STA 137 Project

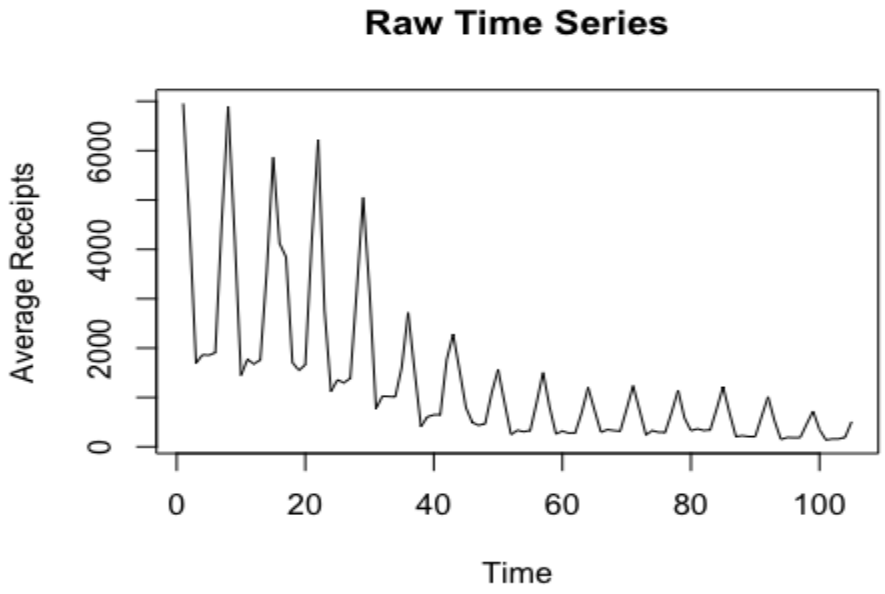
Troy Lui

Due: December 13, 2018

I. Introduction

The goal of the project is to fit a time series to a model predicting the daily average receipts per theater for the movie Chicago given time. To start, a quick description of the data explains that it is a time series that picked up a daily average of receipts per theater for the movie Chicago. The time span of the sample taken was from January 3, 2003 to April 23, 2003. This data is considered a time series since the data collected is spaced between successive and equal amounts of time, or, each data point collected is observed from a specific day from January 3, 2003 to April 23, 2003. Time series modeling is important to this data since it can forecast or backcast points outside the interval using the model created from the interval; however, do note that given that the data is about a movie and how many receipts it receives on average daily, backcasting past December 27, 2002 (movie release date) will give false predictions since the movie's true daily average will fall to 0. In the same way, movie theaters may eventually stop broadcasting the movie Chicago, which will also take to the average to 0. Lastly, of typical time series modeling, it is good to note that the farther forecasting into the future or backcasting into the past, the less accurate the model will be to the true daily average of receipts.

II. Materials and Methods

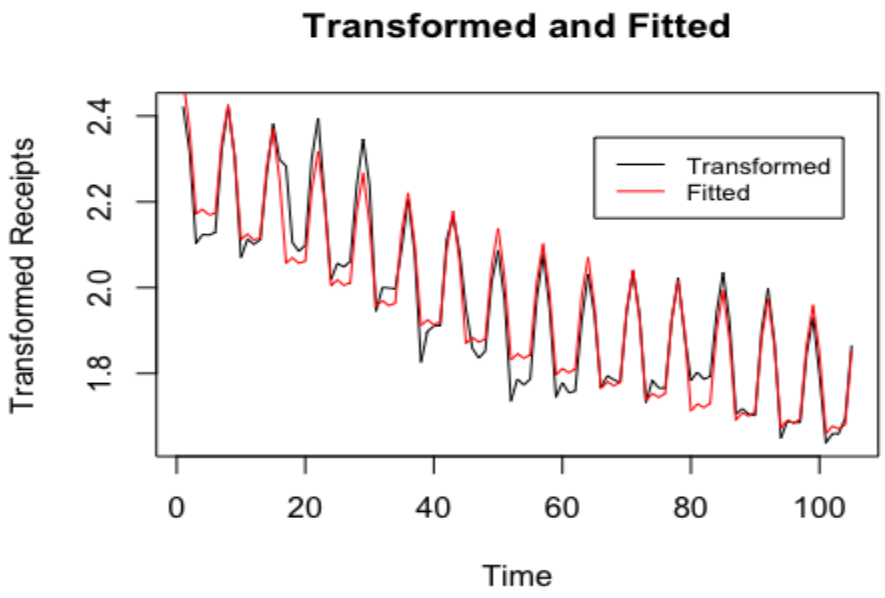


When taking a first look at the raw time series above, the data does not appear to have constant mean or variance. Variance tends to get smaller and smaller as time goes on. Through eye diagnostic, there seems to be some type of trend and seasonality present

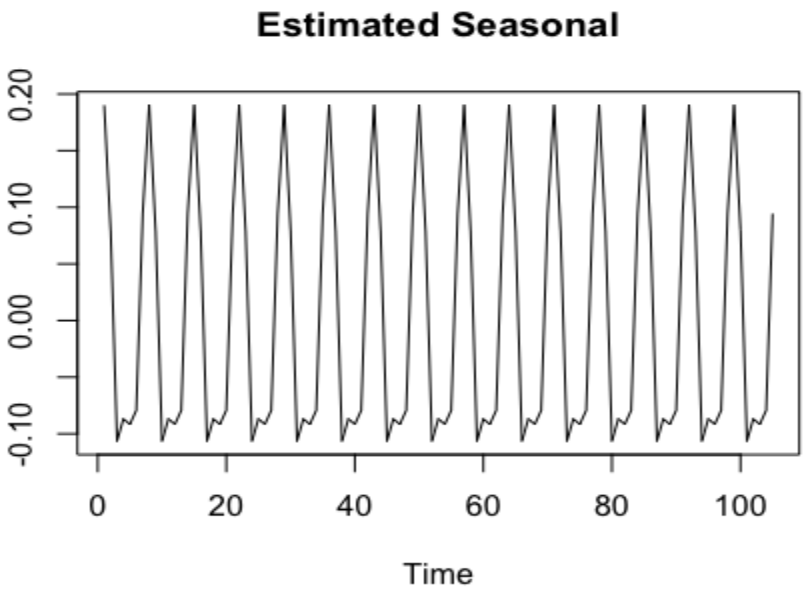
(roughly every 7 days). With this in mind, the model being taken into consideration includes a times series modeling with trend, rough, and seasonality. A general model would be *obs = rough + trend + seasonality*, which considers all three. The general strategy to model this will be to estimate the seasonality and polynomial trend then use trend and seasonality to estimate the rough. From the rough, auto.arima( ) will be used to find the best model for it and its coefficients.

III. Results

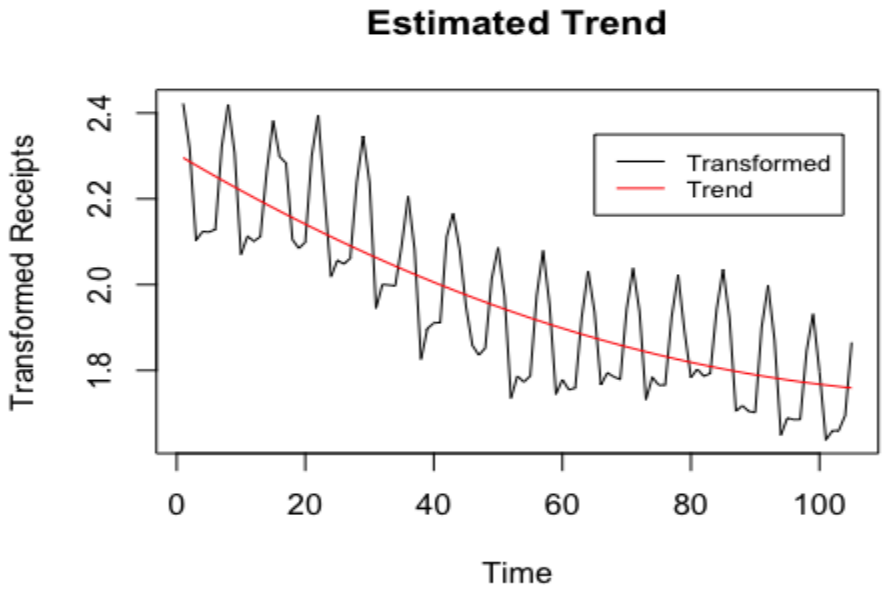
As mentioned, the raw time series plot is harder to work with since the mean and variance are not constant. In light of this, it is necessary to choose an appropriate Box-Cox transformation and transform the data accordingly. When picking the optimal lambda to transform the data, every value in increments of 0.1 from -2 to 2 were taken and it was found that a lambda of 0.1 maximized R^2 the most. When selecting the type of transformation, X_t^{lambda} and $\ln(X_t)$ [where X_t represents the series] were being considered; however, since lambda $\neq 0$, X_t^{lambda} was used to transform the data.



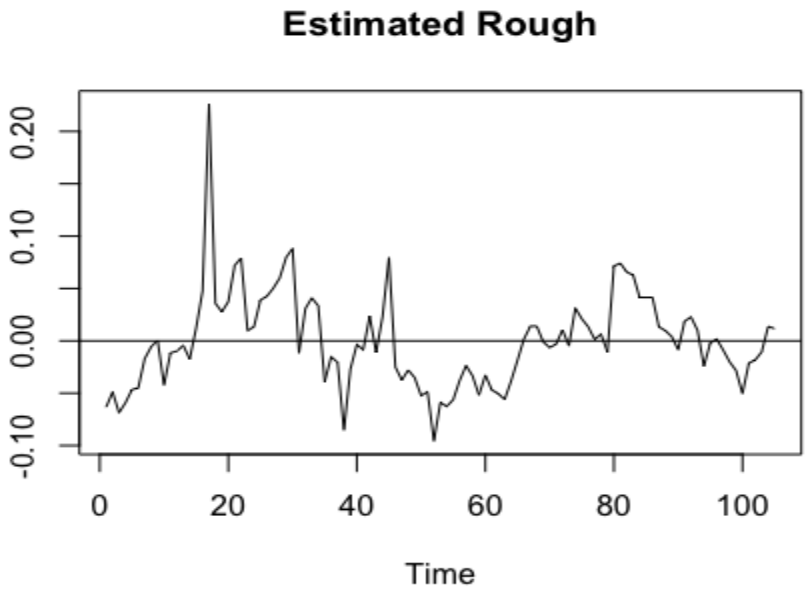
The data was then transformed through lambda = 0.1 and received the transformed plot above. As shown by the black line in the plot above, the transformed data is of better variance when compared to the raw time series data. Therefore, the transformed data will be worked with for better modeling.



After transforming the data, an estimation of the trend and seasonal components were taken. When figuring out the value of days for the seasonality, an eye diagnostic was done to determine that there was that seasonality occurred over a course of 6 to 7 days at a time. To test this, multiple values were used and, without overfitting, the seasonal value of 7 seemed to fit the data the best. Using this value, seasonal components of 0.19015608, 0.07969558, -0.10655403, -0.08673700, -0.09173823, -0.0792244, and 0.094402 were found. As shown above, the estimated seasonal plot shows seasonality over the 105 observations.

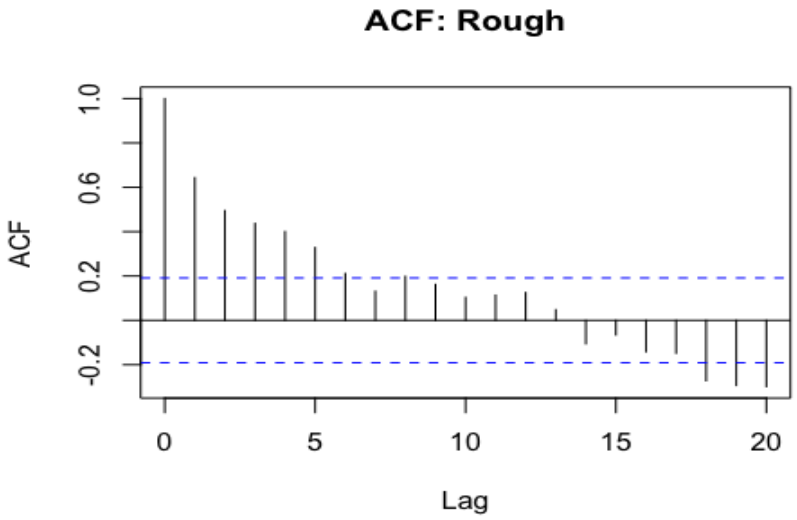


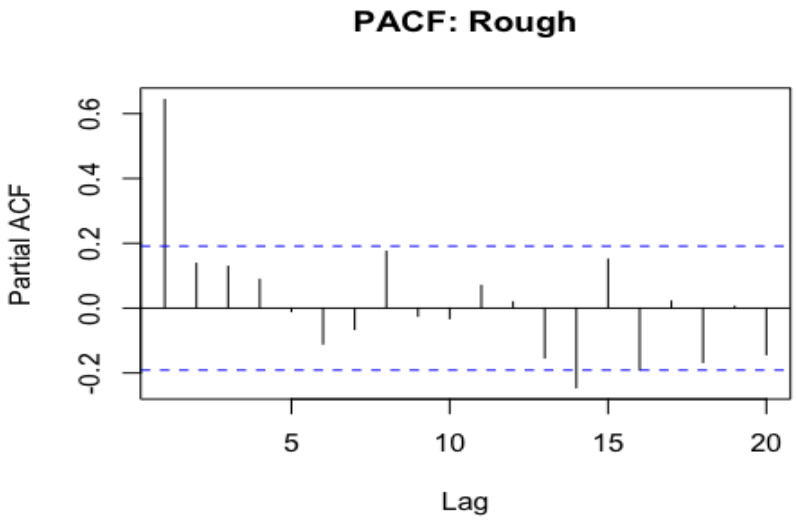
After taking seasonality of the transformed data, the polynomial trend was taken of the data as well. As shown above, the estimated trend plot shows the estimated trend over the transformed time series. When taking a look, the estimated polynomial trend fits the transformed data fairly well. All in all, when taking a look at the estimated trend and estimated seasonality, we can see in the transformed and fitted values plot that the combination of degree 2 of polynomial trend and seasonal value of 7 fits the transformed data fairly well.

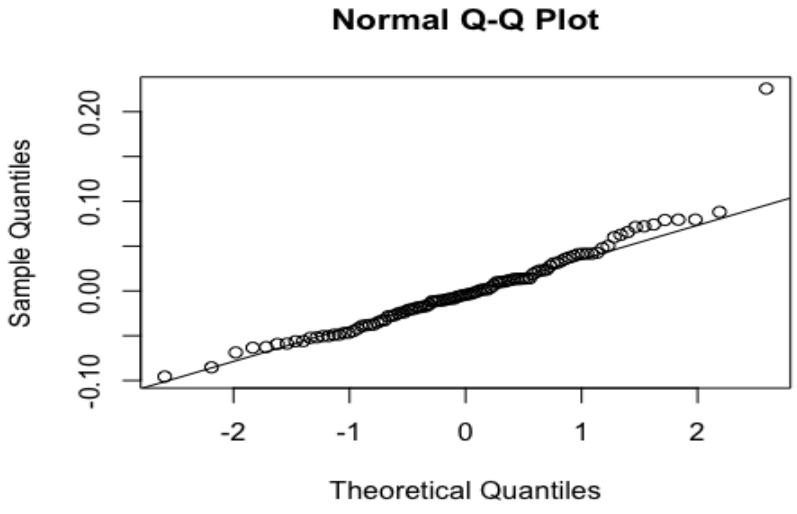


Last of the model, the rough was estimated by taking the transformed data's values and subtracting them from seasonal components and trend values. Since the model being fitted equals obs = rough + trend + seasonal, this method will estimate the rough. Shown above is the estimated rough and at first glance, aside from the large value around time 17, the estimated rough appears to be stationary with equal mean and equal variance. When testing for stationarity, the KPSS test was used, which has a null hypothesis of stationary and alternative hypothesis of not stationary. The p-value of the test came out to 0.1, which means that at an alpha level of 0.05, we fail to reject that the estimated rough is stationary.

In addition, the normal probability plot shown below shows the assumption of normality is reasonable for the estimated rough. When taking a look at the ACF and PACF plots (also next page), the ACF plot lags off while the PACF cuts off at lag 1. Initial diagnostics suggest that a AR(1) model is appropriate for the estimated rough (though may change). The ACF plot also shows significant lags 1-5, which suggest that the estimated rough is not independent and identically distributed (more analysis is needed). Lastly, when using the Box-Ljung test, the null hypothesis is independence and the alternative hypothesis is not independent. From the test, the p-value was approximately 2.2e-16, which is significantly low. Therefore, the test rejects that the estimated rough is independent.

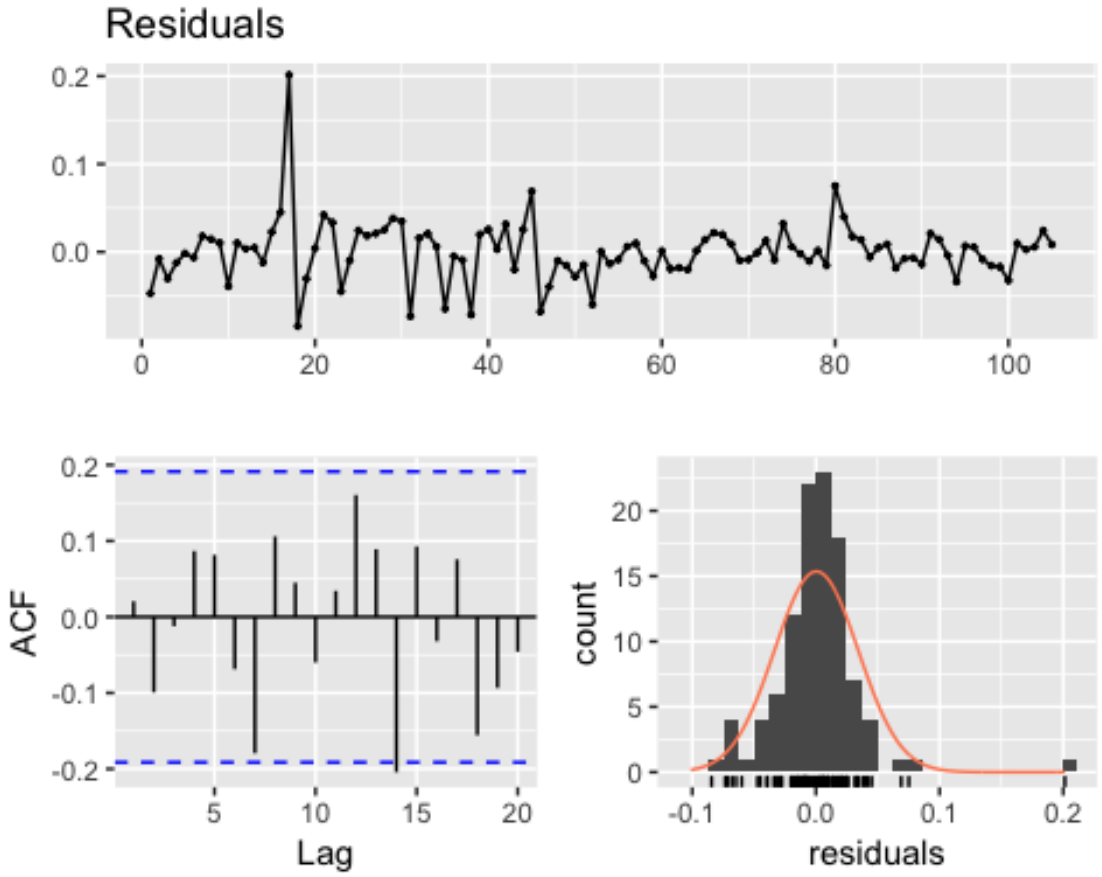

ACF: Rough
ACF
1.0
0.6
0.2
-0.2
0
5
10
15
20
Lag

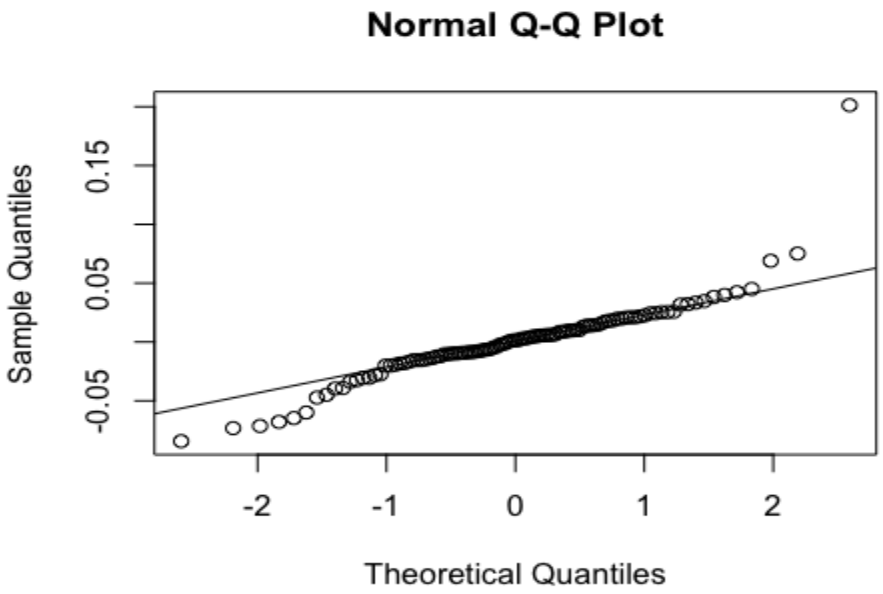

PACF: Rough
Partial ACF
0.6
0.4
0.2
0.0
-0.2
5
10
15
20
Lag


Normal Q-Q Plot
Sample Quantiles
0.20
0.10
0.00
-0.10
-2
-1
0
1
2
Theoretical Quantiles

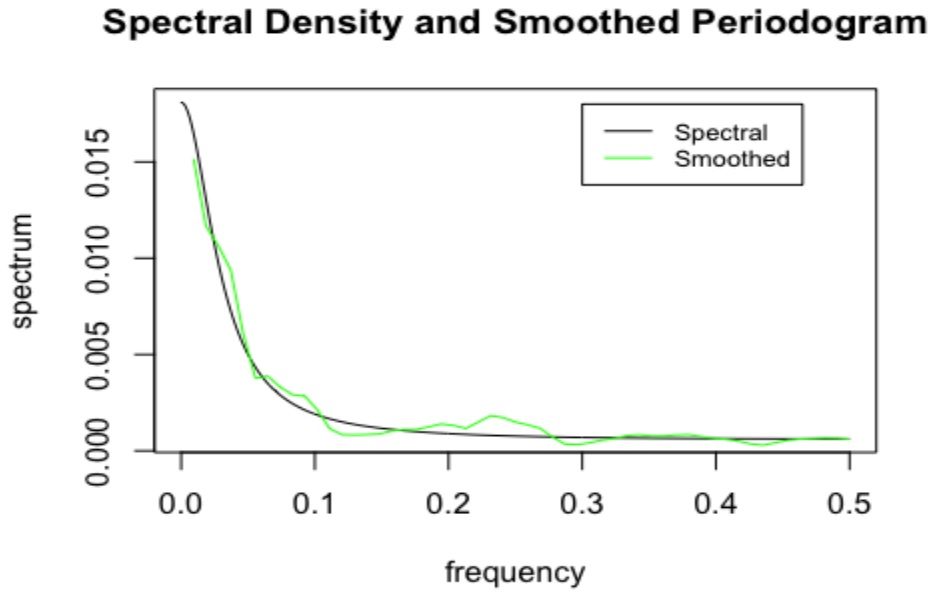
After receiving the estimated rough and conducting diagnostics, the function auto.arima( ) was used to find the best model that fits the time series. From the function, it was found that the estimated rough frame's best model is an ARIMA(1,0,1) with zero mean. The coefficients of the model included phi = 0.832 with a standard error of 0.085 and theta = -0.3363 with a standard error of 0.1526. Sigma^2 was estimated as 0.001159. Do note that the initial diagnostics suggesting an AR(1) model from the ACF and PACF plots were incorrect after further analysis.

Doing a few more diagnostics, it was found that the residuals of the estimated rough roughly follow white noise from the ACF of the residuals. In addition, from the histogram showing residual distribution and normal probability plot (next page), the assumption of normality is not unreasonable. Lastly, the Box Ljung test has a p-value of 0.54, which assumes independence. All in all, this suggests that forecasts from the model produced will do decently well in predicting the true value.

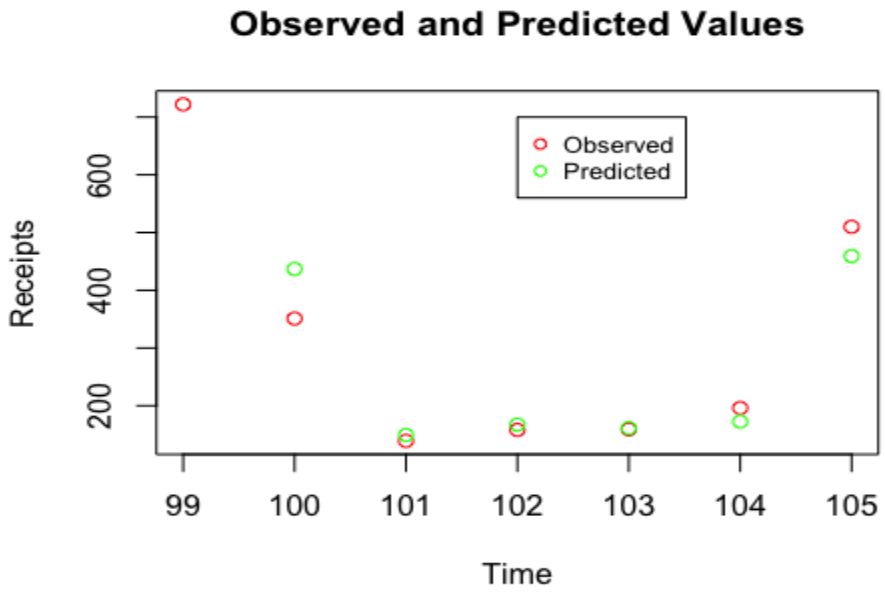




The spectral density function of the fitted ARIMA(1,0,1) model and smoothed periodogram (6 neighbors) is shown below. The smoothed periodogram actually estimates the spectral density fairly well and even at its worse estimating (around 0.2 to 0.3), it doesn't vary from the spectral density function too much. One thing to note about the spectral density function is that it peaks around 0 since the AR coefficient, phi, is 0.832.



In the plot below, the final model was refit to the estimated rough minus the last seven observations. From the estimated rough, a forecast of those seven observations were made, added to forecasted season and trend, and then untransformed by taking the forecasts to the tenth power to receive the true predictions. These forecasts were plotted against the actual observations and the plot below was the result. As shown, the forecasted points through the final model produced a fairly accurate representation of the actual observed data. To note, linear extrapolation was used for forecasting the trend.



IV. Conclusion and Discussion

In summary, the final model fitted is an ARIMA(1,0,1) with zero mean and:

$$Y_t = (m_t + s_t + X_t)^{10}$$

[m_t = transformed trend, s_t = transformed seasonality, X_t = transformed rough]

$$\text{where } X_t = \Phi X_{t-1} + \theta X_{t-1}$$

[$\Phi = 0.832$, $\theta = -0.3363$]

Some main take-aways from the process of getting the final model include that estimating the trend and estimating the seasonality component of the time series were necessary to estimate the rough. From this, a rough was fitted to a model using the auto.arima( ) function. An interesting take away from fitting the model using auto.arima versus fitting a model using ACF and PACF diagnostics is that ACF and PACF plots give a general idea of what models to fit;

however, the model that they may seem to suggest may not be the best model in actuality. Therefore, further analysis outside of just the ACF and PACF plots are needed to verify since eye diagnostics are not always reliable. Lastly, when fitting the final model, trend, seasonality, and the rough were forecasted separately, added, and then untransformed to fit the original data. The transforming of the data makes it easier to work with but one must always be mindful in untransforming the data afterwards. All in all, the final model fitted did a nice job in predicting the actual points observed in the data set.

Appendix

```
chic = read.delim("/Users/Troy/Downloads/chicago.txt")
colnames(chic) = c('receipts','date')
ts.plot(chic$receipts, ylab = "Average Receipts", main = "Raw Time Series")
source("/Users/Troy/Downloads/trndseas.R")
tchic = chic$receipts
tchic = as.numeric(tchic)
n = length(tchic)
lambdas = seq(-2,2,0.1)
s = 7
deg = 2
mod = trndseas(tchic,s,lambdas,deg)
str(mod)
mod$lamopt
mod$season
ts.plot(tchic^mod$lamopt, main = "Transformed and Fitted", ylab = "Transforme
d Receipts")
lines(mod$fitted, col = "red")
legend(65, 2.35, legend = c("Transformed", "Fitted"), col = c("black", "red")
, lty = 1, cex = 0.8)
ts.plot(tchic^mod$lamopt, main = "Estimated Trend", ylab = "Transformed Recei
pts")
lines(mod$trend, col = "red")
legend(65, 2.35, legend = c("Transformed", "Trend"), col = c("black", "red"),
lty = 1, cex = 0.8)
seas = rep(mod$season, length.out = n)
seastrend = seas + mod$trend
transframe = tchic^mod$lamopt
roughframe = transframe-seastrend
ts.plot(roughframe, main = "Estimated Rough", ylab = ''); abline(h=0)
ts.plot(seas, main = "Estimated Seasonal", ylab = '')
library(tseries)
acf(roughframe, lag.max = 20, main = "ACF: Rough")
pacf(roughframe,lag.max = 20, main = "PACF: Rough")
qqnorm(roughframe); qqline(roughframe)
Box.test(roughframe, lag = 10, type = 'Ljung-Box')
kpss.test(roughframe)
library(forecast)
roughmod = auto.arima(roughframe, stepwise = F, approximation = F)
res = roughmod$residuals
checkresiduals(res)
Box.test(res, type = "Ljung-Box", lag = 10)
shapiro.test(res)
qqnorm(res); qqline(res)
library(astsa)

coef.ar1_1 = 0.832
coef.ma1_1 = -0.3363
sigma2 = 0.001159
```

```r
mod_spec = arma.spec(ar=c(coef.ar1_1), ma = c(coef.ma1_1), var.noise = sigma2
, log = 'no', main = 'Spectral Density and Smoothed Periodogram')
mod_smooth = spec.pgram(roughframe, log = 'no', spans = 6, main = '', col = "
red", plot = F)
lines(mod_smooth$freq, mod_smooth$spec, col = "green")
legend(0.3, 0.018, legend = c("Spectral", "Smoothed"), col = c("black", "gree
n"), lty = 1, cex = 0.8)
m = floor(n/2)
spans = (1:(m-1))*2+1
pgrm_raw = spec.pgram(transframe, log='no', plot= F)$spec
Q <- numeric(length(spans))
for(j in 1:length(spans)){
  L <- spans[j]
  pgrm_smooth <- spec.pgram(transframe, spans=L,log="no", plot=F)$spec
  Q[j] <- sum((pgrm_smooth - pgrm_raw) ^ 2) + sum((pgrm_raw)^2)/(L-1)
}
#plot(x = spans, y=Q, type = 'b')
#spans[which.min(Q)]
##25 span
#last 7 days gone
rough_no7 = roughframe
rough_no7 = rough_no7[-c(99,100,101,102,103,104,105)]
mod_ARMA23 = arima(rough_no7, order = c(1,0,1))

#forecast trend
h = 7
library(Hmisc)
ind_old = 1:98
ind_new = c(99,100,101,102,103,104,105)
trend_f = approxExtrap(ind_old, mod$trend, xout= ind_new)$y

#forecast season
season_f = rep(mod$season, length.out = n)[-(1:98)]

#make sure to untransform later
fcast = predict(mod_ARMA23, n.ahead = 7)
x_fc = fcast$pred

truepredicted = x_fc+trend_f+season_f
untransform_predict = truepredicted^10
plot(c(99,100,101,102,103,104,105), chic$receipts[99:105], col = "red", main
= "Observed and Predicted Values", xlab = "Time", ylab = "Receipts")
points(c(99,100,101,102,103,104,105),untransform_predict, col = "green")
legend(102, 700, legend = c("Observed", "Predicted"), col = c("red", "green")
, pch = 1, cex = 0.8)
```