Seoul Bike Sharing Demand Analysis and Prediction

Ahmad Sadeed (asadeed2), Deepa Nemmili Veeravalli (deepan2), Rui Zou (ruizou4)

2022-07-30

Description of the data file

This data file contains count of public bikes rented at each hour in Seoul Bike Sharing System with the corresponding weather data and holidays information. It has 14 variables and 8760 observations. We are interested in using Rented.Bike.Count (a numeric variable) as our response variable and explore how other factors (3 categorical variables and several continuous numeric variables) affect the count of bikes rented at each hour. Among the other 13 variables which we plan to use as potential predictors, we know from intuition that some may have more importance than others, like temperature, humidity, wind speed, visibility, seasons, and holiday, etc.

Background information on the data set

The original data comes from http://data.seoul.go.kr. The holiday information comes from SOUTH KOREA PUBLIC HOLIDAYS. A clean version can be found at UCI Machine Learning Repository.

Attribute Information:

- Date : month/day/year
- Rented Bike count - Count of bikes rented at each hour
- Hour - Hour of the day
- Temperature - Temperature in Celsius
- Humidity - %
- Windspeed - m/s
- Visibility - 10m
- Dew point temperature - Celsius
- Solar radiation - MJ/m2
- Rainfall - mm
- Snowfall - cm
- Seasons - Winter, Spring, Summer, Autumn
- Holiday - Holiday, No holiday
- Functional Day - Functional or Non-functional days of rental bike system

Our Interest

This data set is interesting to us both personally and business-wise. Recently we have seen a rise in the delivery, accessibility, and usage of regular and electric rental bikes. There are clear environmental, health, and economical benefits associated with the usage of bikes as a mode of transportation. We would like to find out what factors lead to an increase in number of bikes rented and what factors have inverse effect on using

rental bikes. Learning about such factors can help a bike rental business manage its inventory and supply without any hindrance. It can also help cities plan accordingly due to an increase of bikers, e.g. opening up more bike lanes during certain days or seasons. Environmentally, we will have a better understanding of the feasibility of turning a city into a "bike city" or looking at alternative options if a city is not friendly to bikers due to harsh weather conditions.

Data in R

The data file can be successfully loaded into R. We have printed out the structure and first few rows of the data file below.

The column names in the `csv` file contains measurement units (like `Wind speed (m/s)`, `Solar Radiation (MJ/m2)`) and characters such as ° and %. We load the data using cleaned up column names.

```
columns = c("Date","Rented","Hour","Temp","Humidity",
            "Wind","Visibility","Dew",
            "Radiation","Rain","Snow","Season","Holiday",
            "Functioning")
bike = read.csv("../data/SeoulBikeData.csv", col.names = columns)
str(bike)
```

```
## 'data.frame':    8760 obs. of  14 variables:
##  $ Date       : chr  "01/12/2017" "01/12/2017" "01/12/2017" "01/12/2017" ...
##  $ Rented     : int  254 204 173 107 78 100 181 460 930 490 ...
##  $ Hour       : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temp       : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -7.6 -6.5 ...
##  $ Humidity   : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind       : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5 ...
##  $ Visibility : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 1928 ...
##  $ Dew        : num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -19.5 -19.3 -19.8 -22.4 ...
##  $ Radiation  : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rain       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snow       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season     : chr  "Winter" "Winter" "Winter" "Winter" ...
##  $ Holiday    : chr  "No Holiday" "No Holiday" "No Holiday" "No Holiday" ...
##  $ Functioning: chr  "Yes" "Yes" "Yes" "Yes" ...
```

```
head(bike)
```

```
##         Date Rented Hour Temp Humidity Wind Visibility   Dew Radiation Rain
## 1 01/12/2017    254    0 -5.2       37  2.2       2000 -17.6         0    0
## 2 01/12/2017    204    1 -5.5       38  0.8       2000 -17.6         0    0
## 3 01/12/2017    173    2 -6.0       39  1.0       2000 -17.7         0    0
## 4 01/12/2017    107    3 -6.2       40  0.9       2000 -17.6         0    0
## 5 01/12/2017     78    4 -6.0       36  2.3       2000 -18.6         0    0
## 6 01/12/2017    100    5 -6.4       37  1.5       2000 -18.7         0    0
##   Snow Season    Holiday Functioning
## 1    0 Winter No Holiday         Yes
## 2    0 Winter No Holiday         Yes
## 3    0 Winter No Holiday         Yes
## 4    0 Winter No Holiday         Yes
## 5    0 Winter No Holiday         Yes
## 6    0 Winter No Holiday         Yes
```

```
bike$Date = as.Date(bike$Date, '%d/%m/%Y')
range(bike$Date)
```

```
## [1] "2017-12-01" "2018-11-30"
```

```
bike$Month = as.numeric(format(bike$Date,'%m'))
bike$Weekday = weekdays(bike$Date, abbreviate = TRUE)
bike$Weekend = ifelse(bike$Weekday == 'Sat' | bike$Weekday =='Sun', "Yes", "No")
```

We first converted Date into the proper date format for R to work with. Then we checked the range of the dates in our data set, which is one year's data from 2017-12-01 to 2018-11-30. So we probably don't need the year variable here. But we created several other variables like month, weekday and weekend and think these variables will help us better understand the seasonality and weekly fluctuations in bike demand.

```
bike$Season = as.factor(bike$Season)
bike$Holiday = as.factor(bike$Holiday)
bike$Functioning = as.factor(bike$Functioning)
#bike$Hour = as.factor(bike$Hour)
#bike$Month = as.factor(bike$Month)
bike$Weekday = as.factor(bike$Weekday)
bike$Weekend = as.factor(bike$Weekend)
str(bike)
```

```
## 'data.frame':    8760 obs. of  17 variables:
##  $ Date       : Date, format: "2017-12-01" "2017-12-01" ...
##  $ Rented     : int  254 204 173 107 78 100 181 460 930 490 ...
##  $ Hour       : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temp       : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -7.6 -6.5 ...
##  $ Humidity   : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind       : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5 ...
##  $ Visibility : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 1928 ...
##  $ Dew        : num  -17.6 -17.6 -17.7 -17.6 -18.6 -18.7 -19.5 -19.3 -19.8 -22.4 ...
##  $ Radiation  : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rain       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snow       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Season     : Factor w/ 4 levels "Autumn","Spring",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ Holiday    : Factor w/ 2 levels "Holiday","No Holiday": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Functioning: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Month      : num  12 12 12 12 12 12 12 12 12 12 ...
##  $ Weekday    : Factor w/ 7 levels "Fri","Mon","Sat",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Weekend    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

We successfully coerced the categorical variables into factors.

Exploratory data analysis

```
bike_num = subset(bike, select = -c(Date, Season, Holiday, Functioning, Weekday, Weekend) )
round(cor(bike_num), 2)
```

```
##             Rented  Hour  Temp Humidity  Wind Visibility   Dew Radiation  Rain
## Rented        1.00  0.41  0.54    -0.20  0.12       0.20  0.38      0.26 -0.12
## Hour          0.41  1.00  0.12    -0.24  0.29       0.10  0.00      0.15  0.01
## Temp          0.54  0.12  1.00     0.16 -0.04       0.03  0.91      0.35  0.05
## Humidity     -0.20 -0.24  0.16     1.00 -0.34      -0.54  0.54     -0.46  0.24
## Wind          0.12  0.29 -0.04    -0.34  1.00       0.17 -0.18      0.33 -0.02
## Visibility    0.20  0.10  0.03    -0.54  0.17       1.00 -0.18      0.15 -0.17
## Dew           0.38  0.00  0.91     0.54 -0.18      -0.18  1.00      0.09  0.13
## Radiation     0.26  0.15  0.35    -0.46  0.33       0.15  0.09      1.00 -0.07
## Rain         -0.12  0.01  0.05     0.24 -0.02      -0.17  0.13     -0.07  1.00
## Snow         -0.14 -0.02 -0.22     0.11  0.00      -0.12 -0.15     -0.07  0.01
## Month         0.13  0.00  0.22     0.14 -0.16       0.06  0.24     -0.03  0.01
##               Snow Month
## Rented     -0.14  0.13
## Hour       -0.02  0.00
## Temp       -0.22  0.22
## Humidity    0.11  0.14
## Wind        0.00 -0.16
## Visibility -0.12  0.06
## Dew        -0.15  0.24
## Radiation  -0.07 -0.03
## Rain        0.01  0.01
## Snow        1.00  0.05
## Month       0.05  1.00
```

```
# Comment out for now since the chart will be too big.
# Maybe we just include some most important variables in the chart?
#pairs(bike_num)
```

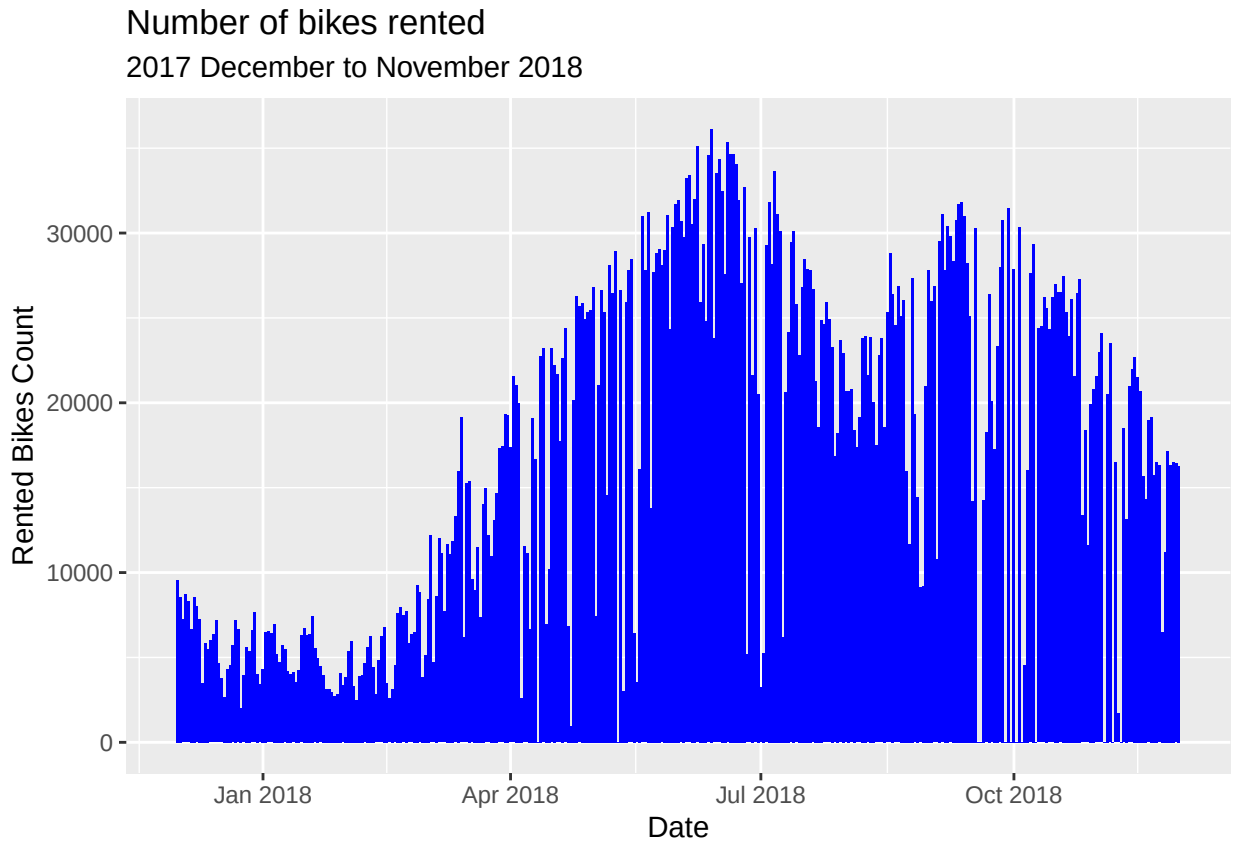
```
library(corrr)
library(dplyr)

correlations = corrr::correlate(bike_num)
top_5 = head(dplyr::arrange(corrr::stretch(correlations, remove.dups = TRUE), desc(r)), 5)
top_5
```

```
## # A tibble: 5 x 3
##   x        y         r
##   <chr>    <chr> <dbl>
## 1 Temp     Dew   0.913
## 2 Rented   Temp  0.539
## 3 Humidity Dew   0.537
## 4 Rented   Hour  0.410
## 5 Rented   Dew   0.380
```

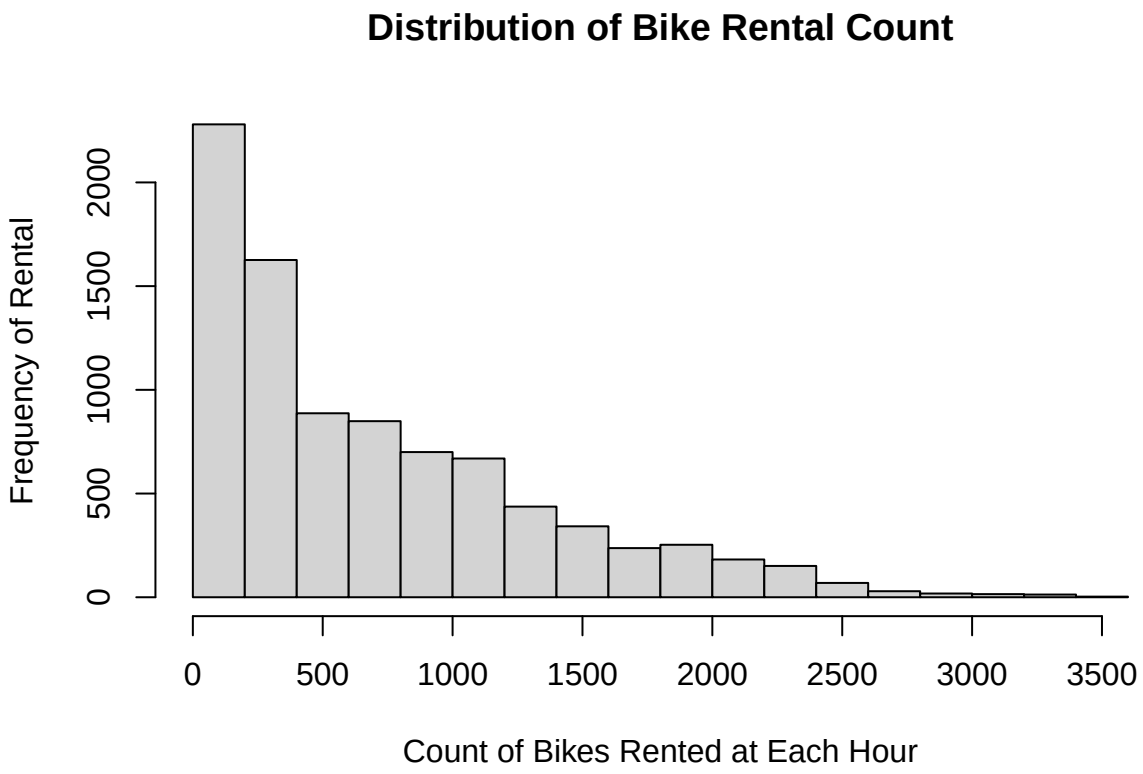
We printed out the top 5 highly correlated variables in the data set. We can see we have some highly correlated variables in the data set, which could suggest multicollinearity. We may want to address this later in the modeling process since we are interested in interpreting the coefficients.

```
library(ggplot2)
ggplot(data = bike, aes(x = Date, y = Rented)) +
      geom_bar(stat = "identity", fill = "blue") +
      labs(title = "Number of bikes rented ",
```

```
          subtitle = "2017 December to November 2018",
          x = "Date", y = "Rented Bikes Count")
```

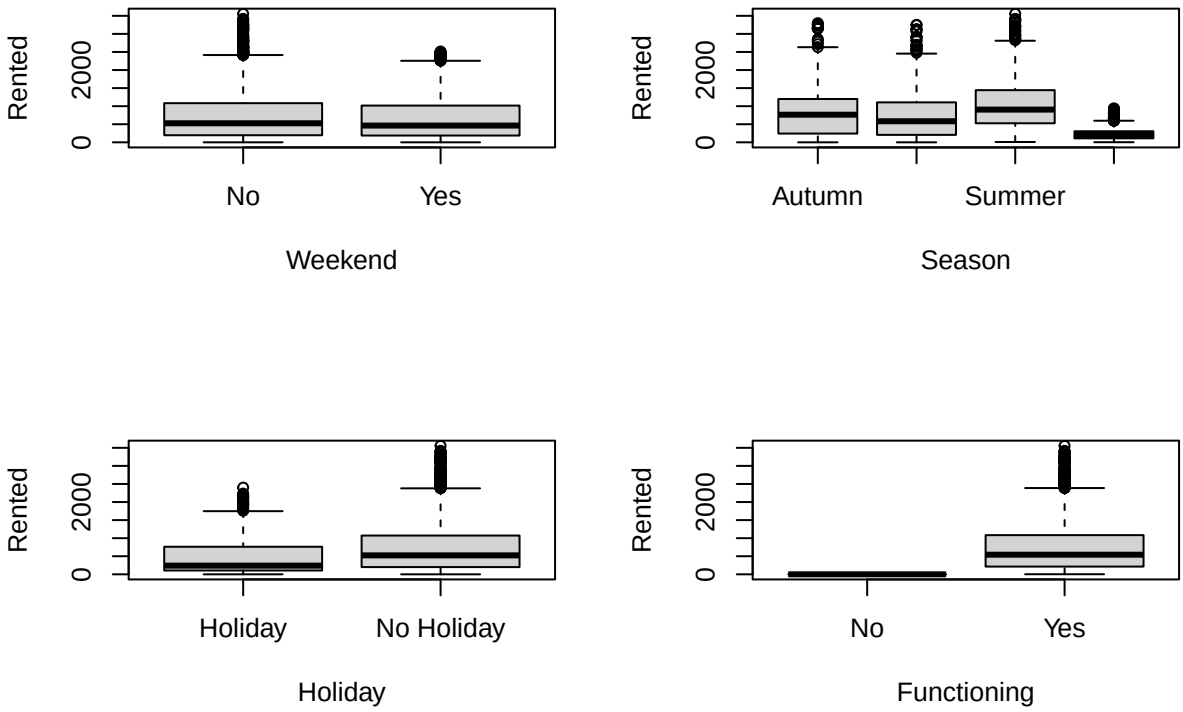



```
hist(bike$Rented,
     breaks = 25,
     ylab = 'Frequency of Rental',
     xlab = 'Count of Bikes Rented at Each Hour',
     main = 'Distribution of Bike Rental Count')
```



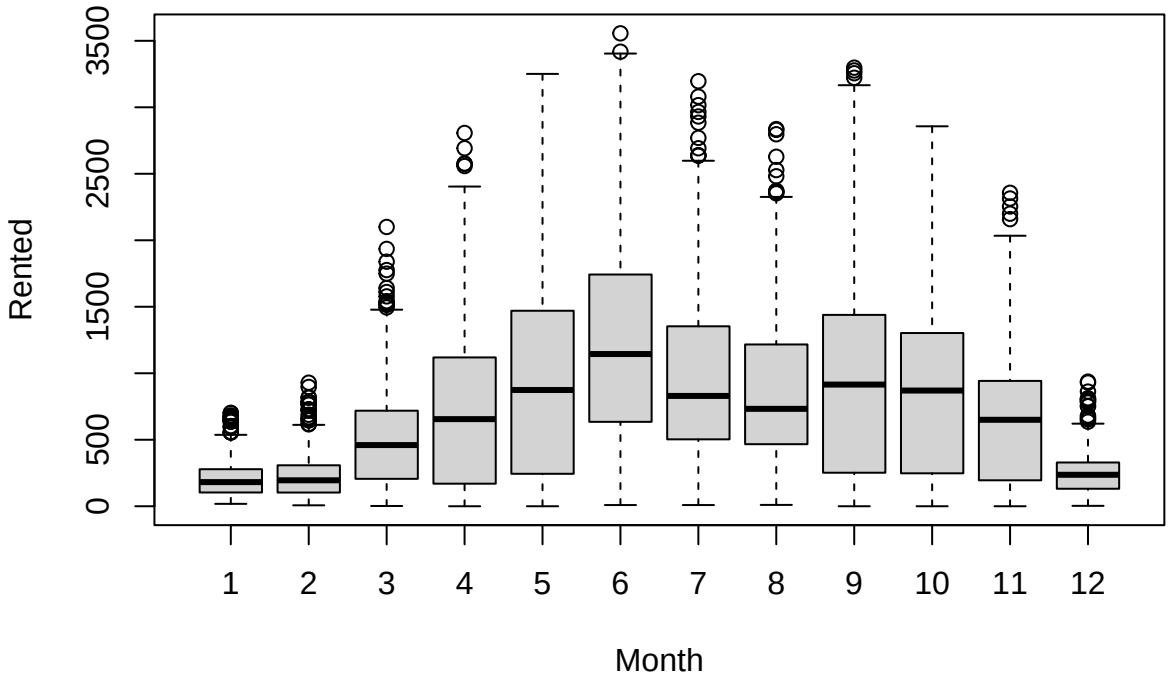
From the histogram of the response variable above, we can see the distribution is highly skewed, which means transformation may help our modeling process later.

```
par(mfrow=c(2, 2))

plot(Rented ~ Weekend, data = bike)
plot(Rented ~ Season, data = bike)
plot(Rented ~ Holiday, data = bike)
plot(Rented ~ Functioning, data = bike)
```



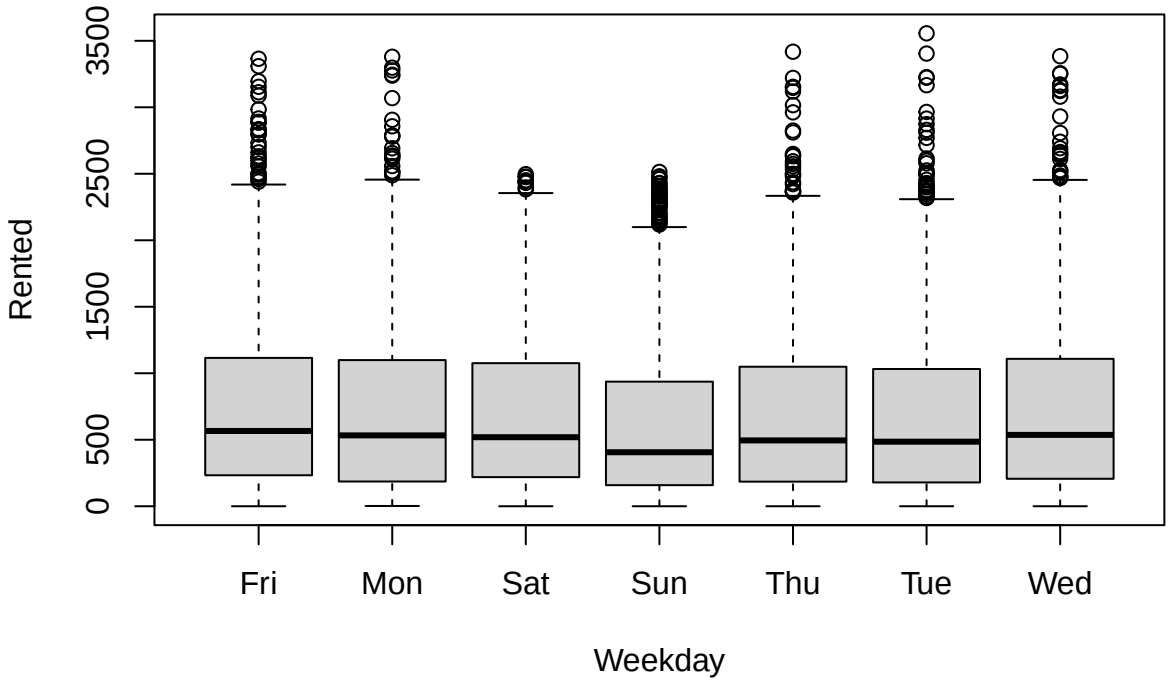
We can see we usually have higher rented bike counts on weekdays and non-holidays - perhaps more people use rental bikes as a commute method instead of using it for leisure purpose. We have highest rented bike counts during summer and lowest counts during winter, which makes sense. We have more rented bike counts during functioning days of the rental bike system, which makes sense too.

```
plot(Rented ~ as.factor(Month),
     xlab = "Month",
     data = bike)
```



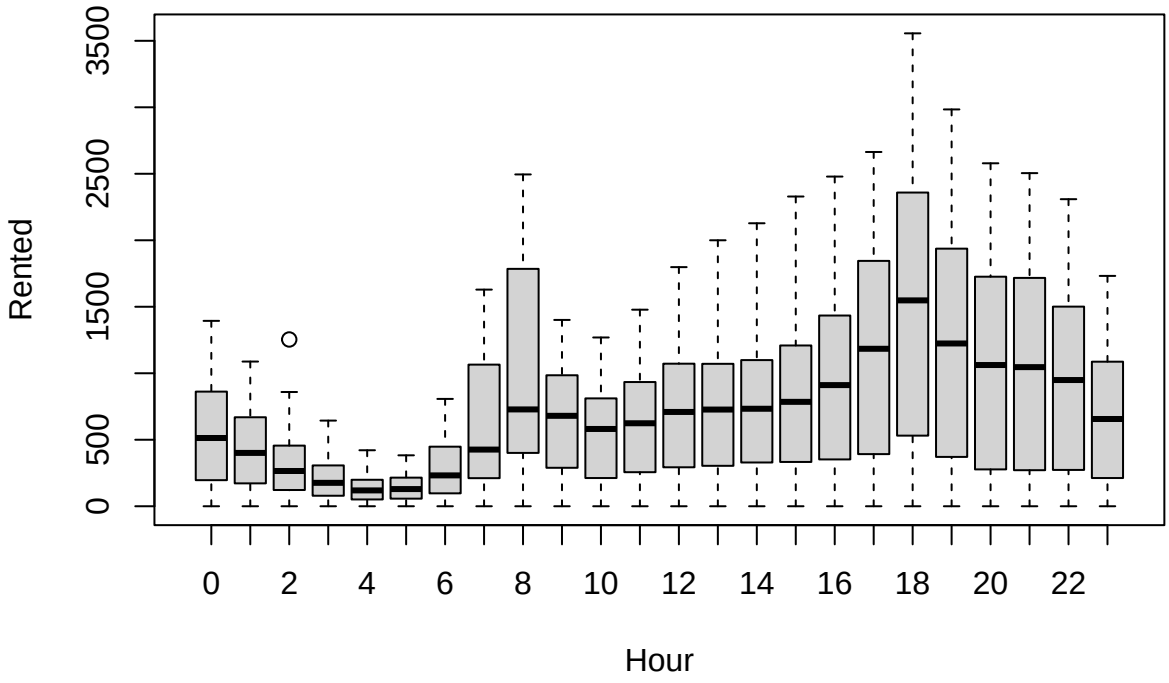
Further drill seasons down to month, we can see the rental bike count reaches the peak in Jun and the lowest point in Jan.

```
plot(Rented ~ Weekday, data = bike)
```



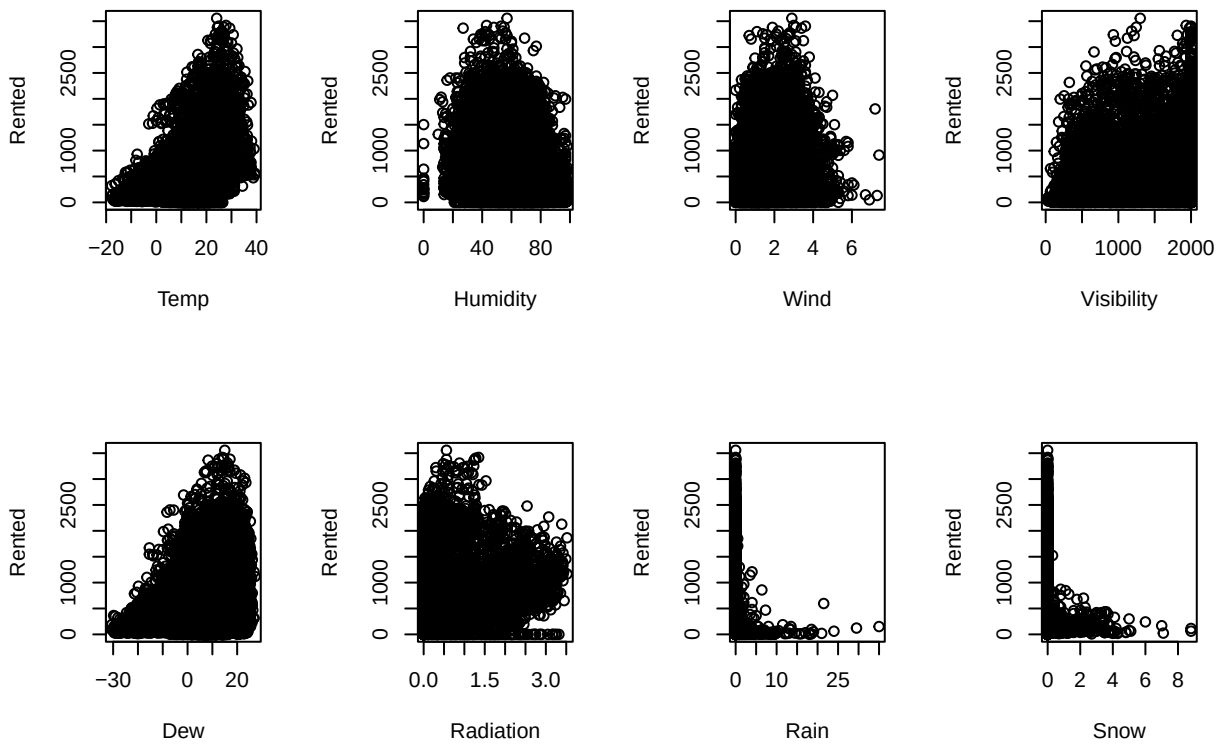
Generally speaking, we have lower demands on Saturday and Sunday, while other weekdays have similar higher demand.

```
plot(Rented ~ as.factor(Hour),
     xlab = "Hour",
     data = bike)
```



We can two peaks on the rental bike count vs hour chart: one at 8 AM and the other one at 6 PM, which correspond with the peak commute hours.

```
par(mfrow=c(2, 4))

plot(Rented ~ Temp, data = bike)
plot(Rented ~ Humidity, data = bike)
plot(Rented ~ Wind, data = bike)
plot(Rented ~ Visibility, data = bike)
plot(Rented ~ Dew, data = bike)
plot(Rented ~ Radiation, data = bike)
plot(Rented ~ Rain, data = bike)
plot(Rented ~ Snow, data = bike)
```



Rented bike counts generally increase as temperature and dew point temperature rise, but decrease quickly once they pass the optimal range. For humidity and wind speed, there also exist an obvious optimal range that lead to highest rented bike counts. The better the visibility, the higher the rented bike count is. Rainfall and Snowfall cause a sharply decreased demand of rental bikes.

Modeling

First of all, we take a look at the most basic model - an additive model using all the predictors in their original format. The result is not too bad. We got an adjusted R^2 of 0.558 and an extremely small p-value. It looks like this base model can explain more than 50% of the variance in the response variable. We also notice that we obviously have a variable that can be completely derived from another variable - Weekend, so it's redundant. Let's try to improve the model.

```
mod_naive = lm(Rented ~ ., data = bike)
summary(mod_naive)
```

```
##
## Call:
## lm(formula = Rented ~ ., data = bike)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1157   -275    -56    206   2226
##
## Coefficients: (1 not defined because of singularities)
```

```
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.66e+04   3.17e+03    8.40  < 2e-16 ***
## Date               -1.50e+00   1.78e-01   -8.44  < 2e-16 ***
## Hour                2.73e+01   7.29e-01   37.44  < 2e-16 ***
## Temp                1.84e+01   3.65e+00    5.04  4.8e-07 ***
## Humidity           -1.08e+01   1.03e+00  -10.56  < 2e-16 ***
## Wind                1.73e+01   5.07e+00    3.42  0.00063 ***
## Visibility          2.21e-03   9.84e-03    0.22  0.82204
## Dew                 9.44e+00   3.82e+00    2.47  0.01340 *
## Radiation          -8.33e+01   7.55e+00  -11.04  < 2e-16 ***
## Rain               -5.73e+01   4.24e+00  -13.53  < 2e-16 ***
## Snow                3.29e+01   1.12e+01    2.94  0.00327 **
## SeasonSpring       -4.02e+02   3.84e+01  -10.47  < 2e-16 ***
## SeasonSummer       -2.94e+02   2.53e+01  -11.59  < 2e-16 ***
## SeasonWinter       -7.64e+02   5.38e+01  -14.20  < 2e-16 ***
## HolidayNo Holiday   1.27e+02   2.15e+01    5.91  3.6e-09 ***
## FunctioningYes      9.52e+02   2.66e+01   35.75  < 2e-16 ***
## Month               1.94e+00   1.82e+00    1.06  0.28734
## WeekdayMon         -5.46e+01   1.72e+01   -3.18  0.00149 **
## WeekdaySat         -6.77e+01   1.71e+01   -3.96  7.7e-05 ***
## WeekdaySun         -1.40e+02   1.71e+01   -8.14  4.4e-16 ***
## WeekdayThu         -3.04e+01   1.71e+01   -1.77  0.07629 .
## WeekdayTue         -2.71e+01   1.72e+01   -1.58  0.11508
## WeekdayWed         -2.11e+00   1.72e+01   -0.12  0.90213
## WeekendYes                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 429 on 8737 degrees of freedom
## Multiple R-squared:  0.559,  Adjusted R-squared:  0.558
## F-statistic:  503 on 22 and 8737 DF,  p-value: <2e-16
```

Let's drop some variables: - Date: Too many distinct values for a categorical variable. - Dew: Has high correlation with Temperature. - Weekend: Created for data exploration purposes but all the information can be derived from Weekday. - Season: Can be derived from Month.

```
bike_cln = subset(bike, select = -c(Date, Dew, Weekend, Season))
str(bike_cln)
```

```
## 'data.frame':    8760 obs. of  13 variables:
##  $ Rented     : int  254 204 173 107 78 100 181 460 930 490 ...
##  $ Hour       : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Temp       : num  -5.2 -5.5 -6 -6.2 -6 -6.4 -6.6 -7.4 -7.6 -6.5 ...
##  $ Humidity   : int  37 38 39 40 36 37 35 38 37 27 ...
##  $ Wind       : num  2.2 0.8 1 0.9 2.3 1.5 1.3 0.9 1.1 0.5 ...
##  $ Visibility : int  2000 2000 2000 2000 2000 2000 2000 2000 2000 1928 ...
##  $ Radiation  : num  0 0 0 0 0 0 0 0 0.01 0.23 ...
##  $ Rain       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Snow       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Holiday    : Factor w/ 2 levels "Holiday","No Holiday": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Functioning: Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Month      : num  12 12 12 12 12 12 12 12 12 12 ...
##  $ Weekday    : Factor w/ 7 levels "Fri","Mon","Sat",..: 1 1 1 1 1 1 1 1 1 1 ...
```

A basic additive model on the cleaner dataset gives us a R^2 at 0.533.

```
mod_additive = lm(Rented ~ ., data = bike_cln)
summary(mod_additive)$adj.r.squared
```

```
## [1] 0.533
```

Checking for multicollinearity in the data. We don't have high variance inflation factors at this moment.

```
library(faraway)
vif(mod_additive)[vif(mod_additive) > 5]
```

```
## named numeric(0)
```

Let's try to convert Hour and Month to factor variables - although they are numeric numbers now, they can only have certain values and we can't say the difference in average rented bike counts between Hour 1 and 2 will be the same as the difference in average rented bike counts between Hour 17 and 18. After conversion, the model gives a much better adjusted R^2 score at 0.6995 now.

```
# Copy dataset to test Hour and Month as factor variables
bike_factor = data.frame(bike_cln)
bike_factor$Hour = as.factor(bike_factor$Hour)
bike_factor$Month = as.factor(bike_factor$Month)
# Build the model using Hour and Month as factor variables
mod_additive_factor = lm(Rented ~ ., data = bike_factor)
summary(mod_additive_factor)$adj.r.squared
```

```
## [1] 0.6995
```

Checking for multicollinearity in the data. Now we can see Temperature and some Month variables are high variance inflation factors. This makes sense since Temperature usually has some correlation with seasonality / month.

```
vif(mod_additive_factor)[vif(mod_additive_factor) > 5]
```

```
##   Temp Month6 Month7 Month8 Month9
## 11.310  6.047  8.367  8.978  6.197
```

Let's try to fit an interaction model using the dataset without the factor variables conversion. The adjusted R^2 score is better than the additive model using the same dataset but worse than the additive model using the dataset with Hour and Month as factor variables.

```
mod_interact = lm(Rented ~ . ^ 2, data = bike_cln)
summary(mod_interact)$adj.r.squared
```

```
## [1] 0.6611
```

Let's try to fit an interaction model using the dataset with the factor variables conversion. The adjusted R^2 score has been improved greatly to 0.9025.

```
mod_interact_factor = lm(Rented ~ . ^ 2, data = bike_factor)
summary(mod_interact_factor)$adj.r.squared
```

```
## [1] 0.9025
```

**RZ Note: I changed the model comparison contents here since I believe anova need to compare nested models?

By comparing the additive model and the interaction model using the two datasets, we can see the p-value is extremely small in both cases. So we prefer the interaction model.

```
anova(mod_additive, mod_interact)
```

```
## Analysis of Variance Table
##
## Model 1: Rented ~ Hour + Temp + Humidity + Wind + Visibility + Radiation +
##     Rain + Snow + Holiday + Functioning + Month + Weekday
## Model 2: Rented ~ (Hour + Temp + Humidity + Wind + Visibility + Radiation +
##     Rain + Snow + Holiday + Functioning + Month + Weekday)^2
##   Res.Df      RSS  Df Sum of Sq    F Pr(>F)
## 1   8742 1.70e+09
## 2   8623 1.22e+09 119 482548029 28.8 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod_additive_factor, mod_interact_factor)
```

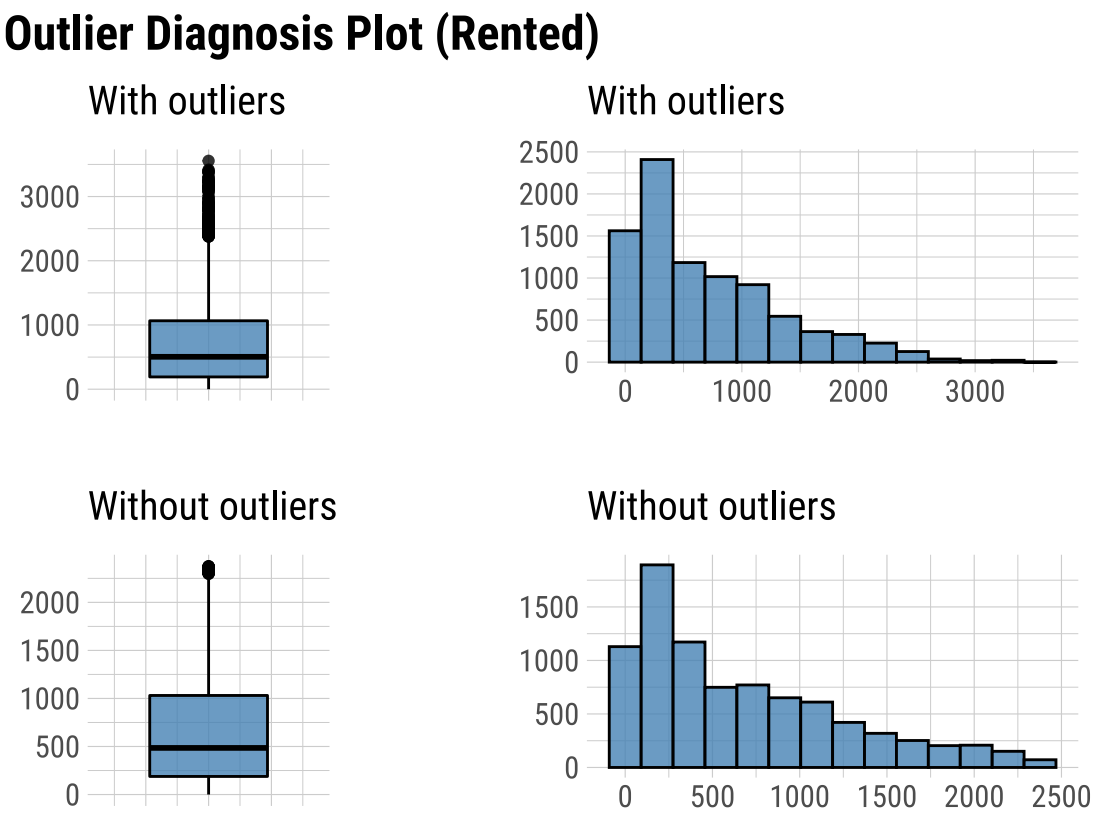
```
## Analysis of Variance Table
##
## Model 1: Rented ~ Hour + Temp + Humidity + Wind + Visibility + Radiation +
##     Rain + Snow + Holiday + Functioning + Month + Weekday
## Model 2: Rented ~ (Hour + Temp + Humidity + Wind + Visibility + Radiation +
##     Rain + Snow + Holiday + Functioning + Month + Weekday)^2
##   Res.Df      RSS  Df Sum of Sq    F Pr(>F)
## 1   8710 1.09e+09
## 2   7887 3.20e+08 823 769074322 23.1 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
diagnose_numeric(bike_cln)
```

```
## # A tibble: 10 x 10
##    variables    min     Q1      mean  median      Q3     max  zero minus outlier
##    <chr>      <dbl>  <dbl>     <dbl>   <dbl>   <dbl>   <dbl> <int> <int>   <int>
##  1 Rented       0   191     705.      504.   1065.   3556      295     0     157
##  2 Hour         0     5.75   11.5      11.5    17.2    23      365     0       0
##  3 Temp       -17.8   3.5    12.9      13.7    22.5    39.4     21  1433       0
##  4 Humidity     0    42      58.2      57      74      98       17     0       0
##  5 Wind         0     0.9     1.72      1.5     2.3     7.4     74     0     161
##  6 Visibility  27   940    1437.     1698    2000    2000        0     0       0
##  7 Radiation    0     0       0.569     0.01    0.93    3.52  4300     0     641
```

```
##  8 Rain          0     0       0.149     0       0      35     8232     0     528
##  9 Snow          0     0       0.0751    0       0       8.8   8317     0     443
## 10 Month         1     4       6.53      7      10      12       0     0       0
```

```
bike_cln %>% plot_outlier(Rented)
```

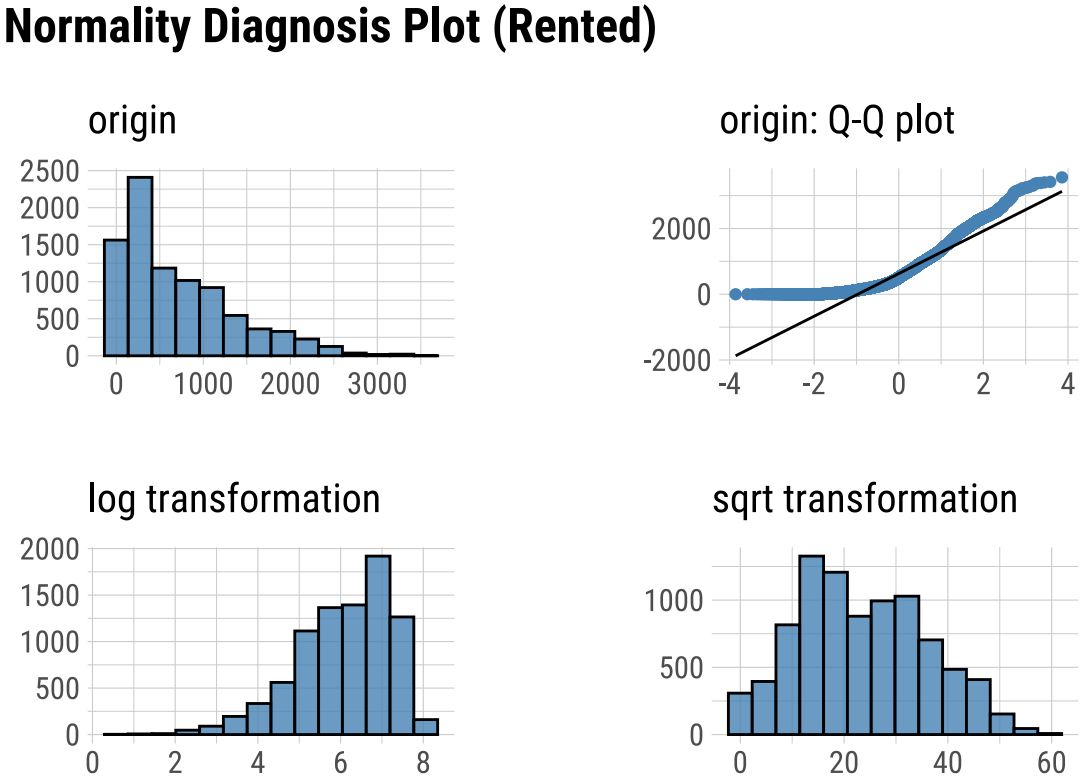



```
normality(bike_cln)
```

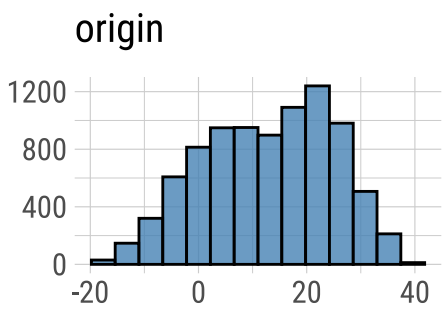
```
## # A tibble: 10 x 4
##    vars       statistic  p_value sample
##    <chr>          <dbl>    <dbl>  <dbl>
##  1 Rented         0.879 2.08e-52   5000
##  2 Hour           0.952 9.94e-38   5000
##  3 Temp           0.980 2.96e-26   5000
##  4 Humidity       0.981 1.99e-25   5000
##  5 Wind           0.944 5.62e-40   5000
##  6 Visibility     0.833 2.76e-58   5000
##  7 Radiation      0.705 3.95e-69   5000
##  8 Rain           0.111 4.40e-93   5000
##  9 Snow           0.178 2.94e-91   5000
## 10 Month          0.940 5.48e-41   5000
```

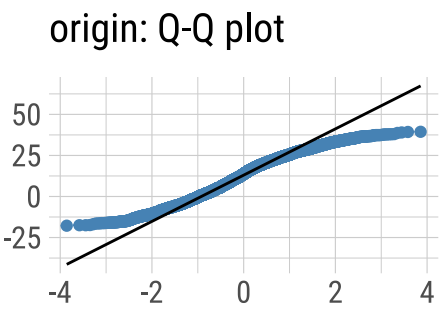
Check normality of each numeric variable.

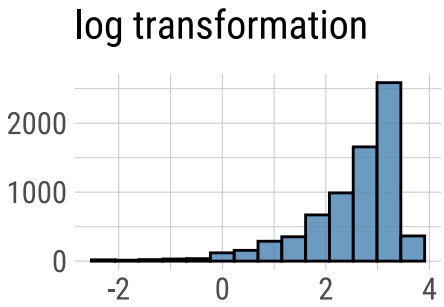
```
bike_factor %>% plot_normality(Rented, Temp, Humidity, Wind, Visibility,
                               Radiation, Rain, Snow)
```

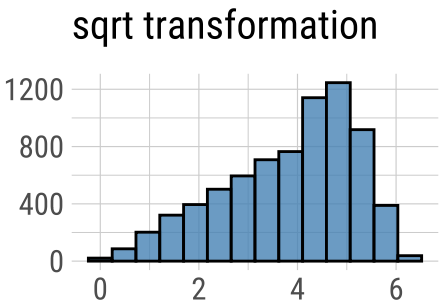


Normality Diagnosis Plot (Temp)

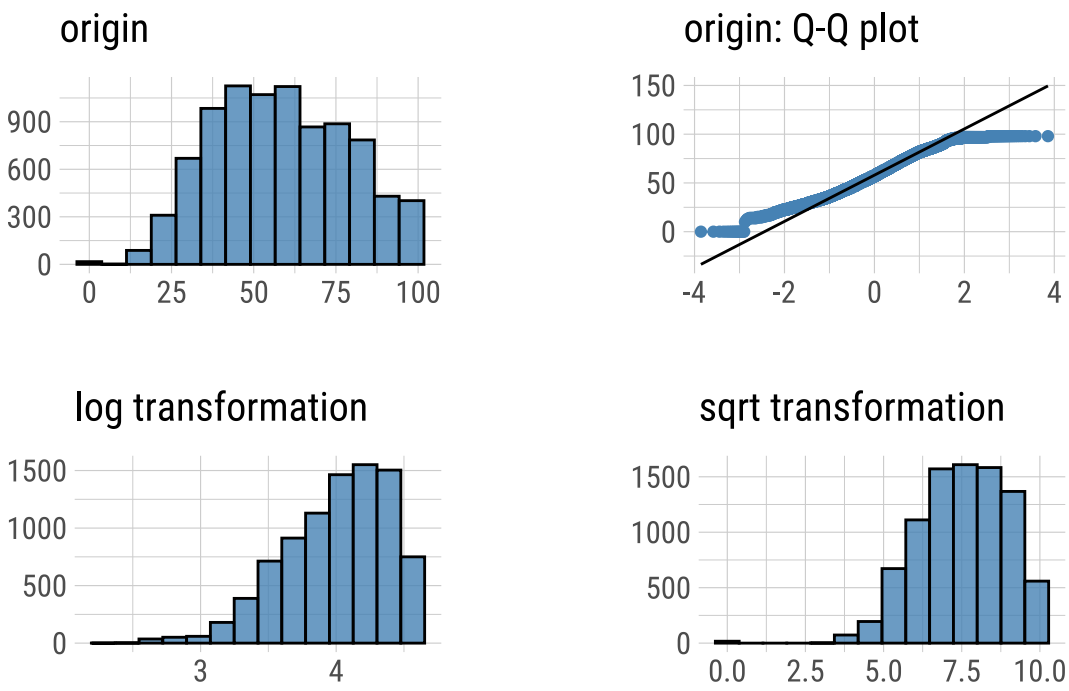




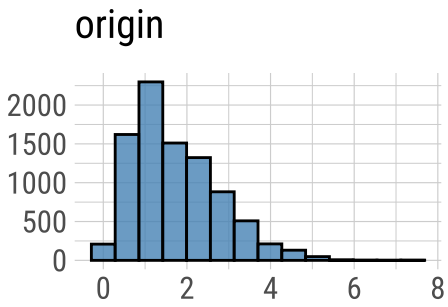


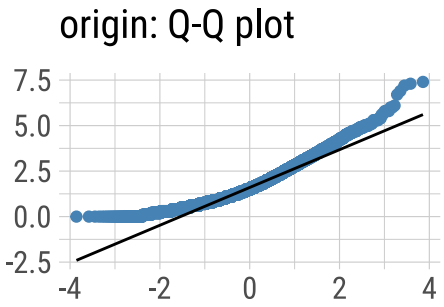


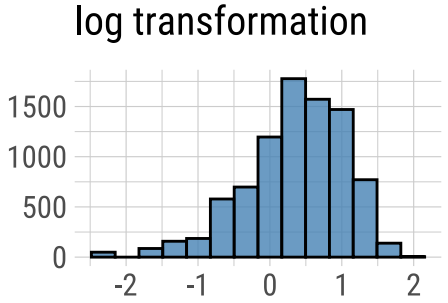
Normality Diagnosis Plot (Humidity)

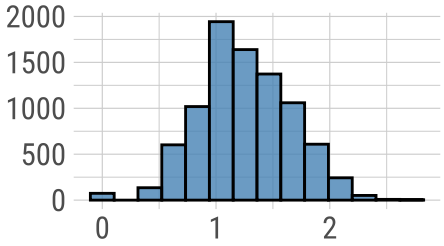


Normality Diagnosis Plot (Wind)

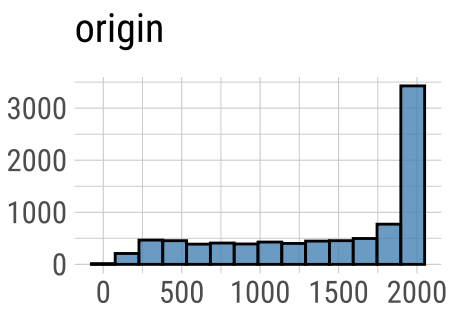


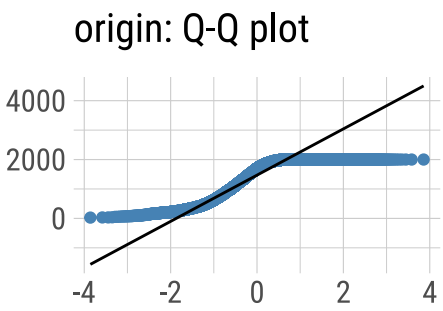


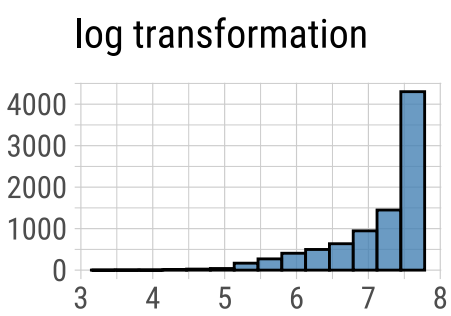


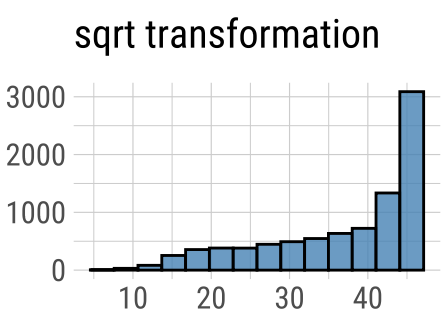


Normality Diagnosis Plot (Visibility)

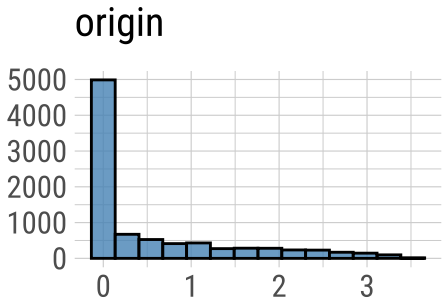


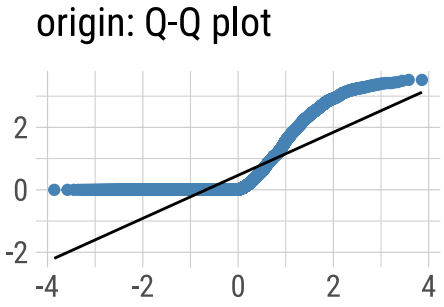


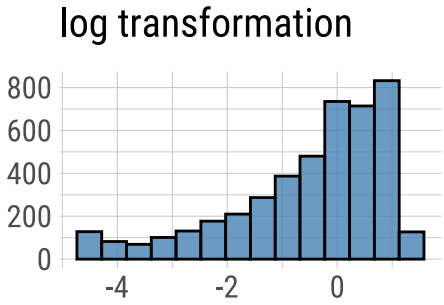


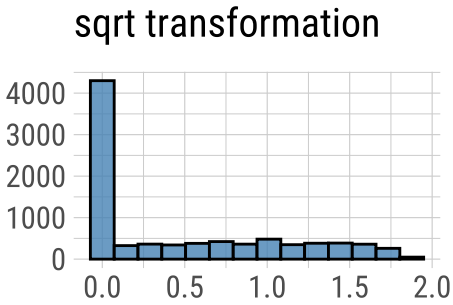


Normality Diagnosis Plot (Radiation)

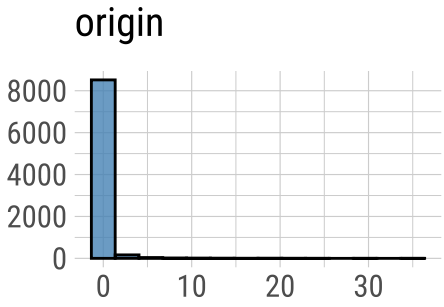


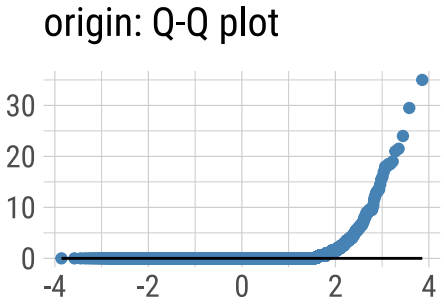


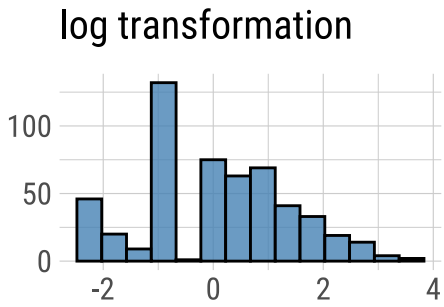


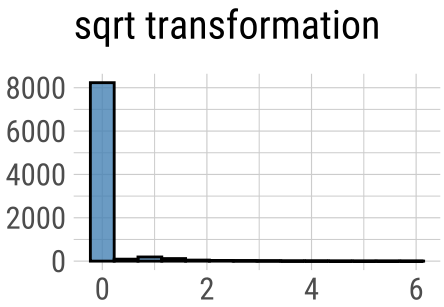


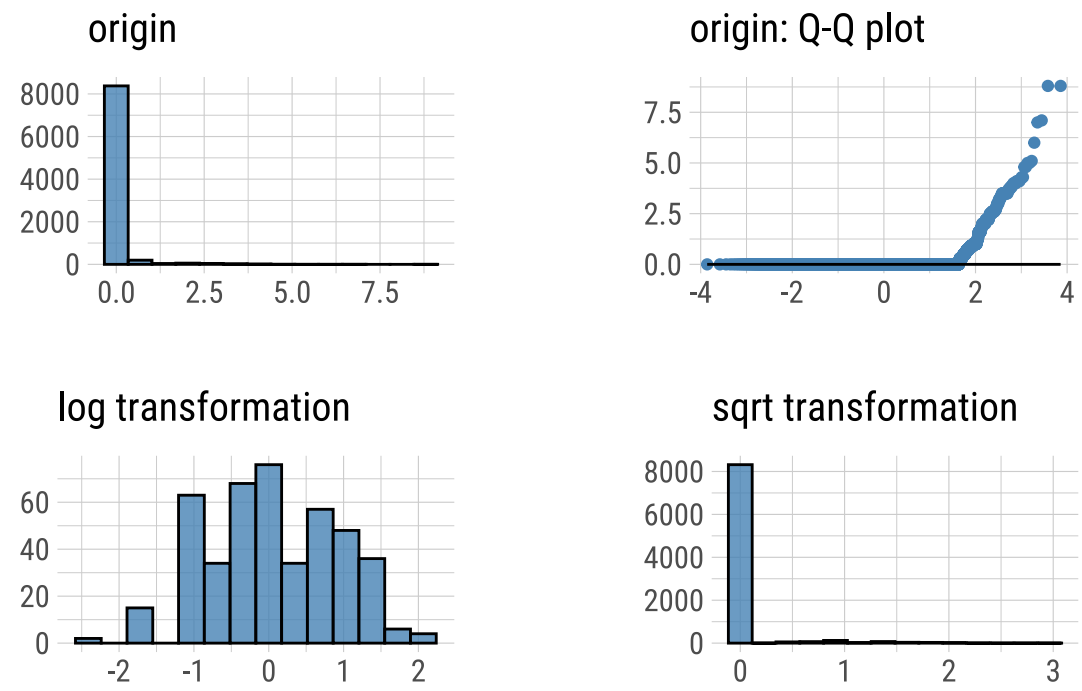
Normality Diagnosis Plot (Rain)











We will use Hour as factor going forward.

Check model after taking square root of the response variable:

```
# add 1 to response variable to avoid errors in log
bike_factor$Rented = bike_factor$Rented + 1
```

After taking square root of the response variable, the adjusted R^2 is further improved to 0.9187 now.

```
mod_interact_sq = lm(sqrt(Rented) ~ . ^ 2, data = bike_factor)
summary(mod_interact_sq)$adj.r.squared
```

```
## [1] 0.9187
```

Check log transformation on the response variable. The adjusted R^2 is further improved to 0.9363 now.

```
mod_interact_log = lm(log(Rented) ~ . ^ 2, data = bike_factor)
summary(mod_interact_log)$adj.r.squared
```

```
## [1] 0.9363
```

Let's run AIC backward searching on the interaction model.

```r
mod_int_aic = step(mod_interact, direction = "backward", trace = 0)
coef(mod_int_aic)
```

```
##                   (Intercept)                          Hour
##                     -1.374e+03                    -2.159e+01
##                          Temp                      Humidity
##                      4.574e+01                     1.862e+01
##                          Wind                    Visibility
##                      1.906e+02                    -5.017e-02
##                     Radiation                          Rain
##                      2.103e+01                    -1.039e+03
##                          Snow             HolidayNo Holiday
##                      3.232e+02                     6.993e+01
##                FunctioningYes                         Month
##                      8.647e+02                     3.570e+01
##                    WeekdayMon                    WeekdaySat
##                      2.202e+02                     6.953e+01
##                    WeekdaySun                    WeekdayThu
##                      1.193e+02                     1.364e+02
##                    WeekdayTue                    WeekdayWed
##                      9.314e+01                     1.924e+02
##                     Hour:Temp                 Hour:Humidity
##                      1.696e+00                    -3.841e-01
##                Hour:Visibility                Hour:Radiation
##                      5.378e-03                     5.280e+00
##                     Hour:Rain                     Hour:Snow
##                     -1.145e+00                     3.801e+00
##       Hour:HolidayNo Holiday           Hour:FunctioningYes
##                      9.429e+00                     3.284e+01
##                    Hour:Month                Hour:WeekdayMon
##                      5.729e-01                     3.900e+00
##                Hour:WeekdaySat                Hour:WeekdaySun
##                     -6.662e+00                    -8.051e+00
##                Hour:WeekdayThu                Hour:WeekdayTue
##                      1.397e+00                     2.705e+00
##                Hour:WeekdayWed                  Temp:Humidity
##                      3.123e+00                    -7.947e-01
##                     Temp:Wind                 Temp:Visibility
##                      3.511e+00                    -1.324e-02
##                  Temp:Radiation                     Temp:Snow
##                     -1.636e+01                    -3.053e+01
##            Temp:FunctioningYes                  Humidity:Wind
##                      3.038e+01                    -2.466e+00
##            Humidity:Visibility               Humidity:Radiation
##                      3.562e-03                     7.183e+00
##                  Humidity:Rain                 Humidity:Snow
##                      1.190e+01                    -2.995e+00
##       Humidity:FunctioningYes                  Humidity:Month
##                     -1.185e+01                    -3.074e-01
##            Humidity:WeekdayMon               Humidity:WeekdaySat
##                     -3.368e+00                     8.524e-01
##            Humidity:WeekdaySun               Humidity:WeekdayThu
##                     -1.938e+00                    -1.472e+00
```

```
##          Humidity:WeekdayTue          Humidity:WeekdayWed
##                    6.641e-01                    -4.029e-01
##             Wind:Visibility               Wind:Radiation
##                   -2.832e-02                    -5.628e+01
##                    Wind:Rain                    Wind:Month
##                    8.640e+00                    -2.967e+00
##             Wind:WeekdayMon              Wind:WeekdaySat
##                    1.321e+00                     2.449e+00
##             Wind:WeekdaySun              Wind:WeekdayThu
##                    3.088e+01                     4.478e+00
##             Wind:WeekdayTue              Wind:WeekdayWed
##                   -3.457e+01                     1.327e+01
##        Visibility:Radiation              Visibility:Snow
##                    9.062e-02                    -1.022e-01
## Visibility:HolidayNo Holiday             Visibility:Month
##                    8.980e-02                    -8.143e-03
##       Visibility:WeekdayMon        Visibility:WeekdaySat
##                   -9.851e-02                    -5.402e-03
##       Visibility:WeekdaySun        Visibility:WeekdayThu
##                   -9.231e-02                    -5.095e-02
##       Visibility:WeekdayTue        Visibility:WeekdayWed
##                   -7.087e-03                    -4.694e-02
##              Radiation:Snow  Radiation:HolidayNo Holiday
##                   -6.405e+01                    -7.078e+01
##    Radiation:FunctioningYes              Radiation:Month
##                   -1.440e+02                    -7.877e+00
##         Radiation:WeekdayMon         Radiation:WeekdaySat
##                   -4.436e+01                     1.339e+02
##         Radiation:WeekdaySun         Radiation:WeekdayThu
##                    7.390e+01                    -2.440e+01
##         Radiation:WeekdayTue         Radiation:WeekdayWed
##                   -7.826e-01                     1.300e+01
##       Rain:HolidayNo Holiday          Rain:FunctioningYes
##                   -7.658e+01                    -4.878e+01
##                   Rain:Month                   Snow:Month
##                   -4.208e+00                    -1.260e+01
##              Snow:WeekdayMon              Snow:WeekdaySat
##                   -1.562e+01                     1.846e+02
##              Snow:WeekdaySun              Snow:WeekdayThu
##                    1.241e+02                    -8.055e+01
##              Snow:WeekdayTue              Snow:WeekdayWed
##                   -6.831e+01                    -1.265e+02
## HolidayNo Holiday:WeekdayMon HolidayNo Holiday:WeekdaySat
##                   -4.692e+01                    -2.258e+02
## HolidayNo Holiday:WeekdaySun HolidayNo Holiday:WeekdayThu
##                   -7.129e+01                    -7.939e+01
## HolidayNo Holiday:WeekdayTue HolidayNo Holiday:WeekdayWed
##                   -2.082e+02                    -2.630e+02
##             Month:WeekdayMon             Month:WeekdaySat
##                    1.393e+01                     3.796e+00
##             Month:WeekdaySun             Month:WeekdayThu
##                    9.988e+00                     9.091e+00
##             Month:WeekdayTue             Month:WeekdayWed
##                    1.030e+01                     1.377e+01
```

```
summary(mod_int_aic)$adj.r.squared
```

```
## [1] 0.6612
```

Evaluating the metrics

Adjusted R^2 Score

We have already seen the adjusted R^2 in the previous section. Now let's summarize them.

```
adj_r2 = data.frame(matrix(ncol = 1, nrow = 0))
colnames(adj_r2) = c("Adjusted R_2 Score")

adj_r2[1, ] = summary(mod_naive)$adj.r.squared
adj_r2[2, ] = summary(mod_additive)$adj.r.squared
adj_r2[3, ] = summary(mod_additive_factor)$adj.r.squared
adj_r2[4, ] = summary(mod_interact)$adj.r.squared
adj_r2[5, ] = summary(mod_interact_factor)$adj.r.squared
adj_r2[6, ] = summary(mod_interact_sq)$adj.r.squared
adj_r2[7, ] = summary(mod_interact_log)$adj.r.squared
adj_r2[8, ] = summary(mod_int_aic)$adj.r.squared

row.names(adj_r2) = c("mod_naive",
                      "mod_additive",
                      "mod_additive_factor",
                      "mod_interact",
                      "mod_interact_factor",
                      "mod_interact_sq",
                      "mod_interact_log",
                      "mod_int_aic")

knitr::kable(adj_r2, "pipe")
```

	Adjusted R_2 Score
mod_naive	0.5576
mod_additive	0.5330
mod_additive_factor	0.6995
mod_interact	0.6611
mod_interact_factor	0.9025
mod_interact_sq	0.9187
mod_interact_log	0.9363
mod_int_aic	0.6612

The interaction model with log transformation of the response variable with the factor variables has the best adjusted R^2 score.

Cross-validated RMSE

Define the function to calculate cross-validated RMSE of different models.

```
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}

loocv_rmse = data.frame(matrix(ncol = 1, nrow = 0))
colnames(loocv_rmse) = c("Cross-validated RMSE")

loocv_rmse[1, ] = calc_loocv_rmse(mod_naive)
loocv_rmse[2, ] = calc_loocv_rmse(mod_additive)
loocv_rmse[3, ] = calc_loocv_rmse(mod_additive_factor)
loocv_rmse[4, ] = calc_loocv_rmse(mod_interact)
loocv_rmse[5, ] = calc_loocv_rmse(mod_interact_factor)
loocv_rmse[6, ] = calc_loocv_rmse(mod_int_aic)

row.names(loocv_rmse) = c("mod_naive",
                          "mod_additive",
                          "mod_additive_factor",
                          "mod_interact",
                          "mod_interact_factor",
                          "mod_int_aic")

knitr::kable(loocv_rmse, "pipe")
```

	Cross-validated RMSE
mod_naive	429.6
mod_additive	441.2
mod_additive_factor	354.8
mod_interact	379.2
mod_interact_factor	227.3
mod_int_aic	377.1

We can see the interaction model with the factor variables has the lowest cross-validated RMSE. However, we can't easily apply this function to the models with transformed response variables.

RMSE on test dataset

Split data into train/test to test the models. We are only using the dateset with the Hour and Month factor variables now, since we know the factor variables greatly boosted the model performance.

```
set.seed(420)
bike_idx = sample(1:nrow(bike_factor), 8000)
bike_trn = bike_factor[bike_idx, ]
bike_tst = bike_factor[-bike_idx, ]
```

Define the function to calculate RMSE.

```
RMSE <- function(model, data, trans = "") {
  n = nrow(data)
  y_hat = predict(model, data)
  if(trans=="log") {
```

```
    resid = data$Rented - exp(y_hat)
  } else if (trans=="sqrt"){
    resid = data$Rented - y_hat ^ 2
  } else {
    resid = data$Rented - y_hat
  }
  sqrt(sum(resid ^ 2) / n)
}
```

We can see the interaction model with sqrt transformation on the response variable has the lowest RMSE on the test dataset.

```
mod_additive_trn = lm(Rented ~ ., data = bike_trn)
mod_interact_trn = lm(Rented ~ . ^ 2, data = bike_trn)
mod_interact_sq_trn = lm(sqrt(Rented) ~ . ^ 2, data = bike_trn)
mod_interact_log_trn = lm(log(Rented) ~ . ^ 2, data = bike_trn)

test_rmse = data.frame(matrix(ncol = 1, nrow = 0))
colnames(test_rmse) = c("Test Dataset RMSE")

test_rmse[1, ] = RMSE(mod_additive_trn, bike_tst)
test_rmse[2, ] = RMSE(mod_interact_trn, bike_tst)
test_rmse[3, ] = RMSE(mod_interact_sq_trn, bike_tst, trans = "sqrt")
test_rmse[4, ] = RMSE(mod_interact_log_trn, bike_tst, trans = "log")

row.names(test_rmse) = c("mod_additive_trn",
                         "mod_interact_trn",
                         "mod_interact_sq_trn",
                         "mod_interact_log_trn")

knitr::kable(test_rmse, "pipe")
```

	Test Dataset RMSE
mod_additive_trn	364.8
mod_interact_trn	232.6
mod_interact_sq_trn	204.1
mod_interact_log_trn	242.1

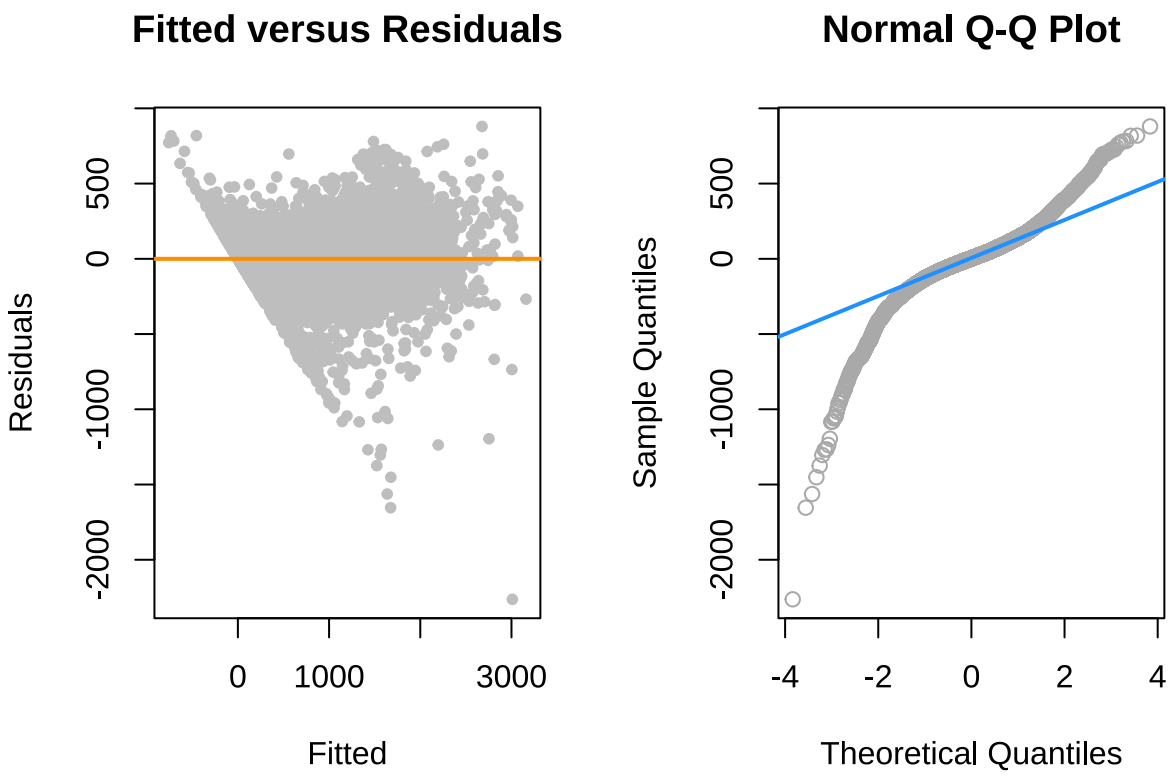
We can see the interaction model with sqrt transformation of the response variable has the lowest RMSE on the test dataset.

Additional Visulizations

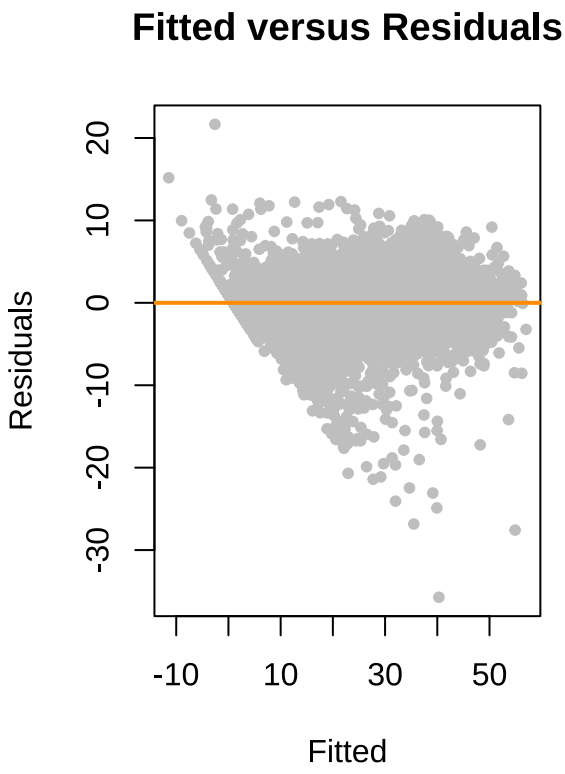
Check fitted vs residuals for the best models from the metrics evaluation section:

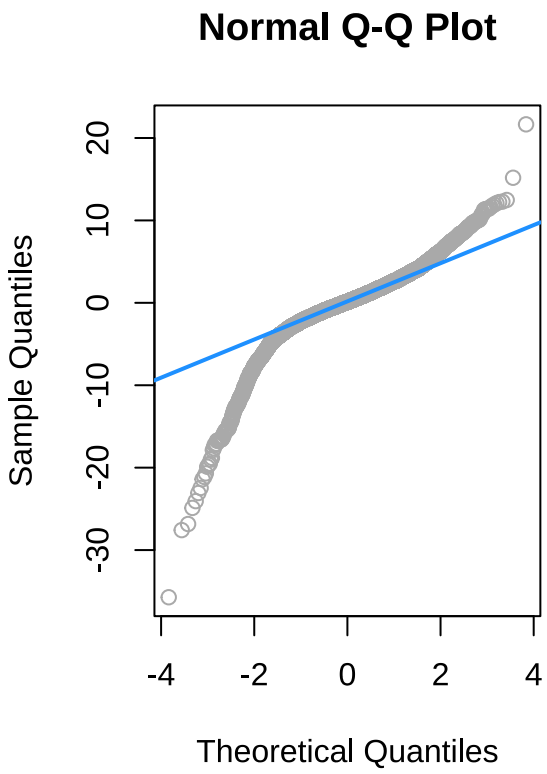
```
par(mfrow = c(1, 2))

plot(fitted(mod_interact_trn), resid(mod_interact_trn), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
```

```
qqnorm(resid(mod_interact_trn), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(mod_interact_trn), col = "dodgerblue", lwd = 2)
```

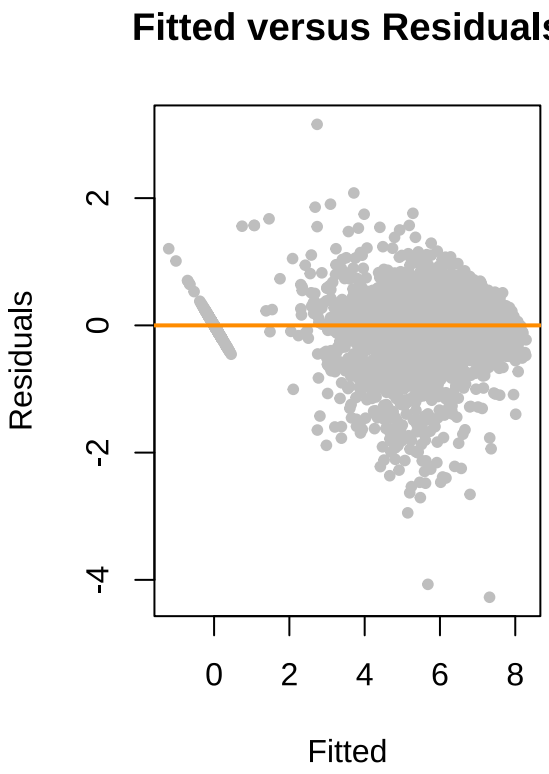


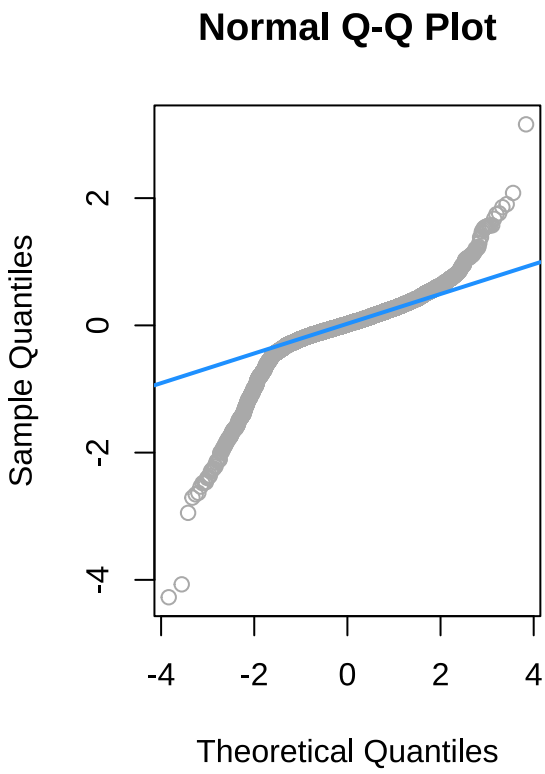

```
par(mfrow = c(1, 2))

plot(fitted(mod_interact_sq_trn), resid(mod_interact_sq_trn), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(mod_interact_sq_trn), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(mod_interact_sq_trn), col = "dodgerblue", lwd = 2)
```






```r
par(mfrow = c(1, 2))

plot(fitted(mod_interact_log_trn), resid(mod_interact_log_trn), col = "grey", pch = 20,
     xlab = "Fitted", ylab = "Residuals", main = "Fitted versus Residuals")
abline(h = 0, col = "darkorange", lwd = 2)

qqnorm(resid(mod_interact_log_trn), main = "Normal Q-Q Plot", col = "darkgrey")
qqline(resid(mod_interact_log_trn), col = "dodgerblue", lwd = 2)
```





Conclusions