Roshan Shrestha Final Self Reflection

Link to my Shiny app

Link to my Github

Q. Import, manage, and clean data.

In my shiny project there was many instants we I had to load csv files. I have loaded the csv dataset then cleaned the data set by removing row with NA and empty values and separated into different csv files as per my need in shiny project. I can also import different types of files by different ways.

Importing csv file:

```r
fifa <- read.csv("Fifa.csv", header = T)
head(fifa)
```

```
##          short_name                            long_name player_positions
## 1          L. Messi      Lionel AndrÃ©s Messi Cuccittini       RW, ST, CF
## 2    R. Lewandowski                   Robert Lewandowski               ST
## 3 Cristiano Ronaldo Cristiano Ronaldo dos Santos Aveiro           ST, LW
## 4         Neymar Jr       Neymar da Silva Santos JÃºnior          LW, CAM
## 5      K. De Bruyne                      Kevin De Bruyne          CM, CAM
## 6          J. Oblak                            Jan Oblak               GK
##   overall potential value_eur wage_eur age        dob height_cm weight_kg
## 1      93        93  78000000   320000  34  6/24/1987       170        72
## 2      92        92 119500000   270000  32  8/21/1988       185        81
## 3      91        91  45000000   270000  36 02/05/1985       187        83
## 4      91        91 129000000   270000  29 02/05/1992       175        68
## 5      91        91 125500000   350000  30  6/28/1991       181        70
## 6      91        93 112000000   130000  28 01/07/1993       188        87
##             club_name Club_Rank            league_name club_position
## 1 Paris Saint-Germain         6         French Ligue 1            RW
## 2  FC Bayern MÃ¼nchen         1   German 1. Bundesliga            ST
## 3   Manchester United         9 English Premier League            ST
## 4 Paris Saint-Germain         6         French Ligue 1            LW
## 5     Manchester City         2 English Premier League           RCM
## 6 AtlÃ©tico de Madrid        10 Spain Primera Division            GK
##   club_jersey_number club_joined club_contract_valid_until Country_Rank
## 1                 30  08/10/2021                      2023            4
## 2                  9  07/01/2014                      2023           28
## 3                  7   8/27/2021                      2023            8
## 4                 10  08/03/2017                      2025            2
## 5                 17   8/30/2015                      2025            1
## 6                 13   7/16/2014                      2023           64
##            Rough Rough.1 nationality_name     Continent nation_position
## 1    Afghanistan    Asia        Argentina South America              RW
## 2  Aland Islands  Europe           Poland        Europe              RS
## 3        Albania  Europe         Portugal        Europe              ST
## 4        Algeria  Africa           Brazil South America
## 5 American Samoa Oceania          Belgium        Europe             RCM
## 6        Andorra  Europe         Slovenia        Europe
##   nation_jersey_number preferred_foot weak_foot skill_moves
## 1                   10           Left         4           4
## 2                    9          Right         4           4
## 3                    7          Right         4           5
## 4                   NA          Right         5           5
## 5                    7          Right         5           4
## 6                   NA          Right         3           1
##   international_reputation release_clause_eur pace shooting passing dribbling
## 1                        5          144300000   85       92      91        95
## 2                        5          197200000   78       92      79        86
## 3                        5           83300000   87       94      80        88
## 4                        5          238700000   91       83      86        94
## 5                        4          232200000   76       86      93        88
## 6                        5          238000000   NA       NA      NA        NA
##   defending physic
## 1        34     65
## 2        44     82
## 3        34     75
## 4        37     63
## 5        64     78
## 6        NA     NA
```

Importing Excel file:

```
library(readxl)
fifaxl = read_excel("Fifa.xlsx")
```

```
## New names:
## * Rough -> Rough...20
## * Rough -> Rough...21
```

```
fifaxl
```

```
## # A tibble: 20 x 36
##    short_name long_name   player_positions overall potential value_eur wage_eur
##    <chr>      <chr>       <chr>              <dbl>     <dbl>     <dbl>    <dbl>
##  1 L. Messi   "Lionel An~ RW, ST, CF            93        93  78000000   320000
##  2 R. Lewando~ "Robert Le~ ST                   92        92 119500000   270000
##  3 Cristiano ~ "Cristiano~ ST, LW               91        91  45000000   270000
##  4 Neymar Jr  "Neymar da~ LW, CAM               91         0 129000000   270000
##  5 K. De Bruy~ "Kevin De ~ CM, CAM              91        91 125500000   350000
##  6 J. Oblak   "Jan Oblak" GK                    91        93 112000000   130000
##  7 K. Mbappe  "Kylian Mb~ ST, LW                91        95 194000000   230000
##  8 M. Neuer   "Manuel Pe~ GK                    90        90  13500000    86000
##  9 M. ter Ste~ "Marc-Andr~ GK                   90        NA  99000000   250000
## 10 H. Kane    "Harry Kan~ ST                    90        90 129500000   240000
## 11 N. Kante   "N'Golo Ka~ CDM, CM               90        90 100000000   230000
## 12 K. Benzema "Karim Ben~ CF, ST                89        89  66000000   350000
## 13 T. Courtois "Thibaut C~ GK                   89        91  85500000   250000
## 14 H. Son     "ì†\u0090í~ LM, CF, LW            89        89 104000000   220000
## 15 Casemiro   "Carlos He~ CDM                   89        89  88000000   310000
## 16 V. van Dijk "Virgil va~ CB                   89        89  86000000   230000
## 17 S. Mane    "Sadio Man~ LW                    89        89 101000000   270000
## 18 M. Salah   "Mohamed S~ RW                    89        89 101000000   270000
## 19 Ederson    "Ederson S~ GK                    89        91  94000000   200000
## 20 J. Kimmich  "Joshua Wa~ CDM, RB               89        90 108000000   160000
## # ... with 29 more variables: age <dbl>, dob <chr>, height_cm <dbl>,
## #   weight_kg <dbl>, club_name <chr>, Club_Rank <chr>, league_name <chr>,
## #   club_position <chr>, club_jersey_number <dbl>, club_joined <chr>,
## #   club_contract_valid_until <dbl>, Country_Rank <dbl>, Rough...20 <chr>,
## #   Rough...21 <chr>, nationality_name <chr>, Continent <chr>,
## #   nation_position <chr>, nation_jersey_number <dbl>, preferred_foot <chr>,
## #   weak_foot <dbl>, skill_moves <dbl>, international_reputation <dbl>,
## #   release_clause_eur <dbl>, pace <dbl>, shooting <dbl>, passing <dbl>,
## #   dribbling <dbl>, defending <dbl>, physic <dbl>
```

Import files by here::here :

```
library(here)
```

```
## here() starts at /home/shrestro/518 Stat/Project_R/Final
```

```
dt <- readr::read_csv(here::here('data','Fifa.csv'))
```

```
## Warning: Duplicated column names deduplicated: 'Rough' => 'Rough_1' [21]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   short_name = col_character(),
##   long_name = col_character(),
##   player_positions = col_character(),
##   dob = col_character(),
##   club_name = col_character(),
##   Club_Rank = col_character(),
##   league_name = col_character(),
##   club_position = col_character(),
##   club_joined = col_character(),
##   Rough = col_character(),
##   Rough_1 = col_character(),
##   nationality_name = col_character(),
##   Continent = col_character(),
##   nation_position = col_character(),
##   preferred_foot = col_character()
## )
## i Use `spec()` for the full column specifications.
```

Importing of image by using here::here.

```
library(imager)
```

```
## Loading required package: magrittr
```

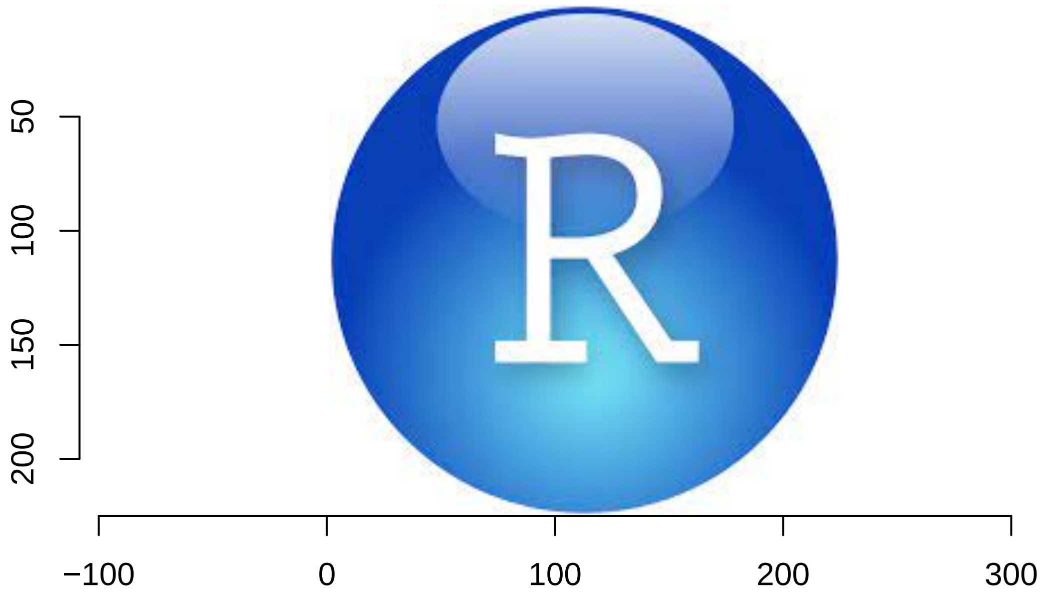
```
##
## Attaching package: 'imager'
```

```
## The following object is masked from 'package:magrittr':
##
##     add
```

```
## The following objects are masked from 'package:stats':
##
##     convolve, spectrum
## The following object is masked from 'package:graphics':
##
##     frame

## The following object is masked from 'package:base':
##
##     save.image
```

```
img <- jpeg::readJPEG(here::here('images','r.jpeg'))
img1<-load.image('~/518 Stat/Project_R/Final/images/r.jpeg')
plot(img1)
```



I am also able to write cvs from the data frame where we can easily download the the csv file:

```
write.csv(fifaxl,"~/518 Stat/Project_R/fifaxl.csv", row.names = TRUE)
```

Clean the data set.

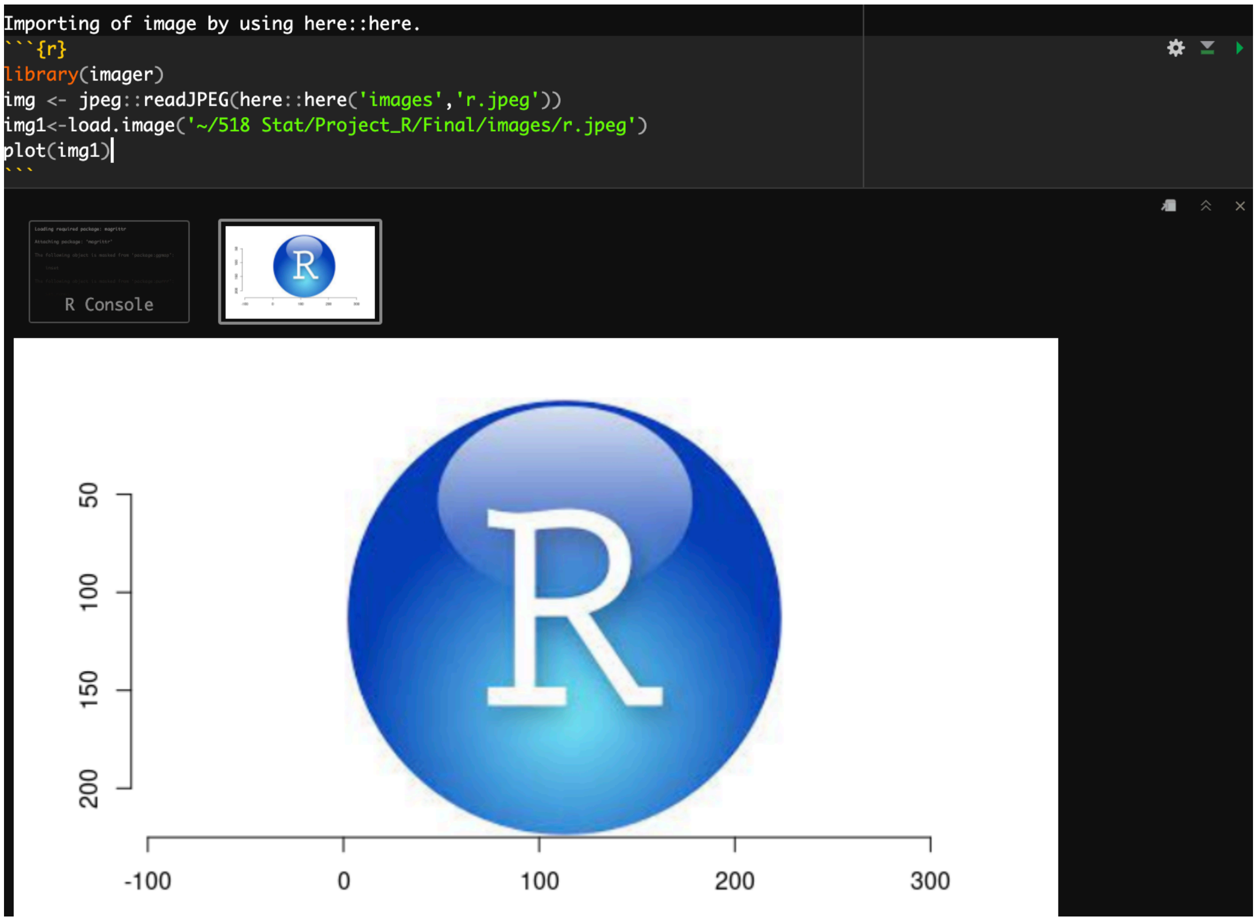
I am able to clean the data by removing the NA, empty values and zero values from the dataset. For my shiny app I have imported many files from various sources and clean the data as per the requirement of the project.

Removing the row with NA values, zero values and empty cell.

```
fifaxl = fifaxl[!(is.na(fifaxl$potential) | fifaxl$potential=="" | fifaxl$potential=="0" ),]

fifaxl
```

```
## # A tibble: 18 x 36
##    short_name  long_name   player_positions overall potential value_eur wage_eur
##    <chr>       <chr>       <chr>              <dbl>     <dbl>     <dbl>    <dbl>
##  1 L. Messi    "Lionel An~ RW, ST, CF            93        93  78000000   320000
##  2 R. Lewando~ "Robert Le~ ST                    92        92 119500000   270000
##  3 Cristiano ~ "Cristiano~ ST, LW                91        91  45000000   270000
##  4 K. De Bruy~ "Kevin De ~ CM, CAM               91        91 125500000   350000
##  5 J. Oblak    "Jan Oblak" GK                    91        93 112000000   130000
##  6 K. Mbappe   "Kylian Mb~ ST, LW                91        95 194000000   230000
##  7 M. Neuer    "Manuel Pe~ GK                    90        90  13500000    86000
##  8 H. Kane     "Harry Kan~ ST                    90        90 129500000   240000
```



~~~
Importing of image by using here::here.
```{r}
library(imager)
img <- jpeg::readJPEG(here::here('images','r.jpeg'))
img1<-load.image('~/518 Stat/Project_R/Final/images/r.jpeg')
plot(img1)
```
~~~

Figure 1: Image

```
##  9 N. Kante    "N'Golo Ka~ CDM, CM               90        90 100000000   230000
## 10 K. Benzema  "Karim Ben~ CF, ST               89        89  66000000   350000
## 11 T. Courtois "Thibaut C~ GK                   89        91  85500000   250000
## 12 H. Son      "ì†\u0090í~ LM, CF, LW           89        89 104000000   220000
## 13 Casemiro    "Carlos He~ CDM                  89        89  88000000   310000
## 14 V. van Dijk "Virgil va~ CB                   89        89  86000000   230000
## 15 S. Mane     "Sadio Man~ LW                   89        89 101000000   270000
## 16 M. Salah    "Mohamed S~ RW                   89        89 101000000   270000
## 17 Ederson     "Ederson S~ GK                   89        91  94000000   200000
## 18 J. Kimmich  "Joshua Wa~ CDM, RB              89        90 108000000   160000
## # ... with 29 more variables: age <dbl>, dob <chr>, height_cm <dbl>,
## #   weight_kg <dbl>, club_name <chr>, Club_Rank <chr>, league_name <chr>,
## #   club_position <chr>, club_jersey_number <dbl>, club_joined <chr>,
## #   club_contract_valid_until <dbl>, Country_Rank <dbl>, Rough...20 <chr>,
## #   Rough...21 <chr>, nationality_name <chr>, Continent <chr>,
## #   nation_position <chr>, nation_jersey_number <dbl>, preferred_foot <chr>,
## #   weak_foot <dbl>, skill_moves <dbl>, international_reputation <dbl>,
## #   release_clause_eur <dbl>, pace <dbl>, shooting <dbl>, passing <dbl>,
## #   dribbling <dbl>, defending <dbl>, physic <dbl>
```

Manage the data

I can isolation with in the data set by using dplyr function where I used select, filter, groupby, summarize and mutate.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
country <- fifaxl %>%
  select(nationality_name,wage_eur) %>%
  group_by(nationality_name) %>%
  summarize(Avg_Wages = mean(wage_eur, na.rm = TRUE))

head(country)
```

```
## # A tibble: 6 x 2
##   nationality_name Avg_Wages
##   <chr>                <dbl>
## 1 Argentina           320000
## 2 Belgium             300000
## 3 Brazil              255000
## 4 Egypt               270000
## 5 England             240000
## 6 France              270000
```

In the above code, I am able to group the data set by nation name and calculate the average wages of each country with the help of dyplyr and data pipelines.

I am able to write write loop function to calculate the mean from the given data set.

Writing loop to calculate mean of overall ranking of players.

```
my_sum = 0
len = length(fifaxl$overall)
for(i in 1:len){
    my_sum = my_sum + fifaxl$overall[i]
}
mean_overall = my_sum/len
mean_overall
```

```
## [1] 90
```

To avoid writing complicated and long code I am also able to write map function .

I am also able to use map function for finding the mean of the data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.4     v purrr   0.3.4
## v tibble  3.1.2     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x imager::add()       masks magrittr::add()
## x stringr::boundary() masks imager::boundary()
## x tidyr::extract()    masks magrittr::extract()
## x tidyr::fill()       masks imager::fill()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
## x purrr::set_names()  masks magrittr::set_names()
```

```
map1 <- fifaxl %>% select(overall,wage_eur) %>% map_dbl(mean)
map1
```

```
##  overall wage_eur
##     90.0 243666.7
```

Here I have merge the data of two different csv to form one data frame.

```
one <- read.csv("one.csv")
two <- read.csv("two.csv")
# one
# two

total <- merge(one,two,by="short_name")
head(total)
```

```
##          short_name                           long_name player_positions
## 1          Casemiro   Carlos Henrique Venancio Casimiro              CDM
## 2 Cristiano Ronaldo Cristiano Ronaldo dos Santos Aveiro           ST, LW
## 3          Ederson            Ederson Santana de Moraes               GK
## 4           H. Kane                          Harry Kane               ST
## 5            H. Son       ì†\u0090í\u009d¥ë¯¼ å å…´æ…œ       LM, CF, LW
## 6        J. Kimmich                Joshua Walter Kimmich          CDM, RB
##   overall potential value_eur wage_eur age
## 1      89         89  88000000   310000  29
## 2      91         91  45000000   270000  36
## 3      89         91  94000000   200000  27
## 4      90         90 129500000   240000  27
## 5      89         89 104000000   220000  28
## 6      89         90 108000000   160000  26
```

According to the requirement of shiny app project I have used pivot wider and longer to manage the data as per the need of the project.

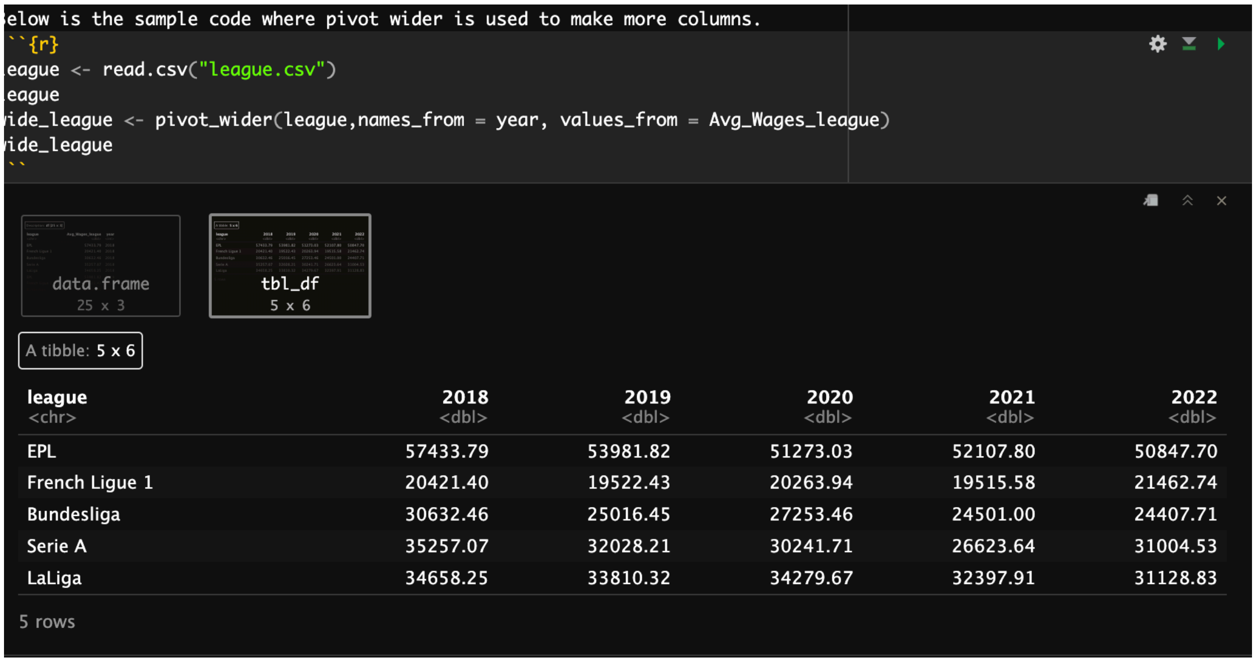
Below is the sample code where pivot wider is used to make more columns.

```
league <- read.csv("league.csv")
league
```

```
##             league Avg_Wages_league year
## 1              EPL         57433.79 2018
## 2  French Ligue 1         20421.40 2018
## 3      Bundesliga         30632.46 2018
## 4         Serie A         35257.07 2018
## 5          LaLiga         34658.25 2018
## 6              EPL         53981.82 2019
## 7  French Ligue 1         19522.43 2019
## 8      Bundesliga         25016.45 2019
## 9         Serie A         32028.21 2019
## 10         LaLiga         33810.32 2019
## 11            EPL         51273.03 2020
## 12 French Ligue 1         20263.94 2020
## 13     Bundesliga         27253.46 2020
## 14        Serie A         30241.71 2020
## 15         LaLiga         34279.67 2020
## 16            EPL         52107.80 2021
## 17 French Ligue 1         19515.58 2021
## 18     Bundesliga         24501.00 2021
## 19        Serie A         26623.64 2021
## 20         LaLiga         32397.91 2021
## 21            EPL         50847.70 2022
## 22 French Ligue 1         21462.74 2022
## 23     Bundesliga         24407.71 2022
## 24        Serie A         31004.53 2022
## 25         LaLiga         31128.83 2022
```

```
wide_league <- pivot_wider(league,names_from = year, values_from = Avg_Wages_league)
wide_league
```

```
## # A tibble: 5 x 6
##   league         `2018` `2019` `2020` `2021` `2022`
##   <chr>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 EPL            57434. 53982. 51273. 52108. 50848.
## 2 French Ligue 1 20421. 19522. 20264. 19516. 21463.
## 3 Bundesliga     30632. 25016. 27253. 24501. 24408.
## 4 Serie A        35257. 32028. 30242. 26624. 31005.
## 5 LaLiga         34658. 33810. 34280. 32398. 31129.
```



```
elow is the sample code where pivot wider is used to make more columns.
``{r}
eague <- read.csv("league.csv")
eague
ide_league <- pivot_wider(league,names_from = year, values_from = Avg_Wages_league)
ide_league
``
```

A tibble: 5 x 6

league <chr>	2018 <dbl>	2019 <dbl>	2020 <dbl>	2021 <dbl>	2022 <dbl>
EPL	57433.79	53981.82	51273.03	52107.80	50847.70
French Ligue 1	20421.40	19522.43	20263.94	19515.58	21462.74
Bundesliga	30632.46	25016.45	27253.46	24501.00	24407.71
Serie A	35257.07	32028.21	30241.71	26623.64	31004.53
LaLiga	34658.25	33810.32	34279.67	32397.91	31128.83

5 rows

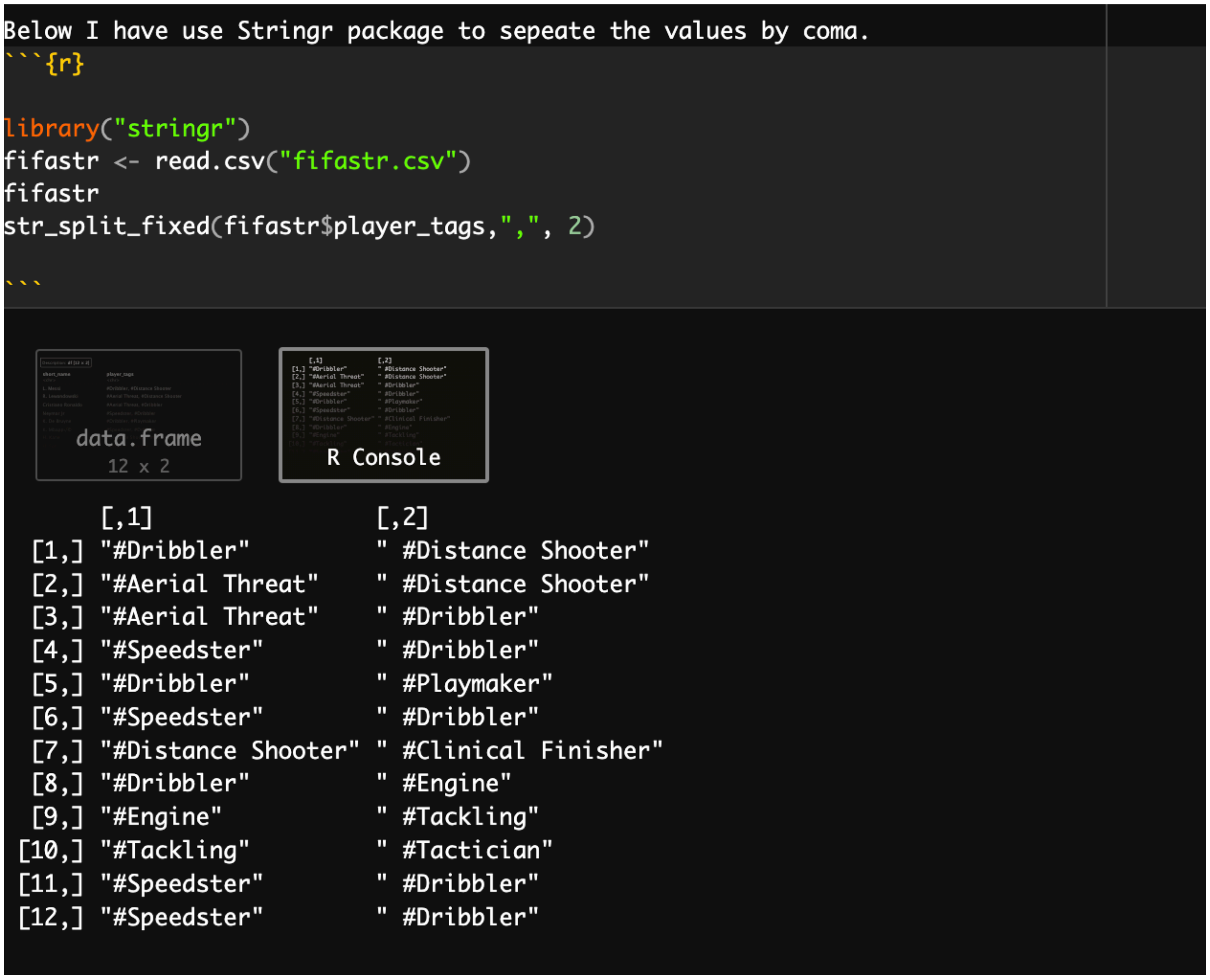
Below I have use Stringr package to sepeate the values by coma.

```r
library("stringr")
fifastr <- read.csv("fifastr.csv")
fifastr
```

```
##            short_name                          player_tags
## 1            L. Messi          #Dribbler, #Distance Shooter
## 2      R. Lewandowski     #Aerial Threat, #Distance Shooter
## 3   Cristiano Ronaldo             #Aerial Threat, #Dribbler
## 4           Neymar Jr                 #Speedster, #Dribbler
## 5         K. De Bruyne                 #Dribbler, #Playmaker
## 6           K. Mbappv©                 #Speedster, #Dribbler
## 7             H. Kane #Distance Shooter, #Clinical Finisher
## 8              H. Son                    #Dribbler, #Engine
## 9            Casemiro                    #Engine, #Tackling
## 10        V. van Dijk                 #Tackling, #Tactician
## 11           S. Manv©                 #Speedster, #Dribbler
## 12           M. Salah                 #Speedster, #Dribbler
```

```r
str_split_fixed(fifastr$player_tags,",", 2)
```

```
##       [,1]                 [,2]
##  [1,] "#Dribbler"          " #Distance Shooter"
##  [2,] "#Aerial Threat"     " #Distance Shooter"
##  [3,] "#Aerial Threat"     " #Dribbler"
##  [4,] "#Speedster"         " #Dribbler"
##  [5,] "#Dribbler"          " #Playmaker"
##  [6,] "#Speedster"         " #Dribbler"
##  [7,] "#Distance Shooter"  " #Clinical Finisher"
##  [8,] "#Dribbler"          " #Engine"
##  [9,] "#Engine"            " #Tackling"
## [10,] "#Tackling"          " #Tactician"
## [11,] "#Speedster"         " #Dribbler"
## [12,] "#Speedster"         " #Dribbler"
```



~~~
Below I have use Stringr package to sepeate the values by coma.
```{r}

library("stringr")
fifastr <- read.csv("fifastr.csv")
fifastr
str_split_fixed(fifastr$player_tags,",", 2)

```
~~~

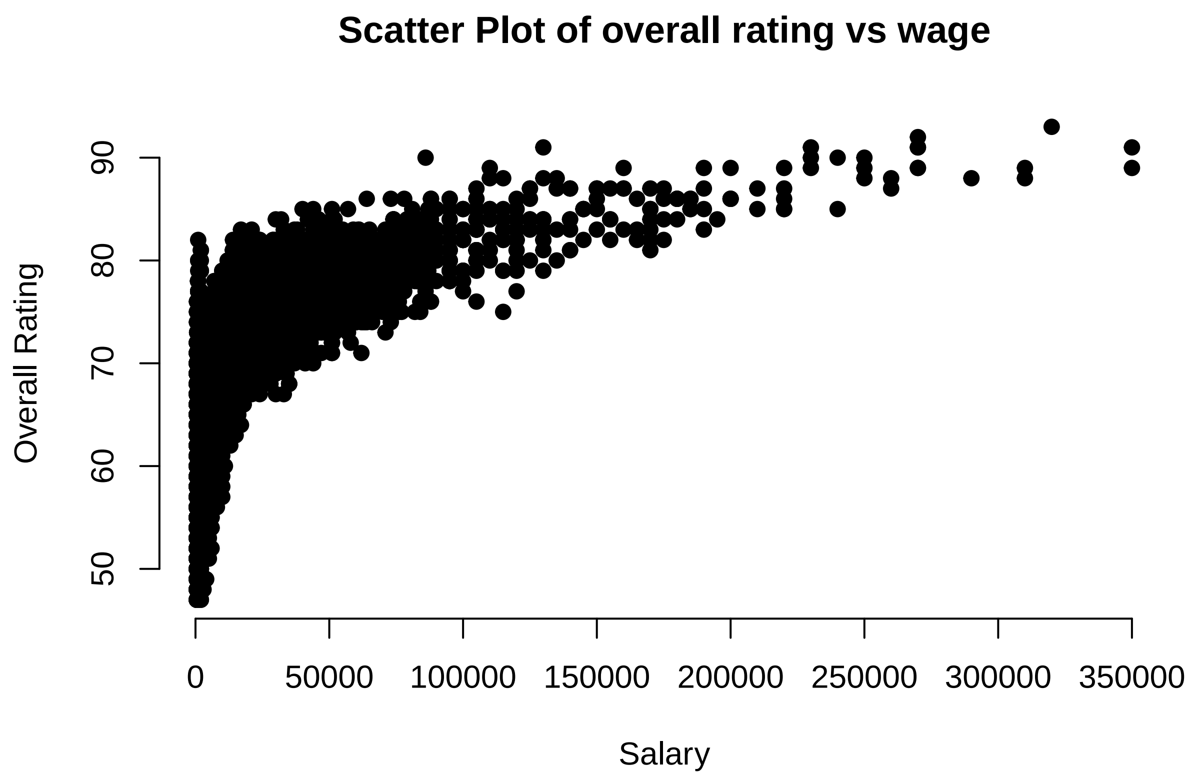
```
      [,1]                 [,2]
 [1,] "#Dribbler"          " #Distance Shooter"
 [2,] "#Aerial Threat"     " #Distance Shooter"
 [3,] "#Aerial Threat"     " #Dribbler"
 [4,] "#Speedster"         " #Dribbler"
 [5,] "#Dribbler"          " #Playmaker"
 [6,] "#Speedster"         " #Dribbler"
 [7,] "#Distance Shooter"  " #Clinical Finisher"
 [8,] "#Dribbler"          " #Engine"
 [9,] "#Engine"            " #Tackling"
[10,] "#Tackling"          " #Tactician"
[11,] "#Speedster"         " #Dribbler"
[12,] "#Speedster"         " #Dribbler"
```

Figure 2: shinyr

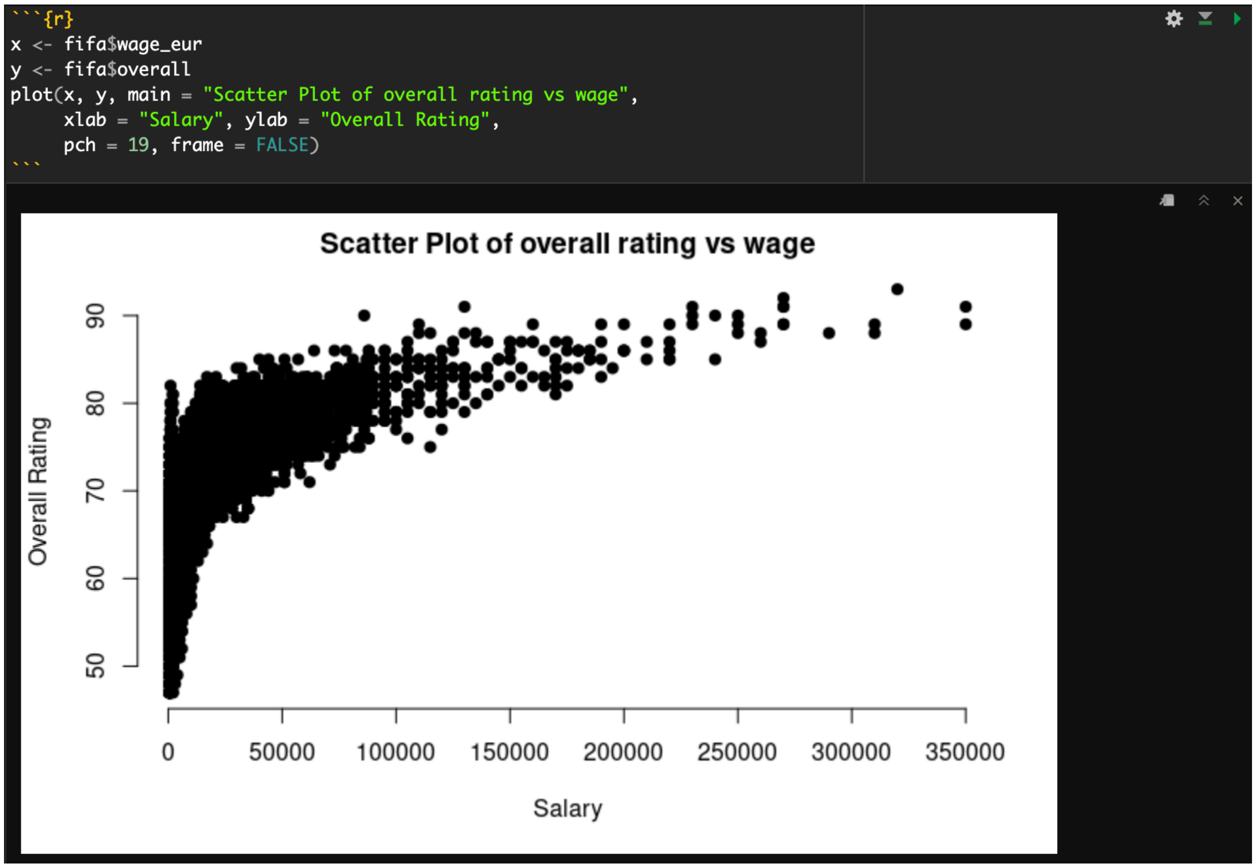
Q. Create graphical displays and numerical summaries of data for exploratory analysis and presentations.

ggplot2 is a popular package used to make graphical diagram such as bar graph, line graph and many more graphs.By using ggplot2 I have made Bar graph,line graph, and scatter plot.

```r
x <- fifa$wage_eur
y <- fifa$overall
plot(x, y, main = "Scatter Plot of overall rating vs wage",
     xlab = "Salary", ylab = "Overall Rating",
     pch = 19, frame = FALSE)
```

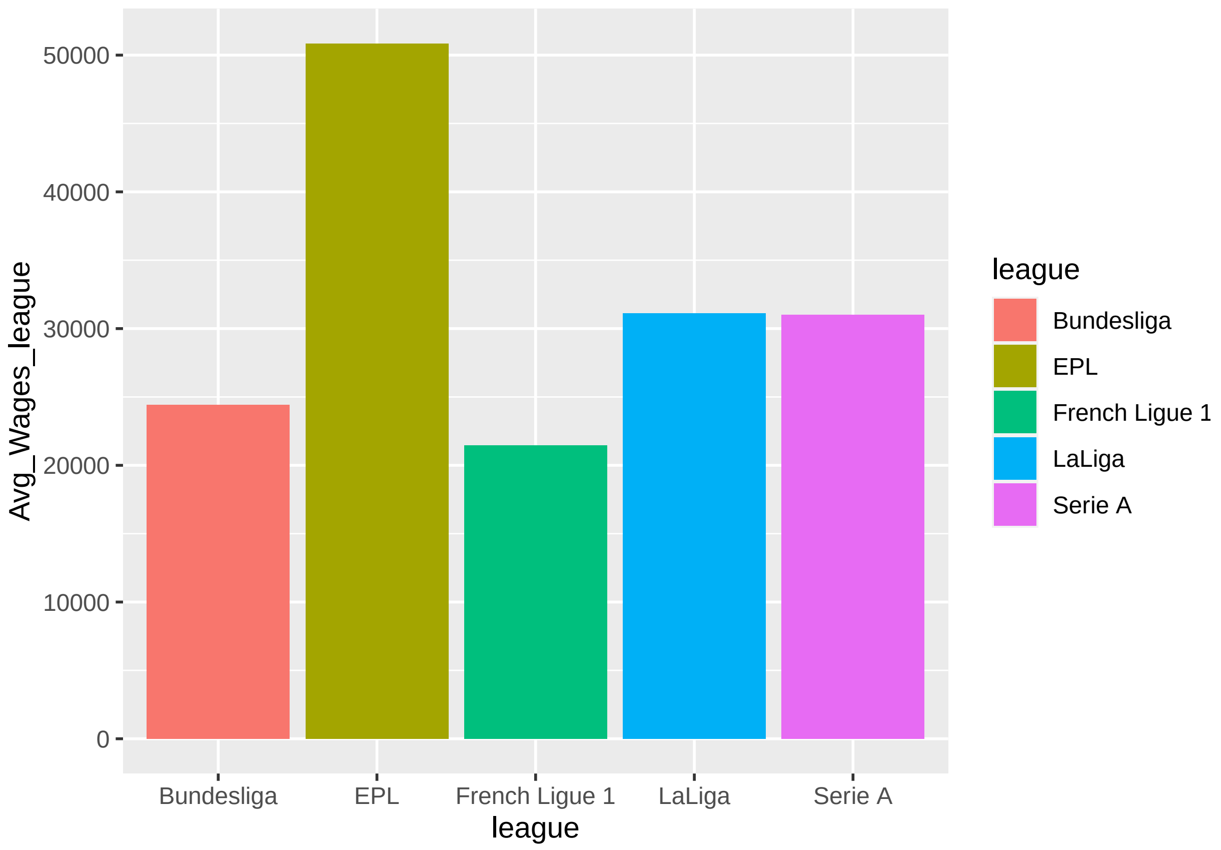


```r
bar<-read.csv("Bargraph.csv")
g <- ggplot(bar, aes( y = Avg_Wages_league, x = league,fill=league))
    g+ geom_bar(stat='identity') +scale_x_discrete(guide = guide_axis(check.overlap = TRUE))
```



```{r}
x <- fifa$wage_eur
y <- fifa$overall
plot(x, y, main = "Scatter Plot of overall rating vs wage",
     xlab = "Salary", ylab = "Overall Rating",
     pch = 19, frame = FALSE)
```

Figure 3: scatter



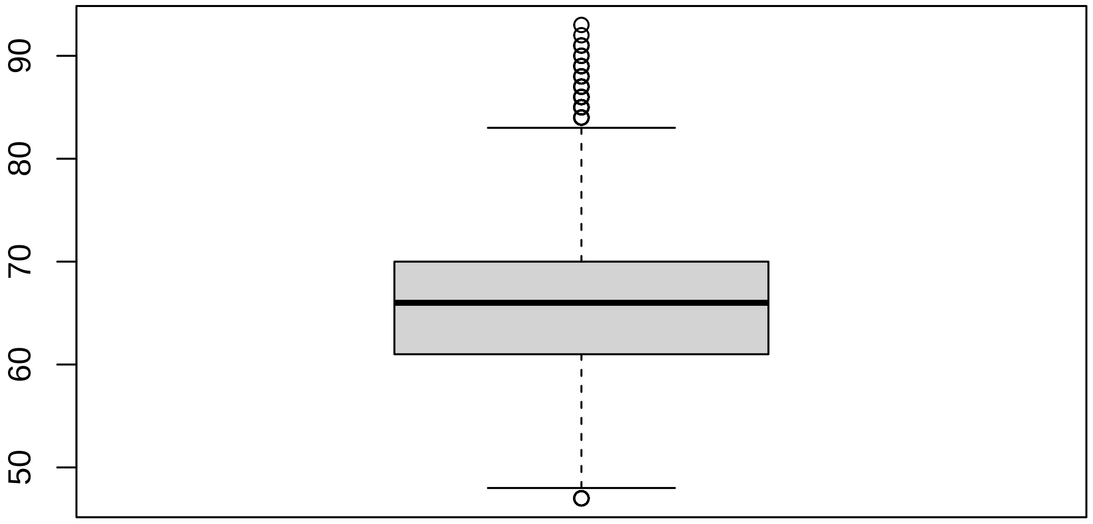
summary function is helpful in getting insight of the data where we get to know the median, maximun, minimum value.

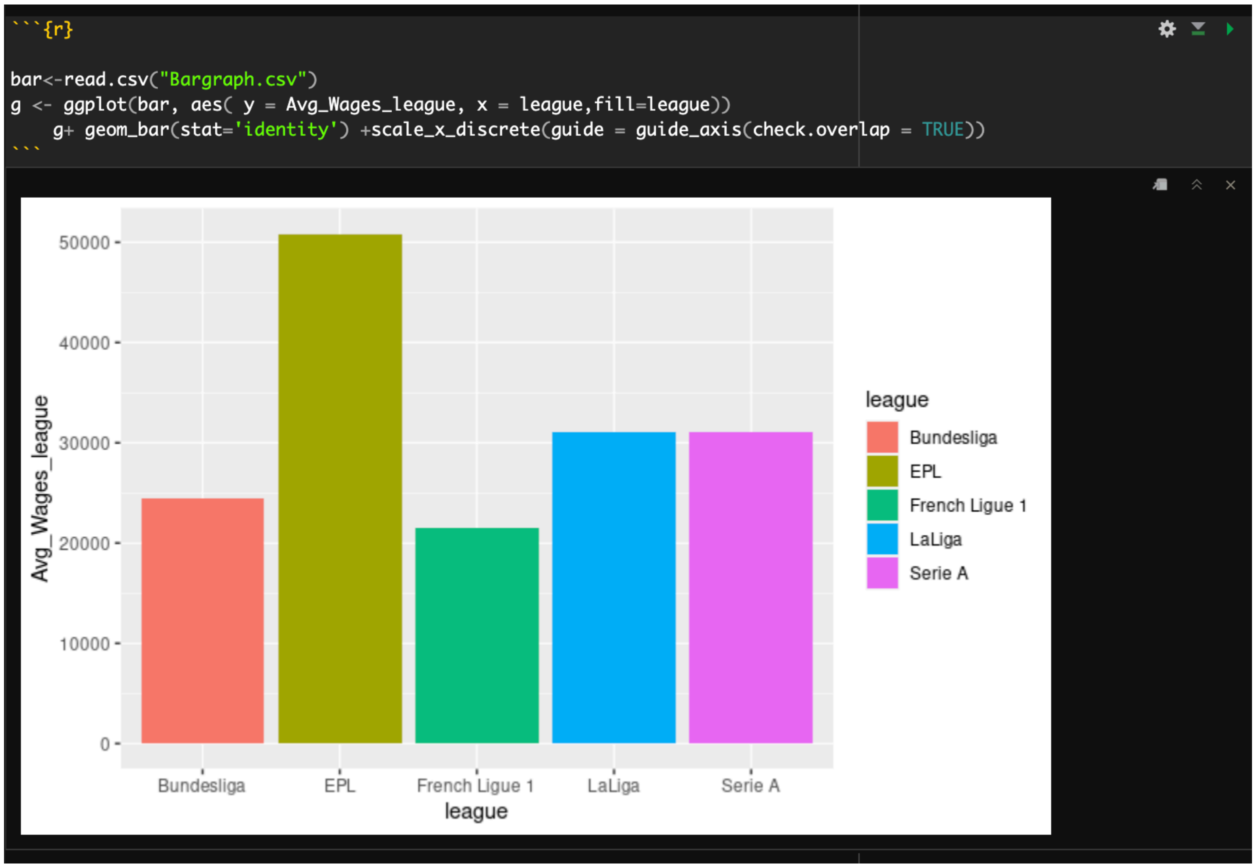
```
summary(fifa$overall)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.00   61.00   66.00   65.77   70.00   93.00
```

We can also know median, maximum, minimum values from box plot.

```
boxplot(fifa$overall)
```

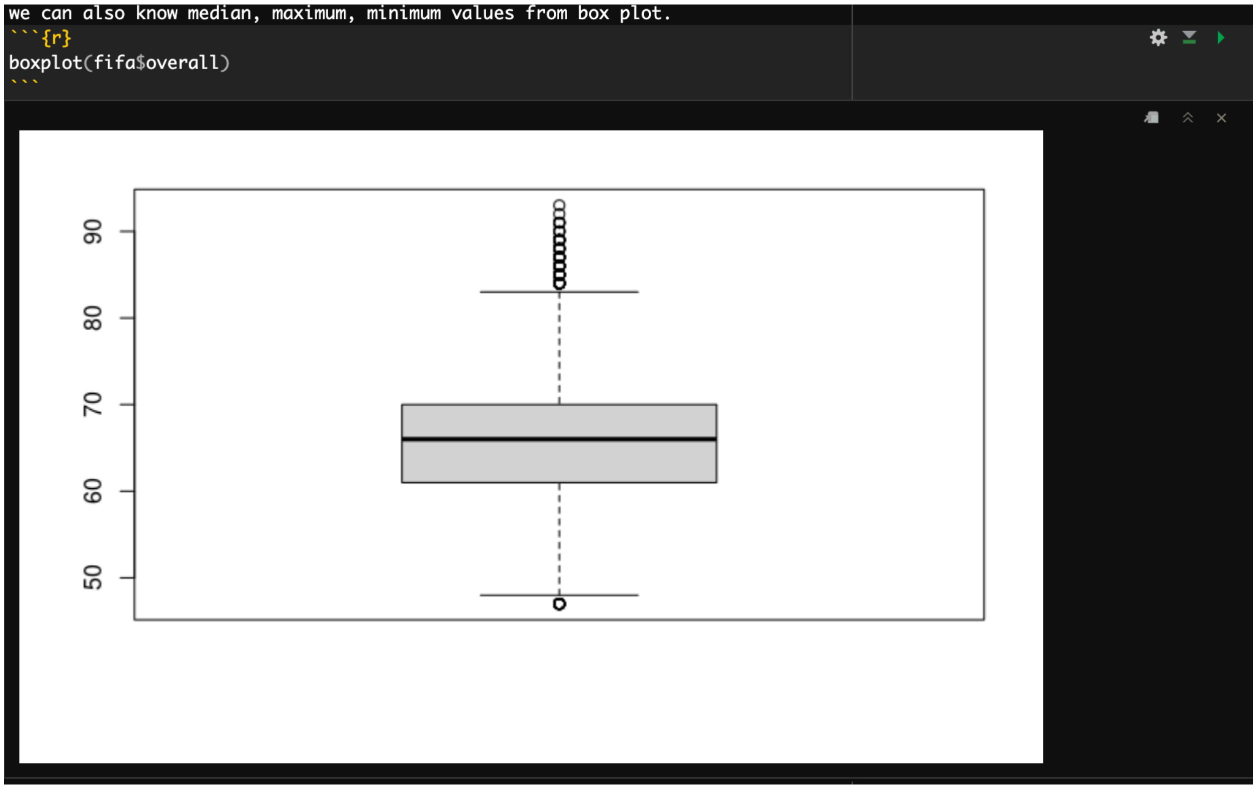




~~~
```{r}

bar<-read.csv("Bargraph.csv")
g <- ggplot(bar, aes( y = Avg_Wages_league, x = league,fill=league))
    g+ geom_bar(stat='identity') +scale_x_discrete(guide = guide_axis(check.overlap = TRUE))
```
~~~

Figure 4: bargraph



~~~
we can also know median, maximum, minimum values from box plot.
```{r}
boxplot(fifa$overall)
```
~~~

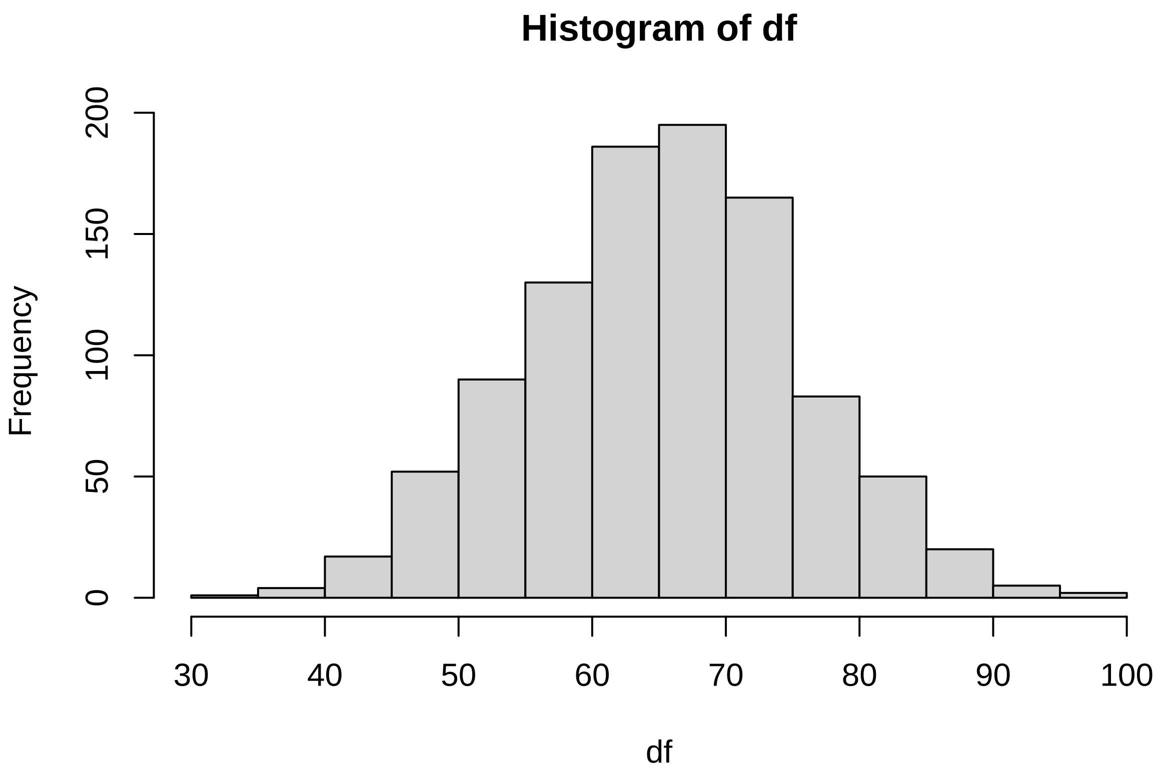
Figure 5: boxplot

Q. Analysis by simulation and bootstrapping

With simulation I can also create data similar to fifa overall dataset and perform hypothesis testing.

```r
df <- rnorm(1000, mean = 65, sd = 10)

hist(df)
```



Here we have create random normal variable with mean 65 and stranded deviation 10 and shown the data in histogram.

```
summary(df)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.97   58.60   65.42   65.29   72.11   96.28
```

Now lets create a sample from the df variable for our hypothesis testing. Where we will calculate the mean of the sample data and than compare it the df data frame mean. Our hypothesis would be H0: mean = 65.0928 and H1: mean   65.0928 .

```
set.seed(40)

sampledf = sample(df,100,replace = TRUE)
mean(sampledf)
```

```
## [1] 64.64108
```

```
sd(sampledf)
```

```
## [1] 9.867187
```

We can test the hypothesis by p-value with the mean and dataset of the sampledf to find if we accept the the hypothesis or fail to reject it at 95% coincidence inteval.

```
t.test(sampledf,y = NULL,  c("two.sided","less","greater"),mu = 65.0928, conf=0.95)
```

```
##
##  One Sample t-test
##
## data:  sampledf
## t = -0.4578, df = 99, p-value = 0.6481
## alternative hypothesis: true mean is not equal to 65.0928
## 95 percent confidence interval:
##  62.68322 66.59894
## sample estimates:
## mean of x
##  64.64108
```

So p value is 0.1133 and the significance level is 0.05 where p- value is greater than confidence level. Hence we fail to reject the null hypothesis which means we accept the hypothesis H0.

```
summary(fifa$overall)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.00   61.00   66.00   65.77   70.00   93.00
```

Lets perform hypothesis testing on actual fifa 22 dataset

```
set.seed(520)
sample1 = sample(fifa$overall,50,replace = FALSE)
mean(sample(fifa$overall,50,replace = FALSE))
```

```
## [1] 66.64
```

```
sd(sample1)
```

```
## [1] 6.630203
```

```
mean(fifa$overall)
```

```
## [1] 65.77218
```

our hypothesis would be mean(H0) = 65.77218 and other hypothesis would be H1: mean 65.77218.

```
t.test(sample1,y = NULL,  c("two.sided","less","greater"),mu = 65.77218, conf=0.95)
```

```
##
##  One Sample t-test
##
## data:  sample1
## t = 0.39228, df = 49, p-value = 0.6966
## alternative hypothesis: true mean is not equal to 65.77218
## 95 percent confidence interval:
##  64.25572 68.02428
## sample estimates:
## mean of x
##     66.14
```

So p value is 0.6966 and the significance level is 0.05 where p- value is greater than confidence level. Hence we fail to reject the null hypothesis which means we accept the hypothesis H0.

With the help of R I am also able to calculate the linear regression for the variable such as player wage and their overall rating.

```
fifa_reg <- lm(wage_eur~overall, data  = fifaxl)
summary(fifa_reg)
```

```
##
## Call:
## lm(formula = wage_eur ~ overall, data = fifaxl)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -157667  -20397    2872   34795  114795
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -517872    1260703  -0.411    0.687
## overall         8462      14007   0.604    0.554
##
## Residual standard error: 71420 on 16 degrees of freedom
## Multiple R-squared:  0.0223, Adjusted R-squared:  -0.03881
## F-statistic: 0.365 on 1 and 16 DF,  p-value: 0.5542
```

#bootstrapping: My plan here is to take sample from overall player data than perform bootstrapping to those data to know the distribution of mean along with their frequency.

```
set.seed(520)
sample1 = sample(fifa$overall,50,replace = FALSE)
n <- length(sample1)
Boot <- 10000
```

Here I have taken 50 sample of overall player data from fifa dataset where replace is false, which means the data from the same row is not selected. I have set seed because set seed allows us to get same set of random values every time while executing the code and we do not get different means and different output.

```
Bootsample <- matrix(sample(sample1, size = n*Boot, replace=TRUE), nrow = n, ncol= Boot)
```

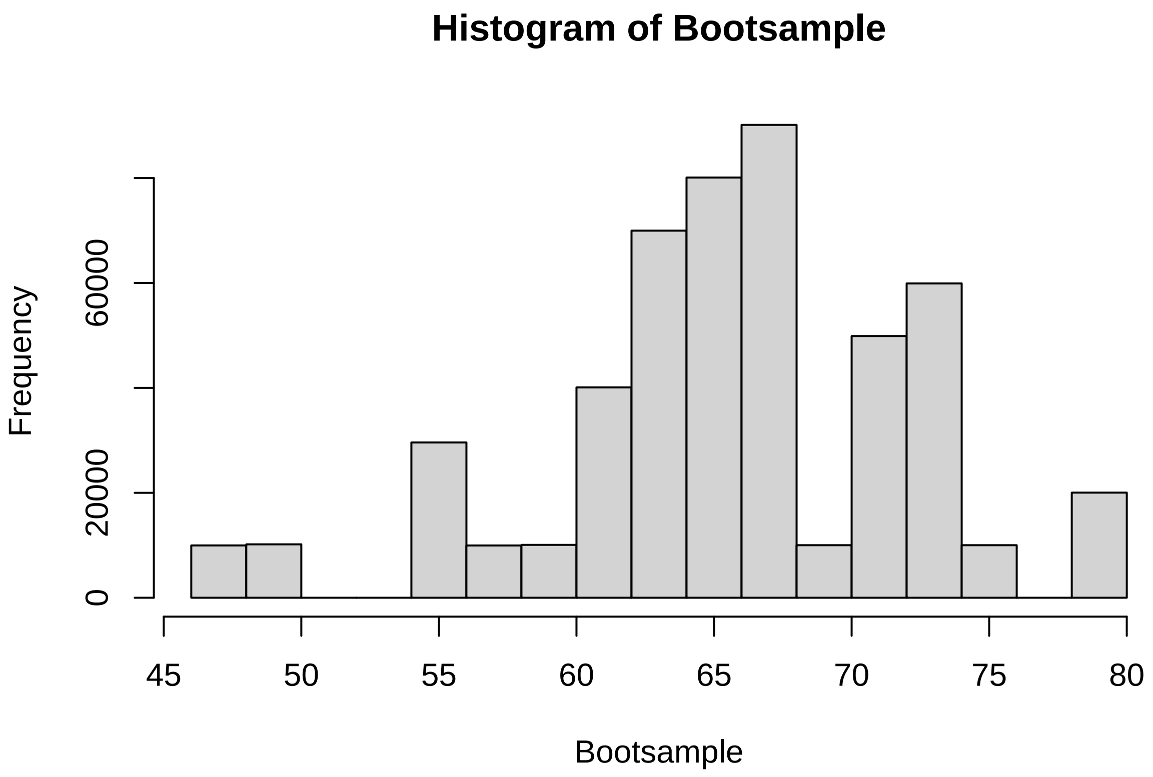
Here I have create a bootstrap which is 50 by 10000 matrix (where size = 50 * 10000). Where 50 is number of row and 10000 is number of columns. I have taken sample from the sample1 data set which is again sample of 50 data from fifa overall player dataset. for the sample taken from sample1 in the Bootsample I have set replace equals to true which means that repetition selection from the same rows are allowed while taking the sample to form the bootstrap.

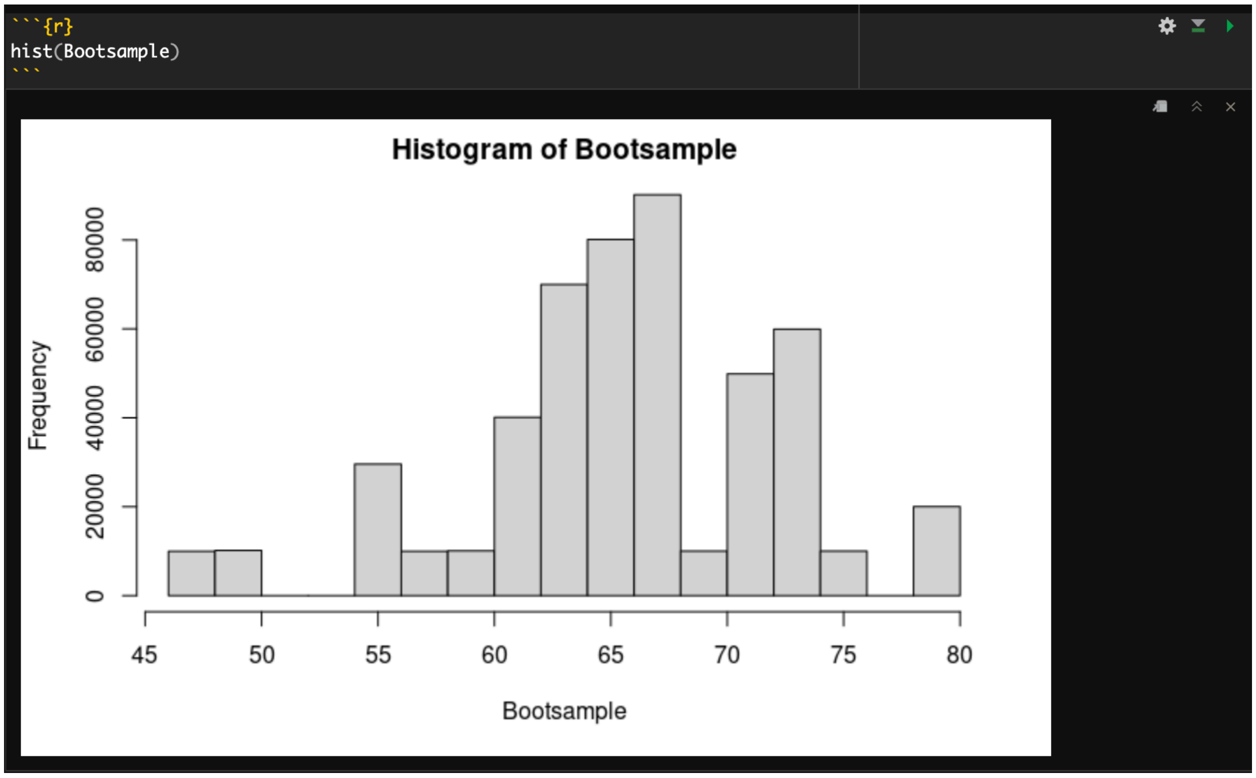
```
dim(Bootsample)
```

```
## [1]    50 10000
```

We can see that we have created Bootsample with 50 by 10000 matrix.

```
hist(Bootsample)
```





```{r}
hist(Bootsample)
```

Figure 6: hist

I have created a histogram of the the Bootsample to get more insight about the data.

```r
sort(sample1)
```

```
##  [1] 47 50 55 55 56 57 60 61 62 62 62 63 63 64 64 64 64 64 65 65 65 66 66 66 66
## [26] 66 67 67 67 67 68 68 68 68 68 70 71 71 71 71 72 73 73 73 73 74 74 76 79 80
```

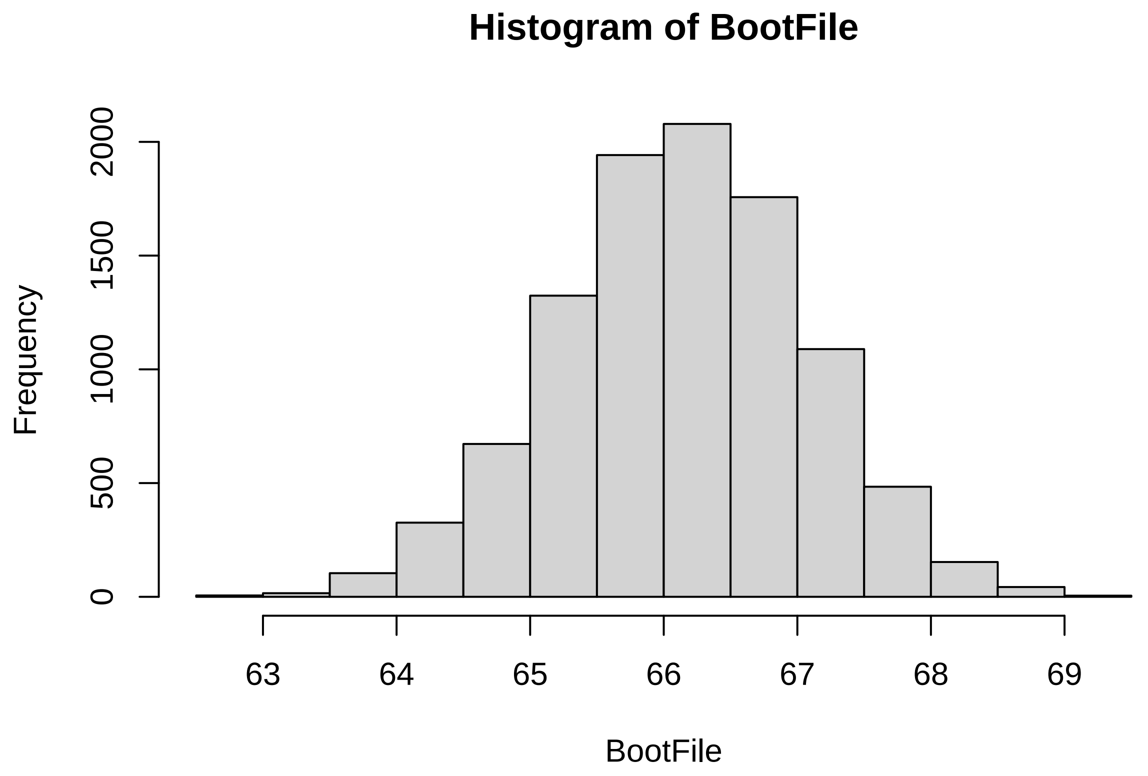
There is gap between 50 and 55 because in out data set sample1 we do not have the values between 50 and 55 and similar is the case between 76 and 79.

```r
BootFile <- rep(0,Boot)
for (i in 1:Boot){
  BootFile[i] <- mean(Bootsample[1:50,i])
}
```

I have create a vector with the name BootFile with 0 values. Again I have written code to calculate the mean from every 10000 columns and assign those values to BootFile.

This is the means of every 50 rows in the bootstrap.

```r
#mean of 10000 Bootstrapping
hist(BootFile)
```



Here is the histogram of the means from the bootstapping.

```r
sum(BootFile > 64 & BootFile < 68)
```

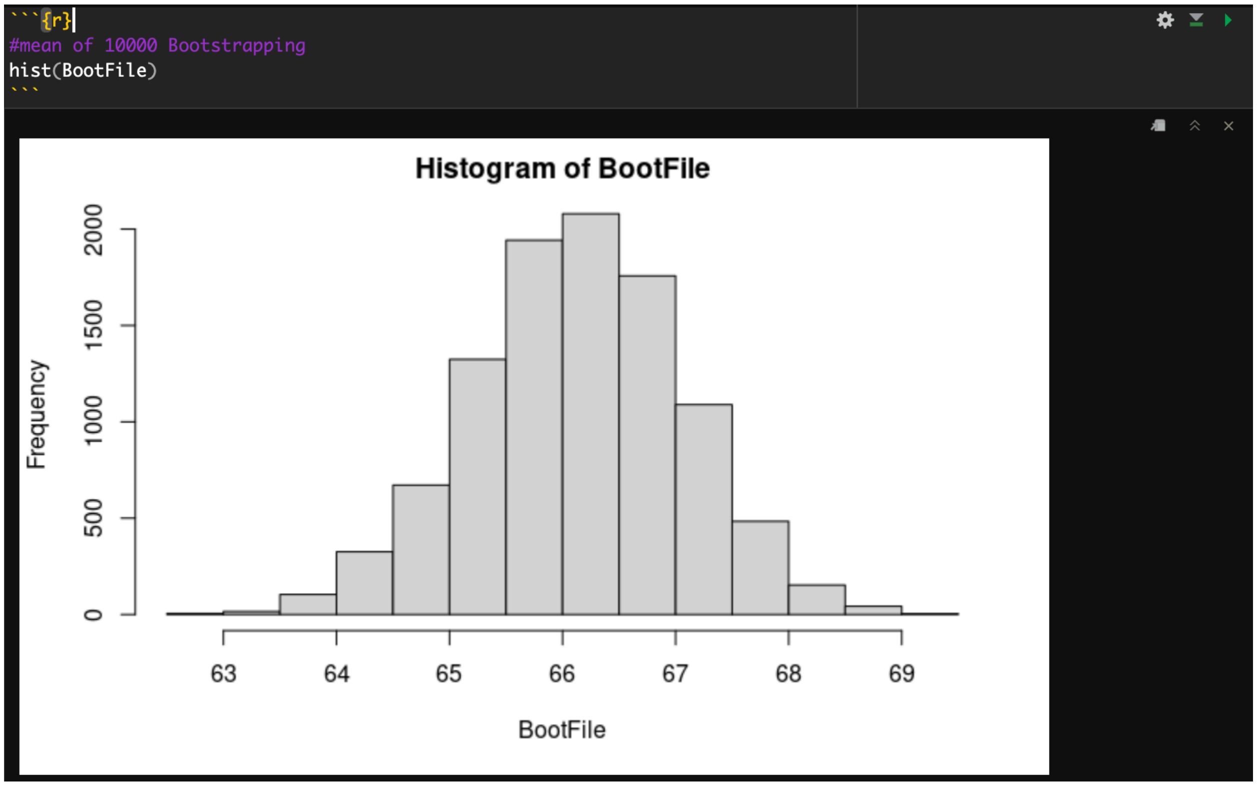
```
## [1] 9662
```

```r
sum(BootFile > 65 & BootFile < 67)
```

```
## [1] 7054
```

```r
sum(BootFile > 62.5 & BootFile < 69.5)
```

```
## [1] 9999
```



```{r}
#mean of 10000 Bootstrapping
hist(BootFile)
```

Figure 7: histboot

We can see that from the sample1 data when we calculate the the mean from the sample1 dataset when replace is true we can get mean between 62.5 and 69.5 but there is the 96.9% probability of getting mean between 64 and 68. We can also say that the data are normally distributed.

Q. Use source documentation and other resources to troubleshoot and extend R programs.

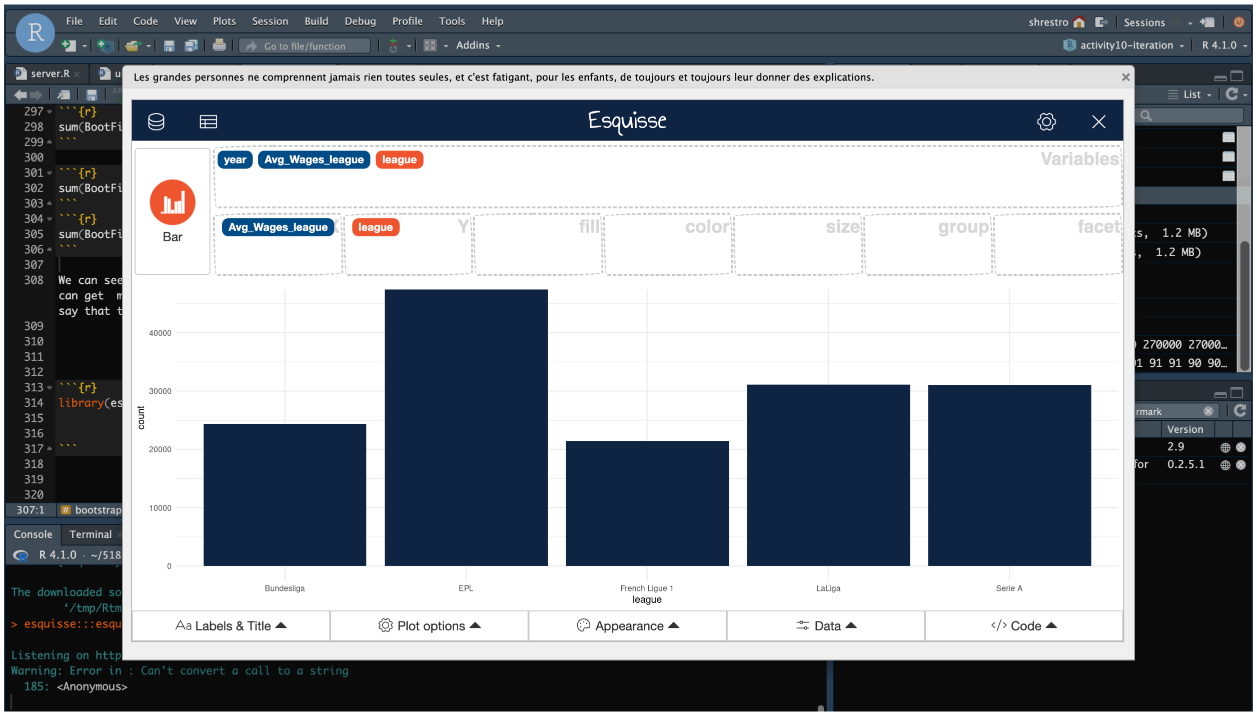
While learning, doing activity and project I had face my problem with the code and to solve it I refereed to r documentation, stackoverflow, and google to solve my issue. While solving my issue I got more insight in R and its syntax. Facing problem and reading documentation has help to develop my r knowledge and improve my understanding in R. Further I have also learn about many new packages like gganimate, esquisse and many more packages. Below is the snapshot of esquisse package used to make bar graph with drag and drop like tableau.

Q. Write clear, efficient, and well-documented R programs.

From the class activities and from R documentations online I have learned to write a well documented R code. I have also learn to write clean and effective code by using map, pivot, join, stringr, dplyr, tidyverse. The class activities has been effective and helped me grasp a knowledge of R. Additionally, working on my final project has strengthen my knowledge on working with R and Rmd files.

Question 4: Based on the progress you have made (i.e., see your response in (3)), what final grade would you give yourself for this course? Try to stick to the major grade levels (“A”, “B”, “C”, or “D or below”). Please reach out to me if you have concerns or were unable to finish your final project.

Since I have met all the objective, done the project, class activities and implemented it in this rmd file as well as in shiny app. I have given my best to learn R programming and performed many hundred line of code and learned from fixing the error while writing the code. Hence I will rate myself an “A” for the progress made through this semester.



Les grandes personnes ne comprennent jamais rien toutes seules, et c'est fatigant, pour les enfants, de toujours et toujours leur donner des explications.

~~~
```{r}
sum(BootFi
~~~

```{r}
sum(BootFi
```
```{r}
sum(BootFi
```

We can see
can get m
say that t

```{r}
library(es


```
```

```
307:1  bootstrap

Console  Terminal
R 4.1.0 · ~/518

The downloaded so
      '/tmp/Rtm
> esquisse:::esqu

Listening on http
Warning: Error in : Can't convert a call to a string
  185: <Anonymous>
```

```
s,  1.2 MB)
,  1.2 MB)

) 270000 27000…
1 91 91 90 90…
```

Figure 8: esquisse

Question 5: Do you have any other thoughts or reflections about the course that you'd like to share?

R is very powerful tool in doing statistical analysis and that is why I wanted to learn R programming. Hence my further plan would be to continue learn R and implement thing learned from this course in performing data analysis. Additionally, I would like to make interactive site similar like our final project and creating my own portfolio in R. Thank for showing me path to be better at R program and would also like to Thank you for your contribution.
```