Applied Statistic and Data Visualization

Assignment

By

Syed Ali Murtaza

1

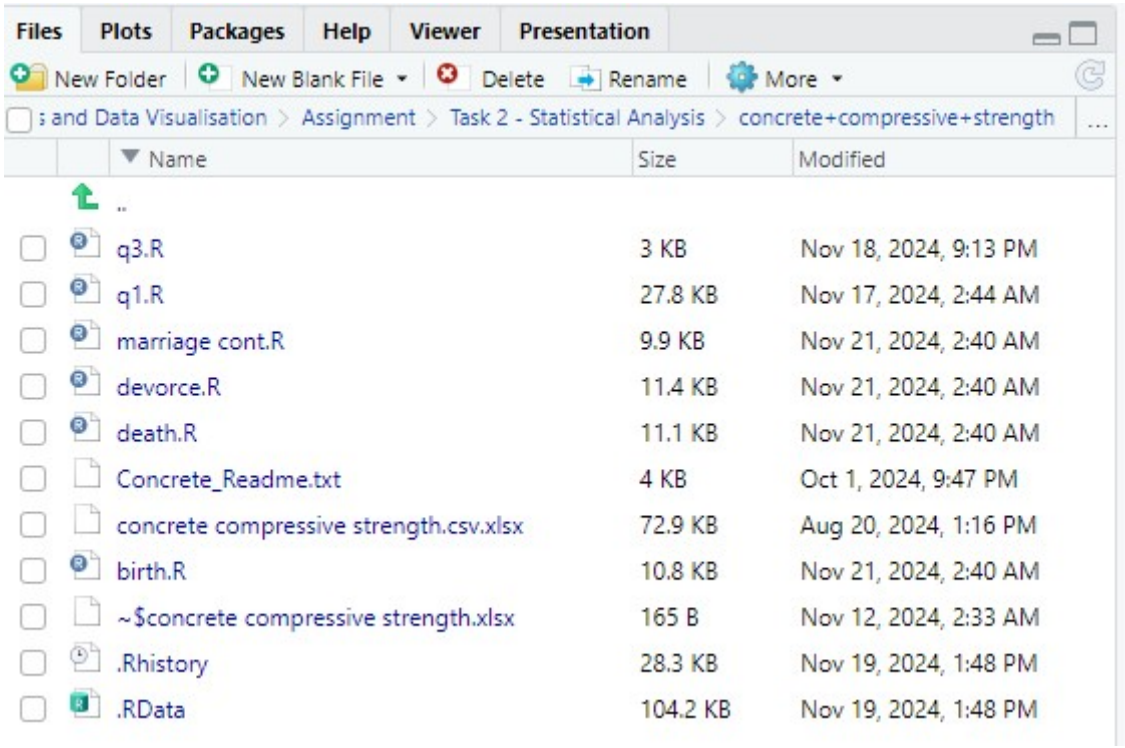
Part 2

Your analysis should:

- **Include initial Exploratory Data Analysis (EDA), for example, calculation of relevant descriptive statistics and visualisations to help you explore and understand the data.**
- **Conduct appropriate correlation analysis for the variables and evaluate the results in the context of your chosen scenario.**
- **Formulate regression problem(s) relating to your chosen scenario and application of appropriate regression techniques on the dataset.**
- **Formulate hypotheses relating to your chosen scenario and use appropriate tests to test them.**

For this part of the assignment, I am going to use the Concrete Compressive Strength dataset.

First of all I opened the R studio and then in the output panel I navigate the directory to the folder where I saved my datafiles and also set it as a working directory.



Exploring data graphically:-

For exploring the data first of all I load all the necessary libraries as

```r

install.packages("datarium")
install.packages("tidyverse")
install.packages("corrplot")
install.packages("rcompanion")
library(datarium)
library(tidyverse)
library(corrplot)
library(rcompanion)
library(readxl)
library(ggplot2)
library(dplyr)
library(corrplot)

```

Then I load my dataset into R as follow

```r
df <- read_excel("C:/Users/Chaudhary Computer/Desktop/Applied Statistics and Data Visual
# Using relative path:

```

When I data is loaded into R I visualize the data as

```r

head(df)
tail(df)
summary(df)
names(df)
str(df)

```

Now I am going to visualize the data because visualization is very important at the beginning of analysis of every data to understand the pattern and to suggest possible modeling strategies.

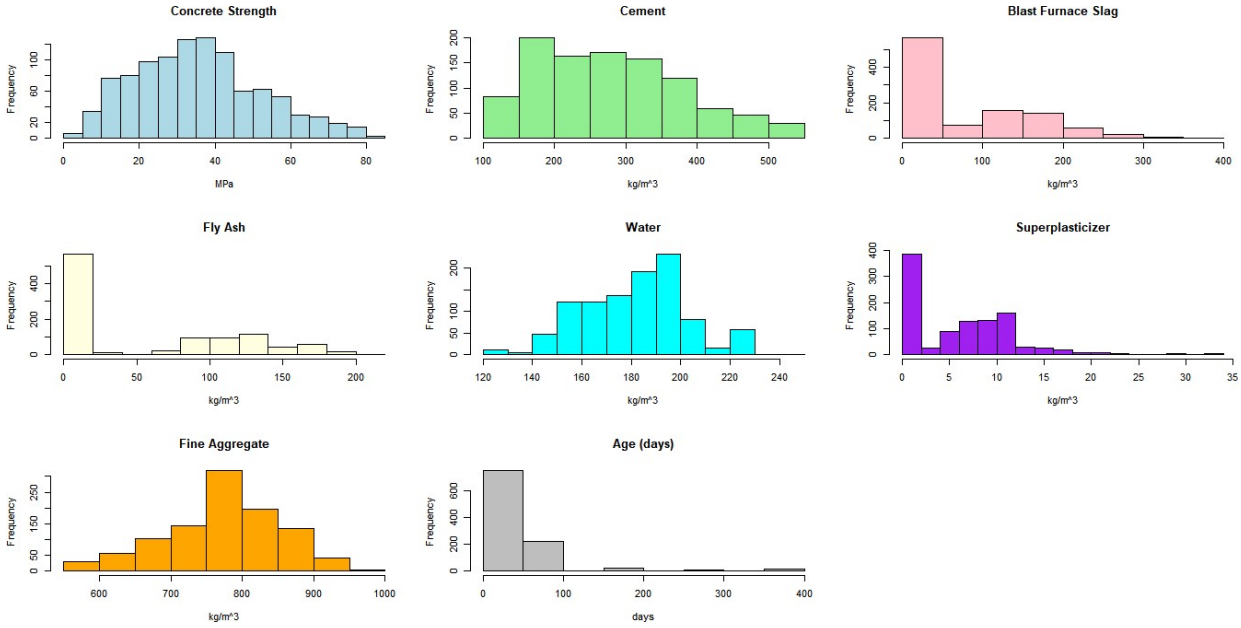
So I will visualize the data using different plot types.

Histograms:-

I visualize the histograms for all the columns in the dataset which are numeric to see the full distribution of the dataset.

```r

# Adjust layout to fit multiple histograms
par(mfrow = c(3, 3))  # 3 rows, 3 columns

# Plot histograms for each column
hist(df$`Concrete compressive strength(MPa, megapascals)`, main = "Concrete Strength", col = "lightblue", xlab = "MPa")
hist(df$`Cement (component 1)(kg in a m^3 mixture)`, main = "Cement", col = "lightgreen", xlab = "kg/m^3")
hist(df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`, main = "Blast Furnace Slag", col = "pink", xlab = "kg/m^3")
hist(df$`Fly Ash (component 3)(kg in a m^3 mixture)`, main = "Fly Ash", col = "lightyellow", xlab = "kg/m^3")
hist(df$`Water  (component 4)(kg in a m^3 mixture)`, main = "Water", col = "cyan", xlab = "kg/m^3")
hist(df$`Superplasticizer (component 5)(kg in a m^3 mixture)`, main = "Superplasticizer", col = "purple", xlab = "kg/m^3")
hist(df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`, main = "Fine Aggregate", col = "orange", xlab = "kg/m^3")
hist(df$`Age (day)`, main = "Age (days)", col = "grey", xlab = "days")

# Reset layout to default
par(mfrow = c(1, 1))

```

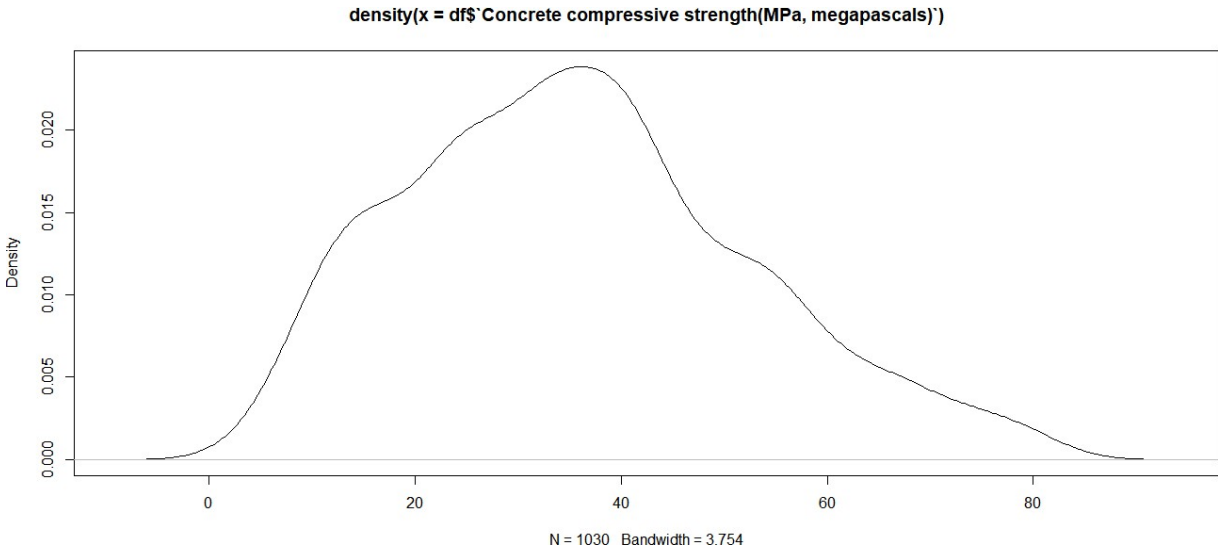
Output:-



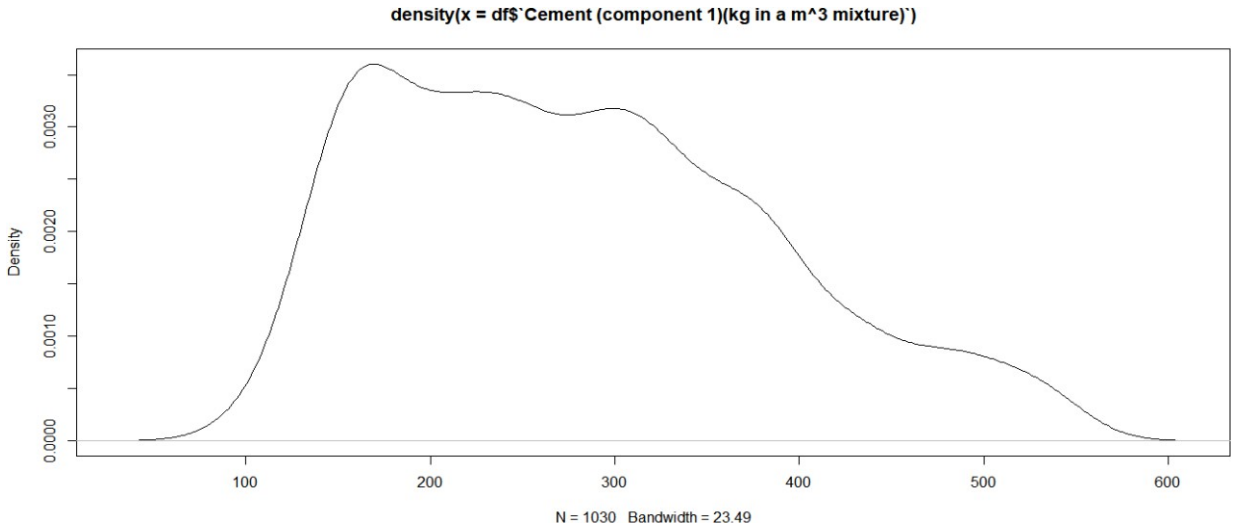
Density graphs:-

Density graphs are used to show distribution of data across the mean value which can be used to see that the data is normal or not. So I am going to built density graph for each column of my data set as follow.

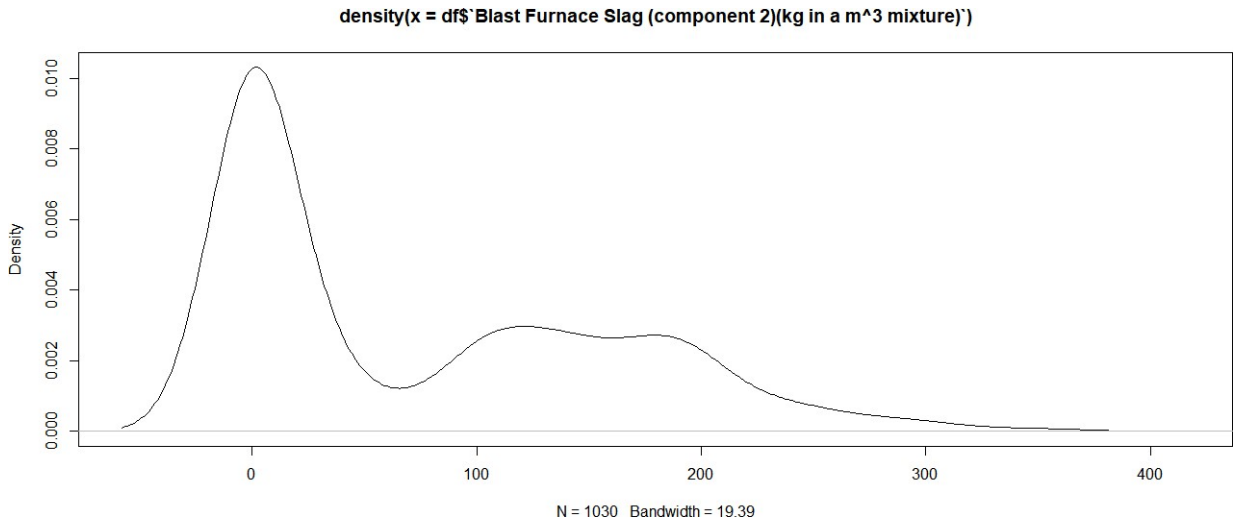
```
46 #Concrete compressive strength(MPa, megapascals
47 plot(density(df$`Concrete compressive strength(MPa, megapascals)`))
48
```



```r
#Cement (component 1)(kg in a m^3 mixture)
plot(density(df$`Cement (component 1)(kg in a m^3 mixture)`))
```

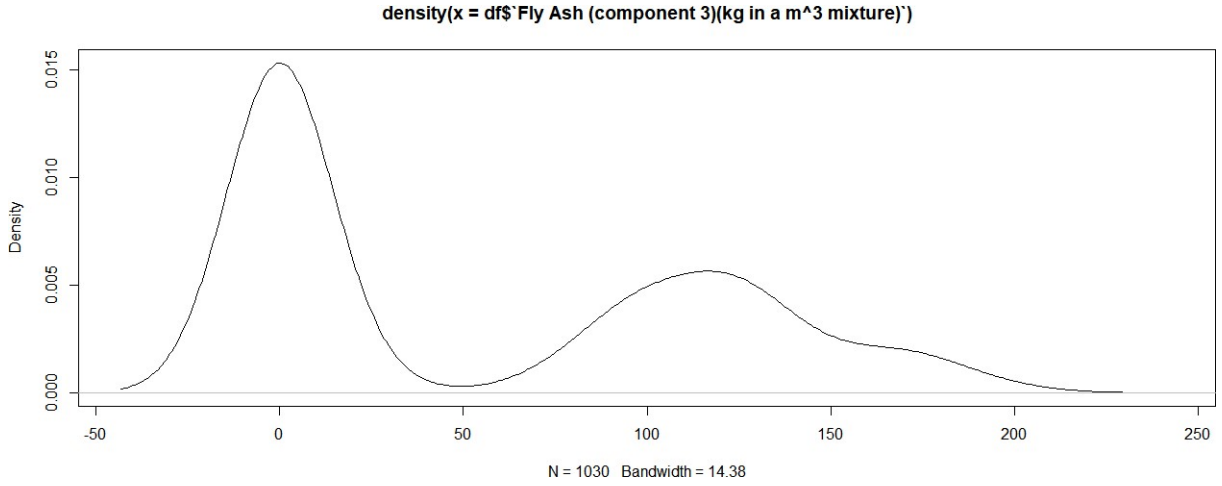



```r
#Blast Furnace Slag (component 2)(kg in a m^3 mixture)
plot(density(df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`))
```

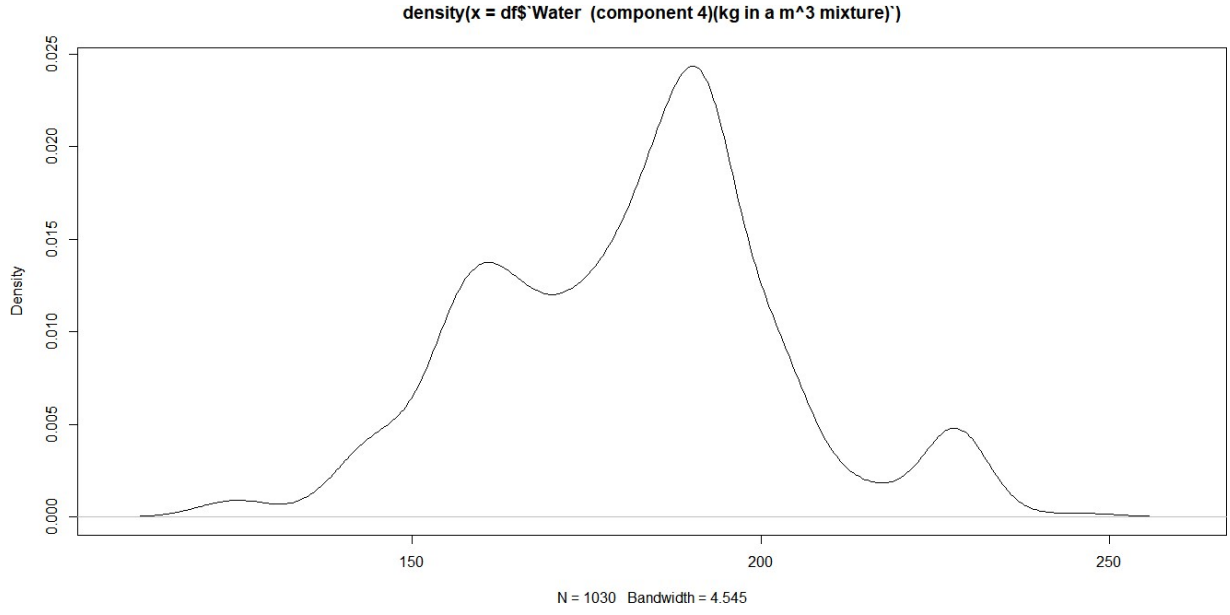



```r
#Fly Ash (component 3)(kg in a m^3 mixture)
plot(density(df$`Fly Ash (component 3)(kg in a m^3 mixture)`))
```

1

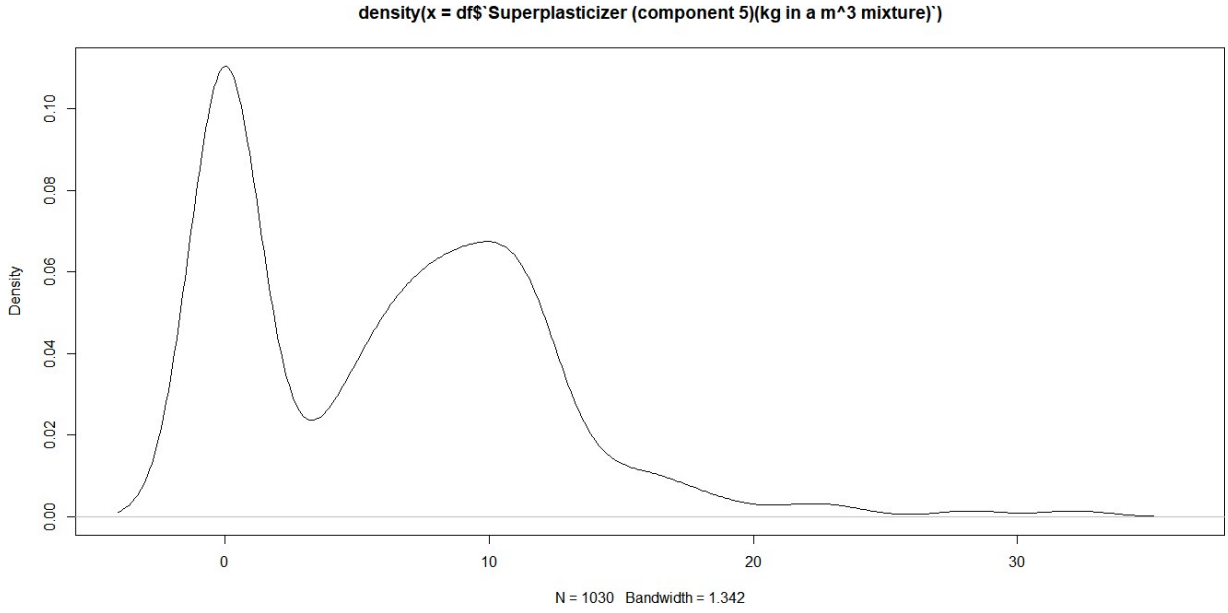



```
57
58
59  #water  (component 4)(kg in a m^3 mixture)
60  plot(density(df$`water  (component 4)(kg in a m^3 mixture)`))
61
62
```

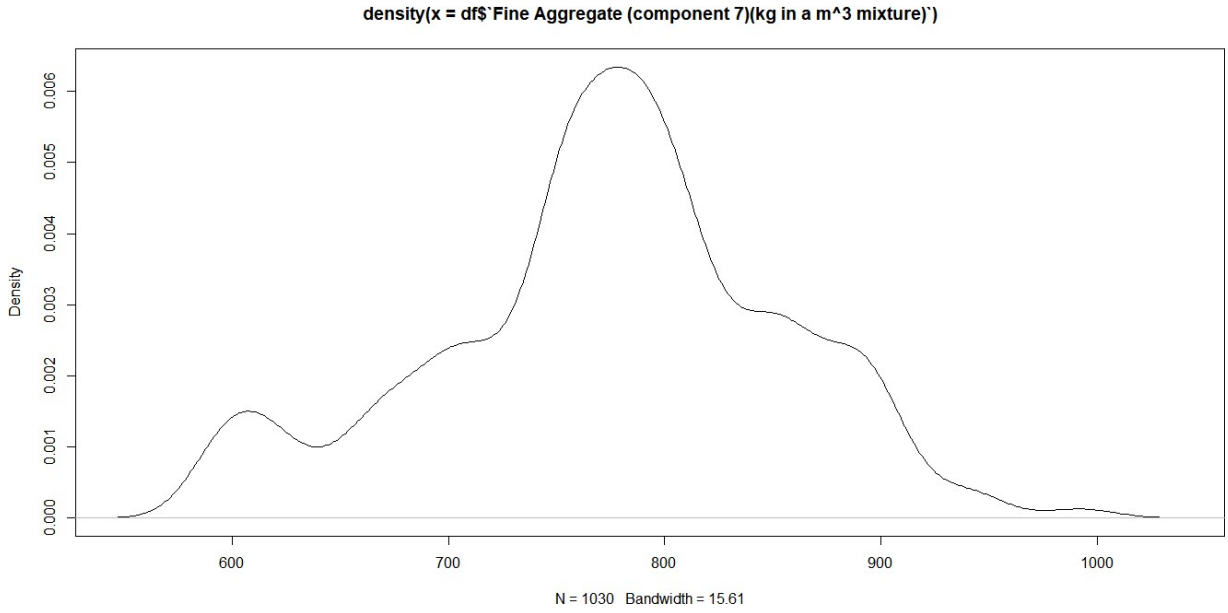



```
61
62
63  #Superplasticizer (component 5)(kg in a m^3 mixture)
64  plot(density(df$`Superplasticizer (component 5)(kg in a m^3 mixture)`))
65
66
```

1

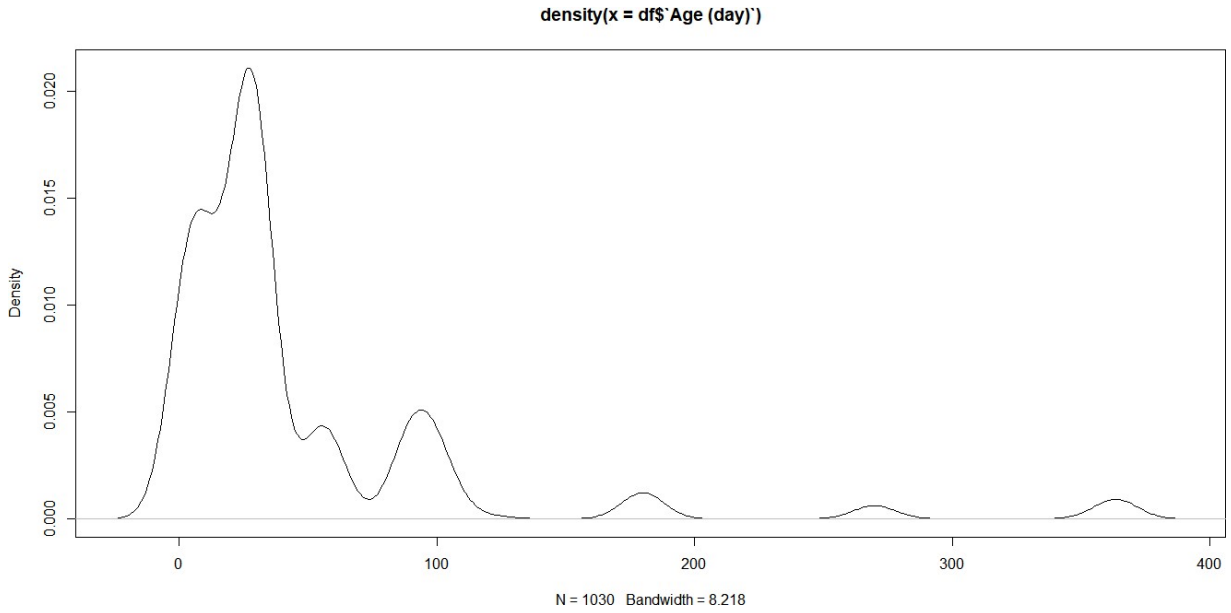



```
65
66
67  #Fine Aggregate (component 7)(kg in a m^3 mixture)
68  plot(density(df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`))
69
70
```



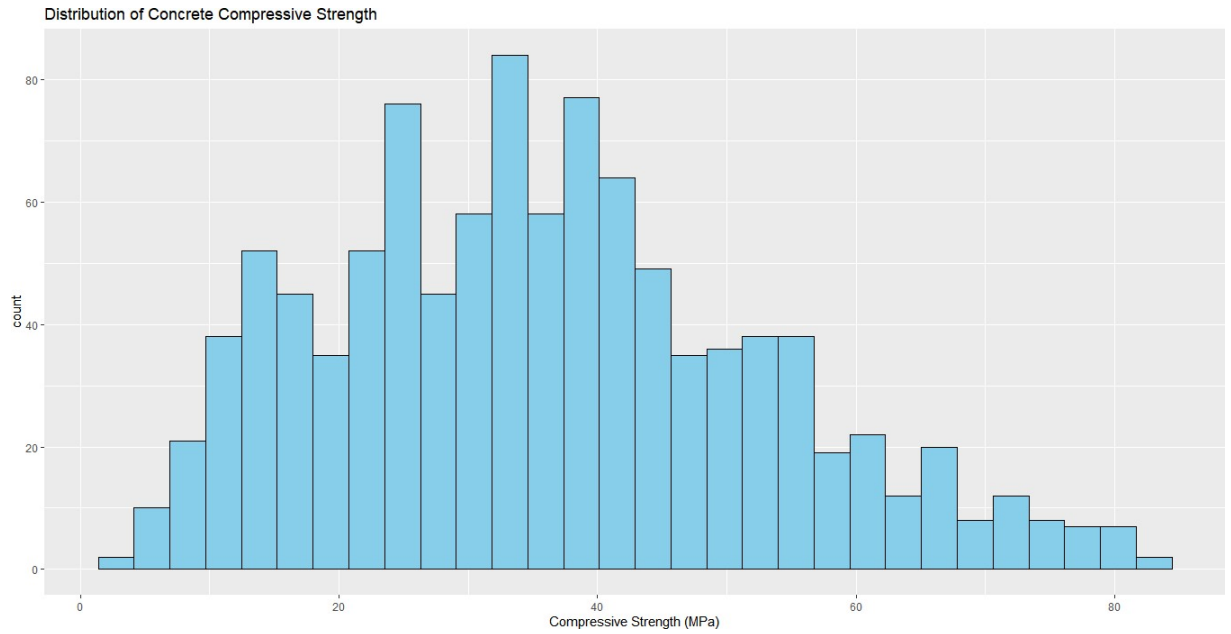

```
69
70  #Age (day)
71  plot(density(df$`Age (day)`))
72
```

1



Distribution of target variable:-

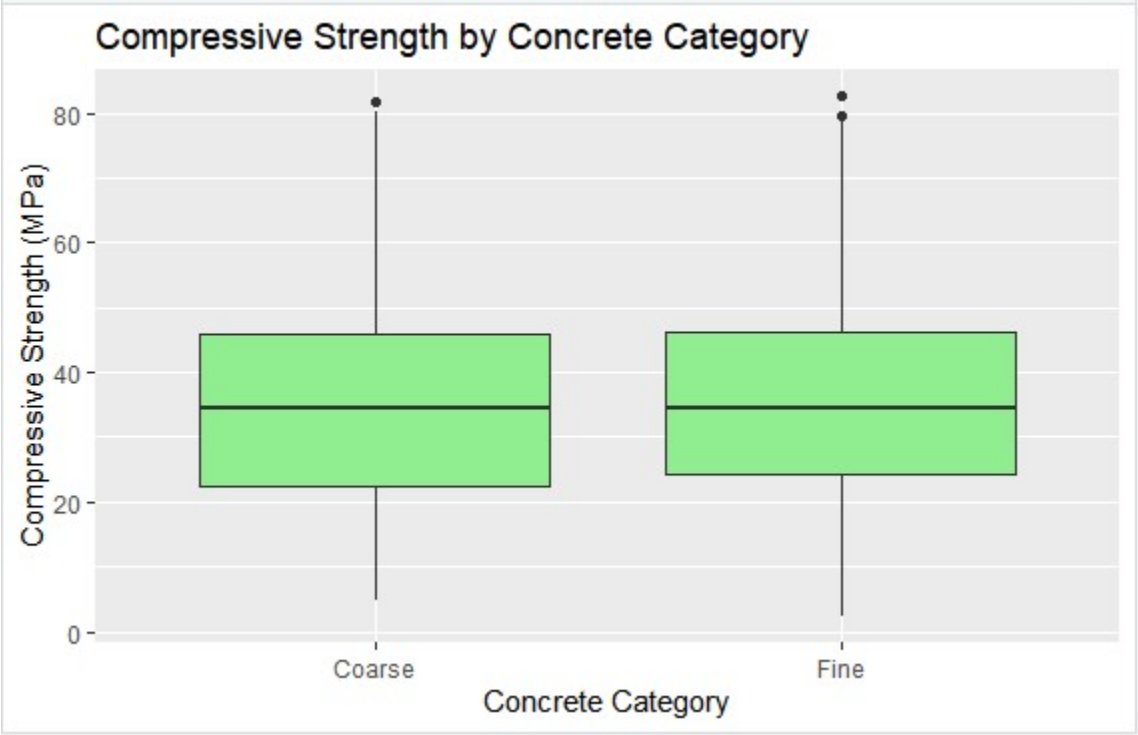
```
75
76
77  #Distribution of Concrete Compressive Strength
78  ggplot(df, aes(x = `Concrete compressive strength(MPa, megapascals)`)) +
79    geom_histogram(bins = 30, fill = "skyblue", color = "black") +
80    labs(title = "Distribution of Concrete Compressive Strength", x = "Compressive Strength (MPa)")
81
82
```



Boxplot:-

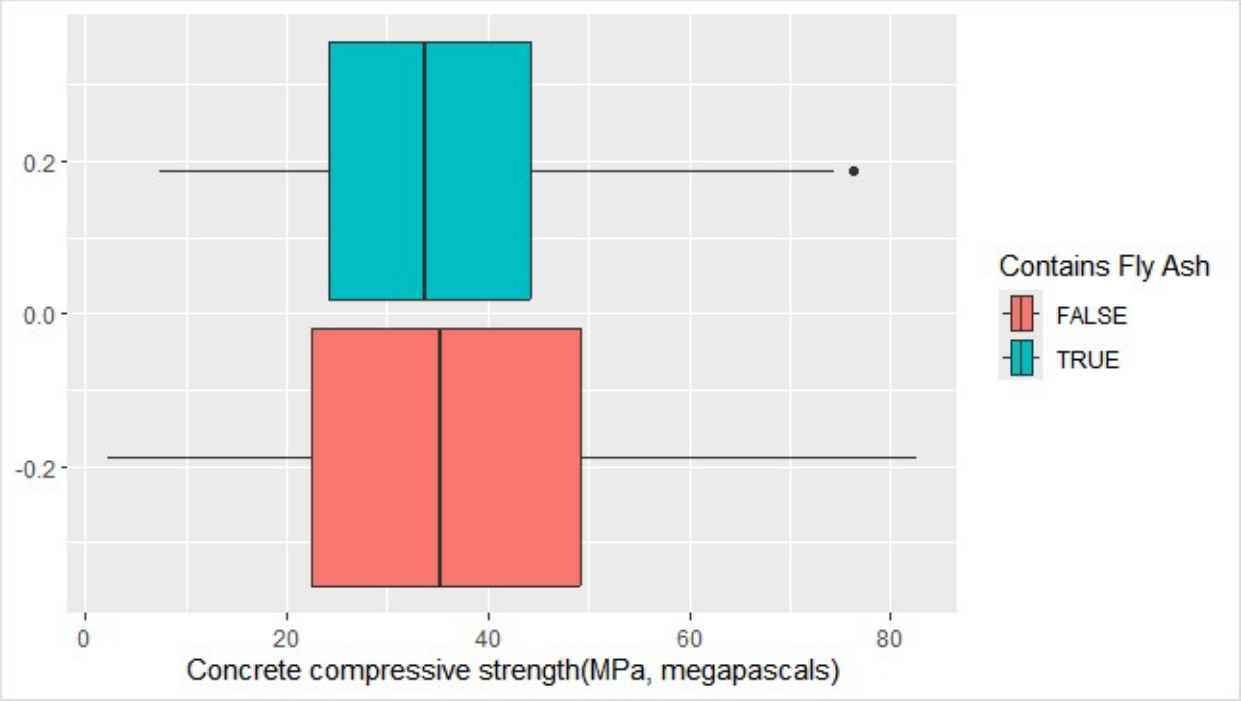
Now I will see the box plot of compressive strength by concrete category to see how the data is distributed by this category and also to have a look on outliers as well.

```
82
83  # Boxplot of Strength by Concrete Category
84  ggplot(df, aes(x = `Concrete Category`, y = `Concrete compressive strength(MPa, megapascals)`)) +
85    geom_boxplot(fill = "lightgreen") +
86    labs(title = "Compressive Strength by Concrete Category", x = "Concrete Category", y = "Compressive Strength (MPa)")
87
88
```



Now I will plot the box plot between compressive concrete strength and the categorical variable which is Contain Fly Ash to see the distribution of data.

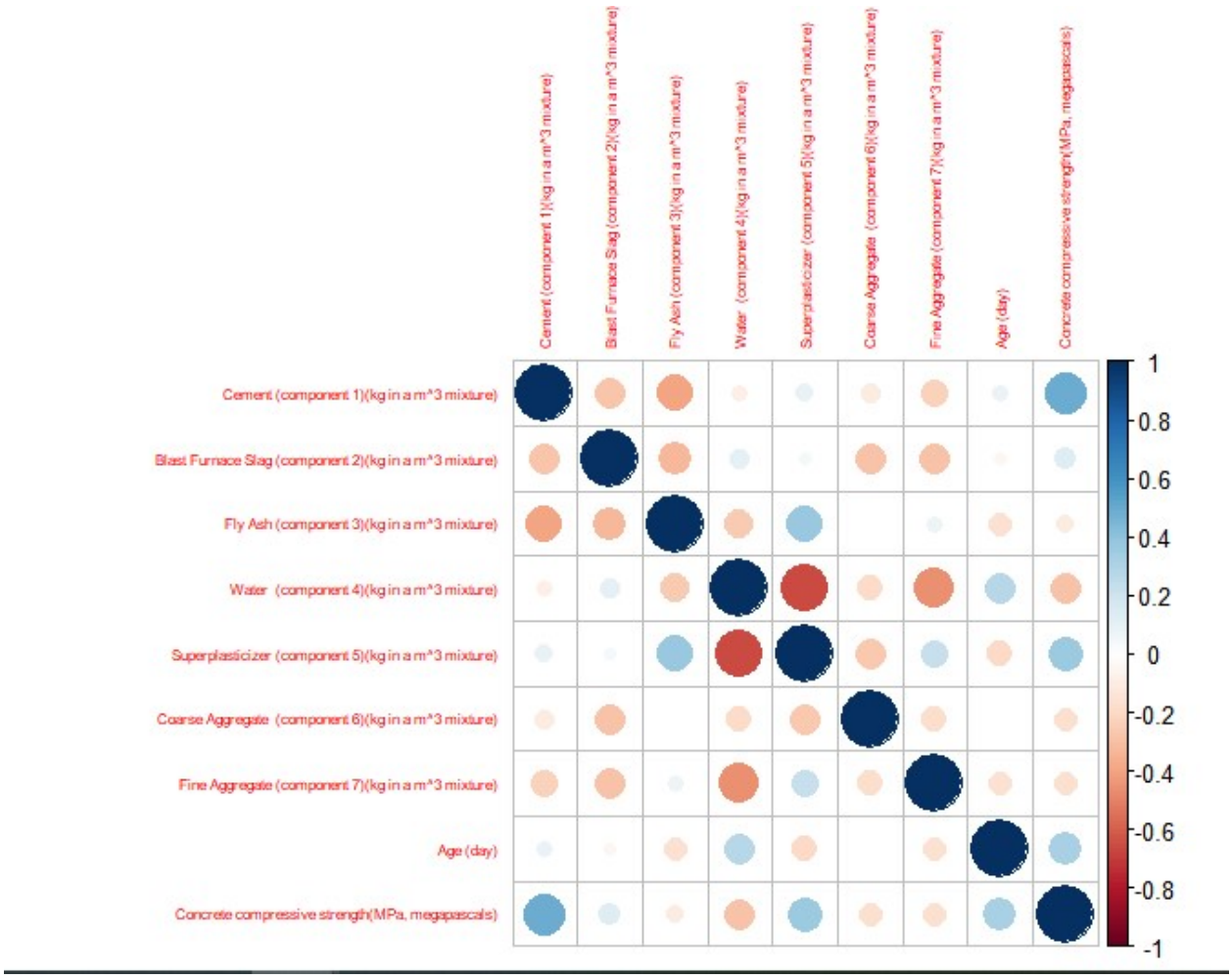
```
88
89  #box plot  of Strength by Contains Fly Ash
90  ggplot(df, aes(x = `Concrete compressive strength(MPa, megapascals)`, fill = `Contains Fly Ash`)) +
91    geom_boxplot()
92
93
94
```



Correlation Matrix for Numeric Variable:-

Now I will plot the Correlation Matrix for Numeric Variable to see how the numerical variables are realted to each other

```
89
90  # Correlation Matrix for Numeric Variables
91  corr_matrix <- cor(numeric_cols, use = "complete.obs")
92  corrplot(corr_matrix, method = "circle", tl.cex = 0.5)  # Adjust value as needed
93
```

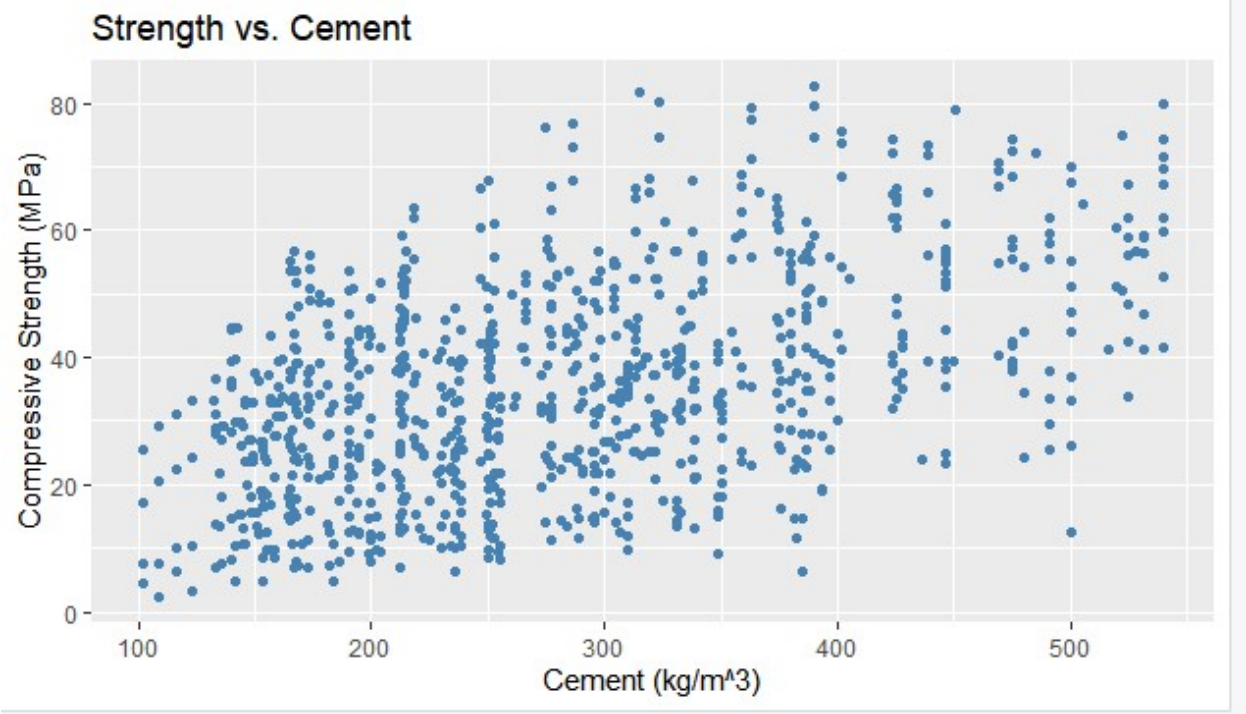


Now from this correlation matrix we can see that cement has the highest correlation with the compressive concrete strength.

Scatter plot:-

Now I will plot the scatter plot of cement and compressive concrete strength to check is there any linearity or not?

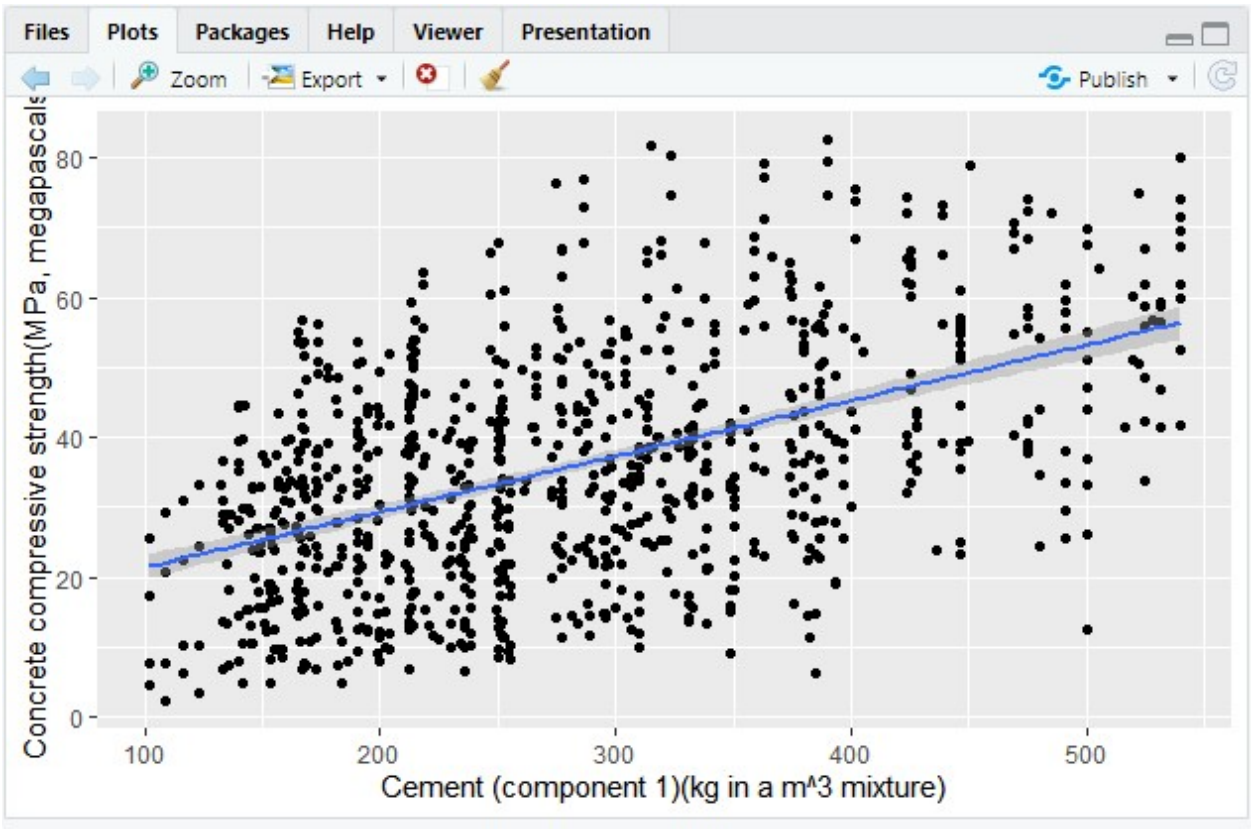
```
94
95  # Scatter plot of strength vs. cement
96  ggplot(df, aes(x = `Cement (component 1)(kg in a m^3 mixture)`, y = `Concrete compressive strength(MPa, megapascals)`)) +
97    geom_point(color = "steelblue") +
98    labs(title = "Strength vs. Cement", x = "Cement (kg/m^3)", y = "Compressive Strength (MPa)")
99
100
```



From this graph we can see that cement is partially linear with compressive concrete strength.

So now I will built a linear line between them

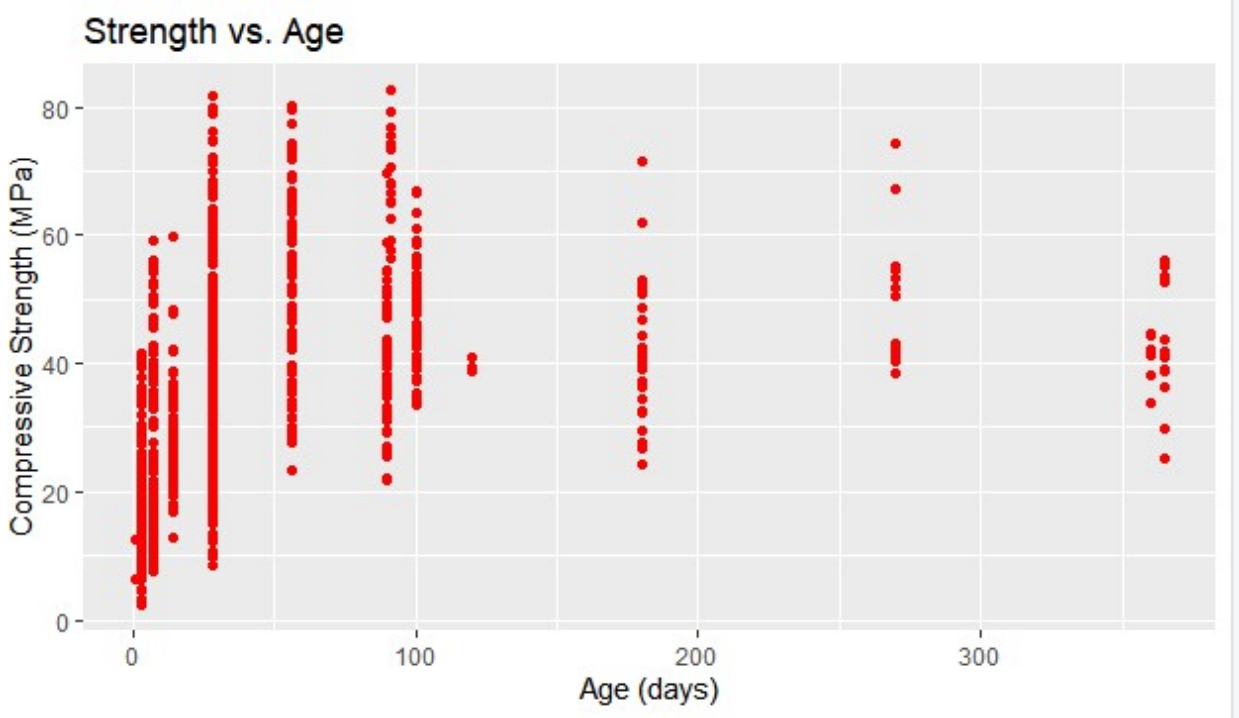
1



Now I will see the linearity between age and compressive strength

```
101
102 # Scatter plot of strength vs. age
103 ggplot(df, aes(x = `Age (day)`, y = `Concrete compressive strength(MPa, megapascals)`)) +
104   geom_point(color = "red") +
105   labs(title = "Strength vs. Age", x = "Age (days)", y = "Compressive Strength (MPa)")
106
107
108
```

1



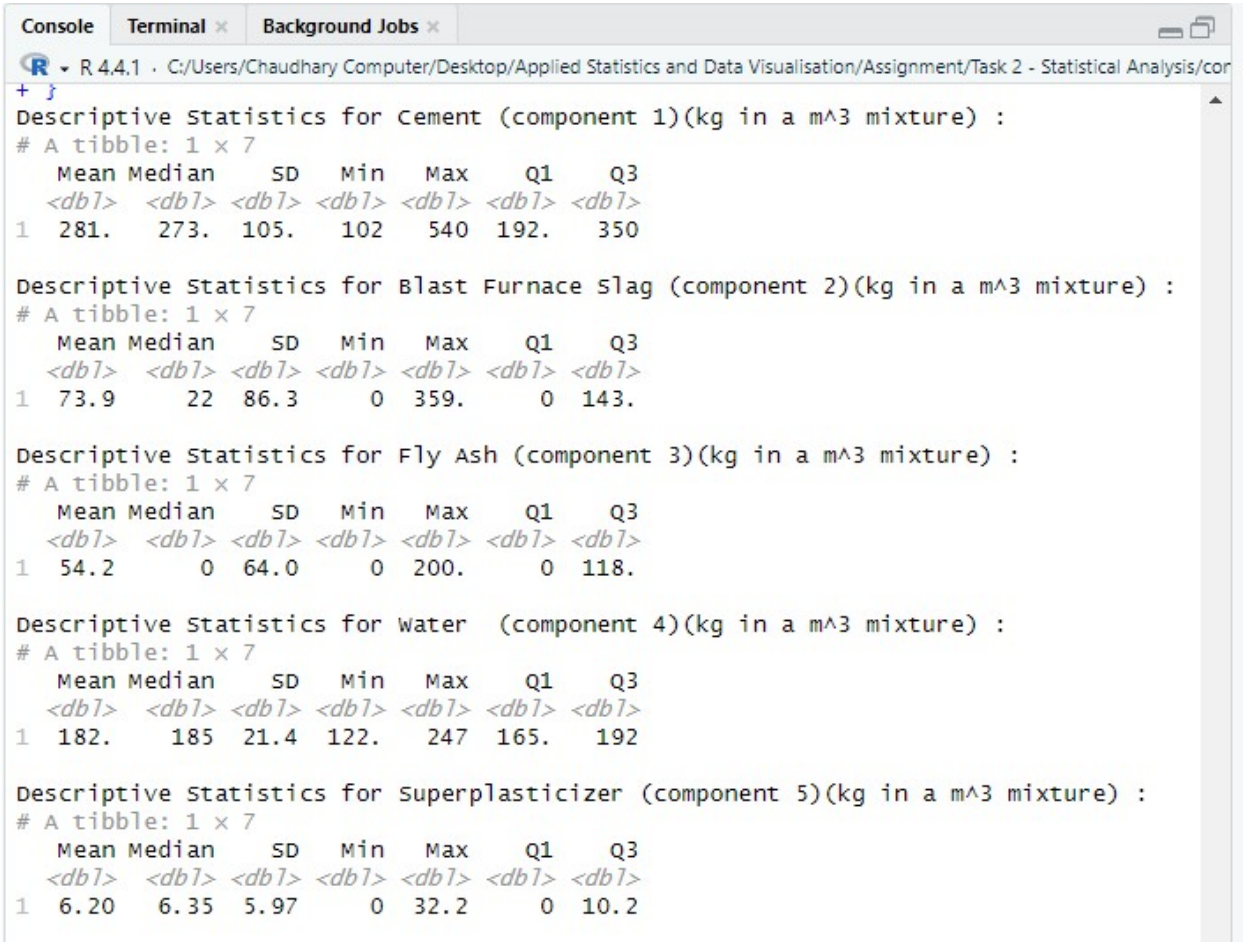
Descriptive analysis:-

Now I write a code that will do descriptive analyses of only numeric column of dataset. So that we have more clear picture of data.

```
126
127 # Calculate and display descriptive statistics
128 for (col_name in names(df)) {
129   if (is.numeric(df[[col_name]])) {
130     cat("Descriptive Statistics for", col_name, ":\n")
131
132     # Calculate statistics
133     stats <- df %>%
134       summarise(
135         Mean = mean(.data[[col_name]], na.rm = TRUE),
136         Median = median(.data[[col_name]], na.rm = TRUE),
137         SD = sd(.data[[col_name]], na.rm = TRUE),
138         Min = min(.data[[col_name]], na.rm = TRUE),
139         Max = max(.data[[col_name]], na.rm = TRUE),
140         Q1 = quantile(.data[[col_name]], 0.25, na.rm = TRUE),
141         Q3 = quantile(.data[[col_name]], 0.75, na.rm = TRUE)
142       )
143
144     # Print results
145     print(stats)
146     cat("\n")
147   }
148 }
149
```

1

Output:-


Console
Terminal
Background Jobs
R 4.4.1 · C:/Users/Chaudhary Computer/Desktop/Applied Statistics and Data Visualisation/Assignment/Task 2 - Statistical Analysis/cor
+ }
Descriptive Statistics for Cement (component 1)(kg in a m^3 mixture) :
A tibble: 1 × 7
Mean Median SD Min Max Q1 Q3
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 281. 273. 105. 102 540 192. 350
Descriptive Statistics for Blast Furnace Slag (component 2)(kg in a m^3 mixture) :
A tibble: 1 × 7
Mean Median SD Min Max Q1 Q3
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 73.9 22 86.3 0 359. 0 143.
Descriptive Statistics for Fly Ash (component 3)(kg in a m^3 mixture) :
A tibble: 1 × 7
Mean Median SD Min Max Q1 Q3
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 54.2 0 64.0 0 200. 0 118.
Descriptive Statistics for Water (component 4)(kg in a m^3 mixture) :
A tibble: 1 × 7
Mean Median SD Min Max Q1 Q3
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 182. 185 21.4 122. 247 165. 192
Descriptive Statistics for Superplasticizer (component 5)(kg in a m^3 mixture) :
A tibble: 1 × 7
Mean Median SD Min Max Q1 Q3
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 6.20 6.35 5.97 0 32.2 0 10.2

```
Descriptive Statistics for Coarse Aggregate  (component 6)(kg in a m^3 mixture) :
# A tibble: 1 x 7
   Mean Median    SD   Min   Max    Q1    Q3
  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  973.    968  77.8   801  1145   932 1029.

Descriptive Statistics for Fine Aggregate (component 7)(kg in a m^3 mixture) :
# A tibble: 1 x 7
   Mean Median    SD   Min   Max    Q1    Q3
  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  774.   780.  80.2   594  993.  731.   824

Descriptive Statistics for Age (day) :
# A tibble: 1 x 7
   Mean Median    SD   Min   Max    Q1    Q3
  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  45.7     28  63.2     1   365     7    56

Descriptive Statistics for Concrete compressive strength(MPa, megapascals) :
# A tibble: 1 x 7
   Mean Median    SD   Min   Max    Q1    Q3
  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  35.8   34.4  16.7  2.33  82.6  23.7  46.1

> |
```

Conduct appropriate correlation analysis for the variables and evaluate the results in the context of your chosen scenario.

Correlation Between Two Continuous Variables:-

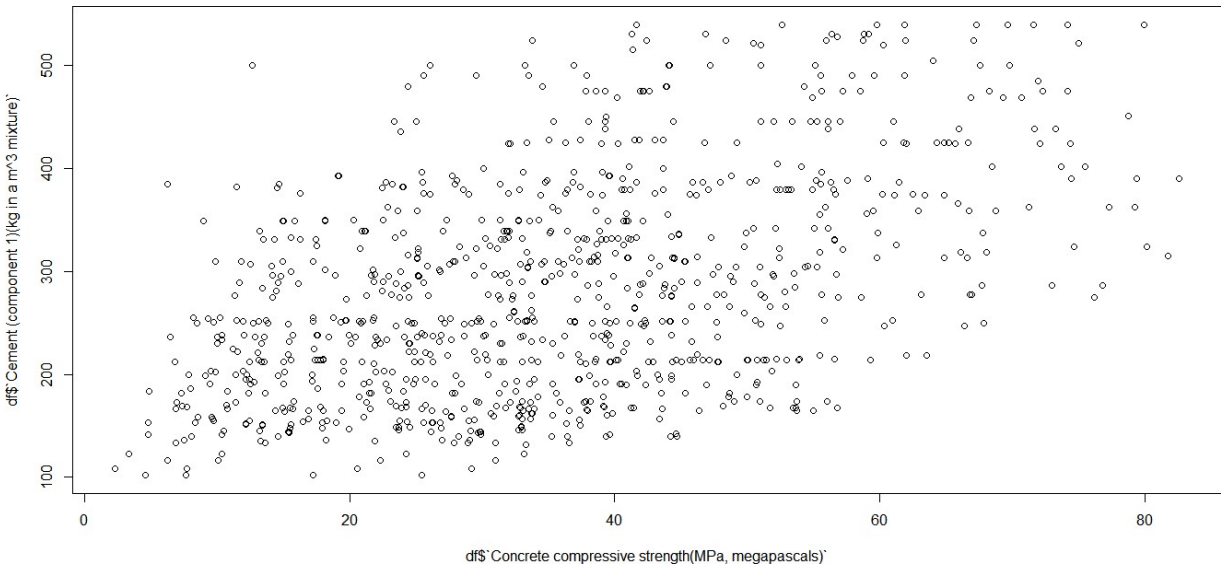
Now I will conduct correlation analysis between two continouse variable and also determine their value of correlance. Here I assumed that if the correlation between two continouse variable is linear and both of them are normally distributed as well than I will apply Pearson correlation. If the correlation between 2 variable is non-linear and the normality is not assured than I will apply Spearman correlation.

Correlation b/w compressive concrete strength and cement:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
168
169 names(df)
170 df
171 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Cement (component 1)(kg in a m^3 mixture)`)
172
```



From this visualization we can see that there is a linear relationship but this is not a perfect linear but we assume it is linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
172
173 # Shapiro-Wilk Test
174 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
175 shapiro_test2 <- shapiro.test(df$`Cement (component 1)(kg in a m^3 mixture)`)
176
177 # Output Results
178 print(shapiro_test1)
179 print(shapiro_test2)
180
```

```
        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Cement (component 1)(kg in a m^3 mixture)`
W = 0.95896, p-value < 2.2e-16
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

According to assumption now I will apply spearman correlation analysis on them yto find the correlation.

```
180
181
182 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Cement (component 1)(kg in a m^3 mixture)`,method="spearman")
183
184
```

```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Cement (component 1)(kg in a m^3 mixture)`,method="spearman")
[1] 0.4776012
>
```

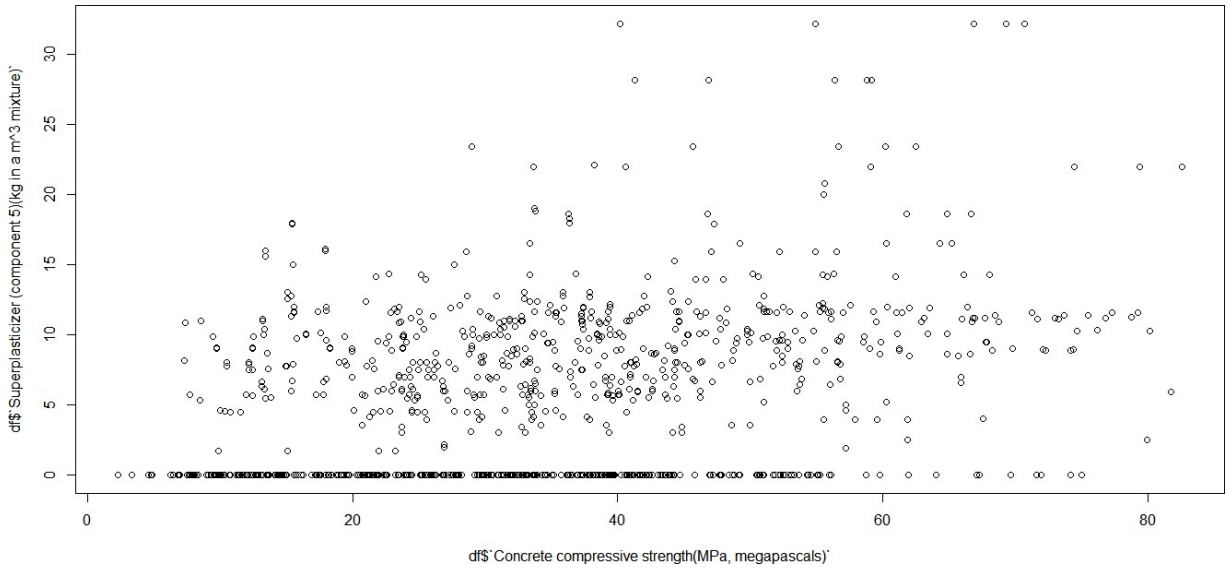
From result we can conclude that there is a positive correlative between them.

Correlation b/w compressive concrete strength and Superplasticizer:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
188
189 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Superplasticizer (component 5)(kg in a m^3 mixture)`)
190
```



From this visualization we can see that there is not linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
191
192 # Shapiro-Wilk Test
193 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
194 shapiro_test2 <- shapiro.test(df$`Superplasticizer (component 5)(kg in a m^3 mixture)`)
195
196 # Output Results
197 print(shapiro_test1)
198 print(shapiro_test2)
199
```

```
> print(shapiro_test1)

        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Superplasticizer (component 5)(kg in a m^3 mixture)`
W = 0.86605, p-value < 2.2e-16
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

According to assumption now I will apply spearman correlation analysis on them to find the correlation.

```
200
201 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Superplasticizer (component 5)(kg in a m^3 mixture)`,method="spearman")
202
203
204
```

```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Superplasticizer (component 5)(kg in a m^3 mixture)`,method="spearman")
[1] 0.3475889
>
```

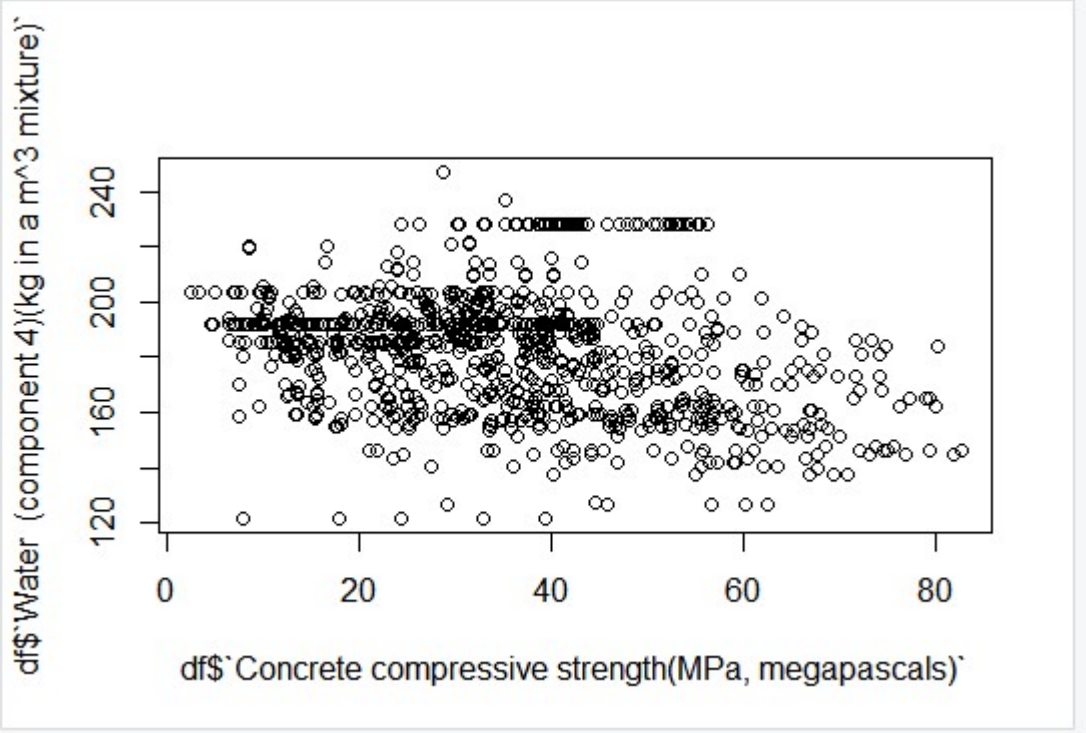
Since there is a positive correlative between them.

Correlation b/w compressive concrete strength and Water:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
204
205 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Water  (component 4)(kg in a m^3 mixture)`)
```



From this visualization we can see that there is not linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
206 # Shapiro-Wilk Test
207 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
208 shapiro_test2 <- shapiro.test(df$`Water  (component 4)(kg in a m^3 mixture)`)
209
```

```
>
> # Output Results
> print(shapiro_test1)

        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Water  (component 4)(kg in a m^3 mixture)`
W = 0.9804, p-value = 1.473e-10
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

1

According to assumption now I will apply spearman correlation analysis on them to find the correlation.

```
213
214 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Water  (component 4)(kg in a m^3 mixture)`,method="spearman")
215
216
```

```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Water  (component 4)(kg in a m^3 mixture)`,method="spearman")
[1] -0.3083707
>
```

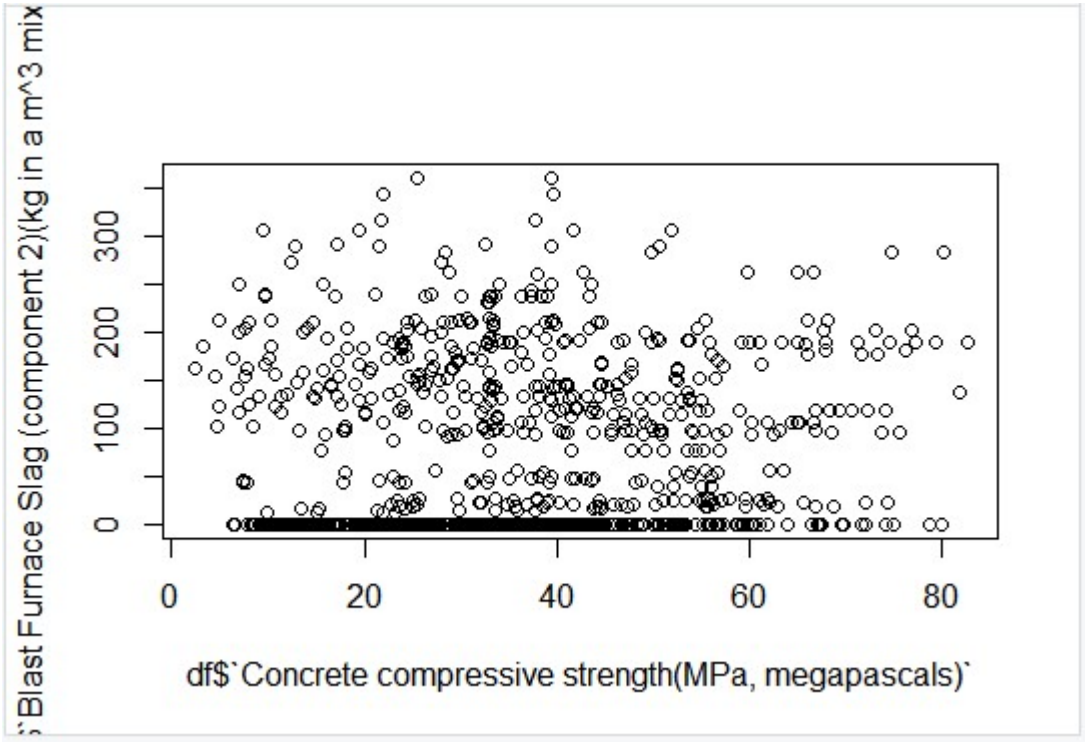
Since there is a negative correlative between them.

Correlation b/w compressive concrete strength and Blast Furnace Slag:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
217
218 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`)
219
```



From this visualization we can see that there is not linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
220 # Shapiro-Wilk Test
221 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
222 shapiro_test2 <- shapiro.test(df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`)
223
224 # Output Results
225 print(shapiro_test1)
226 print(shapiro_test2)
227
```

```
> # Output Results
> print(shapiro_test1)

        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`
W = 0.81241, p-value < 2.2e-16
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

According to assumption now I will apply spearman correlation analysis on them to find the correlation.

```
227
228
229 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,method="spearman")
230
```

```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,method="spearman")
[1] 0.1624734
>
```

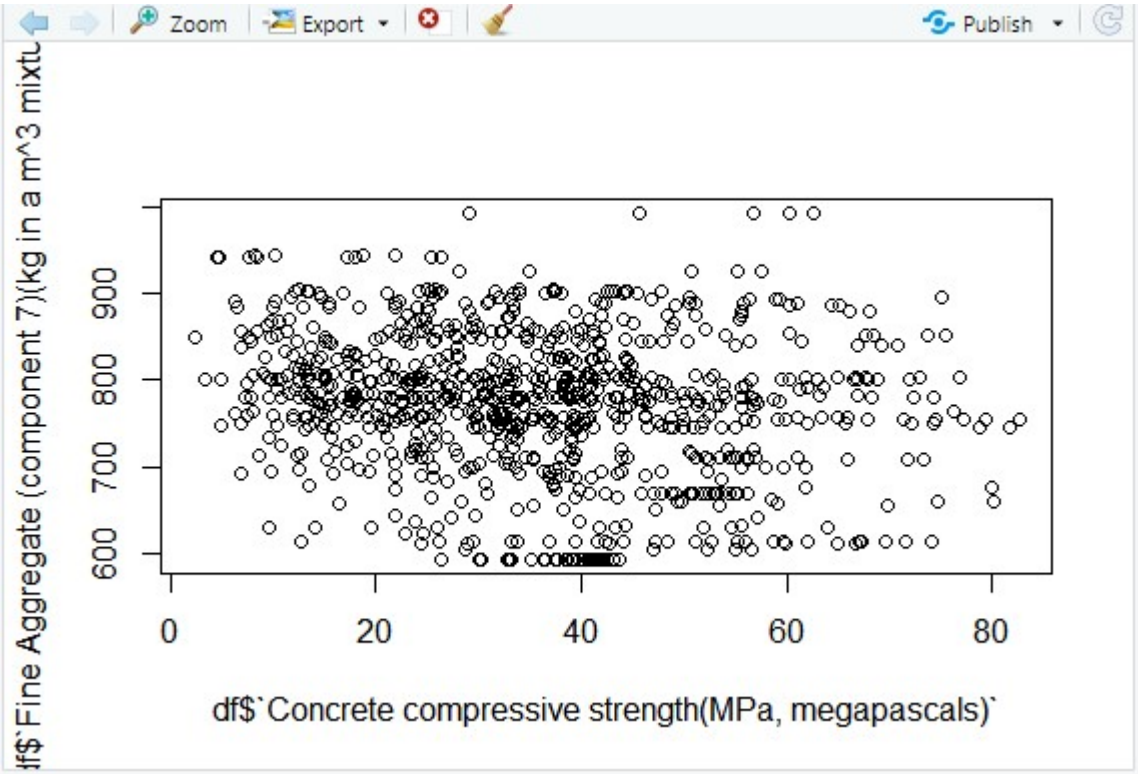
Since there is a positive correlative between them.

Correlation b/w compressive concrete strength and Fine Aggregate:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
232
233 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`)
234
```



From this visualization we can see that there is not linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
234
235 # Shapiro-Wilk Test
236 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
237 shapiro_test2 <- shapiro.test(df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`)
238
239 # Output Results
240 print(shapiro_test1)
241 print(shapiro_test2)
242
```

1

```
> # Output Results
> print(shapiro_test1)

        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`
W = 0.98067, p-value = 1.843e-10
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

According to assumption now I will apply spearman correlation analysis on them to find the correlation.

```
243
244 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`,method="spearman")
245
246
```

```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Fine Aggregate (component 7)(kg in a m^3 mixture)`,method="spearman")
[1] -0.179991
>
```

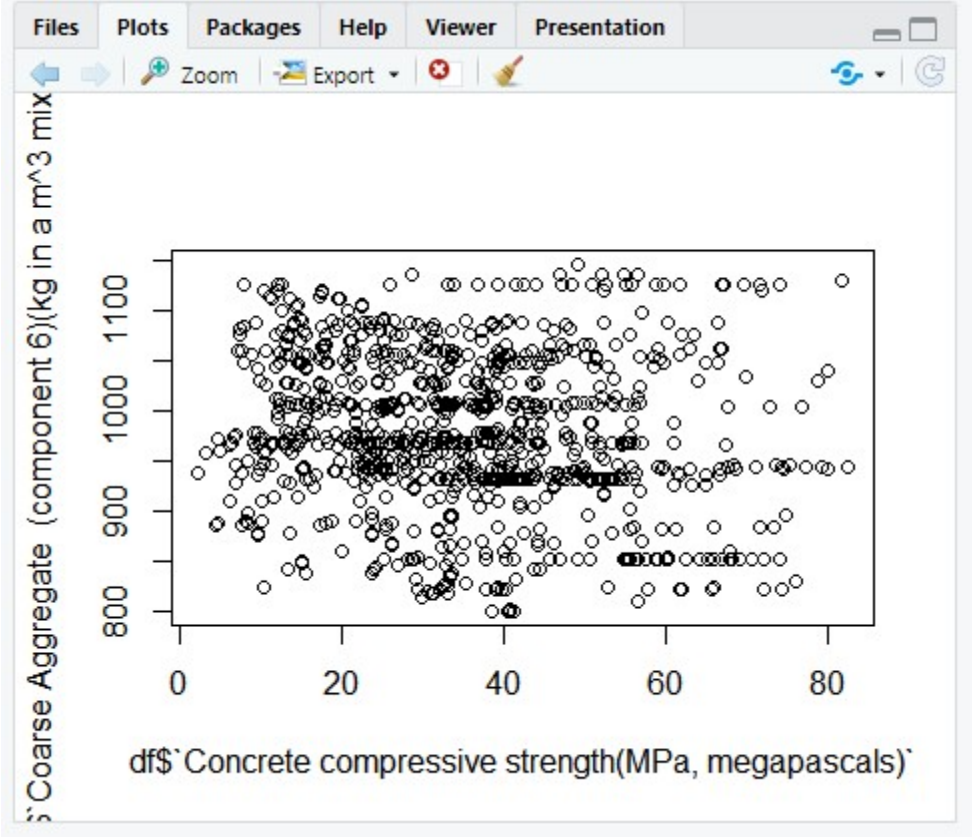
Since there is a negative correlative between them.

Correlation b/w compressive concrete strength and Coarse Aggregate:-

I will do the correlation analysis between these variables and assumption I have tell above. So I will take every step according to these assumptions.

First of all I plot a scatter plot between these two variables to check the linearity.

```
251
252 plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`)
253
254 # Shapiro-Wilk Test
```



From this visualization we can see that there is not linear.

Now I will do shipro-wilk test on these variables individually to see these variables are normally distributed or not.

```
253
254 # Shapiro-Wilk Test
255 shapiro_test1 <- shapiro.test(df$`Concrete compressive strength(MPa, megapascals)`)
256 shapiro_test2 <- shapiro.test(df$`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`)
257
258 # Output Results
259 print(shapiro_test1)
260 print(shapiro_test2)
261
```

```
> # Output Results
> print(shapiro_test1)

        Shapiro-Wilk normality test

data:  df$`Concrete compressive strength(MPa, megapascals)`
W = 0.97979, p-value = 9.023e-11

> print(shapiro_test2)

        Shapiro-Wilk normality test

data:  df$`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`
W = 0.98245, p-value = 8.346e-10

>
```

Now from above results we can see that both the resultants values are below 0.05. So both variables are not normally distributed.

According to assumption now I will apply spearman correlation analysis on them to find the correlation.

```
262
263 cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`,method="spearman")
264
265
```

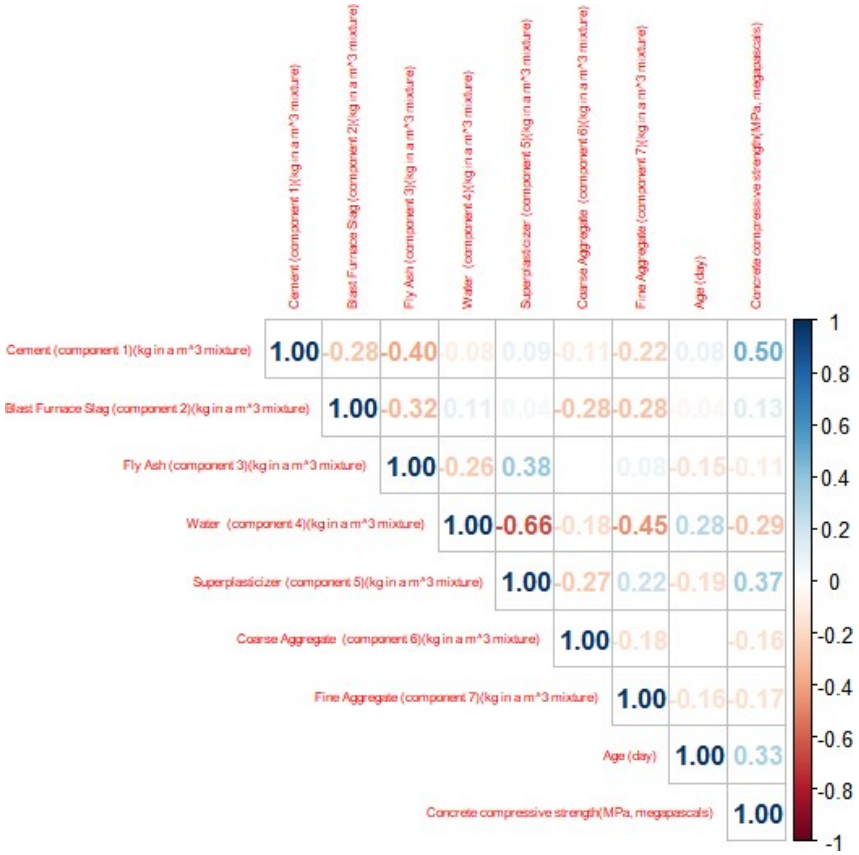
```
> cor(df$`Concrete compressive strength(MPa, megapascals)`,df$`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`,method="spearman")
[1] -0.1835157
>
```

Since there is a negative correlative between them.

Overall correlative of numerical variable:-

Now here we can see the overall correlation of numerical variables.

```
268
269 continuose_discrete<-df %>% select(-'Concrete Category',-'Contains Fly Ash')
270 head(continuose_discrete)
271 round(cor(continuose_discrete, method = "spearman"), digit=2)
272 corrplot(cor(continuose_discrete),method="number",type="upper",tl.cex = 0.5)
273
```



Correlation Between Numerical and Categorical (Nominal Scale) Variables

I use a point biserial correlation coefficient to measure the relationship between numerical variable and categorical variable.
Point-biserial correlation coefficient measures a value between -1 and 1.

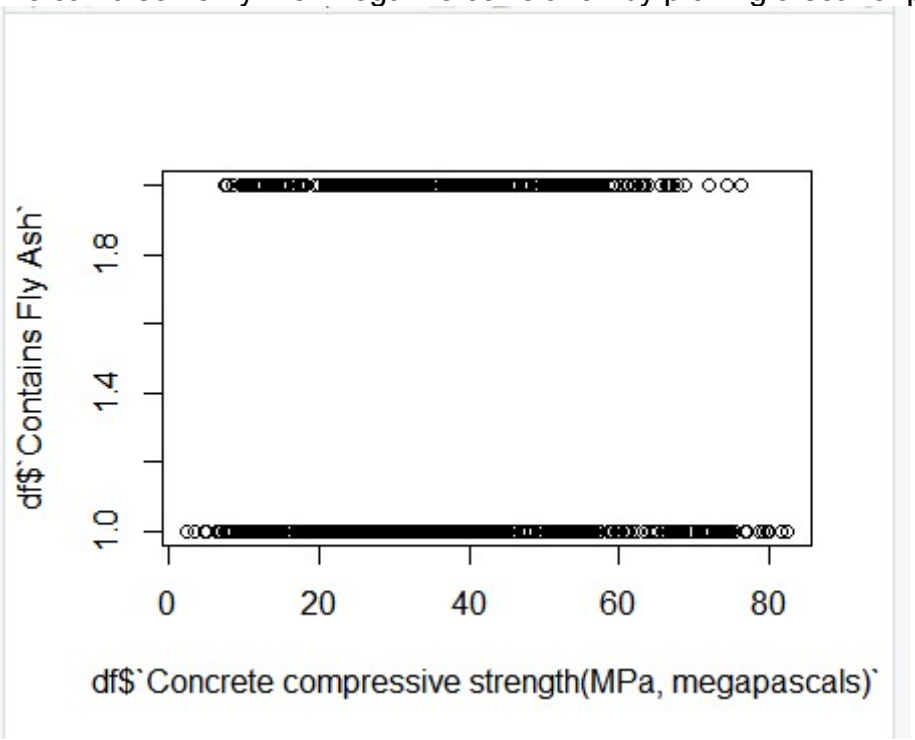
- -1 indicates a perfectly negative correlation
- 0 indicates no correlation
- 1 indicates a perfectly positive correlation

Now here I am going to do the point-biserial relationship between concrete strength and Contains Fly Ash.

```
276
277 install.packages("psych")
278 library(psych)  # You may need to install this package
279 # Convert the binary categorical variable to a factor
280 df$`Contains Fly Ash` <- as.factor(df$`Contains Fly Ash`)
281
282 # Point-biserial correlation
283
284 point_biserial <- cor.test(as.numeric(df$`Concrete compressive strength(MPa, megapascals)`),
285                            as.numeric(df$`Contains Fly Ash`), method = "pearson")
286 print(point_biserial)
287
```

```
+                          as.numeric(df$`Contains Fly Ash`), method = "pearson")
> print(point_biserial)

        Pearson's product-moment correlation

data:  as.numeric(df$`Concrete compressive strength(MPa, megapascals)`) and as.numeric(df$`Contains Fly Ash
`)
t = -2.0269, df = 1028, p-value = 0.04293
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.123698037 -0.002016098
sample estimates:
        cor
-0.06309154
```

From above results correlation coefficient is -0.063 its mean there is negative correlation between the two variable.
We can also verify their negative correlation by plotting a scatter plot as well.

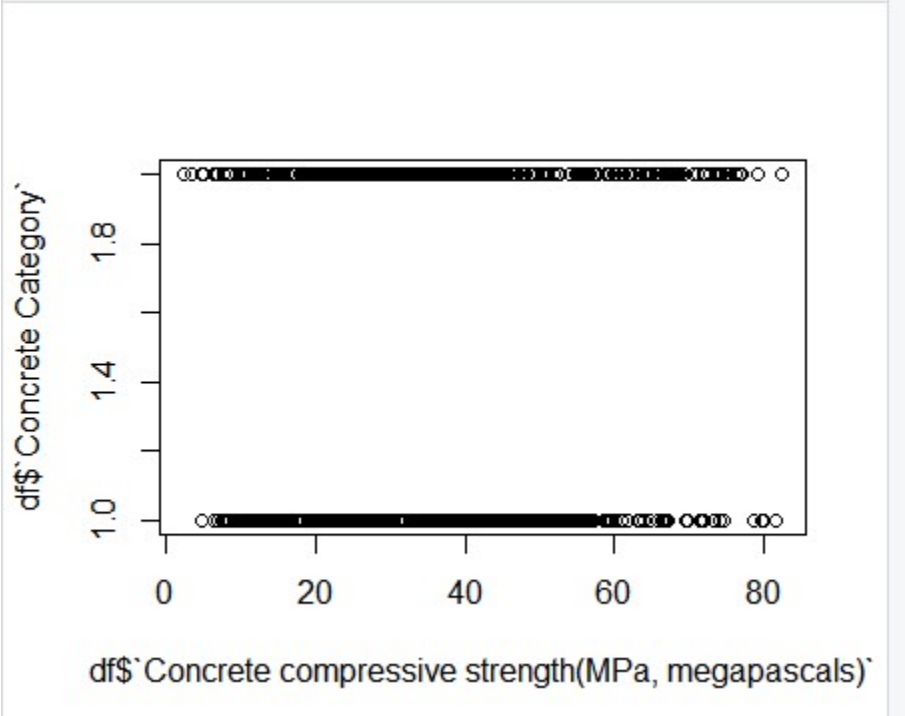


Now I am going to do biserial correlation between concrete strength and concrete category.

```
288
289
290 # Convert the binary categorical variable to a factor
291 df$`Concrete Category` <- as.factor(df$`Concrete Category`)
292
293
294 # Point-biserial correlation for Concrete Category
295
296 point_biserial <- cor.test(as.numeric(df$`Concrete Category`),
297                            as.numeric(df$`Concrete Category`), method = "pearson")
298 print(point_biserial)
299
```

```
> print(point_biserial)

        Pearson's product-moment correlation

data:  as.numeric(df$`Concrete Category`) and as.numeric(df$`Concrete Category`)
t = Inf, df = 1028, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 1 1
sample estimates:
cor
  1
```

From above we can see that the resultant variable has a value of 1 mean these variables have a position correlation.
We can also verify this correlation resultant by plotting as scatter plot as follow between these 2 variables.



1

Formulate regression problem(s) relating to your chosen scenario and application of appropriate regression techniques on the dataset.

Formulation of Regression Problem:-

The goal of this regression analysis is to model and predict concrete compressive strength based on various components in the concrete mixture. The main aim of this analysis is to determine which component contribute highest amount in the concrete mixture and to formulate equations for predictions as well. Furthermore I also predicted that how the various factors effect the concrete mixture.

Step-by-step implication of regression model:-

First of all I check the variables and their data type as

```
575  #check variables and data types
576  str(df)
577
```

```
> plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Contains Fly Ash`)
> plot(df$`Concrete compressive strength(MPa, megapascals)`,df$`Concrete Category`)
> str(df)
tibble [1,030 × 11] (S3: tbl_df/tbl/data.frame)
 $ Cement (component 1)(kg in a m^3 mixture)            : num [1:1030] 540 540 332 332 199 ...
 $ Blast Furnace Slag (component 2)(kg in a m^3 mixture): num [1:1030] 0 0 142 142 132 ...
 $ Fly Ash (component 3)(kg in a m^3 mixture)           : num [1:1030] 0 0 0 0 0 0 0 0 0 0 ...
 $ Water  (component 4)(kg in a m^3 mixture)            : num [1:1030] 162 162 228 228 192 228 228 228 228 2
28 ...
 $ Superplasticizer (component 5)(kg in a m^3 mixture)  : num [1:1030] 2.5 2.5 0 0 0 0 0 0 0 0 ...
 $ Coarse Aggregate  (component 6)(kg in a m^3 mixture) : num [1:1030] 1040 1055 932 932 978 ...
 $ Fine Aggregate (component 7)(kg in a m^3 mixture)    : num [1:1030] 676 676 594 594 826 ...
 $ Age (day)                                            : num [1:1030] 28 28 270 365 360 90 365 28 28 28 ...
 $ Concrete Category                                    : Factor w/ 2 levels "Coarse","Fine": 1 1 1 1 2 1 1
1 1 1 ...
 $ Contains Fly Ash                                     : Factor w/ 2 levels "FALSE","TRUE": 1 1 1 1 1 1 1 1
1 1 ...
 $ Concrete compressive strength(MPa, megapascals)      : num [1:1030] 80 61.9 40.3 41.1 44.3 ...
>
```

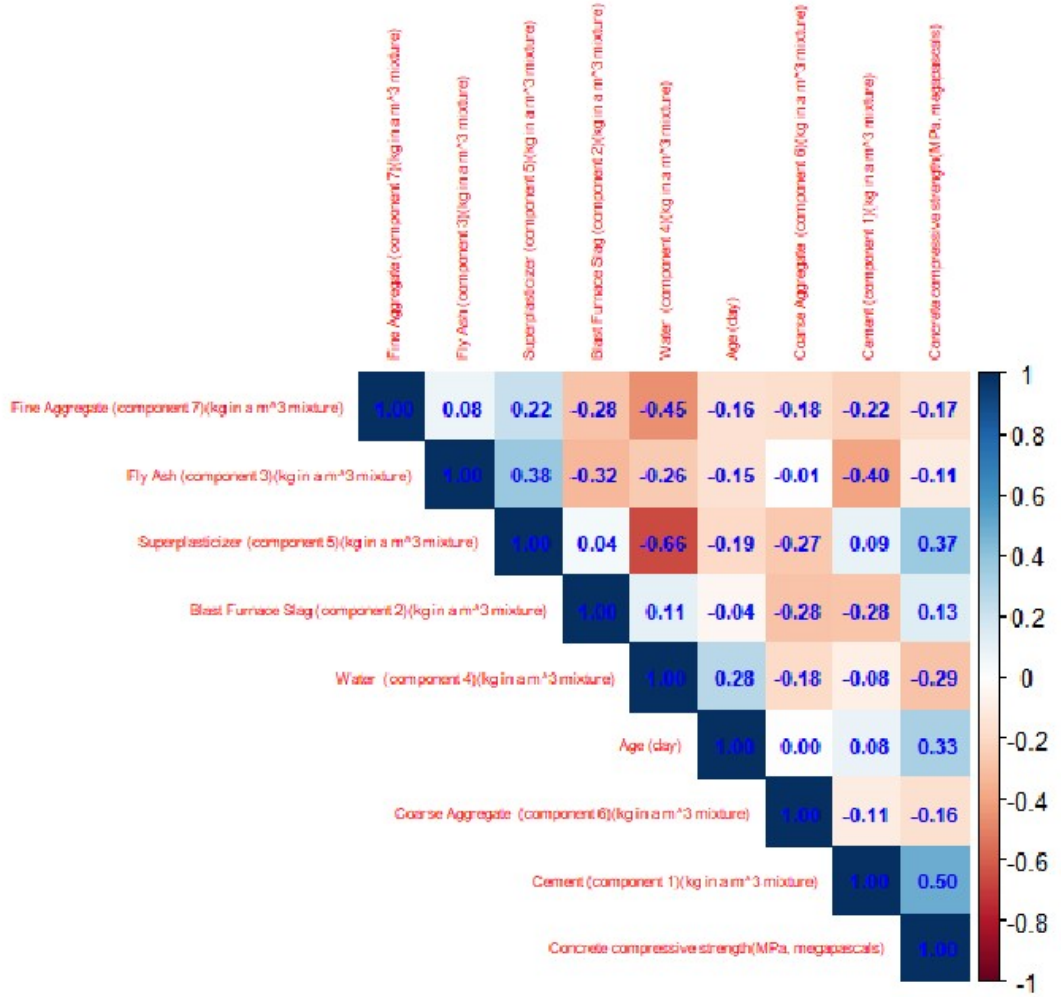
Now I have to apply regression analysis so for this purpose I am going to extract only the numeric columns as follow

```
578  # Extract only numerical columns
579  df_reduced <- df %>% select_if(is.numeric)
580
581  # Display the first few rows of numerical data
582  head(df_reduced)
583
```

```
Console  Terminal  Background Jobs
R 4.4.1 · C:/Users/Chaudhary Computer/Desktop/Applied Statistics and Data Visualisation/Assignment/Task 2 - Statistical Analysis/concrete+compressive+strength/
# A tibble: 6 × 9
  Cement (component 1)(kg in a m^3…¹ Blast Furnace Slag (…² Fly Ash (component 3…³ Water  (component 4)…⁴
                               <dbl>                  <dbl>                  <dbl>                  <dbl>
1                              540                     0                      0                     162
2                              540                     0                      0                     162
3                              332.                  142.                     0                     228
4                              332.                  142.                     0                     228
5                              199.                  132.                     0                     192
6                              266                   114                      0                     228
# i abbreviated names: ¹`Cement (component 1)(kg in a m^3 mixture)`,
#   ²`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,
#   ³`Fly Ash (component 3)(kg in a m^3 mixture)`, ⁴`Water  (component 4)(kg in a m^3 mixture)`
# i 5 more variables: `Superplasticizer (component 5)(kg in a m^3 mixture)` <dbl>,
#   `Coarse Aggregate  (component 6)(kg in a m^3 mixture)` <dbl>,
#   `Fine Aggregate (component 7)(kg in a m^3 mixture)` <dbl>, `Age (day)` <dbl>,
#   `Concrete compressive strength(MPa, megapascals)` <dbl>
>
```

Then I build a correlation matrix between all numerical variables.

```
582
583 # Compute the correlation matrix
584 cor_matrix <- cor(numerical_data, use = "complete.obs")  # Handles missing values if any
585
586 # Plot the correlation matrix
587 corrplot(cor_matrix, method = "color", type = "upper",
588          tl.cex = 0.5,            # Adjust text label size
589          tl.col = "red",        # Text color for labels
590          order = "hclust",        # Hierarchical clustering order
591          addCoef.col = "blue",  # Show correlation coefficients on plot
592          number.cex = 0.7)        # Size of correlation coefficient text
593
```



Now from above above confusion matrix we can see that the cement has the highest value of correlation with compressive concrete strength.
So I will use farward step wise method to implement regression problem in which the correlation analysis starts from hight correlation value to lower correlation value.

Now I am going to implement SLR(simple linear regression) model on cement and compressive strength.

```
#the correlation b/w cement and compressive strenght is high
#using farward stepwise method
# SLR model: Compressive Strength ~ Cement Content
model_cement <- lm(`Concrete compressive strength(MPa, megapascals)` ~ `Cement (component 1)(kg in a m^3 mixture)`, data = df_reduced)

# Model summary
summary(model_cement)
```

```
> summary(model_cement)

Call:
lm(formula = `Concrete compressive strength(MPa, megapascals)` ~
    `Cement (component 1)(kg in a m^3 mixture)`, data = df_reduced)

Residuals:
    Min      1Q  Median      3Q     Max
-40.594 -10.952  -0.572   9.992  43.241

Coefficients:
                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                          13.442795   1.296925   10.37   <2e-16 ***
`Cement (component 1)(kg in a m^3 mixture)`  0.079580   0.004324   18.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.5 on 1028 degrees of freedom
Multiple R-squared:  0.2478,    Adjusted R-squared:  0.2471
F-statistic: 338.7 on 1 and 1028 DF,  p-value: < 2.2e-16

>
```

From the above results we can see that the value of multiple R-squared is 0.24 which means that the 24% of variability in compressive concrete strength can be determined by the cement. This 24% value is consider as high as compared to other individual components.
Both coefficients are significant and the SLR equation will be:

Concrete strenght = 13.442795+ 0.079580*cement component

Now to visualize the fitted regression line I am going to draw a scatter plot as follow.

```
# Plot the regression line
plot(df_reduced$`Cement (component 1)(kg in a m^3 mixture)`, df_reduced$`Concrete compressive strength(MPa, megapascals)`,
     main="Compressive Strength vs Cement Content",
     xlab="Cement Content (kg)", ylab="Compressive Strength (MPa)")
abline(model_cement, col="blue")
```

Checking SLR Assumption:-

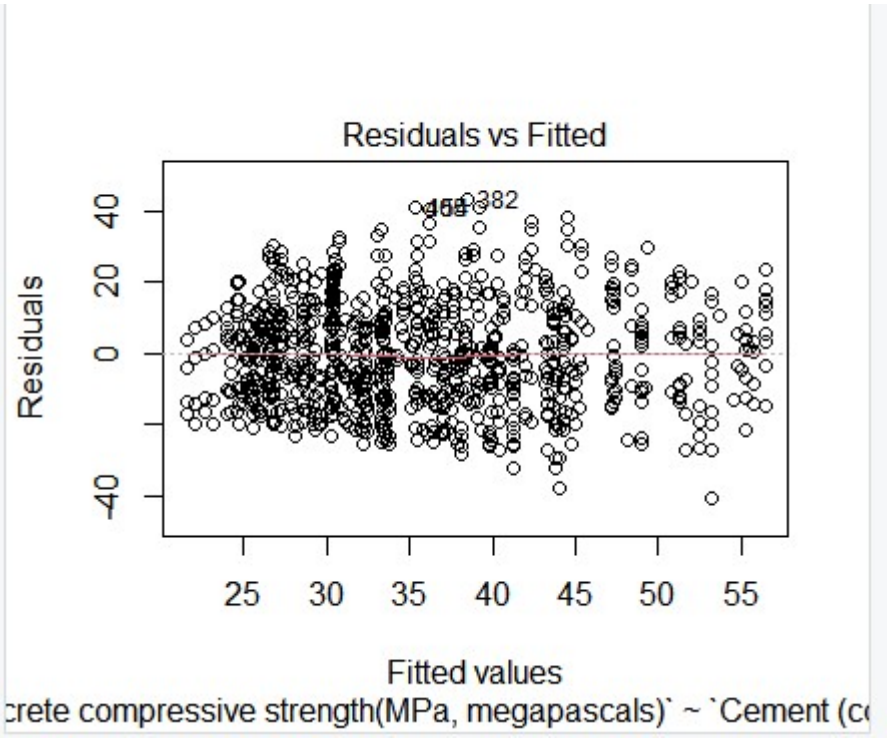
- **Linearity:-**

In the EDA I have already analyzed that both of them are linear.

- **Residuals' Independence:-**

The correlation should be approximately zero. So I will verify through a scatter plot as

```
#residual independence
plot(model_cement, 1)
```

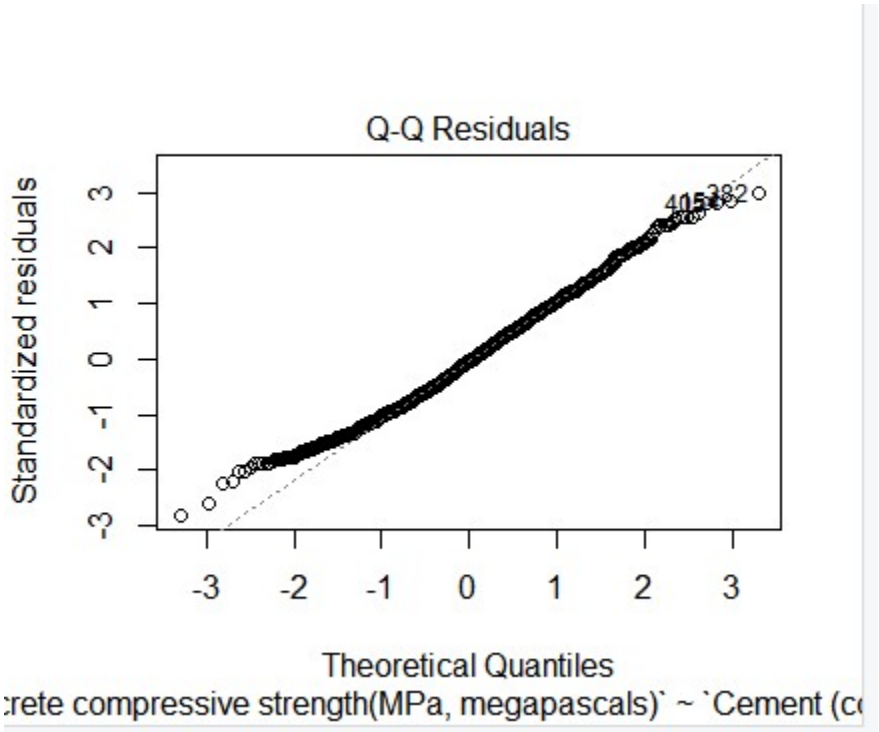


So we can see that the red line is at zero which means there is no correlation.

Normality of residuals:-

The residuals must be normally distributed.

```
613
614
615  #check normality of residual independence
616  plot(model_cement,2)
617
```

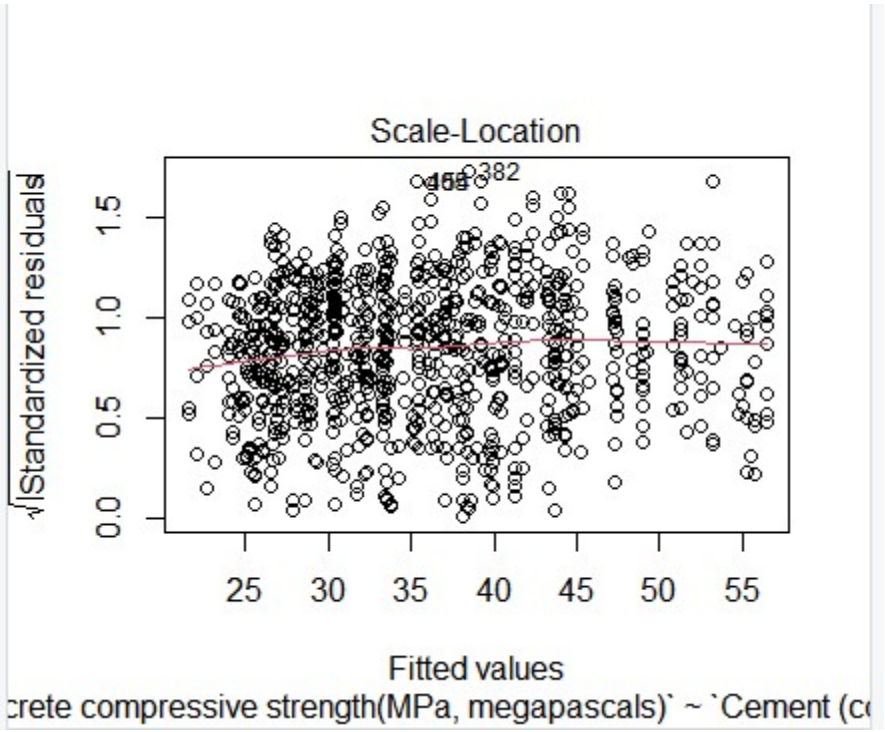


The residual are near the red line approximately so this assumption also met.

Equal variances of the residuals (Homoscedasticity):-

This assumption say the variance of the residual are constant and are not related to fitted value.

```
618
619  #checking Equal variances of the residuals (Homoscedasticity)
620  plot(model_cement,3)
621
622
```



From this graph we can see that there is no proper pattern around the red line mean that residuals have roughly equal variability at all fitted values.

Report the result:-

Now to report the result I will take the whole column of cement from my data set and calculate the compressive concrete strength using the formula

Concrete strenght = 13.442795+ 0.079580*cement component

Then I will perform the SLR on this data and show the results

```
624 #reporting the results
625 #for cement value 332.5
626
627 data <- data.frame(
628   cement_component = df$`Cement (component 1)(kg in a m^3 mixture)`,
629   compressive_strength = df$`Concrete compressive strength(MPa, megapascals)`
630 )
631 # Perform the simple linear regression
632 model <- lm(compressive_strength ~ cement_component, data = data)
633
634 # Print the summary of the model to get the results
635 summary(model)
636
637 # Output the coefficients
638 coefficients(model)
639
640
```

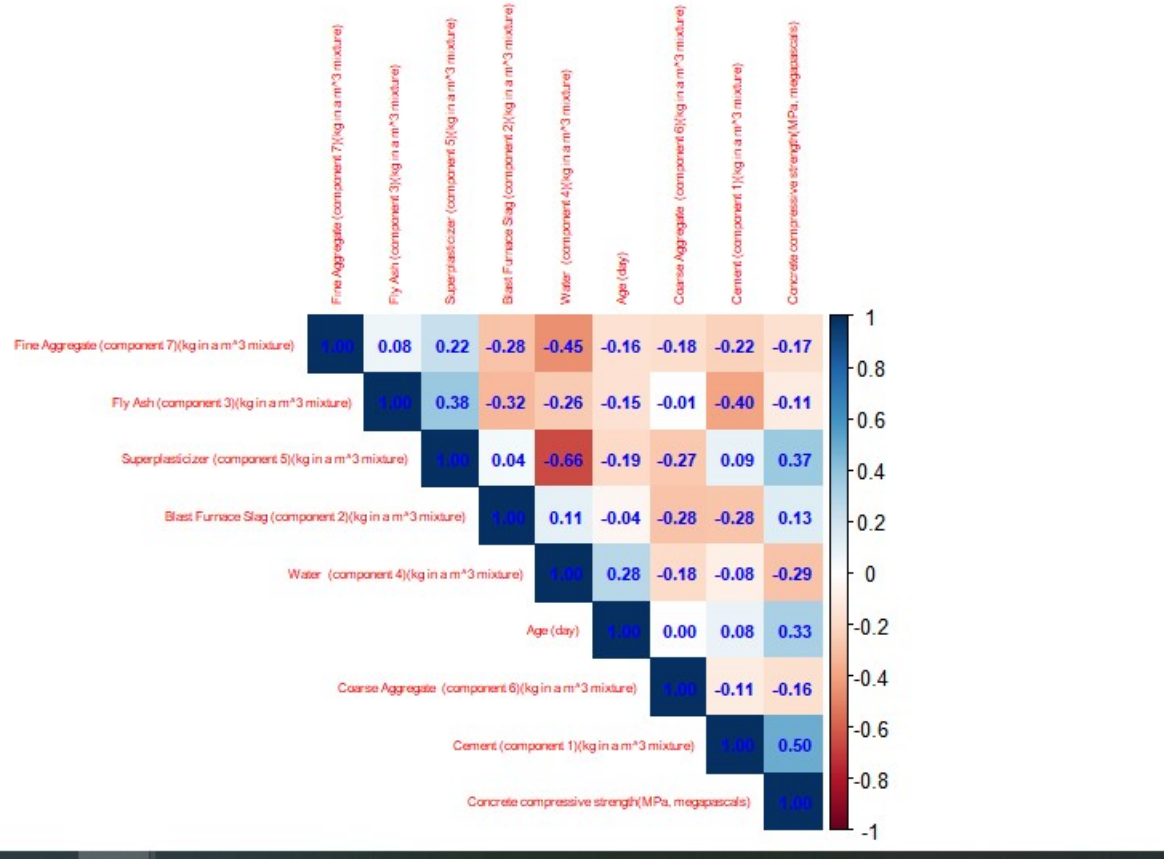
```
Call:
lm(formula = compressive_strength ~ cement_component, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-40.594 -10.952  -0.572   9.992  43.241

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      13.442795   1.296925   10.37   <2e-16 ***
cement_component  0.079580   0.004324   18.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.5 on 1028 degrees of freedom
Multiple R-squared:  0.2478,    Adjusted R-squared:  0.2471
F-statistic: 338.7 on 1 and 1028 DF,  p-value: < 2.2e-16
```

From the above results we can see that there is low value for R^2 and high value for residual error which indicate that cement is not fully explain the compressive concrete strength. While p-value is low which indicate that the cement is a strong predictor of compressive concrete strength. So I need to add more predictor variables that will fully explain the compressive concrete strength.

MLR:-

As we are using farward bias approach. Now for the purpose of applying MLR for better results I am again going to plot confusion matrix and see the relative correlation of different variables with concrete strength and apply MLR with highest value of correlation towards the variables with lower values of correlation respectively.

```r
# Plot the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper",
         tl.cex = 0.5,            # Adjust text label size
         tl.col = "red",          # Text color for labels
         order = "hclust",        # Hierarchical clustering order
         addCoef.col = "blue",    # Show correlation coefficients on plot
         number.cex = 0.7)        # Size of correlation coefficient text
```



From above correlation matrix we can see that after cement highest correlation is with superplaster than with Age and than with Blast Furnace Slag.

Now I am going to build MLR with superplaster and cement.

```
653
654  # Fit the model with the square root transformation of compressive strength
655  model_2 <- lm((`Concrete compressive strength(MPa, megapascals)`) ~
656                  `Cement (component 1)(kg in a m^3 mixture)` +
657                  `Superplasticizer (component 5)(kg in a m^3 mixture)`,
658                data = df_reduced)
659
660  # Display the summary of the model
661  summary(model_2)
662
```

```
> summary(model_2)

Call:
lm(formula = (`Concrete compressive strength(MPa, megapascals)`) ~
    `Cement (component 1)(kg in a m^3 mixture)` + `Superplasticizer (component 5)(kg in a m^3 mixture)`,
    data = df_reduced)

Residuals:
    Min      1Q  Median      3Q     Max
-33.949 -10.032  -0.515   9.117  43.676

Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                                       9.190242   1.250302    7.35 4.03e-13 ***
`Cement (component 1)(kg in a m^3 mixture)`       0.074794   0.004036   18.53  < 2e-16 ***
`Superplasticizer (component 5)(kg in a m^3 mixture)` 0.902460   0.070603   12.78  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.47 on 1027 degrees of freedom
Multiple R-squared:  0.3511,    Adjusted R-squared:  0.3498
F-statistic: 277.8 on 2 and 1027 DF,  p-value: < 2.2e-16
```

from above results we can see that value of R square is 0.35 which means that 35% of concrete strength depends on cement and superplaster and the less value of p shows that these variable has high correlation with concrete strength.

MLR equation:-

Concrete compressive strength (MPa)=9.1902+(0.0748×Cement (kg))+(0.9025×Superplasticizer (kg))

Model 3:-

Now I am again going to build another MLR model which also includes water.

```
669
670  # Fit the  model
671  model_3 <- lm((`Concrete compressive strength(MPa, megapascals)`) ~
672                  `Cement (component 1)(kg in a m^3 mixture)` +
673                  `Superplasticizer (component 5)(kg in a m^3 mixture)` +
674                  `Water  (component 4)(kg in a m^3 mixture)`,
675                data = df_reduced)
676
677  # Display the summary of the model
678  summary(model_3)
```

```
> summary(model_3)

Call:
lm(formula = (`Concrete compressive strength(MPa, megapascals)`) ~
    `Cement (component 1)(kg in a m^3 mixture)` + `Superplasticizer (component 5)(kg in a m^3 mixture)` +
        `Water  (component 4)(kg in a m^3 mixture)`, data = df_reduced)

Residuals:
    Min      1Q  Median      3Q     Max
-33.688 -10.070  -0.383   8.916  41.664

Coefficients:
                                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     19.881468   5.291765   3.757 0.000182 ***
`Cement (component 1)(kg in a m^3 mixture)`      0.074565   0.004031  18.500  < 2e-16 ***
`Superplasticizer (component 5)(kg in a m^3 mixture)`  0.775463   0.093275   8.314 2.92e-16 ***
`Water  (component 4)(kg in a m^3 mixture)`     -0.054189   0.026065  -2.079 0.037863 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.45 on 1026 degrees of freedom
Multiple R-squared:  0.3538,    Adjusted R-squared:  0.3519
F-statistic: 187.2 on 3 and 1026 DF,  p-value: < 2.2e-16
```

From above results we can see that these variables make 35% of the total concrete strength and the low values of p suggest stronger correlation. The value of intercept is also lower than 0.05 which is good. But the model does not show good value of R square because these components are not giving good predictions for concrete strength so we need to add more components.

MLR equation:-

Concrete Compressive Strength (MPa)=19.88+0.0746·Cement+0.7755·Superplasticizer−0.0542·Water

Model 4:-

Now I am going to add another variable as well.

```
683  # Fit the model
684  model_4 <- lm((`Concrete compressive strength(MPa, megapascals)`) ~
685                  `Cement (component 1)(kg in a m^3 mixture)` +
686                  `Superplasticizer (component 5)(kg in a m^3 mixture)` +
687                  `Water  (component 4)(kg in a m^3 mixture)` +
688                  `Fine Aggregate (component 7)(kg in a m^3 mixture)`+
689                  `Coarse Aggregate  (component 6)(kg in a m^3 mixture)`+
690                  `Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,
691                data = df_reduced)
692
693  # Display the summary of the model
694  summary(model_4)
695
```

```
> summary(model_4)

Call:
lm(formula = (`Concrete compressive strength(MPa, megapascals)`) ~
    `Cement (component 1)(kg in a m^3 mixture)` + `Superplasticizer (component 5)(kg in a m^3 mixture)` +
        `Water  (component 4)(kg in a m^3 mixture)` + `Fine Aggregate (component 7)(kg in a m^3 mixture)` +
        `Coarse Aggregate  (component 6)(kg in a m^3 mixture)` +
        `Blast Furnace Slag (component 2)(kg in a m^3 mixture)`,
    data = df_reduced)

Residuals:
    Min      1Q  Median      3Q     Max
-32.858  -9.671  -0.371   9.055  35.294

Coefficients:
                                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                                     97.466015  19.986117   4.877 1.25e-06 ***
`Cement (component 1)(kg in a m^3 mixture)`      0.074995   0.004848  15.468  < 2e-16 ***
`Superplasticizer (component 5)(kg in a m^3 mixture)`    0.407396   0.111726   3.646 0.000279 ***
`Water  (component 4)(kg in a m^3 mixture)`     -0.215917   0.037539  -5.752 1.17e-08 ***
`Fine Aggregate (component 7)(kg in a m^3 mixture)`     -0.037083   0.007979  -4.647 3.80e-06 ***
`Coarse Aggregate  (component 6)(kg in a m^3 mixture)`  -0.020935   0.008133  -2.574 0.010187 *
`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`  0.040552   0.006111   6.636 5.23e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.53 on 1023 degrees of freedom
Multiple R-squared:  0.4409,    Adjusted R-squared:  0.4376
F-statistic: 134.5 on 6 and 1023 DF,  p-value: < 2.2e-16

>
```

Here the value of R square suggest that 44% of concrete strength depends on these variable and low value of p suggest strong correlation. The value of pr(>t) is also lower than 0.05

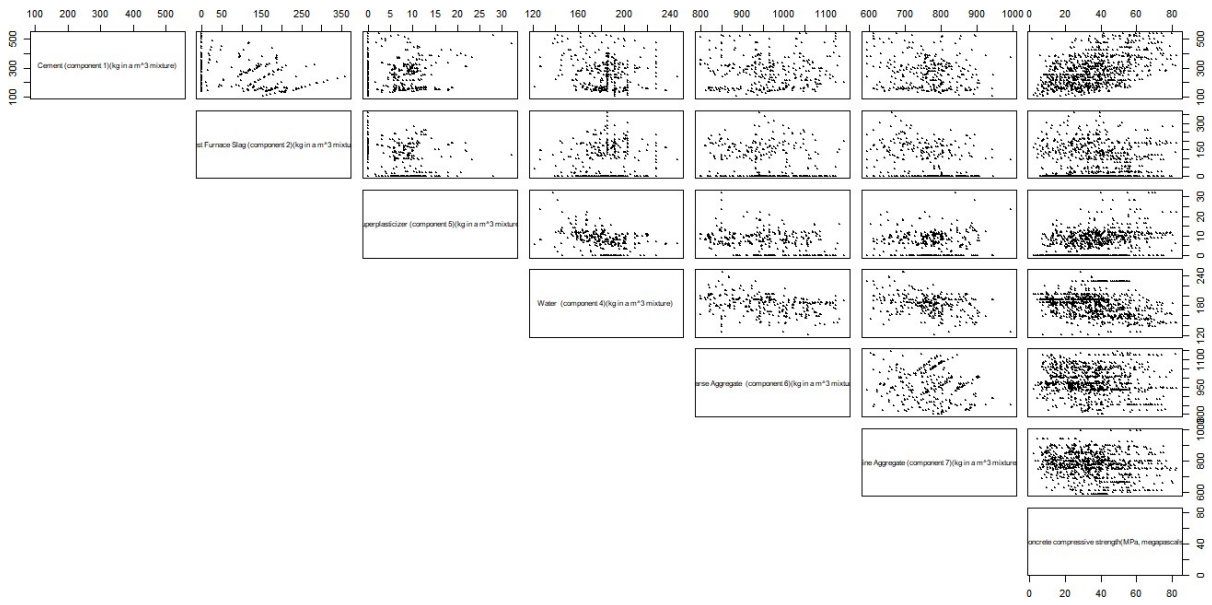
MLR equation:-

Concrete Compressive Strength (MPa)=97.47+0.075·Cement+0.407·Superplasticizer−0.216·Water−0.037·Fine Aggregate−0.021·Coarse Aggregate+0.041·Blast Furnace Slag

Checking Assumptions for MLR:-

Linearity:-

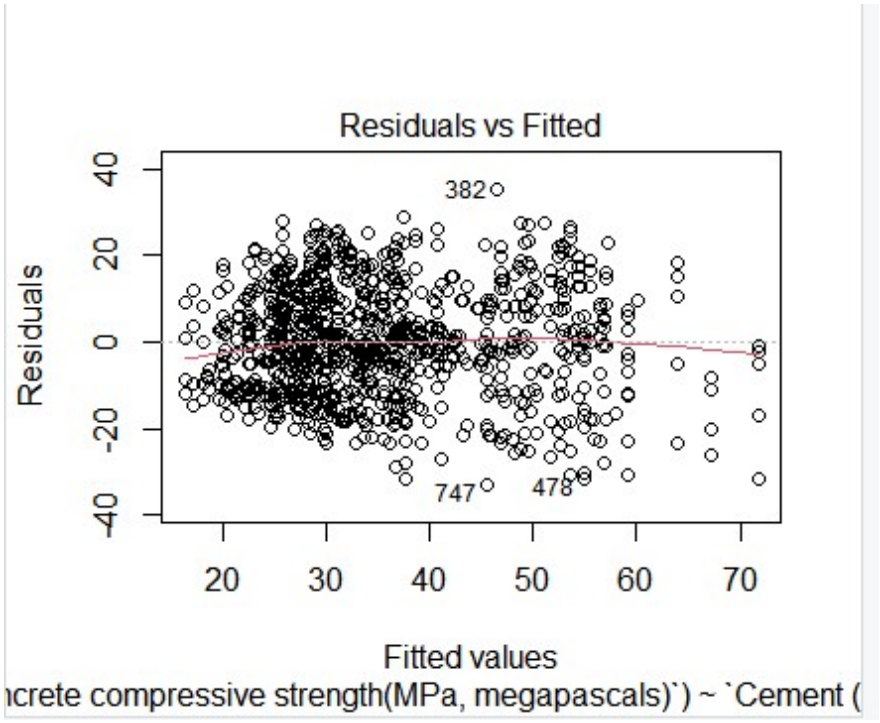
```
698
699 names(df_reduced)
700
701 #checking normality
702 pairs(df_reduced[,c(1,2,5,4,6,7,9)], lower.panel = NULL, pch = 10,cex = 0.0001)
703
```



From the first row we can see that all the variables are somehow linear.

Residual independence:-

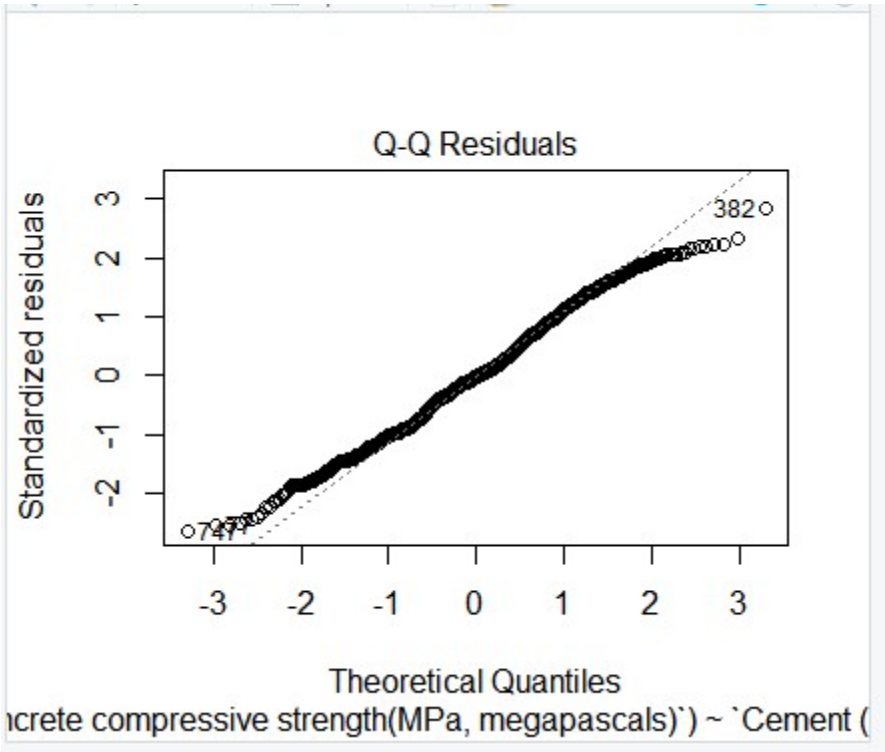
```
704  #checking residual independence
705  #plot(model_2,1)
706  #plot(model_3,1)
707  plot(model_4,1)
708
```



from this graph we can see that the red line is at zero means there is no correlation. So this assumption is valid.

Normality of residuals:-

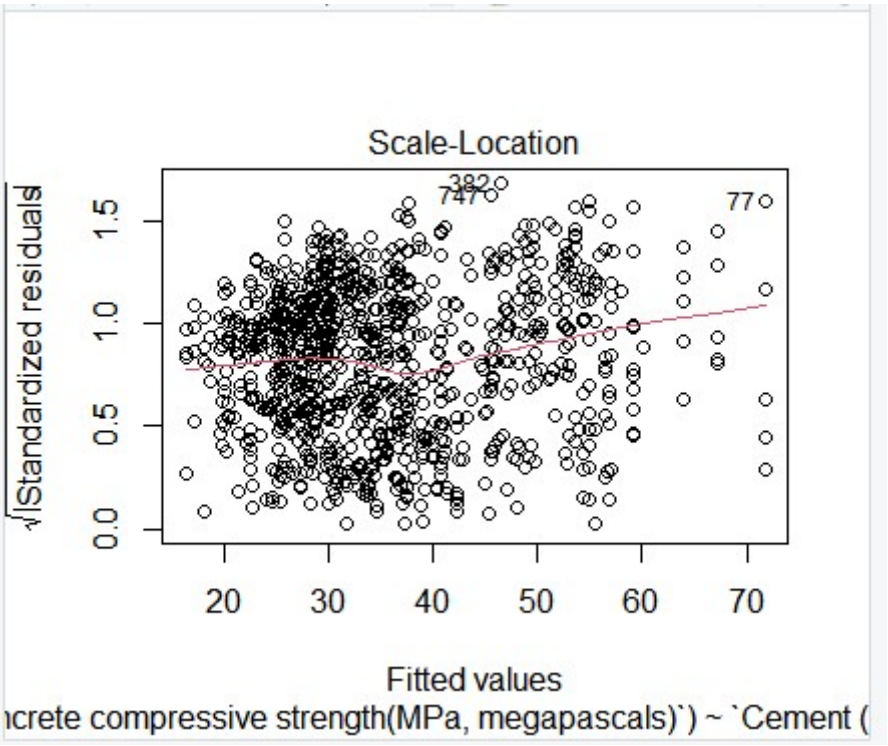
```
710
711  #Normality of residuals:
712  #plot(model_2, 2)
713  #plot(model_3, 2)
714  plot(model_4, 2)
715
716
```



we can see that all the observations are approximately near to red line. So this assumption also met.

Equal variances of the residuals (Homoscedasticity):-

```
716
717  #Equal variances of the residuals (Homoscedasticity)
718  #plot(model_2, 3)
719  #plot(model_3, 3)
720  plot(model_4, 3)
721
```



from above graph we can see that the observations are randomly split across the redline. So this assumption also met.

No multicollinearity:-

```
725
726 #No multicollinearity
727 #vif(model_2)
728 #vif(model_3)
729 vif(model_4)
730
```

```
> vif(model_4)
            `Cement (component 1)(kg in a m^3 mixture)`
                                                1.683258
  `Superplasticizer (component 5)(kg in a m^3 mixture)`
                                                2.920270
            `Water  (component 4)(kg in a m^3 mixture)`
                                                4.213499
    `Fine Aggregate (component 7)(kg in a m^3 mixture)`
                                                2.683400
 `Coarse Aggregate  (component 6)(kg in a m^3 mixture)`
                                                2.621644
`Blast Furnace Slag (component 2)(kg in a m^3 mixture)`
                                                1.822778
>
```

from above we can see that all the values are less than 5. So its mean there is no collinearity between independents variable and this assumption also met.

Report the results:-

Let's assume the following values for a new data point:

- Cement = 350 kg/m³
- Superplasticizer = 4.5 kg/m³
- Water = 200 kg/m³
- Fine Aggregate = 600 kg/m³
- Coarse Aggregate = 900 kg/m³
- Blast Furnace Slag = 150 kg/m³

Now my aim is to predict the concrete strength using these value in the MLR equation.

MLR equation:-

Concrete Compressive Strength (MPa)=97.47+0.075·Cement+0.407·Superplasticizer−0.216·Water−0.037·Fine Aggregate−0.021·Coarse Aggregate+0.041·Blast Furnace Slag

```
#report the results
# Define the values for the new data point
new_data <- data.frame(
  Cement = 350,              #  value for Cement (kg in a m^3 mixture)
  Superplasticizer = 4.5,    #  value for Superplasticizer (kg in a m^3 mixture)
  Water = 200,               #  value for Water (kg in a m^3 mixture)
  Fine_Aggregate = 600,      #  value for Fine Aggregate (kg in a m^3 mixture)
  Coarse_Aggregate = 900,    #  value for Coarse Aggregate (kg in a m^3 mixture)
  Blast_Furnace_Slag = 150   #  value for Blast Furnace Slag (kg in a m^3 mixture)
)

# Apply the regression equation to make the prediction
predicted_compressive_strength <- 97.47 +
  0.075 * new_data$Cement +
  0.407 * new_data$Superplasticizer -
  0.216 * new_data$Water -
  0.037 * new_data$Fine_Aggregate -
  0.021 * new_data$Coarse_Aggregate +
  0.041 * new_data$Blast_Furnace_Slag

# Print the predicted concrete compressive strength
print(predicted_compressive_strength)

```

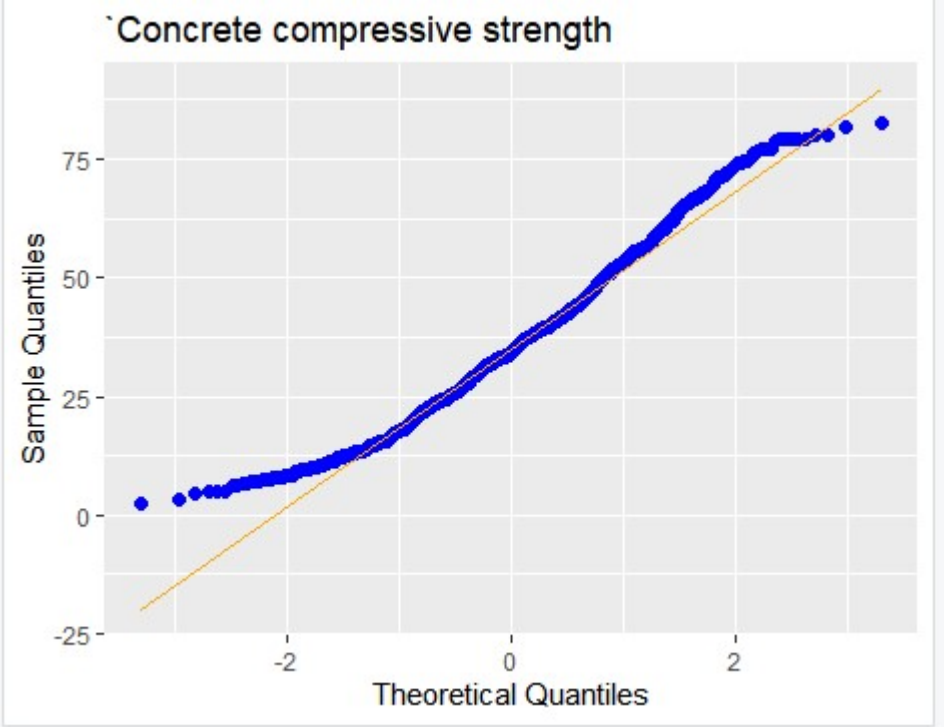
```
>
> # Print the predicted concrete compressive strength
> print(predicted_compressive_strength)
[1] 47.4015
```

From above result we can see that the compressive concrete strength would be 47 using these variables.

Formulate hypotheses relating to your chosen scenario and use appropriate tests to test them.

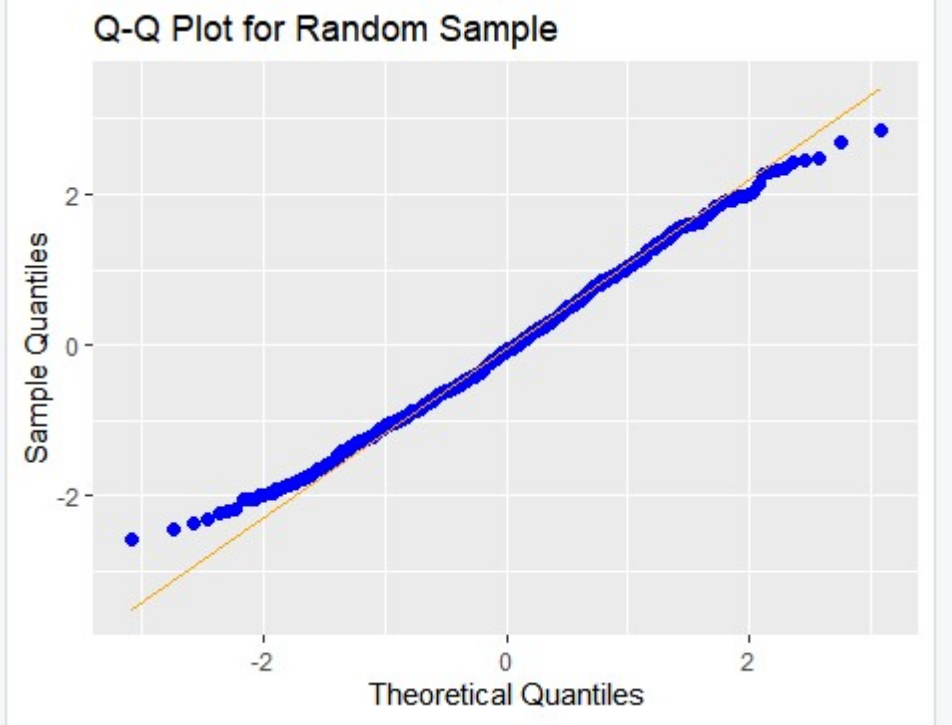
Now for applying hypothesis and using test on it first of all I would check that my data is normal or not for this purpose I write this code

```
#first of all iam checking the normality of CCS

# Create the Q-Q plot using ggplot2
ggplot(data = data.frame(sample = df$`Concrete compressive strength(MPa, megapascals)`), aes(sample = sample)) +
  stat_qq(size = 2, color = "blue") +            # Q-Q points
  stat_qq_line(color = "orange") +               # Q-Q line
  xlab("Theoretical Quantiles") +                # X-axis label
  ylab("Sample Quantiles") +                     # Y-axis label
  ggtitle("`Concrete compressive strength")          # Title for the plot

```



By viewing this graph I can see that the data is not normal but I write a code to see how the normal data looks like by using the random numbers as

```
# Create the Q-Q plot using ggplot2
ggplot(data = data.frame(sample = random_sample), aes(sample = sample)) +
  stat_qq(size = 2, color = "blue") +            # Q-Q points
  stat_qq_line(color = "orange") +               # Q-Q line
  xlab("Theoretical Quantiles") +                # X-axis label
  ylab("Sample Quantiles") +                     # Y-axis label
  ggtitle("Q-Q Plot for Random Sample")          # Title for the plot

```

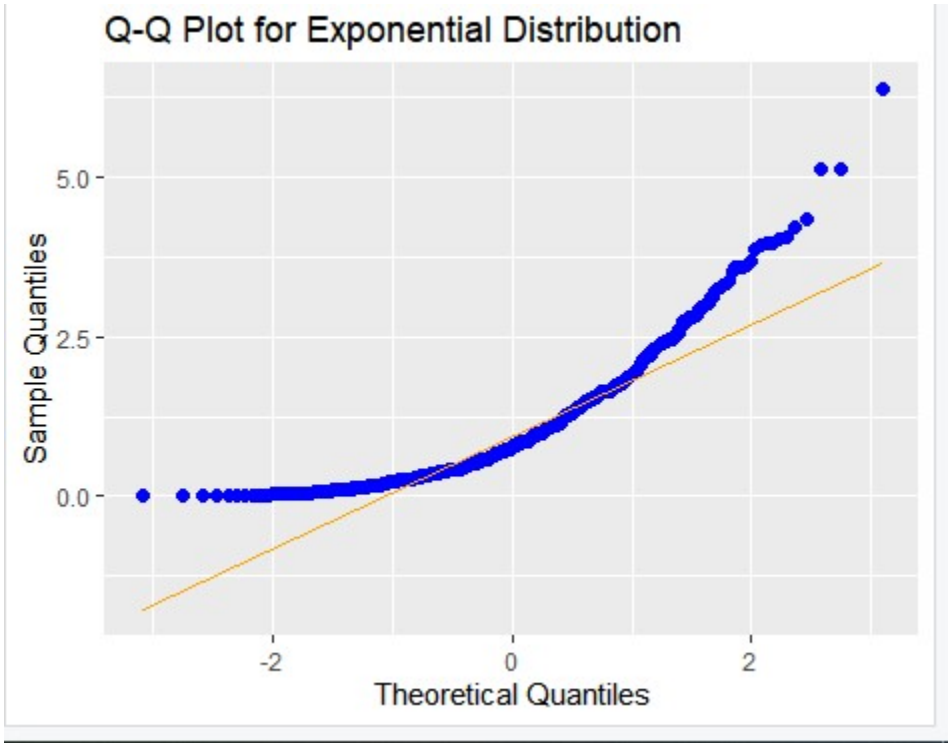


Now by comparing above both graphs I concluded that my data is not normal.

I write a code to see that my data resembles to exponential distribution or not so that I would apply transformation accordingly.

```
# Generate a random sample from an exponential distribution
random_sample2 <- rexp(500)

# Create the Q-Q plot using ggplot2
ggplot(data = data.frame(sample = random_sample2), aes(sample = sample)) +
  stat_qq(size = 2, color = "blue") +            # Q-Q points
  stat_qq_line(color = "orange") +               # Q-Q line
  xlab("Theoretical Quantiles") +                # X-axis label
  ylab("Sample Quantiles") +                     # Y-axis label
  ggtitle("Q-Q Plot for Exponential Distribution") # Title for the plot

```

1



By comparing I see that by data is not exponential distribution as well.

As I have seen that my data is deviated from head and tail around the red line. So I apply shipro-wilk test on my data to conform that it is normal or not.

```
352 #Now the there is deviation from tail and head lets try another test to justify whethe
353
354 data_to_test <- na.omit(df$`Concrete compressive strength(MPa, megapascals)`)
355
356 # Apply the Shapiro-Wilk test
357 shapiro_test_result <- shapiro.test(data_to_test)
358
359 # Print the results
360 print(shapiro_test_result)
361
```

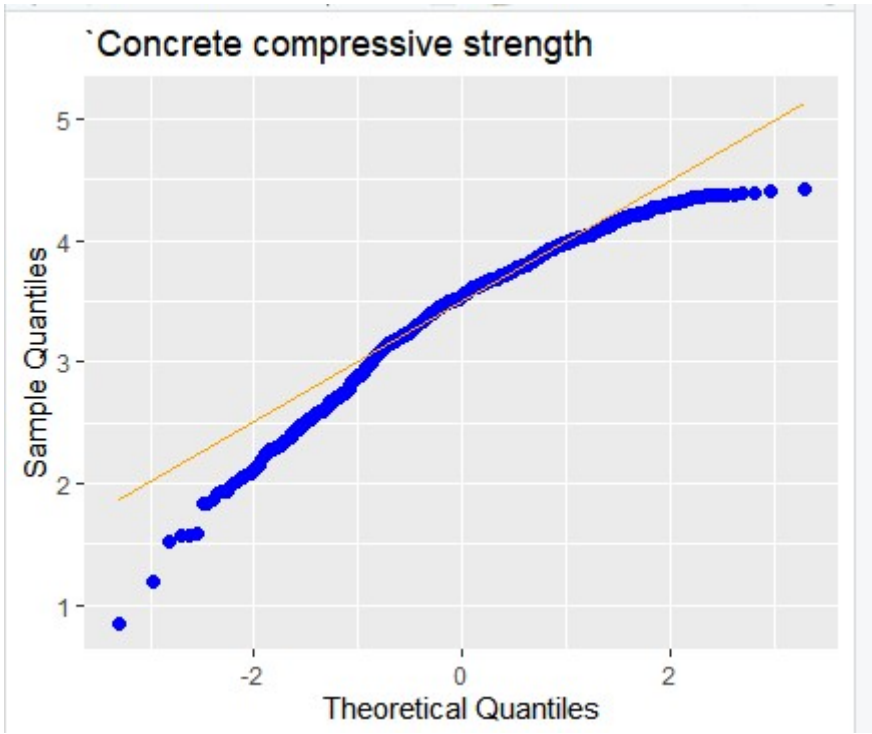
```
> # Print the results
> print(shapiro_test_result)

        Shapiro-Wilk normality test

data:  data_to_test
W = 0.97979, p-value = 9.023e-11

>
```

From this result I can see that the value of p is less than 0.05 so it is conformed that the data is not normal.

1

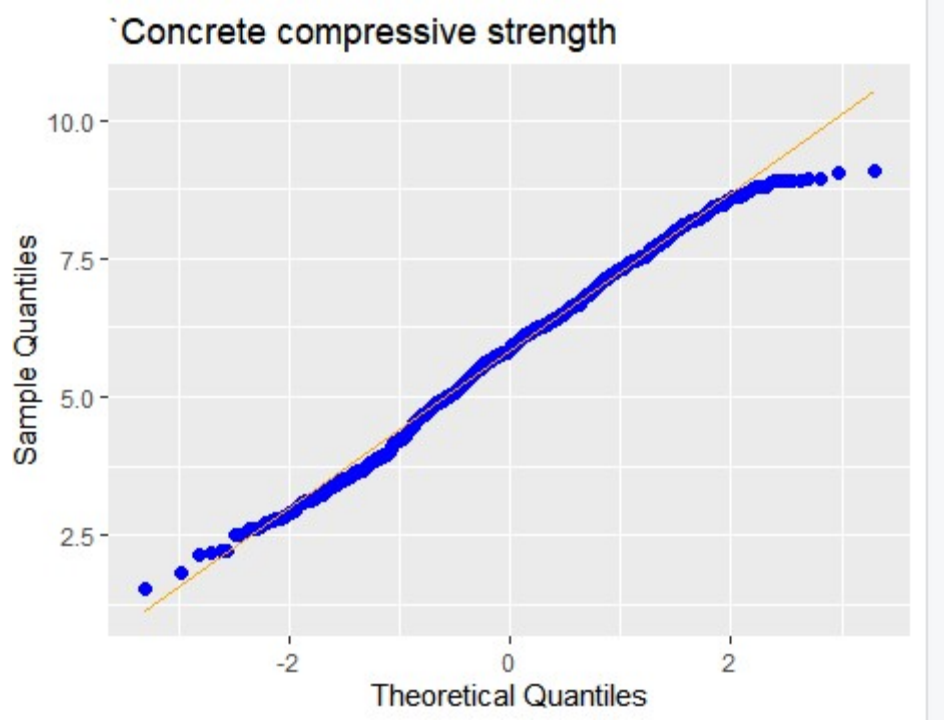
So I am going to apply different transformation techniques to make my data normal. First of all I apply Log transformation on my data and use q-q plot to see visualization of transformed data.

```
# Apply log transformation
df$`Log Concrete compressive strength(MPa, megapascals)` <- log(df$`Concrete compressive strength(MPa, megapascals)`)

# Print the first few rows of the transformed dataframe to verify
head(df)

# Create the Q-Q plot using ggplot2
ggplot(data = data.frame(sample = df$`Log Concrete compressive strength(MPa, megapascals)`), aes(sample = sample)) +
  stat_qq(size = 2, color = "blue") +            # Q-Q points
  stat_qq_line(color = "orange") +               # Q-Q line
  xlab("Theoretical Quantiles") +                # X-axis label
  ylab("Sample Quantiles") +                     # Y-axis label
  ggtitle("`Concrete compressive strength")          # Title for the plot

```



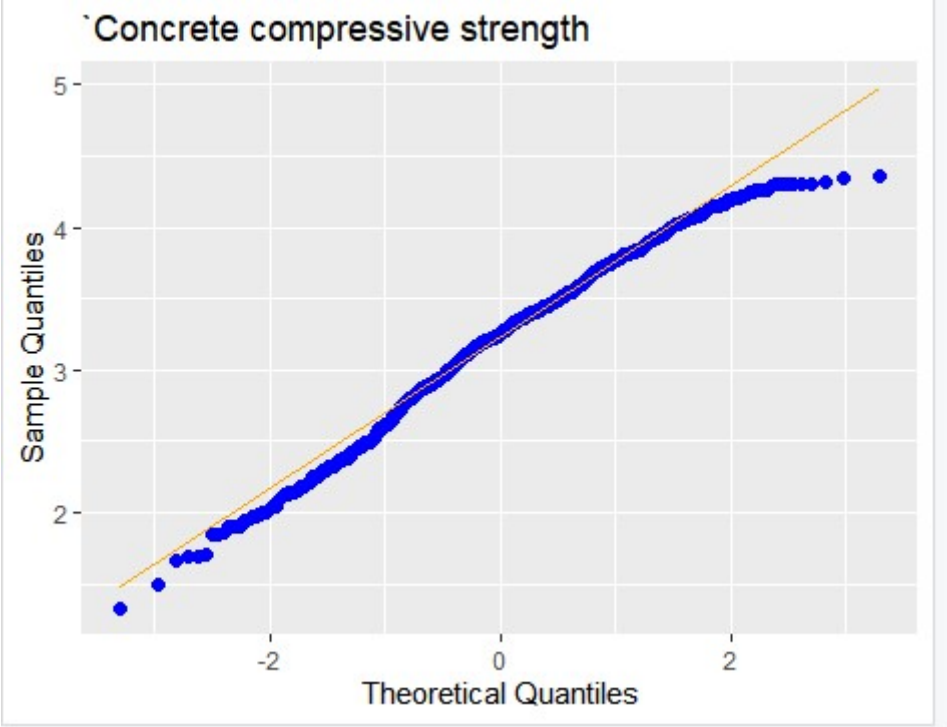
Now from this graph we can see that the data is normalize so I am going to apply square root transformation on data to make it normalize and visualize the transformed data using q-q plot.

```

# Apply square root transformation
df$`Square Root Concrete compressive strength(MPa, megapascals)` <-
  sqrt(df$`Concrete compressive strength(MPa, megapascals)`)

# Print the first few rows of the transformed dataframe to verify
head(df)


# Create the Q-Q plot using ggplot2
ggplot(data = data.frame(sample = df$`Square Root Concrete compressive strength(MPa, megapascals)`), aes(sample = sample)) +
  stat_qq(size = 2, color = "blue") +            # Q-Q points
  stat_qq_line(color = "orange") +               # Q-Q line
  xlab("Theoretical Quantiles") +                # X-axis label
  ylab("Sample Quantiles") +                     # Y-axis label
  ggtitle("`Concrete compressive strength")          # Title for the plot

```



From this visualization the I see that the data is again not normalize so I am going to apply cube root transformation on it to make it normalize and view the transformed data using q-q plot again.

```
398
399  # Apply cube root transformation
400  df$cube_root <-
401    df$`Concrete compressive strength(MPa, megapascals)`^(1/3)
402
403  # Print the first few rows of the transformed dataframe to verify
404  head(df)
405
406  # Create the Q-Q plot using ggplot2
407  ggplot(data = data.frame(sample = df$cube_root), aes(sample = sample)) +
408    stat_qq(size = 2, color = "blue") +            # Q-Q points
409    stat_qq_line(color = "orange") +               # Q-Q line
410    xlab("Theoretical Quantiles") +                # X-axis label
411    ylab("Sample Quantiles") +                     # Y-axis label
412    ggtitle("`Concrete compressive strength")          # Title for the plot
413
```



Now from above graph we can see that the data is again not normalize.

So I assume that the data is inherently not normalize and I will apply non-parametric tests on my data to justify my hypothesis.

Hypothesis 1:-

For this time I assumed myself a civil engineer who investigating whether the use of fly ash in different types of concrete (such as course or fine) is independent of the concrete type or if there is some connection.

To test this hypothesis, I formulate two hypotheses as

Null Hypothesis (H_0):

There is no connection between concrete category and contain fly ashes

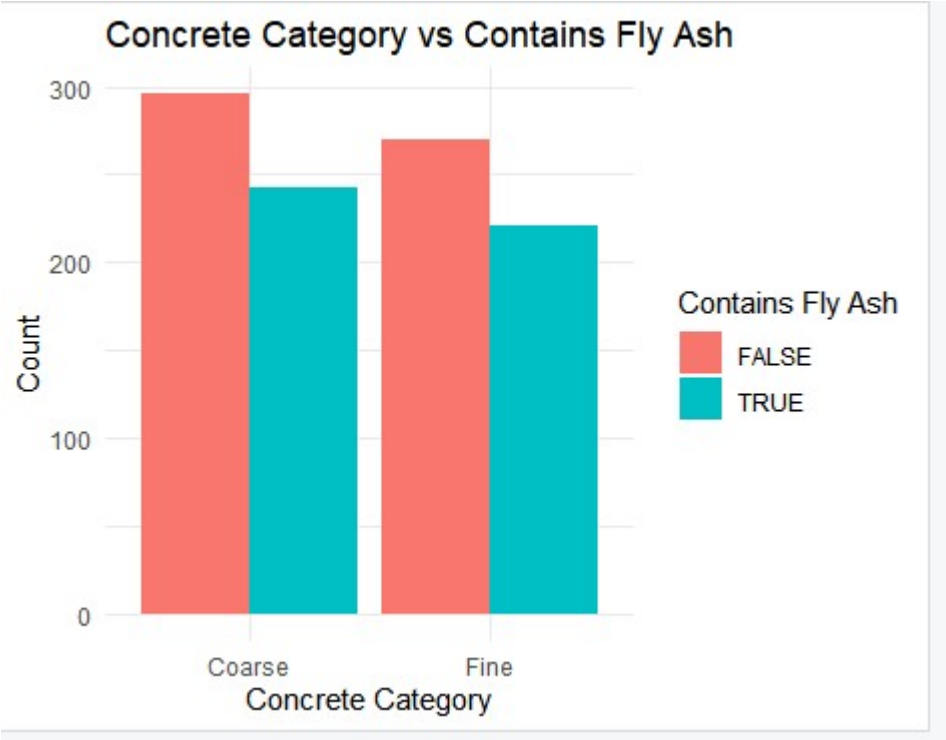
Alternative Hypothesis (H_1):

There is a connection between concrete category and contain fly ashes

So to test this hypothesis first of I all I visualize both variables as

1

```
427
428 # Create a bar plot using ggplot2
429 ggplot(df, aes(x = `Concrete Category`, fill = `Contains Fly Ash`)) +
430   geom_bar(position = "dodge") +
431   labs(
432     title = "Concrete Category vs Contains Fly Ash",
433     x = "Concrete Category",
434     y = "Count",
435     fill = "Contains Fly Ash"
436   ) +
437   theme_minimal()
438
```



Chi-square test of independence:-

Now I am going to apply this test to test my hypothesis as

```
438
439 #now apply chi-square test of independence
440 Contigency_table<-table(df$`Concrete Category`,df$`Contains Fly Ash`)
441 print(Contigency_table)
442 chisq.test(Contigency_table)
443
```

```
  Fine      270  221
> chisq.test(Contigency_table)

        Pearson's Chi-squared test with Yates' continuity correction

data:  Contigency_table
X-squared = 5.7212e-30, df = 1, p-value = 1
```

From above we can see that the value of is greater than 0.05 so we accept null hypothesis that there is no relation between concrete category and fly ash.

Hypothesis 2:-

For this time a civil engineer is investigating whether the concrete compressive strength depends on concrete category or not.

To test this hypothesis, I formulate two hypotheses as

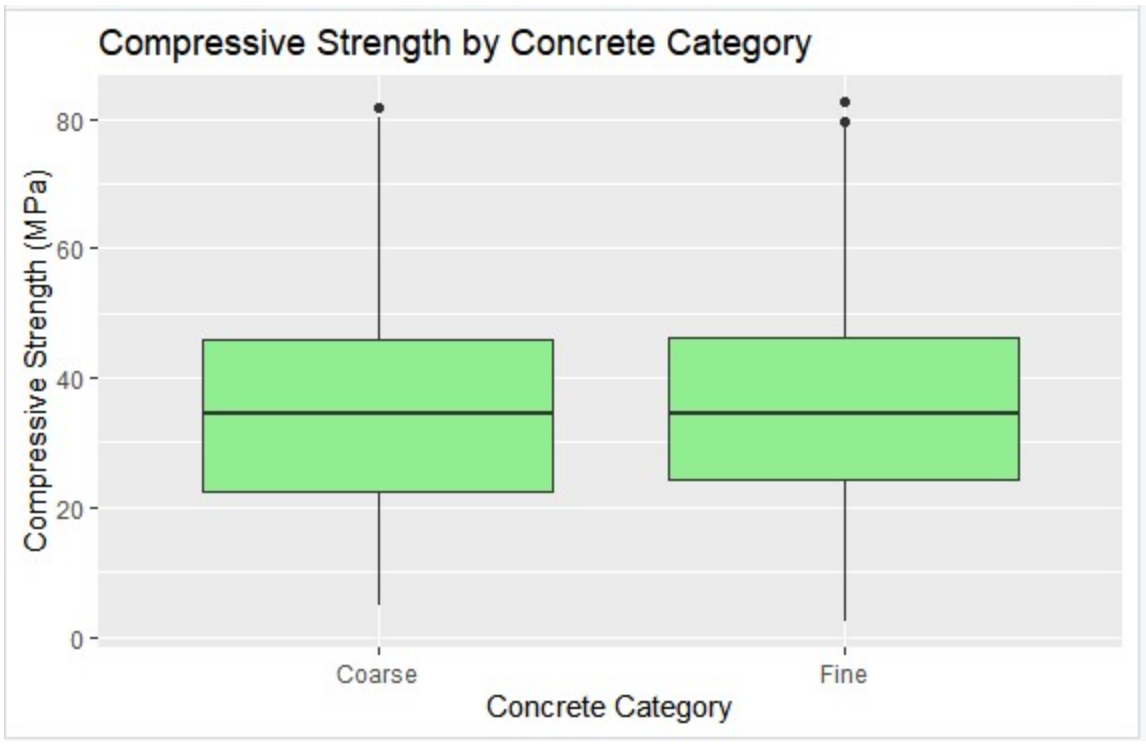
Null Hypothesis (H_0):

There is no significant difference in the mean Compressive Strength between the different Concrete Categories

Alternative Hypothesis (H_1):

There is a significant difference in the mean Compressive Strength between at least one pair of Concrete Categories

So to test this hypothesis first of I all I visualize both variables as



kruskal_test:-

As the dataset is not normal so I am using non-parametric tests for better results.

```
462
463 # kruskal_test (Concrete Category vs. Compressive Strength)
464 kruskal_test_result <- kruskal.test(`Concrete compressive strength(MPa, megapascals)` ~ `Concrete Category`, data = df)
465 print(kruskal_test_result)
466
```

```
> print(kruskal_test_result)

        Kruskal-Wallis rank sum test

data:  Concrete compressive strength(MPa, megapascals) by Concrete Category
Kruskal-Wallis chi-squared = 0.92415, df = 1, p-value = 0.3364
```

From this result we can see that the value of p is greater than 0.05 so we will not reject the null hypothiesis that there is no significant difference in the mean Compressive Strength between the different Concrete Categories

Hypothesis 3:-

For this time a civil engineer is investigating relationship between the water-cement ratio and the compressive strength of concrete. In general it should be inverse in relationship. So engineer wants to investigate it.

To test this hypothesis, I formulate two hypotheses as

1

Null Hypothesis (H_0):

There is no significant relationship between the water-cement ratio and the compressive strength of concrete.

Alternative Hypothesis (H_1):

There is a significant inverse relationship between the water-cement ratio and the compressive strength of concrete.

Spearman's rank correlation test:-

As the data is not normal and to find above discussed kind of relationship we used spearmans rank correlation test.

```
#hypothesis 3
#Null Hypothesis (H0): There is no significant relationship between the water-cement ratio and the compressive strength of concrete.
#Alternative Hypothesis (H1): There is a significant inverse relationship between the water-cement ratio and the compressive strength of concrete.

# Calculate water-cement ratio (Water / Cement)
df$water_cement_ratio <- df$`Water  (component 4)(kg in a m^3 mixture)` / df$`Cement (component 1)(kg in a m^3 mixture)`

# Perform Spearman's rank correlation test between Water-Cement Ratio and Concrete Compressive Strength
cor_test_water_cement <- cor.test(df$water_cement_ratio, df$`Concrete compressive strength(MPa, megapascals)`, method = "spearman")

# Print the result
print(cor_test_water_cement)
```

```
> # Print the result
> print(cor_test_water_cement)

        Spearman's rank correlation rho

data:  df$water_cement_ratio and df$`Concrete compressive strength(MPa, megapascals)`
S = 277169381, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho
-0.5218969
```

From the output we can see that the rho=-0.521 which is less than 0.05 so we reject the null hypothesis and accept the alternative hypothesis which states that there is a significant inverse relationship.

Task 3: Time Series Modelling (20 marks)

For this task, you should imagine you are a Data Scientist who has been asked to conduct time series modelling for your chosen scenario.
Your report should:
• Briefly introduce your understanding of the chosen scenario and your task
• Provide initial exploration of the data, e.g., visualisation of the time series data, decomposition into trend and seasonal components, etc.
• Present all steps of your time series modelling. This should include screenshots of both your code and your output as well as any data visualisations. You should also explain why you have chosen to use the techniques you have chosen.
• Evidence that you have properly tested any assumptions relating to the time series
modelling techniques you have used.
• Interpret and explain the output generated from your analysis.
• Finally, you should briefly conclude by summarising the main findings from your time
series analysis, including a comparison of the models and a recommendation on which is better suited to the data.

Answer:-

Introduction:-

I have choosed the data set of UK Vital Statistics and my analysis is specifically for live births in England and Wales. The objective is to analyze the meaning full trend, forecast future value and determine a model for forecasting. I assumed the senerio that the data set records annual live births for UK and Wales over several decades. The data set includes

Years:- the years of observation.

Live births:- The number of live births recorded in England and wales.

Exploration of data:-

First of all I loaded all necessary libraries

```r
# Load required libraries
install.packages("readxl")
library(readxl)
library(ggplot2)
library(forecast)
library(tseries)
install.packages("TTR")
install.packages("forecast")
library(TTR)
library(forecast)
install.packages("corrplot")
library(corrplot)
```

Then in the excel file there is some written material on the top of data set I removed that written material in excel and then load the file into R and then preview the data set

```r

# Load the Excel file
data <- read_excel("C:/Users/Chaudhary Computer/Downloads/Vital statistics in the UK.xlsx", sheet = "Birth")

# Preview the data to find the starting row for relevant data
print(head(data, 15))
```

```
   Year  Number of live births: Un…¹ Number of live birth…² Number of live birth…³ Number of live birth…⁴ Number of live birt
   <chr> <chr>                                       <dbl> <chr>                  <chr>                  <chr>
 1 2021  694685                                     624828 595948                 28781                  47786
 2 2020  681560                                     613936 585195                 28638                  46809
 3 2019  712680                                     640370 610505                 29704                  49863
 4 2018  731213                                     657076 625651                 31274                  51308
 5 2017  755042                                     679106 646794                 32176                  52861
 6 2016  774835                                     696271 663157                 32936                  54488
 7 2015  777165                                     697852 664399                 33279                  55098
 8 2014  776352                                     695233 661496                 33544                  56725
 9 2013  778803                                     698512 664517                 33747                  56014
10 2012  812970                                     729674 694241                 35238                  58027
11 2011  807776                                     723913 688120                 35598                  58590
12 2010  807271                                     723165 687007                 35952                  58791
13 2009  790204                                     706248 671058                 34937                  59046
14 2008  794383                                     708711 672809                 35650                  60041
15 2007  772245                                     690013 655357                 34414                  57781
# ℹ abbreviated names: ¹`Number of live births: United Kingdom`, ²`Number of live births: England and Wales`,
```

Then I check the data types

```r

#Now check the variables and data types for each variable.
str(data)

```

```
> str(data)
tibble [172 × 18] (S3: tbl_df/tbl/data.frame)
 $ Year                                  : chr [1:172] "2021" "2020" "2019" "2018" ...
 $ Number of live births: United Kingdom : chr [1:172] "694685" "681560" "712680" "731213" ...
 $ Number of live births: England and Wales: num [1:172] 624828 613936 640370 657076 679106 ...
 $ Number of live births: England        : chr [1:172] "595948" "585195" "610505" "625651" ...
 $ Number of live births: Wales          : chr [1:172] "28781" "28638" "29704" "31274" ...
 $ Number of live births: Scotland       : chr [1:172] "47786" "46809" "49863" "51308" ...
 $ Number of live births: Northern Ireland : chr [1:172] "22071" "20815" "22447" "22829" ...
 $ Total fertility rate: United Kingdom  : chr [1:172] "1.53" "1.56" "1.63" "1.68" ...
 $ Total fertility rate: England and Wales : chr [1:172] "1.55" "1.58" "1.65" "1.7" ...
 $ Total fertility rate: England         : chr [1:172] "1.55" "1.59" "1.66" "1.7" ...
 $ Total fertility rate: Wales           : chr [1:172] "1.5" "1.47" "1.54" "1.63" ...
 $ Total fertility rate: Scotland        : chr [1:172] "1.31" "1.29" "1.37" "1.42" ...
 $ Total fertility rate: Northern Ireland  : chr [1:172] "1.81" "1.71" "1.82" "1.85" ...
 $ ...14                                 : logi [1:172] NA NA NA NA NA NA ...
 $ ...15                                 : logi [1:172] NA NA NA NA NA NA ...
 $ ...16                                 : logi [1:172] NA NA NA NA NA NA ...
 $ ...17                                 : logi [1:172] NA NA NA NA NA NA ...
 $ ...18                                 : num [1:172] NA NA NA NA NA NA 1 NA NA NA ...
>
```

Then I extracted only the relevant columns of my senerio and preview them

```
22
23  # Extract the relevant columns
24  cleaned_dataBEW <- data[, c("Year", "Number of live births: England and Wales")]
25
26  # Preview the extracted data
27  head(cleaned_dataBEW)
28
```

```
# A tibble: 6 × 2
  Year  `Number of live births: England and Wales`
  <chr>                                      <dbl>
1 2021                                      624828
2 2020                                      613936
3 2019                                      640370
4 2018                                      657076
5 2017                                      679106
6 2016                                      696271
>
```

Then I remove the empty rows and preview the data

```
32
33  # Remove rows with NA
34  cleaned_dataBEW <- na.omit(cleaned_dataBEW)
35
36  # Preview cleaned data
37  head(cleaned_dataBEW)
38
```

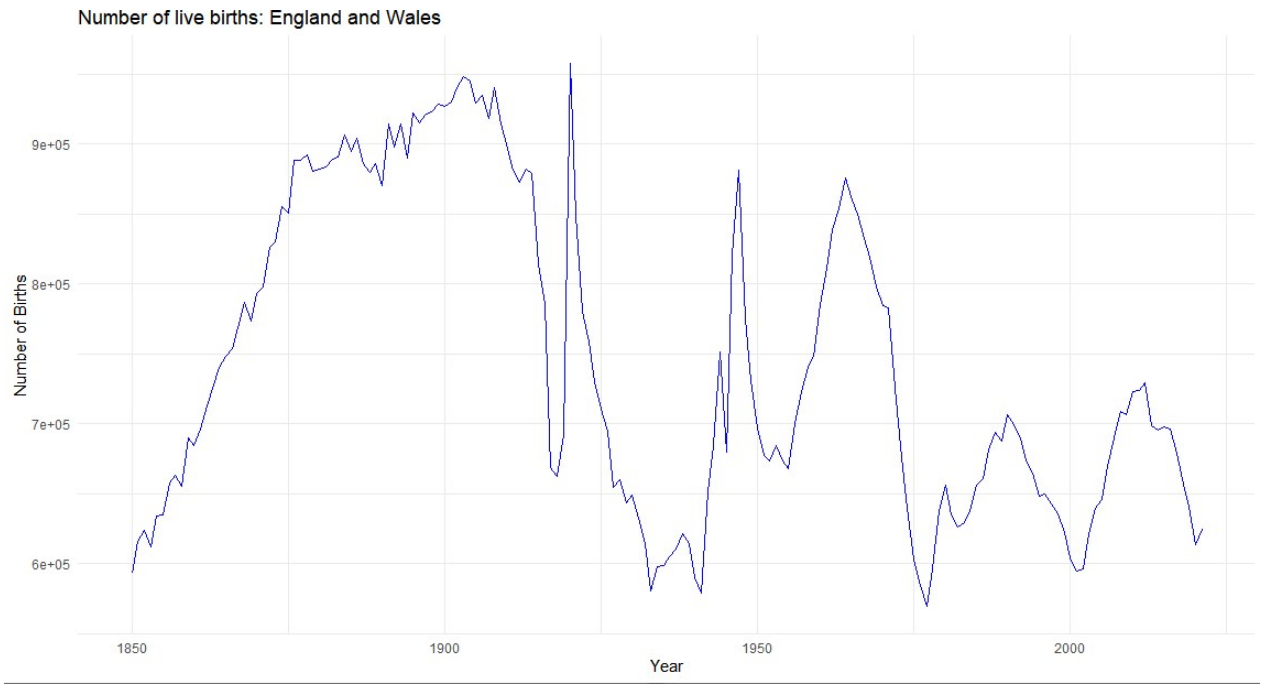
Then I convert the columns into numeric

```
30
31  # Convert columns to numeric
32  cleaned_dataBEW$Year <- as.numeric(cleaned_dataBEW$Year)
33  cleaned_dataBEW$`Number of live births: England and Wales` <- as.numeric(cleaned_dataBEW$`Number of live births: England and Wales`)
34
```

1

Then I plot the data to see the trend

```
43
44  # Plot the time series data
45  ggplot(cleaned_dataBEW, aes(x = Year, y = `Number of live births: England and Wales`)) +
46    geom_line(color = "blue") +
47    labs(title = "Number of live births: England and Wales",
48         x = "Year",
49         y = "Number of Births") +
50    theme_minimal()
51
```



Plotting time series:-

Now to plot a time series, first of all I convert the data into time series object.

```
52
53
54
55  # Convert to a time series object
56  births_tsEW <- ts(cleaned_dataBEW$`Number of live births: England and Wales`, start = min(cleaned_dataBEW$Year), frequency = 1)
57  #again checking the data
58  births_tsEW
59
60
```

Then I plot the time series data as

```
60
61  # Plot the time series
62  plot(births_tsEW, main = "Number of live births: England and Wales", ylab = "Births", xlab = "Year")
63
64
65
```

1

From above graph you can see that it is not a seasonal time series because you can't see any repeating pattern or fluctuations with in fixed intervals. It shows long term trends like if you see in graph there is a trend after 1945 than increase than decrease. It represents potential irregular variations as you can see as well.

Decomposing time series:-

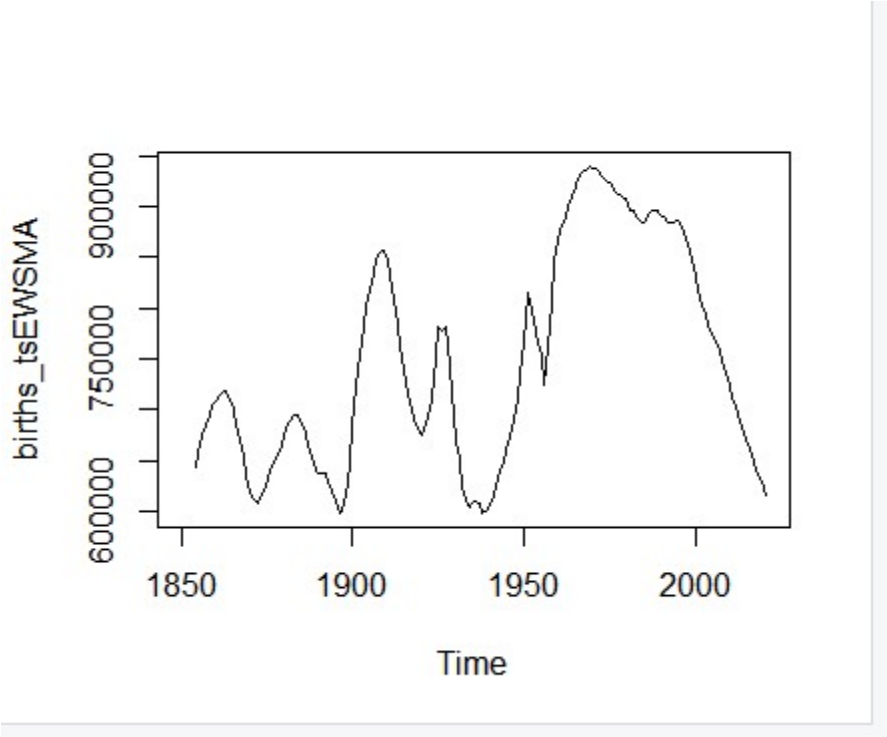
As our data is non-seasonal and non-seasonal data always include a trend component and irregular component. So I am going to decompose the time series data into its components.

For this purpose I am going to use smoothing method.

First I plot for n=5

```
65
66  # smoothing
67  births_tsEWSMA <- SMA(births_tsEW,n=5)
68  plot.ts(births_tsEWSMA)
69
70
```

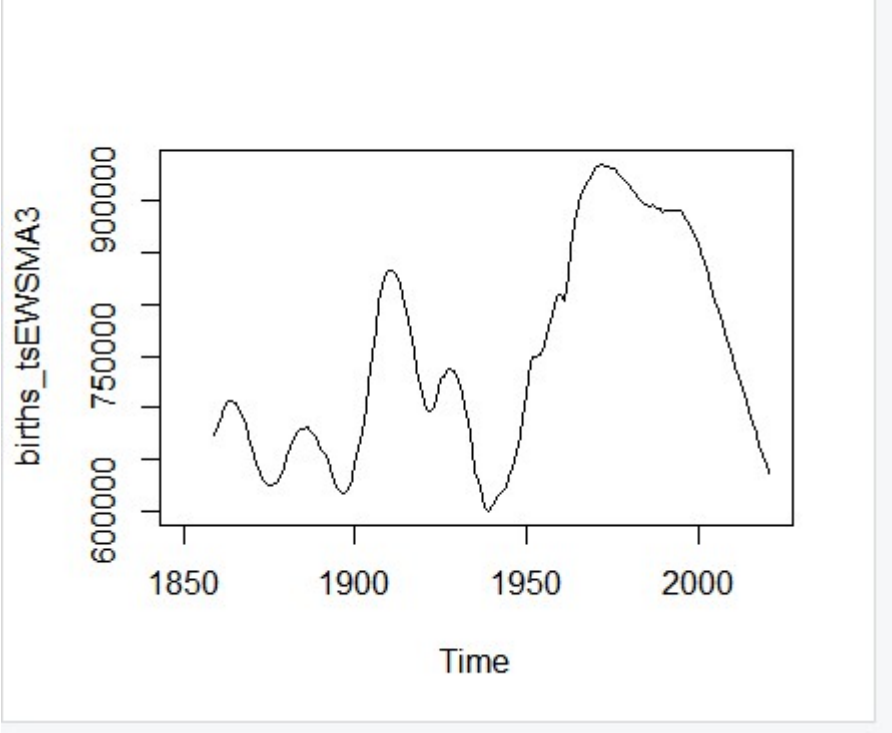
1



From above graph we can see that there is a lot of fluctions so we need to increase smooth the data using higher order.

For n=10

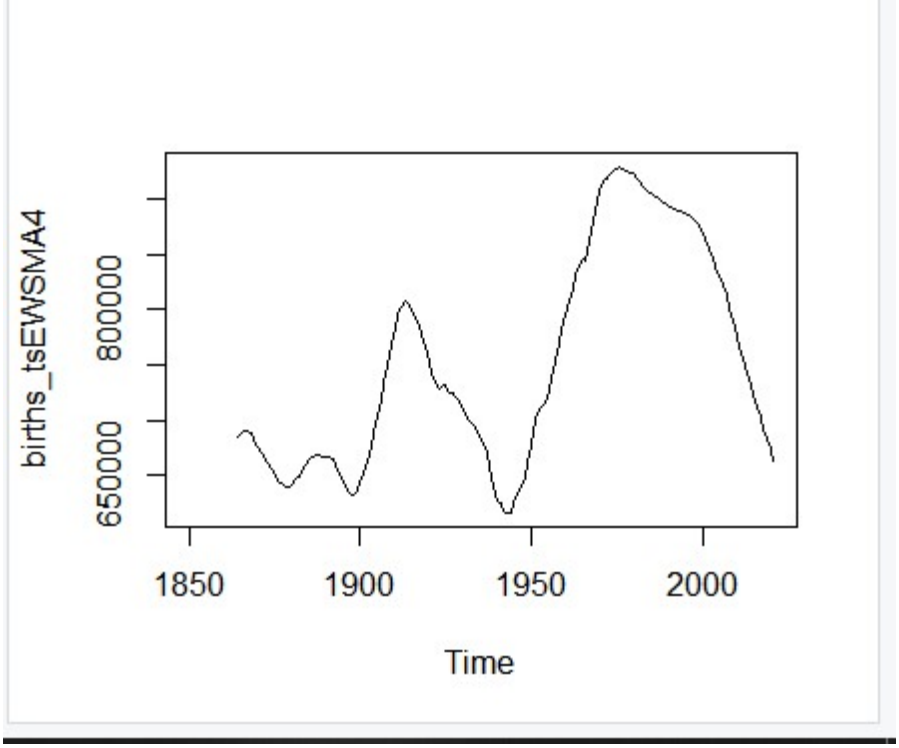
```
72
73
74  births_tsEWSMA3 <- SMA(births_tsEW,n=10)
75  plot.ts(births_tsEWSMA3)
76
77
78
```



Again a lot of fluctuations.

Now for n=15

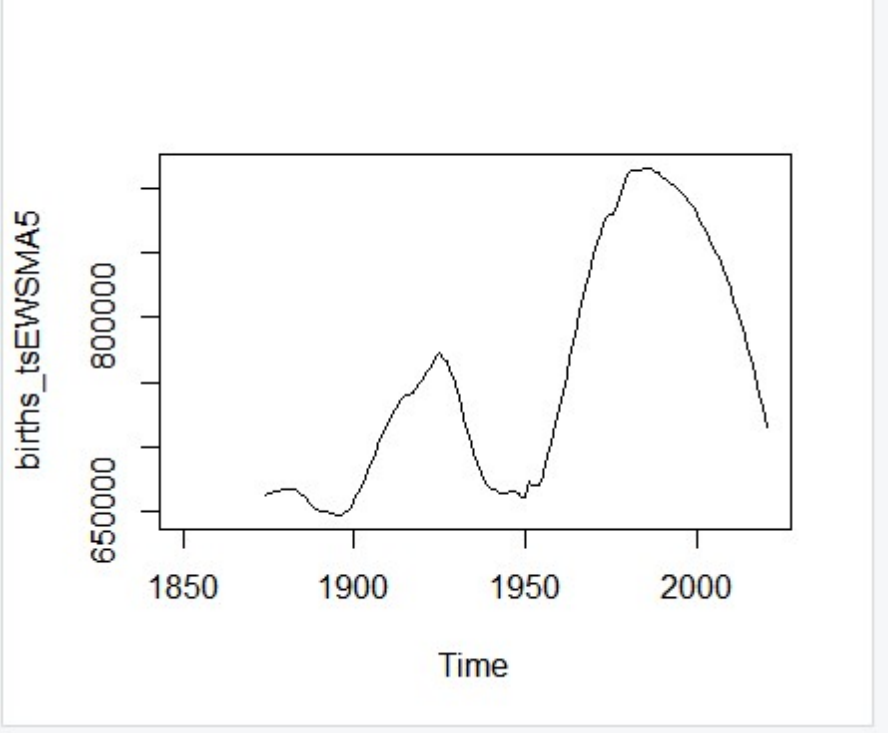
```
77
78
79  births_tsEWSMA4 <- SMA(births_tsEW,n=15)
80  plot.ts(births_tsEWSMA4)
81
82
83
```



There are a little bit fluctuations so lets try higher order to remove it as well.

For n=25

```
82
83
84  births_tsEWSMA5 <- SMA(births_tsEW,n=25)
85  plot.ts(births_tsEWSMA5)
86
```



This a clear picture of live births. We can see that the number of births increase gradually from 1900 followed by a slower growth than after 1945 there is a boom of live births and after 2000 there is a decrease in live births.

Forecasts using smoothing:-

As from smoothing we have seen that our time series model is increasing or decreasing with non-seasonality. So for this case I am applying Holt's exponential smoothing using HoltWinters() function to make short-term forecasts.

```
88
89  #forecasts
90  births_tsEWforecasts <- HoltWinters(births_tsEW, gamma=FALSE)
91  births_tsEWforecasts
92
```

```
> births_tsEWforecasts <- HoltWinters(births_tsEW, gamma=FALSE)
> births_tsEWforecasts
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = births_tsEW, gamma = FALSE)

Smoothing parameters:
 alpha: 1
 beta : 0.02332378
 gamma: FALSE

Coefficients:
        [,1]
a 593422.000
b  -4365.199
```

1

From above we can see that the value of alpha is 1 which means weight is given to the most recent observation when calculating the level of time series.

The value of B=0.02333 means that trend is updated more gradually and is not overly influenced by recent changes.

So this means if we forecast this model than the projection will reflect negative trend but will adjust gradually.

Now I will calculate the SSE for this model.

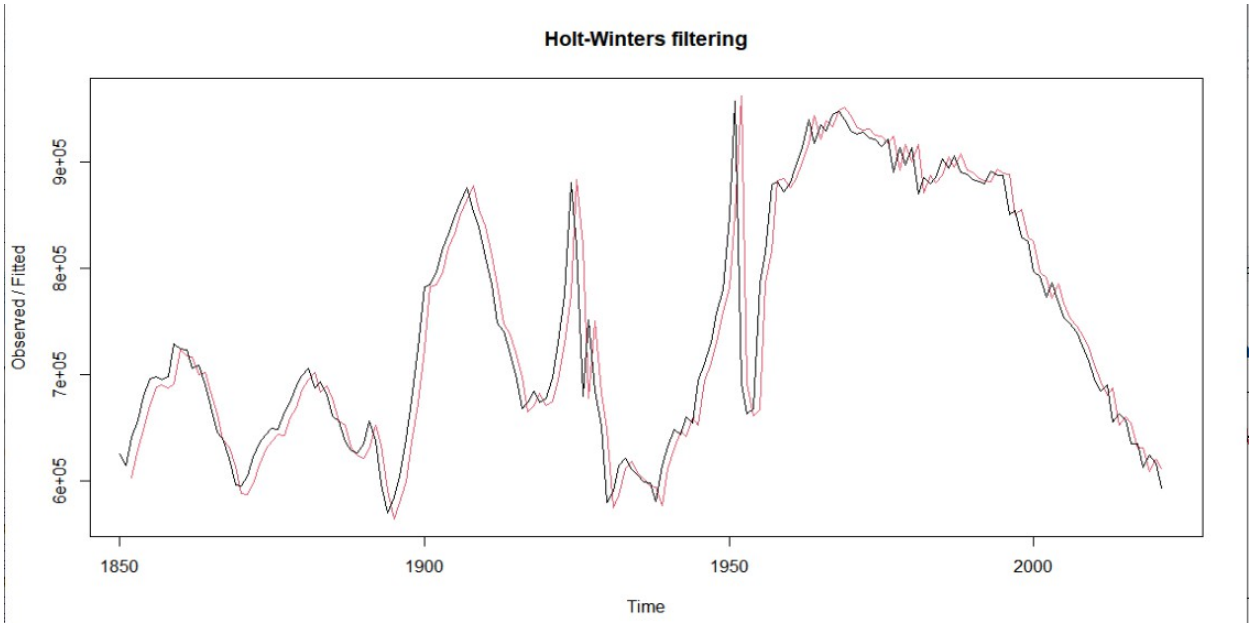
```
92
93  #fore cast error
94  births_tsEWforecasts$SSE
95
```

```
> #fore cast error
> births_tsEWforecasts$SSE
[1] 218205548229
>
```

As the number of data set is too large so SSE for this model is acceptable.

Then I plotted the original time series as black and forecasted time series as red

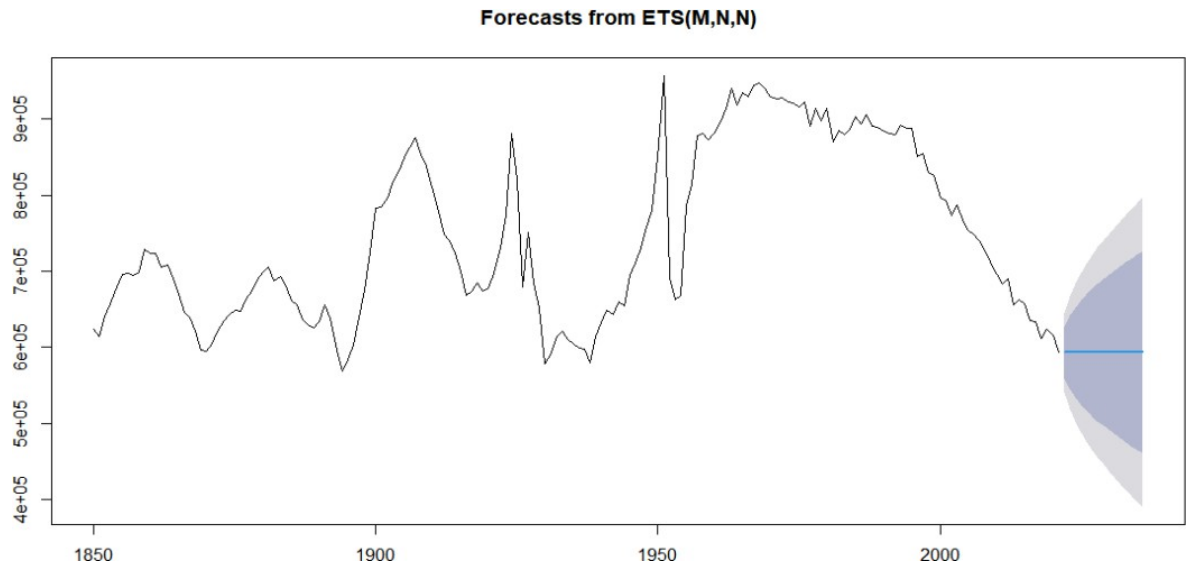
```
95
96  #plot
97  plot(births_tsEWforecasts)
98
99
```



1

Then I plotted the forecast plot for next h=15 values

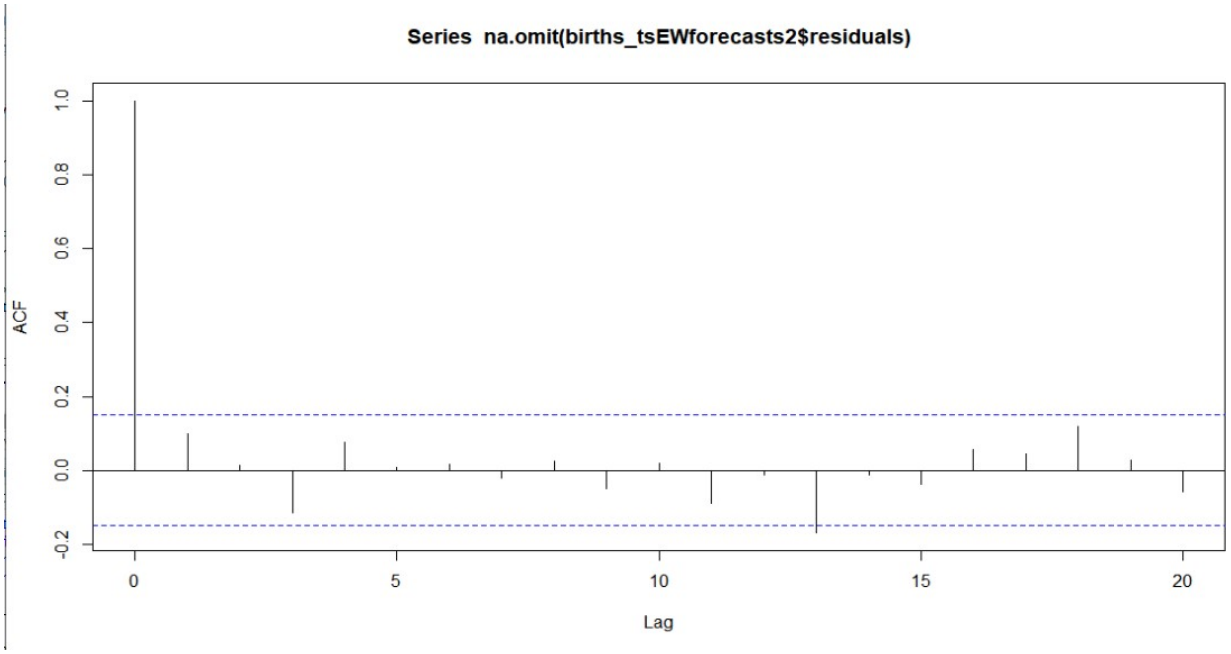
```
102
103
104 births_tsEWforecasts2 <- forecast(births_tsEW, h=15)
105 plot(births_tsEWforecasts2)
106
107
```



The forecasts graph is shown as a blue line, have 80% prediction intervals as an purple shaded area, and the 95% prediction intervals as a gray shaded area.

Now I am going to plot correlogram check whether the predictive model can be improved by checking whether the insample forecast error shows non-zero auto correlation at lag 1-20.

```
107
108 # ACF plot of residuals
109 acf(na.omit(births_tsEWforecasts2$residuals), lag.max=20)
110
111 # Box-Ljung test on residuals for lags up to 20
112 Box.test(na.omit(births_tsEWforecasts2$residuals), lag=20, type="Ljung-Box")
113
114
```




```
> # Box-Ljung test on residuals for lags up to 20
> Box.test(na.omit(births_tsEWforecasts2$residuals), lag=20, type="Ljung-Box")

        Box-Ljung test

data:  na.omit(births_tsEWforecasts2$residuals)
X-squared = 17.539, df = 20, p-value = 0.6177

>
```

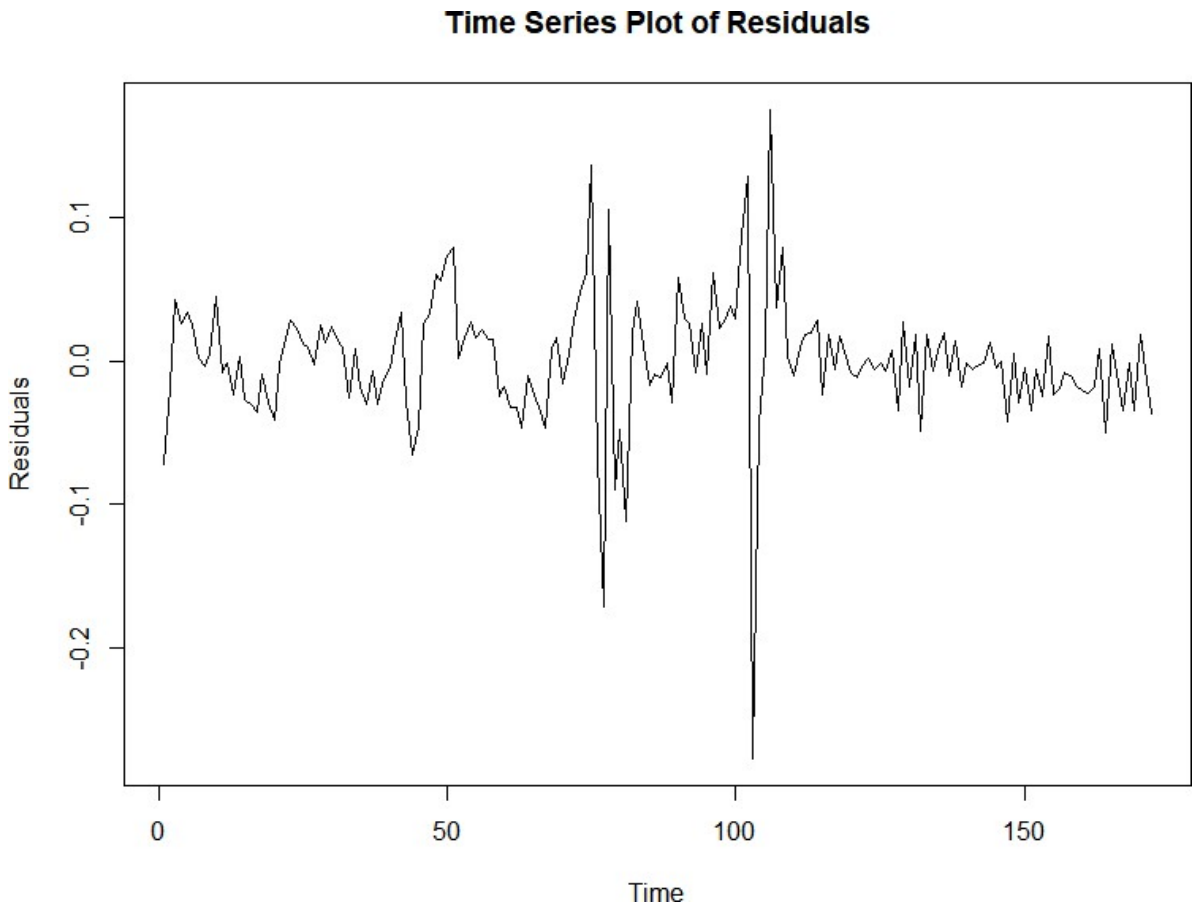
From above we can see that autocorrelation for the in sample forecast errors lag at almost 13 for the significance bounds, however we can expect one out of 20 to exceed its normal.

When we see the output of Ljung-box test, the value of p is 0.6177 which indicates that there is a little evidence of non-zero autocorrelation in the insample forecast error at lag 1-20.

Now I am going to check that forecast errors have constant variance over time and normally distributed with mean zero. For this purpose I will plot the forecast error which will shows the histogram of distribution of forecast error with an overlaid normal curve

```
# Remove NA values from residuals
births_tsEWforecasts2$residuals <- births_tsEWforecasts2$residuals[!is.na(births_tsEWforecasts2$residuals)]


# Plot time series of residuals
plot.ts(births_tsEWforecasts2$residuals, main="Time Series Plot of Residuals", ylab="Residuals", xlab="Time")

```

1



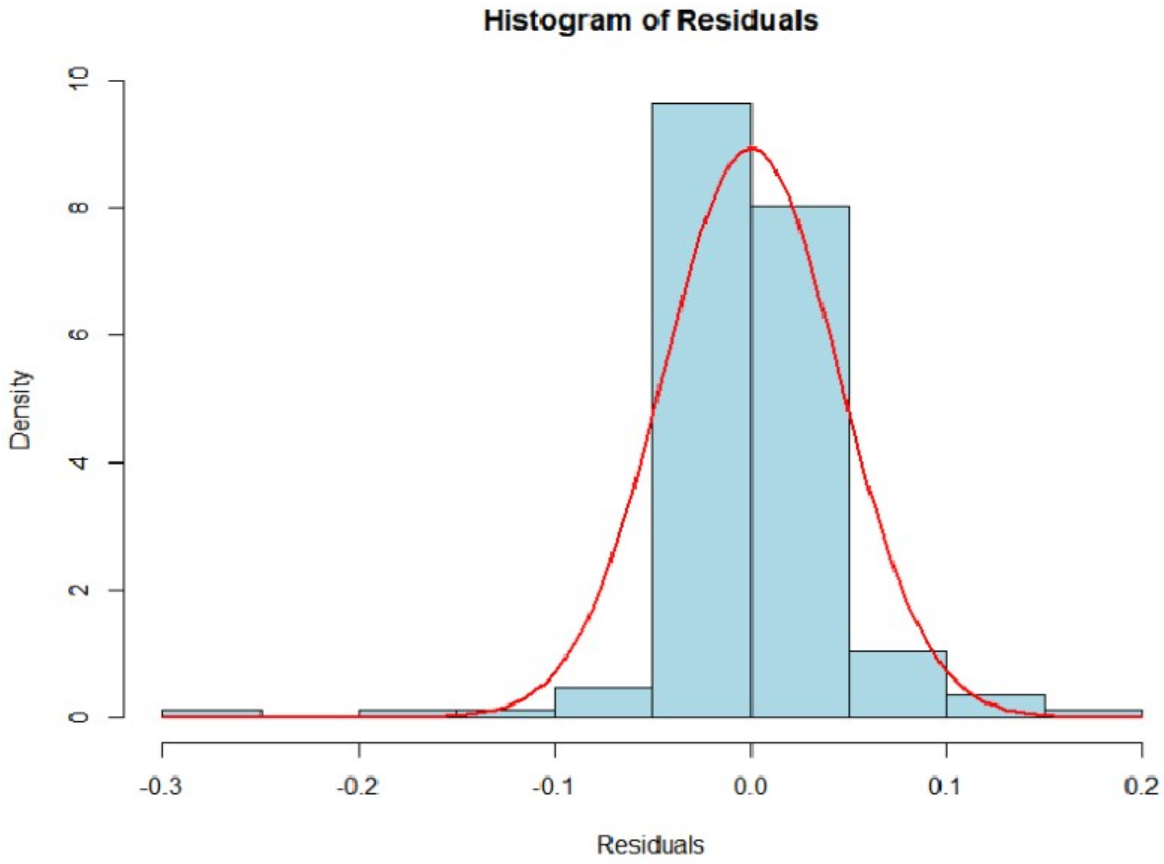
From above you can see that there is a constant variance over time but there is spike at 100, which can cause heteroscedasticity but I performed studentized Breusch-Pagan test to verify that that heteroscedasticity exist or not.

```
120
121 residuals <- residuals(births_tsEWforecasts2)
122
123 time <- 1:length(residuals)
124 model <- lm(residuals ~ time)
125
126
127 library(lmtest)
128 bptest(model)
129
130
> bptest(model)

        studentized Breusch-Pagan test

data:  model
BP = 0.0038306, df = 1, p-value = 0.9506

>
```

1

From above we can see that value of BP=0.003 which mean there is no heteroscedasticity.
Then I plot a histogram of residual error to check the normality.

```
134  # Create a histogram of the residuals
135  hist(births_tsEWforecasts2$residuals,
136       main = "Histogram of Residuals",
137       xlab = "Residuals",
138       col = "lightblue",
139       border = "black",
140       freq = FALSE)  # freq = FALSE ensures the y-axis is density
141
142  # Add a normal distribution curve
143  curve(dnorm(x,
144              mean = mean(births_tsEWforecasts2$residuals, na.rm = TRUE),
145              sd = sd(births_tsEWforecasts2$residuals, na.rm = TRUE)),
146        add = TRUE,
147        col = "red",
148        lwd = 2)
149
150
```



From above we can see that histogram of forecast error show that forecast errors are normally distributed and with mean zero and constant variance.

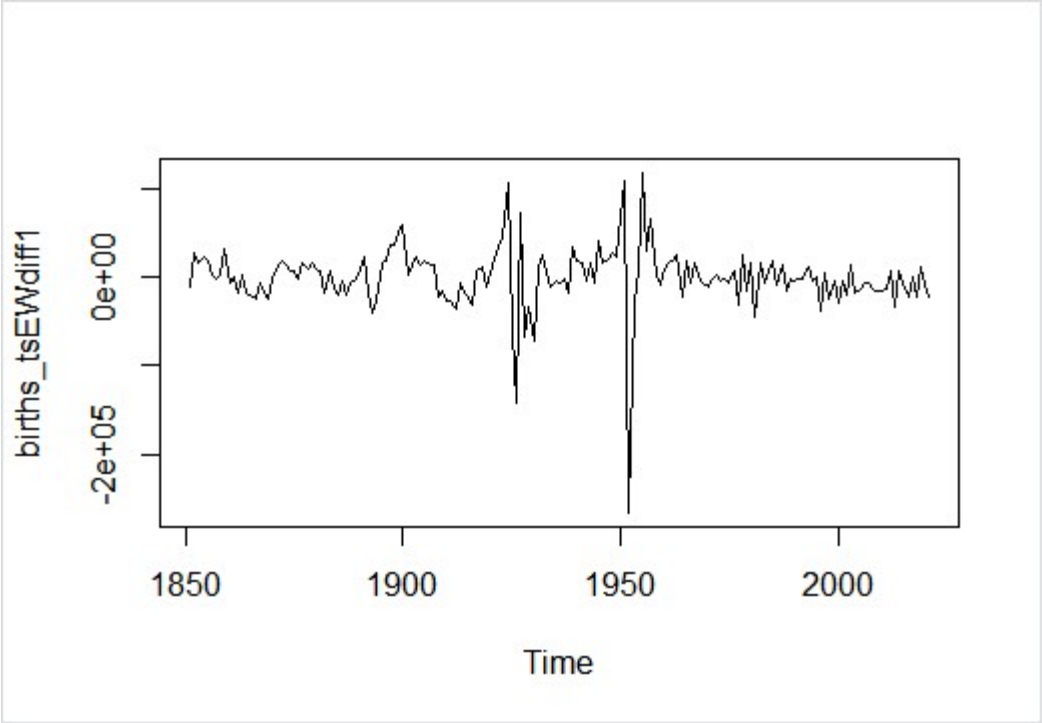
ARIMA MODEL

Differencing a time series:-

Now I am going to apply Arima model on my dataset for this purpose first of all I have to ensure that time series model should be stationary in mean for this purpose I have to take the difference in the time series model of the data set.

First I will apply the difference of 1.

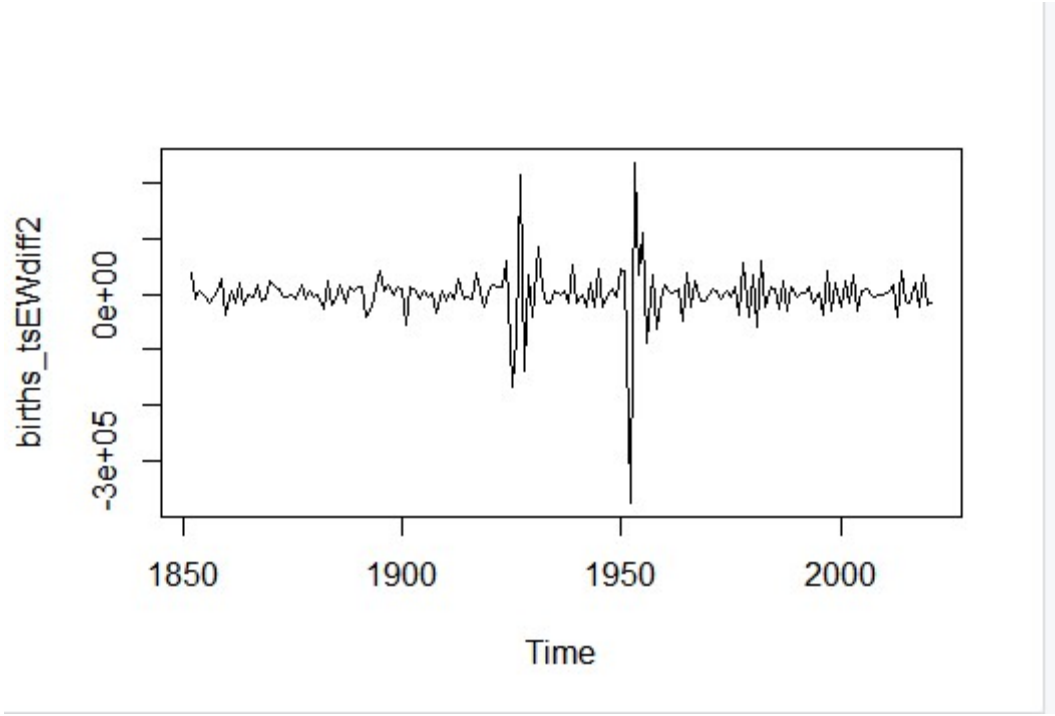
```
155 #difference the time series
156 births_tsEWdiff1 <- diff(births_tsEW, differences=1)
157 plot.ts(births_tsEWdiff1)
158
159
```



From above we can see that graph is not stationary at mean so we need to increase the difference.

Now for diff=2

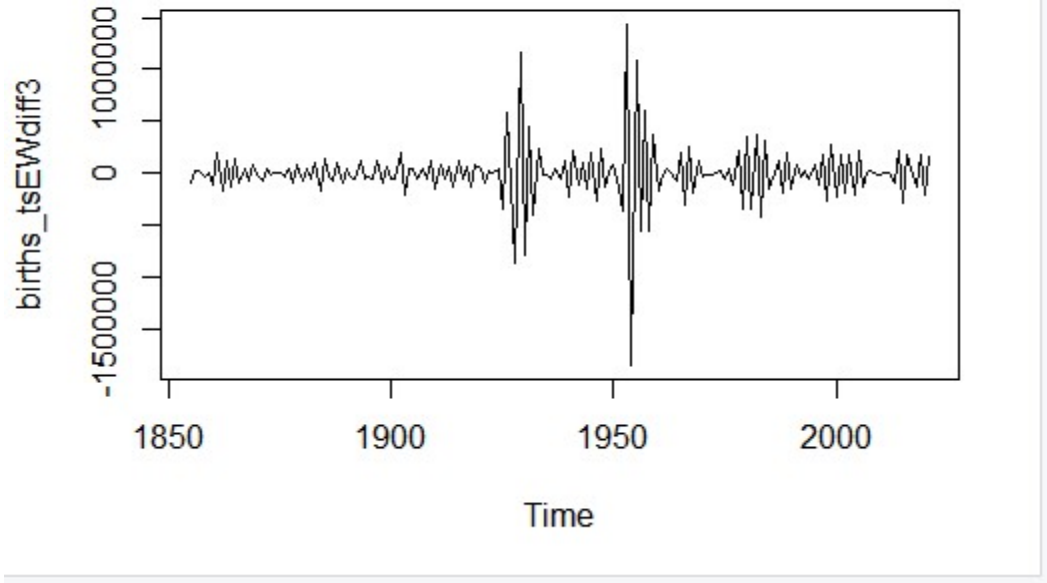
```
160
161 #diff=2
162 births_tsEWdiff2 <- diff(births_tsEW, differences=2)
163 plot.ts(births_tsEWdiff2)
164
```



Again there is a need to increase the difference level.

Now for diff=5

```
165
166  births_tsEWdiff3 <- diff(births_tsEW, differences=5)
167  plot.ts(births_tsEWdiff3)
168
```



1

Now from above we can see that now roughly it is stationary in mean and variance so we don't need to increase the difference further.
Then I apply Augmented Dickey-Fuller test to test whether the model is stationary or not.

```
174
175 adf_test <- adf.test(births_tsEWdiff1)
176 print(adf_test)
177
178
179
```

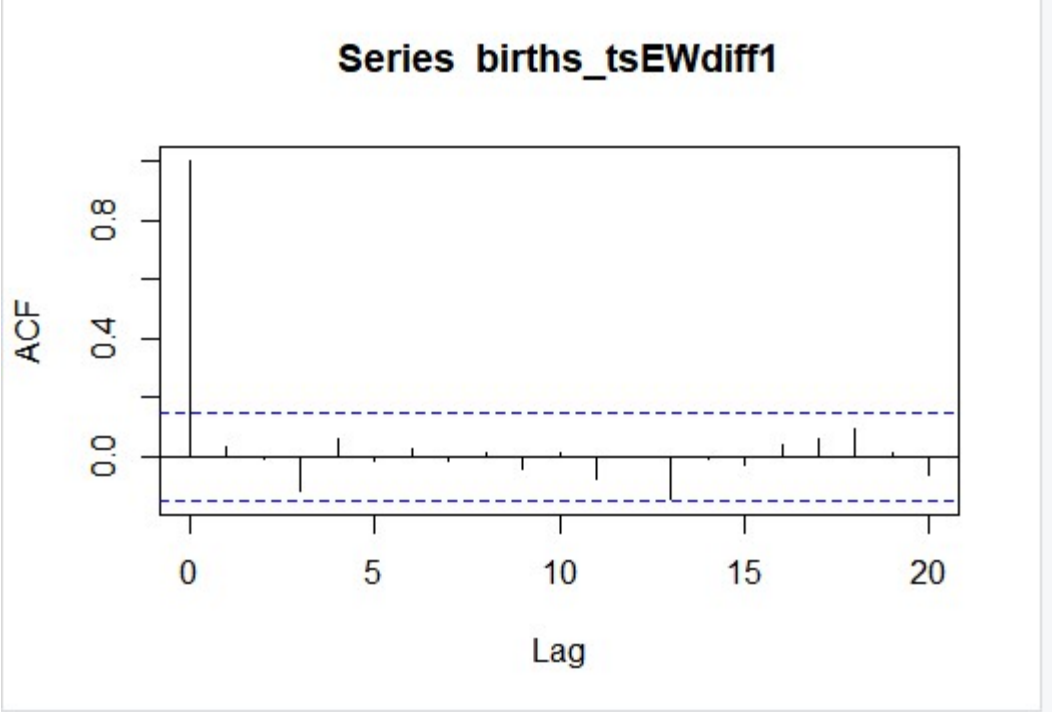
```
        Augmented Dickey-Fuller Test

data:  births_tsEWdiff1
Dickey-Fuller = -5.2987, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary

>
```

From above we came to know that for diff=1 our model is stationary so we don't need to consider the diff of higher orders.

Selecting a candidate for ARIMA Model:-
No our time series is stationary as we conformed from above so now we are going to predict the value of p and q for our arima model. For this purpose I am going to plot correlogram and partial correlogram of my stationary time series

ACF:-
Now to find the value of q I plotted ACF

```
179 #for acf
180 acf(births_tsEWdiff1, lag.max=20) # plot a correlogram
181 acf(births_tsEWdiff1, lag.max=20, plot=FALSE)
182
```

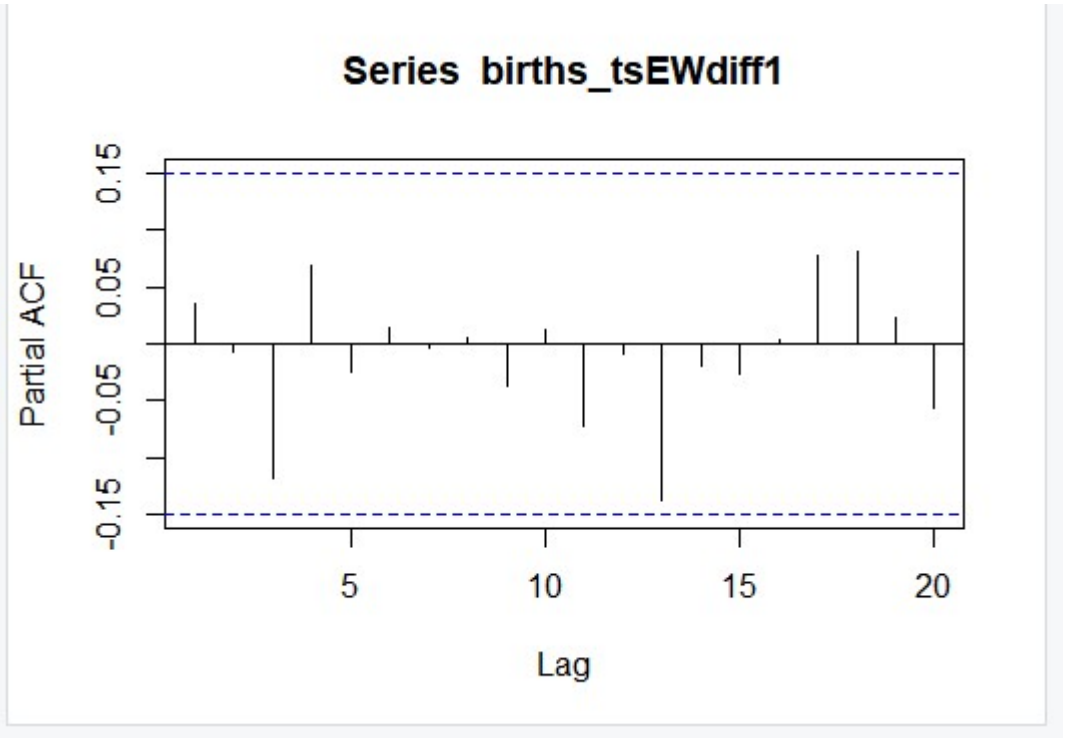


Now from above we can see that there is spike at 1 which means the value of q=1.

PCF:-

Now to determine value of p I plotted PCF

```
182
183 #for PCF
184 pacf(births_tsEWdiff1, lag.max=20) # plot a correlogram
185 pacf(births_tsEWdiff1, lag.max=20, plot=FALSE)
186
187
```



From above graph we can see that the value for q=0.

Auto.arima:-

```
187
188  auto.arima(births_tsEW)
189
190
```

```
> auto.arima(births_tsEW)
Series: births_tsEW
ARIMA(0,1,0)

sigma^2 = 1.238e+09:  log likelihood = -2032.71
AIC=4067.42   AICc=4067.44   BIC=4070.56
>
```

Forecasting using ARIMA Model:-

I will consider the model arima (0,1,1)

As we selected the parameters for arima model now I am going to estimate the parameters for that arima model.

```
190
191  #Forecasting Using an ARIMA Model
192  births_tsEWarima <- arima(births_tsEW, order=c(0,1,1)) # fit an ARIMA(0,1,1) model
193  births_tsEWarima
194
195
```

1

```
> births_tsEWarima

Call:
arima(x = births_tsEW, order = c(0, 1, 1))

Coefficients:
          ma1
      0.0342
s.e.  0.0759

sigma^2 estimated as 1.236e+09:  log likelihood = -2032.61,  aic = 4069.22
>
```

Here we are estimating the parameters for diff=1 and q=0 and p=1 which can also be written as X_t - mu = Z_t -(theta * Z_t-1) where theta is estimated in above output which is 0.0342.

Specifying the confidence level for prediction intervals:-

```
195
196  #Specifying the confidence level for prediction intervals
197  births_tsEWforecasts <- forecast(births_tsEWarima, h=5, level=c(99.5))
198  births_tsEWforecasts
199
```

```
> births_tsEWforecasts
     Point Forecast   Lo 99.5   Hi 99.5
2022       592664.3 493969.1 691359.5
2023       592664.3 450680.8 734647.8
2024       592664.3 417798.9 767529.6
2025       592664.3 390188.4 795140.1
2026       592664.3 365915.4 819413.1
>
```

The above graph gives as a forecast for the live births as well as 99.5% perdiction interval for those prediction.

Without Specifying the confidence level for prediction intervals:-

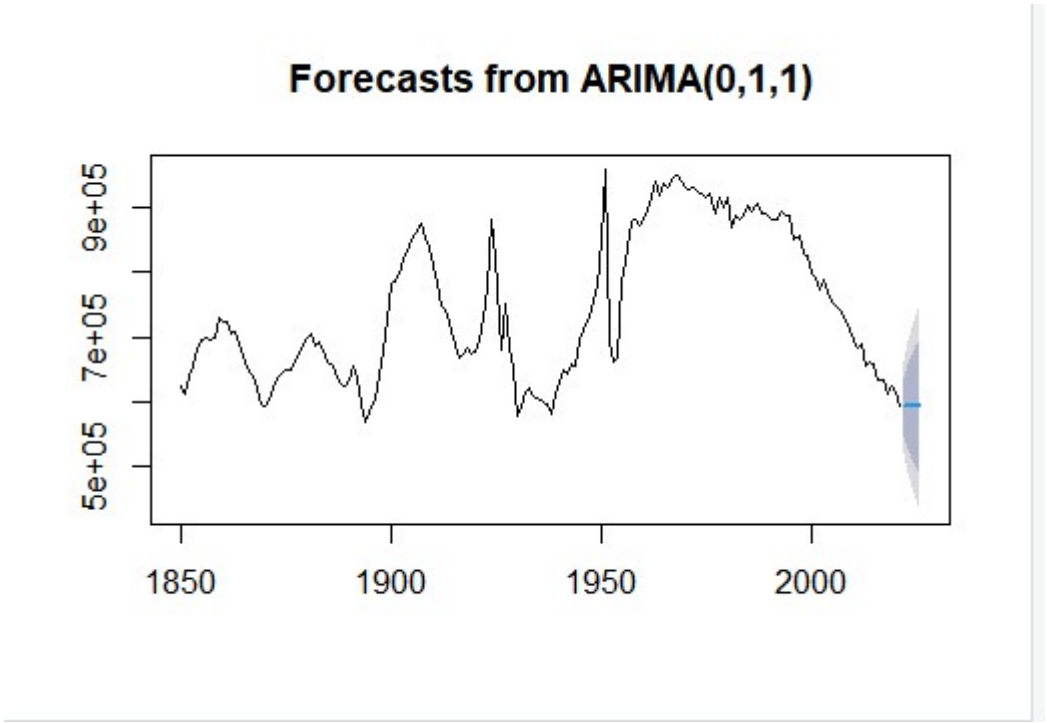
```
199
200  # Without Specifying the confidence level for prediction intervals
201  births_tsEWforecasts <- forecast(births_tsEWarima, h=5)
202  births_tsEWforecasts
203
204
```

```
> births_tsEWforecasts <- forecast(births_tsEWarima, h=5)
> births_tsEWforecasts
     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
2022       592664.3 547605.0 637723.6 523752.0 661576.5
2023       592664.3 527841.7 657486.9 493526.7 691801.9
2024       592664.3 512829.5 672499.1 470567.5 714761.1
2025       592664.3 500223.9 685104.6 451288.9 734039.6
2026       592664.3 489142.1 696186.5 434340.7 750987.8
>
```

Now we can plot the predicted live births using our arima model

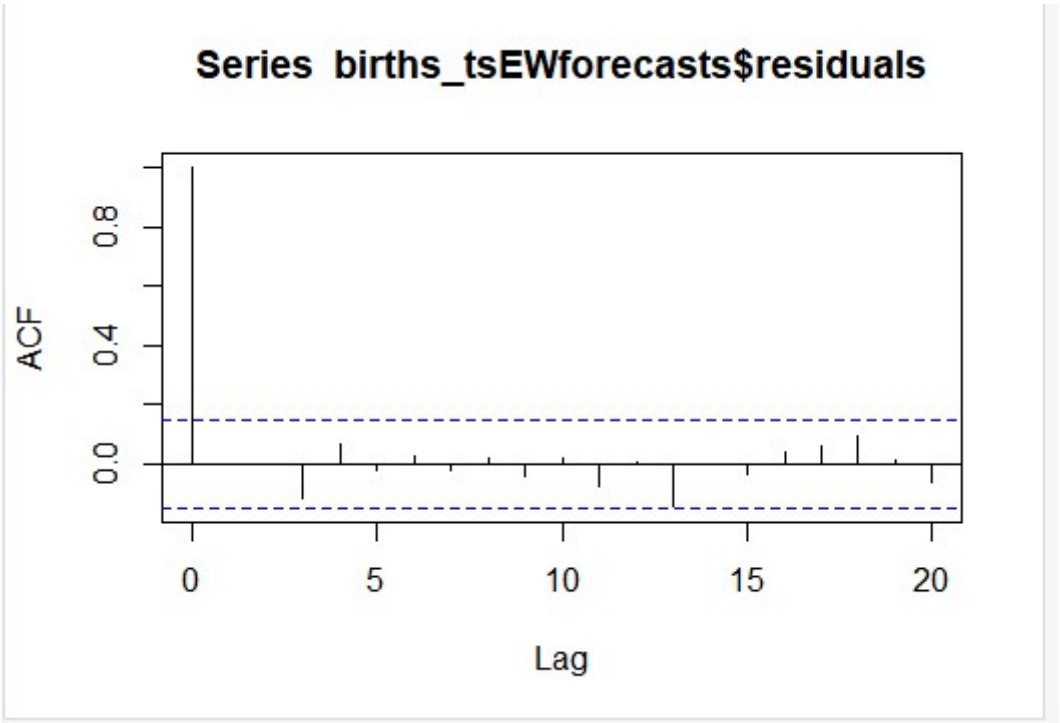
```
204
205  #plotting
206  plot(births_tsEWforecasts)
207
```

1



Now I am going to investigate that whether the forecast errors are normally distributed at mean zero with constant variance.

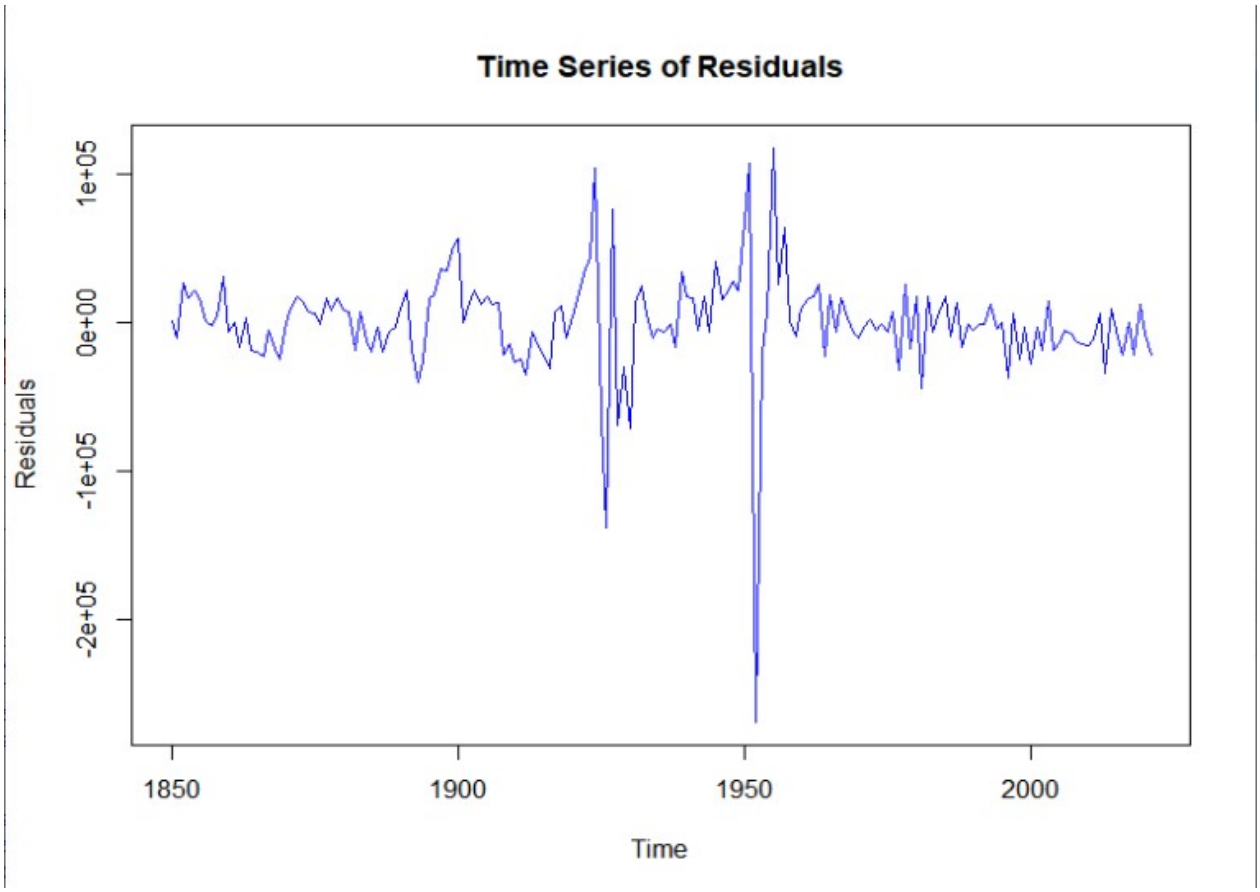
```
207
208 #make a correlogram of the forecast errors for our ARIMA(0,1,1)
209 acf(births_tsEWforecasts$residuals, lag.max=20)
210 Box.test(births_tsEWforecasts$residuals, lag=20, type="Ljung-Box")
211
> Box.test(births_tsEWforecasts$residuals, lag=20, type="Ljung-Box")

        Box-Ljung test

data:  births_tsEWforecasts$residuals
X-squared = 12.39, df = 20, p-value = 0.902

>
```



From above we can see that non of the auto correlation exceed the significance bound and the value of p= 0.902, so we conclude that there is a very little evidence of non-zero auto correlation in the forecast error.

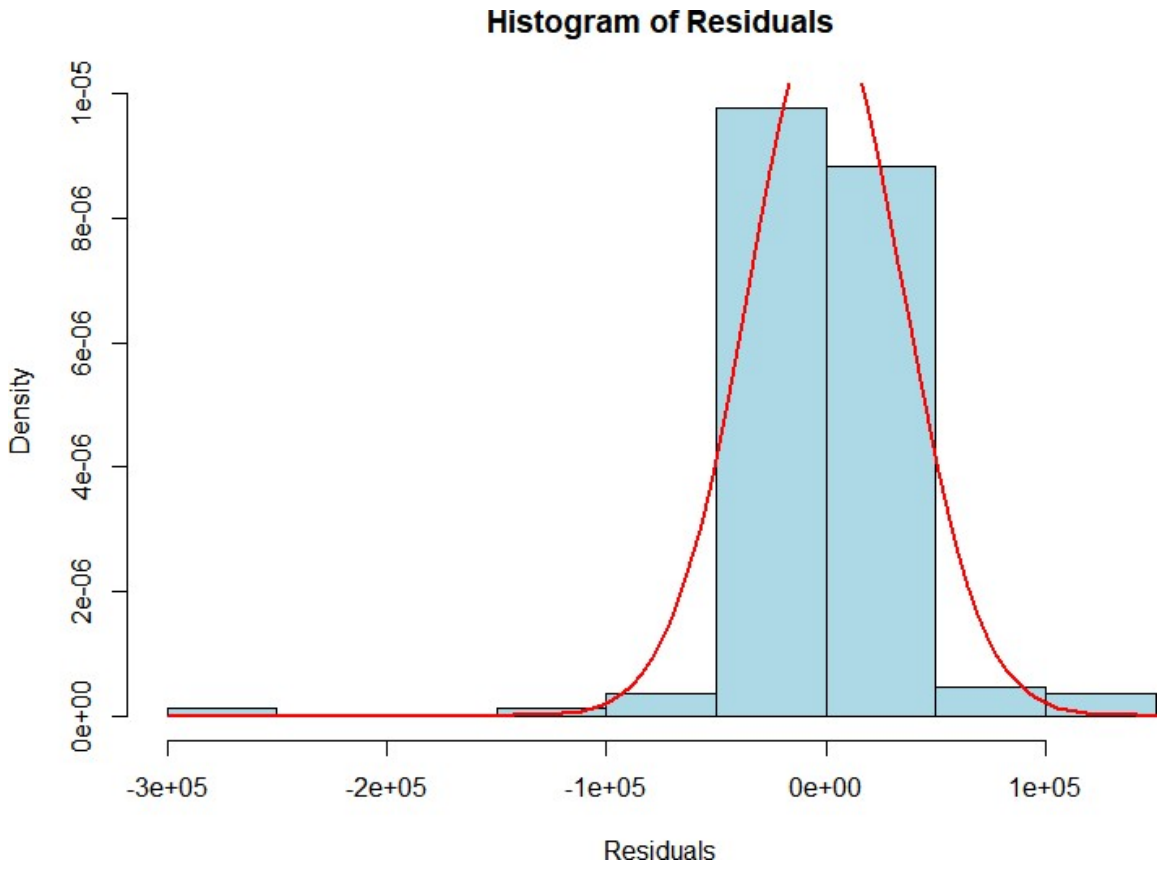
Now to check whether the forecast errors are normally distributed with mean at zero and constant variance I will plot a time series and horizontal bar plot with a time series.

```
213 # Plot the time series of residuals
214 plot.ts(births_tsEWforecasts$residuals, main = "Time Series of Residuals", ylab = "Residuals", col = "blue")
215
```




```
215
216 # Plot the histogram of residuals
217 hist(births_tsEWforecasts$residuals, main = "Histogram of Residuals", xlab = "Residuals", col = "lightblue")
218
219
220 hist(births_tsEWforecasts$residuals, main = "Histogram of Residuals", xlab = "Residuals", col = "lightblue", freq = FALSE)
221 curve(dnorm(x, mean = mean(births_tsEWforecasts$residuals, na.rm = TRUE),
222             sd = sd(births_tsEWforecasts$residuals, na.rm = TRUE)),
223       add = TRUE, col = "red", lwd = 2)
224
```

1



From above we can see that the variance is roughly constant over time and histogram shows that errors are normally distributed at mean 0 and constant variance.

Finally, you should briefly conclude by summarising the main findings from your time series analysis, including a comparison of the models and a recommendation on which is better suited to the data.

Findings:-

- The raw time series displays a long term declining trends with fluctuations so I applied differencing method to make it smooth and stationary as conformed by Augmented Dickey-Fuller test.
- I applied SMA method to check the underlying trends in data.
- Holt winter method capture the declining trend.
- The data does not shows autocorrelation which indicates a well fitted model.
- Differencing the series and applying ARIMA model to the data set could improve fit by explicitly modeling autocorrelation structures.
- Arima model of (0,1,1) was generated from ACF and PCF.

1

- When I apply auto.arima model it gives me the parameter as (0,1,0) but when I apply ACF it give me a q=1. I considered to take q=1 because auto.arima model also consider other values to predict p and q values but according to result my manual findings was better.