Australian Beer Production Forecasting Model

By : Minh Pham

30 November 2021

Abstract :

The main objective regarding this project was to achieve an accurate forecast of quarterly beer production in Australia with the usage of a time series dataset. The dataset selected proves to be interesting because of the fact that currently 85% of Australia's Beer production gets sold in Australia and it generates around $16 billion a year in economic activity. With the use of various statistical methods and applications such as box-cox and log transformations, seasonal and nonseasonal differencing, and the analysis of ACF and PACF plots, one is able to identify various candidate models.

Through analysis of AIC, we were able to select two models that best fit the data: SARIMA(2,1,1)(0,1,2)s=4 model (A) and SARIMA(1,1,1)(0,1,2)s=4 model (B). They both passed diagnostic checking of roots for stationarity and invertibility. With the utilization of various tests such as Shaprio-Wilk test, Box-Pierce test, Ljung-Box test, and McLeod-Li test we were able to confirm independence, goodness of fit, and normality of their corresponding residuals. Both models passed these tests so they prove to both be good candidates for forecasting. However, we see that model B has a smaller p value (parsimony : AIC tends to overestimate p) so we select SARIMA(1,1,1)(0,1,2)s=4 to forecast the data 12 steps ahead and determine whether Australian beer production will increase or decrease in the future.

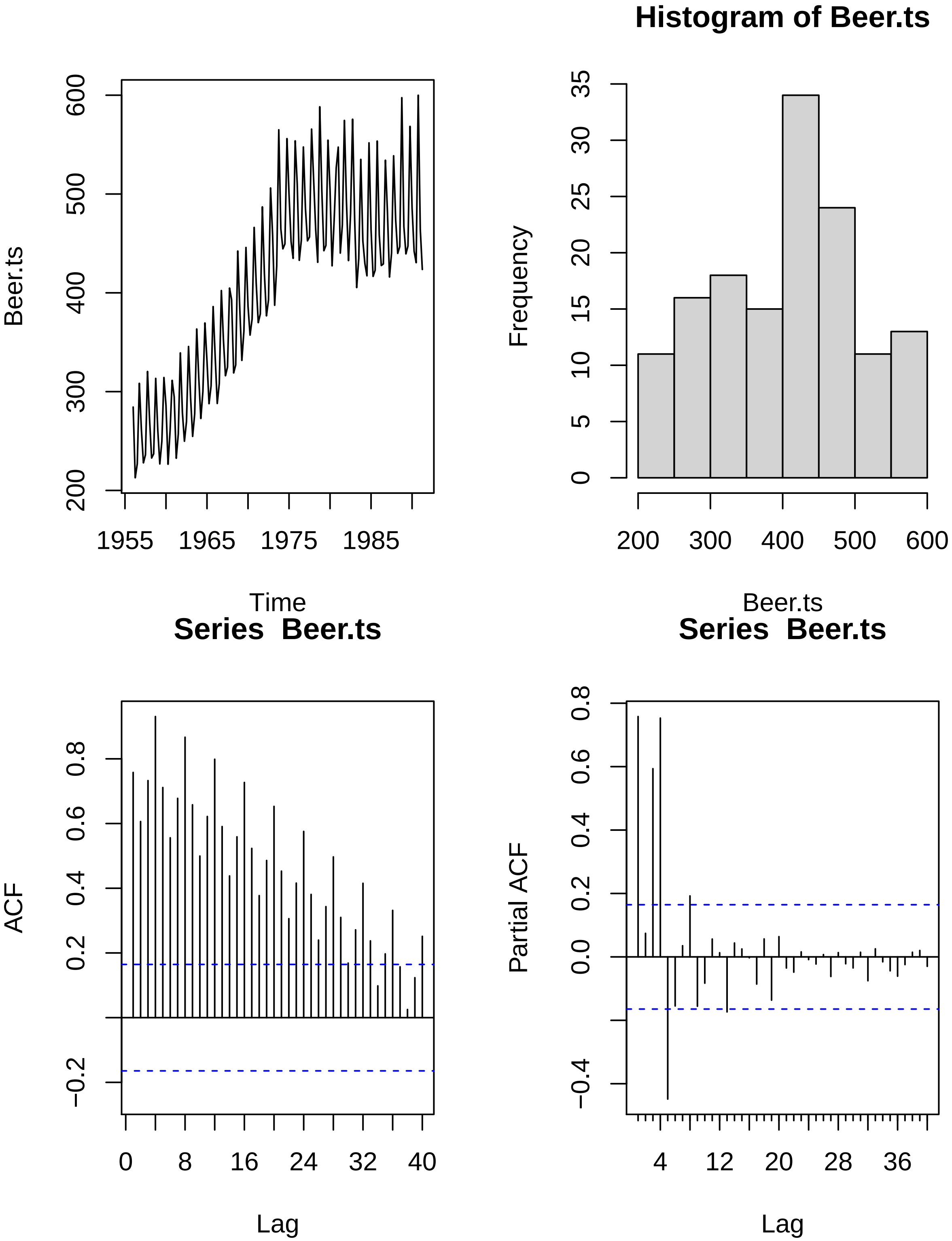
Introduction :

Australia's beer industry has a large impact on its own economy, it is estimated that $16 billion dollars of Australia's economy is rooted in the beer industry. According to ACIL Allen Consulting, Economic Contribution of the Australian Brewing Industry 2018-19 from Producers to Consumers, March 2020, Australian-made beers contributed around $254 million to the ingredients and agriculture industry, $582 million to the materials and packaging industry, $281 million to the transport and freight industry, $490 million to the marketing and sales industry and $198 million to the administration industry. It provides numerous amounts of jobs for Australia's community, for each full-time job in beer making, on average there are 21.6 other jobs created in the economy. Through forecasting this data, we can determine whether the beer industry will still be prevalent in their production.

This dataset was obtained through the tsdl library which originated from the Australian Bureau of Statistics. It consists of 154 different quarterly observations ranging from March of 1956 to June of 1994. The data is presented in megalitres which is equivalent to 1000000 liters per. We were able to use r studio to produce the forecast and r markdown to create this report. The first step in the time series project was the splitting of the data into a test and training set to validate the forecast of the final model. Upon first view of the model with the use of visualization techniques such as time series plotting, it is clear that the variance could be reduced so box-cox and log transformations were used, then the variance between each model was compared and the lowest was selected (the model with log transformation). Afterwards, decomposition was applied to visualize that there is an apparent trend and seasonal component within the time series dataset so differencing at lags 1 to remove trend and lag 4 to remove seasonality was required since it is a quarterly dataset. Then, through analysis of the PACF and ACF graphs of the differenced model, candidate models were identified and the best models (SARIMA(2,1,1)(0,1,2)s=4 and SARIMA(1,1,1)(0,1,2)s=4) were chosen by comparing AIC values and choosing the ones with the lowest. After finding the best candidate models, diagnostic testing of the roots and residuals were done with plots and various tests to validate the models. Even though both models passed diagnostic testing, with the concept of parsimony (choose models with less parameters) we were able to determine that model B (SARIMA(1,1,1)(0,1,2)s=4) was the best model out of the two models because it has a lower p value than model A (SARIMA(2,1,1)(0,1,2)s=4). Then, a forecast of the model to see 12 steps ahead was preformed. From the forecast graph, it is clear that the test data lies within the confidence intervals so we conclude that the forecast is valid. In the end, with the given forecast we were able to conclude that the Australia's beer industry will continue to maintain or increase production which in return will maintain a positive impact on Australia's economy.

Part 1 : Basic analysis

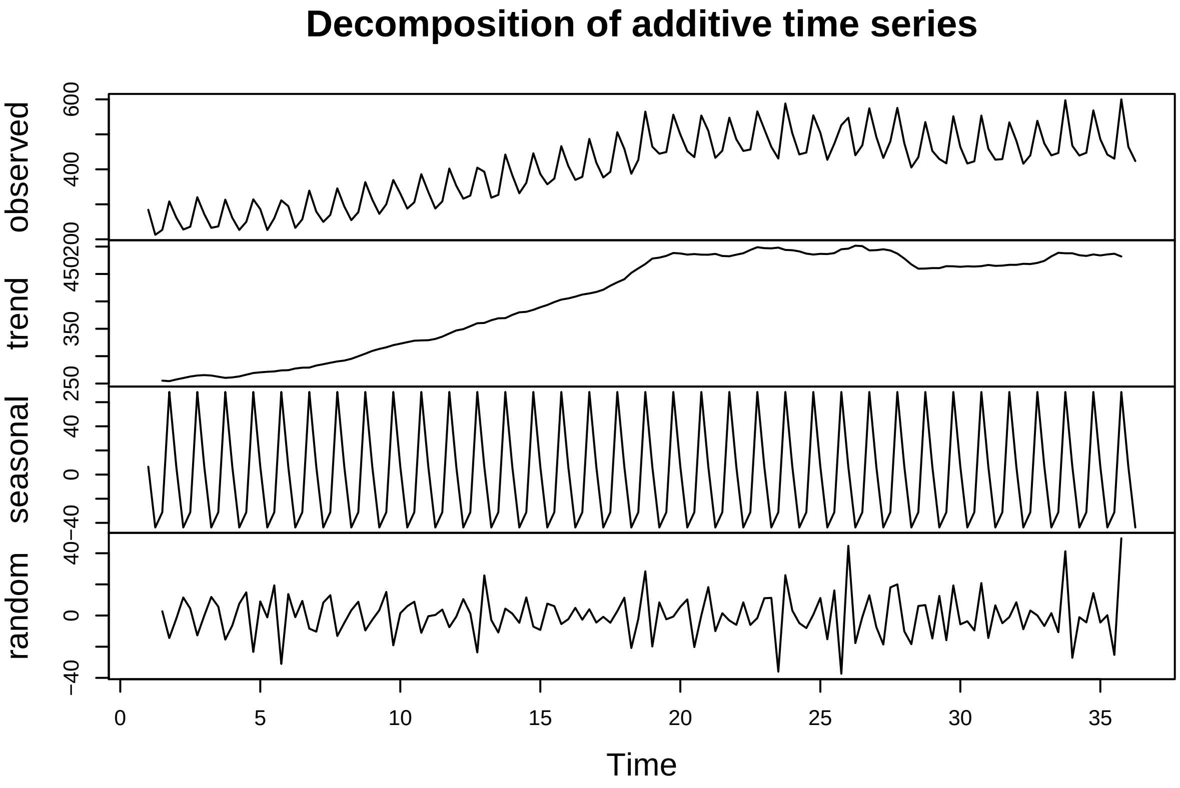
The first step in our project is to visualize the data to see if there are any apparent trends or seasonality and to determine if a transformation is needed.



Initially, from the time series plot we can see that some sort of transformation is needed due to it not looking stationary (there is non constant mean and variance over time). Furthermore, from the histogram plot we see that it is slightly skewed and the ACF's remain large and periodic.

Decomposition and checking trend

We can use the decompose function to split the time series data and visualize its various componenets such as, trend, seasonality, and stationarity.

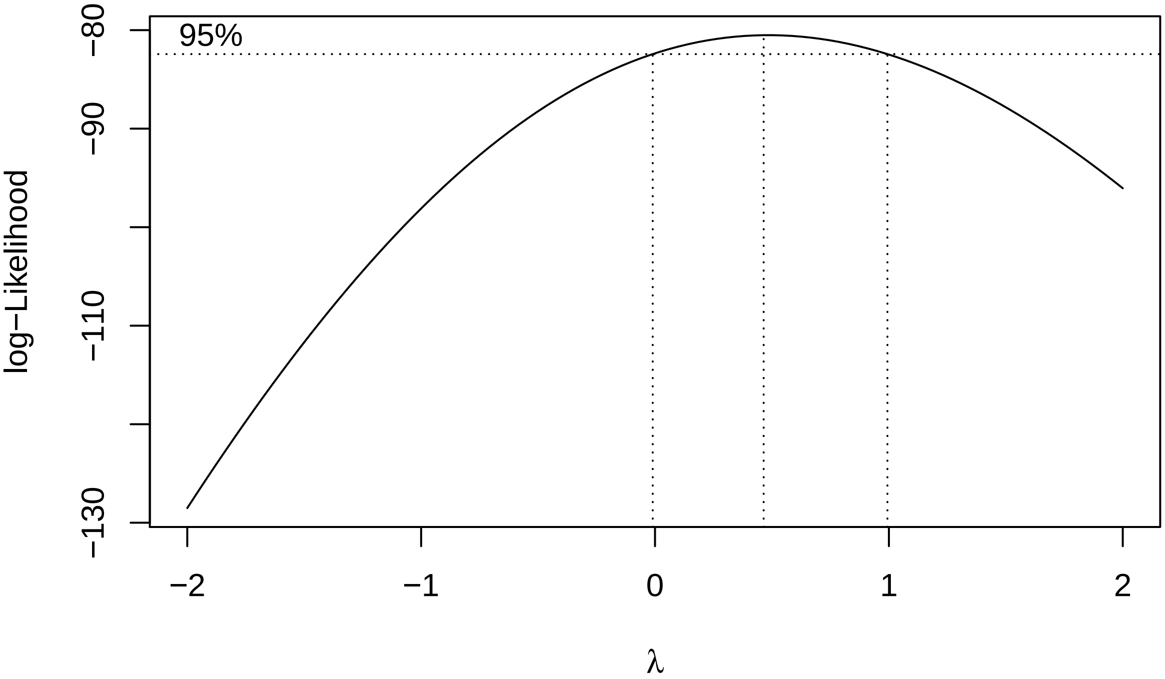


From the decomposition, we can clearly see that there is a trend and seasonal component as well. This would require us to difference the model at lag 4 since it is a quarterly seasonal component and lag 1 since there is a trend.

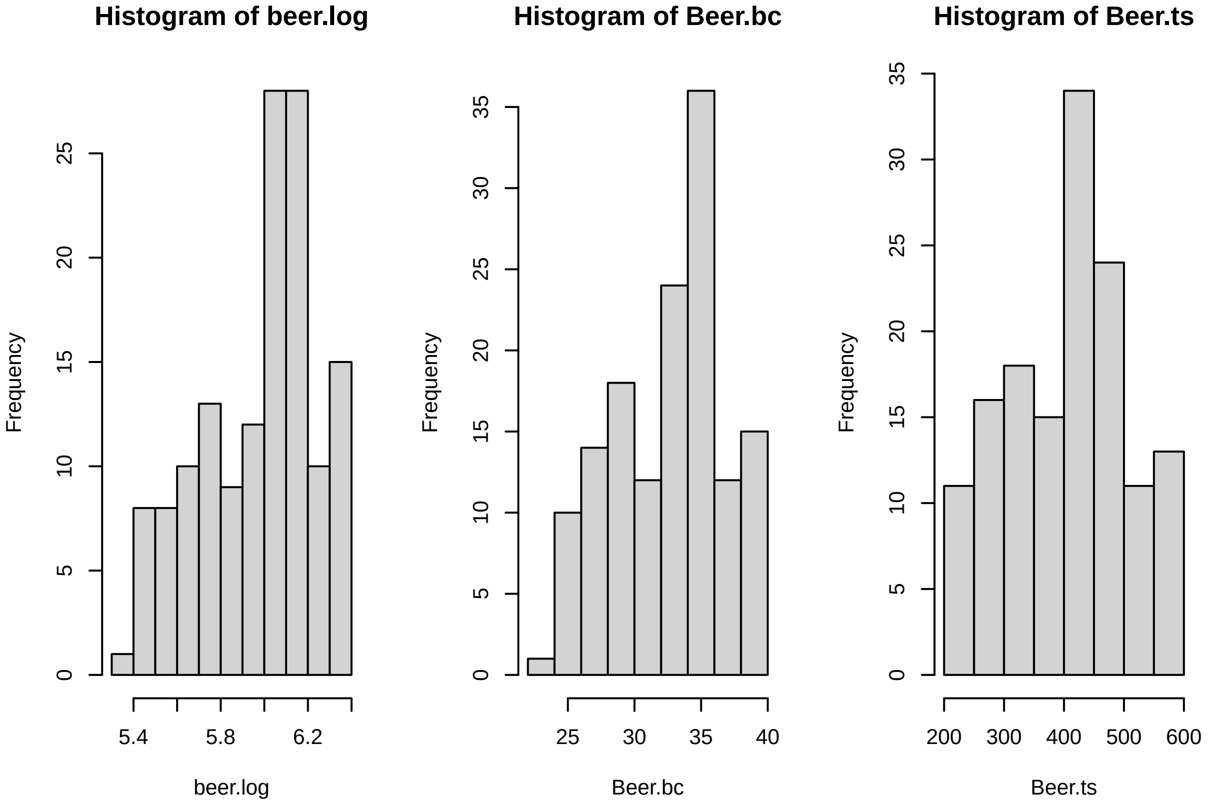
Part 2 : Transformations

From the inital histogram of the time series data, we can see that it requires some sort of variance stabalization so we chose to do some transformations :

We first check the value of lambda for the given model :



From this graph we can see that lambda's confidence interval contains 0 so we could also utilize log transformation as well as box-cox



```
## [1] "beer.log var : 0.0694043170164377"
```

```
## [1] "beer.bc var : 16.9587879155819"
```

```
## [1] "beer.ts var : 9905.72320447508"
```

From the three histograms, we can see that they are all slightly skewed so instead we compare the variance of each one. From the variance we can see that the log transformed time series has a significantly lower variance than the other two, so we choose to use the log transformed model in our project.

Part 3 : Differencing

From the decomposition in part 1, we can see that the time series needs to be differenced at lags 4 and 1 to remove its seasonality and trend components.

```
## [1] "beer differenced at lag 4 once var : 0.00207447600070139"
```

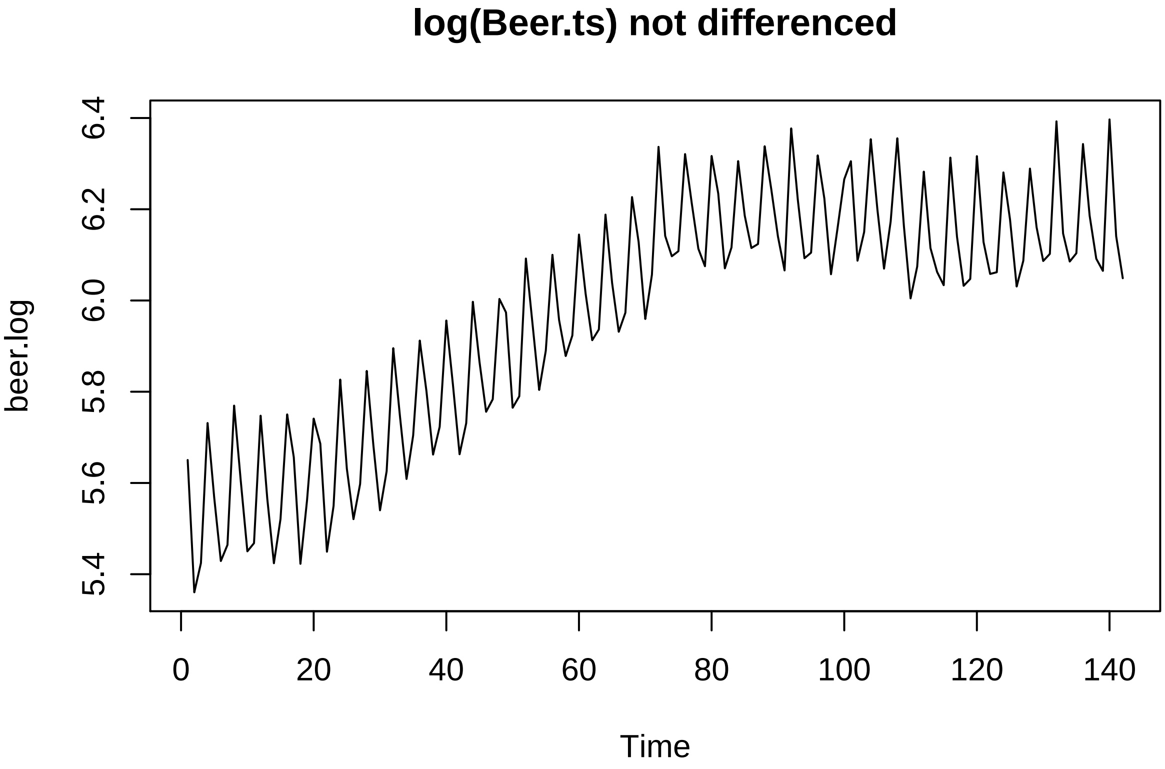
```
## [1] "beer differenced at lag 4 twice var : 0.00438150672975723"
```

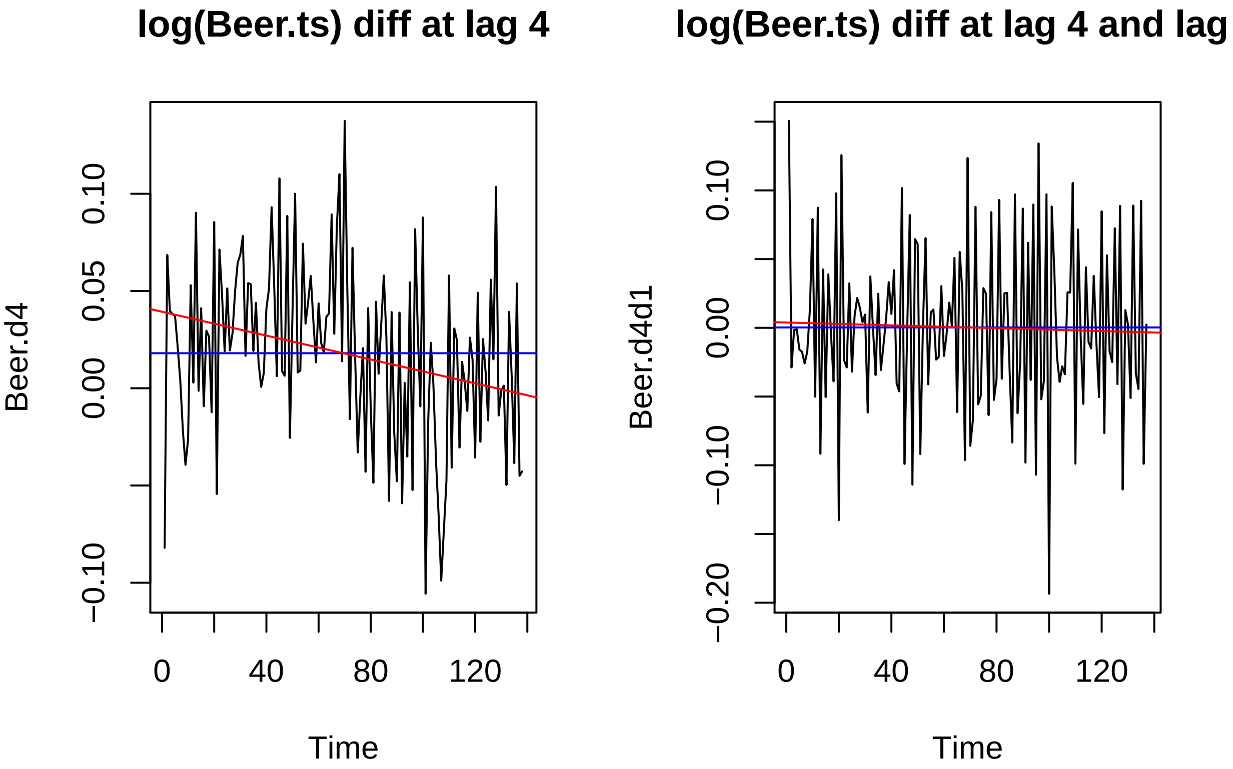
Here we choose to only difference once at lag 4 since the data is quarterly and also when we difference it twice, the variance increases.

```
## [1] "beer differenced at lag 4 once and lag 1 once : 0.00393706585041396"
```

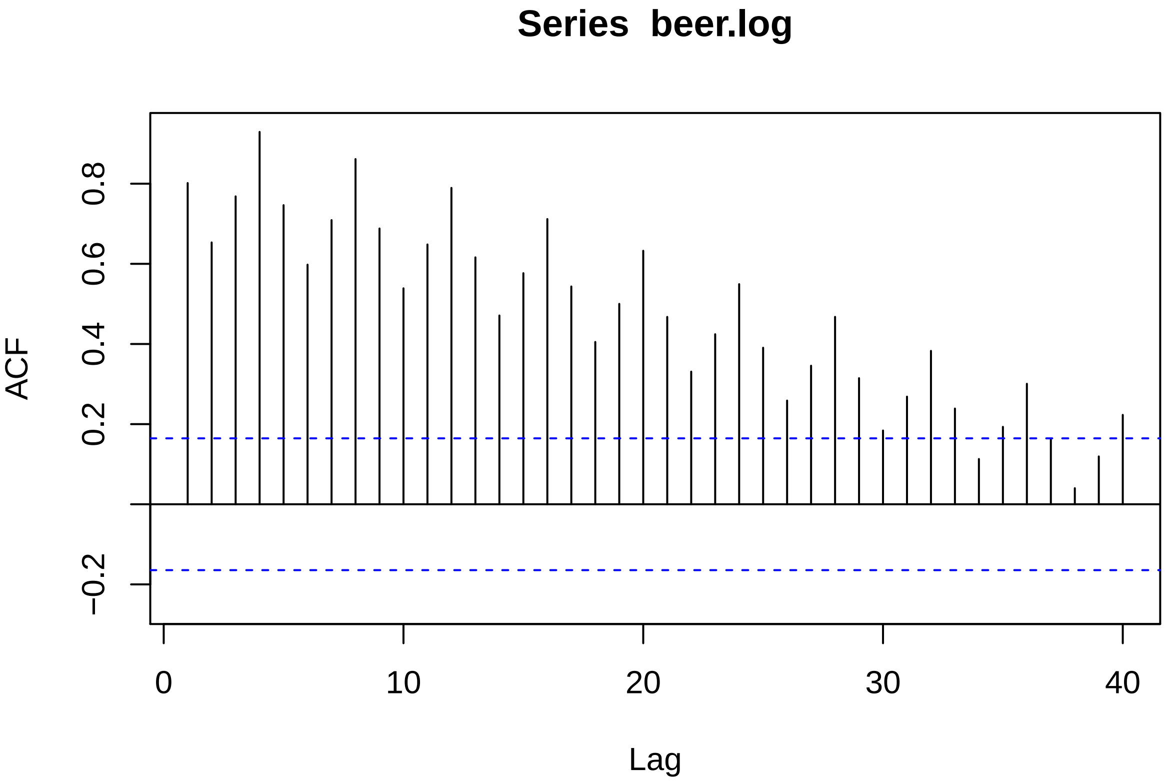
```
## [1] "beer differenced at lag 4 once and lag 1 twice : 0.0125819323747731"
```

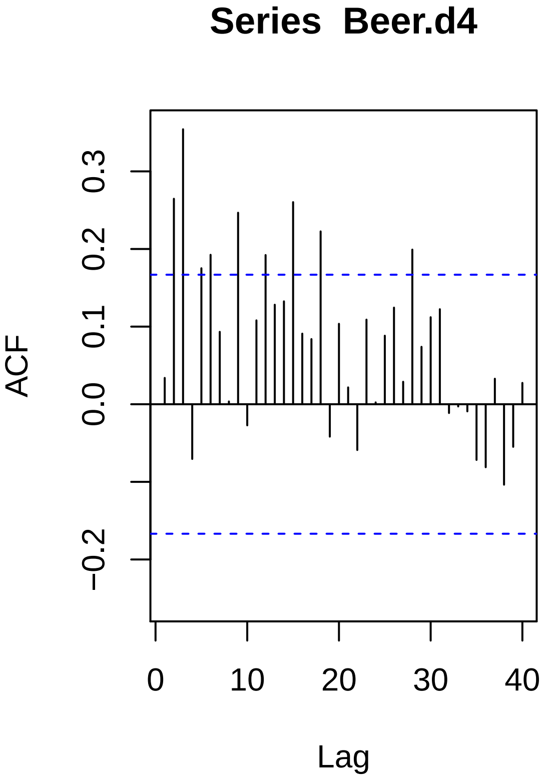
Similarly, we choose to only difference at lag 1 once because when we difference it twice, the variance increases. Eventhough, the variance increases when we differenced at lag 1 once, when we look at the time series plots and ACF graphs, we see that there is still a trend component when we only difference it at lag 4. Therefore, we still difference at lag 1 since it makes the model more stationary with only a small increase in variance.

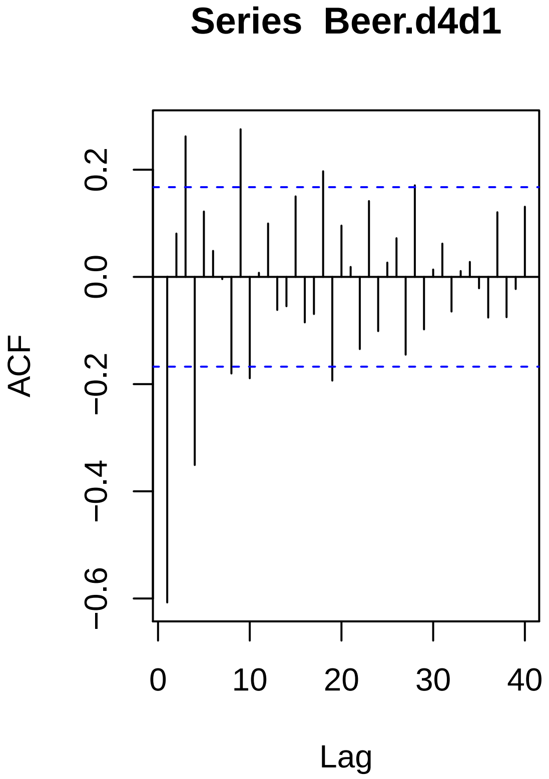




From the time series plots, we can see that seasonality component was removed when we differenced at lag 4 however, there was still a trend present. When we further differenced at lag 1, we see that the trend component has been removed. The data looks stationary, but we also need to check the ACF graphs as well.





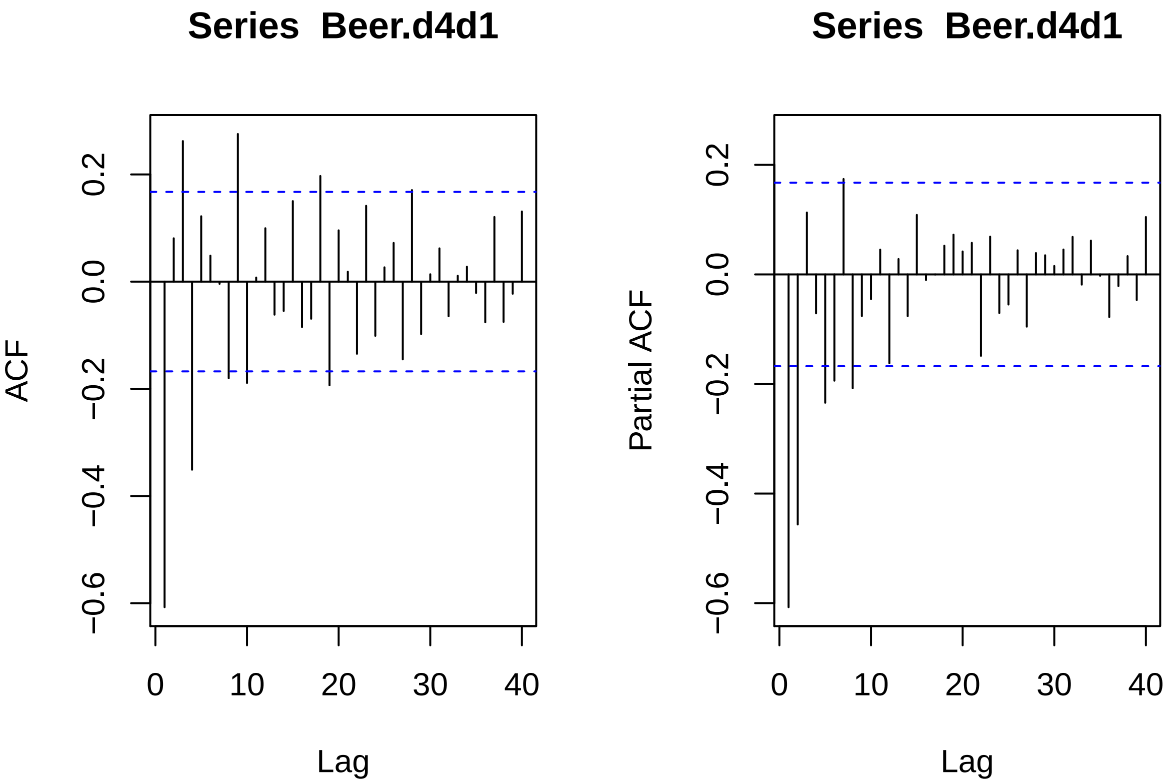


From the ACF graph of the undifferenced model we can clearly see that there is a slow decay with seasonality which indicates that it is nonstationarity.

Now, when looking at the second graph of the model differenced at lag 4, we can see that there is no more apparent seasonal component but however, there still seems to be a slow decay which indicates nonstationarity.

Finally, when viewing the last graph, which is differenced at lags 4 and 1, we see that the ACF decay corresponds more to a stationary process. Therefore, we choose to work with the model that was differenced at lags 4 and 1.

Part 4 : ACF and PACF of differenced model :



From the ACF graph we can see significant lags at 3, 4, 8, 9, 10, 18, and 19.

From the PACF graph we can see significant lags at 1, 2, 5, 6, 7, and 8

Here some possible candidate models are :

s = 4,

D = 1,

d = 1,

Q = 2,

P = 0,

q = 1, 3, 9, 10, 18, or 19

p = 1,2,5,6, 7 or 8

Part 5 : Trying different models :

```
##
## Call:
## arima(x = beer.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ma1     sma1     sma2
##       -0.3975  -0.5443  -0.6640  -0.0808
## s.e.   0.1017   0.0872   0.1018   0.1013
##
## sigma^2 estimated as 0.001336:  log likelihood = 256.94,  aic = -503.88
```

```
##
## Call:
## arima(x = beer.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ma1     sma1     sma2
##       -0.6835  -0.2851  -0.2875  -0.6359  -0.1211
## s.e.   0.1967   0.1580   0.1979   0.1018   0.1033
##
## sigma^2 estimated as 0.001304:  log likelihood = 258.46,  aic = -504.93
```

```
##
## Call:
## arima(x = beer.log, order = c(5, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2     ar3      ar4      ar5      ma1     sma1     sma2
##       -0.5378  -0.1453  0.0543  -0.2742  -0.1913  -0.4283  -0.3387  -0.3306
## s.e.   0.2904   0.2711  0.1724   0.3021   0.1675   0.2852   0.3021   0.2153
##
## sigma^2 estimated as 0.001296:  log likelihood = 258.93,  aic = -499.87
```

```
##
## Call:
## arima(x = beer.log, order = c(6, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6     ma1     sma1
##       -1.7849  -1.3389  -0.6085  -0.3940  -0.4444  -0.2545  0.8378  -0.4201
## s.e.   0.1421   0.2102   0.2217   0.2402   0.2770   0.1309  0.1240   0.1891
##          sma2
##       -0.1947
## s.e.   0.1419
##
## sigma^2 estimated as 0.001258:  log likelihood = 260.79,  aic = -501.59
```

```
##
```

```
## Call:
## arima(x = beer.log, order = c(7, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6      ar7      ma1
##       -0.8929  -0.4803  -0.1564  -0.9256  -0.8516  -0.4917  -0.1144  -0.0523
## s.e.   0.7344   0.6641   0.3252   0.0770   0.6767   0.6303   0.3288   0.7396
##         sma1     sma2
##       0.2491  -0.7506
## s.e.  0.0905   0.0838
##
## sigma^2 estimated as 0.001234:  log likelihood = 260.18,  aic = -498.35

##
## Call:
## arima(x = beer.log, order = c(8, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3     ar4     ar5     ar6     ar7      ar8
##       -1.4529  -1.0266  -0.4490  0.0005  0.1596  0.1483  0.0645  -0.0961
## s.e.   0.4196   0.4310   0.2928  0.3798  0.4654  0.3380  0.2090   0.1309
##          ma1    sma1    sma2
##       0.5153  -0.797  0.0704
## s.e.  0.4156   0.319  0.2456
##
## sigma^2 estimated as 0.001249:  log likelihood = 261.32,  aic = -498.64

##
## Call:
## arima(x = beer.log, order = c(1, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ma1      ma2     ma3     sma1     sma2
##       -0.4423  -0.5282  -0.0422  0.1692  -0.7527  -0.0277
## s.e.  17.6070  17.6349  17.1720  6.9240   0.1322   0.1169
##
## sigma^2 estimated as 0.001306:  log likelihood = 258.34,  aic = -502.69

##
## Call:
## arima(x = beer.log, order = c(2, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ma1     ma2      ma3     sma1     sma2
##       -0.8675  -0.7626  -0.0831  0.2991  -0.4411  -0.4446  -0.2396
## s.e.   0.1389   0.1265   0.1627  0.1463   0.1493   0.1028   0.0943
##
## sigma^2 estimated as 0.001284:  log likelihood = 259.56,  aic = -503.12

## Warning in arima(beer.log, order = c(5, 1, 3), seasonal = list(order = c(0, :
```

```
## possible convergence problem: optim gave code = 1

##
## Call:
## arima(x = beer.log, order = c(5, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1     ar2      ar3      ar4      ar5      ma1      ma2     ma3
##       -0.7723  0.0607  -0.0462  -0.9618  -0.7361  -0.1541  -0.4500  0.3702
## s.e.  15.2456  0.8098   0.2003   1.4052  13.9018  16.5330  16.8339  6.6320
##         sma1     sma2
##       0.2247  -0.7752
## s.e.  1.8784   1.8740
##
## sigma^2 estimated as 0.001236:  log likelihood = 260.41,  aic = -498.82

## Warning in arima(beer.log, order = c(6, 1, 3), seasonal = list(order = c(0, :
## possible convergence problem: optim gave code = 1

##
## Call:
## arima(x = beer.log, order = c(6, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3      ar4      ar5      ar6     ma1      ma2
##       -0.9934  -0.0704  -0.0447  -0.9589  -0.9407  -0.1302  0.0603  -0.5145
## s.e.   0.2640   0.2675   0.0437   0.0516   0.2666   0.2474  0.2597   0.1130
##          ma3    sma1     sma2
##       0.3194  0.2529  -0.7470
## s.e.  0.1955  0.0951   0.0918
##
## sigma^2 estimated as 0.001224:  log likelihood = 260.8,  aic = -497.59

##
## Call:
## arima(x = beer.log, order = c(7, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##           ar1     ar2     ar3     ar4     ar5     ar6     ar7     ma1      ma2
##       -1.1854  0.0572  0.8728  0.6275  0.1945  0.1372  0.1795  0.2347  -0.8496
## s.e.   0.5948  0.5476  0.2903  0.6390  0.4400  0.1456  0.1692  0.5966   0.1034
##           ma3     sma1     sma2
##       -0.3851  -0.5458  -0.0980
## s.e.   0.5548   0.2331   0.1552
##
## sigma^2 estimated as 0.001243:  log likelihood = 261.21,  aic = -496.41

##
## Call:
## arima(x = beer.log, order = c(8, 1, 3), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1     ar2     ar3      ar4      ar5     ar6      ar7      ar8
##      -0.1190  0.7105  0.0189  -0.4680  -0.3100  0.0236  -0.0480  -0.3144
## s.e.   0.1268  0.1132  0.1101   0.1834   0.1074  0.1798   0.0904   0.1190
##          ma1      ma2     ma3     sma1     sma2
##      -0.9317  -0.4622  0.8299  -0.5044  -0.0319
## s.e.   0.1179   0.1920  0.1163   0.1811   0.1354
##
## sigma^2 estimated as 0.001133:  log likelihood = 265.71,  aic = -503.42
```

Removing terms with 0 in their confidence intervals :

```
##
## Call:
## arima(x = beer.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1      ar2      ma1     sma1     sma2
##      -0.6835  -0.2851  -0.2875  -0.6359  -0.1211
## s.e.   0.1967   0.1580   0.1979   0.1018   0.1033
##
## sigma^2 estimated as 0.001304:  log likelihood = 258.46,  aic = -504.93
```

```
## Warning in arima(beer.log, order = c(2, 1, 1), seasonal = list(order = c(0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
##
## Call:
## arima(x = beer.log, order = c(2, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), fixed = c(NA, 0, 0, NA, 0), method = "ML")
##
## Coefficients:
##          ar1  ar2  ma1     sma1  sma2
##      -0.6486    0    0  -0.7626     0
## s.e.   0.0680    0    0   0.0626     0
##
## sigma^2 estimated as 0.001583:  log likelihood = 245.43,  aic = -484.86
```

```
##
## Call:
## arima(x = beer.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), method = "ML")
##
## Coefficients:
##          ar1      ma1     sma1     sma2
##      -0.3975  -0.5443  -0.6640  -0.0808
## s.e.   0.1017   0.0872   0.1018   0.1013
##
## sigma^2 estimated as 0.001336:  log likelihood = 256.94,  aic = -503.88
```

```
##
## Call:
## arima(x = beer.log, order = c(1, 1, 1), seasonal = list(order = c(0, 1, 2),
##     period = 4), fixed = c(NA, NA, NA, 0), method = "ML")
##
## Coefficients:
##           ar1      ma1     sma1  sma2
##       -0.4085  -0.5325  -0.7249     0
## s.e.   0.0997   0.0853   0.0707     0
##
## sigma^2 estimated as 0.001343:  log likelihood = 256.62,  aic = -505.24
```

We would choose models SARIMA(2,1,1)(0,1,2)s=4 (Model A) and SARIMA(1,1,1)(0,1,2)s=4 (Model B) since they have the lowest AIC values.

Now from model A, SARIMA(2,1,1)(0,1,2)s=4, we see that 0 lies within the confidence interval of ar2, sma1, and ma1. However, when they are set to 0, the AIC increases so the unfixed model is used instead.

Now looking at model B, SARIMA(1,1,1)(0,1,2)s=4, we see that 0 lies within the confidence interval of sma2. When setting sma2 to 0, there is a decrease in the AIC so we use the model with sma2 being fixed to 0.

Models Chosen :

A : Log SARIMA(2,1,1)(0,1,2)s=4

$$X_t(1+0.6835B+0.2851B^2)(1-B)(1-B^4) = Z_t(1-0.285B)(1-0.6359B^4-0.1211B^8)$$
$$\nabla_1\nabla_4 ln(U_t)(1+0.6835B+0.2851B^2) = Z_t(1-0.285B)(1-0.6359B^4-0.1211B^8)$$

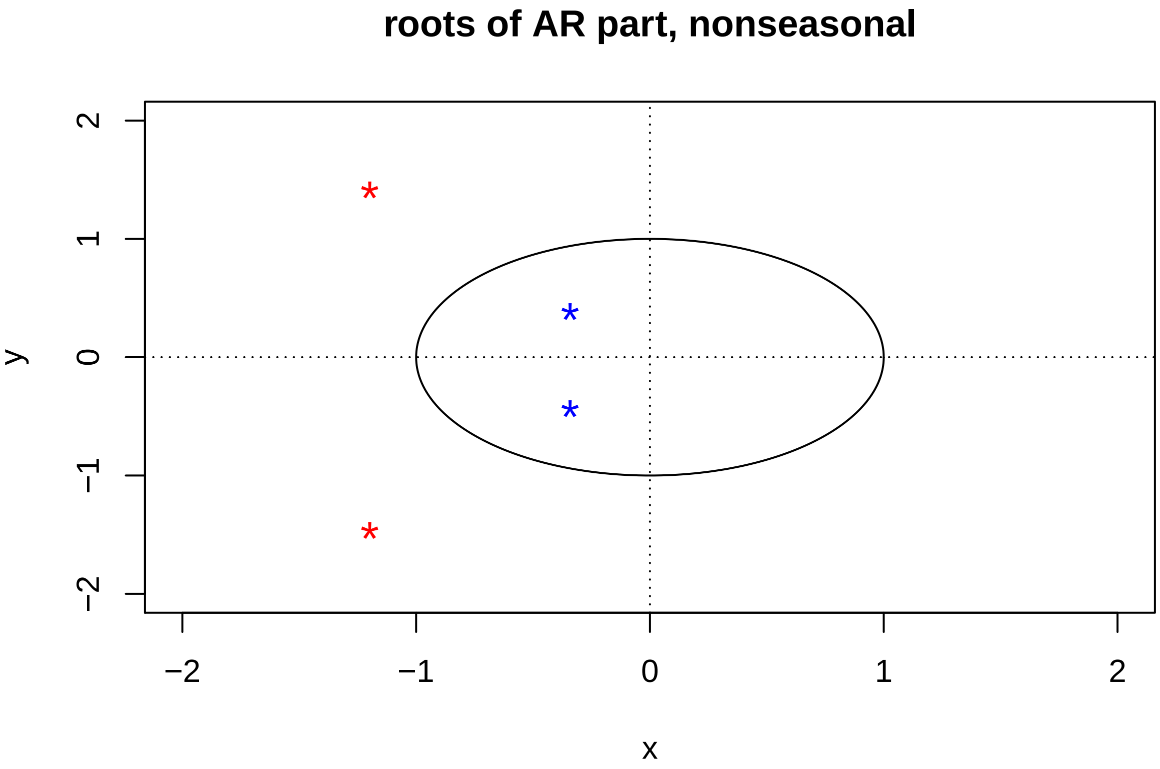
B : Log SARIMA(1,1,1)(0,1,2)s=4 w/ sma2 being fixed to 0

$$X_t(1+0.4085B)(1-B)(1-B^4) = Z_t(1-0.5325B)(1-0.7249B^4)$$
$$\nabla_1\nabla_4 ln(U_t)(1+0.4085B) = Z_t(1-0.5325B)(1-0.7249B^4)$$

Part 6 : Checking stationarity + invertibility

Check for roots Model A : SARIMA(2,1,1)(0,1,2)s=4 :

$$\nabla_1 \nabla_4 ln(U_t)(1 + 0.6835B + 0.2851B^2) = Z_t(1 - 0.285B)(1 - 0.6359B^4 - 0.1211B^8)$$



```
## [1] "root of MA part, nonseasonal : 3.47826086956522+0i"
```

```
## [1] "root of MA part, seasonal : 1.26690899250123+0i"
## [2] "root of MA part, seasonal : -6.51794119729066-0i"
```

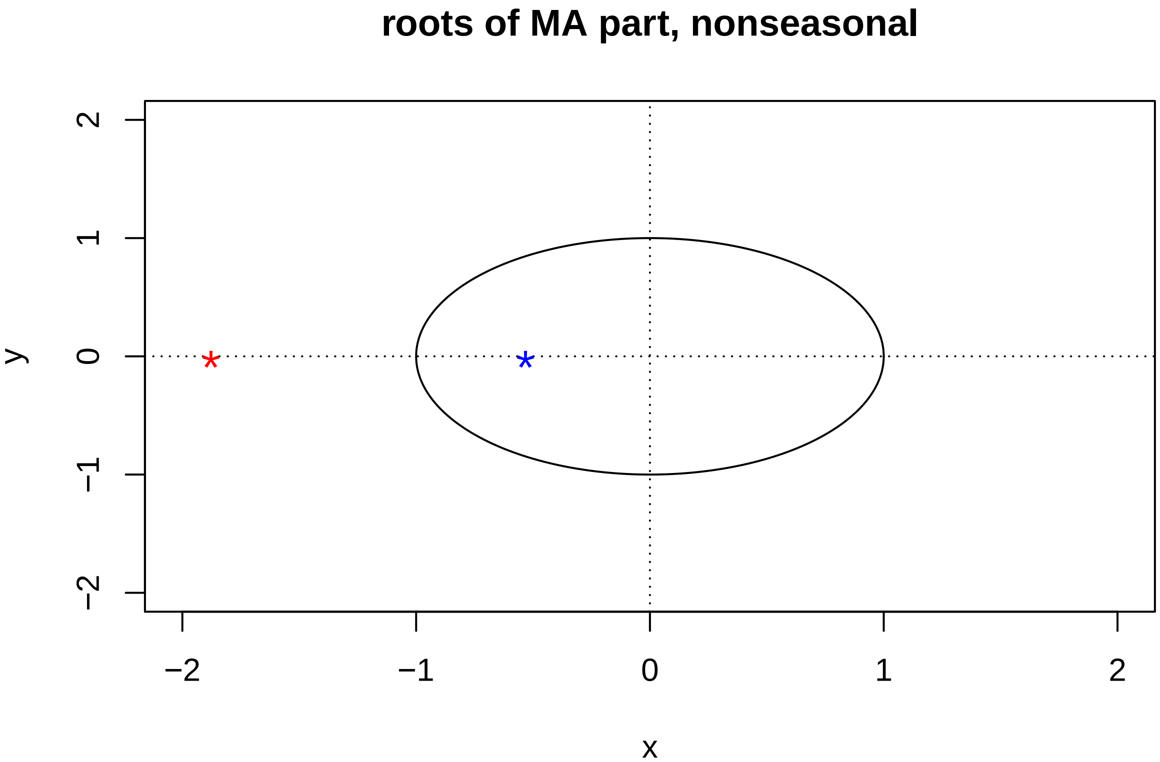
From the plot we can see that the roots of the nonseasonal autoregressive component of model A is outside of the unit circle which means that it is stationary. Furthermore, from the polyroot function we see that the roots of the seasonal and nonseasonal moving average components of this model is also outside of the unit circle (|roots| > 1) meaning that it is also invertible.

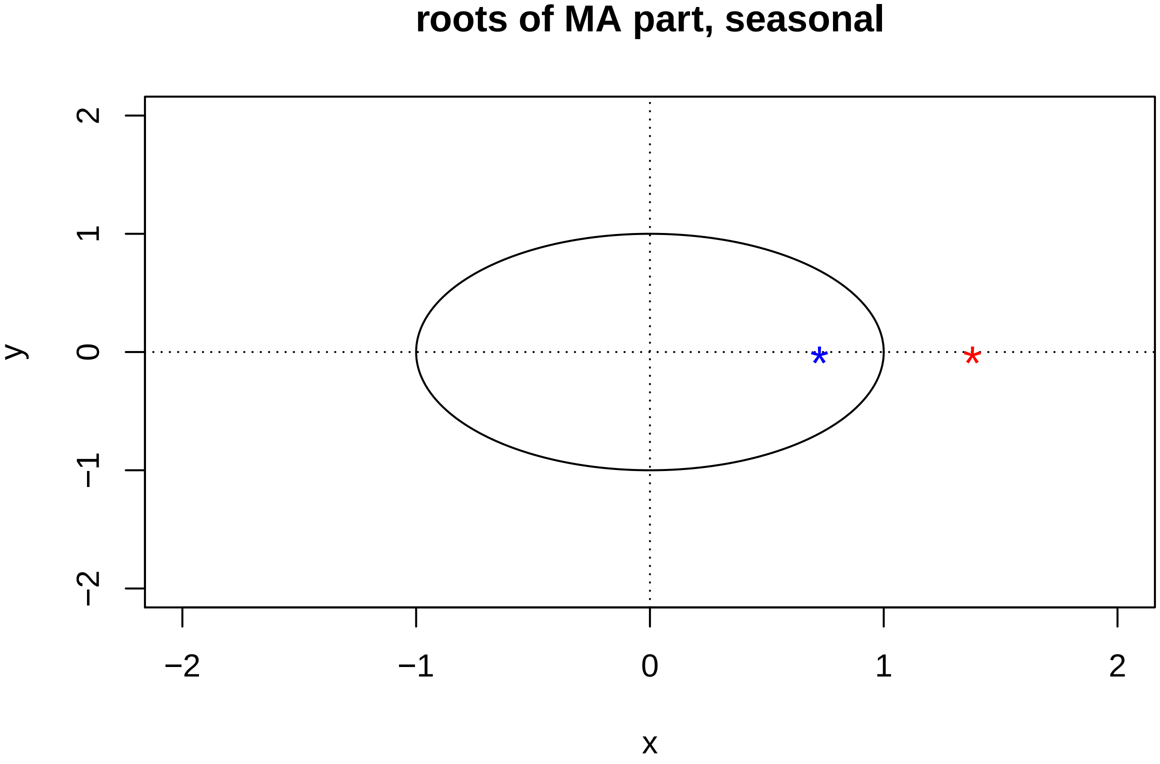
Therefore, we conclude that this model is both stationary and invertible and passed this portion of diagnostic testing.

Check for roots Model B : SARIMA(1,1,1)(0,1,2)s=4 :

$$\nabla_1\nabla_4 ln(U_t)(1+0.4085B) = Z_t(1-0.5325B)(1-0.7249B^4)$$

```
## [1] "root of AR part, nonseasonal : -2.44798041615667+0i"
```



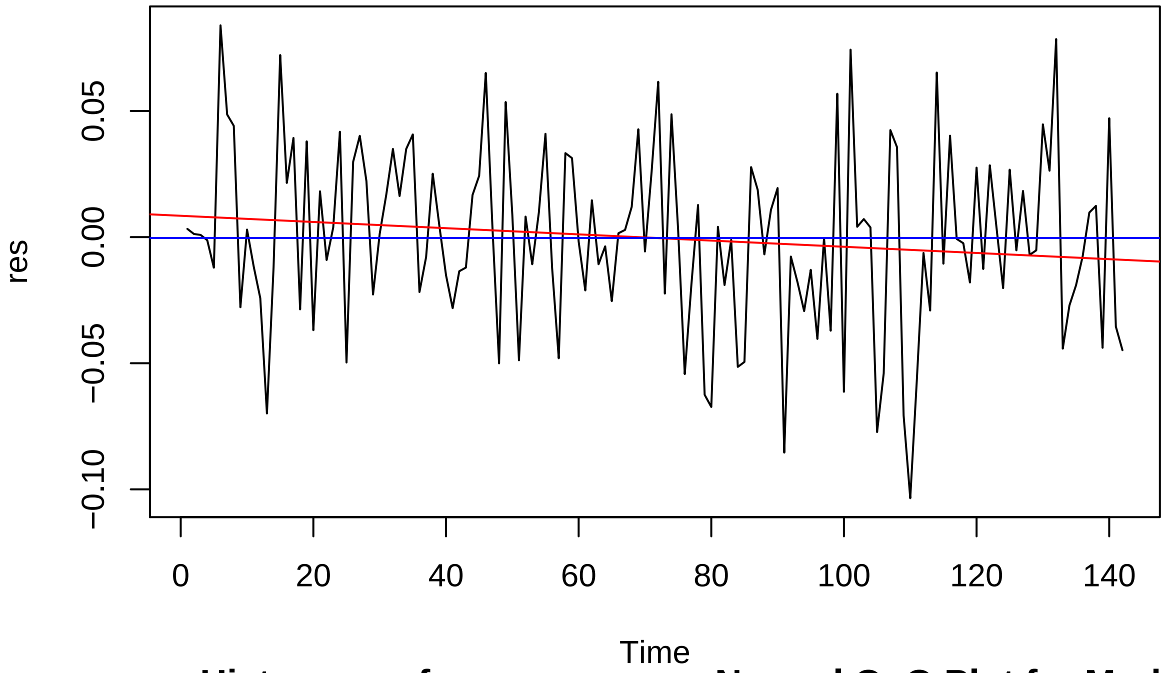


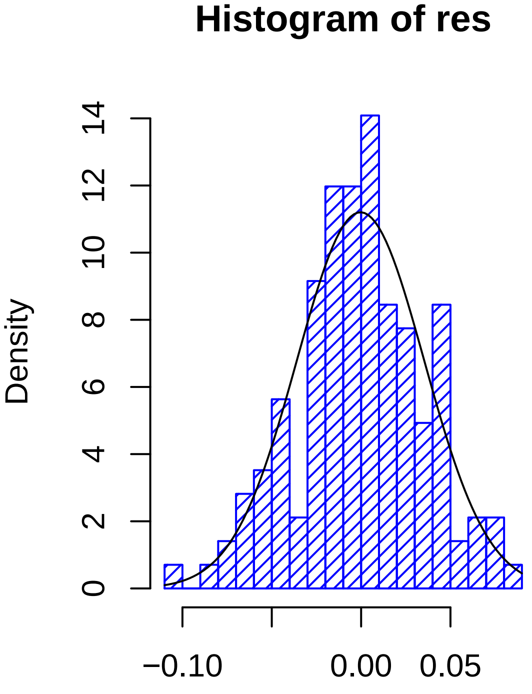
From the polyroot function we see that the roots of the nonseasonal AR component of model B lie outside of the unit circle ($|\text{root}| > 1$) which means that we can conclude that this model is stationary. Furthermore, we can also conclude that it is invertible because we see that the roots from the seasonal and nonseasonal MA components are outside the unit circle.

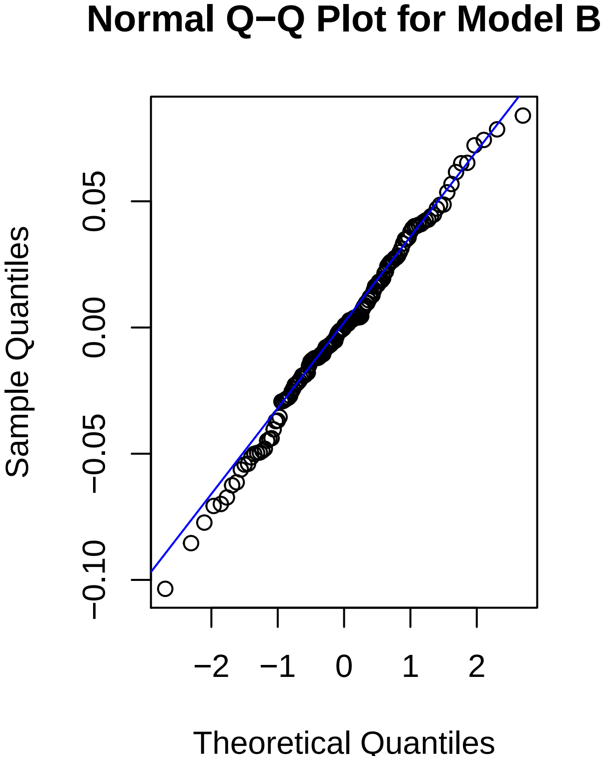
Therefore, we conclude that this model is stationary and invertible.

Part 7 : Diagnostic testing

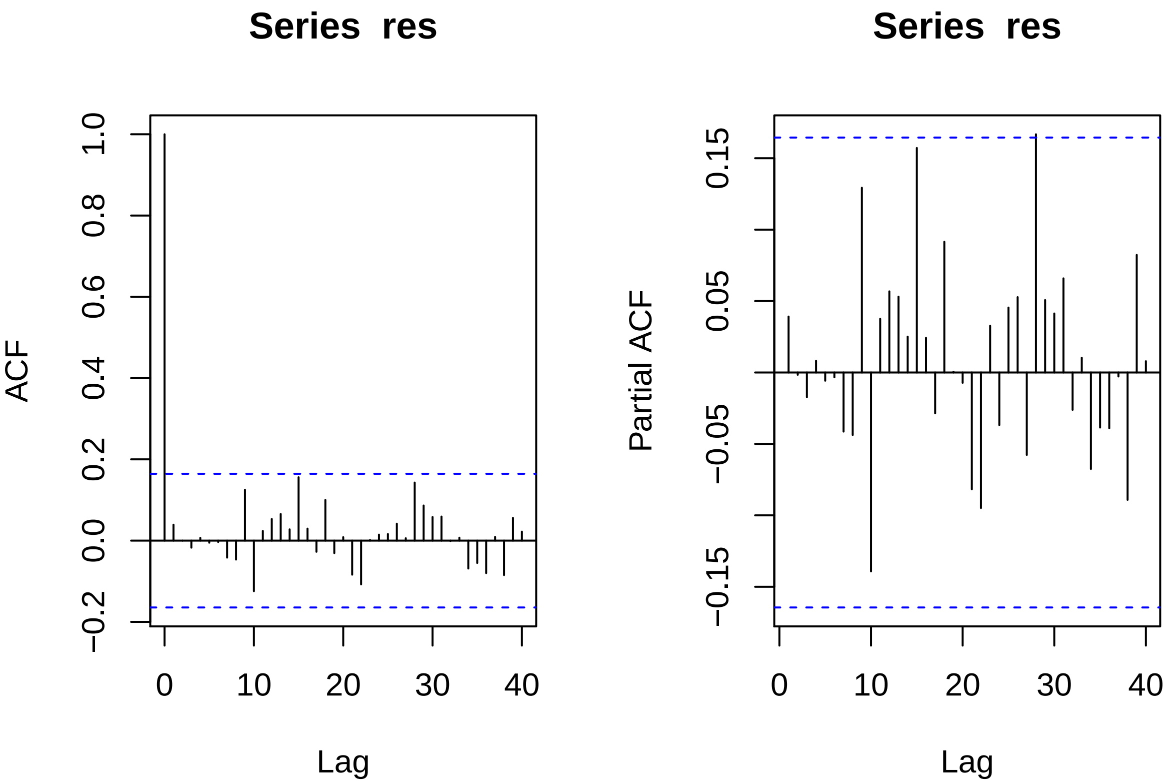
Diagnostic testing: Model A : SARIMA(2,1,1)(0,1,2)s=4







From the time series plot, we can see close to no change in variance, no seasonality, no trend, and the sample mean lies very close to 0 so it resembles white noise. From the histogram and normal QQ plot, we see that the residuals also closely follow a normal distribution.



When looking at the ACF and PACF graphs of the residuals, one can see that there are no significant lags and that all of them lie within the confidence interval so they could be counted as 0. This confirms that the residuals resemble a white noise distribution.

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.99394, p-value = 0.8149
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 5.7384, df = 10, p-value = 0.8367
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 6.2201, df = 10, p-value = 0.7964
```

```
##
##  Box-Ljung test
##
## data:  (res)^2
## X-squared = 13.511, df = 12, p-value = 0.333
```

For the Shapiro-Wilk test, this model obtained a p-value of 0.8149 which is above our 0.05 p-value threshold. Therefore, we don't reject the null hypothesis of the shapiro-wilk test and conclude that the residuals of the model follow a normal distribution.

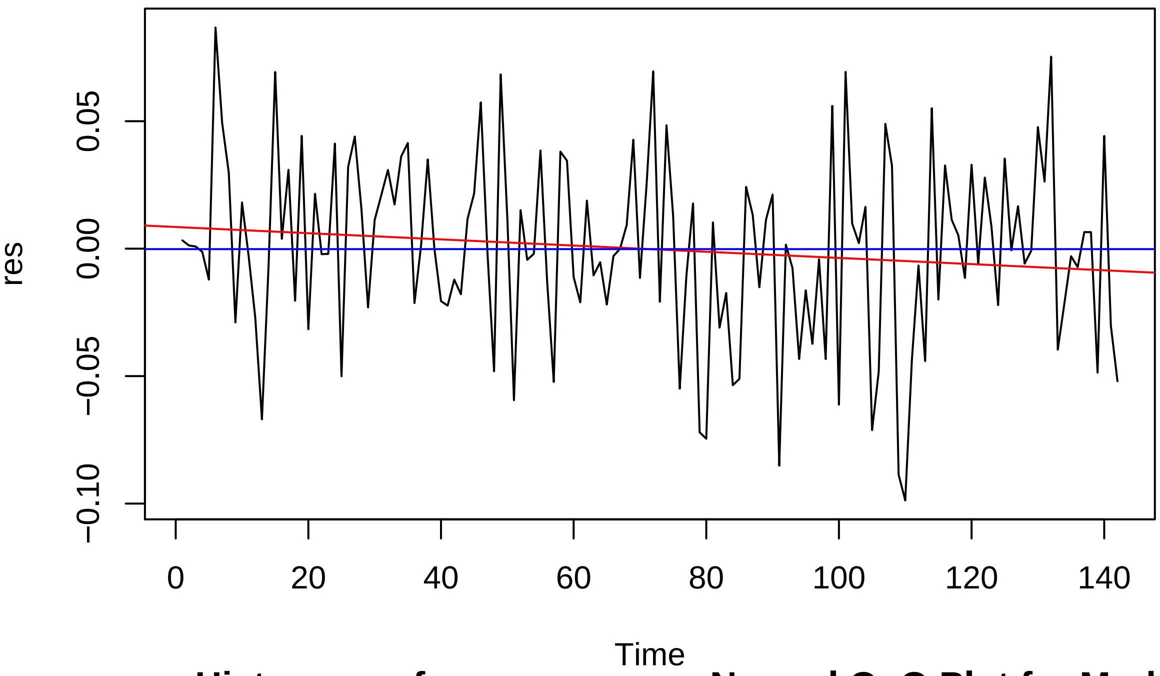
For the other 3 portmanteau tests we see similar results so we would not reject the null hypothesis for any of them. We conclude that the residuals does not show non-linear dependence and follow a gaussian WN(0,1).

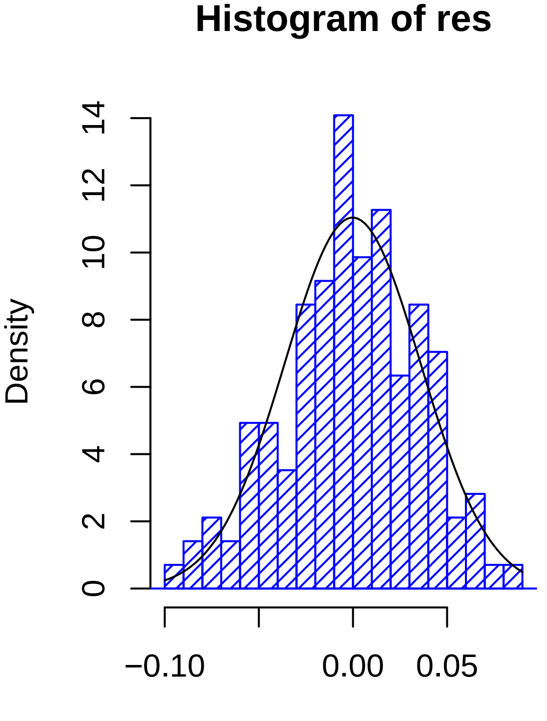
```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.001268
```

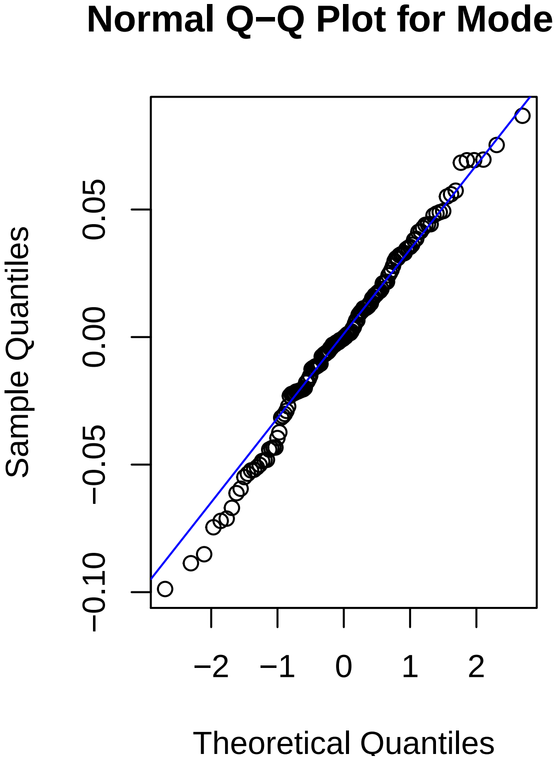
Additionally, r automatically selected order 0 for the residuals so we can conclude that it is in fact an AR(0) model which is white noise.

This model passed every diagnostic test so it is a prime candidate for forecasting.

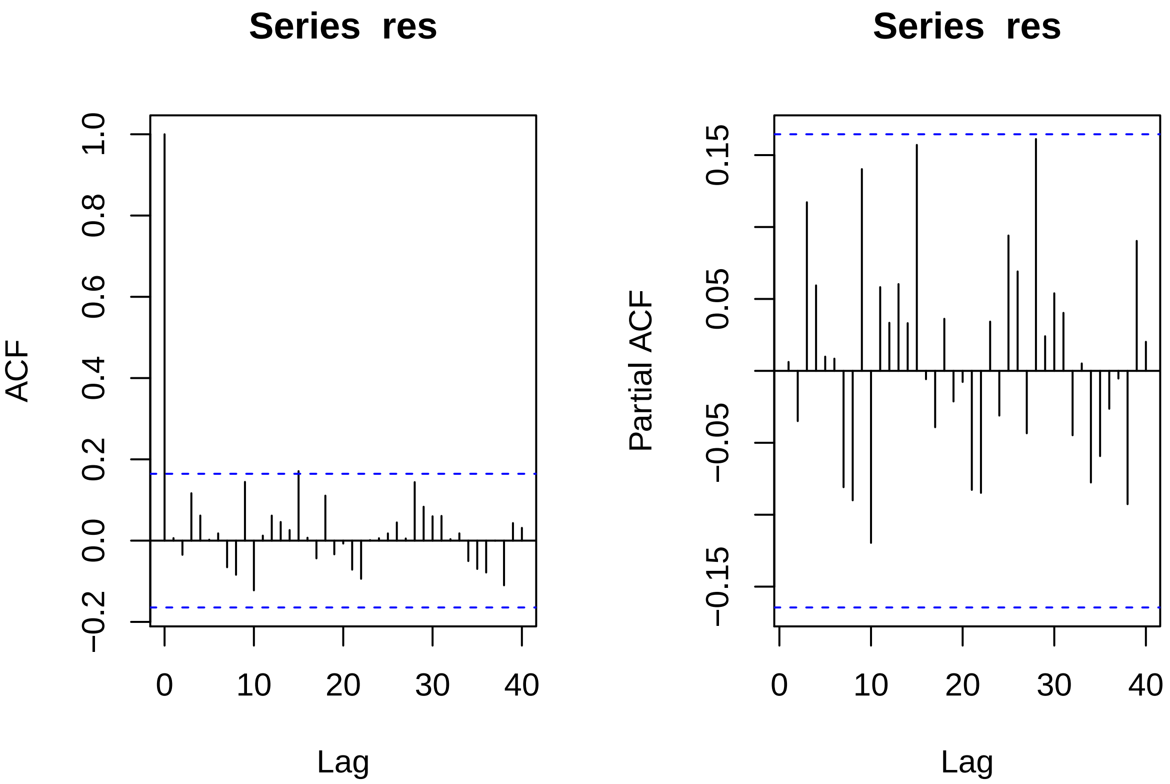
Diagnostic testing: Model B : SARIMA(1,1,1)(0,1,2)s=4







From the time series plot, we can see close to no change in variance, no seasonality, no trend, and the sample mean lies very close to 0 so it resembles white noise. From the normal QQ plot and histogram, we see that the residuals also closely follow a normal distribution.



When looking at the ACF and PACF graphs of the residuals, one can see that there are no significant lags and that all of them lie within the confidence interval so they can be counted as 0. This confirms that the residuals resembles a white noise distribution.

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.9916, p-value = 0.5631
```

```
##
##  Box-Pierce test
##
## data:  res
## X-squared = 9.9619, df = 8, p-value = 0.2677
```

```
##
##  Box-Ljung test
##
## data:  res
## X-squared = 10.677, df = 8, p-value = 0.2207
```

```
##
##  Box-Ljung test
##
## data:  res^2
## X-squared = 7.7879, df = 12, p-value = 0.8015
```

For the Shapiro-Wilk test, this model obtained a p-value of 0.5631 which is above our 0.05 p-value threshold. Therefore, we don't reject the null hypothesis of the shapiro-wilk test and conclude that the residuals of the model follow a normal distribution.

For the other 3 portmanteau tests we see similar results so we would not reject the null hypothesis for any of them. We conclude that the residuals does not show non-linear dependence and follow a gaussian WN(0,1).

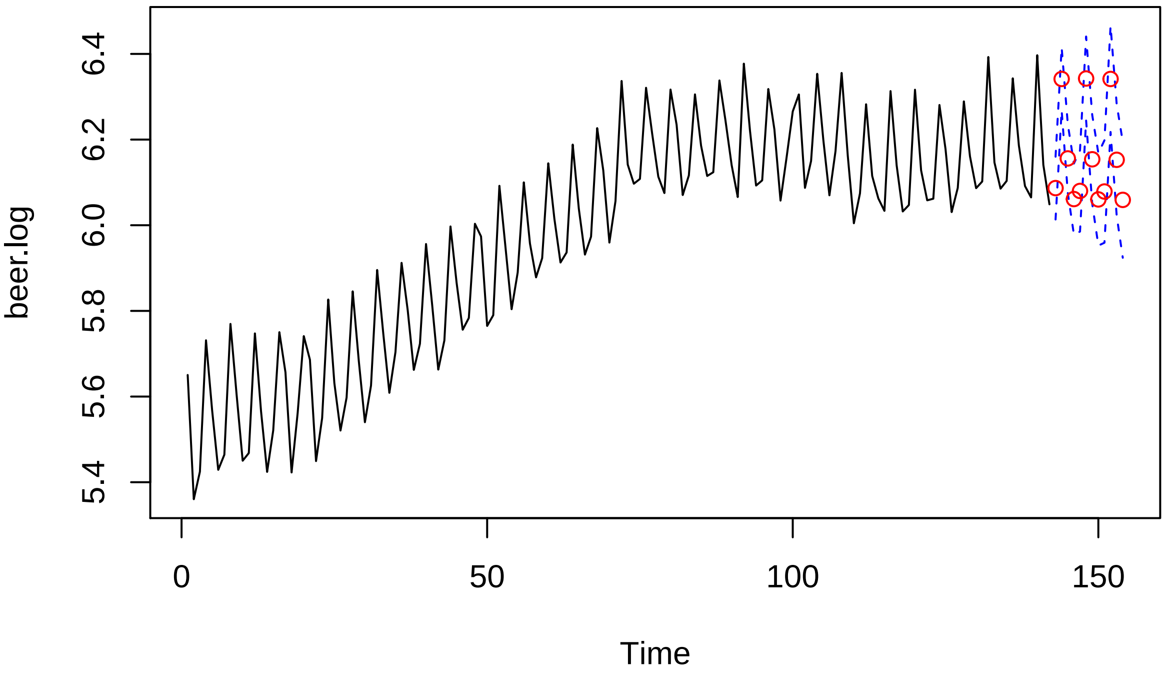
```
##
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.001306
```

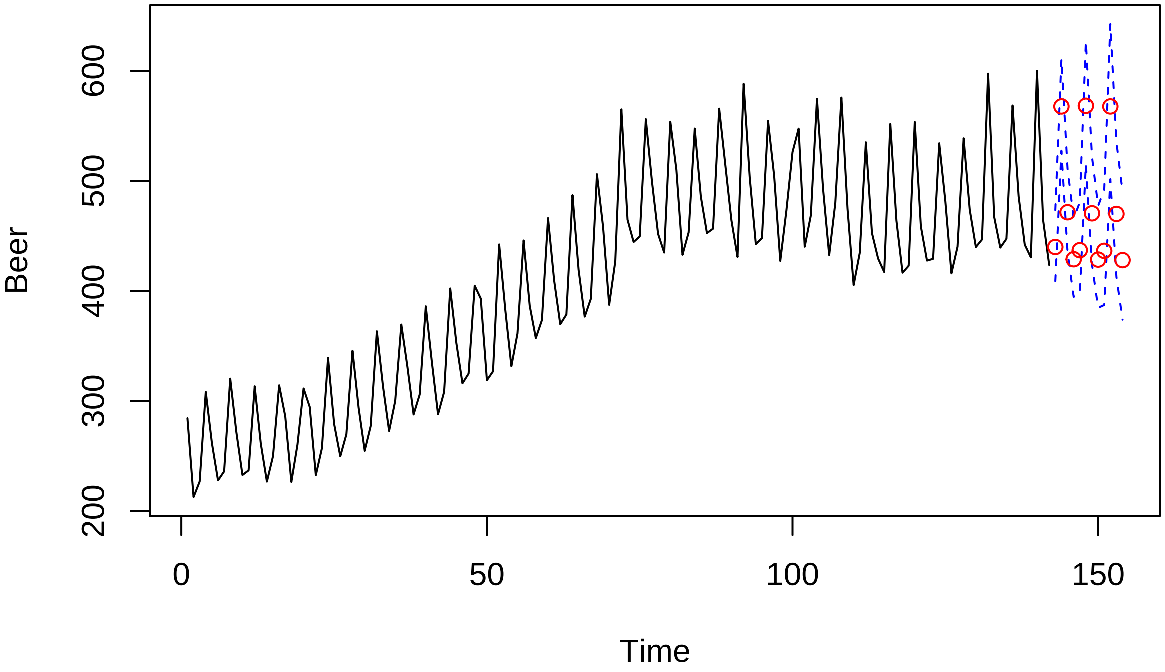
Additionally, r automatically selected order 0 for the residuals so we can conclude that it is in fact an AR(0) model which is white noise.

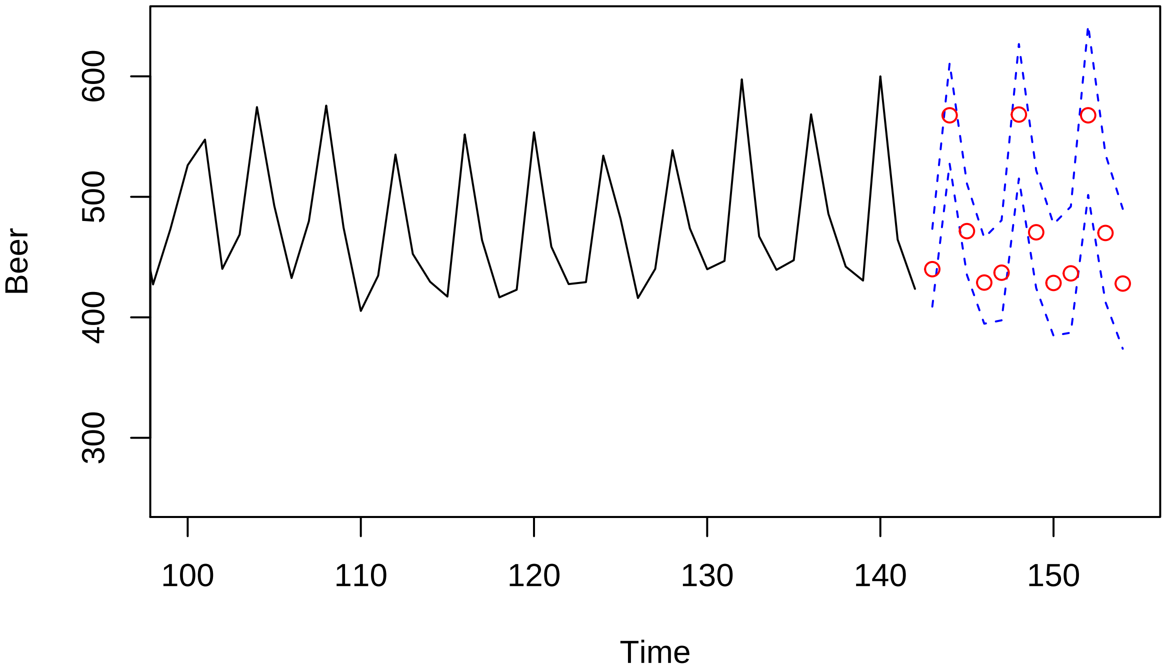
This model passed every diagnostic test so it is a prime candidate for forecasting.

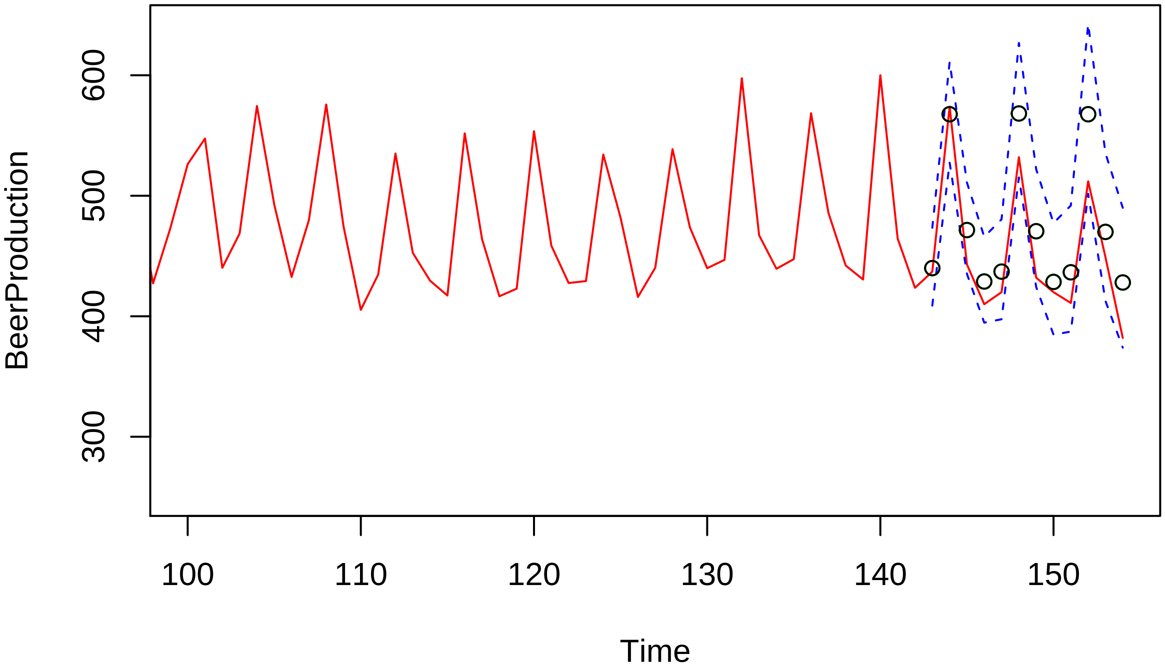
Here, we would choose to forecast model B since, model B has a lower p value than model A and the principle of parsimony states that the model with fewer parameters is better.

Part 8 : Forcasting Model B









Initially, we had split the dataset into a training and test set. The test set contained the last 12 observations. Here we see that the test set are within the prediction intervals therefore, we state that this forecast is valid. Here from the forecasts, we see that the Australian beer industry will continue to maintain and maybe even increase the amount of production of beer.

Conclusion

In conclusion, with the use of transformations, differencing, and diagnostic testing, we are able to determine that model SARIMA(1,1,1)(0,1,2)s=4 was the best model to forecast the given time series dataset. Its formula is given as : $\nabla_1\nabla_4 ln(U_t)(1+0.4085B) = Z_t(1-0.5325B)(1-0.7249B^4)$. The main objective of this project was to forecast the amount beer production in Australia in order to determine whether the Australian beer industry will continue to have a positive impact on the economy. With the given forecast, I was able to determine that the Australia's beer industry will either continue to increase or maintain production which will in return have a positive impact on Australia's economy since it plays a huge role in it. This is because from the forecast we see that the 95% confidence interval on average actually increases in value the later the data, meaning that the beer production will either increase or maintain over large periods of time.

References

Australian Bureau of Statistics

ACIL Allen Consulting, Economic Contribution of the Australian Brewing Industry 2018-19 from Producers to Consumers, March 2020

Appendix

Initializiation of Library

Training / Test Split

```
BeerProduction <- tsdl[[99]]
length(BeerProduction) # 154 obs
BeerProduction = BeerProduction[c(1:154)]
Beer = BeerProduction[c(1:142)] # train set with 154 - 12 obs
Beer.test = BeerProduction[c(143:154)] # test set wtih 12 obs
```

Plot time series and ACF / PACF

```
par(mfrow=c(1,2))
Beer.ts = ts(Beer, frequency = 4, start = c(1956,1))
plot.ts(Beer.ts)
hist(Beer.ts) #histogram is slightly skewed so we try transformations
```

```
Acf(Beer.ts, lag.max = 40)
Pacf(Beer.ts, lag.max = 40)
```

Decomposition

```
y <- ts(as.ts(Beer), frequency = 4)
decomp <- decompose(y)
plot(decomp) # there is a trend + seasonal component
```

Transformations + Looking at variance

```
bcTransform <- boxcox(Beer~ as.numeric(1:length(Beer.ts)))
```

```
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
Beer.bc <-  (1/lambda)*(Beer^lambda-1)
beer.log <- log(Beer)
par(mfrow=c(1,3))
hist(beer.log)
hist(Beer.bc)
hist(Beer.ts)
```

```
print(paste0('beer.log var : ', var(beer.log)))
print(paste0('beer.bc var : ', var(Beer.bc)))
print(paste0('beer.ts var : ', var(Beer.ts)))
# original log b/c variance is significantly lower
```

Differencing + Looking at variance

```
# remove seasonality
Beer.d4 <- diff(beer.log, 4)
print(paste0('beer differenced at lag 4 once var : ', var(Beer.d4)))
Beer.d42 <- diff(Beer.d4, 4)
print(paste0('beer differenced at lag 4 twice var : ', var(Beer.d42)))
Beer.d4d1 <- diff(Beer.d4, 1)
print(paste0('beer differenced at lag 4 once and lag 1 once : ', var(Beer.d4d1)))
Beer.d4d2 <- diff(Beer.d4d1,1)
print(paste0('beer differenced at lag 4 once and lag 1 twice : ', var(Beer.d4d2)))
```

Acf and Pacf of differenced model

```
par(mfrow = c(1,2))
Acf(Beer.d4d1, lag.max = 40)
Pacf(Beer.d4d1, lag.max = 40)
```

Comparing AIC of different models

```
arima(beer.log, order=c(1,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 503.88
arima(beer.log, order=c(2,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 504.93
arima(beer.log, order=c(5,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 499.87
arima(beer.log, order=c(6,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 501.59
arima(beer.log, order=c(7,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 498.35
arima(beer.log, order=c(8,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# -498.64
arima(beer.log, order=c(1,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 502.69
arima(beer.log, order=c(2,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 503.12
arima(beer.log, order=c(5,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
```

```
## Warning in arima(beer.log, order = c(5, 1, 3), seasonal = list(order = c(0, :
## possible convergence problem: optim gave code = 1
```

```
# - 498.82
arima(beer.log, order=c(6,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
```

```
## Warning in arima(beer.log, order = c(6, 1, 3), seasonal = list(order = c(0, :
## possible convergence problem: optim gave code = 1
```

```
# - 497.59
arima(beer.log, order=c(7,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 496.41
arima(beer.log, order=c(8,1,3), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# - 503.42
```

Removing components that have 0 within confidence interval

```
arima(beer.log, order=c(2,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# get rid of ar2 sma 2 and ma 1
arima(beer.log, order=c(2,1,1), seasonal = list(order = c(0,1,2), period = 4), fixed = c(NA,0,0,NA,0), m
```

```
## Warning in arima(beer.log, order = c(2, 1, 1), seasonal = list(order = c(0, :
## some AR parameters were fixed: setting transform.pars = FALSE
```

```
arima(beer.log, order=c(1,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
# get rid of sma2
arima(beer.log, order=c(1,1,1), seasonal = list(order = c(0,1,2), period = 4), fixed = c(NA,NA,NA,0), me
```

Checking for roots

```
# SARIMA(2,1,1)(0,1,2)s=4

plot.roots(NULL,polyroot(c(1, 0.6835, 0.2851)), main="roots of AR part, nonseasonal")
```

```
print(paste0('root of MA part, nonseasonal : ', polyroot(c(1,-0.2875))))
print(paste0('root of MA part, seasonal : ', polyroot(c(1,-0.6359,-0.1211))))

## all outside unit circle therefore, Model A is invertible + stationary

# SARIMA(1,1,1)(0,1,2)s=4
print(paste0('root of AR part, nonseasonal : ', polyroot(c(1,0.4085))))
plot.roots(NULL,polyroot(c(1,0.5325)), main= 'roots of MA part, nonseasonal')
```

```
plot.roots(NULL,polyroot(c(1,-0.7249)), main= 'roots of MA part, seasonal')
```

```
## stationary + invertible
```

Diagnostic testing Model A:

```
fit <- arima(beer.log, order=c(2,1,1), seasonal = list(order = c(0,1,2), period = 4), method="ML")
res <- residuals(fit)
plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```

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

hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
qqnorm(res,main= "Normal Q-Q Plot for Model B")
qqline(res,col="blue")
```

```
par(mfrow = c(1,2))
Acf(res, lag.max = 40)
Pacf(res, lag.max = 40)
```

```
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 2)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf = 2)
Box.test((res)^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

Diagnostic testing Model B:

```
fit <- arima(beer.log, order=c(1,1,1), seasonal = list(order = c(0,1,2), period = 4), fixed = c(NA,NA,NA
res <- residuals(fit)

plot.ts(res)
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
```

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

hist(res,density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m <- mean(res)
std <- sqrt(var(res))
curve( dnorm(x,m,std), add=TRUE )
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model B")
qqline(res,col="blue")
```

```
par(mfrow = c(1,2))
Acf(res, lag.max = 40)
Pacf(res, lag.max = 40)
```

```
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 4)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf = 4)
Box.test(res^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
```

Forecasting model B

```
fit.A <- arima(beer.log, order=c(1,1,1), seasonal = list(order = c(0,1,2), period = 4), fixed = c(NA,NA
pred.tr <- predict(fit.A, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se # upper bound for prediction
L.tr= pred.tr$pred - 2*pred.tr$se # lower bound
ts.plot(beer.log, xlim=c(1,length(beer.log)+12), ylim = c(min(beer.log),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(beer.log)+1):(length(beer.log)+12), pred.tr$pred, col="red")
```

```
pred.orig <- exp(pred.tr$pred)
U= exp(U.tr)
L= exp(L.tr)
ts.plot(Beer, xlim=c(1,length(Beer)+12), ylim = c(min(Beer),max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Beer)+1):(length(Beer)+12), pred.orig, col="red")
```

```
ts.plot(Beer, xlim = c(100,length(Beer)+12), ylim = c(250,max(U)))
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Beer)+1):(length(Beer)+12), pred.orig, col="red")
```

```
ts.plot(BeerProduction, xlim = c(100,length(Beer)+12), ylim = c(250,max(U)), col="red")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(Beer)+1):(length(Beer)+12), pred.orig, col="green")
points((length(Beer)+1):(length(Beer)+12), pred.orig, col="black")
```