Time Series Report

on Non-farm Payroll Employment

Anne Lin

5/10/2021

Abstract

The project aims to find a time series model for the total number of Non-farm Payroll Employment of the United States from 01/01/2010 to 01/01/2019 for future prediction. The main techniques used in this project include data transformation (Box-cox transformation, log-transformation and square root transformation), differencing, analysis of sample ACF and PACF, diagnostic checking of models (Shapiro-Wilk normality test, Box-Pierce test, Box-Ljung test, Mc-Leod Li test and Yule-Walker test), and model prediction. After the analysis, we worked out a satisfactory model which predcited the data from 01/01/2019 to 01/01/2020 well.

Introduction

Employment is related to the economy situation. One crucial element of interpreting the employment is the total number of non-farm employment, which is a measure of the number of U.S. workers in the economy that excludes proprietors, private household employees, unpaid volunteers, farm employees, and the unincorporated self-employed. This accounts for approximately 80% of the workers who contribute to GDP, so analyzing this measure is beneficial. If we can forecast increases in the number of non-farm employment, it indicates that businesses will hiring more people, which also suggest businesses growth in the future.

Here we will analyze the monthly non-farm employment from 01/01/2010 to 01/01/2020, with 108 observations from 01/01/2010 to 12/01/2018 as training data for building the time series model, and 13 observations from 01/01/2019 to 01/01/2020 as tests for our model. The data set is from Federal Reserve Economic Data, which is monthly number of Nonfarm Payroll Employment from 1949 to 2021. All processes are done through RStudio.

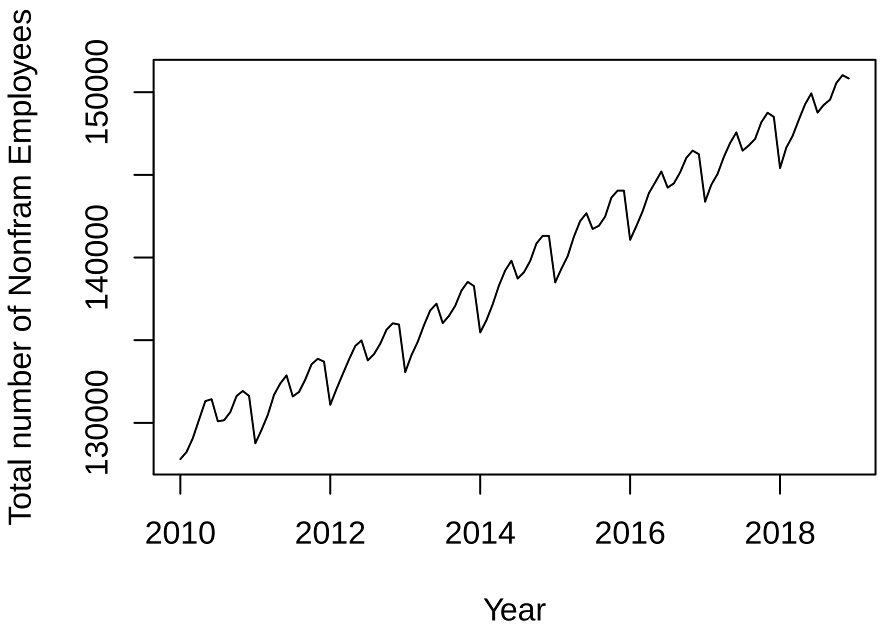
The main objective of this project is to **find a time series model that best fits the data and to use the model to perform forecasting**. The techniques used include data **pre-processing** (*to make the dataset analyzable*), **transforming and differencing** (*to make the data stationary*), **interpreting from sample ACF and PACF** (*to identify parameters for the model*), **diagnostic checking** (*to check if the distribution of the residuals of models resembles white noise*), and **predicting** the number of non-farm employment in the condition that no special crises happened, which can be used to compare with the result in the COVID-19 situation to see the impact of the pandemic.

The result is that the SARIMA$(6,1,0)\times(0,1,1)_{12}$ model best fits our data and performed well in forecasting, and the explicit algebraic form of the model is $(1-0.2070_{0.1045}B^5-0.4179_{0.1125}B^6)(1-B)(1-B^{12})\ln(X_t) = (1-0.5499_{0.1190}B^{12})Z_t$.

Sections

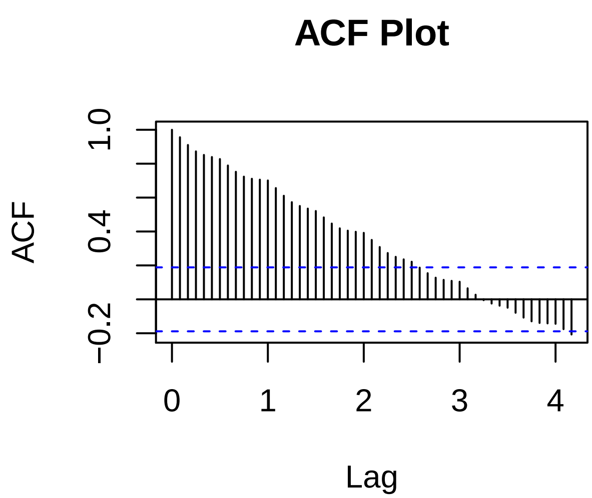
Loading dataset

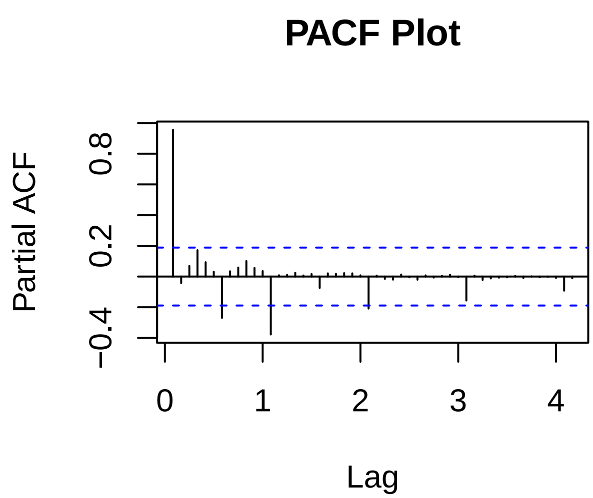
We first need to import the dataset, selecting data we wanted to analyze and plot the time series data to have the first and general look. The plot is as follows:



From the plot we can see a apparent **upward and seasonal trend, but no clear change in variance.** Also, there is no apparent sharp increase or decrease in labor force participation rate. The frequency for the seasonality is around 1 year because between 2 years there appeared to have 2 similar "M"-shaped number of employment change.

To more intuitively see the period for seasonality, we plot the ACF and PACF of original time series.





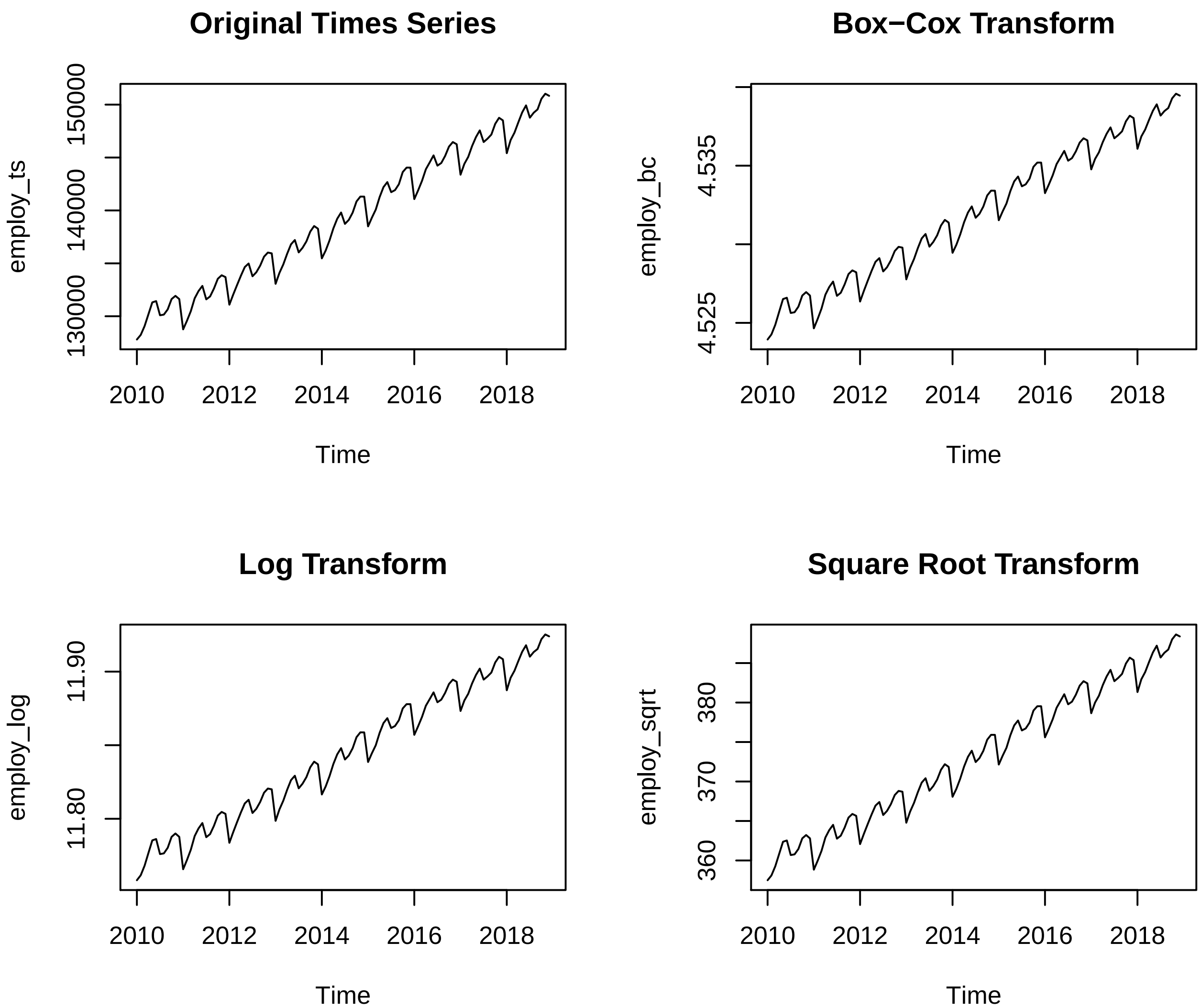
The slowly decay of ACF shows seasonality, and from the PACF plot, the period for seasonality is around 1 year, which should be 12 months.

Transformation and differencing

Transformation

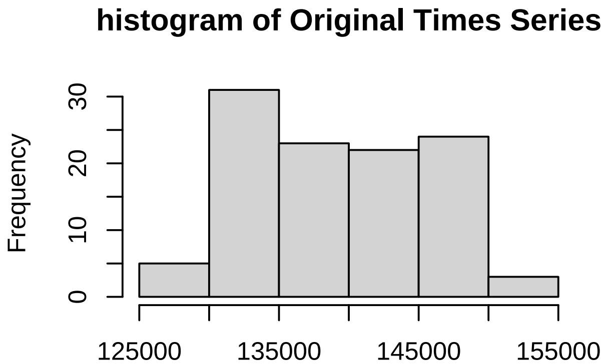
Since there is no clear change in variance, we firstly assume that we don't have to do transformations on the data (such as Box-Cox transformation, log transformation or square root transformation.)

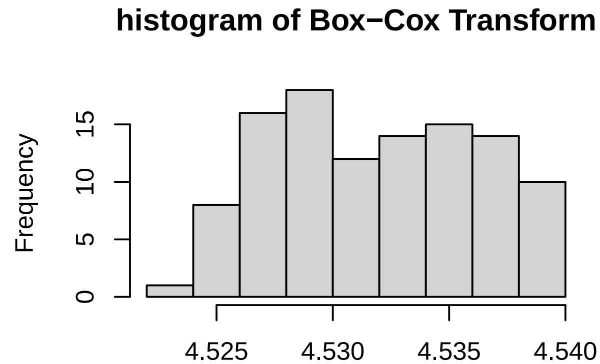
However, to check if our assumption is correct, we firstly plot the original data with the transformed data to see the disctribution of original data and transformed data.

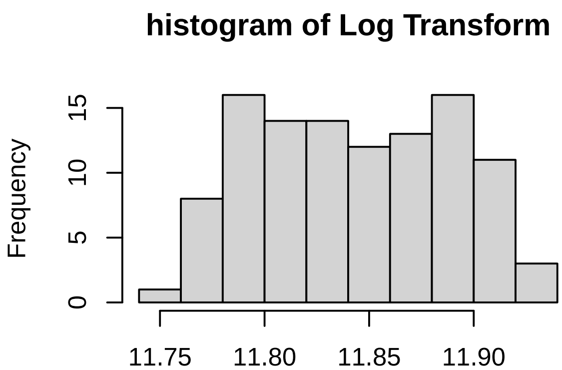


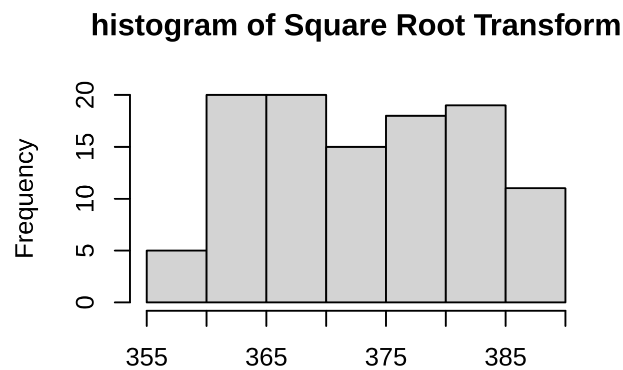
There is no sharp differences between the original time series plot and the time series plots with transformations.

However, we still need to make sure that our data is symmetric and around normal. Plotting the histogram of the original time series and the transformed ones, we get the following result:



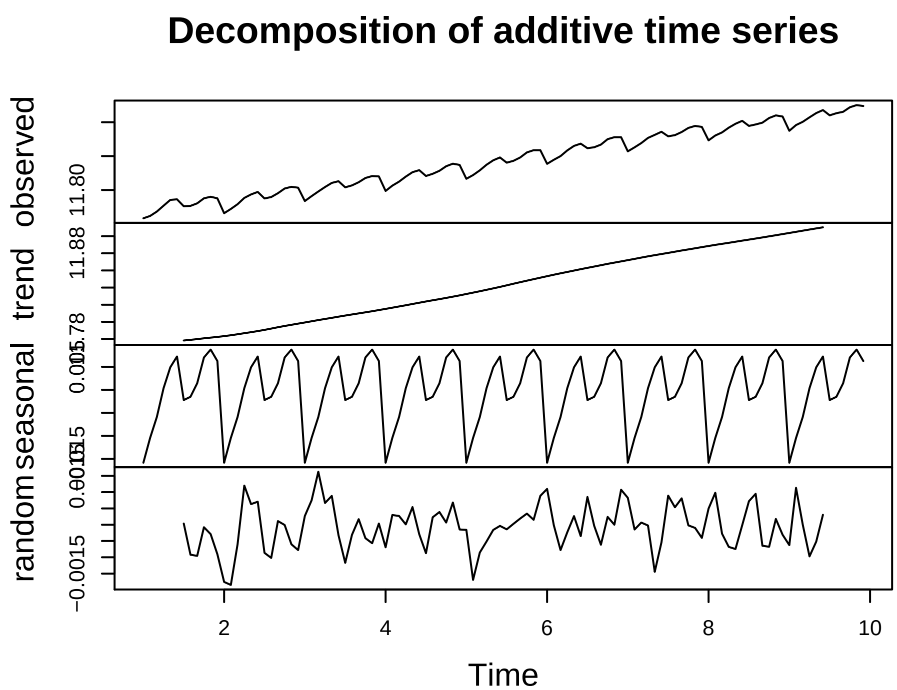






We can see that, the log-transformed data is more symmetric and around normal than original one. Therefore, **we choose to do log transformation on the data.**

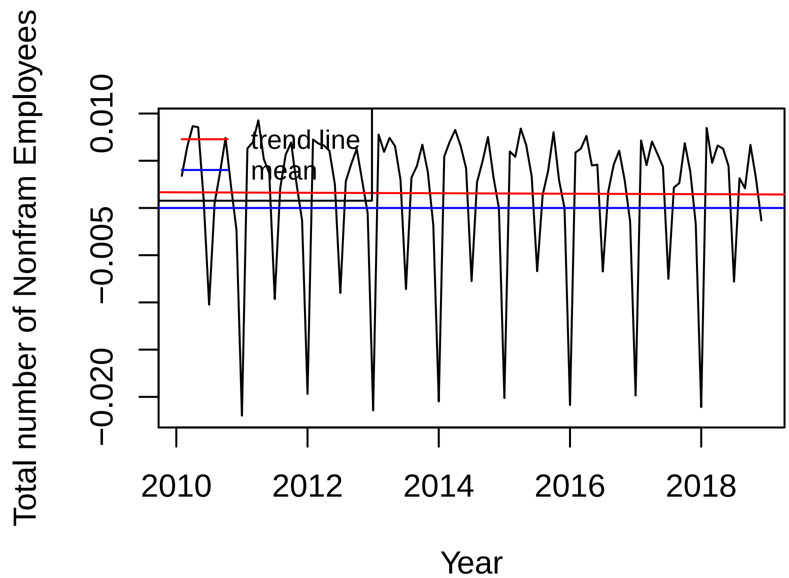
Then we draw the decomposition of the log transformed data.



There is a approximately linear increasing trend, with seasonality of around 1 year.

Remove Trend

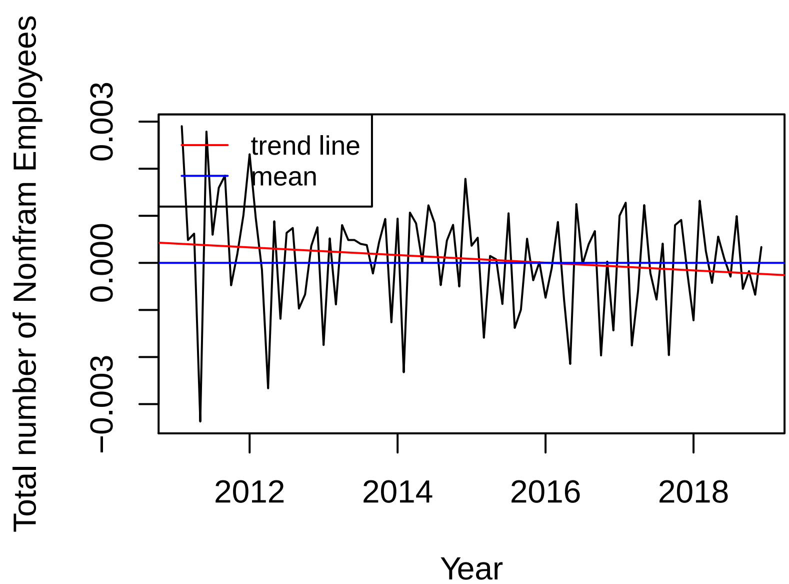
Since there is a clear linear upward trend, to eliminate trend, we need to difference at lag 1. After differencing, we have the time series plot as follows.



The trend line now is almost horizontal, meaning there is no trend on this time series data. The variance of the series differenced at lag 1 is *5.936e-05* . To double check if we need to difference again at lag 1, we calculated the variance of the series after secondly differenced at lag 1. The variance is *0.0001142* . Since the variance increased, there is need to difference the time series again at lag 1. Now the trend is eliminated.

Remove Seasonality

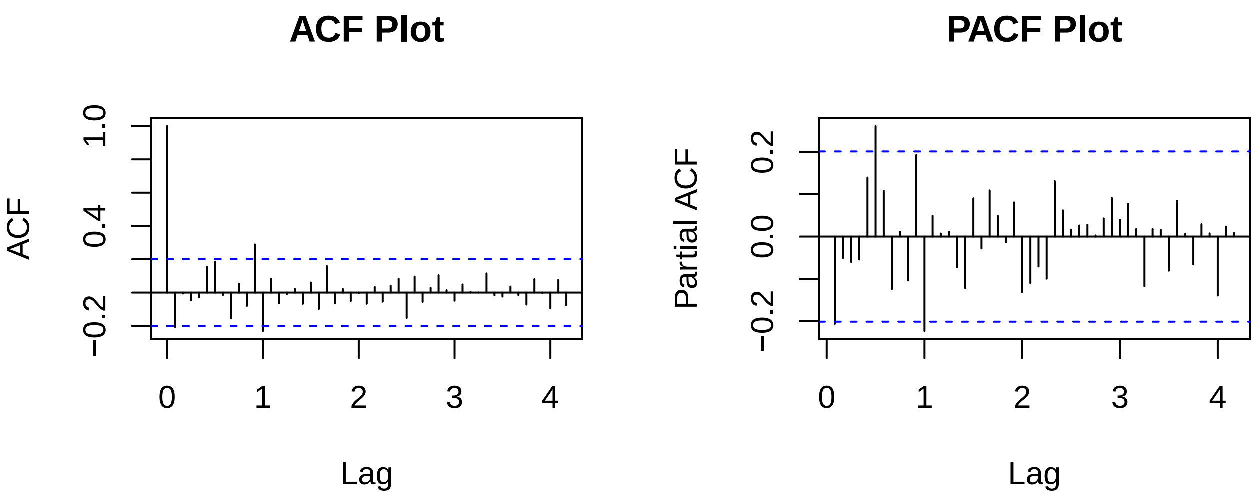
Seasonality exists and the recurring pattern occurs every year, so we need to **difference at lag 12 to reduce seasonality**. After differencing, we have the time series plot as follows.



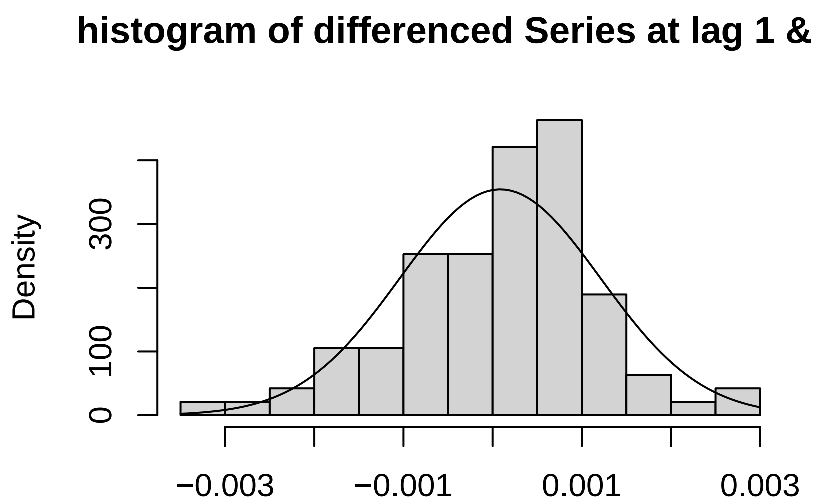
Although the trend line is not strictly horizontal, considering the scale of y-axis is 0.001, we consider the slope as very small, so generally, the line is almost horizontal. The result for the slope is as below:

seq(2011, 2019, length.out = 95)
-8.139e-05

Indeed, the slope is almost 0, so we conclude that we removed trend and seasonality. Now the time series data seems stationary, and the corresponding plot of ACF and PACF is as follows.



We can see that the ACF and PACF is behaving stationary. After all the making the data stationary, we lastly check if the final data is in normal distribution. The histogram is as follows:



The distribution is close to a normal distribution than the distribution without differencing. Hence, we can conclude that our data now is stationary, and we can proceed to estimate the model for our data.

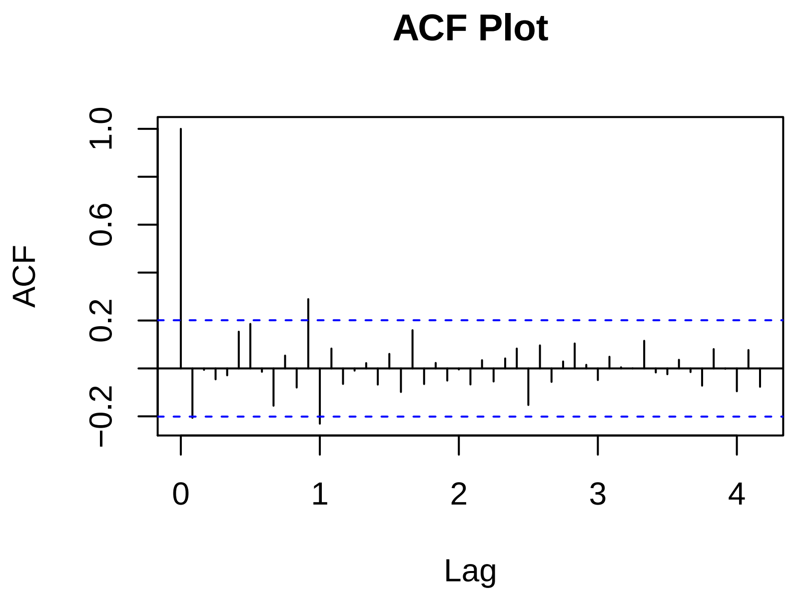
Model Identification

Determine Models

We are going to observe the ACF to determine the Moving Average part and observe the PACF to the Autoregressive part of the possible time series model. Assume that we have SARIMA$(p, d, q) \times (P, D, Q)_s$ model.

Firstly, since we know the seasonality is 12 months and we differenced at lag 1 and lag 12 each time, we get SARIMA$(p, 1, q) \times (P, 1, Q)_{12}$.

Then we observe at the ACF.



Modeling the seasonal part (P, D, Q): focus on the seasonal lags $h = 1s, 2s$, etc (with $s = 12$).

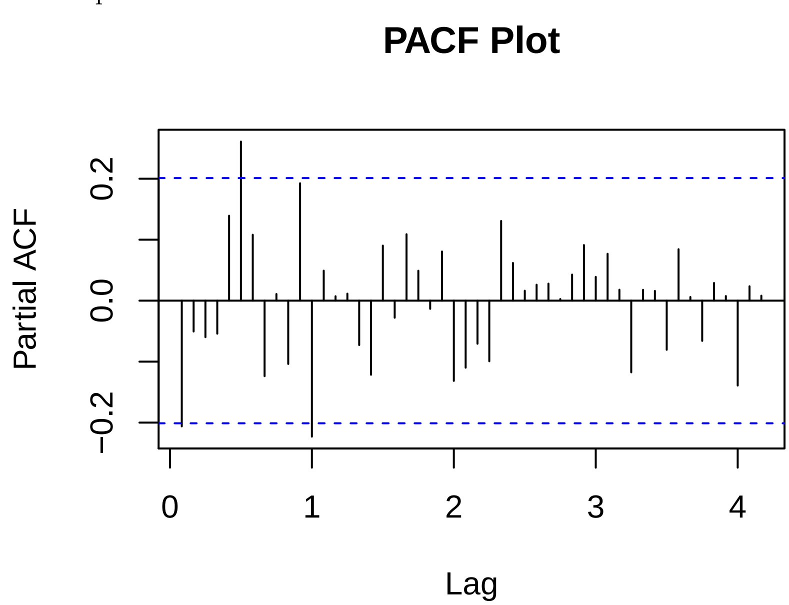
- The ACF shows one strong peak at $h = 1s = 12$ outside of the 95% confidence interval and smaller peaks appearing at the following. A good choice for the seasonal MA part could be $\mathbf{Q} = \mathbf{1}$.

Modeling the non-seasonal part (p, d, q): focus on lags $h = 1, \cdots, 11$.

- The ACF seems to have peaks at lag 1 and lag 11. Since the outstanding peaks at $s - 1 = 11$ is common, we disregard this situation. Hence, a good choice for the non-seasonal MA part could be $\mathbf{q} = \mathbf{1}$.

We also consider pure MA(12) model.

Next we observe the PACF plot.



Modeling the seasonal part (P, D, Q): focus on the seasonal lags $h = 1s, 2s$, etc (with $s = 12$).

- The PACF shows only one strong peaks at $h = 1s = 12$. A good choice for the seasonal AR part could be $\mathbf{P = 1}$.

Modeling the non-seasonal part (p , d, q): focus on lags $h = 1, \cdots, 11$.

- The PACF is strong at lags 1 and 6. A good choice for the AR part could be $\mathbf{p = 1}$ or $\mathbf{p = 6}$.

We also consider pure AR(12) model.

In conclusion, we have four candidate models:

1. SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$.
2. MA(12).
3. SARIMA$(6, 1, 1) \times (1, 1, 1)_{12}$.
4. AR(12).

Then we will compare and select the models for the best fit.

The AICc for the first model SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ is *-1021* .

The AICc for the second model MA(12) is *-1007* .

The AICc for the third model SARIMA$(6, 1, 1) \times (1, 1, 1)_{12}$ is *-1025* .

The AICc for the fourth model AR(12) is *-1021* .

From above, we select three models with the lowest three AICcs: SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$, SARIMA$(6, 1, 1) \times (1, 1, 1)_{12}$ and AR(12).

Modify Models

1. The model coefficients summary for SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ is

```
##
## Call:
## arima(x = ddemploy, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##          ar1      ma1    sar1     sma1  intercept
##       0.2814  -0.4497  0.2229  -0.5886      1e-04
## s.e.  0.3140   0.2771  0.2761   0.2441      1e-04
##
## sigma^2 estimated as 1.075e-06:  log likelihood = 516.82,  aic = -1021.64
```

We can see that 95% confidence interval for some variables includes 0 (for `ma1`), so we choose to test models without those variables and check the AICc to choose the model with lower AICc.

After all the calculations, the best modification of the SARIMA$(1, 1, 1) \times (1, 1, 1)_{12}$ model is SARIMA$(0, 1, 0) \times (1, 1, 1)_{12}$. The coefficients output is

```
##
## Call:
## arima(x = ddemploy, order = c(0, 0, 0), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##         sar1     sma1  intercept
##       0.1745  -0.5821      1e-04
## s.e.  0.2433   0.2149      1e-04
```

```
##
## sigma^2 estimated as 1.1e-06:  log likelihood = 515.55,  aic = -1023.09
```

Note that the intercept is comparatively small, so we consider ignoring it. Hence, explicit form of this model is

$$(1 - 0.1745_{0.2433}B^{12})(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5821_{0.2149}B^{12})Z_t.$$

(Refer Appendix for full selection process.) The AICc for this modified model is *-1023* , which is indeed lower than previous model.

2. The model coefficients summary for SARIMA$(6, 1, 1) \times (1, 1, 1)_{12}$ is

```
##
## Call:
## arima(x = ddemploy, order = c(6, 0, 1), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4     ar5     ar6      ma1    sar1
##       0.0301  -0.0709  -0.0752  -0.0209  0.1983  0.3853  -0.1922  0.0806
## s.e.  0.2229   0.1014   0.1034   0.1043  0.1099  0.1250   0.2171  0.2771
##          sma1  intercept
##       -0.5939      1e-04
## s.e.   0.2491      1e-04
##
## sigma^2 estimated as 8.903e-07:  log likelihood = 524.68,  aic = -1027.37
```

We can see that 95% confidence interval for some variables includes 0 (for `ar2`, `ar3`, `ar4`, `ma1`), so we choose to test models without those variables and check the AICc to choose the model with lower AICc.

After all the calculations, the best modification of the SARIMA$(6, 1, 1) \times (1, 1, 1)_{12}$ model is SARIMA$(6, 1, 0) \times (0, 1, 1)_{12}$ with coefficients of `ar1`, `ar2`, `ar3` and `ar4` being zero. The coefficients output is

```
##
## Call:
## arima(x = ddemploy, order = c(6, 0, 0), seasonal = list(order = c(0, 0, 1),
##     period = 12), transform.pars = FALSE, fixed = c(0, 0, 0, 0, NA, NA, NA,
##     NA), method = "ML")
##
## Coefficients:
##       ar1  ar2  ar3  ar4     ar5     ar6     sma1  intercept
##         0    0    0    0  0.2070  0.4179  -0.5499      1e-04
## s.e.    0    0    0    0  0.1045  0.1125   0.1190      2e-04
##
## sigma^2 estimated as 9.248e-07:  log likelihood = 522.85,  aic = -1035.7
```

The intercept is comparatively small, so we consider ignoring it. Hence, explicit form of this model is

$$(1 - 0.2070_{0.1045}B^5 - 0.4179_{0.1125}B^6)(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5499_{0.1190}B^{12})Z_t.$$

(Refer Appendix for full selection process.) The AICc for this modified model is *-1034* , which is indeed lower than previous model.

3. The model coefficients summary for AR(12) is

```
##
## Call:
## arima(x = ddemploy, order = c(12, 0, 0), method = "ML")
##
## Coefficients:
```

```
##          ar1      ar2     ar3      ar4     ar5     ar6     ar7      ar8
##      -0.1803  -0.0747  0.0133  -0.0230  0.1910  0.3042  0.1031  -0.2030
## s.e.   0.1002   0.1005  0.0982   0.1036  0.1023  0.1053  0.1057   0.1061
##          ar9     ar10    ar11     ar12  intercept
##      -0.0231  -0.1013  0.1830  -0.2908      1e-04
## s.e.   0.1089   0.1089  0.1092   0.1093      1e-04
##
## sigma^2 estimated as 8.603e-07:  log likelihood = 526.89,  aic = -1025.78
```

We can see that 95% confidence interval for some variables includes 0 (for `ar1`, `ar2`, `ar4`, `ar8`, `ar9` and `ar10`), so we choose to test models without those variables and check the AICc to choose the model with lower AICc.

After all the calculations, the best modification of the AR(12) model is AR(12) with coefficients of `ar1`, `ar2`, `ar3`, `ar4`, `ar7`, `ar8`, `ar9` and `ar10` being 0. The coefficients output is

```
##
## Call:
## arima(x = ddemploy, order = c(12, 0, 0), transform.pars = FALSE, fixed = c(0,
##     0, 0, 0, NA, NA, 0, 0, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##      ar1  ar2  ar3  ar4     ar5     ar6  ar7  ar8  ar9  ar10    ar11     ar12
##        0    0    0    0  0.1613  0.2636    0    0    0     0  0.2396  -0.2853
## s.e.    0    0    0    0  0.1012  0.1055    0    0    0     0  0.1062   0.1068
##      intercept
##          1e-04
## s.e.      2e-04
##
## sigma^2 estimated as 9.602e-07:  log likelihood = 522.08,  aic = -1032.16
```

Explicit form of this model is

$$(1 - 0.1613_{0.1012}B^5 - 0.2636_{0.1055}B^6 - 0.2396_{0.1062}B^{11} + 0.2853_{0.1068}B^{12})(1 - B)(1 - B^{12})\ln(X_t) = Z_t.$$

(Refer Appendix for full selection process.) The AICc for this modified model is

In conclusion, the two models we selected for further analysis is

Model (A): $\mathbf{SARIMA(0,1,0)\times(1,1,1)_{12}}$

$$\mathbf{(1 - 0.1745_{0.2433}B^{12})(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5821_{0.2149}B^{12})Z_t},$$

Model (B): $\mathbf{SARIMA(6,1,0)\times(0,1,1)_{12}}$

$$\mathbf{(1 - 0.2070_{0.1045}B^5 - 0.4179_{0.1125}B^6)(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5499_{0.1190}B^{12})Z_t}.$$

Model (C): $\mathbf{AR(12)}$

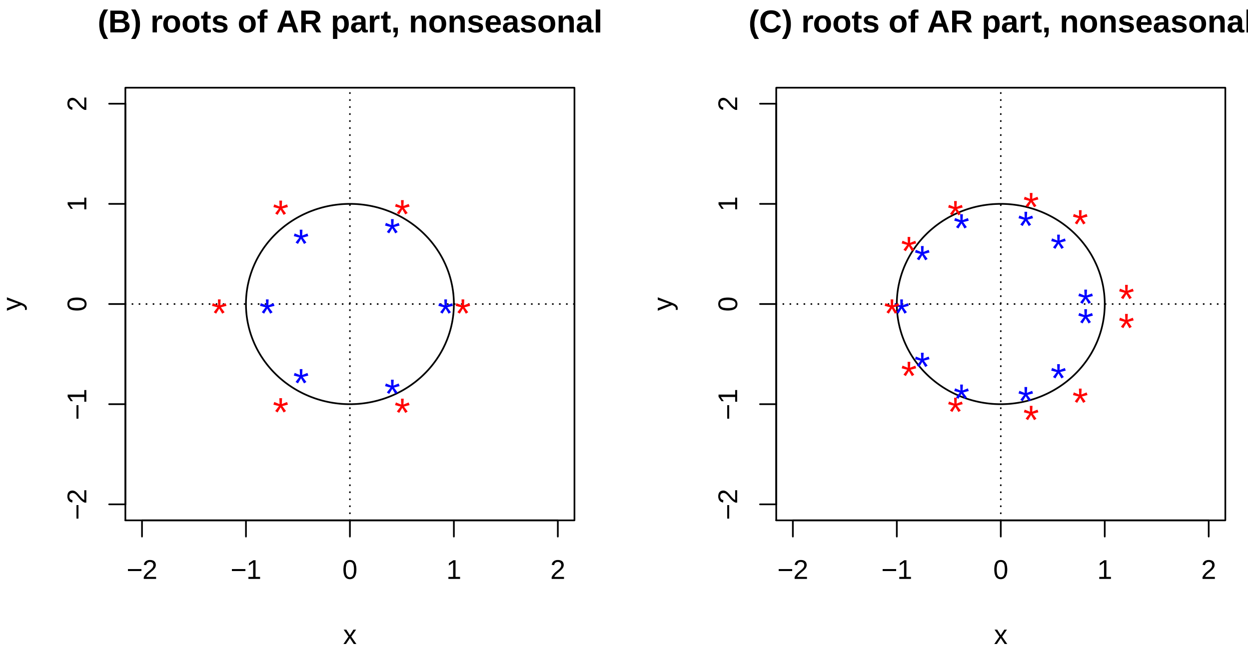
$$\mathbf{(1 - 0.1613_{0.1012}B^5 - 0.2636_{0.1055}B^6 - 0.2396_{0.1062}B^{11} + 0.2853_{0.1068}B^{12})(1 - B)(1 - B^{12})\ln(X_t) = Z_t}$$

Check Invertibility and Stationarity

Now we check the invertibility and stationarity of the models.

- Model(A) is stationary because $|\Phi_1| = 0.1745 < 1$. Model(A) is invertible because $|\Theta_1| = 0.5821 < 1$.
- Model(B) is stationary because the roots of non-seasonal AR part are all outside of the unit circle. Model(B) is invertible because $|\Theta_1| = 0.5499 < 1$.

- Model(C) is stationary the roots of non-seasonal AR part are all outside of the unit circle. Model(C) is invertible because all AR models are invertible.



(Note that the red points are roots)

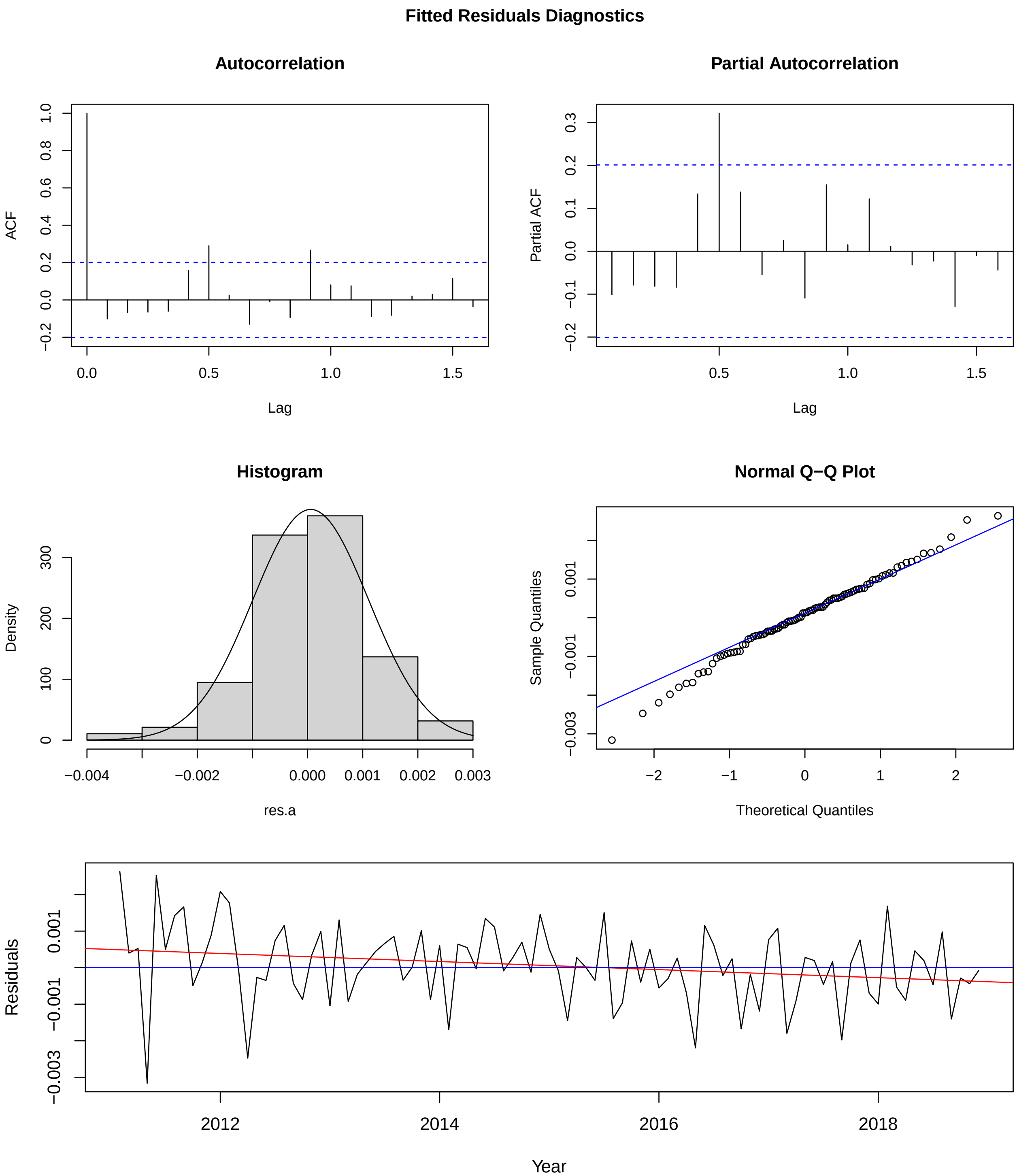
All models are both invertible and stationary, so Model(A), Model(B) and Model(C) are all feasible models for our problem. Then we proceed to diagnostic chekcing for all three models to determine the final model.

Diagnostic Checking

Residual Analysis

In order to determine the best fit model, we are going to check the residuals of the two feasible models. The residuals of a well-fitted model should resemble Gaussian White Noise.

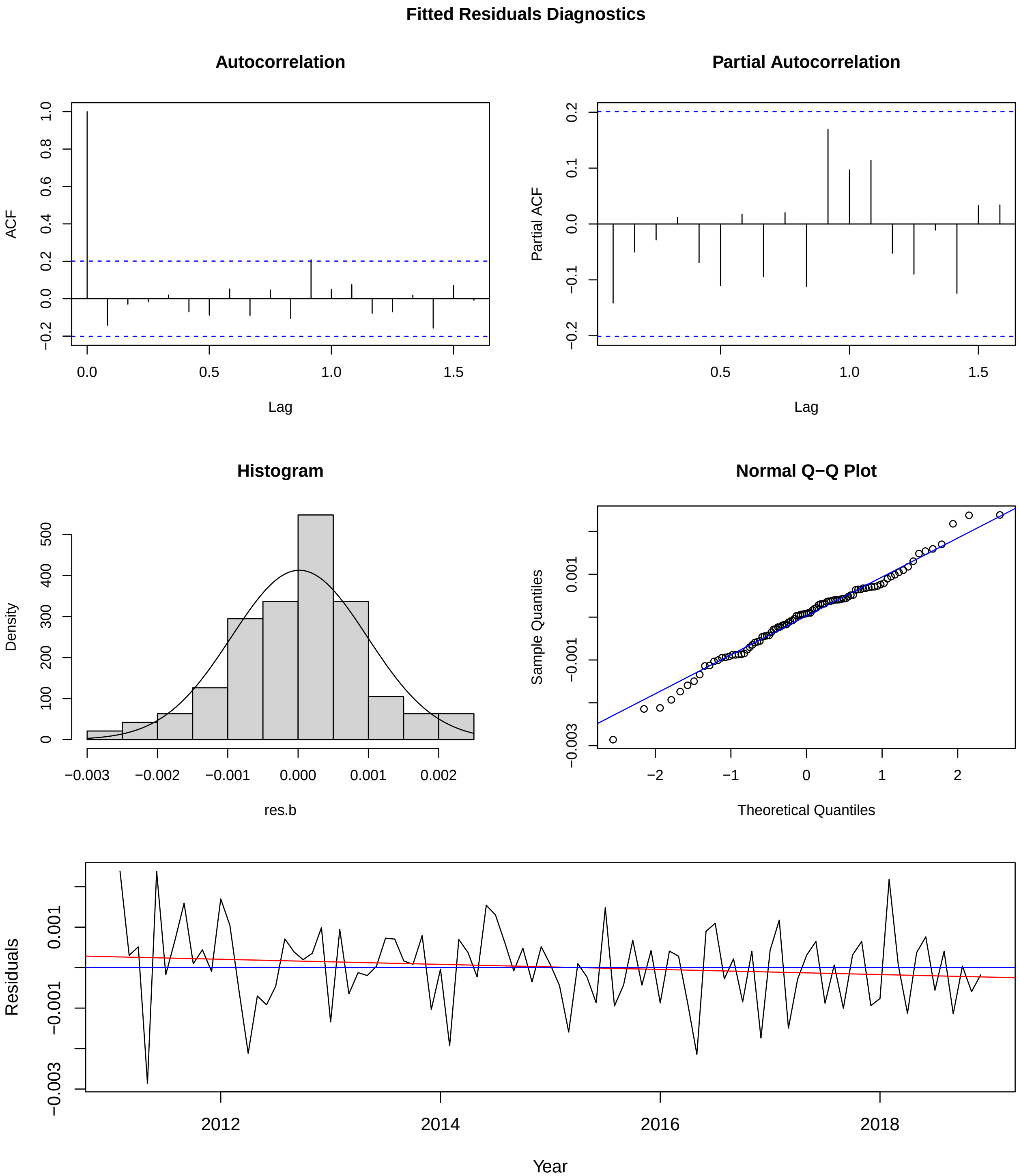
Firstly, we check the residuals for Model(A) $(1 - 0.1745_{0.2433}B^{12})(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5821_{0.2149}B^{12})Z_t$. We plot the ACF, PACF, histogram and Normal Q-Q plot of residuals, as well as the time series data to have a general overview.



From the fitted residuals plot, we can see that the residuals don't have a strong trend or change in variance or seasonality (Although trend appears but we consider it as very small to ignore). The ACF and PACF of the residuals have almost all values within the confidence interval with only one or two exceptions (except lag=0 for ACF which should always be 1). The Normal Q-Q plots show signs of being a bit heavily tailed, but not much. The histogram also seems to be close to a normal distribution with mean 0 and standard deviation 1. Therefore, the residuals of Model(A) may resemble white noise, but whether we will use Model(A) as our final model will depend on further tests.

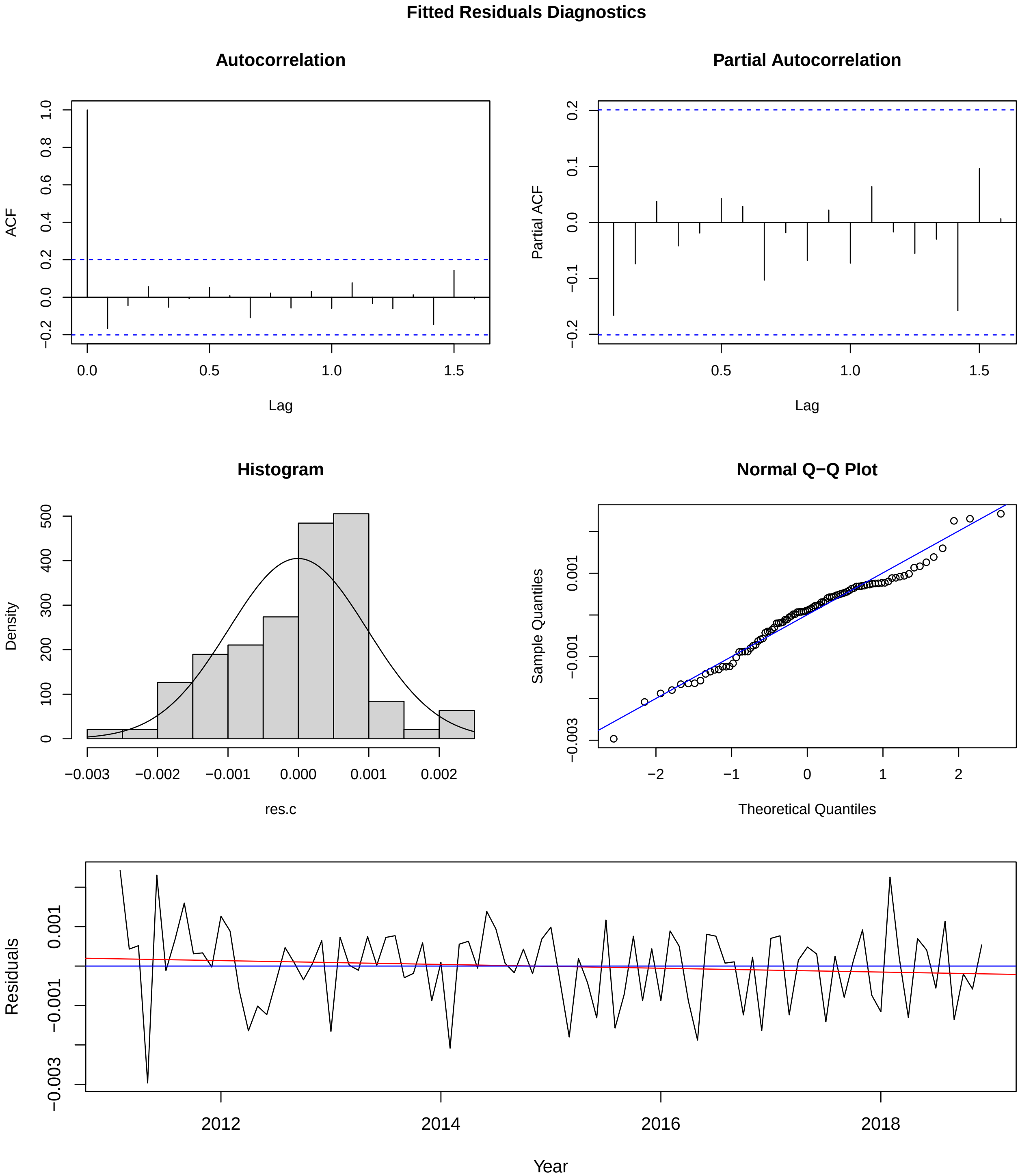
For Model(B) $(1-0.2070_{0.1045}B^5-0.4179_{0.1125}B^6)(1-B)(1-B^{12})\ln(X_t)=(1-0.5499_{0.1190}B^{12})Z_t$, the

ACF, PACF, histogram and Normal Q-Q plot of residuals, as well as the time series data as follows.



We observe that the Model(B) has similar residual behavior as Model(A), and even better that the ACF and PACF of the residuals of Model(B) have all values within the confidence interval. Therefore, we can also conclude that the residuals for Model(B) performs better than Model(A), and it also should resemble White Noise.

For Model(C) $(1-0.1613_{0.1012}B^5-0.2636_{0.1055}B^6-0.2396_{0.1062}B^{11}+0.2853_{0.1068}B^{12})(1-B)(1-B^{12})\ln(X_t) = Z_t$, the ACF, PACF, histogram and Normal Q-Q plot of residuals, as well as the time series data as follows.



Similar to Model(B), the ACF and PACF of the residuals of Model(C) have all values within the confidence interval, but the distribution is a little skewed rather than normal, and the QQ plot line is not strictly linear, so the residuals of Model(C) performed worse than residuals of Model(B). However, generally, we can still assume that the residuals for Model(C) resemble White Noise and proceed to tests for further analysis.

Tests for Model(A)

The Shapiro-Wilk test for Model(A) is as follows:

Table 2: Shapiro-Wilk normality test: `res.a`

Test statistic	P value
0.9908	0.7612

Here we have 108 data, so the degree of freedom for Portmanteau Tests is $\sqrt{108} \approx 10$. For Model (A), there are 2 parameters (`sar1` and `sma1`). The Box-Pierce test result is as follows:

Table 3: Box-Pierce test: `res.a`

Test statistic	df	P value
15.01	8	0.05886

The Ljung-Box test result is as follows:

Table 4: Box-Ljung test: `res.a`

Test statistic	df	P value
16.32	8	0.03802 *

The McLeod-Li test result is as follows:

Table 5: Box-Ljung test: `res.a^2`

Test statistic	df	P value
14.46	10	0.1529

The Yule-Walker test result for Model(A) is

```
##
## Call:
## ar(x = res.a, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
##       1        2        3        4        5        6
## -0.1540  -0.0611  -0.0533  -0.0410   0.1692   0.3217
##
## Order selected 6  sigma^2 estimated as  1.011e-06
```

The residuals of Model(A) have p-value less than 0.05 only for Ljung-Box test, so we reject that the residuals are independently distributed, and it has Yule-Walker of order selected 6 rather than 0. Although it passed the Shapiro-Wilk normality test, Box-Pierce test and McLeod-Li test, Model(A) is generally not a good choice.

Tests for Model(B)

The Shapiro-Wilk test for Model(B) is as follows:

Table 6: Shapiro-Wilk normality test: `res.b`

Test statistic	P value
0.9862	0.4226

Here we have 108 data, so the degree of freedom for Portmanteau Tests is $\sqrt{108} \approx 10$. For Model (B), there are 3 parameters (`ar5`, `ar6`, `sma1`). The Box-Pierce test result is as follows:

Table 7: Box-Pierce test: `res.b`

Test statistic	df	P value
5.525	7	0.5962

The Ljung-Box test result is as follows:

Table 8: Box-Ljung test: `res.b`

Test statistic	df	P value
5.984	7	0.5416

The McLeod-Li test result is as follows:

Table 9: Box-Ljung test: `res.b^2`

Test statistic	df	P value
12.36	10	0.2617

The Yule-Walker test result for Model(B) is

```
##
## Call:
## ar(x = res.b, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  9.343e-07
```

Notice that all p-values for the tests for Model(B) are all larger than 0.05, and the Yule-Walker test for Model(B) have "Order selected 0", meaning the residuals should all be fitted into AR(0) (White Noise). Therefore, we can conclude that Model(B) passed Diagnostic Checking, so it ready to be used for forecasting.

Tests for Model(C)

The Shapiro-Wilk test for Model(C) is as follows:

Table 10: Shapiro-Wilk normality test: `res.c`

Test statistic	P value
0.9753	0.06959

Here we have 108 data, so the degree of freedom for Portmanteau Tests is $\sqrt{108} \approx 10$. For Model (C), there are 4 parameters (`ar5`, `ar6`, `ar11`, `ar12`). The Box-Pierce test result is as follows:

Table 11: Box-Pierce test: `res.c`

Test statistic	df	P value
5.164	6	0.5229

The Ljung-Box test result is as follows:

Table 12: Box-Ljung test: `res.c`

Test statistic	df	P value
5.497	6	0.4818

The McLeod-Li test result is as follows:

Table 13: Box-Ljung test: `res.c^2`

Test statistic	df	P value
10.6	10	0.3898

The Yule-Walker test result for Model(B) is

```
##
## Call:
## ar(x = res.c, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
## Coefficients:
##       1
## -0.1661
##
## Order selected 1  sigma^2 estimated as  9.537e-07
```

Note that Model(B) didn't pass the Yule-Walker test because it has "Order selected 1", meaning the residuals can't all be fitted into AR(0) (White Noise).

Since Mode(B) has a better performance than Model(A) and Model(C) for residuals, we choose Model(B) as our final model.

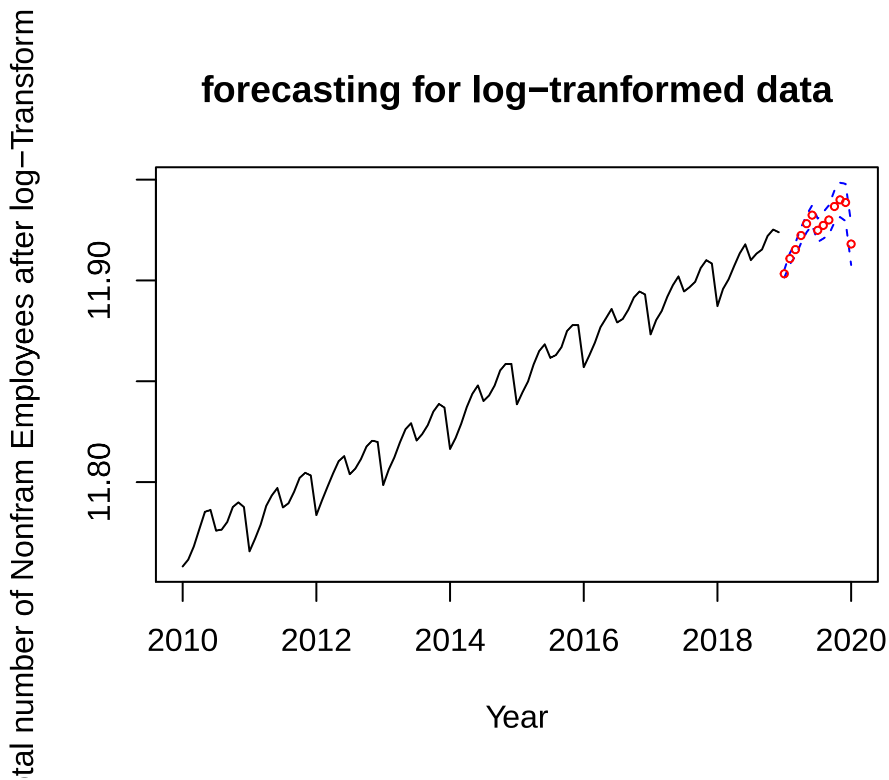
Our final Model for the original data follows SARIMA$(\mathbf{6}, \mathbf{1}, \mathbf{0}) \times (\mathbf{0}, \mathbf{1}, \mathbf{1})_{\mathbf{12}}$. The algebraic expression is

$$\mathbf{(1 - 0.2070_{0.1045}B^5 - 0.4179_{0.1125}B^6)(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5499_{0.1190}B^{12})Z_t}.$$

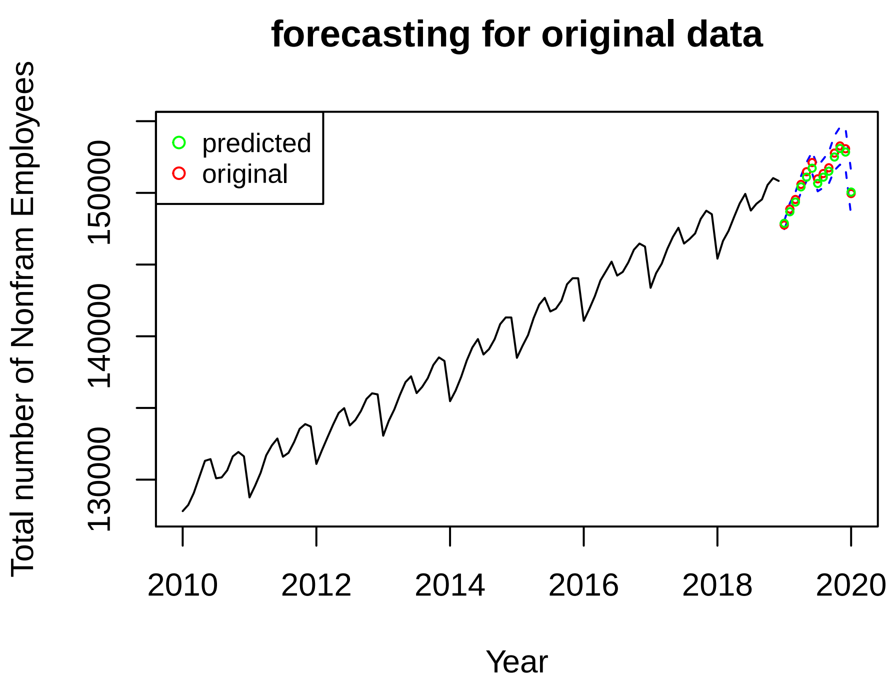
Forecasting

We are going to predict the next 13 time points by using the SARIMA$(6, 1, 0) \times (0, 1, 1)_{12}$ model, and compare it with the true data.

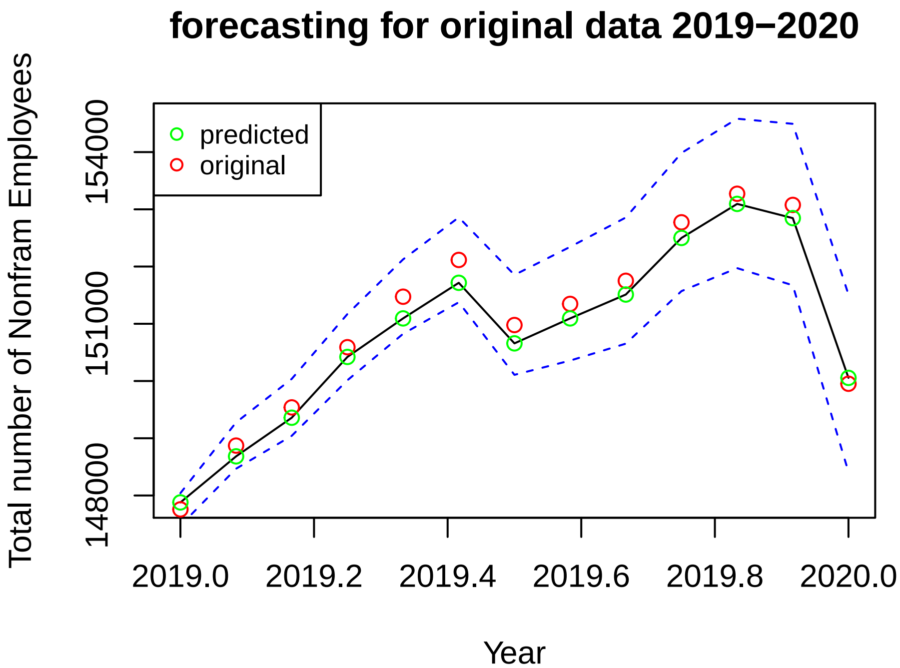
Here we plot the predicted montly number of Non-farm Employment from 01/01/2019 to 01/01/2020.



Remember that we used log transformation on data for model identification, but for forecasting and prediction, it needs to be the original data. We here transform the data back by taking the exponential on the prediction results (if we have $Y_t = \ln(X_t)$, then $X_t = \exp(Y_t)$). The final original forecast is as follows:



Zooming in to only looking at the 2019-2020 prediction part, we have



The blue dashed lines are the confidence intervals, and the green points are the true value. We can see that our predicted values (red points), are all in the confidence intervals and very close to the true value, with a large portion of overlapping. Therefore, we can conclude that our selected Model(B) is satisfactory.

Conclusion

Since the predictions are very close to the true values and all fall in the confidence intervals, we can conclude that our final model for the number of Non-farm Emplotment data is SARIMA$(6, 1, 0) \times (0, 1, 1)_{12}$ with explicit form $(1 - 0.2070_{0.1045}B^5 - 0.4179_{0.1125}B^6)(1 - B)(1 - B^{12})\ln(X_t) = (1 - 0.5499_{0.1190}B^{12})Z_t$. We successfully predicted the data of the 13 following months by using the data from 2010-01-01 to 2018-12-01 monthly, not seasonally adjusted data.

Acknowledgement

I want to express my sincere thanks for Professor Raya Feldman: thank you so much for giving us the great lectures this quarter, and thank you so much for taking your time making appointment with me and leading me to the right track for the final project. I also appreciated my TA Chao Zhang who also provided great help.

Appendix

Reference

Data: https://fred.stlouisfed.org/series/PAYNSA

Modification Process for the First Model

Observing the model coefficients output for model $\text{SARIMA}(11,1,1) \times (1,1,1)_{12}$ with third lowest AICc is

```
fit.i
```

```
##
## Call:
## arima(x = ddemploy, order = c(1, 0, 1), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##          ar1      ma1    sar1     sma1  intercept
##       0.2814  -0.4497  0.2229  -0.5886      1e-04
## s.e.  0.3140   0.2771  0.2761   0.2441      1e-04
##
## sigma^2 estimated as 1.075e-06:  log likelihood = 516.82,  aic = -1021.64
```

Notice that the the 95% confidence interval for `ma1` is $(-0.4497 - 1.96 \cdot 0.2771, -0.4497 + 1.96 \cdot 0.2771)$ which contains 0, so we consider removing the non-seasonal MA part. Then the model becomes $\text{SARIMA}(1,1,0) \times (1,1,1)_{12}$. Then the AICc for the modified model is

```
fit.i1 <- arima(ddemploy, order = c(1,0,0),
                seasonal = list(order = c(1,0,1),period = 12),
                method = "ML", transform.pars = FALSE)
AICc(fit.i1)
```

```
## [1] -1022.176
```

It is lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.i1
```

```
##
## Call:
## arima(x = ddemploy, order = c(1, 0, 0), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##           ar1    sar1     sma1  intercept
##       -0.1390  0.2248  -0.5778      1e-04
## s.e.   0.1116  0.2727   0.2383      1e-04
##
## sigma^2 estimated as 1.089e-06:  log likelihood = 516.31,  aic = -1022.62
```

Notice that the 95% confidence interval for `ar1` is $(-0.1390-1.96\cdot0.1116, -0.1390+1.96\cdot0.1116)$ which contains 0, so we consider removing the non-seasonal AR part. Then the model becomes $\text{SARIMA}(0,1,0) \times (1,1,1)_{12}$. The AICc for the modified model is

```
fit.i1 <- arima(ddemploy, order = c(0,0,0),
                seasonal = list(order = c(1,0,1),period = 12),
                method = "ML", transform.pars = FALSE)
AICc(fit.i1)
```

```
## [1] -1022.828
```

It is a little lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.i1
```

```
##
## Call:
## arima(x = ddemploy, order = c(0, 0, 0), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##         sar1     sma1  intercept
##       0.1745  -0.5821      1e-04
## s.e.  0.2433   0.2149      1e-04
##
## sigma^2 estimated as 1.1e-06:  log likelihood = 515.55,  aic = -1023.09
```

Now there is no further modification needed. Hence, the best modified model is SARIMA$(0,1,0) \times (1,1,1)_{12}$.

Modification Process for the Second Model

Observing the model coefficients output for model SARIMA$(6,1,1) \times (1,1,1)_{12}$ with second lowest AICc is

```
# third model summary
fit.iii
```

```
##
## Call:
## arima(x = ddemploy, order = c(6, 0, 1), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, method = "ML")
##
## Coefficients:
##          ar1      ar2      ar3      ar4     ar5     ar6      ma1    sar1
##       0.0301  -0.0709  -0.0752  -0.0209  0.1983  0.3853  -0.1922  0.0806
## s.e.  0.2229   0.1014   0.1034   0.1043  0.1099  0.1250   0.2171  0.2771
##         sma1  intercept
##      -0.5939      1e-04
## s.e.   0.2491      1e-04
##
## sigma^2 estimated as 8.903e-07:  log likelihood = 524.68,  aic = -1027.37
```

Notice that the the 95% confidence interval for `ma1` is $(-0.1922 - 1.96 \cdot 0.2171, -0.1922 + 1.96 \cdot 0.2171)$ which contains 0, so we consider removing the non-seasonal MA part, making it SARIMA$(6,1,0) \times (1,1,1)_{12}$. Similarly, the 95% confidence interval for `ar2`, `ar3`, `ar4` all contains 0, fixing the coefficient for them to be 0. Then the AICc for the modified model is

```
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
               fixed = c(NA,0,0,0,NA,NA,NA,NA,NA))
AICc(fit.iii1)
```

```
## [1] -1031.016
```

It is lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.iii1
```

```
##
```

```
## Call:
## arima(x = ddemploy, order = c(6, 0, 0), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, fixed = c(NA, 0, 0, 0, NA, NA, NA,
##     NA, NA), method = "ML")
##
## Coefficients:
##           ar1  ar2  ar3  ar4     ar5     ar6    sar1     sma1  intercept
##       -0.1279    0    0    0  0.1933  0.4175  0.0484  -0.5368      1e-04
## s.e.   0.1068    0    0    0  0.1089  0.1162  0.2742   0.2429      1e-04
##
## sigma^2 estimated as 9.17e-07:  log likelihood = 523.57,  aic = -1033.13
```

Now the 95% confidence interval for `ar1` is $(-0.1279 - 1.96 \cdot 0.1068, -0.1279 + 1.96 \cdot 0.1068)$ which contains 0, so we consider making the coefficient for `ar1` 0. Then the AICc for the modified model is

```
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
               fixed = c(0,0,0,0,NA,NA,NA,NA,NA))
AICc(fit.iii1)
```

```
## [1] -1031.587
```

It is lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.iii1
```

```
##
## Call:
## arima(x = ddemploy, order = c(6, 0, 0), seasonal = list(order = c(1, 0, 1),
##     period = 12), transform.pars = FALSE, fixed = c(0, 0, 0, 0, NA, NA, NA,
##     NA, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2  ar3  ar4     ar5     ar6    sar1     sma1  intercept
##         0    0    0    0  0.2073  0.4175  0.0026  -0.5518      1e-04
## s.e.    0    0    0    0  0.1069  0.1167  0.2406   0.2176      2e-04
##
## sigma^2 estimated as 9.247e-07:  log likelihood = 522.85,  aic = -1033.7
```

Since the coefficient for `sar1` is very small, we consider dropping the `sar1` part, making the model $\mathrm{SARIMA}(6,1,0) \times (0,1,1)_{12}$. Then the AICc for the modified model is

```
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(0,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
               fixed = c(0,0,0,0,NA,NA,NA,NA))
AICc(fit.iii1)
```

```
## [1] -1034.03
```

It is lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.iii1
```

```
##
## Call:
## arima(x = ddemploy, order = c(6, 0, 0), seasonal = list(order = c(0, 0, 1),
##     period = 12), transform.pars = FALSE, fixed = c(0, 0, 0, 0, NA, NA, NA,
##     NA), method = "ML")
```

```
##
## Coefficients:
##       ar1  ar2  ar3  ar4     ar5     ar6     sma1  intercept
##         0    0    0    0  0.2070  0.4179  -0.5499      1e-04
## s.e.    0    0    0    0  0.1045  0.1125   0.1190      2e-04
##
## sigma^2 estimated as 9.248e-07:  log likelihood = 522.85,  aic = -1035.7
```

Now there is no further modification needed. Hence, the best modified model is SARIMA$(6,1,0)\times(0,1,1)_{12}$.

Modification Process for the Third Model

Observing the model coefficients output for model SARIMA$(6,1,1)\times(1,1,1)_{12}$ with second lowest AICc is

```
# forth model summary
fit.iv
```

```
##
## Call:
## arima(x = ddemploy, order = c(12, 0, 0), method = "ML")
##
## Coefficients:
##           ar1      ar2     ar3      ar4     ar5     ar6     ar7      ar8
##       -0.1803  -0.0747  0.0133  -0.0230  0.1910  0.3042  0.1031  -0.2030
## s.e.   0.1002   0.1005  0.0982   0.1036  0.1023  0.1053  0.1057   0.1061
##           ar9     ar10    ar11     ar12  intercept
##       -0.0231  -0.1013  0.1830  -0.2908      1e-04
## s.e.   0.1089   0.1089  0.1092   0.1093      1e-04
##
## sigma^2 estimated as 8.603e-07:  log likelihood = 526.89,  aic = -1025.78
```

The 95% confidence interval for `ar1`, `ar2`, `ar4`, `ar8`, `ar9` and `ar10` includes 0, so we choose to fix the coefficient for them to be 0. Then the AICc for the modified model is

```
fit.iv1 <- arima(ddemploy, order = c(12,0,0),
                method = "ML", transform.pars = FALSE,
                fixed = c(0,0,NA,0,NA,NA,NA,0,0,0,NA,NA,NA))
AICc(fit.iv1)
```

```
## [1] -1024.524
```

It is lower than previous AICc, showing that this model is better. The model coefficient now becomes

```
fit.iv1
```

```
##
## Call:
## arima(x = ddemploy, order = c(12, 0, 0), transform.pars = FALSE, fixed = c(0,
##     0, NA, 0, NA, NA, NA, 0, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2     ar3  ar4     ar5     ar6     ar7  ar8  ar9  ar10    ar11
##         0    0  0.0064    0  0.1729  0.2849  0.0951    0    0     0  0.2395
## s.e.    0    0  0.0911    0  0.1013  0.1075  0.1021    0    0     0  0.1069
##          ar12  intercept
##       -0.3032      1e-04
## s.e.   0.1077      2e-04
##
```

```
## sigma^2 estimated as 9.502e-07:  log likelihood = 522.51,  aic = -1029.02
```

Notice that the coefficient for `ar3` and `ar7` is very small, we consider making it 0.

```
fit.iv1 <- arima(ddemploy, order = c(12,0,0),
                 method = "ML", transform.pars = FALSE,
                 fixed = c(0,0,0,0,NA,NA,0,0,0,0,NA,NA,NA))
AICc(fit.iv1)
```

```
## [1] -1027.662
```

The AICc also drops, now the coefficients become

```
fit.iv1
```

```
##
## Call:
## arima(x = ddemploy, order = c(12, 0, 0), transform.pars = FALSE, fixed = c(0,
##     0, 0, 0, NA, NA, 0, 0, 0, 0, NA, NA, NA), method = "ML")
##
## Coefficients:
##       ar1  ar2  ar3  ar4     ar5     ar6  ar7  ar8  ar9  ar10    ar11     ar12
##         0    0    0    0  0.1613  0.2636    0    0    0     0  0.2396  -0.2853
## s.e.    0    0    0    0  0.1012  0.1055    0    0    0     0  0.1062   0.1068
##       intercept
##           1e-04
## s.e.      2e-04
##
## sigma^2 estimated as 9.602e-07:  log likelihood = 522.08,  aic = -1032.16
```

Now there is no further modification needed. Hence, the best modified model is AR(12) with coefficients of `ar1`, `ar2`, `ar3`, `ar4`, `ar7`, `ar8`, `ar9` and `ar10` being 0.

Codes

```
knitr::opts_chunk$set(echo=F,
                      eval = T,
                      results='markup',
                      message=F,
                      warning=F,
                      fig.height=4,
                      fig.width=5,
                      fig.align='center')

library(tidyverse)
library(pander)
library(ggplot2)
library(tsdl)
library(astsa)
library(dse)
library(rgl)
library(qpcR)

# load data
employ <- read.csv("PAYNSA.csv")

# change the class of date variable
employ$DATE <- as.Date.character(employ$DATE,"%Y-%m-%d")

# select data we wanted
employ <- employ%>%
  filter(DATE>="2010-01-01",DATE<="2020-01-01")

# separate training data
employ_train <- employ %>%
  filter(DATE>="2010-01-01",DATE<"2019-01-01")

# separate testing data
employ_test <- employ %>%
  filter(DATE>="2019-01-01",DATE<="2020-01-01")

# change data to time series
employ_ts <- ts(employ_train[,2],start = c(2010, 1), frequency = 12)
employ_ts_test <- ts(employ_test[,2],start = c(2019, 1), frequency = 12)

# plot
ts.plot(employ_ts,gpars=list(xlab="Year", ylab="Total number of Nonfram Employees"))

# plot original ACF & PACF
op = par(mfrow = c(1,2))
acf(employ_ts,lag.max = 50, main = "ACF Plot")
pacf(employ_ts,lag.max = 50, main = "PACF Plot")

# Box-cox transformation
t <- 1:length(employ_ts)
bcTransform <- MASS::boxcox(employ_ts ~ t,plotit = FALSE)
lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

```r
employ_bc <- (1/lambda)*(employ_ts^lambda - 1)

# log transform
employ_log <- log(employ_ts)

# square root transform
employ_sqrt <- sqrt(employ_ts)

# compare transformations
par(mfrow=c(2,2))
ts.plot(employ_ts, main = "Original Times Series")
ts.plot(employ_bc, main = "Box-Cox Transform")
ts.plot(employ_log, main = "Log Transform")
ts.plot(employ_sqrt, main = "Square Root Transform")

# compare histogram of transformations
par(mfrow=c(2,2))
hist(employ_ts, xlab = "", main = "histogram of Original Times Series")
hist(employ_bc, xlab = "", main = "histogram of Box-Cox Transform")
hist(employ_log, xlab = "", main = "histogram of Log Transform")
hist(employ_sqrt, xlab = "", main = "histogram of Square Root Transform")

y <- ts(as.ts(employ_log), frequency = 12)
decomp <- decompose(y)
plot(decomp)
# difference at lag 1
demploy <- diff(employ_log, 1)

# plot
ts.plot(demploy,gpars=list(xlab="Year", ylab="Total number of Nonfram Employees"))
abline(lm(demploy ~ seq(2010,2019, length.out = 107)),col = "red")
abline(h = 0, col = "blue")
legend("topleft", legend=c("trend line", "mean"),
       col=c("red","blue"), lty = 1, cex=0.8)
# calculate variance of series differenced at lag 1
var(demploy) %>% pander()

# calculate variance of series differenced again at lag 1
var(diff(demploy, 1)) %>% pander()

# difference at lag 12
ddemploy <- diff(demploy, 12)

# plot
ts.plot(ddemploy,gpars=list(xlab="Year", ylab="Total number of Nonfram Employees"))
abline(lm(ddemploy ~ seq(2011,2019, length.out = 95)),col = "red")
abline(h = 0, col = "blue")
legend("topleft", legend=c("trend line", "mean"),
       col=c("red","blue"), lty = 1, cex=0.8)

# slope of trend line
lm(ddemploy ~ seq(2011,2019, length.out = 95))$coef[2] %>%pander()
```

```r
# plot the final ACF and PACF
op = par(mfrow = c(1,2))
acf(ddemploy,lag.max = 50, main = "ACF Plot")
pacf(ddemploy,lag.max = 50, main = "PACF Plot")

# histogram of differenced data
hist(ddemploy,breaks=20, xlab="",
     main = "histogram of differenced Series at lag 1 & 12", prob=TRUE)
m <- mean(ddemploy)
std <- sqrt(var(ddemploy))
curve(dnorm(x,m,std), add=TRUE)

acf(ddemploy,lag.max = 50, main = "ACF Plot")

pacf(ddemploy,lag.max = 50, main = "PACF Plot")

# first model
fit.i <- arima(ddemploy, order = c(1,0,1),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE)
AICc(fit.i)%>% pander()

# second model
fit.ii <- arima(ddemploy, order = c(0,0,23),
               method = "ML")
AICc(fit.ii)%>% pander()

# third model
fit.iii <- arima(ddemploy, order = c(6,0,1),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE)
AICc(fit.iii) %>% pander()

# second model
fit.iv <- arima(ddemploy, order = c(12,0,0),
               method = "ML")
AICc(fit.iv)%>% pander()

# first model summary
fit.i

# modified first model
fit.a <- arima(ddemploy, order = c(0,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE)
fit.a

# AICc of modified first model
AICc(fit.a) %>% pander()

# third model summary
fit.iii
```

```r
# modified third model
fit.b <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(0,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
               fixed = c(0,0,0,0,NA,NA,NA,NA))
fit.b
# AICc of modified third model
AICc(fit.b) %>% pander()

# fourth model summary
fit.iv

fit.c <- arima(ddemploy, order = c(12,0,0),
               method = "ML", transform.pars = FALSE,
               fixed = c(0,0,0,0,NA,NA,0,0,0,0,NA,NA,NA))
fit.c
source("plot.root.R")
op = par(mfrow = c(1,2))
plot.roots(NULL,polyroot(c(1,0, 0, 0, 0, -0.2070, -0.4179)),
           main="(B) roots of AR part, nonseasonal")

plot.roots(NULL,polyroot(c(1,0, 0, 0, 0, -0.1613, -0.2636, 0, 0, 0, -0.2396, 0.2853)),
           main="(C) roots of AR part, nonseasonal")

# residuals for model A
res.a <- residuals(fit.a)
par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(res.a,main = "Autocorrelation")
# pacf
pacf(res.a,main = "Partial Autocorrelation")
# Histogram
hist(res.a,main = "Histogram", prob=TRUE)
m <- mean(res.a)
std <- sqrt(var(res.a))
curve(dnorm(x,m,std), add=TRUE)
# q-q plot
qqnorm(res.a)
qqline(res.a,col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics", outer=TRUE)

ts.plot(res.a, gpars=list(xlab="Year", ylab = "Residuals"))
abline(lm(res.a ~ seq(2011,2019, length.out = 95)),col = "red")
abline(h = 0, col = "blue")

# residuals for model B
res.b <- residuals(fit.b)
par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
```

```r
# acf
acf(res.b,main = "Autocorrelation")
# pacf
pacf(res.b,main = "Partial Autocorrelation")
# Histogram
hist(res.b,main = "Histogram", prob=TRUE)
m <- mean(res.b)
std <- sqrt(var(res.b))
curve(dnorm(x,m,std), add=TRUE)
# q-q plot
qqnorm(res.b)
qqline(res.b,col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics", outer=TRUE)
ts.plot(res.b, gpars=list(xlab="Year", ylab = "Residuals"))
abline(lm(res.b ~ seq(2011,2019, length.out = 95)),col = "red")
abline(h = 0, col = "blue")

# residuals for model C
res.c <- residuals(fit.c)
par(mfrow=c(1,2),oma=c(0,0,2,0))
# Plot diagnostics of residuals
op <- par(mfrow=c(2,2))
# acf
acf(res.c,main = "Autocorrelation")
# pacf
pacf(res.c,main = "Partial Autocorrelation")
# Histogram
hist(res.c,main = "Histogram", prob=TRUE)
m <- mean(res.c)
std <- sqrt(var(res.c))
curve(dnorm(x,m,std), add=TRUE)
# q-q plot
qqnorm(res.c)
qqline(res.c,col ="blue")
# Add overall title
title("Fitted Residuals Diagnostics", outer=TRUE)
ts.plot(res.c, gpars=list(xlab="Year", ylab = "Residuals"))
abline(lm(res.c ~ seq(2011,2019, length.out = 95)),col = "red")
abline(h = 0, col = "blue")

# Shapiro-Wilk test for Model(A)
shapiro.test(res.a) %>% pander()

# Box-Pierce for Model(A)
Box.test(res.a, lag = 10, type = c("Box-Pierce"), fitdf = 2) %>% pander()

# Ljung-Box test for Model(A)
Box.test(res.a, lag = 10, type = c("Ljung-Box"), fitdf = 2) %>% pander()

# McLeod-Li test for Model(A)
Box.test(res.a^2, lag = 10, type = c("Ljung-Box"), fitdf = 0) %>% pander()
```

```
# Yule-Walker for Model(A)
ar(res.a, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Shapiro-Wilk test for Model(B)
shapiro.test(res.b) %>% pander()

# Box-Pierce for Model(B)
Box.test(res.b, lag = 10, type = c("Box-Pierce"), fitdf = 3) %>% pander()

# Ljung-Box test for Model(B)
Box.test(res.b, lag = 10, type = c("Ljung-Box"), fitdf = 3) %>% pander()

# McLeod-Li test for Model(B)
Box.test(res.b^2, lag = 10, type = c("Ljung-Box"), fitdf = 0) %>% pander()

# Yule-Walker for Model(B)
ar(res.b, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# Shapiro-Wilk test for Model(C)
shapiro.test(res.c) %>% pander()

# Box-Pierce for Model(C)
Box.test(res.c, lag = 10, type = c("Box-Pierce"), fitdf = 4) %>% pander()

# Ljung-Box test for Model(C)
Box.test(res.c, lag = 10, type = c("Ljung-Box"), fitdf = 4) %>% pander()

# McLeod-Li test for Model(C)
Box.test(res.c^2, lag = 10, type = c("Ljung-Box"), fitdf = 0) %>% pander()

# Yule-Walker for Model(C)
ar(res.c, aic = TRUE, order.max = NULL, method = c("yule-walker"))

# forcast before transformation
f<- arima(employ_log, order=c(6,1,0),
          seasonal = list(order = c(0,1,1), period = 12),
          fixed = c(0,0,0,0,NA,NA,NA), transform.pars = FALSE,
          method="ML")
#forecast(f)

pred.tr <- predict(f, n.ahead = 13)
U.tr = pred.tr$pred + 2*pred.tr$se
L.tr = pred.tr$pred - 2*pred.tr$se

ts.plot(employ_log, xlim=c(2010,2020), ylim = c(min(employ_log),max(U.tr)),
        xlab="Year",
        ylab="Total number of Nonfram Employees after log-Transformation",
        main="forecasting for log-tranformed data")

lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points(pred.tr$pred, col="red", pch = 1, cex = 0.5)
```

```r
# transformed back
pred.orig <- exp(pred.tr$pred)
U= exp(U.tr)
L= exp(L.tr)
ts.plot(employ_ts, xlim=c(2010,2020), ylim = c(min(employ_ts),max(U)),
        xlab="Year",
        ylab="Total number of Nonfram Employees",
        main="forecasting for original data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
# predicted values
points(pred.orig, col="red", pch = 1, cex = 0.5)
# true values
points(employ_ts_test, col="green",pch = 1, cex = 0.5)

legend("topleft", legend=c("predicted", "original"),
       col=c("green","red"), pch = 1, cex=0.8)

ts.plot(employ_ts_test, xlim=c(2019,2020), ylim = c(min(employ_ts_test),max(U)),
        xlab="Year",
        ylab="Total number of Nonfram Employees",
        main="forecasting for original data 2019-2020")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
# predicted values
points(pred.orig, col="red")
# true values
points(employ_ts_test, col="green")

legend("topleft", legend=c("predicted", "original"),
       col=c("green","red"), pch = 1, cex=0.8)
fit.i
fit.i1 <- arima(ddemploy, order = c(1,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE)
AICc(fit.i1)
fit.i1
fit.i1 <- arima(ddemploy, order = c(0,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE)
AICc(fit.i1)
fit.i1
# third model summary
fit.iii
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
               fixed = c(NA,0,0,0,NA,NA,NA,NA,NA))
AICc(fit.iii1)
fit.iii1
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
               seasonal = list(order = c(1,0,1),period = 12),
               method = "ML", transform.pars = FALSE,
```

```
                fixed = c(0,0,0,0,NA,NA,NA,NA,NA))
AICc(fit.iii1)
fit.iii1
fit.iii1 <- arima(ddemploy, order = c(6,0,0),
                seasonal = list(order = c(0,0,1),period = 12),
                method = "ML", transform.pars = FALSE,
                fixed = c(0,0,0,0,NA,NA,NA,NA))
AICc(fit.iii1)
fit.iii1
# forth model summary
fit.iv
fit.iv1 <- arima(ddemploy, order = c(12,0,0),
                method = "ML", transform.pars = FALSE,
                fixed = c(0,0,NA,0,NA,NA,NA,0,0,0,NA,NA,NA))
AICc(fit.iv1)
fit.iv1
fit.iv1 <- arima(ddemploy, order = c(12,0,0),
                method = "ML", transform.pars = FALSE,
                fixed = c(0,0,0,0,NA,NA,0,0,0,0,NA,NA,NA))
AICc(fit.iv1)
fit.iv1
```