PSTAT174 Final Project Total Vehicle Sales Forecasting

Yuchen Zheng

6/2/2021

Abstract

The goal of this project is to study monthly total vehicle sales in the U.S. from 2010 to 2019 and to build a model that best fits the data so that it could be used for forecasting. The data analyzed in this project is time series data and Box-Jenkins methodology is used. The data is partitioned, examined and differenced. The best model is selected using AICc score and checked with diagnostic checking on residuals. The final model selected is a SARIMA model with one autoregressive coefficient and one moving average coefficient. The confidence interval produced by the selected model captures all the true values. However, the selected model does not satisfy all the required properties for a model used for time series forecasting. Therefore, more advanced modeling techniques are needed to improve the model for it to make better predictions on similar dataset.

Introduction

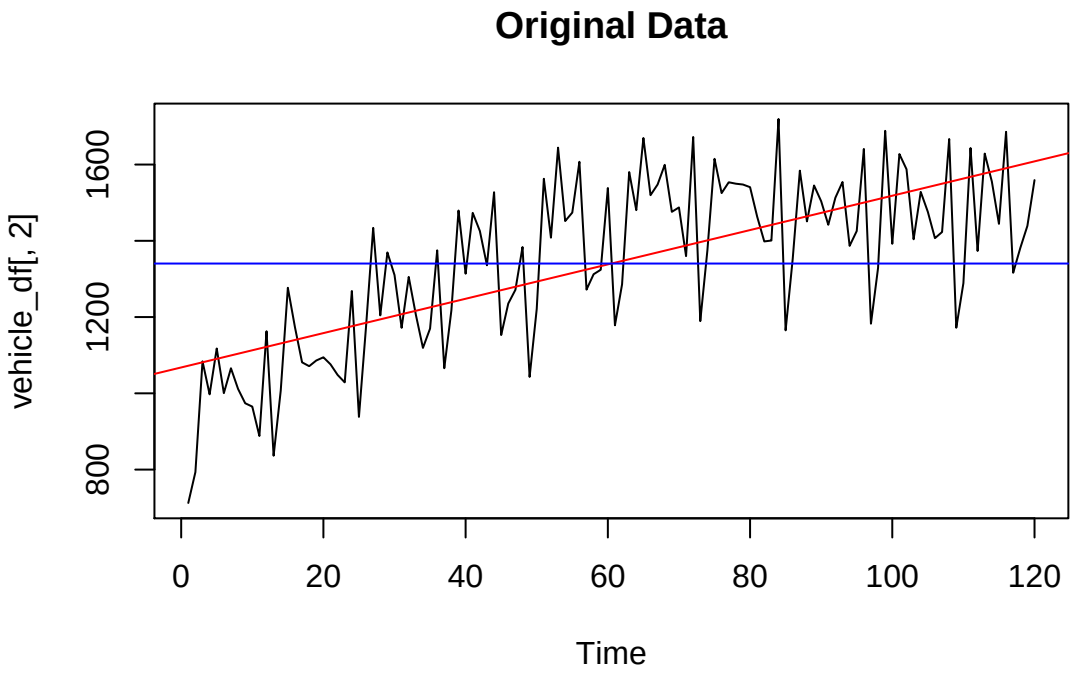
The dataset to be analyzed in this study is the monthly total vehicle sales (thousand of units) in the U.S. from 2010 to 2019. The dataset was obtained from the Federal Reserve Economic Data Database. This data set is worth to be analyzed because it could give us valuable information of the vehicle sales industry in the second decade of 21st century. The project results could allow various sectors of society such as manufacturers, government and car companies to understand the trend of market demand for vehicles, make predication for future vehicles sales and carry out effective sales strategy for the next decade. There are a total of 120 observations in this dataset and the dataset is partitioned into two parts, the training dataset and the testing dataset. The training dataset includes observations from the year of 2010 to 2018 and the testing dataset includes observations of 2019. The goal is to build a Box-Jenkins model that best fits the training dataset and then to be used for making predictions of the monthly vehicle sales in 2019. The performance of the model is evaluated by comparing the predictions and the true observations in 2019.

The training dataset is not stationary by looking at the histogram, Autocorrelation Function (ACF) plot and Partial Autocorrelation (PACF) plot. Therefore, the dataset is differenced at lag 1 and lag 12 to make it a stationary time series. Possible number of parameters for both moving average part and autoregressive part of the model are selected based on the ACF and PACF to obtain tentative models. Fourteen tentative models are selected to check for unit roots and only three of them do not have unit roots. Then diagnostic checking on residuals is done on the three models by using Shapiro-Wilk's normality test, Box-Pierce test, Ljung-Box test, McLeod-Li test, and Yule-Walker test to check if the residuals of the models satisfy the normality and independence assumptions.

The p-value for the Shapiro-Wilk's normality test for Model D was less than 0.05 which means normaility assumption for the model is violated. The ACFs and PACFs of the residuals for Model D are significant at some lag which means the residuals are slightly correlated. The p-values for all other tests for Model D were greater than 0.05 which means Model D satisfies the assumptions from those tests. Despite some violations indicated from the diagnostic checking, model D is the best model and is used for forecasting. The confidence interval produced by the selected model captures all the true values. Overall, the model selected has good predictive ability.

Data Examination

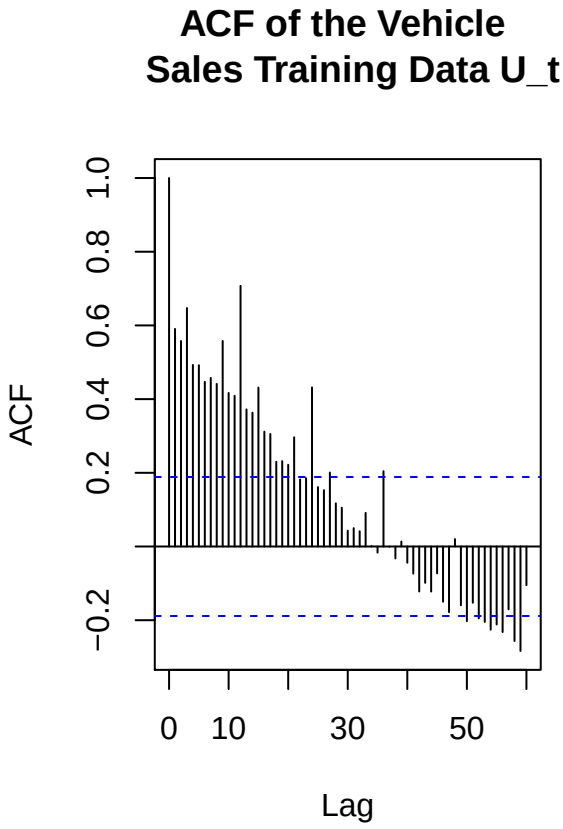
Firstly, the entire dataset is plotted to observe if there are any change of patterns. Since the Box-Jenkins methodology is predicted on the assumption that what happened in the past influences what will happen in the future, it is important to make sure the data does not have sudden sharp change of pattern within the study period.

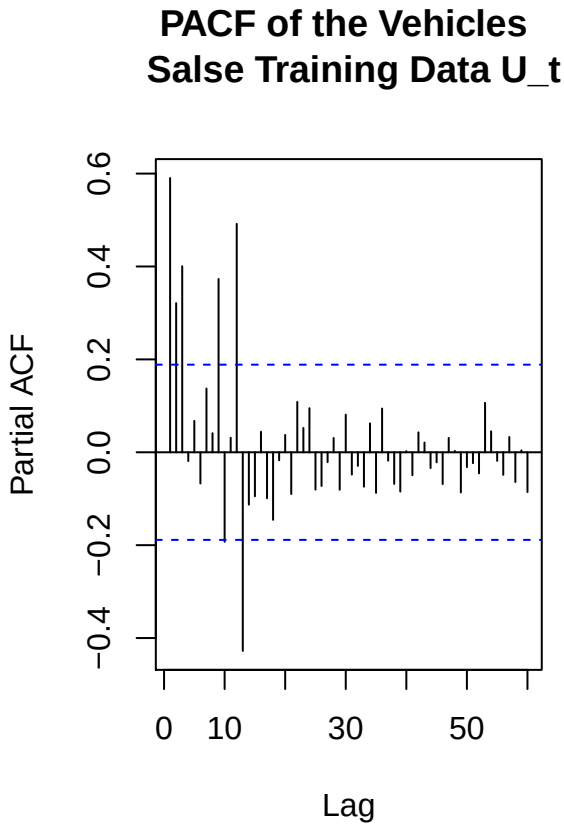


No sudden sharp change of behavior is observed in this plot. The data seems to have an steadily increasing trend and some seasonality. Therefore, Box-Jenkins methodology is suitable to be applied for making forecasts on this dataset.

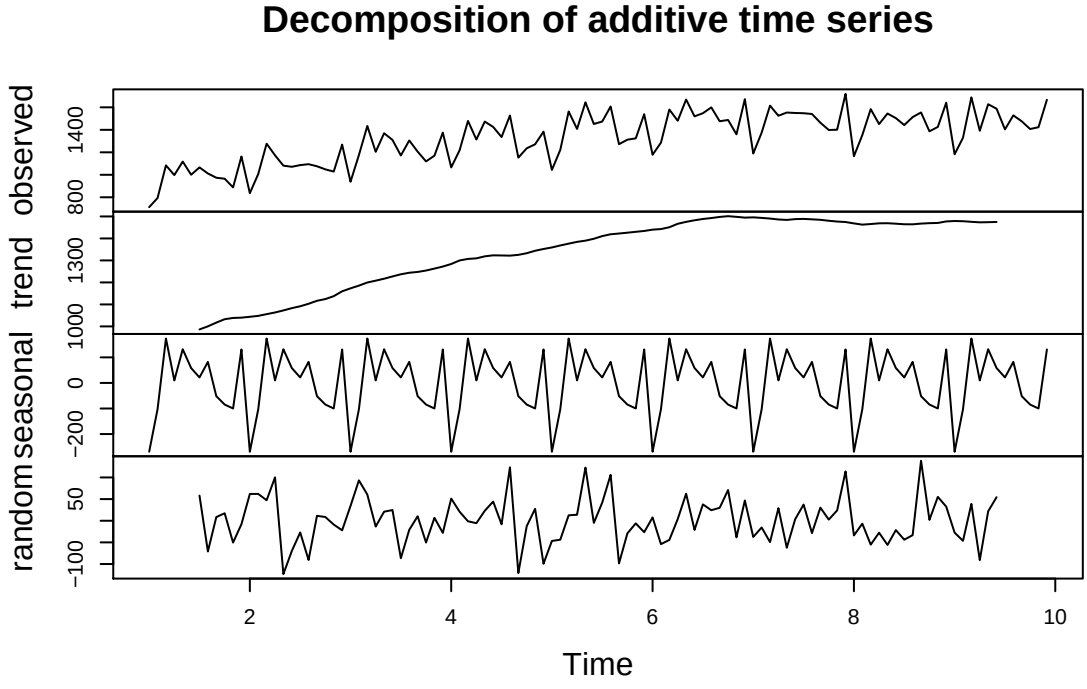
Next, the dataset is partitioned into the training dataset and the testing dataset. A plot and a histogram are generated on the training dataset to show any patterns.

Immediate observations from the left plot include an increasing linear trend and seasonality. An increasing linear trend means it has a nonconstant mean, but the variance seems to be constant. The histogram of the traning dataset does not seem to be symmetric. Values are lower on the left hand side. It can be concluded that the time series is not stationary.





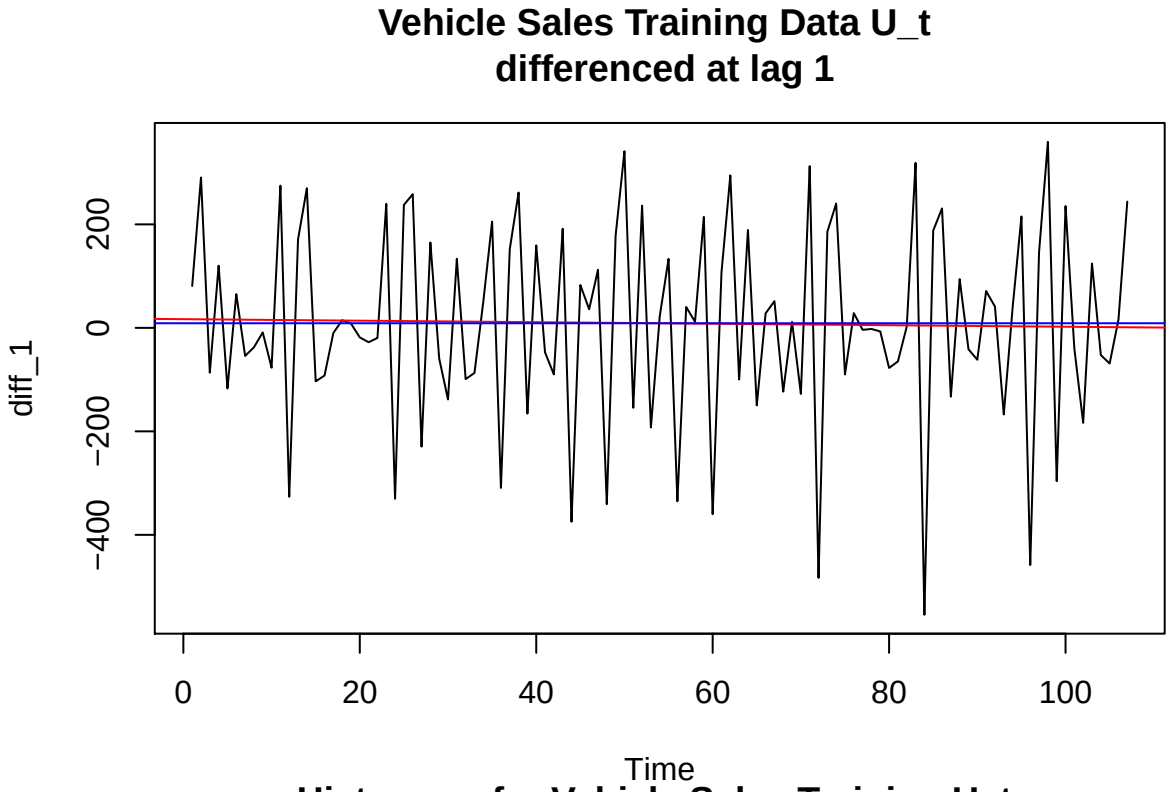
Both ACF and PACF plots indicate the time series is nonstationary. ACF values remain large and periodic.

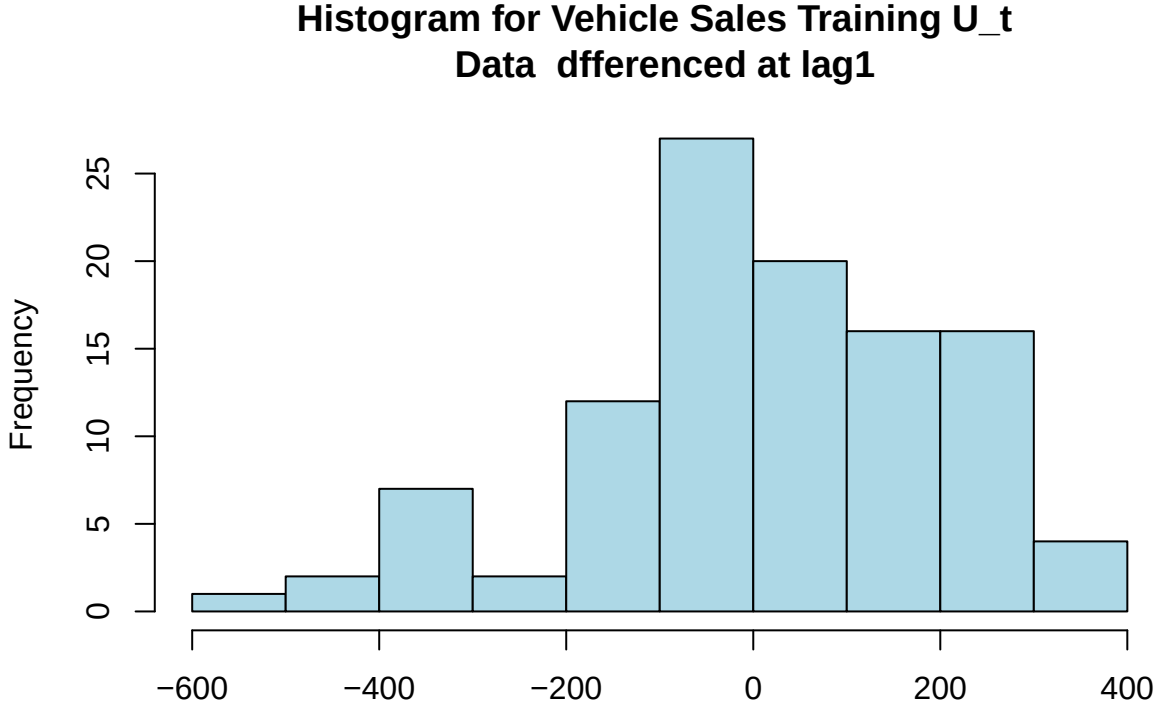


Decomposition of the training dataset also shows seasonality and an upward trend, but the variance seems to be constant. Thus, there is no need to transform the data, but it is necessary to difference the data to make it a stationary time series.

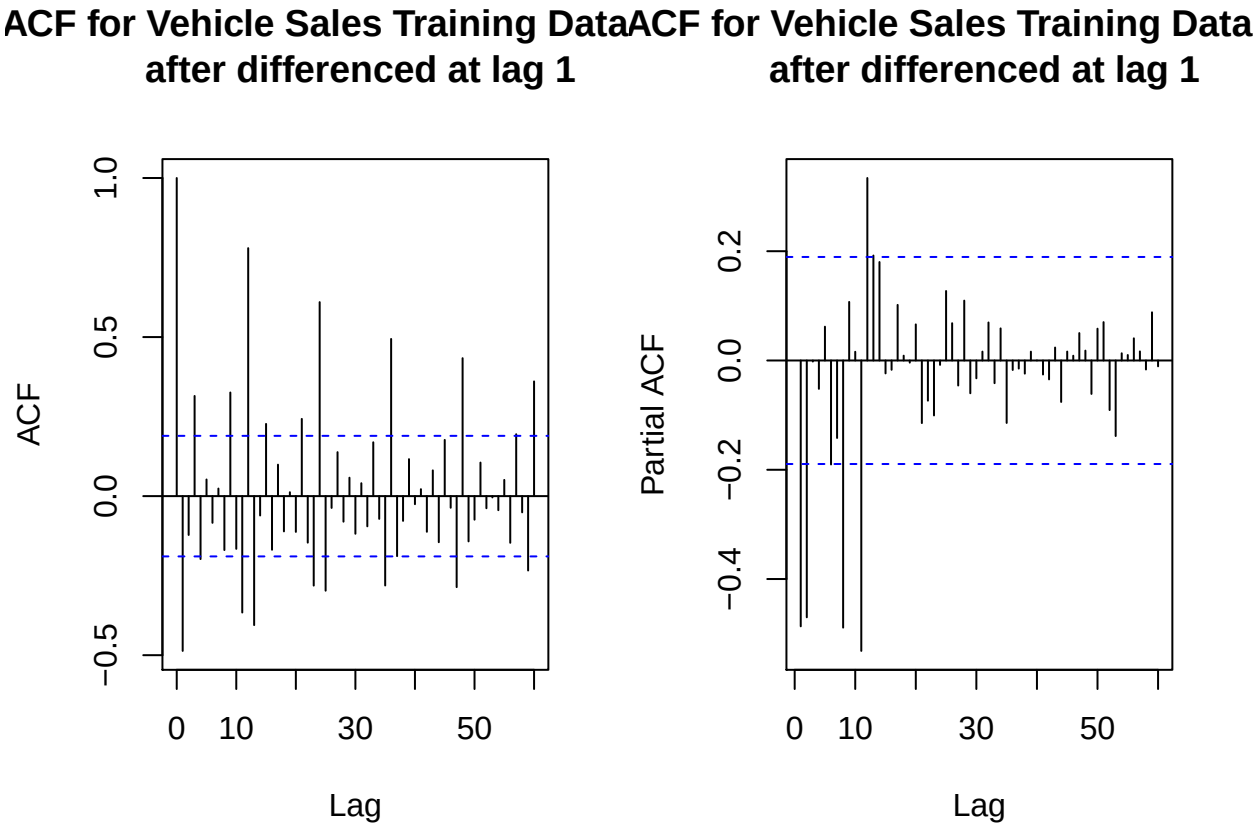
Data Differencing

Given the data has an upward linear trend, the data is differenced at lag 1 to remove the linear trend.

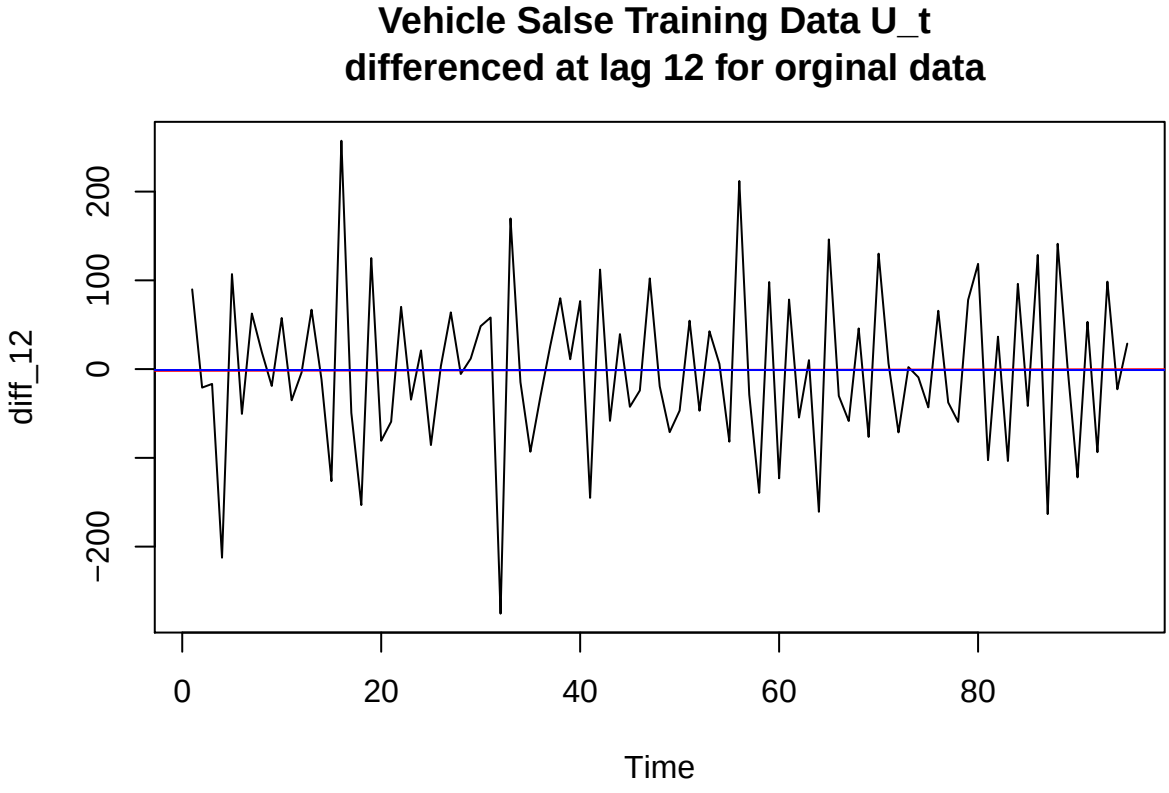


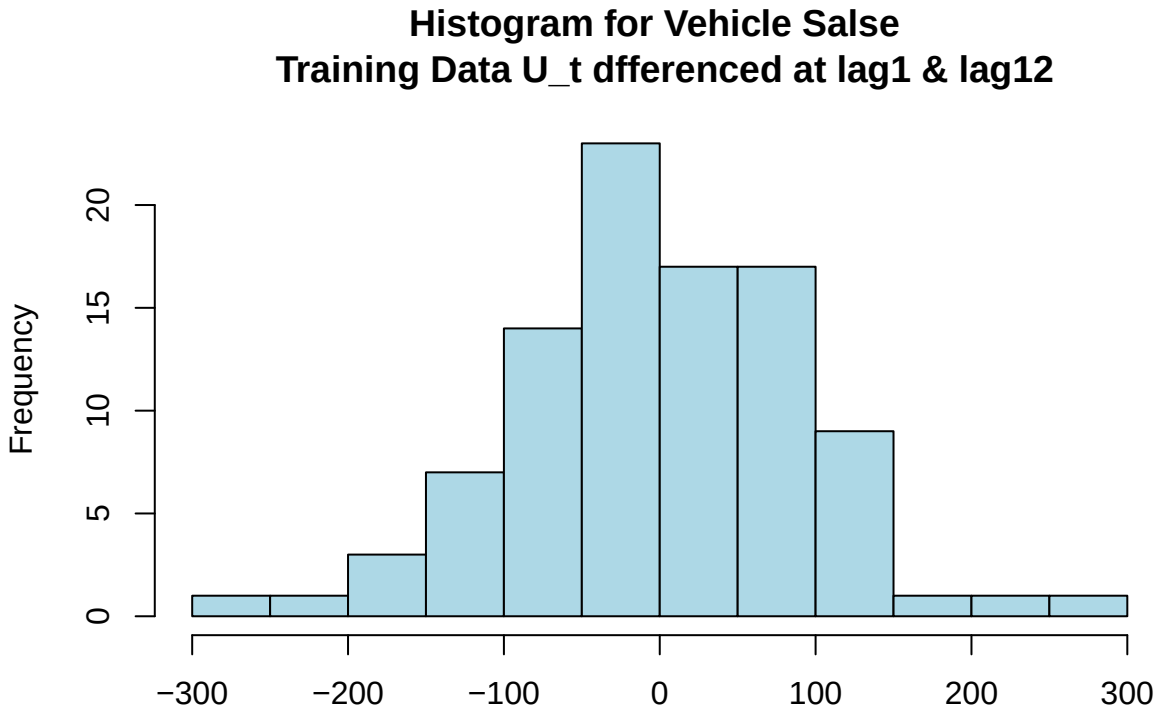


After differencing at lag 1, the time series plot does not show an increasing linear trend any more and the mean decreases to 8.9 which is closed to 0. It still shows some seasonality in the data. The variance decreases from 50523.04 to 37037.8 and seems to be constant. The histogram seems to be somewhat left-skewed.

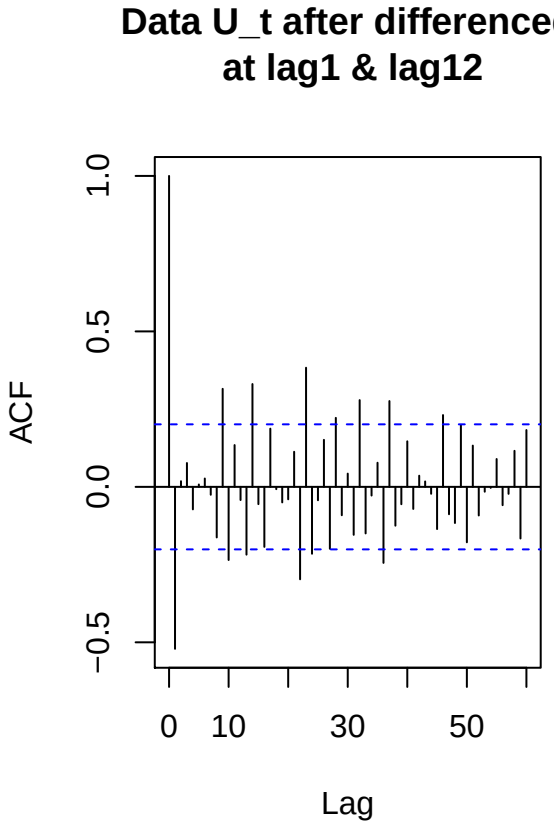


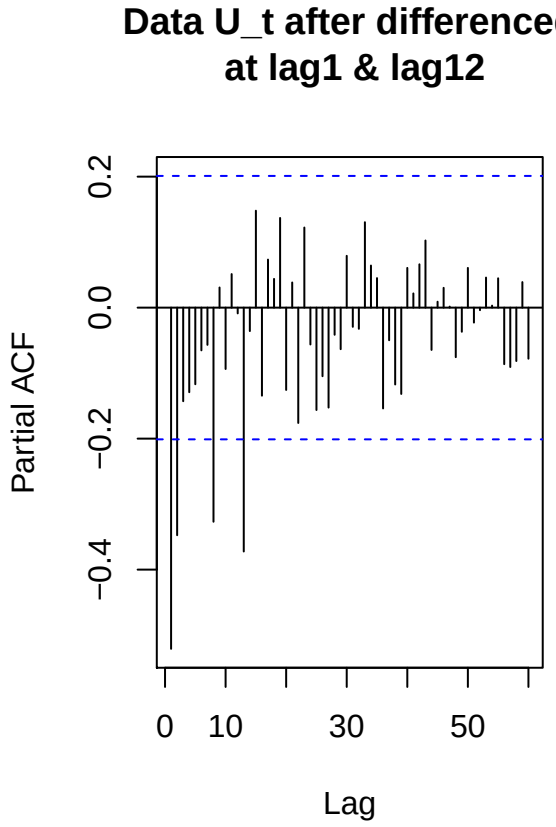
ACF values remain large at large lags, specifically at lags that are multiple of 12 which indicates seasonality. PACF values also remain large. Since there is still seasonality, the data is differenced at lag 12 to remove seasonality.





After differencing at lag 12, the plot resemble a stationary time series data plot. There is no linear trend or obvious seasonality. Now it has a mean of -1.0. The variance looks constant and decreases from 37037.8 to 8359.72. The histogram seems to be symmetric and has a normal distribution. Looking at these two plots, I believe the data is a stationary time series now. Next, I checked the ACF and PACF values to confirm it.

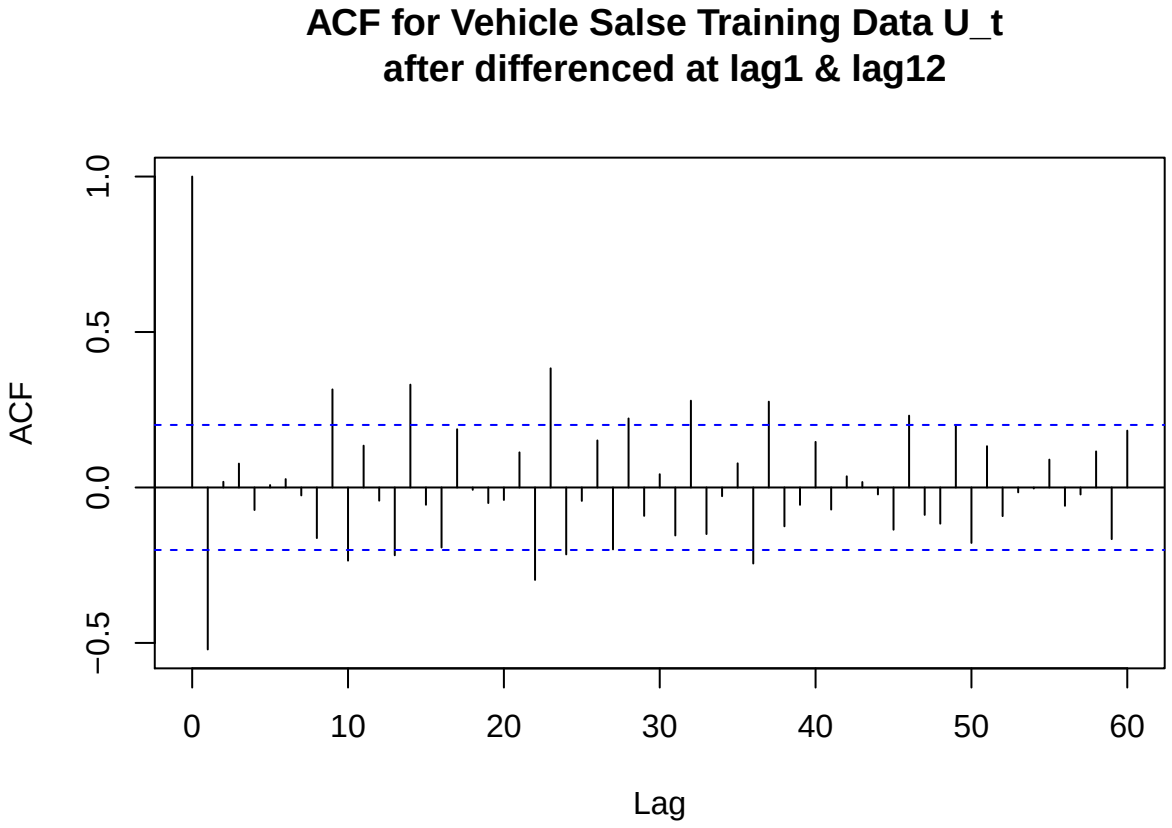


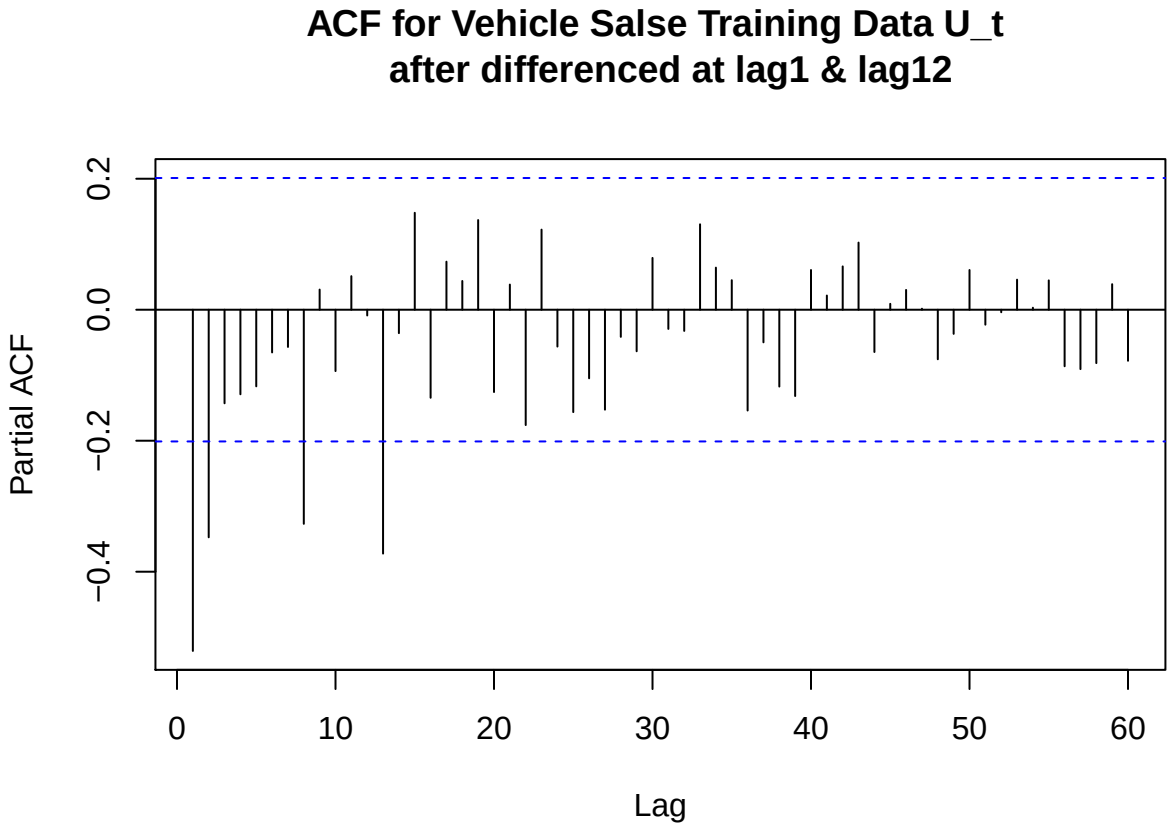


ACF values remain large at large lags, but there is no obvious periodic pattern. PACF only have values outside the confidence interval at small lags. Since the ACF still remain large which indicates the data is still not stationary after differencing at lag 1 and lag 12, I tried differecing the data at 12 one more time to see if ACF values will look better. After differencing at lag 12 twice, the variance increases from 8359.72 to 17672.93 which indicates overdifferencing. Since the plot and the histogram indicate the data is stationary and differecing the data at 12 one more time leads to an increase of variance, I believe differencing the data at lag 1 and then at lag 12 once is sufficient. Large ACF values at large lags after differencing might due to the nature of the data itself. I believe it is safe to proceed to identify models with the differencing data.

Model Identification

To identify possible models, we need to look at the differenced data's ACF and PACF.





ACF is significant at lag 13, 23, 36. These numbers are approximate multiple of 12, so Q could be 1,2,3. Within one period, ACF is significant at 1, 9. Since 9 is 3 lag away from 12, so q could be 3. Thus, possible q could be 1,3 and possible Q could be 2,3. PACF is significant at 1,2,8, 13, so P could be 1 and p could be 2,4. Therefore, all the possible models are: SRIMA(p,d,q)x(P,D,Q)s with s=12, D=1, d=1, Q=2,3, q=1,2,3, P=1, p=2,4.

Model Fitting

Start with Pure SAR Models

```
library(qpcR)
#SAR(p=2,d=1,q=0)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(2,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1     ar2     sar1
##       -0.7627  -0.380  -0.2296
## s.e.   0.0989   0.096   0.1110
##
## sigma^2 estimated as 4999:  log likelihood = -540.02,  aic = 1088.03
```

```
AICc(arima(vehicle, order=c(2,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML'))
```

```
## [1] 1088.263
```

I start with p=2 and P=1. The confidence intervals for all the coefficients in this model don't contain 0 which means all the coefficients are significant.

```
#SAR(p=4,d=1,q=0)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(4,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 0), seasonal = list(order = c(1, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4    sar1
##       -0.7889  -0.4794  -0.1624  -0.0942  -0.180
## s.e.   0.1030   0.1312   0.1420   0.1073   0.121
##
## sigma^2 estimated as 4941:  log likelihood = -539.36,  aic = 1090.71
```

```
AICc(arima(vehicle, order=c(4,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')) #1091.30
```

```
## [1] 1091.303
```

Then I check model with p=4 and the AICc value for it is greater than the model with p=2, so in this step the model with the lowest AICc model is SARIMA(2,1,0)x(1,1,0)s=12. Next, I add MA component q=1 into model SARIMA(2,1,0)x(1,1,0)s=12.

```
#SAR(p=2,d=1,q=1)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     ar2      ma1     sar1
##       0.0125  0.0950  -0.8514  -0.1053
## s.e.  0.1337  0.1255   0.0830   0.1112
##
## sigma^2 estimated as 4775:  log likelihood = -537.83,  aic = 1085.67
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')) #1086.05
```

```
## [1] 1086.053
```

After adding q=1, the AICc decreases, but the confidence interval for ar1, ar2 and sar1 all contain 0, so I set them to be 0 one at a time.

```
#model 6
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,NA,NA,NA), method =
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     fixed = c(0, NA, NA, NA), method = "ML")
##
## Coefficients:
```

```
##       ar1     ar2      ma1     sar1
##         0  0.0908  -1.1816  -0.1067
## s.e.    0  0.1181   0.0921   0.1101
##
## sigma^2 estimated as 3420:  log likelihood = -537.84,  aic = 1083.67
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,NA,NA,NA),metho
```

```
## [1] 1084.062
```

```
arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ma1     sar1
##       -0.0326  -0.8070  -0.0991
## s.e.   0.1406   0.0995   0.1121
##
## sigma^2 estimated as 4806:  log likelihood = -538.11,  aic = 1084.21
```

```
AICc(arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0),period = 12), method = 'ML')) #1084.443
```

```
## [1] 1084.443
```

```
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(0,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     ar2      ma1
##       0.0300  0.0834  -0.8622
## s.e.  0.1304  0.1238   0.0794
##
## sigma^2 estimated as 4826:  log likelihood = -538.28,  aic = 1084.55
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(0,1,0),period = 12), method = 'ML')) #1086.053
```

```
## [1] 1084.781
```

The AICc for these three models all have lower AICc than the first one. Since in these three models, the confidence intervals for other coeffients except ma1 contain 0, I set different combination of coefficients to 0 to see how the AICc values change.

```
# Model 4
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,NA,NA,0), method =
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     fixed = c(0, NA, NA, 0), method = "ML")
##
## Coefficients:
```

```
##       ar1     ar2      ma1  sar1
##         0  0.0732  -1.1765     0
## s.e.    0  0.1177   0.0947     0
##
## sigma^2 estimated as 3489:  log likelihood = -538.3,  aic = 1082.6
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,NA,NA,0),method
```

```
## [1] 1082.991
```

```
# Model 5 / Model D
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(NA,0,NA,0), method =
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     fixed = c(NA, 0, NA, 0), method = "ML")
##
## Coefficients:
##           ar1  ar2     ma1  sar1
##       -0.0060    0  -0.826     0
## s.e.   0.1327    0   0.088     0
##
## sigma^2 estimated as 4850:  log likelihood = -538.49,  aic = 1082.99
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(NA,0,NA,0),method
```

```
## [1] 1083.377
```

```
#Model 7
arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(1, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     method = "ML")
##
## Coefficients:
##           ar1      ma1     sar1
##       -0.0326  -0.8070  -0.0991
## s.e.   0.1406   0.0995   0.1121
##
## sigma^2 estimated as 4806:  log likelihood = -538.11,  aic = 1084.21
```

```
AICc(arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0),period = 12),method = 'ML')) ##1084.443
```

```
## [1] 1084.443
```

```
# Model 3
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,0,NA,0), method = '
```

```
##
## Call:
## arima(x = vehicle, order = c(2, 1, 1), seasonal = list(order = c(1, 1, 0), period = 12),
##     fixed = c(0, 0, NA, 0), method = "ML")
##
## Coefficients:
##       ar1  ar2      ma1  sar1
```

```
##         0    0  -1.2070     0
## s.e.    0    0   0.0976     0
##
## sigma^2 estimated as 3329:  log likelihood = -538.5,  aic = 1080.99
```

```
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,0,NA,0),method =
```

```
## [1] 1081.379
```

The model with the lowest AICc value for this step is SARIMA(2,1,1)x(1,1,0)s=12 with ar1, ar2 and sar1 set to 0. This model is equivalent to SARIMA(0,1,1)x(0,1,0)s=12. Next I add Q = 2 to this model.

```
# Model 1 / Model A (lowest AICc)
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.7566  -0.4936  -0.5059
## s.e.   0.0712   0.4330   0.2328
##
## sigma^2 estimated as 3463:  log likelihood = -532.76,  aic = 1073.52
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),method = 'ML')) #1073.747
```

```
## [1] 1073.747
```

```
# Model 2 / Model C
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),fixed=c(NA,0, NA),method = 'ML
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2), period = 12),
##     fixed = c(NA, 0, NA), method = "ML")
##
## Coefficients:
##           ma1  sma1     sma2
##       -0.8183     0  -0.2787
## s.e.   0.0650     0   0.1359
##
## sigma^2 estimated as 4555:  log likelihood = -536.47,  aic = 1078.93
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),fixed=c(NA,0, NA), method
```

```
## [1] 1079.163
```

After adding Q=2, the AICc decreases. In the first model, the confidence interval for sma1 contain 0, so I set it 0 and check the AICc. AICc increases. Therefore, the model with lowest AICc value in this part is SAR(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12.

Start with Pure SMA Models

```
#start with pure MA
arima(vehicle, order=c(0,1,3),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 3), seasonal = list(order = c(0, 1, 3), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     ma2     ma3     sma1     sma2     sma3
##       -0.7900  0.0165  0.0031  -0.3933  -0.3223  -0.2843
## s.e.   0.1063  0.1525  0.1014   0.1955   0.1741   0.1547
##
## sigma^2 estimated as 3451:  log likelihood = -531.01,  aic = 1076.02
```

```
AICc(arima(vehicle, order=c(0,1,3),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) #1076.85
```

```
## [1] 1076.855
```

I start with q=3 and Q=3 and AICc for this model is 1076.855. Then I remove coefficients in the moving average part one at a time.

```
#model 12
arima(vehicle, order=c(0,1,2),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 2), seasonal = list(order = c(0, 1, 3), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     ma2     sma1     sma2     sma3
##       -0.7907  0.0197  -0.3940  -0.3219  -0.2840
## s.e.   0.1038  0.1069   0.1948   0.1743   0.1533
##
## sigma^2 estimated as 3451:  log likelihood = -531.01,  aic = 1074.02
```

```
AICc(arima(vehicle, order=c(0,1,2),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) # 1074.6
```

```
## [1] 1074.612
```

ma3 is removed and AICc decreases from 1076.855 to 1074.612.

```
#model 8 / Model B
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1     sma2     sma3
##       -0.7760  -0.3912  -0.3275  -0.2813
## s.e.   0.0663   0.1940   0.1722   0.1529
##
## sigma^2 estimated as 3452:  log likelihood = -531.03,  aic = 1072.06
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) #1072.44
```

```
## [1] 1072.448
```

ma2 is removed and AICc decreases from 1074.612 to 1072.448. In this model, ma1 is significant, so I stop removing coefficient from the moving average part. Next, I start removing coefficients from the seasonal moving average part.

```
#model10
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 2), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1     sma1     sma2
##       -0.7566  -0.4936  -0.5059
## s.e.   0.0712   0.4330   0.2328
##
## sigma^2 estimated as 3463:  log likelihood = -532.76,  aic = 1073.52
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12), method = 'ML')) #1073.74
```

```
## [1] 1073.747
```

The AICc value increases after Q changes from 3 to 2. Therefore, I stop with Q=3.
Since in model with the lowest AICc SRIMA(0,1,1)x(0,1,3)s=12, confidence interval for sma2 contain 0, I set it to 0 to check AICc value.

```
# Model 14
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), fixed = c(NA,NA,0,NA), method
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     fixed = c(NA, NA, 0, NA), method = "ML")
##
## Coefficients:
##           ma1    sma1  sma2     sma3
##       -0.7776  -0.322     0  -0.4063
## s.e.   0.0653   0.151     0   0.1379
##
## sigma^2 estimated as 3917:  log likelihood = -532.89,  aic = 1073.78
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), fixed = c(NA,NA,0,NA), m
```

```
## [1] 1074.173
```

AICc increases from 1072.448 to 1074.173. Thus, in this step the model with the lowest AICc is SRIMA(0,1,1)x(0,1,3)s=12. Next, I add seaonal autoregressive component P=1 into SRIMA(0,1,1)x(0,1,3)s=12.

```
#model 13
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(0, 1, 1), seasonal = list(order = c(1, 1, 3), period = 12),
##     method = "ML")
##
## Coefficients:
##           ma1    sar1     sma1     sma2     sma3
```

```
##       -0.7762  0.0063  -0.3964  -0.3243  -0.2792
## s.e.   0.0668  0.3039   0.3198   0.2283   0.1852
##
## sigma^2 estimated as 3453:  log likelihood = -531.03,  aic = 1074.06
```

```
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')) #1074.64
```

```
## [1] 1074.647
```

The AICc value increases after I added P=1 into the model, so I stopped adding new component to it. Instead, I add autoregressive component q = 4 into the model with the lowest AICc and then again remove the coefficient in the autoregressive part one at a time to see how the AICc value changes.

```
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     method = "ML")
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ma1     sma1     sma2     sma3
##       0.0279  0.0494  0.1338  -0.0269  -0.8169  -0.4423  -0.3049  -0.2527
## s.e.  0.1493  0.1390  0.1290   0.1217   0.1110   0.2054   0.1722   0.1623
##
## sigma^2 estimated as 3401:  log likelihood = -530.41,  aic = 1078.83
```

```
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')) #1082.66
```

```
## [1] 1082.664
```

In this model, the coefficient ar4 is very closed to 0 and the confidence interval for it contains 0, so I set it to 0 in the next model and check the AICc.

```
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(NA,NA,NA,0,NA,NA,NA,
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     fixed = c(NA, NA, NA, 0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##          ar1     ar2     ar3  ar4      ma1     sma1     sma2     sma3
##       0.0338  0.0541  0.1375    0  -0.8266  -0.4328  -0.3037  -0.2637
## s.e.  0.1456  0.1364  0.1275    0   0.0995   0.2000   0.1714   0.1542
##
## sigma^2 estimated as 3404:  log likelihood = -530.44,  aic = 1076.88
```

```
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(NA,NA,NA,0,NA,NA
```

```
## [1] 1078.33
```

AICc decreases for this model and the confidence interval for ar2 contains 0, so set ar2 to 0 in the next model.

```
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(NA,0,NA,0,NA,NA,NA,NA
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     fixed = c(NA, 0, NA, 0, NA, NA, NA, NA), method = "ML")
```

```
##
## Coefficients:
##          ar1  ar2     ar3  ar4      ma1     sma1     sma2     sma3
##       0.0068    0  0.1172    0  -0.7993  -0.4211  -0.3069  -0.2721
## s.e.  0.1383    0  0.1182    0   0.0900   0.1926   0.1699   0.1532
##
## sigma^2 estimated as 3412:  log likelihood = -530.51,  aic = 1075.03
```

```
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(NA,0,NA,0,NA,NA
```

```
## [1] 1076.484
```

AICc decreases for this model and the confidence interval for ar1 contains 0, so set ar1 to 0 in the next model.

```
#model11
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(0,0,NA,0,NA,NA,NA,NA)
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     fixed = c(0, 0, NA, 0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2     ar3  ar4      ma1     sma1     sma2     sma3
##         0    0  0.1158    0  -1.2560  -0.4219  -0.3058  -0.2723
## s.e.    0    0  0.1148    0   0.1075   0.1925   0.1682   0.1526
##
## sigma^2 estimated as 2163:  log likelihood = -530.52,  aic = 1073.03
```

```
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(0,0,NA,0,NA,NA,
```

```
## [1] 1074.487
```

AICc decreases for this model and the confidence interval for ar4 contains 0, so set ar4 to 0 in the next model.

```
#model 9
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(0,0,0,0,NA,NA,NA,NA)
```

```
##
## Call:
## arima(x = vehicle, order = c(4, 1, 1), seasonal = list(order = c(0, 1, 3), period = 12),
##     fixed = c(0, 0, 0, 0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2  ar3  ar4      ma1     sma1     sma2     sma3
##         0    0    0    0  -1.2887  -0.3913  -0.3275  -0.2813
## s.e.    0    0    0    0   0.1100   0.1940   0.1722   0.1529
##
## sigma^2 estimated as 2078:  log likelihood = -531.03,  aic = 1072.06
```

```
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(0,0,0,0,NA,NA,N
```

```
## [1] 1073.514
```

AICc decreases for this model. This model has the lowest AICc and is equivalent to the model with the lowest AICc that checked previously. Therefore in this part, the model with the lowest AICc is SRIMA(0,1,1)x(0,1,3)s=12.

Now I obtain two models with the lowest AICc value from both parts:
Model A:

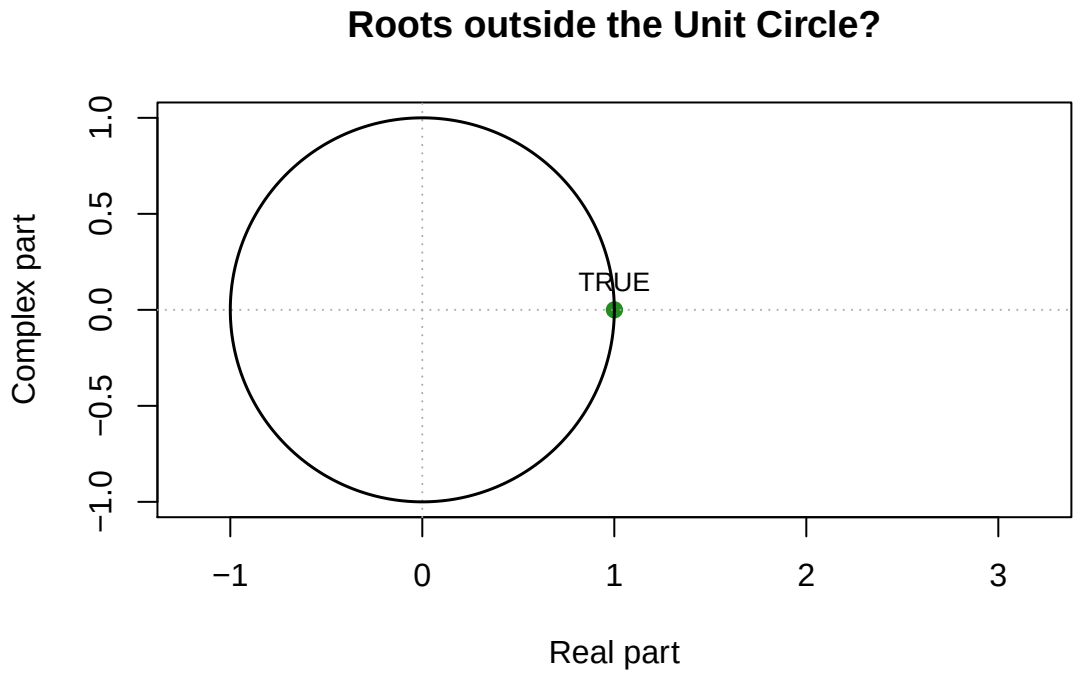
SARIMA(0,1,1)×(0,1,2)s=12 with AICc = 1073.747
$\nabla_1\nabla_{12}U_t = (1 - 0.7566B)(1 - 0.4936B^{12} - 0.5059B^{24})Z_t$

Model B: SRIMA(0,1,1)x(0,1,3)s=12 with AICc = 1072.448
$\nabla_1\nabla_{12}U_t = (1 - 0.776B)(1 - 0.3912B^{12} - 0.3275B^{24} - 0.2813B^{36})Z_t$

Next I check whether the models that I obtain with the lowest AICc values from both parts have unit roots.

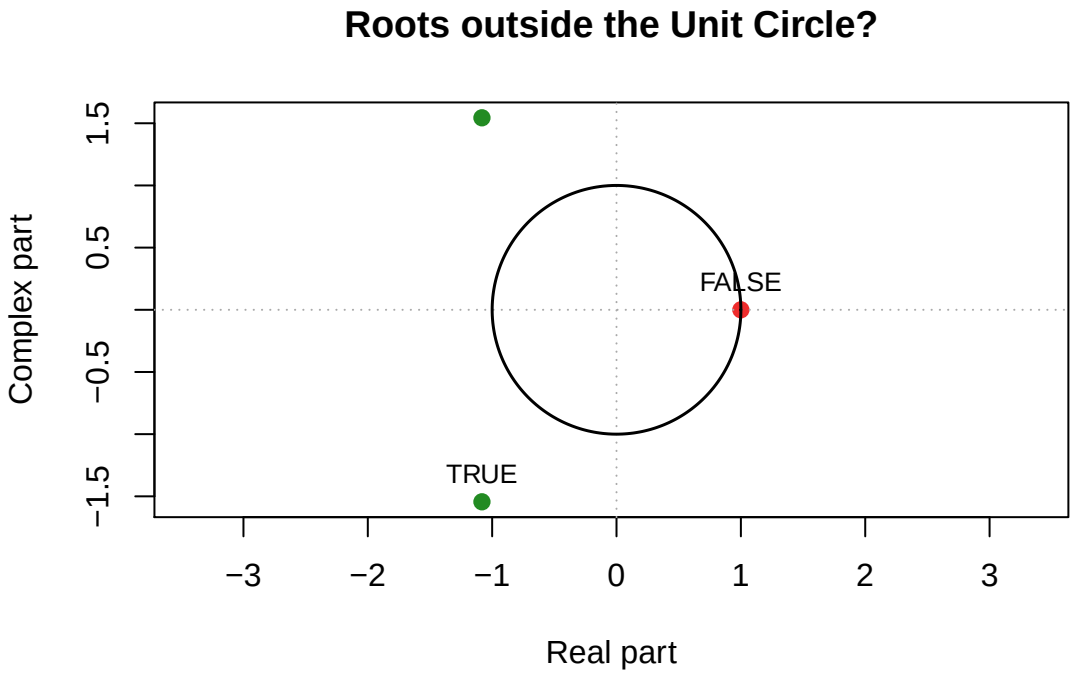
```
library(UnitCircle)
#Model A
uc.check(pol_=c(1, -0.4936  -0.5059), plot_output = TRUE)
```

```
##     real complex outside
## 1 1.0005       0    TRUE
## *Results are rounded to 6 digits.
```




```
#Model B
uc.check(pol_=c(1, -0.3912, -0.3275, -0.2813), plot_output = TRUE)
```

```
##        real   complex outside
## 1  1.000000  0.000000   FALSE
## 2 -1.082119  1.544002    TRUE
## 3 -1.082119 -1.544002    TRUE
## *Results are rounded to 6 digits.
```



Since both models have units roots in the seasonal moving average part, these two models can not be used for forecasting. In theory having units roots in the seasonal moving average part means the data is overdifferenced in the seasonal part. However, the ACF values after differencing at lag 1 are significant at lags that are multiple of 12 which indicates a need to difference the data at lag 12 to remove seasonality. After differencing the data at lag 12, the variance does get smaller compared to the variance of differenced data at lag 1. Therefore, I believe it's necessary to difference the data at lag 12 to remove seasonality. Note that overdifferencing of data indicated by the unit roots in the seasonal part is consistent with seeing large ACF values at large lags after differencing at lag 12. Therefore, this problem could be due to the nature of the dataset itself.

I proceed to check unit roots for models with low AICc values to find a model that could be used for forecasting. After checking unit roots of 14 models, 3 models do not have unit roots. These three models are:

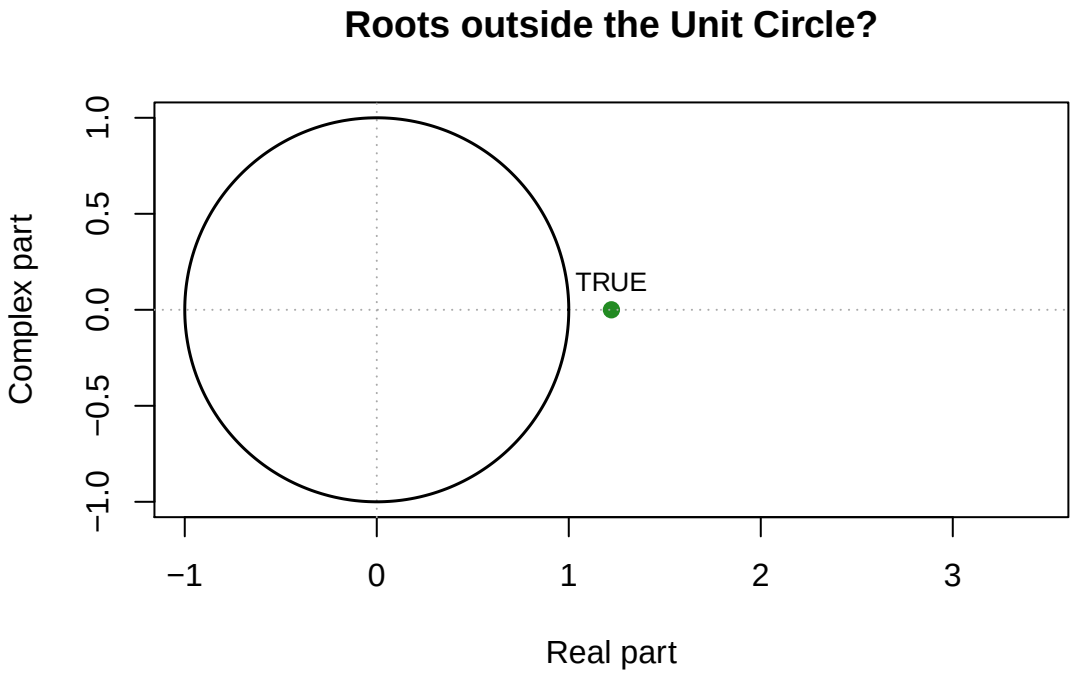
Model C (AICc = 1079.163):
SARIMA(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12 with sma1 = 0
$\nabla_1\nabla_{12}U_t = (1 - 0.8183B)(1 - 0.2787B^{24})Z_t$

Model D (AICc = 1083.377):
SARIMA(p=1,d=1,q=1)×(P=0,D=1,Q=0)s=12
$(1 + 0.006B)(1 - B)(1 - B^{12})U_t = (1 - 0.826B)Z_t$

Model E (AICc = 1084.443):
SARIMA(p=1,d=1,q=1)×(P=1,D=1,Q=0)s=12
$(1 + 0.0326B)(1 + 0.0991B^{12})(1 - B)(1 - B^{12})U_t = (1 - 0.807B)Z_t$

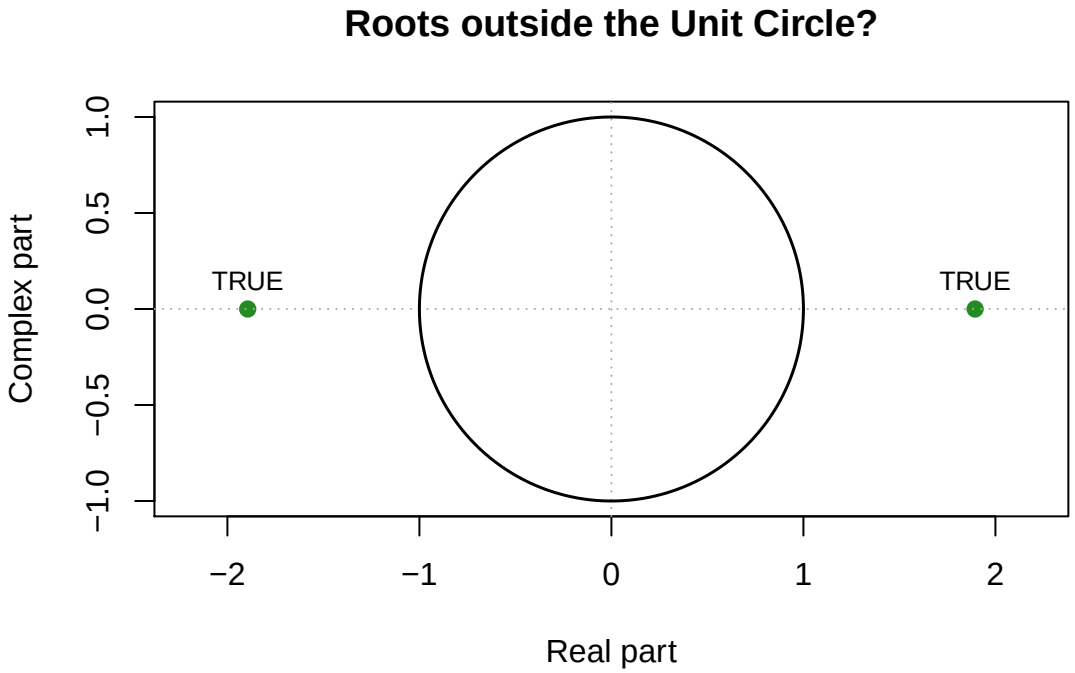
```
library(UnitCircle)
#Model C (2): SARIMA(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12 with sma1 = 0
uc.check(pol_=c(1, -0.8183), plot_output = TRUE)
```

```
##       real complex outside
## 1 1.222046       0    TRUE
## *Results are rounded to 6 digits.
```




```
uc.check(pol_=c(1, 0, -0.2787), plot_output = TRUE)
```

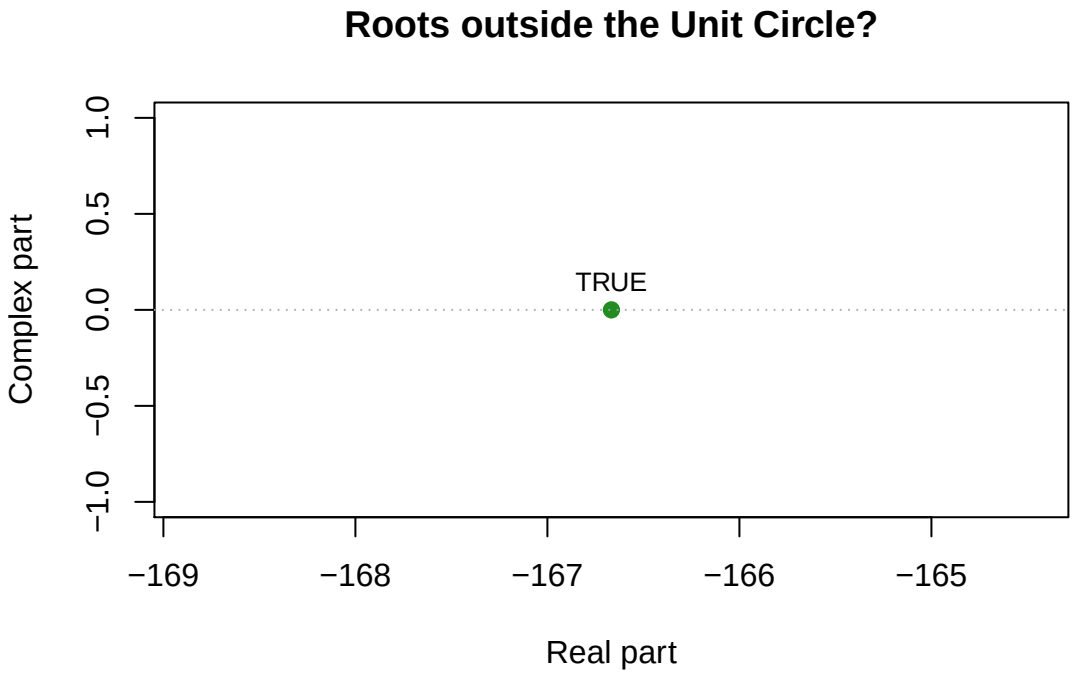
```
##        real complex outside
## 1  1.894225       0    TRUE
## 2 -1.894225       0    TRUE
## *Results are rounded to 6 digits.
```




```
#Model D (5): SARIMA(p=1,d=1,q=1)×(P=0,D=1,Q=0)s=12
uc.check(pol_=c(1, 0.0060,0), plot_output = TRUE)
```

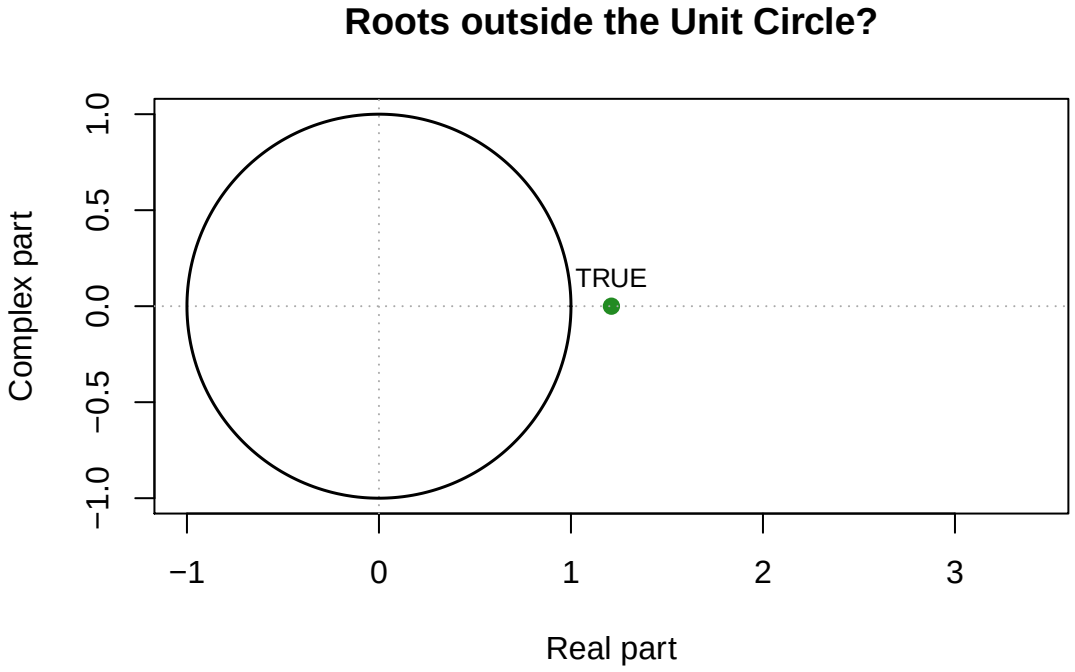
```
##        real complex outside
## 1 -166.6667       0    TRUE
```

```
## *Results are rounded to 6 digits.
```




```
uc.check(pol_=c(1, -0.826,0), plot_output = TRUE)
```

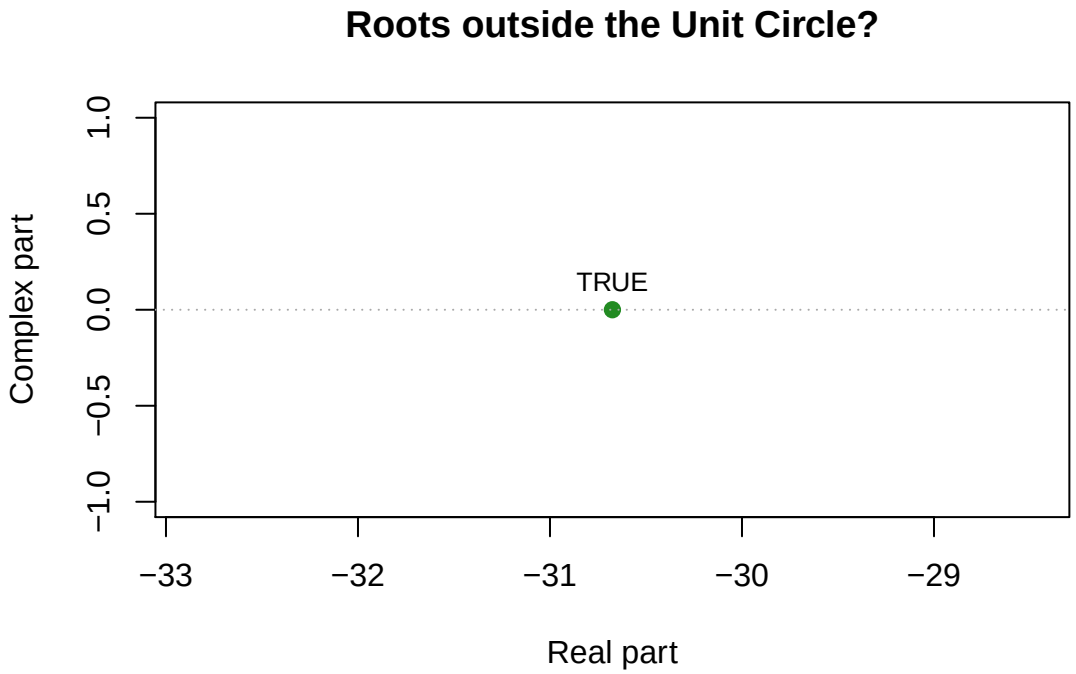
```
##       real complex outside
## 1 1.210654       0    TRUE
## *Results are rounded to 6 digits.
```




```
#Model E (7): SARIMA(p=1,d=1,q=1)×(P=1,D=1,Q=0)s=12
uc.check(pol_=c(1, 0.0326), plot_output = TRUE)
```

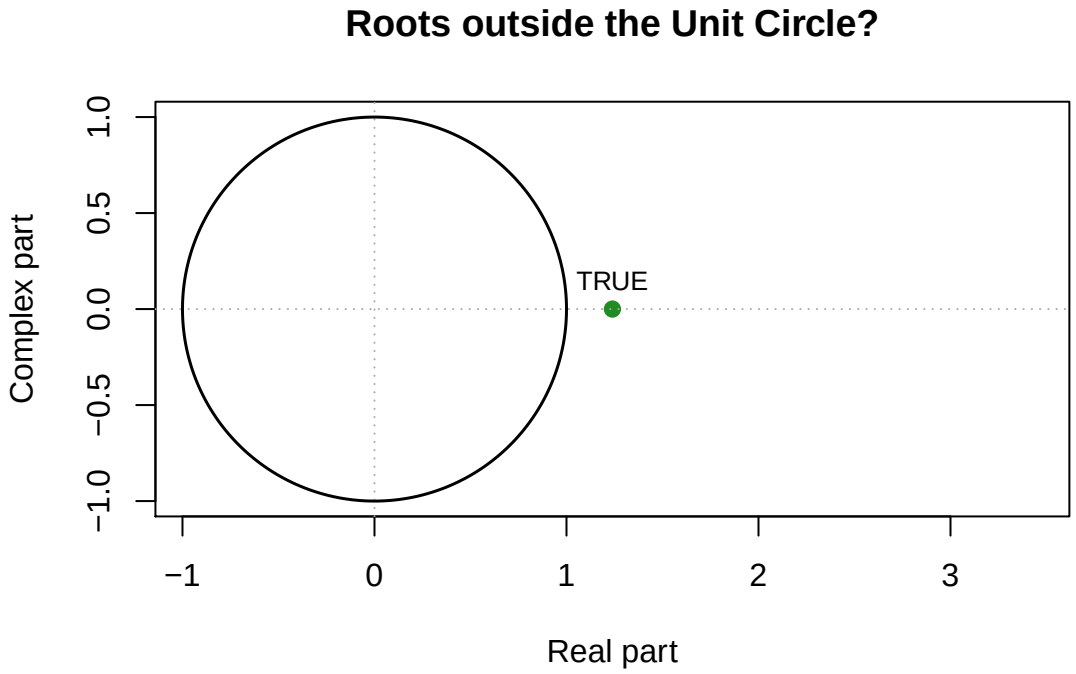
```
##       real complex outside
```

```
## 1 -30.67485       0    TRUE
## *Results are rounded to 6 digits.
```




```
uc.check(pol_=c(1, -0.8070), plot_output = TRUE)
```

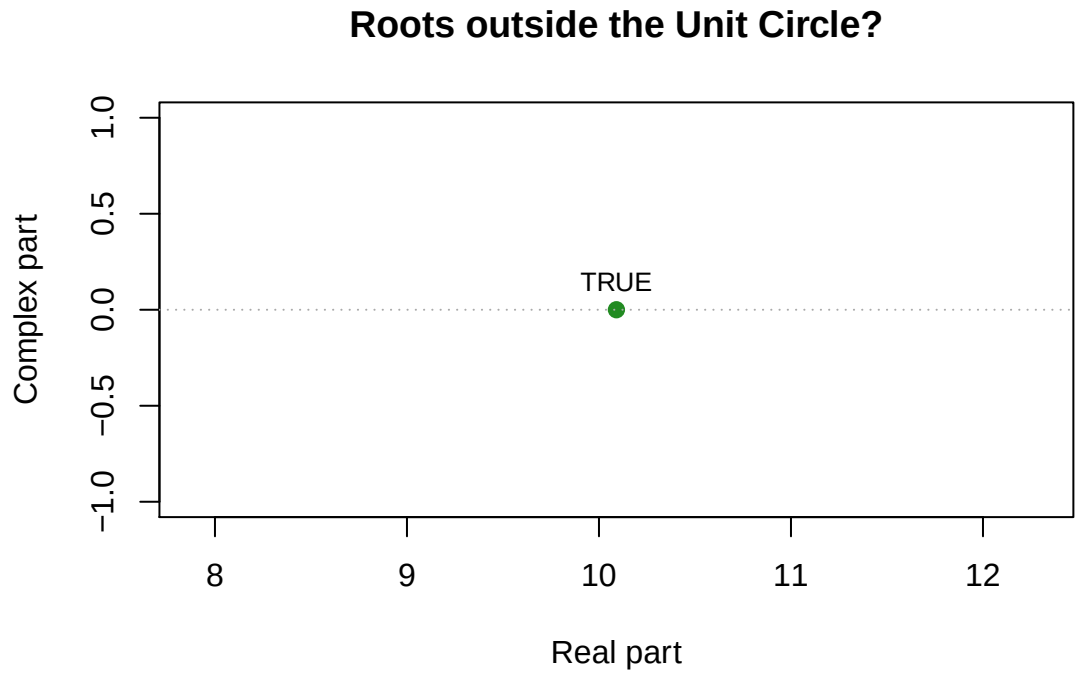
```
##       real complex outside
## 1 1.239157       0    TRUE
## *Results are rounded to 6 digits.
```




```
uc.check(pol_=c(1, -0.0991), plot_output = TRUE)
```

```
##       real complex outside
```

```
## 1 10.09082       0    TRUE
## *Results are rounded to 6 digits.
```

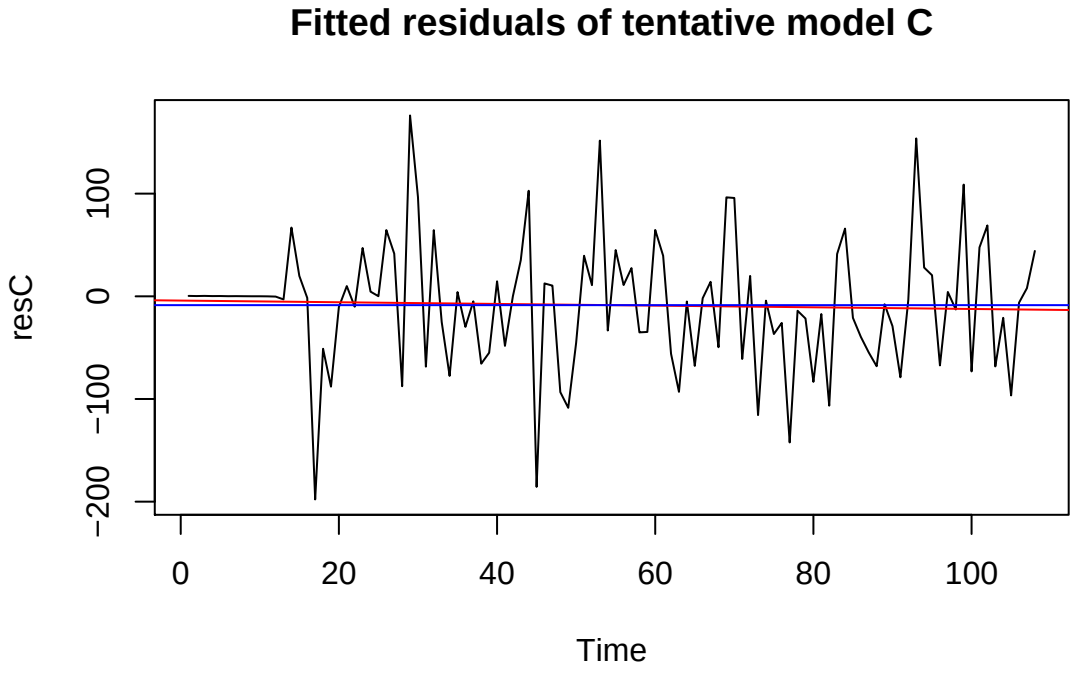


Diagnostic Checking on Residuals

Diagnostic checking is done for model C and model D because they have lower AICc values.

Model C: $\nabla_1 \nabla_{12} U_t = (1 - 0.8183B)(1 - 0.2787B^{24})Z_t$

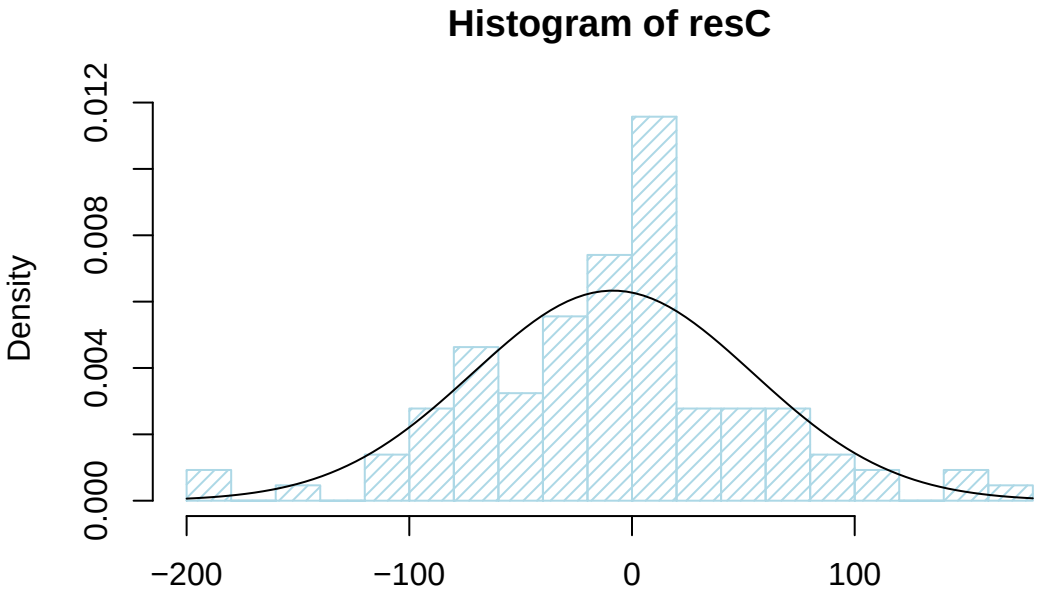
If the model fits the data well, the residuals of the model should resemble gaussian white noise WN(0,1).



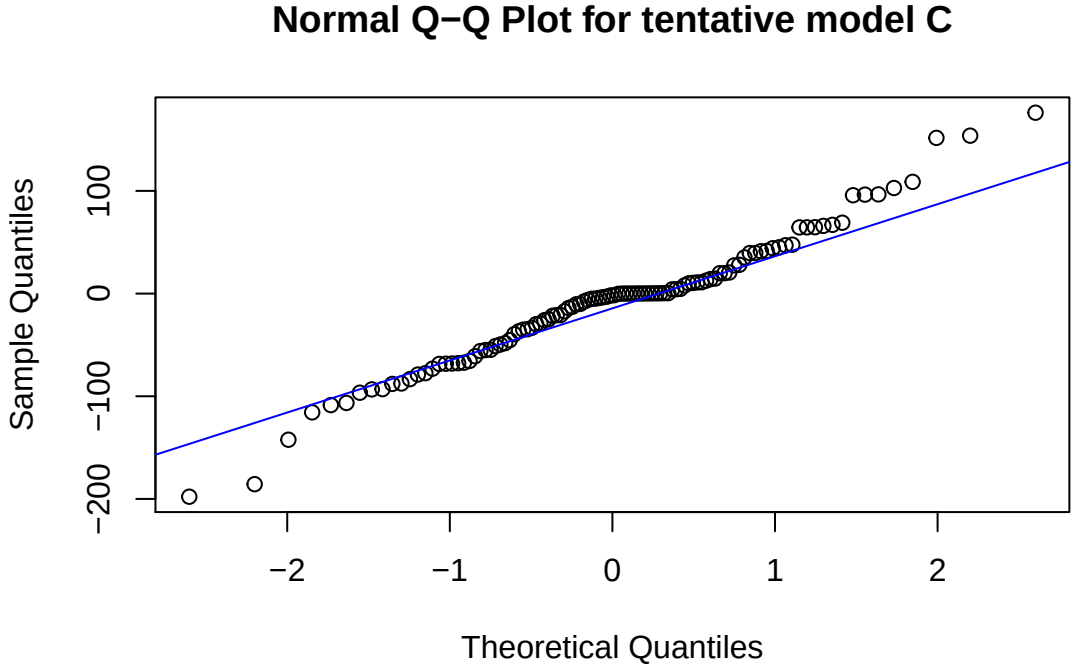
In the time series plot of the residuals, it shows a slight downward trend but no seasonality or change of variance.

```
## [1] -8.608933
```

```
## [1] 63.005
```



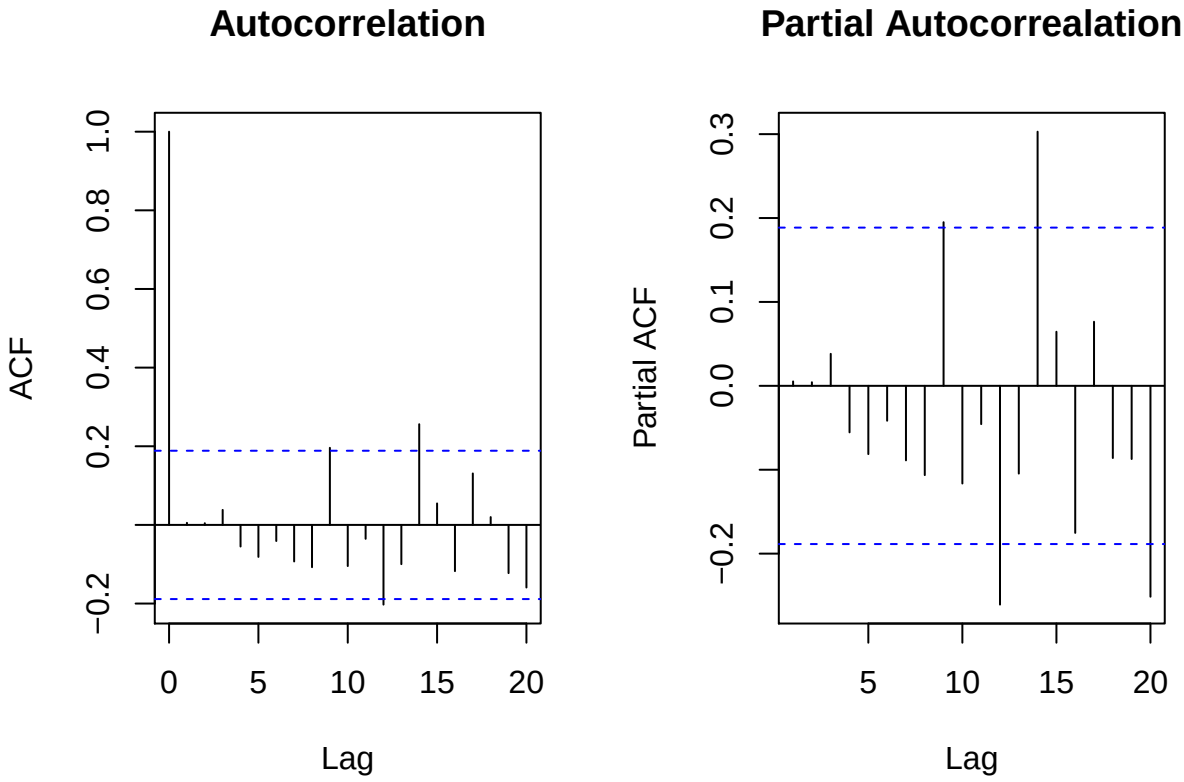
The histogram of the residuals does not show a normal distribution due to the spike in the center and some extreme values on the tails. The mean of residuals is -8.6 and variance is 63.005. The mean and the variance of the residuals indicate the residuals are not gaussian white noise.




```
##
##  Shapiro-Wilk normality test
##
## data:  resC
```

```
## W = 0.97346, p-value = 0.0293
```

Most points in the the Q-Q plot follow the diagonal line, but they do not follow a straight line and some points on both ends are very far away from the diagonal line. We will check with the Shapiro-Wilk normality test. The p-value of the Shapiro-Wilk normality test is less than 0.05 which means we have to reject the assumption of normality for model C. In short, the residuals are not normally distributed and the assumption of normality of this model is violated.



Some sample ACFs and PACFs of redisulas are outside the confidence interval at large lags. They don't resemble white noise.

```
##
##  Box-Pierce test
##
## data:  resC
## X-squared = 8.9044, df = 8, p-value = 0.3504
```

```
##
##  Box-Ljung test
##
## data:  resC
## X-squared = 9.8082, df = 8, p-value = 0.2787
```

```
##
##  Box-Ljung test
##
## data:  resC^2
## X-squared = 2.548, df = 10, p-value = 0.9902
```

Both Box-Pierce and Box-Ljung tests if the residuals resemble white noise. The third test (McLeod-Li test) tests residuals for non-linear dependence. The p-values of the all these three tests are greater than 0.05, so we fail to reject the null hypothesis that the residuals resemble white noise and the residuals do not have

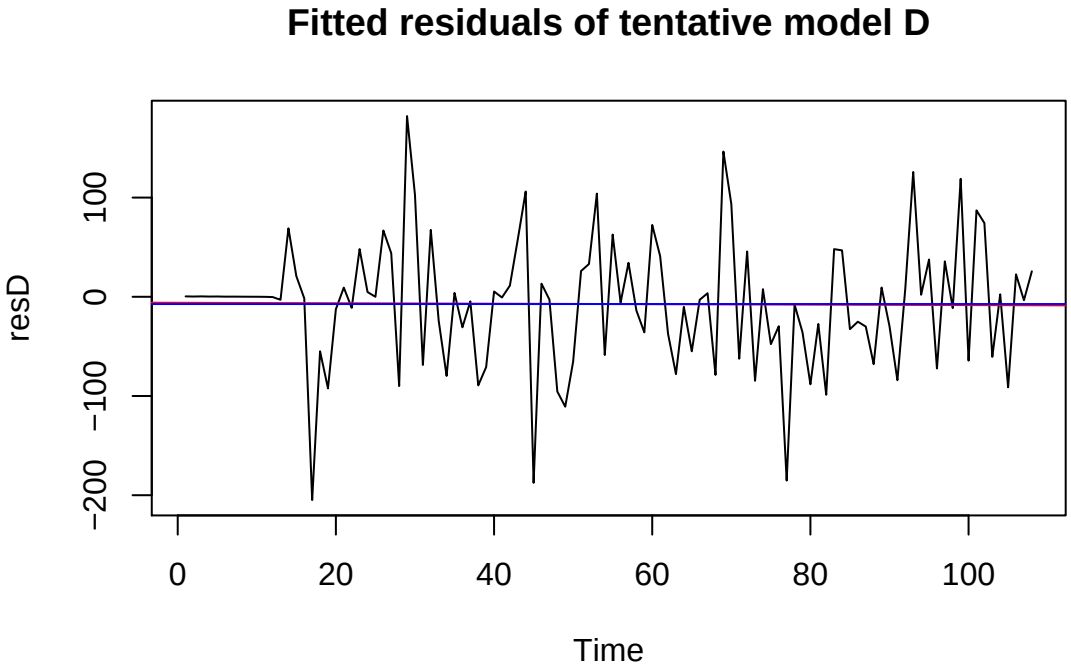
non-linear dependence.

```
##
## Call:
## ar(x = resC, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
##       1        2        3        4        5        6        7        8
## -0.0009   0.0774   0.0582  -0.0677  -0.1202  -0.0219  -0.0806  -0.2072
##       9       10       11       12       13       14       15       16
##  0.2121  -0.0940  -0.0360  -0.2908  -0.1174   0.2640   0.0228  -0.1691
##      17       18       19       20
##  0.0954  -0.0588  -0.0820  -0.2514
##
## Order selected 20  sigma^2 estimated as  3319
```

The residuals are fitted into AR model using Yule-Walker and the AR(20) is selected, which is not white noise.

To sum up, out of all the diagnostic procedures that are used to check whether the residuals of Model C resemble gaussian white noise, only the portmanteau tests indicate the residuals of Model C are white nosie and all other tests and plots indicate otherwise. Therefore, it can be concluded that Model C does not satisfy the required properties. Hence, Model C is not a good model to be used for forecasting.

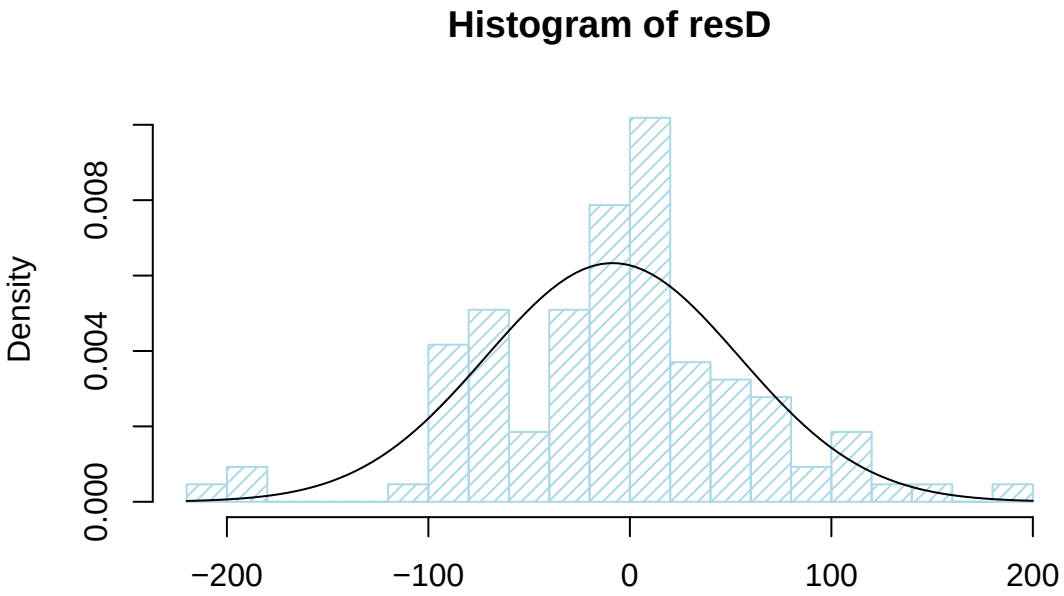
Model D: $(1 + 0.006B)(1 - B)(1 - B^{12})U_t = (1 - 0.826B)Z_t$



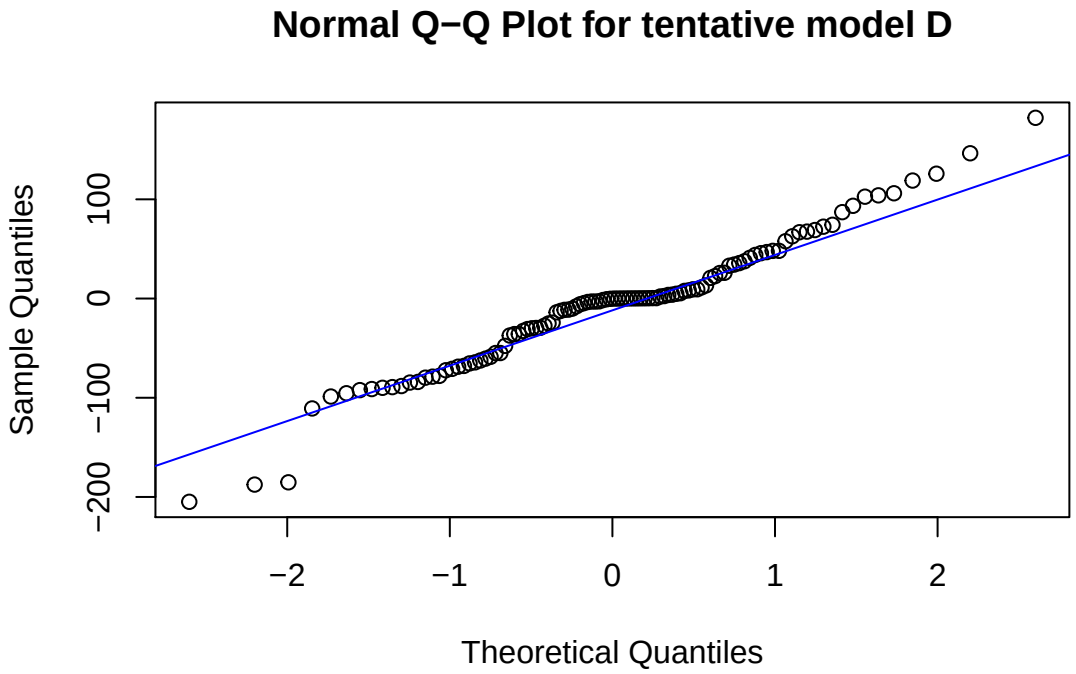
In the time series plot of the residuals, it does not show any visible trend, seasonality or change of variance.

```
## [1] -7.304511
```

```
## [1] 65.20786
```



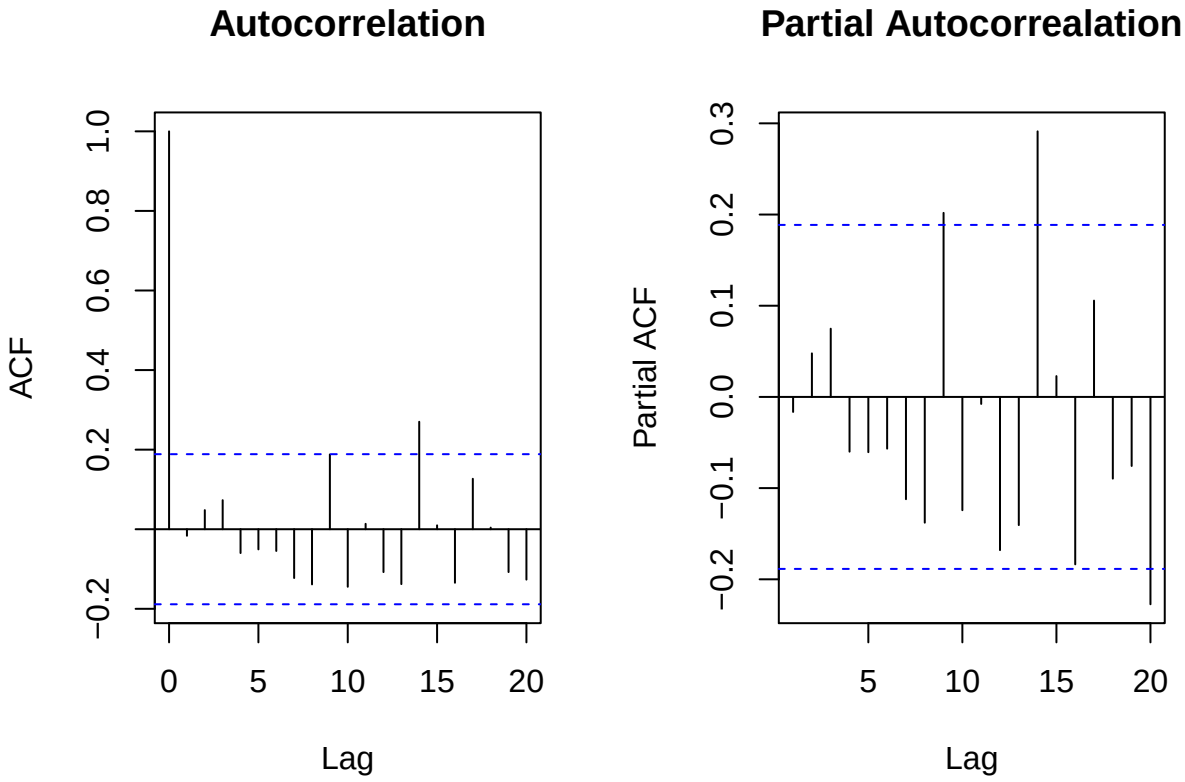
The histogram of the residuals does not show a normal distribution due to the spike in the center and some extreme values on the tails. The mean of residuals is -7.3 and variance is 65.20. The mean and the variance of the residuals indicate the residuals are not gaussian white noise.




```
##
##  Shapiro-Wilk normality test
##
## data:  resD
## W = 0.97068, p-value = 0.0173
```

Most points in the the Q-Q plot follow the diagonal line, but the points do not follow a straight line and some points on both ends are very far away from the diagonal line. We will check with the Shapiro-Wilk normality test. The p-value of the Shapiro-Wilk normality test is less than 0.05 which means we have to reject the assumption of normality for model D. In short, the residuals are not normally distributed and the

assumption of normality of this model is violated.



Some sample ACFs and PACFs of redisulas are outside the confidence interval at large lags. They don't resemble white noise.

```
##
##  Box-Pierce test
##
## data:  resD
## X-squared = 11.621, df = 8, p-value = 0.1689
```

```
##
##  Box-Ljung test
##
## data:  resD
## X-squared = 12.779, df = 8, p-value = 0.1197
```

```
##
##  Box-Ljung test
##
## data:  resD^2
## X-squared = 2.615, df = 10, p-value = 0.9891
```

Both Box-Pierce and Box-Ljung tests if the residuals resemble white noise. The third test (McLeod-Li test) tests residuals for non-linear dependence. The p-values of the all these three tests are greater than 0.05, so we fail to reject the null hypothesis that the residuals resemble white noise and the residuals do not have non-linear dependence.

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

```
##
## Order selected 0  sigma^2 estimated as  4252
```

The residuals are fitted into AR model using Yule-Walker and the AR(0) is selected, which is white noise.

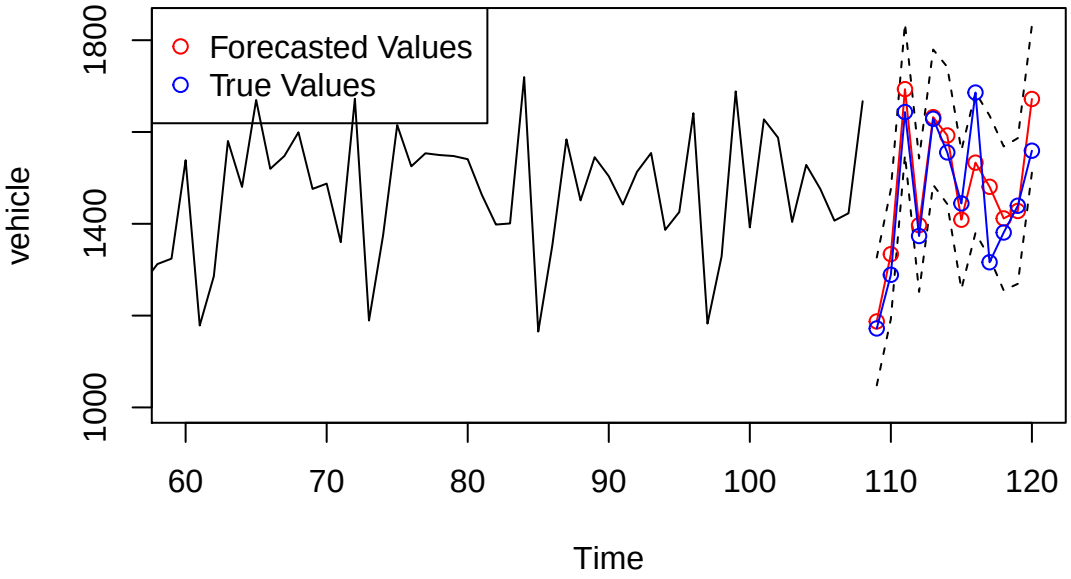
To sum up, out of all the diagnostic procedures that are used to check whether the residuals of Model D resemble gaussian white noise, the time series residuals plot, the portmanteau tests and the Yule-Walker estimation indicate the residuals of Model D are white noise, but all other tests indicate otherwise. Since Model D pass more tests than Model C, I think Model D is a better model. Although Model D does not satisfy all the required properties, there is no other methods learned in this lecture could be done to improve the model, so Model D is the best model out of all the models that are checked and will be used for forecasting.

Forecasting

Recall that the goal of this project is to predict monthly total vehicle sales in 2019 in the U.S. using monthly total vehicle sales data from 2010 to 2018. The model used for forecasting is: $(1+0.006B)(1-B)(1-B^{12})U_t = (1-0.826B)Z_t$

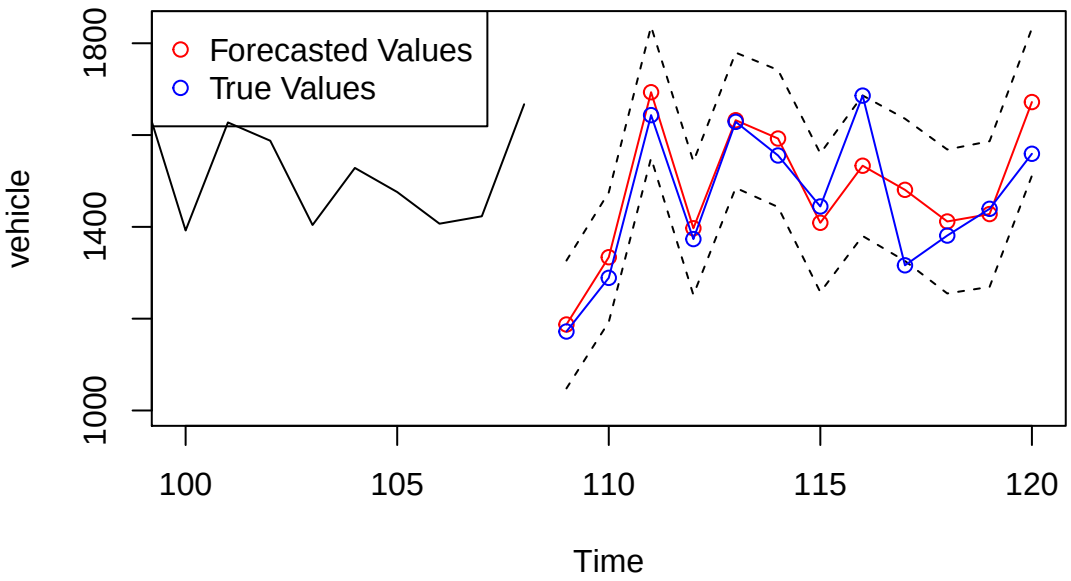
```
##           DATE TOTALNSA pred.values
## 109 2019-01-01 1172.004    1187.305
## 110 2019-02-01 1288.921    1333.893
## 111 2019-03-01 1643.423    1693.316
## 112 2019-04-01 1373.379    1396.986
## 113 2019-05-01 1628.757    1632.331
## 114 2019-06-01 1555.493    1592.465
## 115 2019-07-01 1444.447    1408.863
## 116 2019-08-01 1685.949    1533.192
## 117 2019-09-01 1316.202    1480.676
## 118 2019-10-01 1381.028    1411.737
## 119 2019-11-01 1439.322    1427.876
## 120 2019-12-01 1559.229    1671.899
```

This table shows the true values and the predicted values of monthly vehicle sales in 2019 in the U.S.



A graph with the prediction values (red points) is produced on the original series. We can see that all the true values are within the confidence interval and they are consistent with the trend and seasonality of the original data. Next, a graph focused on the prediction time period is produced in order to better compare

the predicted values and the true values.



From this graph, we can see the the forecasted values are very closed to the true values except in the month of August, September and December. Overall, Model D gives good predictions for the monthly vehicle sales in 2019.

Conclusion

The model obtained in this project is $(1+0.006B)(1-B)(1-B^{12})U_t = (1-0.826B)Z_t$. Although the model does not satisfy all the properties required for a forecasting model for the time series data, it successfully predicts the monthly total vehicle sales in 2019 in the U.S. using data from 2010 to 2018. Due to the nature of the data itself, some more advanced modeling techniques could be used to improve the model or build a better model so that all the available information in the data could be used to make predictions.

I would like to express my thanks to my PSTAT174 instructor Professor Raya Feldman for her guidance and support throughout my project. I would also like to thank the Federal Reserve Bank of St. Louis for make this dataset public.

References

U.S. Bureau of Economic Analysis, Total Vehicle Sales [TOTALNSA], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/TOTALNSA, May 26, 2021.

Appendix

```
#read in data
vehicle_df <- read.csv("TOTALNSA.csv", header = TRUE)

#partition dataset: train(2010-2018), test(2019)
vehicle_train <- vehicle_df[1:108, ]
vehicles_test <- vehicle_df[109:120, ]

#save train data into vehicle
vehicle <- vehicle_train[,2]
```

```r
#create a TS object
vehicle.ts <- ts(vehicle, start = c(2010,1), frequency = 12)
ts.plot(vehicle.ts, main = "Original Data")

#plot training data with trend and mean
plot.ts(vehicle, main = "Original Data")
fit <- lm(vehicle ~ as.numeric(1:length(vehicle)))
abline(fit, col = 'red')
abline(h = mean(vehicle), col = 'blue')

#plot histogram
hist(vehicle, col = 'light blue', xlab = '', main = 'Histogram for Vehicle Sales Trainning Data')

#ACF & PACF for original data
acf(vehicle, lag.max = 60, main = 'ACF of the Original Data')
pacf(vehicle, lag.max =60, main = 'PACF of the Original Data')

#mean and variance for original data
mean(vehicle)
var(vehicle)
```

```r
library(ggplot2)
library(ggfortify)
y <- ts(as.ts(vehicle), frequency = 12)
decomp <- decompose(y)
plot(decomp)
```

```r
#differenced at lag 1 to remove linear trend original data
diff_1 <- diff(vehicle, lag =1)
plot.ts(diff_1, main = 'Original Data differenced at lag 1')
Ofit.diff1 <- lm(diff_1 ~ as.numeric(1:length(diff_1)))
abline(Ofit.diff1, col = 'red')
abline(h = mean(diff_1), col = 'blue')

#variance after differencing at lag1 for original data
var(diff_1)
mean(diff_1)

#P/Acf for vehicle original data after differenced at lag 1
acf(diff_1, lag.max = 60, main = 'ACF for vehicle original data after differenced at lag 1' )
pacf(diff_1, lag.max = 60, main = 'ACF for vehicle original data after differenced at lag 1' )

#histogram
hist(diff_1, col = 'light blue', xlab = '', main = 'Histogram for original vehicle dfferenced at lag1')
```

```r
#differenced at lag 12 to remove seasonality for original data
diff_12 <- diff(diff_1, lag =12)
plot.ts(diff_12, main = 'Data differenced at lag 12 for orginal data')
Ofit.diff12 <- lm(diff_12 ~ as.numeric(1:length(diff_12)))
abline(Ofit.diff12, col = 'red')
abline(h = mean(diff_12), col = 'blue')

#check variance and mean after differencing at lag 1 &lag 12 for original data
var(diff_12)
```

```
mean(diff_12)

#P/ACF for data after differenced at lag1 & lag12
acf(diff_12, lag.max = 60, main = 'ACF for vehicle original data after differenced at lag1 & lag12' )
pacf(diff_12, lag.max = 60, main = 'ACF for vehicle original data after differenced at lag1 & lag12' )

#histogram
hist(diff_12, col = 'light blue', xlab = '', main = 'Histogram for original vehicle dfferenced at lag1 &
```

```
#differenced at lag 12 the second time
diff_12_1 <- diff(diff_12, lag =12)
var(diff_12_1)
```

```
library(qpcR)
#SAR(p=2,d=1,q=0)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(2,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(2,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')) #1088.26

#SAR(p=4,d=1,q=0)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(4,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(4,1,0),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')) #1091.30

#SAR(p=2,d=1,q=1)×(P=1,D=1,Q=0)s=12
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')) #1086.05
#ar1, 2, sar1 contain 0

#model 6
#set ar1 to 0
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,NA,NA,NA), method =
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,NA,NA,NA),metho
#ar2, sar1 contain 0

#set ar2 to 0 => p=1
arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(1,1,1),seasonal=list(order=c(1,1,0),period = 12), method = 'ML')) #1084.443

#set sar1 to 0 => P=0
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(0,1,0), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(0,1,0),period = 12), method = 'ML')) #1086.053

# Model 4
#set ar1, sar1 to 0
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,NA,NA,0), method =
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,NA,NA,0),method
#1082.991
#ar2 contians 0

# Model 5 / Model D
#set ar2, sar1 to 0
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(NA,0,NA,0), method =
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(NA,0,NA,0),method
#1083.377
```

```
# Model 3
#set ar1, ar2, sar1 to 0
arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0), period = 12), fixed=c(0,0,NA,0), method = '
AICc(arima(vehicle, order=c(2,1,1),seasonal=list(order=c(1,1,0),period = 12), fixed=c(0,0,NA,0),method =
#1081.379

# Model 1 / Model A
#SARIMA(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12 (lowest AICc)
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),method = 'ML')
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),method = 'ML')) #1073.747
#sma1 contains 0

# Model 2 / Model C
#set sma1 = 0
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),fixed=c(NA,0, NA),method = 'ML
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),fixed=c(NA,0, NA), method
```

```
#start with pure MA
arima(vehicle, order=c(0,1,3),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(0,1,3),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) #1076.85

#model 12
arima(vehicle, order=c(0,1,2),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(0,1,2),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) # 1074.6

#model 8 / Model B
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')) #1072.44

#model10
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12), method = 'ML')) #1073.74

# Model 14
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), fixed = c(NA,NA,0,NA), method
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,3), period = 12), fixed = c(NA,NA,0,NA), m
# 1074.173

#model 13
arima(vehicle, order=c(0,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(0,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')) #1074.64

arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12), method = 'ML')
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(1,1,3), period = 12), method = 'ML')) #1082.66

arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(NA,NA,NA,0,NA,NA,NA,
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(NA,NA,NA,0,NA,N

arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(NA,0,NA,0,NA,NA,NA,N
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(NA,0,NA,0,NA,NA

#model11
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(0,0,NA,0,NA,NA,NA,NA
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(0,0,NA,0,NA,NA,
```

```r
#model 9
arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3),period = 12), fixed = c(0,0,0,0,NA,NA,NA,NA)
AICc(arima(vehicle, order=c(4,1,1),seasonal=list(order=c(0,1,3), period = 12),fixed = c(0,0,0,0,NA,NA,NA
```

```r
#check units roots for 14 models
library(UnitCircle)
#tentative model1: SAR(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12 (has unit root)
uc.check(pol_=c(1, -0.4936  -0.5059), plot_output = TRUE)

#tentative model2: SAR(p=0,d=1,q=1)×(P=0,D=1,Q=2)s=12 with sma1 = 0 (no unit root)
uc.check(pol_=c(1, -0.8183), plot_output = TRUE)
uc.check(pol_=c(1, 0, -0.2787), plot_output = TRUE)

#tentative model3: SAR(p=0,d=1,q=1)×(P=0,D=1,Q=0)s=12 (has unit root)
uc.check(pol_=c(1, -1.2070), plot_output = TRUE)

#tentative model4: SAR(p=2,d=1,q=1)×(P=0,D=1,Q=0)s=12 (has unit root)
uc.check(pol_=c(1, 0,  -0.0732), plot_output = TRUE)
uc.check(pol_=c(1,-1.1765, 0 ), plot_output = TRUE)

#tentative model5: SAR(p=1,d=1,q=1)×(P=0,D=1,Q=0)s=12 (no unit root)
uc.check(pol_=c(1, 0.0060,0), plot_output = TRUE)
uc.check(pol_=c(1, -0.826,0), plot_output = TRUE)

#tentative model6: #SAR(p=2,d=1,q=1)×(P=1,D=1,Q=0)s=12 with ar1 = 0 (has unit root)
uc.check(pol_=c(1,0, -0.0908), plot_output = TRUE)
uc.check(pol_=c(1, -1.1816), plot_output = TRUE)
uc.check(pol_=c(1, 0.1067), plot_output = TRUE)

# model 7  (no unit roots)
uc.check(pol_=c(1, 0.0326), plot_output = TRUE)
uc.check(pol_=c(1, -0.8070), plot_output = TRUE)
uc.check(pol_=c(1, -0.0991), plot_output = TRUE)

#model 8
uc.check(pol_=c(1, -0.3912, -0.3275, -0.2813), plot_output = TRUE) #(has unit root)

#model 9 (has unit root)

#model 10 (has unit root)
uc.check(pol_=c(1, -0.4936, -0.5059), plot_output = TRUE)

#model 11 (has unit root)
uc.check(pol_=c(1, 0,0,-0.1158,0), plot_output = TRUE)

#model12 (has u)
uc.check(pol_=c(1, -0.7907,0.0197), plot_output = TRUE)
uc.check(pol_=c(1, -0.3940, -0.3219, -0.2840), plot_output = TRUE)

#model 13
uc.check(pol_=c(1,0.0063,  -0.3964,  -0.3243,  -0.2792), plot_output = TRUE)

#model 14
```

```
uc.check(pol_=c(-0.322, 0, -0.4063), plot_output = TRUE)
```

```
#Diagnostic checking for Model C
ten_modelC <- arima(vehicle, order=c(0,1,1),seasonal=list(order=c(0,1,2), period = 12),fixed=c(NA,0, NA

#plot residuals
resC<-residuals(ten_modelC)
plot.ts(resC, main = 'Fitted residuals of tentative model C')
fit_resC <- lm(resC ~ as.numeric(1:length(resC)))
abline(fit_resC, col = 'red')
abline(h=mean(resC), col='blue')

#mean and var of residuals
m<-mean(resC)
std<-sqrt(var(resC))
m
std

#histogram for residuals
hist(resC, density = 20, breaks = 15, col = 'light blue', xlab='', prob=TRUE)
curve(dnorm(x, m, std), add=TRUE)

#Examine Normal QQ plot
qqnorm(resC, main = 'Normal Q-Q Plot for tentative model C')
qqline(resC, col='blue')

#Ran Shapiro-Wilk test of normality
shapiro.test(resC)

#ACF/PACF for fitted residuals
acf(resC, main = 'Autocorrelation')
pacf(resC, main = 'Partial Autocorrealation')

#test for independence of residuals
Box.test(resC, lag=10, type=c('Box-Pierce'), fitdf=2)
Box.test(resC, lag=10, type=c('Ljung-Box'), fitdf=2)
Box.test(resC^2, lag=10, type=c('Ljung-Box'), fitdf=0)

#Use Yule-Walker estimation: should fit into AR(0)
ar(resC, aic=TRUE, order.max = NULL, method=c('yule-walker'))
```

```
#Diagnostic Checking for Model D
ten_modelD <- arima(vehicle, order=c(1,1,1),seasonal=list(order=c(0,1,0), period = 12), method = 'ML')

#plot residuals
resD<-residuals(ten_modelD)
plot.ts(resD, main = 'Fitted residuals of tentative model D')
fit_resD <- lm(resD ~ as.numeric(1:length(resD)))
abline(fit_resD, col = 'red')
abline(h=mean(resD), col='blue')

#mean and var of residuals
mD<-mean(resD)
stdD<-sqrt(var(resD))
```

```
mD
stdD

#histogram for residuals
hist(resD, density = 20, breaks = 15, col = 'light blue', xlab='', prob=TRUE)
curve(dnorm(x, m, std), add=TRUE)

#Examine Normal QQ plot
qqnorm(resD, main = 'Normal Q-Q Plot for tentative model D')
qqline(resD, col='blue')

#Ran Shapiro-Wilk test of normality
shapiro.test(resD)

#ACF/PACF for fitted residuals
acf(resD, main = 'Autocorrelation')
pacf(resD, main = 'Partial Autocorrealation')

#test for independence of residuals
Box.test(resD, lag=10, type=c('Box-Pierce'), fitdf=2)
Box.test(resD, lag=10, type=c('Ljung-Box'), fitdf=2)
Box.test(resD^2, lag=10, type=c('Ljung-Box'), fitdf=0)

#Use Yule-Walker estimation: should fit into AR(0)
ar(resD, aic=TRUE, order.max = NULL, method=c('yule-walker'))
```

```
#forecasting using model D
library(forecast)
forecast <- forecast(ten_modelD)
pred.values <- forecast$mean[1:12]
cbind(vehicles_test, pred.values)

#plot predicted values with original series
pred.tr <- predict(ten_modelD, n.ahead=12)
u.tr <- pred.tr$pred + 2*pred.tr$se
l.tr <- pred.tr$pred - 2*pred.tr$se
ts.plot(vehicle, xlim = c(60, length(vehicle)+12), ylim = c(1000, max(u.tr)))
lines(u.tr, col = 'black', lty = 'dashed')
lines(l.tr, col = 'black', lty = 'dashed')
points(109:120, pred.tr$pred, col = 'red')
lines(109:120, pred.tr$pred, col = 'red')
points(109:120, vehicles_test[,2], col= "blue")
lines(109:120, vehicles_test[,2], col= "blue")
legend("topleft", pch = 1, col = c('red','blue'), legend = c('Forecasted Values', 'True Values'))

#zoom in in the forecast part
pred.tr <- predict(ten_modelD, n.ahead=12)
u.tr <- pred.tr$pred + 2*pred.tr$se
l.tr <- pred.tr$pred - 2*pred.tr$se
ts.plot(vehicle, xlim = c(100, length(vehicle)+12), ylim = c(1000, max(u.tr)))
lines(u.tr, col = 'black', lty = 'dashed')
lines(l.tr, col = 'black', lty = 'dashed')
points(109:120, pred.tr$pred, col = 'red')
lines(109:120, pred.tr$pred, col = 'red')
```

```
points(109:120, vehicles_test[,2], col= "blue")
lines(109:120, vehicles_test[,2], col= "blue")
legend("topleft", pch = 1, col = c('red','blue'), legend = c('Forecasted Values', 'True Values'))
```