## Bike Rental Prediction

library("tidyverse") #need to call the library before you use the package

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library("rpivotTable")  
library("knitr")  
library("psych")

## Warning: package 'psych' was built under R version 4.3.2

##   
## Attaching package: 'psych'  
##   
## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library("RColorBrewer") # Can be installed if using the color palettes from the package  
library("ggplot2")  
library("scales")

##   
## Attaching package: 'scales'  
##   
## The following objects are masked from 'package:psych':  
##   
## alpha, rescale  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

library("randomForest")

## Warning: package 'randomForest' was built under R version 4.3.2

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:psych':  
##   
## outlier  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library("caret")

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library("stats")

## Dataset Description

* Dataset required: Bike Rental.xlsx

Sales Transactions.xlsx contains the entire process from membership to rental and return has been automated in bike-sharing systems. Each of the column is defined as follows:

* instant : Record index
* dteday: Date
* season: Season (1: spring, 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
* Time Of Day: Working day (1: neither weekend nor holiday, 0: other days)
* weathersit: Weather situation (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

#import excel file into RStudio  
library("readxl")  
setwd("C:/Users/Admin/Downloads")  
#import xlsx file into RStudio  
df <- read\_excel("Bike Rental.xlsx")

## Exploratory data analysis

* Display the structure of the dataset

str(df)

## 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 ...

* Data type conversion

df$dteday <- as.Date(df$dteday)  
# Display the summary statistics  
summary(df)

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

# Check for missing values   
missing\_values <- colSums(is.na(df))  
  
# Display the columns with missing values (if existed)  
print("Columns with Missing Values:")

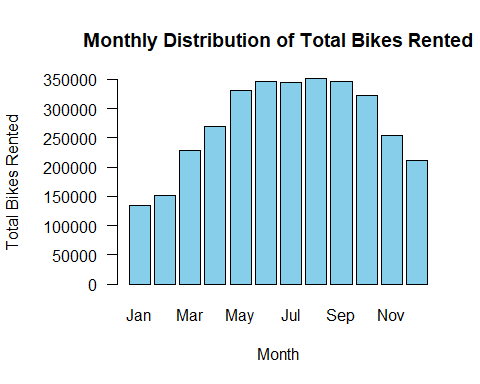
## [1] "Columns with Missing Values:"

print(missing\_values[missing\_values > 0]) #There is no missing value

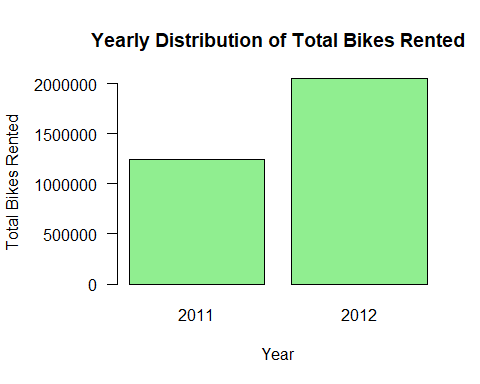
## named numeric(0)

## Data vizualization

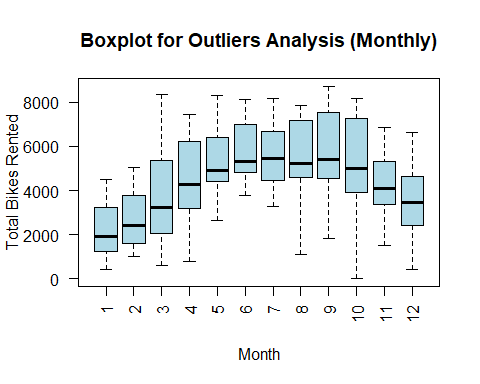
# Convert 'dteday' to Date type if it's not already  
df$dteday <- as.Date(df$dteday, format="%m/%d/%Y")  
  
# Extract month and year from 'dteday'  
df$month <- as.numeric(format(df$dteday, "%m"))  
df$year <- ifelse(df$yr == 0, 2011, 2012) # Convert 'yr' to the actual year  
  
# Aggregate data by month and year  
monthly\_totals <- aggregate(cnt ~ month, df, sum)  
yearly\_totals <- aggregate(cnt ~ year, df, sum)  
  
# Sort the data by month and year to ensure proper ordering in plots  
monthly\_totals <- monthly\_totals[order(monthly\_totals$month), ]  
yearly\_totals <- yearly\_totals[order(yearly\_totals$year), ]  
  
# Set up the plot with adjusted margins  
par(mar=c(5, 6, 4, 2) + 0.1) # Increase the left margin  
  
# Monthly distribution of the total number of bikes rented  
barplot(monthly\_totals$cnt, names.arg=month.abb[monthly\_totals$month],  
 main="Monthly Distribution of Total Bikes Rented",  
 xlab="Month", ylab="", col="skyblue", las=1)  
  
# Add the y-axis label manually to avoid overlap  
mtext("Total Bikes Rented", side=2, line=5)



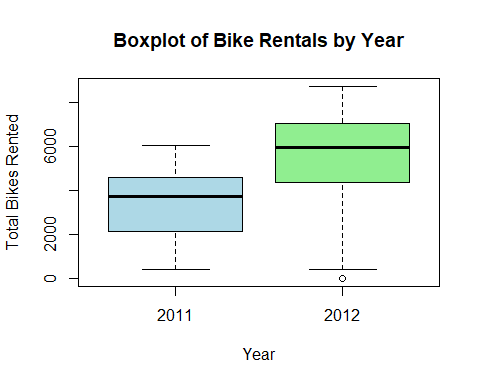
# Reset to default margins after plotting  
par(mar=c(5, 4, 4, 2) + 0.1)  
  
  
# Set up the plot with adjusted margins  
par(mar=c(5, 6, 4, 2) + 0.1) # Increase the left margin to prevent y-label overlap  
  
# Yearly distribution of the total number of bikes rented  
barplot(yearly\_totals$cnt, names.arg=yearly\_totals$year,  
 main="Yearly Distribution of Total Bikes Rented",  
 xlab="Year", ylab="", col="lightgreen", las=1)  
# Add the y-axis label manually to avoid overlap  
mtext("Total Bikes Rented", side=2, line=5)



# Reset to default margins after plotting  
par(mar=c(5, 4, 4, 2) + 0.1)  
  
# Boxplot for outlier analysis, split by 'month'  
boxplot(df$cnt ~ factor(df$month),  
 main="Boxplot for Outliers Analysis (Monthly)",  
 xlab="Month", ylab="Total Bikes Rented", col="lightblue",  
 las=2) # las=2 makes the axis labels perpendicular to the axis



# Create a boxplot for the 'cnt' variable, split by 'year'  
boxplot(cnt ~ year, data = df,   
 main = "Boxplot of Bike Rentals by Year",   
 xlab = "Year",   
 ylab = "Total Bikes Rented",  
 col = c("lightblue", "lightgreen"),  
 outline = TRUE) # 'outline = TRUE' will plot the outliers as individual points



### Monthly Distribution of Total Bikes Rented

* The graph shows a pattern that suggests seasonality in bike rentals. There is an increase in the number of bikes rented starting from spring months (March, April) peaking in the summer months (June, July, August, September), and then a decline as it moves into the fall and winter months (October, November, December). This pattern typically correlates with weather conditions, where warmer months are more favorable for biking activities.
* Peak Rental Period: The highest number of bike rentals occurs during the summer months, with the peak appearing to be in June or July. This could be due to several factors such as favorable weather, increased tourism, or more daylight hours.
* Lowest Rental Period: The lowest rentals are in the winter months, with January having the fewest rentals. Cold weather, snow, and shorter days could be contributing factors to this trend.
* Trend Over the Year: There’s a clear upward trend in rentals from January to the summer months, and then a downward trend as the year progresses towards winter.

### Yearly Distribution of Total Bikes Rented

* There is a significant increase in the total number of bikes rented from 2011 to 2012. This could indicate a growing popularity or expansion of the bike rental service.

### Boxplot for monthly outliers

* Every month appears to have at least one outlier, with all of them being on the upper side, which could indicate days with unusually high bike rentals. July uniquely shows outliers on both the lower and upper sides, suggesting both unusually low and high rental numbers in that month.

### Boxplot of Bike Rentals by Year

* 2011: The boxplot for 2011 shows the median bike rentals is below 4000. The spread of the data, as shown by the interquartile range (the box), is narrower compared to 2012, indicating less variability in daily rental counts. The “whiskers” extend to include most of the data, and there don’t appear to be any outliers.
* 2012: For 2012, the median is higher, above 4000, indicating an overall increase in the median number of bike rentals. The interquartile range is wider, suggesting greater variability in bike rentals in 2012 compared to 2011. There is at least one outlier indicated by the dot below the lower “whisker”, which represents days with unusually low bike rentals for that year.
* 2012, There is one visible outlier below the lower whisker of the boxplot for that year. This indicates at least one day (or more, if data points overlap) where the total bike rentals were significantly lower than usual.
* There are no outliers for 2011 within the range defined by the whiskers, suggesting that all the bike rental counts for that year fell within what’s considered a typical range, without any unusually high or low days.

##Descriptive Statistics

# Calculate summary statistics for the bike rental counts  
summary\_stats <- summary(df$cnt)  
mean\_cnt <- mean(df$cnt)  
median\_cnt <- median(df$cnt)  
sd\_cnt <- sd(df$cnt)  
  
# Print summary statistics  
print(summary\_stats)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 22 3152 4548 4504 5956 8714

print(paste("Mean:", mean\_cnt))

## [1] "Mean: 4504.3488372093"

print(paste("Median:", median\_cnt))

## [1] "Median: 4548"

print(paste("Standard Deviation:", sd\_cnt))

## [1] "Standard Deviation: 1937.21145161877"

# Calculate summary statistics for weather conditions  
summary\_temp <- summary(df$temp)  
summary\_hum <- summary(df$hum)  
summary\_windspeed <- summary(df$windspeed)  
  
# Print weather conditions summary statistics  
print(summary\_temp)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.05913 0.33708 0.49833 0.49538 0.65542 0.86167

print(summary\_hum)

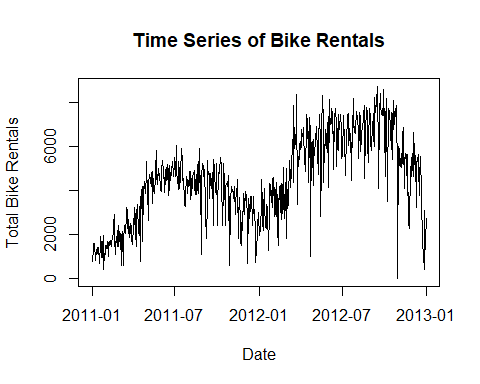
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.5200 0.6267 0.6279 0.7302 0.9725

print(summary\_windspeed)

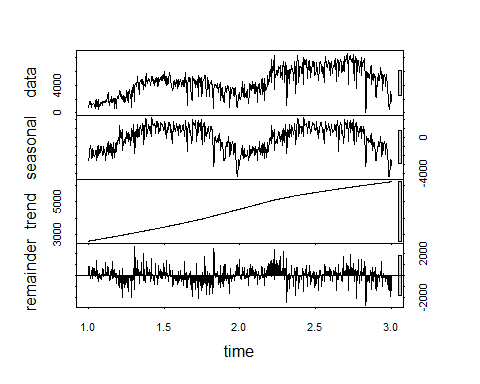
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.02239 0.13495 0.18097 0.19049 0.23321 0.50746

## Time Series Analysis

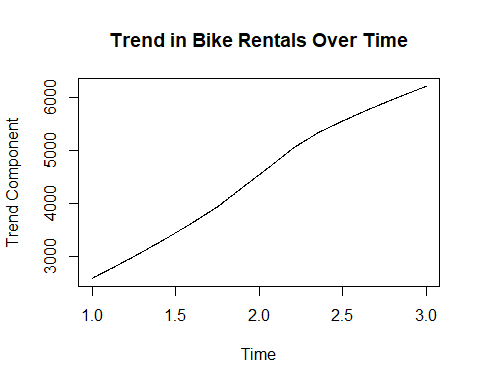
# Visualizing the Time Series  
df$dteday <- as.Date(df$dteday)  
plot(df$dteday, df$cnt, type = 'l', xlab = 'Date', ylab = 'Total Bike Rentals', main = 'Time Series of Bike Rentals')



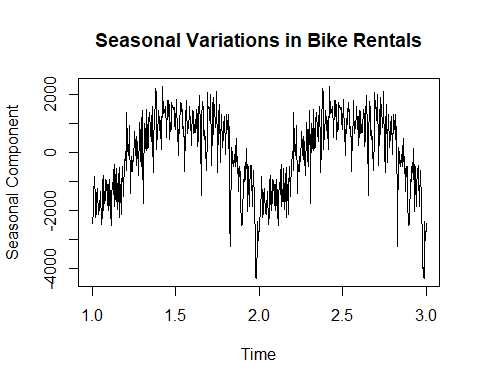
# Decomposition  
ts\_data <- ts(df$cnt, frequency = 365) # Adjust frequency based on your data (365 for daily data with yearly seasonality)  
decomposed <- stl(ts\_data, s.window = "periodic")  
plot(decomposed)



#Trend Analysis  
trend <- decomposed$time.series[,2]  
plot(trend, type = 'l', xlab = 'Time', ylab = 'Trend Component', main = 'Trend in Bike Rentals Over Time')



#Seasonality Analysis  
seasonal <- decomposed$time.series[,1]  
plot(seasonal, type = 'l', xlab = 'Time', ylab = 'Seasonal Component', main = 'Seasonal Variations in Bike Rentals')



* Seasonality: The plot shows a clear seasonal pattern in bike rentals, with peaks and troughs occurring regularly throughout the time series. This suggests that bike rentals are likely influenced by seasonal factors, which could include weather changes, tourism seasons, or public holidays.
* Trend: There seems to be an overall increasing trend in bike rentals over time, particularly noticeable when comparing the peaks from early 2011 to those in 2012.
* Cyclic Behavior: Although the main pattern is seasonal, there may also be shorter-term cycles within each year, possibly related to factors like school holidays, local events, or other recurring activities not directly related to the weather.
* Variance: The variance in bike rentals appears to be relatively constant over time, suggesting that the size of the peaks and troughs does not change much from year to year. This constant variance is a good property for certain time series models, which assume homoscedasticity (constant variance).
* Outliers: There do not appear to be any obvious outliers that would significantly distort the analysis or the fitting of time series models.

## Machine Learning

### Random Forest

# Set seed for reproducibility  
set.seed(42)  
  
# Split the data into training and test sets  
# Typically, 70% to 80% of the data is used for training and the rest for testing  
train\_index <- sample(seq\_len(nrow(df)), size = floor(0.8 \* nrow(df)))  
train\_set <- df[train\_index, ]  
test\_set <- df[-train\_index, ]  
  
# Train the Random Forest model  
rf\_model <- randomForest(cnt ~ ., data = train\_set, ntree = 100) # You can adjust the number of trees  
# Make predictions on the test set  
predictions <- predict(rf\_model, newdata = test\_set)  
  
# Evaluate the model performance  
# Use RMSE (Root Mean Squared Error) and R^2  
rmse\_test <- sqrt(mean((test\_set$cnt - predictions)^2))  
r\_squared\_test <- cor(test\_set$cnt, predictions)^2  
  
  
print(paste("RMSE on test set:", rmse\_test))

## [1] "RMSE on test set: 278.950325530476"

print(paste("R-squared on test set:", r\_squared\_test))

## [1] "R-squared on test set: 0.981978378096821"

* The model achieved a Root Mean Squared Error (RMSE) of approximately 278.95. This indicates that, on average, the model’s predictions are within roughly 279 units of the actual bike rental counts.
* The R-squared value is approximately 0.98198 (or 98.198%), which suggests that the model explains 98.198% of the variance in the bike rental counts

### Linear Regression

# Assuming df is your dataframe, 'cnt' is the target variable, and weather-related variables are present  
  
# Prepare the data  
set.seed(123) # for reproducibility  
index <- createDataPartition(df$cnt, p = 0.8, list = FALSE)  
train\_data <- df[index, ]  
test\_data <- df[-index, ]  
  
# Define the formula for the model, including only weather-related predictors  
# Here's an example where 'temp', 'hum', 'windspeed', and 'weathersit' are the predictors  
model\_formula <- cnt ~ temp + hum + windspeed + weathersit  
  
# Train the model using Linear Regression  
lm\_model <- lm(model\_formula, data = train\_data)  
  
# Summarize the model  
summary(lm\_model)

##   
## Call:  
## lm(formula = model\_formula, data = train\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4028 -1073 -121 1056 3673   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3487.1 377.6 9.234 < 2e-16 \*\*\*  
## temp 6367.6 339.6 18.749 < 2e-16 \*\*\*  
## hum -1400.3 550.5 -2.543 0.01123 \*   
## windspeed -3288.9 821.6 -4.003 7.07e-05 \*\*\*  
## weathersit -449.6 138.8 -3.239 0.00127 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1435 on 582 degrees of freedom  
## Multiple R-squared: 0.4479, Adjusted R-squared: 0.4441   
## F-statistic: 118 on 4 and 582 DF, p-value: < 2.2e-16

# Make predictions on the test set  
predictions <- predict(lm\_model, newdata = test\_data)  
  
# Evaluate the model performance  
# Calculate RMSE  
rmse <- sqrt(mean((test\_data$cnt - predictions)^2))  
  
# Calculate R-squared  
r\_squared <- summary(lm\_model)$r.squared  
  
# Print the performance metrics  
print(paste("RMSE:", rmse))

## [1] "RMSE: 1332.20092101415"

print(paste("R-squared:", r\_squared))

## [1] "R-squared: 0.447896050261612"

* The R-squared value of 0.4479 suggests that approximately 44.79% of the variability in bike rentals is explained by the model
* temp: The coefficient for temp is 6367.6, indicating that for each unit increase in the normalized temperature, the count of bike rentals increases by approximately 6367.6, holding all other variables constant. This significant positive relationship suggests that warmer temperatures are associated with higher bike rentals.
* The equation predicting the total number of bike rentals (cnt) based on weather conditions can be written as: cnt = 3487.1 + 6367.6 × (temp)− 1400.3 × (hum) − 3288.9 × (windspeed) − 449.6 ×(weathersit)