Predicting Bike Share Ridership based on Weather Data in Seattle

Joey Rodriguez and Daniel Bhatti

2024-11-22

Introduction

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

Exploratory Data Analysis

Data Cleaning

The data `trip.csv` and `weather.csv.xls` were downloaded from Kaggle (3). The `trip` data frame contains 275,091 cases (or rides) and 12 variables describing each ride. These data were collected over 901 days from 13 October 2014 to 31 March 2017. The relevant variables from the original 12 in this dataset are `start_time` (day and time trip started, in PST) and `trip_duration` (time of trip in seconds). The `weather` data frame contains 689 cases (or days) and 21 variables describing the weather that day. These data were collected from 13 October 2014 to 31 August 2016, or 689 days. Notice that the dates covered by the `weather` data set are a proper subset of the dates covered by the `trip` data set.

We began by aggregating trip data for each day we have data for. 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`. 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 28 columns (variables). The variable names are listed in the table below with brief descriptions.

Table 1: Variable Descriptions (689 days, 28 variables)

Variable	Description
Max_Temperature_F	Maximum temperature (°F)
Mean_Temperature_F	Mean temperature (°F)
Min_TemperatureF	Minimum temperature (°F)
Max_Dew_Point_F	Maximum dew point (°F)

Variable	Description
MeanDew_Point_F	Mean dew point (°F)
Min_Dewpoint_F	Minimum dew point (°F)
Max_Humidity	Maximum humidity (%)
Mean_Humidity	Mean humidity (%)
Min_Humidity	Minimum humidity (%)
Max_Sea_Level_Pressure_In	Maximum sea-level pressure (inches Hg)
Mean_Sea_Level_Pressure_In	Mean sea-level pressure (inches Hg)
Min_Sea_Level_Pressure_In	Minimum sea-level pressure (inches Hg)
Max_Visibility_Miles	Maximum visibility (miles)
Mean_Visibility_Miles	Mean visibility (miles)
Min_Visibility_Miles	Minimum visibility (miles)
Max_Wind_Speed_MPH	Maximum wind speed (MPH)
Mean_Wind_Speed_MPH	Mean wind speed (MPH)
Max_Gust_Speed_MPH	Maximum gust speed (MPH)
Precipitation_In	Precipitation (inches)
Events	Weather events (e.g., Rain, Snow)
temp_range	Temperature range (°F)
date	Date of the observation
day_number	Days since 12 October 2014
total_trips	Count of total trips
total_durations	Sum of total duration for all trips (seconds)
average_durations	Average ride duration (seconds)
weekday_weekend	Encodes weekends: 1 if Saturday or Sunday, 0 otherwise
season	Encodes seasons: 0 if Spring, 1 if Summer, 2 if Fall, 3 if Winter
fall_winter	Encodes wet season: 1 if Fall or Winter, 0 if Summer or Spring

Notice that nine variables in our data dictionary were created from other variables:

- `temp_range` was created by the difference: `Max_Temperature_F - Min_TemperatureF`
- `date` strips `"%m/%d/%Y"` from the full `starttime "%m/%d/%Y %H:%M"`
- `day_number` are the days beginning 13 October 2014, the first day of observation
- `total_trips` are the total trips recorded for that day
- `total_durations` are the total durations for all trips that day, aggregating `tripduration`
- `avg_durations` are the average durations for a trip each day, dividing the sum of trip durations by the total number of trips each day
- `weekday_weekend`
- `season` was created based on `date`, according the the summer and winter solstices and the spring and fall equinoxes in the Northern hemisphere.
- `fall_winter` was created based on whether the season was Fall or Winter, which roughly coincides with the wet season in the Puget Sound Region from October to April (SOURCE).

Understanding Outliers

The figure below plots daily bike ridership in Seattle, with the total rides taken each day in blue circles and the sum of the durations of the rides taken each day in red triangles. This figure suggests that outliers in

total riders tend to coincide with outliers in ride durations. For instance, the day with the highest bike riders – 941 on Sunday, April 20, 2015 – was also the day with the second highest sum of ride durations (359.7 hours). It's not clear from lookup what caused bike ridership to be so high on this day; like much of the data we gather from the real world, this result was influenced by many factors that day.

36 days earlier on Sunday, March 15, 2015 was the second-wettest March day on record in the Puget Sound Region (SOURCE). The rain was so severe that a mudslide occured in Western Seattle. Knowing this, you'd expect March 15 to have been a bad day for cycling. Only 34 trips took place on this day with a combined ride duration of just 6.3 hours. This was the second worst day for cycling behind Sunday, December 27, 2015 with just 30 trips and 4.5 hours. The coincidence between trips and durations explains the flattening of the data – the decrease in variation from the mean – for the average ride durations per day.

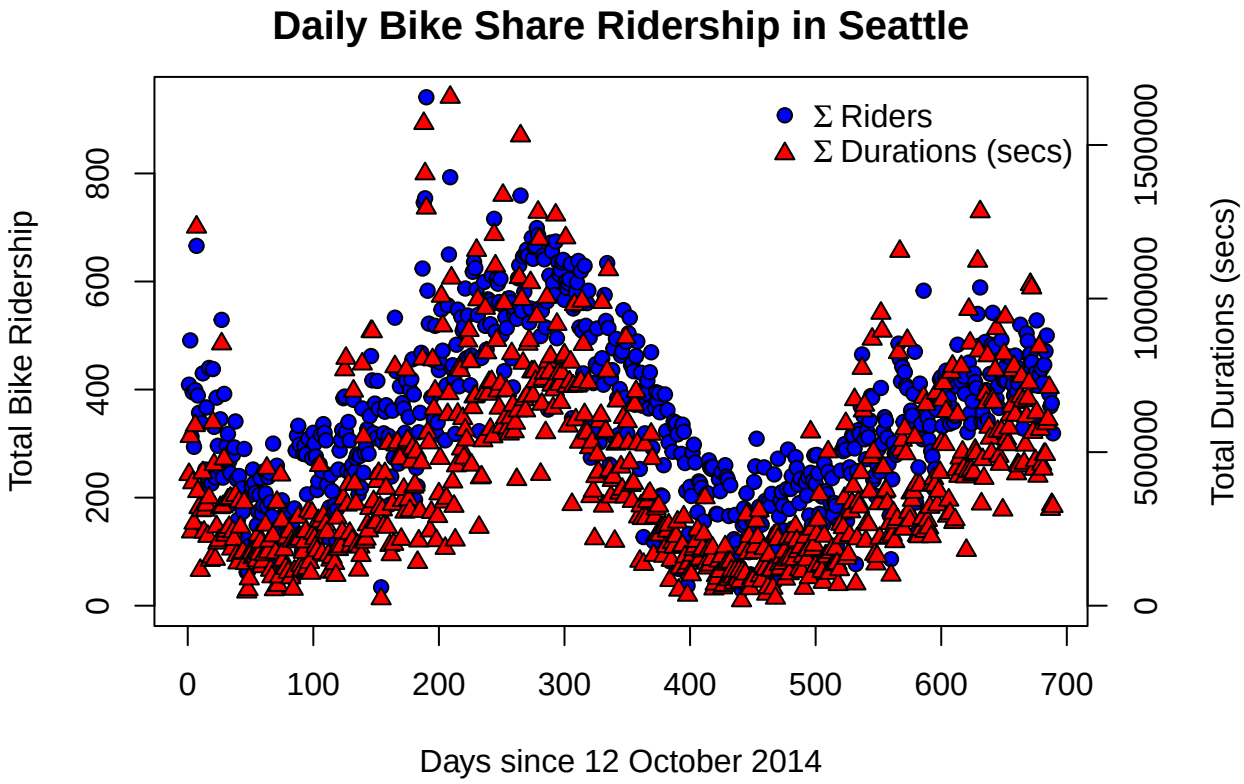


Figure 1: Daily Bike Share Ridership and Durations in Seattle.

Selecting the Response Variable

The three candidates for a good response variable were created from the `trip.csv` data set, described in the `ridership` data frame, and merged into our final data frame: `total_trips`, `trip_durations`, and `avg_durations`. We briefly discuss the merits of each response variable before a quantitative judgement:

- `total_trips` is the most intuitive measure for bike ridership on a given day. It directly answers the question "How many trips were there?" for a given day. It gives us a picture of how willing people in the service area were to hop on a bike.
- `total_durations` gives a more complete picture for the ridership on a given day. Once a rider hopped

on a bike, how long did they ride before docking it? This gives us a picture of how willing riders in the service area were to stay on their bikes.

- `avg_durations` controls for the interaction between bike ridership and ridership durations. By dividing total ridership over total durations, we understand the willingness of those in the service area to both picking up a bike and keep riding on that bike.

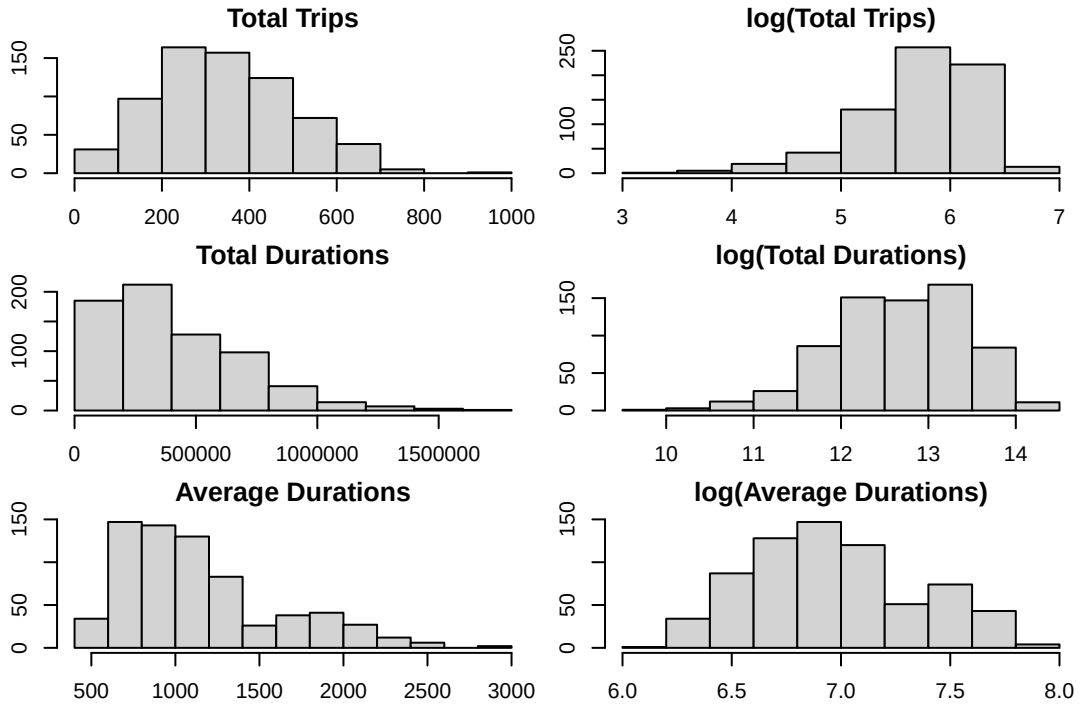


Figure 2: Variables Measuring Bike Share Ridership in Seattle

We note (i) that `total_trips` and `total_durations` are highly correlated (>0.82), (ii) that average durations is bimodal, total durations is right-skewed, and total trips is roughly normal, and (iii) total durations makes for the easiest interpretation without being transformed. We therefore use `total_trips` as our response variable going forward.

The majority of our predictors are continuous variables. For each of the features with recorded min, mean, and max — Visibility, Temperature, Dew Point, Humidity, and Sea Level Pressure – we calculate their correlations with the response Total Trips. By feature, it turns out that Max Temperature, Min Visibility, Mean Dew Point, Mean Humidity, and Min Sea Level Pressure have the highest correlation with the response (Table 2).

Total Trips' high correlation with Max Temperature may be explained by the temperature at midday, when it is usually highest. Midday may also be a peak time for ridership. We added each of these summary statistics with the highest correlation to Total Trips to our baseline model. Even though `Max_Gust_Speed_MPH` had higher response correlation than either `Max_Wind_Speed_MPH` or `Mean_Wind_Speed_MPH`, it also had 410 missing values. We opted to add `Mean_Wind_Speed_MPH` which had the next-highest response correlation (Figure 3). We also added Precipitation_In (zero-inflated continuous variable) and Events (dummy variable based on whether an event occurred that day).

Table 2: Correlation between Weather Features and Total Trips

Feature	Min	Mean	Max
Temperature (°F)	0.640	0.750	0.786
Visibility (miles)	0.470	0.364	0.058
Dew Point (°F)	0.396	0.452	0.433
Humidity (%)	-0.648	-0.680	-0.579
Sea Level Pressure (in)	0.180	0.079	-0.065

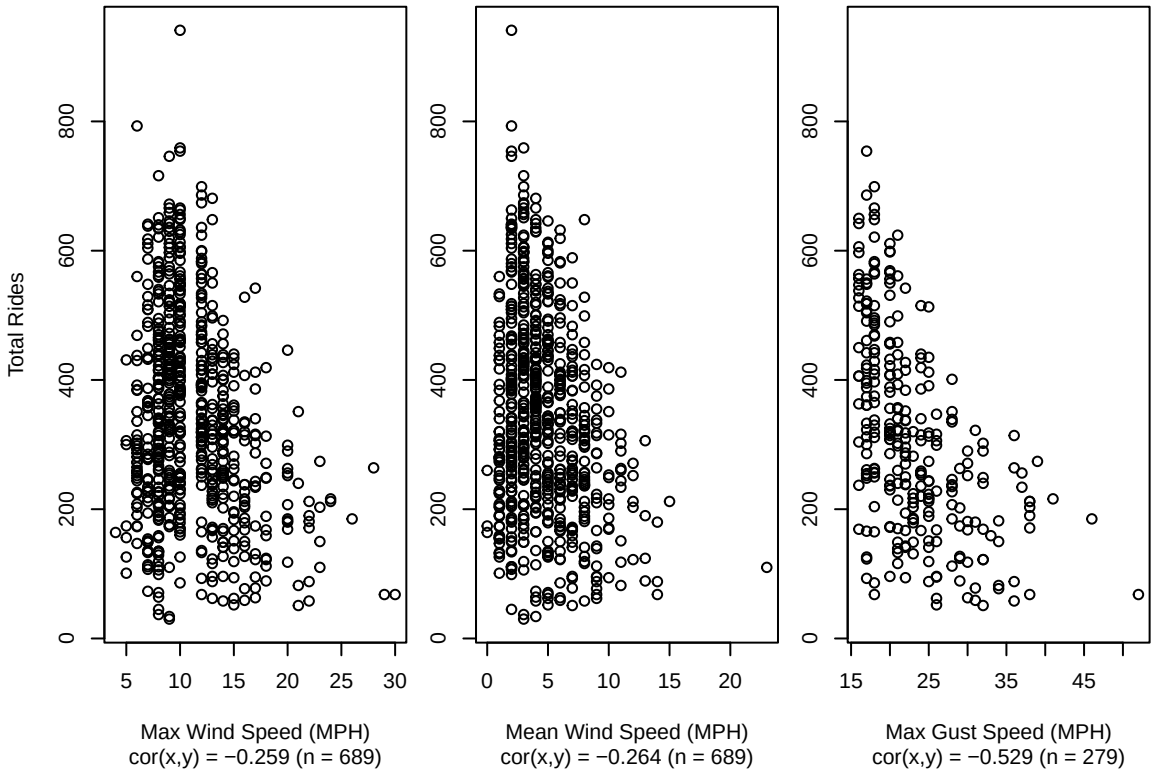


Figure 3: Effect of Wind Speed on Bike Share Ridership in Seattle.

Model Selection

Our baseline model using the finding from exploratory data analysis is:

$$\begin{aligned}\texttt{Total\_Trips}_i = \beta_{i0} &+ \beta_{i1}\texttt{Mean\_Humidity}_i + \beta_{i2}\texttt{MeanDew\_Point\_F}_i \\ &+ \beta_{i3}\texttt{Mean\_Wind\_Speed\_MPH}_i + \beta_{i4}\texttt{Max\_Temperature\_F}_i + \beta_{i5}\texttt{Min\_Visibility\_Miles}_i \\ &+ \beta_{i6}\texttt{Min\_Sea\_Level\_Pressure\_In}_i + \beta_{i7}\texttt{Precipitation\_In}_i + \beta_{i8}\texttt{Events}_i\end{aligned}$$

We removed `Min_Visibility_Miles`, `Min_Sea_Level_Pressure_In` and `Events` because they were insignificant predictors for the response. Our full model at this point (RSE = 83.09, R2 = 0.7141, R2adj= 0.712, all terms significant to 0.01) is:

$$\begin{aligned}\texttt{Total\_Trips}_i^{full} &= \beta_{i0} + \beta_{i1}\texttt{Max\_Temperature\_F}_i + \beta_{i2}\texttt{MeanDew\_Point\_F}_i \\ &+ \beta_{i3}\texttt{Mean\_Wind\_Speed\_MPH}_i + \beta_{i4}\texttt{Mean\_Humidity}_i + \beta_{i5}\texttt{Precipitation\_In}_i\end{aligned}$$

Transformations failed to produce superior models. Power Transform produced coefficients that were mostly close to one. Forward and backward selection both failed to produce a superior model, although they chose models similar to ours.

A partial model (RSE = 83.45, R2 = 0.7112, R2adj= 0.7095, all terms infinitesimal) without the term `Max_Temperature_F` had good summary statistics and diagnostic plots.

$$\begin{aligned}\texttt{Total\_Trips}_i^{part} &= \beta_{i0} + \beta_{i1}\texttt{MeanDew\_Point\_F}_i \\ &+ \beta_{i2}\texttt{Mean\_Wind\_Speed\_MPH}_i + \beta_{i3}\texttt{Max\_Humidity}_i + \beta_{i4}\texttt{Precipitation\_In}_i\end{aligned}$$

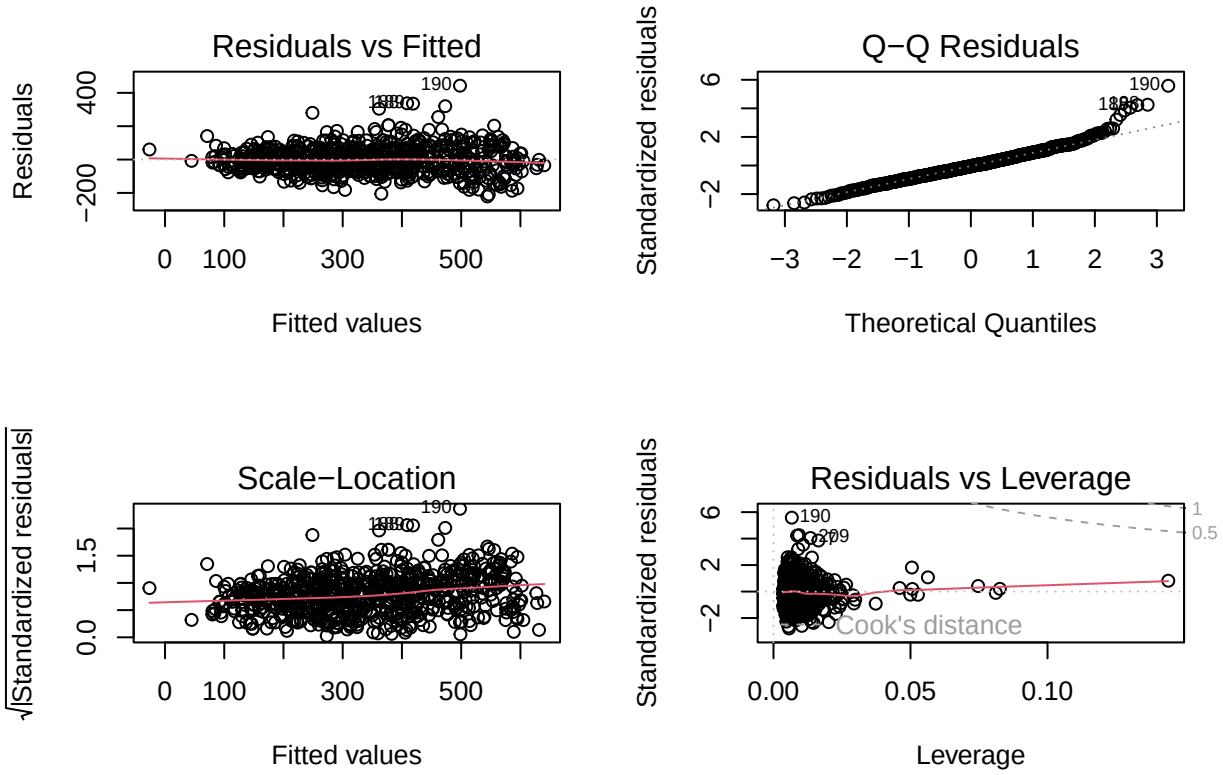
Judged side by side, both models have good summary statistics, diagnostic plots, and interpretable coefficients. The full model provides a better fit over the partial model based on partial F-test comparison (F = 6.9579, p = 0.0085). However, VIF is abnormally high for Mean_Humidity (VIF = 9.05), MeanDew_Point_F (VIF = 10.68), and Max_Temperature_F (VIF = 19.09) in the full model. It turns out that Max_Temperature_F has high correlation with MeanDew_Point_F (0.72) and Mean_Humidity (-0.67). VIF makes clear the confounding results in stepwise selection (Max_Temperature_F makes the best 1-predictor model). The partial model has VIF values near one. Thus, the partial model is the better model.

Finally, we strengthen our model by considering temporal variation. Adding `weekday_weekend` accounts for weekly changes in ridership based on commute changes for leisure, work and school. Adding `season` accounts for changes in ridership based on the assumption of four seasons while adding `fall_winter` accounts for changes in ridership based on the assumption of a wet and dry season:

$$\begin{aligned}\texttt{Total\_Trips}_i^{temp} &= \beta_{i0} + \beta_{i1}\texttt{MeanDew\_Point\_F}_i \\ &+ \beta_{i2}\texttt{Mean\_Wind\_Speed\_MPH}_i + \beta_{i3}\texttt{Max\_Humidity}_i + \beta_{i4}\texttt{Precipitation\_In}_i \\ &+ \beta_{i5}\texttt{weekday\_weekend}_i + \beta_{i6}\texttt{season}_i + \beta_{i7}\texttt{fall\_winter}_i\end{aligned}$$

Considering both the high correlation between variables `season` and `fall_winter` and the weakness of the `season` variable, we opt to drop this variable. Our final model is:

$$\begin{aligned}\texttt{Total\_Trips}_i^{fin} &= \beta_{i0} + \beta_{i1}\texttt{MeanDew\_Point\_F}_i \\ &+ \beta_{i2}\texttt{Mean\_Wind\_Speed\_MPH}_i + \beta_{i3}\texttt{Max\_Humidity}_i + \beta_{i4}\texttt{Precipitation\_In}_i \\ &+ \beta_{i5}\texttt{weekday\_weekend}_i + \beta_{i6}\texttt{fall\_winter}_i\end{aligned}$$



The equation for our final regression model is:

$$\texttt{total\_trips} = 277.1643 - 4.95(\texttt{Mean\_Humidity}) + 5.90(\texttt{MeanDew\_Point}) - 118.151(\texttt{Precipitation\_In}) - 7.91(\texttt{Mean\_Wind\_Speed}) + 2.94(\texttt{Max\_Temperature})$$

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.

Cross Validation, AIC/BIC, PowerTransform, VIF,

Conclusion

Interpretations

Limitations

Extensions

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! [INSERT PARAGRAPH SUMMARIZING CONCLUSIONS FROM THE RESEARCH]

Appendix

Code Used for Data Cleaning

```
trip = read_csv('pronto-cycle-share-trip-data.csv')
# map unique dates to integers starting at 1
# strips the date from its current format
trip$date <- as.Date(trip$starttime, format = "%m/%d/%Y %H:%M")
unique_dates <- sort(unique(trip$date)) # this collects unique dates
# this maps unique date to the integers, starting at 1
date_to_number <- setNames(seq_along(unique_dates), as.character(unique_dates))
# this adds the integer mapping as a column, day_number
trip$day_number = date_to_number[as.character(trip$date)]
trip$count = 1 # this adds a one to each obs; useful for add
trip = dplyr::select(trip, count, tripduration, day_number)
```

```
# construct new df, ridership, that aggregates trips by day
ridership = trip %>% group_by(day_number) %>%
  summarise(total_trips = sum(count),
            total_durations = round(sum(tripduration), 1),
            .groups = 'drop'); dim(ridership)
```

```
weather = read_csv('weather.csv.xls')
# calculates temperature range for each day
weather$temp_range = weather$Max_Temperature_F - weather$Min_TemperatureF
# strips the date from its current format
weather$date <- as.Date(weather$Date, format = "%m/%d/%Y")
# maps unique date to the integers, like the chunk above
date_to_number <- setNames(seq_along(unique_dates), as.character(unique_dates))
weather$day_number = date_to_number[as.character(weather$date)]
weather = weather[,-1] # remove the old date
# this will be our data frame going forward
df = left_join(weather,ridership, by='day_number'); dim(df)
df$avg_durations = round(df$total_durations / df$total_trips, 1)
```

```
df$weekday_weekend <- ifelse(weekdays(df$date) %in% c("Saturday", "Sunday"),1,0)
# Define a function to classify seasons based on actual start dates
get_season <- function(date) {
  year <- lubridate::year(date)
  spring_start <- as.Date(paste0(year, "-03-20"))
  summer_start <- as.Date(paste0(year, "-06-21"))
  fall_start   <- as.Date(paste0(year, "-09-22"))
  winter_start <- as.Date(paste0(year, "-12-21"))
  ifelse(date >= spring_start & date < summer_start, 0,  # Spring
  ifelse(date >= summer_start & date < fall_start,   1,  # Summer
  ifelse(date >= fall_start & date < winter_start,   2,  # Fall
  3)))  # Winter
}
# Apply the function to the 'date' variable
df$season <- sapply(df$date, get_season)
# Create fall/winter dummy: 1 if Fall or Winter, 0 otherwise
df$fall_winter <- ifelse(df$season %in% c(2, 3), 1, 0)
```