STAC67 FINAL PROJECT

Group 19

2022-04-10

Background and Significance

Seoul Rental Bike data is a compilation of information such as time (date, hour), environmental climate (daily temperature, humidity, season, etc.) and whether there are special occasions such as holidays that may impact the demand for hourly bike rentals in Seoul. The data is provided at: https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand (https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand), in a dataframe with 8760 rows under 14 columns. This data is collected up until 2018 which covers the many seasons and its corresponding seasonal weather, which is included among the explanatory variables and the dependent variable hourly bike demand. In this study, given the available variables, is hourly demand impacted by the excess of leisurely time during holidays? Can we derive a demographic profile of the individual that frequents the use of this infrastructure? Can we couple this with employment density? Is weather a deterrent? This analysis serves to promote improvements to Seoul's biking infrastructure by informing supply to hourly demand for availability of bikes where it is to be communicated to municipal operators where they can make informed decisions. Unfortunately, the absence of regional or locational data for hourly demand such as bike stops will limit the scope of this analysis.

We will complete a basic data cleaning routine, and a series of data explortation exercises to gain an intuitive understanding of the data and to begin to build a profile of Seoul's hourly demand for bikes. This is followed by an analysis of significant predictors for a parsimonious dataset.

Setup

```
library(tidyverse)
library(readxl)
library(car)
library(olsrr)
library(ggpubr)
library(broom)
library(MASS)
#Library(GGally)
library(corrplot)
```

Data Import

```
dfBikeRaw <- read_excel("SeoulBikeData.xlsx")
```

Data Cleaning

```
dfBikeClean = dfBikeRaw
colnames(dfBikeClean) <- c("Date",
                       "Rented_Bike_Count",
                       "Hour",
                       "Temperature",
                       "Humidity",
                       "Wind_Speed",
                       "Visibility",
                       "Dew_Point_Temp",
                       "Solar_Radiation",
                       "Rainfall",
                       "Snowfall",
                       "Seasons",
                       "Holiday",
                       "Functioning_Days")
```

```
dfBikeClean$Seasons_Numeric <- factor(dfBikeClean$Seasons)
dfBikeClean$Seasons_Numeric <- as.double(unclass(dfBikeClean$Seasons_Numeric))

dfBikeClean$Holiday_Numeric <- factor(dfBikeClean$Holiday)
dfBikeClean$Holiday_Numeric <- as.double(unclass(dfBikeClean$Holiday_Numeric))

dfBikeClean$Functioning_Days_Numeric <- factor(dfBikeClean$Functioning_Days)
dfBikeClean$Functioning_Days_Numeric <- as.double(unclass(dfBikeClean$Functioning_Days_Numeric))

unique(dfBikeClean$Seasons_Numeric)
```

```
## [1] 4 2 3 1
```

```
unique(dfBikeClean$Holiday_Numeric)
```

```
## [1] 2 1
```

```
unique(dfBikeClean$Functioning_Days_Numeric)
```

```
## [1] 2 1
```

Converting the Categorical variables into numerics: 1. Winter | Spring | Summer | Autumn -> 4 | 2 | 3 | 1 2. No Holiday | Holiday -> 2 | 1 3. Yes | No -> 2 | 1

```
head(dfBikeClean)
```

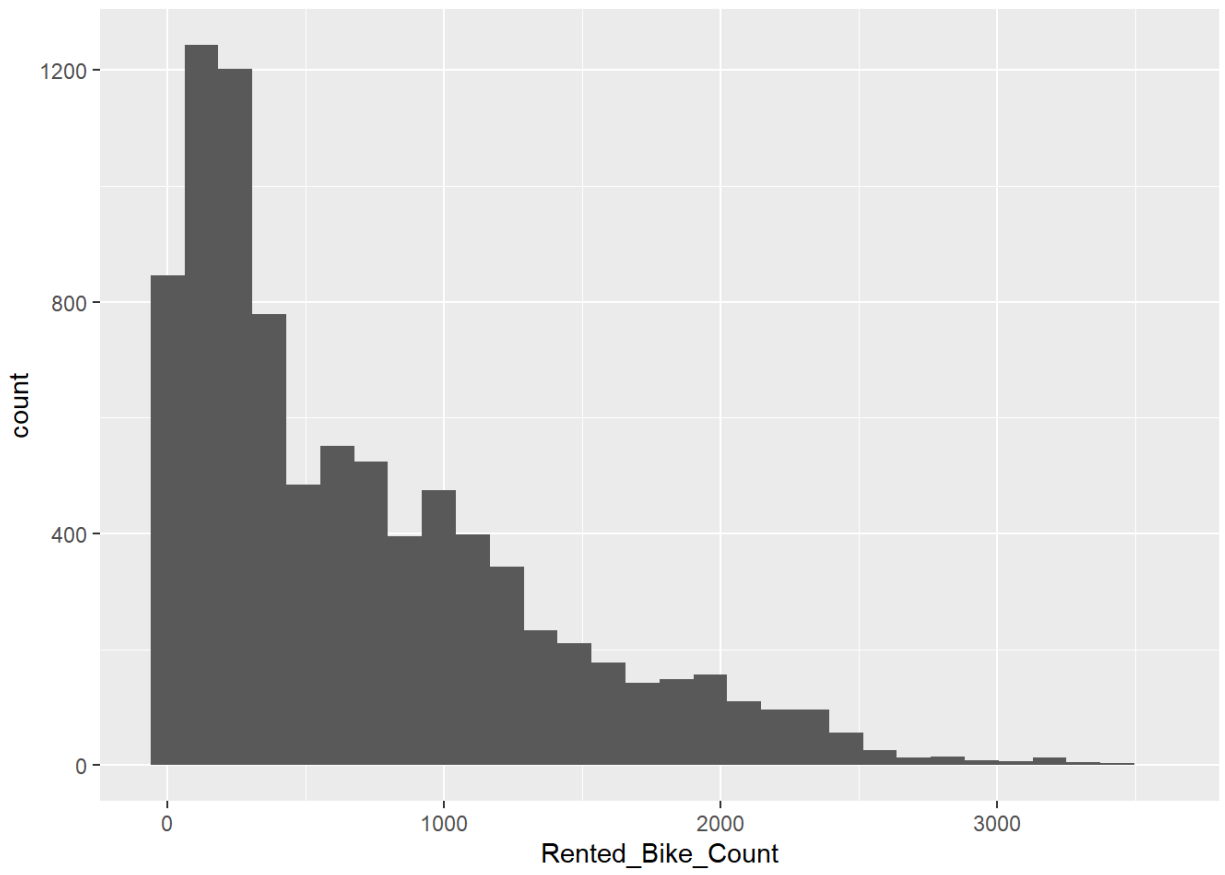
```
## # A tibble: 6 x 17
##   Date  Rented_Bike_Count  Hour Temperature Humidity Wind_Speed Visibility
##   <chr>             <dbl> <dbl>       <dbl>    <dbl>      <dbl>      <dbl>
## 1 42747               254     0        -5.2       37        2.2       2000
## 2 42747               204     1        -5.5       38        0.8       2000
## 3 42747               173     2        -6         39        1         2000
## 4 42747               107     3        -6.2       40        0.9       2000
## 5 42747                78     4        -6         36        2.3       2000
## 6 42747               100     5        -6.4       37        1.5       2000
## # ... with 10 more variables: Dew_Point_Temp <dbl>, Solar_Radiation <dbl>,
## #   Rainfall <dbl>, Snowfall <dbl>, Seasons <chr>, Holiday <chr>,
## #   Functioning_Days <chr>, Seasons_Numeric <dbl>, Holiday_Numeric <dbl>,
## #   Functioning_Days_Numeric <dbl>
```

```
dfBikeQuant <- dfBikeClean[-c(1,12:14)]
head(dfBikeQuant)
```

```
## # A tibble: 6 x 13
##   Rented_Bike_Count  Hour Temperature Humidity Wind_Speed Visibility
##               <dbl> <dbl>       <dbl>    <dbl>      <dbl>      <dbl>
## 1               254     0        -5.2       37        2.2       2000
## 2               204     1        -5.5       38        0.8       2000
## 3               173     2        -6         39        1         2000
## 4               107     3        -6.2       40        0.9       2000
## 5                78     4        -6         36        2.3       2000
## 6               100     5        -6.4       37        1.5       2000
## # ... with 7 more variables: Dew_Point_Temp <dbl>, Solar_Radiation <dbl>,
## #   Rainfall <dbl>, Snowfall <dbl>, Seasons_Numeric <dbl>,
## #   Holiday_Numeric <dbl>, Functioning_Days_Numeric <dbl>
```

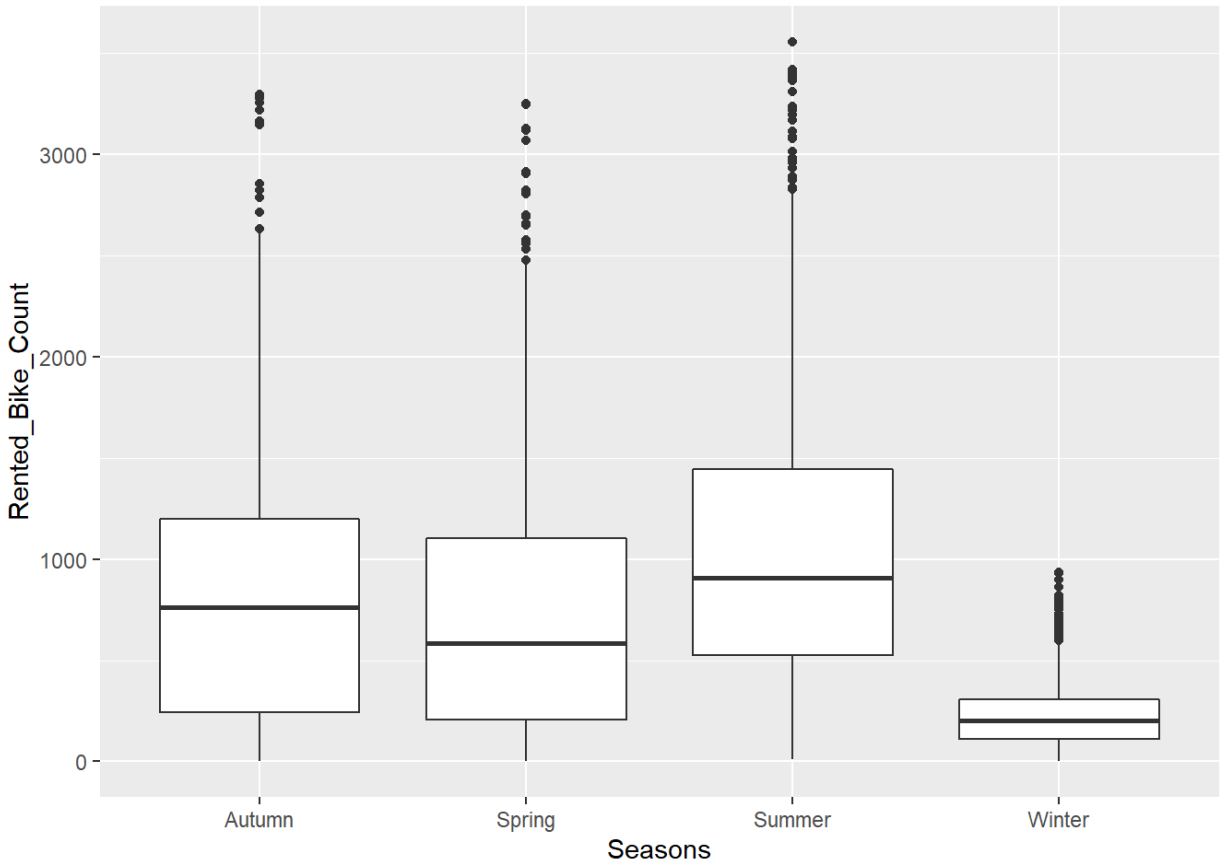
Exploratory Data Analysis

```
ggplot(dfBikeClean, aes(x = Rented_Bike_Count)) + geom_histogram(bins = 30)
```



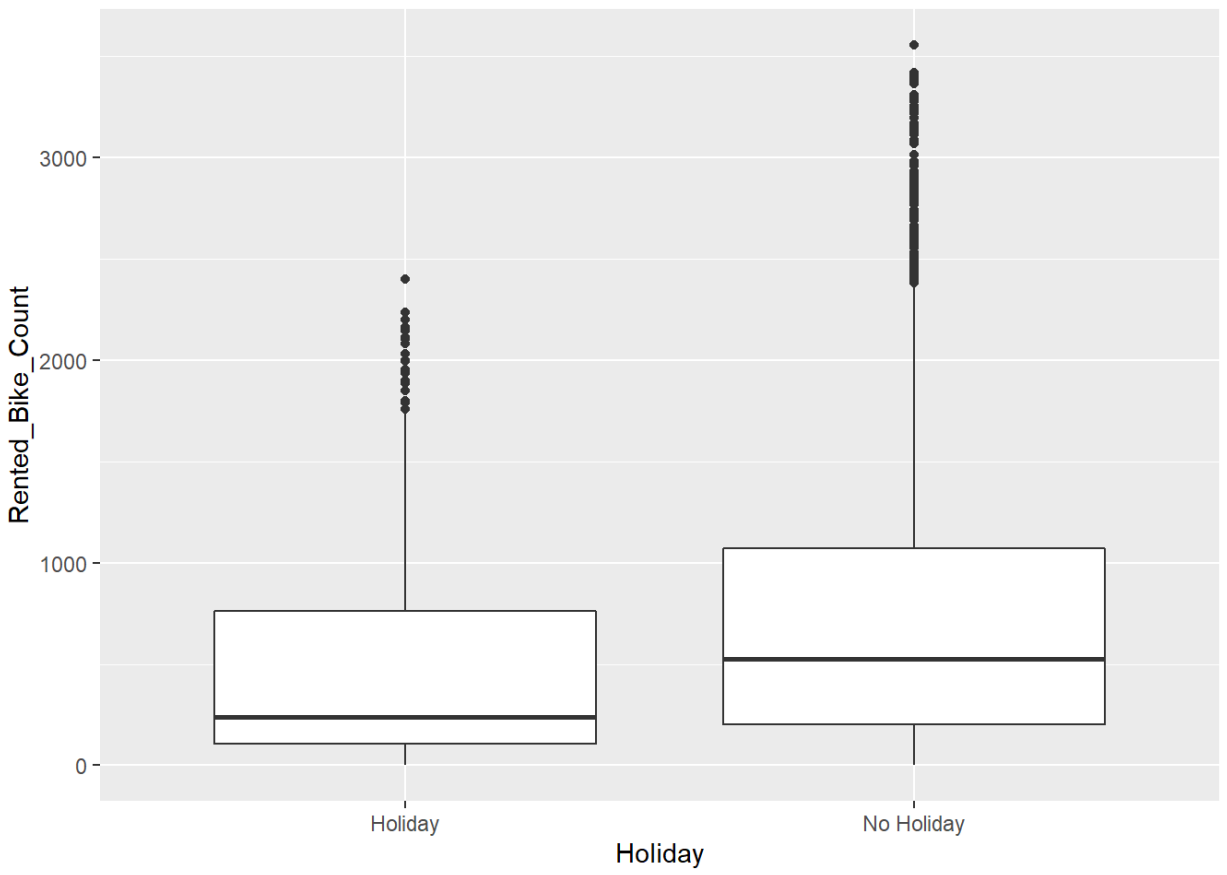
The basic histogram for daily bike rentals show an F distribution. However, this graph does not tell a lot of information on the trends for bike rentals.

```
ggplot(dfBikeClean, aes(x = Seasons, y = Rented_Bike_Count)) + geom_boxplot()
```



Exploring a little deeper, the amount of daily bike rentals can be categorized by the season. From this graph, we find that the amount of average amount of daily bike rentals during winter is drastically smaller than the other three seasons.

```
ggplot(dfBikeClean, aes(x = Holiday, y = Rented_Bike_Count)) + geom_boxplot()
```



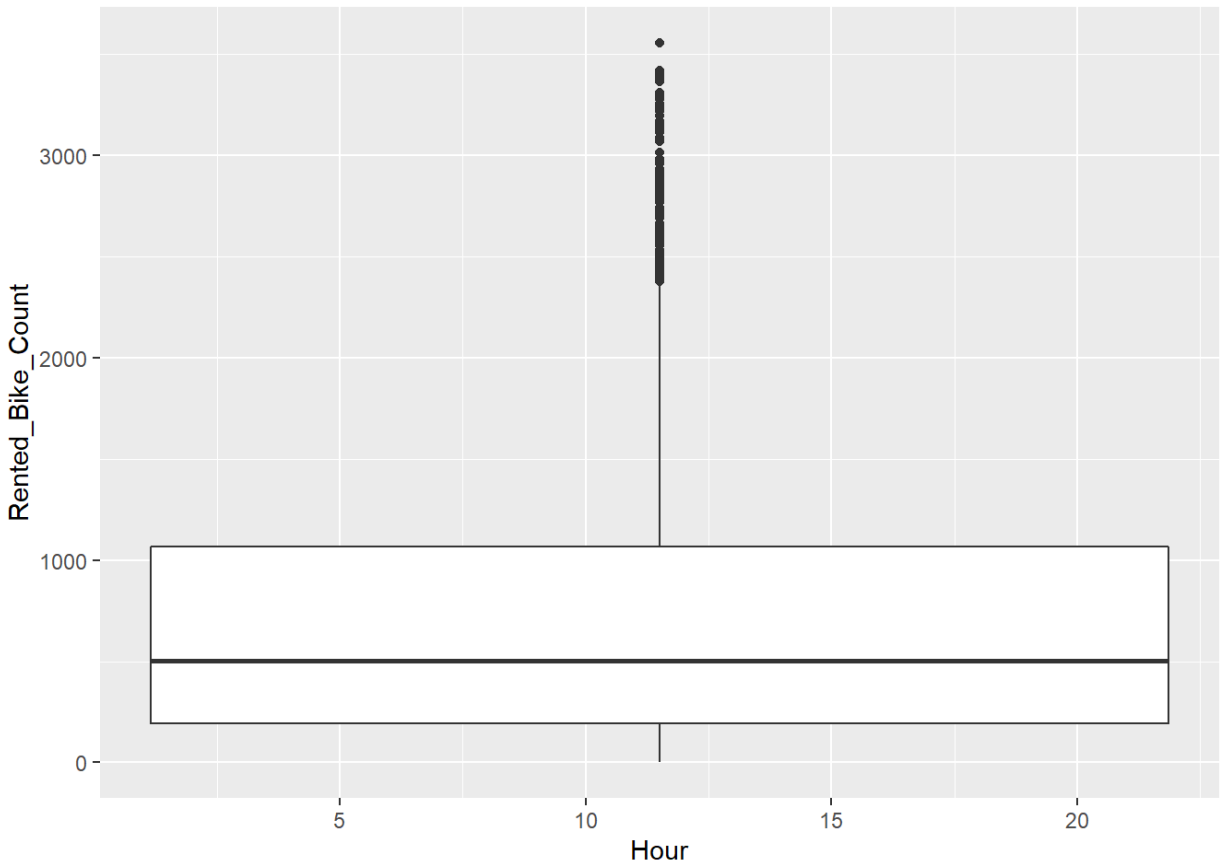
There is also an effect on the average daily bike rentals from whether the day of rental was a holiday or not. We can see that the average is higher on a non-holiday, and it has a higher range as well.

This is a little contradictory to what one would think of at first. The basic reasoning that “more bike rentals would occur on a holiday due to individuals having more time” is nullified by this graph.

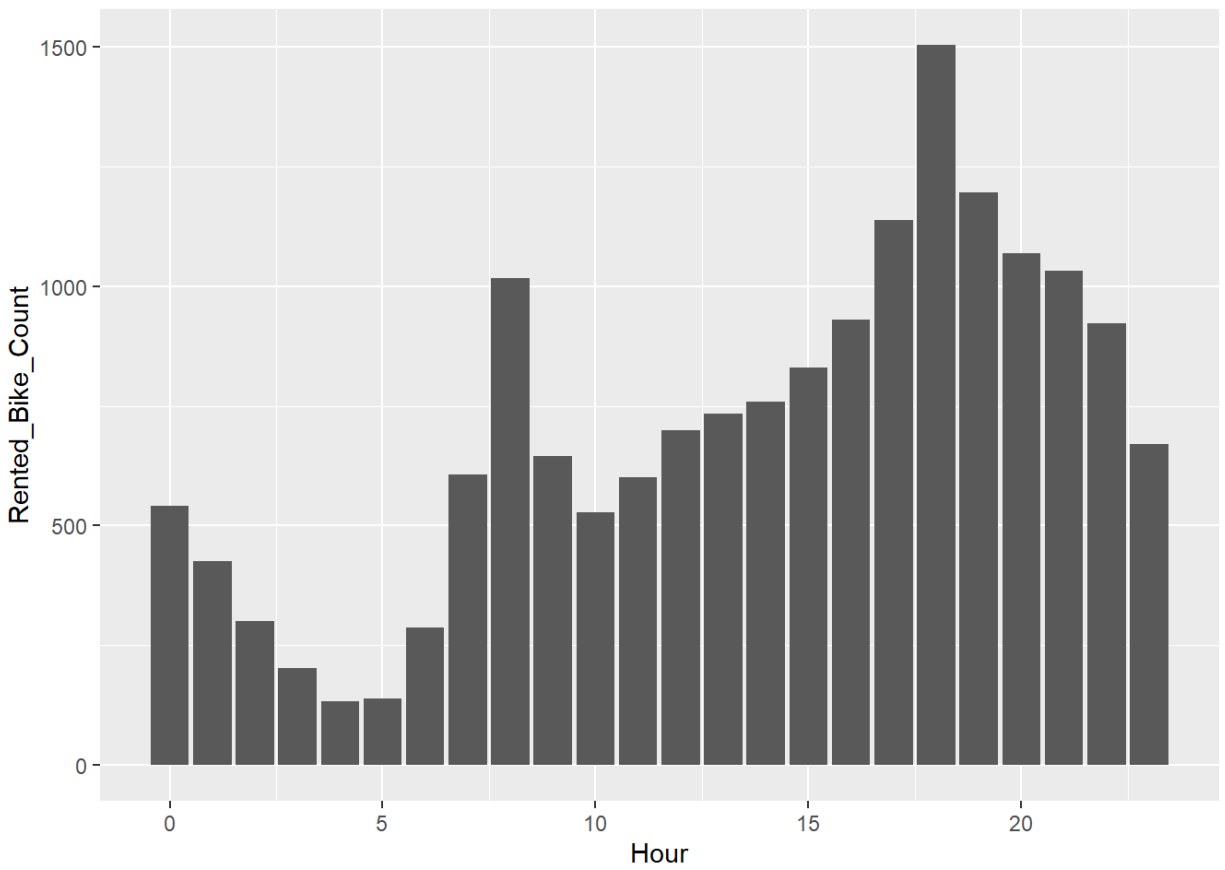
There are a few possible explanations for this: 1. Bike rentals are more comprised of work commuters 2. People prefer to spend their holiday on other activities

```
ggplot(dfBikeClean, aes(x = Hour, y = Rented_Bike_Count)) + geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```




```
ggplot(dfBikeClean, aes(Hour, Rented_Bike_Count)) +
geom_bar(position = "dodge",
           stat = "summary",
           fun = "mean")
```



The chart above show the average bike rental for each hour of the day. This graph tells us that a large portion of the bike rentals happen during the evening and late night than compared to the morning. Interestingly, there are two periods where bike rentals have particularly large spikes, at hour 8 (8 am) and hour 18 (6 pm). This corroborates with a previous conjecture that a high amount of rental traffic comes from work commuters.

Model

Basic Model

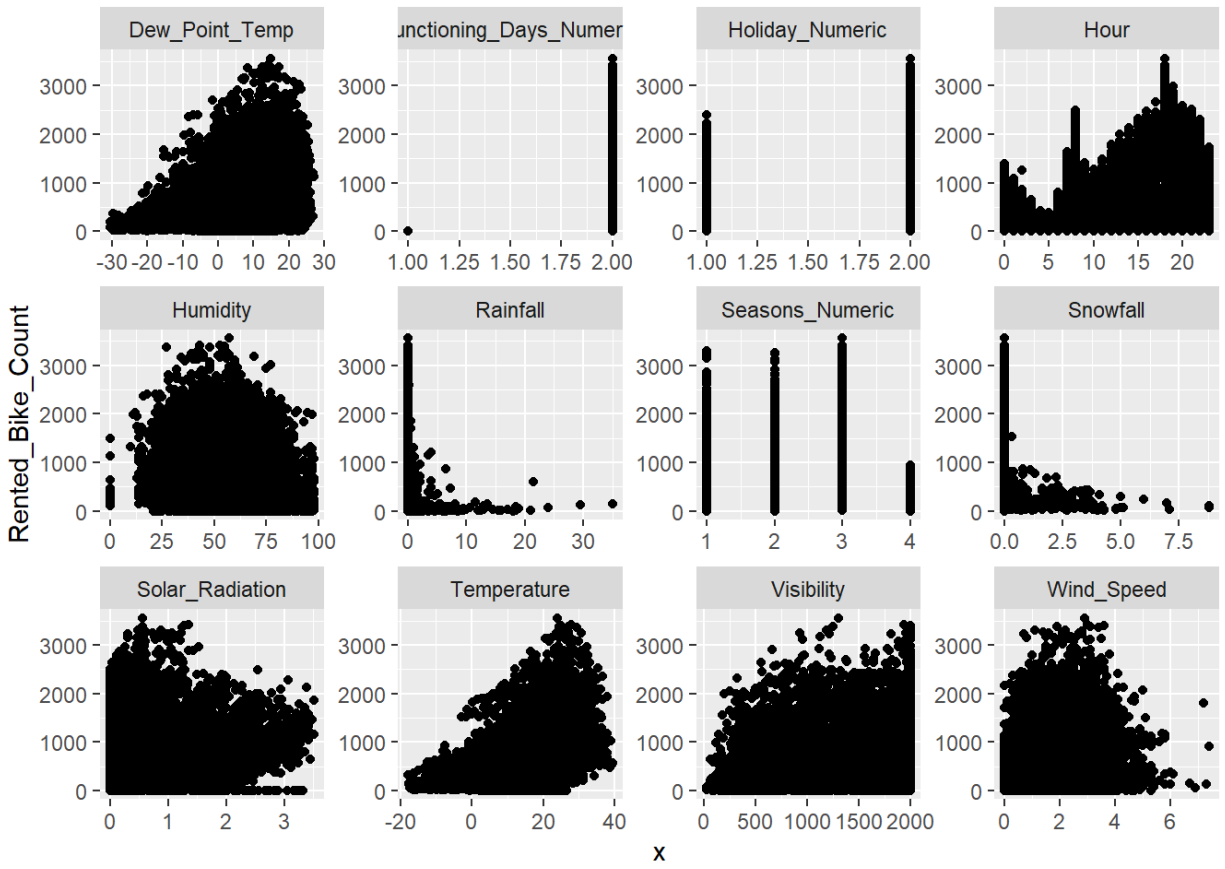
```
head(dfBikeQuant)
```

```
## # A tibble: 6 x 13
##   Rented_Bike_Count  Hour Temperature Humidity Wind_Speed Visibility
##               <dbl> <dbl>       <dbl>    <dbl>      <dbl>      <dbl>
## 1               254     0        -5.2       37        2.2       2000
## 2               204     1        -5.5       38        0.8       2000
## 3               173     2        -6         39        1         2000
## 4               107     3        -6.2       40        0.9       2000
## 5                78     4        -6         36        2.3       2000
## 6               100     5        -6.4       37        1.5       2000
## # ... with 7 more variables: Dew_Point_Temp <dbl>, Solar_Radiation <dbl>,
## #   Rainfall <dbl>, Snowfall <dbl>, Seasons_Numeric <dbl>,
## #   Holiday_Numeric <dbl>, Functioning_Days_Numeric <dbl>
```

In this section, we will be looking to develop a multiple regression model that will help quantify the effects the independent variables have on bike rental count.

We begin be building a basic model using all the variables and evaluate the performance

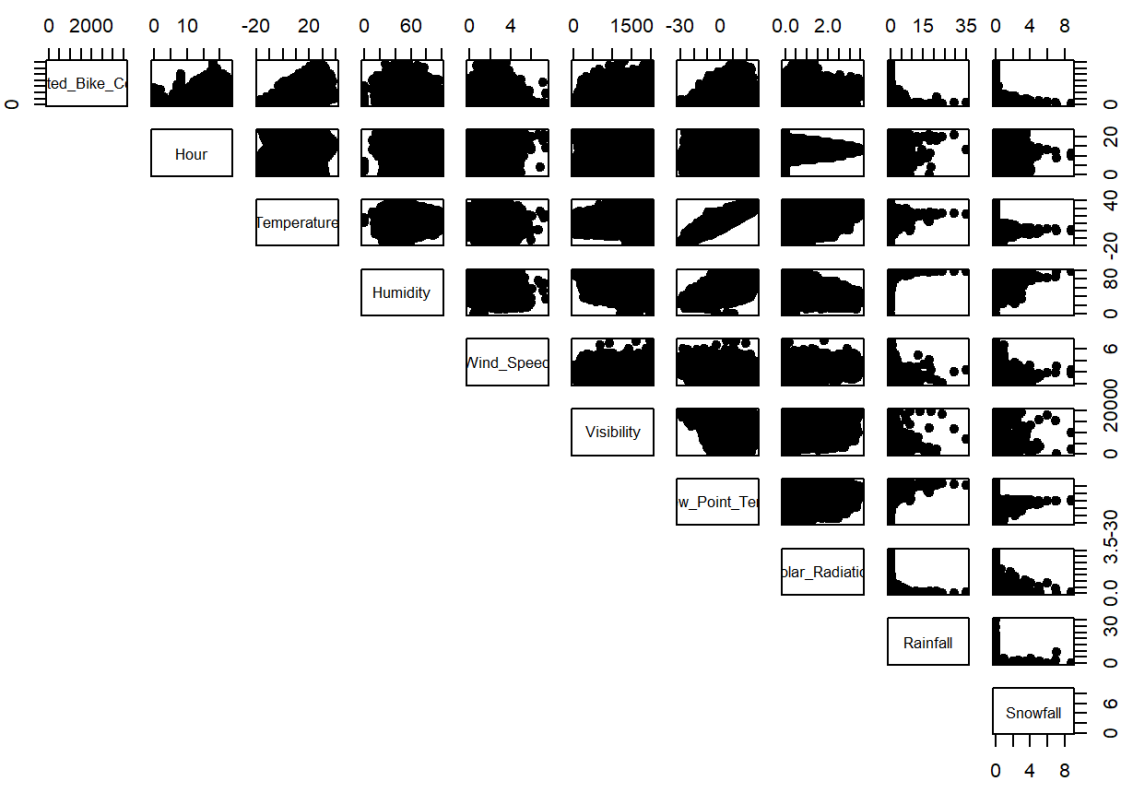
```
dfBikeQuant %>%
pivot_longer(
-Rented_Bike_Count,
names_to="xname", values_to="x"
) %>%
ggplot(aes(x = x, y = Rented_Bike_Count)) + geom_point() +
facet_wrap(~xname, scales = "free") -> g
g
```




```
basicModel <- lm(Rented_Bike_Count ~
           Hour +
           Temperature +
           Humidity +
           Wind_Speed +
           Visibility +
           Dew_Point_Temp +
           Solar_Radiation +
           Rainfall +
           Snowfall + factor(Seasons_Numeric) +
           factor(Holiday_Numeric) +
           factor(Functioning_Days_Numeric),
          data = dfBikeQuant)
```

We have an model p value of < 2.2e-16, which is about zero. The explanatory variables in the model is significant overall. The R-squared, defined as how much of the varaince is explained by the model, is 0.5504. This means that just over half of the variance can be explained by this model. This is a decent result for a basic model, but can it be further improved by a model selection method such as backwards elimination.

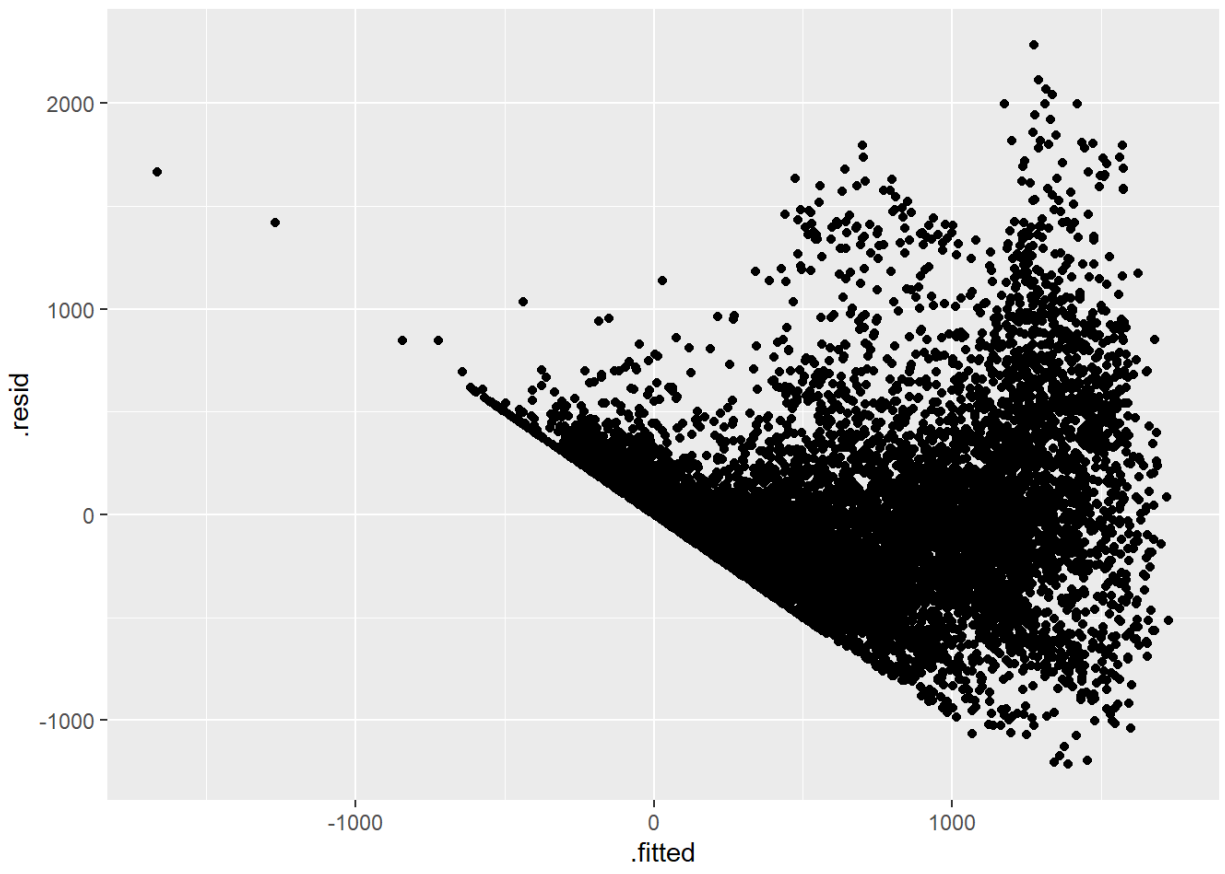
```
pairs(dfBikeQuant[,1:10], pch = 19, lower.panel=NULL)
```



The figure above shows the all of the independent variables plotted against each other, except for the categorical varialbes. These graphs help visualize the correlation between each variable.

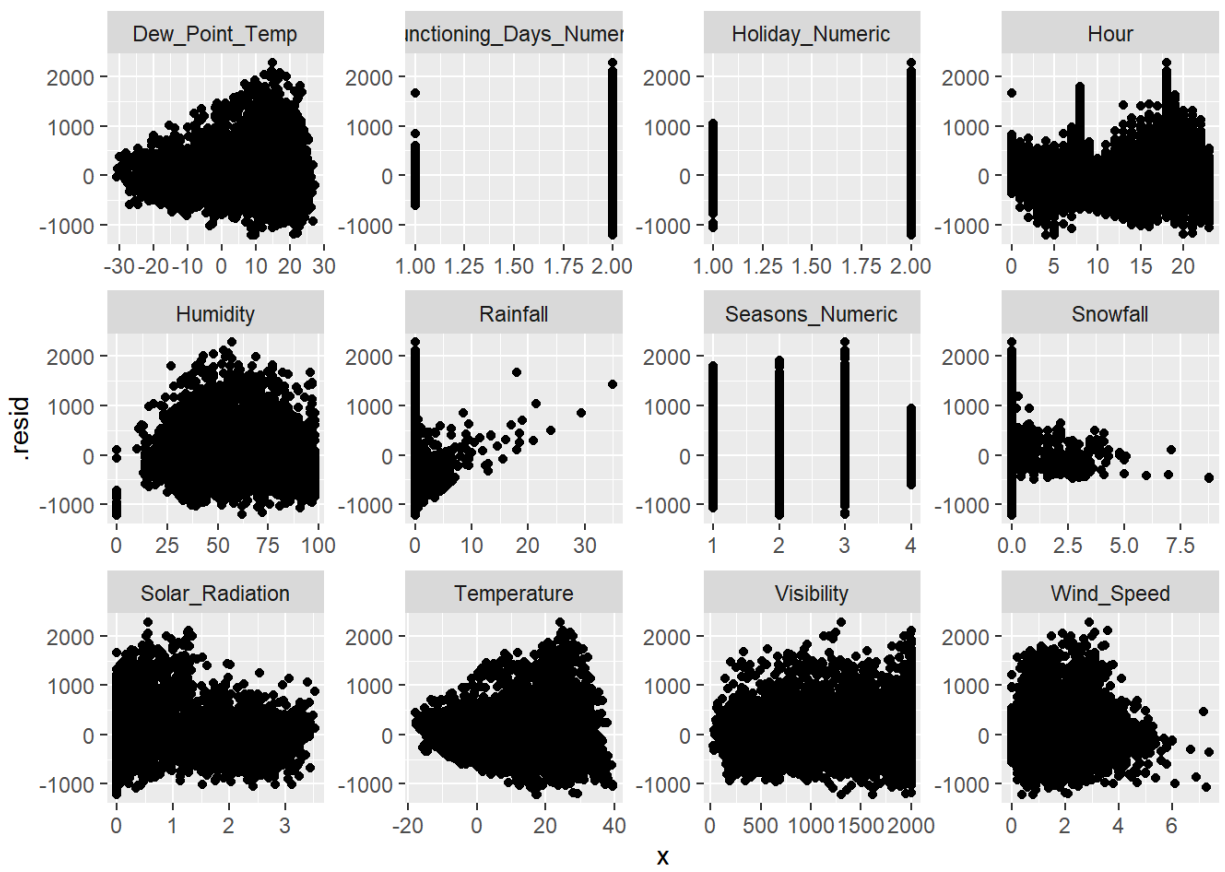
There is a wide range of correlations. Furthermore, there seems to be a reasonably linear relationship between Temperature and Dew_Point_Temperature. This correlation between variables is a good starting point for investigating multicollinearity.

```
ggplot(basicModel, aes(y = .resid, x = .fitted)) + geom_point()
```



Above is the residuals vs. fitted graph. From what we can see, the residuals do not have the good scattered distribution that we would like to see from this graph. There is a noticeable cut-off on the bottom half of the graph.

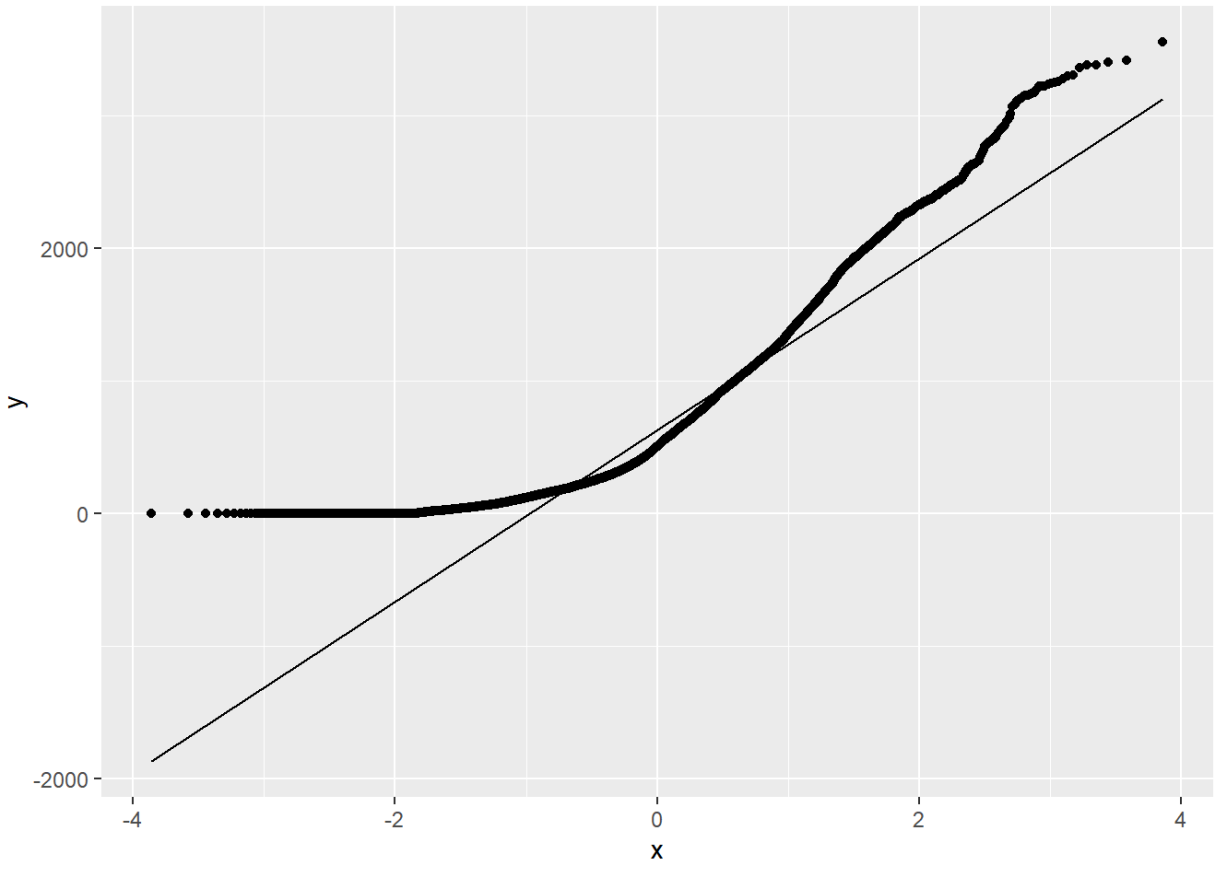
```
basicModel %>% augment(dfBikeQuant) -> basicModel.a
basicModel.a %>%
  pivot_longer(
    c(Hour:Functioning_Days_Numeric),
    names_to="xname", values_to="x"
  ) %>%
  ggplot(aes(x = x, y = .resid)) +
  geom_point() + facet_wrap(~xname, scales = "free") -> g
g
```



The individual residuals vs each explanatory variable charts do not show and significant trends. The distributions, outside of the categorical variables, are better scattered than the resid vs fitted plot, except for snowfall and rainfall.

The odd distribution of snowfall and rainfall can be explained by the heavy geometric distribution of the varaibles.

```
ggplot(basicModel,aes(sample=Rented_Bike_Count))+stat_qq()+stat_qq_line()
```



The normal quantile plot of residuals has some glaring problems. The lower half of the plot is extremely high from the normal line. It even observes a flat slope in the beginning. The upper tail of the plot is also high above the normal. Overall, there is an upward curve of the plot.

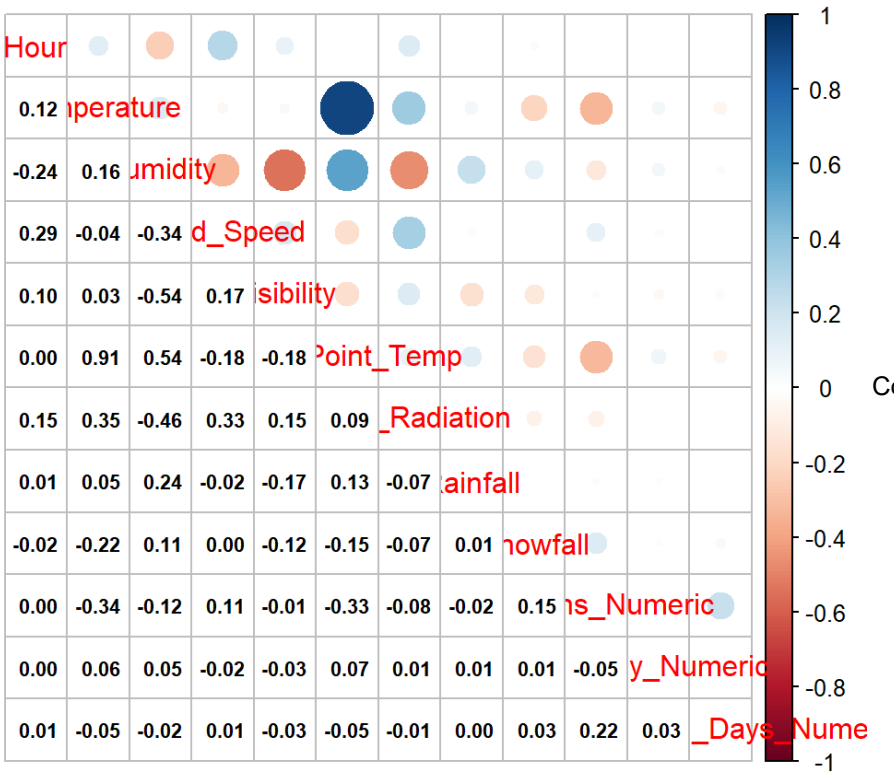
Interaction between variables

There may be some relationship between the independent variables that if incorporated into the model, can help improve the accuracy of our model.

```
bikes_VIF = vif(basicModel)
```

Clearly (VIF(temperature) = 89.477069, VIF(Humidity) = 20.553911, VIF(Dew_Point_Temperature) = 117.298694) > 10, suggesting early multicollinearity somewhere in the data. We can verify this with the correlation plot

```
corrplot.mixed(cor(dfBikeQuant[,2:13]), lower.col = 'black', number.cex = .7)
```



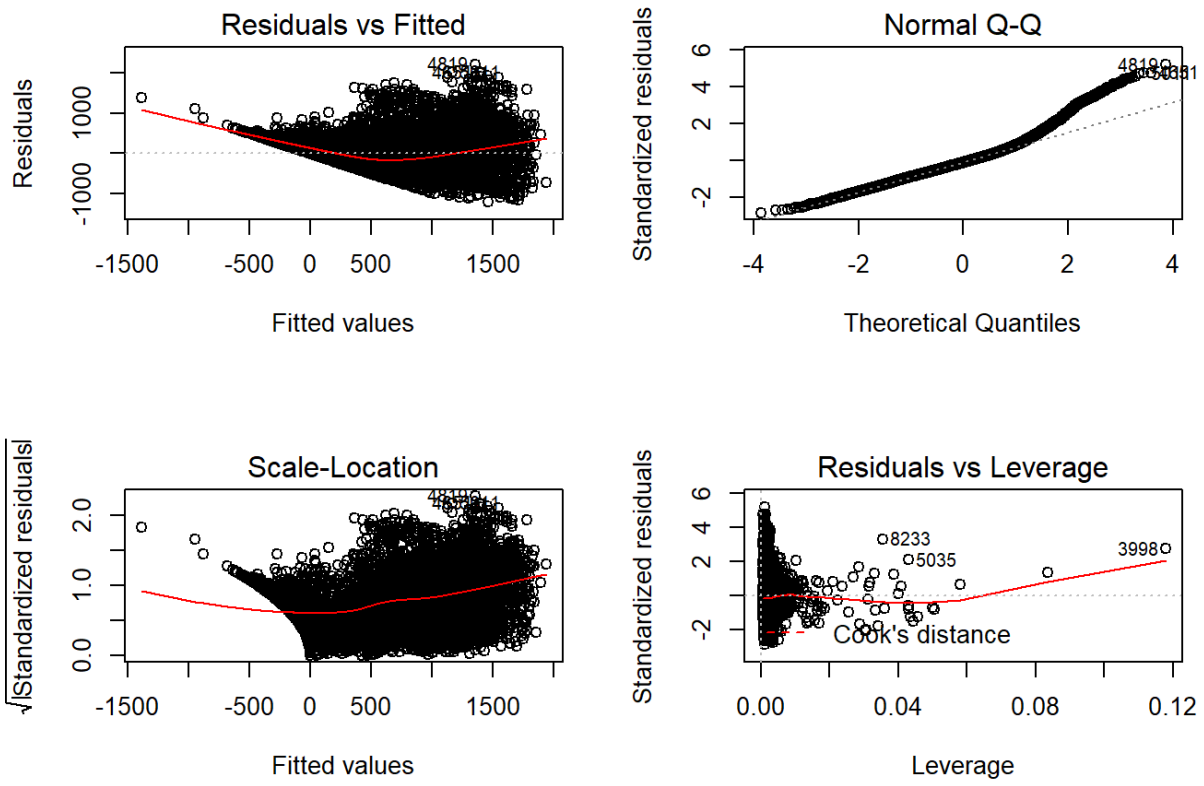
Correlation between temperature and

Dew_point_temp is 0.91 Correlation between humidity and dew_point_temp is 0.54 Correlation between Humidity and visibility is -0.54 Correlation between humidity and radiation is -0.46

We will add in an interaction model for each of these pairs.

```
InteractionModel <- lm(Rented_Bike_Count ~
           Hour +
           Temperature +
           Humidity +
           Wind_Speed +
           Visibility +
           Dew_Point_Temp +
           Solar_Radiation +
           Rainfall +
           Snowfall +
           factor(Seasons_Numeric) +
           factor(Holiday_Numeric) +
           factor(Functioning_Days_Numeric) +
           Humidity:Visibility +
           Humidity:Dew_Point_Temp +
           Humidity:Solar_Radiation +
           Temperature:Dew_Point_Temp,
          data = dfBikeQuant)
```

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



The normal QQ plot has been significantly improved. The high values at the lower end of the tail has been reduced, and now exhibits more normal behaviour. The upper end of the graph still exhibits high values.

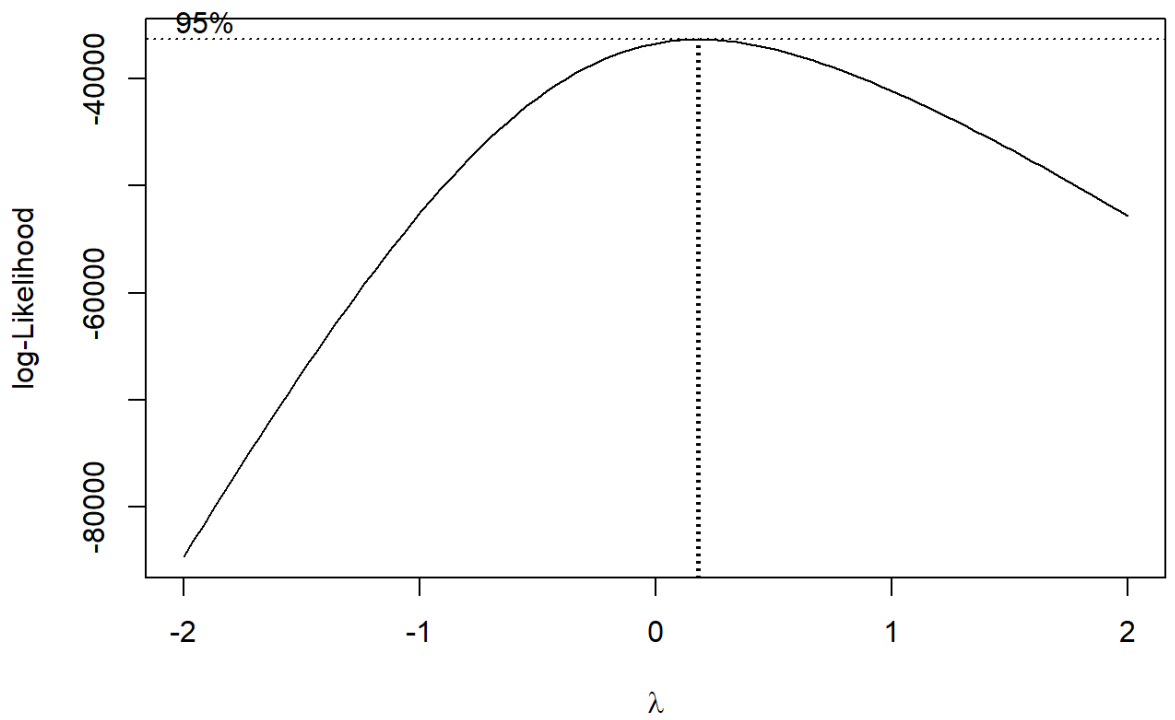
The residuals plot now has a wider spread, and a slight “trumpet” shape.

The R^2 increased from 0.5504 to 0.5712. The interaction variables helped explain more of the variances within our data, increasing the accuracy of the basic model.

We can further try to transform the dependent variables into a normal shape through box-cox.

```
bike_fit <- lm(formula = Rented_Bike_Count+1 ~
                 Hour +
                 Temperature +
                 Humidity +
                 Wind_Speed +
                 Dew_Point_Temp +
                 Solar_Radiation +
                 Rainfall +
                 factor(Seasons_Numeric) +
                 factor(Holiday_Numeric) +
                 factor(Functioning_Days_Numeric) +
                 Humidity:Visibility +
                 Humidity:Dew_Point_Temp +
                 Humidity:Solar_Radiation +
                 Temperature:Dew_Point_Temp,
               data = dfBikeQuant)

bike_boxcox <- boxcox(bike_fit)
```

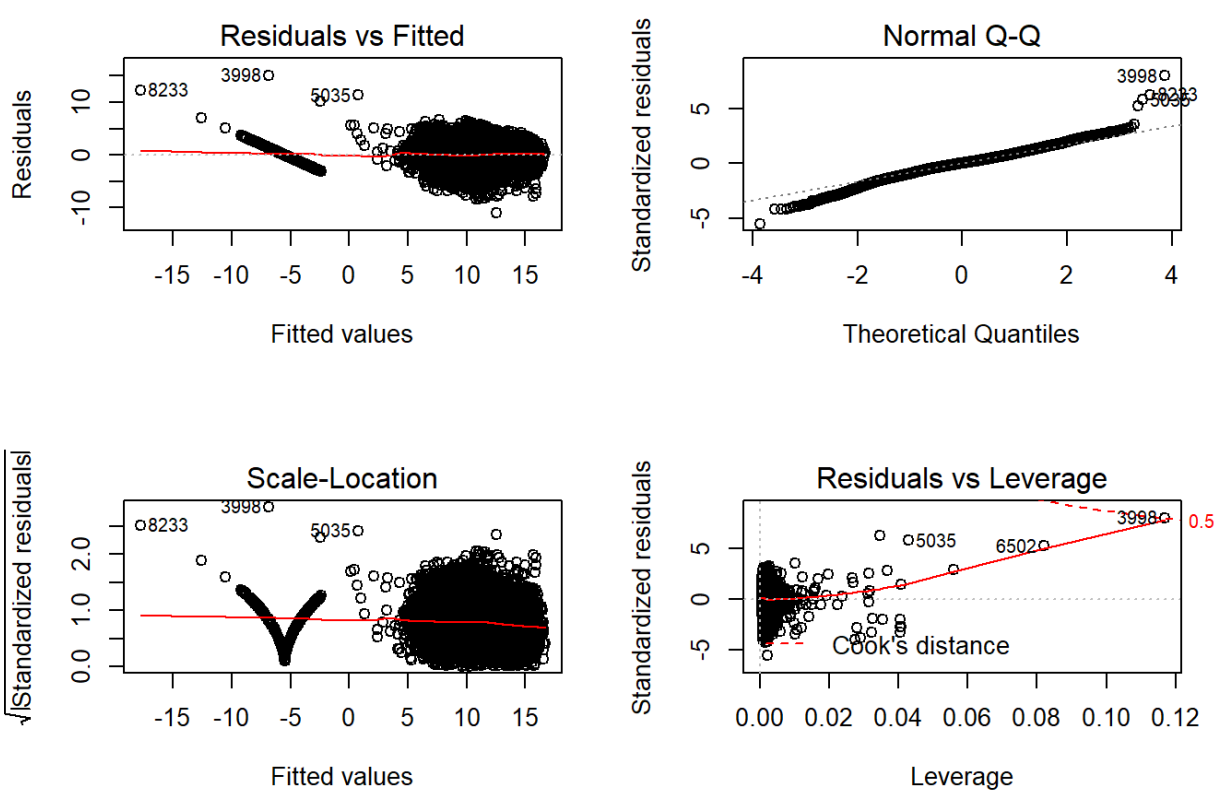



```
lambda_boxcox_bike <- bike_boxcox$x[which.max(bike_boxcox$y)]
```

The transformation that maximizes likelihood is 0.1818182. We will refit the model with the transformed model:

```
boxCoxModel <- lm((((dfBikeQuant$Rented_Bike_Count^lambda_boxcox_bike)-1)/lambda_boxcox_bike) ~
                  Hour +
                  Temperature +
                  Humidity +
                  Wind_Speed +
                  Dew_Point_Temp +
                  Solar_Radiation +
                  Rainfall +
                  factor(Seasons_Numeric) +
                  factor(Holiday_Numeric) +
                  factor(Functioning_Days_Numeric) +
                  Humidity:Visibility +
                  Humidity:Dew_Point_Temp +
                  Humidity:Solar_Radiation +
                  Temperature:Dew_Point_Temp,
                data = dfBikeQuant)
```

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



The box-cox transformation has significantly improved the R^2 values compared to the Cook's distance model. The R^2 improved from 0.5712 to 0.8035. The adjusted R^2 has improved from 0.5703 to 0.8031/0.6881.

The transformation has changed the Normal QQ plot. it has smoothed out the plot, showing more normal, but there are still a few outlier points on the upper tail.

The residual plots are heavily influenced by the transformation.

Backwards Elimination

We can perform a backwards elimination technique to remove the variables that have a statistically insignificant impact on the prediction results.

```
tidy(boxCoxModel) %>% arrange(p.value)
```

```
## # A tibble: 17 x 5
##    term                              estimate std.error statistic   p.value
##    <chr>                                <dbl>     <dbl>     <dbl>     <dbl>
##  1 factor(Functioning_Days_Numeric)2  1.82e+1   1.23e-1    149.   0
##  2 Hour                               1.26e-1   3.40e-3     37.0  4.09e-278
##  3 Rainfall                          -4.81e-1   2.00e-2    -24.0  1.20e-123
##  4 factor(Seasons_Numeric)4          -1.86e+0   9.63e-2    -19.3  1.56e- 81
##  5 factor(Seasons_Numeric)2          -8.75e-1   6.35e-2    -13.8  1.04e- 42
##  6 Humidity                          -7.18e-2   5.44e-3    -13.2  1.87e- 39
##  7 Temperature:Dew_Point_Temp        -2.76e-3   2.10e-4    -13.2  2.75e- 39
##  8 (Intercept)                       -5.05e+0   5.06e-1     -9.98 2.42e- 23
##  9 factor(Holiday_Numeric)2           9.75e-1   9.95e-2      9.79 1.55e- 22
## 10 Solar_Radiation                   -8.72e-1   1.03e-1     -8.47 2.86e- 17
## 11 Humidity:Solar_Radiation           1.73e-2   2.13e-3      8.14 4.55e- 16
## 12 Humidity:Visibility                4.56e-6   6.60e-7      6.90 5.50e- 12
## 13 Humidity:Dew_Point_Temp           -6.87e-4   1.03e-4     -6.65 3.16e- 11
## 14 Dew_Point_Temp                     1.35e-1   2.14e-2      6.32 2.79e- 10
## 15 Temperature                        6.43e-2   2.07e-2      3.11 1.89e-  3
## 16 factor(Seasons_Numeric)3          -1.76e-1   8.95e-2     -1.96 4.95e-  2
## 17 Wind_Speed                         4.52e-2   2.36e-2      1.91 5.57e-  2
```

We perform backwards by removing the variables one by one and refitting the model until eventually ending up with a set of variables that are all important to the prediction results.

We can perform this process manually by removing the variable with the highest p-value that is larger than the level of significance, refit the model without this variable, and repeat.

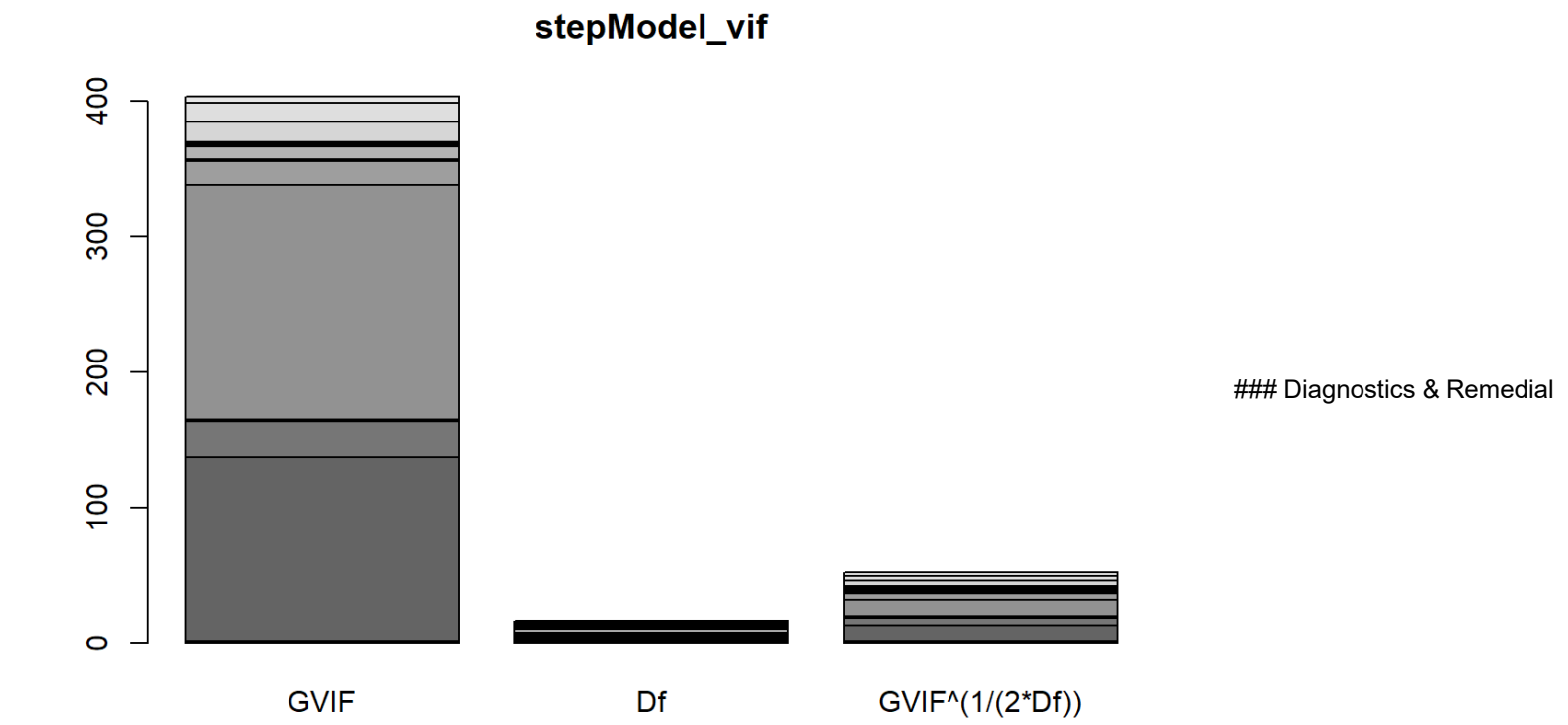
The step() function can automatically perform this process.

```
stepModel <- step(boxCoxModel, direction = "backward", test="F")
```

```
## Start:  AIC=12077.32
## (((dfBikeQuant$Rented_Bike_Count^lambda_boxcox_bike) - 1)/lambda_boxcox_bike) ~
##     Hour + Temperature + Humidity + Wind_Speed + Dew_Point_Temp +
##         Solar_Radiation + Rainfall + factor(Seasons_Numeric) +
##         factor(Holiday_Numeric) + factor(Functioning_Days_Numeric) +
##         Humidity:Visibility + Humidity:Dew_Point_Temp + Humidity:Solar_Radiation +
##         Temperature:Dew_Point_Temp
##
##                                   Df Sum of Sq    RSS   AIC    F value
## <none>                                        34640 12077
## - Wind_Speed                       1        15  34654 12079     3.6616
## - Humidity:Dew_Point_Temp          1       175  34815 12120    44.1865
## - Humidity:Visibility              1       189  34829 12123    47.6323
## - Humidity:Solar_Radiation         1       262  34902 12141    66.2372
## - factor(Holiday_Numeric)          1       380  35020 12171    95.9384
## - Temperature:Dew_Point_Temp       1       688  35328 12248   173.6793
## - factor(Seasons_Numeric)          3      1657  36297 12481   139.3825
## - Rainfall                         1      2288  36928 12636   577.6044
## - Hour                             1      5415  40055 13348  1366.7468
## - factor(Functioning_Days_Numeric)  1     87750 122390 23132 22147.8132
##                                     Pr(>F)
## <none>
## - Wind_Speed                         0.05571 .
## - Humidity:Dew_Point_Temp         3.164e-11 ***
## - Humidity:Visibility             5.500e-12 ***
## - Humidity:Solar_Radiation        4.546e-16 ***
## - factor(Holiday_Numeric)         < 2.2e-16 ***
## - Temperature:Dew_Point_Temp      < 2.2e-16 ***
## - factor(Seasons_Numeric)         < 2.2e-16 ***
## - Rainfall                        < 2.2e-16 ***
## - Hour                            < 2.2e-16 ***
## - factor(Functioning_Days_Numeric) < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

No variables were removed from backwards elimination as they are all statistically significant.

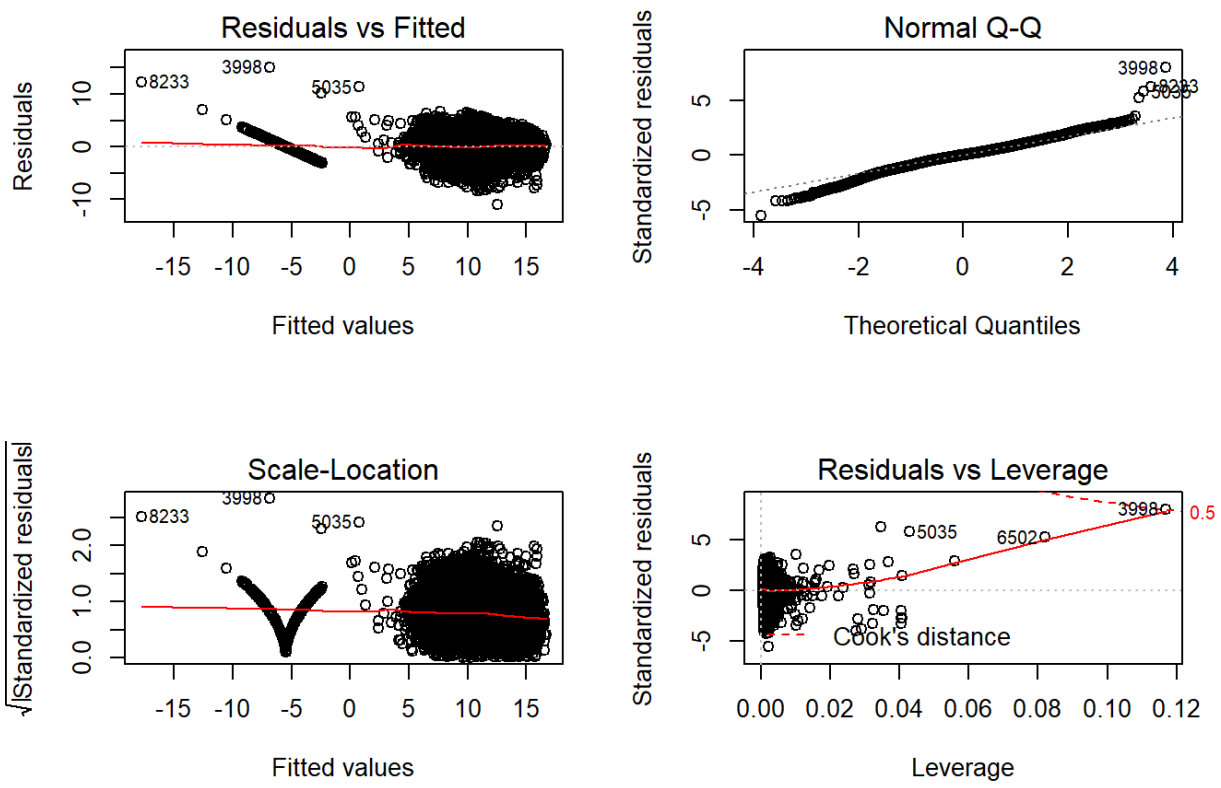
```
stepModel_vif <- vif(stepModel)
barplot(stepModel_vif, main = "stepModel_vif")
```



Diagnostics & Remedial

Measures In this section, we will perform a few diagnostics and remedial measures to help further improve out model.

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

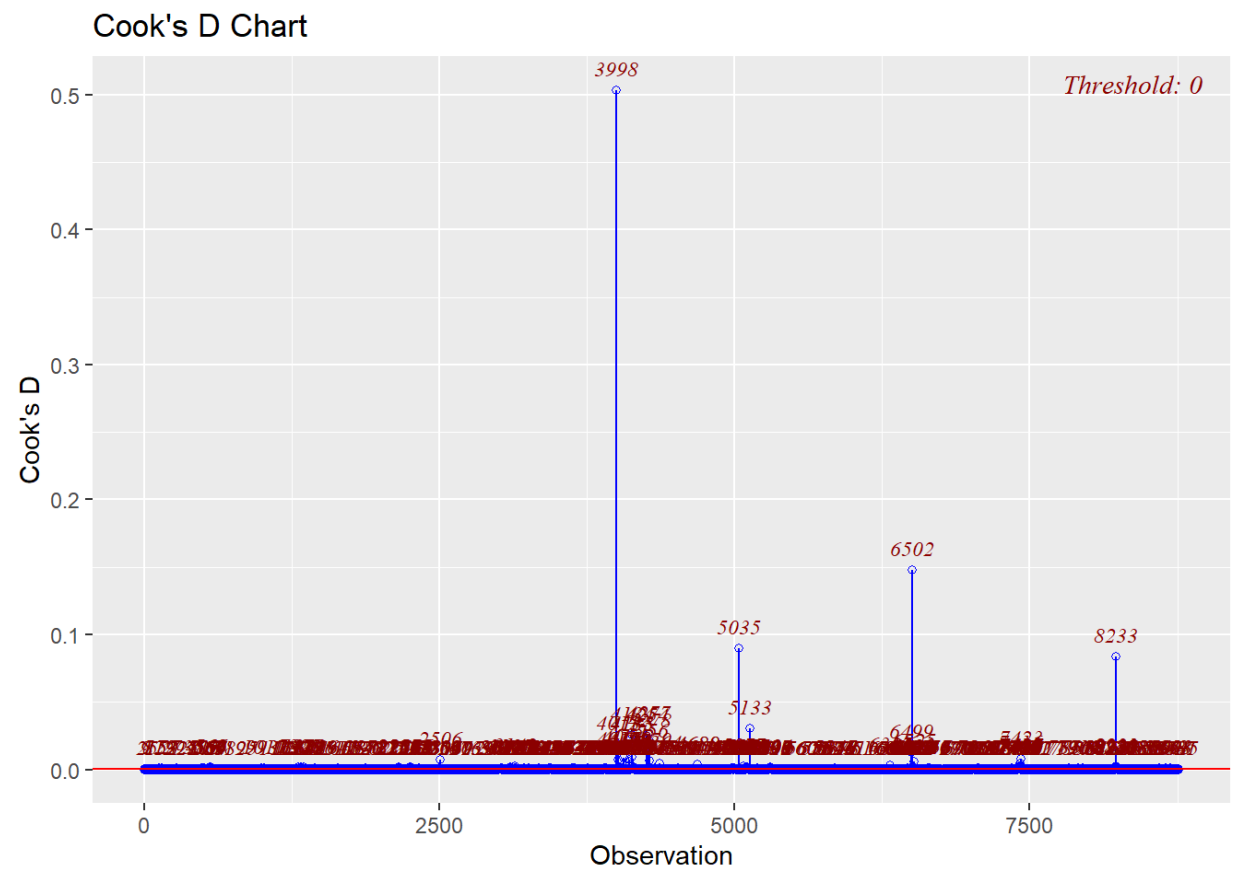


The charts above are the diagnostic charts of the completed backward elimination model.

They are the same as the model after box-cox transformation as no variables were removed during backwards elimination.

Another remedial measure we can perform is a Cook’s Distance evaluation to remove any significant outliers within the data.

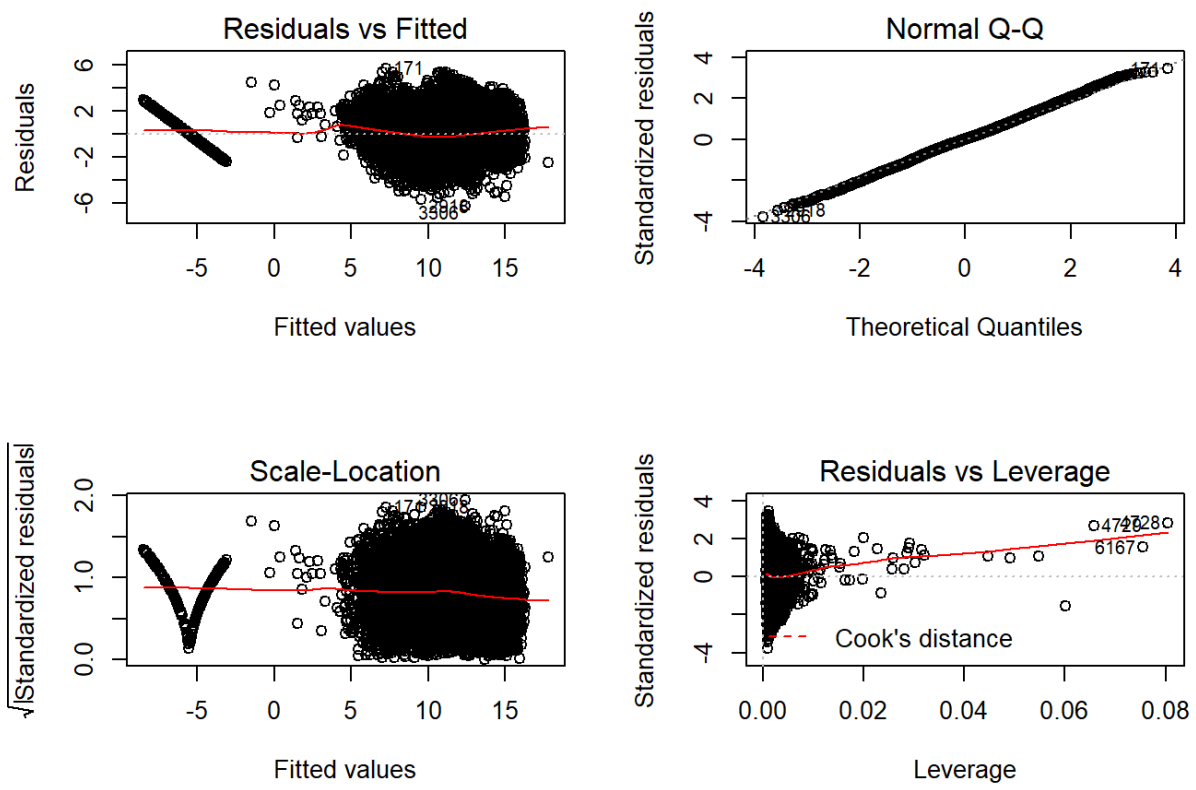
```
ols_plot_cooksd_chart(stepModel)
```



```
cooksd <- cooks.distance(stepModel)
influential <- as.numeric(names(cooksd)[(cooksd > (4/(nrow(dfBikeQuant))))])
dfBikeQuant1 <- dfBikeQuant[-influential, ]
```

```
CookModel <- lm((((dfBikeQuant1$Rented_Bike_Count^lambda_boxcox_bike)-1)/lambda_boxcox_bike) ~
                  Hour +
                  Temperature +
                  Humidity +
                  Wind_Speed +
                  Dew_Point_Temp +
                  Solar_Radiation +
                  Rainfall +
                  factor(Seasons_Numeric) +
                  factor(Holiday_Numeric) +
                  factor(Functioning_Days_Numeric) +
                  Humidity:Visibility +
                  Humidity:Dew_Point_Temp +
                  Humidity:Solar_Radiation +
                  Temperature:Dew_Point_Temp,
                data = dfBikeQuant1)
```

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



The R^2 improved from 0.8035 to 0.8511 The adjusted R^2 has improved from 0.8031 to 0.8509. The above charts show the diagnostics of the model after removing outliers.

Observing the normal plot, the outliers at the upper tail of the plot were removed and the model now has close to normal behaviour.

The residuals plot is more spread out on the majority right side of the plot, however there is still the abnormal behaviour on the left side of the plot.

Discussion and Conclusion

From the analysis we have found: demand for bikes coincide with times of day before or after work, suggesting the use of bikes as a last mile utility over, say, walking or other forms of last mile mobility. … which we can deduce a profile of a frequent user of the bike rental service.

Extreme weather conditions such as rain or snow are conducive to predicting the use of bikes as a last mile as (refer to coef of snow, rain, respectively under summary). However, we can find the opposite reception pattern in other cities with winter weather. In a Finnish study of bike usage, the city planners had found that the maintenance of bike infrastructure like dedicated bike paths, maintenance of those paths and public roads that make it possible for routine, hassle-free biking [1]. It is, therefore, possible to deduce that the lack of winter hourly bike demand is a result of the poor maintenance of bike paths or lack of attention to the design of bike-friendly municipal infrastructure to maintain the program during the winter. Given the population density of Seoul, the year round use of bike infrastructure can aid in the reduction of overall pollution.

This analysis serves to promote improvements to Seoul's biking infrastructure by informing supply to hourly demand for availability of bikes where it is to be communicated to municipal operators where they can make informed decisions. We have found the optimal model for predicting hourly demand for bikes is the CooksModel following a boxcox transformation and the inclusion of interaction variables between (Humidity, Visibility), (Humidity, Dew_Point_Temp), (Humidity, Solar_Radiation), (Temperature, Dew_Point_Temp). Unfortunately, the absence of regional or locational data for hourly demand will limit the scope of this analysis such as proximity to public points of interest, transit stations, and local commercial density.

References

[1] Talvipyöräily Laajuus Motiivit Ja Esteet Terveysvaikutukset. PDF Ilmainen lataus. (n.d.). Retrieved April 10, 2022, from https://docplayer.fi/7216725-Talvipyoraily-laajuus-motiivit-ja-esteet-terveysvaikutukset.html (https://docplayer.fi/7216725-Talvipyoraily-laajuus-motiivit-ja-esteet-terveysvaikutukset.html)

[2] https://www.statista.com/statistics/1012470/south-korea-commonly-used-transportation/ (https://www.statista.com/statistics/1012470/south-korea-commonly-used-transportation/)

[3] SOUTH KOREA PUBLIC HOLIDAYS. (n.d.). Seoul Bike Sharing Demand Data Set. UCI Machine Learning Repository: Seoul Bike Sharing Demand Data Set. Retrieved April 7, 2022, from https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand (https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand)