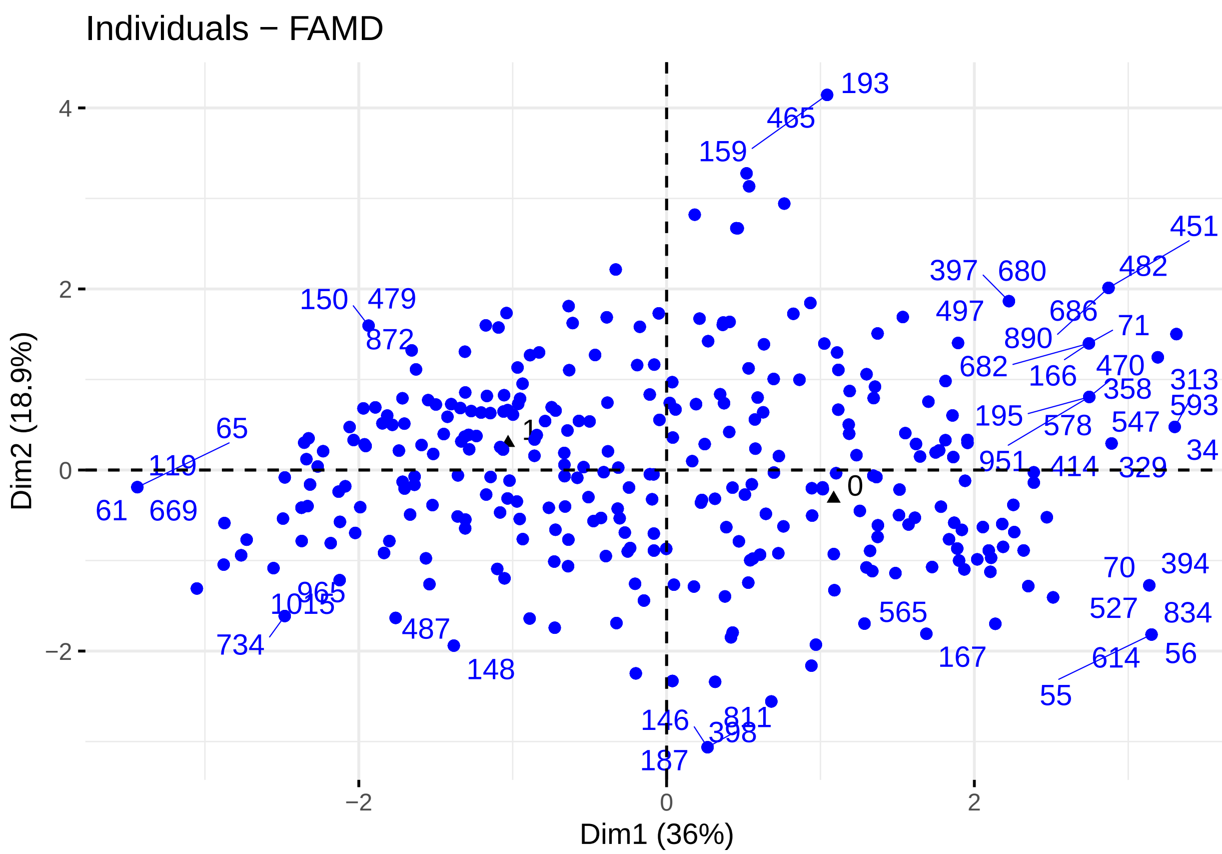
```
ind = get_famd_ind(res.fad)
ind
```

```
## FAMD results for individuals
##  ===================================================
##   Name       Description
## 1 "$coord"   "Coordinates"
## 2 "$cos2"    "Cos2, quality of representation"
## 3 "$contrib" "Contributions"
```

we can plot the observations

```
fviz_famd_ind(res.fad, repel = TRUE)
```

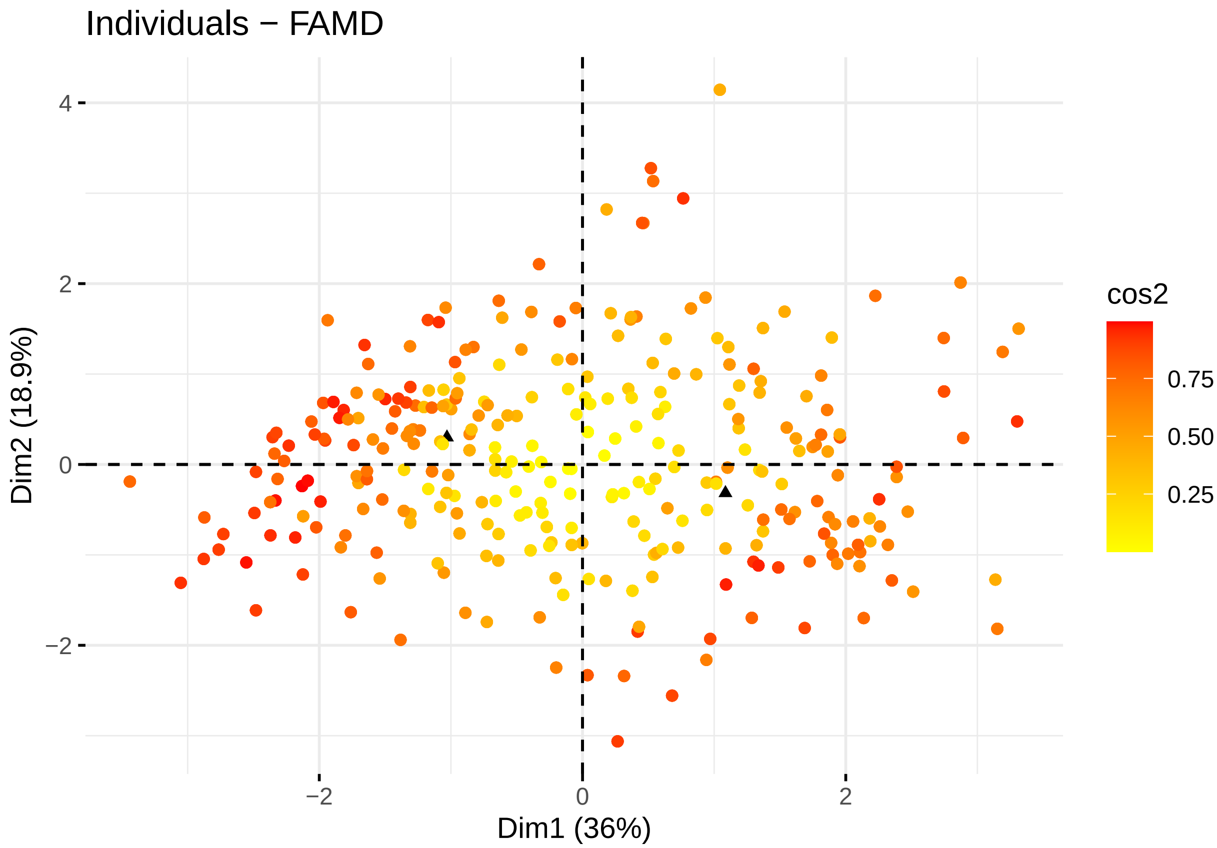
```
## Warning: ggrepel: 976 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



We have too many observations so the graphic is not very clear.

We can color the observations following their representation qualities

```
fviz_famd_ind(res.fad, col.ind = "cos2", gradient.cols = c("yellow", "red"), repel = TRUE, geom = "poin
```

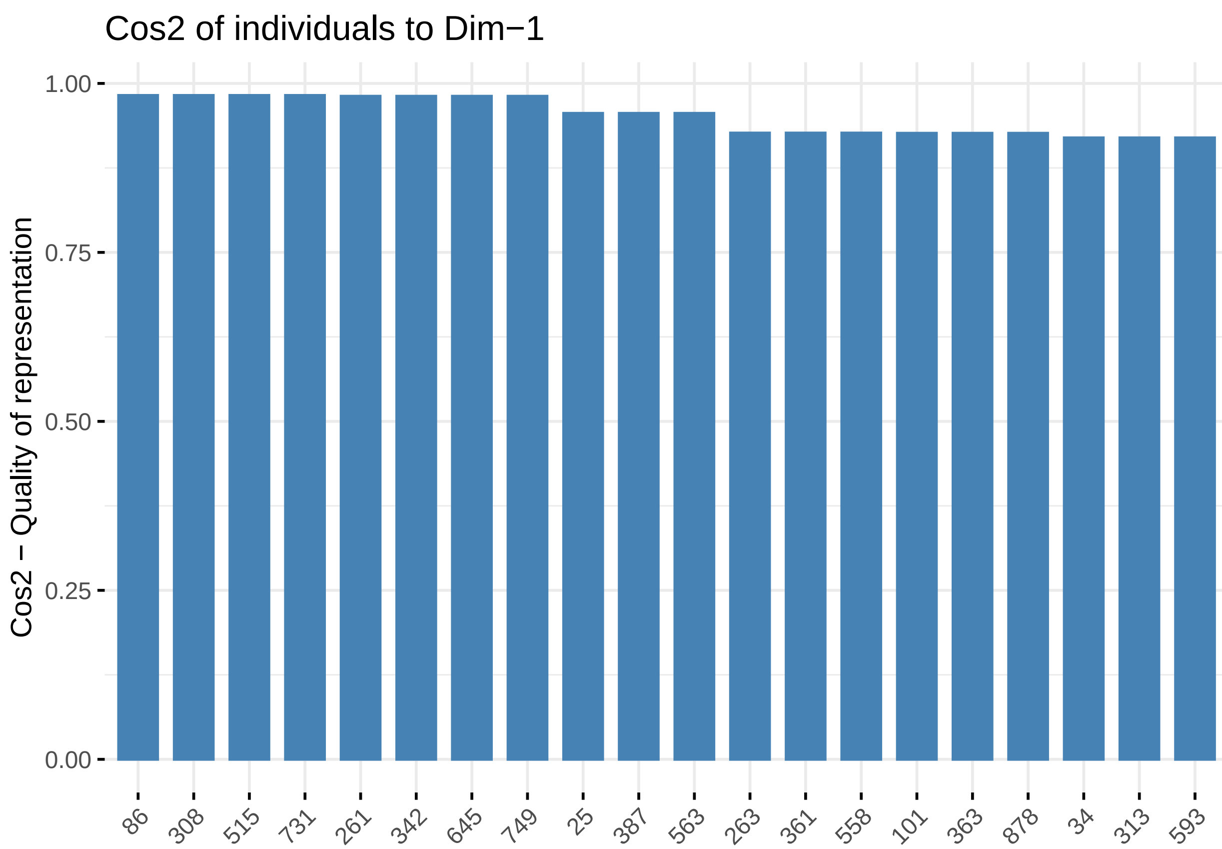


Many individuals have a good representation quality (over than 75 percent)

We can keep the top 20 of individuals which have the most important representation qualities to dimensions

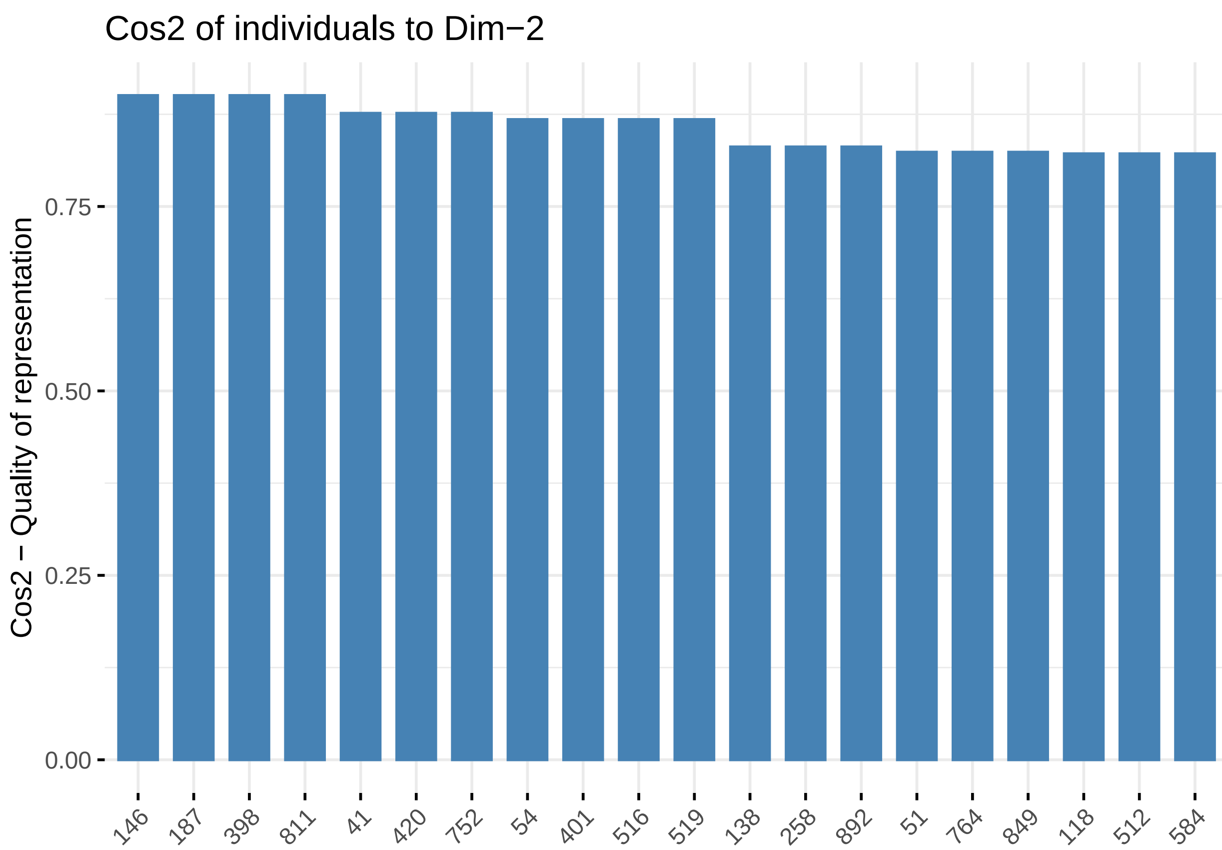
To Dim 1

```
fviz_cos2(res.fad, choice = "ind", top = 20)
```



To Dim 2

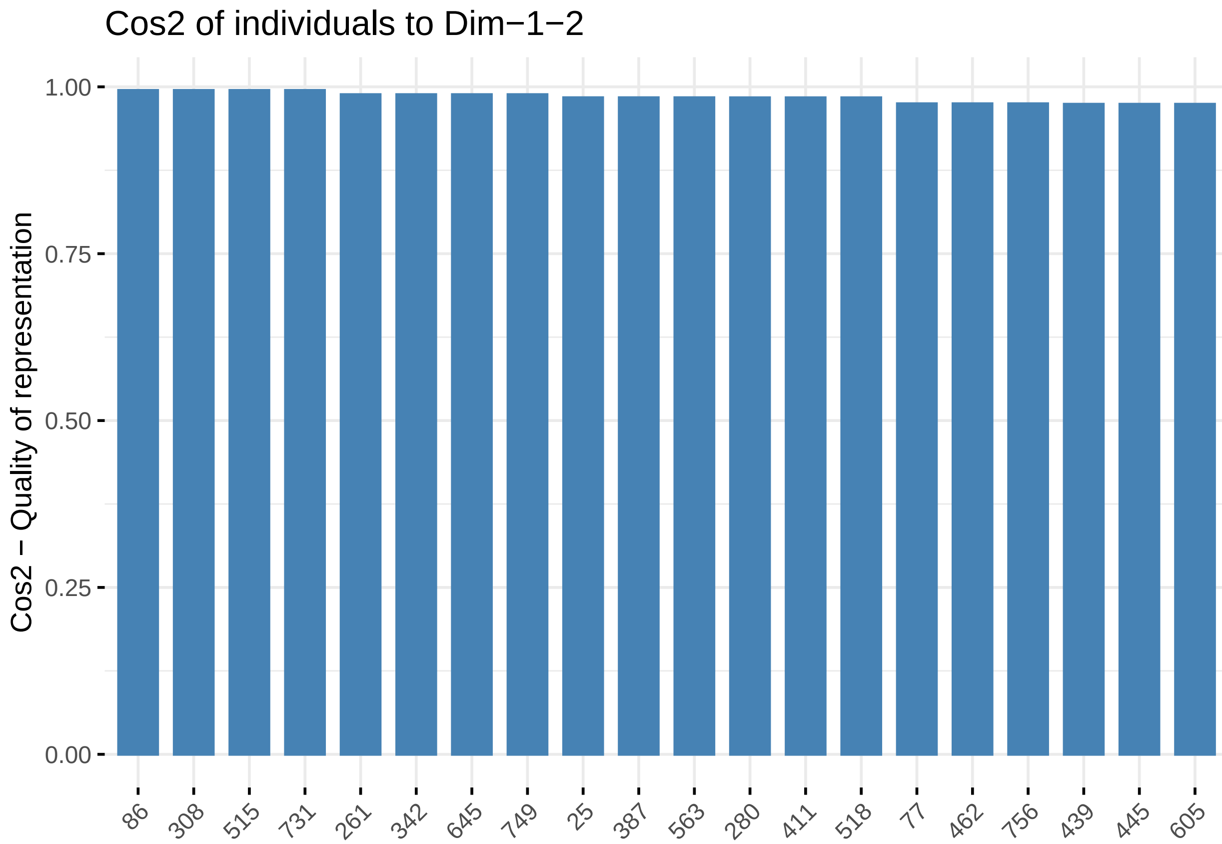
```
fviz_cos2(res.fad, choice = "ind", top = 20, axes = 2)
```



For both of axes 1 and 2 we obtain over than 75 percent of representation quality for the top 20 of individuals But we doesn't obtain the same individuals. Let's plot for axes 1 and 2

To axes 1 and 2

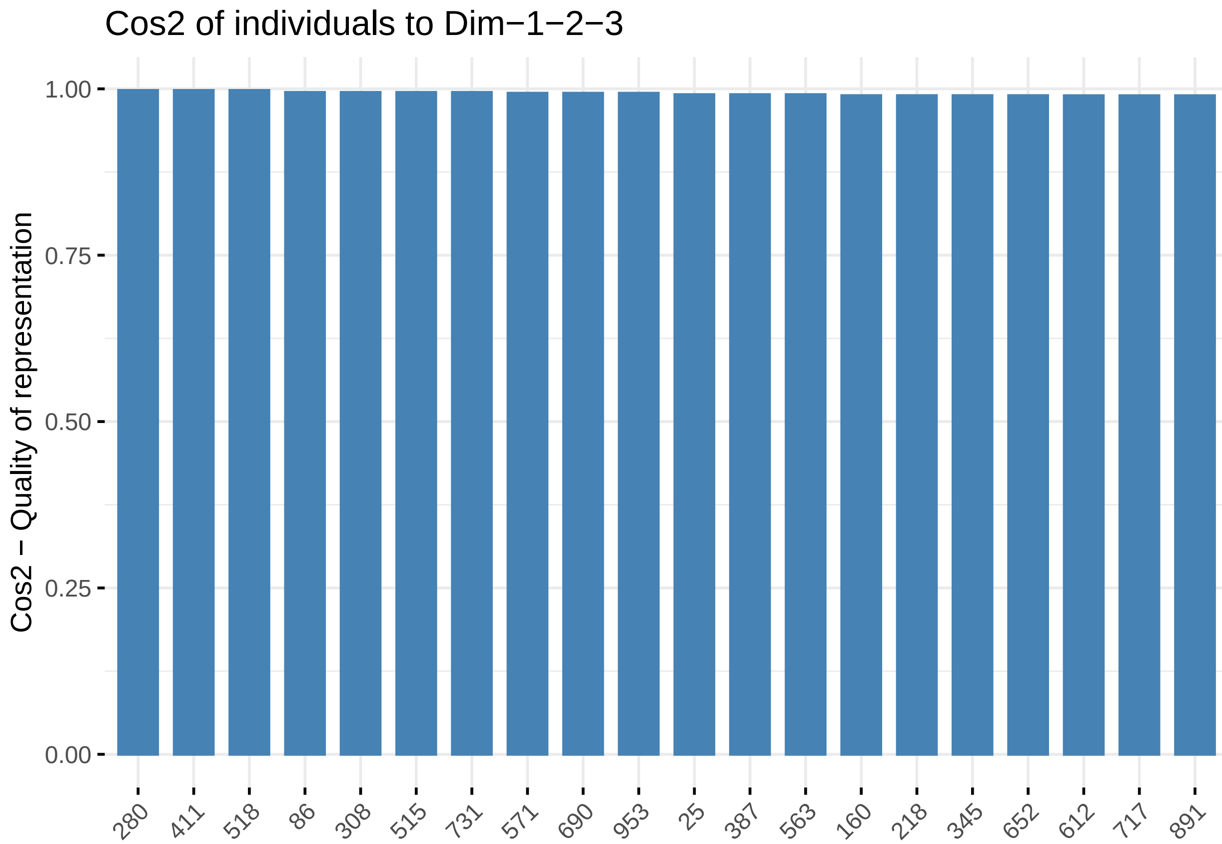
```
fviz_cos2(res.fad, choice = "ind", top = 20, axes = c(1, 2))
```



The representation quality of individuals top 20 is over than 90 %. That is very important.

To axes 1, 2 and 3

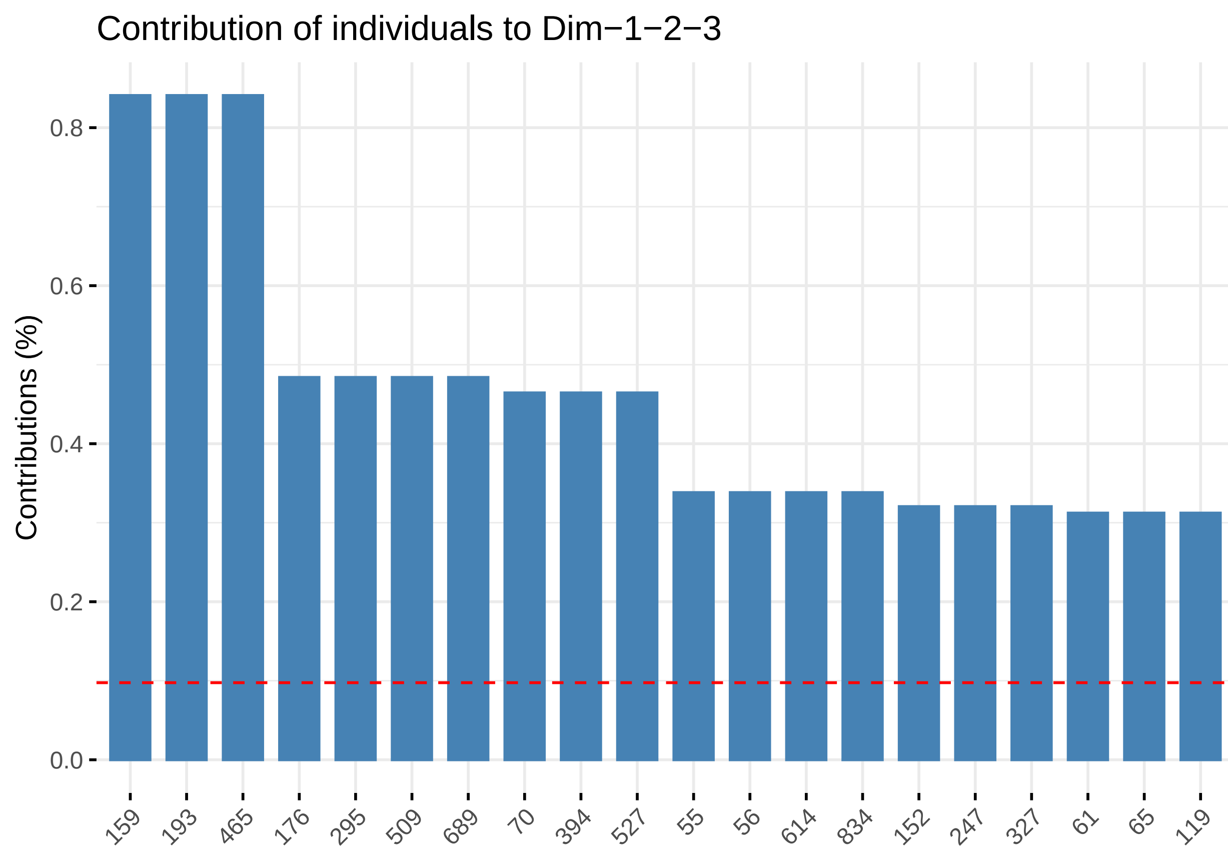
```
fviz_cos2(res.fad, choice = "ind", top = 20, axes = c(1, 2, 3))
```



Let's visualize now the contributions of the individuals to axes

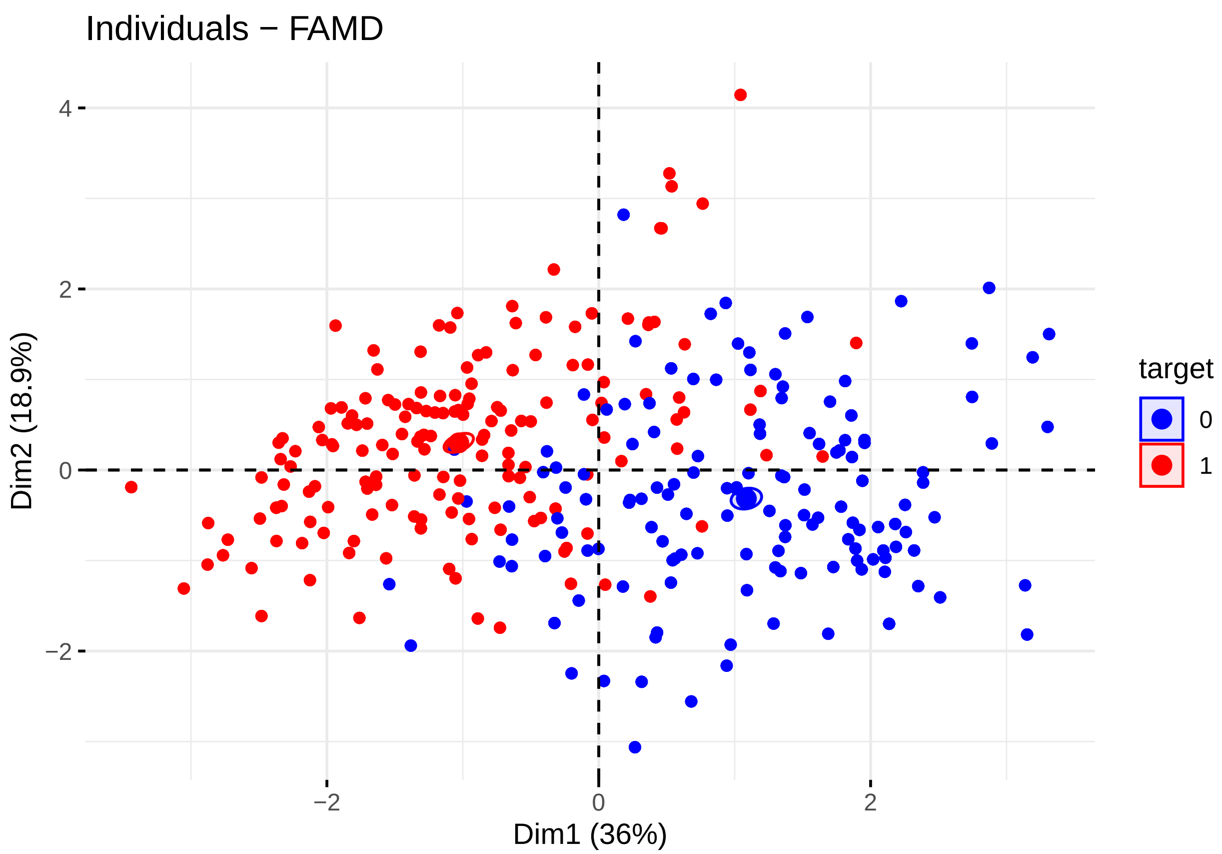
To axe 1, 2 and 3

```
fviz_contrib(res.fad, choice = "ind", top = 20, axes = c(1, 2, 3))
```



Let's plot finally the clusters with the categorical variable, target

```
fviz_famd_ind(res.fad,
              geom = "point",
              habillage = "target",
              palette = c("blue", "red"),
              addEllipses = TRUE,
              ellipse.type = "confidence",
              repel = TRUE)
```



Individuals − FAMD
Dim2 (18.9%)
Dim1 (36%)
4
2
0
−2
−2
0
2
target
0
1