Candy Production Data Time Series Analysis

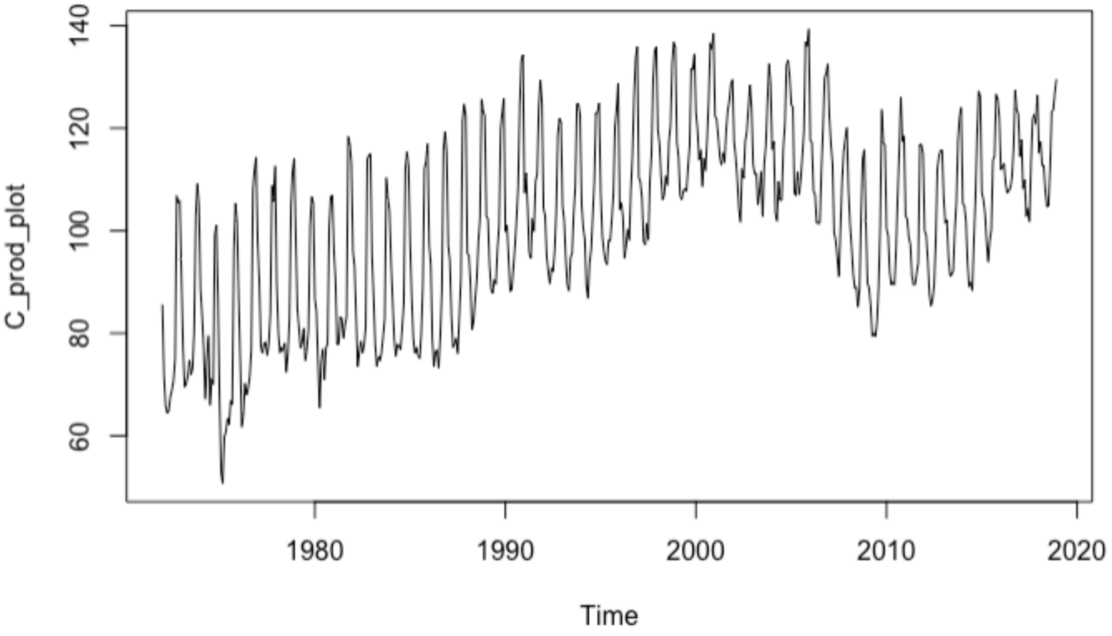
by Keshav Khanna

Summary

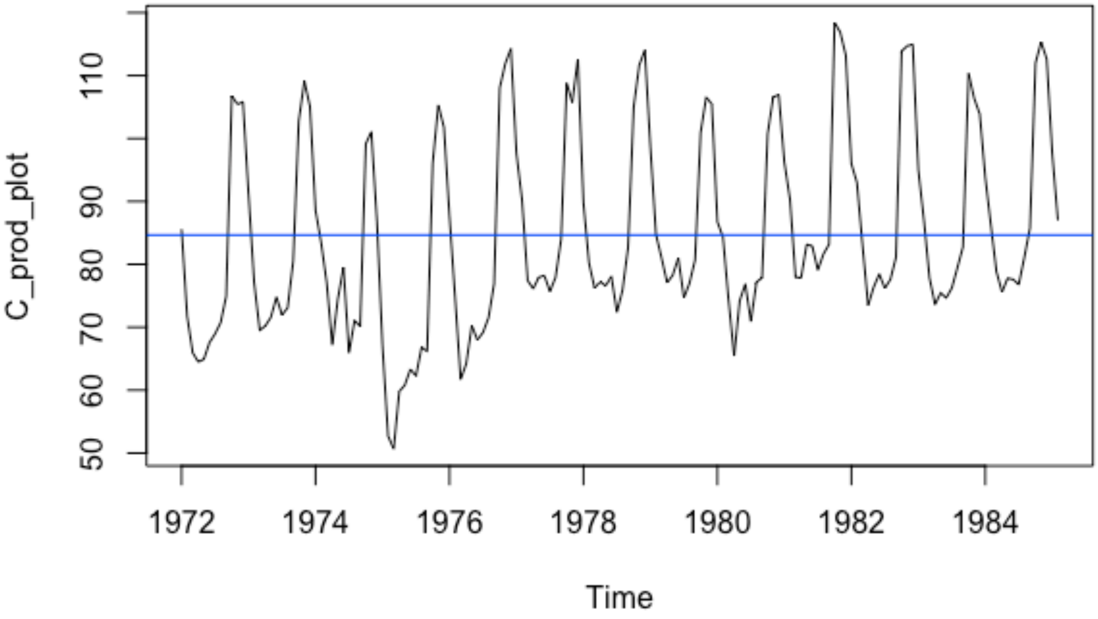
This project aims at predicting monthly candy production for 36 months starting from January 1982 to observe if candy production follows the same seasonal pattern as the past values where production increases dramatically for the months of October till December and falls thereafter. Time series techniques like differencing, parsimony analysis, and diagnostic checking have been used before forecasting the data. At the end, it is observed that test set values are within the confidence intervals and are very close to forecasted values which shows that findings are coherent with seasonal pattern observed in the dataset.

Introduction

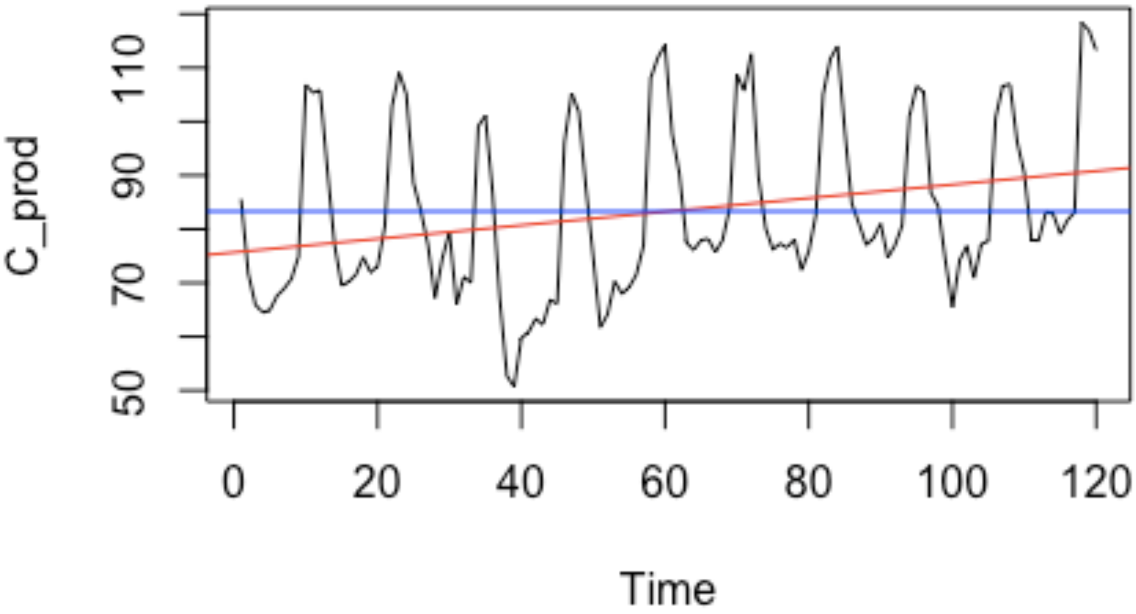
The dataset used for the time series analysis is US candy production by month, as a percent of 2012 production. This dataset has been collected from Kaggle and is provided by Rachael Tatman. Moreover, R-Studio software has been used to analyze this time series data. As observed in the dataset, the candy production levels increase dramatically from months of October till January, depicting increase in supply to match with increased demand, which could be due to the festivals like Thanksgiving, Halloween, Christmas, and New Year when candy consumption is a lot. The production falls significantly after these months. This pattern makes the time series highly seasonal. The aim of time series analysis is predicting similar production levels for the months from January 1982 to January 1985. Various time series techniques like differencing, parsimony analysis, and diagnostic checking have been used before forecasting the data. At the end, positive results are achieved as the predicted values lie very close to the test values and test values were in the confidence intervals.



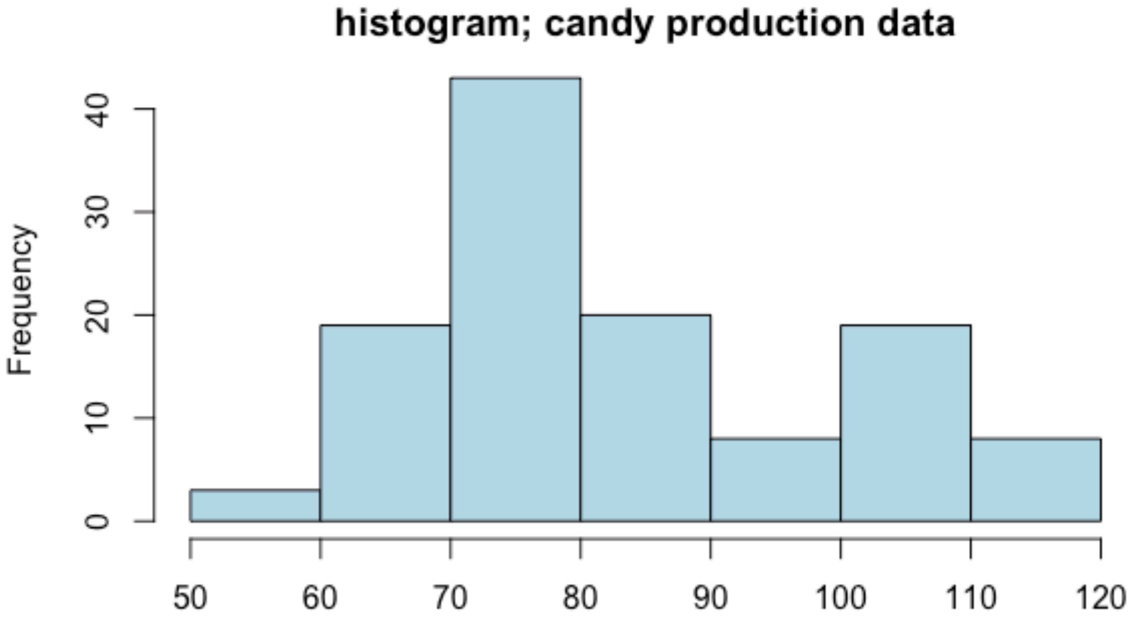
The original US Candy Production by month is from January 1972 to August 2017. It has been trimmed to include values until January 1985, which gives total of 156 data points.

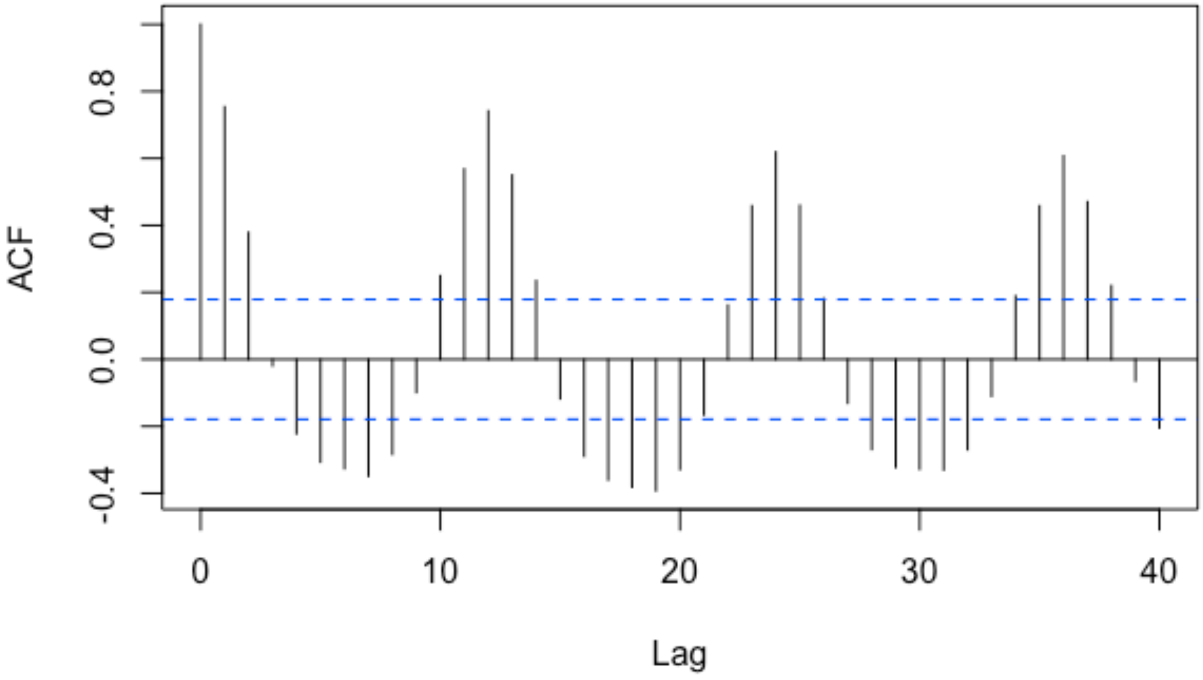


After this, the data has been partitioned into a training and a test set. Values from 1 to 120 are included in the training set (U_t) and values from 121 to 156 are included in the test set. Next, this training set is plotted and it is seen that this data is highly non-stationary. It has a linear trend and seasonality. It is also easy to see that the mean grows because of the increasing trend and the variance remains more or less constant.

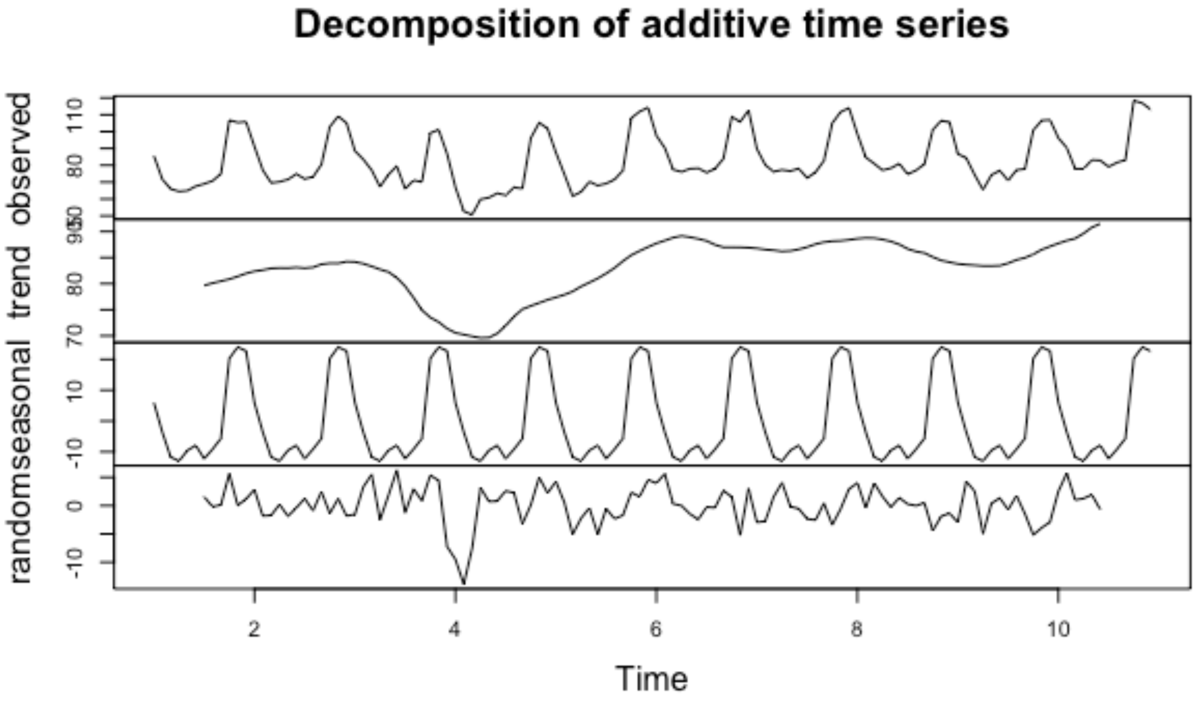


To justify constant variance, we take plot the histogram of our training values and find that the histogram is more or less symmetric. Hence no transformation is required. It is also observed that ACF of our training values remain large and period which shows linear trend and seasonality.

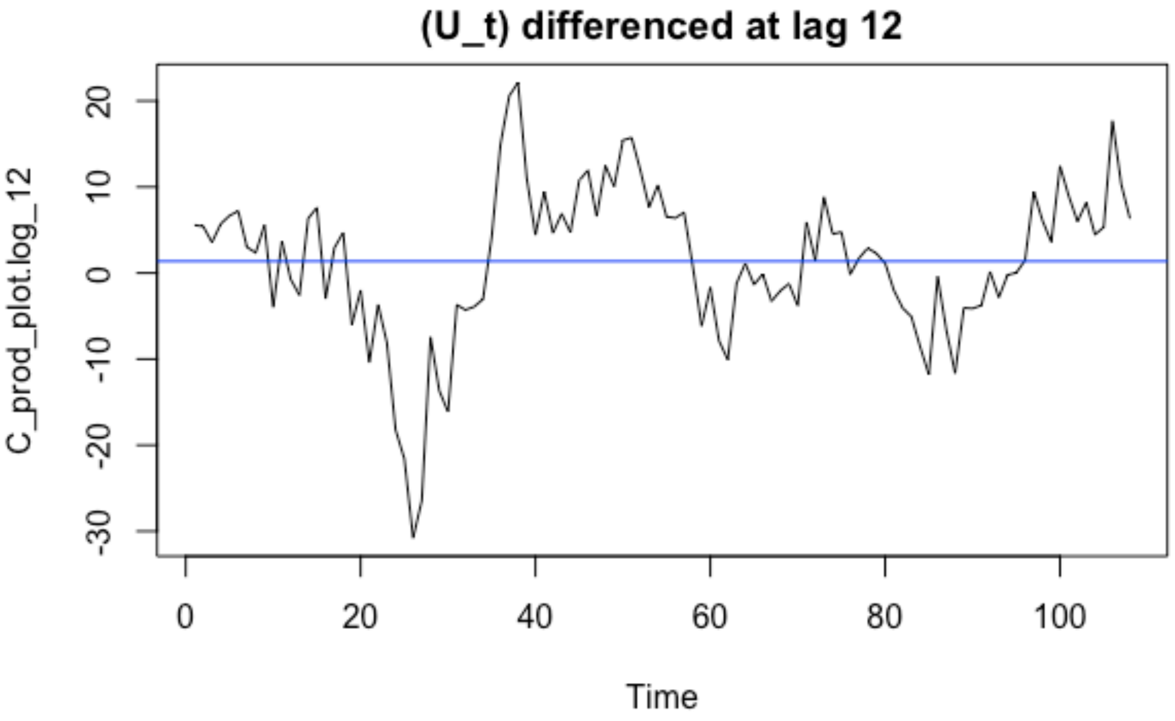

histogram; candy production data
Frequency
0
10
20
30
40
50
60
70
80
90
100
110
120


ACF
-0.4
0.0
0.4
0.8
0
10
20
30
40
Lag

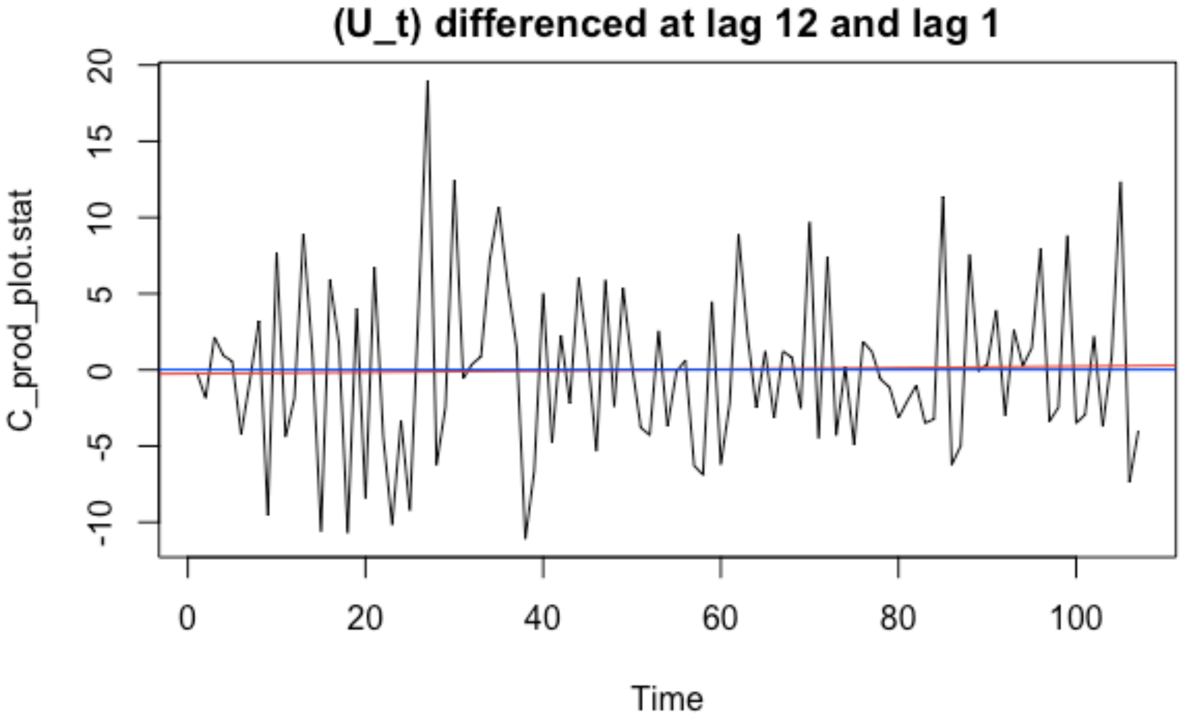
Moreover, decomposition of our data (U_t) shows seasonality and almost linear trend.



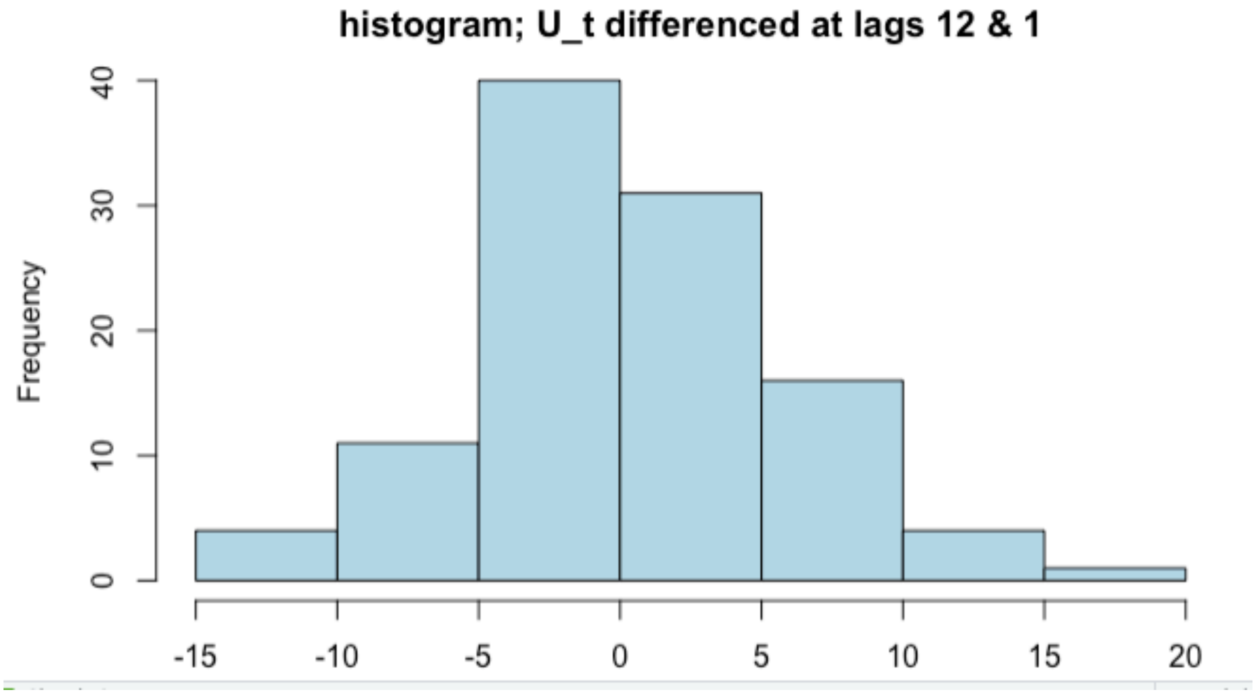
Our next step is removing the seasonality by differencing U_t at lag 12. Next, U_t is differenced at lag 1 to remove trend. Our data is stationary now and looks symmetric and almost Gaussian that fits normal curve well.

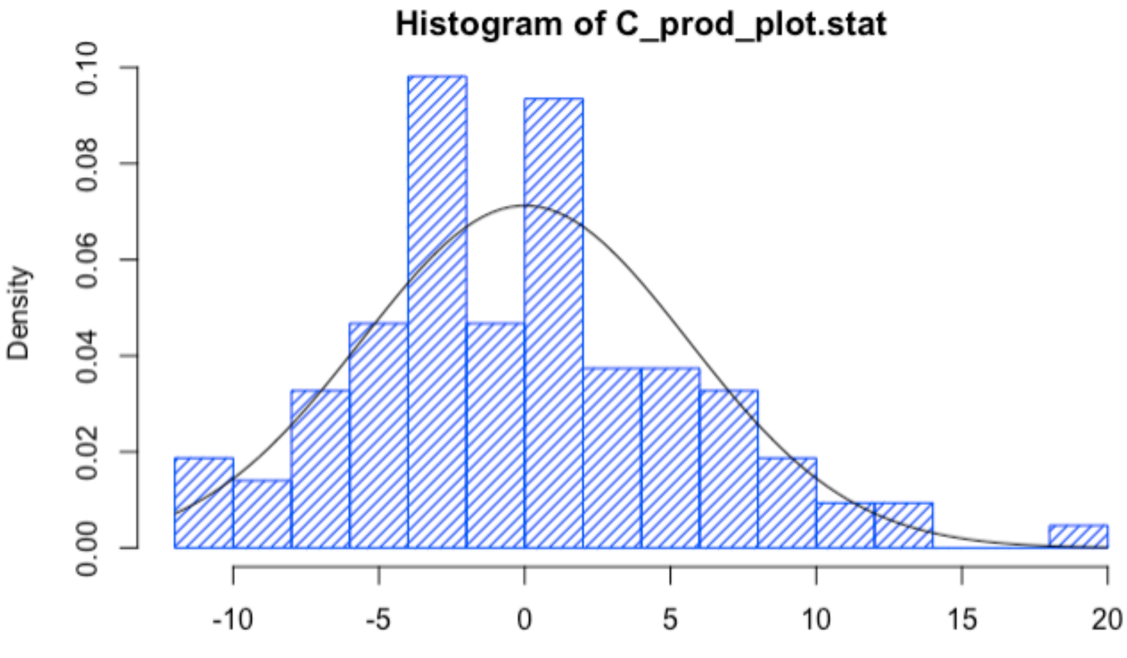


(U_t) differenced at lag 12

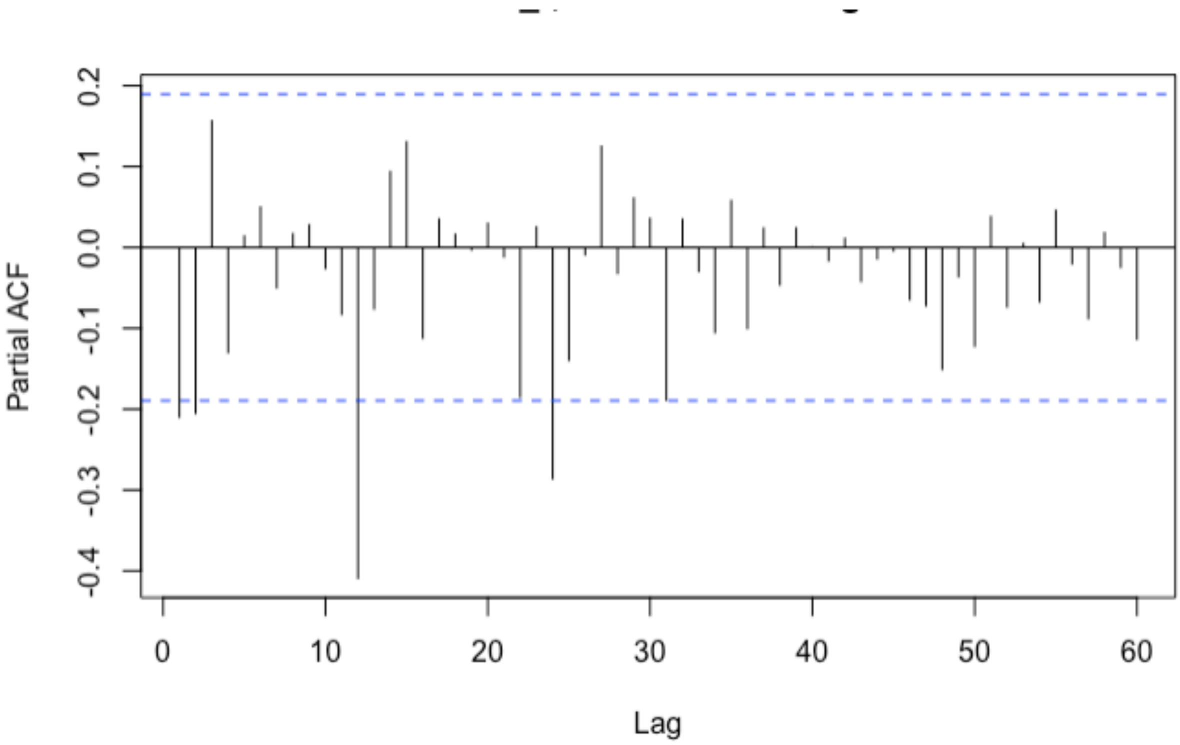


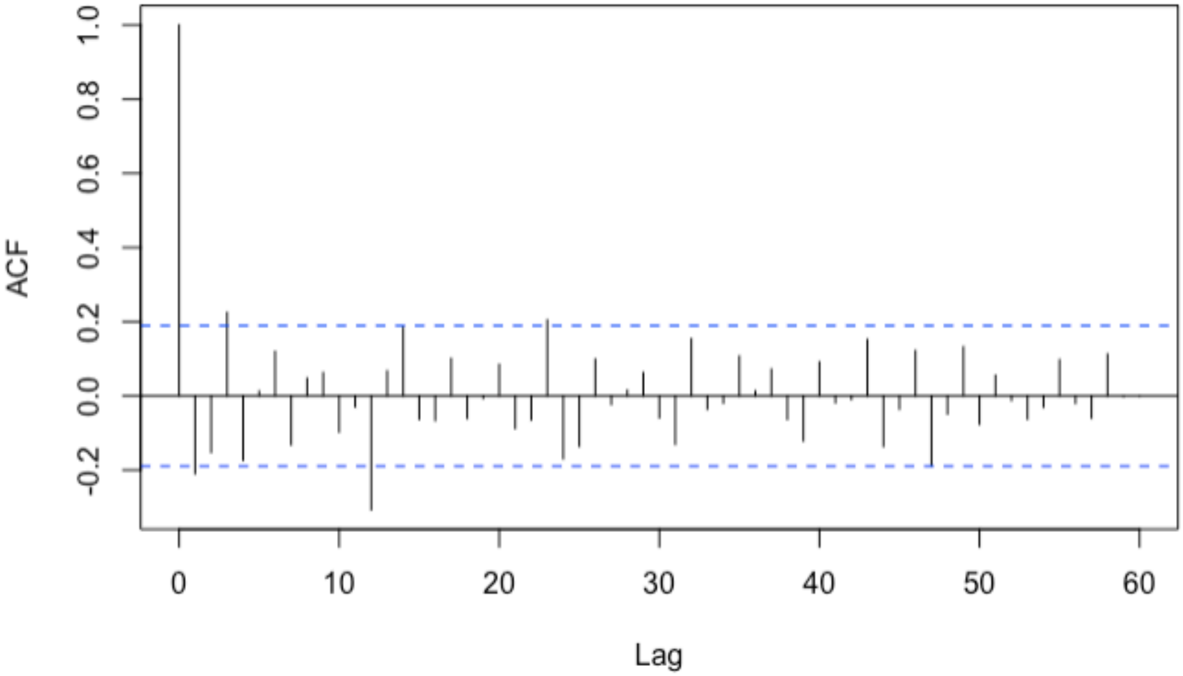
(U_t) differenced at lag 12 and lag 1





We will consider only SARIMA models as our candidate models since we differenced at lag 12 and lag 1. D will be 1 since our time series was differenced once at lag 12 to remove seasonality and s =12 since our original data was seasonal with period 12. d will be 1 since our time series was differenced once at lag 1 to remove trend . The choice of specific P’s, Q’s, q’s, p’s will be done by examining ACF and PACF of our differenced data. As noticed in the graph of PACF, PACF is outside the confidence intervals at lag 12 and lag 24, so the choice of P will be 1 or 2. Before lag 12, PACF is outside the confidence intervals at lag 1 and lag 2, so the choice of p will be 1 or 2. ACF is outside the confidence intervals at lag 12 only so the choice of Q will be 1. Before lag 12, ACF is outside the confidence intervals at lag 1 and lag 3. So the choice of q will be 1 or 3.

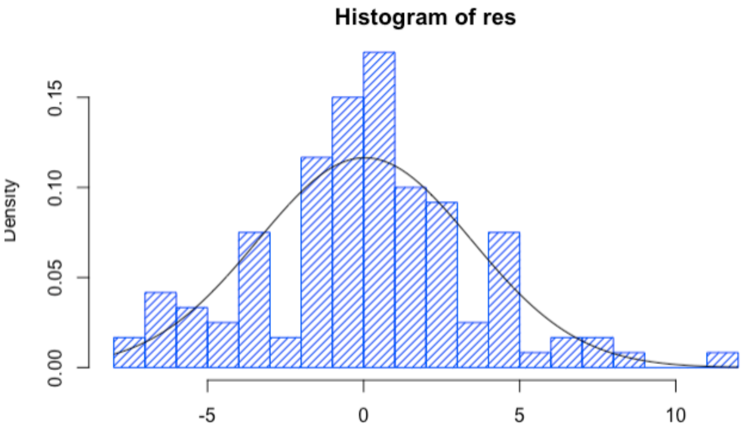


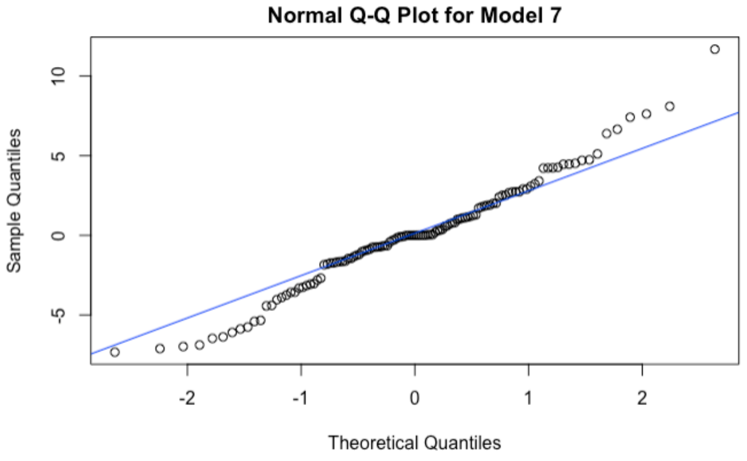


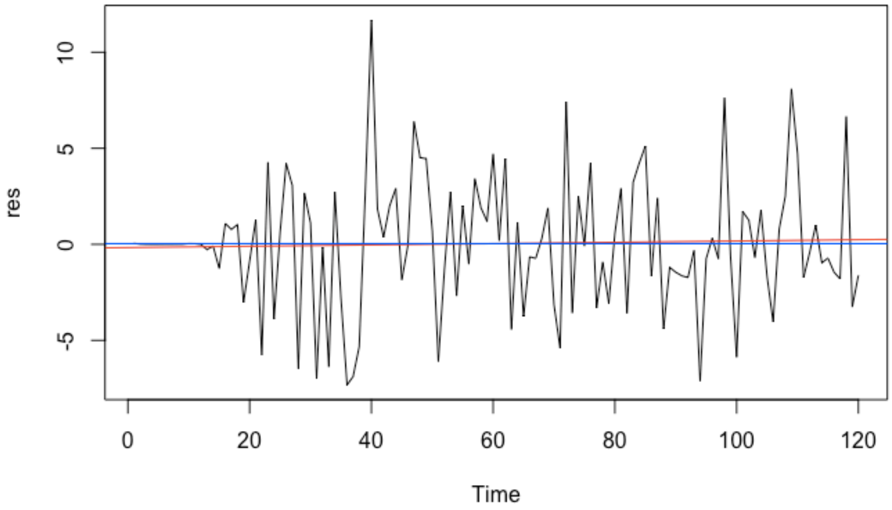
Thus, our list of candidate models will have parameters p=1,2;q=1,3;P=1,2;Q=1;D=1;d=1;s=12. After calculating the AICc for all our models, we will select the models with the lowest AICc by using the principle of parsimony. We find that model with parameters q=3; p=2; Q=1 ; P=1 has the lowest AICc and is named as model 4. Also, model with parameters q=1; p=2; Q=1 ; P=2 has the second lowest AICc and is named as model 7. Therefore we select these models to check for their invertibility and stationarity. We find that model 7 is stationary and invertible because its roots for its seasonal/non-seasonal MA and AR part are outside the unit circle. Also, model 4 is not stationary and invertible because one of its roots for its non-seasonal AR part is outside the unit circle. Hence we will only select model 7 for diagnostic checking.

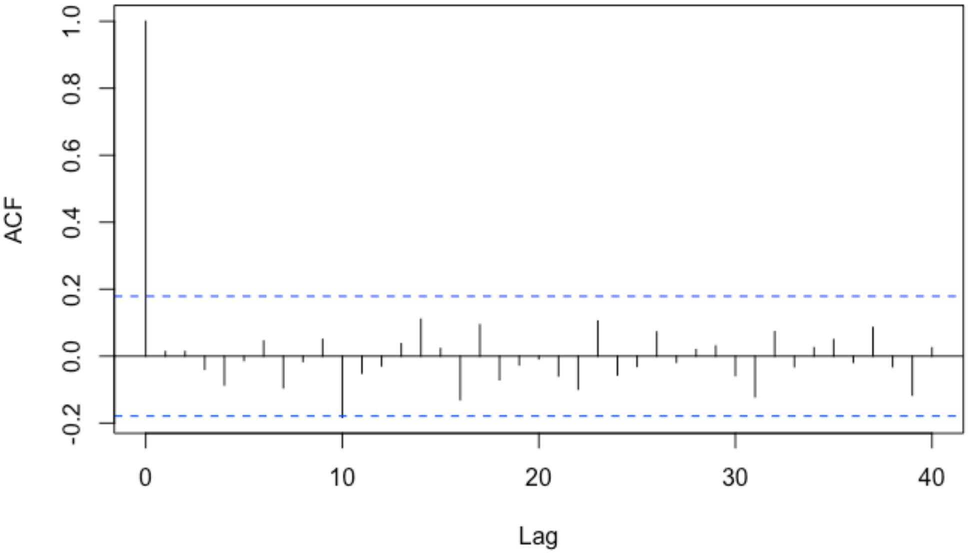
After running diagnostic checks, we find that ACF and PACF of residuals are within the confidence intervals and can be counted as zeros. Moreover there is no visible trend, no change of variance, and no seasonality. The histogram looks Gaussian and normal Q-Q also looks fine as majority of the sample quantiles lie on or very close to the line. Next, we find that Shapiro-Wilk test of normality, McLeod-Li Test, Ljung-Box Test, and Box Pierce test have a p value greater than 0.05, which means all our diagnostic tests pass. Hence model 7 will become our fitted model that will be used for forecasting. It can be algebraically written as:

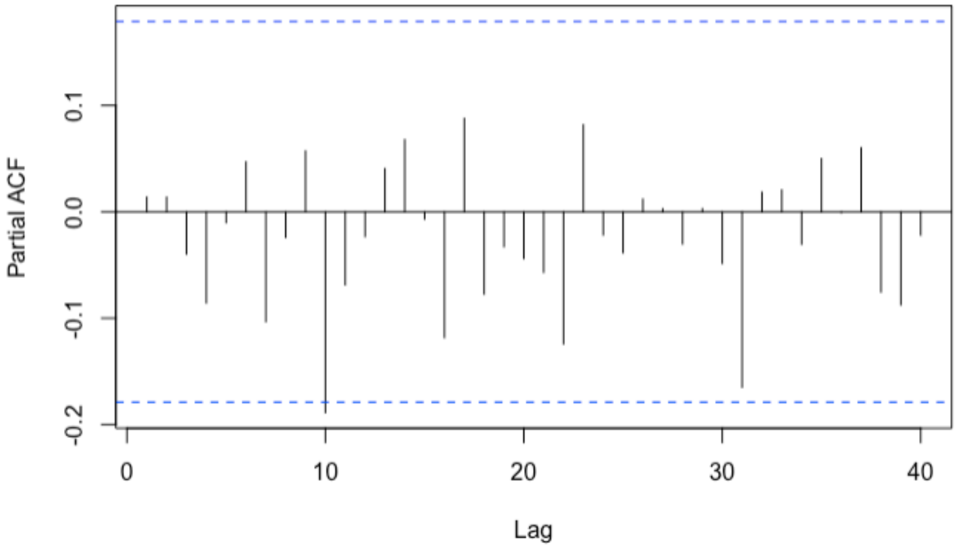
(1+0.8717(B)+0.3806(B^2))(1-0.0241(B^12)+0.1995(B^24)) X_t = (1+0.6558B) (1-0.97(B^12)) Z_t , where Z_t~WN(0,13.05)

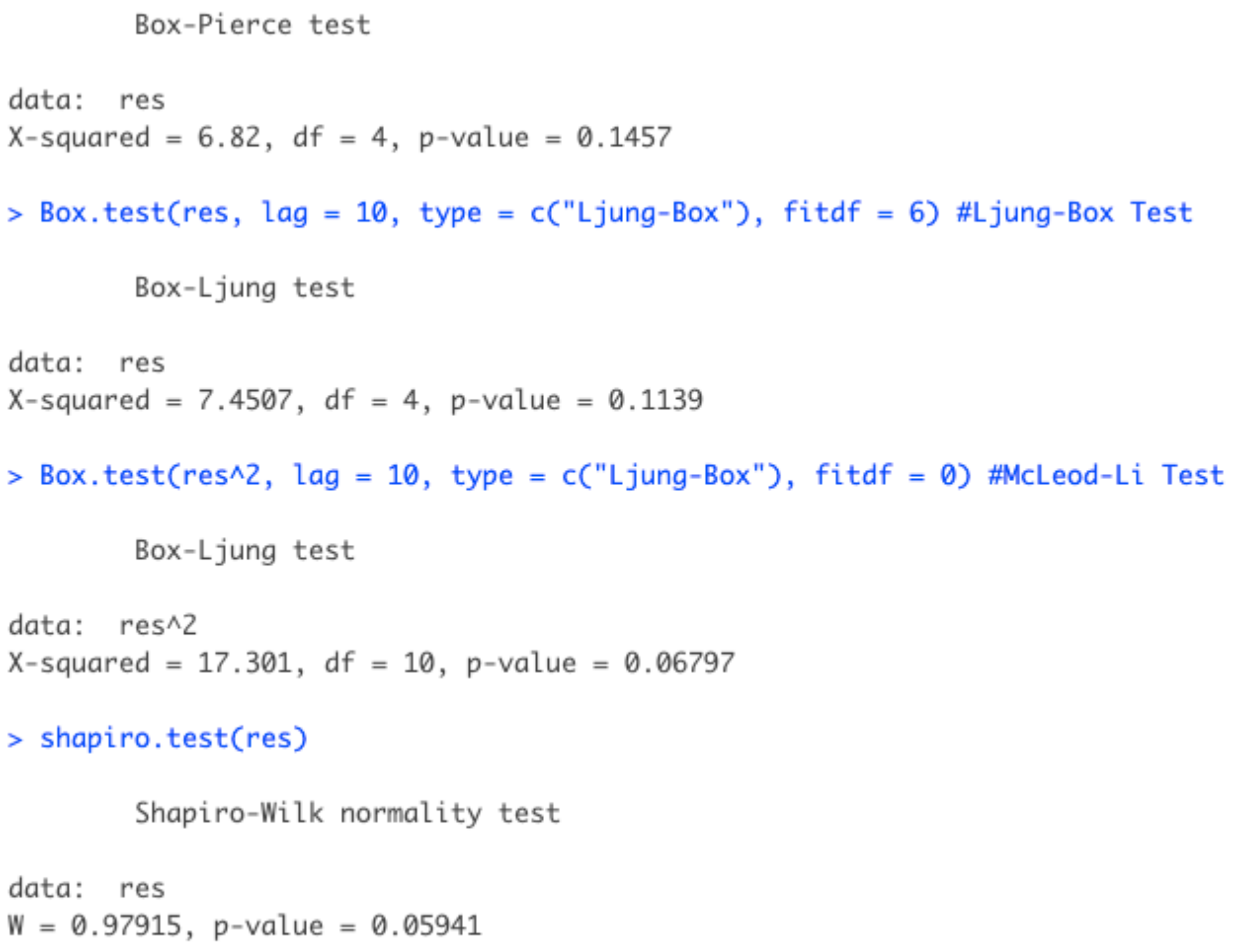

Histogram of res


Normal Q-Q Plot for Model 7


res
10
5
0
-5
0
20
40
60
80
100
120
Time

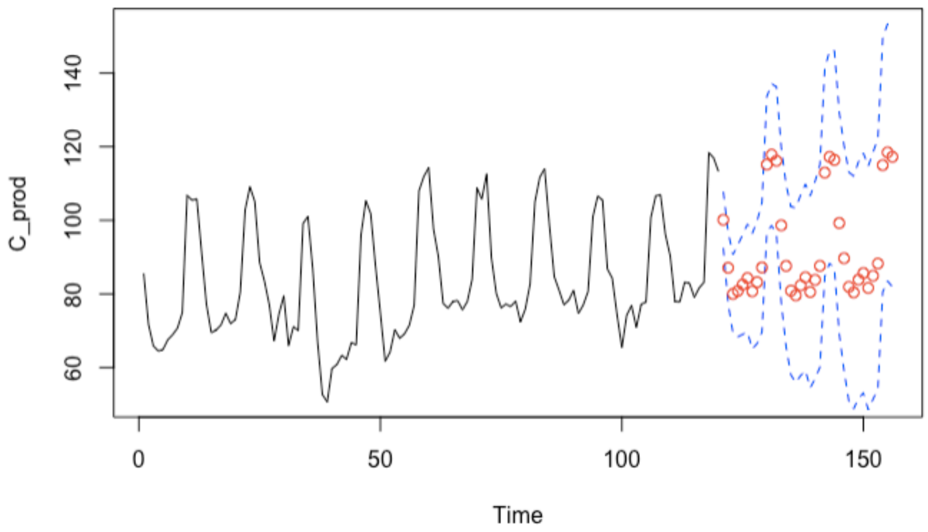

ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
10
20
30
40
Lag


Partial ACF
0.1
0.0
-0.1
-0.2
0
10
20
30
40
Lag

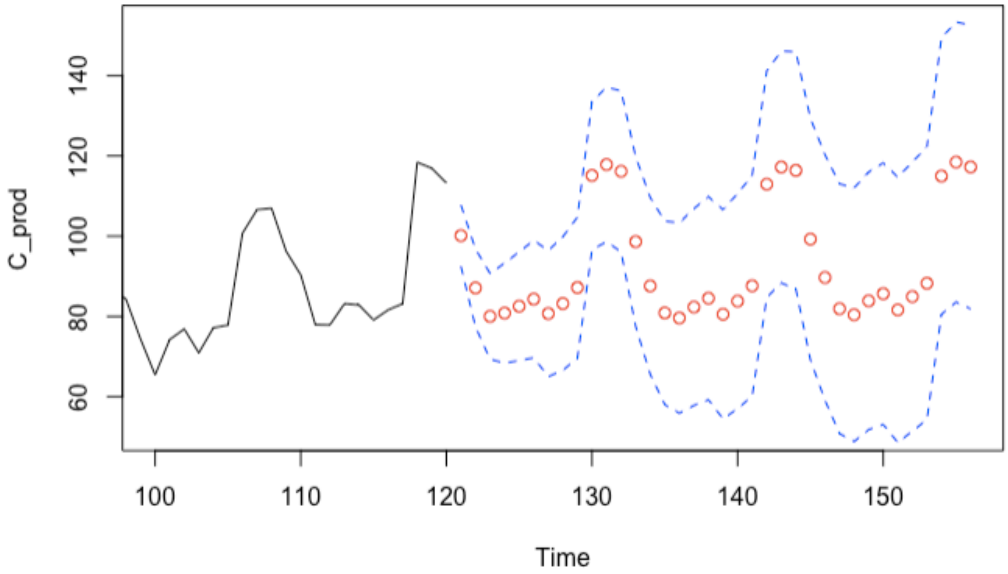


```
        Box-Pierce test

data:  res
X-squared = 6.82, df = 4, p-value = 0.1457

> Box.test(res, lag = 10, type = c("Ljung-Box"), fitdf = 6) #Ljung-Box Test

        Box-Ljung test

data:  res
X-squared = 7.4507, df = 4, p-value = 0.1139

> Box.test(res^2, lag = 10, type = c("Ljung-Box"), fitdf = 0) #McLeod-Li Test

        Box-Ljung test

data:  res^2
X-squared = 17.301, df = 10, p-value = 0.06797

> shapiro.test(res)

        Shapiro-Wilk normality test

data:  res
W = 0.97915, p-value = 0.05941
```

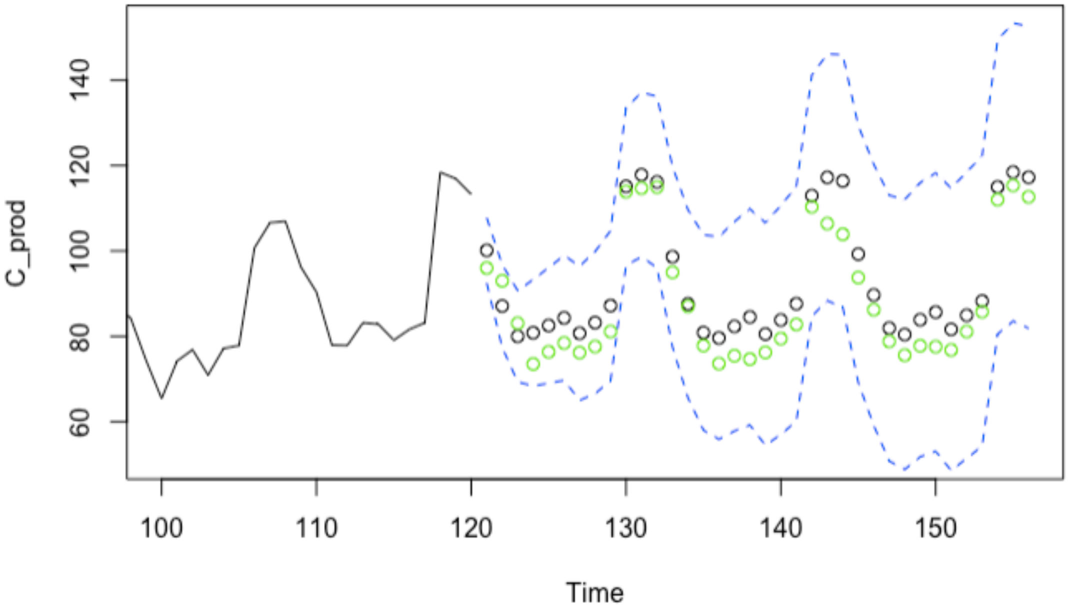
This graph shows 36 forecasts on original data



This graph shows zoomed version of our 36 forecasted values(red) starting from entry 100



This graph shows zoomed version of our 36 forecasted values(black) starting from entry 100 along with the test set values (green).



We observe that test set values are within the confidence intervals and very close to forecasted values.

Conclusion

We see that our testing data is within the confidence intervals and very close to the predicted values. Also, at time equals to 121 till 129 represent months from January to September and predicted values of these months are much lower than values at time equals to 130 till 132, which represent months from October to December. This pattern repeats again for next 24 months. This shows that candy production is higher for months of October, November, and December to cover increased demand for candy during this season due to the festivals like Thanksgiving, Halloween, Christmas, and New Year due to increased consumption of candies. Hence, our final model equation is (1+0.8717(B)+0.3806(B^2))(1-0.0241(B^12)+0.1995(B^24)) X_t = (1+0.6558B)(1-0.97(B^12)) Z_t ,where Z_t~WN(0,13.05). At the end, I would like to thank professor Raya Feldman and teaching assistant Youhong Lee for helping me with this project.

References

Professor Feldman, Raya. “Lecture Let’s Do a Time Series Project!”. *Gauchospace,* Winter ’22, https://gauchospace.ucsb.edu/courses/pluginfile.php/3746412/mod_resource/content/1/Lecture%2015-AirPass%20slides.pdf

Tatman, Rachael. “US Candy Production by Month from January 1972 to August 2017”. *Kaggle*. Datasets. https://www.kaggle.com/rtatman/us-candy-production-by-month

Appendix

~~~~
```{r}
#Plotting Original Time-Series Data
setwd("/Users/keshavkhanna/Desktop")
C_prod1 = read.csv("candy_production.csv") #US monthly candy production data
C_prod1
C_prod_plot = ts(C_prod1[,2], start = c(1972,1), frequency = 12)
plot.ts(C_prod_plot)
```
```{r}
setwd("/Users/keshavkhanna/Desktop")
C_prod = read.csv("candy_production.csv") #US monthly candy production data
C_prod_plot = ts(C_prod[,2], start = c(1972,1), end = c(1985,2), frequency = 12)
plot.ts(C_prod_plot)
nt=length(C_prod_plot)
fit1 <- lm(C_prod_plot ~ as.numeric(1:nt))
mean(C_prod_plot)[1]
abline(h=mean(C_prod_plot), col="blue")
```
```{r}
# Partition data-set to two parts for model training and model validation; work with training set:
C_prod = C_prod_plot[c(1:120)]  #Training Set
C_prod.test = C_prod_plot[c(121:156)] #Test Set
plot.ts(C_prod)
fit <- lm(C_prod ~ as.numeric(1:length(C_prod))); abline(fit, col="red")
abline(h=mean(C_prod), col="blue")
hist(C_prod, col="light blue", xlab="", main="histogram; candy production traning data") #plots
acf(C_prod,lag.max=40, main="ACF of the candy production training data")
```
```{r}
# To produce decomposition of (U_t):
library(ggplot2)
y = ts(as.ts(C_prod), frequency = 12)
decomp = decompose(y)
plot(decomp)
#Clearly the time series shows a linear trend and seasonality
```
```{r}
#Difference at lag 12 to remove seasonality
C_prod_plot.log_12 <- diff(C_prod, lag=12)
plot.ts(C_prod_plot.log_12 , main="(U_t) differenced at lag 12")
~~~~

~~~~
var(C_prod_plot.log_12)
fit <- lm(C_prod_plot.log_12 ~ as.numeric(1:length(C_prod_plot.log_12)))
mean(C_prod_plot.log_12)
abline(h=mean(C_prod_plot.log_12), col="blue")

#Difference at lag 1 to remove trend
C_prod_plot.stat <- diff(C_prod_plot.log_12, lag=1)
plot.ts(C_prod_plot.stat, main="(U_t) differenced at lag 12 and lag 1")

#Plotting trend and mean line again
fit <- lm(C_prod_plot.stat ~ as.numeric(1:length(C_prod_plot.stat)))
abline(fit, col="red")
mean(C_prod_plot.stat)
abline(h=mean(C_prod_plot.stat), col="blue")
```
```{r}
acf(C_prod_plot.stat, lag.max=60, main="ACF of U_t, differenced at lags 12 and 1")
hist(C_prod_plot.stat, col="light blue", xlab="", main="histogram; U_t differenced at lags 12 & 1")
hist(C_prod_plot.stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m<-mean(C_prod_plot.stat)
std<- sqrt(var(C_prod_plot.stat))
curve( dnorm(x,m,std), add=TRUE )
pacf(C_prod_plot.stat, lag.max=60, main="PACF of the U_t, differenced at lags 12 and 1")

#As noticed in the graph of PACF, PACF is outside the confidence intervals at lag 12 and lag 24, so the choice of P will be 1 or 2. Before lag 12, PACF is outside the confidence intervals at lag 1 and lag 2, so the choice of p will be 1 or 2. As noticed in the graph of ACF, ACF is outside the confidence intervals at lag 12 only so the choice of Q will be 1. Before lag 12, ACF is outside the confidence intervals at lag 1 and lag 3. So the choice of q will be 1 or 3.
```
```{r}
#List of candidate models based on ACF and PACF
library(rgl)
library(qpcR)

#List of candidate models p=1,2;q=1,3;P=1,2;Q=1;D=1;d=1;s=12

#Candidate model 1 p=1;q=1;P=1;Q=1
arima(C_prod, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12),method="ML")
AICc(arima(C_prod, order=c(1,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML"))
~~~~

~~~
#Candidate model 2 p=1;q=3;P=1;Q=1
arima(C_prod, order=c(1,1,3), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(1,1,3), seasonal = list(order = c(1,1,1), period =
12),method="ML"))

#Candidate model 3 q=1; p=2; Q=1 ; P=1
arima(C_prod, order=c(2,1,1), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(2,1,1), seasonal = list(order = c(1,1,1), period = 12),
method="ML"))

#Candidate model 4 q=3; p=2; Q=1 ; P=1
arima(C_prod, order=c(2,1,3), seasonal = list(order = c(1,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(2,1,3), seasonal = list(order = c(1,1,1), period = 12),
method="ML"))

#Candidate model 5 p=1;q=3;P=2;Q=1
arima(C_prod, order=c(1,1,3), seasonal = list(order = c(2,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(1,1,3), seasonal = list(order = c(2,1,1), period = 12),
method="ML"))

#Candidate model 6 p=1;q=3;P=2;Q=1
arima(C_prod, order=c(1,1,3), seasonal = list(order = c(2,1,1), period = 12),  method="ML")
AICc(arima(C_prod, order=c(1,1,3), seasonal = list(order = c(2,1,1), period = 12) ,
method="ML"))

#Candidate model 7 q=1; p=2; Q=1 ; P=2
arima(C_prod, order=c(2,1,1), seasonal = list(order = c(2,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
method="ML"))

#Candidate model 8 q=3; p=2; Q=1 ; P=2
arima(C_prod, order=c(2,1,3), seasonal = list(order = c(2,1,1), period = 12), method="ML")
AICc(arima(C_prod, order=c(2,1,3), seasonal = list(order = c(2,1,1), period = 12),
method="ML"))

# Model 4 has the lowest AICc
# Model 7 has the second lowest AICc
#Using the Principle of Parsimony the lowest AICc, I will consider Model 4 and Model 7


```
```{r}
# To test invertibility and seasonality of models
~~~

~~~~
library(UnitCircle)

# Model 4
uc.check(pol_ = c(1, 1.1634,-0.9995), plot_output = TRUE)  #Non seasonal AR part
uc.check(pol = c(1,1.0001,0.7927), plot_output = TRUE) #Non-seasonal MA part
uc.check(pol = c(1,-0.0062), plot_output = TRUE) #Seasonal AR part
uc.check(pol = c(1,-0.9998), plot_output = TRUE) #Seasonal MA part

#All roots are not outside the unit circle. Model is not stationary and invertible. Hence model 4 is
rejected


#Model 7

uc.check(pol_ = c(1, 0.8717,0.3806), plot_output = TRUE)  #Non seasonal AR part
uc.check(pol = c(1,0.6558), plot_output = TRUE) #Non-seasonal MA part
uc.check(pol = c(1,-0.0241,0.1995), plot_output = TRUE) #Seasonal AR part
uc.check(pol = c(1,-0.97), plot_output = TRUE) #Seasonal MA part

#All roots are outside the unit circle. Model is stationary and invertible. Hence we will move
forward with model 7 for diagnostic checking


```



```{r}
#Diagnostic checking of model 7

fit = arima(x = C_prod, order = c(2, 1, 1), seasonal = list(order = c(2, 1, 1), period = 12),
    method = "ML")
res <- residuals(fit)
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model 7")
qqline(res,col="blue")
~~~~

~~~
#ACF and PACF of the residuals and ACF of res^2
acf(res, lag.max=40)
pacf(res, lag.max=40)
acf(res^2, lag.max=40)


#Residual tests
Box.test(res, lag = 10, type = c("Box-Pierce"), fitdf = 6) #Box Pierce Test
Box.test(res, lag = 10, type = c("Ljung-Box"), fitdf = 6) #Ljung-Box Test
Box.test(res^2, lag = 10, type = c("Ljung-Box"), fitdf = 0) #McLeod-Li Test
shapiro.test(res)

ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

#No trend, no visible change of variance, no seasonality
#Sample mean is almost zero:
#Histogram and Q-Q plot look OK

```




```{r}

#Hence our fitted model for forecasting will be

# (1+0.8717(B)+0.3806(B^2))(1-0.0241(B^12)+0.1995(B^24))X_t = (1+0.6558B)
(1-0.97(B^12))Z_t where Z_t~WN(0,13.05)

library(forecast)
fit.A <- arima(C_prod, order=c(2,1,1), seasonal = list(order = c(2,1,1), period = 12),
method="ML")
forecast(fit.A) # prints forecasts with prediction bounds in a table

#We are going to be forecasting next 36 values for our time series

#To produce graph with 36 forecasts on original data:
pred.tr <- predict(fit.A, n.ahead = 36)
U.tr= pred.tr$pred + 2*pred.tr$se # upper bound of prediction interval
L.tr= pred.tr$pred - 2*pred.tr$se # lower bound of prediction interval
ts.plot(C_prod, xlim=c(1,length(C_prod)+36), ylim = c(min(C_prod),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
~~~

```
points((length(C_prod)+1):(length(C_prod)+36), pred.tr$pred, col="red")

#To zoom starting from 100 entry
ts.plot(C_prod, xlim = c(100,length(C_prod)+36), ylim = c(min(C_prod),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(C_prod)+1):(length(C_prod)+36), pred.tr$pred, col="red")

#To plot zoomed forecasts and true values:
ts.plot(C_prod, xlim = c(100,length(C_prod)+36), ylim = c(min(C_prod),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(C_prod)+1):(length(C_prod)+36), pred.tr$pred, col="black")
points((length(C_prod)+1):(length(C_prod)+36), C_prod.test, col="green")

```