Practice CA, MCA and Clustering

Marc Mendez & Joel Cantero

30 de abril de 2019

1. Read the PCA_quetaltecaen data.

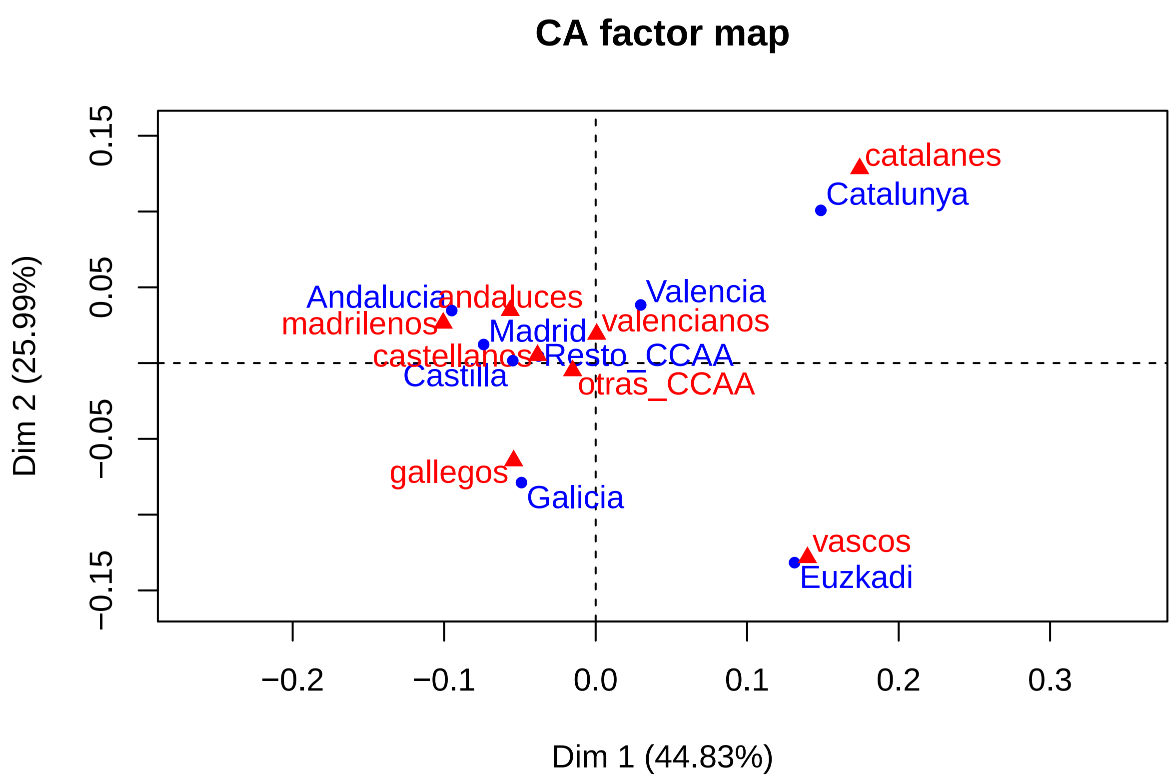
```
setwd("~/Desktop/UPC 18:19/S2 18:19/MVA/Homework 4")
quetal <- read.delim("PCA_quetaltecaen.txt","\t", header = TRUE)
names(quetal)[7]<- "madrilenos"
quetal.data <- quetal[-1]
row.names(quetal.data) <- quetal$CCAA
```

The dataset PCA_quetaltecaen is written in a txt file where the columns are spaced by tabulations.This file includes one strange caracther, so to deal with it, we will leave madrile??os as madrilenos to avoid any problem with the ISO encodings.

2. Perform a CA of this data. How many dimensions are significant?. Interpret the first factorial plan.

So here we execute the CA to our dataframe.

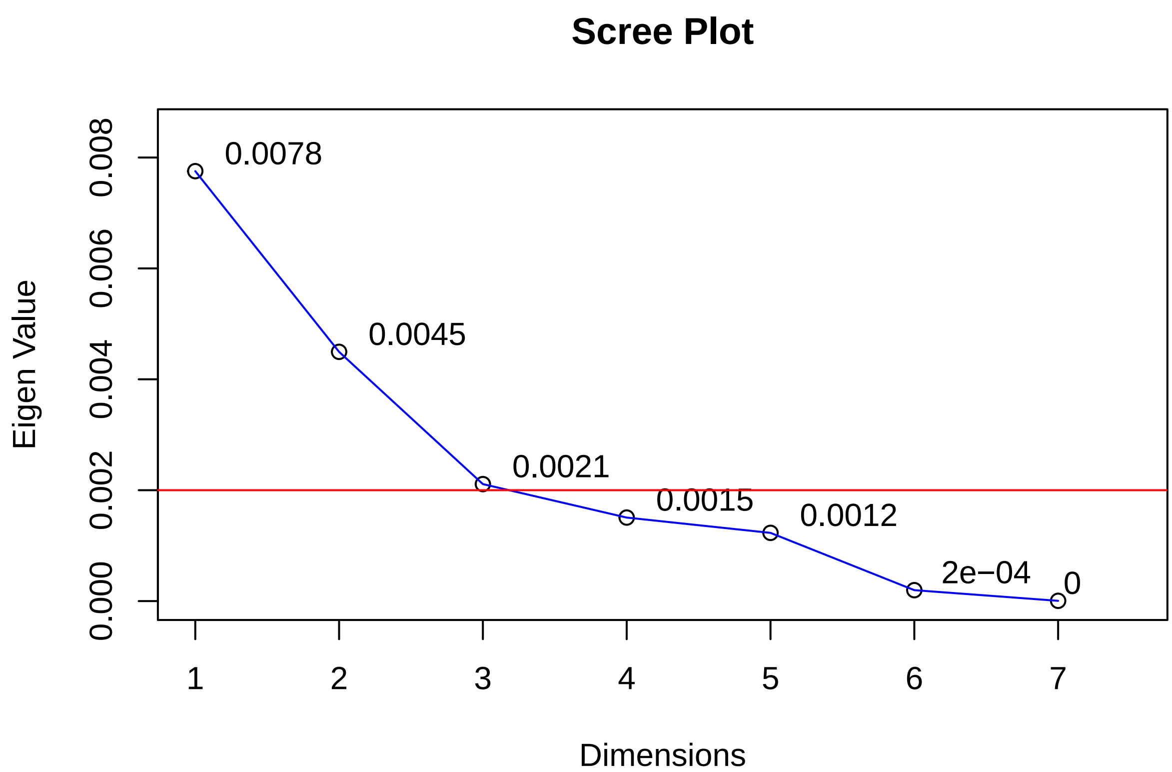
```
ca.quetal <- CA(quetal.data)
```



We can easily see that the relation between habitants of its regions is very high. We see this by the distance between "Euzkadi" and "vascos" or "Catalunya" and "catalanes", as they are the most clear examples but we can see it with the others too. So Catalunya and Euzkadi are the most distinguished regions, Galicia is a little bit away from the center cluster, and all the others are more or less close to the center.

To select the number of dimensions that are significant, we are going to aply as the other Homeworks, the last Elbow rule.

```
x <- as.data.frame(ca.quetal$eig)
x.eigenV <- x$eigenvalue
x.eigenVec <- x$`percentage of variance`
x.scree <- data.frame(Number = seq(1, length(x.eigenV)), Value = x.eigenV)
plot(x.scree, main="Scree Plot",
     xlab="Dimensions", ylab="Eigen Value" , ylim=c(0, max(x.eigenV) + 0.1*max(x.eigenV)), xlim=c(1, ler
lines(x.scree$Value, col="blue")
textxy(x.scree$Number, x.scree$Value, round(x.scree$Value, 4), cex=1)
abline(h=0.002, col="red")
```




```
sum(x.eigenVec[1:3])
```

```
## [1] 83.01099
```

We choose the first 3 dimensions as significant, which retain 83.01099% of the information. ##3. For the PCA_quetaltecaen data, compute the contribution of each cell to the total inertia, that is: (fij - f.i x f.j)^2/(fi.x f.j). Compute the percentage of inertia due to the diagonal cells.

```
intertiaOfEachCell <- function(data){
    f <- data/sum(data)
    fi <- rowSums(f)
    fj <- colSums(f)

    contribution <- (f)

    for (i in seq(1, nrow(f))) {
```

```r
        for(j in seq(1, ncol(f))) {
            temp <- (fi[i] * fj[j])
            contribution[i, j] <- (((f[i, j] - temp)^2)/temp)
        }
    }
    return(contribution)
}
quetal.intertiaCell <- intertiaOfEachCell(quetal.data)
```

For doing this we will create a function that calculates the total intertia of each cell as, later on we will need to do it again. We do it that way because with the inertia of each cell, then we can calculate the total of rows and columns. To check if the values are correct we will compare the sum of the total inertia to the eigen values which will have to be the same.

```r
quetal.totalIntertia <- sum(quetal.intertiaCell)
totalEigenV <- sum(x.eigenV)
quetal.totalIntertia
```

```
## [1] 0.01729803
```

```r
totalEigenV
```

```
## [1] 0.01729803
```

As we can see are the same so the computation was made correctly. Moreover, we need to calculate the percentatge inertia that comes from the diagonal cells.

```r
quetal.sumDiag <- sum(diag(as.matrix(quetal.intertiaCell)))
quetal.diagIntertia <- quetal.sumDiag*100/quetal.totalIntertia
quetal.diagIntertia
```

```
## [1] 74.19063
```

In our case the percentatge of information that contains the diagonal is way too much(74%) so in the next part we will try to nullify it.

4. Clearly, the overloaded diagonal of the data set influences the results obtained (the overall inertia is mainly due to this overload diagonal). Try to nullify this influence by imputing the diagonal values by the independence hypothesis values of the product of marginal probabilities (=n x fi.x f.j). Take into account that each imputation modifies the marginal, hence you need an iterative algorithm.

To nullify the overloaded diagonal we will apply the method seen in class. It is something similar to what we can see under this paragraph.

```r
quetal.data.2 <- quetal.data
for (x in seq(1, 10)) {
  for (x in seq(1, nrow(quetal.data.2))) {
    n <- sum(quetal.data.2)
    f2 <- quetal.data.2/n
    fi2 <- rowSums(f2)
    fj2 <- colSums(f2)
    quetal.data.2[x,x] <- n * fi2[x] * fj2[x]
  }
}
```

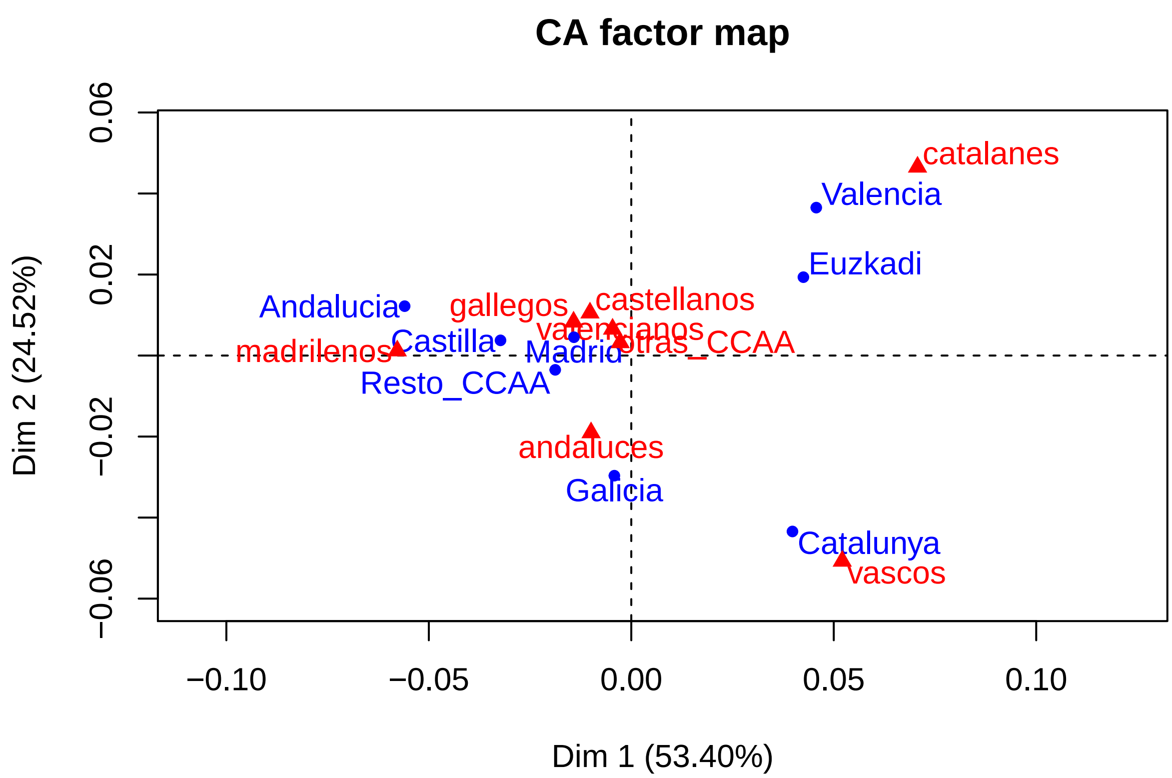
Once the diagonal is nullified, try to calculate the inertia again and see if the diagonal is still overloaded or the problem its solved.

```
quetal.intertiaCell.2 <- intertiaOfEachCell(quetal.data.2)
quetal.totalIntertia.2 <- sum(quetal.intertiaCell.2)
quetal.sumDiag.2 <- sum(diag(as.matrix(quetal.intertiaCell.2)))
quetal.diagIntertia.2 <- quetal.sumDiag.2*100/quetal.totalIntertia.2
quetal.diagIntertia.2
```

```
## [1] 7.532618e-11
```

As we can see the value is much lower than before so the diagonal is not overloaded and the problem its solved. ##5. Perform a new CA upon the quetaltecaen table, with the modified diagonal and interpret the results.

```
ca.quetal.2 <- CA(quetal.data.2)
```



The first thing we can clearly see is the relation of the CCAA with themselves is quite different than before. In the first CA, the relation of a CCAA with itself was huge, now that the diagonal is modified shows other results that are not biased by the selfrelation that we had before.

If we analyze deeper, catalanes and vascos are far from the center of the cloud, which means that people from other CCAA don't like neither of them.

But we can say that catalan people like vascos, and Valencia and Euzkadi like catalanes.

The cloud of points is quite big but there is no clear difference apart from Catalunya and Euzkadi, which means that all the other CCAA get along well with each other.

This makes sense with the reality as Catalonia and Euzkadi have deep country feeling that is not shared with any other CCAA. We know that they are the communities that want the independece from Spain and this is probably the fact that make this 2 communities get allong well. We could also try to explain whit Valencia as they share language with catalonia(more or less), so this can make them have good relation.

6. Read the file ???mca_car.csv??? containing the data and its dictionary about the cars and their characteristics found in specialized magazines. The final goal will be to find a model to predict the price of cars as function of its characteristics. First we will perform a visualization of the information contained in the dataset, then we will perform a clustering of cars. The data has been previously preprocessed to have it in categorical form.

```r
car <- read.csv("mca_car.csv", sep=";")
car.data <- car[-1]
row.names(car.data) <- car$iden
summary(car.data)
```

```
##                  cilindrada           potencia     combustible
##  Cil_(1.5e+03,1.8e+03]:124   Pot_(105,130]: 99   Diesel  : 84
##  Cil_(1.8e+03,2e+03]  :118   Pot_(130,180]:109   Gasolina:406
##  Cil_(2.6e+03,8e+03]  : 94   Pot_(180,500]: 89
##  Cil_(2e+03,2.6e+03]  : 67   Pot_(35,75]  : 96
##  Cil_(900,1.5e+03]    : 87   Pot_(75,105] : 97
##
##
##                 revoluciones  cilindros           longitud
##  Rev_(3.8e+03,5e+03]   :108   Ncil_4:358   Long_(300,400]:118
##  Rev_(5.5e+03,5.7e+03]: 62   Ncil_5: 24   Long_(400,430]: 80
##  Rev_(5.7e+03,6e+03]   :131   Ncil_6: 78   Long_(430,450]:134
##  Rev_(5e+03,5.5e+03]   :126   Ncil_8: 30   Long_(450,480]:123
##  Rev_(6e+03,7.5e+03]   : 63                Long_(480,550]: 35
##
##
##             ancho               altura              maletero
##  Anch_(140,160]: 69   Alt_(110,136]:107   Malet_(280,400]:115
##  Anch_(160,168]:117   Alt_(136,139]:133   Malet_(400,450]: 76
##  Anch_(168,170]:116   Alt_(139,141]:119   Malet_(450,500]: 98
##  Anch_(170,175]: 92   Alt_(141,143]: 71   Malet_(500,700]:101
##  Anch_(175,200]: 96   Alt_(143,200]: 60   Malet_[0,280]  :100
##
##
##                    peso         plazas          velocidad
##  Pes_(1.2e+03,1.4e+03]:110   Plaz_2: 31   Vel_(110,170]:113
##  Pes_(1.4e+03,2.5e+03]: 87   Plaz_4: 21   Vel_(170,185]: 99
##  Pes_(1e+03,1.2e+03]  :142   Plaz_5:406   Vel_(185,200]: 89
##  Pes_(640,940]        :102   Plaz_7: 32   Vel_(200,220]:107
##  Pes_(940,1e+03]      : 49                Vel_(220,350]: 82
##
##
##        poca_aceleracion      traccion              consumo
##  Acel_(11,13.5]: 85     4x4      : 38   Cons_(11.3,20] : 92
##  Acel_(13.5,25]: 97     Delantera:312   Cons_(4.5,7.6] :104
##  Acel_(4.5,8.3]:107     Trasera  :140   Cons_(7.6,8.5] :100
##  Acel_(8.3,9.7]: 92                     Cons_(8.5,9.5] : 92
##  Acel_(9.7,11] :109                     Cons_(9.5,11.3]:102
##
##
##             coste.Km       precio             marca        precio_categ
```

```
##  Cost_(11.5,13.5]:101   Min.   :  865   MERCEDES  : 43   cheap    :123
##  Cost_(13.5,15]  : 84   1st Qu.: 1803   RENAULT   : 41   expensive:121
##  Cost_(15,17.5]  :106   Median : 2794   VOLKSWAGEN: 39   luxury   :108
##  Cost_(17.5,30]  : 91   Mean   : 4104   PEUGEOT   : 35   medium   :138
##  Cost_(6.5,11.5] :108   3rd Qu.: 4726   OPEL      : 34
##                         Max.   :50000   FORD      : 31
##                                         (Other)   :267
```

```r
car.data$precio <- as.numeric(car.data$precio)
```

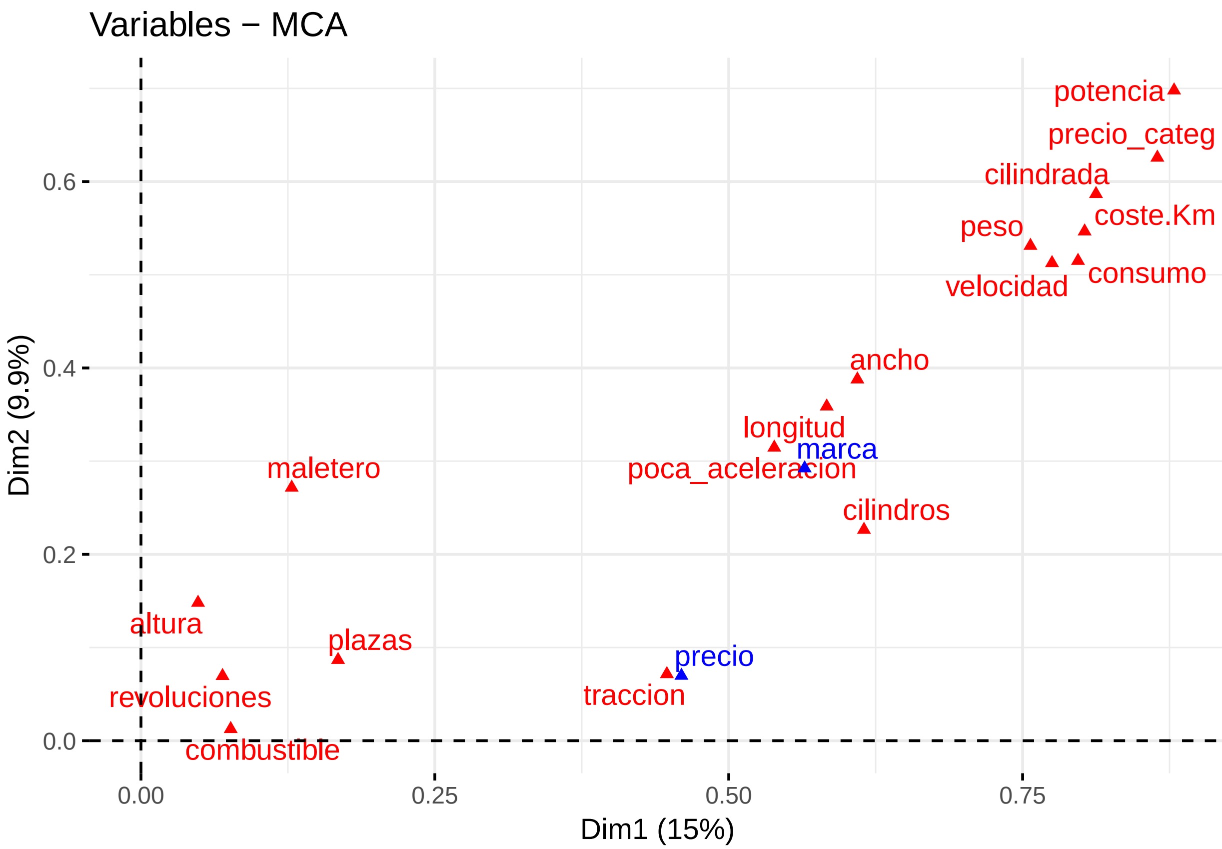
The first step is to read the data. This data had a wrong charachter that made a row have strange results. After solving this we will change precio as numeric row that has the price of a car. Then, once the data is currently prepared we will start to visualize it through a MCA. This will show us information about the importance of each feature to each dimension.

7. With the obtained data frame perform a Multiple Correspondence Analysis. Take the brand and price (either categorical or continuous) as supplementary variables, whereas the remaining ones are active.

To do the MCA, first we need to set Brand(column 18) and Price(column 17) as suplementary variables. After it we will execute MCA function to perform the anlaysis and see the impacto from each one of the features compared to dimensions 1 and 2.

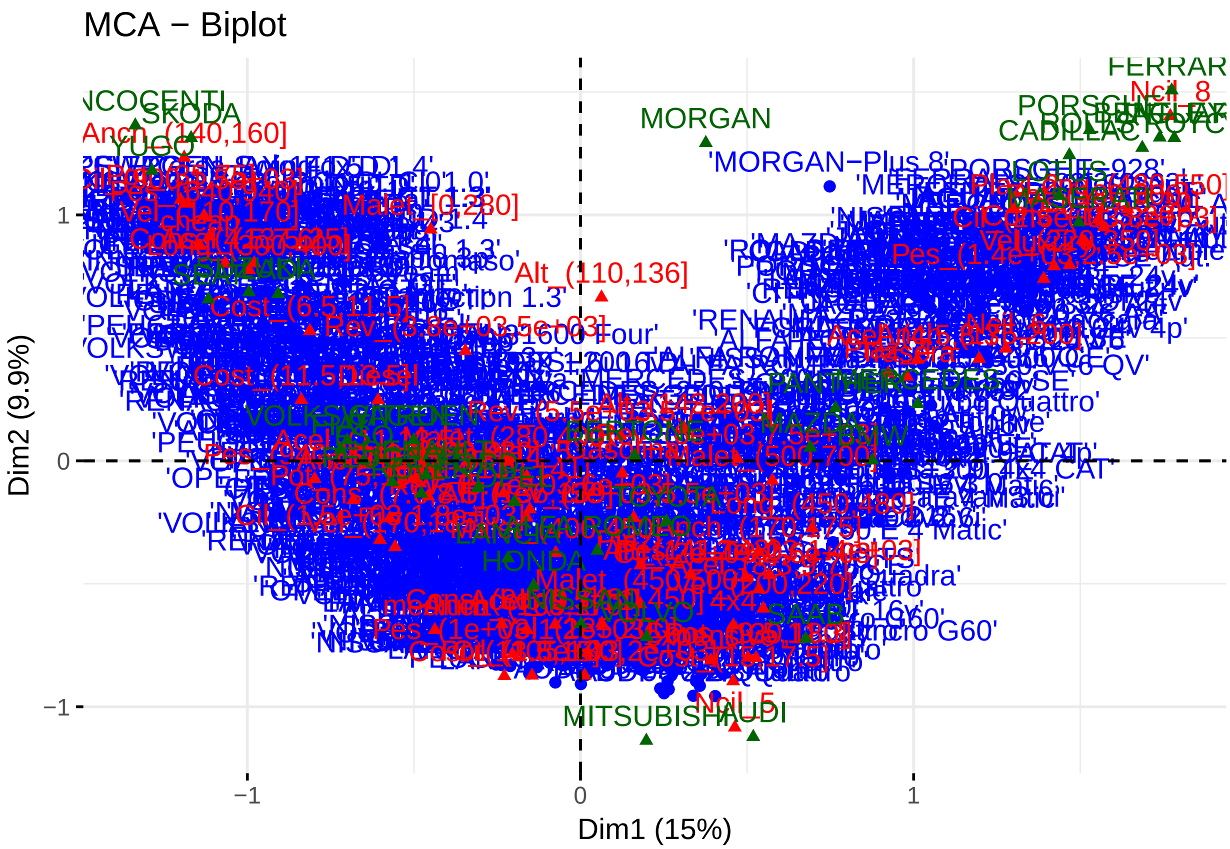
```r
mca.car <- MCA(car.data, ncp = 19,  quali.sup = c(18), quanti.sup = c(17), graph=FALSE)
```

```r
fviz_mca_var(mca.car, choice = "mca.cor",
             repel = TRUE,
             ggtheme = theme_minimal())
```



In this graphic we can see that only 25% of the information retained is explained by the first 2 dimensions, so this means there are other a lot of aspects are not explained by these 2 dimensions. Another thing we have to see here is that potencia, cilindrada, consumo/costeKm are the ones which have more impact on the price. Makes sense as the most expensive cars usually have good engines.

```
fviz_mca(mca.car)
```



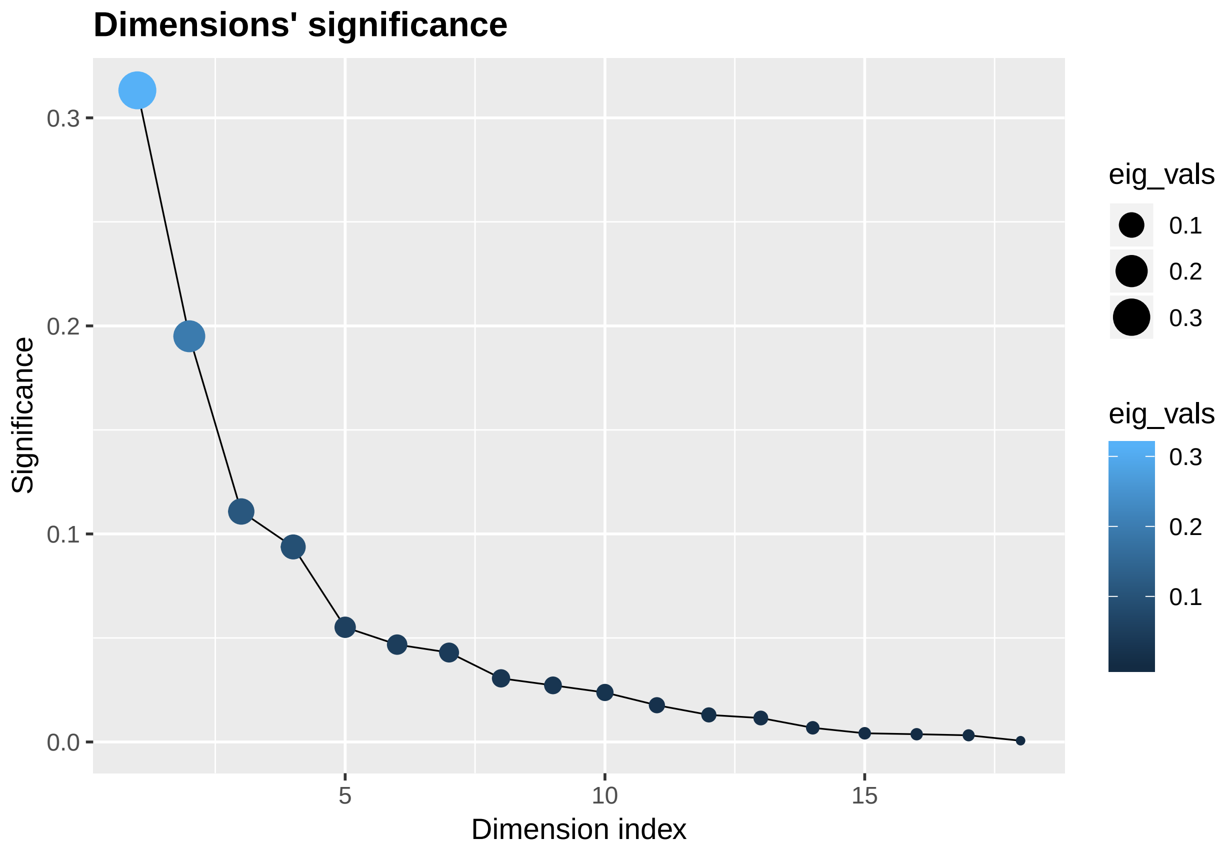
8. **Interpret the first two obtained factors.**

9. **Decide the number of significant dimensions that you retain (by subtracting the average eigenvalue and represent the new obtained eigenvalues in a new screeplot).**

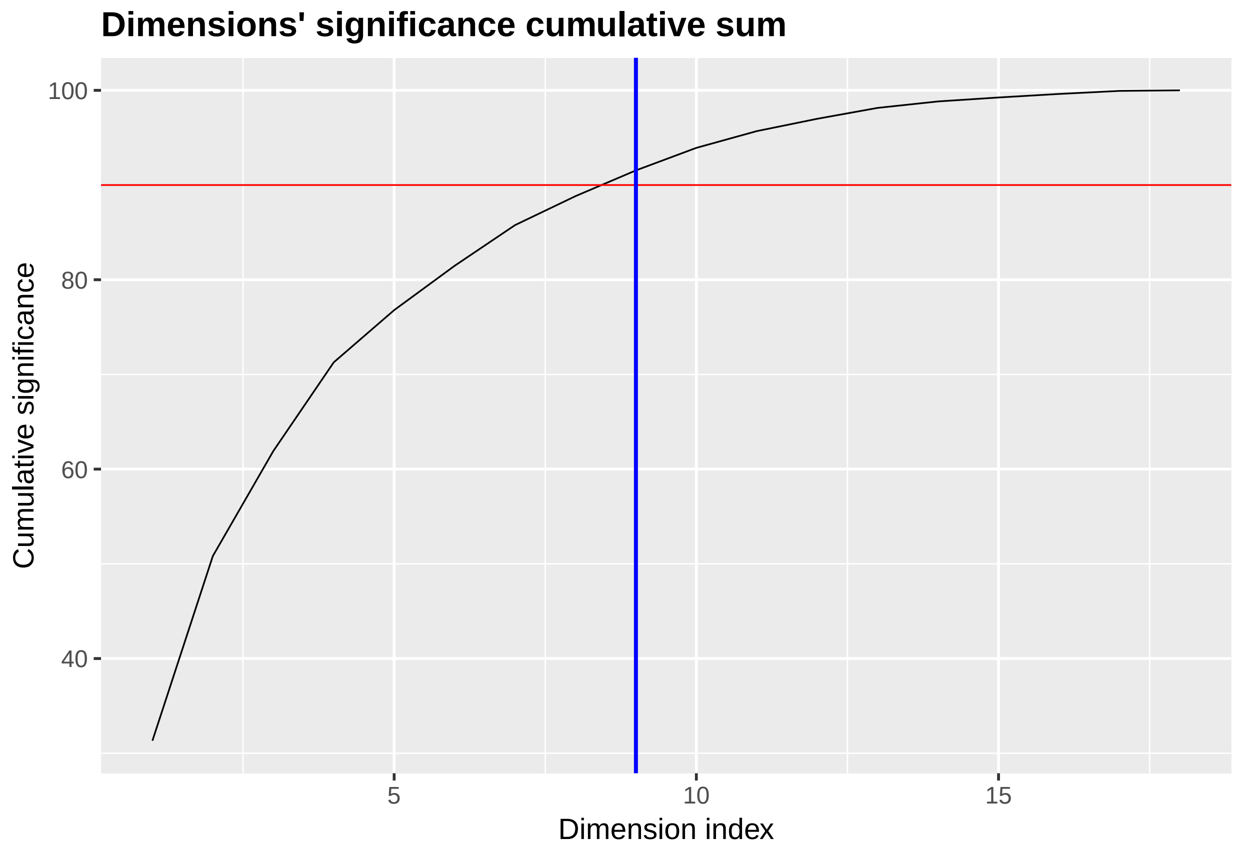
```
eigen_values <- as.data.frame(mca.car$eig)$eigenvalue
mean_eigen <- mean(eigen_values)
eigen_values <- eigen_values[as.data.frame(mca.car$eig)$eigenvalue > mean_eigen]
eigen_values <- eigen_values - mean_eigen
```

```
eig_vals = eigen_values/sum(eigen_values)
eig_df = as.data.frame(cbind(eig_vals, X = seq(length(eig_vals))))
```

```
ggplot(eig_df, aes(x=X, y=eig_vals)) +
    geom_line(size=0.3) +
    geom_point(aes(size = eig_vals, colour = eig_vals)) +
    ggtitle("Dimensions' significance") +
    ylab("Significance") +
    xlab("Dimension index") +
    theme(plot.title = element_text(lineheight=.8, face="bold"))
```



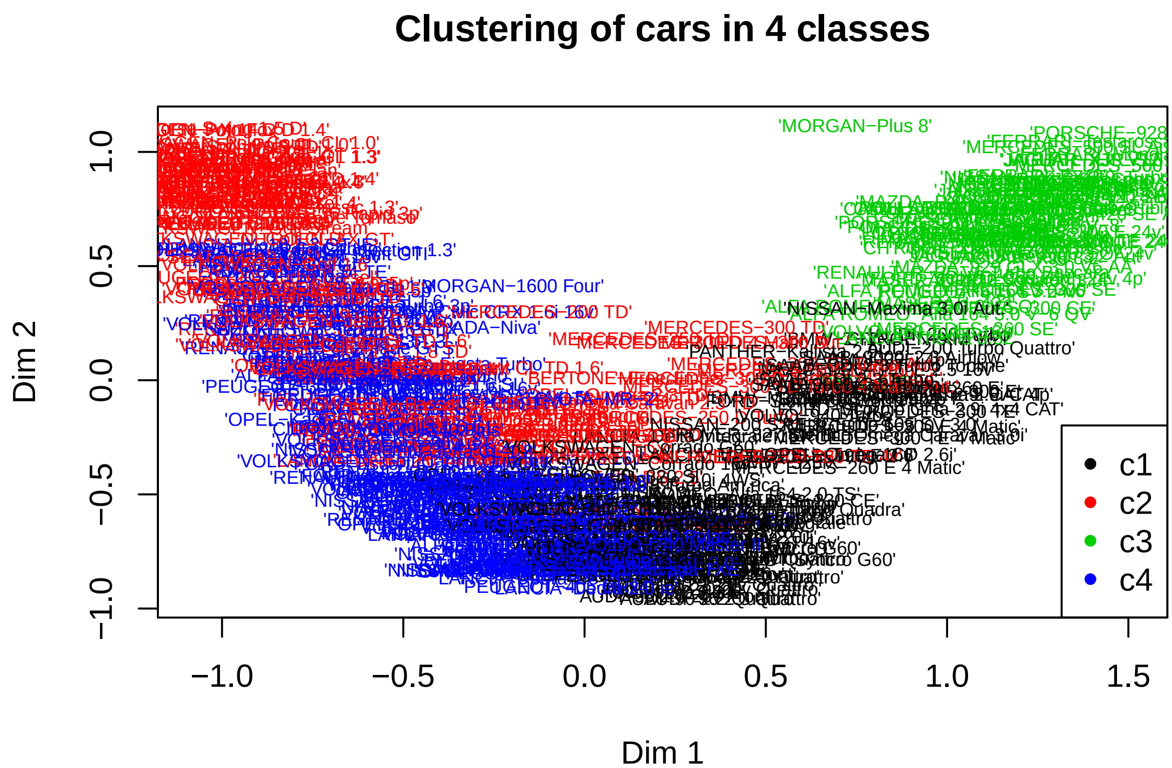

```
cumSum = cumsum(100*eigen_values/sum(eigen_values))
cumSumDF = as.data.frame(cbind(cumSum, X = seq(length(cumSum))))


ggplot(cumSumDF, aes(x=X, y=cumSum)) +
    #geom_area() +
    geom_line(size=0.3) +
    #geom_point(aes(size = cumSum, colour = cumSum)) +
    ggtitle("Dimensions' significance cumulative sum") +
    ylab("Cumulative significance") +
    xlab("Dimension index") +
   theme(plot.title = element_text(lineheight=.8, face="bold")) +
    geom_hline(yintercept = 90, color = "red", size = 0.3) + # Threshold
    geom_vline(xintercept = 9, color = "blue", size=0.7)
```



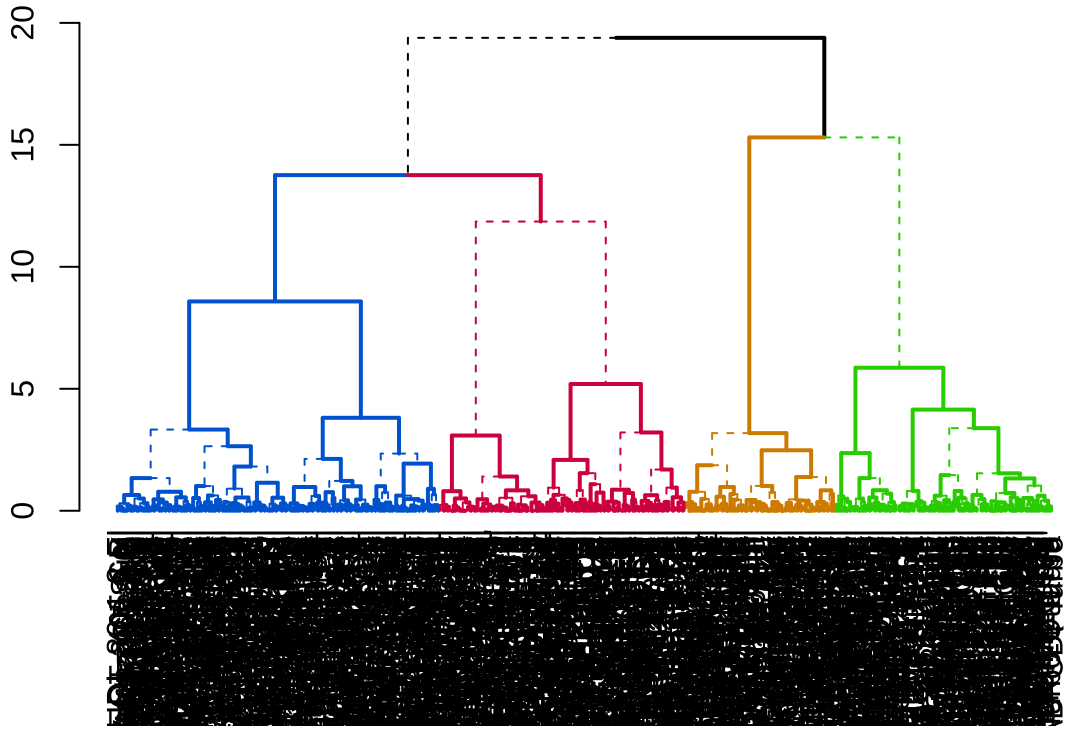
10. Perform a hierarchical clustering with the significant factors, decide the number of final classes to obtain and perform a consolidation operation of the clustering.

```
Psi <- as.matrix(mca.car$ind$coord[, 1:4])
dist_matrix = dist(Psi)
cluster <- hclust(dist_matrix, method='ward.D2')
barplot(cluster$height)
```

```r
number_clusters = 4
c1 <- cutree(cluster, number_clusters)
plot(Psi,type="n",main="Clustering of cars in 4 classes")
text(Psi,col=c1,labels=rownames(Psi),cex = 0.6)
legend("bottomright",c("c1","c2","c3","c4"),pch=20,col=c(1:4))
```



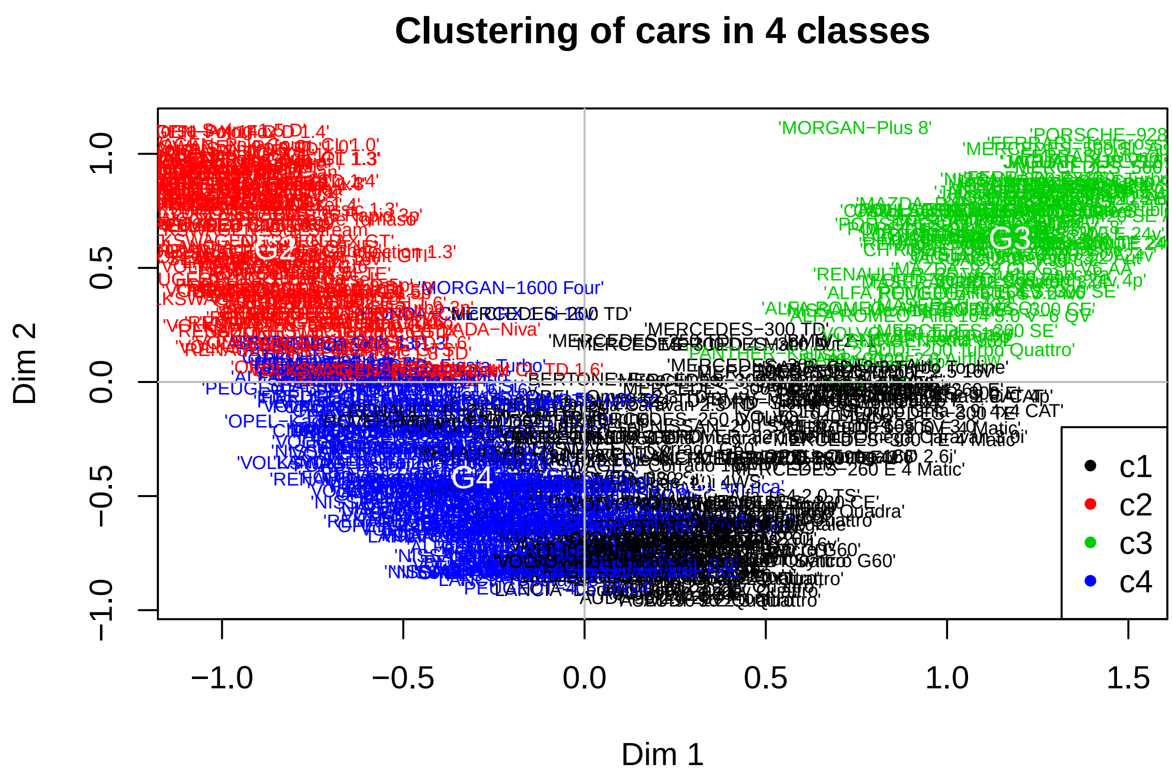

```r
colors = hsv(c(0.6, 0.95, 0.1, 0.3), 1, 0.8, 1)
dend <- as.dendrogram(cluster)
dend <- dend %>%
    color_branches(k = number_clusters, col=colors) %>%
    set("branches_lwd", c(2,1,2)) %>%
    set("branches_lty", c(1,2,1))

plot(dend)
```



```r
cdg <- aggregate(Psi,list(c1),mean)[,2:(4+1)]
```

```r
# And consolidate the clustering using k-means
# to avoid overlapping conditions between successive nodes
k_def <- kmeans(Psi,centers=cdg)

# SAME AS BEFORE
plot(Psi,type="n",main="Clustering of cars in 4 classes")
text(Psi,col=k_def$cluster,labels=rownames(Psi),cex = 0.6)
abline(h=0,v=0,col="gray")
legend("bottomright",c("c1","c2","c3","c4"),pch=20,col=c(1:4))
text(k_def$centers,labels=c("G1","G2", "G3", "G4"),col="white", face="bold")
```

```
## Warning in text.default(k_def$centers, labels = c("G1", "G2", "G3",
## "G4"), : "face" is not a graphical parameter
```

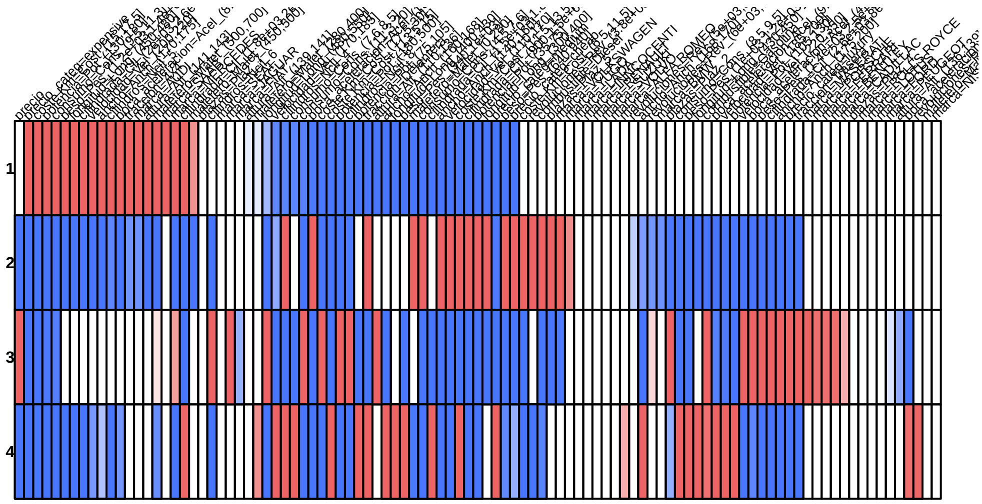


11. Using the function catdes interpret and name the obtained clusters and represent them in the first factorial display.

```r
a <- catdes(cbind(as.factor(k_def$cluster),car),1, 0.05)
a$quanti
```

```
## $`1`
## NULL
##
## $`2`
##           v.test Mean in category Overall mean sd in category Overall sd
## precio -7.265282         1513.586     4103.808       391.4615   4390.683
##             p.value
## precio 3.722606e-13
##
## $`3`
##          v.test Mean in category Overall mean sd in category Overall sd
## precio 14.27364         10246.43     4103.808       7329.992   4390.683
##             p.value
## precio 3.194734e-46
##
## $`4`
##           v.test Mean in category Overall mean sd in category Overall sd
## precio -5.932922          2355.51     4103.808       605.9226   4390.683
##             p.value
## precio 2.975907e-09
```

```
plot(a)
```



```
plot(mca.car$ind$coord, col=k_def$cluster)
```