The Quantitative Data Analysis of the European Football Players

Hawra Nawrozzadeh

11 September, 2023

0. Introduction

This data analysis on dataset of European Football Players will include 4 sections: Data Quality and Cleaning, Explanatory Data Analysis (EDA), and statistical analysis for research questions in section 3 (building a model for player potential) and 4 (building a model for the variable high wage indicator), based on this metadata:

Column Name	Column Description
sofifa_id	Player ID code
potential	player potential overall attribute – measured on a scale 0-100
wage_eur	weekly player wage in Eur
age	player age
height_cm	Player height in cm
weight_kg	Player weight in Kg
club_name	Name of the player's club
preferred_foot	player preferred foot
pace	player pace attribute – measured on a scale 0-100
shooting	player shooting attribute– measured on a scale 0-100
passing	player passing attribute– measured on a scale 0-100
dribbling	player dribbling attribute– measured on a scale 0-100
defending	player defending attribute– measured on a scale 0-100
physic	player physic attribute– measured on a scale 0-100
power_strength	player strength attribute– measured on a scale 0-100
power_long_shots	player long shots attribute– measured on a scale 0-100
high.wage.ind	Binary variable based on weekly wage - Is weekly wage above 8000 Euro

0.1 Loading the Libraries

The following R library packages will be utilised to perform the data analysis:

```r
# For data organisation and data cleaning
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
```

```r
# For Data Visualisation
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: ’gridExtra’

## The following object is masked from ’package:dplyr’:
##
##     combine
```

```r
library(grid)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(reshape2)
```

1. Organise and clean the data

1.1 Loading the data

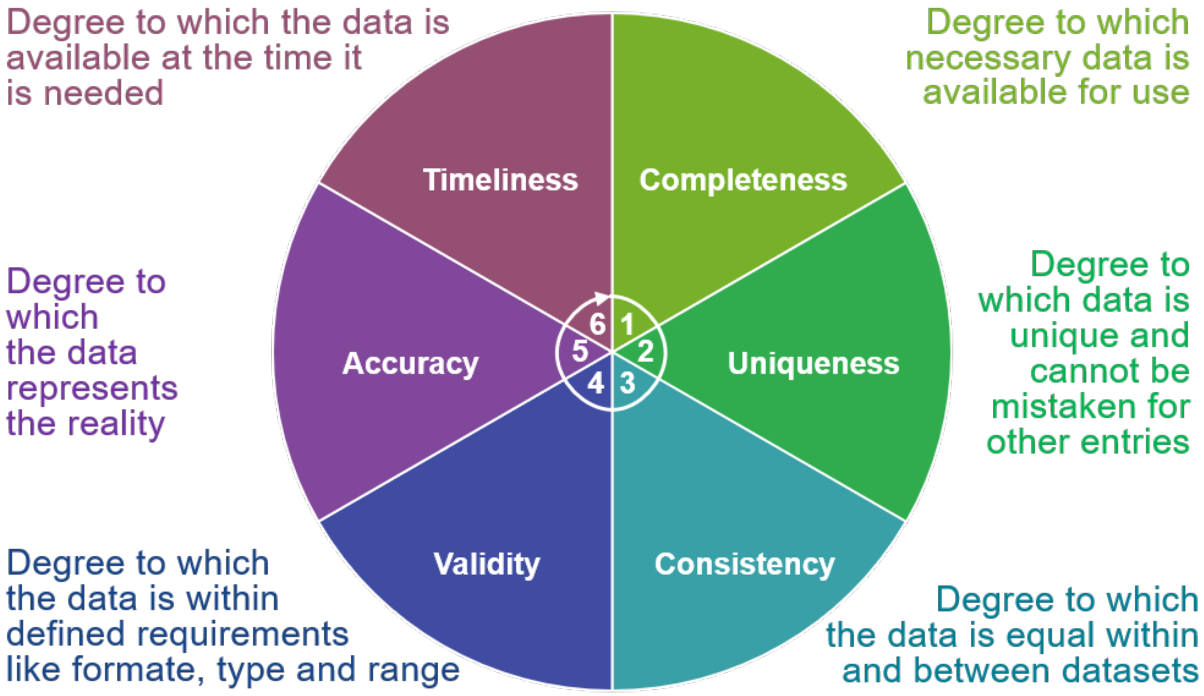
The dataset is read and loaded with the following code:

```r
football <- read.csv("Data/football.csv", stringsAsFactors = FALSE)
```

1.2 Data quality analysis

Data quality analysis is an essential process to optimise the data and statistics. The quality of the dataset refers to accuracy, consistency, completeness and reliability of the data (see framework below). Measuring and checking the quality of the data identifies potential errors, that should be resolved appropriately, to prevent misleading results and conclusions (1,2,3).

The analysis will involve looking at the summary of the data, to check and confirm which variables are numerical continuous or categorical, and if the categorical are stored as factors. This will also check for outliers, data which does not fit within the range of data being observed, including any missing data. Visualising the data will check for any other unusual behaviour or patterns, where appropriate.



Degree to which the data is available at the time it is needed
Degree to which necessary data is available for use
Timeliness
Completeness
Degree to which the data represents the reality
Accuracy
6
1
5
2
4
3
Uniqueness
Degree to which data is unique and cannot be mistaken for other entries
Validity
Consistency
Degree to which the data is within defined requirements like formate, type and range
Degree to which the data is equal within and between datasets

1.2.1 Eyeballing and confirming the data types and classification of the data set

Firstly, the structure of the dataset provided is examined by using the `str()` function. Afterwards,`table()` is used to check the different levels of the categorical variables.

```r
# To find the structure of 'football' data frame
str(football)
```

```
## 'data.frame':    514 obs. of  17 variables:
##  $ sofifa_id        : int  231747 212218 188377 235790 211300 183512 207410 157481 208418 177458 ...
##  $ potential        : int  95 90 85 93 88 83 86 82 82 81 ...
##  $ wage_eur         : num  160000 200000 170000 105000 155000 35000 120000 41000 68000 66000 ...
##  $ age              : int  21 26 30 21 24 30 26 34 26 31 ...
##  $ height_cm        : int  178 189 183 188 184 181 176 187 181 186 ...
##  $ weight_kg        : int  73 85 70 83 76 80 80 74 72 81 ...
##  $ club_name        : chr  "Paris Saint-Germain" "Manchester City" "Manchester City" "Chelsea" ...
##  $ preferred_foot   : chr  "Right" "Left" "Right" "Left" ...
##  $ pace             : int  96 63 92 84 89 83 75 45 91 66 ...
##  $ shooting         : int  86 50 63 81 83 68 69 41 81 52 ...
##  $ passing          : int  78 72 76 79 74 76 83 62 75 63 ...
##  $ dribbling        : int  91 68 77 85 87 76 88 63 84 60 ...
##  $ defending        : int  39 88 80 45 41 80 69 86 36 84 ...
##  $ physic           : int  76 81 82 67 72 83 70 75 65 77 ...
##  $ power_strength   : int  76 85 79 69 76 80 63 81 59 76 ...
##  $ power_long_shots: int  79 47 69 78 79 76 75 53 82 62 ...
##  $ high.wage.ind    : int  1 1 1 1 1 1 1 1 1 1 ...
```

```r
# To see the different levels of the `preferred_foot` and `high.wage.ind` (as table() gives the frequen
table(football$high.wage.ind)
```

```
##
##   0   1
## 363 151
```

```r
# Lets view the different football club that exist in this dataset
View(table(football$club_name))
```

As illustrated, there are 514 observations and 17 variables, indicating that most of the continuous numerical variables are integers (whole numbers). The exception is **'wage_eur'**, which is numeric (or double), possibly suggesting some presence of decimal data values. This also confirms that **'club_name'**, **'preferred_foot'** and **'high.wage.ind'** are categorical, where **'high.wage.ind'** is a binary variable, containing only two levels, *1* and *0*. Using `View()` confirms that this dataset holds data on 341 different European Football Clubs.

1.2.2 Checking for unusual or abnormal behaviour

`summary()` provides some of the measurements of both central tendency and variability, to obtain further detail:

```r
summary(football)
```

```
##    sofifa_id        potential         wage_eur            age
##  Min.   :104476   Min.   :54.00   Min.   :     6   Min.   :16.00
##  1st Qu.:211483   1st Qu.:67.00   1st Qu.:  1000   1st Qu.:21.00
##  Median :232608   Median :71.00   Median :  4000   Median :25.00
##  Mean   :227195   Mean   :71.66   Mean   : 10810   Mean   :25.19
##  3rd Qu.:246961   3rd Qu.:75.00   3rd Qu.: 10750   3rd Qu.:29.00
##  Max.   :258945   Max.   :95.00   Max.   :200000   Max.   :70.00
##    height_cm       weight_kg       club_name         preferred_foot
##  Min.   :162.0   Min.   : 60.00   Length:514         Length:514
##  1st Qu.:176.0   1st Qu.: 70.00   Class :character   Class :character
##  Median :180.0   Median : 74.00   Mode  :character   Mode  :character
##  Mean   :180.1   Mean   : 74.28
##  3rd Qu.:184.0   3rd Qu.: 78.00
##  Max.   :214.0   Max.   :161.00
##       pace          shooting         passing        dribbling
##  Min.   :-81.00   Min.   :22.00   Min.   :29.00   Min.   :-57.00
##  1st Qu.: 62.00   1st Qu.:44.00   1st Qu.:51.00   1st Qu.: 59.00
##  Median : 68.00   Median :54.50   Median :58.00   Median : 64.00
##  Mean   : 67.78   Mean   :53.21   Mean   :57.89   Mean   : 63.31
##  3rd Qu.: 75.00   3rd Qu.:63.00   3rd Qu.:65.00   3rd Qu.: 70.00
##  Max.   : 96.00   Max.   :86.00   Max.   :83.00   Max.   : 91.00
##    defending         physic      power_strength  power_long_shots
##  Min.   :16.00   Min.   :37.00   Min.   :32.00   Min.   :16.00
##  1st Qu.:36.00   1st Qu.:58.00   1st Qu.:57.00   1st Qu.:41.00
##  Median :55.00   Median :65.00   Median :66.00   Median :54.00
##  Mean   :51.12   Mean   :64.36   Mean   :64.96   Mean   :51.94
##  3rd Qu.:64.00   3rd Qu.:72.00   3rd Qu.:74.00   3rd Qu.:64.00
##  Max.   :88.00   Max.   :86.00   Max.   :92.00   Max.   :82.00
##  high.wage.ind
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.2938
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

Issues discovered includes **'preferred_foot'** and **'high.wage.ind'**, which are not considered as factors, and therefore do not show the different levels of categories. All players' attributes should be between 0 to 100, however, **'pace'** and **'dribbling'** are two attributes which are not within this range, both containing negative values. Furthermore, there seems to be an unusual range of values for **'wage_eur'**, **'age'**, **'height_cm'** and **'weight_kg'**. The information regarding the range is evident when observing the minimum and maximum values of each variable from this output.

The `filter()` function from dyplr package will be used to confirm that **'high.wage.ind'** only contains 1 and 0 [5]:

```
# Using the 'filter()' function to exact any values that are not 0 or 1
football %>%
  filter(!(high.wage.ind == 0 | high.wage.ind == 1))    # the output for this is empty, confirming that
```

```
##  [1] sofifa_id        potential        wage_eur         age
##  [5] height_cm        weight_kg        club_name        preferred_foot
##  [9] pace             shooting         passing          dribbling
## [13] defending        physic           power_strength   power_long_shots
## [17] high.wage.ind
## <0 rows> (or 0-length row.names)
```

```
# Note: The `%>%` is the pipe function that utilities the dyplr's functions
```

This output is empty, confirming that is this a binary variable.

`table()` function will be used to check that **'preferred_foot'** consists of the two correct levels, *"Right"* and *"Left"*

```
table(football$preferred_foot)
```

```
##
##  Left right Right
##   136     1   377
```

As it shows there is input that has *'right'* instead of *'Right'*.

1.2.3 Checking for any missing values

Missing values are often expressed as **null**, **NA**, or even **empty value**, all meaning the same. `colSums()` and `which()`, with the help of `is.null()` and `is.na()` will be used to extract any possible missing values as R can detect these in different ways.

```
# To see if there is any NA within each variable (and if so how many in each variable)
colSums(is.na(football))
```

```
##        sofifa_id        potential         wage_eur              age 
##                0                0                0                0 
##        height_cm        weight_kg        club_name   preferred_foot 
##                0                0                0                0 
##             pace         shooting          passing        dribbling 
##                0                0                0                0 
##        defending           physic   power_strength power_long_shots 
##                0                0                0                0 
##    high.wage.ind 
##                0
```

```
# To see if there is any rows contain NULL
which(is.null(football))
```

```
## integer(0)
```

```
# To see if there is any empty rows
football[which(football == " "), c(1:17)]
```

```
##  [1] sofifa_id        potential        wage_eur         age             
##  [5] height_cm        weight_kg        club_name        preferred_foot  
##  [9] pace             shooting         passing          dribbling       
## [13] defending        physic           power_strength   power_long_shots
## [17] high.wage.ind   
## <0 rows> (or 0-length row.names)
```

As the output indicates, there are no missing values at all, which is a good sign.

1.2.4 Identifying other outliters

`table()` will identify outliers in **‘age’**, **‘wage__eur’**, **height__cm**, and **weight__cm**

```
# Getting the frequencies for  wage_eur
table(football$wage_eur)
```

```
## 
## 6.0001    500    550    600    650    700    750    800    850    900    950 
##      1     62      7      6      1      5      4      4     12      7      3 
##   1000   2000   3000   4000   5000   6000   7000   8000   9000  10000  11000 
##     40     68     35     25     36     19     17     11     13      9      8 
##  12000  13000  14000  15000  16000  17000  18000  19000  20000  21000  22000 
##     13      5      5      3      8      4      4      4      4      3      2 
##  23000  24000  26000  27000  28000  29000  30000  31000  32000  34000  35000 
##      2      3      3      4      4      4      3      1      1      1      3 
##  36000  38000  41000  42000  45000  46000  47000  48000  49000  50000  51000 
##      2      1      3      1      1      1      1      1      2      2      3 
##  55000  58000  59000  60000  64000  66000  68000  74000  95000  1e+05 105000 
##      1      1      2      1      1      2      1      2      1      1      1 
## 120000 155000 160000 170000  2e+05 
##      1      1      1      1      1
```

```
# Now for age
table(football$age)
```

```
##
## 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 70
##  1  9 24 25 44 36 27 40 45 40 38 26 26 35 23 18 16 14 10  6  6  1  1  2  1
```

```
# For height_cm
table(football$height_cm)
```

```
##
## 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181
##   1   1   1   4   1   2   5   5  17   9  18  15  14  33  25  29  38  30  30  15
## 182 183 184 185 186 187 188 189 190 191 193 194 195 196 198 214
##  27  35  31  22  22  17  24  12  11   7   5   2   2   1   2   1
```

```
# For weight_cm
table(football$weight_kg)
```

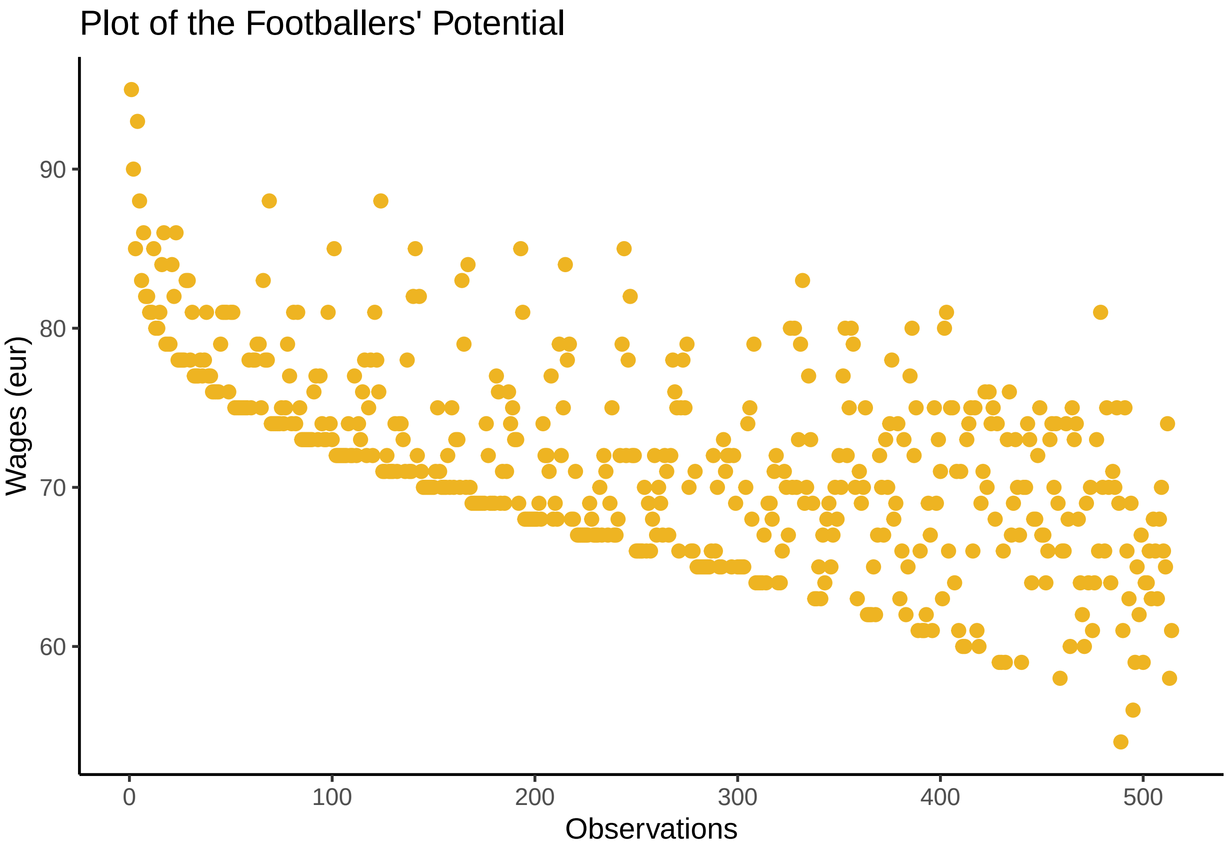
```
##
##  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79
##   8   4   9   7   7  16  15  15  24  15  39  23  27  35  28  41  20  29  26  20
##  80  81  82  83  84  85  86  87  88  89  91  92  93  94 161
##  25   6  17  11  13   8   8   3   6   2   2   1   2   1   1
```

As indicated, **'wage__eur'** has one outlier, being a very small decimal and **'age'** has one outlier that is a 70 year old player. Additionally, **'wage__eur'** does have two values that are in the form of exponent values, but these are not outliers. The outlier for height and weight have unusual values which seem to be the maximum value, and does not fit in the respective range.

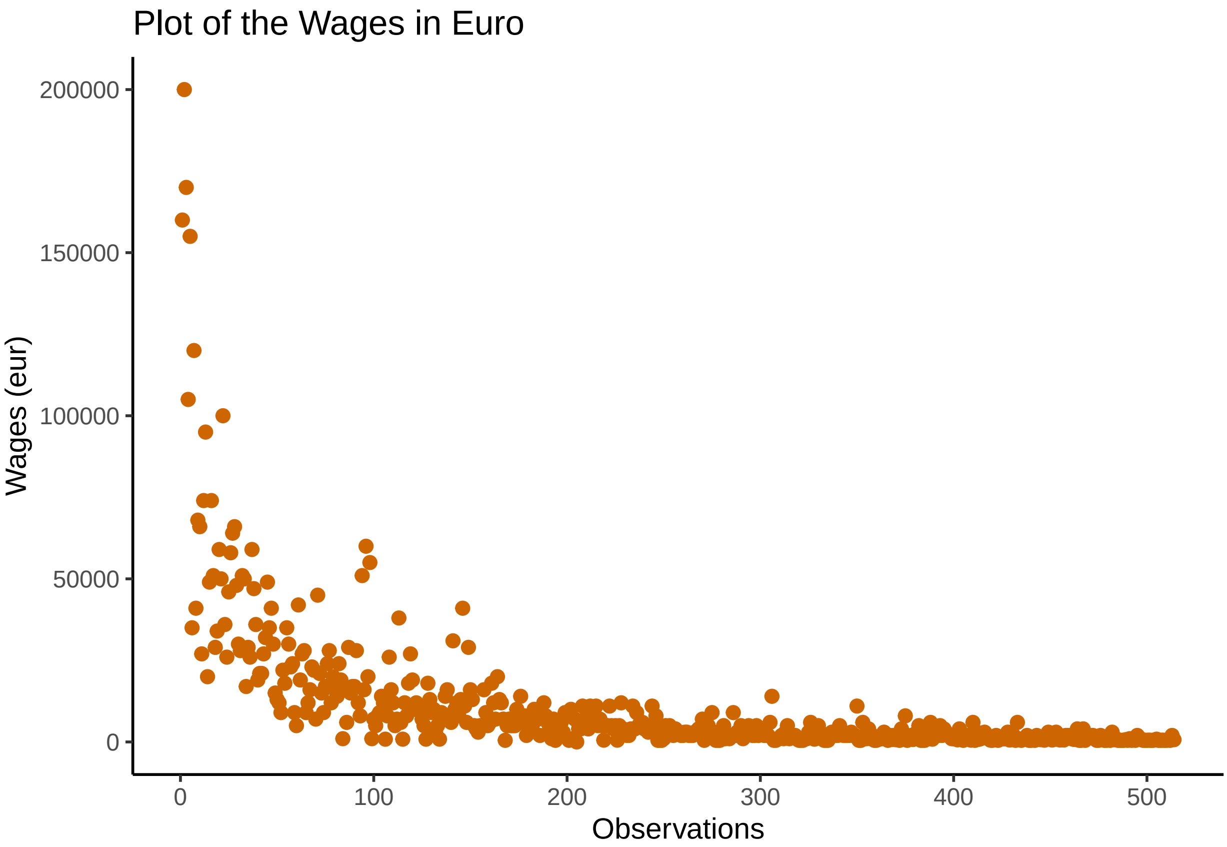
1.2.5 Data Visualisation to look out for any other errors or patterns

Visualisation of the numerical variable will highlight any other unusual behaviour or patterns, using a very powerful and dynamic graphical package called ggplot2[6]. Sub-setting the data that contains only the continuous variables will help visualise these variables 7, 8, 9.

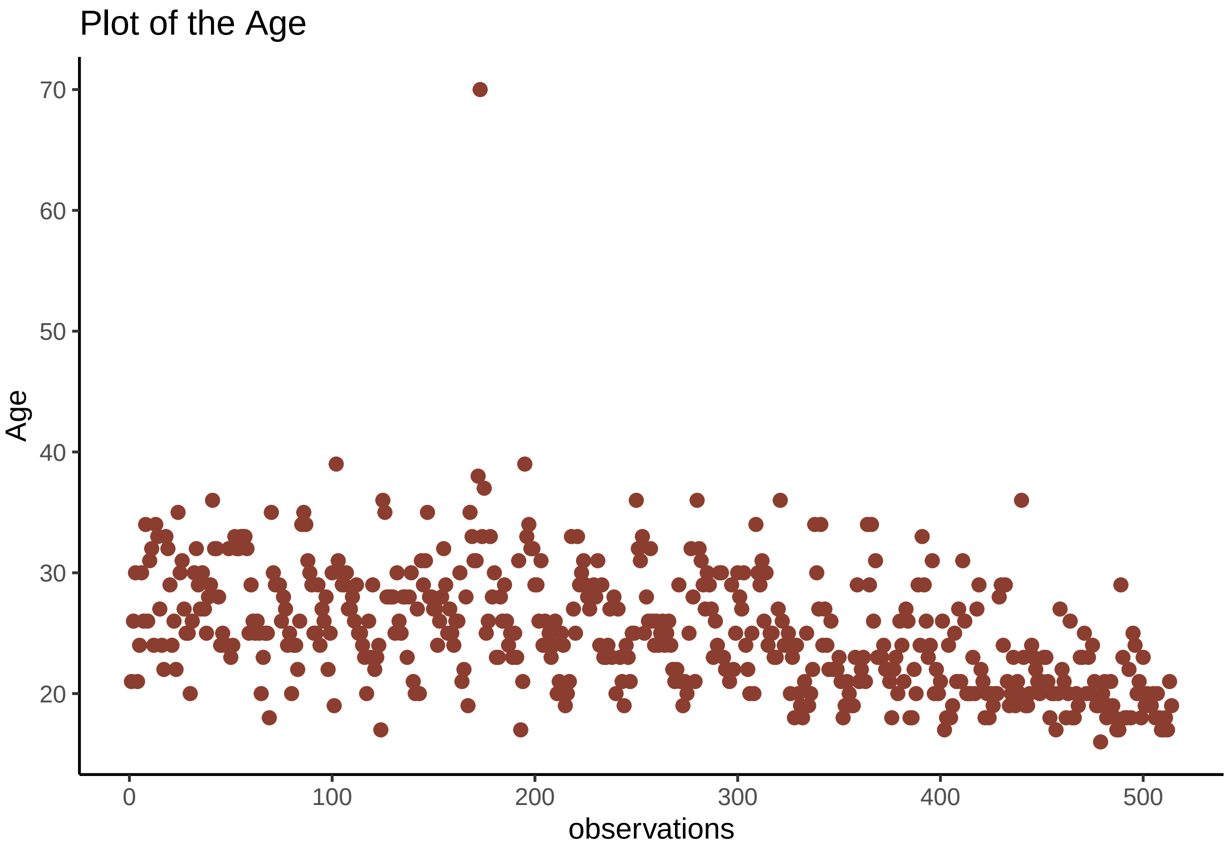
```
# Lets make a subset data frame with all the outliers that we see far
football_num <- football %>%
  select(-c("sofifa_id", "club_name", "preferred_foot"))

# Plot and visualise the `potential` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = potential)) +      # Getting all the data points
  geom_point(size=2, color = "goldenrod2") +
  ggtitle("Plot of the Footballers' Potential") +
  xlab("Observations") + ylab("Wages (eur)") +
  theme_classic()
```



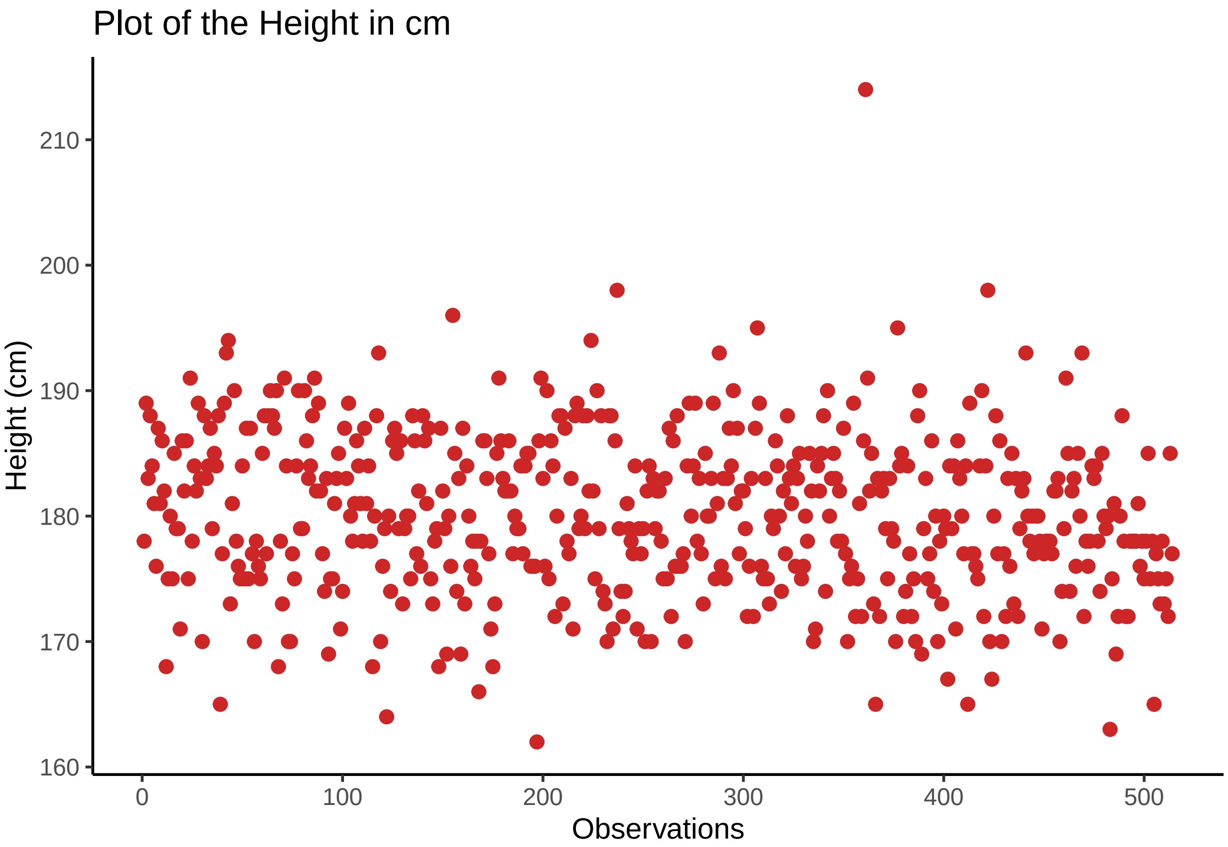
```
# Plot and visualise the `wage_eur` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = wage_eur)) +
  geom_point(size=2, color = "darkorange3") +
  ggtitle("Plot of the Wages in Euro") +
  xlab("Observations") + ylab("Wages (eur)") +
  theme_classic()
```



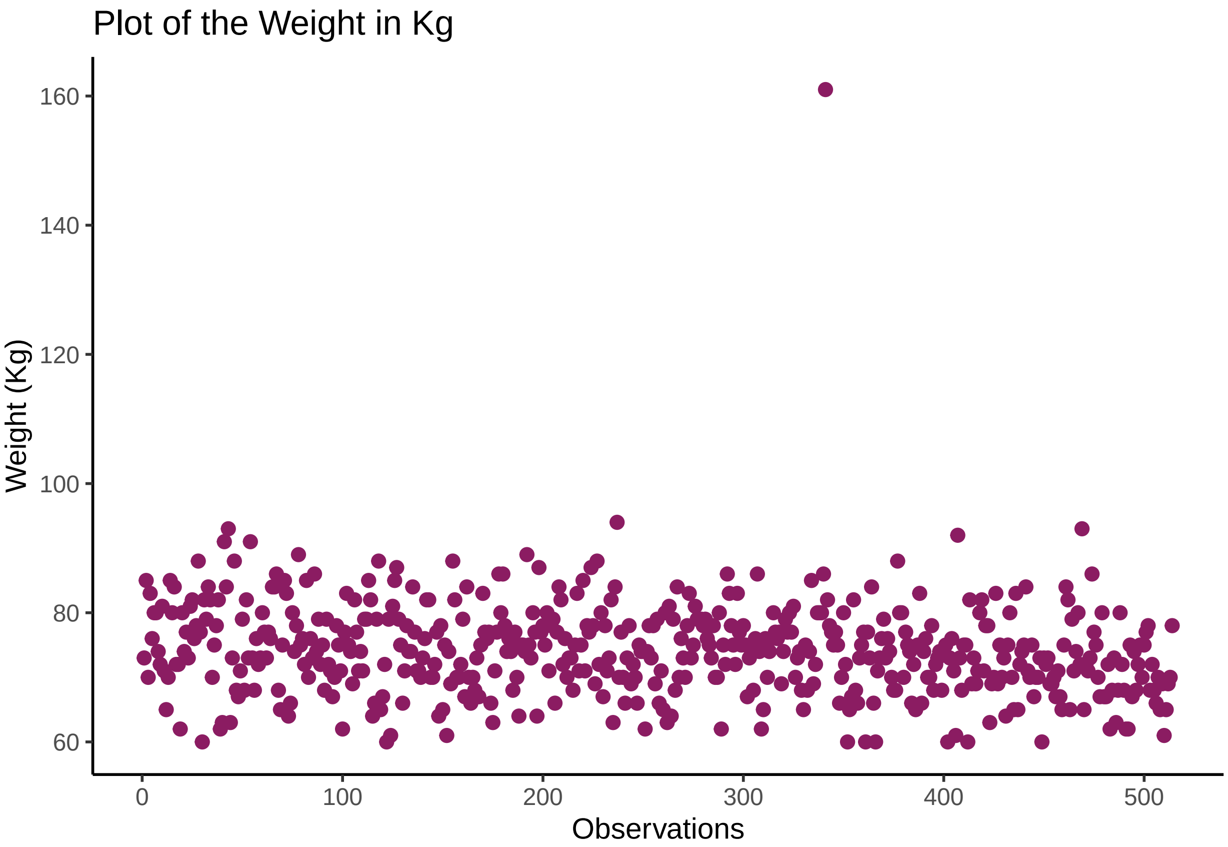
```r
# Plot and visualise the `age` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = age)) +
  geom_point(size=2, color = "coral4") +
  ggtitle("Plot of the Age") +
  xlab("observations") + ylab("Age") +
  theme_classic()
```



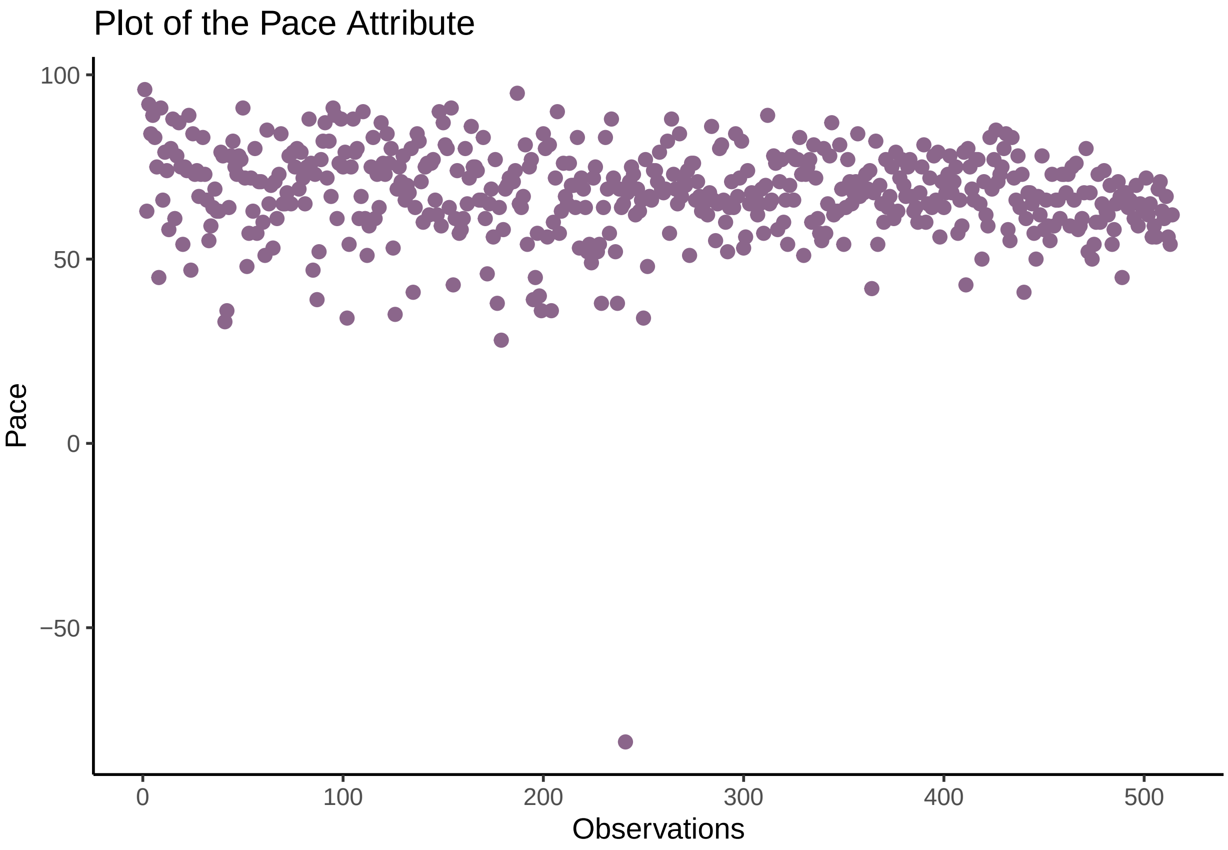
```r
# Plot and visualise the `height_cm` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = height_cm)) +
  geom_point(size=2, color = "firebrick3") +
  ggtitle("Plot of the Height in cm") +
  xlab("Observations") + ylab("Height (cm)") +
  theme_classic()
```



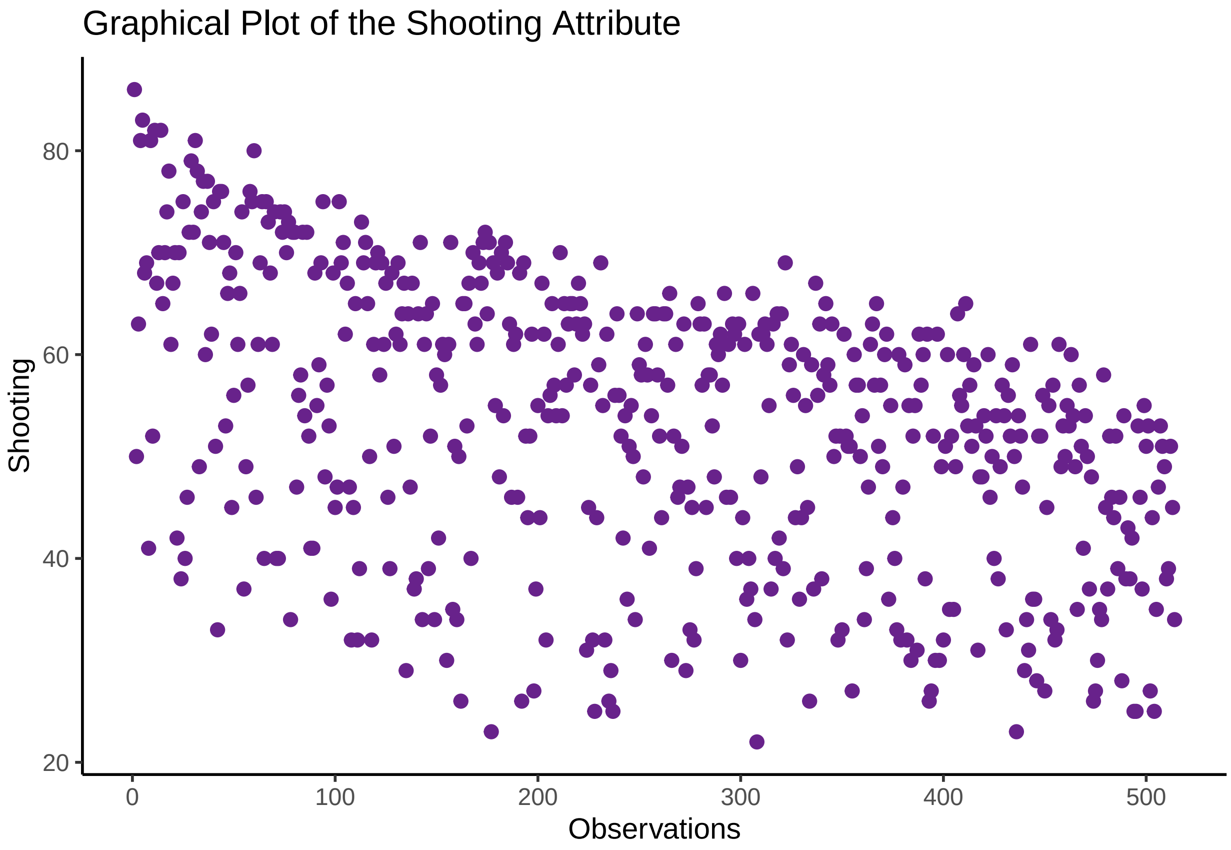
```
# Plot and visualise the `weight_cm` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = weight_kg)) +
  geom_point(size=2, color = "maroon4") +
  ggtitle("Plot of the Weight in Kg") +
  xlab("Observations") + ylab("Weight (Kg)") +
  theme_classic()
```



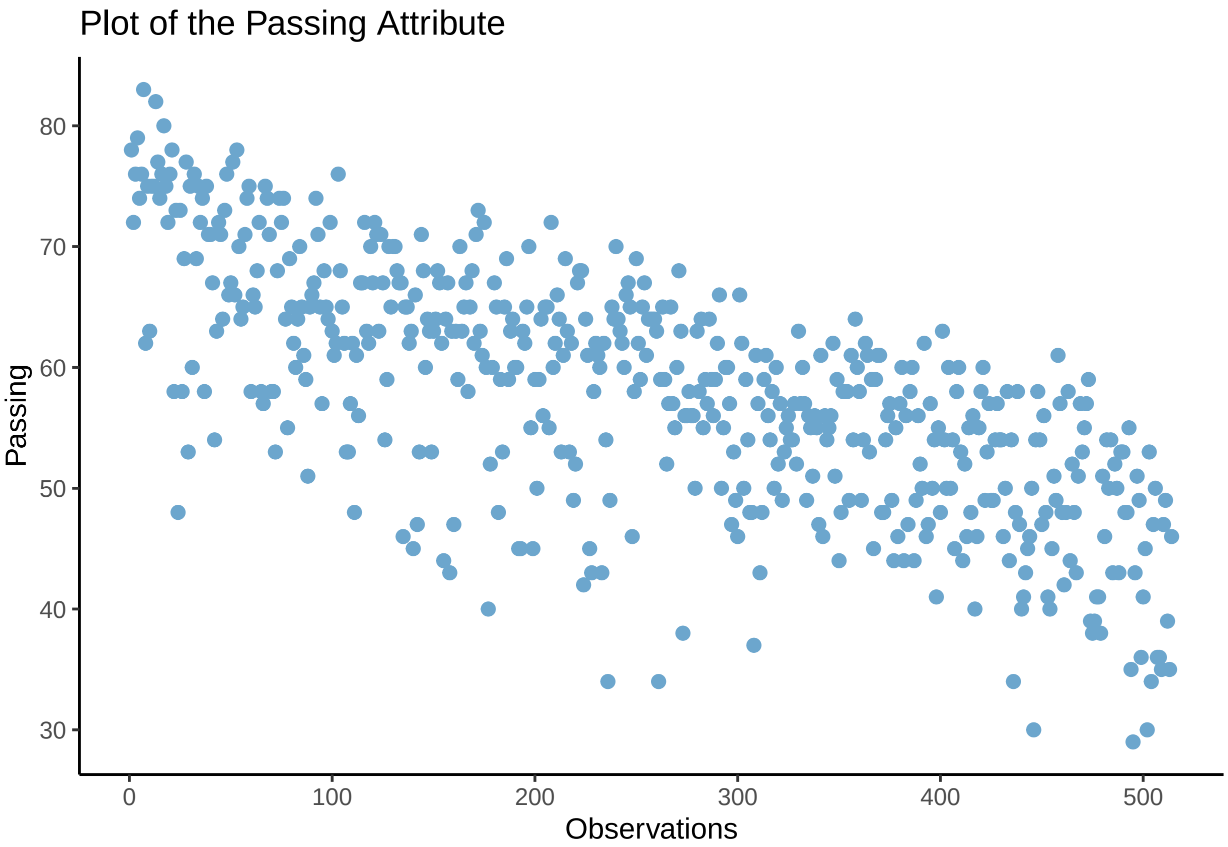
```
# Plot and visualise the `pace` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = pace)) +
  geom_point(size=2, color = "plum4") +
  ggtitle("Plot of the Pace Attribute") +
  xlab("Observations") + ylab("Pace") +
  theme_classic()
```



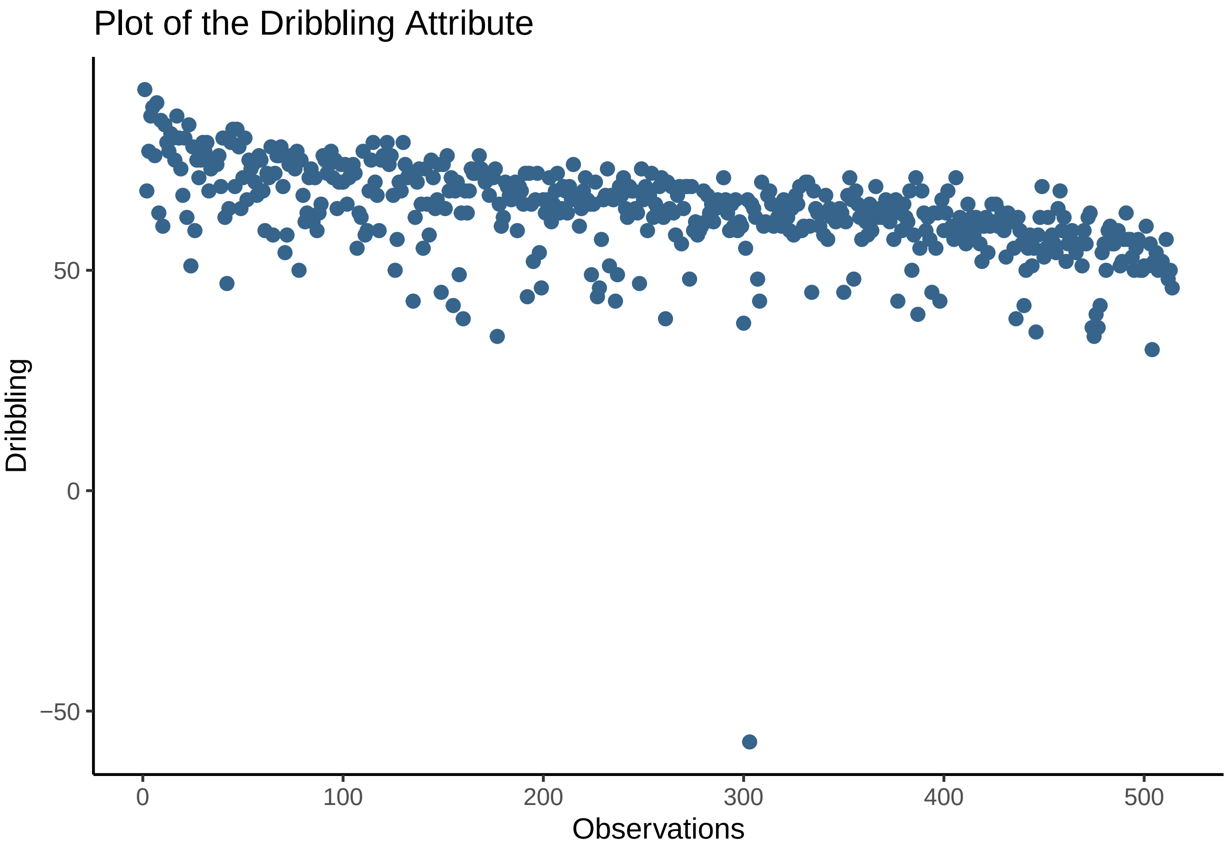
```
# Plot and visualise the `shooting` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = shooting)) +
  geom_point(size=2, color = "darkorchid4") +
  ggtitle("Graphical Plot of the Shooting Attribute") +
  xlab("Observations") + ylab("Shooting") +
  theme_classic()
```



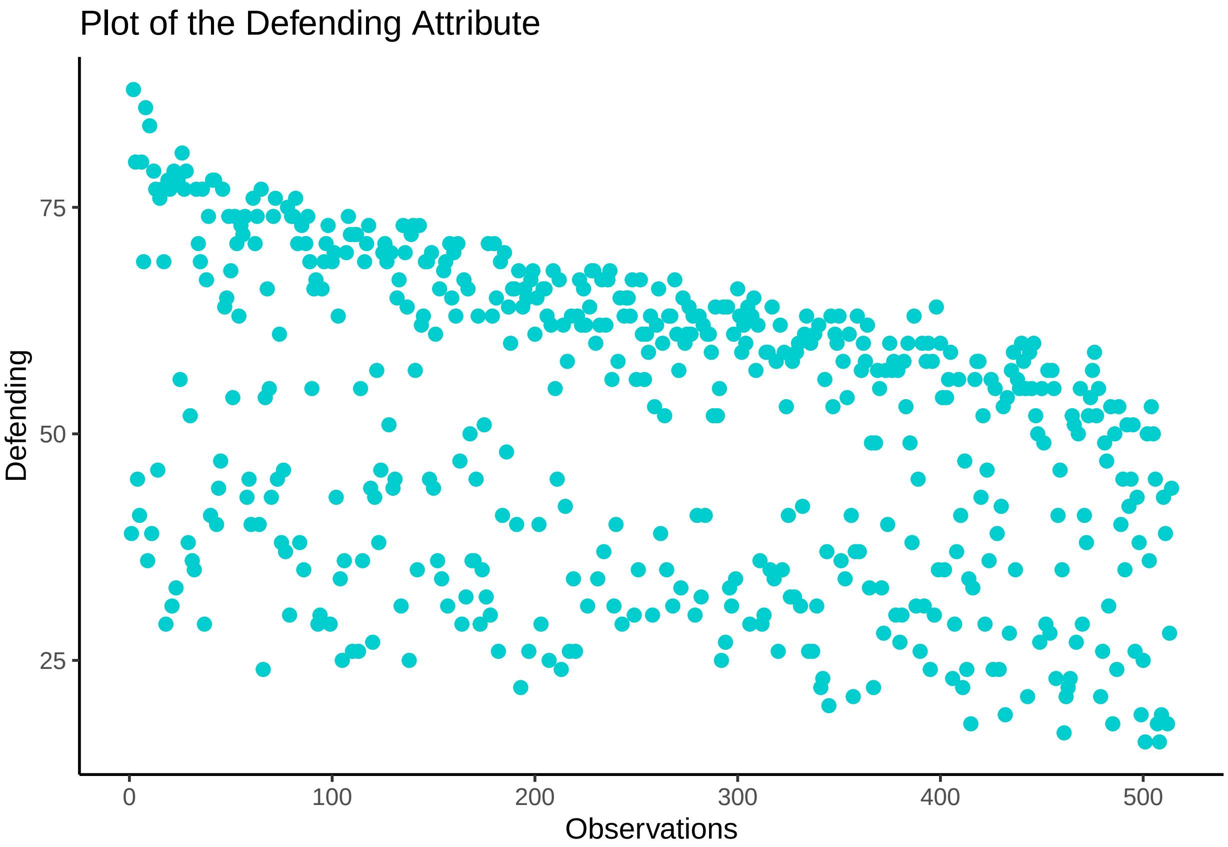
```
# Plot and visualise the `passing` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = passing)) +
  geom_point(size=2, color = "skyblue3") +
  ggtitle("Plot of the Passing Attribute") +
  xlab("Observations") + ylab("Passing") +
  theme_classic()
```



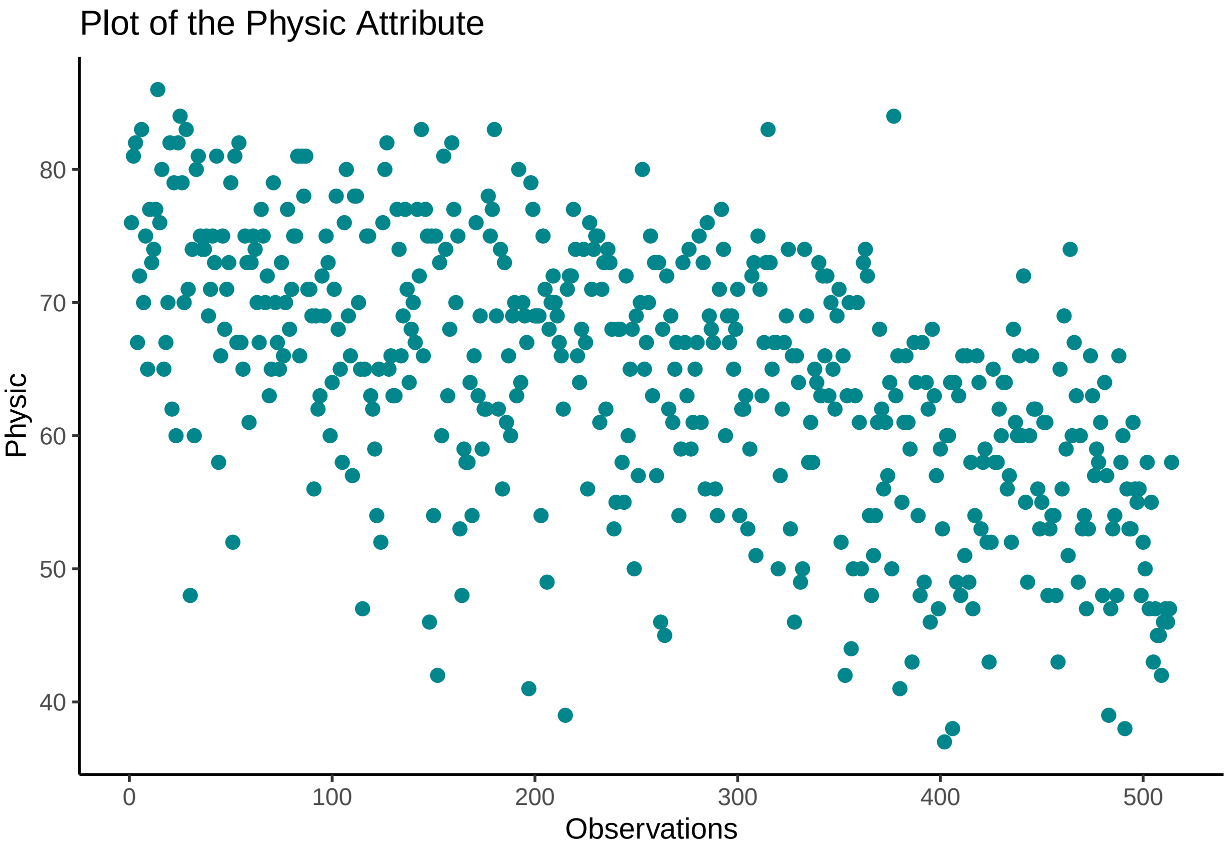
```
# Plot and visualise the `dribbling` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = dribbling)) +
  geom_point(size=2, color = "steelblue4") +
  ggtitle("Plot of the Dribbling Attribute") +
  xlab("Observations") + ylab("Dribbling") +
  theme_classic()
```



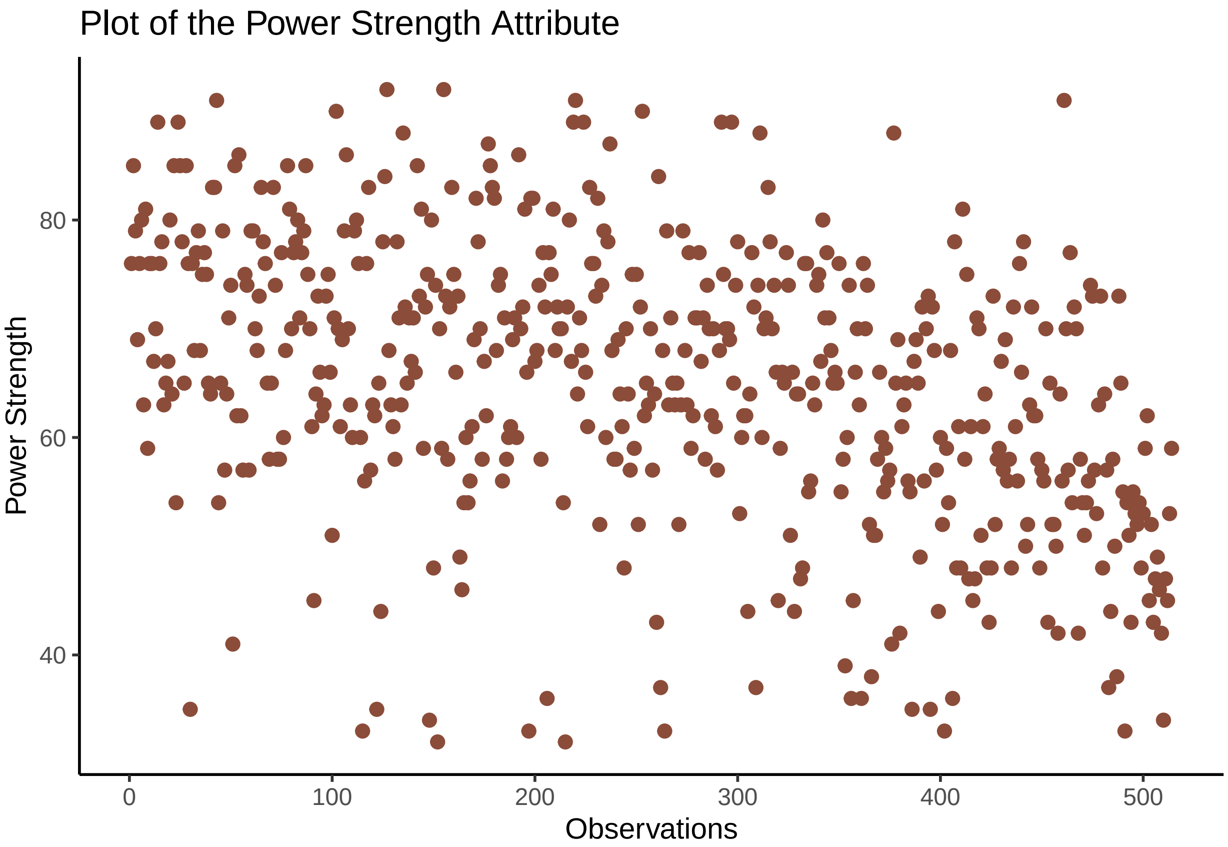
```
# Plot and visualise the `defending` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = defending)) +
  geom_point(size=2, color = "cyan3") +
  ggtitle("Plot of the Defending Attribute") +
  xlab("Observations") + ylab("Defending") +
  theme_classic()
```



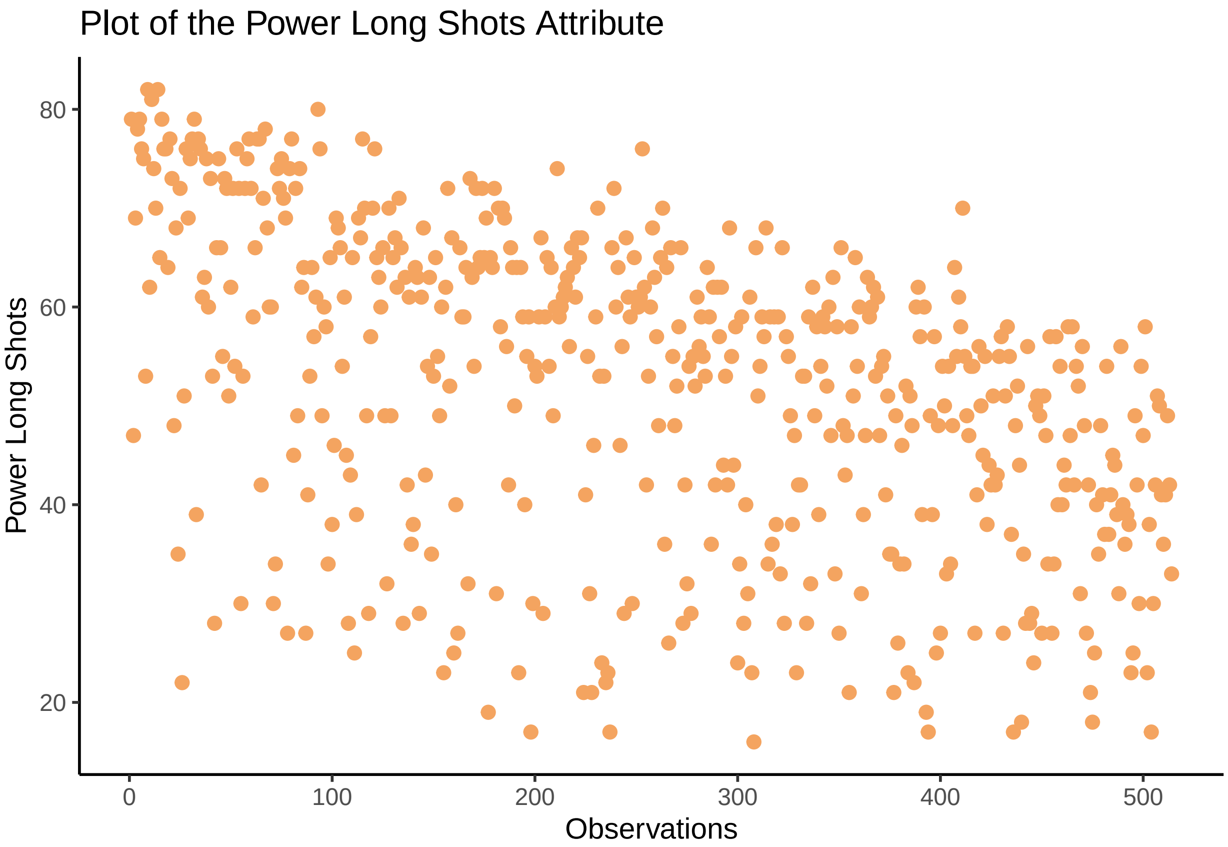
```
# Plot and visualise the `physics` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = physic)) +
  geom_point(size=2, color = "turquoise4") +
  ggtitle("Plot of the Physic Attribute") +
  xlab("Observations") + ylab("Physic") +
  theme_classic()
```



```
# Plot and visualise the `power_strength` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = power_strength)) +
  geom_point(size=2, color = "salmon4") +
  ggtitle("Plot of the Power Strength Attribute") +
  xlab("Observations") + ylab("Power Strength") +
  theme_classic()
```



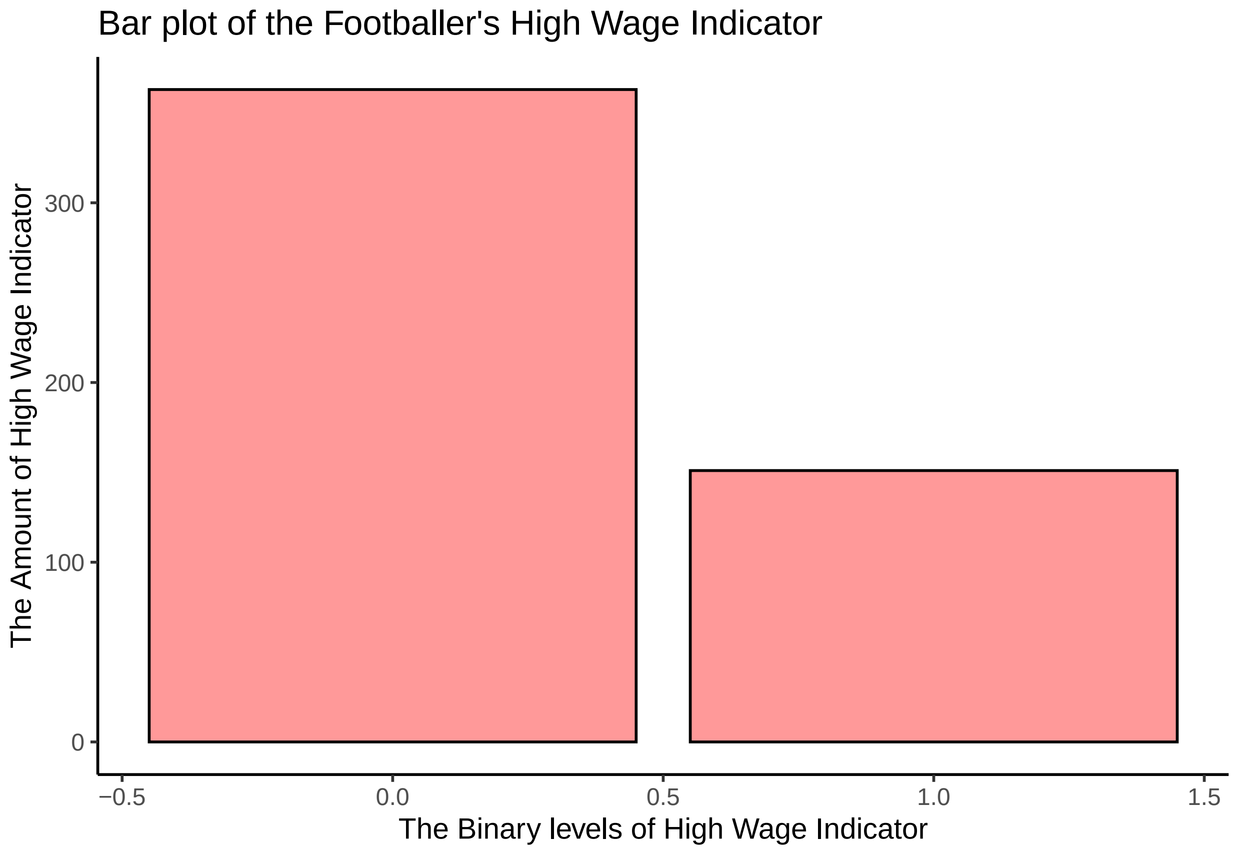
```
# Plot and visualise the `poer_long_shots` variable
ggplot(football_num, aes(x = 1:nrow(football_num), y = power_long_shots)) +
  geom_point(size=2, color = "sandybrown") +
  ggtitle("Plot of the Power Long Shots Attribute") +
  xlab("Observations") + ylab("Power Long Shots") +
  theme_classic()
```



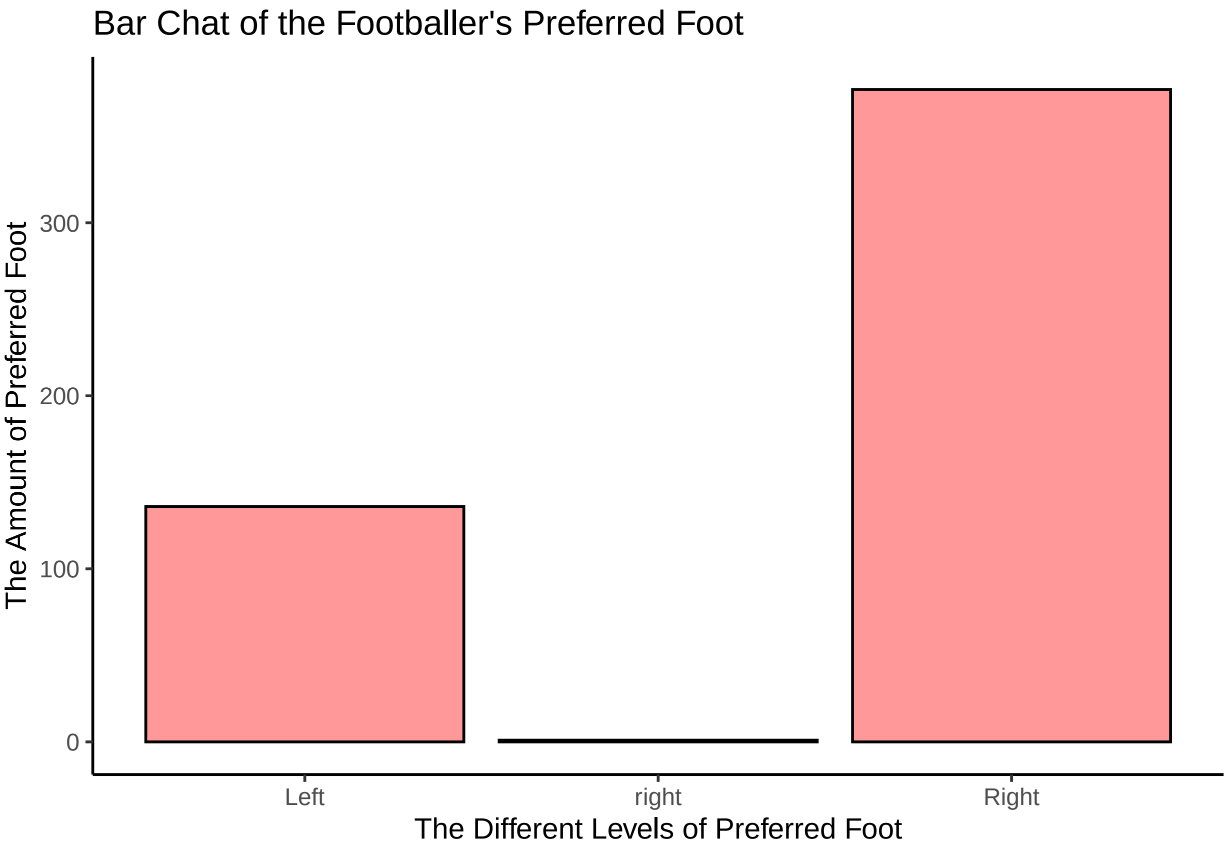
Plotting these variables confirms the presence of outliers in **'age'**, **'wage__eur'**, **pace, dribbling**, **height__cm**, and **weight__cm**. The other continuous variables do not seem to show outliers or other unusual behaviour.

Bar plots using **ggplot2** will help visualise the different levels of the categorical variables

```r
# Plotting the high wage indicator
ggplot(football)+ geom_bar(aes(x=high.wage.ind), fill="#FF9999", colour="black") +
  labs(title = "Bar plot of the Footballer's High Wage Indicator") +
  xlab("The Binary levels of High Wage Indicator") +
  scale_y_continuous(name="The Amount of High Wage Indicator") +
  theme_classic()
```



```r
# Plotting the preferred foot
ggplot(football)+ geom_bar(aes(x=preferred_foot), fill="#FF9999", colour="black") +
  labs(title = "Bar Chat of the Footballer's Preferred Foot") +
  xlab("The Different Levels of Preferred Foot") +
  scale_y_continuous(name="The Amount of Preferred Foot") +
  theme_classic()
```



These bar plots confirm previous observations that **'preferred_foot'** has three levels, where it should have two levels, *"Right"* and *"Left"*

1.3 Data cleaning

The data quality check discovered issues in the dataset, which should be addressed and corrected, where appropriate, via a cleaning process. This will optimise organisation of the data and help with further data wrangling and downstream data analysis. This will also prevent misleading results, and consequently will provide reproducible data and research 10, 11, 12

As there will be changes and modifications to the dataset, the dataset will be saved into a new object, to keep the original dataset.

```
# As there is a likely of making changes and modifying the data frame, it is best save football into an
football_df <- football
```

Removing the negative values found in **'pace'** and **'dribbling'** would be the appropriate way to address and resolve this issue, as footballers' attributes cannot contain negative values.

```
# Finding out how many rows of 'pace' variable contains negative values
football_df[which(football_df$pace < 0), c("pace")]
```

```
## [1] -81
```

```
# Fixing this negative value and making it positive:
football_df$pace[football_df$pace == -81] <- 81
#Lets double check if this issue was fixed
subset(football_df, pace < 0)
```

```
##  [1] sofifa_id        potential        wage_eur         age
##  [5] height_cm        weight_kg        club_name        preferred_foot
##  [9] pace             shooting         passing          dribbling
## [13] defending        physic           power_strength   power_long_shots
## [17] high.wage.ind
## <0 rows> (or 0-length row.names)
```

```
# Will do the same process for the dribbling variable
football_df[which(football_df$dribbling < 0), c("dribbling")]
```

```
## [1] -57
```

```
football_df$dribbling[football_df$dribbling == -57] <- 57
subset(football_df, dribbling < 0)
```

```
##  [1] sofifa_id        potential        wage_eur         age
##  [5] height_cm        weight_kg        club_name        preferred_foot
##  [9] pace             shooting         passing          dribbling
## [13] defending        physic           power_strength   power_long_shots
## [17] high.wage.ind
## <0 rows> (or 0-length row.names)
```

The next issue to be resolved is **'preferred_foot'**, as there was one value that was *'right'* instead of being *'Right'*. Although they have the same meaning, R would interpret these as two different categories, which can cause problems later on when doing further data analysis for the research questions.

```
football_df$preferred_foot[football_df$preferred_foot == "right"] <- "Right"
# To double check that this issue was fixed:
table(football_df$preferred_foot)
```

```
##
##  Left Right
##   136   378
```

This variable should also be converted to factor to improve further data analysis.

```
football_df$preferred_foot <- as.factor(football_df$preferred_foot)
# To double check
is.factor(football_df$preferred_foot)
```

```
## [1] TRUE
```

In regards to the outliers identified in **'age'**, **'wage_eur'**, **pace**, **dribbling**, **height_cm**, and **weight_cm**, only the error found in **'wage_eur'** can be addressed correctly by removing the decimal point, to change the amount from 6 to 60000 Euro. This change would be justified as the average minimum weekly wage for European football players is approximately 25,000 Euro and therefore this is clearly a human error in the data input 13.

```r
# Changing and fixing the decimal number
football_df$wage_eur[football_df$wage_eur == 6.0001] <- 60000
# Checking at the new change
table(football_df$wage_eur)
```

```
##
##    500    550    600    650    700    750    800    850    900    950   1000
##     62      7      6      1      5      4      4     12      7      3     40
##   2000   3000   4000   5000   6000   7000   8000   9000  10000  11000  12000
##     68     35     25     36     19     17     11     13      9      8     13
##  13000  14000  15000  16000  17000  18000  19000  20000  21000  22000  23000
##      5      5      3      8      4      4      4      4      3      2      2
##  24000  26000  27000  28000  29000  30000  31000  32000  34000  35000  36000
##      3      3      4      4      4      3      1      1      1      3      2
##  38000  41000  42000  45000  46000  47000  48000  49000  50000  51000  55000
##      1      3      1      1      1      1      1      2      2      3      1
##  58000  59000  60000  64000  66000  68000  74000  95000  1e+05 105000 120000
##      1      2      2      1      2      1      2      1      1      1      1
## 155000 160000 170000  2e+05
##      1      1      1      1
```

The reason for being inappropriate to make modifications to the remaining outliers is due to the limitation to access of information around the dataset. Trying to change the outliers with little information can lead to false and misleading results, and consequently incorrect and false reproducible data. It would be frowned upon to try and fix data to create a different dataset.

Furthermore, these outliers should be dealt with carefully as they may hold important information or might be part of an interesting case. For example, as shown below in the output, although there is a football player who is 70 years old, this player’s other information seems to be within the acceptable range, and therefore other circumstances need to be considered 10 , 14.

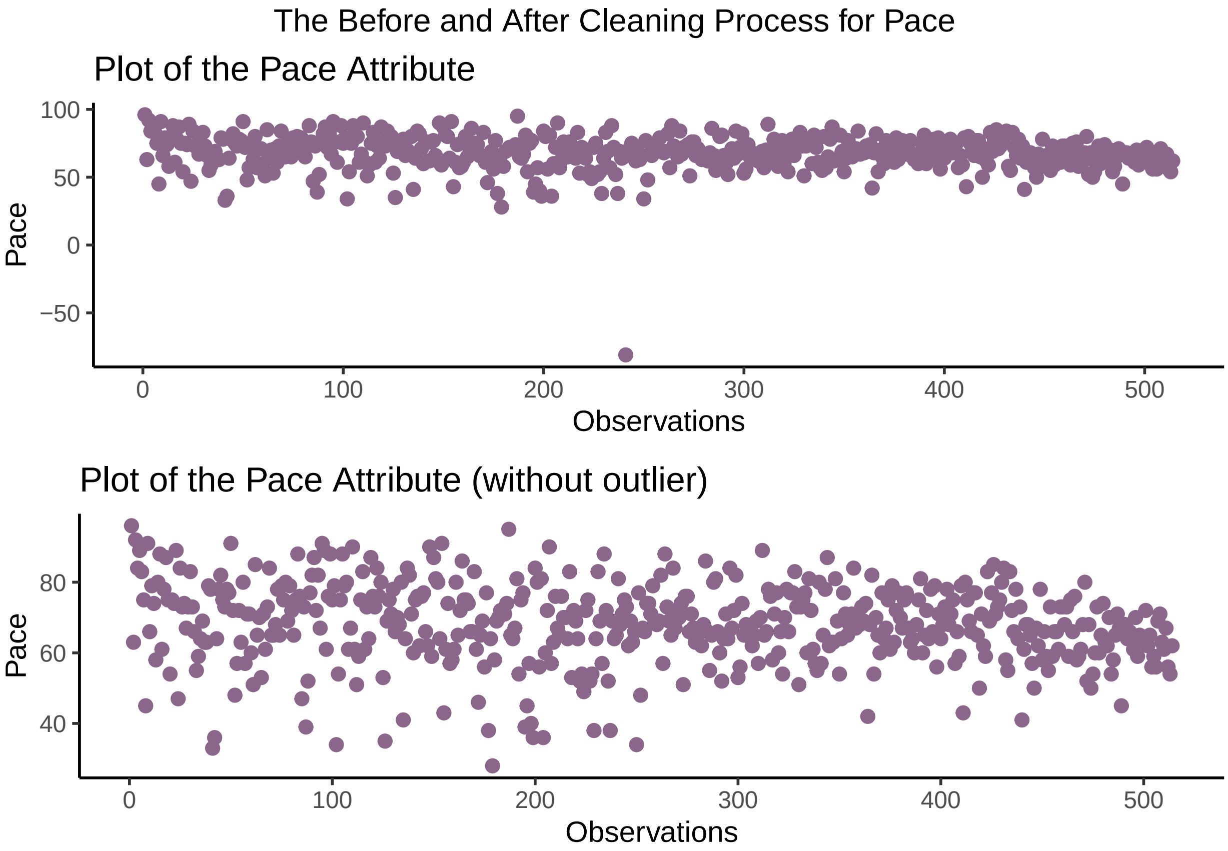
```r
subset(football_df, age == 70)
```

```
##     sofifa_id potential wage_eur age height_cm weight_kg
## 173    201891        69     5000  70       177        77
##                     club_name preferred_foot pace shooting passing dribbling
## 173 TSV Egger Glas Hartberg          Right   65       71      63        67
##     defending physic power_strength power_long_shots high.wage.ind
## 173        29     69             70               65             0
```

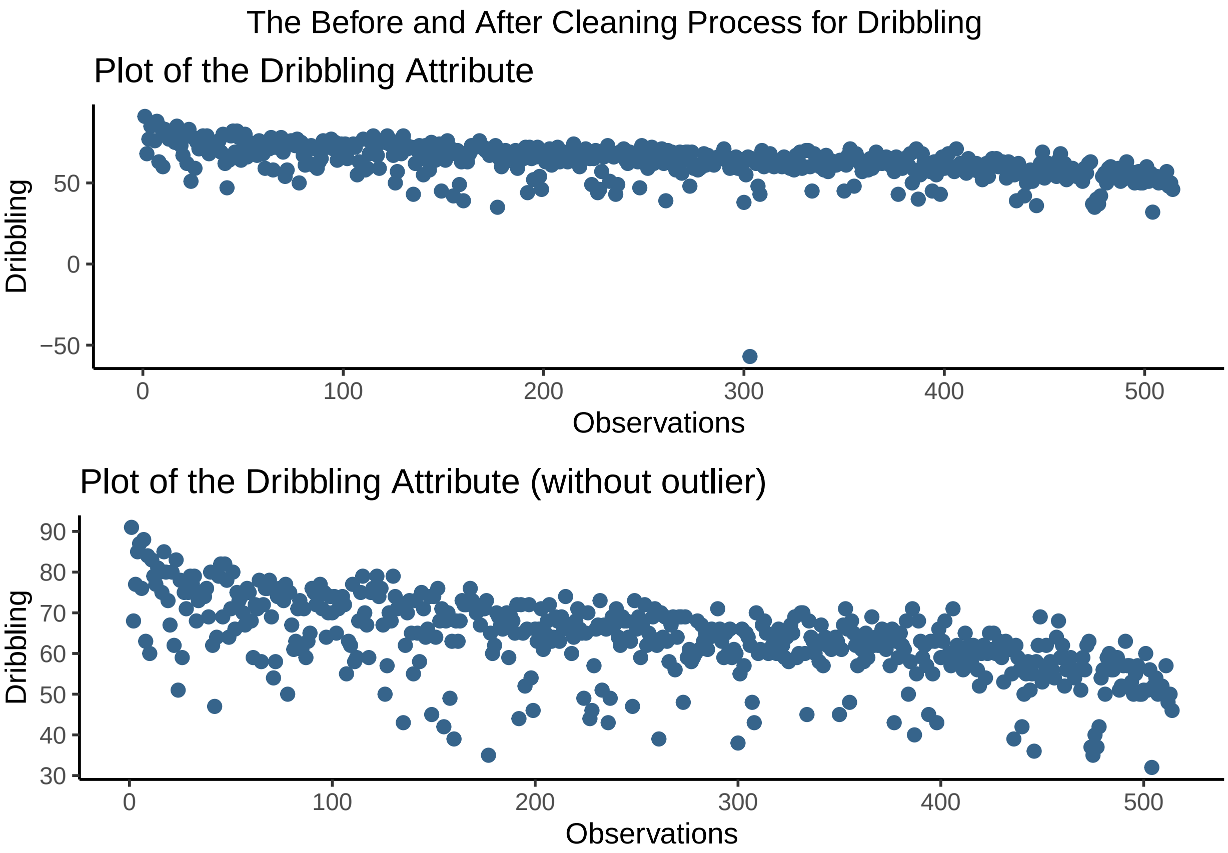
ggplot2 and **gridExtra** library package will be used to visualise the before and after cleaning process, made in the **‘pace’**, **‘dribbling’**, **‘wage_eur’**, and **‘preferred_foot’** [15], 16.

```r
# A) before and after for pace
# Before the cleaning process was performed for the pace
pacePlot_before <- ggplot(football_num, aes(x = 1:nrow(football_num), y = pace)) +
  geom_point(size=2, color = "plum4") +
  ggtitle("Plot of the Pace Attribute") +
  xlab("Observations") + ylab("Pace") +
  theme_classic()

# After the cleaning process was performed for the pace
```

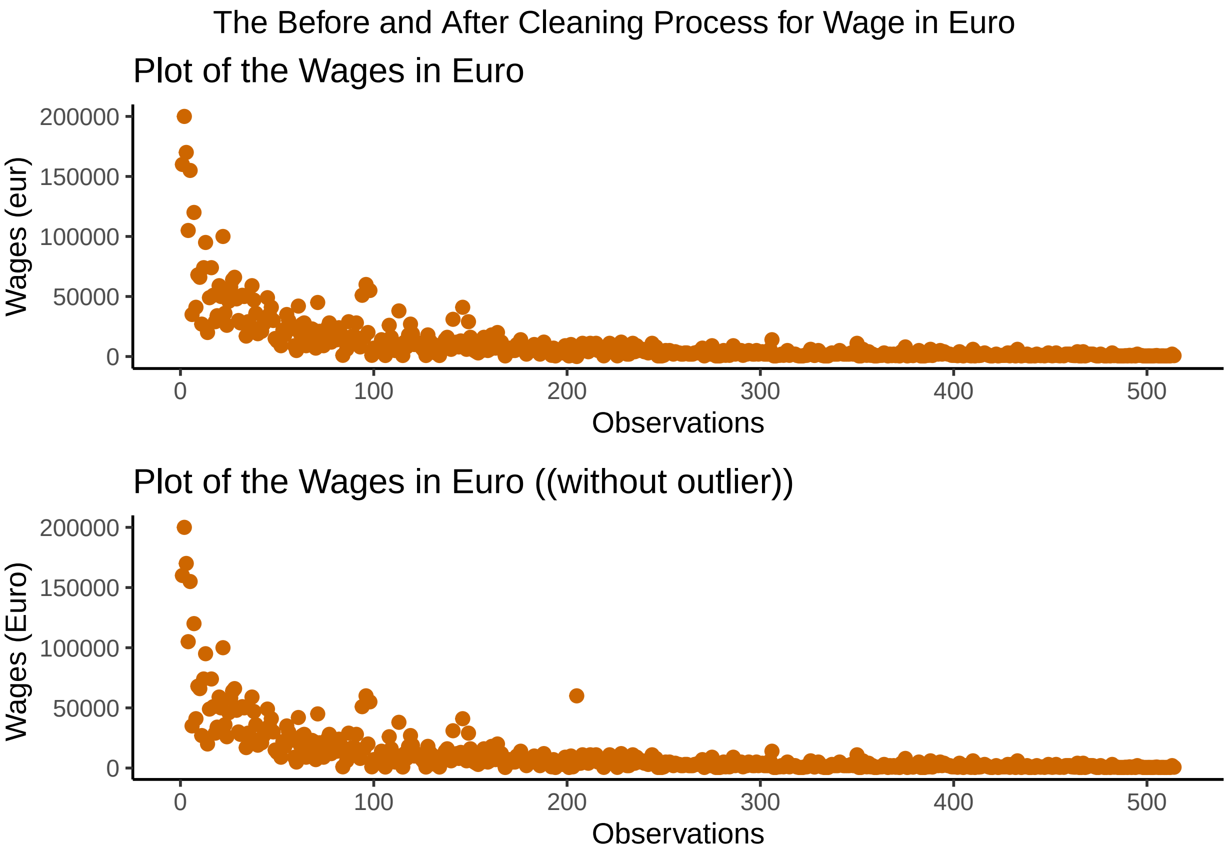
```
pacePlot_after <- ggplot(football_df, aes(x = 1:nrow(football_df), y = pace)) +
  geom_point(size=2, color = "plum4") +
  ggtitle("Plot of the Pace Attribute (without outlier)") +
  xlab("Observations") + ylab("Pace") +
  theme_classic()

# Let visualise the before and after affect in 1 grid
grid.arrange(pacePlot_before, pacePlot_after, top = "The Before and After Cleaning Process for Pace")
```



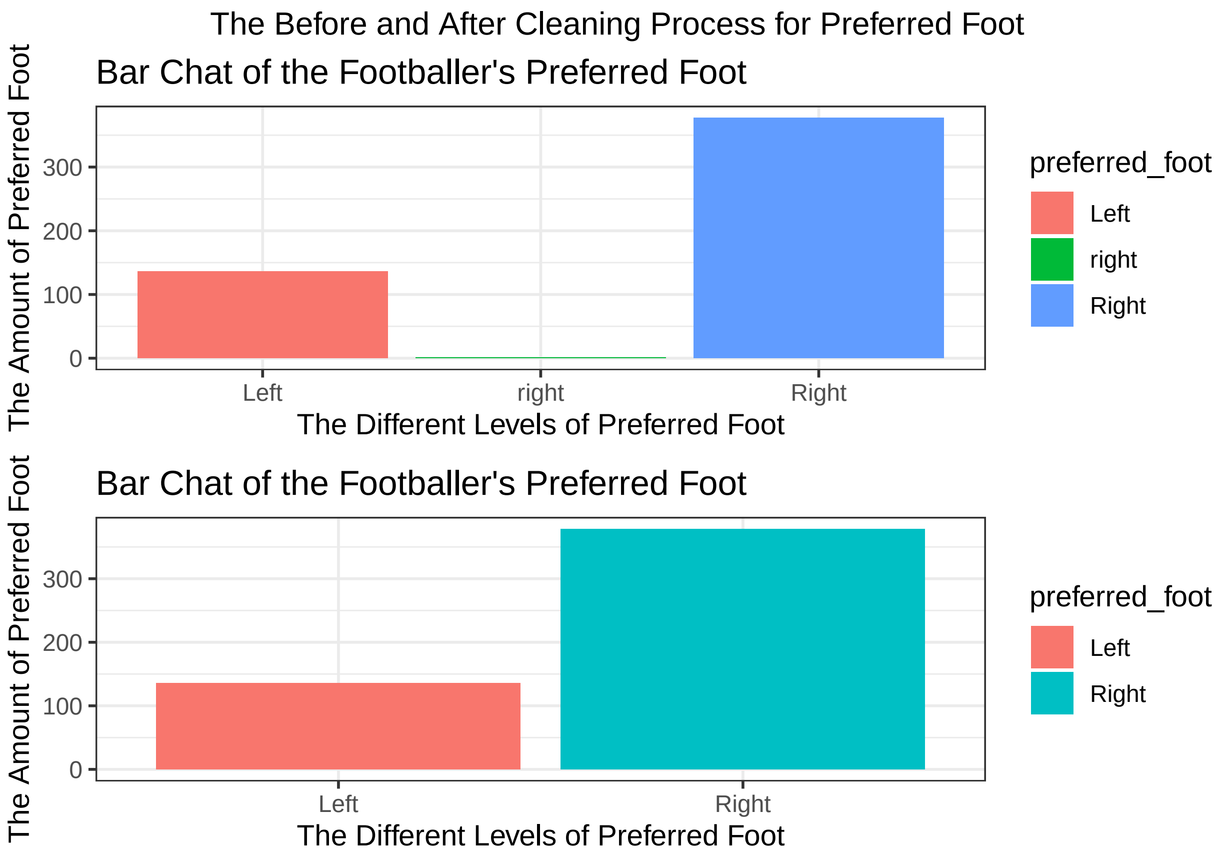
```
# B) before and after for dribbling
# Before the cleaning process was performed for the dribbling
dribblingPlot_before <- ggplot(football_num, aes(x = 1:nrow(football_num), y = dribbling)) +
  geom_point(size=2, color = "steelblue4") +
  ggtitle("Plot of the Dribbling Attribute") +
  xlab("Observations") + ylab("Dribbling") +
  theme_classic()

# After the cleaning process was performed for the dribbling
dribblingPlot_after <- ggplot(football_df, aes(x = 1:nrow(football_df), y = dribbling)) +
  geom_point(size=2, color = "steelblue4") +
  ggtitle("Plot of the Dribbling Attribute (without outlier)") +
  xlab("Observations") + ylab("Dribbling") +
  theme_classic()

# Let visualise the before and after affect in 1 grid
grid.arrange(dribblingPlot_before, dribblingPlot_after, top = "The Before and After Cleaning Process fo
```



```
# C) before and after for wage_eur
# Before the cleaning process was performed for the wage_eur
wage_before <- ggplot(football_num, aes(x = 1:nrow(football_num), y = wage_eur)) +
  geom_point(size=2, color = "darkorange3") +
  ggtitle("Plot of the Wages in Euro") +
  xlab("Observations") + ylab("Wages (eur)") +
  theme_classic()

# After the cleaning process was performed for the wage_eur
wage_after <- ggplot(football_df, aes(x = 1:nrow(football_df), y = wage_eur)) +
  geom_point(size=2, color = "darkorange3") +
  ggtitle("Plot of the Wages in Euro ((without outlier))") +
  xlab("Observations") + ylab("Wages (Euro) ") +
  theme_classic()

# Let visualise the before and after affect in 1 grid
grid.arrange(wage_before, wage_after, top = "The Before and After Cleaning Process for Wage in Euro")
```



```
# C) before and after for preferred_foot
# Before the cleaning process was performed for the preferred_foot
pfoot_before <- ggplot(football, aes(x=preferred_foot, fill= preferred_foot), colour="black")+ geom_bar
  labs(title = "Bar Chat of the Footballer's Preferred Foot") +
  xlab("The Different Levels of Preferred Foot") +
  scale_y_continuous(name="The Amount of Preferred Foot") +
  theme_bw()

# After the cleaning process was performed for the preferred_foot
pfoot_after <- ggplot(football_df, aes(x=preferred_foot, fill=preferred_foot), colour="black")+ geom_ba
  labs(title = "Bar Chat of the Footballer's Preferred Foot") +
  xlab("The Different Levels of Preferred Foot") +
  scale_y_continuous(name="The Amount of Preferred Foot") +
  theme_bw()
# Let visualise the before and after affect in 1 grid
grid.arrange(pfoot_before, pfoot_after, top = "The Before and After Cleaning Process for Preferred Foot
```



As indicated, having performed a small change to correct the error in these variables improved the quality of the variables, and consequently improved the dataset. The cleaning process has also improved the data points for wage, with the change in value made. As it visually shows, the **'preferred_foot'** has now the correct categorical levels, *"Right"* and *"Left"*, which could be beneficial in, for example, assessing the relationship between other variables such as 'shooting' to see if players shoot better with either left or right foot, or with no difference.

A further cleaning process will be performed to improve the readability of the data and data visualisation. To improve the data visualisation, a new column will be created, containing the logistic data type of high.wage.ind, where 0 is FALSE and 1 is TRUE, for players that earn less or equal to, or greater than 8000 Euro weekly, respectively 5, 17.

```
# creating and adding a new column
football_df <- football_df %>%
  mutate( high.wage.ind.log = high.wage.ind)
head(football_df)
```

```
##   sofifa_id potential wage_eur age height_cm weight_kg               club_name
## 1    231747        95   160000  21       178        73     Paris Saint-Germain
## 2    212218        90   200000  26       189        85         Manchester City
## 3    188377        85   170000  30       183        70         Manchester City
## 4    235790        93   105000  21       188        83                 Chelsea
## 5    211300        88   155000  24       184        76       Manchester United
## 6    183512        83    35000  30       181        80 Athletic Club de Bilbao
##   preferred_foot pace shooting passing dribbling defending physic
## 1          Right   96       86      78        91        39     76
## 2           Left   63       50      72        68        88     81
## 3          Right   92       63      76        77        80     82
## 4           Left   84       81      79        85        45     67
## 5          Right   89       83      74        87        41     72
## 6           Left   83       68      76        76        80     83
##   power_strength power_long_shots high.wage.ind high.wage.ind.log
## 1             76               79             1                 1
## 2             85               47             1                 1
## 3             79               69             1                 1
## 4             69               78             1                 1
## 5             76               79             1                 1
## 6             80               76             1                 1
```

```
# Now Convert this binary variable into logical, where 0 is FALSE and 1 is TRUE
football_df$high.wage.ind.log <- as.logical(football_df$high.wage.ind.log)
head(football_df)
```

```
##   sofifa_id potential wage_eur age height_cm weight_kg               club_name
## 1    231747        95   160000  21       178        73     Paris Saint-Germain
## 2    212218        90   200000  26       189        85         Manchester City
## 3    188377        85   170000  30       183        70         Manchester City
## 4    235790        93   105000  21       188        83                 Chelsea
## 5    211300        88   155000  24       184        76       Manchester United
## 6    183512        83    35000  30       181        80 Athletic Club de Bilbao
##   preferred_foot pace shooting passing dribbling defending physic
## 1          Right   96       86      78        91        39     76
## 2           Left   63       50      72        68        88     81
## 3          Right   92       63      76        77        80     82
## 4           Left   84       81      79        85        45     67
## 5          Right   89       83      74        87        41     72
## 6           Left   83       68      76        76        80     83
##   power_strength power_long_shots high.wage.ind high.wage.ind.log
## 1             76               79             1              TRUE
## 2             85               47             1              TRUE
## 3             79               69             1              TRUE
## 4             69               78             1              TRUE
## 5             76               79             1              TRUE
## 6             80               76             1              TRUE
```

The `rename()` function from dplyr will be use to improve and rename the variables 18.

```
# Renaming the variables
football_df <- football_df %>%
  rename(ID = sofifa_id,
         wage = wage_eur,
         height = height_cm,
         weight = weight_kg,
         "club name" = club_name,
         "preferred foot" = preferred_foot,
         "power strength" = power_strength,
         "power long shots" = power_long_shots,
         "high wage indicator" = high.wage.ind)
# To see the new changes
head(football_df)
```

```
##       ID potential   wage age height weight               club name
## 1 231747        95 160000  21    178     73     Paris Saint-Germain
## 2 212218        90 200000  26    189     85         Manchester City
## 3 188377        85 170000  30    183     70         Manchester City
## 4 235790        93 105000  21    188     83                 Chelsea
## 5 211300        88 155000  24    184     76       Manchester United
## 6 183512        83  35000  30    181     80 Athletic Club de Bilbao
##   preferred foot pace shooting passing dribbling defending physic
## 1          Right   96       86      78        91        39     76
## 2           Left   63       50      72        68        88     81
## 3          Right   92       63      76        77        80     82
## 4           Left   84       81      79        85        45     67
## 5          Right   89       83      74        87        41     72
## 6           Left   83       68      76        76        80     83
##   power strength power long shots high wage indicator high.wage.ind.log
## 1             76               79                   1              TRUE
## 2             85               47                   1              TRUE
## 3             79               69                   1              TRUE
## 4             69               78                   1              TRUE
## 5             76               79                   1              TRUE
## 6             80               76                   1              TRUE
```

2. Exploratory Data Analysis (EDA)

2.1 EDA plan

EDA is a type of analysis of data, where a summary of the main characteristics of the dataset is provided, usually with the help of visual aids. Performing EDA is important to help develop a better understanding about the data and it can help answer many questions that can arise when dealing with the data. The EDA in this RMarkdown will involve performing different data visualisations in accordance to the different data classification of the different variables. For instance, plotting a histogram with density plot to visualise the distributions of the continuous variables, would be one approach for uni-variate visualisation. For multi-variate visualisation, plotting boxplots to visualise the relationship between categorical variables and numerical variables would be another approach. Furthermore, some hypothesis testing will be involved to address questions around factors such as the distribution 19, 20, 21.

2.2 EDA and summary of results

2.2.1 Univariate EDA

As shown, the summary with all changes has improved the dataset and provides more useful information.

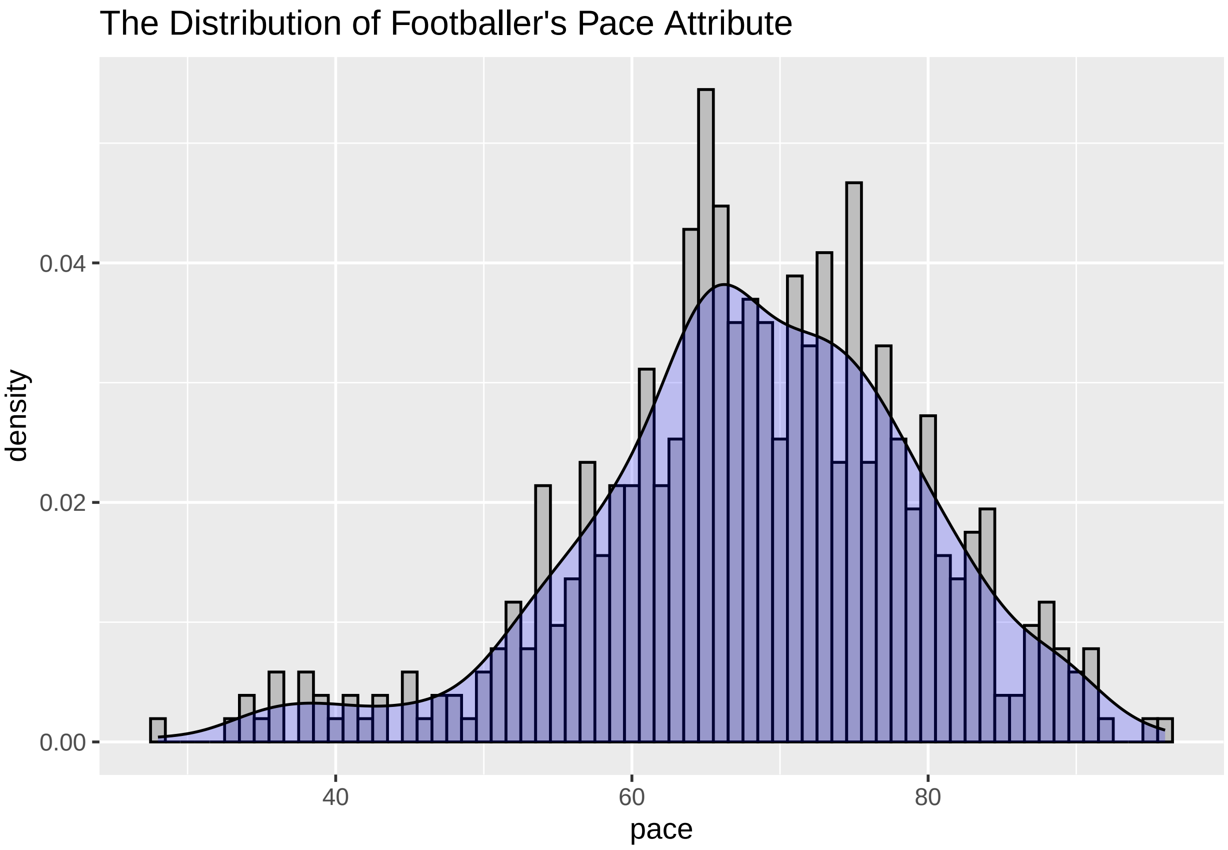
```
summary(football_df)
```

```
##        ID           potential          wage             age
##  Min.   :104476   Min.   :54.00   Min.   :   500   Min.   :16.00
##  1st Qu.:211483   1st Qu.:67.00   1st Qu.:  1000   1st Qu.:21.00
##  Median :232608   Median :71.00   Median :  4000   Median :25.00
##  Mean   :227195   Mean   :71.66   Mean   : 10926   Mean   :25.19
##  3rd Qu.:246961   3rd Qu.:75.00   3rd Qu.: 11000   3rd Qu.:29.00
##  Max.   :258945   Max.   :95.00   Max.   :200000   Max.   :70.00
##      height          weight        club name         preferred foot
##  Min.   :162.0   Min.   : 60.00   Length:514          Left :136
##  1st Qu.:176.0   1st Qu.: 70.00   Class :character   Right:378
##  Median :180.0   Median : 74.00   Mode  :character
##  Mean   :180.1   Mean   : 74.28
##  3rd Qu.:184.0   3rd Qu.: 78.00
##  Max.   :214.0   Max.   :161.00
##       pace          shooting        passing        dribbling
##  Min.   :28.00   Min.   :22.00   Min.   :29.00   Min.   :32.00
##  1st Qu.:62.00   1st Qu.:44.00   1st Qu.:51.00   1st Qu.:59.00
##  Median :68.00   Median :54.50   Median :58.00   Median :64.00
##  Mean   :68.09   Mean   :53.21   Mean   :57.89   Mean   :63.53
##  3rd Qu.:75.75   3rd Qu.:63.00   3rd Qu.:65.00   3rd Qu.:70.00
##  Max.   :96.00   Max.   :86.00   Max.   :83.00   Max.   :91.00
##    defending         physic      power strength  power long shots
##  Min.   :16.00   Min.   :37.00   Min.   :32.00   Min.   :16.00
##  1st Qu.:36.00   1st Qu.:58.00   1st Qu.:57.00   1st Qu.:41.00
##  Median :55.00   Median :65.00   Median :66.00   Median :54.00
##  Mean   :51.12   Mean   :64.36   Mean   :64.96   Mean   :51.94
##  3rd Qu.:64.00   3rd Qu.:72.00   3rd Qu.:74.00   3rd Qu.:64.00
##  Max.   :88.00   Max.   :86.00   Max.   :92.00   Max.   :82.00
##  high wage indicator high.wage.ind.log
##  Min.   :0.0000      Mode :logical
##  1st Qu.:0.0000      FALSE:363
##  Median :0.0000      TRUE :151
##  Mean   :0.2938
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

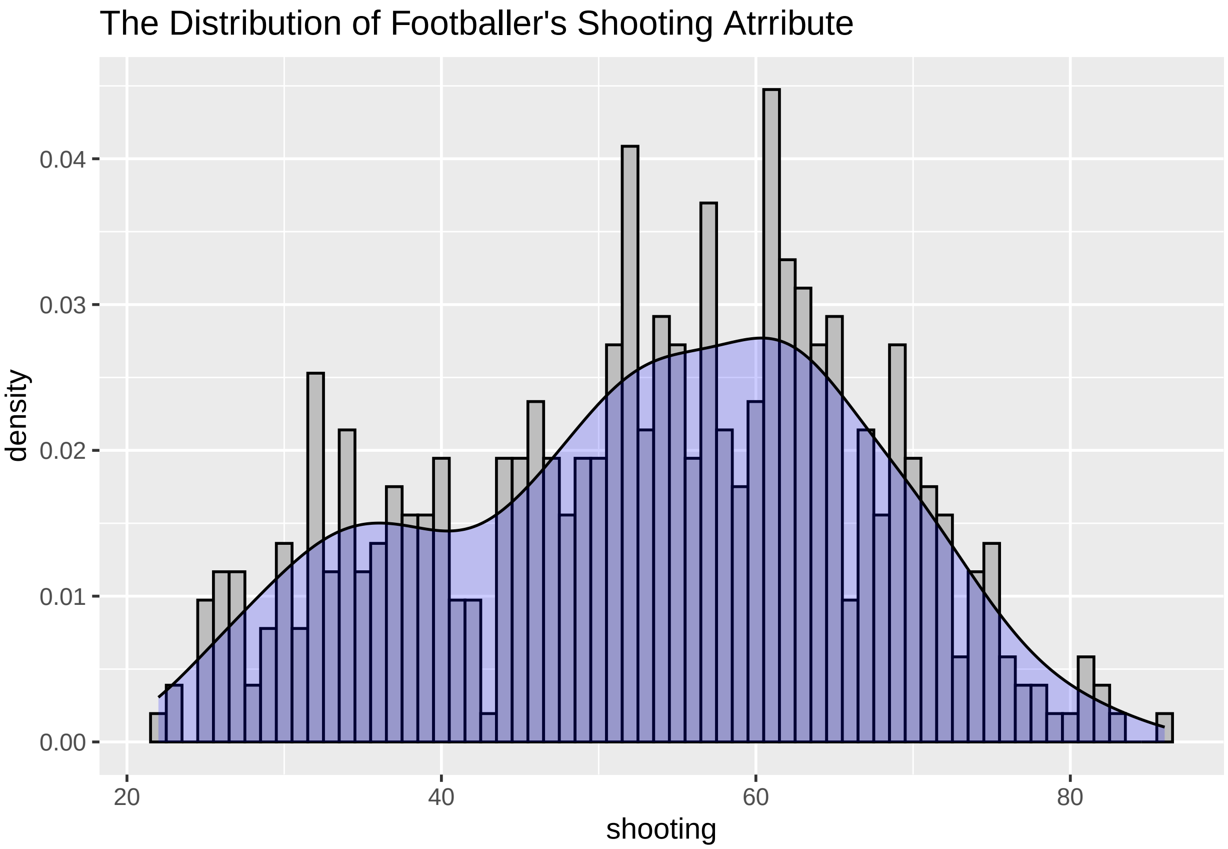
A histogram, with the density plot as an over layer, is an effective way of exploring the distributions and looking for other patterns in the continuous variables 22, 23.

```
# histogram with density overlay for the attributes
ggplot(football_df, aes(x=pace)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Pace Attribute")
```

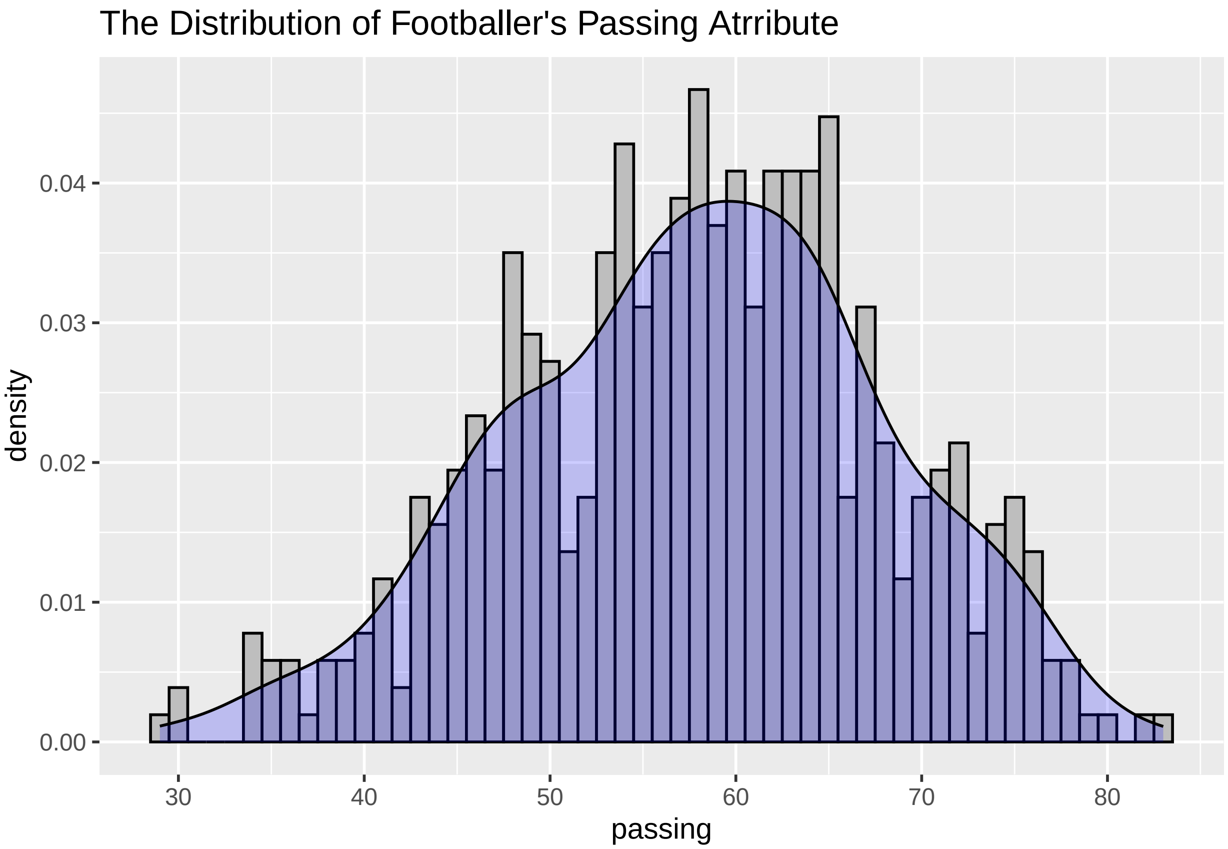
```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



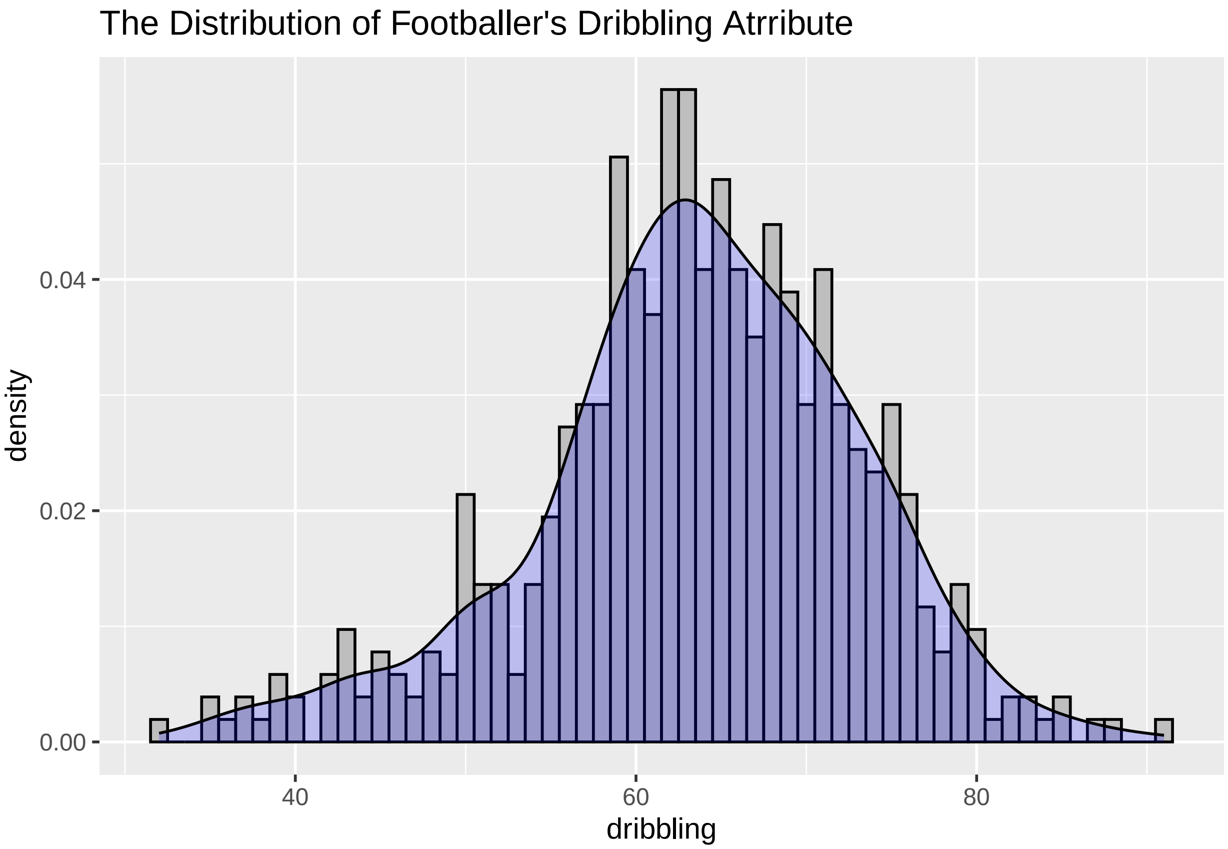
```
ggplot(football_df, aes(x=shooting)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Shooting Atrribute")
```



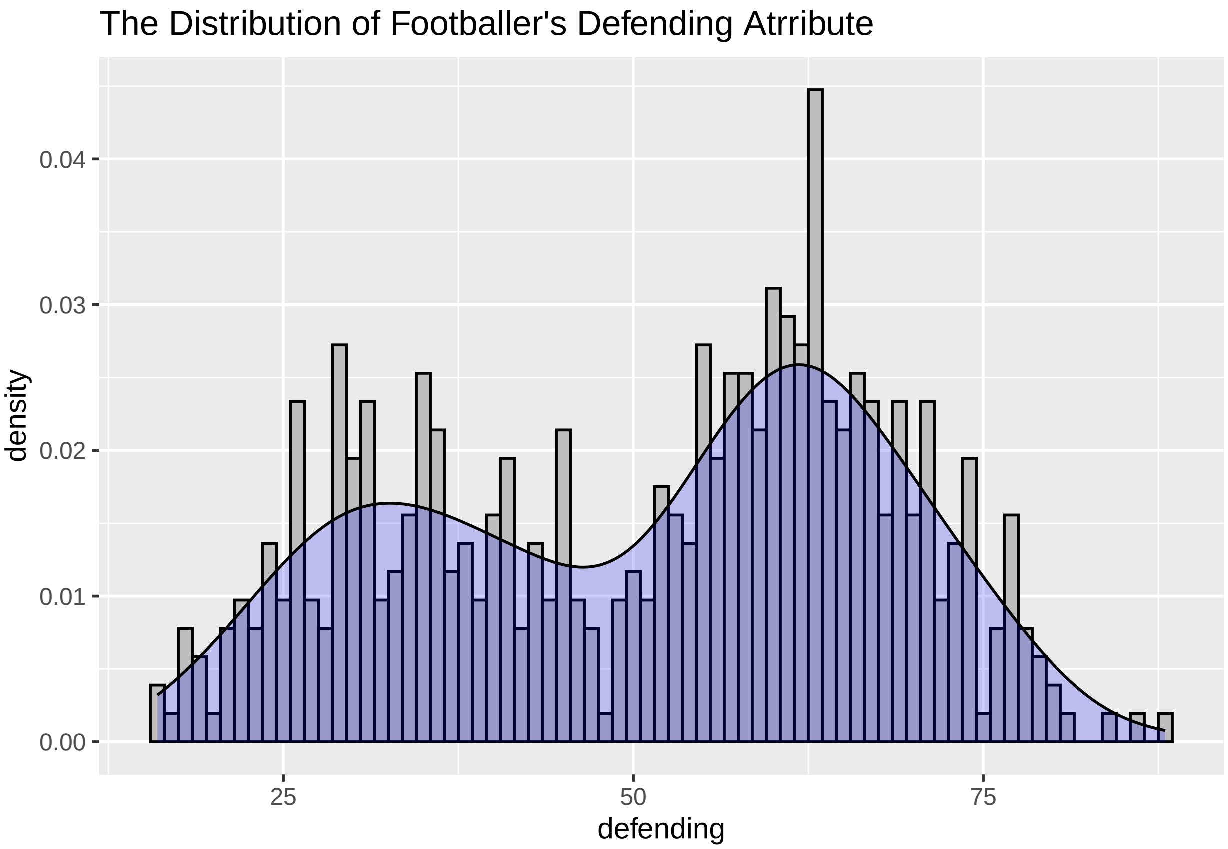
```
ggplot(football_df, aes(x=passing)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Passing Atrribute")
```



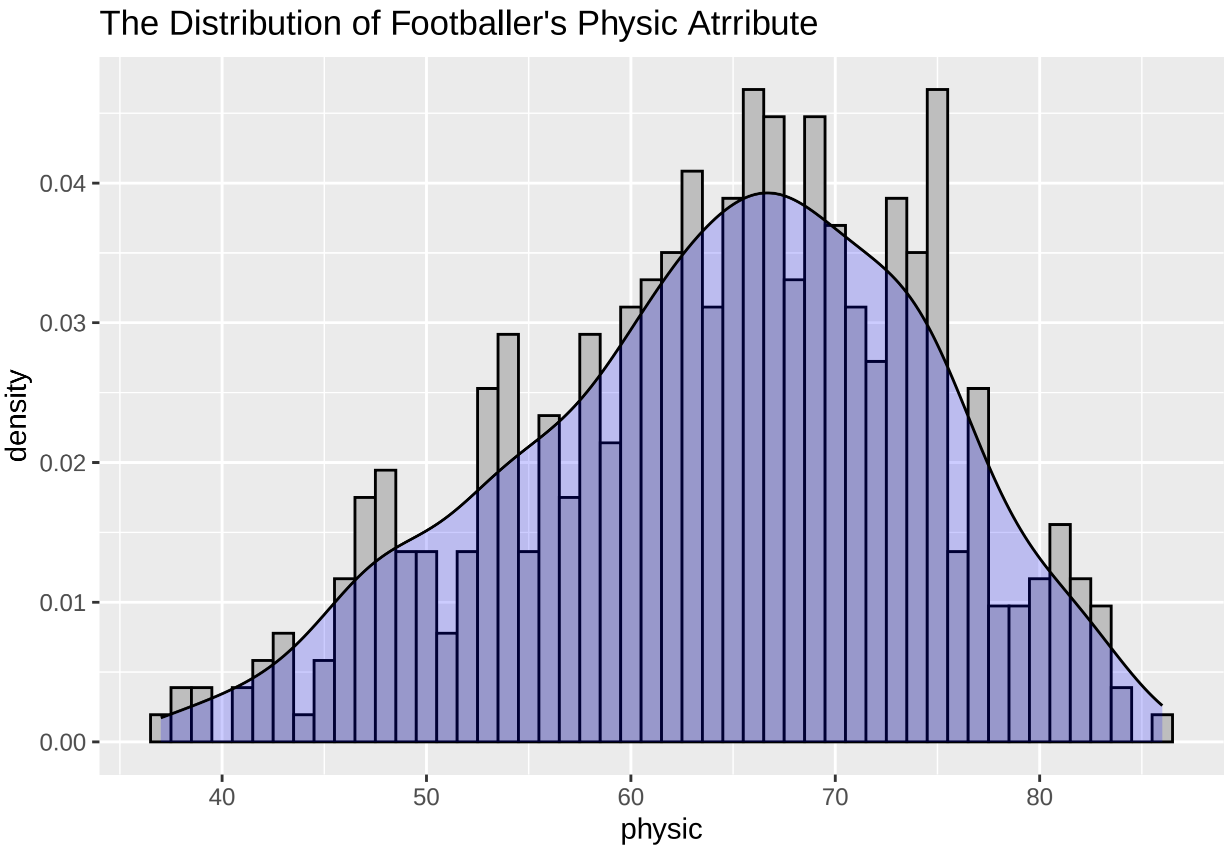
```
ggplot(football_df, aes(x=dribbling)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Dribbling Atrribute")
```



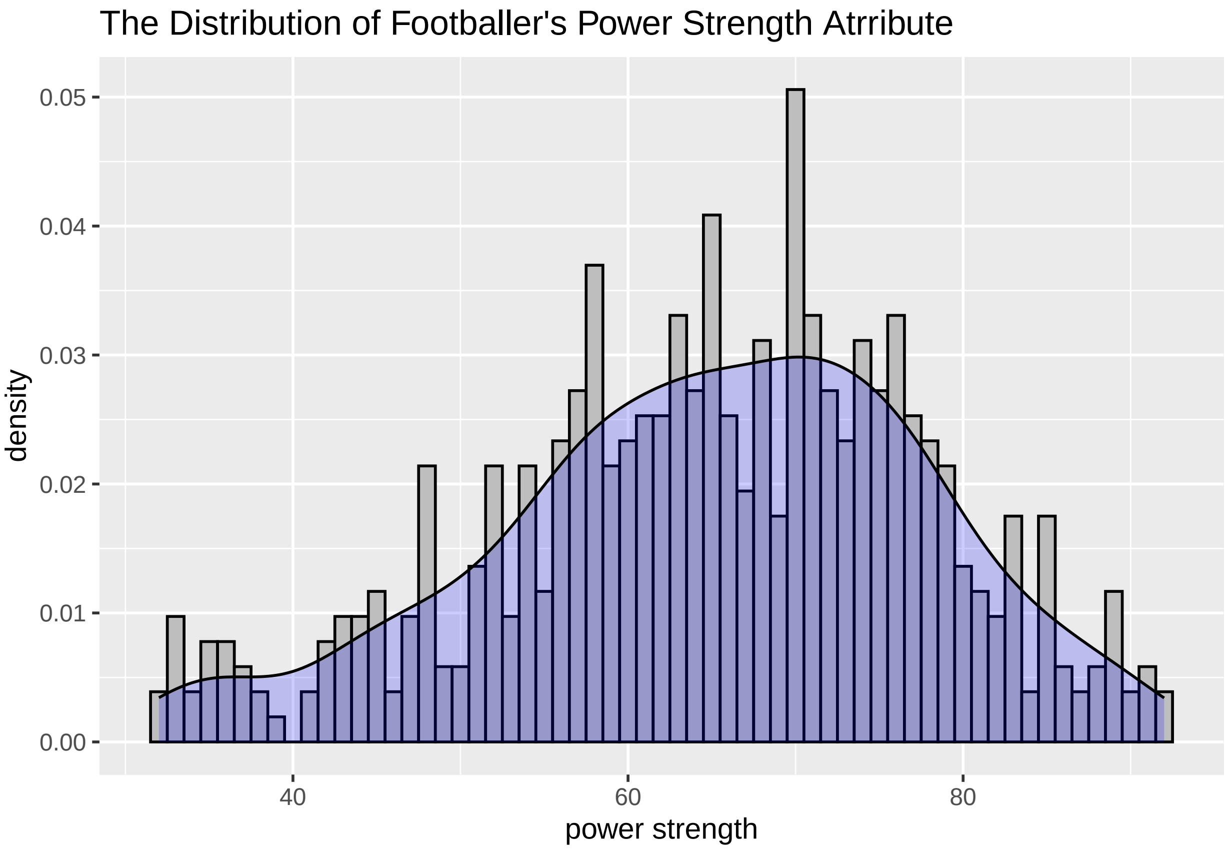
```r
ggplot(football, aes(x=defending)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Defending Atrribute")
```



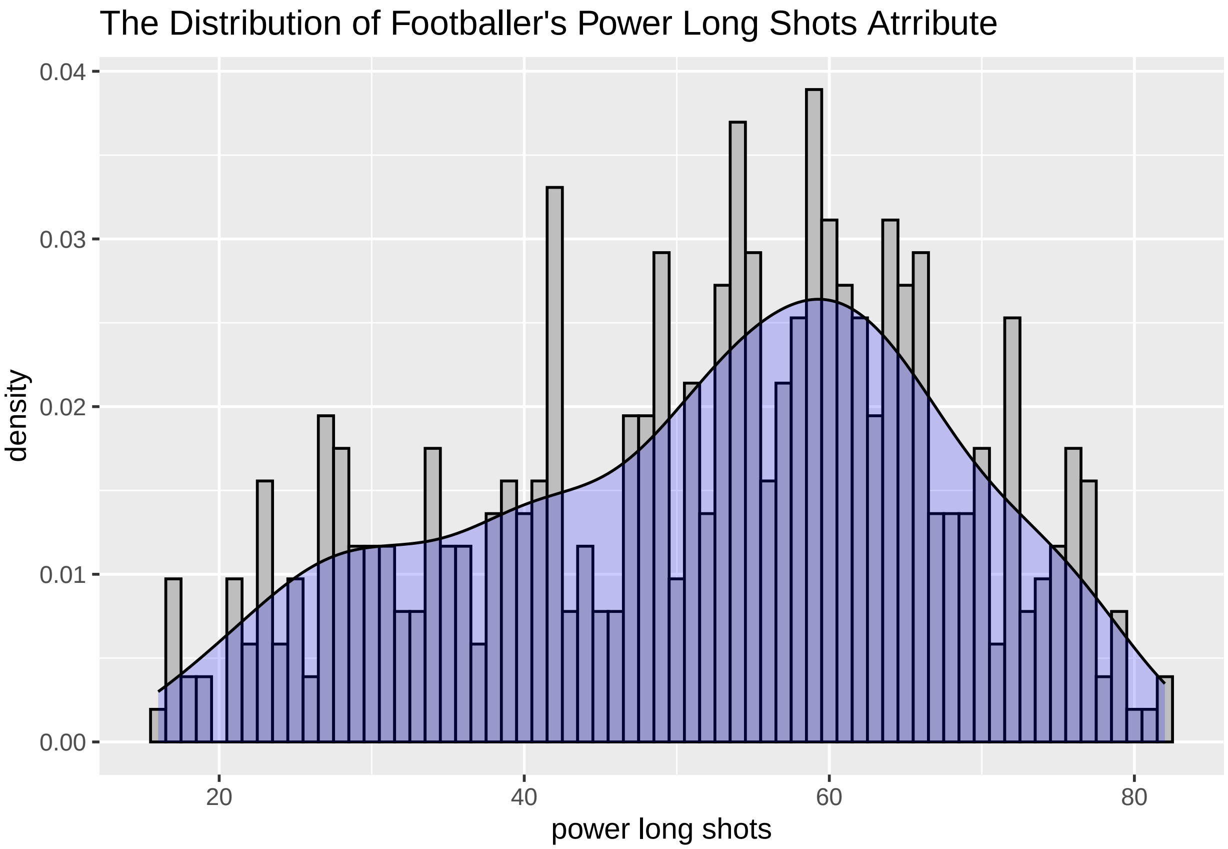
```
ggplot(football_df, aes(x=physic)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Physic Atrribute")
```



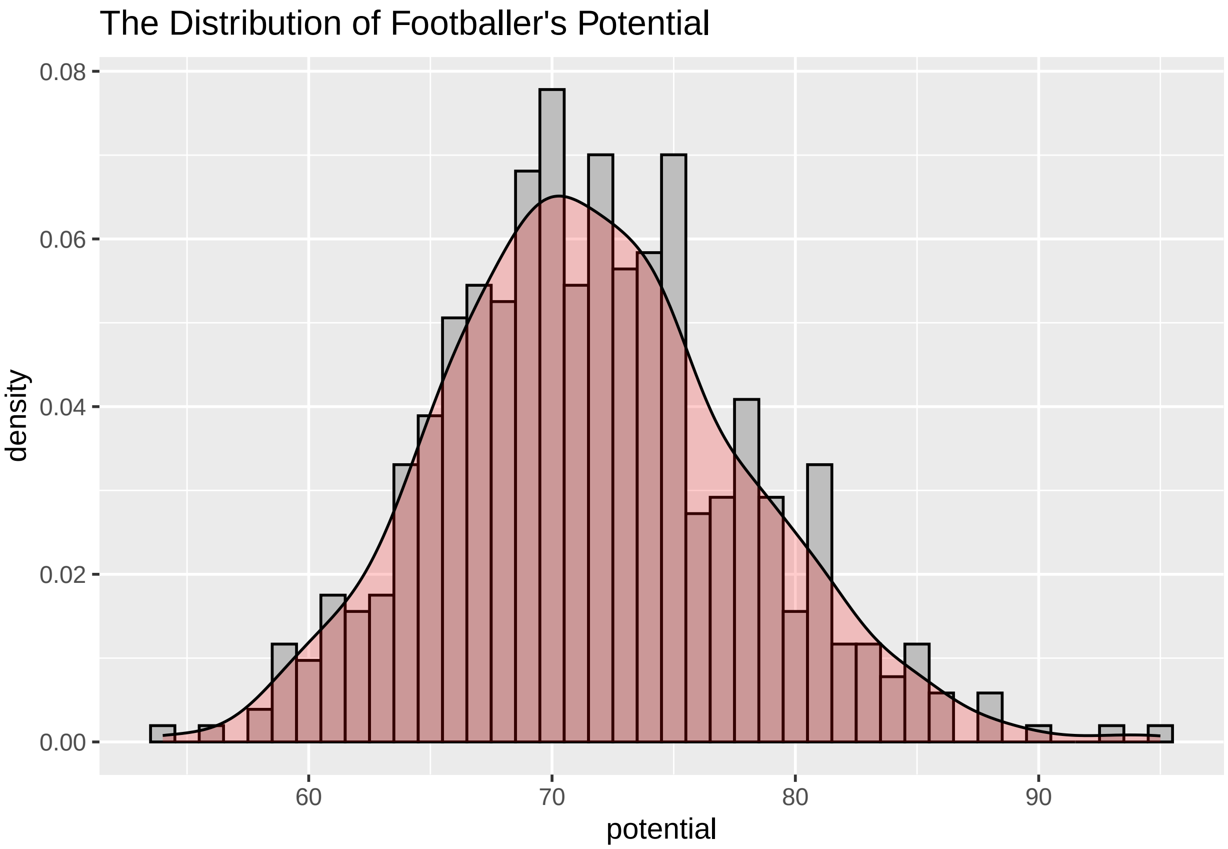
```r
ggplot(football_df, aes(x=`power strength`)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Power Strength Atrribu
```



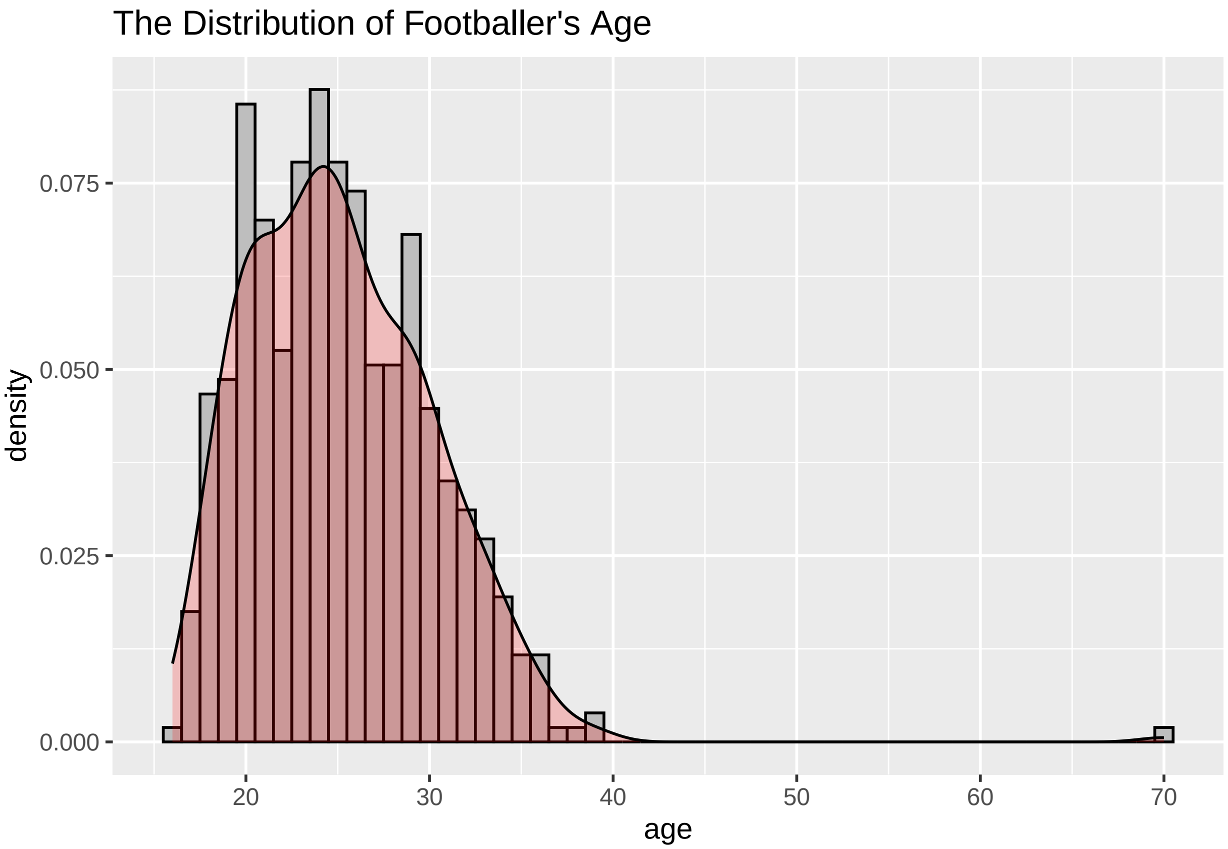
```
ggplot(football_df, aes(x=`power long shots`)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="blue") + ggtitle("The Distribution of Footballer's Power Long Shots Atrri
```



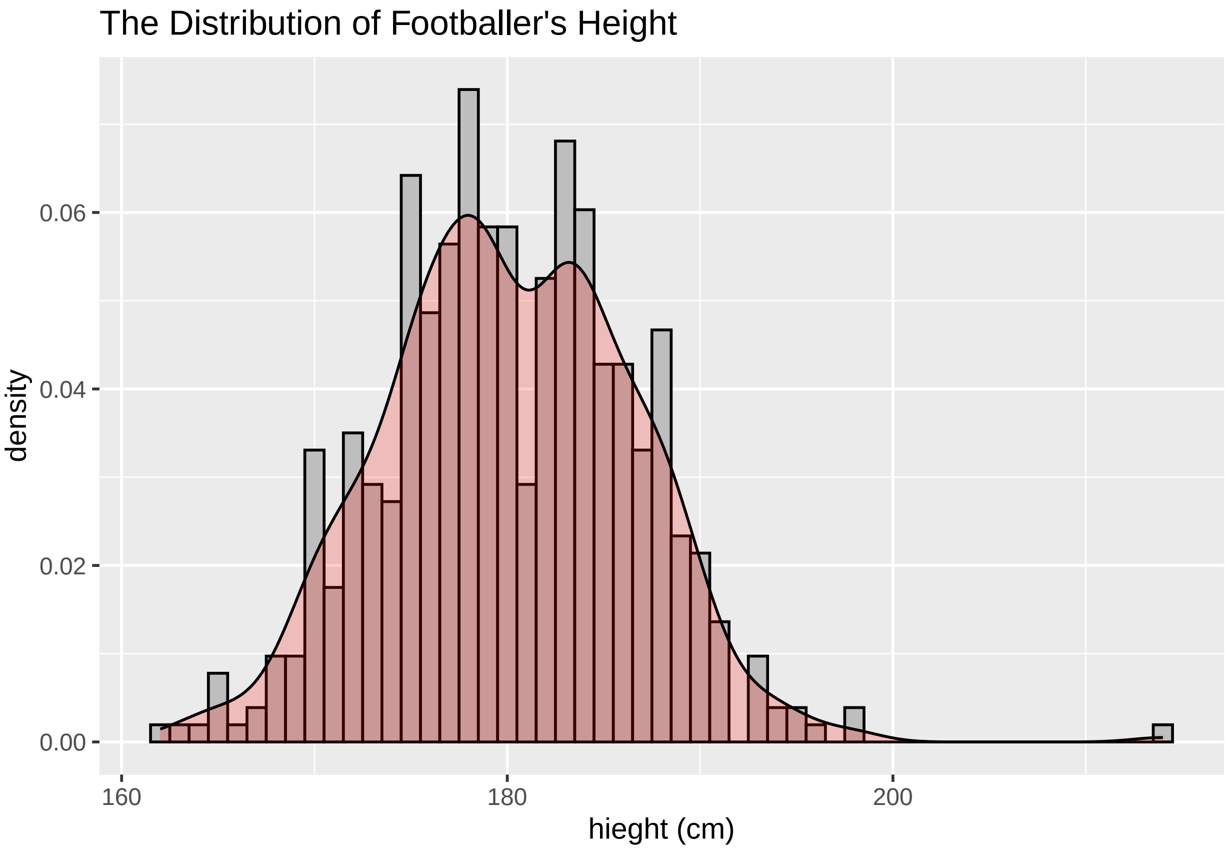
```
# histogram with density overlay for other continuous numerical variable (potential, age, height, and w
ggplot(football_df, aes(x=potential)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="red") + ggtitle("The Distribution of Footballer's Potential")
```



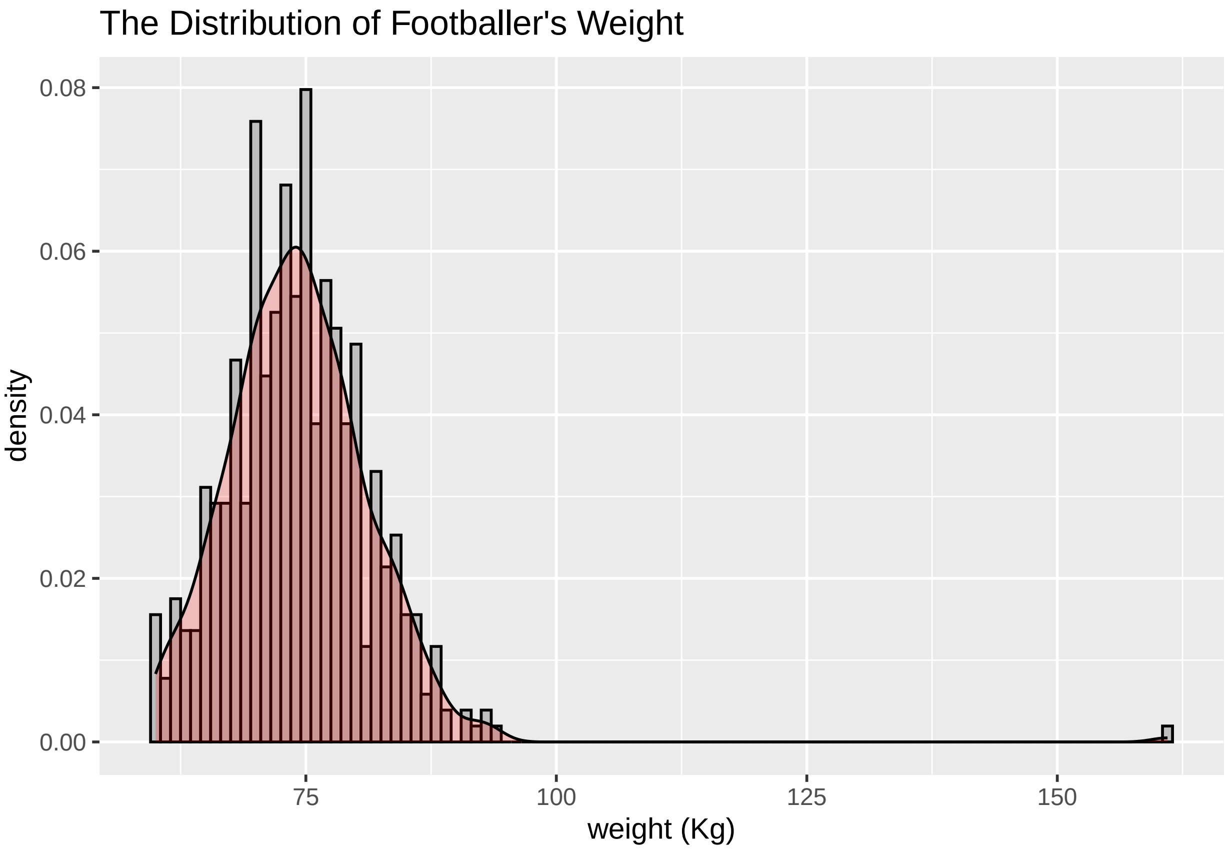
```
ggplot(football_df, aes(x=age)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="red") + ggtitle("The Distribution of Footballer's Age")
```



```
ggplot(football_df, aes(x=height)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="red") + ggtitle("The Distribution of Footballer's Height") + xlab("hiegh
```



```
ggplot(football_df, aes(x=weight)) +
  geom_histogram(aes(y=..density..), binwidth=1, colour="black", fill="grey") +
  geom_density(alpha=.2, fill="red") + ggtitle("The Distribution of Footballer's Weight") + xlab("weigh
```



All the attributes seem to have a normal distribution. However, there are some variables that seem to be slightly skewed such as **'pace'** and **'age'**. Additionally, the outliers are also visible within **'age'**, **'height'** and **'weight'**. Using **Shapiro-Wilk** normality test is a useful method of double checking if these continuous variables are normally distributed. This is beneficial as the information provides guidance on what further statistical analysis, such as the type of correlation and regression, can be performed 24.

```
shapiro.test(football_df$potential)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$potential
## W = 0.99109, p-value = 0.003441
```

```
shapiro.test(football_df$wage)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$wage
## W = 0.49599, p-value < 2.2e-16
```

```
shapiro.test(football_df$age)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  football_df$age
## W = 0.91342, p-value < 2.2e-16
```

```
shapiro.test(football_df$height)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$height
## W = 0.98729, p-value = 0.0001876
```

```
shapiro.test(football_df$weight)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$weight
## W = 0.85413, p-value < 2.2e-16
```

```
shapiro.test(football_df$pace)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$pace
## W = 0.98129, p-value = 3.609e-06
```

```
shapiro.test(football_df$shooting)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$shooting
## W = 0.97914, p-value = 1.019e-06
```

```
shapiro.test(football_df$passing)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$passing
## W = 0.99363, p-value = 0.02914
```

```
shapiro.test(football_df$dribbling)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$dribbling
## W = 0.98445, p-value = 2.644e-05
```

```
shapiro.test(football_df$defending)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$defending
## W = 0.9566, p-value = 3.7e-11
```

```
shapiro.test(football_df$physic)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$physic
## W = 0.98315, p-value = 1.144e-05
```

```
shapiro.test(football_df$`power strength`)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$‘power strength‘
## W = 0.98411, p-value = 2.111e-05
```

```
shapiro.test(football_df$`power long shots`)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  football_df$‘power long shots‘
## W = 0.9699, p-value = 8.931e-09
```

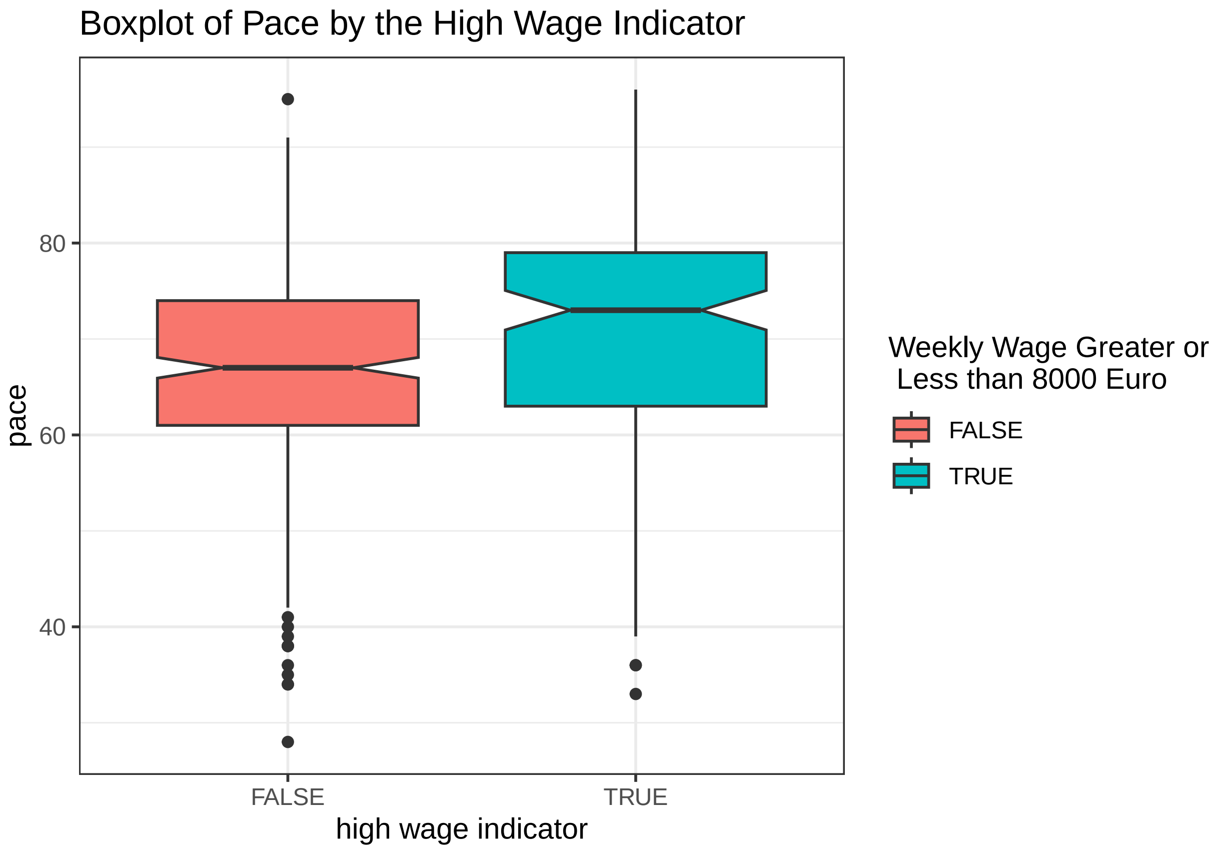
The p-values for all the Shapiro-Wilk normality tests are less than 0.05, and therefore are all significant. This indicates that all the continuous variables are normally distributed.

2.1.2 Multivariate EDA

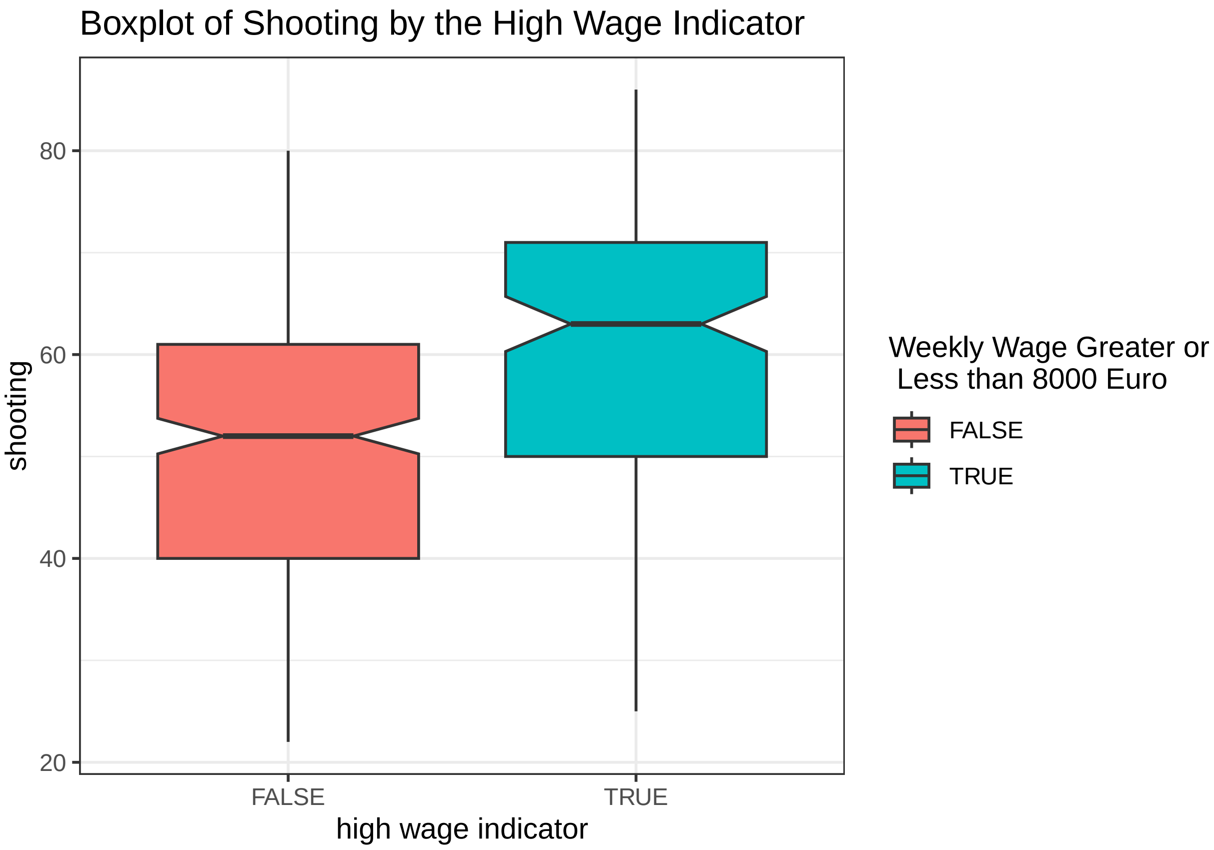
In this subsection, the data for multiple variables and the different relationships will be explored and visualised.

A boxplot will be performed to explore the relationship between high wage indicator with the player's attributes along with their other features 25, 26.

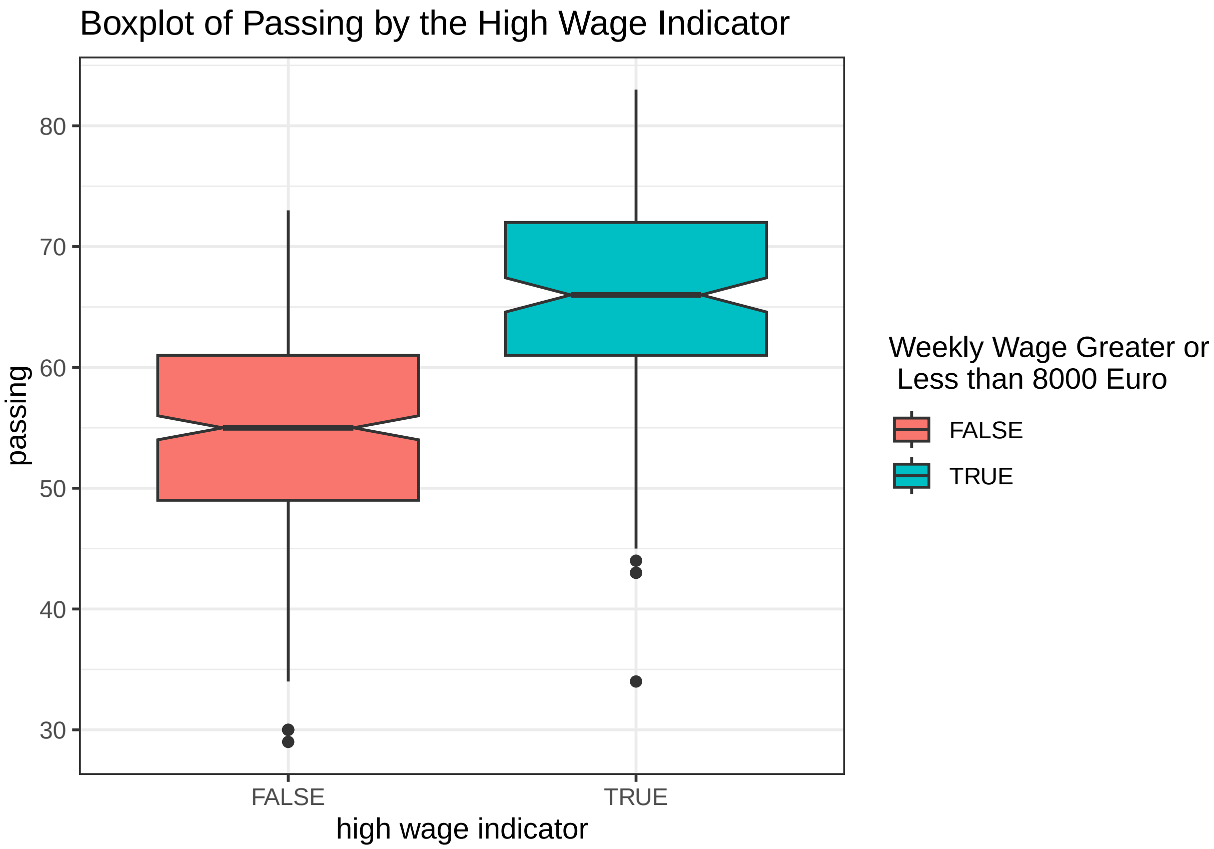
```
# Note: The notch is to show and display the confidence levels as well
# Relationship between pace and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=pace, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE
```



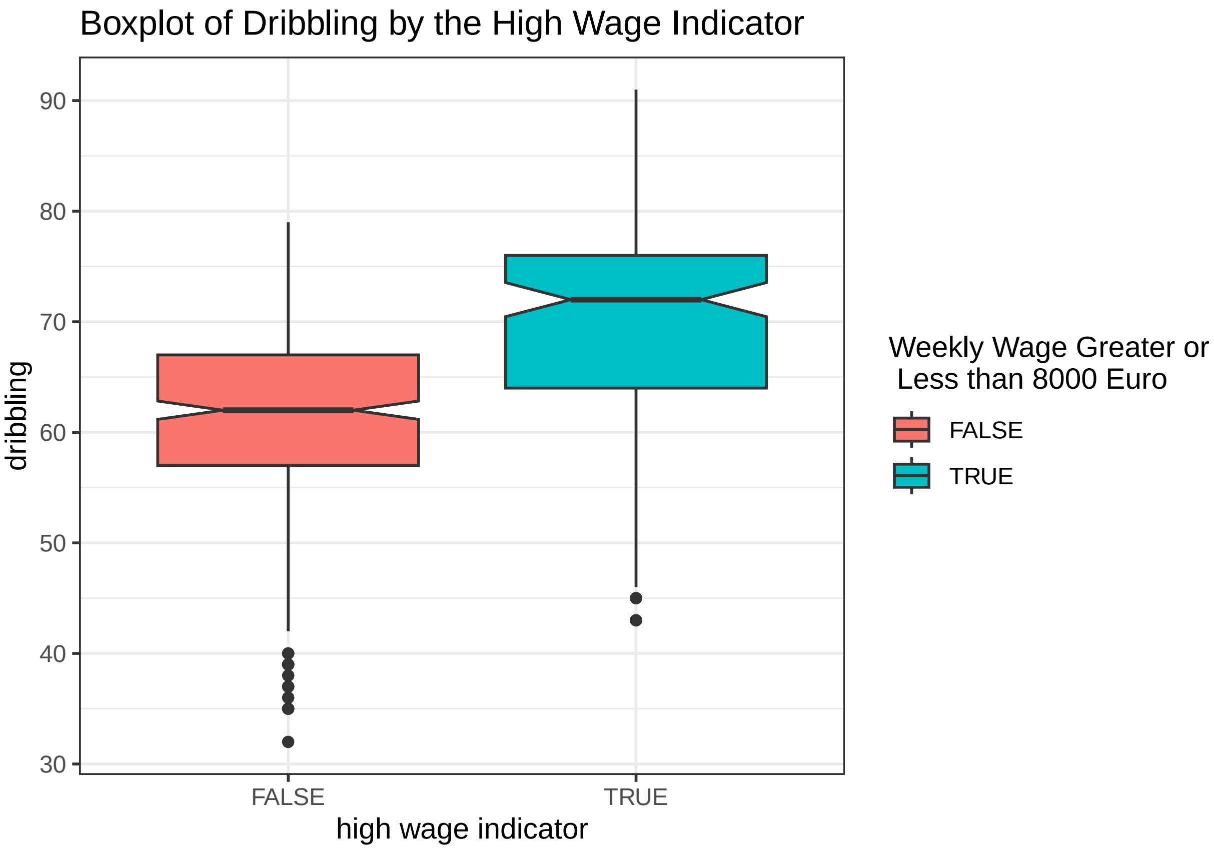
```
# Relationship between shooting and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=shooting, fill=high.wage.ind.log)) +geom_boxplot(notch=TI
```



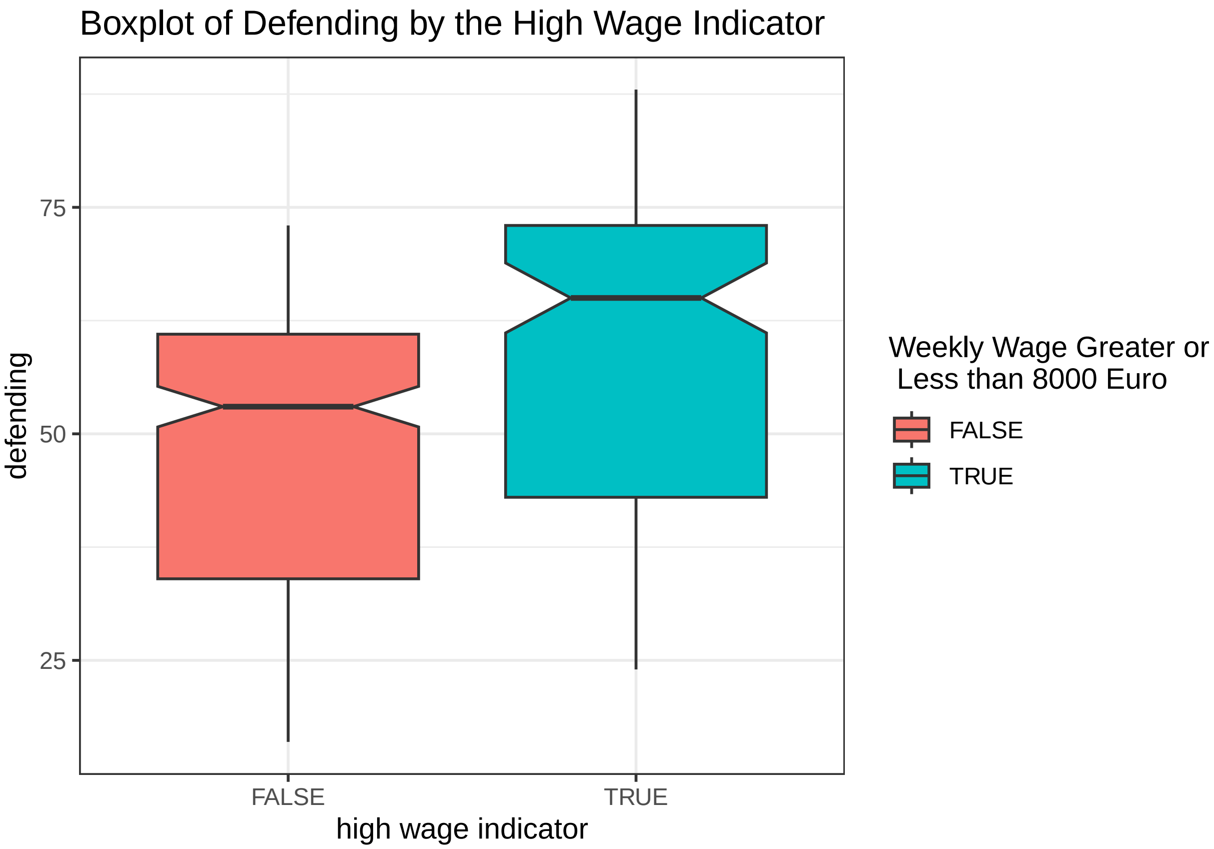
```
# Relationship between passing and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=passing, fill=high.wage.ind.log)) +geom_boxplot(notch=TRU
```



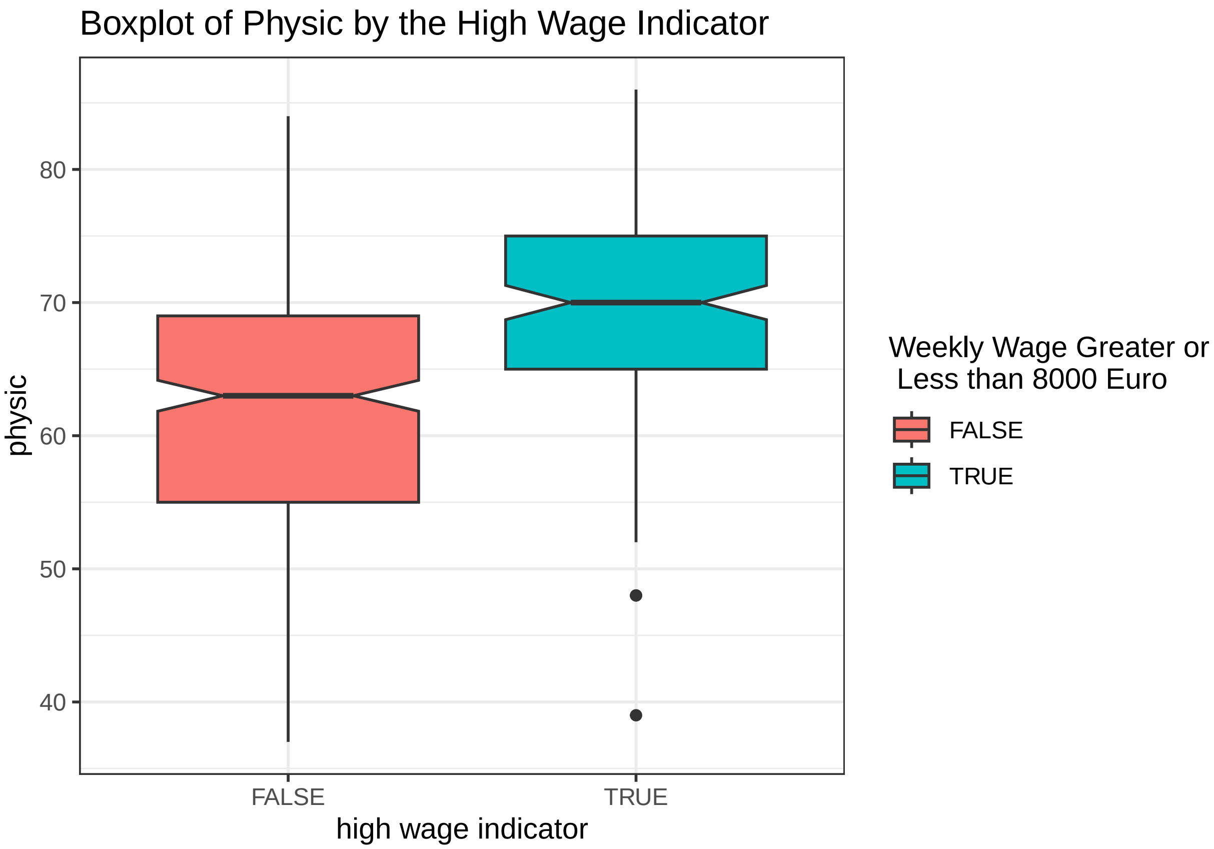
```
# Relationship between dribbling and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=dribbling, fill=high.wage.ind.log)) +geom_boxplot(notch='
```



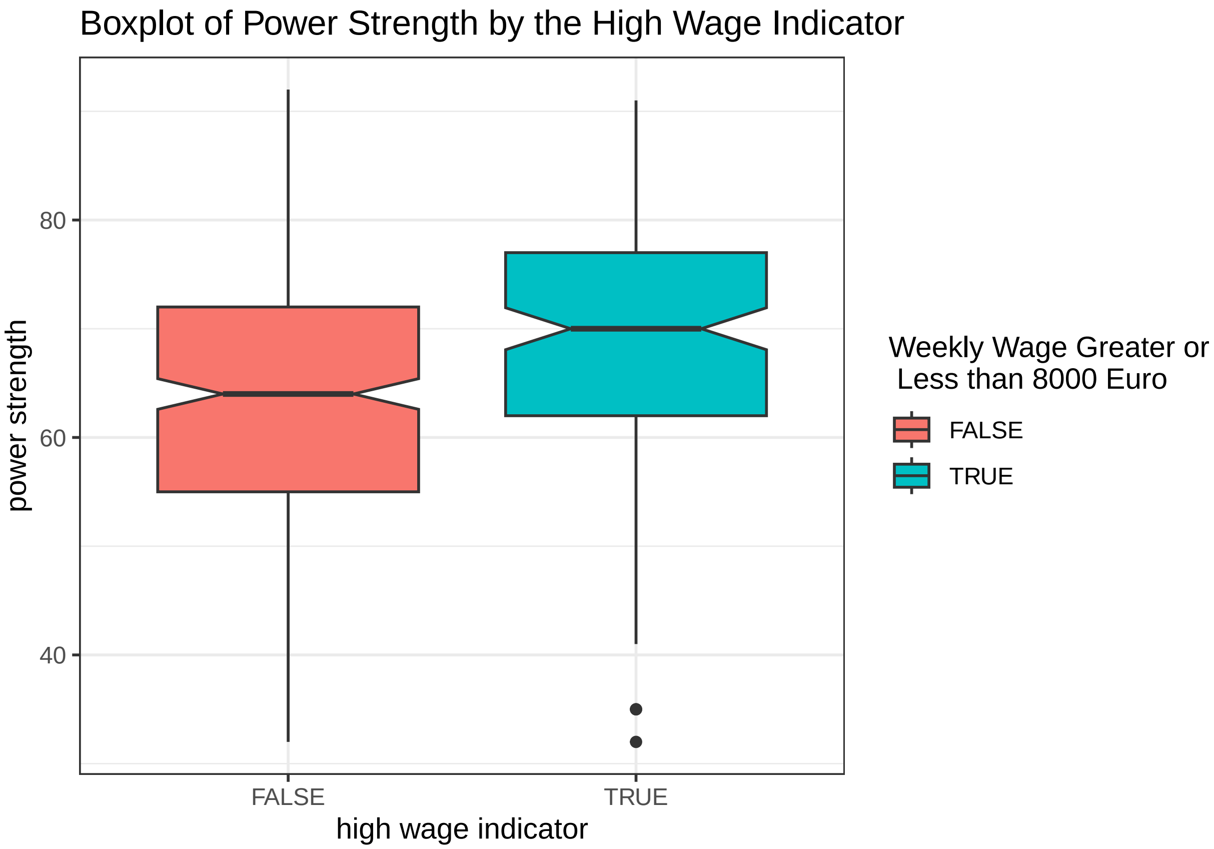
```
# Relationship between defending and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=defending, fill=high.wage.ind.log)) +geom_boxplot(notch=
```



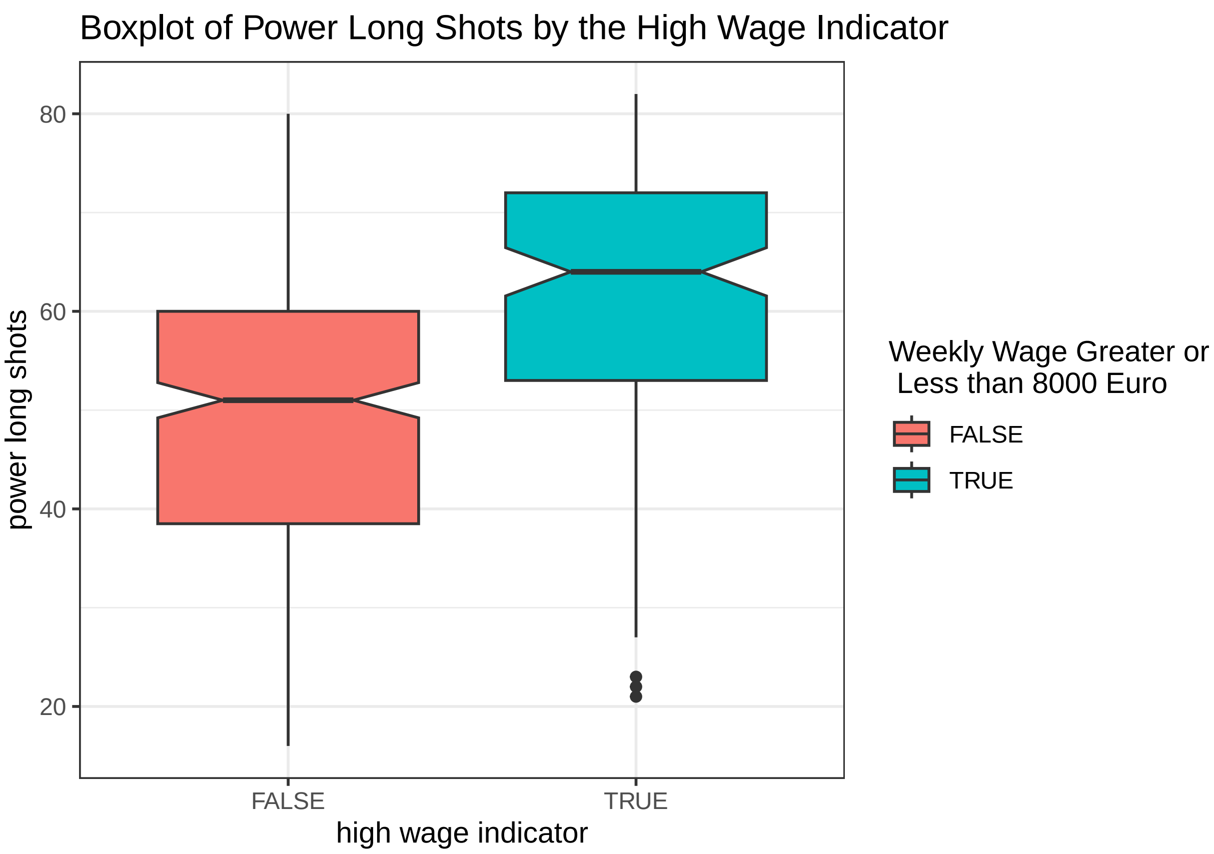
```
# Relationship between physic and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=physic, fill=high.wage.ind.log)) +geom_boxplot(notch=TRU
```



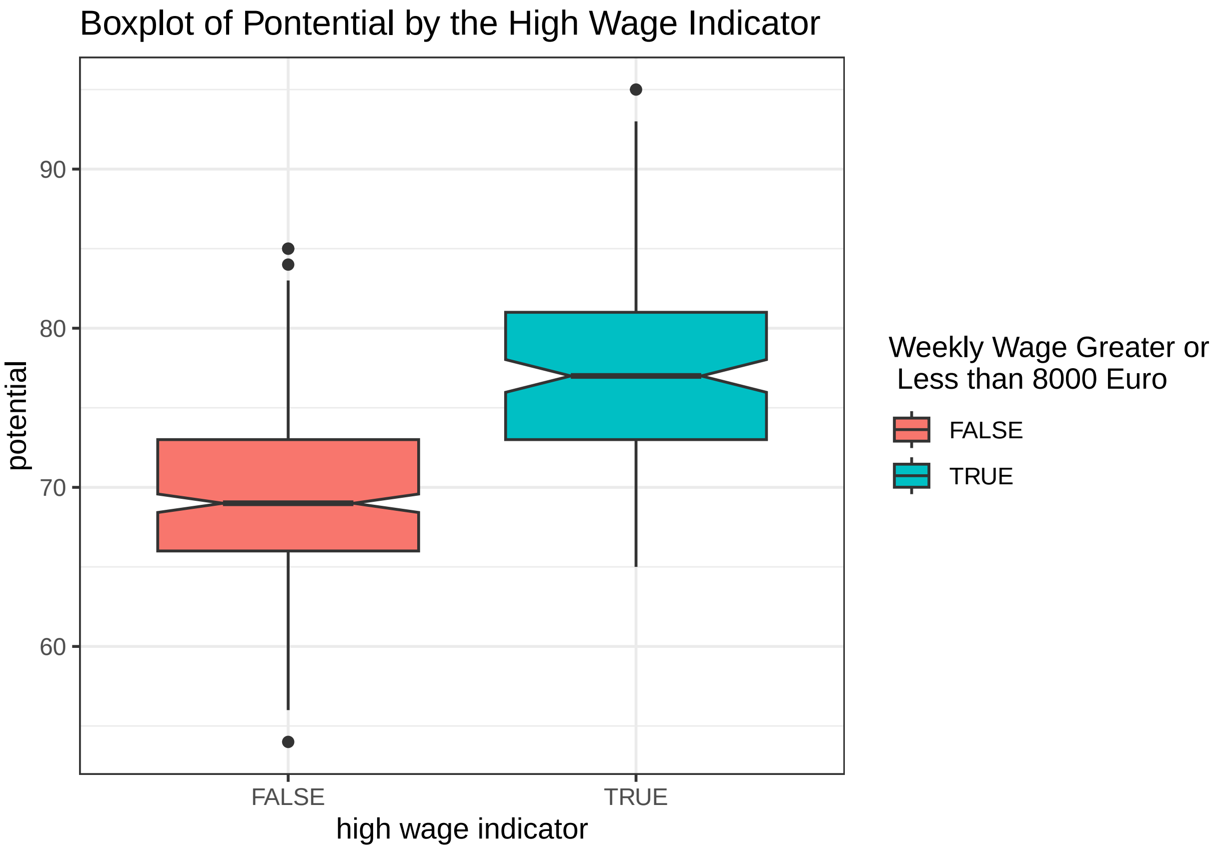
```r
# Relationship between power strength and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=`power strength`, fill=high.wage.ind.log)) + geom_boxplo
```



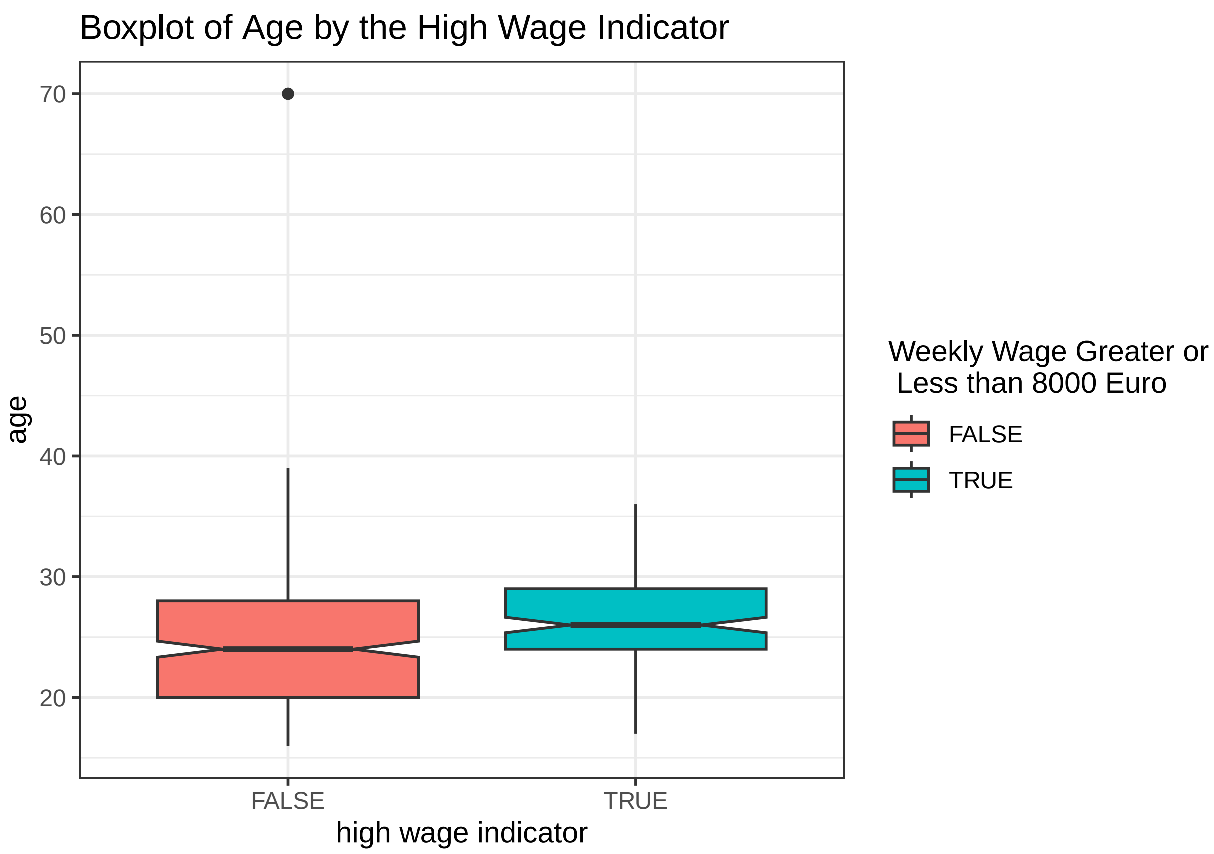
```
# Relationship between power long shot and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=`power long shots`, fill=high.wage.ind.log)) +geom_boxpl
```



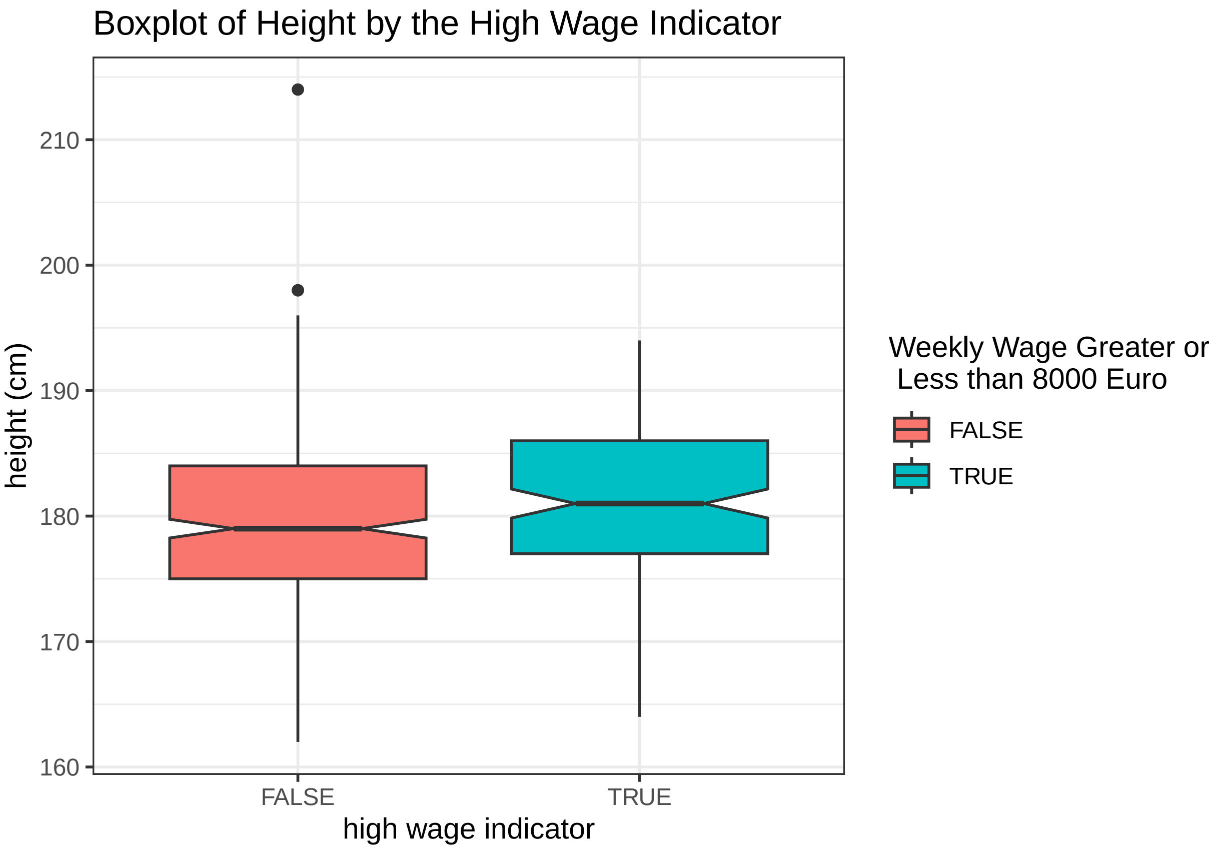
```
# Relationship between potential and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=potential, fill=high.wage.ind.log)) +geom_boxplot(notch='
```



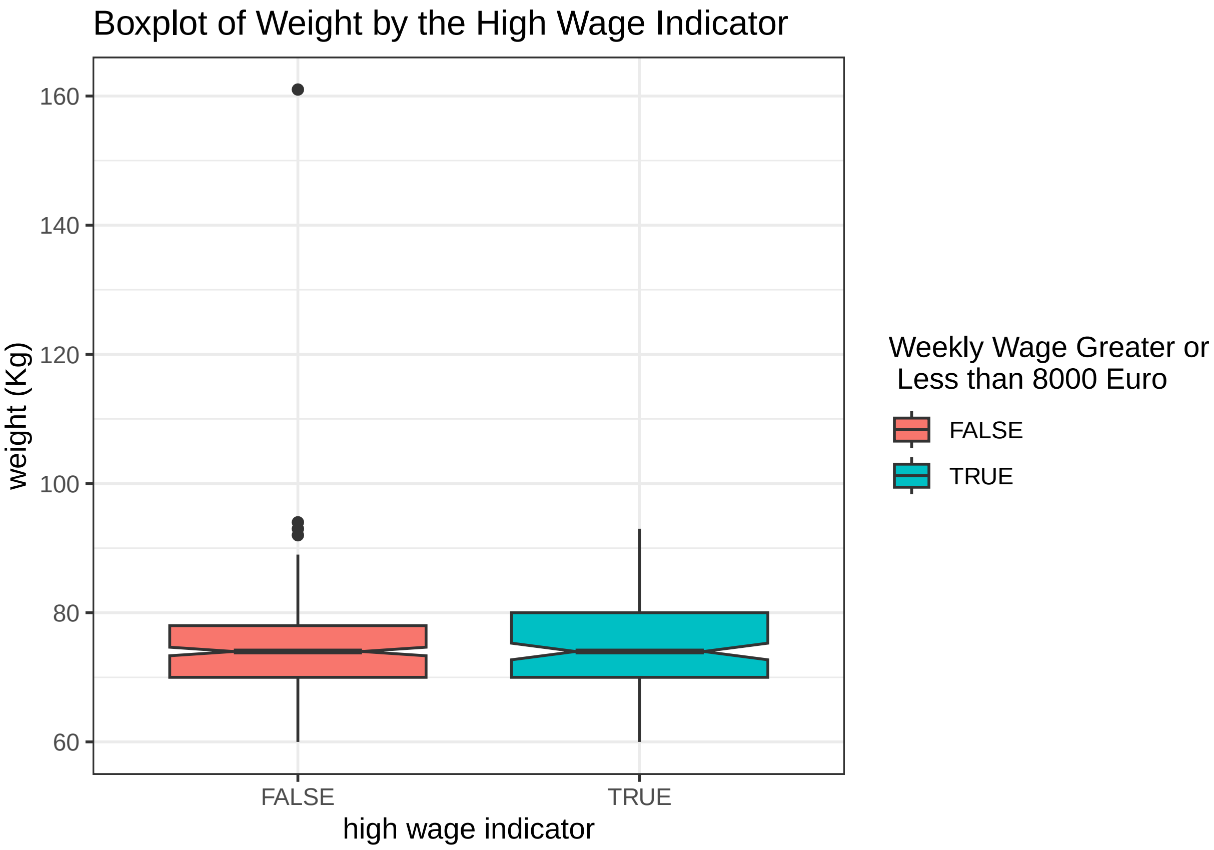
```
# Relationship between age and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=age, fill=high.wage.ind.log)) +geom_boxplot(notch=TRUE) +
```



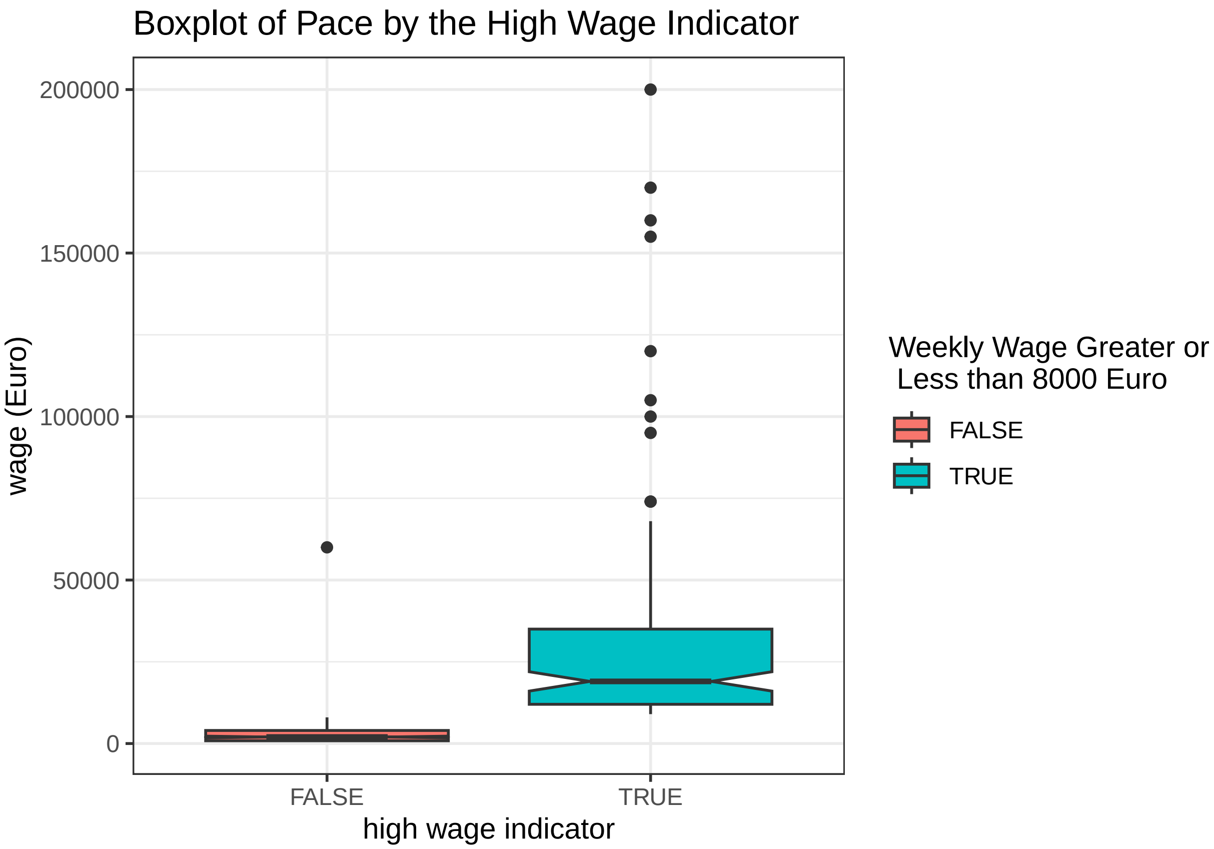
```
# Relationship between height and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=height, fill=high.wage.ind.log)) +geom_boxplot(notch=TRU
```



```
# Relationship between weight and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=weight, fill=high.wage.ind.log)) +geom_boxplot(notch=TRU
```

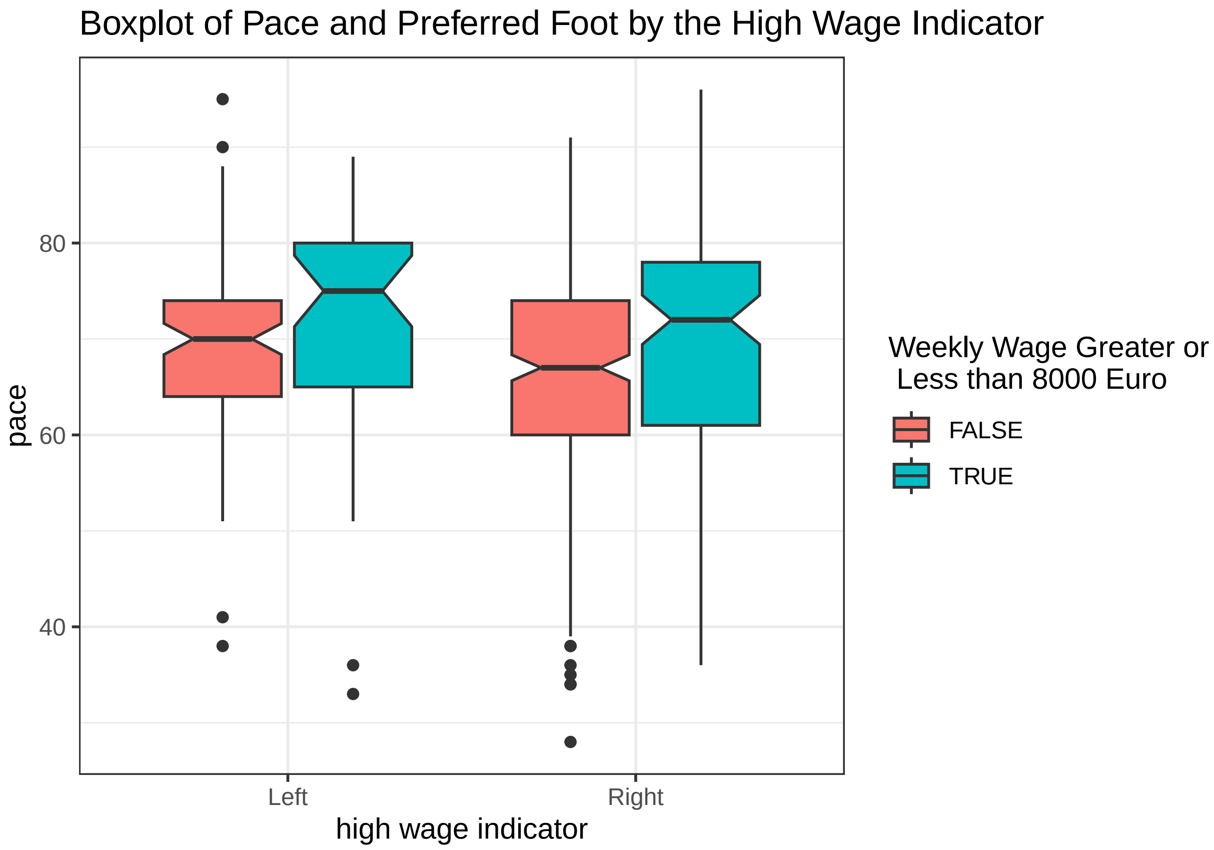


```
# Relationship between wage and the high wage indicator
ggplot(football_df, aes(x=high.wage.ind.log, y=wage, fill=high.wage.ind.log)) +geom_boxplot(notch=TRUE)
```

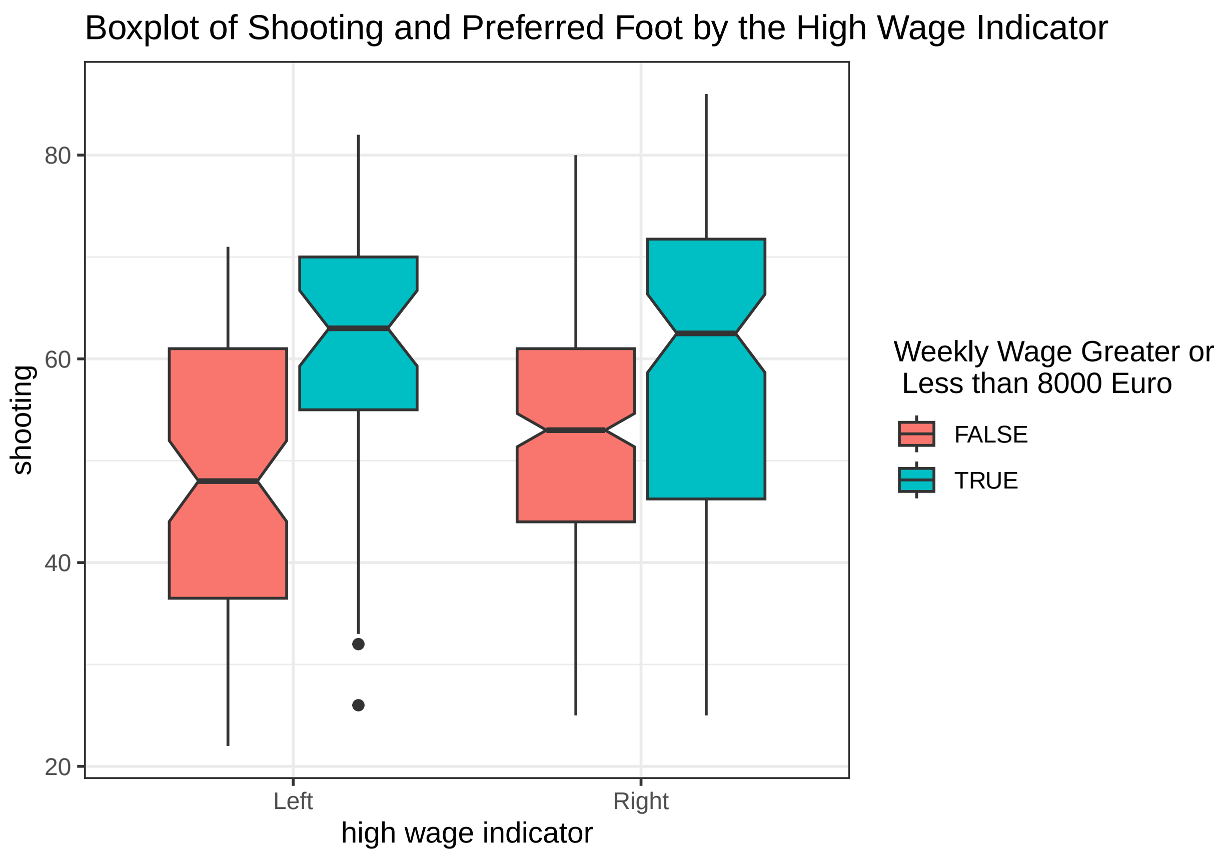


The variations for whether a player earns 8000 Euro weekly or not, varies within the different attributes and other characteristics of the player. For instance, defending has the greatest variation whereas weight has very little variation. Despite this, all attributes and characteristics of players seem to have higher median for those earning 8000 Euro or more. This could suggest that those players are more skillfull, therefore earning more. This is useful information as it provides an insight for further statistical investigation for the given research for section 4. The next process carried out is to see if the player's preferred foot when playing has an impact on this.

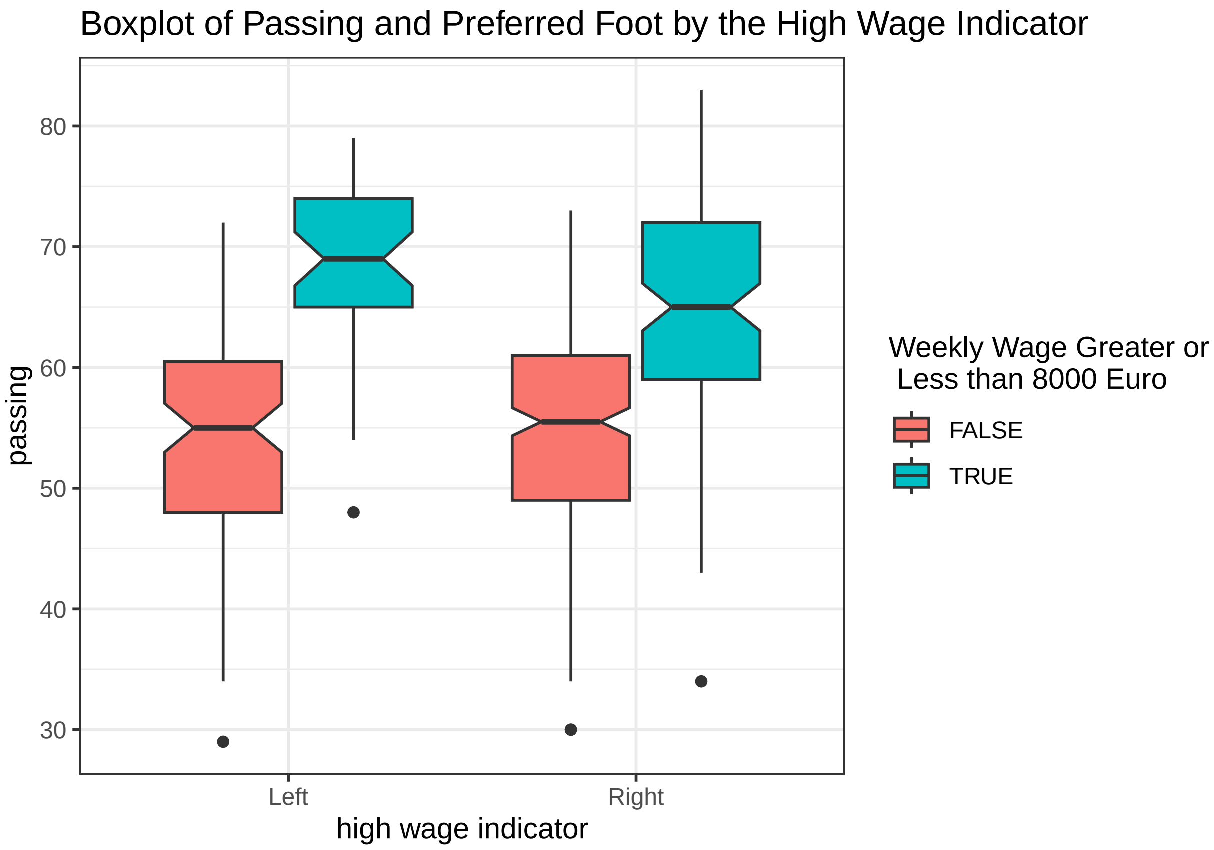
```
# Relationship between pace and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=pace, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE)
```



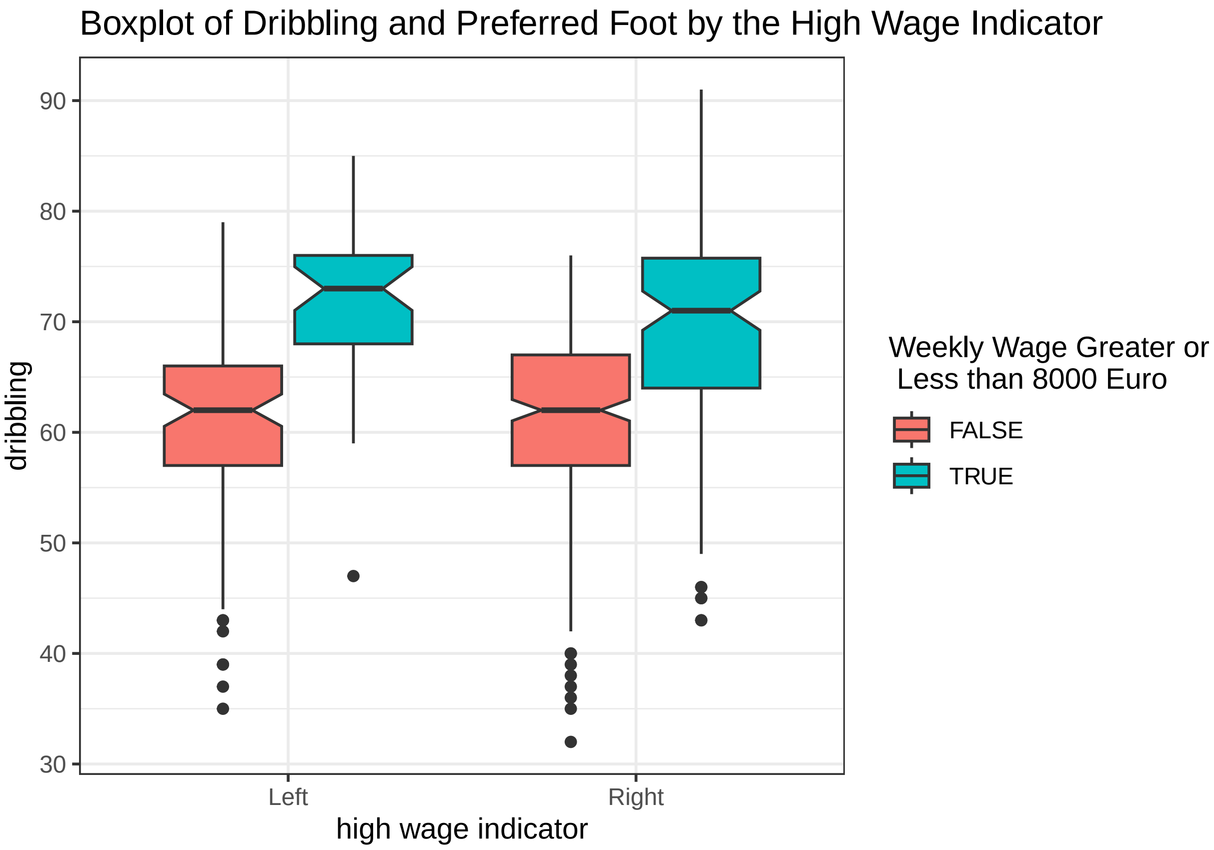
```
# Relationship between shooting and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=shooting, fill=high.wage.ind.log)) + geom_boxplot(notch=T
```



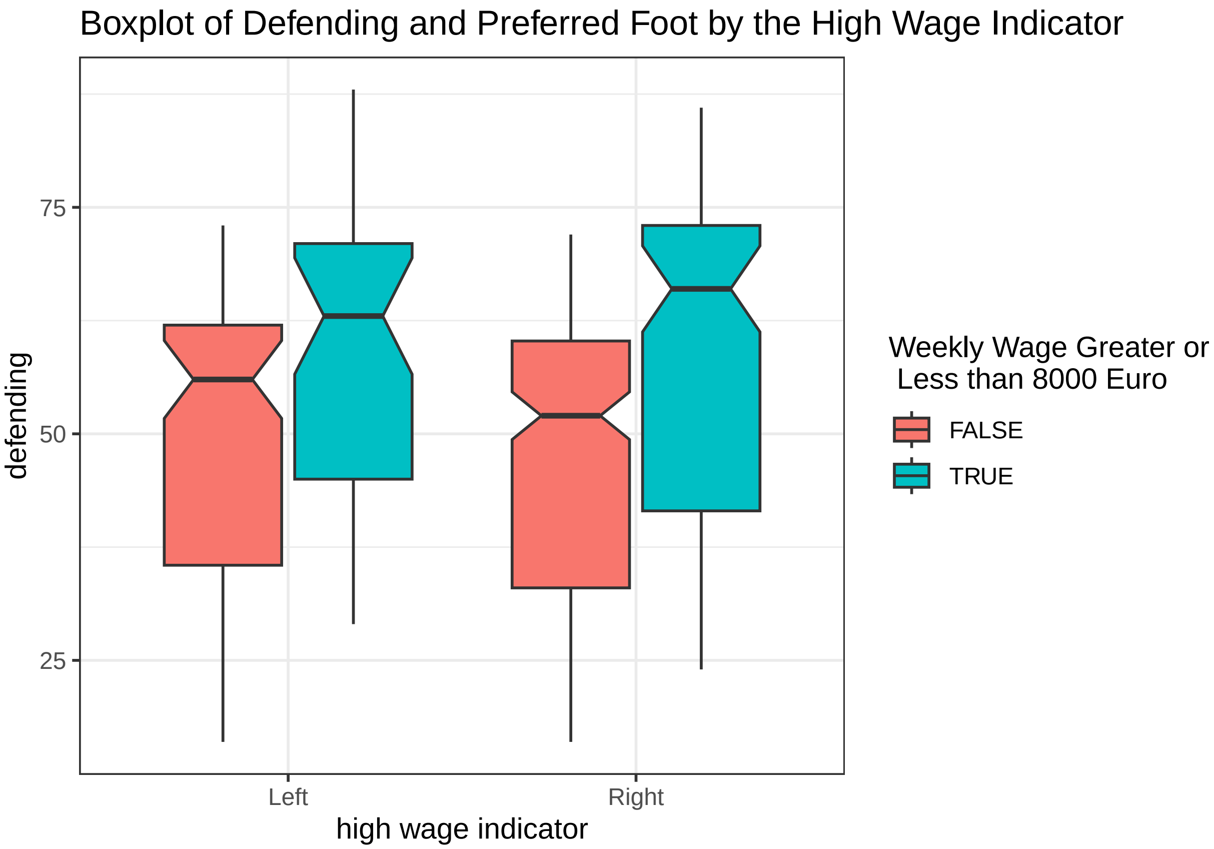
```
# Relationship between passing and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=passing, fill=high.wage.ind.log)) + geom_boxplot(notch=TRU
```



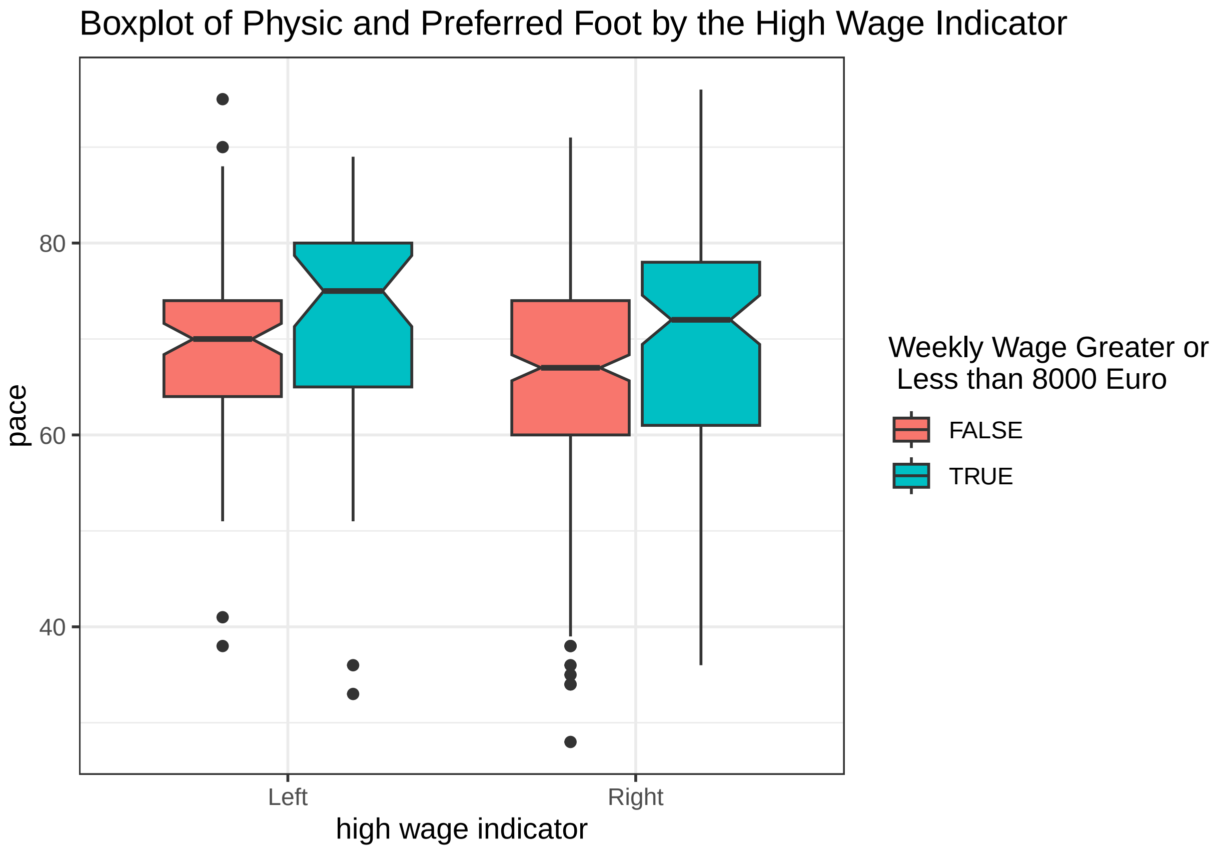
```
# Relationship between dribbling and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=dribbling, fill=high.wage.ind.log)) + geom_boxplot(notch=
```



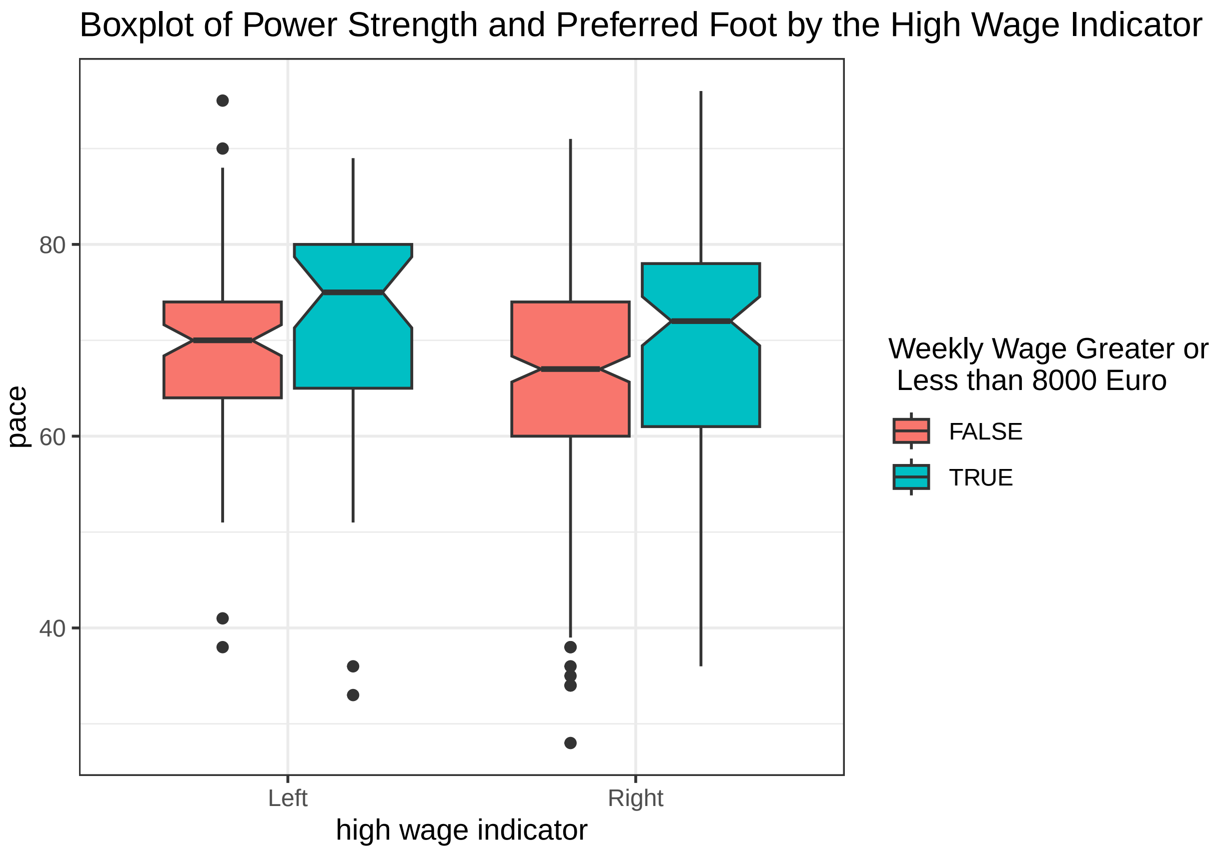
```
# Relationship between defending and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=defending, fill=high.wage.ind.log)) + geom_boxplot(notch=
```



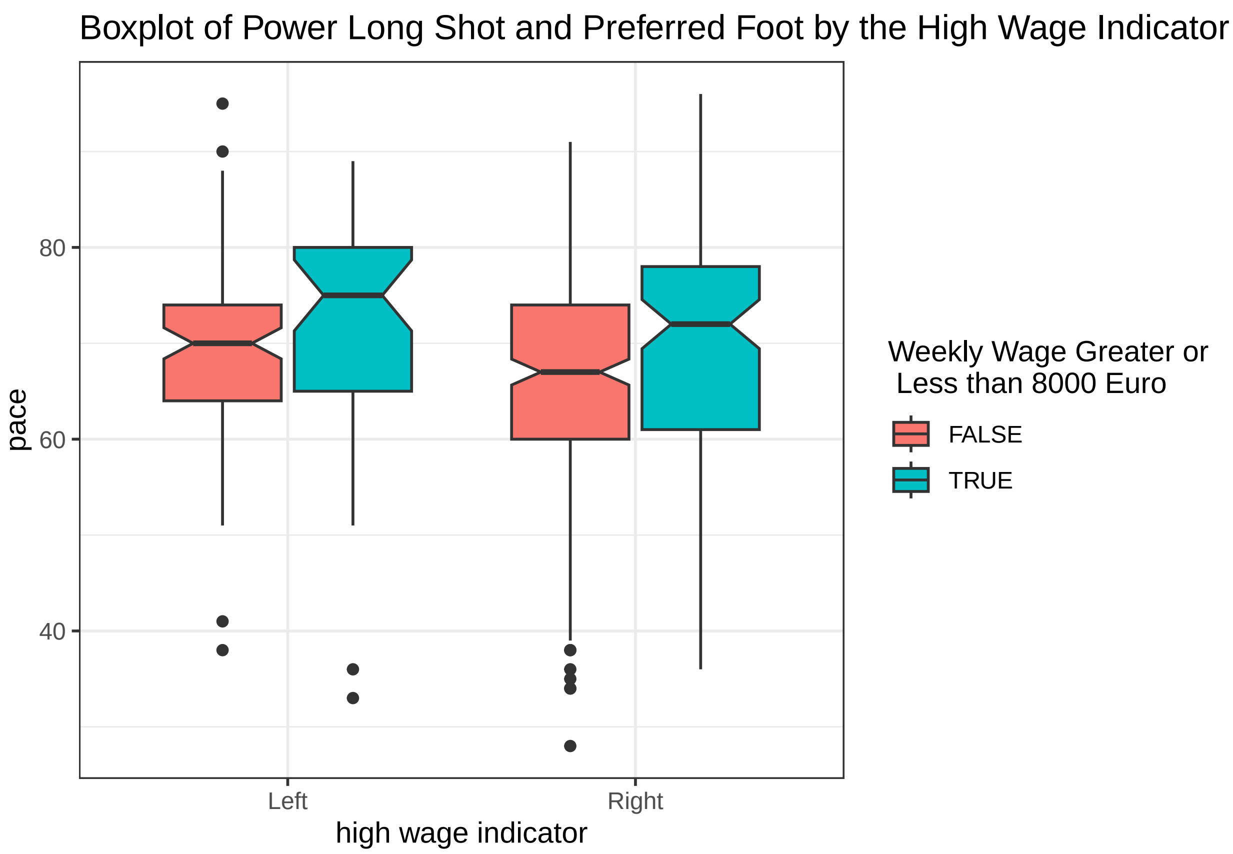
```
# Relationship between physic and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=pace, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE)
```



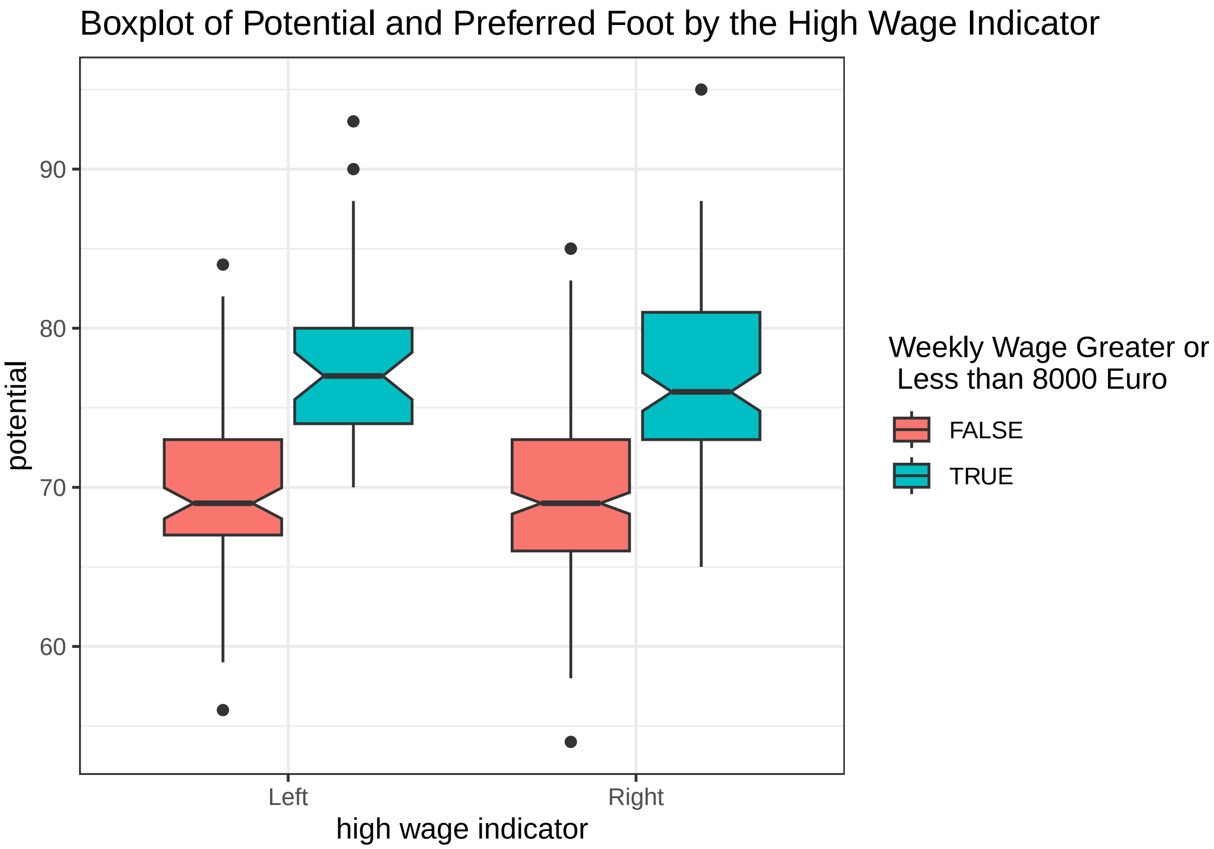
```r
# Relationship between power strength and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=pace, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE)
```


Boxplot of Power Strength and Preferred Foot by the High Wage Indicator

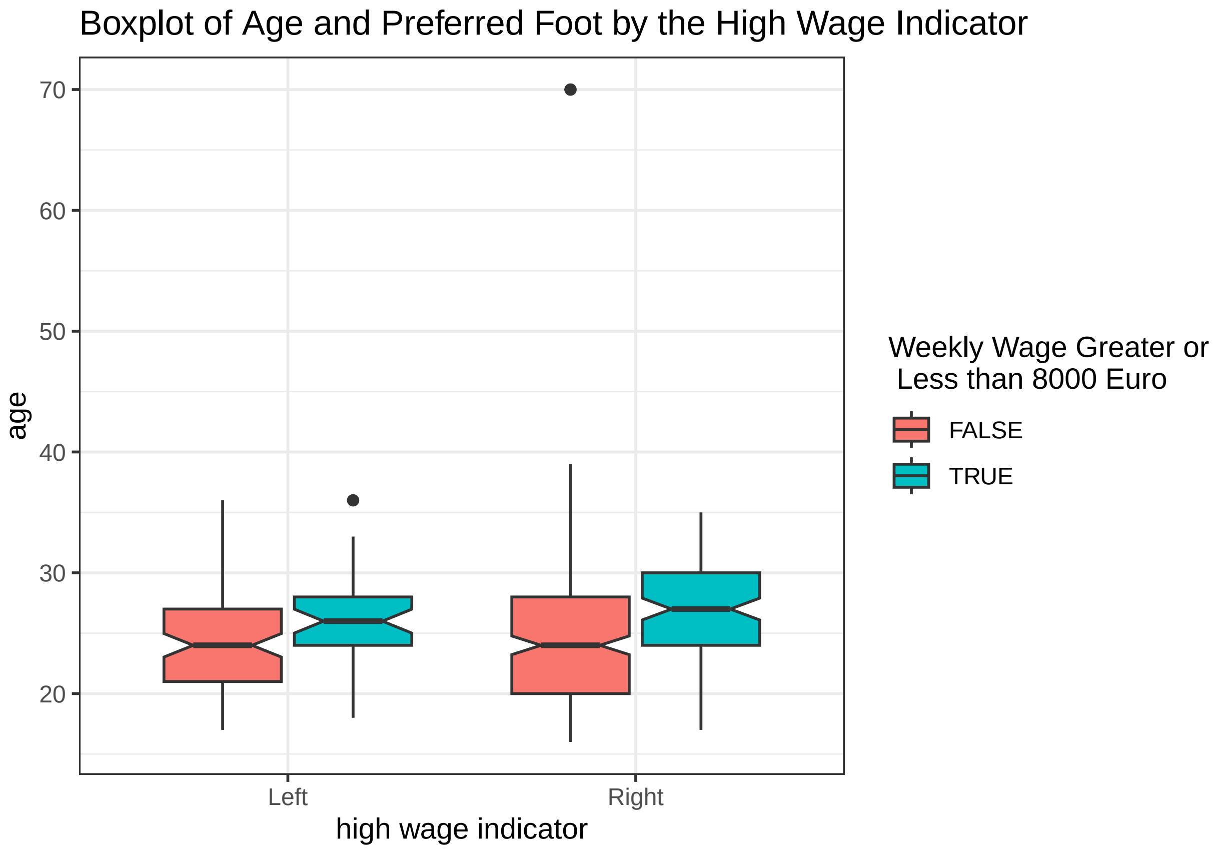
```
# Relationship between power long shot and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=pace, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE)
```



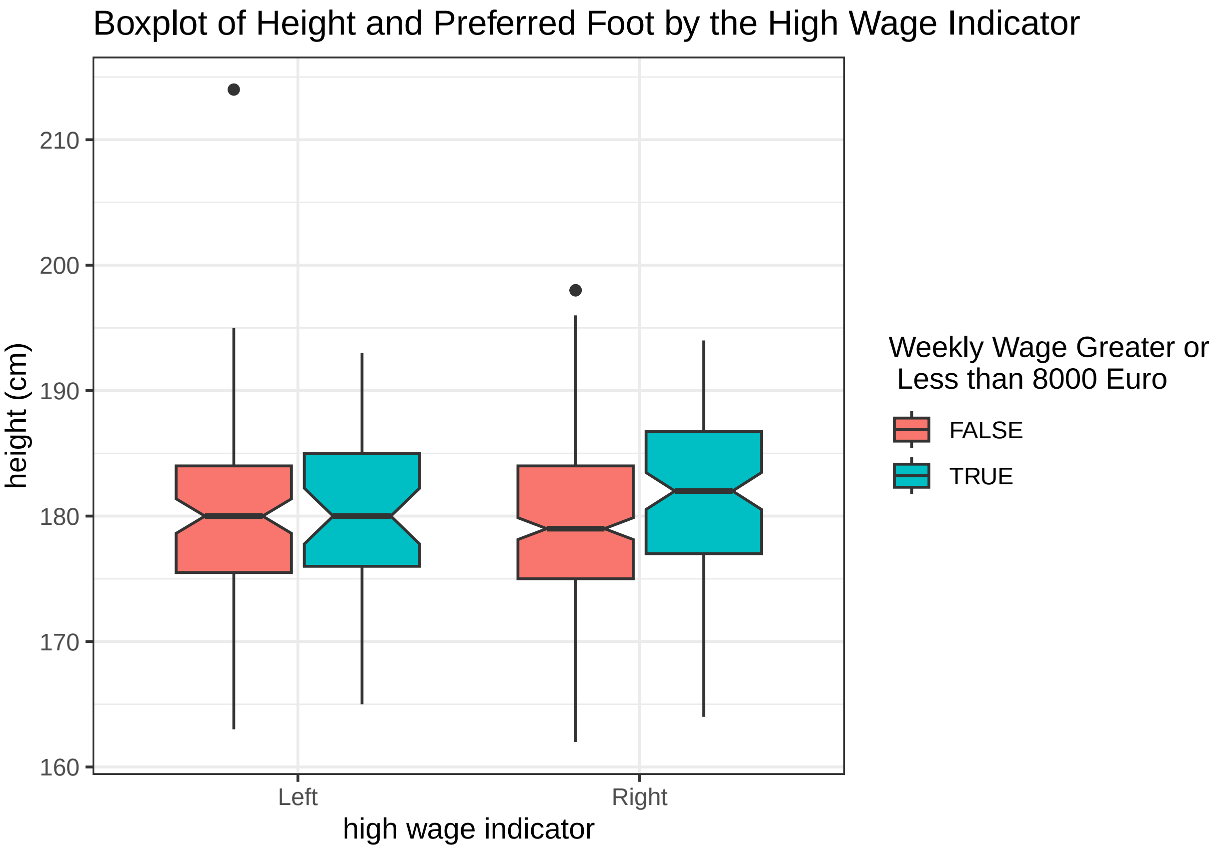
```
# Relationship between potential and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=potential, fill=high.wage.ind.log)) + geom_boxplot(notch=
```



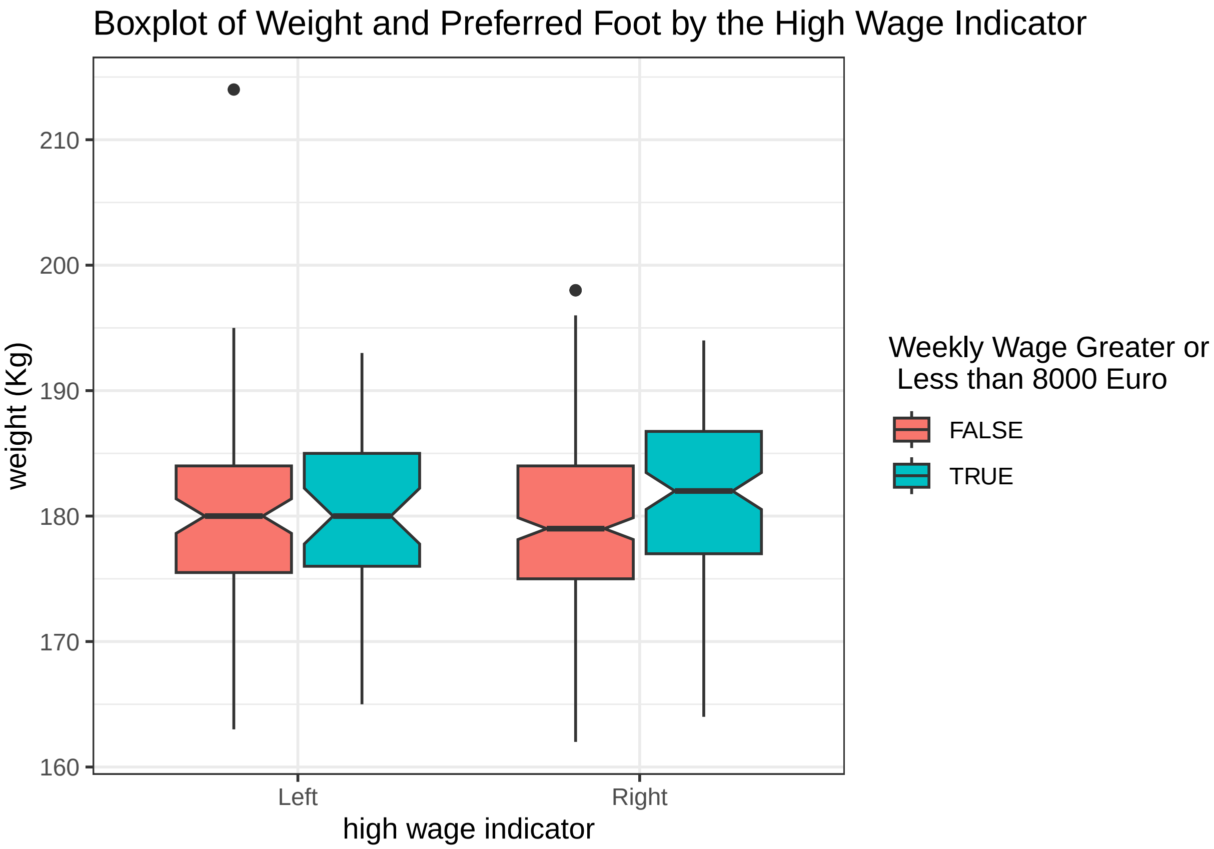
```
# Relationship between age and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=age, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE) +
```



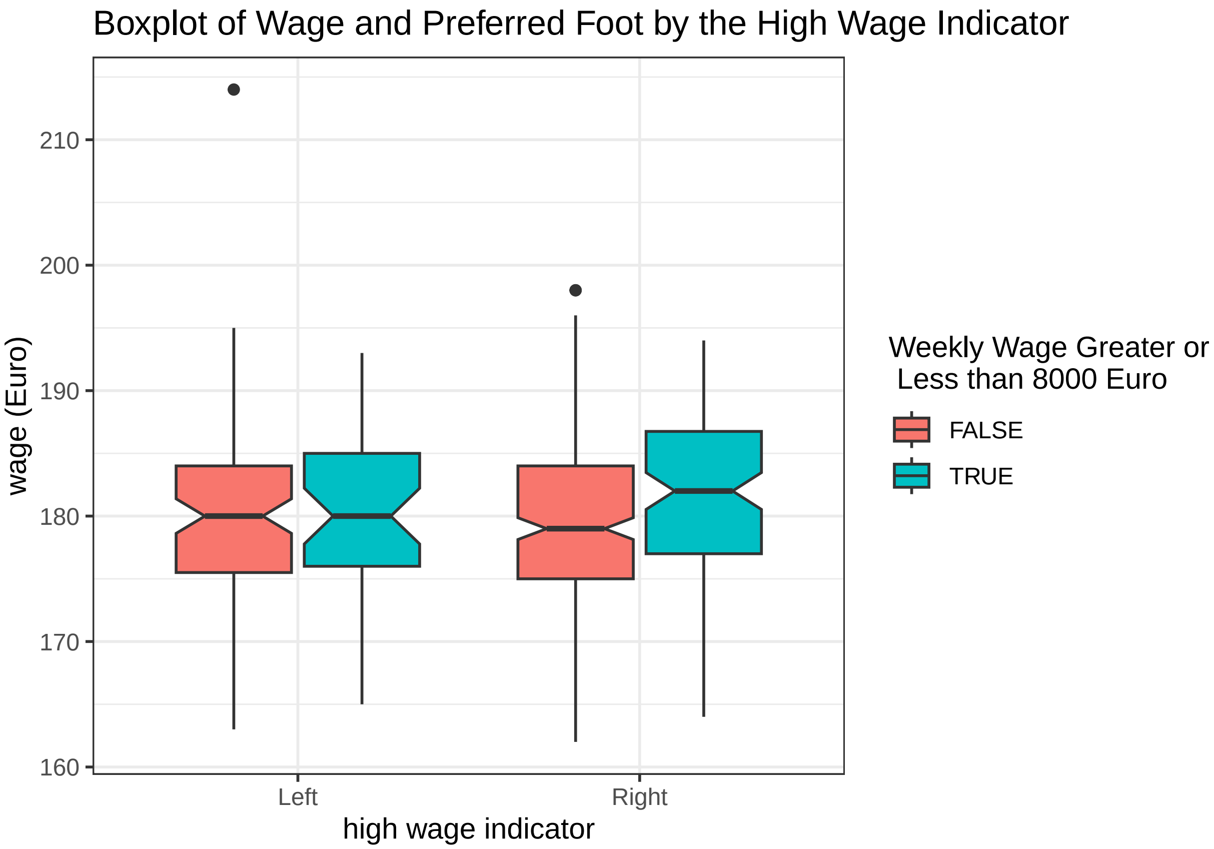
```
# Relationship between height and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=height, fill=high.wage.ind.log)) + geom_boxplot(notch=TRUE
```



```
# Relationship between weight and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=height, fill=high.wage.ind.log)) + geom_boxplot(notch=TRU
```



```
# Relationship between wage and preferred foot by the high wage indicator
ggplot(football_df, aes(x=`preferred foot`, y=height, fill=high.wage.ind.log)) + geom_boxplot(notch=TRU
```



These boxplots show that whether the player prefers playing with their right or left foot does not have an impact on their attributes or the amount of money they earn.

A heatmap is a very powerful way of showing the correlation between all continuous variables. As all the variables are normally distributed, extracting and implementing Pearson's correlation coefficient, `r`, is appropriate.

Before creating and plotting the heatmap, the main following stages need be carried out:

1. Create the correlation matrix with the help of `cor()` function, which will extract `r`
2. Create and apply user-defined functions to return the upper triangular parts of the correlation matrix. This will represent the "white" part of the axis of the heatmap.
3. Reorder the correlation matrix for clearer visualisation of the heatmap by creating and applying another user-defined function
4. Melt the correlation matrix, which essentially "melts" and converts the matrix into a molten data frame using the `melt()` function.

   - In particular, it converts the reordered matrix to change it from wide-format matrix to long-format matrix in order to create two different axis for the correlation matrix. This eases the transition of the data when plotting the heatmap

27 28 29

```
# 1. Create the correlation matrix with the help of `cor()` function
# Subsetting and creating a new dataframe, containg all the continous numerical variable
football_num_new <- football_df %>%
  select(-c("ID", "club name", "preferred foot", "high wage indicator", "high.wage.ind.log"))
# Now creating the correlation matrix using cor() and rounding the numbers to 2 decimal places
```

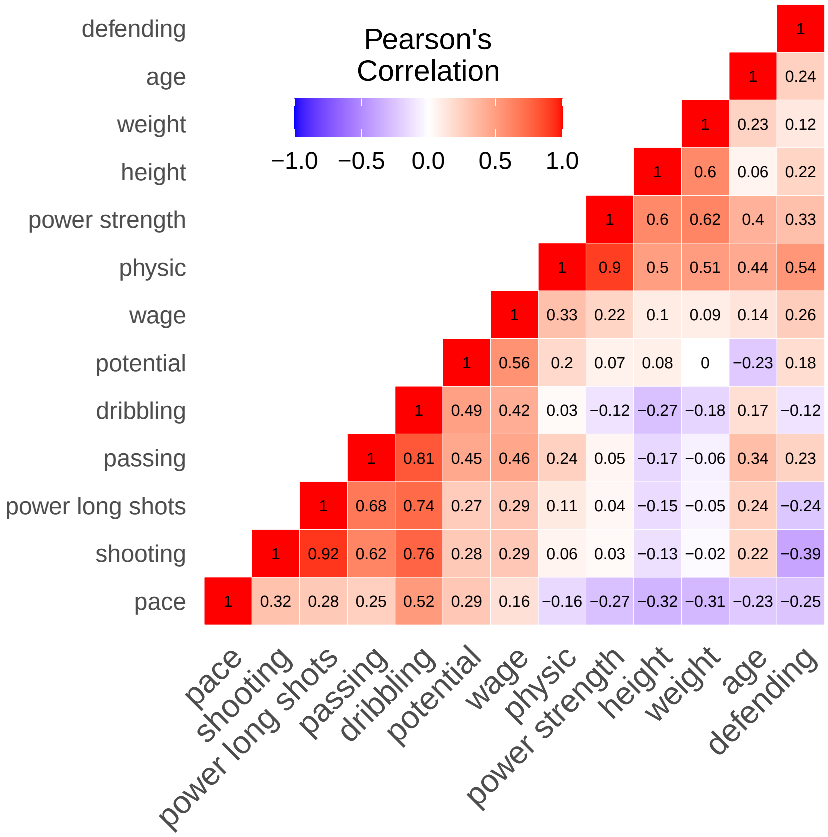
```
corr_matrix <- round(cor(football_num_new), 2)
corr_matrix
```

```
##                  potential wage   age height weight  pace shooting passing
## potential             1.00 0.56 -0.23   0.08   0.00  0.29     0.28    0.45
## wage                  0.56 1.00  0.14   0.10   0.09  0.16     0.29    0.46
## age                  -0.23 0.14  1.00   0.06   0.23 -0.23     0.22    0.34
## height                0.08 0.10  0.06   1.00   0.60 -0.32    -0.13   -0.17
## weight                0.00 0.09  0.23   0.60   1.00 -0.31    -0.02   -0.06
## pace                  0.29 0.16 -0.23  -0.32  -0.31  1.00     0.32    0.25
## shooting              0.28 0.29  0.22  -0.13  -0.02  0.32     1.00    0.62
## passing               0.45 0.46  0.34  -0.17  -0.06  0.25     0.62    1.00
## dribbling             0.49 0.42  0.17  -0.27  -0.18  0.52     0.76    0.81
## defending             0.18 0.26  0.24   0.22   0.12 -0.25    -0.39    0.23
## physic                0.20 0.33  0.44   0.50   0.51 -0.16     0.06    0.24
## power strength        0.07 0.22  0.40   0.60   0.62 -0.27     0.03    0.05
## power long shots      0.27 0.29  0.24  -0.15  -0.05  0.28     0.92    0.68
##                  dribbling defending physic power strength power long shots
## potential             0.49      0.18   0.20           0.07             0.27
## wage                  0.42      0.26   0.33           0.22             0.29
## age                   0.17      0.24   0.44           0.40             0.24
## height               -0.27      0.22   0.50           0.60            -0.15
## weight               -0.18      0.12   0.51           0.62            -0.05
## pace                  0.52     -0.25  -0.16          -0.27             0.28
## shooting              0.76     -0.39   0.06           0.03             0.92
## passing               0.81      0.23   0.24           0.05             0.68
## dribbling             1.00     -0.12   0.03          -0.12             0.74
## defending            -0.12      1.00   0.54           0.33            -0.24
## physic                0.03      0.54   1.00           0.90             0.11
## power strength       -0.12      0.33   0.90           1.00             0.04
## power long shots      0.74     -0.24   0.11           0.04             1.00
```

```
# 2. Create and apply user-define functions to return the lower and upper triangular parts of the corre

# This involves getting the lower and upper triangles part of the correlation martix which "Returns a m

# First creating user-defined function that gets the upper triangle of the correlation matrix
get_UpperTri <- function(corr_matrix){
    corr_matrix[lower.tri(corr_matrix)]<- NA
    return(corr_matrix)
}

# Will now apply this function to the correlation matrix:
upper_Tri <- get_UpperTri(corr_matrix)
#upper_Tri

# 3. Reorder the correlation matrix for clearer visualisation of the heatmap
# First creating the user-defined function that involves retrieving the computed distance matrix ('as.d
reorder_corr_matrix <- function(corr_matrix){
# Use correlation between variables as distance
  distance <- as.dist((1-corr_matrix)/2)      # computes the distance matrix computation that are measu
  hcluster <- hclust(distance)                      # computes and applies a Hierarchical clustering, w
  corr_matrix <-corr_matrix[hcluster$order, hcluster$order]
}

# Now applying this reorder function to correlation matrix.
corr_matrix <- reorder_corr_matrix(corr_matrix)

# Will now apply the upper triangular function to this new reodered correlation matrix
upper_Tri <- get_UpperTri(corr_matrix)

# 4. Melt the new and reordered correlation matrix using the 'melt()' function
# melted_corr_matrix <- melt(corr_matrix)
# head(melted_corr_matrix)
#View(melt(corr_matrix))

# Then melt the correlation matrix
melted_corr_matrix <- melt(upper_Tri, na.rm = TRUE)
```

The desired heatmap will be created and plotted, displaying the correlation coefficient values.

```
# Creating and Plotting the heatmap
heatmap <- ggplot(melted_corr_matrix, aes(Var2, Var1, fill = value)) +      # value of matrix are the d
  geom_tile(color = "white") +                                              # fills each square or 'til
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0, limit = c(-1,1), space =
                       name="Pearson's\nCorrelation") +                     # Adding gradient colour sc
  theme_minimal() +                                                         # Applying a minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank()) +                                   # Adding other extra detail
  coord_fixed()                                                             # Ensuring that coordinatio

# Creating another heatmap with the correaltion coefficient values
rValues_heatmap <- heatmap +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 2) +    # The size of the text for
  theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal") +        # Again adding details for the display such as making that
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
rValues_heatmap
```



This heatmap shows that there are many positive correlations of varying degrees as indicated by the different shades of red. For instance, there is a positive correlation between **'potential'** and **'wage'**, indicating that as the potential increases in value, the weekly wage earning tends to increase, where the r value of 0.59, indicating a positive correlation. There are also negative correlations indicated by the different shades of blue. For example, the r value at -0.39 for the correlation between **'shooting'** and **'defending'** shows a negative correlation, implying that as the shooting increases in value, the defending decreases. Additionally, there is no correlation between the **'potential'** and **'weight'** as the r value is 0. There also seems to be some high levels of correlation, for example between **'shooting'** and power long shots, where the r value is 0.92.

An additional correlation test will be performed to double check if the no correlation between **'potential'** and **'weight'** is significant:

```
cor.test(football$weight_kg, football_df$potential)
```

```
##
##  Pearson's product-moment correlation
##
## data:  football$weight_kg and football_df$potential
## t = -0.078108, df = 512, p-value = 0.9378
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.08991216  0.08305997
## sample estimates:
##          cor
## -0.003451919
```

The output confirms that there is no correlation between these two. This, along with the heatmap, is useful information to help in drawing conclusions with regards to the given research question for section 3.

2.3 Additional insights and issues

Performing an in-depth EDA has provided an insight into the different types of relationships between the different variables. This information will provide guidance on the suitable statistical approach to perform and answer the research questions in relation to player's potential and binary target attributes, for section 3 and 4, respectively, as well as helping in drawing conclusions.

3. Modelling

3.1 Build a model for player potential

The heatmap correlation provides an insight and will help in carrying out and drawing conclusions for the following research question: ***"What factors have an effect on the football player's potential?"***

Performing ANCOVA, a type of analysis of variance, is the appropriate statistical approach. This tests all the possible factors, including numerical and categorical variables, which affect or have a relationship with player's potentials. The Null (H_0) and Alternative Hypothesis (H_1) is stated below:

H_0 : There are no variables that have an effect on the player's potentials

H_1 : There are variables that have an effect on the player's potentials

This statistical approach will involve using the `lm()` function, which is part of regression analysis, to fit multiple regression lines into one model:

$$y = a + b1 \times x1 + b2 \times x2....bk \times xk$$

Where :

y = dependent variable (the potential in this case)

x = independent or explanatory variable

a = y-intercept level (when x is 0)

b = the slope (effect on x) {this is also known as beta}

Creating multiple regression lines for one model will help to determine if the correlation between the potential dependent variable and the different explanatory implies causation.

The explanatory variable will include all numerical and categorical variables except for **weight**, **club name** and **high.wage.ind.log**. **Weight** is not included as there was no correlation between **potential** and **weight** as indicated from the EDA. **Club name** is not included as there are many different clubs all with very small frequencies or variations. This would lead to complexity if added to the model. Finally, **high.wage.ind.log** is a Boolean version of high wage indicator, therefore, there is no need to add this into the model. Using these explanatory variables will be the start of the **Maximal Model**, then applying the `step()` function to breakdown the model step by step until we reach a **minimal adequate model**. The minimal adequate model includes all the necessary or significant relationships of the explanatory variables with the dependent variable. Using `summary()` will show the diagnostic values, which are the **f-value** (from the F-statistics), the **multiple R-squared**, and the Coefficient of **a** and **b**. These diagnostics are used to determine and make conclusions about the output results. Finally, the diagnostic outcome of the `lm()` functions will be plotted in order to assess further the level of goodness of fit of the model. 30

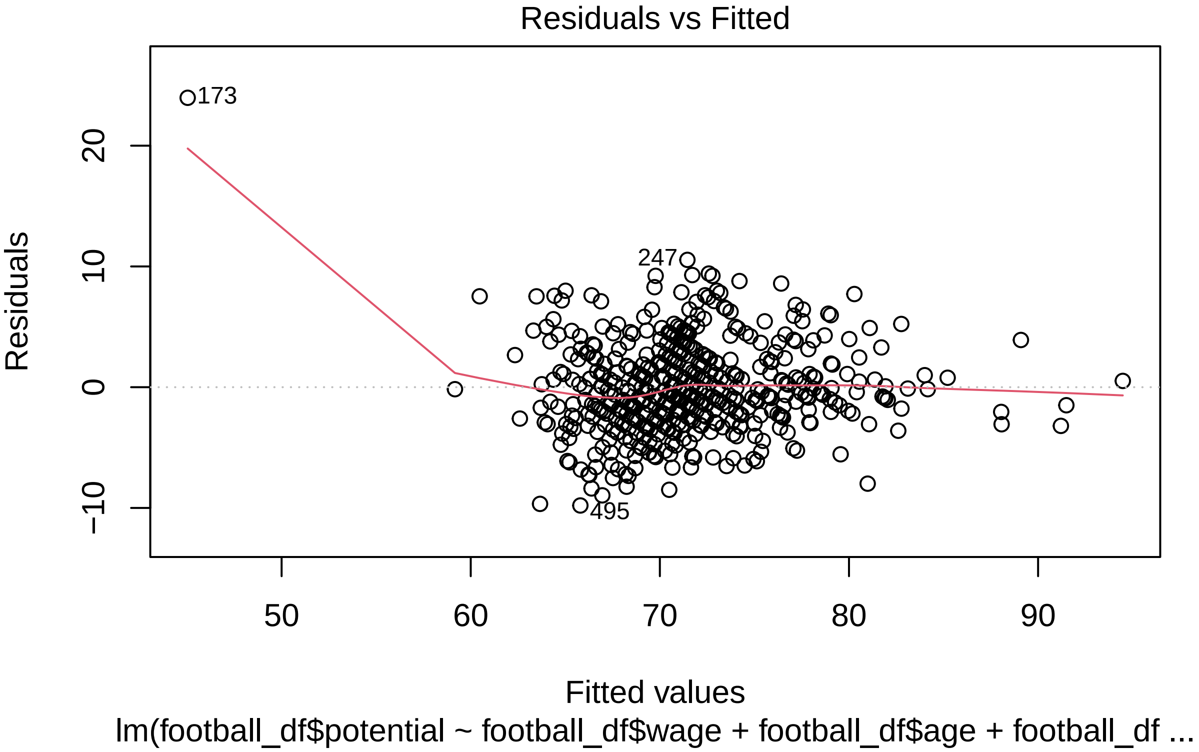
```r
# 1) Creating Maximal Model with the following formula template: `summary(lm(aov(data$DV ~ data$IDV_1 +
potentialModel_Max <- lm(aov(football_df$potential ~ football_df$wage + football_df$age + football_df$he

# 2) Applying the `step()` function to the Maximal Model
potentialModel_Min <- step(potentialModel_Max)
```

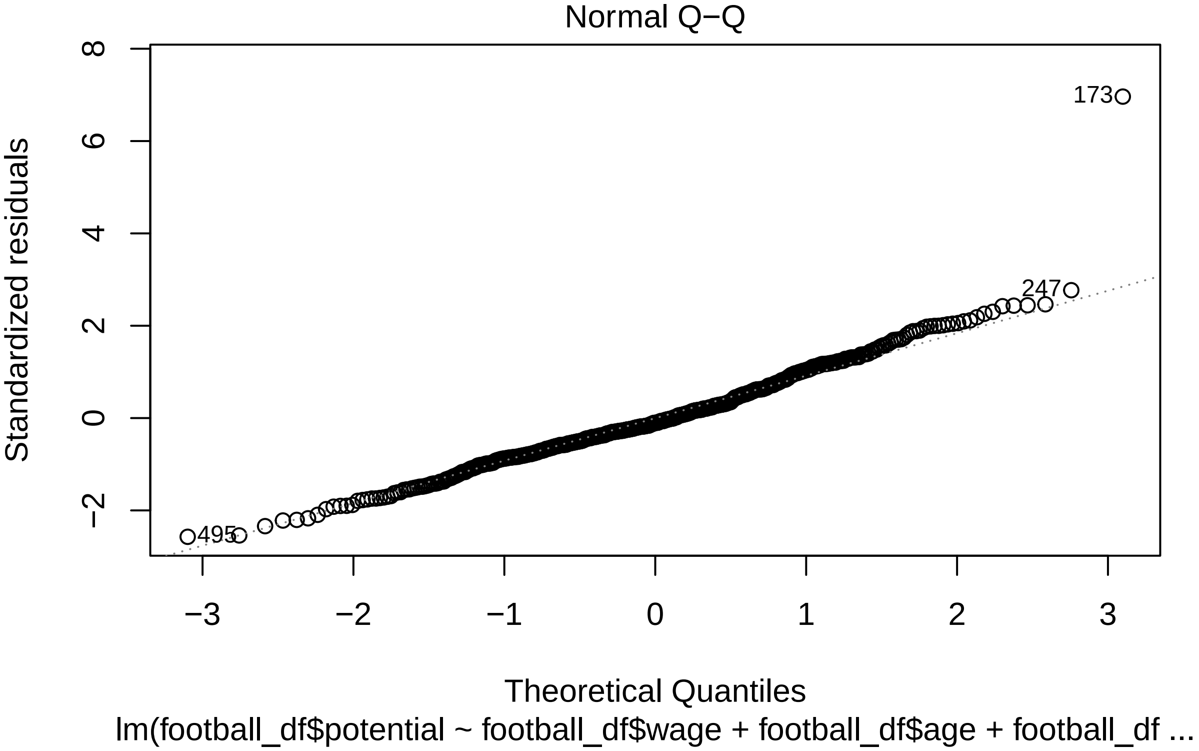
```
## Start:  AIC=1398.36
## football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$‘preferred foot‘ + football_df$pace +
##     football_df$shooting + football_df$passing + football_df$dribbling +
##     football_df$defending + football_df$physic + football_df$‘power strength‘ +
##     football_df$‘power long shots‘ + football_df$high.wage.ind
##
##                                   Df Sum of Sq     RSS    AIC
## - football_df$‘power strength‘    1      0.06  7393.1 1396.4
## - football_df$pace                1     11.05  7404.1 1397.1
## - football_df$passing             1     13.03  7406.1 1397.3
## - football_df$‘preferred foot‘    1     14.01  7407.1 1397.3
## <none>                                         7393.0 1398.4
## - football_df$physic              1     31.38  7424.4 1398.5
## - football_df$height              1     41.90  7434.9 1399.3
## - football_df$defending           1     87.68  7480.7 1402.4
## - football_df$shooting            1     88.41  7481.5 1402.5
## - football_df$‘power long shots‘  1    180.51  7573.6 1408.8
## - football_df$dribbling           1    501.01  7894.1 1430.1
## - football_df$wage                1    570.37  7963.4 1434.6
## - football_df$high.wage.ind       1    572.63  7965.7 1434.7
## - football_df$age                 1   2967.06 10360.1 1569.8
##
## Step:  AIC=1396.36
## football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$‘preferred foot‘ + football_df$pace +
##     football_df$shooting + football_df$passing + football_df$dribbling +
##     football_df$defending + football_df$physic + football_df$‘power long shots‘ +
##     football_df$high.wage.ind
##
##                                   Df Sum of Sq     RSS    AIC
## - football_df$pace                1     11.88  7405.0 1395.2
## - football_df$passing             1     12.98  7406.1 1395.3
## - football_df$‘preferred foot‘    1     14.19  7407.3 1395.3
## <none>                                         7393.1 1396.4
## - football_df$height              1     45.58  7438.7 1397.5
## - football_df$shooting            1     88.36  7481.5 1400.5
## - football_df$defending           1    100.56  7493.7 1401.3
## - football_df$physic              1    146.85  7540.0 1404.5
## - football_df$‘power long shots‘  1    180.47  7573.6 1406.8
## - football_df$dribbling           1    503.95  7897.1 1428.3
## - football_df$high.wage.ind       1    572.58  7965.7 1432.7
## - football_df$wage                1    573.83  7966.9 1432.8
## - football_df$age                 1   2989.80 10382.9 1568.9
##
## Step:  AIC=1395.19
## football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$‘preferred foot‘ + football_df$shooting +
##     football_df$passing + football_df$dribbling + football_df$defending +
##     football_df$physic + football_df$‘power long shots‘ + football_df$high.wage.ind
##
##                                  Df Sum of Sq     RSS    AIC
## - football_df$‘preferred foot‘    1     11.22  7416.2 1394.0
## - football_df$passing             1     17.41  7422.4 1394.4
## <none>                                         7405.0 1395.2
## - football_df$height              1     59.50  7464.5 1397.3
## - football_df$shooting            1     93.53  7498.5 1399.6
## - football_df$defending           1    112.15  7517.1 1400.9
## - football_df$physic              1    136.86  7541.8 1402.6
## - football_df$‘power long shots‘  1    179.06  7584.0 1405.5
## - football_df$dribbling           1    544.29  7949.3 1429.7
## - football_df$wage                1    568.94  7973.9 1431.2
## - football_df$high.wage.ind       1    569.80  7974.8 1431.3
## - football_df$age                 1   3144.89 10549.9 1575.1
##
## Step:  AIC=1393.97
## football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$shooting + football_df$passing +
##     football_df$dribbling + football_df$defending + football_df$physic +
##     football_df$‘power long shots‘ + football_df$high.wage.ind
##
##                                  Df Sum of Sq     RSS    AIC
## - football_df$passing             1      19.9  7436.1 1393.3
## <none>                                         7416.2 1394.0
## - football_df$height              1      60.6  7476.8 1396.2
## - football_df$shooting            1      91.7  7507.9 1398.3
## - football_df$defending           1     111.3  7527.5 1399.6
## - football_df$physic              1     136.3  7552.5 1401.3
## - football_df$‘power long shots‘  1     182.6  7598.8 1404.5
## - football_df$dribbling           1     548.3  7964.5 1428.6
## - football_df$wage                1     564.6  7980.8 1429.7
## - football_df$high.wage.ind       1     569.8  7986.0 1430.0
## - football_df$age                 1    3171.4 10587.6 1575.0
##
## Step:  AIC=1393.34
## football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$shooting + football_df$dribbling +
##     football_df$defending + football_df$physic + football_df$‘power long shots‘ +
##     football_df$high.wage.ind
##
##                                  Df Sum of Sq     RSS    AIC
## <none>                                         7436.1 1393.3
## - football_df$height              1      56.4  7492.5 1395.2
## - football_df$shooting            1      99.6  7535.6 1398.2
## - football_df$physic              1     128.2  7564.2 1400.1
## - football_df$‘power long shots‘  1     168.0  7604.1 1402.8
## - football_df$defending           1     213.0  7649.1 1405.9
## - football_df$wage                1     571.6  8007.7 1429.4
## - football_df$high.wage.ind       1     583.4  8019.5 1430.2
## - football_df$dribbling           1    1046.2  8482.3 1459.0
## - football_df$age                 1    3185.5 10621.5 1574.6
```

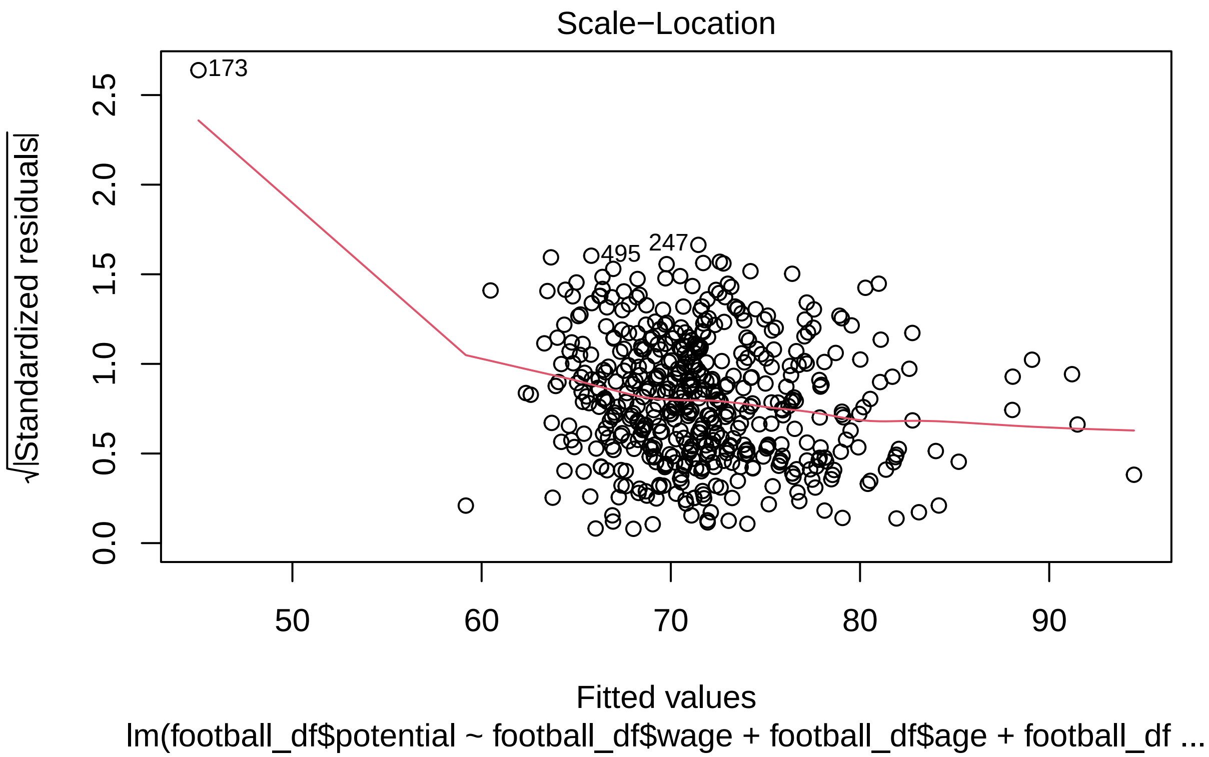
```r
# Lets look at our model using the `summary()`
summary(potentialModel_Min)
```

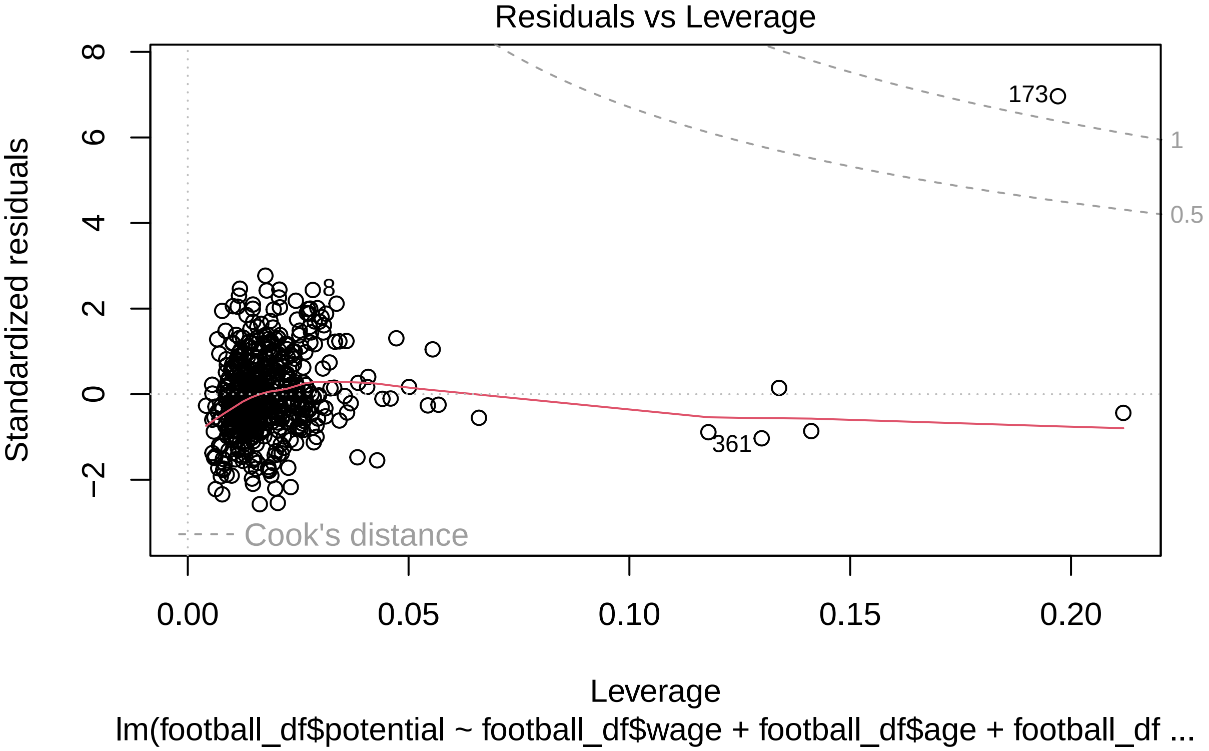
```
##
## Call:
## lm(formula = football_df$potential ~ football_df$wage + football_df$age +
##     football_df$height + football_df$shooting + football_df$dribbling +
##     football_df$defending + football_df$physic + football_df$`power long shots` +
##     football_df$high.wage.ind)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7964 -2.3620 -0.3934  2.3525 23.9670
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.626e+01  6.173e+00   7.494 3.02e-13 ***
## football_df$wage                6.560e-05  1.054e-05   6.224 1.02e-09 ***
## football_df$age                -5.718e-01  3.891e-02 -14.694  < 2e-16 ***
## football_df$height              6.509e-02  3.329e-02   1.955 0.051102 .
## football_df$shooting            1.085e-01  4.176e-02   2.598 0.009659 **
## football_df$dribbling           2.769e-01  3.289e-02   8.421 3.91e-16 ***
## football_df$defending           6.416e-02  1.689e-02   3.800 0.000163 ***
## football_df$physic              7.931e-02  2.691e-02   2.948 0.003352 **
## football_df$`power long shots` -1.021e-01  3.027e-02  -3.374 0.000797 ***
## football_df$high.wage.indTRUE   3.154e+00  5.016e-01   6.288 6.97e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.841 on 504 degrees of freedom
## Multiple R-squared:  0.6268, Adjusted R-squared:  0.6201
## F-statistic: 94.04 on 9 and 504 DF,  p-value: < 2.2e-16
```

```r
# 3) Will now plot the diagnostics
plot(potentialModel_Min)
```


Residuals vs Fitted
Residuals
Fitted values
lm(football_df$potential ~ football_df$wage + football_df$age + football_df ...
173
247
495
20
10
0
−10
50
60
70
80
90


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(football_df$potential ~ football_df$wage + football_df$age + football_df ...
173
247
495
8
6
4
2
0
−2
−3
−2
−1
0
1
2
3





3.2 Critique model using relevant diagnostics

The minimal adequate model has a total of 9 significant relationships between explanatory variables with the potential. Coefficients such as wage and dribbling show significance at 0.001, and height shows some significance at 0.05. The linear equation for this model can be formulated as:

$$y = (4.626e{+}01){+}(3.02e{-}13)\times(wage){+}(6.560e{-}05)\times age{-}(5.718e{-}01)\times height{+}(6.509e{-}02)\times shooting{+}(1.085e{-}01)\times(2.76$$

These significant coefficients (the beta) of the different explanatory variables resulting from the minimal adequate model, imply causal effect on the potential. For example, for every change in the footballer's **'potential'**, there is also an increase in **'dribbling'** and **'shooting'**, by 2.769e-01 and 1.085e-01, respectively, which implies that players show more potential when their attributes in **'dribbling'** and **'shooting'** are greater. Whereas, as the **'age'** increases, the player's **'potential'** decreases by -5.718e-01, indicating that younger footballers have more potential than older one.

The F-statistics is also significant as the f-value is large, and the p-value is significant. The R-squared is also greater then 50%. Therefore, we can conclude that there is significant relationship between football player's potential and the explanatory variables from the minimal model. In other words, these explanatory variables, can explain whether a football player has high potential or not. Furthermore, the diagnostics also indicate that there were not many errors and the regression model fitted the real data quite well (as difference between the variances is significant), with 62% being explained by the variance.

On the contrary however, when plotting the diagnostics to further assess the level of goodness of fit of the model, many issues were raised. The **residual vs the fitted** plot is not random and is all clustered in the centre. The line is also not straight and in line, therefore this contradicts diagnostic values and suggests many inconsistencies with the variances. The **quantile-quantile (Q-Q)** plot, which tests if the errors are not normally distributed, shows a few outliers which support the diagnostic values that there is a lot of error and little accuracy in the estimation of the plot. Both plots highlight that outlier of the data point, 176, seems to be one of the main causes of inconsistencies with the variances. Multicollinearity may also be another reason why goodness of fit of this model is very poor and inconsistent. Multicollinearity occurs when there is there is high correlation between independent variables in regression models. This is problematic as independent variables should be independent, to determine which explanatory variables should fit into the model, and determines which independent variable explains the behaviour of the dependent variable better, in this case, potential 31.

In conclusion, the alternative hypothesis is accepted that there are factors that have an effect or relationship with the player's potential due to the diagnostic values being significant. However, this is a very poor model, and further improvements and other approaches need to be carried out to improve the goodness and accuracy of the fit of the model 30.

3.3 Suggest improvements to your model

One way to improve the model is by identifying the outliers and removing them, as they were the main cause of inconsistency of variances and inaccuracy in the model. Then, a transformation to the model would be applied. This would involve applying and using mathematical expressions such as power or natural log on each data point to transform the variables of the statistical analysis 30, 32. Using transformation will improve the scale to help or ease the interpretation of the plot. Performing these stages would greatly improve the goodness of fit of the model, making the data more accurate and reproducible. Additionally, if multicollinearity was in fact present, then removing the high correlation between the independent variables would improve the accuracy of the data 31 .

4. Extension work

4.1 Model the likelihood of a player having a weekly wage above 8000 Euro (using the high.wage.ind variable provided).

The boxplots and the heatmap correlations provide an essential insight and will help in carrying out and drawing conclusions for the following research question: ***"What factors effect the likelihood of a football***

player earning a weekly wage above 8000 Euro?”

As this research question involves a likelihood, performing Logistic Regression, a type of regression analysis, is the appropriate statistical approach. Logistic Regression will assess and model how the different explanatory variables (both numerical and categorical) will affect the probability of binary outcome within the dependent variable, in this case, the high wage indicator. This binomial dependent variable contains 0 for earning same or less than 8000 Euro, and 1 for earning above 8000 Euro. The Null (H_0) and Alternative Hypothesis (H_1) is stated as below:

H_0 : There are no factors that have an effect on the likelihood of a player earning a weekly wage above 8000 Euro

H_1 : There are factors that have an effect on the likelihood of a player earning a weekly wage above 8000 Euro

The explanatory variable will include all variables except for **club name**, **high.wage.ind.log** (and **wage**) for the same reasons these were not included in the section 3 research question. **Wage** is not included as this would result in multicollinearity due the high relations between **wage** and **high.wage.ind**, and ultimately affect this regression analysis. A maximal model will initially be used and `step()` will be applied to achieve the minimal adequate model of the significance relationships. Instead of using `lm()`, `glm()` will be used, the **Generalised Linear Model**. GLM is a family of models that provides different ways of modelling various dependent variable types. This is the linear equation used for the model:

$$log(\frac{p}{1-p}) = a + b1 \times x1 + b2 \times x2....bk \times xk$$

where $log(\frac{p}{1-p})$ is the logit

In this case, the **Binomial** of the GLM will be used for logistic regression, to implement the logistic transformation (logit) to the linear equation in order to achieve outcome of the equation to be between 0 to 1 (between the 2 outcomes of binary dependent variable). Finally, `exp(coef))` will be used to extract the odd ratio to find and interpret the probability of the effect of the explanatory variables on the binary dependent variable outcome. 30

```
# 1) Creating the maximal model based on the formula: `glm(data$DV ~ data$IDV_1 + data$IDV_2 + data$IDV
wageIndModel_Max <- glm(football_df$`high wage indicator` ~ football_df$potential + football_df$age + f

# 2) Applying the step function to the Maximal Model
wageIndModel_Min <- step(wageIndModel_Max)
```

```
## Start:  AIC=346.44
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$‘preferred foot‘ +
##     football_df$pace + football_df$shooting + football_df$passing +
##     football_df$dribbling + football_df$defending + football_df$physic +
##     football_df$‘power strength‘ + football_df$‘power long shots‘
##
##                                  Df Deviance    AIC
## - football_df$physic              1   318.48 344.48
## - football_df$‘power long shots‘  1   318.53 344.53
## - football_df$‘preferred foot‘    1   318.64 344.64
## - football_df$shooting            1   318.66 344.66
## - football_df$dribbling           1   318.77 344.77
## - football_df$‘power strength‘    1   319.43 345.43
## <none>                                318.44 346.44
## - football_df$weight              1   320.55 346.55
## - football_df$passing            1   320.84 346.84
## - football_df$pace               1   320.88 346.88
## - football_df$height             1   323.44 349.44
## - football_df$defending          1   323.75 349.75
## - football_df$age                1   333.04 359.04
## - football_df$potential          1   388.54 414.54
##
## Step:  AIC=344.48
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$‘preferred foot‘ +
##     football_df$pace + football_df$shooting + football_df$passing +
##     football_df$dribbling + football_df$defending + football_df$‘power strength‘ +
##     football_df$‘power long shots‘
##
##                                  Df Deviance    AIC
## - football_df$‘power long shots‘  1   318.57 342.57
## - football_df$shooting            1   318.67 342.67
## - football_df$‘preferred foot‘    1   318.67 342.67
## - football_df$dribbling           1   318.79 342.79
## <none>                                318.48 344.48
## - football_df$weight              1   320.57 344.57
## - football_df$passing             1   320.92 344.92
## - football_df$pace                1   320.93 344.93
## - football_df$‘power strength‘    1   320.98 344.98
## - football_df$height              1   323.55 347.55
## - football_df$defending           1   325.88 349.88
## - football_df$age                 1   333.07 357.07
## - football_df$potential           1   388.55 412.55
##
## Step:  AIC=342.57
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$‘preferred foot‘ +
##     football_df$pace + football_df$shooting + football_df$passing +
##     football_df$dribbling + football_df$defending + football_df$‘power strength‘
##
##                                Df Deviance    AIC
## - football_df$‘preferred foot‘  1   318.76 340.76
## - football_df$dribbling         1   318.86 340.86
## - football_df$shooting          1   319.63 341.63
## <none>                              318.57 342.57
## - football_df$weight            1   320.72 342.72
## - football_df$pace              1   321.00 343.00
## - football_df$‘power strength‘  1   321.20 343.20
## - football_df$passing           1   321.44 343.44
## - football_df$height            1   323.57 345.57
## - football_df$defending         1   326.45 348.45
## - football_df$age               1   333.35 355.35
## - football_df$potential         1   390.15 412.15
##
## Step:  AIC=340.76
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$pace +
##     football_df$shooting + football_df$passing + football_df$dribbling +
##     football_df$defending + football_df$‘power strength‘
```

```
##
##                                  Df Deviance    AIC
## - football_df$dribbling         1   319.13 339.13
## - football_df$shooting          1   319.83 339.83
## <none>                              318.76 340.76
## - football_df$weight            1   321.03 341.03
## - football_df$pace              1   321.05 341.05
## - football_df$passing           1   321.47 341.47
## - football_df$‘power strength‘  1   321.64 341.64
## - football_df$height            1   323.73 343.73
## - football_df$defending         1   326.64 346.64
## - football_df$age               1   333.62 353.62
## - football_df$potential         1   390.22 410.22
##
## Step:  AIC=339.13
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$pace +
##     football_df$shooting + football_df$passing + football_df$defending +
##     football_df$‘power strength‘
##
##                                  Df Deviance    AIC
## - football_df$shooting          1   321.08 339.08
## <none>                              319.13 339.13
## - football_df$weight            1   321.58 339.58
## - football_df$‘power strength‘  1   321.75 339.75
## - football_df$pace              1   322.97 340.97
## - football_df$height            1   324.07 342.07
## - football_df$passing           1   324.68 342.68
## - football_df$defending         1   326.69 344.69
## - football_df$age               1   334.95 352.95
## - football_df$potential         1   403.89 421.89
##
## Step:  AIC=339.08
## football_df$‘high wage indicator‘ ~ football_df$potential + football_df$age +
##     football_df$height + football_df$weight + football_df$pace +
##     football_df$passing + football_df$defending + football_df$‘power strength‘
##
##                                  Df Deviance    AIC
## <none>                              321.08 339.08
## - football_df$weight            1   323.68 339.68
## - football_df$‘power strength‘  1   324.42 340.42
## - football_df$pace              1   324.97 340.97
## - football_df$height            1   326.37 342.37
## - football_df$defending         1   327.40 343.40
## - football_df$age               1   338.87 354.87
## - football_df$passing           1   340.31 356.31
## - football_df$potential         1   408.46 424.46
```

```
# Lets look at our model using the `summary()`
summary(wageIndModel_Min)
```

```
##
## Call:
## glm(formula = football_df$‘high wage indicator‘ ~ football_df$potential +
##     football_df$age + football_df$height + football_df$weight +
##     football_df$pace + football_df$passing + football_df$defending +
##     football_df$‘power strength‘, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8735  -0.4595  -0.1859   0.2548   3.2750
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -47.91449    6.51358  -7.356 1.89e-13 ***
## football_df$potential         0.30237    0.03852   7.851 4.14e-15 ***
## football_df$age               0.15795    0.04365   3.618 0.000297 ***
## football_df$height            0.07832    0.03194   2.452 0.014196 *
## football_df$weight           -0.06061    0.03772  -1.607 0.108130
## football_df$pace              0.02907    0.01498   1.941 0.052286 .
## football_df$passing           0.09384    0.02249   4.172 3.02e-05 ***
## football_df$defending         0.02490    0.01008   2.471 0.013478 *
## football_df$‘power strength‘   0.03409    0.01889   1.804 0.071158 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 622.45  on 513  degrees of freedom
## Residual deviance: 321.08  on 505  degrees of freedom
## AIC: 339.08
##
## Number of Fisher Scoring iterations: 6
```

The exponential of the coefficients will now be extracted to obtain the odd ratios

```
exp(coef(wageIndModel_Min))
```

```
##                  (Intercept)        football_df$potential
##                 1.552393e-21                 1.353065e+00
##              football_df$age           football_df$height
##                 1.171108e+00                 1.081468e+00
##           football_df$weight             football_df$pace
##                 9.411934e-01                 1.029502e+00
##          football_df$passing        football_df$defending
##                 1.098389e+00                 1.025216e+00
## football_df$‘power strength‘
##                 1.034680e+00
```

Performing logistic regression for this research question has resulted in 8 significant relationships between the explanatory variables with the high wage indicator. Explanatory variables such as **‘potential’** and **‘passing’** show significance at around 0, and explanatory variables such as pace are significant at 0.05. Also, the odd ratio of these significant coefficients are all greater than 1, indicating that these explanatory variables affect the likelihood of a football player earning a weekly wage above 8000 Euro. For example, for a unit increase in **‘passing’**, the odds of a football player earning more than 8000 Euro weekly increases by a factor of 1.098389e+00. Therefore, the Null Hypothesis is rejected.

Furthermore, **Deviance** and **Akaike's Information Criteria (AIC)** are two measurements of `step()` function. They both measure the model fit. Deviance assesses the variances, the lower the deviance, the greater the variances, and consequently, the better the model. AIC is a single value which indicates how well the model fits the data, given the size of the dataset. The lower the AIC score, the better the fit of the model. Both Deviance and AIC have decreased greatly from the initial maximal model, therefore, this suggests that the minimal adequate model is a good fit of the data 30.

References

[1] Shepperd, M. (2021) 5.2 The central role of data quality | CS5702 Modern Data Book. Available at: https://bookdown.org/martin_shepperd/ModernDataBook/C5_CentralRole.html (Accessed: 12 January 2022).

[2] Area, W.N.C.-N. (2022) Data Quality. Available at: https://www.northampton.gov.uk/info/100004/your-council/1115/data-quality (Accessed: 12 January 2022).

[3] OMNISCI (2021) What is Data Quality? Available at: https://www.omnisci.com/technical-glossary/data-quality (Accessed: 12 January 2022).

[4] Lean-Data (2021) Data Quality – Lean-Data. Available at: https://www.lean-data.nl/tag/data-quality/ (Accessed: 12 January 2022).

[5] dplyr (2022) A Grammar of Data Manipulation • dplyr. Available at: https://dplyr.tidyverse.org/ (Accessed: 12 January 2022).

[6] ggplot2 (2022) Create Elegant Data Visualisations Using the Grammar of Graphics. Available at: https://ggplot2.tidyverse.org/ (Accessed: 12 January 2022).

[7] Schork, J. (2022) R Plot Only One Variable in ggplot2 Plot (2 Examples) | Draw Scatterplot. Available at: https://statisticsglobe.com/plot-only-one-variable-in-ggplot2-plot-r (Accessed: 12 January 2022).

[8] sape (2017) ggplot2 Quick Reference: colour (and fill) | Software and Programmer Efficiency Research Group. Available at: http://sape.inf.usi.ch/quick-reference/ggplot2/colour (Accessed: 12 January 2022).

[9] datalab.cc (2020) Using Colors in R > datalab.cc, datalab.cc. Available at: https://datalab.cc/rcolors (Accessed: 12 January 2022).

[10] Shepperd, M. (2021) Chapter 5 Data Quality, Cleaning and Imputation | CS5702 Modern Data Book. Available at: https://bookdown.org/martin_shepperd/ModernDataBook/Chap5DataCleaning.html (Accessed: 12 January 2022). [11] Tableau (2022) Guide To Data Cleaning: Definition, Benefits, Components, And How To Clean Your Data, Tableau. Available at: https://www.tableau.com/learn/articles/what-is-data-cleaning (Accessed: 12 January 2022).

[12]Data Clarity (2020) 'Data Cleansing Guide: What is Data Cleansing and Why is it Important?', Data Clarity, 24 June. Available at: https://www.dataclarity.uk.com/2020/06/24/data-cleansing-guide-what-is-data-cleansing-and-why-is-it-important/ (Accessed: 12 January 2022).

[13] Quora (2017) What is the salary of the average football (soccer) player in different leagues and divisions in Europe?, Quora. Available at: https://www.quora.com/What-is-the-salary-of-the-average-football-soccer-player-in-different-leagues-and-divisions-in-Europe (Accessed: 12 January 2022).

[14] singh, R. (2020) 'It's all about Outliers', Analytics Vidhya, 31 August. Available at: https://medium.com/analytics-vidhya/its-all-about-outliers-cbe172aa1309 (Accessed: 12 January 2022).

[15] Auguie, B. and Antonov, A. (2017) gridExtra: Miscellaneous Functions for 'Grid' Graphics. Available at: https://CRAN.R-project.org/package=gridExtra (Accessed: 12 January 2022).

[16] Anderson, R.D.P., Sean Kross, and Brooke (2022) 4.5 The grid Package | Mastering Software Development in R. Available at: https://github.com/rdpeng/RProgDA (Accessed: 12 January 2022).

[17] Data Cornering (2020) Convert R TRUE and FALSE values to 1 and 0, and vice versa - Data Cornering. Available at: https://datacornering.com/convert-r-true-and-false-values-to-1-and-0-and-vice-versa/ (Accessed: 12 January 2022).

[18] DataScience Made Simple (2021) 'Rename the column name in R using Dplyr', DataScience Made Simple. Available at: https://www.datasciencemadesimple.com/rename-the-column-name-in-r-using-dplyr/ (Accessed: 12 January 2022).

[19] Restori, M. (2021) What is Exploratory Data Analysis, Chartio. Available at: https://chartio.com/learn/data-analytics/what-is-exploratory-data-analysis/ (Accessed: 12 January 2022).

[20] Patil, P. (2018) What is Exploratory Data Analysis? Available at: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15 (Accessed: 12 January 2022).

[21] StatisticsHowTo (2022) Data Analysis & Exploratory Data Analysis (EDA), Statistics How To. Available at: https://www.statisticshowto.com/probability-and-statistics/data-analysis/ (Accessed: 12 January 2022).

[22] Cookbook for R (2012) Plotting distributions (ggplot2). Available at: http://www.cookbook-r.com/Graphs/Plotting_distributions_(ggplot2)/ (Accessed: 12 January 2022).

[23] Holtz, Y. (2018) The R Graph Gallery – Help and inspiration for R charts, The R Graph Gallery. Available at: https://www.r-graph-gallery.com/ (Accessed: 12 January 2022).

[24] Technik, D. (2019) 'Shapiro-Wilk Test for Normality in R | R-bloggers', 8 August. Available at: https://www.r-bloggers.com/2019/08/shapiro-wilk-test-for-normality-in-r/ (Accessed: 12 January 2022).

[25] Holtz, Y. (2018) Control ggplot2 boxplot colors. Available at: https://www.r-graph-gallery.com/264-control-ggplot2-boxplot-colors.html (Accessed: 12 January 2022).

[26] Alboukadel (2019) 'How to Change GGPlot Labels: Title, Axis and Legend: Title, Axis and Legend', Datanovia, 12 January. Available at: https://www.datanovia.com/en/blog/how-to-change-ggplot-labels/ (Accessed: 12 January 2022).

[27] STHDA (2022) ggplot2: Quick correlation matrix heatmap - R software and data visualization - Easy Guides - Wiki - STHDA. Available at: http://www.sthda.com/english/wiki/ggplot2-quick-correlation-matrix-heatmap-r-software-and-data-visualization (Accessed: 12 January 2022).

[28] DataCamp (2022) lower.tri function - RDocumentation. Available at: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/lower.tri (Accessed: 13 January 2022).

[29] DataCamp (2022) melt function - RDocumentation. Available at: https://www.rdocumentation.org/packages/reshape2/versions/1.4.4/topics/melt (Accessed: 13 January 2022).

[30] Crawley, M.J. (2014) Statistics: an introduction using R. Second;2nd; Chichester, West Sussex, UK: John Wiley & Sons, Inc (Book, Whole). Available at: https://go.exlibris.link/8bxSrVDg.

[31] Frost, J. (2017) 'Multicollinearity in Regression Analysis: Problems, Detection, and Solutions', Statistics By Jim, 2 April. Available at: http://statisticsbyjim.com/regression/multicollinearity-in-regression-analysis/ (Accessed: 13 January 2022).

[32] Stephanie (2015) Tukey Ladder of Powers / Power Ladder: Definition, Statistics How To. Available at: https://www.statisticshowto.com/tukey-ladder-of-powers/ (Accessed: 13 January 2022).