May 2023

UK Bike Sharing Report

Statical Data Analytics Course
Università degli Studi di Napoli Federico II

Prof. Roberta Siciliano
Prof. Michele Staiano

REZA FARNAGHI
A.MOHAMMAD CHINISAZ
POUYA SATTARI
S.EHSAN TAHERI
NARGES DAVOUDI

Introduction

The London Bike Sharing Dataset is a comprehensive collection of data related to the bike-sharing system in London, United Kingdom. It provides valuable insights into the usage patterns, trends, and dynamics of the bike-sharing service, offering researchers, analysts, and enthusiasts an opportunity to explore and analyze various aspects of urban transportation.

About Dataset

The dataset contains information gathered over a specific period, including detailed records of bike trips, weather conditions, and corresponding time and location data. These details enable researchers to examine factors that influence bike-sharing usages, such as weather conditions, time of day, and day of the week, among others.

The London Bike Sharing Dataset offers a range of variables to explore. It typically includes information such as the starting and ending station of each trip, the duration of the trip, the bike ID, user types (e.g., casual or registered), and the timestamp of the start and end of the journey. Additionally, the dataset often includes weather-related data like temperature, humidity, wind speed, and weather conditions at the time of the trip.

Researchers can utilize this dataset to gain insights into user behavior, identify peak demand periods, evaluate the impact of weather on bike-sharing usage, optimize bike station placements, and assess the overall performance of the bike-sharing system. The dataset can also serve as a basis for developing predictive models, creating visualization tools, and conducting statistical analyses to support decision-making processes in urban planning and transportation management.

By analyzing the London Bike Sharing Dataset, researchers and practitioners can contribute to enhancing the efficiency, sustainability, and accessibility of bike-sharing systems, as well as gain a deeper understanding of the dynamics of urban transportation in one of the world's largest cities.

Metadata

- "timestamp" - timestamp field for grouping the data
- "cnt" - the count of new bike shares
- "t1" - the real temperature in C
- "t2" - the temperature in C "feels like"
- "hum" - humidity in percentage
- "wind_speed" - wind speed in km/h
- "weather_code" - category of the weather
- "is_holiday" - boolean field - 1 holiday / 0 non-holiday
- "is_weekend" - boolean field - 1 if the day is weekend
- "season" - category field meteorological seasons:
  0-spring ; 1-summer; 2-fall; 3-winter.
- "weathe_code" category description:
  1 = Clear; mostly clear but have some values with haze/fog/patches of fog/ fog in the vicinity
  2 = scattered clouds / few clouds
  3 = Broken clouds
  4 = Cloudy
  7 = Rain/ light Rain shower/ Light rain
  10 = rain with thunderstorm
  26 = snowfall
  94 = Freezing Fog

Statistical thinking

The London Bike Sharing Dataset is a comprehensive collection of data related to the bike-sharing system in London, United Kingdom. It provides valuable insights into the usage patterns, trends, and dynamics of the bike-sharing service, offering researchers, analysts, and enthusiasts an opportunity to explore and analyze various aspects of urban transportation.

Problem Statement

Londoners can use a bicycle rental system for commuting within the city every day. These trips can be to get to work or school or to go sightseeing on a pleasant summer day. The number of rented bicycles depends on important factors such as air temperature, air humidity, time of day, etc. **In this project, we are trying to predict the number of bicycles that can be rented every hour of the day.**

Exploratory Data Analysis

Data Pre-Processing and Cleaning

```r
# library(dplyr)
library(skimr)
library(ggplot2)
library(lubridate)
library(tibble)
library(caret)
library(e1071)
library(rpart)
library(PerformanceAnalytics)
library(xgboost)
library(randomForest)
library(scales)
library(gridExtra)

# set a theme for ggplots
theme_set(
  theme_bw(base_size = 15)+
  theme(plot.title.position = "plot")

# load our dataset
bike <- read.csv(file = "london_merged.csv", header = T)

# print the dimension of the data
dim(bike)
```

In order to pre-process the data, we should pay attention to a few aspects of the data including the variable types, the number of missing data in each column, scaling the variables (if necessary), excluding unused variables, etc.

Extract year, month, day and hour variable from the timestamp column

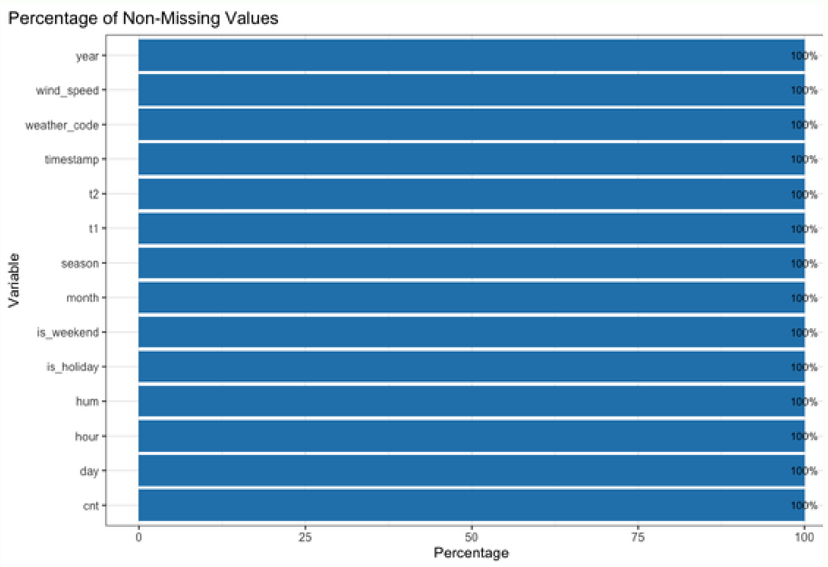
```
# add year, month, day, and hour to the data

bike <- bike %>% mutate(
  year=year(bike$timestamp),
  month=month(bike$timestamp),
  day=day(bike$timestamp),
  hour=hour(bike$timestamp))
```

Check for Number of Missing Values in the Data

As we can see in the above table, there are no missing values in the data.

```
# count number of missing values in each column
nm <- colSums(is.na(bike))
# convert missing count information to a table
tibble(Variable=names(nm), Number_of_Missing=as.vector(nm)) %>%
  knitr::kable()

non_missing_percentage <- colMeans(!is.na(bike)) * 100
non_missing_df <- data.frame(Variable = names(non_missing_percentage),
Percentage = non_missing_percentage)

# Plot for non missing values
ggplot(data = non_missing_df, aes(x = Variable, y = Percentage)) +
  geom_bar(stat = "identity", fill = "steelblue", position =
position_stack(vjust = 1)) +
  geom_text(aes(label = paste0(round(Percentage), "%")), vjust = 0.5) +
  labs(title = "Percentage of Non-Missing Values", x = "Variable", y =
"Percentage") +
  coord_flip()
```

Variable	Number of Missing
timestamp	0
cnt	0
t1	0
t2	0
hum	0
wind_speed	0
weather_code	0
is_holiday	0
is_weekend	0
season	0
year	0
month	0
day	0
hour	0



Exclude Unused Variables from the Data

All the variables can be useful in this analysis exept timestamp so it is exculded from further analyses.

```
# exclude timestamp
bike <- bike %>% select(-timestamp)
```

Take a look to Dataset again

```
# print the first 10 rows of the data
head(bike, n=10)
```

```
##    cnt  t1   t2   hum wind_speed weather_code is_holiday is_weekend season year  month day hour
## 1  182 3.0  2.0  93.0        6.0            3          0          1      3 2015     1   4    0
## 2  138 3.0  2.5  93.0        5.0            1          0          1      3 2015     1   4    1
## 3  134 2.5  2.5  96.5        0.0            1          0          1      3 2015     1   4    2
## 4   72 2.0  2.0 100.0        0.0            1          0          1      3 2015     1   4    3
## 5   47 2.0  0.0  93.0        6.5            1          0          1      3 2015     1   4    4
## 6   46 2.0  2.0  93.0        4.0            1          0          1      3 2015     1   4    5
## 7   51 1.0 -1.0 100.0        7.0            4          0          1      3 2015     1   4    6
## 8   75 1.0 -1.0 100.0        7.0            4          0          1      3 2015     1   4    7
## 9  131 1.5 -1.0  96.5        8.0            4          0          1      3 2015     1   4    8
## 10 301 2.0 -0.5 100.0        9.0            3          0          1      3 2015     1   4    9
```

```
# print the last 10 rows of the data
tail(bike, n=10)
```

```
##        cnt  t1  t2  hum wind_speed weather_code is_holiday is_weekend season  year month day
hour
## 17405  765 6.0 2.0 73.5         22            3          0          0      3  2017     1   3   14
## 17406  845 6.0 2.0 71.0         27            4          0          0      3  2017     1   3   15
## 17407 1201 6.0 2.0 71.0         26            4          0          0      3  2017     1   3   16
## 17408 2742 6.0 2.0 73.5         21            3          0          0      3  2017     1   3   17
## 17409 2220 5.0 1.0 81.0         22            2          0          0      3  2017     1   3   18
## 17410 1042 5.0 1.0 81.0         19            3          0          0      3  2017     1   3   19
## 17411  541 5.0 1.0 81.0         21            4          0          0      3  2017     1   3   20
## 17412  337 5.5 1.5 78.5         24            4          0          0      3  2017     1   3   21
## 17413  224 5.5 1.5 76.0         23            4          0          0      3  2017     1   3   22
## 17414  139 5.0 1.0 76.0         22            2          0          0      3  2017     1   3   23
```

Descriptive Statistics

The descriptive statistics includes data summarization and data visualization.

Data Summarization

```
# data summarization
skim(bike)
```

Data summary	
Name	bike
Number of rows	17414
Number of columns	13
Column type frequency:	
factor	8
numeric	5
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
weather_code	0	1	FALSE	4	1: 6150, 2: 4034, 4: 3679, 3: 3551
is_holiday	0	1	FALSE	2	0: 17030, 1: 384
is_weekend	0	1	FALSE	2	0: 12444, 1: 4970
season	0	1	FALSE	4	0: 4394, 1: 4387, 3: 4330, 2: 4303
year	0	1	FALSE	3	201: 8699, 201: 8643, 201: 72
month	0	1	FALSE	12	5: 1488, 1: 1487, 8: 1484, 12: 1484
day	0	1	FALSE	31	6: 576, 14: 576, 21: 576, 22: 576
hour	0	1	FALSE	24	16: 730, 12: 729, 15: 729, 13: 728

Descriptive Statistics

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
cnt	0	1	**1143.10**	**1085.11**	**0.0**	**257**	**844.0**	**1671.75**	**7860.0**	
t1	0	1	**12.47**	**5.57**	**-1.5**	**8**	**12.5**	**16.00**	**34.0**	
t2	0	1	**11.52**	**6.62**	**-6.0**	**6**	**12.5**	**16.00**	**34.0**	
hum	0	1	**72.32**	**14.31**	**20.5**	**63**	**74.5**	**83.00**	**100.0**	
wind_speed	0	1	**15.91**	**7.89**	**0.0**	**10**	**15.0**	**20.50**	**56.5**	

The above results show that the London bike-sharing data is summarised based on the variable type. In the following subsection, the data are visualized in more detail.

Dataset Summary

cnt = Count	
Min.	0.00
1st Qu.	260
Median	846
Mean	1146
3rd Qu.	1676
Max.	7860

t1 = Temperature 1	
Min.	-1.5
1st Qu.	8.5
Median	12.5
Mean	12.5
3rd Qu.	16.0
Max.	34.0

t2 = feels like temperature	
Min.	-6.00
1st Qu.	6.00
Median	12.50
Mean	11.56
3rd Qu.	16.00
Max.	34.00

hum = humidity level	
Min.	20.50
1st Qu.	63.00
Median	74.50
Mean	72.28
3rd Qu.	83.00
Max.	100.00

wind_speed = wind speed	
Min.	0.00
1st Qu.	10.00
Median	15.00
Mean	15.92
3rd Qu.	20.50
Max.	56.50

weather_code	
Min.	1.000
1st Qu.	1.000
Median	2.000
Mean	2.721
3rd Qu.	3.000
Max.	26.000

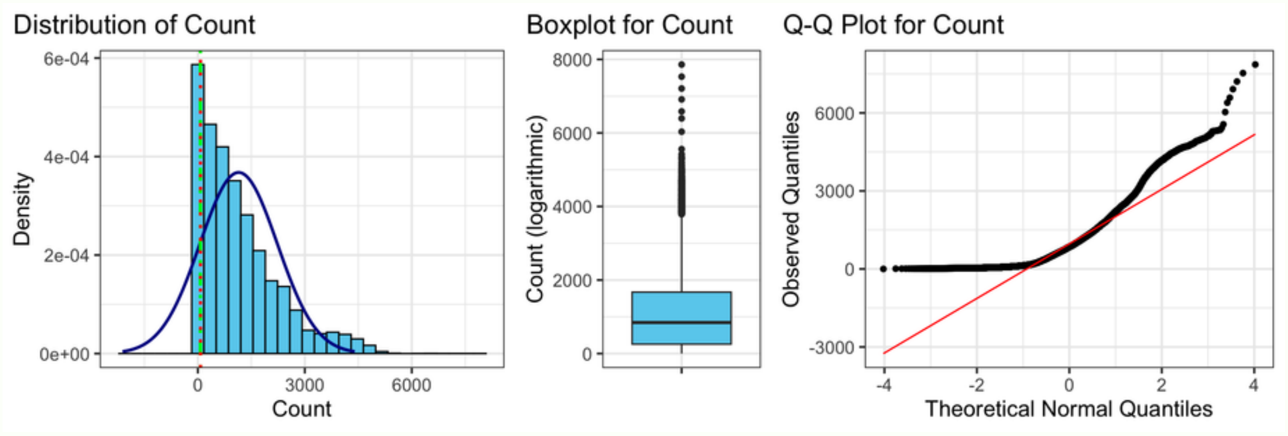
is_holiday	
Min.	0.00000
1st Qu.	0.00000
Median	0.00000
Mean	0.02076
3rd Qu.	0.00000
Max.	1.00000

is_weekend	
Min.	0.0000
1st Qu.	0.0000
Median	0.0000
Mean	0.2852
3rd Qu.	1.0000
Max.	1.0000

season	
Min.	0.000
1st Qu.	0.000
Median	1.000
Mean	1.486
3rd Qu.	2.000
Max.	3.000

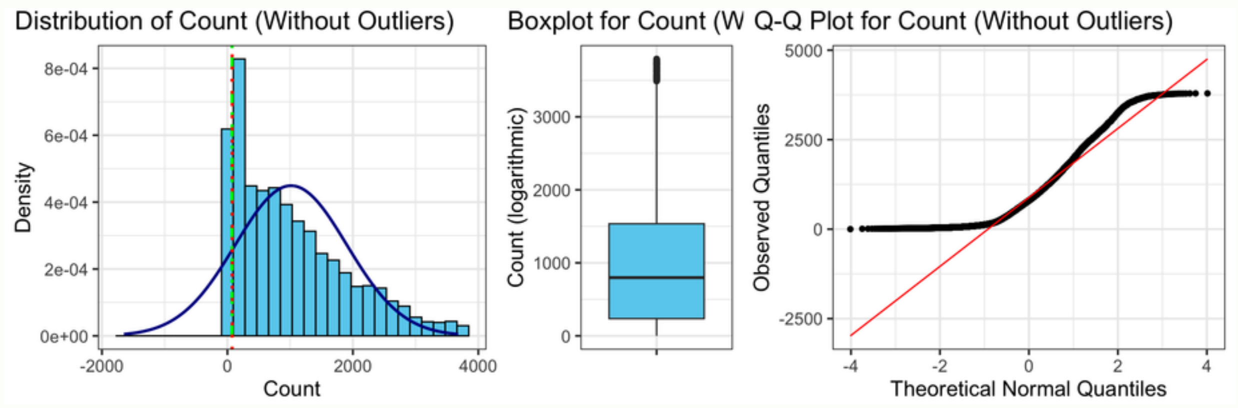
year	
Min.	2015
1st Qu.	2015
Median	2016
Mean	2016
3rd Qu.	2016
Max.	2017

Data Visualization

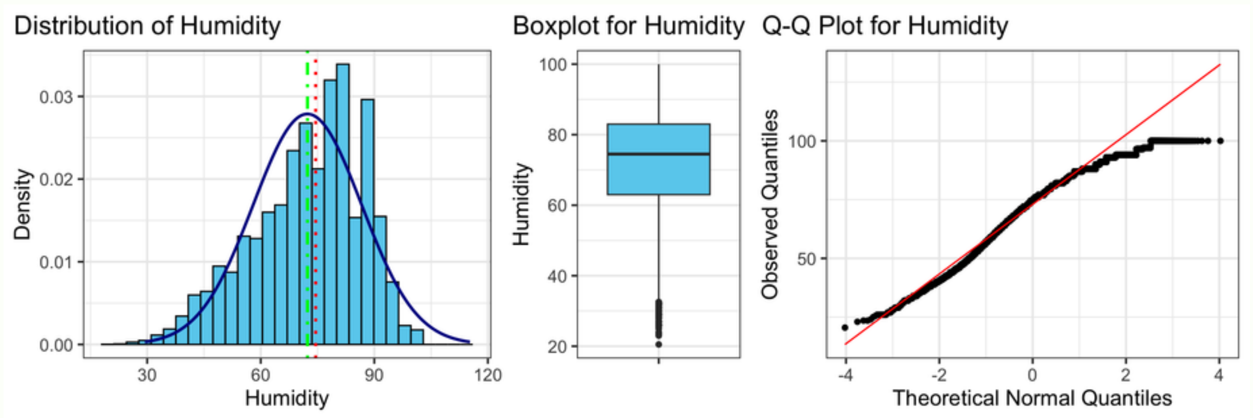


- **Variation in Bike Shares:** The histogram and box plot show that the total number of bike shares varies from 0 to slightly above 5000.
- **Central Tendency:** The average (represented by the red dashed line) and the median (represented by the green dotted line) of the variable are closer to the lower end of the range, indicating that the majority of data points have lower bike share counts.
- **Majority of Data:** The observation mentions that most of the data falls between 0 and approximately 1800 bike shares, indicating that this range contains a significant portion of the dataset.
- **Outliers:** The box plot identifies a few values above the typical range of the data, shown as black points. These values are considered outliers due to their unusually higher bike share counts.
- The data points generally align closely with the diagonal line, suggesting a relatively normal distribution. However, there are noticeable deviations from the line at the extreme ends, indicating the presence of outliers or departures from normality in those areas.

Outlier removed



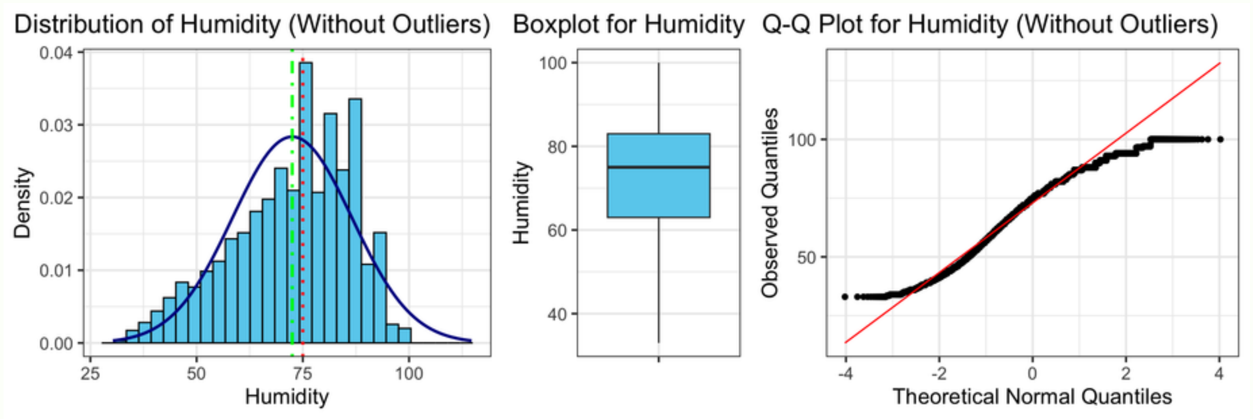
Data Visualization



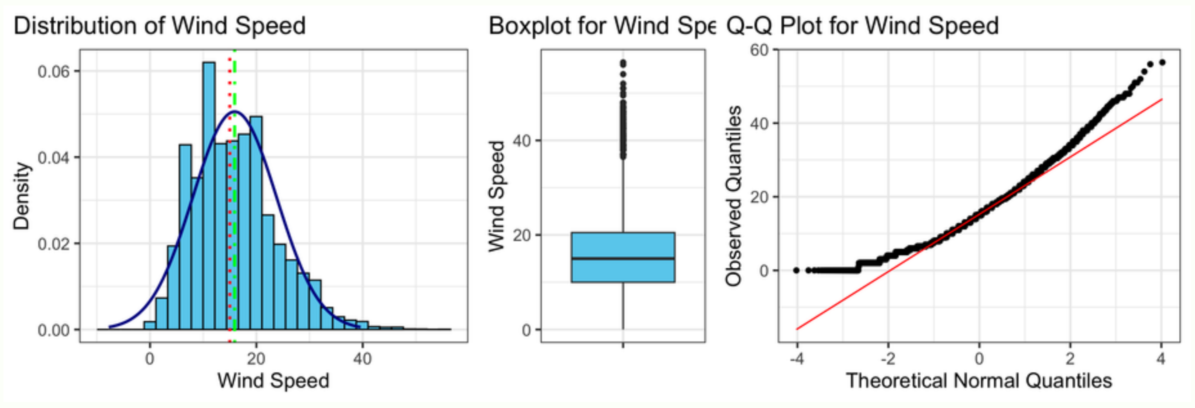
Regarding the histogram of the humidity variable, it indicates that the majority of observations are concentrated around the range of 40% to 90% humidity. This suggests that humidity levels within this range are more common in London. There are fewer instances of extremely low or high humidity values, as shown by the decreasing frequency as we move away from the central range.

Distribution of humidity values deviates from a normal distribution. It means that the observed humidity values don't follow the expected pattern based on a normal distribution. Specifically, the plot reveals that the extreme values of humidity are more frequent than what would be expected in a normal distribution. This suggests that there might be outliers or certain factors affecting the humidity levels that cause them to deviate from a normal pattern.

Outlier removed



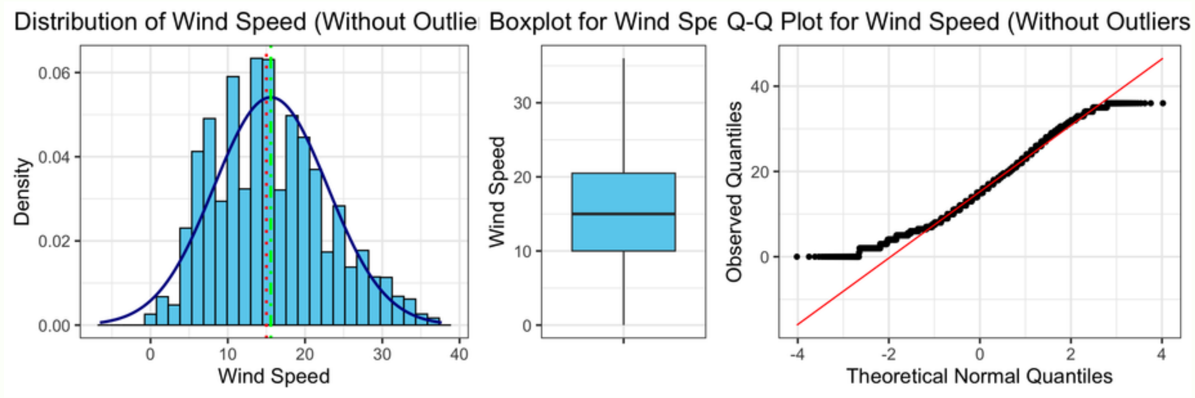
Data Visualization



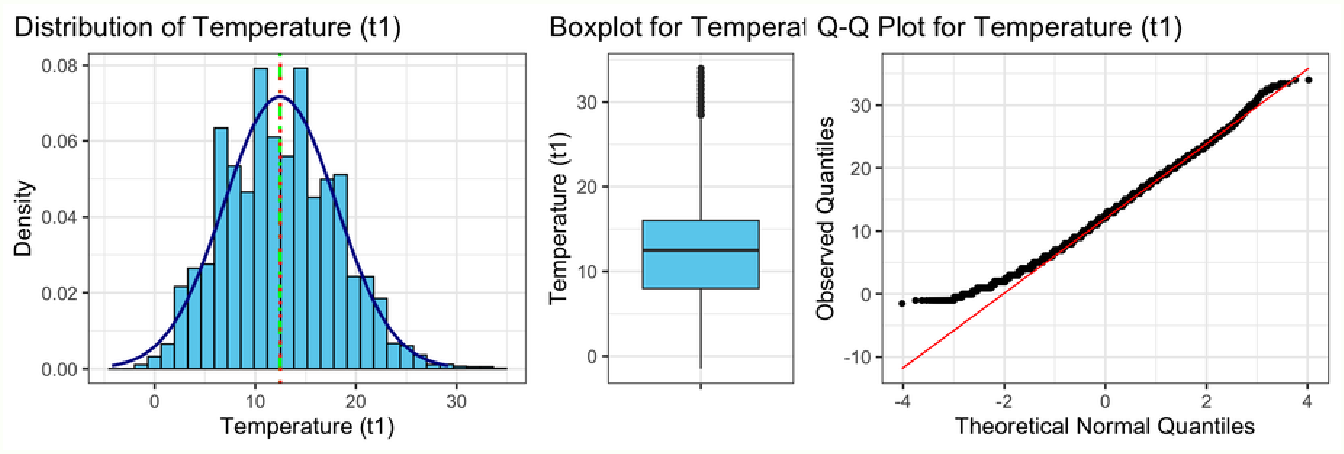
It shows that the distribution is right-skewed, with the majority of observations concentrated at lower wind speeds. This suggests that lower wind speeds are more common in London. As the wind speed increases, the frequency of observations gradually decreases, indicating that higher wind speeds are less frequent. The histogram provides an overview of the distribution and highlights the most common wind speed ranges in the dataset.

The distribution of wind speed values deviates from a normal distribution. The plot shows a noticeable curvature, indicating a departure from linearity. This suggests that the wind speed values are not normally distributed and may be influenced by certain factors or conditions. It is important to consider this non-normality when analyzing the wind speed variable and take appropriate statistical measures accordingly.

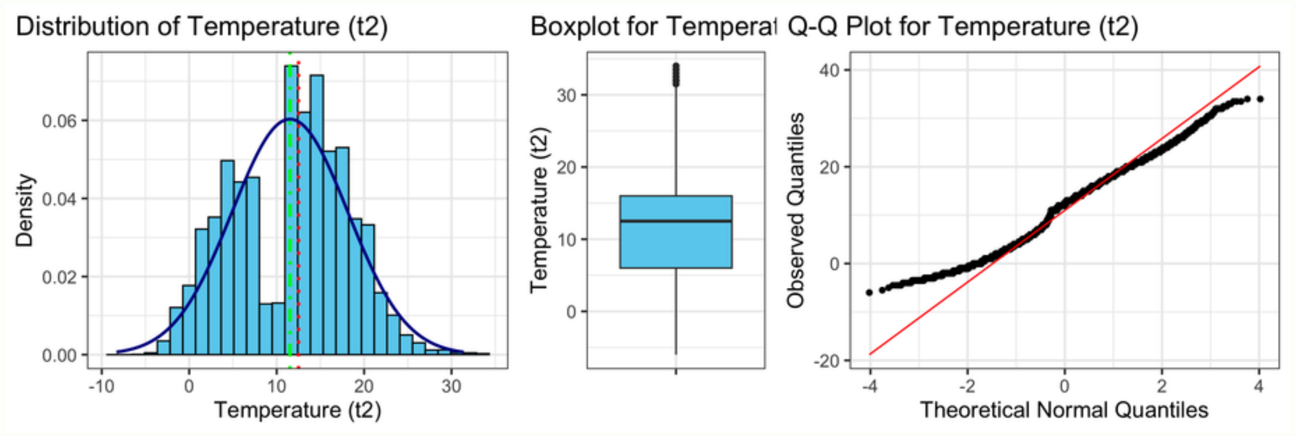
Outlier removed



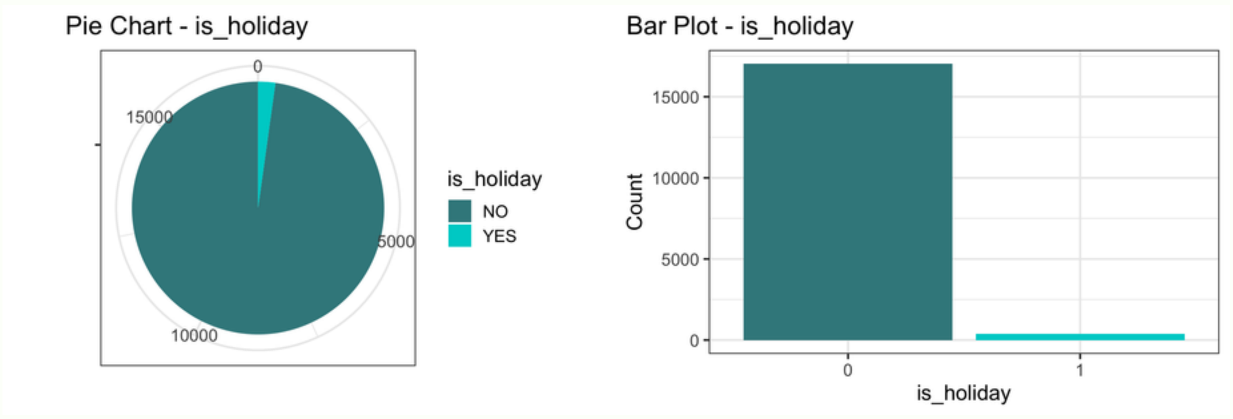
Data Visualization



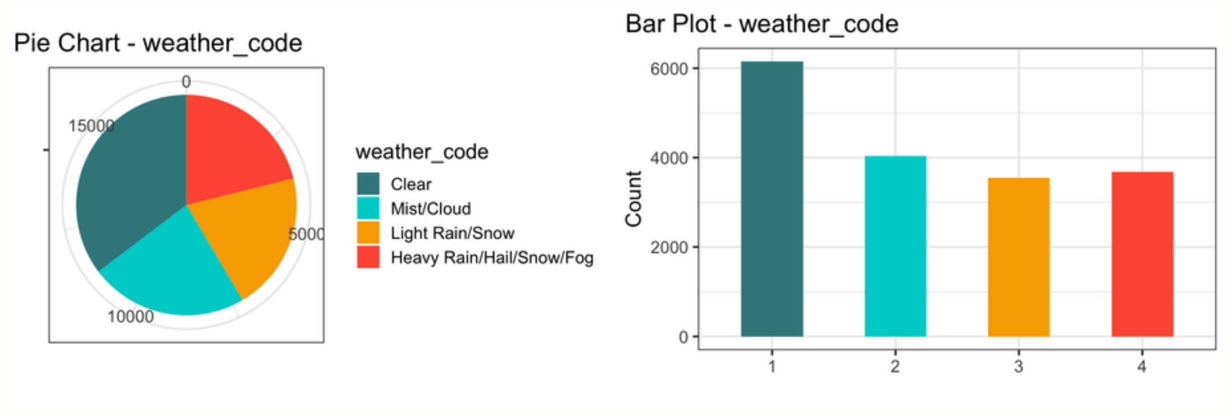
The temperature variable shows a concentration of observations between 10 to 25 degrees Celsius, indicating common temperatures in London. The count variable displays a right-skewed distribution, with higher occurrences of lower rental counts and a gradual decrease as the count increases.
This QQ plot reveals a relatively linear relationship between the observed quantiles and the theoretical quantiles, indicating that the data is approximately normally distributed. There are minimal deviations from the diagonal line, suggesting that the temperature variable follows a normal distribution with few outliers or departures from normality. This implies that the distribution of temperature values is consistent with a normal distribution pattern.



Data Visualization

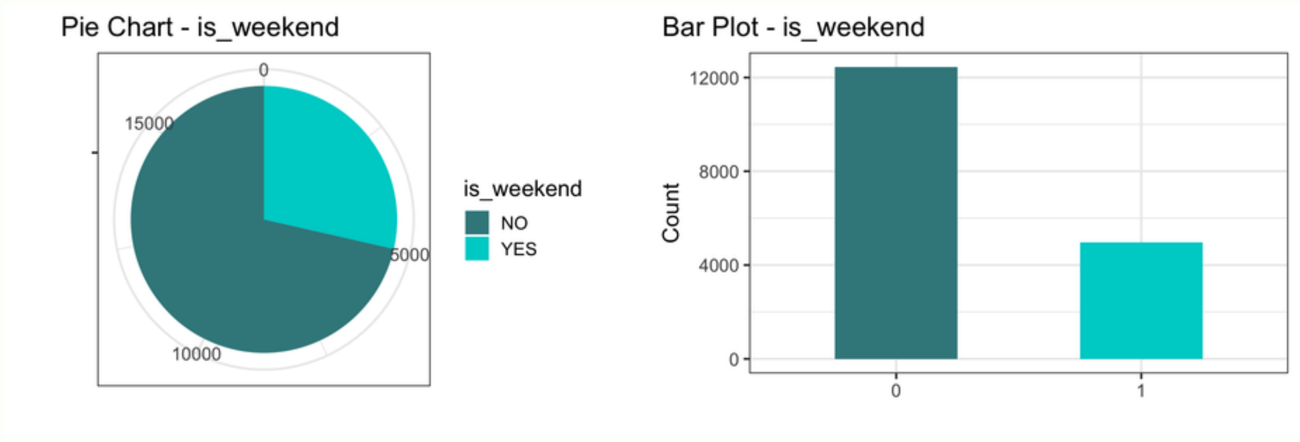


The chart shows that the majority of observations (around 84%) are labeled as "Not a holiday," while the remaining observations (approximately 16%) are labeled as "Holiday." This indicates that the dataset consists mostly of non-holiday days, with a smaller proportion of holiday days.

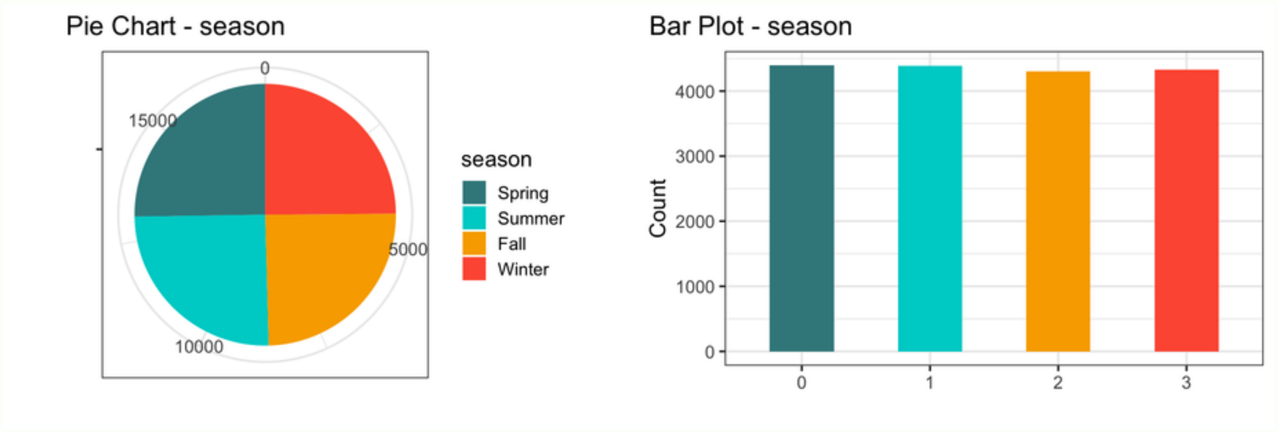


The largest portion of the pie is attributed to weather code 1, which represents "Clear" weather conditions. The other segments of the pie represent different weather codes corresponding to various weather conditions such as "Mist", "Light rain", "Cloudy", and others. This indicates that the majority of observations in the dataset experienced clear weather conditions, while other weather conditions occurred less frequently.

Data Visualization

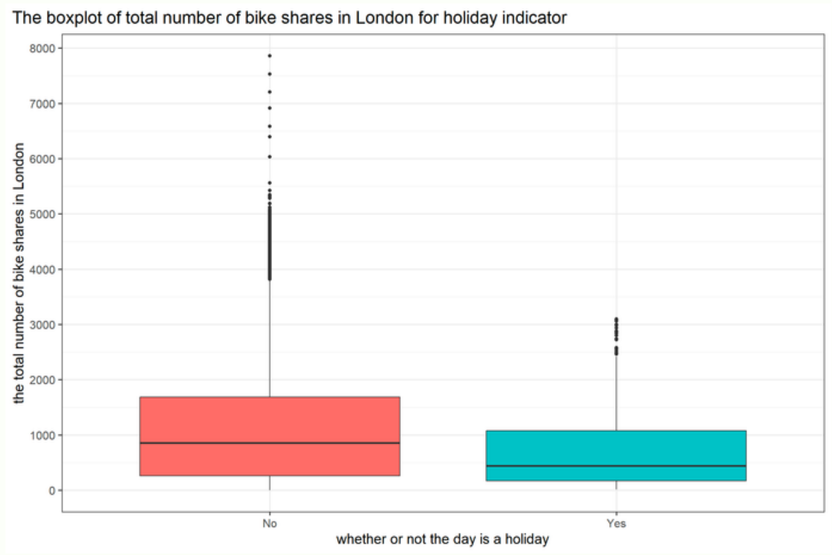


The larger section of the pie corresponds to "Not Weekend", indicating that a majority of the observations in the dataset are from non-weekend days. The smaller section represents "Weekend" observations, indicating that they are less frequent in the dataset. This suggests that the dataset contains a larger proportion of non-weekend data points compared to weekend data points.

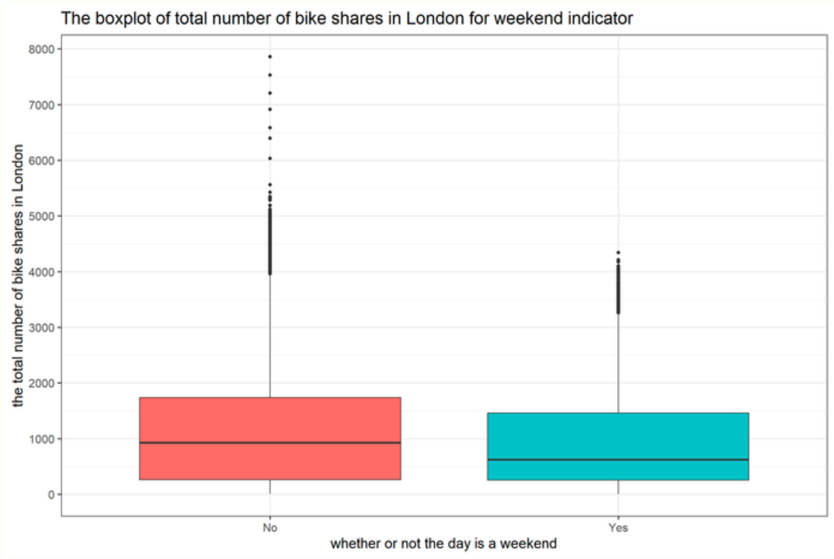


The majority of observations in the dataset belong to the "Spring" season, as indicated by the largest section of the pie. The "Summer" and "Autumn" seasons have similar proportions, while the "Winter" season has the smallest proportion of observations. This suggests that the dataset is biased towards the "Spring" season, with relatively fewer observations from the other seasons.

Data Visualization

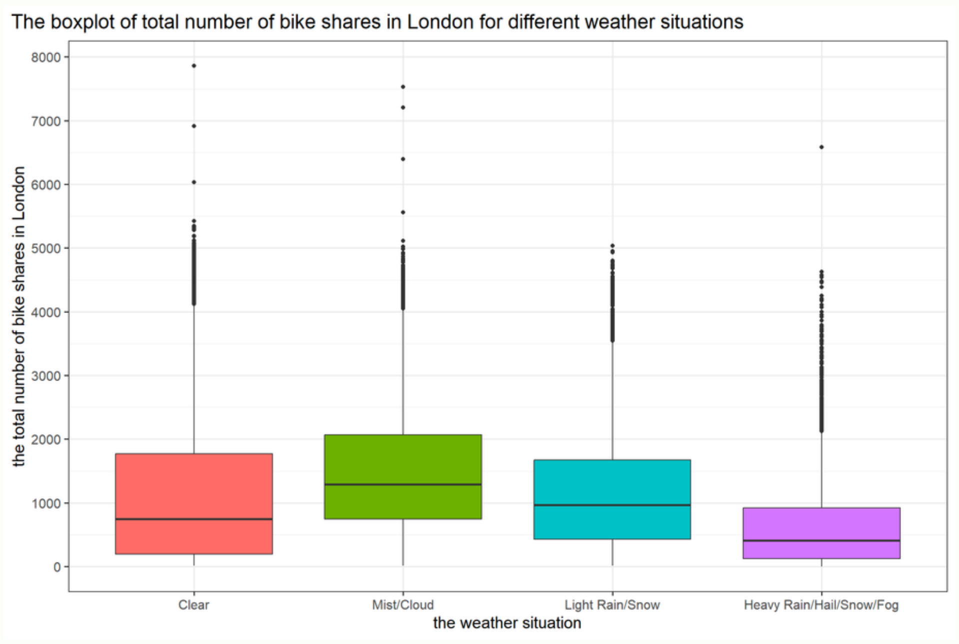


- The median for holidays is lower compared to non-weekends, indicating that the median number of bike rentals is generally lower on weekends.
- The box for weekends is smaller than the box for non-weekends, suggesting less variability in bike rentals on weekends.

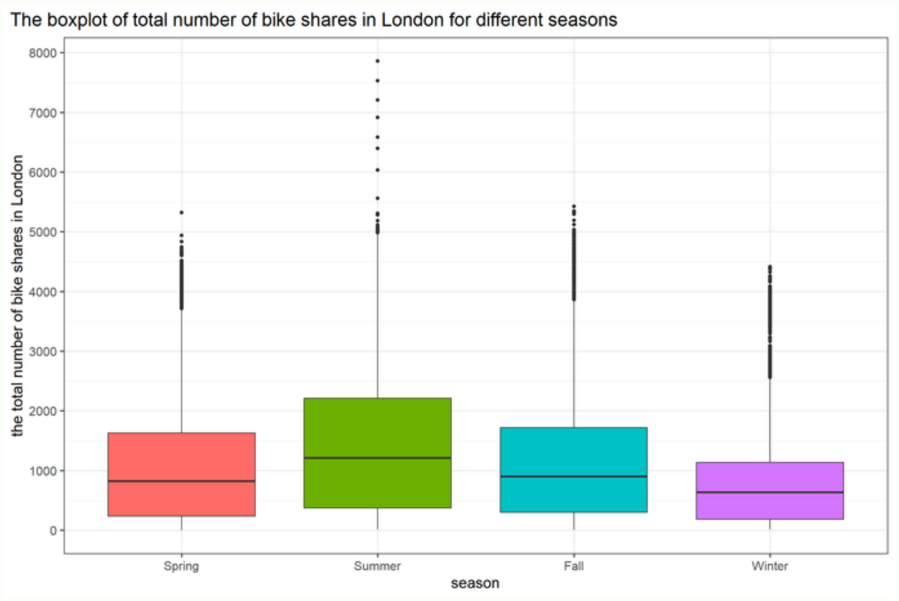


- The median for weekends is lower compared to non-weekends, indicating that the median number of bike rentals is generally lower on weekends.
- The box for weekends is smaller than the box for non-weekends, suggesting less variability in bike rentals on weekends.

Data Visualization

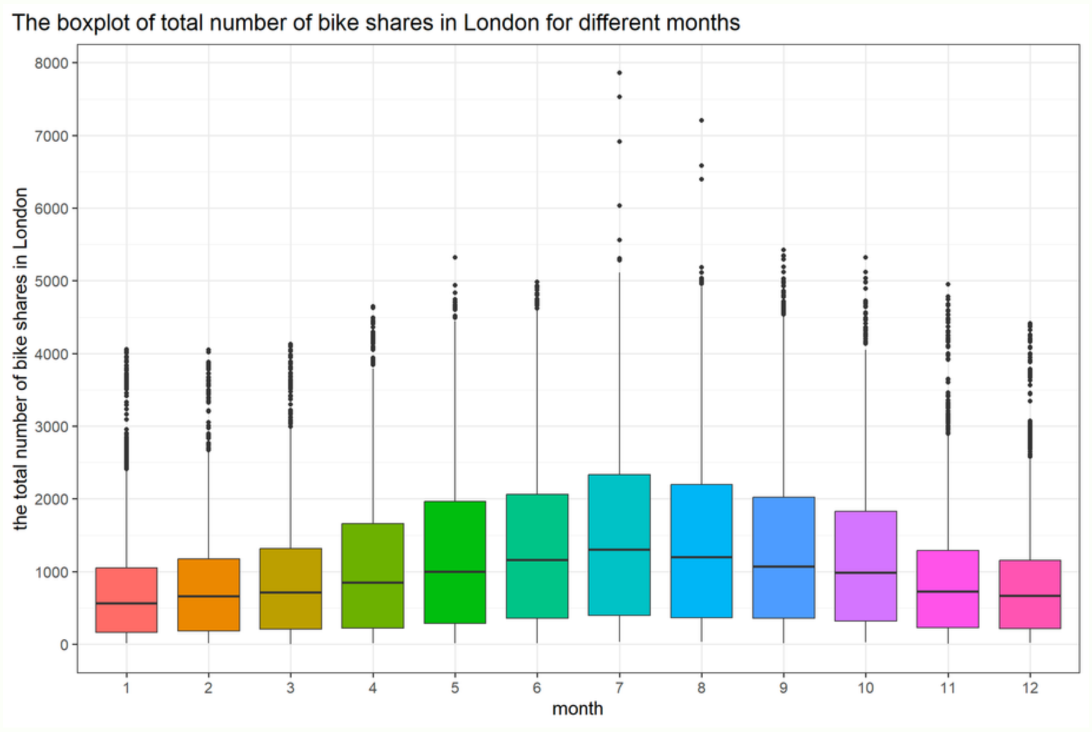


- Weather situation 2 (Mist or fog) has the highest median among all the weather codes, indicating that this weather condition is associated with higher bike rental counts on average.
- Weather situation 1 (Clear or few clouds) and weather situation 3 (Light snow or rain) have similar medians, suggesting that these weather conditions are associated with relatively lower bike rental counts compared to weather situation 2.
- Weather situation 4 (Heavy rain, ice pellets, or snow) has the lowest median, indicating that this weather condition is associated with the lowest bike rental counts on average.

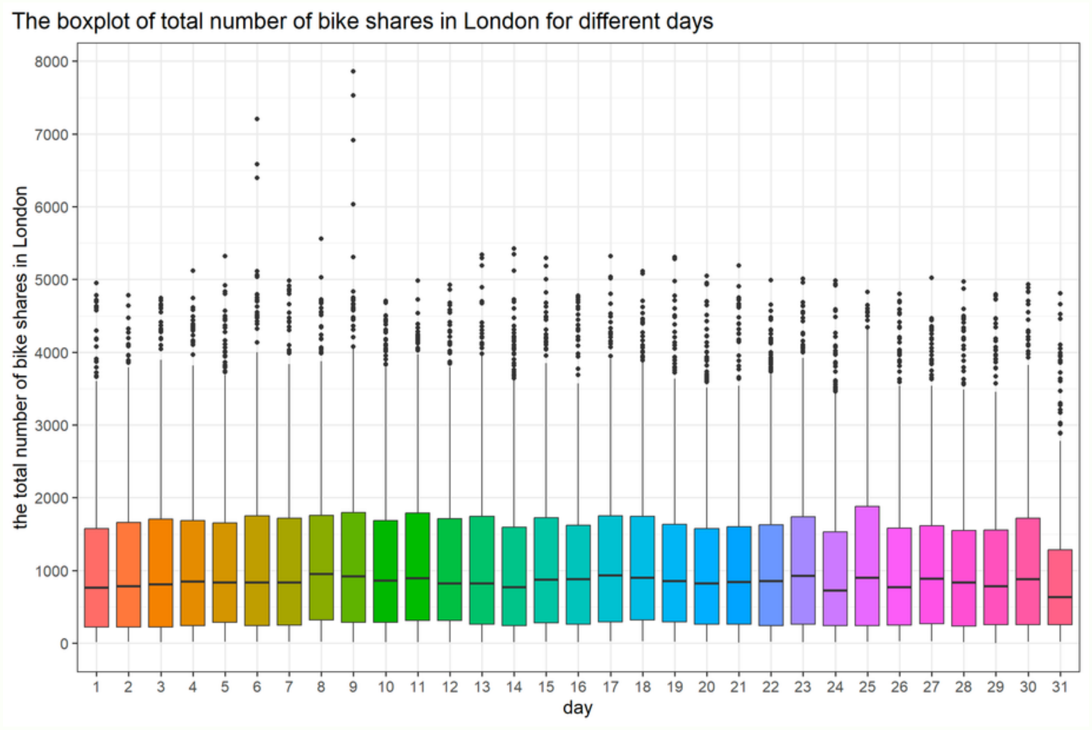


- The "spring" and "fall" seasons have similar medians and IQRs, indicating comparable bike rental patterns.
- The "autumn" season has a slightly higher median and a larger IQR, suggesting a wider range of bike rentals during this season.
- The "winter" season has the lowest median and the smallest IQR, indicating relatively lower bike rentals and a narrower range of values.

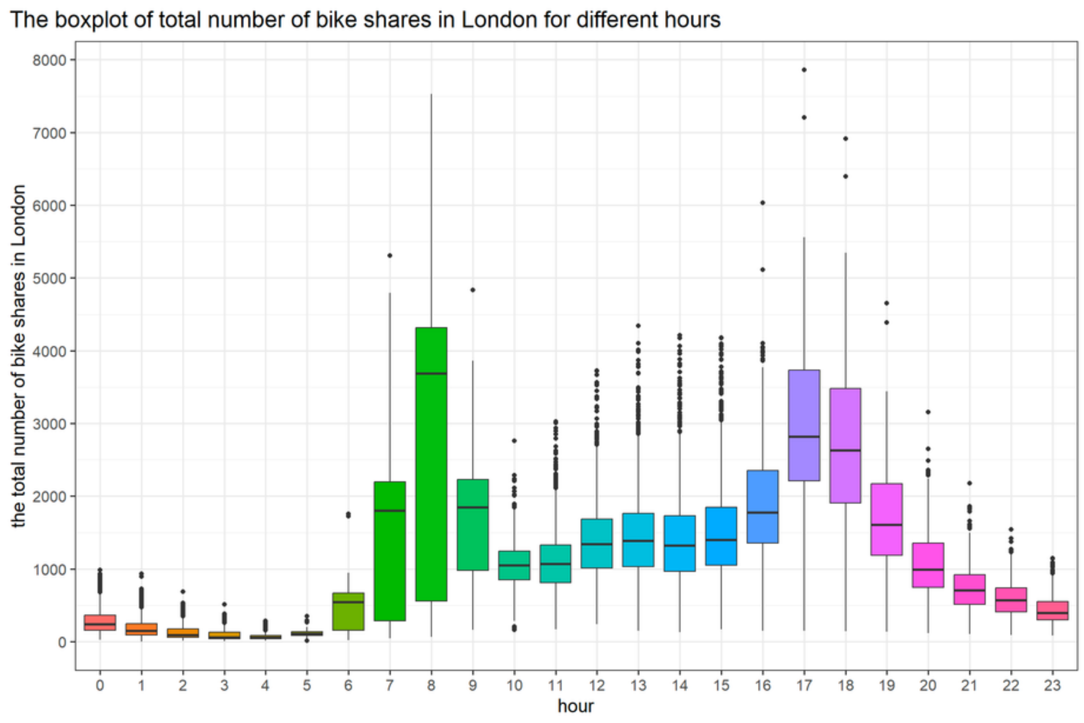
Data Visualization



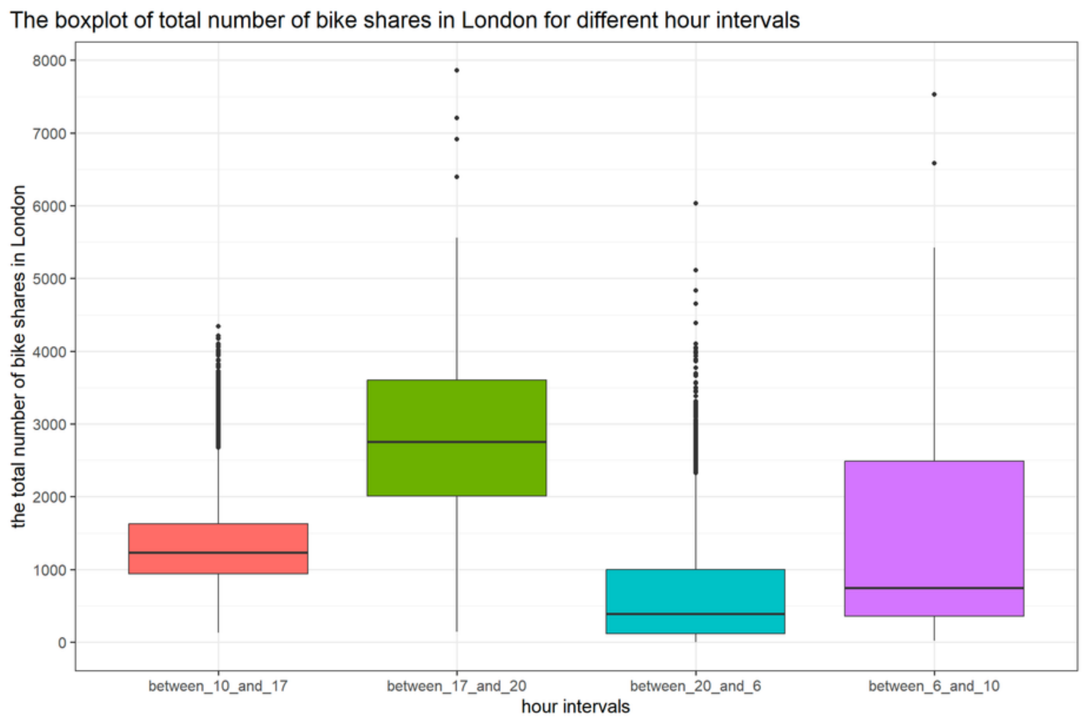
- The median bike share count appears to be relatively consistent across the months, with slight variations.
- The range of bike share counts is wider in some months compared to others, as indicated by the length of the whiskers.
- There are a few outliers in certain months, suggesting unusual instances of exceptionally high or low bike share counts.



Data Visualization

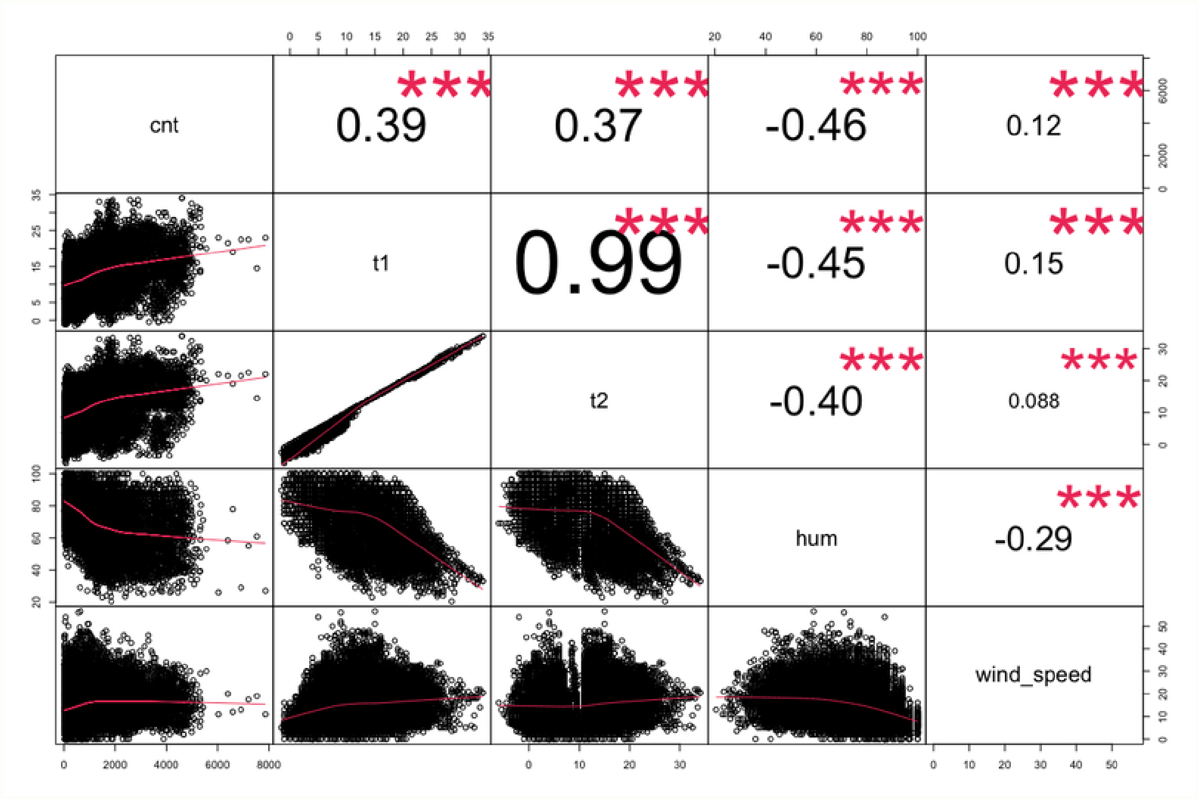


- The median bike share count tends to be higher during the morning and evening rush hours, specifically between 7 to 9 and 16 to 19. This suggests that there is a greater demand for bike rentals during these peak commuting hours.
- The counts during the late night and early morning hours (between 24 and 5) are relatively lower, indicating reduced bike rental activity during these periods.
- The range of counts varies across different hours, with wider ranges observed during the peak hours and narrower ranges during the off-peak hours.



The "between_17_and_20" hour interval exhibits a higher number of bike share counts, and the median line is positioned in the middle of the box, indicating a relatively balanced distribution of bike rentals during this time period.
The "between_6_and_10" hour interval exhibits a wide range of bike share counts, but the median value suggests that the majority of rentals during this time period are relatively low.

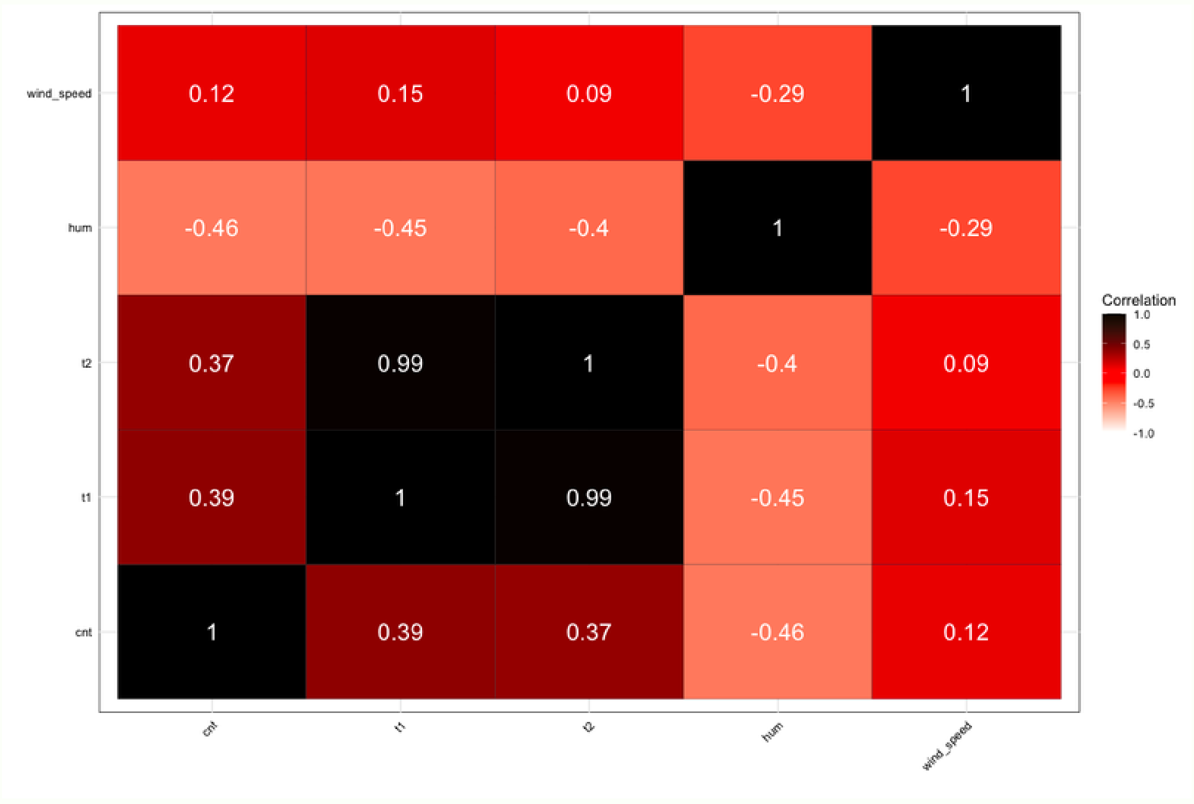
Bivariate Data, Scatter Plots and Corr values



The scatter plots for t1 and t2, there is a strange diagonal trend showing that higher Bike share counts leads to higher temperatures; and a correlation value of just over 0.5 for both of these features supports this observation. Conversely, the plots for hum and wind_speed show a slightly negative correlation, indicating that there are fewer bike shares in London on days with high humidity or wind_speed. The red stars beside the correlation values are comming from the correlation test with the null hypothesis that the correlation is not significant between pairs of variables. Since the p-value of each test is lower than 0.01, the null hypothesis is rejected at all the correlation tests and the three stars proves the argument.

Data Visualization

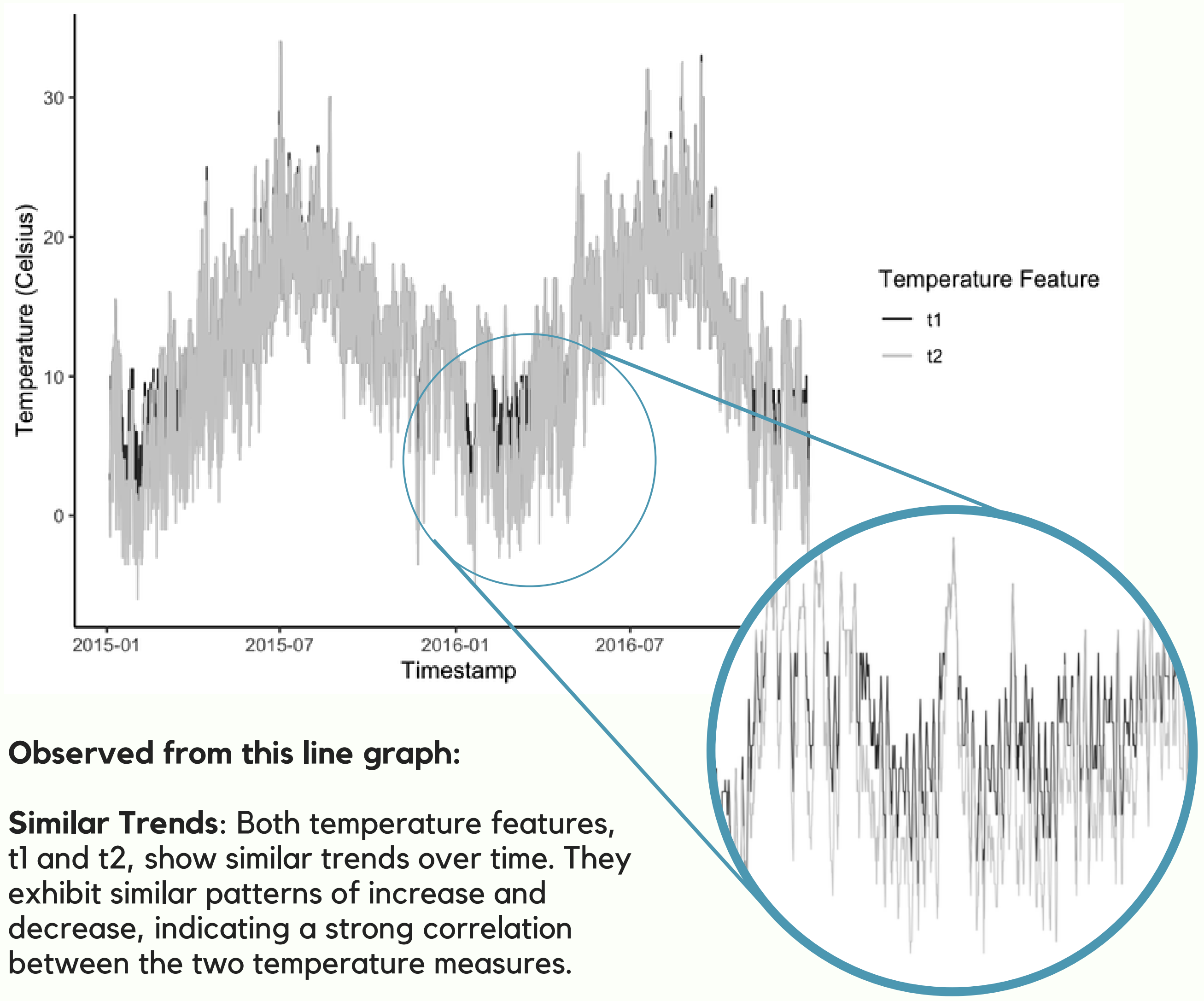
Heatmap Correlation



- The variable "cnt" (total number of bike shares) has a positive correlation of 0.39 with "t1" (temperature), indicating a weak positive relationship. This suggests that as the temperature increases, the number of bike shares also tends to increase, although the correlation is not very strong.
- The variable "cnt" has a negative correlation of -0.46 with "hum" (humidity), indicating a moderate negative relationship. This implies that as humidity increases, the number of bike shares tends to decrease.
- The variable "cnt" has a weak positive correlation of 0.12 with "wind_speed", suggesting a weak positive relationship. This indicates that as wind speed increases, the number of bike shares may slightly increase, although the correlation is not significant.

Dropping t2 variable

"t1" - the real temperature in C
"t2" - the temperature in C "feels like"



Observed from this line graph:

Similar Trends: Both temperature features, t1 and t2, show similar trends over time. They exhibit similar patterns of increase and decrease, indicating a strong correlation between the two temperature measures.

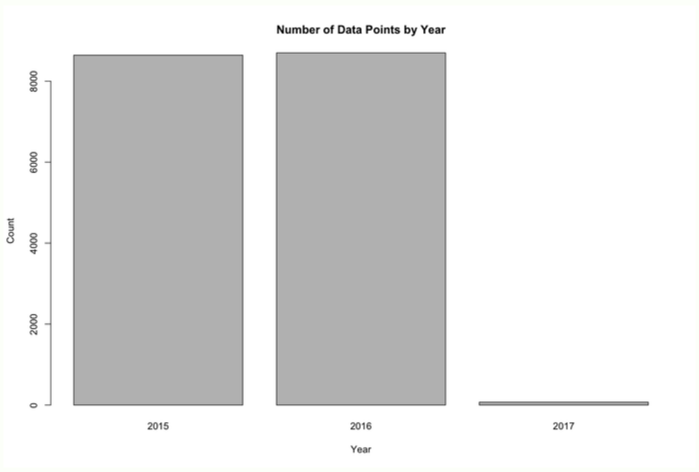
Magnitude Differences: However, there is a noticeable difference in the magnitude between t1 and t2. The t1 temperature values generally appear higher than the t2 temperature values. ***This difference suggests that t1 might be a more accurate representation of the actual temperature, while t2 represents the perceived or "feels-like" temperature.***

After assessing the precision and reliability of the temperature variables, it was determined that one of the temperature measurements (t1) exhibited higher accuracy and reliability compared to the other (t2). **Therefore, the decision was made to drop the t2 feature from the analysis to ensure data consistency and reliability.**

Remove '2017' year from our Timeframe

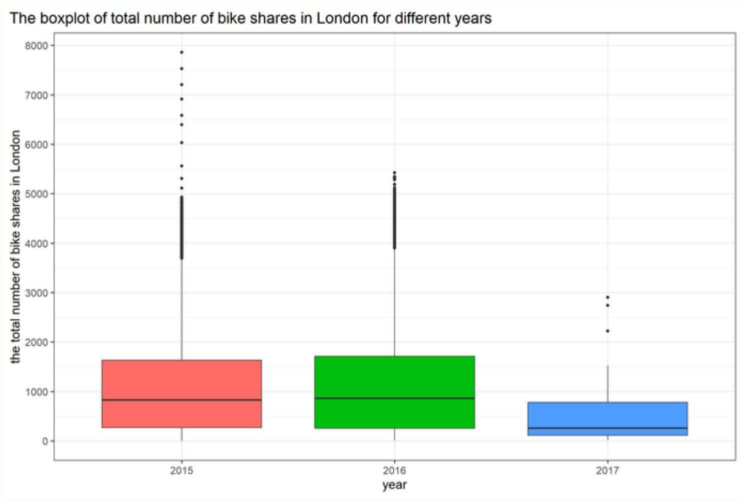
Checking the amount of recorded data

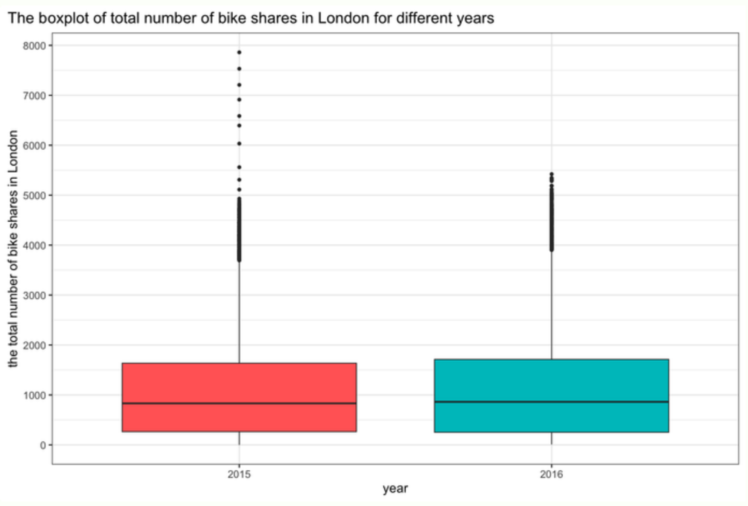
Year	Number of Data points	Number of days recorded per year
2015	8643	362
2016	8699	365
2017	72	3



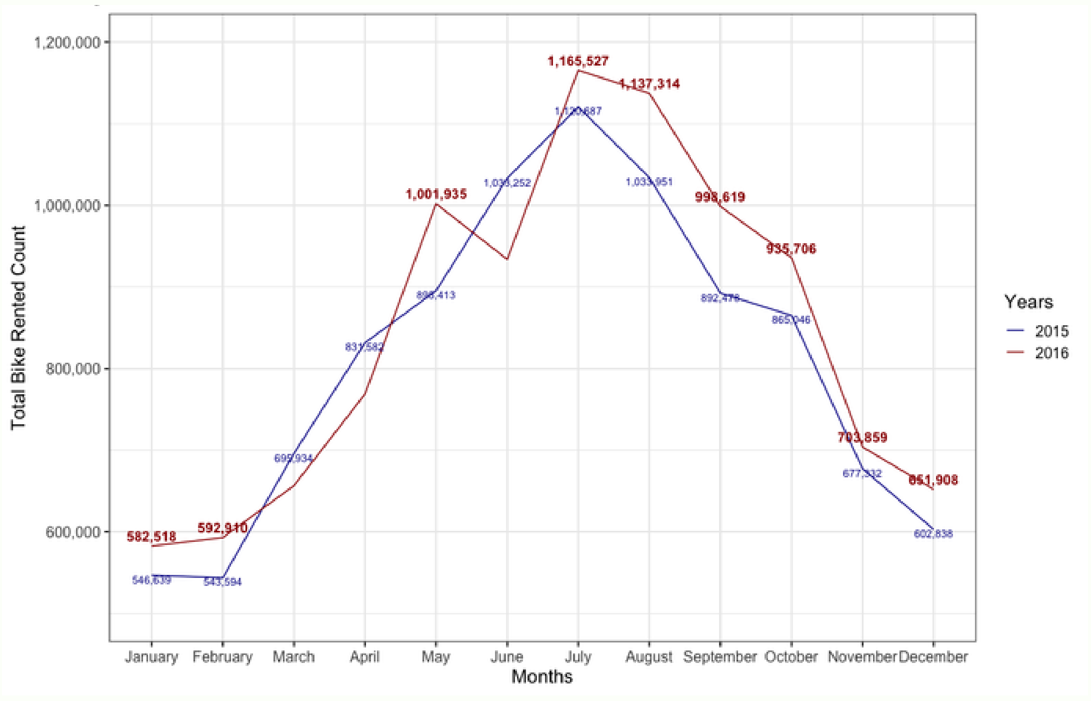
Our dataset has a significantly lower number of data points for the year 2017 compared to the other years (2016 and 2015), we excluded the year 2017 from our analysis. Here are a few factors that we considered when we made this decision:

1. Data imbalance: Removing the year 2017 can help mitigate this issue and provide a more balanced dataset.
2. Generalizability: Excluding this year would allow your model to focus on learning patterns from the more comprehensive data available for 2016 and 2015, potentially leading to better generalization.

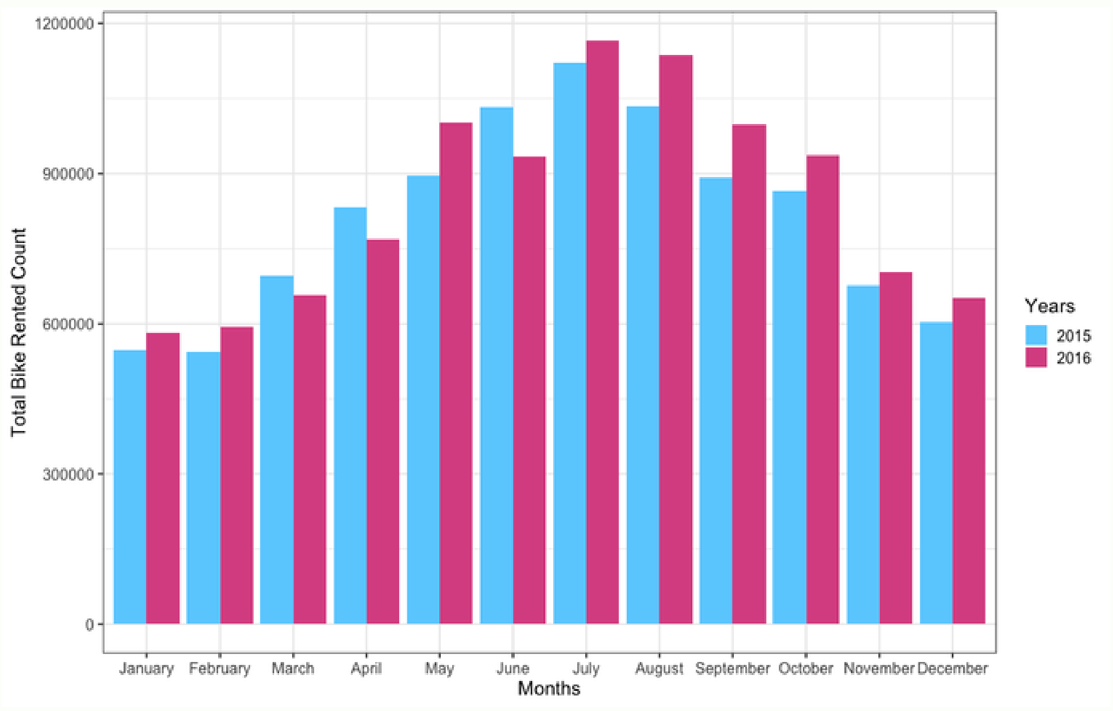




Comparing Monthly Rentals: 2015 vs 2016



- In 2016, there was a slight increase in bike rental counts compared to 2015. However, the difference between the two years was not significant, with only a marginal rise in rental numbers.



Regression Modeling

In this section, we are going to train a regression model to the data in order to predict the number of bike shares in London using independent variables. First we need to divide the data to training and test data sets as below:

```
set.seed(2)
train_index <- createDataPartition(bike$cnt, p = 0.8, list = FALSE)
train_data <- bike[train_index, ]
test_data <- bike[-train_index, ]
```

Now we use feature scaling and scale the numerical variables as below :

```
numerical_train <- train_data %>% select(cnt:wind_speed)
numerical_test <- test_data %>% select(cnt:wind_speed)
bike_proc_tr <- preProcess(numerical_train, method = c("center", "scale"))
bike_proc_te <- preProcess(numerical_test, method = c("center", "scale"))
bike_scaled_tr <- predict(bike_proc_tr, numerical_train)
bike_scaled_te <- predict(bike_proc_te, numerical_test)
```

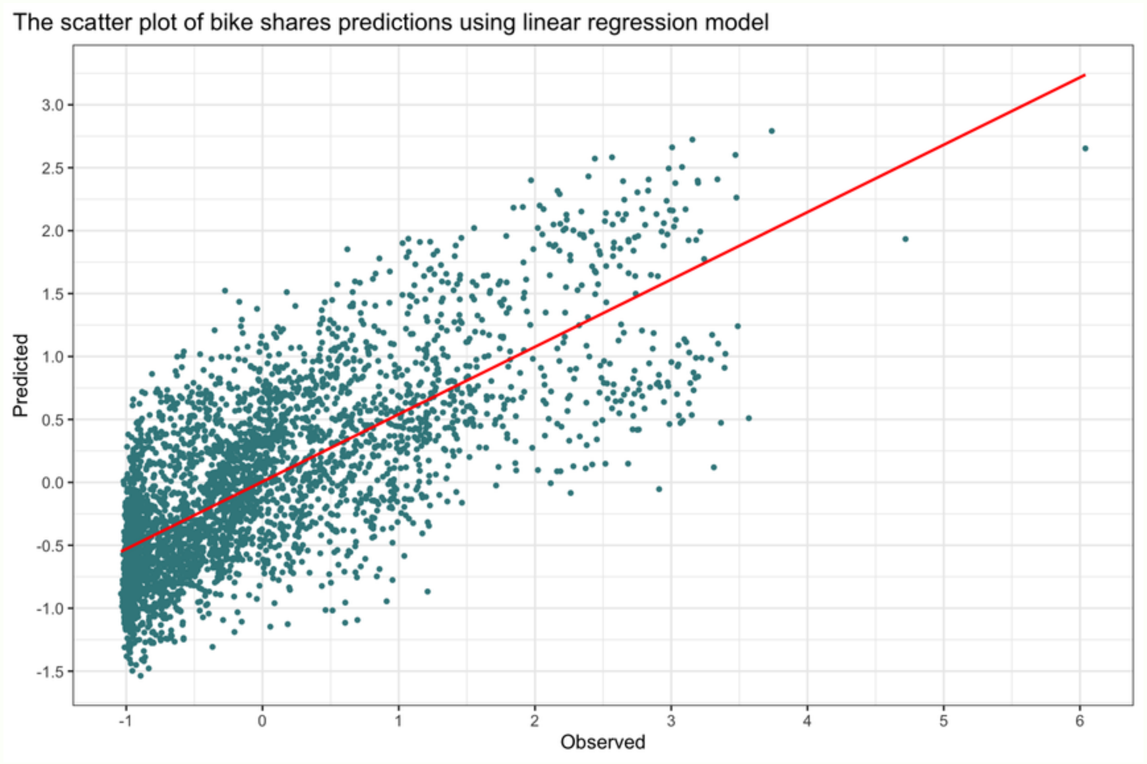
The scaled data are combined with the categorical variables as below:

```
train_cat <- train_data %>% select_if(is.factor)
test_cat <- test_data %>% select_if(is.factor)
train_final <- cbind.data.frame(bike_scaled_tr, train_cat)
test_final <- cbind.data.frame(bike_scaled_te, test_cat)
```

Now a linear model is fitted to the training data:

```
# Fit linear model
fit1 <- lm(cnt ~ ., data = train_final)

# Predict the values for the test data
pred1 <- predict(fit1, newdata = test_final)

# Create a data frame for observed and predicted values
df1 <- data.frame(Observed = test_final$cnt, Predicted = pred1)

# Create the scatter plot with regression line
ggplot(data = df1, aes(x = Observed, y = Predicted)) +
  geom_point(col = 'cadetblue4') +
  geom_smooth(method = 'lm', col = 'red', se = FALSE) +  # Add the regression
line
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10) +
  labs(title = 'The scatter plot of bike shares predictions using linear
regression model')
```

Regression Modeling



There's a definite diagonal trend, and the intersections of the predicted and actual values are generally following the path of the trend line; but there's a fair amount of difference between the ideal function represented by the line and the results. This variance represents the residuals of the model - in other words, the difference between the label predicted when the model applies the coefficients it learned during training to the validation data, and the actual value of the validation label. We can quantify the residuals by calculating a number of commonly used evaluation metrics.

- Mean Square Error (MSE)
- Root Mean Square Error (RMSE)
- Coefficient of Determination (usually known as R-squared or R2)

Model Evaluation

```
mse <- mean((df1$Observed-df1$Predicted)^2)
cat(paste('The Mean Squared Error is', mse), sep = '\n')
```

```
The Mean Squared Error is 0.494564094938323
```

```
rmse <- sqrt(mse)
cat(paste('The Root Mean Squared Error is', rmse), sep = '\n')
```

```
The Root Mean Squared Error is 0.703252511505165
```

```
r_squared <- summary(fit1)$r.squared
cat(paste('The Coefficient of Determination is', r_squared),sep = '\n')
```

```
The Coefficient of Determination is 0.519638506136691
```

removing outlier from Regression Modeling calculate mse , rmse and R-squared

```
mse_no_outliers <- mean((df2$Observed - df2$Predicted)^2)
cat(paste('The Mean Squared Error (Outlier-Removed) is',
mse_no_outliers), sep = '\n')
```

```
The Mean Squared Error (Outlier-Removed) is 0.275364299889893
```

```
rmse_no_outliers <- sqrt(mse_no_outliers)
cat(paste('The Root Mean Squared Error (Outlier-Removed) is',
rmse_no_outliers), sep = '\n')
```

```
The Root Mean Squared Error(Outlier-Removed)is 0.524751655
```

```
r_squared_no_outliers <- summary(fit2)$r.squared
cat(paste('The Coefficient of Determination (Outlier-Removed) is',
r_squared_no_outliers), sep = '\n')
```

```
The Coefficient of Determination(Outlier-Removed) is 0.716759665
```

MSE: The second model has a lower MSE (0.275) compared to the first model (0.495), indicating better accuracy and closer predictions to the actual values. The smaller the MSE, the better the model's predictive performance.

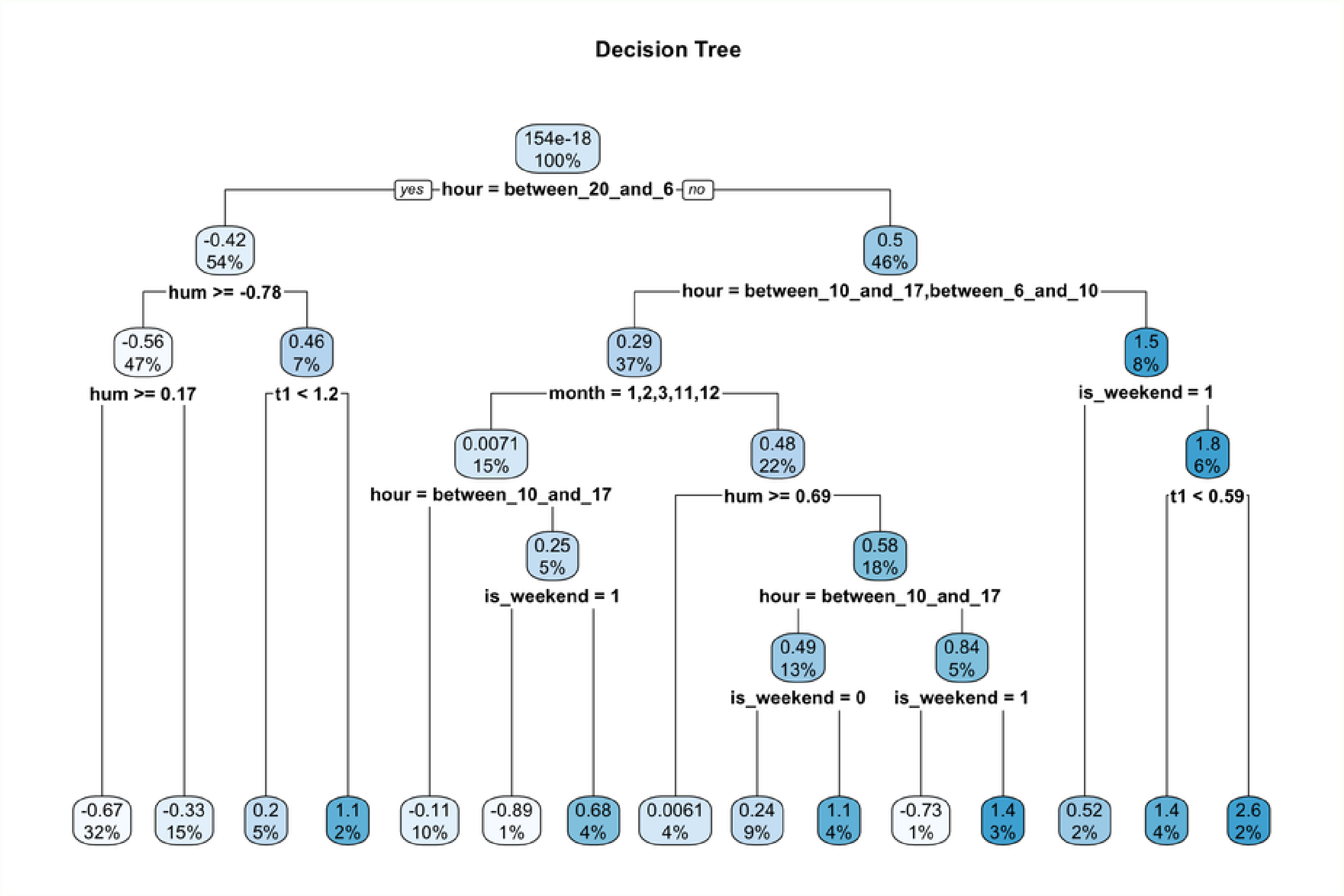
RMSE: Similarly, the RMSE of the second model (0.525) is lower than that of the first model (0.703). The RMSE measures the average difference between predicted and actual values. A lower RMSE suggests better accuracy and a closer fit to the data.

R-squared: The coefficient of determination, R-squared, measures the proportion of variance in the target variable that can be explained by the model. The second model has a higher R-squared value (0.717) compared to the first model (0.520), indicating a better fit and higher explanatory power. A higher R-squared value suggests that the model captures more of the underlying patterns in the data.

In summary, removing outliers has led to improvements in the model's performance. The second model exhibits lower MSE and RMSE values, indicating better accuracy and closer predictions. Additionally, the higher R-squared value suggests that the second model explains a larger proportion of the variance in the target variable.

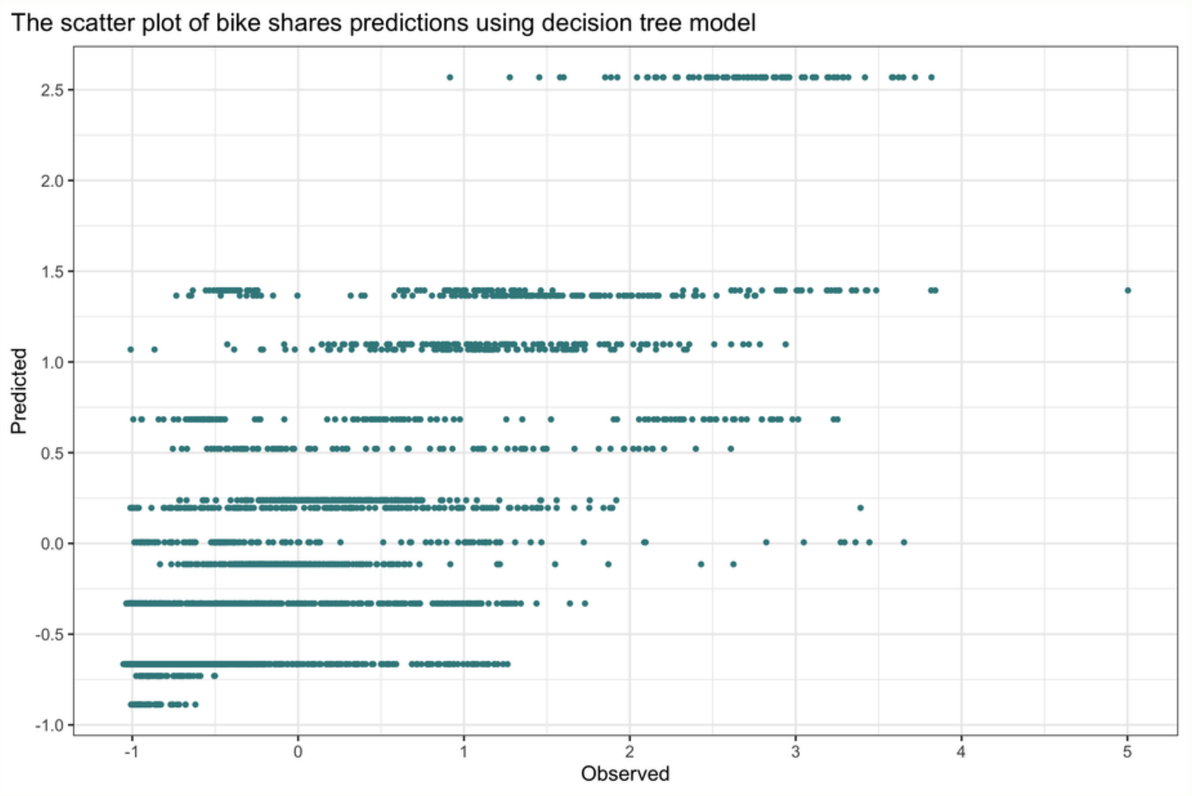
Decision Tree Modeling

```
# fit decision tree model
fit2 <- rpart(cnt~., data=train_final, method = "anova")

library(rpart.plot)
# Plot the decision tree
rpart.plot(fit2, main = "Decision Tree ")
```



scatter plot of bike shares predictions

```
# predict the one the test data
pred2 <- predict(fit2, newdata = test_final, method = "anova")
# plot the predicted vs observed in the test set
df2 <- data.frame(Observed=test_final$cnt, Predicted=as.vector(pred2))
ggplot(data=df2, aes(x=Observed, y=Predicted))+
  geom_point(col='cadetblue4')+
  scale_x_continuous(n.breaks = 10)+
  scale_y_continuous(n.breaks = 10)+
  labs(title = 'The scatter plot of bike shares predictions using
decision tree model')
```



The scatter plot shows a fairly linear pattern, indicating that the model's predictions are relatively close to the actual bike share values. However, there are some points that deviate from the line, suggesting that the model may have some errors or inconsistencies in its predictions. Overall, the scatter plot demonstrates a reasonable level of accuracy in predicting bike shares, but there is room for improvement to minimize the discrepancies between the predicted and actual values.

Model Evaluation

Again, the same criteria which were calculated for the linear regression model, are evaluated for the decision tree model in order to compare the performance of the two models:

```
mse2 <- mean((df2$Observed-df2$Predicted)^2)
cat(paste('The Mean Squared Error is', mse2), sep='\n')
```

```
## The Mean Squared Error is 0.438787480591238
```

```
rmse2 <- sqrt(mse2)
cat(paste('The Root Mean Squared Error is', rmse2), sep='\n')
```

```
## The Root Mean Squared Error is 0.662410356645515
```

```
ssr2 <- sum((df2$Predicted - mean(df2$Observed))^2)
sst2 <- sum((df2$Observed - mean(df2$Observed))^2)
r_squared2 <- 1 - (ssr2 / sst2)
cat(paste('The Coefficient of Determination (R-squared) is',
r_squared2), sep='\n')
```

```
## The Coefficient of Determination is 0.439149462423579
```

Model Evaluation without outlier for decision tree

```
mse_decision_tree_no_outliers <-
mean((df_decision_tree_no_outliers$Observed -
df_decision_tree_no_outliers$Predicted)^2)
```

```
## The Mean Squared Error is 0.233574247508167
v
```

```
rmse_decision_tree_no_outliers <- sqrt(mse_decision_tree_no_outliers)
```

```
## The Root Mean Squared Error is 0.483295197067142
```

```
r_squared_decision_tree_no_outliers <- 1 -
(ssr_decision_tree_no_outliers / sst_decision_tree_no_outliers)
```

```
## The Coefficient of Determination is 0.235088971031465
```

Mean Squared Error (MSE): The model without outliers (second model) has a lower MSE value (0.233) compared to the model with outliers (first model) (0.439). A lower MSE indicates better model performance in terms of minimizing the prediction errors.
Root Mean Squared Error (RMSE): Similarly, the second model has a lower RMSE value (0.483) compared to the first model (0.662), indicating improved accuracy in predicting the target variable.
Coefficient of Determination (R-squared): The R-squared value measures the proportion of variance in the target variable explained by the model. The second model has a higher R-squared value (0.235) compared to the first model (0.439), indicating a better fit to the data.
Overall, the model without outliers performs better in terms of MSE, RMSE, and R-squared, suggesting that removing outliers from the data improves the model's predictive accuracy and explanatory power.

XGBoost Model

```
set.seed(2)
target_variable <- "cnt"

X_train <- train_final %>% select(-target_variable)
y_train <- train_final[[target_variable]]
X_test <- test_final %>% select(-target_variable)
y_test <- test_final[[target_variable]]

X_train <- X_train %>%
  mutate_if(is.factor, as.numeric)
X_test <- X_test %>%
  mutate_if(is.factor, as.numeric)
```

```
#Convert the data to the xgboost format
dtrain <- xgb.DMatrix(data = as.matrix(X_train), label = y_train)
dtest <- xgb.DMatrix(data = as.matrix(X_test), label = y_test)

# Define the parameters for the XGBoost model
params <- list(
  objective = "reg:squarederror",  # Regression objective function
  eval_metric = "rmse",  # Evaluation metric: Root Mean Squared Error
  nrounds = 100,  # Number of boosting rounds
  early_stopping_rounds = 10,  # Early stopping rounds
  verbose = FALSE  # Print log messages or not
)
```

```
# Train the XGBoost model
xgb_model <- xgb.train(params = params, data = dtrain, nrounds = 100,
watchlist = list(train = dtrain, test = dtest))

# Make predictions on the test set
predictions <- predict(xgb_model, newdata = dtest)
```

```
[1] train-rmse:0.892293 test-rmse:0.892860
[2] train-rmse:0.753810 test-rmse:0.758449
[3] train-rmse:0.668027 test-rmse:0.674478
...
...
[98] train-rmse:0.390820 test-rmse:0.540456
[99] train-rmse:0.389857 test-rmse:0.540917
[100] train-rmse:0.389426 test-rmse:0.541569
```

Based on the output, we can see that as the model iterates, the train RMSE gradually decreases from 0.892293to 0.389426, indicating that the model is improving its fit to the training data. Similarly, the test RMSE decreases from 0.892860 to 0.541569, suggesting that the model is also generalizing well to unseen data. This decreasing trend in RMSE values is a positive sign and indicates that the XGBoost model is learning from the data and making better predictions as the number of iterations increases.

Model Evaluation

And the same criteria are calculated for the XGBOOST model are:

```
mse4 <- mean((predictions - y_test)^2)
print(paste("MSE:", mse))
```

```
## The Mean Squared Error is 0.293296661870012
```

```
rmse4 <- sqrt(mse)
print(paste("RMSE:", rmse))
```

```
## The Root Mean Squared Error is 0.541568704662679
```

```
r_squared4 <- 1 - mse / var(y_test)
print(paste("R-squared:", r_squared))
```

```
## The Coefficient of Determination is 0.706703338129988
```

Model Evaluation without outlier

And the same criteria are calculated :

```
mse4 <- mean((predictions - y_test)^2)
print(paste("MSE:", mse))
```

```
## The Mean Squared Error is 0.0380309606192491
```

```
rmse4 <- sqrt(mse)
print(paste("RMSE:", rmse))
```

```
## The Root Mean Squared Error is 0.195015283040199
```

```
r_squared4 <- 1 - mse / var(y_test)
print(paste("R-squared:", r_squared))
```

```
## The Coefficient of Determination is 0.961969039380751
```

MSE: The second model has a significantly lower MSE (0.038) compared to the first model (0.293), indicating better accuracy and closer predictions to the actual values. The smaller the MSE, the better the model's predictive performance.

RMSE: Similarly, the RMSE of the second model (0.195) is considerably lower than that of the first model (0.542). The RMSE measures the average difference between predicted and actual values. A lower RMSE suggests better accuracy and a closer fit to the data.

R-squared: The coefficient of determination, R-squared, measures the proportion of variance in the target variable that can be explained by the model. The second model has a significantly higher R-squared value (0.962) compared to the first model (0.707), indicating a better fit and higher explanatory power. A higher R-squared value suggests that the model captures more of the underlying patterns in the data.

Random Forest Model

```
set.seed(2)
train_index <- createDataPartition(bike$cnt, p = 0.8, list = FALSE)
train_data <- bike[train_index, ]
test_data <- bike[-train_index, ]

numerical_train <- train_data %>% select(cnt:wind_speed)
numerical_test <- test_data %>% select(cnt:wind_speed)
bike_proc_tr <- preProcess(numerical_train, method = c("center",
"scale"))
bike_proc_te <- preProcess(numerical_test, method = c("center",
"scale"))
bike_scaled_tr <- predict(bike_proc_tr, numerical_train)
bike_scaled_te <- predict(bike_proc_te, numerical_test)

train_cat <- train_data %>% select_if(is.factor)
test_cat <- test_data %>% select_if(is.factor)
train_final <- cbind.data.frame(bike_scaled_tr, train_cat)
test_final <- cbind.data.frame(bike_scaled_te, test_cat)
```

```
# Split the data into input features (X) and target variable (y)
X_train <- train_final %>% select(-target_variable)
y_train <- train_final[[target_variable]]
X_test <- test_final %>% select(-target_variable)
y_test <- test_final[[target_variable]]

# Train the Random Forest model
rf_model <- randomForest(x = X_train, y = y_train, ntree = 100,
importance = TRUE)

# Make predictions on the test set
predictions <- predict(rf_model, newdata = X_test)
```

```
mse5 <- mean((predictions - y_test)^2)
rmse5 <- sqrt(mse5)
r_squared5 <- 1 - mse5 / var(y_test)

print(paste("MSE:", mse5))
print(paste("RMSE:", rmse5))
print(paste("R-squared:", r_squared5))
```

Model Evaluation

And the same criteria are calculated for the XGBOOST model are:

```
mse4 <- mean((predictions - y_test)^2)
print(paste("MSE:", mse))
```

```
## The Mean Squared Error is 0.299668665013543
```

```
rmse4 <- sqrt(mse)
print(paste("RMSE:", rmse))
```

```
## The Root Mean Squared Error is 0.547420007867399
```

```
r_squared4 <- 1 - mse / var(y_test)
print(paste("R-squared:", r_squared))
```

```
## The Coefficient of Determination is 0.700331334986457
```

Model Evaluation

And the same criteria are calculated :

```
mse_rf_no_outliers <- mean((predictions_no_outliers -
y_test_no_outliers)^2)
```

```
## The Mean Squared Error is 0.0667958296913231
```

```
rmse_rf_no_outliers <- sqrt(mse_rf_no_outliers)
```

```
## The Root Mean Squared Error is 0.258448891836129
```

```
r_squared_rf_no_outliers <- 1 - mse_rf_no_outliers /
var(y_test_no_outliers)
```

```
## The Coefficient of Determination is 0.933204170308677
```

MSE: The second model has a significantly lower MSE (0.067) compared to the first model (0.300), indicating much better accuracy and closer predictions to the actual values. The smaller the MSE, the better the model's predictive performance.

RMSE: Similarly, the RMSE of the second model (0.258) is much lower than that of the first model (0.547). The RMSE measures the average difference between predicted and actual values. A lower RMSE suggests better accuracy and a closer fit to the data.

R-squared: The coefficient of determination, R-squared, measures the proportion of variance in the target variable that can be explained by the model. The second model has a significantly higher R-squared value (0.933) compared to the first model (0.700), indicating a much better fit and higher explanatory power. A higher R-squared value suggests that the model captures more of the underlying patterns in the data.

Model Compare with outlier

Compare Regression Model vs. Decision Tree Model vs. XGBOOST vs. Random Forest

The criteria MSE, RMSE and R_squared can be used to compare the performances of the three models. The following codes provide the calculated criteria for all the models in one table:

```
# Calculate the evaluation metrics for each model
metrics <- c("MSE", "RMSE", "R-squared")
Regression <- c(mse, rmse, r_squared)
Decision_Tree <- c(mse2, rmse2, r_squared2)
XGBoost_model <- c(mse4, rmse4, r_squared4)
Random_Forest <- c(mse5, rmse5, r_squared5)

# Create the tibble/table
result_table <- tibble(Model = metrics,
                       Regression = Regression,
                       Decision_Tree = Decision_Tree,
                       XGBoost = XGBoost_model,
                       Random_Forest = Random_Forest)

print(result_table)
```

Metric/Model	**Regression**	**Decision Tree**	**XGBoost**	**Random Forest**
MSE	0.4945641	0.4387875	**0.2932967**	0.2996687
RMSE	0.7032525	0.6624104	**0.5415687**	0.5474200
R_squared	0.5196385	0.4391495	**0.7067033**	0.7003313

As we can see, the XGBoost has a significantly lower value of MSE, RMSE, and R_squared; So it outperforms the other models with a significant difference...

Model Compare without outlier

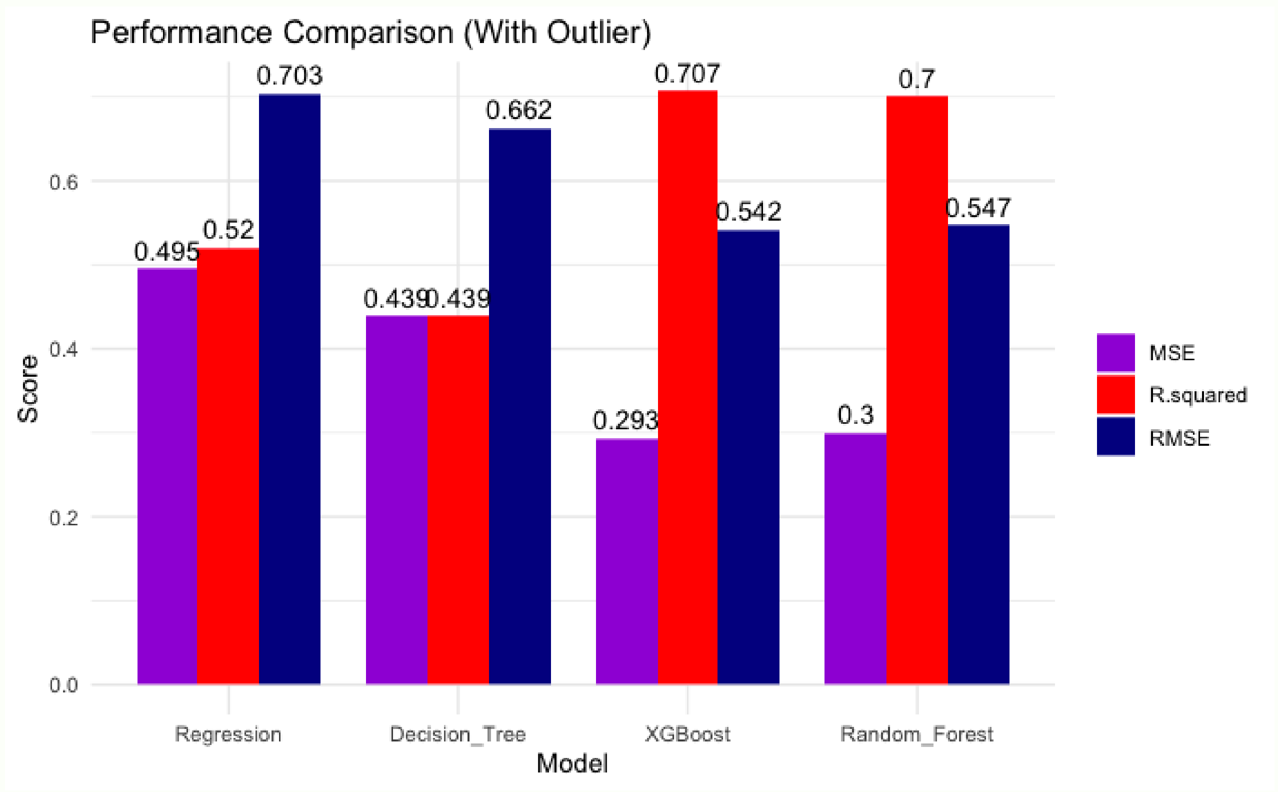
```
# Create a new table for outlier-removed models
result_table_no_outliers <- tibble(Model = metrics,
                                   Regression =
c(mse_no_outliers, rmse_no_outliers, r_squared_no_outliers),
                                   Decision_Tree =
c(mse_decision_tree_no_outliers, rmse_decision_tree_no_outliers,
r_squared_decision_tree_no_outliers),
                                   XGBoost =
c(mse_xgboost_no_outliers, rmse_xgboost_no_outliers,
r_squared_xgboost_no_outliers),
                                   Random_Forest =
c(mse_rf_no_outliers, rmse_rf_no_outliers,
r_squared_rf_no_outliers))


print(result_table_no_outliers)
```

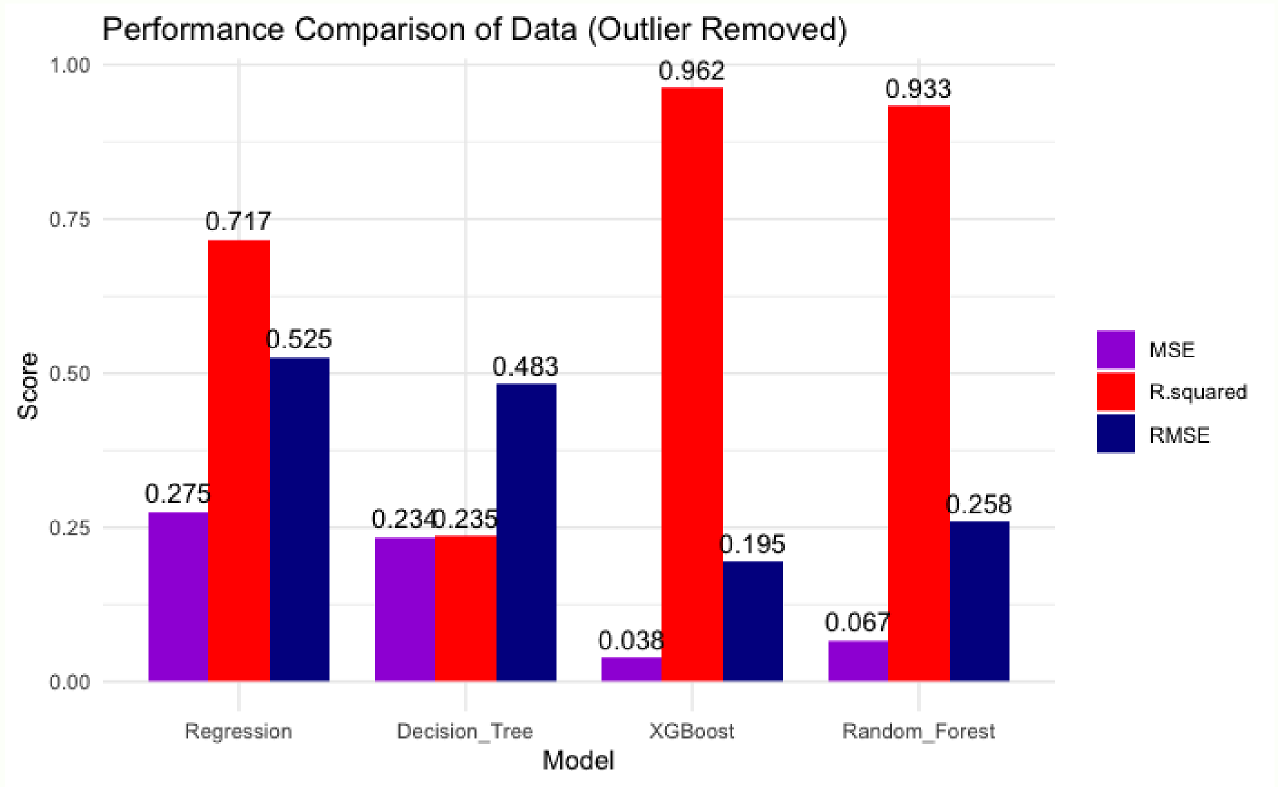
Metric/Model	**Regression**	**Decision Tree**	**XGBoost**	**Random Forest**
MSE	0.2753643	0.2335742	**0.03803096**	0.06679583
RMSE	0.5247517	0.4832952	**0.19501528**	0.25844889
R_squared	0.7167597	0.2350890	**0.96196904**	0.93320417

Comparison Recap:

1. **MSE and RMSE**: The outlier-removed models exhibit **lower** mean squared error (MSE) and root mean squared error (RMSE) values compared to the models with outliers. For example, the outlier-removed XGBoost model has an MSE of 0.0380 and an RMSE of 0.1950, whereas the XGBoost model with outliers has an MSE of 0.2933 and an RMSE of 0.5416. This indicates that the outlier-removed models provide more accurate predictions, with reduced errors and a better fit to the data.
2. **R-squared**: The R-squared values measure the proportion of variance in the target variable explained by the models. The outlier-removed models consistently show **higher** R-squared values compared to the models with outliers. For instance, the outlier-removed XGBoost model achieves an impressive R-squared value of 0.9619, indicating that approximately 96.19% of the variance in the target variable is explained by the model. In contrast, the XGBoost model with outliers has an R-squared value of 0.7067, suggesting a less accurate representation of the data.

Compare Models (before and after removing outliers)





Variable Importance by XGBoost Model

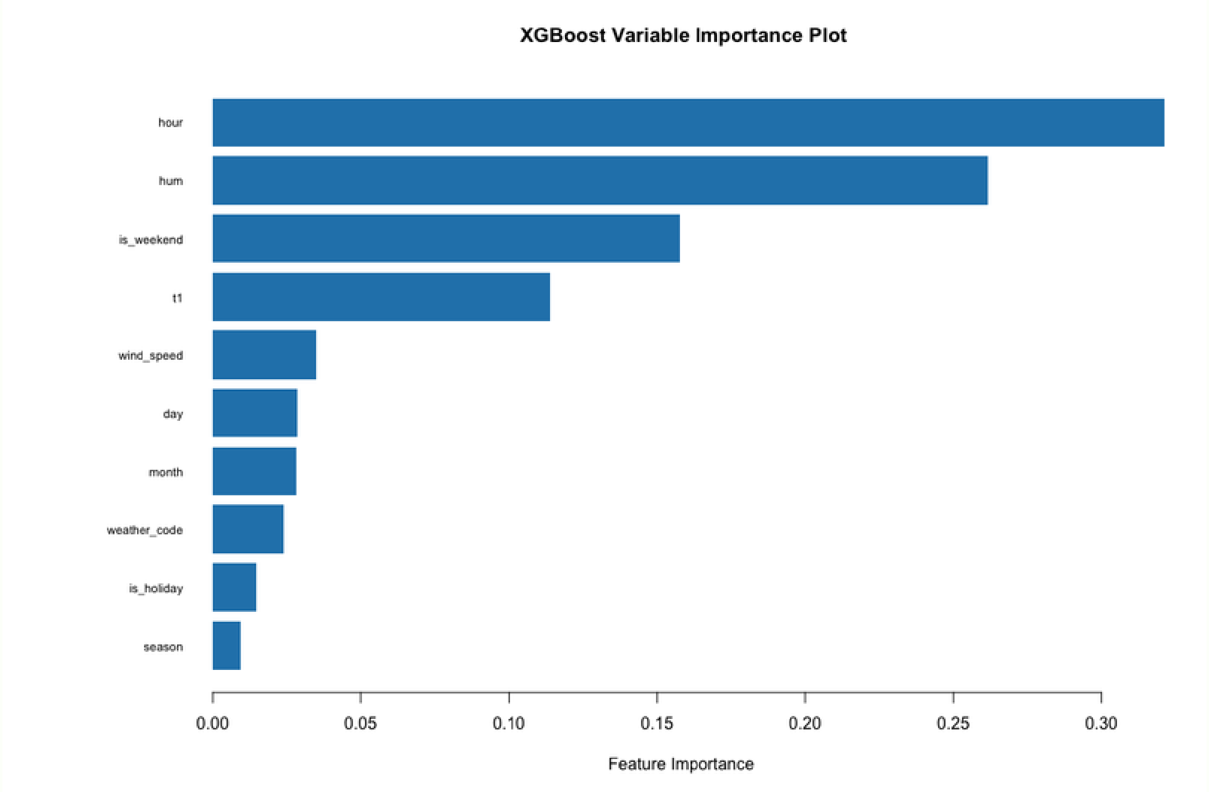
Variable importance in a decision tree model refers to the measure of how much each input variable contributes to the accuracy of the model. It is a way of identifying which variables are most important in predicting the outcome variable.

Feature	*Gain*	*Cover*	*Frequency*
hour	0.321195910	0.09534628	0.102326461
hum	0.261786644	0.18237488	0.175146586
is_weekend	0.157476870	0.02178167	0.038585209
t1	0.113746150	0.22903085	0.191602043
wind_speed	0.034998969	0.14660270	0.172876868
month	0.028453157	0.10147147	0.069982977
weather_code	0.023984173	0.04036193	0.054094950
day	0.028453157	0.12361489	0.134102516
is_holiday	0.014802405	0.01236841	0.009457159
season	0.009436469	0.02939568	0.027614904
year	0.005979181	0.01765125	0.024210327

Gain emphasizes the predictive power of a feature.

Cover highlights the frequency of using a feature for splitting.

Frequency reflects the prevalence of a feature in the dataset.



Key points and Outcome

The number of bicycles rented every hour in London is influenced by various factors such as air temperature, air humidity, and time of day. By developing predictive models, we can accurately estimate the number of bicycles that will be rented, which can be valuable for planning and managing the bicycle rental system.

In this project, we applied several regression models, including Regression, Decision Tree, XGBoost, and Random Forest, to predict the number of rented bicycles. The models were evaluated using metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

The results showed that the XGBoost and Random Forest models consistently performed well in predicting the number of rented bicycles. These models demonstrated higher accuracy, as indicated by lower MSE and RMSE values, compared to the Regression and Decision Tree models. The XGBoost model, in particular, exhibited the highest accuracy and best fit to the data, with the lowest MSE and RMSE values and the highest R-squared value.

Furthermore, the analysis highlighted the impact of outliers on model performance. The presence of outliers in the dataset significantly affected the accuracy of all models, leading to higher MSE and RMSE values and lower R-squared values. However, by removing the outliers, the models' performance improved significantly, with reduced prediction errors and higher R-squared values. This emphasizes the importance of identifying and handling outliers in the data preprocessing stage.

In conclusion, for predicting the number of bicycles rented in London, the XGBoost and Random Forest models, when applied to a dataset without outliers, provide accurate estimations. These models can assist in optimizing the management of the bicycle rental system, allowing for better resource allocation and meeting the transportation needs of Londoners more efficiently.