

Data Extraction and Processing

Project: Olympics Data

Roll No.: N051 N052 N055 N060	Name: Priyal Kadam Akshay Patil Ritanshu Gupta Yash Dukhande
Prog/Yr/Sem: M.B.A. Tech CE (D)	Batch: D2

INTRODUCTION

We have taken a dataset of football players from the game FIFA 23 . As we surf through the report we can take a look at how we used R programming for data importing , data structure , data analysis , cleaning data etc .

PROBLEM STATEMENT

Perform a comprehensive analysis of FIFA player data to gain deep insights into player performance, attribute rankings, and the factors influencing player success. This analysis will hope to provide a holistic understanding of the football players ecosystem and answer critical questions for fans, clubs, and enthusiasts.

Packages used

Code:

```
library(tidyverse)
library(magrittr)
library(DataExplorer)
library(maps)
library(plotly)
library(DT)
library(tidytext)
library(gridExtra)
library(factoextra)
```

Data Description

Data Importing

Link : <https://www.kaggle.com/datasets/javagarm/fifa-19-complete-player-dataset>

Code:

```
df=read.csv("fifa.csv")
```

```
head(df)
```

Output:

```
> df=read.csv("fifa.csv")
> head(df)
```

	X	ID	Name	Age	Photo
1	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png
2	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png
3	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png
4	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png
5	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png
6	5	183277	E. Hazard	27	https://cdn.sofifa.org/players/4/19/183277.png

	Nationality	Flag	Overall	Potential
1	Argentina	https://cdn.sofifa.org/flags/52.png	94	94
2	Portugal	https://cdn.sofifa.org/flags/38.png	94	94
3	Brazil	https://cdn.sofifa.org/flags/54.png	92	93
4	Spain	https://cdn.sofifa.org/flags/45.png	91	93
5	Belgium	https://cdn.sofifa.org/flags/7.png	91	92
6	Belgium	https://cdn.sofifa.org/flags/7.png	91	91

	Club	Club.Logo	Value	Wage
1	FC Barcelona	https://cdn.sofifa.org/teams/2/light/241.png	€110.5M	€565K
2	Juventus	https://cdn.sofifa.org/teams/2/light/45.png	€77M	€405K
3	Paris Saint-Germain	https://cdn.sofifa.org/teams/2/light/73.png	€118.5M	€290K
4	Manchester United	https://cdn.sofifa.org/teams/2/light/11.png	€72M	€260K
5	Manchester City	https://cdn.sofifa.org/teams/2/light/10.png	€102M	€355K
6	Chelsea	https://cdn.sofifa.org/teams/2/light/5.png	€93M	€340K

	Special	Preferred.Foot	International.Reputation	Weak.Foot	Skill.Moves
1	2202	Left	5	4	4
2	2228	Right	5	4	5
3	2143	Right	5	5	5
4	1471	Right	4	3	1
5	2281	Right	4	5	4
6	2142	Right	4	4	4

	Work.Rate	Body.Type	Real.Face	Position	Jersey.Number	Joined
1	Medium/ Medium	Messi	Yes	RF	10	Jul 1, 2004
2	High/ Low	C. Ronaldo	Yes	ST	7	Jul 10, 2018
3	High/ Medium	Neymar	Yes	LW	10	Aug 3, 2017
4	Medium/ Medium	De Gea	Yes	GV	1	Jul 1, 2011

Data Structure

Code:

```
dim(df)
```

Output:

```
> dim(df)
[1] 18207      89
\
```

There are 89 columns and 18207 rows

Code:

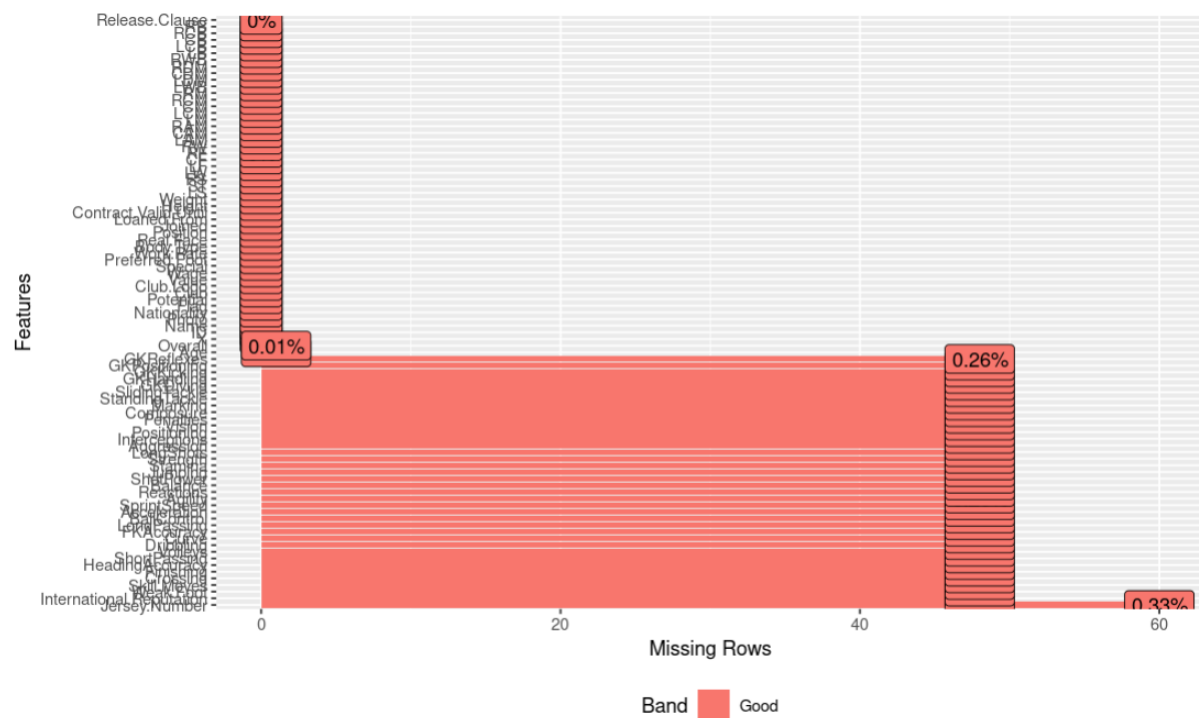
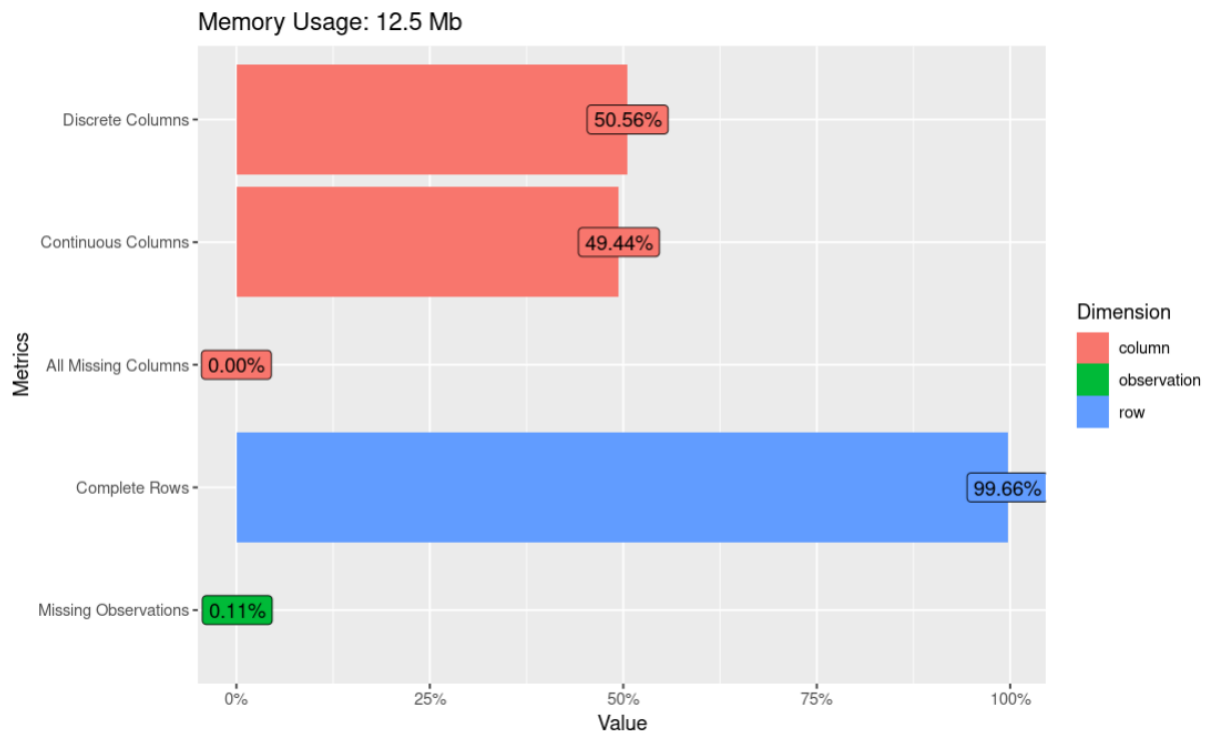
```
introduce(df)
```

```
plot_intro(df)
```

```
plot_missing(df)
```

Output:

```
> introduce(df)
rows columns discrete_columns continuous_columns all_missing_columns total_missing_values complete_rows total_observations memory_usage
1 18207      89              45                44              0              1838              18145              1620423              13138600
```



Data Manipulation

Creating League Variable and Sampling:

The data does not include league variable but we can extract leagues from Club variable.

Code:

```
bundesliga = c(
  "1. FC Nürnberg", "1. FSV Mainz 05", "Bayer 04 Leverkusen", "FC Bayern München",
  "Borussia Dortmund", "Borussia Mönchengladbach", "Eintracht Frankfurt",
  "FC Augsburg", "FC Schalke 04", "Fortuna Düsseldorf", "Hannover 96",
  "Hertha BSC", "RB Leipzig", "SC Freiburg", "TSG 1899 Hoffenheim",
  "VfB Stuttgart", "VfL Wolfsburg", "SV Werder Bremen"
)

premierLeague = c(
  "Arsenal", "Bournemouth", "Brighton & Hove Albion", "Burnley",
```

```

"Cardiff City", "Chelsea", "Crystal Palace", "Everton", "Fulham",
"Huddersfield Town", "Leicester City", "Liverpool", "Manchester City",
"Manchester United", "Newcastle United", "Southampton",
"Tottenham Hotspur", "Watford", "West Ham United", "Wolverhampton Wanderers"
)
laliga = c(
"Athletic Club de Bilbao", "Atlético Madrid", "CD Leganés",
"Deportivo Alavés", "FC Barcelona", "Getafe CF", "Girona FC",
"Levante UD", "Rayo Vallecano", "RC Celta", "RCD Espanyol",
"Real Betis", "Real Madrid", "Real Sociedad", "Real Valladolid CF",
"SD Eibar", "SD Huesca", "Sevilla FC", "Valencia CF", "Villarreal CF"
)
seriea = c(
"Atalanta", "Bologna", "Cagliari", "Chievo Verona", "Empoli", "Fiorentina", "Frosinone", "Genoa",
"Inter", "Juventus", "Lazio", "Milan", "Napoli", "Parma", "Roma", "Sampdoria", "Sassuolo", "SPAL",
"Torino", "Udinese"
)
superlig = c(
"Akhisar Belediyespor", "Alanyaspor", "Antalyaspor", "Medipol Basaksehir FK", "BB
Erzurumspor", "Besiktas JK",
"Bursaspor", "Çaykur Rizespor", "Fenerbahçe SK", "Galatasaray SK", "Göztepe
SK", "Kasimpasa SK",
"Kayserispor", "Atiker Konyaspor", "MKE Ankaragücü", "Sivasspor", "Trabzonspor", "Yeni
Malatyaspor"
)
ligue1 = c(
"Amiens SC", "Angers SCO", "AS Monaco", "AS Saint-Étienne", "Dijon FCO", "En Avant de
Guingamp",
"FC Nantes", "FC Girondins de Bordeaux", "LOSC Lille", "Montpellier HSC", "Nîmes
Olympique",
"OGC Nice", "Olympique Lyonnais", "Olympique de Marseille", "Paris Saint-Germain",
"RC Strasbourg Alsace", "Stade Malherbe Caen", "Stade de Reims", "Stade Rennais FC",
"Toulouse Football Club"
)
eredivisie = c(
"ADO Den Haag", "Ajax", "AZ Alkmaar", "De Graafschap", "Excelsior", "FC Emmen", "FC
Groningen",
"FC Utrecht", "Feyenoord", "Fortuna Sittard", "Heracles Almelo", "NAC Breda",
"PEC Zwolle", "PSV", "SC Heerenveen", "Vitesse", "VVV-Venlo", "Willem II"
)
liganos = c(
"Os Belenenses", "Boavista FC", "CD Feirense", "CD Tondela", "CD Aves", "FC Porto",
"CD Nacional", "GD Chaves", "Clube Sport Marítimo", "Moreirense FC", "Portimonense
SC", "Rio Ave FC",
"Santa Clara", "SC Braga", "SL Benfica", "Sporting CP", "Vitória Guimarães", "Vitória de
Setúbal"
)
df$League = NA
df$Country = NA
df$League[df$Club %in% bundesliga] = "Bundesliga"

```

```

df$League[df$Club %in% premierLeague] = "Premier League"
df$League[df$Club %in% laliga] = "La Liga"
df$League[df$Club %in% seriea] = "Serie A"
df$League[df$Club %in% superlig] = "Süper Lig"
df$League[df$Club %in% ligue1] = "Ligue 1"
df$League[df$Club %in% liganos] = "Liga Nos"
df$League[df$Club %in% eredivisie] = "Eredivisie"
df$Country[df$League == "Bundesliga"] = "Germany"
df$Country[df$League == "Premier League"] = "UK"
df$Country[df$League == "La Liga"] = "Spain"
df$Country[df$League == "Serie A"] = "Italy"
df$Country[df$League == "Süper Lig"] = "Turkey"
df$Country[df$League == "Ligue 1"] = "France"
df$Country[df$League == "Liga Nos"] = "Portugal"
df$Country[df$League == "Eredivisie"] = "Netherlands"
df$League = as.character(df$League)
df$Country = as.character(df$Country)

```

String manipulation:

Value and Wage variables has described as discrete variables. We should transform them into continuous variable.

Code:

```
head(df$Value)
```

Output:

```

> head(df$Value)
[1] "€110.5M" "€77M"      "€118.5M" "€72M"      "€102M"      "€93M"

```

Code:

```

df$Values = str_remove_all(df$Value,"€")
df$Values = str_replace_all(df$Values,"K", "000")
df$Values = str_remove_all(df$Values,"M")
df$Values = as.numeric(df$Values)
df$Wages = str_remove_all(df$Wage,"€")
df$Wages = str_replace_all(df$Wages,"K", "000")
df$Wages = as.numeric(df$Wages)
df$Values = ifelse(df$Values < 1000, df$Values * 1000000, df$Values)

```

Create Position Class:

Every players has a position on the football pitch. We can create Position Class variable by using Position information.

Code:

```
unique(df$Position)
```

Output:

```

> unique(df$Position)
[1] "RF" "ST" "LW" "GK" "RCM" "LF" "RS" "RCB" "LCM" "CB" "LDM" "CAM" "CDM" "LS" "LCB" "RM" "LAM" "LM" "LB" "RDM" "RW" "CM" "RB" "RAM" "CF" "RWB"
[27] "LWB" ""
>

```

Code:

```

defence <- c("CB", "RB", "LB", "LWB", "RWB", "LCB", "RCB")
midfielder <- c("CM", "CDM", "CAM", "LM", "RM", "LAM", "RAM", "LCM", "RCM", "LDM", "RDM")
df$Class = ""
df$Class[df$Position %in% "GK"] = "Goal Keeper"
df$Class[df$Position %in% defence] = "Defender"
df$Class[df$Position %in% midfielder] = "Midfielder"
df$Class[!df$Position %in% c("GK", defence, midfielder)] = "Forward"

```

```
rm(defence, midfielder)
```

Height and Weight:

Height and Weight variables convert cm and kg units.

Code:

```
df$Height = round((as.numeric(substr(df$Height, start = 1, stop = 1)) * 30.48) +  
(as.numeric(substr(df$Height, start = 3, stop = 5)) * 2.54))  
df$Weight = round(as.numeric(substr(df$Weight, start = 1, stop = 3)) / 2.204623)
```

Correction of Preferred foot variable:

Code:

```
foot_filter = df$Preferred.Foot %in% c("Left", "Right")  
df = df[foot_filter, ]  
df$Preferred.Foot = as.factor(as.character(df$Preferred.Foot))
```

Rename some variables:

Code:

```
df %<>%  
  rename(  
    "Heading.Accuracy"= HeadingAccuracy,  
    "Short.Passing"= ShortPassing,  
    "FK.Accuracy" = FKAccuracy,  
    "Long.Passing"= LongPassing,  
    "Ball.Control"= BallControl,  
    "Sprint.Speed"= SprintSpeed,  
    "Shot.Power"= ShotPower,  
    "Long.Shots"= LongShots,  
    "Standing.Tackle"= StandingTackle,  
    "Sliding.Tackle"= SlidingTackle,  
    "GK.Diving"= GKDiving,  
    "GK.Handling"= GKHandling,  
    "GK.Kicking"= GKKicking,  
    "GK.Positioning"= GKPositioning,  
    "GK.Reflexes"= GKReflexes  
  )
```

Remove Unnecessary Variables:

Code:

```
df = df[, !names(df) %in% c("ID", "Body.Type", "Real.Face", "Joined", "Loaned.From",  
"Release.Clause", "Photo", "Flag", "Special", "Work.Rate")]
```

Tidying Data

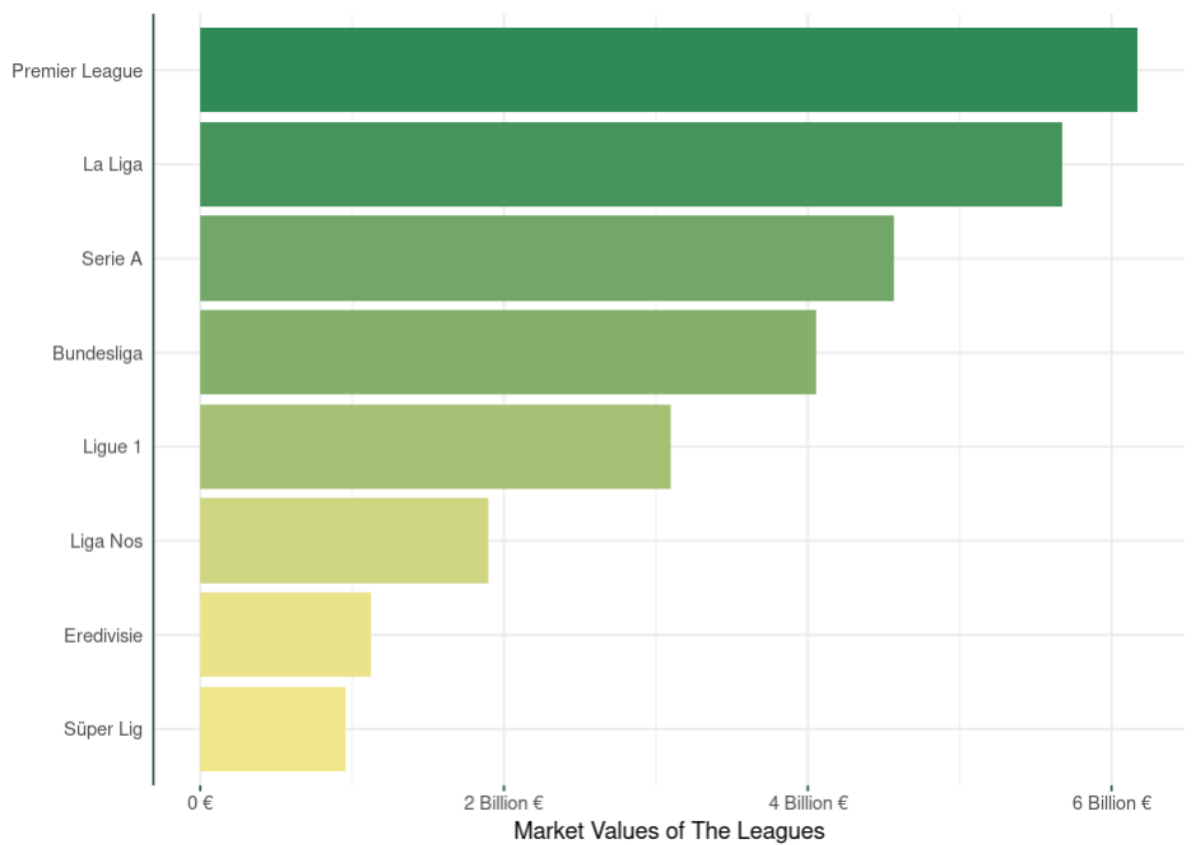
Code:

```
df=na.omit(df)  
introduce(df)  
plot_missing(df)
```

Output:

```
> df=na.omit(df)  
> introduce(df)  
  rows columns discrete_columns continuous_columns all_missing_columns total_missing_values complete_rows total_observations memory_usage  
1 4333      84              38                46                  0                  0         4333         363972         2645288
```


Output:



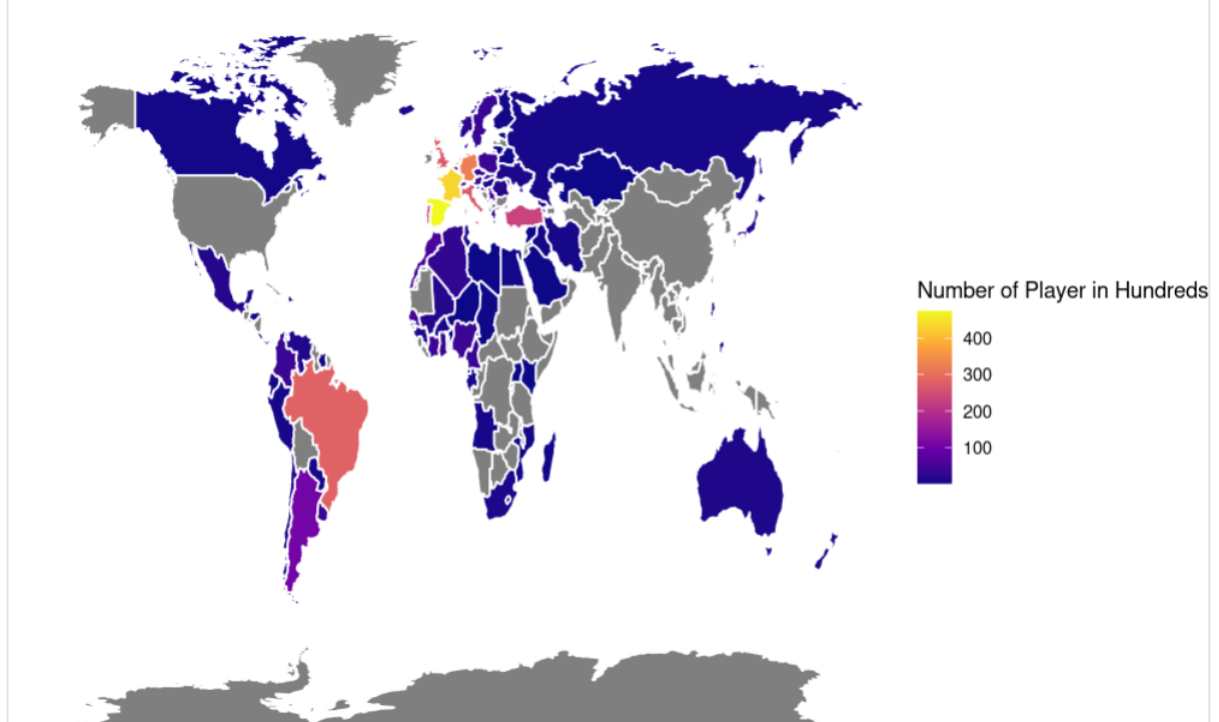
Interactive world map and number of players

Code:

```
options(repr.plot.width = 12, repr.plot.height = 8)
world_map = map_data("world")
numofplayers = world_map %>%
  mutate(region = as.character(region)) %>%
  left_join((df %>% mutate(Nationality = as.character(Nationality),
    Nationality = if_else(Nationality %in% "England",
      "UK", Nationality)) %>%
    #filter(League == "Bundesliga") %>%
    count(Nationality, name = "Number of Player") %>%
    rename(region = Nationality) %>%
    mutate(region = as.character(region))), by = "region")
ggplot(numofplayers, aes(long, lat, group = group))+
  geom_polygon(aes(fill = `Number of Player` ), color = "white", show.legend = TRUE)+
  scale_fill_viridis_c(option = "C")+
  theme_void()+
  labs(fill = "Number of Player in Hundreds",
    title = "Number of Player with ggplot2")
```


Output:

Number of Player with ggplot2

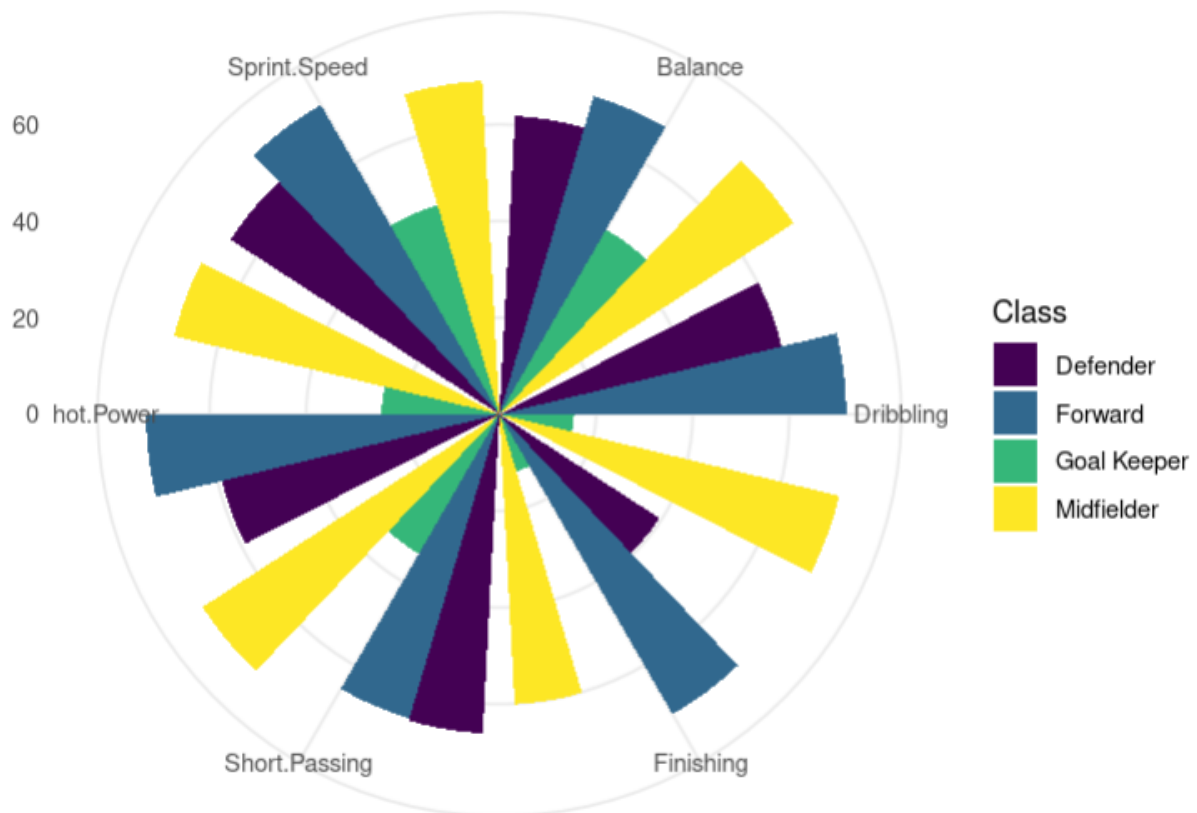


Average summary statistics of players by position class in the Premier League

Code:

```
options(repr.plot.width = 12, repr.plot.height = 8)
df %>%
  filter(League == "Premier League") %>%
  select(Class, Sprint.Speed, Dribbling, Shot.Power, Finishing, Balance, Short.Passing) %>%
  group_by(Class) %>%
  summarise_at(vars(Sprint.Speed:Short.Passing), funs(mean)) %>%
  gather(variables, values, -Class) %>%
  ggplot(aes(variables, values, fill = Class))+
  geom_col(position = "dodge")+
  coord_polar()+
  scale_fill_ordinal()+
  theme_minimal()+
  labs(x = NULL, y = NULL)
```

Output:



Correlation:

Code:

```
kor = df %>%  
  filter(League == "La Liga", Class == "Forward") %>%  
  select(Name, Preferred.Foot, Finishing, Shot.Power)  
cor.test(kor$Shot.Power, kor$Finishing, method = "pearson")  
cor.test(kor$Shot.Power, kor$Finishing, method = "kendall")  
hypo = cor.test(kor$Shot.Power, kor$Finishing, method = "spearman")  
hypo
```

Output:

Pearson's product-moment correlation

```
data: kor$Shot.Power and kor$Finishing
t = 12.023, df = 113, p-value < 0.00000000000000022
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.6560646 0.8198210
sample estimates:
      cor
0.749175
```

Kendall's rank correlation tau

```
data: kor$Shot.Power and kor$Finishing
z = 8.7156, p-value < 0.00000000000000022
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.5674854
```

```
Warning message in cor.test.default(kor$Shot.Power, kor$Finishing, method = "spearman"):
"Cannot compute exact p-value with ties"
```

Spearman's rank correlation rho

```
data: kor$Shot.Power and kor$Finishing
S = 64431, p-value < 0.00000000000000022
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.7457925
```