project

Haoyu Gao

12/10/2018

1. Introduction

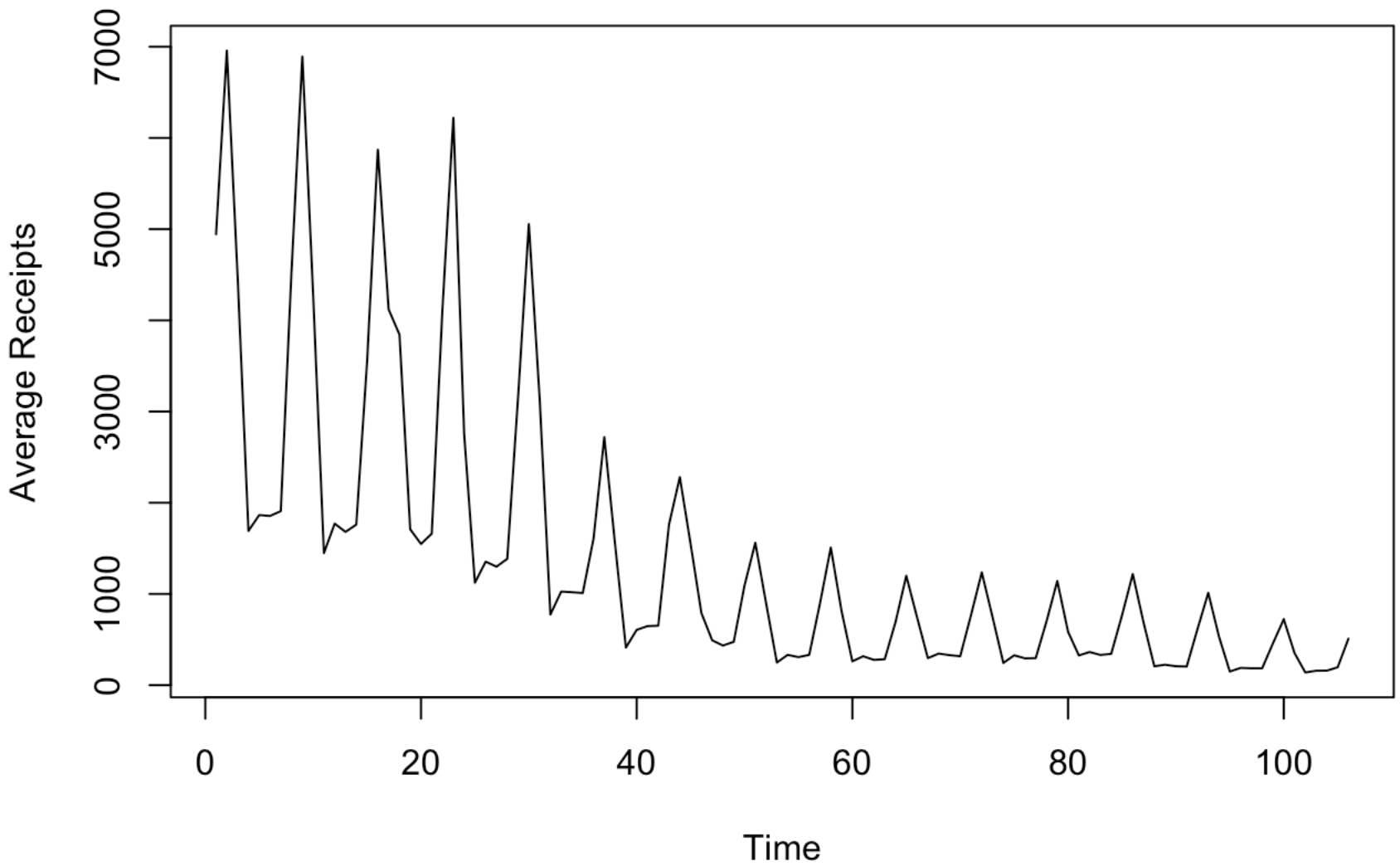
This is a time series analysis report. I will follow the steps provided in Project.docx.

The data contains the average receipts per theater for the movie Chicago from Jan 3, 2003 to April, 2003. Since we have data vary with time, time series analysis is appropriate. Through time series modeling, hopefully we can decompose the data into seasonal components, trend components, and the fluctuation than can be explained by the probability distribution. We will also look at the prediction performance by the time series model, and its spectral analysis to better understand the data.

2. Modelling Scheme

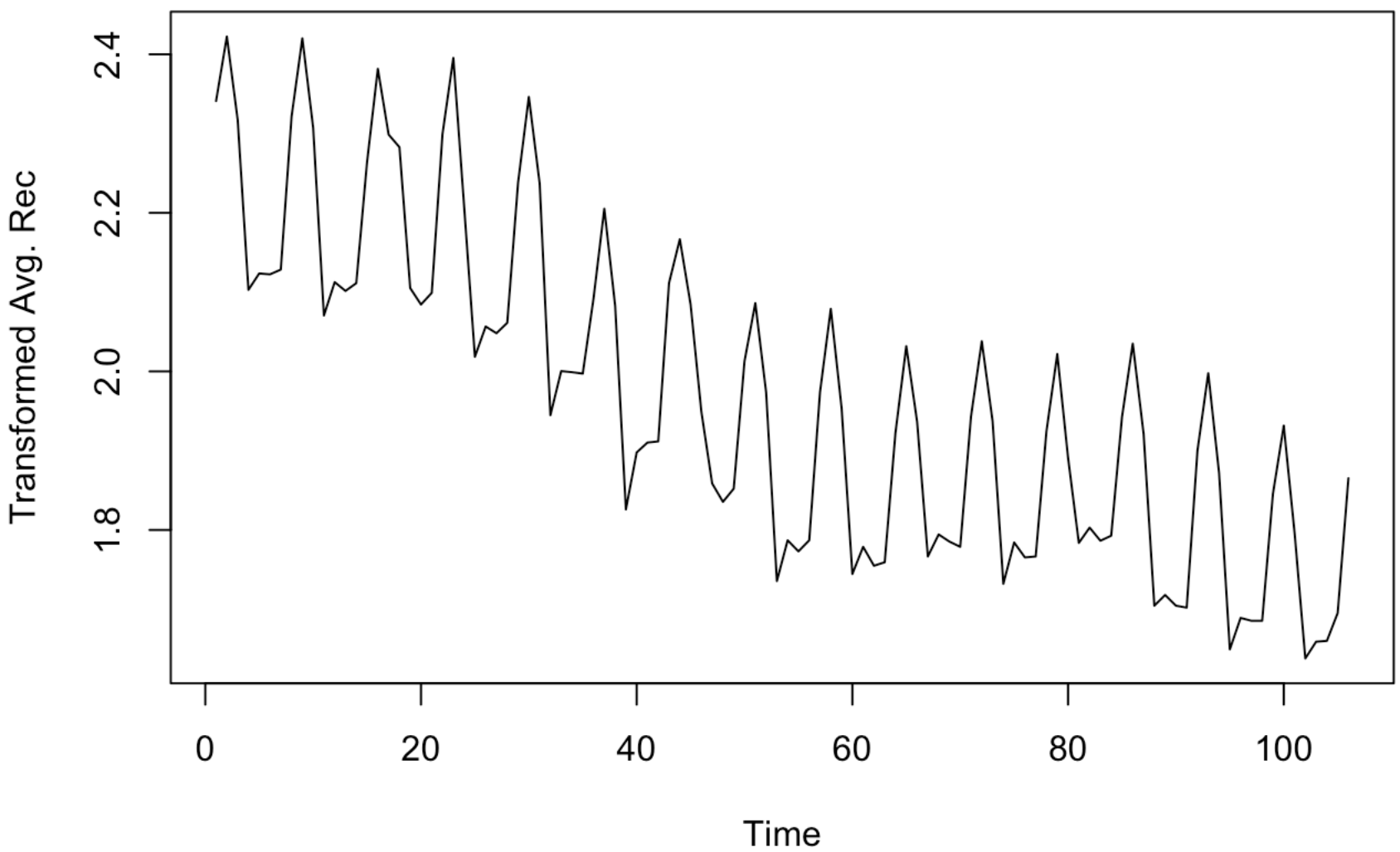
By the nature of the movie, one would expect higher receipts shortly after its release, and then downward trend in the long run. One would also need to take the seasonal effect into account, as it is likely that there will be more people going to the movie over the weekends. Therefore, in the time series model, one needs to think about the trend, the seasonal effect, and the rough part.

3. Data Transformation



Looking at the time series plot, we could spot an obvious downward trend. Also seasonality is there as well, a closer look reveals a peak every 7 days, and this is likely due to weekend traffic. Naturally, I will use s=7 in the following section.

The trndseas script returns a lambda of 0.1. Re-plotting the transformed data $y^{.1}$:

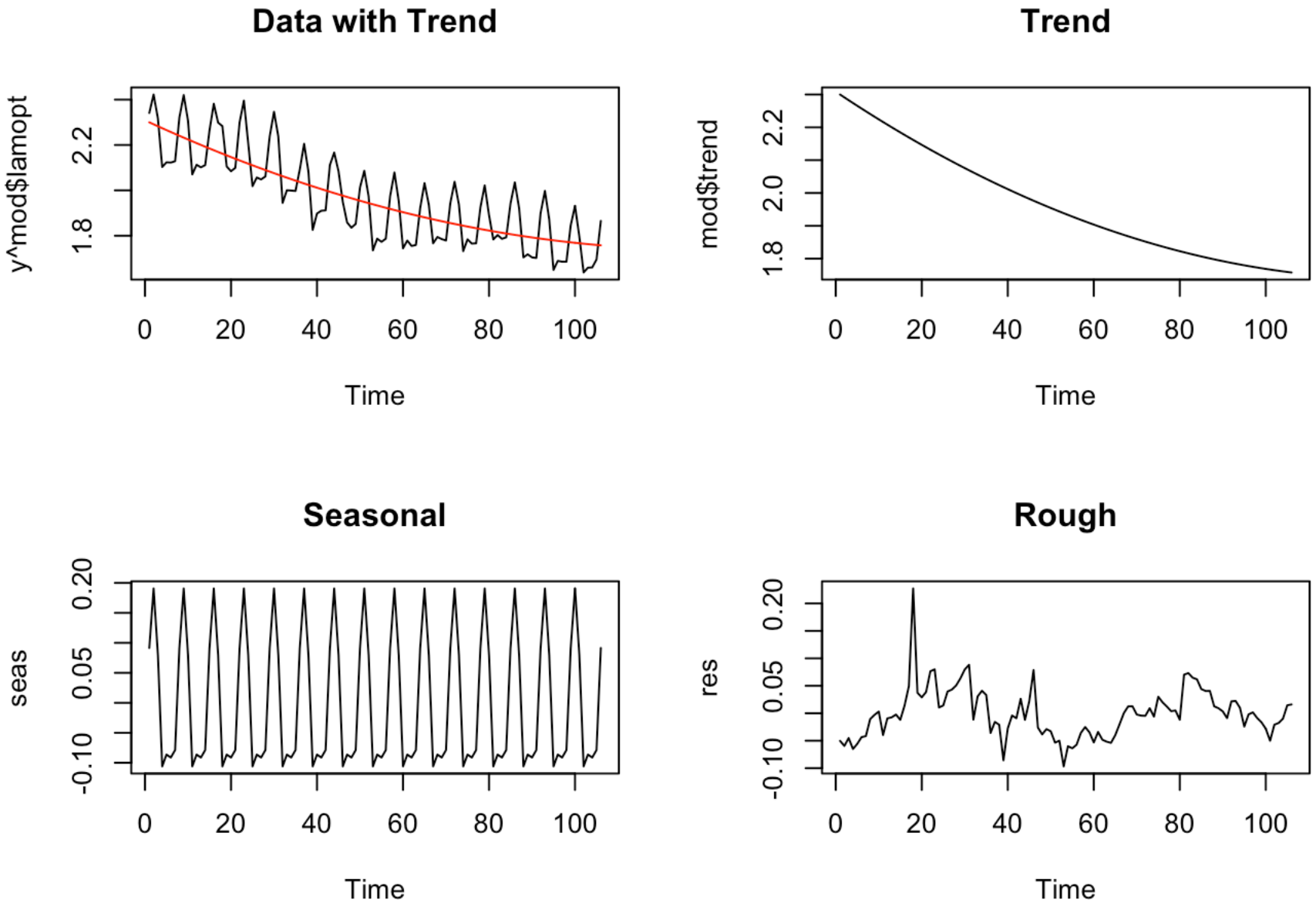


4. Trend & Seasonality

Here I used 2nd order polynomial, which I found is enough the explain the downward trend. Any higher order of polynomial means that model is subject to overfitting. And a seasonality of 7 is chosen as the data is sensitive to the day of the week.

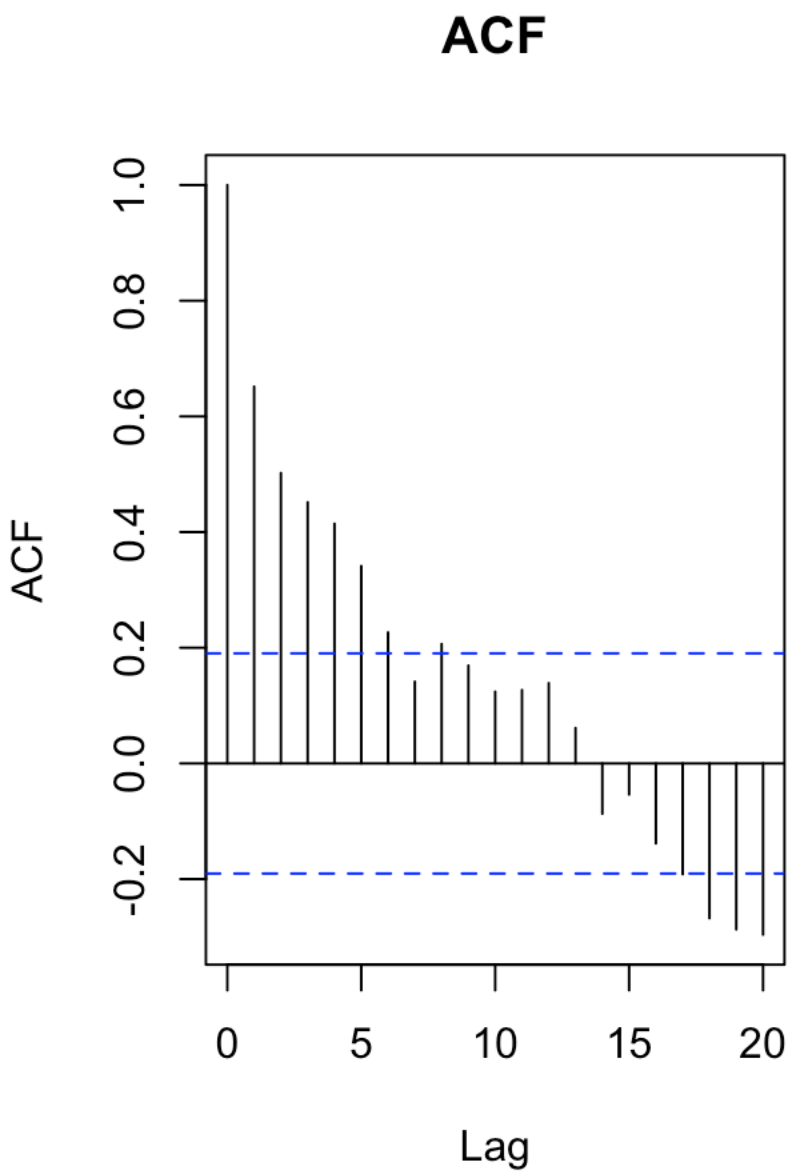
5. Components Plots

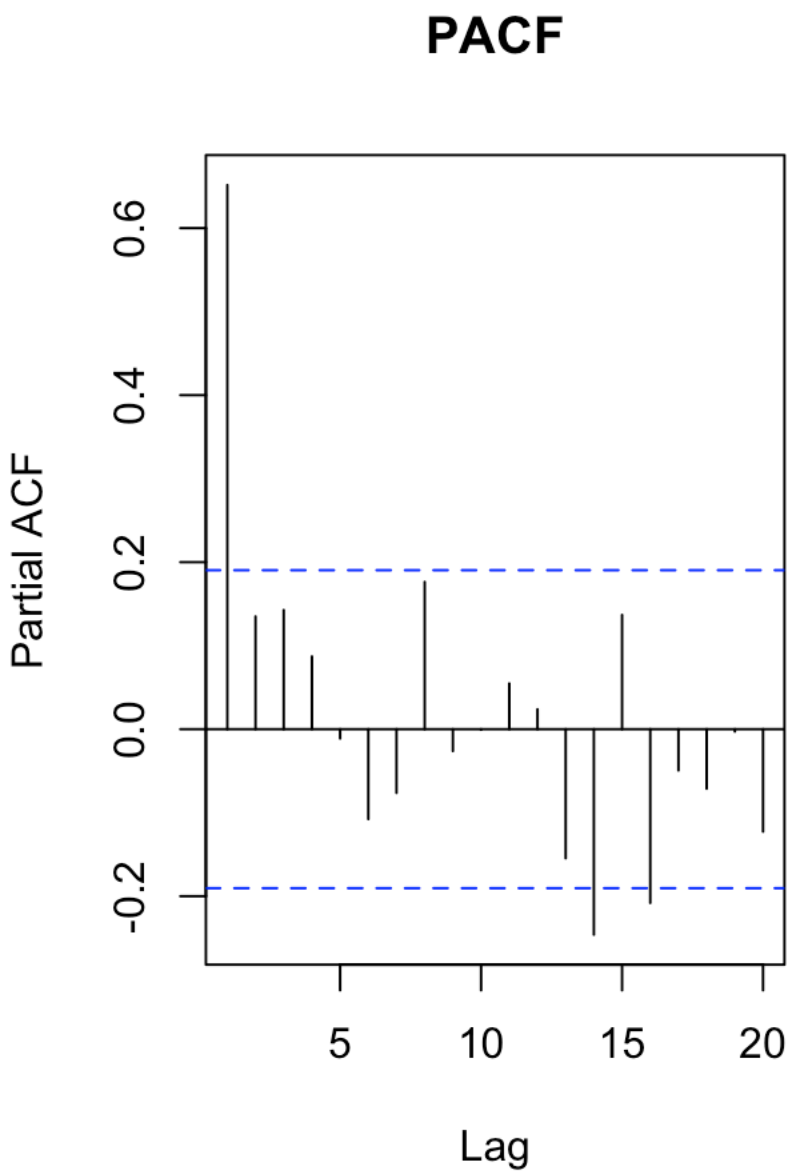
Here we can plot the trend, seasonal, and rough part seperately.

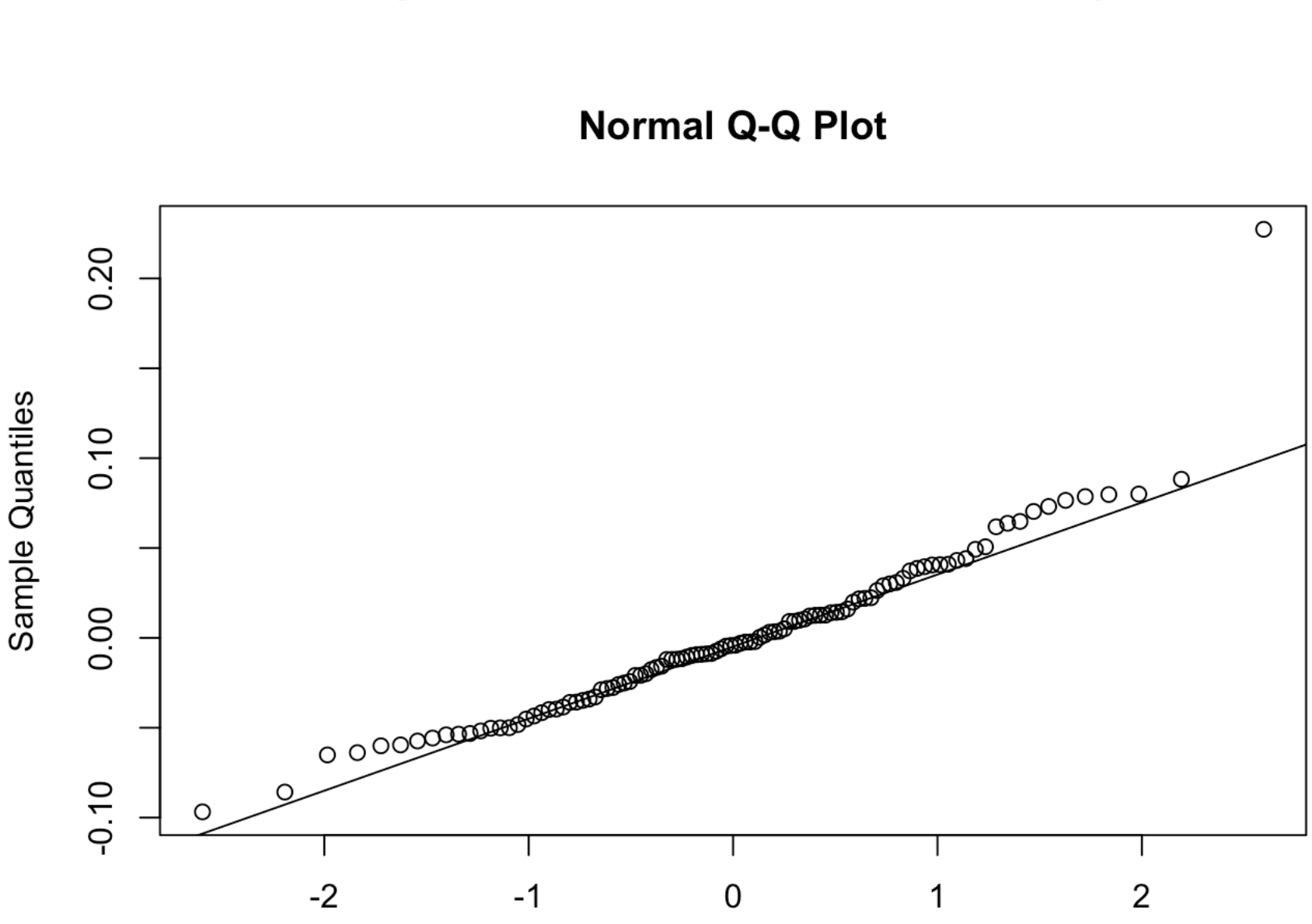


6. Diagnostics for the Rough Part

From the last part, we can see that the rough part appears to be random when I plot it against the time. The rough part has an ACF that tails off, indicating the possible MA terms. Similarly with PACF, there are possibily AR terms as well. Looking at Normal Q-Q plot, the series is close to normal.


ACF
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
20
Lag


PACF
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

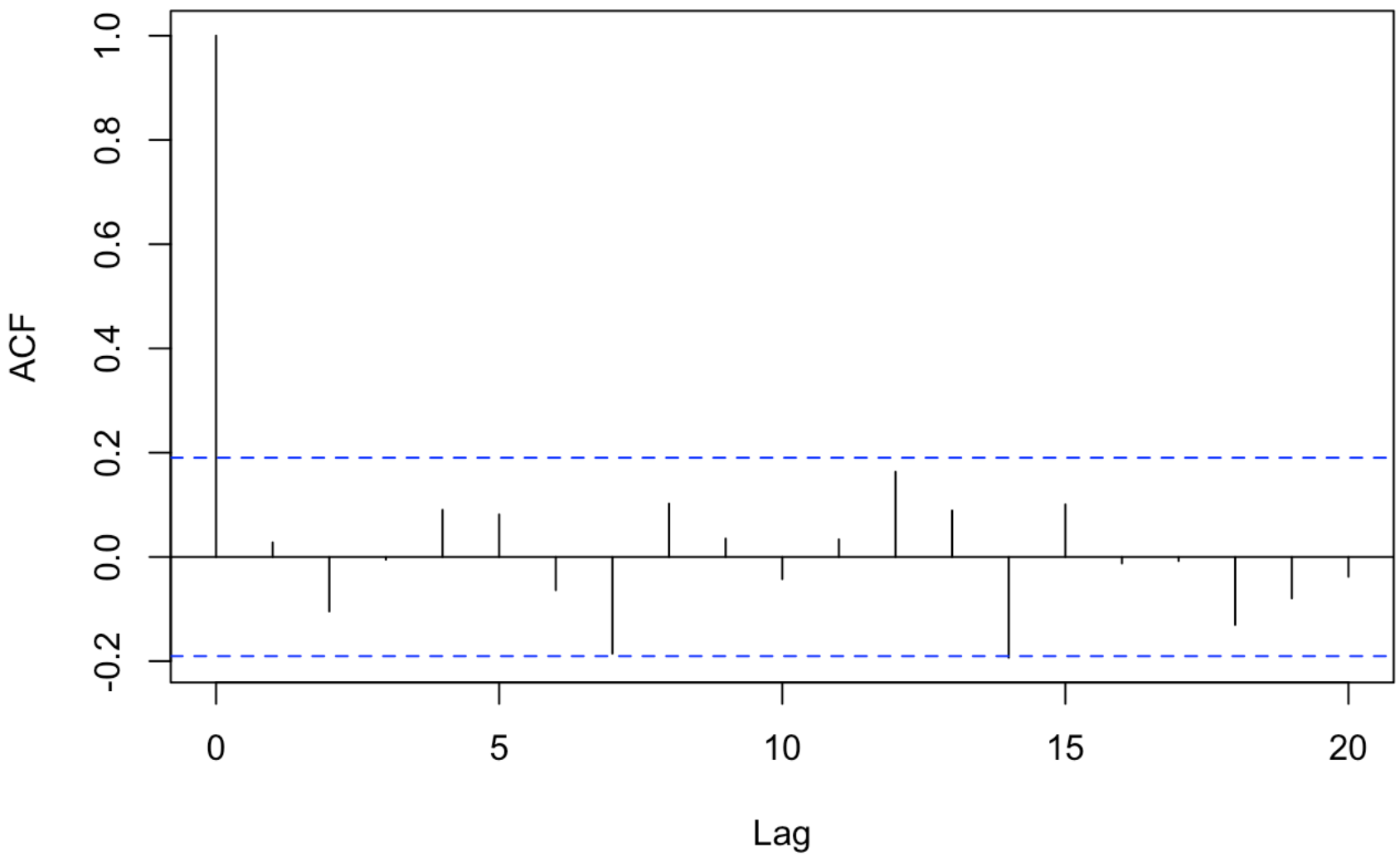
The p-value for Box-Ljung test is very small, and the assumption of independence is rejected.

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 147.59, df = 10, p-value < 2.2e-16
```

7. ARMA

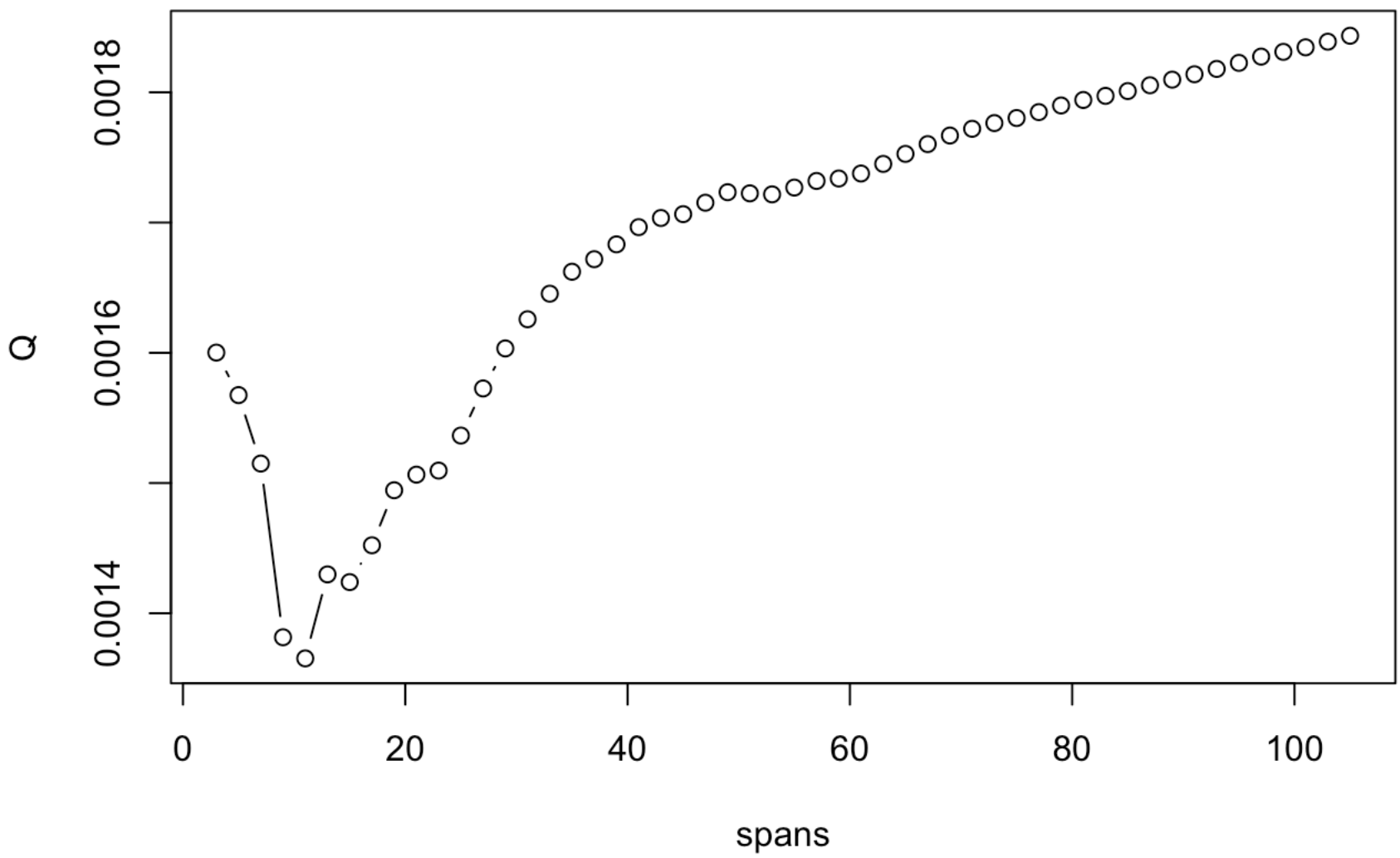
Using the AICC criterion, we obtained an ARMA(1,1) model. The ACF plot shows the residuals cut off after lag 1, which means the residuals are close to white noise.

```
## Series: res
## ARIMA(1,0,1) with zero mean
##
## Coefficients:
##          ar1      ma1
##       0.8356  -0.3401
## s.e.  0.0827   0.1507
##
## sigma^2 estimated as 0.001146:  log likelihood=209.15
## AIC=-412.29   AICc=-412.06   BIC=-404.3
```



8. Spectral density

I will use modified Daniell's kernel to select the span L to smooth the periodogram.

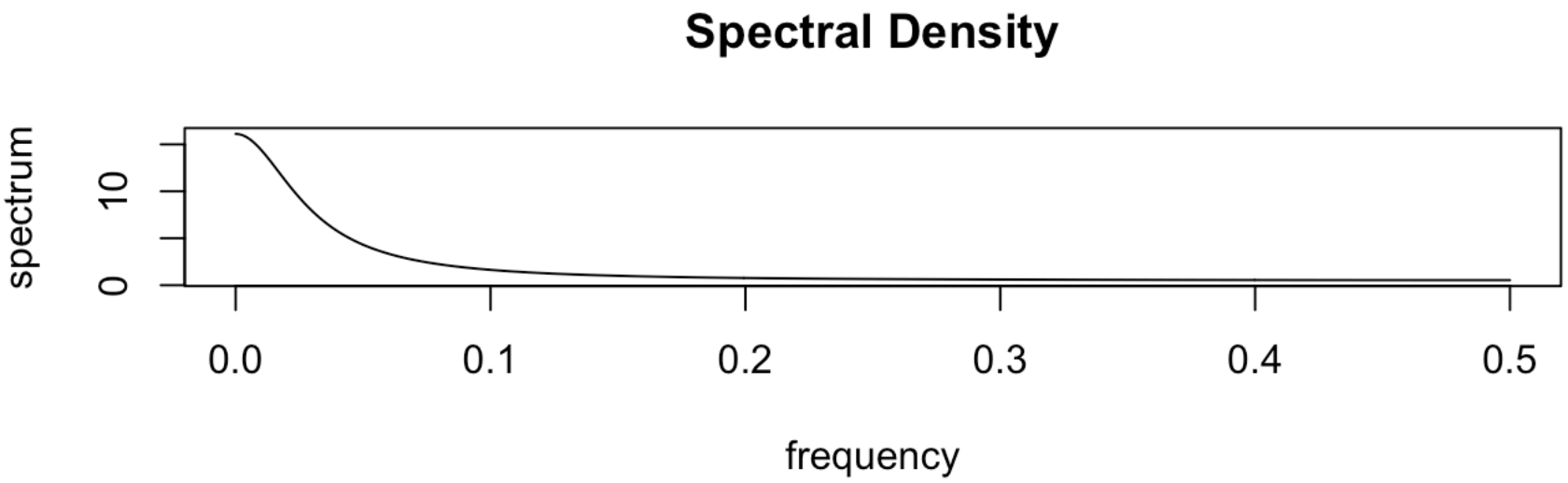


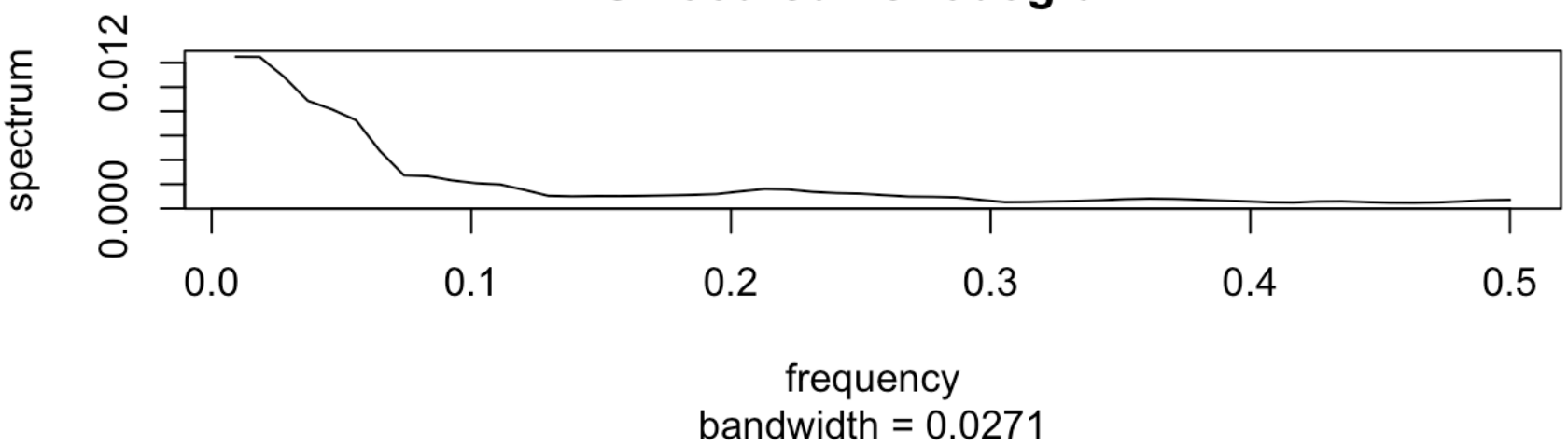

```
## [1] 11
```

Using span L=11 and plotting its smoothed periodogran with the spectral density function of ARMA(1,1)

```
##
## Attaching package: 'astsa'
```

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





9. Model Prediction

In this section, I refit the model using all the data except for the last 7 days, and use this model to forecast the last 7 observations. Firstly, recalulating the trend and seasonality, I predicted the last 7 values with linear extrapolation (for trend):

```
## 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
```

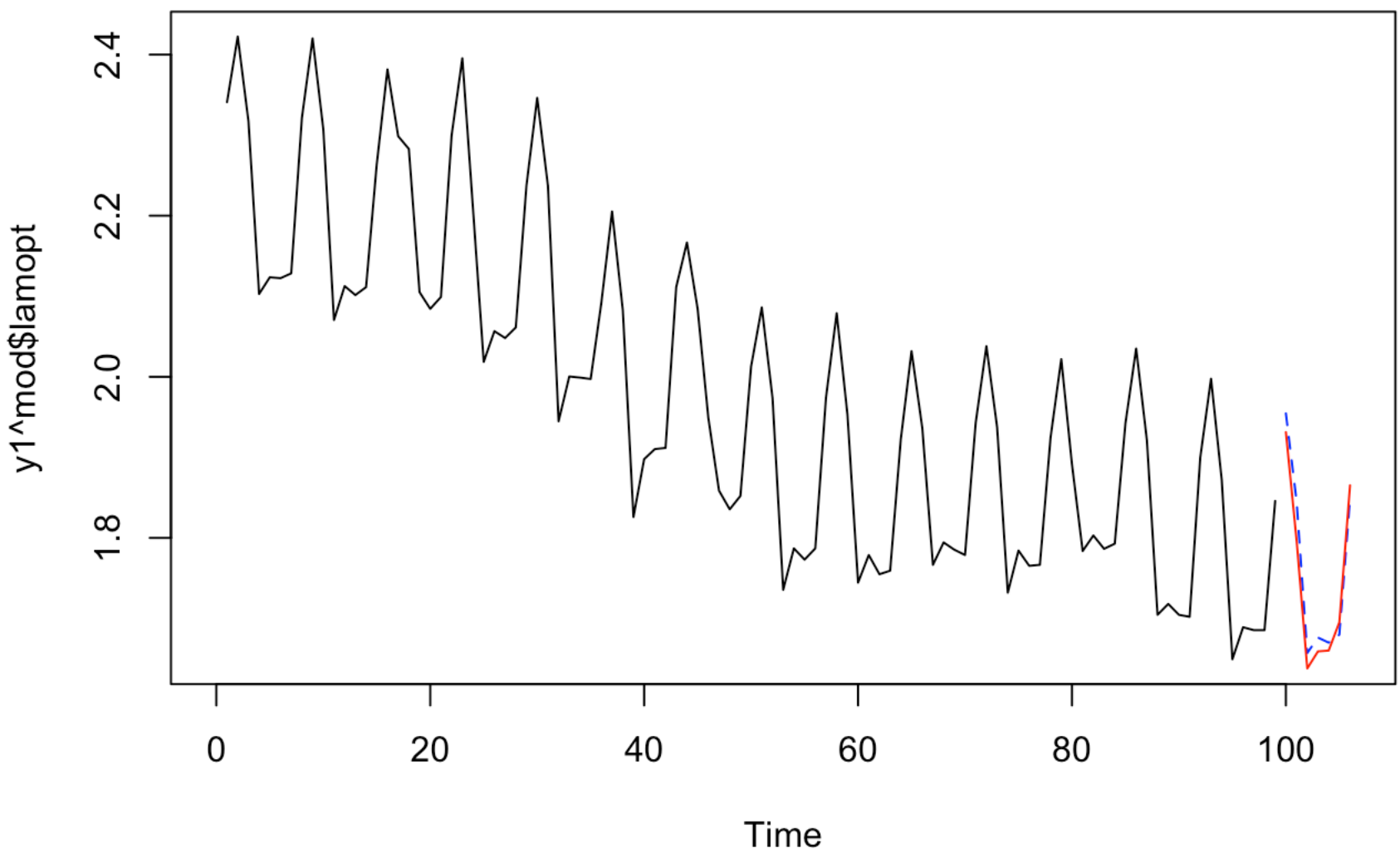
```
## [1] 1.778094 1.776337 1.774580 1.772823 1.771066 1.769308 1.767551
```

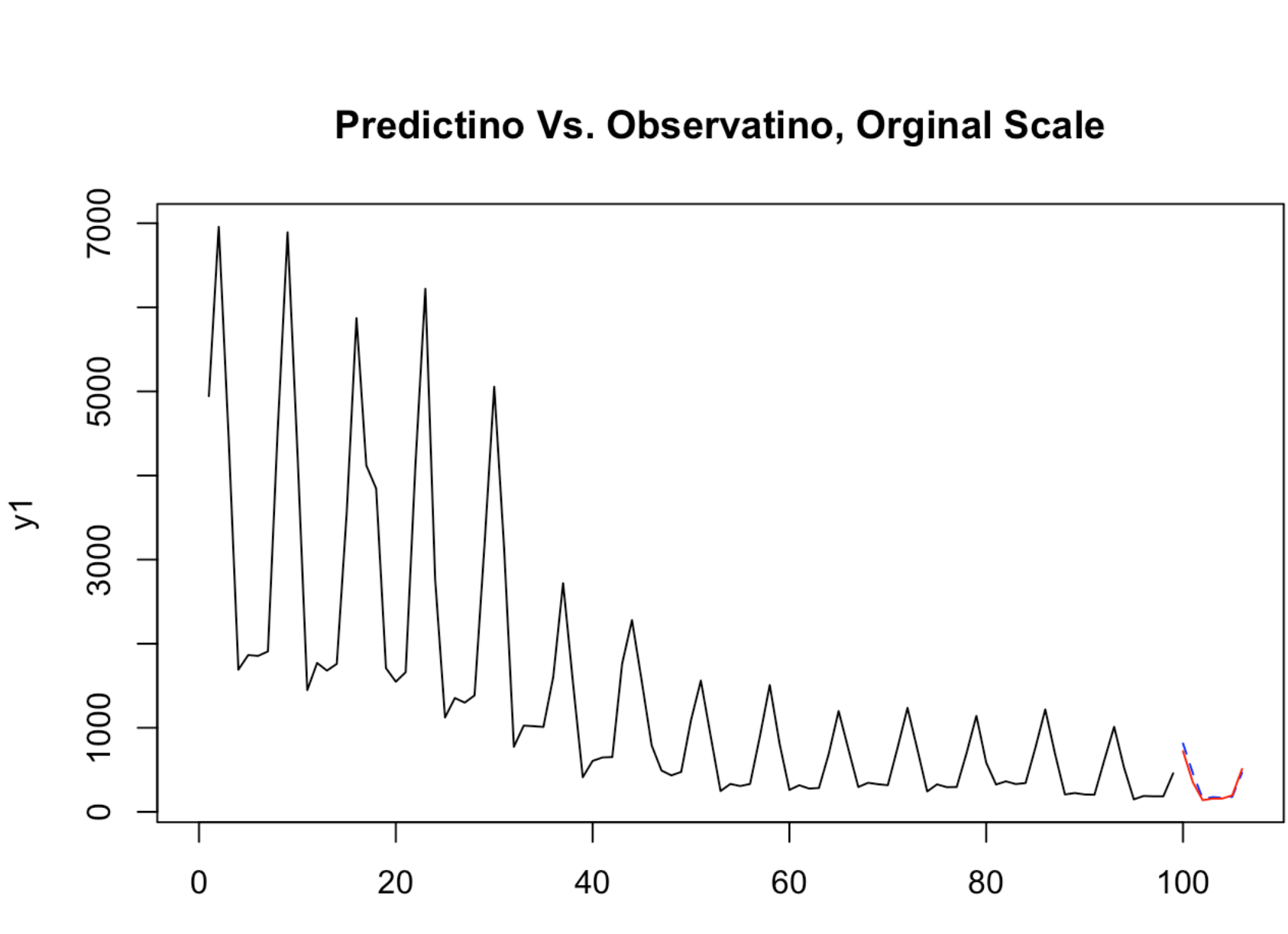
```
## [1]  0.19194823  0.08296700 -0.10545937 -0.08596265 -0.09162172 -0.08094546
## [7]  0.08907396
```

The predicted rough part:

```
## Time Series:
## Start = 100
## End = 106
## Frequency = 1
## [1] 0.03433678 0.03812458 0.04060763 0.04228741 0.04344429 0.04424985
## [7] 0.04481480
```

Finally, plotting the prediction for the last 7 days and the observation in the same graph:


Prediction Vs. Observation, Transformed
y1^mod$lamopt
1.8
2.0
2.2
2.4
0
20
40
60
80
100
Time


Predictino Vs. Observatino, Orginal Scale
y1
0
1000
3000
5000
7000
0
20
40
60
80
100

Time

Where the dotted line corresponds to the predicted value. Pretty close.

9. Summary

The final model is an ARMA(1,1) model. When I tested its performance on the last 7 days, the result is pretty close to the original data. Final verdict: very cool.

Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)
trndseas=function(y,seas,lam,degtrnd){

# 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)
}

chicago <- read.delim("~/Desktop/fall_2018/sta 137/project/chicago.txt", header=FALSE
)
y <- chicago$V1
n <- length(y)
lambdas <- seq(-3, 3, 0.1)
ts.plot(y, ylab='Average Receipts')
deg <- 2
s <- 7
mod <- trndseas(y, s, lambdas, deg)
ts.plot(y ^ mod$lamopt, main='Transformed', ylab='Transformed Avg. Rec')
par(mfrow = c(2,2))
ts.plot(y ^ mod$lamopt, main='Data with Trend')
lines(mod$trend, col='red')
ts.plot(mod$trend, main='Trend')
seas <- rep(mod$season, length.out = n)
ts.plot(seas, main='Seasonal')
res <- mod$residual
ts.plot(res, main='Rough')
par(mfrow = c(1,2))
acf(res, main='ACF')
pacf(res, main='PACF')
par(mfrow = c(1,1))
qqnorm(res)
qqline(res)
{Box.test(res, lag=10, type='Ljung-Box')}
library(forecast)
fit <- auto.arima(res, d=0, ic = c("aicc"))
fit
acf(fit$residual, main='ACF - residuals')
m <- floor(n/2)
spans <- (1:(m-1))*2+1
pgrm_raw <- spec.pgram(res, log="no", plot=F)$spec
Q <- numeric(length(spans))
for(j in 1:length(spans)){
  L <- spans[j]
  pgrm_smooth <- spec.pgram(res, spans=L,log="no", plot=F)$spec
  Q[j] <- sum((pgrm_smooth - pgrm_raw) ^ 2) + sum((pgrm_raw)^2)/(L-1)
}
plot(x=spans, y=Q, type='b')
L <- spans[which.min(Q)]; L
```

```r
library(astsa)
par(mfrow=c(2,1))
arma.spec(ar=c(0.8356),ma=c(-0.3401),log='no', main='Spectral Density')
spec.pgram(res, spans=L, log="no")
y1 = y[1:99]
y_obs = y[100:106]
n <- length(y1)
deg <- 2
s <- 7
mod <- trndseas(y1, s, lambdas, deg)
h <- 7
library(Hmisc)
ind_old <- 1:n
ind_new <- n + 1:h
trend=approxExtrap(ind_old, mod$trend, xout=ind_new)$y
season=rep(mod$season, length.out = n+h)[-(1:n)]

fit = arima(mod$residual, order=c(1,0,1))
phi <- fit$coef[1]
theta <- fit$coef[2]
psi <- ARMAtoMA(ar=phi, ma=theta, lag.max=40)
h <- 7
sigma2 <- mod$sigma2
trend
season
predict(fit, n.ahead=h)$se

fcast <- predict(fit, n.ahead=h)
x_fc <- fcast$pred + trend + season
plot.ts(y1 ^ mod$lamopt, xlim=c(0,n+h), main='Prediction Vs. Observation, Transformed
')
lines(x=n+(1:h), y=x_fc,col='blue',lty=2)
lines(x=n+(1:h), y=y_obs ^ mod$lamopt,col='red')

plot.ts(y1, xlim=c(0,n+h), main='Predictino Vs. Observatino, Orginal Scale')
lines(x=n+(1:h), y=x_fc^10,col='blue',lty=2)
lines(x=n+(1:h), y=y_obs,col='red')
```