bike_rental

2024-02-18

1. Exploratory data analysis:

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

Load dataset

```
library(readxl)
bike_data<- read_excel('day.xlsx')

View(bike_data)
```

Load necessary libraries

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library(caTools)
```

Data type conversion

```
bike_data$dteday <- as.Date(bike_data$dteday)
bike_data$season <- as.factor(bike_data$season)
```

Missing value analysis

```
print(colSums(is.na(bike_data)))
```

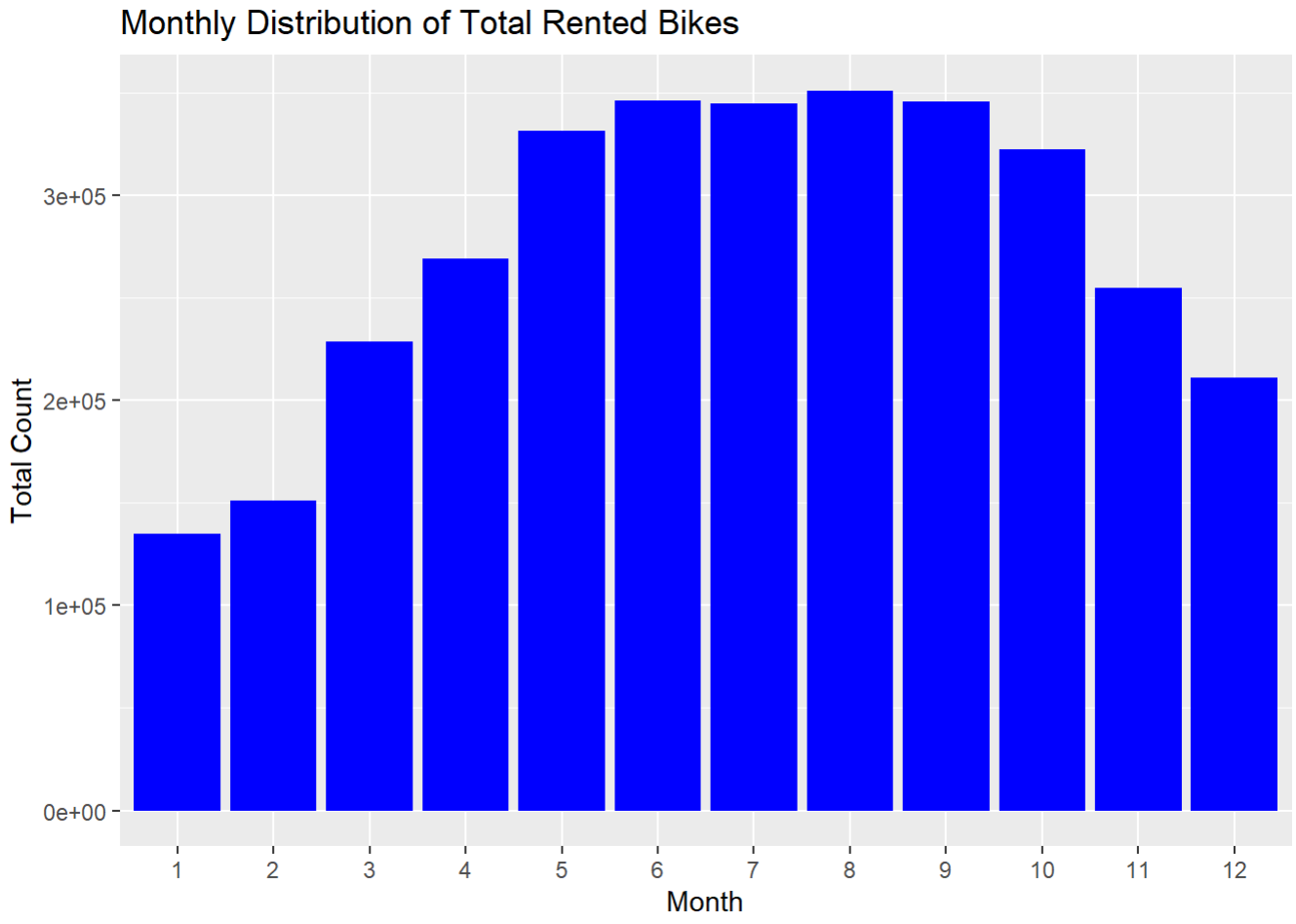
```
##    instant     dteday     season         yr       mnth    holiday    weekday
##          0          0          0          0          0          0          0
## workingday weathersit       temp      atemp        hum  windspeed     casual
##          0          0          0          0          0          0          0
## registered        cnt
##          0          0
```

2. Attributes distribution 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

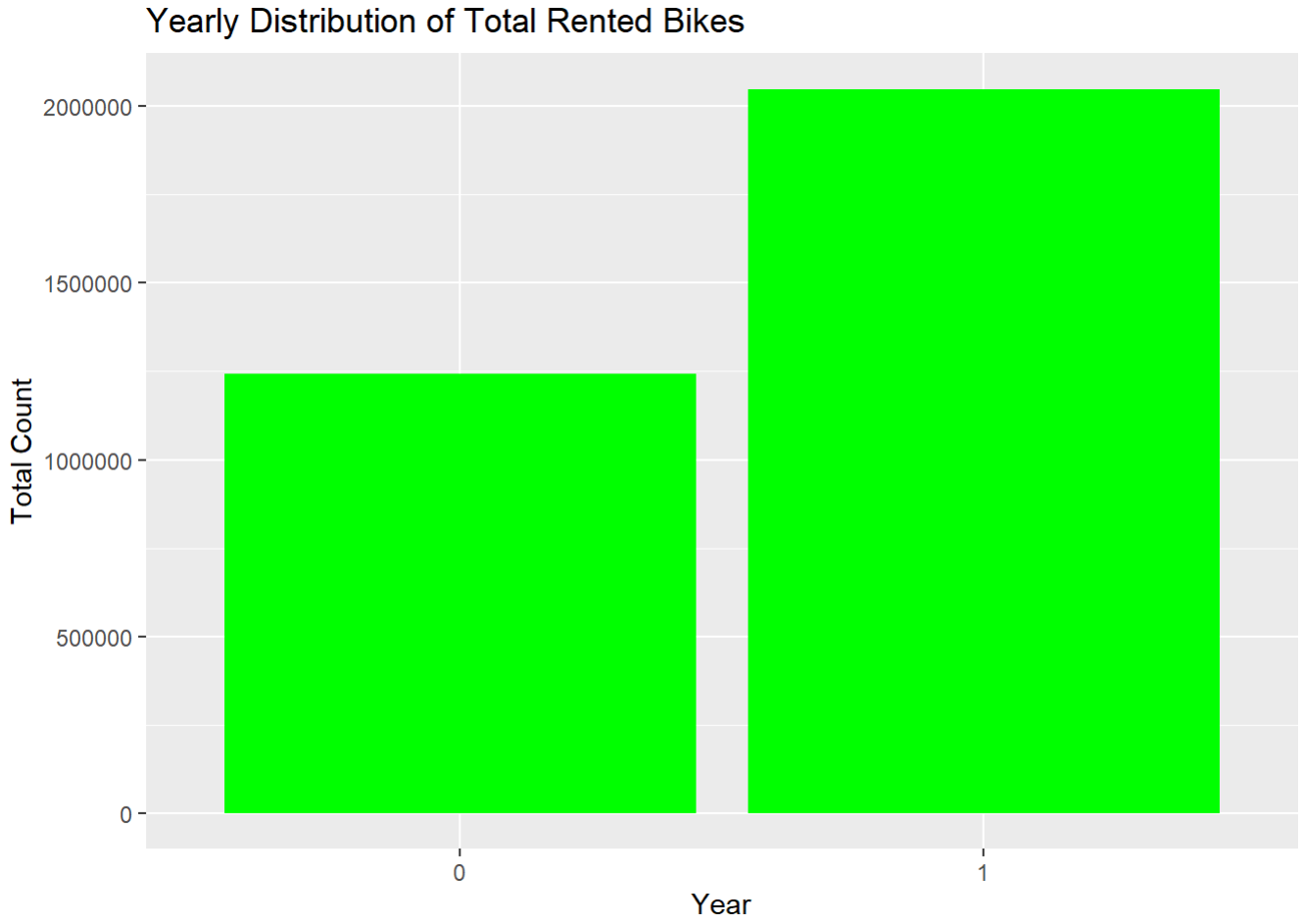
Monthly distribution of total rented bikes

```
ggplot(bike_data, aes(x = factor(mnth), y = cnt)) +
  geom_bar(stat = 'identity', fill = 'blue') +
  labs(title = 'Monthly Distribution of Total Rented Bikes',
       x = 'Month',
       y = 'Total Count')
```



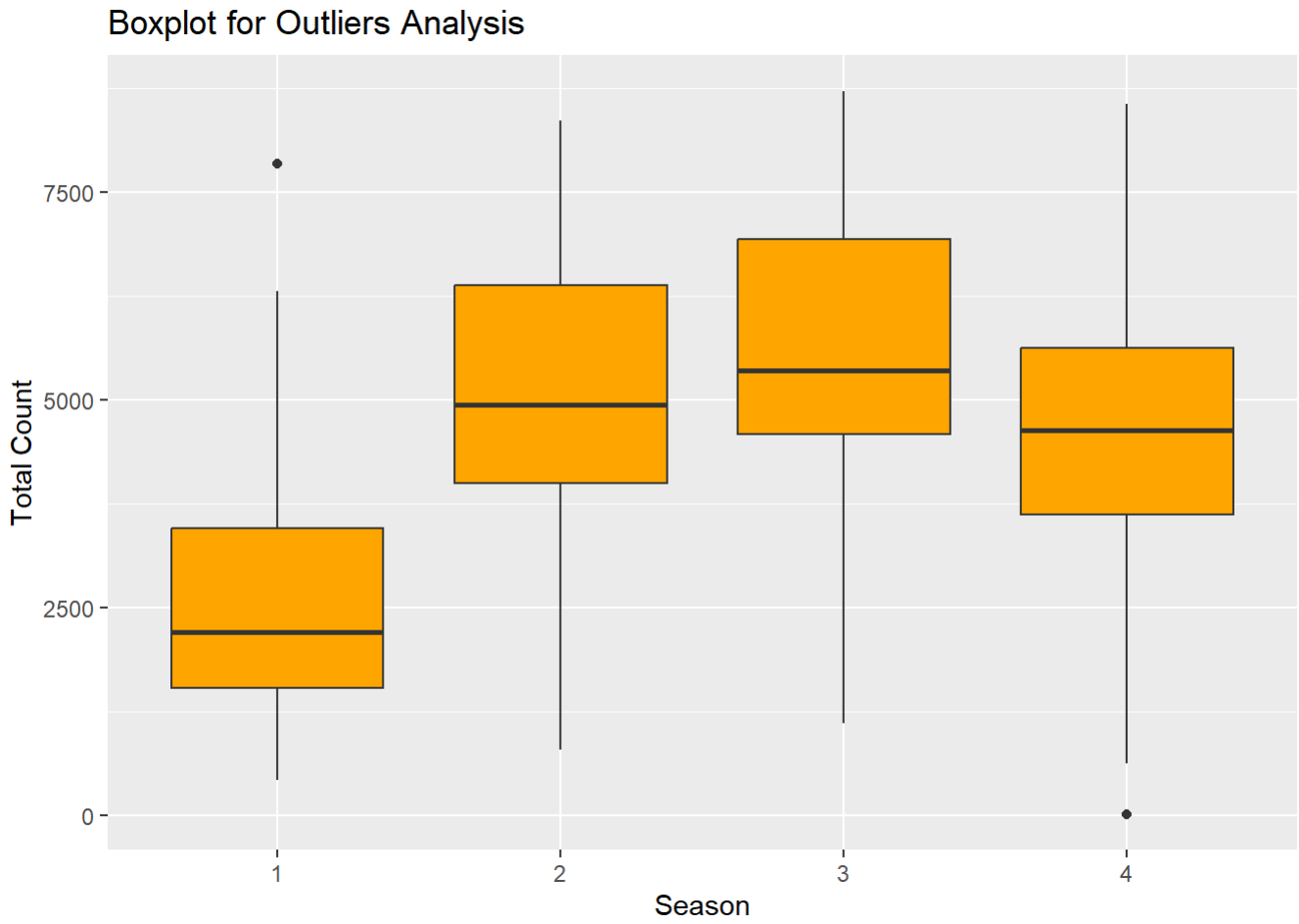
Yearly distribution of total rented bikes

```
ggplot(bike_data, aes(x = factor(yr), y = cnt)) +
  geom_bar(stat = 'identity', fill = 'green') +
  labs(title = 'Yearly Distribution of Total Rented Bikes',
       x = 'Year',
       y = 'Total Count')
```



Boxplot for outliers analysis

```
ggplot(bike_data, aes(x = season, y = cnt)) +
  geom_boxplot(fill = 'orange') +
  labs(title = 'Boxplot for Outliers Analysis',
       x = 'Season',
       y = 'Total Count')
```



3. Split the dataset into train and test dataset

Drop irrelevant columns for the prediction model

```
X <- bike_data[, !(names(bike_data) %in% c('instant', 'dteday', 'casual', 'registered', 'cn
t'))]
y <- bike_data$cnt
```

Split the dataset into training and testing sets

```
set.seed(42)
split <- sample.split(y, SplitRatio = 0.8)
X_train <- subset(X, split == TRUE)
X_test <- subset(X, split == FALSE)
y_train <- y[split == TRUE]
y_test <- y[split == FALSE]
```

4. Create a model using the random forest algorithm

Initialize the Random Forest Regressor model

```
rf_model <- randomForest(y_train ~ ., data = X_train, ntree = 100, seed = 42)
```

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

Make predictions on the test set

```
y_pred <- predict(rf_model, newdata = X_test)
```

Evaluate the model performance

```
mse <- mean((y_test - y_pred)^2)
cat('Mean Squared Error on Test Data:', mse, '\n')
```

```
## Mean Squared Error on Test Data: 594485.5
```

visualize the predictions vs actual values if needed

```
plot(y_test, type = 'l', col = 'blue', lty = 1, ylim = c(0, max(y_test, y_pred)), ylab = 'Tot
al Count', xlab = 'Data Points')
lines(y_pred, col = 'red', lty = 2)
legend('topright', legend = c('Actual', 'Predicted'), col = c('blue', 'red'), lty = 1:2)
```