Monthly Sales of U.S. Houses (in Thousands) from 1965 to 1975

Anum Damani

June 4, 2021

Abstract

The data set that I used is monthly sales of U.S. houses (in thousands) from the year 1965 to 1975. This data is useful because it can be used to help predict the future. My goal for this project was to find the best model that captured the data well and use it to forecast monthly sales of U.S. houses in 1975. In order to do this, I used the Box-Jenkins methodology. The best model that I found indeed captured the data nicely during forecasting, which means that the Box-Jenkins methodology worked well on the data. I found that the first few months of 1975 were predicted very accurately. After the third month in 1975, the prediction was not exactly accurate but it was still successful. Overall, my goals for this project were achieved and I was able to successfully use the best model to predict monthly sales of U.S. houses (in thousands) in the year 1975.

Introduction

I want to forecast monthly sales of U.S. houses (in thousands) in the year 1975. The dataset I will analyze is called "Monthly sales of U.S. houses (thousands) 1965 – 1975." The dataset includes 132 observations, and I will be calling this data Y_t where $t = 1, 2, 3, ..., 132$. The source of the data set is "Abraham & Ledolter (1983)." This data was found using the TSDL package in R.[1] The software I used is R (RStudio). I believe that this data set is fascinating because I enjoy learning about the housing market and it would be interesting to analyze the behavior of monthly sales of U.S. houses over time. This data set is important because there was an economic recession in the U.S. during 1973-1975 and the Arab-Israeli War occurred in 1973, so it is valuable to analyze the impact of these events on the housing market during that time.

I did not need to apply transformations in order to get a stationary series. I differenced the data at lag 1 to eliminate trend and then at lag 12 to eliminate seasonality. Then, I analyzed the ACF and PACF to find that SAR(2)$_{12}$ (for the differenced data), SARIMA(0,1,0)(2,1,1)$_{12}$ (for the original data), and SMA(1)$_{12}$ (for the differenced data) would be good candidate models; later, I eliminated SARIMA(0,1,0)(2,1,1)$_{12}$ and SMA(1)$_{12}$ because these models had a unit root. In addition, from the ACF and PACF, I determined other possible choices for p's and q's. I found the most complex model, constructed 95% confidence intervals, set coefficients to zero if the confidence intervals contained zero, and modified this complex model. The modified model became SARIMA(1,0,0)(2,0,0)$_{12}$ (Model A).

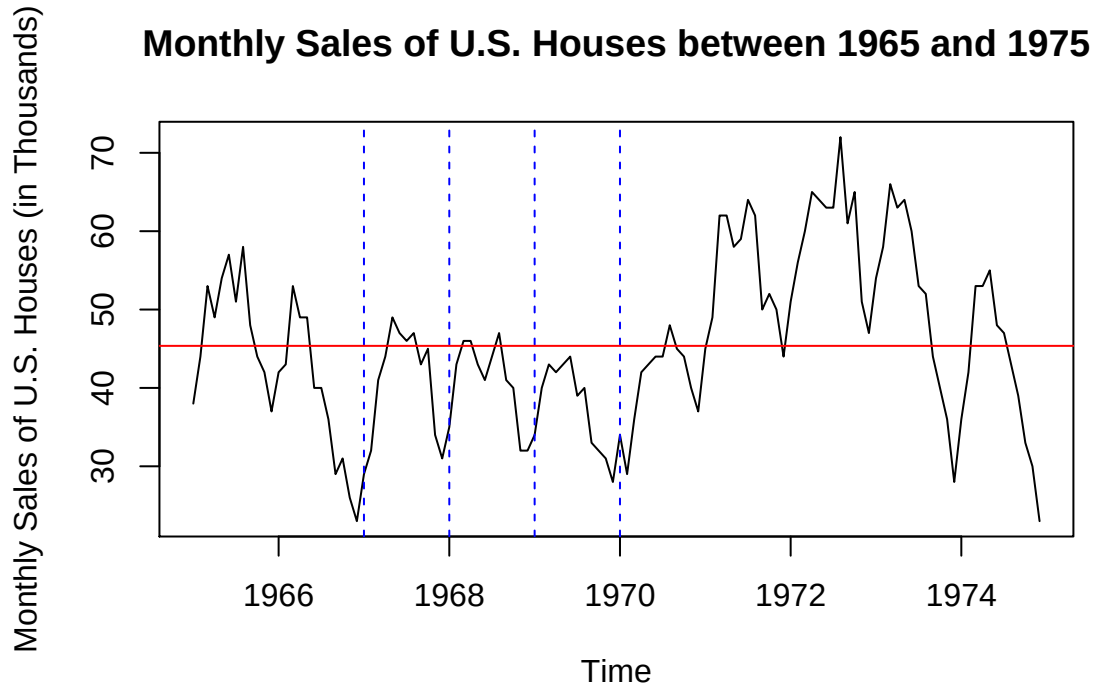
I compared AICc values of all the models that did not have unit roots, and found that SARIMA(1,0,0)(2,0,0)$_{12}$ (Model A) and SAR(2)$_{12}$ (Model B) had the lowest AICc values. Both Model A and Model B only had AR parts so they are invertible by construction. I found the roots of Model A and Model B and determined that both models are stationary. I performed diagnostic checking on Model A and Model B and found that Model A passes all diagnostic checking, while Model B does not pass diagnostic checking. Hence, Model A was used for forecasting.

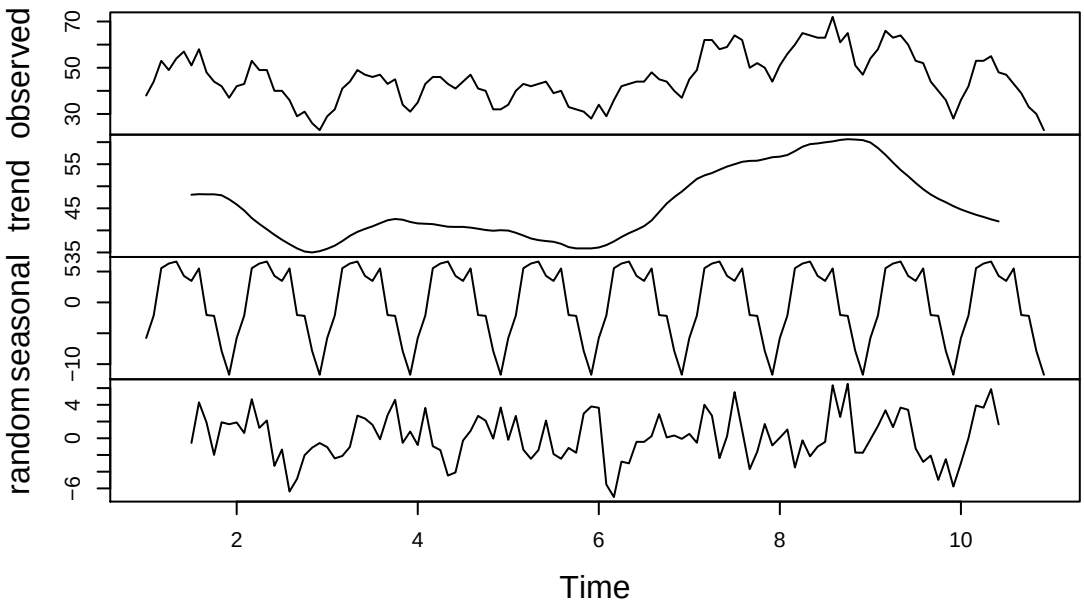
Using the SARIMA(1,1,0)(2,1,1)$_{12}$ model (Model A), I proceeded to the forecasting stage to forecast monthly sales of U.S. houses (in thousands) in the year 1975. Some months were predicted well because we can see visually that the true data points and predicted data points overlapped. However, after a few months, there was an underprediction. Despite this, the true data and predicted data points were still within confidence intervals. I think that the SARIMA(1,1,0)(2,1,1)$_{12}$ model successfully forecasted monthly sales of U.S. houses (in thousands) in the year 1975.

Section 1: Plot and Analyze Time Series

This is the time series data:

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1965  38  44  53  49  54  57  51  58  48  44  42  37
## 1966  42  43  53  49  49  40  40  36  29  31  26  23
## 1967  29  32  41  44  49  47  46  47  43  45  34  31
## 1968  35  43  46  46  43  41  44  47  41  40  32  32
## 1969  34  40  43  42  43  44  39  40  33  32  31  28
## 1970  34  29  36  42  43  44  44  48  45  44  40  37
## 1971  45  49  62  62  58  59  64  62  50  52  50  44
## 1972  51  56  60  65  64  63  63  72  61  65  51  47
## 1973  54  58  66  63  64  60  53  52  44  40  36  28
## 1974  36  42  53  53  55  48  47  43  39  33  30  23
## 1975  29  33  44  54  56  51  51  53  45  45  44  38
```



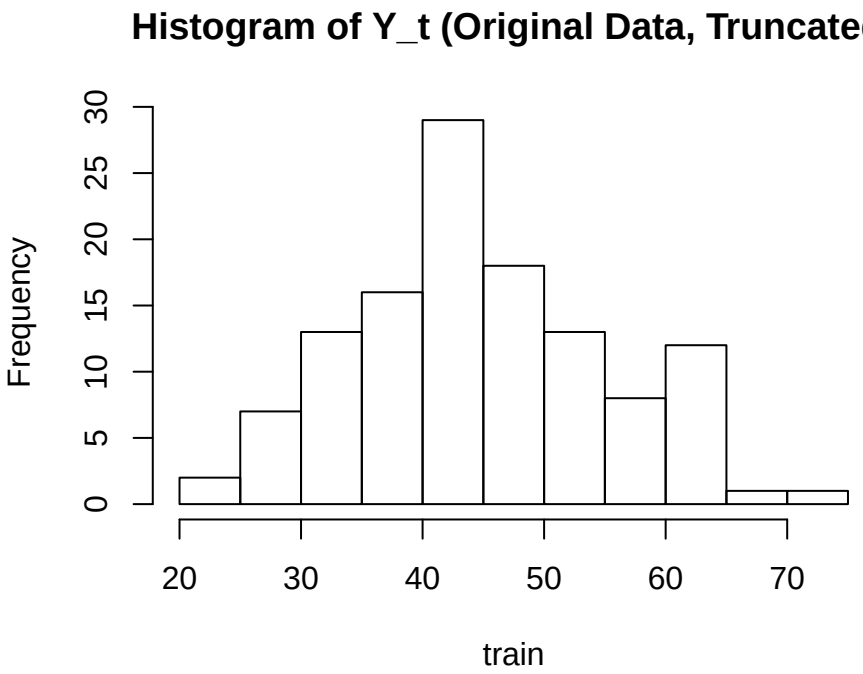


Top: Time series plot of Y_t (original data, truncated).

Bottom: Decomposition of additive time series.

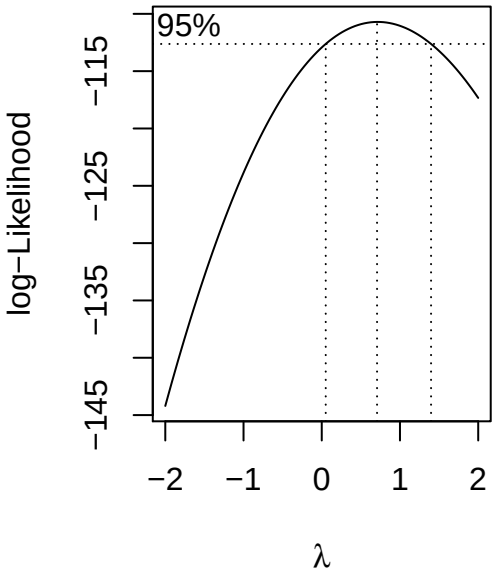
Above is the plot of the original time series, with "Time" on x-axis and "Monthly Sales of U.S. Houses (in Thousands)" on the y-axis. This plot of the time series shows monthly sales of U.S. houses between the years 1965 to 1975. I think there is some trend in the data, but it may not be linear and might be polynomial. The blue dotted lines on the plot above show the seasonal cycles of the Monthly Sales of U.S. Houses. The red line indicates the mean of the data. This time series looks choppy and it seems like there are sharp changes in behavior. This indicates that there may be variability in monthly sales of U.S. houses over time. By looking at the "Decomposition of additive time series" plot, we can clearly see the trend and seasonal component of the time series. Therefore, this data is non-stationary.

Section 2: Tranformations or Differencing?



Above: Histogram of Y_t (Original Data, Truncated).

According to the histogram above for Y_t (Original Data, Truncated), the histogram looks symmetric and approximately normally-distributed, and the variance seems even. So, a transformation is not needed to stabilize variance. But, I still tried to apply a boxcox transform to see if it would be necessary. This plot below shows why the data does not need a boxcox transform:

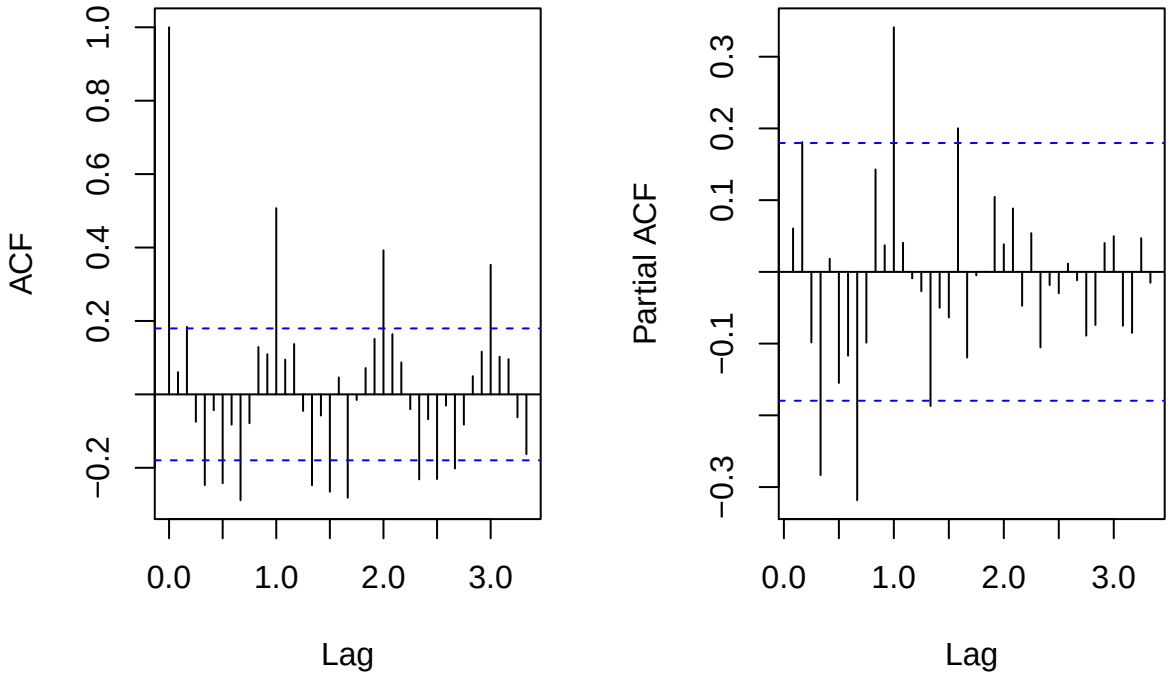


Above: Box-Cox Plot.

```
## [1] 0.7070707
```

Lambda is 0.7070707. Looking at the Box-Cox plot above, the confidence interval includes lambda = 1, which means that the data is already approximately normally-distributed and a Box-Cox transformation is not necessary.

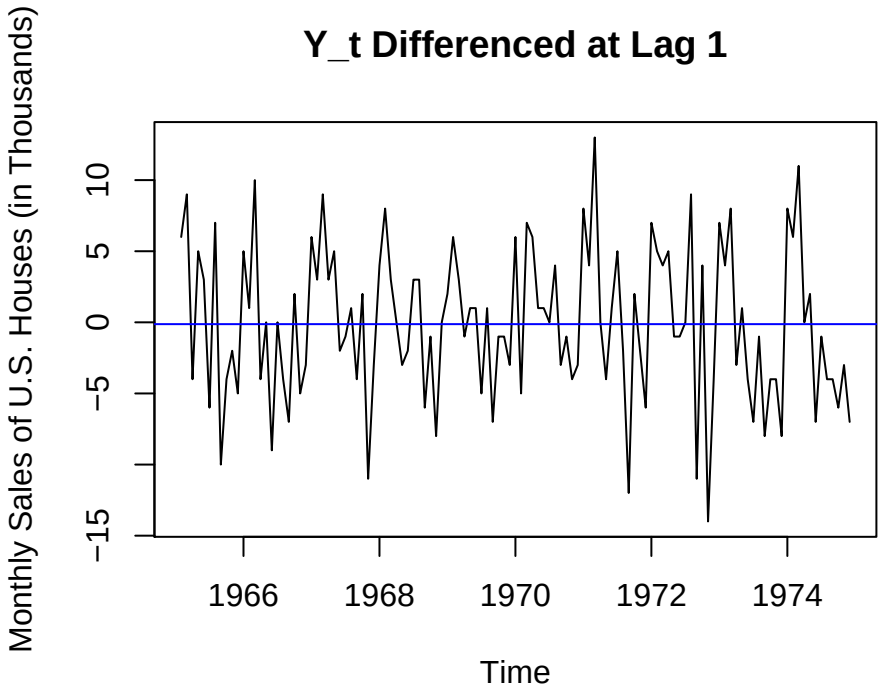
Now, since there is a time trend in the original data, it is necessary to apply differencing at lag 1 to remove the trend. So, I differenced the data at lag 1. I obtain the following ACF & PACF plots of Y_t differenced at lag 1:



Above (Left side): ACF of Y_t differenced at lag 1.

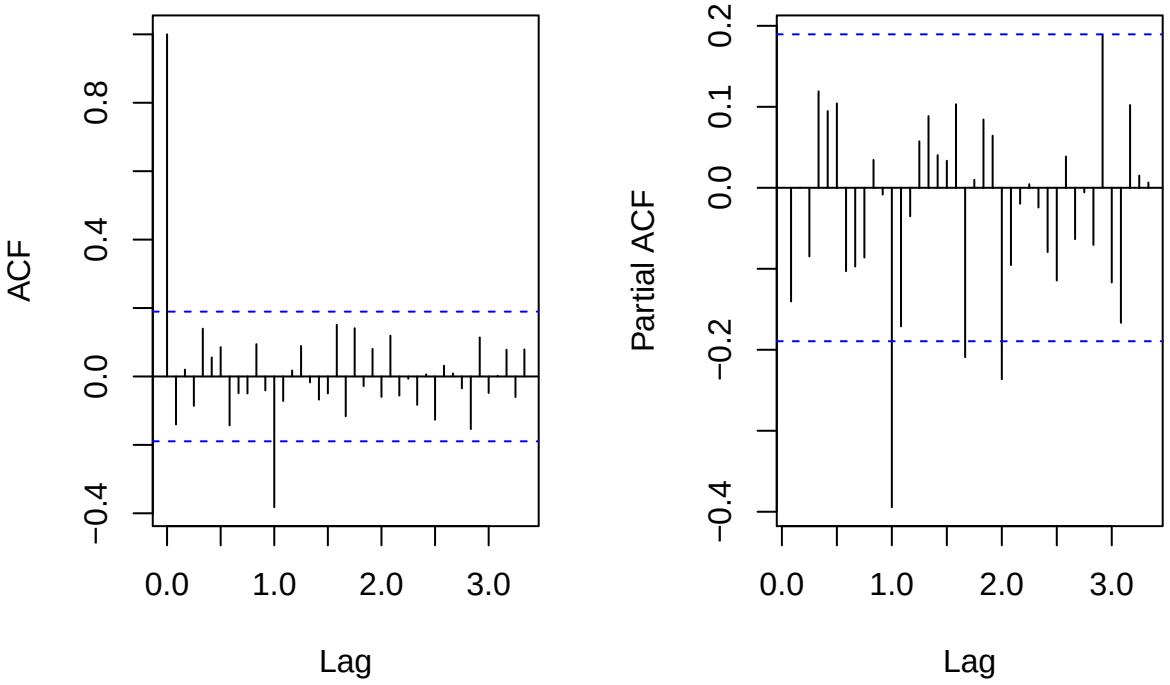
Above (Right side): PACF of Y_t differenced at lag 1.

Looking at the ACF and PACF plots of Y_t differenced at lag 1 above, it is clear that there is seasonality. We see that at lags 12, 24, and 36 there are peaks in the ACF. This is the time series plot of Y_t differenced at lag 1:



Above: Time series plot after differencing at lag 1.

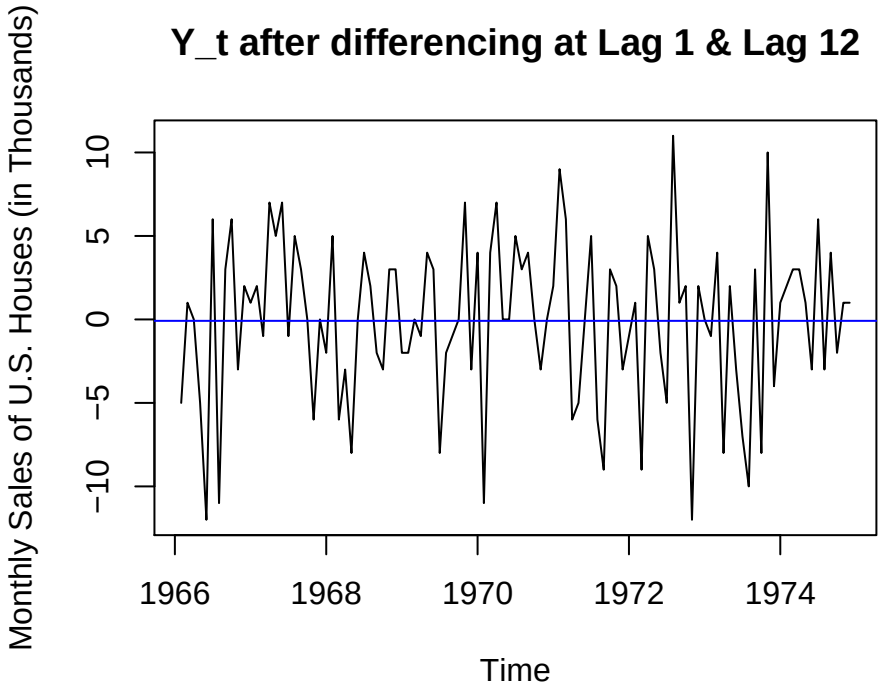
After observing the time series plot after differencing at lag 1 (pictured directly above), it is clear that the data looks a little more stationary now because there is no more trend. However, the ACF above revealed that there is seasonality, so it is necessary to also difference at lag 12 because this original time series has a seasonal component and it is monthly data. After differencing at lag 12, I obtain the following ACF and PACF plots:



Above (Left side): ACF of Y_t differenced at lags 1 & 12.

Above (Right side): PACF of Y_t differenced at lags 1 & 12.

Above, we have the ACF of Y_t differenced at lags 1 and 12 on the left side, and we have the PACF of Y_t differenced at lags 1 and 12 on the right side.



Above: Time series plot after differencing at lags 1 & 12.

By looking at the time series plot after differencing at lags 1 and 12, it is apparent that the data looks stationary. It appears as though there is no trend, no seasonality, and low variance.

The variance of the data that is differenced at lags 1 and 12 is 23.75701. In order to ensure that this data is indeed stationary, I differenced again at lag 1 to see whether the variance increases. Indeed, if I difference again at lag 1, the variance increases from 23.75701 to 54.45391. So, differencing d = 1 times at lag 1 to remove trend and differencing D = 1 times at lag 12 in order to remove seasonality is good enough to ensure that the data is stationary.

Here is a comparison of the variances:

```
## [1] 111.898
```

```
## [1] 28.73821
```

```
## [1] 23.75701
```

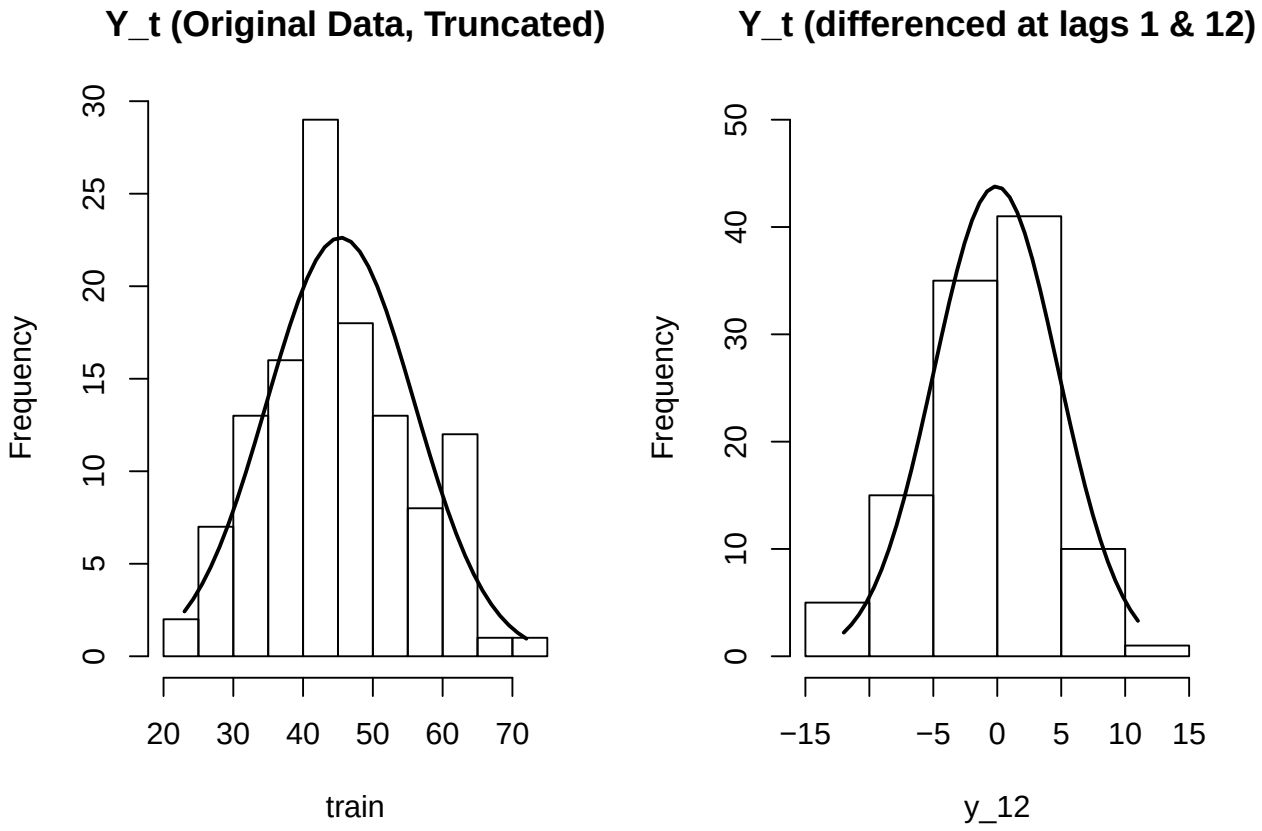
```
## [1] 54.45391
```

The variance of the original, truncated series Y_t is 111.898.

The variance after differencing Y_t at lag 1 is 28.73821.

The variance after differencing Y_t at lags 1 and 12 is 23.75701.

The variance after differencing Y_t at lag 1, lag 12, and again at lag 1 is 54.45391.



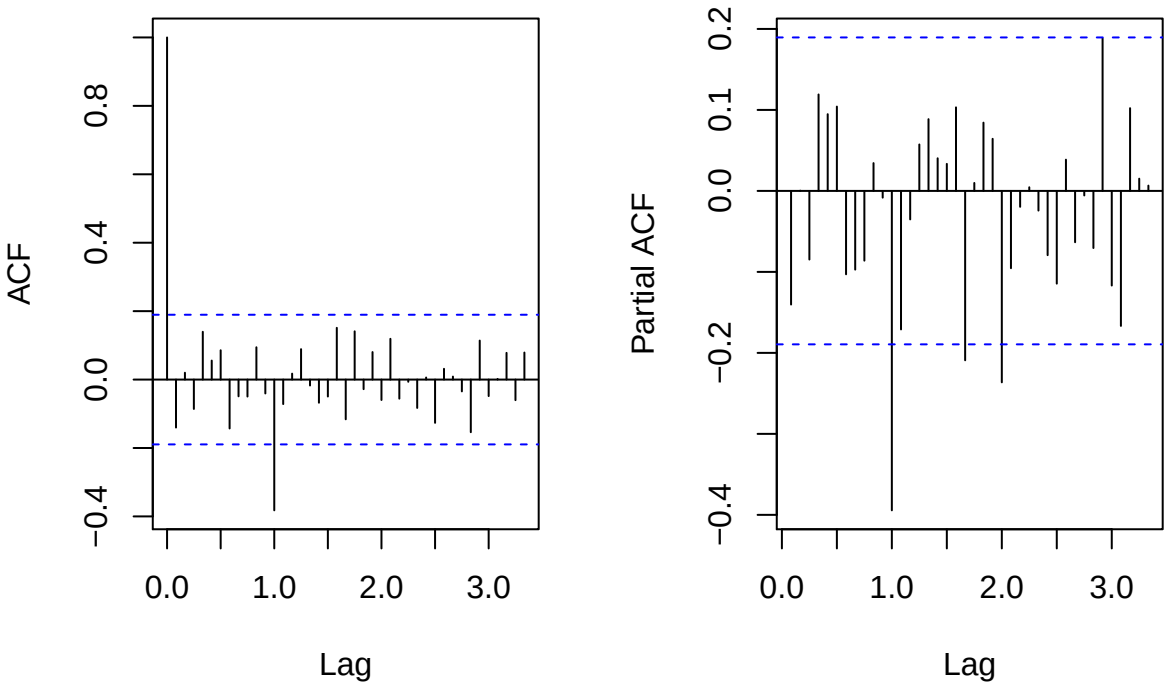
Above (Left side): Histogram of Y__t (Original data, truncated).

Above (Right side): Histogram of Y__t differenced at lags 1 & 12.

Above are the histograms of the truncated series Y_t and Y_t after differencing at lags 1 and 12. Looking at the histogram of Y_t, it seems like the data is somewhat Gaussian, but it could be better; it does not look very symmetric. Looking at the histogram of Y_t after differencing at lags 1 and 12, it seems that the data looks more Gaussian and normally-distributed and a little more symmetric in comparison to the original data, truncated (Y_t without differencing).

Section 3: Preliminary Identification of Models

Below are the ACF and PACF plots after differencing Y_t at Lags 1 & 12:



Above (Left side): ACF of Y_t differenced at lags 1 & 12.

Above (Right side): PACF of Y_t differenced at lags 1 & 12.

Looking at the ACF and PACF plots above for Y_t differenced at Lags 1 & 12, some suitable choices for p could be 0 or 1 or 8. I believe p could be 0 because in the PACF, between lag 1 and lag 12, there are no lags outside confidence intervals. I believe p could be 1 in case p = 0 does not result in a model that is complex enough. I thought p could be 8 because in the PACF, the 8th lag between lags 12 and 24 is outside confidence intervals. A suitable choice for q is 0 because looking at the ACF, there are no significant lags outside confidence intervals between lags 0 and 12; I do not believe that there could be other possible choices for q other than 0. Some suitable choices for P are 2 or 0 or 1. I thought P could be 2 because looking at the PACF, there are lags outside confidence intervals at lags 12 and 24 (at 2 lags). I considered P = 0 and P = 1 as possible choices for P in the case that P = 2 didn't work. Some suitable choices for Q are 0 or 1. I think Q could be 1 because Q = 1 because in the ACF, the last lag outside confidence intervals is at lag 12. I believed it was worth trying Q = 0 in case Q = 1 did not work. Since it is monthly data, $s = 12$.

By examining the ACF and PACF, I have preliminarily identified candidate models to be $\text{SAR}(2)_{12}$ (for the differenced data), $\text{SARIMA}(0,1,0)(2,1,1)_{12}$ (for the original data), and $\text{SMA}(1)_{12}$ (for the differenced data).

I have determined that $\text{SAR}(2)_{12}$ is a candidate model because if we look at the PACF, it is apparent that P = 2 because the last non-zero lag which is outside confidence intervals is at lag 24.

I believe that $\text{SARIMA}(0,1,0)(2,1,1)_{12}$ is a candidate model. I know that p = 0 because in the PACF, between lags 0 and 12, there is no significant lag that is outside confidence intervals. Also, d = 1 because I differenced the data at lag 1. In addition, q = 0 because looking at the ACF, there are no significant lags outside confidence intervals between lags 0 and 12. P = 2 because looking at the PACF, there are lags outside confidence intervals at lags 12 and 24. D = 1 because I differenced the data at lag 12 in order to remove seasonality. Lastly, Q = 1 because looking at the ACF, the last lag outside confidence intervals is at lag 12.

Finally, $\text{SMA}(1)_{12}$ is a candidate model because if we look at the ACF, the last non-zero lag that is outside confidence intervals is at lag 12.

Section 4: Fitting Models

In the previous section, the three models that I preliminarily identified using ACF/PACF are $SAR(2)_{12}$ (for the differenced data), $SARIMA(0,1,0)(2,1,1)_{12}$ (for the original data), and $SMA(1)_{12}$ (for the differenced data). Also, in the previous section, I determined some suitable choices for p, q, P, Q (as stated previously, p could be 0 or 1 or 8, q is 0, P could be 2 or 0 or 1, Q could be 0 or 1). I took the most complex model which is $SARIMA(8,0,0)(2,0,0)_{12}$ and constructed 95% confidence intervals for the coefficients. Then, I noticed that for the ar2, ar3, ar4, ar5, ar6, ar7, and ar8 coefficients, zero is contained in the confidence intervals. So, I set the ar2, ar3, ar4, ar5, ar6, ar7, and ar8 coefficients to zero, and then the model became $SARIMA(1,0,0)(2,0,0)_{12}$. Therefore, I modified the complex model $SARIMA(8,0,0)(2,0,0)_{12}$ so that it becomes $SARIMA(1,0,0)(2,0,0)_{12}$.

I also tried many different models with different combinations of capital and lowercase p's and q's such as: $SARIMA(1,0,0)(2,0,1)_{12}$, $SARIMA(1,0,0)(2,0,0)_{12}$, $SARIMA(1,0,0)(0,0,0)_{12}$, $SARIMA(1,0,0)(1,0,0)_{12}$, $SARIMA(8,0,0)(1,0,0)_{12}$, $SARIMA(8,0,0)(0,0,0)_{12}$, $SARIMA(8,0,0)(0,0,1)_{12}$, $SARIMA(1,0,0)(0,0,1)_{12}$.

Most of my models did not work because they have a unit root. The models that do not have a unit root and have the two lowest AICc are $SARIMA(1,0,0)(2,0,0)_{12}$ (Model A) and $SAR(2)_{12}$ (Model B).

I have estimated the coefficients for Model A and Model B using ML estimation, and I also found the AICc values:

```
## Loading required package: minpack.lm
```

```
## Loading required package: rgl
```

```
## Loading required package: robustbase
```

```
## Loading required package: Matrix
```

```
##
## Call:
## arima(x = y_12, order = c(1, 0, 0), seasonal = list(order = c(2, 0, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1     sar1     sar2  intercept
##       -0.2558  -0.6500  -0.3554    -0.0740
## s.e.   0.0963   0.0987   0.1007     0.1656
##
## sigma^2 estimated as 15.71:  log likelihood = -302.39,  aic = 614.78
```

```
## [1] "AICc for SARIMA(1,0,0)(2,0,0)_{12} (MODEL A) (for the differenced data):"
```

```
## [1] 615.1695
```

```
##
## Call:
## arima(x = y_12, order = c(0, 0, 0), seasonal = list(order = c(2, 0, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##          sar1     sar2  intercept
```

```
##       -0.5905  -0.3654    -0.0767
## s.e.   0.0987   0.1022     0.2200
##
## sigma^2 estimated as 16.81:  log likelihood = -305.76,  aic = 619.53
```

```
## [1] "AICc for SAR(2)_{12} (MODEL B) (for the differenced data):"
```

```
## [1] 619.7619
```

Top Output: Estimating Coefficients & Finding AICc for Model A.

Bottom Output: Estimating Coefficients & Finding AICc for Model B.

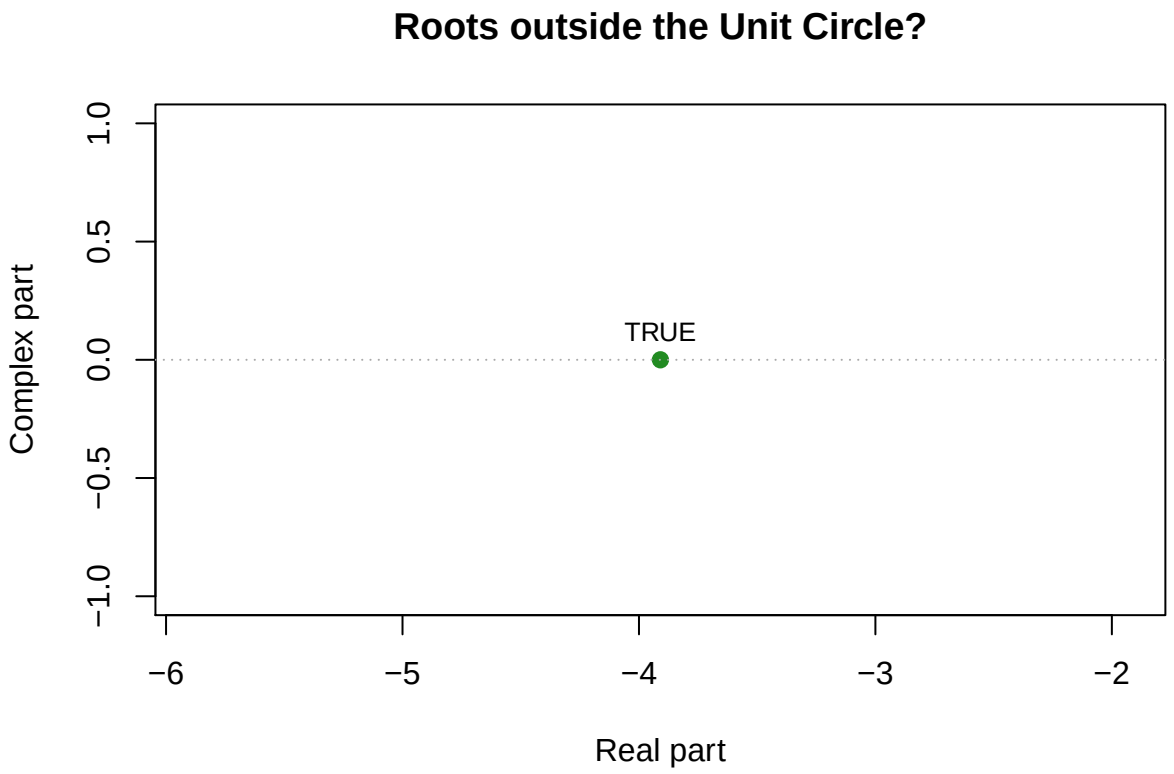
In the output above, we see that for SARIMA$(1,0,0)(2,0,0)_{12}$ (Model A), the coefficient ar1 is estimated to be -0.2558, the coefficient sar1 is estimated to be -0.6500 and the coefficient sar2 is estimated to be -0.3554. The 95% confidence interval for the coefficient ar1 is $(-0.2558 - 1.96(0.0963), -0.2558 + 1.96(0.0963))$ which simplifies to become $(-0.444548, -0.067052)$. The 95% confidence interval for the coefficient sar1 is $(-0.6500 - 1.96(0.0987), -0.6500 + 1.96(0.0987))$ which simplifies to become $(-0.843452, -0.456548)$. The 95% confidence interval for the coefficient sar2 is $(-0.3554 - 1.96(0.1007), -0.3554 + 1.96(0.1007))$ which simplifies to become $(-0.552772, -0.158028)$. Note that none of these confidence intervals contain zero.

For SAR$(2)_{12}$ (Model B), the coefficient sar1 is estimated to be -0.5905 and the coefficient sar2 is estimated to be -0.3654. The 95% confidence interval for the coefficient sar1 is $(-0.5905 - 1.96(0.0987), -0.5905 + 1.96(0.0987))$ which simplifies to become $(-0.783952, -0.397048)$. The 95% confidence interval for the coefficient sar2 is $(-0.3654 - 1.96(0.1022), -0.3654 + 1.96(0.1022))$ which simplifies to become $(-0.565712, -0.165088)$. Note that none of these confidence intervals contain zero.

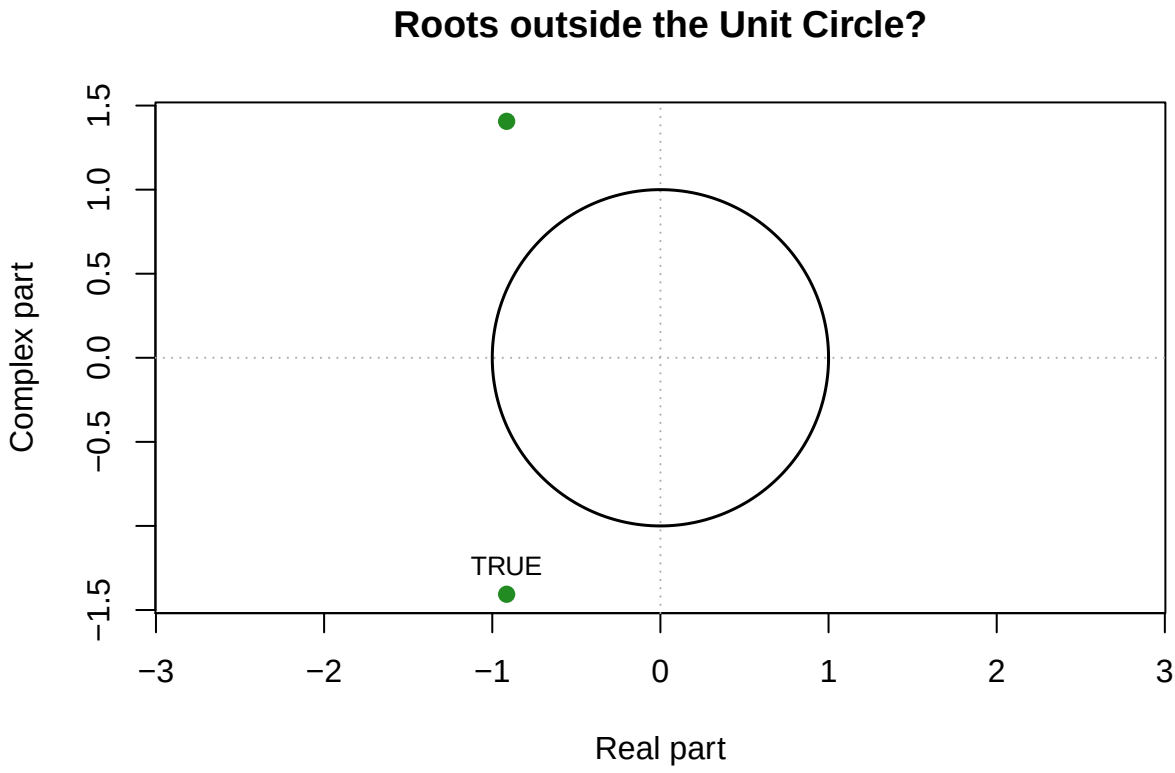
The model with the lowest AICc value of 615.1695 is SARIMA$(1,0,0)(2,0,0)_{12}$ (Model A). The model with the second lowest AICc value of 619.7619 is SAR$(2)_{12}$ (Model B). I can check for unit roots using the uc.check function in R now.

For the AR part (non-seasonal) of Model A, it is pure AR(1) with a coefficient of -0.2558; the absolute value of -0.2558 is 0.2558, which is strictly smaller than 1. I can check Model A (SARIMA$(1,0,0)(2,0,0)_{12}$) officially in R to see if there are any unit roots for the AR part:

```
##        real complex outside
## 1 -3.909304       0    TRUE
## *Results are rounded to 6 digits.
```




```
##        real   complex outside
## 1 -0.914463  1.406232    TRUE
## 2 -0.914463 -1.406232    TRUE
## *Results are rounded to 6 digits.
```



Top Plot: Showing/Checking the roots of the AR part (non-seasonal) of Model A.

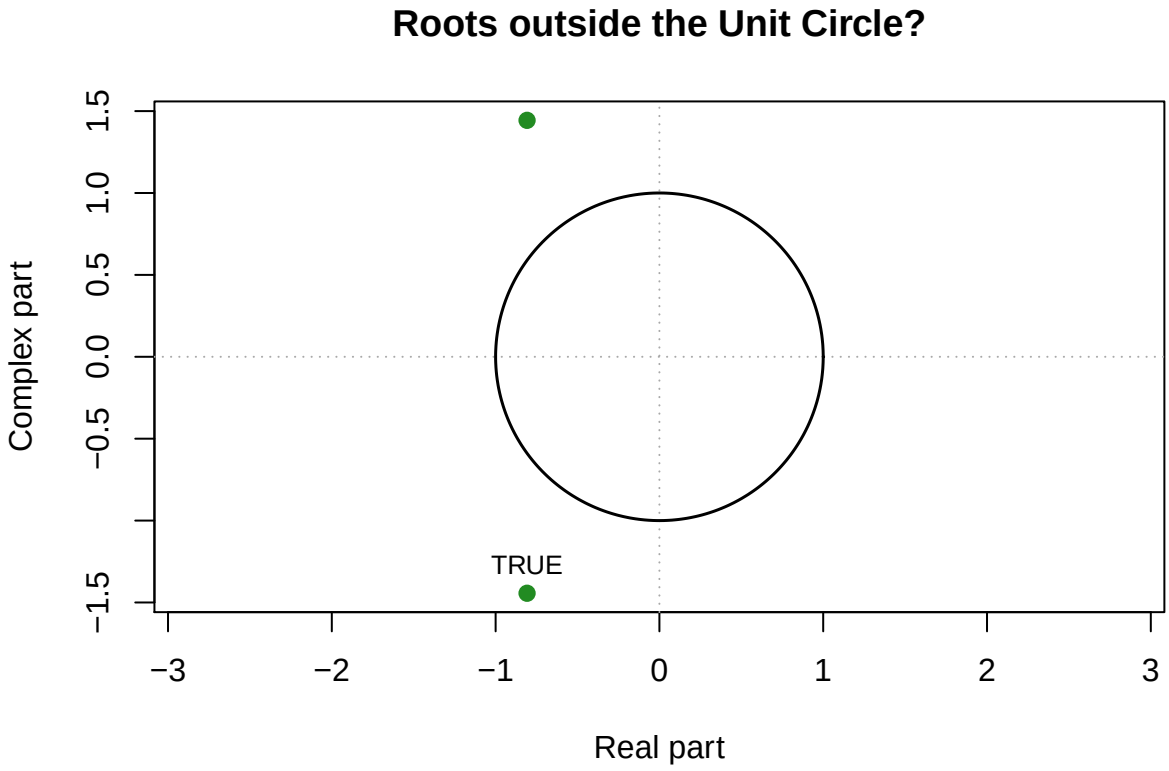
Bottom Plot: Checking the roots of the AR part (seasonal) of Model A.

In Model A, there is no MA part and when I estimated the coefficients for this model, we could see that all of the coefficients are AR coefficients. So it is apparent that there is only an AR part in Model A. Since AR models are always invertible by construction, checking for unit roots is helpful in order to determine if this model is stationary.

For the AR part (non-seasonal), the characteristic polynomial is $\theta(z) = 1 + 0.2558z$. For the AR part (seasonal), the characteristic polynomial is $\theta(z) = 1 + 0.6500z + 0.3554z^2$. As we can see from above, the root of the non-seasonal AR part is outside the unit circle and the roots of the seasonal AR part are outside the unit circle. Since the roots of the non-seasonal and seasonal polynomials lie outside the unit circle, the Model A (SARIMA(1,0,0)(2,0,0)$_{12}$) has MA(∞) representation; therefore, SARIMA(1,0,0)(2,0,0)$_{12}$ is stationary. Hence, SARIMA(1,0,0)(2,0,0)$_{12}$ is stationary and invertible.

I can check SAR(2)$_{12}$ (Model B) to see if there are any unit roots:

```
##        real   complex outside
## 1 -0.808019  1.443549    TRUE
## 2 -0.808019 -1.443549    TRUE
## *Results are rounded to 6 digits.
```



Above Plot: Showing/Checking the roots of Model B.

It is clear that there is only a seasonal AR part in Model B. Model B is SAR(2)$_{12}$, so $P = 2$ and the other coefficients are zero. When I estimated the coefficients for Model B, all of the coefficients were seen to be seasonal AR coefficients. The characteristic polynomial for Model B is $\theta(z) = 1 + 0.5905z + 0.3654z^2$. AR models are always invertible. It is clear from the output above that all of the roots of Model B are outside the unit circle. Therefore, SAR(2)$_{12}$ (Model B) has MA(∞) representation, and is therefore, stationary. Hence, SAR(2)$_{12}$ (Model B) is stationary and invertible.

I found the fitted model equation for SARIMA(1,0,0)(2,0,0)$_{12}$ (Model A) (note that I am using d = 1 and D = 1 because I used y_12 which is the time series differenced once at lag 1 and once at lag 12):

Since p = 1: $\phi(B) = (1 - \phi_1 B)$

Since d = 1: $(1 - B)$

Since q = 0: $\theta(B) = 1$

Since P = 2: $\Phi(B) = (1 - \Phi_1 B^{12} - \Phi_2 B^{24})$

Since D = 1: $(1 - B^s)^D = (1 - B^{12})^1$

Since Q = 0: $\Theta(B) = 1$

Also, s = 12.

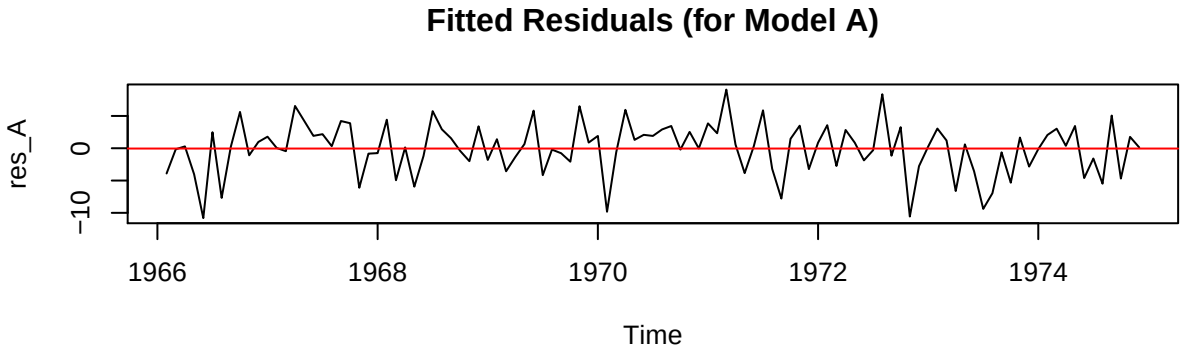
Therefore, the fitted model in algebraic form for Model A is: $(1 - \phi_1 B)(1 - \Phi_1 B^{12} - \Phi_2 B^{24})(1 - B^{12})(1 - B)X_t = Z_t$. After plugging in the coefficients that I estimated, the fitted model equation for Model A is $(1 + 0.2558_{(0.0963)}B)(1 + 0.6500_{(0.0987)}B^{12} + 0.3554_{(0.1007)}B^{24})(1 - B^{12})(1 - B)X_t = Z_t$; the subscripts correspond to the standard error of the coefficients. Also, $\hat{\sigma}_z^2 = 15.71$.

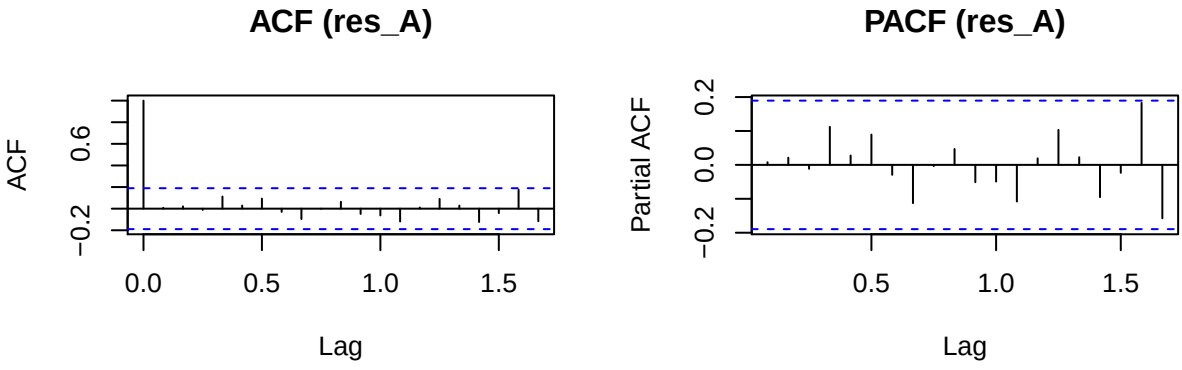
Now, I can do diagnostic checking on SARIMA(1,0,0)(2,0,0)$_{12}$ (Model A):

```
## [1] -0.04586404
```

```
## [1] 15.85168
```

```
## [1] 3.981417
```





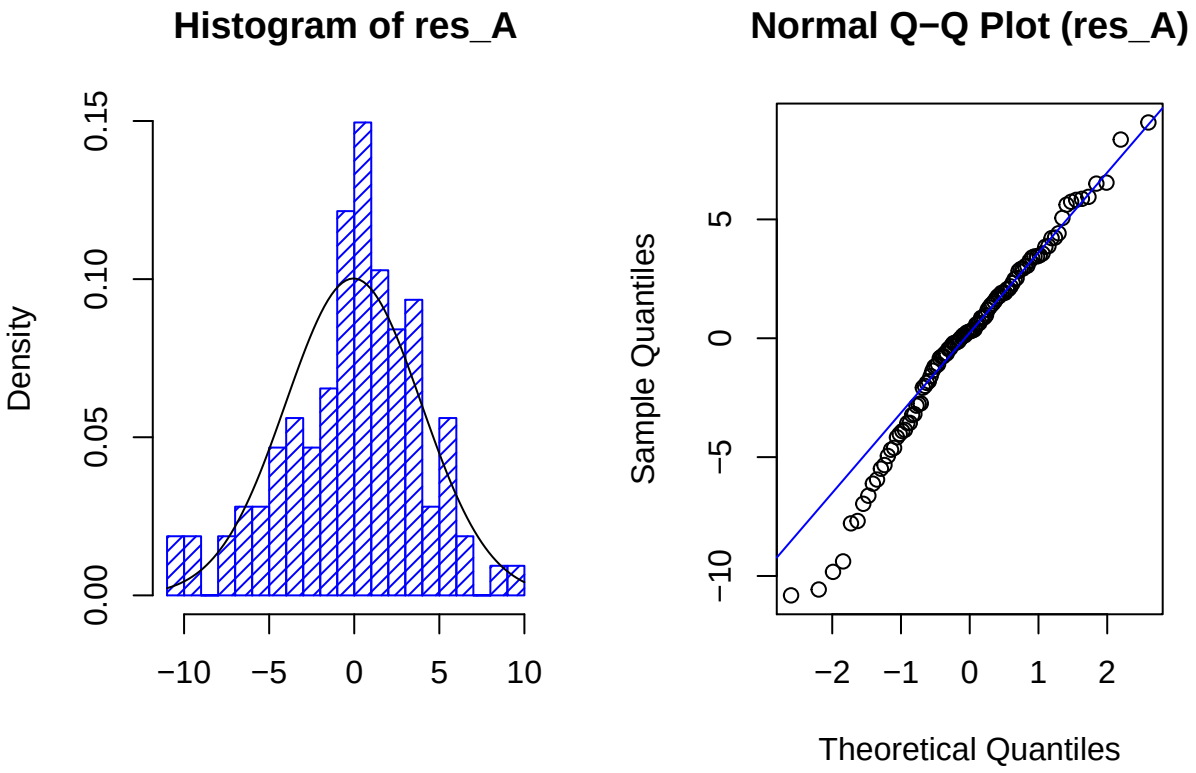

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  res_A
## W = 0.97675, p-value = 0.05729
```

```
##
##  Box-Pierce test
##
## data:  res_A
## X-squared = 4.2067, df = 8, p-value = 0.838
```

```
##
##  Box-Ljung test
##
## data:  res_A
## X-squared = 4.5641, df = 8, p-value = 0.803
```

```
##
##  Box-Ljung test
##
## data:  res_A^2
## X-squared = 12.023, df = 11, p-value = 0.3619
```



I fitted the residuals of model A using Yule-Walker estimation to see if the order selected is zero because I want to check if the residuals resemble white noise:

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

```
##
##
## Order selected 0  sigma^2 estimated as  15.85
```

Looking at the time series plot for the fitted residuals for Model A above, there is no clear trend, no visible change of variance, and no seasonality. The mean of the residuals is -0.04586404, which is close to zero. Looking at the Q-Q plot for Model A, it seems like the blue line fits through most of the points and it is somewhat close to a straight line. The histogram for the residuals for Model A suggests a heavy-tailed distribution. Although the histogram suggests a heavy-tailed distribution, since the Shapiro-Wilk normality test gives a p-value of 0.05729 (which is slightly greater than the threshold of 0.05), we do not reject the assumption of normality.

By looking at the ACF and PACF of the residuals of Model A, we can see that all lags are within confidence intervals so they can be counted as zeros, which suggests white noise. The Box-Pierce Test and Ljung-Box Test check for independence, while the McLeod-Li Test tests the residuals for nonlinear dependence, and these are important tests to check in order to determine if the residuals resemble white noise. The Box-Pierce Tests gives a p-value of 0.838, the Ljung-Box Test gives a p-value of 0.803, and the McLeod-Li Test gives a p-value of 0.3619. For the Box-Pierce Test, Ljung-Box Test, and McLeod-Li Test, all p-values are greater than 0.05, so we fail to reject the WN hypothesis. Using the ar() function and Yule-Walker method of estimation, I determined that the residuals can be fitted to AR(0), i.e. White Noise. Model A passes diagnostic checking and can be used for forecasting.

I will also do diagnostic checking for Model B. I have found the equation for $\text{SAR}(2)_{12}$ (Model B):

Since p = 0: $\phi(B) = 1$

Since d = 1: $(1 - B)$

Since q = 0: $\theta(B) = 1$

Since P = 2: $\Phi(B) = (1 - \Phi_1 B^{12} - \Phi_2 B^{24})$

Since D = 1: $(1 - B^s)^D = (1 - B^{12})^1$

Since Q = 0: $\Theta(B) = 1$

Also, s = 12.

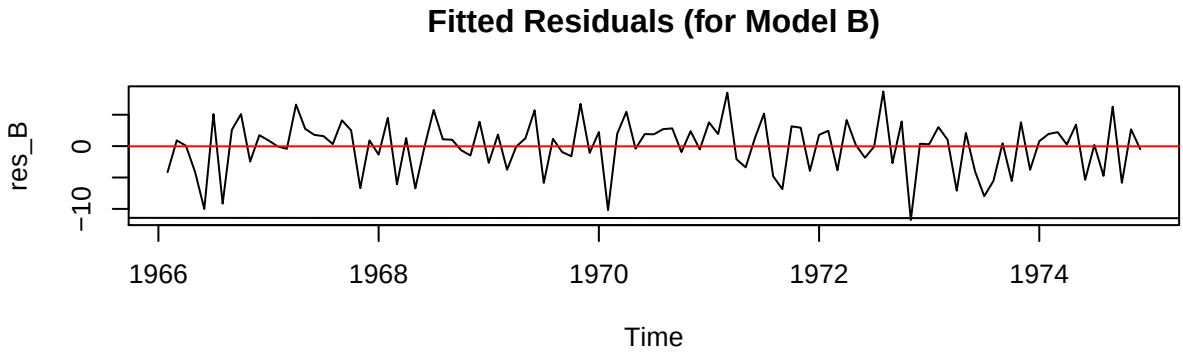
Therefore, the fitted model in algebraic form for Model B is: $(1 - \Phi_1 B^{12} - \Phi_2 B^{24})(1 - B^{12})(1 - B)X_t = Z_t$. After plugging in the coefficients that I estimated, the fitted model equation for Model B is $(1 + 0.5905_{(0.0987)} B^{12} + 0.3654_{(0.1022)} B^{24})(1 - B^{12})(1 - B)X_t = Z_t$; the subscripts correspond to the standard error of the coefficients. Also, $\hat{\sigma}_z^2 = 16.81$.

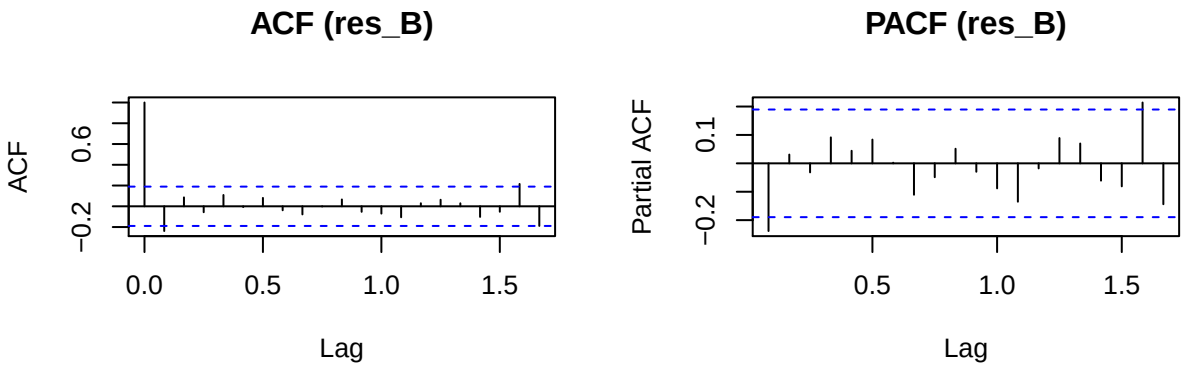
Diagnostic Checking for $\text{SAR}(2)_{12}$ (Model B):

```
## [1] -0.0362314
```

```
## [1] 16.96733
```

```
## [1] 4.119142
```



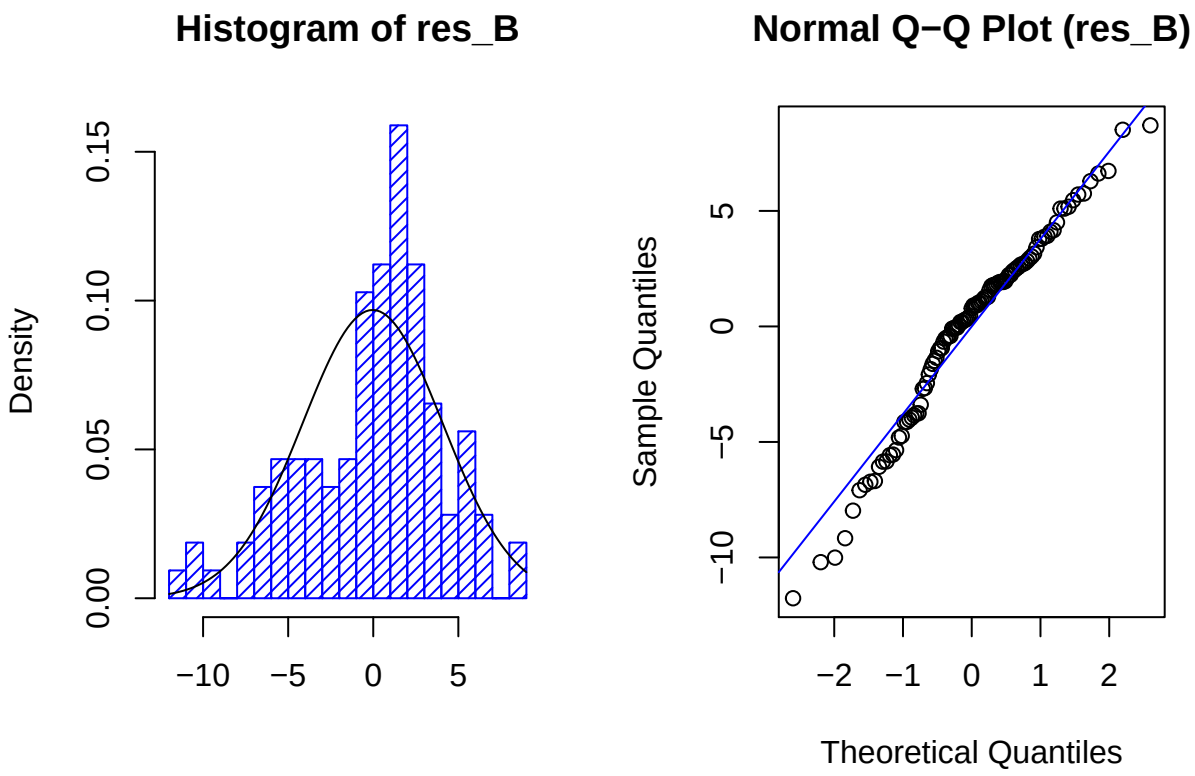



```
##
##  Shapiro-Wilk normality test
##
## data:  res_B
## W = 0.97039, p-value = 0.01717

##
##  Box-Pierce test
##
## data:  res_B
## X-squared = 10.768, df = 9, p-value = 0.2919

##
##  Box-Ljung test
##
## data:  res_B
## X-squared = 11.296, df = 9, p-value = 0.256

##
##  Box-Ljung test
##
## data:  res_B^2
## X-squared = 10.833, df = 11, p-value = 0.4573
```




```
##
## Call:
## ar(x = res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
##       1
## -0.2382
##
## Order selected 1  sigma^2 estimated as  16.16
```

Looking at the time series plot for the fitted residuals for Model B, there is no clear trend, no visible change of variance, and no seasonality. The mean of the residuals for Model B is -0.0362314, which is close to zero. The histogram for Model B does not look very symmetric. Looking at the Q-Q plot, it is somewhat close to a straight line. Since the Shapiro-Wilk normality test gives a p-value of 0.01717 (which is less than the threshold of 0.05), we reject the assumption of normality.

By looking at the ACF and PACF of the residuals for Model B, we can see that some lags are outside confidence intervals.

The Box-Pierce Test gives a p-value of 0.2919, the Ljung-Box Test gives a p-value of 0.4573, and the McLeod-Li Test gives a p-value of 0.4573. For the Box-Pierce Test, Ljung-Box Test, and McLeod-Li Test, all p-values are greater than 0.05, so we fail to reject the WN hypothesis.

The residuals for Model B can be fitted to AR(1), which is problematic because we wanted residuals to be fitted to AR(0) (White Noise). Since the residuals are not normal and cannot be fitted to white noise, Model B does not pass diagnostic checking and therefore, will not be used for forecasting.

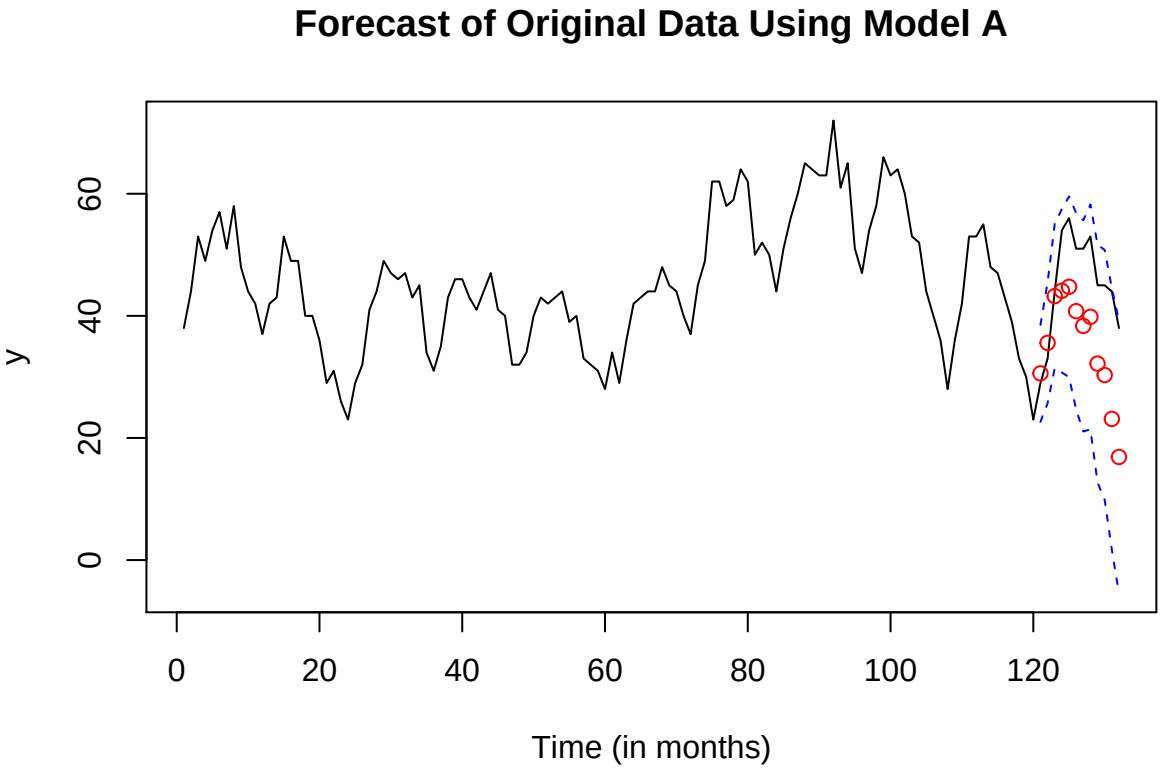
Hence, Model A is the final model since it passed all diagnostic checking. So, SARIMA$(1,0,0)(2,0,0)_{12}$ (Model A) is the best model for the differenced data. The fitted model equation for Model A is: $(1 + 0.2558_{(0.0963)}B)(1+0.6500_{(0.0987)}B^{12}+0.3554_{(0.1007)}B^{24})(1-B^{12})(1-B)X_t = Z_t$. This model was obtained using AICc and it is not one of the models that I preliminarily identified using ACF/PACF. I had to do some model modification in order to find this model. I used Model A for forecasting.

Section 5: Forecasting

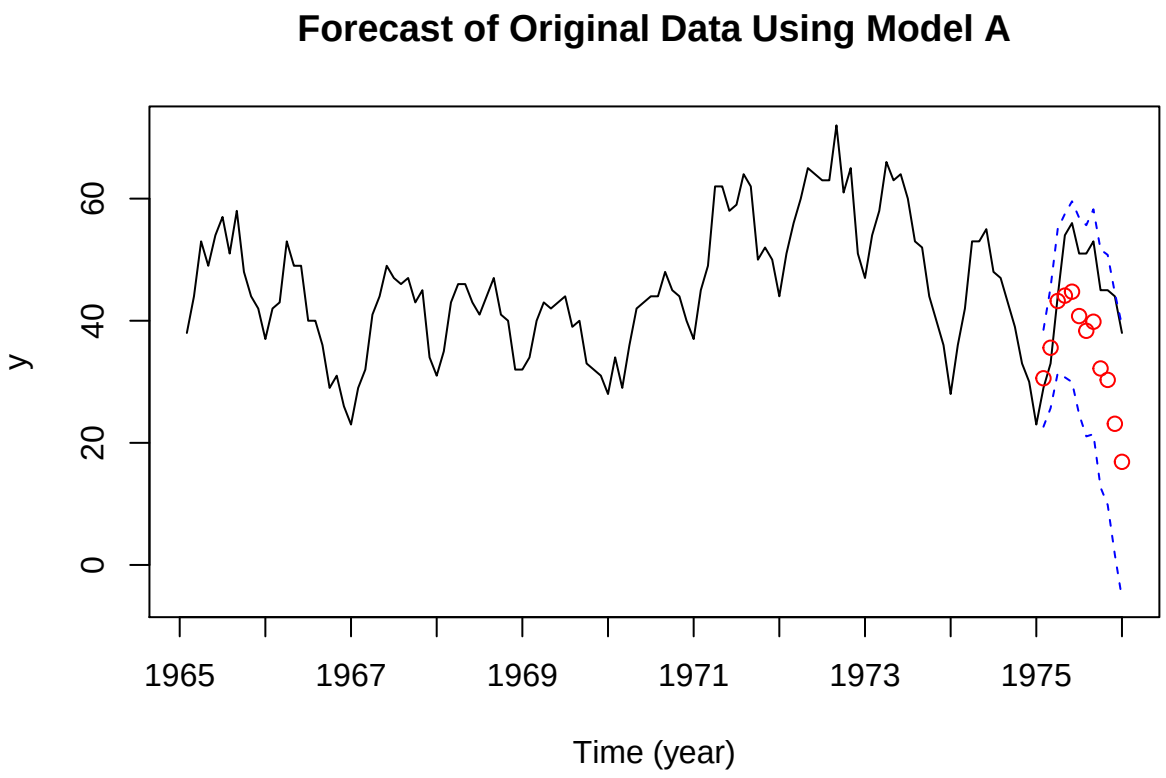
My original dataset (which is monthly data) has a total of 132 observations. My train data is 120 observations. My test data is 12 observations, which is for the year 1975. I have used Model A, my best model, and forecasted monthly sales of U.S. houses (in thousands) for the year 1975. In previous steps, I was working with the differenced data; I made sure to switch back to the original data before I started the forecasting process.

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method                 from
##   autoplot.Arima         ggfortify
##   autoplot.acf           ggfortify
##   autoplot.ar            ggfortify
##   autoplot.bats          ggfortify
##   autoplot.decomposed.ts ggfortify
##   autoplot.ets           ggfortify
##   autoplot.forecast      ggfortify
##   autoplot.stl           ggfortify
##   autoplot.ts            ggfortify
##   fitted.ar              ggfortify
##   fortify.ts             ggfortify
##   residuals.ar           ggfortify
```

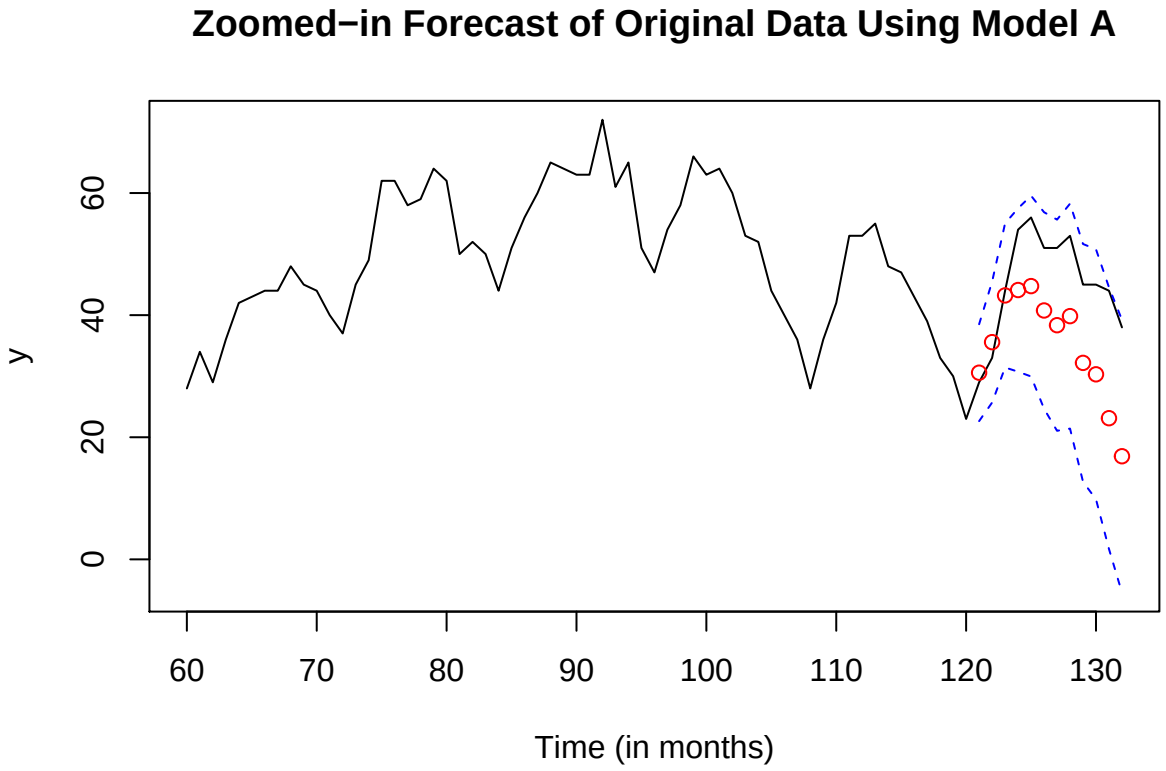


Above: Forecast of Original Data Using Model A (with months on the x-axis).

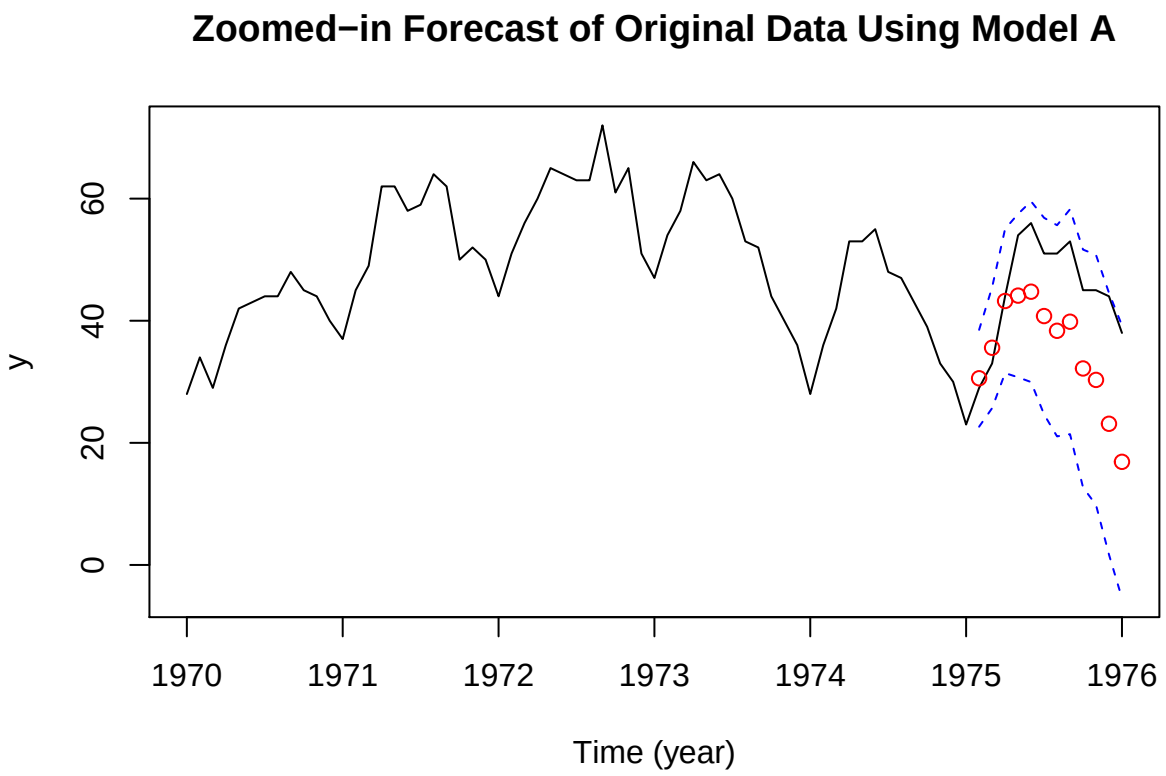


Above: Forecast of Original Data Using Model A (with years on the x-axis).

I have also created a zoomed-in plot labeled "Zoomed-in Forecast of Original Data Using Model A":



Above: Zoomed-In Forecast of Original Data Using Model A (with months on the x-axis).



Above: Zoomed-In Forecast of Original Data Using Model A (with years on the x-axis).

In the plots labeled "Forecast of Original Data Using Model A" and "Zoomed-in Forecast of Original Data Using Model A," the black line is the original data, the red circles are the forecasts, and the dashed blue lines are the prediction intervals. The test set is the original data (black line) from month 121 to month 132. By the plots, we can see that the first few months are predicted well but after the third month of 1975, the forecasts are lower than the the original data. This indicates an underprediction. Perhaps, since the histogram of the residuals for Model A suggests a heavy tail distribution, it explains why the prediction points don't reach the peaks after the third month of 1975. However, the true data values (black line) and the predicted values are still within confidence intervals. We can see that the upper bound of the prediction interval visually mimics the shape of the true data points in 1975. So, the prediction intervals capture the data well, and it models well based on the training data. Therefore, I believe that this forecast is good.

It is important to note that during the Arab-Israeli War which occurred in 1973, Arab countries imposed an oil embargo against the U.S. in order to "gain leverage in the post-war peace negotiations."[2] This event contributed to the downfall of the U.S.economy between 1973 to 1975, and more specifically, caused stagflation.[3] Stagflation is defined as a decline in real GDP along with an increase in inflation. This war contributed to the downfall of the U.S. economy and contributed to the recession.

I think that the true data values are higher than the predicted values due to the U.S.'s response to the U.S. recession from 1973 to 1975. In the plot of the original data, we can see that the sales of U.S. houses were the highest right before 1973 and then began to decline. Due to the pattern of rise in sales and decrease in sales from mid-1972 to the start of 1975, the predicted values for the year 1975 are justified. However, on October 28, 1974 the Equal Credit Opportunity Act was passed by Congress which made it unlawful for creditors to discriminate against credit applicants "on the basis of race, color, religion, national origin, sex, marital status."[4] I believe that the Equal Credit Opportunity Act helped many people access opportunities for home financing, and therefore helped the U.S. housing market start recovering from the recession; this may explain why the monthly sales of U.S. houses increased from the beginning of 1975 to the fourth month of 1975.

Conclusion

My goals for this project were find the best model based on Box-Jenkins Methodology for the data that I chose and to forecast monthly sales of U.S. houses in the year 1975. I differenced my training data at lags 1 and 12, analyzed the ACF/PACF in order to preliminarily identify p's and q's. After this, I determined the most complex model, constructed 95% confidence intervals, noticed that many of the confidence intervals contain zero, and set those coefficients to zero. So, through this process of model modification, I found Model A (SARIMA(1,0,0)(2,0,0)$_{12}$) for the differenced data. I compared models and did diagonostic checking for Model A and Model B. Only Model A passed diagnostic checking, so Model A was determined to be the best model. The model equation corresponding to Model A is $(1 + 0.2558_{(0.0963)}B)(1 + 0.6500_{(0.0987)}B^{12} + 0.3554_{(0.1007)}B^{24})(1 - B^{12})(1 - B)X_t = Z_t$. I used Model A to forecast monthly sales of U.S. houses in 1975. I found that the forecasted points and true data points were within confidence intervals. I observed that the first three months of 1975 were predicted very accurately. The true data points and forecasts for the rest of the year 1975 were still within confidence intervals. So, overall, it was a good prediction. I believe that my goals were achieved because I found the best model to be Model A and I was able to use this model to forecast monthly sales of U.S. houses in 1975.

Firstly, I would like to express my sincere gratitude to Professor Raya Feldman for all of her assistance, feedback, and encouragement throughout the development of this report. I also want to thank the Teaching Assistants, Chao Zhang and Youhong Lee, for their guidance during the process of completing this project.

References

1. The data set was obtained from the TSDL package in R. The subject of the data is "Sales," the data set description is "Monthly sales of U.S. houses (thousands) 1965 – 1975," and the source of the data set is "Abraham & Ledolter (1983)."

2. U.S. Department of State. (n.d.). Oil Embargo, 1973–1974. U.S. Department of State. https://history.state.gov/milestones/1969-1976/oil-embargo.

3. National Museum of American History. (2021, April 1). Energy Crisis. National Museum of American History. https://americanhistory.si.edu/american-enterprise-exhibition/consumer-era/energy-crisis.

4. The United States Department of Justice. (2020, July 22). The Equal Credit Opportunity Act. The United States Department of Justice. https://www.justice.gov/crt/equal-credit-opportunity-act-3.

Appendix (code with comments)

Code for Section 1 (Plot and Analyze Time Series):

```
# loading required packages
library(tsdl) # this is the library I used to obtain the data set
```

```
library(ggplot2)
library(ggfortify)

y <- tsdl[[6]] # original dataset from TSDL package in R
train1 <- y[1:120] # training data to build a model

# training data converted to time series:
train <- ts(train1, start=c(1965,1), end=c(1974,12), frequency=12)

test1 <- y[121:132] # test data

# test data converted to time series:
test <- ts(test1, start=c(1975,1), end=c(1975,12), frequency=12)

ts.plot(train, ylab = "Monthly Sales of U.S. Houses (in Thousands)",
        main = "Monthly Sales of U.S. Houses between 1965 and 1975") # time series plot of train
abline(v = ts(c(1967,1968,1969, 1970)), col = "blue", lty = 2) # to show the seasonal cycles
abline(h = mean(train), col="red") # mean of data

z <- ts(as.ts(train), frequency = 12)
decomp <- decompose(z)

# decomposition of additive time series:
plot(decomp) # separated into observed, trend, seasonal, and random parts
```

Code for Section 2 (Tranformations or Differencing?):

```
# histogram of train (original data):
hist(train, main = 'Histogram of Y_t (Original Data, Truncated)')
```

```
library(MASS)
par(mfrow = c(1,2))

# I tried Box-Cox transformation:

t <- 1:length(train)
fit <- lm(train ~ t)
bcTransform <- boxcox(train ~ t, plotit = TRUE) # Since the confidence interval includes 1,
# a transformation is not necessary.
# The confidence interval includes lambda = 1 (and the optimal value for lamda is 1) which means
# that the data is already normally distributed and a
```

```r
# Box-Cox transformation is not necessary.

lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))] # finding lamda
lambda # lambda is 0.70707077
```

```r
y_1 = diff(train, 1) # apply differencing at lag 1 to remove time trend

par(mfrow = c(1, 3))
acf(y_1, lag.max = 40, main = "") # plotting ACF

pacf(y_1, lag.max = 40, main = "") # plotting PACF

# time series plot of the train data differenced at lag 1 only:
ts.plot(y_1, ylab = "Monthly Sales of U.S. Houses (in Thousands)",
        main = "Y_t Differenced at Lag 1")
abline(h = mean(y_1), col="blue")
```

```r
y_12 = diff(y_1, 12) # apply differencing at lag 12 to remove seasonality
par(mfrow = c(1, 3))

acf(y_12, lag.max = 40, main = "") # plotting ACF
pacf(y_12, lag.max = 40, main = "") # plotting PACF

# time series plot of the train data differenced at lags 1 & 12:
ts.plot(y_12, ylab = "Monthly Sales of U.S. Houses (in Thousands)",
        main = "Y_t after differencing at Lag 1 & Lag 12")
abline(h = mean(y_12), col="blue")
```

```r
var(train) # The variance of train is 111.898.
var(y_1) # The variance after differencing the original time series at lag 1 is 29.23077
var(y_12) # The variance after differencing the time series at lags 1 and 12 is 23.0846

y_12_1 = diff(y_12, 1) # differencing y_12 again at lag 1 to check if variance increases
var(y_12_1) # variance after differencing at lag 1, lag 12, and again lag 1 is 52.80081
```

```r
# Plotting 2 histograms side-by-side (histogram of y and histogram of y_12):
par(mfrow = c(1,2))

g <- hist(train, main = 'Y_t (Original Data, Truncated)') # histogram of Y_t
# layering normal curve over histogram:
fit_x <- seq(min(train), max(train), length = 40)
fit_y <- dnorm(fit_x, mean = mean(train), sd = sd(train))
fit_y <- fit_y * diff(g$mids[1:2]) * length(train)
lines(fit_x, fit_y, col = "black", lwd = 2)

h <- hist(y_12, main = 'Y_t (differenced at lags 1 & 12)', ylim = c(0,50))
# layering normal curve over histogram:
xfit <- seq(min(y_12), max(y_12), length = 40)
yfit <- dnorm(xfit, mean = mean(y_12), sd = sd(y_12))
yfit <- yfit * diff(h$mids[1:2]) * length(y_12)
lines(xfit, yfit, col = "black", lwd = 2)
```

Code for Section 3 (Preliminary Identification of Models):

```
par(mfrow = c(1, 2))

acf(y_12, lag.max = 40, main = "") # plotting ACF of Y_t differenced at lags 1 & 12
pacf(y_12, lag.max = 40, main = "") # plotting PACF of Y_t differenced at lags 1 & 12
```

Code for Section 4 (Fitting Models):

```
# Trying different models, estimating coefficients, finding AICc values:

# SAR(2):
arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# AICc is 619.7619


# This model is not good; has unit root in MA part:
arima(y_12, order=c(0,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML") # SMA(1)
AICc(arima(y_12, order=c(0,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML"))


# This model is SARIMA(0,1,0)(2,1,1)_{12}:
# the model is not good, it has unit root in MA part:
arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,1), period = 12), method="ML")
AICc(arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,1), period = 12), method="ML"))


# We can start with the most complex model:
arima(y_12, order=c(8,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(8,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# AICc is 628.5168
# constructing 95% CI for coefficients of SARIMA(8,0,0)(2,0,0)_{12}:
# 95% CI for ar1 is (-0.443768, -0.058432); does not contain zero
-0.2511 - 1.96*(0.0983) # lower bound
-0.2511 + 1.96*(0.0983) # upper bound

# 95% CI for ar2 is (-0.172864, 0.212864); contains zero
0.0200 - 1.96*(0.0984) # lower bound
0.0200 + 1.96*(0.0984) # upper bound

# 95% CI for ar3 is (-0.208808, 0.182408); contains zero
-0.0132 - 1.96*(0.0998) # lower bound
-0.0132 + 1.96*(0.0998) # upper bound

# 95% CI for ar4 is (-0.07746, 0.31846); contains zero
0.1205 - 1.96*(0.1010) # lower bound
0.1205 + 1.96*(0.1010) # upper bound

# 95% CI for ar5 is (-0.140136, 0.258136); contains zero
```

```r
0.0590 - 1.96*(0.1016) # lower bound
0.0590 + 1.96*(0.1016) # upper bound

# 95% CI for ar6 is (-0.115628, 0.303028); contains zero
0.0937 - 1.96*(0.1068) # lower bound
0.0937 + 1.96*(0.1068) # upper bound

# 95% CI for ar7 is (-0.221992, 0.200192); contains zero
-0.0109 - 1.96*(0.1077) # lower bound
-0.0109 + 1.96*(0.1077) # upper bound

# 95% CI for ar8 is (-0.27304, 0.12484); does not contain zero
-0.0741 - 1.96*(0.1015) # lower bound
-0.0741 + 1.96*(0.1015) # upper bound

# 95% CI for sar1 is (-0.8447, -0.4331); does not contain zero
-0.6389 - 1.96*(0.1050) # lower bound
-0.6389 + 1.96*(0.1050) # upper bound

# 95% CI for sar2 is (-0.526488, -0.103912); does not contain zero
-0.3152 - 1.96*(0.1078) # lower bound
-0.3152 + 1.96*(0.1078) # upper bound
# For ar2, ar3, ar4, ar5, ar6, ar7, ar8, it is clear that zero is
# contained in the confidence intervals,
# so we can set all of those coefficients to zero. Therefore, we can modify this model
# to make it a more simple model like SARIMA(1,0,0)(2,0,0)_{12}


# This model is SARIMA(1,0,0)(2,0,0)_{12}:
arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# AICc is 615.1695 (lower than the complex model)
```

```r
# Trying even more different models, estimating coefficients, finding AICc values:

# This model is not good; it has unit root in MA part:
arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,1), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,1), period = 12), method="ML"))

arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# AICc is 615.1695


arima(y_12, order=c(1,0,0), seasonal = list(order = c(0,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(0,0,0), period = 12), method="ML"))
# AICc is 645.5959


arima(y_12, order=c(1,0,0), seasonal = list(order = c(1,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(1,0,0), period = 12), method="ML"))
# AICc is 623.9365
```

```r
arima(y_12, order=c(8,0,0), seasonal = list(order = c(1,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(8,0,0), seasonal = list(order = c(1,0,0), period = 12), method="ML"))
# AICc is 633.69


arima(y_12, order=c(8,0,0), seasonal = list(order = c(0,0,0), period = 12), method="ML")
AICc(arima(y_12, order=c(8,0,0), seasonal = list(order = c(0,0,0), period = 12), method="ML"))
 # AICc is 654.8264


# this model is not good; it has unit root in MA part:
arima(y_12, order=c(8,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML")
AICc(arima(y_12, order=c(8,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML"))


# this model is not good; has unit root in MA part:
arima(y_12, order=c(1,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(0,0,1), period = 12), method="ML"))
```

```r
library(qpcR)
# SARIMA(1,0,0)(2,0,0)_{12} (MODEL A):
arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
print("AICc for SARIMA(1,0,0)(2,0,0)_{12} (MODEL A) (for the differenced data):")
AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# Model A's AICc is 615.1695

# SAR(2)_{12} (MODEL B):
arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
print("AICc for SAR(2)_{12} (MODEL B) (for the differenced data):")
AICc(arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))
# Model B's AICc is 619.7619
```

```r
# constructing 95% confidence intervals for SARIMA(1,0,0)(2,0,0)$_{12}$ (Model A):

# for coefficient ar1, 95% CI is (-0.444548, -0.067052)
-0.2558 - 1.96*(0.0963) # this simplifies to -0.444548
-0.2558 + 1.96*(0.0963) # this simplifies to -0.067052

# for coefficient sar1, 95% CI is (-0.843452, -0.456548)
-0.6500 - 1.96*(0.0987) # this simplifies to -0.843452
-0.6500 + 1.96*(0.0987) # this simplifies to -0.456548

# for coefficient sar2, 95% CI is (-0.552772, -0.158028)
-0.3554 - 1.96*(0.1007) # this simplifies to -0.552772
-0.3554 + 1.96*(0.1007) # this simplifies to -0.158028
```

```r
# constructing 95% confidence intervals for SAR(2)$_{12}$ (Model B):

# for coefficient sar1, 95% CI is (-0.783952, -0.397048)
-0.5905 - 1.96*(0.0987) # this simplifies to -0.783952
-0.5905 + 1.96*(0.0987) # this simplifies to -0.397048
```

```r
# for coefficient sar2, 95% CI is (-0.565712, -0.165088)
-0.3654 - 1.96*(0.1022) # this simplifies to -0.565712
-0.3654 + 1.96*(0.1022) # this simplifies to -0.165088
```

```r
# MODEL A
# arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")

source("plot.roots.R")
library(UnitCircle)
library(qpcR)

# For the AR part (non-seasonal), we can clearly tell
# that the coefficient (which is 0.2558) is strictly smaller than 1
# but I can check the roots of the AR part (non-seasonal) in R:
uc.check(pol = c(1, 0.2558), plot_output = TRUE)

# checking the roots of the AR part (seasonal):
uc.check(pol = c(1, 0.6500, 0.3554), plot_output = TRUE)
```

```r
# MODEL B
# arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")

uc.check(pol = c(1, 0.5905, 0.3654), plot_output = TRUE)
```

```r
# MODEL A DIAGNOSTIC CHECKING:

#arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
#AICc(arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML"))

# y_12 is the data differenced at lags 1 and 12

fit.i <- arima(y_12, order=c(1,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
res_A = residuals(fit.i) # fitted residuals of Model A

mean(res_A) # mean of fitted residuals of Model A
var(res_A) # variance of fitted residuals of Model A
sqrt(var(res_A)) # standard deviation of fitted residuals of Model A

layout(matrix(c(1,1,2,3),2,2,byrow=T))
ts.plot(res_A, main = "Fitted Residuals (for Model A)") # plot of fitted residuals of Model A
t = 1:length(res_A)
fit.res.A = lm(res_A~t)
abline(fit.res.A)
abline(h = mean(res_A), col = "red") # adding best fit line

acf(res_A,main = "ACF (res_A)") # ACF of residuals

pacf(res_A,main = "PACF (res_A)") # PACF of residuals

shapiro.test(res_A) # Shapiro Test

Box.test(res_A, lag = 11, type = c("Box-Pierce"), fitdf = 3) # Box-Pierce Test
# fitdf is 3 b/c there are 3 estimated parameters for Model A
```

```r
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11

Box.test(res_A, lag = 11, type = c("Ljung-Box"), fitdf = 3) # Ljung-Box Test
# fitdf is 3 b/c there are 3 parameters estimated for Model A
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11

Box.test(res_A^2, lag = 11, type = c("Ljung-Box"), fitdf = 0) # McLeod-Li Test:Ljung-Box for squares
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11

par(mfrow=c(1,2))
# Histogram
hist(res_A, density=20, breaks=20, col="blue", xlab="", prob=TRUE) # histogram of residuals
m_A <- mean(res_A)
std_A <- sqrt(var(res_A))
curve(dnorm(x,m_A,std_A), add=TRUE) # layering normal distribution curve over histogram

qqnorm(res_A, main = 'Normal Q-Q Plot (res_A)') # normal Q-Q plot
qqline(res_A,col ="blue") # adding best fit line to Q-Q plot

ar(res_A, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# ORDER SELECTED 0, so residuals fitted to AR(0), i.e. White Noise
```

```r
# MODEL B DIAGNOSTIC CHECKING

fit.i1 <- arima(y_12, order=c(0,0,0), seasonal = list(order = c(2,0,0), period = 12), method="ML")
res_B = residuals(fit.i1) # fitted residuals

mean(res_B) # mean of fitted residuals of Model B
var(res_B) # variance of fitted residuals of Model B
sqrt(var(res_B)) # standard deviation of fitted residuals of Model B

layout(matrix(c(1,1,2,3),2,2,byrow=T))
ts.plot(res_B, main = "Fitted Residuals (for Model B)") # plot of Fitted Residuals
t = 1:length(res_B)
fit.res.B = lm(res_B~t)
abline(fit.res.B)
abline(h = mean(res_B), col = "red")


acf(res_B,main = "ACF (res_B)") # ACF of residuals

pacf(res_B,main = "PACF (res_B)") # PACF of residuals


shapiro.test(res_B)

Box.test(res_B, lag = 11, type = c("Box-Pierce"), fitdf = 2) # Box-Pierce Test
# fitdf is 2 b/c there are 2 estimated parameters for Model B
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11
```

```r
Box.test(res_B, lag = 11, type = c("Ljung-Box"), fitdf = 2) # Ljung-Box Test
# fitdf is 2 b/c there are 2 parameters estimated for Model B
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11

Box.test(res_B^2, lag = 11, type = c("Ljung-Box"), fitdf = 0) # McLeod-Li Test:Ljung-Box for squares
# there are 120 observations in train; sqrt(120) = 10.95 (rounded up to 11)
# so lag = 11


par(mfrow=c(1,2))
# Histogram
hist(res_B, density=20, breaks=20, col="blue", xlab="", prob=TRUE) # histogram of residuals
m_B <- mean(res_B)
std_B <- sqrt(var(res_B))
curve( dnorm(x,m_B,std_B), add=TRUE) # layering normal distribution curve over histogram

qqnorm(res_B, main = 'Normal Q-Q Plot (res_B)' ) # Normal q-q plot
qqline(res_B,col ="blue") # adding best fit line to Q-Q plot

ar(res_B, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# ORDER SELECTED 1, not white noise
```

Code for Section 5 (Forecasting):

```r
# Creating ORIGINAL plot of forecast (where MONTHS are on the x-axis):

library(forecast)
# switched back to original data (truncated)
fit.i <- arima(train, order=c(1,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
forecast(fit.i)

pred.tr <- predict(fit.i, n.ahead = 12)
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
plot(x = 1:132, y = c(as.vector(train), as.vector(test)),
     ylim = c(min(min(train), min(L.tr)), max(max(train),max(U.tr))), type = "l",
     main = "Forecast of Original Data Using Model A", xlab = 'Time', ylab = 'y')
lines(121:132, as.vector(U.tr), col="blue", lty="dashed") # adding upper bound to plot
lines(121:132, as.vector(L.tr), col="blue", lty="dashed") # adding lower bound to plot

points(121:132, pred.tr$pred, col="red") # predicted points

# true values are shown by the black line
# blue dashed lines are confidence intervals
# red points are the predicted values
```

```r
# Creating ORIGINAL plot of forecast (where YEARS are on the x-axis):

library(forecast)
```

```r
# switched back to original data
fit.i <- arima(train, order=c(1,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
#forecast(fit.i)

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

plot(x = 1:132, y = c(as.vector(train), as.vector(test)),
     ylim = c(min(min(train), min(L.tr)), max(max(train),max(U.tr))), type = "l",
     main = "Forecast of Original Data Using Model A", xlab = 'Time (year)',
     ylab = 'y', xaxt='n')
lines(121:132, as.vector(U.tr), col="blue", lty="dashed") # adding upper bound to plot
lines(121:132, as.vector(L.tr), col="blue", lty="dashed") # adding lower bound to plot

points(121:132, pred.tr$pred, col="red") # predicted points
axis(1, c(0, 12, 24, 36, 48, 60, 72, 84, 96, 108, 120, 132), tick = TRUE,
     labels = c(1965, 1966, 1967, 1968, 1969, 1970, 1971, 1972, 1973, 1974, 1975, 1976))
```

```r
# Creating a ZOOMED-IN plot of forecast (where MONTHS are on the x-axis):

library(forecast)
fit.i <- arima(train, order=c(1,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
#forecast(fit.i) # prints forecasts with prediction bounds in a table

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

plot(x = 60:132, y = c(as.vector(train)[60:120], as.vector(test)),
     ylim = c(min(min(train), min(L.tr)), max(max(train),max(U.tr))), type = "l",
     main = 'Zoomed-in Forecast of Original Data Using Model A', xlab = 'Time',
     ylab = 'y')
lines(121:132, as.vector(U.tr), col="blue", lty="dashed") # adding upper bound to plot
lines(121:132, as.vector(L.tr), col="blue", lty="dashed") # adding lower bound to plot

points(121:132, pred.tr$pred, col="red") # predicted points

# true values are shown by the black line
# blue dashed lines are confidence intervals
# red points are the predicted values
```

```r
# Creating a ZOOMED-IN plot of forecast (where YEARS are on the x-axis):

library(forecast)
fit.i <- arima(train, order=c(1,1,0), seasonal = list(order = c(2,1,0), period = 12), method="ML")
#forecast(fit.i) # prints forecasts with prediction bounds in a table

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

plot(x = 60:132, y = c(as.vector(train)[60:120], as.vector(test)),
```

```
     ylim = c(min(min(train), min(L.tr)), max(max(train),max(U.tr))), type = "l",
     main = 'Zoomed-in Forecast of Original Data Using Model A',
     xlab = 'Time (year)', ylab = 'y', xaxt='n')
lines(121:132, as.vector(U.tr), col="blue", lty="dashed") # adding upper bound to plot
lines(121:132, as.vector(L.tr), col="blue", lty="dashed") # adding lower bound to plot

points(121:132, pred.tr$pred, col="red") # predicted points
axis(1, c(60, 72, 84, 96, 108, 120, 132), tick = TRUE,
     labels = c(1970, 1971, 1972, 1973, 1974, 1975, 1976))

# true values are shown by the black line
# blue dashed lines are confidence intervals
# red points are the predicted values
```