2016

Time Series: Final Project

Erica Mangino

MAS 640

Time Series Analysis of Monthly Hotel Room Occupancy

Erica Mangino

Abstract

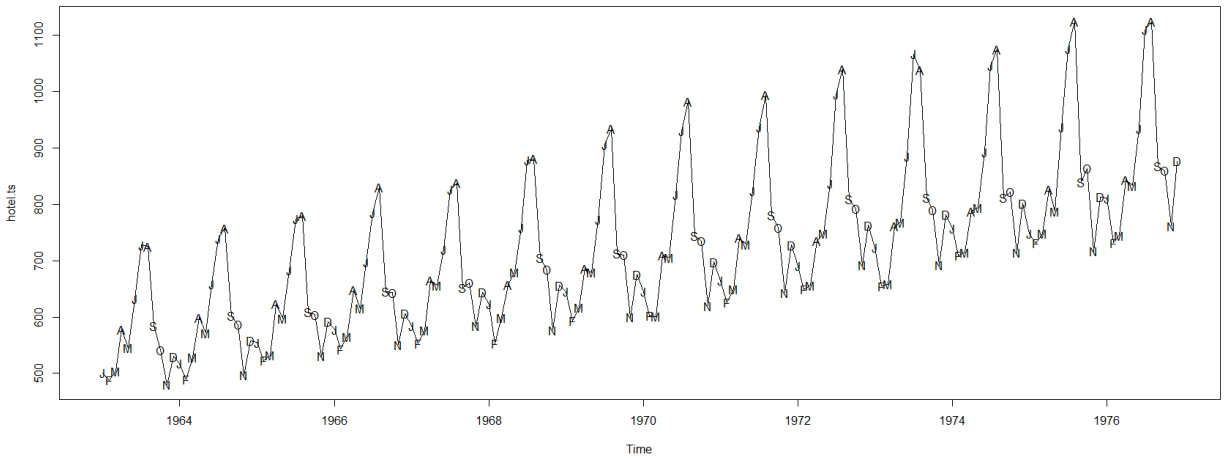
The goal of this project is to analyze the monthly hotel room occupancies and try and use time series methodologies to predict the future room occupancies in the next two years. First the data was analyzed visually as time passed. Then a seasonal ARIMA model was chosen and fitted to capture both the upward linear trend as well as the seasonality in the data. A candidate model was chosen for further investigation. After the residual tests and overfitting were done, the model was used for forecasting. Two years after the last date in the data was forecasted out. The values predicted were with 97.5 percent prediction limits.

Introduction

In the hospitality industry the livelihood of your business is solely dependent on how the number of customers and how often your services are being used. Hotel occupancy (the number of rooms booked) can not only determine how much the company will make but also staffing needs, supply and cleaning demands, room pricing, etc. It's important for the industry to adopt analytics to gain better insights for making decisions on any of these things.

In the past these types of decisions are made based on the previous week's room occupancy. With forecasting the decisions can be made sooner and budgeting can be done better. As more and more data is collected the better the forecasts will be.

The current data set looks at monthly hotel room occupancies from January 1963 to December 1976.[1] Below you will see the plot of the original data.



[1] Source: O'Donovan (1983), in file: data/odonovan25, Description: Monthly hotel occupied room av. '63-'76 B.L.Bowerman http://datamarket.com/data/list/?q=provider:tsdl

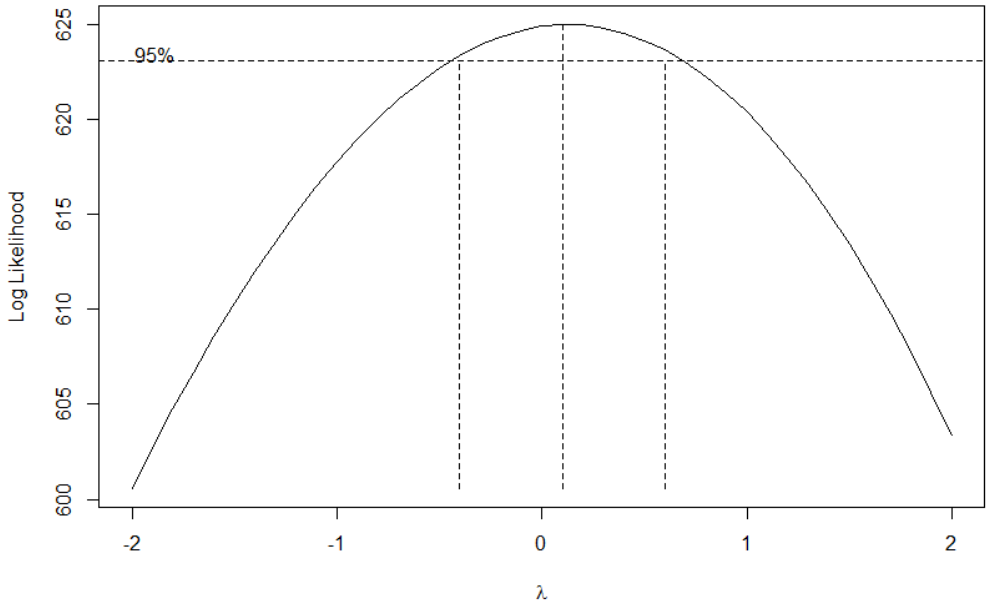
As you can see from the plot, the data looks to be seasonal as well as having an upward linear trend. Which means that the average hotel occupancy increased as the years went on but differed in their month to month trend.

Model Specifications

During the previous section the data was introduced. It was found that there was an increasing trend over time as well as a strong seasonal trend from month to month.

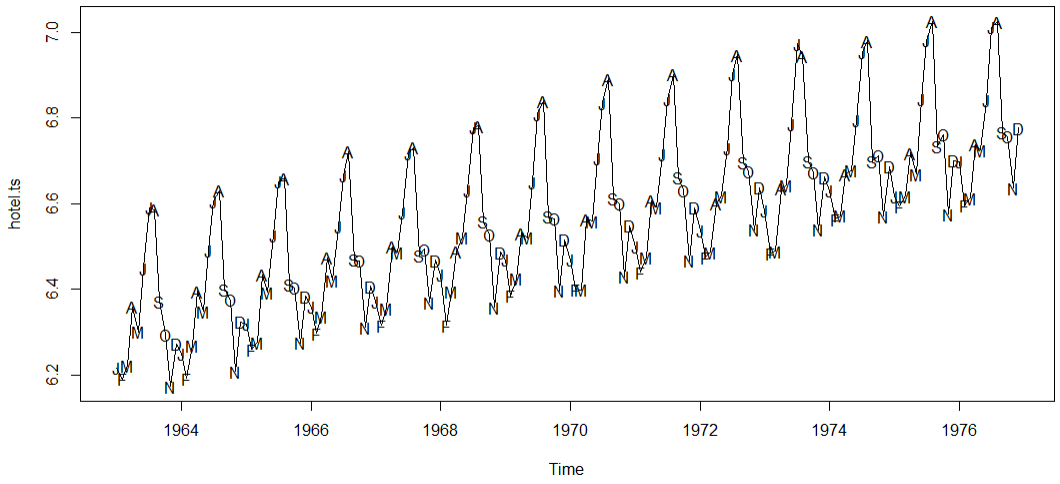
Taking a closer look at the plot July and August are the months with the highest occupancies while November and February are continuously associated with the lowest occupancies.

Before looking at models that would be relevant to the data, a Box Cox test was done to determine if any transformations needed to be made to the data. Below is the plot that will help determine which transformation to pick.

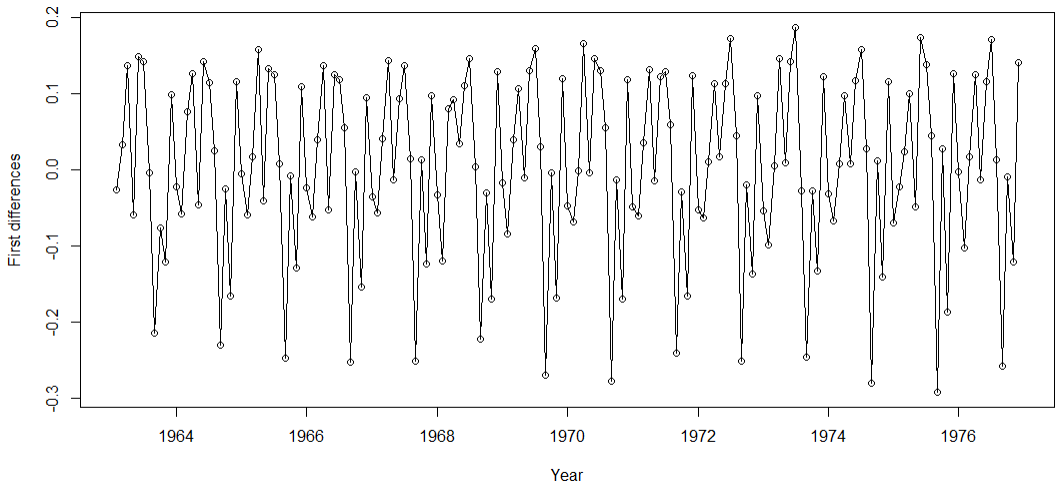


From the BoxCox plot a log transformation of the data was taken because the λ with the maximum log likelihood is closest to zero which indicates a log transformation.

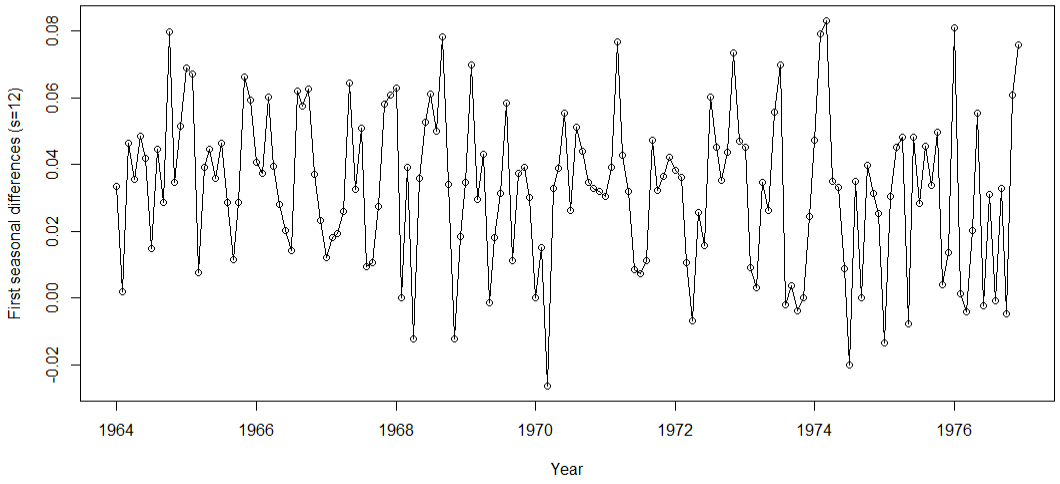
The plot below shows the data plotted on its new scale.



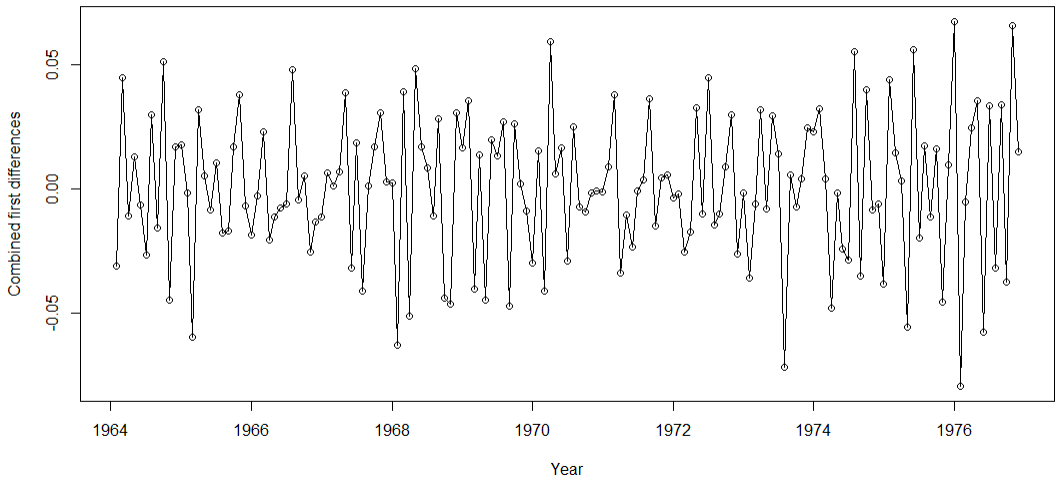
The next transformation that needed to be made would be to get rid of the trends by differencing to make the residuals look like a white noise process. This is done because by finding the model that best represents a white noise process on the current data, the better the fit of the model on the data.



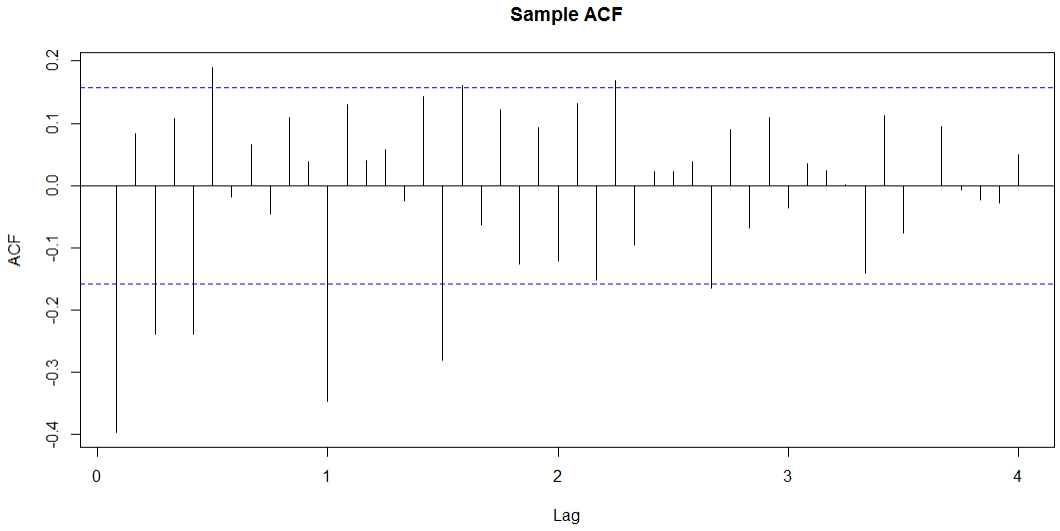
The above plot shows the first difference taken on the time series data to remove the linear trend. The below plot shows the data after the first difference was taken on the seasonal trend.



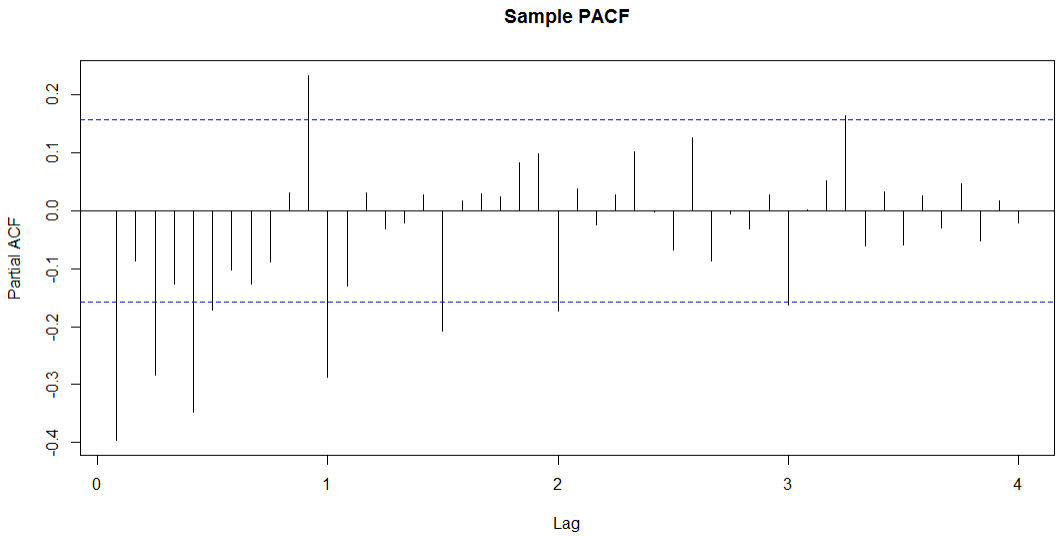
Then the differences were combined to remove both trends at the same time. The below plot shows the result of the combined differencing.



The next steps taken were to look at the ACF and PACF plots on the differenced data to determine the correct ARMA models for each the linear trend and seasonal trend.



The above sample ACF shows spikes at lags k= 1, 3, 5, 6, 12, and 18 and then continues to tail off. The below PACF shows spikes at lags k= 1, 3, 5, 11, 12, and 18 and the proceeds to cut off. Based on these graphs it was determined that the seasonal trend should not be every year but rather quarterly or every 6 months. Once it was decided that 6 months would be used as the seasonality an ARMA(1,0) x $ARMA(2,0)_6$.

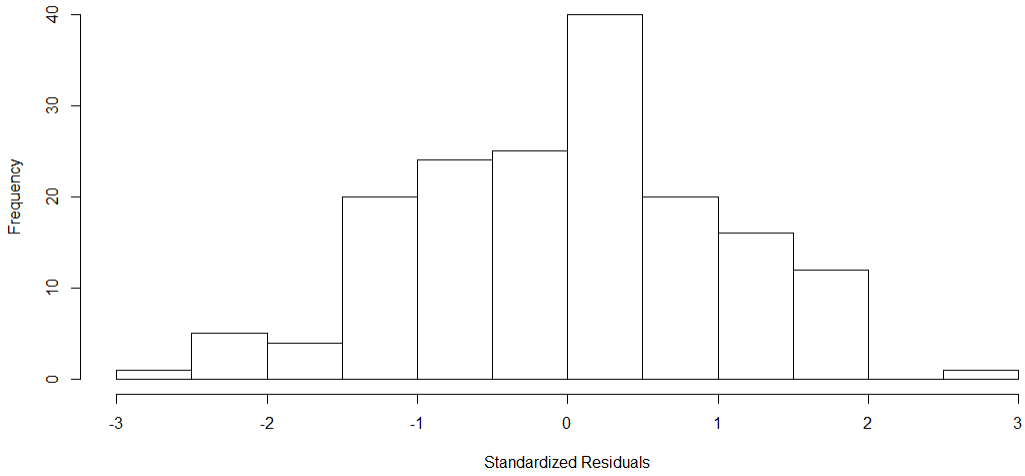


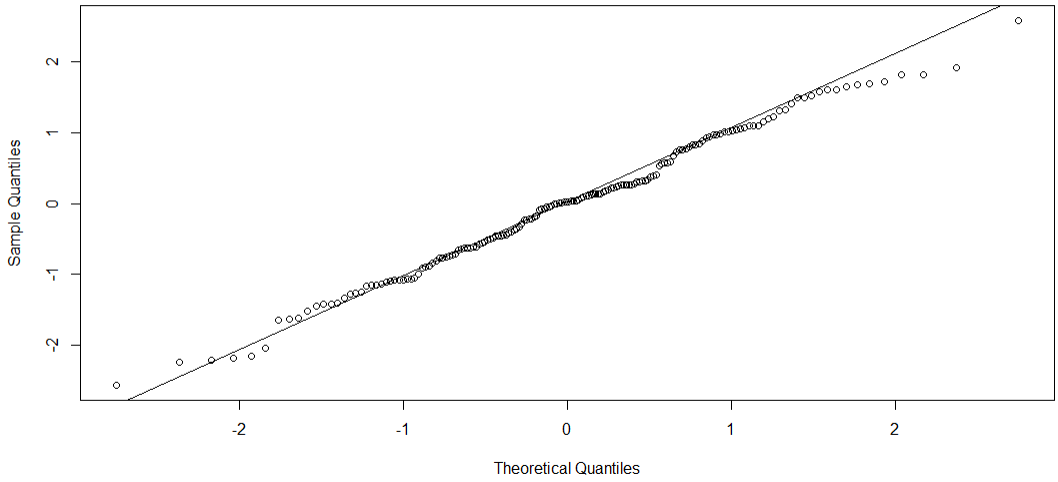
Fitting and Diagnostics

By fitting and testing the model, the ARIMA(1,1,0) x $ARIMA(2,1,0)_6$ model proved to be the best model for the data. All coefficients were significant, so we continue on with the

residual test and overfitting to do further analysis on the fit of the model. For all residual tests the process must prove to resemble that of the residuals of a white noise process. If they do not then the model will eliminated and the process will start over with a new model.

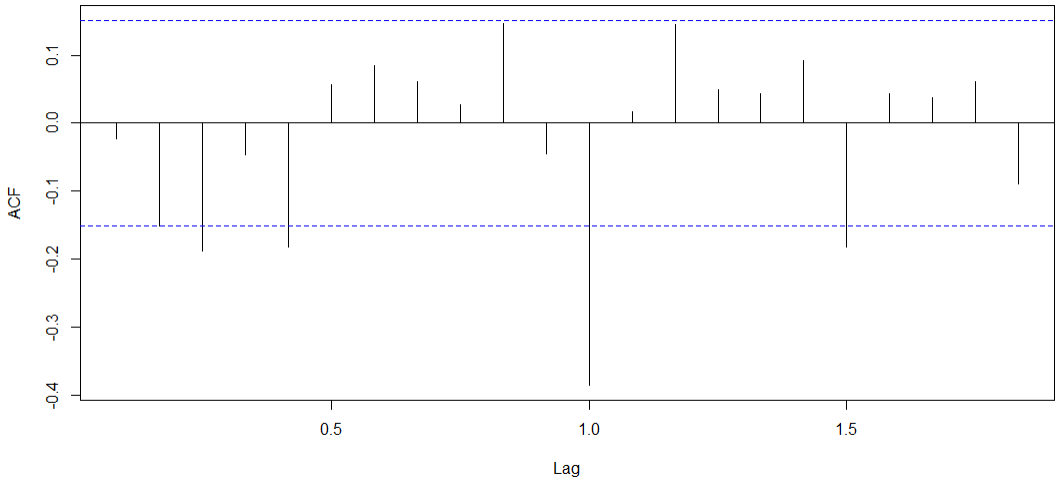
Above shows the plot of the standardized residuals of the proposed model. This will show the independence of the data. The residuals show that there is no relationship between each point and that they are centered around 0, resembling the white noise process, this passes the test for the residual plot. To further test the independence, the Runs Test was done. To determine independence the p-value of the Runs Test needs to be greater than .05, if the p-value is less than .05 the residuals are not independent. The p-value of this model is .263 which is greater than .05 so it is accepted that the residuals are independent.





The next tests done are for normality, the above histogram and qq-plot shows that the residuals are relatively normal and centered on zero. To further test normality, the Shapiro test is run. To determine if the standardized residuals are normal, the p-value of the model must be greater than .05, if it is less than .05 then the residuals are not normally distributed and the model would not represent a white noise process. The p-value of the model is .5678 which is greater than .05, meaning the residuals are normally distributed.

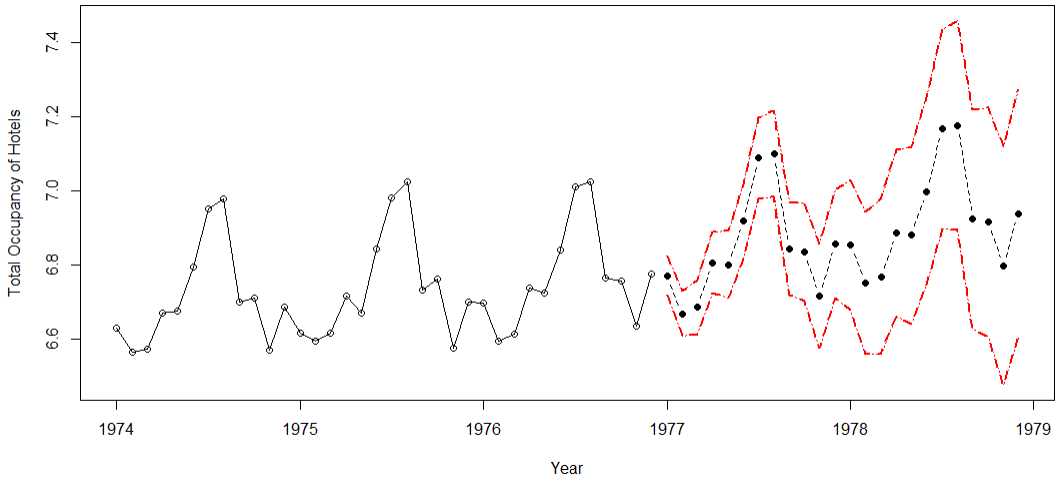
The ACF residuals shows a relatively iid normal white noise process, with only one major spike at lag k=12, we will determine through overfitting if a better model can be produced and this could be fixed.



Next overfitting was done to see if a better model could be chosen. This is done by comparing 4 different models to the previous model's fit. The results of this indicate that the best model was the model originally chosen, so it will be used to do forecasting.

Forecasting

The ARIMA(1,1,0) x ARIMA(2,1,0)$_6$ model was then fit to the data and set to predict the next 24 months starting in Jan 1977. The below plot shows the original data set starting in 1974 along with the forecasted data along with the prediction intervals indicated by the red lines.

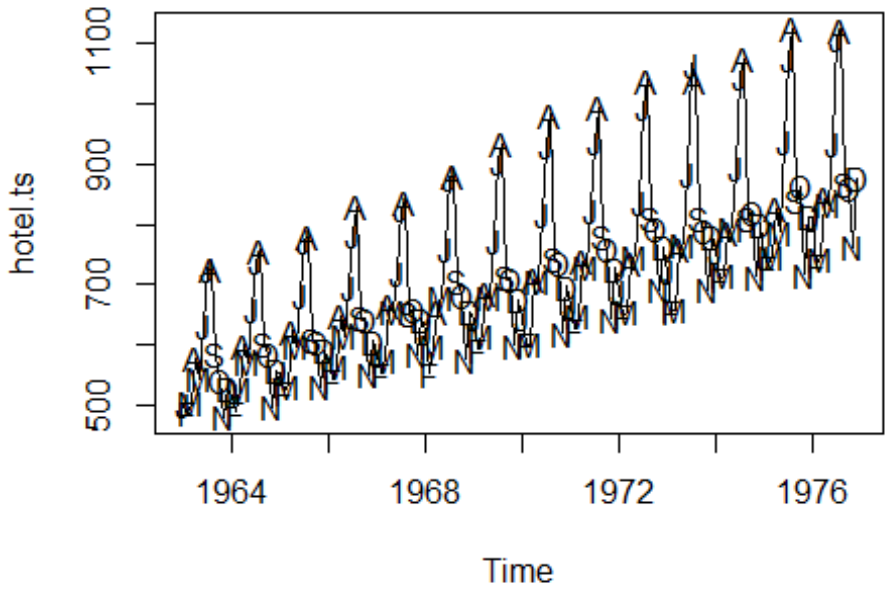


Discussion

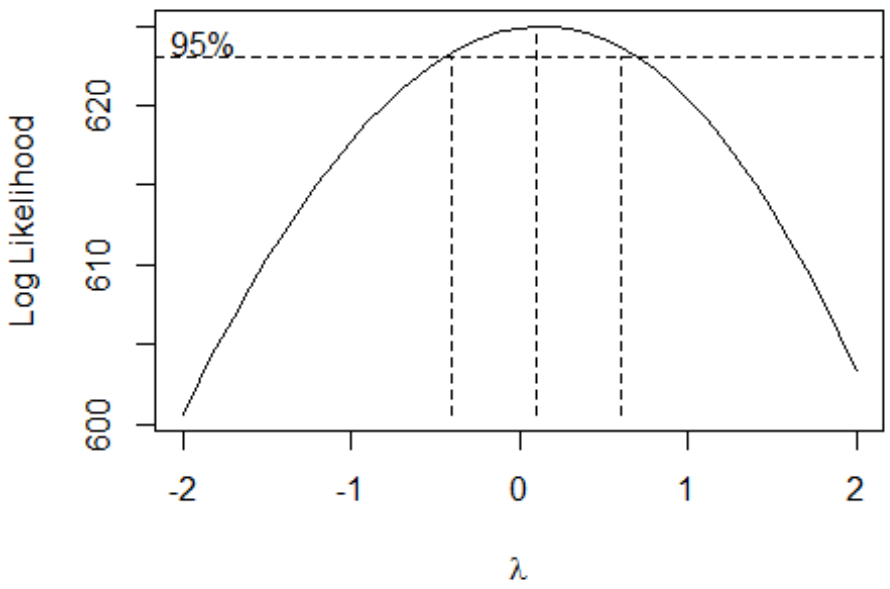
Overall the end model was generally good at predicting the seasonal nonstationary time series. It has relatively good predicting power with forecasting the linear trend as well as the seasonality of the future months. During the process in testing the ACF of the Residuals, one of the lags was outside of the confidence interval. With more analysis on candidate models to determine if there is a better model for predicting or better models should have been chosen in the early steps of the process.

Appendix

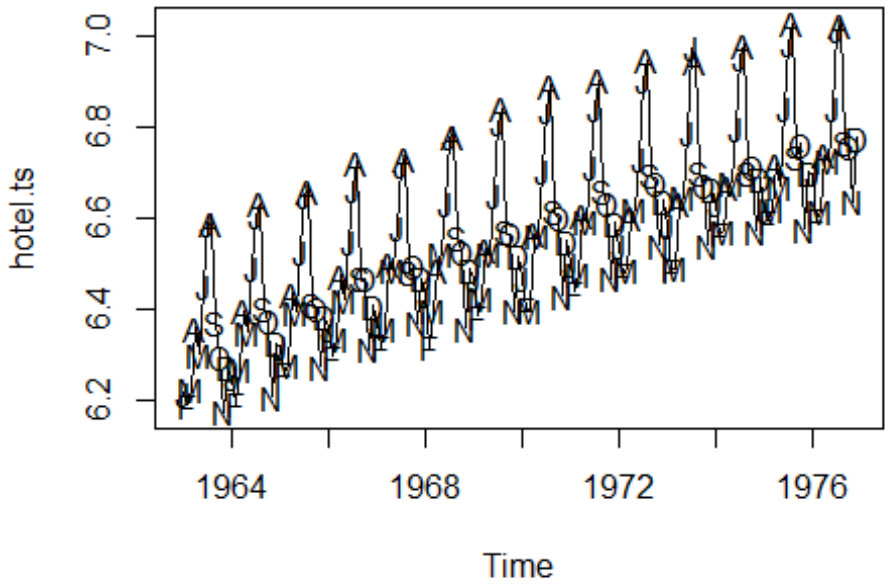
```
library(TSA)

hotel<-read.csv("monthly-hotel-occupied-room-av-6.csv")
hotel.ts<-ts(hotel$Occupied,start = c(1963, 1), frequency = 12)

plot(hotel.ts,type="l")
points(y=hotel.ts,x=time(hotel.ts),pch=as.vector(season(hotel.ts)))
```



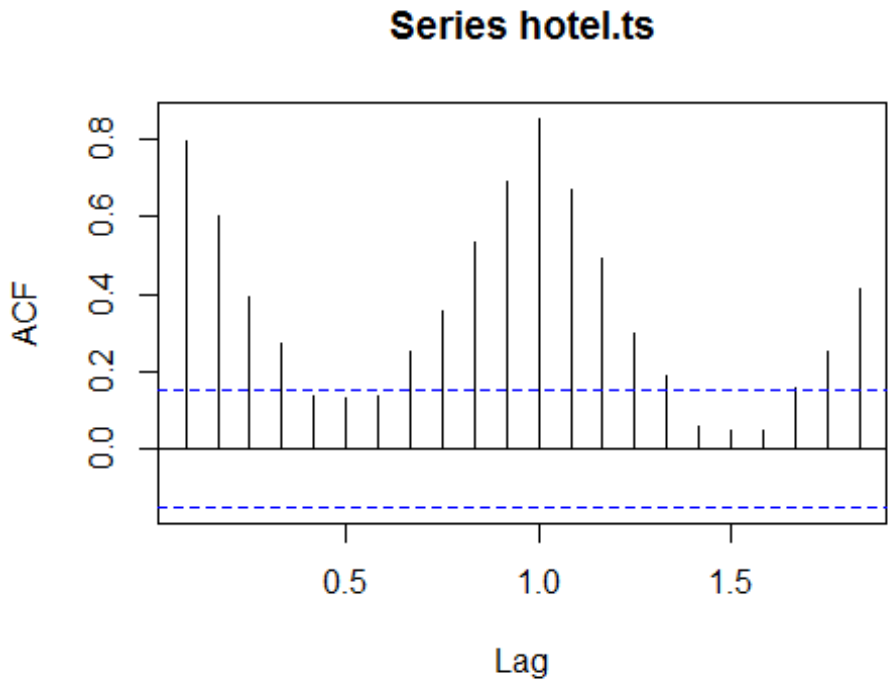

```
##Transformations

BoxCox.ar(hotel.ts, method = "burg")
```



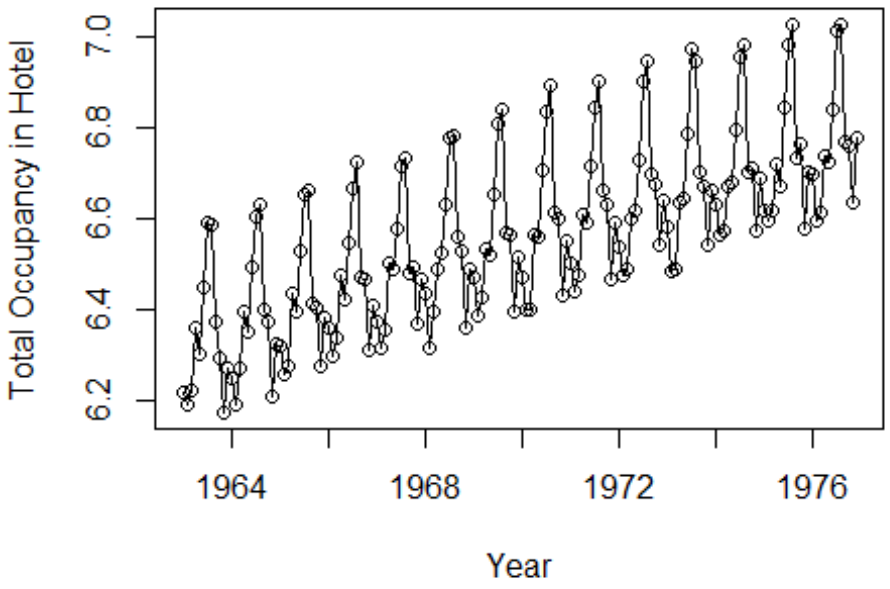

```r
hotel.ts<-log(hotel.ts)

plot(hotel.ts,type="l")
points(y=hotel.ts,x=time(hotel.ts),pch=as.vector(season(hotel.ts)))
```



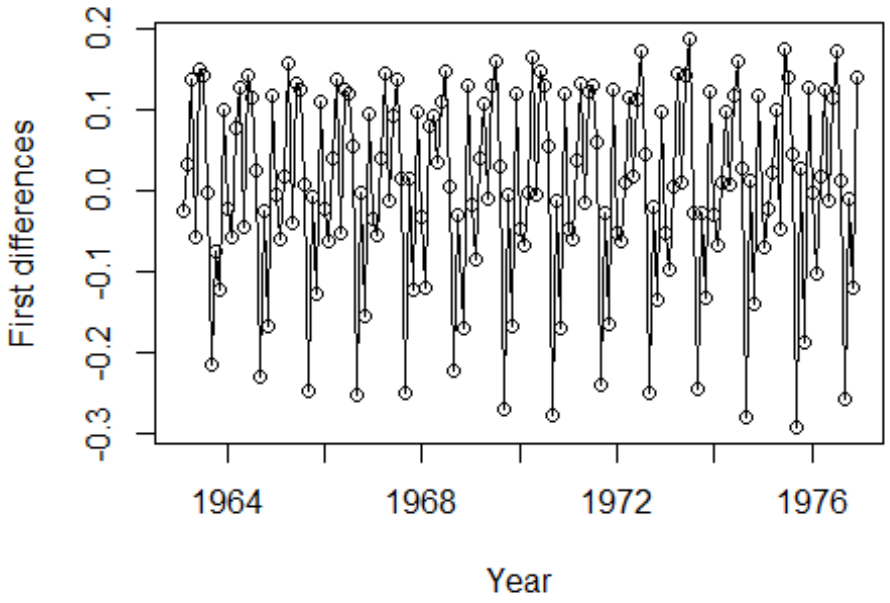
```r
acf(hotel.ts)
```



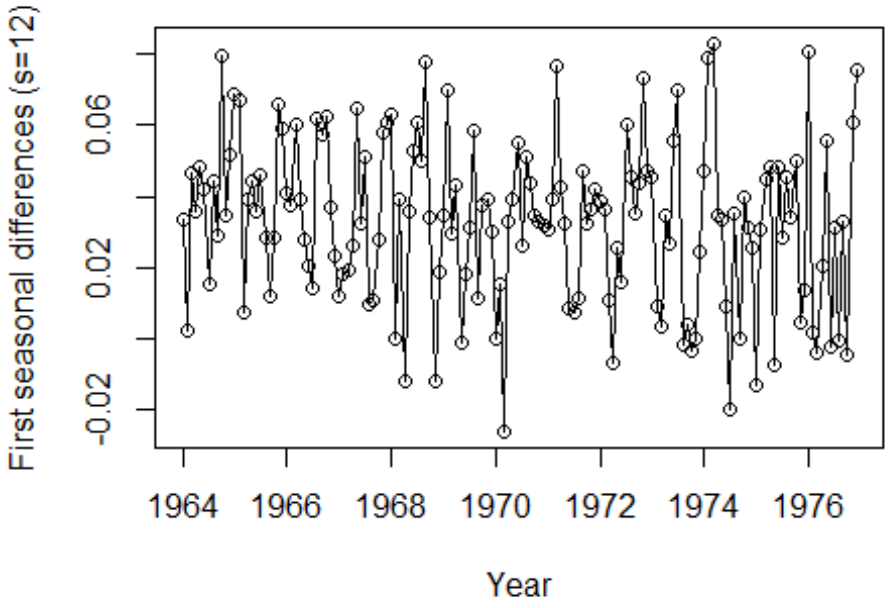

```r
##difference to get rid of the trend


plot(hotel.ts,ylab="Total Occupancy in Hotel",xlab="Year",type="o")
```



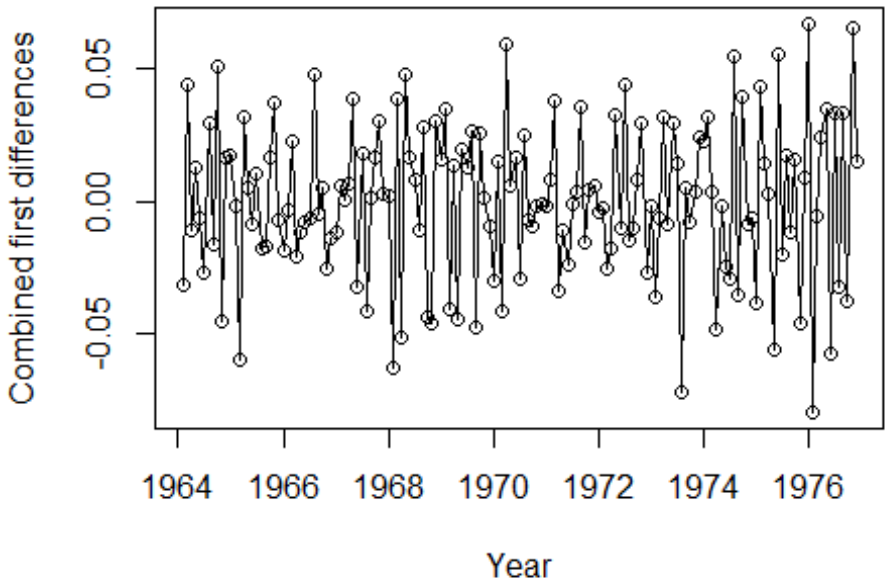

```
plot(diff(hotel.ts),ylab="First differences",xlab="Year",type="o")
```




```
plot(diff(hotel.ts,lag=12),ylab="First seasonal differences (s=12)",xlab="Yea
r",type="o")
```

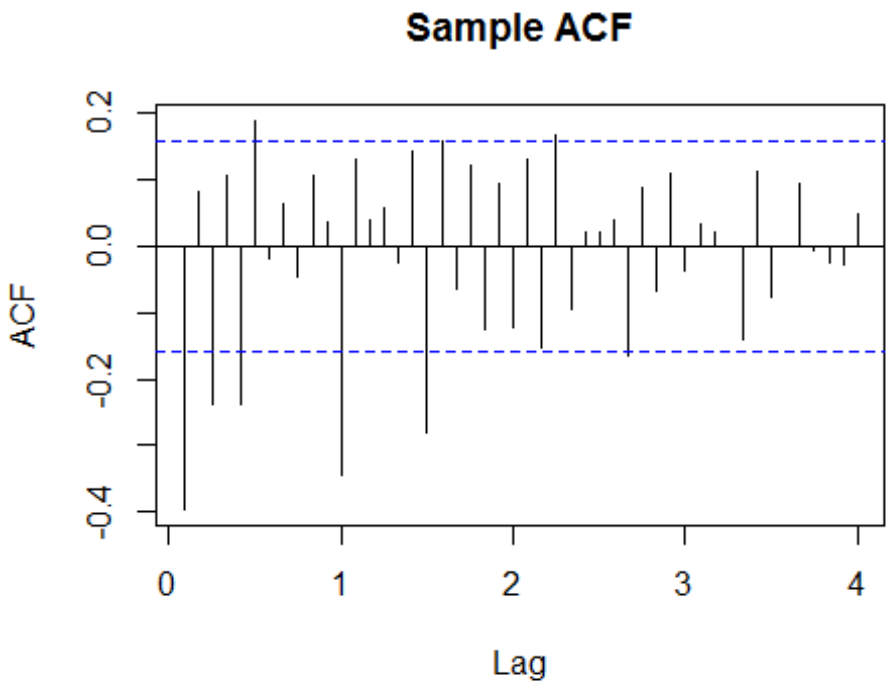



```
plot(diff(diff(hotel.ts),lag=12),ylab="Combined first differences",xlab="Year
",type="o")
```

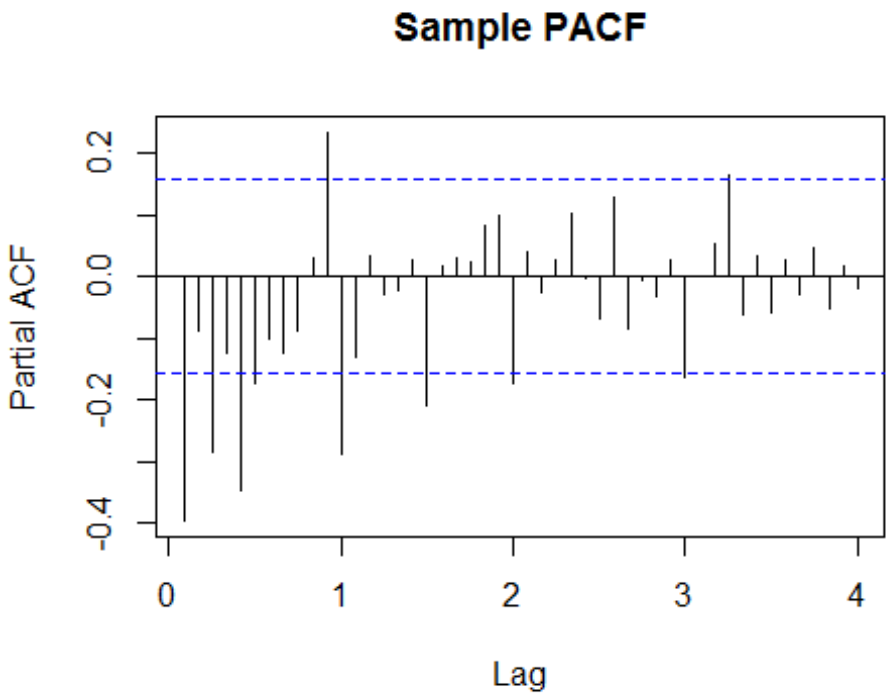



```
hotel.ts.dif<-diff(diff(hotel.ts,lag=12))
```

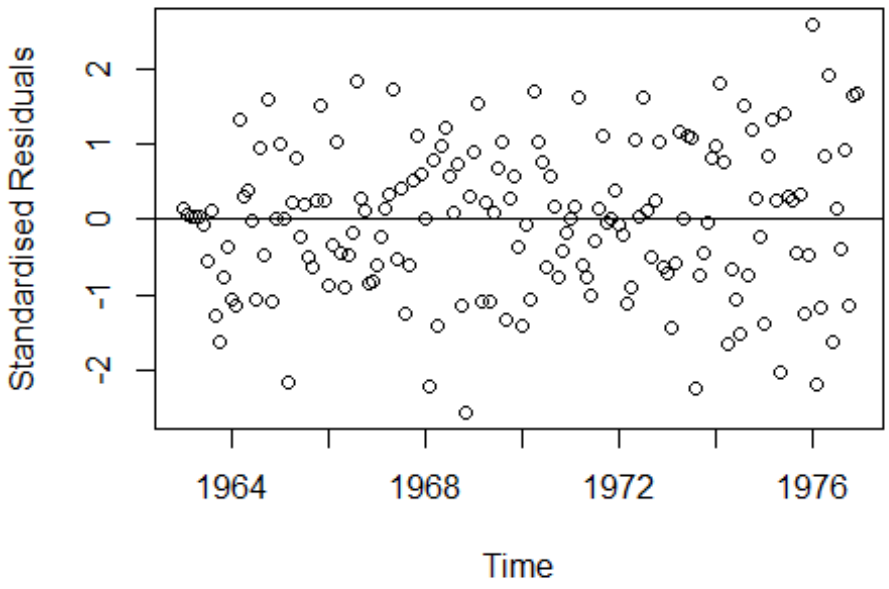
```
##ACF and PACF


acf(hotel.ts.dif,main="Sample ACF",lag.max = 48)
```



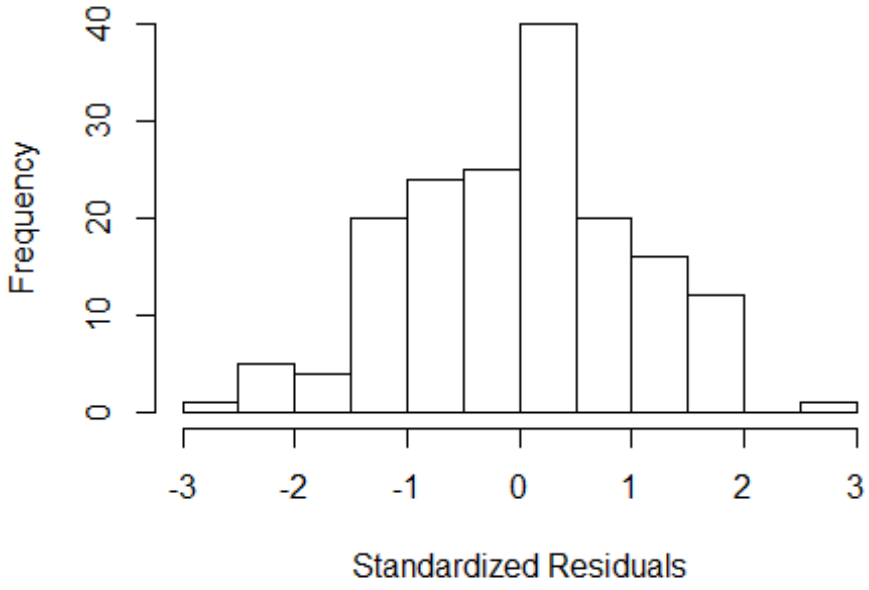

```
#spikes at 1, 3, 5, 6, 12, 18 tails off

pacf(hotel.ts.dif,main="Sample PACF",lag.max = 48)
```



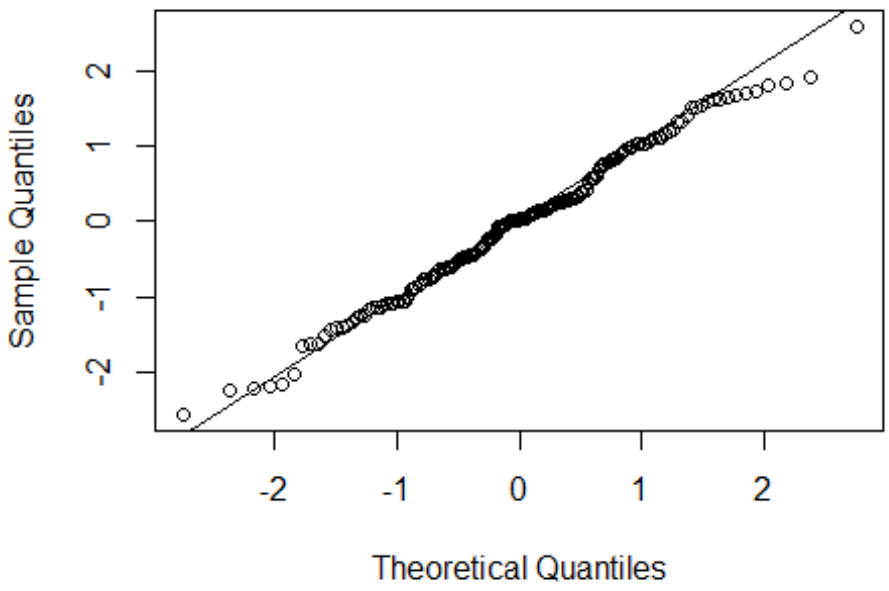

```
#spikes at 1, 3, 5, 11, 12, 18 cuts off after 18

##Fit and diagnosis

#ARIMA(1,1,0) x ARIMA(2,1,0)_{6}
hotel.arima010.arima210 = arima(hotel.ts,order=c(1,1,0),method='ML',seasonal=
list(order=c(2,1,0),period=6))

###Residual Tests

plot(rstandard(hotel.arima010.arima210),xlab="Time", ylab="Standardised Resid
uals",type="p")
abline(h=0)
```



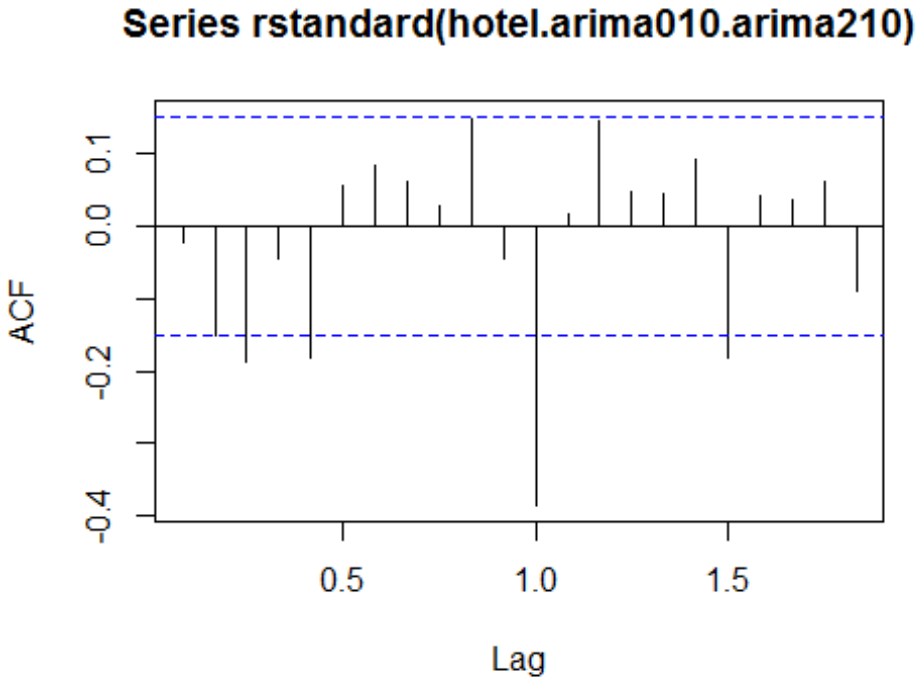

```
hist(rstandard(hotel.arima010.arima210),xlab = "Standardized Residuals",main
= "")
```




```
qqnorm(rstandard(hotel.arima010.arima210),main="")
qqline(rstandard(hotel.arima010.arima210))
```




```
acf(rstandard(hotel.arima010.arima210))
```




```
##Shapiro Test - Normality

shapiro.test(rstandard(hotel.arima010.arima210))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(hotel.arima010.arima210)
## W = 0.99277, p-value = 0.5678
```

```
##Runs Test - Independence
```

```
runs(rstandard(hotel.arima010.arima210))
```

```
## $pvalue
## [1] 0.263
##
## $observed.runs
## [1] 77
##
## $expected.runs
## [1] 84.70238
##
## $n1
## [1] 79
##
## $n2
## [1] 89
##
## $k
## [1] 0
```

```
# Overfitting
arima(hotel.ts,order=c(2,1,0),method='ML',seasonal=list(order=c(2,1,0),period
=6))
```

```
##
## Call:
## arima(x = hotel.ts, order = c(2, 1, 0), seasonal = list(order = c(2, 1, 0)
,
##     period = 6), method = "ML")
##
## Coefficients:
##           ar1      ar2     sar1    sar2
##       -0.3933  -0.0866  -0.8439  0.1478
## s.e.   0.0809   0.0816   0.0817  0.0822
##
## sigma^2 estimated as 0.0006967:  log likelihood = 344.66,  aic = -681.31
```

```
arima(hotel.ts,order=c(1,1,1),method='ML',seasonal=list(order=c(2,1,0),period
=6))
```

```
##
## Call:
## arima(x = hotel.ts, order = c(1, 1, 1), seasonal = list(order = c(2, 1, 0)
,
##     period = 6), method = "ML")
```

```
##
## Coefficients:
##          ar1      ma1     sar1    sar2
##       0.2564  -0.9957  -0.8562  0.1377
## s.e.  0.0830   0.0503   0.0845  0.0848
##
## sigma^2 estimated as 0.0005101:  log likelihood = 366.39,  aic = -724.78
```

```
arima(hotel.ts,order=c(1,1,0),method='ML',seasonal=list(order=c(3,1,0),period
=6))
```

```
##
## Call:
## arima(x = hotel.ts, order = c(1, 1, 0), seasonal = list(order = c(3, 1, 0)
,
##     period = 6), method = "ML")
##
## Coefficients:
##           ar1     sar1     sar2     sar3
##       -0.3984  -0.7767  -0.2862  -0.5052
## s.e.   0.0754   0.0737   0.0967   0.0744
##
## sigma^2 estimated as 0.0005377:  log likelihood = 362.98,  aic = -717.96
```

```
arima(hotel.ts,order=c(1,1,0),method='ML',seasonal=list(order=c(2,1,1),period
=6))
```

```
##
## Call:
## arima(x = hotel.ts, order = c(1, 1, 0), seasonal = list(order = c(2, 1, 1)
,
##     period = 6), method = "ML")
##
## Coefficients:
##           ar1     sar1    sar2     sma1
##       -0.3768  -0.3573  0.6225  -0.9414
## s.e.   0.0749   0.0901  0.0902   0.0731
##
## sigma^2 estimated as 0.0006173:  log likelihood = 351.67,  aic = -695.34
```

```
# Forecasting
# Full data set: 1/63-12/76
hotel.arima010.arima210.fit = arima(hotel.ts,order=c(1,1,0),method='ML',seaso
nal=list(order=c(2,1,0),period=6))
# MMSE forecasts
hotel.arima010.arima210.predict <- predict(hotel.arima010.arima210.fit,n.ahea
d=24)
round(hotel.arima010.arima210.predict$pred,3)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 1977 6.772 6.669 6.688 6.806 6.802 6.919 7.089 7.101 6.845 6.836 6.717
## 1978 6.854 6.752 6.769 6.887 6.881 6.999 7.167 7.178 6.924 6.916 6.798
##        Dec
```

```
## 1977 6.858
## 1978 6.939
```

```
round(hotel.arima010.arima210.predict$se,3)
```

```
##        Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
## 1977 0.027 0.031 0.037 0.042 0.046 0.050 0.056 0.060 0.064 0.068 0.071
## 1978 0.090 0.098 0.107 0.115 0.122 0.129 0.137 0.145 0.152 0.158 0.165
##        Dec
## 1977 0.075
## 1978 0.171
```

```
# Display prediction intervals (24 months ahead)
year.temp = c(1977,1977.083,1977.166,1977.250,1977.333,1977.416,1977.500,1977
.583,1977.666,1977.750,1977.833,1977.916)
year.temp.2 = c(year.temp,year.temp+1)

# Compute prediction intervals
lower.pi<-hotel.arima010.arima210.predict$pred-qnorm(0.975,0,1)*hotel.arima01
0.arima210.predict$se
upper.pi<-hotel.arima010.arima210.predict$pred+qnorm(0.975,0,1)*hotel.arima01
0.arima210.predict$se
data.frame(Month=year.temp.2,lower.pi,upper.pi)
```

```
##       Month lower.pi upper.pi
## 1  1977.000 6.719931 6.823885
## 2  1977.083 6.607264 6.730602
## 3  1977.166 6.614282 6.761271
## 4  1977.250 6.723947 6.888987
## 5  1977.333 6.710499 6.892544
## 6  1977.416 6.820743 7.018086
## 7  1977.500 6.980239 7.197986
## 8  1977.583 6.983454 7.217681
## 9  1977.666 6.719551 6.969895
## 10 1977.750 6.703113 6.968349
## 11 1977.833 6.577212 6.856632
## 12 1977.916 6.711175 7.004064
## 13 1978.000 6.678850 7.029730
## 14 1978.083 6.560351 6.943986
## 15 1978.166 6.559383 6.978547
## 16 1978.250 6.661535 7.111605
## 17 1978.333 6.641127 7.120733
## 18 1978.416 6.745437 7.252650
## 19 1978.500 6.897976 7.436820
## 20 1978.583 6.894575 7.461374
## 21 1978.666 6.627192 7.221287
## 22 1978.750 6.606226 7.226190
## 23 1978.833 6.475859 7.120733
## 24 1978.916 6.604412 7.273241
```

```
# Original series starts at Month = Jan 1963
# Note: Argument n1=c(1974,1) starts plot at Time = Jan 1974
# Note: Argument pch=16 produces a small black circle (MMSE prediction)
```

```r
plot(hotel.arima010.arima210.fit,n.ahead=24,col='red',type='b',pch=16,n1=c(19
74,1),ylab="Total Occupancy of Hotels",xlab="Year")
# Put prediction interval lines on plot (darker than default)
lines(y=lower.pi,x=year.temp.2,lwd=2,col="red",lty="dashed")
lines(y=upper.pi,x=year.temp.2,lwd=2,col="red",lty="dashed")
```