R studio Assignment
Bike Rental Prediction

Submitted by: Mohit Rohilla

Description

Problem Statement:

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.

Data Set: day.csv

Data Description

Variable	**Description**
instant	Record index
dteday	Date
season	Season (1: springer, 2: summer, 3: fall, 4: winter)
yr	Year (0: 2011, 1:2012)
mnth	Month (1 to 12)
holiday	Weather day is a holiday or not
weekday	Day of the week
workingday	Working day (1: neither weekend nor holiday, 0: other days)
weathersit	1: Clear, few clouds, partly cloudy, partly cloudy 2: Mist + cloudy, mist + broken clouds, mist + few clouds, mist 3: Light snow, light rain + thunderstorm + scattered clouds, light rain + scattered clouds 4: Heavy rain + ice pallets
temp	Normalized temperature in Celsius; The values are divided into 41 (max)
atemp	Normalized feeling temperature in Celsius; The values are divided into 50 (max)
hum	Normalized humidity; The values are divided into 100 (max)
windspeed	Normalized wind speed; The values are divided into 67 (max)
casual	Count of casual users
registered	Count of registered users
cnt	Count of total rental bikes including both casual and registered

Steps to Perform:

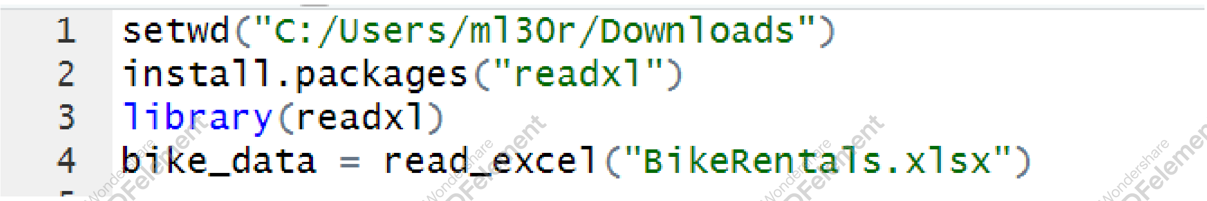
1. Exploratory data analysis
- Load dataset and libraries
- Perform data type conversion of the attributes
- Carry out the missing value analysis

2. Attributes distributions and trends
- Plot monthly distribution of the total number of bikes rented
- Plot yearly distribution of the total number of bikes rented
- Plot boxplot for outliers' analysis

3. Split the dataset into train and test dataset
4. Create a model using the random forest algorithm
5. Predict the performance of the model on the test datasetq

Step1: Load data into R studio

```
setwd("C:/Users/ml30r/Downloads")
install.packages("readxl")
library(readxl)
bike_data = read_excel("BikeRentals.xlsx")
```

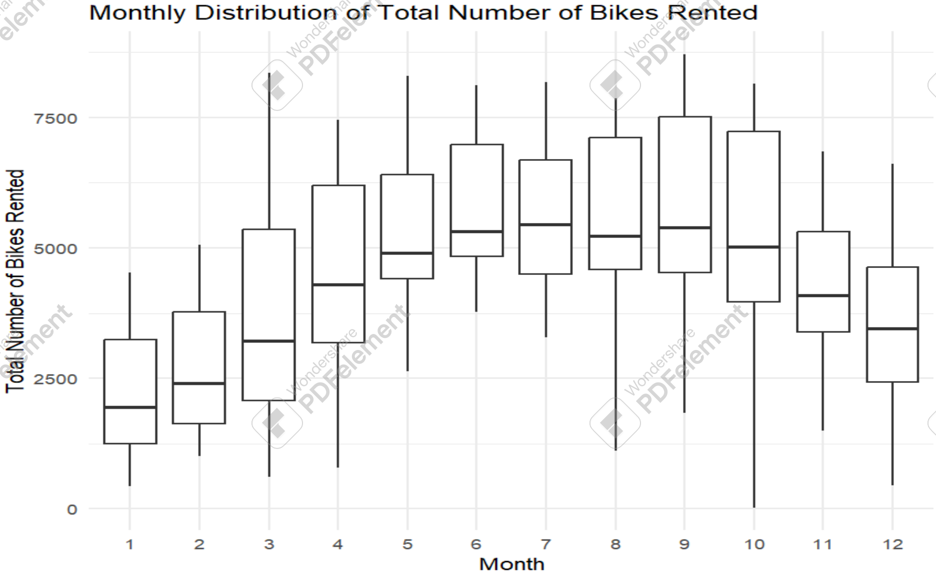


```
setwd("C:/Users/ml30r/Downloads")
install.packages("readxl")
library(readxl)
bike_data = read_excel("BikeRentals.xlsx")
```

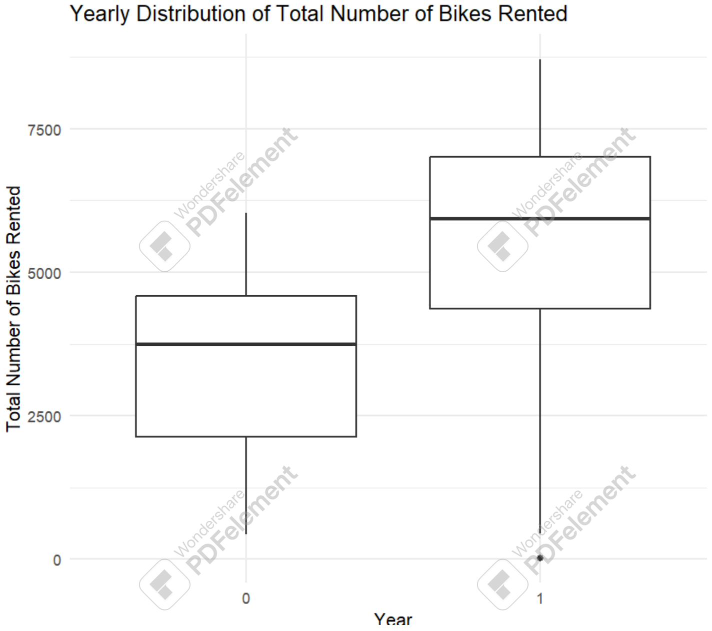
Task 1: Exploratory data analysis

```
# Convert columns to appropriate types

bike_data$season = as.factor(bike_data$season)
bike_data$yr = as.factor(bike_data$yr)
bike_data$mnth = as.factor(bike_data$mnth)
bike_data$holiday = as.factor(bike_data$holiday)
bike_data$weekday = as.factor(bike_data$weekday)
bike_data$workingday = as.factor(bike_data$workingday)
bike_data$weathersit = as.factor(bike_data$weathersit)

# Convert 'dteday' to Date type
bike_data$dteday <- as.Date(bike_data$dteday)

# View data structure to confirm types
str(bike_data)
```

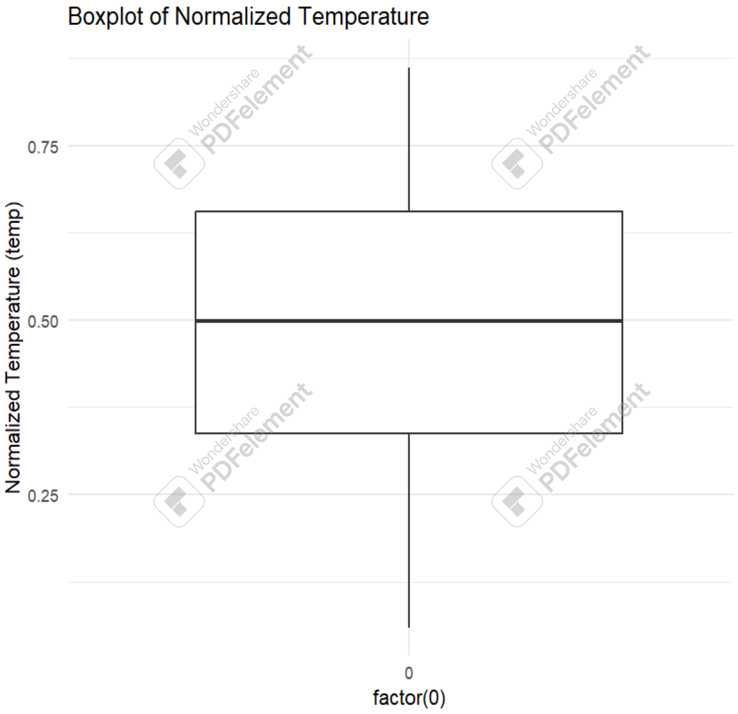
```r
install.packages("ggplot2")
library(ggplot2)

# Plot monthly distribution of the total number of bikes rented
ggplot(bike_data, aes(x = mnth, y = cnt)) +
  geom_boxplot() +
  labs(title = "Monthly Distribution of Total Number of Bikes Rented",
       x = "Month", y = "Total Number of Bikes Rented") +
  theme_minimal()
```



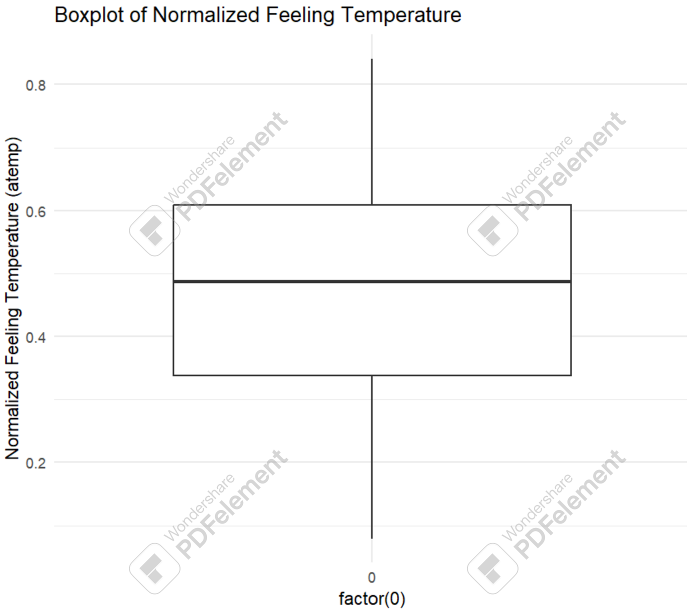
```r
# Plot yearly distribution of the total number of bikes rented
ggplot(bike_data, aes(x = yr, y = cnt)) +
  geom_boxplot() +
  labs(title = "Yearly Distribution of Total Number of Bikes Rented",
       x = "Year", y = "Total Number of Bikes Rented") +
  theme_minimal()
```



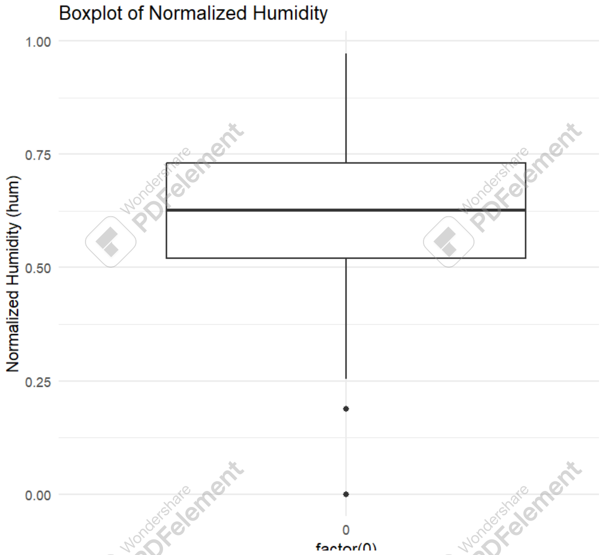
```
# Boxplot for outliers analysis (for temp, atemp, hum, windspeed)
ggplot(bike_data) +
  geom_boxplot(aes(x = factor(0), y = temp)) +
  labs(title = "Boxplot of Normalized Temperature", y = "Normalized Temperature
(temp)") +
  theme_minimal()
```



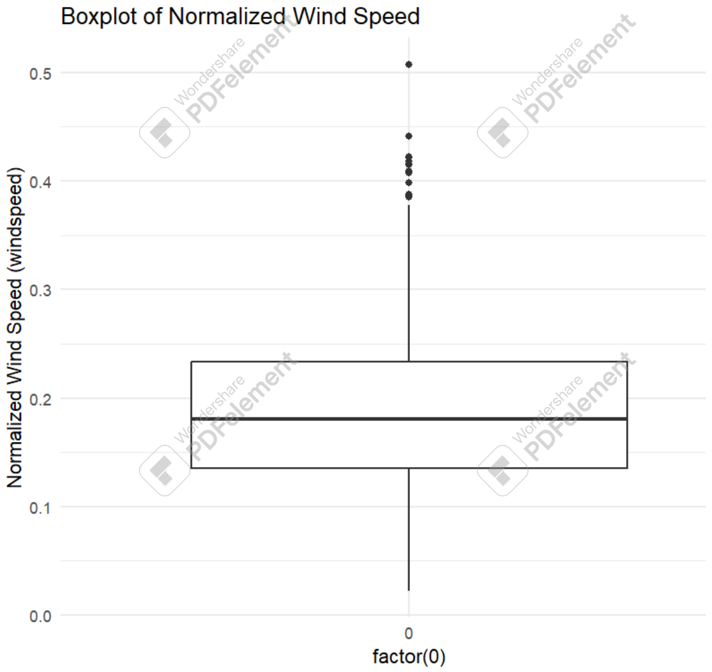
```
ggplot(BikeRentals) +
  geom_boxplot(aes(x = factor(0), y = atemp)) +
  labs(title = "Boxplot of Normalized Feeling Temperature", y = "Normalized Feeling
Temperature (atemp)") +
  theme_minimal()
```



```
ggplot(bike_data) +
  geom_boxplot(aes(x = factor(0), y = hum)) +
  labs(title = "Boxplot of Normalized Humidity", y = "Normalized Humidity (hum)") +
  theme_minimal()
```

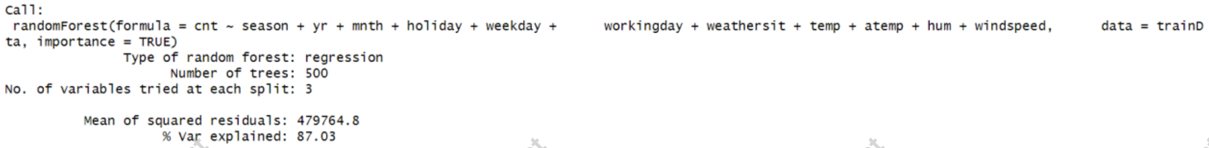


```
ggplot(bike_data) +
  geom_boxplot(aes(x = factor(0), y = windspeed)) +
  labs(title = "Boxplot of Normalized Wind Speed", y = "Normalized Wind Speed
(windspeed)") +
  theme_minimal()
```



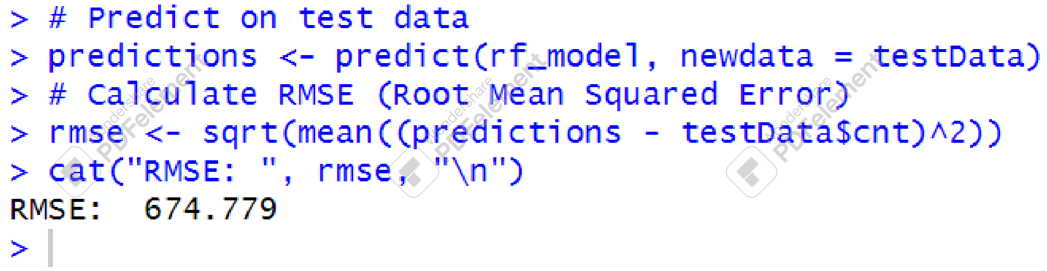
Task 3: Split the dataset into train and test dataset

```
set.seed(123)
# For reproducibility
install.packages("caret")
library(caret)
install.packages("lattice")
library(lattice)
trainIndex = createDataPartition(bike_data$cnt, p = 0.8, list = FALSE)
trainData = bike_data[trainIndex, ]
testData = bike_data[-trainIndex, ]

# Train the Random Forest model
install.packages("randomForest")
library(randomForest)
rf_model <- randomForest(cnt ~ season + yr + mnth + holiday + weekday +
workingday +
                weathersit + temp + atemp + hum + windspeed,
               data = trainData,
               importance = TRUE)

# Print model summary
print(rf_model)
```



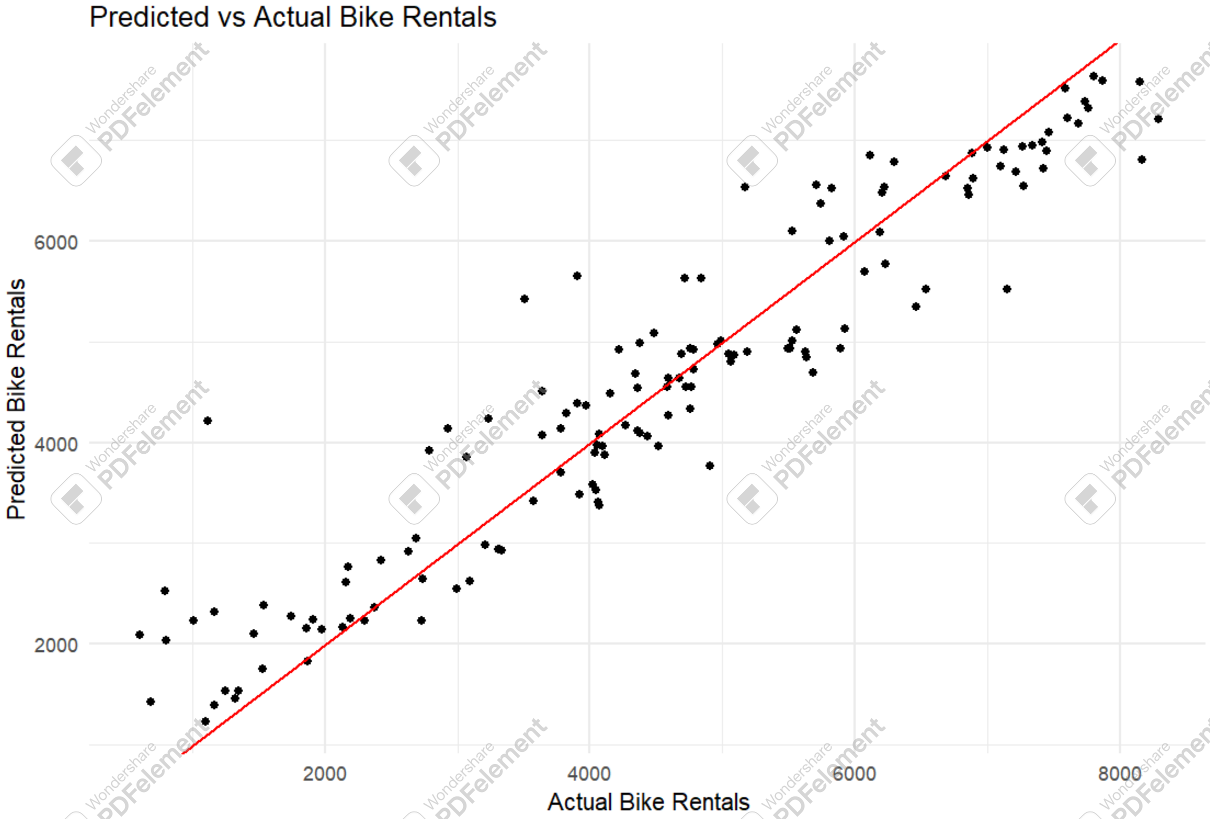
```
Call:
 randomForest(formula = cnt ~ season + yr + mnth + holiday + weekday +      workingday + weathersit + temp + atemp + hum + windspeed,      data = trainD
ta, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 479764.8
                    % Var explained: 87.03
```

```
# Predict on test data
predictions <- predict(rf_model, newdata = testData)

# Calculate RMSE (Root Mean Squared Error)
rmse <- sqrt(mean((predictions - testData$cnt)^2))
cat("RMSE: ", rmse, "\n")
```



```
> # Predict on test data
> predictions <- predict(rf_model, newdata = testData)
> # Calculate RMSE (Root Mean Squared Error)
> rmse <- sqrt(mean((predictions - testData$cnt)^2))
> cat("RMSE: ", rmse, "\n")
RMSE:  674.779
> 
```

```
# Plot predicted vs actual values
ggplot(testData, aes(x = cnt, y = predictions)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  labs(title = "Predicted vs Actual Bike Rentals",
       x = "Actual Bike Rentals", y = "Predicted Bike Rentals") +
  theme_minimal()
```


Predicted vs Actual Bike Rentals
Predicted Bike Rentals
Actual Bike Rentals