Time Series FInal Project

Truc Le

3/8/2021

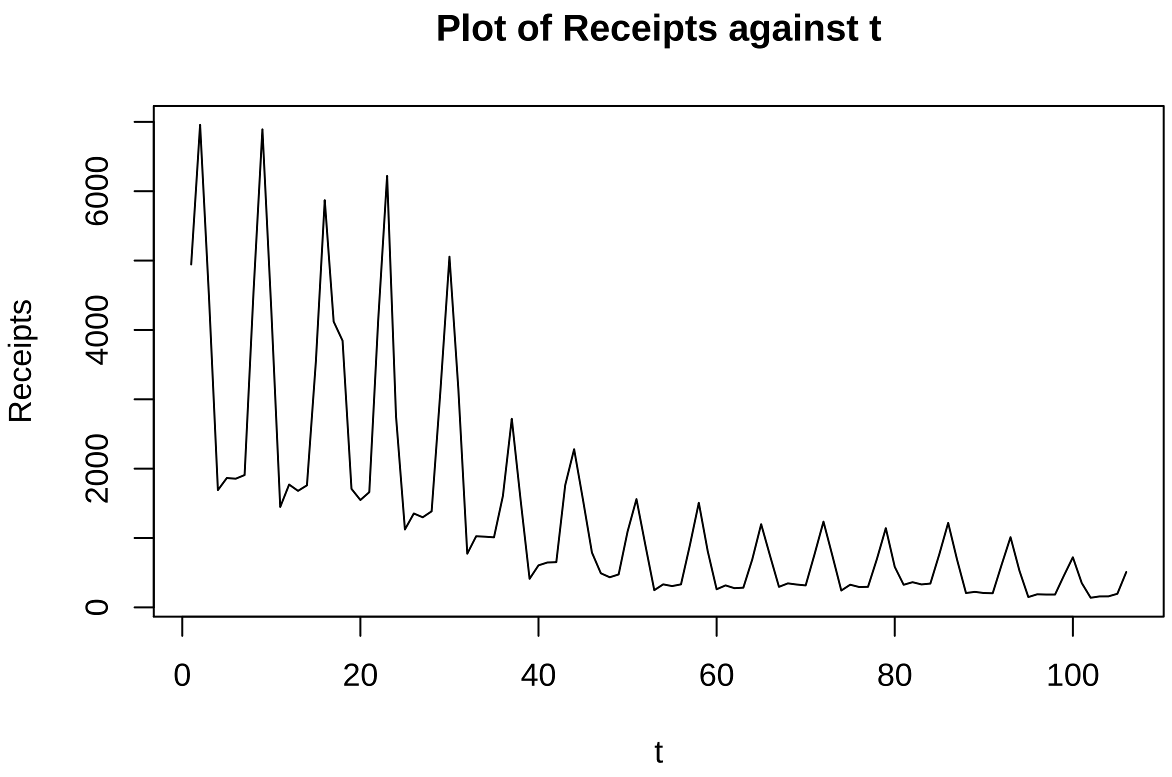
I Introduction

The goal of this report is to fit a time series model to the Chicago dataset in order to predict the daily average reciepts per theatres for the movie Chicago for the given four and a half months time period. The dataset contains the daily average reciepts per theatres for the movie Chicago from January 3, 2003 up until April 18, 2003, and it is considered a time series since the data was taken over a period of time within equaled spaced increments. Time series modeling is important for this dataset as it can forecast the data points outside of the given time intervals by using model it already created from the current data points it possess. However, there are constraints to this time series prediction model, it is illogical to forecast before the movie was released and after it is no longer in theatre as it would give inaccurate predictions for the daily average reciepts per theatres. In addition, like any time series model, it should be noted that any forecasting farther away from the given model would have a higher probability of incurracy in predicting the daily average reciepts per theatres for the movie Chicago.

II. Materials and Methods: Description of the Data and Methods Used in the Analysis

For this project, we were provided a file (`Chicago.txt`), which contains daily average receipts (`Receipts`) per theater for the movie Chicago. The data covers the time period January 3, 2003 to April 23, 2003 (time $t = 1, ..., 106$).

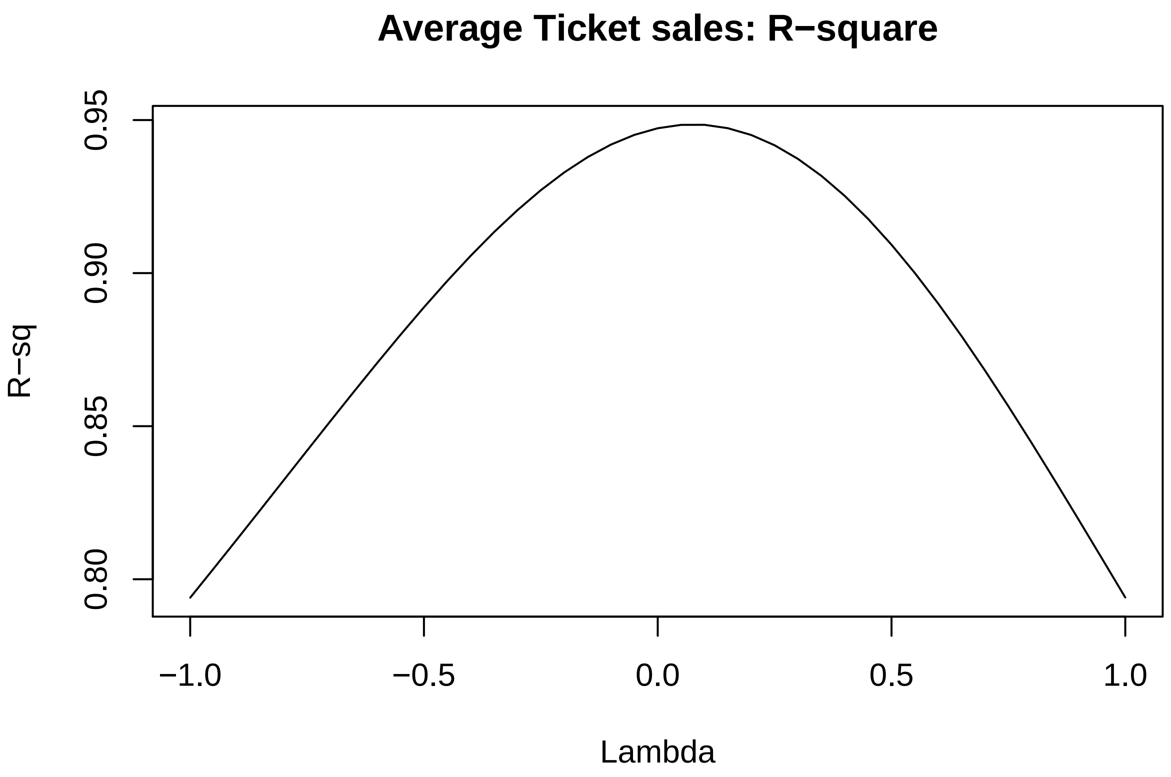
A simple plot of `Receipts` against `t`:



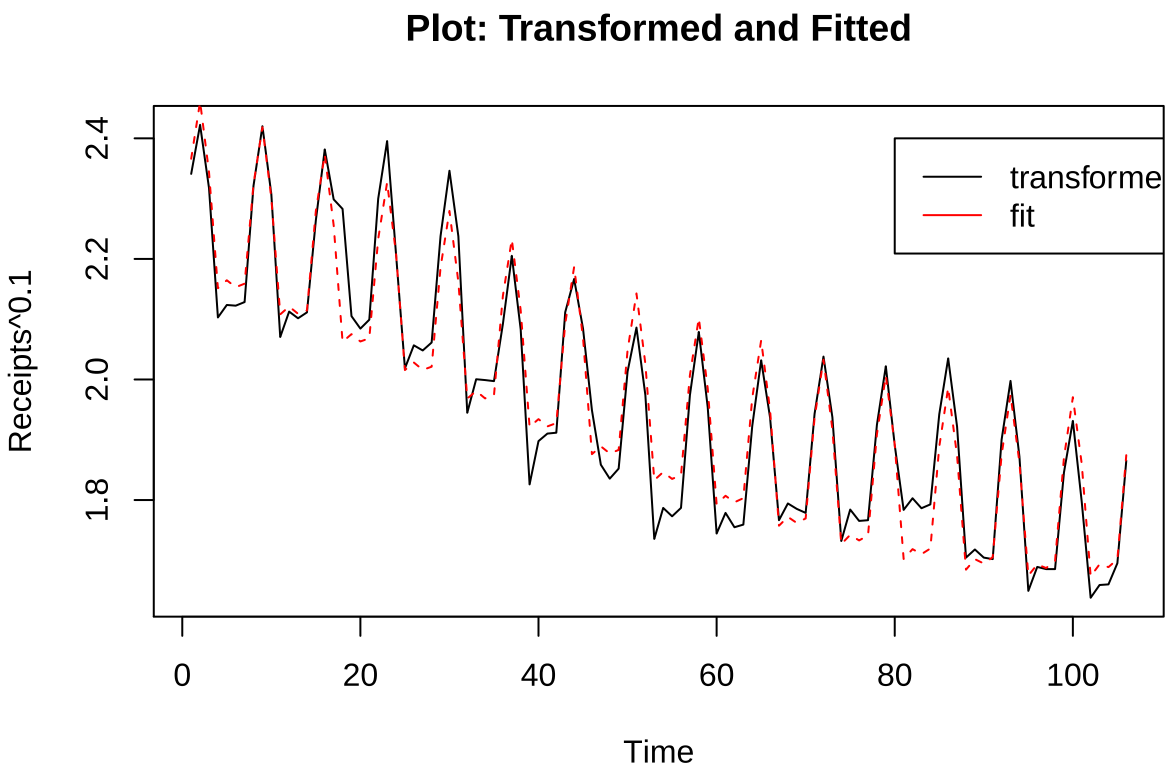
From the `Plot of Receipts against t`, it is observed that `Receipts` seems to have trend, seasonal, and rough components, with a cycle of rougly 7 days. A noticeable problem observed from the above plot is that the spread of `Receipts` changes over time (i.e. the variance is not constant). To address this problem, a Box-Cox transformation will be apply to the dataset, which will make the spread of `Receipts` more constant overtime.

To fit a time-series model to the dataset, we will use the basic model of $Y_t = m_t + s_t + X_t,\ t = 1, ..., n = 106$, where Y_t denotes `Receipts`, m_t denotes the trend, s_t denotes the seasonal component, and X_t denotes the rough part. To estimate m_t and s_t, the function `trndseas()`, which was provided in the file `trndseas.R`, will be used. After obtaining the $\hat{m}_t$ and $\hat{s}_t$ (the estimated trend and seasonal effects respectively),the rough part is then estimated by subtracting the estimated trend and seasonal effects from the observed values (i.e. $\hat{X}_t = Y_t - \hat{m}_t - \hat{s}_t$). After obtaining the estimated rough, the `auto.arima()` function will be used to find the best ARMA model. This ARMA model will then be used to estimate the rough in the time-series prediction model.

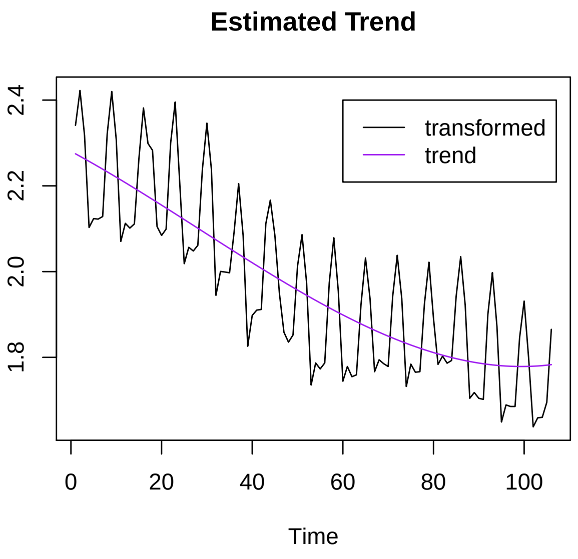
III. Results: Explanation of the Results in our Analysis

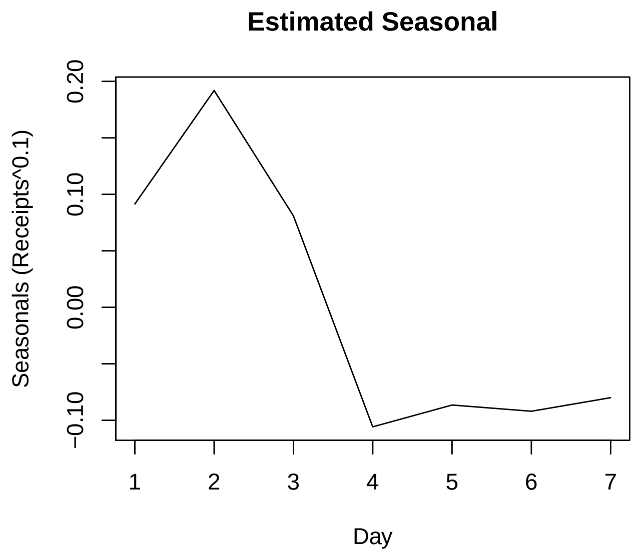


As previously discussed, due to the inconsistent variance observed from the `Plot of Receipts against t`, it is necessary to perform a Box-Cox transformation on the original data before a time series model could be fitted. In order to choose an appropiate Box-Cox transformation, the optimal lambda must be picked first. To find the optimal lambda value, a `Lambda vs. R-squared plot` is contructed to approximate the interval of where the optimal lambda is in. As seen in the `Lambda vs. R-squared plot`, it is observed that the optimal lambda is within the interval of [-1,1], therefore, every values in increment of 0.05 from -1 to 1 was taken to find the optimal lambda values using the `trndseas()` function. The result from the `trndseas()` function shows that the optimal lambda value is 0.1. The data is transformed using the X_t^λ transformation model since the calculated optimal lambda in this case is not equal to 0.

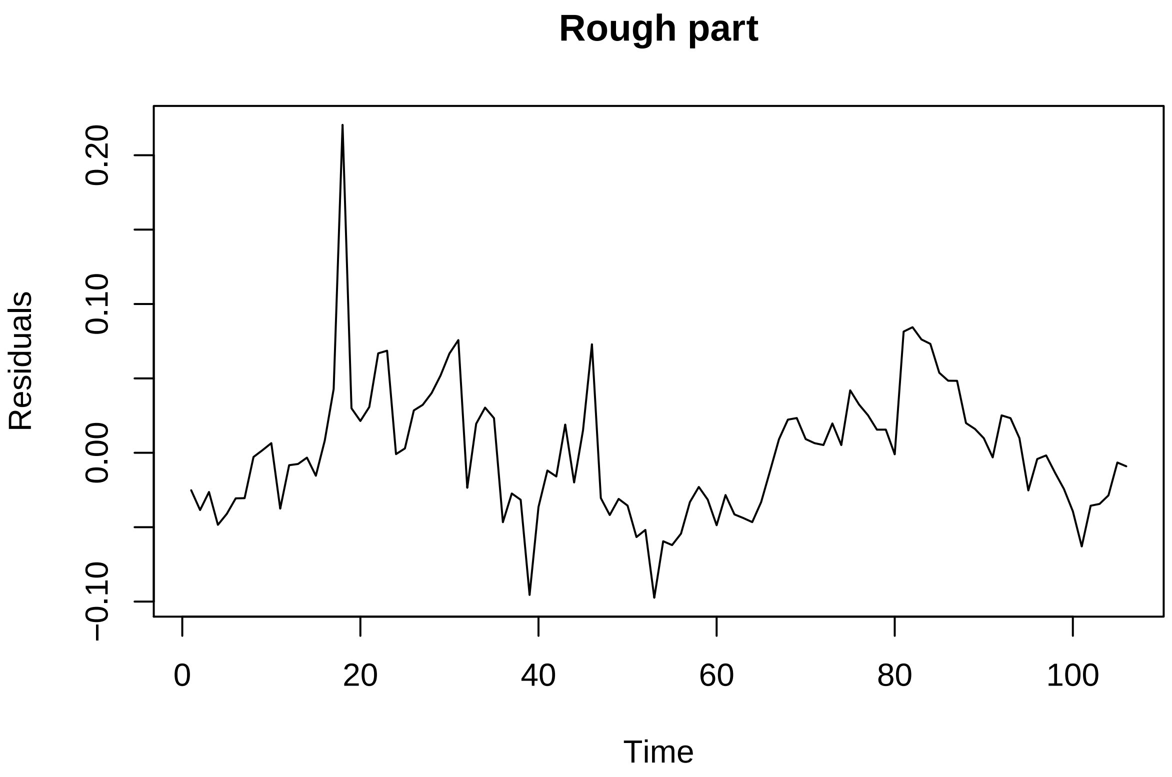


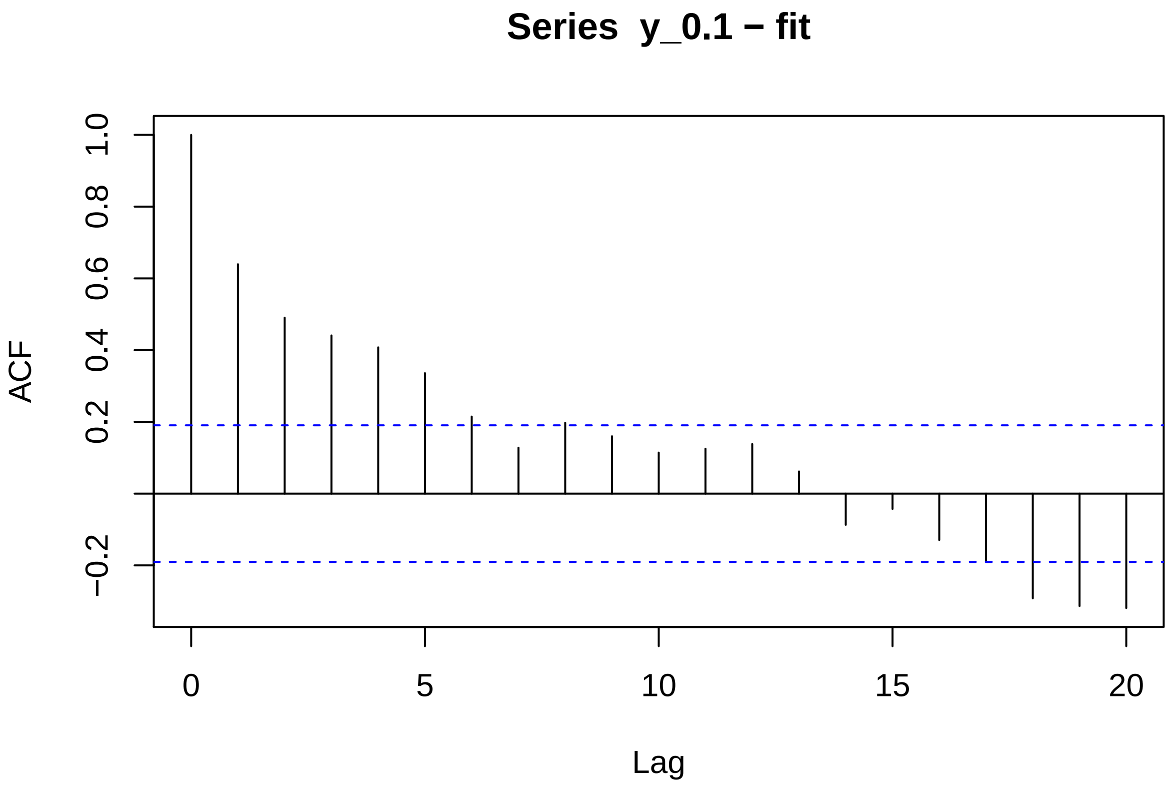
As observed from the black line (transformed) in `Plot: Transformed and Fitted` the transformed data gives a more consistent variance when compared to the original data as seen in the `Plot of Receipts against t`. Therefore, this newly transformed data will be use to fit the time series model moving forward.



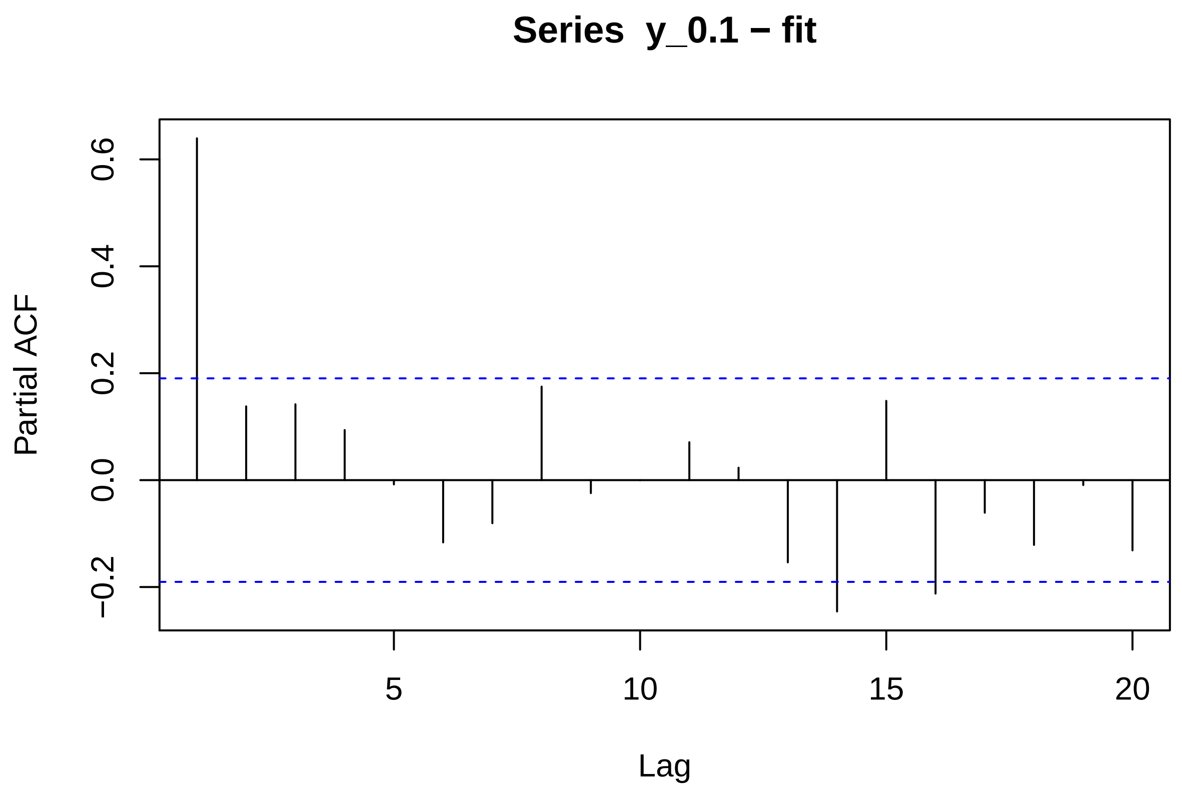


The next step after transforming the data using the optimal lambda value of 0.1, the trend and the seasonality components of the transformed dataset, m_t and s_t, were estimated using the `trndseas()` function. As observed in the `Estimated Trend` plot above, the estimated polynomial of degree 3 trend line fits the transformed data very well, as it lays superimposed over the middle of the transformed data while following the decreasing trend that the transformed line go through overtime. In addition, from the `Estimated Seasonals for 1/y^0.1` plot, which only show one cycle, it can be observed that seasonality occured in a cycle of 7 days before restarting the cycle again.





From the ACF plot of X_t, it is observed that the ACF values almost seem to be monotonously decreasing. There are some lags here and there where this observation does not hold true, but in general, the ACF values from lags 1 through 20 seem to be decreasing. Initially, lags 1 through 6 are significant. The lags then become insignificant, but they return to being significant at lag 18, and continue to be significant until lag 20. Based on the ACF plot, we should not model X_t using an MA model.



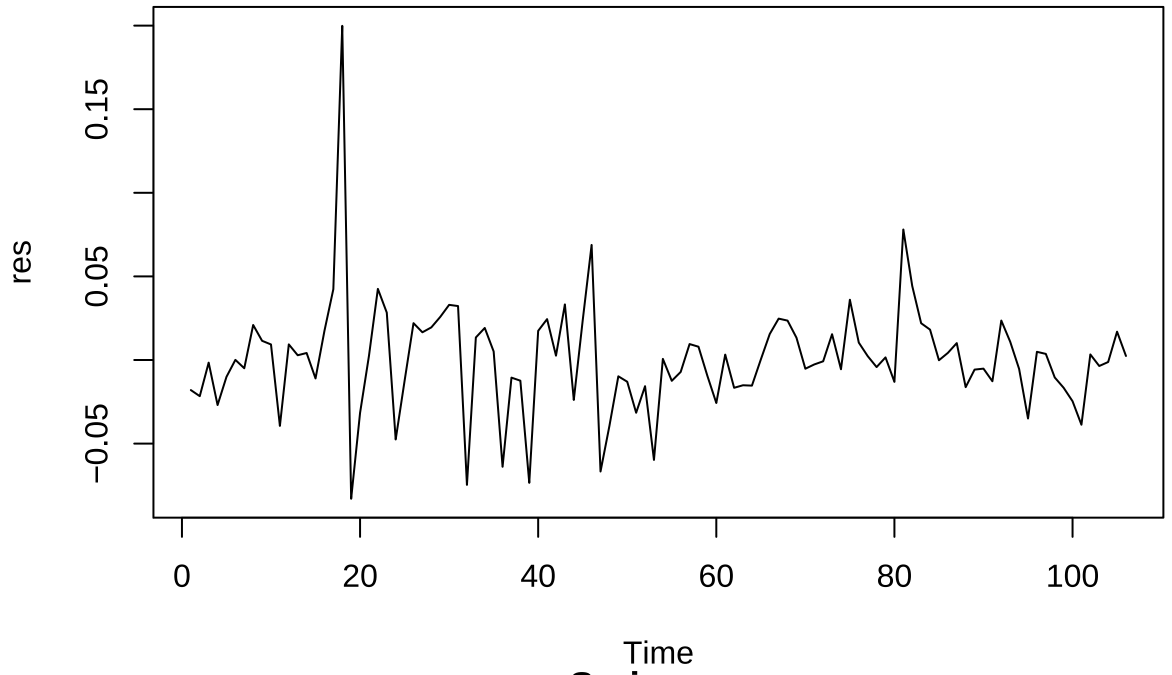
From the PACF plot of X_t, it is observed that there are not many significant lags. Lags 1, 14, and 16 are significant. If it is believed that lags 14 and 16 are not truly significant, but only appear significant due to randomness, then X_t could be modeled using an AR(1) model. However, if lags 14 and 16 are truly significant, then X_t cannot be modeled using an AR(1) model.

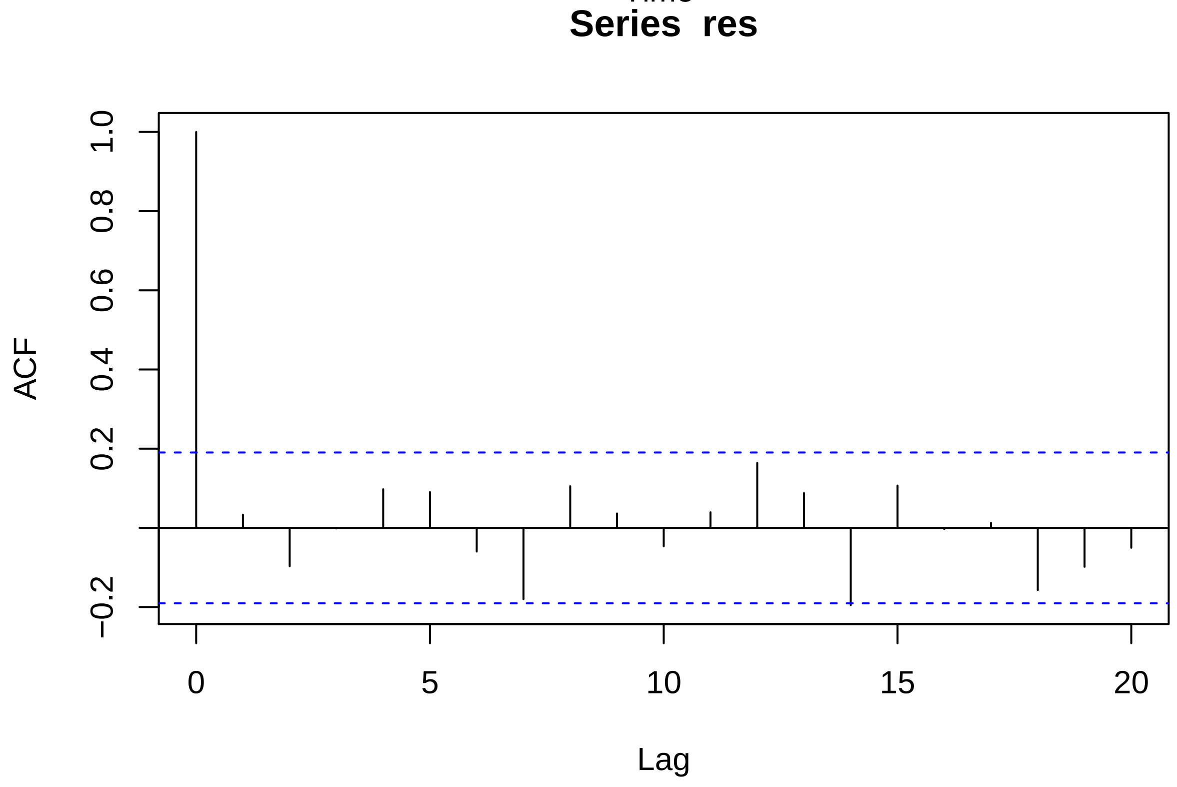
```
##
##  ARIMA(0,0,0) with zero mean     : -358.7914
##  ARIMA(0,0,0) with non-zero mean : -356.7133
##  ARIMA(0,0,1) with zero mean     : -393.0308
##  ARIMA(0,0,1) with non-zero mean : -390.9123
##  ARIMA(0,0,2) with zero mean     : -400.8769
##  ARIMA(0,0,2) with non-zero mean : -398.7171
##  ARIMA(0,0,3) with zero mean     : -402.2014
##  ARIMA(0,0,3) with non-zero mean : -399.9985
##  ARIMA(0,0,4) with zero mean     : -403.845
##  ARIMA(0,0,4) with non-zero mean : -401.5983
##  ARIMA(0,0,5) with zero mean     : -407.4243
##  ARIMA(0,0,5) with non-zero mean : -405.1307
##  ARIMA(1,0,0) with zero mean     : -412.1853
##  ARIMA(1,0,0) with non-zero mean : -410.0703
##  ARIMA(1,0,1) with zero mean     : -413.6869
##  ARIMA(1,0,1) with non-zero mean : -411.542
##  ARIMA(1,0,2) with zero mean     : -412.4599
##  ARIMA(1,0,2) with non-zero mean : -410.2827
##  ARIMA(1,0,3) with zero mean     : -410.4856
##  ARIMA(1,0,3) with non-zero mean : -408.2568
##  ARIMA(1,0,4) with zero mean     : -409.3232
##  ARIMA(1,0,4) with non-zero mean : -407.0339
##  ARIMA(1,0,5) with zero mean     : -408.1786
##  ARIMA(1,0,5) with non-zero mean : -405.8375
##  ARIMA(2,0,0) with zero mean     : -412.1838
##  ARIMA(2,0,0) with non-zero mean : -410.0301
##  ARIMA(2,0,1) with zero mean     : -412.19
##  ARIMA(2,0,1) with non-zero mean : -410.0114
##  ARIMA(2,0,2) with zero mean     : -410.3259
##  ARIMA(2,0,2) with non-zero mean : -408.1025
##  ARIMA(2,0,3) with zero mean     : -408.7075
##  ARIMA(2,0,3) with non-zero mean : -406.4171
##  ARIMA(2,0,4) with zero mean     : -406.8295
##  ARIMA(2,0,4) with non-zero mean : -404.4983
##  ARIMA(2,0,5) with zero mean     : -406.4006
##  ARIMA(2,0,5) with non-zero mean : -404.0102
##  ARIMA(3,0,0) with zero mean     : -412.2302
##  ARIMA(3,0,0) with non-zero mean : -410.0403
##  ARIMA(3,0,1) with zero mean     : -410.6531
##  ARIMA(3,0,1) with non-zero mean : -408.4275
##  ARIMA(3,0,2) with zero mean     : -411.1676
##  ARIMA(3,0,2) with non-zero mean : -408.8754
##  ARIMA(3,0,3) with zero mean     : Inf
##  ARIMA(3,0,3) with non-zero mean : Inf
##  ARIMA(3,0,4) with zero mean     : -404.8234
##  ARIMA(3,0,4) with non-zero mean : -402.4351
##  ARIMA(3,0,5) with zero mean     : Inf
##  ARIMA(3,0,5) with non-zero mean : Inf
##  ARIMA(4,0,0) with zero mean     : -411.112
##  ARIMA(4,0,0) with non-zero mean : -408.888
##  ARIMA(4,0,1) with zero mean     : -408.8636
##  ARIMA(4,0,1) with non-zero mean : -406.5937
##  ARIMA(4,0,2) with zero mean     : Inf
##  ARIMA(4,0,2) with non-zero mean : -406.6386
##  ARIMA(4,0,3) with zero mean     : -406.6467
##  ARIMA(4,0,3) with non-zero mean : Inf
##  ARIMA(4,0,4) with zero mean     : Inf
##  ARIMA(4,0,4) with non-zero mean : Inf
##  ARIMA(4,0,5) with zero mean     : Inf
##  ARIMA(4,0,5) with non-zero mean : Inf
##  ARIMA(5,0,0) with zero mean     : -408.8637
##  ARIMA(5,0,0) with non-zero mean : -406.5937
##  ARIMA(5,0,1) with zero mean     : -406.6076
##  ARIMA(5,0,1) with non-zero mean : -404.2922
##  ARIMA(5,0,2) with zero mean     : Inf
##  ARIMA(5,0,2) with non-zero mean : Inf
##  ARIMA(5,0,3) with zero mean     : -404.2716
##  ARIMA(5,0,3) with non-zero mean : -401.8343
##  ARIMA(5,0,4) with zero mean     : Inf
##  ARIMA(5,0,4) with non-zero mean : Inf
##  ARIMA(5,0,5) with zero mean     : Inf
##  ARIMA(5,0,5) with non-zero mean : Inf
##
##
##
##  Best model: ARIMA(1,0,1) with zero mean
```

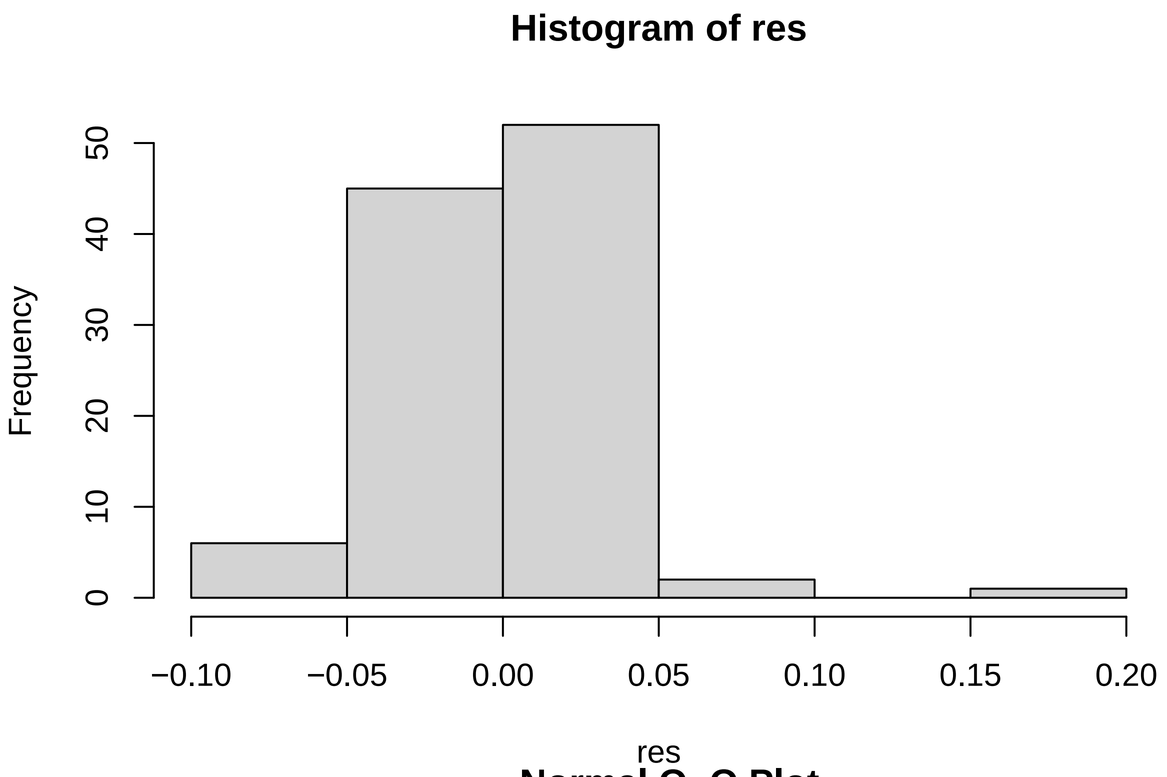
```
##         ar1         ma1   intercept
##  0.82564261 -0.34426853 -0.00149683
```

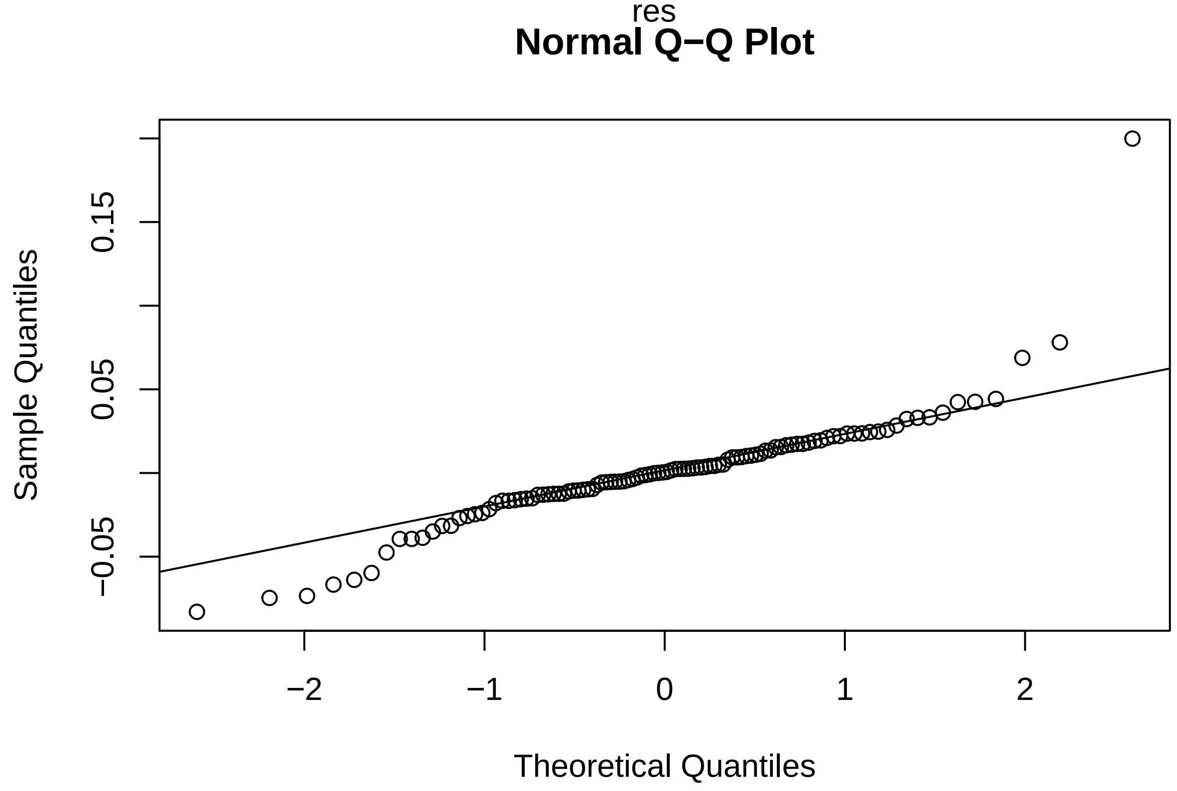
```
##                     ar1           ma1     intercept
## ar1        7.411166e-03 -0.0104390217 -8.253908e-05
## ma1       -1.043902e-02  0.0237533646  1.177348e-04
## intercept -8.253908e-05  0.0001177348  1.378879e-04
```

```
## [1] 0.001107958
```

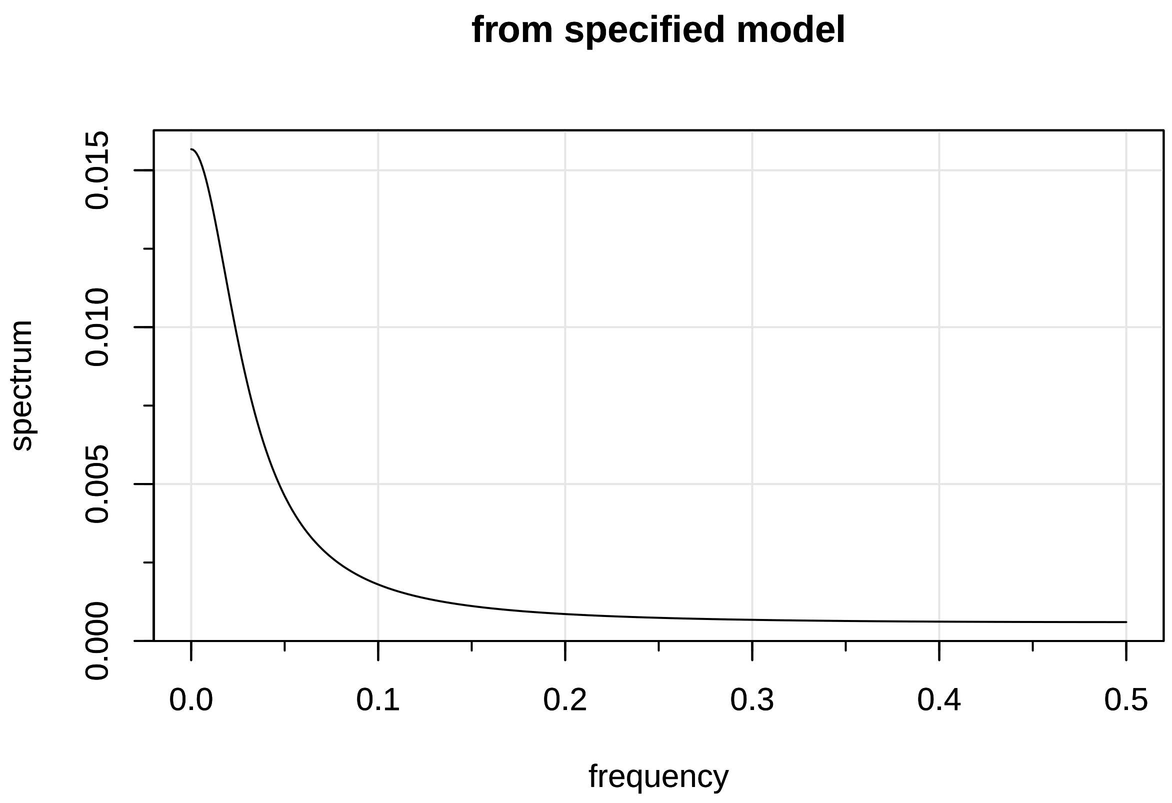


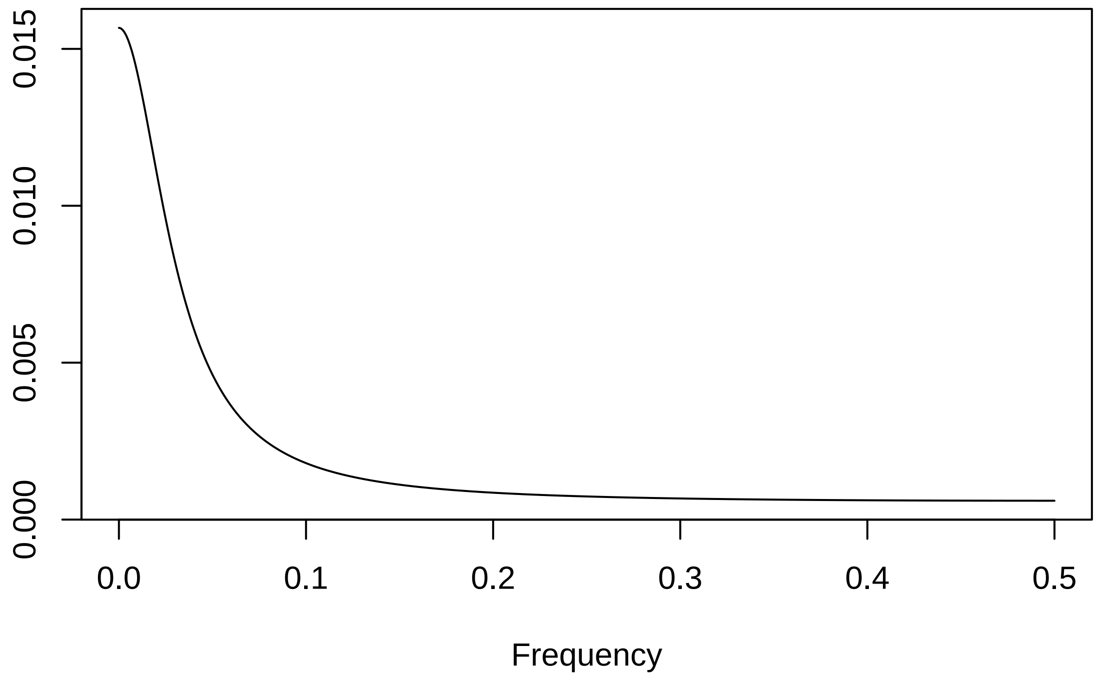






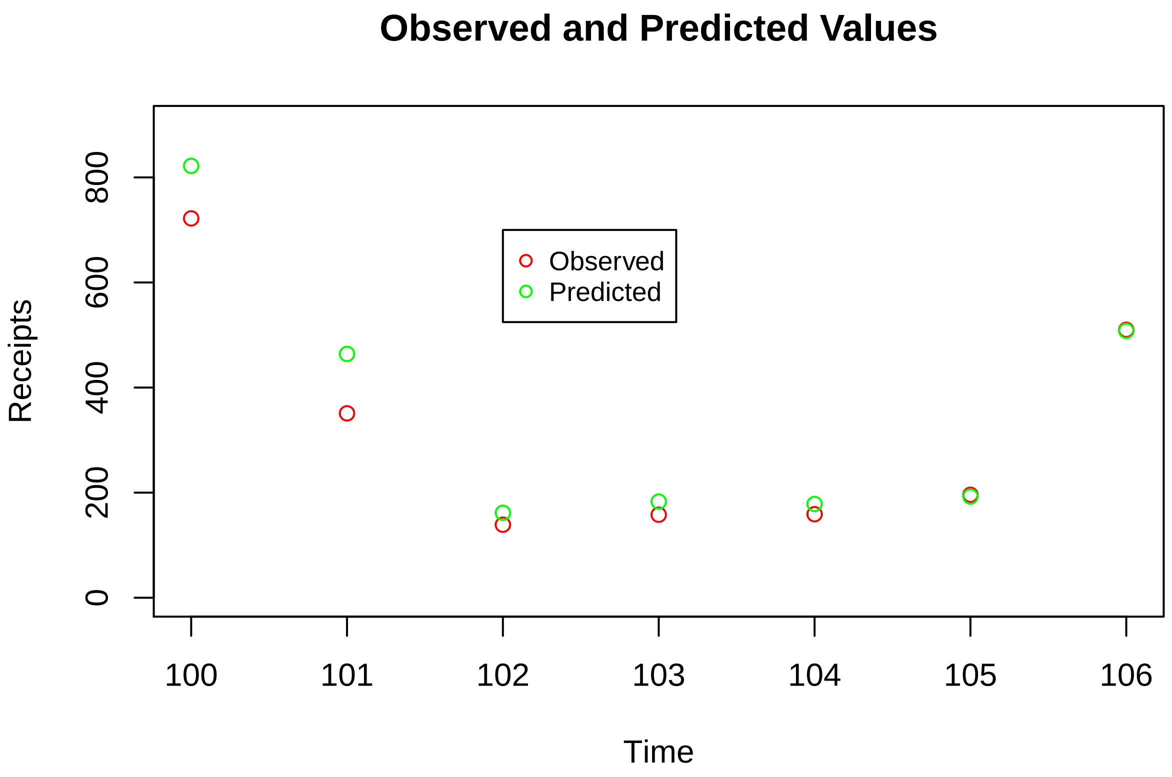
```
## $ctr
##  [1] 0.001597661 0.001607506 0.001570925 0.001428083 0.001390315 0.001447826
##  [7] 0.001438244 0.001460271 0.001500426 0.001510431 0.001509113 0.001531196
## [13] 0.001564247 0.001592351 0.001612854 0.001630694 0.001645669 0.001653710
##
## $kopt
## [1] 5

##
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':
##
##     gas
```





Spec
spectrum
0.015
0.010
0.005
0.000
0.0

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```



IV. Conclusion and Discussion

The objective of this report is to analyze daily average receipts per theater for the movie Chicago (Receipts) using time series methods. First, a Box-Cox transformation was applied to Receipts so that the variance of Receipts would be more constant overtime. Then, the trend m_t and the seasonal components s_t were estimated using the provided function trndseas(). After obtaining $\hat{m}_t$ and $\hat{s}_t$, the estimated rough $\hat{X}_t$ was obtained by subtracting $\hat{m}_t$ and $\hat{s}_t$ from transformed values of Receipts.

Next, using the AICC criterion andauto.arima(), an ARMA(1,1) model was selected to model X_t. Subsequent plots of the residuals of the ARMA(1,1) model show that there are no significant lags. This indicates that the ARMA(1,1) model is a good model for X_t.

Then, the spectral density function of the ARMA(1,1) model and a smoothed periodogram were plotted on the same graph. Using specselect(), a function provided in the discussion handout, it is observed that the most appropriate smoothed periodogram is when span = 11 (k = 5). At this span, the smoothed periodogram achieves a balance between smoothing the periodogram and preserving detail.

Finally, using all the data except for the last 7 days, the ARMA(1,1) model was refitted. This model was used to forecast the last 7 observations. While the predictions were not perfect, they came considerably close to the true observed values.

```
knitr::opts_chunk$set(echo = TRUE)
library(MASS)
source("https://raw.githubusercontent.com/Truc-T-Le/Time-Series-Project/main/trndseas.R")
chicago <- read.csv("https://raw.githubusercontent.com/Truc-T-Le/Time-Series-Project/main/chicago.csv")
library("forecast")
y <- chicago$reciepts
t <- 1:106
plot(t, y, type = "l", ylab = "Receipts", main = "Plot of Receipts against t")
trndseas=function(y,seas,lam,degtrnd){
```

```r
# requires the R-package 'pracma'

# fits  a trend plus seasonal for the "best" Box-Cox
# transformation.

# input: y, observed series; seas, seasons

# input: lam, the grid of Box-Cox transformations (lambda values)

# input: degtrnd, degree of the polynomial trend, if
# degtrnd=0, then the fitted trend is constant.

# output:  coef, regression coefficients - the
# first degtrnd+1 values for the trend part and the
# rest associated with the seasonals

# output: fit, fitted y-values; res, residuals,

# output: trend, fitted trend; season, fitted seasonals

# output: rsq, adjusted r-square values for different lambda in the

# output: lamopt, the value of lambda (among those supplied
# in the vector lam) at which r-square is maximum.

m=length(lam)
n=length(y)

# Part of design matrix for estimating trend
if(degtrnd>0) {
   tm=seq(1/n,1,by=1/n)
   x1=poly(tm,degree=degtrnd,raw=TRUE)
   x1=cbind(rep(1,n),x1)
   } else {
    x1=as.matrix(rep(1,n),ncol=1)
   }

# Part of design matrix for estimating seasonality
x2=NULL
if(seas>1){
sn=rep(1:seas,length.out=n)
x2=factor(sn,levels=unique(sn),ordered=TRUE)
x2=model.matrix(~x2-1)
m2=ncol(x2)
m21=m2-1
x2=x2[,1:m21]-matrix(rep(x2[,m2],m21),ncol=m21,nrow=nrow(x2),byrow=F)
}

x=cbind(x1,x2)  # design matrix

xx=t(x)%*%x
rsq=rep(1,m)
m1=ncol(x1)     #degtrnd+1
```

```
m11=m1+1
mx=ncol(x)      # degtrnd+1+seas-1

for(i in 1:m) {
  if (lam[i]==0) {
    yt=log(y)
  } else {
    yt=y^lam[i]
   }
  xy=t(x)%*%yt
  coef=solve(xx,xy)
  fit=x%*%coef
  res=yt-fit
  ssto=(n-1)*var(yt)
  sse=t(res)%*%res
  rsq[i]=1-((n-1)/(n-mx))*sse/ssto
  }

  ii=which.max(rsq)
  lamopt=lam[ii]
  if (lamopt==0) {
    yt=log(y)
  } else {
    yt=y^lamopt
   }
  xy=t(x)%*%yt
  coef=solve(xx,xy)
  fit=x%*%coef
  trnd=x1%*%coef[1:m1]
  season=NULL
  if(seas>1){
  season=c(coef[m11:mx],-sum(coef[m11:mx]))
  }
  res=yt-fit

  result=list(coef=coef,fitted=fit,trend=trnd,residual=res,season=season,rsq=rsq,lamopt=lamopt)
  return(result)
}

#3/4/5
y<-  chicago$reciepts
tm<- 1:106
seas <- 7
lam=seq(-1,1,by=0.05)
ff=trndseas(chicago$reciepts,seas,lam,3)
rsq=ff$rsq
ff=trndseas(chicago$reciepts,seas,0.1,3)
trend=ff$trend
season=ff$season
fit=ff$fit
Day=1:7
plot(lam,rsq,type="l",xlab="Lambda",ylab="R-sq",main="Average Ticket sales: R-square")
```

```r
#fitted
{plot.ts(y^0.1,ylab="Receipts^0.1",main='Plot: Transformed and Fitted')
points(tm,fit,type='l',lty=2, col="red")
legend(80,2.4, c("transformed","fit"), lty=c(1,1,2), col=c("black", "red"))}

par(mfrow=c(1,2))
#trend line
{plot.ts(y^0.1,ylab="",main='Estimated Trend')
points(tm,trend,type='l', col="purple")
legend(60,2.4, c("transformed","trend"), lty=c(1,1,2), col=c("black", "purple"))}

plot(Day,season,type='l',ylab='Seasonals (Receipts^0.1)',main='Estimated Seasonal')
y_0.1 <-(y)^0.1

plot(y_0.1-fit, type = "l", xlab = "Time", ylab = "Residuals", main = "Rough part")
#6
acf(y_0.1-fit)
pacf(y_0.1-fit)
#7
a<-auto.arima(y_0.1-fit, stepwise = F, approximation = F, trace= TRUE, ic = "aicc", max.order = 10)

mod_ARMA<- arima(y^0.1-fit, order =c(1,0,1))
mod_ARMA$coef
mod_ARMA$var.coef
mod_ARMA$sigma2
res <- mod_ARMA$residuals
ts.plot(res)

acf(res)
hist(res)

qqnorm(res);qqline(res)


#8

specselect=function(y,kmax)
  {
  ii=spec.pgram(y,log="no",plot=FALSE)
  ii=ii$spec
  cc=norm(as.matrix(ii),type="F")^2
  ctr=rep(1,kmax) ###criterion function
  for(k in 1:kmax)
    {
    ss=2*k+1; kk=1/(2*k)
    ff=spec.pgram(y,spans=ss,log="no",plot=FALSE)
    fspec=ff$spec
    ctr[k]=norm(as.matrix(ii-fspec),type="F")^2+kk*cc
  }
  kopt=which.min(ctr)
  result=list(ctr=ctr,kopt=kopt)
  return(result)
}
```

```
specselect(y_0.1-fit,18)


library(astsa)
coef.ar <- mod_ARMA$coef[1]
coef.ma <- mod_ARMA$coef[2]
sigma2 <- mod_ARMA$sigma2
mod_spec <-arma.spec(ar=coef.ar, ma=coef.ma, var.noise=sigma2, log='no')
plot(mod_spec$freq, mod_spec$spec, type='l', xlab='Frequency', ylab='')


#8.a
{arma.spec(ar=coef.ar, ma=coef.ma,var.noise = sigma2, log = "no",main="Spectral Density and Smooth Peri
smooth<-spec.pgram(y_0.1-fit, log = "no", spans = 11, main="", xlab="", ylab="", plot=F )
lines(smooth$freq, smooth$spec, col = "green")
legend(0.3, 0.018, legend = c("Spectral", "Smoothed"), col = c("black", "green"), lty = 1, cex = 0.8)}


#8.b

rough_no7 = y_0.1-fit
rough_no7 = rough_no7[-c(100,101,102,103,104,105,106)]
mod_ARMA11 = arima(rough_no7, order = c(1,0,1))

#forecast trend

library(Hmisc)
row_old = 1:99
row_pred = c(100,101,102,103,104,105,106)
trend_pred = approxExtrap(row_old, ff$trend, xout= row_pred)$y

season_pred = rep(ff$season, length.out = length(chicago$reciepts))[-(1:99)]

forecast = predict(mod_ARMA11, n.ahead = 7)
x_fc = forecast$pred

truepredicted = x_fc+trend_pred+season_pred
untransform_predict = truepredicted^10
{plot(c(100,101,102,103,104,105,106), chicago$reciepts[100:106], col = "red", main = "Observed and Predi
points(c(100,101,102,103,104,105,106),untransform_predict, col = "green")
legend(102, 700, legend = c("Observed", "Predicted"), col = c("red", "green"), pch = 1, cex = 0.8)}
# this is the code appendix
```