Saba Paya
ID: 104483616
Course: Statistics 170

Forecasting the Monthly Unemployment Rate of California

I. Introduction

The general notion of the unemployment rate is that it is affected by the global market. It makes sense when you think about it, that when a country's economy is doing well in comparison to others, the unemployment level will be lower as more people are employed because there is not so much worry about money and paying for labor. However, when the economy takes a hit, that is when you see levels of unemployment change as they tend to increase. This can be seen by looking at U.S. economy and the recession of 2008 which lead to great levels unemployment.

In order to understand this trend that occurs of lower unemployment rates during a booming economy and higher unemployment rates during a fall economy, I will look at two variables, the production of goods and the education level, and zone in on specifically California's unemployment rates to determine whether we can predict future unemployment rates. The primary purpose of this paper is to compare several time series models to see which one gives the best forecasting performance in the short and long run.

In this paper, I will forecast the unemployment rate in the state of California. I hypothesize that there is a negative relationship between the number of goods produced per month, and the monthly unemployment rate in California. Further, I expect that the higher the education level achieved by an individual, the less likely they are to be unemployed. Seasonal effects are expected in the summer and winter as there are more opportunities for employment during those months due to the Holiday's such as Christmas that tend to increase the production of goods and lead to more employment opportunities. As the economy decreases and a recession hits, it is anticipated that the unemployment rate will increase, thus an increasing trend is expected. However, when the economy is doing better, a decreasing trend is expected as unemployment is anticipated to decrease.

In order to model the unemployment rate and its relation with the other variables, I collected data from FRED [1]. Variable unemp [2] measures Unemployment Rate in California. Variable goods [3] measures the Production: Consumer Durable Goods. Variable edu [4] measures the Employment-Population Ratio: Bachelor's Degree and Higher, 25 years and over. At each of the corresponding references for the three variables, we can see that it says the data are not seasonally adjusted.

II. The Data

Here, I describe the variables used for the analysis. I have given a short name and description to each of the variables used. I also detail the units, number of observations, time period, and frequency of each of the variables.

The variable unemp [2] measures the percent of unemployment in California per month. The length of the series is n=300. The time period for this variable is from January 1992 to December 2016, and the frequency is 12 as it is monthly data.

The variable goods [3] measures the Index 2010=100 for Industrial Production: Durable Consumer Goods per month. The length of the series is n=300. The time period for this variable is from January 1992 to December 2016, and the frequency is 12 as it is monthly data.

The variable edu [4] measures the percent of Employment-Population Ratio: Bachelor's Degree and Higher, 25 years and over per month. The length of the series is n=300. The time period for this variable is from January 1992 to December 2016, and the frequency is 12 as it is monthly data.

Table 1 contains a summary that makes it easier to see the variable descriptions and short names.

Table 1: Summary of the Data

Short Name	Variable Description	Time Period	Length	Mean	Standard Deviation
unemp	Unemp measures the percent of Unemployment in California per month.	January 1992 – December 2016	n=300	$\bar{x}$=7.464	s=2.216321
goods	Goods measures the index 2010=100 of Industrial Production of Durable Consumer Goods per month.	January 1992 – December 2016	n=300	$\bar{x}$=104.1379	s=15.6428
edu	Edu measures the percent of employment to population ratio of people that have a bachelor's degree or higher and are over 25 years per month.	January 1992 – December 2016	n=300	$\bar{x}$=76.031	s=2.49954

III. Description of the Data

From Figure 1, we can see that there is a trend in the target variable of unemp. The points decrease, then increase, then decrease, and so forth exhibiting a curved pattern. The curve in the data may be due to the increase and decrease of values accelerating and decelerating over time. The time plot of the target variable in Figure 1 looks to have cyclic movements as there seem to be cycles in the pattern. There is also a seasonal pattern, but that pattern is not so constant as the amplitudes and length of the patterns change over time. Figure 1 reveals that the unemployment rate in California was at its highest around the year 2010. This may be due to the recession that occurred around that time which most likely lead to a decrease in the production of goods and economic activity. Further, we would not need to transform the target variable to make it series variance stationary as it already looks to be stationary, and applying transformations to the variable don't seem to change the time series plot in any drastic way. The time plot for goods has an increasing trend followed by a sudden shift around 2008 where the production of goods drops significantly and then rises again. This shift can also be attributed to the recession that happened around that time. The time plot for education has a fairly constant continuous decreasing trend from 1992 to 2016.

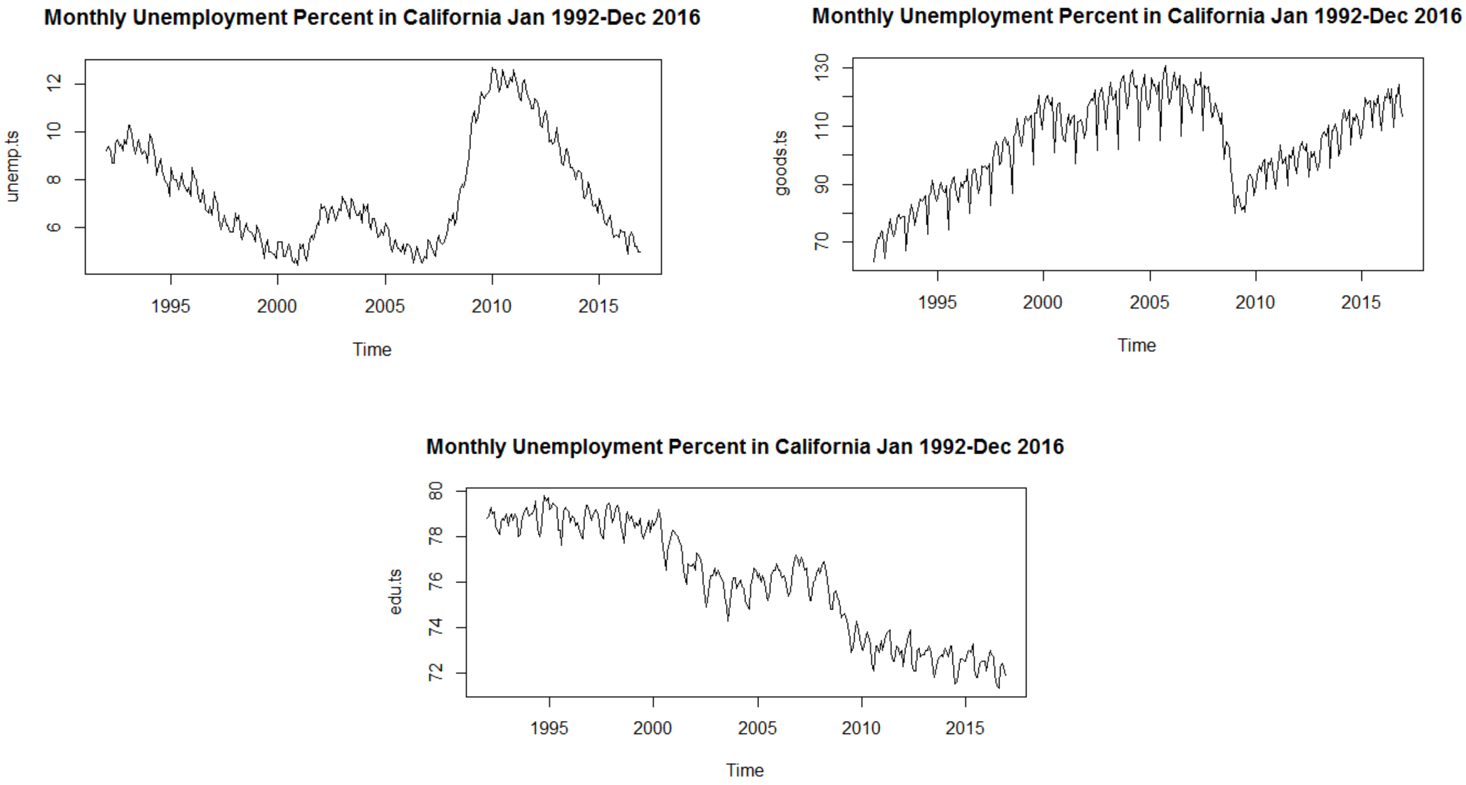


Figure 1: Time Series for Each Variable, unemp(top left), goods(top right), edu(bottom)

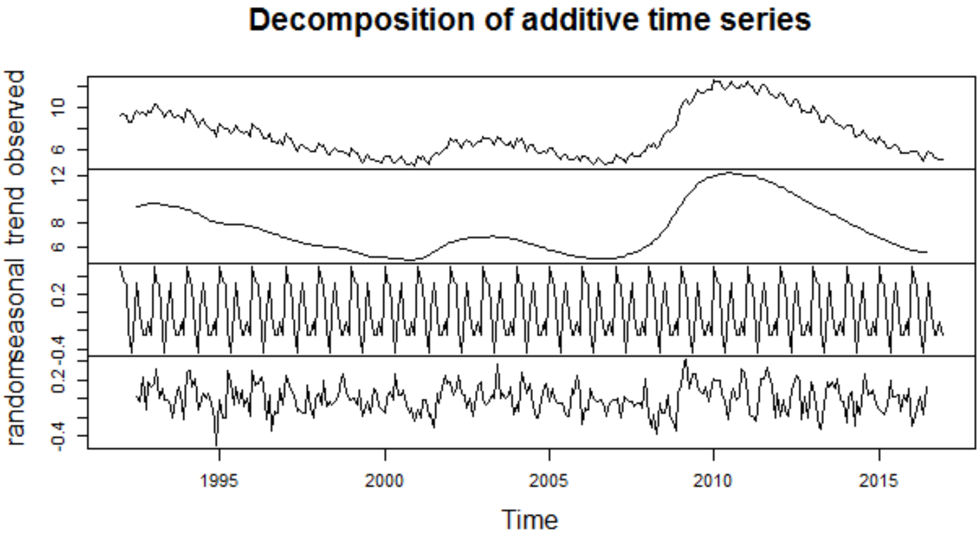


Figure 2: Additive Decomposition of Raw Unemp Data

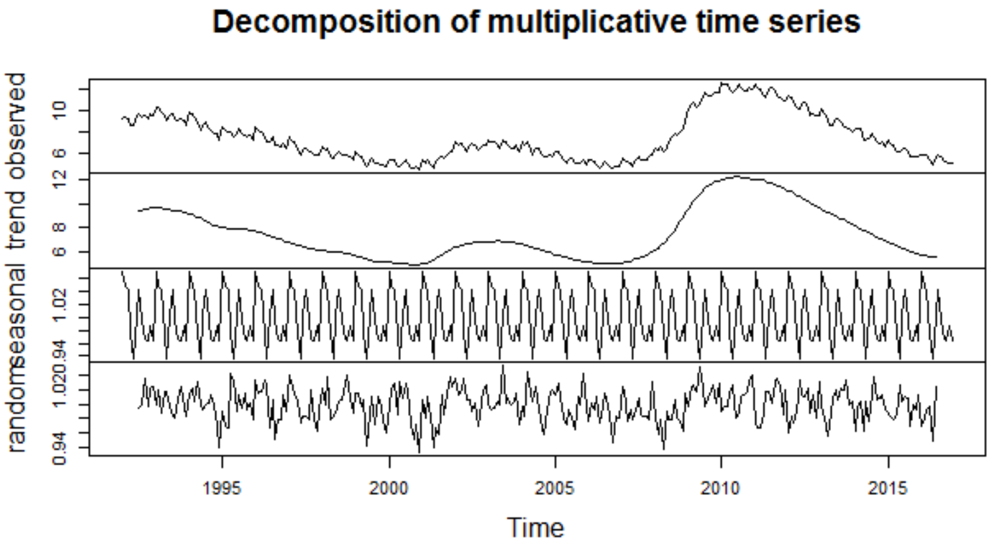


Figure 3: Multiplicative Decomposition of Raw Unemp Data

Figure 2 shows the additive decomposition of the unemp series and Figure 3 reflects the multiplicative decomposition of the same series. Figures 2 and 3 show the observed series, the smoothed trend, seasonal pattern of the random series, and the random component. Both Figures 2 and 3 show a changing trend as the series shifts from decreasing to increasing. The two decomposition models are very similar in their observed, trend, and seasonal series'. However, they vary in their random series. The variance of the random component in Figure 2 showcases less variability and has a smaller range for the randomness of the data. In Figure 3, we can see that the randomness is more spread out as the range of the random component is greater. As such, I would say that the additive decomposition model is more appropriate for the data. Further, the seasonal series for unemp is increasing and decreasing over time suggesting that the additive model would be more appropriate.

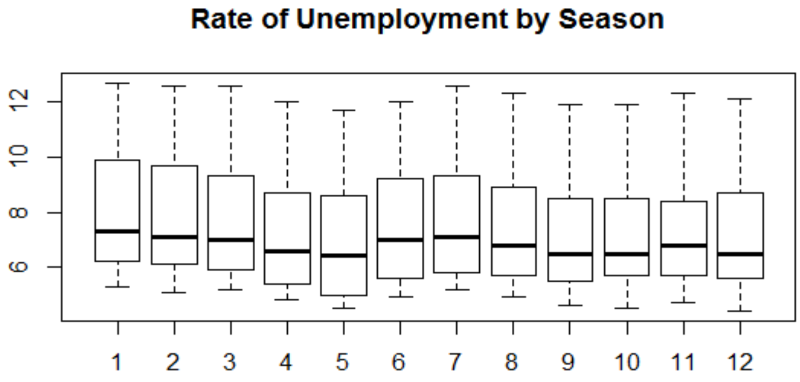


Figure 4: Seasonal Plot of Unemployment

The seasonal plot in Figure 4 reveals that unemployment is most prominent during the month of January as it has the highest mean unemployment rate and least prominent during the late fall and winter months of September, October, and December. These lower unemployment rates may be due to the increase in the production of goods during the fall and winter as people tend to purchase more in preparation for the holiday seasons of Thanksgiving, Black Friday, Hanukkah, and Christmas. This, in turn, leads to the hiring of seasonal employees thus lowering the unemployment rate. Further, the higher unemployment rate during the month of January may attributed to the decrease of people purchasing goods as the holidays are over creating a decrease in the production of goods and the layoff of employees and seasonal workers.

Autocorrelation Structure of the Random Component

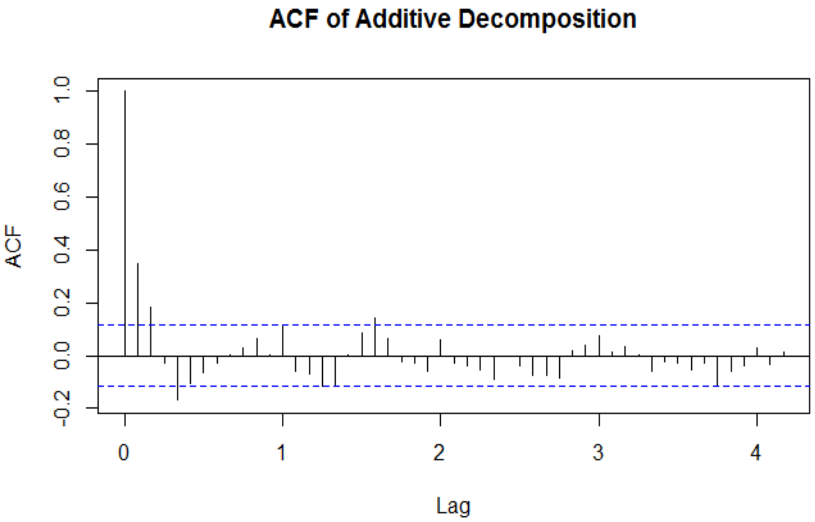


Figure 5: ACF of the Random Term of Additive Decomposition of Unemp (Y_t) in its raw form

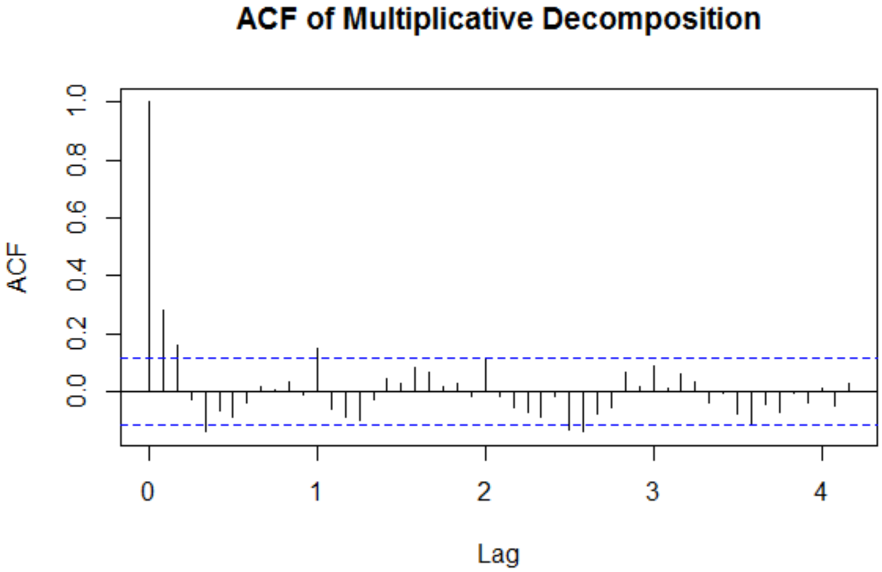


Figure 6: ACF of the Random Term of Multiplicative Decomposition of Unemp (Y_t) in its raw form

Looking at Figures 5 and 6, we can see that unemp has stationary correlation. There are significant lag points indicating that the data is not completely random. Further, there are no smooth patterns or seasonality present in the ACF plots indicating that there is a stationary correlation structure. Focusing on Figure 6 as this is the more appropriate model (the additive decomposition model) for the target variable, r_1, r_2, and r_4 are the statistically significant lags.

IV. ARIMA Modeling

In Section III, I concluded, using decomposition, that my untransformed series Y_t has a trend and some seasonality to it. In Figure 7, we can see that the ACF of Y_t is non-stationary as there is a smooth decreasing pattern. As such, I am going to use ARIMA modeling to difference Y_t ,the target variable, to make it stationary. In this section, I will look at the ACF of the differenced data to show mean stationarity. However, before I do any differencing to the untransformed data, I am going to take out the last 12 observations from the data as those values will be used to forecast the model later. In table 2, I have described each of the variables and their descriptions without the last 12 observations.

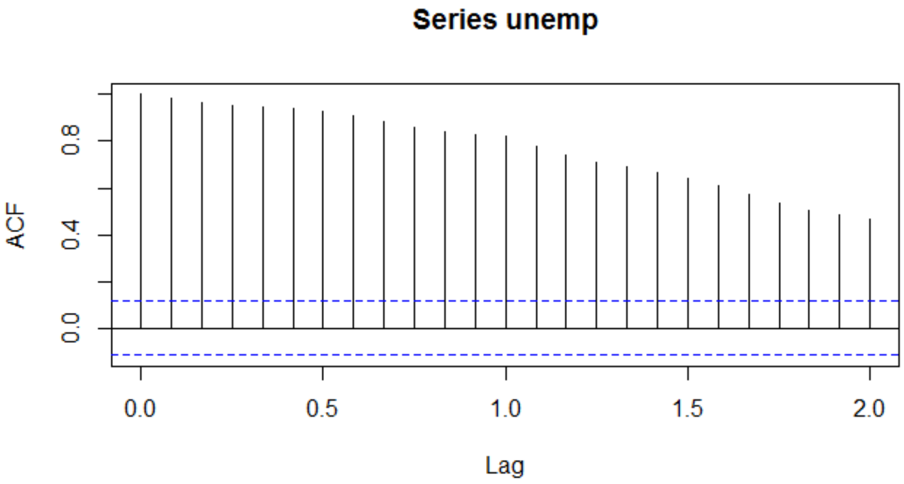


Figure 7: ACF of Y_t in its raw form

Table 2: Summary of Data with n-12 observations

Variable	Symbol	Definition	Time Period
unemp	Y_t	Monthly unemployment rate of California	January 1992 - December 2015
goods	X_t	Monthly production of durable goods	January 1992 - December 2015
edu	Z_t	Monthly percent of employment to population ratio	January 1992 - December 2015

In order to make the data stationary I, first, regularly differenced the data, but the corresponding ACF plot, Figure 8, showcased a slight seasonal trend and non-stationarity. Then, I seasonally differenced the data, but the ACF showed a smooth decrease and many significant lags indicating non-stationarity. I, finally, regularly then seasonally differenced, and my Y_t became stationary. I chose the regular and seasonal differencing to fit my model because it looked to be more stationary as the ACF is closer to that of white noise, Figure 9. Only regular differencing would have also worked fine, but regular and seasonal provided a plot that had better mean stationarity. The final untransformed and differenced time series of the stationary target variable is Y_t* = $(1\text{-}B^{12})(1\text{-}B)Y_{t.}$ The one star indicates that the variable has been differenced only. Figure 9 showcases the ACF of $(1\text{-}B^{12})(1\text{-}B)Y_t$ and reveals stationarity as there is no smoothness or seasonality present. The significant r_k's are r_2, r_3, r_5, and r_{12}.

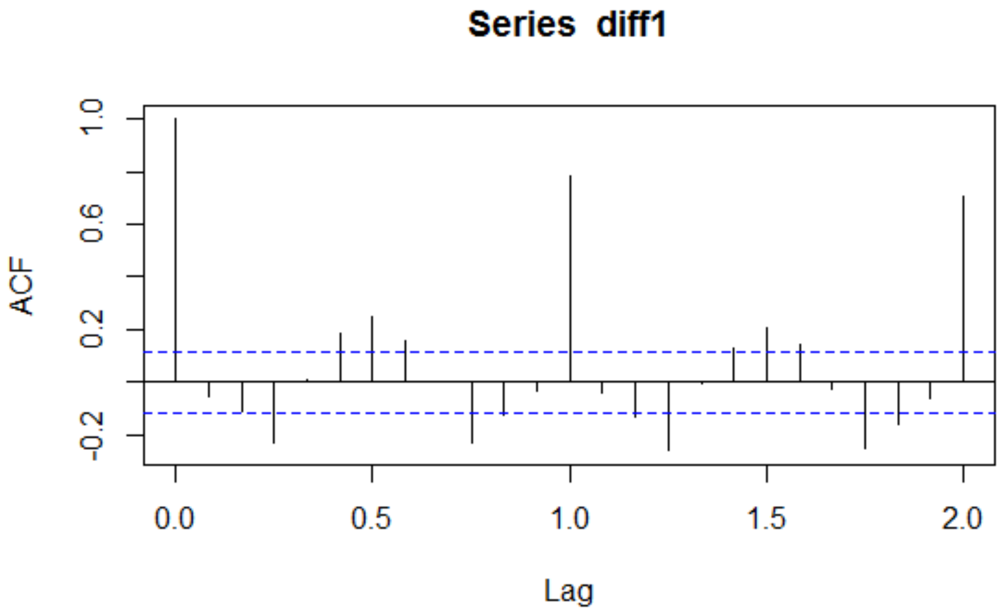


Figure 8: ACF of Y_t after Regular differencing $(1\text{-}B)Y_t$

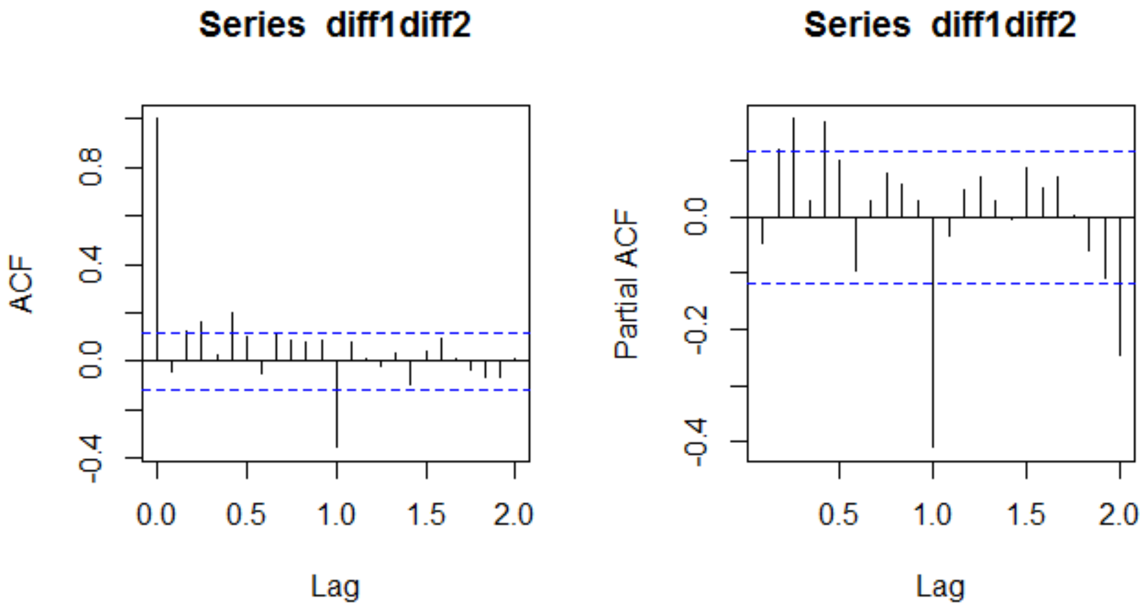


Figure 9: ACF of $(1\text{-}B^{12})(1\text{-}B)Y_t$ **Figure 10: PACF of $(1\text{-}B^{12})(1\text{-}B)Y_t$**

Looking at the regular part of the Y_t in Figures 9 and 10, there is no correlation with the regular part of the ACF and the PACF cuts off after lag 3 or 5. Further the PACF dies down rather quickly. Figures 9 and 10 showcase seasonality but no correlation for the regular differenced part. However, when I went to fit the best model, I identified an AR(1) and MA(2) to fit the regular part best. The ACF and the PACF of the regular differences, Figures 9 and 10, do not showcase that those should be the fitted models. For the seasonal part, the ACF dies down quickly and the PACF cuts off after lag 12. After fitting both AR(1) and AR(2), I concluded that AR(1) gives more white noise and a lower AIC. The PACF in figure 10 also dies down rather quickly and ACF, figure 9, cuts off at lag 12. As such, I tried MA(1) and MA(2) and concluded that MA(1) was a better fit, with the lower AIC. The final model that fit the data best was the ARIMA model $(2,1,1)(1,1,1)_{12.}$

I chose to work with the ARIMA model of $(2,1,1)(1,1,1)_{12}$ over the other models tested because it had the lowest AIC value and the lowest p-value from the t-tests on the model coefficients conducted. Figure 11 shows the ACF of the residuals of the fitted model of Y, and the ACF looks like the ACF of a white noise process which indicates that the model is good. To double check that the residuals are actually white noise, I did the Ljung-Box white noise test. The Ljung-Box test up to lags 6,12, 18, and 24 have p-values that are larger than 0.05. This suggests that the residual series in white noise and all p_k are significantly different from 0.

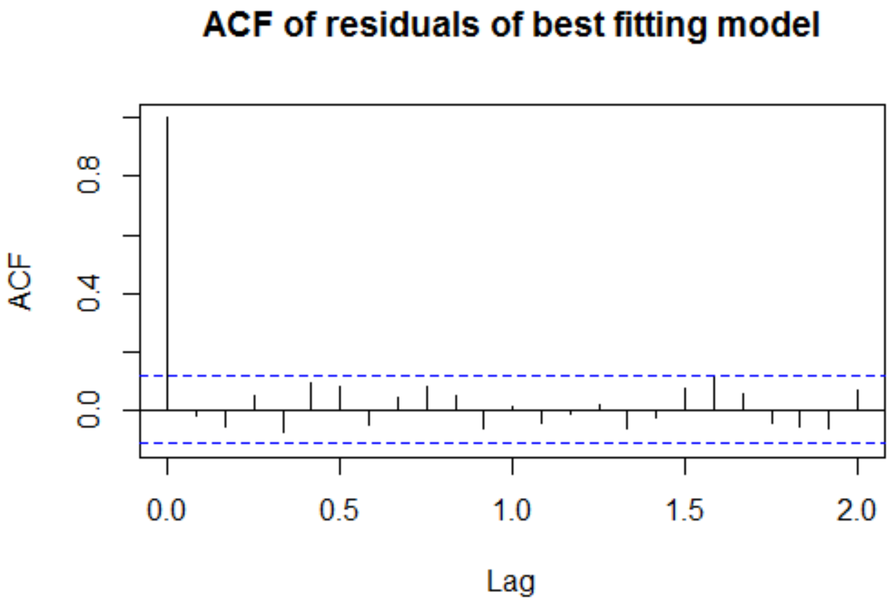


Figure 11: ACF of ARIMA$(2,1,1)(1,1,1)_{12}$

Symbolic notation: ARIMA$(2,1,1)(1,1,1)_{12}$.

Polynomial notation:
$(1-\alpha_{12}B^{12})(1-\alpha_1B-\alpha_2B^2)(1-B^{12})(1-B)Y_t = (1 + \beta_{12}B^{12})(1+ \beta_1B)w_t$

$(1-0.07146577B^{12})(1-0.67848020B-0.26380676B^{2})(1-B^{12})(1-B)Y_t = (1-0.79843192B^{12})(1-0.0777806359B)w_t$

Regression Notation:
$Y_t = 1.678Y_{t-1} - 0.414Y_{t-2} - 0.264Y_{t-3} + 1.0712Y_{t-12} + 1.798Y_{t-13} - 0.0913Y_{t-14} - 0.-.245Y_{t-15} - 0.0231Y_{t-25}\ 0.0674Y_{t-26} + 0.019Y_{t-27} - 0.0778\, w_{t-1} -0.0798w_{t-12} + 0.621w_{t-13} + w_t$

To forecast, I refit the model with the arima function which will forecast the untransformed target variable of unemployment rate. R will forecast the values of the series and return the mean to them but it does not directly give us the actual values of the series as predictions. I also make a time series of the last 12 observations I kept to compare with the forecasts (the test data). With the two time series objects, I can predict.

Figure 12 shows the enitre time series and the forecasted value for the out of sample 12 months, and the confidence interval for the true future value of the series. The actual data is inside the interval and the forecasted and actual data follow the same general trend. Table 7 contains the values of the forecasts for Y_t 12 steps ahead using ARIMA modeling.

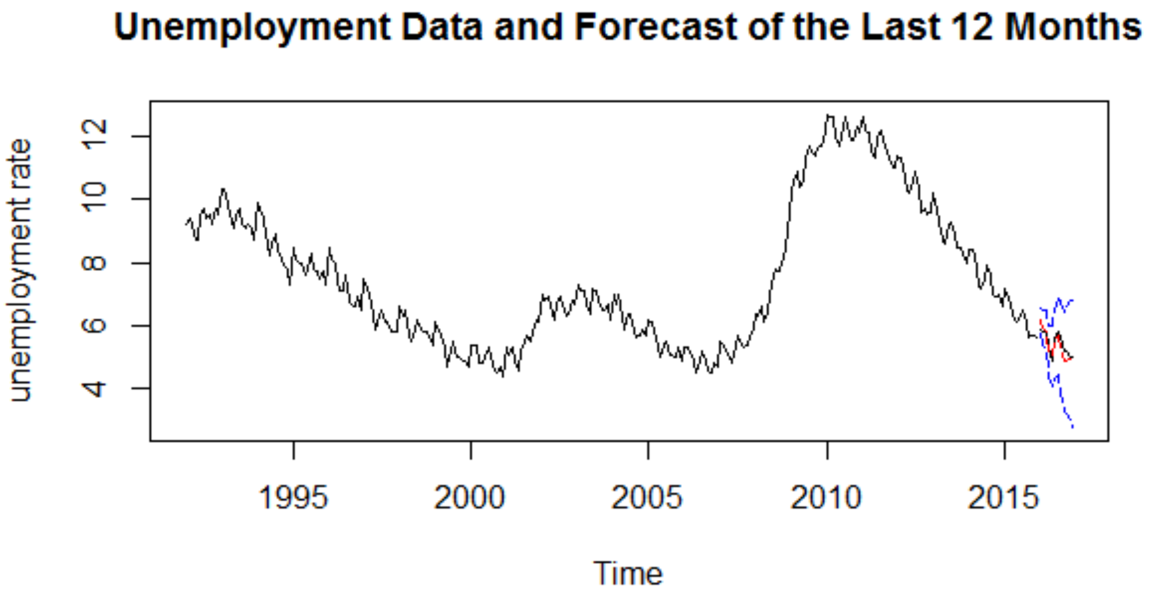


Figure 12: Forecast (in red) and confidence interval for unemp, the actual value of the time series in the future (in blue). The actual data is in black.

V. Vector Autoregression

I applied transformations to see if those would improve the variance of the models. However, I chose not to transform any of the variables because applying the transformations didn't change seem to change the trends. Then, I differenced the data to make each of the variables mean stationary as they were non-stationary prior to it. In order to make all three of the variables mean stationary, I first regularly and then seasonally differenced them. Figures 8, 13, and 14 show that regular differencing for each of the variables was not enough. There were still trends in the data and many significant lags. Y_t, X_t, and Z_t looked to be stationary after regular and seasonal differencing as there were less autocorrelation and a lower number of significant lags. Figures 15, 16, and 17 are all very close to white noise models and have a significant lag at 12.

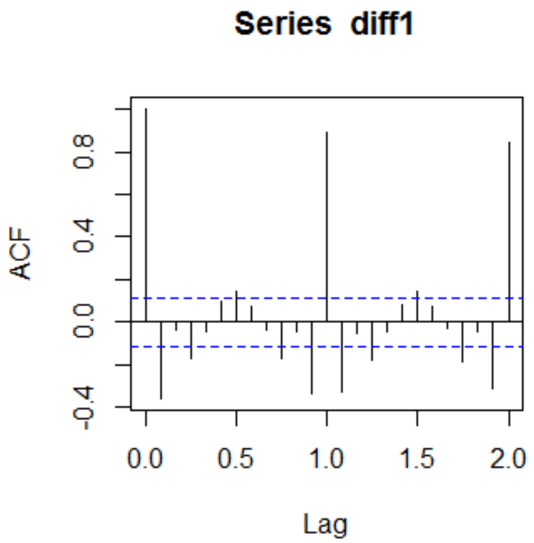


Figure 13: ACF of Regular difference (1-B)Y_t of goods

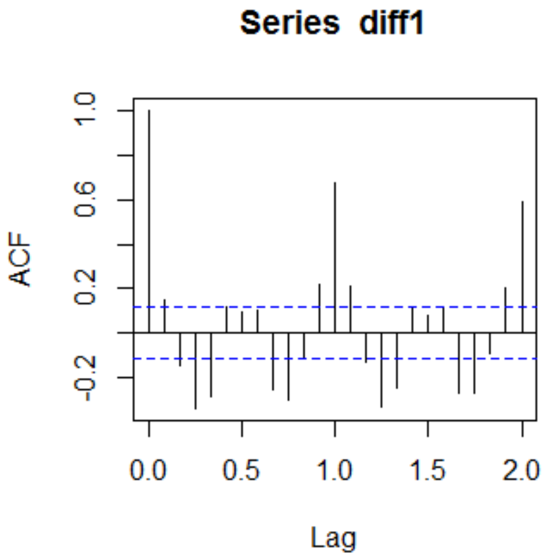


Figure 14: ACF of Regular difference (1-B)Y_t of edu

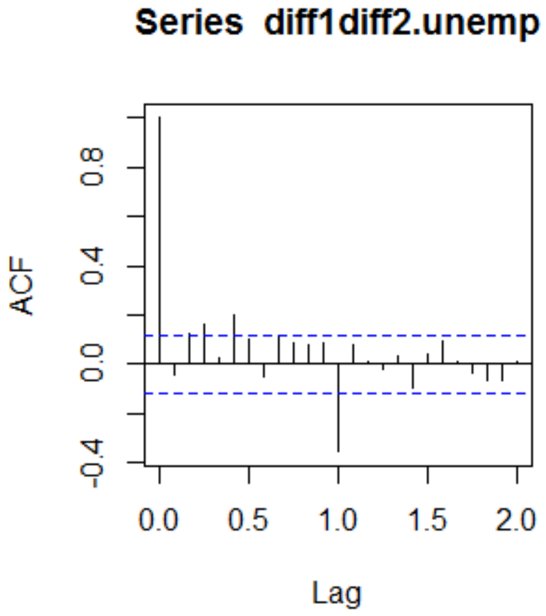


Figure 15: ACF of $Y_t^*=(1-B^{12})(1-B)Y_t$

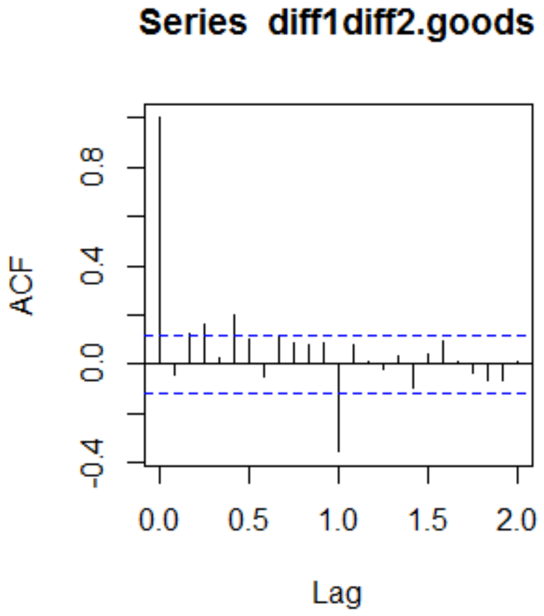


Figure 16: ACF of $X_t^*=(1-B^{12})(1-B)X_t$

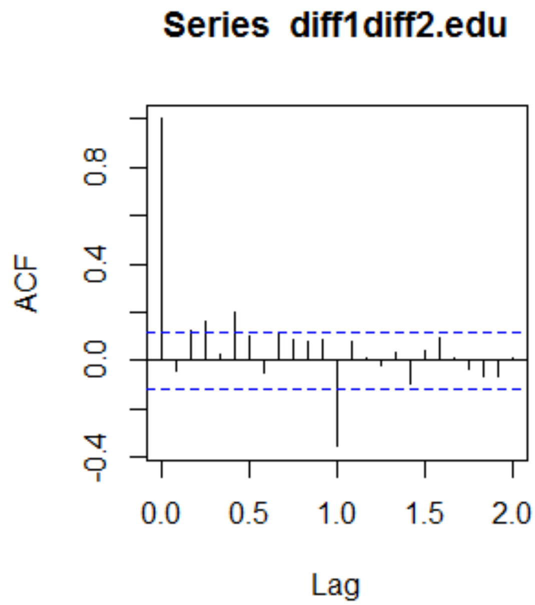


Figure 17: ACF of $Z_t^*=(1-B^{12})(1-B)Z_t$

Variable	Polynomial Notation
Unemp (Y_t)	Y_t *= $(1-B^{12})(1-B)Y_t$
Goods (X_t)	$X_t^* = (1-B^{12})(1-B)X_t$
Edu (Z_t)	$Z_t^* = (1-B^{12})(1-B)Z_t$

Table 3: Stationary Variables in Polynomial Notation

The star next to each of the variables in table 3 indicates that variable has been differenced.

Section V.1 Cross Correlation

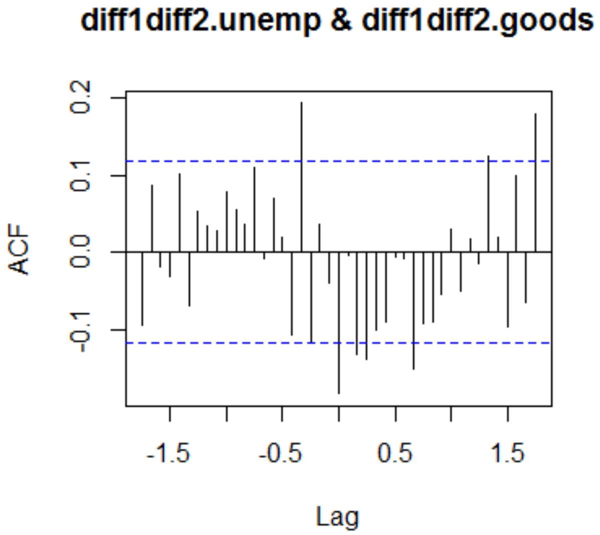


Figure 18: Cross Correlation of (Y_t^*, X_t^*)

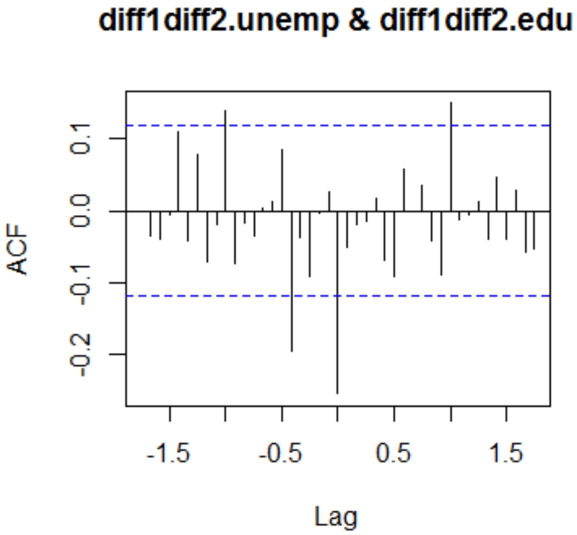


Figure 19: Cross Correlation of (Y_t^*, Z_t^*)

Figures 18 and 19 showcases the cross correlation of the differenced target variable with the two other differenced variables.

Table 4: Summary of Cross-Correlation Analysis of Stationary Time Series

ccf	Description
unemp with goods $(Y_{t,}^{*} X_t^{*})$	Figure 18 indicates that there are more significant spikes at negative lags which indicates that goods depends on past values of unemp. This indicates that unemp leads goods. Unemp affects goods.
unemp with edu (Y_t^{*}, Z_t^{*})	Figure 19 indicates that there are more significant spikes at positive lags which indicates that unemp depends on edu. This indicates that edu leads unemp. Edu affects unemp.

Section V.2 Unit Root Tests and Cointegration tests

This section focuses on the unit root and cointegration tests as they determine whether the series have a common stochastic trend prior to performing vector autoregression. Tables 5 and 6 give the conclusions from these tests.

Table 5: Conclusions of Unit Root Test

Non-Differenced Variables	Unit Root Test
unemp (Y_t)	The p-value is 0.8106 indicating that there is a unit root. We fail to reject the H_0 that unit root is 1 indicating that the time series is non-stationary.
goods (X_t)	The p-value is 0.6554 indicating that there is a unit root. We fail to reject the H_0 that unit root is 1 indicating that the time series is non-stationary.
edu (Z_t)	The p-value is 0.06464 indicating that there is a unit root. We fail to reject the H_0 that unit root is 1 indicating that the time series is non-stationary.

Table 6: Conclusions of Cointegration Test

Non-Differenced Variables	Cointegration Test
(unemp, goods) (Y_t, X_t)	The p-value is 0.01 which is significant. We can reject the H_0 that Y_t and X_t series are cointegrated
(unemp, edu) (Y_t, Z_t)	The p-value is 0.01 which is significant. We can reject the H_0 that Y_t and Z_t series are cointegrated

From the unit root tests, we can see that the series seem nonstationary. However, conducting the cointegration test shows that the series are cointegrated with the target variable. Individually, the variables are not stationary but they share a a stochastic trend and are stationary when together. The unit root, the results of which can be seen in Table 5, all produced non-stationary time series. As a result, I tried differencing the data. Using the results from the differenced data, I obtained the results for the contegration test. However, to avoid overfitting, I fit a VAR model directly to the variables without any differencing.

Section V.3 Fitting the VAR

Using the AR function to find the best VAR model, I got the best model to be VAR(27). This, however, is a very large VAR model and would be difficult to work with. I tried working with smaller order models of 1, 2, 3, and 4. However, each one of those models still presented models with a significant number of lags. I decided to use a VAR(15) model to fit my variables because this way the model is made smaller and the seasonality within the data is taken care of, while still having the ACF of the residuals of each variable close to white noise. The ACF of the residuals for each variable at VAR(15) indicates stationarity and can be seen in Figures 20, 21, and 22.

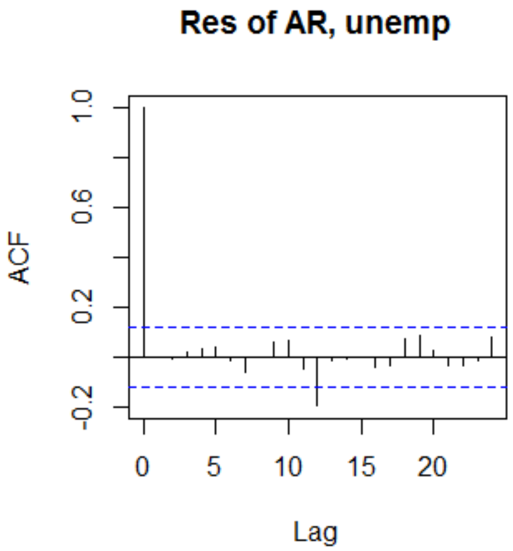


Figure 20: ACF of Residuals of Y_t of VAR(15)

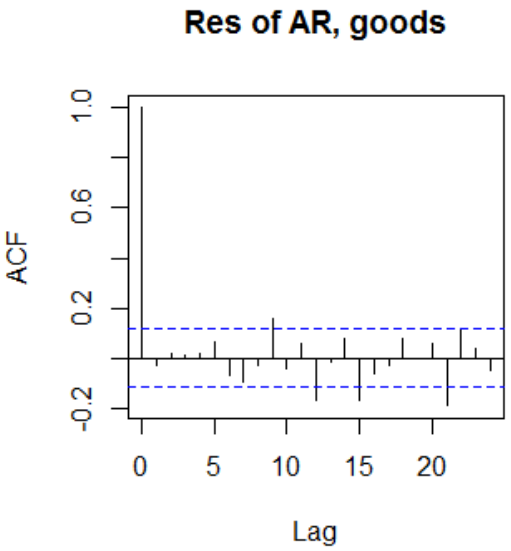


Figure 21: ACF of Residuals of X_t of VAR(15)

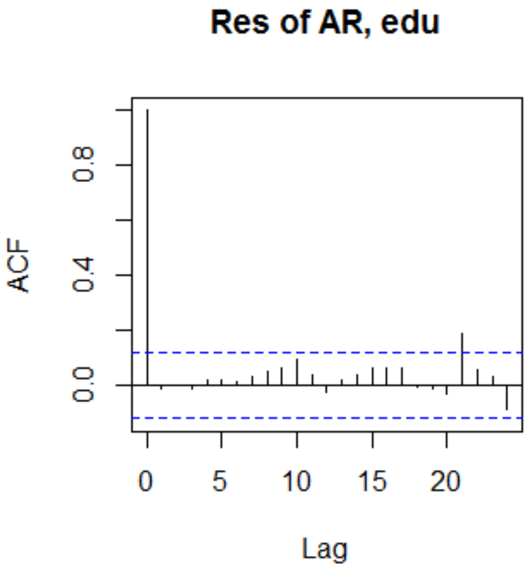


Figure 22: ACF of Residuals of Z_t VAR(15)

In Figures 20 and 21, there seems to be a faint seasonality present while Figure 22 doesn't seem to have any seasonality present. The ACF plots of unemp and edu, figures 20 and 22 look to be white noise. However, they are not completely white noise as they seem to have some significant lags but the lags are further out. Figure 20 has a significant lag present at 12 and Figure 22 has a significant lag present at lag 21. Figure 21 isn't exactly white noise as it has some significant lags towards the beginning at lags 9 and 12 and later at lags 15 and 21. However, it is close to white noise. However, this may be due to the fact that I fit the model at a lower order of 15 instead of the original model of 27.

In Figure 23, the cross correlation of the residuals of (Y_t, X_t) showcases a significant spike at 0 indicating that unemp leads goods which is consistent to the results gained from Figure 18. In Figure 24, the cross correlation of the residuals of (X_t, Y_t) showcases a significant spike at 0 indicating that unemp leads edu. The plots in Figures 23 and 24 showcase, for the most part, white noise which indicates that the variables are not highly dependent on each other. Figure 23 has 3 significant lags and Figure 24 has one significant lag. The significant lags present at lag 0 are due to seasonality that is still persistent in the data.

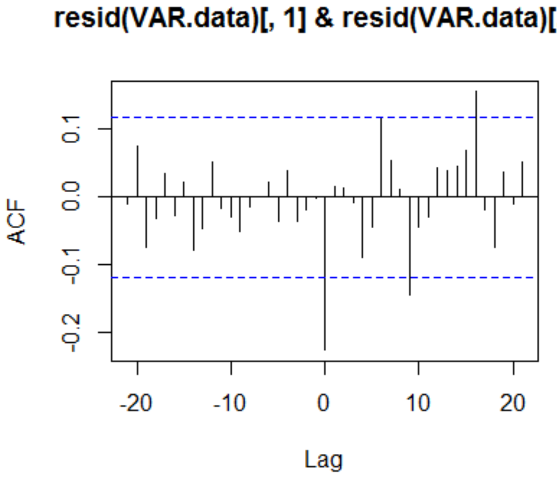


Figure 23: Cross Correlation of Residuals of (Y_t, X_t)

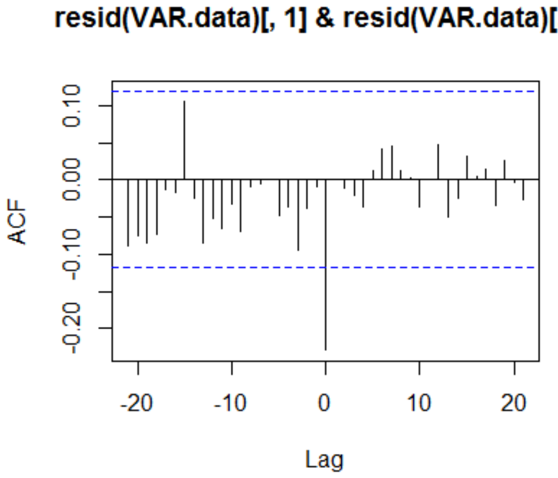


Figure 24: Cross Correlation Residuals of (Y_t, Z_t)

Fitted Model for AR(15):

$Y_t = 0.503 + 0.77y_{t-1}{}^{**} - 0.00629x_{t-1} + 0.000495z_{t-1} + 0.168y_{t-2} - 0.077x_{t-2} + 0.0632z_{t-2} + 0.066y_{t-3} – 0.0041x_{t-3} + 0.0122z_{t-3} - 0.0372y_{t-4} + 0.0067x_{t-4} + 0.041z_{t-4} + 0.1184y_{t-5} + 0.0146x_{t-5}{}^{*} – 0.138z_{t-5}{}^{*} + 0.00126y_{t-6} – 0.0178x_{t-6}{}^{*} – 0.0534z_{t-6} – 0.0988y_{t-7} + 0.00598x_{t-7} + 0.0377z_{t-7} + 0.076y_{t-8} - 0.0022x_{t-8} + 0.0688z_{t-8} – 0.056y_{t-9} + 0.0056x_{t-9} + 0.0405z_{t-9} – 0.0573y_{t-10} – 0.0045x_{t-10} - 0.0154z_{t-10} – 0.064y_{t-11} – 0.004x_{t-11} – 0.0669z_{t-11} + 0.502y_{t-12}{}^{*} + 0.002x_{t-12} + 0.0398z_{t-12} - 0.275y_{t-13}{}^{*} + 0.005x_{t-13} + 0.0534z_{t-13} - 0.1248y_{t-14} + 0.006x_{t-14} - 0.04025z_{t-14} -0.014y_{t-15} - 0.00015x_{t-15} - 0.046z_{t-15}$

$X_t = 18.2 - 2.04y_{t-1}{}^{*} + 0.692x_{t-1}{}^{**} + 0.337z_{t-1} + 1.77y_{t-2} + 0.237x_{t-2}{}^{*} - 1.17z_{t-2} - 0.7803y_{t-3} – 0.0158x_{t-3} + 0.7134z_{t-3} - 0.955y_{t-4} + 0.0426x_{t-4} - 0.791z_{t-4} - 1.18y_{t-5} + 0.0437x_{t-5} + 1.612z_{t-5}{}^{*} – 1.34y_{t-6} + 0.034x_{t-6} – 1.83z_{t-6}{}^{*} + 3.332y_{t-7}{}^{*} - 0.1205x_{t-7}{}^{*} + 0.5419z_{t-7} – 1.107y_{t-8} + 0.0154x_{t-8} - 1.379z_{t-8} + 0.3087y_{t-9} – 0.0413x_{t-9} + 1.07z_{t-9} - 0.635y_{t-10} – 0.00007x_{t-10} - 0.977z_{t-10} - 0.0616y_{t-11} – 1.089x_{t-11}{}^{*} + 2.094z_{t-11}{}^{*} - 0.971y_{t-12} + 0.712x_{t-12}{}^{**} - 1.49z_{t-12} - 0.215y_{t-13} - 0.496x_{t-13}{}^{**} + 9.326z_{t-13} + 0.814y_{t-14} - 0.156x_{t-14}{}^{*} - 1.024z_{t-14} - 3.747y_{t-15} + 0.1096x_{t-15} + 1.217z_{t-15}$

Z_t = -0.482 - 0.067y_{t-1} + 0.011x_{t-1} + 0.708z_{t-1}** + 0.186y_{t-2} + 0.0035x_{t-2} + 0.028z_{t-2} + 0.017y_{t-3} – 0.0091x_{t-3} + 0.106z_{t-3} - 0.259y_{t-4}* - 0.009x_{t-4} - 0.144z_{t-4} - 0.354y_{t-5}* - 0.014x_{t-5}* + 0.0785z_{t-5} + 0.302y_{t-6}* + 0.0204x_{t-6}** – 0.051z_{t-6} + 0.02y_{t-7} - 0.0058x_{t-7} + 0.0264z_{t-7} + 0.178y_{t-8} + 0.008x_{t-8} + 0.062z_{t-8} – 0.0531 6y_{t-9} + 0.0045x_{t-9} +0.087z_{t-9} + 0.081y_{t-10} + 0.000014x_{t-10} - 0.0476z_{t-10} – 0.0613y_{t-11} + 0.0015x_{t-11} + 0.073z_{t-11} + 0.0324y_{t-12} + 0.0046x_{t-12}* + 0.1831z_{t-12} - 0.021y_{t-13} - 0.0057x_{t-13} - 0.016z_{t-13} + 0.0479y_{t-14} - 0.0118x_{t-14} - 0.0615z_{t-14} - 0.045y_{t-15} + 0.00189x_{t-15} - 0.029z_{t-1}

One star next to the variables indicates that they are significant at 0.05 level and two stars indicates that they are significant at the 0.001 level.

Based on the results from the fitted model, we would say that Y_t gets affected by itself at time t-1, t-13 (the corresponding lags are 1,13), gets affected by X_t at time t-5 and t-6 (I the corresponding lags are 5,6), and by Z_t at time t-5 (the corresponding lags are 5).

X_t gets affected by itself at t-1, t-2, t-7, t-11, t-12, t-13 and t-14 (lags 1, 2, 7, 11, 12, 13, 14). X_t gets affected by Y_t at t-1 and t-7 (lags 1 and 7). X_t gets affected by Z_t at t-5, t-6, t-11 (lags 5, 6, 11).

Z_t gets affected by itself at t-1 (lags 1). Z_t is affected by Y_t at t-4, t-5, and t-6 (lags 2,4,5,6). Z_t is affected by X_t at t-5, t-6, and t-12 (lags 5, 6, 12).

Section V.4 Impulse Response Functions

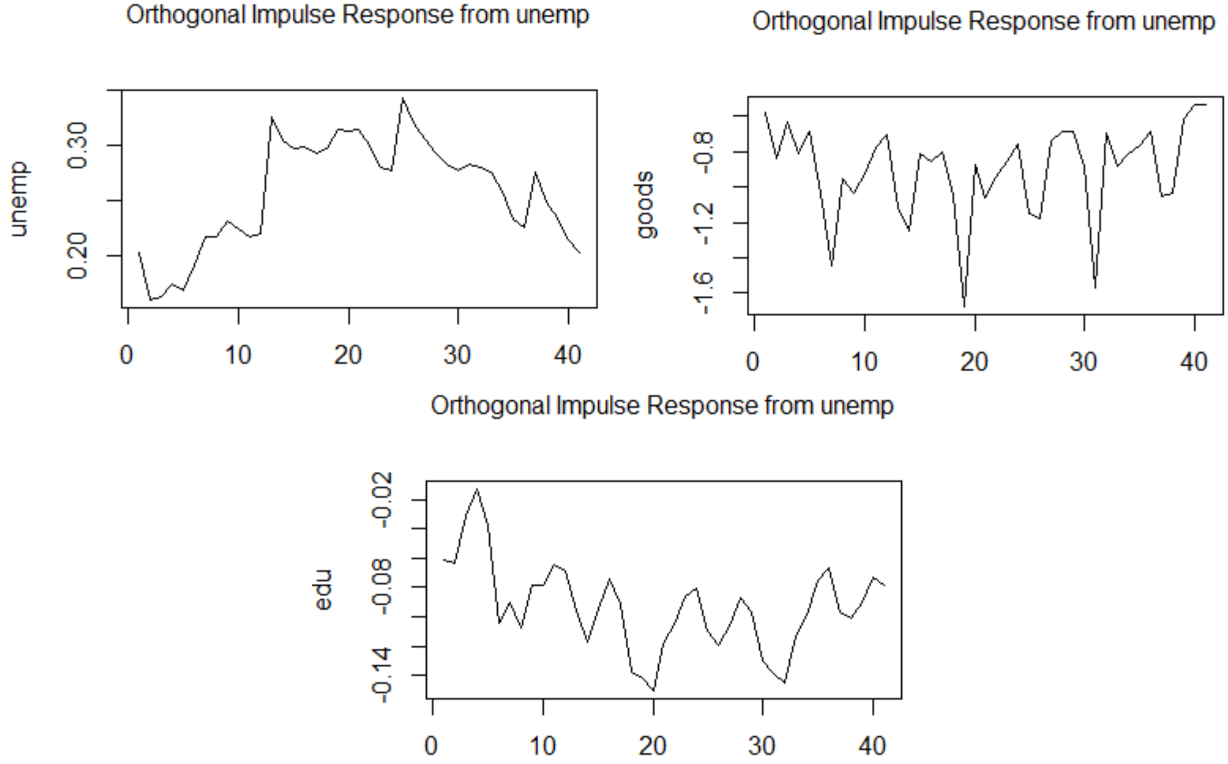


Figure 25: Impulse Shock on Y_t and the Response of Y_t (top left), X_t (top right), and Z_t (bottom)

In Figure 25, we can that the shock on Y_t to itself and the other two variables. When there is a shock on Y_t it initially decreases and then follows an increasing and then decreasing pattern as it looks like it is coming down and stabilizing at 0. The shock on Y_t causes an initial decease in goods which then increases and decreases while the model overall is increasing to 0. The effect of the shock on Z_t is met with an initial increase and then a decrease that persists. Towards the end, the model looks to be increasing towards 0. All of them take a while to stabilize. Even after 40 periods, they have still not reached 0. However, the response of $X_{t,}$ is approaching very close to 0.

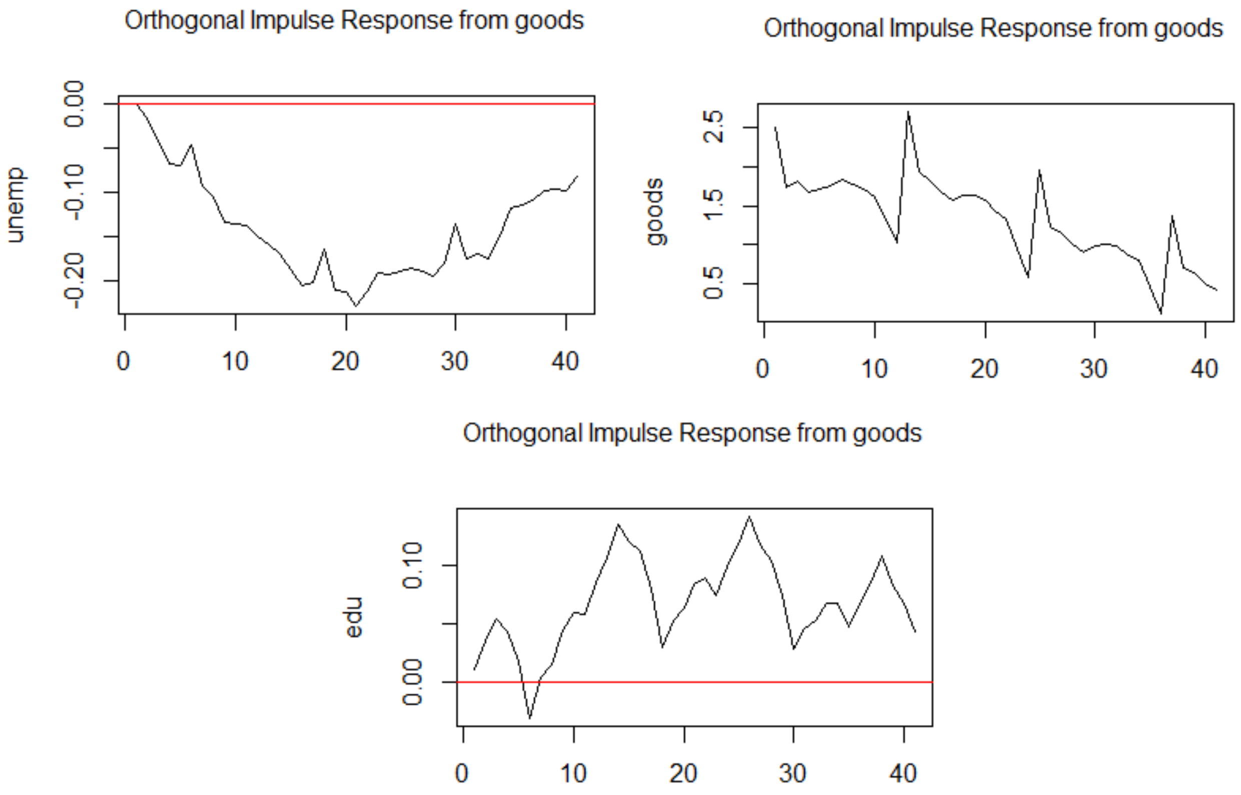


Figure 26: I Impulse Shock on X_t and the Response of Y_t (top left), X_t (top right), and Z_t (bottom)

In Figure 26, we can that the shock on X_t to itself and the other two variables. When there is a shock on X_t, it decreases Y_t until period 20 where it then begins to increase and make its way back to 0. The shock on X_t causes an initial decease to itself which then follows a fairly regular fluctuating patter of increasing and decreasing as it is over all decreasing. The effect of the shock on Z_t is met with an increase and then a decrease that goes from a positive value to negative value and then back to positive value where it persists. Throughout the time period it constantly changes from being positive to negative while always being fairly close to the 0 line. All of them take a while to stabilize. Even after 40 periods, they still have not reached 0. However, the response on Y_t and Z_t look like they are making their way to stabilize at 0.

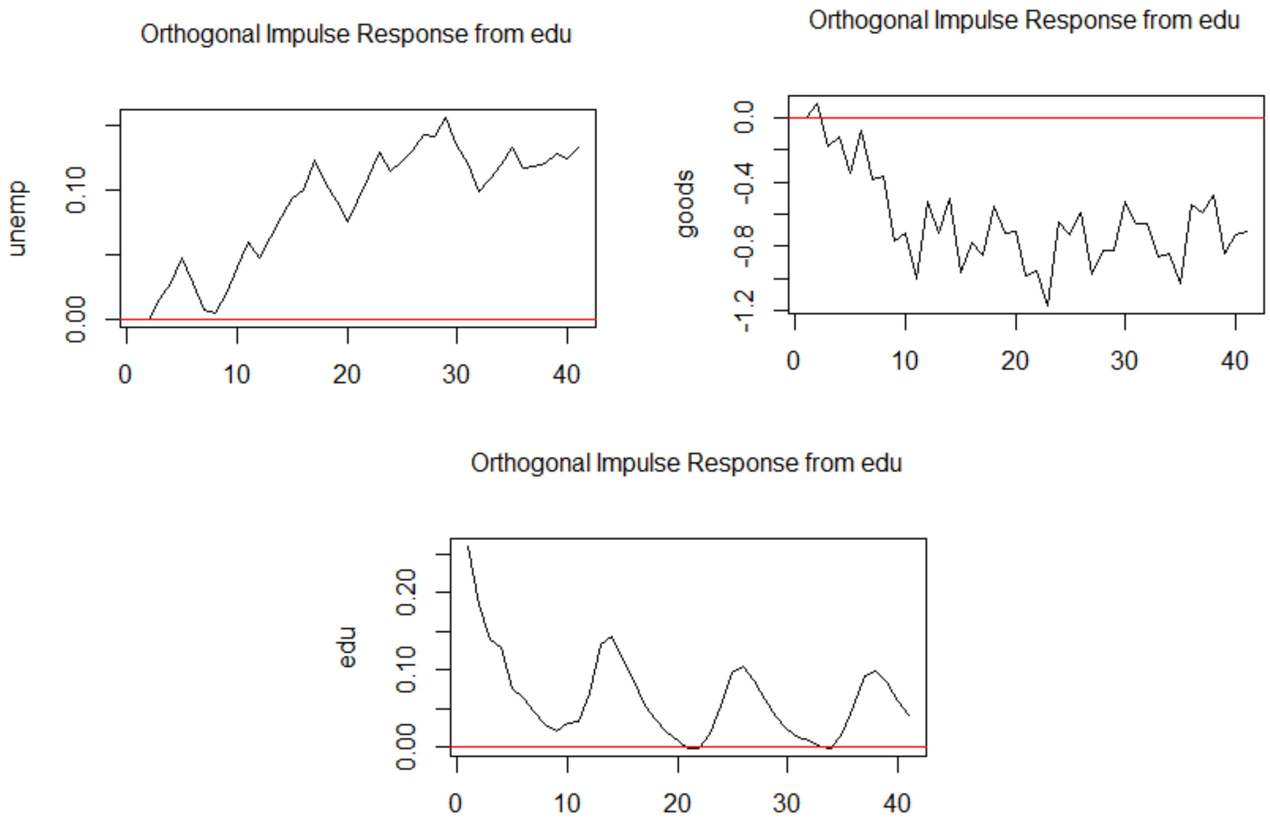


Figure 27: Impulse Shock on Z_t and the Response of Y_t (top left), X_t (top right), and Z_t (bottom)

In Figure 27, we can that the shock on Z_t to itself and the other two variables. When there is a shock on Z_t, Y_t increases and then decreases throughout the time period while having an overall increasing trend. The shock on Z_t causes an initial increase in X_t which then decreases greatly until period 10 and then follows an increasing/decreasing pattern while overall looking like it is increasing. The effect of the shock on itself is met with a decrease and then a curved increase that is persistent throughout. At about periods 22 and 35, it crosses from being a positive value to a negative value for a brief moment and then is positive after. The model looks like it is going towards the 0 line over time. All of them take a while to stabilize. Even after 40 periods, they have not reached 0. However, the response on itself looks like it is making its way to stabilizing at 0.

Section V.5 Forecasts

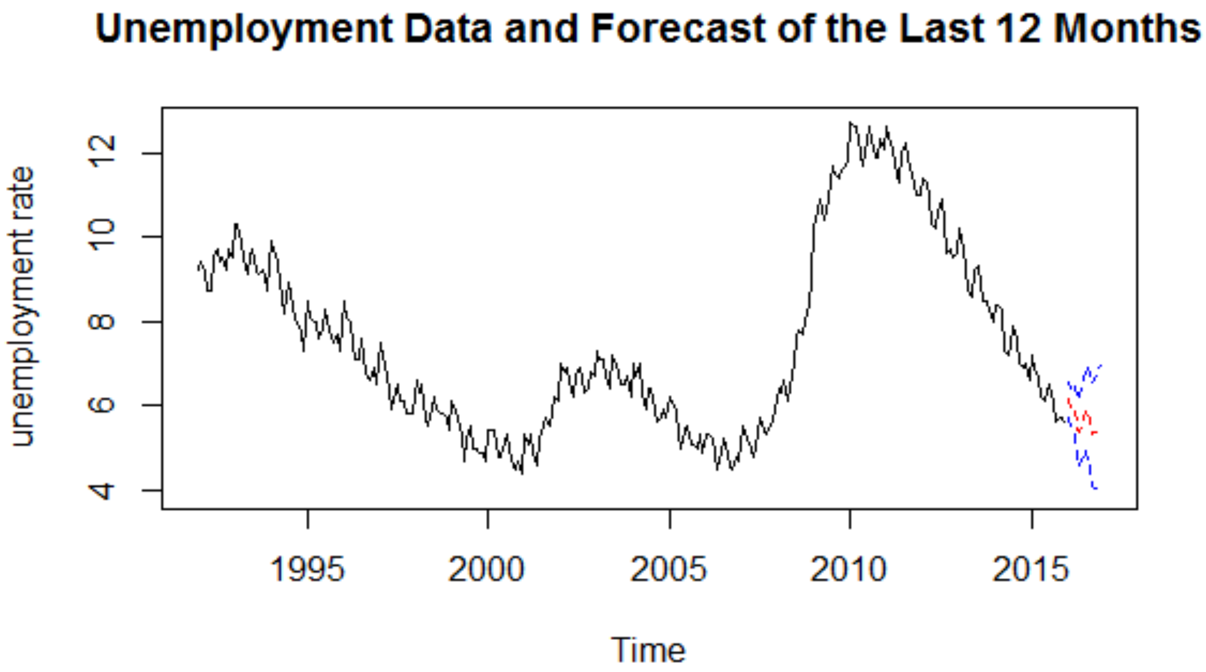


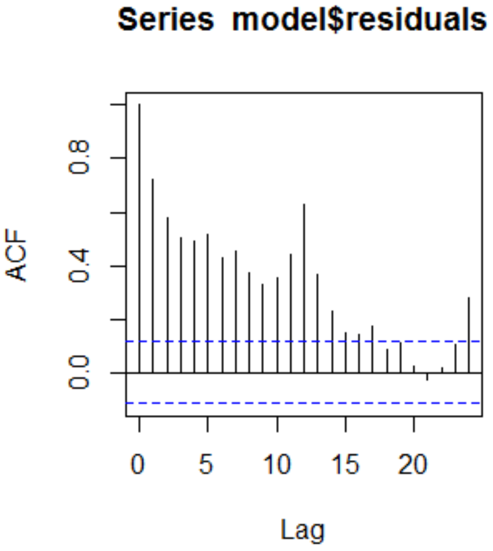
Figure 28: Forecast (in red) and confidence interval for unemp, the actual value of the time series in the future (in blue). The actual data is in black.

Figure 28 shows the entire time series, the forecasted value for the out of sample 12 months, and the confidence interval for the true future value of the series. The actual data is inside the interval and the forecasted and actual data follow the same general trend. Table 7 contains the values of the forecasts for Y_t 12 steps ahead using VARS modeling.

VI. Time Series Regression

VI.1 Fitting Linear Regression Model

Using linear regression, I use my untransformed target variable, unemp, as the dependent variable as it showcases variance stationarity. The two other variables, goods and edu, are used as the independent variables, both in their untransformed state.



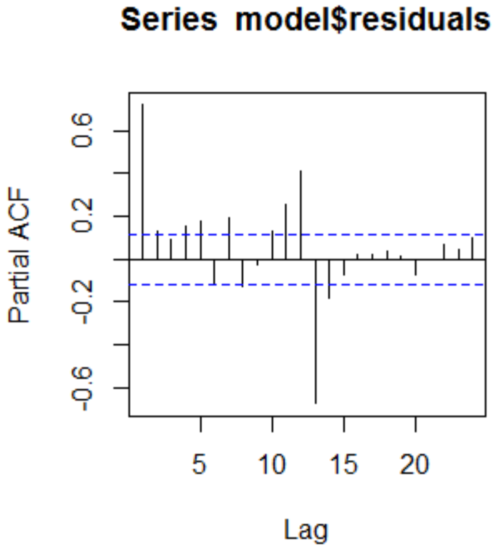


Figure 29: ACF of residuals of linear regression model

Figure 30: PACF of residuals of linear regression model

Figure 29 shows that the model is non-stationary as the there is a clear decreasing trend and many significant lags and takes time to die down. Using Figure 31, we can check if the residuals satisfy the other assumptions. Looking at the standardized residuals plot in Figure 31, we see that the residuals have no evidence of being larger as time increases. There are no long runs and they are moving from negative to positive values quickly so the assumption of constant variance is satisfied. The residuals are also between –3 and 3, which is expected if they are normal. The histogram in Figure 29 shows that all the residuals are for the most part symmetrically scattered around 0 with a normal shape. The qqplot in Figure 31 suggests that the residuals are fairly normal, there are a couple of points at the tails that don't follow the normal trend. However, this is not strong enough evidence to say that the residuals are not normal.

Because figure 29 suggests that there is a non-stationary autocorrelation structure in the residuals, we look at the PACF to identify an ARMA model. From Figure 29, we see that the ACF takes a while to cut off, and from figure 30, we see that there are about 13 significant lags before the PACF cuts off suggesting that AR(13) would be an appropriate fit to the model.

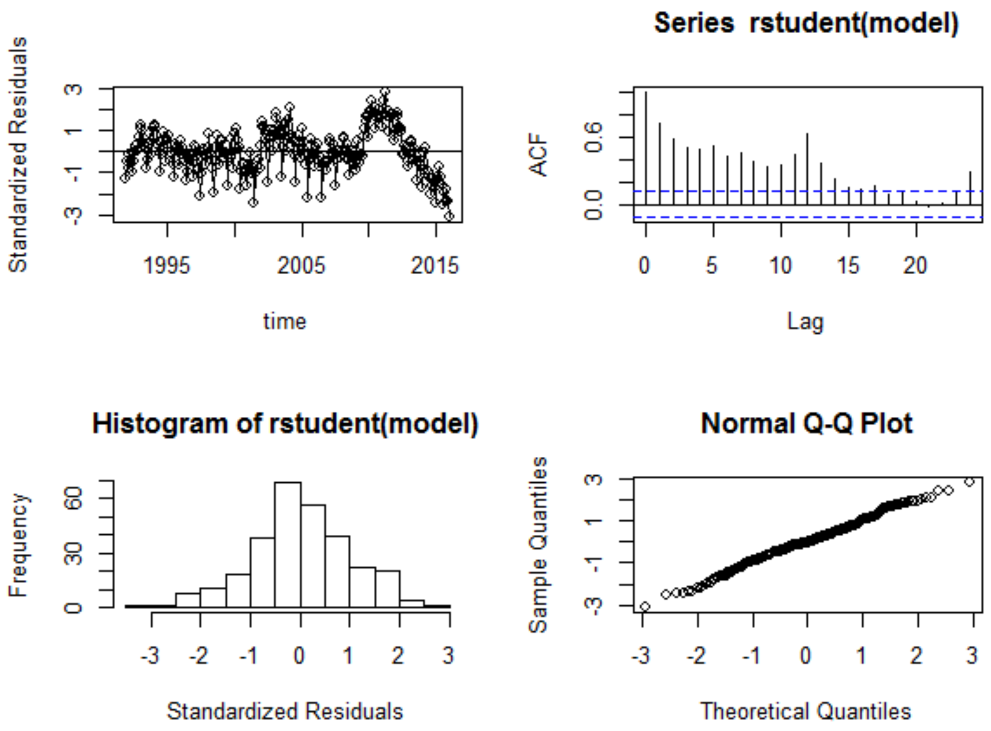


Figure 31: Plots to check normality. Standardized residuals (top left), ACF of linear model (top right), Histogram (bottom left), QQPlot (bottom right)

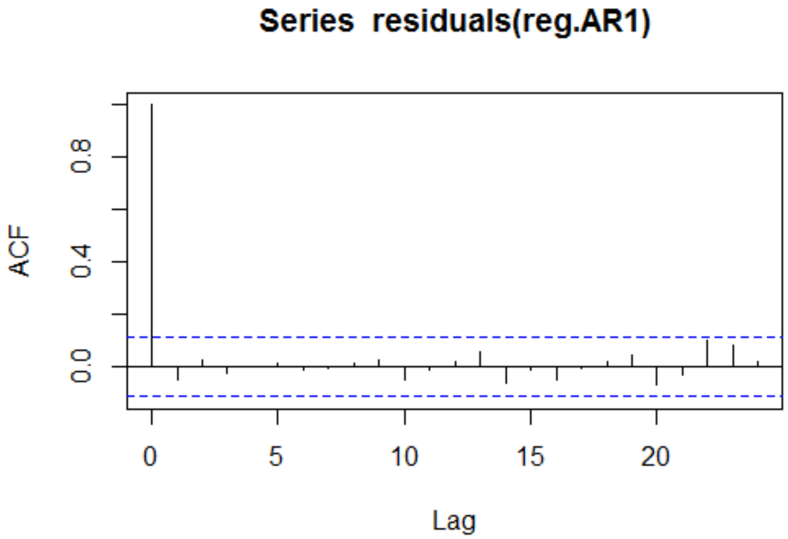
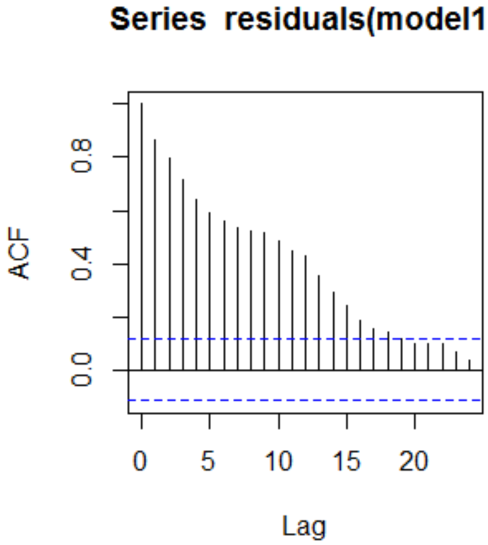


Figure 32: ACF of Residuals of ARIMA(13,0,0)

Because the residuals are correlated in Figure 29, we fit an ARMA model and check if we get white noise. We fit an AR(13) model as indicated from the ACF and PACF in Figures 29 and 30, and ACF of the residuals of the model, Figure 32, indicates white noise. Because such a high AR model was needed to be fit to get a white noise model, the better option would be to take care of the autocorrelated residuals using dummy variables for months and polynomials for trends.

VI.2 Fitting Linear Model with Dummy Variables



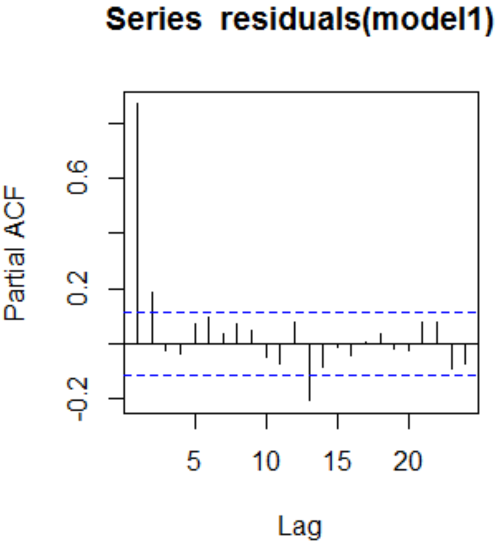


Figure 33: ACF of residuals of linear regression model with dummies and trends

Figure 34: PACF of linear regression model with dummies and trends

From Figure 33, we see that there is smooth decay in the ACF of the residuals indicating no stationarity and autocorrelation with the model. Because there is no stationarity, I look at the PACF, figure 34, and determine that an AR(13) will give a stationary model. Figure 35, shows that the residuals of an AR(13) gives white noise and indicates stationarity. We can now fit a gls model to the linear model with dummies and trends.

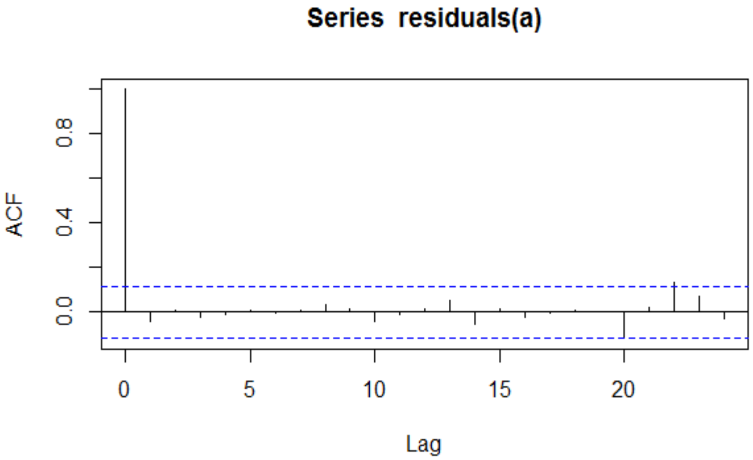


Figure 35: ACF of residuals of AR (13) of linear regression with dummies and trends

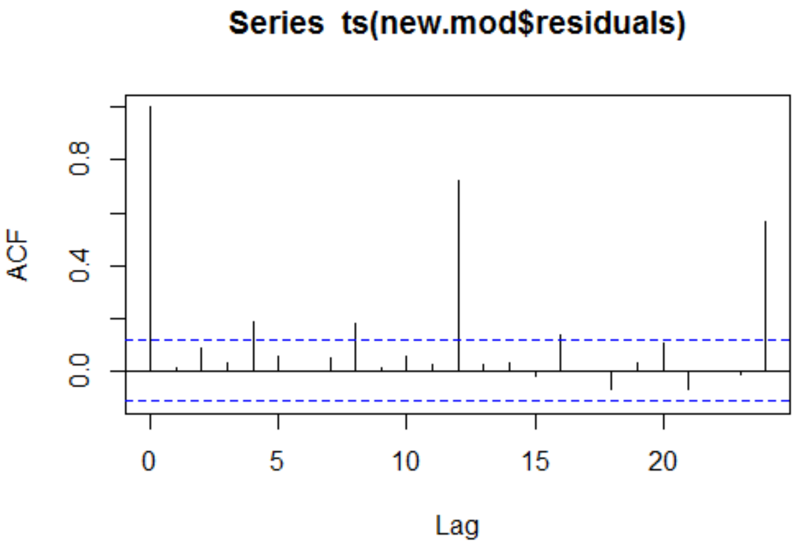


Figure 36: ACF of Residuals of GLS model

In order to get the residuals of the gls model, we had to transform the variables the way gls transforms the variables. After doing so, the residuals of the gls model, given in Figure 36, indicate approximately white noise except for the few significant lags at r_k 4, 8, 12, and 24. The significant lag at 12 may be due to a seasonality still present. However, adding more trends to the linear model did not change the residual plot by much.

<u>Regression Equation:</u>

Y_t = -29.183 – 0.007X_t – 0.134Z_t -0.081factor(months)$_2$ – 0.119factor(months)$_3$ –
(Std Error)
(161.841) (.004) (0.038) (0.061) (0.064)
(P value)
(0.857) (0.121) (0.0006) (0.189) (0.064)

0.636factor(months)$_4$ -0.873factor(months)$_5$ -0.504factor(months)$_6$ -0.366factor(months)$_7$ –
(Std Error)
(0.063) (0.069) (0.071) (0.076)
(P value)
(0.000) (0.000) (0.000) (0.000)

0.607factor(months)$_8$ – 0.727factor(months)$_9$ -0.683factor(months)$_{10}$ – 0.564factor(months)$_{11}$ –
(Std Error)
(0.075) (0.069) (0.068) (0.057)
(P value)
(0.000) (0.000) (0.000) (0.000)

0.741factor(months)$_{12}$ +0.024times
(Std Error)
(0.058) (0.081)
(P value)
(0.000) (0.767)

Section VI. Forecasts

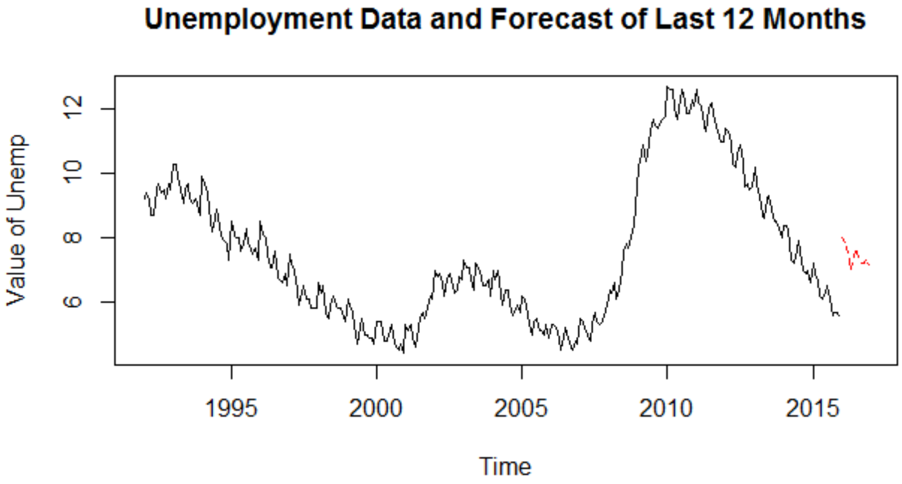


Figure 35: Regression Forecast (in red) and the actual data is in black.

Figure 35 shows the entire time series and the forecasted value for the out of sample 12 months. The forecast of the next 12 months does not look that great. The start of the forecast is very high compared to where the 2015 data ends and doesn't seem to be following the overall general trend of the data very well. Table 7 contains the values of the forecasts for Y_t 12 steps ahead using Regression modeling and we can see that predicted unemployment values for all the months are greater than the actual values thus giving us a high RMSE and producing the worst of the four forecasting models.

VII. Exponential Smoothing

VII.1 Type of Exponential Smoothing

In regards to my data, the seasonal exponential smoothing fits it the best. The data for unemp, the target variable, has a trend and a seasonal component, and the exponential smoothing with the additive model provides the best fit. This is consistent with Section III of this paper that discusses additive and multiplicative decomposition where the decomposition of the additive term is the better model indicated by the ACF of the residuals in Figure 5.

Before settling on this model as the best one, I fit a Holt Winters model without trend and seasonal component, with a trend and no seasonal, and one with the seasonal model as multiplicative. Running plots, on all of these models and calculating the RMSE for each, I found that the additive model was the best. The fitted model in Figure 36 shows that the fitted lines in red follow the actual data very closely while not overfitting or leading the actual data's trend. Further, the RMSE of the additive model was lowest at 0.199.

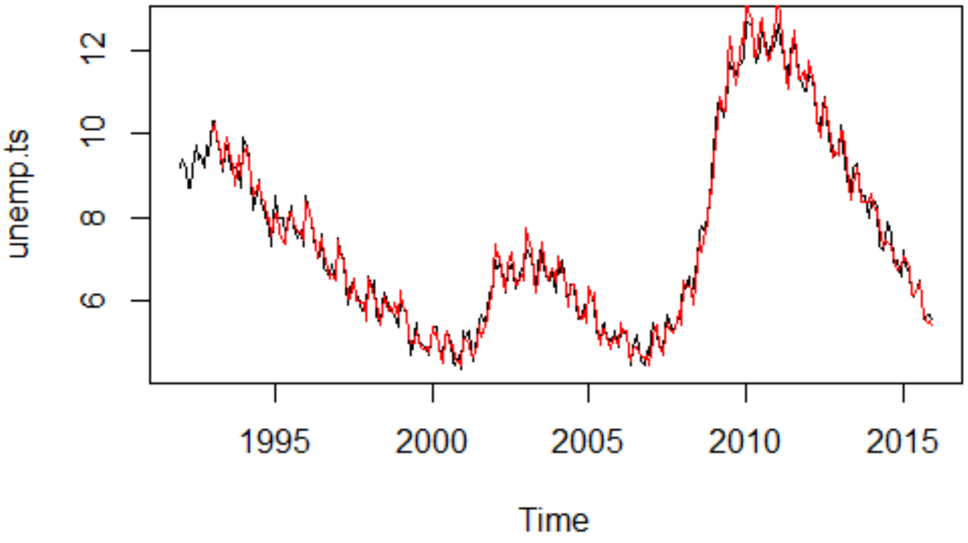


Figure 36: Holt winters Additive model, fitted data (in red) and actual data (in black)

The optimal adjustment parameters for alpha, beta, and gamma using the Holt Winters additive method are 0.6335012, 0.1544274, and 0.7460405, respectively.

Exponential Smoothing Equation:

$\hat{Y}_{t+h} = a_t + h *b_t + s_{t+1+(h-1)12}$

$a_t = 0.634(Y_t - s_{t-p}) + (1-0.634) (a_{t-1} + b_{t-1})$
$b_t = 0.154(a_t - a_{t-1}) + (1-0.154) b_{t-1}$
$s_t = 0.746(Y_t - a_t) + (1-0.746) s_{t-p}$

VII.2 Forecasts

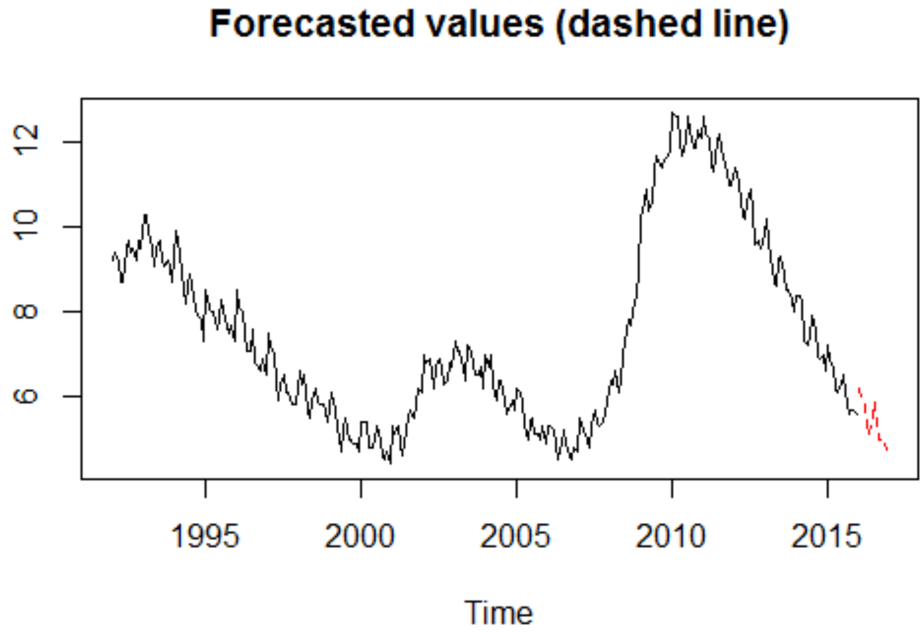


Figure 37: Exponential Smoothing Forecast is in red and the actual data is in black

Figure 37 shows the entire time series and the forecasted value for the out of sample 12 months. The forecasted data is in red and follows the general pattern of the data. The end of 2015, where the actual data leaves off is decreasing and the forecasted data in red follows that trend of decreasing. Table 7 contains the values of the forecasts for Y_t 12 steps ahead using Exponential Smoothing and we can see that the predicted values close to the values in the raw sample data thus providing the lowest RMSE and the best forecast of the four models.

VIII. Results

In this paper, I used four different methods, ARIMA, VARS, Regression, and Exponential Smoothing to forecast the unemployment rate in California and determine which model gave predicted values closest to the actual values. To do so, the univariate techniques of ARIMA and Exponential Smoothing worked with only the target variable of unemployment rate while the multivariate methods of VARS and Regression worked with all three variables: unemp, goods, and edu.

To conduct the ARIMA model, we had to first make sure that the target variable was both mean and variance stationary. In order to make it variance stationary, pre-transformations need to be done if the data was not stationary. However, my target variable was already, for the most part variance stationary and applying log and quartic transformations to it did not change the variance much. As such, I chose to work with the untransformed variable. In order, to make it mean stationary, I had to regularly and seasonally difference the data. Looking at the ACF and the PACF of the regularly and seasonally differenced data, I found that an $ARIMA(2,1,1)(1,1,1)_{12}$ provided the best model as it had a low AIC value, Ljung Box test results that indicated white noise, and an ACF of the residuals of the model that further indicated white noise. With the forecasting, I found that ARIMA had an RMSE of 0.207. The advantage with arima is that it can fit an approximation to any time series, but the disadvantage is that it may not always be a good approximation.

The next univariate method conducted was Exponential Smoothing. With exponential smoothing, no pre-transformations and differencing are required so I used unemp in its original form. The additive seasonal exponential smoothing method was the best exponential smoothing method as the data has a seasonal trend to it and the additive decomposition is the better fit for the data. Further, it provided the lowest RMSE and best fitted model of all the models tried indicating that the additive seasonal is the best. An advantage to exponential smoothing which allows it to be such an accurate predictor is that it takes into account a fraction of the error that is made with the previous prediction.

The multivariate method of VARS required pre-transformations for variance stationarity and differencing for mean stationarity on all three of the variables. All three variables looked to be variance stationary and applying transformations did not change them. As such, the variables were used in their untransformed, natural state. However, differencing for each variable, regularly and seasonally, was needed to achieve mean stationarity. Then, a VAR model of 15 was fitted to determine which variables lead in the movement of two stationary time series. I found that unemp leads goods and edu leads unemp.

The next multivariate method of Regression required pre-transforming if necessary but no differencing. Since all the variables were already variance stationary, they were not transformed and used in their natural form. The residuals in the regression model were autocorrelated and in order to take care of them, I used dummy variables and trends A disadvantage of regression is that information of the independent variable is needed and most of the time we don't have that. This is not a great model for forecasting.

Table 7 contains the forecasts of all methods mentioned above given with their monthly predictions and overall RMSE.

Table 7: Forecasting Data

Time Period (2016)	Raw Sample Data (Test Data)	ARIMA Forecast	VARS Forecast	Regression Forecast	Exponential Smoothing Forecast	Average of 4 Forecasts
January	5.9	6.164	6.137	7.991	6.191	6.621
February	5.8	5.958	5.918	7.858	5.998	6.433
March	5.8	5.803	5.918	7.751	5.822	6.324
April	5.4	5.181	5.463	7.279	5.185	5.777
May	4.9	5.041	5.370	7.012	5.109	5.633
June	5.6	5.413	5.708	7.465	5.497	6.021
July	5.8	5.692	5.929	7.695	5.858	6.294
August	5.6	5.339	5.694	7.427	5.501	5.990
September	5.2	4.888	5.339	7.236	4.971	5.609
October	5.2	4.902	5.365	7.224	4.963	5.614
November	5.0	4.920	5.463	7.378	4.905	5.667
December	5.0	4.789	5.443	7.183	4.739	5.539
RMSE		0.207	0.259	2.031	0.188	0.542

In the long run, Exponential Smoothing performed the best with an RMSE of 0.188. I was surprised as to how all the methods did fairly well and were fairly close in their RMSE except for Regression. The RMSE for regression was at 2.031, much higher than the other three methods. This caused the average of the four forecasts to have higher monthly predicted values and an increased RMSE which is not what I was expecting to happen with the average. I was anticipating the average to be the best model as it takes into account all the methods and in theory should provide the best model. However, that was clearly not the case.

Table 8: Forecasting the Short-Term Data

Time Period (2016)	Raw Sample Data (Test Data)	ARIMA Forecast	VARS Forecast	Regression Forecast	Exponential Smoothing Forecast	Average of 4 Forecasts
January	5.9	6.164	6.137	7.991	6.191	6.621
February	5.8	5.958	5.918	7.858	5.998	6.433
Short Term RMSE		0.218	0.187	2.075	0.249	0.678

In the short run, VARS performed the best with an RMSE of 0.187. It is interesting to note that in the long term, a univariate method provided the best result, but in the short term a multivariate model proved best. This may be attributed to the fact that the short term relies more on the other variables to make better predictions as less data is available whereas the long run relies heavily on the target variable.

For all methods except VARS, the short-term prediction increased the RMSE values indicating that for my data, the short-term forecast may not be a good predictor of unemp. This makes sense with what I believed to be true. Unemployment is affected more by the economy and the production of goods, but I also thought education level would play a higher role in the unemployment level as you would expect higher educated people to have more employment opportunities thus reducing the unemployment level. However, unemployment is based more off the economy and less off people's education level. The seasonal boxplot, Figure 4, exemplifies this as more people find employment during the Holiday seasons compared to the rest of the year due to seasonal employment. As such, goods and edu, only played a role when forecasting for short term when less information of the unemployment rate was provided. When there is more data on the unemp provided, the long run does better with just forecasting using that variable.

References:

[1] FRED: https://fred.stlouisfed.org/
[2] FRED: https://fred.stlouisfed.org/series/CAURN
[3] FRED: https://fred.stlouisfed.org/series/IPB51100N
[4] FRED: https://fred.stlouisfed.org/series/LNU02327662
[5] Cowpertwait, P.S.P., Metcalfe, A.V. (2009). Introductory Time Series with R. Springer-Verlag

Appendix 1:
R Script for Section III

```
data <- read.csv("C:/Users/sabap/Desktop/170 Project/PartI/Paya-104483616.csv")
summary(data)

#Mean of Variables
mean(unemp.ts)
mean(goods.ts)
mean(edu.ts)

#Standard Deviation of Variables
sd(unemp.ts)
sd(goods.ts)
sd(edu.ts)

#Turning Variables into Time Series
unemp.ts <- ts(data[,2], start = 1992, fr = 12)
goods.ts <- ts(data[,3], start = 1992, fr = 12)
edu.ts <- ts(data[,4], start = 1992, fr = 12)

#Time Series Plots
par(mfrow=c(3,1))
plot(unemp.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016", type="l")
plot(goods.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016",  type="l")
plot(edu.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016",  type="l")

#Additive Decomposition of the Data
unemp.add = decompose(unemp.ts, type = "add")
plot(unemp.add)

#Multiplicative Decomposition of the Data
unemp.mult = decompose(unemp.ts, type = "multi")
plot(unemp.mult)

#Seasonal Plot of the Data
boxplot(unemp.ts ~ cycle(unemp.ts), main = "Rate of Unemployment by Season", xlab="Month",
ylab="Percent")

#ACF of Additive Decomposition of the Data
par(mfrow=c(1,2))
acf(unemp.add$random, lag=50, main="ACF of Additive Decomposition", na.action = na.omit)

#ACF of Multiplicative Decomposition of the Data
acf(unemp.mult$random, lag=50, main="ACF of Multiplicative Decomposition", na.action = na.omit)
```

Appendix 2:
R Script for Section IV

```
#Working with n-12 data
data <- read.csv("C:/Users/sabap/Desktop/170 Project/PartI/Paya-104483616.csv")
newdata<-data[1:288,-1]

unemp1.ts <- ts(newdata[,1], start = 1992, fr = 12)
```

```
#Differencing the Data
diff1=diff(unemp1.ts, lag=1, differences = 1)
acf(diff1)

diff12=diff(unemp1.ts, lag=12, difference=1)
acf(diff12)

diff1diff2=diff(diff1, lag=12, differences = 1)
acf(diff1diff2)

#Partial ACF of Y**
pacf(diff1diff2)

#Identifying an ARIMA model
m1<-arima(unemp1.ts, order=c(2,1,1), seas=list(order=c(1,1,1), 12))
acf(residuals(m1))

 m1.2<-arima(unemp1.ts, order=c(1,1,2), seas=list(order=c(1,0,1), 12))
acf(residuals(m1.2))

m2<-arima(unemp1.ts, order=c(2,1,1), seas=list(order=c(1,1,0), 12))
acf(residuals(m2))

m3<-arima(unemp1.ts, order=c(1,1,2), seas=list(order=c(1,1,0), 12))
acf(residuals(m3))

m4<-arima(unemp1.ts, order=c(1,1,2), seas=list(order=c(1,1,1), 12))
acf(residuals(m4))

#Testing to see which ARIMA model is the best
AIC(m1); AIC(m2); AIC(m3); AIC(m4)

t.test(m1$coef)
t.test(m2$coef)
t.test(m3$coef)
t.test(m4$coef)

#Ljung-Box test
Box.test(m1$residuals, lag=6, type="Ljung")
Box.test(m1$residuals, lag=12, type="Ljung")
Box.test(m1$residuals, lag=18, type="Ljung")
Box.test(m1$residuals, lag=24, type="Ljung")

#Best ARIMA model
m1<-arima(unemp1.ts, order=c(2,1,1), seas=list(order=c(1,1,1), 12))
par(mfrow=c(1,2))
acf(residuals(m1), main="ACF of residuals of best fitting model")
pacf(residuals(m1), main="PACF of residuals of best fitting model")
coefficients(m1)

#Forecasting 12 steps ahead
forecast = predict(m1, n.ahead=12)
forecast
```

```
my.predict=ts(forecast$pred, st=2016, fr=12)
my.predict

#Create time object out of sample test data
real.data=ts(data$unemp[289:300], st=2016, fr=12)
real.data

#Confidence Intervals
pcil=ts(forecast$pred-1.96*forecast$se, st=2016, fr=12)
pciu=ts(forecast$pred+1.96*forecast$se, st=2016, fr=12)

#Plot of Raw and Forecasted Data
ts.plot(cbind(unemp1.ts,real.data, my.predict, pcil, pciu), lty=c(1,1,5,5,5),
      col=c("black", "black", "red", "blue", "blue"),
      ylab="unemployment rate", main="Unemployment Data and Forecast of the Last 12 Months")

#Root Mean Square Error of Forecast
forecast.error=real.data - my.predict
mse.forecast=sqrt((sum(forecast.error^2))/12)
mse.forecast
```

Appendix 3:
R Script for Section V

```
unemp1.ts <- ts(newdata[,1], start = 1992, fr = 12)
goods1.ts <- ts(newdata[,2], start = 1992, fr = 12)
edu1.ts <- ts(newdata[,3], start = 1992, fr = 12)

plot(unemp1.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016", type="l")
plot(goods1.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016",  type="l")
plot(edu1.ts, main = "Monthly Unemployment Percent in California Jan 1992-Dec 2016",  type="l")

#######
###Differencing the Data
######
#Differencing Unemp Variable
par(mfrow=c(3,1))
diff1=diff(unemp1.ts, lag=1, differences = 1)
acf(diff1)

diff12=diff(unemp1.ts, lag=12, difference=1)
acf(diff12)

diff1diff2.unemp=diff(diff1, lag=12, differences = 1)
acf(diff1diff2.unemp)

#Differencing Goods Variable
diff1=diff(goods1.ts, lag=1, differences = 1)
acf(diff1)
```

```
diff12=diff(goods1.ts, lag=12, difference=1)
acf(diff12)

diff1diff2.goods=diff(diff1, lag=12, differences = 1)
acf(diff1diff2.goods)

#Differencing Edu Variable
diff1=diff(edu1.ts, lag=1, differences = 1)
acf(diff1)

diff12=diff(edu1.ts, lag=12, difference=1)
acf(diff12)

diff1diff2.edu=diff(diff1, lag=12, differences = 1)
acf(diff1diff2.edu)

#########
####Section V.1 Cross Correlation
#########
par(mfrow=c(1,2))
ccf(diff1diff2.unemp, diff1diff2.goods)
ccf(diff1diff2.unemp, diff1diff2.edu)

########
####Section V.2
########
#Unit Root Test
library(tseries)
adf.test(unemp1.ts)
adf.test(goods1.ts)
adf.test(edu1.ts)

#Cointegration Test
po.test(cbind(diff1.unemp, diff1.goods))
po.test(cbind(diff1.unemp, diff1.edu))

########
####Section V.3
########
#Fitting VARS model
library(vars)
AR.data <- ar(newdata, method="burg", dmean=T, intercept=F)
AR.data$order

VAR.data <- VAR(newdata, p=15, type="const")
coef(VAR.data)
```

```
 #Analyzing Residuals of AR(15)
acf(resid(VAR.data)[,1],main="Res of AR, unemp")
acf(resid(VAR.data)[,2], main="Res of AR, goods")
acf(resid(VAR.data)[,3], main="Res of AR, edu")

par(mfrow=c(1,2))
ccf(resid(VAR.data)[,1], resid(VAR.data)[,2])
ccf(resid(VAR.data)[,1], resid(VAR.data)[,3])

########
####Section V.4
########
#Impulse Response Analysis
#Shock unemp and see what happens to unemp and the other variables over time
irf.y=irf(VAR.data, impulse = "unemp", response = "unemp", boot =
        FALSE,n.ahead=40)
irf.x=irf(VAR.data, impulse = "unemp", response = "goods", boot =
        FALSE,n.ahead=40)
irf.z=irf(VAR.data, impulse = "unemp", response = "edu", boot =
        FALSE,n.ahead=40)
#plot the three impulse response variables
plot(irf.y)
plot(irf.x)
plot(irf.z)

#Shock goods and see what happens to goods and the other variables over time
irf.y2=irf(VAR.data, impulse = "goods", response = "unemp", boot =
        FALSE,n.ahead=40)
irf.x2=irf(VAR.data, impulse = "goods", response = "goods", boot =
        FALSE,n.ahead=40)
irf.z2=irf(VAR.data, impulse = "goods", response = "edu", boot =
        FALSE,n.ahead=40)

#plot the three impulse response variables
plot(irf.y2)
plot(irf.x2)
plot(irf.z2)

#Shock edu and see what happens to edu and the other variables over time
irf.y3=irf(VAR.data, impulse = "edu", response = "unemp", boot =
        FALSE,n.ahead=40)
irf.x3=irf(VAR.data, impulse = "edu", response = "goods", boot =
        FALSE,n.ahead=40)
irf.z3=irf(VAR.data, impulse = "edu", response = "edu", boot =
        FALSE,n.ahead=40)
```

```
#plot the three impulse response variables
plot(irf.y3)
plot(irf.x3)
plot(irf.z3)

#######
####Section V.5
#######
#Forecast
VAR.pred <- predict(VAR.data, n.ahead=12)
VAR.pred
unemp.pred <- ts(VAR.pred$fcst$unemp[,1],st=2016,fr=12)

#Confidence Interval
ciu<-ts(VAR.pred$fcst$unemp[,3], st=2016, frequency=12)
cil<-ts(VAR.pred$fcst$unemp[,2], st=2016, frequency=12)

#Put predicted values in the time plot
ts.plot(cbind(window(unemp1.ts, start = 1992), unemp.pred, cil, ciu),
        lty=c(1,2,2,2),col=c("Black", "Red", "Blue", "Blue"), ylab="unemployment rate",
        main="Unemployment Data and Forecast of the Last 12 Months")

#RMSE
real.data=ts(data$unemp[289:300], st=2016, fr=12)
real.data

var.forecast.error=real.data - unemp.pred
var.mse.forecast=sqrt((sum(var.forecast.error^2))/12)
var.mse.forecast
```

Appendix 4:
R Script for Section VI

```
data <- read.csv("C:/Users/sabap/Desktop/170 Project/PartI/Paya-104483616-Installment3.csv")
newdata<-data[1:288,-1]

#Turning All Variables into Time Series
unemp.ts <- ts(newdata[,1], start = 1992, fr = 12)
goods.ts <- ts(newdata[,2], start = 1992, fr = 12)
edu.ts <- ts(newdata[,3], start = 1992, fr = 12)

data.ts=cbind(unemp.ts, goods.ts, edu.ts)

#Fitting a Regression Model
model<-lm(unemp.ts~ goods.ts+edu.ts+factor(months))
par(mfrow=c(1,2))
acf(model$residuals)
pacf(model$residuals)
```

```
#Checking the Residuals and Other Assumptions
par(mfrow=c(2,2))
plot(y=rstudent(model), x=as.vector(time(unemp.ts)), xlab="time",
    ylab="Standardized Residuals", type="o")
abline(h=0)
acf(rstudent(model))
hist(rstudent(model), xlab="Standardized Residuals")
qqnorm(rstudent(model))

dev.off()

#Finding the AR fit
acf(ts(rstudent(model)))
pacf(ts(rstudent(model)))

reg.AR1 <- arima(ts(model$residuals), order=c(13,0,0), include.mean = F)
acf(residuals(reg.AR1))

#Creating a Dummy Variable
months=cycle(unemp.ts)
months

model1=lm(data.ts~factor(months)-1); summary(model1)
model2=lm(data.ts~factor(months)); summary(model2)

plot(ts(fitted(model2), freq=12, start=c(1992,1)), col="red", ylab="data and fitted values", type="l",
    ylim=range(c(fitted(model2), data.ts)))
lines(data.ts)

#Fitting a Seasonal Model
times=time(unemp.ts)
model1<-lm(unemp.ts~goods.ts+edu.ts+times+times^2+times^3+factor(months))
summary(model1)
acf(residuals(model1))
pacf(residuals(model1))

arima.mod=arima(ts(rstudent(model1)), order=c(13,0,0))
acf(residuals(arima.mod))

dev.off()

###GLS Model Seasonal Dummies and Time Trend
library(nlme)
gmodel=gls(unemp.ts~goods.ts+edu.ts+factor(months)+times+times^2+times^3, correlation =
         corARMA(c( 0.7095,  0.2357,  -0.0229, -0.0358,
                    0.0073, 0.0090,  -0.0336, 0.0354,
                    0.1371, -0.0069,  -0.0842,  0.2485,
                   -0.2557), p=13))
gmodel
summary(gmodel)
acf(residuals(gmodel))

#ACF of GLS model
```

```
ystar=unemp.ts[-1]- 0.7095 *unemp.ts[-length(unemp.ts)] - 0.2357*unemp.ts[-length(unemp.ts)]
+ 0.0229*unemp.ts[-length(unemp.ts)] + 0.0358*unemp.ts[-length(unemp.ts)] -0.0073 *unemp.ts[-
length(unemp.ts)] - 0.0090*unemp.ts[-length(unemp.ts)] + 0.0336*unemp.ts[-length(unemp.ts)]
0.0354*unemp.ts[-length(unemp.ts)] - 0.1371*unemp.ts[-length(unemp.ts)] + 0.0069*unemp.ts[-
length(unemp.ts)]  + 0.0842*unemp.ts[-length(unemp.ts)] - 0.2485*unemp.ts[-length(unemp.ts)] +
0.2557*unemp.ts[-length(unemp.ts)]

xstar=goods.ts[-1]- 0.7095 *goods.ts[-length(goods.ts)] - 0.2357*goods.ts[-length(goods.ts)]
+ 0.0229*goods.ts[-length(goods.ts)] + 0.0358*goods.ts[-length(goods.ts)] -0.0073 *goods.ts[-
length(goods.ts)]
- 0.0090*goods.ts[-length(goods.ts)] + 0.0336*goods.ts[-length(goods.ts)] -  0.0354*goods.ts[-
length(goods.ts)]
- 0.1371*goods.ts[-length(goods.ts)] + 0.0069*goods.ts[-length(goods.ts)]  + 0.0842*goods.ts[-
length(goods.ts)]
- 0.2485*goods.ts[-length(goods.ts)] + 0.2557*goods.ts[-length(goods.ts)]

zstar=edu.ts[-1]- 0.7095 *edu.ts[-length(edu.ts)] - 0.2357*edu.ts[-length(edu.ts)]
+ 0.0229*edu.ts[-length(edu.ts)] + 0.0358*edu.ts[-length(edu.ts)] -0.0073 *edu.ts[-length(edu.ts)]
- 0.0090*edu.ts[-length(edu.ts)] + 0.0336*edu.ts[-length(edu.ts)] -  0.0354*edu.ts[-length(edu.ts)]
- 0.1371*edu.ts[-length(edu.ts)] + 0.0069*edu.ts[-length(edu.ts)]  + 0.0842*edu.ts[-length(edu.ts)]
- 0.2485*edu.ts[-length(edu.ts)] + 0.2557*edu.ts[-length(edu.ts)]

mod=lm(ystar~xstar+zstar)
acf(ts(mod$residuals))

#Forecast Plot
new.t<-seq(2016, len=288, by=1/12)
new.dat2=data.frame(months=rep(1:12,1), times=new.t)
predicted2=predict(gmodel, new.dat2)[1:12]

hat.x=ts(predicted2, st=2016, fr=12)
ts.plot(unemp.ts, hat.x, lty=1:2, main="Unemployment Data and Forecast of Last 12 Months",
ylab="Value of Unemp", xlab="Time", col=c("black", "red"))

#RMSE
real.data=ts(data$unemp[289:300], st=2016, fr=12)
real.data

reg.forecast.error=real.data - predicted2
reg.mse.forecast=sqrt((sum(reg.forecast.error^2))/12)
reg.mse.forecast
```

Appendix 5:

R Script for Section VII

```
unemp.ts <- ts(newdata[,1], start = 1992, fr = 12)
exp2<- HoltWinters(unemp.ts, beta = FALSE, gamma=FALSE) #without trend and without seasonal
component
exp3<- HoltWinters(unemp.ts, gamma=FALSE)  #with trend and no seasoanl component
exp4<- HoltWinters(unemp.ts, seasonal = "add")
exp5<- HoltWinters(unemp.ts, seasonal = "mult")

plot(unemp.ts)
lines(fitted(exp2)[, 1], type="l", col="red")
```

```
plot(unemp.ts)
lines(fitted(exp3)[, 1], type="l", col="red")
plot(unemp.ts)
lines(fitted(exp5)[, 1], type="l", col="red")
plot(unemp.ts)
lines(fitted(exp4)[, 1], type="l", col="red")

exp.mse.forecast2=sqrt(exp2$SSE/length(data$unemp))
exp.mse.forecast2
exp.mse.forecast3=sqrt(exp3$SSE/length(data$unemp))
exp.mse.forecast3
exp.mse.forecast4=sqrt(exp4$SSE/length(data$unemp))
exp.mse.forecast4
exp.mse.forecast5=sqrt(exp5$SSE/length(data$unemp))
exp.mse.forecast5

#Forecast 12 steps ahead
exp.predict=predict(exp4, n.ahead=12)
exp.predict

#Plot with forecasted values
ts.plot(unemp.ts, predict(exp4, n.ahead=12), main="Forecasted values (dashed line)", lty=1:2,
col=c("black", "red"))

#RMSE
real.data=ts(data$unemp[289:300], st=2016, fr=12)
real.data

exp.forecast.error=real.data - exp.predict
exp.mse.forecast=sqrt((sum(exp.forecast.error^2))/12)
exp.mse.forecast

#####################
#####Conclusion######
#####################
 #Calculating Average RMSE of all 4 forecasts
real.data=ts(data$unemp[289:300], st=2016, fr=12)
real.data

avg.data=c(6.6208, 6.433,6.324,5.777,5.633,6.021,6.294,
          5.99,5.609,5.614,5.667,5.539)

forecast.error=real.data - avg.data
mse.forecast=sqrt((sum(forecast.error^2))/12)
mse.forecast

#Calculating RMSE for Short term
real.data1 <- ts(c(5.9, 5.8), st = 2016, frequency = 12)
my.predict1 <- ts(c(6.164,5.958), st = 2016, frequency = 12)
error.forecast1 <- real.data1 - my.predict1
arima.mse.forecast1 <- sqrt((sum(error.forecast1^2))/2)
arima.mse.forecast1
```

```
real.data2 <- ts(c(5.9, 5.8), st = 2016, frequency = 12)
my.predict2 <- ts(c(6.137,5.918), st = 2016, frequency = 12)
var.forecast.error1 <- real.data2 - my.predict2
var.mse.forecast1 <- sqrt((sum(var.forecast.error1^2))/2)
var.mse.forecast1

real.data3 <- ts(c(5.9, 5.8), st = 2016, frequency = 12)
my.predict3 <- ts(c(7.991,7.858), st = 2016, frequency = 12)
reg.forecast.error1 <- real.data3 - my.predict3
reg.mse.forecast1 <- sqrt((sum(reg.forecast.error1^2))/2)
reg.mse.forecast1

real.data4 <- ts(c(5.9, 5.8), st = 2016, frequency = 12)
my.predict4 <- ts(c(6.191,5.998), st = 2016, frequency = 12)
exp.forecast.error1 <- real.data4 - my.predict4
exp.mse.forecast1 <- sqrt((sum(exp.forecast.error1^2))/2)
exp.mse.forecast1

real.data5 <- ts(c(5.9, 5.8), st = 2016, frequency = 12)
my.predict5 <- ts(c(6.621,6.433), st = 2016, frequency = 12)
avg.forecast.error1 <- real.data5 - my.predict5
avg.mse.forecast1 <- sqrt((sum(avg.forecast.error1^2))/2)
avg.mse.forecast1
```