TP_Classification

Chainez_MEDJOUTI

2024-05-24

Etape 1: Importation et description des données

```
data <- read.csv2("/Users/chainez/Documents/Classification/TP evaluation/foot.csv")
```

Le jeu de données “foot.csv” comprend des informations détaillées sur 393 joueurs de football, décrites par 36 variables. Voici un aperçu des principales variables :

Variables Clés Player: Nom du joueur (caractère)

Pos : Poste du joueur sur le terrain (caractère)

Squad : Équipe à laquelle le joueur appartient (caractère)

Age : Âge du joueur (numérique)

Born : Année de naissance du joueur (numérique)

MP : Nombre de matchs joués (numérique)

Min : Minutes jouées (numérique)

Goals : Nombre de buts marqués (numérique)

Shots : Nombre de tirs (caractère)

SoT : Tirs cadrés (caractère)

ShoDist : Distance moyenne des tirs (caractère)

PasTotCmp. : Passes complètes totales (caractère)

PasShoCmp. : Passes courtes complètes (caractère)

PasMedCmp. : Passes moyennes complètes (caractère)

PasLonCmp. : Passes longues complètes (caractère)

PasAss : Passes décisives (caractère)

PPA : Passes en profondeur (caractère)

PasAtt : Passes tentées (caractère)

CK : Corners (caractère)

SCA : Actions menant à un tir (caractère)

GCA : Actions menant à un but (caractère)

Tkl : Tacles (caractère)

Blocks : Blocages (caractère)

Int : Interceptions (caractère)

Clr : Dégagements (caractère)

Err : Erreurs menant à un tir ou un but (caractère)

Touches : Touches de balle (caractère)

ToAtt : Tentatives de dribble (caractère)

Carries : Conduites de balle (caractère)

CPA : Passes en profondeur complètes (caractère)

Rec : Réceptions de passes (caractère)

CrdY : Cartons jaunes (caractère)

CrdR : Cartons rouges (caractère)

Off : Hors-jeu (caractère)

Crs : Centres (caractère)

Recov : Récupérations (caractère)

I - Préparation du jeu de données.

```
print(summary(data))
```

```
##     Player              Pos               Squad               Age       
##  Length:393         Length:393         Length:393         Min.   :16.00  
##  Class :character   Class :character   Class :character   1st Qu.:23.00  
##  Mode  :character   Mode  :character   Mode  :character   Median :26.00  
##                                                           Mean   :26.24  
##                                                           3rd Qu.:29.00  
##                                                           Max.   :38.00  
##       Born            MP             Min             Goals       
##  Min.   :1984   Min.   : 1.00   Min.   :   3.0   Min.   : 0.000  
##  1st Qu.:1993   1st Qu.: 7.00   1st Qu.: 270.0   1st Qu.: 0.000  
##  Median :1996   Median :14.00   Median : 756.0   Median : 0.000  
##  Mean   :1996   Mean   :12.33   Mean   : 799.4   Mean   : 1.453  
##  3rd Qu.:1999   3rd Qu.:18.00   3rd Qu.:1252.0   3rd Qu.: 2.000  
##  Max.   :2006   Max.   :23.00   Max.   :2070.0   Max.   :25.000  
##     Shots              SoT              ShoDist          PasTotCmp.       
##  Length:393         Length:393         Length:393         Length:393        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##   PasShoCmp.         PasMedCmp.         PasLonCmp.           PasAss        
##  Length:393         Length:393         Length:393         Length:393        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##
```

```
##
##
##      PPA               PasAtt              CK               SCA
##  Length:393         Length:393         Length:393         Length:393
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      GCA               Tkl              Blocks              Int
##  Length:393         Length:393         Length:393         Length:393
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      Clr               Err              Touches             ToAtt
##  Length:393         Length:393         Length:393         Length:393
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Carries             CPA               Rec               CrdY
##  Length:393         Length:393         Length:393         Length:393
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      CrdR              Off               Crs              Recov
##  Length:393         Length:393         Length:393         Length:393
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
```

Je commence par transformer les variables quantitatives en variables numériques. Cette étape est nécessaire pour garantir que les données sont dans le bon format pour les analyses statistiques et les visualisations.

```
data <- data %>%
  mutate(across(c(4, 6:36), as.numeric))
```

Je m'assure que chaque joueur n'apparaît qu'une seule fois, en conservant toutes les colonnes associées au joueur unique.

Les postes des joueurs sont regroupés en quatre catégories principales : attaquants (FW), milieux de terrain (MF), gardiens de but (GK), et défenseurs (DF). Cette simplification permet une analyse plus claire et plus cohérente des positions des joueurs.

Je mets les noms des joueurs en tant qu'identifiants de ligne, ce qui facilite le suivi et la manipulation des données pour chaque joueur individuellement.

Je supprime des valeurs aberrantes pour plusieurs variables (Goals, PPA, CK, Blocks, ToAtt, Carries, CPA).

Ces valeurs extrêmes peuvent biaiser les analyses et sont donc éliminées pour obtenir des résultats plus représentatifs.

Enfin, je filtre les joueurs pour ne garder que ceux qui ont joué un nombre significatif de minutes pendant la saison. La moyenne des minutes jouées est de 809.5495, et le premier quartile est de 285.75 minutes. Seuls les joueurs ayant joué plus de 285.75 minutes sont conservés, assurant que l'analyse se concentre sur les joueurs ayant une contribution significative.

```
data <- data %>% distinct(Player, .keep_all = TRUE) # Supprimer les joueurs en doublon

unique(data$Pos) # Re travaillé le poste des joueurs pour n'avoir que 4 modalités : FW = attaquant, MF=
```

```
##  [1] "DF"   "FWMF" "MF"   "FW"   "GK"   "MFFW" "FWDF" "DFMF" "MFDF" "DFFW"
```

```
data <- data %>%
  mutate(Pos = case_when(
    Pos %in% c("FW", "FWMF", "FWDF", "DFFW") ~ "FW",
    Pos %in% c("MF", "MFFW", "MFDF", "DFMF") ~ "MF",
    Pos %in% c("GK") ~ "GK",
    Pos %in% c("DF") ~ "DF",
    TRUE ~ Pos
  ))
unique(data$Pos)
```

```
## [1] "DF" "FW" "MF" "GK"
```

```
row.names(data) <- data$Player # Mettre le nom des joueurs en row.names

data=data[data$Goals!=25,]
data=data[data$Goals!=17,]
data=data[data$PPA!=4.69,]
data=data[data$CK!=6.24,]
data=data[data$Blocks!=3.18,]
data=data[data$ToAtt!=8.1,]
data=data[data$Carries!=96.4,]
data=data[data$CPA!=3.25,]

#  Ne garder que les joueurs ayant significativement joué sur la saison
mean(data$Min) # 809.5495
```

```
## [1] 794.5984
```

```
quantile(data$Min, probs = c(0.25, 0.5, 0.75)) # 25% correspond a 285.75
```

```
##     25%     50%     75%
##  274.75  754.50 1248.75
```

```
data <- data %>%
  filter(Min >= 285.75)
```

```r
cat("Taille du jeu de données :", nrow(data), "observations\n")
```

```
## Taille du jeu de données : 280 observations
```

```r
cat("Nombre de variables :", ncol(data), "\n\n")
```

```
## Nombre de variables : 36
```

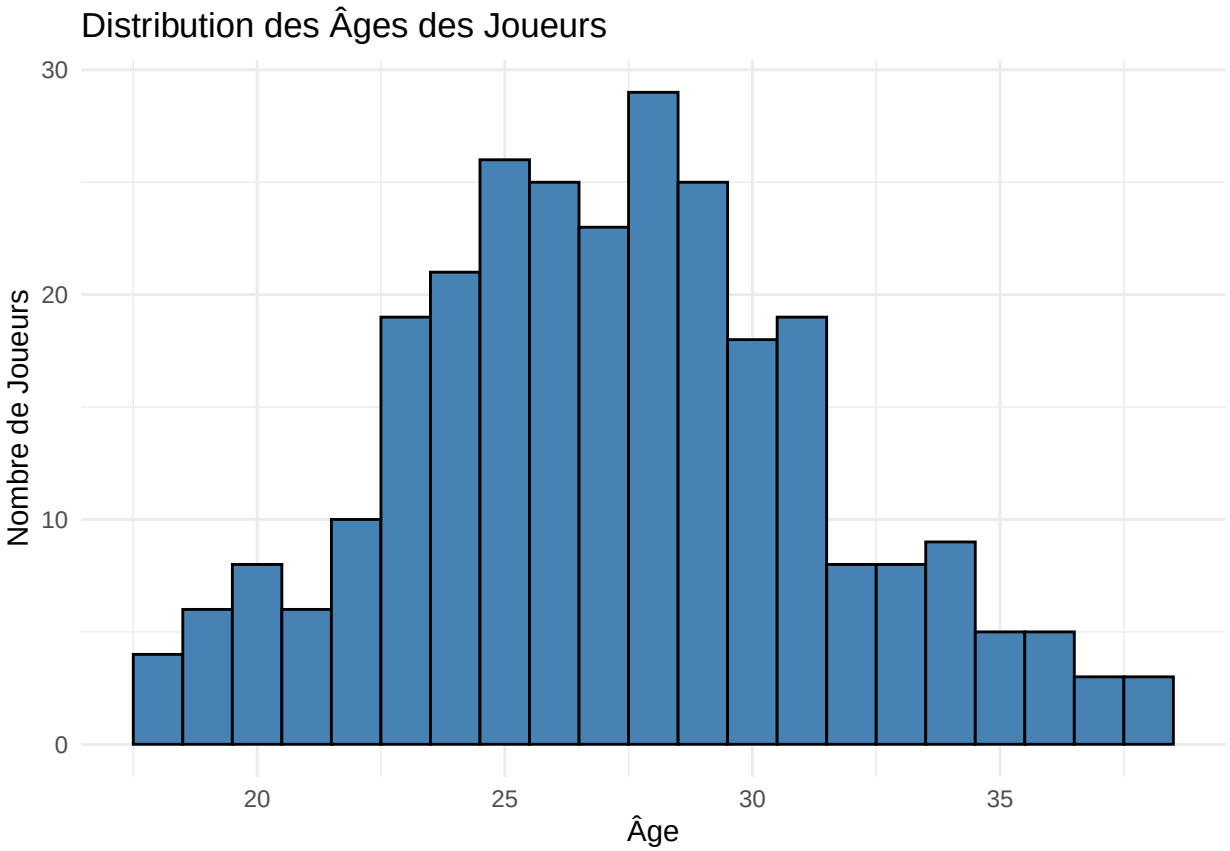
```r
print(colSums(is.na(data))) # Données manquantes
```

```
##     Player        Pos      Squad        Age       Born         MP        Min
##          0          0          0          0          0          0          0
##      Goals      Shots        SoT    ShoDist PasTotCmp. PasShoCmp. PasMedCmp.
##          0          0          0          0          0          0          0
## PasLonCmp.     PasAss        PPA     PasAtt         CK        SCA        GCA
##          0          0          0          0          0          0          0
##        Tkl     Blocks        Int        Clr        Err    Touches      ToAtt
##          0          0          0          0          0          0          0
##    Carries        CPA        Rec       CrdY       CrdR        Off        Crs
##          0          0          0          0          0          0          0
##      Recov
##          0
```

À présent, nous avons un jeu de données comportant 280 lignes pour 36 variables comprenant aucune valeur manquantes.

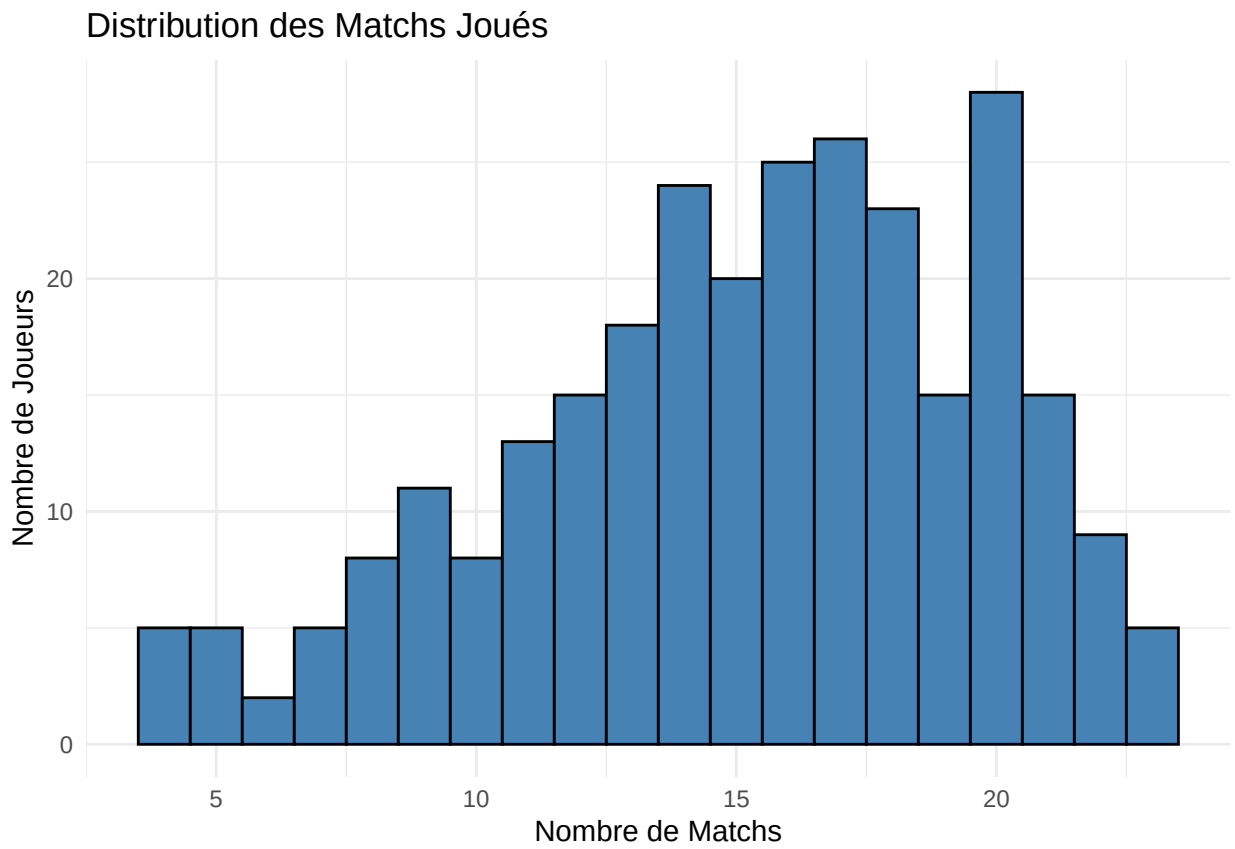
II) Statistiques Descriptives univariées

```r
ggplot(data, aes(x = Age)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution des Âges des Joueurs",
       x = "Âge",
       y = "Nombre de Joueurs")
```



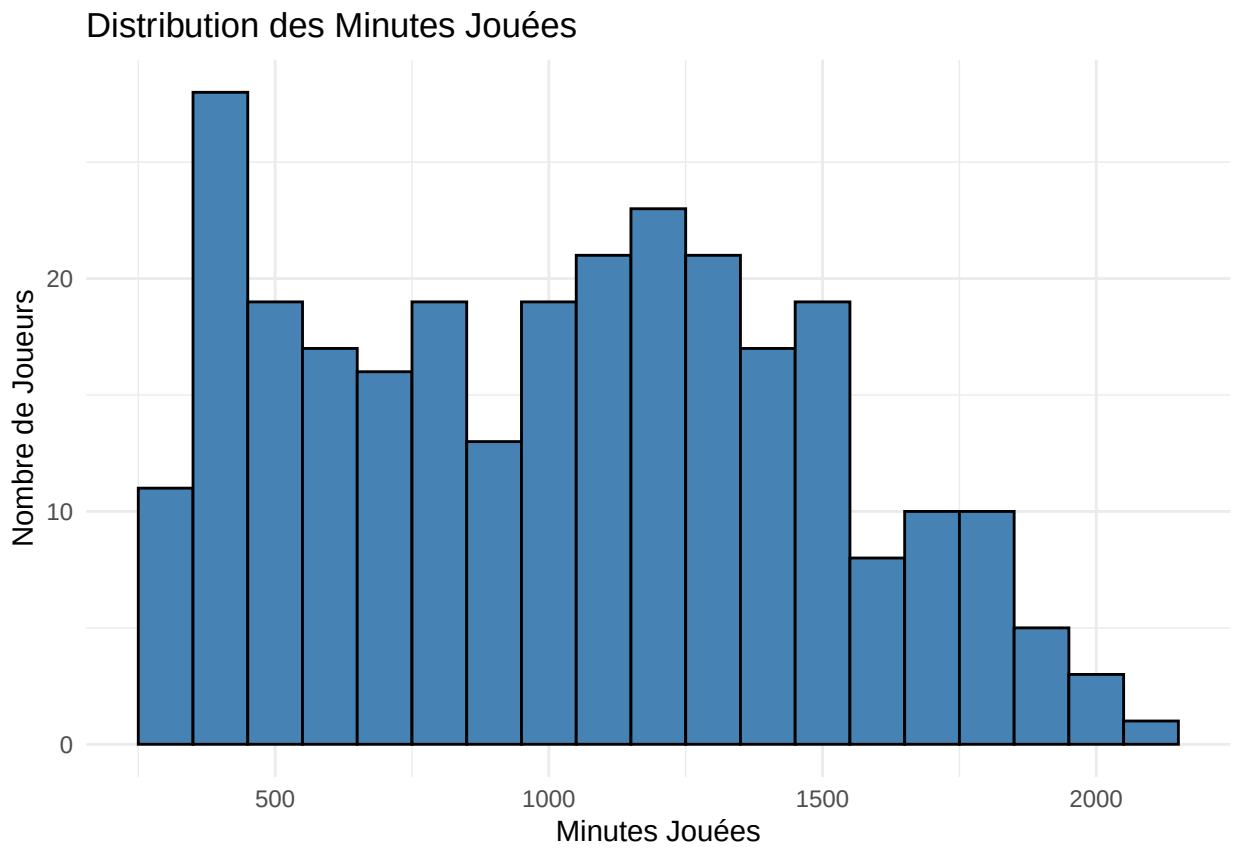
Âge : Les joueurs ont entre 16 et 38 ans avec une moyenne de 26.24 ans.

```
ggplot(data, aes(x = MP)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution des Matchs Joués",
       x = "Nombre de Matchs",
       y = "Nombre de Joueurs")
```



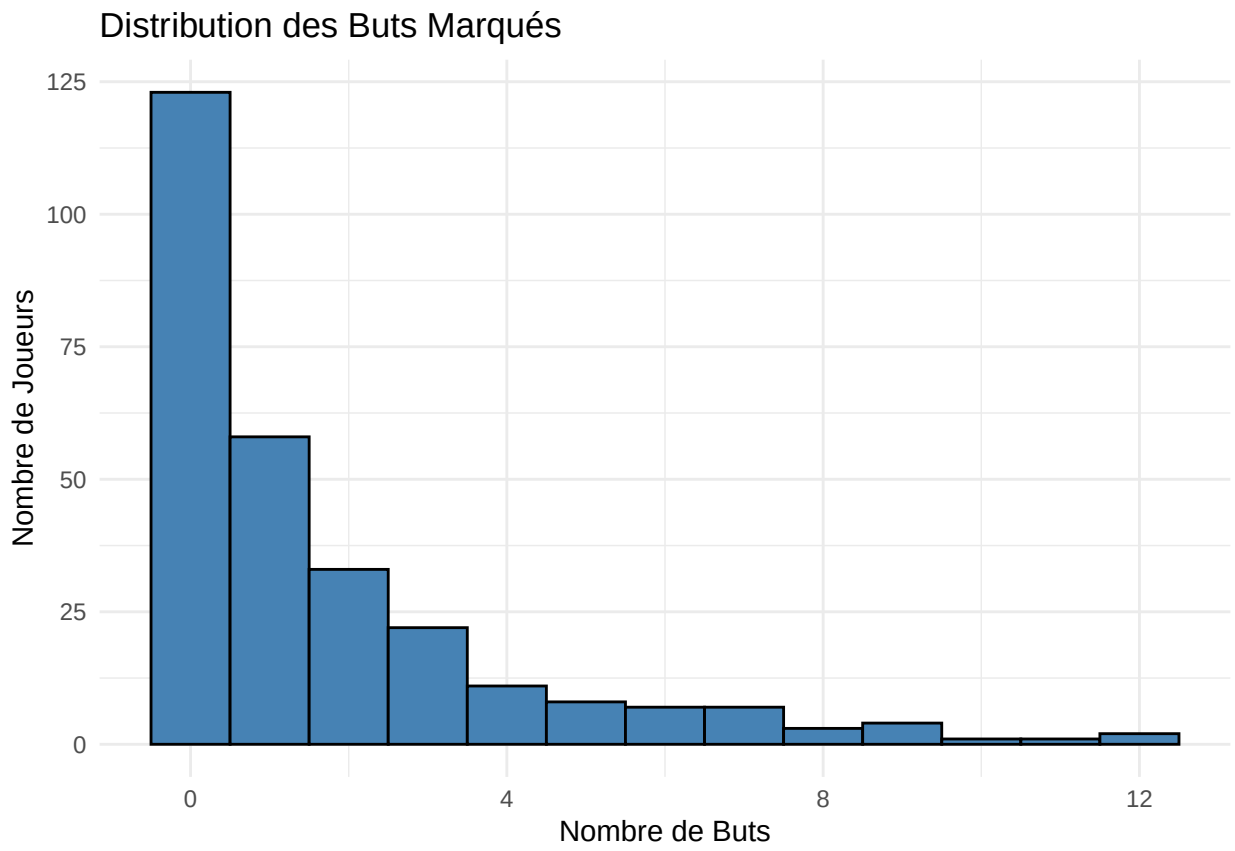
MP : Le nombre de matchs joués varie de 1 à 23, avec une moyenne de 12.33 matchs.

```
ggplot(data, aes(x = Min)) +
  geom_histogram(binwidth = 100, fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution des Minutes Jouées",
       x = "Minutes Jouées",
       y = "Nombre de Joueurs")
```



Min : Les minutes jouées vont de 3 à 2070, avec une moyenne de 799.4 minutes.

```
ggplot(data, aes(x = Goals)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  theme_minimal() +
  labs(title = "Distribution des Buts Marqués",
       x = "Nombre de Buts",
       y = "Nombre de Joueurs")
```

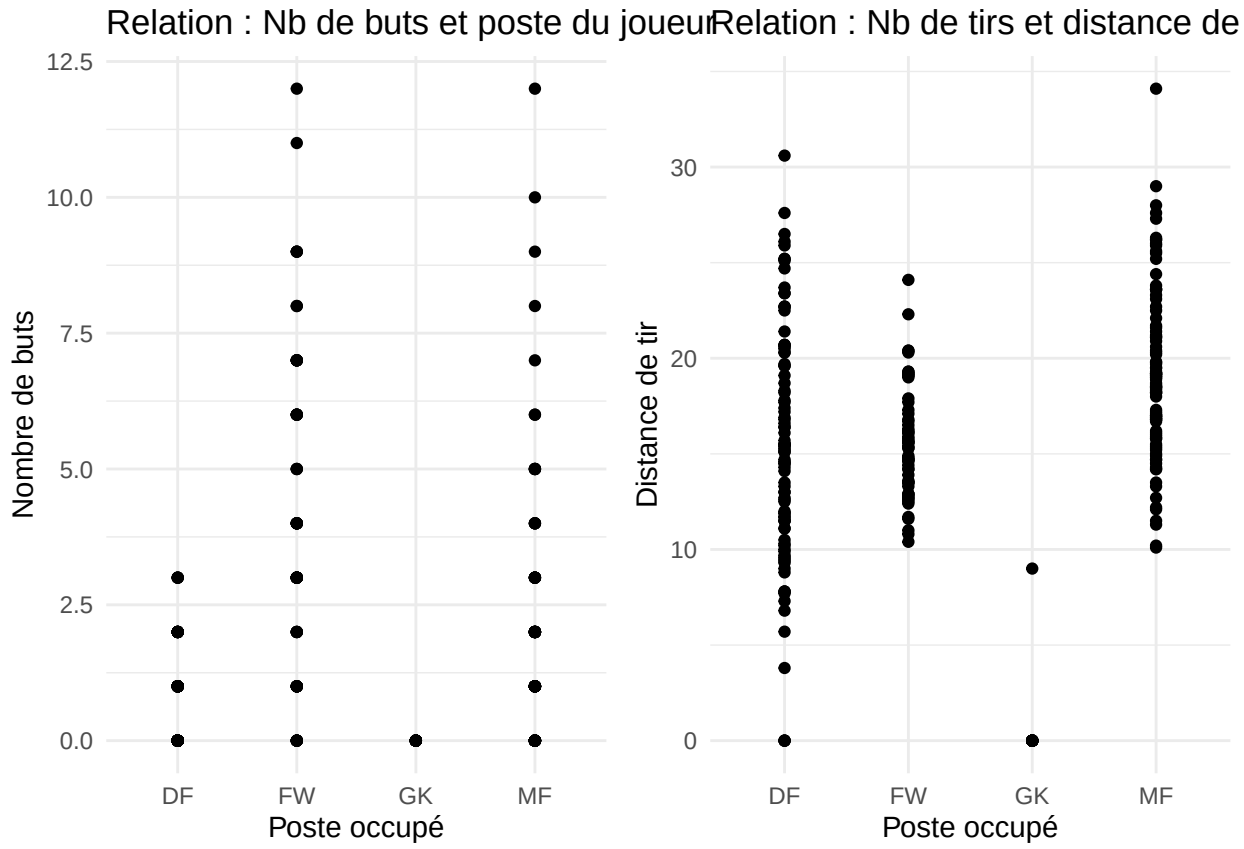


Goals : Le nombre de buts marqués varie de 0 à 25, avec une moyenne de 1.453 buts.

III) Statistique descriptive bivariées

```
library(ggplot2)
# 2. Relation entre le nombre de buts et le poste du joueur
scarplot1 <- ggplot(data = data, aes(x = Pos, y = Goals)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Nb de buts et poste du joueur",
       x = "Poste occupé",
       y = "Nombre de buts") +
  theme_minimal()

# 3. Relation entre le nombre de tirs et la distance de tir par poste
scarplot2 <- ggplot(data = data, aes(x = Pos, y = ShoDist)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Nb de tirs et distance de tir par poste",
       x = "Poste occupé",
       y = "Distance de tir") +
  theme_minimal()

grid.arrange(scarplot1, scarplot2, ncol = 2)
```

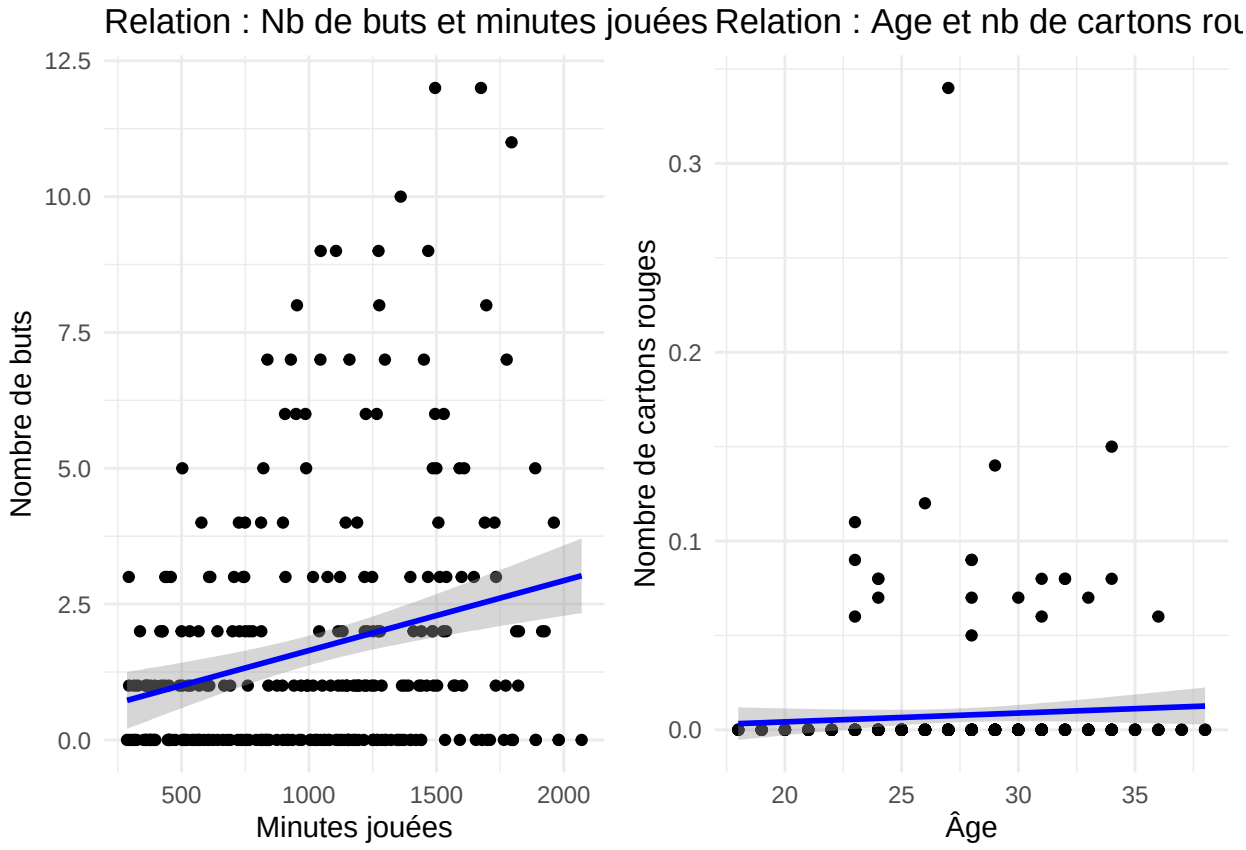
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
library(gridExtra)
# 4. Relation entre le nombre de buts et les minutes jouées
scarplot3 <- ggplot(data = data, aes(x = Min, y = Goals)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Nb de buts et minutes jouées",
       x = "Minutes jouées",
       y = "Nombre de buts") +
  theme_minimal()

# 6. Relation entre l'âge et le nombre de cartons rouges
scarplot4 <- ggplot(data = data, aes(x = Age, y = CrdR)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Age et nb de cartons rouges",
       x = "Âge",
       y = "Nombre de cartons rouges") +
  theme_minimal()

grid.arrange(scarplot3, scarplot4, ncol = 2)
```

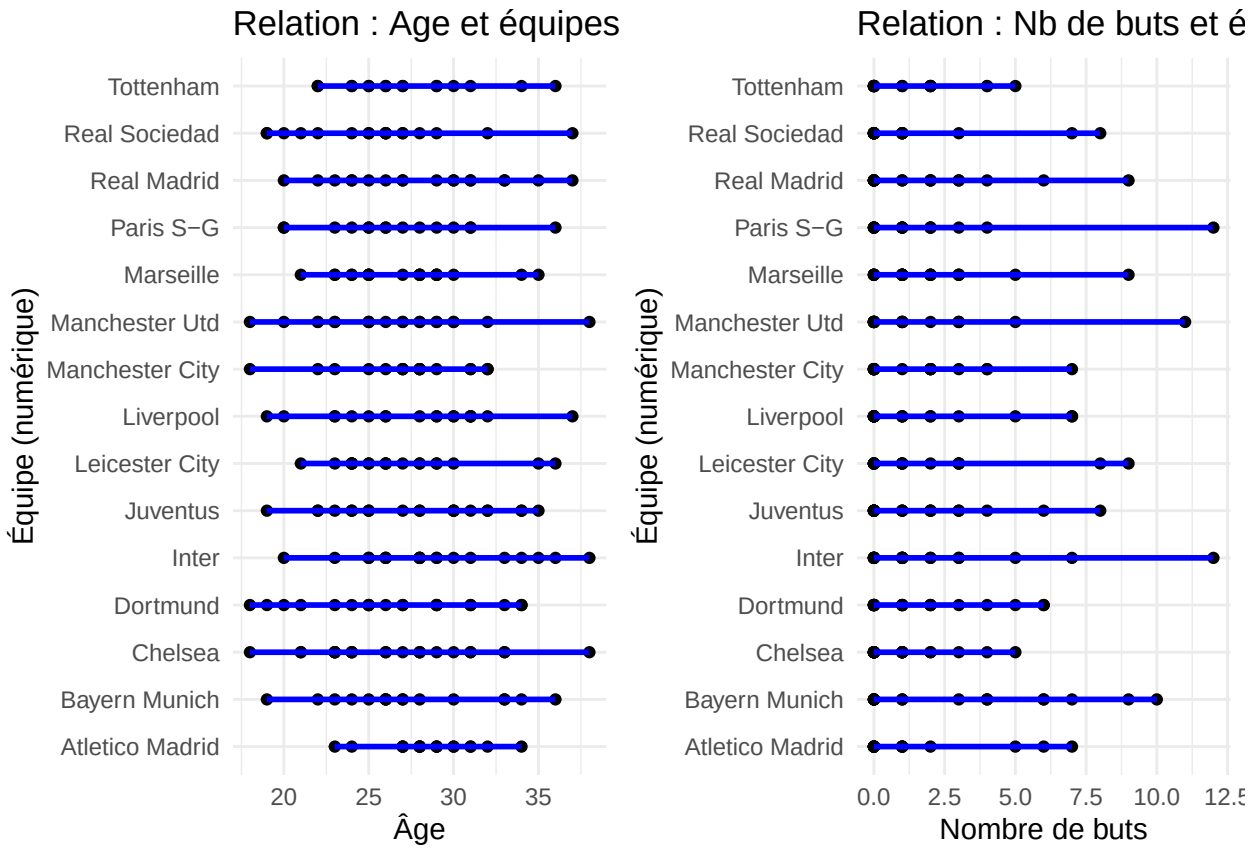
```
## `geom_smooth()` using formula = 'y ~ x'
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# 7. Relation entre l'âge et les équipes
scarplot7 <- ggplot(data = data, aes(x = Age, y = Squad)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Age et équipes",
       x = "Âge",
       y = "Équipe (numérique)") +
  theme_minimal()

# 8. Relation entre le nombre de buts et les équipes
scarplot8 <- ggplot(data = data, aes(x = Goals, y = Squad)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation : Nb de buts et équipes",
       x = "Nombre de buts",
       y = "Équipe (numérique)") +
  theme_minimal()

grid.arrange(scarplot7, scarplot8, ncol = 2)
```

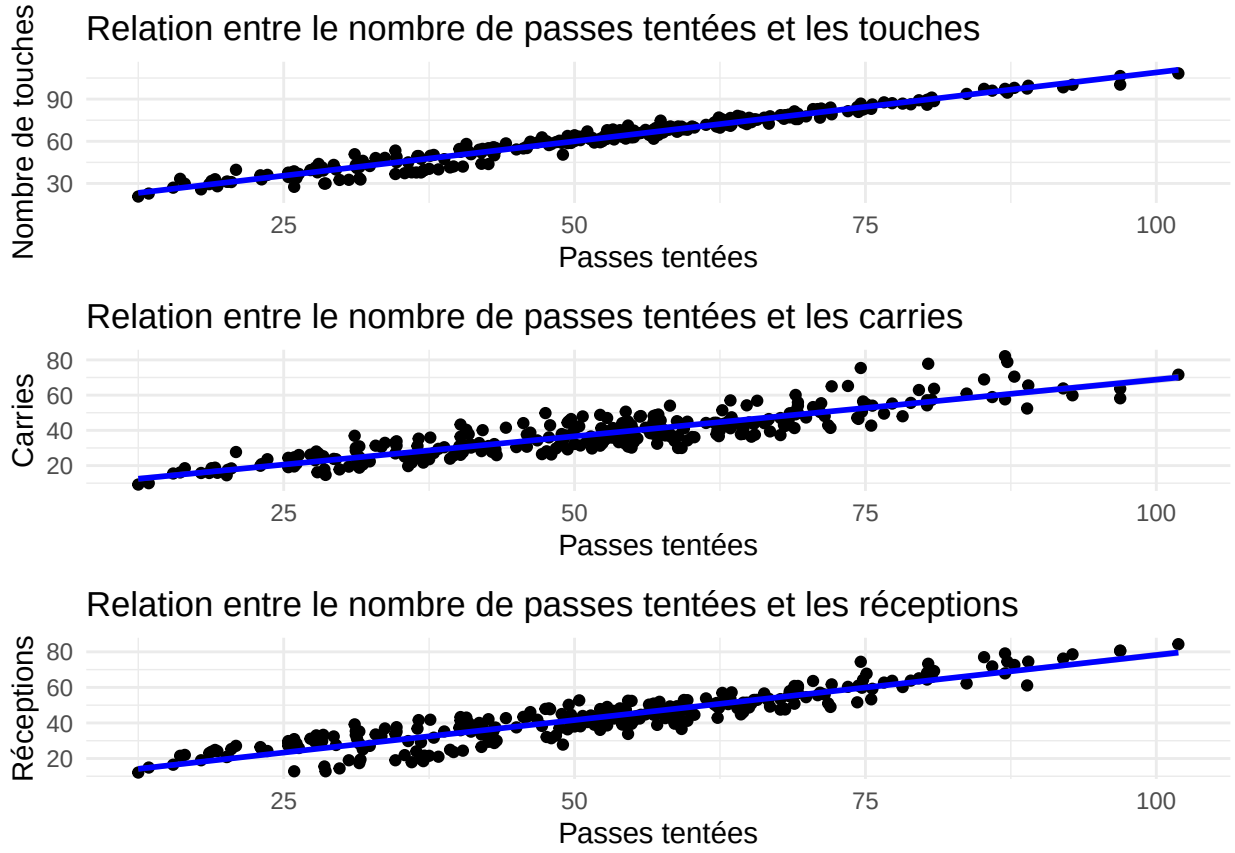
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# 9. Relation entre le nombre de passes tentées et les touches
scarplot9 <- ggplot(data = data, aes(x = PasAtt, y = Touches)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation entre le nombre de passes tentées et les touches",
       x = "Passes tentées",
       y = "Nombre de touches") +
  theme_minimal()

# 10. Relation entre le nombre de passes tentées et les carries
scarplot10 <- ggplot(data = data, aes(x = PasAtt, y = Carries)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation entre le nombre de passes tentées et les carries",
       x = "Passes tentées",
       y = "Carries") +
  theme_minimal()

# 11. Relation entre le nombre de passes tentées et les réceptions
scarplot11 <- ggplot(data = data, aes(x = PasAtt, y = Rec)) +
  geom_point() +
  geom_smooth(method = "lm", col = "blue") +
  labs(title = "Relation entre le nombre de passes tentées et les réceptions",
       x = "Passes tentées",
       y = "Réceptions") +
  theme_minimal()
```

```
grid.arrange(scarplot9, scarplot10, scarplot11, ncol =1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



IV) Analyse des corrélations entre les variables

Nous commençons par établir une matrice de corrélation entre l'ensemble des variables quantitatives.

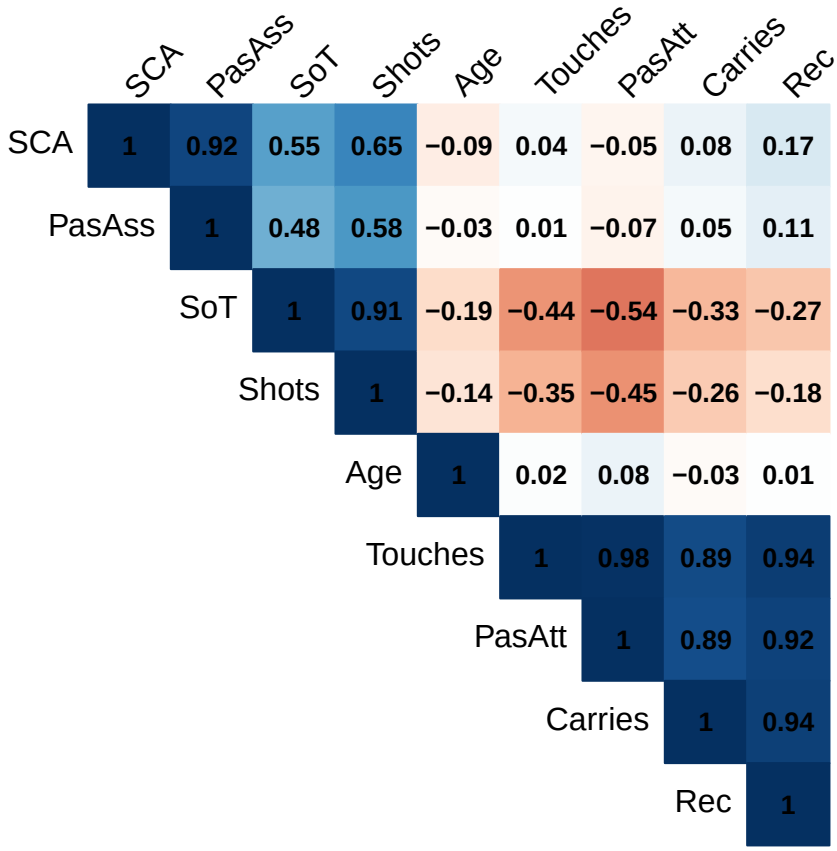
Les couleurs de cette matrice indiquent la force et la direction des corrélations : plus c'est bleu, plus la corrélation est positive ; plus c'est rouge, plus la corrélation est négative.

Par exemple, nous pouvons observer sur le graphique que plus le nombre total de tirs (Shots) est élevé, plus le nombre et le pourcentage de tentatives de passe sont faibles (PasTotDist, PasShoCmp, PasMedCmp, PasAtt, Touches). Cela peut être dû au poste occupé par le joueur, en effet, un attaquant est plus souvent amené à tirer qu'à faire des passes.

De plus, nous constatons que plus le nombre total de tirs (Shots) est élevé, plus le nombre de tirs cadrés est élevé.

```
quanti <- data[c(4, 6:36)]

# Calculer la matrice de corrélation
matrice_correlation <- cor(quanti)
```

```r
variables_interessantes <- c("Age", "SoT", "Shots", "SCA", "Touches", "Carries", "Rec", "PasAss", "PasA
data_subset <- data %>% select(all_of(variables_interessantes))

# Calculer la matrice de corrélation pour le sous-ensemble
matrice_correlation_subset <- cor(data_subset, use = "complete.obs")

# Créer la heatmap de corrélation avec des paramètres ajustés
corrplot(matrice_correlation_subset, method = "color", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", # Couleur des coefficients
         cl.pos = "n", # Position de la légende de couleur
         cl.cex = 1.2, # Taille de la légende de couleur
         number.cex = 0.8) # Taille des chiffres des coefficients
```



Etape 2 : Analyse en composantes principales (ACP)

I) Centrer et réduire les données

Dans une ACP, l'effet de taille peut influencer les résultats. Pour éviter cet effet, il est préférable de centrer et réduire les données. Cela consiste à soustraire la moyenne de chaque variable et à diviser par l'écart-type. Cette transformation permet d'harmoniser les échelles des variables, de s'assurer qu'elles ont des variances similaires et de garantir que les unités de mesure sont cohérentes.

```r
# Centrer et réduire les données
donnees_centrees_reduites <- scale(quanti,center = TRUE,scale=TRUE)
```

II) Choix du nombre d'axe factoriel

On crée l'ACP sur nos données centrées réduites avec comme paramètre de départ 32 composantes principales à conserver.

```r
quanti.acp <- PCA(donnees_centrees_reduites, scale.unit = TRUE, ncp = 32, graph = F)
```

L'objectif principal d'une analyse en composantes principales (ACP) est de réduire la dimensionnalité des données. Pour ce faire, nous prenons les variables de départ et créons des composantes principales, qui sont des combinaisons linéaires de ces variables. L'enjeu est de conserver le maximum d'information possible. Pour y parvenir, nous analysons la table des valeurs propres. Cette table fournit, pour chaque composante, la valeur propre, le pourcentage de la variance expliquée et le pourcentage cumulé de la variance expliquée.

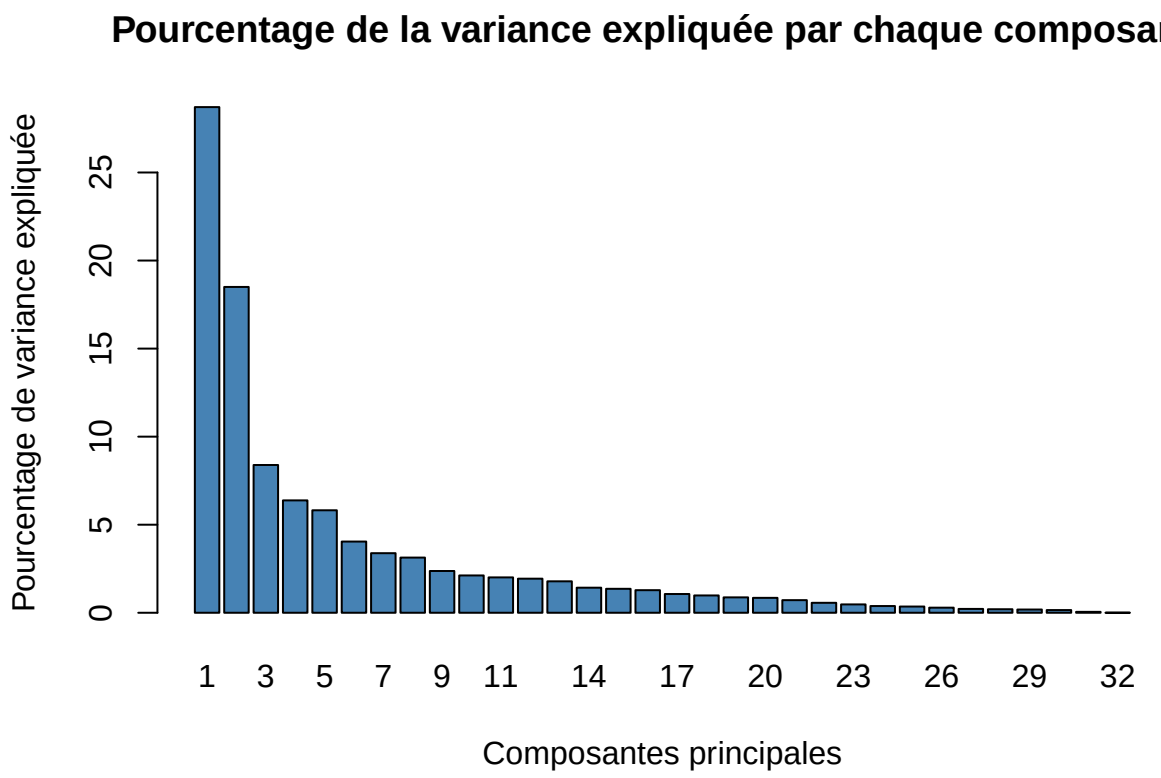
La première approche utilisée est la technique de la variance expliquée. Je choisis le nombre d'axes qui expliquent au moins 65% de la variabilité. D'après la table des valeurs propres, c'est à partir de la 5e composante que nous atteignons ce seuil de 65%.

```r
valeurspropres <- quanti.acp$eig
knitr::kable(data.frame(quanti.acp$eig))
```

	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
comp 1	9.1881840	28.7130750	28.71307
comp 2	5.9200098	18.5000306	47.21311
comp 3	2.6848244	8.3900763	55.60318
comp 4	2.0416976	6.3803049	61.98349
comp 5	1.8621577	5.8192427	67.80273
comp 6	1.2939171	4.0434908	71.84622
comp 7	1.0824410	3.3826281	75.22885
comp 8	1.0025682	3.1330256	78.36187
comp 9	0.7588401	2.3713752	80.73325
comp 10	0.6777239	2.1178871	82.85114
comp 11	0.6425191	2.0078722	84.85901
comp 12	0.6192260	1.9350813	86.79409
comp 13	0.5712415	1.7851297	88.57922
comp 14	0.4555732	1.4236663	90.00289
comp 15	0.4352731	1.3602284	91.36311
comp 16	0.4104004	1.2825011	92.64562
comp 17	0.3407023	1.0646948	93.71031
comp 18	0.3146984	0.9834326	94.69374
comp 19	0.2795974	0.8737420	95.56748
comp 20	0.2706313	0.8457228	96.41321
comp 21	0.2291235	0.7160109	97.12922
comp 22	0.1806627	0.5645711	97.69379
comp 23	0.1516716	0.4739739	98.16776
comp 24	0.1220414	0.3813794	98.54914
comp 25	0.1130367	0.3532398	98.90238
comp 26	0.0925714	0.2892856	99.19167
comp 27	0.0695931	0.2174784	99.40915

	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
comp 28	0.0641312	0.2004101	99.60956
comp 29	0.0587777	0.1836804	99.79324
comp 30	0.0487073	0.1522103	99.94545
comp 31	0.0164130	0.0512907	99.99674
comp 32	0.0010439	0.0032621	100.00000

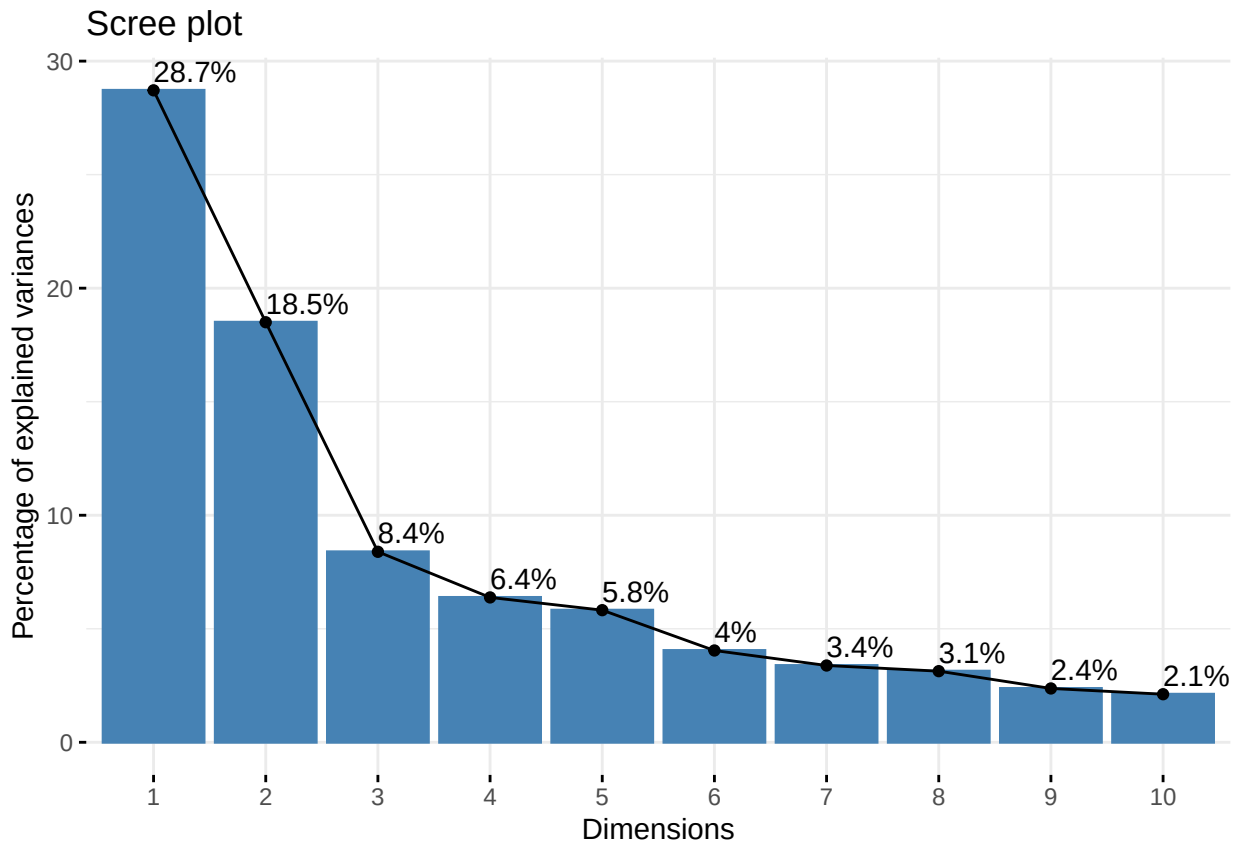
```
barplot(valeurspropres[, 2], names.arg=1:nrow(valeurspropres),
       main = "Pourcentage de la variance expliquée par chaque composante",
       xlab = "Composantes principales",
       ylab = "Pourcentage de variance expliquée",
       col ="steelblue")
```



La seconde approche est la règle du coude. Cette méthode consiste à choisir le nombre d'axes factoriels correspondant à une décroissance significative des valeurs propres. Sur le graphique, je constate que ce décrochage se situe entre la 5e et la 6e composante.

En se référant à ces deux approches, j'ai choisi de retenir 5 axes factoriels pour notre analyse.

```
# Créer le graphique des valeurs propres
fviz_eig(quanti.acp, addlabels = TRUE)
```



III) Représentation du cercle de corrélation

Axe 1

A) Contribution des variables

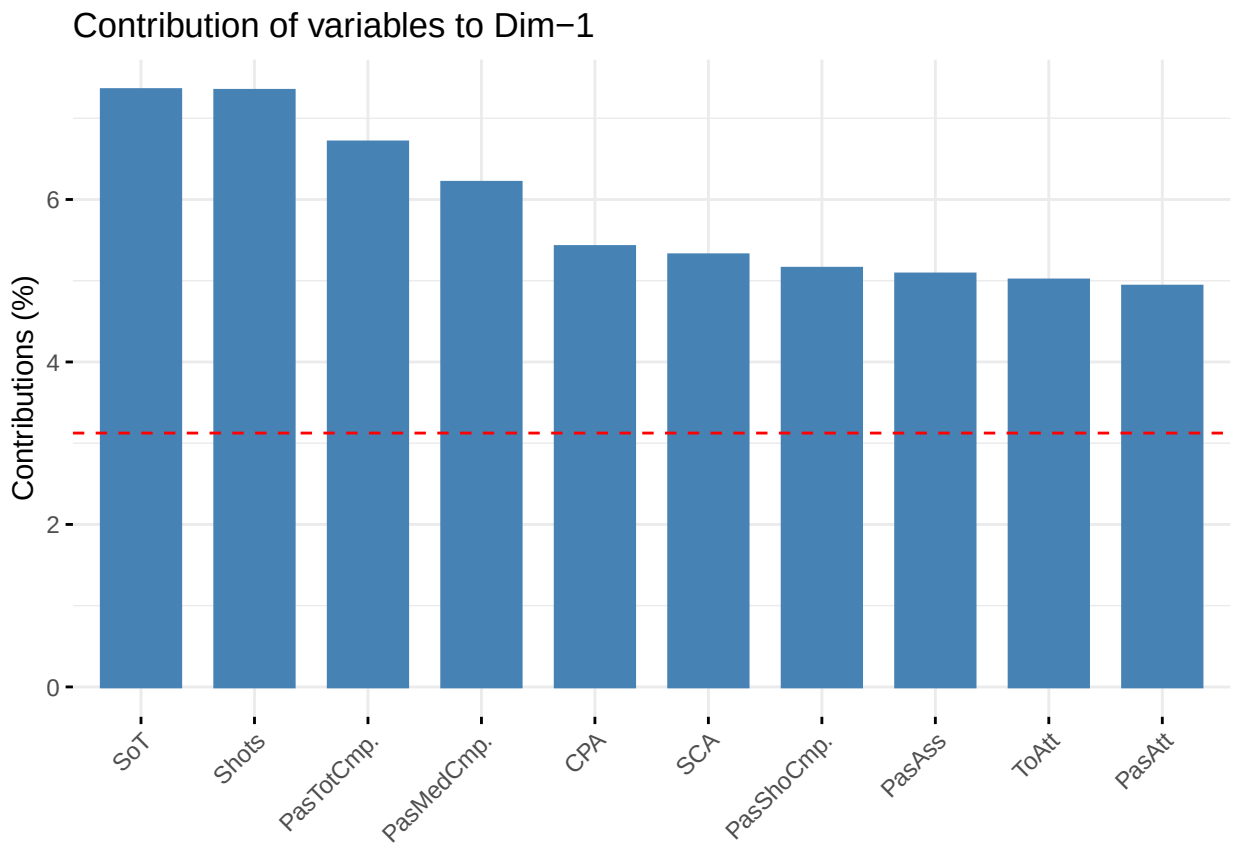
Lorsqu'on observe le cercle des corrélation des variables, plus une variable est proche du cercle, mieux elle est représentée. Nous analyserons uniquement les variables bien représentées à chaque fois.

Toutes les variables affichées en rouge sont bien représentées selon les dimensions 1 et 2.

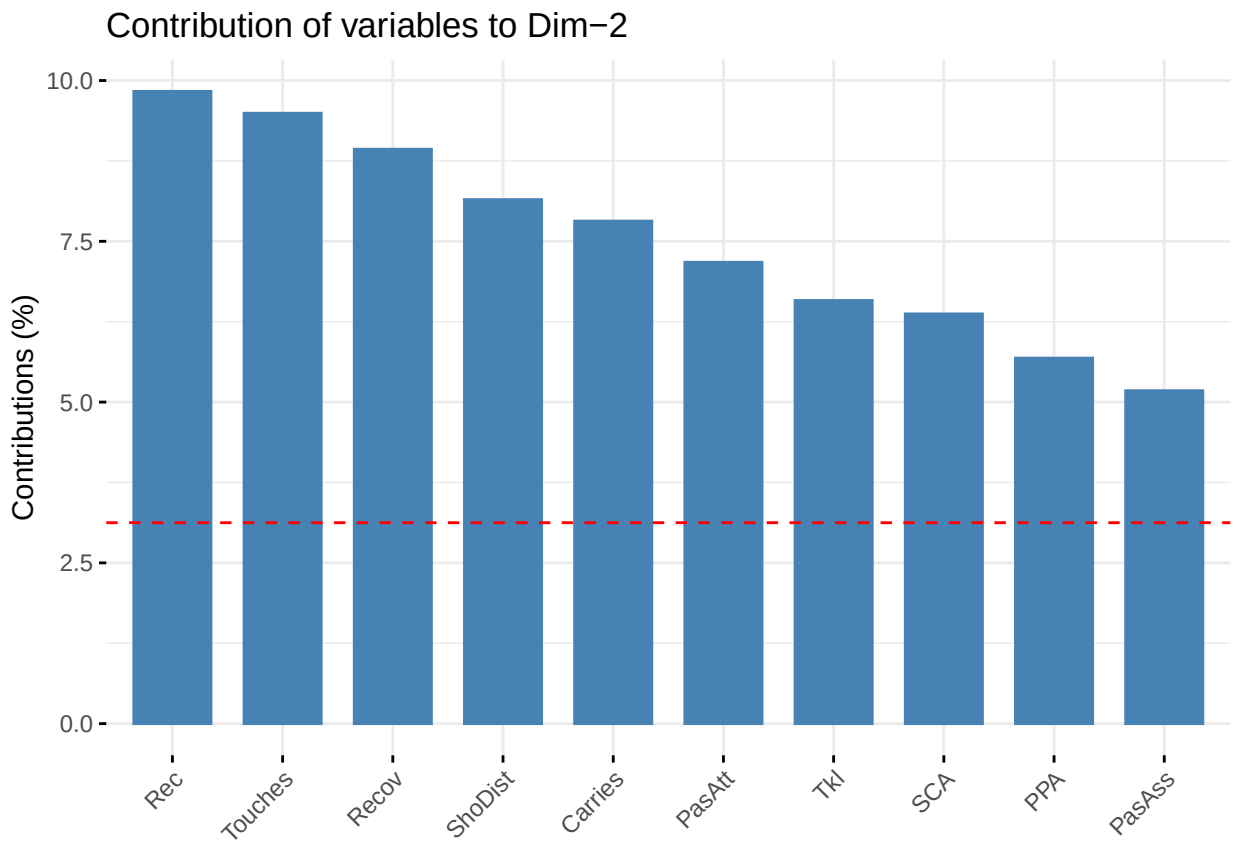
Les variables en bleu ne sont bien représentées par aucune des deux dimensions.

Les variables en orange sont bien représentées par une seule des deux dimensions. Par exemple, la variable PasToCmp est bien représentée selon la première dimension, mais pas selon la seconde. À l'inverse, Rec est bien représentée selon la seconde dimension, mais pas selon la première.

```
fviz_contrib(quanti.acp, choice = "var", axes = 1, top = 10)
```




```
fviz_contrib(quanti.acp, choice = "var", axes = 2, top = 10)
```



B) Corrélations entre les variables.

Le cercle de corrélation permet aussi d'analyser les relations entre les variables :

Variables proches l'une de l'autre : Les variables situées près l'une de l'autre sur le cercle sont fortement corrélées positivement. Cela signifie qu'elles varient de manière similaire. Par exemple, la variable Shots est proche de la variable Goals, cela indique que plus un joueur tire, plus il a de chances de marquer.

Variables opposées : Les variables situées à l'opposé l'une de l'autre sur le cercle sont fortement corrélées négativement. Cela signifie que lorsque l'une augmente, l'autre tend à diminuer. Par exemple, PasToCmp est opposée à Shots, cela suggère que les joueurs qui tentent plus de passes tirent moins. On peut penser que ceux qu'il y a les milieu de terrains d'une part et les attaquant d'une autres.

Variables orthogonales : Les variables situées à angle droit l'une de l'autre ne sont pas corrélées. Elles varient indépendamment l'une de l'autre. Par exemple, si Rec est perpendiculaire à PPA, cela signifie qu'il n'y a pas de relation directe entre le nombre de passes reçues et le nombre de passes réussies dans la surface de réparation.

C) Identification des groupes de joueurs.

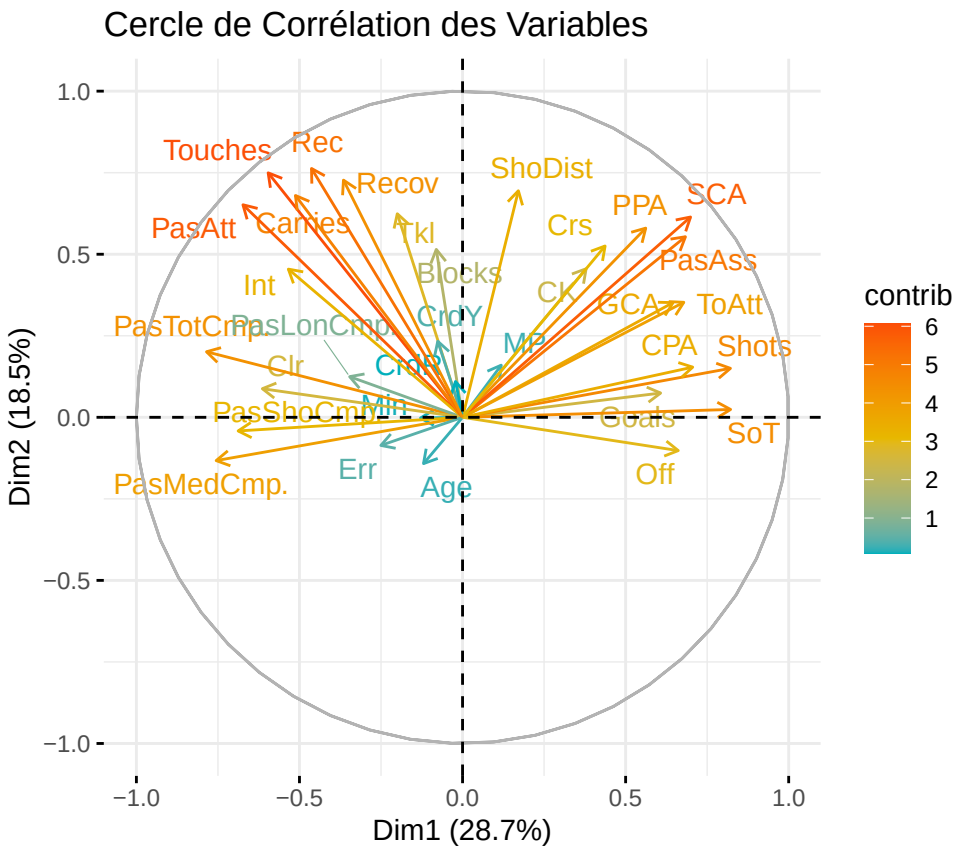
Sur le cercle, on peut déjà commencer à repérer différents groupes de joueurs :

En haut à droite du cercle : Les joueurs situés dans cette zone sont ceux qui jouent près du but adverse. Ils sont souvent impliqués dans des actions offensives proches du but, comme les attaquants ou les milieux offensifs.

En haut à gauche du cercle : Les joueurs situés dans cette zone sont ceux qui dribblent fréquemment. Ils sont souvent impliqués dans des actions de débordement et de création d'opportunités, comme les ailiers et certains milieux de terrain créatifs.

En observant ces groupes, nous pouvons mieux comprendre les rôles et les comportements des joueurs sur le terrain en fonction de leur position sur le cercle de corrélation.

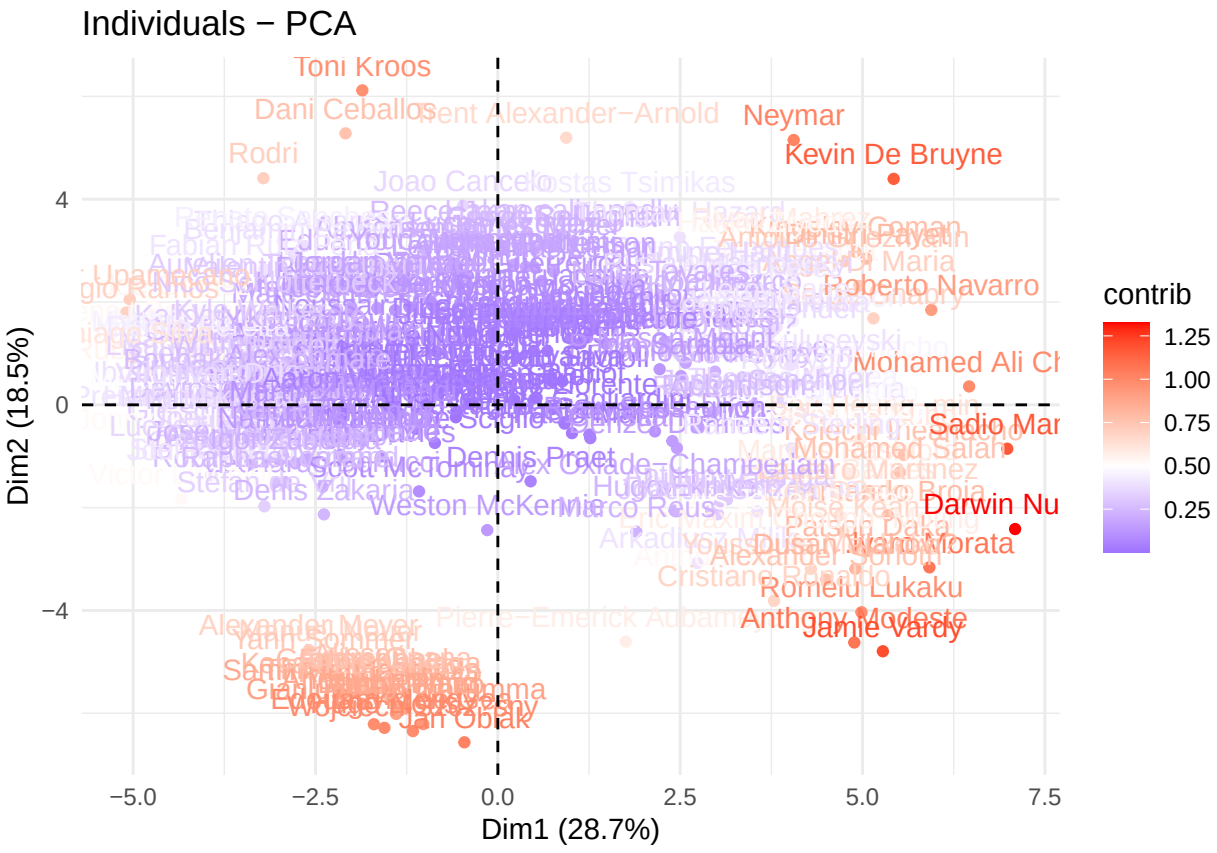
```
fviz_pca_var(quanti.acp,
             col.var = "contrib", # Utiliser la qualité de contribution (contrib) pour la couleur
             axes = c(1,2),
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), # Palette de couleurs
             repel = TRUE, # Éviter le chevauchement des étiquettes
             title = "Cercle de Corrélation des Variables")
```




```
fviz_pca_ind(quanti.acp,  col.ind="contrib") +
scale_color_gradient2(low="blue", mid="white",
                      high="red", midpoint=0.50)+
  theme_minimal()
```

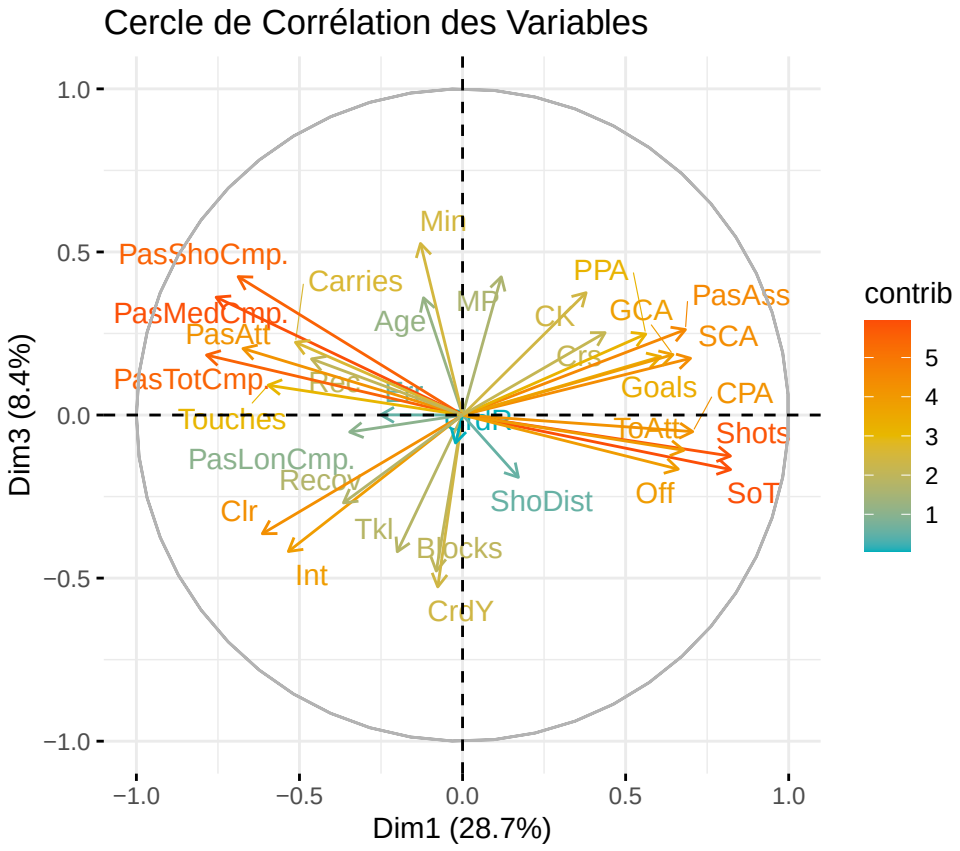
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <e1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Beno<ee>t Badiashile' in 'mbcsToSbcs': dot substituted
## for <ee>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>ngel Correa' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas Hern<e1>ndez' in 'mbcsToSbcs': dot substituted for
## <e1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Jose Mar<ed>a Gimenez' in 'mbcsToSbcs': dot substituted
## for <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Pablo Mar<ed>n' in 'mbcsToSbcs': dot substituted for
## <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>lvaro Morata' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Thomas M<fc>ller' in 'mbcsToSbcs': dot substituted for
## <fc>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas V<e1>zquez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

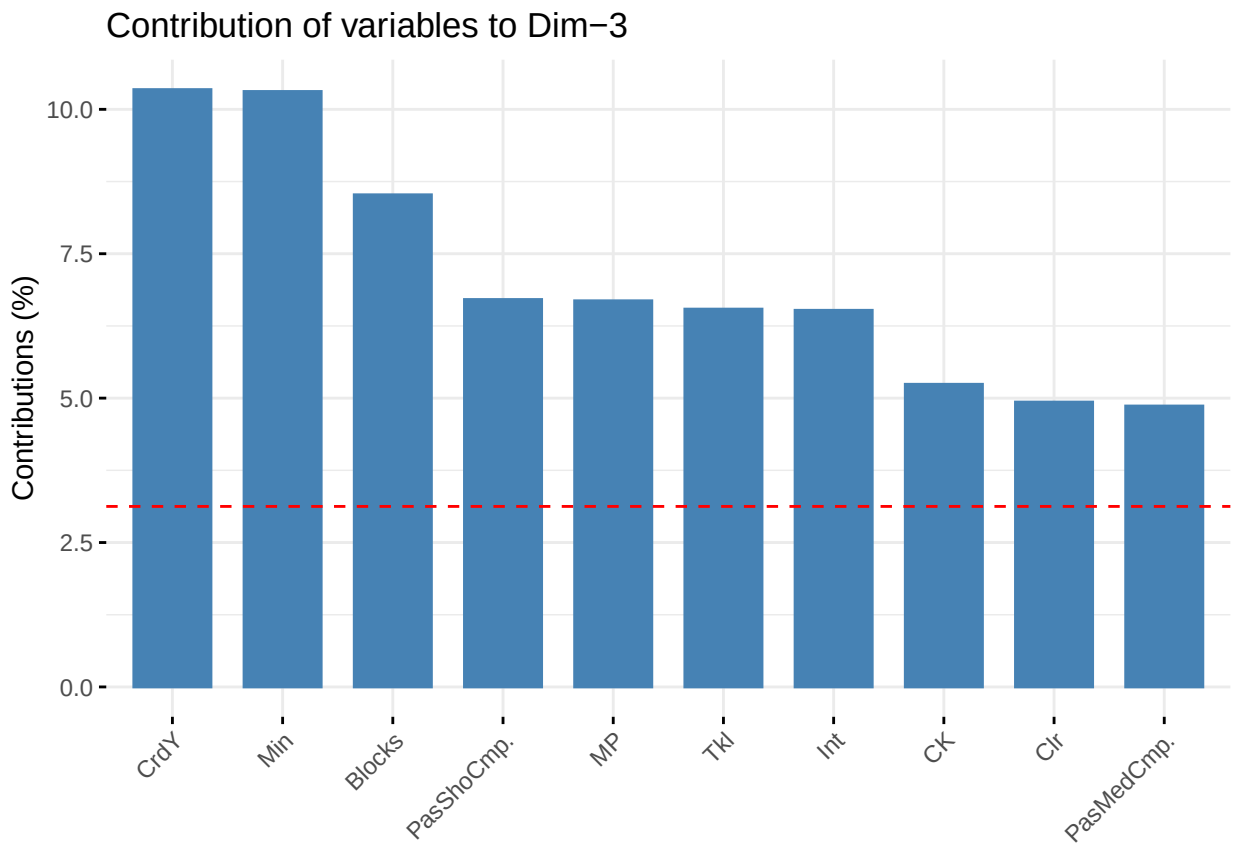


Axe 3

```
fviz_pca_var(quanti.acp,
             col.var = "contrib", # Utiliser la qualité de contribution (contrib) pour la couleur
             axes = c(1,3),
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), # Palette de couleurs
             repel = TRUE, # Éviter le chevauchement des étiquettes
             title = "Cercle de Corrélation des Variables")
```



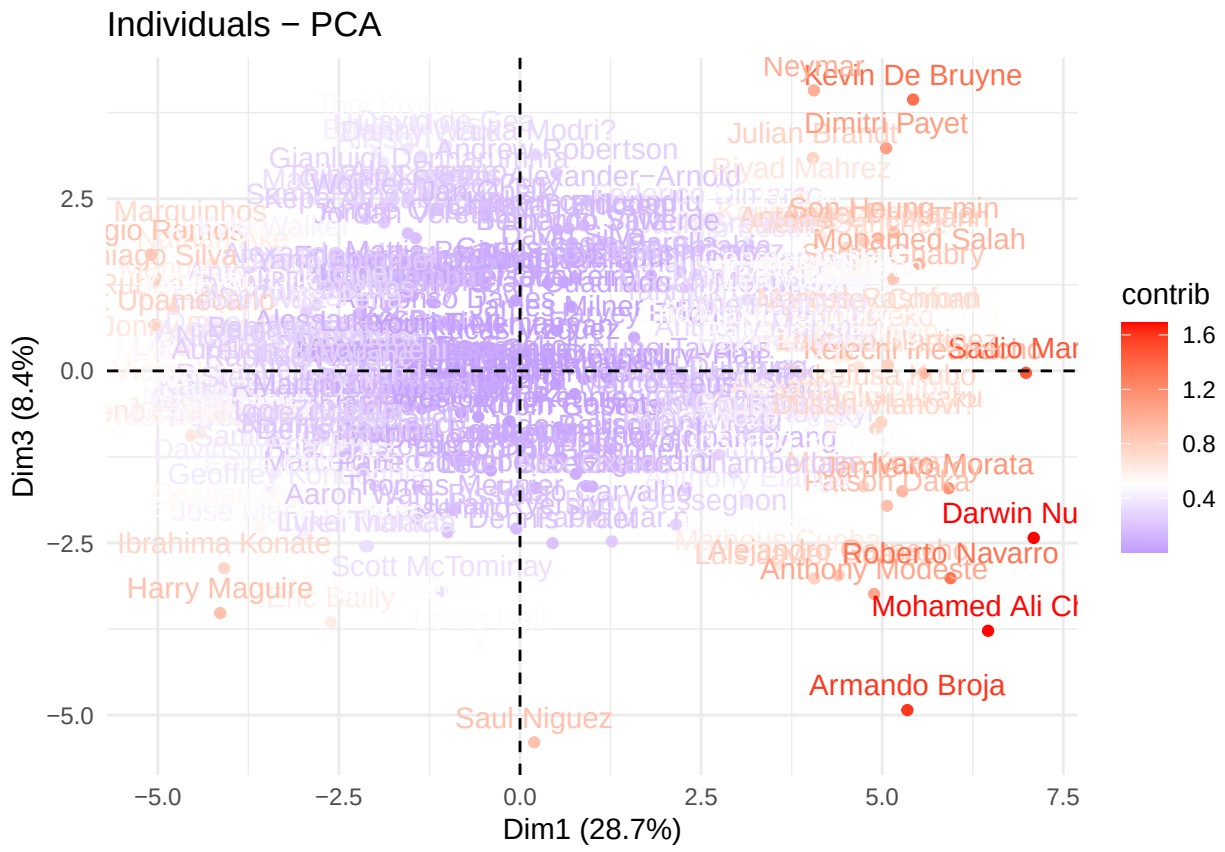

```
fviz_contrib(quanti.acp, choice = "var", axes = 3, top = 10)
```




```
fviz_pca_ind(quanti.acp,
             col.ind = "contrib",
             axes = c(1, 3)) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0.50) +
  theme_minimal()
```

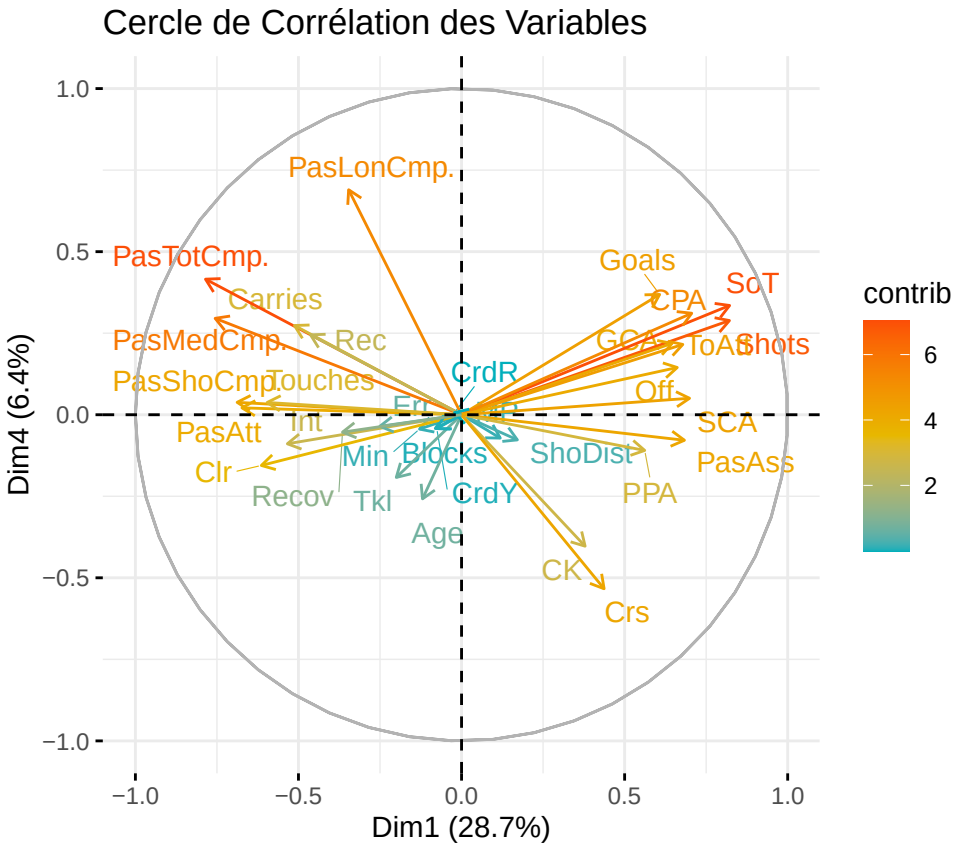
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <e1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Beno<ee>t Badiashile' in 'mbcsToSbcs': dot substituted
## for <ee>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>ngel Correa' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas Hern<e1>ndez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Jose Mar<ed>a Gimenez' in 'mbcsToSbcs': dot substituted
## for <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Pablo Mar<ed>n' in 'mbcsToSbcs': dot substituted for
## <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>lvaro Morata' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Thomas M<fc>ller' in 'mbcsToSbcs': dot substituted for
## <fc>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas V<e1>zquez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

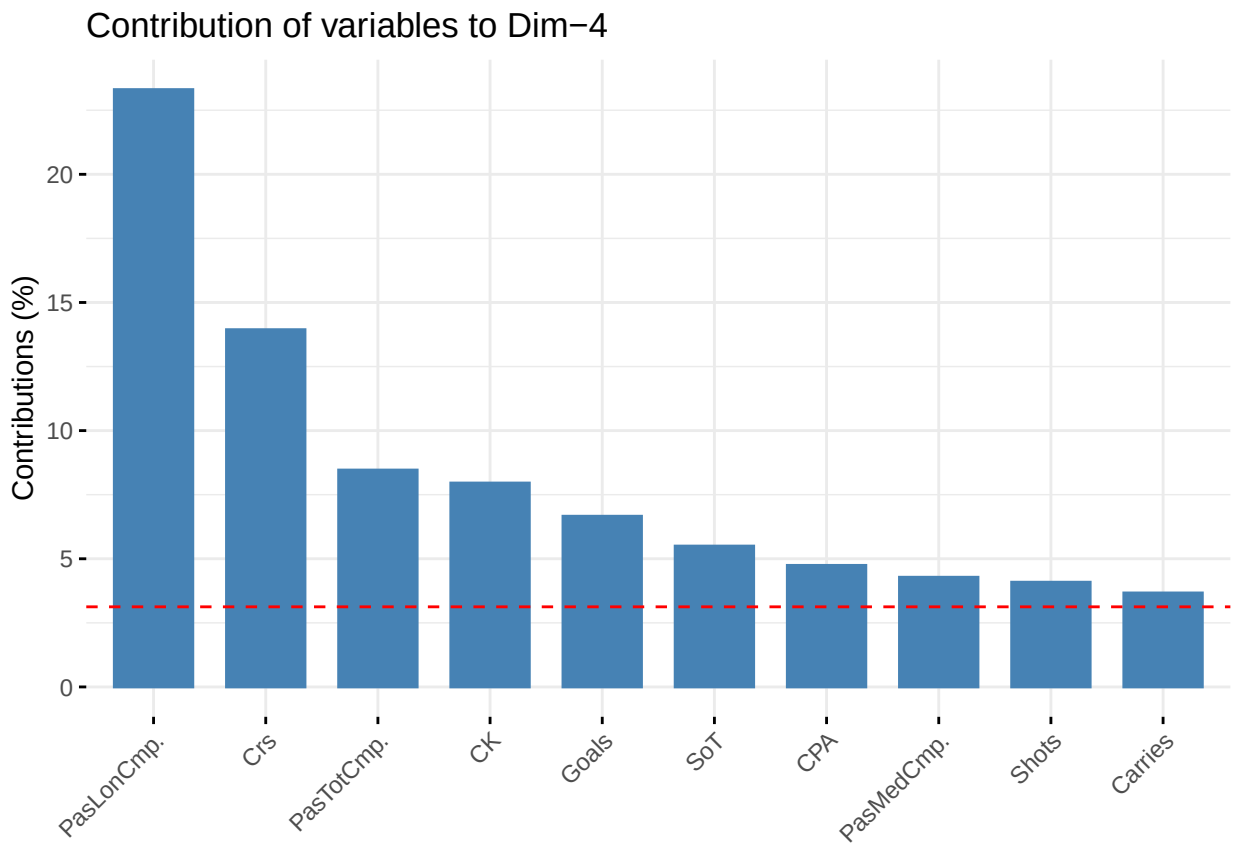


Axe 4

```r
fviz_pca_var(quanti.acp,
             col.var = "contrib", # Utiliser la qualité de contribution (contrib) pour la couleur
             axes = c(1,4),
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), # Palette de couleurs
             repel = TRUE, # Éviter le chevauchement des étiquettes
             title = "Cercle de Corrélation des Variables")
```



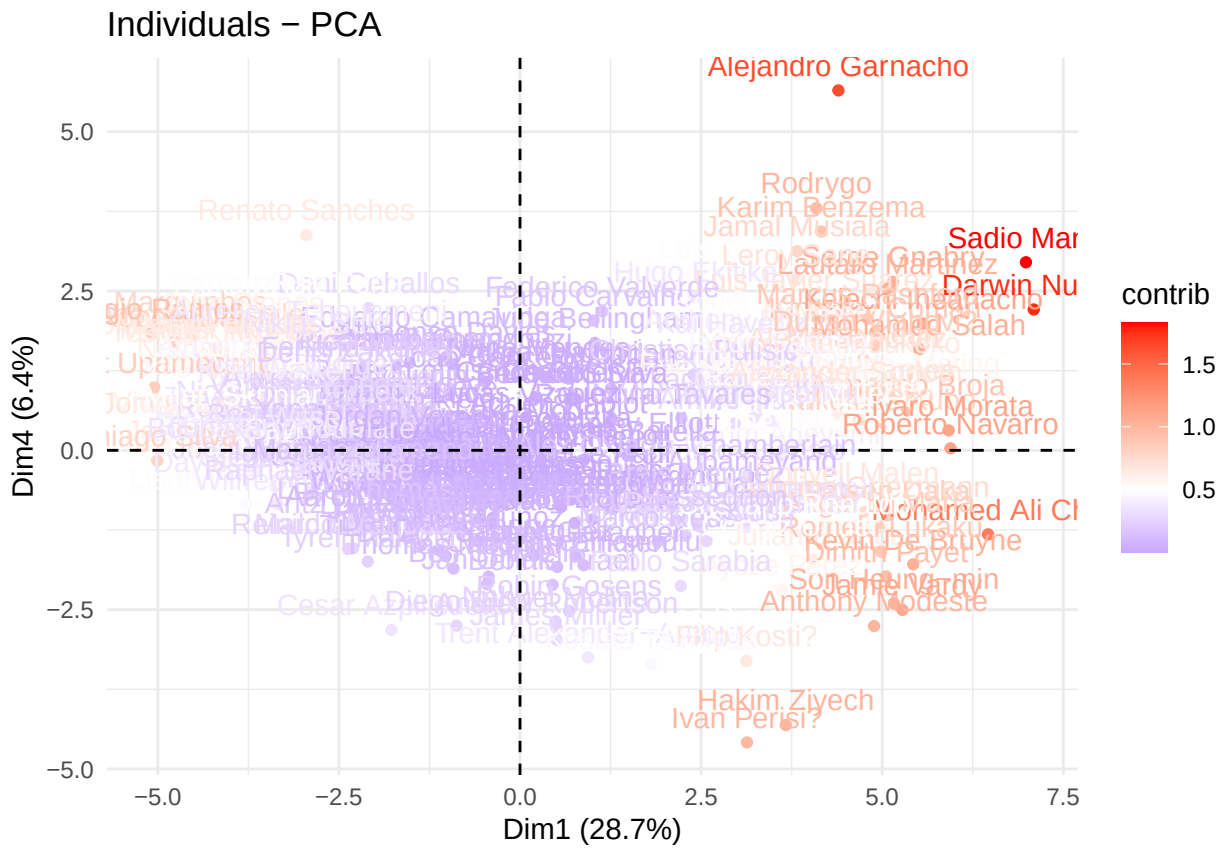

```r
fviz_contrib(quanti.acp, choice = "var", axes = 4, top = 10)
```




```
fviz_pca_ind(quanti.acp,
             col.ind = "contrib",
             axes = c(1, 4)) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0.50) +
  theme_minimal()
```

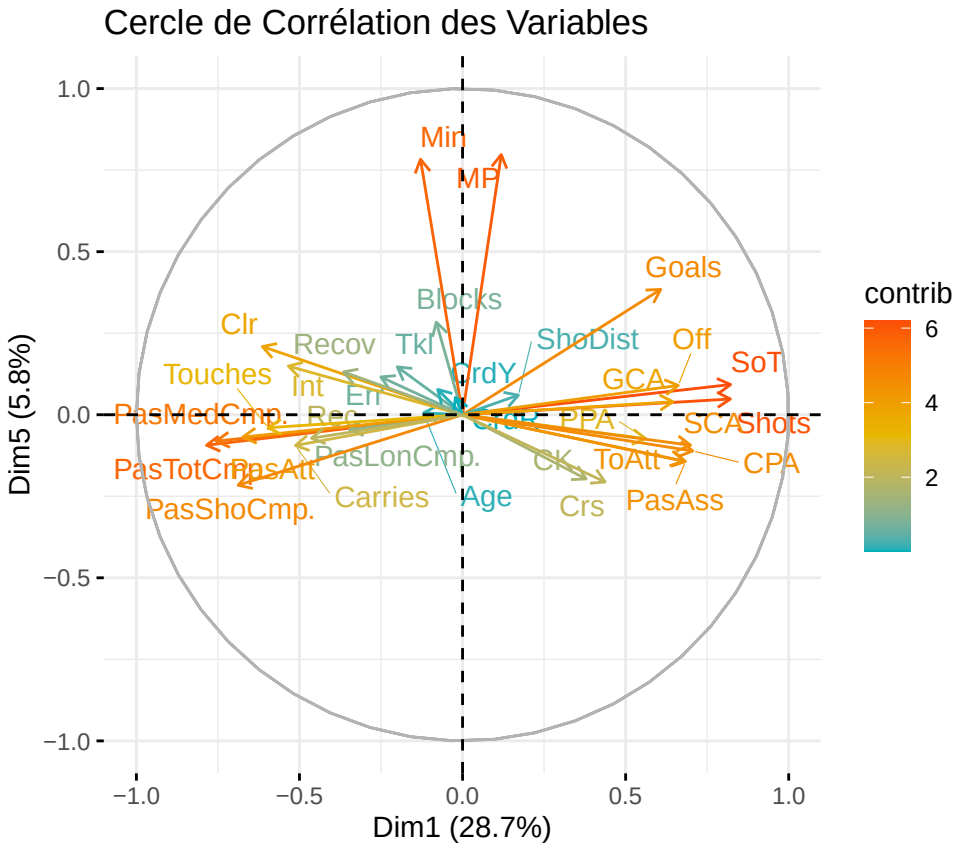
```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <e1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Beno<ee>t Badiashile' in 'mbcsToSbcs': dot substituted
## for <ee>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>ngel Correa' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas Hern<e1>ndez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Jose Mar<ed>a Gimenez' in 'mbcsToSbcs': dot substituted
## for <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Pablo Mar<ed>n' in 'mbcsToSbcs': dot substituted for
## <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>lvaro Morata' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Thomas M<fc>ller' in 'mbcsToSbcs': dot substituted for
## <fc>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas V<e1>zquez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

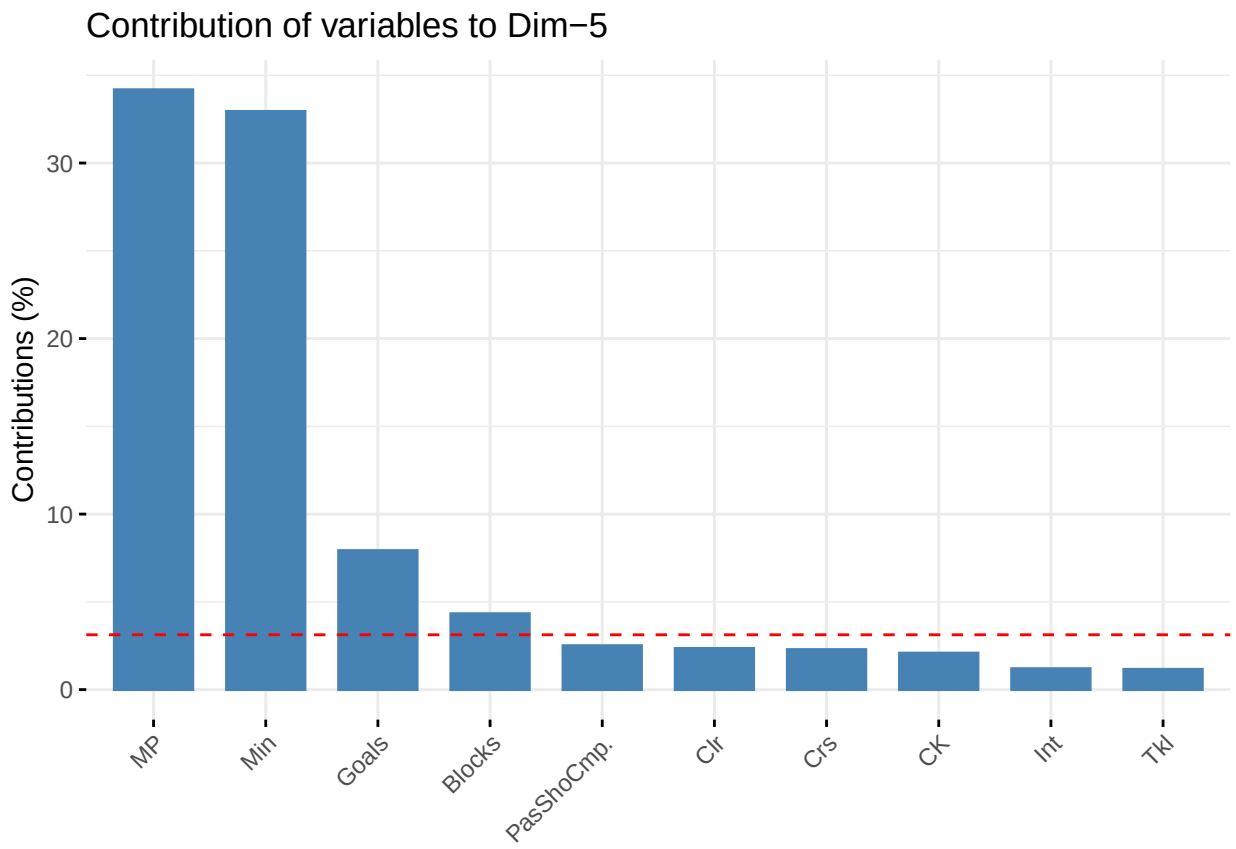


Axe 5

```r
fviz_pca_var(quanti.acp,
             col.var = "contrib", # Utiliser la qualité de contribution (contrib) pour la couleur
             axes = c(1,5),
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), # Palette de couleurs
             repel = TRUE, # Éviter le chevauchement des étiquettes
             title = "Cercle de Corrélation des Variables")
```



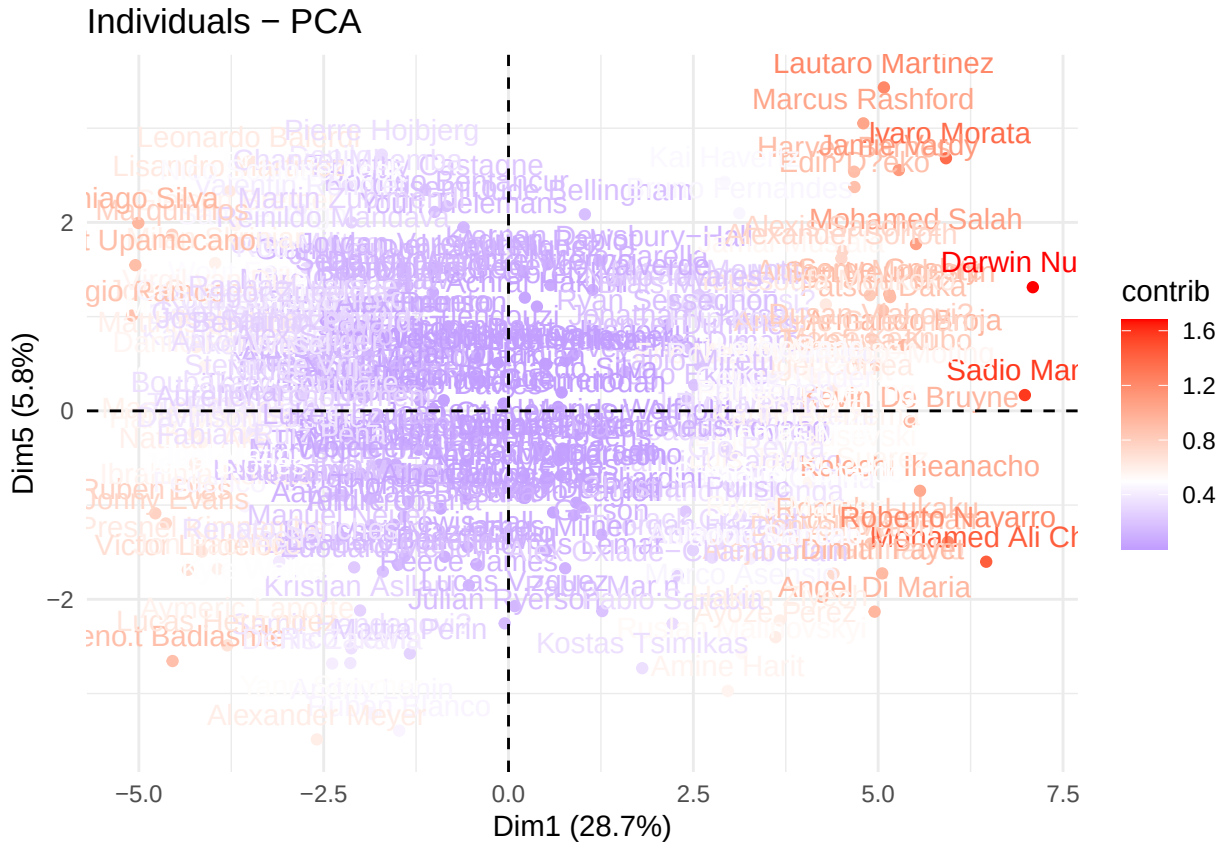

```r
fviz_contrib(quanti.acp, choice = "var", axes = 5, top = 10)
```




```
fviz_pca_ind(quanti.acp,
             col.ind = "contrib",
             axes = c(1, 5)) +
  scale_color_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0.50) +
  theme_minimal()
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <e1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Juli<e1>n <c1>lvarez' in 'mbcsToSbcs': dot substituted
## for <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Beno<ee>t Badiashile' in 'mbcsToSbcs': dot substituted
## for <ee>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>ngel Correa' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas Hern<e1>ndez' in 'mbcsToSbcs': dot substituted for
## <e1>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Jose Mar<ed>a Gimenez' in 'mbcsToSbcs': dot substituted
## for <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Pablo Mar<ed>n' in 'mbcsToSbcs': dot substituted for
## <ed>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '<c1>lvaro Morata' in 'mbcsToSbcs': dot substituted for
## <c1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Thomas M<fc>ller' in 'mbcsToSbcs': dot substituted for
## <fc>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Lucas V<e1>zquez' in 'mbcsToSbcs': dot substituted for
## <e1>
```



Etape 3 : Classification mixte

I) Partitionnement initial

Pour cette étape, nous avons d'abord calculé la taille de l'échantillon (n) et choisi la taille des sous-échantillons (q). Ensuite, nous avons effectué un partitionnement des données réduites et centrées à l'aide de l'algorithme K-means avec 18 centres et 10 répétitions de départ. En utilisant ces résultats, nous avons construit un dendrogramme basé sur une classification ascendante hiérarchique (CAH) avec la méthode de Ward. Ce dendrogramme a été visualisé pour une meilleure compréhension de la structure des données.

```
n = nrow(donnees_centrees_reduites)
n
```

```
## [1] 280
```

```
q = floor(n/10)
q
```

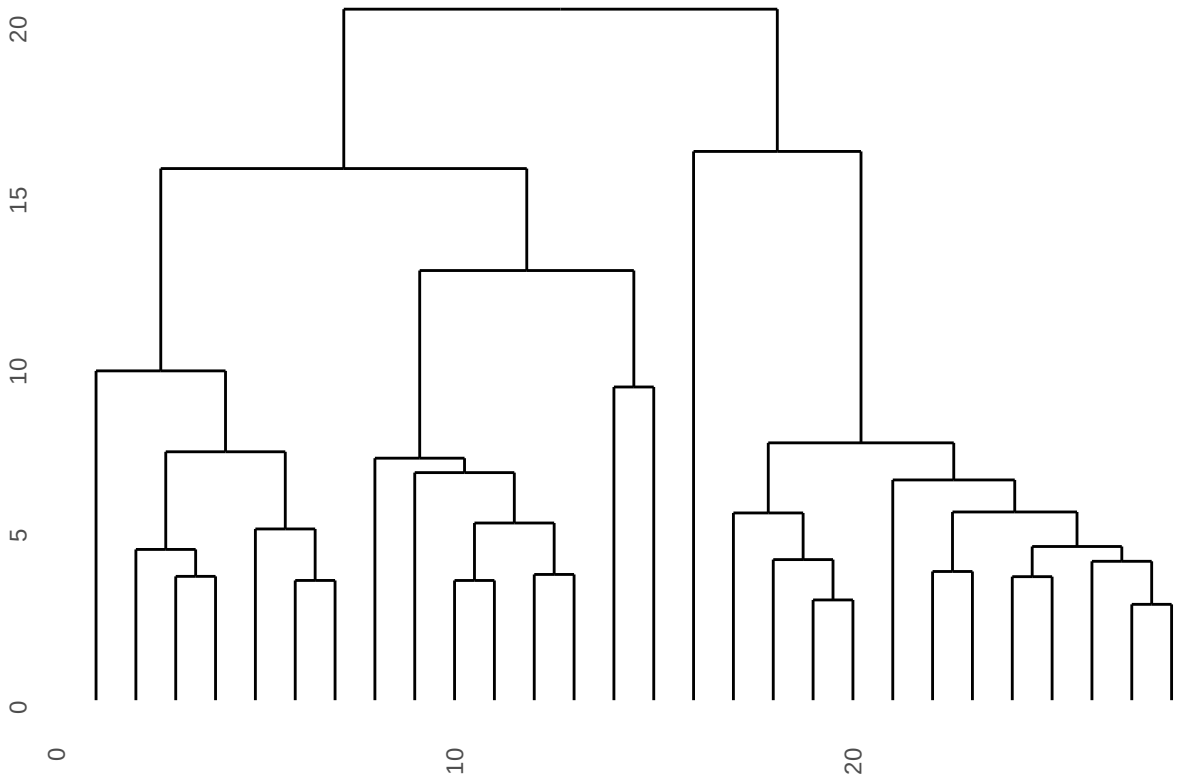
```
## [1] 28
```

```
partition = donnees_centrees_reduites %>% kmeans(centers = 28, nstart = 10)
```

```
donnees = partition$centers
```

```
dendro = donnees %>% dist %>% hclust(method = "ward.D2")
```

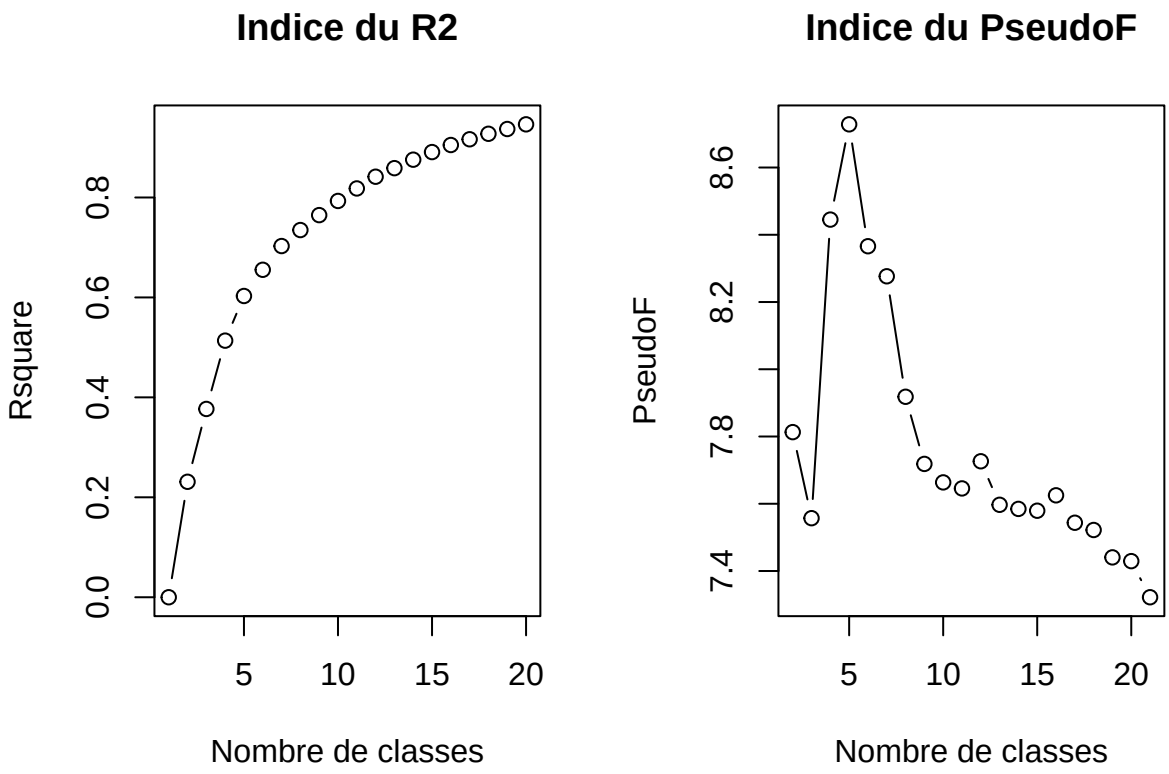
```
dendro %>% ggdendrogram(labels = FALSE)
```



II) Pseudo F-Statistics

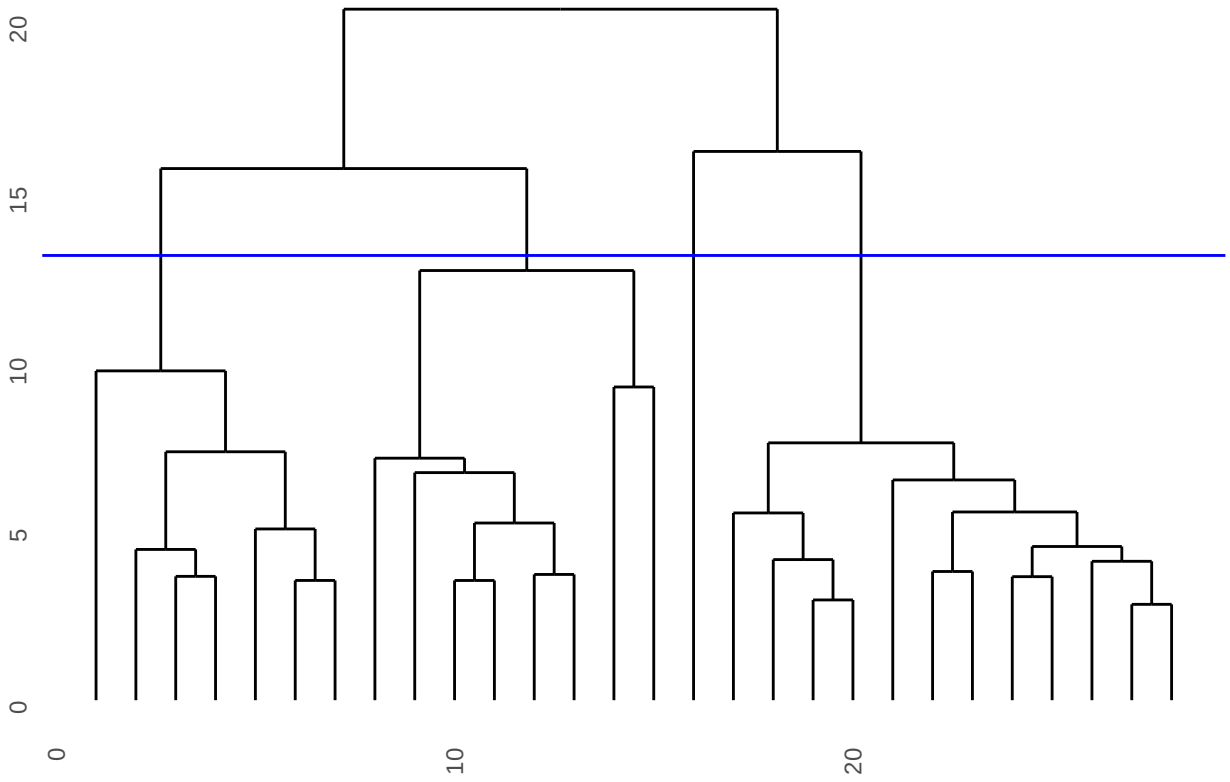
Pour évaluer la qualité de la partition obtenue, nous avons calculé le pseudo F en utilisant les centres de chaque cluster et le dendrogramme généré précédemment.

```
r2psf = PseudoF.R2(donnees, dendro)
```



Nous avons ensuite visualisé le dendrogramme. Au vu des résultats obtenus, je choisis de conserver 4 classes.

```
dendro %>% ggdendrogram(labels = FALSE)+
  geom_hline(yintercept = 13, color = "blue")
```



III) Classification finale

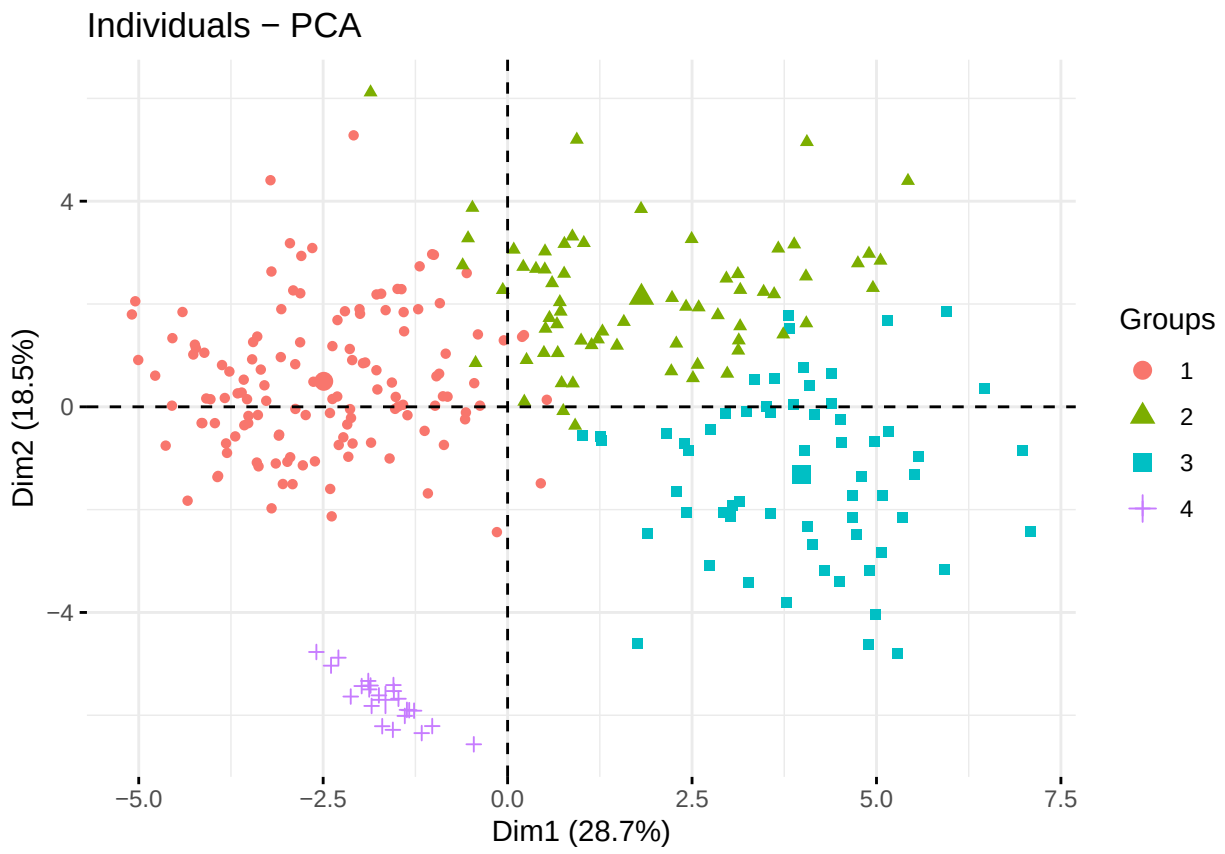
Pour la classification finale, nous avons utilisé l'algorithme K-means avec 4 centres, répété 10 fois. Les profils sur les variables quantitatives ont été générés pour chaque cluster.

```
clusters = donnees_centrees_reduites %>% bind_cols(data.frame(cluster = km$cluster %>% factor))
```

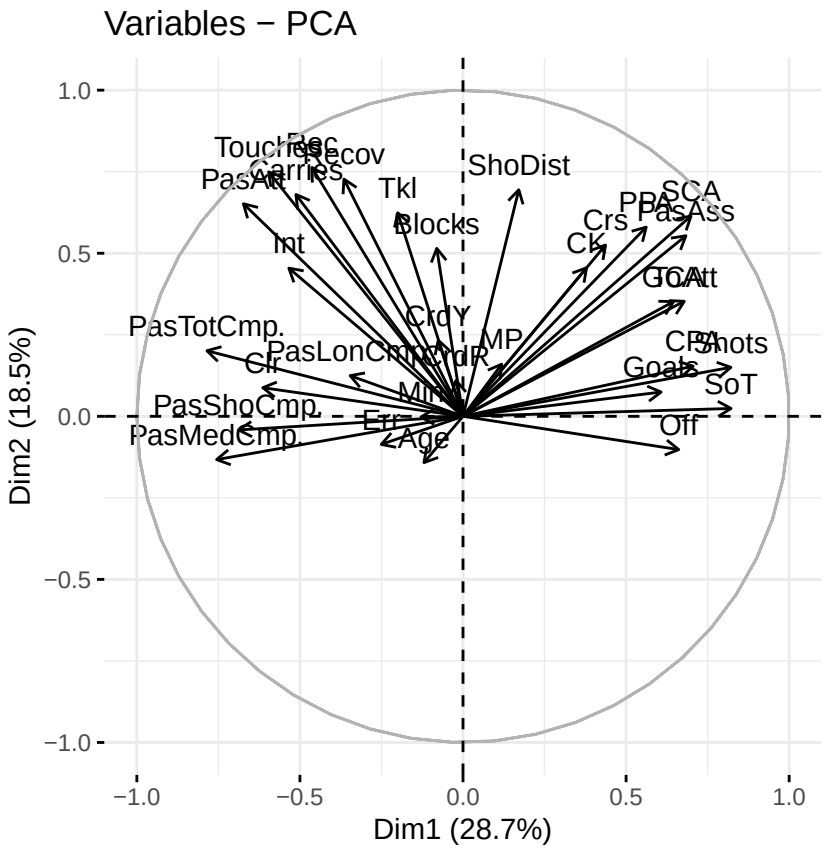
```
clust = km$cluster %>% factor
#profils = catdes(clust, num.var = 32)
```

IV) Profils sur variables quantitatives :

```
quanti.acp %>% fviz_pca_ind(label = "none", habillage = clust)
```




```
quanti.acp %>% fviz_pca_var(axes = c(1,2))
```



Répartition des Variables par Clusters :

```
table(clust, data$Pos)
```

```
##
## clust DF FW GK MF
##     1 85  0  0 46
##     2 11 14  0 41
##     3  2 47  0 12
##     4  0  0 22  0
```

Cluster 1 : Défenseurs (DF) et millieux de terrain (MF)

Ce groupe se compose majoritairement de défenseurs (DF) et de quelques milieux de terrain (MF). Ils se caractérisent par leur implication dans les dégagements, les interceptions fréquentes et le contrôle régulier du ballon. Ils ont également tendance à contrôler le ballon régulièrement et à réussir un grand nombre de passes. ils se démarquent également par leur capacité à récupérer le ballon, à réaliser des tacles et des tirs à distance. Ils sont aussi très impliqués dans les touches de balles et les réceptions de passes, car ils sont les principaux distributeurs et moteurs de l'équipe sur le terrain.

Cluster 2 : Attaquants (FW) et millieux de terrain (MF)

Les joueurs de ce groupe se distinguent par leur capacité à marquer un grand nombre de buts (Goals) et par leurs actions menant à un tir (SCA). Ils tentent également de nombreux tirs cadrés (SOT). Leur rôle principal sur le terrain est de créer et de concrétiser des occasions de but.

Cluster 3 : Les remplacant et les remplacé

Ce cluster est principalement composé de joueurs ayant moins de temps de jeu, souvent en tant que remplaçants ou remplacés j'imagine. Ils se distanguent par le fait d'avoir des score bas dans toutes les variables. Ils se distinguent par des scores bas dans toutes les variables, reflétant leur participation limitée et leur impact moindre sur le jeu.

Cluster 4 : Gardiens de but (GK)

Ce cluster est principalement composé de joueurs de position de gardien de but (GK). Les gardiens de but se distinguent leur participation fréquente aux matchs et aux minutes de jeu. Ils affichent également un taux élevé de réussite dans les passes longues.