Time Series Analysis on California Unemployment Insurance Claims

Jiayi Huang [SID: 26013785]

UC Berkeley Stat 153 Final Project Fall 2017

Professor David R. Brillinger

Table of Contents

Part 1: Introduction

It is essential for the Office of Unemployment Insurance in California to analyze the relationship between time and Unemployment Insurance Claims in order to predict how much money the department should set aside as the reserve to cover potential losses in the near future. The time series model of the weekly California Unemployment Insurance Claims can also give us some insights into the long-lasting macroeconomics problem of unemployment and government spending on social welfare.

Part 2: The Question

The scientific question motivating my work is "How well do different time series models predict weekly California unemployment insurance claims based on its historical claims?"

Part 3: The Data and Methods

This project utilizes weekly claim data from California Unemployment Insurance Claims (in number of claimed filed) along with weekly California Unemployment Rates (%) since 1986 published by the Office of Unemployment Insurance under the United States Department of Labor. The data starts on the filed week of January 3rd, 1987 and ends on the filed week of October 7th, 2017, which reflects the weeks ending in between December 27th, 1986 and September 30th, 2017. There are 1601 weekly observations. Every observation contains the weekly claim amount filed in California. I will split the dataset into a training set and a test set. The training set will be used to train my predictive time series models and the test set will be used to access of goodness of fit of the models. At the end, I will use the best performing models to predict the weekly California unemployment insurance claims for the remaining of 2017 to September 30th, 2018, or one year into the future.

Data Description

With the original raw dataset, each row represents a week's information. The original columns that are useful to my analysis are 'Initial Claim', the number of jobless claims filed by individuals seeking to receive unemployment insurance (UI) benefits during the most recent filed week; 'Reflecting week ended', the week during which the claims were filed; 'Continued Claim', the number of unemployed workers that qualify for benefits under UI, who have been covered by UI.

Missing Data

I noticed that there are 5 weeks of missing data (weeks ended on 1987-03-14, 1987-06-27, 1987-07-04, 1987-07-11, 1987-07-18). I manually imputed data for those weeks with missing data by assuming uniform changes in claims between the weeks before and after the missing week; thus I took the average of those two weeks' data values to be the value for the missing week(s).

New Variable - Total Claims

Since I am interested in both the number of weekly stock (old) and flow (new) insurance claims, I created a new variable, Total Claims, which represents the total number of UI claims in a given week by summing the two types of jobless claims, initial claims and continued claims, in that week. I will proceed with total claims for the rest of my analysis.

Time Series Conversion

I converted the total claims amount described above into a time series dataset in R starting in week (1986-12-27 in decimal form) while taking leap years into account.

Part 4: Time Series Modeling

My goal is find the best performing predictive models within each of the three types of time series models, ARIMA, ARCH-GARCH, and Spectral Model to fit my data. I will first examine the stationarity and seasonality assumptions, choosing a training set and a test set, fitting several models for

each type of time series model, and determining the best model for each type based on predictive performance using my test set by comparing the root mean squared residuals and Akaike information criterion (AIC) when applicable.

Section 4.1: Exploratory Data Analysis

I conducted some initial exploratory data analyses by looking at the summary of my time series. The minimum weekly total UI claim is 251600, the median is 478500, the mean is 499600, and the maximum is 940800. After filling the missing data, I now have a total of 1606 (including the 5 weeks of manually added values) weeks' worth of total UI claims.

After plotting the initial time series data (Figure 1), I see that there are quite a lot of fluctuations.

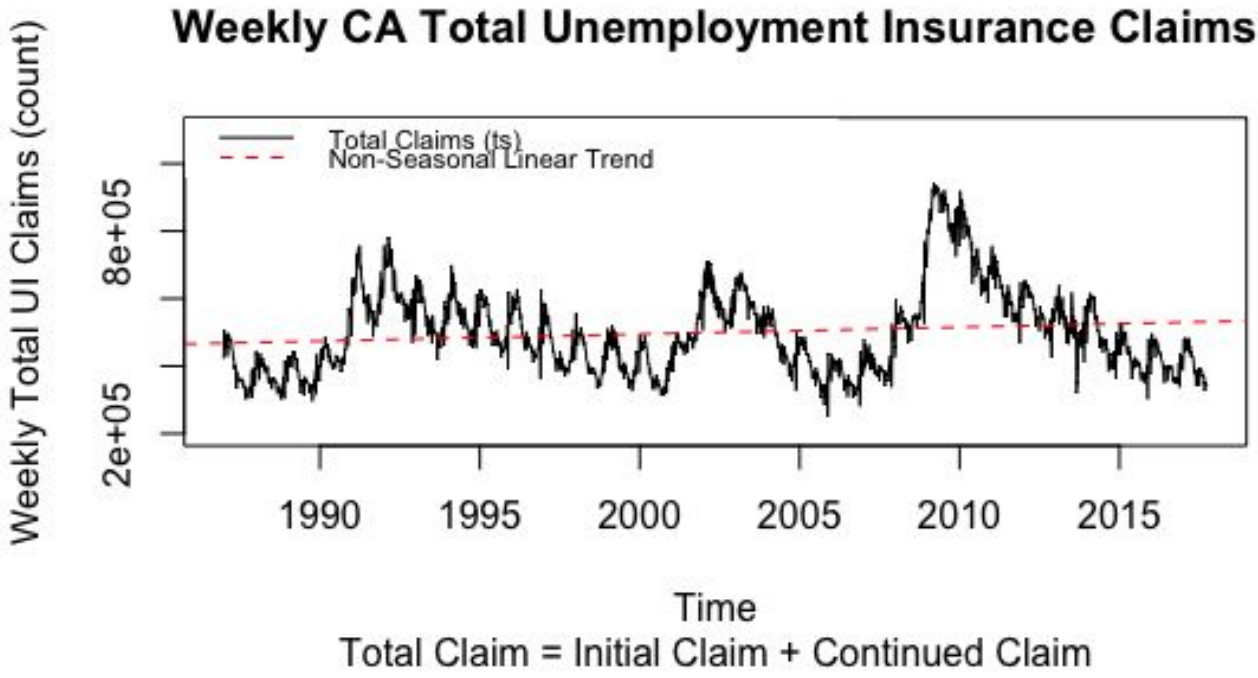


Figure 1

There are many spikes with large variances. The non-season linear trend shows a slight upward, positive linear trend from 1985 to 2017. As time progresses, the average amount of weekly total claims increases as well. However, the data clearly has a downward trend starting around 2010, which is contradictory to the overall trend. This raises a potential problem with whether I am using values too far back into the past when forecasting. This problem is address later in Section 4.3.

Section 4.2: Examining Stationarity Assumptions

The original time series plot (Figure 1) does not seem stationary with constant means and variances. The log-scale of the original time series is not stationary either. Both the first difference of the original time series and the first difference of the log-scale of original time series are stationary and non-explosive. The stationarity conclusions are also based on the results of Augment Dickey-Fuller Tests (Figure 2). For simplicity, I will use the first difference of the original time series to improve stationarity while not introducing unnecessary correlations.

	DF(H1: Stationary)	P-value	DF(H1: Explosive)	P-value	Stationary?	Non-Explosive?
ts	-2.485629	0.37273485	-2.485629	0.6272652	FALSE	TRUE
log(ts)	-2.140385	0.51888704	-2.140385	0.4811130	FALSE	TRUE
diff(ts)	-3.583148	0.03429529	-3.583148	0.9657047	TRUE	TRUE
diff(log(ts))	-3.887578	0.01461002	-3.887578	0.9853900	TRUE	TRUE

Figure 2

Section 4.3: Training Set and Test Set

TRAINING SET (2012-09-29 TO 2016-10-01, N = 210)

Since the time series trend after 2010 seems to be contradicting the overall positive non-seasonal trend (1986–2017) shown in Figure 1, I suspect that I am using past data points that are no longer predicative of new data points. After testing a few ARIMA models with the full training set which contains all data points from 1986 - 2016, the forecasts seemed somewhat inaccurate. Taking instructor's advice, I decided to forego all the data prior to 2012 since I believe the economic activities (e.g. 2008 Great Recession) had unusual effects on the time series itself and they were not long term permanent economic shocks, and therefore not as relevant for me when predicting the claims in 2017 and on. I decided to create a new training set which contain only values starting on 2012-09-29 to

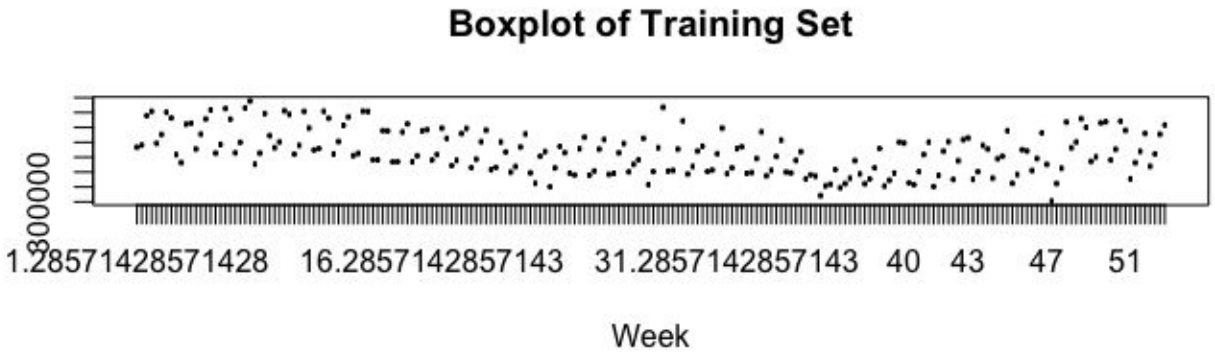
2016-10-01. The rest of the analysis will use this new training set with which the stationary assumption still holds.

Test Set (2016-10-08 to 2017-09-30, n = 52)

Since I am predicting weekly total claims 1 year in advance starting on 10/07/2017, I will save the last 1 year (52 weeks) (10/08/2016 - 09/30/2017) as my test set.

Section 4.4: Examining Seasonality Assumptions

I suspect that there are seasonal trends within the weekly total UI claims since there is seasonal unemployment throughout each year. I employed various graphical methods to examine the seasonality



assumption for the training set. The boxplot (Figure 3) shows potential seasonal trends with varying weekly averages. The ACF suggests a clear polynomial trend that could be caused by seasonality. The ACF of the first difference shows such trend as well, which is undesirable. The ACF of the first and seasonal difference eliminates most, if not all, of these undesired trends. This results supports my assumption of the seasonal weekly trends in my training set. (Figure 4).

Figure 3

Figure 4

Section 4.5: Seasonal ARIMA Modeling [Time Domain]

After confirming the stationarity and weekly seasonality of my training set, I fitted several seasonal ARIMA models to access the goodness of fit. Lags of Models 2.1-2.6 were chosen after

examining the ACF of the first difference to determine q, second difference to determine Q; PACF of the first difference to determine p, second difference to determine P; EACF of the first difference to determine p and q, second difference to determine P and Q (Figure 5). I determined the best model using two criteria, AIC and RMSR of the predicted values versus the test set actual values. The results are consolidated in Figure 6 for comparison. The best performing ARIMA model is ARIMA(1,1,2)(1,1,0)[52], its training forecast plots are shown in Figure 7 and 8.

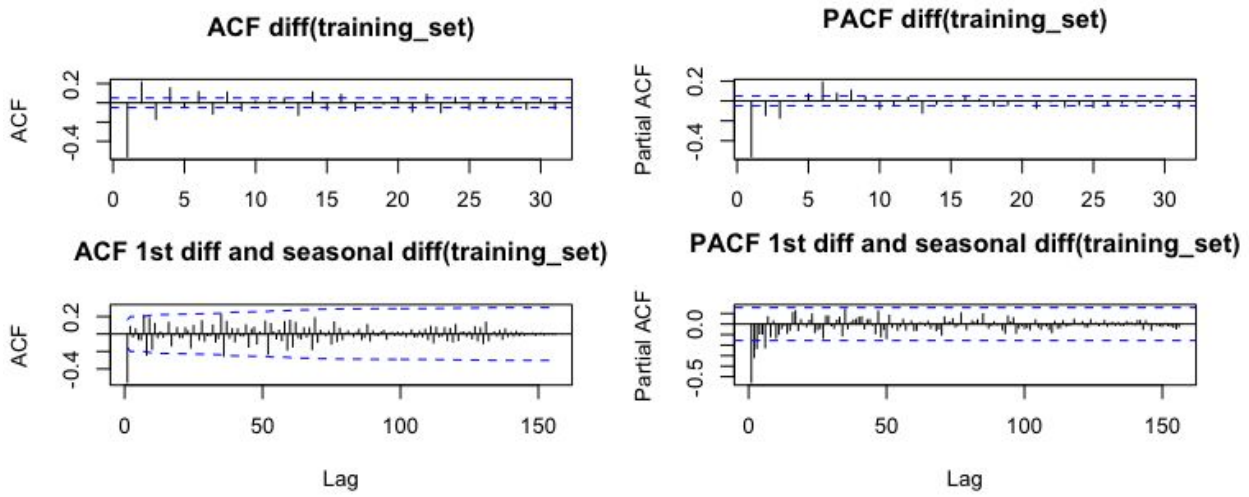


Figure 5

Model	ARIMA(p,d,q)(P,D,Q)[S]	AIC	Root Mean Squared Residuals	Min AIC	Min RMSR
1	(1,1,2)(1,0,0)[52]	4971.269	27494.18	FALSE	FALSE
2.1	(1,1,2)(1,0,1)[52]	4970.450	26913.72	FALSE	FALSE
2.2	(1,1,2)(2,0,1)[52]	4969.185	31429.65	FALSE	FALSE
2.3	(1,1,2)(1,1,1)[52]	3745.835	19596.75	FALSE	FALSE
2.4	(1,1,2)(1,1,2)[52]	3747.848	19564.94	FALSE	FALSE
2.5	(1,1,2)(1,1,0)[52]	3743.836	19488.03	TRUE	TRUE
2.6	(1,1,2)(2,1,0)[52]	3745.765	20328.94	FALSE	FALSE

Figure 6

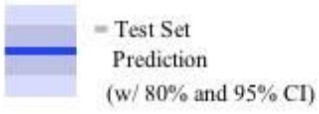
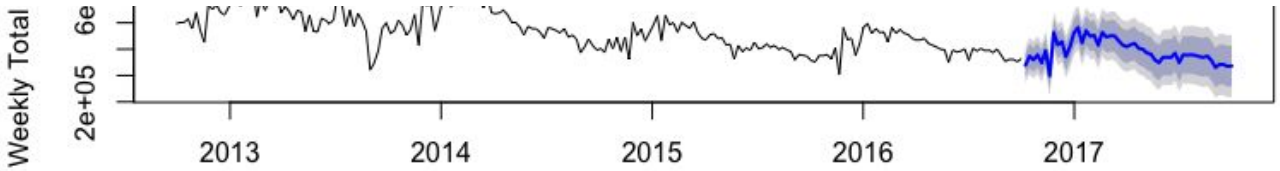


Figure 7



s **(ARIMA(1,1,2)(1,1,0)[52]): Predicted(Red)vs.Actual(Green) Weekly Total Claims**

Figure 8

Section 4.6: ARCH-GARCH Modeling [Time Domain]

First, I ran the McLeod-Li Test to test the validity of ARCH-GARCH models. The result shows that every lag is statistically significant starting at lag 1 and formally confirms strong evidence that ARCH/GARCH may be necessary to model this data. The results of fitting several ARCH-GARCH models to the training set are consolidated in Figure 9. The best performing ARCH-GARCH model with the least RMSR is ARMA(2,2)-GARCH(1,1). Even though its AIC is slighter larger than ARMA(3,3)-GARCH(1,1), it uses less parameters which would result in a lower BIC value. The training forecast plots (one with variance and one plotted against the test set) are shown in Figure 10.

Model	GARCH(p,q) ARMA(p,q)	AIC	Root Mean Squared Residuals	Min AIC	Min RMSR
1	GARCH(1,1) ARMA(0,0)	25.03158	80529.43	FALSE	FALSE
2	GARCH(2,2) ARMA(0,0)	25.04525	80529.51	FALSE	FALSE
3	GARCH(1,1) ARMA(1,1)	23.7762	51320.98	FALSE	FALSE
4	GARCH(1,1) ARMA(2,2)	23.76575	51002.34	FALSE	TRUE
5	GARCH(1,1) ARMA(3,3)	23.70934	53002.78	TRUE	FALSE
6	GARCH(1,1) ARMA(1,2)	23.77021	51540.52	FALSE	FALSE
7	GARCH(1,1) ARMA(2,1)	23.7753	51542.30	FALSE	FALSE

Figure 9

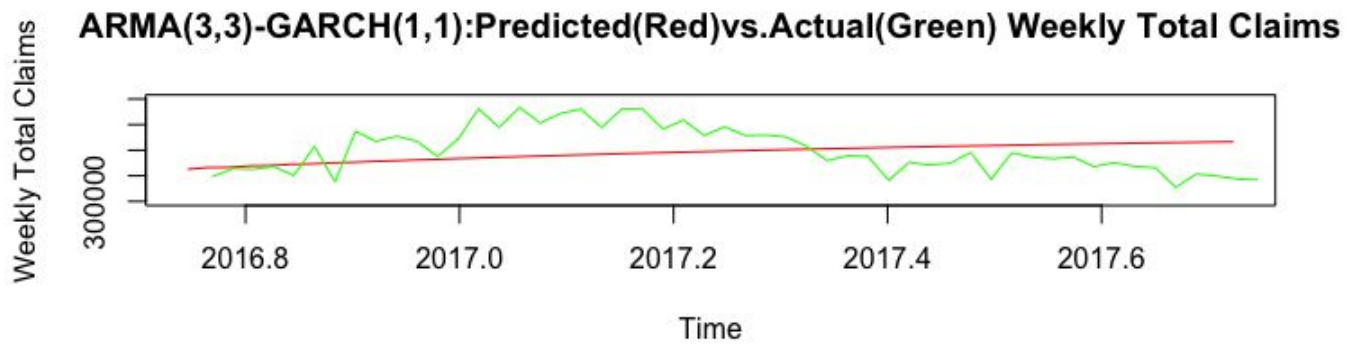


Figure 10

Section 4.7: Spectral Analysis and Modeling [Frequency Domain]

Since my data looks to have some seasonality, this is motivation for me to consider a frequency-side analysis. I first plotted the periodogram (Figure 11) of my training set. It turned out that the top 2 frequencies (freq1, freq2) are nearly zero, meaning that smaller frequencies contribute the most to my data, and that the Fourier Transformation of my data will look very flat as it were a linear model. Here cos1 is defined to be cos(2*pi*t*freq1), sin1 is defined to be sin(2*pi*t*freq1), cos2 and sin2 are defined in a similar manner. The best performing model with the least RMSR is a quadratic fit using two cosine-sine pairs. The training forecast against the test set is shown in Figure 13).

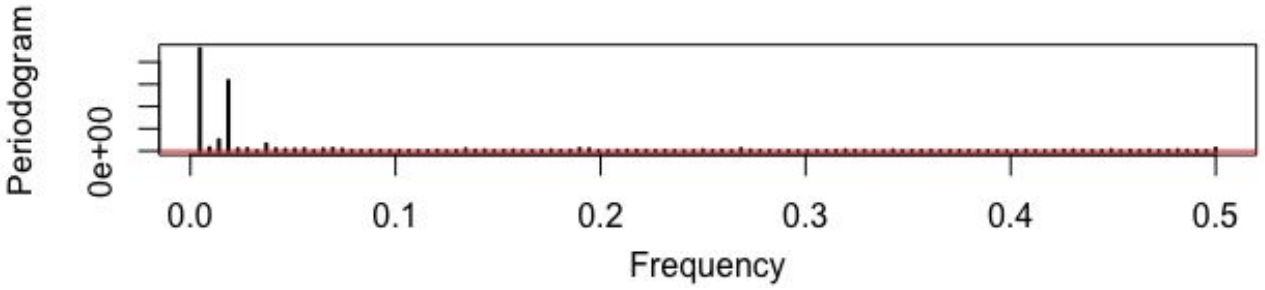


Figure 11

Model	Frequency Model	Root Mean Squared Residuals	Min RMSR
1	lm(y ~ cos1 + sin1)	61291.19	FALSE
2	lm(y ~ cos1 + sin1 + cos2 + sin2)	108862.5	FALSE
3	lm(y ~ poly(cos1 + sin1, 2))	61103.77	FALSE
4	lm(y ~ poly(cos1 + sin1 + cos2 + sin2, 2))	59644.69	TRUE

Figure 12

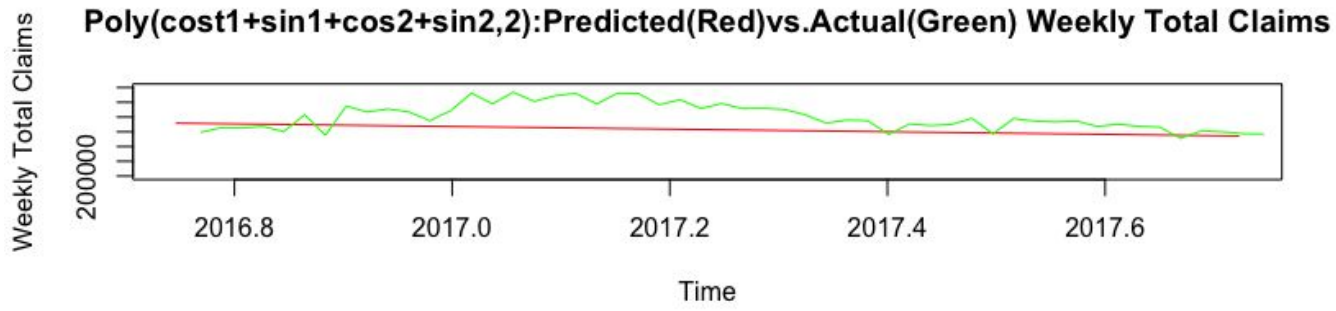


Figure 13

Part 5: Model Forecasting

Out of the best performing models from each of the three types of time series models found earlier, ARIMA(1,1,2)(1,1,0)[52], ARMA(2,2)-GARCH(1,1), and quadratic (cos1 + sin1 + cos2 + sin2), ARIMA(1,1,2)(1,1,0)[52] has the least RMSR, which making it the most accurate predictive model out of all the models I fitted in the analysis. I will use ARIMA(1,1,2)(1,1,0)[52] to forecast the future.

Seasonal ARIMA Forecasting [Time Domain]

Using ARIMA(1,1,2)(1,1,0)[52], I forecasted the weekly total UI claims from September 30th, 2017 to September 30th, 2018. I passed in all the data from 2012/09/29 to 2017/09/30 in order to forecast for the weekly total UI claims from 10/07/2017- 09/30/2018. The forecast plot is shown on Figure 14, with shaded regions indicating the 80% and 95% confidence intervals.

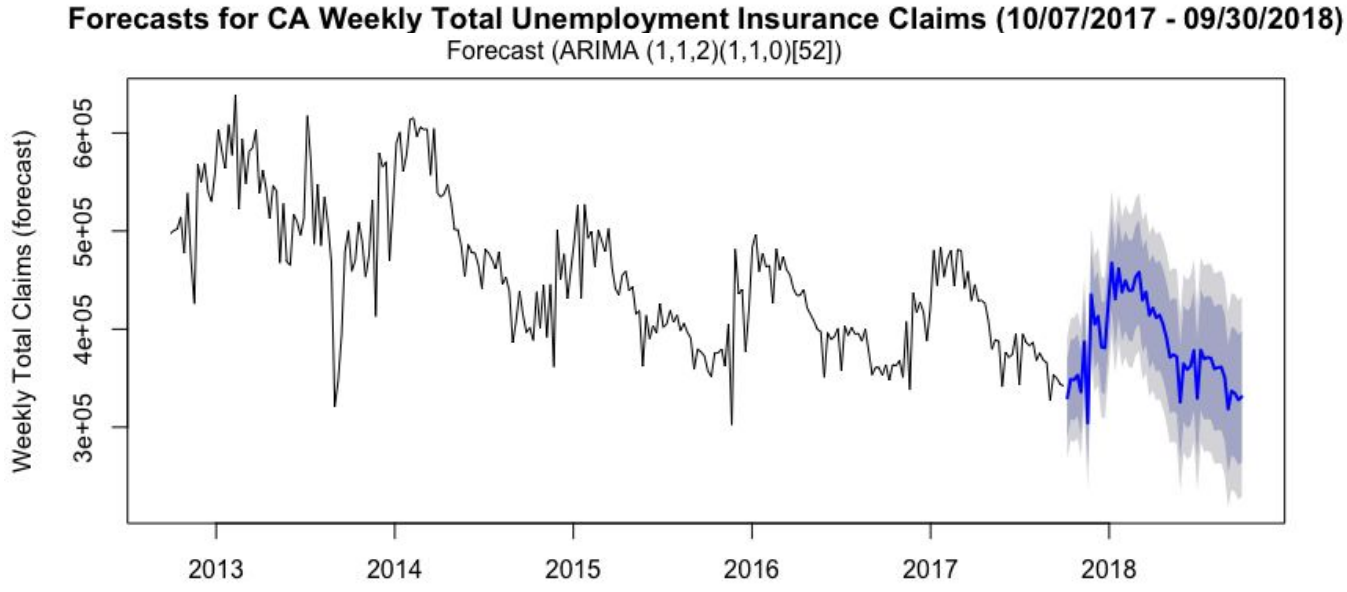


Figure 14

Part 6: Model Diagnostic

Seasonal ARIMA Model Diagnostics [Time Domain]

I first checked the periodogram of the residuals from ARIMA(1,1,2)(1,1,0)[52] to see whether the residuals have no clear pattern and are independently identically normally distributed. The periodogram shows a somewhat uniform random distribution of all frequency which confirms the randomness of the residuals. Then I checked the residual plot, ACF of residuals, the histogram of residuals, and the QQ plot for residuals, all of which show evidence that the residents are approximately normally distributed. (Figure 15)

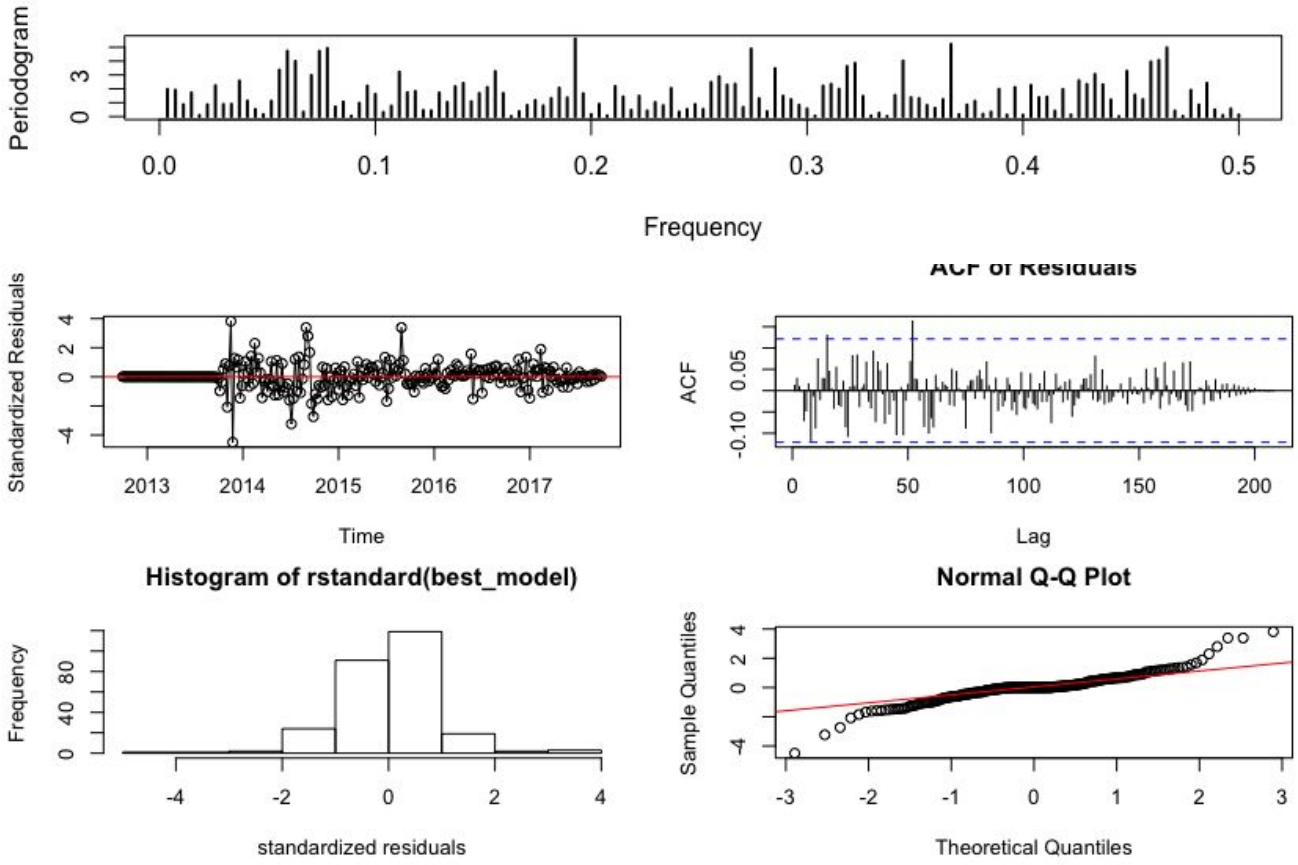


Figure 15

Part 7: Conclusion

The answer to my question, "How well do different time series models predict weekly California unemployment insurance claims based on its historical claims?" is: Out of three possible types of time series models (ARIMA, ARCH-GARCH, Spectral), Seasonal ARIMA(1,1,2)(1,1,0)[52] fits the time series of weekly California total unemployment insurance claims (initial claims + continued claims) the best with the least root mean squared residuals (RMSR), and therefore provides the most accurate predictions for the upcoming year (September 30th, 2017 to September 30th, 2018). ARCH-GARCH and Spectral Models were able to predict the general trends (mostly linear) of the total claims but had quite large RMSR, making them inferior to the ARIMA(1,1,2)(1,1,0)[52] model. Not only does Seasonal ARIMA(1,1,2)(1,1,0)[52] provide the most accurate predictions out of all the time series models fitted, it also satisfies the assumptions of stationarity, seasonality, and residual randomness and normality.

Economic Interpretation

From the forecast of weekly California total UI claims (10/07/2017- 09/30/2018) using model ARIMA(1,1,2)(1,1,0)[52], it looks like the year-long forecasted trend is very similar in shape comparing to that of the previous years starting in 2012. The average of the number of weekly California total UI claims seems to decrease steadily in the forecasted data, following the overall downward trend of the previous years. An economic interpretation of this forecast result is that California has been receiving less and less UI claims after the huge, unexpected increase during and shortly after the Great Recession in 2008. It is also reasonable to suspect that the unemployment rate in California will continue decreasing at a small rate during the next year. This shows signs of convergence to a healthy economic state with relatively low and stable unemployment rate. The forecast result would suggest that the state

government of California to reduce its reserve on unemployment insurance funds at a constant rate during the next year to match the constant decrease of unemployment insurance claims.

Future Direction/Further Research

To further my analysis, I would fit more models of ARIMA, ARCH-GARCH, and Spectral to see if there is a better model to predict the future weekly California total UI claims than ARIMA(1,1,2)-(1,1,0)[52]. One other possible predictive model that I would use is the Autoregressive Distributed Lag model and incorporate the unemployment rate in California to hopefully improve the predications of weekly California total unemployment insurance claims in the future.

Part 8: References

THE DATA

Link to original dataset{https://workforcesecurity.doleta.gov/unemploy/claims_arch.asp})

RESOURCES

Professor David R. Brillinger

Andre Waschka

CRYER, J. D. AND CHAN, K. 2011. *Time series analysis*. New York: Springer.

ran.r-project.org. 2017. https://cran.r-project.org/web/packages/rugarch/rugarch.pdf

Portfolio Probe, www.portfolioprobe.com/2012/07/06/a-practical-introduction-to-garch-modeling/.

"Forecasting time series using ARMA-GARCH in R." *Cross Validated*, stats.stackexchange.com/questions/254101/forecasting-time-series-using-arma-garch-in-r.

[R] garch prediction, stat.ethz.ch/pipermail/r-help/2008-April/158510.html.

Part 9: Appendix (R Code)