**Saving Layoffers by Data Science**

**by Madison Myers, Arvin Sahni and Shih Yu Chang**

**Introduction:**

Layoffs are a very real fear for millions across the globe, particularly during an

economic crisis. Unemployment rates soar, and applying to new jobs gets more and

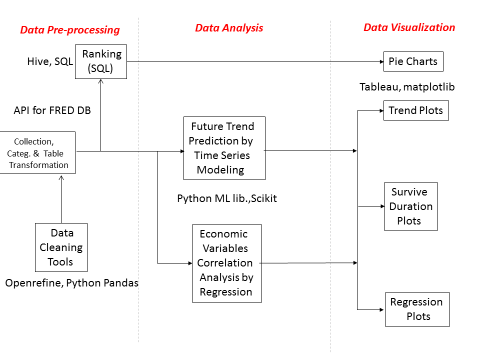
more competitive, leaving many desperate and unable to cover living costs. This is

why we have decided to use data to analyze the connection of factors related to the

economy, e.g., location and industry specifically the duration after layoff that an individual will be able to keep a job given the set of circumstances mentioned above. We also study the relationship between minimum wage and unemployment rate at different economic conditions, before regression, regression and after regression. This economic relationship is studied based on United States data from 2002 to 2013. From the multiple regression analysis results, we conclude that minimum wage has a significant effect on unemployment when the economy was during the recession phase.

**System Architecture:**

The task to predict survival duration can be partitioned as three phases. The first is data pre-processing phase. The main data source is Federal Reserve Economic Data - FRED - St. Louis Fed. After data cleaning by removing non-sense values (using python data clean modules, e.g. pandas) at data tables, we then apply Hadoop, Hive’s SQL interface, and Hive SQL to get high level understanding of economic status by ranking some economic indices. The next phase is to build model to predict survival duration of layoffed employees. The other analysis is to explore the relationship between economic indices and utilize analysis results to verify government policies. The last phase is data visualization. Based on data pre-processing and data analysis results, we will present figures dashboards to illustrate our concolusion , e.g., how long to survive after layoff. We feel that this prediction could help employees and employers more aware of the current or coming economic climate and respective job market. The system diagram is presented at following page.



**Data Source and Acquisition**

FRED is a plentiful database of U.S. and international economic data maintained by the research department of the Federal Reserve Bank of St. Louis. The database is regularly updated as latest data become available. All data provided by this database are presented by time-series format and all data can be downloaded as .csv, excel, pdf, and figure files. Since there are 390,000 time-series from various economical indices contributed by 79 data sources, the data size is about 39 GB. Moreover, this database will update day-by-day. Hence, this FRED database has 3V properties as other Big Data. The website for FRED is: <http://research.stlouisfed.org/fred2/>.

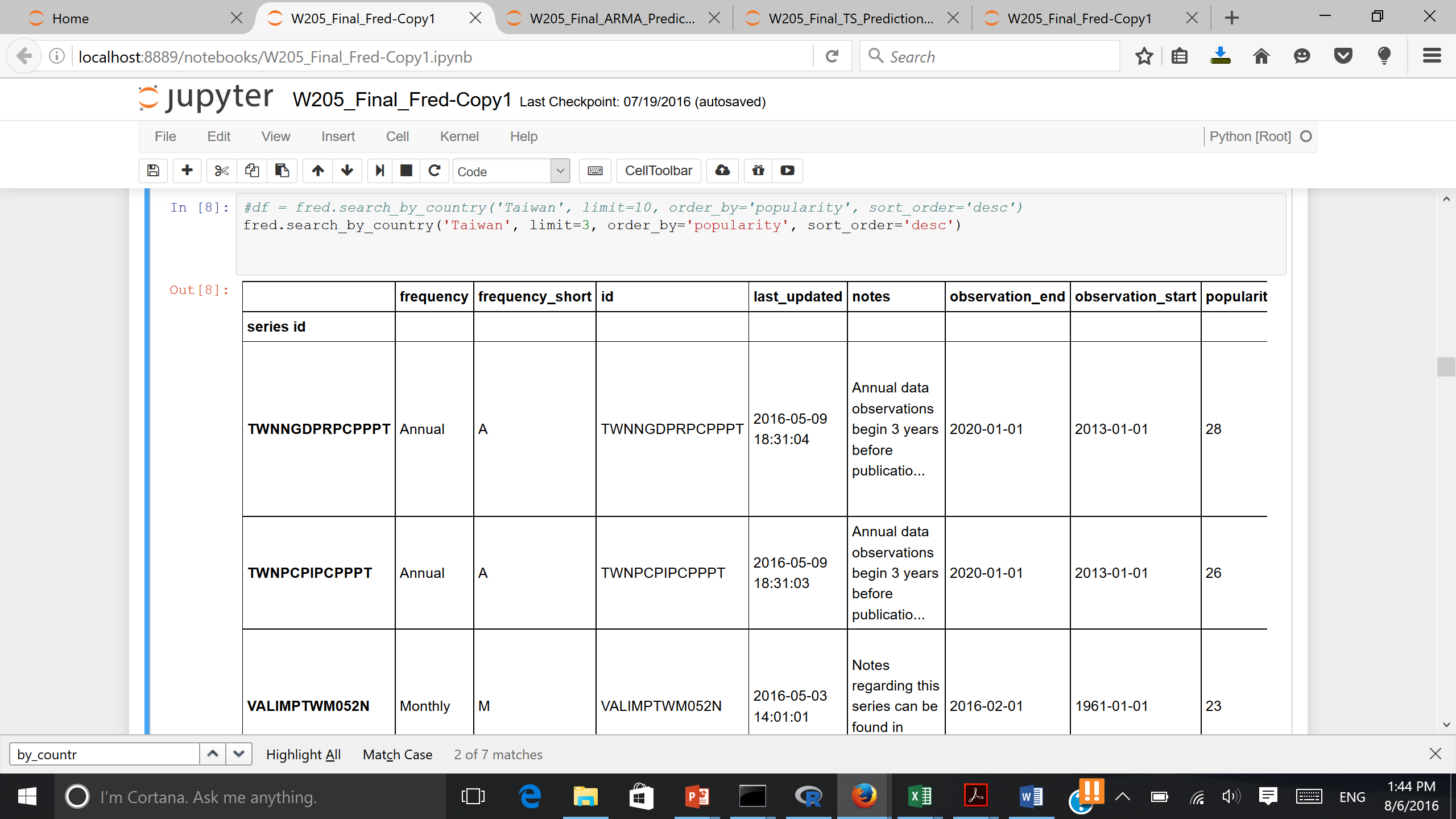
The FRED website provide access KEY (need registration) for developers to write code to acquire data from cloud. We develop python API to access data from FRED website. Different current available APIs, we implement API by building new functions which are capable of search by locations, e.g., states, countries. We provide following codes to demo getting series from FRED and plot it. More details codes can be reviewed at W205\_Final\_Fred-Copy1.ipynb.



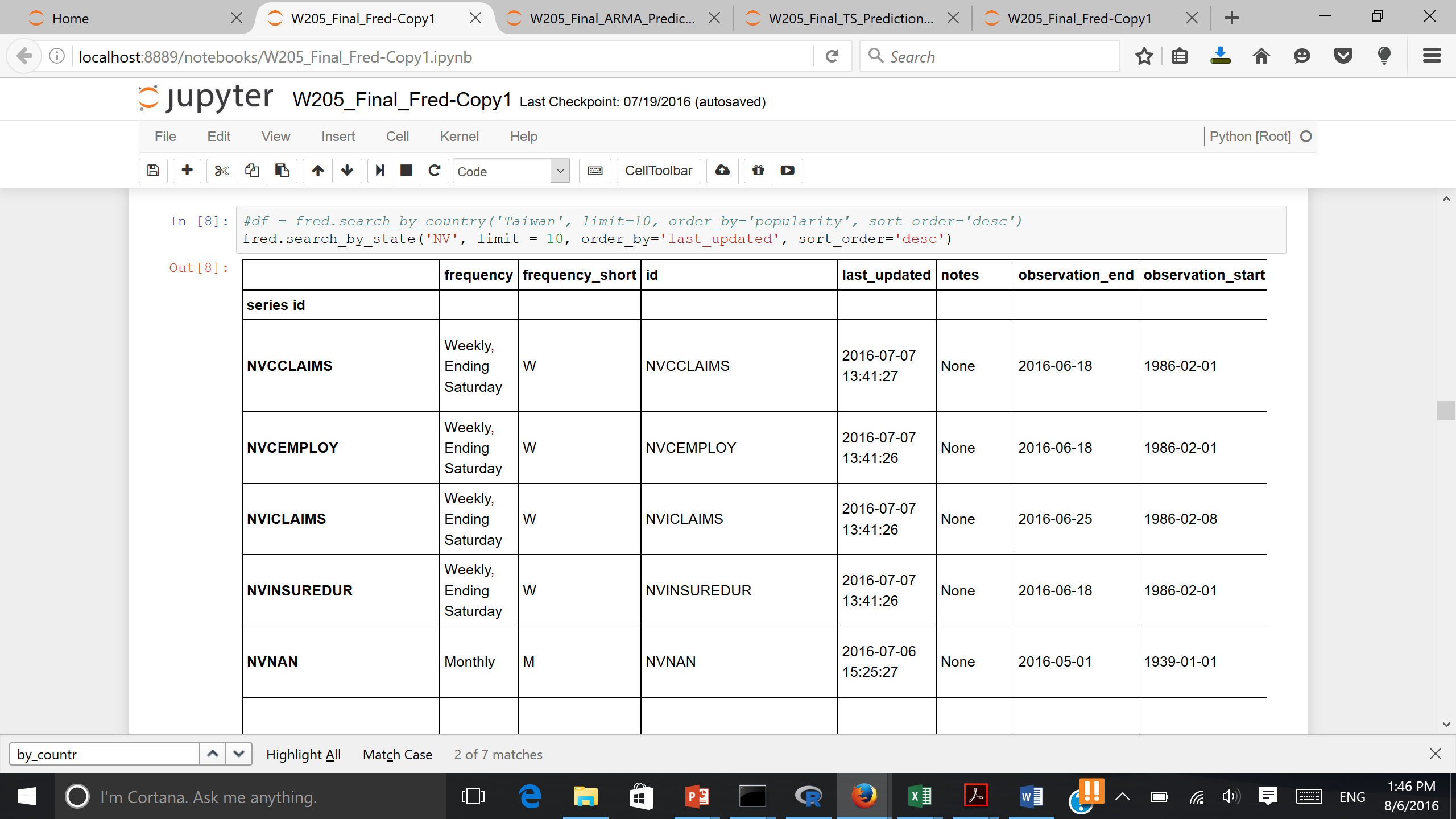


We also implemented FRED API to have geographic query inputs, such as countries and states of USA.

For countries (Taiwan):



For states (Nevada) :



**ARIMA Prediction Model**

In statistics and econometrics, and specially in time series analysis, an autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. These models are fitted to time series data either to better understand the data or to forecast future points in the series (predicting). They are applied in prediction contexts where data show evidence of non-stationarity, where an initial differencing step can be applied to reduce the non-stationarity.

By applying ARIMA prediction mechanism, we can get range of survival time for each countries and states based on each individual ALL assets values at lay off time, say X and living cost of a particular area after some period of time, say 1 years. Then, survival time duration range can be determined as MIN: X / (MAX living cost in prediction), and MAX: X/(MIN living cost in prediction). For example, following code shows that living in Los Angeles, after layoff, the maximum duration is 4.29937748343 months and the minimum duration is 4.06629570388 months, where FRED time-series is CUURA421SEHF02. Following codes implemented our approach.

### Given P, Q, predictions\_ARIMA by residual ### CUURA422SA0, CUURA210SA0L2 (monthly)

series = fred.get\_series('CUURA421SEHF02')

series = series[np.logical\_not(np.isnan(series))]

#res = sm.tsa.ARMA(series, (3, 4)).fit()

#res\_error = sum(res.resid.values\*\*2)

#print(res\_error)

def pred\_ARIMA(p, q, series):

ts\_log = np.log(series)

model = ARIMA(ts\_log, order=(p, 1, q))

results\_ARIMA = model.fit(disp = -1, method = 'css')

predictions\_ARIMA\_diff = pd.Series(results\_ARIMA.fittedvalues, copy=True)

predictions\_ARIMA\_diff\_cumsum = predictions\_ARIMA\_diff.cumsum()

predictions\_ARIMA\_log = pd.Series(ts\_log.ix[0], index=ts\_log.index)

predictions\_ARIMA\_log = predictions\_ARIMA\_log.add(predictions\_ARIMA\_diff\_cumsum,fill\_value=0)

predictions\_ARIMA = np.exp(predictions\_ARIMA\_log)

return predictions\_ARIMA

UpperLimit = 4

p = range(0, UpperLimit + 1)

q = range(0, UpperLimit + 1)

## perform model order optimization##

def opt\_paras(series):

res = sm.tsa.ARMA(series, (0, 0)).fit()

res\_error = sum(res.resid.values\*\*2)

opt\_p = 0

opt\_q = 0

for i in range(0, UpperLimit):

for j in range(0, UpperLimit):

#if i==0 and j==0:

# continue

res = sm.tsa.ARMA(series, (p[i], q[j])).fit()

new\_res\_error = sum(res.resid.values\*\*2)

if (new\_res\_error < res\_error):

res\_error = new\_res\_error

opt\_p = p[i]

opt\_q = q[j]

series\_pred = pred\_ARIMA(opt\_p, opt\_q, series)

return opt\_p, opt\_q, mean\_squared\_error(series, series\_pred)

result = opt\_paras(series)

print(result)

plt.plot(series)

plt.plot(pred\_ARIMA(result[0], result[1], series), color='green')

# get what you need for predicting future ahead

res = sm.tsa.ARMA(series, order=(result[0], result[1])).fit()

params = res.params

residuals = res.resid

p = res.k\_ar

q = res.k\_ma

k\_exog = res.k\_exog

k\_trend = res.k\_trend

steps = 12

CPI = \_arma\_predict\_out\_of\_sample(params, steps, residuals, p, q, k\_trend, k\_exog, endog=series, exog=None, start=len(series))

print(CPI)

#### Calculate survival period ####

# initial assest of an employee get layoff

initial\_assets = 20000

#CPI = [300, 400, 500, 600]

# Housing price moneth pay 1982 as basis

CA\_month = 750

print("Living in Los Angeles, the maximum duration is ", (initial\_assets/(CA\_month \* 3 \* min(CPI)/100)), "months; "

"the minimum duration is ", (initial\_assets/(CA\_month \* 3 \* max(CPI)/100)), "months." )

**Exponential Smoothing (ES) Prediction Model**

Exponential smoothing is another method to perform prediction. A basic technique for smoothing time series data, particularly for recursively applying as many as three [low-pass filters](https://en.wikipedia.org/wiki/Low-pass_filter) with exponential window functions. Such techniques have been applied in various domain that is not intended to be strictly accurate or reliable for every situation. It is an easily calculation procedure for approximately calculating or recalling some value, or for making some determination based on prior assumptions from historical observation, such as seasonality. Like any application of repeated low-pass filtering, the observed phenomenon may be an essentially [random process](https://en.wikipedia.org/wiki/Stochastic_process), or it may be an orderly process with noise. The most basic ES model is the [simple moving average](https://en.wikipedia.org/wiki/Simple_moving_average) that the past observations are weighted equally. Exponential smoothing assigns exponentially decreasing weights as the observation get older. In other words, recent observations are given relatively more weight in forecasting than the older observations.

We implemented ES as following:

### Implementing Exponential Smoothing###

### Test series

y = fred.get\_series('CUURA311SEHA')

series = y.values

plt.plot(series)

def ini\_trend(series, slen):

sum = 0.0

for i in range(slen):

sum += float(series[i+slen] - series[i]) / slen

return sum / slen

def ini\_seasonal\_components(series, slen):

seasonals = {}

season\_averages = []

n\_seasons = len(series)//slen

# compute season averages

for j in range(n\_seasons):

season\_averages.append(sum(series[slen\*j:slen\*j+slen])/float(slen))

# compute initial values

for i in range(slen):

sum\_of\_vals\_over\_avg = 0.0

for j in range(n\_seasons):

sum\_of\_vals\_over\_avg += series[slen\*j+i]-season\_averages[j]

seasonals[i] = sum\_of\_vals\_over\_avg/n\_seasons

return seasonals

def Exponential\_Smoothing(series, slen, alpha, beta, gamma, n\_preds):

result = []

seasonals = ini\_seasonal\_components(series, slen)

for i in range(len(series)+n\_preds):

if i == 0: # initial values

smooth = series[0]

trend = ini\_trend(series, slen)

result.append(series[0])

continue

if i >= len(series): # we are forecasting

m = len(series) - i + 1

result.append((smooth + m\*trend) + seasonals[i%slen])

else:

val = series[i]

# Additive ES

last\_smooth, smooth = smooth, alpha\*(val-seasonals[i%slen]) + (1-alpha)\*(smooth+trend)

trend = beta \* (smooth-last\_smooth) + (1-beta)\*trend

seasonals[i%slen] = gamma\*(val-smooth) + (1-gamma)\*seasonals[i%slen]

result.append(smooth+trend+seasonals[i%slen])

return result

future\_windows = 36

series\_pred = Exponential\_Smoothing(series, 12, opt\_results[0], opt\_results[1], opt\_results[2], future\_windows)

#plt.plot(series\_pred)

#print(mean\_squared\_error(series, series\_pred))

print(series\_pred[len(series) + 1])

print(series\_pred[len(series) + future\_windows - 1])

Because ES prediction errors are sensitive for model weights selections, we also provide following codes to get optimal ES prediction model by minimizing mean-squared-error of past historical data.

### Optimizing Exponential Smoothing Parameters ###

series = y.values

#print(mean\_squared\_error(series, series\_pred))

# alpha, beta, gamma

resolution = 20

alpha = np.linspace(0.0, 1.0, resolution)

beta = np.linspace(0.0, 1.0, resolution)

gamma = np.linspace(0.0, 1.0, resolution)

def opt\_paras(series, slen):

Min\_MSE = mean\_squared\_error(series, Exponential\_Smoothing(series, slen, 0.001, 0.001, 0.001, 0))

opt\_alpha = 0.001

opt\_beta = 0.001

opt\_gamma = 0.001

for i in range(0, resolution):

for j in range(0, resolution):

for k in range(0, resolution):

new\_MSE = mean\_squared\_error(series, Exponential\_Smoothing(series, slen, alpha[i], beta[j], gamma[k], 0))

if (new\_MSE < Min\_MSE):

Min\_MSE = new\_MSE

opt\_alpha = alpha[i]

opt\_beta = beta[j]

opt\_gamma = gamma[k]

series\_pred = Exponential\_Smoothing(series, slen, opt\_alpha, opt\_beta, opt\_gamma, 0)

return opt\_alpha, opt\_beta, opt\_gamma, mean\_squared\_error(series, series\_pred)

opt\_results = opt\_paras(series, 12)

print(opt\_results)

future\_windows = 36

series\_pred = Exponential\_Smoothing(series, 12, opt\_results[0], opt\_results[1], opt\_results[2], future\_windows)

#plt.plot(series\_pred)

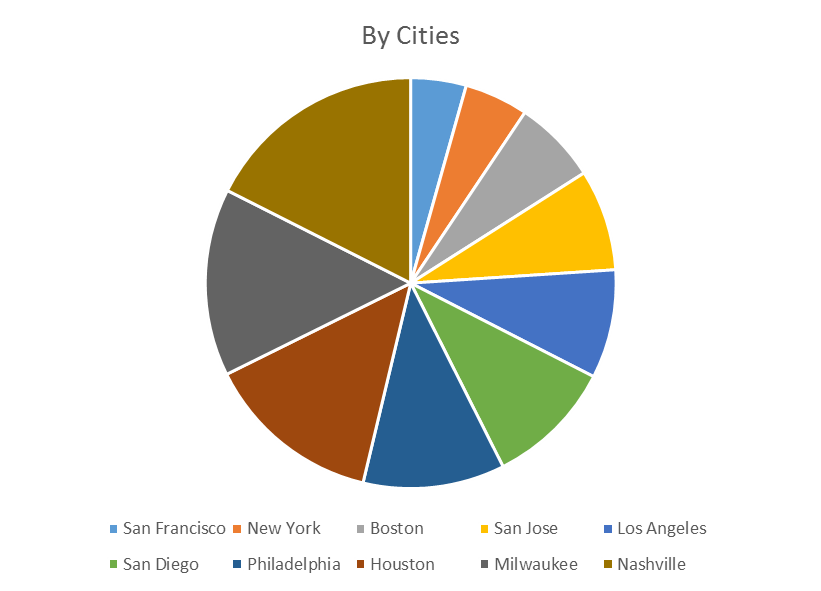
#print(mean\_squared\_error(series, series\_pred))

print(series\_pred[len(series) + 1])

print(series\_pred[len(series) + future\_windows - 1])

More details codes can be reviewed at W205\_Final\_ARMA\_Prediction-Copy1.ipynb and W205\_Final\_TS\_Prediction-Copy1.ipynb

Following pie-chart shows the proportion of time can survive after layoff with respect to different cities.



Following pie-chart shows the proportion of time can survive after layoff with respect to different countries.

Following bar-chart shows the survival months after layoff by cities predicted by ARIMA model.

Following bar-chart shows the survival months after layoff by cities predicted by exponential smooth model.

Following bar-chart shows the survival months after layoff by states predicted by ARIMA model if initial individual asset value is US 20000.

Following bar-chart shows the survival months after layoff by states predicted by ES model given initial individual asset value is US 20000..

Following bar-chart shows the survival months after layoff by countries predicted by ARIMA model.

Following bar-chart shows the survival months after layoff by countries predicted by ES model.

**The Effect of Minimum Wage to Unemployment**

This part aims to model and quantify the relationship between minimum wage and

unemployment rate. This economic relationship is studied based on basis in the United States from 2002 to 2013 from FRED. This time period is broken up into three phases: before-recession (2002 Jan. to 2006 Dec.), recession (2007 Jan. to 2009 Dec.), and after-recession (2010 Jan. to 2013 Dec.). For the multiple regression models, it was concluded that minimum wage had a significant effect on unemployment when the economy was during the recession phase. R code (LinearRegresUnemployment.R) implemented our study about the effect of minimum wage to unemployment, and education level vs minimum wage relationship.

**Multiple Regression Model**

The multiple regressions for each time period are defined below as following equation.

𝑢𝑛𝑒𝑚𝑝𝑙𝑜𝑦𝑚𝑒𝑛𝑡 rate = 𝛽0 + 𝛽1 𝑚𝑖𝑛𝑊𝑎𝑔𝑒 + 𝛽2 𝐸𝑑𝑢𝐿𝑒𝑣𝑒𝑙 + 𝛽3 𝐺𝐷𝑃

The multiple regression model for the before-recession and after-recession showed an insignificance for the minimum wage and unemployment. It also can be seen from the table that the coefficients for GDP per capita are very small and seem almost insignificant and yet that is not exactly the case for GDP per capita; consider the fact that GDP per capita is in tens of thousands of dollars. As a result, even an extremely small coefficient, 𝛽3, can still have some effect on the unemployment level. The percentage of high school graduates percentage, education level, is shown to be significant at every phases.

For the recession model, minimum wage was significant at the 15% level and education level once again was significant at all levels. GDP per capita was not significant during this period of time. The minimum wage increased significance for the recession model as compared to the pre-recession model can be explained by the booming of the economy. During the recession, the unemployment rate greatly increased which can be attributed to the poor economic structure at the time. This affected on the model and made it seem like the increase in unemployment rate was correlated to minimum wage. To assess whether this model is appropriate the F-test was completed again, resulting in joint significance at the low level. Consequently, this model is adequate and minimum wage is critical. However, the recession could have also made the significance between unemployment rate and minimum wage more apparent. When firms are faced with an economic hardship, the higher minimum wages makes it harder to recruit labor. Economic reasoning would be used to explain this causation as opposed to recession causing bias in the model. Table below is about multiple regression model results summarized from following R-code summary at each economic phase.

**Table : Multiple Regression Model Results**

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

|  |  |  |  |
| --- | --- | --- | --- |
| Ind. Variables | Before-Recession | Recession | After-Recession |
| Min\_Wage | NA | 0.164 | NA |
| Hgih\_School\_Percentage | 0.647 \*\*\* | 0.815 \*\*\* | 0.442 \*\*\* |
| G\_D\_P | ~0 \*\*\* | ~0 | ~0 \*\*\* |
| Intercept | 4.857 \*\*\* | 2.496 | 13.872 \*\*\* |
| R-square | 0.908 | 0.992 | 0.956 |

before-regression (Jan. 2002 to Dec. 2006):

lm(formula = pre\_Unemployment\_Rate ~ pre\_Min\_Wage + pre\_Hgih\_School\_Percentage +

pre\_G\_D\_P, data = df)

Residuals:

Min 1Q Median 3Q Max

-0.29278 -0.14961 0.00555 0.09804 0.32181

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 4.857e+00 9.576e-01 5.071 4.48e-06 \*\*\*

pre\_Min\_Wage NA NA NA NA

pre\_Hgih\_School\_Percentage 6.473e-01 9.729e-02 6.653 1.20e-08 \*\*\*

pre\_G\_D\_P -2.149e-04 4.119e-05 -5.218 2.63e-06 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1611 on 57 degrees of freedom

Multiple R-squared: 0.9079, Adjusted R-squared: 0.9046

F-statistic: 280.9 on 2 and 57 DF, p-value: < 2.2e-16

Within-regression (Jan. 2007 to Dec. 2009):

lm(formula = inn\_Unemployment\_Rate ~ inn\_Min\_Wage + inn\_Hgih\_School\_Percentage +

inn\_G\_D\_P, data = df)

Residuals:

Min 1Q Median 3Q Max

-0.3452 -0.1519 -0.0241 0.1265 0.3770

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 2.4963806 4.6388070 0.538 0.594

inn\_Min\_Wage 0.1638688 0.1264321 1.296 0.204

inn\_Hgih\_School\_Percentage 0.8147887 0.0388150 20.992 <2e-16 \*\*\*

inn\_G\_D\_P -0.0001597 0.0003353 -0.476 0.637

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.197 on 32 degrees of freedom

Multiple R-squared: 0.9919, Adjusted R-squared: 0.9912

F-statistic: 1309 on 3 and 32 DF, p-value: < 2.2e-16

After-regression (Jan. 2010 to Dec. 2013):

lm(formula = aft\_Unemployment\_Rate ~ aft\_Min\_Wage + aft\_Hgih\_School\_Percentage +

aft\_G\_D\_P, data = df)

Residuals:

Min 1Q Median 3Q Max

-0.47352 -0.09843 -0.00910 0.13919 0.34008

Coefficients: (1 not defined because of singularities)

Estimate Std. Error t value Pr(>|t|)

(Intercept) 13.8722351 2.8354133 4.892 1.31e-05 \*\*\*

aft\_Min\_Wage NA NA NA NA

aft\_Hgih\_School\_Percentage 0.4421829 0.0786007 5.626 1.12e-06 \*\*\*

aft\_G\_D\_P -0.0005878 0.0001368 -4.298 9.12e-05 \*\*\*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1913 on 45 degrees of freedom

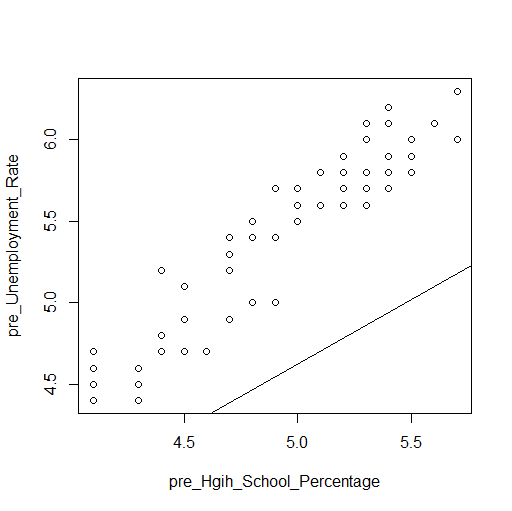
Multiple R-squared: 0.9557, Adjusted R-squared: 0.9537

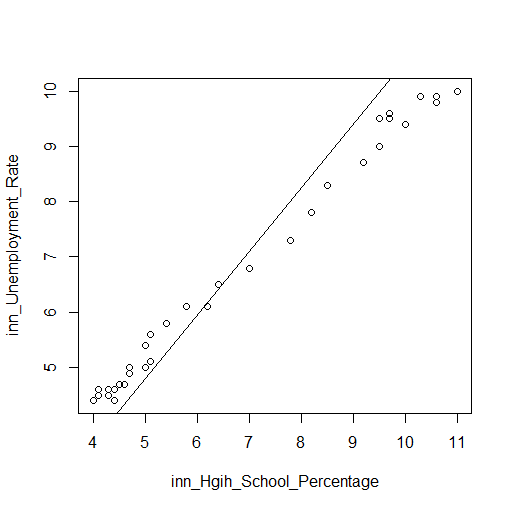
F-statistic: 484.8 on 2 and 45 DF, p-value: < 2.2e-16

**Education and Unemployment**

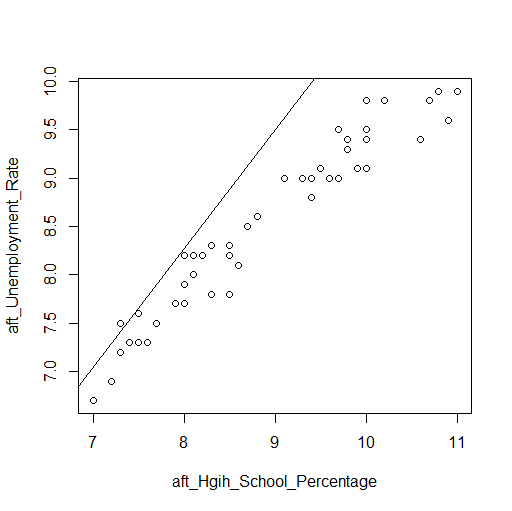
We generated education level and unemployment regression plot for three phases, before-regression, within-regression, and after-regression. According to following figures, there is a highly correlated relationship between education level (percentage of people get high school degree) and unemployment. But, surprisingly, we observed that higher percentage of high-school educated people in USA will have higher unemployment rate. This quite different from general sense.

before-regression (Jan. 2002 to Dec. 2006):



Within-regression (Jan. 2007 to Dec. 2009):

After-regression (Jan. 2010 to Dec. 2013):



**Conclusions:**

In this project, we made following contributions: (1) We developed API to get time-series from FRED website up-to-date. Our API allows us to get time-series data from geographical query inputs, e.g., states and countries. (2) In order to estimate survival duration of layoff, we adopted ARIMA and ES model to predict survival duration based on individual financial status and his/her local living expanses. (3) We also discovered fact that minimum wage has a significant effect on unemployment when the economy condition is at regression by multiple regression analysis. Moreover, we also observed that higher percentage of high-school educated people in USA will have higher unemployment rate. This quite different from general feeling.