Bike Rental Prediction

Description: In bike-sharing systems, the entire process from membership to rental and return has been automated. Using these systems, users can easily rent a bike from one location and return it to another. Hence, a bike rental company wants to understand and predict the number of bikes rented daily based on the environment and seasons.

Objective: The objective of this case is to predict bike rental counts based on environmental and seasonal settings with the help of a machine learning algorithm.

1. Exploratory data analysis

- Load dataset and libraries

```
setwd(choose.dir())

install.packages("readxl")

library(readxl)

install.packages("dplyr")

library(dplyr)

install.packages("ggplot2")

library(ggplot2)

install.packages("caret")

library(caret)

install.packages("randomForest")

library(randomForest)


Bike_rental_data <- read_excel("day.xlsx")


View(Bike_rental_data)
```

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1	2011-01-01	1	0	1	0	6	0	2	0.3441670	0.3636250	0.805833	0.1604460	331	654	985
2	2	2011-01-02	1	0	1	0	0	0	2	0.3634780	0.3537390	0.696087	0.2485390	131	670	801
3	3	2011-01-03	1	0	1	0	1	1	1	0.1963640	0.1894050	0.437273	0.2483090	120	1229	1349
4	4	2011-01-04	1	0	1	0	2	1	1	0.2000000	0.2121220	0.590435	0.1602960	108	1454	1562
5	5	2011-01-05	1	0	1	0	3	1	1	0.2269570	0.2292700	0.436957	0.1869000	82	1518	1600
6	6	2011-01-06	1	0	1	0	4	1	1	0.2043480	0.2332090	0.518261	0.0895652	88	1518	1606
7	7	2011-01-07	1	0	1	0	5	1	2	0.1965220	0.2088390	0.498696	0.1687260	148	1362	1510
8	8	2011-01-08	1	0	1	0	6	0	2	0.1650000	0.1622540	0.535833	0.2668040	68	891	959
9	9	2011-01-09	1	0	1	0	0	0	1	0.1383330	0.1161750	0.434167	0.3619500	54	768	822
10	10	2011-01-10	1	0	1	0	1	1	1	0.1508330	0.1508880	0.482917	0.2232670	41	1280	1321
11	11	2011-01-11	1	0	1	0	2	1	2	0.1690910	0.1914640	0.686364	0.1221320	43	1220	1263
12	12	2011-01-12	1	0	1	0	3	1	1	0.1727270	0.1604730	0.599545	0.3046270	25	1137	1162
13	13	2011-01-13	1	0	1	0	4	1	1	0.1650000	0.1508830	0.470417	0.3010000	38	1368	1406

summary(Bike_rental_data)

```
> summary(Bike_rental_data)
    instant          dteday                       season           yr
 Min.   :  1.0   Min.   :2011-01-01 00:00:00   Min.   :1.000   Min.   :0.0000
 1st Qu.:183.5   1st Qu.:2011-07-02 12:00:00   1st Qu.:2.000   1st Qu.:0.0000
 Median :366.0   Median :2012-01-01 00:00:00   Median :3.000   Median :1.0000
 Mean   :366.0   Mean   :2012-01-01 00:00:00   Mean   :2.497   Mean   :0.5007
 3rd Qu.:548.5   3rd Qu.:2012-07-01 12:00:00   3rd Qu.:3.000   3rd Qu.:1.0000
 Max.   :731.0   Max.   :2012-12-31 00:00:00   Max.   :4.000   Max.   :1.0000
      mnth          holiday           weekday        workingday      weathersit
 Min.   : 1.00   Min.   :0.00000   Min.   :0.000   Min.   :0.000   Min.   :1.000
 1st Qu.: 4.00   1st Qu.:0.00000   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:1.000
 Median : 7.00   Median :0.00000   Median :3.000   Median :1.000   Median :1.000
 Mean   : 6.52   Mean   :0.02873   Mean   :2.997   Mean   :0.684   Mean   :1.395
 3rd Qu.:10.00   3rd Qu.:0.00000   3rd Qu.:5.000   3rd Qu.:1.000   3rd Qu.:2.000
 Max.   :12.00   Max.   :1.00000   Max.   :6.000   Max.   :1.000   Max.   :3.000
      temp             atemp              hum          windspeed
 Min.   :0.05913   Min.   :0.07907   Min.   :0.0000   Min.   :0.02239
 1st Qu.:0.33708   1st Qu.:0.33784   1st Qu.:0.5200   1st Qu.:0.13495
 Median :0.49833   Median :0.48673   Median :0.6267   Median :0.18097
 Mean   :0.49538   Mean   :0.47435   Mean   :0.6279   Mean   :0.19049
 3rd Qu.:0.65542   3rd Qu.:0.60860   3rd Qu.:0.7302   3rd Qu.:0.23321
 Max.   :0.86167   Max.   :0.84090   Max.   :0.9725   Max.   :0.50746
     casual        registered         cnt
 Min.   :   2.0   Min.   :  20   Min.   :  22
 1st Qu.: 315.5   1st Qu.:2497   1st Qu.:3152
 Median : 713.0   Median :3662   Median :4548
 Mean   : 848.2   Mean   :3656   Mean   :4504
 3rd Qu.:1096.0   3rd Qu.:4776   3rd Qu.:5956
 Max.   :3410.0   Max.   :6946   Max.   :8714
```

str(Bike_rental_data)

```
> str(Bike_rental_data)
tibble [731 × 16] (S3: tbl_df/tbl/data.frame)
 $ instant   : num [1:731] 1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : POSIXct[1:731], format: "2011-01-01" "2011-01-02" ...
 $ season    : num [1:731] 1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : num [1:731] 0 0 0 0 0 0 0 0 0 0 ...
 $ mnth      : num [1:731] 1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : num [1:731] 0 0 0 0 0 0 0 0 0 0 ...
 $ weekday   : num [1:731] 6 0 1 2 3 4 5 6 0 1 ...
 $ workingday: num [1:731] 0 0 1 1 1 1 1 0 0 1 ...
 $ weathersit: num [1:731] 2 2 1 1 1 1 2 2 1 1 ...
 $ temp      : num [1:731] 0.344 0.363 0.196 0.2 0.227 ...
 $ atemp     : num [1:731] 0.364 0.354 0.189 0.212 0.229 ...
 $ hum       : num [1:731] 0.806 0.696 0.437 0.59 0.437 ...
 $ windspeed : num [1:731] 0.16 0.249 0.248 0.16 0.187 ...
 $ casual    : num [1:731] 331 131 120 108 82 88 148 68 54 41 ...
 $ registered: num [1:731] 654 670 1229 1454 1518 ...
 $ cnt       : num [1:731] 985 801 1349 1562 1600 ...
>
```

- **Perform data type conversion of the attributes**

```
str(Bike_rental_data)

Bike_rental_data1 <- Bike_rental_data %>%
  mutate(instant=as.integer(instant),
         dteday=as.Date(dteday),
         season = as.factor(season),
         yr=as.factor(yr),
         mnth=as.factor(mnth),
         holiday=as.factor(holiday),
         weekday=as.factor(weekday),
         workingday=as.factor(workingday),
         weathersit=as.factor(weathersit)
         )

str(Bike_rental_data1)
```

```
> str(Bike_rental_data1)
tibble [731 × 16] (S3: tbl_df/tbl/data.frame)
 $ instant   : int [1:731] 1 2 3 4 5 6 7 8 9 10 ...
 $ dteday    : Date[1:731], format: "2011-01-01" "2011-01-02" ...
 $ season    : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ yr        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ mnth      : Factor w/ 12 levels "1","2","3","4",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ holiday   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ weekday   : Factor w/ 7 levels "0","1","2","3",..: 7 1 2 3 4 5 6 7 1 2 ...
 $ workingday: Factor w/ 2 levels "0","1": 1 1 2 2 2 2 2 1 1 2 ...
 $ weathersit: Factor w/ 3 levels "1","2","3": 2 2 1 1 1 1 2 2 1 1 ...
 $ temp      : num [1:731] 0.344 0.363 0.196 0.2 0.227 ...
 $ atemp     : num [1:731] 0.364 0.354 0.189 0.212 0.229 ...
 $ hum       : num [1:731] 0.806 0.696 0.437 0.59 0.437 ...
 $ windspeed : num [1:731] 0.16 0.249 0.248 0.16 0.187 ...
 $ casual    : num [1:731] 331 131 120 108 82 88 148 68 54 41 ...
 $ registered: num [1:731] 654 670 1229 1454 1518 ...
 $ cnt       : num [1:731] 985 801 1349 1562 1600 ...
```

- **Carry out the missing value analysis**

```
missing_values <- Bike_rental_data1 %>%
 summarise_all(~sum(is.na(.)))

print(missing_values)
```

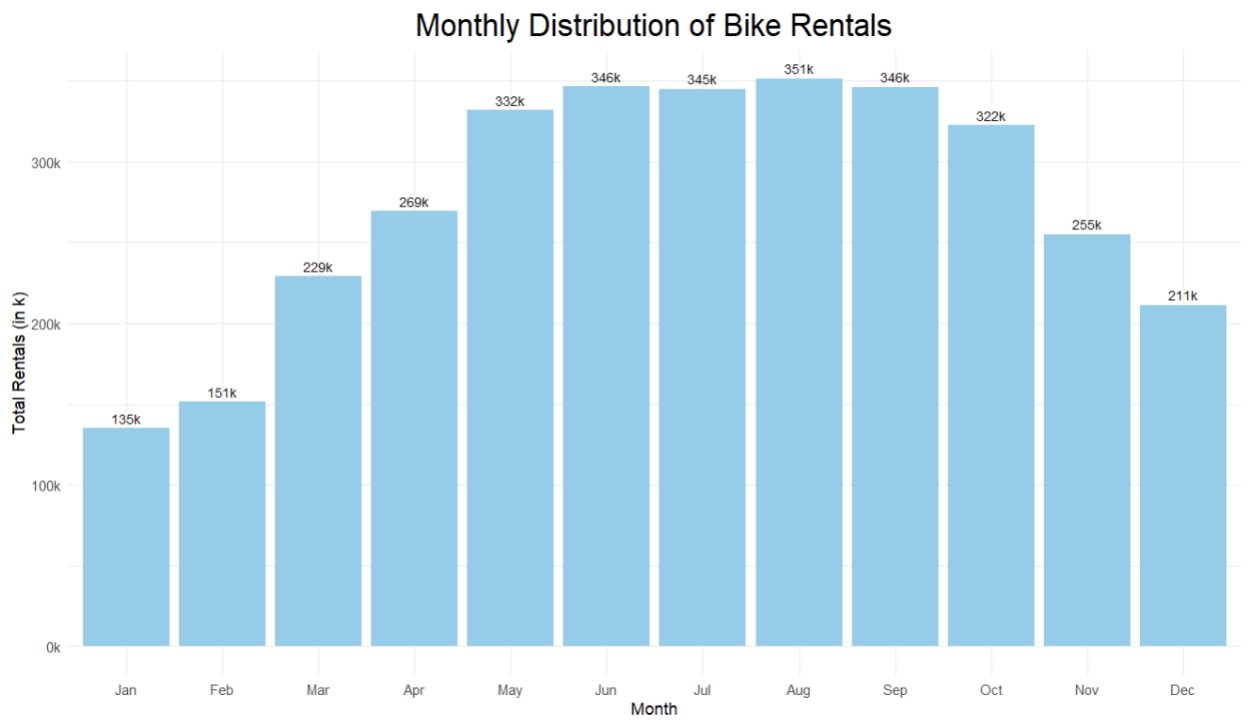
```
> print(missing_values)
# A tibble: 1 × 16
  instant dteday season    yr  mnth holiday weekday workingday weathersit  temp atemp
    <int>  <int>  <int> <int> <int>   <int>   <int>      <int>      <int> <int> <int>
1       0      0      0     0     0       0       0          0          0     0     0
# i 5 more variables: hum <int>, windspeed <int>, casual <int>, registered <int>,
#   cnt <int>
> |
```

2. Attributes distributions and trends

- Plot monthly distribution of the total number of bikes rented

```
monthly_rentals <- Bike_rental_data1 %>%
 group_by(mnth) %>%
 summarise(total_rentals=sum(cnt))
```

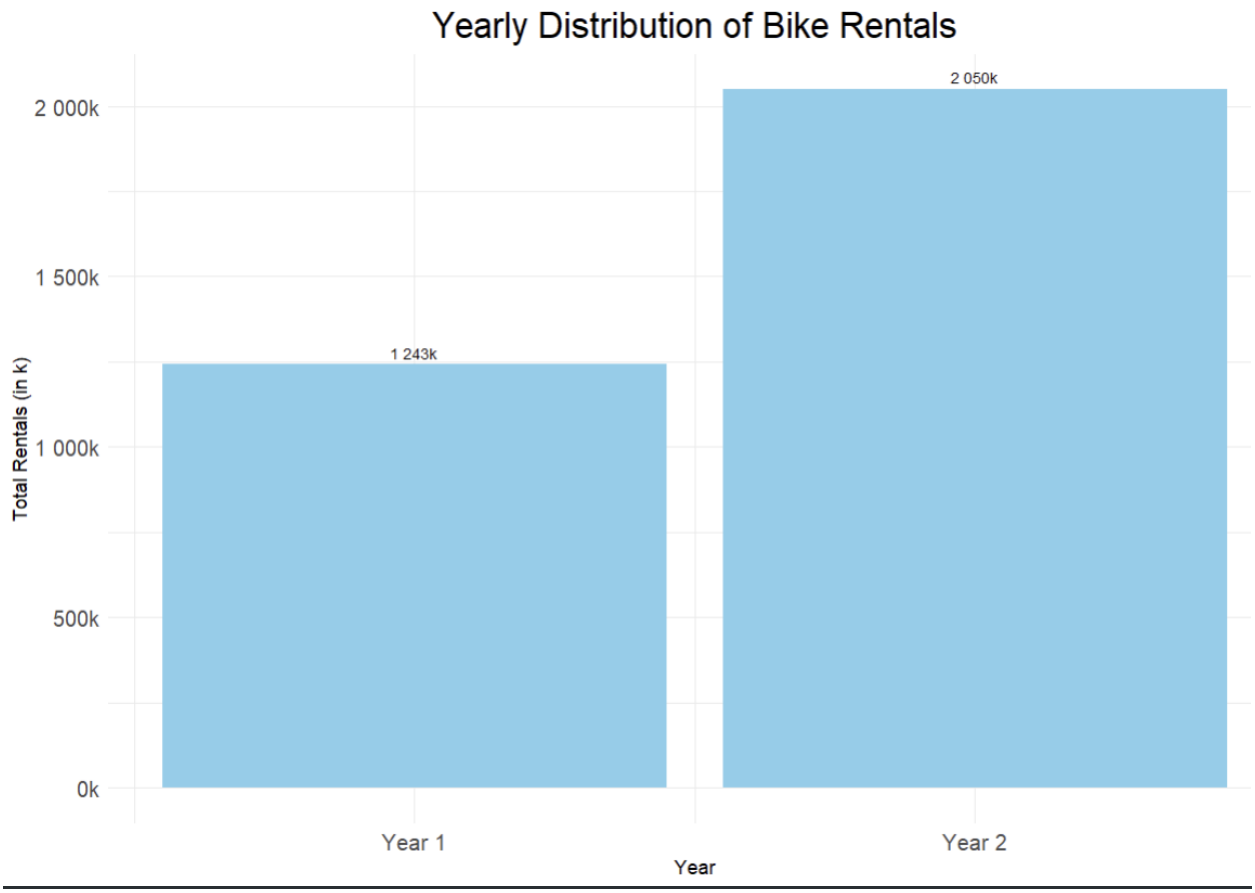
```
ggplot(monthly_rentals, aes(x = mnth, y = total_rentals)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(
    aes(label = scales::number_format(scale = 1e-3, accuracy = 1, suffix = "k")(total_rentals)),
    vjust = -0.5,
    size = 3,
    color = "black"
  ) + # Add labels in thousands (k, without decimals) on top of bars
  labs(
    title = "Monthly Distribution of Bike Rentals",
    x = "Month",
    y = "Total Rentals (in k)"
  ) +
  scale_x_discrete(labels = c(
    "1" = "Jan", "2" = "Feb", "3" = "Mar", "4" = "Apr",
    "5" = "May", "6" = "Jun", "7" = "Jul", "8" = "Aug",
    "9" = "Sep", "10" = "Oct", "11" = "Nov", "12" = "Dec"
  )) +
  scale_y_continuous(
    labels = scales::number_format(scale = 1e-3, accuracy = 1, suffix = "k")
  ) + # Format Y-axis labels in thousands (k, without decimals) with "k" suffix
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20, hjust = 0.5) # Adjust size and center title
  )
```



- **Plot yearly distribution of the total number of bikes rented**

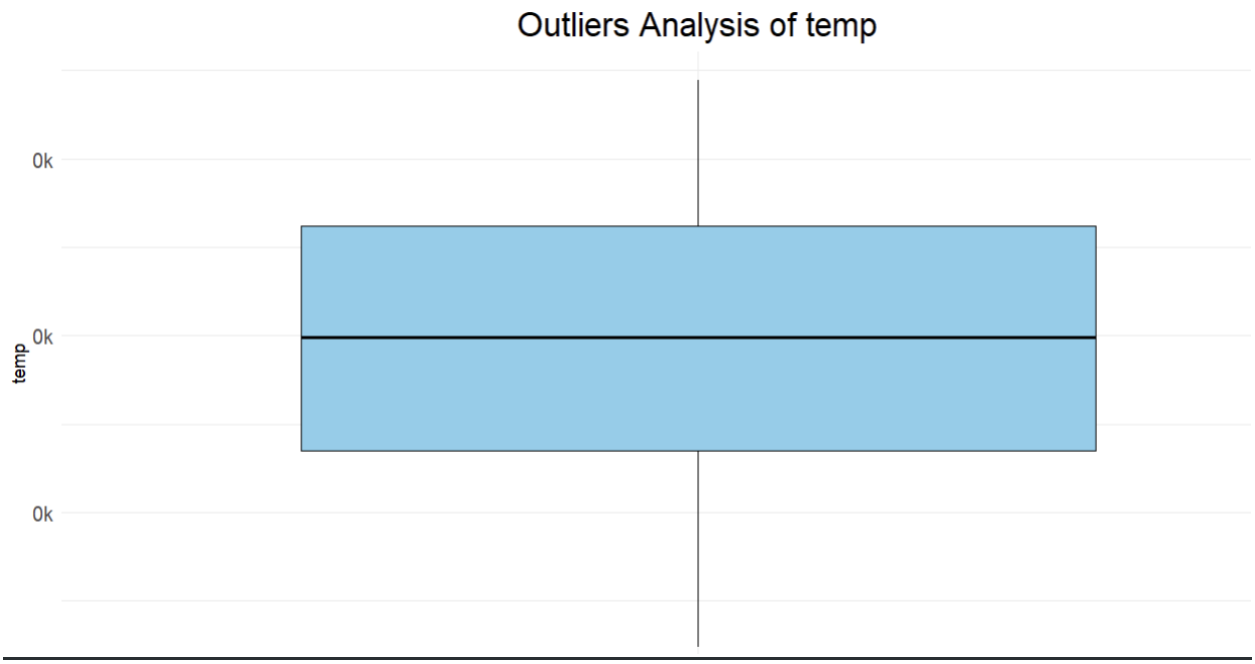
```
Bike_rental_data1 <- Bike_rental_data1 %>%
  mutate(yr = as.numeric(yr))
yearly_rentals <- Bike_rental_data1 %>%
  group_by(yr) %>%
  summarise(total_rentals = sum(cnt))


ggplot(yearly_rentals, aes(x = yr, y = total_rentals)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = scales::number_format(scale = 1e-3, accuracy = 1, suffix = "k")(total_rentals)),
            vjust = -0.5, size = 3, color = "black") + # Add labels on top of bars
  labs(
    title = "Yearly Distribution of Bike Rentals",
```

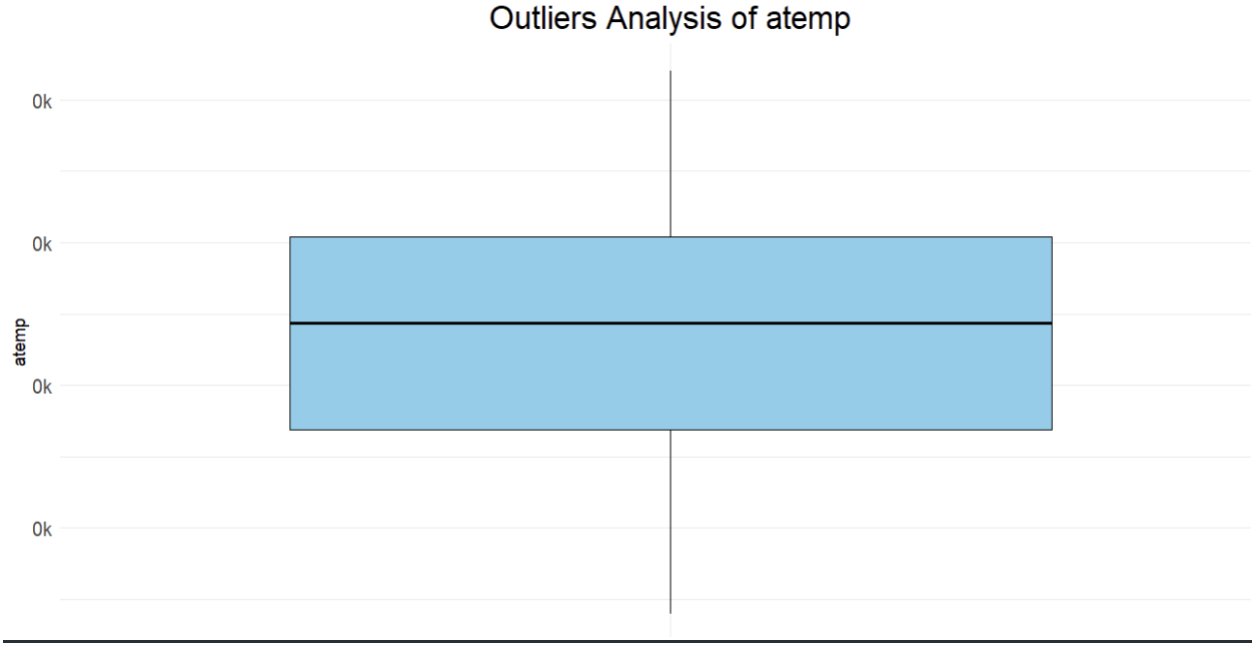
```
  x = "Year",
  y = "Total Rentals (in k)"
) +
scale_x_continuous(
  labels = c("Year 1", "Year 2"), # Specify custom labels
  breaks = 1:2 # Specify the breaks for the custom labels
) +
scale_y_continuous(labels = scales::number_format(scale = 1e-3, accuracy = 1, suffix = "k")) +
theme_minimal() +
theme(
  plot.title = element_text(size = 20, hjust = 0.5), # Adjust title size and center it
  axis.text.x = element_text(size = 12), # Adjust X-axis label font size
  axis.text.y = element_text(size = 12) # Adjust Y-axis label font size
)
```

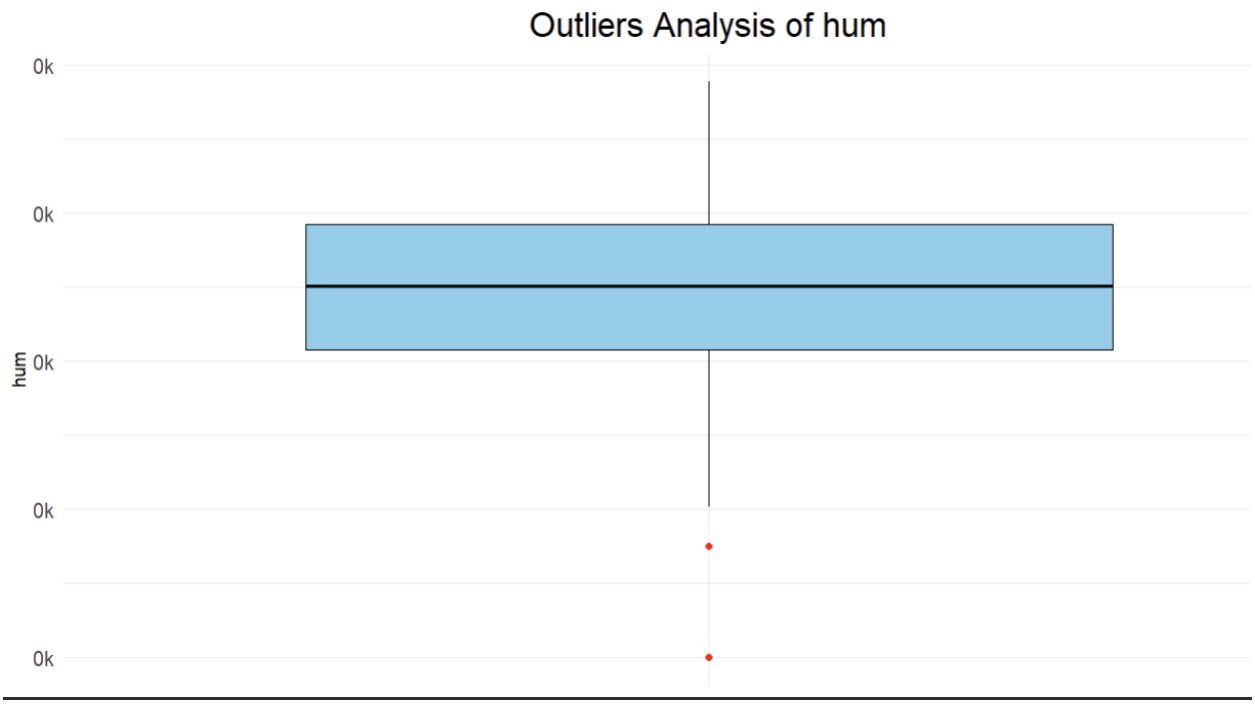


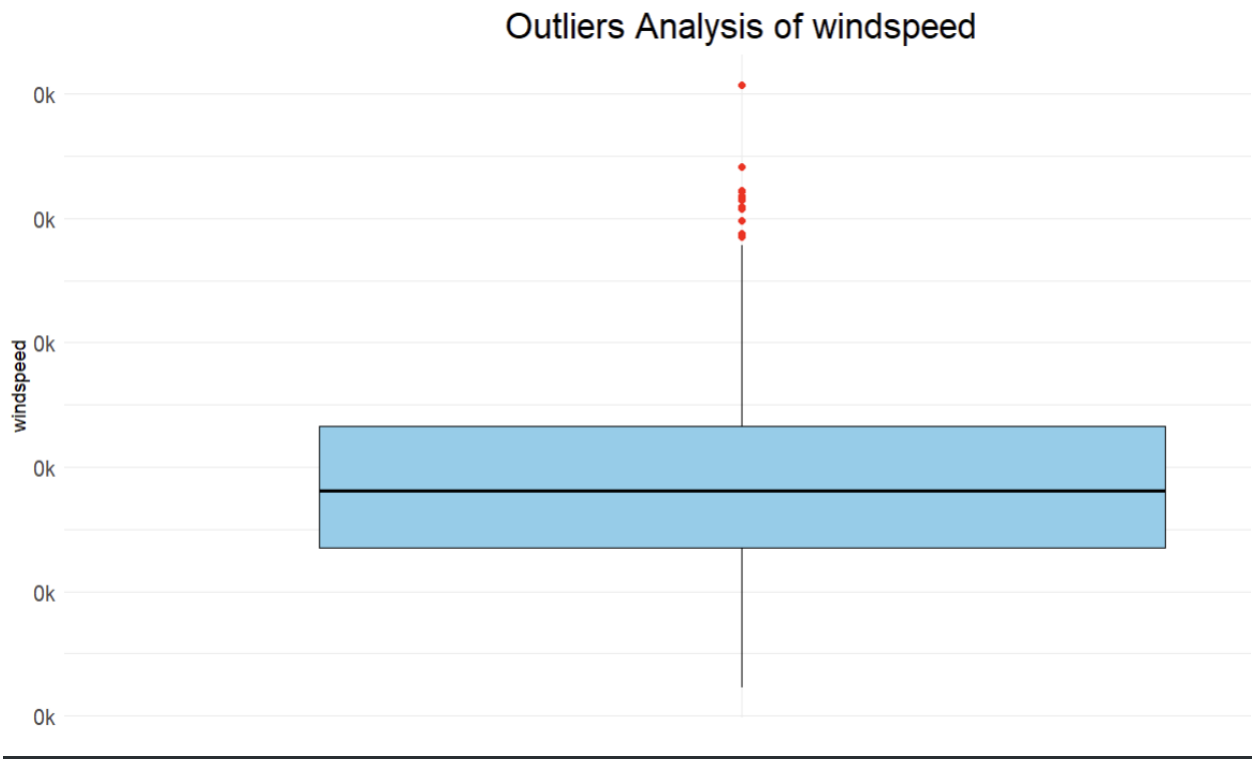
- **Plot boxplot for outliers analysis**

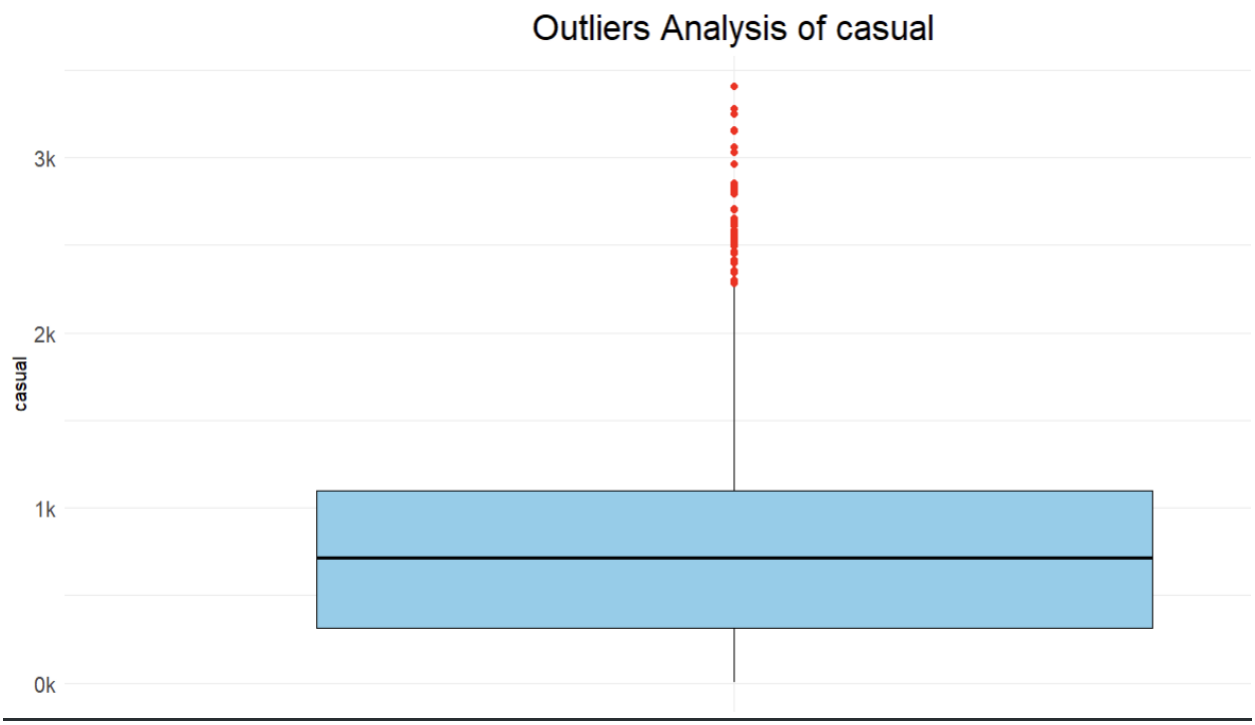
```
numeric_variables <- c("temp", "atemp", "hum", "windspeed", "casual", "registered", "cnt")
boxplots <- list()


for (var in numeric_variables) {
  p <- ggplot(Bike_rental_data1, aes(x = "", y = !!sym(var))) +
    geom_boxplot(fill = "skyblue", color = "black", outlier.color = "red") +
    labs(
      title = paste("Outliers Analysis of", var),
      x = "",
      y = var
    ) +
  scale_y_continuous(labels = scales::number_format(scale = 1e-3, accuracy = 1, suffix = "k")) +
    theme_minimal() +
    theme(
      plot.title = element_text(size = 20, hjust = 0.5), # Adjust title size and center it
      axis.text.x = element_blank(), # Remove X-axis labels
      axis.ticks.x = element_blank(), # Remove X-axis ticks
      axis.text.y = element_text(size = 12) # Adjust Y-axis label font size
    )
  boxplots[[var]] <- p
}

boxplots # Print the boxplots
```

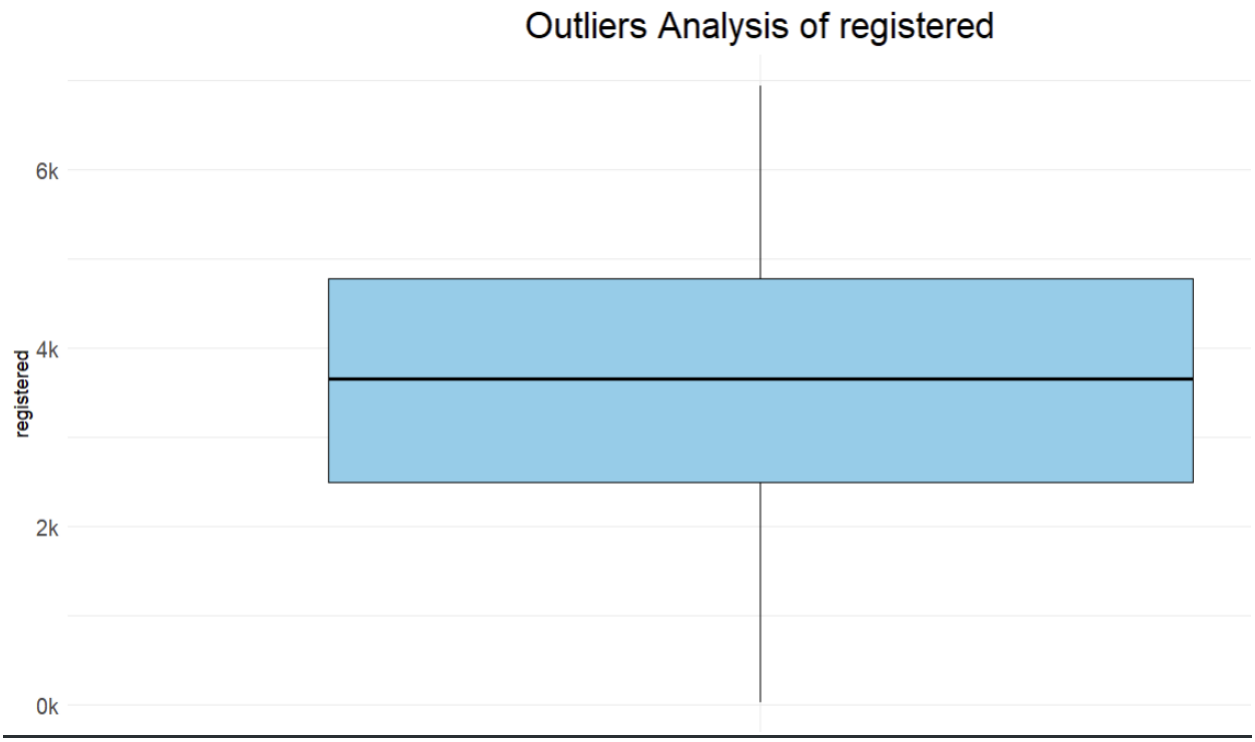

Outliers Analysis of temp
0k
0k
0k
temp

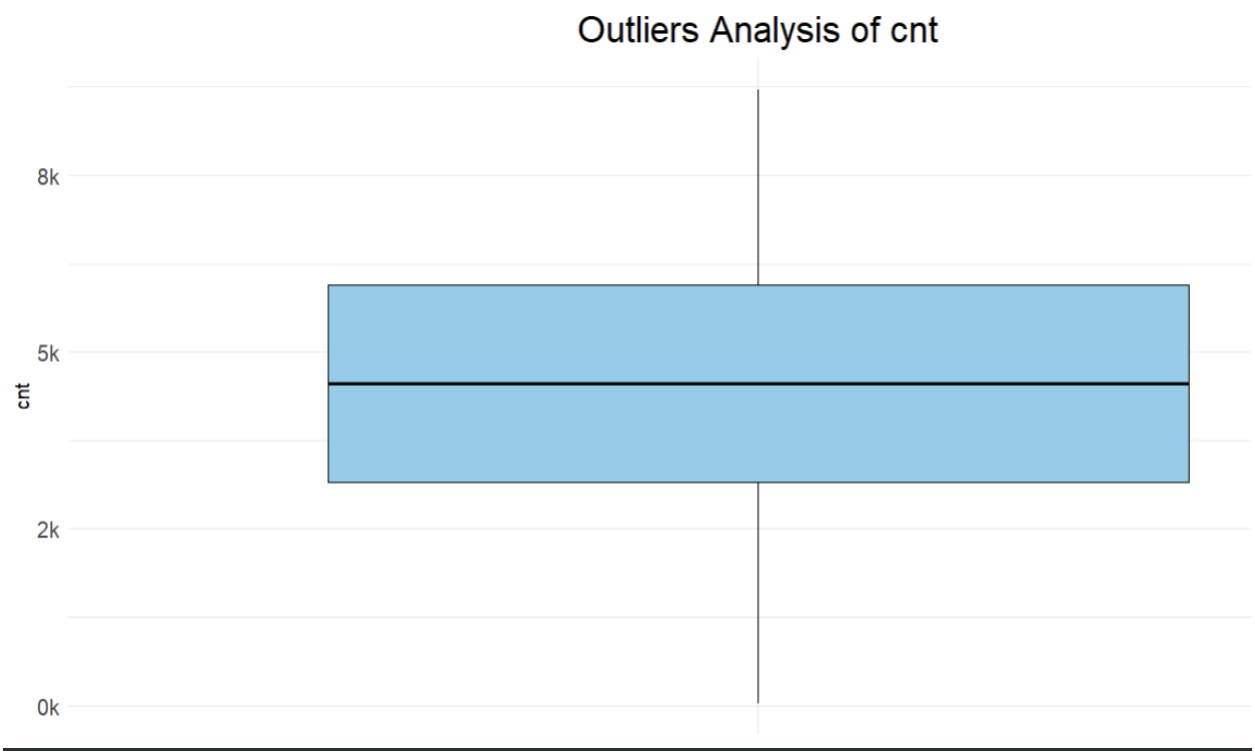

Outliers Analysis of atemp
0k
0k
0k
0k
atemp


Outliers Analysis of hum
0k
0k
hum
0k
0k
0k


Outliers Analysis of windspeed
0k
0k
windspeed
0k
0k
0k
0k


Outliers Analysis of casual
3k
2k
1k
0k
casual


Outliers Analysis of registered
6k
4k
2k
0k
registered



3. Split the dataset into train and test dataset

```
set.seed(123)

trainIndex <- createDataPartition(Bike_rental_data1$cnt, p = 0.7, list = FALSE)
training_data <- Bike_rental_data1[trainIndex, ]

training_data
```

```
> training_data
# A tibble: 515 × 16
   instant dteday     season    yr mnth  holiday weekday workingday weathersit
     <int> <date>     <fct>  <dbl> <fct> <fct>   <fct>   <fct>      <fct>
 1       4 2011-01-04 1          1 1     0       2       1          1
 2       5 2011-01-05 1          1 1     0       3       1          1
 3       6 2011-01-06 1          1 1     0       4       1          1
 4       7 2011-01-07 1          1 1     0       5       1          2
 5       8 2011-01-08 1          1 1     0       6       0          2
 6      11 2011-01-11 1          1 1     0       2       1          2
 7      12 2011-01-12 1          1 1     0       3       1          1
 8      13 2011-01-13 1          1 1     0       4       1          1
 9      14 2011-01-14 1          1 1     0       5       1          1
10      16 2011-01-16 1          1 1     0       0       0          1
```

test_data <- Bike_rental_data1[-trainIndex,]

test_data

```
> test_data
# A tibble: 216 × 16
   instant dteday     season    yr mnth  holiday weekday workingday weathersit
     <int> <date>     <fct>  <dbl> <fct> <fct>   <fct>   <fct>      <fct>
 1       1 2011-01-01 1          1 1     0       6       0          2
 2       2 2011-01-02 1          1 1     0       0       0          2
 3       3 2011-01-03 1          1 1     0       1       1          1
 4       9 2011-01-09 1          1 1     0       0       0          1
 5      10 2011-01-10 1          1 1     0       1       1          1
 6      15 2011-01-15 1          1 1     0       6       0          2
 7      18 2011-01-18 1          1 1     0       2       1          2
 8      20 2011-01-20 1          1 1     0       4       1          2
 9      28 2011-01-28 1          1 1     0       5       1          2
10      29 2011-01-29 1          1 1     0       6       0          1
```

4. Create a model using the random forest algorithm

model <- randomForest(cnt ~ season + yr + mnth + holiday + weekday + workingday

+ weathersit + temp + atemp + hum + windspeed + casual

+ registered,

data = training_data)

predictions <- predict(model, newdata = test_data)

model

```
Call:
 randomForest(formula = cnt ~ season + yr + mnth + holiday + weekday +      workingd
ay + weathersit + temp + atemp + hum + windspeed +      casual + registered, data =
training_data)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 80900.07
                    % Var explained: 97.84
> |
```

5. Predict the performance of the model on the test dataset

test_predictions <- predict(model, newdata = test_data)

rmse <- sqrt(mean((test_data$cnt - test_predictions)^2))

rmse

cat("Root Mean Squared Error (RMSE):", rmse, "\n")

```
> test_predictions <- predict(model, newdata = test_data)
>
> rmse <- sqrt(mean((test_data$cnt - test_predictions)^2))
> rmse
[1] 302.2371
> cat("Root Mean Squared Error (RMSE):", rmse, "\n")
Root Mean Squared Error (RMSE): 302.2371
> |
```

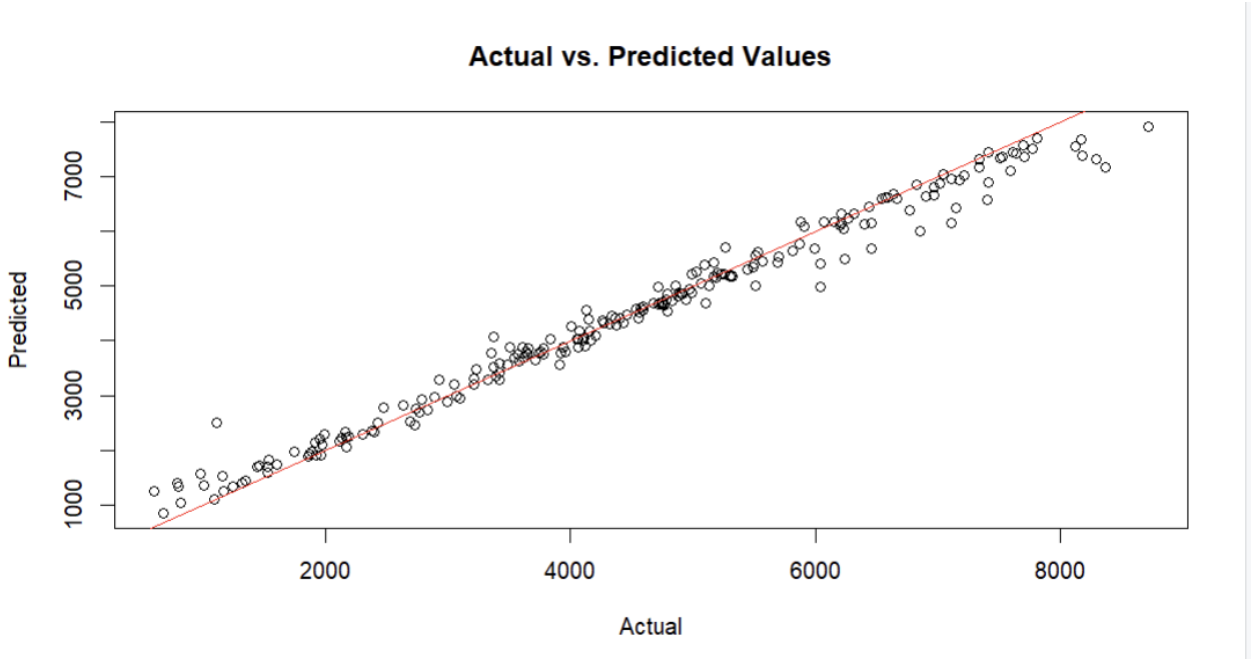
r_squared <- 1 - (sum((test_data$cnt - test_predictions)^2) / sum((test_data$cnt - mean(test_data$cnt))^2))

cat("R-squared (R2):", r_squared, "\n")

```
> r_squared <- 1 - (sum((test_data$cnt - test_predictions)^2) / sum((test_data$cnt
mean(test_data$cnt))^2))
> cat("R-squared (R2):", r_squared, "\n")
R-squared (R2): 0.9756154
>
```

plot(test_data$cnt, test_predictions, xlab = "Actual", ylab = "Predicted", main = "Actual vs. Predicted Values")

abline(0, 1, col = "red") # Add a diagonal line for reference



Project Completed By

Mayur Nivadekar