PSTAT 174 Final Project

Female Unemployment in the US: Time Series Analysis

Authors: Evan Azevedo, Manpal Sidhu, Jay Singh, Allen Wang

UCSB Statistics Department

PSTAT 174

Professor Bapat

Table of Contents:

Abstract

The prediction of female unemployment is useful since this can give the government and other institutions information to accurately predict how well off females are, and how much they may need to be assisted in the US. The objective of this project report is to predict monthly U.S. female unemployment using methods of time series analysis on a dataset of US Women over 20's monthly unemployment from 1948 - 1981. We build a SARIMA model using the lowest AICc and BIC score based on the stationary data, which we obtain from transforming and differencing the original data. Diagnostic checks are performed on the final model. Lastly, we made our forecast for March 1st, 1981 to December 1st, 1981. Our predicted values are within the confidence level of 95% and are approximately close to the true value from the original dataset.

1 Introduction

Female unemployment in the US is a useful measure to predict, since it can allow institutions such as the government to see if women are being discriminated against in employment, as well as allowing them to predict how much in aid they may need during the month. Thus, it is important to examine the trend and predict its value for future reference.

The data we found from DataMarket.com provided us with the unemployment in thousands of females in the United States ranging from January 1948 to December 1981. The dataset we chose has a reasonably large sample size (n=408), and hence is suitable for analysis. This dataset is interesting because it can be interpreted in tandem with a knowledge of US history. This time period includes the end of World War II, the bulk of the Cold War, as well as the US's meteoric rise as a global power. Strangely, we see a rise in unemployment as we head into the 1970's, and a large spike leading into 1981.

From the plot of the dataset, we can see a clear upward trend, which indicates increasing unemployment, as well as a seasonal pattern at a uniform interval. We then difference the data at lag 12, and then lag 1, and find that the variance of the

transformed and differenced data decreases significantly, which implies that there indeed was seasonality and a trend in the original data. Using time series analysis, we thus can make the data stationary and allows us to fit it into a model that allows us to forecast women's future unemployment level.

We identify 16 possible SARIMA models based on the ACF and PACF plots of the data, and then further narrow down the choices to 2 possible models based on the AICc and BIC model selection criterion. We estimate the parameters of the model using the MLE method. After running diagnostic checks such as the Shapiro-Wilk test, Box-Pierce test, and Ljung-Box test, and considering their AICc and BIC values, we choose the final model as SARIMA(1,1,1,) × $(0,1,1)_{12}$. Lastly, we forecast the future female unemployment figures up to 10 months ahead and compare them with their true values. The results show that all of our model's observed values fall within the 95% confidence interval values, thus proving the feasibility of our final model, although there is still room for improvement.

2 Data Analysis

The data includes 2 variables: date on a monthly basis and the female unemployment count in thousands. There are 408 observations in total, but we cut the last 10 observations to compare with our forecasted values from the final model. Thus, there are 398 data points for our time series analysis. For our preliminary data exploration, we save the unemployment data in time series form and plot it with all 398 observations. Figure 2.1 shows the plot of the time series data.

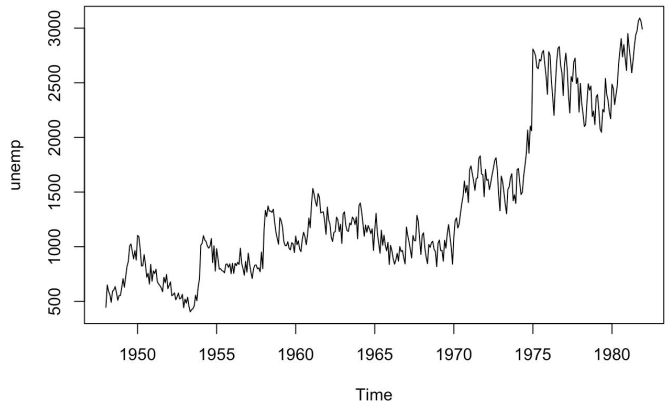


Figure 2.1 - The initial time series plot of our dataset.

In this plot, we see an upward trend that appears somewhat exponential. There is also a seasonal component as we see large peaks and troughs around every 2 years. There are sharp changes throughout the graph but especially in the last quarter we notice very sharp ups and downs. To show the seasonal and trend components of this series, we implement a decomposition model $Y_t = m_t + s_t + S_t$, where m_t is the trend component, s_t is the seasonal component, and S_t is the stationary component. Figure 2.2 below shows the decomposition of the original signal.

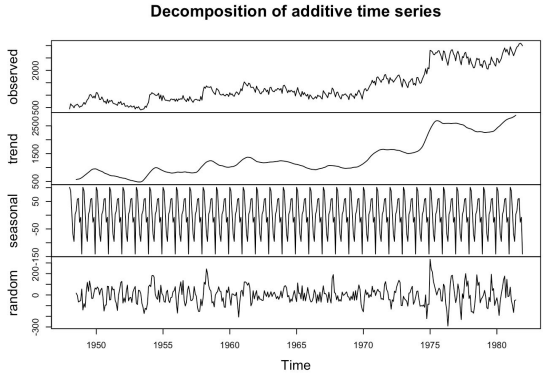


Figure 2.2 - Decomposition of the initial time series

From this graph, we see that we can easily decompose the original signal into a linear combination of components. Now, we must transform the original series into a stationary one by removing those components of the signal.

3 Stationarity and Transformations

We first use the Box-Cox transformation to modify the distribution of the dataset. We find the transformation parameter $\lambda = -0.02020202$. Thus, we get the following transformation of the original time series, V_t:

$$Y_t = V_t^{-0.02020202}.$$

This transformation reduces the variance of the dataset from 421856.9 to 0.1638705.

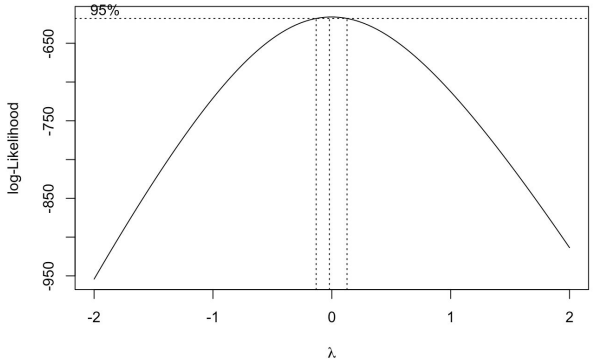


Fig. 3.1 Optimal lambda for Box-Cox transformation

This results in a PACF graph shown fig. 3.2 in Appendix. This suggests a seasonality factor of 12 months, or 1 year, and that the process does not yet display stationarity.

To remove the seasonality component, we difference the time series at lag 12. This results in the time series shown in figure 3.3 in the Appendix, and with ACF and PACF shown in figure 3.4 below.

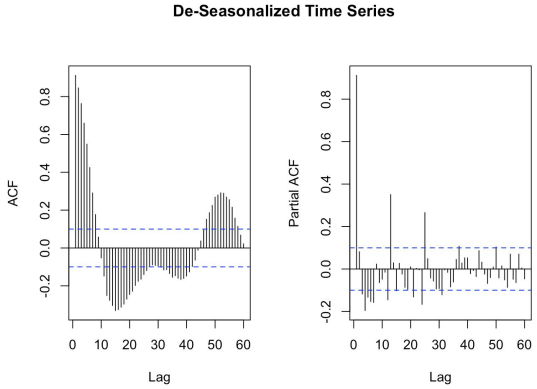


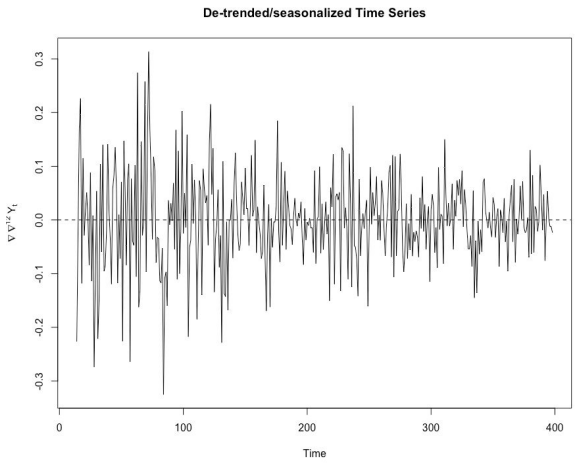
Fig. 3.4 - ACF and PACF of the resulting time series. Indicates further differencing necessary.

From fig. 3.4, we infer that further differencing is needed, and compare differencing twice at lag 1. The variance is decreased by differencing at lag 1; however, it increases when we difference a second time, so we choose the transformation:

$$X_t = \nabla \, \nabla^{12} Y_t$$

,

resulting in the final time series plot and ACF/PACF graphs shown in figures 3.5 and 3.6 below.



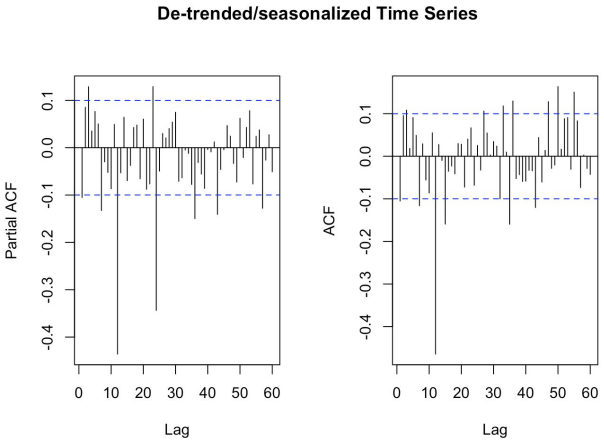


Figure 3.5 - Detrended and seasonalized time series graph

Figure 3.6 - ACF and PACF of the finally transformed time series

We use the Augmented Dickey-Fuller test to determine whether or not our time series is stationary. The null hypothesis is that X_t is not stationary and the alternative hypothesis is that X_t is stationary. From our test result, we obtain a p-value of 0.01, and thus can reject the null hypothesis and conclude that X_t is indeed stationary at the 95% confidence level. We can now select and build our model now with this data.

4 Model Identification

Due to the seasonality of the data, an appropriate model would be a SARIMA model to analyze the data. The structure of SARIMA model is as below:

$$SARIMA(p,d,q) \times (P,D,Q)_s$$

where p is the order of non-seasonal AR process, d is the non-seasonal differencing, q is the the order of non-seasonal MA process, P is the order of seasonal AR process, D is the seasonal differencing, Q is the order of seasonal AR process, and s is the period of the time lag.
In our case, we have s=12 due to the seasonality of 12 months. Then, we need to find d and D based on the differencing procedure in the previous step. Thus, we have d=1 and D=1 since we difference the transformed data at lag 12 to remove seasonality and then differenced again at lag 1 to remove trend. Next, we need to find p, q and P, Q based on the ACF and PACF plots for the preliminary model identification.

We plot the ACF and PACF of stationary time series X_t in order to identify the seasonal terms P and Q. Since our seasonal component s is 12 so we need to check value of ACF and PACF at seasonal lags 12, 24, 36, 48, etc. Therefore, for the seasonal terms P and Q, ACF plot cuts off after lag 12 (Q=1) and PACF plot tails off exponentially (P=0).

Then, we find p, q by looking at the ACF and PACF plots at lag 1, 2, 3, 4...11. Thus, for the non-seasonal terms p and q, the ACF plot cuts off after lag 3 (q=3), and the PACF plot cuts off after lag 3(p=3). Or, we can also we that the ACF and PACF plots tail off exponentially after lag 3, and hence we have an ARMA model with max(p,q) = 3. Thus, we conclude that p and q can take values in 0,1,2,3. We thus fix the seasonal terms P and Q, and test all possible combinations of the non-seasonal terms p, q from 0 to 3. Thus, we have 16 models to consider.

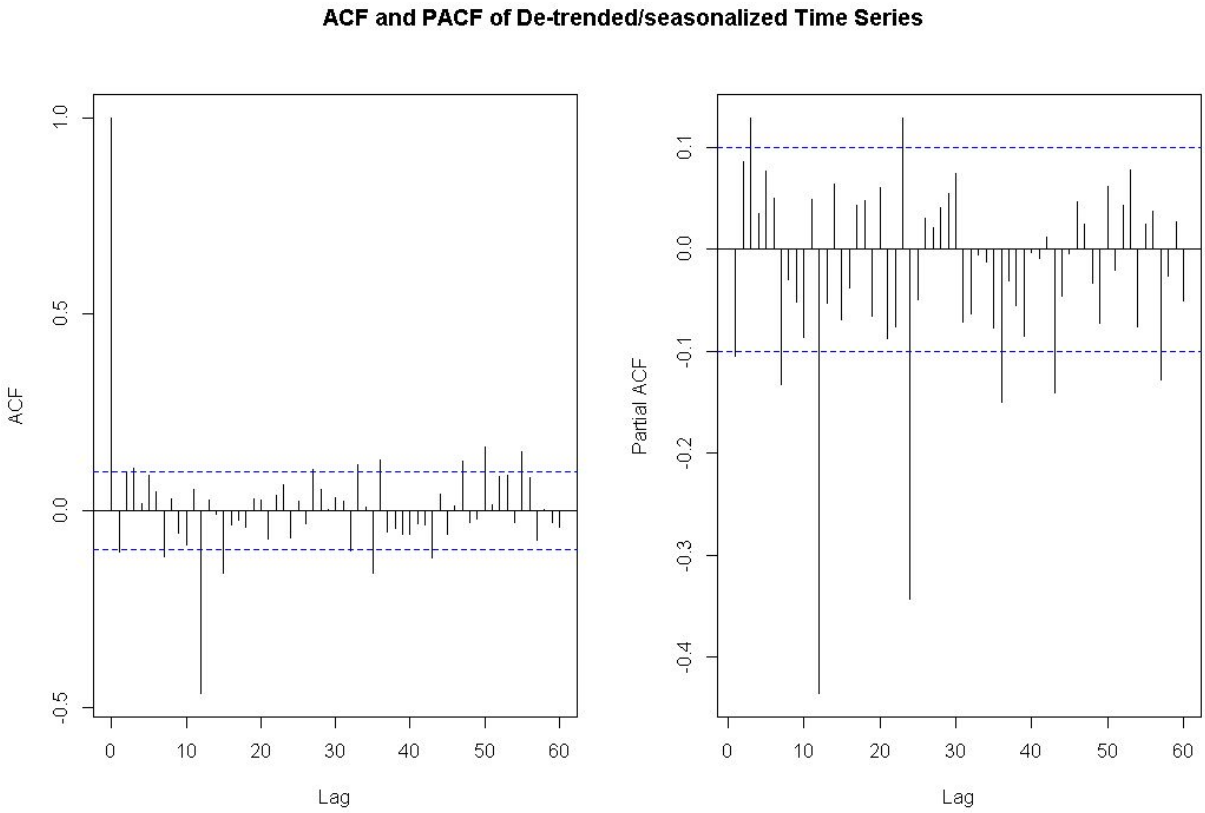


Figure 4.1

From the preliminary 16 models identified, we use the information criterion such as the AICc and BIC values to select the best possible models by selecting the models with minimum scores. The corrected version of Akaike's Information criterion (AICc) gives us the smallest result when p=2 and q=1, and the second smallest when p=1 and q=0. The Bayesian Information criterion (BIC) gives us the smallest value when p=1 and q=0, and the second smallest when p=0 and q=1. Therefore, we select the two possible models based on the best AICc and BIC values:

$$SARIMA(2,1,1) \times (0,1,1)_{12} \text{ and } SARIMA(1,1,0) \times (0,1,1)_{12}.$$

	q=0	q=1	q=2
p=0	-12.20630	-12.21978	-12.22161
p=1	-12.22294	-12.22929	-12.22953
p=2	-12.22279	-12.22984	-12.21810

Table 4.1 AICc

	q=0	q=1	q=2
p=0	-13.20139	-13.20493	-13.19684
p=1	-13.20808	-13.20453	-13.19487
p=2	-13.19803	-13.19519	-13.17358

Table 4.2 BIC

For the two models we selected above, we fit and estimate the coefficients based on the MLE method.

	Model 1	Model 2
AR(1)	0.4794	-0.1447
AR(2)	0.1636	-
MA(1)	-0.6187	-
SMA(1)	-0.8379	-0.8450

Table 4.3 Coefficients of Two SARIMA Models

We have the models where X_t is the transformed and differenced data, $X_t = \nabla \nabla_{12} V_t^{-0.02020202}$

$$\text{Model1 : SARIMA}(1,1,1) \times (0,1,1)_{12}$$

$$(1 - 0.4794B - 0.1636B^2)X_t = (1 + 0.6187B)(1 + 0.8379B^{12})Z_t$$

$$\text{where } Z_t \sim N(0, 1.768e^{-6})$$

$$\text{Model2 : SARIMA}(0,1,2) \times (0,1,1)_{12}$$

$$(1 + 0.1447B)X_t = (1 + 0.8379B^{12})Z_t$$

$$\text{where } Z_t \sim N(0, 1.768e^{-6})$$

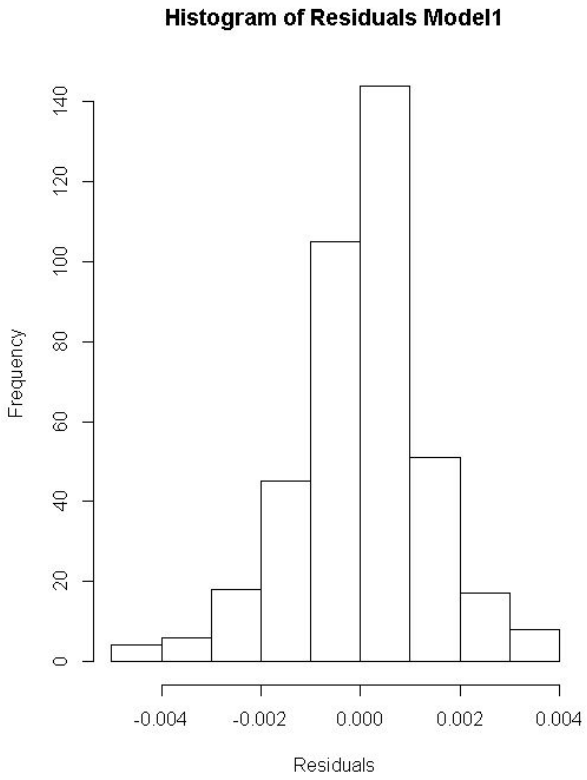
Thus, we now must check the roots of the polynomials to check the causality and invertibility of the models. We thus plot the roots (Appendix Figures 4 & 5) and see that all the

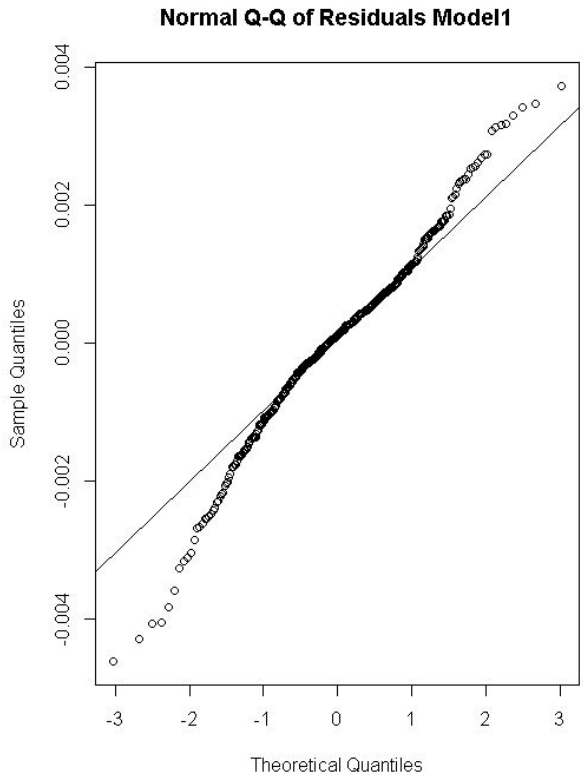
roots(color red) lie outside the unit circle for both models, and the absolute value of the coefficients of both models are all less than 1. Hence, we conclude that both models are causal and invertible.

5 Diagnostics

Now that we have identified two possible models and estimated their parameters, we must validate the assumptions of the models, which includes checking normality, independence, and constant variance of the errors.

To check if our residuals are normally distributed, we plot the histogram and normal Q-Q plot of the residuals in Figure 5.1 The histogram of the residuals of both models is symmetric about 0, and has a bell shape similar to the normal distribution. For the normal Q-Q plot, the points most lie around or on the predicted line, however there does appear to be some heavy tails. We then perform the Shapiro-Wilk test on our residuals at the $\alpha = 0.05$ level, which is seen in Table 5.1





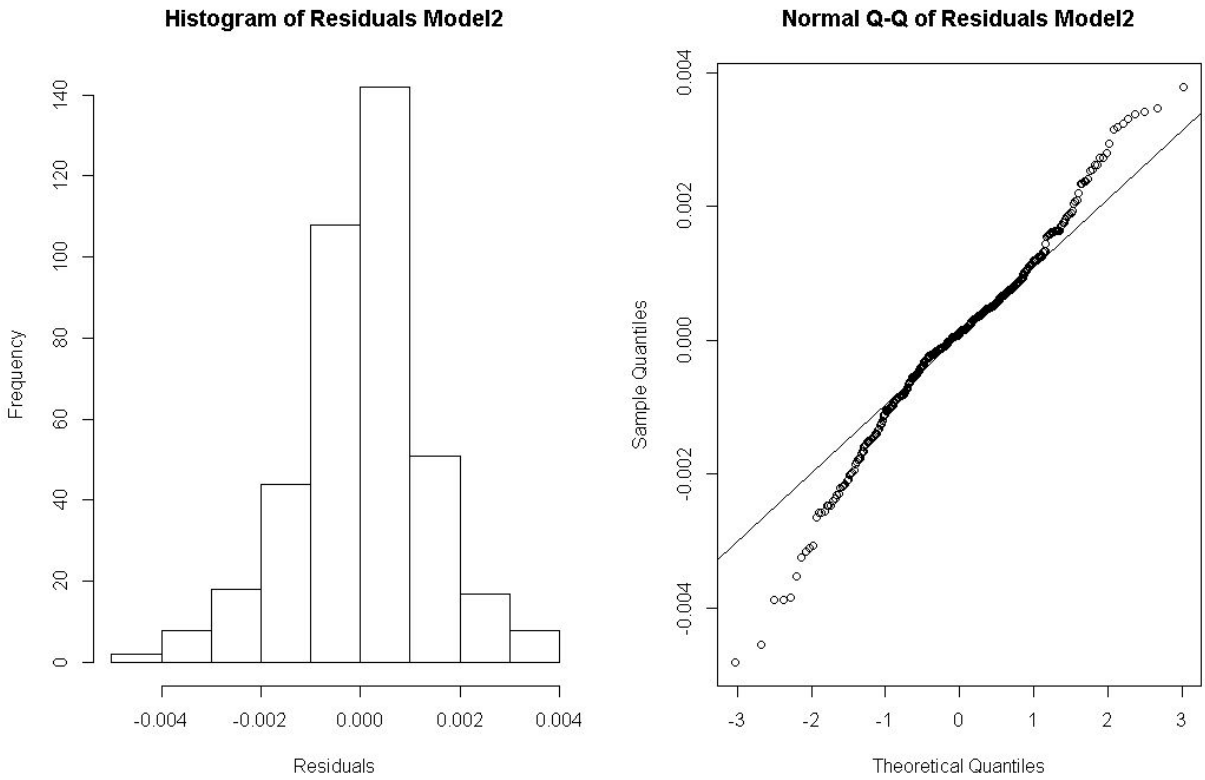


Figure 5.1

H_0 = residuals are normal

H_1 = residuals are not normal

	W.Statistics	P-value
Model 1	0.9800548	2.663467e-05
Model 2	0.9788833	1.479684e-05

Table 5.1

The residuals are not normal, due to the heavy tails. However, we tried fitting many different models with differing values of p,q,P, and Q and still came up with non-normal residuals. Hence, this problem cannot be solved with basic time series techniques, and hence requires other more complicated models to be modeled accurately.

Next, we check if the residuals are serially correlated with their own lagged values. Thus, we perform the Ljung-Box test, as well as the Box-Pierce test, on the residuals at the $\alpha = 0.05$ level on our models

H_0 = Residuals are serially uncorrelated

H_a = Residuals are serially correlated

	Model 1 P-value	Model 2 P-value
Box-Pierce	0.2004331	0.05128922
Ljung-Box	0.1842567	0.04538893

Table 5.2

The results in table 5.2 above show that the p-value > 0.05 in only the test for model1, and hence we do not reject the assumption of serially uncorrelated residuals for model1. However, for model2, one of the p-values is below 0.05, and hence we accept the alternative hypothesis that the residuals of model2 are serially correlated.

We now must check if the residuals in our model have constant variance, since they must do so in order for our model estimation and prediction to be accurate. Thus, we check for heteroskedasticity, or the violation of constant variance of errors, by analyzing the ACF and PACF of the squared residuals. These plots should lie within 95% of the white noise limits. The results in Figure 5.2 shows that for both models, most of our values lie within the bounds. The ones that exceed the bounds can be seen as outliers in our dataset. Hence, we can conclude that the constant variance of the errors is not violated, and thus heteroskedasticity is not a problem for our models.

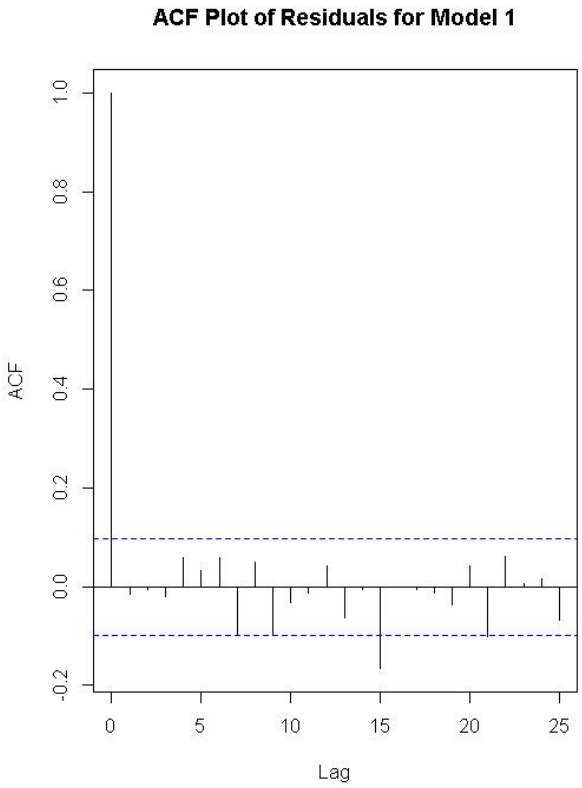
Thus, after the diagnostic checks, we choose model1 over model2 as our final model since model2 failed the serial correlation checks, which model1 did not.

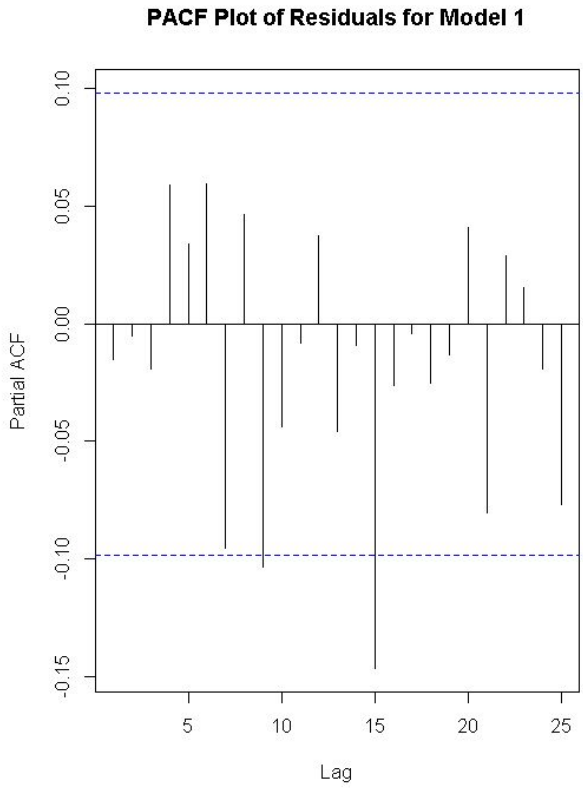
Let X_t be the transformed and differenced data. Then, $X_t = \nabla \nabla_{12} V_t^{-0.02020202}$

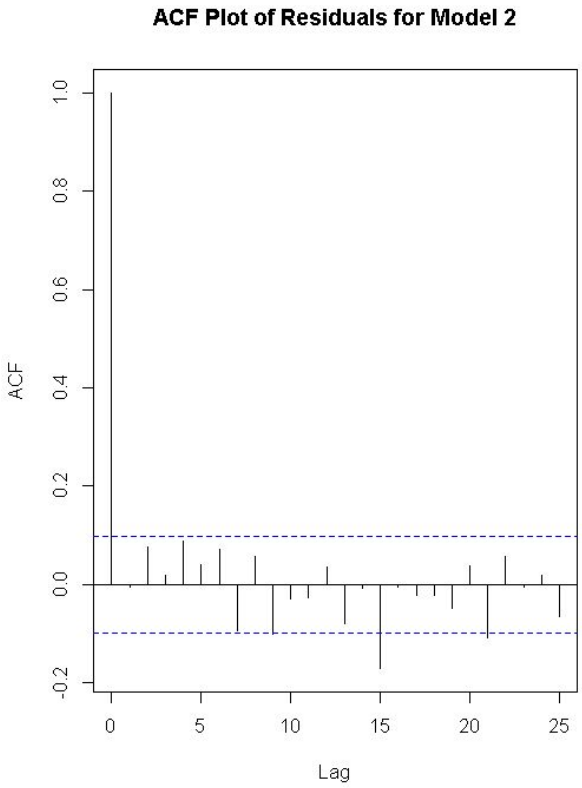
Final Model = $SARIMA(2,1,1) \times (0,1,1)_{12}$

$$(1 - 0.4794B - 0.1636B^2)X_t = (1 + 0.6187B)(1 + 0.8379B^{12})Z_t$$

$$where\ Z_t \sim N(0, 1.768e^{-6})$$


ACF Plot of Residuals for Model 1
ACF
Lag
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25


PACF Plot of Residuals for Model 1
Partial ACF
Lag
-0.15
-0.10
-0.05
0.00
0.05
0.10
5
10
15
20
25


ACF Plot of Residuals for Model 2
ACF
Lag
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25

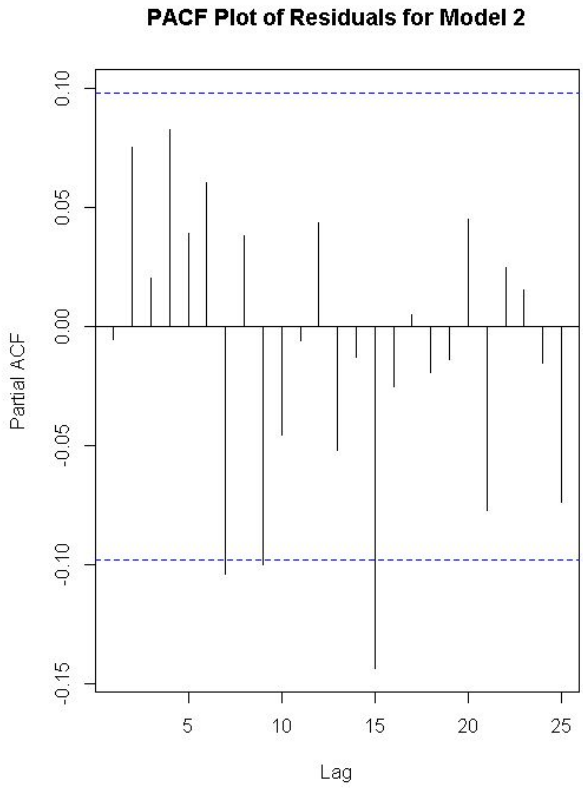

PACF Plot of Residuals for Model 2
Partial ACF
Lag
-0.15
-0.10
-0.05
0.00
0.05
0.10
5
10
15
20
25

Figure 5.2

6 Forecasting

This is the primary goal for assembling our model. We now forecast 10 values ahead—the female unemployment in the US from 2/1/1981 until 12/1/1981 in monthly intervals. Figure 6.1 shows the transformed data and Figure 6.2 shows the original data. The 10 red dots represent the 10 forecasted values; the blue lines represent the boundaries of the confidence interval. A clearer and closer view of the section can be seen in Figure 6.3. The true observed values are the 10 green dots.

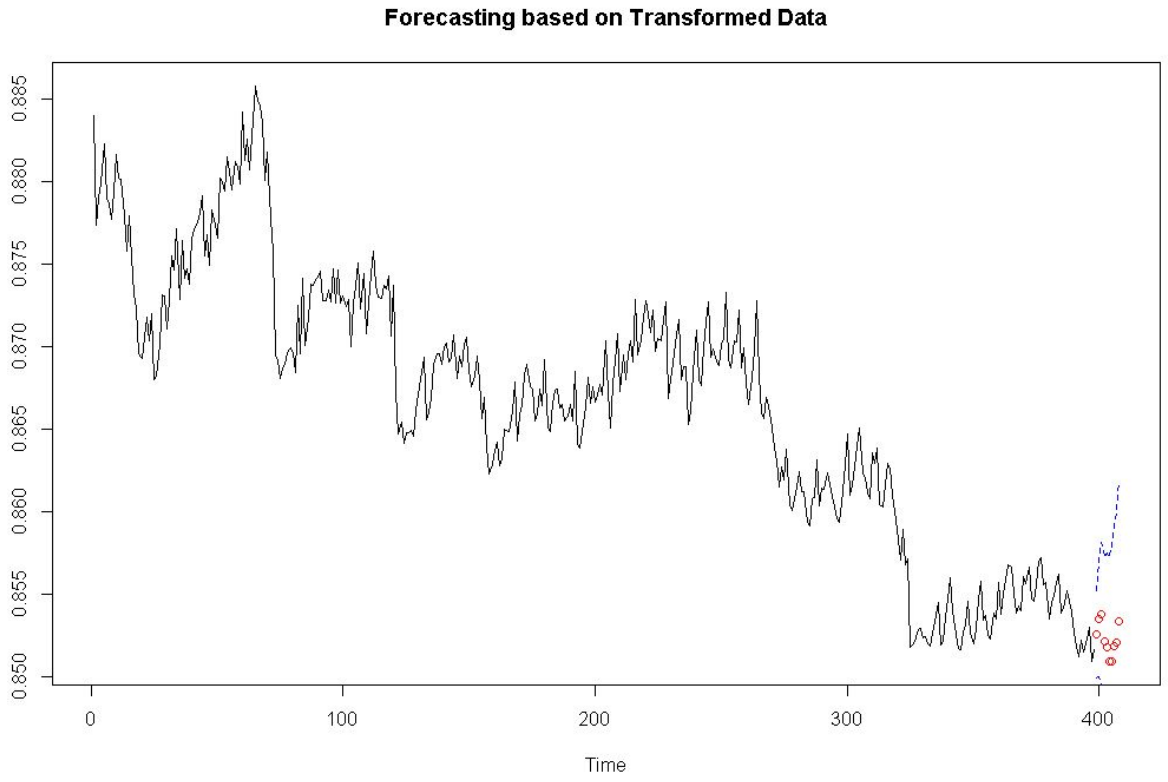


Figure 6.1

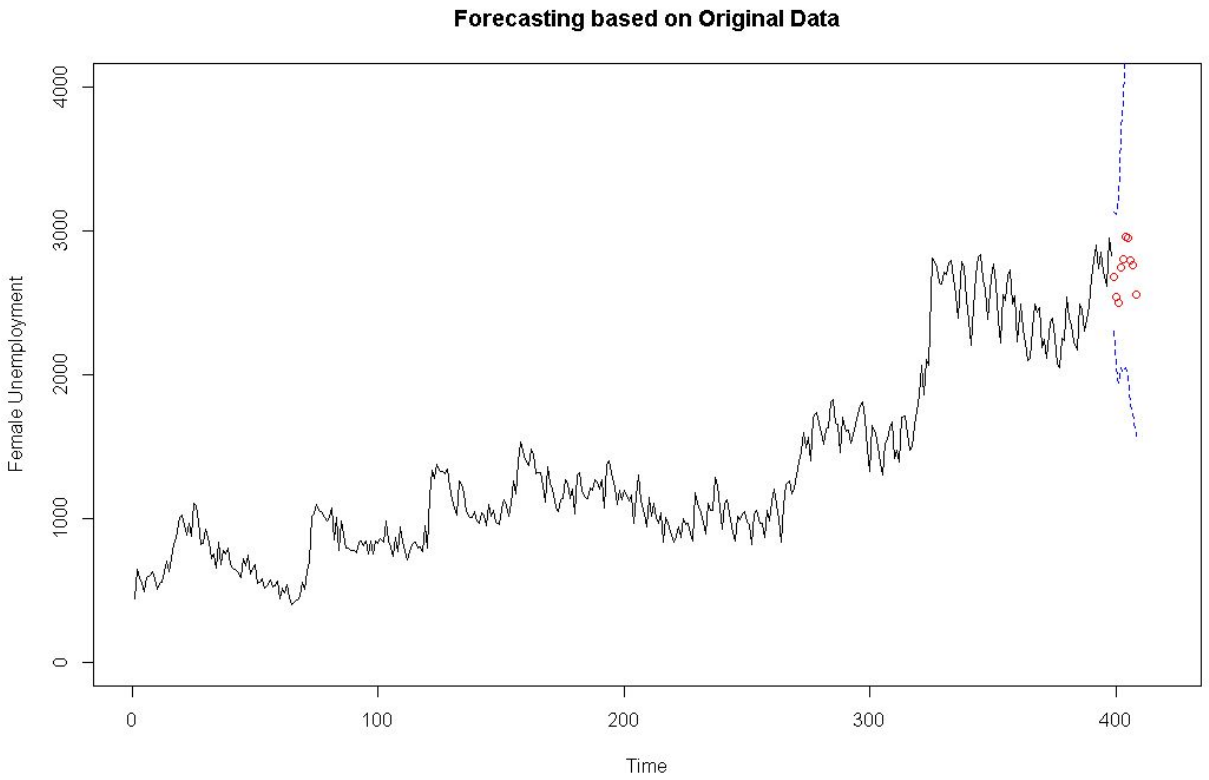

Forecasting based on Original Data
Female Unemployment
0
1000
2000
3000
4000
0
100
200
300
400
Time

Figure 6.2

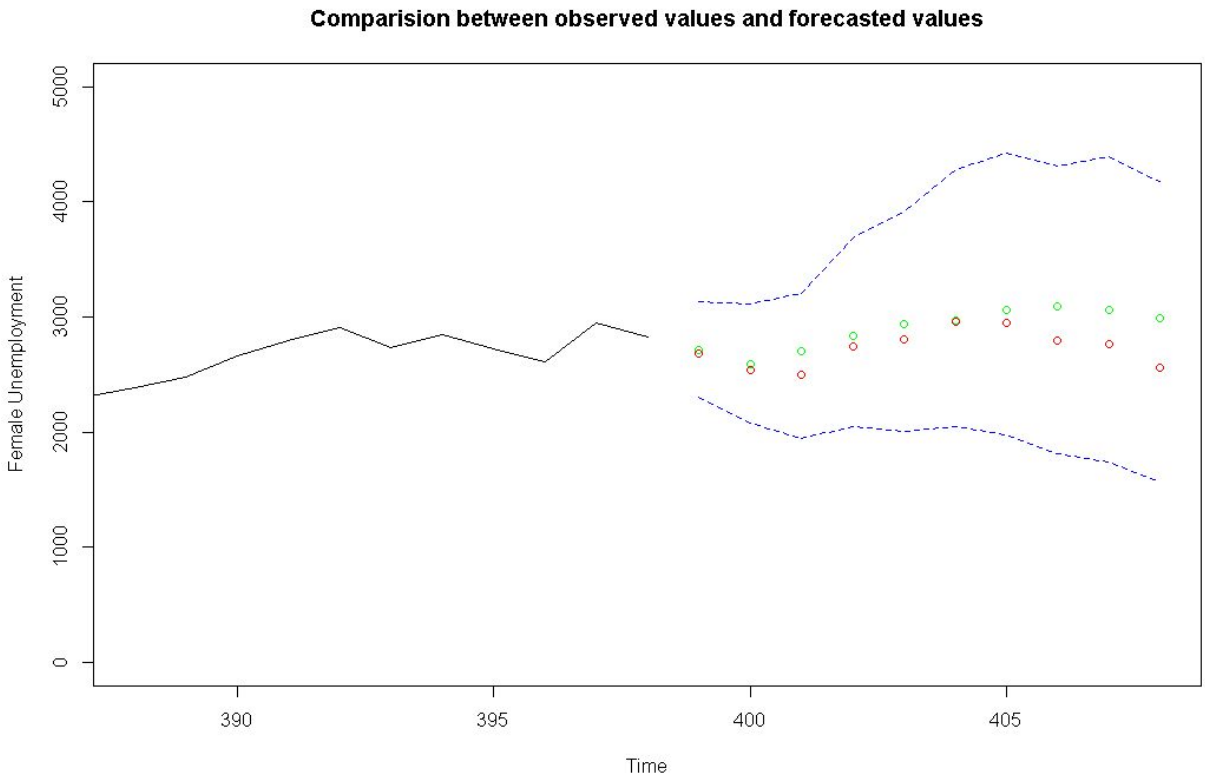


Figure 6.3

Our final model captures the trend and seasonality of the data, since it follows the same pattern that the true observed values follow. Additionally, our predicted values are close to the true observed values, but they do start to diverge as the prediction lead grows. The one-step ahead prediction is practically spot on, but the tenth-step ahead prediction has some error. This is likely due to the fact that our residuals in our model were not normally distributed. Thus, our final model is reasonable, but has its limitations.

7 Conclusion

Our goal is to construct a time series model that can explain the unemployment figures of females in the United States starting from January 1st 1941 and predict its future value up to 10 months ahead. We observed an upward trend and seasonality in the data, likely due to the increasing US population and the seasonal behavior of jobs and layoffs. After making our data stationary, we fit it into several models, and carried out the model selection process. Then, we

ran model diagnostic checks on them and selected the best model. Let X_t be the transformed and differenced data. Then, $X_t = \nabla \nabla_{12} V_t^{-0.02020202}$

$$\text{Final Model} = SARIMA(2,1,1) \times (0,1,1)_{12}$$

$$(1 - 0.4794B - 0.1636B^2)X_t = (1 + 0.6187B)(1 + 0.8379B^{12})Z_t$$

$$\text{where } Z_t \sim N(0, 1.768e^{-6})$$

Then we forecasted the monthly female unemployment figures from March 1st 1941 to December 1st 1981. Our forecasted values are close to the observed values, but there is an error that grows as the prediction is further out. Thus, this proves that our final model is reasonable for predicting ahead only in the near future, but still has its limitations and can be improved on by considering other more complicated time series techniques.

8 References

Adrian Trapletti and Kurt Hornik (2018). tseries: Time Series Analysis and Computational Finance. R package version 0.10-45.

David Stoffer (2017). astsa: Applied Statistical Time Series Analysis. R package version 1.8. https://CRAN.R-project.org/package=astsa

Hyndman R, Athanasopoulos G, Bergmeir C, Caceres G, Chhay L, O'Hara-Wild M, Petropoulos F, Razbash S, Wang E, Yasmeen F (2018). _forecast: Forecasting functions for time series and linear models_. R package version 8.4, <URL: http://pkg.robjhyndman.com/forecast>.

Hyndman RJ, Khandakar Y (2008). "Automatic time series forecasting: the forecast package for R." _Journal of Statistical Software_, *26*(3), 1-22. <URL: http://www.jstatsoft.org/article/view/v027i03>.

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Venables, W. N. & Ripley, B. D. (2002) Modern Applied Statistics with S. Fourth Edition. Springer, New York. ISBN 0-387-95457-0

https://datamarket.com/data/set/22lq/monthly-us-female-20-years-and-over-unemployment-figur

es-103-1948-1981#!ds=22lq&display=line

9 Appendix

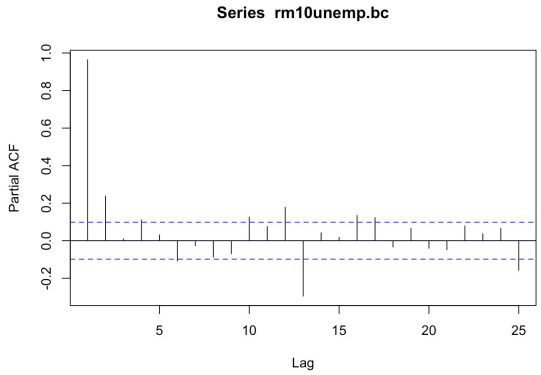


Fig. 3.2 - Box-Cox transformed PACF

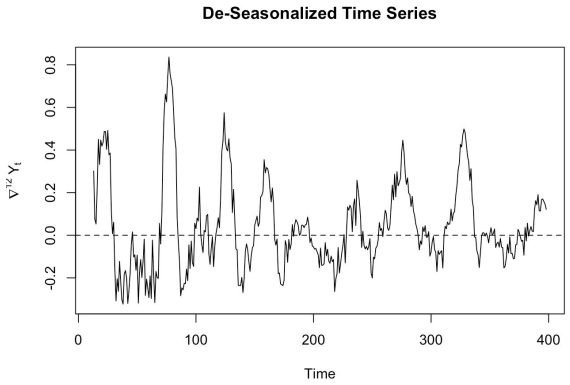


Fig. 3.3 - Graph of the 1 year deseasonalized time series. Shows it's zero mean property.

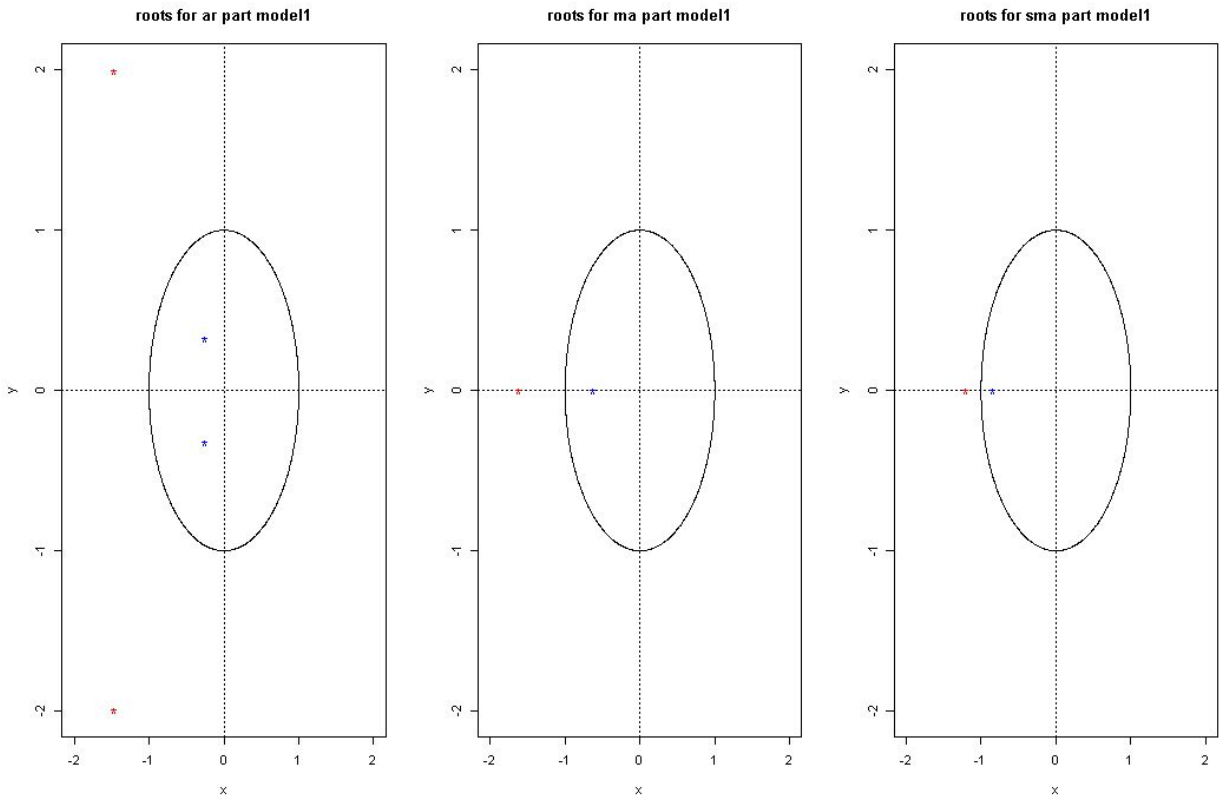

roots for ar part model1
roots for ma part model1
roots for sma part model1
x
y

Figure 4

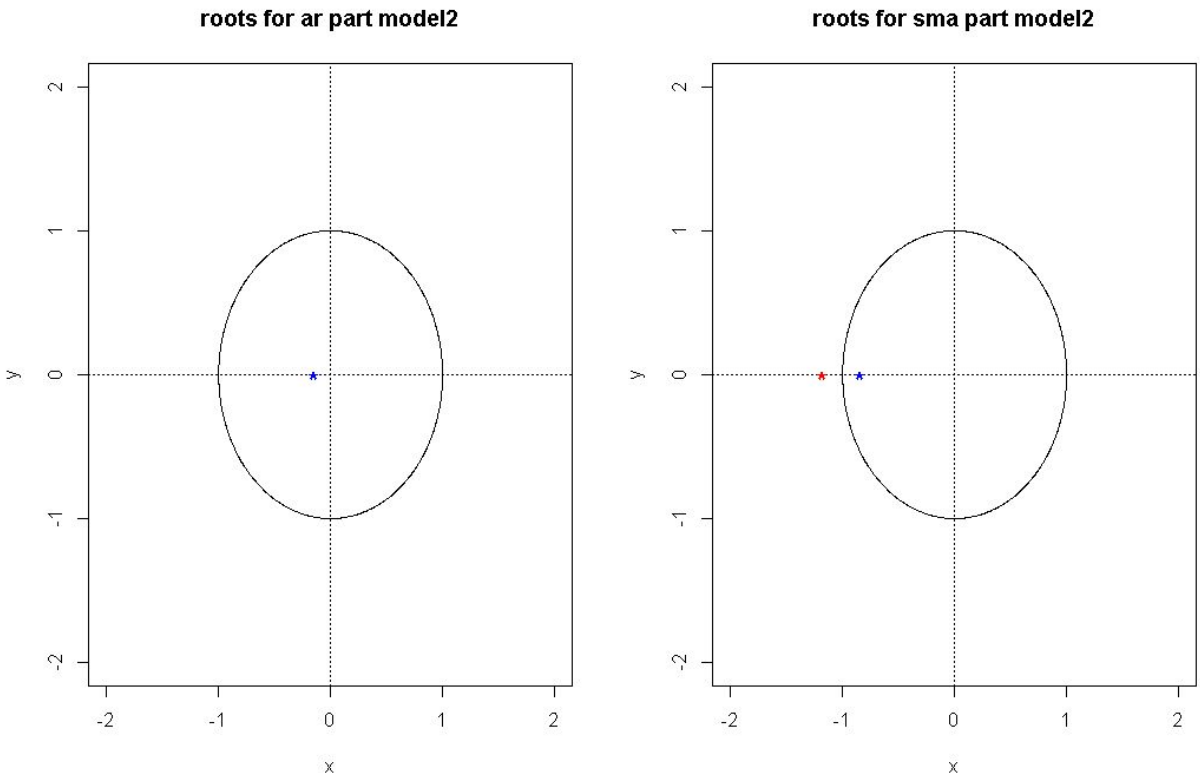


Figure 5

R Code

```
---
title: "PSTAT 174 Final Project"
author: "Evan Azevedo, Manpal Sidhu, Jay Singh, Allen Wang“
date: "12/5/2018"
output: html_document
---



```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
libraries <- c("ggplot2", "MASS", "astsa", "tseries", "forecast")
lapply(libraries, library, character.only = TRUE)
```
```

```
comp="/Users/SSMS1303-27/Desktop"
jay="/Users/Jay/Documents/LVL_5_HACKING/SerialTimeLords"
evan="/Users/evanazevedo/Documents/School/PSTAT 174/SerialTimeLords/"
manpal="/Users/Rajwinder/Desktop"

setwd(jay)
```

```{r plot analyze}
data <- read.csv("monthly-us-female-20-years-and-o.csv", header=TRUE,
col.names=c("Month", "Female Over 20 Unemployment"))
# Convert into time series
unemp = ts(data[,2], start = c(1948,1), frequency = 12)
ts.plot(unemp)
```

There seems to be an upward trend that appears somewhat exponential. There is also a seasonal component as we see large peaks and troughs around every 2 years. There are sharp changes throughout the graph but especially in the last quarter we notice very sharp ups and downs.

```{r modeling}
decom = decompose(unemp) #get decomposition plot
decom$seasonal
plot(decom) #plot decomposition

# Box Cox Txf
t = 1:length(unemp)
fit = lm(unemp ~ t)
bcTransform = boxcox(unemp ~ t, plotit=TRUE)
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

```
#unemp.bc = (unemp^lambda)
unemp.bc = ((unemp^lambda)-1)/lambda

rm10unemp.bc = ts(unemp.bc[1:(length(unemp.bc)-10)]) #remove last 10 values for forecasting
rm10unemp = ts(unemp[1:(length(unemp)-10)])
# Compare the Box Cox transformed with the original
ts.plot(rm10unemp,main = "Original data",ylab = expression(X[t]))
ts.plot(rm10unemp.bc,main = "Box-Cox tranformed data", ylab = expression(Y[t]))
var(rm10unemp)
var(rm10unemp.bc) # Look at how much better the variance is!

# Plot acf and PACF
acf(rm10unemp.bc)
pacf(rm10unemp.bc) # We should difference at lag 12 and lag 1

# In conclusion: note the spike in ACF at 1.
# This dataset has a seasonality component of 12 months, or 1 year

rm10unemp.bc.12 = diff(rm10unemp.bc,lag = 12)
rm10unemp.bc.12.1 = diff(rm10unemp.bc.12,lag = 1)
rm10unemp.bc.12.1.1 = diff(rm10unemp.bc.12.1,lag=1)
var(rm10unemp.bc) #base variance
var(rm10unemp.bc.12) #variance decreased
var(rm10unemp.bc.12.1) #variance decreased
var(rm10unemp.bc.12.1.1) #variance increased
#no further differencing, since variance increased

plot(rm10unemp.bc.12,main = "De-Seasonalized Time Series",ylab =
expression(nabla^{12}~Y[t]))
abline(h = 0,lty = 2)

op = par(mfrow = c(1,2))
acf(rm10unemp.bc.12,lag.max = 60,main = "")
```

```
pacf(rm10unemp.bc.12,lag.max = 60,main = "")
title("De-Seasonalized Time Series", line = -1, outer=TRUE)

ts.plot(rm10unemp.bc.12.1,main = "De-trended/seasonalized Time Series",ylab =
expression(nabla~nabla^{12}~Y[t]))
abline(h = 0,lty = 2)

#op = par(mfrow = c(1,2))
acf(rm10unemp.bc.12.1,lag.max = 60,main = "")
pacf(rm10unemp.bc.12.1,lag.max = 60,main = "")
title("De-trended/seasonalized Time Series",line = -1, outer=TRUE)

stationdata = rm10unemp.bc.12.1
#adf.test(y12)

#identifying model, and P,Q
acf(stationdata,lag.max = 60,main="")
pacf(stationdata,lag.max = 60,main="")
#P=0,Q=1
#check lags in multiples of 12s, since 12 is the seasonality
#acf cuts off after lag 12
#pacf cuts off after lag 24

#now get p,q
acf(stationdata,lag.max = 11,main="")
pacf(stationdata,lag.max = 11,main="")
#p=7,q=0
#check lags 11 or less
#acf tails off exponentially
#pacf cuts off after lag 7
#test at these values or lower
AICc<-numeric()
for (p in 0:7) {
```

```
  AICc<-c(AICc,astsa::sarima(rm10unemp.bc,p,1,0,2,1,1,12,details=FALSE)$AICc)
}
AICc<-matrix(AICc,nrow = 8,byrow = TRUE)
AICc
#aic chooses p=1,q=0 as first smallest
#p=4,q=0 as second smallest
BIC<-numeric()
for (p in 0:7) {
  BIC<-c(BIC,astsa::sarima(rm10unemp.bc,p,1,0,2,1,1,12,details=FALSE)$BIC)
}
BIC<-matrix(BIC,nrow = 8,byrow = TRUE)
BIC
#bic chooses p=1,q=0 as first smallest
#p=4,q=0 as second smallest

#Based on AICc and BIC values, select 2 models
#Model 1: SARIMA(2,1,1)(0,1,1)12
#Model 2: SARIMA(1,1,0)(0,1,1)12

fit1 <- arima(rm10unemp.bc, order = c(1,1,0), seasonal = list(order=c(2,1,1),period = 12),
method = "ML")
fit1
```{r}
plot.roots(NULL, polyroot(c(1,-0.1387)), main = "roots for ar part")
plot.roots(NULL, polyroot(c(1,-0.1097,-0.1127)), main = "roots for sar part")
plot.roots(NULL, polyroot(c(1,-0.7423)), main = "roots for sma part")
#absolute value of all coefficients is less than 1

#Model 1 EQN:

fit2 = arima(rm10unemp.bc, order = c(4,1,0), seasonal = list(order=c(2,1,1),period = 12), method
= "ML")
fit2
```
```

```
plot.roots(NULL, polyroot(c(1,-0.1315,0.0664,0.0130,0.0796)), main = "roots for ar part")
plot.roots(NULL, polyroot(c(1,-0.1121,-0.1184)), main = "roots for sar part")
plot.roots(NULL, polyroot(c(1,-0.7398)), main = "roots for sma part")
#absolute value of all coefficients is less than 1

#Model 2 EQN:

#now for diagnostic checks
resid1<-residuals(fit1) #residuals for model 1
resid2<-residuals(fit2) #residuals for model 2

#histogram and qq plot for Model 1
hist(resid1)
qqnorm(resid1)
qqline(resid1)

#histogram and qq plot for Model 2
hist(resid2)
qqnorm(resid2)
qqline(resid2)

Shap<-matrix(c(shapiro.test(resid1)$statistic, shapiro.test(resid1)$p.value,
shapiro.test(resid2)$statistic, shapiro.test(resid2)$p.value), nrow = 2, byrow = TRUE)
#greater than 0.05 means we pass the test, and errors are indeed normal
rownames(Shap)<-c("Model1","Model2")
colnames(Shap)<-c("W_Statistic","P-value")
Shap<-data.frame(Shap)
Shap

#now for independence/serial correlation checks
b1 <- Box.test(resid1, lag=12, type = "Box-Pierce", fitdf = 2)$p.value
b2 <- Box.test(resid1, lag=12, type = "Ljung-Box", fitdf = 2)$p.value
```

```
b1
b2
#want b1 and b2 to be above 0.05

b3 <- Box.test(resid2, lag=12, type = "Box-Pierce", fitdf = 2)$p.value
b4 <- Box.test(resid2, lag=12, type = "Ljung-Box", fitdf = 2)$p.value
b3
b4
#want b3 and b4 to be above 0.05
boxT<-matrix(c(b1,b2,b3,b4), nrow = 2, byrow = FALSE)
rownames(boxT)<-c("Box-Pierce","Ljung-Box")
colnames(boxT)<-c("Model1_Pvalue", "Model2_Pvalue")
boxT
#now check for constant variance of the residuals
#model1
acf(resid1)
pacf(resid1)
#model2
acf(resid2)
pacf(resid2)

#pick Model (1 or 2) since either AICc/BIC smaller, or one of them fails diagnostic checks

#forecasting based on selected final model
fitfinal = fit1 #fit1 or fit2 is the final model

pred.tr<-predict(fitfinal,n.ahead = 10)
U.tr= pred.tr$pred + 2*pred.tr$se #upperbound for transformed data CI
L.tr= pred.tr$pred - 2*pred.tr$se #lowerbound for transformed data CI
ts.plot(rm10unemp.bc, xlim=c(1,length(rm10unemp.bc)+10), main = "Forecasting based on
Transformed Data", ylab = "")
lines(U.tr, col = "blue", lty = "dashed")
lines(L.tr, col = "blue", lty = "dashed")
```

```
n = length(rm10unemp.bc)
coors = (n+1):(n+10)
notcoors = (1):(n)
points(coors, pred.tr$pred, col = "red")

#now have to back transform to get actual predictions of original time series
pred.orig<- pred.tr$pred^(1/lambda) #backtransform to get prediction of original data
U = U.tr^(1/lambda)
L = L.tr^(1/lambda) #backtransform of CI
#now plot the forecasts of original data

origdata = ts(data[,2], start = c(1948,1))
ts.plot(origdata, xlim = c(1,length(origdata)+10), ylim = c(0,4000), main = "Forecasting based on
Original Data", ylab = "Female Unemployment")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points(coors, pred.orig, col = "red")
points(notcoors, origdata[notcoors], col = "black")
points(coors, origdata[coors], col = "green")

#zooming in to the predictions
inv1 = length(origdata) - 20
inv2 = length(origdata)
ts.plot(origdata, xlim =c(inv1,inv2), ylim = c(0,5000), main = "Comparision between observed
values and forecasted values", ylab = "Female Unemployment")
points(coors, origdata[coors], col = "green")
points(notcoors, origdata[notcoors], col = "black")
lines(U, col = "blue", lty = "dashed")
lines(L, col = "blue", lty = "dashed")
points(coors, pred.orig, col = "red")
#close to observed value, within the confidence interval bounds, and hence is good forecasting
```