Tarea_7

Juliana Quirós, Alberto

Realizo el ACP con la base de datos "mtcars".

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

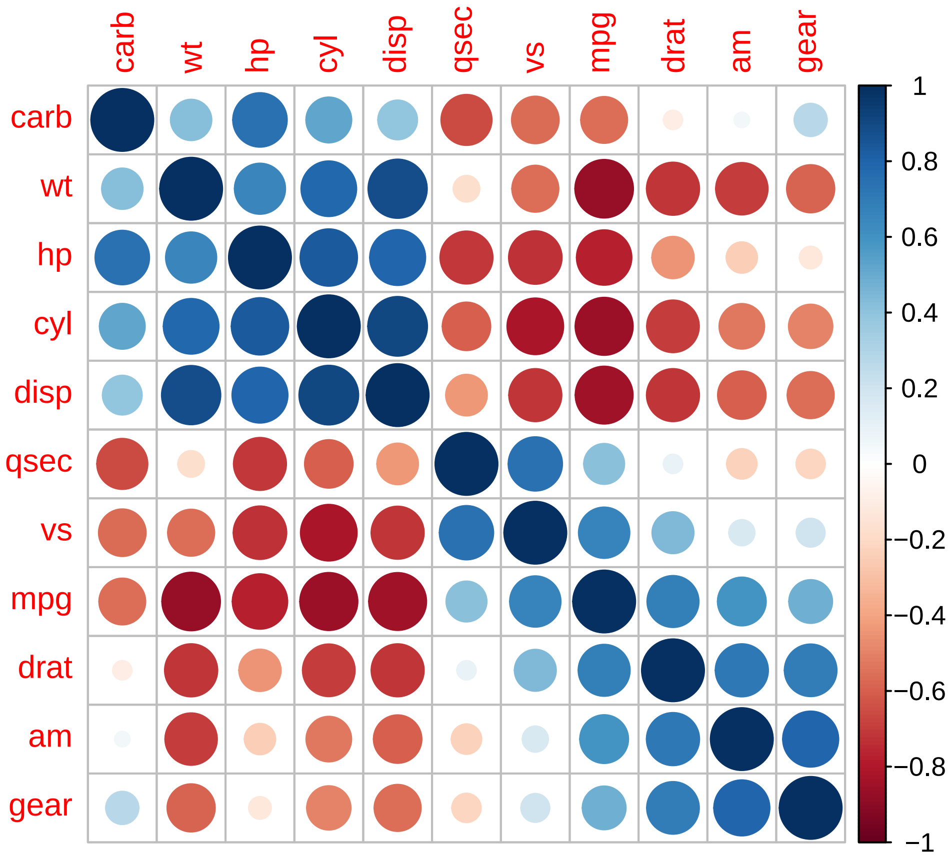
```
library(corrr)
library(DT)
library(hornpa)
# Datos de tendencia central-medias
apply(mtcars, 2, mean)
```

```
##        mpg        cyl       disp         hp       drat         wt       qsec
##  20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
##         vs         am       gear       carb
##   0.437500   0.406250   3.687500   2.812500
```

```
# ¿hay algún valor perdido?no
colSums(is.na(mtcars))
```

```
##  mpg  cyl disp   hp drat   wt qsec   vs   am gear carb
##    0    0    0    0    0    0    0    0    0    0    0
```

```
# se pinta la matriz de correlaciones
# de las columnas de mtcars
corrplot(cor(mtcars), order = "hclust")
```




```r
# tiene que haber cierta colinealidad
# para extraer Componentes. ejecutar
# componentes con la correlación de las
# variables Toma como argumento la
# matriz de puntuaciones, no la de
# correlaciones ejecuta internamente la
# matriz de correlaciones asume también
# las columnas como variables
pca.res <- prcomp(mtcars, scale = TRUE)
# saturaciones de cada variable en cada
# factor las columnas son los
# autovectores de la descomposición en
# Autovectores/autovalores
pca.res$rotation
```

```
##             PC1         PC2         PC3          PC4         PC5         PC6
## mpg  -0.3625305  0.01612440 -0.22574419 -0.022540255  0.10284468 -0.10879743
## cyl   0.3739160  0.04374371 -0.17531118 -0.002591838  0.05848381  0.16855369
## disp  0.3681852 -0.04932413 -0.06148414  0.256607885  0.39399530 -0.33616451
## hp    0.3300569  0.24878402  0.14001476 -0.067676157  0.54004744  0.07143563
## drat -0.2941514  0.27469408  0.16118879  0.854828743  0.07732727  0.24449705
## wt    0.3461033 -0.14303825  0.34181851  0.245899314 -0.07502912 -0.46493964
## qsec -0.2004563 -0.46337482  0.40316904  0.068076532 -0.16466591 -0.33048032
## vs   -0.3065113 -0.23164699  0.42881517 -0.214848616  0.59953955  0.19401702
## am   -0.2349429  0.42941765 -0.20576657 -0.030462908  0.08978128 -0.57081745
## gear -0.2069162  0.46234863  0.28977993 -0.264690521  0.04832960 -0.24356284
## carb  0.2140177  0.41357106  0.52854459 -0.126789179 -0.36131875  0.18352168
##               PC7          PC8          PC9        PC10         PC11
## mpg   0.367723810 -0.754091423  0.235701617  0.13928524 -0.124895628
## cyl   0.057277736 -0.230824925  0.054035270 -0.84641949 -0.140695441
## disp  0.214303077  0.001142134  0.198427848  0.04937979  0.660606481
## hp   -0.001495989 -0.222358441 -0.575830072  0.24782351 -0.256492062
## drat  0.021119857  0.032193501 -0.046901228 -0.10149369 -0.039530246
```

```
## wt   -0.020668302 -0.008571929  0.359498251  0.09439426 -0.567448697
## qsec  0.050010522 -0.231840021 -0.528377185 -0.27067295  0.181361780
## vs   -0.265780836  0.025935128  0.358582624 -0.15903909  0.008414634
## am   -0.587305101 -0.059746952 -0.047403982 -0.17778541  0.029823537
## gear  0.605097617  0.336150240 -0.001735039 -0.21382515 -0.053507085
## carb -0.174603192 -0.395629107  0.170640677  0.07225950  0.319594676
```

```
# para ver los autovalores
pca.var = pca.res$sdev^2
pca.var
```

```
##  [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612
##  [7] 0.13526199 0.12290143 0.07704665 0.05203544 0.02204441
```

```
# se muestra el ratio de cada autovalor
var.ratio = pca.var/sum(pca.var)
var.ratio
```

```
##  [1] 0.600763659 0.240951627 0.057017934 0.024508858 0.020313737 0.019236011
##  [7] 0.012296544 0.011172858 0.007004241 0.004730495 0.002004037
```

```
# tomamos solo dos componentes. Los más
# importantes
pca.res <- prcomp(mtcars, scale = TRUE, rank = 2)
pca.res$rotation
```

```
##             PC1         PC2
## mpg  -0.3625305  0.01612440
## cyl   0.3739160  0.04374371
## disp  0.3681852 -0.04932413
## hp    0.3300569  0.24878402
## drat -0.2941514  0.27469408
## wt    0.3461033 -0.14303825
## qsec -0.2004563 -0.46337482
## vs   -0.3065113 -0.23164699
## am   -0.2349429  0.42941765
## gear -0.2069162  0.46234863
## carb  0.2140177  0.41357106
```

```
# las puntuaciones de los ejemplares
# (los vehículos) en los componentes
pca.res$x
```

```
##                                PC1        PC2
## Mazda RX4           -0.6468627420  1.7081142
## Mazda RX4 Wag       -0.6194831460  1.5256219
## Datsun 710          -2.7356242748 -0.1441501
## Hornet 4 Drive      -0.3068606268 -2.3258038
## Hornet Sportabout    1.9433926844 -0.7425211
## Valiant             -0.0552534228 -2.7421229
## Duster 360           2.9553851233  0.3296133
## Merc 240D           -2.0229593244 -1.4421056
## Merc 230            -2.2513839535 -1.9522879
## Merc 280            -0.5180912217 -0.1594610
## Merc 280C           -0.5011860079 -0.3187934
## Merc 450SE           2.2124096339 -0.6727099
## Merc 450SL           2.0155715693 -0.6724606
## Merc 450SLC          2.1147047372 -0.7891129
## Cadillac Fleetwood   3.8383725118 -0.8149087
```

```
## Lincoln Continental  3.8918495626 -0.7218314
## Chrysler Imperial    3.5363862158 -0.4145024
## Fiat 128            -3.7955510831 -0.2920783
## Honda Civic         -4.1870356784  0.6775721
## Toyota Corolla      -4.1675359344 -0.2748890
## Toyota Corona       -1.8741790870 -2.0864529
## Dodge Challenger     2.1504414942 -0.9982442
## AMC Javelin          1.8340369797 -0.8921886
## Camaro Z28           2.8434957523  0.6701037
## Pontiac Firebird     2.2105479148 -0.8600504
## Fiat X1-9           -3.5176818134 -0.1192950
## Porsche 914-2       -2.6095003965  2.0141425
## Lotus Europa        -3.3323844512  1.3568877
## Ford Pantera L       1.3513346957  3.4448780
## Ferrari Dino        -0.0009743305  3.1683750
## Maserati Bora        2.6270897605  4.3107016
## Volvo 142E          -2.3824711412  0.2299603
```

En nuestro caso, hemos de modificar el argumento "k" a 11, ya que tenemos 11 variables.

```
# ¿cómo se hace el test paralelo para
# ver cuantos retener? comparando los
# autovalores de la simulación de la
# función hornpa (0.95) con los que ha
# arrojado nuestro análisis pca.var

simulacion <- hornpa(k = 11, size = 50, reps = 500,
    seed = 1234)
```

```
##
##  Parallel Analysis Results
##
## Method: pca
## Number of variables: 11
## Sample size: 50
## Number of correlation matrices: 500
## Seed: 1234
## Percentile: 0.95
##
## Compare your observed eigenvalues from your original dataset to the 95 percentile in the table below generate
##
##  Component  Mean  0.95
##          1 1.840 2.089
##          2 1.567 1.714
##          3 1.382 1.504
##          4 1.218 1.330
##          5 1.076 1.175
##          6 0.943 1.046
##          7 0.819 0.913
##          8 0.707 0.805
##          9 0.594 0.689
##         10 0.486 0.580
##         11 0.368 0.468
```

```
pca.var
```

```
##  [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612
##  [7] 0.13526199 0.12290143 0.07704665 0.05203544 0.02204441
```

```r
# inciso para hacer
# autovectores/autovalores en la de
# correlaciones
ev <- eigen(cor(mtcars))
ev$values
```

```
##  [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612
##  [7] 0.13526199 0.12290143 0.07704665 0.05203544 0.02204441
```

```r
ev$vectors
```

```
##              [,1]        [,2]        [,3]         [,4]        [,5]        [,6]
##  [1,]  0.3625305 -0.01612440 -0.22574419 -0.022540255 -0.10284468 -0.10879743
##  [2,] -0.3739160 -0.04374371 -0.17531118 -0.002591838 -0.05848381  0.16855369
##  [3,] -0.3681852  0.04932413 -0.06148414  0.256607885 -0.39399530 -0.33616451
##  [4,] -0.3300569 -0.24878402  0.14001476 -0.067676157 -0.54004744  0.07143563
##  [5,]  0.2941514 -0.27469408  0.16118879  0.854828743 -0.07732727  0.24449705
##  [6,] -0.3461033  0.14303825  0.34181851  0.245899314  0.07502912 -0.46493964
##  [7,]  0.2004563  0.46337482  0.40316904  0.068076532  0.16466591 -0.33048032
##  [8,]  0.3065113  0.23164699  0.42881517 -0.214848616 -0.59953955  0.19401702
##  [9,]  0.2349429 -0.42941765 -0.20576657 -0.030462908 -0.08978128 -0.57081745
## [10,]  0.2069162 -0.46234863  0.28977993 -0.264690521 -0.04832960 -0.24356284
## [11,] -0.2140177 -0.41357106  0.52854459 -0.126789179  0.36131875  0.18352168
##               [,7]         [,8]         [,9]       [,10]        [,11]
##  [1,]  0.367723810  0.754091423  0.235701617  0.13928524 -0.124895628
##  [2,]  0.057277736  0.230824925  0.054035270 -0.84641949 -0.140695441
##  [3,]  0.214303077 -0.001142134  0.198427848  0.04937979  0.660606481
##  [4,] -0.001495989  0.222358441 -0.575830072  0.24782351 -0.256492062
##  [5,]  0.021119857 -0.032193501 -0.046901228 -0.10149369 -0.039530246
##  [6,] -0.020668302  0.008571929  0.359498251  0.09439426 -0.567448697
##  [7,]  0.050010522  0.231840021 -0.528377185 -0.27067295  0.181361780
##  [8,] -0.265780836 -0.025935128  0.358582624 -0.15903909  0.008414634
##  [9,] -0.587305101  0.059746952 -0.047403982 -0.17778541  0.029823537
## [10,]  0.605097617 -0.336150240 -0.001735039 -0.21382515 -0.053507085
## [11,] -0.174603192  0.395629107  0.170640677  0.07225950  0.319594676
```

```r
############### es la demostracción de
############### que se puede hacer con
############### descomposición
############### directamente
```

```r
## Expansión ##
simulacion
```

```
##    Component  Mean  0.95
## 1          1 1.840 2.089
## 2          2 1.567 1.714
## 3          3 1.382 1.504
## 4          4 1.218 1.330
## 5          5 1.076 1.175
## 6          6 0.943 1.046
## 7          7 0.819 0.913
## 8          8 0.707 0.805
## 9          9 0.594 0.689
## 10        10 0.486 0.580
## 11        11 0.368 0.468
```

```r
pca.var
```

```
##  [1] 6.60840025 2.65046789 0.62719727 0.26959744 0.22345110 0.21159612
##  [7] 0.13526199 0.12290143 0.07704665 0.05203544 0.02204441
```

Tal y como indica nuestra comparación con las simulaciones del análisis paralelo, obtenemos 2 factores. Valoramos saturaciones para comprobar qué variables cargan en cada factor:

```r
library(factoextra)
```

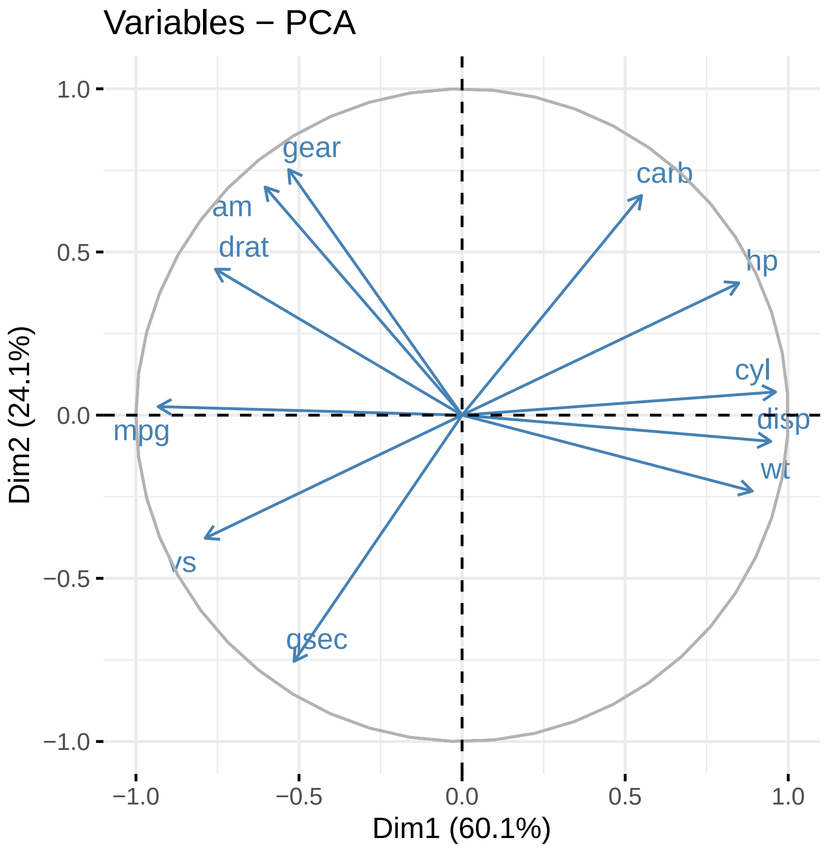
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
summary(pca.res)$rotation
```

```
##             PC1         PC2
## mpg  -0.3625305  0.01612440
## cyl   0.3739160  0.04374371
## disp  0.3681852 -0.04932413
## hp    0.3300569  0.24878402
## drat -0.2941514  0.27469408
## wt    0.3461033 -0.14303825
## qsec -0.2004563 -0.46337482
## vs   -0.3065113 -0.23164699
## am   -0.2349429  0.42941765
## gear -0.2069162  0.46234863
## carb  0.2140177  0.41357106
```

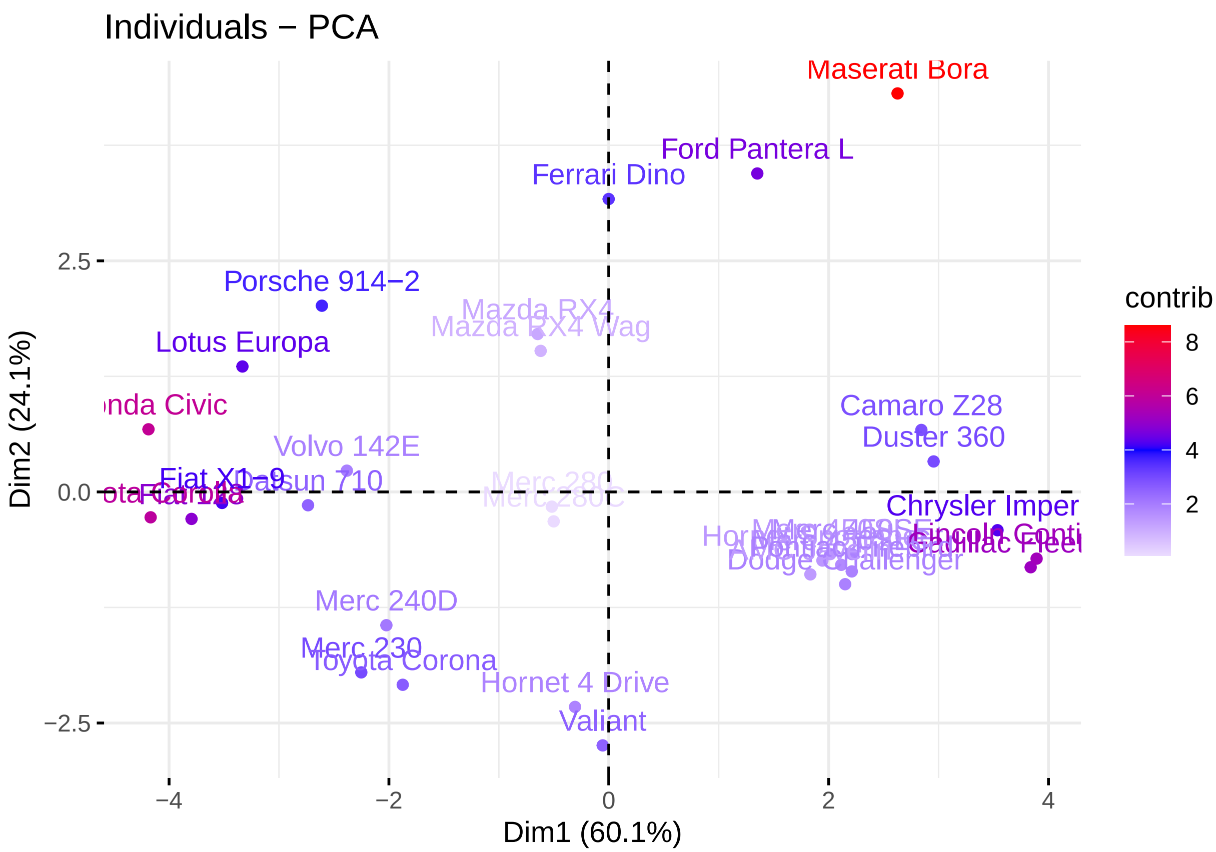
```r
fviz_pca_var(pca.res, col.var = "steelblue",
    repel = TRUE, gradient.cols = c("#00AFBB",
        "#E7B800", "#FC4E07"), ylim = c(-1,
        1))
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```



Como podemos observar, con el factor 1 correlacionan en mayor medida el número de cilindros(cyl) la cilindrada(disp), los caballos(hp) y el peso(wt).
Por otra parte, con el factor 2, el tipo de transmisión(am),marchas(gear) y número de carburadores(carb). El resto de variables parece tener una influencia menor en ambos factores.

```
fviz_pca_ind(pca.res, col.ind = "contrib") +
    scale_color_gradient2(low = "white",
        mid = "blue", high = "red", midpoint = 4)
```



En este último gráfico, observamos qué casos han contribuido en mayor medida a cada factor.