Space-Time Prediction of Bike Share Demand

Palak Agarwal

November 13, 2020

1 Introduction

The San Francisco Municipal Transportation Agency (SFMTA) is a department of the City and County of San Francisco responsible for the management of all ground transportation in the city. The SFMTA has oversight over the Municipal Railway (Muni) public transit, as well as bicycling, paratransit, parking, traffic, walking, and taxis. In 2013 they partnered with public agencies to plan and implement the original Bay Area Bike Share pilot project and are now leading San Francisco's efforts to work with the private sector partner to expand the system to over ten times its pilot size. They patnered with Lyft to expand the regional system which has since been re-branded with all-new equipment as Bay Wheels. In 2017, as independent stationless bikeshare emerged as a big new trend, they were among the first U.S. cities to create a regulatory and permitting framework to address this fast-moving phenomenon and insure that bikeshare in all its forms is safe, orderly and equitable for all San Franciscans.

One of the big operational challenges of bike share systems is "re-balancing" - getting bikes to stations that are anticipated to have demand but lack bikes. Figuring out how to do this is one of the keys to operating a successful system. With the introduction of the stationless bikeshare system the problem of re-balancing can be eliminated by a certain percentage. But it creates new problems as the bikes are further spread out and its harder to make sure that the bike is picked up. For the assignment, the data used is only from docked stations and does not include the stationless bikes.

For making the bike-share system more efficient i.e. to re-balance it many strategies can be introduced. I think the most effect strategy would be to use a reward system. If we know the capacity of a bike station and knew that the user was going to that station we could incentivize them to drop off the bike to another bike station or we could also incentivize them to use a bike which has not been touched for more than a two weeks. This will allow for an efficient demand and supply system.

Today's trend is similar to the trend tomorrow and this week's trend will be similar to next week trends. Hence, we will be able to predict how many bikes and from which station to high accuracy as you will see in the later in the model and assignment. If we know how many rides are going to take place, it is easier to use that data to manipulate which bikes are being used and which are not. For example, the Financial District of downtown SF is very busy and has many stations in the same block. If for some reason one station is more active than the other, we can use rewards to make people pick and drop bikes from different stations.

Lyft using its platform and app to show the users where the bikes are available so that can be used to make users pick up bikes that have not been touched for a while and also have them drop it off to locations where the demand is high and there is a shortage of bikes. SFMTA already has concessions for different individuals and communities which the bike share system can tap into as well. Knowing the capacity of a docking station will help us know if it has any open slots or not which can be a factor to provide users with offers to drop the bike off at other locations or if a busy dock is empty the user can be incentivized to drop off the bike there.

2.0 Setup

2.1 Libraries and themes

Let's load relevant libraries and some graphic themes.

```
## Warning: package 'riem' was built under R version 4.0.3
```

```
## Warning: package 'RSocrata' was built under R version 4.0.3
```

2.2 Data Wrangling

Reading in the bike data.

After that we use the data to create bins of 15 and 60 minute intervals by rounding.

2.3 Census Data

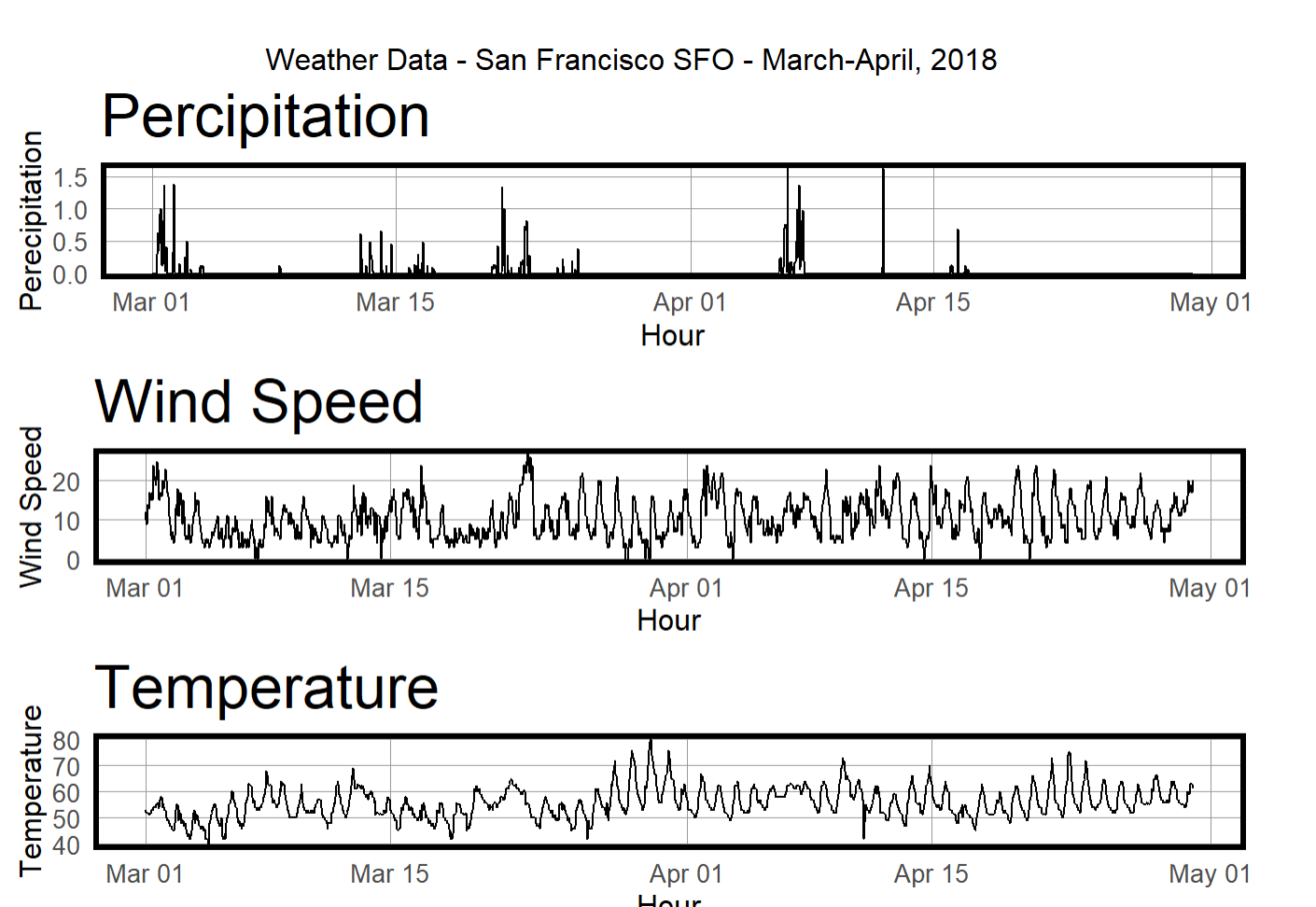
For the analysis, I am using 2018 census data for San Francisco and Alameda county.

2.4 Connect census data to the the Bikeshare data

Now lets add the spatial information to our bikeshare data as origin and destination data, first joining the origin station, then the destination station to our census data.

2.5 Weather Data

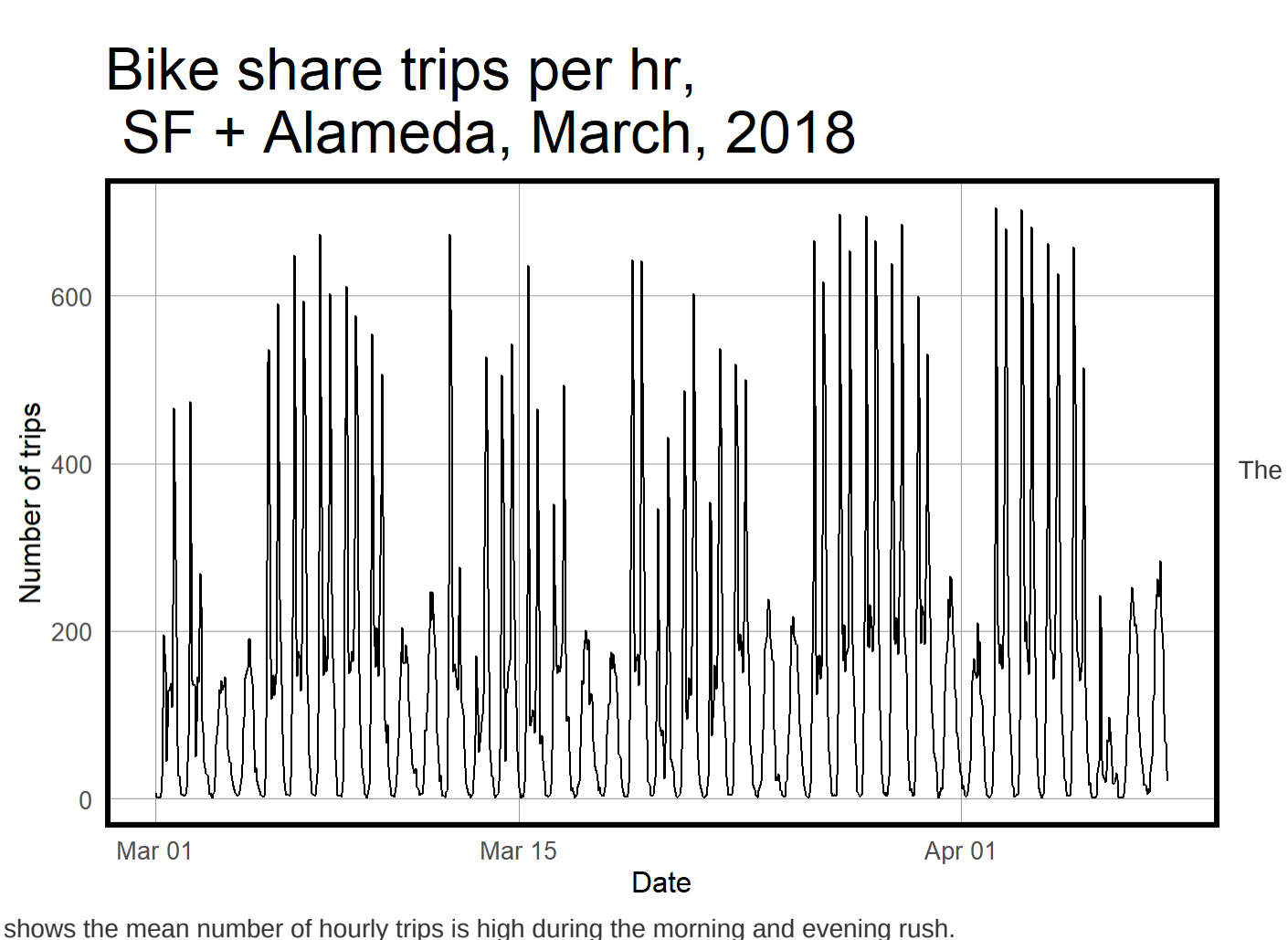
As weather plays an important factor in defining if one is likely to bike or not, e.g. "does precipitation appear to affect ridership during rush hour?". So we import weather data from SFO airport and get temperature, wind speed, precipitation on an hourly basis and plot the temperature and precipitation trends over our study period.
As you can see from the plot below, most days during the study months were pleasant and there wasnt much percipitation.



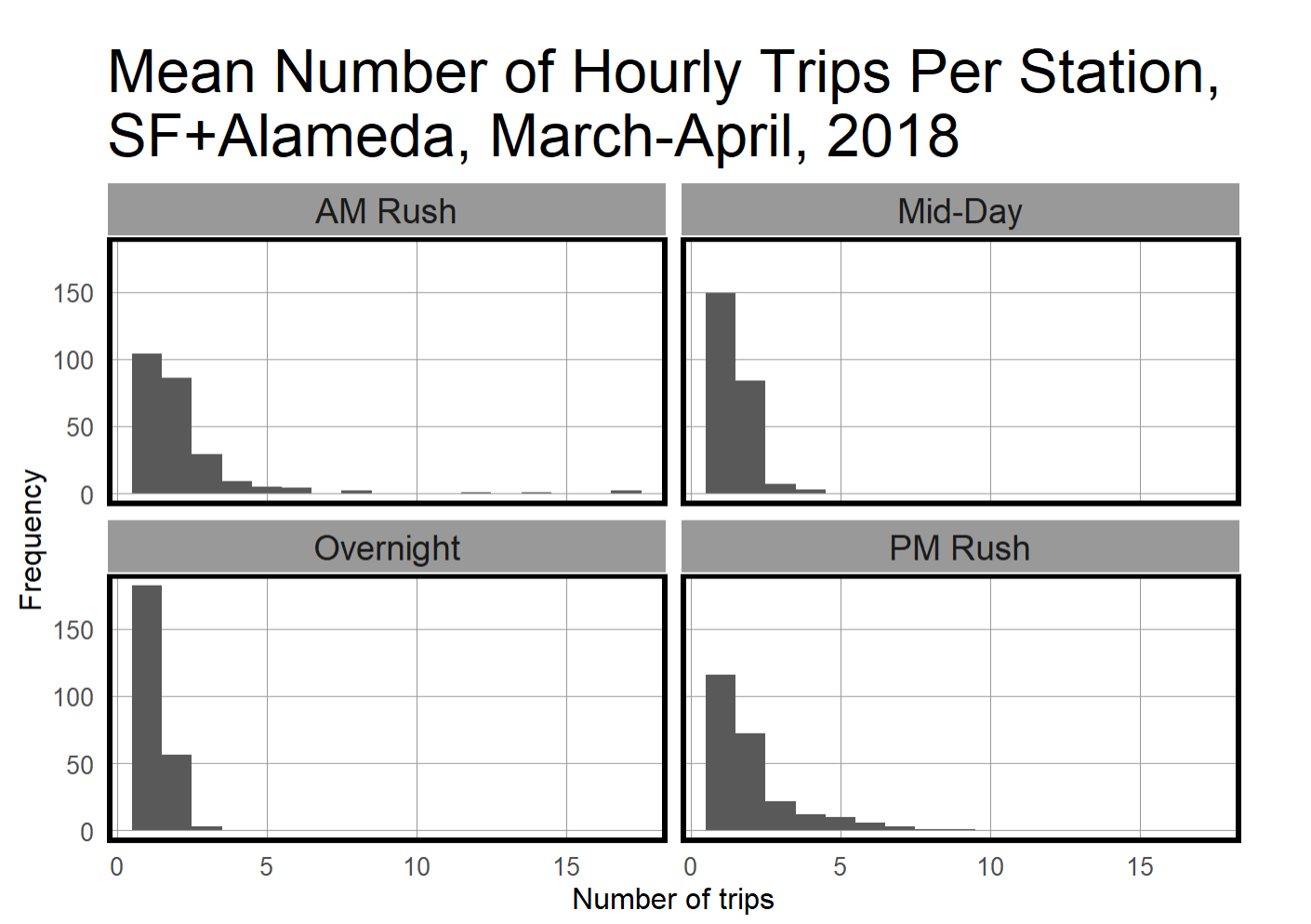
3 Data Exploration

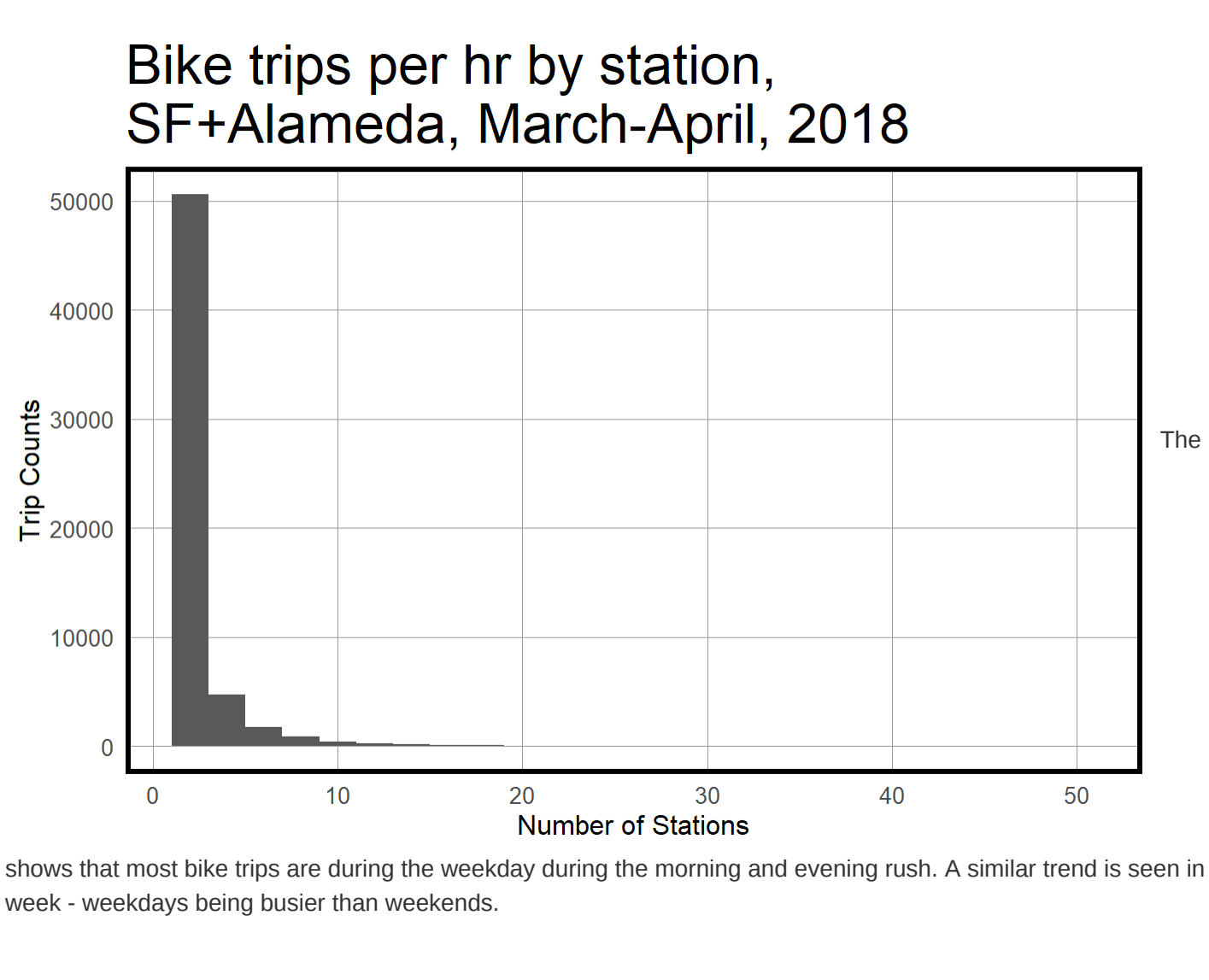
We begin by examining the time and frequency components of our data.

First, we look at the overall time pattern - there is clearly a daily periodicity and there are lull periods on weekends.

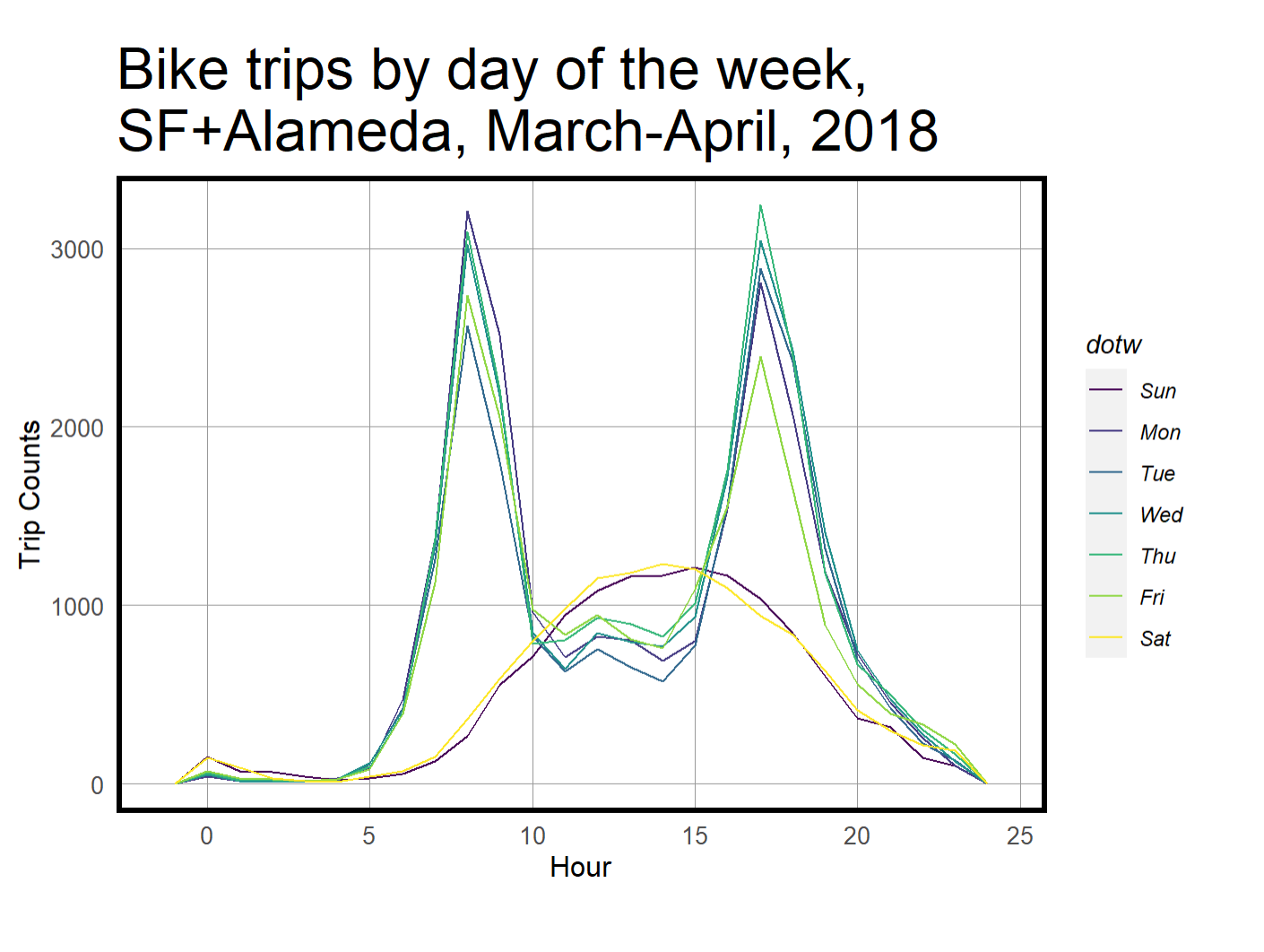


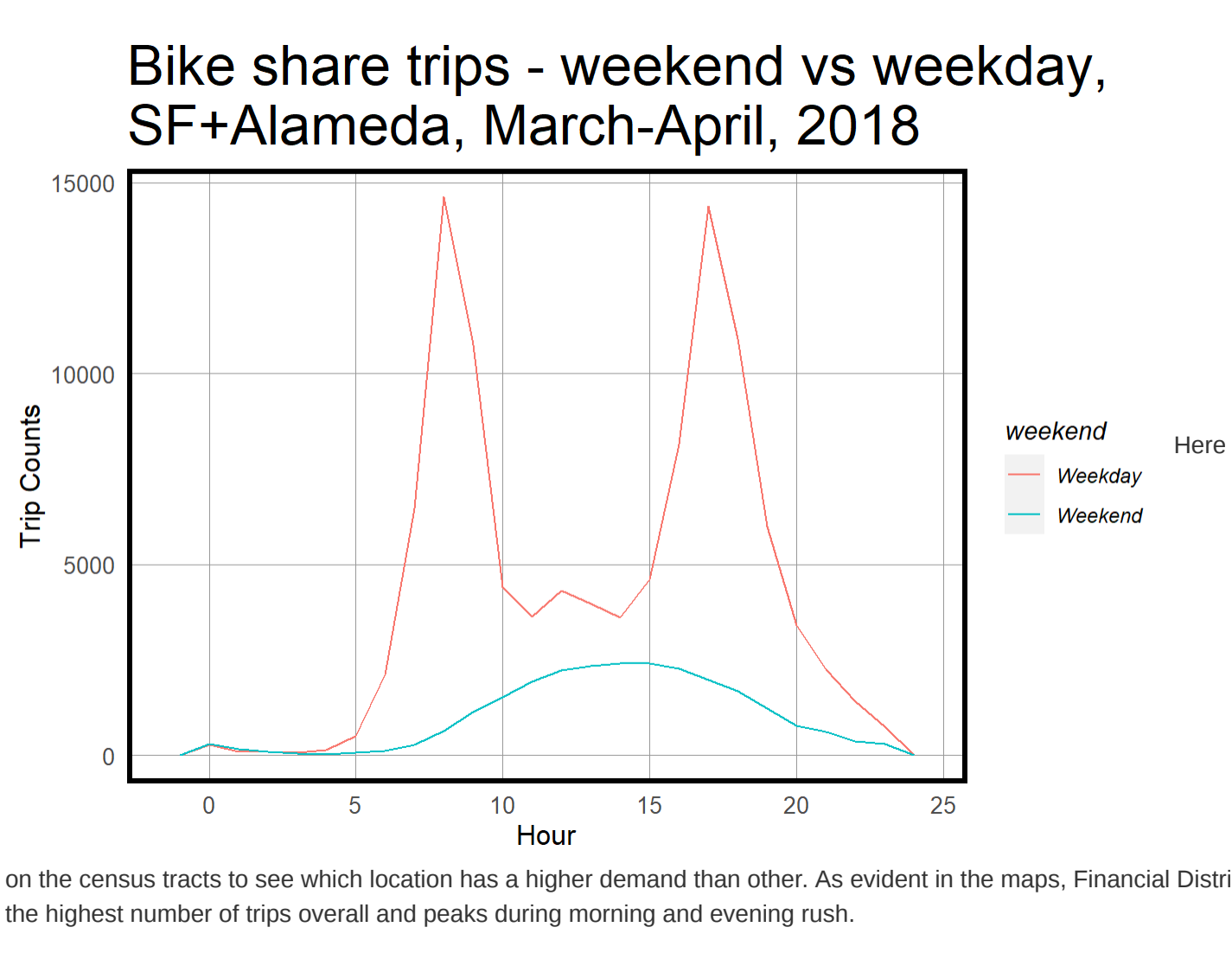
The below plot shows the mean number of hourly trips is high during the morning and evening rush.



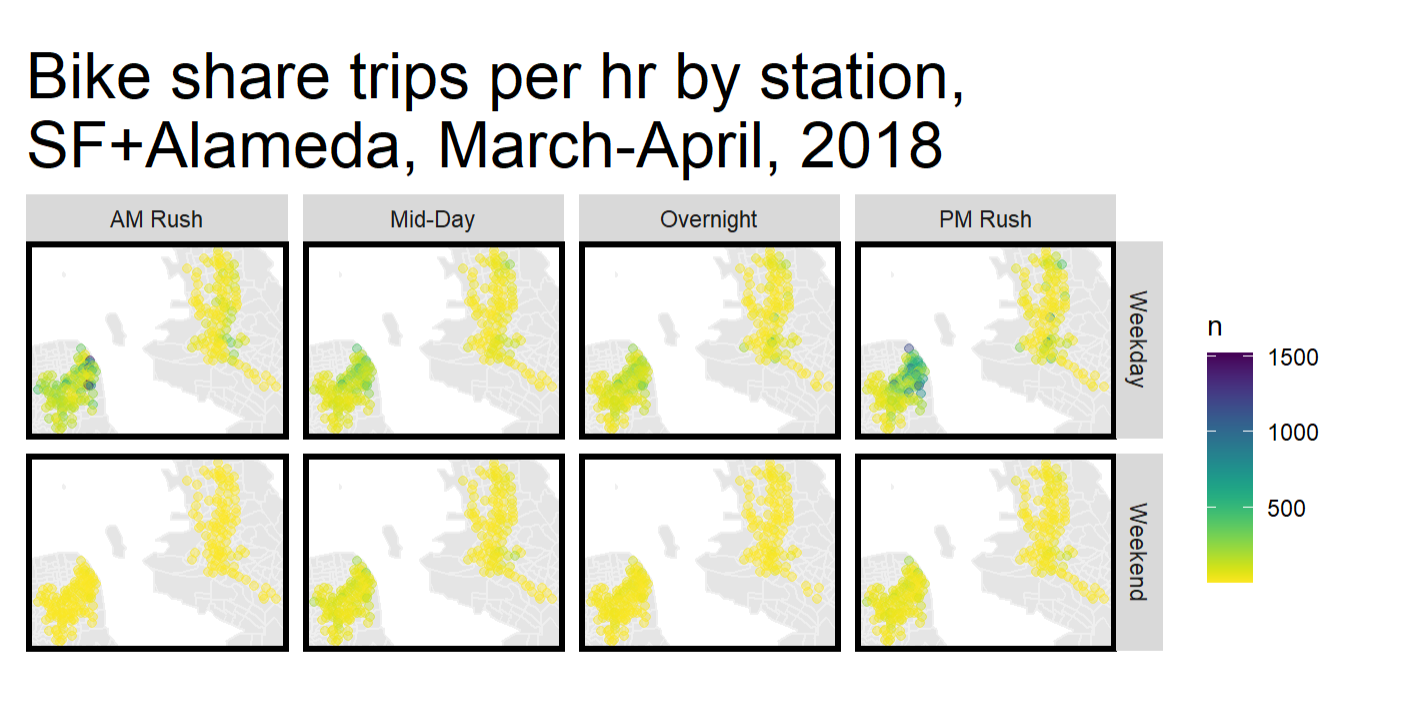


The plot below shows that most bike trips are during the weekday during the morning and evening rush. A similar trend is seen in the days of the week - weekdays being busier than weekends.





Here we the station on the census tracts to see which location has a higher demand than other. As evident in the maps, Financial District in SF sees the highest number of trips overall and peaks during morning and evening rush.



4 Create Space-Time Panel

We create a time-series panel which basically is a unique combination of station id to the hour and day.This is done in order to create a "panel" data set where each time period in the study is represented by a row - whether an observation took place then or not. So if a station didn't have any trips originating from it at a given hour, we still need a zero in that spot in the panel.

We start by determining the maximum number of combinations. Then we compare that to the actual number of combinations. We create an empty data frame study.panel, is created that has each unique space/time observations. This is done using the expand.grid function and unique. Along the way, we keep tabs on the number of rows our data have - nrow shows that the count is still correct. We then join the station name, tract and lat/lon and create the full panel by summarizing counts by station for each time interval, keep census info and lat/lon information along for joining later to other data.

```
## [1] 225747
```

```
## [1] 225747
```

5 Feature Engineering

5.1 Time Lags

As seen in the exploratory, morning and evening rush makes a difference in the demand for bikes. So here we create time lag variables which will give us additional information about the demand during a given time period.

We can evaluate the correlations in these lags. As you can see from the table below it is pretty strong. There's a Pearson's R of 0.72 for the lagHour. As mentioned in the introduction, this makes sense as the demand pattern for this hour is similar to that of last year and todays pattern is similar to that of yesterday and tomorrows.

```
## # A tibble: 6 x 2
##   Variable   correlation
##   <fct>            <dbl>
## 1 lagHour           0.77
## 2 lag2Hours         0.4
## 3 lag3Hours         0.16
## 4 lag4Hours         0.03
## 5 lag12Hours       -0.25
## 6 lag1day           0.72
```

5.2 Exposure Features

As the pick up points for the bikes, depends on the stations it is important to look at the location of these stations and factors that might affect it. Here we look at three exposure features - **proximity to parks, transit stops and tourist landmarks.** The stations tend to be clustered as the system is still expanding, so after calculating the distance to each of these features, we categorized it into categories - close (1), moderate (2) and far (3).

```
## Warning: no simple feature geometries present: returning a data.frame or tbl_df
```

6.1 Linear Regressions

We split our data into a training and a test set. We create five linear models using the `lm` function.We create the models using our training data `ride.Train` . The first models include only temporal controls, but the later ones contain all of our lag information and other exposure features modeled in the previous section.

6.2 Predict for test data

We create a function called `model_pred` which we can then `map` onto each data frame in our nested structure. As you can see from the table below, the MAE reduces as we add more features and temporal features to the regression.

```
## # A tibble: 10 x 8
##     week data     Regression     Prediction  Observed Absolute_Error   MAE sd_AE
##    <dbl> <list>   <chr>          <list>      <list>   <list>         <dbl> <dbl>
##  1    13 <tibble~ ATime_FE       <dbl [40,3~ <dbl [4~ <dbl [40,338]> 0.849  1.68
##  2    14 <tibble~ ATime_FE       <dbl [40,8~ <dbl [4~ <dbl [40,824]> 0.789  1.50
##  3    13 <tibble~ BSpace_FE      <dbl [40,3~ <dbl [4~ <dbl [40,338]> 0.829  1.53
##  4    14 <tibble~ BSpace_FE      <dbl [40,8~ <dbl [4~ <dbl [40,824]> 0.770  1.39
##  5    13 <tibble~ CTime_Space_FE <dbl [40,3~ <dbl [4~ <dbl [40,338]> 0.826  1.52
##  6    14 <tibble~ CTime_Space_FE <dbl [40,8~ <dbl [4~ <dbl [40,824]> 0.775  1.37
##  7    13 <tibble~ DTime_Space_F~ <dbl [40,3~ <dbl [4~ <dbl [40,338]> 0.699  1.21
##  8    14 <tibble~ DTime_Space_F~ <dbl [40,8~ <dbl [4~ <dbl [40,824]> 0.613  1.06
##  9    13 <tibble~ ETime_Space_F~ <dbl [40,3~ <dbl [4~ <dbl [40,338]> 0.699  1.21
## 10    14 <tibble~ ETime_Space_F~ <dbl [40,8~ <dbl [4~ <dbl [40,824]> 0.613  1.06
```

6.3 Cross Validation

Cross validation is important as it tells us about the generalizability of a model. A good model is one which is generalizable and can be built upon. To check our models generalizability, we take a sample of the data and run a 100 k-fold validation on it. You can see that the MAE is 0.59 which is quite low. Above table shows that the testing data had a MAE of 0.62.

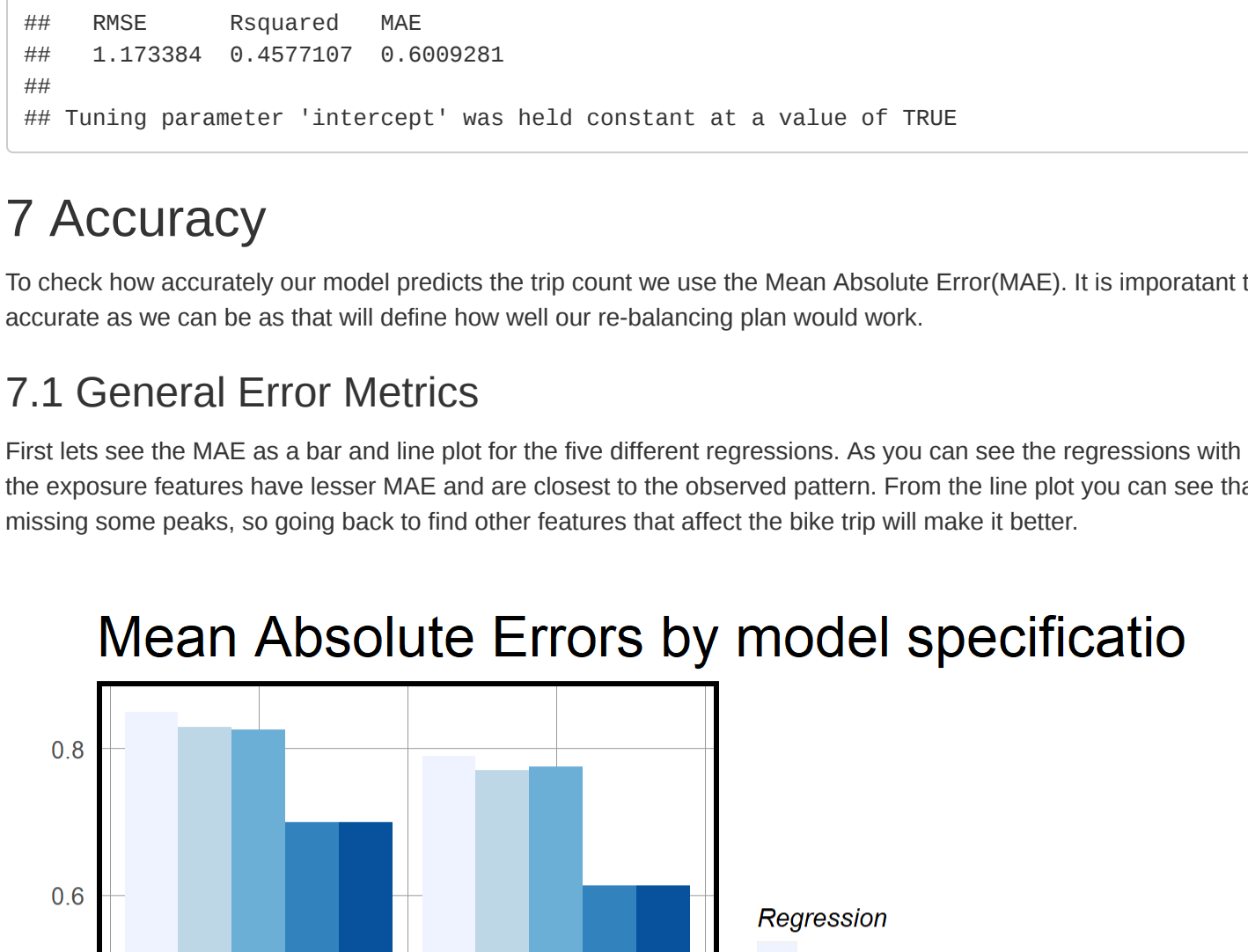
```
## Linear Regression
##
## 99989 samples
##    11 predictor
##
## No pre-processing
## Resampling: Cross-Validated (100 fold)
## Summary of sample sizes: 98989, 98989, 98990, 98989, 98989, 98989, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   1.173384  0.4577107  0.6009281
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

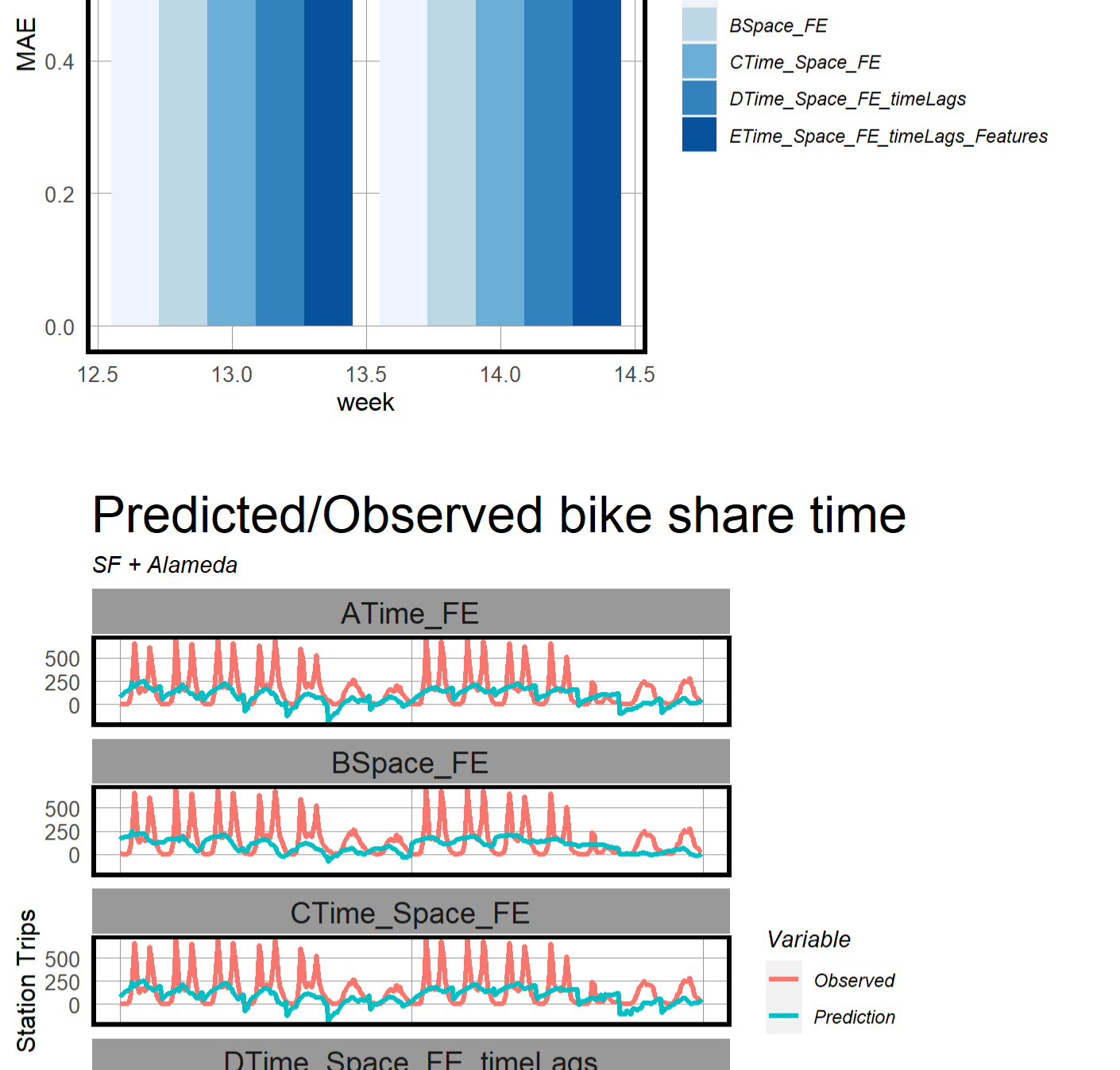
7 Accuracy

To check how accurately our model predicts the trip count we use the Mean Absolute Error(MAE). It is imporatant to be as accurate as we can be as that will define how well our re-balancing plan would work.

7.1 General Error Metrics

First lets see the MAE as a bar and line plot for the five different regressions. As you can see the regressions with time lags and the exposure features have lesser MAE and are closest to the observed pattern. From the line plot you can see that we are missing some peaks, so going back to find other features that affect the bike trip will make it better.

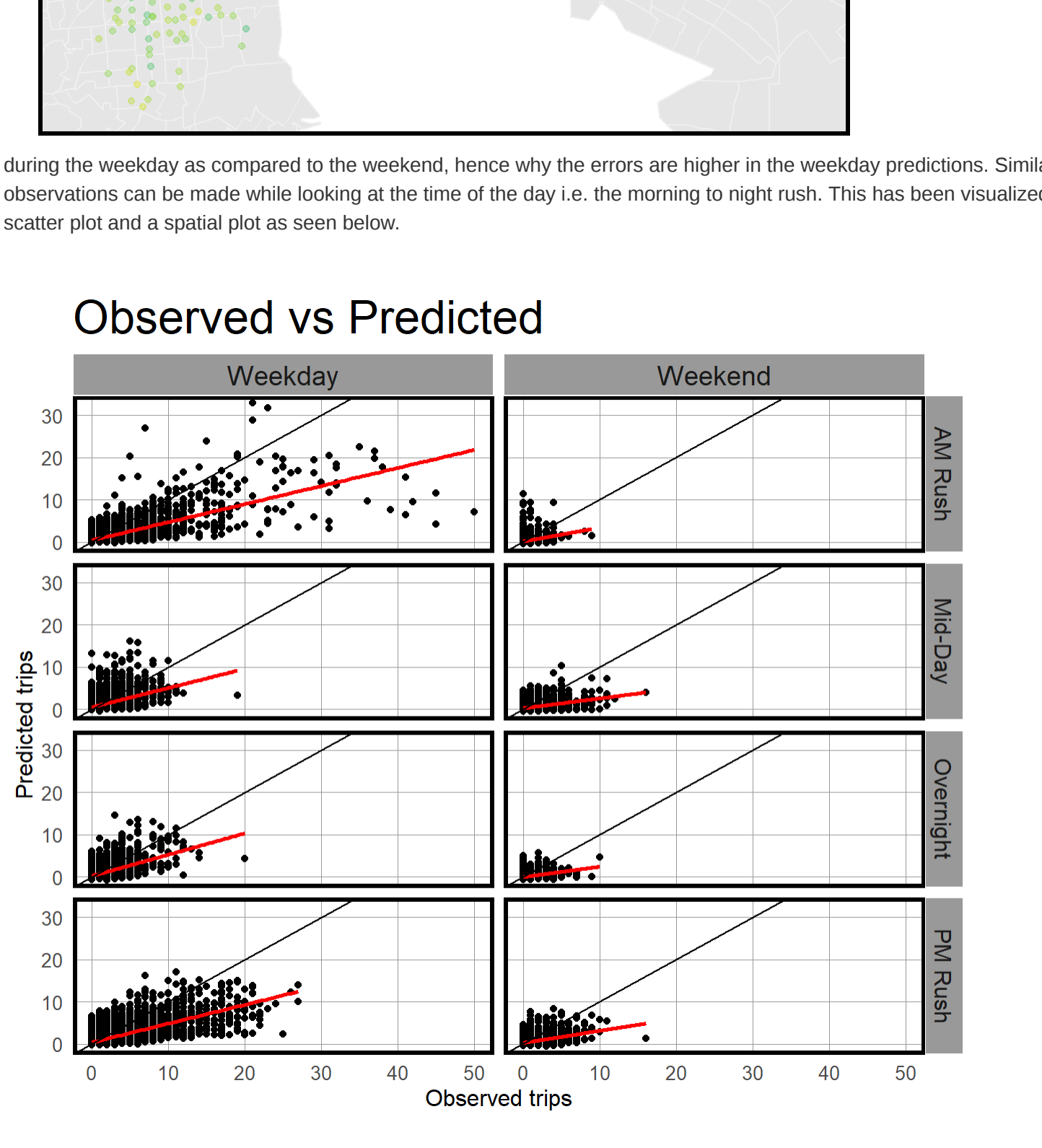


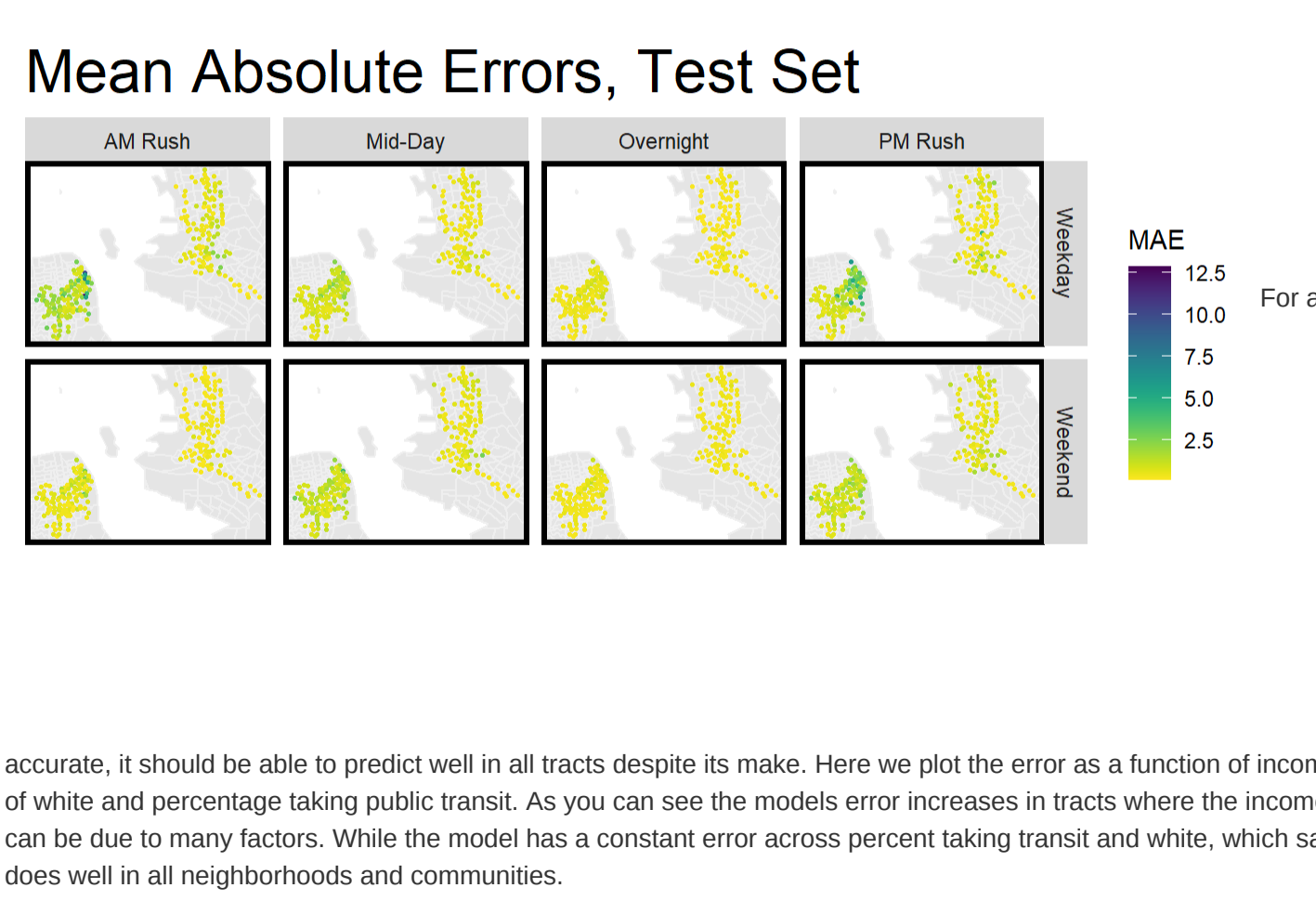


7.2 Specific Error Metrics

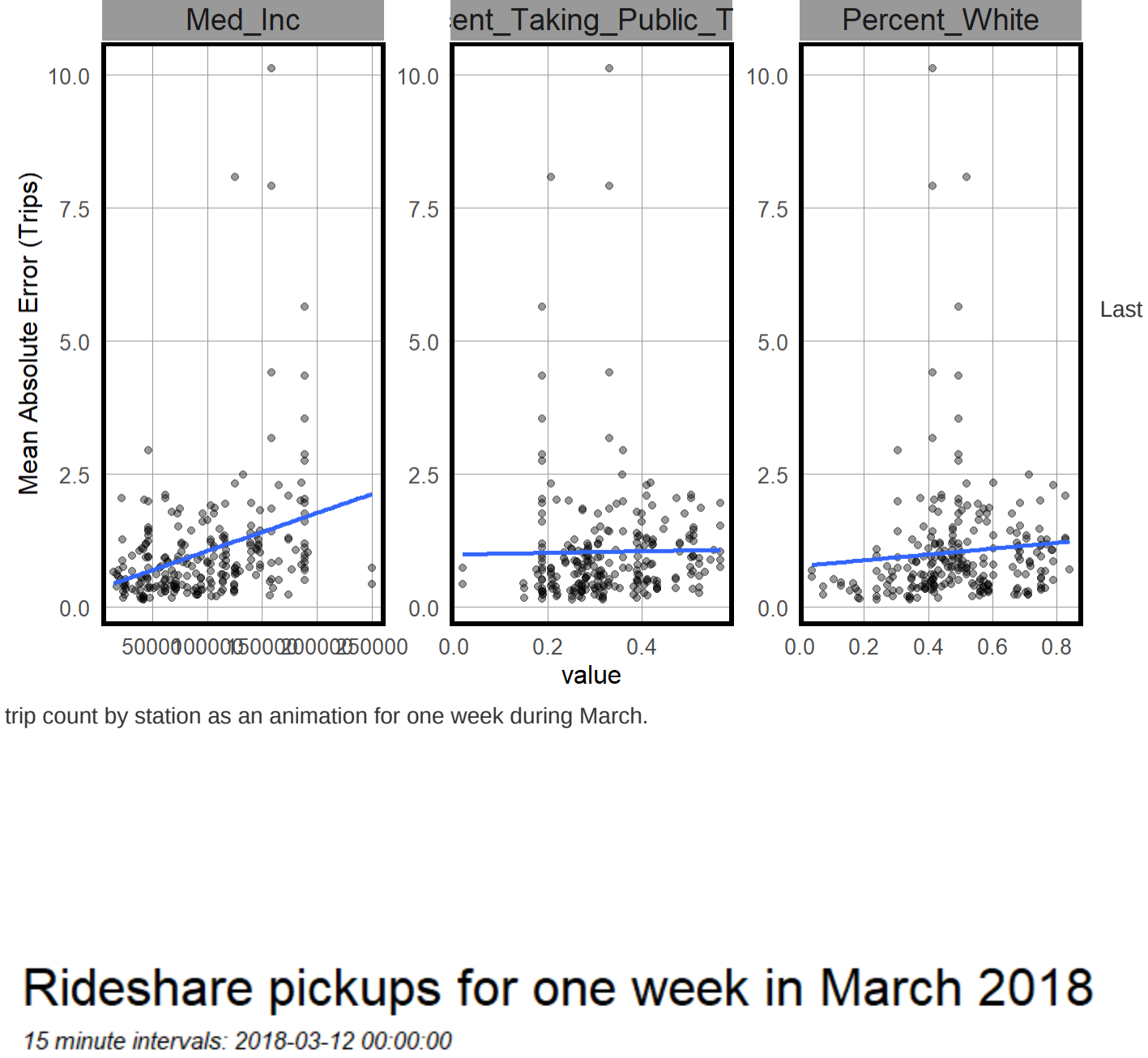
The highest error is in and around the Financial District in SF. That area has the highest bike ride number, hence why the error is also high. The CBD area is so important to SF that modeling some specific exposure features to that area might help improve the results.

The ridership is high during the weekday as compared to the weekend, hence why the errors are higher in the weekday predictions. Similar observations can be made while looking at the time of the day i.e. the morning to night rush. This has been visualized both as a scatter plot and a spatial plot as seen below.

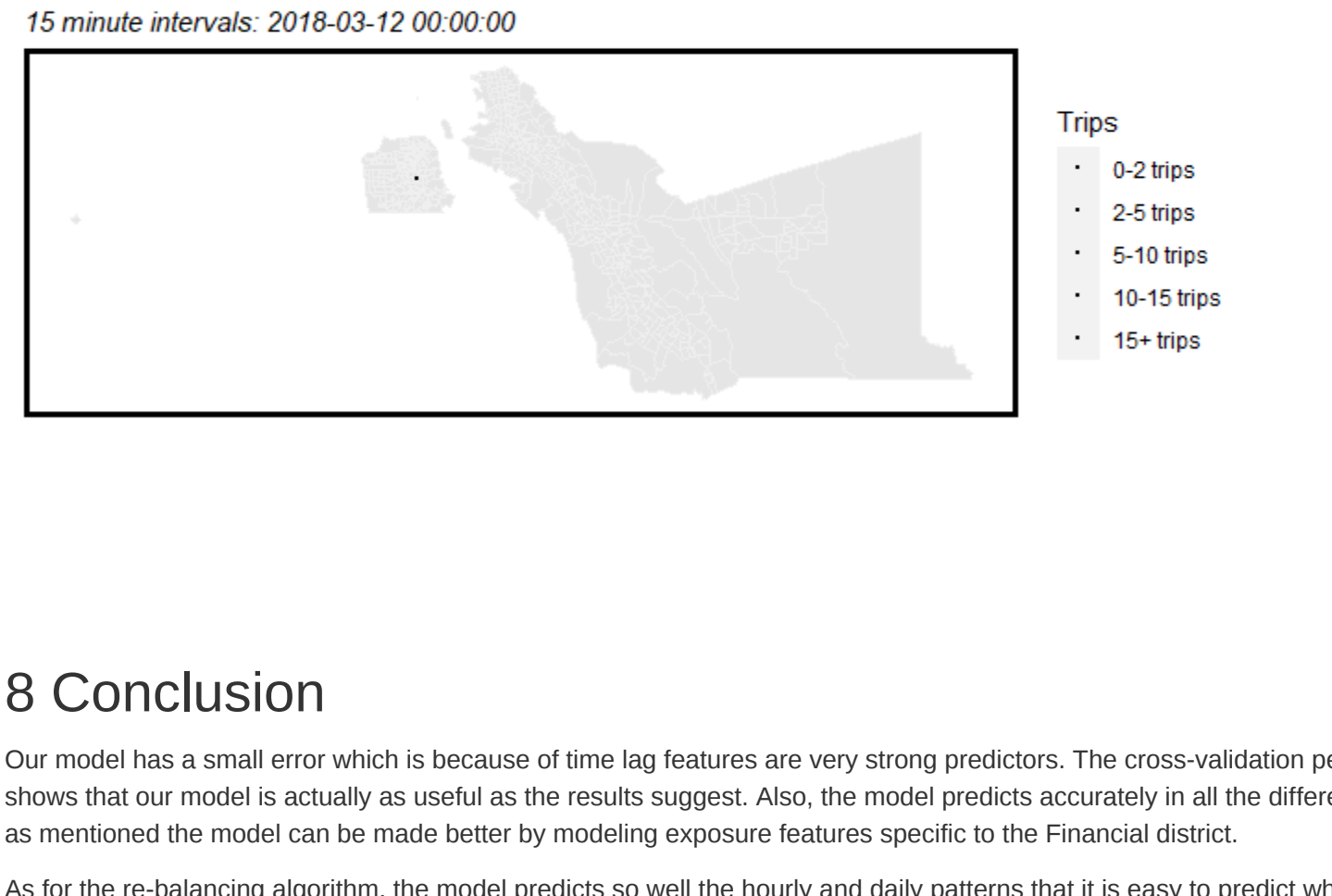




For any model to be accurate, it should be able to predict well in all tracts despite its make. Here we plot the error as a function of income, percentage of white and percentage taking public transit. As you can see the models error increases in tracts where the income is high which can be due to many factors. While the model has a constant error across percent taking transit and white, which says the model does well in all neighborhoods and communities.



Last lets look at the trip count by station as an animation for one week during March.



8 Conclusion

Our model has a small error which is because of time lag features are very strong predictors. The cross-validation performed, and shows that our model is actually as useful as the results suggest. Also, the model predicts accurately in all the different tracts and as mentioned the model can be made better by modeling exposure features specific to the Financial district.

As for the re-balancing algorithm, the model predicts so well the hourly and daily patterns that it is easy to predict when and how is likely to make the trip to a specific destination or from a specific location. Hence, we can use this data to incentivize the user as needed to manage the supply and demand of bikes.