Football Data Analysis

Dor Kanfer

June 23, 2021

Lab 2: Visualization Through `ggplot`

Contents:

Q0.Submission Instructions

This lab will be submitted in pairs using GitHub (if you don't have a pair, please contact us).
Please follow the steps in the GitHub-Classroom Lab 2 (https://classroom.github.com/g/6_Wy5z44) to create your group's Lab 2 repository.
Important: your team's name must be `FamilyName1_Name1_and_FamilyName2_Name2` .
You can collaborate with your partner using the git environment; You can either make commits straight to master, or create individual branches (recommended). However, once done, be sure to merge your branches to master - you will be graded using the most recent *master* version - your last push and merge before the deadline.
Please do not open/review other peoples' repositories - we will be notified by GitHub if you do.

Your final push should include this Rmd file (with your answers) together with the html file that is outputted automatically by knitr when you knit the Rmd. Anything else will be disregarded. In addition, please adhere to the following file format:
`Lab_2_FamilyName1_Name1_and_FamilyName2_Name2.Rmd/html`

Some questions may require data wrangling and manipulation which you need to decide on.
In some graphs you may need to change the graph limits. If you do so, please include the outlier points you have removed in a separate table.

Show numbers in plots/tables using standard digits and not scientific display. That is: 90000000 and not 9e+06.
Round numbers to at most 3 digits after the dot - that is, 9.456 and not 9.45581451044

The required libraries are listed below the instructions. You are allowed to add additional libraries if you want. If you do so, *please explain what libraries you've added, and what is each new library used for*.

Background:

You've been hired as a data analyst at at football (soccer) club. Since this is a small and under-funded club, you will not have access to real-football data, but to data from the football computer game fifa18. Your job is to analyze this dataset and extract meaningful insights from the data in order to help your club make better decisions.

Data File:

You will load and analyze the fifa18 football dataset file called “fifa_data.csv”.
The dataset contains detailed information about each player in the game, including: names, age, nationality, overall ability, estimated potential ability, current club and league, market value, salary (wage), ability at different football skills (also called ‘attributes’, e.g. Ball.control, Sprint.speed …), ability to play at different position in the game (CF, CM, …) and the preferred positions of the player.

Required Libraries:

```
library(ggplot2)
library(ggcorrplot)
library(dplyr)
library(corrplot)
library(scales)   # needed for formatting y-axis labels to non-scientific type
library(radarchart)
library(tidyr)
library(tidyverse)
library(reshape2) # melt
library(ggthemes)
library(rworldmap) # world map
library(modelr)
library(radarchart) #Spider chart
##############################################
library(e1071) #Q1.c -  skewness() and kurtosis()
library(grid) # geom_segment
library(ggrepel)# Use ggrepel::geom_label_repel
library(fmsb) # for spider chart
library(colormap) # for spider chart's colors
library(kableExtra) # library for cosmetic display
knitr::opts_chunk$set(error = TRUE)

options("scipen"=100, "digits"=4)  # avoid scientific display of digits. Take 4 digits.
```

Q1. Basic Univariate Statistics (30 pt)

First, you are requested to load the fifa18 dataset and find and display general information about the players.

a. Make a plot showing the `overall` ability distribution of all players. How skewed is the distributions? does it have fat tails?
   Plot on top of the `overall` distribution a Normal distribution matching its first two moments. Is the distribution described well by a Normal distribution? explain.

b. Make a plot comparing the multiple `overall` ability *distributions* of players according to the `continent` of the players. Describe which continents have especially good/bad players.

c. Make a plot showing the density of players’ `value` distribution.
   Next, make a separate plot showing the density distribution of the *log* of players’ `value` .
   Which of the two visualizations is better? explain.

d. Are the top-10 players with the highest `value` also the top-10 best players in terms of `overall` ability? Show tables for both and compare.
   Who is the best player not in the top-10 valued players?

e. Show a table of the *10 youngest* and *10 oldest* teams in terms of *average* players `age` .

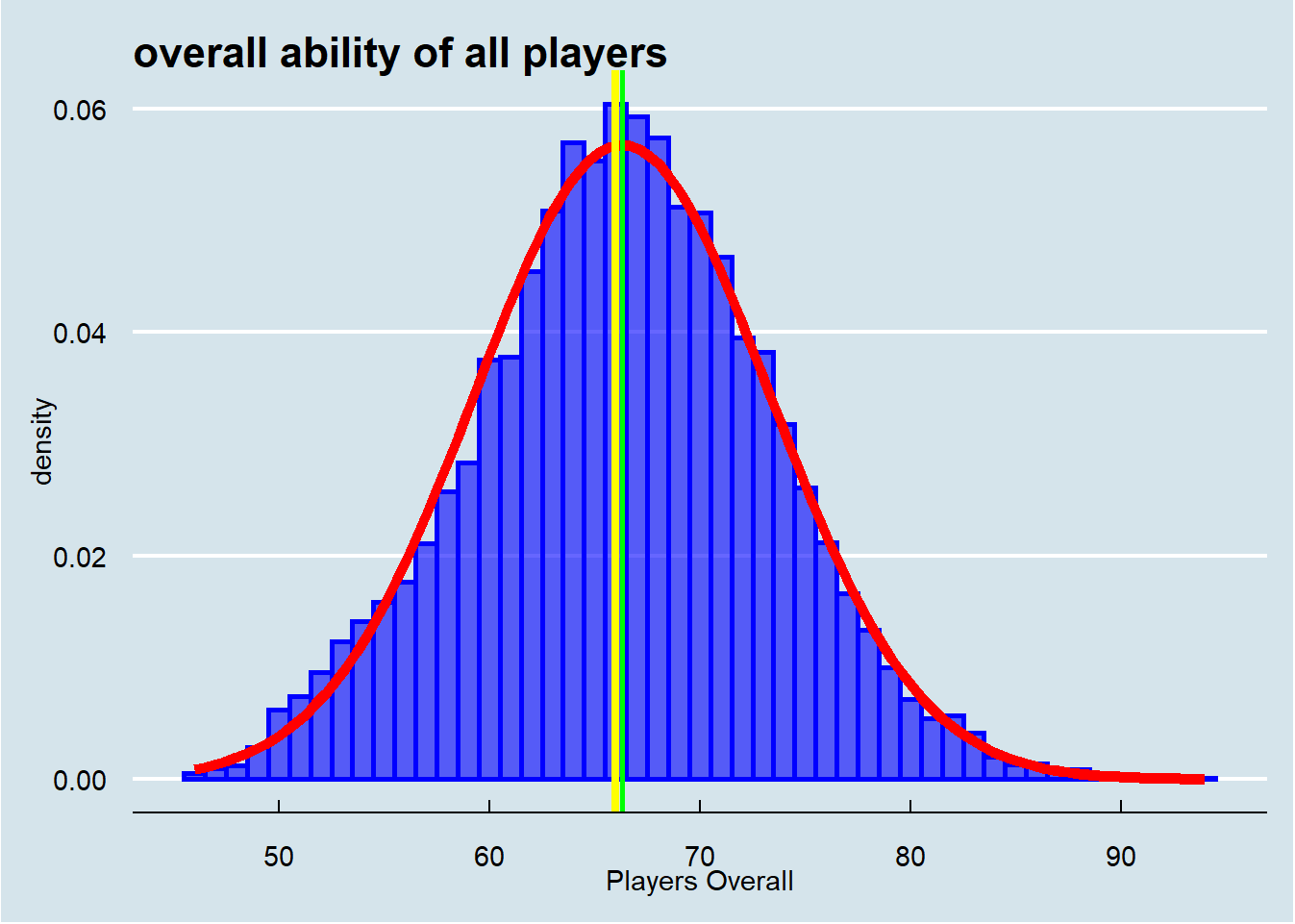
Loading the data:

```
# fifa_players <- data.frame(read.csv(url("https://raw.githubusercontent.com/DataScienceHU/DataAnalysisR_2020/master/data/fifa_data.csv")))
fifa_players <- data.frame(read.csv(url("https://raw.githubusercontent.com/DataScienceHU/DataAnalysisR_2020/master/data/fifa_data.csv")))

# Pre-processing:
for (i in c(3,6,7,10:71)) {
  fifa_players[,i]<-as.numeric((fifa_players[,i]))
}
fifa<-na.omit(fifa_players)
fifa_players <- fifa
fifa_players_info <- fifa[,c(1:11)] # players general info
fifa_players_attribures <- fifa[,c(1,12:45, 6)] # players different skills. Add overall
fifa_players_positions <- fifa[,c(1,46:72,6,7)] # players ability at different positions . Add overall
fifa_players_indicators <- fifa[,c(1,6,7,10,11)] # players general ability
```

solution:

1.a

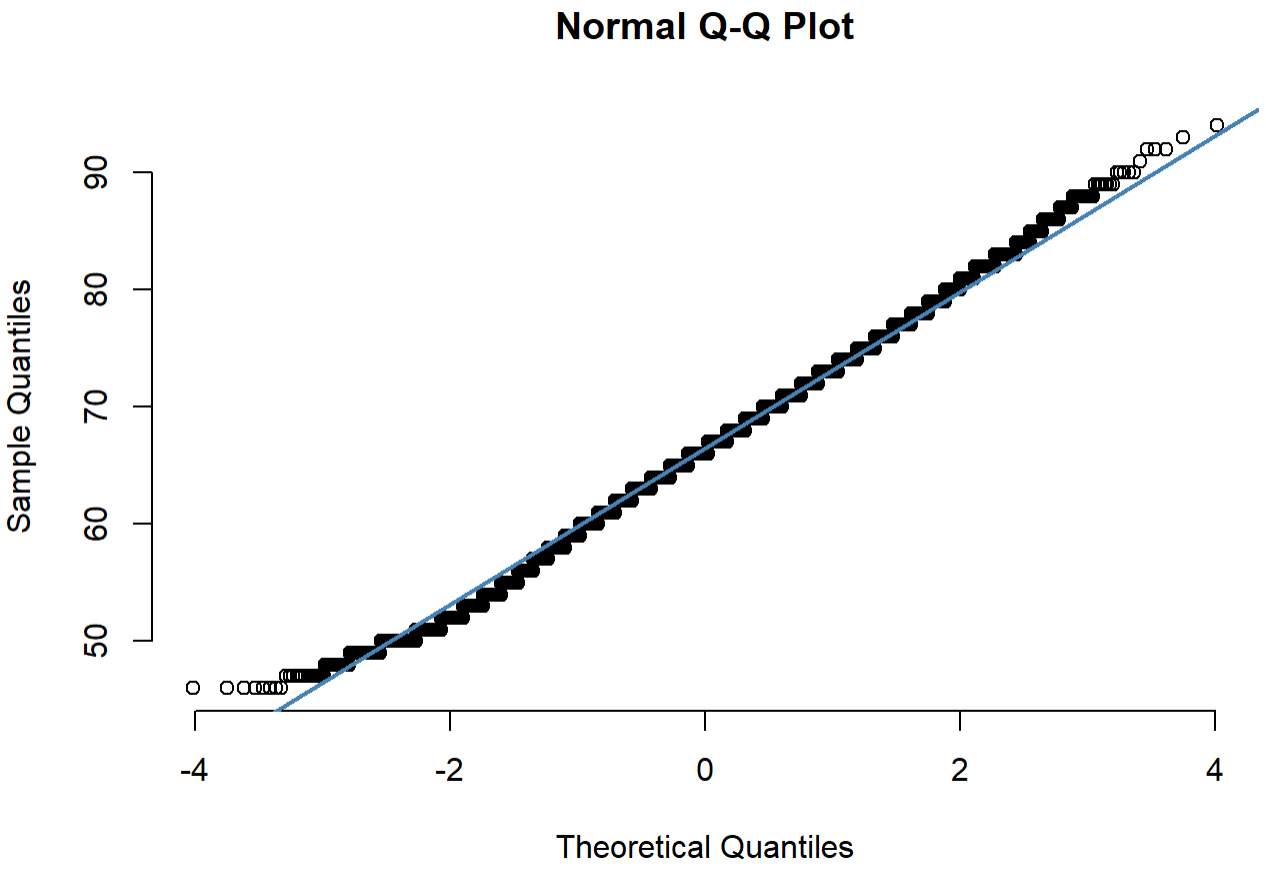
```
ggplot(data = fifa_players,aes(x = Overall))+
  geom_histogram(aes(y=..density..),col = "blue", fill = "blue", lwd = 0.9,bins = 49,
  alpha = 0.6)+ stat_function(fun = dnorm,args = list(mean = mean(fifa$Overall),sd = sd(fifa$Overall)),
      col = "red", lwd = 2)+ theme_economist()+
  labs(title = "overall ability of all players") +
  xlab("Players Overall")+ geom_vline(xintercept = median(fifa_players$Overall),col = "yellow",lwd = 1.5) +
  geom_vline(xintercept = mean(fifa_players$Overall),col = "green",lwd = 1)
```



the yellow vertical line is the median moment. the green vertical line is the mean moment

examine the normality of the plot with qqplot:

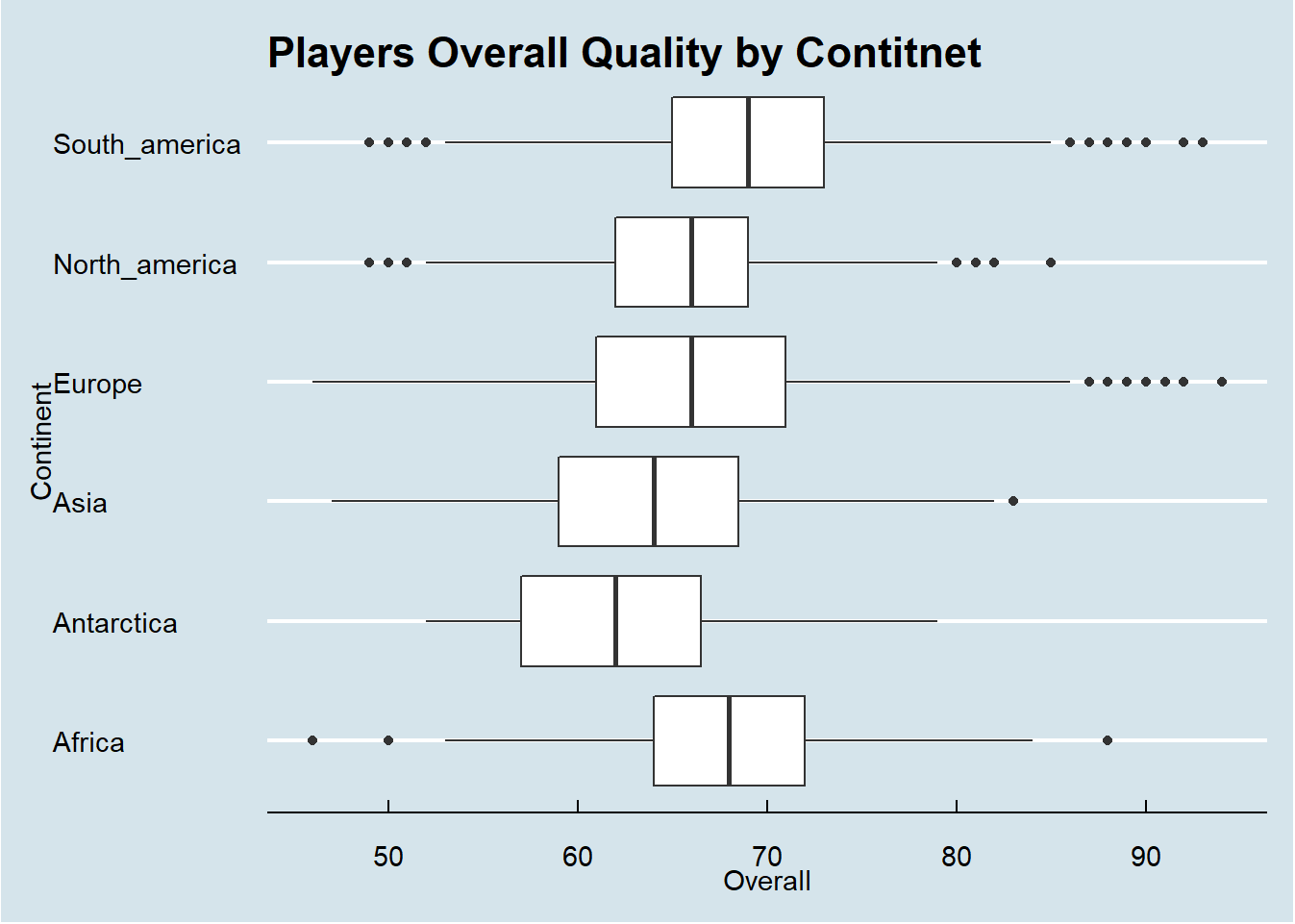
```
qqnorm(fifa$Overall, pch = 1, frame = FALSE)
qqline(fifa$Overall, col = "steelblue", lwd = 2)
```



as we can see, according to there histogram and qqplot graphs, there is a bit of a right tail but except this, our distribution described well by a Normal distribution.

1.b

```
agg2 =aggregate(Overall~ Continent,fifa_players,mean)
agg3 =aggregate(Overall~ Continent,fifa_players,var)

ggplot(fifa_players,aes(x=Continent,y = Overall))+
  geom_boxplot()+
  xlab("Continent")+
  ylab("Overall")+
  coord_flip()+ theme_economist()+
  labs(title="Players Overall Quality by Contitnet")
```



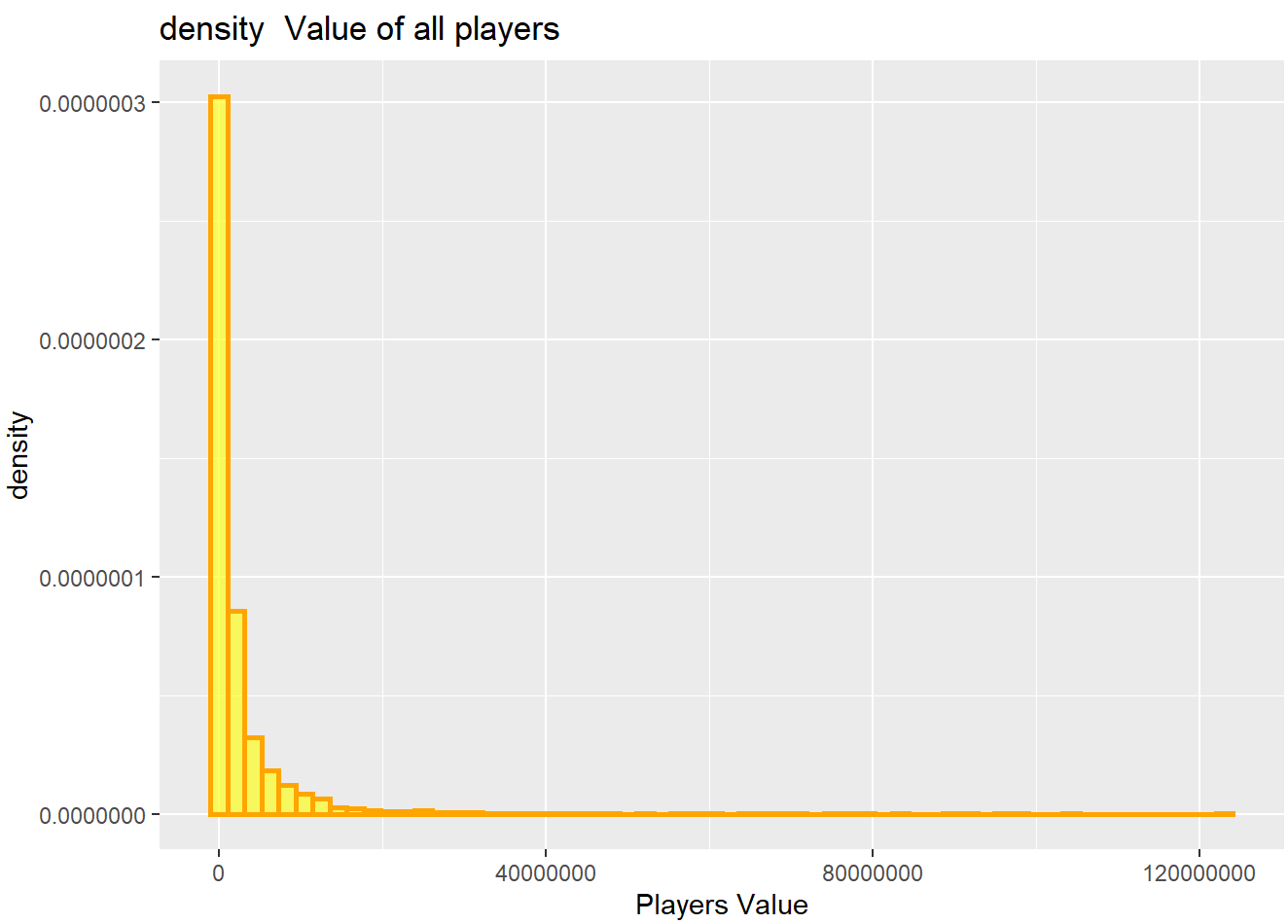

```
table = agg2[order(agg2$Overall),]
kable(table)
```

Continent	Overall
2Antarctica	62.15
3Asia	63.98
5North_america	65.48
4Europe	65.82
1Africa	67.89
6South_america	68.72

as we can see, Antarctica has the lowest mean value, and on the other hand - South America has the highest mean value.

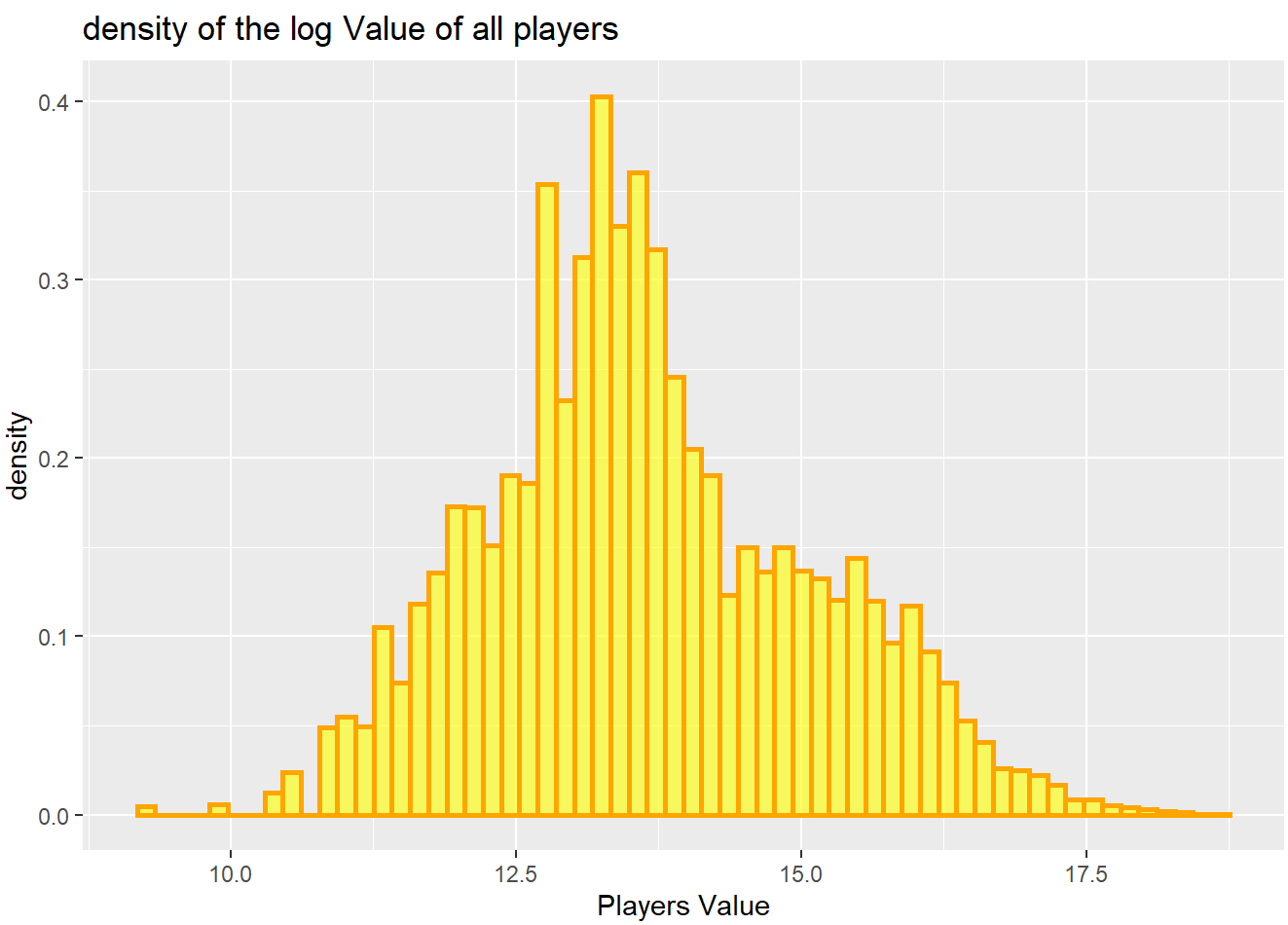
1.c

```
options(scipen = 999)
ggplot(data = fifa_players,aes(x = Value))+
  geom_histogram(aes(y=..density..),col = "orange", fill = "yellow", lwd = 0.9,bins = 60,
alpha = 0.6)+
  labs(title = "density  Value of all players") +
  xlab("Players Value")
```




```
fifa_players$log_value = log(fifa_players$Value)

ggplot(data = fifa_players,aes(x = log_value))+
  geom_histogram(aes(y=..density..),col = "orange", fill = "yellow", lwd = 0.9,bins = 60,
                 alpha = 0.6)+
  labs(title = "density of the log Value of all players") +
  xlab("Players Value")
```

```
## Warning: Removed 227 rows containing non-finite values (stat_bin).
```



as we can see, before we log the value, the numbers were high and the distribution didn't look normal at all. after we computed the log on all the values, the distribution became more like normal, with values that make it more easy and comfortable to watch and analyze. thus, the log visualization is better.

1.d

```
top_value = fifa_players %>% select(Name, Value) %>% arrange(desc(Value))
top_overall = fifa_players %>% select(Name, Overall) %>% arrange(desc(Overall))
```

taking the best 10 players in each category:

```
top10_value = top_value[1:10,]
top10_overall = top_overall[1:10,]
kable(top10_value)
```

Name	Value
Neymar	123000000
L. Messi	105000000
L. Suárez	97000000
Cristiano Ronaldo	95500000
R. Lewandowski	92000000
E. Hazard	90500000
K. De Bruyne	83000000
P. Dybala	79000000
T. Kroos	79000000
G. Higuaín	77000000

```
kable(top10_overall)
```

Name	Overall
Cristiano Ronaldo	94
L. Messi	93
Neymar	92
M. Neuer	92
L. Su¡rez	92
R. Lewandowski	91
E. Hazard	90
De Gea	90
G. Higuaˆn	90
T. Kroos	90

there are 2 players that are in the top 10 players by overall that not in the top 10 players by value - De Gea and M. Neuer. comparing these two players (which they are a goalkeeper), M. Neuer (92) has a higher overall score than De Gea (90). thus, M. Neuer is is the best player from top-10 overall players who’s not in the top-10 valued players.

1.e

```
agg4 = aggregate(Age~Club,fifa_players,mean)

age = agg4 %>% select(Club,Age) %>% arrange(Age)

youngest_10_clubs = head(age,10)
oldests_10_clubs = tail(age,10)
all_in_one_table = full_join(youngest_10_clubs,oldests_10_clubs)
```

```
## Joining, by = c("Club", "Age")
```

```
kable(all_in_one_table)
```

Club	Age
Sevilla Atl©tico	19.79
FC Barcelona B	20.38
Werder Bremen II	21.46
LOSC Lille	21.63
PSV	21.88
Crewe Alexandra	21.88
FC Nordsj¡lland	22.00
Ajax	22.07
KRC Genk	22.08
Barnsley	22.10
Clube Atl©tico Paranaense	30.00
Gr×mio Foot-Ball Porto Alegrense	30.00
Sydney FC	30.00
Jeonbuk Hyundai Motors	30.33
Adelaide United	30.40
Associa§£o Chapecoense de Futebol	30.60
FC Seoul	30.75
Western Sydney Wanderers	30.75

Club	Age
Brisbane Roar	31.00
Newcastle Jets	31.00

we can see in the final table that the first 10 teams are with the lowest value of mean age, and the next 10 teams in the table are with the highest value of mean age.

Q2. Scouting Report (30 pt)

You are in charge of the scouting division. The goal of this division is to follow players' `potential` and `overall` ability, and identify undervalued players - that is, players whose current value is lower compared to what would be expected based on their predicted future ability.

a. Plot the *average* `potential` ability by `age` of all players, for players 35 years old or younger

b. Plot the *average difference* between a player's `overall` ability to `potential` ability as a function of `age`, up to age 35. At what ages should we expect to find players for future development based on this graph?

c. We are seeking young ($age \leq 21$) players with high `potential` (> 70). Show a scatter plot of these players comparing their `potential` ability (x-axis) and current `value` (y-axis).
Find the 10 most-undervalued players, i.e. having the lowest `value` compared to their predicted value by `potential` using a simple linear regression model.
Calculate for each of them what is a fair `value` matching their `potential` that you be willing to pay in order to by them to your club and show these 10 players with their name, `age`, `overall` ability, `potential`, actual `value` and fair `value` it a table.

d. Your boss wants to fly abroad to recruit promising players. Use the `rworldmap` package to display the world map and color each country based on the *median* `potential` of players from this nationality.

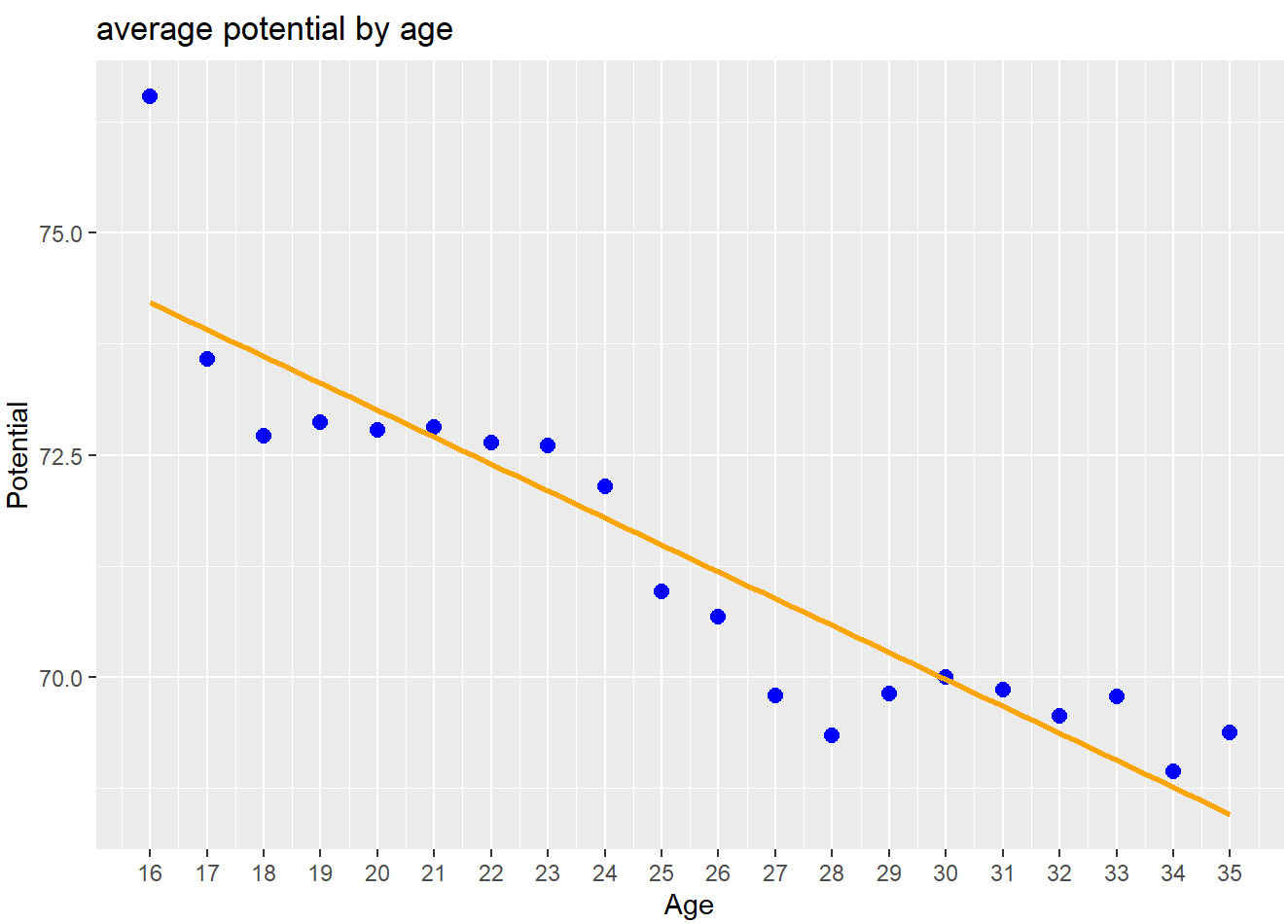
e. Repeat the above analysis but this time display a world map where each country is colored by the *median ratio* of `potential` to `value` of players. Find an under-valued country you'd recommend to travel to (i.e. a country with cheap players compared to their `potential` average quality).

solution:

2.a

```
age_under_35 = filter(.data = fifa_players, Age <= 35)
agg5 = aggregate(Potential~Age,age_under_35,mean)

ggplot(agg5,aes(Age, Potential)) + geom_point(col = "blue",lwd = 2.5) +
  scale_x_continuous(n.breaks = 19) +
  ggtitle(label = "average potential by age")+
  geom_smooth(method = "lm",se = F,lwd = 1,col = "orange")
```

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



2.b

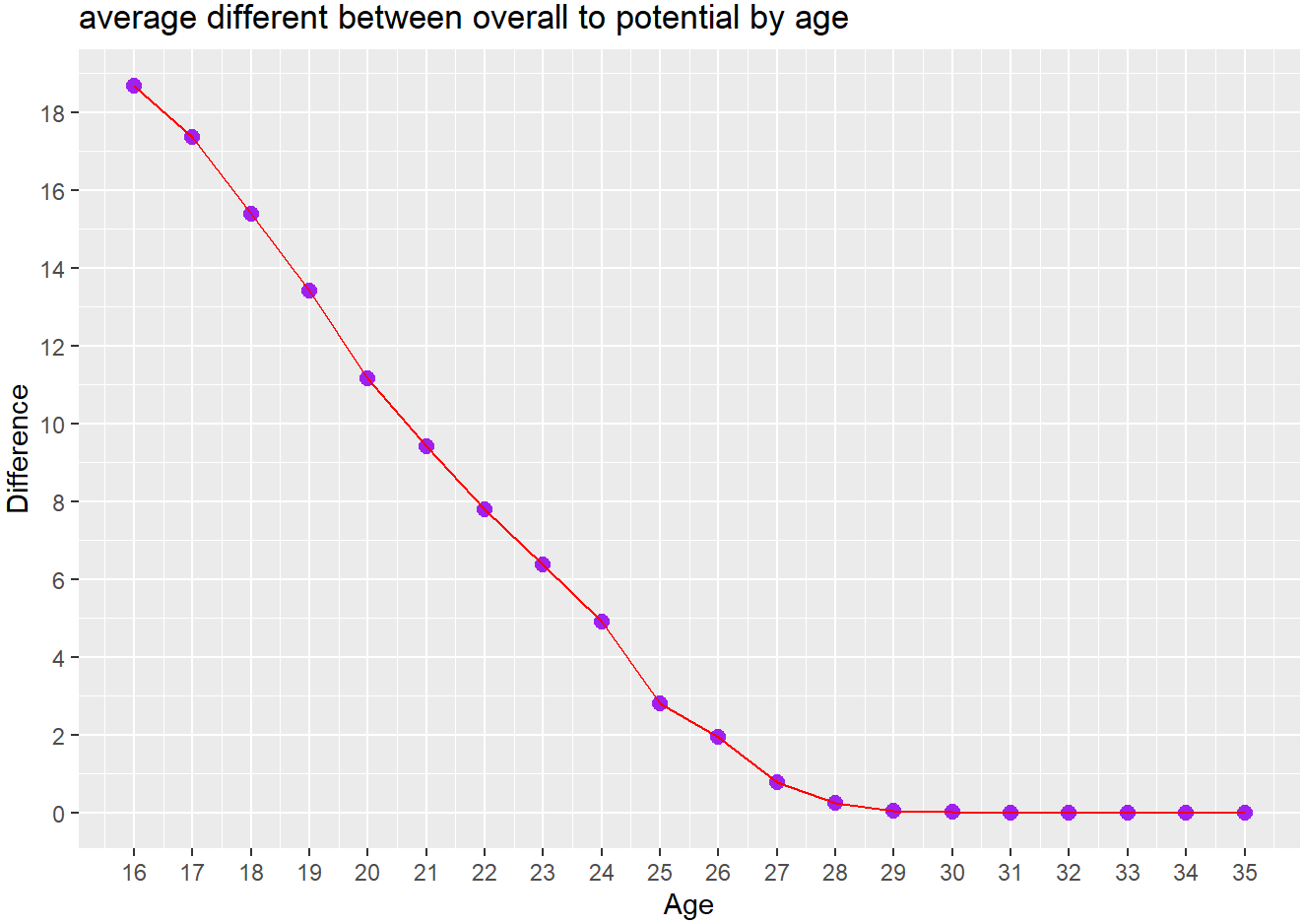
```
agg6 = aggregate(Overall~Age,age_under_35,mean)
agg_total = full_join(agg5,agg6)
```

```
## Joining, by = "Age"
```

```
agg_total$dif = agg_total$Potential - agg_total$Overall
```

different between overall to potential:

```
ggplot(agg_total,aes(Age,dif)) + geom_point(col = "purple",lwd = 2.5)+ geom_line(col = "red") +
  scale_x_continuous(n.breaks = 19) + scale_y_continuous(n.breaks = 10)+
  ggtitle(label = "average different between overall to potential by age") +
  labs(y = "Difference")
```

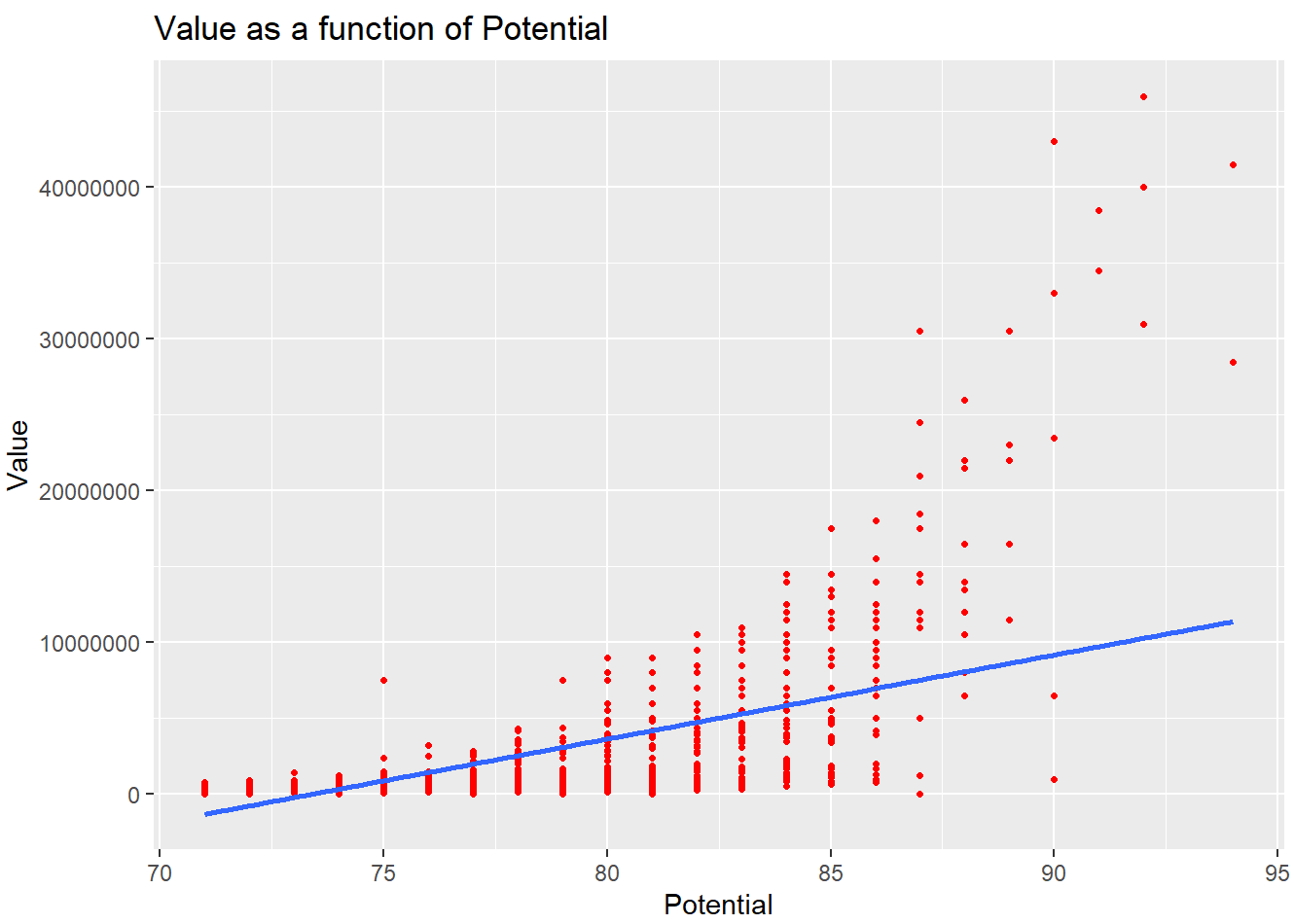


as we can see, the biggest different between potential and overall is at the age of 16, which the different value is 18.69. from that age the different starts to decline, and around age 26 the different is around 1. at the age of 30 the different is zero, so we can conclude that a player reach his max potential at the age of 30-31. so the best time to buy a player is when the different reach its max value, which is at the age of 16.

2.c

```
age_under_21 = filter(.data = fifa_players,
                      Age <= 21, Potential > 70)

ggplot(age_under_21,aes(Potential, Value)) +
  geom_point(col = "red", lwd = 0.9) +
  geom_smooth(method = "lm",se = F,lwd = 1) +
  ggtitle(label = "Value as a function of Potential")
```

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



Finding the 10 most-undervalued players :

```
most_uv = age_under_21[,c("Name", "Value", "Potential")]
fm1 = lm(age_under_21$Value~age_under_21$Potential)
most_uv$pred = fm1$fitted.values
most_uv$dif = most_uv$pred - most_uv$Value
most_uv = arrange(.data = most_uv,desc(dif))
the_most_uv_names = head(most_uv$Name,10)
our_best_10 =  head(most_uv,10)

most_uv2 = age_under_21[,c("Name","Age", "Overall","Potential", "Value")]
fm2 = lm(age_under_21$Value~age_under_21$Potential)
most_uv2$pred = fm2$fitted.values
most_uv2$dif = most_uv2$pred - most_uv2$Value
most_uv2 = arrange(.data = most_uv2,desc(dif))
the_most_uv2_names = head(most_uv2$Name,10)
our_best2_10 =  head(most_uv2,10)

kable(our_best2_10)
```

Name	Age	Overall	Potential	Value	pred	dif
A. Gomes	16	64	90	97500091564588181458		
W. Far±ez	19	73	87	075045327504532		
M. Edwards	18	65	87	1200000075045326304532		
J. Sancho	17	63	86	80000069538906153890		
C. Fr¼chtl	17	65	86	97500069538905978890		
E. Abouchabaka	17	62	85	65000064032485753248		
R. Sessegnon	17	67	86	1300000069538905653890		
V. Thill	17	63	85	80000064032485603248		

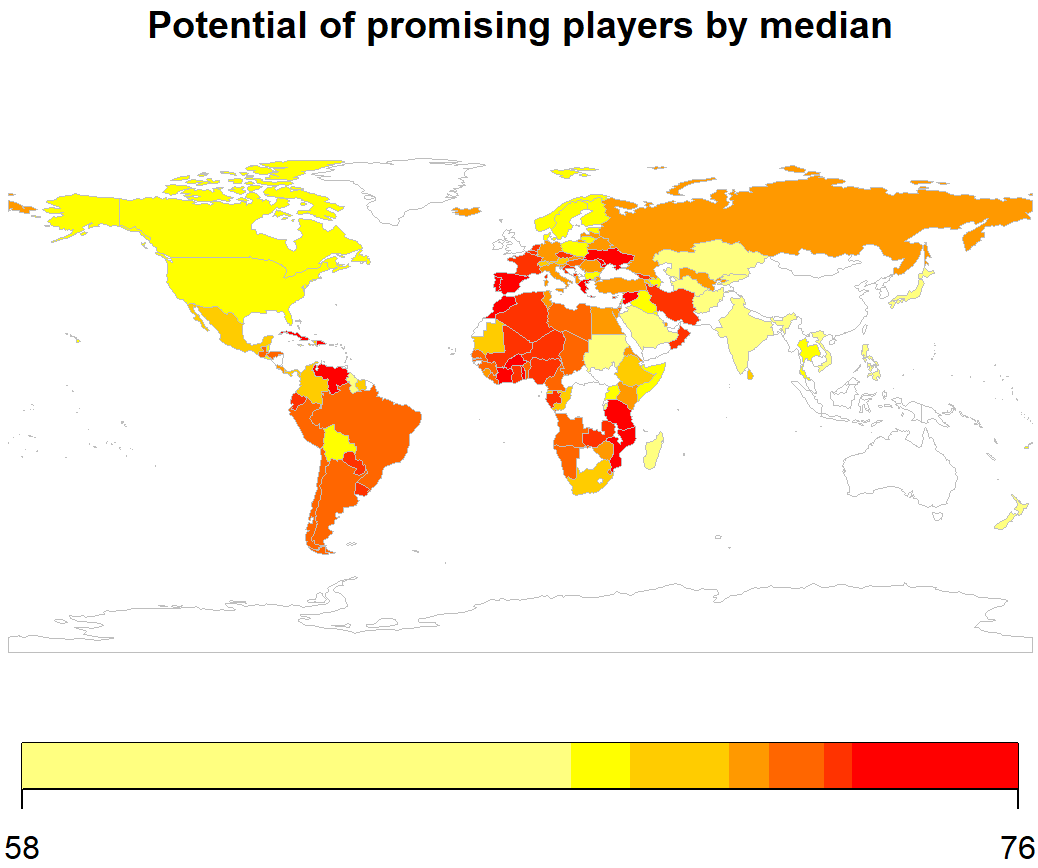
Name	Age	Overall	Potential	Value	pred	dif
J. Arp	17	63	85	825000640324855782 48		
B. Woodburn	17	65	85	1100000640324853032 48		

2.d

```
library(rworldmap)

potential_median = aggregate(fifa_players$Potential,
by = list(fifa_players$Nationality),
FUN = median)%>% rename(Nationality = Group.1, Median = x)

data_map = joinCountryData2Map(dF = potential_median,
joinCode = "NAME", nameJoinColumn = "Nationality",verbose = F)
```

```
## 133 codes from your data successfully matched countries in the map
## 4 codes from your data failed to match with a country code in the map
## 110 codes from the map weren't represented in your data
```

```
mapCountryData(mapToPlot = data_map,
nameColumnToPlot = "Median",
mapTitle = "Potential of promising players by median")
```

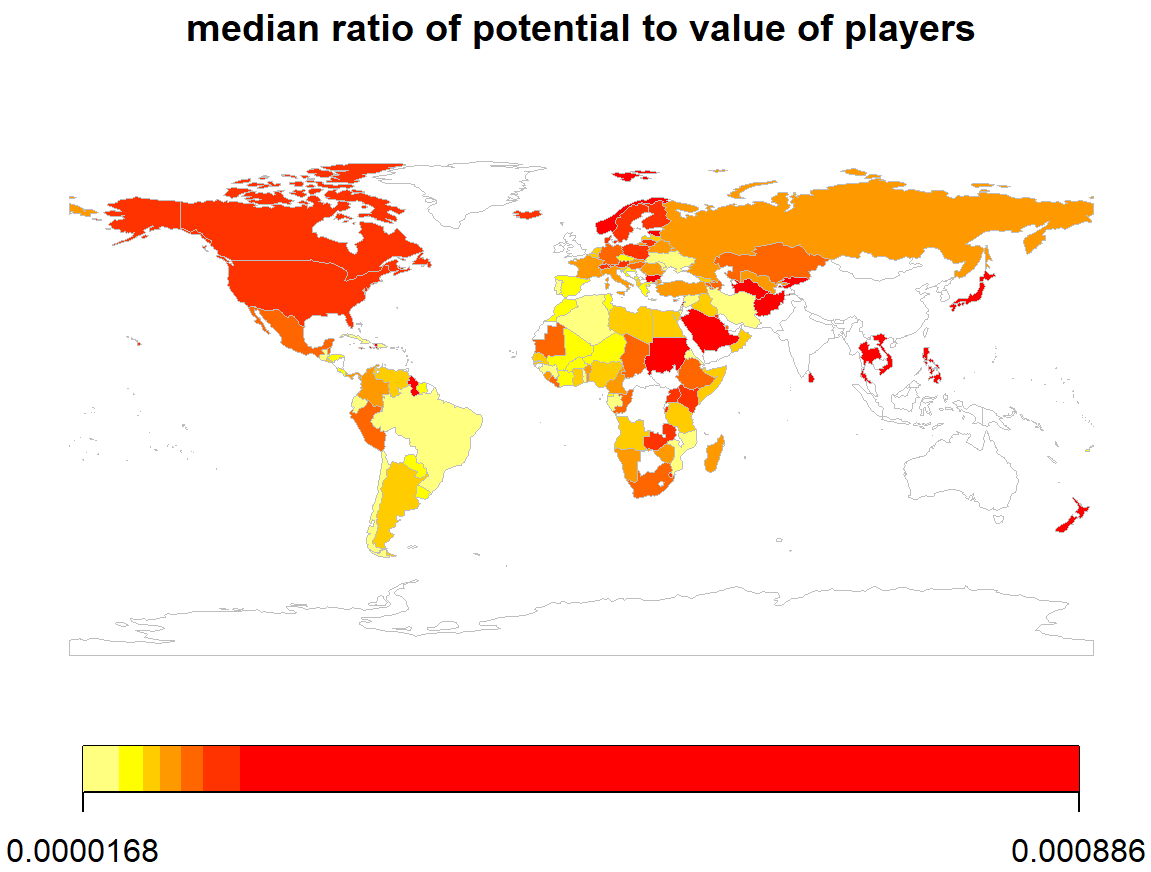


2.e

```
temp = fifa_players

# potential - value's Ratio
temp = mutate(temp, ratio = Potential/Value)

# ratio by nationality Aggregate
nationality_ratio = aggregate(temp$ratio ,by=list(temp$Nationality),FUN=median) %>% rename(Nation
ality=Group.1,Median_ratio = x) %>% filter(!Median_ratio %in% c("Inf"))

# map2
map_median_ratio = joinCountryData2Map(nationality_ratio, joinCode = "NAME", nameJoinColumn = "Na
tionality",verbose = F)
```

```
## 131 codes from your data successfully matched countries in the map
## 4 codes from your data failed to match with a country code in the map
## 112 codes from the map weren't represented in your data
```

```
mapCountryData(mapToPlot = map_median_ratio, nameColumnToPlot = "Median_ratio", mapTitle = "media
n ratio of potential to value of players")
```



Q3. Correlations Analysis (30 pt)

In this question we find and display different skills and their correlations

a. We are interested in finding out which positions are similar in terms of players' performance. Extract the 26 non-goalkeeper positions (`CAM`, `CB`, `...`, `ST`). Calculate the correlation between players' ability in each pair of positions and show a heatmap correlation-plot of the correlations' matrix. What

three positions have the *least* average correlations with other skills?
We are interested in finding out which skills are similar in terms of players' performance at the position. Extract the 29 skills for non-goalkeeper players (Acceleration, ..., Volleys, except 'GK.*' skills). Calculate the correlation between players' ability in each pair of skills and show a heatmap correlation-plot of the correlations' matrix. What two skills seem least correlated with other skills?

b. Consider the following indicators of players performance: `overall` players' performance, their `potential`, their salary (`wage`) and their market `value`. Show a correlation-plot of players' *34* skill levels (`Acceleration`, ..., `Volleys`) vs. these four indicators. Find the *10* skills with the highest *average* correlation with the four inidcators and list them in a table.

c. Build a team of *11 different* players with the following rules:

- For each of the *26* non-goalkeaper positions (*26* from above plus goalkeaper, `GK`), find the player with the best performance at this position.
- Find the goal keaper (`Preffered.Positions` is `GK`) with the best `overall` performance.
- From the players obtained above, find *11 distinct* players maximizing the average `overall` performance of the team, with the constraint that there must be a goalkeaper (preferred position `GK`).
- List the players in a table including their `overall` performance and the team average `overall` score.
Next, peak six *different* players of your choice from your team, one of which is the goalkeaper. Using the function `radarchart::chartJSRadar`, graph their abilities (individually for all 6 players) in the top *10* skills according to 3.b in a radar chart (also called 'spider chart').

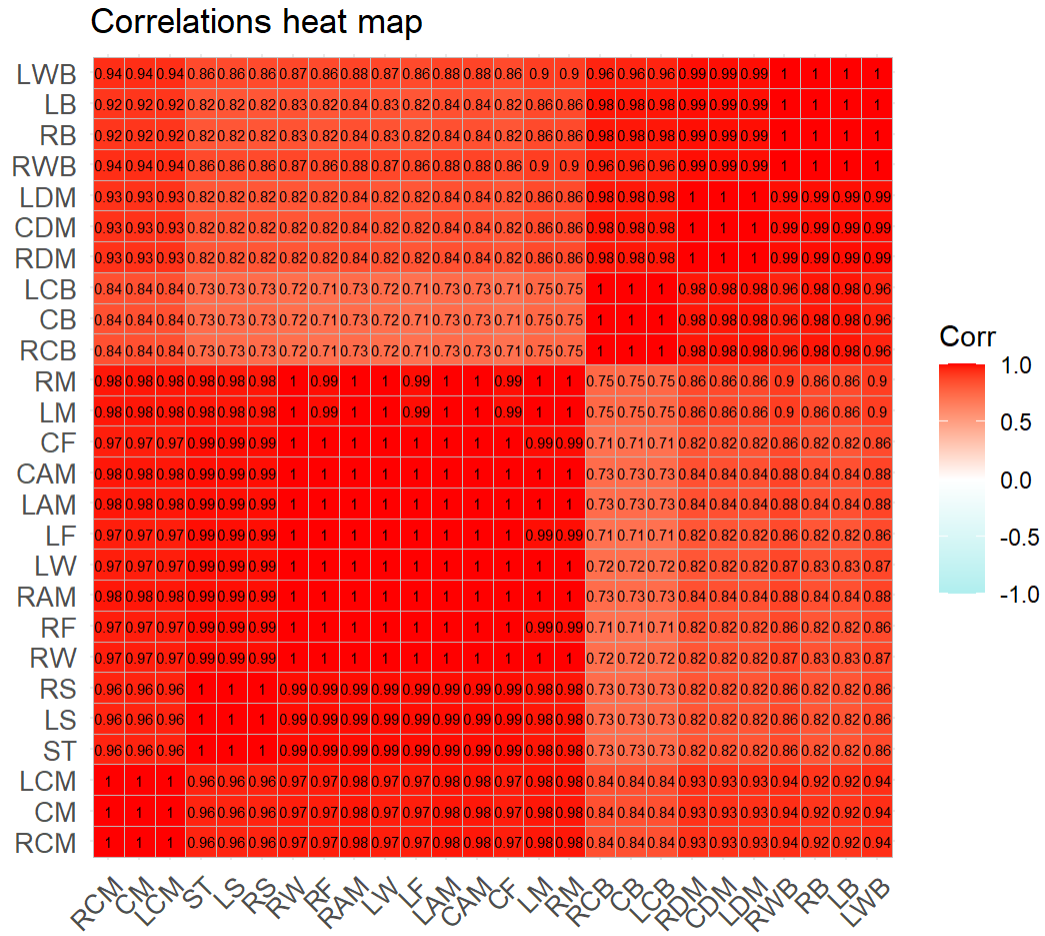
d. We are interested in determining how the player's abilities in different positions changes with age. Repeat the analysis of question 2.a., but this time show the *34* different skills
Which skills peak at youngest/oldest ages?

e. Your boss suggests that some players may be currently under-payed compared to their performance, and that we can acquire them by offering them a higher salary (`wage`).
Fit a multiple regression model predicting player's `overall` performance based on their `wage` and `age`.
Find the 10 players with the highest difference between their `overall` performance level and the regression model prediction, and list them in a table.

solution:

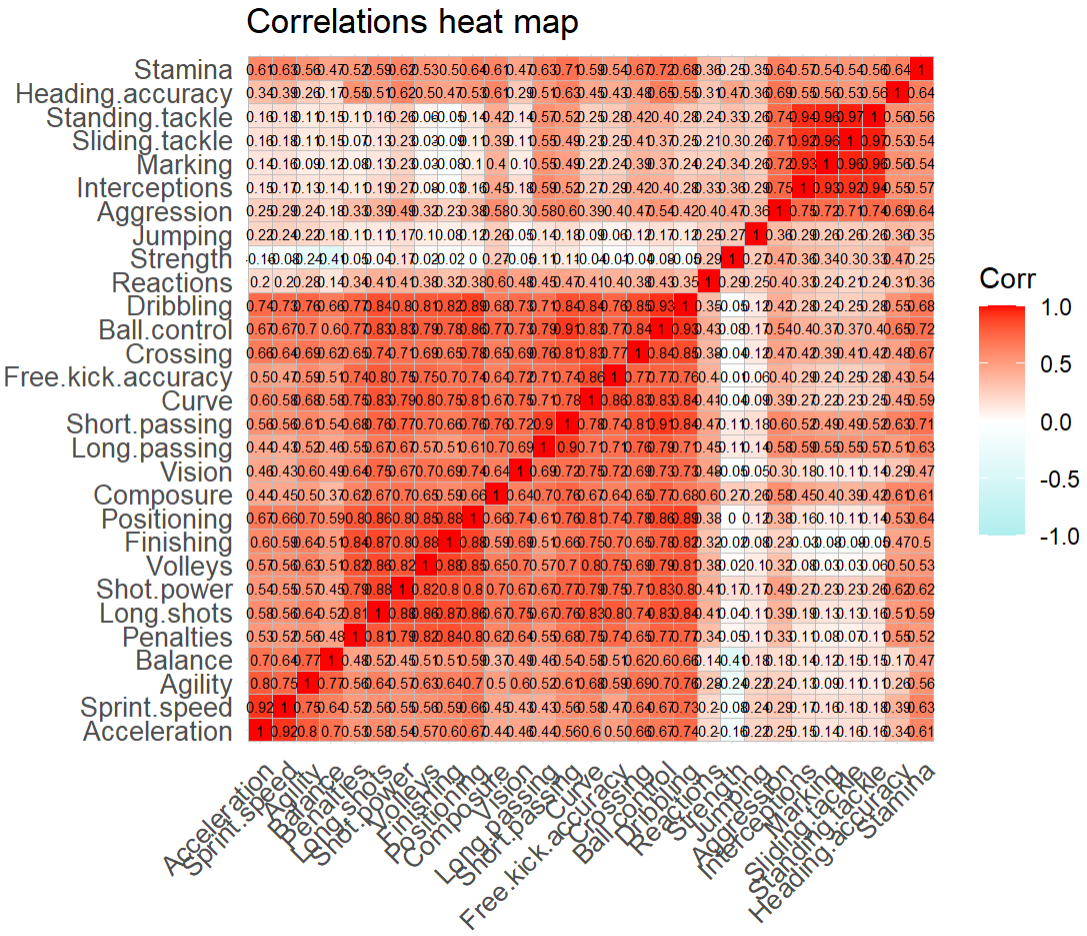
3.a

```
players_positions = fifa_players_positions[,c(2:27)]

pos_cor = cor(players_positions)

ggcorrplot(pos_cor, hc.order = T,lab = T ,lab_size = 2
           ,title = "Correlations heat map",
           colors = c("paleturquoise","white","red"),tl.cex = 10)
```



according to the plot colors, the three positions have the least average correlations with other skills are the lines with the brightest color, which is : LCB, CB, RCB.

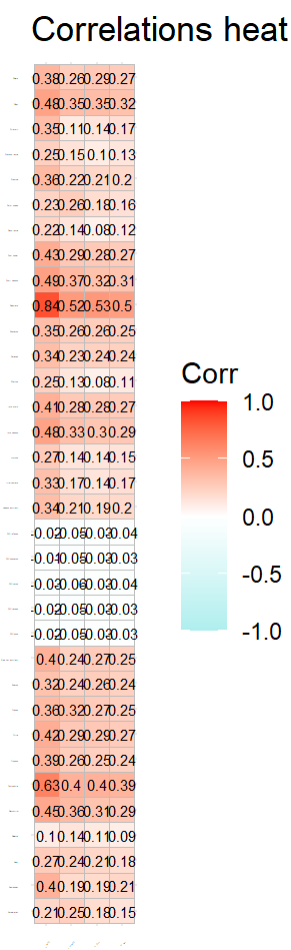
```
# exclude G.K stats and Overall stats
players_skiils = fifa_players_attribures[,c(2:12, 18:35)]

skills_cor = cor(players_skiils)

ggcorrplot(skills_cor, hc.order = T,lab = T ,lab_size = 2
           ,title = "Correlations heat map",
           colors = c("paleturquoise","white","red"),tl.cex = 10)
```



according to the plot colors, the two skills seem least correlated with other skills are the lines with the brightest color, which is : Strength, Jumping.

3.b

```
skills_34 =  fifa_players_attribures[,c(2:35)]

players_performance = fifa_players_indicators[,c(2:5)]

comp_cor = cor(x = players_performance, y = skills_34)

ggcorrplot(comp_cor,lab = T ,lab_size = 2
           ,title = "Correlations heat map",
           colors = c("paleturquoise","white","red"),tl.cex = 0.7)
```




```
# calculate the average cor for each row:

matrix_cor = as.data.frame(t(comp_cor))
table = rowMeans(matrix_cor)
matrix_cor$mean = table

# Find the 10 skills with the highest average correlation:

top_av_cor = matrix_cor %>% select(Overall, Potential, Value, Wage, mean) %>% arrange(desc(mean))
top_10_av_cor = top_av_cor[1:10,]
kable(top_10_av_cor)
```

	Overall	Potential	Value	Wage	mean
Reactions	0.8410	0.5203	0.5327	0.5036	0.5994
Composure	0.6327	0.4033	0.4027	0.3917	0.4576
Short.passing	0.4910	0.3743	0.3250	0.3092	0.3749
Vision	0.4836	0.3452	0.3482	0.3211	0.3745
Ball.control	0.4550	0.3636	0.3097	0.2936	0.3555
Long.passing	0.4778	0.3322	0.3002	0.2905	0.3502
Shot.power	0.4287	0.2904	0.2800	0.2713	0.3176
Curve	0.4150	0.2862	0.2873	0.2687	0.3143
Long.shots	0.4142	0.2750	0.2814	0.2656	0.3090
Dribbling	0.3599	0.3182	0.2694	0.2501	0.2994

3.c

```
#creating a data frame (df) containing the 26 positions and the players names
pos_names = fifa_players[,c(2,46:71)]
long_names = c()
positions_name = c(names(fifa_players_positions[2:27]))
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

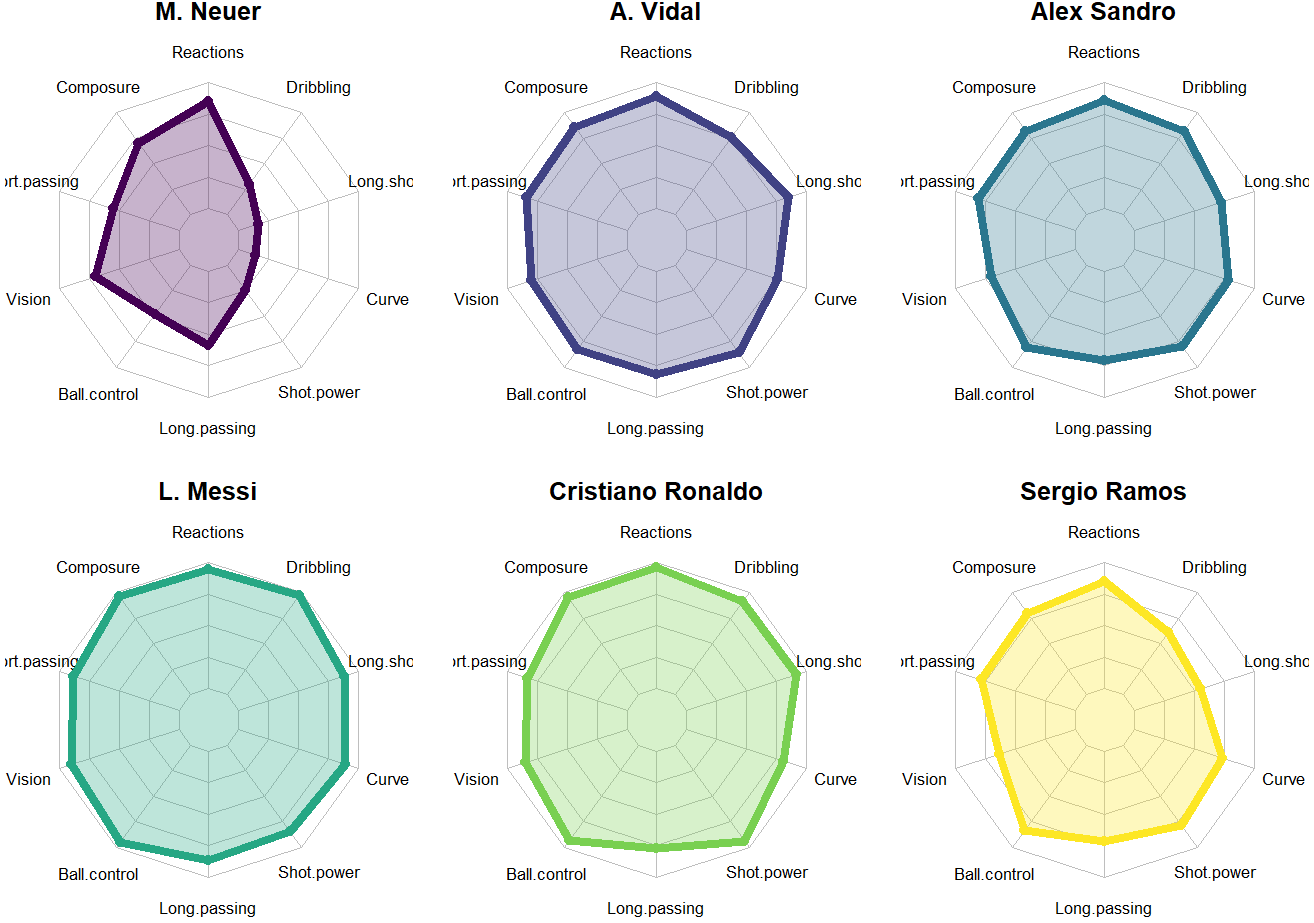
```
#finding the best player in each position
for (i in positions_name){
  tempo = pos_names$Name[pos_names[[i]] == max(pos_names[[i]])]
  long_names = c(long_names, tempo)
}
gk = fifa_players %>% select(Name,Overall,Preferred.Positions) %>% filter(Preferred.Positions ==
"GK ")
best_gk = head(data.frame(gk$Name, gk$Overall) %>% arrange(desc(gk$Overall)),1)
best_gk$Position = "GK"
names(best_gk) = c("Name","Overall", "Position")
# the goal-keaper with the best overall performance is M. Neuer

unique_players = as.data.frame(unique(long_names))

#building the team
#making sure every player from long_names appears once

#finding out which Marcelo
who_is_marcelo = fifa_players  %>% filter(Name == "Marcelo")%>% select(ID, Name, RWB, Overall)
progress = who_is_marcelo %>% arrange(desc(Overall))
progress$RWB = NULL
marcelo_final = progress[1, 2:3]
#finding the overall for each player

uniques_without_marcelo = unique_players[unique_players != "Marcelo"]
average_overall = c()
for (i in uniques_without_marcelo) {
  t = fifa_players$Overall[fifa_players$Name == i]
  average_overall = c(average_overall, t)
}
uniques_without_marcelo = as.data.frame(uniques_without_marcelo)
uniques_without_marcelo$Overall = average_overall
names(uniques_without_marcelo) = c("Name", "Overall")
players_11 = full_join(uniques_without_marcelo, marcelo_final)
```

```
## Joining, by = c("Name", "Overall")
```

```
#top 10 highest value players:
players_10 = players_11 %>% arrange(desc(Overall)) %>%  slice(1:10)

the_best_gk = best_gk[1,1:2]
# combining the gk with the top 10 others (11 players total):
our_open_11 = rbind(players_10,the_best_gk)

kable(our_open_11)
```

Name	Overall
Cristiano Ronaldo	94
L. Messi	93
Sergio Ramos	90
T. Kroos	90
A. Vidal	87
Marcelo	87
Alex Sandro	86
D. Alaba	86
R. Nainggolan	86
Azpilicueta	85
M. Neuer	92

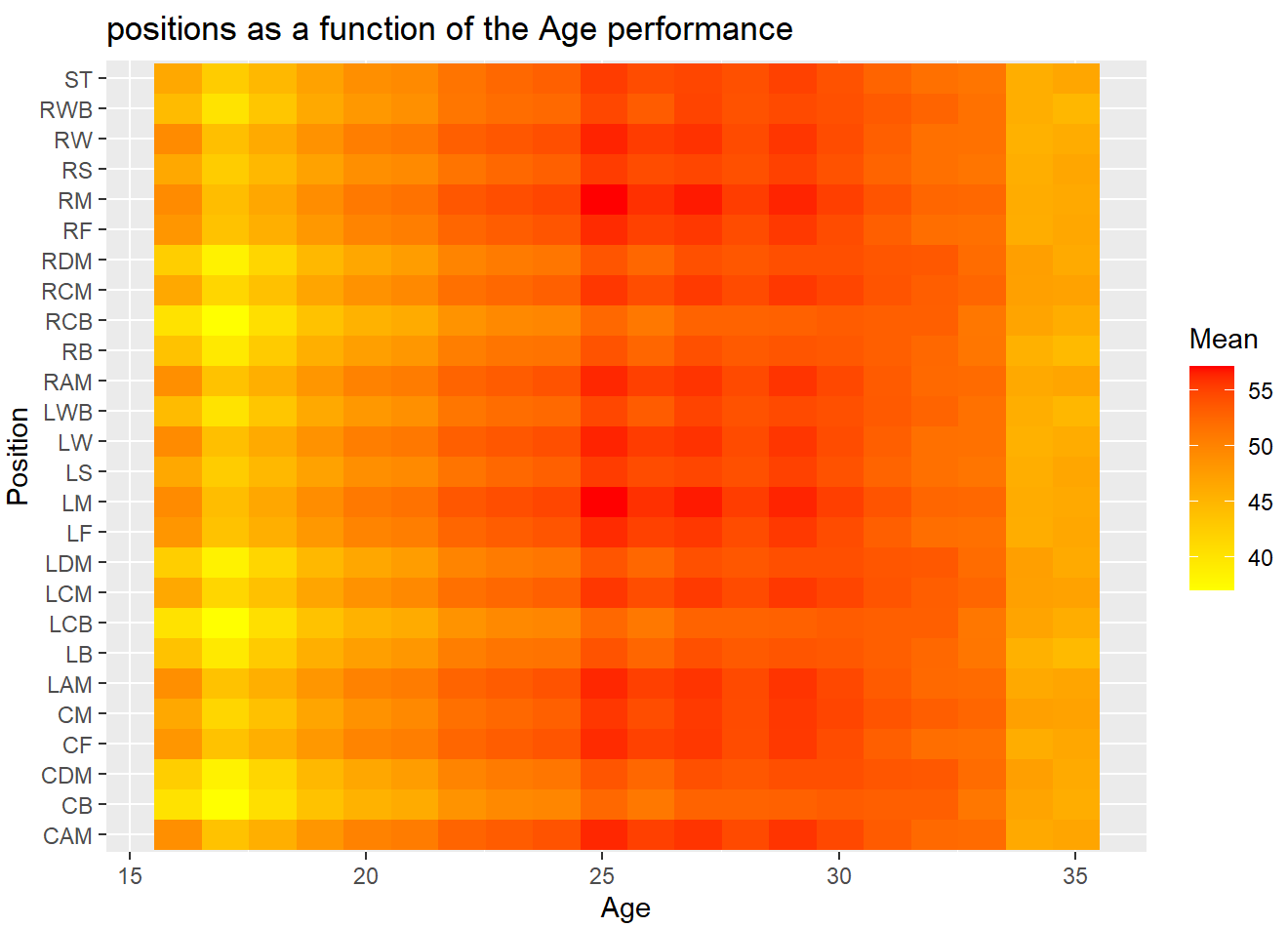
```
team_mean = mean(our_open_11$Overall)

#spider section:

df = as.data.frame(matrix( sample( 2:20 , 60 , replace = T) , ncol = 10, byrow = TRUE))
colnames(df) = c("Reactions" , "Composure" , "Short.passing" , "Vision" , "Ball.control", "Long.p
assing" , "Shot.power" , "Curve", "Long.shots", "Dribbling" )
 # choosing 6 players:
spider6 = fifa %>% filter(Name == "L. Messi" |
                                    Name == "Cristiano Ronaldo" |
                                    Name == "A. Vidal"|
                                    Name =="Sergio Ramos"|
                                    Name =="M. Neuer"|
                                    Name == "Alex Sandro" ) %>% select(Name,Reactions,Composure,S
hort.passing,Vision, Ball.control,Long.passing,Shot.power,Curve,Long.shots,Dribbling)

df = rbind(rep(100,1), rep(0,10), spider6 %>% select(2:11))

colors_border = colormap(colormap=colormaps$viridis, nshades = 6, alpha = 1)
colors_in = colormap(colormap = colormaps$viridis, nshades = 6, alpha = 0.3)

headline = spider6[1:6,1]
# syntax to showing all 6 graphs in one frame
par(mar = rep(0.8,4))
par(mfrow = c(2,3))
#using "for" loop for each one of the six players:
for(i in 1:6){
    radarchart( df[c(1, 2, i + 2),],
    pcol = colors_border[i] , pfcol = colors_in[i] , plwd = 4, plty = 1,
    cglcol = "grey", cglty = 1, axislabcol = "grey",
    caxislabel = seq(0,20,5), cglwd = 0.8,vlcex = 0.8,title = headline[i]
    )
}
```



as we can see, clearly - Messi is the best player in the world right now, comparing to the other 5 other players - Messi's field is cover the biggest space.

3.d

```
pos_new = fifa_players_positions[,c(2:27)]

mean_posss = pos_new
mean_posss = cbind(fifa_players$Age, pos_new)
names(mean_posss) = c("Age",colnames(pos_new))
mean_posss = mean_posss %>% filter(Age <= 35)

mean_posss = melt(mean_posss, "Age")

mean_posss = aggregate(value ~ Age + variable , mean_posss , mean)

colnames(mean_posss) = c("Age","Position","Mean")

ggplot(data = mean_posss, aes(x = Age, y = Position, fill = Mean))+
  geom_raster()+ scale_fill_gradient(low = "yellow",
high = "red")+ ggtitle("positions as a function of the Age performance")
```



as we can see - the mean performance gets higher when the color moves from yellow(40) to red(55). we can't find a position that act different than the others - basically all of the positions reach the peak around Age 25-30, and from there its start to go back down.

3.e

```
fm3 = lm(fifa_players$Overall~fifa_players$Age+fifa_players$Wage)
sub_data = fifa_players %>% select(Name, Overall)
sub_data$predicted_overall = fm3$fitted.values
sub_data$dif = sub_data$predicted_overall - sub_data$Overall
sub_data = sub_data %>% arrange(desc(dif))
sub_10 = sub_data %>% select(Name, dif)
kable(head(sub_10,10))
```

Name	dif
Cristiano Ronaldo	65.14
L. Messi	64.99
L. Su¡rez	57.15
G. Bale	35.93
L. Modri‡	33.41
R. Lewandowski	32.10
B. Richardson	31.17
T. Kroos	30.11
S. Ag¼ero	29.85
K. Benzema	28.03

Q4. Fix Problematic Plots (10 pt)

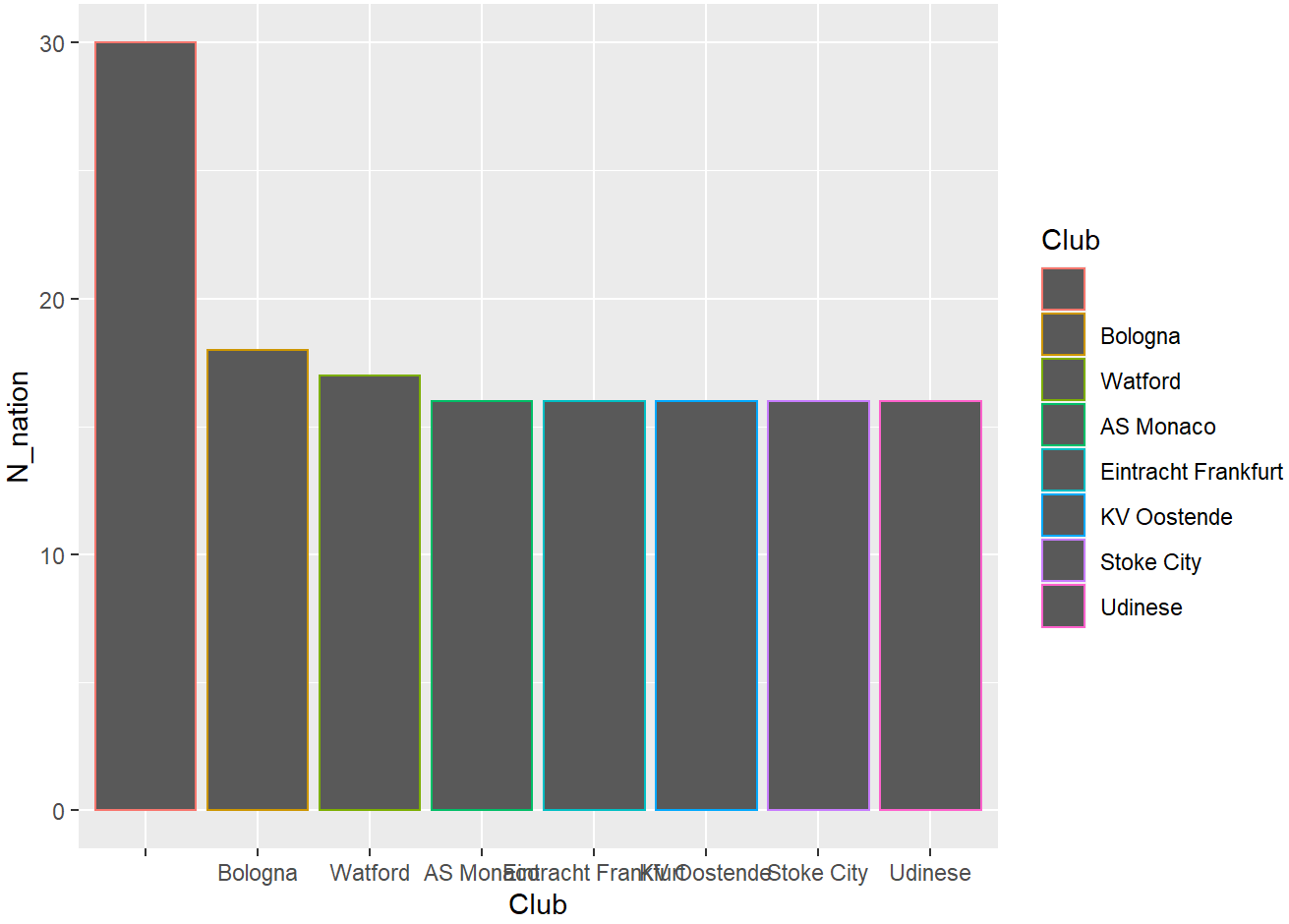
The previous data-analyst of the club was fired for producing poor plots. See below two bar plots that he made including their code.

a. Describe in your own words what did your predecessor try to show in each of the two plots.
b. Find *at least* three *different* problematic issues with his plots, and explain them.
c. Fix the problematic issues above in the code below to generate new, improved plots.
   You will get an additional *bonus* point for finding any additional problem and fixing it.
   (identifying the *same* problem in the two plots counts as *one* problem).

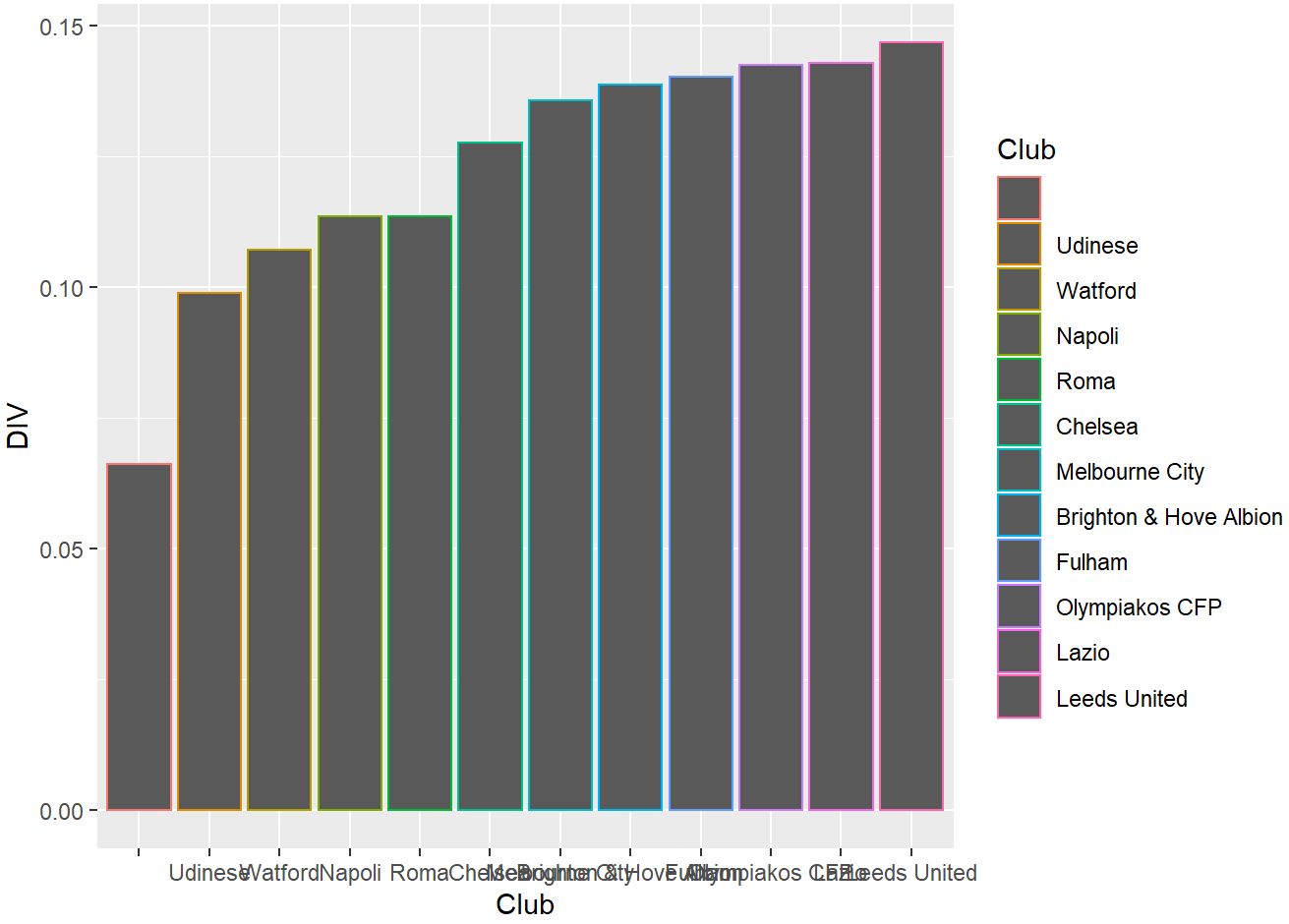
```
# A measure of category's diversity
DIV <- function(category_vec){
  t <- table(category_vec)
  p <- t/sum(t)
  return(sum(p^2))
}

cleaned_data <- fifa_players %>% select(Nationality,Club) %>% na.omit()

number_of_nationality_in_club <- cleaned_data %>% group_by(Club, Nationality) %>% summarise(count
= n()) %>% group_by(Club) %>% summarise(N_nation=n()) %>% arrange(desc(N_nation)) %>% mutate(Club
= factor(Club, level=unique(Club)))
```

```
## `summarise()` has grouped output by 'Club'. You can override using the `.groups` argument.
```

```
DIV_in_club <- cleaned_data %>% group_by(Club) %>% summarise(DIV = DIV(Nationality))%>% arrange(D
IV)%>% mutate(Club = factor(Club,level=unique(Club)))  # arrange(desc(DIV)) %>%

# Plot number of different nationalities in each club
g <- ggplot(data = number_of_nationality_in_club %>% head(8), aes(x = Club, y = N_nation,color =
 Club))
g + geom_bar(stat="identity")
```




```
# Plot DIV (diversity?) of different nationalities in each club
g <- ggplot(data = DIV_in_club %>% head(12),aes(x = Club,y = DIV, color = Club))
g <- g + geom_bar(stat="identity")
g
```



solution:

4.a

The function DIV gets an input of vector of categories (Nationalities) and compute the frequencies - how many times each country appears. After that the function divide by the total number of countries, getting the relative frequency of each country. The function then returns the sum of the squared probabilities. this function make the compute for each club, in order to describe how scattered the composition of the club.

about the plots:

the first and the second plots shows and describe the diversity of each club. The first plot shows quantitatively how many different countries constitute the club's composition. The plot shows the 8 clubs with the most nationalities constituting them. The second plot shows the 12 most divers countries relatively. Ignoring the size of the club, this plot takes into consideration the relative diversity and shows the 12 most ones.

4.b

3 different problematic issues with the plots :

1. Titles - there were no titles at all to the plots.
2. Overriding lables - the labels on the X-axis are overriding one another.
3. Relevant labels - the xlab and ylab doesn't have a relevant names. It's not obvious what the plots are trying to show. for example - the ylab in the second plot have the name "DIV". we cant understand easily what is the meaning of the plot with this label and what the data-analyst are trying to show.
4. Empty strings - the data frame "cleaned_data" tries to deal with NA's while in fact there were none, but it didn't deal with empty strings for clubs or nationalities.

5. Same variable for 2 different plots - the data-analyst set the same variable (g) for two different plots, which is override the former and can lead to mistakes and problems with the program.
6. The thin colored frame of the bins - its really hard to notice and distinguish between the colors with such a thin frames like that. if you choose to color the frame - make it clearly visible.
7. Uneasy to understand the ggplot's syntax - the use of pipe inside the ggplot make it really long and difficult to understand. we can do it outside the ggplot or take the countries we want with more simple ways like [m:n].

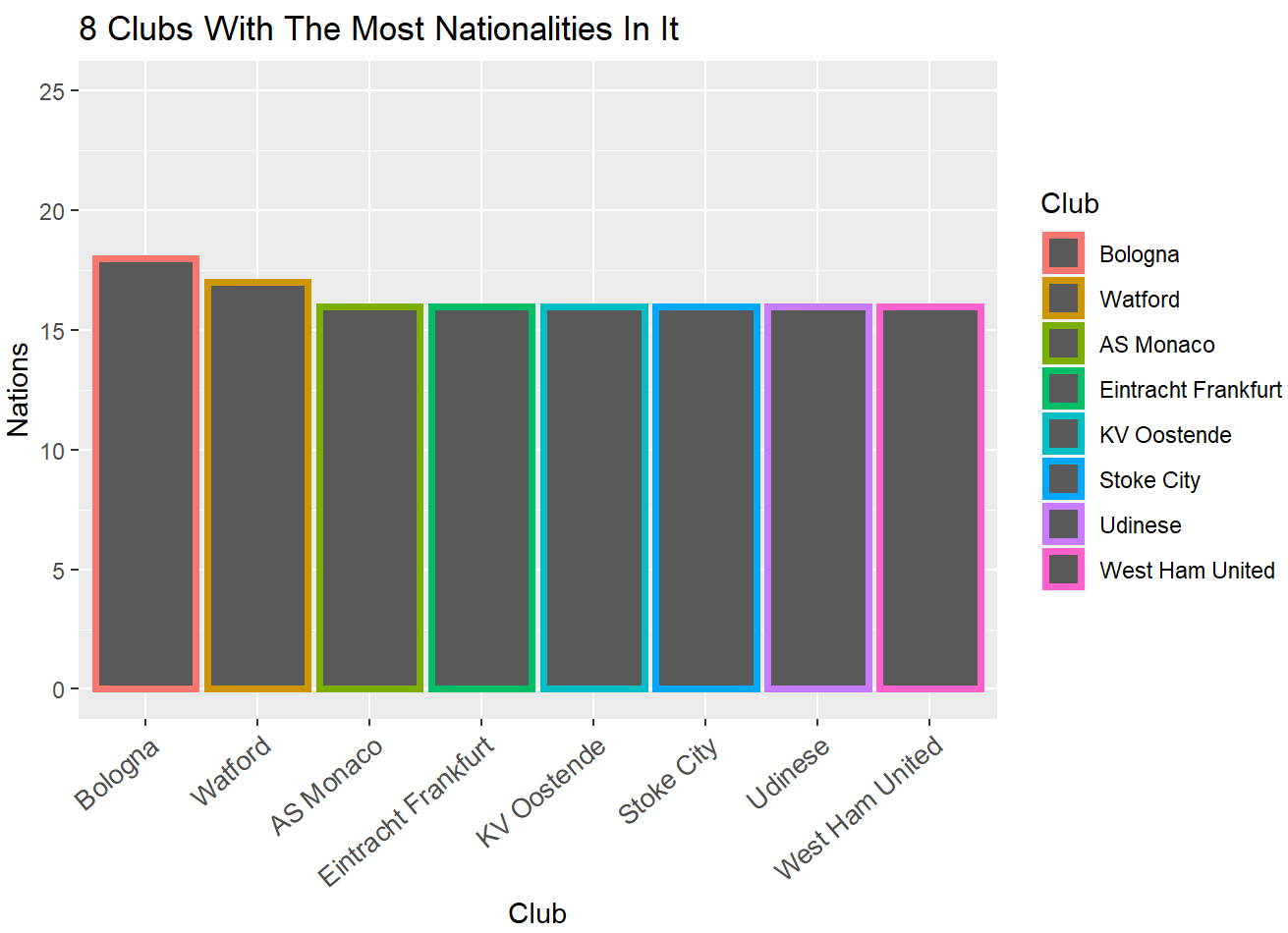
4.c

Our corrections to the code :

```
DIV <- function(category_vec)
{
  t <- table(category_vec)
  p <- t/sum(t)
  return(sum(p^2))
}
#getting rid of blank strings:
cleaned_data = fifa_players %>% select(Nationality,Club) %>% filter(Club!="")%>% filter(Nationali
ty!="")

number_of_nationality_in_club = cleaned_data %>% group_by(Club,Nationality) %>% summarise(count =
n()) %>% group_by(Club) %>% summarise(N_nation=n()) %>% arrange(desc(N_nation)) %>% mutate(Club =
factor(Club,level = unique(Club)))
```

```
## `summarise()` has grouped output by 'Club'. You can override using the `.groups` argument.
```

```
DIV_in_club = cleaned_data %>% group_by(Club) %>% summarise(DIV = DIV(Nationality))%>% arrange(DI
V)%>% mutate(Club = factor(Club,level = unique(Club)))

# first plot :
 plot1 = ggplot(data = number_of_nationality_in_club[1:8,], aes(x = Club,y = N_nation,color = Clu
b)) +
  geom_bar(stat="identity", size = 1.35) + ylab("Nations") +ylim(0,25) +
  ggtitle(label = "8 Clubs With The Most Nationalities In It") +
  theme(axis.text.x = element_text(angle = 40, hjust = 1, size = 10))

plot1
```




```
# second plot :
plot2 = ggplot(data = DIV_in_club[1:12,], aes(x = Club,y = DIV, color = Club)) +
  geom_bar(stat = "identity", size = 1.35) +
  ggtitle(label = "12 Most Relatively Diversed Clubs")+ ylim(0, 0.21)+
  labs(y = "variety") +
  theme(axis.text.x = element_text(angle = 40, hjust = 1, size = 10))

plot2
```