Long-term Unemployment Time-Series Forecasting

Anh Nguyen - xc8374

Tong Xu - IM7348

Quyen Vu - sn6244

Summary

Long-term unemployment is when workers are jobless for 27 weeks or more. To be counted as such by the Bureau of Labor Statistics, they must have actively sought employment during the previous four weeks. As the unemployment rate in the U.S is reported to have dropped to the lowest level since 1969, job postings are expected to increase significantly and it should be easier to find jobs. A question we care about is what has been the trend for the long-term unemployment group, whether long-term unemployment rate has dropped also or that reported lowest rate is just for the short-term unemployment workers only. Long-term unemployment is less mentioned but is definitely a vital element which should be considered to assess the condition of the economy in general, and of the unemployment, in particular. This project report presents 3 basic time-series forecasting models of long-term unemployment based on labor force flows data extracted from the website of Federal Reserve Bank of St. Louis. We ran 3 types of model on the dataset : SES, Holt-Winter's and Auto Arima. All our models has approximate low MAPE. However, overfitting is a main thing that may have led to the high accuracy of the models.

Introduction

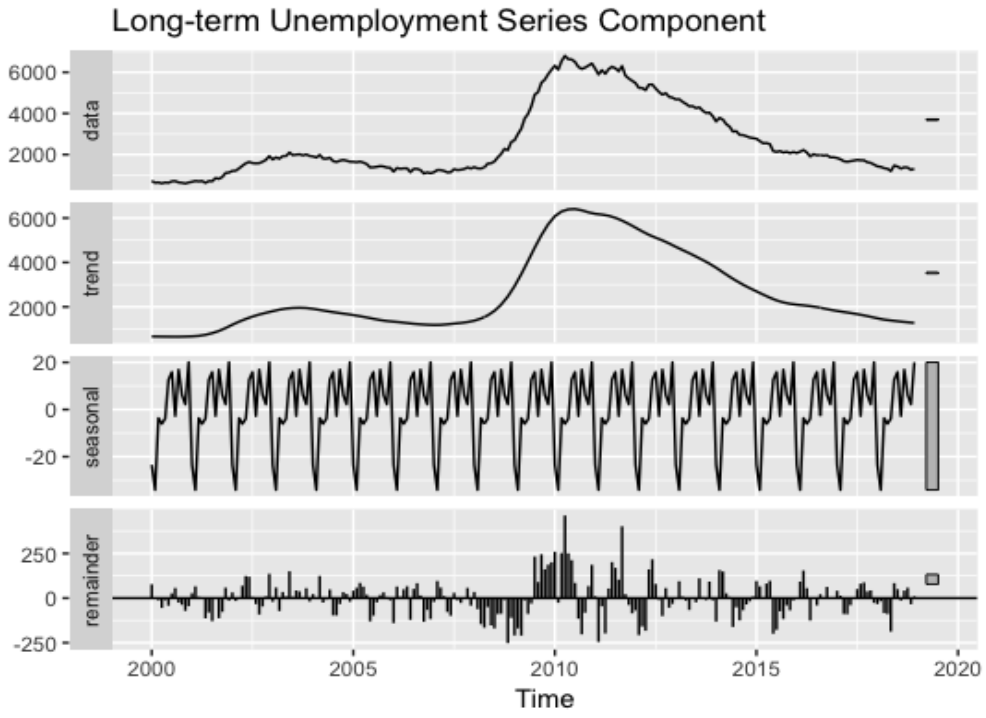
As described on the website of Federal Reserve Bank of St. Louis., the dataset is originally provided by the U.S. Bureau of Labor Statistics, with monthly data on the unemployment rate in the U.S by gender from year 1948 to 2018. The series comes from the 'Current Population Survey (Household Survey)'. As we only wanted to analyze the unemployment rate for a shorter specific period, we chose to retrieve the data from fred.stlouisfed.org, taking period from 2000 Jan – 2018 Dec. The website has the option that allowed us to customize any time range of our choice, which made extracting a csv file from 2000 to 2018 very easy.

The dataset contains 228 data periods, it has no missing values and is already cleaned so no further step of preparing the data is required. The dataset is ready for us to do analyzing, and a part of it is shown below:

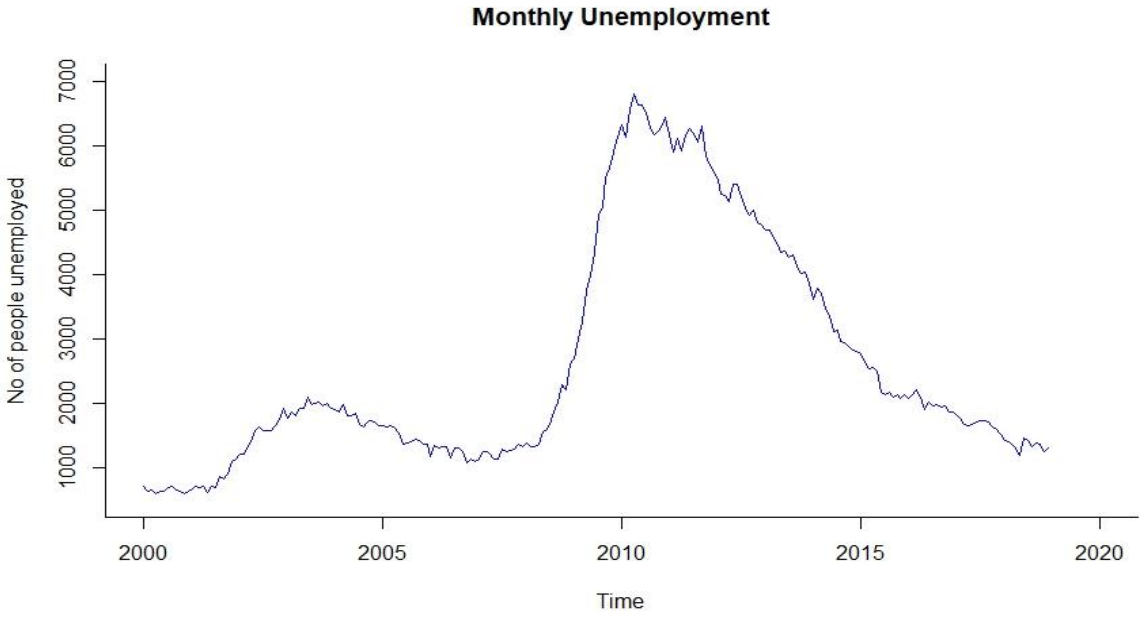
	observation_date	UEMP27OV
1	2000-01-01	721
2	2000-02-01	629
3	2000-03-01	646
4	2000-04-01	599
5	2000-05-01	643
6	2000-06-01	627
7	2000-07-01	698
8	2000-08-01	709
9	2000-09-01	646
10	2000-10-01	627
11	2000-11-01	593
12	2000-12-01	642

Main Chapter

Data Exploration and Visualization

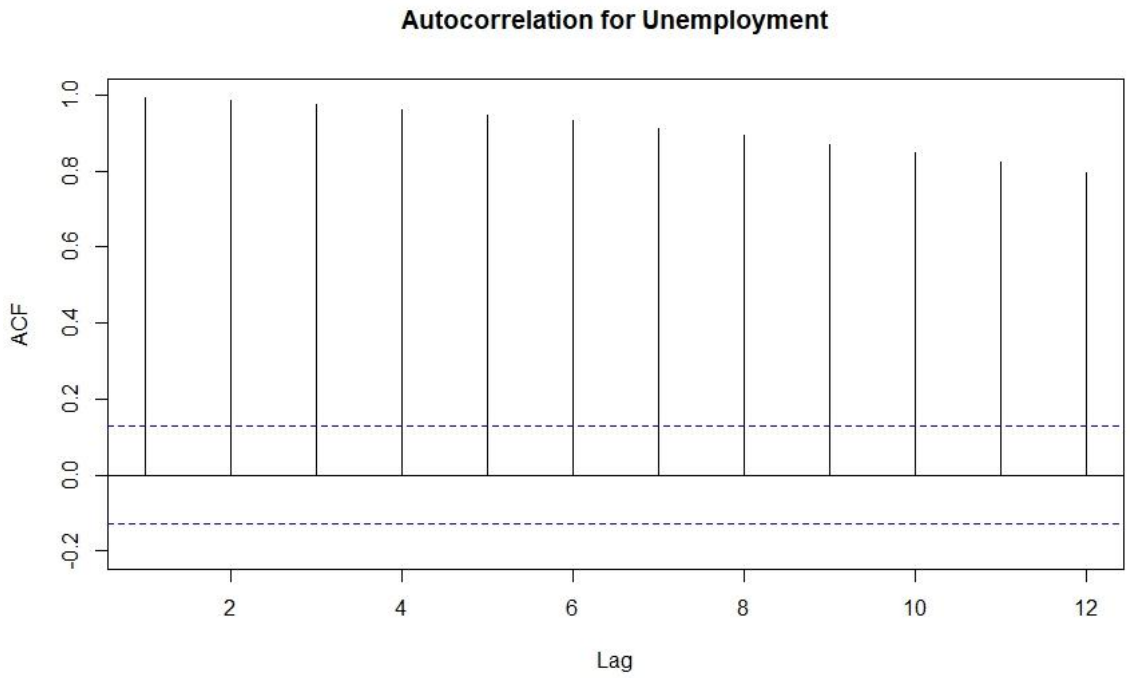


Firstly, we used the stl() function to decompose the dataset. As could be seen from the graph above, the dataset showed an overall of going upward from 2000 to central period (2010-2012) and then going down afterward. To be more specific, the number of long-term unemployment counts according to each time points are demonstrated in the graph below:



As we could see from the graph, from 2000 to 2008, though going up and down, the number of unemployment people observed its highest level in 2004 at around 2000 thousand people. This figure decreased after that to around 1500 thousand people in 2008 and soared crazily from 2009 to peak at almost 7000 thousand people in 2010. This trend seems accordingly reflect the Great Recession of the United States at that period. "According to the Department of Labor, roughly 8.7 million jobs (about 7%) were shed from February 2008 to February 2010, and real GDP contracted by 4.2% between Q4 2007 and Q2 2009, making the Great Recession the worst economic downturn since the Great Depression (of 1930)"[1].The number of long-term unemployed people was demonstrated to decrease gradually after that as the economy slowly got recovered, more jobs were created.

[1] Wikipedia https://en.wikipedia.org/wiki/Great_Recession_in_the_United_States



We then applied the Acf() function to identify possible time series components. As can be observed from the plot above, positive autocorrelation coefficients are present for all the lags. The first 3 lags are the highest and the slowly decreases which indicates an upward trend. Lag 12 is high above horizontal thresholds which indicates monthly seasonality. Since all the lags are statistically significant, the unemploy.ts time series is not a random walk and therefore, predictable.

In order to double check the predictability of the data, we use the first differencing of the historical data and Acf() function to confirm our finding.

Series: unemploy.ts

ARIMA(1,0,0) with non-zero mean

Coefficients:

ar1	mean

0.9957 1597.190

s.e. 0.0037 1265.272

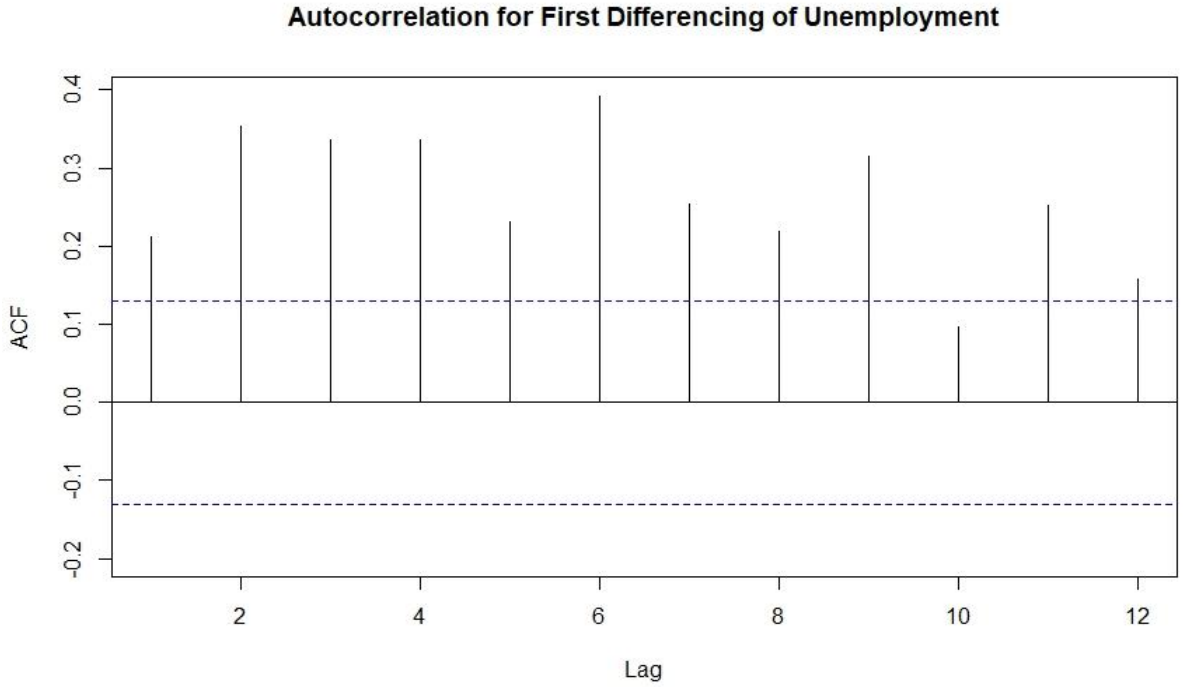
sigma^2 estimated as 19839: log likelihood=-1452.97

AIC=2911.94   AICc=2912.04   BIC=2922.23

Training set error measures:

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training set	6.51662	140.2309	100.689	0.06346213	4.496782	0.1458234	0.2137261

The coefficient of the ar1 (Yt-1) variable, 0.9957, is still below 1. Since it is very close to 1, we performed another test to see if it is predictable.

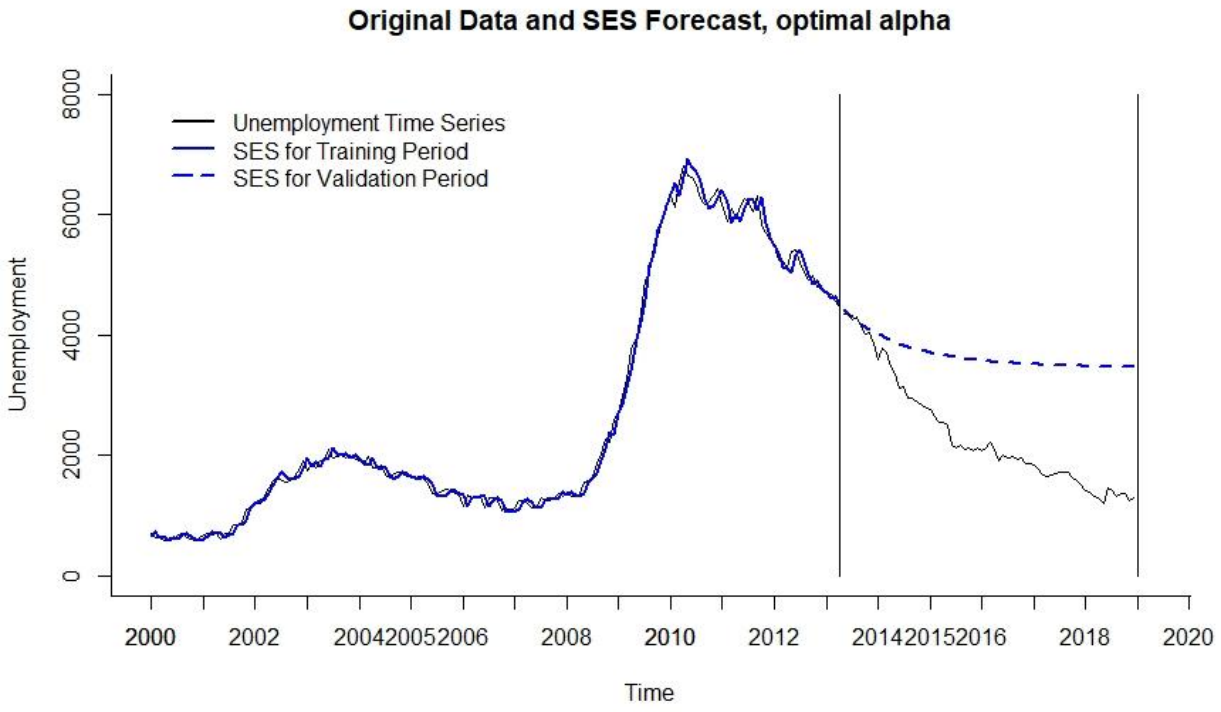


Based on the plot for first differencing above, again all autocorrelation coefficients of the lags are statistically significant. Therefore, using the first differencing, we can confirm that sales.ts is not a random walk and is predictable.

Model 1: Simple Exponential Smoothing (SES)

```
> summary(ses.opt)
ETS(A,Ad,N)

Call:
 ets(y = train.ts, model = "AAN")

  Smoothing parameters:
    alpha = 0.7646
    beta  = 0.2453
    phi   = 0.9354

  Initial states:
    l = 658.9687
    b = 6.0777

  sigma:  132.7941

     AIC     AICc      BIC
2383.364 2383.913 2401.815

Training set error measures:
                    ME     RMSE      MAE        MPE    MAPE      MASE          ACF1
Training set 3.938936 130.7027 97.31698 0.3337881 4.51468 0.1353619 9.588258e-05
```

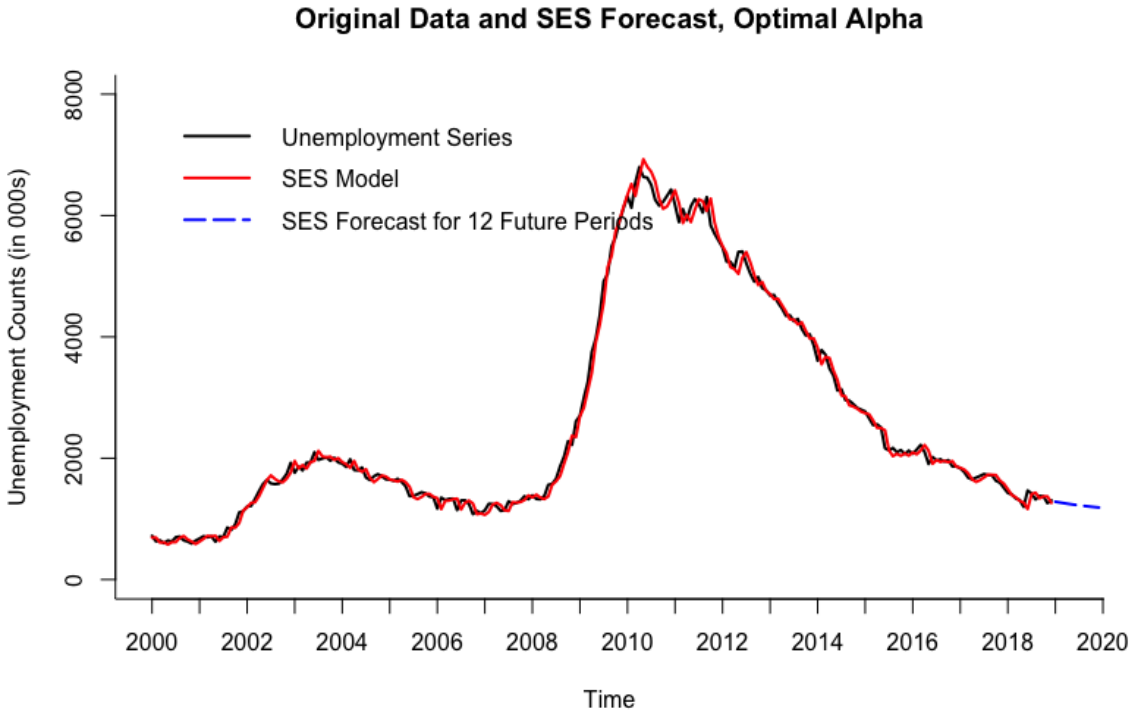
The first model we applied was the simple exponential smoothing model by using ets() function with specified model of 'AAN' with automated selection of smoothing constants. 'AAN' means additive error, additive trend and no seasonality. As a result, a SES model generated by the function had the optimal value for exponential smoothing constant (alpha) of 0.7646, smoothing constant for trend estimate (beta) of 0.2453 and phi of 0.9354. The alpha value of this model indicates that that the model's level component tends to be more local adjusted. The model's trend is globally adjusted as beta is relatively small. The plot of how the model fits with the original data is shown below.



As we can see from above, the model fits very well with training data, with minimal difference while it doesn't fit well with the validation data. One reason may be the overfitting of the data because it fits too closely to the trend of the training data and thus fails to predict coming records. Overfitting problem is further proven by studying the statistical measures such as MAPE and RMSE of both training and validation data.

```
> round(accuracy(ses.opt.pred, valid.ts), 3)
                    ME     RMSE      MAE     MPE   MAPE  MASE  ACF1 Theil's U
Training set     3.939  130.703   97.317   0.334  4.515 0.135 0.000        NA
Test set     -1322.303 1485.628 1323.897 -74.918 74.955 1.841 0.953    18.202
```

The MAPE and RMSE values of the validation data (test set) are way higher than the training data so overfitting is definitely a concern for this model. After that, the SES model is also applied for the entire dataset and to make prediction for the next 12 months period in 2019 as shown below.

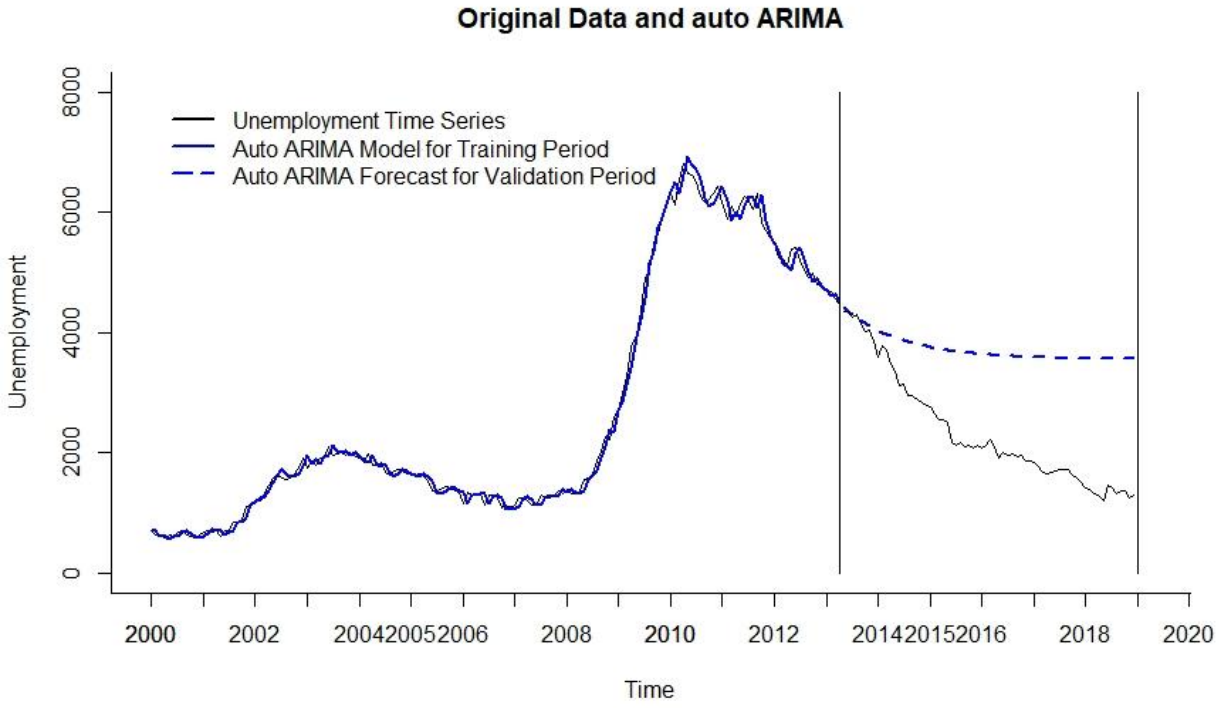


Model 2: Auto ARIMA

```
> summary(train.auto.arima)
Series: train.ts
ARIMA(1,1,2)

Coefficients:
          ar1      ma1     ma2
      0.9298  -0.9340  0.2161
s.e.  0.0376   0.0874  0.0743

sigma^2 estimated as 17497:  log likelihood=-1001.14
AIC=2010.29   AICc=2010.55   BIC=2022.56

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE          ACF1
Training set 4.728052 130.612 96.96572 0.4079428 4.476007 0.1348733 -0.0003976125
```

The second model we utilized was the auto ARIMA model to see if we could have a better forecast with no overfitting problem. The generated ARIMA model had three parameters which was used to forecast data with level and trend components, no seasonality. This ARIMA model has an AR component with order 1 (p=1) for trend, order 1 differencing (d=1) to remove linear trend and order 2 moving average (q=2) for error lags. It has a low MAPE value of 4.47 and a

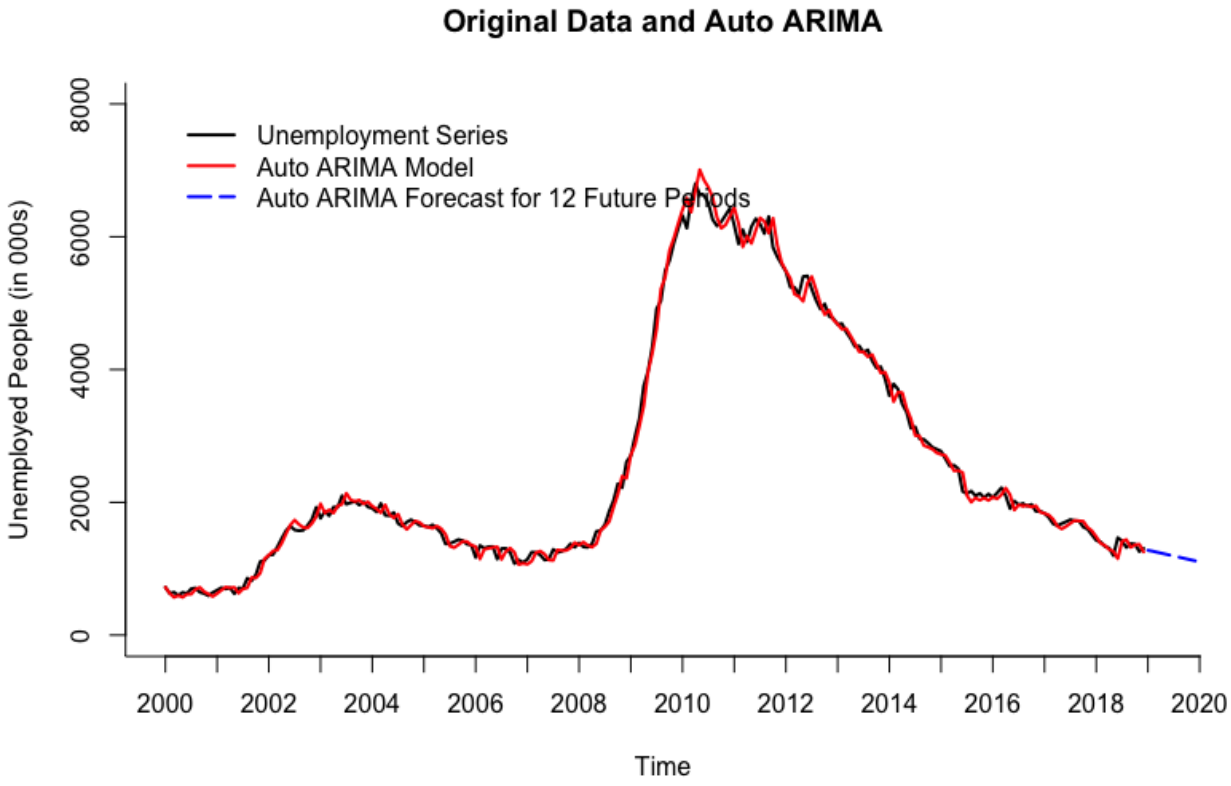
very small ACF1 value which mean the autocorrelation is integrated into the model. The plot of how the model compares with the original data is shown below.



As we can see from above, again the model fits very well with training data, with minimal to no difference to the original set. On the other hand, it doesn't capture the validation data very well which means the forecasting is not very accurate. Overfitting of the data may be the explanation for it because it fits too closely to the trend and the pattern of the training data and thus fails to predict coming records as well. We can also see the that the plot of auto ARIMA model and SES model are very identical to each other which indicates that both models have similar issues. Overfitting problem is further proven by studying the statistical measures such as MAPE and RMSE of both training and validation data.

```
> round(accuracy(train.auto.arima.pred, valid.ts), 3)
                    ME      RMSE       MAE      MPE   MAPE  MASE  ACF1 Theil's U
Training set     4.728   130.612    96.966    0.408  4.476 0.135 0.000        NA
Test set     -1380.793 1550.226 1382.190 -78.189 78.221 1.923 0.953    18.984
```

The MAPE and RMSE values of the validation data (test set) are way higher than the training data so overfitting is definitely a concern for this model. After that, the auto ARIMA model is also applied for the entire dataset and to make prediction for the next 12 months period in 2019 as shown below.



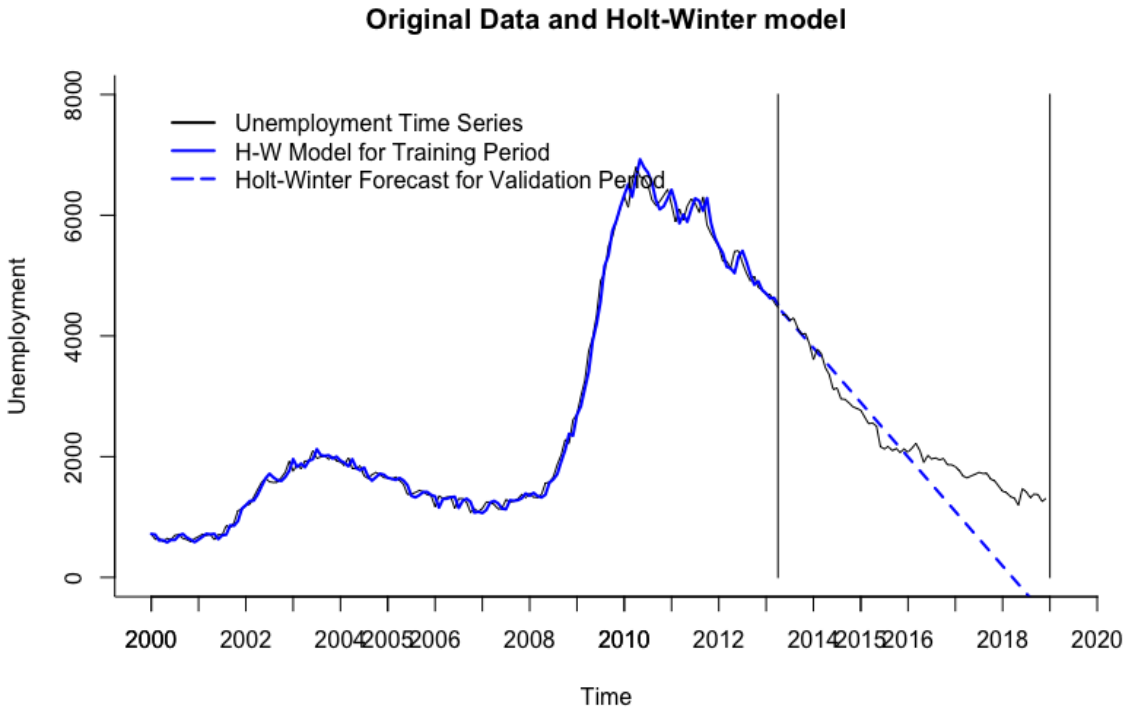
Model 3 : Holt-Winter

We used the ets() function to develop a Holt-Winter's model with automated selection of error, trend, and seasonality options, and automated selection of smoothing parameters for the training partition. As a result, a HW model automatically generated by the function had the (M, A, N) options, i.e., multiplicative error, additive trend, and no seasonality.

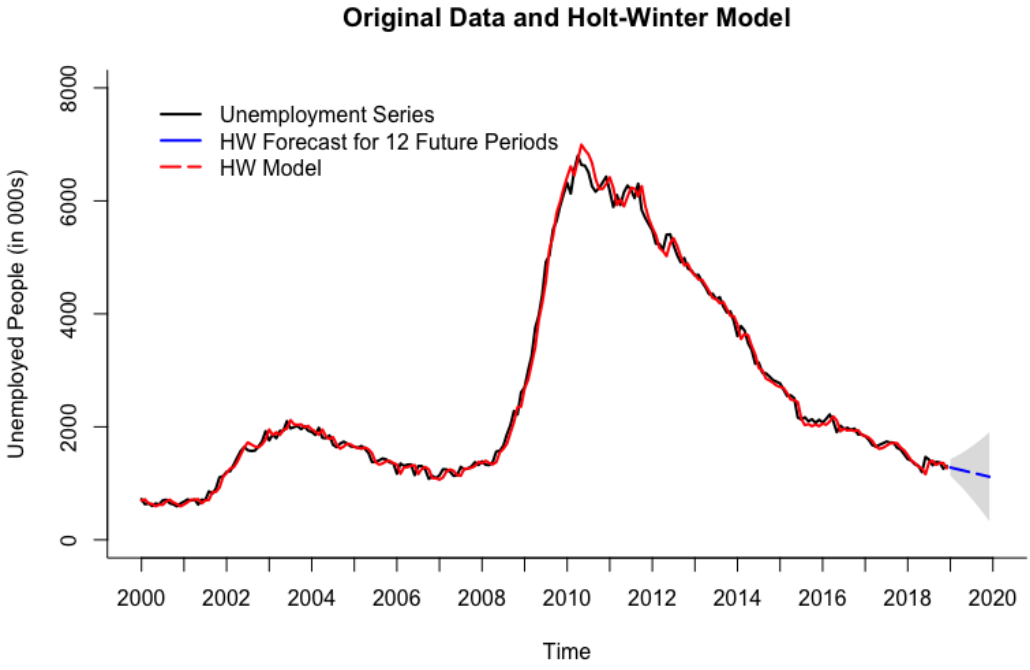
```
ETS(M,A,N)

Call:
 ets(y = unemploy.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.7245
    beta  = 0.1443

  Initial states:
    l = 694.3273
    b = -1.3823

  sigma:  0.058

     AIC     AICc      BIC
3431.177 3431.447 3448.323
```

The optimal value for exponential smoothing constant (alpha) is 0.7245, smoothing constant for trend estimate (beta) is 0.1443, and no smoothing constant for seasonality. The alpha value of this model indicates that that the model's level component tends to be more local adjusted. The model's trend is globally adjusted as beta is close to zero.

Again, below is a graph that we plotted to see how the model is doing for both training set and validation set. The model fits very well with training data while it doesn't fit the validation data very well.



After that, the Holt-Winter's model is also applied for the entire dataset and to make prediction for the next 12 months period in 2019, which shows a downward trend.



Model Comparison

```
> round(accuracy(ses.all.pred$fitted, unemploy.ts), 3)
           ME    RMSE    MAE   MPE  MAPE  ACF1 Theil's U
Test set 0.9 121.731 89.694 0.226 4.122 0.007     0.934
> round(accuracy(auto.arima.pred$fitted, unemploy.ts), 3)
            ME    RMSE    MAE   MPE  MAPE   ACF1 Theil's U
Test set 0.619 123.555 90.536 0.343 4.122 -0.012     0.942
> round(accuracy(HW.ZZZ.pred$fitted, unemploy.ts), 3)
             ME    RMSE   MAE   MPE  MAPE  ACF1 Theil's U
Test set -0.411 125.431 91.73 0.287 4.156 0.151     0.929
> round(accuracy((naive(unemploy.ts))$fitted, unemploy.ts), 3)
            ME    RMSE     MAE   MPE  MAPE  ACF1 Theil's U
Test set 2.577 140.512 101.529 0.081 4.493 0.211         1
```

From the model comparison, we can see a summarized table which contains MAPE and RMSE values for each model applied. We also compared them with naive model to see if our models have dramatically better results than the naive model. We can observe that from all models that were utilized on the data, the SES model has the lowest RMSE and MAPE values. The second best model is the auto ARIMA model which has the same MAPE value but slightly higher RMSE value. However, the 3 models that were utilized are not significantly better than the naive model. They only differ by a small margin so naive model is a good forecast method for this dataset as well.

Conclusion

After exploring the data and testing various models, we have concluded that SES model has the best performance upon predicting monthly unemployment. However, we also discovered that the high accuracy of this model may be due to the overfitting problem.

Below is the forecast for unemployment of 2019 using Simple Exponential Smoothing method, which shows that it is going to keep going downward.

	Point Forecast	Lo 0	Hi 0
Jan 2019	1283.281	1283.281	1283.281
Feb 2019	1271.634	1271.634	1271.634
Mar 2019	1260.693	1260.693	1260.693

Apr 2019 1250.416 1250.416 1250.416

May 2019 1240.762 1240.762 1240.762

Jun 2019 1231.693 1231.693 1231.693

Jul 2019 1223.175 1223.175 1223.175

Aug 2019 1215.173 1215.173 1215.173

Sep 2019 1207.656 1207.656 1207.656

Oct 2019 1200.596 1200.596 1200.596

Nov 2019 1193.963 1193.963 1193.963

Dec 2019 1187.733 1187.733 1187.733

This study is able to predict the trend for the long-term unemployment group which keeps going downward for 2019. However, after we test the accuracy of three different forecasting methods and comparing them with naive forecasting method, we didn't find the three models are way better than naive forecast. Even though predictability tests above shows the time series data is not a random walk, we still received the result that forecasts poorly, which is probably because it has an abnormal trend, where at some points it increases rapidly and at some other points it decreases quickly. Therefore, even if it is not a random walk, it has poor predictability level and thus very challenging to make accurate forecasting. This is the main limitation of our SES model.

In order to improve the prediction for unemployment, other factors need to be incorporated as well such as the US economy condition, number of established companies and so forth. The influence of world wide economy, trading policies and federal regulations may also be taken into consideration to have a better forecasting. This will require more complex forecasting methods and algorithms that are beyond the course of this class.

Bibliography

Wikipedia https://en.wikipedia.org/wiki/Great_Recession_in_the_United_States

Appendices

Dataset and Data Collection

Source: https://fred.stlouisfed.org/series/UEMP27OV

Download: https://github.com/anguyen152-stat660/TimeSeries/blob/master/UEMP27OV_2000to2018.xls