MTH 313: Data Analysis Project

Jessica Rankins

May 12, 2017

1 Introduction

My dataset is a time series that tracks Internet traffic in bits over time that can be seen in Figure 1. The traffic is from a single private Internet service provider's transatlantic link. The data was collected every five minutes from June 7, 2005 at 06:57 to July 31, 2005 at 11:17.

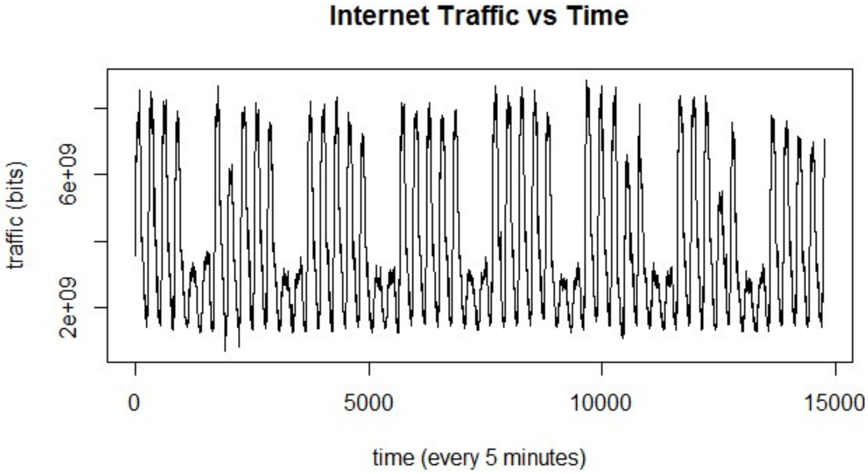


Figure 1: Plot of Original Traffic Data over Time

From an initial view of the plot, there seems to be clear seasons and perhaps constant seasonal variance. There seems to be a daily season that accounts for the peaks and a weekly season that accounts for the different peak heights. However, it is difficult to tell if there is a significant trend or cycles.

Following these initial analyses, in order to more accurately predict future usage, I plan to fit, analyze, and compare multiple models. These will include a simple trend model for time series, a multiplicative Classical Decomposition model, an additive Classical Decomposition model, and an ARIMA model following the Box-Jenkins methodology. To compare these models, I will remove the last 100 data points from my series before I start modeling. I will then use forecasting procedures on each model and compare how well they predict these 100 data points.

Hopefully models similar to these could be used to predict future usage of similar Internet service providers. This could be very helpful insofaras enabling the providers to find the best time to do maintenance on their systems by knowing when the usage is at a minimum, or allowing a user to know when the system may respond slower due to a higher usage.

2 Simple Trend Model

Fitting a linear trend model on my data showed that time was significant with a p-value of $.0000113 < \alpha = 0.05$. Fitting a quadratic trend model again showed time was significant with the same p-value, but not time squared (it had a p-value of 0.111 which is not less than $\alpha = 0.05$). Thus, time seemed to be statistically significant, but not time squared, so I used an initial linear trend model of

$$\begin{aligned} y_t &= \beta_0 + \beta_1 t + \epsilon_t \\ \hat{y_t} &= \hat{\beta_0} + \hat{\beta_1} t \\ &= 3947148537 - 18495t \end{aligned}$$

using the coefficient values approximated by R.

As can be seen in the graph of the residuals vs time in Figure 2, there was extreme positive autocorrelation for this linear trend model because there was a cyclic pattern in the residuals.

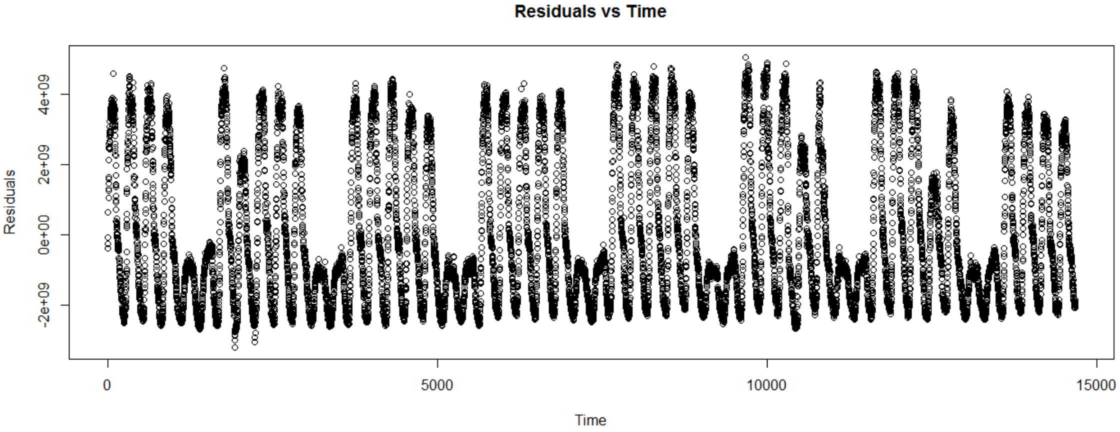


Figure 2: Residuals vs Time for Linear Trend Fit Shows Positive Autocorrelation

To double check for positive autocorrelation, I did a Durbin-Watson test on the linear trend fit:

H_0: The error terms are not correlated.
H_a: The error terms are positively correlated.
d = 0.0043409
p-value <2.2e-16

Since the test statistic was so close to 0 and the p-value was less than $\alpha = 0.05$, I rejected the null hypothesis and concluded that the error terms had positive autocorrelation. Hence, I attempted to model this autocorrelation as seasons in future models. However in this model, I knew that my errors did not follow a normal distribution and so I needed to model my autocorrelation with a first order autoregressive process given by $\epsilon_t = \phi_1 \epsilon_{t-1} + a_t$ where a_t follows a normal distribution.

Now that I began modeling my error autocorrelation, I needed to estimate ϕ_1 in $y_t = \beta_0 + \beta_1 t + \epsilon_t$ where $\epsilon_t = \phi_1 \epsilon_{t-1} + a_t$. To do this, I used the Cochrane-Orcutt Procedure by using the code in Appendix A. First, I did a regression on $\epsilon_{1:T-1}$ and $\epsilon_{2:T}$ using the residuals from my linear fit model to get an estimate for ϕ_1 (Appendix A: 33-40). This regression did not account for autocorrelation, so I calculated the generalized differences and fit $y^* = \beta_0^* + \beta_1 t^*$ (Appendix A: 43-45). I then found $\hat{\beta_0} = \hat{\beta_0^*}/(1 - \hat{\phi_1})$ and calculated the residuals using this calculation (Appendix A: 48-49). I then fit my regression to $\epsilon_t = \phi_1 \epsilon_{t-1} + a_t$ to get a new estimate of $\hat{\phi_1}$ (Appendix A: 52-56). I then repeated this process starting from the generalized differences until $\hat{\phi_1}$ converged to 0.9978598.

Thus my final simple trend model was

$$\hat{y_t} = 4143069517 - 50484.46t + 0.9978598\epsilon_{t-1}$$

after getting the new intercept and slope from this procedure.

3 Multiplicative Classical Decomposition

To model my data using multiplicative decomposition, I used

$$y_t = (TR_t) \times (SN_t) \times (CL_t) \times (IR_t)$$

where TR_t represents the trend, SN_t represents the seasonality, CL_t represents the cycles, and IR_t is the irregular error.

3.1 Seasonality

The first thing I attempted to model was the seasonality of my data. As mentioned before, there seemed to be both a daily and a weekly season present. I began by considering the daily seasons. The first daily season length I used was $(24 \times 60)/5 = 288$ since there are 24×60 minutes in a day and my data was taken every 5 minutes. However, when I plotted the data with each week on top of the last, the peaks did not quite line up correctly as I expected them to. Without my peaks lining up correctly based on this season length, my estimated daily seasonality would be incorrect and get worse at approximating my data over time. Adjusting the length of the season little by little and re-plotting the weeks on top of each other, I instead found a season length of 283 that made the peaks line up much better (Appendix B: lines 36-38).

Next, I took a centered moving average of my time series with an order of the daily season length (283). This averaged the first 283 consecutive points in my series, then it dropped the first observation and added the 284th observation to calculate the second average. This continued for the rest of my data points. Every 283 consecutive points were averaged together to eliminate seasonal and irregular variations. Since I was interested in seasonality, I divided my series by this centered moving average to get the seasonal and irregular variations (Appendix B: lines 39-40).

I then made a matrix with 283 columns and populated it with my data so that each column represented one observation time in a day. For each column, I averaged all values in that column together to get the average for each time of day. I then normalized these averages so that they would sum to my daily seasonal length. This gave me one day of seasons, so I then repeated this season for the number of days my data has (Appendix B: lines 43-47). Figure 3 shows the daily seasonality I constructed during this process.

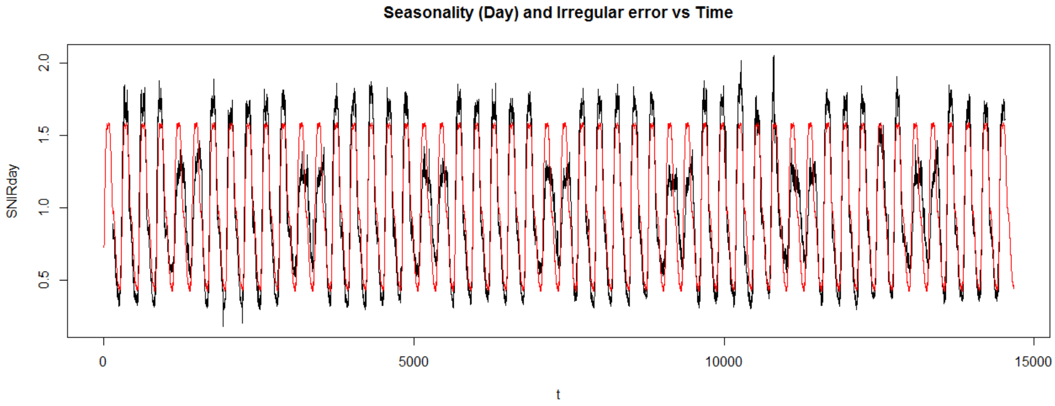


Figure 3: Daily Seasonality Fit plotted on Total Seasonality with Irregular Fluctuations

The last thing I did concerning daily seasons was to deseasonalize my series. To do this, I divided the series by the daily seasonality I found previously so that I would not account for this seasonality more than once.

I then considered weekly seasons. At first, I tried a season length of $(7 \times 24 \times 60) = 2016$ since there are $7 \times 24 \times 60$ minutes in a week and my data was taken every 5 minutes. However, like before, when I plotted my data with each week on top of the last, the peaks did not line up correctly. I then tried a season length of seven times the length of the daily season ($283 \times 7 = 1981$). When I plotted my weeks on top of each other again, they lined up much better (Appendix B: lines 55-57).

The calculations for the weekly seasons were very similar to the daily seasons. I took a centered moving average of my time series (with daily seasons excluded) with an order of the weekly season length (1981). This averaged the first 1981 consecutive points in my series, then it dropped the first observation and added the 1982nd observation to calculate the second average. This continued for the rest of my data points. Every 1981 consecutive points were averaged together to eliminate seasonal and irregular variations. I then divided my series by this centered moving average to get the seasonal and irregular variations (Appendix B: lines 58-59).

Following this, I made a matrix with 1981 columns and populated it with my data so that each column represented one observation time in a week. I averaged each column together to get the average for each observation time, and then normalized these averages so that they summed to my weekly seasonal length. This gave me one week of seasons, so I then repeated this season for the number of weeks my data has (Appendix B: lines 62-66). Figure 4 shows the weekly seasonality I constructed during this process.

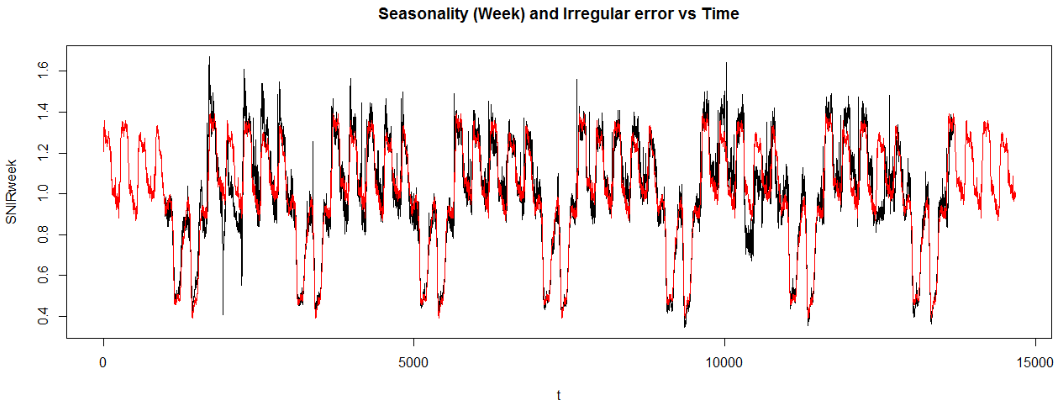


Figure 4: Weekly Seasonality Fit Plotted on Non-Daily Seasonality and Irregular Fluctuations

To deseasonalize my series (that I previously divided by daily seasonality) of weekly seasons, I divided my series by the weekly seasonality I just found.

3.2 Trend

Taking this deseasonalized data, I fit a linear, quadratic, and cubic trend. In each trend, all powers of time were significant (p-values less than 0.05). For simplicity, I opted to use the linear trend of

$$tr_t = \hat{\beta}_0 + \hat{\beta}_1 t$$
$$tr_t = 3806000000 - 10910t$$

with coefficient values generated by R. To detrend my data and consider cycles, I took my deseasonalized series and divided by the trend. This left me with cycles and irregular fluctuations in my data.

3.3 Cycles and Irregular Fluctuations

Looking at my deseasonalized and detrended data in Figure 5, there did not seem to be well-defined cycles. There was a small overall fluctuation, but it was difficult to tell if this was significant without more data. Therefore, I opted to set the $CL_t = 1$ in my overall equation so that I did not account for cycles.

Thus, I assumed the deseasonalized and detrended data I was left with was my irregular fluctuations. Since I could not model irregular fluctuations, I estimated IR_t to be $ir_t = 1$. Additionally, since I approximated cl_t to be 1, my final multiplicative decomposition model was:

$$y_t = (tr_t) \times (sn_t)$$

where tr_t is the approximation for the trend and sn_t is the approximation for the seasons equal to the daily seasonality multiplied by the weekly seasonality. My final model can be seen in Figure 6.

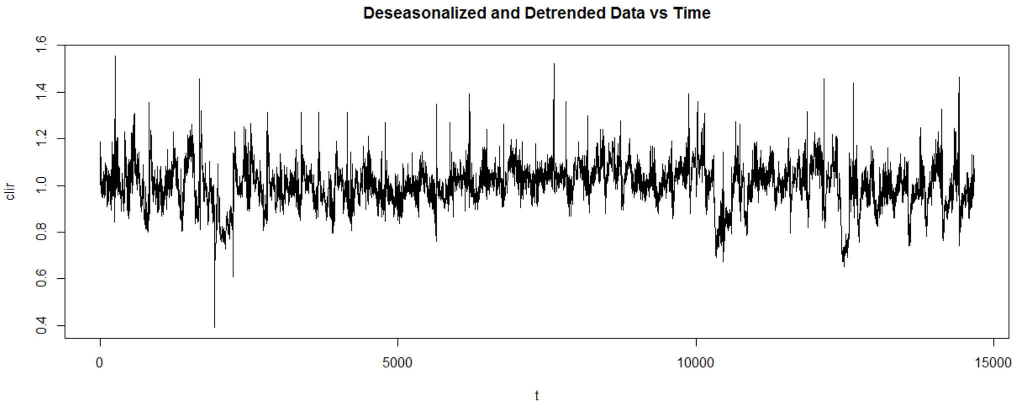


Figure 5: Series without Seasons and Trend: Left with Cycles and Irregular

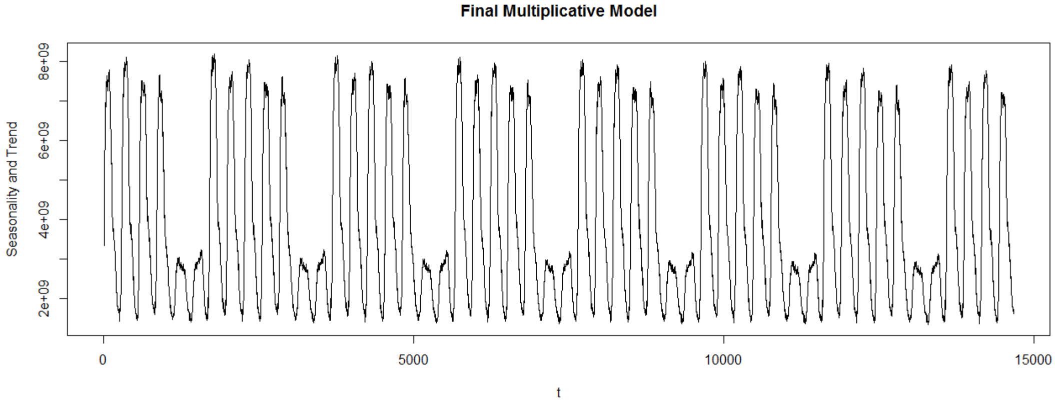


Figure 6: Final Multiplicative Classical Decomposition Model

4 Additive Classical Decomposition

My original plan was to fit a multiplicative Holt-Winters model on my data, but due to the large number of data points in my seasons, I was unable to fit any exponential smoothing model directly. Instead, I performed additive classical decomposition to compare to my previous multiplicative model.

To model my data using additive decomposition, I used

$$y_t = (TR_t) + (SN_t) + (CL_t) + (IR_t)$$

where TR_t represents the trend, SN_t represents the seasonality, CL_t represents the cycles, and IR_t is the irregular error.

4.1 Seasonality

As before, I attempted to model the daily and weekly seasonality first. I began by considering the daily seasons by using daily season length of 283. I took a centered moving average of my time series with an order of the daily season length to eliminate seasonal and irregular variations. I then subtracted my series by this centered moving average to get the seasonal and irregular variations.

I then made a matrix with 283 columns and populated it with my data so that each column represented one observation time in a day. For each column, I averaged all values together to get the average for each time of day. I then normalized these averages to sum to my daily seasonal length. This gave me one day of seasons, so I repeated this season for the number of days my data has. Figure 7 shows the daily seasonality I constructed during this process. Notice that the additive daily seasonality is centered around 0 as opposed to the multiplicative daily seasonality begin centered around 1. I then deseasonalized my series by subtracting the daily seasonality from my series.

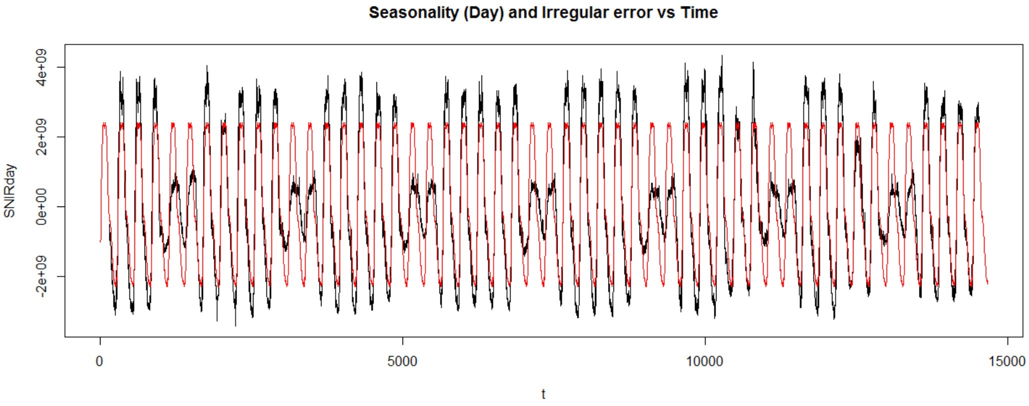


Figure 7: Daily Seasonality Fit plotted on Total Seasonality with Irregular Fluctuations

I then considered weekly seasons. I used a weekly season length of (283 × 7 = 1981). I again took a centered moving average of my time series (with daily seasons excluded) with an order of the weekly season length (1981) and then subtracted this centered moving average from my series. I then made a matrix with 1981 columns and populated it with my data so that each column represented one observation time in a week. I averaged each column together to get the average for each observation time, then normalized these averages so that they summed to my weekly seasonal length. This gave me one week of seasons, so I repeated this season for the number of weeks my data has. Figure 8 shows the weekly seasonality I constructed during this process. I then deseasonalized my series that was previously deseasonalized for daily seasonality by subtracting the weekly seasonality from it.

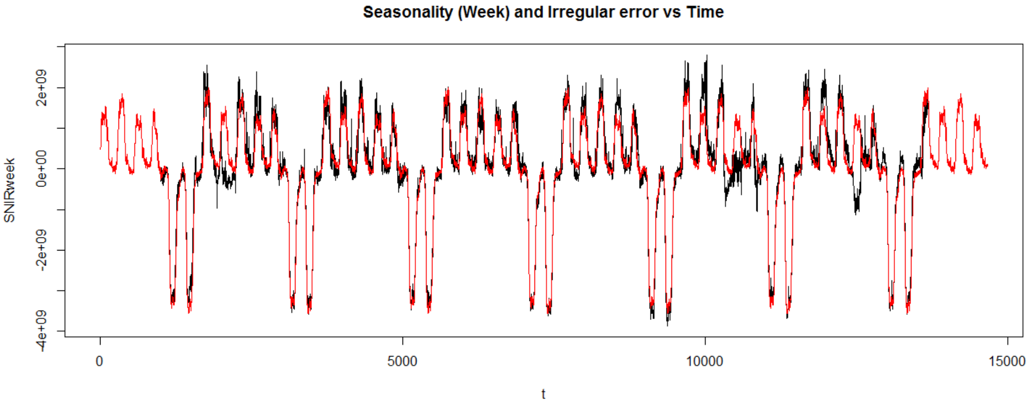


Figure 8: Weekly Seasonality Fit Plotted on Non-Daily Seasonality and Irregular Fluctuations

4.2 Trend

Taking this deseasonalized data, I fit a linear, quadratic, and cubic trend. In each trend, all powers of time were significant (p-values less than 0.05). For simplicity, I opted to use the linear trend of

$$tr_t = \hat{\beta}_0 + \hat{\beta}_1 t$$
$$tr_t = 3886000000 - 15610t$$

with coefficient values generated by R. I then detrended my series by subtracting the trend from my deseasonalized series. This left me with cycles and irregular fluctuations.

4.3 Cycles and Irregular Fluctuations

Looking at my deseasonalized and detrended data in Figure 9, there did not seem to be any well-defined cycles. There was a small overall fluctuation, but it was difficult to tell if this was significant without more data over a longer period of time. Therefore, I opted to set the $CL_t = 0$ in my overall equation so that I did not account for cycles.

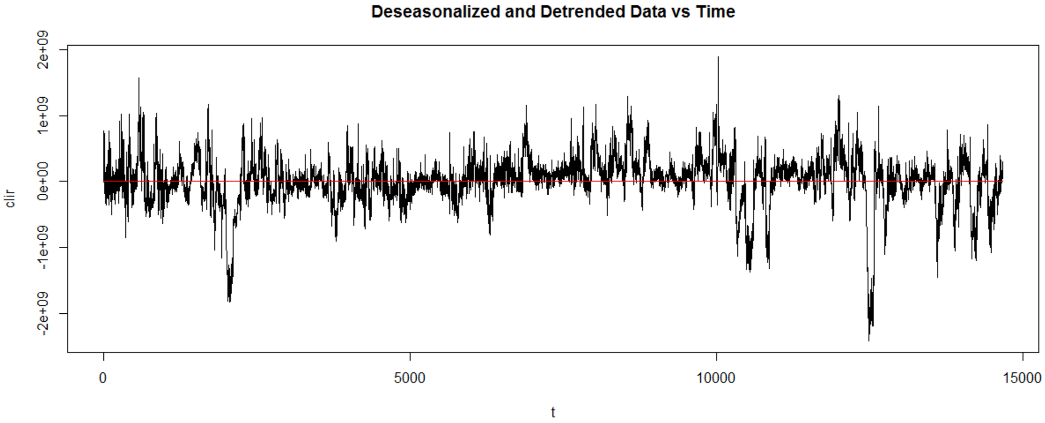


Figure 9: Series without Seasons and Trend: Left with Cycles and Irregular

Thus, I assumed that the deseasonalized and detrended data that I was left with was my irregular fluctuations. Following the formulation for additive decomposition, my final additive decomposition model was

$$y_t = (tr_t) + (sn_t) + (cl_t)$$

where tr_t is the approximation for the trend, sn_t is the approximation for the seasons equal to the daily seasonality multiplied by the weekly seasonality, and cl_t is the approximation for the cycles. Since I could not model irregular fluctuations, IR_t was estimated to be $ir_t = 0$ and so it was not included in the model. Additionally, since I approximated cl_t to be 0, I can leave it out to get my final additive decomposition model as:

$$y_t = (tr_t) + (sn_t)$$

as seen in Figure 10.

5 ARIMA model

Since my series has significant seasonality, I took the first differences of my data to try to get rid of my seasonality. However, the ACF and PACF of the first differences did not die down quickly. I decided to take the second differences, and the ACF seems to cut off after lag 1 or so as seen in Figure 11 and the PACF dies down as seen in Figure 12.

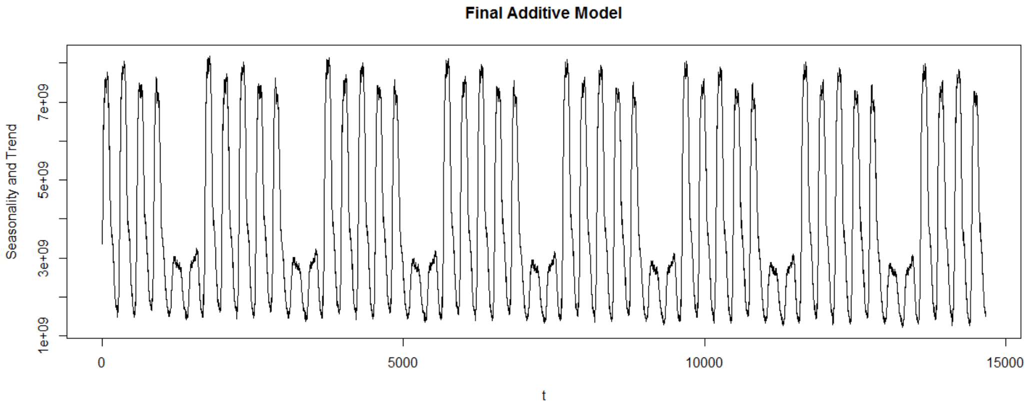


Figure 10: Final Additive Classical Decomposition Model

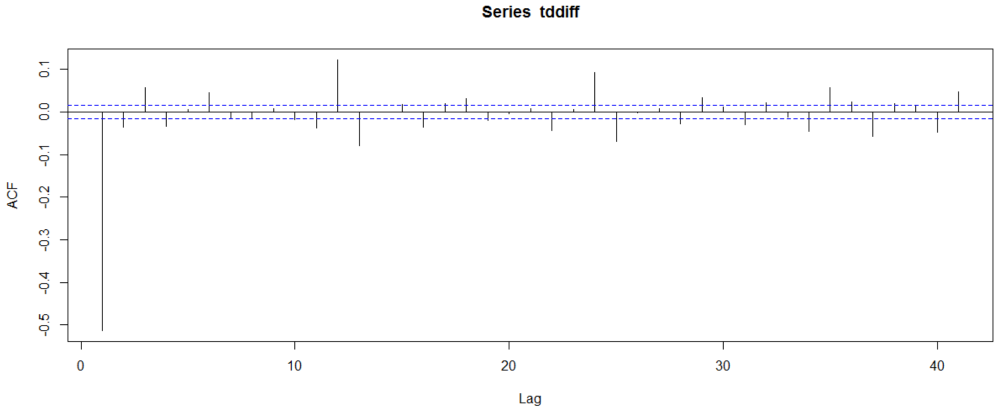


Figure 11: ACF for Second Differences

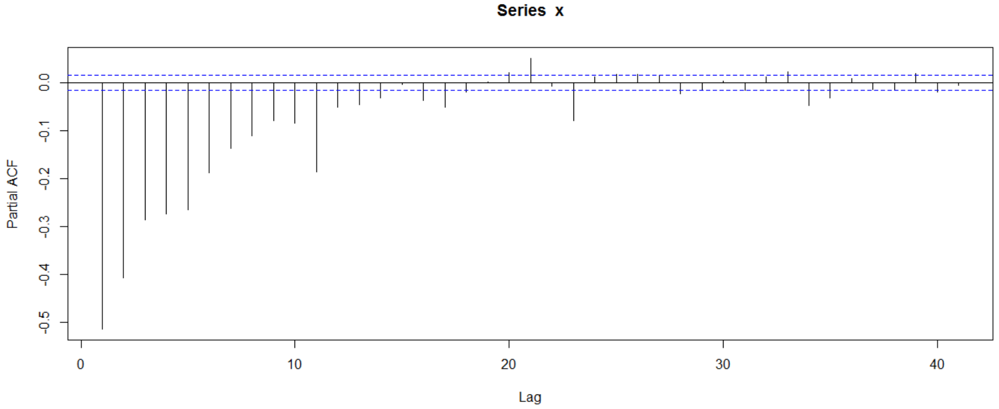


Figure 12: PACF for Second Differences

I first tried to fit an AR(1) model to the second differences. The AIC corrected for this fit was 598464.2. However, performing a Ljung-Box test on this fit showed that the model was inadequate and the residual were not white noise because the p-value was 0.000.

Instead, I tried auto arima on the second differences. The model that was fit was ARIMA(1,2,3) with estimates that can be seen in Table 1. One important thing you can tell from this table is that all of our parameters are significant (each t-value is larger than 2). This fit also had an AIC corrected value of 590832.5, which is less than my previous ARIMA model and so it is better. In addition, the Ljung-Box test on this fit gave a p-value of 0.9463 which means it is highly probable that the residuals were white noise and that our model adequately explained the data.

Table 1: Estimates for ARIMA(1,2,3)

Coefficient	Approximation	T-value
ϕ_1	0.3247	7.4302
θ_1	-1.4502	-33.0341
θ_2	0.4102	7.4717
θ_3	0.0988	6.4155

6 Comparison of Models by Forecasting

Since I took 100 data points from the last part of my dataset before fitting models, I then used the models I formulated to try to predict these 100 data points. In each of the following plots, the black lines plot the actual data that the model had access to. The red lines in the plots are the actual data that the model did not have access to and is trying to predict. The blue lines represent the predictions made by the specified model.

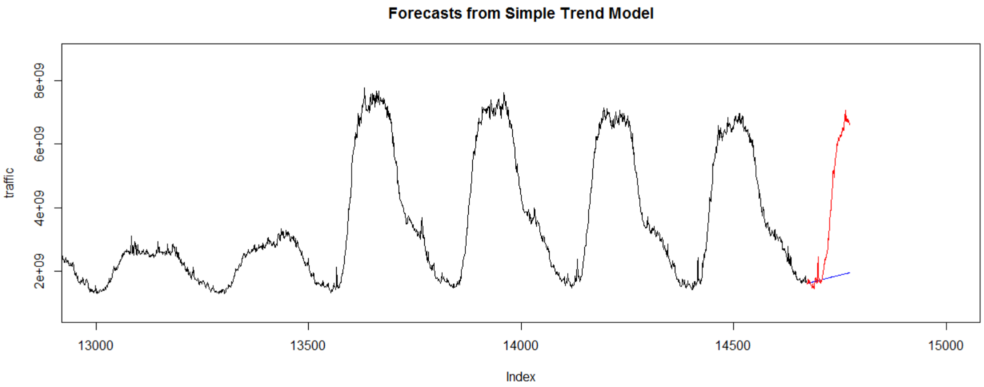


Figure 13: Forecasts using Simple Trend Model

My simple trend model with autocorrelation taken into account was able to use the trend that I fitted throughout its predictions. The first order autoregressive component of the model was less useful. The last residuals ϵT was quite negative, and so when they were used throughout the predictions, they decreased the prediction line created by the trend component. However, this autoregressive component diminishes quickly, so the impact was not as extreme as it could have been. In any case, the model did not seem to predict the data points very well.

Both my multiplicative and additive decomposition models, although traditionally used for descriptive statistics, predicted quite well. I believe this is due to the consistency of the seasonality in the data and the ability of these models to directly account for this seasonality. These two models were the closest to the actual data.

Although my ARIMA model seemed to be able to model my time series quite well, the predictions show that the model is unable to effectively use the previous seasonality of the data in predictions. In this way, the model was unable to accurately predict data points.

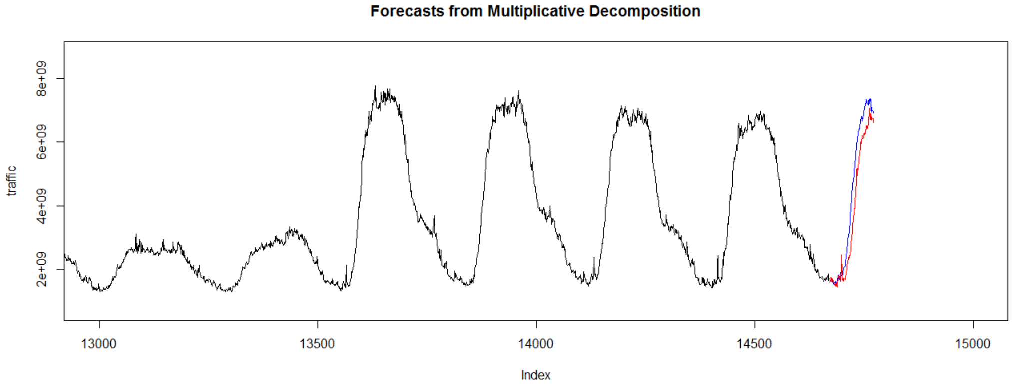


Figure 14: Forecasts using Multiplicative Decomposition Model

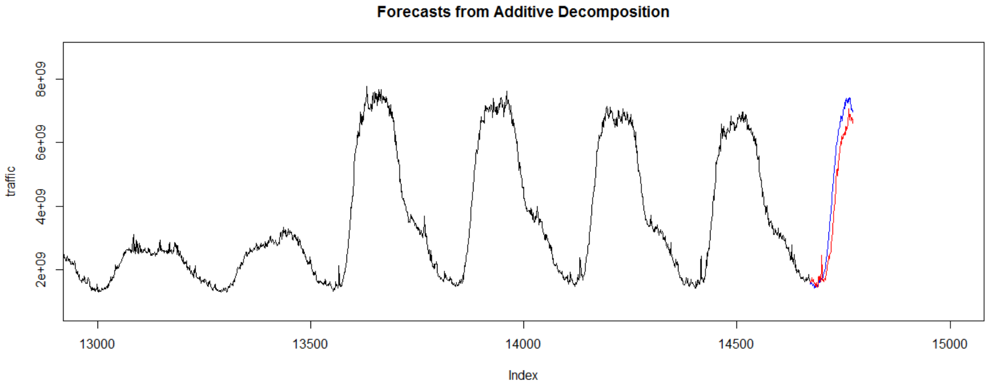


Figure 15: Forecasts using Additive Decomposition Model

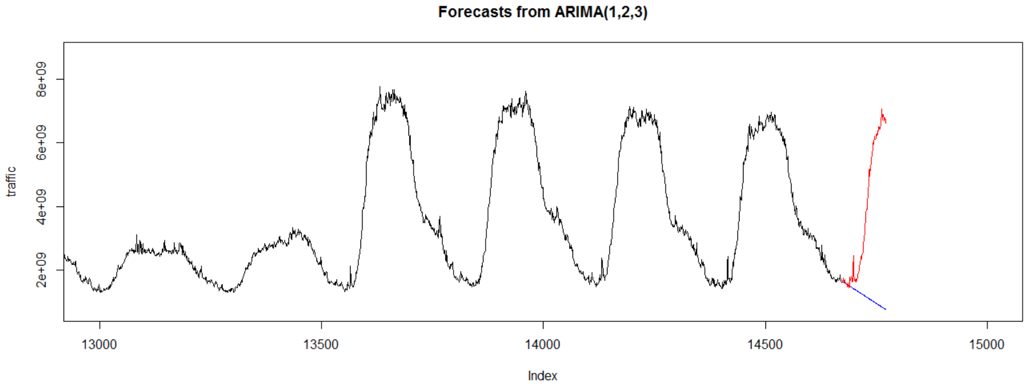


Figure 16: Forecasts using ARIMA Model

In addition to the prediction plots, I have attached a table of the SSE for the prediction interval for each model in Table 2. Overall, the decomposition models clearly predicted better than both the ARIMA and Simple Trend models as can be seen in both the graphs and the SSE values. The ARIMA model predicted the worst, and it seems the Multiplicative Decomposition model predicted the best.

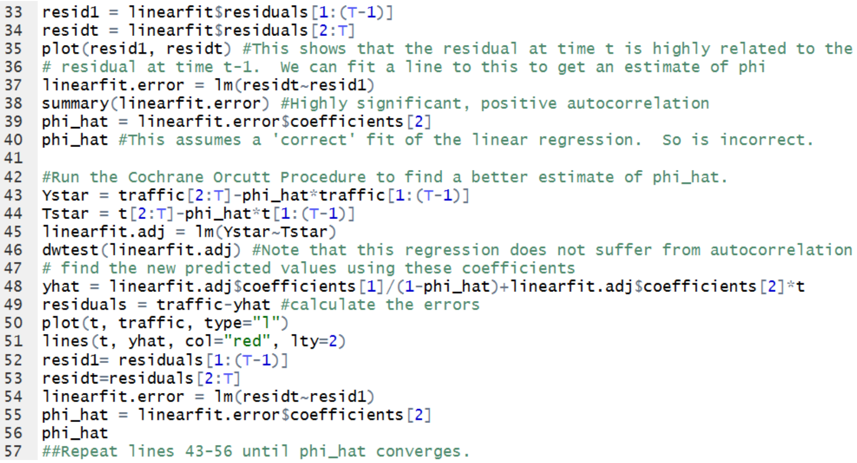
Table 2: SSE Comparison

Model	SSE
Simple Trend	7.752402e+20
Multiplicative Decomp	5.884099e+19
Additive Decomp	5.94801e+19
ARIMA	1.183459e+21

7 Appendices

7.1 Appendix A

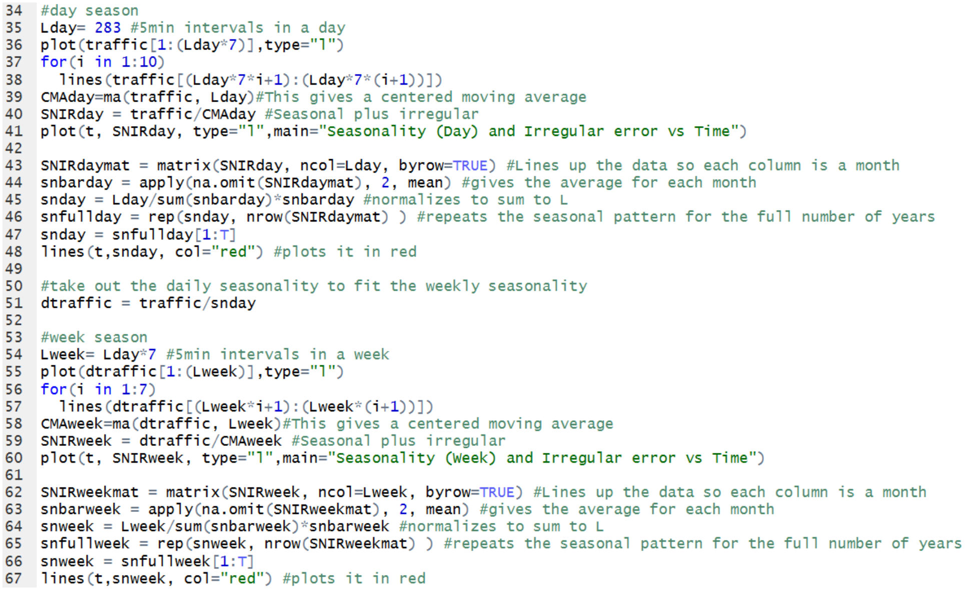
R code dealing with performing the Cochrane-Orcutt procedure.



```
33 resid1 = linearfit$residuals[1:(T-1)]
34 residt = linearfit$residuals[2:T]
35 plot(resid1, residt) #This shows that the residual at time t is highly related to the
36 # residual at time t-1.  We can fit a line to this to get an estimate of phi
37 linearfit.error = lm(residt~resid1)
38 summary(linearfit.error) #Highly significant, positive autocorrelation
39 phi_hat = linearfit.error$coefficients[2]
40 phi_hat #This assumes a 'correct' fit of the linear regression.  So is incorrect.
41 
42 #Run the Cochrane Orcutt Procedure to find a better estimate of phi_hat.
43 Ystar = traffic[2:T]-phi_hat*traffic[1:(T-1)]
44 Tstar = t[2:T]-phi_hat*t[1:(T-1)]
45 linearfit.adj = lm(Ystar~Tstar)
46 dwtest(linearfit.adj) #Note that this regression does not suffer from autocorrelation
47 # find the new predicted values using these coefficients
48 yhat = linearfit.adj$coefficients[1]/(1-phi_hat)+linearfit.adj$coefficients[2]*t
49 residuals = traffic-yhat #calculate the errors
50 plot(t, traffic, type="l")
51 lines(t, yhat, col="red", lty=2)
52 resid1= residuals[1:(T-1)]
53 residt=residuals[2:T]
54 linearfit.error = lm(residt~resid1)
55 phi_hat = linearfit.error$coefficients[2]
56 phi_hat
57 ##Repeat lines 43-56 until phi_hat converges.
```

7.2 Appendix B

R code dealing with finding the daily and weekly seasonality in Multiplicative Decomposition.



```
#day season
Lday= 283 #5min intervals in a day
plot(traffic[1:(Lday*7)],type="l")
for(i in 1:10)
  lines(traffic[(Lday*7*i+1):(Lday*7*(i+1))])
CMAday=ma(traffic, Lday)#This gives a centered moving average
SNIRday = traffic/CMAday #Seasonal plus irregular
plot(t, SNIRday, type="l",main="Seasonality (Day) and Irregular error vs Time")

SNIRdaymat = matrix(SNIRday, ncol=Lday, byrow=TRUE) #Lines up the data so each column is a month
snbarday = apply(na.omit(SNIRdaymat), 2, mean) #gives the average for each month
snday = Lday/sum(snbarday)*snbarday #normalizes to sum to L
snfullday = rep(snday, nrow(SNIRdaymat) ) #repeats the seasonal pattern for the full number of years
snday = snfullday[1:T]
lines(t,snday, col="red") #plots it in red

#take out the daily seasonality to fit the weekly seasonality
dtraffic = traffic/snday

#week season
Lweek= Lday*7 #5min intervals in a week
plot(dtraffic[1:(Lweek)],type="l")
for(i in 1:7)
  lines(dtraffic[(Lweek*i+1):(Lweek*(i+1))])
CMAweek=ma(dtraffic, Lweek)#This gives a centered moving average
SNIRweek = dtraffic/CMAweek #Seasonal plus irregular
plot(t, SNIRweek, type="l",main="Seasonality (Week) and Irregular error vs Time")

SNIRweekmat = matrix(SNIRweek, ncol=Lweek, byrow=TRUE) #Lines up the data so each column is a month
snbarweek = apply(na.omit(SNIRweekmat), 2, mean) #gives the average for each month
snweek = Lweek/sum(snbarweek)*snbarweek #normalizes to sum to L
snfullweek = rep(snweek, nrow(SNIRweekmat) ) #repeats the seasonal pattern for the full number of years
snweek = snfullweek[1:T]
lines(t,snweek, col="red") #plots it in red
```