Analyse en composantes principales sur le jeu de données FIFA 2022

Axel-Cleris Gailloty

2023-02-18

# Introduction

Dans ce travail, je propose de revenir sur l’étude que nous avons réalisée ensemble en classe sur le jeu de données FIFA 2022 afin de vous fournir plus d’élements d’aide à l’interprétation d’une analyse en composantes principales avec R.

## Présentation du jeu de données

<https://sports-statistics.com/soccer/fifa-2022-dataset-csvs/>

## Lecture du jeu de données

Le jeu de données est au format Excel, nous pouvons utiliser la fonction read\_excel() qui provient du package *readxl* pour l’importer dans R.

library(readxl)  
fifa22 <- read\_excel("fifa22.xlsx")

## L’échantillon

Nous pouvons afficher les 10 premières lignes du jeu de données pour avoir un aperçu des données.

fifa22[1:10, 1:10]

| short\_name | player\_positions | overall | potential | value\_eur | wage\_eur | age | dob | height\_cm | weight\_kg |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L, Messi | RW, ST, CF | 93 | 93 | 78000000 | 320000 | 34 | 1987-06-24 | 170 | 72 |
| R, Lewandowski | ST | 92 | 92 | 119500000 | 270000 | 32 | 1988-08-21 | 185 | 81 |
| Cristiano Ronaldo | ST, LW | 91 | 91 | 45000000 | 270000 | 36 | 1985-02-05 | 187 | 83 |
| Neymar Jr | LW, CAM | 91 | 91 | 129000000 | 270000 | 29 | 1992-02-05 | 175 | 68 |
| K, De Bruyne | CM, CAM | 91 | 91 | 125500000 | 350000 | 30 | 1991-06-28 | 181 | 70 |
| J, Oblak | GK | 91 | 93 | 112000000 | 130000 | 28 | 1993-01-07 | 188 | 87 |
| K, Mbappé | ST, LW | 91 | 95 | 194000000 | 230000 | 22 | 1998-12-20 | 182 | 73 |
| M, Neuer | GK | 90 | 90 | 13500000 | 86000 | 35 | 1986-03-27 | 193 | 93 |
| M, ter Stegen | GK | 90 | 92 | 99000000 | 250000 | 29 | 1992-04-30 | 187 | 85 |
| H, Kane | ST | 90 | 90 | 129500000 | 240000 | 27 | 1993-07-28 | 188 | 89 |

# Réaliser une ACP sur le jeu de données

## Le principe d’une ACP

L’ACP permet de décrire un jeu de données, de le résumer, d’en réduire la dimensionnalité. L’ACP réalisée sur les individus du tableau de données répond à différentes questions :

Etude des individus (i.e. des athlètes) : deux athlètes sont proches s’ils ont des résultats similaires. On s’intéresse à la variabilité entre individus. Y a-t-il des similarités entre les individus pour toutes les variables ? Peut-on établir des profils d’athlètes ? Peut-on opposer un groupe d’individus à un autre ? Etude des variables (i.e. des performances) : on étudie les liaisons linéaires entre les variables. Les objectifs sont de résumer la matrice des corrélations et de chercher des variables synthétiques: peut-on résumer les performances des athlètes par un petit nombre de variables ? Lien entre les deux études : peut-on caractériser des groupes d’individus par des variables ?

Nous allons réaliser une ACP sur le jeu de données FIFA 222 en ne prenant que les 100 premiers joueurs du jeu de données et les variables suivantes :

* “height\_cm”, “weight\_kg”, “skill\_dribbling”, “skill\_ball\_control”, “movement\_acceleration”, “movement\_agility”, “power\_shot\_power”, “power\_jumping”, “power\_stamina”, “power\_strength”, “mentality\_aggression”, “mentality\_interceptions”, “defending\_marking\_awareness”, “defending\_standing\_tackle”, “defending\_sliding\_tackle”, “goalkeeping\_diving”, “goalkeeping\_handling”, “goalkeeping\_kicking”, “goalkeeping\_speed”

comme variables quantitatives illustratives

* “overall”, “potential”, “value\_eur”, “pace”, “shooting”, “passing”, “dribbling”, “defending”,“physic” et comme variable qualitative illustrative
* “body\_type”.

## Séparer les variables dans des objets R pour une meilleure visibilité.

Nous allons mettre les noms des colonnes dans des variables R pour mieux comprendre le code.

### A quoi servent les variables quanti actives ?

Les variables quantitatives illustratives servent à calculer les composantes de l’ACP. Elles entrent dans le calcul de la matrice de variance-covariance.

var\_quanti\_actives <- c("height\_cm", "weight\_kg", "skill\_dribbling", "skill\_ball\_control",  
"movement\_acceleration", "movement\_agility", "power\_shot\_power",  
"power\_jumping", "power\_stamina", "power\_strength", "mentality\_aggression",  
"mentality\_interceptions", "defending\_marking\_awareness",  
"defending\_standing\_tackle", "defending\_sliding\_tackle", "goalkeeping\_diving",  
"goalkeeping\_handling", "goalkeeping\_kicking", "goalkeeping\_speed")

### A quoi servent les variables quantitatives illustratives ?

Les variables illustratives n’influencent pas la construction des composantes principales de l’analyse. Elles aident à l’interprétation des dimensions de variabilité.

var\_quanti\_illustratives <- c("overall", "potential", "value\_eur", "pace", "shooting", "passing", "dribbling","defending","physic")

### A quoi servent les variables qualitatives illustratives

Comme les variables quantitatives illustratives, elles aident à l’interprétation des composantes. Elles sont qualitatives (modalités discrètes) : par club, par league, par nationalité …

l’ACP ne se base que sur des données numériques.

var\_quali\_illustratives <- c("body\_type")

### Filtrer le jeu de données

fifa\_100 <- fifa22[1:100,]

Ne prendre que les colonnes (variables) qui nous intéressent

variables\_acp <- c(var\_quanti\_actives, var\_quanti\_illustratives, var\_quali\_illustratives)

fifa\_100 <- fifa\_100[, variables\_acp]

Parmi les colonnes que nous avons sélectionnée, rien n’indique le nom des joueurs. Si nous ne précisions pas les noms des joueurs, les joueurs seront numérotés de 1 à 100 comme dans le tableau suivant.

fifa\_100[1:10, 1:10]

| height\_cm | weight\_kg | skill\_dribbling | skill\_ball\_control | movement\_acceleration | movement\_agility | power\_shot\_power | power\_jumping | power\_stamina | power\_strength |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 170 | 72 | 96 | 96 | 91 | 91 | 86 | 68 | 72 | 69 |
| 185 | 81 | 85 | 88 | 77 | 77 | 90 | 85 | 76 | 86 |
| 187 | 83 | 88 | 88 | 85 | 86 | 94 | 95 | 77 | 77 |
| 175 | 68 | 95 | 95 | 93 | 96 | 80 | 64 | 81 | 53 |
| 181 | 70 | 88 | 91 | 76 | 79 | 91 | 63 | 89 | 74 |
| 188 | 87 | 12 | 30 | 43 | 67 | 59 | 78 | 41 | 78 |
| 182 | 73 | 93 | 91 | 97 | 92 | 86 | 78 | 88 | 77 |
| 193 | 93 | 30 | 46 | 54 | 51 | 68 | 77 | 43 | 80 |
| 187 | 85 | 21 | 30 | 38 | 39 | 66 | 79 | 35 | 78 |
| 188 | 89 | 83 | 85 | 65 | 71 | 91 | 79 | 83 | 85 |

fifa\_100 <- data.frame(fifa\_100)  
rownames(fifa\_100) <- head(fifa22$short\_name, 100)

Maintenant chaque ligne correspond au nom d’un joueur.

fifa\_100[1:10, 1:10]

|  | height\_cm | weight\_kg | skill\_dribbling | skill\_ball\_control | movement\_acceleration | movement\_agility | power\_shot\_power | power\_jumping | power\_stamina | power\_strength |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L, Messi | 170 | 72 | 96 | 96 | 91 | 91 | 86 | 68 | 72 | 69 |
| R, Lewandowski | 185 | 81 | 85 | 88 | 77 | 77 | 90 | 85 | 76 | 86 |
| Cristiano Ronaldo | 187 | 83 | 88 | 88 | 85 | 86 | 94 | 95 | 77 | 77 |
| Neymar Jr | 175 | 68 | 95 | 95 | 93 | 96 | 80 | 64 | 81 | 53 |
| K, De Bruyne | 181 | 70 | 88 | 91 | 76 | 79 | 91 | 63 | 89 | 74 |
| J, Oblak | 188 | 87 | 12 | 30 | 43 | 67 | 59 | 78 | 41 | 78 |
| K, Mbappé | 182 | 73 | 93 | 91 | 97 | 92 | 86 | 78 | 88 | 77 |
| M, Neuer | 193 | 93 | 30 | 46 | 54 | 51 | 68 | 77 | 43 | 80 |
| M, ter Stegen | 187 | 85 | 21 | 30 | 38 | 39 | 66 | 79 | 35 | 78 |
| H, Kane | 188 | 89 | 83 | 85 | 65 | 71 | 91 | 79 | 83 | 85 |

## Réalisation de l’ACP avec FactoMineR

Pour réaliser une ACP avec FactoMineR, il fournir à la fonction PCA les arguments suivants :

* X : le jeu de données
* scale.unit : TRUE ou FALSE (pour réduire et centrer les données numériques)
* ncp : Le nombre de composantes principales
* quanti.sup : Les indices des variables quantitatives illustratives
* quali.sup : Les indices des variables qualitatives illustratives
* ind.sup : Les indices des lignes contenant les individus illustratifs.
* graph : TRUE ou FALSE pour indiquer si R doit afficher les graphiques de l’ACP

Nous ne sommes pas obligés de renseigner tous les paramètres de la fonction PCA.

La condition minimale à satisfaire pour réaliser une ACP avec la fonction PCA c’est de fournir en argument au paramètre X un jeu de données ne contenant que des variables numériques et ne contenant aucune donnée manquante.

Dans ce cas une ACP sera réalisée en considérant toutes les colonnes comme variables actives.

Dans la pratique nous voulons avoir un contrôle sur les résultats de l’ACP donc nous allons donner plus de précisions à la fonction PCA.

Dans notre présent cas nous allons préciser les indices des variables actives, variables quantitatives illustratives et des variables qualitatives illustratives.

### Trouver les positions des colonnes dans le jeu de données

idx\_var\_actives <- match(var\_quanti\_actives, colnames(fifa\_100))

idx\_var\_quanti\_illustratives <- match(var\_quanti\_illustratives, colnames(fifa\_100))

idx\_var\_quali\_illustratives <- match(var\_quali\_illustratives, colnames(fifa\_100))

# Charger la librairie FactoMineR  
library(FactoMineR)  
library(factoextra) # Pour représenter graphiques

Le chargement a nécessité le package : ggplot2

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

premiere\_acp <- PCA(X = fifa\_100, scale.unit = TRUE, quanti.sup = idx\_var\_quanti\_illustratives,   
 quali.sup = idx\_var\_quali\_illustratives, graph = FALSE)

Warning in PCA(X = fifa\_100, scale.unit = TRUE, quanti.sup =  
idx\_var\_quanti\_illustratives, : Missing values are imputed by the mean of the  
variable: you should use the imputePCA function of the missMDA package

## Combien de composantes retenir ?

Comme tout objet dans R, nous pouvons utiliser la fonction summary() sur l’objet *premiere\_acp* pour afficher certaines informations.

summary(premiere\_acp)

Call:  
PCA(X = fifa\_100, scale.unit = TRUE, quanti.sup = idx\_var\_quanti\_illustratives,   
 quali.sup = idx\_var\_quali\_illustratives, graph = FALSE)   
  
  
Eigenvalues  
 Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6 Dim.7  
Variance 9.547 4.167 1.873 1.047 0.767 0.369 0.285  
% of var. 50.248 21.933 9.860 5.509 4.037 1.941 1.501  
Cumulative % of var. 50.248 72.181 82.041 87.549 91.586 93.527 95.029  
 Dim.8 Dim.9 Dim.10 Dim.11 Dim.12 Dim.13 Dim.14  
Variance 0.217 0.167 0.156 0.130 0.080 0.055 0.046  
% of var. 1.141 0.882 0.819 0.685 0.423 0.288 0.240  
Cumulative % of var. 96.170 97.051 97.871 98.555 98.978 99.266 99.506  
 Dim.15 Dim.16 Dim.17 Dim.18 Dim.19  
Variance 0.028 0.022 0.019 0.016 0.009  
% of var. 0.149 0.115 0.100 0.085 0.045  
Cumulative % of var. 99.655 99.770 99.870 99.955 100.000  
  
Individuals (the 10 first)  
 Dist Dim.1 ctr cos2 Dim.2 ctr  
L, Messi | 3.928 | 0.864 0.078 0.048 | -3.479 2.905  
R, Lewandowski | 2.857 | 0.591 0.037 0.043 | -0.374 0.033  
Cristiano Ronaldo | 3.758 | 0.342 0.012 0.008 | -1.332 0.426  
Neymar Jr | 3.933 | 1.321 0.183 0.113 | -3.624 3.151  
K, De Bruyne | 2.615 | 2.024 0.429 0.599 | -0.590 0.084  
J, Oblak | 7.200 | -6.992 5.121 0.943 | 0.261 0.016  
K, Mbappé | 3.419 | 1.301 0.177 0.145 | -2.278 1.245  
M, Neuer | 7.505 | -6.894 4.978 0.844 | 0.310 0.023  
M, ter Stegen | 8.221 | -7.291 5.568 0.786 | 0.954 0.218  
H, Kane | 2.843 | 0.364 0.014 0.016 | 0.467 0.052  
 cos2 Dim.3 ctr cos2   
L, Messi 0.784 | 0.758 0.306 0.037 |  
R, Lewandowski 0.017 | 2.560 3.497 0.803 |  
Cristiano Ronaldo 0.126 | 3.179 5.395 0.716 |  
Neymar Jr 0.849 | -0.177 0.017 0.002 |  
K, De Bruyne 0.051 | 0.008 0.000 0.000 |  
J, Oblak 0.001 | -0.362 0.070 0.003 |  
K, Mbappé 0.444 | 1.812 1.753 0.281 |  
M, Neuer 0.002 | 0.693 0.257 0.009 |  
M, ter Stegen 0.013 | -0.068 0.002 0.000 |  
H, Kane 0.027 | 2.501 3.339 0.774 |  
  
Variables (the 10 first)  
 Dim.1 ctr cos2 Dim.2 ctr cos2   
height\_cm | -0.532 2.970 0.284 | 0.623 9.302 0.388 |  
weight\_kg | -0.579 3.517 0.336 | 0.544 7.111 0.296 |  
skill\_dribbling | 0.910 8.677 0.828 | -0.304 2.215 0.092 |  
skill\_ball\_control | 0.930 9.052 0.864 | -0.228 1.251 0.052 |  
movement\_acceleration | 0.649 4.416 0.422 | -0.532 6.793 0.283 |  
movement\_agility | 0.618 4.006 0.382 | -0.692 11.478 0.478 |  
power\_shot\_power | 0.570 3.404 0.325 | -0.283 1.921 0.080 |  
power\_jumping | 0.133 0.185 0.018 | 0.484 5.630 0.235 |  
power\_stamina | 0.931 9.080 0.867 | -0.021 0.011 0.000 |  
power\_strength | -0.003 0.000 0.000 | 0.717 12.345 0.514 |  
 Dim.3 ctr cos2   
height\_cm 0.374 7.474 0.140 |  
weight\_kg 0.490 12.823 0.240 |  
skill\_dribbling 0.181 1.756 0.033 |  
skill\_ball\_control 0.151 1.209 0.023 |  
movement\_acceleration 0.180 1.726 0.032 |  
movement\_agility 0.013 0.010 0.000 |  
power\_shot\_power 0.597 18.995 0.356 |  
power\_jumping 0.421 9.482 0.178 |  
power\_stamina 0.027 0.038 0.001 |  
power\_strength 0.590 18.551 0.348 |  
  
Supplementary continuous variables  
 Dim.1 cos2 Dim.2 cos2 Dim.3 cos2   
overall | -0.175 0.031 | -0.150 0.022 | 0.245 0.060 |  
potential | -0.120 0.014 | -0.034 0.001 | 0.247 0.061 |  
value\_eur | 0.197 0.039 | -0.151 0.023 | 0.217 0.047 |  
pace | 0.018 0.000 | -0.514 0.264 | 0.234 0.055 |  
shooting | -0.062 0.004 | -0.675 0.456 | 0.475 0.226 |  
passing | 0.065 0.004 | -0.584 0.341 | -0.213 0.045 |  
dribbling | 0.031 0.001 | -0.860 0.740 | 0.006 0.000 |  
defending | 0.138 0.019 | 0.827 0.683 | -0.501 0.251 |  
physic | 0.021 0.000 | 0.783 0.613 | 0.368 0.135 |  
  
Supplementary categories  
 Dist Dim.1 cos2 v.test Dim.2 cos2  
Lean (170-) | 4.652 | 2.626 0.319 0.850 | -1.551 0.111  
Lean (170-185) | 1.998 | 1.800 0.811 1.932 | -0.285 0.020  
Lean (185+) | 2.085 | -1.133 0.295 -0.745 | 1.406 0.455  
Normal (170-) | 4.242 | 1.185 0.078 0.383 | -3.678 0.752  
Normal (170-185) | 1.039 | 0.458 0.194 0.464 | 0.123 0.014  
Normal (185+) | 2.883 | -1.948 0.457 -1.850 | 2.016 0.489  
Stocky (185+) | 7.414 | -7.103 0.918 -2.299 | -0.482 0.004  
Unique | 0.437 | 0.020 0.002 0.088 | -0.217 0.246  
 v.test Dim.3 cos2 v.test   
Lean (170-) -0.760 | -3.182 0.468 -2.325 |  
Lean (170-185) -0.462 | -0.714 0.128 -1.729 |  
Lean (185+) 1.399 | -0.381 0.033 -0.565 |  
Normal (170-) -1.802 | -1.022 0.058 -0.747 |  
Normal (170-185) 0.189 | -0.690 0.441 -1.577 |  
Normal (185+) 2.898 | -0.166 0.003 -0.356 |  
Stocky (185+) -0.236 | -0.869 0.014 -0.635 |  
Unique -1.472 | 0.322 0.544 3.264 |

L’objet eig nous renvoie un tbaleau qui nous permet de déterminer combien de composantes retenir dans l’ACP.

La règle de Kaiser-Guttman “Le nombre des valeurs-propres supérieures à l’unité d’une matrice d’inter-corrélation est égal au nombre de facteur à extraire”.

La règle de Kaiser nous aide à sélectionner le nombre optimal de composantes à retenir dans l’analyse. Nous regardons les valeurs de eigenvalues >= 1. Dans le présent exemple nous allons retenir 4 composantes car à partir de la 5e composante la valeur propre est inférieure à 1.

data.frame(premiere\_acp$eig)

|  | eigenvalue | percentage.of.variance | cumulative.percentage.of.variance |
| --- | --- | --- | --- |
| comp 1 | 9.5471330 | 50.2480682 | 50.24807 |
| comp 2 | 4.1672307 | 21.9327930 | 72.18086 |
| comp 3 | 1.8733818 | 9.8599043 | 82.04077 |
| comp 4 | 1.0466180 | 5.5085157 | 87.54928 |
| comp 5 | 0.7669903 | 4.0367911 | 91.58607 |
| comp 6 | 0.3687937 | 1.9410196 | 93.52709 |
| comp 7 | 0.2852761 | 1.5014534 | 95.02855 |
| comp 8 | 0.2167906 | 1.1410032 | 96.16955 |
| comp 9 | 0.1674949 | 0.8815523 | 97.05110 |
| comp 10 | 0.1556909 | 0.8194258 | 97.87053 |
| comp 11 | 0.1300895 | 0.6846817 | 98.55521 |
| comp 12 | 0.0803336 | 0.4228086 | 98.97802 |
| comp 13 | 0.0546775 | 0.2877761 | 99.26579 |
| comp 14 | 0.0456224 | 0.2401180 | 99.50591 |
| comp 15 | 0.0282535 | 0.1487024 | 99.65461 |
| comp 16 | 0.0219337 | 0.1154405 | 99.77005 |
| comp 17 | 0.0190689 | 0.1003628 | 99.87042 |
| comp 18 | 0.0160618 | 0.0845357 | 99.95495 |
| comp 19 | 0.0085590 | 0.0450475 | 100.00000 |

Chaque composante résume la variance (intertie) du jeu de donnée. Par exemple, la première composante explique à elle seule 50% de la variance du jeu de données. La composante 2 explique 22% de la variance, la composante 3 explique 9,0% etc…

Cumulativement les 4 composantes que nous retiendront dans l’analyse expliquent 87% de la variance totale du jeu de données.

Nous avons sélectionné 19 variables actives dans l’analyse en composantes principales. Ces variables sont **height\_cm, weight\_kg, skill\_dribbling, skill\_ball\_control, movement\_acceleration, movement\_agility, power\_shot\_power, power\_jumping, power\_stamina, power\_strength, mentality\_aggression, mentality\_interceptions, defending\_marking\_awareness, defending\_standing\_tackle, defending\_sliding\_tackle, goalkeeping\_diving, goalkeeping\_handling, goalkeeping\_kicking, goalkeeping\_speed**.

Grâce à l’ACP nous avons pû extraire 87% de l’information (l’inertie / variance) contenue dans 19 variables grâce à 4 variables synthétiques que nous appelons composantes principales.

Le reste de l’analyse consite donc à décomposer quelles variables initiales (variables actives) entrent dans la composition de chacune des composantes.

## Quel % de variance arrive-t-on à expliquer avec ces composantes ?

On explique 87% de l’information totale du jeu de données avec 4 composantes.

### Une composante est une variable synthétique

Une composante est une variable synthétique qu’on a produite en combinant plusieurs autres variables du jeu de données initiale. C’est comme si nous passons d’un jeu de données composé de 19 variables à un jeu de donnée comportant 4 variables. Ces 4 variables contiennent 87% de l’information (l’inertie) des 19 variables.

head(premiere\_acp$ind$coord[, 1:4], 10)

Dim.1 Dim.2 Dim.3 Dim.4  
L, Messi 0.8639975 -3.4791101 0.757518877 -0.09909159  
R, Lewandowski 0.5911543 -0.3735085 2.559612629 0.09019679  
Cristiano Ronaldo 0.3422720 -1.3324889 3.179235062 0.31692814  
Neymar Jr 1.3212620 -3.6237944 -0.177133227 -0.10031339  
K, De Bruyne 2.0244058 -0.5900705 0.007897385 -0.31451301  
J, Oblak -6.9919111 0.2609832 -0.361708412 0.10880091  
K, Mbappé 1.3010843 -2.2775042 1.812225234 0.15612222  
M, Neuer -6.8938294 0.3100558 0.693224836 2.53958834  
M, ter Stegen -7.2906748 0.9535446 -0.068233113 -2.88743392  
H, Kane 0.3644423 0.4665339 2.500966364 -0.19761714

## CONTRIBUTIONS DES VARIABLES DANS LA FORMATION DES COMPOSANTES (AXES)

Dans le tableau suivant on a la contribution de chaque variable sur chaque composante.

data.frame(premiere\_acp$var$contrib)

|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 |
| --- | --- | --- | --- | --- | --- |
| height\_cm | 2.9695910 | 9.3024729 | 7.4738528 | 0.1081234 | 5.7529521 |
| weight\_kg | 3.5166848 | 7.1109956 | 12.8230670 | 0.0079525 | 2.5589778 |
| skill\_dribbling | 8.6774838 | 2.2147705 | 1.7561643 | 0.3773605 | 1.2914646 |
| skill\_ball\_control | 9.0520261 | 1.2510708 | 1.2092937 | 0.0114848 | 1.9997358 |
| movement\_acceleration | 4.4156500 | 6.7926582 | 1.7260254 | 3.5834977 | 5.1409644 |
| movement\_agility | 4.0060637 | 11.4781632 | 0.0096276 | 0.1884923 | 2.7455472 |
| power\_shot\_power | 3.4043781 | 1.9206961 | 18.9950148 | 0.2352840 | 6.5307485 |
| power\_jumping | 0.1849522 | 5.6303160 | 9.4817076 | 5.5828326 | 61.4001784 |
| power\_stamina | 9.0799427 | 0.0109319 | 0.0380706 | 0.0098265 | 0.0002790 |
| power\_strength | 0.0000763 | 12.3445726 | 18.5512450 | 0.0044995 | 0.4894286 |
| mentality\_aggression | 6.4286011 | 5.1461117 | 0.6899345 | 0.0644162 | 1.2615961 |
| mentality\_interceptions | 4.8801318 | 9.0696394 | 6.0931991 | 0.1794466 | 0.3811871 |
| defending\_marking\_awareness | 4.7075976 | 8.9590209 | 6.8539531 | 0.1668909 | 0.1833431 |
| defending\_standing\_tackle | 5.3708491 | 8.9370339 | 4.4473400 | 0.0062492 | 0.2376443 |
| defending\_sliding\_tackle | 4.7239332 | 9.6578578 | 5.9028024 | 0.0214779 | 0.0905763 |
| goalkeeping\_diving | 9.5382489 | 0.0031340 | 1.3796286 | 0.6280016 | 0.5268139 |
| goalkeeping\_handling | 9.6501930 | 0.0086747 | 1.3613914 | 0.3666088 | 0.4699497 |
| goalkeeping\_kicking | 9.3746580 | 0.0007133 | 1.1692877 | 0.6657360 | 0.2820795 |
| goalkeeping\_speed | 0.0189386 | 0.1611666 | 0.0383943 | 87.7918191 | 8.6565336 |

La variable height\_cm contribue à hauteur de 2.9% à former l’axe 1, la variable goalkeeping\_diving contribue à hauteur de 9.53% à former l’axe 1.

La somme de toutes les contributions des variables sur un axe (composante) est égale à 100.

sum(premiere\_acp$var$contrib[, 1])

[1] 100

## Afficher les poids des variables qui forment l’axe 1

Les variables contribuent différemment à définir la première composante. Nous pouvons par ordre décroissant la contribution des variables sur la dimension 1.

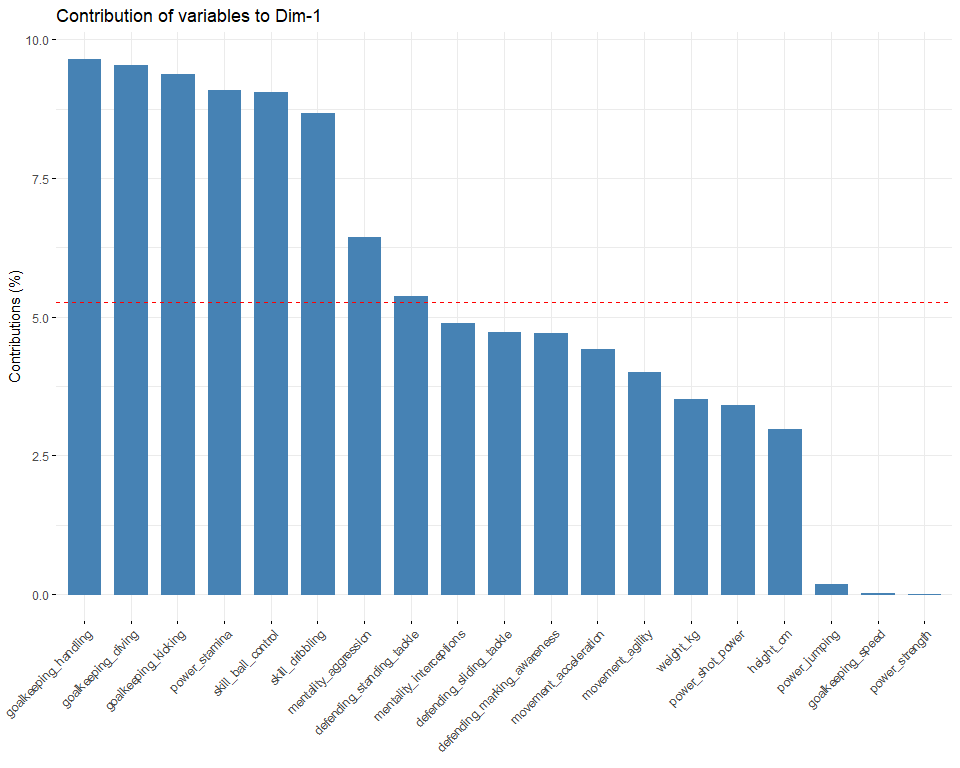
sort(premiere\_acp$var$contrib[, 1], decreasing = TRUE)

goalkeeping\_handling goalkeeping\_diving   
 9.650193e+00 9.538249e+00   
 goalkeeping\_kicking power\_stamina   
 9.374658e+00 9.079943e+00   
 skill\_ball\_control skill\_dribbling   
 9.052026e+00 8.677484e+00   
 mentality\_aggression defending\_standing\_tackle   
 6.428601e+00 5.370849e+00   
 mentality\_interceptions defending\_sliding\_tackle   
 4.880132e+00 4.723933e+00   
defending\_marking\_awareness movement\_acceleration   
 4.707598e+00 4.415650e+00   
 movement\_agility weight\_kg   
 4.006064e+00 3.516685e+00   
 power\_shot\_power height\_cm   
 3.404378e+00 2.969591e+00   
 power\_jumping goalkeeping\_speed   
 1.849522e-01 1.893860e-02   
 power\_strength   
 7.626043e-05

On voit par exemple que sur l’axe 1 la variable goalkeeping\_handling a une plus grande contribution que les autres.

Nous pouvons aussi représenter graphiquement le pourcentage de contribution des variables sur la dimension 1.

fviz\_contrib(premiere\_acp, choice = "var", axes = 1)



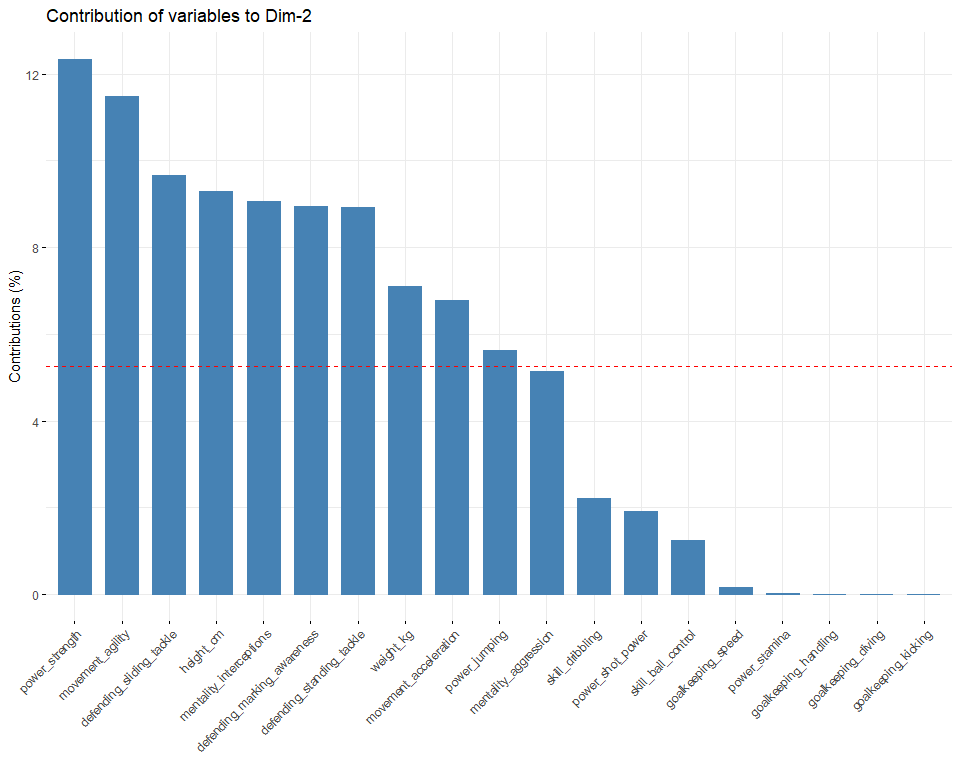
## Afficher les poids des variables qui forment l’axe 2

sort(premiere\_acp$var$contrib[, 2], decreasing = TRUE)

power\_strength movement\_agility   
 12.344572598 11.478163213   
 defending\_sliding\_tackle height\_cm   
 9.657857789 9.302472868   
 mentality\_interceptions defending\_marking\_awareness   
 9.069639435 8.959020938   
 defending\_standing\_tackle weight\_kg   
 8.937033856 7.110995568   
 movement\_acceleration power\_jumping   
 6.792658190 5.630315993   
 mentality\_aggression skill\_dribbling   
 5.146111689 2.214770462   
 power\_shot\_power skill\_ball\_control   
 1.920696119 1.251070798   
 goalkeeping\_speed power\_stamina   
 0.161166562 0.010931925   
 goalkeeping\_handling goalkeeping\_diving   
 0.008674713 0.003134013   
 goalkeeping\_kicking   
 0.000713271

Sur l’axe 2, c’est la variable power\_strength qui a la plus forte contribution. Le poids relatif de cet axe est important dans la construction de l’axe. Pour se donner une meilleure idée de ce que pourrait représenter cet axe nous pouvons afficher tous les poids relatifs des variables par odre croissant.

fviz\_contrib(premiere\_acp, choice = "var", axes = 2)



## LES COORDONNÉES DES VARIABLES SUR LES AXES

Nous venons de voir à combien de % chaque variable contribue sur chaque axe. on aimerait savoir le signe de chaque variable sur les axes.

data.frame(premiere\_acp$var$coord)

|  | Dim.1 | Dim.2 | Dim.3 | Dim.4 | Dim.5 |
| --- | --- | --- | --- | --- | --- |
| height\_cm | -0.5324573 | 0.6226199 | 0.3741842 | -0.0336398 | -0.2100585 |
| weight\_kg | -0.5794330 | 0.5443635 | 0.4901275 | 0.0091231 | -0.1400968 |
| skill\_dribbling | 0.9101928 | -0.3038003 | 0.1813826 | -0.0628452 | -0.0995259 |
| skill\_ball\_control | 0.9296284 | -0.2283309 | 0.1505147 | -0.0109637 | -0.1238458 |
| movement\_acceleration | 0.6492827 | -0.5320392 | 0.1798195 | 0.1936634 | 0.1985716 |
| movement\_agility | 0.6184369 | -0.6916079 | 0.0134299 | 0.0444161 | 0.1451140 |
| power\_shot\_power | 0.5701057 | -0.2829131 | 0.5965309 | -0.0496238 | -0.2238084 |
| power\_jumping | 0.1328820 | 0.4843844 | 0.4214601 | 0.2417249 | 0.6862459 |
| power\_stamina | 0.9310608 | -0.0213438 | 0.0267060 | -0.0101413 | 0.0014630 |
| power\_strength | -0.0026983 | 0.7172355 | 0.5895215 | 0.0068624 | -0.0612688 |
| mentality\_aggression | 0.7834201 | 0.4630878 | 0.1136886 | 0.0259652 | 0.0983683 |
| mentality\_interceptions | 0.6825780 | 0.6147787 | -0.3378593 | 0.0433373 | -0.0540710 |
| defending\_marking\_awareness | 0.6704033 | 0.6110181 | -0.3583304 | 0.0417937 | -0.0374997 |
| defending\_standing\_tackle | 0.7160741 | 0.6102678 | -0.2886445 | -0.0080873 | -0.0426932 |
| defending\_sliding\_tackle | 0.6715655 | 0.6344015 | -0.3325388 | 0.0149931 | 0.0263574 |
| goalkeeping\_diving | -0.9542690 | 0.0114281 | -0.1607660 | 0.0810727 | 0.0635658 |
| goalkeeping\_handling | -0.9598525 | 0.0190130 | -0.1596999 | 0.0619435 | 0.0600372 |
| goalkeeping\_kicking | -0.9460502 | -0.0054519 | -0.1480041 | 0.0834728 | 0.0465137 |
| goalkeeping\_speed | 0.0425217 | -0.0819523 | -0.0268192 | 0.9585640 | -0.2576718 |

## Afficher les 5 premières coordonnées positives sur l’axe 1

sort(premiere\_acp$var$coord[, 1], decreasing = TRUE)[1:5]

power\_stamina skill\_ball\_control skill\_dribbling   
 0.9310608 0.9296284 0.9101928   
 mentality\_aggression defending\_standing\_tackle   
 0.7834201 0.7160741

Sur l’axe 1, les variables power\_stamina, skill\_ball\_control, skill\_dribbling ont des coordonnées positives. cela signifie que si un individu du jeu de données a une coordonnée positive c’est que cet individu est représenté par ces variables.

## Afficher les 5 premières coordonnées négatives sur l’axe 1

sort(premiere\_acp$var$coord[, 1], decreasing = FALSE)[1:5]

goalkeeping\_handling goalkeeping\_diving goalkeeping\_kicking   
 -0.9598525 -0.9542690 -0.9460502   
 weight\_kg height\_cm   
 -0.5794330 -0.5324573

## Afficher les 5 premières coordonnées positives sur l’axe 2

sort(premiere\_acp$var$coord[, 2], decreasing = TRUE)[1:5]

power\_strength defending\_sliding\_tackle   
 0.7172355 0.6344015   
 height\_cm mentality\_interceptions   
 0.6226199 0.6147787   
defending\_marking\_awareness   
 0.6110181

### Afficher les 5 premières coordonnées négatives sur l’axe 2

sort(premiere\_acp$var$coord[, 2], decreasing = FALSE)[1:5]

movement\_agility movement\_acceleration skill\_dribbling   
 -0.6916079 -0.5320392 -0.3038003   
 power\_shot\_power skill\_ball\_control   
 -0.2829131 -0.2283309

## Corrélation entre les variables et les dimensions

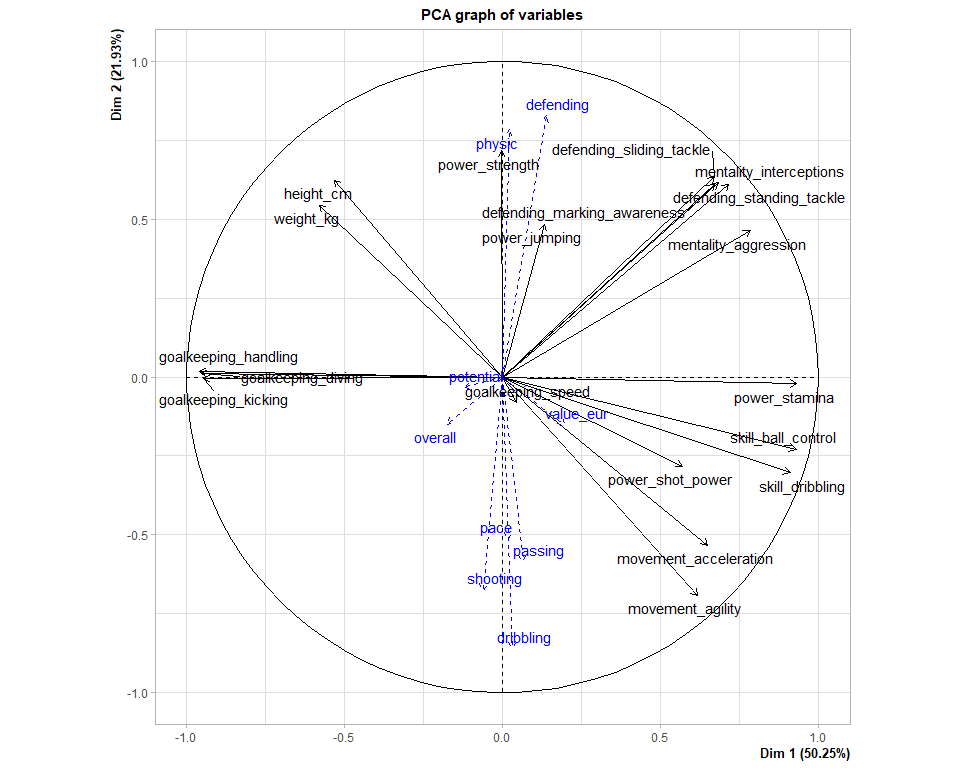
dimdesc(premiere\_acp, c(1,2))

$Dim.1  
  
Link between the variable and the continuous variables (R-square)  
=================================================================================  
 correlation p.value  
power\_stamina 0.9310608 1.057853e-44  
skill\_ball\_control 0.9296284 2.798423e-44  
skill\_dribbling 0.9101928 2.692916e-39  
mentality\_aggression 0.7834201 5.824363e-22  
defending\_standing\_tackle 0.7160741 5.565933e-17  
mentality\_interceptions 0.6825780 5.238686e-15  
defending\_sliding\_tackle 0.6715655 2.051306e-14  
defending\_marking\_awareness 0.6704033 2.361122e-14  
movement\_acceleration 0.6492827 2.737652e-13  
movement\_agility 0.6184369 7.073146e-12  
power\_shot\_power 0.5701057 5.968991e-10  
value\_eur 0.1972808 4.913873e-02  
height\_cm -0.5324573 1.185330e-08  
weight\_kg -0.5794330 2.682184e-10  
goalkeeping\_kicking -0.9460502 9.222093e-50  
goalkeeping\_diving -0.9542690 3.417323e-53  
goalkeeping\_handling -0.9598525 6.620247e-56  
  
Link between variable abd the categories of the categorical variables  
================================================================  
 Estimate p.value  
body\_type=Stocky (185+) -6.591048 0.02073359  
  
$Dim.2  
  
Link between the variable and the continuous variables (R-square)  
=================================================================================  
 correlation p.value  
defending 0.8266670 3.253779e-26  
physic 0.7826747 6.758755e-22  
power\_strength 0.7172355 4.699250e-17  
defending\_sliding\_tackle 0.6344015 1.375379e-12  
height\_cm 0.6226199 4.647055e-12  
mentality\_interceptions 0.6147787 1.016163e-11  
defending\_marking\_awareness 0.6110181 1.467626e-11  
defending\_standing\_tackle 0.6102678 1.578390e-11  
weight\_kg 0.5443635 4.792828e-09  
power\_jumping 0.4843844 3.285453e-07  
mentality\_aggression 0.4630878 1.225685e-06  
skill\_ball\_control -0.2283309 2.232021e-02  
power\_shot\_power -0.2829131 4.343500e-03  
skill\_dribbling -0.3038003 2.120761e-03  
pace -0.5136892 4.610158e-08  
movement\_acceleration -0.5320392 1.222851e-08  
passing -0.5840214 1.792800e-10  
shooting -0.6751369 1.326081e-14  
movement\_agility -0.6916079 1.634877e-15  
dribbling -0.8601548 2.059502e-30  
  
Link between the variable and the categorical variable (1-way anova)  
=============================================  
 R2 p.value  
body\_type 0.1455218 0.0379278  
  
Link between variable abd the categories of the categorical variables  
================================================================  
 Estimate p.value  
body\_type=Normal (185+) 2.349482 0.003283813

## REPRÉSENTATION SIMULTANÉE DES DEUX PREMIERS AXES

Le graphique suivant s’appelle le cercle de corrélation. On représente simultanément deux dimensions (axes). On lit ce graphique de gauche à droite puis de bas en haut. Les axes représentent des corrélations. Si une variable pointe vers la gauche, c’est qu’elle est négativement corrélée à l’axe 1 et vice versa. Si une variable pointe vers le bas elle est négativement corrélée à l’axe 2 et vice versa. Ce graphique nous permet de voir quelles variables s’opposent entre elles sur chaque axe.

plot.PCA(premiere\_acp, axes = c(1, 2), choix = "var")



## LES INDIVIDUS

## Afficher les 10 premiers individus ayant des coordonnées positives sur l’axe 1

sort(premiere\_acp$ind$coord[, 1], decreasing = TRUE)[1:10]

N, Kanté A, Robertson J, Kimmich M, Verratti Jesús Navas   
 3.142613 2.788556 2.762806 2.734115 2.625595   
Marcos Llorente A, Hakimi Jordi Alba João Cancelo L, Modrić   
 2.512007 2.475914 2.436642 2.376720 2.372374

## Afficher les 10 premiers individus ayant des coordonnées négatives sur l’axe 1

sort(premiere\_acp$ind$coord[, 1], decreasing = FALSE)[1:10]

T, Courtois Alisson K, Casteels S, Handanovič P, Gulácsi   
 -7.868107 -7.561502 -7.535096 -7.521622 -7.422478   
 W, Szczęsny M, ter Stegen G, Donnarumma K, Schmeichel J, Oblak   
 -7.294438 -7.290675 -7.129166 -7.103182 -6.991911

## Afficher les 10 premiers individus ayant des coordonnées positives sur l’axe 2

sort(premiere\_acp$ind$coord[, 2], decreasing = TRUE)[1:10]

V, van Dijk G, Chiellini K, Koulibaly M, Hummels Piqué M, de Ligt   
 3.910655 3.882630 3.829444 3.609459 3.567733 3.427404   
 A, Laporte L, Bonucci M, Škriniar Rúben Dias   
 3.396182 3.384239 3.297347 3.295519

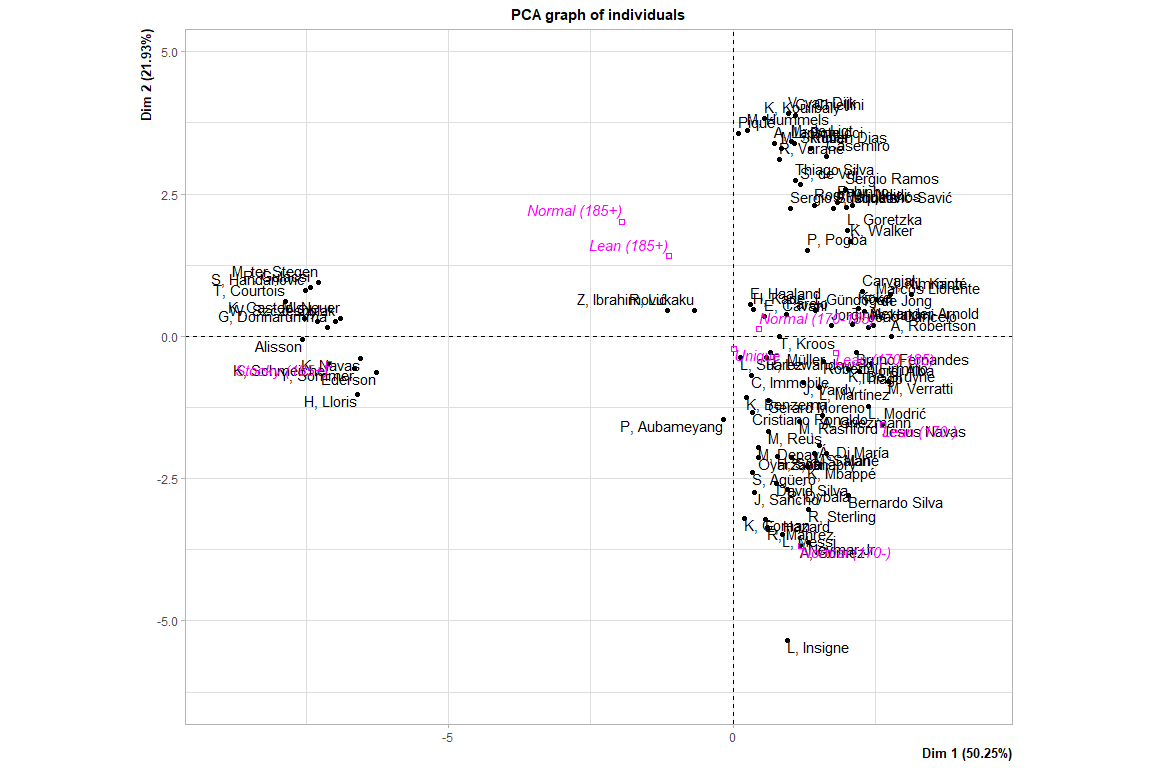
## Afficher les 10 premiers individus ayant des coordonnées négatives sur l’axe 2

sort(premiere\_acp$ind$coord[, 2], decreasing = FALSE)[1:10]

L, Insigne A, Gómez Neymar Jr L, Messi R, Mahrez   
 -5.338622 -3.677799 -3.623794 -3.479110 -3.359841   
 E, Hazard K, Coman R, Sterling Bernardo Silva J, Sancho   
 -3.209868 -3.203207 -3.036538 -2.791582 -2.749813

## Représenter les individus sur les deux axes

plot.PCA(premiere\_acp, axes = c(1, 2), choix = "ind")

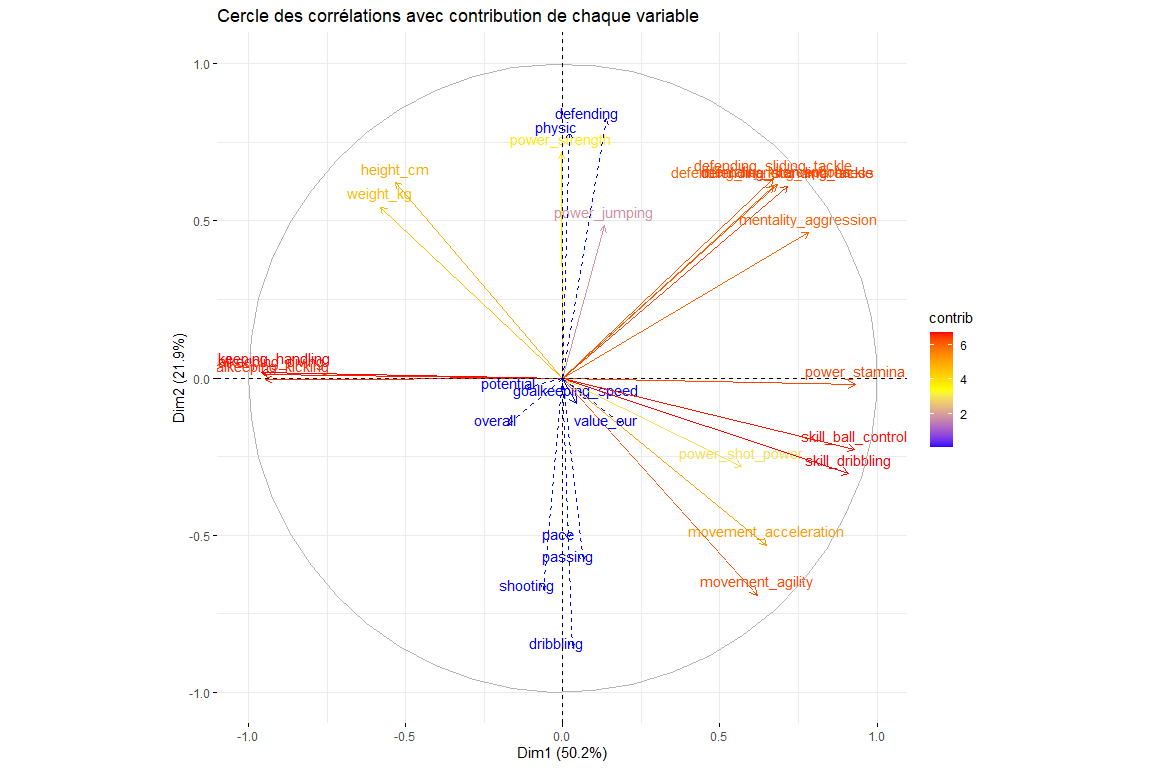


Les composantes 1 et 2 arrivent à faire une grande distinction entre les individus du jeu de données.

`

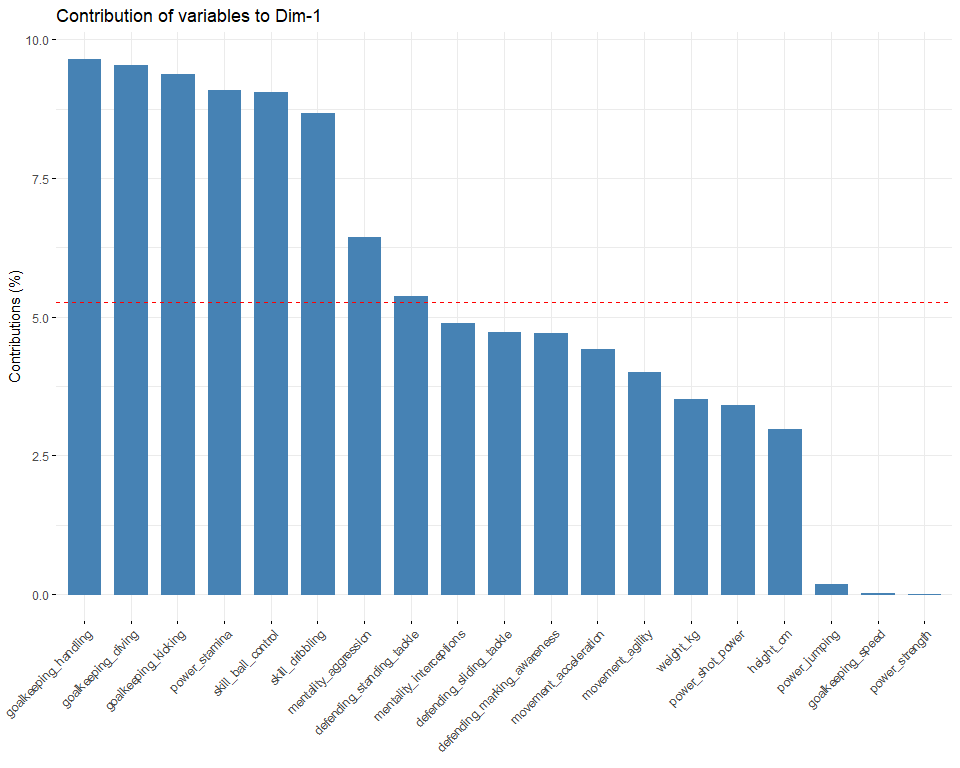
### 2 Représenter le cercle de corrélation puis mettre en avant la contribution de chaque variable

fviz\_pca\_var(premiere\_acp, col.var="contrib", axes = c(1,2), gradient.cols = c("blue", "yellow", "red"), title = "Cercle des corrélations avec contribution de chaque variable")

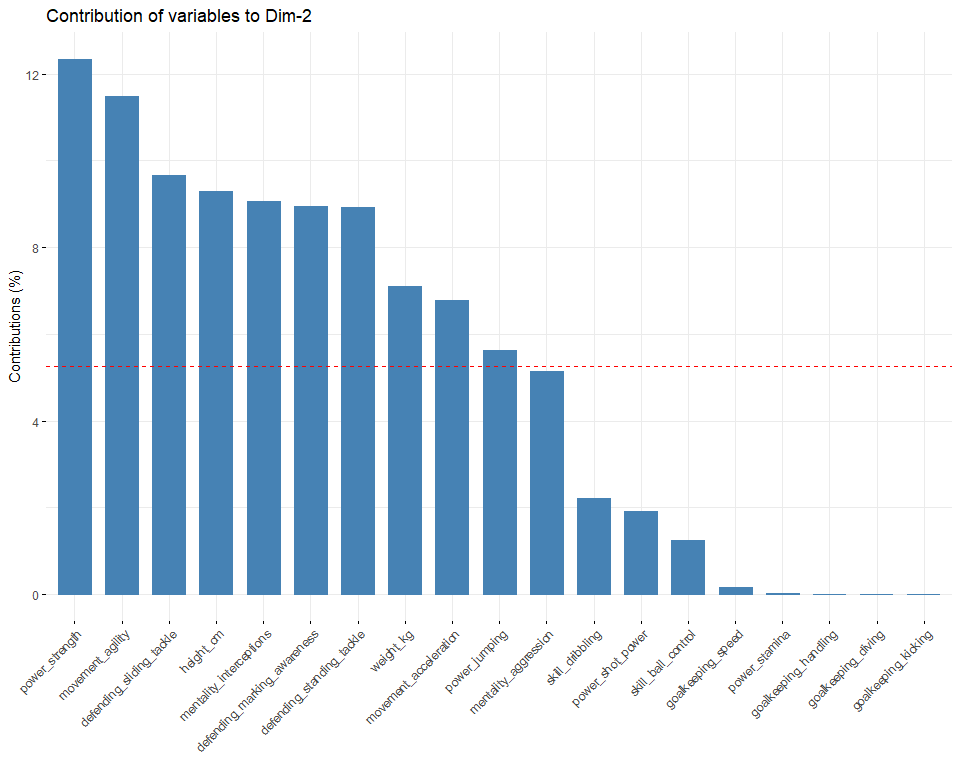


### Représenter le % de contribution de chaque variable sur les axes 1 et 2

fviz\_contrib(premiere\_acp, choice = "var", axes = 1)



fviz\_contrib(premiere\_acp, choice = "var", axes = 2)



# CLASSIFICATION ASCENDANTE HIERARCHIQUE

## CLASSER LES JOUEURS DANS 5 CLUSTERS

On a un jeu de données de 100 joueurs, comment peut-on classer les joueurs en 5 groupes homogènes ? On utilise la fonction HCPC du package FactoMineR, en précisant en argument l’objet acp qu’on a créé avec la fonction PCA, puis on précise le nombre de clusters (groupes, classes) qu’on veut produire. L’argument graph = FALSE instruit R de ne représenter les résultats graphiquement.

classif <- HCPC(premiere\_acp, nb.clust = 5, graph = FALSE)

## QUELLE(S) COMPOSANTE(S) CARACTÉRISENT LES PLUS LES INDIVIDUS DE CHAQUE CLUSTER ?

On peut utiliser une propriété de l’objet classif pour afficher la description de chaque cluster.

classif$desc.axes

Link between the cluster variable and the quantitative variables  
================================================================  
 Eta2 P-value  
Dim.1 0.9710515 3.985630e-72  
Dim.2 0.8407178 5.188359e-37  
Dim.3 0.6293291 1.039484e-19  
  
Description of each cluster by quantitative variables  
=====================================================  
$`1`  
 v.test Mean in category Overall mean sd in category Overall sd  
Dim.1 -9.61829 -7.11018 6.931955e-16 0.4384079 3.089844  
 p.value  
Dim.1 6.693444e-22  
  
$`2`  
 v.test Mean in category Overall mean sd in category Overall sd  
Dim.3 6.592117 2.455677 4.058733e-16 0.7196673 1.368715  
 p.value  
Dim.3 4.335989e-11  
  
$`3`  
 v.test Mean in category Overall mean sd in category Overall sd  
Dim.1 2.063221 1.002524 6.931955e-16 0.4640714 3.089844  
Dim.2 -7.184421 -2.306366 -1.407095e-15 1.0784803 2.041380  
 p.value  
Dim.1 3.909160e-02  
Dim.2 6.749241e-13  
  
$`4`  
 v.test Mean in category Overall mean sd in category Overall sd  
Dim.2 7.771677 2.917443 -1.407095e-15 0.6613834 2.041380  
Dim.1 2.214230 1.258123 6.931955e-16 0.5733747 3.089844  
 p.value  
Dim.2 7.745396e-15  
Dim.1 2.681297e-02  
  
$`5`  
 v.test Mean in category Overall mean sd in category Overall sd  
Dim.1 3.673506 2.212603 6.931955e-16 0.5615386 3.089844  
Dim.3 -5.086956 -1.357242 4.058733e-16 0.8761187 1.368715  
 p.value  
Dim.1 2.39245e-04  
Dim.3 3.63857e-07

Ces résultats nous permettent de savoir quelle(s) composante(s) caractérisent les mieux les individus qui appartiennent à un cluster. On a créé 5 clusters, le tableau affiche donc les caractéristiques de chaque cluster.

v.test: c’est une statistique calculée qui permet de tester la significativité du lien du cluster avec le(s) composante(s). Si sa valeur est supérieure à 1.96 c’est que le lien entre le cluster et l’axe est significatif. Il y a un lien entre v.test et la p.value. Plus la v.test est grande moins la p.value sera. Une p.value inférieure ou égale à 0.05 indique que le test statistique est significatif.

Le signe de la v.test Si positif : les individus qui sont dans le cluster ont en moyenne une coordonnée positive sur l’axe considérée

Si négatif : les individus qui sont dans le cluster ont en moyenne une coordonnée négative sur l’axe considérée.

Interprétation résultat cluster 1 Les individus qui sont dans le cluster sont caractérisés essentiellement par la dimension 1. Les joueurs qui sont classés dans ce clusters ont en moyenne une coordonnée égale à -7.11018 tandis que dans le reste du jeu de données les joueurs ont en moyenne une coordonnée de 6.931955e-16 (proche de 0) sur cette dimension.

## QUELS INDIVIDUS SONT LES PLUS REPRÉSENTATIFS DANS CHAQUE CLUSTER ?

### Les individus parangon

Le but de la classification ascendante hiérarchique c’est de regrouper les individus qui se ressemblent le plus (minimiser la variance intra cluster) et faire en sorte que chaque cluster diffère d’un autre (maximiser la variance entre chaque cluster).

Les individus parangon sont les individus qui se rapprochent le plus du centre de chaque cluster. Ce sont les individus “moyens” de chaque cluster. C’est eux qui caractérisent le mieux le cluster étudié.

classif$desc.ind$para

Cluster: 1  
G, Donnarumma W, Szczęsny J, Oblak Y, Sommer Alisson   
 0.7437392 0.7764569 0.8745005 1.1648592 1.7734468   
------------------------------------------------------------   
Cluster: 2  
R, Lewandowski H, Kane C, Immobile L, Suárez K, Benzema   
 0.7057555 0.7562182 0.8249288 0.9437102 0.9907515   
------------------------------------------------------------   
Cluster: 3  
 S, Gnabry M, Salah P, Dybala Oyarzabal Á, Di María   
 0.5125072 0.5198456 0.5664524 0.6265379 0.7321610   
------------------------------------------------------------   
Cluster: 4  
 Rúben Dias L, Bonucci Thiago Silva A, Laporte S, de Vrij   
 0.6053112 0.7313895 0.7423503 0.8298838 0.8657299   
------------------------------------------------------------   
Cluster: 5  
T, Alexander-Arnold João Cancelo A, Robertson F, de Jong   
 0.4016233 0.5214053 0.8291440 0.9979184   
 Bruno Fernandes   
 1.0795893

### QUELS INDIVIDUS SONT LES MOINS REPRÉSENTATIFS DANS CHAQUE CLUSTER ?

## Les individus spécifiques

Ce sont les individus les plus éloignés du centre du cluster. On aurait pu les mettre dans un autre cluster. Ils sont à la frontière de plusieurs clusters.

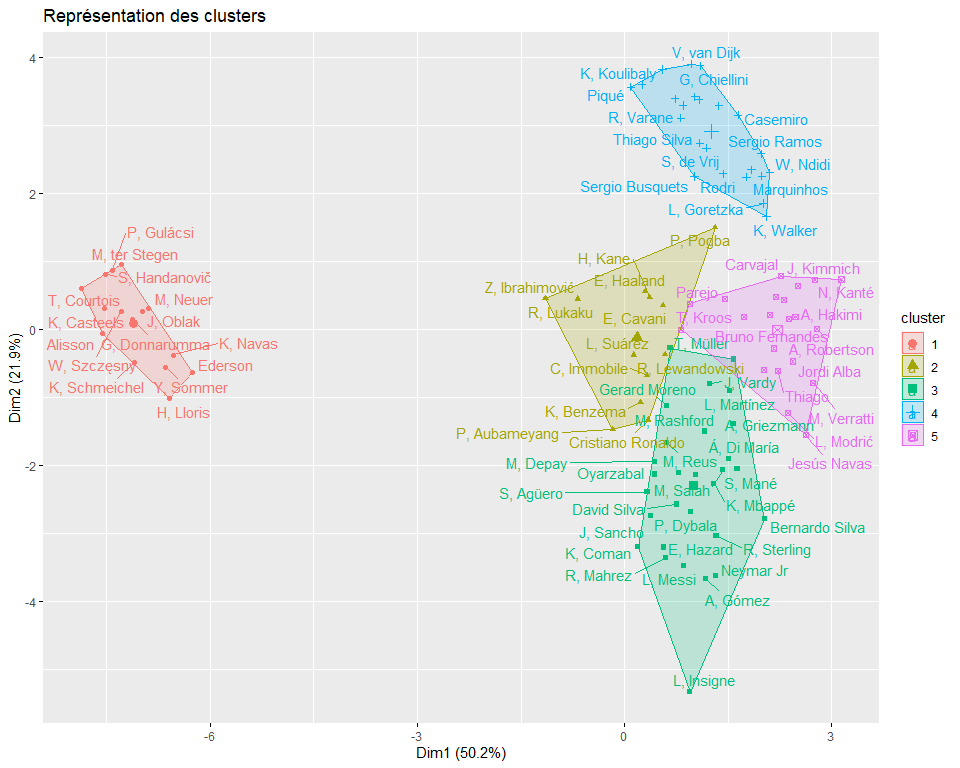
classif$desc.ind$dist

Cluster: 1  
 H, Lloris Ederson P, Gulácsi T, Courtois K, Casteels   
 9.402346 9.207166 8.921628 8.706093 8.660642   
------------------------------------------------------------   
Cluster: 2  
 Z, Ibrahimović R, Lukaku E, Haaland H, Kane   
 4.787525 4.584261 4.415862 3.669592   
Cristiano Ronaldo   
 3.280196   
------------------------------------------------------------   
Cluster: 3  
L, Insigne A, Gómez E, Hazard Neymar Jr R, Mahrez   
 5.575780 3.977211 3.952106 3.924649 3.877662   
------------------------------------------------------------   
Cluster: 4  
 M, Hummels G, Chiellini K, Koulibaly V, van Dijk Piqué   
 4.506813 4.489410 4.364257 4.351577 4.284480   
------------------------------------------------------------   
Cluster: 5  
 T, Kroos M, Verratti Parejo Jesús Navas A, Robertson   
 4.947775 4.490787 3.922122 3.917639 3.701767

factoextra::fviz\_cluster(classif, repel = TRUE, title = "Représentation des clusters")

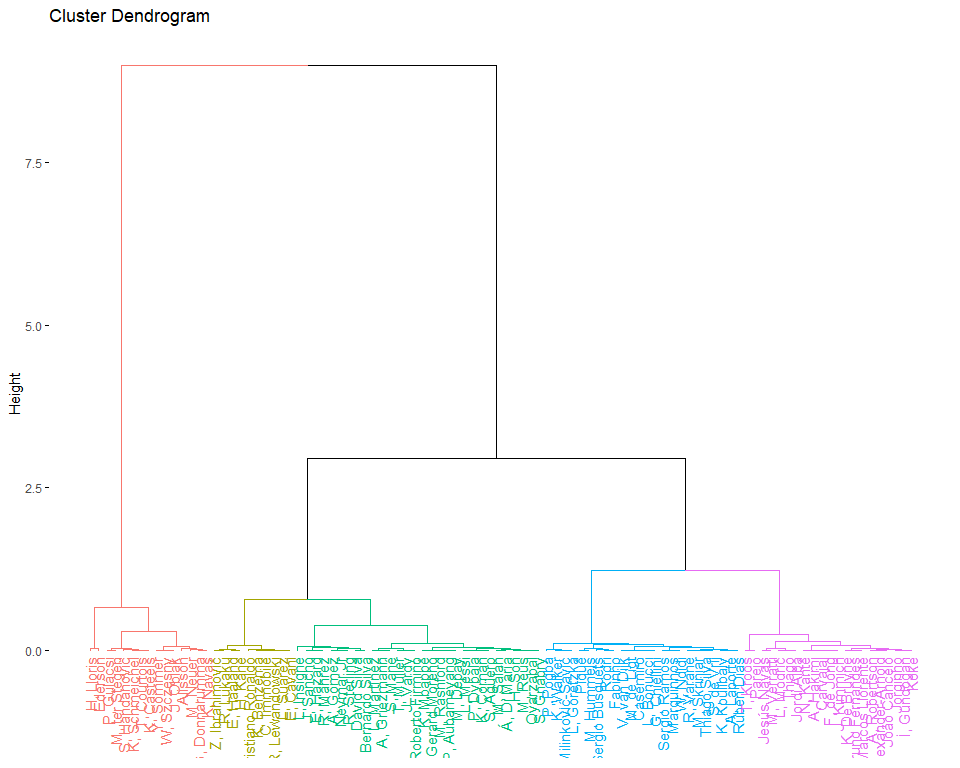
Warning: argument title is deprecated; please use main instead.

Warning: ggrepel: 19 unlabeled data points (too many overlaps). Consider  
increasing max.overlaps



fviz\_dend(classif)

Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
of ggplot2 3.3.4.  
ℹ The deprecated feature was likely used in the factoextra package.  
 Please report the issue at <]8;;https://github.com/kassambara/factoextra/issueshttps://github.com/kassambara/factoextra/issues]8;;>.



classif2 <- HCPC(premiere\_acp, graph = FALSE)

fviz\_cluster(classif2)

