Divvy Bike Weekly Usage Prediction

MSCA 31006 – Time Series Analysis and Forecasting

Group 6

Yue Wu, Chuyu Chen, Howard Lin, Jack Gao

January 6, 2022

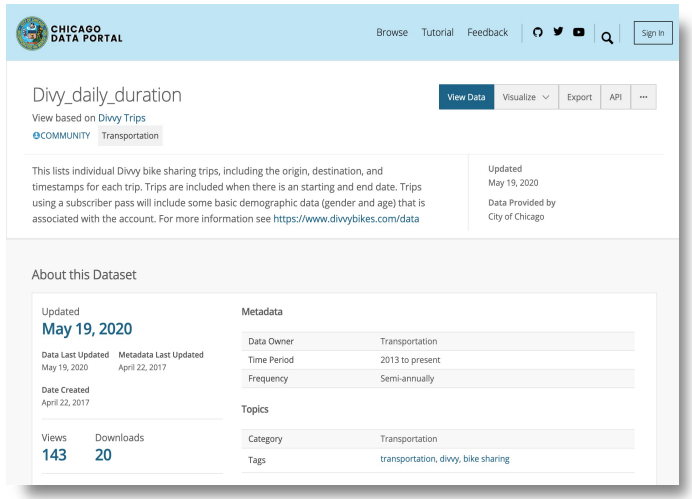
Agenda

1. Data Source & Analysis Goal
2. EDA
3. Modeling & Forecasting
4. Model Selection
5. Summary & Next Steps

1. Data Source & Analysis Goal

Data Source & Analysis Goal

1. Chicago Data Portal

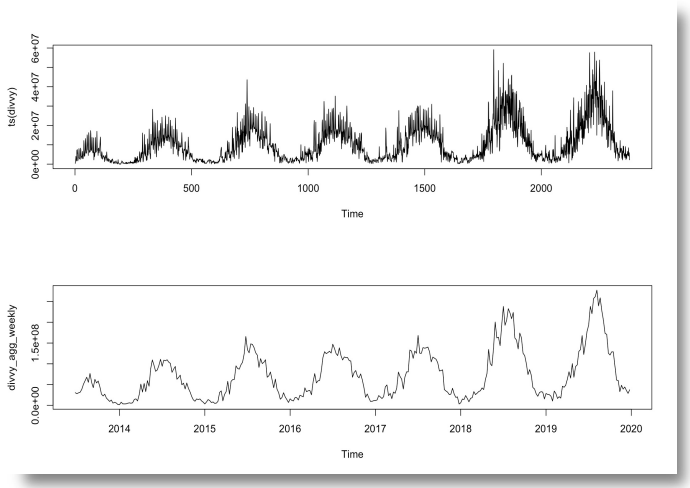


Frequency: Daily total trip duration

Time Period: 6/27/2013 – 12/31/2019

Unit of Measurement: Second

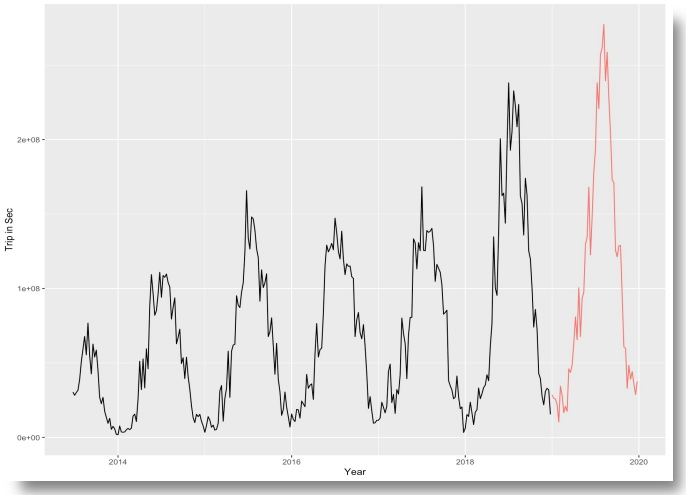
2. Data Processing



Missing Data: Impute 2 missing data with 0

Frequency Transform: Aggregate daily data into **weekly** data.

3. Train-Test Split



Train Period: 6/30/2013 – 12/31/2018 (287 weeks)

Test Period: 1/1/2019 – 12/31/2019 (52 weeks)

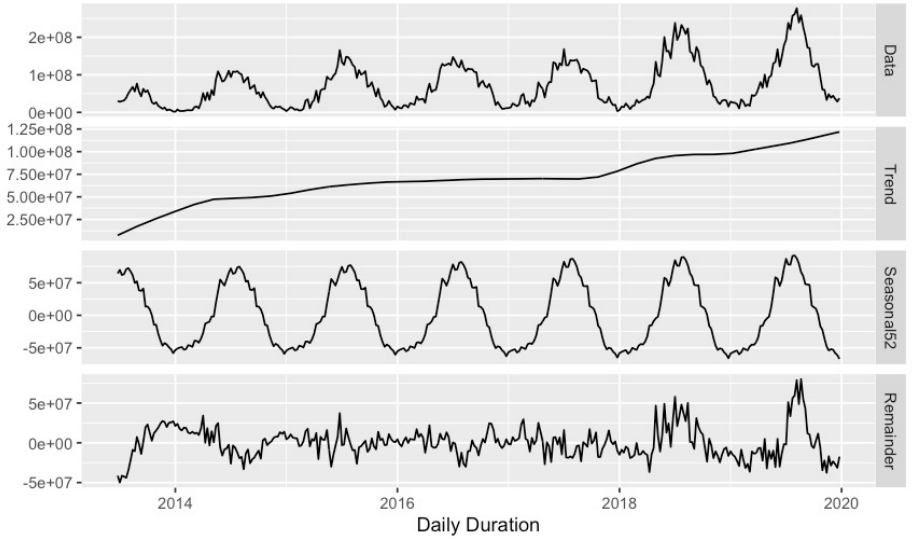
Analysis Goal:

- Forecast the weekly usage of Divvy bike of different models and compare the forecast results
- The goal is to help Divvy Bike maximize profits by forecasting the demand of bike sharing programs and better optimize the bike usage, placement, and profitability
- In addition, we hope to measure the bike transportation activity and have a better understanding of the mobility in a city

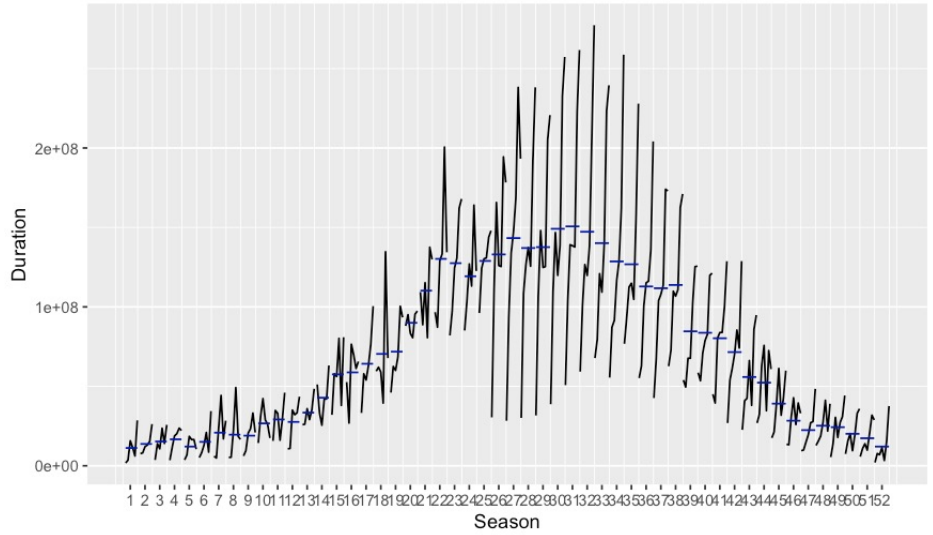
2. Exploratory Data Analysis

Exploratory Data Analysis

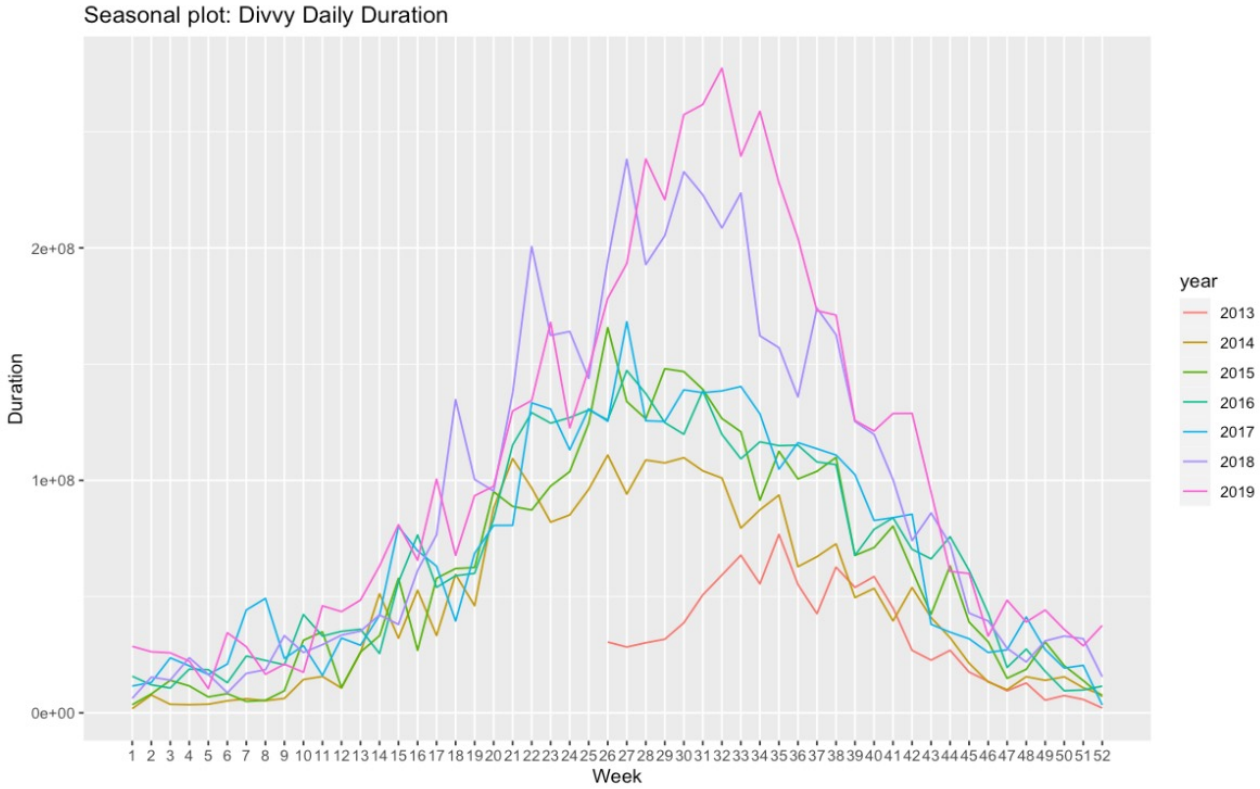
Multiple seasonal decomposition plot of Divvy weekly duration



Seasonal subseries plot of Divvy weekly duration



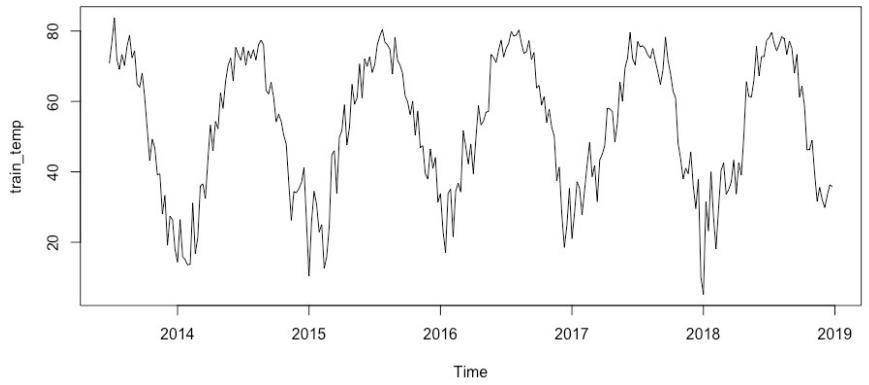
Plot of Divvy daily trip duration by year



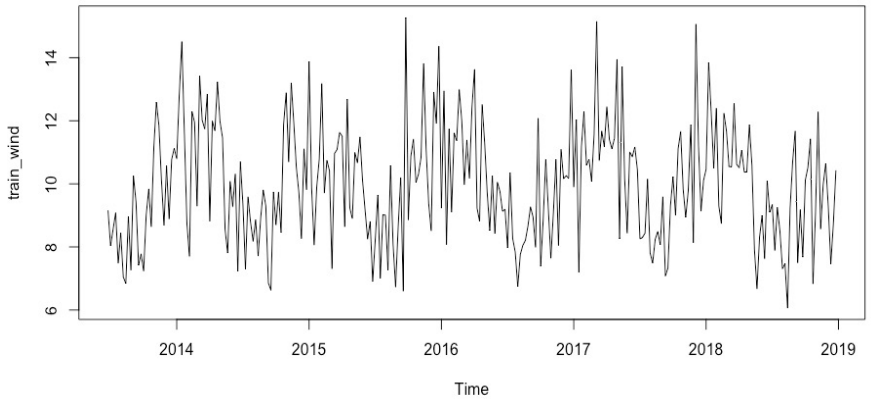
Exploratory Data Analysis

Other variables and respective correlations

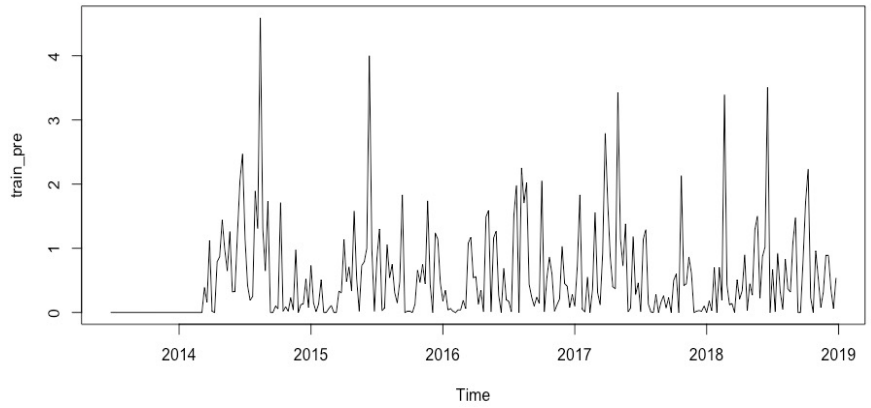
1. Weekly Average Temperature (corr = 0.833)



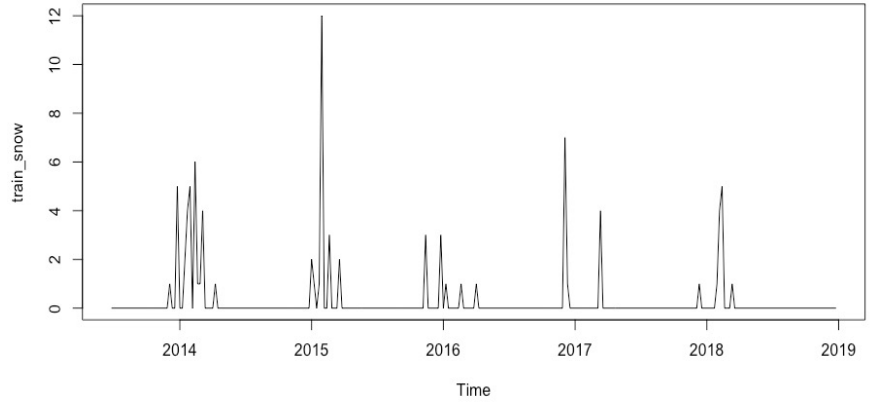
2. Weekly Average Windspeed (corr = -0.479)



3. Weekly Total Precipitation (corr = 0.168)

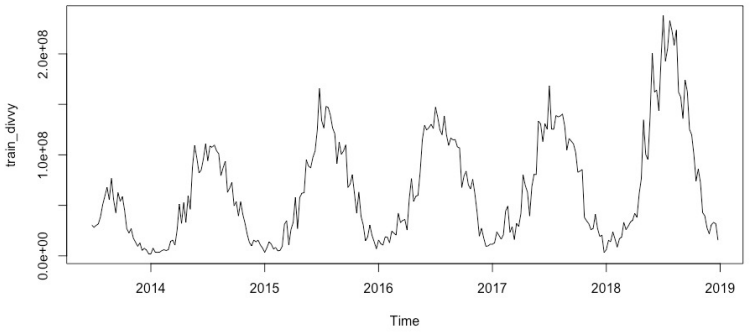


4. Weekly Total Snow Depth (corr = -0.263)

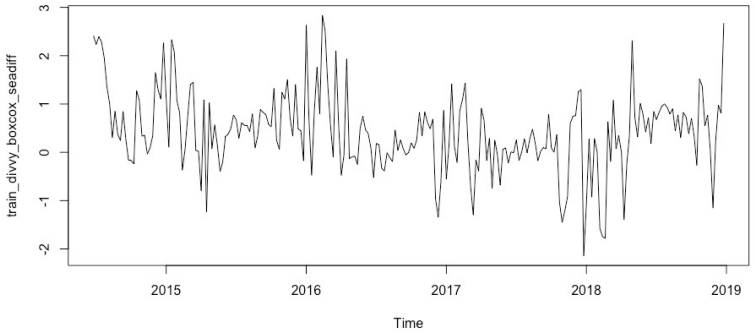


Exploratory Data Analysis

BoxCox, Seasonal Differencing, Trend Differencing

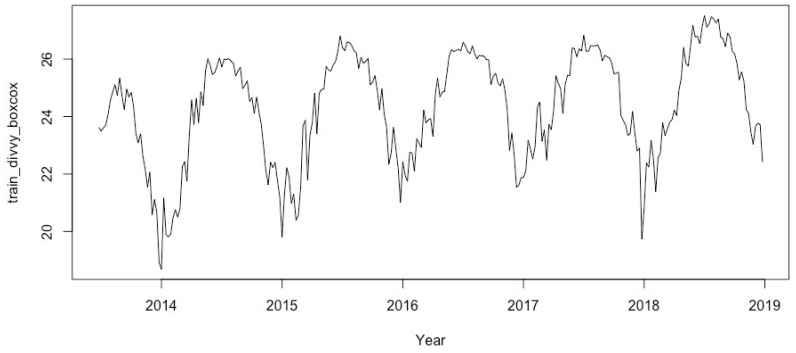


1. Original Data
- Increasing variance
- Seasonality
- Trend not stationary

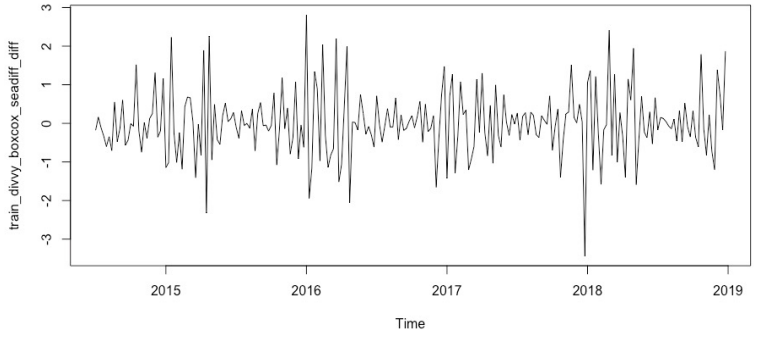


3. After Seasonal Diff (lag = 52)
- Constant variance
- No Seasonality
- Trend not stationary

2. After BoxCox (lambda = 0.035)
- Constant variance
- Seasonality
- Trend not stationary



4. After Diff (lag = 1)
- Constant variance
- No Seasonality
- Trend stationary



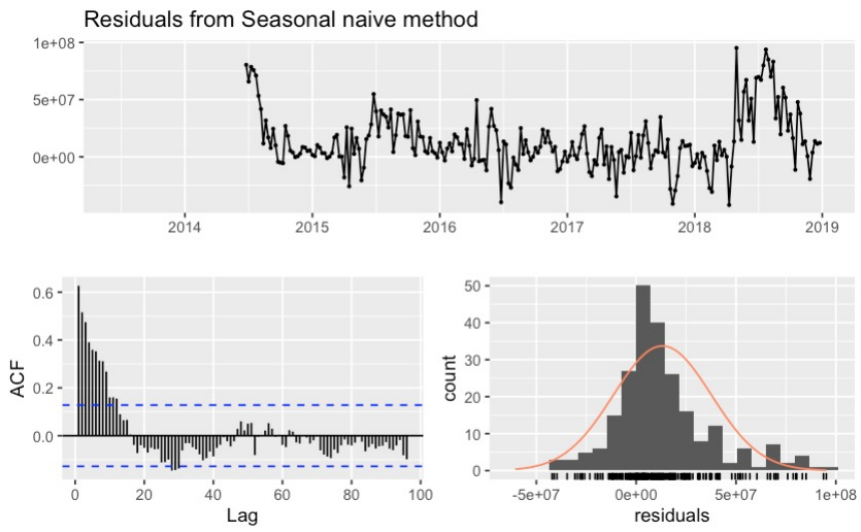
3. Modeling & Forecasting

Modeling & Forecasting

Seasonal Naïve (Benchmark)

Model = snaive

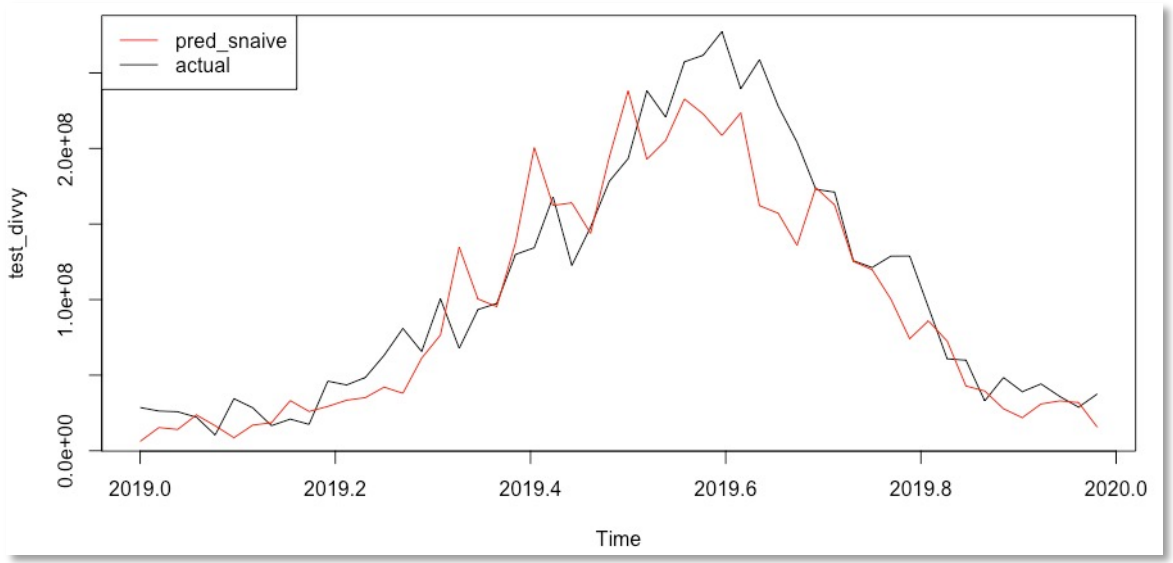
- **AICc:** N/A
- **Residuals: Not** White Noise




```
        Ljung-Box test

data:  Residuals from Seasonal naive method
Q* = 452.06, df = 57, p-value < 2.2e-16

Model df: 0.   Total lags used: 57
```

Forecast from snaive

- **Test MAPE:** 0.2752

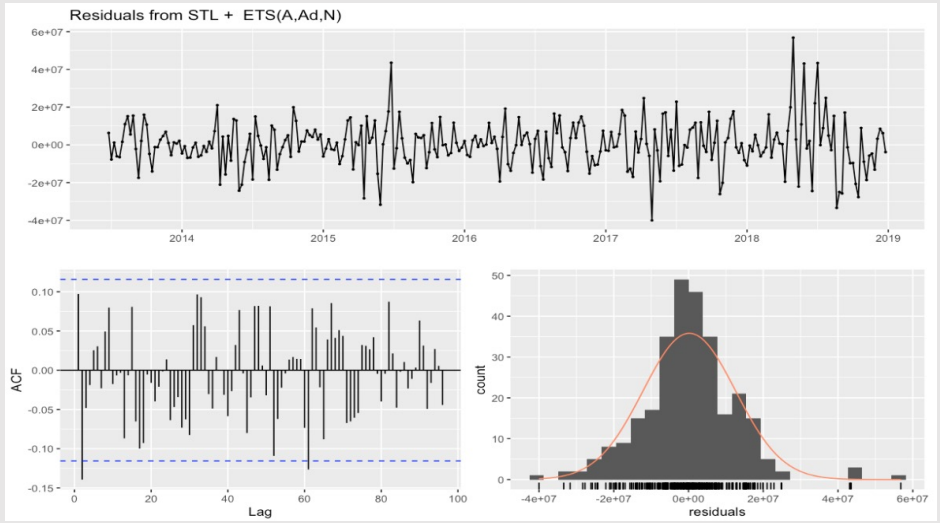


Modeling & Forecasting

Exponential Smoothing Model

Best Model = STL + ETS(A,Ad,N)

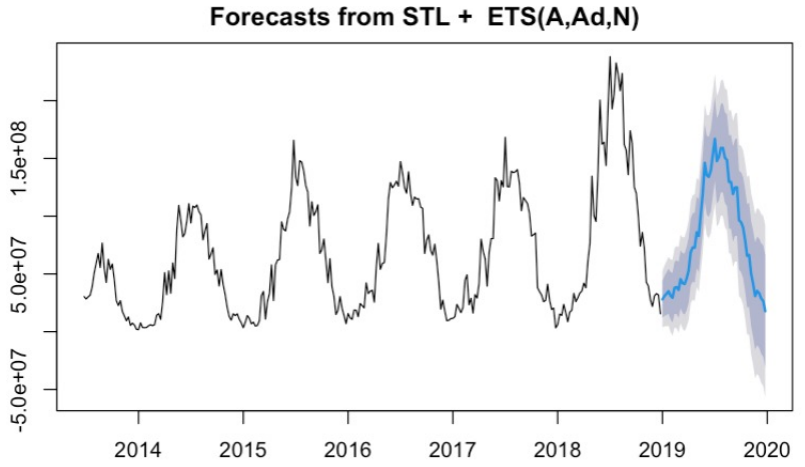
- **AICc:** 11010.01
- **Residuals:** White Noise

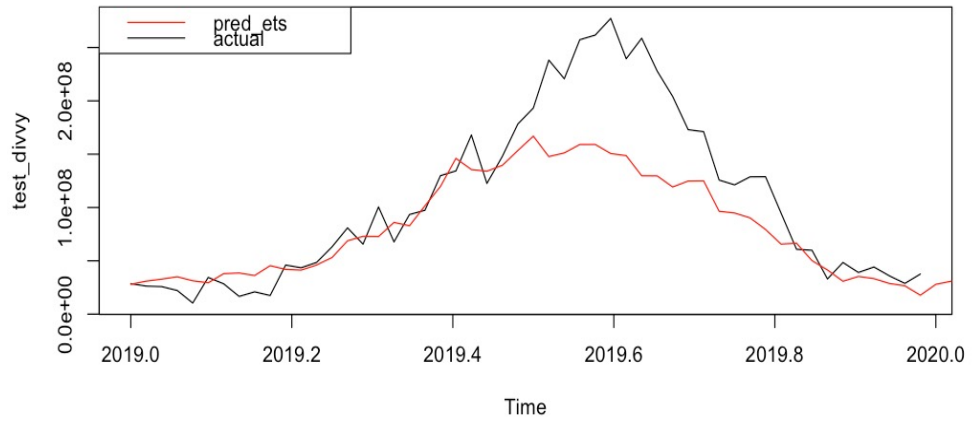



```
        Ljung-Box test

data:  Residuals from STL + ETS(A,Ad,N)
Q* = 61.321, df = 52, p-value = 0.1764

Model df: 5.   Total lags used: 57
```

Forecast from STL + ETS(A,Ad,N)

- **Test MAPE:** 0.3260



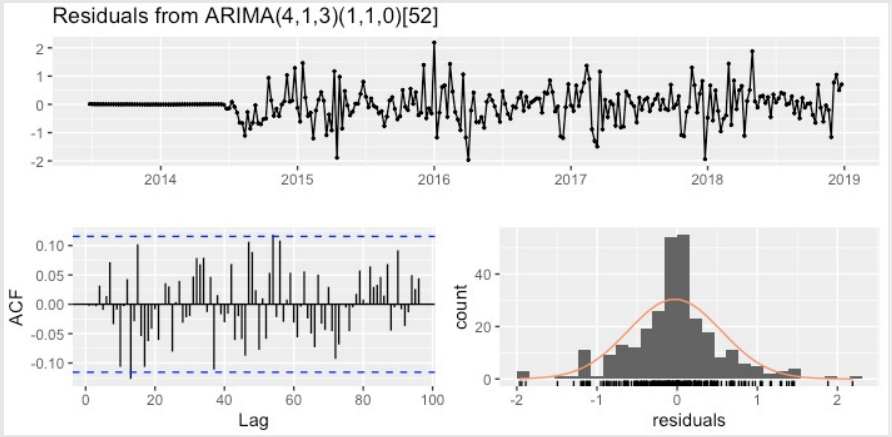


Modeling & Forecasting

Seasonal Arima (SARIMA)

Best Model = ARIMA (4,1,3) (1,1,0) [52]

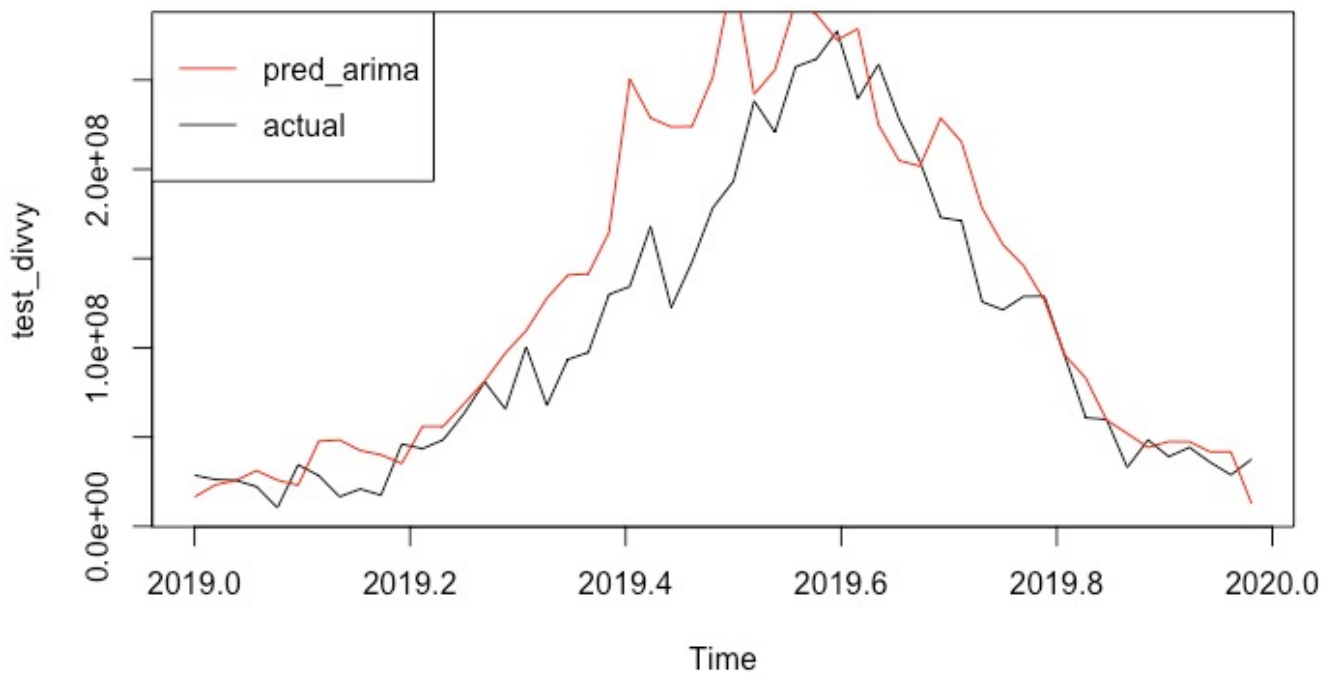
- **AICc:** 490.2
- **Residuals:** White Noise




```
> checkresiduals(arima)

        Ljung-Box test

data:  Residuals from ARIMA(4,1,3)(1,1,0)[52]
Q* = 65.523, df = 49, p-value = 0.05738

Model df: 8.   Total lags used: 57
```

Forecast from ARIMA (4,1,3) (1,1,0) [52]

- **Test MAPE:** 0.3768

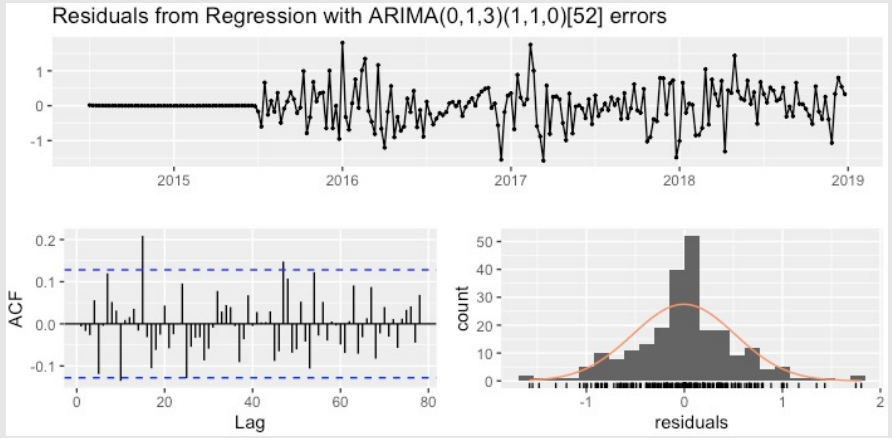


Modeling & Forecasting

Regression with Arima Error

Best Model = ARIMA (0,1,3) (1,1,0) [52] errors

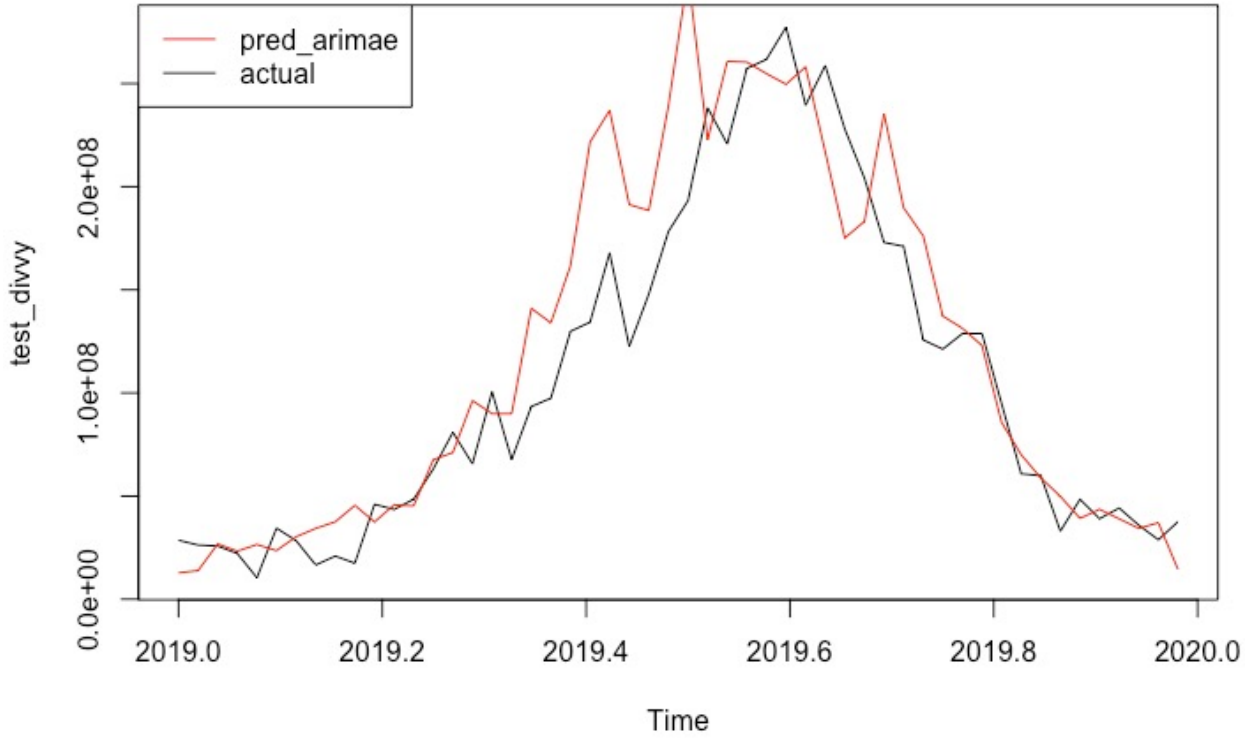
- **AICc:** 345.35
- **Residuals:** Not White Noise, but close
- **Xreg:** Train_temp + Seasonal Diff + 1st Order Diff
- **Y:** Lambda = 0.0349, d = 1, D = 1




```
> checkresiduals(model_arimae)

        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,3)(1,1,0)[52] errors
Q* = 59.302, df = 42, p-value = 0.04025

Model df: 5.   Total lags used: 47
```

Forecast from ARIMA (0,1,3) (1,1,0) [52] errors

- **Test MAPE:** 0.3078



Modeling & Forecasting

Vector AutoRegression (VAR)

```
> VARselect(data,lag.max=5,type='both')$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     3      3      1      3
```

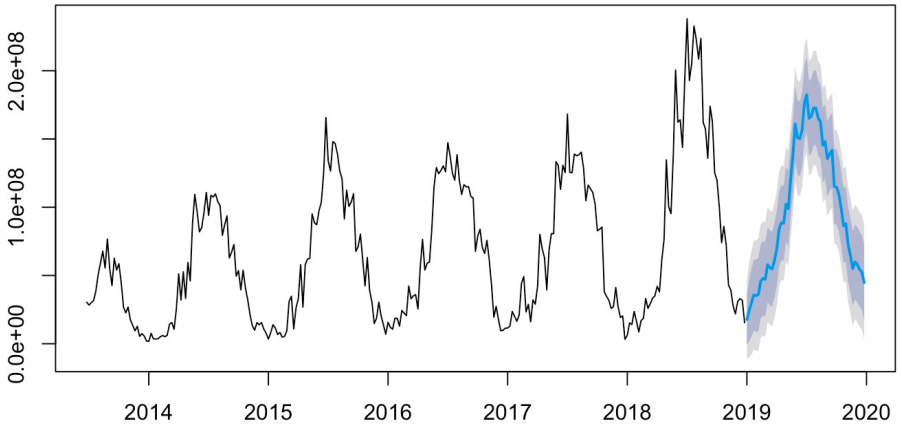
Based on the result above, we would try lag length of 1 and 3; Residuals resemble white noise when lag length is 3.

```
> var1 <- VAR(data[,1:2],p=1,type='both',season = 52)
> serial.test(var1,lags.pt=10,type='PT.asymptotic')

        Portmanteau Test (asymptotic)

data:  Residuals of VAR object var1
Chi-squared = 52.23, df = 36, p-value = 0.03927
```

```
> var2 <- VAR(data[,1:2],p=3,type='both',season = 52)
> serial.test(var2,lags.pt=10,type='PT.asymptotic')

        Portmanteau Test (asymptotic)

data:  Residuals of VAR object var2
Chi-squared = 30.372, df = 28, p-value = 0.3456
```

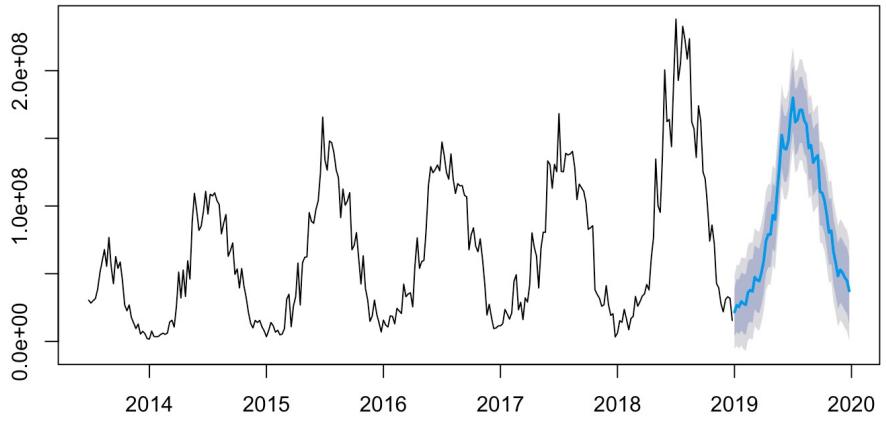
Forecast from VAR (lag=1)

- **Test MAPE:** 0.3989



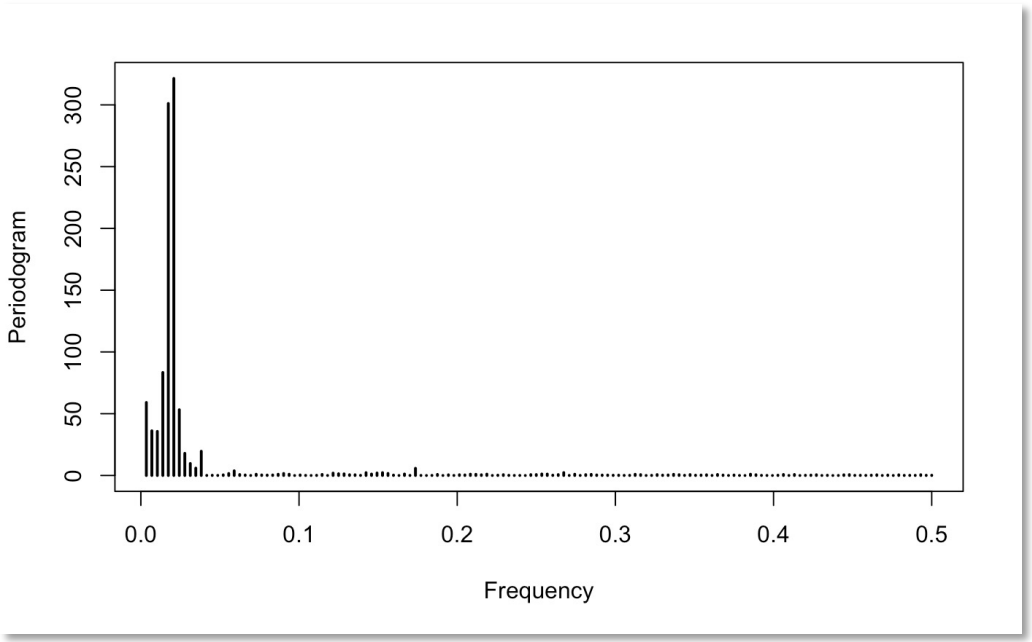
Forecast from VAR (lag=3)

- **Test MAPE:** 0.2949



Modeling & Forecasting

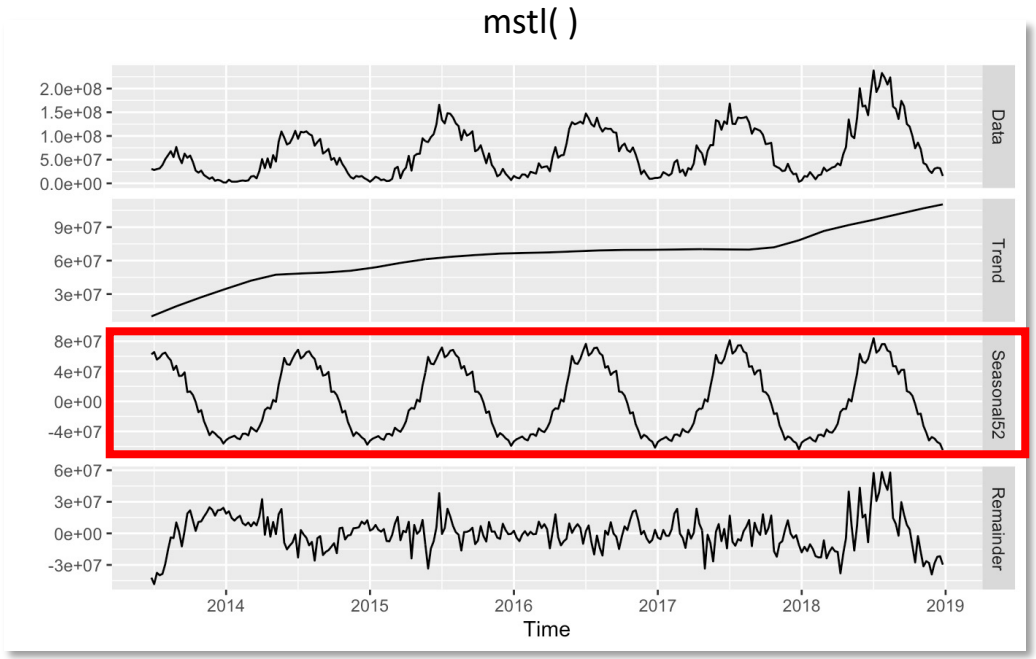
Fourier with Arima Errors



Periodogram

Two highest frequency: 48 & 57.6 weeks

=> Approximate to **52 weeks**

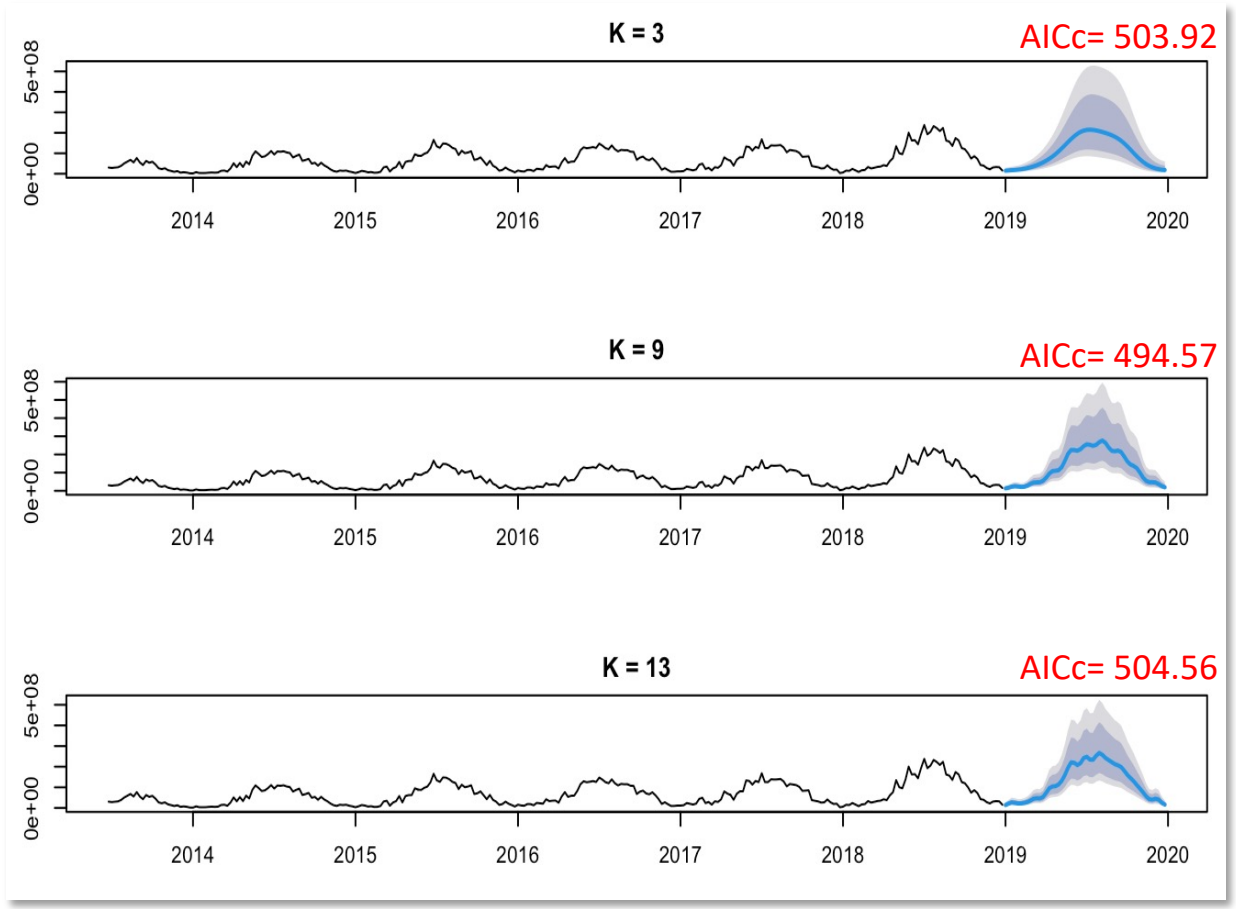


STL

Only One seasonal pattern shown (52 weeks)

Modeling & Forecasting

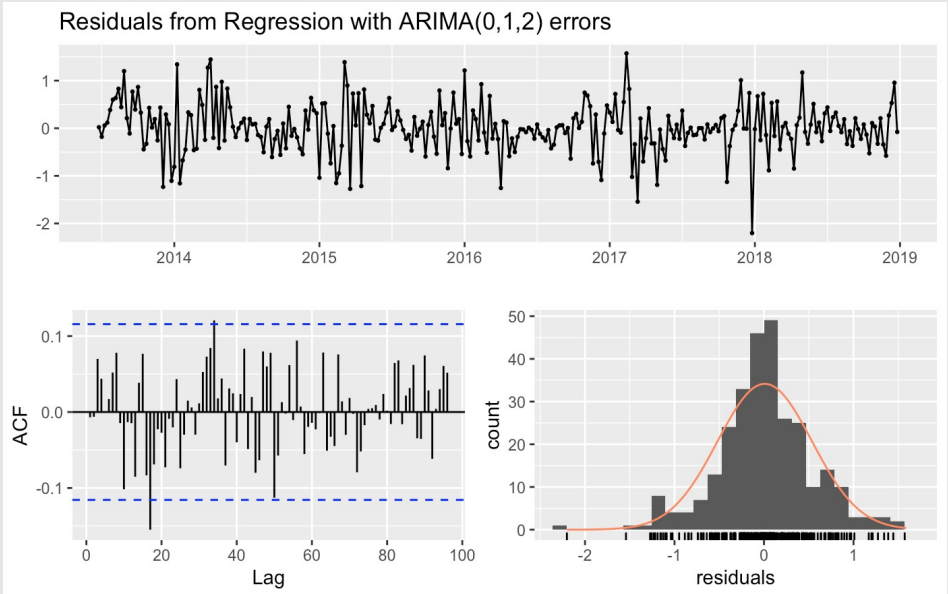
Fourier with Arima Errors *(Cont.)*



K = 9 has the smallest AICc.

Fourier K=9 with Arima(0,1,2) Errors

- **Residuals: Not** White Noise



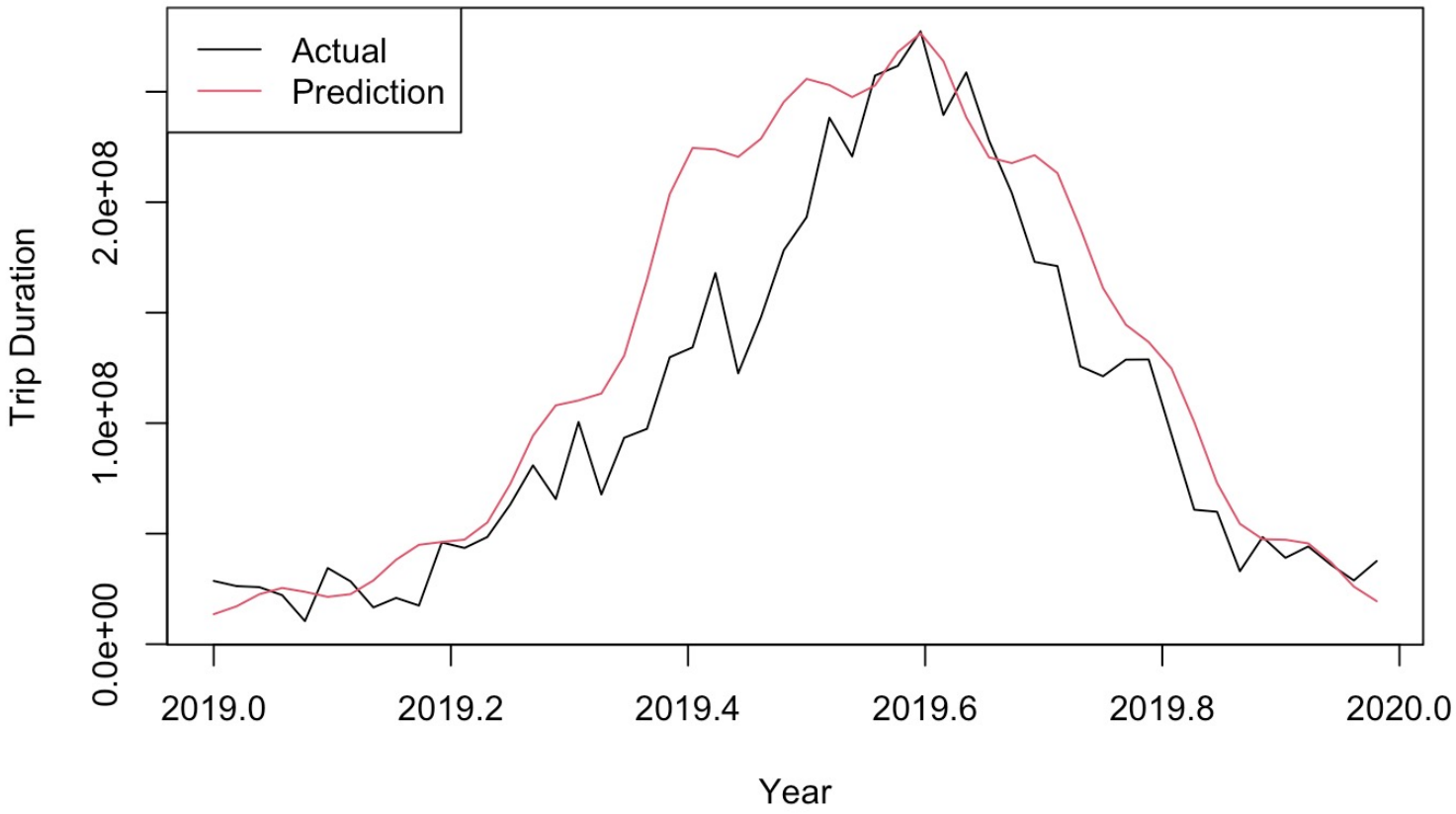

```
        Ljung-Box test

data:  Residuals from Regression with ARIMA(0,1,2) errors
Q* = 64.7, df = 36, p-value = 0.002325

Model df: 21.   Total lags used: 57
```

Modeling & Forecasting

Fourier with Arima Errors *(Cont.)*

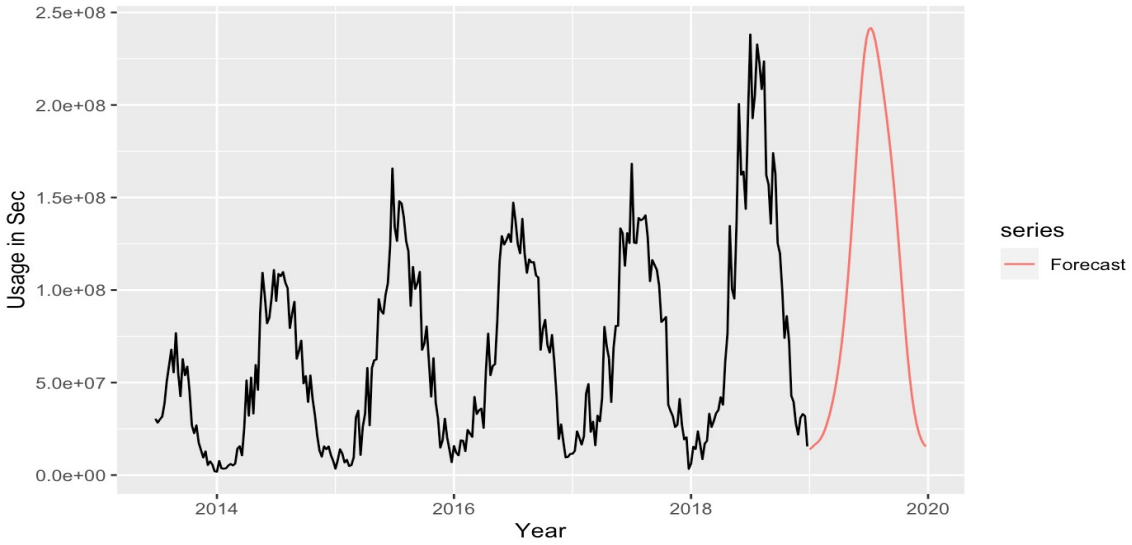
Fourier K=9 with ARIMA(0,1,2) Errors

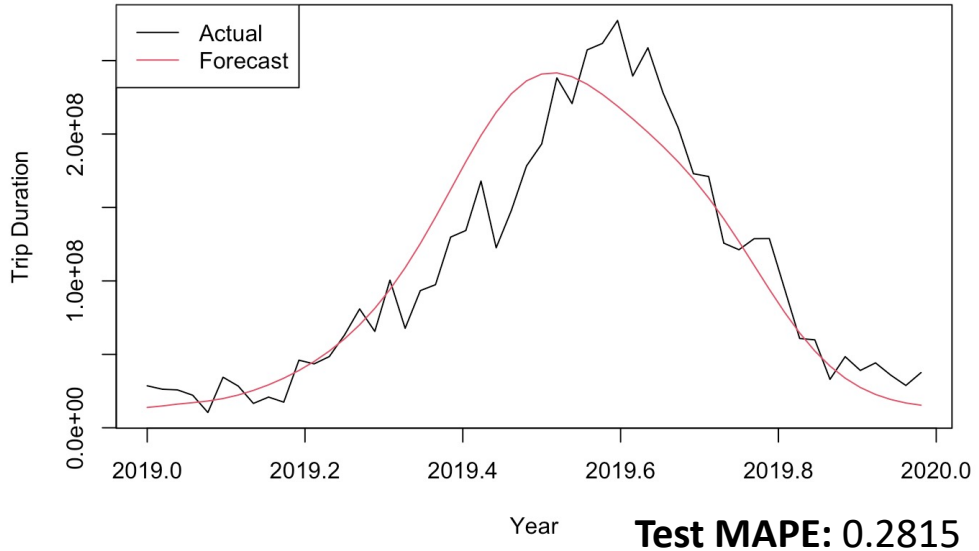
- **Test MAPE:** 0.3375



Modeling & Forecasting

TBATS

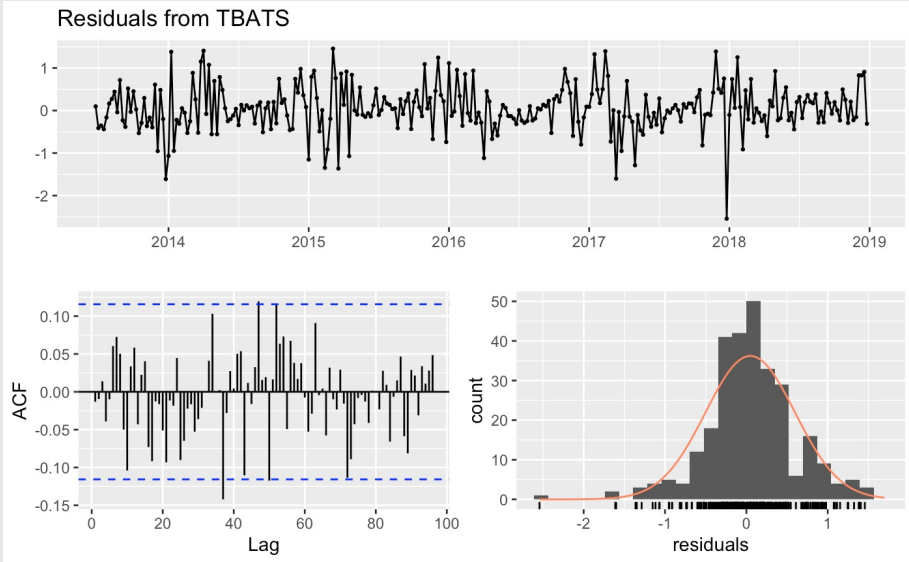




Test MAPE: 0.2815

TBATS (1, {2,1}, 0.873, {<52,3>})

Residuals: Not Resemble White Noise




```
        Ljung-Box test

data:  Residuals from TBATS
Q* = 63.714, df = 38, p-value = 0.005579

Model df: 19.   Total lags used: 57
```

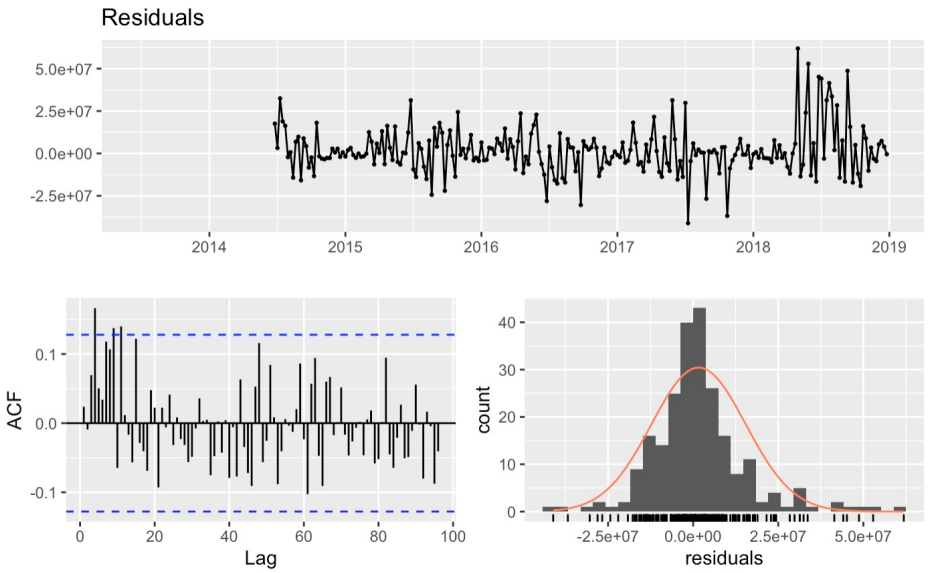
Modeling & Forecasting

Neural Network Autoregression

Best Model = NNAR(7,1,4)[52]

- Set a Box-Cox transformation with lambda = 0 to ensure the forecasts stay positive
- Equivalent to an ARIMA$(7,0,0)(1,0,0)_{52}$ model but **without** stationary restrictions
- p=7 is chosen from the optimal linear model fitted to the seasonally adjusted data
- 4 neurons in the hidden layer

```
> (fit <- nnetar(train_divvy, lambda=0))
Series: train_divvy
Model:  NNAR(7,1,4)[52]
Call:   nnetar(y = train_divvy, lambda = 0)

Average of 20 networks, each of which is
a 8-4-1 network with 41 weights
options were - linear output units

sigma^2 estimated as 0.06795
```



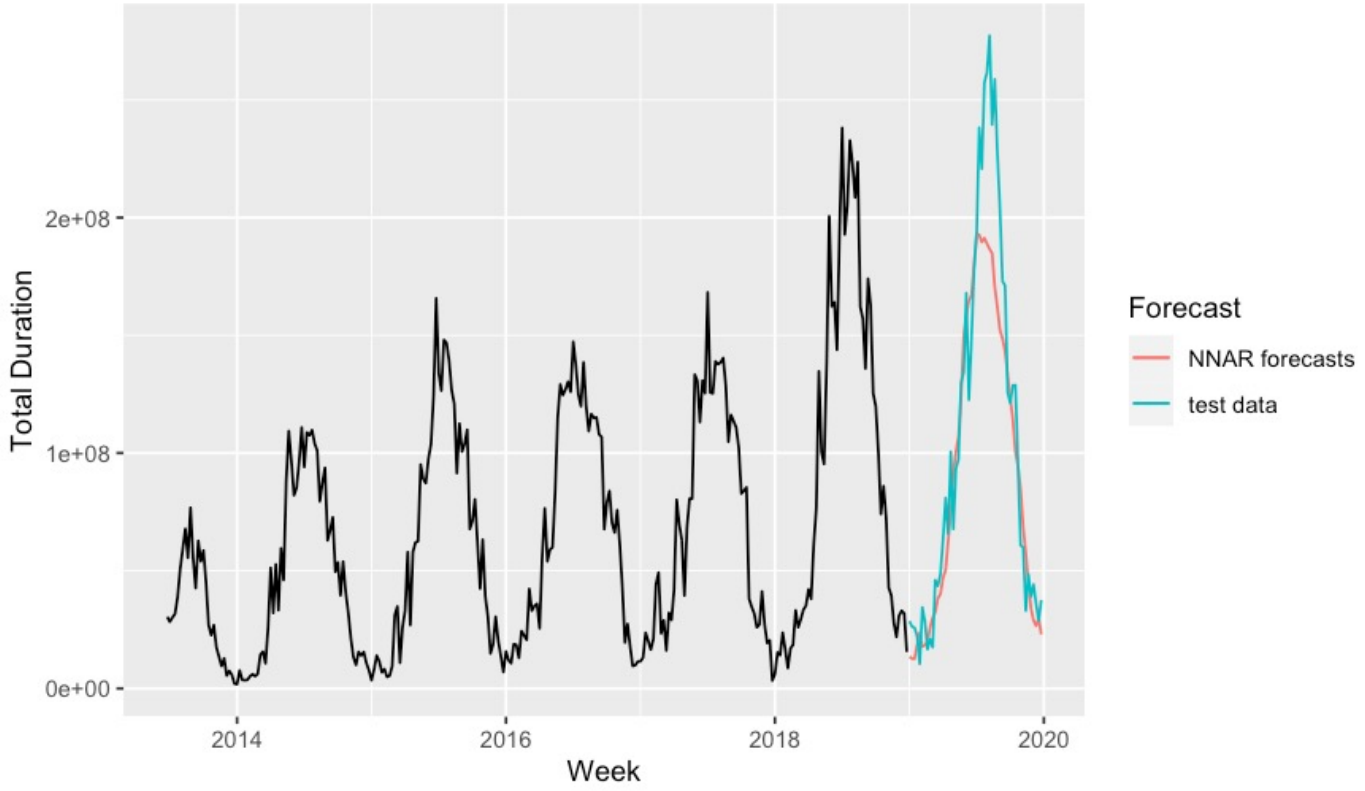

```
        Box-Ljung test

data:  nnetar_model$residuals
X-squared = 52.96, df = 11, p-value = 1.826e-07
```

Modeling & Forecasting

Neural Network Autoregression *(Cont.)*

Forecast from NNAR(7,1,4)[52]

- **Test MAPE:** 0.2549



4. Modeling Selection

Model Selection

Model	MAPE	AICc
Seasonal Naïve	0.2752	-
Exponential Smoothing Model	0.3352	1219.54
Seasonal Arima (SARIMA)	0.3768	490.20
Regression with Arima Error	0.3078	345.35
VAR (lag=1)	0.3989	12253.08
VAR (lag=3)	0.2949	12162.90
Fourier with Arima Errors	0.3375	494.57
TBATS	0.2815	1320.08
Neural Network Autoregression	0.2549	-

5. Conclusions & Next Steps

Conclusions & Next Steps

Conclusions

- Based on MAPE value, Neural Network Autoregression performed the best for our dataset. However, even though the model produced the best trendline for our dataset, there is huge gap between the NNAR forecasts and actual data during peak season
- Surprisingly, Seasonal Naïve generated the second-best prediction in terms of MAPE. We suppose it is probably because the patterns in divvy bike data are fairly periodic, repetitive, and regular
- We also observed that the patterns of Exponential Smoothing Model, Regression with Arima Error, and VAR forecasting models were directional

Next Steps

- As our findings suggest that multiple models achieved good forecast in application to our dataset, different models can be effectively combined and engaged profitably for divvy bike usage prediction
- For potential model improvement, we hope to experiment with some other variables, such as the number of tourists and the dummy variable of whether it is vacation time, which were not used in this project scope, and derive new insights
- In future models, we are also considering the following feature engineering:
  - Convert the numeric variables into binary variables (e.g., snowy day as 1; non-snowy day as 0, etc.)
  - For null or 0 values, we can add it by 1 for better modelling purposes


Thank You!