Predicting Citi Bike Trip Demand

Rishi Goutam, Srikar Pamidi, James Goudreault

Predicting Citi Bike Trip Demand

Using a neural network model to predict ridership based on time and weather

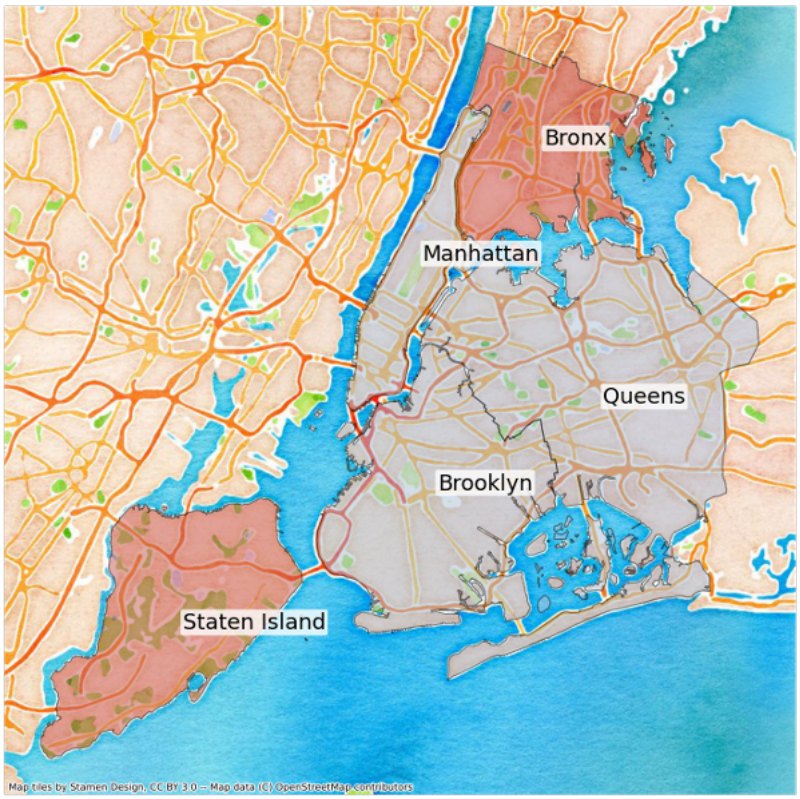
April 7, 2022 – Rishi Goutam, Srikar Pamidi, James Goudreault

Predicting Citi Bike Trip Demand

In this article, we show how we analyzed the Citi Bike dataset and built a model to predict ridership based on seasonality and weather.

Introduction

Citi Bike opened in New York City in 2013Citi Bike operates in multiple areas and has been active for several years. We focused our analysis on the primary NYC boroughs Citi Bike operates in (Manhattan, Brooklyn, and Queens). Unless otherwise specified, statistics and graphics in this article are for the year 2019 and these boroughs



and has since grown in ridership, bikes, and bike dock stations. Predicting demand for bikes is important for Motivate, Citi Bike's parent company, in order to both reduce operating costs and increase ridership. Costs are incurred by having under-utilized bikes on the streets by wear-and-tear on bikes due to exposure to the elements or other forms of damage, so it is necessary to warehouse bikes if they are not serving riders. However, having too few a number of bikes available leads to a poor customer experience and loss of revenue due to dissatisfied customers.

We aim to first understand what features drive demand and then to create a predictive model. Finally, we compare our model against the actual number of trips to evaluate its usefulness.

Analyzing the data

We focused our exploratory data analysis on:

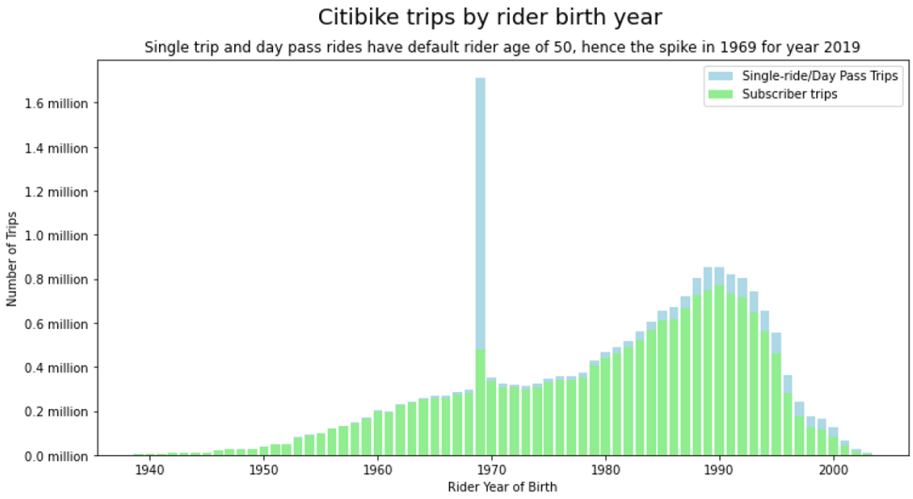
- Rider demographics
- Citi Bike's growth and resilience in the face of COVID-19
- Time (Seasonality)

- Weather

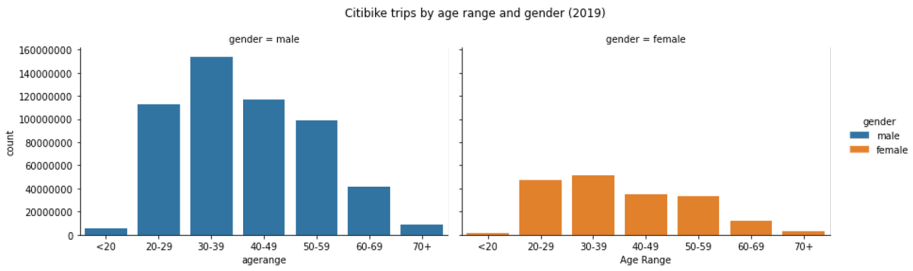
And determined that time and some weather conditions would make for good predictors for a time-series model. Here is that analysis.

Demographics

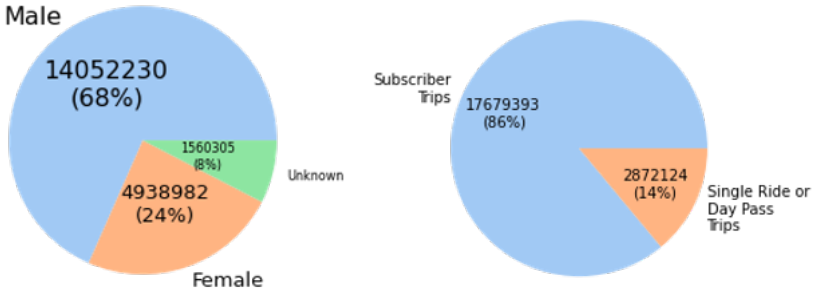
By age, most riders are between 20 and 40 years and are mostly male. Citi Bike has two classes of riders–annual subscribers and single-ride or day pass purchasers. We see a default age of 50 years for riders purchasing one-off trips or passes in the chart below.



TODO FIX THE Y AXIS HERE:

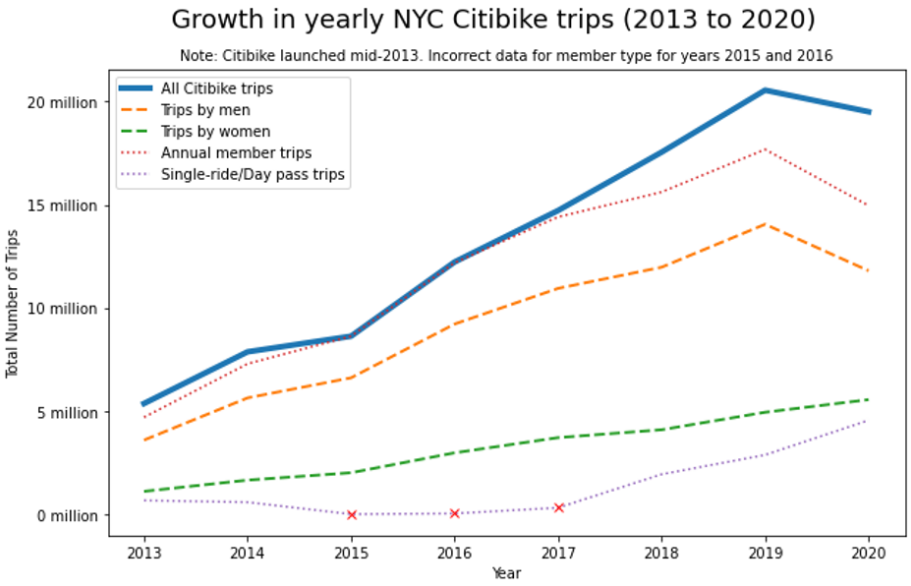


We can see the gender and customer type distribution through the much-maligned pie chart

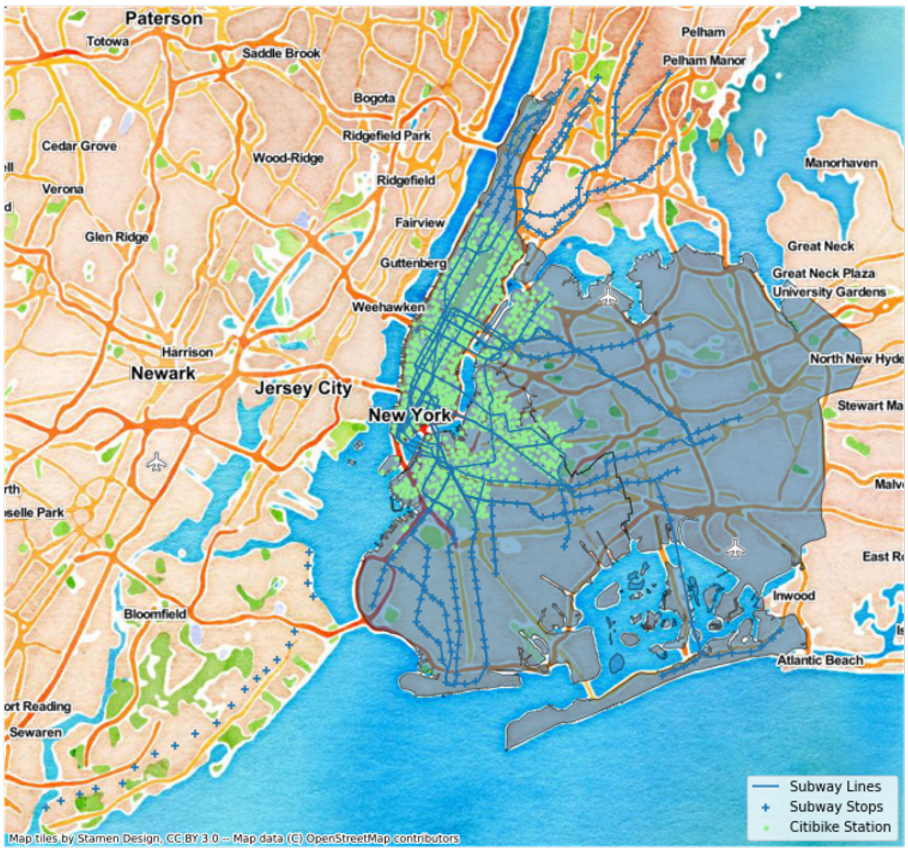


Growth and Resilience of Citi Bike

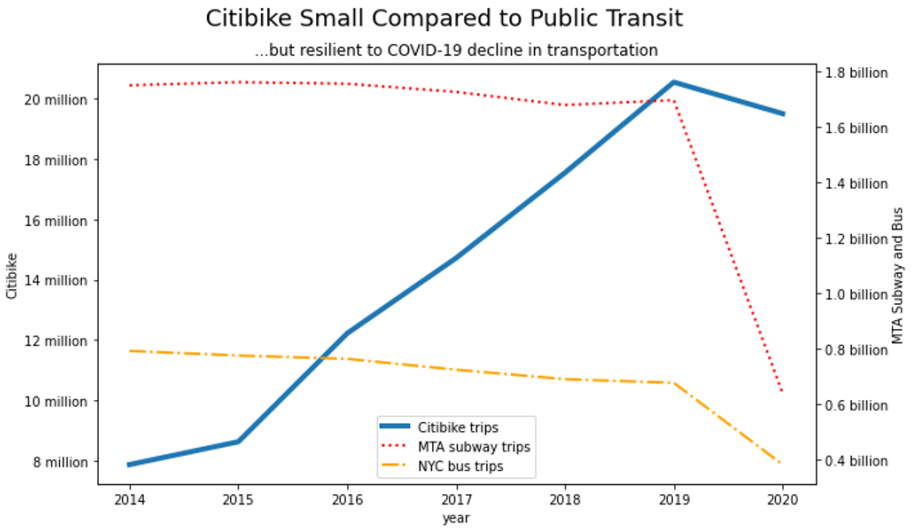
As the number of Citi Bike trips grows, operational efficiency is more important to company finances. In addition, there is need for accurately predicting demand and rebalancing stations effectively



Bike stations appear to expand along subway lines…potential for further expansion into Brooklyn, the Bronx, or Queens?



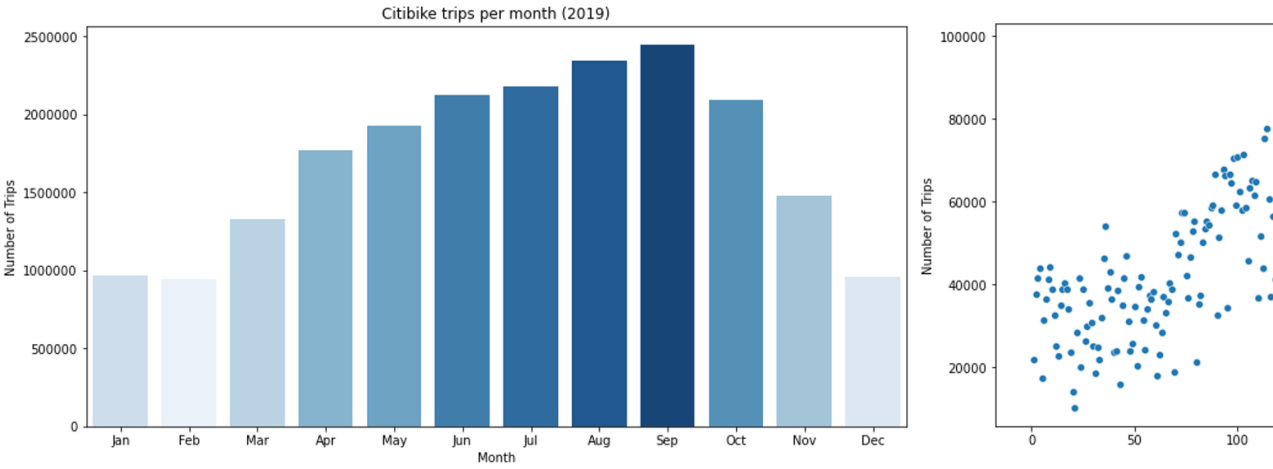
While Citi Bike is not used as much as mass transit in NYC, it offered a way for city residents to move from A to B during the pandemic that wasn't in an enclosed space…perhaps this is why it didn't see as sharp a drop in demand compared to the NYC subway or buses



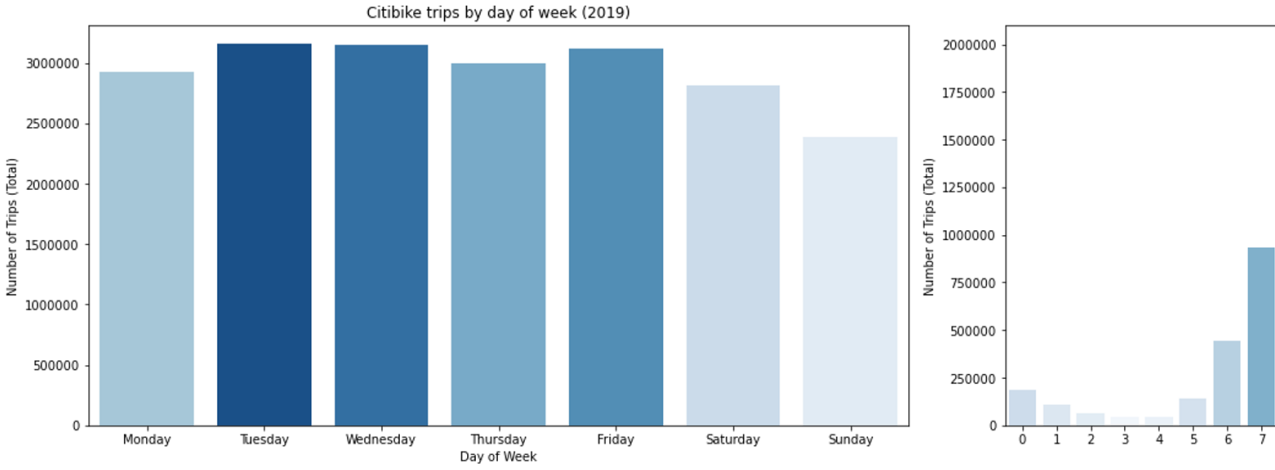
Temporal Analysis

We see increased usage in the summer as one might expect, but not all summer days prove to have high counts.

Labor Day 2019 shows reduced demand...and weather might also play an effect. We examine that later.



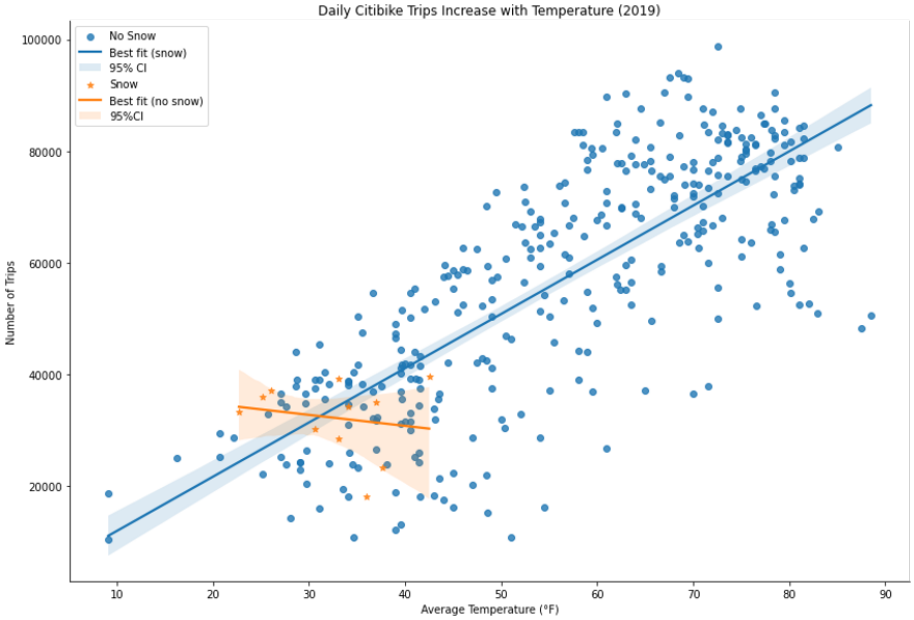
Surprisingly, weekends, especially Sunday, seem to have lower trip counts on average than weekday. Sunday truly is the day of rest.



Looks like commuters make up a bulk of trips given the high trip counts around 8am and 5pm. This also explains the reduced number of trips on weekends.

Weather

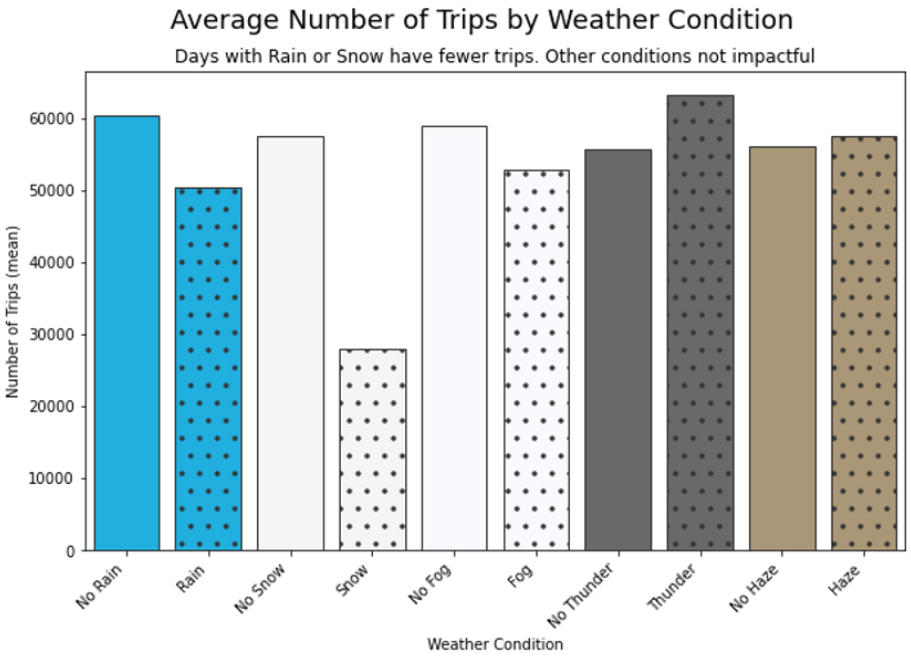
We began by looking at daily average temperature and found that trip demand is highly correlated with it. We can use this as a model predictor!



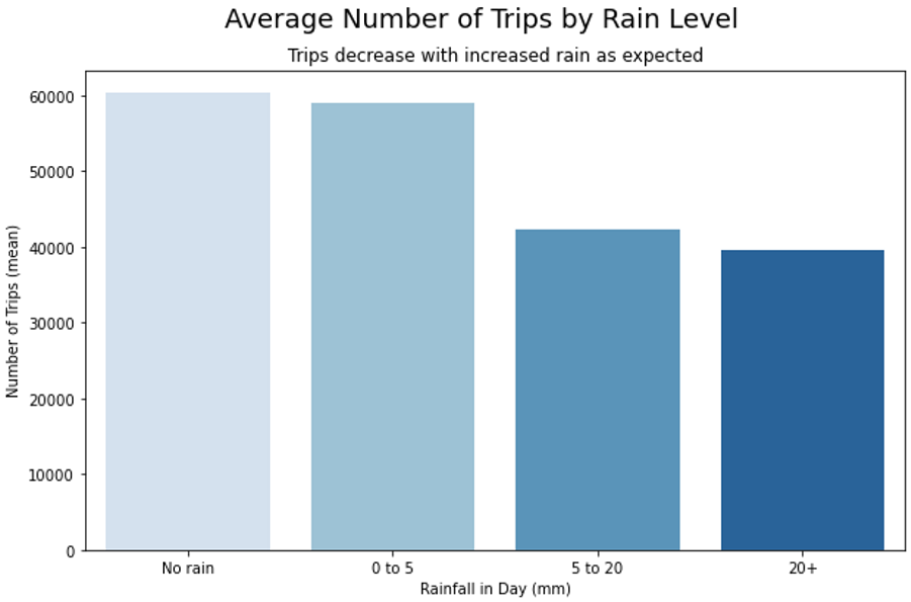
We wanted to investigate whether weather conditions might have an effect on number of trips...however, only saw decrease for days with precipitation

(rain/snow)

Conditions like fog (all types), thunder, or haze were low in terms of number of days and did not have as strong an effect (although we were surprised by the positive effect)



Digging deeper into rain, we wanted to see if the amount of rain mattered…and it does! (as expected)



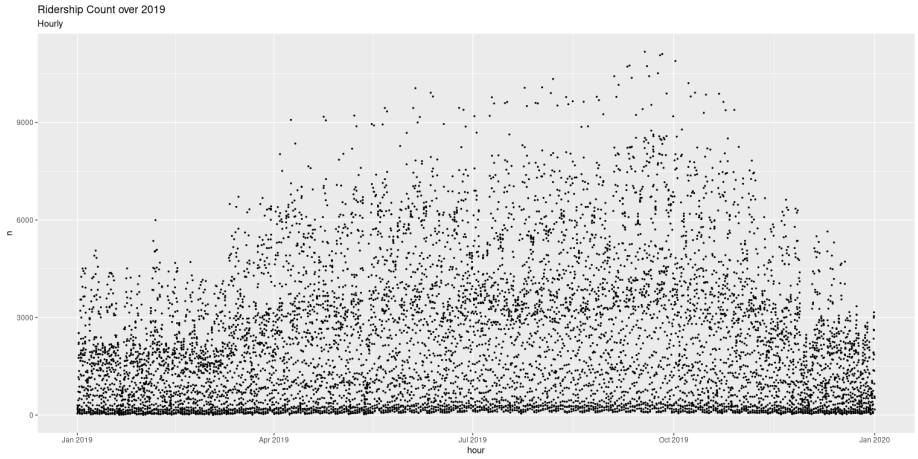
Based on this analysis, we decided to create a model incorporating time, average temperature, and amount of precipitation in order to predict the number of trips.

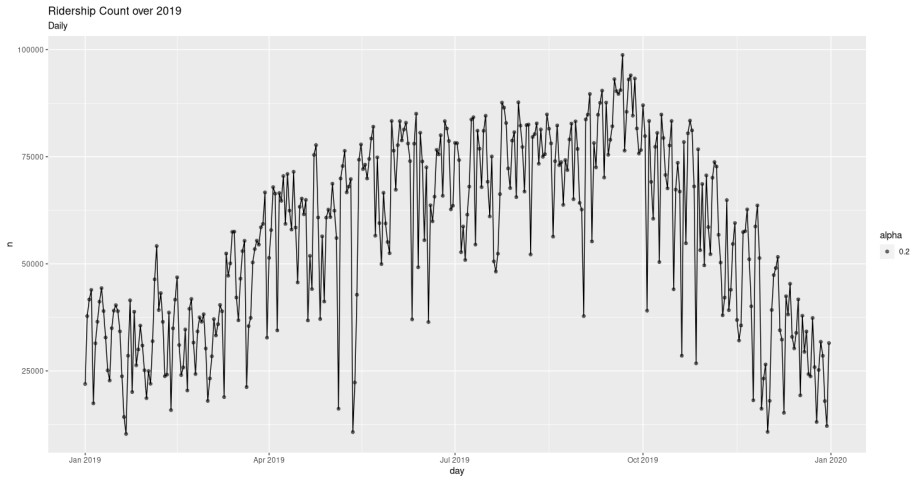
Trip Demand Prediction Models

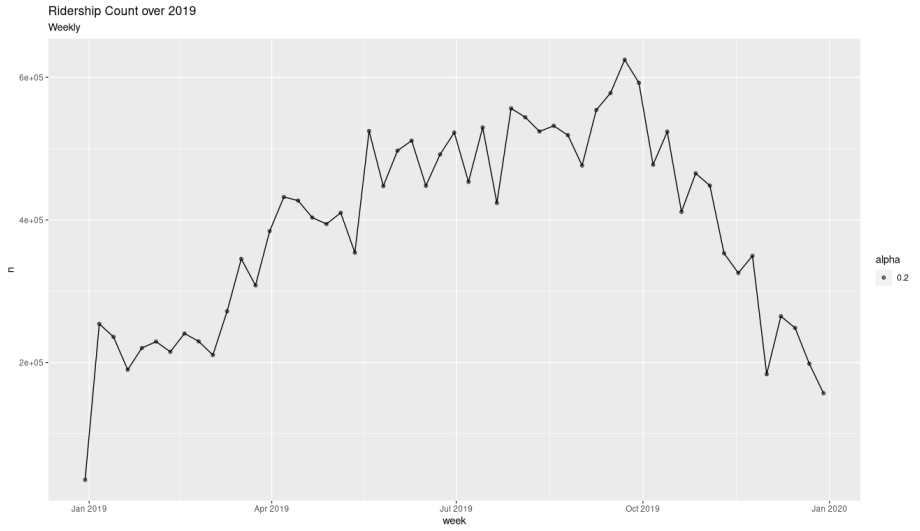
We attempted two models, the first of our models is the traditional SARIMA model, the second was a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN). In this, we further distinguish the models by the time resolution, and whether or not the model was including weather data (i.e. had multidimensional inputs). Notably absent from our treatment is a vectorized SARIMA model or SVARIMA; however, this was due to time considerations and the fact that LSTM-RNN models on average demonstrate far better performance on multidimensional time series than traditional models and approaches.

The population data is found in in the seconds resolution; however, at this resolution, observations are not continously present and we have long gaps for many seconds of the year with zero ridership. We take a look at the hourly ridership data from 2019; whatever model we create for one year we can easily generalize to multiple years.

As a preliminary assesment of the data, we look at the ridership over the year along different timescales–in doing so we can begin to understand trends and seasonality relationships within the day. Changing the scale of our data, as we will see, is akin to passing the "wave" of our time series through a low-pass filter, giving us lower frequency relationships. Therefore, if there is high seasonality at low timescale, this will be erased as we increase our timescale and aggregate over larger steps.


Ridership Count over 2019
Hourly
9000
6000
3000
0
n
Jan 2019
Apr 2019
Jul 2019
Oct 2019
Jan 2020
hour


Ridership Count over 2019
Daily
100000
75000
50000
25000
n
Jan 2019
Apr 2019
Jul 2019
Oct 2019
Jan 2020
day
alpha
0.2



We can see from the above graphs that weekly resolution is far too sparse to caputure meaningful relationships. Therefore, we would like to build models that predict at the Hourly timescale if we can, and if not, then use the Daily timescale

At the sub hourly timescale, the data became too unweildy and noisy for a years worth, let alone for the many years of data Citibike has available. However in future extensions of this project we would like to take a second level resolution for one week for one station and predict the ridership at that level.

Our models were thus:

1. Hourly SARIMA, which did not converge to parameters.
2. Daily SARIMA, which converged to parameters, but had low resolution.
3. Daily LSTM-RNN, without weather, which had high RMS error.
4. Daily LSTM-RNN with weather, which had reduced RMS error.
5. Hourly LSTM-RNN without weather, which had great success.
6. Hourly LSTM-RNN for a specific station, with weather.

We compared these models by producing the Daily and Hourly RMSE and comparing them to find the RMSE minimizing model.

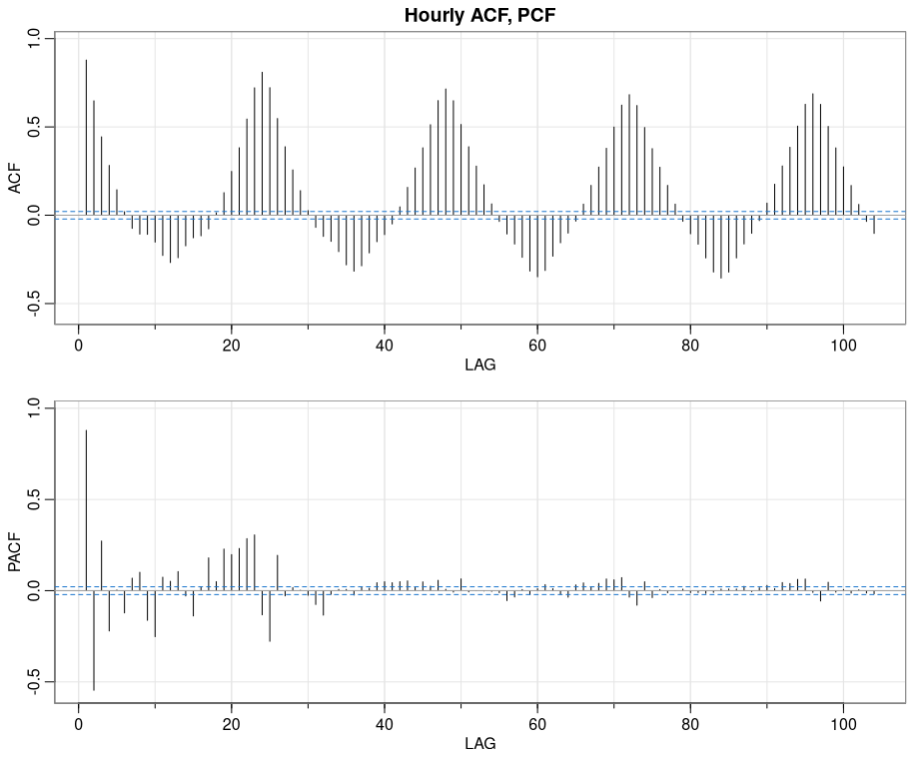
Seasonal-differencing autoregressive integrated moving average (SARIMA)

Hourly SARIMA

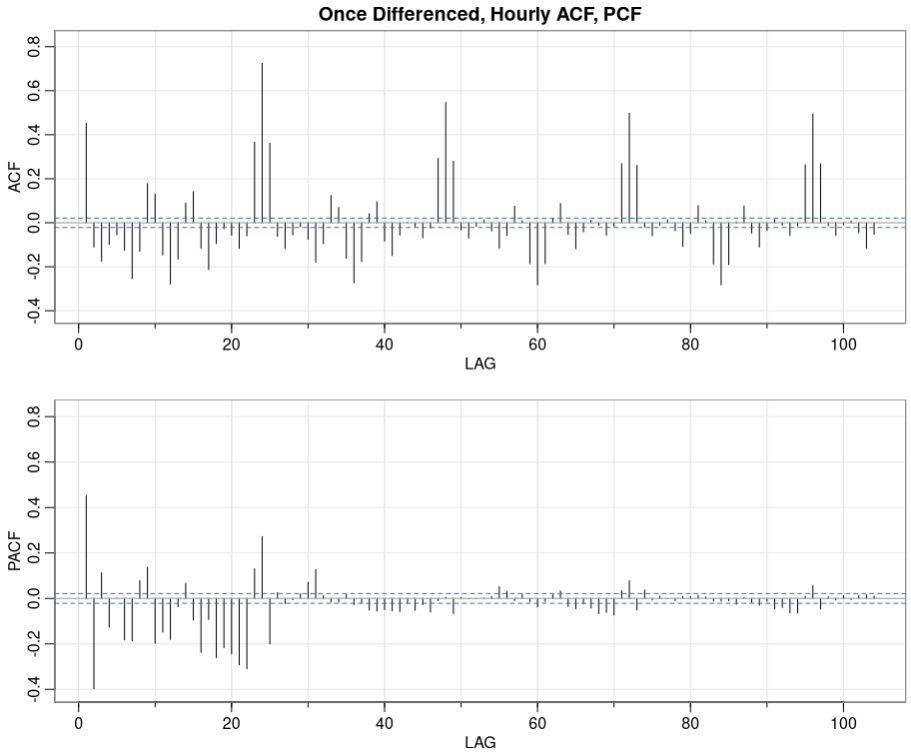
To begin, we start with a model that does not account for the weather, or uses deep learning, only traditional statistical metrics based on past data.

Differencing

Looking at the Autcorrelation and Partial Autocorrelation Function plots (ACF/PACF) we can find the amount of autocorrelation given various lags of the data.



Looking at this hourly ACF/PACF plot we see that there is something very strange occuring. While there is clear seasonality and periodicity, the autocorrelation becomes negative periodically as well. Furthermore, the Partial autocorrelation does not disappear beneath the tolerance value at large lags and instead we see that there is still significant new contribution to the autocorrelation function even at large lags. This does not bode particularly well for using the SARIMA approach. Let's see if differencing helps remove this behavior.



Looking at the once and twice differenced PACF plots, we can see that differencing twice creates a large amount of negative residue in our PACF plot artificially. This means using None or Once differenced data may yield a good model. We'll go with singly differenced data. This doesn't bode well for the model, however, since our transformed data ACF/PACF does not look close to the ACF/PACF of white noise, as it would in a good ARIMA model fit.

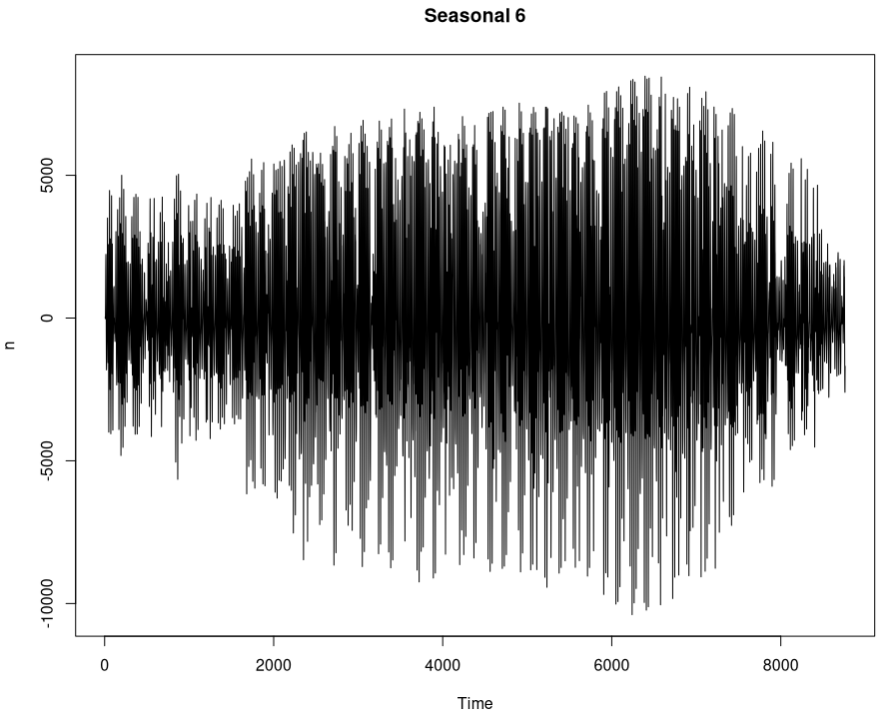
Box-Ljung

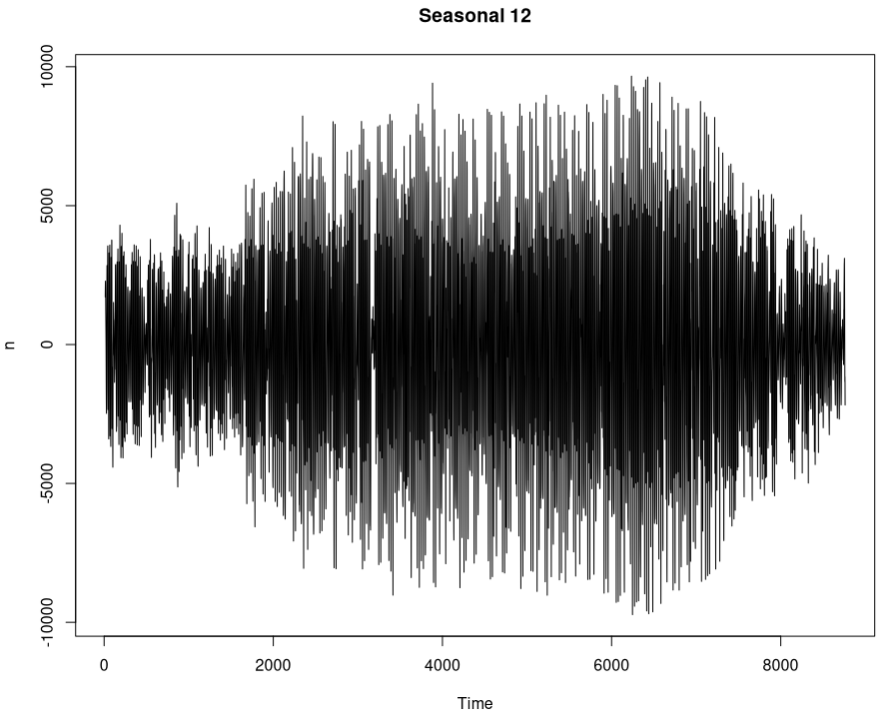
```
d %>% Box.test(type="Ljung-Box", lag = log(length(d)))

    Box-Ljung test

data:  .
X-squared = 13310, df = 9.0781, p-value < 2.2e-16

> d.d %>% Box.test(type="Ljung-Box", lag = log(length(d.d)))

    Box-Ljung test

data:  .
X-squared = 3411.1, df = 9.078, p-value < 2.2e-16
```

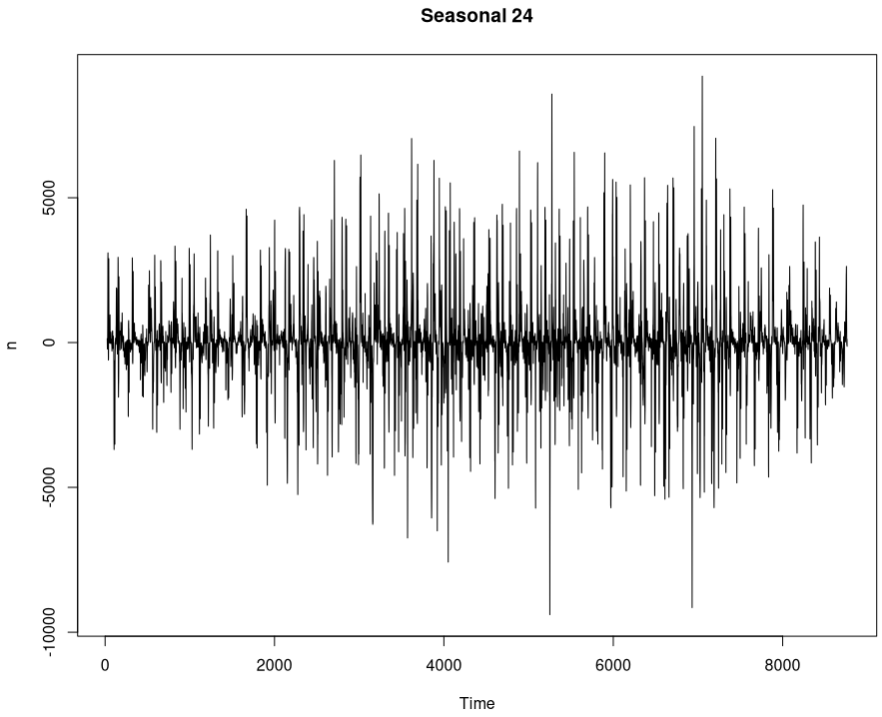
Doing the Box-Ljung test on both no differencing and single diferencing gives us a significant Box-Ljung Score, so we are free to proceed building the ARIMA model. Now we add the seasonality to our ARIMA model.

Seasonal Differencing

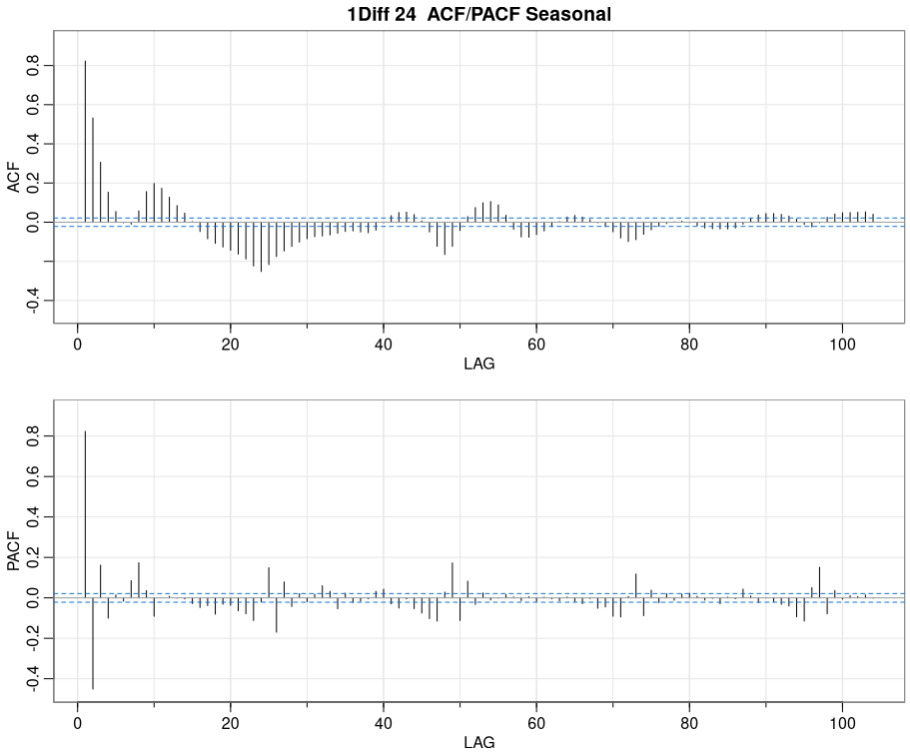
We use different seasonal differencing intervals (6hrs, 12hrs, 24hrs) and choose as our period that interval that leaves the normalized ridership data looking closest to white noise.

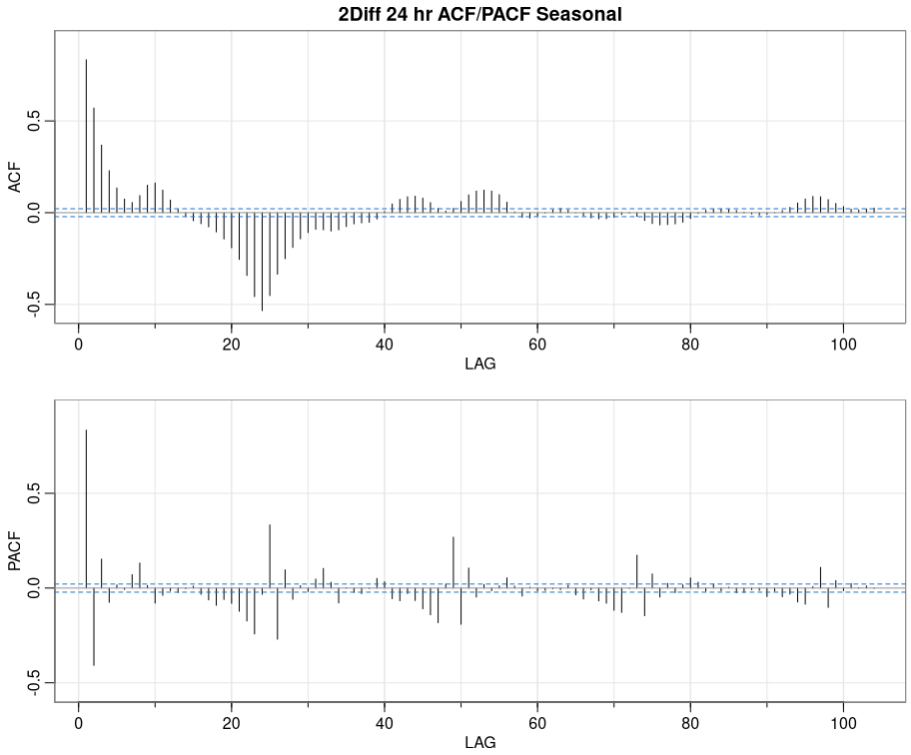



Seasonal 12
n
10000
5000
0
-5000
-10000
0
2000
4000
6000
8000
Time



24 hour seasons (Daily) have the clearest effect on the data. Looking at the ACF/PACF graphs of using seasonal differencing show a better result, but there are still large irregularities, and the autocorrelations do not dissapate after large lags.


1Diff 24 ACF/PACF Seasonal
ACF
PACF
LAG
LAG



The twice differenced data seems only marginally better than the first, and adds more negative residue to the model. Therefore we stick to single differencing

Fitting

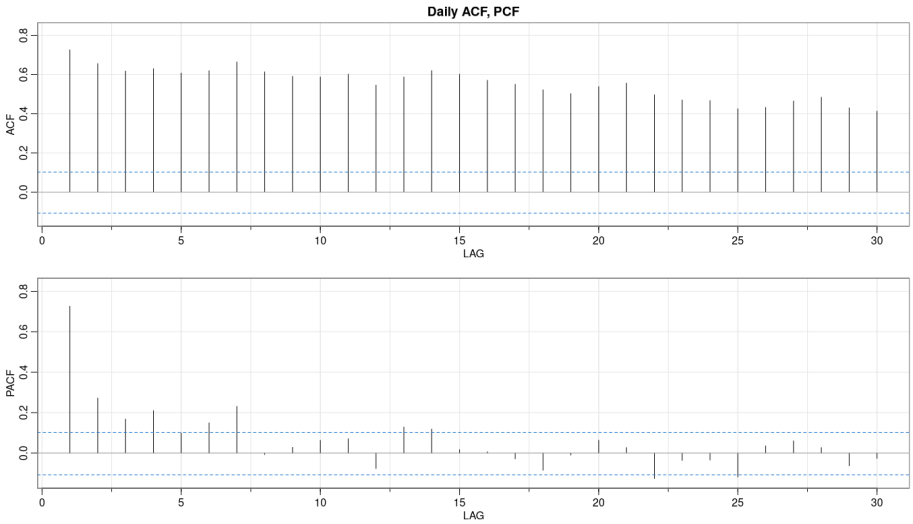
```
#d = 1 , normal differencing
#qmax = 4 , ACF gives us qmax
#pmax = 5 , PACF gives us pmax

#2ce differenced, 24 month seasons, with q_max_seas= 13, p_max_seas = 4

#_____________________________________SARIMA Full

set <- data %>% count(hour) %>% ungroup() %>% select(n) %>% ts()
d = 1
DD = 2
p_max = 2
q_max = 7
p_s_max = 2
q_s_max = 2
per = 24
for (p in 1:p_max) {
```

```
  for (q in 1:q_max) {
    for (p_seasonal in 1:p_s_max) {
      for (q_seasonal in 1:q_s_max) {
        if (p + d + q + p_seasonal + DD + q_seasonal <= (p_max + q_max + p_s_max +
                                                         q_s_max + d + DD)) {
          model <-
            arima(
              x = set,
              order = c((p - 1), d, (q - 1)),
              seasonal = list(order = c((p_seasonal - 1), DD, (q_seasonal - 1)
              ), period = per)
            )
          pval <-
            Box.test(model$residuals, lag = log(length(model$residuals)))
          sse <- sum(model$residuals ^ 2)
        }
      }
    }
  }
}
```

We construct our model, as shown above, and search through various combinations of parameters to find one that passes our residual p-test (i.e. does not have any correlations in the residue data; accept null hypothesis), and minimize the AIC complexity of the model.

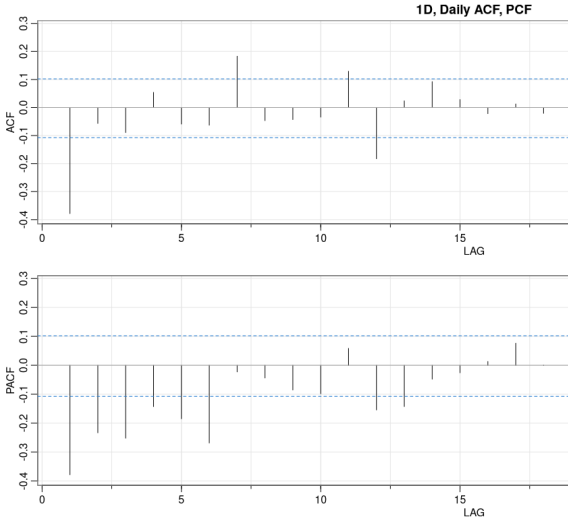
However, these parameters don't converge at all. The model gets stuck trying to calculate even the lowest level of these. We cannot find a model using this, but we shouldn't be too surprised–we had a lot of red flags while analyzing the data. Let's change to the Daily scale and try a SARIMA Model.

Daily SARIMA:

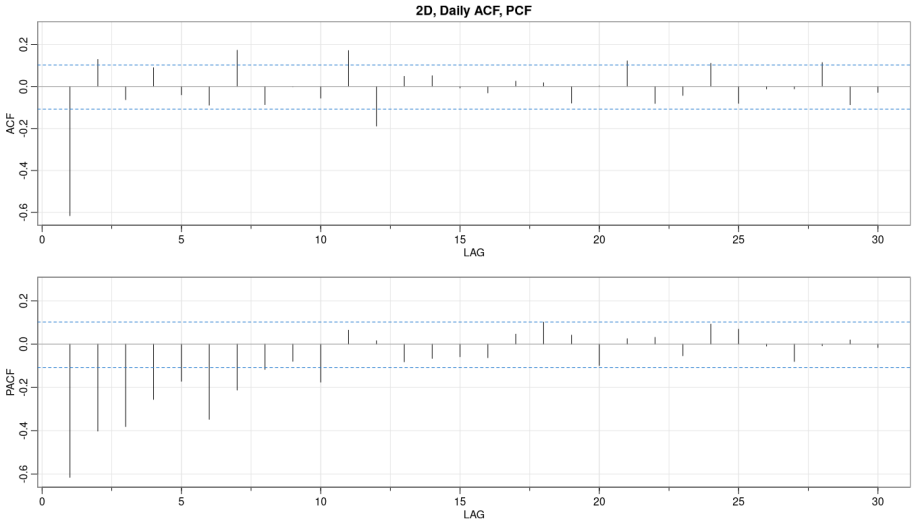


Differencing

Unlike the ACF/PACF plot before, this gives us a seasonality that is reasonable to interpret. We see that we have a constant seasonality in our ACF plot, which has a corresponding vanishing of PACF below the threshold as lags become larger, and as less new Autocorrelation is being found.



We try differencing to remove the seasonality in the ACF plot.



These singly and double differenced plots look close to white noise. However, double differencing introduces a significant amount of negative residue to the data. Therefore, we stick with single differencing.

Box-Ljung

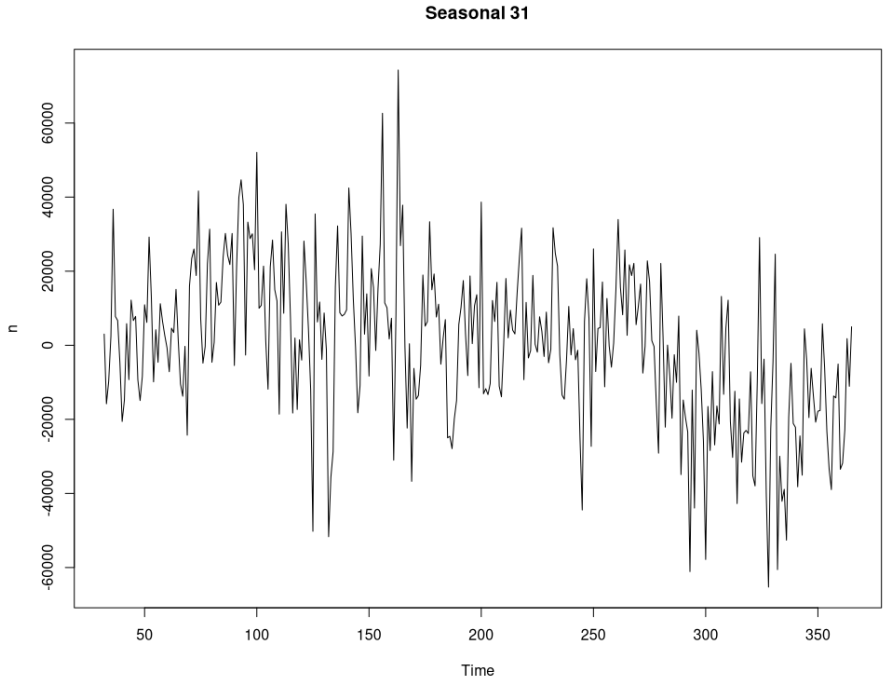
We do the Box-Ljung test to see if our autocorrelations are significant enough with this differencing to justify making a sarima model. We want to reject the null hypothesis that the autocorrelations are not significantly different than those normally distributed around the population mean of autocorrelations.

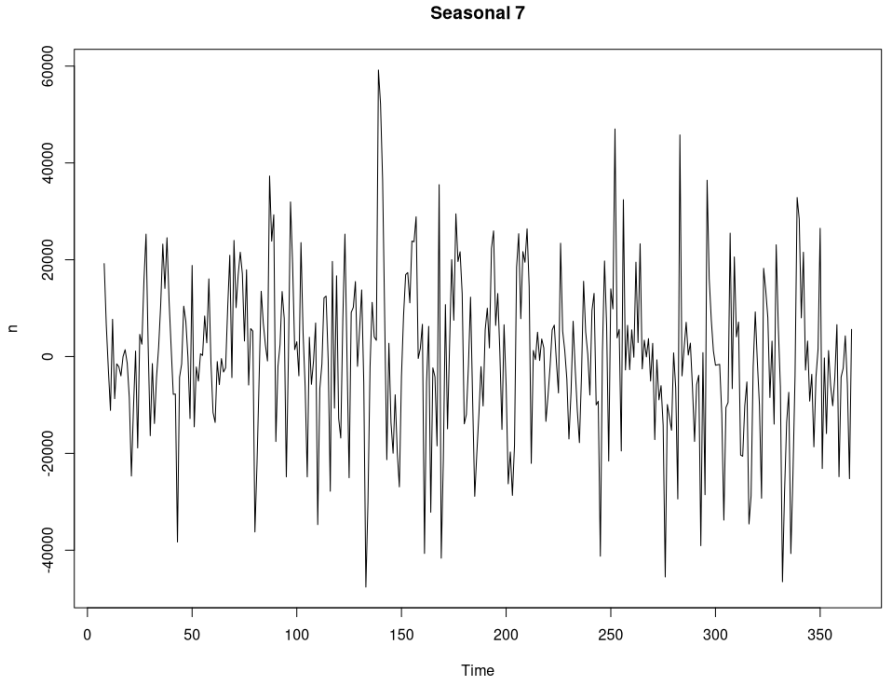
```
d.d %>% Box.test(type="Ljung-Box", lag = log(length(d.d)))
# p = 6.14e-11 for single differencing

    Box-Ljung test

data:  .
X-squared = 59.058, df = 5.8972, p-value = 6.145e-11
```

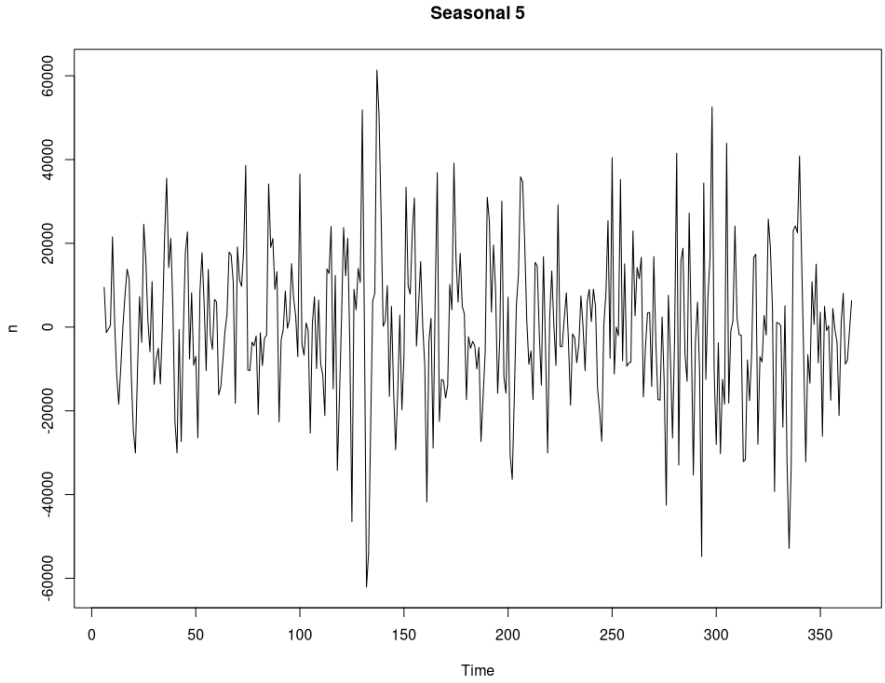
Given this Box-Ljung test result, we can proceed to make our SARIMA model.

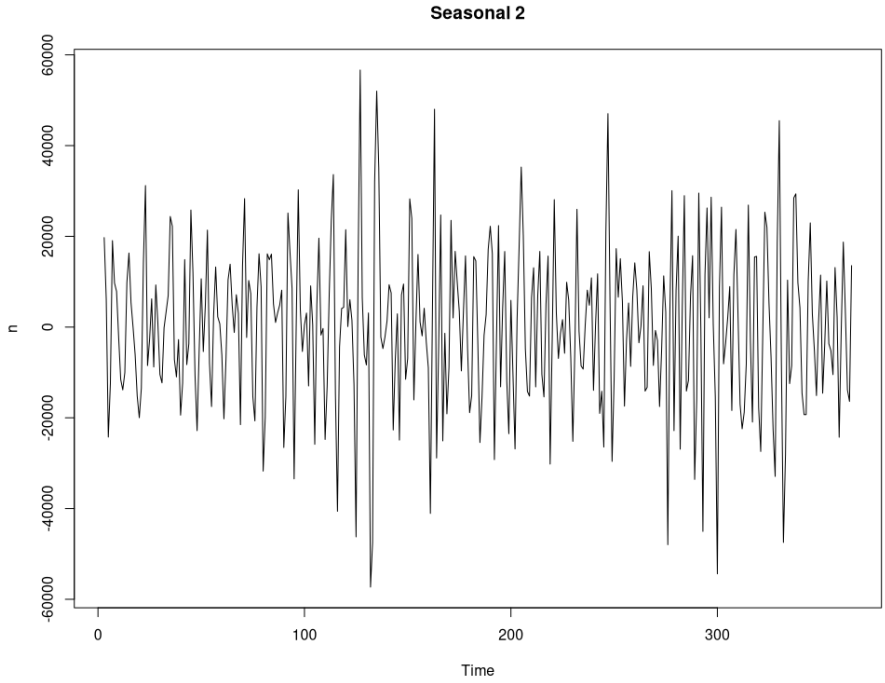
Seasonality

We try 31 day (1 month) seasons, 7 day (1 week), 5 day (1 work week), and 2 day periods. Recall that by seasonally differencing, we are trying to attain data that looks like white noise i.e. is stationary.

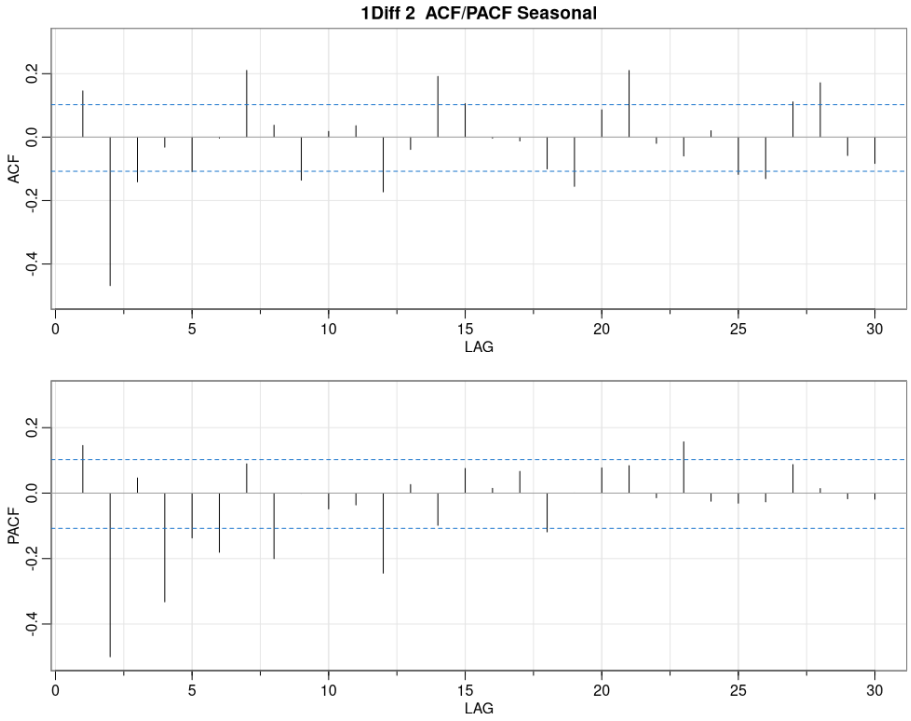

Seasonal 31
n
60000
40000
20000
0
-20000
-40000
-60000
50
100
150
200
250
300
350
Time

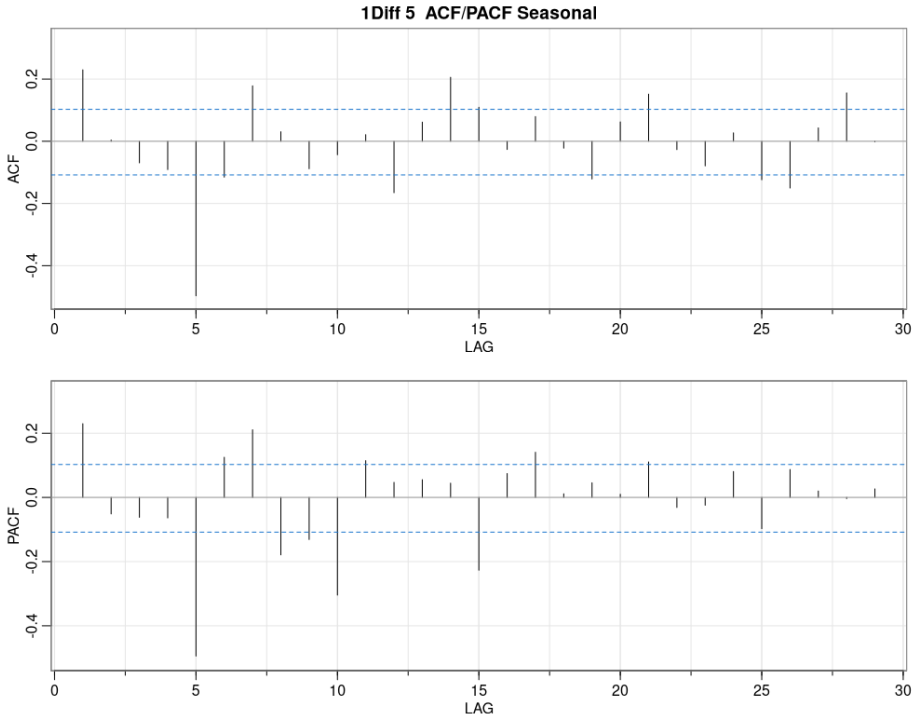

Seasonal 7
n
60000
40000
20000
0
-20000
-40000
0
50
100
150
200
250
300
350
Time

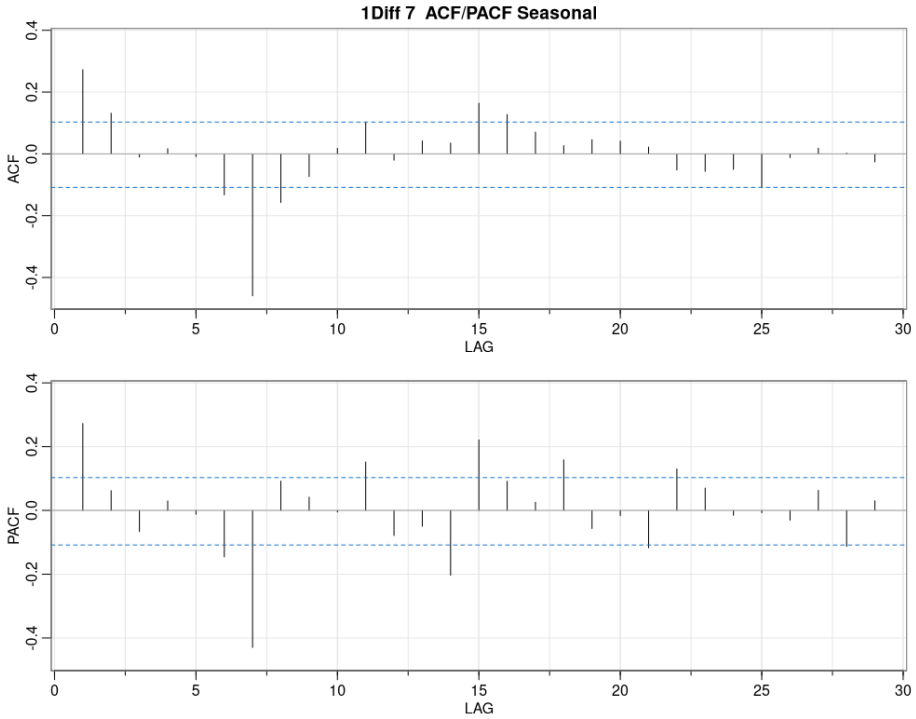

Seasonal 5
n
60000
40000
20000
0
-20000
-40000
-60000
0
50
100
150
200
250
300
350
Time



Looking at the plots, we can see that our best seasonality is found in the two day timeframe, with a close second at 5 and 7 day seasonality. Comparing their ACF/PACF plots below:


1Diff 2 ACF/PACF Seasonal
ACF
0.2
0.0
-0.2
-0.4
0
5
10
15
20
25
30
LAG
PACF
0.2
0.0
-0.2
-0.4
0
5
10
15
20
25
30
LAG


1Diff 5 ACF/PACF Seasonal
ACF
0.2
0.0
-0.2
-0.4
0
5
10
15
20
25
30
LAG
PACF
0.2
0.0
-0.2
-0.4
0
5
10
15
20
25
30
LAG



We can conclude that using seven-day seasonality is probably the best, even though it does not give us the most stationary differencing, because it adds the least additional residue artificially in the PACF plots, which is an indication of overfitting.

Model

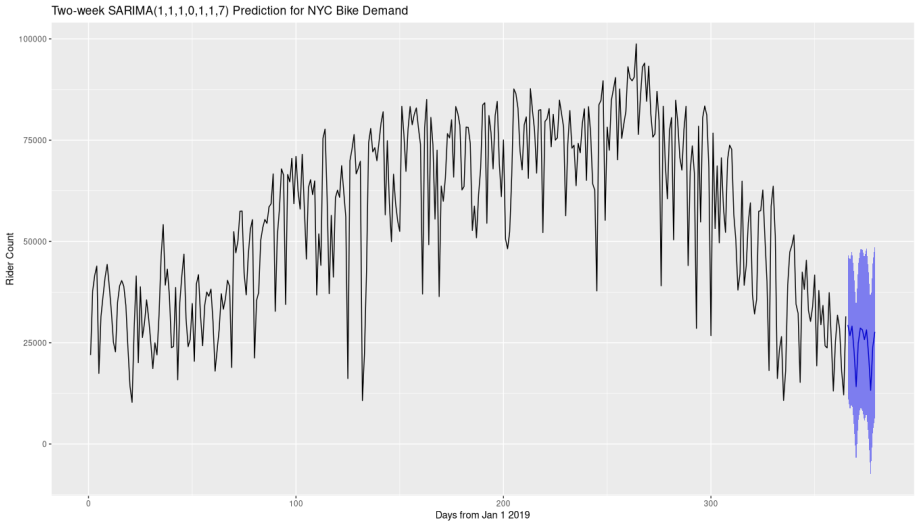
We now have all the information we need to find our optimal model parameters.

```
#d = 1 , normal differencing
#qmax = 2 , ACF gives us qmax
#pmax = 2 , PACF gives us pmax

#1ce differenced, 4 seasons, with q_max_seas= 1, p_max_seas = 1

#_____________________________________SARIMA Full

set <- data %>% count(day) %>% ungroup() %>% select(n) %>% ts()
d= 1
DD= 1
p_max= 2
q_max=3
p_s_max=2
q_s_max=2
```

```
per= 7
for(p in 1:p_max){
  for(q in 1:q_max){
    for(p_seasonal in 1:p_s_max){
      for(q_seasonal in 1:q_s_max){
        if(p+d+q+p_seasonal+DD+q_seasonal<=(p_max+q_max+p_s_max+q_s_max+d+DD)){
          model<-arima(x=set, order = c((p-1),d,(q-1)), seasonal = list(order=c((p_seasonal-
          pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
          sse<-sum(model$residuals^2)
          cat(p-1,d,q-1,p_seasonal-1,DD,q_seasonal-1,per, 'AIC=', model$aic, ' SSE=',sse,' p
        }
      }
    }
  }
}
```

After running the loop above, we receive the results below: we can only choose those that have p-values which make us reject the alternative hypothesis and accept the null hypothesis, for these models have no significant trends in their residuals, and have captured the data's patterns well.

```
0 1 0 0 1 0 7 AIC= 8093.003  SSE= 145542442362  p-VALUE= 2.449596e-12
0 1 0 0 1 1 7 AIC= 7900.616  SSE= 78139054119  p-VALUE= 1.415534e-13
0 1 0 1 1 0 7 AIC= 8019.592  SSE= 1.17339e+11  p-VALUE= 8.992806e-15
0 1 0 1 1 1 7 AIC= 7900.593  SSE= 77926049526  p-VALUE= 2.563505e-13
0 1 1 0 1 0 7 AIC= 7965.833  SSE= 100341574312  p-VALUE= 2.174389e-05
0 1 1 0 1 1 7 AIC= 7766.01  SSE= 53256832791  p-VALUE= 0.1580151
0 1 1 1 1 0 7 AIC= 7873.321  SSE= 76650546235  p-VALUE= 0.002042164
0 1 1 1 1 1 7 AIC= 7768.009  SSE= 53333608968  p-VALUE= 0.1579503
0 1 2 0 1 0 7 AIC= 7945.996  SSE= 94400139286  p-VALUE= 0.2509538
0 1 2 0 1 1 7 AIC= 7762.23  SSE= 53386356470  p-VALUE= 0.7210931
0 1 2 1 1 0 7 AIC= 7861.747  SSE= 73799152343  p-VALUE= 0.1976797
0 1 2 1 1 1 7 AIC= 7764.062  SSE= 53263786824  p-VALUE= 0.7258721
1 1 0 0 1 0 7 AIC= 8031.964  SSE= 121922910749  p-VALUE= 7.977976e-08
1 1 0 0 1 1 7 AIC= 7830.021  SSE= 63723886594  p-VALUE= 2.929021e-06
1 1 0 1 1 0 7 AIC= 7947.96  SSE= 95357172998  p-VALUE= 3.547598e-07
1 1 0 1 1 1 7 AIC= 7831.23  SSE= 63700528510  p-VALUE= 3.495699e-06
1 1 1 0 1 0 7 AIC= 7941.309  SSE= 93170173485  p-VALUE= 0.713379
1 1 1 0 1 1 7 AIC= 7761.436  SSE= 53381920877  p-VALUE= 0.821252
1 1 1 1 1 0 7 AIC= 7858.621  SSE= 73160746869  p-VALUE= 0.3871591
1 1 1 1 1 1 7 AIC= 7763.162  SSE= 53251964663  p-VALUE= 0.8333338
1 1 2 0 1 0 7 AIC= 7942.432  SSE= 92941926050  p-VALUE= 0.7958765
1 1 2 0 1 1 7 AIC= 7763.211  SSE= 53394595014  p-VALUE= 0.8631816
1 1 2 1 1 0 7 AIC= 7859.434  SSE= 72919303794  p-VALUE= 0.5261746
1 1 2 1 1 1 7 AIC= 7764.888  SSE= 53258078603  p-VALUE= 0.8779478

#SARIMA(1 1 1 0 1 1 7) AIC= 7761.436  SSE= 53381920877  p-VALUE= 0.821252
```



- Converges to parameters
- Prediction and visualization
- Daily RMSE, avg hourly RMSE
- Hypothesis and next step for model

Long short-term memory recurrent neural network (LSTM)

Intro to RNN and LSTMRNN

Moving from SARIMA to LSTMRNN

Timeseries and Backtesting

Daily LSTMRNN with hourly weather,

- EDA and visuals
- Seasonality, ACF, periodicity,
- Structure of Layers and LSTMRNN
- Backtesting and Visualization
- Predictions and visualization
- Daily RMSE and av hourly RMSE
- Hypothesis and next step to improve

Daily LSTMRNN with weather

- Weather EDA, why we think to add temp and precipitation.
- Seasonality ACF periodicity,
- Layerstructure and adding more dimesions for lag
- Predictions and Vis.
- Daily RMSE and av hourly RMSE

- Hypothesis and next step to improve

Hourly LSTMRNN without weather

- Hourly EDA, for busiest station.
- Seasonality ACF periodicity,
- Layerstructure and adding more dimesions for lag
- Predictions and Vis.
- Daily RMSE and av hourly RMSE
- Hypothesis and next step to improve

Hourly LSTMRNN for a specific station, with weather

- Hourly Weather EDA, why we think to add temp and precipitation.
- Seasonality ACF periodicity,
- Layerstructure and adding more dimesions for lag
- Predictions and Vis.
- Daily RMSE and av hourly RMSE
- Hypothesis and next step to improve

Conclusion

Our exploratory data analysis informed our model feature selection and we attempted two time-series models (SARIMA and LSTM) to predict trip demand on both an hourly and daily basis. We found that hourly models were better overall. Further improvements to the model would be in reducing the root mean square error in predictions and, perhaps, incorporating additional predictors.

TODO LINK TO THINGY

Separately, we analyzed bike rebalancing. You can read the full write-up here.