STA 137 Final Project

Team Member

- Chen Zhang 997767317 chnzh@ucdavis.edu
- Rong Duan 917822313 rduan@ucdavis.edu
- Xialin Sang 917780316 xsang@ucdavis.edu

1 Introduction

This project examines the movie sales in the Chicago area. The data used is the average number of receipts per day per theater collected over January 3, 2003 to April 23, 2003. The data provides interesting insights into the general trend and seasonal trend of movie sales over the four months period. A range of time series analysis is conducted to uncover the underlying patterns of movie sales. First, the data is transferred in considering Box-Cox transformation, then general trend through a polynomial model, next, the stationary part is modeled with ARMA analysis and characterized with spectral analysis. Lastly, a forecast is made by reassembling trend, seasonal trend and the rough stationary components. We believe a better understanding and careful forecasting of movies sales will be beneficial to stakeholders.

2 Materials and Methods

2.1 Data transformation and model build

From the raw daily average receipts shown in Figure 1, it seems the data fluctuate very severely, in which case the huge estimate variance may influence the estimated effect. Hence, consider transformation with the Box-cox method. The best transformation parameter is selected by minimizing $R^2(\lambda)$. The trend part is estimated by polynomials, and the degrees of the polynomial are selected by comparing with the loess fit of the transformed data. The span of the loess smoothing method is carefully selected by subjective choice. The method starts at a small span (under-smoothing) and increases 0.01 at each time, and stops just when over-smoothing shows. Also, the severe data fluctuation may because of daily (seasonal) effects, where weekends tend to sell more tickets. With these in mind, we could construct a time series model as below:

$$Y_t = m_t + s_t + X_t, \qquad t = 1, 2, ..., 106$$

where Y_t is the transformed number of tickets at eachi time, m_t is the trend (smooth part), s_t is the seasonal component, and X_t is the rough part.

2.2 Rough part analysis

The rough part is the remaining part when subtracting estimated trends and seasonal from the transformed data. With the formula: $\hat{X}_t = Y_t - \hat{m}_t + \hat{s}_t$. For the rough part, an ARMA model is selected with the smallest AICC value, and model diagnostic is conducted by checking whether residuals of the ARMA model are white noise. The spectral density function f_j and the smoothed periodogram $\hat{f}_j$ of the final model are compared in the same plot, where j is the proportion of frequency w with the formula $w = \frac{j}{n}$. Specifically, the smoothed periodogram $\hat{f}_j$ is obtained by

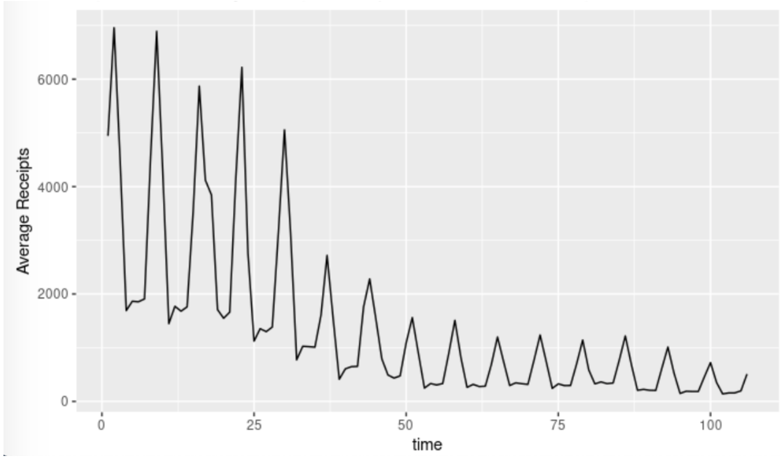

Figure 1: Raw Daily Average Receipts for the Movie Chicago

modified Daniell's method, and the number of neighbors is selected by minimizing the quantity $Q(k) = \sum(I_j - \hat{f}_j)^2 + \frac{1}{2k}\sum I_j^2$, where I_j is the periodogram. In the end, the estimated ability of the model is tested. The last seven days' observation is subtracted as new data, and the model is refitted. The trend and seasonal components are re-estimated, and the rough part of the last seven days is estimated by the ARMA model selected above. The estimating result is displayed by prediction values, prediction intervals, and true values in the same plot.

3 Results

3.1 Series Transformation and Smoothing Fit

Since the raw data is very rough: Figure [1], the Box cox transformation is conducted with potential degrees of polynomials (degree=2, 3, 4, 5). All result shows the best transformation parameter is $\lambda = 0.1$, which minimized R^2. The loess fit of the transformed data Y^λ is compared with the different degrees of polynomials. The selected span for loess is 0.3. The comparison of different degrees of polynomials and loess fit is in Figure [2]. It shows the loess has 3 bounces, and it matches well with the 4 degrees of the polynomial. Thus 4 degrees of the polynomial is selected to fit the trend, and the fitted result is in Figure [3]. It indicates the fitted value (trend + seasonal) fits the transformed data quite well.

3.2 Model Building of the Rough Part

The trend with 4 degrees of the polynomial is in Figure [2] left down part. The seasonal fit result is in table [1]. We can see that Friday, Saturday, and Sunday are positively related to the number of ticket sales, while Monday to Thursday is negatively related to ticket sales. The estimated seasonal effect is plotted in Figure [4].

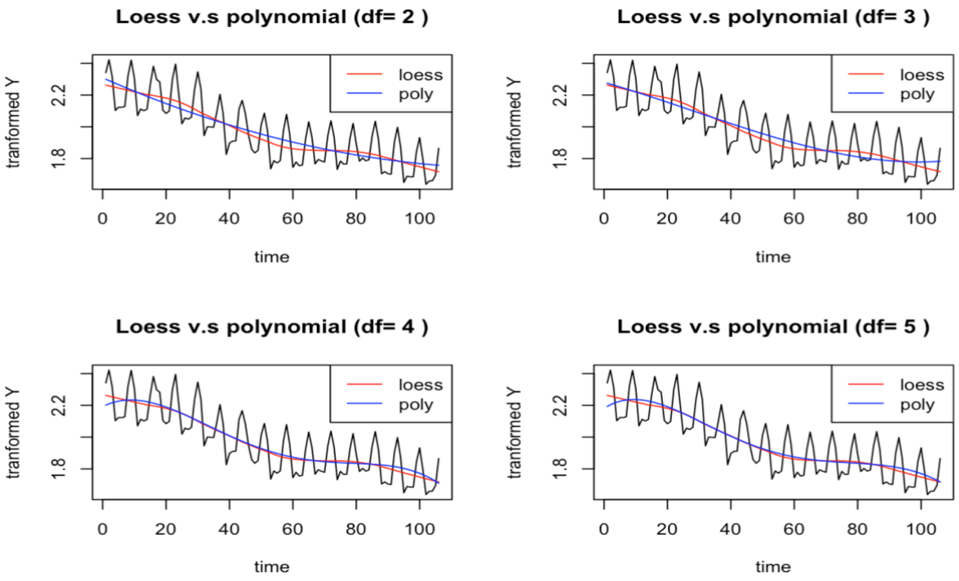

Figure 2: Loess and polynomial smoothing comparison

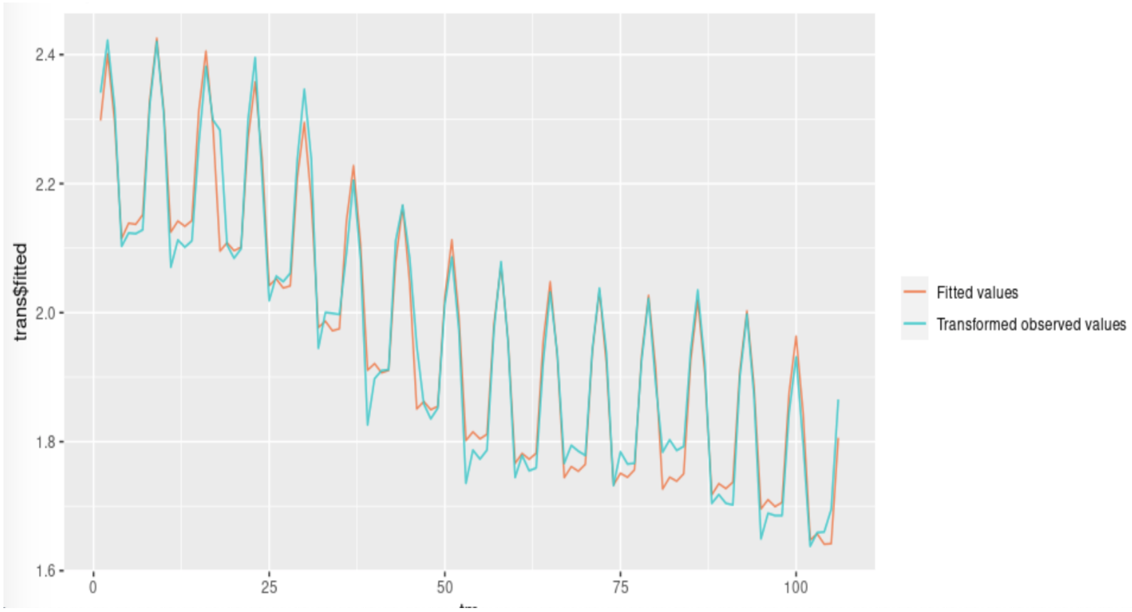

Figure 3: Fitted v.s. Transformed observed value

	Fri.	Sat.	Sun.	Mon.	Tues.	Wed.	Thur.
Estimated parameter	0.096	0.191	0.080	-0.107	-0.088	-0.093	-0.080

Table 1: Seasonal components: estimated parameter

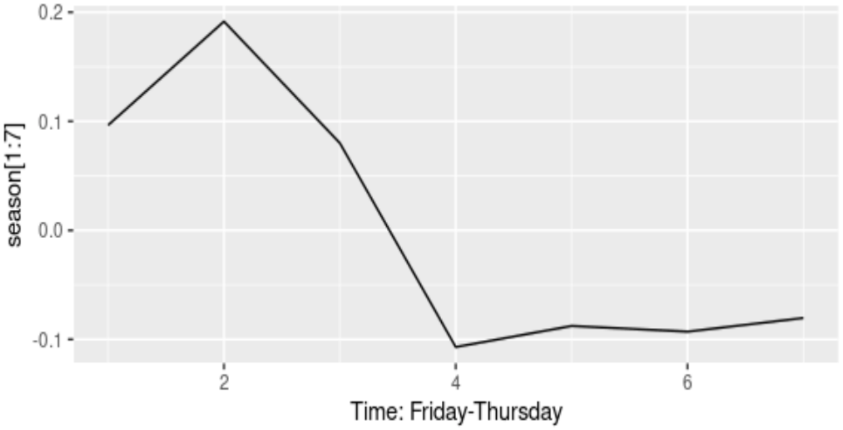

Figure 4: Seasonal components plot

The rough part is plotted in Figure [5], it seems the rough part fluctuated around 0, and no trends show. The variance of the rough part is likely to be constant for the whole time span. In Figure [6], from the histogram plot and QQ-plot, it shows a little right tail in the plots, thus right skew exists in this case. Shapiro-Wilk's method also confirms the observation. With significant test results indicating that the distribution of the rough part is different from a normal distribution. For the test of stationarity, we have conducted the augmented Dickey-Fuller test (ADF) and get a significant result. This result indicates the rough part is stationary. In the autocorrelation plot (ACF), lags 1, 6, and 7 are significant and tails off shows. In the partial autocorrelation plot (PACF), lag 1, 15, and 18 is significant. The rough part is not independent since lag 1 is significant in the ACF plot, and the result from the Ljung-Box test also shows the rough part is not independent with a significant p-value.

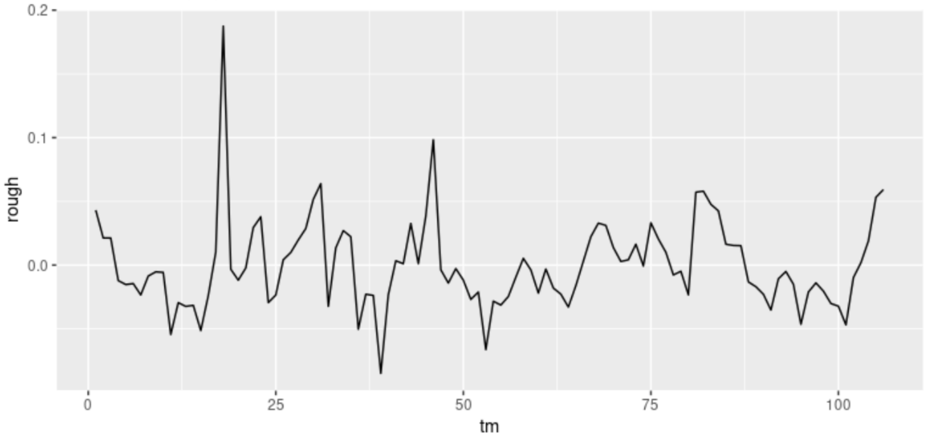

Figure 5: Rough part

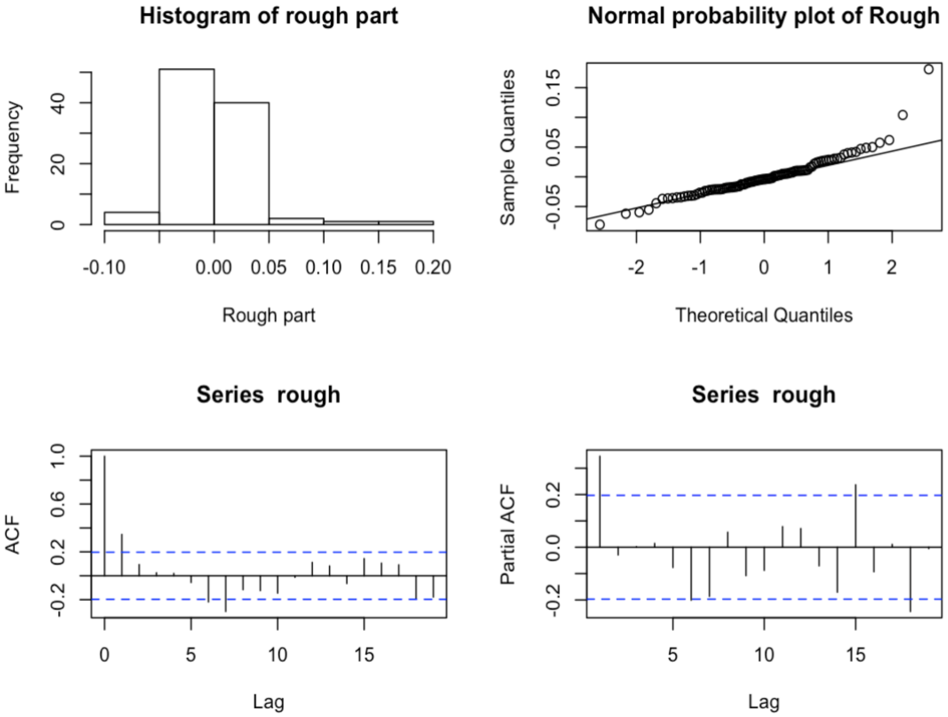

Figure 6: Rough part: Normality, ACF and PACF plot

Consider fitting the rough part by the ARMA model. The selected model is AR(1), which has the smallest AICC value, and also the number of parameters is reasonable. The estimated parameter $phi = 0.4223$, with 95% CI [0.24688, 0.59772], which indicate the parameter is indeed significant. The selected model is diagnostic by the residual ACF plot. In Figure [7], the ACF plot shows the residual is the white noise, thus the model is reasonable for this data. In addition, the estimated parameter is between -1 and 1, which is an indication that the rough part is stationary.

3.3 Estimation of the Spectral Density Function

The spectral density function is estimated by smoothed periodogram. The neighbor of smoothing periodogram has obtained the criteria mentioned in section **2.2**, and it shows 7 neighbors is the most satisfactory choice (Figure [8]). The spectral density function and the smoothing periodogram is plotted in Figure [8], it shows when w around 0.15 the smoothed periodogram is lower than the spectral density function value, and it is higher than the spectral density function value when w is between 0.2 and 0.3. Overall, the smoothing periodogram has a similar trend as the spectral function, thus it estimates the spectral function quite well in this case.

3.4 Forecast Ability

The first 99 cases are re-fitted to obtained the smoothing, seasonal, and the rough part, and the last 7 days is predicted with the established model. The predicted value is obtained from 3 parts:

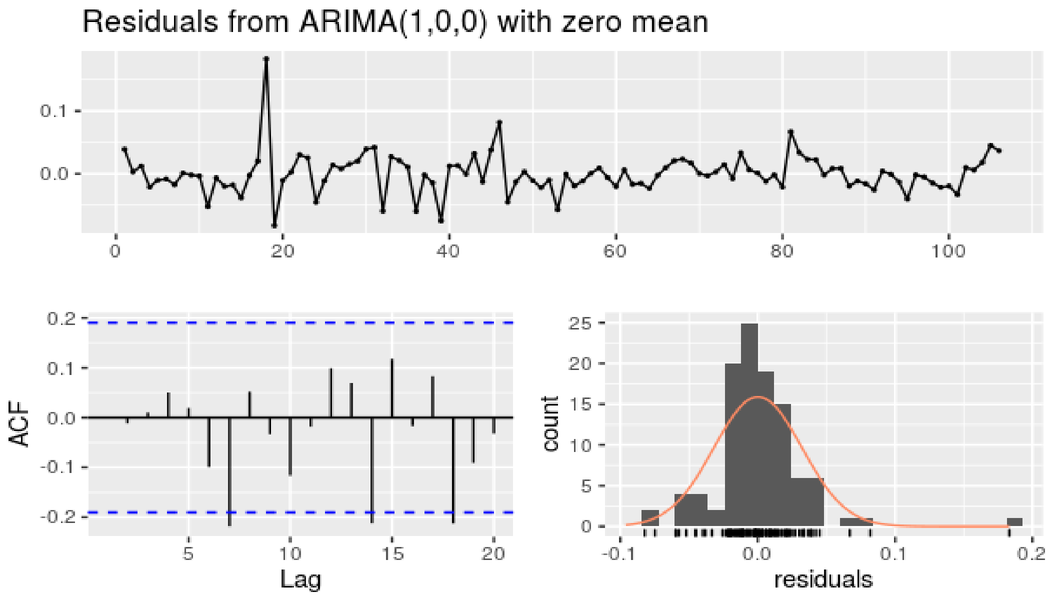

Figure 7: Residual ACF plot

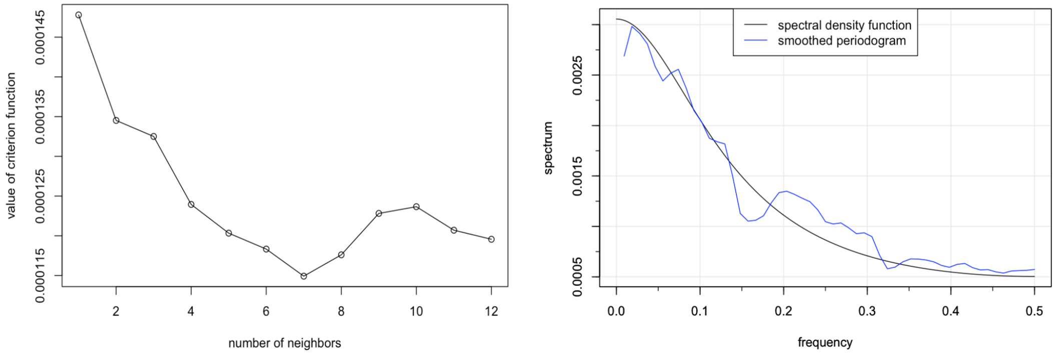

Figure 8: Smoothing neighbor choice and Spectral density estimation

trend, seasonal and rough part. The trend part is obtained from the predicted value of the 4-degree polynomial, and the seasonal part is obtained from the estimated seasonal parameter. The estimated rough part is obtained from the AR(1) model with new parameters. The estimated number of tickets and observed value is in Figure [9]. Both plots show the predicted value for day 100 and day 101 is pretty good. And the estimated value of day 102 to day 106 has a similar value although a little biased. Overall, the performance of the time series analysis estimation is good.

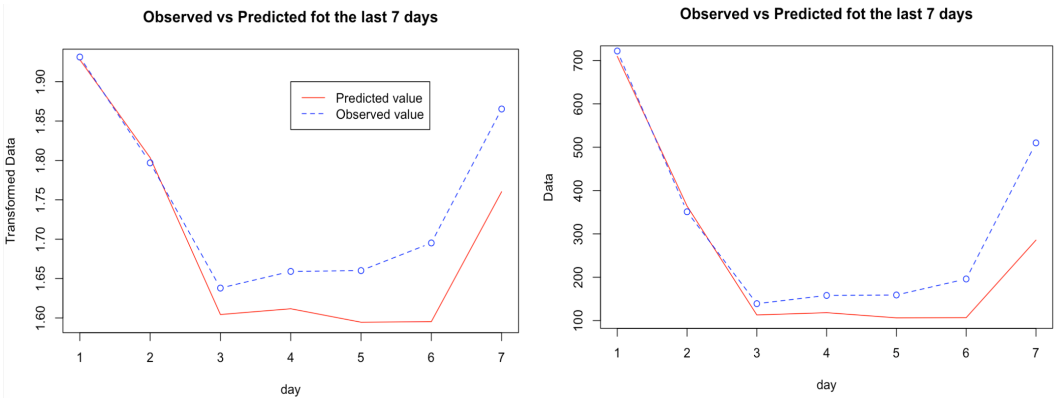

Figure 9: Estimated V.S Observed: Left transformed data; Right raw data

4 Discussion

4.1 Data Analysis Insights

The time series analysis provides valuable insights into the general trend and seasonal trends of movie sales. The general trend shows a downward tendency from January to April. This has set an undertone to movie sales on a monthly time scale. And the seasonal trend precisely cycles over a week, showing a peak on Saturday and higher sales over the weekend. Capturing these patterns helps generate a reasonable 7-day forecasts. Making use of the prediction and movie sales pattern will have wider implication and impacts on movie theaters and movie goers.

4.2 Future Analysis Improvement

From the section **3.4**, only the predicted value and observed value are plotted. However, the predicted interval is more preferable if it is obtained, and this can help the business owner to do a more efficient choice. The predicted interval also consists of 3 parts, the trend, seasonal and rough parts, which might be very wider. In future analysis, some prediction methods can be tried to obtain a reasonable but shorter prediction interval.

5 Conclusion

For the last 7 days, the Observed vs Predicted plot shows a decent fit from combining the trend and stationary components obtained from the previous 99 days, also the observed seasonal effects from data. The last 7 days starts from Saturday (4/12/2003) to Sunday (4/18/2003). Notably the gap between predicted from observed number of receipts grow wider when forecasting over a longer lead time. This phenomenon is expected as both the general trend gradually and stationary components gains larger variance predicting further out in time.

Appendix: R Code

```
# 137 project
chi=read.table('/Users/rduan/Desktop/STA␣137/chicago.txt')
str(chi)
n=c(1:106)
# (3) plot the data
plot(n,chi$V1,type="l")
str(chi)
# seasonal, transformation
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)
```

```r
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)
}
#  plot the raw data
n=c(1:106)
plot(n,chi$V1)
# Loess (span=0.3) and poly compariation
par(mfrow=c(2,2))
for(i in c(2:5)){
  d=i
  trans=trndseas(data$Receipts,sea,lam,d)
  trend=trans$trend
  plot(c(1:106),chi$V1^lamda,type="l",main=paste("Loess␣v.s␣
   polynomial␣(df=",d, ")"),xlab = "time",ylab="tranformed␣Y")
  lines(loess.smooth(c(1:106),chi$V1^lamda,span=0.3),col="red")
  lines(c(1:106),trend,col="blue")
  legend("topright",legend =c("loess","poly"),lty=c(1, 1), col =c("
   red","blue"))
}
# optimal tansformation
sea=7
lam=seq(-2,2,0.1)
par(mfrow=c(2,2))
d=4
trans=trndseas(data$Receipts,sea,lam,d)
trans
lamda=trans$lamopt
# extract fitted, seasonal, rough, etc.
fitted=trans$fitted # fitted=trend+season
trend=trans$trend
season=trans$season
rough=trans$residual
#head(trend+rep(season,length.out=106))
#head(fitted)
plot((chi$V1)^lamda,type="l")
lines(fitted,col="red")
lines(trend,type="l",col="blue")
week=c(1:7)
names(week)=c("Fri","Sta","Sun","Mon","Tus","Wed","Turs")
plot(week,season,type="l")
# rough part diagnostic
mean(rough)
var(rough)
par(mfrow=c(2,2))
plot(rough)
plot(rough,type="l")
hist(rough, main = "Histogram␣of␣rough␣part", xlab = "Rough␣part")
qqnorm(rough, main = "Normal␣probability␣plot␣of␣Rough")
qqline(rough)
acf(rough)
```

```
pacf(rough)
Box.test(rough, lag=10, type="Ljung-Box") # null hypothesis of
   independence
# select arma model
AICc<-matrix(0,6,6)
for(i in 1:6){
  for(j in 1:6){
    AICc[i,j]<-sarima(rough,p=i-1,d=0,q=j-1,details=FALSE)$AICc
  }
}
AICc
best=numeric(6)
for(i in 1:6){
  best[i]=AICc[,i][which.min(AICc[,i])]
}
best
# use auto model selection
auto.arima(rough, max.p = 5,max.q = 5, stepwise=F, approximation=F,ic
   ="aicc")
# compare specific model
sarima(rough,p=1,d=0,q=1,details=FALSE)$AICc
sarima(rough,p=1,d=0,q=0,details=FALSE)$AICc
sarima(rough,p=2,d=0,q=5,details=FALSE)$AICc
sarima(rough,p=3,d=0,q=3,details=FALSE)$AICc
# fit model
model<-arima(rough,order=c(1,0,0))
acf(model$residual) # residual white noise choose arma(1,0)
summary(model)
# spectrual density
specselect=function(y,kmax){
  # Obtains the values of the criterion function for
  # obtaining the optimal number of neighbors for
  # spectral density estimate for modified Daniell's method.
  # input: y, observed series; kmax=max number of neighbors to be
   considered
  # output: ctr - the criterion function
  # output: kopt - the value of k at which the criterion function
  # is minimized
  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=rough,12)
par(mfrow=c(1,1))
plot(c(1:12),specselect(rough,12)$ctr,type="o",xlab="number␣of␣
   neighbors",ylab="value␣of␣criterion␣function",main="Neighbor␣
   selection␣for␣the␣smoothed␣periodogram") #7
spec_smooth <-spec.pgram(rough, spans=15, log='no', plot = FALSE)
freq <- spec_smooth$freq
spec <- spec_smooth$spec
# raw
#rf=spec.pgram(rough,log="no")
#plot(rf$freq,rf$spec,type="l" )
plot(freq,spec,type="l" )
# plot spectral density and smoothing
library(astsa)
arma.spec(ar = model$coef[1], var.noise = model$sigma2, log='no')
lines(freq, spec, col = "blue")
legend("top",legend =c("spectral␣density␣function","smoothed␣
   periodogram"),lty=c(1, 1), col =c("black","blue"))

# Forecast
#split data
new_data=data[-c(100:106),]
h <- 5
m=99
```

```
tm2=1:99
#trend
library(Hmisc)
fit2=trndseas(new_data$Receipts,7,0.1,4)
tr2=fit2$trend
tr_pre=approxExtrap(x=tm2,y=tr2, xout=c(100:106), method = "linear")
#rough
r_pre=sarima.for(fit2$residual,7,1,0,0,plot=FALSE)$pred
#season
s_pre=rep(trans$season,length.out=106)[100:106]
# predicted
pre_7=r_pre+s_pre+tr_pre$y
ob_7=(data[100:106,1])^(0.1)
day=1:7
# PLOT
plot(x=day,y=pre_7,type = 'l',col="red",ylab = "Transformed␣Data",
   main = "Observed␣vs␣Predicted␣fot␣the␣last␣7␣days")
lines(x=day, y=ob_7,type ='b',col="blue",lty=2)
legend(4,1.90,legend=c("Predicted␣value", "Observed␣value"),col=c("
   red", "blue"),lty=1:2)

plot(x=day,y=pre_7^10,type = 'l',col="red",ylab = "Data", main = "
   Observed␣vs␣Predicted␣fot␣the␣last␣7␣days")
lines(x=day, y=ob_7^10,type ='b',col="blue",lty=2)
legend(4,1.90,legend=c("Predicted␣value", "Observed␣value"),col=c("
   red", "blue"),lty=1:2)
```