**Project 1**

**John Nguyen**

**Spencer Phillips**

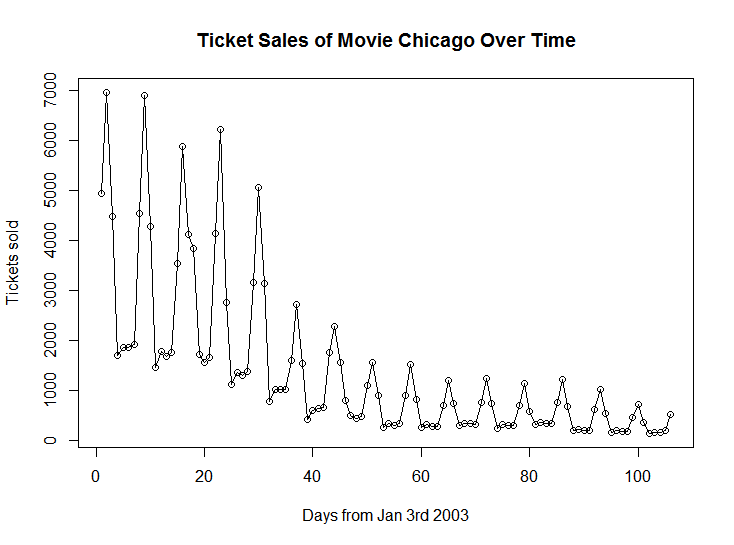
**Jack Daum III**

**May 11, 2016**

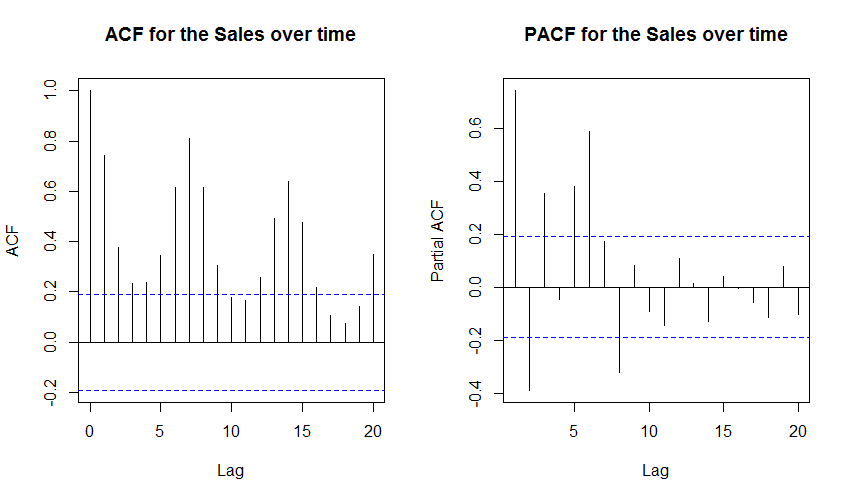
**STA 137  
Professor Prabir Burman**

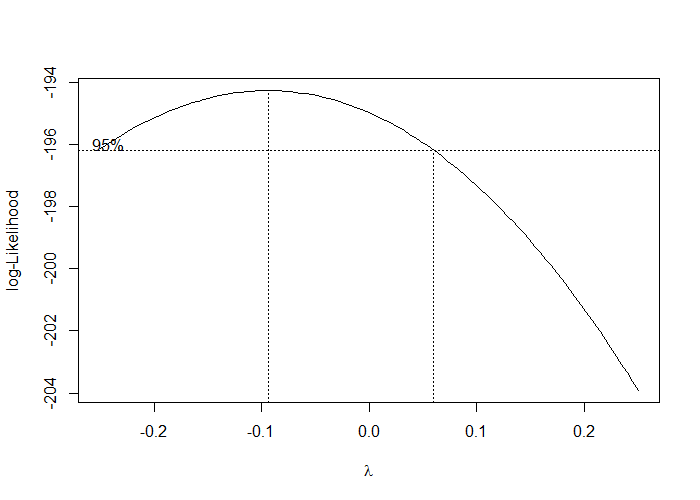
The data being analyzed is composed of receipts produced by theaters for the movie *Chicago* from January 3, 2003 to April 18, 2003. The data is not being compared against individual theaters, rather it is being compared on a total amount of receipts produced by the theaters per day within the time period. This makes it a time series: a quantifiable amount of something compared over a period of time that is divided into segments. Knowing that this is a time series allows us to apply modeling that can help us find any correlations within the data. The models will also allow us to make a predicted forecast on the data using the data we have. This forecast, along with the current data, can then be used as a model for future films in that are similar. We will also be able to see if there were other variables that could have contributed to receipts produced during the time period that may have affected the data.

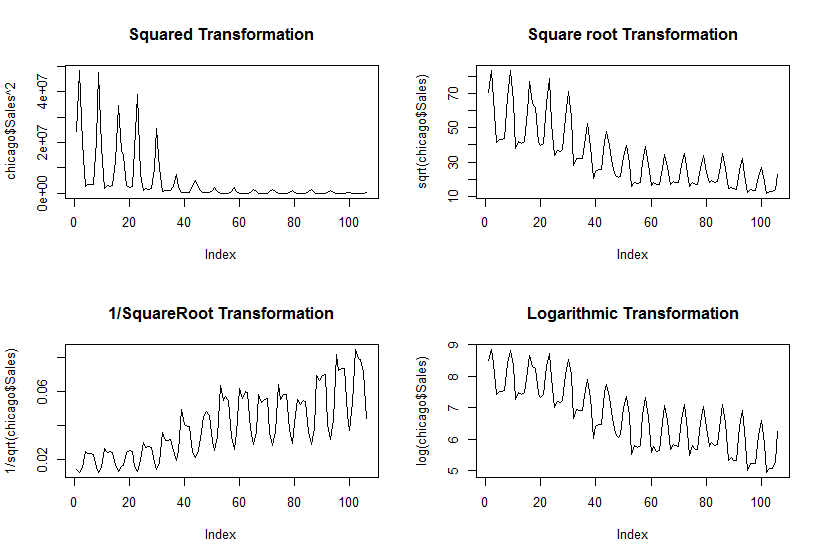
By comparing the number of receipts produced against the time period (separated into days of the week) we see that seasonality is occurring. The seasonality fluctuates on a weekly time scale, specifically from Saturday to Saturday. We can determine that this is because more people were able to go out and watch the movie on Saturdays. During the week, receipts fell to a consistent level then rose on Fridays to their peak on Saturdays and fell again on Sundays to their lows during the workweek. It is important to note that MLK Jr. Day (which is a Monday in the third week) didn’t see as low of a turnout of people going to theaters as compared to other Mondays. It is also important to note that the data suggests that people went to theaters for the movie less and less as time went on.



Based on the PACF and ACF graphs, an appropriate model for the data would be, we assume, AR(8). This is because the ACF plot trails off to after a certain point and the PACF cuts off at lag 8. The partial autocorrelations of lag 9 and higher are negligible, showing AR(8) may be a reasonable model for the data.

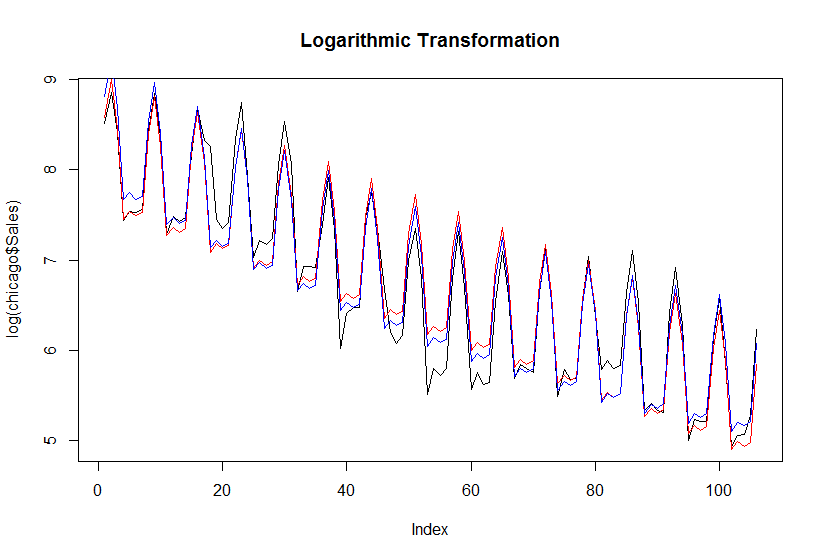
  
PACF provides a better assessment of the data in terms of autocorrelation. There are signs of autocorrelation at lags 1, 2, 3, 5, 6, and 8.





We performed transformations on the data by squaring, taking the square root, taking the square root under one, and taking the natural log of the data. From looking at the transformations, we made the decision to use the logarithmic transformation due to the fact that the variance is similar throughout. All of the models do show a trend, however the logarithmic transformation we perform gives us the least drastic trend among them all. Using the Box-Cox, plot, we see that it trends between lambda being around -0.1 to 0.075, this includes 0 which would call for a logarithmic transformation providing further evidence to our decision of taking the natural log.

We then use the the trendseas() function on the transformed data, using a degree trend of 2 and a seasonality of 7. We chose the degree trend of two by graphing both the trendseas() of degree 1 and 2 with seasonality of 7 for each against each other and getting their R2 values.



In this graph, the black line is the logarithmically transformed plot, the blue line is degree one while the red line is degree two. We see that in many places that the red line is the more optimal choice, being slightly closer to the transformation at multiple points. We also see this with the R2 values with degree 1 having R2= 0.9362437 and degree 2 having R2= 0.9493174. This tells us that degree 2 is a better fit for the model.

Now using the trendseas() we get the following seasonal values for their related days as follows:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Friday | Saturday | Sunday | Monday | Tuesday | Wednesday | Thursday |
| 0.4758571 | 0.9419817 | 0.4161853 | 0.5478656 | -0.4312050 | -0.4602492 | -0.3947043 |

We then extracted the coefficients for the trend and the seasonal components using the trendseas function again giving us the following:

> mod$coef

[,1]

8.3749356

1 -4.3393836

2 1.5678558

x21 0.4758571

x22 0.9419817

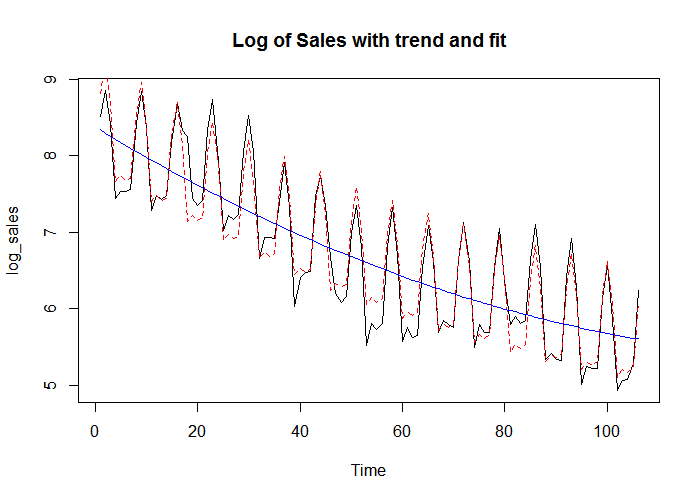
x23 0.4161853

x24 -0.5478656

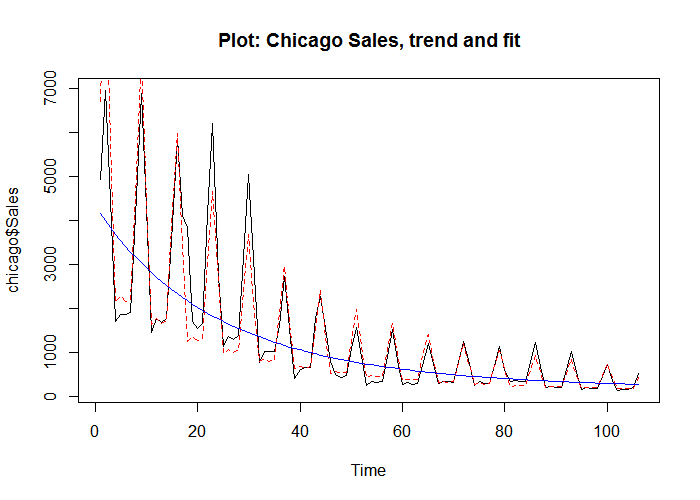
x25 -0.4312050

x26 -0.4602492

We then created plots using the results from trendseas() to make plots for the trend and fit next to our transformed data.



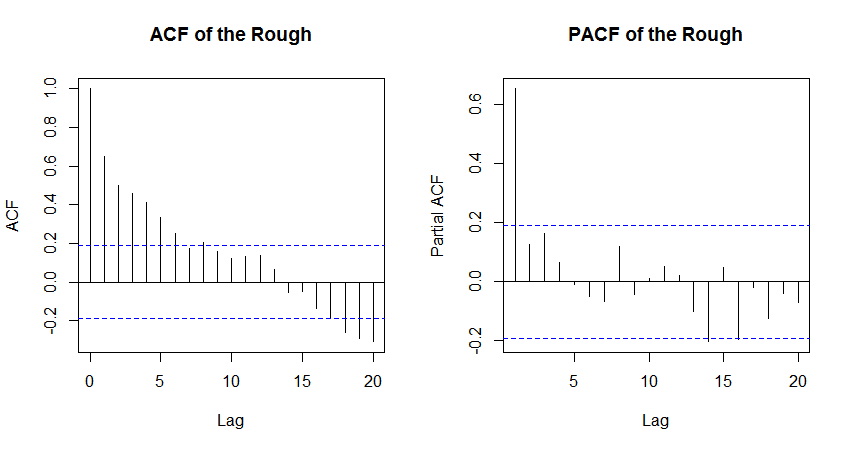
In the above graph, we see that the blue line is the trend, the dotted red line is the fit and the black line is our logarithmically transformed data. The fit is very close to the data that we have showing that this is a good set up.



The plot above is the fit transformed back into the standard form. The blue line is the trend, the black solid line is the original non-transformed data, and the dotted red line is the fit transformed back to standard values We see that the fit is fairly strong and close to the actual data with there still being the negative trend. Do note that the MLK Jr holiday was unable to be forecasted by the fit!



Here we see the seasonality of the days where 1 is equivalent to Friday, 2 is Saturday and so on. We see that most ticket sales are on Friday, Saturday, and Sunday with a huge drop off for Monday through Thursday with all of them being negative values. From it we see that Mondays are by far the worst days for movies, which is odd due to the fact that most movie theaters have discount days on Tuesdays but explained why they pick Tuesdays by the huge outlier that was MLK Jr Day on the Monday of the 3rd week since most American National holidays tend to give Mondays off.



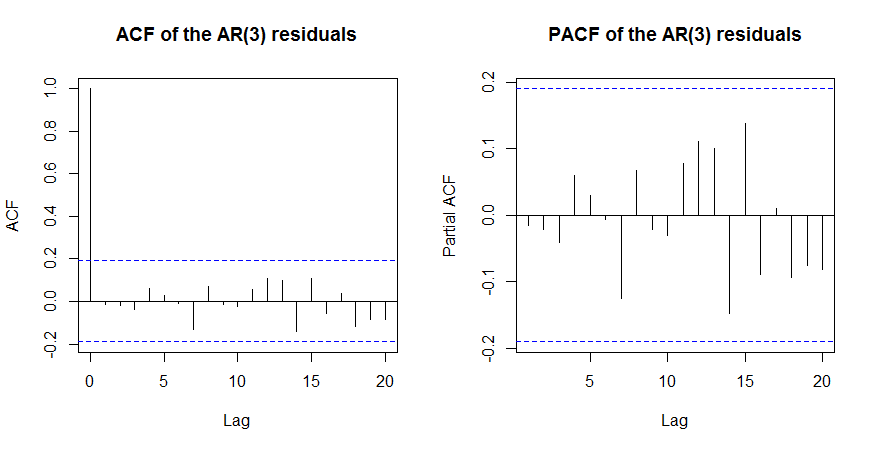
Looking at the ACF of the rough, we see that it tails off instead of ending after a certain lag point. This suggested that we should use the PACF for an AR() model.

The PACF of the Rough shows that the is a heavy autocorrelation at lag 1 and 14 and possibly at lag 16. Due to the strong drop off after lag 1, we could possibly use an AR(1) model, to be sure though, we would need to test for AICC to see which is best.

Testing AICC for AR(q) where q=0,1,2,3,4,5 gives us the following:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AR(0) | AR(1) | AR(2) | AR(3) | AR(4) | AR(5) |
| -6.978861 | -64.74993 | -64.73038 | -65.50266 | -63.8161 | -61.57668 |

This tells us that the best model for us will be AR(3) this is since AR(3) has the lowest AICC. We can test to see if this is good by seeing if the ACF and PACF of the residuals of the AR(3) models do not show any heavy autocorrelations or partial autocorrelations.



When running the ACF and PACF of the residuals of the AR(3) function we ran, we see that there aren’t any heavy autocorrelation or partial correlations with all values falling within the confidence interval lines. This tells us that the residuals have the same properties of white noise telling us that this AR(3) model is a good fit.

We also performed a Box-Ljung test giving us the following:

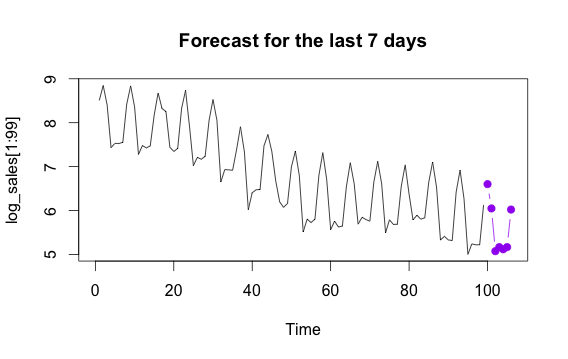
X-squared = 3.4136, df = 10, p-value = 0.97

Since the p-value is above .05 we can conclude that the values are strongly independent.

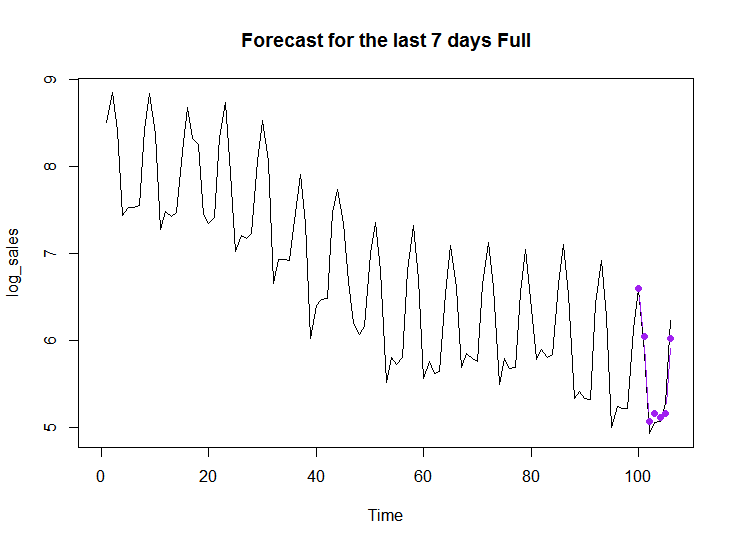
This tells us that our best model will be AR(3) we now can run a time series on this, using our AR(3) model of the transformed data to find predictions of days 100-106 using the values of days 1-99 as our predictors.

For the days we get the following:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Day 100 | Day 101 | Day 102 | Day 103 | Day 104 | Day 105 | Day 106 |
| 6.602255 | 6.048476 | 5.076130 | 5.168278 | 5.118630 | 5.168250 | 6.023004 |



Above is the true log transformed values from 1-99 with the purple dots being the expected values for 100-106.  
Below is all 106 in black with the purple being the predictions for 100-106



This prediction shows that it is very similar to the actual logarithmically transformed data with the main differences being small and after the first two days, overall though, it is very accurate and similar to the original data. This makes sense because the model uses all 106 data points in order to make this estimate so it makes sense this would be fairly accurate.

In conclusion, we find that the final model that works best is the AR(3) model that we have used above due to it having the lowest AICC of all of the AR(q) models that we tested along with having residuals that have properties of white noise. Using the forecasted data we can conclude that the amount of receipts produced for the movie will decrease as time goes on except for the spike on weekends but even then, those spikes will be lower than previous. There is also a consistent variation as time goes on showing that the data isn’t likely to move from the prediction until this movie finds itself leaving theaters.

R-Code

chicago <- read.delim("/Users/John/Downloads/chicago.txt", header=FALSE)

chicago$daynumber = 0:105

colnames(chicago) = c("Sales", "Date", "Daynumber")

library(pracma)

source('/Users/John/Downloads/trndseas.R')

source('/Users/John/Downloads/aicc.R')

#####Main Plot#####

plot(chicago$Sales~chicago$daynumber, main = "Ticket Sales of Movie Chicago Over Time", ylab = "Tickets sold", xlab = "Days from Jan 3rd 2003")

lines(chicago$Sales)

#####Transform Test#####

plot(chicago$Sales^2, type = 'l')

plot(sqrt(chicago$Sales), type = 'l')

plot(1/sqrt(chicago$Sales), type = 'l')

plot(log(chicago$Sales), type = 'l')

#log transformation looks good

#####Notes#####

#note Jan 20th was MLKjr Day, more sales due to people being out of work/school

#peaks are on Saturdays

#not stationary

#note for best modeling scheme

log\_sales = log(chicago$Sales)

#####ACF things#####

acf(log\_sales, main = "ACF for the Log of Sales over time")

pacf(log\_sales, main = "PACF for the Log of Sales over time")

acf(1/sqrt(chicago$Sales),main = "ACF for the 1/sqrt of Sales over time")

pacf(1/sqrt(chicago$Sales),main = "ACF for the 1/sqrt of Sales over time")

acf(sqrt(chicago$Sales),main = "ACF for the sqrt of Sales over time")

pacf(sqrt(chicago$Sales),main = "ACF for the sqrt of Sales over time")

acf(chicago$Sales^2,main = "ACF for the square of Sales over time")

pacf(chicago$Sales^2,main = "ACF for the square of Sales over time")

#####trndseas stuff#####

mod1 = trndseas(chicago$Sales, degtrnd = 1, seas = 7)

mod2 = trndseas(chicago$Sales, degtrnd = 2, seas = 7)

points(mod1$fit, type = 'l', col='red')

points(mod2$fit, type = 'l', col='blue')

mod1$rsq #R^2 = 0.7300727

mod2$rsq #R^2 = 0.8116906

lambdas = seq(-2,2,0.05)

mod1$rsq

lammod = trndseas(chicago$Sales, degtrnd = 2, seas = 7, lam=lambdas)

lammod$lamopt

#best lam = .1

max(mod$rsq)

plot(chicago$Sales^lammod$lamopt)

lines(chicago$Sales^lammod$lamopt)

#####boxcox stuff#####

library(MASS)

boxcox(lm(chicago$Sales~chicago$daynumber), data = chicago, lambda = seq(-0.25, 0.25, length = 10))

mod = trndseas(log\_sales,degtrnd=2,seas=7)

mod$season

plot(chicago$Sales,type='l')

points(mod$fit,type='l',col='red')

max(mod$rsq)

trend=mod$trend

season=mod$season

rsq=mod$rsq

tm=0:105

lam = seq(-2,2,by = 0.05)

fit = mod$fit

days = 1:7

par(mfrow=c(1,1))

plot(lam,rsq,type= 'l',xlab='Lambda',ylab='R-sq',main='Electric sales: R-square')

plot.ts(log\_sales,type = 'l',main='Log of Sales with trend and fit')

points(tm,fit,type='l',lty=2,col= 'red')

points(tm,trend,type='l',col='blue')

plot(days,season,type='l',ylab='Seasonals',main='Seasonals for log of sales')

plot.ts(chicago$Sales,main='Plot: Chicago Sales, trend and fit')

points(tm,exp(fit),type='l',lty=2,col = 'red')

points(tm,exp(trend),type='l',col= 'blue')

legend(0,525, c("data","trend","fit")

lty=c(1,1,2))

s.fit = rep(mod$season,length.out=106)

res = log\_sales - s.fit- trend

acf(res, main = "ACF of the Rough")

pacf(res, main = "PACF of the Rough")

mod1 = arima(res, order = c(3,0,0))

h=7

deg = 2

coef = mod$coef[1:(deg+1)]

time = (99+(1:h))/99

predmat = matrix(rep(time,deg)^rep(1:deg,each=h),nrow=h,byrow=F)

predmat = cbind(rep(1,h),predmat)

m.fc = predmat %\*% coef

s.fc = rep(mod$season,length.out=99+h)

s.fc = s.fc[-(1:99)]

fcast = predict(mod1,n.ahead=h)

x.fc = fcast$pred

y.fc = m.fc + s.fc + x.fc

par(mfrow = c(1, 1))

plot.ts(log\_sales[1:99],xlim=c(0,106), main = "Forecast for the last 7 days")

points(x=99+1:h, y=y.fc, col='purple',type='b',pch=19)

plot.ts(log\_sales,xlim=c(0,106), main = "Forecast for the last 7 days Full")

points(x=99+1:h, y=y.fc, col='purple',type='b',pch=19)