Time Series Models for Modeling Demand for Ridehailing Services

Zian He

December 9, 2018

Contents

Abstract

In the United States, ridehailing services are steadily becoming an important mode of door-to-door transportation. Their rise has been fueled by their convenience, accessibility, and affordability, however, as their popularity increases there is great concerns about the congestion that they generate. The goal of this study the temporal patterns of the ridehailing trips provided by RideAustin in Austin, Texas between June 2016 and April 2017. Specifically, we identified, developed, and tested a seasonal autoregressive integrated moving average (SARIMA) forecasting model as well as conducted a spectral analysis of this trip trip time series. In this work we show that sub-daily, sub-weekly, and, in particular, weekly seasonal effects should be considered when modeling these types of systems. Performance-wise, the frequency-domain model did not prove to be promising as it explained less than 6% of the observed variability while the SARIMA forecast model tended to underestimate the hourly trip counts in direct relation to hourly trip volumes. Both models also violated fundamental modeling assumptions but nonetheless, this research shows that SARIMA models still hold much promise and outlines ways to rectify many of these concerns.

1 Introduction

In the last 10 years, ridehailing companies such as Uber, Lyft, and Didi have revolutionized the world of private passenger transportation services with their smartphone-based platforms that allows their customers to hail a ride. As ridehailing services take hold, they are beginning to alter patterns of human mobility particularly in urban areas in the United States. Their easy accessibility and the relative affordability provide riders with higher levels of mobility than even that afforded by automobile ownership in some respects. Riders may have to wait minutes for service, but they are relived of looking for parking near their destination, a task that is considerably difficult or costly in urban areas where parking is often highly limited, or, for some, the legal risks associated with driving under the influence of alcohol.

Indeed, ridehailing services are a transformational force in Americas cities, however, they are not without problems. One area of grave concern is that these services are exacerbating traffic congestion as they become more popular. Numerous studies (e.g.,[1] [2]) have found that these services are not only increasing travel demand (i.e., generating more trips) but they are also competing with more efficient modes of transport, namely public mass transit. Consequently, understanding ridehailing ridership trends is not only an internal concern of ridehailing companies but also of great interest to transportation planning agencies.

Public studies about ridehailing services are notoriously rare because ridehailing companies almost never share their data, citing, among other reason, economic and customer privacy concerns. One exception, and the focus of this study, is a trip data set released by Ride Austin, a non-profit ridehailing company based in Austin. This data set covers the 1.5 million+ trips provided by the company between June 2016 and April 2017 [1]. It contains several trip characteristics including the time and location of the passenger pick-up and drop-off, the trip duration and distance, vehicle and driver information, and the passenger trip ratings (for the driver).

This report is structured as follows. We begin by describing our data cleaning efforts and the scale at which we study this data set. Then we discuss our data exploration efforts and subsequently, the transformation that we apply to the base time series as informed by the exploration process. The following section outlines the SARIMA model identification process and the estimation of two candidate models, $SARIMA(1,0,2) \times (0,0,1)_{168}$ and $SARIMA(1,0,1) \times (0,0,1)_{168}$. This is followed by a diagnosis of both models where we find issues of serial correlation with the latter model and thus select the former model as our final candidate. Then we move on to the Spectral Analysis of the time series where we find that this approach does not work well. In the following section we report about the $SARIMA(1,0,2) \times (0,0,1)_{168}$ forecasting model and briefly assess it fit. We conclude with a discussion of future work and a summary of our findings.

2 Preliminaries

Before starting our analyses, we (1) first inspected and cleaned our data and (2) aggregated trips into uniform time intervals. Beginning with the former, to detect outlier trips (e.g., trips with questionable trip

[1] https://data.world/andytryba/rideaustin

information) we inspected the data set with *kepler.gl*, an open-source geospatial analytic visualization tool developed by Uber [2]. We also limited our data set to trips that started and ended in Austin and removed trips with travel distances larger than 100,000 m. Consequently, 1,493,671 trips remained after removing 7536 trips.

After cleaning this data set, we aggregated trips on hourly intervals beginning and ending with each hour. This level of aggregation is used frequently in the transportation modeling literature (e.g., [8]), however, the selection of the appropriate level of aggregation remains an open problem in the transportation literature [7].

3 Data Exploratory Analysis

3.1 Preliminary Data Exploration

We plotted the resulting time series and its histogram in Figures 1 and 2 (below). The time series sample mean was reported to be $\mu = 198.205$ while the sample variance was reported at $\sigma^2 = 56407.620$.

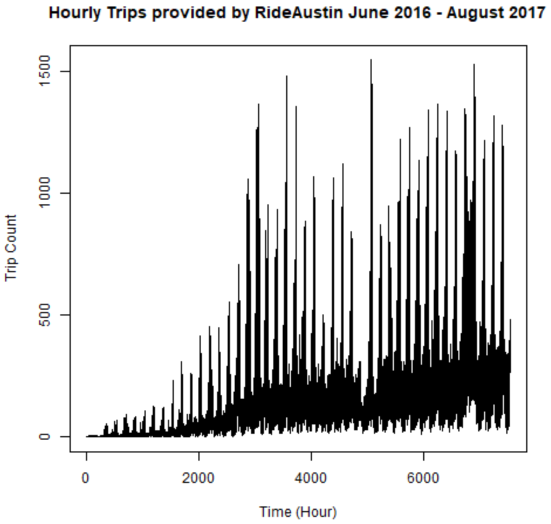

Figure 1: Trip Count to Time

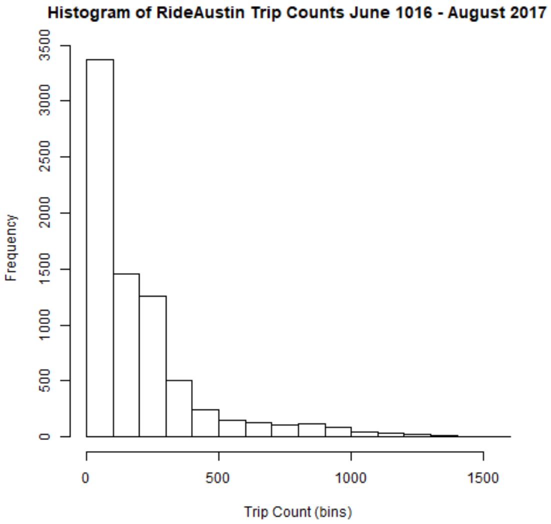

Figure 2: Trip Count Histogram

In terms of trends, we observe that there is a general positive trend in the number of trips both on average and in terms of the variability. The most pronounced increase is observed between hours 2000 and 4000 after which the series appears drop between hours 4000 and 6000 before mostly stabilizing soon thereafter.

3.2 Decomposition Models

To show these changes more clearly, we decomposed the time series into a multiplicative model depicted in Figure 3 (below). In this model, $X_t = m_t * s_t * V_t$ where X_t is the trip count at hour t, m_t is the trend component (at hour t), s_t is the seasonal component, and V_t is a stationary process. We chose the multiplicative model over an additive model because of a strong positive relationship between the trend level and the seasonal variation. This relationship is clearly seen by comparing trip counts and their variability at the beginning and past hour 2500 in time series. This multiplicative relationship as well as the daily and weekly periodicity of the time series is seen in Figure 4 (below) where we plot several representative weekly subsets of the time series (each in a different color). The plot, which begins on Monday (00:00) and ends on Sunday (24:00), clearly shows how both trip counts and their variability increase as the week progresses towards the weekend.

[2] https://kepler.gl/

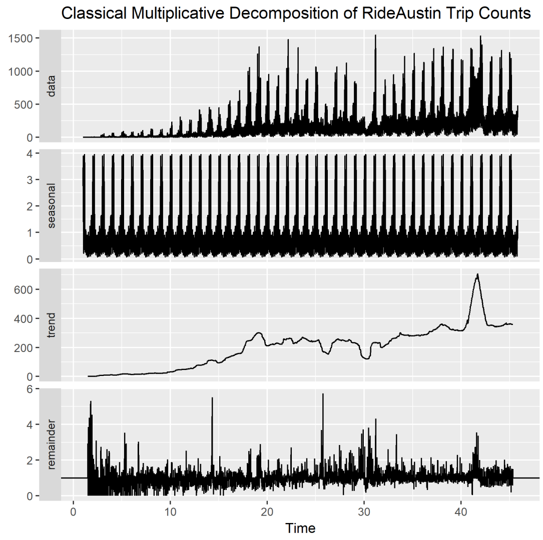

Figure 3: Classical Multiplicative Decomposition

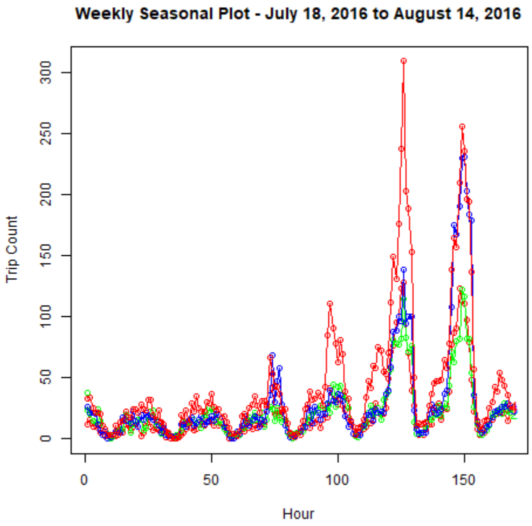

Figure 4: Weekly Seasonal Plot

In all, this preliminary analysis indicates that it is necessary to account for (1) the lack of stationarity and (2) the seasonality in this time series before we begin our model selection.

4 Transformation

In this section we discuss how we transformed this time series to make it more amenable to the time series models we used in this analysis. We began with a transformation of the trip counts to control the time series variance and then we applied differencing transformation to control the observed general and seasonal trends. Nonetheless, after applying these transformation some issues remained with respect to the distribution of the time series. We discuss these problems and their implications in the third subsection.

4.1 Stabilizing Variance

To stabilize the variance we applied a power transformation based on Yeo-Johnson transformations. This was necessary to accommodate the many instances where 0 trips occurred in an hour. Using the *powerTransform* function from the *car* package we found the parameter $\lambda = 0.23$ with upper and lower bands of 0.2247 and 0.2447, respectively. The null hypothesis that the transformation parameter equaled 0 was rejected with a likelihood ratio test (H_0:$\lambda = 0$, LRT $= 1243.934$, df$= 1$, pval$< 2.22e - 16$). Consequently, we transformed our data (X_t) into a new series V_t:

$$V_t = X_t^{0.23} \tag{1}$$

The transformed time series V_t and its histogram are presented below in Figures 5 and 6, respectively. The time series mean is $\mu_V = 2.8935$ and its variance $\sigma_V^2 = 1.1356$, both lower than the values reported for X_t. Overall, the resulting distribution remains somewhat asymmetrical ans left-skewed, yet closer to a normal distribution, however, a general positive trend remains.

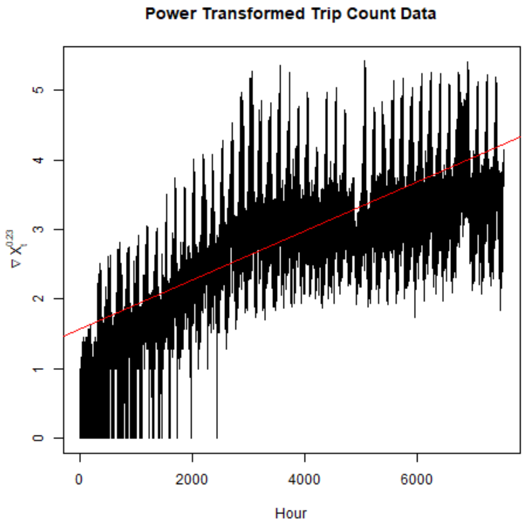

Figure 5: Transformed Time Series V_t

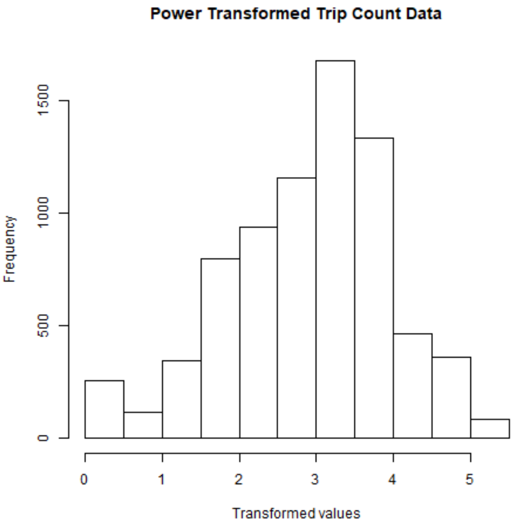

Figure 6: Time Series V_t Histogram

4.2 Removing Trends and Seasonality

After transforming the time series, we differenced the time series to remove the general and seasonal trends. First, we tried a simple single period differencing based on the premise of there being strong serial correlation between the hours of the day:

$$W_t = \nabla V_t = V_t - V_{t-1} \tag{2}$$

This differencing transformation proved have the most impact of any single transformation as the mean dropped to $\mu_W = 0.0638$ and the variance variance dropped to $\sigma_W^2 = 0.0915$. Given the daily and weekly patterns observed in the time series we also considered differencing on a daily(∇_{24}) and weekly(∇_{168}) but in both instances the variance was over a magnitude larger than with W_t. The W_t transformation and is presented below in Figure 7.

Besides the W_t transformation, we considered several combinations of higher order seasonal and non-seasonal differencings with lags of 3, 6, 12, 24, and 168. This decision was prompted by the seasonality trends observed in the original time series. After testing several combinations, we found that a single period non-seasonal differencing along with a 168 period seasonal differencing produced one time series with the best properties:

$$W_t^* = \nabla\nabla_{168}V_t \tag{3}$$

The properties of the this time series included a mean of $\mu_W* = -0.0002$ and a variance of $\sigma_W*^2 = 0.0888$. The final time series W_t^* is presented below in Figure 8. Other comparable time series with respect to the variance included $\nabla_6\nabla_{168}V_t$ ($\sigma_{W_6}^2 = 0.0888$) and $\nabla_{12}\nabla_{168}V_t$ ($\sigma_{W_{12}}^2 = 0.0888$). Given the equal variances, W_t^* is the more intuitive choice as one expects that the trip volume in one hour of the week to be more similar to the trip volume during the same hour from the previous week as compared to the trip volumes from the preceding 6th or 12th hour. In the next section we show provide empirical that supports our seasonal differencing choice.

To conclude our transformations, we apply the Augmented Dickey-Fuller Test to test the stationary of the time series W_t^*. In this test the hypotheses are: $H_0 : W_t^*$ is not stationary and $H_1 : W_t^*$ is stationary. The test produced p-values of 0.01 and so we reject the null hypothesis that W_t^* is not stationary

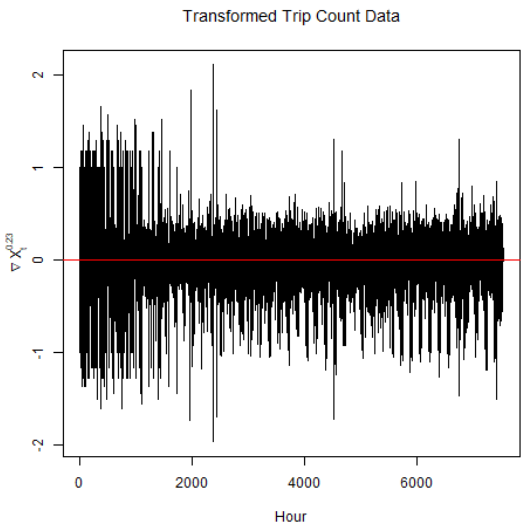

Figure 7: Time Series W_t

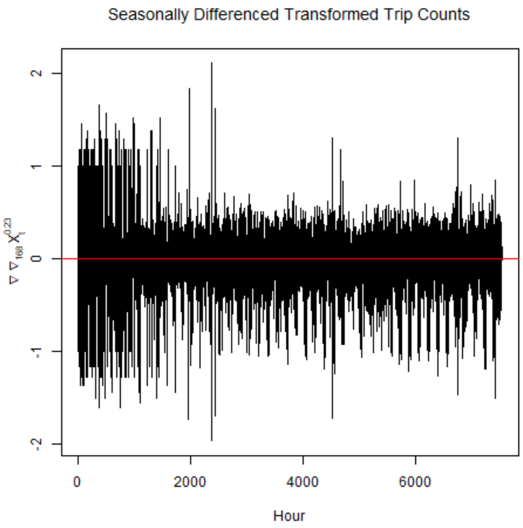

Figure 8: Time Series W_t^*

5 Model Identification and Estimation

To beginning the process of selecting model candidates, the most important finding to consider is the presence of seasonal trends for trip counts that happen on a daily and weekly basis. This suggests the use of a SARIMA time series model which we have already begun building since we differences the the trip count time series with a seasonal lag ∇_{168}. The formulation of a SARIMA model is:

$$SARIMA(p, d, q) \times (P, D, Q)_s \tag{4}$$

Here the order of non-seasonal components are p for the autoregressive (AR) process, d for the differencing, and q for the moving average (MA) process. The order of the seasonal components with a seasonal lag of s are P for the AR process, D for the differencing, and Q for the MA process.

For the trip count time series we have so far identified the differencing parameters $d = 1$ and $D = 1$ with $s = 168$. To establish a starting point for finding the last four parameters we start by analyzing the ACF and PACF plots for a subset of the differenced and transformed time series W_t^*. This new series, denoted W_t^s, excludes the final 336 hours of W_t^* as we reserved these last hours for the forecasting portion of our study.

5.1 Preliminary Model Selection

The ACF for W_t^s at a daily scale (lag = 30) is displayed on the left side of Figure 9 (below). In this ACF we observe that there is an sharp decline between in first 3 lags and that afterwards there are some recurring significant lags in cycles of about 12 lags. The former suggests a SARIMA model with a small, negative non-seasonal and seasonal MA components on a weekly scale while the later suggests that the model might also include a seasonal MA component. However, if we look below at ACFs with longer lag times in Figure 10 (lag = 175) and Figure 11 (lag = 1000) where W_t^s is plotted at a weekly and monthly scale (respectively), we see that there is a dominant weekly trend that tails off. First, this justifies our decision to difference the original time series on a weekly basis. Second, this behavior complicates our analysis solely on the ACF as it appears that it is possible that there is a seasonal AR component with a non-seasonal MA component along the dominant weekly seasonal trend (for for there to be both). A third important observation about these ACFs is the recurring seasonal trends on the sub-weekly scale that persist throughout the time series. As previously mentioned, these trends could not be eliminated with higher order differencing (i.e., including additional 12 and/or 24 seasonal and/or non-seasonal differencings) as they only substantially increased the time series model variance. The cause of the recurring trends could be the result of sub-weekly seasonal AR component but this requires an careful analysis of the PACFs.

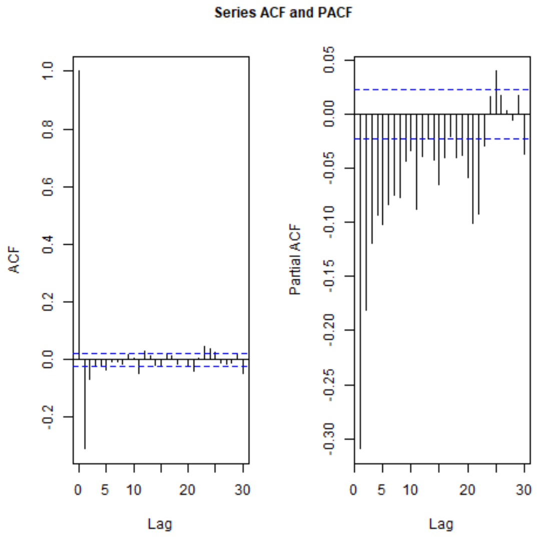

Figure 9: Daily W_t^s ACF and PACF

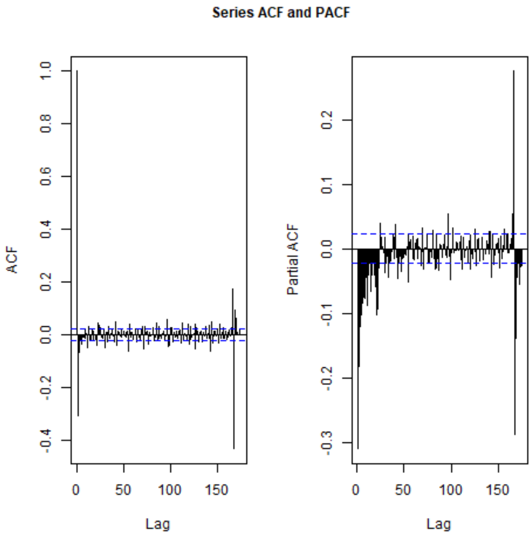

Figure 10: Weekly W_t^s ACF and PACF

The PACF for W_t^s at a daily scale (lag = 30) is displayed on the right side of Figure 9 (above).In this PACF we observe that almost all PACF values between lags 1 and 23 are significant. At first there is a relatively steep decline in absolute terms until lag 10. This is followed by a somewhat erratic pattern of significant and non-significant values between lags 9 and 18 with ”local” spikes at lag 11 and 15. Then there is a substantial increase lags 19 through 21 that is followed by a small but significant lag. On the longer weekly and monthly scale shown in Figure 10, becomes clear that there is a week-long seasonal component that tails off after about 5 weeks as well as sub-weekly seasonal trends that persist throughout the PACF plot particularly on a 24-hour seasonality.

Altogether the convolution of various seasonal trends complicates the model identification process, however, one important starting point is based on an apparent seasonal and non-seasonal MA combination that was previously identified in the ACF plots. This is strongly supported by the decaying and alternating sign values on the weekly lags observed in the monthly PACF. As for the seasonal AR component, it appears that it is present but on a smaller 24 hour seasonality. The key to this is that the values in the first 24 lags resemble those produced by the combination of a seasonal AR and non-seasonal MA component as the is mostly smooth decay that does not alternate in sign until the end of the season (24 hours in this case). Unfortunately, the SARIMA model we consider can only account for one seasonal trend and it cannot be accounted for in our weekly model but the model residuals PACF should help confirm this seasonality if the lags cut off after 24 hours (which they do). As for the non-seasonal AR component it is difficult to identify given the interaction between the MA and seasonal AR components but if it does exist it is likely to have a small order given that the significant lags that occurs outside the weekly lags tend to cut off quickly in the PACF.

Overall, these results suggest a search space for our model where $q = \{1, 2\}$, $P = \{0\}$, and $Q = 1$. The order of p is not as clear and so we shall try iterating through several values to investigate its order. We also investigate the fitted model suggested by *auto.arima* from the *forecast* package to compare the effectiveness of this function (given that it is used in the transportation modeling literature) and we will also iterate through a wide range of low order SARIMA models in order to account for computational challenges related to modeling higher order models.

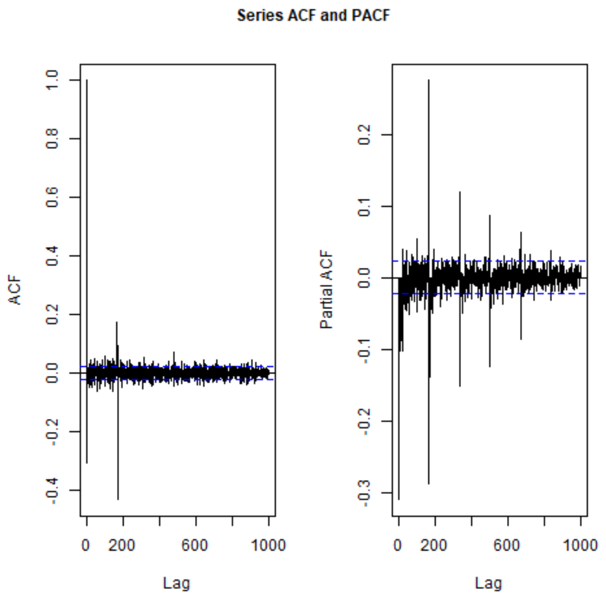

Figure 11: Monthly W_t^* ACF and PACF

5.2 Model Selection

In our experiments we found that the models with the lowest AIC values had at least one positive seasonal AR or MA term. Without these terms models with parameter pairs (p, q) where p and $q = 1, ..., 12$ returned AIC values between 1500 and 1730. The *auto.arima* function returned an ARIMA(2,0,2) model (with an AIC of 1650.78) even when the max p and q parameters were raised to 12. With seasonal AR and MA parameters included, we found at least 15 models with AIC values below 0 within the range of -177 and -1211. In the top five models we found the values $p = \{0, 1\}$, $q = \{1, 2\}$, $P = \{0, 1\}$, and $Q = 1$. The two models with the lowest AIC values, and our model candidates, were $SARIMA(1, 1, 1) \times (0, 1, 1)_{168}$, with $AIC = -1133$, and $SARIMA(1, 1, 2) \times (0, 1, 1)_{168}$, with $AIC = -1211$.

We did not consider any other model as the next comparable model was $SARIMA(0, 1, 2) \times (0, 1, 1)_{168}$ with $AIC = -760$. We note that the model candidates do not conform to our prediction about the seasonal and non-seasonal AR components given the time series PACF. This is in part because we could not compute models with $P \geq 1$ or even with $P = 1$ when $p > 0$. This involved errors with estimating the gradient in the objective function of of time series model with large seasonality values (we started experiencing problem with s values as low as 12). We tried alternative solution methods yet we were still unable to solve most models. Also, models with higher non-seasonal AR orders were not comparable to our candidate models. This includes models with non-seasonal orders of 1 to 6, 12, 18, and 20.

5.3 Model Estimation

For the two candidate SARIMA models we fit and estimated the model coefficients using estimation maximum likelihood. The model coefficients are presented below in Table 1.

	Model 1	Model 2
	$SARIMA(1,1,1) \times (0,1,1)_{168}$	$SARIMA(1,1,2) \times (0,1,1)_{168}$
AR(1)	0.5666 (s.e. = 0.127)	0.7155 (0.0174)
MA(1)	-0.9420 (0.0054)	-1.1480 (0.0219)
MA(2)	-	0.1810 (0.0098)
SMA(1)	-0.8123 (0.0098)	-0.8082 (0.0098)

Table 1: Estimated SARIMA Model Coefficients

The resulting formulaic representation for $SARIMA(1,1,1) \times (0,1,1)_{168}$ is then:

$$(1 - 0.566B)(1 - B)(1 - B^{168})X_t = (1 + 0.9420B)(1 + 0.8123B^{168})Z_t \tag{5}$$

The formulaic representation for $SARIMA(1,1,2) \times (0,1,1)_{168}$ is:

$$(1 - 0.7155B)(1 - B)(1 - B^{168})X_t = (1 + 1.1480B - 0.1810B^2)(1 + 0.8082B^{168})Z_t \tag{6}$$

The final step in our model estimation process is to examine the roots of the model polynomials to check for the causality and invertability of both models. The polynomials roots for Model 1 and 2 are presented in Figures 22 and 23, respectively, in the *Model Estimation* section of the Appendix. All roots (in red) were outside the unit circle for both models and thus, we concluded that both models were causal and invertible.

6 Diagnostics

After identifying our models and estimating their parameters, we move to test the validity of our modeling assumptions. Specifically, we are interested in assessing the normality, independence, and stationarity of the model residuals. This is the last step before forecasting with our model.

6.1 Normality Checking

To check for the normality of the model residuals we first plot their histogram and Q-Q plots and then use the Shapiro Wilk Test. The histogram and Q-Q plots are provided below in Figure 12.

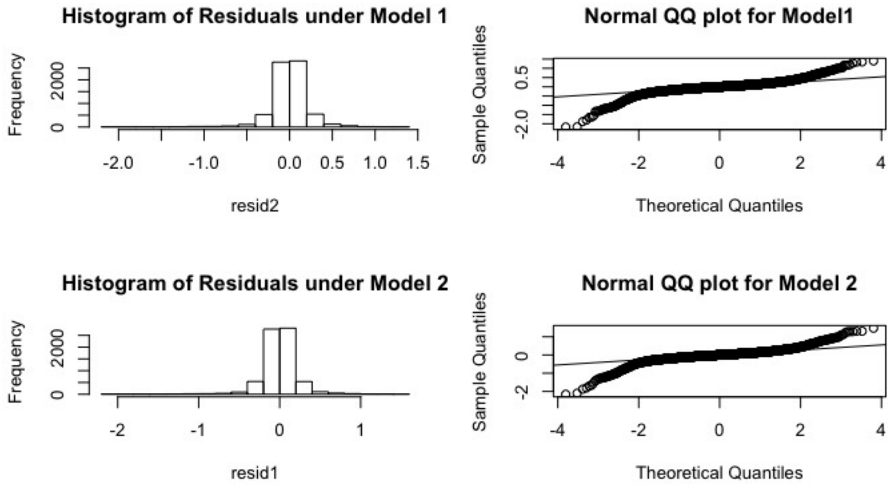

Figure 12: Model Residual Histograms and Q-Q Plots

Clearly, the residuals do not conform to a normal distribution. Although they are symmetrically distributed, they diverge substantially from a normal distribution in that they are far more concentrated around

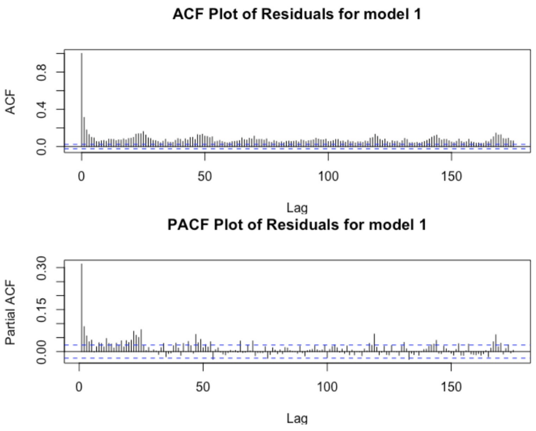

Figure 13: Model 1 Residuals Squared

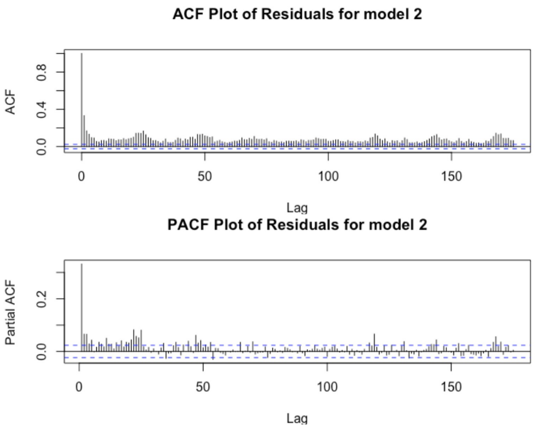

Figure 14: Model 2 Residuals Squared

the mean that would be expected. This conclusion is further supported by Shapiro Wilk Test. In this test we have H_0 : the residuals are normally distributed with an alternative H_1 : the residuals are not normally distributed. At a significance level $\alpha = 0.05$ we reject the hypothesis for both models with $W_{Model1} = 0.88909$ ($p < 2.2e - 16$) and $W_{Model2} = 0.88925$ ($p < 2.2e - 16$).

This result is expected as we are working with count data that is typically not normally distributed. The transformations to normality bring the trip count closer to normality but substantial distortions remain (cf. the histograms in Figures 2 and 6).

6.2 Independence (Serial Correlation) Checking

The next tests concern the independence of the model residuals. Here we apply the Ljung-Box test where we have H_0 : the residuals are not serially correlated and H_1 : the residuals are serially correlated. For Model 1 we fail to reject the hypothesis at significance level $\alpha = 0.05$ with a statistic $\chi^2 = 0.01884$ and $pval = 0.8908$ while for Model 2 we reject the hypothesis with a statistic $\chi^2 = 22.442$ and $pval = 2.2e - 16$. This means that there is serial correlation in Model 2, which is likely due to the absence of the second order non-seasonal MA component. Thus, we concluded that Model 1 is the more appropriate model to use.

6.3 Heteroskedasticity Checking

The final check of our models is for the heteroskedasticity of the model residuals, that is, we are checking whether the model variance changes with time as the time series models we applied assume that there is a constant variance (homoskedasticity). To detect heteroskedasticity we analyze the squared model residual ACF and PACF plots and look for whether they are within 95% of the limits expected from a white noise distribution. These plots are provided below in Figure 13 for Model 1 and Figure 14 for Model 2 (below). Clearly, both models violate the assumption of homoskedasticity as the squared residuals exceed the 95% confidence interval band for white noise. As previously discussed many sub-weekly and sub-daily seasonal trends remain in the time series and so they are likely one source of this problem. The ACFs and PACFs for both models support this claim as they display significant residual squared sizes at lags that coincide with the previously mentioned sub-weekly and sub-daily seasonal trends.

7 Spectral Analysis

A time series can also be expressed as a summation of sine and cosine functions, each with its own frequency and amplitude. Spectral analysis enables us to examine the periodic behavior in the time series. The spectral analysis in this section is done under the final transformed data shown previously in Figure 8.

7.1 Periodogram

A periodogram graphs the spectral density of a signal which is a measure of the relative importance of possible frequency values that might explain the data's cyclical patterns. Thus, it can assist to identify the dominant frequencies that will be used to fit the data into the model as follows:

$$X_t = \mu + \sum_{j=1}^{k} (A_j \cos(2\pi\nu_j t) + B_j \sin(2\pi\nu_j t)) \tag{7}$$

where $v = frequency$

Since a periodogram is best plotted using a stationary data (both Augmented DickeyFuller Test and Kwiatkowski-Phillips-Schmidt-Shin tests show that this data is stationary), a periodogram based on the transformed data Xt is plotted as shown below. As it would be unrealistic to identify all the ν_j, we choose to pick the first 20 most significant ν_j as observed in Figure 15 and then find its corresponding A_j and B_j as the approximation to our model.

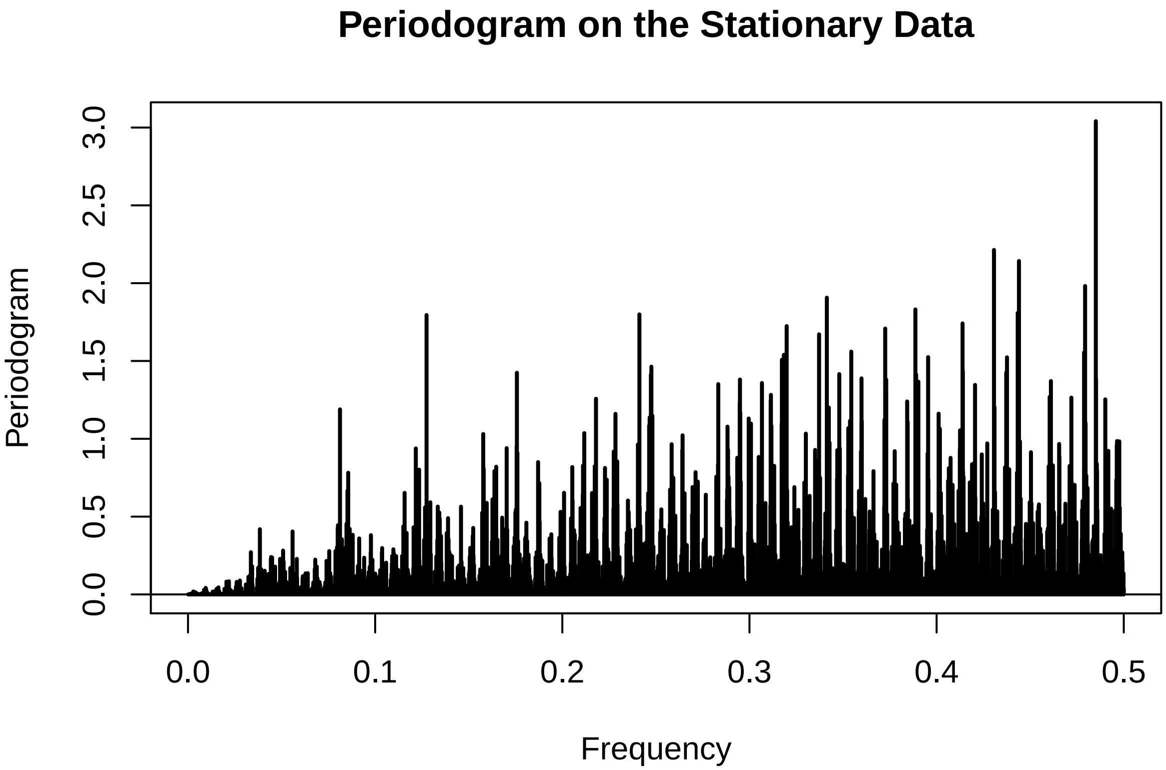

Figure 15: Periodogram on the stationary data

According to our periodogram, the 20 dominant ν_j's are:

$\nu_1 = 0.3413873, \nu_2 = 0.3890322, \nu_3 = 0.4440071, \nu_4 = 0.4307045, \nu_5 = 0.3954120, \nu_6 = 0.3724718, \nu_7 = 0.2412108, \nu_8 = 0.4792996, , \nu_9 = 0.4433284, \nu_{10} = 0.3113886, \nu_{11} = 0.3371793, \nu_{12} = 0.4138727, , \nu_{13} = 0.4852722, \nu_{14} = 0.1274603, \nu_{15} = 0.4434641, \nu_{16} = 0.3730148, \nu_{17} = 0.2589928, \nu_{18} = 0.4308402, \nu_{19} = 0.1757839, \nu_{20} = 0.2832903$

7.2 Using Linear Regression to Determine A_j and B_j

The A_j and B_j in Equation: 7 are estimated based on the linear regression of R. In this case, our $\hat{\mu} = -0.0002381$. To be specific, our final model is approximated as below

$$
\begin{aligned}
X_t \quad = & \quad -0.0002381 \\
- & \quad 0.0252\cos(2\pi \times 0.3414t) + 0.001539\sin(2\pi \times 0.3414t) \\
+ & \quad 0.003352\cos(2\pi \times 0.389t) + 0.02439\sin(2\pi \times 0.389t) \\
- & \quad 0.01008\cos(2\pi \times 0.444t) - 0.0216\sin(2\pi \times 0.444t) \\
- & \quad 0.01419\cos(2\pi \times 0.4307t) - 0.01899\sin(2\pi \times 0.4307t) \\
- & \quad 0.003035\cos(2\pi \times 0.3954t) + 0.02347\sin(2\pi \times 0.3954t) \\
- & \quad 0.009576\cos(2\pi \times 0.3725t) - 0.02113\sin(2\pi \times 0.3725t) \\
- & \quad 0.01659\cos(2\pi \times 0.2412t) + 0.01594\sin(2\pi \times 0.2412t) \\
+ & \quad 0.008689\cos(2\pi \times 0.4793t) - 0.02069\sin(2\pi \times 0.4793t) \\
- & \quad 0.01138\cos(2\pi \times 0.4433t) + 0.0192\sin(2\pi \times 0.4433t) \\
+ & \quad 0.02215\cos(2\pi \times 0.3114t) - 0.0008353\sin(2\pi \times 0.3114t) \\
- & \quad 0.01644\cos(2\pi \times 0.3372t) + 0.01462\sin(2\pi \times 0.3372t) \\
+ & \quad 0.0217\cos(2\pi \times 0.4139t) + 0.00105\sin(2\pi \times 0.4139t) \\
+ & \quad 0.01738\cos(2\pi \times 0.4853t) - 0.01235\sin(2\pi \times 0.4853t) \\
+ & \quad 0.01995\cos(2\pi \times 0.1275t) - 0.007345\sin(2\pi \times 0.1275t) \\
- & \quad 0.01956\cos(2\pi \times 0.4435t) + 0.007629\sin(2\pi \times 0.4435t) \\
+ & \quad 0.02086\cos(2\pi \times 0.373t) + 0.0007472\sin(2\pi \times 0.373t) \\
- & \quad 0.01867\cos(2\pi \times 0.259t) - 0.009097\sin(2\pi \times 0.259t) \\
- & \quad 0.00571\cos(2\pi \times 0.4308t) - 0.01959\sin(2\pi \times 0.4308t) \\
+ & \quad 0.01194\cos(2\pi \times 0.1758t) - 0.01634\sin(2\pi \times 0.1758t) \\
+ & \quad 0.006606\cos(2\pi \times 0.2833t) + 0.01911\sin(2\pi \times 0.2833t)
\end{aligned}
$$

X_t represents the stationary data after transformations and differentiations. A summary of our linear model (refer to Appendix) shows that our R^2 value = 0.05575, which means 5.575% of the variability can be explained using our model. This value is considered reasonable because there are so many ”spikes” in the periodogram and we are only choosing 20 of them. Using the 20 most significant frequencies according to the periodogram is a poor approximation. The periodogram of the fitted data is shown in 16. Comparing the graph, a big difference can be observed.

A comparison between the plot of the usual data and the data approximated using spectral techniques can be seen in the Figure: 17

7.3 Fishers test

The Fishers test enables one to test the residuals of our final model of spectral analysis for the hidden periodicities with unspecified frequency.

$$
\begin{cases} H_0 : X_t \text{ is Gaussian White Noise at level } \alpha \\ H_\alpha : X_t \text{ is not Gaussian White Noise at level } \alpha \end{cases} \tag{8}
$$

The $p - value$ of our test is found to be $0.403339 > 0.05$, which indicates at confidence level $\alpha = 0.05$, there are no hidden periodicities in our data and the residuals are Gaussian white noise.

7.4 Kolmogorov-Smirnov Test

Similar to Fishers test, the Kolmogorov-Smirnov Test is also applied to the residuals of the model to assess whether residuals are Gaussian white noise.

$$
\begin{cases} H_0 : X_t \text{ is Gaussian White Noise} \\ H_\alpha : X_t \text{ is not Gaussian White Noise} \end{cases} \tag{9}
$$

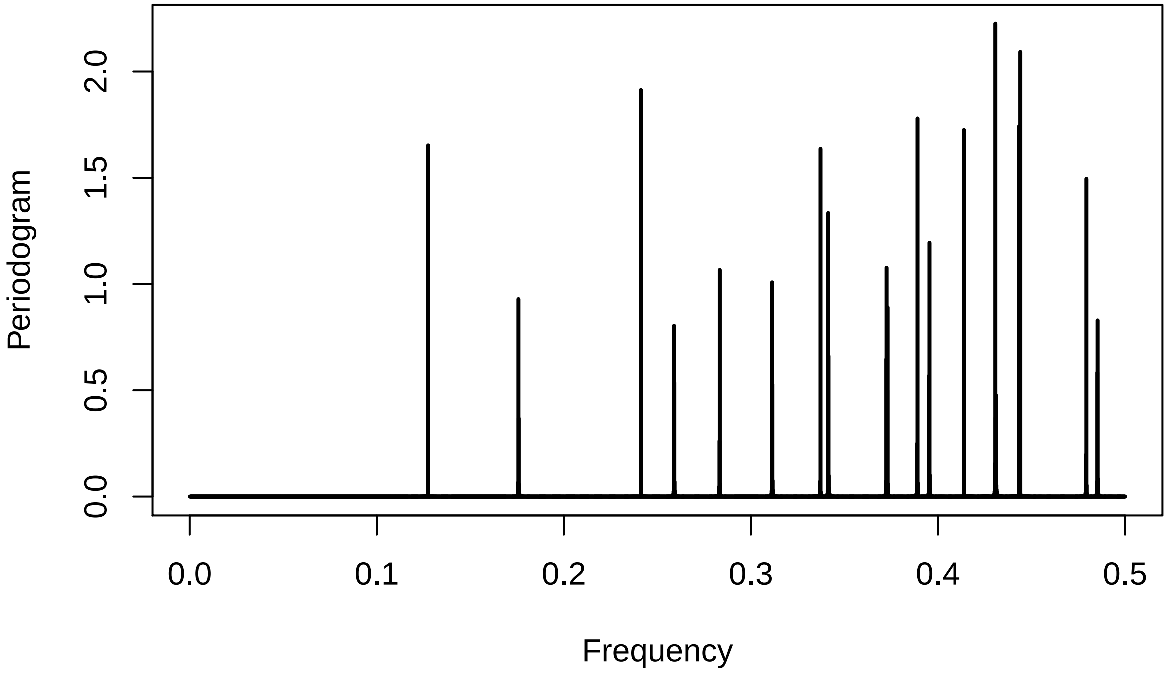

Figure 16: Periodogram on the fitted data

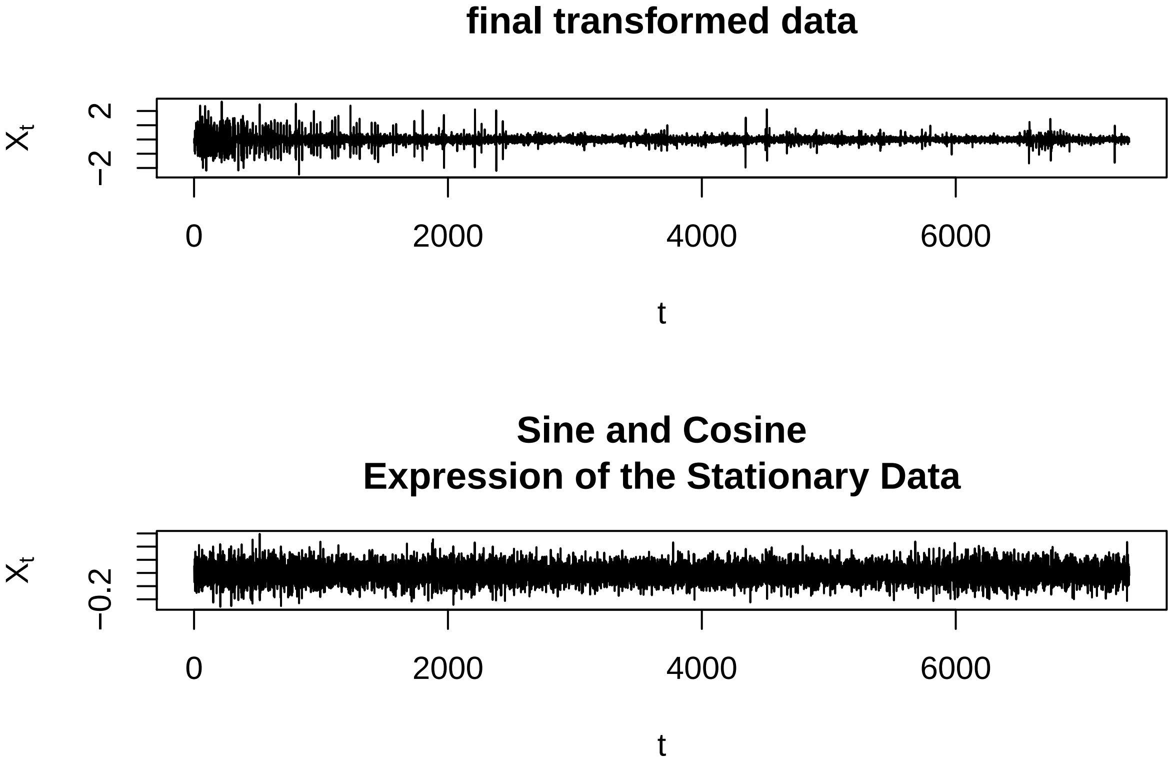

Figure 17: Comparison of Final Transformed Data and Sine and Cosine Expression

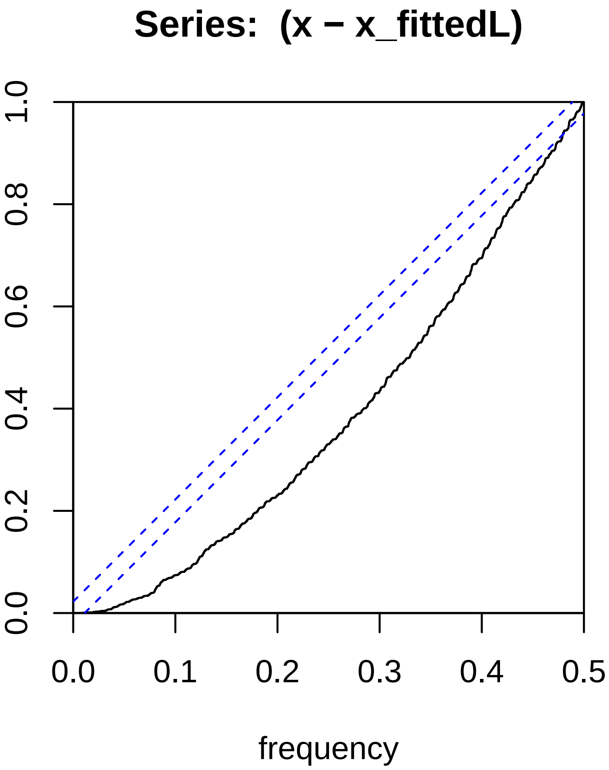

Figure 18: Plot of Cumulative Periodogram

For the Kolmogorov-Smirnov Test, the null hypothesis H_0 is rejected if lots of values are outside the confidence bands. As seen in the cumulative periodogram in Figure 18, almost all of the test values exceed the boundaries. That is to say, the residuals are not Gaussian White Noise. It is reasonable because the low R^2 value in linear regression already indicates that the residuals have not been sufficiently explained by the model.

8 Forecasting

The final section of this project concerns involves testing the forecasting capabilities of the final SARIMA model candidate, $SARIMA(1,0,2) \times (0,0,1)_{168}$. For this analysis we used the final transformed time series W_t^s as our training set and used the 336 remaining hours at the end of the full transformed time series W_t^* as the validation set. We selected a two week-long training set in an effort to test the seasonal component of the SARIMA model. This training period is longer than the 1 to 24-hour training sets used in the literature (e.g., [3] [4]) but using a longer training set provided some insights that are not readily apparent with smaller training sets.

For forecast modeling we used the *forecast* package in R and applied the $SARIMA(1,0,2) \times (0,0,1)_{168}$ to the W_t^s time series with 336 steps predictor. The resulting forecast and the model prediction intervals (at 95%) are plotted below in Figure 19 as, respectively, solid and dashed red lines.

The next step is to undo the power and differencing transformations to analyze a trip count series rather than the transformed series. We note that to recover the original times series we require information of the first 169 observations as they were used in the differencing transformation to build the final time series W_t^*. The code of undoing the differencing operations can be found in Section 11.2.1.

Figure 20 shows the forecast based on trip counts data. In comparing the true time series and the SARIMA forecast model we find that the forecast follows the general shape of the time series but tends to mostly underestimate it. Moreover, this discrepancy is positive correlation with total level of activity as seen by comparing the gap between peak and off-peak hours.

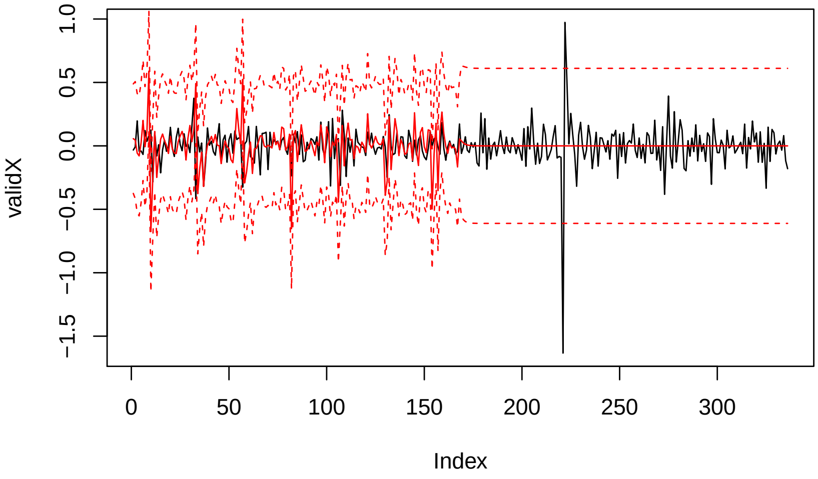

Figure 19: Forecasting based on the transformed data. The black solid line is the original data. The red solid line is the forecasting. The red dashed lines are the 95% prediction interval.

Lastly, the 95% prediction interval upper bounds of the forecasting model in their original and transformed form (i.e., with the power and differencing transforms removed) are presented, respectively, on the left and right of Figure 21. Note that the upper model prediction intervals increase at order 1/0.23 (or 4.347826) while the 95% prediction interval's lower bound converges to zero. The prediction interval's lower bound is expected as count data is strictly non-negative and its upper bound is also expected as errors compound with increasing forecast leads.

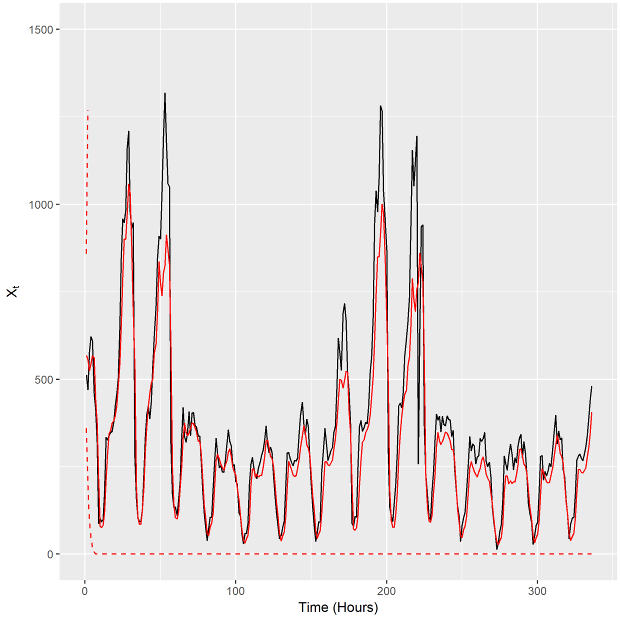

Figure 20: Forecasting based on the original data. The black solid line is the original data. The red solid line is the forecasting. The red dashed lines are the 95% prediction interval.

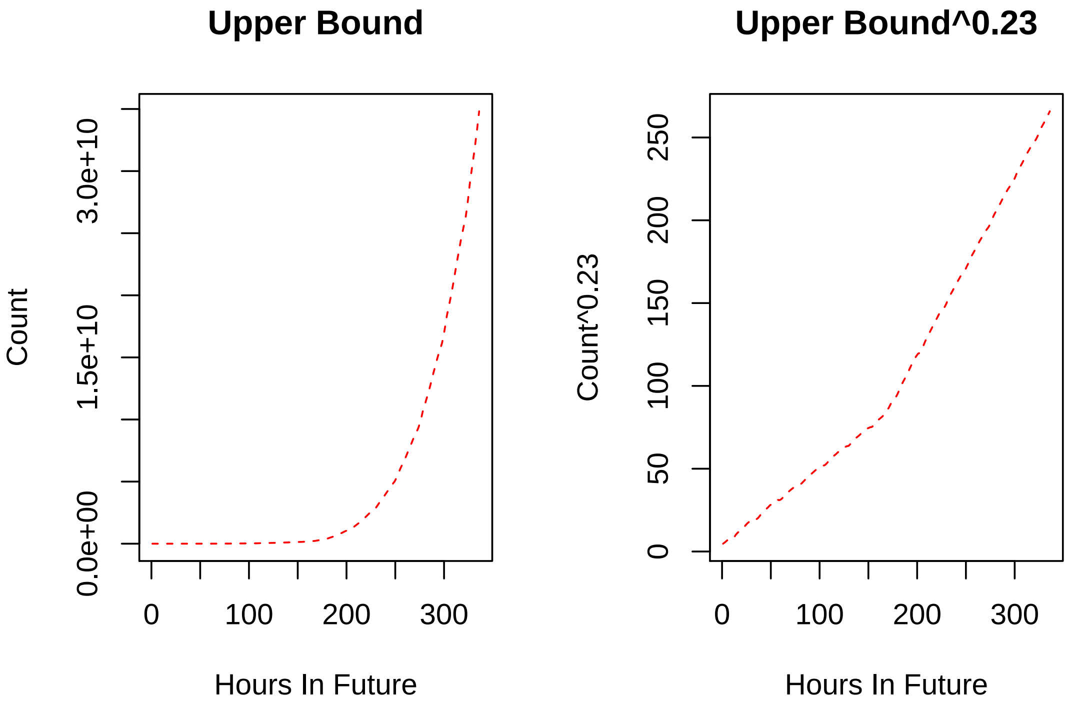

Figure 21: Upper bound and Upper bound Transformed of the Forecasting

9 Future Work

Focusing on ARIMA-based models, despite some of the shortcomings identified with the SARIMA model the results are very promising and support moving forward with research based on this modeling paradigm. Although researchers have identified other modeling paradigms that address many of the shortcoming of ARIMA-based models (e.g., [4]), the simple nature of ARIMA models, their low computational costs, and the availability of free and easy to use software packages make these models an attractive option. These can be significant factors in planning departments that lack the human or computational resources to develop more sophisticated models.

The most immediate work would involve trying different non-differencing transforms to time series to bring the transformed time series closer to normal. In [4] they describe an Anscombe transformation that is more appropriate with count data that has a Poisson distribution (they applied this transformation to taxi trip counts). This could potentially address some of the issues identified in the model diagnosis, particularly the non-normality of residuals. Applying this correction would likely make these models more competitive against other non-ARIMA based models such as those outlined in [4].

Subsequently, more fundamental research would involve developing more sophisticated versions of these SARIMA models. Some options include seasonal multivariate ARIMA (ARIMAX) models that can account for additional variables that affect demand (e.g., rain or holidays), multiseasonal SARIMA models [5] [6] that can account for the sub-weekly and -daily trends observed in the trip count time series and in the model residuals, and ARIMA-GARCH models to address the issues with heteroskedastic residuals. Spatio-temporal ARIMA models are yet another interesting (and more complicated) set of time series models to consider.

10 Conclusion

In this study we developed SARIMA and spectral analysis time series models to model RideAustin's hourly ridehailing trip counts $\{X_t\}$ from June 2016 to April 2017. Applying these standard modeling approaches to this count-based data was challenging work as we encountered several violation of some to some fundamental modeling assumptions related to the time series's stationarity in terms of the mean and variance (homoskedasticity). We observed, among other things, (1) a period of substantial growth for RideAustin and (2) that the transportation process being observed is highly varies along weekly, daily, and sub-daily "seasons". To control for these issues we applied power transformations and differencing to a subset of the time series that excluded the last two weeks. This generated a new time series $W_t^* = \nabla\nabla_{168} X_t^{0.23}$ that we then used to identify several SARIMA models and train two candidate models , $SARIMA(1,1,1) \times (0,1,1)_{168}$ and $SARIMA(1,1,2) \times (0,1,1)_{168}$, and a 20 frequency spectral model. After conducting diagnostic tests for both SARIMA models we selected the $SARIMA(1,1,2) \times (0,1,1)_{168}$ as our final model and used it to forecast the last two weeks of the trip count time series.

Overall, the results of this study were mixed. The final SARIMA models passed the tests for serial correlation, invertability, and causality but failed the test for normality and heteroskedasticity. Likewise, the spectral analysis model failed the test for normality and the linear model only explained 5.575% of the variability. These results highlight the limits of each respective approach and, as discussed in the previous section, other models are perhaps more appropriate. Nevertheless the accessibility of both modeling approaches should keep them as viable modeling options at least for a more restricted set of problems or situations. With the SARIMA-based models, the limit within the domain of transportation is not yet clear. In this work, we went beyond previous works in the literature by showing that the transportation process being models has multiple seasonal trends need to be considered. Finally, the results of the SARIMA forecasting model showed much promise but they are difficult to evaluate the without comparing them directly to other models with the same data set.

References

[1] Y. Babar, G. Burtch. *Examining the Impact of Ridehailing Services on Public Transit Use.* 2017.

[2] R. R. Clewlow and G. S. Mishra. *Disruptive transportation: The adoption, utilization, and impacts of ride-hailing in the United States.* University of California, Davis, Institute of Transportation Studies, Davis, CA, Research Report UCD-ITS-RR-17-07, 2017.

[3] R. Gelda, K. Jagannathan, and G. Raina. *Taxi Dispatches Using Supply Forecasting: A Time-Series Based Approach* In 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016, pp. 13331340.

[4] H. R. Sayarshad and J. Y. J. Chow. *Survey and empirical evaluation of nonhomogeneous arrival process models with taxi data.* Journal of Advanced Transportation, 50(7):12751294, 2016.

[5] J. W. Taylor. *Short-term electricity demand forecasting using double seasonal exponential smoothing.* Journal of the Operational Research Society, 54(8):799805, 2003.

[6] J. W. Taylor. *Triple seasonal methods for short-term electricity demand forecasting.* European Journal of Operational Research, 204(1):139152, 2010.

[7] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias. *Short-term traffic forecasting: Where we are and where were going* Transportation Research Part C: Emerging Technologies, 43:319, 2014.

[8] X. Zhu and D. Guo. *Urban event detection with big data of taxi OD trips: A time series decomposition approach*, Transactions in GIS, 21(3):560-574, 2017.

11 Appendix

Diagnostics

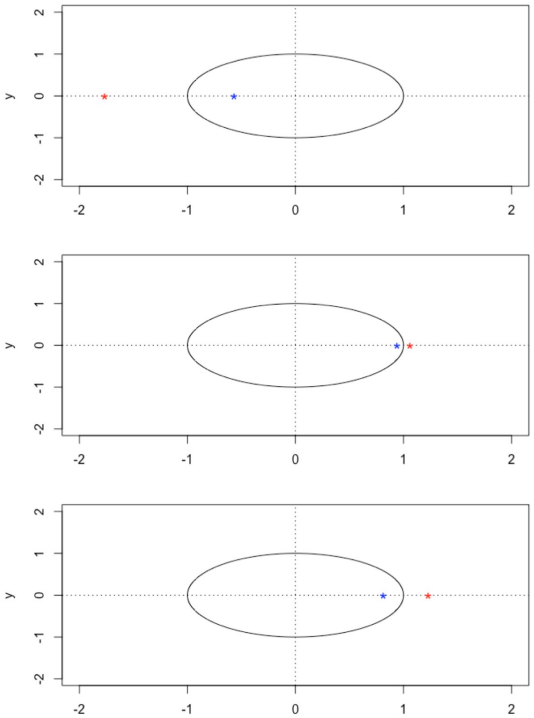

Figure 22: Model 1 Roots - AR (T), MA (M), SMA(B)

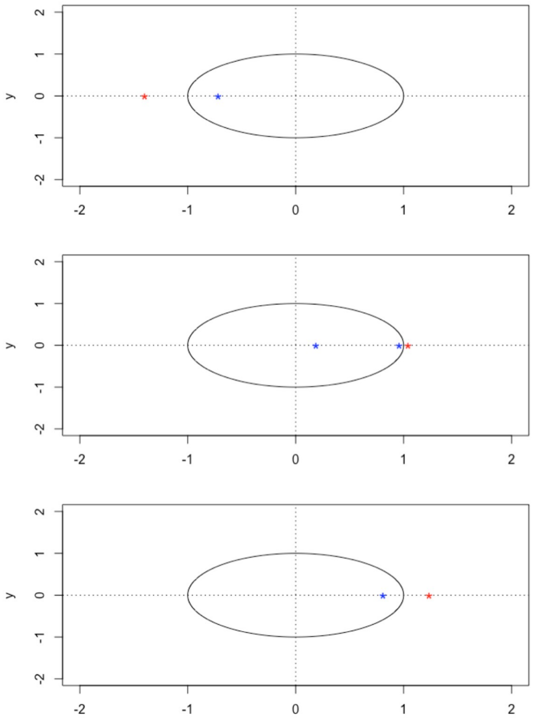

Figure 23: Model 2 Roots - AR (T), MA (C), SMA(B)

Code

Set-up

~~~
```{r setup, include=FALSE}
library(tidyverse)
library(lubridate)
library(stats)
library(urca)
library(car)
library(forecast)
require(MASS)
library(dplyr)
library(AICcmodavg)
```
~~~

Functions

```
#SARIMA  Function that catches errors, returns model parameters and NAs
sarima.can <- function(ts.model, a.vector, o.vector, season.f){
  out <- tryCatch(
    #Try
    withCallingHandlers(
     {
      #Run ARIMA model with Parameters
        arima(ts.model, order = a.vector, seasonal = list(order = o.vector, period = season.f), opti
      },
    #Warning
        warning = function(warn){
          invokeRestart("muffleWarning")
       }
      ),
    #Error
      error = function(err){
        return(NA)
      }
  )
  return(out)
}
```

Clean Data

```
#Read Data
Rides_DataA <- read_csv("Rides_DataA.csv")

#Generate CSV to Explore with Kepler.GL
longrides <- filter(Rides_DataA, distance_travelled >= 50000) %>% dplyr::select(start_location_lat,
write.csv(longrides, "longrides.csv")

#Filter out data with more than 100,000 m in distance_travelled
Trip_Data <-filter(Rides_DataA, distance_travelled <= 100000) %>% dplyr::select(start_location_lat,
rm(Rides_DataA)
```

Generate Time Series

```
#Hourly
Trip_hourly <- Trip_Data %>% mutate(day = as.numeric(date(started_on)), wday = wday(started_on), hou

#Get Hours where count is zero
full_dates <- cbind(rep(16956:17269,each=24),seq(0,23))
full_dates <-as.data.frame(full_dates)
full_dates <- full_dates %>% mutate(wday = wday(as_date(V1)))
Trip_hourly <- full_join(Trip_hourly,full_dates, by = c("day" = "V1", "wday" ="wday", "hour"= "V2"))
Trip_hourly <- arrange(Trip_hourly, day, hour) %>% mutate(count = ifelse(is.na(count), 0, count))
write.csv(Trip_hourly,"trip_hourly.csv")
rm(full_dates)

#Make time series
trip_H <- ts(Trip_hourly)[,4]
```

Explore Time Series

```
  #Time Series Plot
    png("tspolt_trip_H.png")
    plot.ts(trip_H, xlab = "Time (Hour)", ylab = "Trip Count", main = "Hourly Trips provided by RideAu
    dev.off()

  #Histogram
    png("hist_trip_H.png")
      hist(trip_H, xlab = "Trip Count (bins)", main ="Histogram of RideAustin Trip Counts June 1016 -
    dev.off()


#Summary Stats
    ts_mean <- mean(trip_H)
    ts_var <- var(trip_H)

#Decomposition plot
  #Daily
  seasonplot(trip_H, 24, col = rainbow(3), year.labels=TRUE, main="Seasonal Plot - Daily")

  #Weekly
  seasonplot(trip_H, 168, col = rainbow(3), year.labels=TRUE, main="Seasonal Plot - Weekly")

#Seasonal Trends
  trips.trend.24 <- ma(trip_H, order = 24, centre = T)
  trips.trend.168 <- ma(trip_H, order = 168, centre = T)
  plot(as.ts(trips.trend.24), col ="blue")
  lines(trips.trend.168, col ="red", lwd = 2)

#Decompose
  #Daily
  ts.24 <- ts(Trip_hourly[,4], frequency = 24)
  D.a.ts.24 <- decompose(ts.24, "additive")
  D.m.ts.24 <- decompose(ts.24, "multiplicative")

    #Additive
    plot(as.ts(D.a.ts.24$seasonal))
    plot(as.ts(D.a.ts.24$trend))
    plot(as.ts(D.a.ts.24$random))
    plot(D.a.ts.24)

    #multiplicative
    plot(as.ts(D.m.ts.24$seasonal))
    plot(as.ts(D.m.ts.24$trend))
    plot(as.ts(D.m.ts.24$random))
    plot(D.m.ts.24)

  #Weekly
  ts.168 <- ts(Trip_hourly[,4], frequency = 168)

    #Additive
      D.m.ts.24 <- ts.24 %>% decompose(type="additive")  %>%  autoplot() + xlab("Time") +
                    ggtitle("Classical Additive Decomposition of RideAustin Trip Counts")
      ggsave("D.m.ts.168.png", height = 6,  width = 6.25)

    #Multiplicative
      D.m.ts.168 <- ts.168 %>% decompose(type="multiplicative")  %>%  autoplot() + xlab("Time") +
                    ggtitle("Classical Multiplicative Decomposition of RideAustin Trip Counts")
      ggsave("D.m.ts.168.png", height = 6,  width = 6.25)

#ACF and PACF Plots
    png("ACFpolt_trip_H.png")
    op <- par(mfrow=c(1,2))
    acf(trip_H, lag.max = 30, main = "RideAustin Time Series ACFs")
    acf(trip_H, lag.max = 175, main ="")
    par(op)
    dev.off()

    png("PACFpolt_trip_H.png")
    op <- par(mfrow=c(1,2))
    pacf(trip_H, lag.max = 30, main = "RideAustin Time Series PACFs")
    pacf(trip_H, lag.max = 175, main ="")
    par(op)
    dev.off()

#QQ plot
    png("qqpolt_trip_H.png")
    qqnorm(trip_H)
    abline(0,1)
    dev.off()
```

Power Transformation

```
 #Find parameter
  tc.bc <- powerTransform(trip_H ~ 1, family="yjPower")
  summary(tc.bc)

 #Apply Transformation
  trip_H.t <- (trip_H)^(0.23)

 #Time Series Stats
```

```
  ##png("tspolt_trip_H.t.png")
  plot.ts(trip_H.t, xlab = "Hour", ylab = "BOXCOX(Trip Count)", main = "Hourly Trips provided by Ri
  ##dev.off()

  #Summary Stats
  ts.t_mean <- mean(trip_H.t)
  ts.t_var <- var(trip_H.t)

  #ACF and PACF Plots
  ##png("ACFpolt_trip_H.t.png")
  op <- par(mfrow=c(1,2))
  acf(trip_H.t, lag.max = 350)
  pacf(trip_H.t, lag.max = 350)
  par(op)
  ##dev.off()

  #QQ plot
  ##png("qqpolt_trip_H.t.png")
  qqnorm(trip_H.t)
  abline(0,1)
 ## dev.off()

#Differencing
    trip_H.t.d.1.D.168 <- diff(diff(trip_H.t,1),lag =  168, differences = 1)

#Transformed Time Series Plot
  png("tspolt_trip_h_t.png")
    plot(trip_H.t, xlab = "Hour", ylab="",main = "Power Transformed Trip Counts")
    title(ylab = expression(nabla~X[t]^0.23), line = 2)
    abline(lm(trip_H.t~as.numeric(1:length(trip_H.t))),col = 'red')
  dev.off()

  #Histogram
  png("hist_trip_H_t.png")
    hist(trip_H.t, xlab = "Transformed Values",main = "Power Transformed Trip Counts")
  dev.off()

#d(1)Time Series Plot
  png("tspolt_trip_h_t_d_1.png")
    plot(trip_H.t.d.1, xlab = "Hour", ylab="",main = expression("Differenced Transformed Trip Count
    title(ylab = expression(nabla~X[t]^0.23), line = 2)
    abline(lm(trip_H.t.d.1~as.numeric(1:length(trip_H.t.d.1))),col = 'red')
  dev.off()

  png("hist_trip_H_t_1.png")
    hist(trip_H.t.d.1, xlab = "Transformed Values",main = "Differenced Transformed Trip Counts")
 dev.off()

  #QQ plot
  png("qqpolt_trip_H_t_d_1.png")
  qqnorm(trip_H.t.d.1)
  abline(0,1)
 dev.off()
```

```
#d(1) & D(168) Time Series Plot
  png("tspolt_trip_H_t_d_1_D_168.png")
    plot(trip_H.t.d.1, xlab = "Hour", ylab="",main = expression("Seasonally Differenced Transformed
    title(ylab = expression(nabla~nabla[168]~X[t]^0.23), line = 2)
    abline(lm(trip_H.t.d.1.D.168~as.numeric(1:length(trip_H.t.d.1.D.168))),col = 'red')
  dev.off()

  #Histogram
  png("hist_trip_H_t_d_D_168.png")
    hist(trip_H.t.d.1.D.168, xlab = "Transformed values",main = "Seasonally Differenced Transformed
 dev.off()

 png("qqpolt_trip_H_t_d_1_D_168.png")
  qqnorm(trip_H.t.d.1.D.168)
  abline(0,1)
 dev.off()

#ACF and PACF Plots
  png("ACFpolt_trip_H_t_d_1_D_168_L_30.png")
  op <- par(mfrow=c(1,2))
  acf(trip_H.t.d.1.D.168, lag.max = 30, main="")
  pacf(trip_H.t.d.1.D.168, lag.max = 30,main="")
  mtext("Series ACF and PACF", side = 3, line = -2, outer = TRUE, font = 2)
  par(op)
  dev.off()

  png("PACFpolt_trip_H_t_d_1_D_168_L_1000.png")
  #op <- par(mfrow=c(1,2))
  #acf(trip_H.t.d.1.D.168, lag.max = 75, main="")
  pacf(trip_H.t.d.1.D.168, lag.max = 1000,main="")
  mtext("Series PACF", side = 3, line = -2, outer = TRUE, font = 2)
  #par(op)
  dev.off()

  #QQ plot
  png("qqpolt_trip_H.t.d.24.168.png")
  qqnorm(trip_H.t.d.1.D.168)
  abline(0,1)
 dev.off()
```

Model Identification

```
#ARIMA Models
    #Auto.arima suggestion
      auto.arima(trip_H.t.d.1.D.168)

    #Explore Parameter Range
    #ARMA Stats DF
    ARMA_summ <- tibble(p = 0, q=0, sigma2 = 0.00, na.se = 0, aic = 0.00, LL = 0.00, rss = 0.000)

    #Solve Models
    for(i in 1:12){
      for(j in 1:12){
        #Model
    ARMA_can <- arima(trip_H.t, order = c(i,0,j))

    #Update Summary
    ARMA_summ <- ARMA_summ %>% add_row(p = i, q = j, sigma2 = ARMA_can$sigma2, na.se = sum(is.na(

    #Save output
    write.csv(ARMA_summ, "ARIMA_summ.csv")
   }
  }

#SARIMA Models
  #SARIMA Stats DF
   SARIMA_summ <- tibble(p = 0, q=0, P = 0, Q = 0, S = 0, sigma2 = 0.00, na.se = 0.00, aic = 0.00,
   SARIMA_models <- list()
    counter <- 1

  #Set Seasonal Investigation Parameters
   Seas <- c(6,12,24,168)

  #Loop through Models
  for(i in 0:6){
   for(j in 0:6){
    for(m in 0:4){
     for(n in 0:2){
      for(k in 1:length(Seas)){
       print("");cat("Model"); cat(" p:"); cat(i); cat(" q:"); cat(j); cat(" P:"); cat(m); cat
      #ARIMA
       ARIMA_v <- c(i,0,j)
       SARIMA_V <- c(m,0,n)

      #Model
       SARIMA_can <- sarima.can(trip_H.t.d.1.D.168, ARIMA_v, SARIMA_V, Seas[k])

      #If valid solution
       if(!is.na(SARIMA_can)){
        #Add to list
         SARIMA_models[[counter]] <- SARIMA_can


        #Update Summary
         SARIMA_summ <- SARIMA_summ %>% add_row(p = ARIMA_v[1], q = ARIMA_v[3], P = SARIMA_V
       } else{
        SARIMA_summ <- SARIMA_summ %>% add_row(p = ARIMA_v[1], q = ARIMA_v[3], P = SARIMA_V[1
       }#End If

      #Increase counter
       counter <- counter + 1

      #Save output
       write.csv(SARIMA_summ, "SARIMA_summD.csv")
      }
     }
    }
   }
```

```
      }

      #Save and Output Data
        save.image()
         write.csv(SARIMA_summ, "SARIMA_summ.csv")
```

11.1 Spectral Analysis

11.1.1 Prepare

```
library(astsa)
library(tseries)
library(MASS)
library(forecast)
library(TSA)
library(GeneCycle)
rawDat = read.csv("trip_H.t.d.1.D.168.csv",header = TRUE)
names(rawDat) = c("t","x")
x = rawDat$x
ts.plot(x, main = "final transformed data")
```

11.1.2 Periodogram

Here is the test of Stationarity

```
> kpss.test(x)

        KPSS Test for Level Stationarity

data:  x
KPSS Level = 0.0052142, Truncation lag parameter = 11, p-value = 0.1

Warning message:
In kpss.test(x) : p-value greater than printed p-value
> adf.test(x)

        Augmented Dickey-Fuller Test

data:  x
Dickey-Fuller = -29.508, Lag order = 19, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(x) : p-value smaller than printed p-value
>
```

As we can see, the final transformed time series passed both tests.

```
TSA::periodogram(x,main="Periodogram on the Stationary Data")
abline(h=0)
```

11.1.3 Linear Regression

```
LSize = 20
freqWL = p1$freq[order(p1$spec, decreasing=TRUE)] [ 1 : LSize ]
t=1:length(x)
```

```
w=2*pi*t
formulaStr = "x~"
for(ticker in seq(1:LSize)){
  assign(paste("x",toString(2*ticker-1),sep=""),cos(w*freqWL [ ticker ]) )

  assign(paste("x",toString(2*ticker),sep=""),sin(w*freqWL [ ticker ]) )
  formulaStr=paste(formulaStr,paste("x",toString(2*ticker-1),sep=""))
  formulaStr=paste(formulaStr,"+")
  formulaStr=paste(formulaStr,paste("x",toString(2*ticker),sep=""))
  if (ticker<LSize){
    formulaStr=paste(formulaStr,"+")
  }
}
zL = lm(as.formula(formulaStr))
```

11.1.4 Formula Writer

```
LatexFormula=paste("X_t &=& ", toString(signif(ParaL[1],4)),sep = "")
totIter = LSize*2

isCosine = TRUE
for (ticker in seq(1:totIter)){
  currCoef = ParaL[1+ticker]

  if (currCoef>=0){
    if (isCosine){
      currSignString = "\\\\&+&"
    }else{
      currSignString = "+"
    }

    currCoefString = toString(signif(currCoef,4))
    currCoefString = paste(currCoefString,sep = "")
  }else{

    if (isCosine){
      currSignString = "\\\\&-&"
    }else{
      currSignString = "-"
    }

    currCoefString = toString(signif(-currCoef,4))
    currCoefString = paste(currCoefString,sep = "")
  }

  inString = toString(signif(freqWL[ceiling(ticker/2)],4))


  if (isCosine){
    LatexFormula = paste(LatexFormula, currSignString,sep = "")
    LatexFormula = paste(LatexFormula, currCoefString,sep = "")
    LatexFormula = paste(LatexFormula, " \\cos(2\\pi\\times ",sep = "")
    LatexFormula = paste(LatexFormula,inString,sep = "")
    LatexFormula = paste(LatexFormula,"t)",sep = "")
    isCosine = FALSE
  }else{
    LatexFormula = paste(LatexFormula, currSignString,sep = "")
    LatexFormula = paste(LatexFormula, currCoefString,sep = "")
    LatexFormula = paste(LatexFormula, " \\sin(2\\pi\\times ",sep = "")
    LatexFormula = paste(LatexFormula,inString,sep = "")
    LatexFormula = paste(LatexFormula,"t)",sep = "")
    isCosine = TRUE
  }
}
```

11.1.5 Plot the Comparison

```
Para=z$coeff
x_fitted=Para[1]+ Para [2]*x1+Para [3]*x2+Para[4]*x3+Para[5]*x4+Para[6]*x5+Para[7]*x6+
Para[8]*x7+Para[9]*x8+Para[9]*x9+Para[10]*x10
op <- par(mfrow = c(2, 1))       # square plotting region,
plot(x, type = 'l', ylab=expression(X[t]), xlab = "t", main = "final transformed data")
plot(t,x_fitted,type='l',ylab=expression(X[t]),main="Sine and Cosine
Expression of the Stationary Data" )
```

11.1.6 Fisher's Test

```
>  fisher.g.test( (x-x_fittedL) )
[1] 0.403339
```

11.1.7 Kolmogorov-Smirnov Test

```
> cpgram( (x-x_fitted) )
```

11.2 Forecasting

11.2.1 Inverse of diff function with lag

```
removeNeg<-function(x){
  tmp = cumsum1
  tmp[is.negative(tmp)] <- 0
  return(tmp)
}

cumsumWithLag<-function(arrIn, arrFill, lag){
  result = array(numeric())
  preArr = arrFill
  arrParseInIndex =0

  result = arrFill

  while(length(result)<length(arrIn)+lag){
    arrParseInIndex = arrParseInIndex + 1
    if (length(result)%%lag==0){

      preArr = tail(result,lag)
    }

    result = append(result, arrIn[arrParseInIndex] + preArr[1+(arrParseInIndex-1)%%lag])
  }

  return(result)
}
```

11.2.2 Forecasting

```
library(astsa)
library(tseries)
library(MASS)
library(forecast)
library(TSA)
library(GeneCycle)
library(schoolmath)
modelForcast = forecast(timeSeriesCandidate, h=168*2, level=0.95)

forecastMean = removeNeg(cumsumWithLag(cumsumWithLag(
    append(trainX, modelForcast$mean[1:length(modelForcast$mean)]),
    head(diff(origx^0.23,1),168), 168), head(origx^0.23,1), 1 ) )^(1/0.23)


forecastUpper = removeNeg(cumsumWithLag(cumsumWithLag(
    append(trainX,modelForcast$upper[1:length(modelForcast$upper)]),
    head(diff(origx^0.23,1),168), 168), head(origx^0.23,1), 1))^(1/0.23)

forecastLower = removeNeg(cumsumWithLag(cumsumWithLag(
    append(trainX,modelForcast$lower[1:length(modelForcast$lower)]),
    head(diff(origx^0.23,1),168), 168), head(origx^0.23,1), 1))^(1/0.23)
```

11.2.3 Plotting

```
plot(tail(origx,length(validX)), type='l',col='blue', ylab = "Count", xlab = "Hours In Future")
lines(tail(forecastMean,length(validX)), type='l', col='red')
lines(tail(forecastUpper,length(validX)), type='l', lty=2, col='red')
lines(tail(forecastLower,length(validX)), type='l', lty=2, col='red')
```