Analysis on Monthly Civilian Labor Force Level in USA

Pstat 174 Final Project

Danming Wang

12/11/2020

Abstract

This report analyzes the monthly civilian labor force level (number in thousands) in the United States from January 2004 to December 2019. In this report, we find the trend and pattern of the U.S. civilian labor force level and build a SARIMA model on training data of year 2004 to 2016 to make predictions for test data of year 2017 to 2019. We start by identifying visible patterns of the time series plot and ACF and PACF of the training data. Then we transform and difference our training data to make it stationary, thereby building SARIMA model on the stationary series. We make preliminary identifications of the model by examine the ACFs and PACFs. Next, we will find Maximum Likelihood Estimates of model parameters and choose a three candidate models with the smallest AICc. We perform diagnostic checking to see if our models are adequate and select the final model following the principle of parsimony. Finally, we use our final model to perform predictions on the test data and compare the predictions with the true values.

We find that the civilian labor force level keeps increasing in general from 2004 to 2019. It has a yearly pattern of being lowest at the beginning of a year and reaching the highest level in the middle of a year. Our final model is SARIMA$(0,1,2) \times (0,1,1)_{12}$, with model equation $\nabla_1\nabla_{12}X_t = (1 - 0.1243_{(0.0828)}B - 0.1170_{(0.0758)}B^2)(1 - 0.7193_{(0.0684)}B^{12})Z_t$, $\hat{\sigma}_Z^2 = 134048$. It gives reliable forecasts of the test data with all of the true values from the test data of year 2017 to 2019 fall inside the prediction interval.

Introduction

The dataset we used in this report is obtained from the monthly Current Population Survey (CPS) conducted by the U.S. Bureau of Labor Statistics. [1] It contains monthly civilian labor force in thousands of people from January 2004 to December 2019. Our dataset starts from 2004 because data from a distant past is not helpful for predicting the recent future. Also, we do not include data of 2020 since the COVID-19 pandemic is a special case which induces considerable impact on the U.S. civilian labor force level.

The civilian labor force is a term used by the Bureau of Labor Statistics to refer to "Americans (16 years of age and older) whom it considers either employed or unemployed; active-duty military personnel, federal government employees,

retirees, handicapped or discouraged workers, and agricultural workers are not part of the civilian labor force." [2] Nowadays, an increase in the demand for goods and services leads to rapid economic growth. So we need enough people with right skills to meet increasing demand and sustain the economic growth. The civilian labor force is an important labor market measure since it represents the amount of labor resources available for the production of goods and services. And labor force growth is "an important supply constraint on overall economic growth." [3]

In this report, I will use R to discover the trend and pattern of this dataset, build a time series model on this dataset, and use the model to forecast the future changes. We divide the dataset into two parts, a training dataset (Jan 2004 - Dec 2016) for building model and a test dataset (Jan 2017 - Dec 2019) for examining the forecasting performance.

In section 1, we plot the time series and histogram of the training data X_t as well as its ACFs and PACFs. We find that the training data X_t displays an overall increasing trend and seasonal pattern and has approximately constant variance. Also, the histogram of X_t shows slight skewness.

In section 2, we apply transformations and differencing to make the training data X_t stationary. We try different types of transformations including Box-Cox transformation, logarithm transformation, and squareroot transformation, none of which can correct the skewness of X_t. So we proceed to differencing with the original training data X_t. After confirming that X_t has a period of 12 months using periodogram, we difference X_t at lags 12 once to remove seasoanlity. Then we difference X_t at lags 1 to remove linear trend and get a stationary time series $\nabla_1\nabla_{12}X_t$.

In section 3, we do preliminary identification of candidate models by examining the ACFs and PACFs of the stationary time series $\nabla_1\nabla_{12}X_t$. We find that the candidate models are SARIMA models with d=1, D=1, s=12, q=1 or 5, p=1 or 5, Q=1, and P=2 or 3. Another possible model is the Moving Average model MA(17).

In section 4, we fit the candidate models with Maximum Likelihood Estimates of model parameters and set the insignificant parameters to zero according to the 95% confidence interval. We obtain three models A, B, C with the three smallest AICcs and check that all of them are invertible and stationary by looking at the plot of roots and the absolute value of coefficients.

In section 5, we perform diagnostic checking on the residuals for the three models by using time series plot, histogram, normal Q-Q plot, Shapiro-Wilk test, ACFs and PACFs, Yule-Walker estimation, and Portmanteau tests to see if the residuals resemble Gaussian White noise. If so, then our model is adequate. We find that only model A and C pass the diagnostic checking. And then we select model C as the final model by principle of parsimony.

Finally, in section 6, we make predictions for years from 2017 to 2019 using our final model and compare the predictions with true values from the test dataset. Our

predictions capture the trend and seasonality of the original data. And the true values from the test data are within the prediction intervals.

1. Plot and Analyze the Time Series

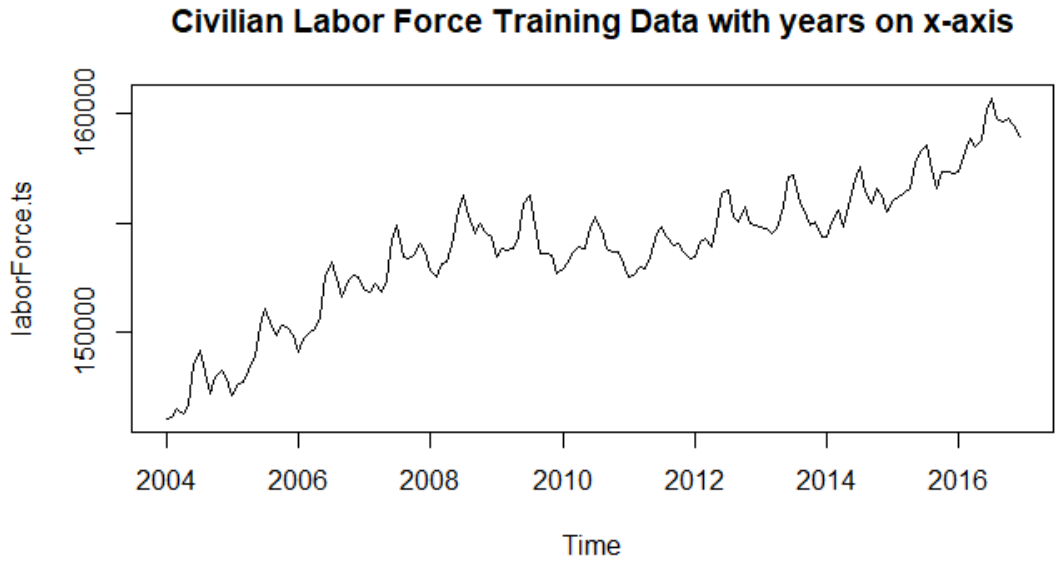
```
summary(laborForce)
```

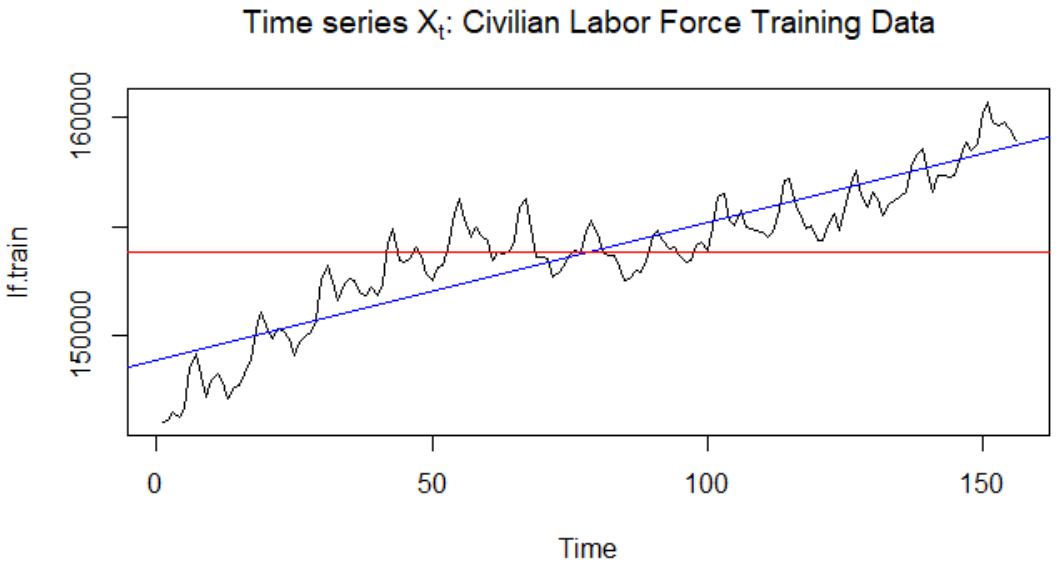
```
##       Year         Month                data       
##  Min.   :2004   Length:192         Min.   :146068  
##  1st Qu.:2008   Class :character   1st Qu.:153107  
##  Median :2012   Mode  :character   Median :154803  
##  Mean   :2012                      Mean   :155359  
##  3rd Qu.:2015                      3rd Qu.:158280  
##  Max.   :2019                      Max.   :164941
```

First, we partition the dataset into two parts: a training dataset with data points from January 2004 to December 2016 (156 data points) and a test dataset with data points from January 2017 to December 2019 (36 data points). Let X_t denote the time series of the monthly total civilian labor force in thousands of people, $\{X_t, t = 1,2,\ldots,156\}$.

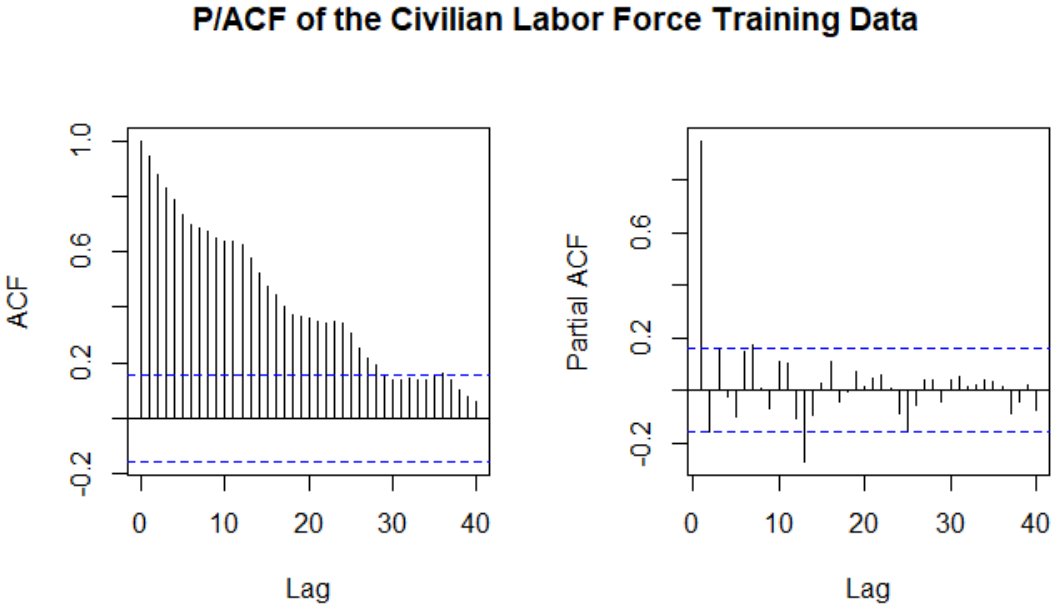
```
lf=laborForce$data
i=length(lf)
lf.train = lf[c(1:(i-36))] # training dataset X_t
lf.test = lf[c((i-35):i)] # test dataset
```

Second, we plot the training dataset.

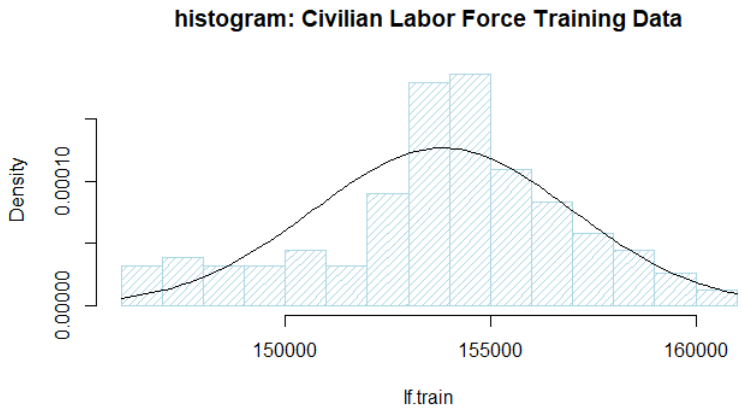




We can immediately observed from the time series plot that our training data X_t is highly non-stationary with positive linear trend and seasonality of 12 months. Each year, the civilian labor force level is lowest at the beginning of a year and then reaches the highest level in the middle of a year. Besides, the time series plot shows that X_t has almost constant variance since there is no apparent sharp changes in behavior. Then we plot the ACF and PACF of X_t to confirm our findings.



The large and periodic ACFs also corresponds to the existence of trend and seasonality, thereby proving the non-stationarity of the original training data X_t.

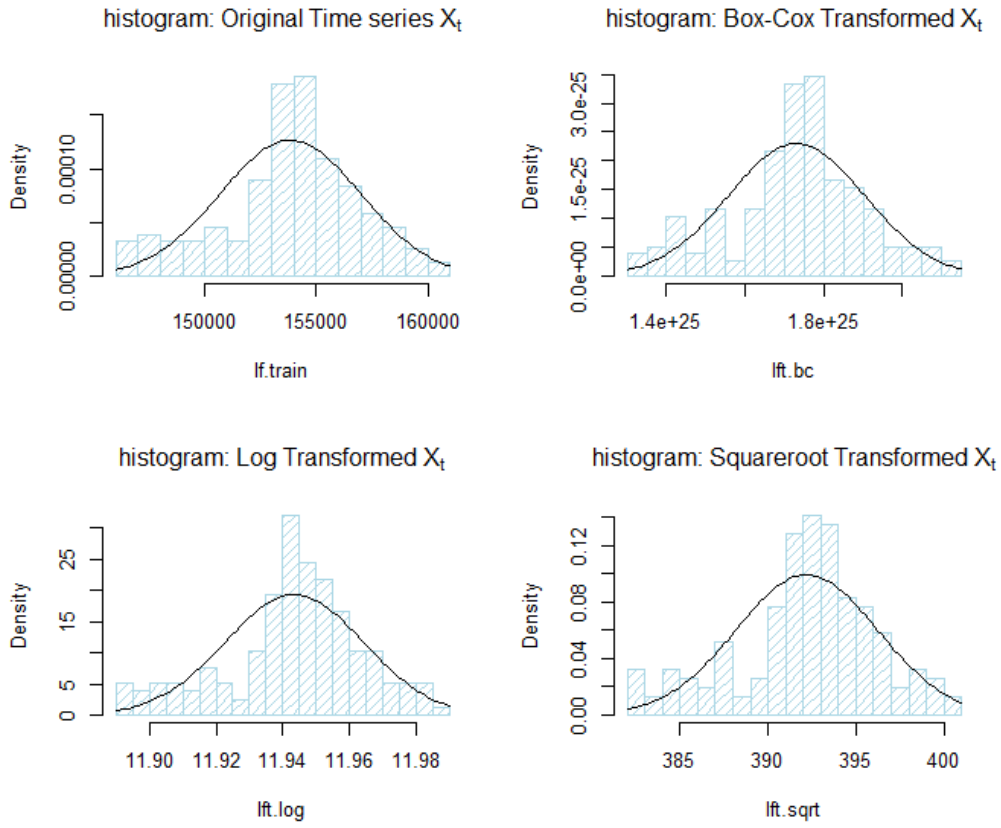


The histogram shows that the distribution of X_t is slightly skewed to the left.

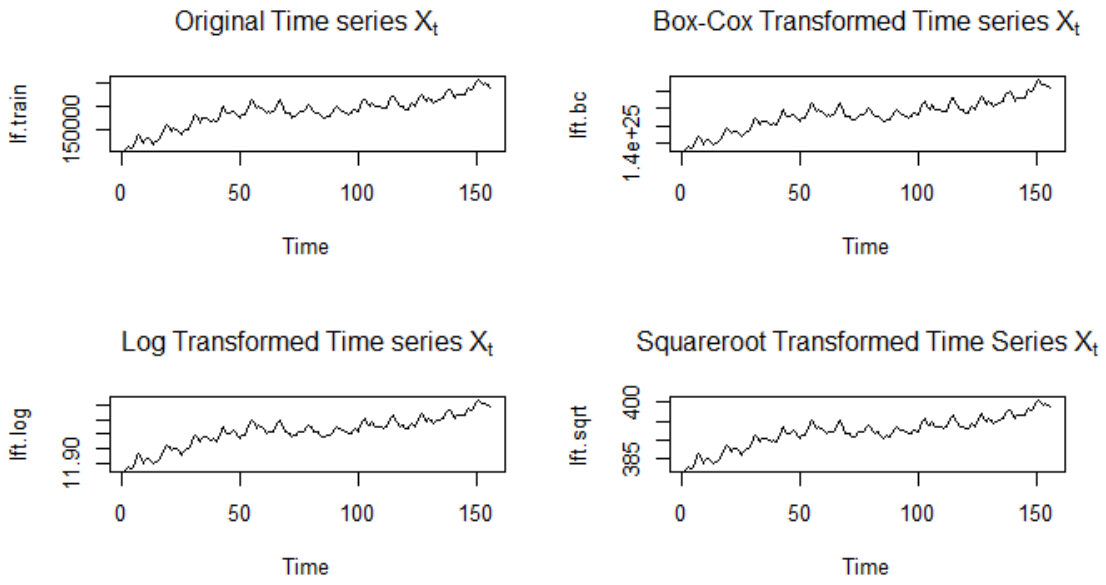
2. Transformation and Differencing

2.1 Transformation

Since the histogram of X_t displays slight skewness, we will perform transformations on X_t to see if we can make the histogram symmetric.

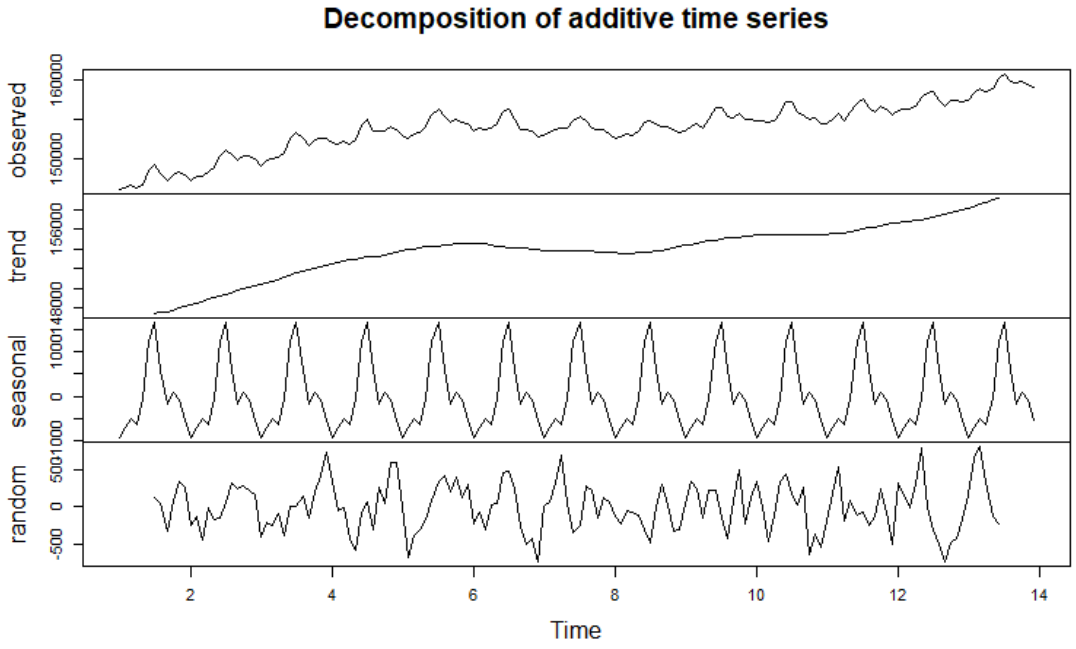


As we can see from the above histograms, none of the Box-Cox, logarithmic, or squareroot transformations can solve the problem of skewness of the training data. Next, we examine the time series plots to see if transformation makes any change.

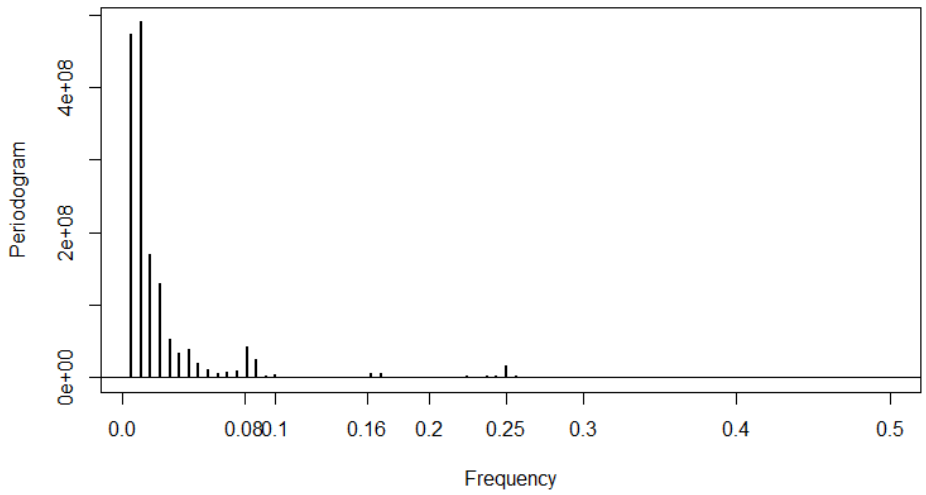


From the time series plots, we can observe that there is no significant change on our time series X_t after those transformations. Thus, we will continue our procedure with the original training data X_t.

2.2 Differencing



Decomposition of X_t displays seasonality of about 12 months and almost linear trend. And we can confirm that the period is indeed 12 months from the periodogram.

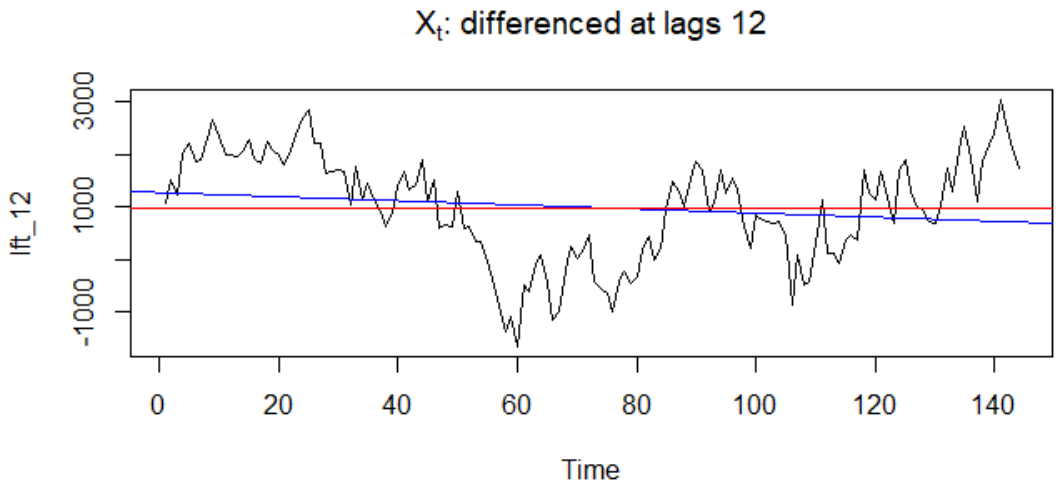


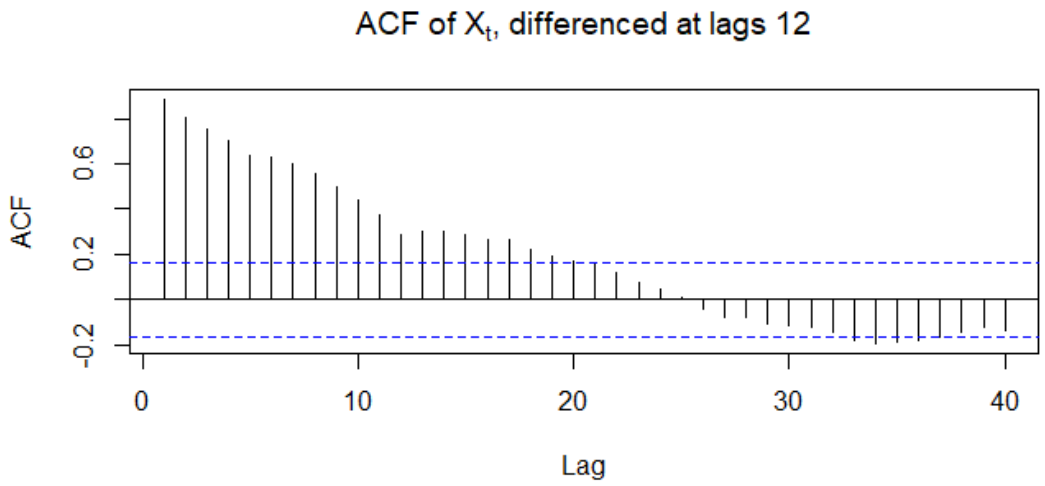
The periodogram has a spike at about $0.08 \approx 1/12$ which indicates a period of 12 months.

Next, we will use differencing to remove the seasonality and linear trend, thereby making X_t a stationary times series.

First, we difference X_t at lags 12 to remove seasonality since X_t has a period of 12 months.

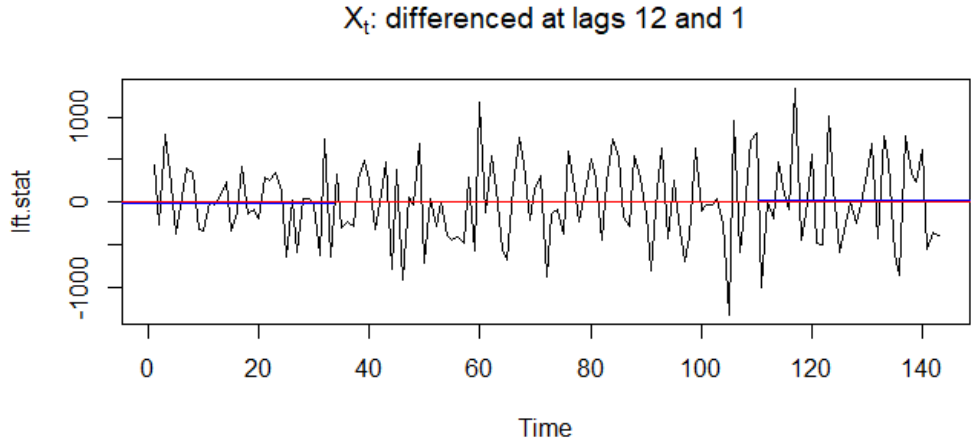
```
lft_12 = diff(lf.train,12)
var(lf.train)

## [1] 9946097

var(lft_12)

## [1] 1008394
```

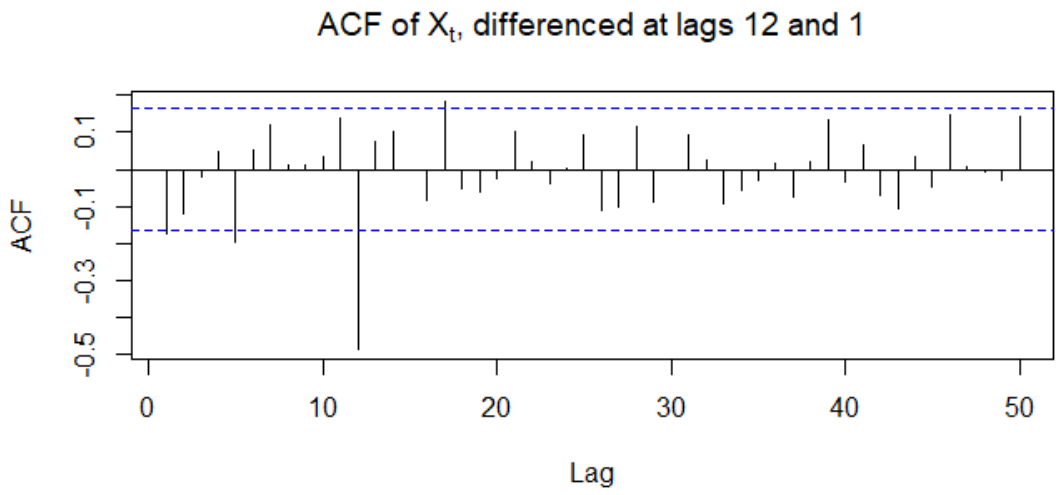




After differencing at lags 12, we can see that seasonality is no longer apparent in the time series plot of X_t differenced at lags 12. Also, as we can see from the plot of ACF of X_t differenced at lags 12, our ACFs are not periodic anymore. The variance drops from 9946097 to 1008394. But the ACFs still decrease slowly, and the time series plot shows that there is still trend. So we will difference $\nabla_{12}X_t$ at lags 1 to remove the trend next.

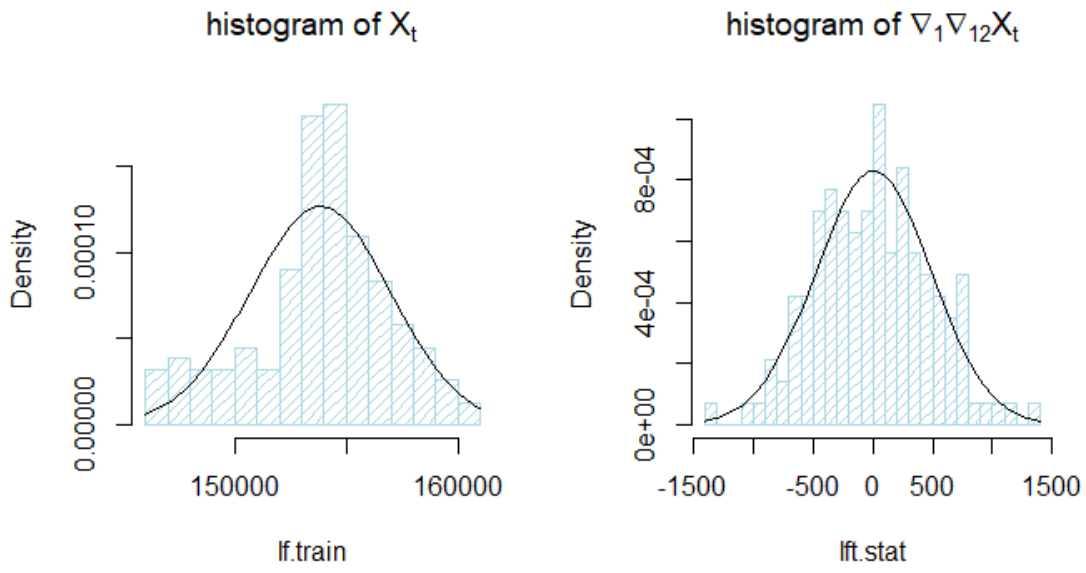
```
lft.stat = diff(lft_12,1)
# Check variance
var(lft_12)

## [1] 1008394

var(lft.stat)

## [1] 230975.8

lft2 = diff(lft.stat,1) # difference at lags 1 again
# Check variance
var(lft2)

## [1] 542809.5
```





Notice that the variance drops from 1008394 to 230975.8 after differencing at lags 1 once and then increases from 230975.8 to 542809.5 if we difference at lags 1 one more time. Thus, differencing at lags 1 once should be enough to remove the linear trend. We can confirm this by examine the time series plot of X_t differenced at lags 12 and then 1, which no longer displays seasonality or trend. Also, the ACF decay of X_t differenced at lags 12 and then 1 corresponds to a stationary process. Therefore, $\nabla_1\nabla_{12}X_t$ is a stationary time series and it is reasonable to work with data $\nabla_1\nabla_{12}X_t$ where X_t =the first 156 observations of the original data for the following analysis.

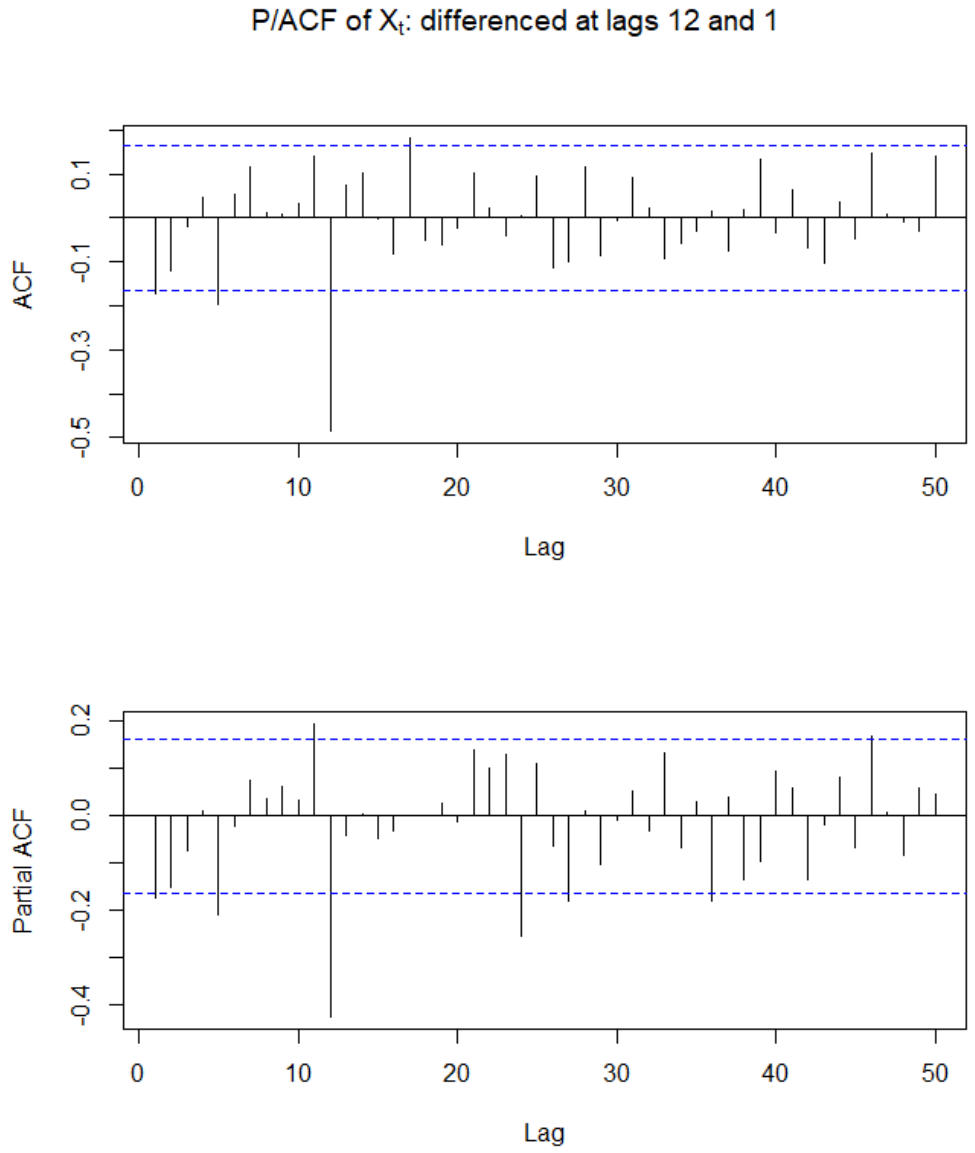
Besides, we can compare the histograms of X_t and $\nabla_1\nabla_{12}X_t$ to see if the distribution of the stationary series is symmetric.



Comparing the histogram of $\nabla_1\nabla_{12}X_t$ with the histogram of the original training data X_t, we can observe that the histogram of $\nabla_1\nabla_{12}$ looks more symmetric and almost

Gaussian. Then we can do the preliminary identification of the model that fits $\nabla_1\nabla_{12}X_t$.

3. Preliminary Identification



From the ACF plot of $\nabla_1\nabla_{12}X_t$, we can see that the ACF is outside the confidence interval at lags 1, 5, 12 and 17. So we may have $q = 1$ or 5 and $Q = 1$.

From the PACF plot of $\nabla_1\nabla_{12}X_t$, we can see that the PACF is outside the confidence interval at lags 1, 5, 11, 12, 24, 27, and 36. So we may have $p = 1$ or 5 and $P = 2$ or 3. Also, it is very likely that $p = 1$ since the PACF is significant at lags 1 and lags $12 - 1 = 11$ but not significant at lags $12 - 5 = 7$.

Since we difference X_t at lags 12 once to remove seasonality and then difference $\nabla_{12}X_t$ at lags 1 once to remove trend, we have $s = 12$, $D = 1$, and $d = 1$.

Thus, a list of candidate models to try is:
SARIMA model with the following parameters: d=1, D=1, s=12; q=1 or 5; p=1 or 5; Q=1; P=2 or 3;
Moving Average model: MA(17).

4. Fit the Model

List of candidate models to try:

SARIMA: d=1, D=1, s=12; q=1-5; p=1-5; Q=1; P=2 or 3;
MA(17).

4.1 Pure SMA Model

First, we will fit the training data into a pure SMA model. We try SMA models with MA parameters: $Q = 1,\ q = 1,2,3,4,5$.

```
for(q in 1:5){
  print(paste("Q,q:",1,q," ", AICc(arima(lf.train,order=c(0,1,q),season
al=list(order=c(0,1,1),period=12),method='ML'))
  ))}
```

```
## [1] "Q,q: 1 1   2111.20812565686"
## [1] "Q,q: 1 2   2111.01375577188"
## [1] "Q,q: 1 3   2112.90620235648"
## [1] "Q,q: 1 4   2113.57681284434"
## [1] "Q,q: 1 5   2112.92969459797"
```

- The AICc is lowest at $Q = 1$ and $q = 2$ with a value of 2111.014. Then we will check the parameters β_j of the model with their approximate 95% confidence intervals $\beta_j \pm (1.96 \cdot se)$.

```
SMA1 = arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA1
```

```
##
## Call:
## arima(x = lf.train, order = c(0, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.1243  -0.1170  -0.7193
## s.e.   0.0828   0.0758   0.0684
##
## sigma^2 estimated as 134048:  log likelihood = -1051.43,  aic = 2108
.86
```

```
AICc(SMA1)
```

```
## [1] 2111.014

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA1$coef-1.96*sqrt(diag(SMA1$var.coef)),SMA1$coef+1.96*sqrt
(diag(SMA1$var.coef)) )
CI

##            [,1]        [,2]
## ma1  -0.2865395  0.03794591
## ma2  -0.2656013  0.03163469
## sma1 -0.8534241 -0.58519810
```

Since the 95% confidence intervals for the parameters `ma1` and `ma2` contain zero, we will set `ma1` and `ma2` to zero on by one.

```
# Set ma1 to zero
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,NA), method="ML")

##
## Call:
## arima(x = lf.train, order = c(0, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), fixed = c(0, NA, NA), method = "ML")
##
## Coefficients:
##       ma1      ma2     sma1
##         0  -0.1038  -0.7326
## s.e.    0   0.0740   0.0665
##
## sigma^2 estimated as 135691:  log likelihood = -1052.53,  aic = 2109
.06

AICc(arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA,NA), method="ML"))

## [1] 2111.216
```

Since the AICc increases from 2111.014 to 2111.216 after removing `ma1`, it means that 0 is not the value for `ma1`. Then we try to remove `ma2`.

```
# Set ma2 to zero
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(NA,0,NA), method="ML")

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

```
##           ma1  ma2     sma1
##       -0.1341    0  -0.7175
## s.e.   0.0977    0   0.0677
##
## sigma^2 estimated as 136285:  log likelihood = -1052.56,  aic = 2109
.13
```

```
AICc(arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(NA,0,NA), method="ML"))
```

```
## [1] 2111.288
```

Since the AICc increases from 2111.014 to 2111.288 after removing `ma2`, it means that 0 is not the value for `ma2`.

So the best model we obtained so far is the pure SMA model SARIMA$(0,1,2) \times (0,1,1)_{12}$ with `ma1` and `ma2` not removed.

- The AICc is second lowest at $Q = 1$ and $q = 1$ with a value of 2111.208. Then we will check the parameters of this model using the 95% confidence intervals.

```
SMA2 = arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA2
```

```
##
## Call:
## arima(x = lf.train, order = c(0, 1, 1), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1     sma1
##       -0.1341  -0.7175
## s.e.   0.0977   0.0677
##
## sigma^2 estimated as 136285:  log likelihood = -1052.56,  aic = 2109
.13
```

```
AICc(SMA2)
```

```
## [1] 2111.208
```

```
# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA2$coef-1.96*sqrt(diag(SMA2$var.coef)),SMA2$coef+1.96*sqrt
(diag(SMA2$var.coef)) )
CI
```

```
##            [,1]       [,2]
## ma1  -0.3254922  0.0573364
## sma1 -0.8502537 -0.5848002
```

Since the 95% confidence interval for the parameter `ma1` is $(-0.3254922, 0.0573364)$, which contains zero, we will set `ma1` to zero.

```
# Set insignificant parameters to zero
arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA), method="ML")

##
## Call:
## arima(x = lf.train, order = c(0, 1, 1), seasonal = list(order = c(0,
 1, 1),
##     period = 12), fixed = c(0, NA), method = "ML")
##
## Coefficients:
##       ma1     sma1
##         0  -0.7307
## s.e.    0   0.0662
##
## sigma^2 estimated as 137608:  log likelihood = -1053.49,  aic = 2108
.97

AICc(arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA), method="ML"))

## [1] 2111.05
```

Since the AICc drops from 2111.208 to 2111.05 after we set the insignificant parameter `ma1` to zero, we can assume that $q = 0$ rather than 1 and obtain the new model SARIMA$(0,1,0) \times (0,1,1)_{12}$.

```
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")

##
## Call:
## arima(x = lf.train, order = c(0, 1, 0), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          sma1
##       -0.7307
## s.e.   0.0662
##
## sigma^2 estimated as 137608:  log likelihood = -1053.49,  aic = 2108
.97

AICc(arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1),
period = 12), method="ML"))

## [1] 2110.998
```

The SARIMA$(0,1,0) \times (0,1,1)_{12}$ model produces an AICc of $2110.998 < 2111.014$.

- The AICc is third lowest at $Q = 1$ and $q = 3$ with a value of 2112.906. Then we will check the parameters of this model.

```
SMA3 = arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA3

##
## Call:
## arima(x = lf.train, order = c(0, 1, 3), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1      ma2      ma3     sma1
##       -0.1120  -0.1186  -0.0364  -0.7217
## s.e.   0.0878   0.0750   0.0787   0.0693
##
## sigma^2 estimated as 133764:  log likelihood = -1051.32,  aic = 2110
.64

AICc(SMA3)

## [1] 2112.906

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA3$coef-1.96*sqrt(diag(SMA3$var.coef)),SMA3$coef+1.96*sqrt
(diag(SMA3$var.coef)) )
CI

##            [,1]        [,2]
## ma1  -0.2840397  0.06001373
## ma2  -0.2655098  0.02832958
## ma3  -0.1907212  0.11785198
## sma1 -0.8574047 -0.58590864
```

Since the 95% confidence intervals for the parameters `ma1`, `ma2`, and `ma3` contain zero, we will set `ma1`, `ma2`, and `ma3` to zero one by one.

```
# Set ma1 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,NA,NA), method="ML")

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

```
##       ma1      ma2      ma3     sma1
##         0  -0.1247  -0.0723  -0.7371
## s.e.    0   0.0754   0.0782   0.0678
##
## sigma^2 estimated as 134732:  log likelihood = -1052.12,  aic = 2110
.24
```

```
AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA,NA,NA), method="ML"))
```

```
## [1] 2112.507
```

The AICc drops from 2112.906 to 2112.507. Then we continue to set `ma2` to zero.

```
# Set ma2 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,0,NA,NA), method="ML")
```

```
##
## Call:
## arima(x = lf.train, order = c(0, 1, 3), seasonal = list(order = c(0,
 1, 1),
##     period = 12), fixed = c(0, 0, NA, NA), method = "ML")
##
## Coefficients:
##       ma1  ma2      ma3     sma1
##         0    0  -0.0328  -0.7330
## s.e.    0    0   0.0812   0.0668
##
## sigma^2 estimated as 137363:  log likelihood = -1053.4,  aic = 2110.
81
```

```
AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,0,NA,NA), method="ML"))
```

```
## [1] 2113.074
```

Since the AICc grows from 2112.507 to 2113.074 after we set `ma2` to zero, it means that `ma2` cannot be removed. Then we try to remove `ma3`.

```
# Set ma3 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,0,NA), method="ML")
```

```
##
## Call:
## arima(x = lf.train, order = c(0, 1, 3), seasonal = list(order = c(0,
 1, 1),
##     period = 12), fixed = c(0, NA, 0, NA), method = "ML")
##
## Coefficients:
##       ma1      ma2  ma3     sma1
```

```
##         0  -0.1038    0  -0.7326
## s.e.    0   0.0740    0   0.0665
##
## sigma^2 estimated as 135691:  log likelihood = -1052.53,  aic = 2109
.06

AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA,0,NA), method="ML"))

## [1] 2111.323
```

The AICc drops from 2112.507 to 2111.323. So the model becomes SARIMA(0,1,3) × $(0,1,1)_{12}$ with `ma1` and `ma3` removed. In other words, this is the SARIMA(0,1,2) × $(0,1,1)_{12}$ model with `ma1` removed.

```
SMA3 = arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1)
, period = 12),fixed=c(0,NA,NA), method="ML")
SMA3

##
## Call:
## arima(x = lf.train, order = c(0, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), fixed = c(0, NA, NA), method = "ML")
##
## Coefficients:
##       ma1      ma2     sma1
##         0  -0.1038  -0.7326
## s.e.    0   0.0740   0.0665
##
## sigma^2 estimated as 135691:  log likelihood = -1052.53,  aic = 2109
.06

AICc(SMA3)

## [1] 2111.216
```

- Since 2111.216 > 2111.014 > 2110.998, the best model we obtained so far is SARIMA(0,1,0) × $(0,1,1)_{12}$, the second-best model is SARIMA(0,1,2) × $(0,1,1)_{12}$, and the third-best model is SARIMA(0,1,2) × $(0,1,1)_{12}$ with `ma1` removed.

4.2 Pure SAR Model

Second, we try to fit the training data in a pure SAR model with AR parameters: $P = 1,2,3$ and $q = 1,2,3,4,5$. And we will see if the AICc of the SAR model is lower than that of the three best SMA models.

```
for(P in 1:3)
  for(p in 1:5){
    aicc = try(AICc(arima(lf.train,order=c(p,1,0),seasonal=list(order=c
```

```
(P,1,0),period=12),method='ML')))
    if (inherits(aicc,"try-error")){
    # skip this iteration if error happens
    next
  }
  print(paste("P,p:",P,p," ", aicc))}
```

```
## [1] "P,p: 1 1   2131.06845035364"
## [1] "P,p: 1 2   2131.30477015434"
## [1] "P,p: 1 3   2132.86128889565"
## [1] "P,p: 1 4   2134.93665129476"
## [1] "P,p: 1 5   2134.31950033963"
## [1] "P,p: 2 1   2120.30699911874"
## [1] "P,p: 2 2   2119.86968906632"
## [1] "P,p: 2 3   2121.62660711892"
## [1] "P,p: 2 4   2123.47487870605"
## [1] "P,p: 2 5   2122.21077224422"
## Error in optim(init[mask], armafn, method = optim.method, hessian =
TRUE,  :
##   non-finite finite-difference value [1]
## [1] "P,p: 3 2   2114.66504906109"
## [1] "P,p: 3 3   2116.62900017102"
## [1] "P,p: 3 4   2117.98323199075"
## [1] "P,p: 3 5   2118.45209655753"
```

The error here means that it is impossible to have $P = 3$ and $p = 1$ for our model. Then the AICc for pure SAR model is lowest at $P = 3$ and $p = 2$ with a value of 2114.665. Then we will check the parameters of this model using the 95% confidence interval.

```
SAR = arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0),
 period = 12), method="ML")
SAR
```

```
##
## Call:
## arima(x = lf.train, order = c(2, 1, 0), seasonal = list(order = c(3,
 1, 0),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2     sar1     sar2     sar3
##       -0.0900  -0.1706  -0.7573  -0.4872  -0.2554
## s.e.   0.0857   0.0835   0.0854   0.1011   0.0913
##
## sigma^2 estimated as 133588:  log likelihood = -1051.13,  aic = 2112
.27
```

```
AICc(SAR)
```

```
## [1] 2114.665
```

```r
# Calculate approximate 95% confidence interval for parameters
CI = cbind(SAR$coef-1.96*sqrt(diag(SAR$var.coef)),SAR$coef+1.96*sqrt(di
ag(SAR$var.coef)) )
CI
```

```
##            [,1]         [,2]
## ar1  -0.2579610  0.078024429
## ar2  -0.3343615 -0.006873892
## sar1 -0.9247109 -0.589822230
## sar2 -0.6854127 -0.288948199
## sar3 -0.4343214 -0.076452772
```

Since the 95% confidence interval for the parameter `ar1` is $(-0.2579610, 0.078024429)$, which contains zero, we will set `ar1` to zero.

```r
# Set insignificant parameters to zero
arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0), perio
d = 12),fixed=c(0,NA,NA,NA,NA), method="ML")
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, x
reg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALS
E

##
## Call:
## arima(x = lf.train, order = c(2, 1, 0), seasonal = list(order = c(3,
 1, 0),
##     period = 12), fixed = c(0, NA, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1      ar2     sar1     sar2     sar3
##         0  -0.1669  -0.7776  -0.5128  -0.2610
## s.e.    0   0.0840   0.0830   0.0985   0.0914
##
## sigma^2 estimated as 134173:  log likelihood = -1051.68,  aic = 2111
.36
```

```r
AICc(arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0),
period = 12), fixed=c(0,NA,NA,NA,NA),method="ML"))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, x
reg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALS
E

## [1] 2113.765
```

Since the AICc drops from 2114.665 to 2113.765 after we set the insignificant parameter `ar1` to zero, the best SAR model we obtained is the pure SAR model SARIMA$(2,1,0) \times (3,1,0)_{12}$ with `ar1` removed.

Since 2113.765 > 2111.216 > 2111.014 > 2110.998, the best three models we obtained so far are still:
SARIMA(0,1,0) × $(0,1,1)_{12}$, SARIMA(0,1,2) × $(0,1,1)_{12}$, and SARIMA(0,1,2) × $(0,1,1)_{12}$ with `ma1` removed.

4.3 MA model

Third, we try to fit our training data into an MA(17) model.

```
MA=arima(lf.train, order=c(0,0,17),method = "ML")
MA

##
## Call:
## arima(x = lf.train, order = c(0, 0, 17), method = "ML")
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5     ma6     ma7     ma8
     ma9
##       1.3389  1.2814  1.2440  1.4309  1.2150  1.0253  1.1899  1.5903
  1.5165
## s.e.  0.1061  0.1418  0.2924  0.3356  0.2153  0.2062  0.2109  0.2678
  0.2974
##         ma10    ma11    ma12    ma13    ma14    ma15    ma16   ma17
##       1.0797  1.0769  1.3639  1.3700  1.3070  1.2833  1.0148  0.415
## s.e.  0.2715  0.2331  0.1948  0.1795  0.2119  0.2944  0.2603  0.125
##         intercept
##       153745.6731
## s.e.     802.6451
##
## sigma^2 estimated as 234235:  log likelihood = -1195.14,  aic = 2426
.29

AICc(MA)

## [1] 2433.282

# Calculate approximate 95% confidence interval for parameters
CI = cbind(MA$coef-1.96*sqrt(diag(MA$var.coef)),MA$coef+1.96*sqrt(diag(
MA$var.coef)) )
CI

##                   [,1]         [,2]
## ma1       1.130888e+00 1.546937e+00
## ma2       1.003479e+00 1.559387e+00
## ma3       6.708925e-01 1.817143e+00
## ma4       7.731052e-01 2.088673e+00
## ma5       7.930269e-01 1.636907e+00
## ma6       6.211838e-01 1.429379e+00
## ma7       7.765537e-01 1.603344e+00
## ma8       1.065445e+00 2.115148e+00
```

```
## ma9       9.336834e-01 2.099300e+00
## ma10      5.475427e-01 1.611762e+00
## ma11      6.199651e-01 1.533895e+00
## ma12      9.820432e-01 1.745751e+00
## ma13      1.018218e+00 1.721870e+00
## ma14      8.916891e-01 1.722272e+00
## ma15      7.062812e-01 1.860235e+00
## ma16      5.045763e-01 1.525117e+00
## ma17      1.699726e-01 6.601022e-01
## intercept 1.521725e+05 1.553189e+05
```

We can see that none of the 95% confidence intervals for the parameters contains zero, so all of these parameters are significant. However, since the AICc for MA(17) model, 2433.282, is much larger than the AICc of the previous models, the best three models we obtained so far are still SARIMA$(0,1,0) \times (0,1,1)_{12}$, SARIMA$(0,1,2) \times (0,1,1)_{12}$, and SARIMA$(0,1,2) \times (0,1,1)_{12}$ with `ma1` removed.

4.4 SARIMA Model

In this section, we will see if adding seasonal or nonseasonal AR part to the SMA model can produce a better model with lower AICc.

4.4.1 Introducing Seasonal AR Part to the SMA Model

SARIMA$(0,1,q) \times (P,1,1)_{12}$ models tried: P=1,2,3; q=0 or 2

```
for(P in 1:3)
  for(q in c(0,2)){
  print(paste("P,p,Q,q:",P,0,1,q," ",
  AICc(arima(lf.train,order=c(0,1,q),seasonal=list(order=c(P,1,1),perio
d=12),method='ML'))))}
```

```
## [1] "P,p,Q,q: 1 0 1 0   2111.59886452695"
## [1] "P,p,Q,q: 1 0 1 2   2112.15322311519"
## [1] "P,p,Q,q: 2 0 1 0   2113.33542814563"
## [1] "P,p,Q,q: 2 0 1 2   2114.00745326828"
## [1] "P,p,Q,q: 3 0 1 0   2114.95466388227"
## [1] "P,p,Q,q: 3 0 1 2   2115.48919106223"
```

The AICc for SARIMA$(0,1,q) \times (P,1,1)_{12}$ model is lowest at $P = 1$ and $q = 0$ with a value of 2111.599. Then we will check the parameters of this model using 95% confidence interval.

```
SARIMA1 = arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1
,1), period = 12), method="ML")
SARIMA1
```

```
##
## Call:
## arima(x = lf.train, order = c(0, 1, 0), seasonal = list(order = c(1,
 1, 1),
```

```
##     period = 12), method = "ML")
##
## Coefficients:
##          sar1    sma1
##       -0.1419  -0.654
## s.e.   0.1144   0.094
##
## sigma^2 estimated as 136300:  log likelihood = -1052.76,  aic = 2109
.52

AICc(SARIMA1)

## [1] 2111.599

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SARIMA1$coef-1.96*sqrt(diag(SARIMA1$var.coef)),SARIMA1$coef+
1.96*sqrt(diag(SARIMA1$var.coef)) )
CI

##            [,1]        [,2]
## sar1 -0.3661728  0.08234346
## sma1 -0.8381608 -0.46980140
```

Since the 95% confidence interval for the parameter `sar1` contains zero, we will set `sar1` to zero.

```
# Set insignificant parameters to zero
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1,1), perio
d = 12),fixed=c(0,NA), method="ML")

## Warning in stats::arima(x = x, order = order, seasonal = seasonal, x
reg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALS
E

##
## Call:
## arima(x = lf.train, order = c(0, 1, 0), seasonal = list(order = c(1,
 1, 1),
##     period = 12), fixed = c(0, NA), method = "ML")
##
## Coefficients:
##       sar1     sma1
##          0  -1.3686
## s.e.     0   0.1241
##
## sigma^2 estimated as 73465:  log likelihood = -1053.49,  aic = 2108.
97

AICc(arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1,1),
period = 12), fixed=c(0,NA),method="ML"))
```

```
## Warning in stats::arima(x = x, order = order, seasonal = seasonal, x
reg =
## xreg, : some AR parameters were fixed: setting transform.pars = FALS
E

## [1] 2111.05
```

After removing `sar1`, the AICc drops from 2111.599 to 2111.05. Also, since $2111.216 > 2111.05 > 2111.014 > 2110.998$, the third best model we obtained so far becomes the SARIMA$(0,1,0) \times (1,1,1)_{12}$ model with `sar1` removed.

Thus, the three best models we have by now are:
SARIMA$(0,1,0) \times (0,1,1)_{12}$, SARIMA$(0,1,2) \times (0,1,1)_{12}$, and SARIMA$(0,1,0) \times (1,1,1)_{12}$ with `sar1` removed.

4.4.2 Introducing Non-seasonal AR Part to the SMA Model

SARIMA$(p,1,q) \times (0,1,1)_{12}$ models tried: p=1,2,3,4,5; q=0 or 2

```
for(p in 1:5)
  for(q in c(0,2)){
  print(paste("P,p,Q,q:",0,p,1,q," ",
  AICc(arima(lf.train,order=c(p,1,q),seasonal=list(order=c(0,1,1),perio
d=12),method='ML'))))}

## [1] "P,p,Q,q: 0 1 1 0   2111.71976373659"
## [1] "P,p,Q,q: 0 1 1 2   2113.02614316933"
## [1] "P,p,Q,q: 0 2 1 0   2111.07607868963"
## [1] "P,p,Q,q: 0 2 1 2   2113.5384551187"
## [1] "P,p,Q,q: 0 3 1 0   2112.60463856056"
## [1] "P,p,Q,q: 0 3 1 2   2110.241753689"
## [1] "P,p,Q,q: 0 4 1 0   2113.67059342838"
## [1] "P,p,Q,q: 0 4 1 2   2110.76424737268"
## [1] "P,p,Q,q: 0 5 1 0   2113.00412965843"
## [1] "P,p,Q,q: 0 5 1 2   2117.3212447721"
```

The AICc for SARIMA$(p,1,q) \times (0,1,1)_{12}$ model is lowest at $p = 3$ and $q = 2$ with a value of 2110.242. Then we will check the parameters of this model using 95% confidence interval.

```
SARIMA2 = arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1
,1), period = 12), method="ML")
SARIMA2

##
## Call:
## arima(x = lf.train, order = c(3, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
```

```
##           ar1      ar2      ar3     ma1     ma2     sma1
##       -1.6306  -1.0406  -0.2028  1.5901  0.8805  -0.7037
## s.e.   0.1514   0.1537   0.0944  0.1468  0.1254   0.0756
##
## sigma^2 estimated as 126264:  log likelihood = -1047.84,  aic = 2107
.68
```

```
AICc(SARIMA2)
```

```
## [1] 2110.242
```

```
# Calculate approximate 95% confidence interval for parameters
CI = cbind(SARIMA2$coef-1.96*sqrt(diag(SARIMA2$var.coef)),SARIMA2$coef+
1.96*sqrt(diag(SARIMA2$var.coef)) )
CI
```

```
##            [,1]        [,2]
## ar1  -1.9272595 -1.33392226
## ar2  -1.3417603 -0.73940686
## ar3  -0.3877341 -0.01778166
## ma1   1.3023551  1.87779192
## ma2   0.6347367  1.12631323
## sma1 -0.8519679 -0.55544687
```

Since none of the 95% confidence interval for the parameters contains zero, all of the parameters are significant. So we obtain an SARIMA model SARIMA(3,1,2) × $(0,1,1)_{12}$ with AICc 2110.242.

Since 2111.05 > 2111.014 > 2110.998 > 2110.242, SARIMA(3,1,2) × $(0,1,1)_{12}$ becomes the best model by now. The second-best model is SARIMA(0,1,0) × $(0,1,1)_{12}$ and the third-best model is SARIMA(0,1,2) × $(0,1,1)_{12}$.

4.5 Summary of Models for lft.stat=$\nabla_1\nabla_{12}X_t$

We name the three best models we obtained as model A, B, and C in the increasing order of AICc.

```
# Model A
arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")
```

```
##
## Call:
## arima(x = lf.train, order = c(3, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ar1      ar2      ar3     ma1     ma2     sma1
##       -1.6306  -1.0406  -0.2028  1.5901  0.8805  -0.7037
## s.e.   0.1514   0.1537   0.0944  0.1468  0.1254   0.0756
```

```
##
## sigma^2 estimated as 126264:  log likelihood = -1047.84,  aic = 2107
.68

# Model B
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")

##
## Call:
## arima(x = lf.train, order = c(0, 1, 0), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##          sma1
##       -0.7307
## s.e.   0.0662
##
## sigma^2 estimated as 137608:  log likelihood = -1053.49,  aic = 2108
.97

# Model C
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")

##
## Call:
## arima(x = lf.train, order = c(0, 1, 2), seasonal = list(order = c(0,
 1, 1),
##     period = 12), method = "ML")
##
## Coefficients:
##           ma1      ma2     sma1
##       -0.1243  -0.1170  -0.7193
## s.e.   0.0828   0.0758   0.0684
##
## sigma^2 estimated as 134048:  log likelihood = -1051.43,  aic = 2108
.86
```

The model equations are listed as follows:

Model A: SARIMA$(3,1,2) \times (0,1,1)_{12}$, $(1 + 1.6306_{(0,1514)}B + 1.0406_{(0.1537)}B^2 + 0.2028_{(0.0944)}B^3)\nabla_1\nabla_{12}X_t = (1 + 1.5901_{(0.1468)}B + 0.8805_{(0.1254)}B^2)(1 - 0.7037_{(0.0756)}B^{12})Z_t$, $\hat{\sigma}_Z^2 = 126264$.

Model B: SARIMA$(0,1,0) \times (0,1,1)_{12}$, $\nabla_1\nabla_{12}X_t = (1 - 0.7307_{(0.0662)}B^{12})Z_t$, $\hat{\sigma}_Z^2 = 137608$.

Model C: SARIMA$(0,1,2) \times (0,1,1)_{12}$, $\nabla_1\nabla_{12}X_t = (1 - 0.1243_{(0.0828)}B - 0.1170_{(0.0758)}B^2)(1 - 0.7193_{(0.0684)}B^{12})Z_t$, $\hat{\sigma}_Z^2 = 134048$.

Notice that orders of model A are mostly consistent with our preliminary identification with the exception of $q = 2$ and $P = 0$. Orders of model B are mostly consistent with our preliminary identification with the exception of $p = 0$, $q = 0$, and $P = 0$. Orders of model C are also mostly consistent with our preliminary identification with the exception of $p = 0$, $q = 2$, and $P = 0$.

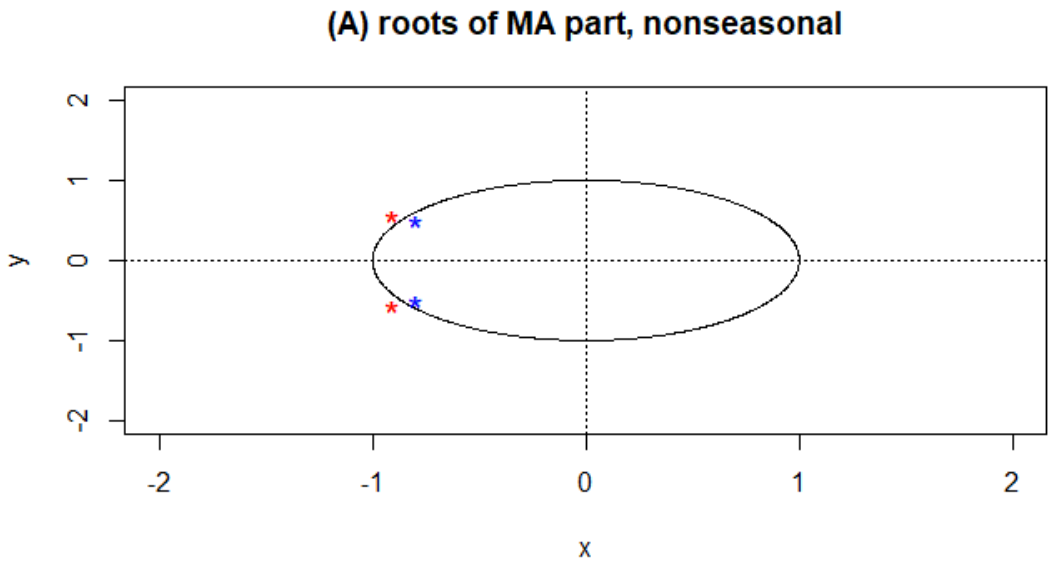
4.6 Check of Stationarity and Invertibility of Fitted Models

To check if our models are stationary and invertible, we just need to check if the AR part is stationary and the if the MA part is invertible since the AR part is always invertible and the MA part is always stationary.

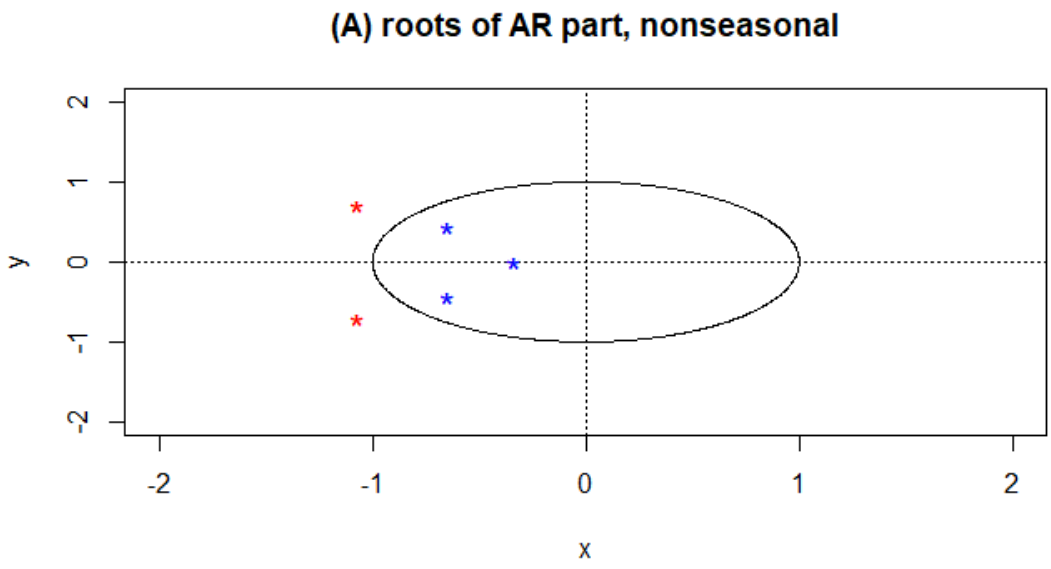
4.6.1 Model A:

Since $|\Phi_1| = |-0.7037| = 0.7037 < 1$, the seasonal MA part is invertible. Then we check if the nonseasonal MA part is invertible and if the nonseasonal AR part is stationary by examining the plot of roots.

```
plot.roots(NULL,polyroot(c(1,1.5901 , 0.8805)),main="(A) roots of MA pa
rt, nonseasonal")
```




```
plot.roots(NULL,polyroot(c(1,1.6306,1.0406,0.2028)),main="(A) roots of
AR part, nonseasonal")
```



Since all roots (in red) are outside of the unit circle, our model A is invertible and stationary.

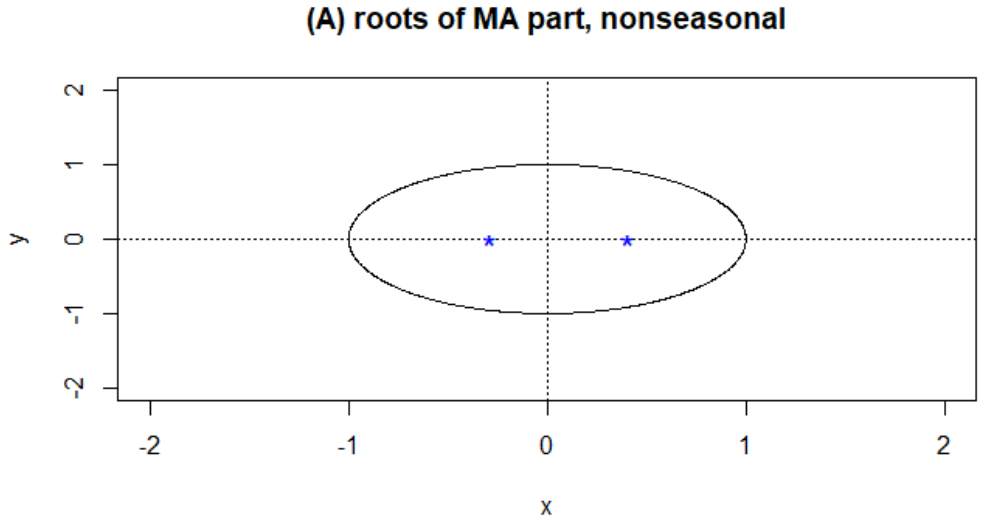
4.6.2 Model B

Model B is always stationary since it is a pure SMA model. And it is invertible since $|\Theta_1| = |-0.7307| = 0.7307 < 1$.

4.6.3 Model C

The seasonal MA part is invertible since $|\Theta_1| = |-0.7193| = 0.7193 < 1$.

```
plot.roots(NULL,polyroot(c(1,-0.1243, -0.1170)),main="(A) roots of MA p
art, nonseasonal")
```



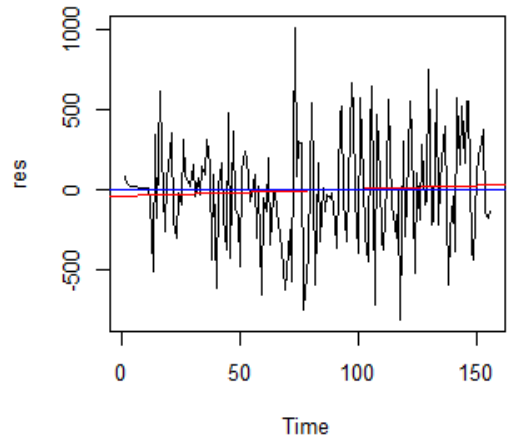
Since all roots (in red) are outside of the unit circle, our model C is invertible and stationary.

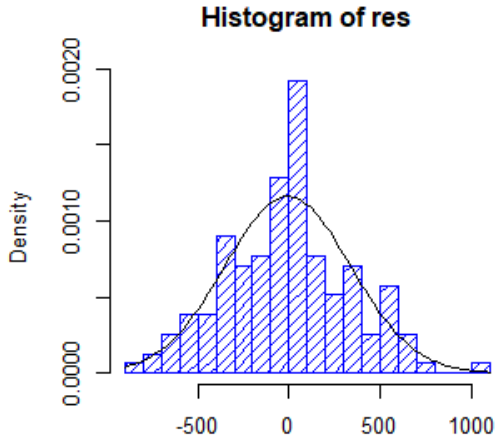
5. Diagnostic Checking

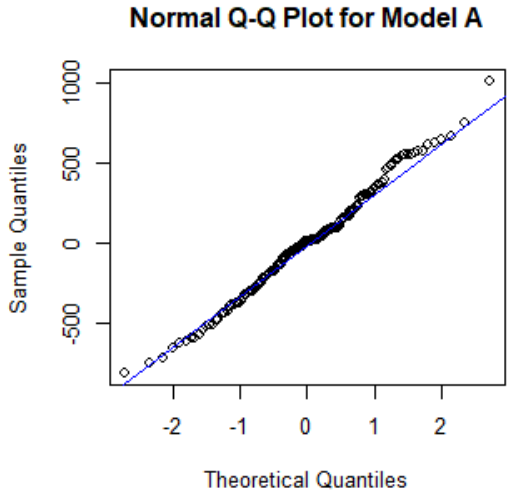
Since all of the three models are invertible and stationary, we proceed to diagnostic checking for residuals. If the model is a good fit, then the residuals should resemble Gaussian white noise.

5.1 Model A

```
fit.A <- arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.A)
```

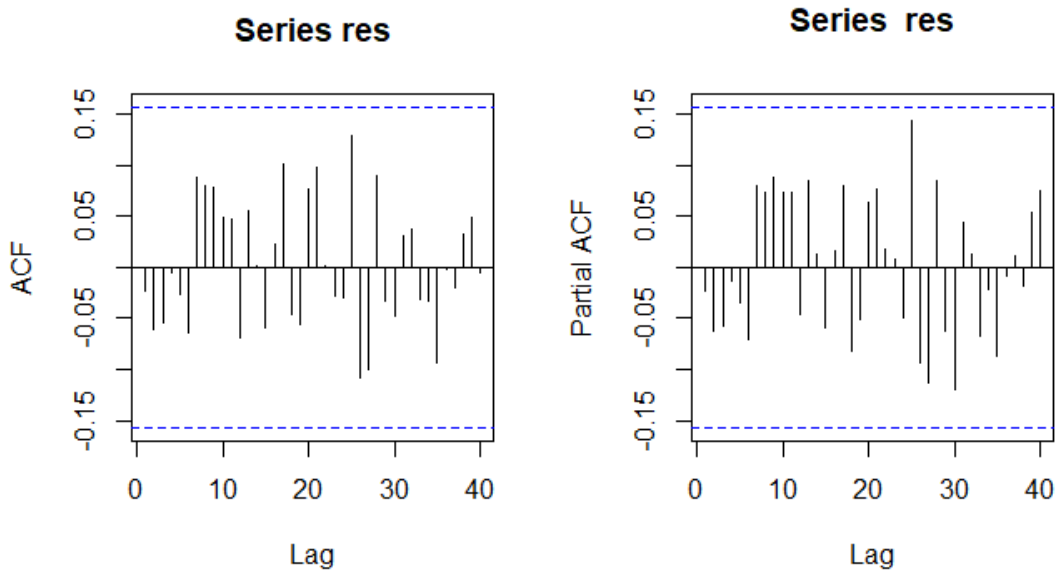






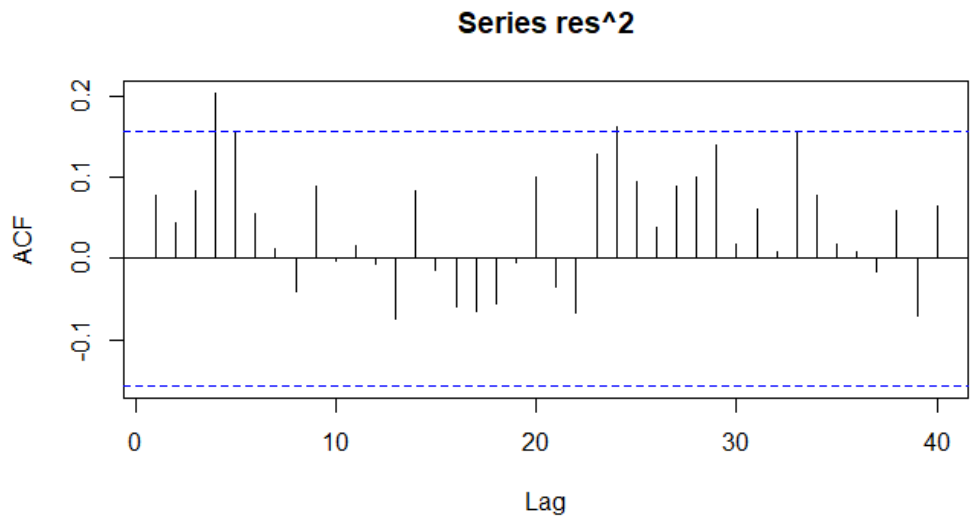
The time series plot of the residuals for model A shows no trend, no visible change of variance, and no seasonality. The histogram approximately resembles a Gaussian

distribution. And in the normal Q-Q plot we can see that the data points form a roughly straight line. We will confirm the normality of the residuals by Shapiro-Wilk test.



All ACFs and PACFs of the residuals for model A are within the confidence intervals and hence can be counted as zeros.

```
shapiro.test(res)
##  Shapiro-Wilk normality test
## data:  res
## W = 0.99257, p-value = 0.5999

Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 5)
##  Box-Pierce test
## data:  res
## X-squared = 6.5281, df = 7, p-value = 0.4796

Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf =5)
##  Box-Ljung test
## data:  res
## X-squared = 6.9463, df = 7, p-value = 0.4345

Box.test(res^2 , lag = 12, type =c("Ljung-Box"), fitdf = 0)
##  Box-Ljung test
## data:  res^2
## X-squared = 15.231, df = 12, p-value = 0.229

acf(res^2, lag.max =40)
```



All p-values of the shapiro test for normality, Box-Pierce test, Ljune-Box test, and McLeod-Li Test are greater than 0.05. So the normality assumption holds. Even though the ACF at lags 4 is outside the confidence interval as shown in the plot of ACFs of the square of residuals, the *p*-value of Mcleod-Li test is large. So we can conclude that there is no non-linear dependence between the residuals and hence the white noise hypothesis holds.

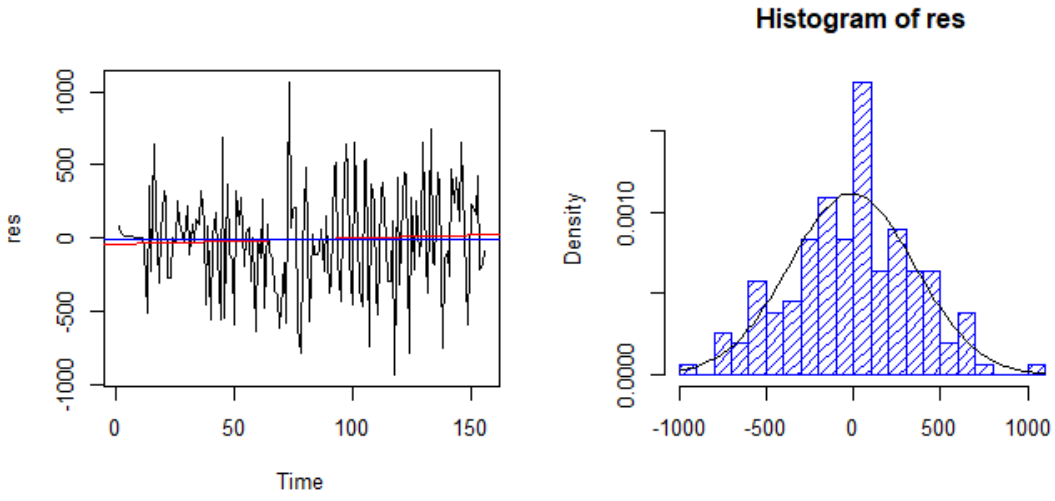
```
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Order selected 0  sigma^2 estimated as  118232
```

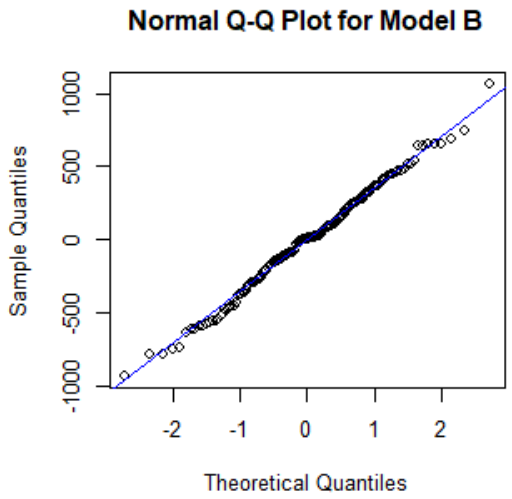
The fitted model for residuals is $AR(0)$, i.e. white noise.

Thus, our model A passed the diagnostic checking and is ready to be used for forecasting.

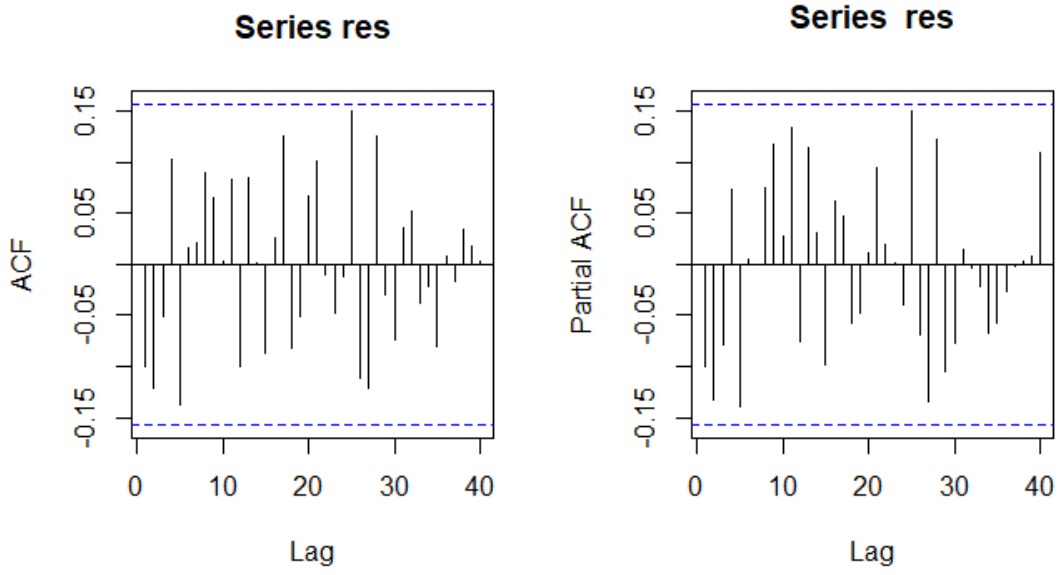
5.2 Model B

```
fit.B <- arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.B)
```





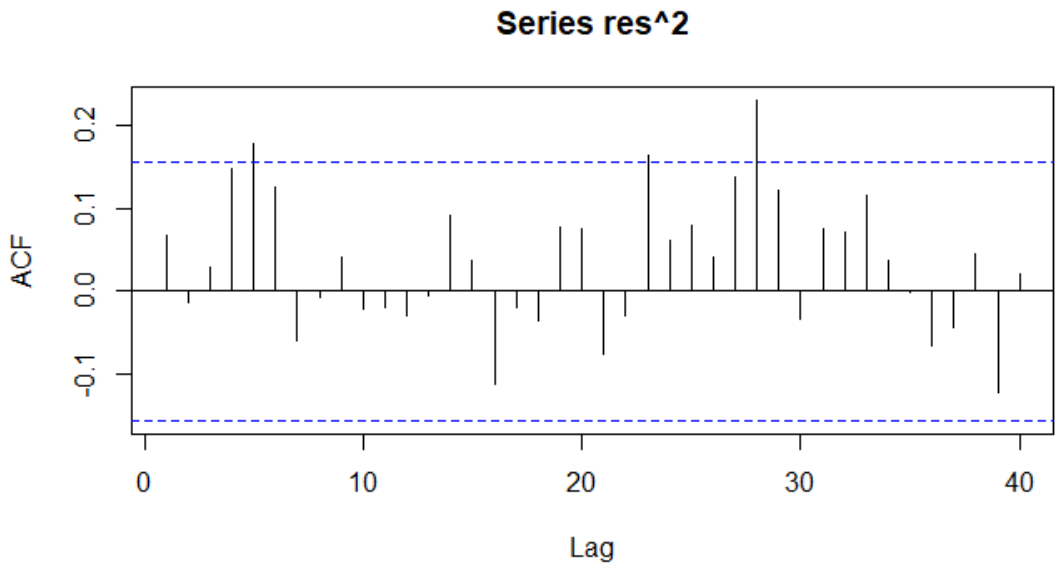
The time series plot of the residuals for model B shows no trend, no visible change of variance, and no seasonality. The histogram approximately resembles a Gaussian distribution. And in the normal Q-Q plot we can see that the data points form a roughly straight line. We will confirm the normality of the residuals by Shapiro-Wilk test.



All ACFs and PACFs of the residuals for model B are within the confidence intervals and hence can be counted as zeros.

```
shapiro.test(res)
##  Shapiro-Wilk normality test
## data:  res
## W = 0.99446, p-value = 0.8211
```

```r
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 0)
##  Box-Pierce test
## data:  res
## X-squared = 13.548, df = 12, p-value = 0.3305

Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf =0)
##  Box-Ljung test
## data:  res
## X-squared = 14.242, df = 12, p-value = 0.2855

Box.test(res^2 , lag = 12, type =c("Ljung-Box"), fitdf = 0)
##  Box-Ljung test
## data:  res^2
## X-squared = 13.419, df = 12, p-value = 0.3393

acf(res^2, lag.max =40)
```



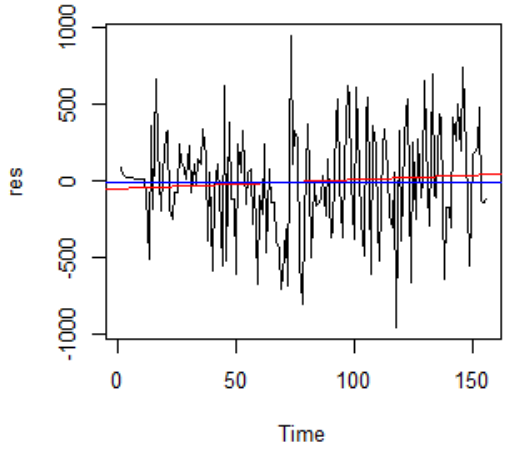
All p-values of the shapiro test for normality, Box-Pierce test, Ljune-Box test, and McLeod-Li Test are greater than 0.05. So the normality assumption holds. Even though the ACFs at lags 5 and 28 are outside the confidence interval as shown in the plot of ACFs of the square of residuals, the p-value of Mcleod-Li test is large. So we can conclude that there is no non-linear dependence between the residuals and hence the white noise hypothesis holds.

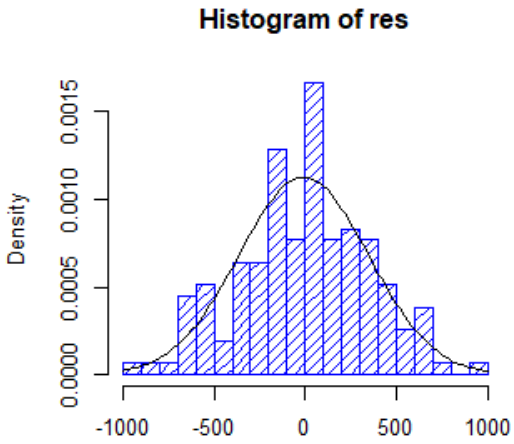
```r
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Coefficients:
##       1        2
## -0.1135  -0.1331
## Order selected 2  sigma^2 estimated as  126770
```

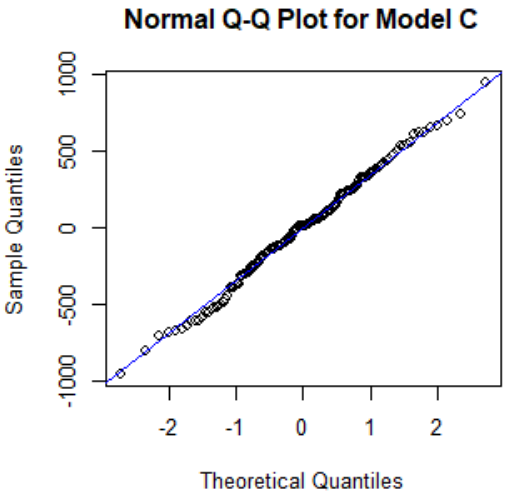
However, the fitted model for residuals is $AR(2)$ rather than the white noise. Thus, our model B does not pass the diagnostic checking. We could improve model B, but since we have already had a model A which passed the diagnostic checking, we will not improve B here.

5.3 Model C:

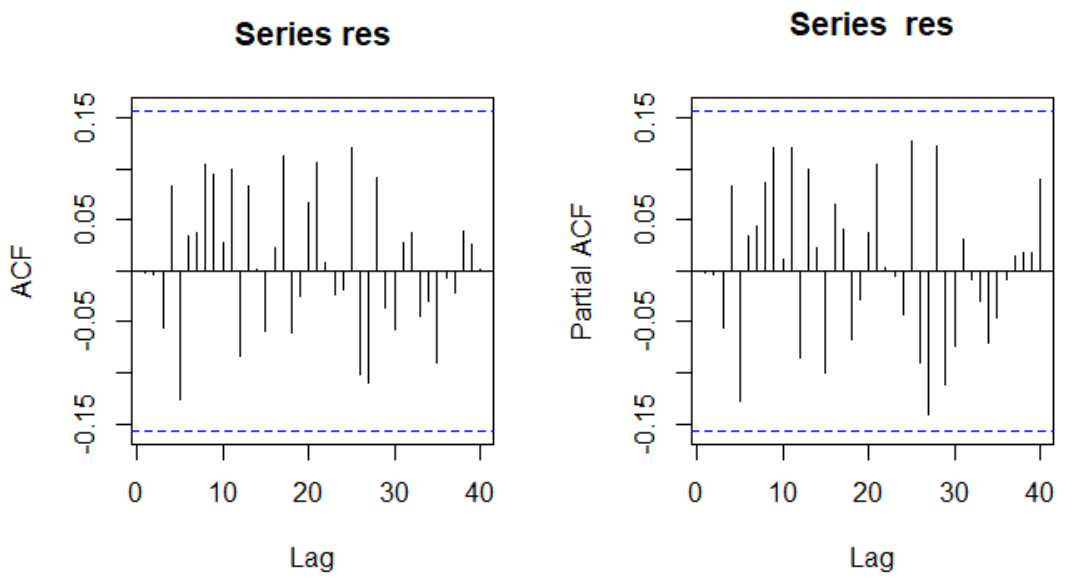
```
fit.C <- arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.C)
```





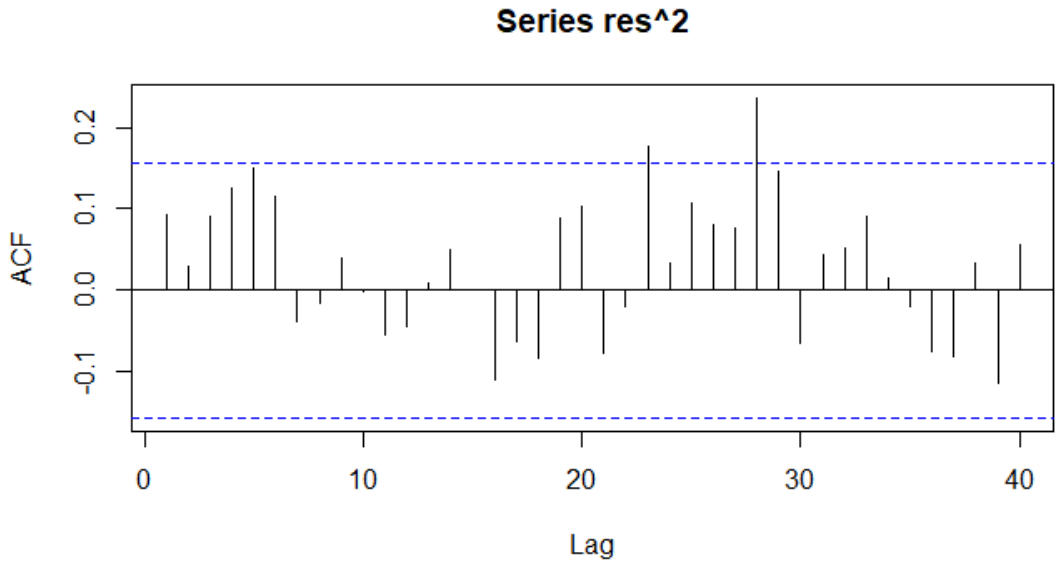


The time series plot of the residuals for model C shows no trend, no visible change of variance, and no seasonality. The histogram approximately resembles a Gaussian distribution. And in the normal Q-Q plot we can see that the data points form a roughly straight line. We will confirm the normality of the residuals by Shapiro-Wilk test.



All ACFs and PACFs of the residuals for model C are within the confidence intervals and hence can be counted as zeros.

```
shapiro.test(res)
##  Shapiro-Wilk normality test
## data:  res
## W = 0.9948, p-value = 0.8562

Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 2)
##  Box-Pierce test
## data:  res
## X-squared = 10.333, df = 10, p-value = 0.4118

Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf =2)
##  Box-Ljung test
## data:  res
## X-squared = 11.005, df = 10, p-value = 0.3571

Box.test(res^2 , lag = 12, type =c("Ljung-Box"), fitdf = 0)
##  Box-Ljung test
## data:  res^2
## X-squared = 12.611, df = 12, p-value = 0.3979

acf(res^2, lag.max =40)
```



All p-values of the shapiro test for normality, Box-Pierce test, Ljune-Box test, and McLeod-Li Test are greater than 0.05. So the normality assumption holds. Even though the ACFs at lags 23 and 28 are outside the confidence interval as shown in the plot of ACFs of the square of residuals, the p-value of Mcleod-Li test is large. So we can conclude that there is no non-linear dependence between the residuals and hence the white noise hypothesis holds.

```
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Call:
## ar(x = res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
## Order selected 0  sigma^2 estimated as  125372
```

The fitted model for residuals is $AR(0)$, i.e. white noise.

Thus, our model C passed the diagnostic checking and is ready to be used for forecasting next.

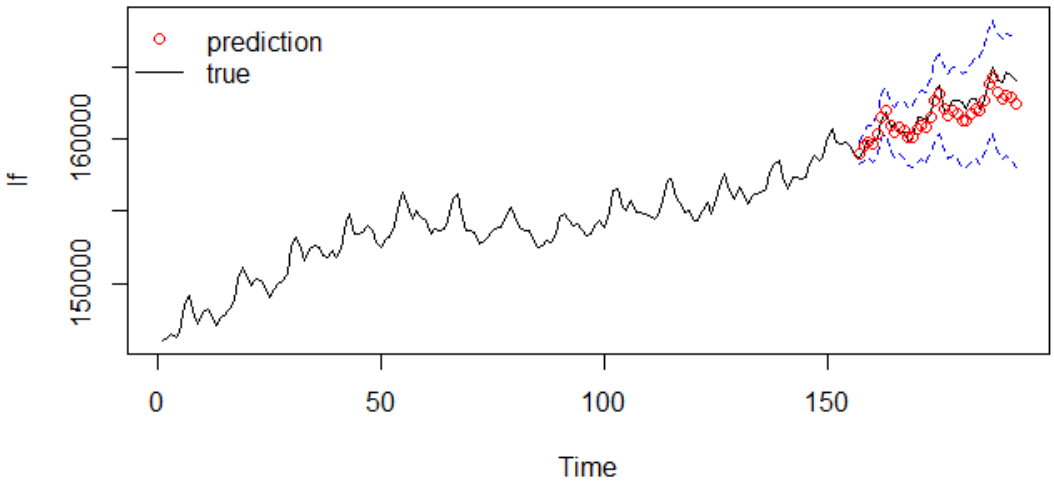
5.4 Choosing the Final Model

Both model A and model C passed diagnostic checking and are appropriate to use for forecasting. The AICc for model A is 2110.242 and the AICc for model C is 2111.014. The difference between the AICc of the two models is not large, but model C has significantly fewer number of parameters. Thus, by using the principle of parsimony, we choose model C, SARIMA$(0,1,2) \times (0,1,1)_{12}$ with model equation $\nabla_1\nabla_{12}X_t = (1 - 0.1243_{(0.0828)}B - 0.1170_{(0.0758)}B^2)(1 - 0.7193_{(0.0684)}B^{12})Z_t$, $\hat{\sigma}_Z^2 = 134048$, as the final model.

6. Forcasting

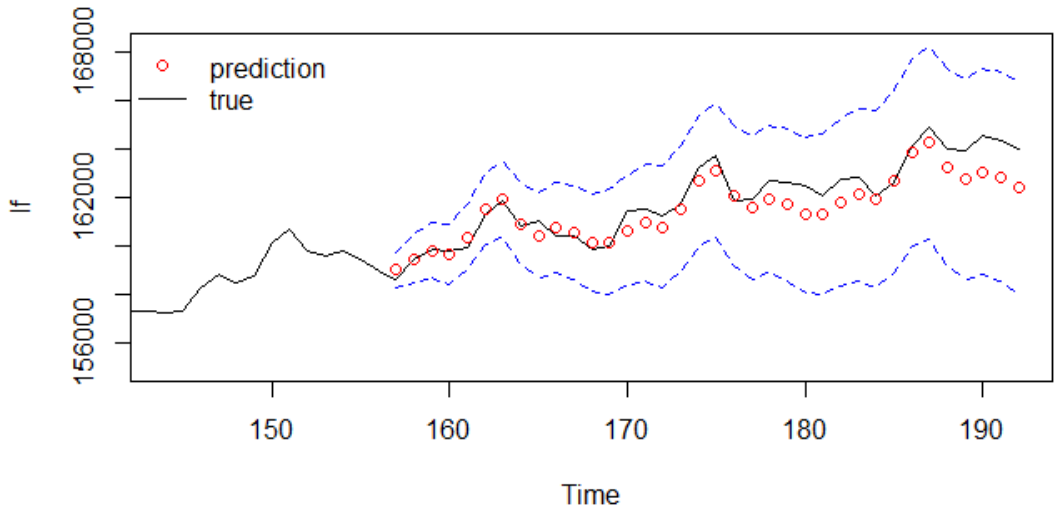
Using model C, we forecast the civilian labor force level from January 2017 to December 2019 and plot our predictions in the plot along with the original data.

```
pred.trC <- predict(fit.C, n.ahead = 36)
U.trC = pred.trC$pred + 2*pred.trC$se # upper bound of prediction inter
val
L.trC = pred.trC$pred - 2*pred.trC$se # Lower bound
```



The black line represents the true values of the civilian labor force level from January 2004 to December 2019. The red circles represent our predictions for 2017-2019, and the blue dashed lines represent the confidence interval for the predictions. We can see that our predictions are consistent with the trend and seasonality of the original data. And the true values from the test data are inside the prediction intervals.

Next, we zoom in the plot to see clearly how well our predictions capture the true values.



We can see that predictions for year 2017 using our model are very close to the true values from the test data, since the predictions (red circles) for year 2017 roughly lie on the black line for original data. Our predictions for year 2018 and 2019 are slightly lower than the true values in general. This deviation might be attributed to the Tax Cuts and Jobs Act that went into effect on January 1, 2018. Note that the civilian labor forces excludes unemployed people who "would like to work but have given up due to lack of opportunities, an injury or illness." [2] The Tax Cuts are designed to "drive up labor demand, directly helping historically disadvantaged people through increased economic activity." [4] The historically disadvantaged people might have been excluded from the civilian labor force since they did not actively find a job. However, increasing labor demand and lower marginal tax rates induced by the Tax Cuts might stimulate those people to find jobs. So the inclusion of the historically disadvantaged people might be the reason why civilian labor force level in 2018 and 2019 grows higher than our prediction based on previous years' levels.

Conclusion

In conclusion, we find that the civilian labor force level increases in general over the time period from January 2004 to December 2019. It displays a yearly pattern of being lowest at the beginning of a year and reaching the highest level in the middle of a year. Our final model is SARIMA$(0,1,2) \times (0,1,1)_{12}$ with model equation $\nabla_1 \nabla_{12} X_t = (1 - 0.1243_{(0.0828)} B - 0.1170_{(0.0758)} B^2)(1 - 0.7193_{(0.0684)} B^{12}) Z_t, \hat{\sigma}_Z^2 = 134048$. It captures the trend and seasonality of the original data well and gives reliable forecasts of the test data. All of the true values from the test data of year 2017 to 2019 fall inside the prediction interval. Predictions of the monthly civilian labor force level in 2017 are relatively more accurate than predictions of the monthly civilian labor force level in 2018 and 2019. And the deviation might have been caused by implementation of Tax Cuts.

I would like to express my very great appreciation to Dr. Feldman for answering my questions regarding this report and providing valuable and constructive suggestions.

References

[1] "Labor Force Statistics from the Current Population Survey," U.S. Bureau of Labor Statistics, 10 December 2020. [Online]. Available: https://data.bls.gov/pdq/SurveyOutputServlet. [Accessed 10 December 2020].

[2] C. Halton, "Civilian Labor Force," 23 July 2018. [Online]. Available: https://www.investopedia.com/terms/c/civilian-labor-force.asp. [Accessed 11 December 2020].

[3] M. Toossi, "Labor force projections to 2022: the labor force participation rate continues to fall," December 2013. [Online]. Available: https://www.bls.gov/opub/mlr/2013/article/labor-force-projections-to-2022-the-labor-force-participation-rate-continues-to-fall.htm. [Accessed 11 December 2020].

[4] T. C. o. E. Advisers, "The Impact of the Trump Labor Market on Historically Disadvantaged Americans," December 2019. [Online]. Available: https://www.whitehouse.gov/wp-content/uploads/2019/12/The-Impact-of-the-Trump-Labor-Market-on-Historically-Disadvantaged-Americans.pdf. [Accessed 11 December 2020].

Appendix

```
library(latex2exp)
library(tidyr)
library(tidyverse)

# Import dataset
laborForce =read.table('CivilianLaborForce.txt',header=TRUE,sep=',')
head(laborForce)

# Reshape the dataset
laborForce=laborForce%>%
  gather(Month,data,Jan:Dec) %>%
  arrange(Year)
head(laborForce)
summary(laborForce)
# Partition dataset to two parts for model training and model validatio
n; work with training set:
lf=laborForce$data
i=length(lf)
lf.train = lf[c(1:(i-36))] # training dataset X_t
lf.test = lf[c((i-35):i)] # test dataset
# To plot training data with years on x-axis:
laborForce.ts = ts(lf.train,start =c(2004,1),frequency = 12)
ts.plot(laborForce.ts, main="Civilian Labor Force Training Data with ye
ars on x-axis")

plot.ts(lf.train, main=TeX('Time series $X_t$: Civilian Labor Force Tra
ining Data'))
fit = lm(lf.train~as.numeric(1:length(lf.train)))
abline(fit,col='blue') # Add trend to the plot
abline(h=mean(lf.train),col='red') # Add mean to the plot
op <- par(mfrow=c(1,2))
acf(lf.train,lag.max=40,main="")
pacf(lf.train,lag.max=40,main="")
title("P/ACF of the Civilian Labor Force Training Data", line = -1, out
er = TRUE)
par(op)
hist(lf.train,main="histogram: Civilian Labor Force Training Data",dens
ity = 20,breaks=20,prob=TRUE, col="light blue")
m =mean(lf.train)
std = sqrt(var(lf.train))
curve(dnorm(x,m,std), add=TRUE)

library(MASS)
t = 1:length(lf.train)
bcTransform = boxcox(lf.train ~ t,lambda = seq(-10,5,1/10),plotit = TRU
E)
lambda = bcTransform$x[which(bcTransform$y==max(bcTransform$y))]
```

```r
lft.bc = (1/lambda)*(lf.train^lambda -1)

op <- par(mfrow=c(2,2))
hist(lf.train,main=TeX('histogram: Original Time series $X_t$'),density
 = 20,breaks=20,prob=TRUE, col="light blue",)
m =mean(lf.train)
std = sqrt(var(lf.train))
curve(dnorm(x,m,std), add=TRUE)

hist(lft.bc,main=TeX('histogram: Box-Cox Transformed $X_t$'),density =
20,breaks=20,prob=TRUE, col="light blue",)
m =mean(lft.bc)
std = sqrt(var(lft.bc))
curve(dnorm(x,m,std), add=TRUE)

lft.log=log(lf.train)
hist(lft.log,main=TeX("histogram: Log Transformed $X_t$"),density = 20,
breaks=20,prob=TRUE, col="light blue",)
m =mean(lft.log)
std = sqrt(var(lft.log))
curve(dnorm(x,m,std), add=TRUE)

lft.sqrt=sqrt(lf.train)
hist(lft.sqrt,main=TeX("histogram: Squareroot Transformed $X_t$"),densi
ty = 20,breaks=20,prob=TRUE, col="light blue",)
m =mean(lft.sqrt)
std = sqrt(var(lft.sqrt))
curve(dnorm(x,m,std), add=TRUE)
par(op)

op <- par(mfrow=c(2,2))
plot.ts(lf.train, main=TeX('Original Time series $X_t$'))
plot.ts(lft.bc, main=TeX('Box-Cox Transformed Time series $X_t$'))
plot.ts(lft.log, main=TeX('Log Transformed Time series $X_t$'))
plot.ts(lft.sqrt, main=TeX("Squareroot Transformed Time Series $X_t$"))
par(op)

# produce decomposition of X_t:
library(ggplot2)
library(ggfortify)
y <- ts(as.ts(lf.train),frequency = 12)
plot(decompose(y))

require(TSA)
periodogram(lf.train); abline(h=0)
axis(1,at=c(0.08,0.16,0.25)) # Confirm that period=12
lft_12 = diff(lf.train,12)
var(lf.train)
var(lft_12)
```

```r
op <- par(mfrow=c(2,1))
ts.plot(lft_12, main=TeX("$X_t$: differenced at lags 12"))
abline(lm(lft_12~as.numeric(1:length(lft_12))),col='blue') # add trend
abline(h=mean(lft_12),col='red') #add mean line
acf(lft_12,lag.max = 40, main=TeX("ACF of X_t, differenced at lags 12")
)
par(op)
lft.stat = diff(lft_12,1)
# Check variance
var(lft_12)
var(lft.stat)

lft2 = diff(lft.stat,1) # difference at lags 1 again
# Check variance
var(lft2)

op<- par(mfrow=c(2,1))
ts.plot(lft.stat, main=TeX("$X_t$: differenced at lags 12 and 1"))
abline(lm(lft.stat~as.numeric(1:length(lft.stat))),col='blue') # add  t
rend
abline(h=mean(lft.stat),col='red') #add mean line
acf(lft.stat,lag.max = 50, main=TeX("ACF of X_t, differenced at lags 12
 and 1"))
par(op)
op <- par(mfrow=c(1,2))
hist(lf.train,main=TeX("histogram of $X_t$"),density = 20,breaks=20,pro
b=TRUE, col="light blue",)
m =mean(lf.train)
std = sqrt(var(lf.train))
curve(dnorm(x,m,std), add=TRUE)

hist(lft.stat, main=TeX("histogram of $\\nabla_1\\nabla_{12}X_t$"),dens
ity = 20,breaks=20,prob=TRUE, col="light blue")
m =mean(lft.stat)
std = sqrt(var(lft.stat))
curve(dnorm(x,m,std), add=TRUE)
par(op)
op <- par(mfrow=c(2,1))
acf(lft.stat,lag.max = 50, main="")
pacf(lft.stat,lag.max = 50, main="")
title(TeX("P/ACF of $X_t$: differenced at lags 12 and 1"), line = -0.5,
 outer = TRUE)
par(op)
library(qpcR)
# SMA models tried
for(q in 1:5){
  print(paste("Q,q:",1,q," ", AICc(arima(lf.train,order=c(0,1,q),season
al=list(order=c(0,1,1),period=12),method='ML'))
  ))}
#model producing the lowest AICc
```

```r
SMA1 = arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA1
AICc(SMA1) #lowest

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA1$coef-1.96*sqrt(diag(SMA1$var.coef)),SMA1$coef+1.96*sqrt
(diag(SMA1$var.coef)) )
CI
# Set ma1 to zero
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA,NA), method="ML"))
# Set ma2 to zero
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(NA,0,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(NA,0,NA), method="ML"))
SMA2 = arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA2
AICc(SMA2)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA2$coef-1.96*sqrt(diag(SMA2$var.coef)),SMA2$coef+1.96*sqrt
(diag(SMA2$var.coef)) )
CI
# Set insignificant parameters to zero
arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,1), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA), method="ML"))
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")
AICc(arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1),
period = 12), method="ML"))
SMA3 = arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1)
, period = 12), method="ML")
SMA3
AICc(SMA3)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SMA3$coef-1.96*sqrt(diag(SMA3$var.coef)),SMA3$coef+1.96*sqrt
(diag(SMA3$var.coef)) )
CI
# Set ma1 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,NA,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
```

```
period = 12),fixed=c(0,NA,NA,NA), method="ML"))
# Set ma2 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,0,NA,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,0,NA,NA), method="ML"))
# Set ma3 to zero
arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1), perio
d = 12),fixed=c(0,NA,0,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,3), seasonal = list(order = c(0,1,1),
period = 12),fixed=c(0,NA,0,NA), method="ML"))
SMA3 = arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1)
, period = 12),fixed=c(0,NA,NA), method="ML")
SMA3
AICc(SMA3)
# SAR models tried: P=1,2,3; p=1,2,3,4,5.
for(P in 1:3)
  for(p in 1:5){
    aicc = try(AICc(arima(lf.train,order=c(p,1,0),seasonal=list(order=c
(P,1,0),period=12),method='ML')))
    if (inherits(aicc,"try-error")){
    # skip this iteration if error happens
    next
  }
  print(paste("P,p:",P,p," ", aicc))}
SAR = arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0),
 period = 12), method="ML")
SAR
AICc(SAR)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SAR$coef-1.96*sqrt(diag(SAR$var.coef)),SAR$coef+1.96*sqrt(di
ag(SAR$var.coef)) )
CI
# Set insignificant parameters to zero
arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0), perio
d = 12),fixed=c(0,NA,NA,NA,NA), method="ML")
AICc(arima(lf.train, order=c(2,1,0), seasonal = list(order = c(3,1,0),
period = 12), fixed=c(0,NA,NA,NA,NA),method="ML"))
MA=arima(lf.train, order=c(0,0,17),method = "ML")
MA
AICc(MA)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(MA$coef-1.96*sqrt(diag(MA$var.coef)),MA$coef+1.96*sqrt(diag(
MA$var.coef)) )
CI
for(P in 1:3)
  for(q in c(0,2)){
  print(paste("P,p,Q,q:",P,0,1,q," ",
```

```
  AICc(arima(lf.train,order=c(0,1,q),seasonal=list(order=c(P,1,1),perio
d=12),method='ML'))))}
SARIMA1 = arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1
,1), period = 12), method="ML")
SARIMA1
AICc(SARIMA1)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SARIMA1$coef-1.96*sqrt(diag(SARIMA1$var.coef)),SARIMA1$coef+
1.96*sqrt(diag(SARIMA1$var.coef)) )
CI
# Set insignificant parameters to zero
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1,1), perio
d = 12),fixed=c(0,NA), method="ML")
AICc(arima(lf.train, order=c(0,1,0), seasonal = list(order = c(1,1,1),
period = 12), fixed=c(0,NA),method="ML"))
for(p in 1:5)
  for(q in c(0,2)){
  print(paste("P,p,Q,q:",0,p,1,q," ",
  AICc(arima(lf.train,order=c(p,1,q),seasonal=list(order=c(0,1,1),perio
d=12),method='ML'))))}
SARIMA2 = arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1
,1), period = 12), method="ML")
SARIMA2
AICc(SARIMA2)

# Calculate approximate 95% confidence interval for parameters
CI = cbind(SARIMA2$coef-1.96*sqrt(diag(SARIMA2$var.coef)),SARIMA2$coef+
1.96*sqrt(diag(SARIMA2$var.coef)) )
CI
# Model A
arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")

# Model B
arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")

# Model C
arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,1), perio
d = 12), method="ML")
source('plot.roots.R')
# Check invertibility of MA part of model A
plot.roots(NULL,polyroot(c(1,1.5901 , 0.8805)),main="(A) roots of MA pa
rt, nonseasonal")

# Check stationarity of AR part of model A
plot.roots(NULL,polyroot(c(1,1.6306,1.0406,0.2028)),main="(A) roots of
AR part, nonseasonal")
```

```r
# Check invertibility of MA part of model C
plot.roots(NULL,polyroot(c(1,-0.1243, -0.1170)),main="(A) roots of MA p
art, nonseasonal")
# Diagnostic checking for model A
fit.A <- arima(lf.train, order=c(3,1,2), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.A)
op <- par(mfrow=c(2,2))
plot.ts(res)
fitt.A =lm(res ~ as.numeric (1:length(res))); abline(fitt.A, col="red")
abline(h=mean(res), col="blue")
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 A")
qqline(res,col ="blue")
par(op)
op <- par(mfrow=c(1,2))
acf(res, lag.max =40)
pacf(res, lag.max =40)
par(op)
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 5)
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf =5)
Box.test(res^2 , lag = 12, type =c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max =40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# Diagnostic checking for model B
fit.B <- arima(lf.train, order=c(0,1,0), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.B)
op <- par(mfrow=c(2,2))
plot.ts(res)
fitt.B =lm(res ~ as.numeric (1:length(res))); abline(fitt.B, col="red")
abline(h=mean(res), col="blue")
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(op)
op <- par(mfrow=c(1,2))
acf(res, lag.max =40)
pacf(res, lag.max =40)
par(op)
shapiro.test(res)
Box.test(res, lag = 12, type = c("Box-Pierce"), fitdf = 0)
```

```r
Box.test(res, lag = 12, type = c("Ljung-Box"), fitdf =0)
Box.test(res^2 , lag = 12, type =c("Ljung-Box"), fitdf = 0)
acf(res^2, lag.max =40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# Diagnostic checking for model C
fit.C <- arima(lf.train, order=c(0,1,2), seasonal = list(order = c(0,1,
1), period = 12), method="ML")
res = residuals(fit.C)
op <- par(mfrow=c(2,2))
plot.ts(res)
fitt.C =lm(res ~ as.numeric (1:length(res))); abline(fitt.C, col="red")
abline(h=mean(res), col="blue")
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 C")
qqline(res,col ="blue")
par(op)
op <- par(mfrow=c(1,2))
acf(res, lag.max =40)
pacf(res, lag.max =40)
par(op)
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)
acf(res^2, lag.max =40)
ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# Forecasting using model C:
pred.trC <- predict(fit.C, n.ahead = 36)
U.trC = pred.trC$pred + 2*pred.trC$se # upper bound of prediction inter
val
L.trC = pred.trC$pred - 2*pred.trC$se # lower bound

ts.plot(lf,ylim=c(min(lf),max(U.trC)))
lines(U.trC, col='blue', lty='dashed')
lines(L.trC, col='blue', lty='dashed')
points((length(lf.train)+1):(length(lf.train)+36),pred.trC$pred,col='re
d')
legend('topleft',bty='n', col=c('red','black'),pch=c(1,NA),lty=c(NA,1),
c('prediction','true'))
ts.plot(lf,xlim=c(144, length(lf.train)+36),ylim=c(155000,max(U.trC)))
lines(U.trC, col='blue', lty='dashed')
lines(L.trC, col='blue', lty='dashed')
points((length(lf.train)+1):(length(lf.train)+36),pred.trC$pred,col='re
d')
#points((length(lf.train)+1):(length(lf.train)+36),lf.test,col='black')
legend('topleft',bty='n', col=c('red','black'),pch=c(1,NA),lty=c(NA,1),
c('prediction','true'))
```