Employment Outcomes for College Graduates

A Time Series Analysis

Anthony Cu

17 March 2025

Contents

Abstract

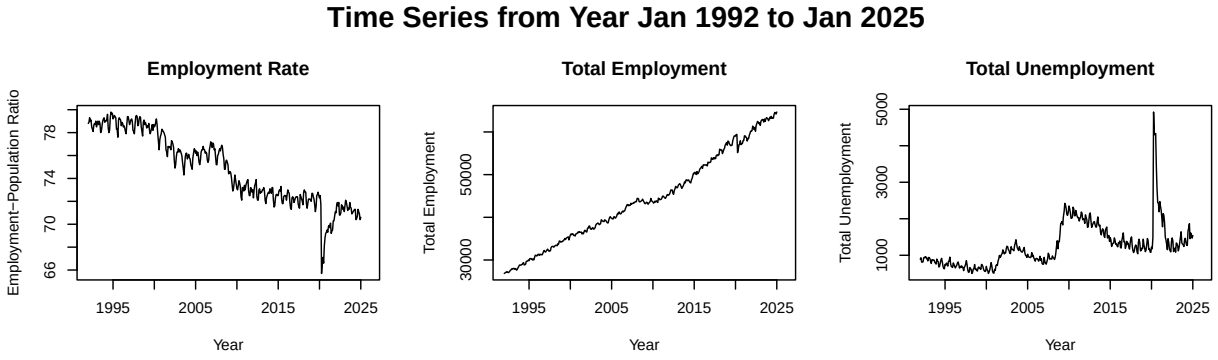
This project analyzes the total employment population of individuals with a bachelor's degree or higher in order to assess workforce trends and forecast employment. Using historical data from January 1992 to December 2018, I evaluate multiple time series models, including SAR, ARIMA, and SARIMA, and select the the optimal model based on AICc scores and diagnostic checking. The SARIMA$(4,1,0) \times (5,1,0)_{12}$ model demonstrates the best fit and is used to forecast total employment populations for 2019. The findings in this project provide insight into labor market trends for those with a bachelor's degree or higher, emphasizing how much of an impact a college degree has on employment. In addition, this project demonstrates the effectiveness of time series modeling in employment forecasting.

Introduction

Understanding employment trends for individuals with higher education is increasingly important, especially as technological advancements, economic shifts, and global events continue to reshape the job market. This project analyzes the total employment population for individuals with a bachelor's degree or higher, offering insight into how workforce opportunities have evolved over the past three decades and what the future may hold. As a soon-to-be college graduate, forecasting these trends is particularly relevant, providing a contextual picture of the job market I and many others will soon enter.

Dataset

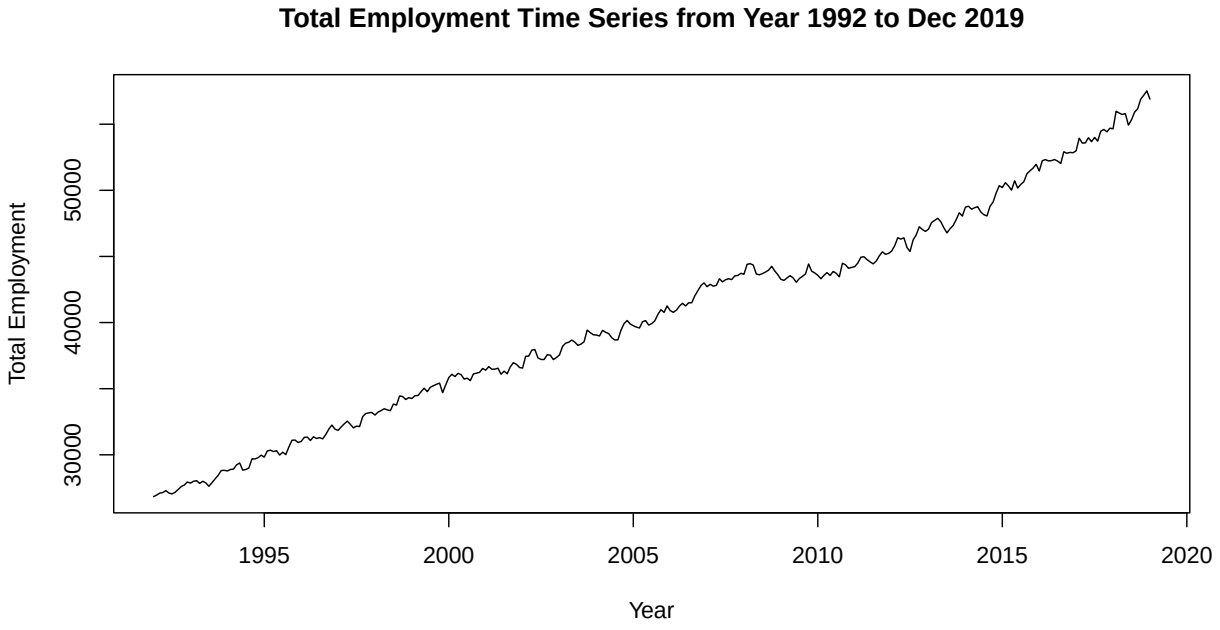
The data I will be using in this project is sourced from the U.S. Bureau of Labor Statistics. It includes monthly datasets on total employment, total unemployment, and the employment-population ratio for individuals (aged 25 and older) with a bachelor's degree or higher, spanning January 1992 to January 2025. These comprehensive datasets provide valuable insight into long-term employment trends for higher educated individuals.



By analyzing these datasets, we can notice the impact of major historical events, such as the 2008 financial crisis and the 2020 COVID-19 pandemic. The total unemployment time series shows the most significant spike following the 2008 crisis, reflecting a sharp peak in unemployment, which is also evident in the employment-population ratio time series plot by the slight downfall. Meanwhile,

the total employment population time series exhibits a slight bulge during this period, though the change is less pronounced. To ensure a more stable dataset with minimal contextual disruptions, I will focus on the **total employment** time series. In addition, to avoid disruptions caused by the COVID-19 crisis in 2020, I subset my dataset to exclude data beyond December 2019. By analyzing this dataset, I will develop an effective forecasting model for employment trends in 2019.

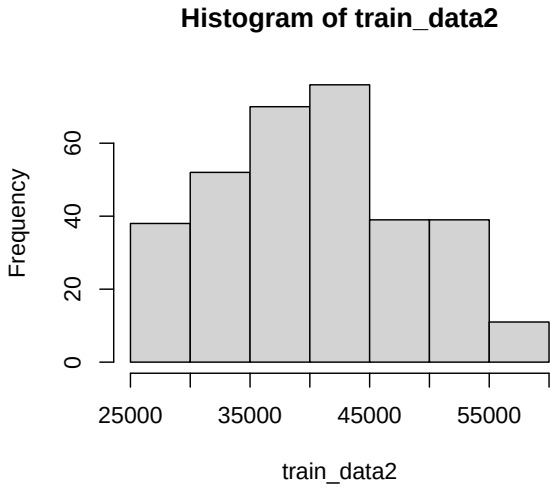
I will use the period from January 1992 to December 2018 as my *training set*, a length of 325 observations, in order to forecast total employment population for the year 2019 (in the months January to December)- the *testing set* of length 12. I will denote the training data as U_t.

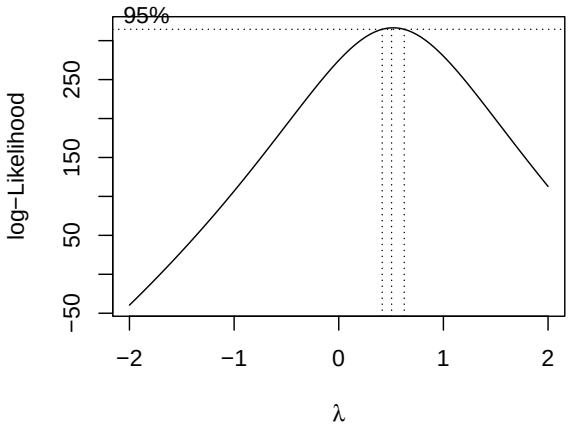


The dataset appears to have a linear trend, as total employment increases from 1992 to 2019. There may be seasonality based on the "M"-shaped pattern in each year.

Transformation of data

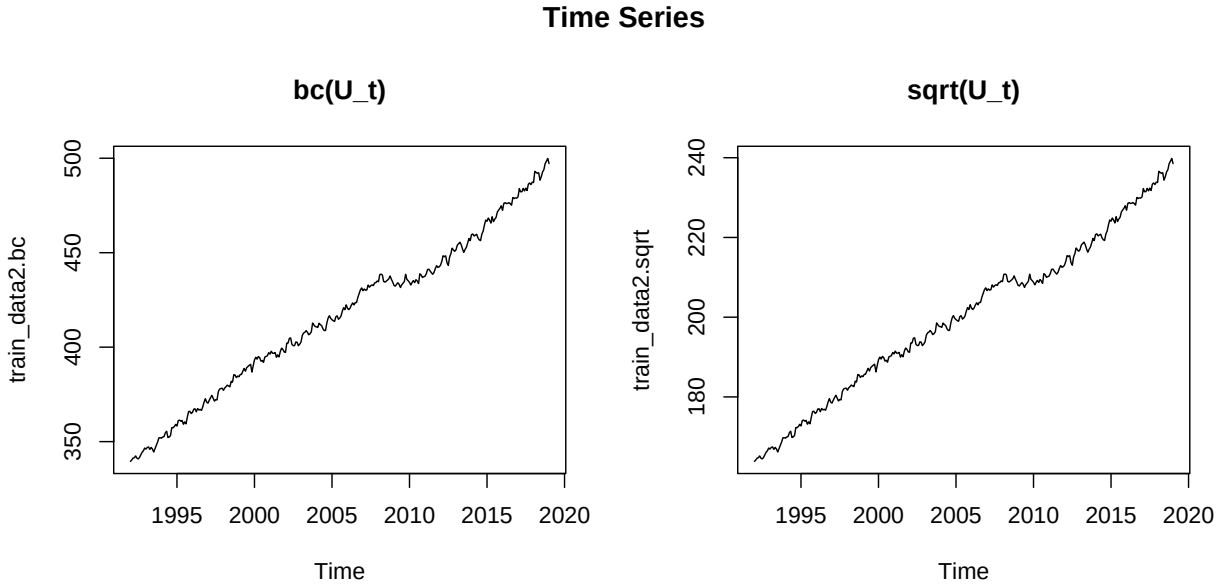
The Box-Cox transformation is applied to determine if any transformation is necessary on the data.

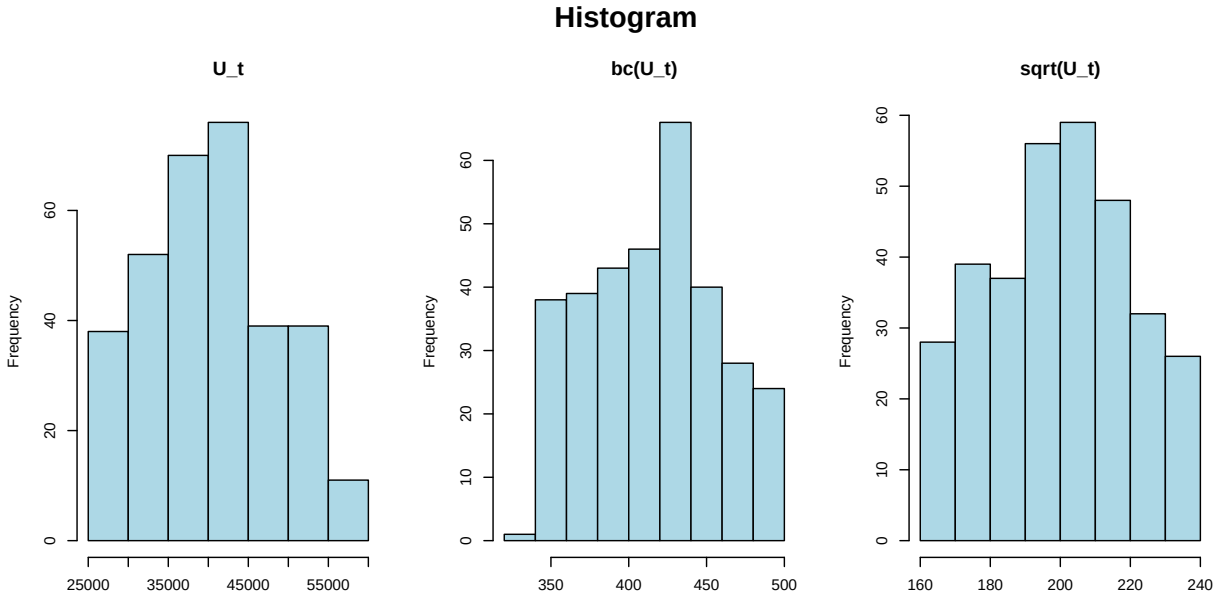





```
[1] lambda:  0.505050505050505
```

The histogram of our training data resembles a normal curve, but it is not a smooth distribution. Since the lambda value is approximately $1/2$, with 0 not contained in the confidence interval based on the plot above, I will also consider a square root transformation.



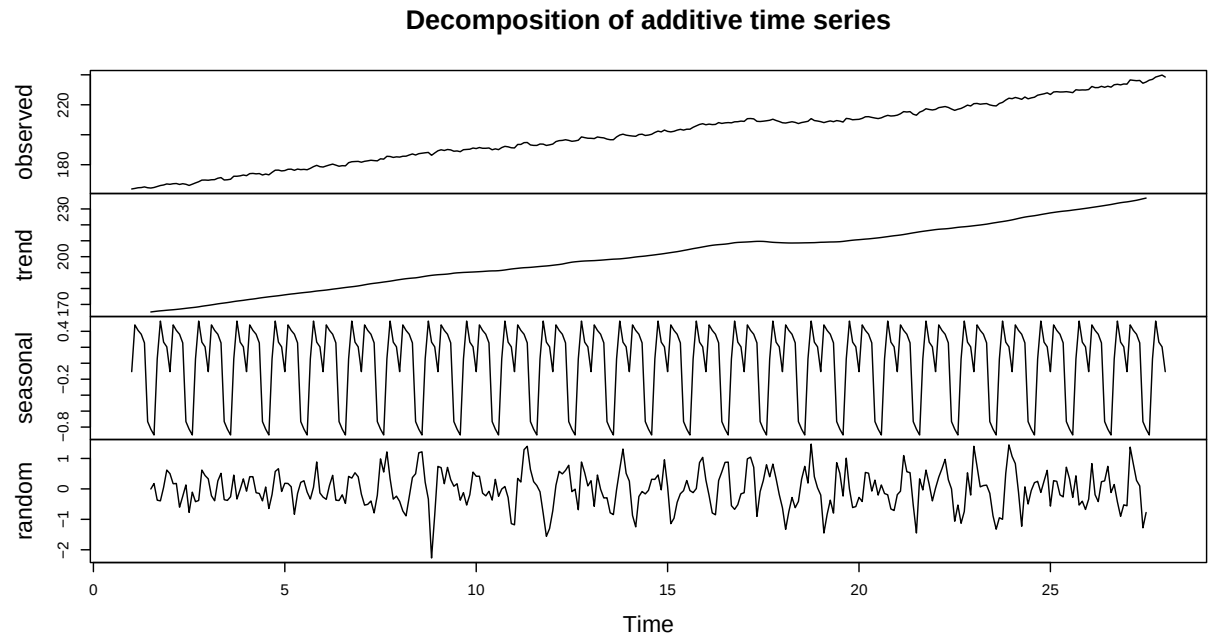


There is not much difference between the time series for the box cox transformation and square root transformation. Similar to the untransformed data, the time series for the transformed datasets still appear to have trend and seasonality.

Observing the histograms of the original, box cox transformed, and square root transformed, I notice that the square root transformation has the most normal, symmetric shape. Thus, I will proceed with the square root transformation.

Decomposition

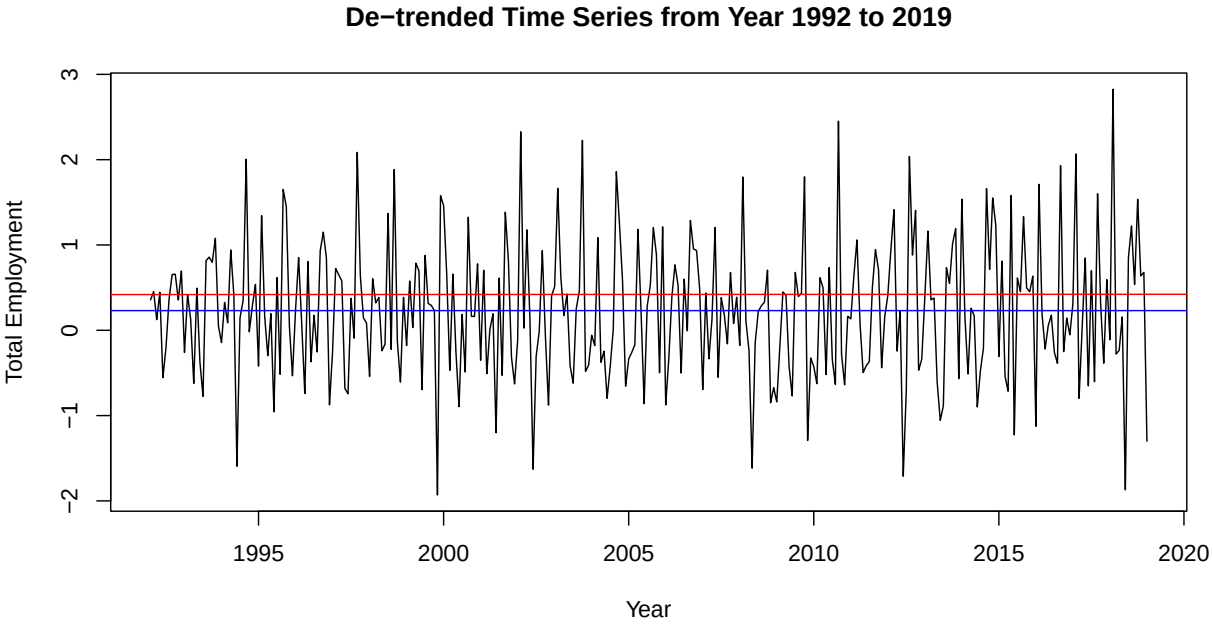
We decompose the transformed data to observe if there exists any trend or seasonality:



Looking at the decomposition of the data, I notice that there exists a linear trend and seasonal component. The random looks like White Noise when its removed of trend and seasonality.

Differencing of data

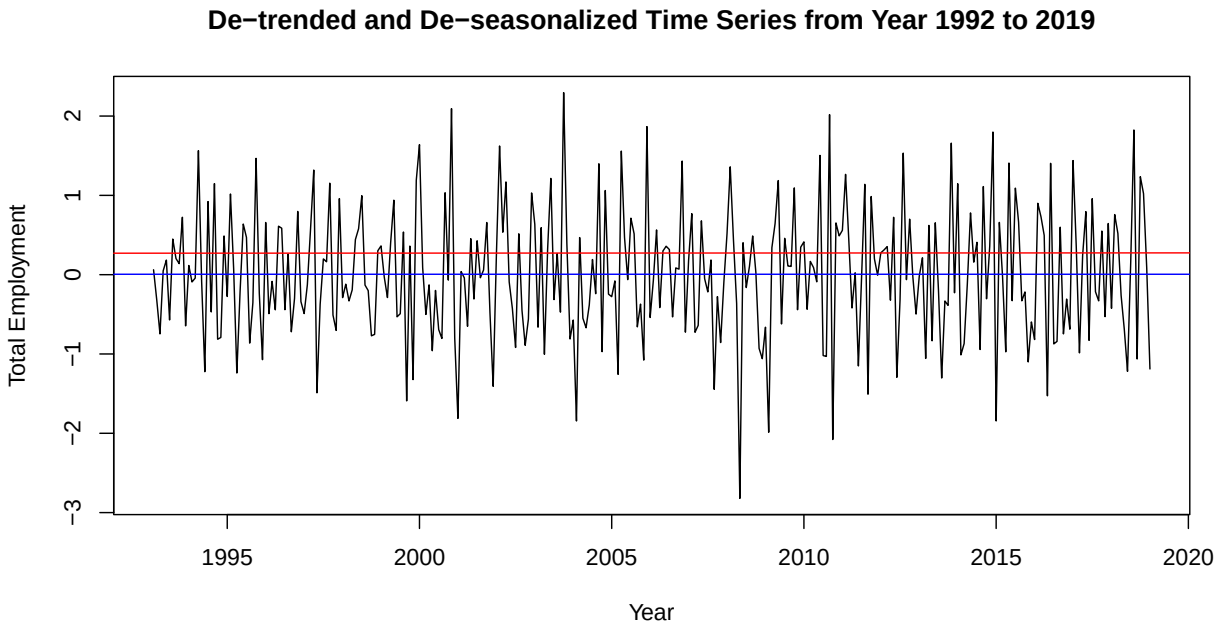
We begin by differencing our transformed dataset at lag 1 to remove trend.




```
[1] Mean:  0.230589188985589
```

```
[1] Variance:  0.626448985034235
```

By differencing at lag 1, we see that the dataset resembles White Noise, in which the positive trend is removed. The mean line (blue) closely resembles the fitted line (red). The de-trended time series appears stationary. We will consider differencing again at lag 12 to remove seasonality.

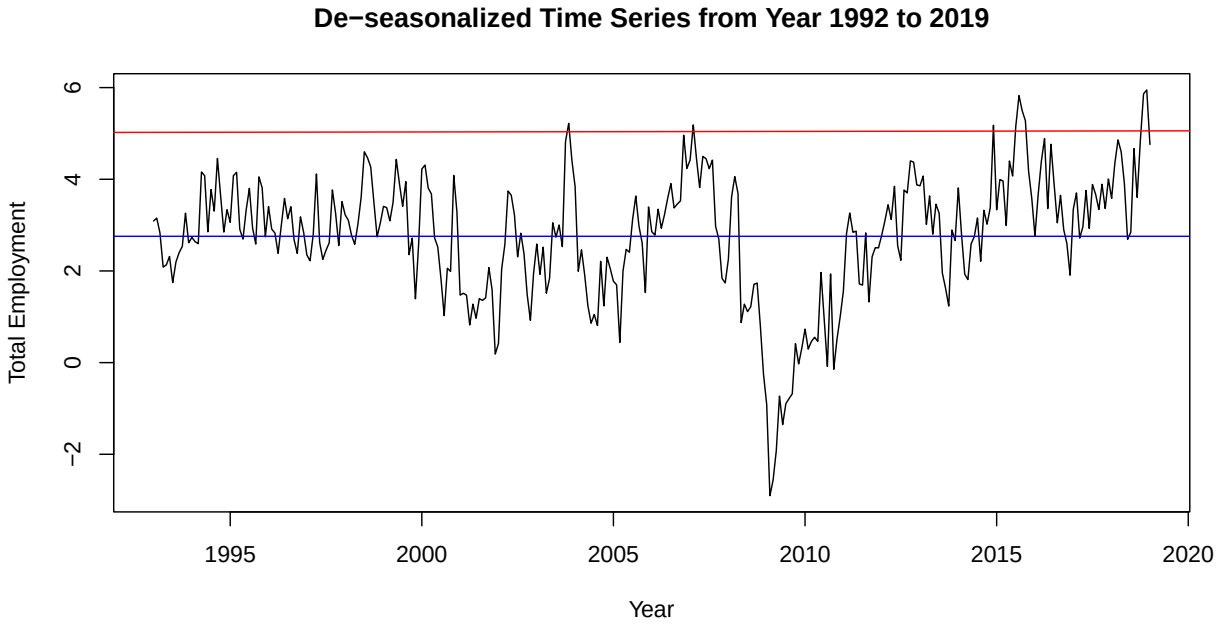



```
[1] Mean:  0.00535106548945371
```

```
[1] Variance:  0.669129362030567
```

The data resembles more White Noise again. However, the variance seems to increase from 0.626 to 0.669 slightly, so this could be a sign of overdifferencing. This time series also appears stationary.

So, we try only differencing at lag 12 (without any differencing at lag 1).



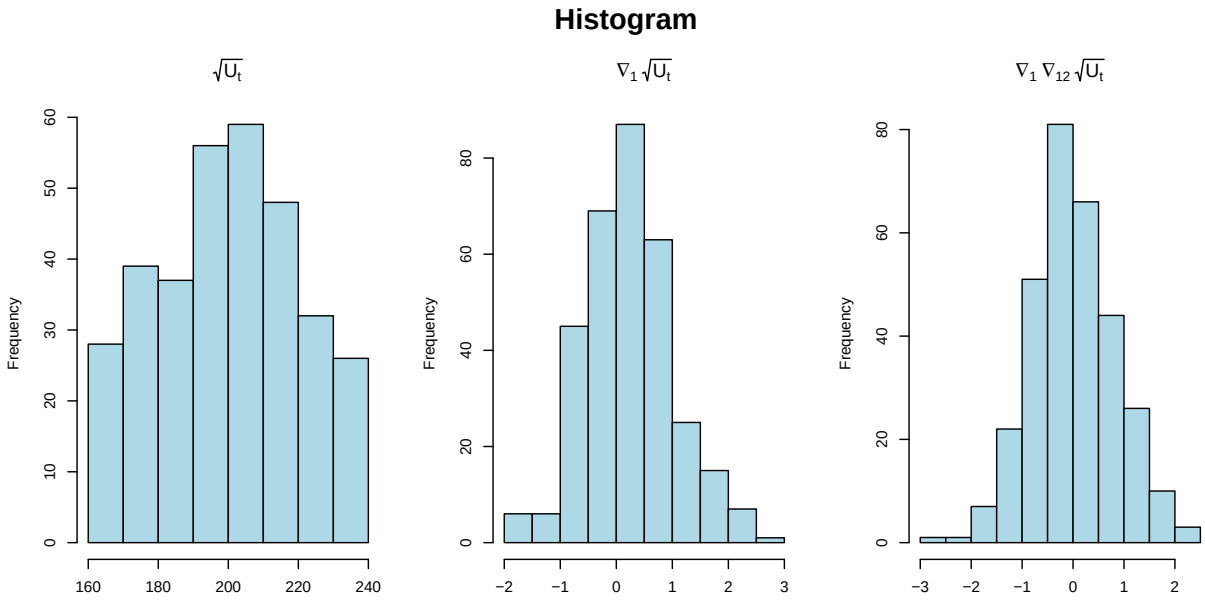

```
[1] Mean:  2.75584583909951
```

```
[1] Variance:  1.9050189171458
```

However, the time series does not appear as White Noise. The mean and fitted line are farther apart as well. In addition, the variance significantly increases. So, I will proceed with only examining either the differenced data at lag 1 only, or the differenced data at both lags 1 and 12 in order to identify potential models to fit.

Model Identification

We examine the histograms of our non-differenced data, de-trended data, and de-trended & de-seasonalized data

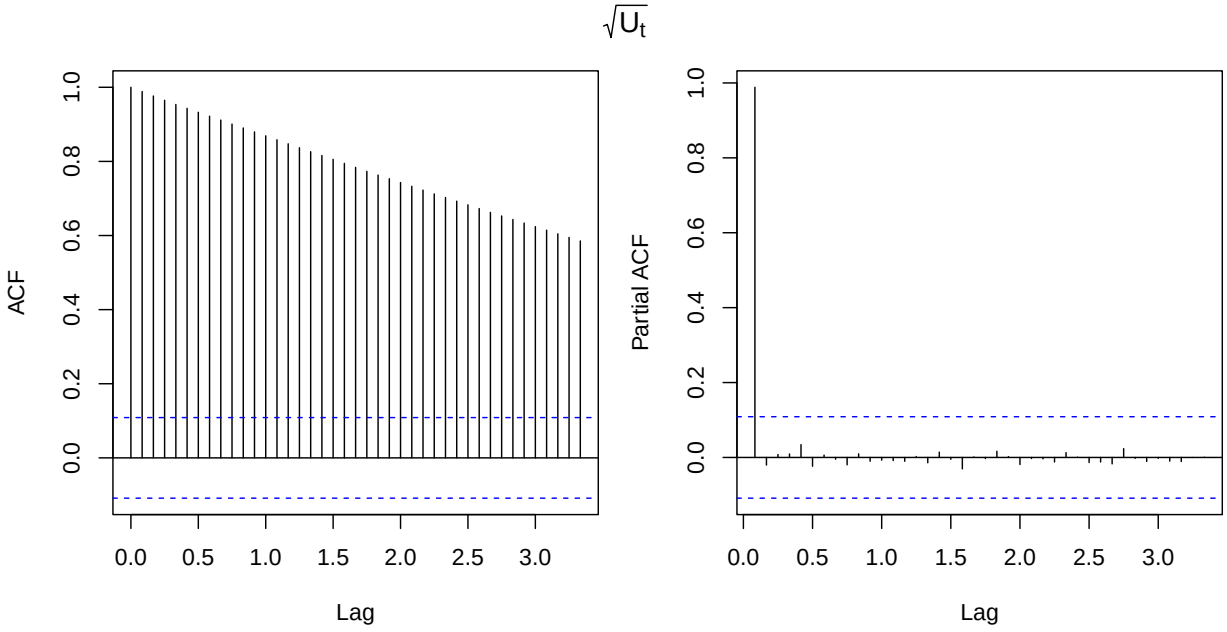


The de-trended & de-seasonalized data appears as the most normal, symmetric curve.

ACF and PACF

We proceed with plotting ACFs and PACFs to identify potential models.

Non-differenced data

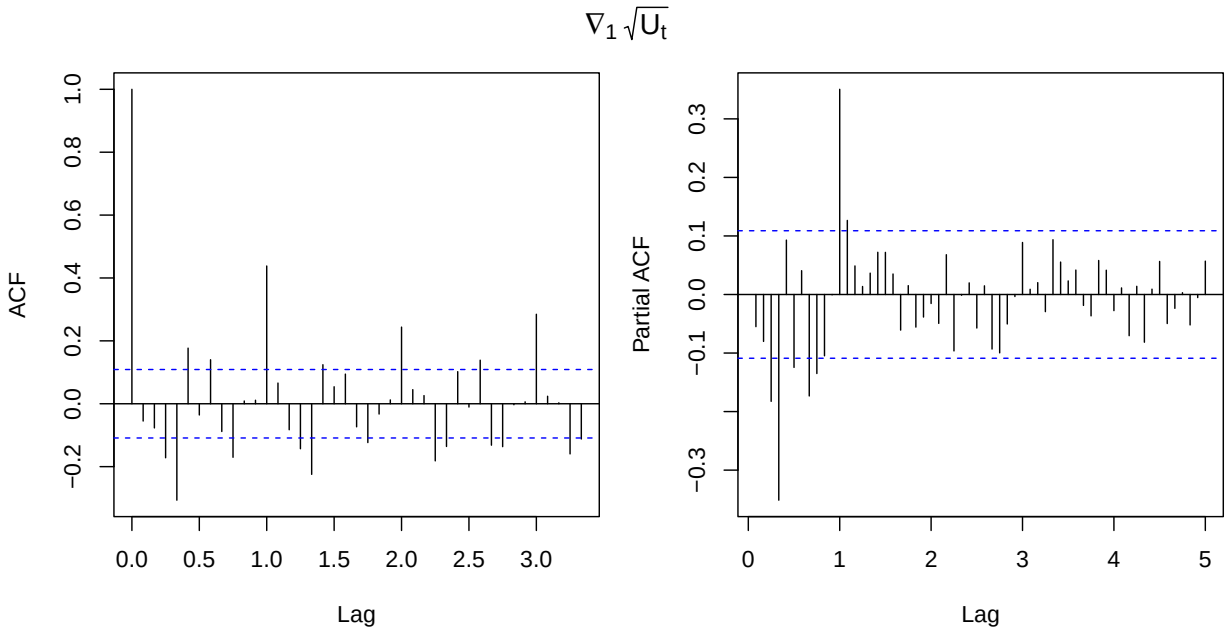


The ACF plot exhibits a trend, as the acf values decay. The PACF plot has a significant pacf at lag 1. Since the PACF plot has this feature, I investigate if the model is purely seasonal autoregressive, since the ACF plot seems to be decaying.

From these plots, I identify the model:

- $\text{SAR}(1)_{12}$

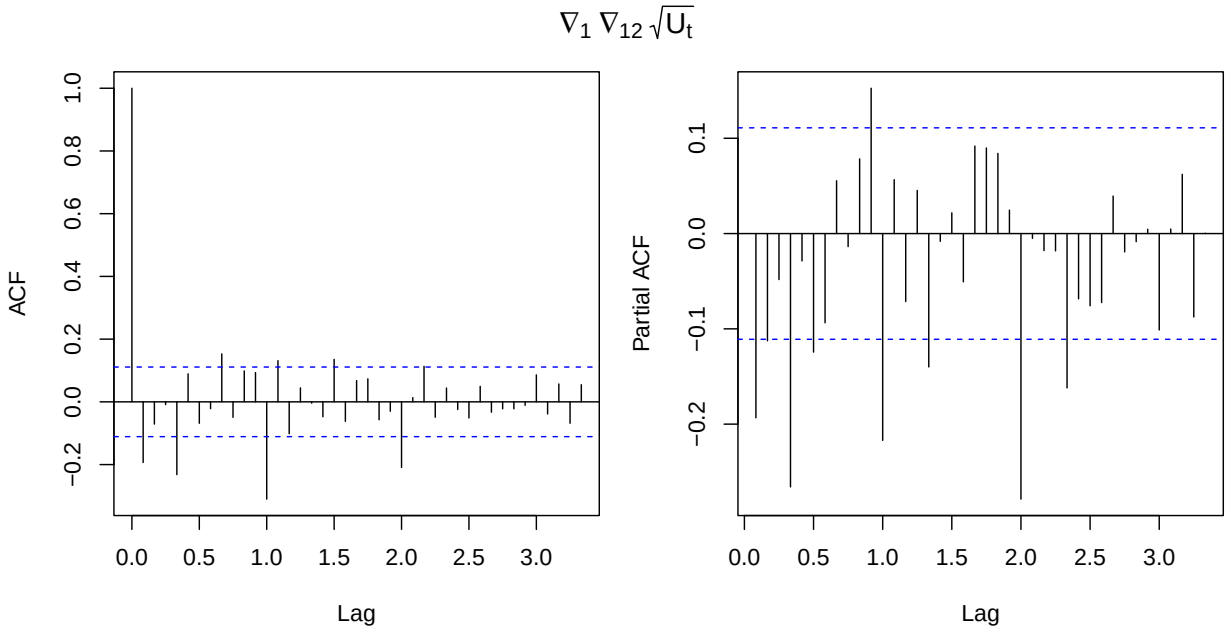
De-trended data



Since this dataset was differenced at lag 1, d = 1. From the ACF plot, I observe a seasonality of significant lags at every s = 12; 1s, 2s, 3s, etc. Within each lag, I see a significant lag of q = 4. From the PACF plot, I notice a significant lag at the s = 12; 1s. There is also a significant pacf value at lag 4, so p = 4. To avoid overcomplicating the model and to avoid fitting complex models with many variables, I will begin by testing:

- $\text{ARIMA}(4, 1, 0)$

De-trended and De-seasonalized data



Since this dataset was differenced at lag 1 and 12, d = 1, D = 1, and s = 12. From the ACF plot, I observe a significant lag at 1s and 2s. This leads me to choose Q = 2. Within each lag, I observe that there is a significant acf at lags 1 and 4; so q = 1 or 4. From the PACF plot, I observe a significant lag at 1s and 2s. This leads me to choose P = 2. Within the seasonal lag, I observe significant pacf at lag 4; so p = 4.

From these plots, we have possible SARIMA models. Some models I can identify are:

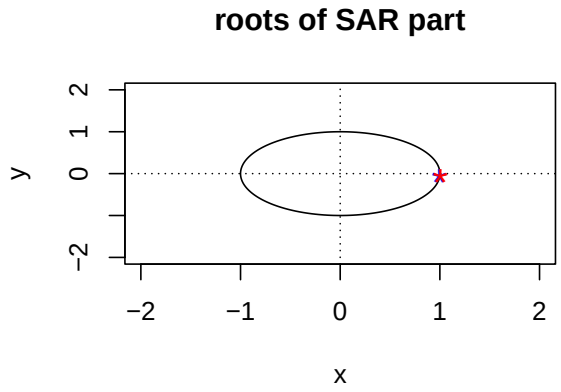
- SARIMA$(4,1,0) \times (2,1,2)_{12}$
- SARIMA$(4,1,0) \times (2,1,0)_{12}$

Model Estimation

$\boxed{\text{SAR}(1)_{12}}$

```
Call:
arima(x = train_data2.sqrt, order = c(0, 0, 0), seasonal = list(order = c(1,
    0, 0), period = 12), method = "ML")

Coefficients:
        sar1  intercept
      0.9961   201.0436
s.e.  0.0016     9.7802
```

```
sigma^2 estimated as 9.488:  log likelihood = -855.86,  aic = 1717.72
```



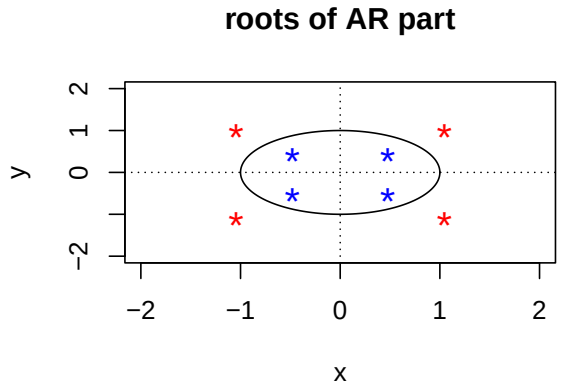
Since the root of the SAR part is *on* the unit circle, the model is **not** stationary. So, we do not consider this model further.

$\boxed{\text{ARIMA}(4,1,0)}$

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), method = "ML")

Coefficients:
         ar1     ar2      ar3      ar4
      0.0146  0.0110  -0.0757  -0.2059
s.e.  0.0545  0.0544   0.0543   0.0546

sigma^2 estimated as 0.644:  log likelihood = -388.55,  aic = 787.11
```

I notice that when fitting this model, the first three estimated AR coefficients are all less than $2 * SE$, indicating that they are not statistically different from 0. So, I re-fit the model fixing these coefficients to 0.

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), fixed = c(0, 0, 0, NA), method = "ML")

Coefficients:
      ar1  ar2  ar3      ar4
        0    0    0  -0.2089
s.e.    0    0    0   0.0546

sigma^2 estimated as 0.6481:  log likelihood = -389.57,  aic = 783.14
```



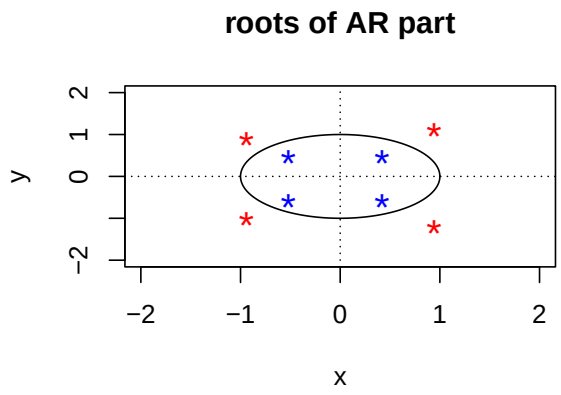
The red points are all outside the unit circle, indicating that the AR part is stationary. Since the model has no MA part, it is invertible and stationary. We can estimate the model as: $(1 + 0.2089B^4)(1 - B)X_t = Z_t$

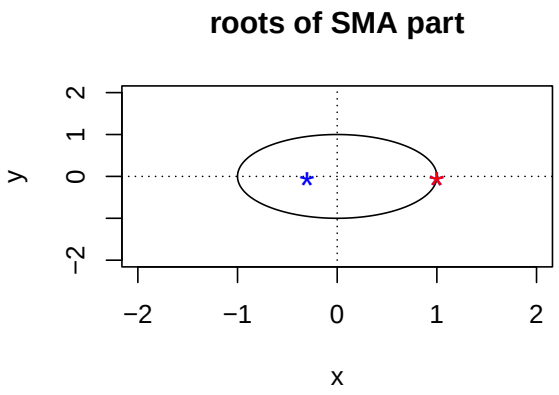
$$\boxed{\text{SARIMA}(4,1,0) \times (2,1,2)_{12}}$$

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(2,
    1, 2), period = 12), method = "ML")

Coefficients:
          ar1      ar2      ar3      ar4     sar1     sar2     sma1     sma2
      -0.2112  -0.1393  -0.0861  -0.2681  -0.2405  -0.0114  -0.4803  -0.5197
s.e.   0.0555   0.0560   0.0562   0.0549   0.1893   0.0842   0.1877   0.1835

sigma^2 estimated as 0.3554:  log likelihood = -299.7,  aic = 617.4
```

After fitting this model, I observe that the third AR coefficient is insignificant. In addition, both of the SAR coefficients are insignificant as well. Thus, I fix these coefficients to 0 as I re-fit the model. Because both of the SAR coefficients reduce to 0, I simplify the model to $\text{SARIMA}(4,1,0) \times (0,1,2)_{12}$.

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(0,
    1, 2), period = 12), fixed = c(NA, NA, 0, NA, NA, NA), method = "ML")

Coefficients:
          ar1      ar2  ar3      ar4     sma1     sma2
      -0.1984  -0.1223    0  -0.2472  -0.6993  -0.3007
s.e.   0.0553   0.0549    0   0.0537   0.0754   0.0644

sigma^2 estimated as 0.3631:  log likelihood = -302.16,  aic = 616.32
```





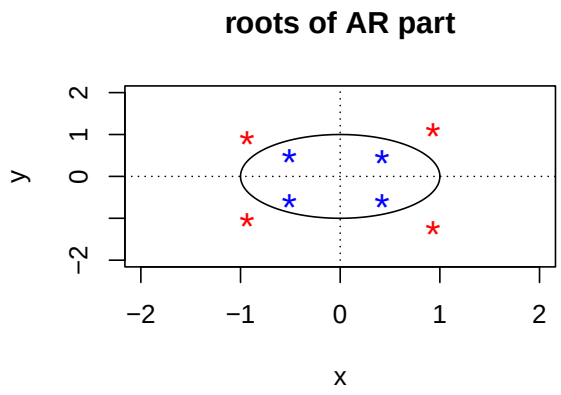
The roots of the AR part are all outside the unit circle, indicating it is stationary. However, the root of the SMA part is *on* the unit circle, indicating that it is **not** invertible. So, I do not proceed to consider this model.

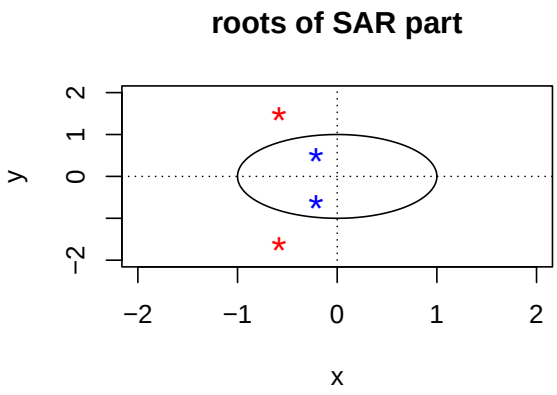
$$\boxed{\text{SARIMA}(4,1,0) \times (2,1,0)_{12}}$$

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(2,
    1, 0), period = 12), method = "ML")

Coefficients:
          ar1      ar2      ar3      ar4     sar1     sar2
      -0.1930  -0.1602  -0.0983  -0.2629  -0.4187  -0.3628
s.e.   0.0557   0.0555   0.0557   0.0547   0.0552   0.0534

sigma^2 estimated as 0.4618:  log likelihood = -324.65,  aic = 663.3
```

After fitting this model, I observe that the third AR coefficient is insignificant, similar to the previous model. Thus, I fix this coefficient to 0 as I re-fit the model.

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(2,
    1, 0), period = 12), fixed = c(NA, NA, 0, NA, NA, NA), method = "ML")

Coefficients:
          ar1      ar2  ar3      ar4     sar1     sar2
      -0.1799  -0.1413    0  -0.2447  -0.4234  -0.3631
s.e.   0.0555   0.0548    0   0.0540   0.0551   0.0534

sigma^2 estimated as 0.4664:  log likelihood = -326.2,  aic = 664.4
```





The roots of the AR and SAR part are all outside the unit circle, indicating it is stationary. There is no MA or SMA part to this model, so it is invertible. Thus, this model is invertible and stationary. We can estimate the model: $(1+0.1799B+0.1413B^2+0.2447B^4)(1+0.4234B^{12}+0.3631B^{24})(1-B)(1-B^{12})X_t = Z_t$

We now check the AICc score for the models that are stationary and invertible, to see which set of parameters minimize this score:

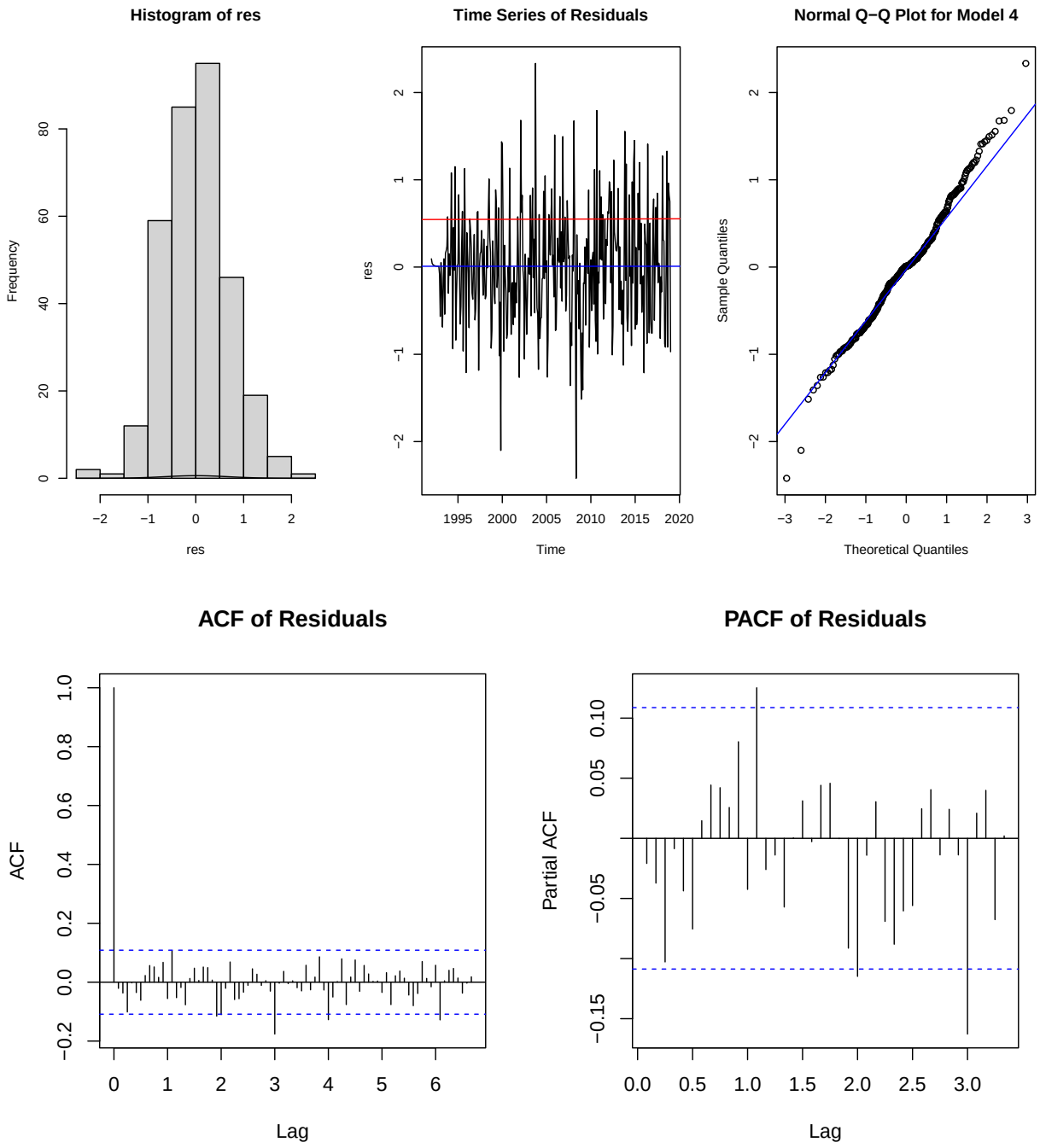
```
[1] Model 2 AICc:  783.138540169298
```

```
[1] Model 4 AICc:  664.398662512026
```

Diagnostic Checking

We now proceed with diagnostic checking for our Model 4 since it has the lowest AICc score.

Analysis of residuals



The distribution of residuals appears normal in the histogram. In addition, the time series of residuals resembles White Noise and the normal QQ plot for this model suggests the residuals are normallu distributed.

However, when observing the ACF and PACF plots of the residuals, there appears to exist some significant acf values at lag 3 as well as significant pacf at lag 12, 24, and 36. This motivates me to increase P, the number of seasonal AR terms at the seasonal lag.

Model Estimation again

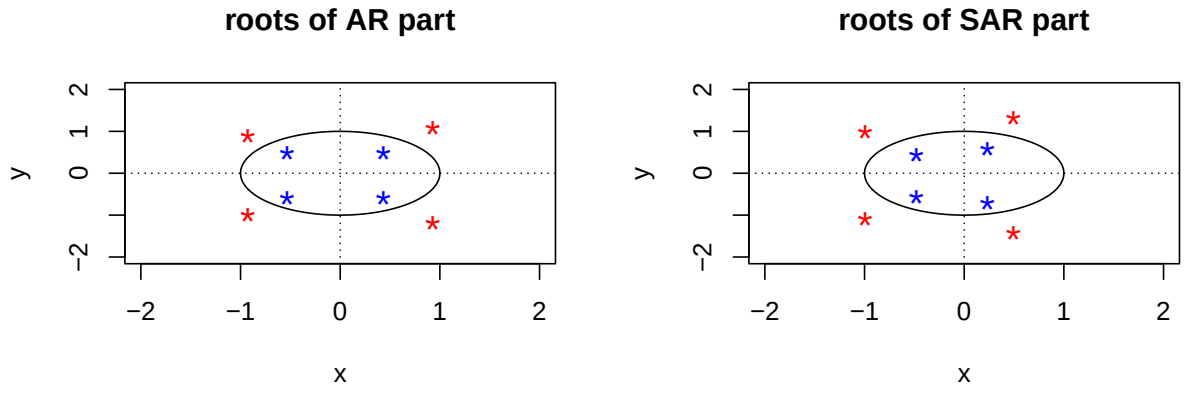
I experiment with $P = 4$ and $P = 5$.

$$\boxed{\text{SARIMA}(4,1,0) \times (4,1,0)_{12}}$$

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(4,
    1, 0), period = 12), method = "ML")

Coefficients:
          ar1      ar2      ar3      ar4     sar1     sar2     sar3     sar4
      -0.2099  -0.1403  -0.1087  -0.2905  -0.4917  -0.5088  -0.2321  -0.2262
s.e.   0.0555   0.0557   0.0555   0.0551   0.0577   0.0621   0.0632   0.0591

sigma^2 estimated as 0.4303:  log likelihood = -315.19,  aic = 648.39
```

After fitting this model, I observe that the third AR coefficient is insignificant, similar to the previous Model 4. Thus, I fix this coefficient to 0 as I re-fit the model.

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(4,
    1, 0), period = 12), fixed = c(NA, NA, 0, NA, NA, NA, NA, NA), method = "ML")

Coefficients:
          ar1      ar2  ar3      ar4     sar1     sar2     sar3     sar4
      -0.1980  -0.1182    0  -0.2691  -0.4930  -0.5072  -0.2263  -0.2252
s.e.   0.0555   0.0549    0   0.0542   0.0578   0.0624   0.0635   0.0593

sigma^2 estimated as 0.4356:  log likelihood = -317.1,  aic = 650.21
```



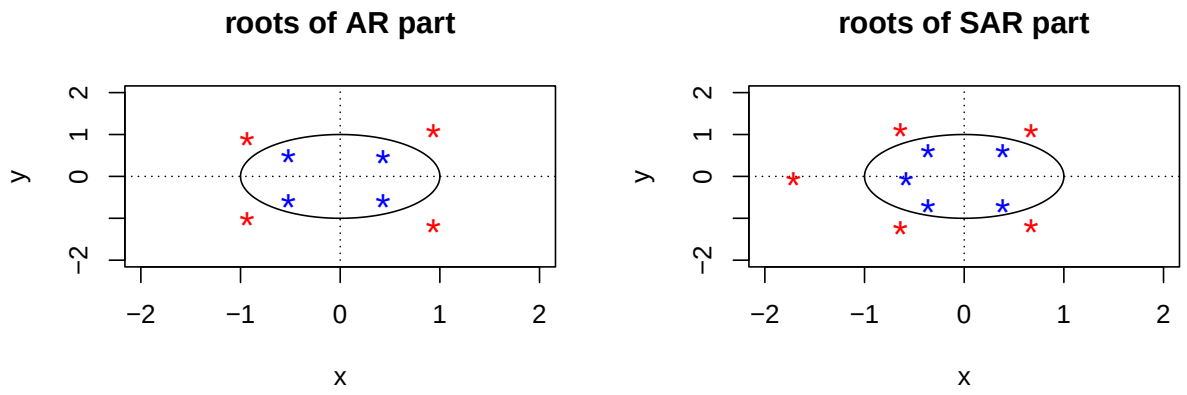
The roots of the AR and SAR part are all outside the unit circle, indicating it is stationary. There is no MA or SMA part to this model, so it is invertible. Thus, this model is invertible and stationary. The estimated model: $(1+0.198B+0.1182B^2+0.2961B^4)(1+0.493B^{12}+0.5072B^{24}+0.2263B^{36}+0.2252B^{48})(1-B)(1-B^{12})X_t = Z_t$

$$\boxed{\text{SARIMA}(4,1,0)\times(5,1,0)_{12}}$$

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(5,
    1, 0), period = 12), method = "ML")

Coefficients:
          ar1      ar2      ar3      ar4     sar1     sar2     sar3     sar4
      -0.1921  -0.1400  -0.1093  -0.2763  -0.5381  -0.5580  -0.3273  -0.3158
s.e.   0.0558   0.0557   0.0556   0.0556   0.0582   0.0631   0.0690   0.0641
         sar5
      -0.1908
s.e.   0.0597

sigma^2 estimated as 0.4139:  log likelihood = -310.25,  aic = 640.51
```

After fitting this model, I observe that the third AR coefficient is insignificant, similar to Model 4 and 5. Thus, I fix this coefficient to 0 as I re-fit the model.

```
Call:
arima(x = train_data2.sqrt, order = c(4, 1, 0), seasonal = list(order = c(5,
    1, 0), period = 12), fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA), method = "ML")

Coefficients:
          ar1      ar2  ar3      ar4     sar1     sar2     sar3     sar4
      -0.1802  -0.1200    0  -0.2570  -0.5388  -0.5551  -0.3208  -0.3155
s.e.   0.0558   0.0551    0   0.0549   0.0583   0.0633   0.0692   0.0644
         sar5
      -0.1907
s.e.   0.0598

sigma^2 estimated as 0.4191:  log likelihood = -312.18,  aic = 642.35
```



With the same reasoning from Model 4 and 5 and based on the plots, this model is invertible and stationary: $(1+0.1802B+0.12B^2+0.257B^4)(1+0.5388B^{12}+0.5551B^{24}+0.3208B^{36}+0.3155B^{48}+0.1907B^{60})(1-B)(1-B^{12})X_t = Z_t$

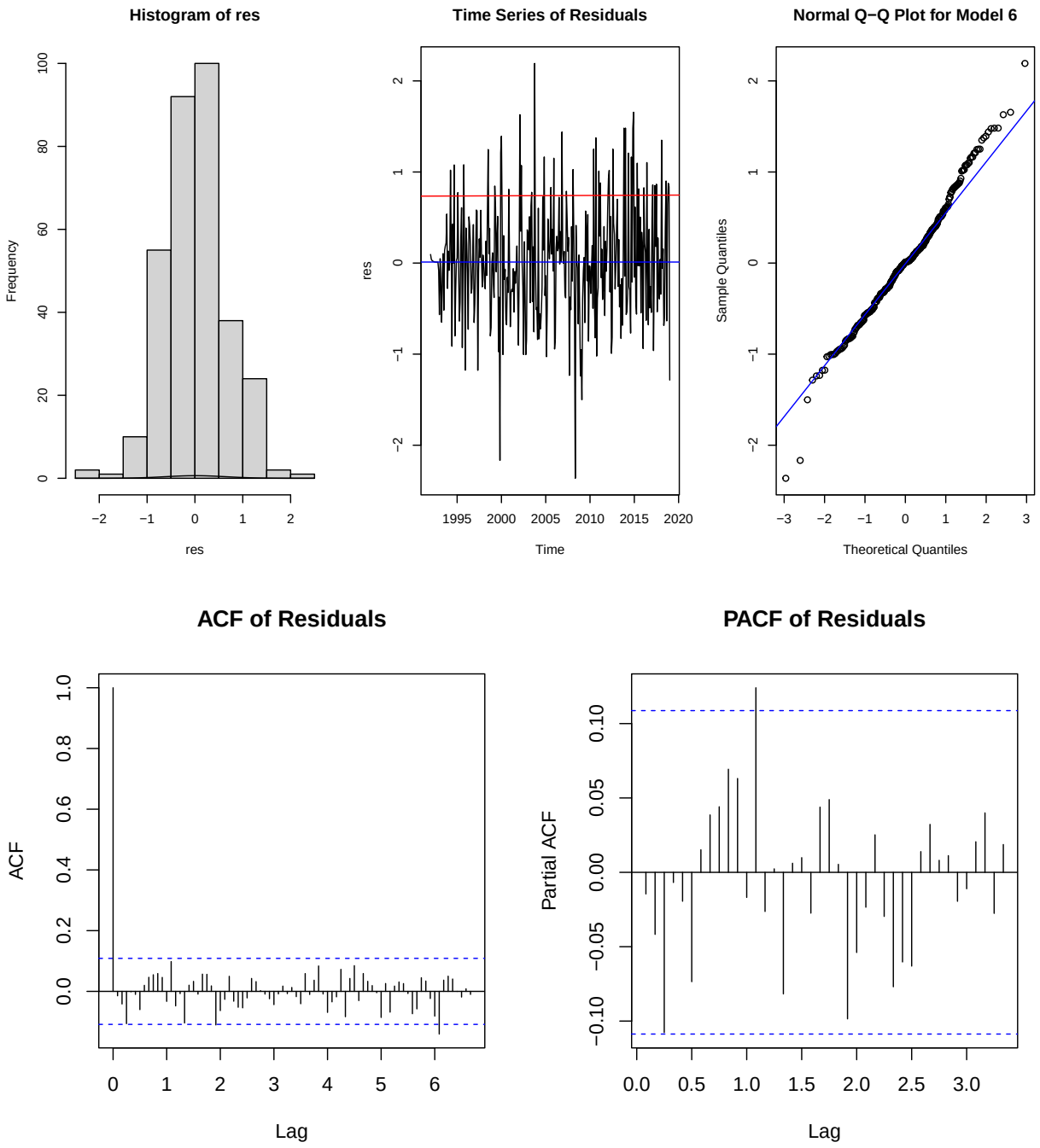
We now check the AICc scores for these two models:

```
[1] Model 5 AICc:  650.206042448755
```

```
[1] Model 6 AICc:  642.35199353813
```

Model 6 has the lower AICc score, so I will proceed with diagnostic checking on it.

Diagnostic Checking again



The histogram and QQ plot for the residuals suggest it to be normal. The time series of the residuals shows no trend or seasonality.

When observing the ACF and PACF plots for the residuals of model, there does not appear any significant acf values (besides slightly at lag 6s, for s=12). The PACF plot shows no significant pacf values as well (besides slightly at lag 1s, for s=12). Almost all of the lags are within the confidence intervals for both ACF and PACF plots.

Portmanteau Tests

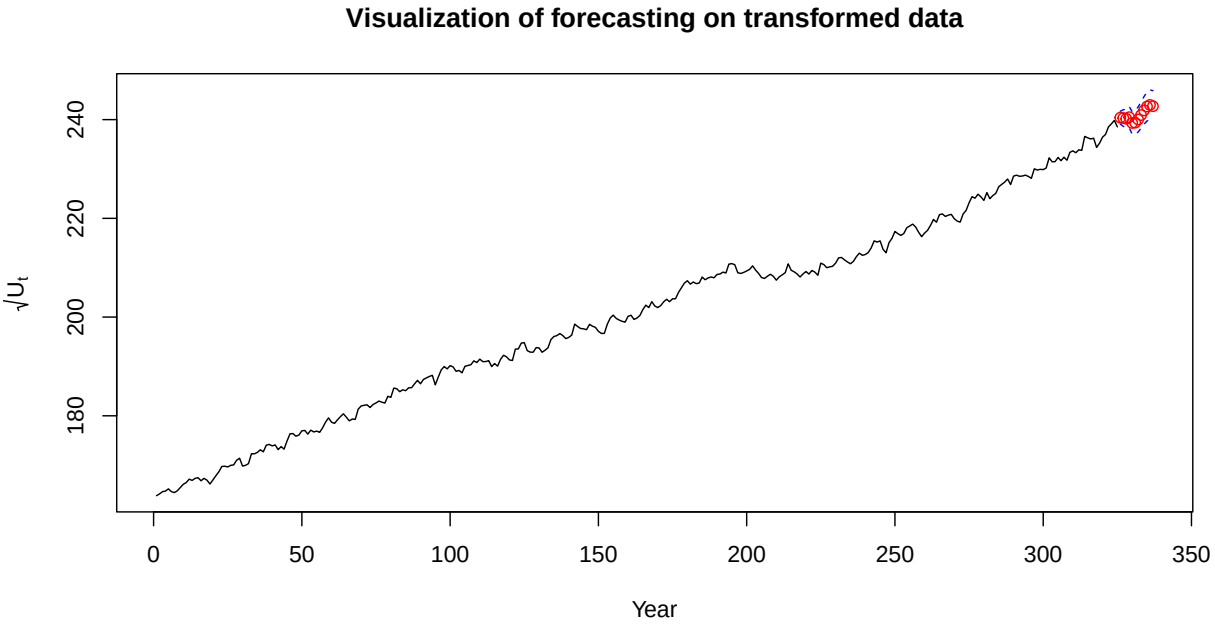
I now proceed with Portmaneau tests for our diagnostic checking, using arguments `fitdf` $= 8$ as the total number of coefficients estimated in the model and `lag` $= 18 = \sqrt{325} \approx 18$:

```
    Shapiro-Wilk normality test

data:  res
W = 0.98773, p-value = 0.007519


    Box-Pierce test

data:  res
X-squared = 17.367, df = 10, p-value = 0.06662


    Box-Ljung test

data:  res
X-squared = 18.035, df = 10, p-value = 0.05438


    Box-Ljung test

data:  res^2
X-squared = 10.637, df = 18, p-value = 0.9091


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


Order selected 0  sigma^2 estimated as  0.4045
```

The Shapiro-Wilk normality test has a p-value less than 0.05, indicating that the residuals do not follow a normal distribution. However, we just need to meet the assumption that the residuals are White Noise. The Box-Pierce, Ljung-Box, and Mcleod-Li test all have p-values greater than 0.05, indicating that the residuals are White Noise. We also test the residuals in Yule-Walker method using AIC, automatically selecting order 0. Thus, we conclude that the model is satisfactory.

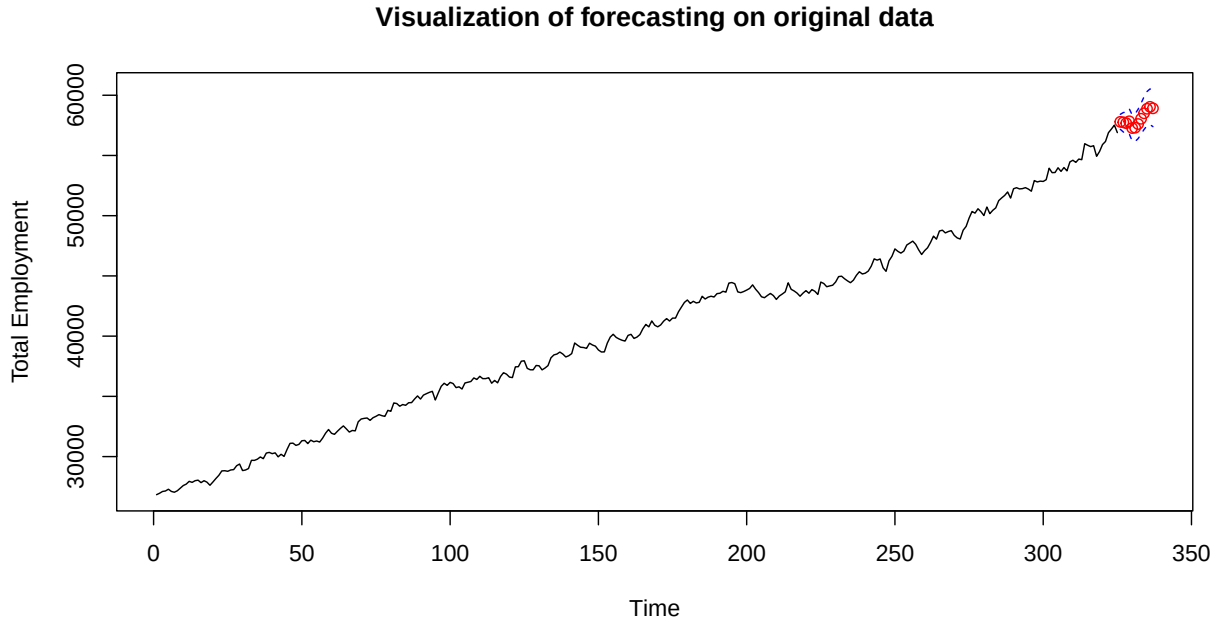
Forecasting

We will now proceed with forecasting the monthly total employment population in the year 2019. I will use the `forecast` library to generate predictions based on our model.



We plot the forecasted points (red) for the months in 2019 attached to our time series plot from January 1992 to December 2018. The blue lines represent the confidence interval for these forecasted points.

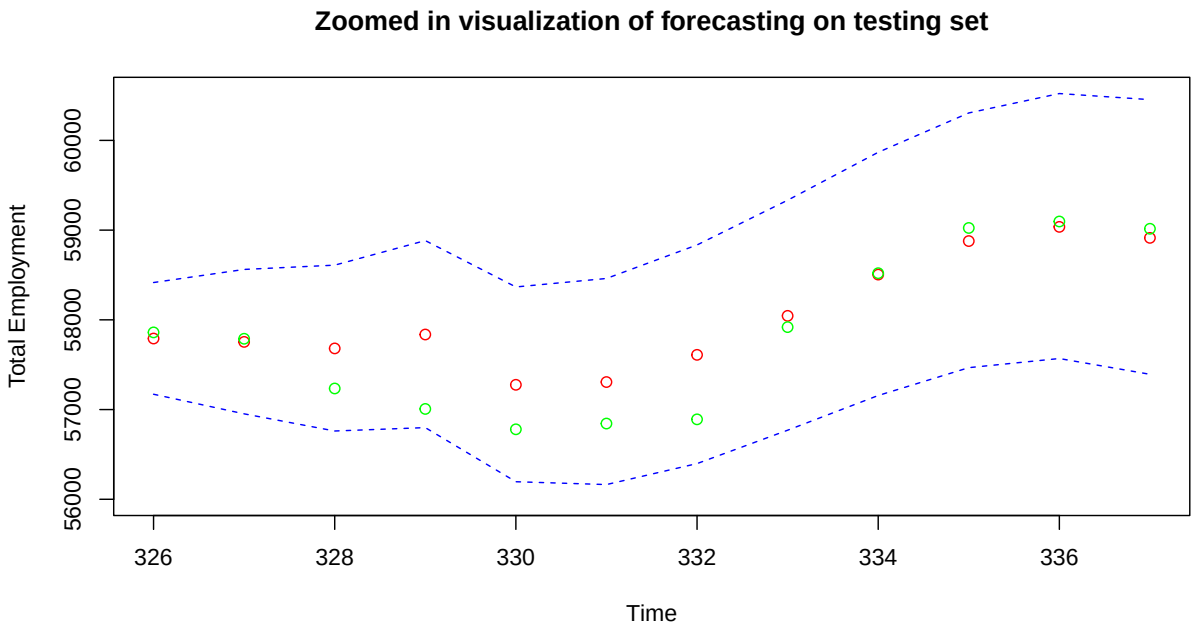
In order to interpret our results, we will un-transform our data.



The red forecasted points are all visibly within the blue confidence interval (lower and upper bounds).

We zoom into our predicted values to compare our forecast with the testing set- the actual monthly

employment totals in the year 2019. The predicted values are colored in red, while the true testing data is colored in green.



Our forecast is successful. The months of January, February, August, September, October, November, and December closely align with the true values.

Conclusion

In this project, I aimed to forecast the monthly total employment population for individuals with a bachelor's degree in 2019 using time series modeling. I evaluated various models and determined that the SARIMA$(4,1,0)\times(5,1,0)_{12}$ model provided the best fit based on diagnostic checking and AICc scores. The model is estimated by the equation: $(1+0.1802B+0.12B^2+0.257B^4)(1+0.5388B^{12}+0.5551B^{24}+0.3208B^{36}+0.3155B^{48}+0.1907B^{60})(1-B)(1-B^{12})X_t=Z_t$. This model successfully produced forecasts for the year 2019, demonstrating that forecasting employment trends for individuals with a bachelor's degree can provide valuable insights into how the job market can look like for graduating college students. This can help companies, educators, and job seekers to make informed decisions. By understanding and identifying these patterns in employment trends, this analysis can aid in workforce planning and offer perspective on the job market for college graduates.

Acknowledgements

- Raya Feldman (Professor)
- Lihao Xiao (Teaching Assistant)

Code Appendix

```
# default code chunk options
knitr::opts_chunk$set(echo = F,
                      message = FALSE,
                      warning = FALSE,
                      comment = NA)

library(MASS)
library(forecast)
source("~/Desktop/Projects/PSTAT 174/Labs/plot.roots.R")

set.seed(123)
# look at rate, employed total, unemployed total
library("readxl")
data <- read_excel("employmentRate_edu.xlsx", range = "A13:M47")
data2 <- read_excel("employed.xlsx", range = "A13:M47")
data3 <- read_excel("unemployed.xlsx", range = "A13:M47")

data <- as.vector(t(as.matrix(data[,-1])))
data2 <- as.vector(t(as.matrix(data2[,-1])))
data3 <- as.vector(t(as.matrix(data3[,-1])))

op <- par(mfrow = c(1,3))
op2 <- par(oma = c(2, 0, 2, 0))
ts.plot(ts(data, start = c(1992, 1), frequency = 12), main = "Employment Rate",
     xlab = "Year", ylab = "Employment-Population Ratio")
ts.plot(ts(data2, start = c(1992, 1), frequency = 12), main = "Total Employment",
     xlab = "Year", ylab = "Total Employment")
ts.plot(ts(data3, start = c(1992, 1), frequency = 12), main = "Total Unemployment",
     xlab = "Year", ylab = "Total Unemployment")
mtext("Time Series from Year Jan 1992 to Jan 2025", outer = T, cex = 1.25, font = 2)
train_data2 <- ts(data2[c(1:325)], start = c(1992, 1), frequency = 12)
test_data2 <-  ts(data2[c(326:337)], start = c(1992, 1), frequency = 12)

ts.plot(train_data2,
        main = "Total Employment Time Series from Year 1992 to Dec 2019",
        xlab = "Year", ylab = "Total Employment")
# we will consider any box cox transformations on the EMPLOYMENT TOTAL
op <- par(mfrow = c(1,2))
hist(train_data2)
bcTransform <- boxcox(train_data2 ~ as.numeric(1:length(train_data2)))

lambda <- bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
print(paste("lambda: ", lambda), quote = F)
train_data2.bc <- (1/lambda)*(train_data2^lambda-1)
```

```r
train_data2.sqrt <- sqrt(train_data2)

op <- par(mfrow = c(1,2))
op2 <- par(oma = c(2, 0, 2, 0))
plot.ts(train_data2.bc, main = "bc(U_t)")
plot.ts(train_data2.sqrt, main = "sqrt(U_t)")
mtext("Time Series", outer = T, cex = 1.25, font = 2)

op <- par(mfrow = c(1,3))
op2 <- par(oma = c(2, 0, 2, 0))
hist(train_data2, col="light blue", xlab="", main="U_t")
hist(train_data2.bc, col="light blue", xlab="", main="bc(U_t)")
hist(train_data2.sqrt, col="light blue", xlab="", main="sqrt(U_t)")
mtext("Histogram", outer = T, cex = 1.25, font = 2)
y <- ts(as.ts(train_data2.sqrt), frequency = 12)
plot(decompose(y))
train_data2.sqrt_1 <- diff(train_data2.sqrt, 1)

ts.plot(train_data2.sqrt_1, main = "De-trended Time Series from Year 1992 to 2019",
     xlab = "Year", ylab = "Total Employment")
fit <- lm(train_data2.sqrt_1 ~ as.numeric(1:length(train_data2.sqrt_1))); abline(fit, col="red"
abline(h = mean(train_data2.sqrt_1), col="blue")

# print mean and variance
print(paste("Mean: ", mean(train_data2.sqrt_1)), quote = F)
print(paste("Variance: ", var(train_data2.sqrt_1)), quote = F)
train_data2.sqrt_12 <- diff(train_data2.sqrt_1, 12)
ts.plot(train_data2.sqrt_12, main = "De-trended and De-seasonalized Time Series from Year 1992
     xlab = "Year", ylab = "Total Employment")
fit <- lm(train_data2.sqrt_12 ~ as.numeric(1:length(train_data2.sqrt_12))); abline(fit, col="re
abline(h=mean(train_data2.sqrt_12), col="blue")
print(paste("Mean: ", mean(train_data2.sqrt_12)), quote = F)
print(paste("Variance: ", var(train_data2.sqrt_12)), quote = F)
train_data2.sqrt_2 <- diff(train_data2.sqrt, 12)

ts.plot(train_data2.sqrt_2, main = "De-seasonalized Time Series from Year 1992 to 2019",
     xlab = "Year", ylab = "Total Employment")
fit <- lm(train_data2.sqrt_2 ~ as.numeric(1:length(train_data2.sqrt_2))); abline(fit, col="red"
abline(h=mean(train_data2.sqrt_2), col="blue")
print(paste("Mean: ", mean(train_data2.sqrt_2)), quote = F)
print(paste("Variance: ", var(train_data2.sqrt_2)), quote = F)
op <- par(mfrow = c(1,3))
op2 <- par(oma = c(2, 0, 2, 0))
hist(train_data2.sqrt, col="light blue", xlab="", main=expression(sqrt(U[t])))
hist(train_data2.sqrt_1, col="light blue", xlab="", main=expression(nabla[1] ~ sqrt(U[t])))
hist(train_data2.sqrt_12, col="light blue", xlab="", main=expression(nabla[1] ~ nabla[12] ~ sq
mtext("Histogram", outer = T, cex = 1.25, font = 2)
```

```r
op <- par(mfrow = c(1,2))
op2 <- par(oma = c(2, 0, 2, 0), mar = c(4, 4, 1, 1))
acf(train_data2.sqrt, lag.max=40, main = "")
pacf(train_data2.sqrt, lag.max=40, main = "")
mtext(expression(sqrt(U[t])), outer = T, cex = 1.25, font = 2)
op <- par(mfrow = c(1,2))
op2 <- par(oma = c(2, 0, 2, 0), mar = c(4, 4, 1, 1))
acf(train_data2.sqrt_1, lag.max=40, main = "")
pacf(train_data2.sqrt_1, lag.max=60, main = "")
mtext(expression(nabla[1] ~ sqrt(U[t])), outer = T, cex = 1.25, font = 2)
op <- par(mfrow = c(1,2))
op2 <- par(oma = c(2, 0, 2, 0), mar = c(4, 4, 1, 1))
acf(train_data2.sqrt_12, lag.max=40, main = "")
pacf(train_data2.sqrt_12, lag.max=40, main = "")
mtext(expression(nabla[1] ~ nabla[12] ~ sqrt(U[t])), outer = T, cex = 1.25, font = 2)
# SAR(1)_12
model1 <- arima(train_data2.sqrt, order=c(0,0,0),
      seasonal = list(order = c(1,0,0), period = 12), method="ML")
model1

op <- par(mfrow = c(1,2))
plot.roots(NULL, polyroot(c(1, -0.9961)), main = "roots of SAR part")
# ARIMA(4, 1, 0)
model2 <- arima(train_data2.sqrt, order=c(4,1,0),
      method="ML")
model2
model2b <- arima(train_data2.sqrt, order=c(4,1,0),
                 fixed = c(0, 0, 0, NA),
                 method="ML")
model2b

op <- par(mfrow = c(1,2))
plot.roots(NULL,polyroot(c(1, 0, 0, 0, 0.2089)), main="roots of AR part")
# SARIMA(4, 1, 0) x (2, 1, 2)_{12}
model3 <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(2, 1, 2), period = 12),
      method="ML")
model3
model3b <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(2, 1, 2), period = 12),
      fixed = c(NA, NA, 0, NA, 0, 0, NA, NA),
      method="ML")
model3c <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(0, 1, 2), period = 12),
      fixed = c(NA, NA, 0, NA, NA, NA),
      method="ML") # same model
model3c
```

```r
# check model 3c
op <- par(mfrow = c(1,2))
plot.roots(NULL, polyroot(c(1, 0.1984, 0.1223, 0, 0.2472)), main = "roots of AR part")
plot.roots(NULL, polyroot(c(1, -0.6993, -0.3007)), main = "roots of SMA part") # on the unit c
# SARIMA(4, 1, 0)x(2, 1, 0)
model4 <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(2,1,0), period = 12),
      method="ML")
model4
model4b <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(2,1,0), period = 12),
      fixed = c(NA, NA, 0, NA, NA, NA),
      method="ML")
model4b

op <- par(mfrow = c(1,2))
plot.roots(NULL,polyroot(c(1, 0.1799, 0.1413, 0, 0.2447)), main="roots of AR part")
# no MA, SMA part
plot.roots(NULL,polyroot(c(1, 0.4234, 0.3631)), main="roots of SAR part")
# all stationary and lies outside unit circle
print(paste("Model 2 AICc: ", model2b$aic), quote = F)
print(paste("Model 4 AICc: ", model4b$aic), quote = F)
res <- residuals(model4b)

op <- par(mfrow = c(1,3))
hist(res)
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = T)
plot.ts(res, main = "Time Series of Residuals")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model 4")
qqline(res,col="blue")

op <- par(mfrow = c(1,2))
acf(res, lag.max=80, main = "ACF of Residuals")
pacf(res, lag.max = 40, main ="PACF of Residuals")
# SARIMA(4, 1, 0)x(4, 1, 0)
model5 <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(4,1,0), period = 12),
      method="ML")
model5
model5b <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(4,1,0), period = 12),
      fixed = c(NA, NA, 0, NA, NA, NA, NA, NA),
      method="ML")
```

```r
model5b

op <- par(mfrow = c(1,2))
plot.roots(NULL,polyroot(c(1, 0.1980, 0.1182, 0, 0.2691)), main="roots of AR part")
# no MA, SMA part
plot.roots(NULL,polyroot(c(1, 0.4930, 0.5072, 0.2263, 0.2252)), main="roots of SAR part")
# SARIMA(4, 1, 0)x(5, 1, 0)
model6 <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(5,1,0), period = 12),
      method="ML")
model6
model6b <- arima(train_data2.sqrt, order=c(4,1,0),
      seasonal = list(order = c(5,1,0), period = 12),
      fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA),
      method="ML")
model6b

op <- par(mfrow = c(1,2))
plot.roots(NULL,polyroot(c(1, 0.1802, 0.1200, 0, 0.2570)), main="roots of AR part")
# no MA, SMA part
plot.roots(NULL,polyroot(c(1, 0.5388, 0.5551, 0.3208, 0.3155, 0.1907)), main="roots of SAR part
# all stationary and lies outside unit circle
print(paste("Model 5 AICc: ", model5b$aic), quote = F)
print(paste("Model 6 AICc: ", model6b$aic), quote = F)
res <- residuals(model6b)

op <- par(mfrow = c(1,3))
hist(res)
m <- mean(res)
std <- sqrt(var(res))
curve(dnorm(x, m, std), add = T)
plot.ts(res, main = "Time Series of Residuals")
fitt <- lm(res ~ as.numeric(1:length(res))); abline(fitt, col="red")
abline(h=mean(res), col="blue")
qqnorm(res,main= "Normal Q-Q Plot for Model 6")
qqline(res,col="blue")

op <- par(mfrow = c(1,2))
acf(res, lag.max=80, main = "ACF of Residuals")
pacf(res, lag.max = 40, main ="PACF of Residuals")
shapiro.test(res)

n <- round(sqrt(length(res)))

Box.test(res, lag = 18, type = c("Box-Pierce"), fitdf = 8)

Box.test(res, lag = 18, type = c("Ljung-Box"), fitdf = 8)
```

```r
Box.test(res^2, lag = 18, type = c("Ljung-Box"), fitdf = 0)

ar(res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
forecast(model6b)
# for indexing purposes
model6b_forecast <- arima(as.numeric(train_data2.sqrt), order=c(4,1,0),
      seasonal = list(order = c(5,1,0), period = 12),
      fixed = c(NA, NA, 0, NA, NA, NA, NA, NA, NA),
      method="ML")

pred.tr <- predict(model6b_forecast, n.ahead = 12)
U.tr <- pred.tr$pred + 2*pred.tr$se
L.tr <- pred.tr$pred - 2*pred.tr$se

ts.plot(as.numeric(train_data2.sqrt),
        xlim = c(1, length(train_data2.sqrt) + 12),
        ylim = c(min(train_data2.sqrt), max(U.tr)),
        xlab = "Year",
        ylab = expression(sqrt(U[t])),
        main="Visualization of forecasting on transformed data")
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
points((length(train_data2.sqrt) + 1):(length(train_data2.sqrt) + 12),
       pred.tr$pred, col = "red", size = 0.3)
pred.orig <- (pred.tr$pred)^2
U <- (U.tr)^2
L <- (L.tr)^2
ts.plot(as.numeric(train_data2),
        xlim=c(1,length(train_data2) + 12),
        ylim = c(min(train_data2), max(U)),
        xlab = "Time",
        ylab = "Total Employment",
        main="Visualization of forecasting on original data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train_data2)+1):(length(train_data2)+12), pred.orig, col="red", size = 0.3)
ts.plot(as.numeric(train_data2),
        xlim = c(326, length(train_data2)+12),
        ylim = c(56000, max(U)),
        xlab = "Time",
        ylab = "Total Employment",
        main="Zoomed in visualization of forecasting on testing set")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points((length(train_data2)+1):(length(train_data2)+12), pred.orig, col="red")
points((length(train_data2)+1):(length(train_data2)+12), test_data2, col="green")
```