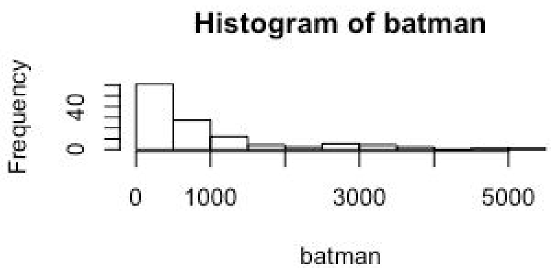
1/

Introduction: The data on batman.xls is the daily average receipts per theater for the movie Batman from June 26, 1989 to October 22, 198. This is a time series because it is so obvious that the set of data is based on time. The time series model achieve to estimate the trend , seasonal components, model the stationary part and forecast the number of the daily receipts.

2/

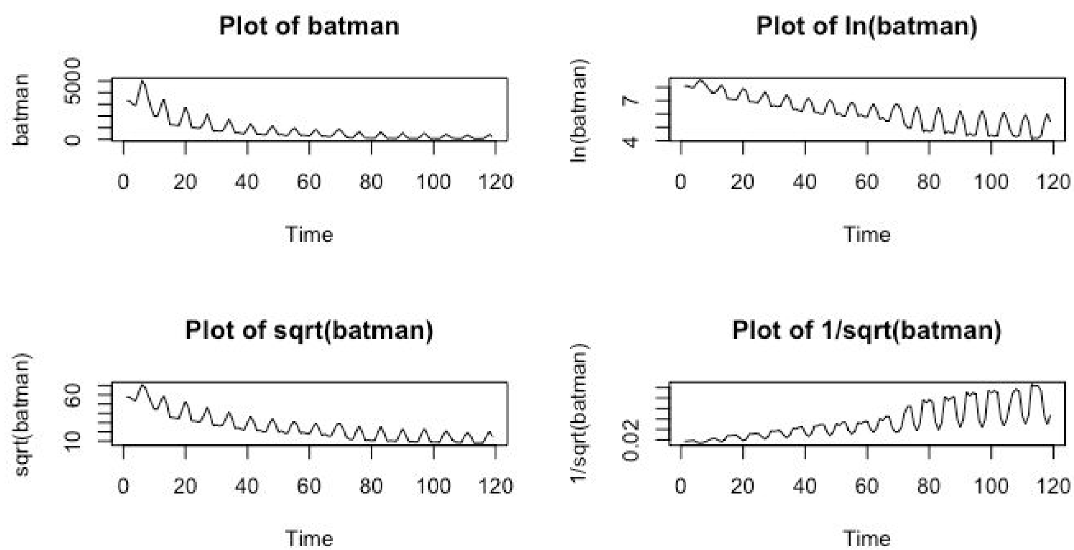


The nature of the variations in the data based on the histogram plot we can see it is heavy-tail on the right so it needs to transformed and type of modeling scheme appropriate here may be a transformed model.

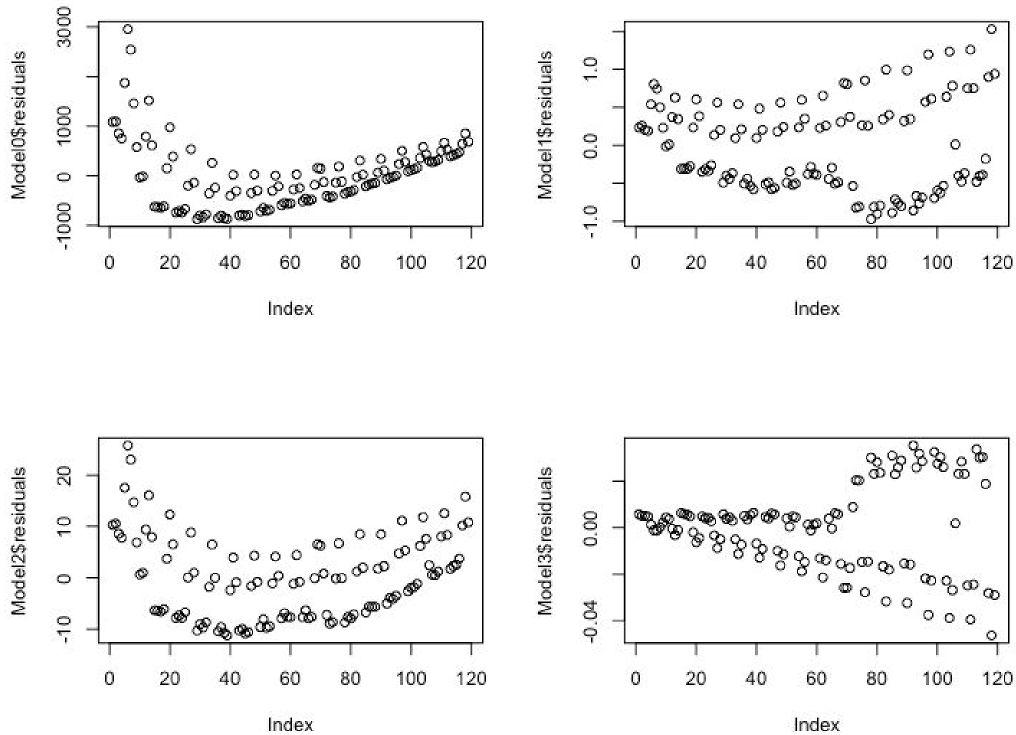
3/

```
> library(readxl)
> batman = read_xlsx("batman.xlsx", col_names = FALSE)
> names(batman) = c("Receipts", "Date", "Day")
> batman = ts(batman[,1])
> batmanT1 = log(batman)
> batmanT2 = sqrt(batman)
> batmanT3 = batman^{-1/2}
> par(mfrow = c(2,2))
> plot.ts(batman, ylab = 'batman', main = 'Plot of batman')
> plot.ts(batmanT1, ylab = 'ln(batman)', main = 'Plot of ln(batman)')
> plot.ts(batmanT2, ylab = 'sqrt(batman)', main = 'Plot of sqrt(batman)')
> plot.ts(batmanT3, ylab = '1/sqrt(batman)', main = 'Plot of 1/sqrt(batman)')
> time= 1:119
> Model0 = lm(batman~time)
> Model1 = lm(batmanT1~time)
> Model2 = lm(batmanT2~time)
> Model3 = lm(batmanT3~time)
> par(mfrow = c(2,2))
```

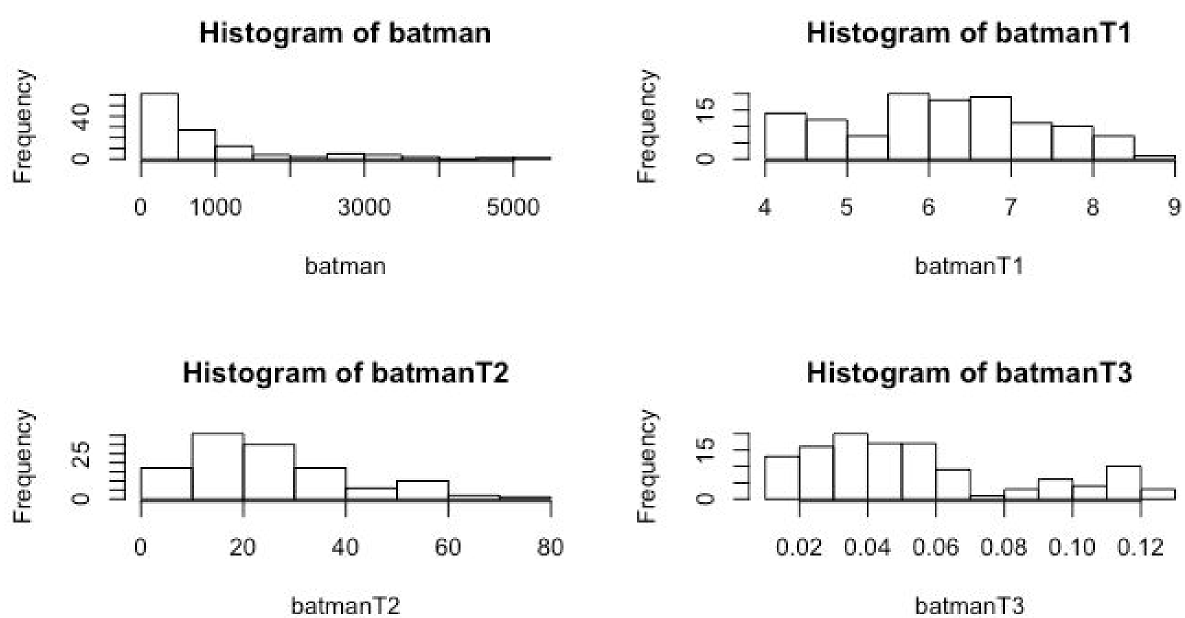
```
> plot(Model0$residuals)
> plot(Model1$residuals)
> plot(Model2$residuals)
> plot(Model3$residuals)
> par(mfrow = c(2,2))
> hist(batman)
> hist(batmanT1)
> hist(batmanT2)
> hist(batmanT3)
```



Plot the data as well as various Box-Cox transformations:
According to the 4 plots of raw and transformed data on batman variable, the model with log of batman has the fluctuations and appeal to be mostly the same over the time period while the rest (including the model with batman and the other two transformed model of batman) has a larger fluctuation variation.
Hence, the most appropriate model that should be used is the logarithm transformation.



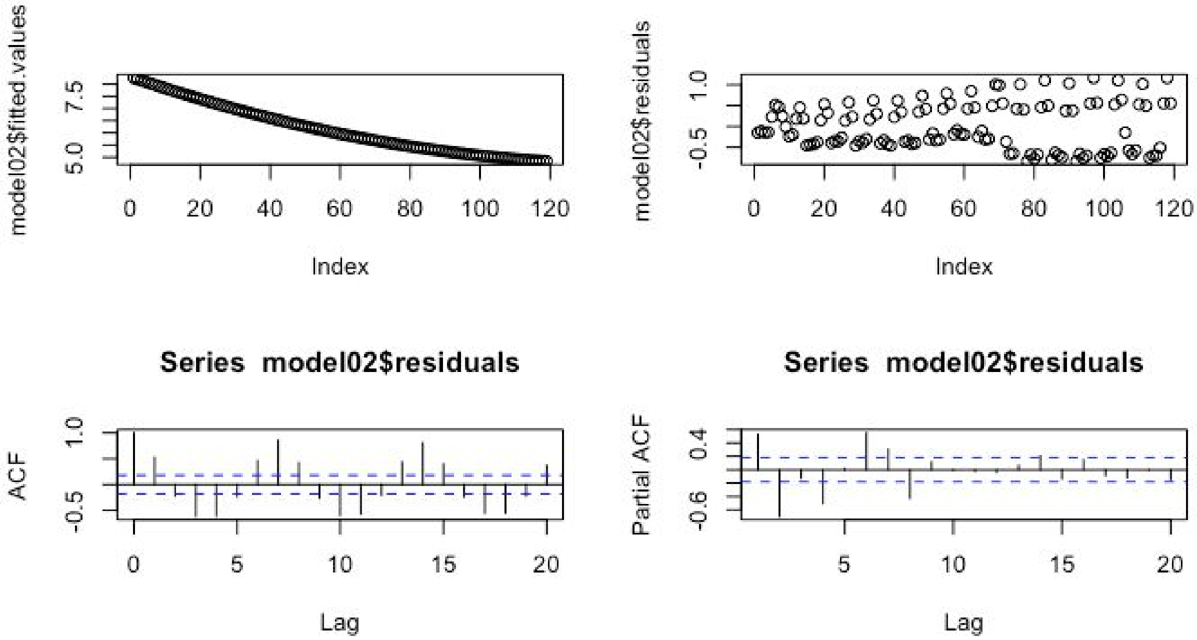
According to the 4 plots of the 4 models' residuals, the residuals that were based on the logarithm model (log(batman) model) displays a variance that seems to be more equal compared to the other 3 models, so the Model1 with log(batman) is the most appropriate model.



Based on the 4 histogram plots, the histogram that were based on the logarithm model (log(batman)) is the most normally distributed histogram compared to the other models, so the the model with log(batman) is the most appropriate model.

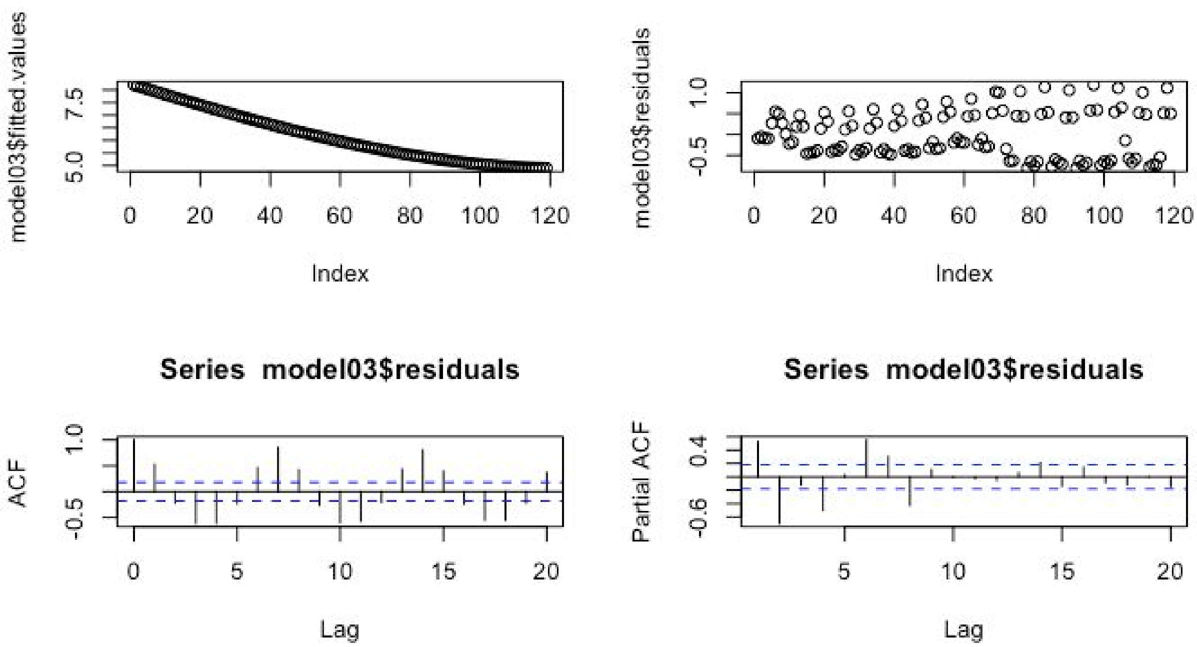
4/

```
> t = 1:119
> model02 = lm(batmanT1~poly(t,2))
> summary(model02)

Call:
lm(formula = batmanT1 ~ poly(t, 2))

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8355 -0.4211 -0.1461  0.4651  1.1662 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.15824    0.05132 119.988  < 2e-16 ***
poly(t, 2)1 -10.85696    0.55988 -19.392  < 2e-16 ***
poly(t, 2)2   1.92908    0.55988   3.446 0.000794 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5599 on 116 degrees of freedom
Multiple R-squared:  0.7698,  Adjusted R-squared:  0.7658 
F-statistic:   194 on 2 and 116 DF,  p-value: < 2.2e-16
> par(mfrow = c(2,2))
> plot(model02$fitted.values)
> plot(model02$residuals)
> acf(model02$residuals)
> pacf(model02$residuals)
> AIC(model02)
[1] 204.6203
```



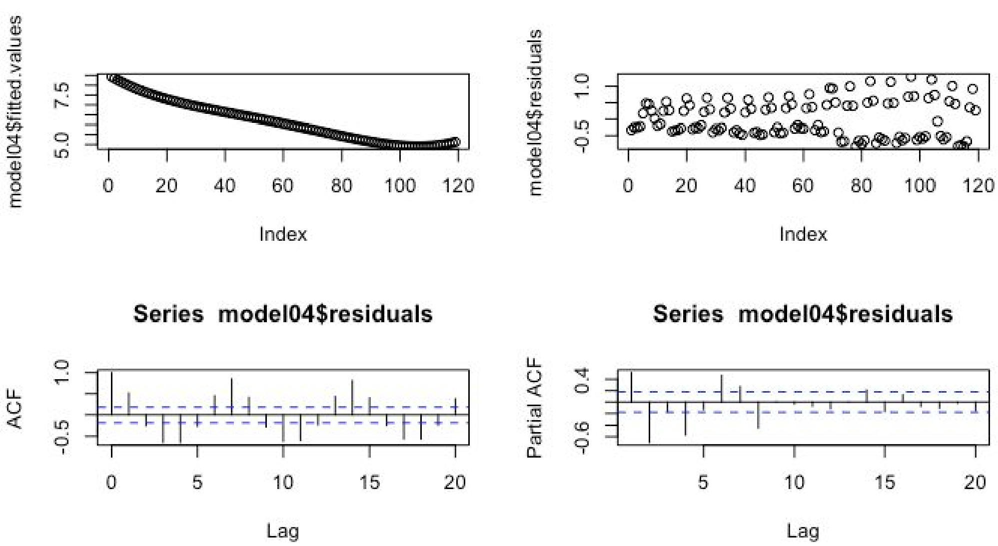

```
> model03 = lm(batmanT1 ~poly(t,3))
> summary(model03)

Call:
lm(formula = batmanT1 ~ poly(t, 3))

Residuals:
     Min       1Q   Median       3Q      Max
-0.81093 -0.43881 -0.09792  0.47960  1.18574

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.1582     0.0515 119.566  < 2e-16 ***
poly(t, 3)1 -10.8570     0.5618 -19.324  < 2e-16 ***
poly(t, 3)2   1.9291     0.5618   3.433 0.000829 ***
poly(t, 3)3   0.2426     0.5618   0.432 0.666682
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5619 on 115 degrees of freedom
Multiple R-squared:  0.7702,  Adjusted R-squared:  0.7642
F-statistic: 128.5 on 3 and 115 DF,  p-value: < 2.2e-16
```

```
> par(mfrow = c(2,2))
> plot(model03$fitted.values)
> plot(model03$residuals)
> acf(model03$residuals)
> pacf(model03$residuals)
> AIC(model03)
[1] 206.4275
```




```
> model04 = lm(batmanT1~poly(t,4))
> summary(model04)

Call:
lm(formula = batmanT1 ~ poly(t, 4))

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8281 -0.4190 -0.2007  0.4518  1.2938 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)   6.15824    0.05111 120.498  < 2e-16 ***
poly(t, 4)1 -10.85696    0.55751 -19.474  < 2e-16 ***
```

```
poly(t, 4)2   1.92908    0.55751   3.460  0.00076 ***
poly(t, 4)3   0.24262    0.55751   0.435  0.66425
poly(t, 4)4   0.93292    0.55751   1.673  0.09699 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5575 on 114 degrees of freedom
Multiple R-squared:  0.7757,  Adjusted R-squared:  0.7678
F-statistic: 98.55 on 4 and 114 DF,  p-value: < 2.2e-16
> par(mfrow = c(2,2))
> plot(model04$fitted.values)
> plot(model04$residuals)
> acf(model04$residuals)
> pacf(model04$residuals)
> AIC(model04)
[1] 205.5398
```



Since the AIC value is the smallest for 2nd degree polynomial model (AIC(model02) = 204.6203), the model with 2nd degree polynomial is the most appropriate. Hence, we choose the quadratic polynomial.

//Applying the trndseas function from trndseas.R file

```
> lam = seq(-1,1,by=0.05)
> ff = trndseas(batmanT1,seas = 5,lam = 1,degtrnd = 2)
> rsq = ff$rsq
```

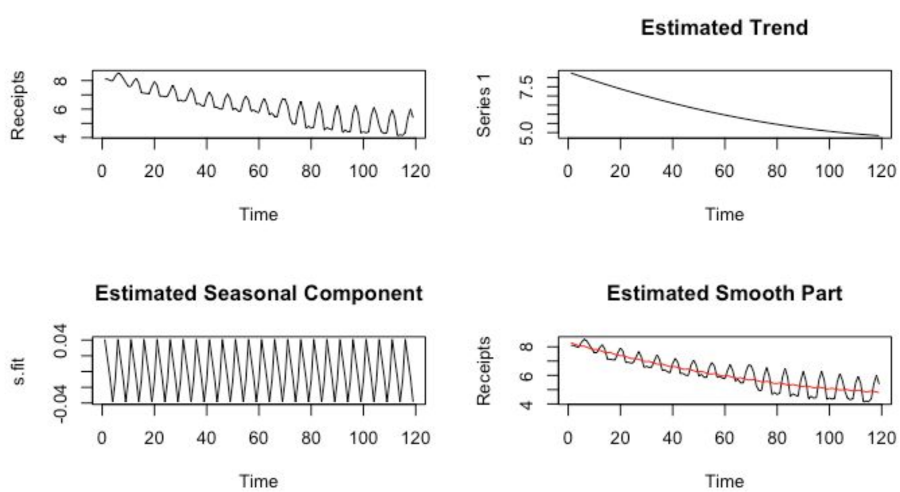
```
> rsq
[1] 0.7703482
> attributes(ff)
$names
[1] "coef"   "fit"    "trend"  "res"    "season" "rsq"    "lamopt"

> m.fit = ff$trend
> ff$season
[1]  0.040975243  0.016536082 -0.007794892 -0.038500238 -0.011216195
```

The seasonal components for the transformed data are:

```
> ff$season
[1]  0.040975243  0.016536082 -0.007794892 -0.038500238 -0.011216195
```

5/

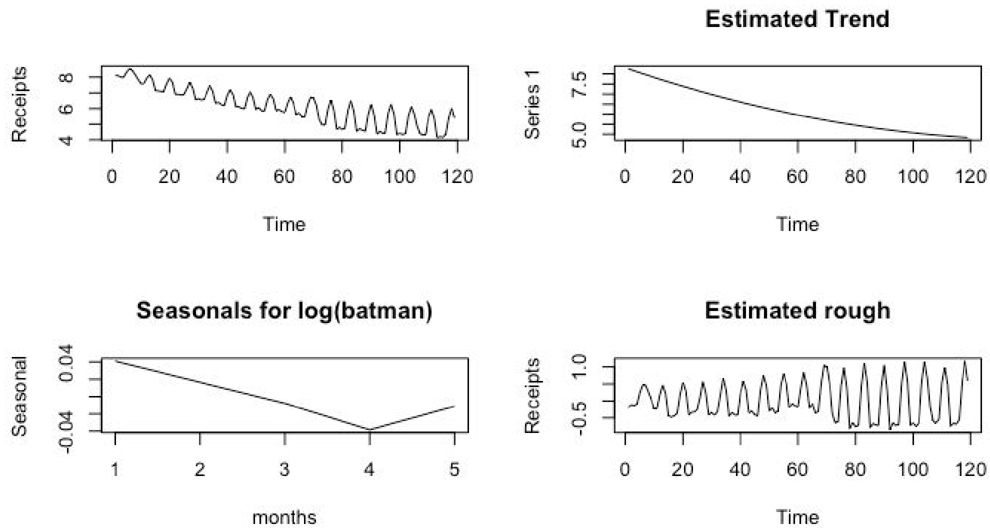
```
> n = length(batmanT1)
> s.fit = rep(ff$season,length.out=n)
> smooth.fit = ff$fit
> par(mfrow=c(2,2))
> plot.ts(batmanT1)
> plot.ts(m.fit, main='Estimated Trend')
> plot.ts(s.fit,main='Estimated Seasonal Component')
> plot.ts(batmanT1,main='Estimated Smooth Part')
> points(smooth.fit,type='l',col='red')
```



Plotting in line types:

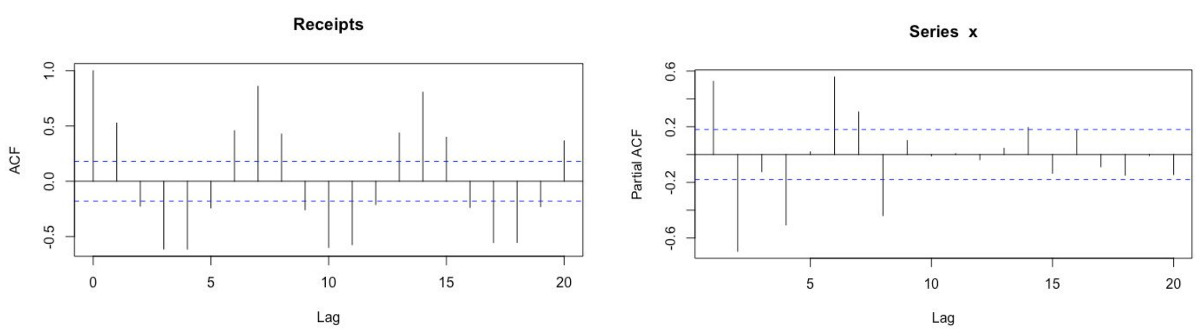
```
> plot.ts(batmanT1)
```

```
> plot.ts(m.fit, main='Estimated Trend')
> months = 1:5
> plot(months, ff$season, type='l', ylab = 'Seasonal', main = 'Seasonals for log(batman)')
> plot.ts(ff$res,type = 'l', main = "Estimated rough")
```



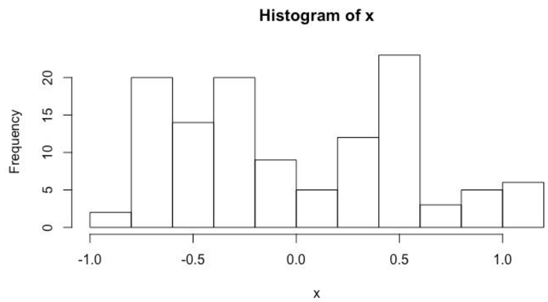
6/

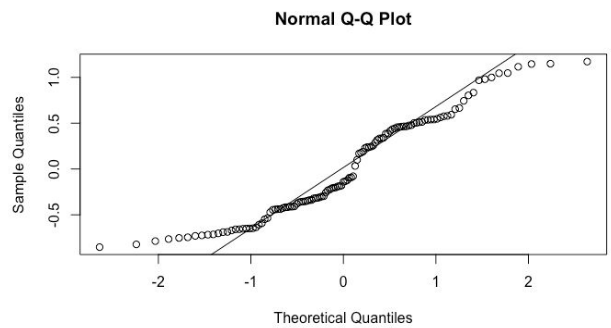
```
> par(mfrow=c(1,1))
> x = batmanT1-m.fit-s.fit
> acf(x)
> pacf(x)
> hist(x)
> qqnorm(x)
> qqline(x)
```



The ACF plot for the estimated rough for the log(batman) [batmanT1] data indicates that most lines/points fall out of the blue lines as none of the correlations of lags is to zero. But more analysis should be performed on the data.

The PACF plot for the estimated rough for the log(batman) [batmanT1] data indicates that correlations of lags 1,2,4,6,7,8,14,16 may not be close to zero, and the rest fall within the blue lines.





The histogram shows that the data doesn't hold the normality assumption of the residuals. The normal QQ plot shows that most of the points don't fall within the linear line with no equal variance, showing that the QQ plot doesn't hold normality assumption of the residuals. Hence, the rough doesn't indicate the normality.

7/

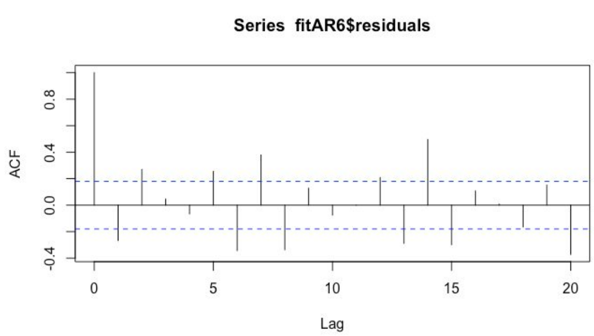
From the above plots we see that the PACF is insignificant after lag 2 and the ACF is significant. We'll try several choices and compare.

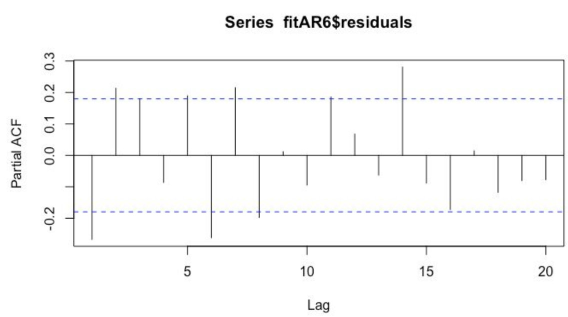
```
> fitAR0 = arima(x,order=c(0,0,0))
> fitAR1 = arima(x,order=c(1,0,0))
> fitAR2 = arima(x,order=c(2,0,0))
> fitAR3 = arima(x,order=c(3,0,0))
> fitAR4 = arima(x,order=c(4,0,0))
> fitAR5 = arima(x,order=c(5,0,0))
> fitAR6 = arima(x,order=c(6,0,0))
> aicc = function(model){
+   n = model$nobs
+   p = length(model$coef)
+   aicc = model$aic + 2*p*(p+1)/(n-p-1)
+   return(aicc)
+ }
> aiccAR0 = aicc(fitAR0)
> aiccAR1 = aicc(fitAR1)
> aiccAR2 = aicc(fitAR2)
> aiccAR3 = aicc(fitAR3)
> aiccAR4 = aicc(fitAR4)
```

```
> aiccAR5 = aicc(fitAR5)
> aiccAR6 = aicc(fitAR6)
> AICC = c(aiccAR0,aiccAR1,aiccAR2,aiccAR3,aiccAR4,aiccAR5,aiccAR6)
> AICC
[1] 200.370356 163.651421  86.380738  86.003140  50.057086  52.251216   3.523824
```

As we can see based on the AICC criterion, the AR(6) has the smallest value 3.523824 so AR(6) is the most appropriate AR model.

```
> par(mfrow=c(1,1))
> plot.ts(fitAR6$residuals)
> acf(fitAR6$residuals)
> pacf(fitAR6$residuals)
> Box.test(fitAR6$residuals,lag=10,'Ljung-Box')

        Box-Ljung test

data:  fitAR6$residuals
X-squared = 77.719, df = 10, p-value = 1.403e-12
```



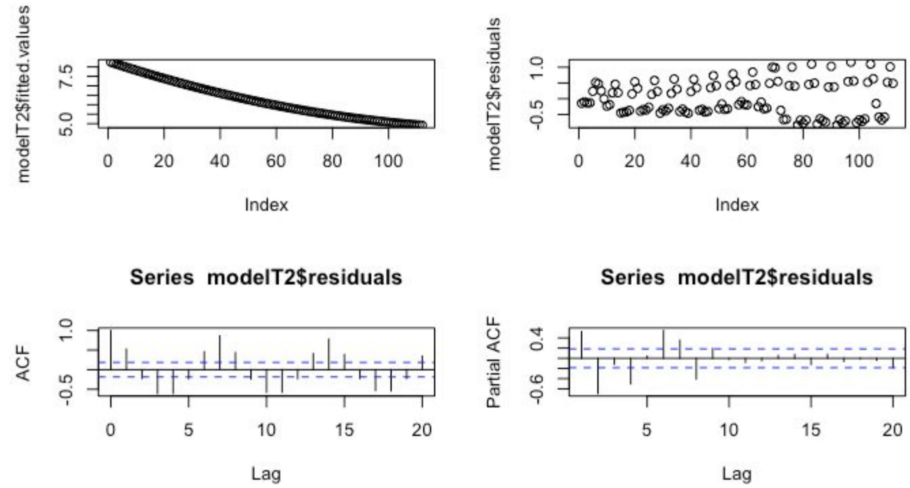


Based on the AR(6) pmodel which is a good fit of the data, since the residuals have white noise and the plot has the insignificant after lag 2 and it is also consistent with white noise so the model is a good fit.

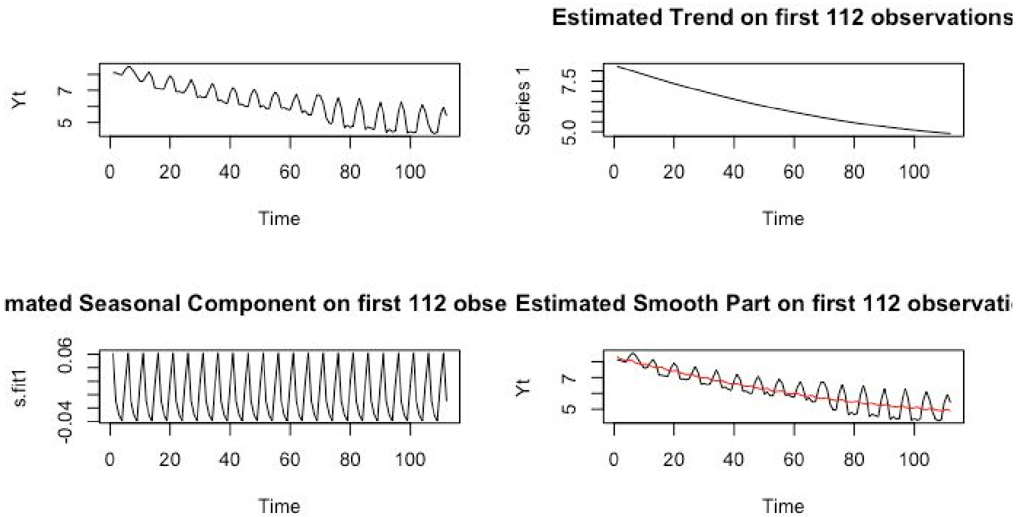
8/

```
> t2 = 1:112
> Yt = batmanT1[1:112]
> modelT2 = lm(Yt~poly(t2,2))
```

```
> summary(modelT2)

Call:
lm(formula = Yt ~ poly(t2, 2))

Residuals:
    Min      1Q  Median      3Q     Max
-0.8428 -0.4103 -0.1420  0.4541  1.1587

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.24252    0.05167 120.817  < 2e-16 ***
poly(t2, 2)1 -10.27437    0.54682 -18.789  < 2e-16 ***
poly(t2, 2)2   1.64669    0.54682   3.011  0.00323 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5468 on 109 degrees of freedom
Multiple R-squared:  0.7686,  Adjusted R-squared:  0.7644
F-statistic: 181.1 on 2 and 109 DF,  p-value: < 2.2e-16

> par(mfrow = c(2,2))
> plot(modelT2$fitted.values)
> plot(modelT2$residuals)
> acf(modelT2$residuals)
> pacf(modelT2$residuals)
> AIC(modelT2)
[1] 187.5852
```




```
> lam = seq(-1,1,by=0.05)
> ff1 = trndseas(Yt,seas = 5,lam = 1,degtrnd = 2)
> ff1 = trndseas(Yt,seas = 5,lam = 1,degtrnd = 2)
> rsq1 = ff1$rsq
> rsq1
[1] 0.7696942
> attributes(ff1)
$names
[1] "coef"   "fit"    "trend"  "res"    "season" "rsq"    "lamopt"

> m.fit1 = ff1$trend
> season1 = ff1$season
> season1
[1]  0.061904670 -0.009797489 -0.030249967 -0.038712646  0.016855431
> n1 = length(Yt)
> s.fit1 = rep(season1,length.out=n1)
> smooth.fit1 = ff1$fit
> par(mfrow=c(2,2))
> plot.ts(Yt)
> plot.ts(m.fit1, main='Estimated Trend on first 112 observations')
> plot.ts(s.fit1,main='Estimated Seasonal Component on first 112 observations')
> plot.ts(Yt,main='Estimated Smooth Part on first 112 observations')
> points(smooth.fit1,type='l',col='red')
```



Plotting in line types:

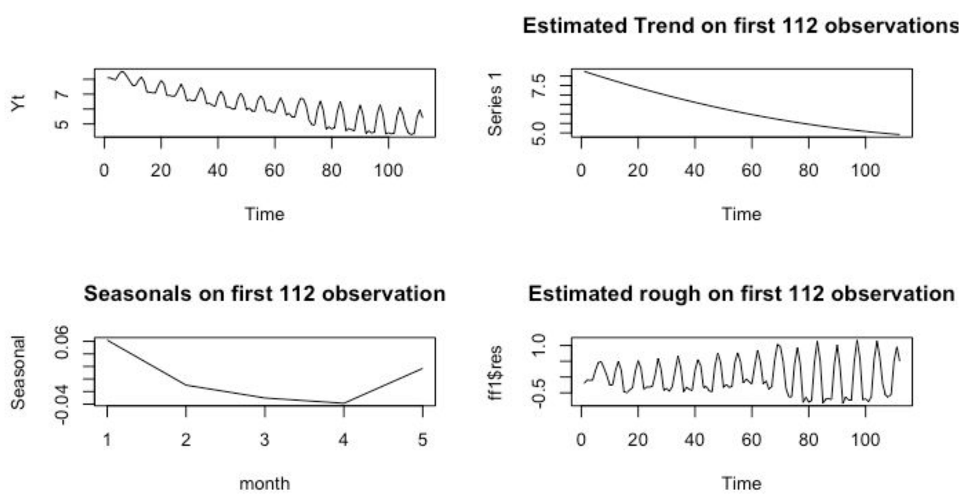
> plot.ts(Yt)

> plot.ts(m.fit1, main='Estimated Trend on first 112 observations')

> month = 1:5

> plot(month, season1, type='l', ylab = 'Seasonal', main = 'Seasonals on first 112 observation')

> plot.ts(ff1$res,type = 'l', main = "Estimated rough on first 112 observation")



> par(mfrow=c(2,2))

> x1 = Yt-m.fit1-s.fit1

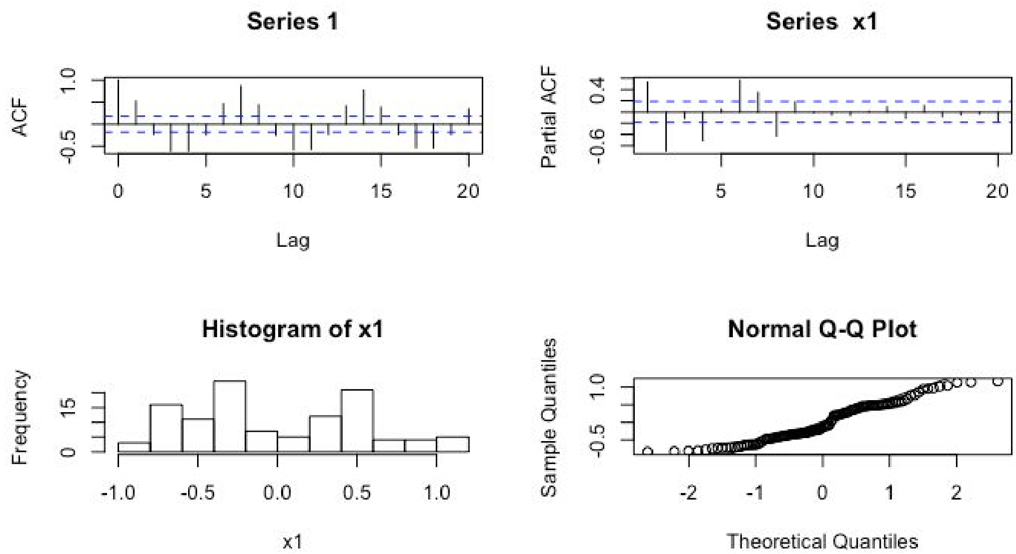
> acf(x1)

> pacf(x1)

> hist(x1)

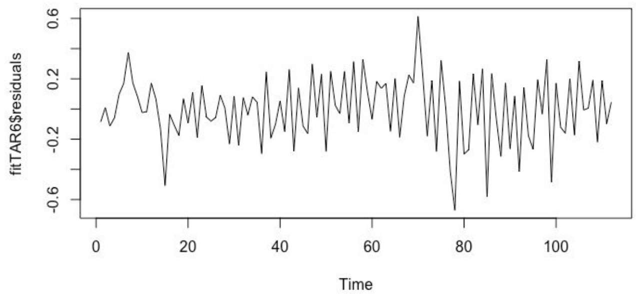
> qqnorm(x1)

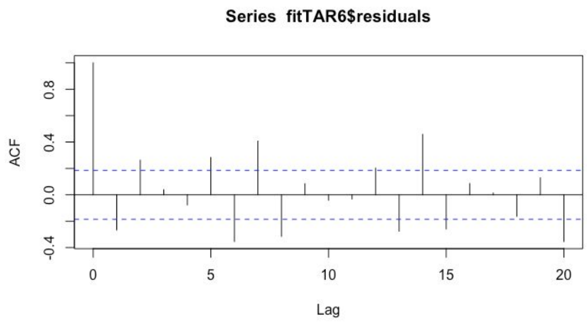
> qqline(x1)

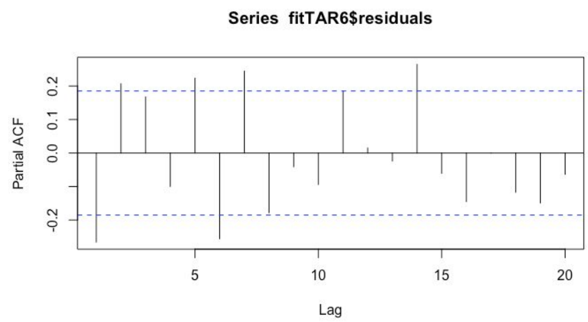


The ACF plot for the estimated rough for the Yt data indicates that most lines/points fall out of the blue lines as none of the correlations of lags is close to zero. But more analysis should be performed on the data.The PACF plot for the estimated rough for the Yt data indicates that it is insignificant after lag 2.

```
> fitTAR6 = arima(x1,order=c(6,0,0))
> aiccTAR6 = fitTAR6$aic + 2*7*(7+1)/(n1-7-1)
> aiccTAR6
[1] 0.9871589
> par(mfrow=c(1,1))
> plot.ts(fitTAR6$residuals)
> acf(fitTAR6$residuals)
> pacf(fitTAR6$residuals)
> Box.test(fitTAR6$residuals,lag=10,'Ljung-Box')

        Box-Ljung test

data:  fitTAR6$residuals
X-squared = 74.974, df = 10, p-value = 4.813e-12
```

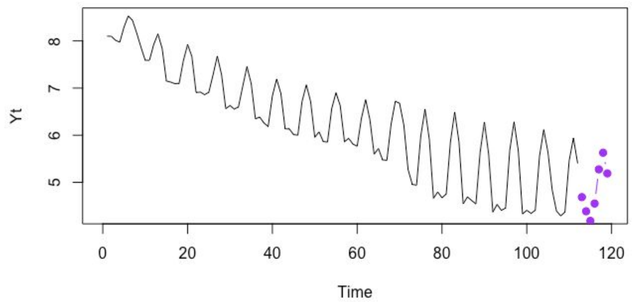







```
> library(Hmisc)
> trend7Days = approxExtrap(m.fit1[1:112],m.fit1[1:105],xout=m.fit1[106:112],
method="linear")[1]
> trend7Days
$x
[1] 4.987823 4.974325 4.961157 4.948321 4.935815 4.923639 4.911794
> ff2 = trndseas(Yt[106:112],seas = 5,lam = 1,degtrnd = 2)
> ff2$season
[1]  0.64605732 -0.05481775 -0.39871756 -0.54050701  0.34798500

> h = 7
> deg = 2
> coef = ff1$coef[1:(deg+1)]
> time1 = (n1+(1:h))/n1
> predmat = matrix(rep(time1,deg)^rep(1:deg,each=h),nrow=h,byrow=FALSE)
> predmat = cbind(rep(1,h),predmat)
> predmat
     [,1]     [,2]     [,3]
[1,]    1 1.008929 1.017937
[2,]    1 1.017857 1.036033
```

```
[3,]    1 1.026786 1.054289
[4,]    1 1.035714 1.072704
[5,]    1 1.044643 1.091279
[6,]    1 1.053571 1.110013
[7,]    1 1.062500 1.128906
> m.fc = predmat %*% coef
> s.fc = rep(ff1$season,length.out=n1+h)
> s.fc = s.fc[-(1:n1)]
> s.fc
[1] -0.030249967 -0.038712646  0.016855431  0.061904670 -0.009797489 -0.030249967
-0.038712646
> fcast = predict(fitTAR6,n.ahead=h)
> x.fc = fcast$pred
> x.fc
Time Series:
Start = 113
End = 119
Frequency = 1
[1] -0.1833342 -0.4633361 -0.7119789 -0.3786019  0.4276200  0.8117001  0.3902845
> y.fc = m.fc + s.fc + x.fc
> y.fc
Time Series:
Start = 113
End = 119
Frequency = 1
        [,1]
[1,] 4.686696
[2,] 4.387048
[3,] 4.183120
[4,] 4.551024
[5,] 5.275353
[6,] 5.629119
[7,] 5.189711
> plot.ts(Yt,xlim=c(0,n1+h))
> points(x=n1+1:h, y=y.fc, col='purple',type='b',pch=19)
```



9/

First we used the data of the of the daily average receipts per theater for the movie Batman and then we interpreted that it is a time series data since the data are observed over time. Then we plot the time series of the data based on the history plot and observed that it is not stationary, so we transformed the data using square, logarithm, 1/square root and we found out the model with log of batman has the fluctuations and appeal to be mostly the same over the time period. After doing Box-Cox transformations, we estimate the trend and the seasonal components based on the transformed data. Next, we plot the ACF and PACF plot of the first difference of log model and identified an ARIMA model. After that we performed AICC test, the AR(6) has the smallest value 3.523824 it the same as our identification of a time series model. So the final model is the model with log(Batman) , applied on the quadratic polynomial. We used all except the last 7 observations of the 112 observation- model and use the model to forecast the last 7 days of data. The predicted values an upward trend except one last day is downward so it seems accurate based on the observed data.