Time Series Analysis: Forecasting Monthly Average Watermelon Sales Price in Nepal from 2013-2021

Edward Ho

Abstract

From the first civilization to now, agriculture has been a pivotal part of our culture. Additionally, agriculture and the commerce go hand-in-hand as agriculture plays a vital part in both the well-being of the nation and the economy as well. Therefore, for this report, I will be analyzing the changing average monthly sales price of watermelon in Nepal from 2013 to 2021 because I want to forecast potential prices for watermelon for both markets and individuals to make financial decisions that would suite their budget.

In this analysis, I will forecast the trend of watermelon prices to potentially gauge periods of cheaper watermelon prices for purchase. Notably, these periods also lead into the start of COVID-19, so I want to reveal if the pandemic may have altered watermelon prices. To answer this question, I will perform a time series analysis using a Box-Jenkins Seasonal Autoregressive integrated moving average model to forecast the sales price of watermelon sold in Neal in 2021.

With no surprise, the results did not sit entirely in my predicted 95% confidence interval because COVID-19 played a large effect in increasing the sales price of watermelon beyond the typical seasonal trend of the sales price. Since the model wasn't accurate in terms of confidence interval and predictive accuracy, I don't believe any future model would be inappropriate in this case because COVID-19 had led to unexpected changes that a model would not be able to forecast or account for.

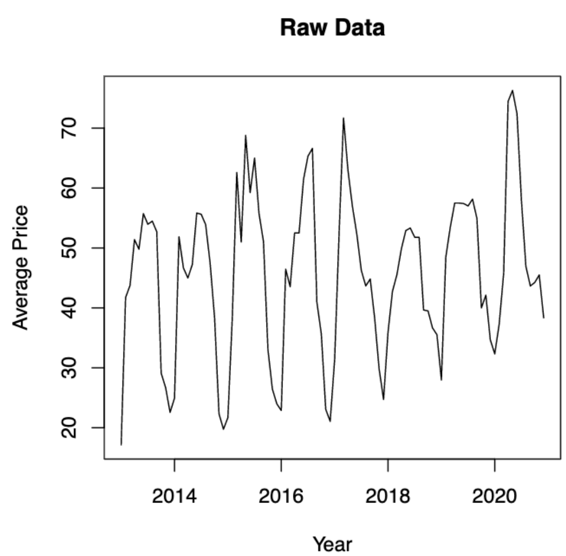
Introduction

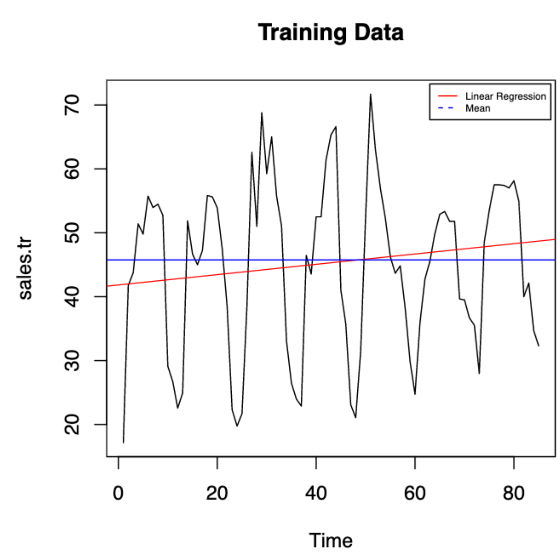
Watermelon is a staple item that is commonly used as an ingredient in many cuisines, desserts, and drinks, therefore, there many places would benefit from a predictive model that would best indicate when it's a good time to purchase watermelon in bulk. Many places often change their menu based on seasonality, availability, and price, so capitalizing on a predictive market to best alter a menu based on these factors would yield the most gains relative to cost.

In order to examine changes in sales price, I utilized a dataset from Kaggle which pulled fruits and vegetable sales prices scrapped from the website of Kalimati Fruits and Vegetable Market Development Board. The dataset contained a multitude of fruits and vegetables, so I first cleaned the dataset to only contain the average watermelon prices from 2013 to 2021. Afterwards, I separated the data into a training and testing set, for which the training set contained 85 observations and the testing set contained 11 observations. Then, I performed transformative processes necessary to make the data relatively stationary and normal. Once the appropriate transformations were made, I analyzed the autocorrelations and partial autocorrelation graphs in order to find two Seasonal Autoregressive integrated moving average (SARIMA) models that was chosen based on the criteria: Akaike's Information Criterion (AICc) score. The model with the lowest AICc score was then used for forecasting. Before that, though, I performed diagnostic testing on both models to ensure the residuals resembled a Gaussian White Noise process. I generally yielded poor results, as seen by the poorly created confidence intervals and a failure for the forecasted points to match the true points, indicating that COVID-19 had a great effect on the sales of watermelon that we weren't able to forecast.

The software that I used to perform these tests and analysis was in RStudio utilizing the packages: lubridate, dpylr, MASS, ggplot2, ggfortify, forecast, MuMIn, readr, xts, astsa, and tidyr. All data manipulation and cleaning were done entirely in RStudio using these packages as well.

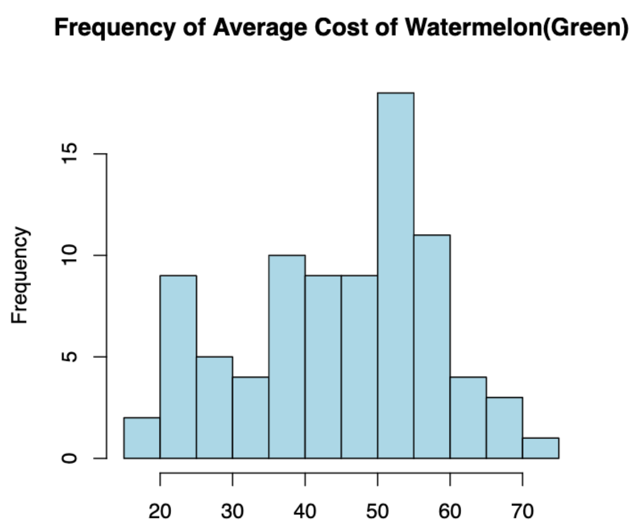
Plot of monthly watermelon sales prices in Nepal (Nepalese Rupee's) from 2013 to 2021

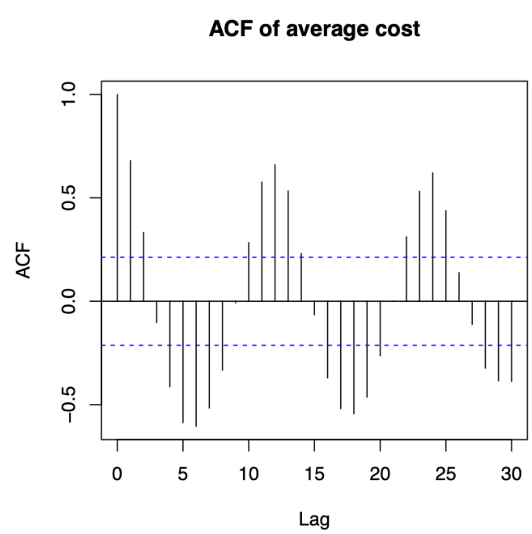




As you might notice above, the two plots above are the time series plot of the raw dataset and the testing dataset. I spliced the dataset so that the training dataset wouldn't contain the period of COVID-19 and only the sales price of watermelon before that period. What's important to note is that the regression and mean line shows that there is non-constant variance and relatively non-constant mean, but as for the mean, this mean only increases slightly. This indicates that our data is non-stationary, so we will likely need to perform further transformations to our data. Another thing to note that there is seasonality in our data which seems to occur yearly, as seen in the raw data with the constant rises towards the middle of the year and then followed by a drop. As I noted previously, although not drastic, there is a shaper increase in price in 2020 and beyond relative to all the other previous seasons.

<u>Histogram and ACF plots of trained data</u>

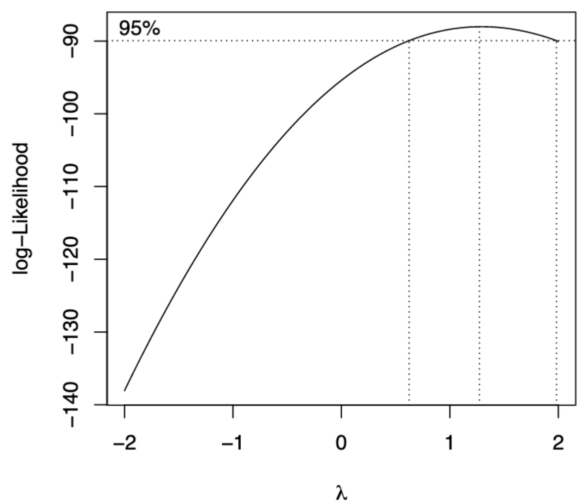




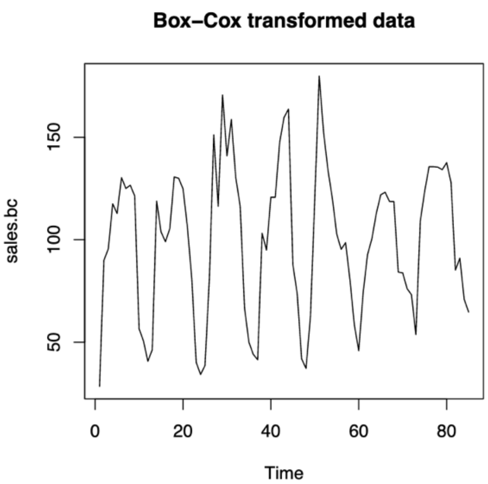
In this case, we will want to examine the ACF and histogram of our trained data to better understand any preliminary details of our data. By looking at the histogram, although it looks relatively normal, it's pretty obvious that there is some left-tailed skew in the data distribution.

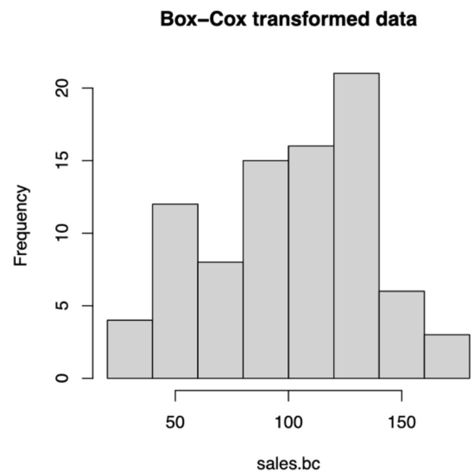
And the ACF reveals a periodic nature of our autocorrelations to fluctuate. Therefore, to deal with the skewed data we will transform data to achieve normality and stable variance, and for the periodic autocorrelations, we will difference our data for seasonality.

Box-cox Transformation of data



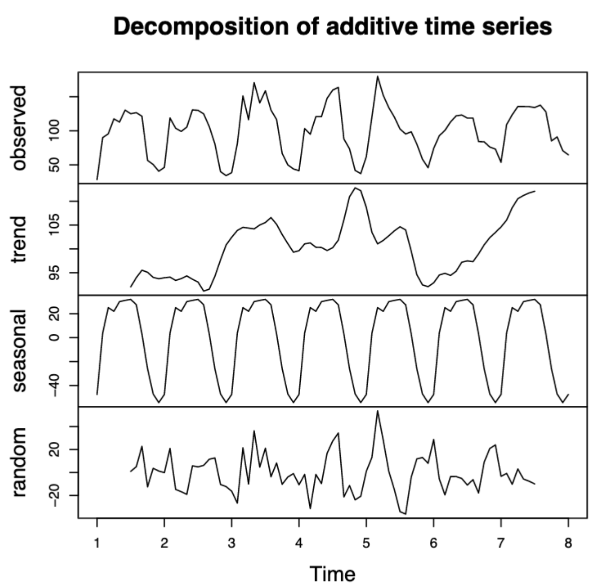
The first transformation that we will be performing is a box-cox transformation to provide us a lambda that might further reveal a potential better transformation to use instead. In our case, we receive a lambda value of 1.272727 which indicates that we might be able to use no transformation since a lambda of 1 would indicate no transformation, but since we want to reduce variance and achieve some normality, we will stick with the box-cox transformation. Below is the histogram and plots of our data following the box-cox transformation.





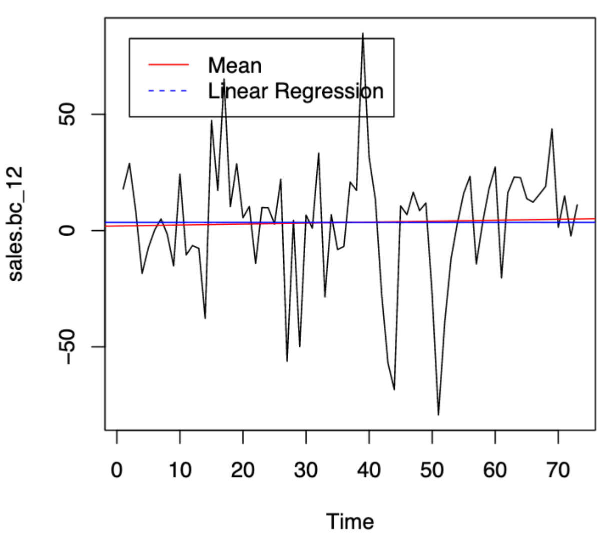
Compared to the original plot, although the data doesn't seem to change too much in histogram and plot, there are some clear differences in which the transformed data seems much more centered around the mean and the variance of the plot seems to have stabilized much more, especially for the data prior to the time 60. In order to see the effectiveness of our transformation, we can check the decomposition of our transformed training data.

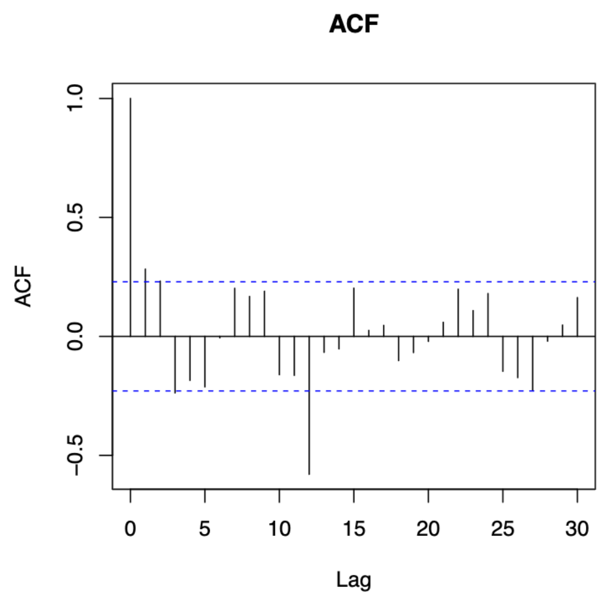
Decomposition of data

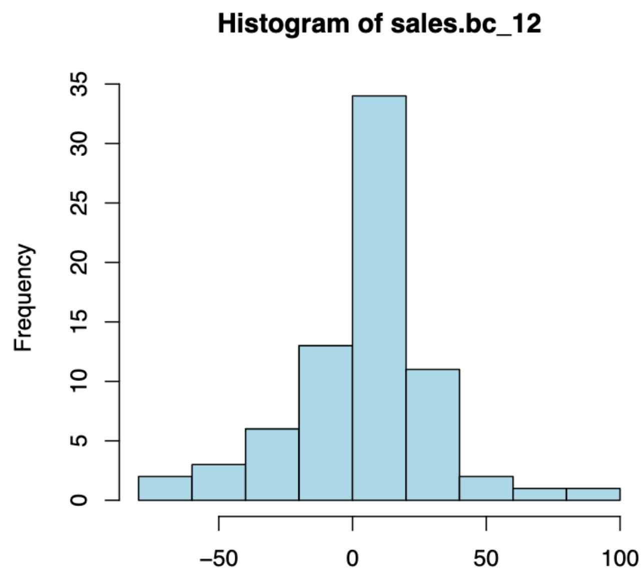


In the decomposition graph, noticeably there is not clear trend in our data. Although one might be able to conclude a slight positive trend, the trend isn't all too significant to perform any differencing on trend. What is noticeable is the seasonality aspect of our data, there is a clear yearly seasonality to our data, which is obvious as watermelon is typically a seasonal fruit which is most popular during the summer periods. When removing seasonality, we see that we are left with a series that resembles white noise as we can see in the random plot. Ultimately, this concludes that our transformation was indeed effective and that we will have to further difference for yearly seasonality to optimize our data in the modeling process.

Seasonal differencing

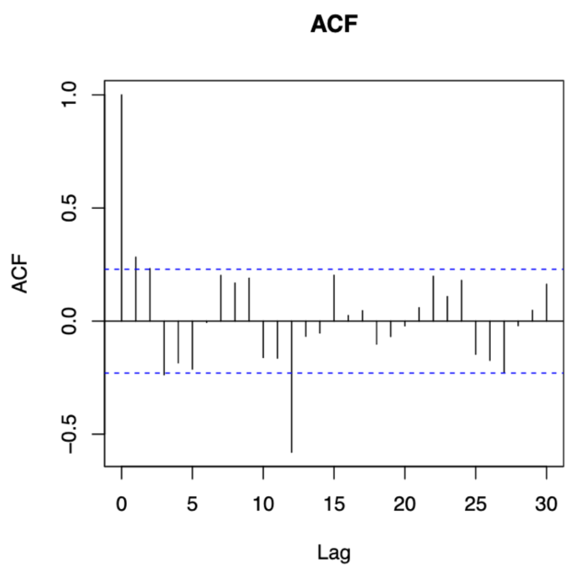


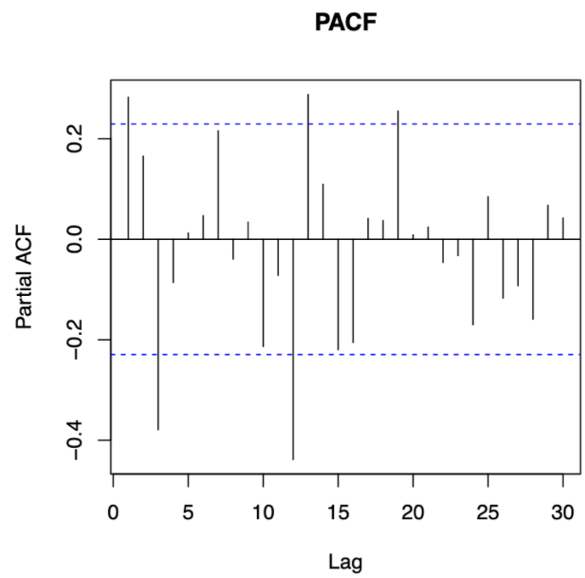




By differencing at lag 12, we have removed seasonality from the plot while reducing the variance. Notice that there is still no trend in our training data and that the ACF plot has significantly improved as there is no longer a periodic aspect to our autocorrelations. Additionally, there is no slow decay of our ACF plot and no trend in our difference data plot which indicates that our data is stationary at this point. Further looking at the histogram of our transformed and differenced training data for yearly seasonality 12 reveals an extremely normally distributed data, therefore we can proceed to modeling.

Creating models





Because we differenced the data and both the ACF and PACF graphs are significantly non-zero at some points, it is likely that we will be attempting to fit a SARIMA model. Observing the ACF graphs, we can see it is outside the confidence intervals at lags 1, 2, and 3 (q=1,2,3). There is also a significant lag at 12 (Q=1). As for the PACF graph, there are significant non-zero lags at lag 1 and 3 (p = 1,3), while also containing another significant point at lag 12, 13, and 19 (P = 1). Since we differenced only for seasonality then we will have no trend differencing (d = 0) but a seasonality difference (D=1) to account for no non-seasonal differencing and one seasonal differencing.

The first model that I attempted fitting was a full SARIMA model with the seasonal and non-seasonal AR and MA terms. The model below performed the best by a slight margin using the lowest AICc as the metric. This model will be indicated as model A, or SARIMA(3,0,1)(1,1,1)$_{12}$.

Model A

```
arima(x = sales.bc, order = c(3, 0, 1), seasonal = list(order = c(1, 1, 1),
    period = 12), method = "ML")

Coefficients:
         ar1     ar2      ar3      ma1      sar1     sma1
      0.7039  0.1819  -0.4045  -0.3276  -0.3137  -0.6277
s.e.  0.2341  0.1579   0.1134   0.2339   0.1666   0.2328

sigma^2 estimated as 280:  log likelihood = -315.31,  aic = 644.62
```

I attempted to fit permutations of q = 1,2,3 with p=1,3 while keeping Q=1, P=1, d=0, and D=1 constant among all models. Therefore, the next model, model B, that I fit was a SARIMA(1,0,3)$(1,1,1)_{12}$ model. Although the models weren't too different, it was best to change some of these factors and maintained the parameters that worked best. Below is the printed output of this fitted model.

Model B

```
arima(x = sales.bc, order = c(1, 0, 3), seasonal = list(order = c(1, 1, 1),
    period = 12), method = "ML")

Coefficients:
         ar1     ma1     ma2     ma3      sar1     sma1
      0.0944  0.2945  0.4622  0.0981  -0.2836  -0.6597
s.e.  0.4717  0.4613  0.1827  0.2480   0.1638   0.2310

sigma^2 estimated as 290.2:  log likelihood = -316.82,  aic = 647.63
```

Noticeably, although both models would perform relatively proficiently since the AICc only is separated by ~3 score, we can perform diagnostic checking further to see which model would perform better in other metrics of residuals as well. But in terms of AICc, model A is definitely more proficient.

Model Formulas: ($\nabla 1 \nabla 12\ ln\ (Ut\ ) = Dt$)

A: $D_t(1 - 0.3137_{(0.1666)}B^{12})(1 + 0.7039_{(.2341)}B + 0.1819_{(.1579)}B^2 - 0.4045_{(.1134)}B^3) = (1- 0.3276_{(.2339)}B)(1 - 0.6597_{(0.2328)}B^{12})Zt$, $\hat{\sigma}_z{}^2 = 280$

B: $D_t(1 - 0.2836_{(0.1638)}B^{12})(1 + 0.0944_{(.4717)}\mathrm{B}) = (1 + 0.2945_{(.46133)}B + 0.4622_{(.1827)}B^2 - 0.0981_{(.2480)}B^3)(1 - 0.6597_{(0.2310)}B^{12})Zt$, $\hat{\sigma}_z{}^2 = 290.2$

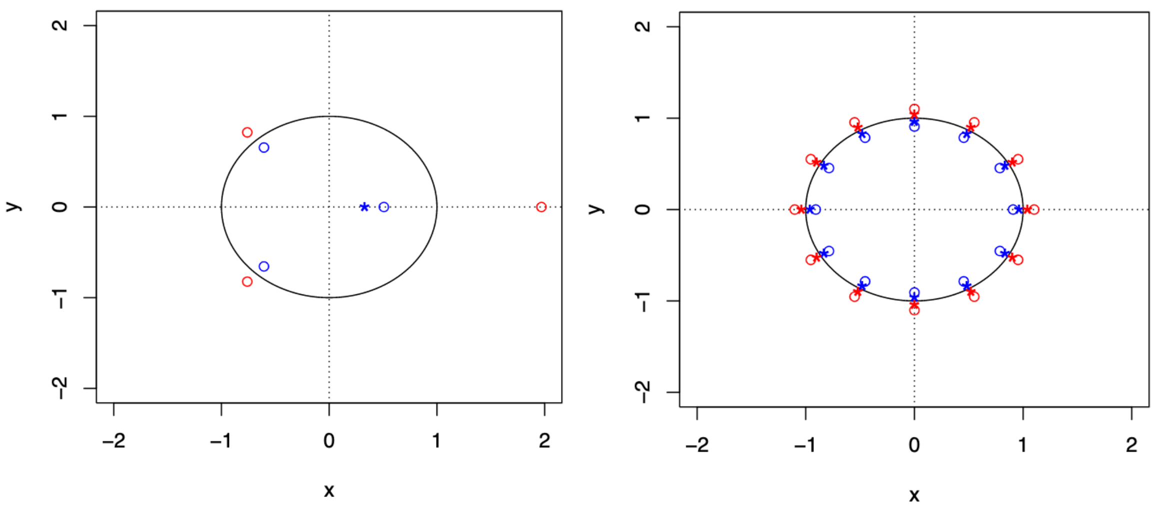
Model Equations:

A: $D_t = -0.7039B_{t-1} - 0.1819B_{t-2} + 0.4045B_{t-3} + 0.3137B_{t-12} + 0.2208B_{t-13} + 0.05706B_{t-14} - 0.12689B_{t-15} + Z_t + 0.3276Z_{t-1} + 0.6597Z_{t-12} + 0.21612Z_{t-13}$

B: $D_t = -0.0944D_{t-1} + 0.2836B_{t-12} + 0.02677B_{t-13} + Z_t + 0.2945Z_{t-1} + 0.4622Z_{t-2} - 0.0981Z_{t-3} - 0.6597Z_{t-12} - 0.19428Z_{t-13} - 0.30491Z_{t-14} - 0.06462Z_{t-15}$

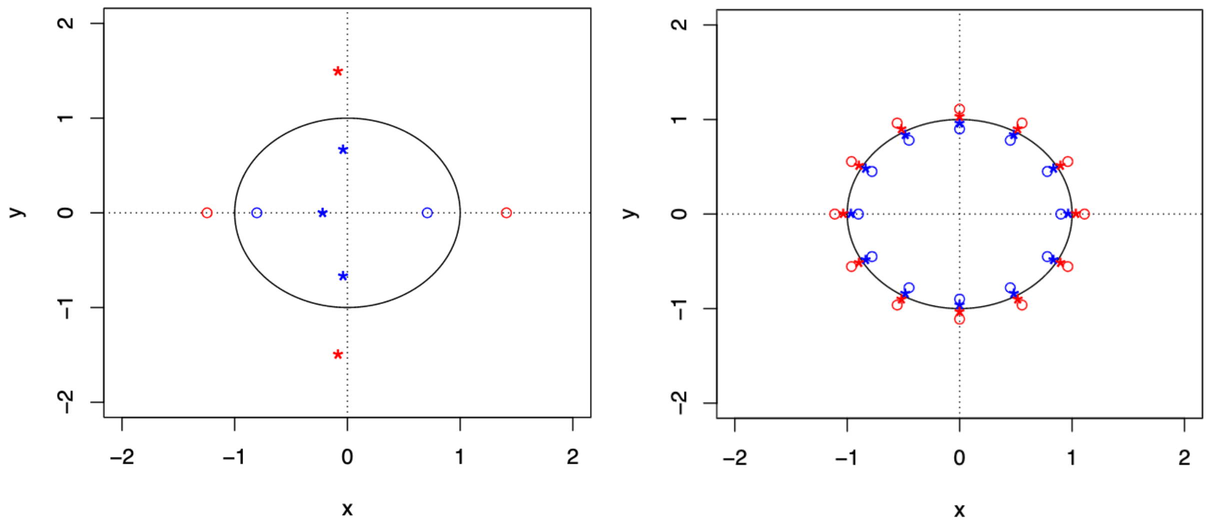
Plotting Roots

Model A Roots



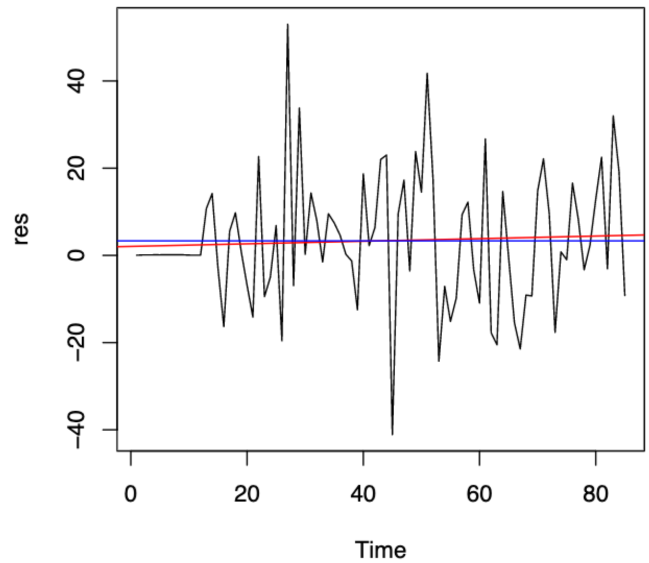
Both model A and B are models that contain MA and AR parameters, so we must check the seasonal AR roots which is displayed as the red circles and stars represent the roots for the MA aspect. Because our roots are all outside the unit circle, we can see that our model is invertible and stationary.

Model B Roots

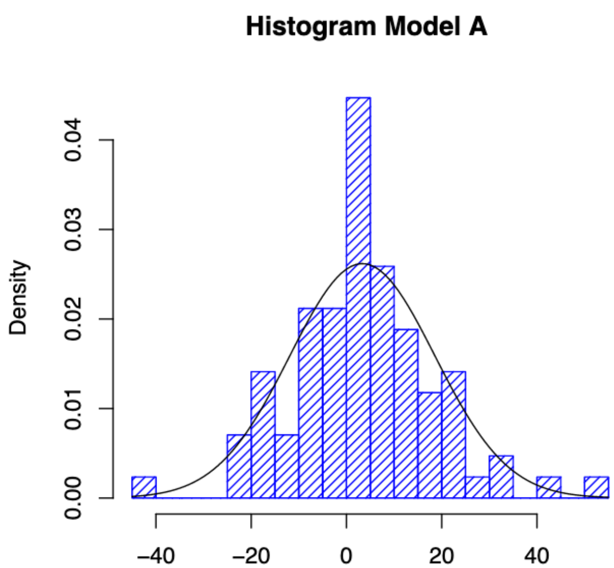


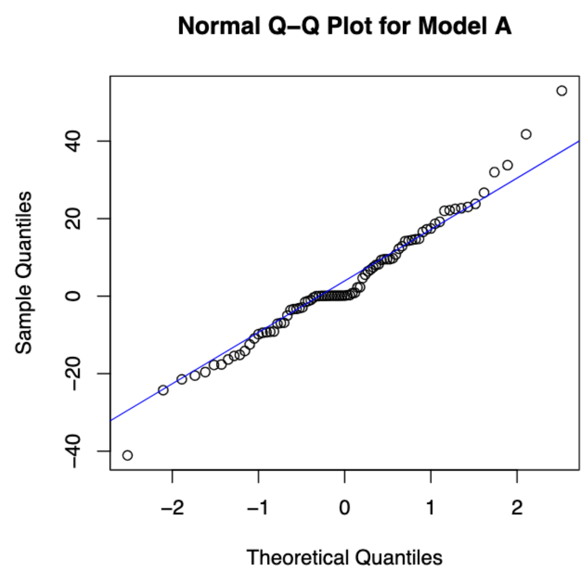
Similar to model A, model B has similar parameters which contain the same MA and AR components. Therefore, we can look at the non-seasonal and seasonal graphic in the same way, where we can notice, again, that all of the roots are outside of the unit circle. Therefore, we can conclude that our model B is also invertible and stationary. Because both of our models are stationary and invertible, then we can proceed with diagnostic checking of the model.

Model A diagnostic check



The graph above plots the residuals as a result of model A. Noticeably, there doesn't seem to be any trend, change in variance, seasonality, and also a mean of approximately 0. Therefore, we can now just check for normality of the residuals using various methods.





Noticeably, the histogram follows a relatively strong normal distribution centered around the mean ~0. Additionally, although the Q-Q plot does seem to contain some outliers that we also can see in our histogram, there does seem to be a powerful indication by the rest of the points following the line strongly indicating that our data is relatively normal. To ensure this assumption though, we will perform more robust, sound methods of normality and residual diagnostics. This method will be through the Portmanteau tests which include: the Shapiro-Wilk, Box-Pierce, Box-Ljung, and Mc-Leod Li tests, to check for the white noise, residual, and normality assumptions.

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97901, p-value = 0.1808
```

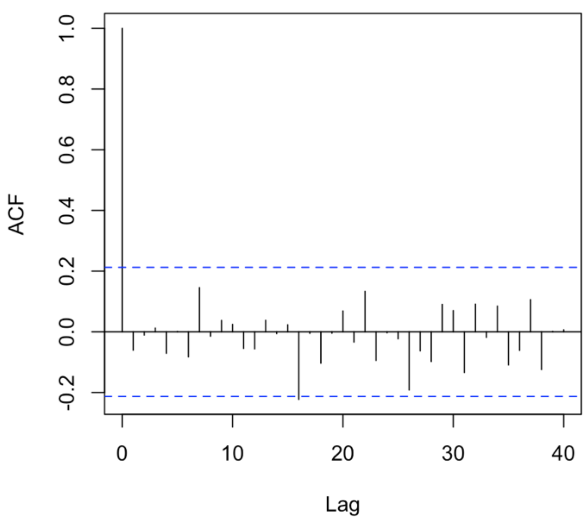
```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 3.3342, df = 6, p-value = 0.7659
```

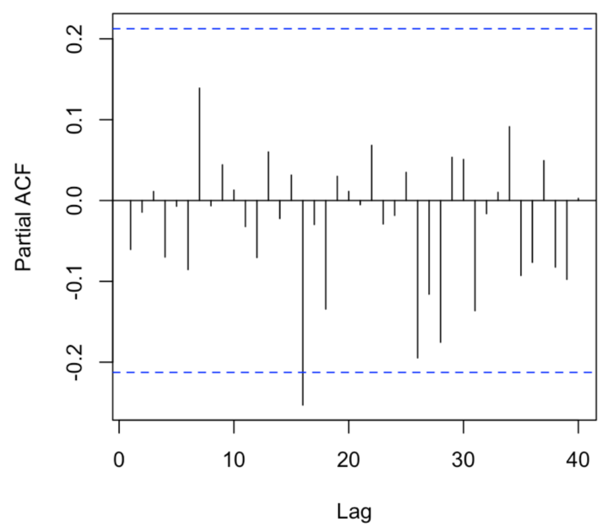
```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 3.6729, df = 6, p-value = 0.7208
```

```
##
##  Mc-Leod Li
##
## data:  res^2
## X-squared = 9.5847, df = 10, p-value = 0.4776
```

```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  232.1
```

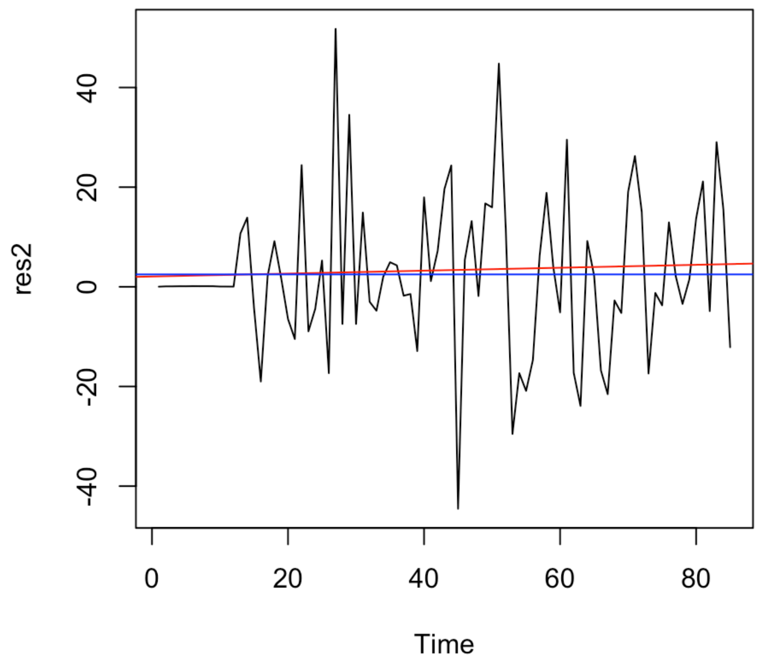
Because of our tests p-values are above an alpha level of 0.05, this means we fail to reject the white noise process. So, we would then proceed to applying the Yule-Walker estimate to residuals. Because the residuals were fitted to AR(0), or the white noise model, we will likely have white noise. Finally, our last diagnostics will be to examine the ACF and PACFs of the resulting model.



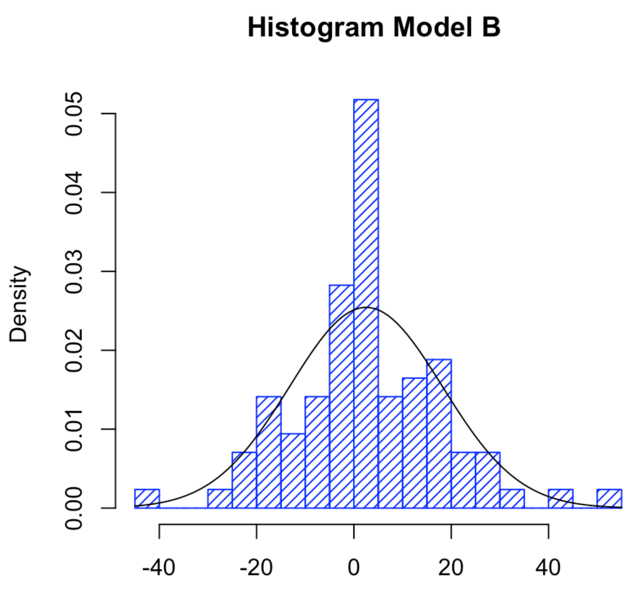


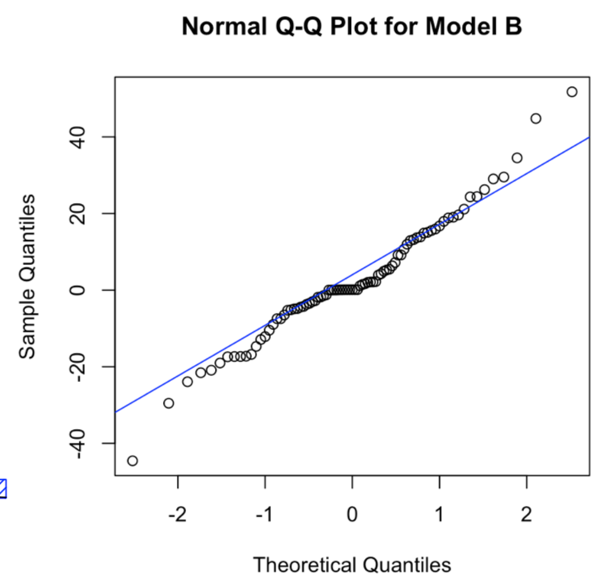
The ACF and PACF of the model A residuals display a stark majority of the lags within the confidence interval, while there are some lags that extend beyond, specifically lag 17 in the ACF and PACF, but due to Bartlett's formula, we can justify counting them within the confidence intervals. All of our diagnostic checks passed for model A meaning the residuals resemble a Gaussian WN(0,1) process.

Model B diagnostic check



Similar to the graph of model A, there doesn’t seem to be any significant or noticeable trend, change in variance, or seasonality. Due to the similar plots, this also indicates that there will likely be similar diagnostic results compared to model A, which is what we plan on expecting.



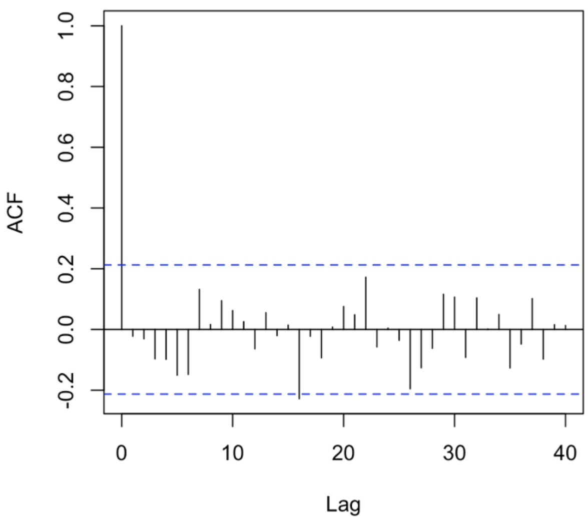


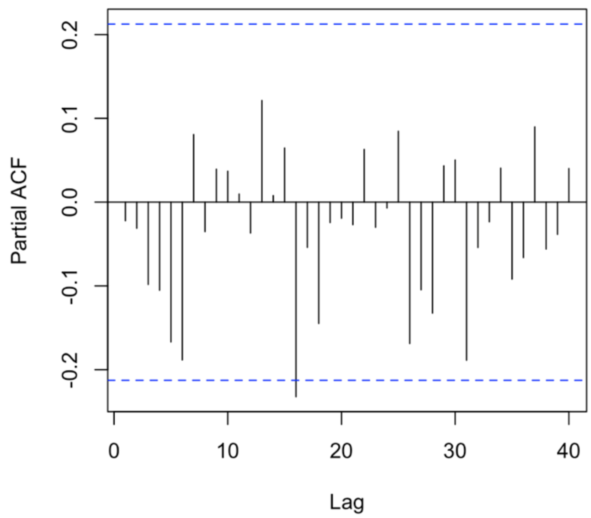
Again, our residuals seem approximately normal with a high peak with some minor outlier values as you could see from the -40 and +40 bins. This is seen in the tail ends of the Q-Q plot which mostly follows the diagonal line indicating normality, with the exception of those few tail outliers. Again, we will proceed with the Portmanteau tests to provide a more robust metric rather than simply visual.

```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.97347, p-value = 0.07676


##
##  Box-Pierce test
##
## data:  res2
## X-squared = 8.1319, df = 6, p-value = 0.2286


##
##  Box-Ljung test
##
## data:  res2
## X-squared = 8.9416, df = 6, p-value = 0.1769


##
##  Mc-Leod Li
##
## data:  res2^2
## X-squared = 7.5253, df = 10, p-value = 0.6751
```

```
##
## Call:
## ar(x = res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  245.9
```

As expected, all our tests result in a p-value above 0.5, although the p-values are much closer to 0.05 than the p-values from model A. This means that we can fail to reject the null hypothesis of white noise and can further apply the Yule-Walker estimate on this model as well. Since the order selected is 0 then that means the Yule-Walker estimate tests has also passed. Therefore, we can proceed to the ACF and PACF plots of the residuals to find if there are any significant/notable lags in the plots.



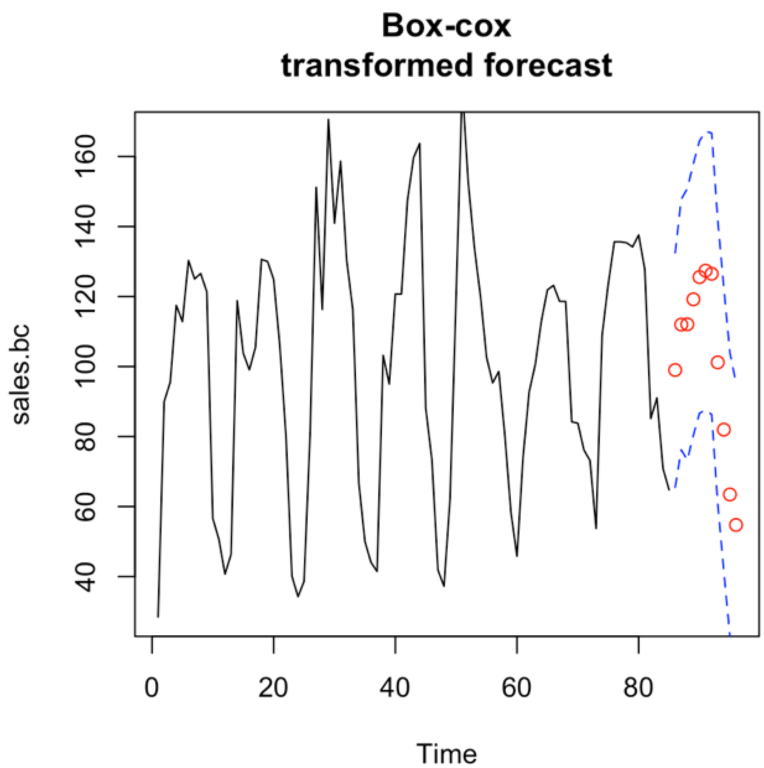


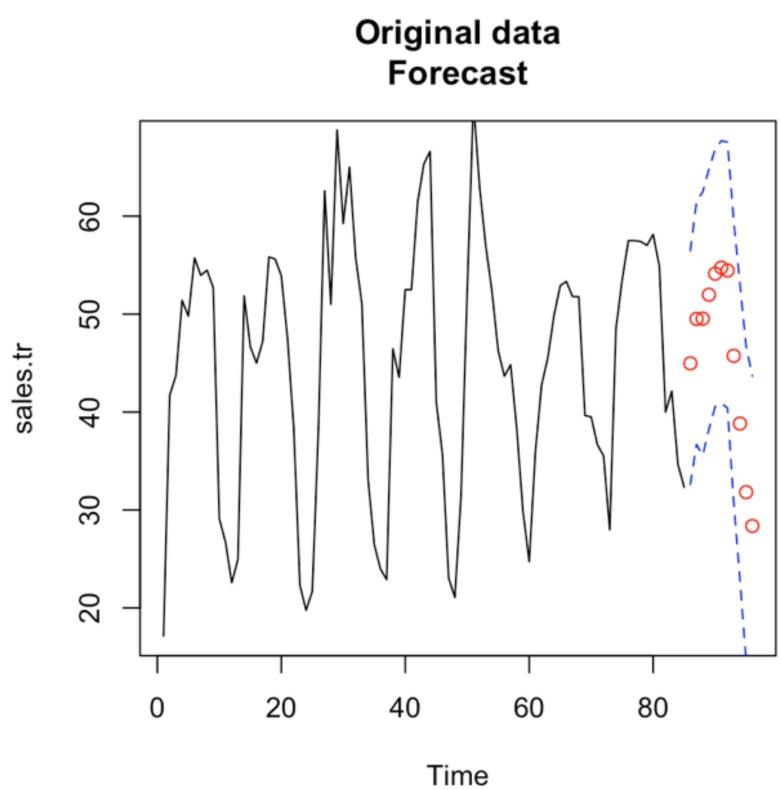
Once again, the ACF and PACF plots contain almost entirely statistically insignificant lags at all levels except for lags 17 in the ACF and PACF plots again. Regardless, this can be written off in the same means as model A's significant lag 17. Because model B has passed all tests, then we can proceed with choosing an appropriate model to forecast our testing data.

Forecasting model decision

In terms of performance, both model A and B had very similar results on all diagnostic tests. For model A, the p-value was higher for all tests except for the Mc-Leod Li test. Ultimately, this may indicate to us that model A is ultimately more sufficient, especially if we go off the original metric of AICc, even though the difference was initially only very minor. But with the new diagnostic, it's clear that model A will definitely be the better model for various other reasons.

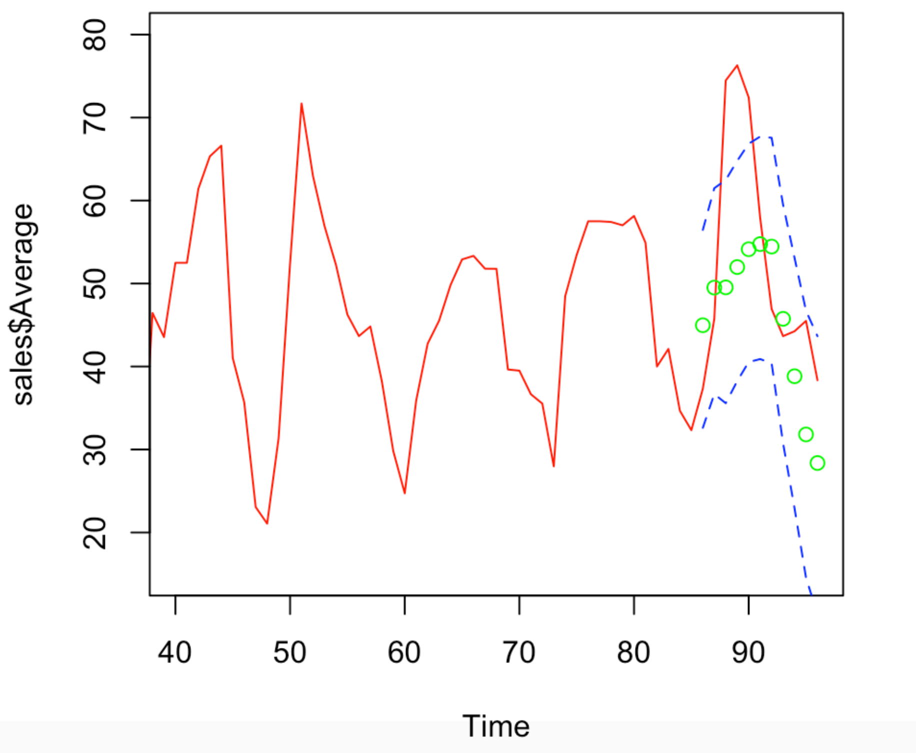
Forecasting





In both of these graphics, these plots display the forecasting of the potential points in both the box-cox transformed values and the scale of the original data. In the blue lines is the 95% confidence interval range, while the red dots are the 11 predictions for the 11 testing set data values that we stored to the side. Initially, it seems like the predictions would be valid under the patterns that we are observing prior, but to test this, we need to put these predictions next to the actual testing values.

<u>Final forecast with all values</u>



For the sale of clarity, I zoomed up the window to only contain a zoomed time frame to further examine the accuracy of our prediction. After taking a closer look on the true testing dataset added to the end, we can see that our prediction was inaccurate. Noticeably, our prediction couldn't capture the upper limit of the real data, as the highest peak of the data seemed to go outside the 95% confidence interval. Ultimately, beyond the first two points, none of the other predictions ever directly sat on the red line, indicating that none of the values were ever accurate predictions of the true values.

Conclusion

The final model used to forecast watermelon sales price for Nepal for the year of 2020-2021 was a SARIMA(3,0,1)(1,1,1)$_{12}$ model given by the equation:

$D_t = -0.7039B_{t-1} - 0.1819B_{t-2} + 0.4045B_{t-3} + 0.3137B_{t-12} + 0.2208B_{t-13} + 0.05706B_{t-14} - 0.12689B_{t-15} + Z_t + 0.3276Z_{t-1} + 0.6597Z_{t-12} + 0.21612Z_{t-13}$

Ultimately, the actual sales price of watermelon wasn't entirely captured within my 95% confidence interval as the lower parts were captured, the prices as it led into the summer months and out of the summer months, but at the peak prices, my confidence interval was incapable of capturing that stark increase relative to the previous years. Because my model was only able to predict 2 out of the 11 predictions, and those only being at the beginning of the year, I will conclude that my model was insufficient in predicting the sales price of watermelon and that it wasn't able to account for the monetary effects of COVID-19 on the market. Although it may be impractical and inaccurate for predictive accuracy, I do believe the model does a sufficient just in examining the underlying effects of COVID-19 particularly in how it might affect markets and restaurants as a result of the increase in watermelon prices that deviated from the norm.

I believe that although this model shouldn't be used for forecasting predictions, I do believe that many markets would benefit from the results of this model because it was maybe potentially allowing places to potentially compensate for this increase in watermelon prices by opting to another menu item that isn't as severely affected by COVID-19. Potentially a more stable and widely used crop would be able to replace these items, such as apples and bananas, which are typically not seasonally and more widely distributed and popular year-round. Additionally, due to the robust market of apples and banana's, there might be potentially less monetary impact as a result of COVID-19, but that would have to be another time series analysis to confirm so.

References

Feldman, R. (2023). Pstat 174: Time Series. Class, UC Santa Barbara

nischal lal shrestha_2022, Kalimati Tarkari Dataset: Fruits,Vegetables Price,https://www.kaggle.com/dsv/3493502, 10.34740/KAGGLE/DSV/3493502, Kaggle, Nischal Lal Shrestha, 2022

Appendix

```
library(lubridate)

library(dplyr)

library(MASS)

library(ggplot2)

library(ggfortify)

library(forecast)

library(MuMIn)

library(readr)

library(xts)

library(astsa)

library(tidyr)

# Initial Graph

## Watermelon Sales

sales_ts <- ts(sales$"Average",start=c(2013,1),frequency=12)

## Plot

plot.ts(sales_ts,xlab = "Year",ylab='Average Price',main = "Raw Data")

##checks variance

var(sales_ts) # checking variance to see if its high


# Split, Testing and Training

##split into training and testing

sales.tr <- sales_ts[1:85] #splitting to only contain the dates before covid-19 dates
```

```
sales.tst <- sales_ts[86:96]


# Graph Training
fit <- lm(sales_ts ~ as.numeric(1:length(sales_ts)))
plot.ts(sales.tr, main = "Training Data")
abline(fit, col="red") # this is the regression line
abline(h=mean(sales_ts), col="blue") # this is the mean line
legend(x="topright", inset = 0.01,legend=c("Linear Regression", "Mean"),
 col=c("red", "blue"), lty=1:2, cex=0.5)


hist(sales.tr, col="light blue", xlab="", main="Frequency of Average Cost of
Watermelon(Green)")
acf(sales.tr,lag.max=30, main="ACF of average cost")


# Box-Cox Transformation
shapiro.test(sales.tr) # shapiro test to see how normal model is currently
bcTransform <- boxcox(sales.tr~ as.numeric(1:length(sales.tr)))
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda
sales.bc = (1/lambda)*((sales.tr^lambda)-1)
plot.ts(sales.bc, main ="Box-Cox transformed data")
hist(sales.bc, main ="Box-Cox transformed data")
shapiro.test(sales.bc) # shapiro test to see how much it improves later
```

```
# Box-cox decomp
y <- ts(as.ts(sales.bc), frequency = 12)
decomp <- decompose(y)
plot(decomp)


# Differencing for seasonality at lag 12
var(sales.bc)
sales.bc_12 <- diff(sales.bc, lag=12) # diff only at lag 12
plot.ts(sales.bc_12, main="ln(U_t) differenced at lag 12")
var(sales.bc_12)
fit <- lm(sales.bc_12 ~ as.numeric(1:length(sales.bc_12)))
abline(fit, col="red")
abline(h=mean(sales.bc_12), col="blue")
legend(x="topleft", inset = 0.05,legend=c("Mean", "Linear Regression"),
 col=c("red", "blue"), lty=1:2, cex=1)
acf(sales.bc_12, lag.max=30,main="ACF") # max lag to account for potential lag of 24
pacf(sales.bc_12, lag.max=30,main="PACF")
hist(sales.bc_12, col="light blue", xlab="")
var(sales.bc_12)


# Fitting
##Q = 1
```

```
##q = 1,3

## P = 1

## p = 1,2,3

fit1 <- arima(sales.bc, order=c(3,0,1), seasonal = list(order = c(1,1,1), period = 12),
method="ML") # testing two permutations of the model

AICc(fit1)

fit2 <- arima(sales.bc, order=c(1,0,3), seasonal = list(order = c(1,1,1), period = 12),
method="ML")

AICc(fit2)


# Model A

fit1


plot.roots(polyroot(c(1,0,0,0,0,0,0,0,0,0,0,0,-0.3137)),polyroot(c(1,0,0,0,0,0,0,0,0,0,0,0,-
0.6277)),main="MA roots Model A(seasonal)")

plot.roots(polyroot(c(1,0.7039,0.1819,-0.4045)),polyroot(c(1,-0.3276)), main="MA roots Model
A(non-seasonal)") # polyroots achieved from the model


# Model B

fit2

plot.roots(polyroot(c(1,0,0,0,0,0,0,0,0,0,0,0,-0.2836)),polyroot(c(1,0,0,0,0,0,0,0,0,0,0,0,-
0.6597)), main="MA & AR roots Model B(seasonal)")
```

```
plot.roots(polyroot(c(1,0.0944,-0.5691)),polyroot(c(1,0.2945,0.4622,0.0981)), main="MA roots
Model B(non-seasonal)")


# Diagnostic checking for model A
##Histogram
res<-residuals(fit1) # this is the residual of model 1
hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE,main="Histogram Model A ")
m <- mean(res) # mean
std <- sqrt(var(res)) # standard deviation
curve( dnorm(x,m,std), add=TRUE )
##plot
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
##qqplot
qqnorm(res,main= "Normal Q-Q Plot for Model A") # test for normality
qqline(res,col="blue")
##acf/pacf
acf(res, lag.max=40)
pacf(res, lag.max=40)
##Portmanteau Tests
shapiro.test(res)
Box.test(res, lag = 10, type = c("Box-Pierce"), fitdf = 4)
```

```
Box.test(res, lag = 10, type = c("Ljung-Box"), fitdf = 4)

Box.test(res^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)

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


#Diagnostic checking for model B

#Histogram

res2<-residuals(fit2)

hist(res2,density=20,breaks=20, col="blue", xlab="", prob=TRUE,main="Histogram Model B ")

m2 <- mean(res2)

std2 <- sqrt(var(res2))

curve( dnorm(x,m2,std2), add=TRUE )

#plot

plot.ts(res2)

fitt2 <- lm(res2 ~ as.numeric(1:length(res2))); abline(fitt, col="red")

abline(h=mean(res2), col="blue")

#qqplot

qqnorm(res2,main= "Normal Q-Q Plot for Model B")

qqline(res2,col="blue")

#acf/pacf

acf(res2, lag.max=40)

pacf(res2, lag.max=40)

#Portmanteau Tests

shapiro.test(res2)
```

```
Box.test(res2, lag = 10, type = c("Box-Pierce"), fitdf = 4)

Box.test(res2, lag = 10, type = c("Ljung-Box"), fitdf = 4)

Box.test(res2^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)

ar(res2, aic = TRUE, order.max = NULL, method = c("yule-walker"))


# Forecasting using model A

forecast(fit1)

pred.tr <- predict(fit1, n.ahead = 11)

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

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

ts.plot(sales.bc, xlim=c(1,length(sales.bc)+11), ylim = c(min(sales.bc),max(U.tr)), main = "Box-
cox
transformed forecast")

lines(U.tr, col="blue", lty="dashed")

lines(L.tr, col="blue", lty="dashed")

points((length(sales.bc)+1):(length(sales.bc)+11), pred.tr$pred, col="red")


#with Original data

pred.orig <- InvBoxCox(pred.tr$pred, lambda)

U= InvBoxCox(U.tr,lambda)

L= InvBoxCox(L.tr,lambda)

ts.plot(sales.tr, xlim=c(1,length(sales.tr)+11), ylim = c(min(sales.tr),max(U)),main = "Original
data
```

```
Forecast")
lines(U, col="blue", lty="dashed") # adding the 95% conf
lines(L, col="blue", lty="dashed")
points((length(sales.tr)+1):(length(sales.tr)+11), pred.orig, col="red")
#zoom
ts.plot(sales.tr, xlim = c(40,length(sales.tr)+11), ylim = c(15,80)) # this is setting to a zoomed
interval
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(sales.tr)+1):(length(sales.tr)+11), pred.orig, col="red")
#add true values
ts.plot(sales$"Average", xlim = c(40,length(sales.tr)+11), ylim = c(15,80),col="red",main =
"Final
Forecast with tests values")
lines(U, col="blue", lty="dashed") # adding the 95% conf
lines(L, col="blue", lty="dashed")
points((length(sales.tr)+1):(length(sales.tr)+11), pred.orig, col="green") # these are the true
predictions
```