Predicting Bike Share Ridership based on Weather Data in Seattle

Joey Rodriguez and Daniel Bhatti

2024-11-22

Introduction

Cycle share has launched in many U.S. cities since its introduction in Washington, D.C. in 2010 (1). One iteration of cycle share was Pronto! based in Seattle, Washington. From 2014 to 2017, 500 Pronto! bikes operated across 54 stations in Seattle's downtown. The City of Seattle, in partnership with Socratica, collected system data during the operating window and provided it to the public via its open data platform. Pronto fell short of the success realized by other bike schemes in the U.S. like Capitol Bikes, Philly's Indego, and NYC's CitiBike. Researchers have used system data to conduct a post-mortem on Pronto! as dockless bike share schemes filled the void (2). In this paper, we will investigate the relationship between weather in the service area and daily ridership. In particular, we want to predict daily ridership based on the weather data.

The data were downloaded from Kaggle (3). The file `trip.csv` contains data on each trip from 13 October 2014 to 31 March 2017, or 901 days. Each case in this dataset is a trip, and there were 275,091 trips over the 901 days. The relevant variables from the original 12 in this dataset are the response variables `start_time` and `trip_duration`. In the file `weather.csv.xls`, a single case corresponds to a single day. This file contains the weather data for each day from 13 October 2014 to 31 August 2016, or 689 days. That is, the dates covered by the weather data are a proper subset of the dates covered by the trip data. After merging these data and creating some of our own variables, we move on to exploratory data analysis for the response variables and predictors. This exploration informs our choice for our final model.

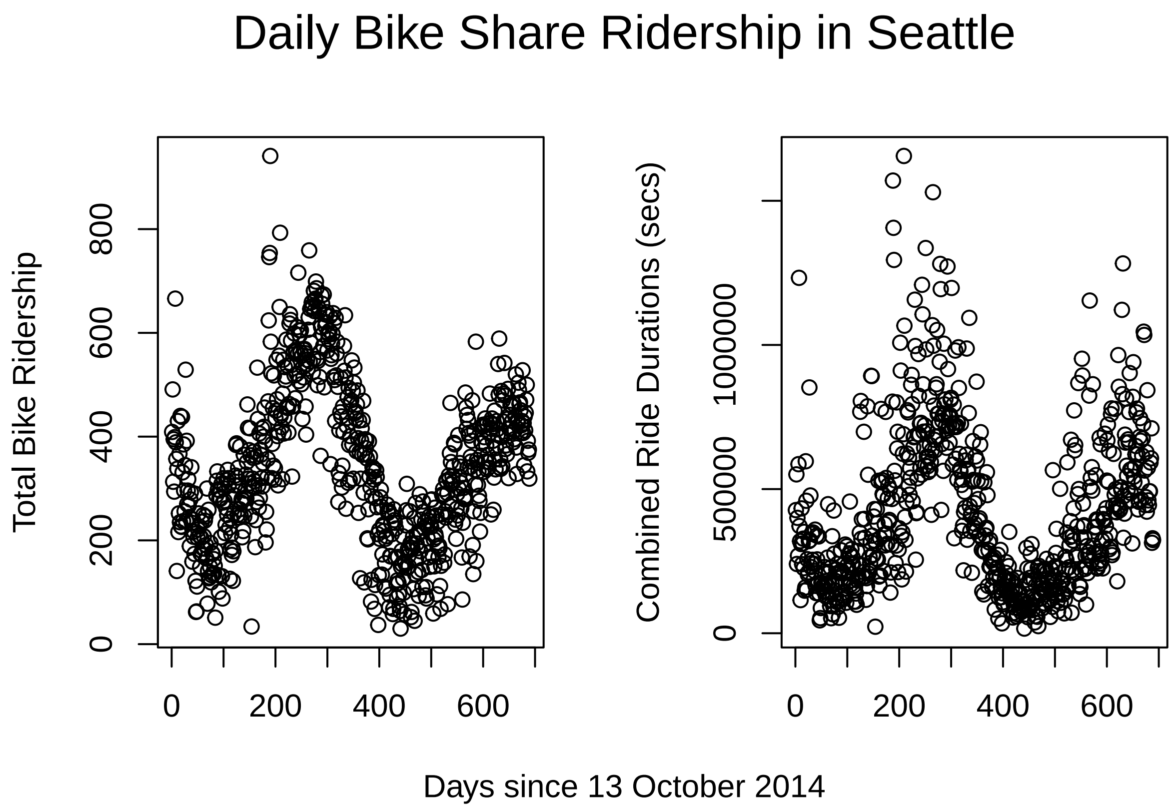
[INSERT PARAGRAPH SUMMARIZING CONCLUSIONS FROM THE RESEARCH]

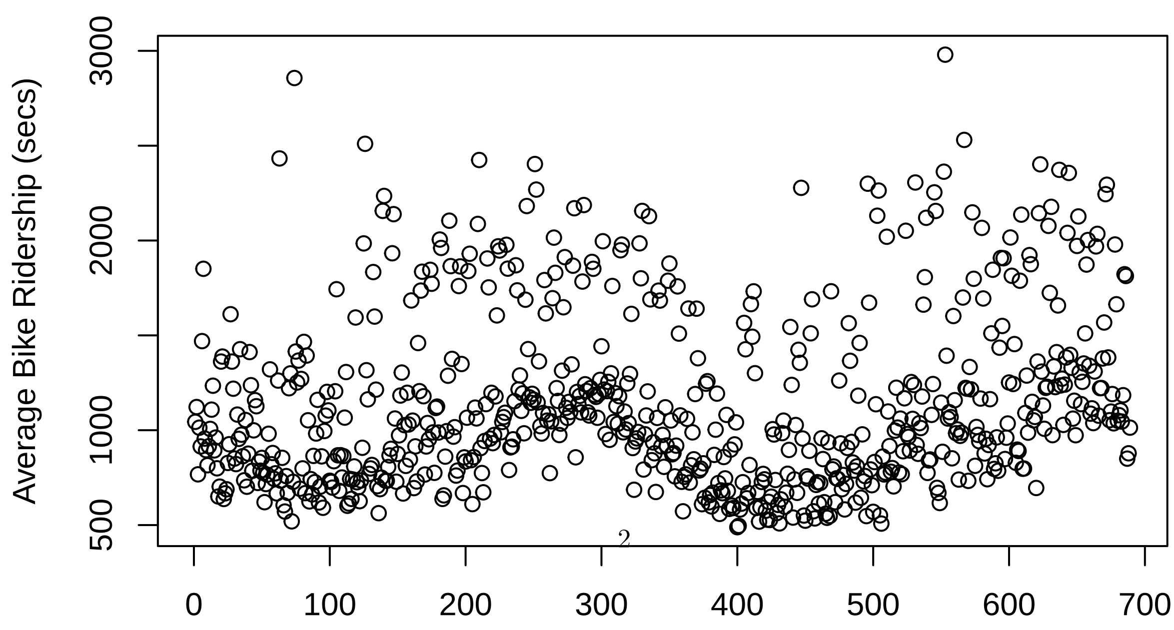
Exploratory Data Analysis

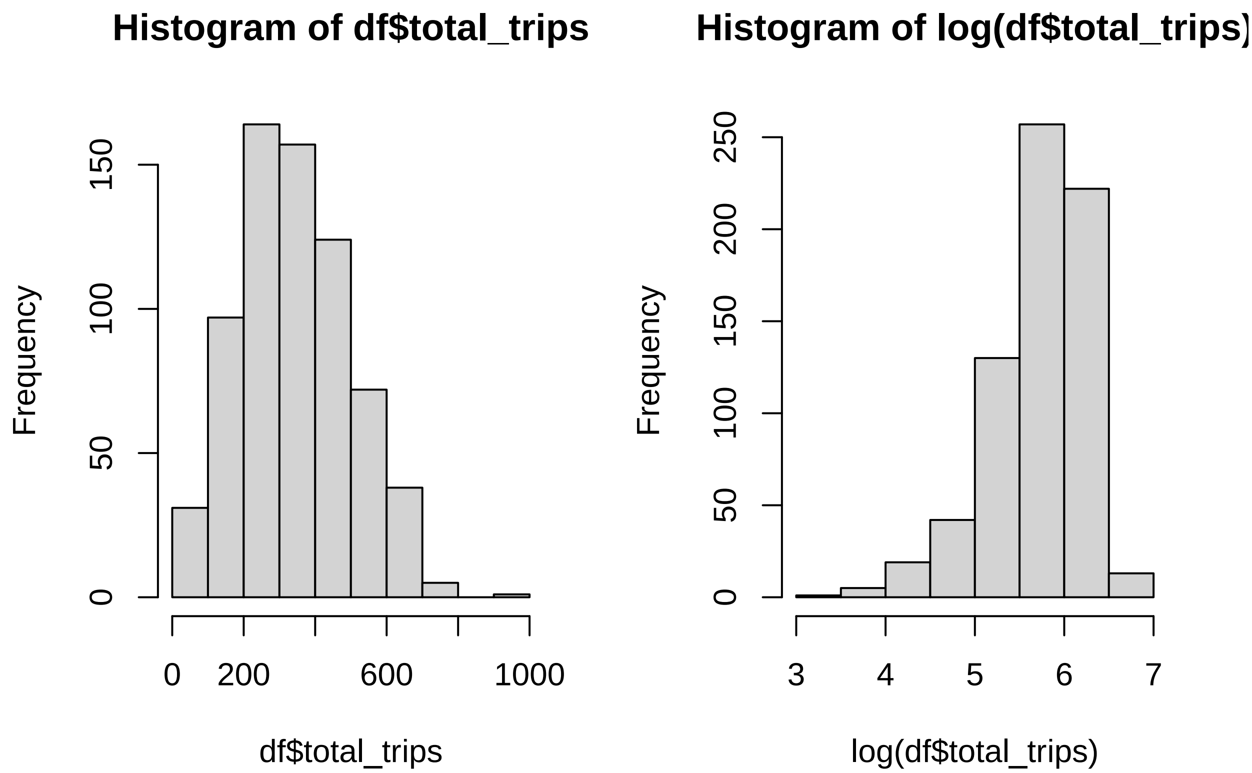
Data Cleaning

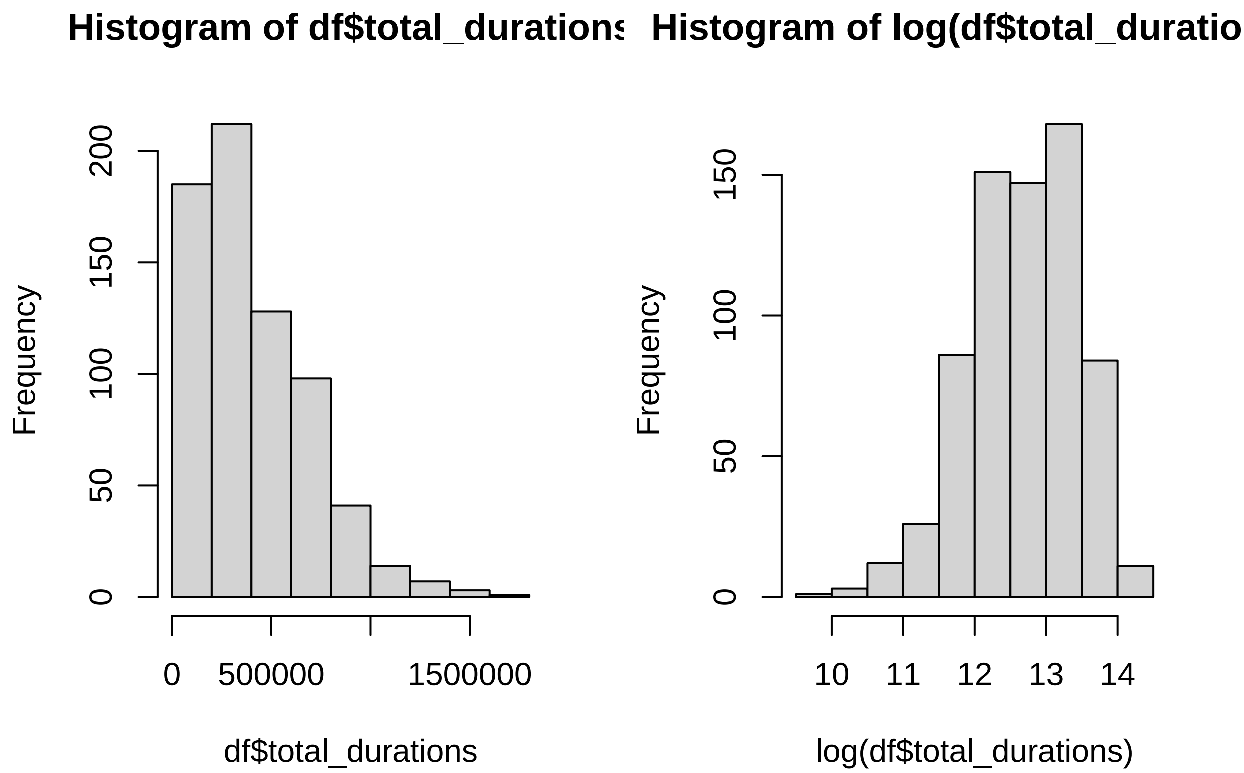
Selecting the Response Variable

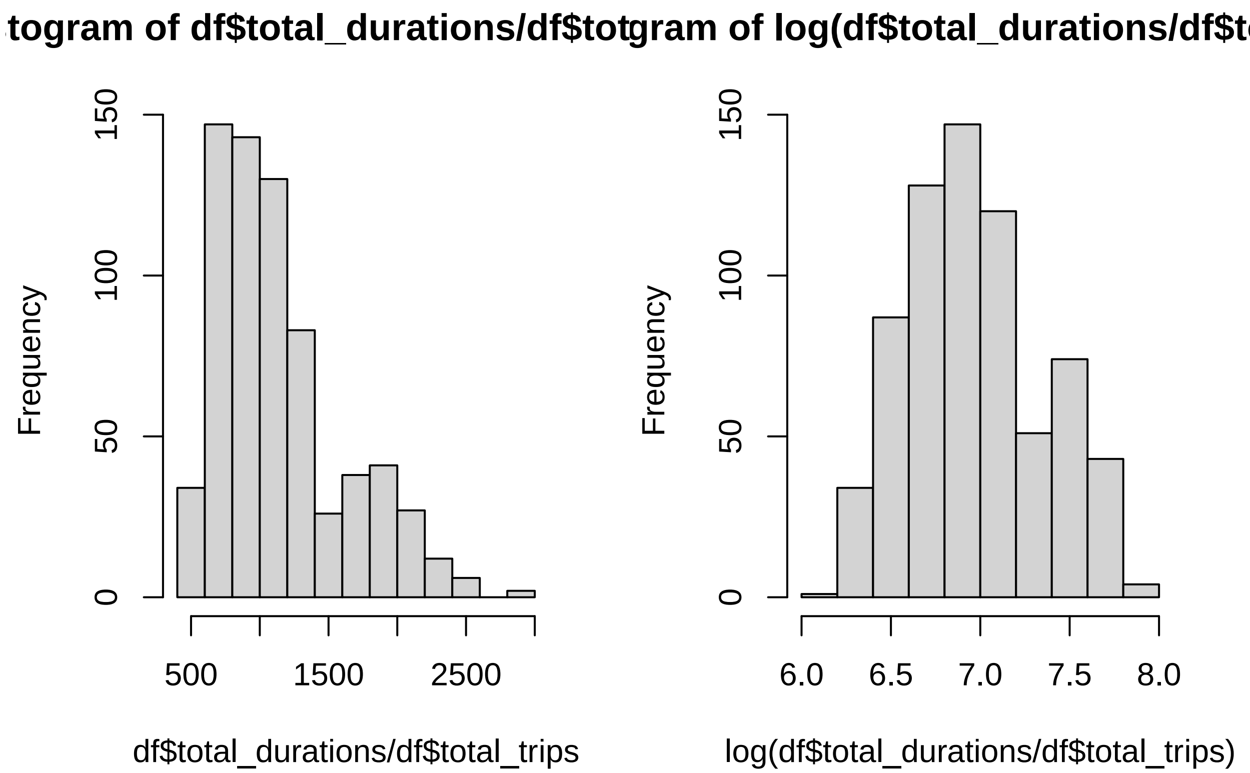
Outliers






Histogram of df$total_trips
Frequency
0
50
100
150
0
200
600
1000
df$total_trips
Histogram of log(df$total_trips)
Frequency
0
50
100
150
200
250
3
4
5
6
7
log(df$total_trips)


Histogram of df$total_durations
Frequency
0
50
100
150
200
0
500000
1500000
df$total_durations
Histogram of log(df$total_duratio
Frequency
0
50
100
150
10
11
12
13
14
log(df$total_durations)



Data selection

The `trip` data frame contains 175,091 cases (or rides) and 12 variables describing each ride. The weather data frame contains 689 cases (or days) and 21 variables describing the weather that day. Ultimately, our goal is to join these two data frames. We began by aggregating trip data for each day we have data for.

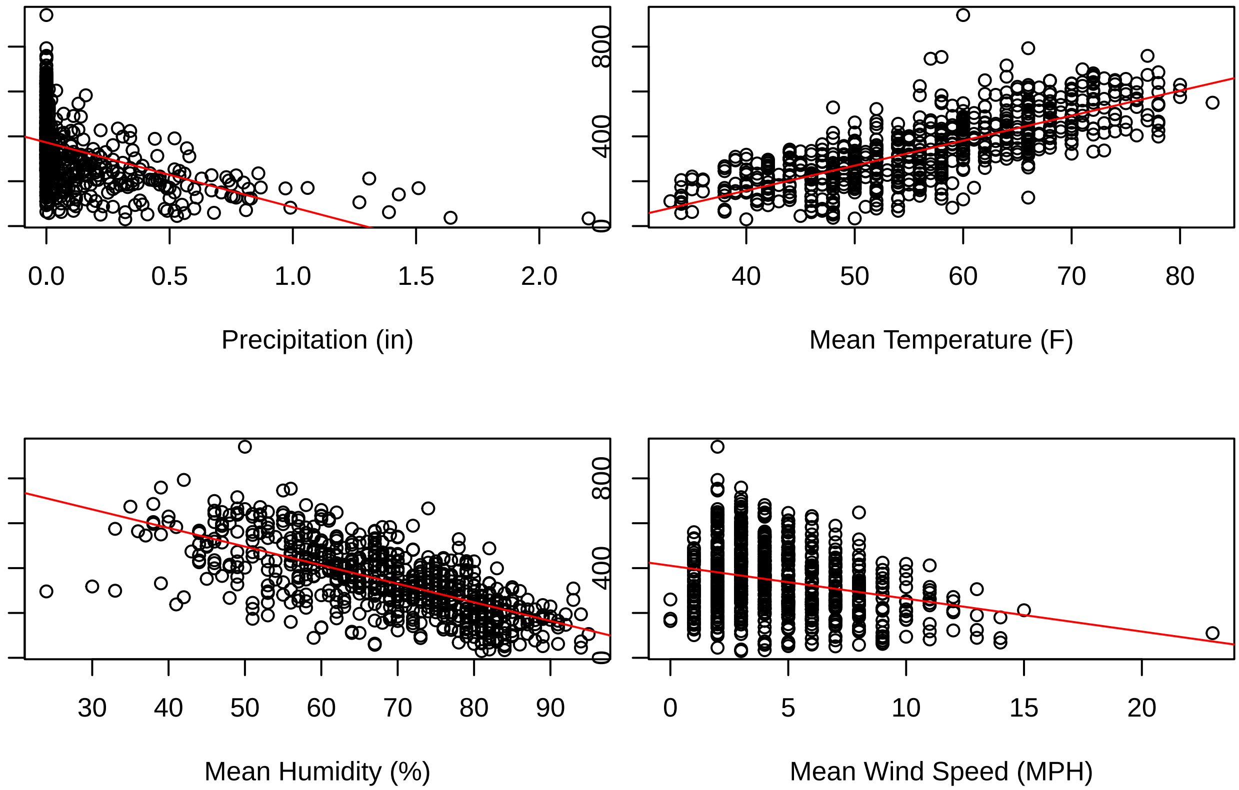
From the `trip` data frame, we selected only two variables: `start_time` and `tripduration`. We stripped the date from the `start_time` (encoded in the format `%m/%d/%Y %H:%M`), collected the unique dates, and mapped unique dates to the natural numbers. The mapping was represented by a new variable called `day_number`. Now, each trip has a duration `tripduration`, a trip `count` (1), and a `day_number` (ranging from 1 to 689). From the `trip` data frame, we created a new data frame called `ridership` that aggregates trips by day. At the end of this, `ridership` has 901 rows (days) and 3 columns (variables): `count`, `tripduration`, and `day_number`.

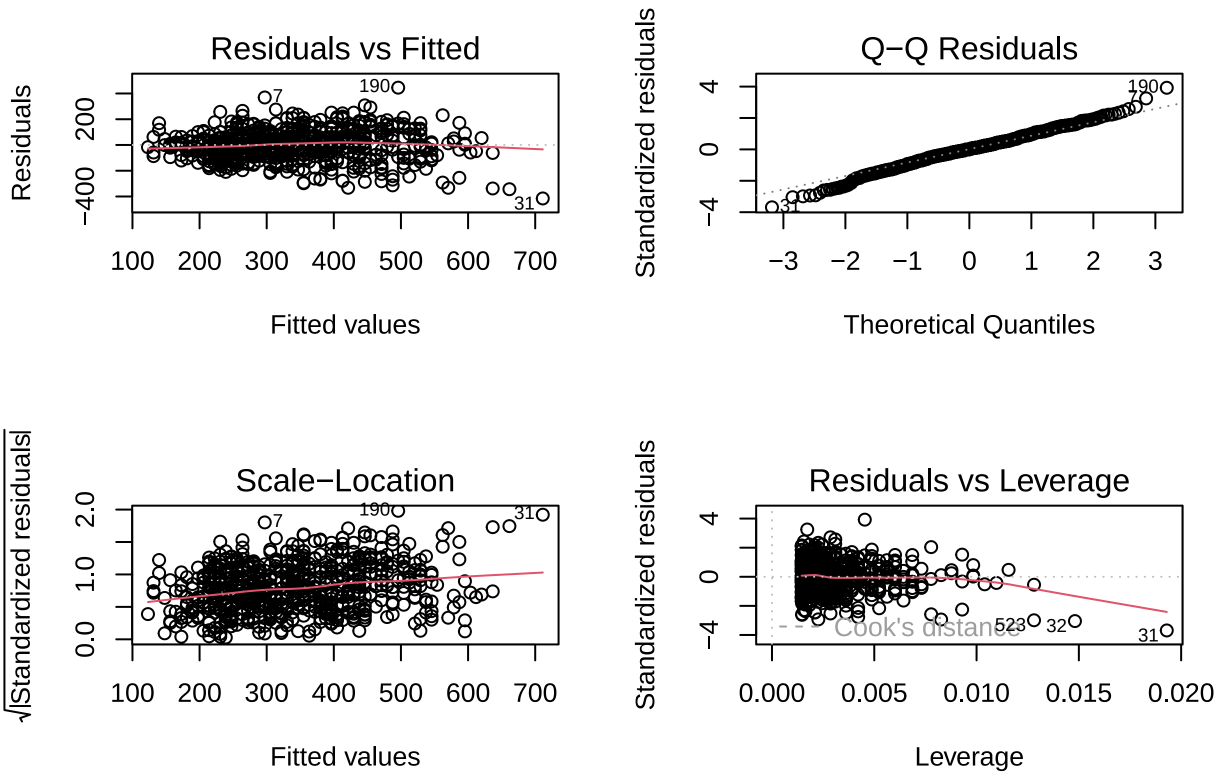
From the `weather` data frame, we used the mapping created for the `trip` data set to map unique dates to the natural numbers. We also created a new variable, `temp_range`, by computing the difference between `Max_Temperature_F` and `Min_TemperatureF` for each day. Because the `trip` data covers 212 days after the last observation in the `weather` data, we want to keep only the observations in `trip` that match the observations in the smaller data frame, `weather`. We created our final data frame, `df`, by left joining weather and ridership by `day_number`. The final data frame contains 689 rows (days) and 25 columns (variables). The variable names are listed in a table below with brief descriptions.

Table 1: Variable Descriptions (689 rows, 25 columns)

Variable	Description
Max_Temperature_F	Maximum temperature in Fahrenheit recorded that day
Mean_Temperature_F	Mean temperature in Fahrenheit recorded that day
Min_TemperatureF	Minimum temperature in Fahrenheit recorded that day
Max_Dew_Point_F	Maximum dew point in Fahrenheit recorded that day
MeanDew_Point_F	Mean dew point in Fahrenheit recorded that day
Min_Dewpoint_F	Minimum dew point in Fahrenheit recorded that day
Max_Humidity	Maximum humidity percentage recorded that day
Mean_Humidity	Mean humidity percentage recorded that day
Min_Humidity	Minimum humidity percentage recorded that day
Max_Sea_Level_Pressure_In	Maximum sea-level pressure in inches recorded that day
Mean_Sea_Level_Pressure_In	Mean sea-level pressure in inches recorded that day
Min_Sea_Level_Pressure_In	Minimum sea-level pressure in inches recorded that day
Max_Visibility_Miles	Maximum visibility in miles recorded that day
Mean_Visibility_Miles	Mean visibility in miles recorded that day
Min_Visibility_Miles	Minimum visibility in miles recorded that day
Max_Wind_Speed_MPH	Maximum wind speed in miles per hour recorded that day
Mean_Wind_Speed_MPH	Mean wind speed in miles per hour recorded that day
Max_Gust_Speed_MPH	Maximum gust speed in miles per hour recorded that day
Precipitation_In	Precipitation in inches recorded that day
Events	Weather events (e.g., Rain, Snow) that occurred that day
temp_range	Temperature range (Max_Temperature_F - Min_TemperatureF)
date	Date of the observation
day_number	Days since 13 October 2014
total_trips	Total trips recorded that day
total_durations	Total duration of all trips recorded that day in seconds

Methods/ Analysis






```
##
## Call:
## lm(formula = total_trips ~ Mean_Humidity, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -415.28  -63.52    1.48   67.50  445.10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   910.0991    23.7690   38.29   <2e-16 ***
## Mean_Humidity  -8.2840     0.3412  -24.28   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113.7 on 687 degrees of freedom
## Multiple R-squared:  0.4619, Adjusted R-squared:  0.4611
## F-statistic: 589.6 on 1 and 687 DF,  p-value: < 2.2e-16
```

Conclusion/Discussion

Model Selection

Our model and how we derived it:

The equation for our final regression model is:

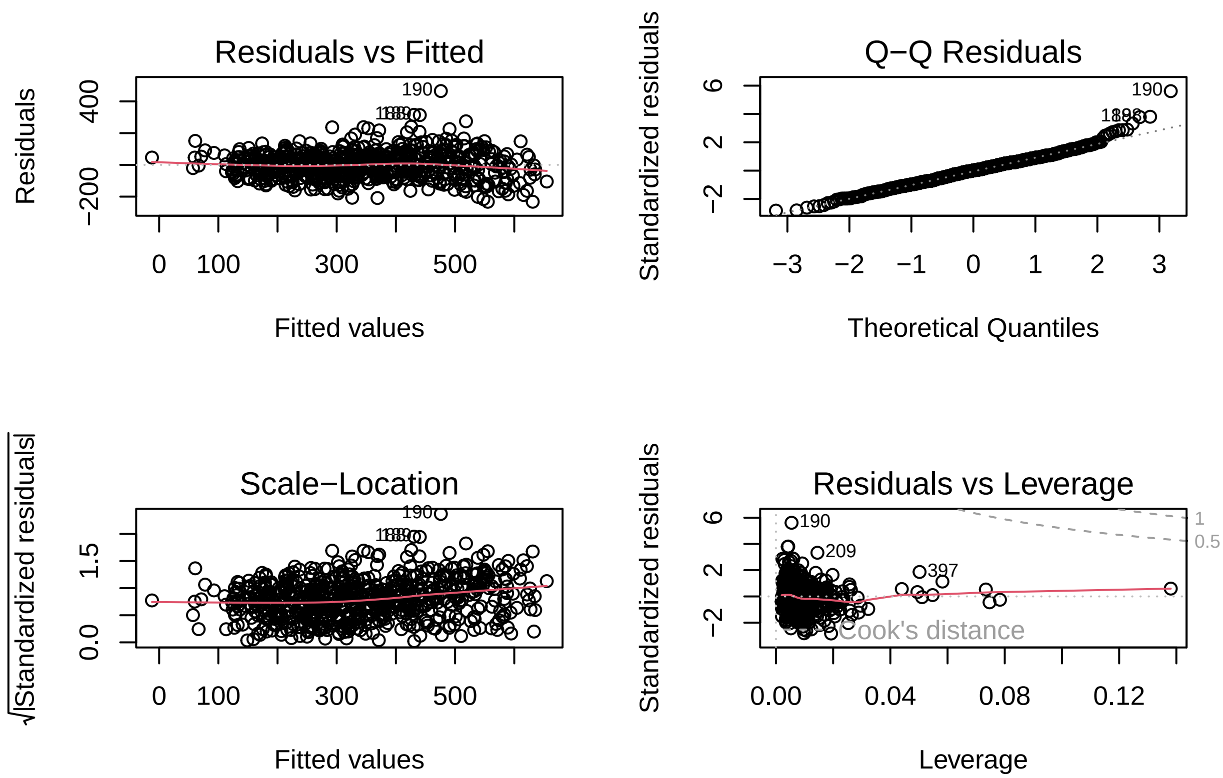
total_trips = 277.1643 − 4.95(Mean_Humidity) + 5.90(MeanDew_Point) − 118.151(Precipitation_In) − 7.91(Mean_Wind_Speed) + 2.94(Max_Temperature)

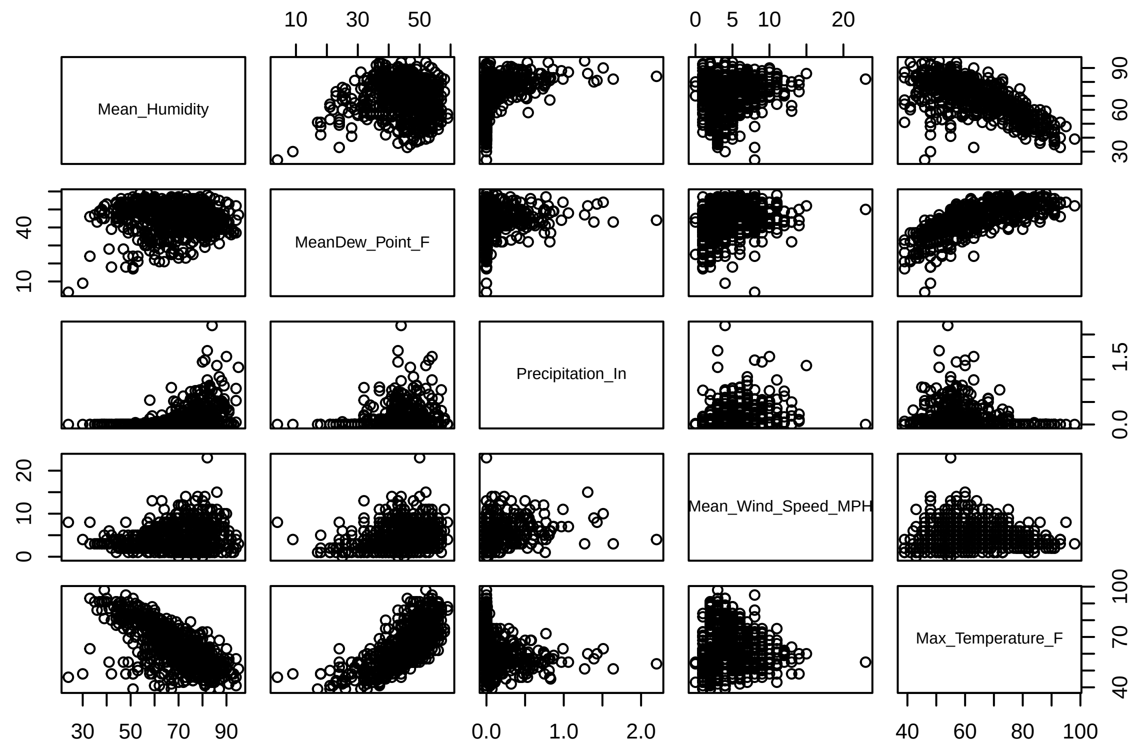
Each regressor is significant to at least the 0.01 level, and the diagnostic plots are satisfactory. The residual plot moving-average looks flat. The normal quantile plot has few departures from the line. The scale location plot is relatively flat, indicating constant variance across fitted values.

In order to obtain this model, we first did some exploratory data analysis, plotting total trips against certain variables and obtaining their correlation. We then attemped several basic (single regressor) linear models of total_trips vs some of our variables. The variables that we considered were: mean temperature, mean humidity, mean dew point, precipitation in inches, mean wind speed, mean miles of visibility, mean sea level pressure, temperature range, and max temperature. These were the original variables we considered on account of the fact that they seemed to have some relationship with total trips based on plots and correlation, and made sense to us as predictors of ridership from an intuitive perspective. After trying several single variable models, we found that few of them had very high R^2, with a notable exception that a weighted least squares model of wind speed high worked very well. We then decided to make a "full" model with all of the variables above, except only using mean temperature instead of temperature range and max temperature as 1. mean and max were extremely highly correlated (>97%) and 2. we thought two temperature variables would be redundant. The output of the full model showed that Mean temperature, mean visibility miles, and mean sea level pressure did not have coefficients that were significantly different from zero. We made a model dropping these parameters and then did an anova (partial f) between the two models to see if we could justifiably drop them and it showed that we could. At this point all of the predictors were significant and the adjusted R^2 was 0.7095. However, we got the idea to try adding temperature range or max temperature to this model. Including temperature range did not improve the R^2, and it was not significant in the model, however, including max temperature did improve the adjusted R^2 and the regressor was also significant. We decided that this would be our final model. Each regressor is significant at at least the 0.01 level, and the diagnostic plots look good. The adjusted R^2 is 0.712. We also tried to use powerTransformations, but it only worked for some of the variables as others did not have strictly positive values. For the variables that did successfully power transform, the adjusted R^2 of the subsequent model was not greatly improved and few of the regressors were significant. Our final model has some nice properties, in that the diagnostic plots show that it satisfies the assumptions for linear regression well, it is perfectly basic in terms of transformations, and partly on account of that, it is not too difficult to interpret. In terms of interpretation, our model predicts that holding all else equal, every 1% increase in average humidity will lead the total number of bike trips to decrease by 4.95. It predicts that holding all else equal, for every 1 degree increase in the average dew point (Fahrenheit for this and all future mentions of degrees) Seattle will see a drop in bike trips of 5.90. Our model predicts that all else equal, for every 1 extra inch of precipitation, total bike rides will drop by 118. It also predicts that holding all else equal, a 1 mile per hour increase in the average wind speed will decrease total bike trips by 7.91. Lastly our model predicts that holding all else equal, a 1 degree increase in the maximum temperature will increase the number of bike trips by 2.94.

```
##
## Call:
## lm(formula = total_trips ~ Mean_Humidity + MeanDew_Point_F +
##     Precipitation_In + Mean_Wind_Speed_MPH + Max_Temperature_F,
##     data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -232.29  -58.08    0.51   49.71  465.04
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          277.1643    69.9164   3.964 8.14e-05 ***
```

```
## Mean_Humidity          -4.9504     0.7503  -6.598 8.37e-11 ***
## MeanDew_Point_F         5.8968     1.3083   4.507 7.73e-06 ***
## Precipitation_In     -118.1507    15.2166  -7.765 3.00e-14 ***
## Mean_Wind_Speed_MPH    -7.9065     1.2656  -6.247 7.35e-10 ***
## Max_Temperature_F       2.9379     1.1138   2.638  0.00854 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 83.09 on 683 degrees of freedom
## Multiple R-squared:  0.7141, Adjusted R-squared:  0.712
## F-statistic: 341.1 on 5 and 683 DF,  p-value: < 2.2e-16
```





Cross Validation

Conclusion

Our work here suggests a path forward for bike share systems looking to bolster their operations and planning with weather data. However, Seattle is a city with temperate weather/climate. These results are not readily generalizable to all cities because when its too hot, people will also not ride bike!

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.