# Classification

**Rapport écrit par :**

**Enzo LERICHE, Samuel DARMALINGON**

**BUT Science des données FI EMS**

**2024-03-27**

IUT de Paris - Rives de Seine
Université Paris Cité

# Table des matières

# Données

## Importation des données

```
setwd("C:/Users/User/OneDrive/Documents/BUT/2eme annee/classification")
foot <- read.csv("Foot.csv",sep = ";",header= TRUE,
encoding="latin1")
```

## Modification de la base de données

Dans le cadre de la préparation des données pour l'analyse de performances des joueurs de football, plusieurs étapes de nettoyage et de transformation des données ont été réalisées sur le jeu de données initial, afin de le rendre plus cohérent et adapté aux objectifs d'analyse :

- **Élimination des joueurs en double :** Afin d'assurer l'unicité des informations pour chaque joueur, seules les premières occurrences des joueurs ont été conservées, en maintenant l'intégralité des données associées à ces occurrences. Cette étape garantit que chaque joueur est représenté une seule fois dans le jeu de données, éliminant ainsi les doublons potentiels.
- **Ajustement des identifiants de lignes :** Les noms des joueurs ont été utilisés comme identifiants de lignes dans le dataframe. Cette modification facilite grandement la référence et l'accès aux données spécifiques à un joueur donné, rendant l'analyse et la manipulation des données plus intuitives.
- **Sélection et épuration des colonnes :** La première colonne, correspondant au nom du joueur, a été supprimée du dataframe principal, étant donné que cette information a été transférée en tant qu'identifiant de ligne.
- **Création de la variable 'Crd' :** Une nouvelle variable, 'Crd', représentant le nombre total de cartons (jaunes + rouges) reçus par un joueur, a été calculée. À la suite de cela, les colonnes originales 'CrdY' (cartons jaunes) et 'CrdR' (cartons rouges) ont été retirées du jeu de données.
- **Simplification de la désignation des positions :** La colonne indiquant la position des joueurs sur le terrain a été simplifiée en ne conservant que les deux premiers caractères de chaque désignation. Cette démarche vise à faciliter le regroupement des joueurs selon des catégories de positions similaires et à rendre l'analyse des rôles sur le terrain plus facile.

```
foot <- foot %>% distinct(Player, .keep_all=TRUE)
row.names(foot)=foot$Player foot <- subset(foot, select = -c(1,3))
foot= foot%>%mutate(Crd= CrdY+CrdR)%>% select(-c(CrdY,CrdR))
foot$Pos = substring(foot$Pos,1,2)
```

## Analyses descriptives de base

Nous allons examiner les analyses descriptives de base pour notre jeu de données sur les performances des joueurs de football. Nous utilisons la fonction *summary()* en R, qui nous fournit des statistiques pour chaque variable. Ces statistiques incluent le minimum, le premier quartile, la médiane, la moyenne, le troisième quartile et le maximum.

```
summary(foot)
```

```
##      Pos                 Age              Born            MP        
##  Length:384         Min.   :16.00   Min.   :1984   Min.   : 1.00  
##  Class :character   1st Qu.:23.00   1st Qu.:1993   1st Qu.: 7.00  
##  Mode  :character   Median :26.00   Median :1996   Median :14.00  
##                     Mean   :26.24   Mean   :1996   Mean   :12.46  
##                     3rd Qu.:29.00   3rd Qu.:1999   3rd Qu.:18.00  
##                     Max.   :38.00   Max.   :2006   Max.   :23.00  
##       Min              Goals             Shots              SoT         
##  Min.   :   3.0   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.0000  
##  1st Qu.: 285.8   1st Qu.: 0.000   1st Qu.: 0.4475   1st Qu.: 0.0000  
##  Median : 765.0   Median : 0.000   Median : 1.0500   Median : 0.2750  
##  Mean   : 809.5   Mean   : 1.474   Mean   : 1.4779   Mean   : 0.4908  
##  3rd Qu.:1254.0   3rd Qu.: 2.000   3rd Qu.: 2.0700   3rd Qu.: 0.7075  
##  Max.   :2070.0   Max.   :25.000   Max.   :10.0000   Max.   :10.0000  
##     ShoDist         PasTotCmp.       PasShoCmp.       PasMedCmp.     
##  Min.   : 0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.00  
##  1st Qu.:10.78   1st Qu.: 75.50   1st Qu.: 85.17   1st Qu.: 78.10  
##  Median :15.50   Median : 81.95   Median : 89.50   Median : 86.50  
##  Mean   :14.41   Mean   : 80.62   Mean   : 87.94   Mean   : 83.36  
##  3rd Qu.:19.20   3rd Qu.: 86.92   3rd Qu.: 93.40   3rd Qu.: 92.70  
##  Max.   :34.10   Max.   :100.00   Max.   :100.00   Max.   :100.00  
##    PasLonCmp.         PasAss             PPA              PasAtt       
##  Min.   :  0.00   Min.   : 0.000   Min.   : 0.0000   Min.   :  1.00  
##  1st Qu.: 50.00   1st Qu.: 0.220   1st Qu.: 0.1475   1st Qu.: 36.95  
##  Median : 60.95   Median : 0.875   Median : 0.6750   Median : 52.55  
##  Mean   : 58.46   Mean   : 1.066   Mean   : 0.9437   Mean   : 51.70  
##  3rd Qu.: 71.40   3rd Qu.: 1.650   3rd Qu.: 1.3325   3rd Qu.: 64.08  
##  Max.   :100.00   Max.   :10.000   Max.   :10.0000   Max.   :190.00  
##        CK                SCA               GCA              Tkl        
##  Min.   : 0.0000   Min.   : 0.000   Min.   :0.0000   Min.   : 0.000  
##  1st Qu.: 0.0000   1st Qu.: 1.025   1st Qu.:0.0000   1st Qu.: 0.830  
##  Median : 0.0000   Median : 2.340   Median :0.1550   Median : 1.480  
##  Mean   : 0.5159   Mean   : 2.448   Mean   :0.2811   Mean   : 1.654  
##  3rd Qu.: 0.2450   3rd Qu.: 3.382   3rd Qu.:0.4500   3rd Qu.: 2.260  
##  Max.   :20.0000   Max.   :15.000   Max.   :5.0000   Max.   :10.000  
##      Blocks             Int               Clr               Err        
```

```
##  Min.   : 0.000   Min.   : 0.0000   Min.   : 0.0000   Min.   :0.00000
##  1st Qu.: 0.540   1st Qu.: 0.1575   1st Qu.: 0.3275   1st Qu.:0.00000
##  Median : 0.960   Median : 0.7000   Median : 0.9250   Median :0.00000
##  Mean   : 1.074   Mean   : 0.8138   Mean   : 1.4266   Mean   :0.02193
##  3rd Qu.: 1.330   3rd Qu.: 1.2025   3rd Qu.: 2.0425   3rd Qu.:0.00000
##  Max.   :20.000   Max.   :10.0000   Max.   :15.0000   Max.   :0.42000
##     Touches          ToAtt           Carries           CPA
##  Min.   :  3.00   Min.   : 0.0000   Min.   :  2.00   Min.   : 0.0000
##  1st Qu.: 47.95   1st Qu.: 0.4775   1st Qu.: 28.18   1st Qu.: 0.0000
##  Median : 63.30   Median : 1.3300   Median : 36.70   Median : 0.1800
##  Mean   : 62.63   Mean   : 1.9886   Mean   : 38.53   Mean   : 0.5984
##  3rd Qu.: 74.40   3rd Qu.: 2.8650   3rd Qu.: 45.95   3rd Qu.: 0.7825
##  Max.   :210.00   Max.   :30.0000   Max.   :140.00   Max.   :10.0000
##       Rec              Off             Crs             Recov
##  Min.   :  3.00   Min.   :0.0000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 32.00   1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.: 3.470
##  Median : 43.10   Median :0.0000   Median : 0.815   Median : 5.000
##  Mean   : 43.85   Mean   :0.2079   Mean   : 1.718   Mean   : 5.108
##  3rd Qu.: 52.17   3rd Qu.:0.2500   3rd Qu.: 2.357   3rd Qu.: 6.255
##  Max.   :170.00   Max.   :4.4400   Max.   :30.000   Max.   :20.000
##       Crd
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.1500
##  Mean   :0.2374
##  3rd Qu.:0.2900
##  Max.   :5.0000
```

Nous passons maintenant à une interprétation de certaines des statistiques clés issues de notre jeu de données sur le football. Voici un résumé rapide des points intriguants ou intéressants :

- **Âge (Age) :** La distribution de l'âge, de 16 à 38 ans, montre une large variété de générations parmi les joueurs.
- **Matchs joués (MP) et minutes jouées (Min) :** L'écart entre le minimum et le maximum de matchs joués (1 à 23) et de minutes jouées (3 à 2070) indique une grande disparité dans le temps de jeu attribué aux joueurs.
- **Buts (Goals) :** Le contraste frappant entre une majorité de joueurs n'ayant pas marqué et un joueur ayant inscrit jusqu'à 25 buts souligne l'existence de performances exceptionnelles et peut indiquer des talents offensifs clés au sein de l'échantillon. Il ne faut pas oublier les différences possibles du fait des différents postes des joueurs.

- **Tirs (Shots) et tirs cadrés (SoT) :** Avec des valeurs maximales atteignant respectivement 10 pour les tirs et les tirs cadrés, ces statistiques mettent en lumière des joueurs particulièrement agressifs et efficaces en attaque.

- **Passes complétées (PasTotCmp%) :** Le fait que certains joueurs aient atteint un taux de réussite de 100% dans leurs passes complétées illustre une précision remarquable, bien que ce soit possiblement dans des contextes de jeu moins risqués.

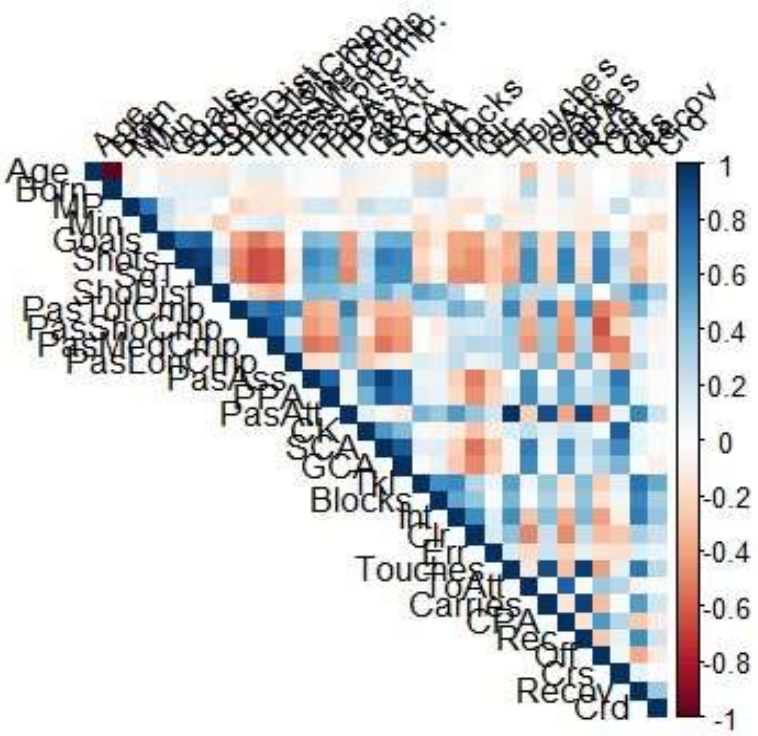
On décide de supprimer les données des joueurs qui ont joués moins de temps que la médiane qui est de 765, pour pouvoir réduire notre jeu de données et qu'il soit plus simple d'analysé.

```
foot <- foot %>%
  filter(Min >765)
```

## Corrélation entre les variables

La matrice de corrélation est un outil clé en statistiques qui illustre clairement comment les variables numériques sont reliées les unes aux autres.

```
foot_var_numeric <-
foot[,sapply(foot,is.numeric)]
cor(foot_var_numeric) par(mfrow=c(1,1))
corrplot(cor(foot_var_numeric),type="upper", tl.col="black",
tl.srt=45,method
="color")
```



Dans notre analyse des données de football, compte tenu du grand nombre de variables, il n'est pas pratique de présenter toutes les corrélations des différentes variables. Néanmoins, certaines d'entre elles se démarquent par leur logique :

- **Corrélation entre Buts (Goals) et Tirs (Shots) :** Une corrélation très élevée est observée, cela nous indique que les joueurs qui tentent le plus de tirs ont tendance à marquer davantage. Cela valide l'hypothèse selon laquelle la fréquence des tirs est un indicateur fiable du nombre de buts qu'un joueur est susceptible de marquer.
- **Corrélation entre Âge (Age) et Année de Naissance (Born) :** une corrélation négative quasi parfaite est observée, ce qui est cohérent étant donné que les joueurs plus jeunes sont nés plus récemment.

## Classification mixte

L'objectif de ce projet est de réaliser une classification des joueurs, et pour ce faire, nous avons choisi d'appliquer une classification mixte.

En effet, la méthode de classification mixte commence par l'application des k-means, qui partitionnent les données en groupes compacts. Ensuite, la classification ascendante hiérarchique (CAH) est utilisée pour créer une hiérarchie de clusters.

### Avec Goals

Pour notre analyse, nous commençons par inclure tous les joueurs, y compris les gardiens de but ("GK"), dans notre classification mixte. Les gardiens jouent un rôle bien différent des autres sur le terrain. Mais, pour voir si ça change quelque chose dans nos groupes, on va aussi faire une analyse sans les gardiens. Ça nous aidera à voir si les groupes de joueurs se forment différemment sans eux.

#### Kmeans

Dans cette section, nous appliquons l'algorithme K-means à notre jeu de données de football:

- Nous retirons les deux premières colonnes du jeu de données pour se concentrer uniquement sur les variables numériques.
- Les données sont standardisées pour que toutes les variables soient comparables.
- Le nombre de clusters est fixé à un dixième du nombre total de lignes dans notre jeu de données standardisé, ce qui nous donne 19 clusters.
- L'algorithme K-means est exécuté avec ces 19 clusters comme points de départ, et le processus est répété 10 fois pour assurer la stabilité des résultats.
- Les centres des clusters obtenus sont sauvegardés.

```r
foots=foot[,c(-1,-2)]

foot.std  = scale(foots)

n = nrow(foot.std)

q= floor(n/10)  #q=19 ici

partition  <-  foot.std  %>%  kmeans(centers=q,  nstart=10)

donnees <- partition$centers
```

### CAH

La suite de notre classification mixte consiste à appliquer la CAH sur les résultats obtenus via l'algorithme K-means :

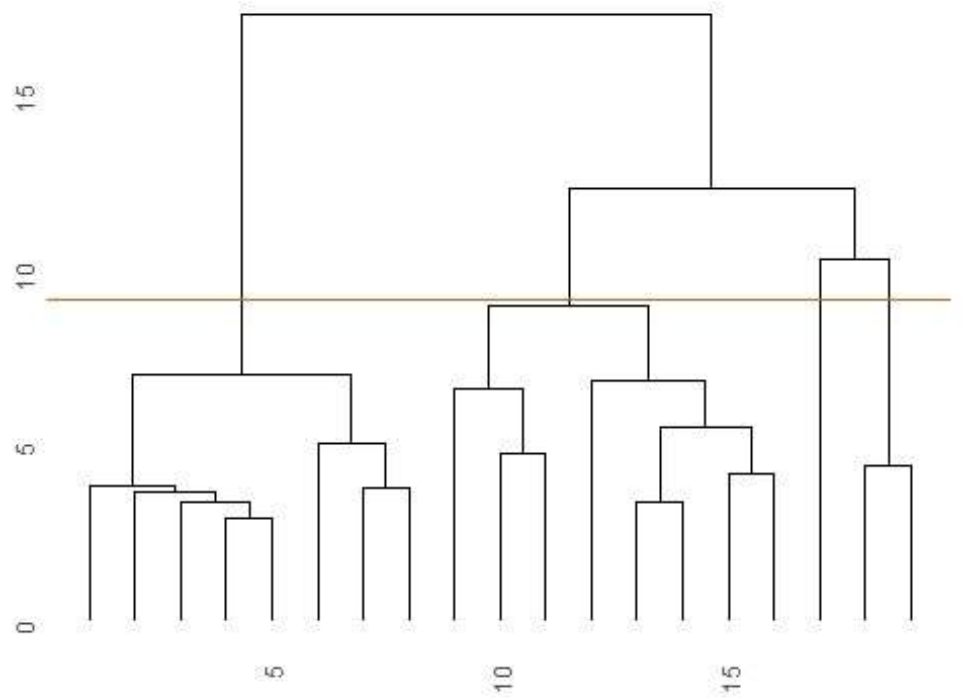
- Nous appliquons la CAH sur les centres des clusters obtenus à partir de l'algorithme K-means. La méthode "ward.D2" est utilisée pour regrouper les données de manière à minimiser la variance au sein des clusters.
- Un dendrogramme est généré pour visualiser la structure des clusters. Nous ajoutons une ligne horizontale à l'interception y = 9 pour indiquer un niveau de coupe potentiel.
- Pour déterminer le nombre optimal de clusters, nous calculons les indices R2 et PseudoF, qui nous permettent de décider du nombre de cluster optimal à faire.

*DENDROGRAMME*

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

dendro  %>%  ggdendrogram(labels=FALSE)  +

geom_hline(yintercept=9,color="chocolate")
```




Le dendrogramme que vous voyez illustre comment les différents groupes de données sont reliés les uns aux autres dans la CAH. Nous avons choisi de couper le dendrogramme à une hauteur de 9, ce qui se traduit par la formation de 4 clusters distincts. Cette décision s'appuie sur l'analyse visuelle du dendrogramme. Pour confirmer ce choix, nous allons utiliser les critères R2 et PseudoF qui vont nous aider à valider ou ajuster le nombre de clusters choisi.

### *R2 et PseudoF*

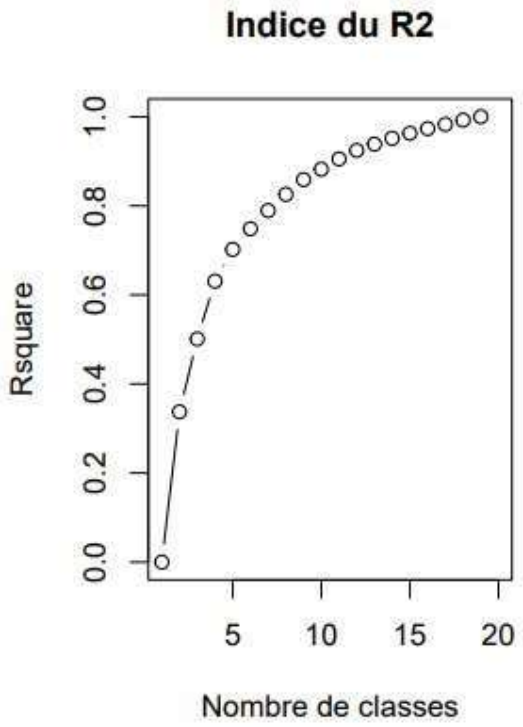
Nous allons maintenant générer les graphiques pour les indices R2 et PseudoF. Le code R utilisé pour ces calculs est celui qui nous a été donné en cours. Ces indices nous aideront à déterminer le nombre optimal de clusters pour notre classification, en complétant l'analyse visuelle faite avec le dendrogramme.

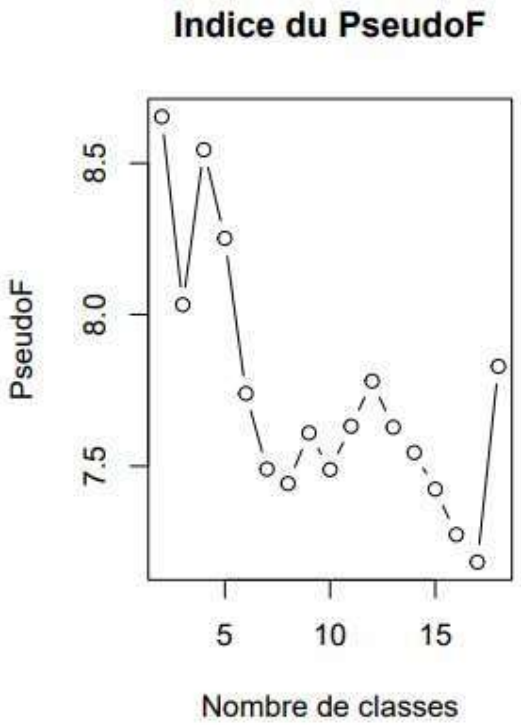
```
R2.PseudoF = function(donnees,dendro, graph=T,
cut=20){   n=nrow(donnees)   R2 = numeric(n)

  Iinter = 0
  g = apply(donnees, 2, mean)
  I = (n-1)/n*sum(diag(var(donnees)))

  for(i in 2:n){
    class =
cutree(dendro,k=i)     ncl =
unique(class)     d =
numeric(length(ncl))      nb =
integer(length(ncl))     for(j
in ncl){       nb[j] =
sum(class==j)
if(nb[j]>1){
        m = apply(donnees[class==j,], 2, mean)
      } else {
        m = donnees[class==j,]
      }
      d[j] = sum((m-g)^2)
    }
    Iinter = (1/n)*sum(nb*d)
    R2[i] = Iinter/I
  }
     ncl = 2:(n-
1)
  PseudoF = R2[ncl]/(1-R2[ncl])*(n-ncl)/(ncl-1)

  if(graph==T){
par(mfrow=c(1,2))
    plot(1:20,R2[1:cut], type = 'b', xlab = "Nombre de classes", ylab =
"Rsqu are")
    title("Indice du R2")
    plot(ncl[1:20],PseudoF[1:cut], type = 'b', xlab = "Nombre de classes",
yl
ab = "PseudoF")
    title("Indice du PseudoF")
  }
  resultat = list(R2,PseudoF)
  names(resultat)=c("Rsquare","PseudoF")
resultat
}
R2.PseudoF(donnees,dendro)
```




```
## $Rsquare
## [1] 0.0000000 0.3373261 0.5010180 0.6308528 0.7021778 0.7485426 0.7892524
## [8] 0.8256616 0.8589087 0.8821921 0.9051202 0.9243958 0.9384853 0.9514938
## [15] 0.9629393 0.9732391 0.9828954 0.9925428 1.0000000
##
## $PseudoF
## [1] 8.653642 8.032643 8.544733 8.251980 7.739724 7.490024 7.442242 7.60950
9
## [9] 7.488392 7.631724 7.780672 7.628141 7.544580 7.423647 7.273591 7.18297
6
## [17] 7.829283
```

L'analyse visuelle des graphiques R2 et PseudoF confirme notre précédente observation faite avec le dendrogramme : conserver quatre classes est bien justifié. Les graphiques montrent que la qualité de la classification est optimale avec ce nombre de groupes, ce qui valide notre choix de découper les données en quatre clusters distincts.

### Classification finale

Nous passons à présent à la classification finale, en utilisant l'algorithme K-means avec quatre centres, basés sur le nombre de classes déterminé comme étant optimal.
L'algorithme est relancé 10 fois pour garantir la stabilité et la fiabilité des clusters finaux obtenus.

```
km <- foot.std %>% kmeans(centers=4, nstart=10)
```

Maintenant que nous avons choisi de travailler avec quatre clusters, nous procédons à l'affectation définitive des groupes. À l'aide de la fonction *cutree*, nous segmentons les données du dendrogramme en quatre clusters distincts. Ensuite, nous transformons les centres de clusters en un dataframe pour une manipulation aisée et attribuons à chaque ligne un facteur de cluster, en les étiquetant de GRP 1 à GRP 4 pour plus de clarté.

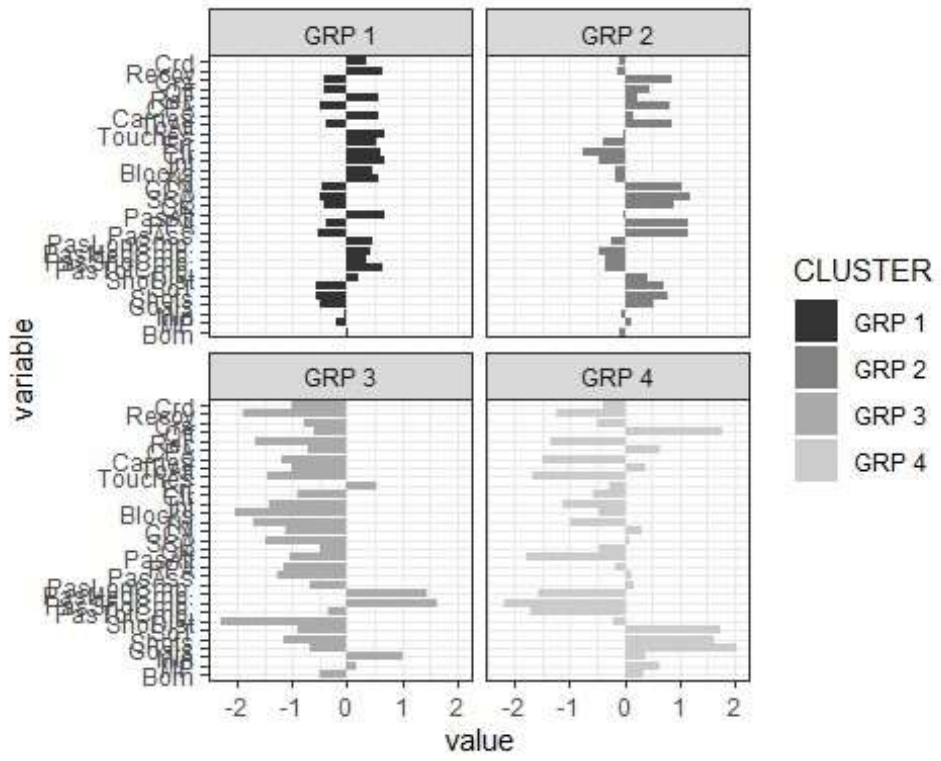
Pour mieux comprendre le profil de chaque cluster, nous calculons la moyenne de toutes les variables pour chaque groupe. Cela nous donne une moyenne des caractéristiques centrales des joueurs dans chaque cluster.

```
res <- cutree(dendro, k=4) df <-
as.data.frame(donnees) df$Cluster
<- factor(res,
levels = 1:4,
                      labels = paste("GRP", 1:4))
clusProfile <- aggregate(df[, 1:31],
                         by = list(df$Cluster),
mean) colnames(clusProfile)[ 1] <- "CLUSTER"
clus_transpose <- melt(clusProfile, id.vars = "CLUSTER")
```

### Représentation graphique

Nous allons à présent visualiser graphiquement les profils moyens des clusters que nous avons définis. Nous créons des diagrammes en barres pour chaque variable, avec des nuances de gris pour distinguer les différents clusters. Grâce à cette représentation, nous serons en mesure de comparer facilement les caractéristiques moyennes entre les groupes et de mieux comprendre les spécificités de chaque cluster.

```
ggplot(clus_transpose) +
  geom_bar(aes(x = variable, y = value,
fill=CLUSTER),stat = "identity") +
scale_fill_grey() +
facet_wrap(~ CLUSTER) +
coord_flip() + theme_bw()
```




Le graphique présente les profils moyens des quatre groupes, et on note des différences marquantes entre eux :

- **GRP 1 :** Ce groupe se caractérise par des valeurs qui tendent vers le positif pour de nombreuses variables, notamment celles liées aux blocks et aux interceptions, ce qui peut indiquer une tendance défensive. Cependant, l'importance des valeurs associées aux passes, aux tirs et aux buts suggère également une implication offensive notable. Ces caractéristiques laissent penser que ce cluster pourrait être composé de joueurs polyvalents, tels que des milieux de terrain offensifs.

- **GRP 2 :** Les joueurs de ce groupe ont des valeurs assez centrées autour de la moyenne, avec plus de variations extrêmes par rapport au GRP 1. Les barres restent relativement courtes pour la plupart, indiquant une homogénéité plus marquée dans les caractéristiques des joueurs.

- **GRP 3 :** Ce groupe se démarque par une prédominance de valeurs négatives sur la majorité des variables, ce qui suggère que les joueurs de ce cluster pourraient avoir des attributs ou des performances généralement inférieures aux moyennes des autres joueurs dans ces catégories. Une hypothèse plausible est que ce cluster pourrait être principalement composé de gardiens de but ("GK"). En effet, par nature de leur poste, les gardiens ont tendance à avoir moins d'occasions de toucher le ballon, de tirer, et de participer à des actions offensives, ce qui expliquerait pourquoi leurs valeurs sont moins élevées sur ces variables par rapport à d'autres joueurs.

- **GRP 4 :** Ce groupe affiche des barres longues qui indiquent des valeurs élevées pour des variables comme les buts et les tirs, suggérant que ces joueurs ont des performances offensives notables. La nature de ces caractéristiques, avec des indicateurs forts dans les actions offensives, laisse penser que ce cluster est probablement constitué d'attaquants. Ce sont des joueurs qui sont activement impliqués dans la création et la concrétisation d'opportunités de but.

Pour vérifier nos hypothèses sur la composition des clusters, nous allons analyser la répartition des postes des joueurs au sein de chacun d'eux. En utilisant la fonction table, nous pourrons observer la proportion de chaque position dans nos clusters, ce qui nous aidera à confirmer si les groupes correspondent effectivement à des rôles spécifiques sur le terrain, comme des attaquants, des milieux de terrain ou des défenseurs.

```
foot$cluster  <- as.character(km$cluster)

table(foot$Pos,foot$cluster)
```

```
##
##        1  2  3  4
##    DF 14 64  0  2
##    FW  5  0  0 28
##    GK  0  0 15  0
##    MF 30 29  0  4
```

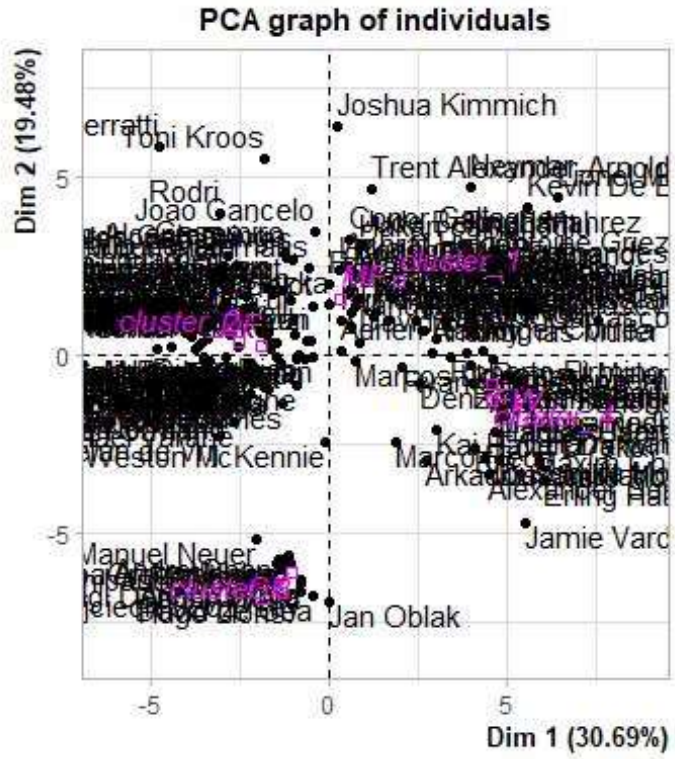
La sortie R révèle clairement la distribution des postes par cluster :

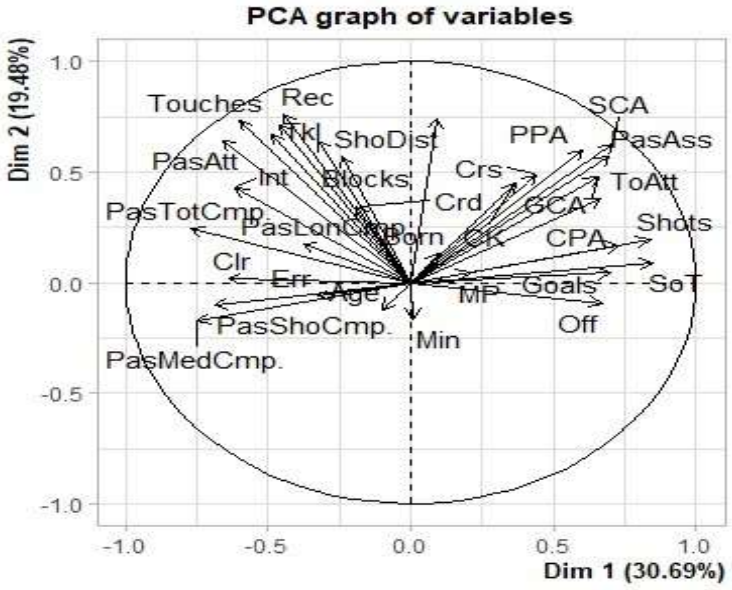
- **GRP 1** présente une répartition équilibrée de milieux de terrain (MF) et un petit nombre d'attaquants (FW) et de défenseurs (DF), ce qui pourrait indiquer une polyvalence des joueurs de ce groupe, capables d'effectuer à la fois des tâches défensives et offensives.
- **GRP 2** a une forte présence de défenseurs (DF) et de milieux de terrain (MF), indiquant un équilibre entre les rôles défensifs et les contributions au jeu de passes et à l'attaque.
- **GRP 3** est exclusivement composé de gardiens de but (GK), ce qui confirme notre hypothèse précédente selon laquelle ce groupe serait distinct en raison des rôles uniques des gardiens sur le terrain.
- **GRP 4** contient majoritairement des attaquants (FW), ce qui est cohérent avec les hautes performances offensives observées dans les caractéristiques de ce cluster.

### Interprétation ACP

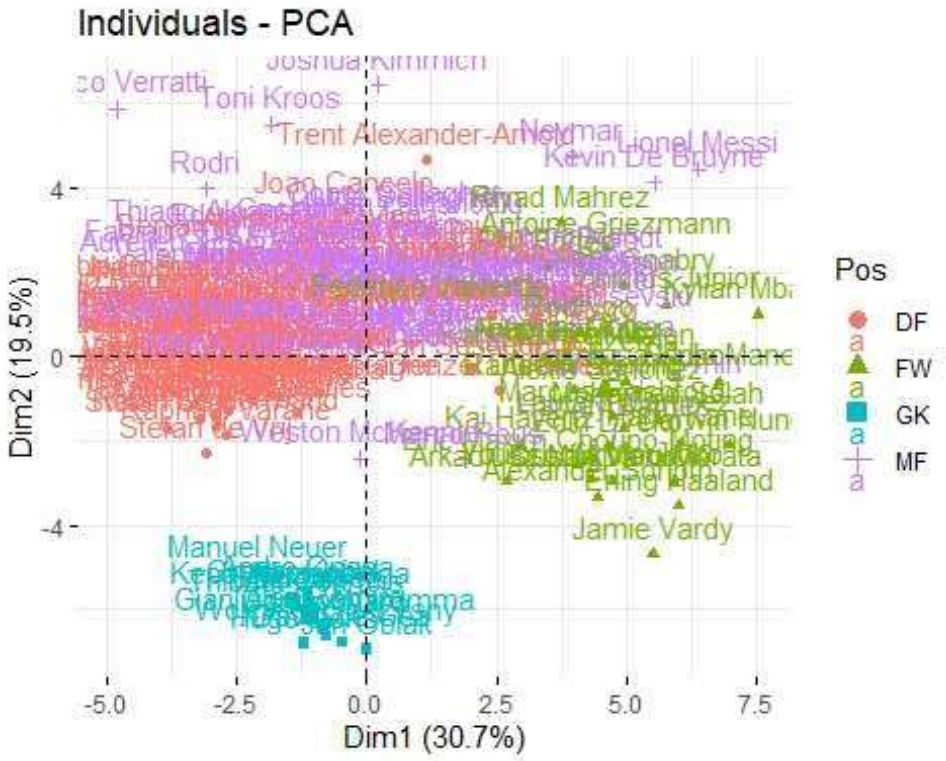
Maintenant que nous avons identifié les clusters, nous allons les représenter visuellement sur une ACP. Chaque cluster sera coloré différemment pour faciliter l'interprétation des regroupements. Cette visualisation nous permettra de voir comment les clusters se distinguent les uns des autres et de confirmer si notre classification reflète des groupements cohérents.

```
foot.acp <- PCA(foot, quali.sup=c(1,34))
```

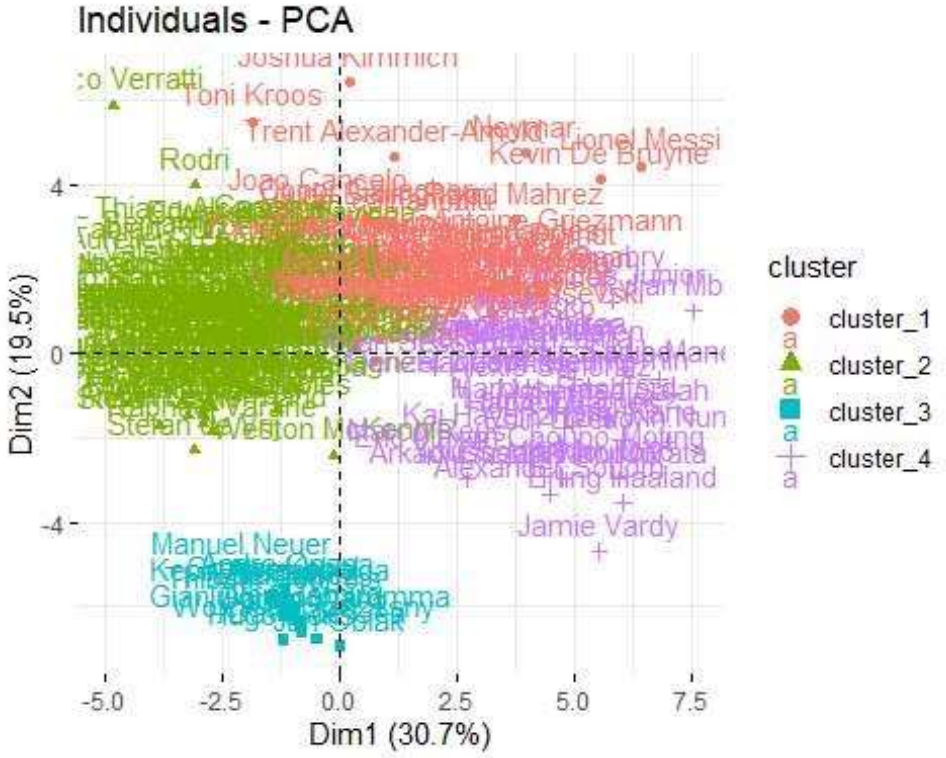


```
fviz_pca_ind(foot.acp,  habillage=1)
```




```
fviz_pca_ind(foot.acp,habillage=34)
```



Les deux graphiques issus de l'Analyse en Composantes Principales (ACP) nous donnent une vision claire de la répartition des individus dans l'espace factoriel.

Sur le premier graphique, les couleurs représentent la position de chaque joueur sur le terrain. Le deuxième graphique utilise des couleurs différentes pour chaque cluster identifié lors de notre classification mixte.

On remarque clairement que les GK sont tout en bas à gauche dans le graphique, ils sont isolés. Cela se confirme aussi sur le 2ème graphique, vu que le cluster 3 est uniquement composé des goals.

On voit que le cluster 2 est composé à la fois de défenseurs et de milieux de terrain. En effet, c'est eux qui tentent le plus de passes, et donc le plus de touches de balles.

Le cluster 1 est composé de 3 postes, les défenseurs, les milieux de terrain et les attaquants.

Le cluster 4, lui est composé majoritairement que des attaquants. En effet, par le cercle de corrélation, ce sont eux qui effectuent le plus de tirs, qui affrontent le plus les défenseurs adverses.

## Sans les Goals

Après avoir réalisé une première classification mixte, nous avons observé une différence notable dans les caractéristiques des joueurs de but par rapport aux autres joueurs. Cependant, les autres joueurs semblaient être répartis dans tous les groupes. Pour explorer davantage les distinctions entre ces joueurs, nous avons décidé de retirer les gardiens de but de notre base de données. Nous avons ensuite procédé à une nouvelle classification mixte afin d'identifier d'éventuelles différences entre ces joueurs restants.

```
foot2 <- foot
foot2 <- foot2 %>% filter(Pos !="GK")
```

### Kmeans

Comme précédent nous avons appliqué l'algorithme K-means à notre jeu de données de football sans les goals :

- Nous retirons les deux premières colonnes ainsi que la dernière colonne qui comportait le cluster auquel chaque joueur appartenait effectuer dans la classification mixte précédente, du jeu de données pour se concentrer uniquement sur les variables numériques.
- Les données sont standardisées pour que toutes les variables soient comparables.
- Le nombre de clusters est fixé à un dixième du nombre total de lignes dans notre jeu de données standardisé, ce qui nous donne 17 clusters.
- L'algorithme K-means est exécuté avec ces 17 clusters comme points de départ, et le processus est répété 10 fois pour assurer la stabilité des résultats.
- Les centres des clusters obtenus sont sauvegardés.

```r
foot2s=foot2[,c(-1,-2,-34)]
foot2.std  = scale(foot2s)

n = nrow(foot2.std)

q= floor(n/10)  #q=17 ici

partition2  <- foot2.std %>% kmeans(centers=q,  nstart=10)

donnees2<-partition2$centers
```

### CAH

Par la suite de notre classification mixte, nous avons appliqué la CAH sur les résultats obtenus via l'algorithme K-means :

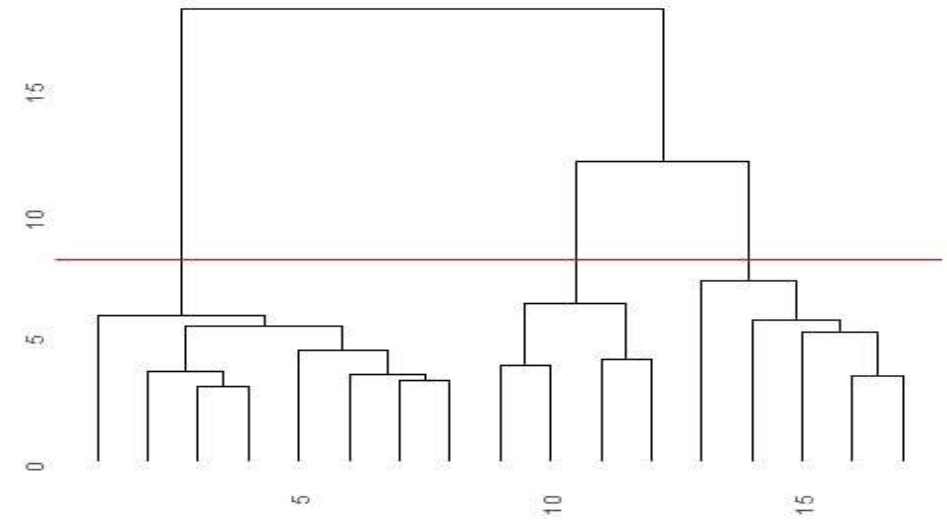
- Nous appliquons la CAH sur les centres des clusters obtenus à partir de l'algorithme K-means avec la méthode "ward.D2.
- Un dendrogramme est généré pour visualiser la structure des clusters. Nous ajoutons une ligne horizontale à l'interception y = 8 pour indiquer un niveau de coupe potentiel.
- Pour déterminer le nombre optimal de clusters, nous calculons les indices R2 et PseudoF, qui nous permettent de décider du nombre de cluster optimal à faire.

#### *DENDROGRAMME*

```r
dendro2  <-  donnees2  %>% dist  %>% hclust(method="ward.D2")

dendro2  %>% ggdendrogram(labels=FALSE)  +

geom_hline(yintercept=8,color="red")
```

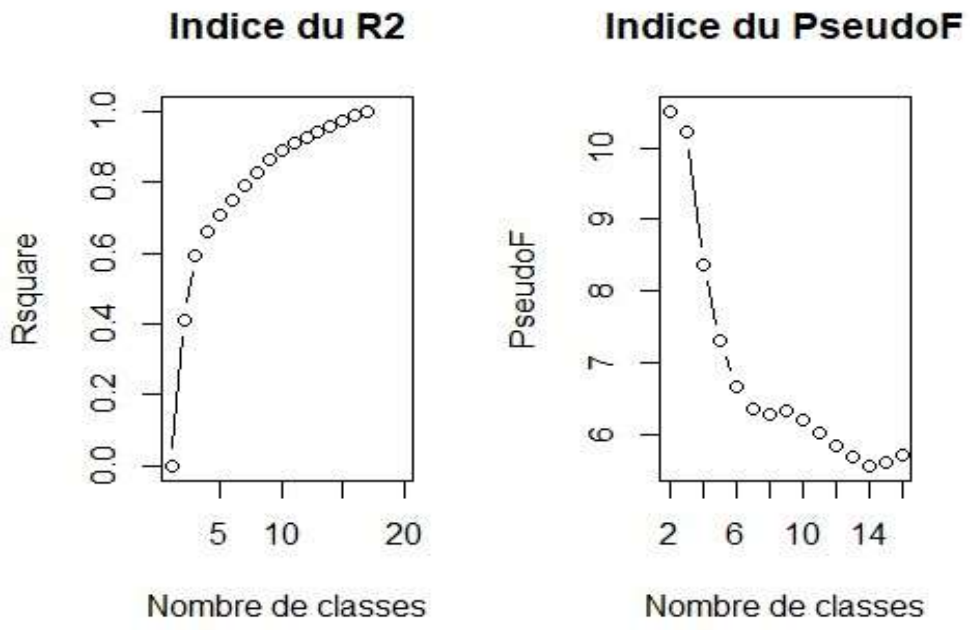



Le dendrogramme visualise les regroupements de données. Nous avons opté pour une découpe du dendrogramme à une hauteur de 8, ce qui conduit à la formation de 3 clusters distincts. Cette décision est basée sur une analyse visuelle du dendrogramme. Afin de confirmer ce choix, nous allons utiliser les indices R2 et PseudoF pour valider ou ajuster le nombre de clusters sélectionné.

### *R2 et PseudoF*

Nous avons fait le R2 et le PseudoF afin de déterminer le nombre optimal de clusters pour notre classification.

```
R2.PseudoF(donnees2,dendro2)
```



```
## $Rsquare
##  [1] 0.0000000 0.4119718 0.5933110 0.6588802 0.7090490 0.7518303
0.7925903
##  [8] 0.8299297 0.8634650 0.8887207 0.9095368 0.9279687 0.9445574
0.9601696
## [15] 0.9751925  0.9884752  1.0000000
##
## $PseudoF
##  [1] 10.508981 10.212168  8.369927  7.311016  6.664902  6.368960
6.274184 ##  [8]  6.324130  6.211644  6.032535  5.855841  5.678888
5.563029  5.615775 ## [15]  5.717944
```

L'inspection visuelle des graphiques R2 et PseudoF renforce notre observation initiale basée sur le dendrogramme : maintenir trois classes est justifié. Les graphiques démontrent que la qualité de la classification est maximale avec ce nombre de groupes, ce qui confirme notre décision de diviser les données en trois clusters distincts.

### Classification finale

La classification finale, en utilisant l'algorithme K-means avec trois centres, basés sur le nombre de classes déterminé comme étant optimal en relançant 10 fois.

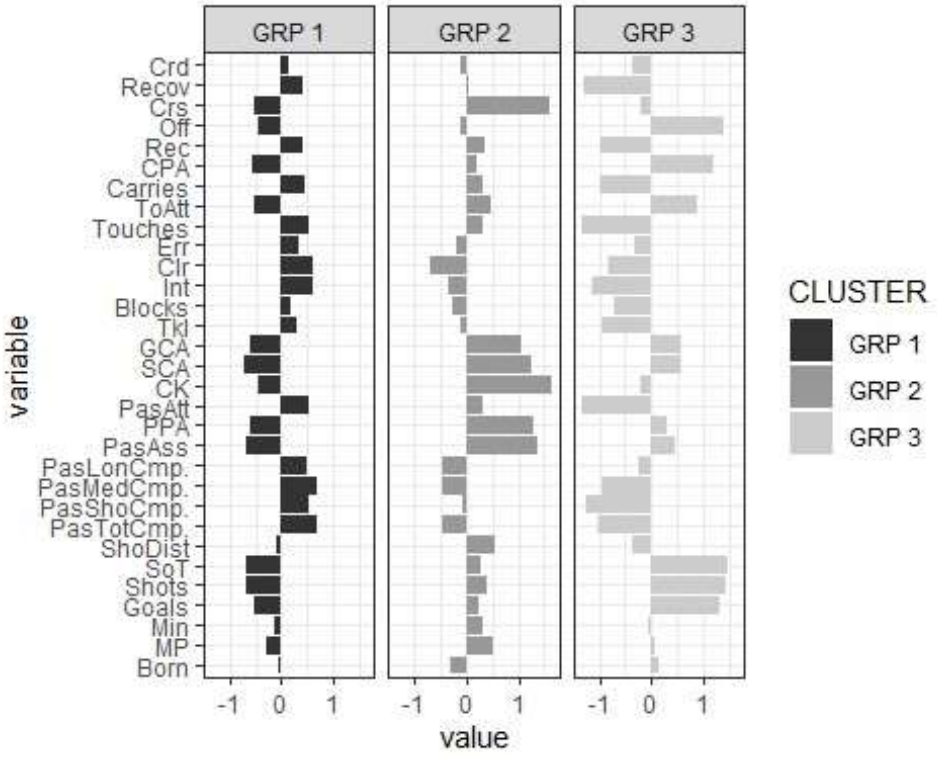
Ensuite, une fois que l'algorithme a créé trois clusters distincts, nous les enregistrons dans un dataframe pour faciliter leur manipulation. À chaque ligne, nous attribuons un facteur de cluster en les étiquetant de GRP 1 à GRP 3. Pour mieux appréhender le profil de chaque cluster, nous calculons ensuite la moyenne de toutes les variables pour chaque groupe.

```
km2 <- foot2.std %>% kmeans(centers=3, nstart=10)

res <- cutree(dendro2, k=3)
df <- as.data.frame(donnees2)
df$Cluster <- factor(res,
                     levels = 1:3,
                     labels = paste("GRP", 1:3))
clusProfile <- aggregate(df[, 1:31],
                         by = list(df$Cluster),
                         mean)
colnames(clusProfile)[ 1] <- "CLUSTER"
clus_transpose2 <- melt(clusProfile, id.vars = "CLUSTER")
```

### Représentation graphique

```
ggplot(clus_transpose2) +
  geom_bar(aes(x = variable, y = value, fill = CLUSTER),
           stat = "identity") +
  scale_fill_grey() +
  facet_wrap(~ CLUSTER) +
  coord_flip() + theme_bw()
```




Le graphique présente les profils moyens des trois groupes, et on note des différences marquantes entre eux :

- **GRP 1 :** Ce groupe se caractérise par des valeurs centrées autour de la moyenne, avec des barres relativement courtes, ce qui suggère une homogénéité entre les joueurs. De plus, les pourcentages de passes réussies sont positifs, ce qui indique que dans ce groupe, les joueurs sont efficaces pour faire circuler la balle sur le terrain.

- **GRP 2 :** Les joueurs de ce groupe se distinguent par des valeurs positives, notamment en ce qui concerne les aspects défensifs tels que les corners, les passes longues et les moyens de dégager la balle près des cages de but.

- **GRP 3 :** Ce groupe se distingue par de nombreuses variables négatives concernant les touches de balles de but. Néanmoins, c'est dans ce groupe que les joueurs ont de grandes valeurs positives en termes de tirs et de buts. Donc, nous pouvons en déduire que ce groupe est composé d'attaquants.

Pour vérifier nos hypothèses sur la composition des clusters, nous allons analyser la répartition des postes des joueurs au sein de chacun d'eux.

```
foot2$cluster <- as.character(km2$cluster)

table(foot2$Pos,foot2$cluster)
```

```
##
##      1  2  3
##  DF 64 14  2
##  FW  0  5 28
##  MF 29 30  4
```

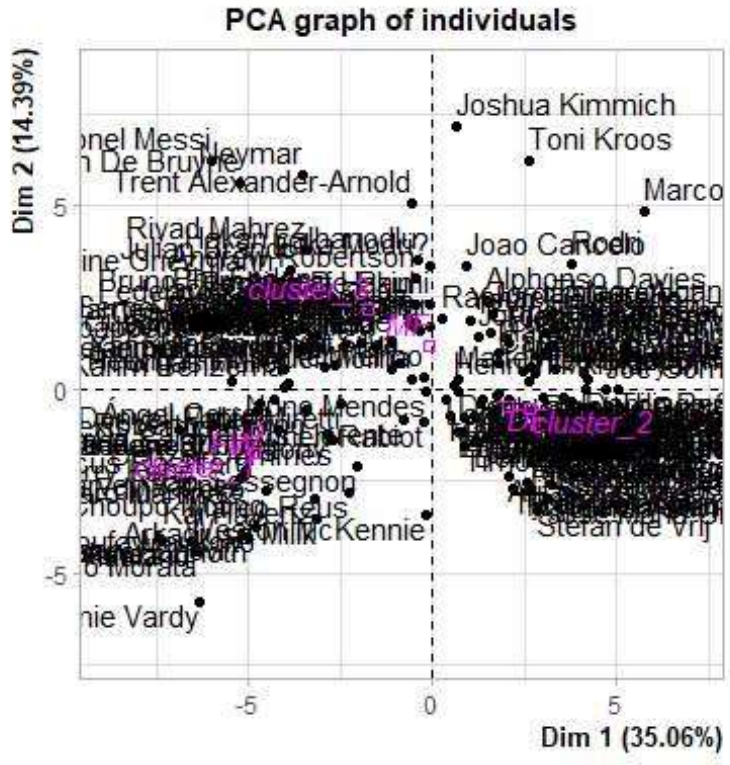
La sortie R montre la distribution des postes par cluster :

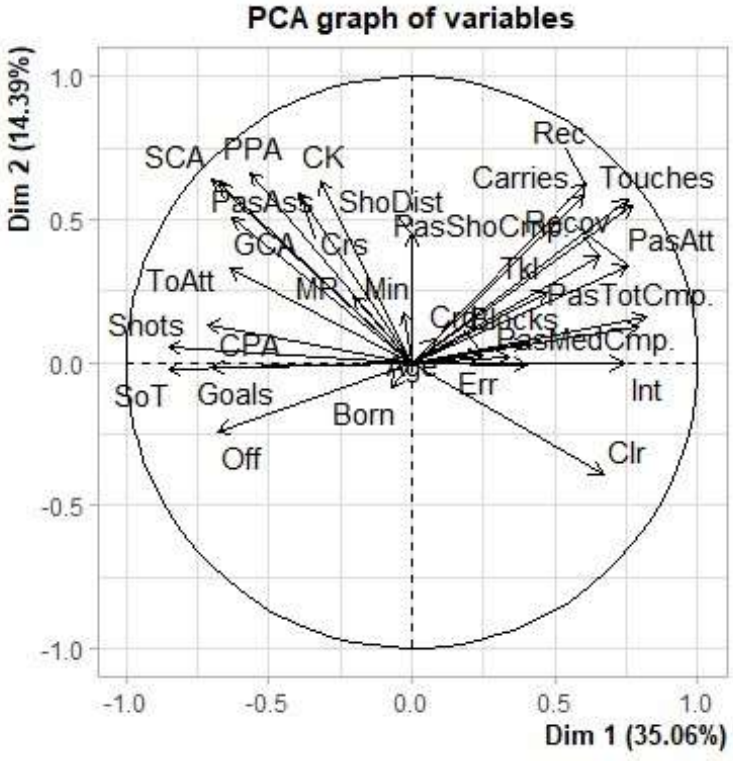
- **GRP 1** a une forte présence de défenseurs (DF) et de milieux de terrain (MF), indiquant un équilibre entre les rôles défensifs et les contributions au jeu de passes et à l'attaque.
- **GRP 2** contient majoritairement des milieux de terrains (MF) et quelques défenseurs (DF).
- **GRP 3** présente une répartition de l'ensemble des postes. Néanmoins, les attaquants (FW) sont majoritairement représentés ce qui explique, le grand nombre de buts et de tirs.

### Interprétation ACP

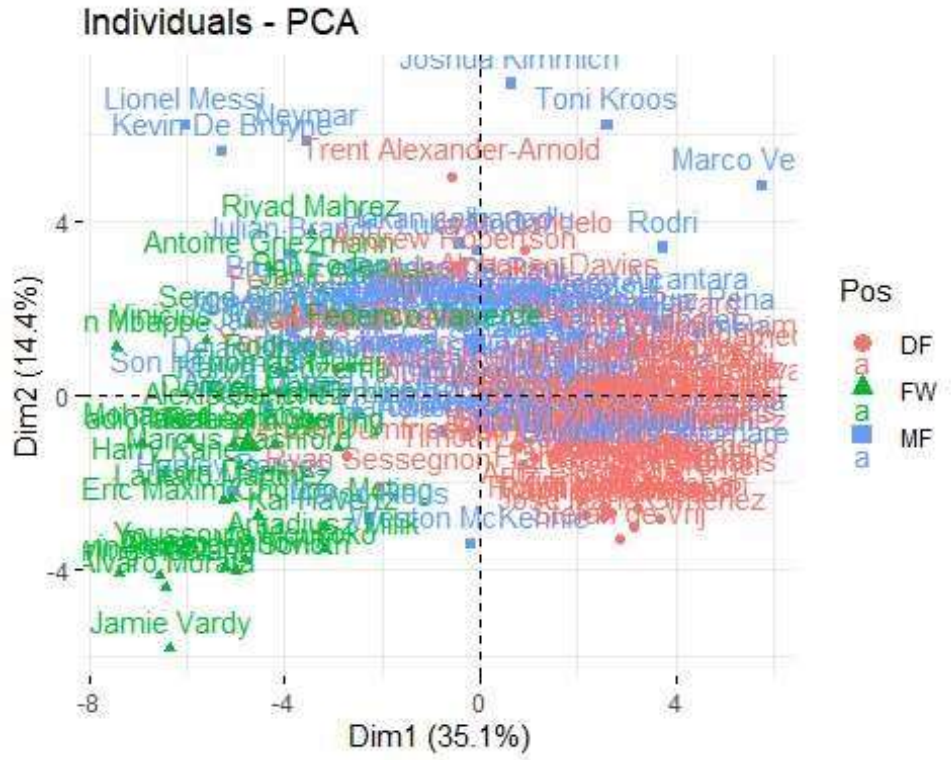
Maintenant que nous avons identifié les clusters, nous allons les représenter visuellement sur une ACP, pour observer les distinctions entre les différents clusters.

```
foot2.acp <- PCA(foot2, quali.sup = c(1,34))
```

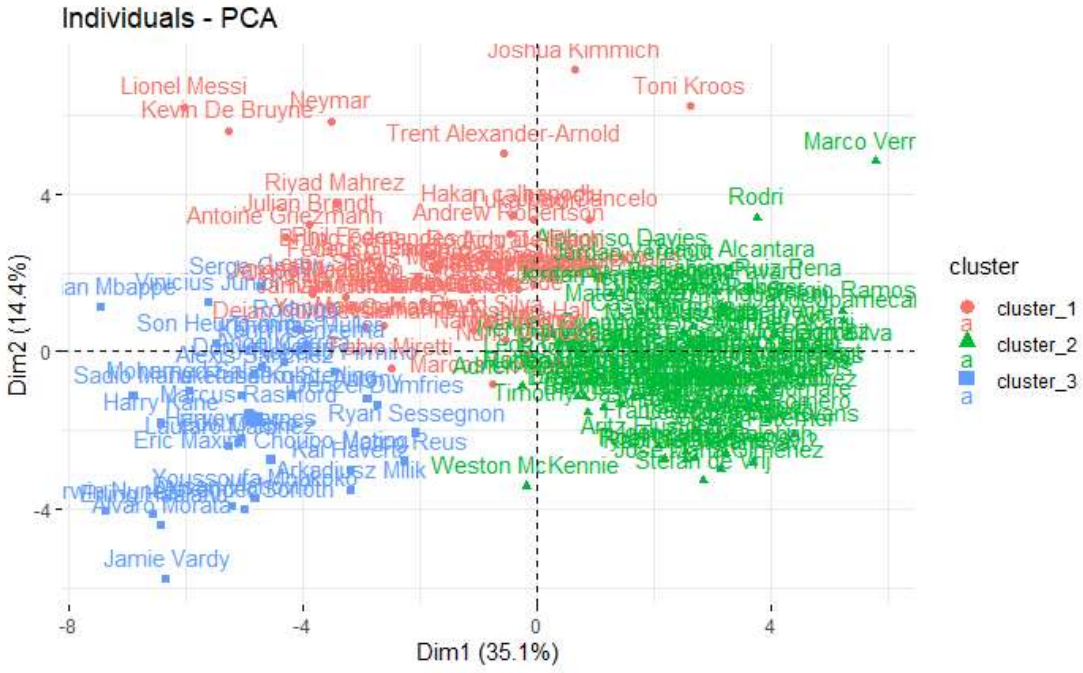



```
fviz_pca_ind(foot2.acp, habillage=1)
```




```
fviz_pca_ind(foot2.acp,  habillage=34)
```



Les deux graphiques issus de l'Analyse en Composantes Principales (ACP) nous donnent une vision claire de la répartition des individus dans l'espace factoriel.

Sur le premier graphique, les couleurs représentent la position de chaque joueur sur le terrain. Le deuxième graphique utilise des couleurs différentes pour chaque cluster identifié lors de notre classification mixte.

Il est clair que les défenseurs (DF) sont représentés à droite, tandis que les attaquants (FW) sont à gauche, comme le confirme le cercle de corrélation montrant que les attaquants réalisent le plus de buts et de tirs, tandis que les défenseurs ont plus de touches et de passes. Les milieux de terrain sont dispersés entre les deux extrêmes.

Le cluster 2 est principalement composé de défenseurs et de quelques milieux de terrain, ce qui s'explique par leur tendance à tenter le plus de passes et donc à avoir plus de touches de balle.

Le cluster 3 comprend des attaquants et quelques milieux, car ce sont eux qui cherchent à marquer des buts et à contribuer à la victoire de leur équipe par des tirs réussis.

Le cluster 1 rassemble des attaquants, des défenseurs et des milieux qui réalisent des touches et des passes décisives contribuant aux buts de l'équipe.