Homework 1: PCA and MDS

Matija Jakovac, Dunja Petrović, Leo Sérra

2025-05-02

```
##   No          TEAM                   PLAYER POSITION GP GS      Min PTS X2P.
## 1  1 PANATHINAIKOS PANAGIOTIS KALAITZAKIS     Guard 30  0  5:56:00 2.1 69.0
## 2  2 PANATHINAIKOS            LUCA VILDOZA    Guard 28  5 14:56:00 5.7 42.0
## 3  3 PANATHINAIKOS                KYLE GUY    Guard  8  1 10:38:00 4.0 71.4
##   X3P.   FT.  OR  DR  TR AST STL  TO BLK BLKA  FC  FD PIR
## 1 25.0 100.0 0.3 0.6 0.9 0.2 0.2 0.2 0.0  0.0 0.8 0.4 2.1
## 2 36.6  76.2 0.4 1.1 1.5 1.5 0.6 1.0 0.0  0.2 0.8 0.6 4.6
## 3 31.6  80.0 0.0 0.9 0.9 0.8 0.2 1.0 0.1  0.0 1.2 0.6 2.4
```

1. First do the exploratory data analysis.

a) Discard the variable "No" from the data set. (1p)

b) Split variable "Min" using strsplit() function. Give the name "aux" to the output. The first element of each row will show the minutes that the player played in total.

```
## Aux:
```

```
## [[1]]
## [1] "5"  "56" "00"
##
## [[2]]
## [1] "14" "56" "00"
##
## [[3]]
## [1] "10" "38" "00"
```

c) Add a numerical variable to the data set named "Min 2" which shows on average how many minutes each player played in the game. (2p)

```
##                    PLAYER      Min2
## 1 PANAGIOTIS KALAITZAKIS   5.933333
## 2            LUCA VILDOZA 14.933333
## 3                KYLE GUY 10.633333
```

d) Check the structure of the data and assign correct type to each variable considering whether it is a categorical or numerical variable. (2p)

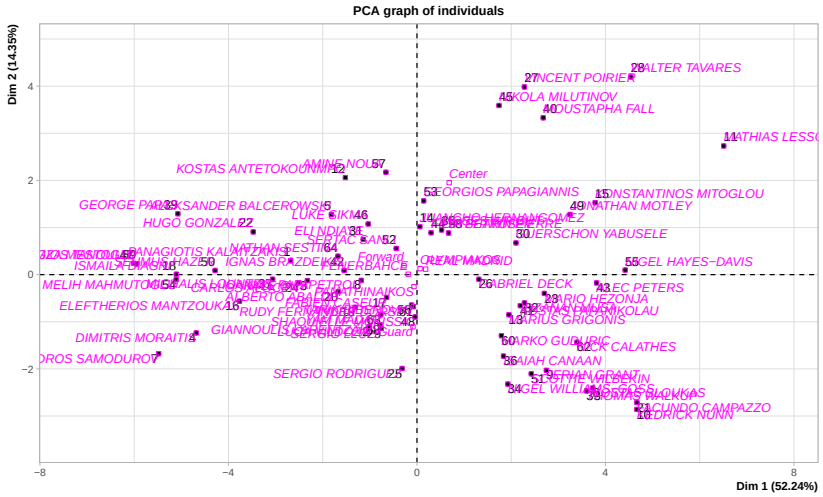
```
## 'data.frame':    64 obs. of  21 variables:
##  $ TEAM    : chr  "PANATHINAIKOS" "PANATHINAIKOS" "PANATHINAIKOS" "PANATHINAIKOS" ...
##  $ PLAYER  : chr  "PANAGIOTIS KALAITZAKIS " "LUCA VILDOZA" "KYLE GUY" "DIMITRIS MORAITIS" ...
##  $ POSITION: chr  "Guard" "Guard" "Guard" "Guard" ...
```

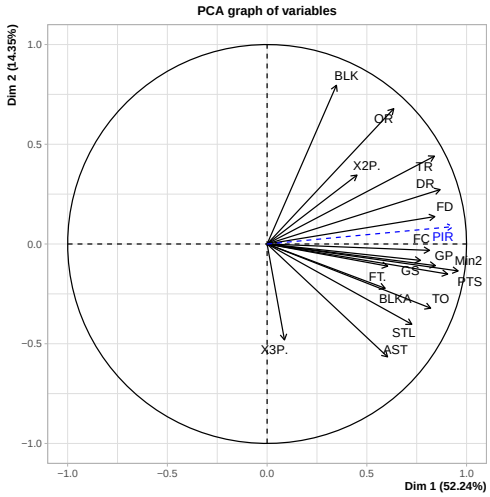
```
##  $ GP      : int  30 28 8 7 24 34 1 16 41 35 ...
##  $ GS      : int  0 5 1 0 9 15 0 4 34 27 ...
##  $ PTS     : num  2.1 5.7 4 1.6 2.8 12.7 3 5.6 8.6 16 ...
##  $ X2P.    : num  69 42 71.4 25 62.9 59.1 0 46.9 49.7 46.6 ...
##  $ X3P.    : num  25 36.6 31.6 75 11.1 41.5 100 51.6 41.6 41 ...
##  $ FT.     : num  100 76.2 80 0 70 85.3 0 80 86.1 95.9 ...
##  $ OR      : num  0.3 0.4 0 0 0.6 0.6 0 0.4 0.5 0.4 ...
##  $ DR      : num  0.6 1.1 0.9 0.3 0.8 2.6 0 1.6 1.8 2.3 ...
##  $ TR      : num  0.9 1.5 0.9 0.3 1.3 3.2 0 2 2.3 2.7 ...
##  $ AST     : num  0.2 1.5 0.8 0.7 0.3 5.6 1 0.7 3.5 3 ...
##  $ STL     : num  0.2 0.6 0.2 0.3 0.2 0.8 0 0.2 1.5 0.9 ...
##  $ TO      : num  0.2 1 1 0.3 0.3 2.4 0 0.4 1.1 3.1 ...
##  $ BLK     : num  0 0 0.1 0 0.4 0 0 0.2 0.1 0.1 ...
##  $ BLKA    : num  0 0.2 0 0.1 0.1 0.4 0 0.2 0.1 0.8 ...
##  $ FC      : num  0.8 0.8 1.2 0.1 1.5 1.8 0 1.4 2.3 2.2 ...
##  $ FD      : num  0.4 0.6 0.6 0 1.2 3 0 0.9 2.1 2.7 ...
##  $ PIR     : num  2.1 4.6 2.4 1.7 3.1 16.1 3 5.4 10.9 11.7 ...
##  $ Min2    : num  5.93 14.93 10.63 2.42 7.6 ...
```

2. Application of PCA.

a) Apply PCA on all the scaled numerical variables in the data set by using PCA() function in FactoMineR package. Treat the categorical variables and the variable "PIR" as suplemantary variables using arguments quali.sup and quanti.sup correctly. (3p)

```
library(FactoMineR)

dffa <- PCA(df, scale.unit = TRUE, ncp = 5, quali.sup = c(1, 2, 3), quanti.sup = 20, graph = TRUE)
```





b) How many components should be extracted? Decide on the number of components considering eigenvalues. (3p)

```
##           eigenvalue percentage of variance cumulative percentage of variance
## comp 1  8.8816066871           52.244745218                           52.24475
## comp 2  2.4393987141           14.349404201                           66.59415
## comp 3  1.1714341258            6.890788975                           73.48494
## comp 4  0.9897481713            5.822048067                           79.30699
## comp 5  0.7554268944            4.443687614                           83.75067
## comp 6  0.5005060239            2.944153082                           86.69483
## comp 7  0.4733502519            2.784413246                           89.47924
## comp 8  0.4538875038            2.669926493                           92.14917
## comp 9  0.3098233318            1.822490187                           93.97166
## comp 10 0.2885977100            1.697633588                           95.66929
## comp 11 0.2142609003            1.260358237                           96.92965
## comp 12 0.1836453503            1.080266766                           98.00992
## comp 13 0.1507320693            0.886659231                           98.89657
## comp 14 0.1084582871            0.637989924                           99.53456
## comp 15 0.0524440553            0.308494443                           99.84306
## comp 16 0.0262661070            0.154506512                           99.99757
## comp 17 0.0004138168            0.002434216                          100.00000
```

```
##
## Components with eigenvalues larger than 1:
```

```
## comp 1 comp 2 comp 3
##      1      2      3
```

According to the Kaiser criterion (eigenvalue > 1), the first 3 components should be extracted.

c) Interpret the loadings/correlations of variables at each dimension (3p).

```
##           Dim.1       Dim.2       Dim.3       Dim.4        Dim.5
## GP   0.84380372 -0.10846063  0.22547222  0.03831107 -0.113006380
## GS   0.76901853 -0.08223333 -0.09483708  0.01393646 -0.279013697
## PTS  0.90541804 -0.15069027 -0.09323276  0.15399198  0.164390903
## X2P. 0.45068068  0.34544820  0.61190896 -0.16969180  0.201191616
## X3P. 0.08708299 -0.48105818  0.14865510  0.81451706  0.008421222
## FT.  0.60460421 -0.11108786  0.61420715 -0.07016205  0.243822653
```

```
## OR   0.63500968  0.67843610 -0.12507397  0.12547408 -0.055196942
## DR   0.86750144  0.27276860 -0.06264887  0.19720235 -0.042242691
## TR   0.83889139  0.44034594 -0.08680832  0.18422301 -0.053740238
## AST  0.60404048 -0.56591092 -0.08312076 -0.25569270 -0.274240419
## STL  0.72557389 -0.40246089  0.04930451 -0.05298973 -0.230889850
## TO   0.82008546 -0.32171577 -0.19595109 -0.24052778  0.063449655
## BLK  0.34756880  0.79575527 -0.08182791  0.05259912 -0.127141957
## BLKA 0.59185153 -0.22221173 -0.39658484  0.02193220  0.594464771
## FC   0.81438875 -0.03197934  0.17103319 -0.11918160 -0.065722057
## FD   0.84084104  0.13754485 -0.24624840 -0.17534692  0.143001482
## Min2 0.95783899 -0.13495801  0.02053780  0.06638688 -0.056566493
```

Dimension 1 –> positive correlation with all variables, strongest correlation with Min2 and PIR

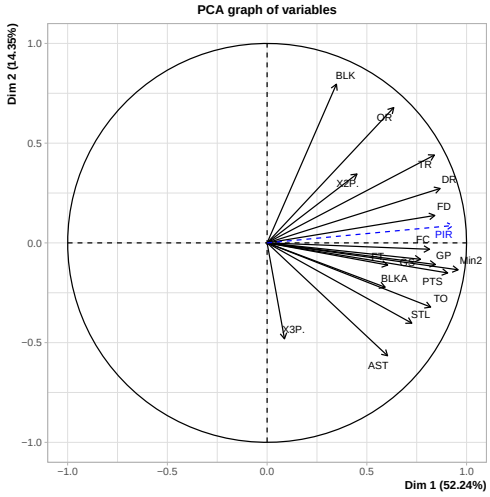
Dimension 2 –> strong positive correlation with X3P. . . strong negative correlations with BLK and OR

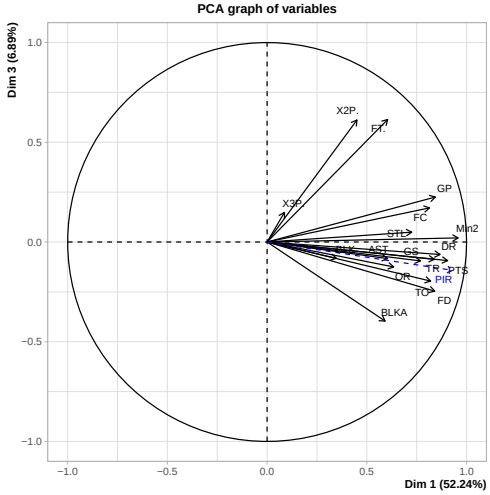
Dimension 3 –> strong positive correlation with X2P. and FT.

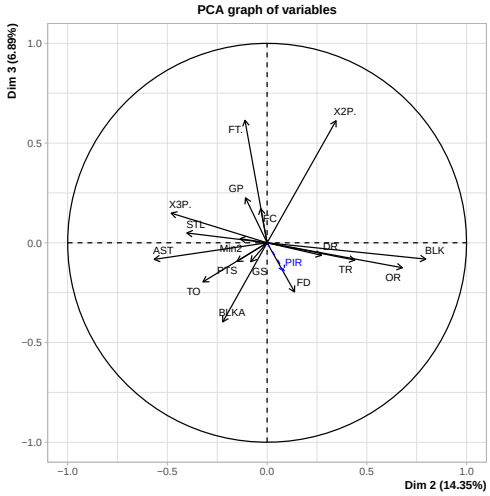
Dimension 4 –> strong positive correlation with X3P., no other particularly strong positive/negative correlations

Dimension 5 –> no particularly strong positive or negative correlations (0.49 positive correlation with BLKA)

d) Use plot.PCA() function to show correlations between variables and the extracted dimensions. (For the variables you should use the argument choix = "var"). Plot all the extracted dimensions changing argument "axes".(3p)







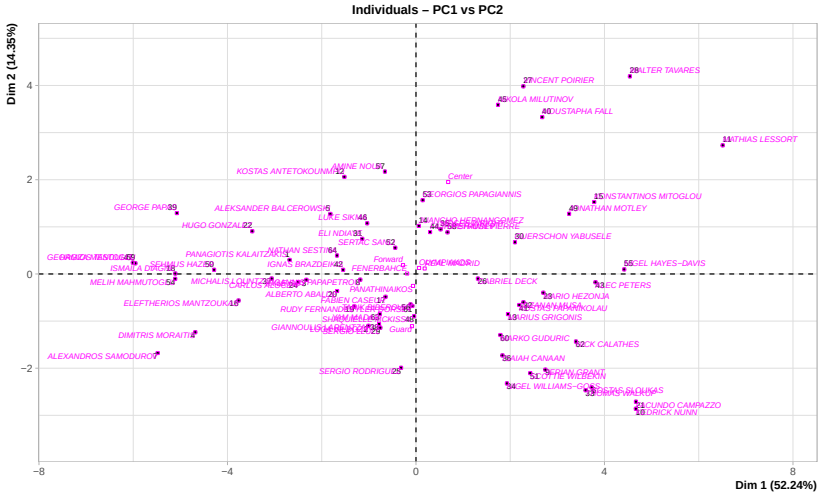
e) Interpret variable plots. How can each dimension be named?

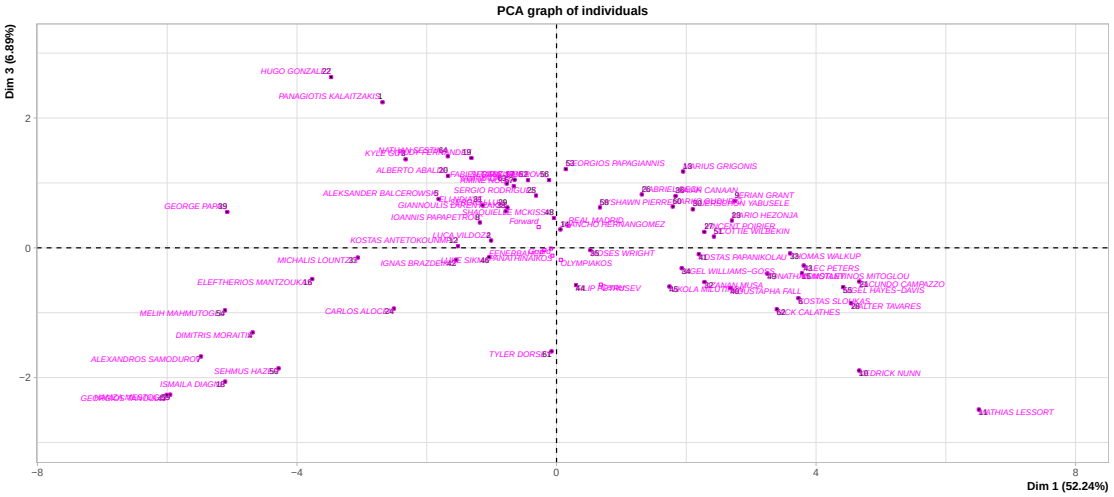
Dimension 1 (Usage/Production volume) - pointing strongly to the positive side are GP, GS, Min2, PTS, TR/DR, PIR, TO, FD and mainly all variables except maybe X3P. which smallest positive correlation with the dimension - players that do appear more on that side are the ones who play a lot and contributed a lot to the team having high box-score numbers such as points, rebounds, fouls and turnovers which happen more often playing more minutes - low-values of those numbers appaers to the bench players

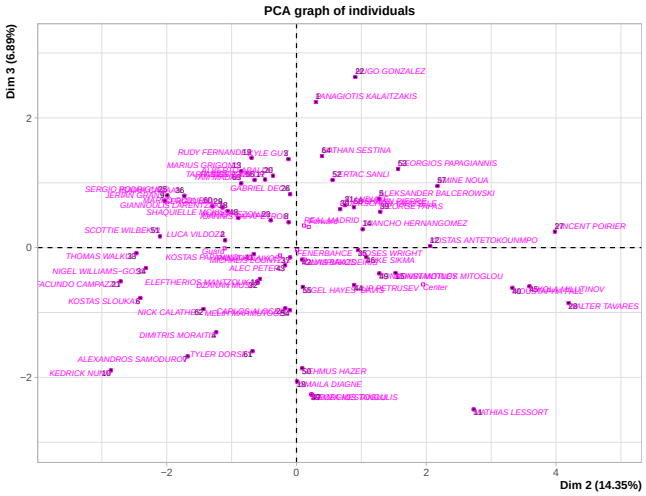
Dimension 2 (Perimeter vs Interior) - having negative correlation with the dimension are X3P., AST and STL while positive correlation are for variable OR, BLK and TR - this correlations represent separation between perimeter play with outside shooting and play-making that are often characteristics of guards versus presence in the paint with rebounding and blocks often connected with centers

Dimension 3 (Finishing efficiency) - X2P. and FT have the most positive correlation with dimension while FD, PIR and BLKA have the most negative ones - having higher 2-point conversion percentage as well as free throw percentage show that player has high efficiency while on the other side if player gets blocked (BLKA) his efficiency will drop

f) Show individual pilots for the extracted dimensions changing argumennt choix="ind" in plot.PCA() function.







g) Interpret the individual plots.

The individual plots show the same correlations as the variable plots but examining rather individuals players.

PC 1 vs PC2

High PC1 and PC2 -> we see high-usage and perimeter oriented players mostly guards that play a lot of minutes, shoot threes and playmake. The examples are Kendrick Nunn, Facundo Campazzo, Kostas Sloukas, Thomas Walkup.

High PC1 and low PC2 -> high-usage players but playing in the interior. Those type of players are starting centers who rebound, block and draw fouls. The examples are Mathias Lessort and Walter Tavares. Low PC1 and high PC2 -> low minute perimeters reserves such as Alexandros Samodurov and Dimitris Moraitis. Low PC1 and PC2 -> low minute interior players such as Ismaila Diagme and Georgios Tsalmpouris.

PC1 vs PC3

High PC3 -> players that are efficient finishers that are rarely blocked such as Hugo Gonzalez and Panagiotis Kalaitzakis. Low PC3 -> are the ones with lower percentages and more shots blocked such as Mathias Lessort despite high usage and Tyler Dorsey.

PC2 vs PC3

High PC2 and PC3 -> efficient and perimeter skills - Rudy Fernandez High PC2 and low PC3 -> not that efficient guards - Kendrick Nunn Low PC2 and high PC3 -> efficient interior players - Vincent Poirier Low PC2 and low PC3 -> not that efficient interior players - Mathias Lessort

3. Application of MDS

a) Apply metric MDS using Euclidean distance on scaled numerical variables.

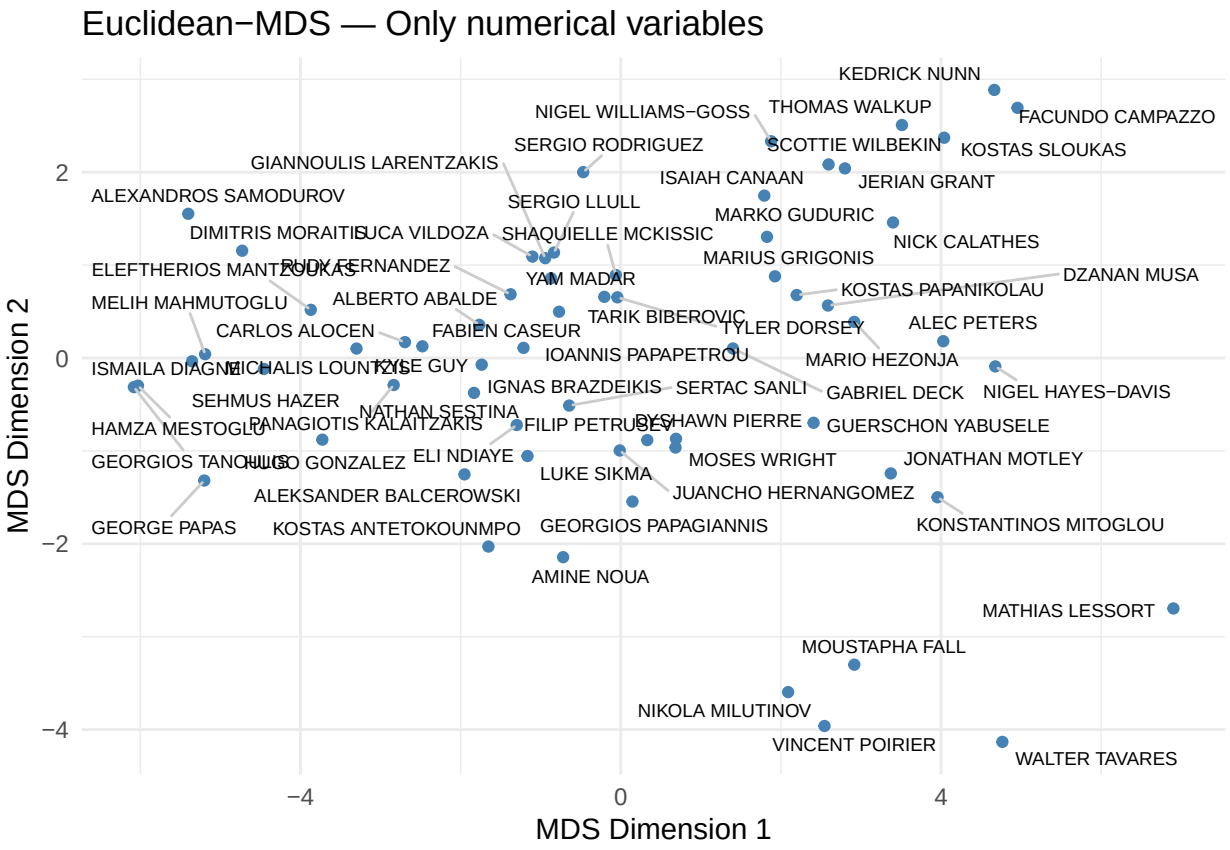
```
num_vars <- c("GP","GS","PTS","X2P.","X3P.","FT.","OR","DR","TR",
              "AST","STL","TO","BLK","BLKA","FC","FD","PIR","Min2")

num_scaled <- scale(df[, num_vars])

dist_mat   <- dist(num_scaled, method = "euclidean")

mds_out    <- cmdscale(dist_mat, k = 2, eig = TRUE)

mds_coords <- as.data.frame(mds_out$points)
colnames(mds_coords) <- c("Dim1", "Dim2")

cat("GOF of MDS:")
```

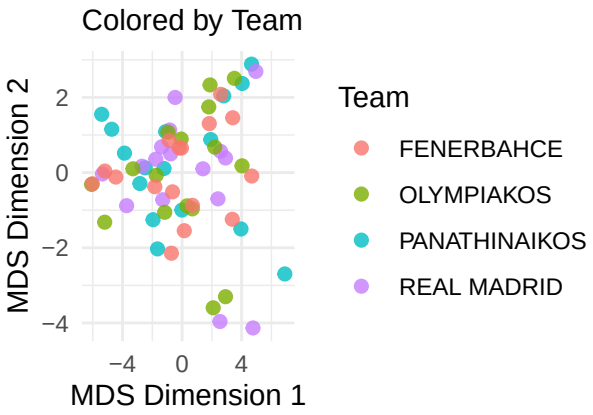
```
## GOF of MDS:
```

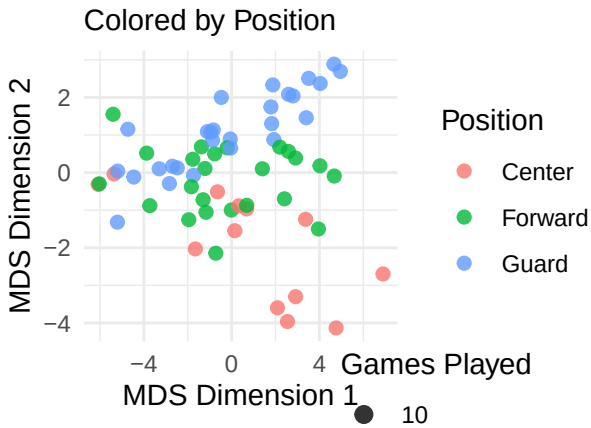
```
mds_out$GOF
```

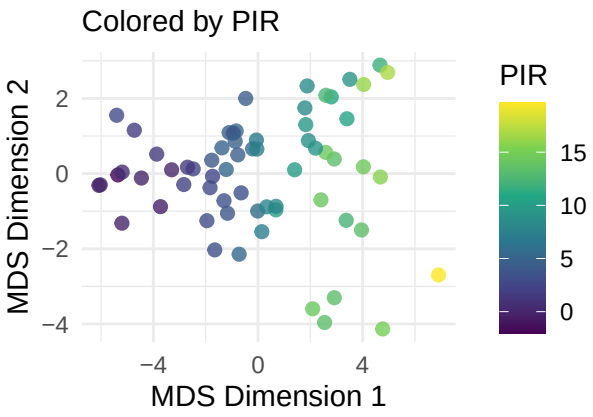
```
## [1] 0.6782979 0.6782979
```

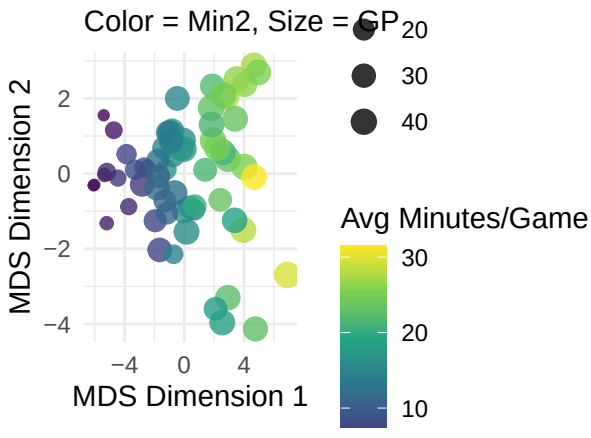
b) Plot the data using the points on the first two coordinates using players names as label.











c) Interpret the plot.

In the first, fully-labeled plot we can nonetheless pick out a few clusters of players with similar profiles:

- **Bottom-right**: Moustapha Fall, Nikola Milutinov, Vincent Poirier, Walter Tavares and Mathias Lessort form a tight group—big men with comparable rebounding/defensive stats.
- **Top-left**: Kedrick Nunn and Facundo Campazzo lie close together—likely high-usage guards with strong assist/steal numbers.
- **Top-right**: Alexandros Samodurov and Dimitris Moraitis cluster together—perhaps sharpshooters or role players with similar per-game outputs.
- Anywhere labels overlap heavily (e.g. Giannis Antetokounmpo and Amine Noua) also suggests very similar stat lines.

So even here you can "read" a few pockets of similar players, but the overcrowding makes it hard to be confident. That's why we switch to aesthetic mappings (color, size) in the subsequent plots—to reveal these groupings more clearly without the noise of overlapping names.

In the next plot, we used color to differentiate players by team. We wanted to see whether teammates would cluster together, but there's no obvious pattern: being on the same team doesn't make players significantly closer in the Euclidean-distance space. In other words, team affiliation isn't driving the distances—other stats are.

In the third plot, we colored by position, and here we do see clusters: players who share the same position tend to lie closer together. When we map PIR (in the fourth plot), high-PIR players cluster with other high-PIR players, and lower-PIR players cluster together and sit farther from the stars.

Finally, in the last plot we map average minutes per game to color and games played to size. We observe that the players who log the most minutes also tend to play the most games—and, looking back, those same players had higher PIR. That all aligns with the expectation that coaches give more playing time to their best performers.

d) Calculate gower distance including variable "POSITION" to the data matrix.

We want a single dissimilarity matrix that mixes all 18 numeric stats plus the categorical POSITION. Therefore, we'll use *cluster::daisy()* with *metric = "gower"*.

```
library(cluster)

cols_ex_d     <- c(num_vars, "POSITION")

gower_df   <- df[, cols_ex_d]
gower_dist <- daisy(gower_df, metric = "gower")  # 0 = identical, 1 = maximally different
```

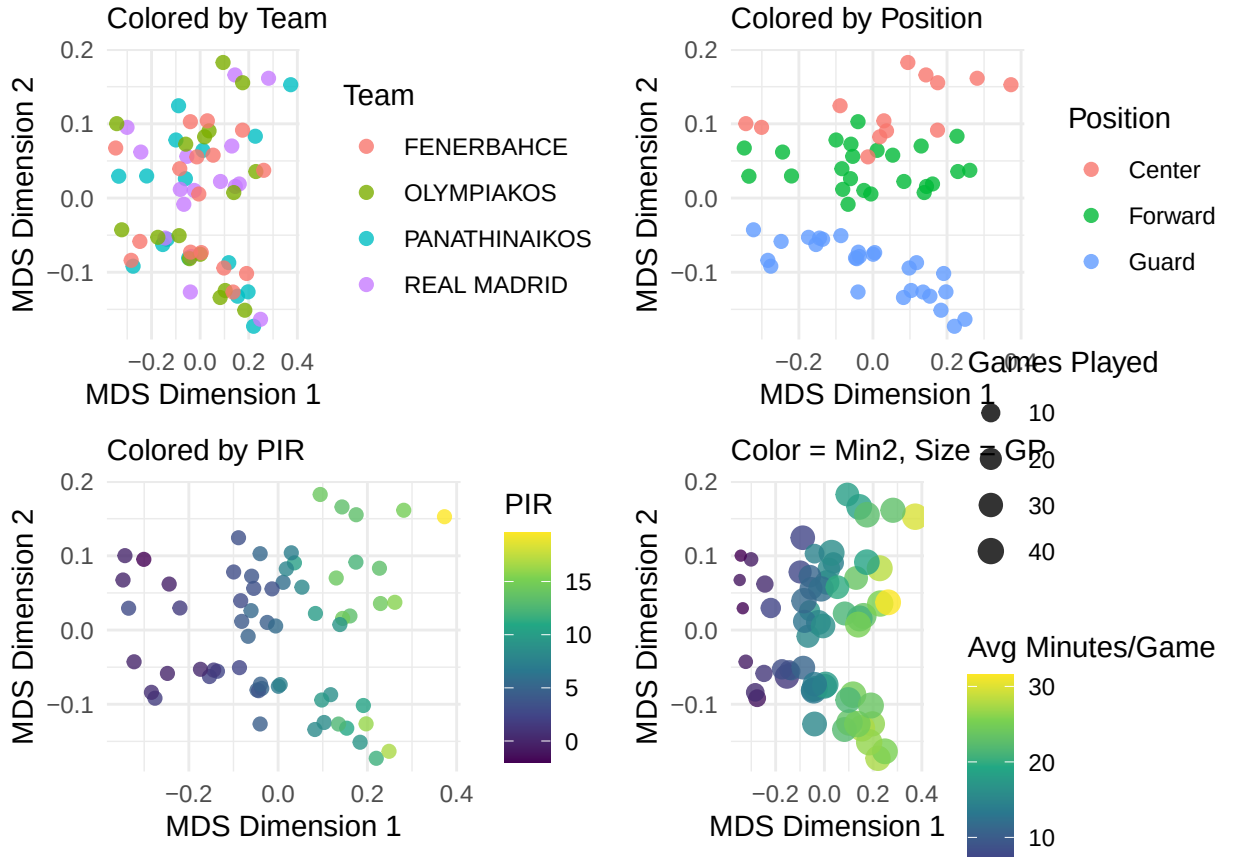
e) Apply metric MDS on gower distance matrix.

```
mds_g <- cmdscale(gower_dist, k = 2, eig = TRUE)

mds_g_df <- as.data.frame(mds_g$points)
colnames(mds_g_df) <- c("Dim1", "Dim2")

mds_g_df$PLAYER   <- df$PLAYER
mds_g_df$TEAM     <- df$TEAM
mds_g_df$POSITION <- df$POSITION
```

f) Plot individual plots on the first two coordinates

As the plot shows, we are able to see that this 2D representation is different from the one before, where the "POSITION" variable wasn't taken into account.

g) Use different categorical and numerical variables as labels so as to explain clusters that are constructed.



In the **Gower-MDS plot,** the two-dimensional configuration looks noticeably different from the Euclidean-MDS plot because **POSITION** is now built into the distance. Players separate not only by their box-score stats but also by their **role**, so you see, for example, **guards**, **forwards**, and **centers** occupying distinct regions in the plot -something you didn't get when POSITION was ignored.

Beyond that key difference, the other overlays tell the same story as before:

- **Team coloring** still shows no obvious team-based clusters: teammates remain scattered rather than grouped.
- **PIR gradient** again runs roughly left-to-right, with higher-PIR players on the right side of the plot.
- **Minutes per game** (color) and **games played** (size) likewise increase as you move to the right, reflecting that your most-utilized stars both play more minutes and appear more often.

In short, adding **POSITION** reorients the overall configuration—separating roles—but the main trend remains.

h) Which MDS do you think better group the individuals? Why?

The MDS that best groups the individuals is the Gower-based MDS, because it incorporates the categorical POSITION alongside the numeric stats. Therefore, forcing the algorithm to treat "different positions" (guard, forward, center) as genuinely different.

- **Euclidean MDS** (numeric only) blends all players into a single cloud, so it's hard to pick out distinct roles.

- **Gower MDS** treats "different positions" as maximally dissimilar, which pulls centers, forwards, and guards into separate regions. For example, all the big men—Tavares, Poirier, Milutinov—cluster tightly on one side, while the point guards—Campazzo, Nunn, Moraitis—cluster on another. Within each of those three role-based clusters, the best players (high PIR, lots of minutes) still stand out as slightly further away from the bench players.

In short, by including **POSITION** in the distance calculation, Gower MDS gives you three clear clouds—centers, forwards, guards—so you immediately see the role-based structure in the league.

As a final note, just because the Gower-MDS with **POSITION** gives clear role-based clusters doesn't mean it's the "ultimate" solution. You might improve the representation by including other categorical variables and recomputing the Gower distances. Ultimately, the "best" embedding is the one that most clearly exposes the particular patterns or groupings you care about in your analysis.