# Time Series Analysis on Employment-Population Ratio

Author: Nathan Wu

Advisor: Raya Feldman

Teaching Assistant: Sunpeng Duan

University of California, Santa Barbara

June 4, 2020

Table of Contents

**Abstract**

In this report, we analyze the Employment-Population Ratio data from the Federal Reserve Bank of St. Louis[1]. We hope to better understand the labor market in the United States and predict its future trends. Our approaches to this task are mainly based on the Box-Jenkins method. We discover that the US Employment-Population Ratio shows a strong seasonality at lag = 12 and has an up-ward trend. To make the dataset suitable for the Box-Jenkins method analysis, we detrend the data and apply Box-Cox transformation. By observing the ACF and PACF of the transformed data we propose several models for further analysis. We measure the proposed models with their AICC values, their invertibility and stationarity, and whether their residuals resemble a Gaussian White Noise. At the end of the report, we propose SARIMA $(1,1,0)\times(1,1,1)_{12}$ as our final model. It passes all the residual diagnostic checks[2]. In the forecasting step, the values in the test set fall within our prediction intervals. Therefore, we conclude that the final model we proposed is a good fit to the dataset, and it is possible to predict the US labor market of the near future by using previous datas.

*Keywords*: Box-Jenkins, SARIMA, Labor Market

[1] U.S. Bureau of Labor Statistics, Employment-Population Ratio [LNU02300000], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/LNU02300000, May 30, 2020.

[2] The residual diagnostic checks we ran include: Shapiro-Wilk normality test, Box-Pierce test, Box-Ljung test, Mc-Leod Li test and Yule-Walker estimation on residuals.

## Introduction and Motivation

"Employment-Population Ratio is a macroeconomic statistic that measures the civilian labor force currently employed against the total working-age population of a region, municipality, or country".[3] "This ratio is used to evaluate the ability of the economy to create jobs. Having a higher ratio means that a larger proportion of the population is employed, which in general will have positive effects on the GDP per capita".[4] In this report, we hope to use statistics analytical tools to form a better understanding on the properties of this ratio and make predictions based on the currently accessible data. In figure 1, we have an overview of the Employment-Population Ratio in the last 70 years.

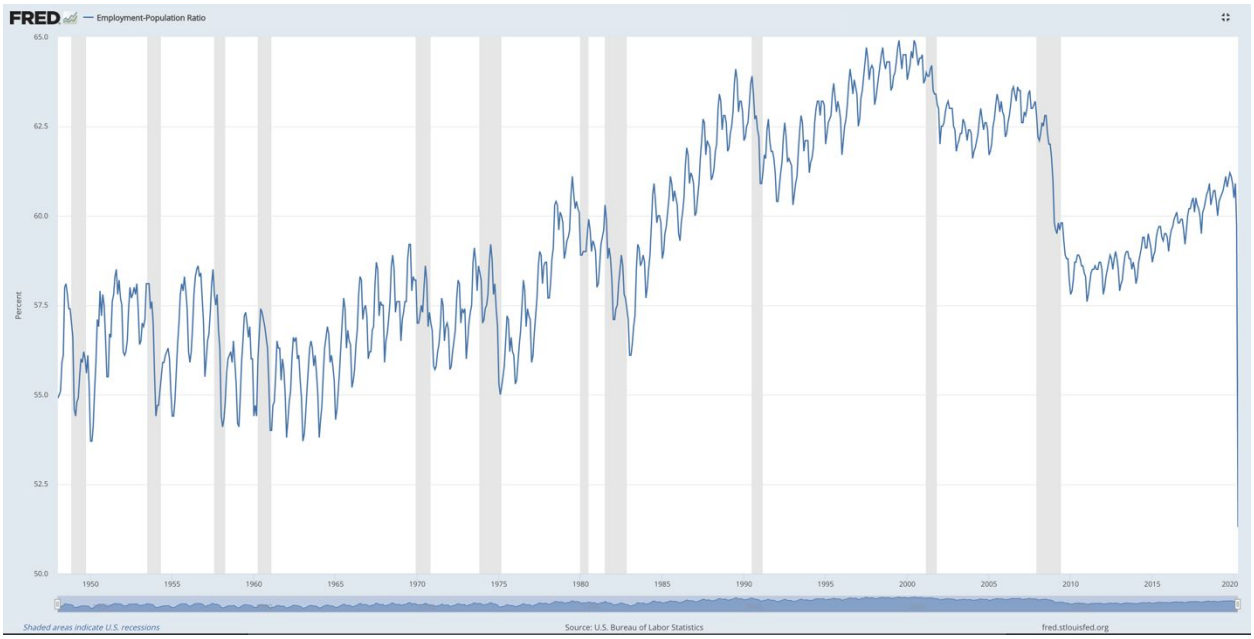


Figure 1. Employment-Population Ratio since 1950

[3] Employment to Population Ratio, Investopedia, https://www.investopedia.com/terms/e/employment_to_population_ratio.asp

[4] Employment-to-population ratio, Wikipedia, https://en.wikipedia.org/wiki/Employment-to-population_ratio

## Data Selection and Pre-processing[5]

From Figure 1. We can observe a clear seasonality and an upward trend starting from 1950. However, we notice that there's a sharp decrease happening in 2008 due to the global financial crisis, which introduces a certain level of volatility to our data. Besides that, the drastic fall in the Employment-Population Ratio at the beginning of 2020, caused by the Covid-19 global pandemic, would make the data unsuitable for building a time series model with Box-Jenkins method. Therefore, we choose a relatively stable segment of the data for this project. Considering the data exhibits a seasonality, we want to choose a segment that consists of at least 120 data points. Therefore, we pick the Employment-Population Ratio from 1981-04-01 to 1998-11-01, which consists of 212 data points. For model evaluation purposes, we split the data into two sets: the training set and the testing set. The last 12 data points will go into the testing set. The selected data is plotted in Figure 2.

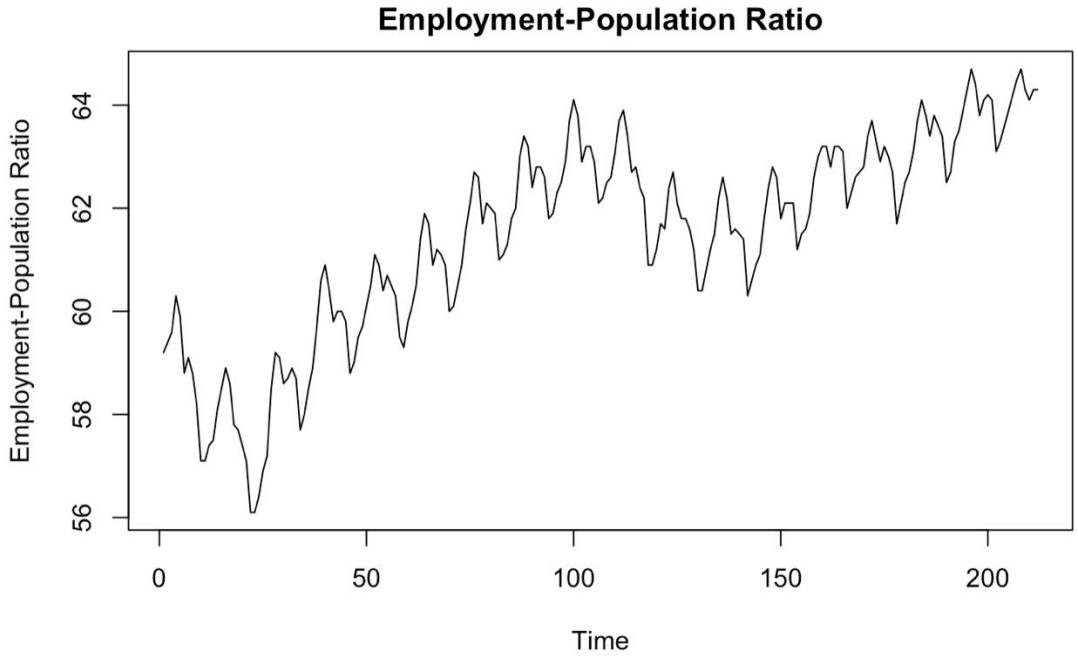


Figure 2. Selected Employment-Population Ratio

[5] Clicking on the title will take you to the corresponding code section in the Appendix.

## Box-Cox Transformation

In Feature 2, we can observe that the variance is not consistent throughout times, which means we might need to transform the data. To further investigate whether a transformation is necessary, we plot out its ACF and its histogram. From the ACF graph in Figure 3, we see the ACF values remain large even after lag 20, and it presents visible seasonalities at lag = 12. We also observe that the data is Left-Skewed by looking at the histogram of the data in Figure 3. Therefore, we decide to apply Box-Cox transformation[6] to our data.

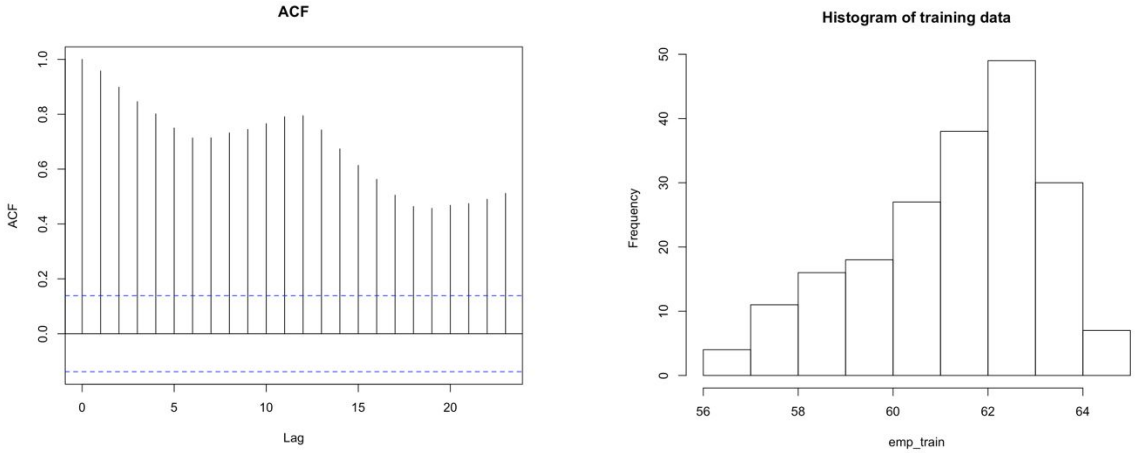


Figure 3. ACF and Histogram of Employment-Population Ratio

Because our dataset consists of all positive values, there is no other processing needed on the data (i.e. adding a positive constant to ensure all values are positive). The Box-Cox transformation test result is shown in Figure 4. We take the suggested $\lambda$ value 9.2 to transform our data into $Y = \frac{X^{9.2}-1}{9.2}$. We then compare the pre-transformation data's histogram and plots with post-transformation ones shown in Figure 5 and Figure 6. It can be found that the

[6] Box-Cox transformation is a statistics method to transform non-normal data to more normal-like. It only works with positive values, therefore sometimes you might need to add a positive constant to your dataset to make your data positive.

transformed data has a more symmetrical histogram and more even variance. The plot of the post-transformation data shown in Figure 6 also suggests a more stable variance than before transformation. Therefore, we decide that the transformation is necessary to make the data more stationary, and we should proceed our analysis with the transformed data.

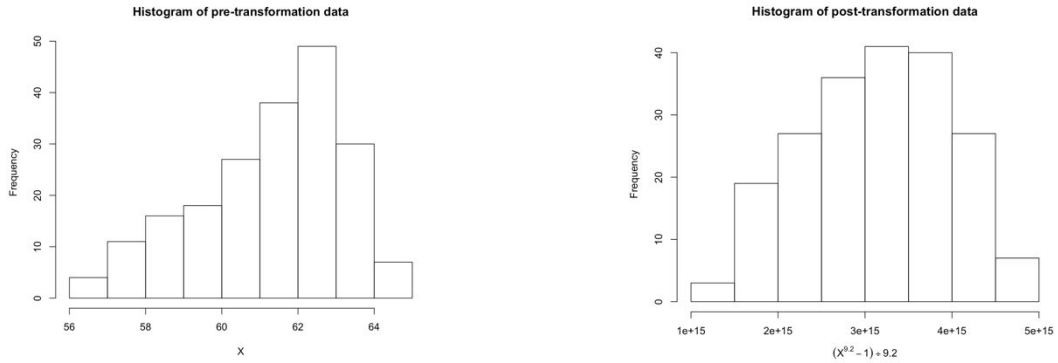


Figure 5. Histogram of Pre-transformation (left) data and Post-transformation data (right)

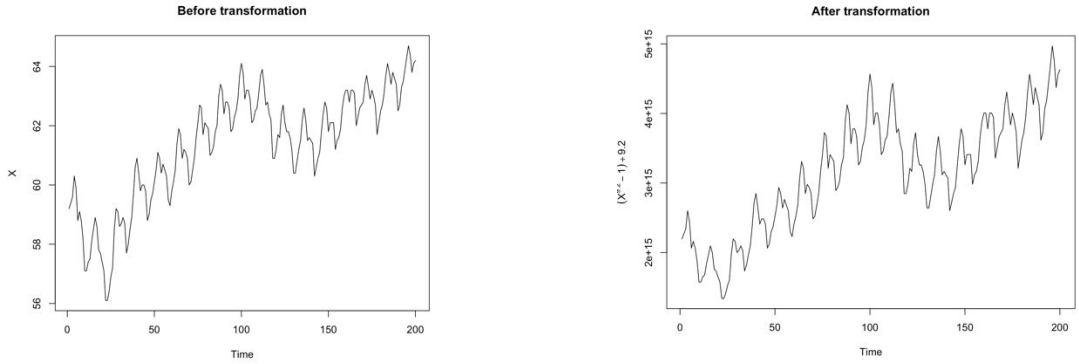


Figure 6. Plot of Pre-transformation (left) Data and Post-transformation Data (right)

## Difference the Transformed Data

We notice a clear trend and seasonality in the transformed data by observing the post-transformation plot in Figure 6. To remove the trend, we take a difference at lag = 1. The variance decreased from 6.822024e+29 to 5.221342e+28 after the differencing. We therefore continue with the differenced data $\nabla_1 Y$ and take another difference at lag = 12 to eliminate the

seasonality. The variance further decreased to 1.085374e+28. At this point of analysis, after applying differencing twice, our data becomes $\nabla_1 \nabla_{12} Y$. To better observe the effects of differencing on our data and to decide if more differencing is needed, we plot out the differenced data and its histogram in Figure 7. Looking at the plot of $\nabla_1 \nabla_{12} Y$, we find it looks more stationary, and the histogram looks less heavily tailed, more symmetrical and more similar to a Gaussian distribution. We therefore conclude that these two differencings are sufficient and proceed to identifying the models with the differenced data.

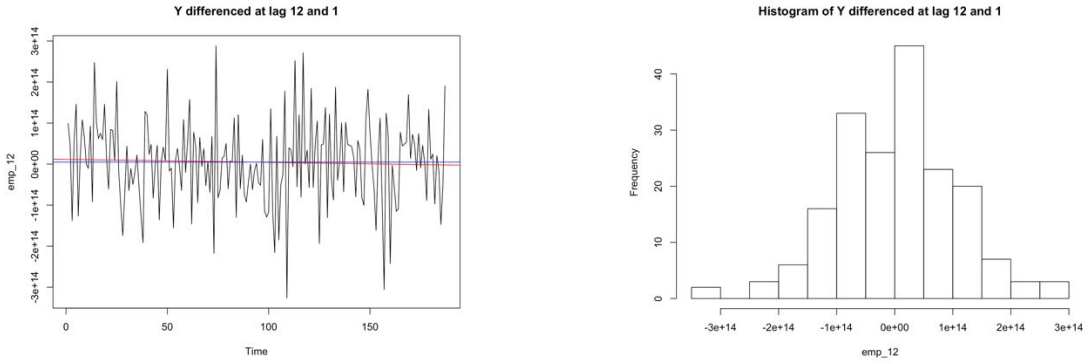


Figure 7. Plot and Histogram of Differenced Data

## Model Identification and Selection

*Model Identification*

To identify the model, we need to look at the differenced data's ACF and PACF as shown in Figure 8. The ACF values give us information about the Moving Average part of our model. The PACF values reveal information about the Autoregressive part of our model. To estimate the p and q values, we should focus on the first 12 lags and find out at which lags the ACF or PACF values stick out of the 95% confidence interval. For P and Q values estimation, we should look for the significant ACF and PACF values at the multiples of lag 12. As we can see in the ACF

graph in Figure 8, for the first 12 lags, only at lag = 1 the ACF is slightly out of the confidence interval, which indicates **q should be equal to 1 or 0**. We then look at the multiples of lag 12 in the ACF graph. We can only see a significant ACF value at lag = 12, which means **Q should be equal to 1**. For the PACF graph, we take a similar process by first estimating the p value and then the P value. In the first 12 lags, we can only observe a significant PACF value at lag = 1. Thus, **we estimate p to be 1**. For the multiples of lag 12, we can see a clear significance of PACF values at lag = 12 and lag = 24, and also notice that at lag = 36 the PACF value is slightly out of confidence interval. Therefore, **we propose P = 2 or 3 as our estimation**.

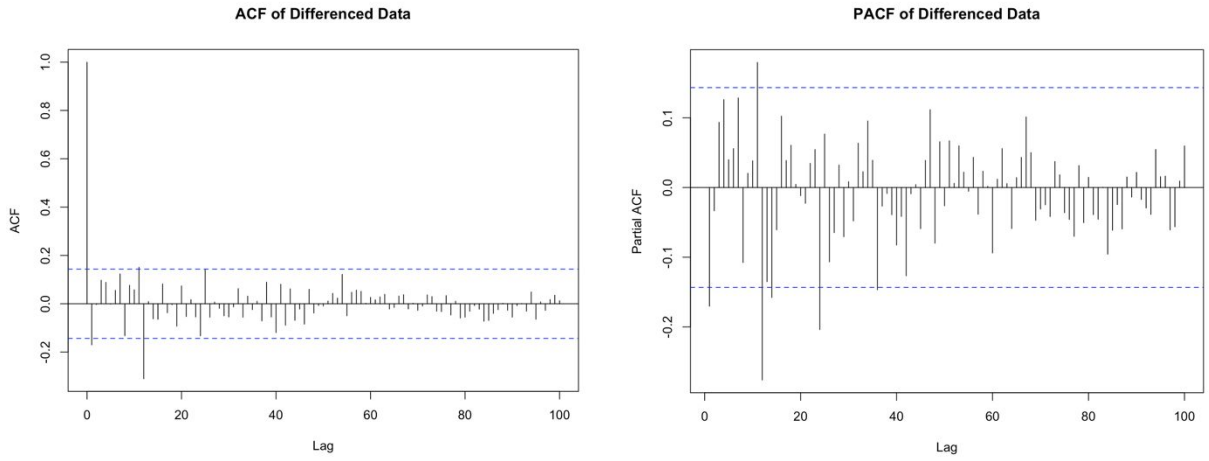


Figure 8. ACF and PACF of Differenced Data

With all the information together, we propose four possible SARIMA models:

Model 1: SARIMA $(1, 1, 1)\times(3, 1, 1)_{12}$

Model 2: SARIMA $(1, 1, 0)\times(3, 1, 1)_{12}$

Model 3: SARIMA $(1, 1, 1)\times(2, 1, 1)_{12}$

Model 4: SARIMA $(1, 1, 0)\times(2, 1, 1)_{12}$

*Model Selection*

We first create estimated models with proposed *p, q, P* and *Q* values using Maximum Likelihood method, then calculate the AICC values for each fitted model. The AICC statistics is defined as $AICC = -2\ln L(\underline{\theta}_q, \underline{\phi}_p, S(\underline{\theta}_q, \underline{\phi}_p)/n) + 2(p + q + 1)n/(n - p - q - 2)$. With fixed *p* and *q*, we want to find the coefficients $\underline{\theta}_q$ and $\underline{\phi}_p$ that minimize the AICC value. During the computation, Model 1 results in a non-finite finite difference error, we therefore only consider the last three models. The AICC scores of these models are 12568.14 for Model 2, 12561.88 for Model 3, and 12559.81 for Model 4. We choose the two models with the lowest AICC scores, Model 3 and Model 4, to proceed our analysis with. By printing out their coefficient values, we find that Model 3 contains an *NA* valued coefficient, thus we discard it and proceed with Model 4. In Model 4, the coefficient of sar2 is -0.0694 with standard error 0.0913, which indicates 0 is within its confidence interval and sar2 might be insignificant to this model. To decide whether we should keep sar2 or not, we fit a new model with sar2 removed and compare the new AICC score with the old one. The AICC score after removing sar2 term drops from 12559.81 to 12558.3. Therefore, we update our model to SARIMA $(1, 1, 0)\times(1, 1, 1)_{12}$. After the update, we find that the coefficient of ar1 term also has 0 within its confidence interval, however, the AICC score increases a little as we remove ar1 (with ar1: 12558.3, without ar1: 12558.85). With regard to the principle of parsimony, despite that the AICC score increases after dropping one term, we decide to keep both models, Model A: **SARIMA $(0, 1, 0)\times(1, 1, 1)_{12}$** and Model B: **SARIMA $(1, 1, 0)\times(1, 1, 1)_{12}$**, to move on to the invertibility and stationarity check. To ensure a model is invertible and stationary, we want all of its roots to be outside of the unit circle. For the two models we pick, the SARIMA $(0, 1, 0)\times(1, 1, 1)_{12}$ has coefficients $|\Phi_1 = 0.2774| < 1$, $|\Theta_1 =$

-0.8695| < 1; and the SARIMA $(1, 1, 0)\times(1, 1, 1)_{12}$ has $|\phi_1 = 0.2774| < 1$, $|\Phi_1 = 0.2980| < 1$, $|\Theta_1 = -0.8736| < 1$. Both of them pass the stationarity and invertibility check and are ready for the model diagnosis.

## Model Diagnosis

Model diagnosis is as important as fitting the models, because it tells us whether our model is a good representation of the data and whether the model fits our data well enough. Even if the diagnosis result indicates that our model is not a good fit, it could still give valuable information to guide us to adjust our models. The key idea behind diagnostic checking is that the residuals of a well fitted model should resemble the Gaussian White Noise. Therefore, we start by plotting out the residuals, its histograms and Normal Q-Q plot to have an overview of the fitting. In Figure 9 and Figure 10, we present those plots of Model A and Model B respectively.

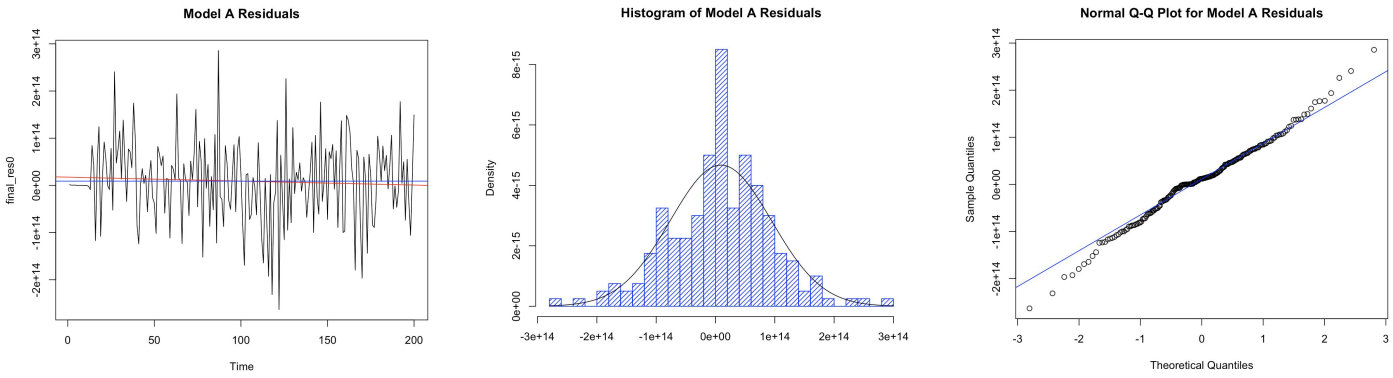


Figure 9. Diagnostic Plot of Model A Residuals

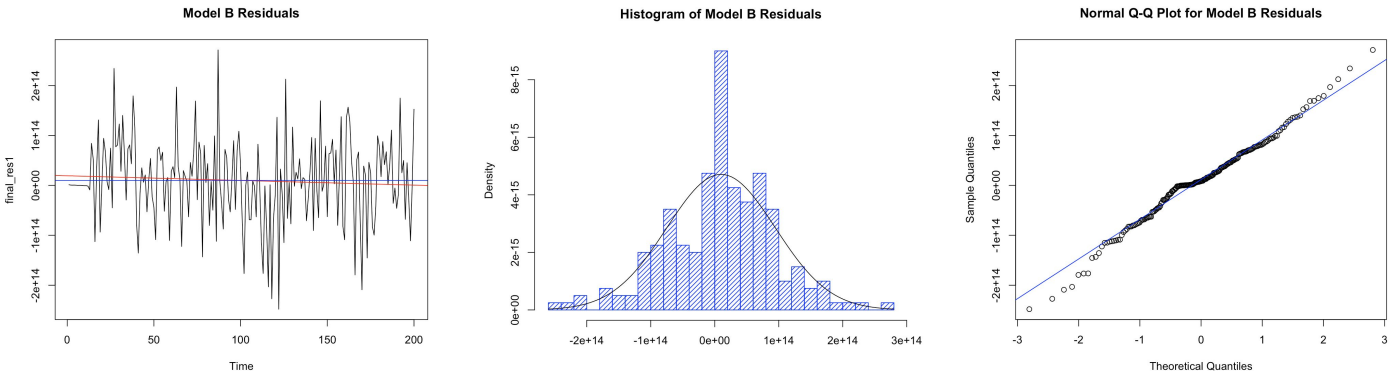


Figure 10. Diagnostic Plot of Model B Residuals

As we can we see form Figure 9 and Figure 10, the plots of both models' residuals show no trend, no seasonality and no change of variance. Except their Normal Q-Q plots showing signs of being a bit heavily tailed, their histogram looks fairly close to a Gaussian distribution. To have a closer look at the residual distributions, we plot the ACF and PACF of the residuals for both models. Since we are expecting the residuals to resemble Gaussian White Noise distributions, we hope to see the ACF and PACF values to be within the 95% confidence intervals for all lags (except for ACF at lag = 0, which should always be equal to 1).

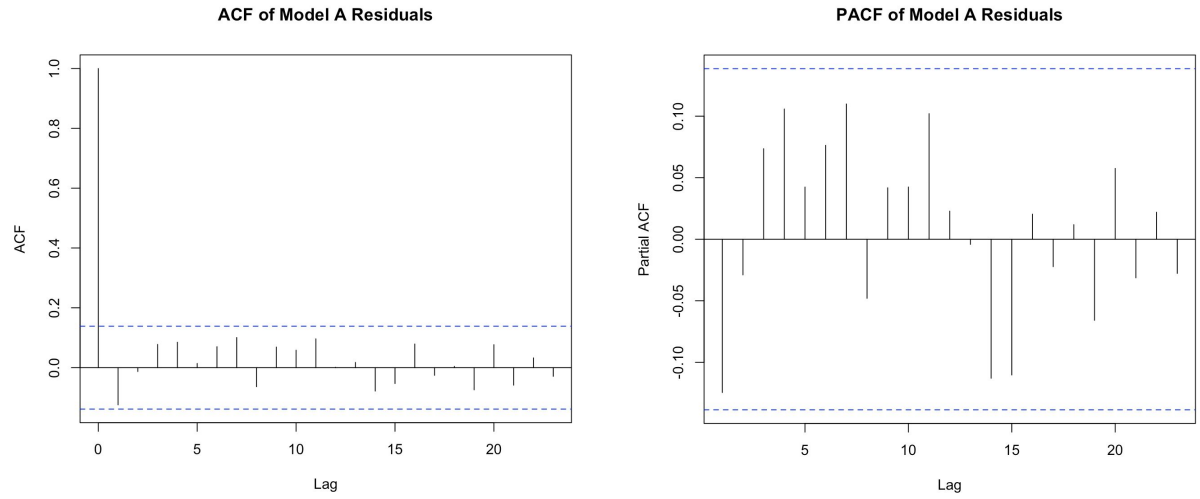


Figure 11. ACF and PACF of Model A Residuals

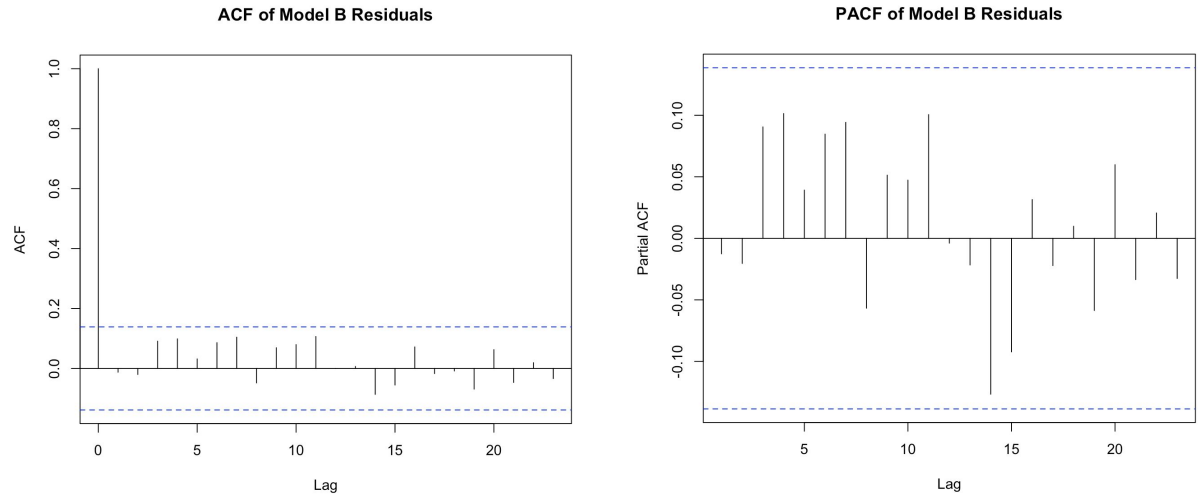


Figure 12. ACF and PACF of Model B Residuals

In Figure 11 and Figure 12, we can see that the residuals of both models have ACF and PACF values that fall nicely within the confidence interval. Therefore we can go ahead and apply the following test in statistics to further investigate the residuals:

**Shapiro-Wilk normality test:**

$H_0$: the data are normally distributed

**Box-Pierce test:**

$H_0$: The data are independently distributed

**Box-Ljung test:**

$H_0$: The data are independently distributed

**Mc-Leod Li test:**

$H_0$: there is no autoregressive conditional heteroskedasticity (ARCH) among the lags considered.

**Yule-Walker estimation and AICC statistics:**

It should give AR(0) as the estimated results.

The results of these tests are as follows:

| | Model A Residuals | Model B Residuals |
|---|---|---|
| **Shapiro-Wilk** | Shapiro-Wilk normality test<br><br>data: final_res0<br>W = 0.98912, p-value = 0.1328 | Shapiro-Wilk normality test<br><br>data: final_res1<br>W = 0.98696, p-value = 0.0629 |
| **Box-Pierce** | Box-Pierce test<br><br>data: final_res0<br>X-squared = 13.124, df = 10, p-value = 0.2168 | Box-Pierce test<br><br>data: final_res1<br>X-squared = 12.536, df = 10, p-value = 0.2508 |
| **Box-Ljung** | Box-Ljung test<br><br>data: final_res0<br>X-squared = 13.65, df = 10, p-value = 0.1896 | Box-Ljung test<br><br>data: final_res1<br>X-squared = 13.125, df = 10, p-value = 0.2168 |
| **Mc-Leod Li** | Box-Ljung test<br><br>data: final_res0^2<br>X-squared = 18.438, df = 12, p-value = 0.103 | Box-Ljung test<br><br>data: final_res1^2<br>X-squared = 15.358, df = 12, p-value = 0.2224 |
| **Yule-Walker** | Call:<br>ar(x = final_res0, aic = TRUE, order.max = NULL, method = c("yule-walker"))<br><br>Coefficients:<br>1<br>-0.1245<br><br>Order selected 1 sigma^2 estimated as 7.213e+27 | Call:<br>ar(x = final_res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))<br><br>Order selected 0 sigma^2 estimated as 7.175e+27 |

Based on the diagnostic results, we can see Model B passes all the tests. On the other hand, Model A seems to have an ar1 term with coefficient -0.1245. But if we look closer, we notice that 0 is within its 95% confidence interval, meaning that the ar1 term could be interpreted as insignificant. In the end, after taking the AICC score calculated previously into consideration, we decide to use Model B, which passes all the diagnostic checks and has a lower AICC score, as

our final model for predictions. The **final model** of our analysis, SARIMA $(1, 1, 0)\times(1, 1, 1)_{12}$, can be written as:

$$(1 + 0.1199_{(0.0740)}B)(1 - 0.2980_{(0.1061)}B^{12})(1 - B)(1 - B^{12})Y_t = (1 - 0.8736_{(0.0943)}B^{12})Z_t$$

## Forecasting and Reverser-Transformation

To further investigate the goodness of fit of our model, we use it to predict the values for the next 12 time steps, and compare the results with the testing set that we preserved at the beginning of our analysis. However, because we have applied Box-Cox Transformation at the start of our analysis, we need to do a reverse-transformation to transform the predicted values back to its original scale. Since we obtained the transformed data Y by following the formula $Y = \frac{X^{\lambda}-1}{\lambda}$, the reverse-transformation formula should be:

$$X = (\lambda Y + 1)^{\frac{1}{\lambda}}$$

Besides applying this reverser-transformation on the predicted values, we also need to apply it on the prediction interval. After the rescaling, we can finally compare the results with the real values in the testing set. If all the test values fall within the prediction interval, then we can conclude that our model is a good fit of the data. To compare the result, we draw both the predicted values and real values on the same graph as shown in Figure 13. The predicted values are represented as green circles; the real values are drawn as a red line; the dashed blue lines show the upper bound and lower bound of the prediction interval.

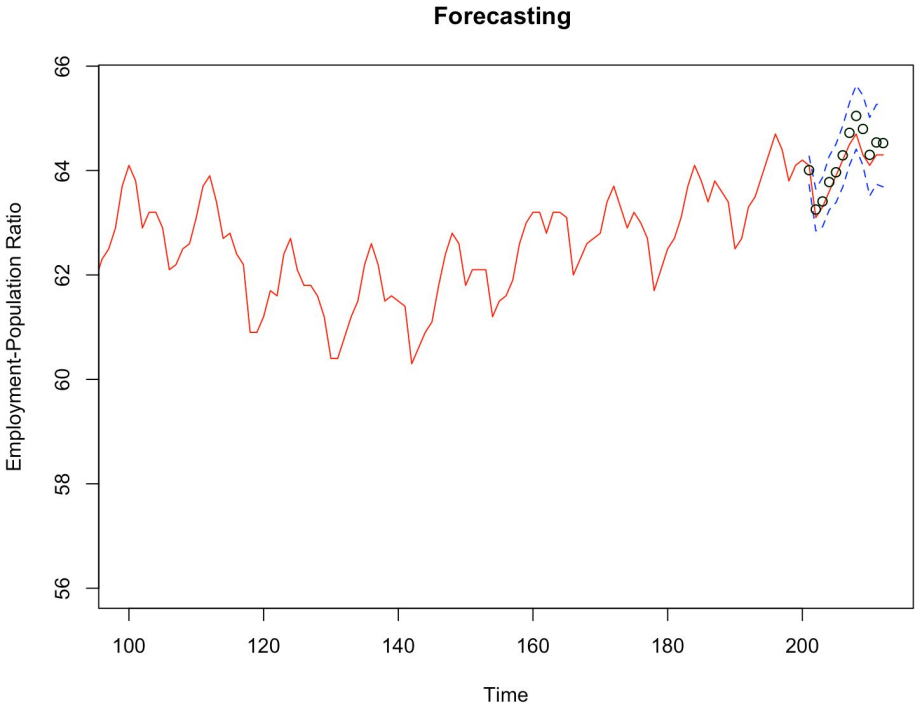


Figure 13. Predicted Values and Real Values

## Conclusion

As we can observe from Figure 13, the test set values fall nicely within the prediction interval. Therefore we can conclude that our model:

$$(1 + 0.1199_{(0.0740)}B)(1 - 0.2980_{(0.1061)}B^{12})(1 - B)(1 - B^{12})Y_t = (1 - 0.8736_{(0.0943)}B^{12})\,Z_t$$

is a good representation of the data, which means it is possible to use statistical tools, more specifically, Box-Jenkins methods, to predict the US labor market in the near future with the data from previous times. However, since in our analysis we only include data that are less volatile, we are not sure of the model's performance on describing datasets that are more unstable.

**Acknowledgement**

I would like to express my special thanks of gratitude to my PSTAT 174 instructor Professor Raya Feldman and my Teaching Assistant Sunpeng Duan for their guidance and support in helping me complete this project. I would also like to thank the Federal Reserve Bank of St. Louis for making this dataset public.

# Appendix

## Data Selection and Pre-processing

```
raw_csv <- read.csv("LNU02300000.csv")
head(raw_csv)
emp_rate <- raw_csv$LNU02300000
emp <- emp_rate[400:611]
emp_train <- emp_rate[400:599]
emp_test <- emp_rate[600:611]
```

## Box-Cox Transformation

```
library(MASS)
bcTransform <- boxcox(emp_train ~ as.numeric(1:length(emp_train)), lambda = seq(-10, 10, 1/10) )
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
emp_bc <- (1/lambda)*(emp_train^lambda-1)
lambda


plot.ts(emp_train, main = "Before transformation", ylab = "X")
plot.ts(emp_bc, main = "After transformation", ylab = expression(("X"^9.2-1)%/%9.2))
hist(emp_train, main = "Histogram of pre-transformation data", xlab = "X")
hist(emp_bc, main = "Histogram of post-transformation data", xlab = expression(("X"^9.2-1)%/%9.2))
```

## Difference the Transformed Data

```
emp_stat <- diff(emp_bc, lag = 1)
emp_12 <- diff(emp_stat, lag = 12)
var(emp_bc)
var(emp_stat)
var(emp_12)
hist(emp_12,  main = "Histogram of Y differenced at lag 12 and 1")
plot.ts(emp_12, main = "Y differenced at lag 12 and 1")
fitt <- lm(emp_12 ~ as.numeric(1:length(emp_12)));
abline(fitt, col="red")
abline(h=mean(emp_12), col="blue")
```

## Model Identification and Selection

```
arima_111_311 <- arima(emp_bc, order=c(1,1,1), seasonal = list(order = c(3,1,1), period = 12),
method="ML")
```

```
arima_110_311 <- arima(emp_bc, order=c(1,1,0), seasonal = list(order = c(3,1,1), period = 12),
method="ML")
arima_111_211 <- arima(emp_bc, order=c(1,1,1), seasonal = list(order = c(2,1,1), period = 12),
method="ML")
arima_110_211 <- arima(emp_bc, order=c(1,1,0), seasonal = list(order = c(2,1,1), period = 12),
method="ML")

library(qpcR)
AICc(arima_110_311)
AICc(arima_111_211)
AICc(arima_110_211)

arima_111_211
arima_110_211

arima_110_111 <- arima(emp_bc, order=c(1,1,0), seasonal = list(order = c(1,1,1), period = 12),
method="ML")

AICc(arima_110_211)
AICc(arima_110_111)

arima_110_111
arima_010_111 <- arima(emp_bc, order=c(0,1,0), seasonal = list(order = c(1,1,1), period = 12),
method="ML")

AICc(arima_110_111)
AICc(arima_010_111)

arima_010_111
```

## Model Diagnosis

```
final_res0 <- residuals(arima_010_111)
final_res1 <- residuals(arima_110_111)

plot.ts(final_res0, main = "Model A Residuals")
abline(fitt, col="red")
```

```
abline(h=mean(final_res0), col="blue")
fitt <- lm(final_res0 ~ as.numeric(1:length(final_res0)));
hist(final_res0,density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram of Model A Residuals")
m0 <- mean(final_res0)
std0 <- sqrt(var(final_res0))
curve( dnorm(x,m0,std0), add=TRUE )
qqnorm(final_res0,main= "Normal Q-Q Plot for Model A Residuals")
qqline(final_res0,col="blue")

plot.ts(final_res1, main = "Model B Residuals")
fitt <- lm(final_res1 ~ as.numeric(1:length(final_res0)));
abline(fitt, col="red")
abline(h=mean(final_res1), col="blue")
hist(final_res1,density=20,breaks=20, col="blue", xlab="", prob=TRUE, main = "Histogram of Model B Residuals")
m1 <- mean(final_res1)
std1 <- sqrt(var(final_res1))
curve( dnorm(x,m1,std1), add=TRUE )
qqnorm(final_res1,main= "Normal Q-Q Plot for Model B Residuals")
qqline(final_res1,col="blue")

acf(final_res0, main = "ACF of Model A Residuals")
pacf(final_res0, main = "PACF of Model A Residuals")

acf(final_res1, main = "ACF of Model B Residuals")
pacf(final_res1, main = "PACF of Model B Residuals")

shapiro.test(final_res0)
Box.test(final_res0, lag = 12, type = c("Box-Pierce"), fitdf = 2)
Box.test(final_res0, lag = 12, type = c("Ljung-Box"), fitdf = 2)
Box.test(final_res0^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(final_res0^2, lag.max=40)
ar(final_res0, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
shapiro.test(final_res1)
Box.test(final_res1, lag = 12, type = c("Box-Pierce"), fitdf = 2)
Box.test(final_res1, lag = 12, type = c("Ljung-Box"), fitdf = 2)
Box.test(final_res1^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
acf(final_res1^2, lag.max=40)
ar(final_res1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

## Forecasting and Reverse-Transformation

```
library(forecast)
forecast(arima_110_111)
pred.tr <- predict(arima_110_111, n.ahead = 12)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se

pred.orig <- (1 + lambda*pred.tr$pred)^(1/lambda)
U= (1 + lambda*U.tr)^(1/lambda)
L= (1 + lambda*L.tr)^(1/lambda)

ts.plot(emp, xlim = c(100,length(emp_train)+12), ylim = c(56,max(U)), col="red", main = "Forecasting ",
ylab = "Employment-Population Ratio")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="green")
points((length(emp_train)+1):(length(emp_train)+12), pred.orig, col="black")
```