Homework1

Gabriel Guerra, Jonàs Salat i Biel Manté

2024-10-15

Contents

1. First do the exploratory data analysis.

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

```
data = data %>% select(-No)
```

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. (1p)

```
aux = strsplit(data$Min,split = ":")
df = data.frame(aux = NA)


b = lapply(1:length(aux),function(i){
  aux[[i]][1] <<-  as.numeric(aux[[i]][1]) * as.numeric(data[i,"GP"])
  aux[[i]][2] <<-  (as.numeric(aux[[i]][2]) * as.numeric(data[i,"GP"]))/60
  df[i,1] <<- as.numeric(aux[[i]][1]) + as.numeric(aux[[i]][2])

})

aux = df
```

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)

```
data = data %>% mutate("Min 2" = aux$aux/GP)
data = data %>% relocate("Min 2" ,.after = "Min")
data = data %>% select(-Min)
```

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

We shold change the variables team, player and position to factor

```
str(data)
```

```
## '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 ...
##  $ Min 2   : num  5.93 14.93 10.63 2.42 7.6 ...
##  $ 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 ...
```

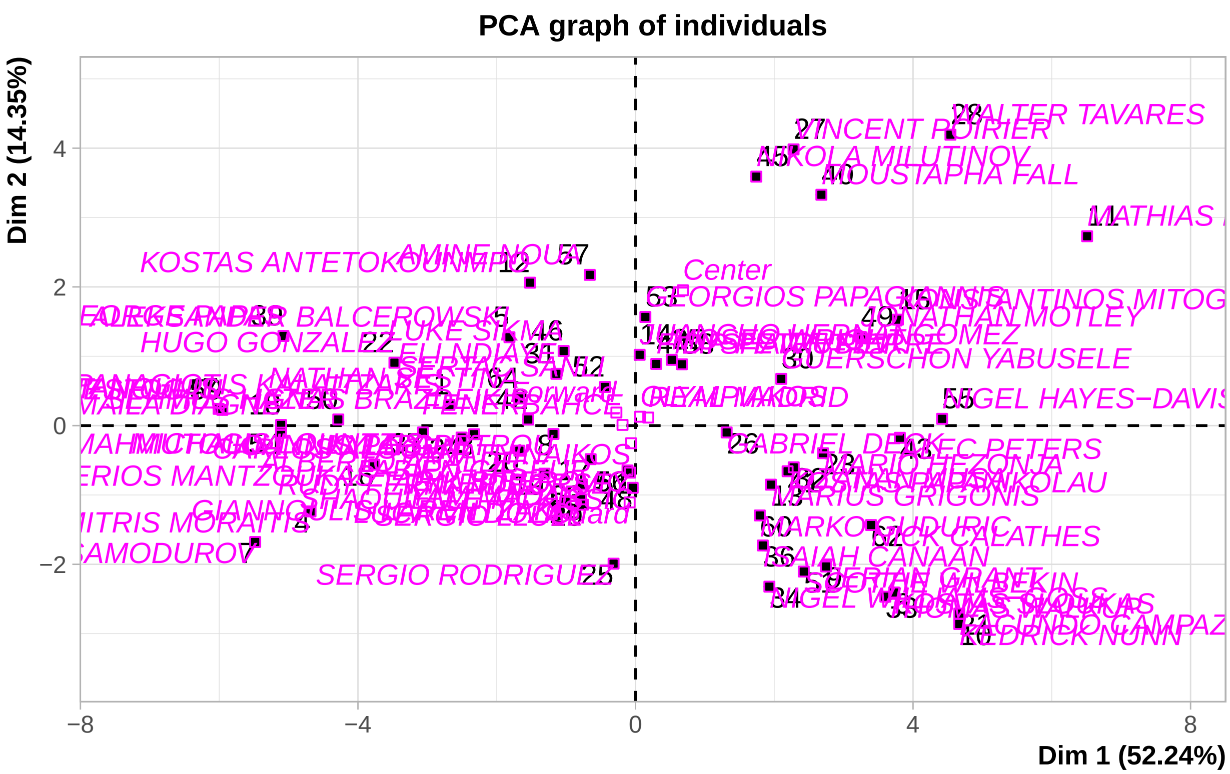
```
##  $ TO        : num  0.2 1 1 0.3 0.3 2.4 0 0.4 1.1 3.1 ...
##  $ BLK       : num  0 0 0.1 0 0.4 0 0 0.2 0.1 0.1 ...
##  $ BLKA      : num  0 0.2 0 0.1 0.1 0.4 0 0.2 0.1 0.8 ...
##  $ FC        : num  0.8 0.8 1.2 0.1 1.5 1.8 0 1.4 2.3 2.2 ...
##  $ FD        : num  0.4 0.6 0.6 0 1.2 3 0 0.9 2.1 2.7 ...
##  $ PIR       : num  2.1 4.6 2.4 1.7 3.1 16.1 3 5.4 10.9 11.7 ...
```

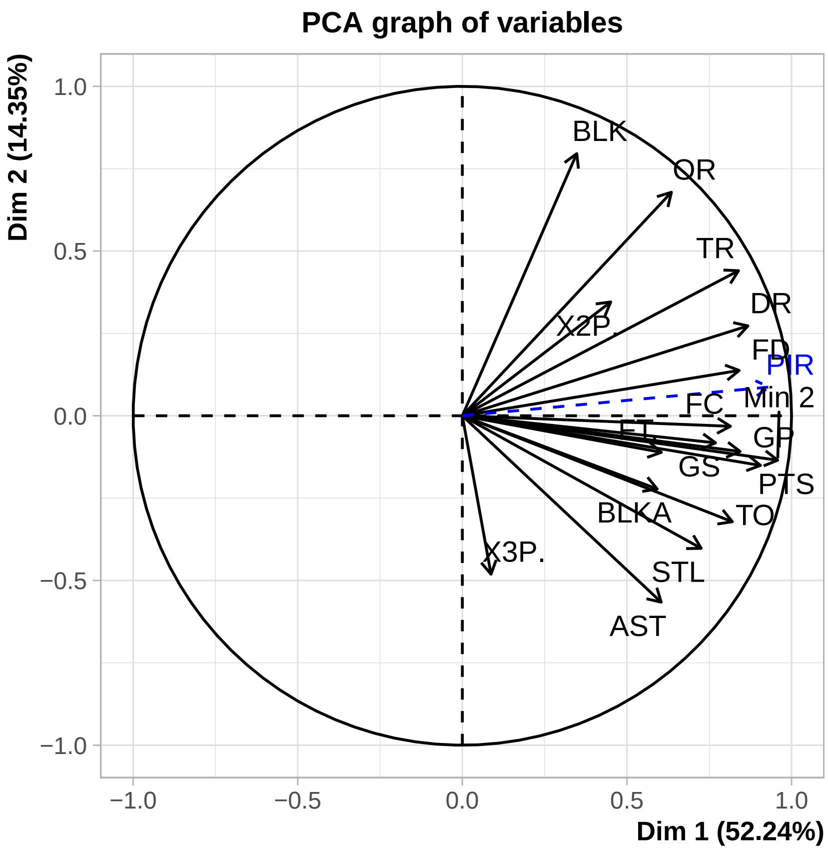
```
data = data %>% mutate_if(is.character,factor)
```

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)

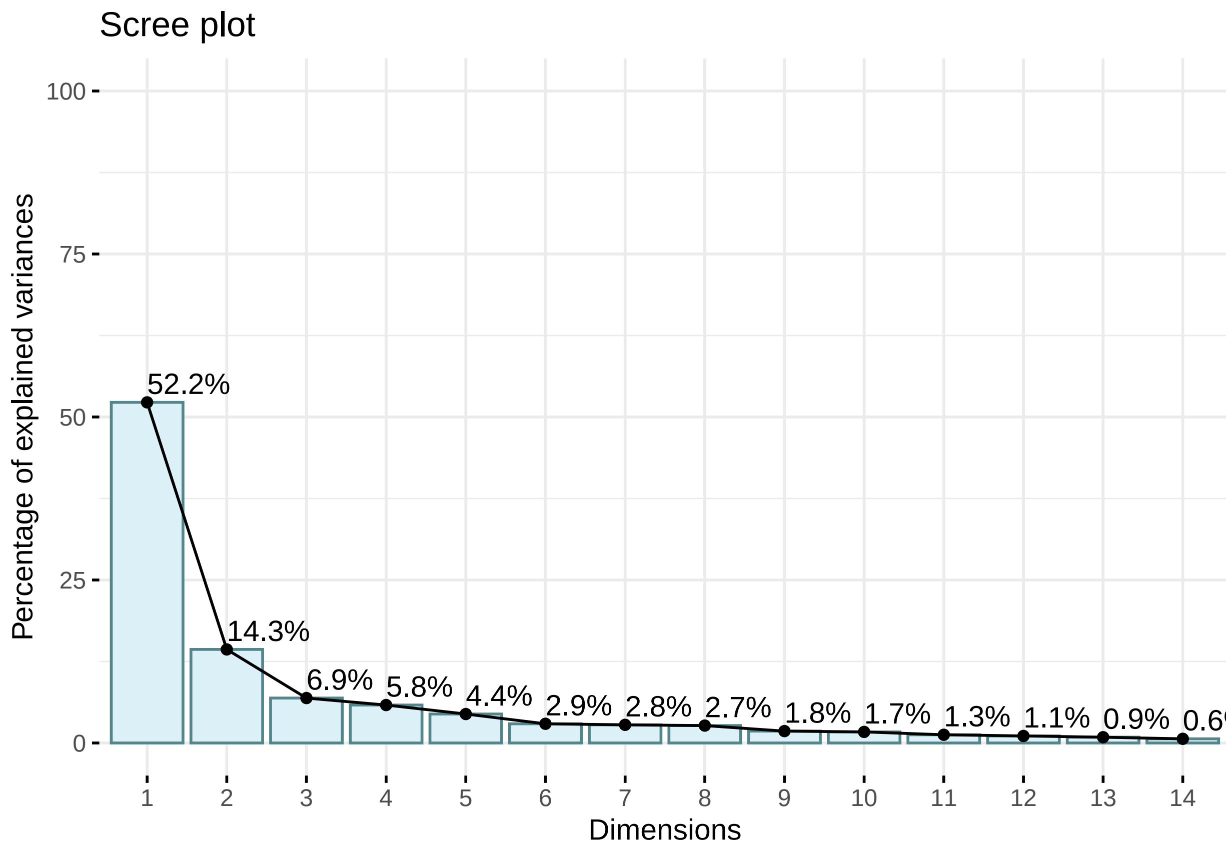
```
pca = PCA(data, quanti.sup = which(colnames(data) == "PIR"),quali.sup = which(sapply(data, is.factor)))
```



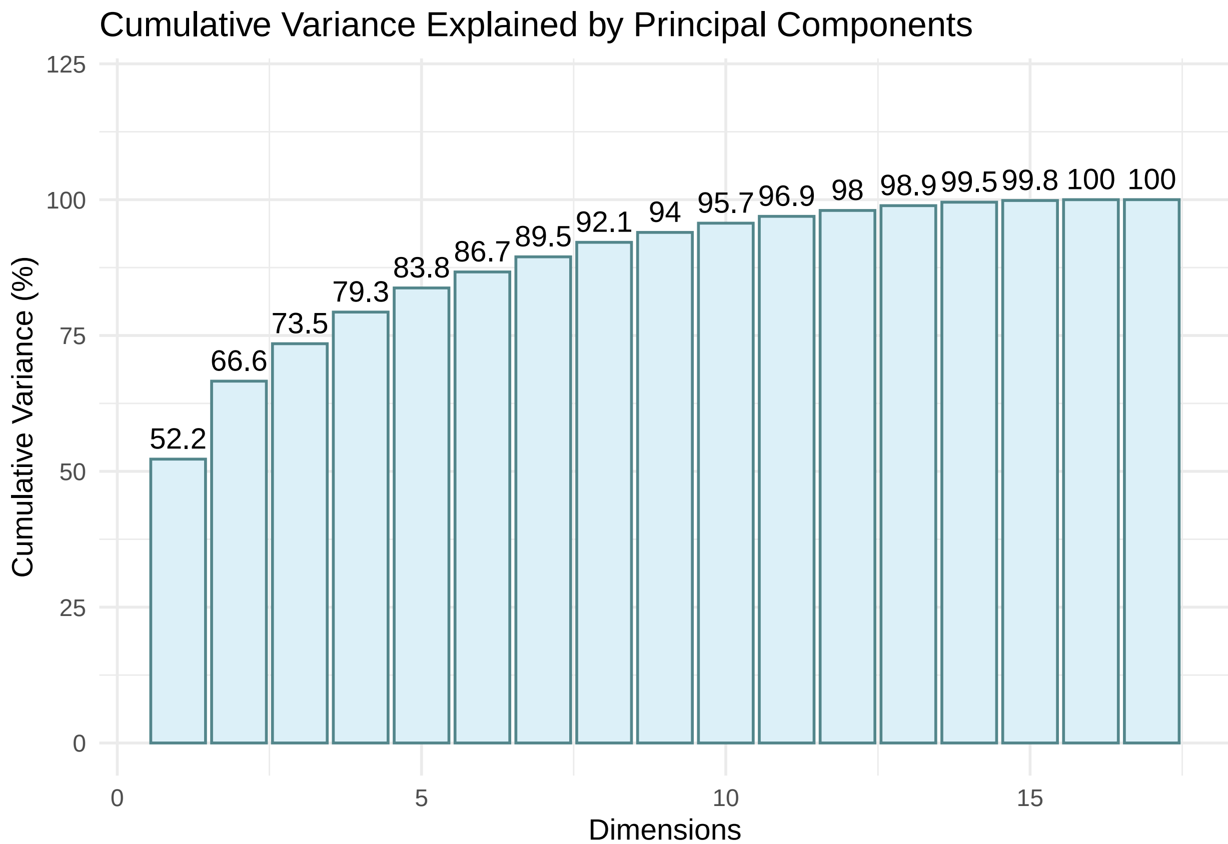


b) How many components should be extracted? Decide on the number of components considering eigenvalues. (3p)

```
fviz_eig(pca, addlabels = TRUE, ylim = c(0, 100),
         barcolor = "#53868B", barfill = "#DCF0F8",
         ncp = 14, geom = c("bar", "line"))
```




```r
eig_vals = get_eig(pca)[,2]
cumulative_var = cumsum(eig_vals)

df = data.frame(Dimension = 1:length(cumulative_var),
                 CumulativeVariance = cumulative_var)

# Plot cumulative variance
ggplot(df, aes(x = Dimension, y = CumulativeVariance)) +
  geom_bar(stat = "identity", fill = "#DCF0F8", color = "#53868B") +
  geom_text(aes(label = round(CumulativeVariance, 1)), vjust = -0.5) +
  ylim(0, 120) +
  labs(title = "Cumulative Variance Explained by Principal Components",
       x = "Dimensions", y = "Cumulative Variance (%)") + theme_minimal()
```



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

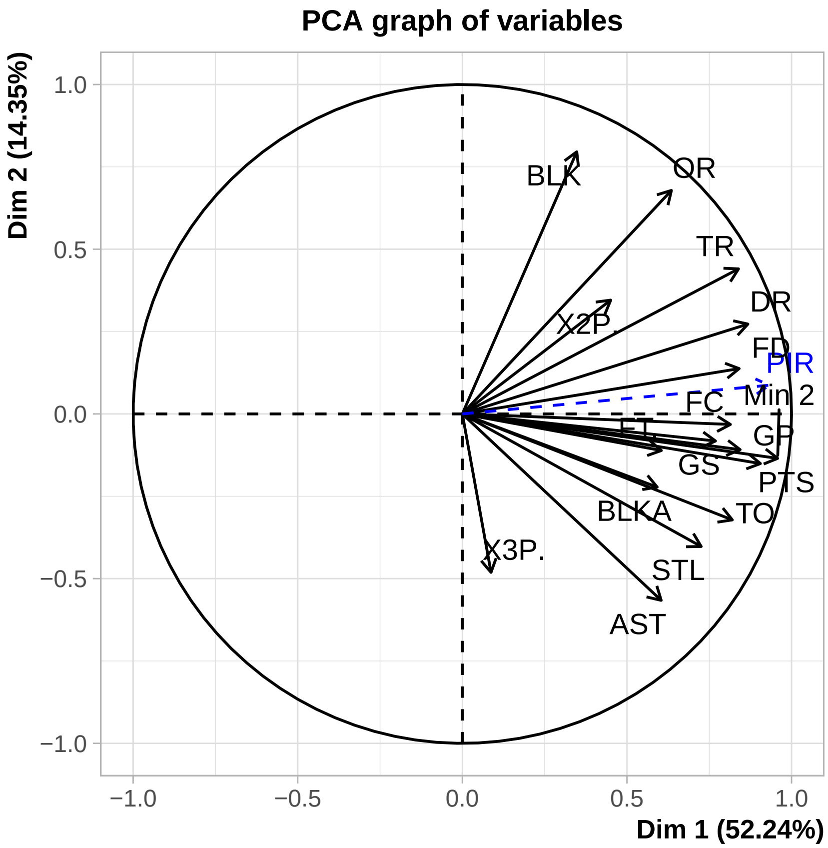
```
pca$var$coord[,1:5]
```

```
##             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
## Min 2  0.95783899 -0.13495801  0.02053780  0.06638688 -0.056566493
## 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
```

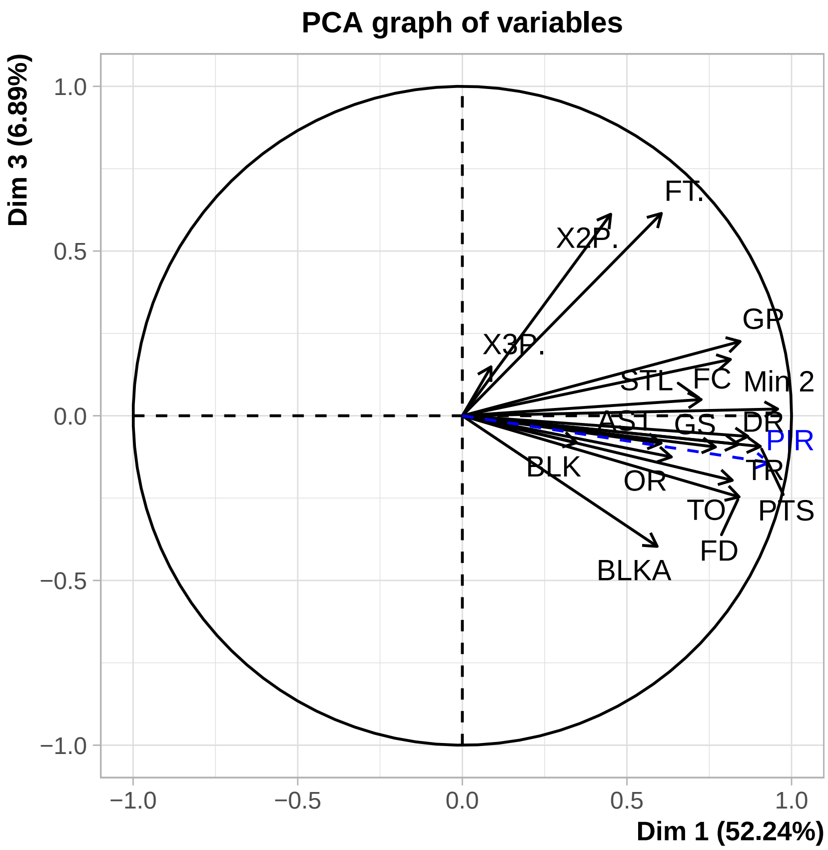
d) Use plot.PCA() function to show correlations between variables and the extracted dimensions. (For the variables you should use the argument choix = "var"). Plot all the extracted dimensions changing argument "axes".(3p)

```
c  = 1:5
c = t(combn(c, m =2))

lapply(1:nrow(c),function(i){
  plot.PCA(pca,choix = "var",axes = c[i,])
})
```

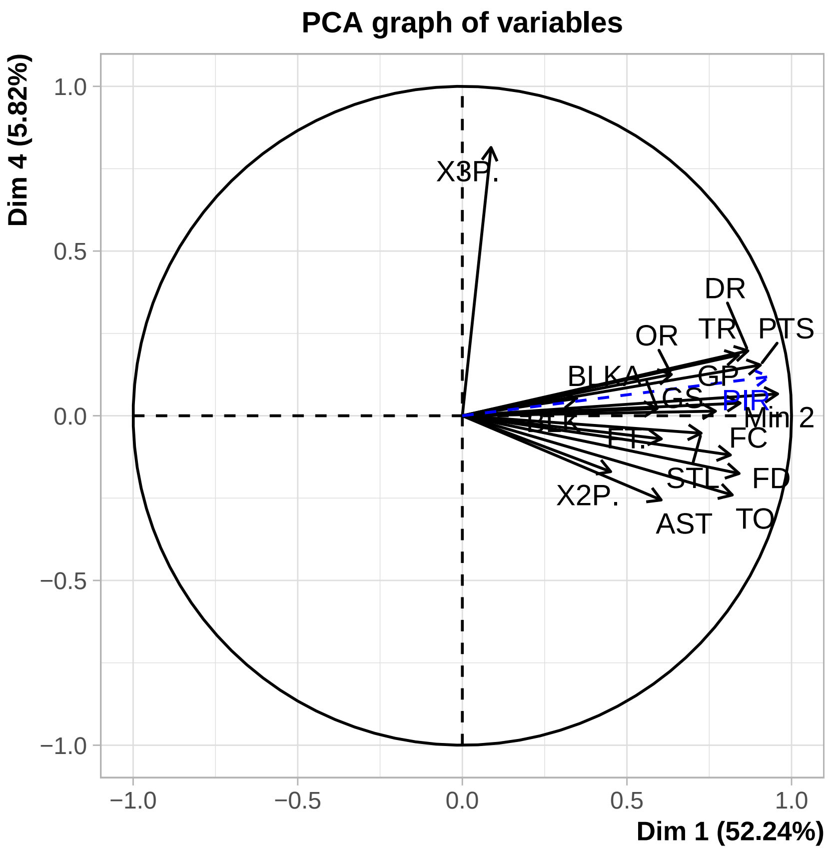
```
## [[1]]
```



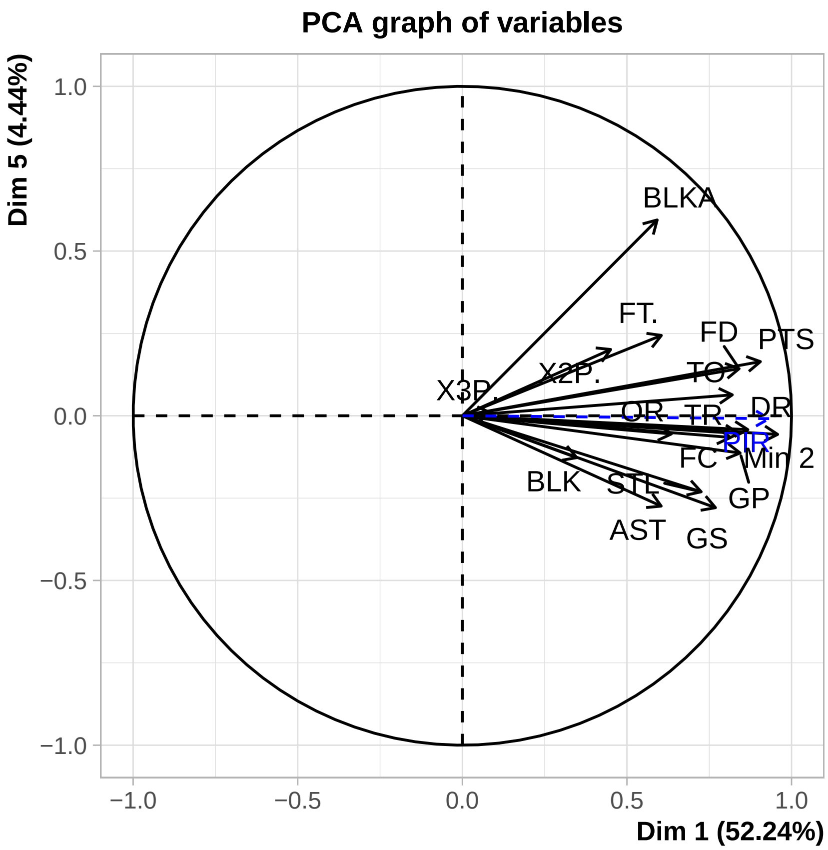

```
##
## [[2]]
```


PCA graph of variables
Dim 3 (6.89%)
Dim 1 (52.24%)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
FT.
X2P.
GP
X3P.
STL
FC
Min 2
AST
GS
DR
PIR
BLK
OR
TR
TO
PTS
FD
BLKA

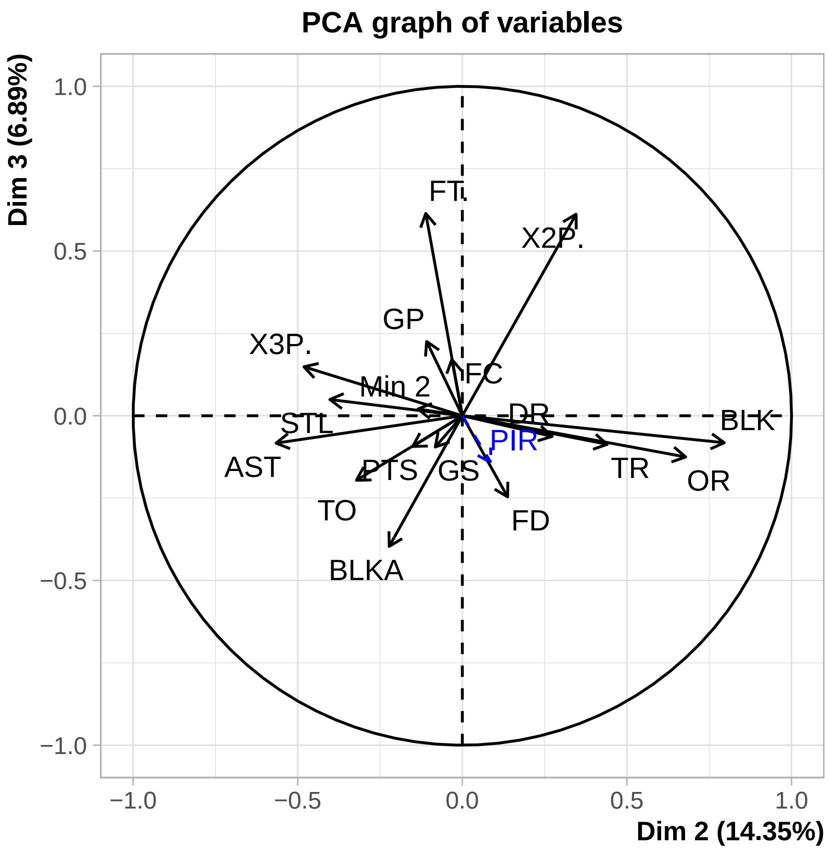
```
##
## [[3]]
```


PCA graph of variables
Dim 4 (5.82%)
Dim 1 (52.24%)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
X3P.
DR
TR
PTS
OR
BLKA
GP
GS
PIR
Min 2
BLK
FT.
FC
STL
FD
X2P.
AST
TO

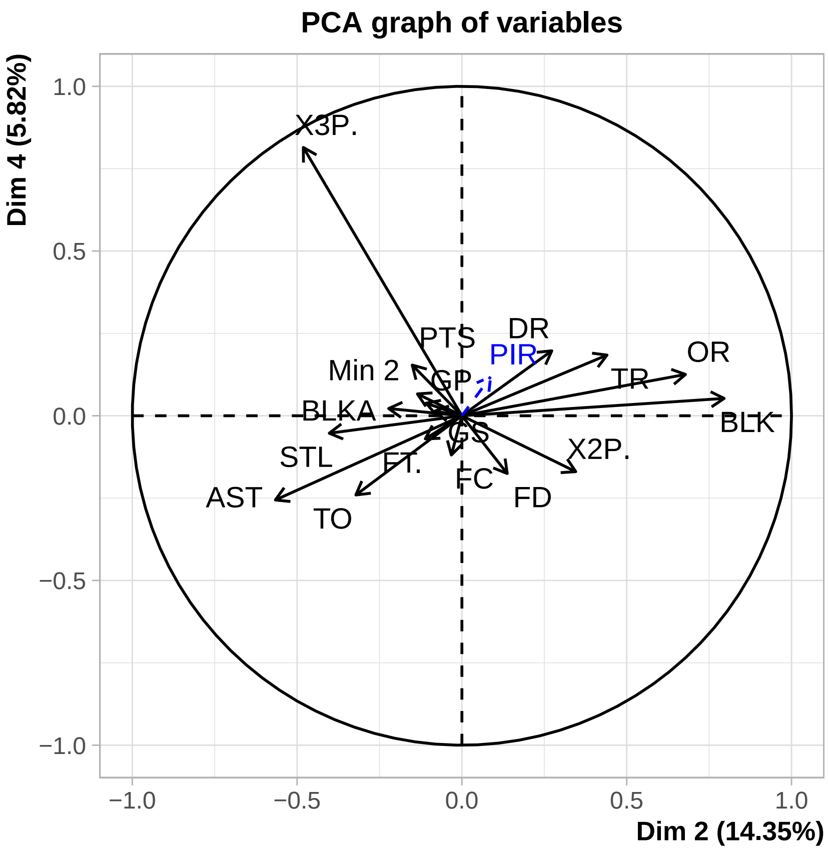
```
##
## [[4]]
```


PCA graph of variables
Dim 5 (4.44%)
Dim 1 (52.24%)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
BLKA
FT.
FD
PTS
X2P.
TO
X3P.
OR
TR
DR
PIR
FC
Min 2
BLK
STL
GP
AST
GS

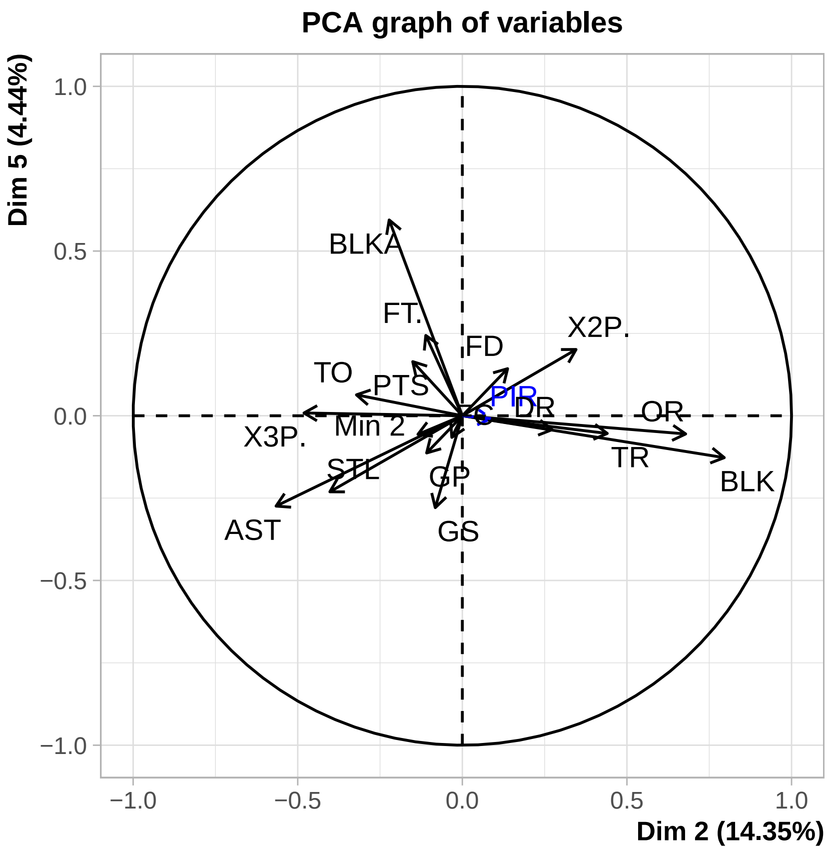
```
##
## [[5]]
```


PCA graph of variables
Dim 3 (6.89%)
Dim 2 (14.35%)
1.0
0.5
0.0
−0.5
−1.0
FT.
X2P.
GP
X3P.
FC
Min 2
DR
BLK
STL
PIR
AST
PTS
GS
TR
OR
TO
FD
BLKA

```
##
## [[6]]
```

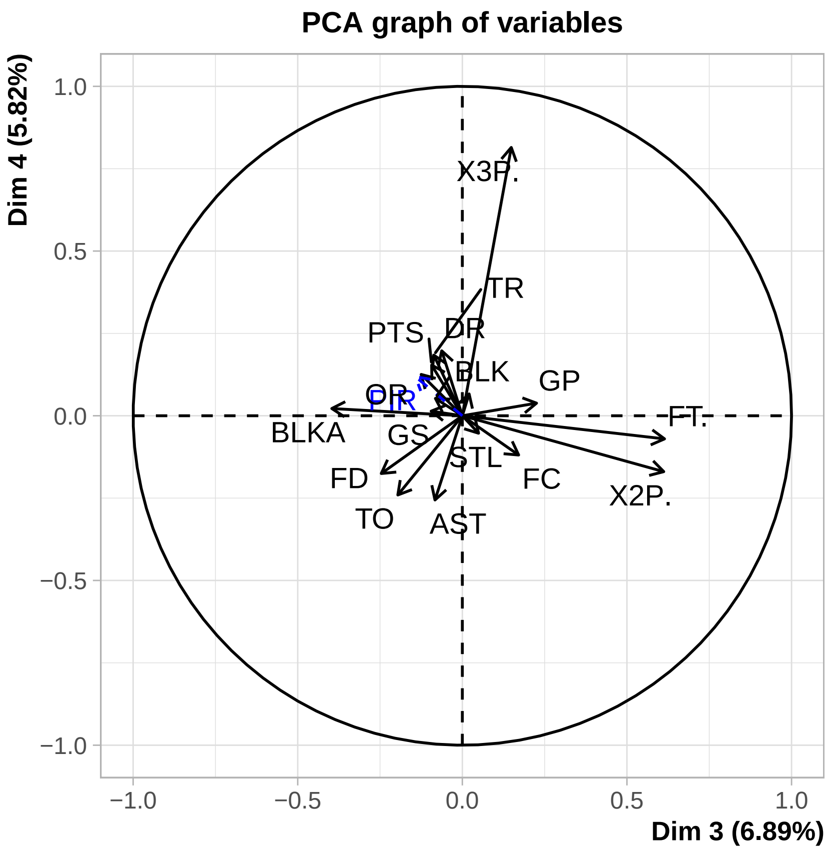

PCA graph of variables
Dim 4 (5.82%)
Dim 2 (14.35%)
X3P.
PTS
DR
PIR
OR
Min 2
GP
TR
BLKA
BLK
GS
X2P.
STL
FT.
FC
AST
FD
TO
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

```
##
## [[7]]
```

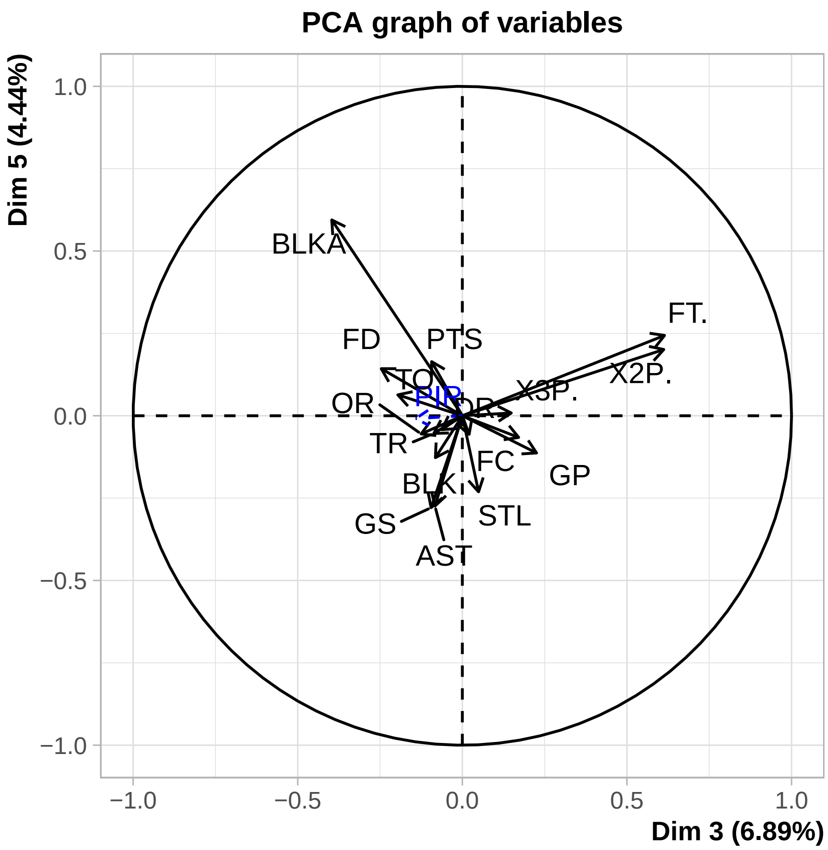


PCA graph of variables
Dim 5 (4.44%)
Dim 2 (14.35%)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
BLKA
FT.
FD
X2P.
TO
PTS
PIR
DR
OR
Min 2
X3P.
TR
STL
GP
BLK
AST
GS

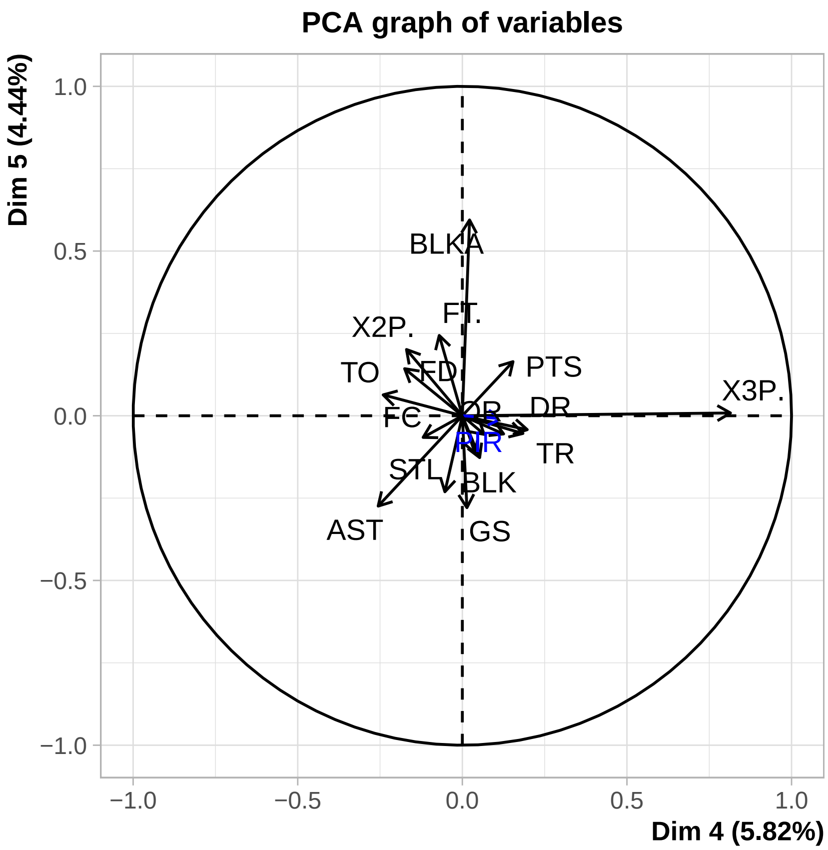

```
##
## [[8]]
```


PCA graph of variables
Dim 4 (5.82%)
Dim 3 (6.89%)
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
X3P.
TR
PTS
DR
BLK
GP
OR
PIR
FT.
BLKA
GS
STL
FD
FC
X2P.
TO
AST

```
##
## [[9]]
```


PCA graph of variables
Dim 5 (4.44%)
Dim 3 (6.89%)
BLKA
FT.
FD
PTS
X2P.
TO
X3P.
OR
PIP
TR
FC
GP
BLK
STL
GS
AST
1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0

```
##
## [[10]]
```



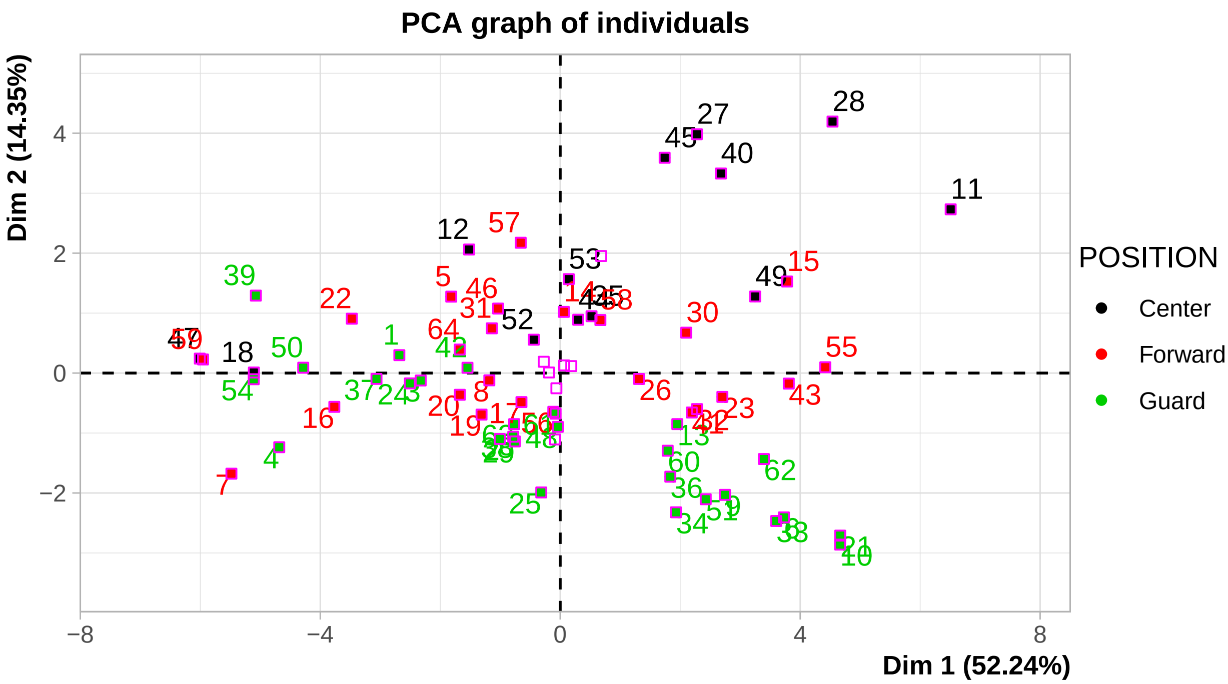
e) Interpret variable plots. How can each dimension be named? (5p)

f) Show individual pilots for the extracted dimensions changing argumennt choix="ind" in plot.PCA() function. (2p)

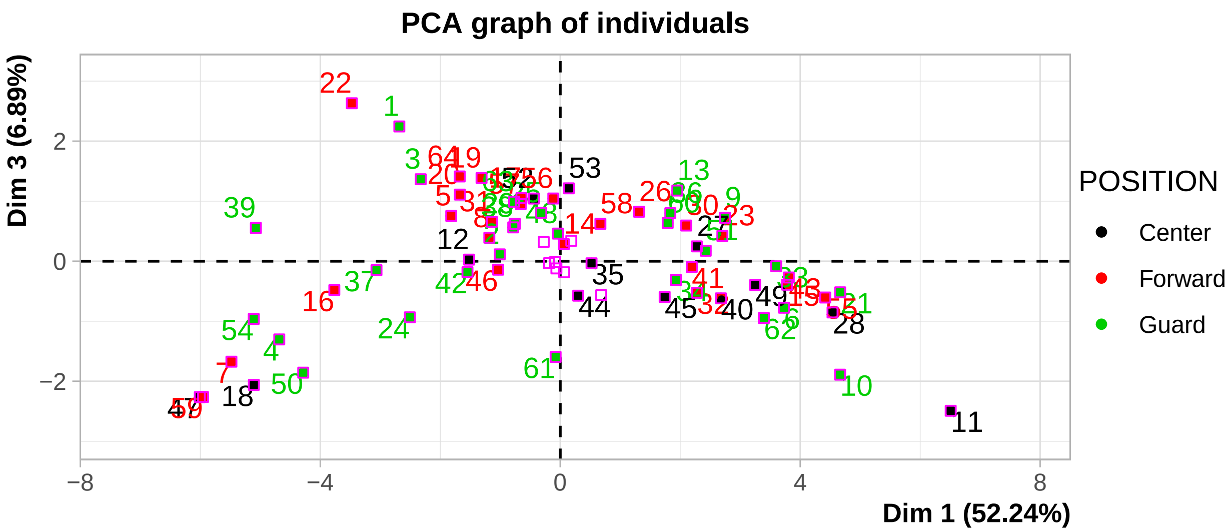
By Position

```
lapply(1:nrow(c),function(i){
  plot.PCA(pca,choix = "ind",axes = c[i,],habillage = 3,label = "ind", col.ind = "blue")
})
```

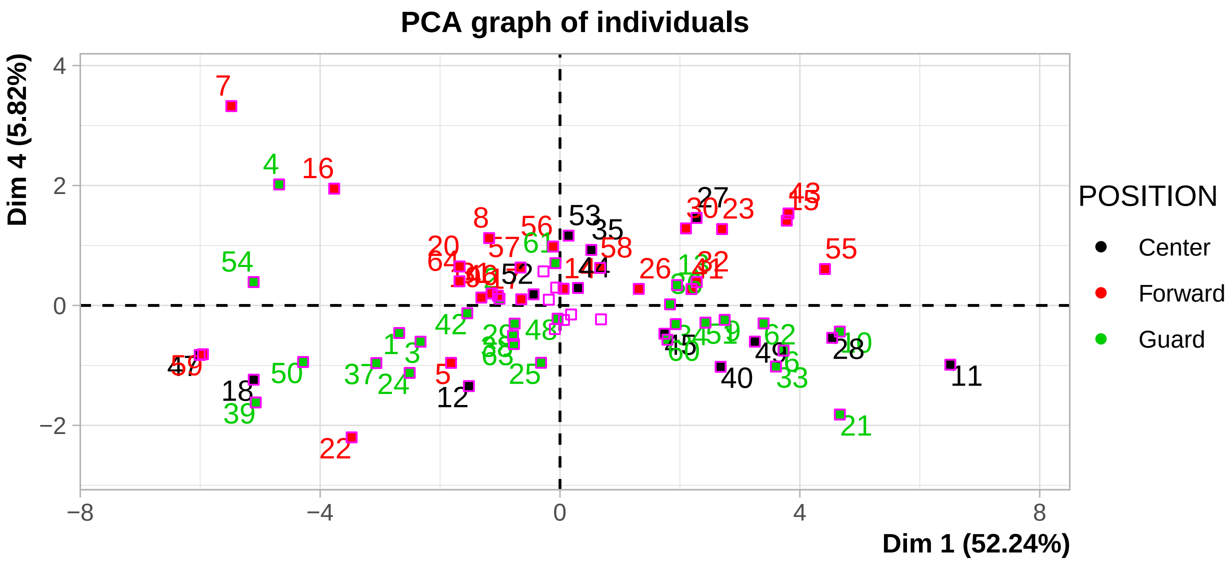
```
## [[1]]
```


PCA graph of individuals
Dim 2 (14.35%)
Dim 1 (52.24%)
POSITION
Center
Forward
Guard

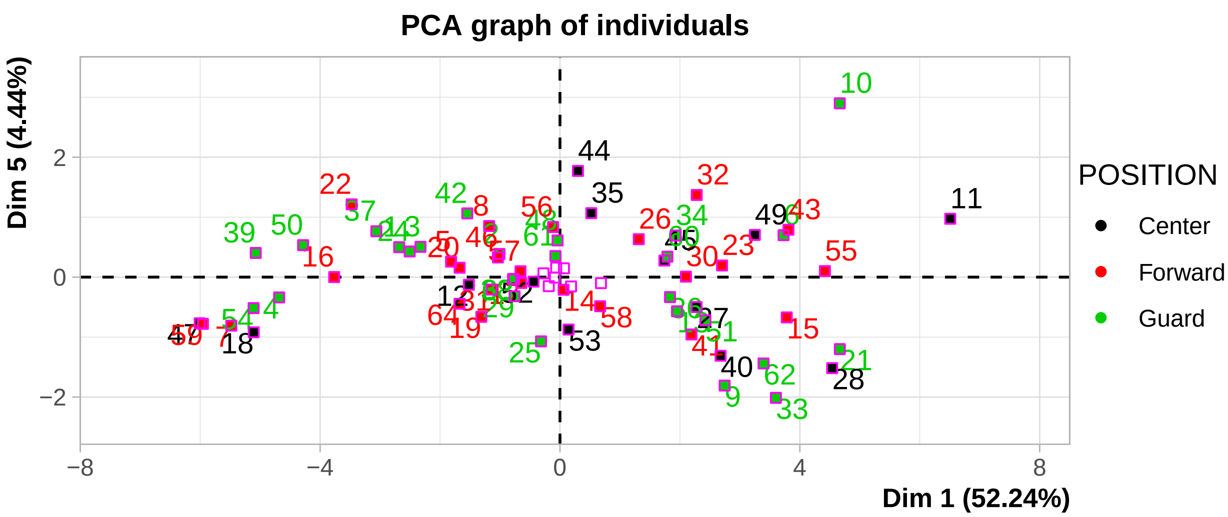
```
##
## [[2]]
```


PCA graph of individuals
Dim 3 (6.89%)
Dim 1 (52.24%)
POSITION
Center
Forward
Guard

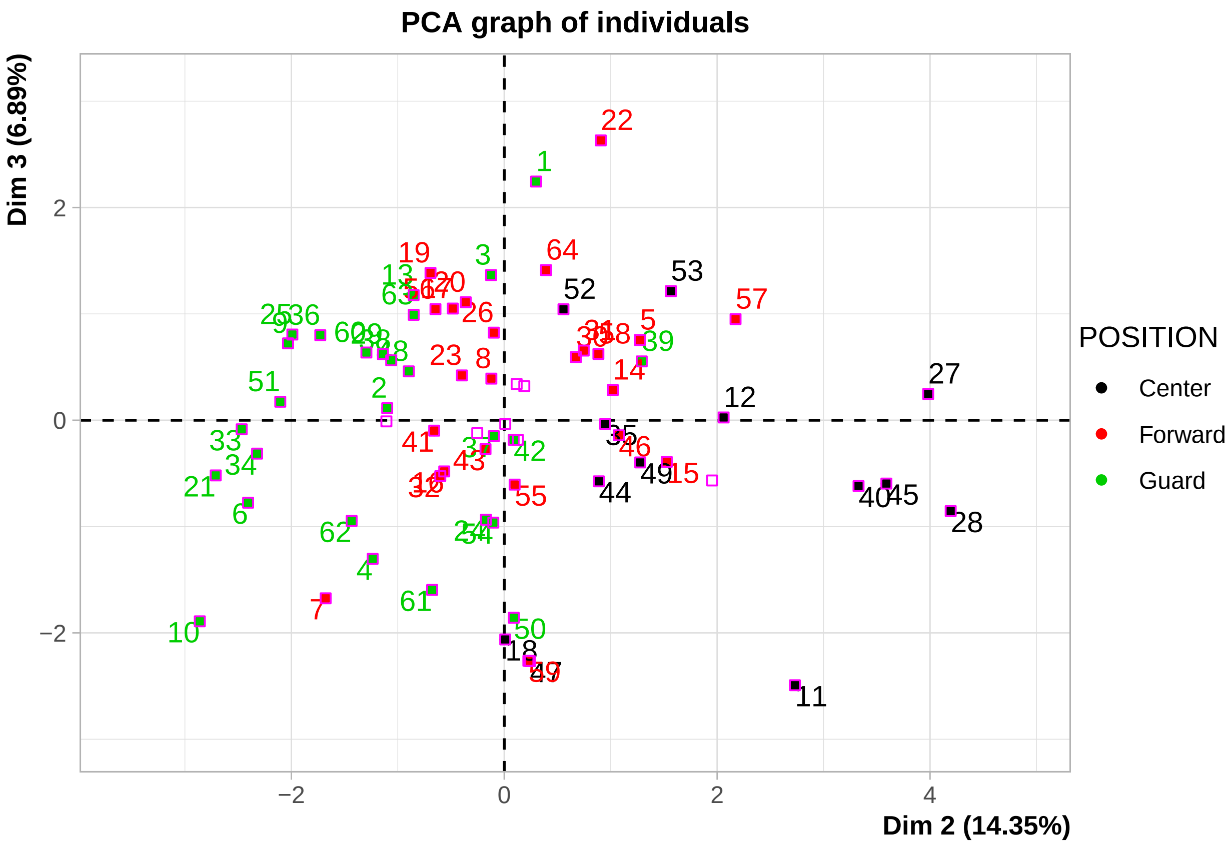
```
##
## [[3]]
```


PCA graph of individuals
Dim 4 (5.82%)
Dim 1 (52.24%)
POSITION
Center
Forward
Guard

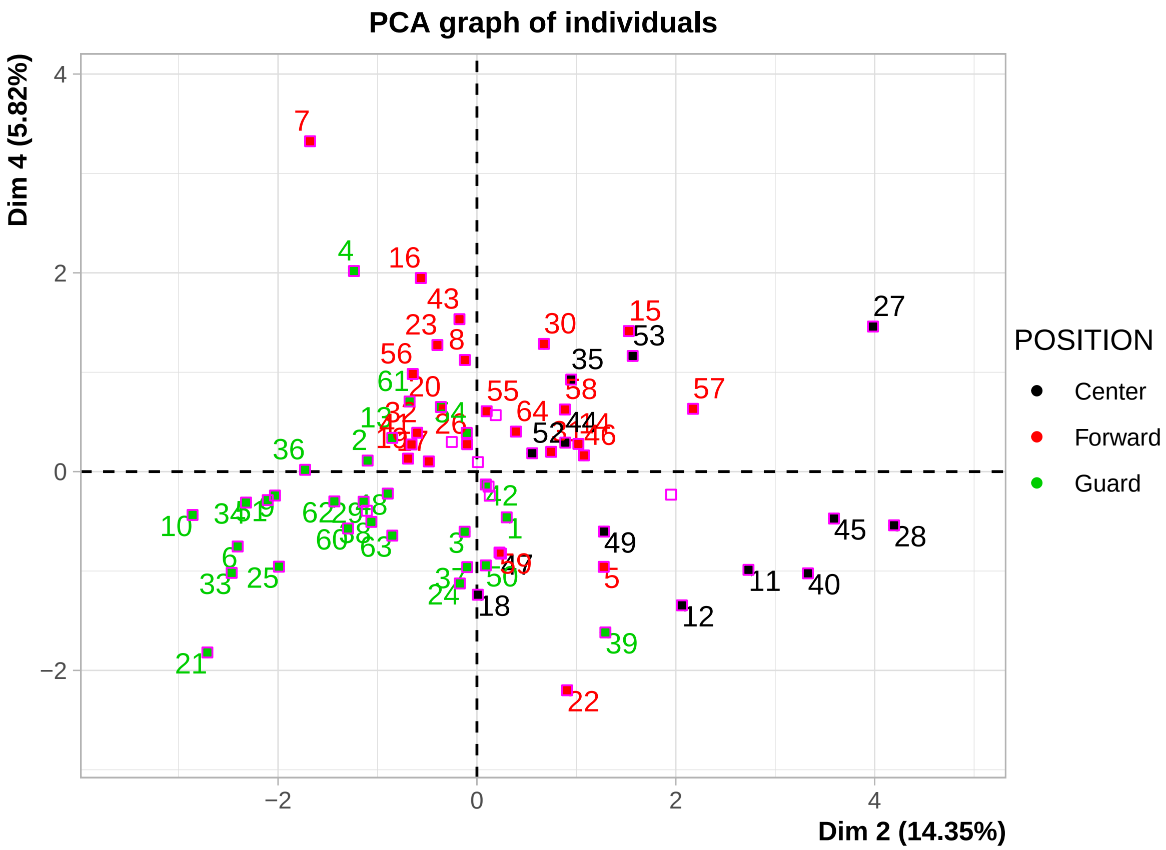
```
##
## [[4]]
```


PCA graph of individuals
Dim 5 (4.44%)
Dim 1 (52.24%)
POSITION
Center
Forward
Guard

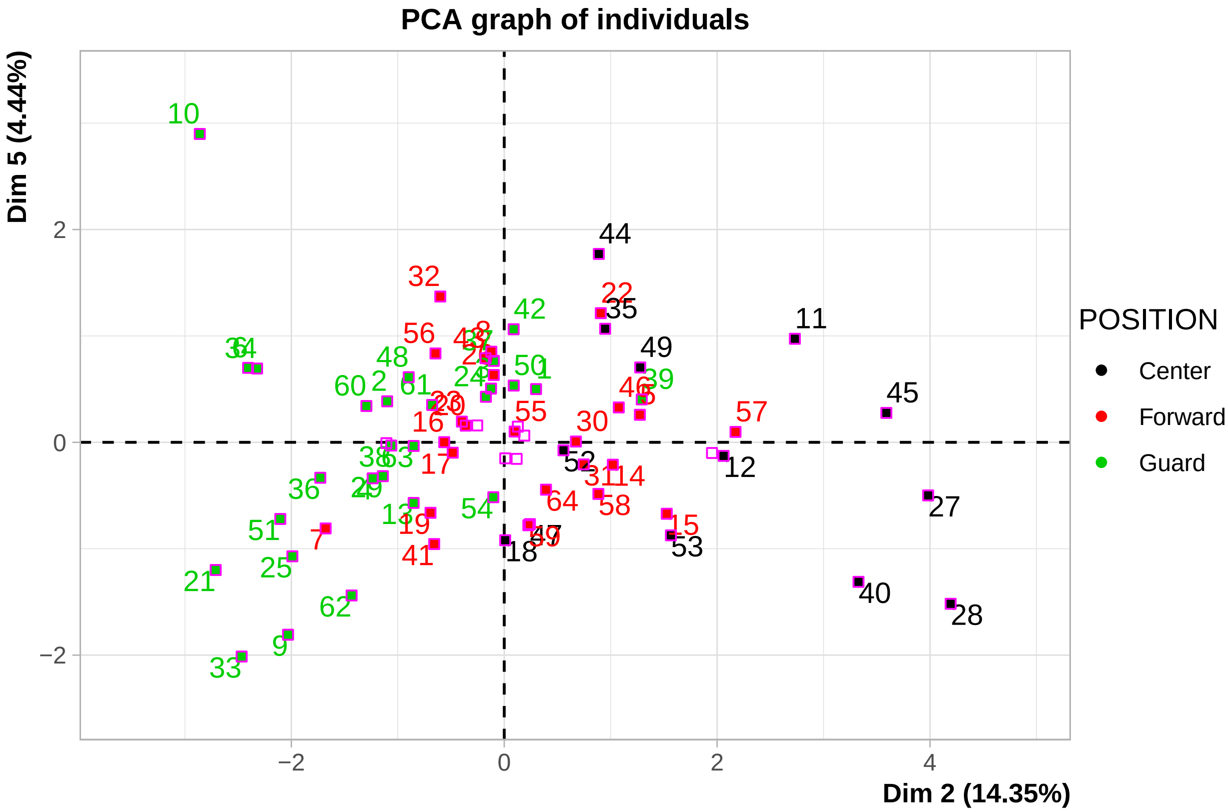
```
##
## [[5]]
```



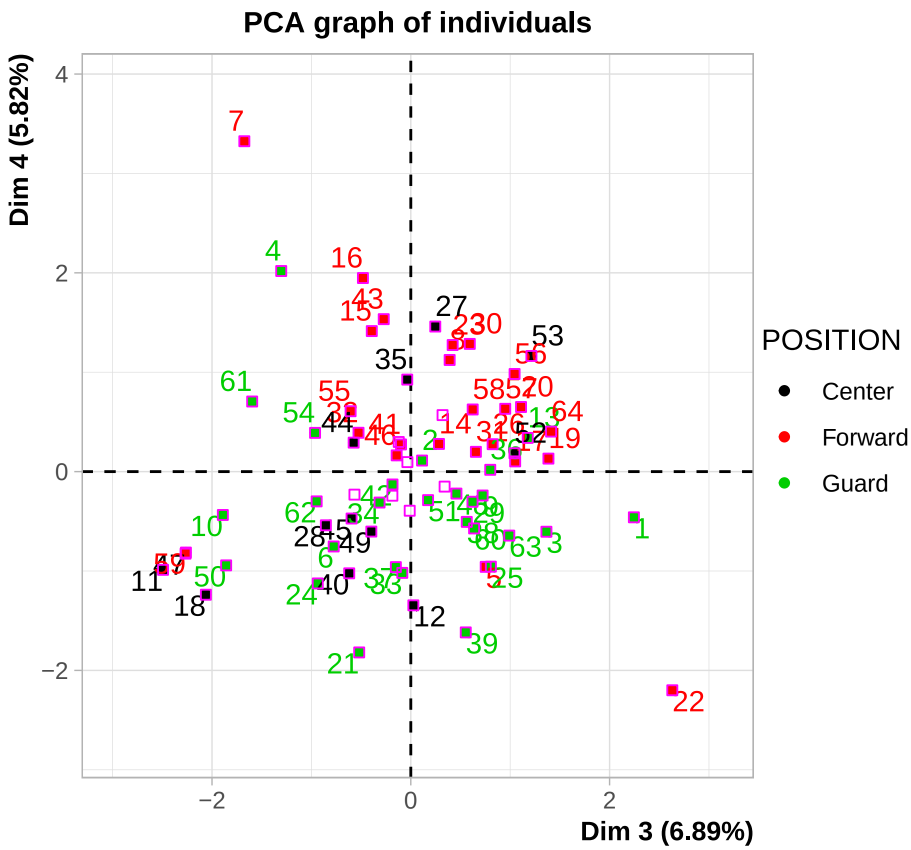
```
##
## [[6]]
```


PCA graph of individuals
Dim 4 (5.82%)
Dim 2 (14.35%)
POSITION
Center
Forward
Guard

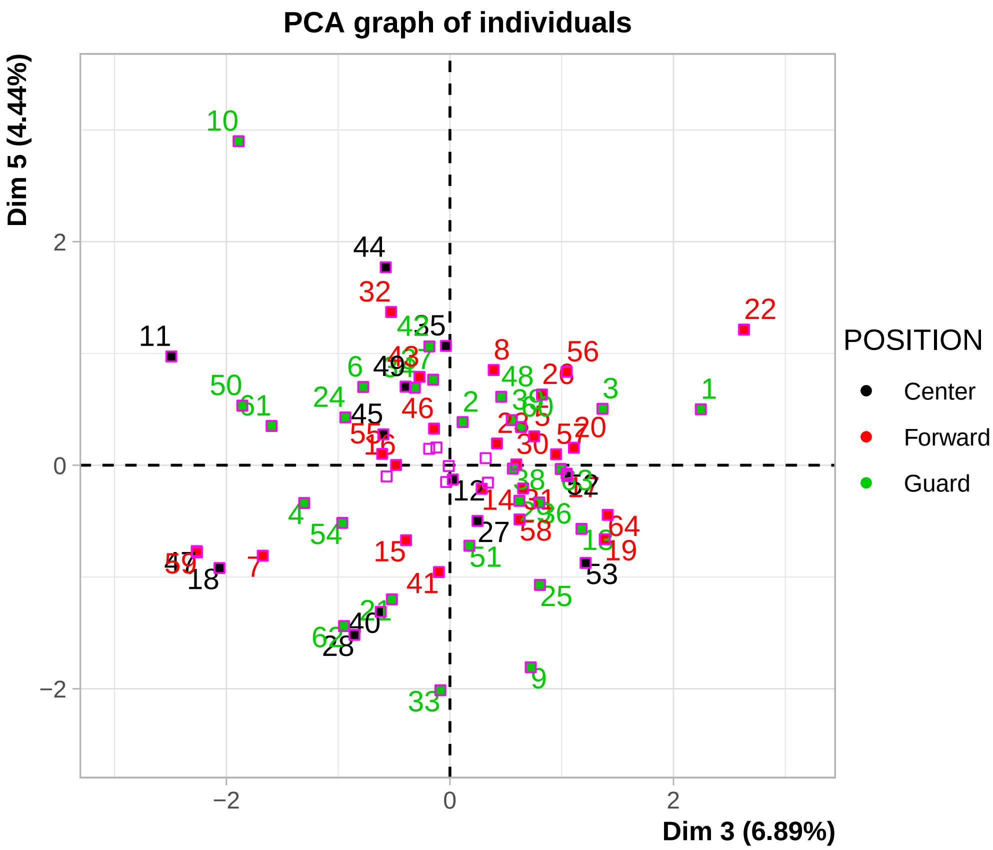
```
##
## [[7]]
```


PCA graph of individuals
Dim 5 (4.44%)
Dim 2 (14.35%)
POSITION
Center
Forward
Guard

```
##
## [[8]]
```

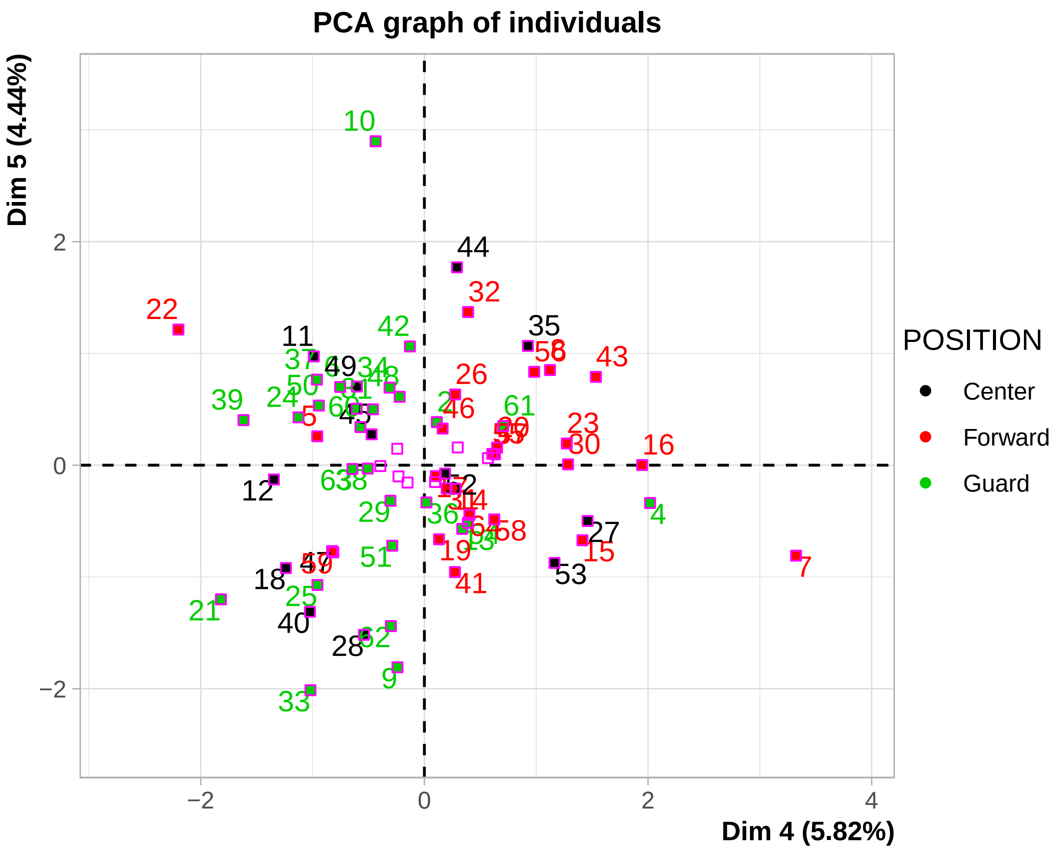

PCA graph of individuals
Dim 4 (5.82%)
Dim 3 (6.89%)
POSITION
Center
Forward
Guard

```
##
## [[9]]
```



PCA graph of individuals
Dim 5 (4.44%)
Dim 3 (6.89%)
POSITION
Center
Forward
Guard

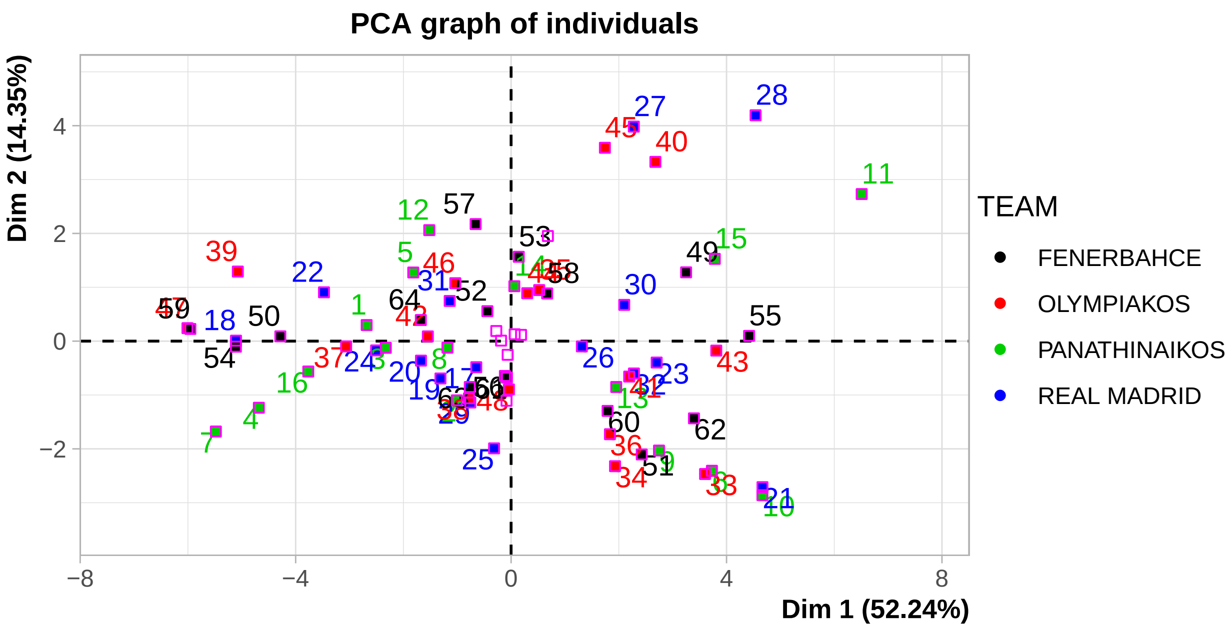

```
##
## [[10]]
```



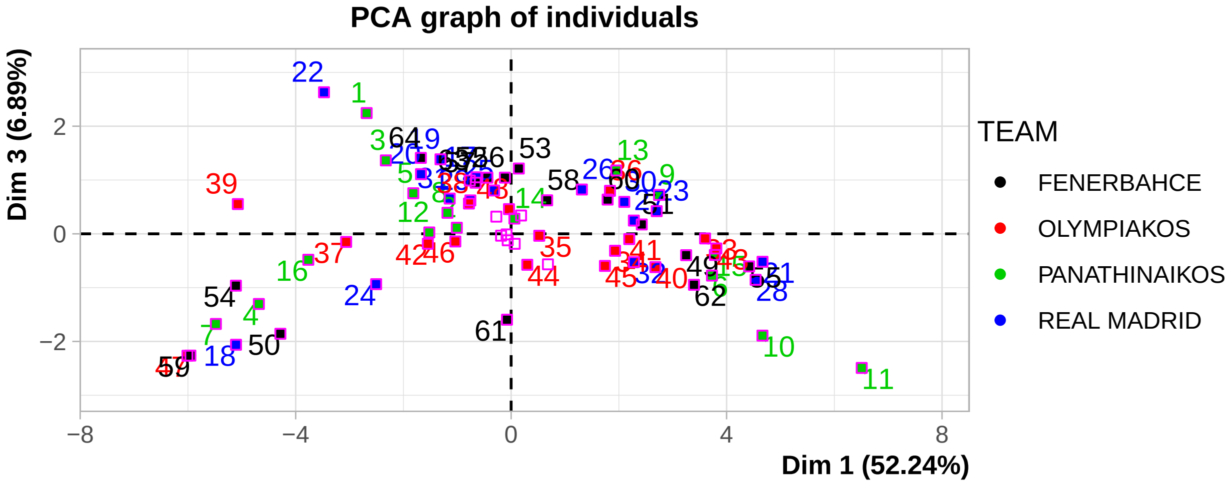
By Team

```
lapply(1:nrow(c),function(i){
  plot.PCA(pca,choix = "ind",axes = c[i,],habillage = 1,label = "ind", col.ind = "blue")
})
```

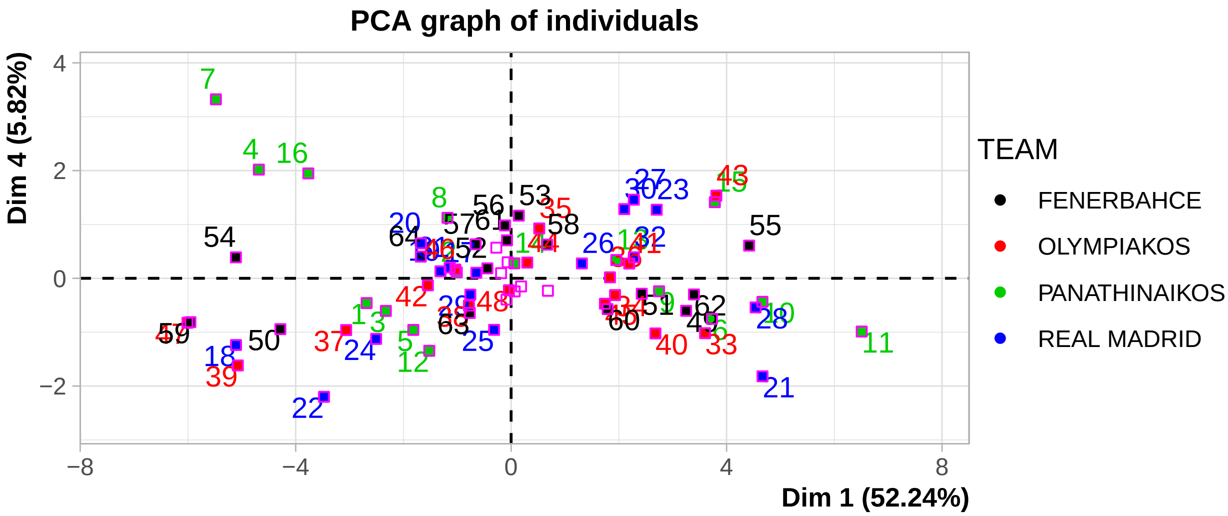
```
## [[1]]
```



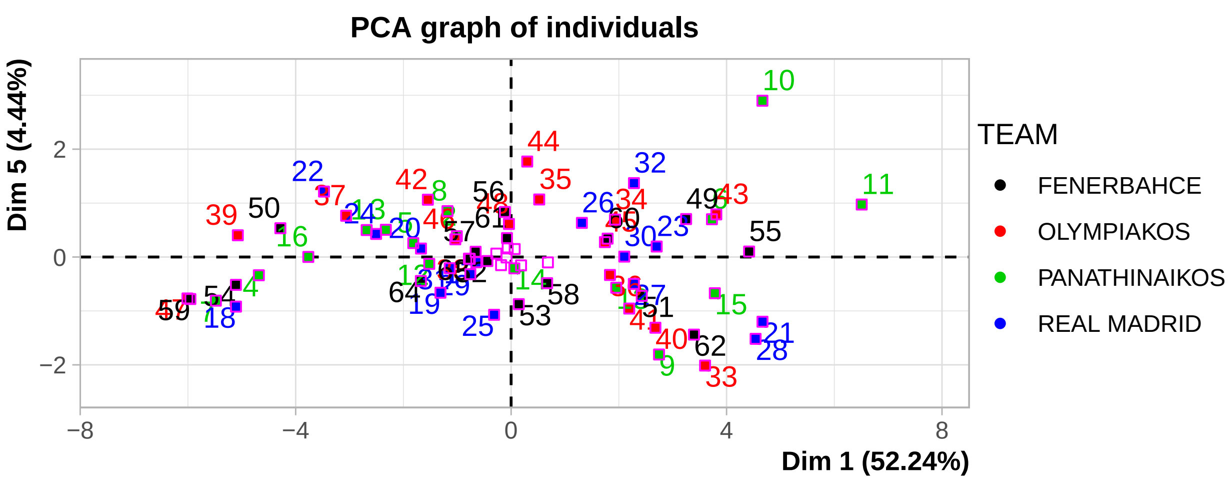
```
##
## [[2]]
```


PCA graph of individuals
Dim 3 (6.89%)
Dim 1 (52.24%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

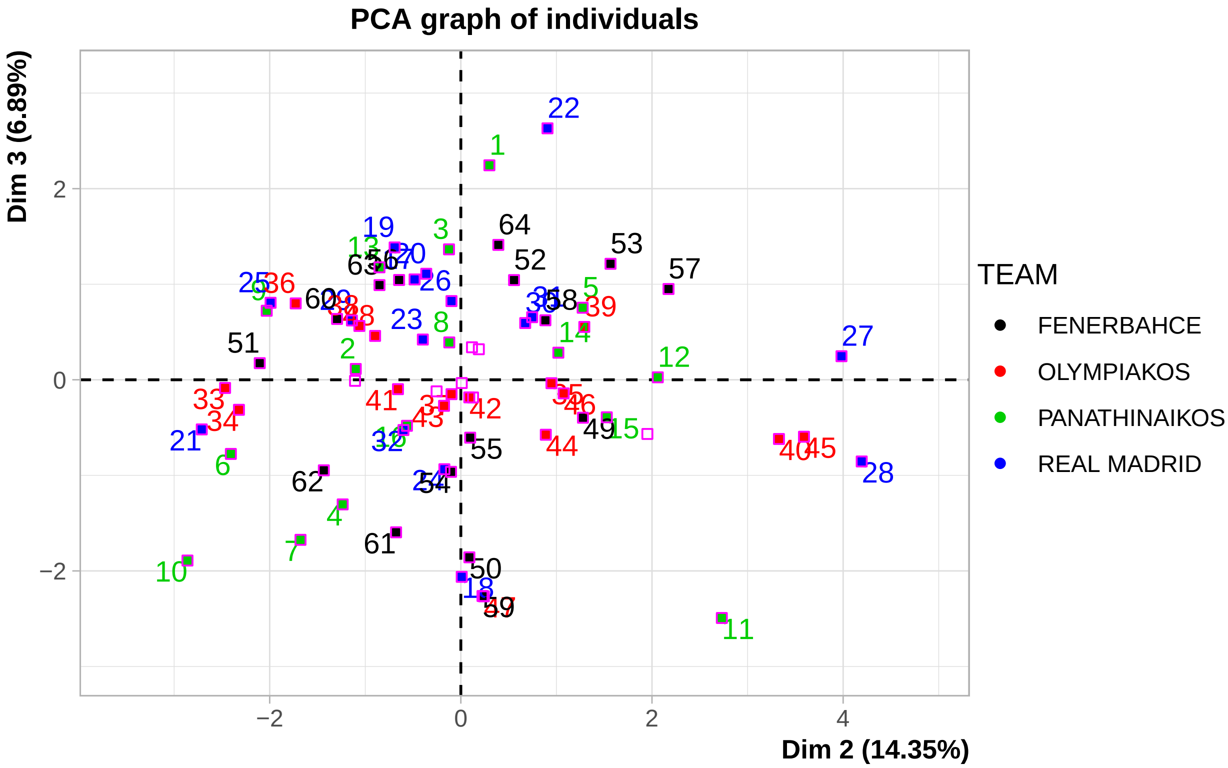
```
##
## [[3]]
```


PCA graph of individuals
Dim 4 (5.82%)
Dim 1 (52.24%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

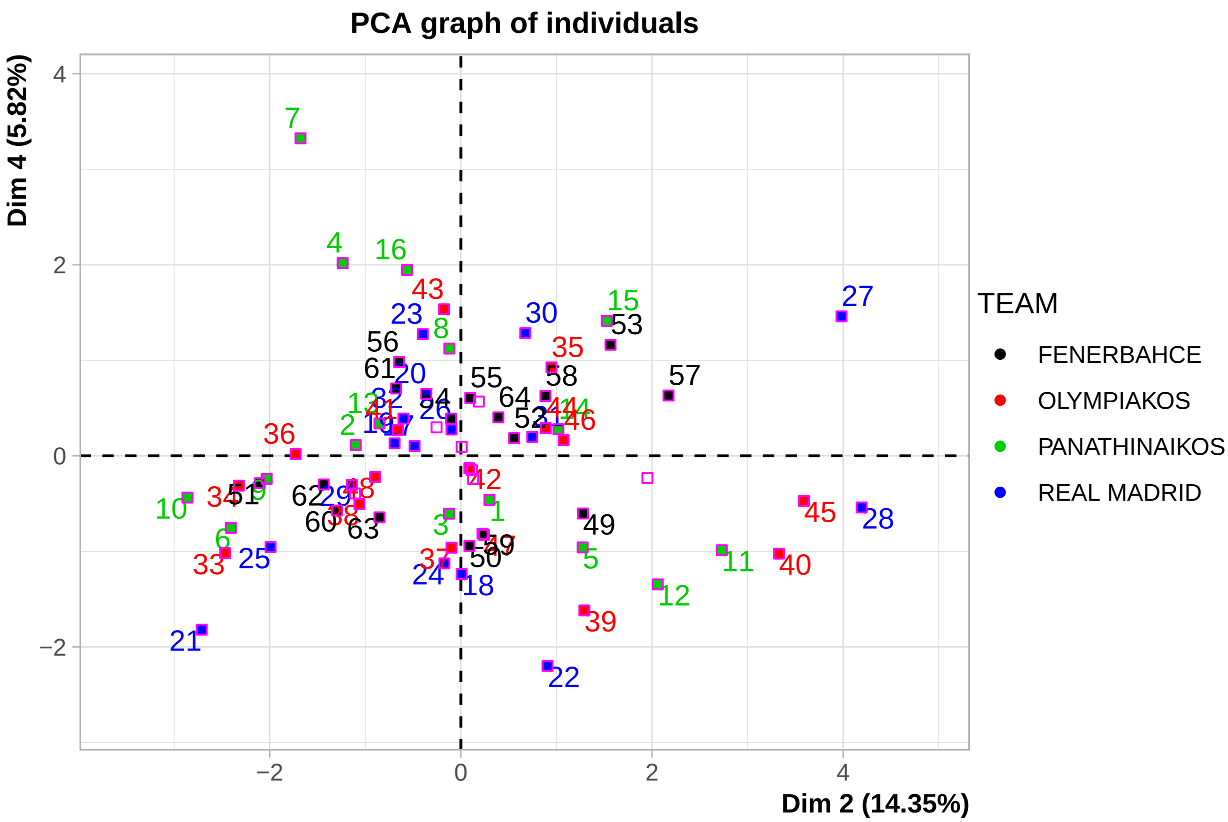
```
##
## [[4]]
```


PCA graph of individuals
Dim 5 (4.44%)
Dim 1 (52.24%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

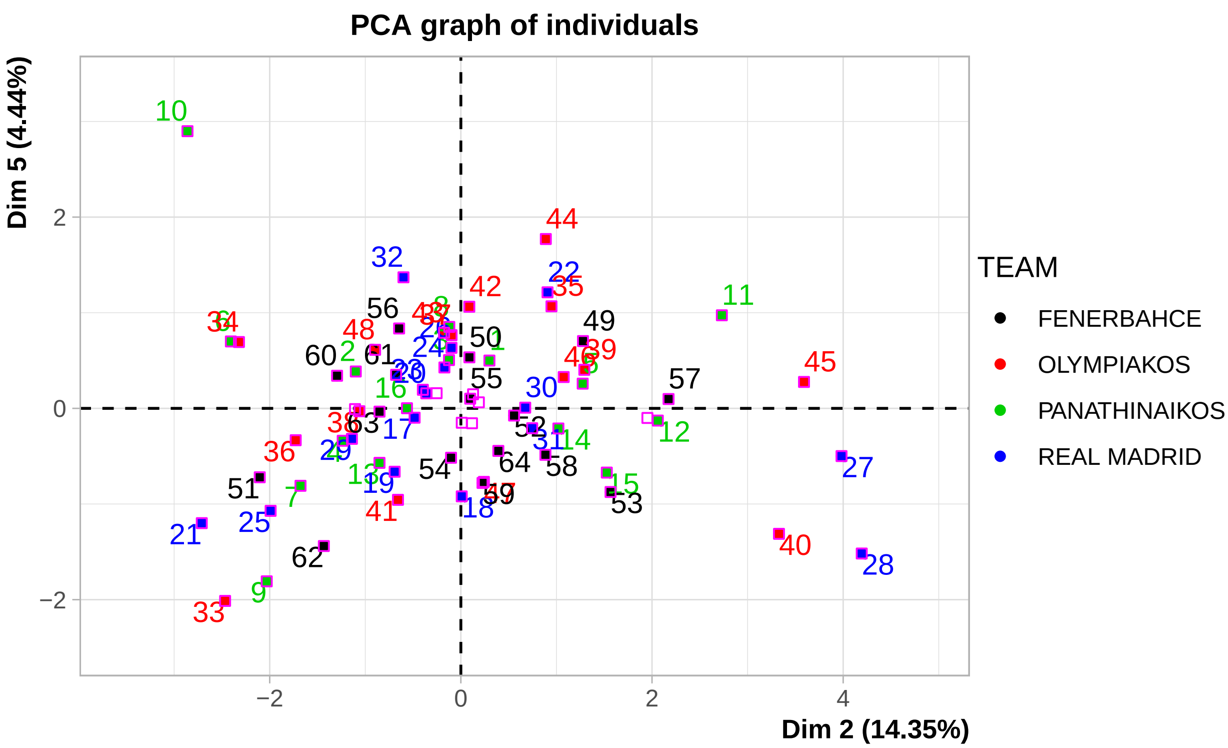
```
##
## [[5]]
```


PCA graph of individuals
Dim 3 (6.89%)
Dim 2 (14.35%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

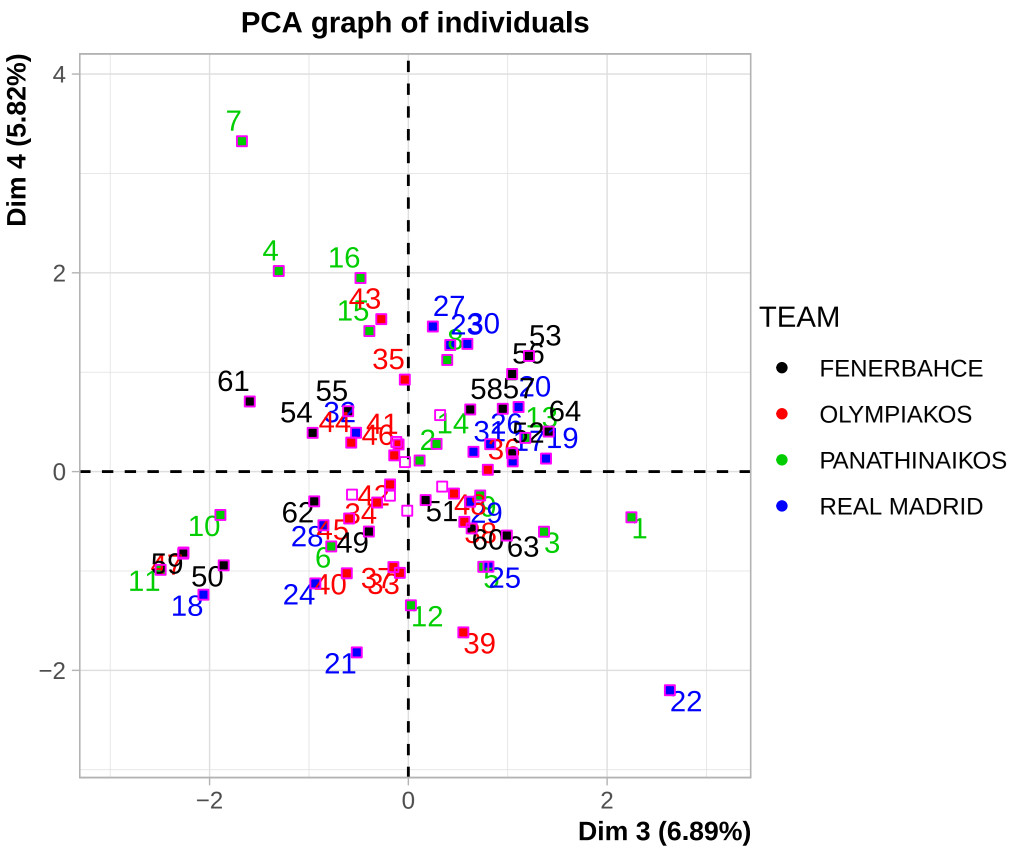
```
##
## [[6]]
```


PCA graph of individuals
Dim 4 (5.82%)
Dim 2 (14.35%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

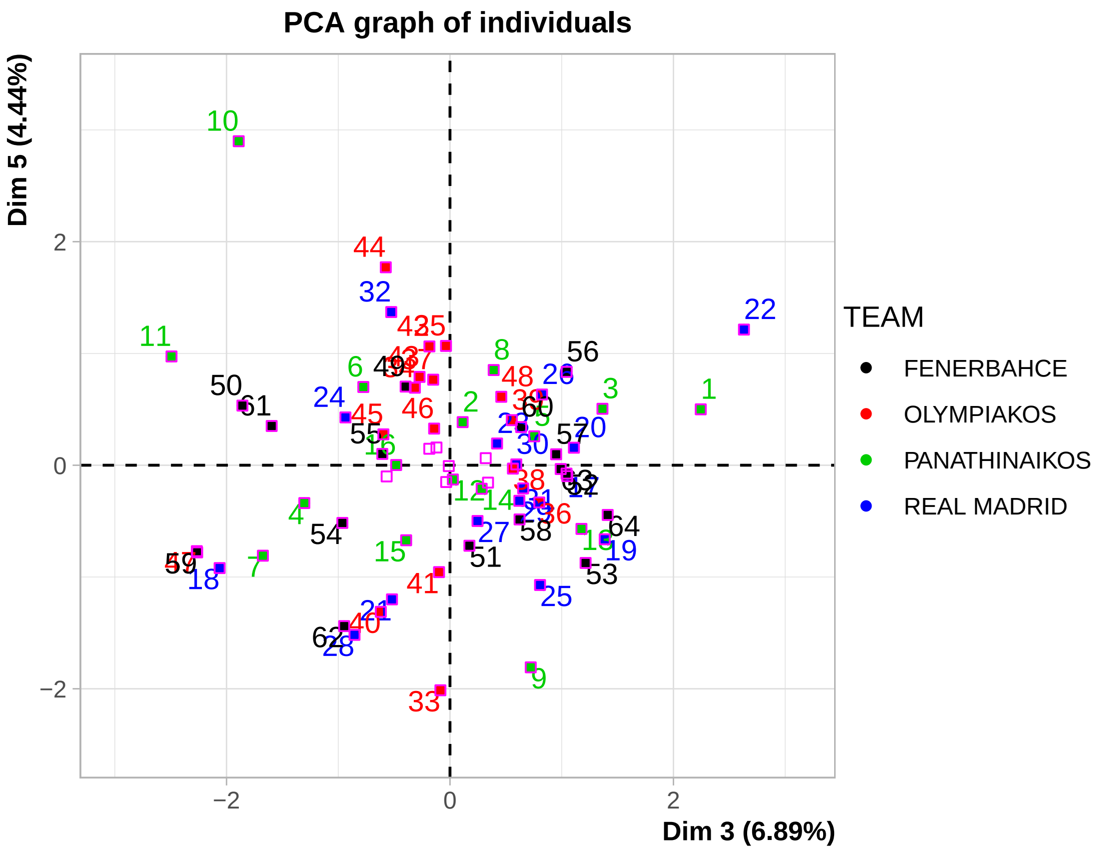
```
##
## [[7]]
```


PCA graph of individuals
Dim 5 (4.44%)
Dim 2 (14.35%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

```
##
## [[8]]
```

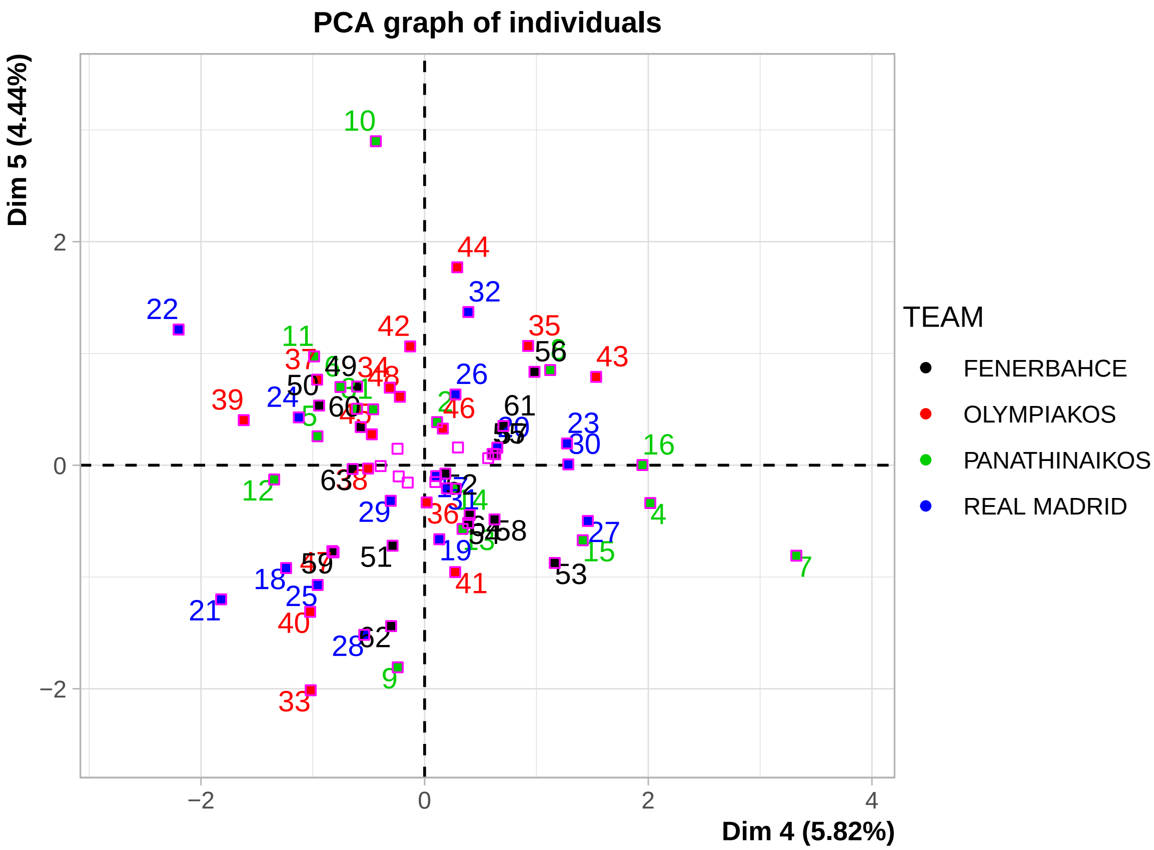

PCA graph of individuals
Dim 4 (5.82%)
Dim 3 (6.89%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

```
##
## [[9]]
```



PCA graph of individuals
Dim 5 (4.44%)
Dim 3 (6.89%)
TEAM
FENERBAHCE
OLYMPIAKOS
PANATHINAIKOS
REAL MADRID

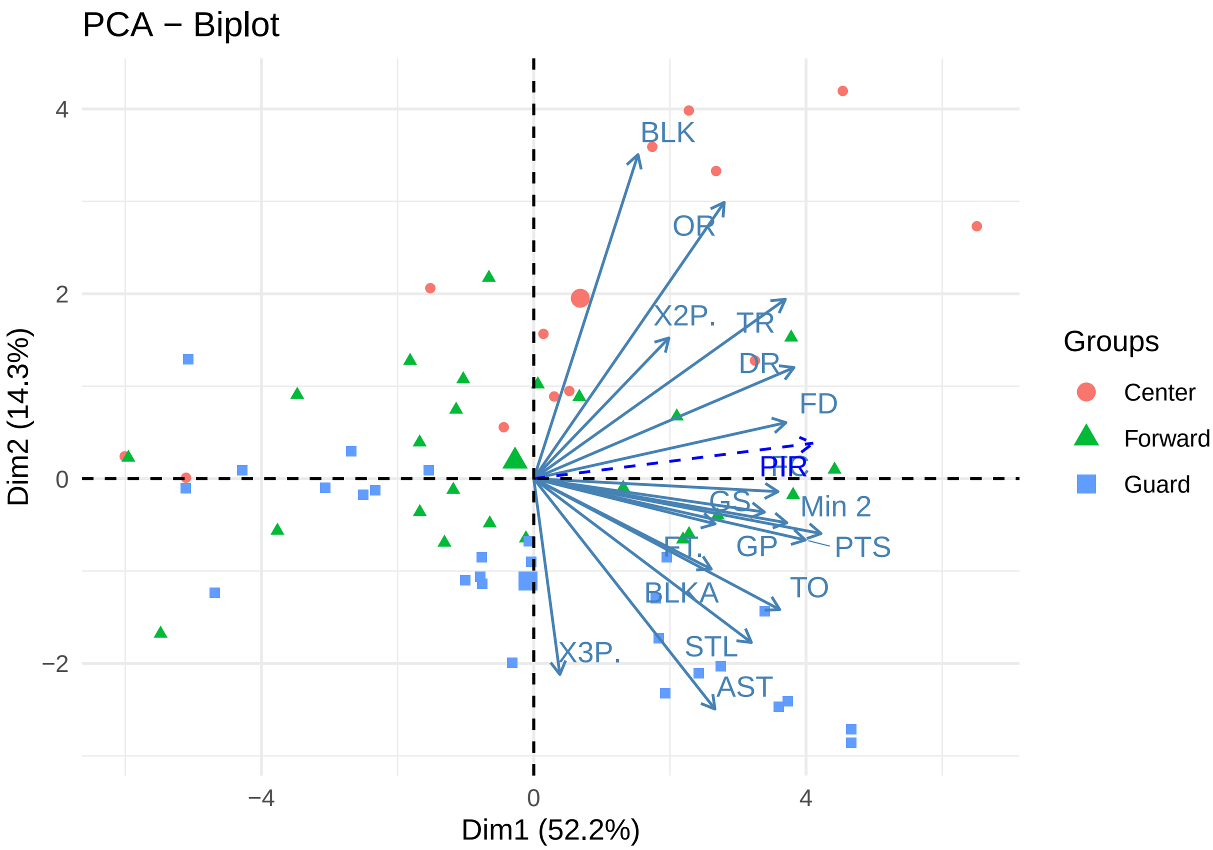

```
##
## [[10]]
```



g) Interpret the individual plots. (3p)

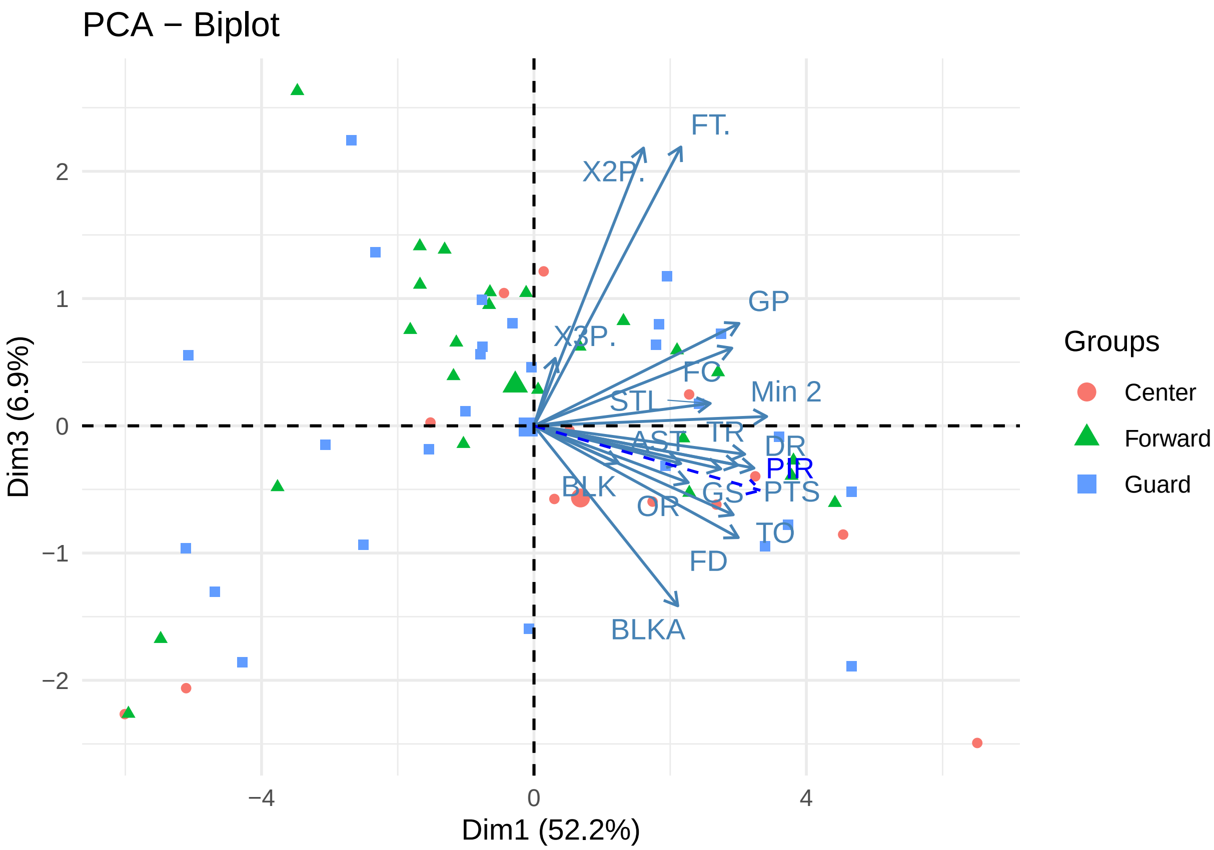
Dimensions 1 and 2

```r
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION) +
  theme_minimal()
```



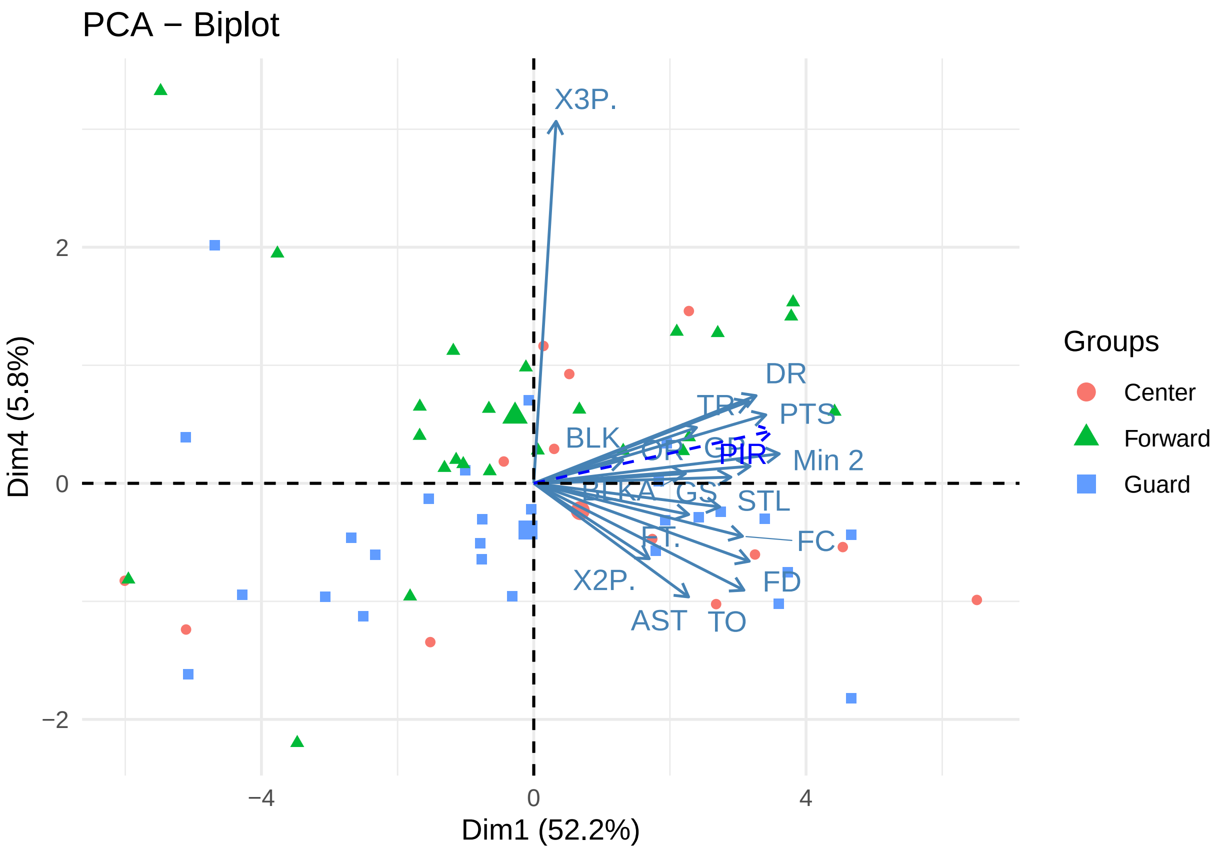
Dimensions 1 and 3

```r
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(1,3)) +
  theme_minimal()
```



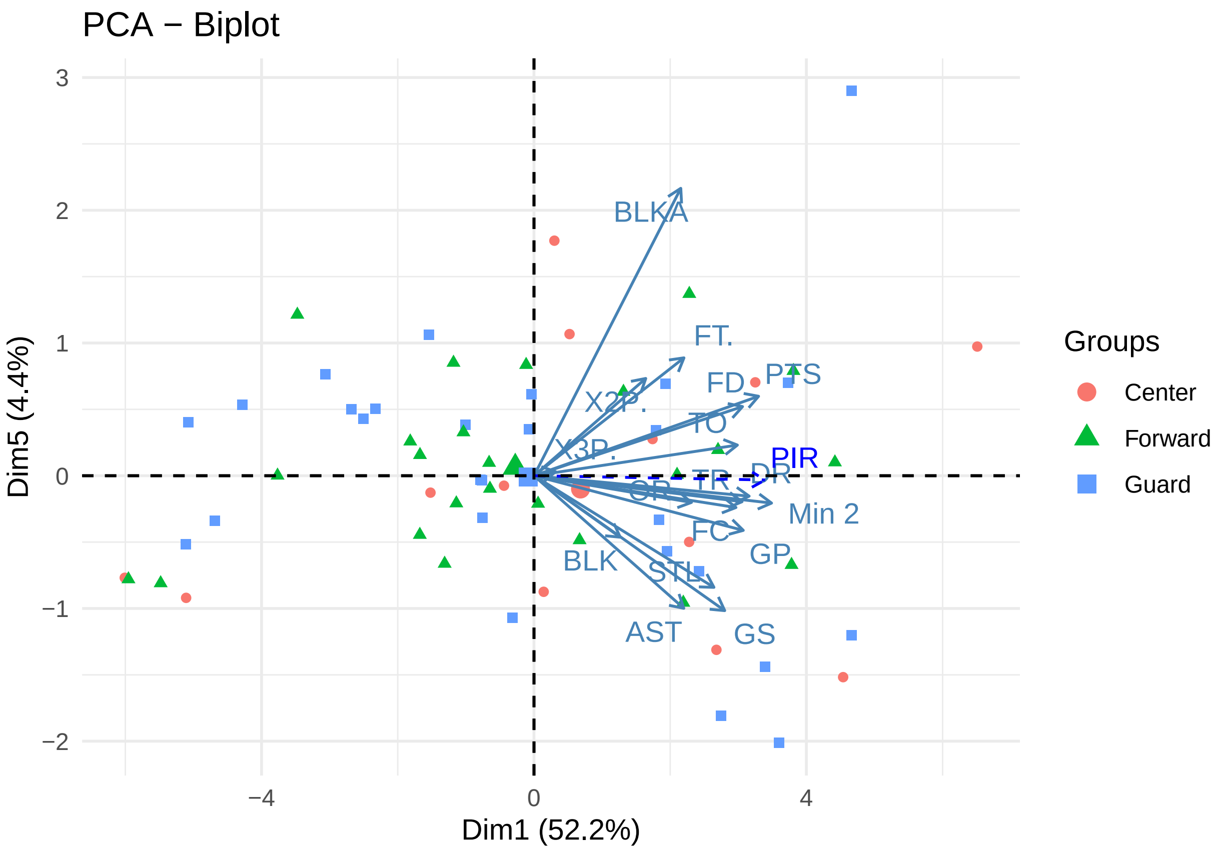
Dimensions 1 and 4

```r
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(1,4)) +
  theme_minimal()
```



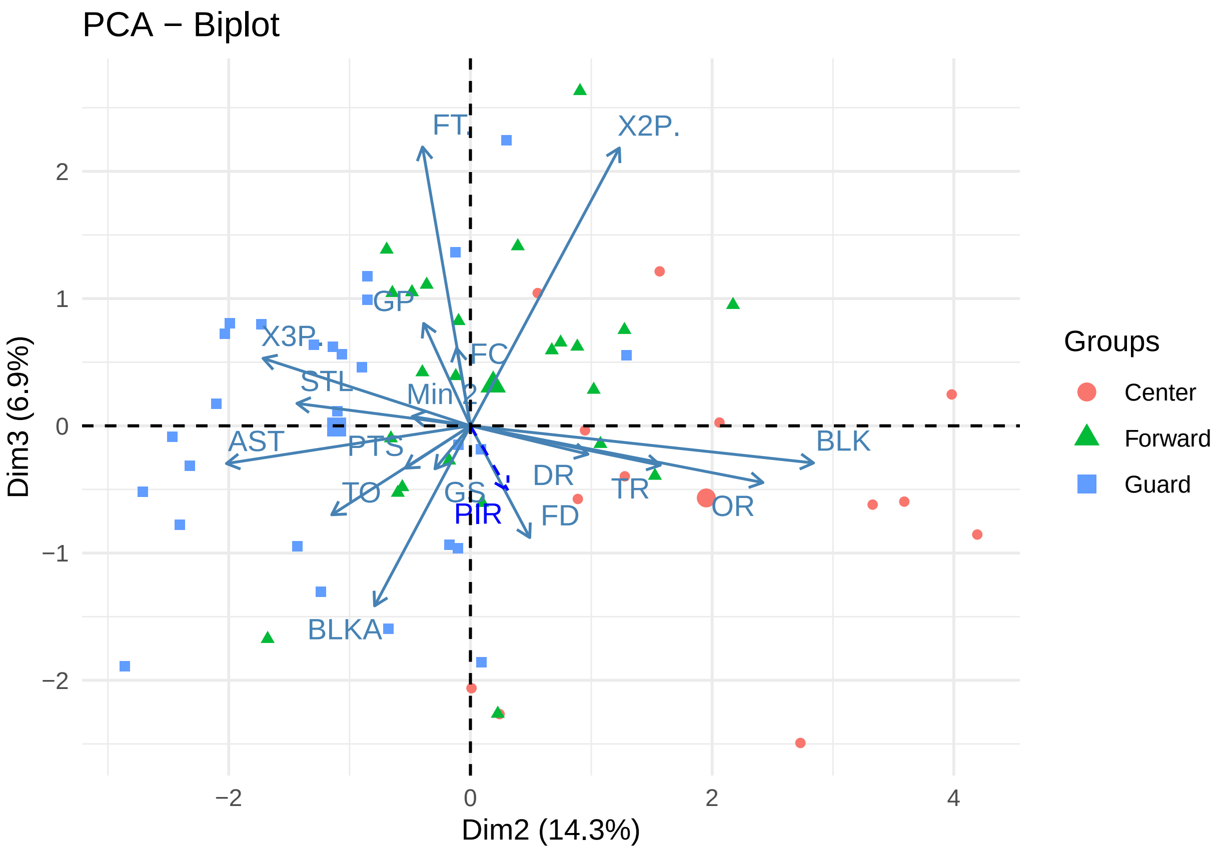
Dimensions 1 and 5

```r
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(1,5)) +
  theme_minimal()
```



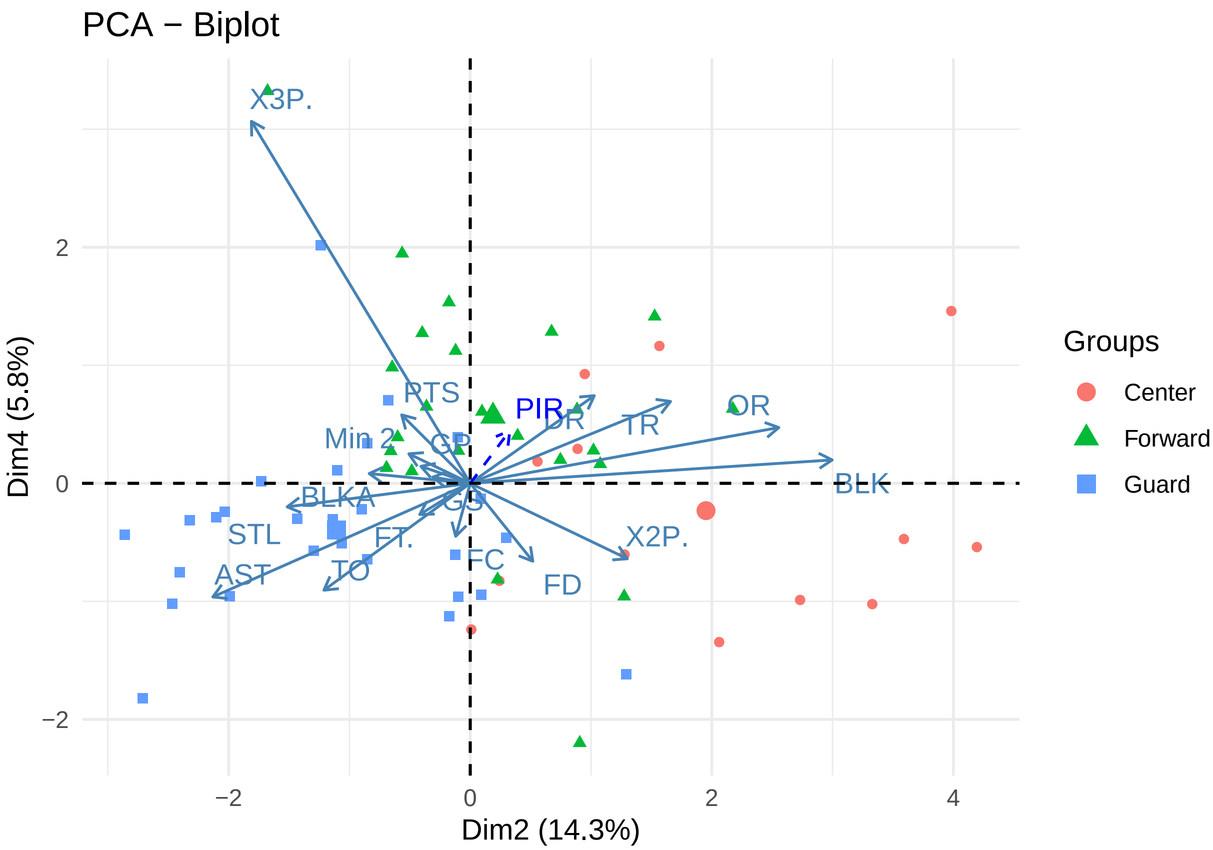
Dimensions 2 and 3

```
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(2,3)) +
  theme_minimal()
```



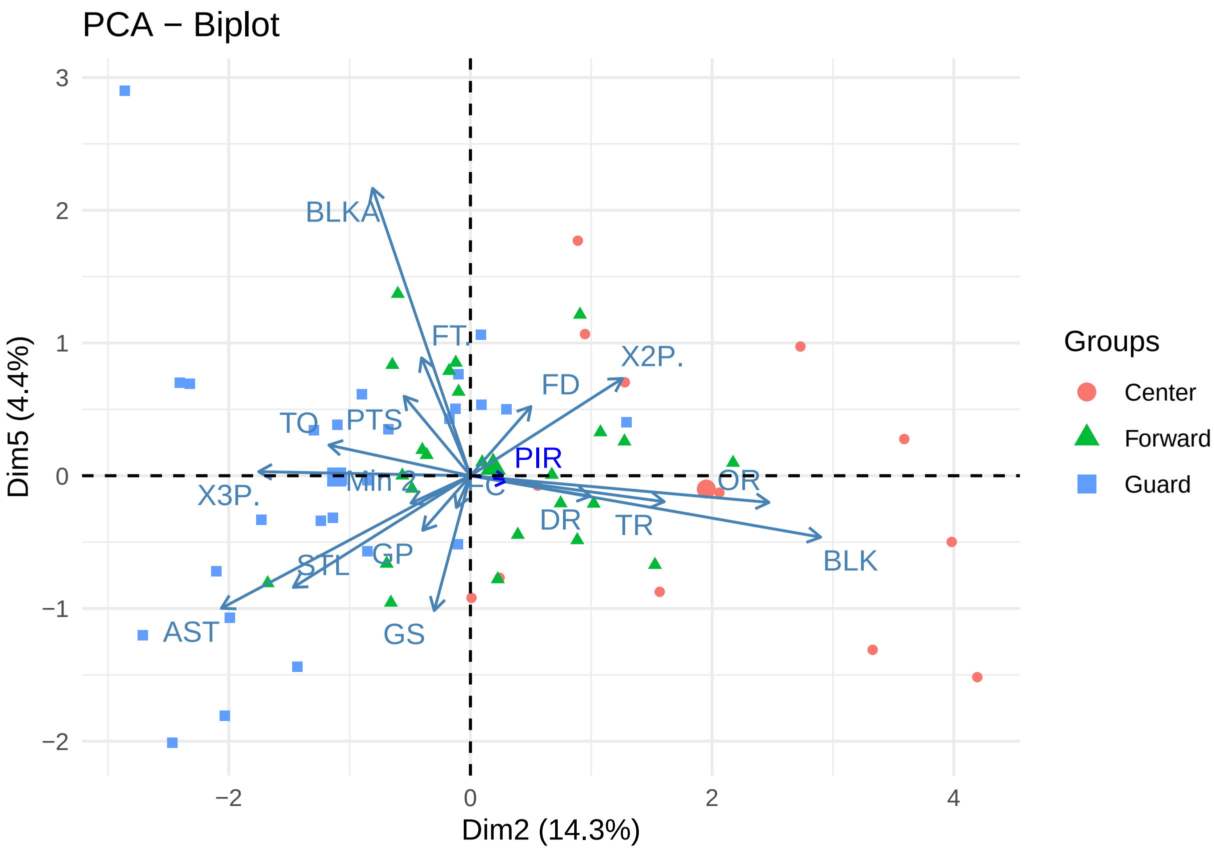
Dimensions 2 and 4

```
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(2,4)) +
  theme_minimal()
```



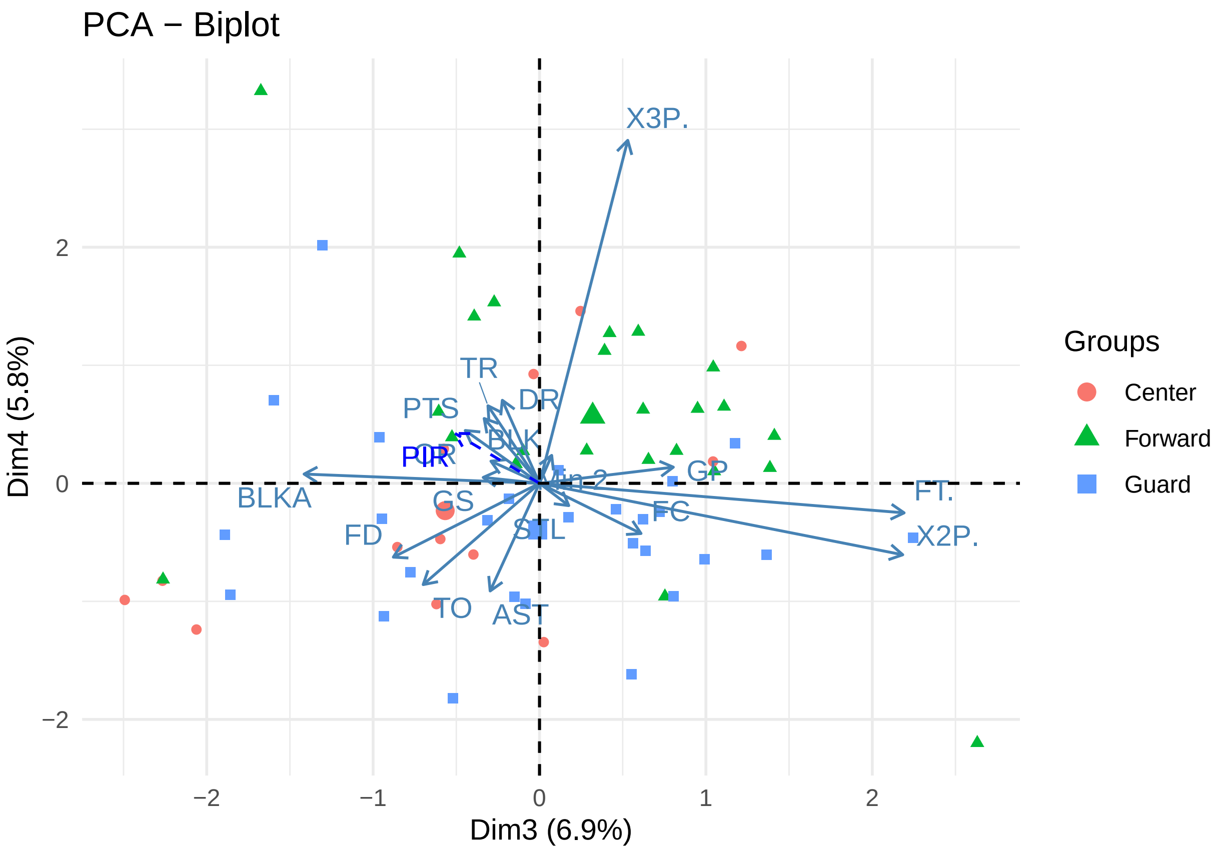
Dimensions 2 and 5

```
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(2,5)) +
  theme_minimal()
```



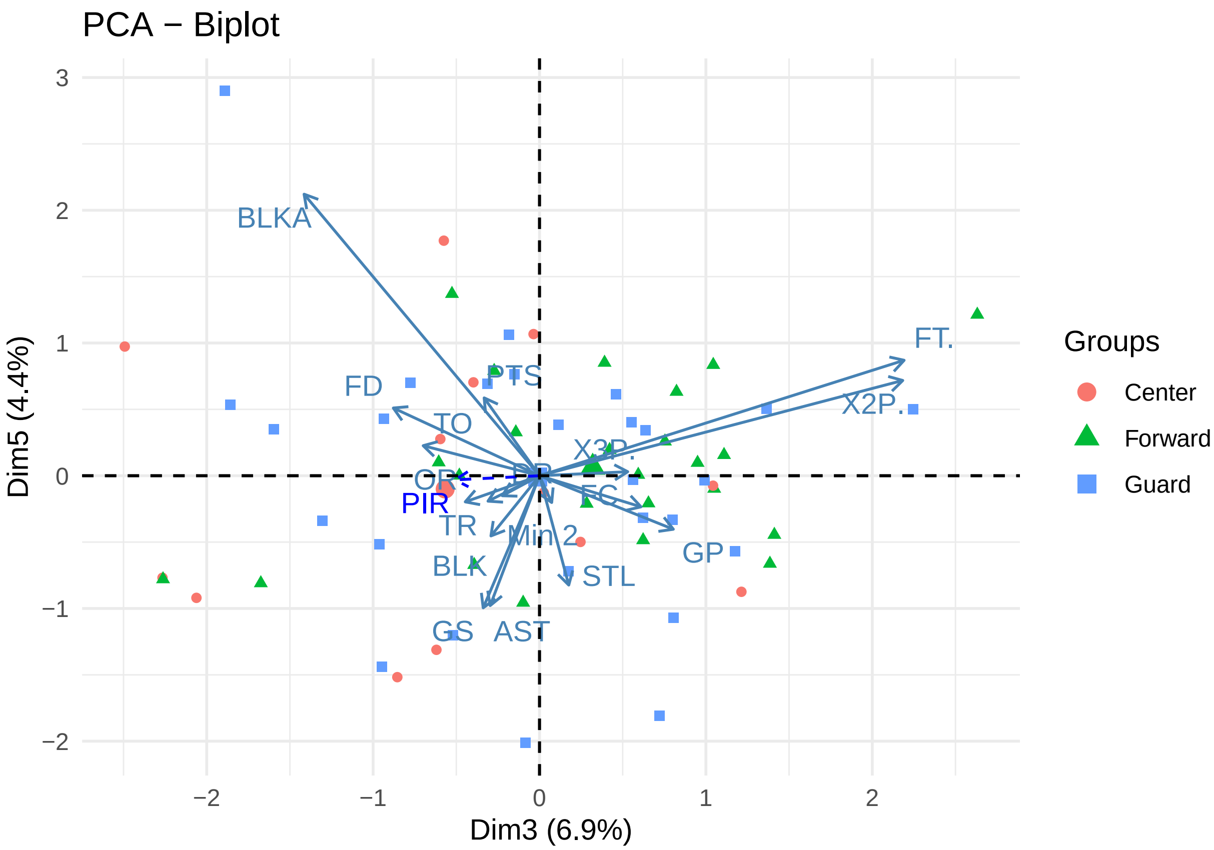
Dimensions 3 and 4

```
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(3,4)) +
  theme_minimal()
```



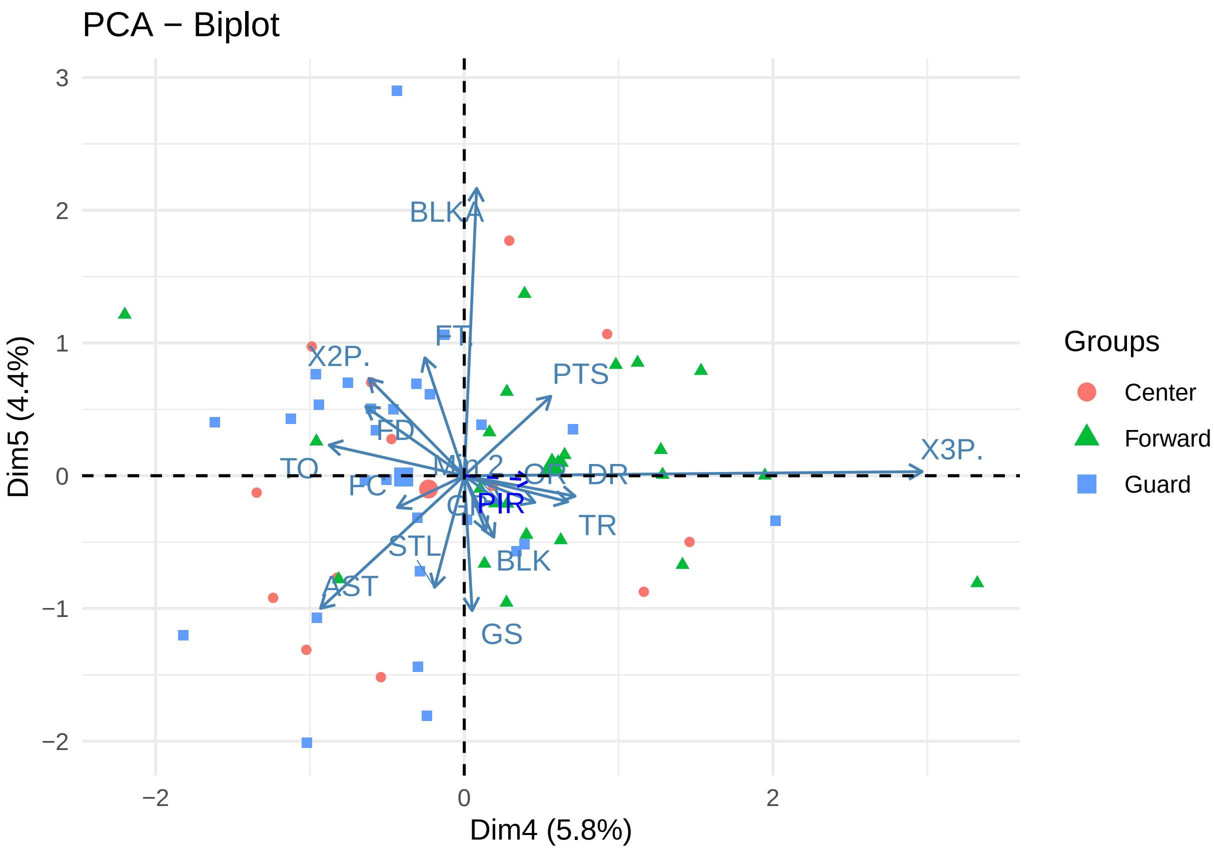
Dimensions 3 and 5

```r
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(3,5)) +
  theme_minimal()
```



Dimensions 4 and 5

```
fviz_pca_biplot(pca, repel = T,label = "var",habillage = data$POSITION,axes = c(4,5)) +
  theme_minimal()
```



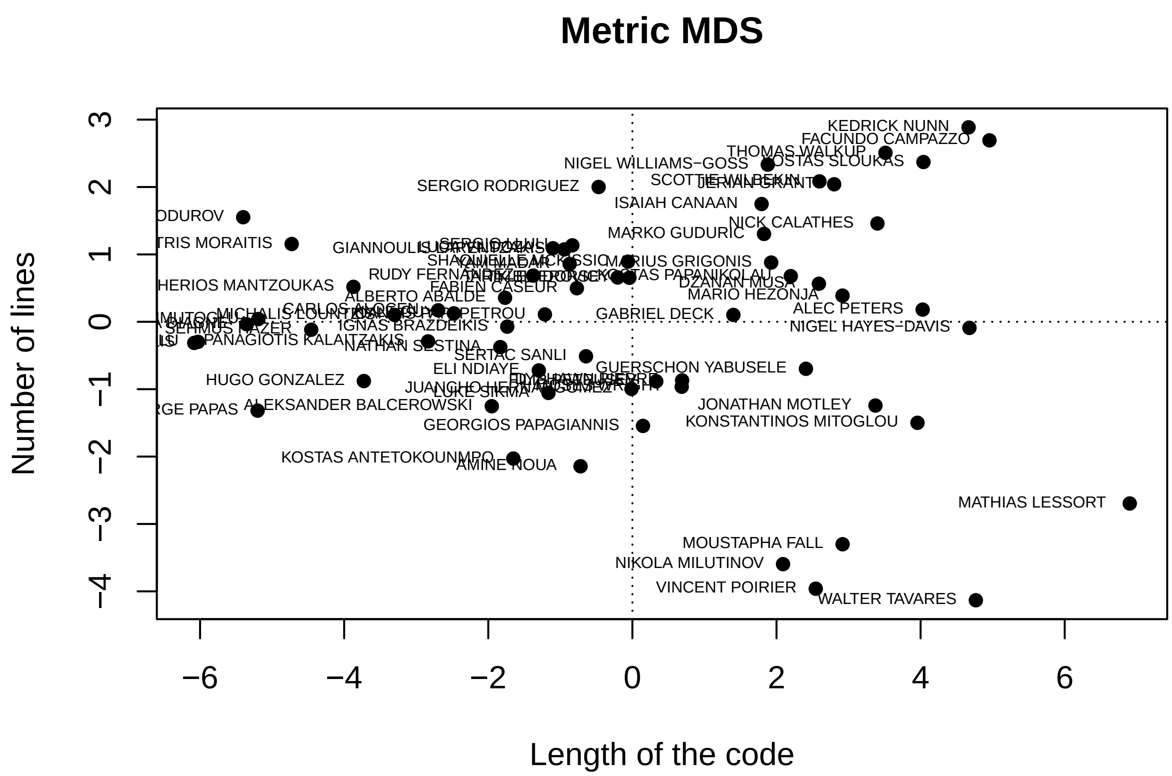
3. Application of MDS.

a) Apply metric MDS using Euclidean distance on scaled numerical variables. (2p)

```
numeric_data = data %>% select_if(is.numeric)
scaled_data = scale(numeric_data)
dist_matrix = dist(scaled_data, method = "euclidean")
mds_result = cmdscale(dist_matrix,eig=TRUE)
```

b) Plot the data using the points on the first two coordinates using players names as label. (2p)

```
x  =  mds_result$points[,1]
y  =  mds_result$points[,2]

plot(x, y, xlab="Length of the code", ylab="Number of lines",
     main="Metric MDS", type="p",pch = 16)
text(x, y, labels = data$PLAYER, cex = 0.5,pos = 2,col = "black")
abline(v = 0, h = 0,lty = 3)
```



c) **Interpret the plot (3p).**

d) **Calculate gower distance including variable "POSITION" to the data matrix (3p).**

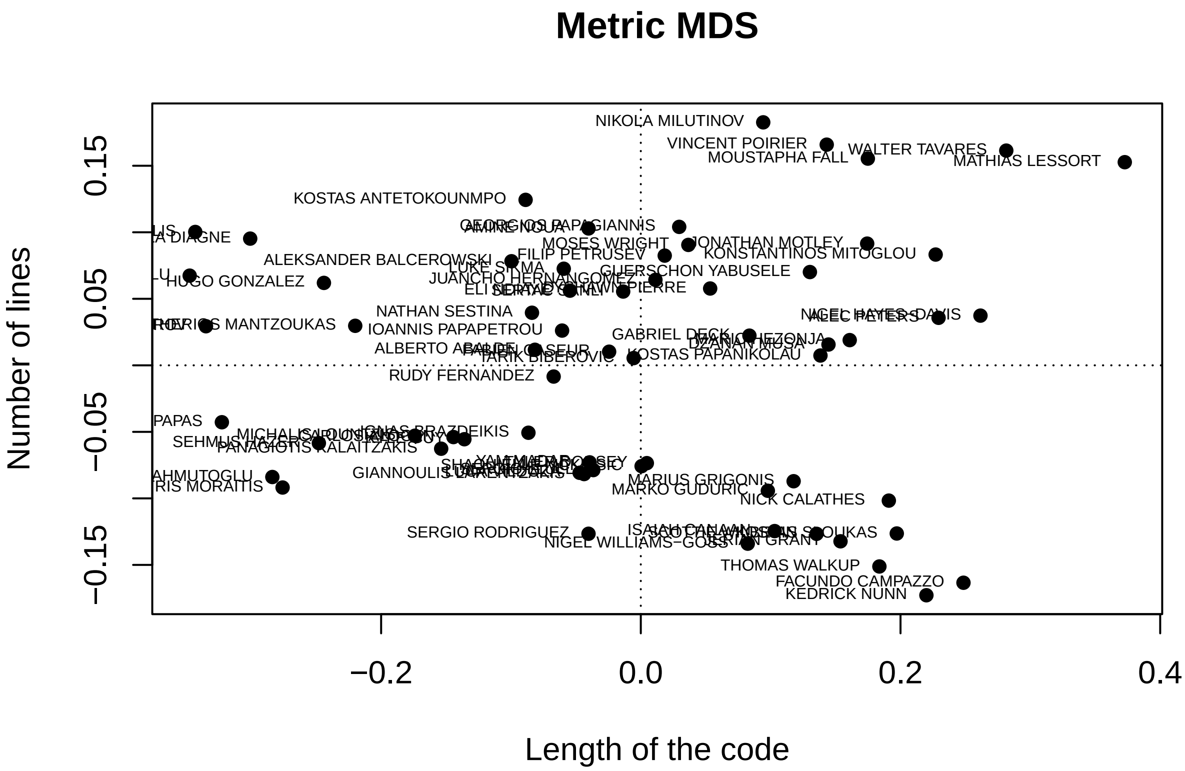
```r
numeric_pos_data = numeric_data %>%
                   add_column(data$POSITION, .before = 1) %>%
                   rename("POSITION" = "data$POSITION")

gower_dist = daisy(numeric_pos_data, metric = "gower")
#print(as.matrix(gower_dist))
```

e) **Apply metric MDS on gower distance matrix (2p).**

```r
mds_result = cmdscale(gower_dist, eig = T)
```

f) **Plot individual plots on the first two coordinates (2p).**

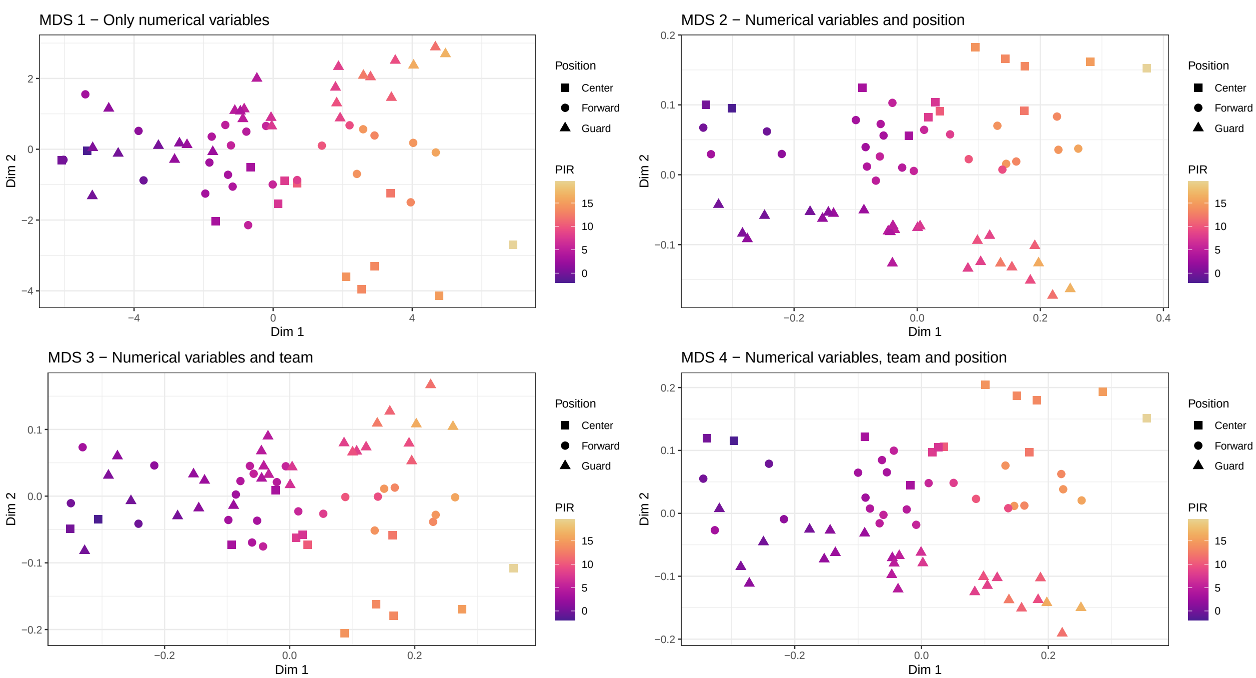
```r
x  =  mds_result$points[,1]
y  =  mds_result$points[,2]

plot(x, y, xlab="Length of the code", ylab="Number of lines",
     main="Metric MDS", type="p",pch = 16)
text(x, y, labels = data$PLAYER, cex=0.5,pos = 2)
abline(v = 0, h = 0,lty = 3)
```



g) **Use different categorical and numerical variables as labels so as to explain clusters that are constructed.(5p)**

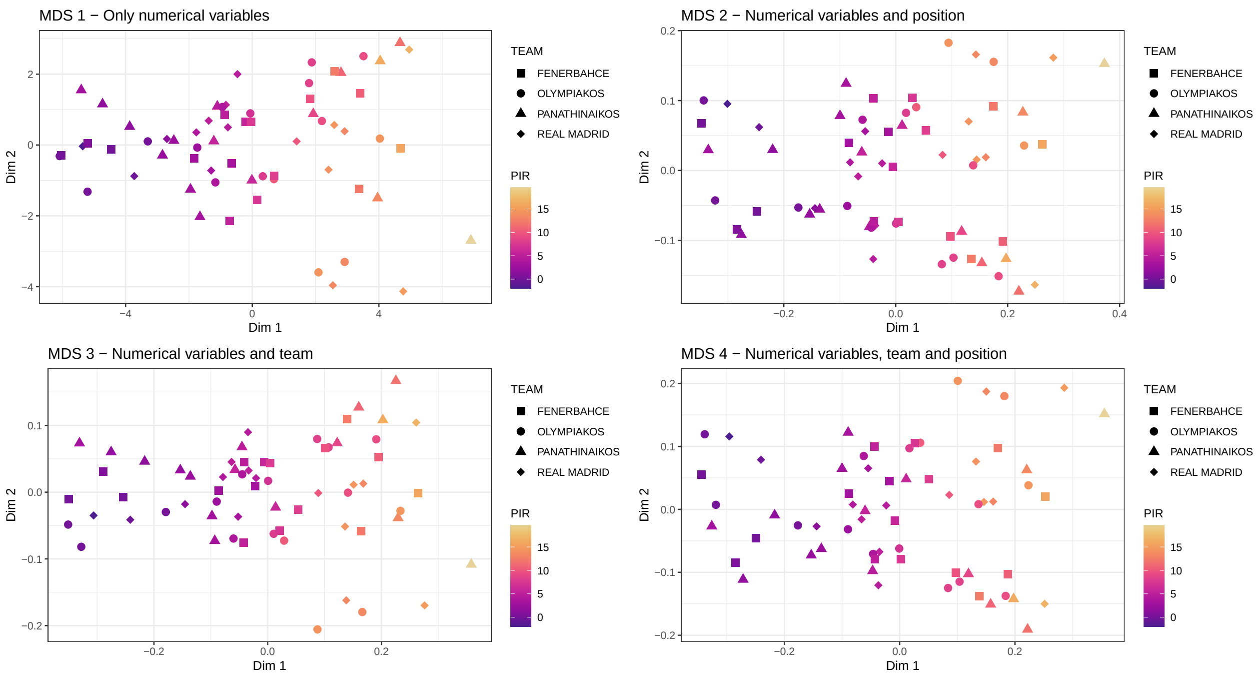
By Performance Index Rating and position

```
grid.arrange(p1_1, p2_1, p3_1 ,p4_1,ncol = 2, nrow = 2)
```



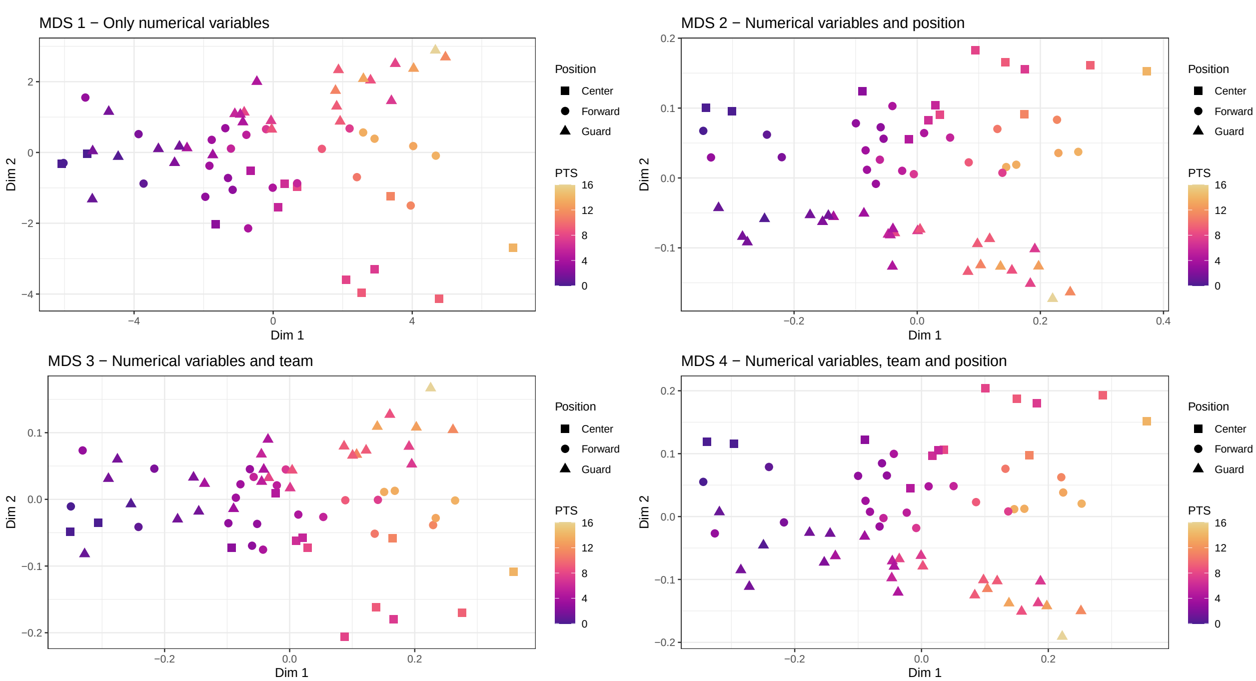
By Performance Index Rating and team

```
grid.arrange(p1_2, p2_2, p3_2 ,p4_2,ncol = 2, nrow = 2)
```



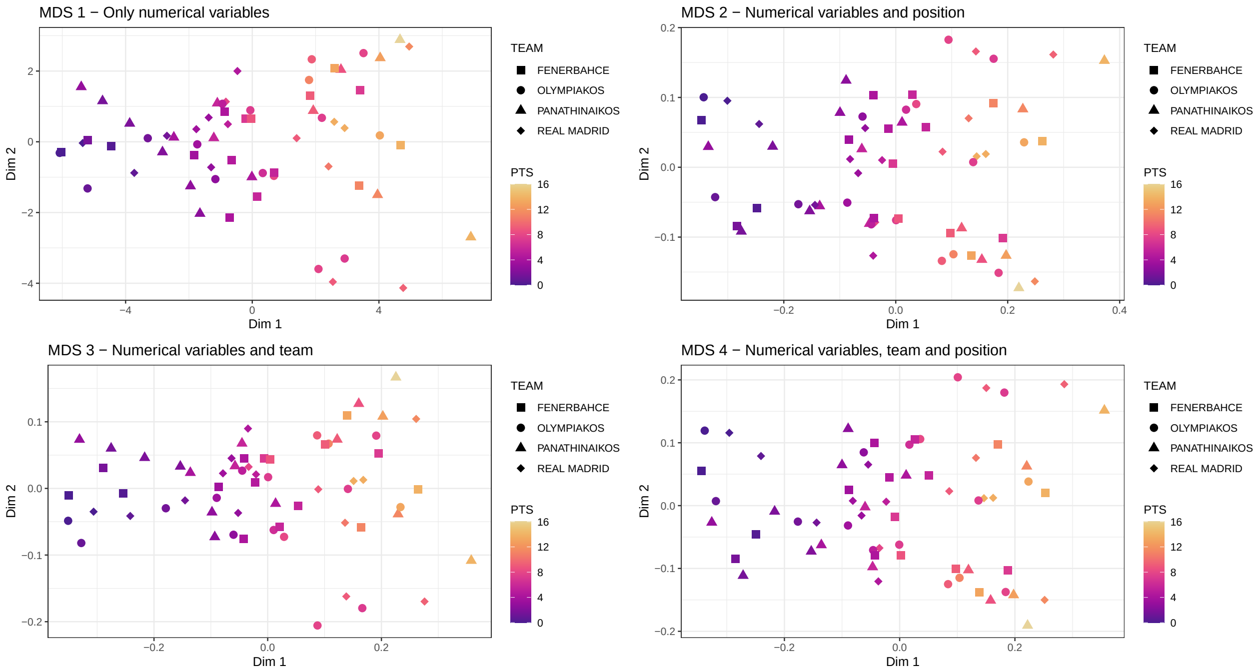
By points scored and position

```
grid.arrange(p1_3, p2_3, p3_3 ,p4_3,ncol = 2, nrow = 2)
```



By points scored and team

```
grid.arrange(p1_4, p2_4, p3_4 ,p4_4,ncol = 2, nrow = 2)
```



h) Which MDS do you think better group the individuals? Why? (3p)