Time Series Analysis of Consumer Confidence Index

Zhongyun Zhang

June 2021

Abstract

This time series project analyzes how consumer confidence index changes with time in the United States in recent 10 years. The question I am addressing is that whether the consumer confidence index of the American citizens have decreased for recent 12 months(2020-2021).

In the data processing step, I used box-cox, square root and log transformation and differencing method. I checked if the graph is stationary by plotting the time series plot and histogram. After model identification, I got several model candidates and applied parsimony principle and AICc to find the best model. By analyzing coefficients and aicc value, I selected two models that I believe them potential to proceed to the next step. In model diagnostics, I utilized normal Q-Q plot for residuals, performed Ran Shapiro-Wilk test of normality, plotted residuals and histogram of residuals, checked sample ACF/PACF of residuals, used Yue-Walker estimation and Portmanteau test(including Box-Pierce, Ljung-Box, and McLead-li). In the end, I got the best model and used it for data forecasting. In the predicted plot, I can see that my forecast value is really close to the true value and within the confidence interval. I conclude that the consumer confidence of US citizen for 2020-2021 did not decreased but hold for almost same value as last year(2019-2020). US citizens still have confidence with their jobs and incomes and willing to purchase, and they tend to spend more and save less.

Introduction

In this project, I will be working with "Consumer Confidence Index" provided by the Organization for Economic Co-operation and Development OECD.org. This data set included consumer confidence index for each country from January 1960 calculated monthly. The index for each month is a ramdom sample based on 5000 households in the country. There are two variables: time and value. However, I will only analyze on "United States" CCI from January 2010 to February 2021. I am aiming to use recent 10 years record to predict the consumer confidence index of US citizens in the next 12 months(April 2020 to March 2021). The problem I am planning to address is that if the consumer confidence index of the United States has decreased for the 2020-2021.

Consumer confidence index is a measure of the potential for developments of households' consumption and savings in the future based on 4 aspects: expected financial situation, sentiment about the general economic situation, unemployment and capability of savings. According to the introduction of the data set, an indicator with signals above 100 means a boost in households' consumer confidence toward the future economic situation. Households may tend to spend money rather than saving. For index value less than 100, it show that households are holding relative pessimistic attitude toward future developments of the economy, so they tend to save more money and spend less.

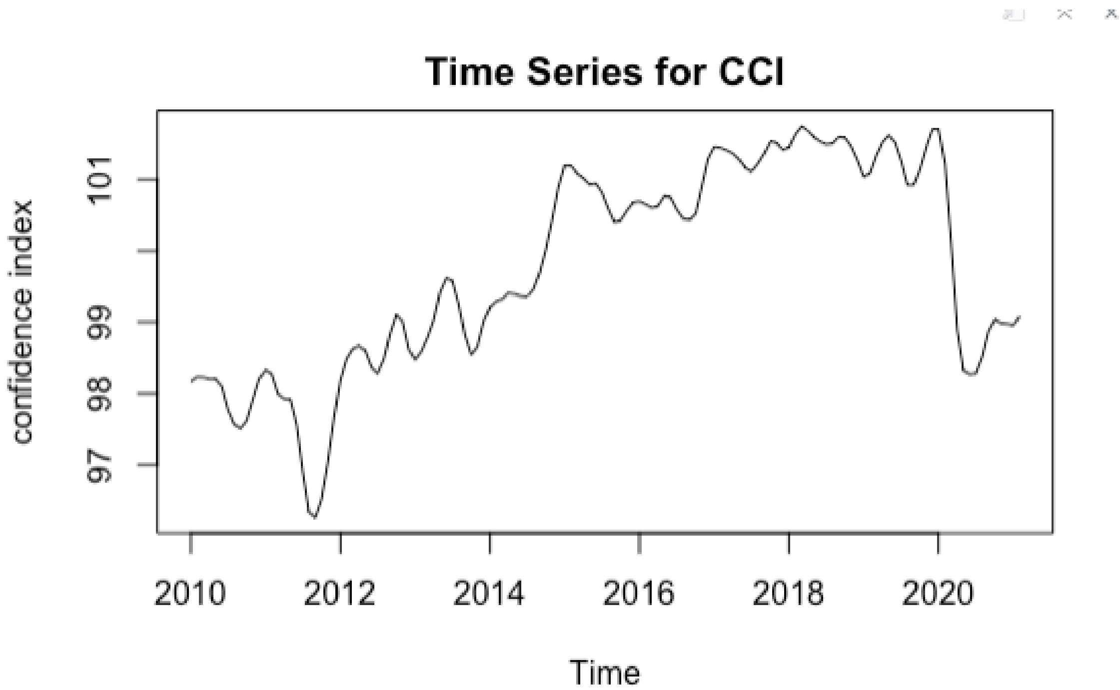
I find this CCI topic interesting to me because this is a index will give suggestion to retailer, bank, manufacture companies and many other companies reminder to make their future decision. In the past one year, the world has experienced huge change and global covid-19. Therefore, I am interested in how households of US citizens react.

To do the forecast, I considered transformation and differencing method in the data procession step. For transformation, I tried box-cox, square root and log transform. Furthermore, I differenced the data twice to remove seasonality, trend therefore make the data stationary. Then I plot ACF/PACF plots to identify models which suit for this data. In the data estimation process, I used parsimony method, check for AICc value and analyze the coefficients to get two model candidates to perform the model diagnostic. I employed normal Q-Q plot of residuals, Ran Shapiro-Wilk test of normality, residuals and histogram of residuals, sample ACF/PACF of residuals, Yue-Walker estimation and Portmanteau test(including Box-Pierce, Ljung-Box, and McLead-li). Finally, I chose the most suitable model and used it to perform the data forecast.

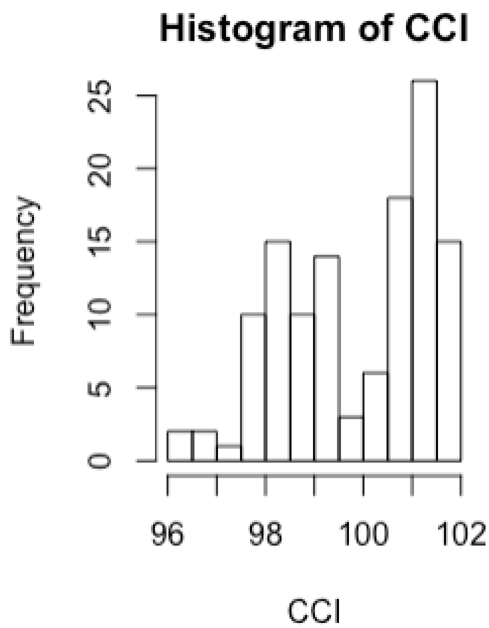
The result of the forecast is different from what I expect. The consumer confidence index for US citizens for April 2020 to March 2021 does not decrease but hold neutral and have the index over 100. This means that households are having positive mind toward the economics, and they tend to spend more and save less. The forecast also falls in 95% interval, which means my forecast is reasonable.

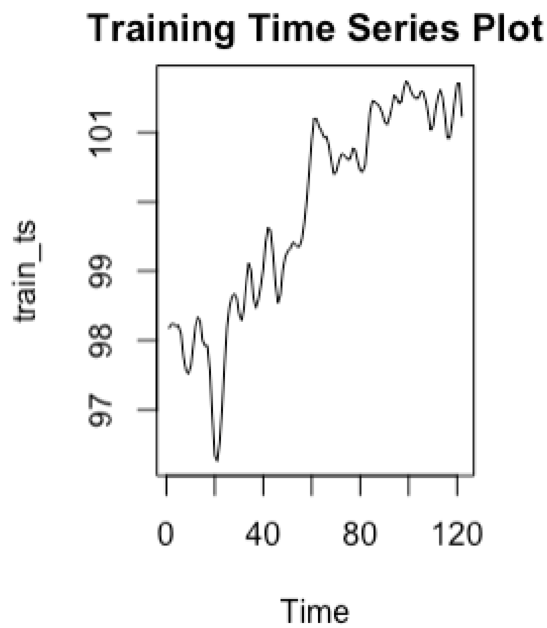
Data Processing

To begin with, we import the data and assign name for each variable. Value is the list of index from Jan2010 to Feb2021 of United States, and it consists of 134 observations.

```
```{r}
#data from Jan2010 to feb2021 of United States
setwd("~/Desktop")
confidence <- read.csv("~/Desktop/consumer_confidence.csv")
#head(confidence)
```
```

Then I plot the time series plot for the raw data. As we can see, the graph is non-stationary which has some seasonality and trend.



I split the data in to training(122 observations) and testing(12 observations) set.

```
```{r}
#split data into training set and test set
train <- confidence[1:122,7]
test <- confidence[123:134,7]
```
```

Then I plot the time series plot and histogram for the training data(code is in appendix). The histogram is not normal and the time series plot seems to have certain trend and non-stationary. I decided to try transformation or/and differencing method to make the data stationary.

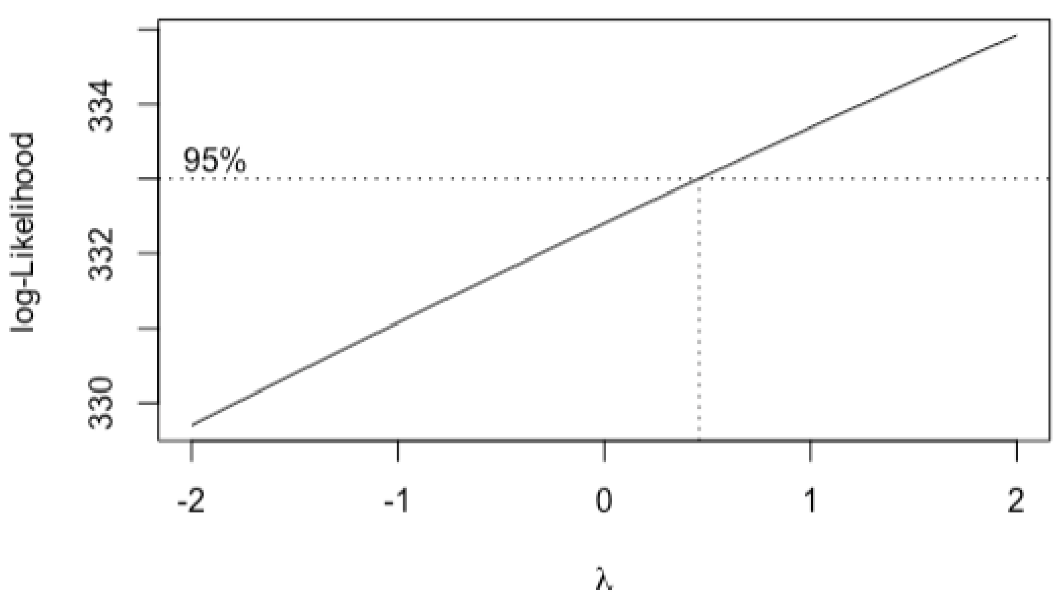




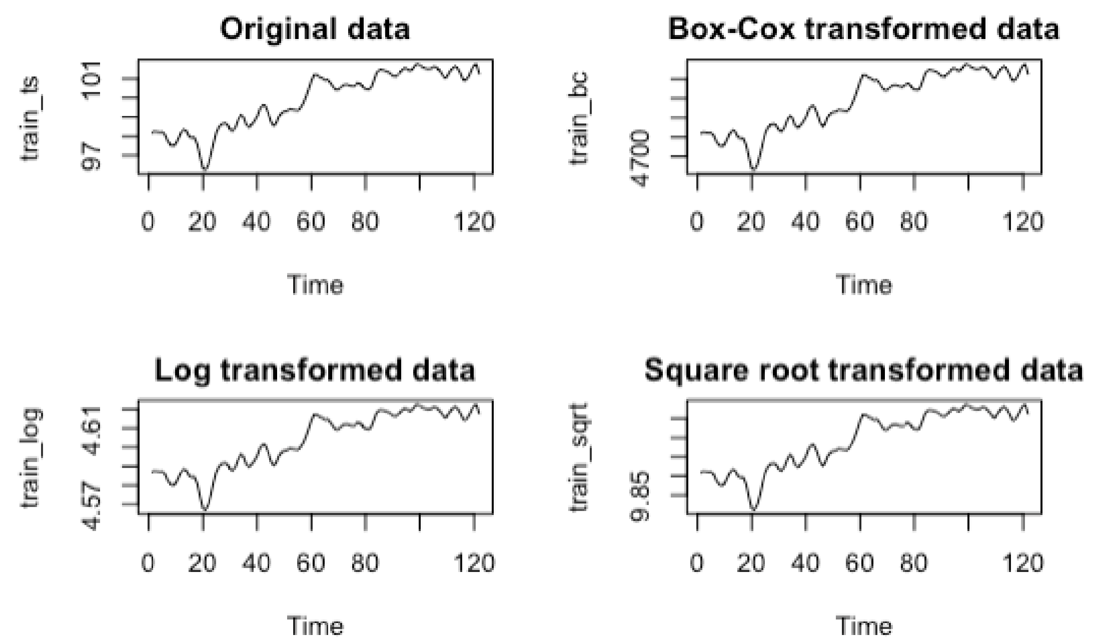
1. Transformation

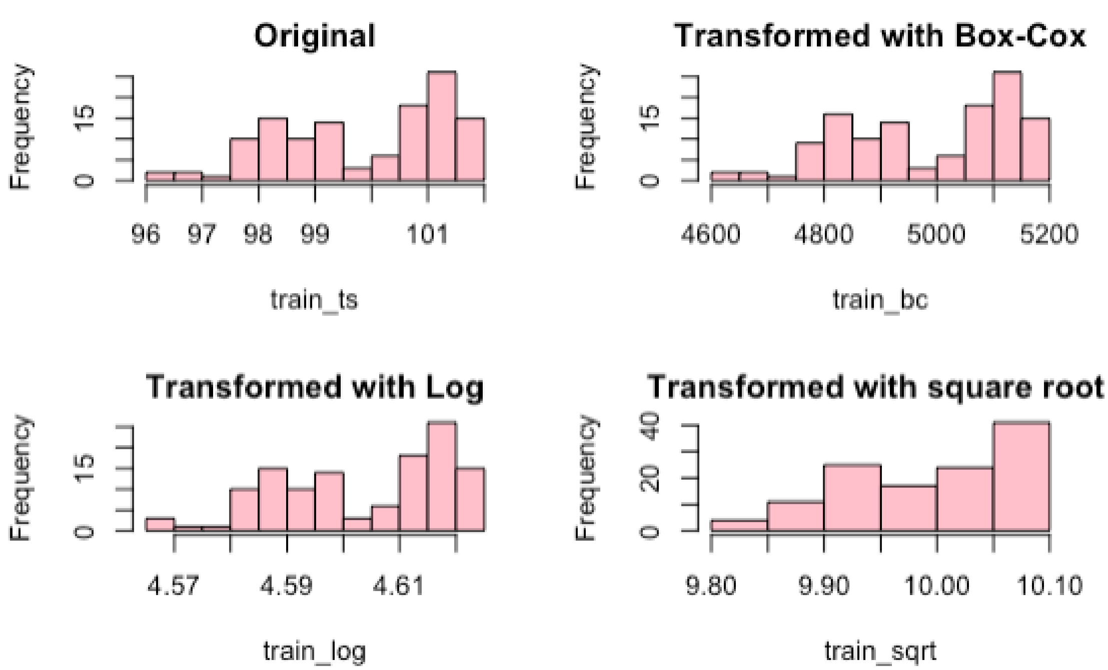
The first transformation method I decided to use is Box-cox transformation. By doing this, I hope this method can help our data look more like a normal distribution.

By working on R codes, we can get our lambda value to determine whether we want to do the Box-cox transformation. For $\lambda = 0$ within the interval, we should apply log transformation. For $\lambda = 0.5$ within the interval, we should use the square root transform. For $\lambda = 1$ within the interval, we do not need to use any transformation. I got the box-cox plot below with lambda value of 2.



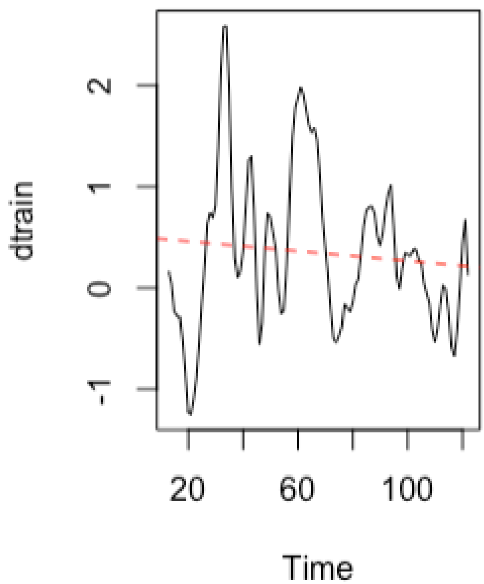
I also did log and square root transformation, although I am less likely to use them. I compare time series plot and histogram of log, box-cox and square root transformed data with my original data. I can see that log and box-cox transformed histogram is symmetric with double bell shaped distribution, while square root transformed histogram has y increase when x increases. None of the plots showed normal distribution. Moreover, the time series plot after transformation does not make big difference comparing to the original plot. Therefore, I will not consider doing a transformation at this moment. I will try differecing the data in the next step.

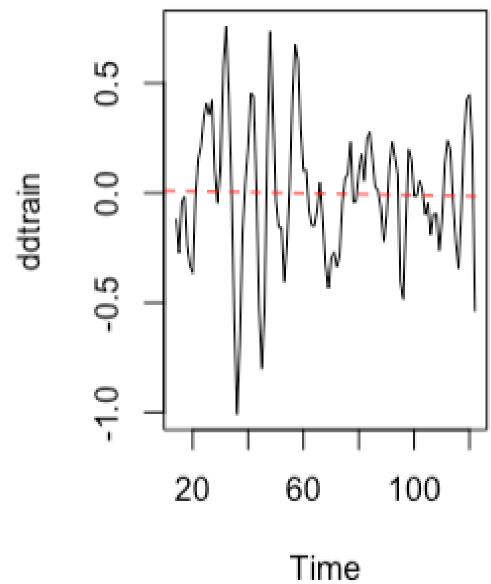




2. Differencing

To start with, I first tried differencing at lag 12. Although it may be hard to see the pattern of 12 on the graph, the data is recorded monthly. Thus, differening at lag 12 in the beginning should be a good choice. However, the time series plot after differencing still show slight trend. I decided to further difference at lag 1 to remove trend. As we can see the time series plot of differencing at lag 12 and lag 1 looks stationary.

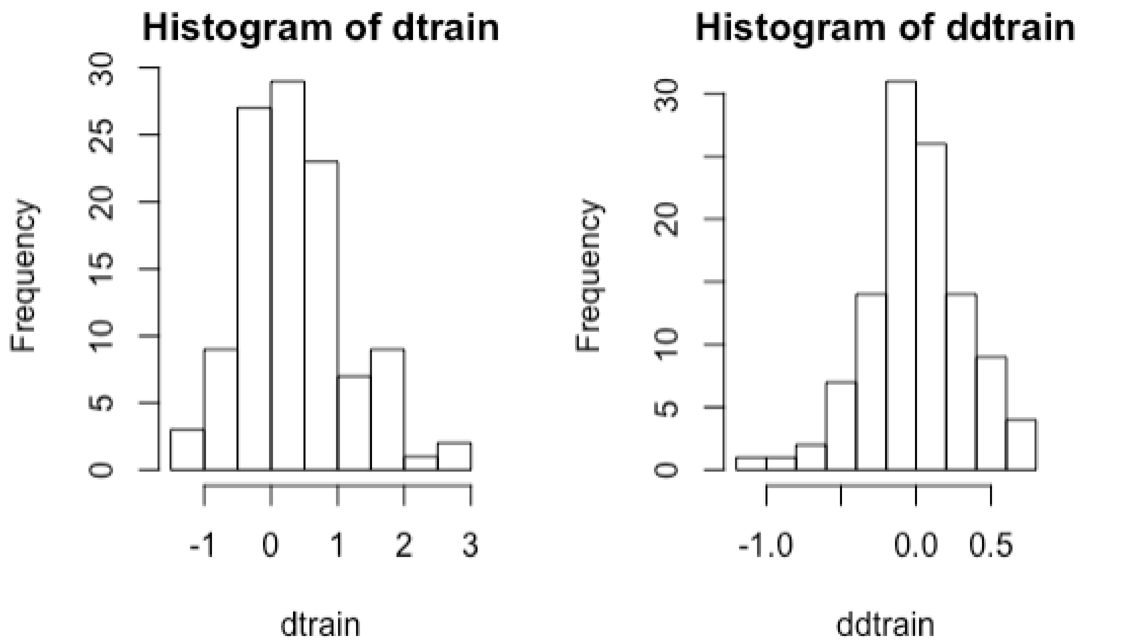




I also checked for variance. The original variance of the data is 2.227. The variance differenced at lag 12 is 0.6242. The variance differencing at lag 12 and lag 1 is 0.09863. There are significant decrease of variance after first and second difference. This means that differencing at lag 12 and lag 1 are significant.

```
[1] "Original vairance 2.22698805748482"
[1] "Variance differenced at lag 12 is 0.624195190709599"
[1] "Variance differenced at lag 12 and 1 is 0.0986266893347774"
```

Furthermore, the histogram with the data differenced at lag 12 removed seasonality and trend, but still a bit right skewed. While the plot with data differenced at lag 12 and lag 1 is normally distributed.



Model Identification

Now, I plot ACF and PACF plot after differencing at lag 12 and lag 1.

For the seasonal part (P, D, Q):

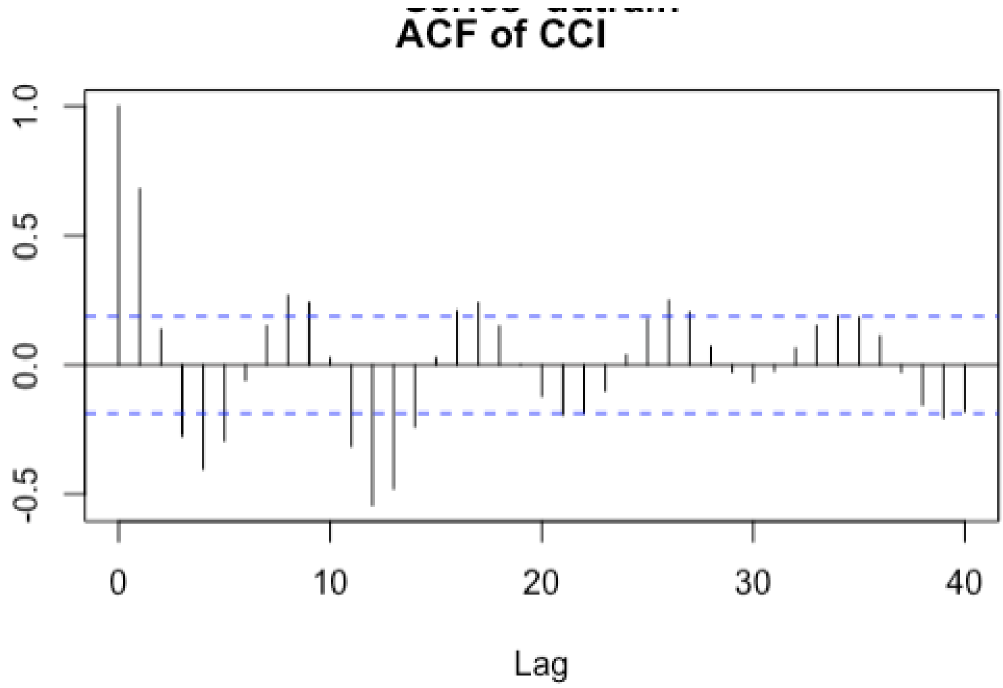
For ACF, I look at h = 12s, 24s, 36s, etc. The ACF shows a strong peak at h = 12s. A good choice for the MA part could be Q = 1.

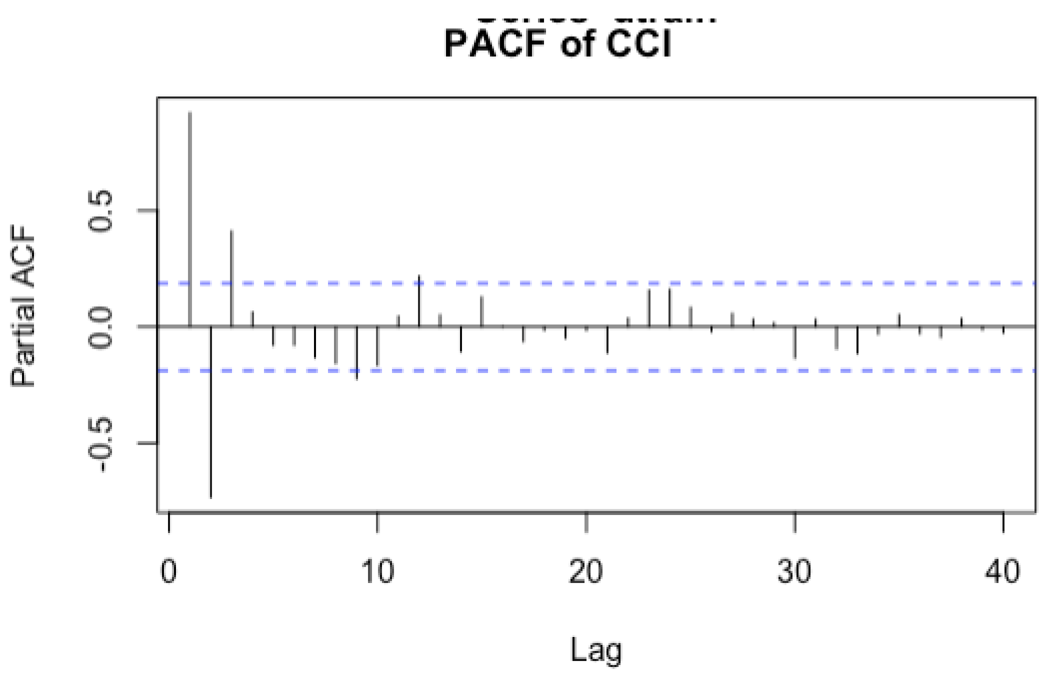
For PACF, I also look at h = 12s, 24s, 36s, etc. The PACF shows A strong peaks at h = 12s. While for h = 24, it is really close to the 95 % confidence interval. A good choice for the AR part could be P = 1 or P = 2 or maybe P = 0.

When modeling the non-seasonal part (p, d, q): In this case focus on the within season lags, h = 1,. . . ,11.

We applied one differencing to remove the trend: d = 1

The ACF seems to be tailing off, or perhaps cuts of at lag. A good choice for the MA part could be q = 5. The PACF cuts off at 3. A good choice for the AR part p could be up to 3.





Model Estimation and Model Selection

Now we want to find the best model with lowest AICc and reasonable coefficient. I start with SARIMA(0,1,1)X(0,1,1)12. However, the coefficient for sma1 is too big. In this way, we will not use this model.

```{r}
arima(train, order=c(0,1,1), seasonal = list(order = c(0,1,1), period = 12),
method="ML")
```

```
Call:
arima(x = train, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 1),
period = 12),
    method = "ML")

Coefficients:
         ma1     sma1
      0.8180  -1.0000
s.e.  0.0479   0.1415

sigma^2 estimated as 0.01848:  log likelihood = 48.34,  aic = -90.68
```

On the other hand, I tried SARIMA(2,1,1)X(0,1,1)12. Although this model has really small aicc, I decided not to use it because still the sma1 coefficient is too large.

```{r}
arima(train, order=c(2,1,1), seasonal = list(order = c(0,1,1), period = 12),
method="ML")
```

```
Call:
arima(x = train, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 1),
period = 12),
    method = "ML")

Coefficients:
         ar1      ar2     ma1     sma1
      1.0525  -0.5957  0.2554  -1.0000
s.e.  0.1100   0.0983  0.1322   0.1704

sigma^2 estimated as 0.01151:  log likelihood = 73.68,  aic = -137.37
```

After trying all possible models, there are two model that has small AICc and reasonable coefficent. The rest of model tried will be in the appendix.

The model with the lowest AICc is:

SARIMA(2,1,0)X(2,1,1)12

```
Call:
arima(x = train, order = c(2, 1, 0), seasonal = list(order = c(2, 1, 1),
period = 12),
    method = "ML")

Coefficients:
         ar1      ar2     sar1     sar2     sma1
      1.1710  -0.6717  -0.2297  -0.1470  -0.7812
s.e.  0.0736   0.0715   0.1586   0.1387   0.1885

sigma^2 estimated as 0.01268:  log likelihood = 73.32,  aic = -134.65
```

Also, the AICc value is:

```{r}
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,1), period
= 12), method="ML"))
```

```
[1] -134.1281
```

The second best model is:

SARIMA(2,1,0)X(2,1,0)12

```
Call:
arima(x = train, order = c(2, 1, 0), seasonal = list(order = c(2, 1, 0),
period = 12),
    method = "ML")

Coefficients:
         ar1      ar2     sar1     sar2
      1.1718  -0.6825  -0.8002  -0.4520
s.e.  0.0738   0.0719   0.0918   0.0958

sigma^2 estimated as 0.01459:  log likelihood = 69.7,  aic = -129.39
```

Also, the AICc value is:

```{r}
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,0), period
= 12), method="ML"))
```

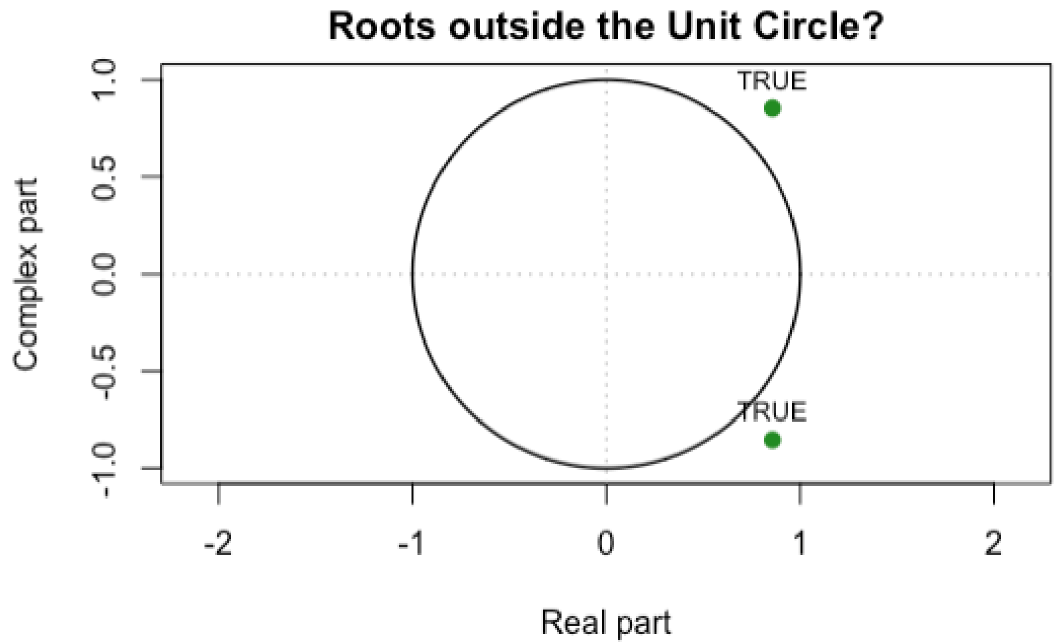
```
[1] -129.0507
```

Model Diagnostics

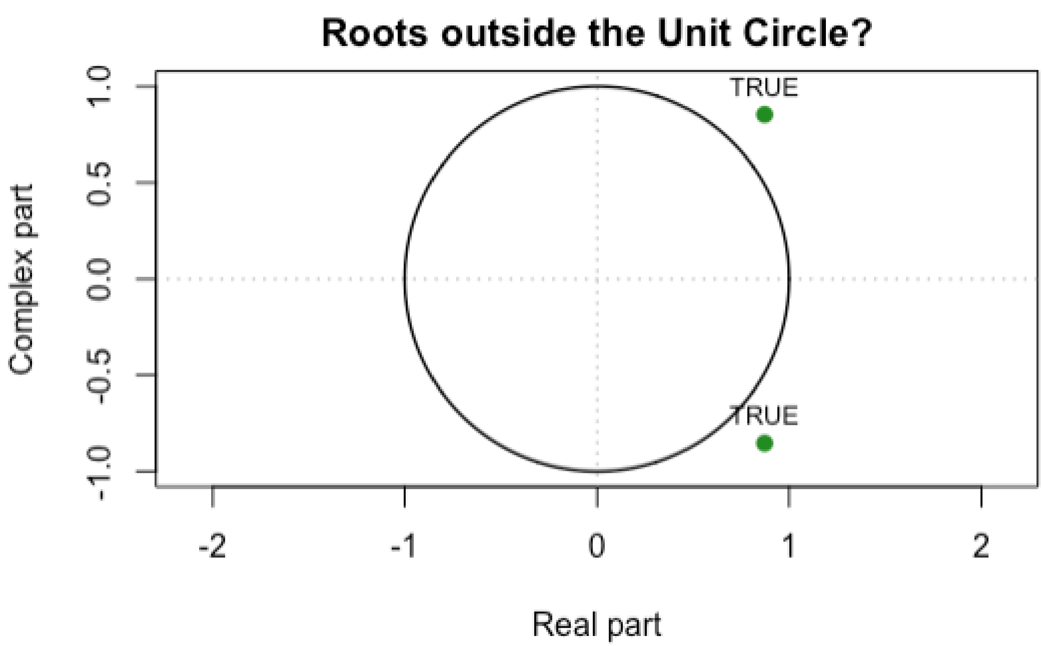
In this section, I will employ diagnostic procedures to get the final model. If the fit is good, then residuals resemble Gaussian WN(0,1)

One important step to start with is to check both models are invertible and causal of the model. I applied R code uc.check() to see if the roots are outside the unit circle. Gladly, both models are invertible and casual.

uc.check() plot for SARIMA(2,1,0)X(2,1,1)12:

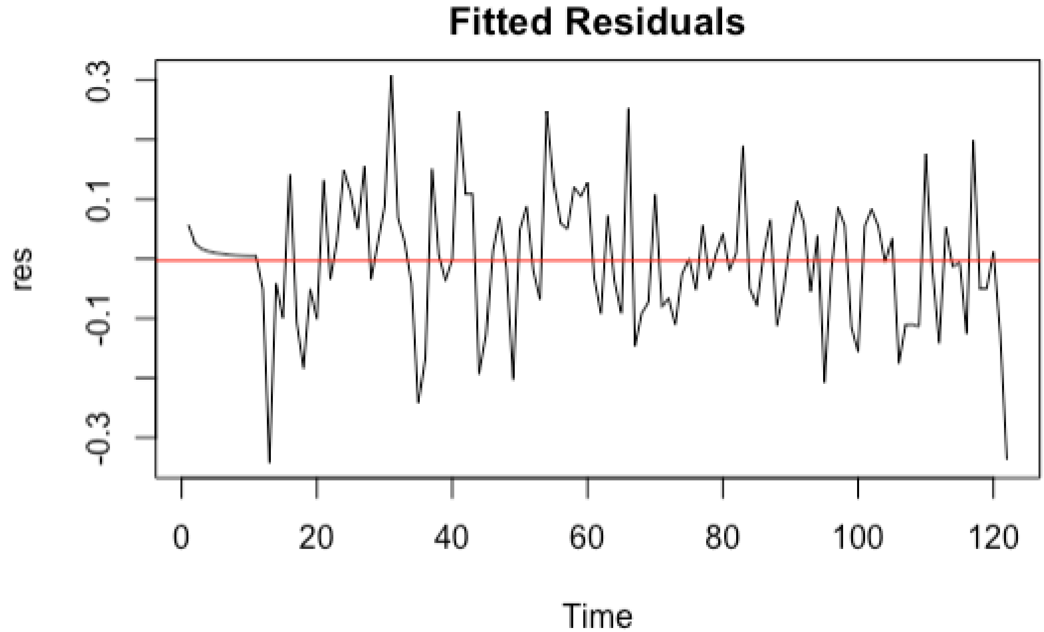


uc.check() plot for SARIMA(2,1,0)X(2,1,0)12:

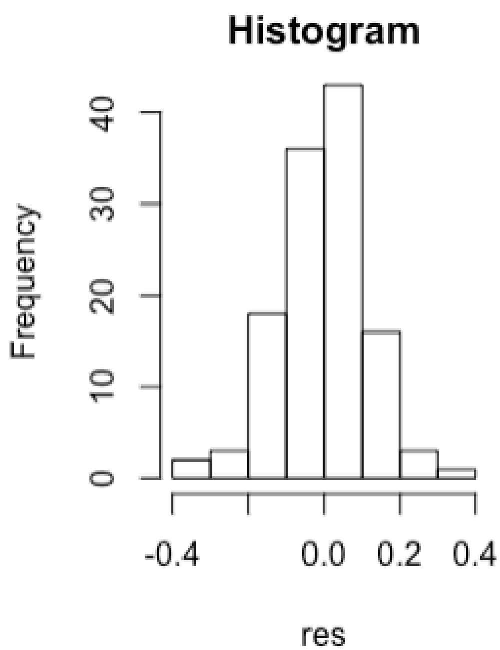


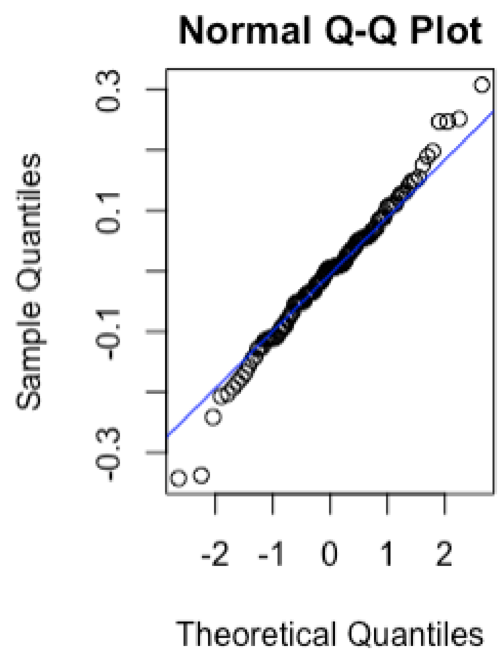
For MODEL 1: SARIMA(2,1,0)X(2,1,1)12

1. Plot Residuals in time series format. As we can see, the plot resemble to white noise. There are no trend, no seasonality, no change of variance.



2. Plot a histogram of residuals. The residuals on the histogram resemble to Gaussian distribution. Examine Normal Q-Q plot. Residuals on the plot is close to straight line. Although there are few outliers, we can ignore then because the most of the residuals formed a straight line.

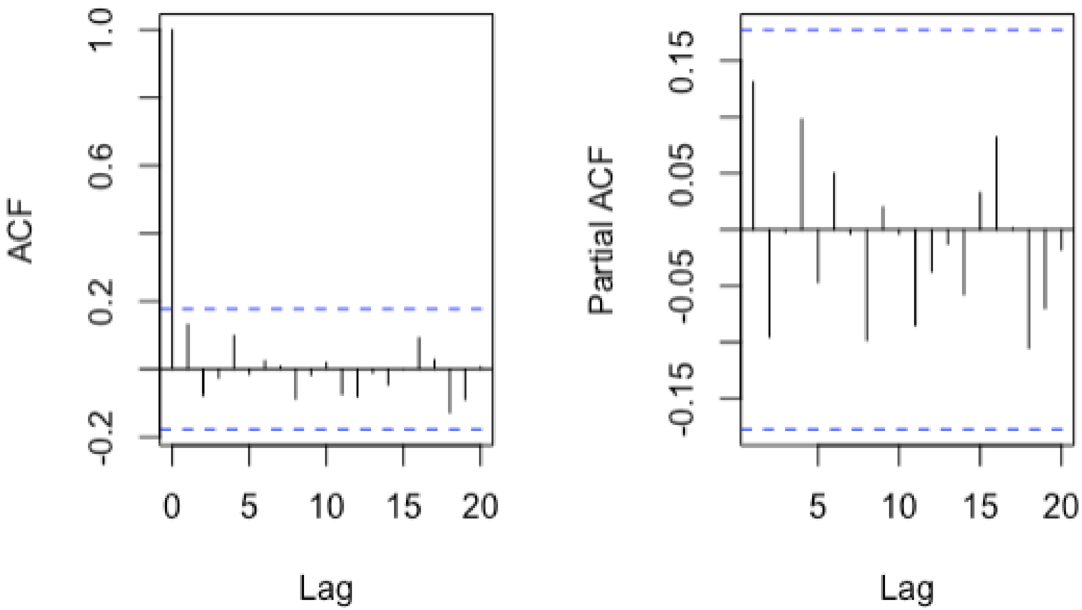




4. Ran Shapiro-Wilk test of normailty. p-value is 0.3817 which is greater than 0.05, therefore we do not reject assumption of normality. This model pass the test of Ran Shapiro-Wilk.

```
        Shapiro-Wilk normality test

data:  res
W = 0.98828, p-value = 0.3817
```

5. Check sample ACF/PACF of residuals. It also resemble to white noise. All sample acf/pacf inside confidence intervals. Data passes acf/pacf test.



7. Portmanteau statistics, with fitdf = q+p = 2+2+1 = 5. All p-values are larger than 0.05. This model pass all three tests.

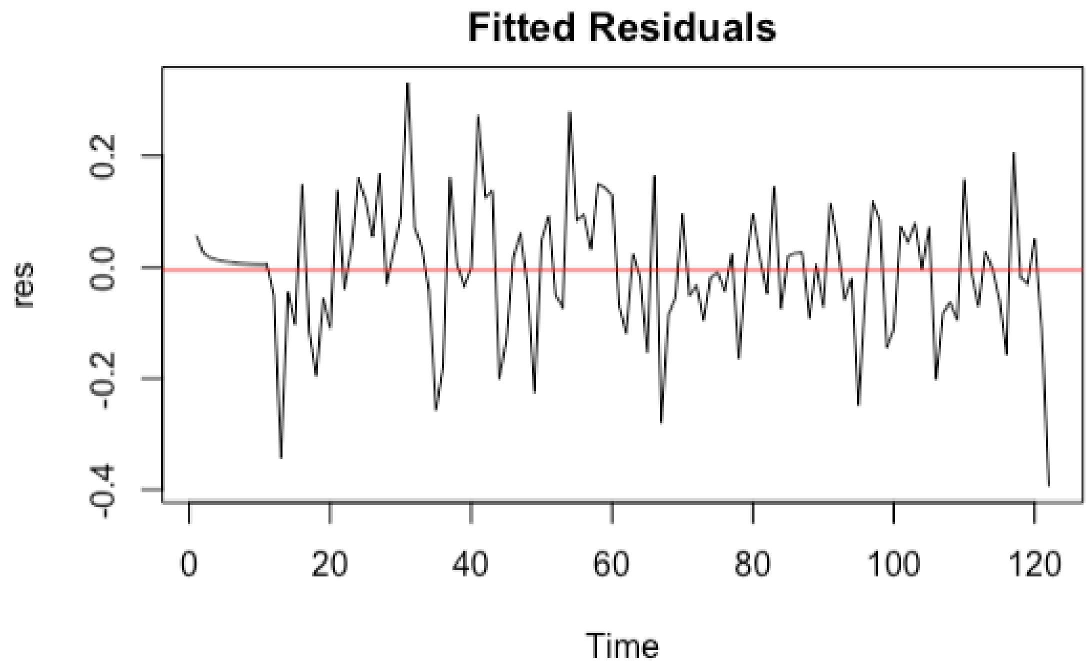
```
        Box-Pierce test

data:  res
X-squared = 6.6376, df = 7, p-value = 0.4676


        Box-Ljung test

data:  res
X-squared = 7.0512, df = 7, p-value = 0.4236


        Box-Ljung test

data:  res^2
X-squared = 9.3359, df = 12, p-value = 0.674
```

6. Use Yule-Walker estimation to residuals: as we can see model CANNOT be fit into AR(0)

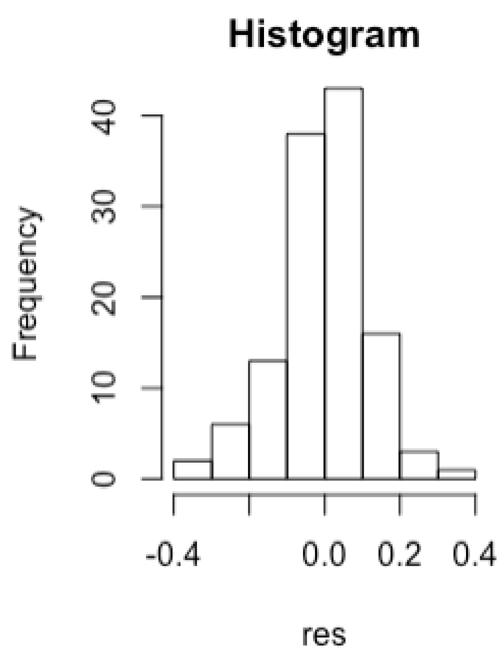
```
Call:
ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))

Coefficients:
     1
0.1311

Order selected 1  sigma^2 estimated as  0.01232
```

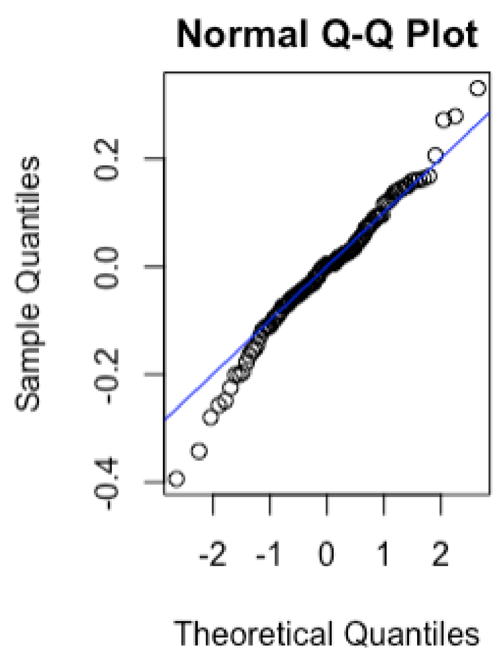
For MODEL 2: SARIMA(2,1,0)X(2,1,0)12

1. Plot Residuals in time series format. As we can see, the plot resemble to white noise. There are no trend, no seasonality, no change of variance.



2. Plot a histogram of residuals. The residuals on the histogram resemble to Gaussian distribution. Examine Normal Q-Q plot. Residuals on the plot is close to straight line. Although there are few outliers, we can ignore then because the most of the residuals formed a straight line.

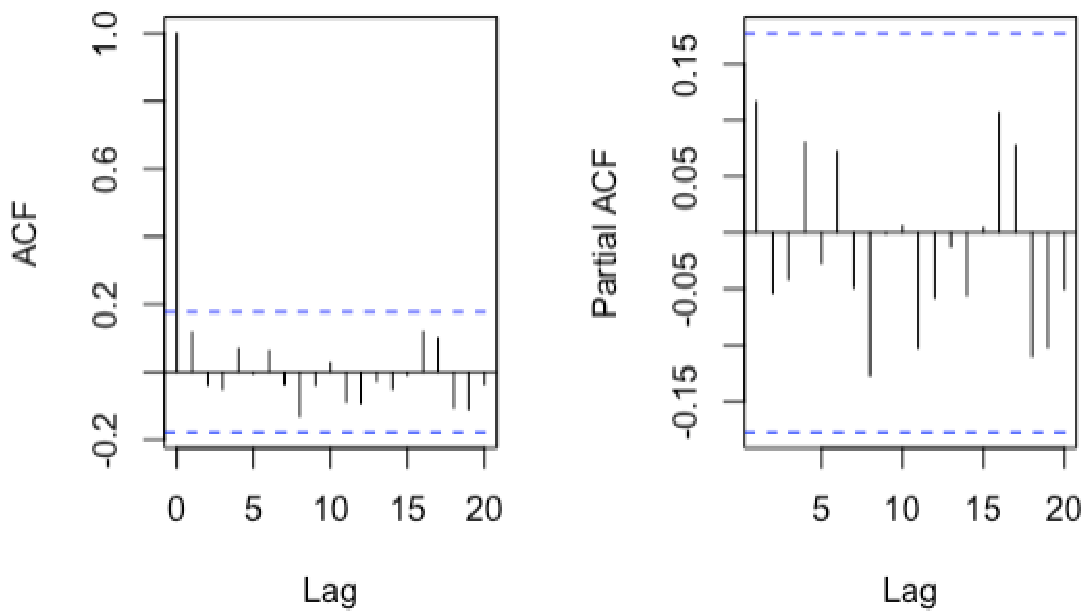




4. Ran Shapiro-Wilk test of normailty. p-value is 0.1001 which is greater than 0.05, therefore we do not reject assumption of normality. This model pass the test of Ran Shapiro-Wilk.

```
Shapiro-Wilk normality test

data:  res
W = 0.98186, p-value = 0.1001
```

5. Check sample ACF/PACF of residuals. it also resemble to white noise. All sample acf/pacf inside confidence intervals. Data passes acf/pacf test.



6. Portmanteau statistics, with fitdf = q+p = 2+2 = 4. All p-values are larger than 0.05. This model pass all three tests.

```
        Box-Pierce test

data:  res
X-squared = 7.7828, df = 8, p-value = 0.455


        Box-Ljung test

data:  res
X-squared = 8.3753, df = 8, p-value = 0.3977


        Box-Ljung test

data:  res^2
X-squared = 7.2833, df = 12, p-value = 0.8383
```

7. Use Yule-Walker estimation: as we can see model can be fit into AR(0).

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


Order selected 0  sigma^2 estimated as  0.01415
```

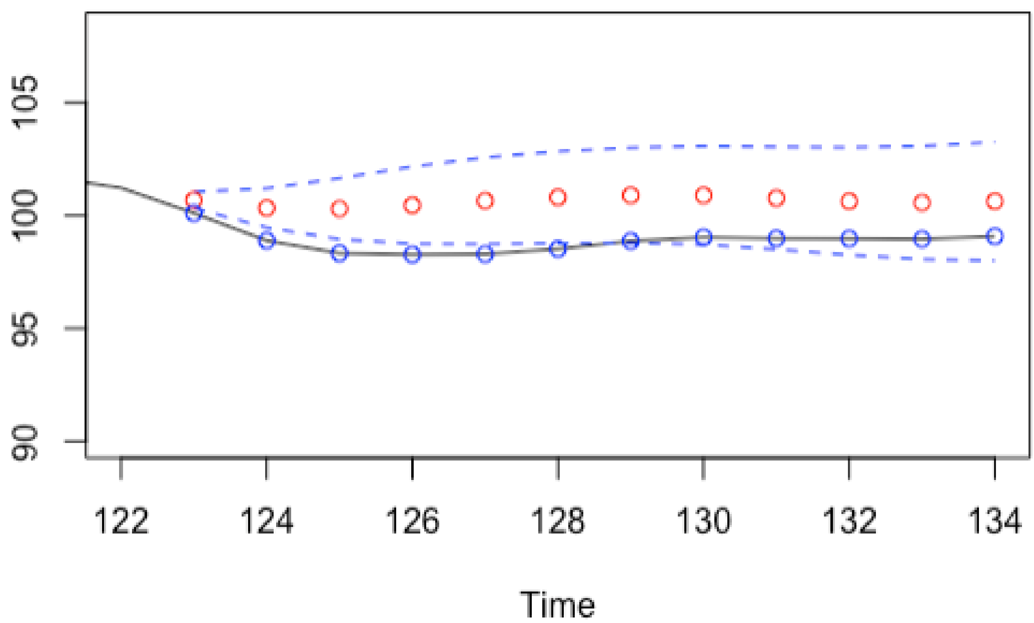
In conclusion, I decided to use model 2: SARIMA(2,1,0)X(2,1,0)12
(1-1.1718(0.0738)B+0.6825(0.0719)B^2)$(1 + 0.8002(0.0918)B^{12} + 0.4520(0.0958)B^{24})\nabla 1\nabla 12\nabla Xt = Zt$

Data Forecast

Since in model B order selected for Yule-Walker is not zero. I will use model 2 to do the forecasting. SARIMA(2,1,0)X(2,1,0)12

Model 2 does not pass the test for Yule-Walker, since its value is not zero. Therefore, I will use model 1 to perform the data forecast with observation 123 to 134.

As we can see, the forecast is quite good, the predicted plots are within the 95 % confidence interval and really close to the true value. Although it seems like predicted value and true value are apart from each other, I want to emphasize that this may because of scale and the confidence interval of the plot is small. In all, I believe the model performed well in the prediction.



Conclusion

Based on my time series analysis, I have discovered that the consumer confidence index for US citizens of this year is neutral, which people would spend more and save less for the following 12 months. They hold stable index for the next year(2020-2021). My goal is achieved, although the result is different than I expected, where I expected that the index will decrease. The final model is:

SARIMA(2,1,0)X(2,1,0)12

(1-1.1718(0.0738)B+0.6825(0.0719)B^2)$(1 + 0.8002(0.0918)B^{12} + 0.4520(0.0958)B^{24})\nabla 1\nabla 12\nabla Xt = Zt$

As a final comment, there are still some aspects to improve my project. For example, the forecast for the next 12 months is slightly above the true value and outside the 95% confidence interval. I believe there room to make better prediction.

Last but not the least, I really want to thanks to PSTAT174 course and professor Feldman. I have never been working on a project alone, and this course gives me chance to let me know my potential. Although this project truly has a lot to improve, I am glad that I tried my critical thinking, comprehensive knowledge learnt in class and Latex to finish this project.

Reference

1. PSTAT174 lecture notes and labs provided by professor Raisa Feldman.
2. Data is adopted from Organisation for Economic Co-operation and Development OECD.org, https://data.oecd.org/leadind/consumer-confidence-index-cci.htm
3. Information and background knowledge about consumer confidence index from Wikipedia, https://en.wikipedia.org/wiki/Consumerconfidenceindex

Appendix

```
‘‘‘r
install.packages(”MASS”)
library(”MASS”)
install.packages(”AICcmodavg”)
library(AICcmodavg)
install.packages(”rgl”)
library(rgl)
install.packages(”qpcR”)
library(qpcR)
install.packages(”UnitCircle”)
library(UnitCircle)
install.packages(”astsa”)
library(astsa)
‘‘‘
‘‘‘r
data from Jan2010 to feb2021 of United States
setwd(” /Desktop”)
confidence ¡- read.csv(” /Desktop/consumer_confidence.csv”)
head(confidence)
‘‘‘ ‘‘‘r
value ¡- confidence[1:134,7]
data_ts ¡- ts(value, start = c(2010,1), frequency = 12)
ts.plot(data_ts,xlab=”time”, ylab=”confidence index”, main=”Time Series for CCI”)
‘‘‘
‘‘‘r
split data into training set and test set
train ¡- confidence[1:122,7]
test ¡- confidence[123:134,7]
‘‘‘
‘‘‘r According to the histogram we decide to transform the data
par(mfrow = c(1,2))
hist(train, main = ”Histogram of CCI”, xlab = ”CCI”)
train_ts ¡- ts(train)
```

ts.plot(train_ts, main=”Training Time Series Plot”)
box-cox transformation
t ¡- as.numeric(1:length(train_ts))
fit ¡- lm(train_ts t)
bcTransform ¡- boxcox(train_ts t, plotit = TRUE)
lambda ¡- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
train_bc ¡- (1/lambda)*(train_tsl̂ambda - 1)
“‘ “‘r
log transform
train_log ¡- log(train_ts)
“‘
“‘r
square root transform
train_sqrt ¡- sqrt(train_ts)
“‘
“‘r
compare transforms
par(mfrow = c(2,2))
ts.plot(train_ts, main = ”Original data”)
ts.plot(train_bc, main = ”Box-Cox transformed data”)
ts.plot(train_log, main = ”Log transformed data”)
ts.plot(train_sqrt, main = ”Square root transformed data”)
“‘
“‘r
compare histogram
par(mfrow = c(2,2))
hist(train_ts, col=”pink”, main=”Original”)
hist(train_bc, col=”pink”, main=”Transformed with Box-Cox”)
hist(train_log, col=”pink”, main=”Transformed with Log”)
hist(train_sqrt, col=”pink”, main=”Transformed with square root ”)
“‘
“‘r
ts.plot(train_ts, main = ”Original data”)
abline(abline(fit), col = ”red”, lty = 2)
original variance acf/pacf
print(paste(”Variance of original data is”,var(train_ts)))
acf(train, lag.max = 40, main=””)
title(”ACF of CCI”, line = -1, outer = TRUE)
pacf(train_ts, lag.max = 40, main=””)
title(”PACF of CCI”, line = -1, outer = TRUE)
“‘
“‘r

print(paste(”Original vairance”, var(train)))
dtrain ¡- diff(traints, lag = 12)
ddtrain ¡- diff(dtrain, lag = 1)
par(mfrow = c(1,2))
ts.plot(dtrain)
abline(lm(dtrain as.numeric(1:length(dtrain))), col = ”red”, lty = 2)
print(paste(”Variance differenced at lag 12 is”, var(dtrain)))
ts.plot(ddtrain)
abline(lm(ddtrain as.numeric(1:length(ddtrain))), col = ”red”, lty = 2)
print(paste(”Variance differenced at lag 12 and 1 is”, var(ddtrain)))
par(mfrow = c(1,2))
hist(dtrain)
hist(ddtrain)
“‘
“‘r
acf(ddtrain, lag.max = 40)
title(”ACF of CCI”, line = -1, outer = TRUE)
pacf(dtrain, lag.max = 40)
title(”PACF of CCI”, line = -1, outer = TRUE)
“‘
“‘r arima(train, order=c(2,1,0), seasonal = list(order = c(1,1,0), period = 12), method=”ML”)
arima(train, order=c(0,1,0), seasonal = list(order = c(0,1,0), period = 12), method=”ML”)
arima(train, order=c(0,1,1), seasonal = list(order = c(1,1,0), period = 12), method=”ML”)
arima(train, order=c(2,1,0), seasonal = list(order = c(1,1,1), period = 12), method=”ML”)
arima(train, order=c(2,1,0), seasonal = list(order = c(5,1,0), period = 12), method=”ML”)
arima(train, order=c(1,1,0), seasonal = list(order = c(0,1,1), period = 12), method=”ML”)
arima(train, order=c(2,1,1), seasonal = list(order = c(2,1,0), period = 12), method=”ML”)
arima(train, order=c(2,1,1), seasonal = list(order = c(2,1,0), period = 12), method=”ML”)
arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,1), period = 12), method=”ML”)
arima(train, order=c(2,1,0), seasonal = list(order = c(5,1,0), period = 12), method=”ML”)
“‘
“‘r
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(1,1,1), period = 12), method=”ML”))
AICc(arima(train, order=c(0,1,0), seasonal = list(order = c(0,1,0), period = 12), method=”ML”))
AICc(arima(train, order=c(0,1,1), seasonal = list(order = c(1,1,0), period = 12), method=”ML”))
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(1,1,1), period = 12), method=”ML”))
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(5,1,0), period = 12), method=”ML”))
AICc(arima(train, order=c(1,1,0), seasonal = list(order = c(0,1,1), period = 12), method=”ML”))
AICc(arima(train, order=c(2,1,1), seasonal = list(order = c(2,1,0), period = 12), method=”ML”))
AICc(arima(train, order=c(2,1,1), seasonal = list(order = c(2,1,0), period = 12), method=”ML”))
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,1), period = 12), method=”ML”)
AICc(arima(train, order=c(2,1,0), seasonal = list(order = c(5,1,0), period = 12), method=”ML”)) “‘
“‘r

Model 1:
fit.i ¡-arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,0), period = 12),method=”ML”)
‘‘‘
‘‘‘r
check for invertibility and causality
uc.check(pol_=c(1,-1.1710,0.6825), plot_output = TRUE)
‘‘‘
‘‘‘r
Statistics of the residuals:
res ¡- residuals(fit.i)
m ¡- mean(res)
ts.plot(res, main = ”Fitted Residuals”)
abline(h = mean(res), col = ”red”)
‘‘‘
‘‘‘r
par(mfrow = c(1,2))
Check the normality assumption:
hist(res,main = ”Histogram”)
qqnorm(res)
qqline(res, col=”blue”)
title(”Fitted Residuals Diagnostics”, outer = TRUE)
‘‘‘
‘‘‘r
Test for the normality of residuals:
shapiro.test(res)
‘‘‘
‘‘‘r
95
par(mfrow=c(1, 2))
acf(res,main = ”Autocorrelation”)
pacf(res,main = ”Partial Autocorrelation”)
‘‘‘
‘‘‘r
Test for independence of residuals:
Box.test(res, lag=12, type=c(”Box-Pierce”), fitdf=5)
Box.test(res, lag=12, type=c(”Ljung-Box”), fitdf=5)
Box.test(res$^2, lag = 12, type = c(''Ljung - Box''), fitdf = 0$)
‘‘‘
‘‘‘r
ar(res, aic = TRUE, order.max = NULL, method = c(”yule-walker”))
‘‘‘‘‘‘r
check for invertibility and causality
uc.check(pol_=c(), plot= TRUE)

‘‘‘
‘‘‘r
Model 2:
fit.i ¡-arima(train, order=c(2,1,0), seasonal = list(order = c(2,1,0), period = 12),method=”ML”)
‘‘‘
‘‘‘r
check for invertibility and causality
uc.check(pol_=c(1,-1.1718,0.6717), plot_output = TRUE)
‘‘‘
‘‘‘r
Statistics of the residuals:
res ¡- residuals(fit.i)
m ¡- mean(res)
ts.plot(res, main = ”Fitted Residuals”)
abline(h = mean(res), col = ”red”)
‘‘‘
‘‘‘r
par(mfrow = c(1,2))
Check the normality assumption:
hist(res,main = ”Histogram”)
qqnorm(res)
qqline(res, col=”blue”)
title(”Fitted Residuals Diagnostics”, outer = TRUE)
‘‘‘
‘‘‘r
Test for the normality of residuals:
shapiro.test(res)
‘‘‘
‘‘‘r
95
par(mfrow=c(1, 2))
acf(res,main = ”Autocorrelation”)
pacf(res,main = ”Partial Autocorrelation”)
‘‘‘
‘‘‘r
Test for independence of residuals:
Box.test(res, lag=12, type=c(”Box-Pierce”), fitdf=4)
Box.test(res, lag=12, type=c(”Ljung-Box”), fitdf=4)
Box.test(res$^2, lag = 12, type = c(”Ljung - Box”), fitdf = 0$)
‘‘‘
‘‘‘r
ar(res, aic = TRUE, order.max = NULL, method = c(”yule-walker”))
‘‘‘‘r

check for invertibility and causality

uc.check(pol_=c(), plot= TRUE)

‘‘‘

‘‘‘r

pred.tr ¡- predict(fit.i, n.ahead= 12)

pred.orig ¡- pred$pred$\hat{2}$

upper= pred.tr$pred + 2 * pred.tr$se

lower= pred.tr$pred - 2 * pred.tr$se

ts.plot(confidence, xlim=c(122,length(train)+12), ylim= c(90,max(upper)+5))

lines(upper, col=”blue”, lty=”dashed”)

lines(lower, col=”blue”, lty=”dashed”)

points((length(train)+1):(length(train)+12), pred.tr$pred, col=”red”)

points((length(train)+1):(length(train)+12),pred.orig, col=”blue”)

‘‘‘