Predicting Movements in the Quantity of Social Security Retirees

P. Christopher J. Daigle

7 December 2018

Table of Contents

[1. Topic 3](#_Toc532130183)

[2. Data 3](#_Toc532130184)

[3. Variables 4](#_Toc532130185)

[4. Summary Statistics 5](#_Toc532130186)

[5. Model / Methodology 5](#_Toc532130187)

[6. Concluding Remarks 7](#_Toc532130188)

[7. References 8](#_Toc532130189)

[A. Appendix: Figures 9](#_Toc532130190)

[B. Python Code 11](#_Toc532130191)

[C. R Markdown 12](#_Toc532130192)

1. **Topic**

“For the great majority of Americans, the most important form of household wealth is the anticipated social security retirement benefits.” (Feldstein, 1974). Predicting movements in social security may be beneficial in determining expectations of the funding mechanism.

Through the collection of data related to the US Financial System, I evaluate changes in the number of people receiving social security retirement benefits. Through utilizing different econometric and machine learning methodologies, I am able to successfully predict at different levels of accuracy if the number of social security recipients will rise or fall.

1. **Data**

The data is of a time-series nature covering a period from 1985 to 2018 with observations at each month. The level is national; regional social security information is not evaluated nor acknowledged. Financial values are inflated to the latest available CPI information, September 2018.

Collection was completed in a few manners. First, I developed a web-scraper in Python with the library BeautifulSoup that parsed information from Yahoo Finance on the Dow Jones Industrial Average and the S&P500. This web-scraper was also applied to the Federal Reserve Bank of St. Louis website for CPI and the Federal Funds Rate. The web-scraper I built did not yield consistent outputs (e.g. two runs of the web-scraper yielded two different results or crashed altogether), so I chose to manually download these from the site.

Utilizing Python with the NumPy and Pandas library along with some SQL syntax, I imported, transformed, and combined the data sets to export a single dataset. Transformations were necessary as the joining key, date, was not standard from each source. At this point, the data set consisted of 405 observations and 22 variables (T = 405, P = 22).

From here, I cleaned the data in R and created a number of synthetic variables ranging from an inflation factor, real dollar values from nominal dollar values, difference values, percent change values, and indicators of positive changes in these values. Variables that were created as the result of differences in their values contained null values for the first time period. Because of this, the data lost the first time period to account for null values (i.e. I dropped the first time period because it was full of NAs). These transformations and variable creations resulted in 100 new variables and the loss of one time period (T = 404, P = 122).

The data began with 405 observations and reduced to 404 relevant observations because of the type of response, changes. The time dimension is still relatively long and so the loss of this one observation appears trivial.

As this data was of a time-series nature, it was necessary to ensure only stationary values were utilized for prediction. I observed different measures of seasonal decomposition and I applied the Augmented Dickey-Fuller Test which tests for stationarity. From that test, I removed all variables that did not appear stationary. This resulted in a shrinking of the data set by 64 variables (T = 404, P = 58).

1. **Variables**

The response I measured is a directional change in the quantity of social security recipients as increasing or decreasing. For this, I begin with a binary variable with the positive class (i.e. y = 1) indicating a positive difference in the quantity of recipients from one month to the next and the negative class (i.e. y = 0) indicating a negative movement in the quantity of recipients from one month to the next. From this, I used a series of stepwise sub-setting to select appropriate factors for prediction. I found forward selection to yield the best predictors. These predictors are as follows:

|  |  |
| --- | --- |
| Pos∆DJIopen | pos∆RealSPadjClose |
| Pos∆DJIclose | %∆RealDJIadjClose |
| pos∆DJIadjClose | %∆RealSPadjClose |
| Pos∆RealSPopen | pos∆RealAverageFemaleSSRetiredPay |
| %∆RealDJIhigh | pos∆AverageFemalSSRetiredPay |

This is a total of ten factors from the 58 I found by ensuring the data was stationary. It is worth noting that many of these variables are closely related so much so that a few are transformations of one another. While this is not a good method for establishing a causal model, my goal in this examination is to evaluate predictors and not perform causal inference.

1. **Summary Statistics**

Contained is a table of summary statistics of relevant variables:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **pos∆DJIopen** | **pos∆DJIclose** | **pos∆DJIadjClose** | **pos∆RealSPopen** | **pos∆averageFemaleSSRetiredPay** | **pos∆RealaverageFemaleSSRetiredPay** | **pos∆RealSPadjClose** | **%∆RealDJIhigh** | **%∆RealDJIadjClose** | **%∆RealSPadjClose** |
| Obs | 404 | 404 | 404 | 404 | 404 | 404 | 404 | 404 | 404 | 404 |
| Min | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | -0.241 | -0.234 | -0.220 |
| Max | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 0.117 | 0.132 | 0.126 |
| Med | 1.000 | 0.000 | 1.000 | 0.000 | 1.000 | 1.000 | 0.000 | 0.006 | 0.008 | 0.009 |
| Avg | 0.639 | 0.490 | 0.606 | 0.418 | 0.606 | 0.611 | 0.468 | 0.006 | 0.006 | 0.006 |
| Var | 0.231 | 0.251 | 0.239 | 0.244 | 0.239 | 0.238 | 0.250 | 0.001 | 0.002 | 0.002 |
| SD | 0.481 | 0.501 | 0.489 | 0.494 | 0.489 | 0.488 | 0.500 | 0.032 | 0.042 | 0.042 |

1. **Model / Methodology**

The basic model of analysis is a logistic regression where the influencing factors are selected via forward stepwise sub-setting where the factors for the model yield the smallest sum of squared residuals (RSS) (ISLR, 2017).

The initial model is as such:

This model, utilizing logistic regression, yields a prediction accuracy of about 93% when trained on about 70% of the data. From this model, I tested more models for prediction accuracy by removing factors at each step related to their statistical significance.

By that method, I was able to remove 7 factors. The restricted model then appeared as:

At the same split for the training and testing of the model as the unrestricted model, this model accurately predicts if the total number of social security recipients will rise at the 95% level and fall at the 91.5% level, with an overall model accuracy of 93%. More specifically, the original model of 10 predictors cut down to three predictors yield the same level of accuracy.

Notably, the logit model was selected for its ability in classification, but the probit model was also tested as the underlying distribution of the positive change in total social security recipients was about normal.

This model yielded the same prediction accuracy as the logit model on the same splits of the training and testing set.

For a more robust analysis of the model’s fit than simply the prediction outcomes, I tested the restricted model (3 factors) and unrestricted model (10 factors) with a likelihood-ratio test.

The findings are that the null hypothesis, i.e. the restricted model fits the data better than the unrestricted model, fails to be rejected. Thus, the restricted model of only three factors can be employed with at least as good of prediction power (i.e. the model with less factors is just as good as the one with more).

1. **Concluding Remarks**

Economics generally focuses on exogeneity. Discerning the causal effect of factors on the response variable is important for policy prescription. As data continues to be generated at a rapid rate, predicting the outcome of an event may become a simpler task. A tradeoff seems necessary between demanding causality and demanding accuracy of determining the likelihood of occurrence.

If the field of economics holds predictive models (forecasting) to the same standard as inferential models (causal relationships), the field as a whole may lose the ability to take advantage of quickly growing information and the field will limit its tools.

While directional changes in the quantity of social security retirees has been shown to be predictable from the data and models employed here, I make no claim to the causal relationship. With only a few and easily accessible pieces of data, it is shown that a signal can be assessed in determining the need to prepare for changes in social security by the policy maker. The power of this tool is not in its ability to explain why people retire, but to predict if people will.

As continued funding for social security is in question, determining whether the number of recipients will rise, or fall, can help to inform the transmission of funds from the system. Of value may also be to supplement this prediction with predicting the number of labor force participants as returns to labor are taxed to fund the social security system.

1. **References**

Feldstein, Martin (1974) Social Security, Induced Retirement, and Aggregate Capital Accumulation. *Journal of Political Economy, 82-5*.

FRED, Federal Reserve Bank of St. Louis (2018). *Consumer Price Index for All Urban Consumers: All Items,* https://fred.stlouisfed.org/series/CPIAUCSL/ accessed October 10, 2018.

FRED, Federal Reserve Bank of St. Louis (2018). *Effective Federal Funds Rate,* https://fred.stlouisfed.org/series/FEDFUNDS accessed October 10, 2018.

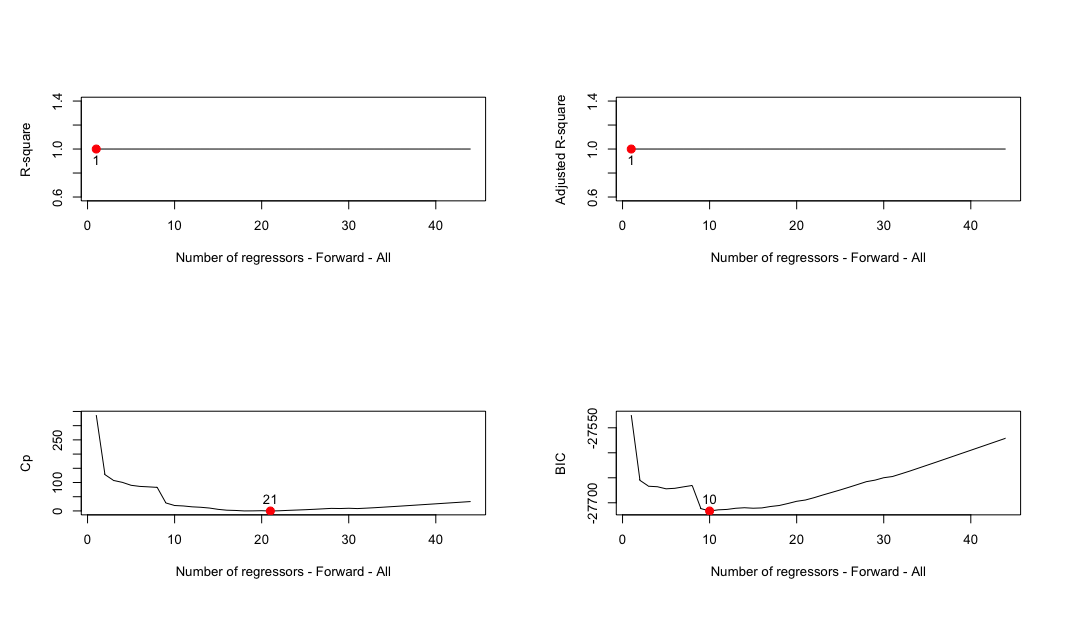
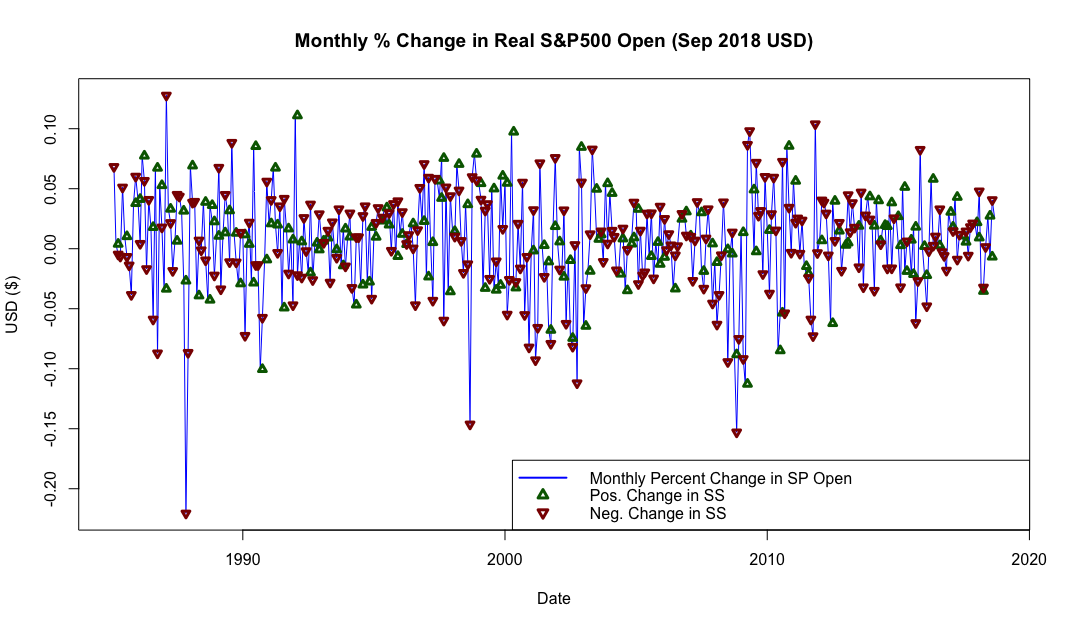
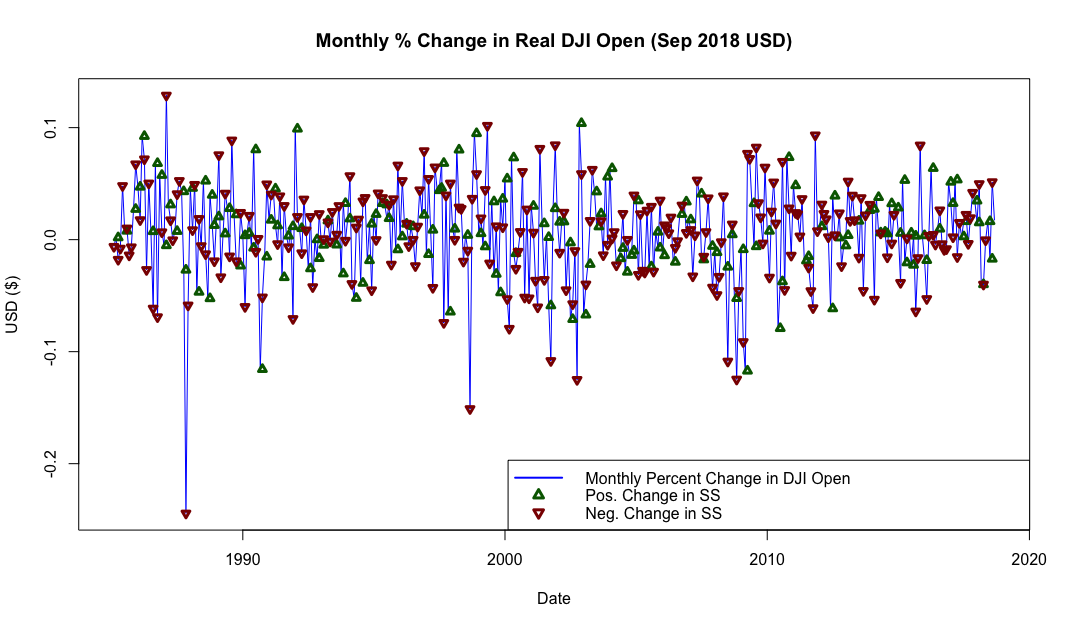
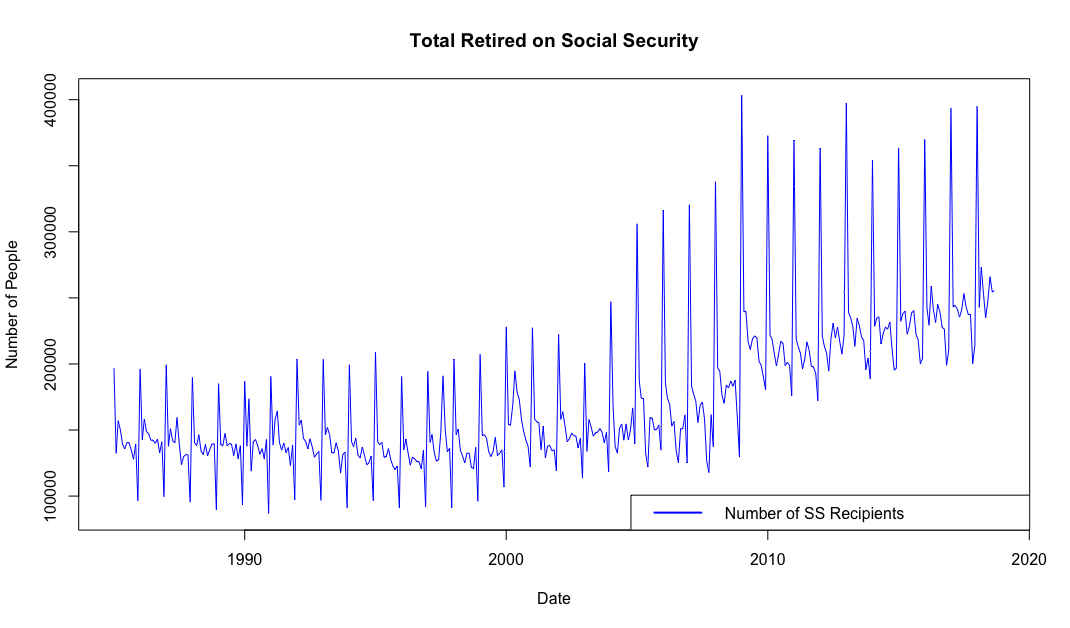
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An Introduction to Statistical Learning with Applications in R (ISLR). Springer New York.

Social Security Administration (2018). *Benefits Awarded by Type of Beneficiary,* https://www.ssa.gov/OACT/ProgData/awards.html accessed October 10, 2018.

Yahoo Finance (2018). *^GSPC Historical Data,* https://finance.yahoo.com/quote/%5EGSPC/history?period1=-630957600&period2=1542520800&interval=1mo&filter=history&frequency=1mo accessed October 10, 2018.

Yahoo Finance (2018). *^DJI Historical Data,* https://finance.yahoo.com/quote/%5EDJI/history?period1=475826400&period2=1542520800&interval=1mo&filter=history&frequency=1mo accessed October 10, 2018.

1. **Appendix: Figures**



1. **Python Code**

#!/usr/bin/env python

# coding: utf-8

# # Data Assembly

# In[1]:

import pandas as pd

import numpy as np

# In[2]:

DJI = pd.read\_csv(

'~/Git/MachineLearningAndBigDataWithR/Data/^DJIMonthly.csv'

, sep = ',')

SP500 = pd.read\_csv(

'~/Git/MachineLearningAndBigDataWithR/Data/^GSPCMonthly.csv'

, sep = ',')

FedFunds = pd.read\_csv(

'~/Git/MachineLearningAndBigDataWithR/Data/FEDFUNDS.csv'

, sep = ',')

SS = pd.read\_csv(

'~/Git/MachineLearningAndBigDataWithR/Data/SSRetired.csv'

, sep = ',')

CPI = pd.read\_csv(

'~/Git/MachineLearningAndBigDataWithR/Data/CPIAUCSL.csv'

, sep = ',')

# ### DJI

# https://finance.yahoo.com/quote/%5EDJI/history?period1=475826400&period2=1542520800&interval=1mo&filter=history&frequency=1mo

# In[3]:

#DJI.head()

DJI.columns = ['date', 'DJIopen', 'DJIhigh', 'DJIlow', 'DJIclose', 'DJIadjClose', 'DJIvolume']

#DJI.head()

# ### S&P 500

# https://finance.yahoo.com/quote/%5EGSPC/history?period1=-630957600&period2=1542520800&interval=1mo&filter=history&frequency=1mo

# In[4]:

#SP500.head()

SP500.columns = ['date', 'SPopen', 'SPhigh', 'SPlow', 'SPclose', 'SPadjClose', 'SPvolume']

#SP500.head()

df = pd.merge(DJI,SP500, how = 'inner', on = 'date')

#df.head()

# ### Federal funds rate

# https://fred.stlouisfed.org/series/FEDFUNDS

# In[5]:

#FedFunds.head()

FedFunds.columns = ['date', 'fedFundRate']

#FedFunds.head()

df = pd.merge(df,FedFunds, how = 'inner', on = 'date')

#df.head()

# ### Retired social security filings

# https://www.ssa.gov/OACT/ProgData/awards.html

# In[6]:

#SS.head()

SS.columns = ['date', 'totalSSRetired', 'averageSSRetiredPay', 'totalMaleSSRetired', 'averageMaleSSRetiredPay', 'totalFemaleSSRetired', 'averageFemaleSSRetiredPay', 'DROPME']

SS.drop('DROPME', axis = 1, inplace=True)

#SS.head()

adjDate = SS['date'].str.split("-", n = 1, expand = True)

#adjDate.head()

adjDate['month'] = np.where(adjDate[0] == 'Jan', '-01-01'

, np.where(adjDate[0] == 'Feb', '-02-01'

, np.where(adjDate[0] == 'Mar', '-03-01'

,np.where(adjDate[0] == 'Apr', '-04-01'

,np.where(adjDate[0] == 'May', '-05-01'

,np.where(adjDate[0] == 'Jun', '-06-01'

,np.where(adjDate[0] == 'Jul', '-07-01'

,np.where(adjDate[0] == 'Aug', '-08-01'

,np.where(adjDate[0] == 'Sep', '-09-01'

,np.where(adjDate[0] == 'Oct', '-10-01'

,np.where(adjDate[0] == 'Nov', '-11-01'

,np.where(adjDate[0] == 'Dec', '-12-01'

, 0))))))))))))

adjDate['year'] = np.where(adjDate[1].astype(int) <= 18, '20'

,np.where(adjDate[1].astype(int) > 18, '19'

, 0))

adjDate['combined'] = adjDate['year'] + adjDate[1] + adjDate['month']

#adjDate.head()

SS['date'] = adjDate['combined']

#SS.head()

df = pd.merge(df,SS, how = 'inner', on = 'date')

#df.head()

# ### CPI - Consumer Price Index for All Urban Consumers: All Items

# https://fred.stlouisfed.org/series/CPIAUCSL/

# In[7]:

#CPI.head()

CPI.columns = ['date', 'cpi']

df = pd.merge(df,CPI, how = 'inner', on = 'date')

#df.head()

# ## Write to file

# In[8]:

df.to\_csv(path\_or\_buf = '~/Git/MachineLearningAndBigDataWithR/Data/assembled.csv', sep=',

1. **R Markdown**