Predicting Movements in the Quantity of Social Security Retirees

P. Christopher J. Daigle

7 December 2018

Table of Contents

[1. Topic 3](#_Toc512677474)

[2. Model / Methodology 3](#_Toc512677475)

[3. Variables 4](#_Toc512677476)

[4. Data 6](#_Toc512677477)

[5. Summary Statistics 6](#_Toc512677478)

[6. Estimates 7](#_Toc512677479)

[7. Separate Approaches 9](#_Toc512677480)

[8. Concluding Remarks 10](#_Toc512677481)

[9. References 12](#_Toc512677482)

[A. Appendix: Figures 15](#_Toc512677483)

[B. Appendix: Pertinent Script 19](#_Toc512677484)

[C. Appendix: Other Script 20](#_Toc512677485)

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 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), 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 joined the data sets. 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).

As this data was of a time-series nature, it was necessary to ensure only stationary values were utilized for prediction. Thus, 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 measure is a directional change in the quantity of social security recipients 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∆DJIclose, pos∆DJIadjClose, pos∆AverageFemalSSRetiredPay, Pos∆RealSPopen, %∆RealDJIhigh, pos∆RealSPadjClose, pos∆RealAverageFemaleSSRetiredPay, %∆RealDJIadjClose, %∆RealSPadjClose

This is a total of ten factors from the 58 I found by ensuring they were 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. I will allow the statistics to do the evaluation and address the economics.

1. **Summary Statistics**
2. **Model / Methodology**

As this is the case, the observations in the first period are lost. The data begins with 405 observations and reduces to 404 relevant observations because of the type of response. The time dimension is still relatively long and so the loss of this one observation appears trivial.

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, this was the final model for prediction:

At the same split for the training and testing of the 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.

1. **Estimates**
2. **Separate Approaches**
3. **Concluding Remarks**
4. **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**
2. **Appendix: Pertinent Script**
3. **Appendix: Other Script**

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

sort Grade

by Grade: sum Yr

sort Yr

by Yr: sum Grade

\* Grade 6 appears most frequently, so I drop all others to so that the proportion passing, the independent to be measured, is appropriate at the district level.

\*By only observing one grade per district, we only observe each district once instead of 10 times, once for each grade, where the proportion passing is given at the school district's grade level and the membership is given at the district level (the bases of measurement are different). I select year to be 2013 as that is the base for all other measurements.

keep if Grade == 6

keep if Yr == 2013

\*IQR

sort Yr

xtile MemIQR = Mem, n(4)

tab MemIQR, gen(iq)

\* Creates 4 variables - iq1, iq2, iq3, iq4 - by innerquartile ranges of about 25%.

\* Observe first quartile

keep if iq1 == 1 & Yr == 2013

egen AvgMem=mean(Mem)

gen AbsMem=abs(AvgMem-Mem)

sort AbsMem

keep in 1

browse Dist

\* Dist 11 is closest to the mean of the 1'st quartile when selecting only on the 6th Grade in 2013. When desiring to check only IQR1, load data and "keep if Dist == 11"

\* Observe second quartile

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

keep if Grade == 6

keep if Yr == 2013

\*IQR

sort Yr

xtile MemIQR = Mem, n(4)

tab MemIQR, gen(iq)

sum Mem

keep if iq2 == 1 & Yr == 2013

egen AvgMem=mean(Mem)

gen AbsMem=abs(AvgMem-Mem)

sort AbsMem

keep in 1

browse Dist

\* Dist 135 is closest to the mean of the 2'nd quartile when selecting only on the 6th Grade in 2013. When desiring to check only IQR3, load data and "keep if Dist == 135"

\* Observe third quartile

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

keep if Grade == 6

keep if Yr == 2013

\*IQR

sort Yr

xtile MemIQR = Mem, n(4)

tab MemIQR, gen(iq)

sum Mem

keep if iq3 == 1 & Yr == 2013

egen AvgMem=mean(Mem)

gen AbsMem=abs(AvgMem-Mem)

sort AbsMem

keep in 1

browse Dist

\* Dist 372 is closest to the mean of the 3'rd quartile when selecting only on the 6th Grade in 2013. When desiring to check only IQR3, load data and "keep if Dist == 372"

\* Observe fourth quartile

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

keep if Grade == 6

keep if Yr == 2013

\*IQR

sort Yr

xtile MemIQR = Mem, n(4)

tab MemIQR, gen(iq)

sum Mem

keep if iq4 == 1 & Yr == 2013

egen AvgMem=mean(Mem)

gen AbsMem=abs(AvgMem-Mem)

sort AbsMem

keep in 1

browse Dist

\* Dist 132 is closest to the mean of the 4th quartile when selecting only on the 6th Grade in 2013. When desiring to check only IQR3, load data and "keep if Dist == 132"

/\*---------------------------\*/

\* Time Series

/\*---------------------------\*/

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

keep if Grade == 6

egen AvgMem=mean(Mem)

gen AbsMem=abs(AvgMem-Mem)

sort AbsMem

keep in 1

browse Dist

\* District 60 is the one nearest the mean.

clear

use "/Users/daiglechris/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/ISATMetrics2.dta"

set more off

keep if Grade == 6 & Dist == 60

browse

sum Yr allp InfTotPPE InfStatePPE FRLProp Day

clear

\*SmallMBP

\*use "/Users/Cdaigle/Dropbox/Education/BSU/2015 2016/Spring/Hon 491/ISAT/Districts/STATA/MATHEMATICSISATDistrictMaster03262016.dta"

\*iMac

\*use "/Users/2011home/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/MATHEMATICSISATDistrictMaster03262016.dta"

\*BigLaptop

use "/Users/2011home/Library/Mobile Documents/com~apple~CloudDocs/Education/UConn/Spring 2018/App Metrics/Project/MATHEMATICSISATDistrictMaster03262016.dta"

set more off

/\*-------------------------------------\*/

\* Generate cohort numbers, starting at 1 in each district

sort dist year grade

gen int cohort = (grade-year) + 2005

replace cohort = 11-cohort if cohort < 1

\* Cohorts distinctive numbering

by dist, sort: gen n\_dist = 1 if \_n == 1

replace n\_dist = sum(n\_dist)

quietly summ cohort

quietly replace cohort = cohort + (n\_dist-1)\*r(max)

quietly list, noobs clean

/\*-------------------------------------\*/

\* Rename

rename dist Dist

rename year Yr

rename grade Grade

rename annualexpenditure AnnExp

rename taxes Tax

rename otherlocalsources OthLcl

rename statesources State

rename federalsources Fed

rename othersources Oth

rename membership Mem

rename totalperpupilexpenditure TotPPE

rename dayweeks Day

rename frlnumber FRL

rename estimatedtotalpopulationofthesch PopEstDist

rename estimatedpopulation517childrenin Pop517EstDist

rename povestimatednumberofrelevantchil PovPop517Dist

rename fyinflationindex InfInd13

rename annualexpenditureadjustedforinfl AnnInfExp

rename totalppeadjustedforinflation InfPPE

rename PovPop517Dist Pov

\* Labels

label variable Dist "District number"

label variable Yr "Year of observation (2007-2013)"

label variable Grade "Grade of tested students in a district (3-12)"

label variable AnnExp "Total annual expenditure in a district for a particular year"

label variable Tax "Total amount of tax revenue in a district for a particular year"

label variable OthLcl "Total amount of revenue from local sources not contained in other revenue streams"

label variable State "Total amount of revenue from state sources"

label variable Fed "Total amount of revenue from federal sources"

label variable Oth "Total amount of revenue from sources not contained in any other revenue stream"

label variable Mem "Total number of students in a district on a particular day"

label variable TotPPE "Total per-pupil-expenditure by district"

label variable Day "Binary variable indicating if a school district has 4-day school weeks"

label variable FRL "Total number of students in a district receiving free or reduced lunch"

label variable PopEstDist "Estimate from US Census of the population in a district"

label variable Pop517EstDist "Estimate from US Census of the population aged 5 to 17 years in a district"

label variable Pov "Estimate from US Census of the population aged 5 to 17 years living in poverty in a district"

label variable InfInd13 "Index for Inflation in a fiscal year with fiscal year 2013 as the base year (Jul-Jun)"

label variable AnnInfExp "Annual expenditure Inflated to 2013 dollars"

label variable InfPPE "Per-pupil-expenditure Inflated to 2013 dollars"

label variable allss "Average scaled score for all tested"

label variable allbb "Percent of below basic for all tested"

label variable allb "Percent of basic for all tested"

label variable allp "Percent of proficient for all tested"

label variable alla "Percent of advanced for all tested"

label variable maless "Average scaled score for all Males tested"

label variable maletested "Number of Males tested"

label variable malebb "Percent of below basic for Males tested"

label variable maleb "Percent of basic for Males tested"

label variable malep "Percent of proficient for Males tested"

label variable malea "Percent of advanced for Males tested"

label variable femaless "Average scaled score for all Females tested"

label variable femaletested "Number of Females tested"

label variable femalebb "Percent of below basic for Females tested"

label variable femaleb "Percent of basic for Females tested"

label variable femalep "Percent of proficient for Females tested"

label variable femalea "Percent of advanced for Females tested"

label variable aianss "Average scaled score for all American Indian or Alaskan Native tested"

label variable aiantested "Number of American Indian or Alaskan Native tested"

label variable aianbb "Percent of below basic for American Indian or Alaskan Native tested"

label variable aianb "Percent of basic for American Indian or Alaskan Native tested"

label variable aianp "Percent of proficient for American Indian or Alaskan Native tested"

label variable aiana "Percent of advanced for American Indian or Alaskan Native tested"

label variable asianss "Average scaled score for all Asian or Pacific Islander tested"

label variable asiantested "Number of Asian or Pacific Islander tested"

label variable asianbb "Percent of below basic for Asian or Pacific Islander tested"

label variable asianb "Percent of basic for Asian or Pacific Islander tested"

label variable asianp "Percent of proficient for Asian or Pacific Islander tested"

label variable asiana "Percent of advanced for Asian or Pacific Islander tested"

label variable bafamss "Average scaled score for all Black / African American tested"

label variable bafamtested "Number of Black / African American tested"

label variable bafambb "Percent of below basic for Black / African American tested"

label variable bafamb "Percent of basic for Black / African American tested"

label variable bafamp "Percent of proficient for Black / African American tested"

label variable bafama "Percent of advanced for Black / African American tested"

label variable nhopiss "Average scaled score for all Native Hawaiian / Other Pacific Islander tested"

label variable nhopitested "Number of Native Hawaiian / Other Pacific Islander tested"

label variable nhopibb "Percent of below basic for Native Hawaiian / Other Pacific Islander tested"

label variable nhopib "Percent of basic for Native Hawaiian / Other Pacific Islander tested"

label variable nhopip "Percent of proficient for Native Hawaiian / Other Pacific Islander tested"

label variable nhopia "Percent of advanced for Native Hawaiian / Other Pacific Islander tested"

label variable whitess "Average scaled score for all White tested"

label variable whitetested "Number of White tested"

label variable whitebb "Percent of below basic for White tested"

label variable whiteb "Percent of basic for White tested"

label variable whitep "Percent of proficient for White tested"

label variable whitea "Percent of advanced for White tested"

label variable hisplatss "Average scaled score for all Hispanic or Latino tested"

label variable hisplattested "Number of Hispanic or Latino tested"

label variable hisplatbb "Percent of below basic for Hispanic or Latino tested"

label variable hisplatb "Percent of basic for Hispanic or Latino tested"

label variable hisplatp "Percent of proficient for Hispanic or Latino tested"

label variable hisplata "Percent of advanced for Hispanic or Latino tested"

label variable tworacesss "Average scaled score for all Other/Unknown tested"

label variable tworacestested "Number of Other/Unknown tested"

label variable tworacesbb "Percent of below basic for Other/Unknown tested"

label variable tworacesb "Percent of basic for Other/Unknown tested"

label variable tworacesp "Percent of proficient for Other/Unknown tested"

label variable tworacesa "Percent of advanced for Other/Unknown tested"

label variable frlss "Average scaled score for all Free or Reduced Lunch tested"

label variable frltested "Number of Free or Reduced Lunch tested"

label variable frlbb "Percent of below basic for Free or Reduced Lunch tested"

label variable frlb "Percent of basic for Free or Reduced Lunch tested"

label variable frlp "Percent of proficient for Free or Reduced Lunch tested"

label variable frla "Percent of advanced for Free or Reduced Lunch tested"

label variable lepss "Average scaled score for all Limited English Proficient tested"

label variable leptested "Number of Limited English Proficient tested"

label variable lepbb "Percent of below basic for Limited English Proficient tested"

label variable lepb "Percent of basic for Limited English Proficient tested"

label variable lepp "Percent of proficient for Limited English Proficient tested"

label variable lepa "Percent of advanced for Limited English Proficient tested"

label variable migss "Average scaled score for all Migrant tested"

label variable migtested "Number of Migrant tested"

label variable migbb "Percent of below basic for Migrant tested"

label variable migb "Percent of basic for Migrant tested"

label variable migp "Percent of proficient for Migrant tested"

label variable miga "Percent of advanced for Migrant tested"

label variable spess "Average scaled score for all Special Education tested"

label variable spetested "Number of Special Education tested"

label variable spebb "Percent of below basic for Special Education tested"

label variable speb "Percent of basic for Special Education tested"

label variable spep "Percent of proficient for Special Education tested"

label variable spea "Percent of advanced for Special Education tested"

label variable tiass "Average scaled score for all Title 1 A tested"

label variable tiatested "Number of Title 1 A tested"

label variable tiabb "Percent of below basic for Title 1 A tested"

label variable tiab "Percent of basic for Title 1 A tested"

label variable tiap "Percent of proficient for Title 1 A tested"

label variable tiaa "Percent of advanced for Title 1 A tested"

/\*----------------------------\*/

\* Set Panel Data

xtset cohort Yr

/\*---------------------------\*/

\* Variable Creation

/\*---------------------------\*/

\*Year Binary

gen Yr07 = 0

replace Yr07 = 1 if Yr == 2007

gen Yr08 = 0

replace Yr08 = 1 if Yr == 2008

gen Yr09 = 0

replace Yr09 = 1 if Yr == 2009

gen Yr10 = 0

replace Yr10 = 1 if Yr == 2010

gen Yr11 = 0

replace Yr11 = 1 if Yr == 2011

gen Yr12 = 0

replace Yr12 = 1 if Yr == 2012

gen Yr13 = 0

replace Yr13 = 1 if Yr == 2013

\*Inflation

gen CPIBase = (231.893+233.001+234.083+234.966+233.206+232.029+232.759+234.595+235.511+235.488+235.979+236.227)/12

gen Seven = (206.7+207.5+207.8+207.1+206.3+206.2+207.79+208.995+210.778+212.036+213.063+212.68)/12

gen Eight = (212.542+212.406+212.92+213.917+214.904+214.733+215.739+216.339+218.533+219.437+221.009+223.04)/12

gen Nine = (223.867+222.823+222.132+221.034+217.113+214.685+215.923+217.095+217.357+217.91+218.567+219.865)/12

gen Ten = (219.484+219.884+220.294+220.447+219.728+219.307+219.989+220.179+220.809+221.202+221.417+221.147)/12

gen Eleven = (221.331+221.523+221.384+221.708+221.671+222.081+223.149+224.431+226.558+227.837+228.516+228.075)/12

gen Twelve = (227.805+228.222+229.147+229.195+228.771+228.117+228.98+229.995+232.039+232.561+233.053+232.701)/12

gen Thirt = (231.893+233.001+234.083+234.966+233.206+232.029+232.759+234.595+235.511+235.488+235.979+236.227)/12

gen CPI07 = CPIBase / Seven

gen CPI08 = CPIBase / Eight

gen CPI09 = CPIBase / Nine

gen CPI10 = CPIBase / Ten

gen CPI11 = CPIBase / Eleven

gen CPI12 = CPIBase / Twelve

gen CPI13 = CPIBase / Thirt

replace InfInd13 = CPI07 if Yr == 2007

replace InfInd13 = CPI08 if Yr == 2008

replace InfInd13 = CPI09 if Yr == 2009

replace InfInd13 = CPI10 if Yr == 2010

replace InfInd13 = CPI11 if Yr == 2011

replace InfInd13 = CPI12 if Yr == 2012

replace InfInd13 = CPI13 if Yr == 2013

gen InfExp = AnnExp\*InfInd13

gen LInfExp = log(InfExp)

gen InfTax = Tax\*InfInd13

gen LInfTax = log(InfTax)

gen InfOthLcl = OthLcl\*InfInd13

gen LInfOthLcl = log(InfOthLcl)

gen InfState = State\*InfInd13

gen LInfState = log(InfState)

gen InfFed = Fed\*InfInd13

gen LInfFed = log(InfFed)

gen InfOth = Oth\*InfInd13

gen LInfOth = log(InfOth)

\*IQR

bys Yr: sum Mem, detail

\*Mem

gen LMem = log(Mem)

\*MemIQR1

gen MemIQR1 = 0

replace MemIQR1 = 1 if Mem <= 282.81 & Yr07 == 1

replace MemIQR1 = 1 if Mem <=284.69 & Yr08 == 1

replace MemIQR1 = 1 if Mem <=285.4 & Yr09 == 1

replace MemIQR1 = 1 if Mem <=270.37 & Yr10 == 1

replace MemIQR1 = 1 if Mem <=251.56 & Yr11 == 1

replace MemIQR1 = 1 if Mem <=251.56 & Yr12 == 1

replace MemIQR1 = 1 if Mem <=253.44 & Yr13 == 1

replace MemIQR1 = . if MemIQR1 == 0

gen IntMemIQR1 = 1 if MemIQR1 == 1

replace IntMemIQR1 = . if MemIQR1 == 0

gen BMemIQR1 = MemIQR1

replace BMemIQR1 = 0 if missing(BMemIQR1)

gen LMemIQR1 = log(MemIQR1)

\*MemIQR2

gen MemIQR2 = 0

replace MemIQR2 = 1 if Mem > 282.81 & Mem <= 750.97 & Yr07 == 1

replace MemIQR2 = 1 if Mem > 284.69 & Mem <= 732 & Yr08 == 1

replace MemIQR2 = 1 if Mem > 270.37 & Mem <= 708.52 & Yr09 == 1

replace MemIQR2 = 1 if Mem > 270.37 & Mem <= 619.86 & Yr10 == 1

replace MemIQR2 = 1 if Mem > 251.56 & Mem <= 593.01 & Yr11 == 1

replace MemIQR2 = 1 if Mem > 251.56 & Mem <= 590.61 & Yr12 == 1

replace MemIQR2 = 1 if Mem > 253.44 & Mem <= 590.41 & Yr13 == 1

replace MemIQR2 = . if MemIQR2 == 0

gen IntMemIQR2 = 1 if MemIQR2 == 1

replace IntMemIQR2 = . if MemIQR2 == 0

gen BMemIQR2 = MemIQR2

replace BMemIQR2 = 0 if missing(BMemIQR2)

gen LMemIQR2 = log(MemIQR2)

\*MemIQR3

gen MemIQR3 = 0

replace MemIQR3 = 1 if Mem > 750.97 & Mem <= 1704.23 & Yr07 == 1

replace MemIQR3 = 1 if Mem > 732 & Mem <= 2011.2 & Yr08 == 1

replace MemIQR3 = 1 if Mem > 708.52 & Mem <= 2039.4 & Yr09 == 1

replace MemIQR3 = 1 if Mem > 619.86 & Mem <= 1727.82 & Yr10 == 1

replace MemIQR3 = 1 if Mem > 593.01 & Mem <= 1588.01 & Yr11 == 1

replace MemIQR3 = 1 if Mem > 590.61 & Mem <= 1516.98 & Yr12 == 1

replace MemIQR3 = 1 if Mem > 590.41 & Mem <= 1597.14 & Yr13 == 1

replace MemIQR3 = . if MemIQR3 == 0

gen IntMemIQR3 = 1 if MemIQR3 == 1

replace IntMemIQR3 = . if MemIQR3 == 0

gen BMemIQR3 = MemIQR3

replace BMemIQR3 = 0 if missing(BMemIQR3)

gen LMemIQR3 = log(MemIQR3)

\*MemIQR4

gen MemIQR4 = 0

replace MemIQR4 = 1 if Mem > 1704.23 & Yr07 == 1

replace MemIQR4 = 1 if Mem > 2011.2 & Yr08 == 1

replace MemIQR4 = 1 if Mem > 2039.4 & Yr09 == 1

replace MemIQR4 = 1 if Mem > 1727.82 & Yr10 == 1

replace MemIQR4 = 1 if Mem > 1588.01 & Yr11 == 1

replace MemIQR4 = 1 if Mem > 1516.98 & Yr12 == 1

replace MemIQR4 = 1 if Mem > 1597.14 & Yr13 == 1

replace MemIQR4 = . if MemIQR4 == 0

gen IntMemIQR4 = 1 if MemIQR4 == 1

replace IntMemIQR4 = . if MemIQR4 == 0

gen BMemIQR4 = MemIQR4

replace BMemIQR4 = 0 if missing(BMemIQR4)

gen LMemIQR4 = log(MemIQR4)

\*Membership of Dists in IQR1

gen MemIQR107 = Mem if MemIQR1 == 1 & Yr07 == 1

gen MemIQR108 = Mem if MemIQR1 == 1 & Yr08 == 1

gen MemIQR109 = Mem if MemIQR1 == 1 & Yr09 == 1

gen MemIQR110 = Mem if MemIQR1 == 1 & Yr10 == 1

gen MemIQR111 = Mem if MemIQR1 == 1 & Yr11 == 1

gen MemIQR112 = Mem if MemIQR1 == 1 & Yr12 == 1

gen MemIQR113 = Mem if MemIQR1 == 1 & Yr13 == 1

\*Membership of Dists in IQR2

gen MemIQR207 = Mem if MemIQR2 == 1 & Yr07 == 1

gen MemIQR208 = Mem if MemIQR2 == 1 & Yr08 == 1

gen MemIQR209 = Mem if MemIQR2 == 1 & Yr09 == 1

gen MemIQR210 = Mem if MemIQR2 == 1 & Yr10 == 1

gen MemIQR211 = Mem if MemIQR2 == 1 & Yr11 == 1

gen MemIQR212 = Mem if MemIQR2 == 1 & Yr12 == 1

gen MemIQR213 = Mem if MemIQR2 == 1 & Yr13 == 1

\*Membership of Dists in IQR3

gen MemIQR307 = Mem if MemIQR3 == 1 & Yr07 == 1

gen MemIQR308 = Mem if MemIQR3 == 1 & Yr08 == 1

gen MemIQR309 = Mem if MemIQR3 == 1 & Yr09 == 1

gen MemIQR310 = Mem if MemIQR3 == 1 & Yr10 == 1

gen MemIQR311 = Mem if MemIQR3 == 1 & Yr11 == 1

gen MemIQR312 = Mem if MemIQR3 == 1 & Yr12 == 1

gen MemIQR313 = Mem if MemIQR3 == 1 & Yr13 == 1

\*Membership of Dists in IQR4

gen MemIQR407 = Mem if MemIQR4 == 1 & Yr07 == 1

gen MemIQR408 = Mem if MemIQR4 == 1 & Yr08 == 1

gen MemIQR409 = Mem if MemIQR4 == 1 & Yr09 == 1

gen MemIQR410 = Mem if MemIQR4 == 1 & Yr10 == 1

gen MemIQR411 = Mem if MemIQR4 == 1 & Yr11 == 1

gen MemIQR412 = Mem if MemIQR4 == 1 & Yr12 == 1

gen MemIQR413 = Mem if MemIQR4 == 1 & Yr13 == 1

\*LInfState by IQR

gen IQR1LInfStatePPE = LInfState if MemIQR1 == 1

gen IQR2LInfStatePPE = LInfState if MemIQR2 == 1

gen IQR3LInfStatePPE = LInfState if MemIQR3 == 1

gen IQR4LInfStatePPE = LInfState if MemIQR4 == 1

\*Sums

gen NonLclExp = Fed + State

gen InfNonLclExp = InfFed + InfState

\*PPE

replace TotPPE = AnnExp / Mem

gen LTotPPE = log(TotPPE)

gen InfTotPPE = InfExp / Mem

gen LInfTotPPE = log(InfTotPPE)

gen TaxPPE = Tax / Mem

gen InfTaxPPE = InfTax /Mem

gen LInfTaxPPE = log(InfTaxPPE)

gen OthLclPPE = OthLcl / Mem

gen InfOthLclPPE = InfOthLcl / Mem

gen LInfOthLclPPE = log(InfOthLclPPE)

gen StatePPE = State / Mem

gen InfStatePPE = InfState / Mem

gen LInfStatePPE = log(InfStatePPE)

gen FedPPE = Fed / Mem

gen InfFedPPE = InfFed / Mem

gen LInfFedPPE = log(InfFedPPE)

gen OtherPPE = Oth / Mem

gen InfOthPPE = InfOth / Mem

gen LInfOthPPE = log(InfOthPPE)

gen NonLclPPE = NonLclExp / Mem

gen InfNonLclPPE = InfNonLclExp / Mem

gen LInfNonLclPPE = log(InfNonLclPPE)

\*Spending

gen Lcl = AnnExp - (State + Fed)

gen InfLcl = Lcl \* InfInd13

gen InfLclPPE = InfLcl / Mem

gen LInfLclPPE = log(InfLclPPE)

\*Proporional Poverty

/\*By proportion of estimated population in poverty (5-17 years old) in a district

or by membership, corr of membership and pop517 very high, similar\*/

gen Pov517Prop = Pov / Mem

\*Proportional FRL

gen FRLProp = FRL / Mem

\*4Day

bys Dist Yr: gen DCount = 1 if Day == 1 & \_n == 1

/\*sorts data by district number then by year and assigns the value 1 to DCount

if it's the first observation (\_n == 1) of a district in a year having a 4day

(Day == 1) schedule.\*/

bys Yr: sum DCount

bys Dist Yr: gen RegCount = 1 if Day == 0 & \_n == 1

\*Membership by 4Day

gen MemDay = DCount \* Mem

bys Yr: sum MemDay

\*Membership by 4day and InfStatePPE

gen MemDayPPE = DCount \* InfStatePPE

bys Yr: sum MemDayPPE

bys Dist Yr: gen DCount07 = 1 if Day == 1 & Yr07 == 1 & \_n == 1

bys Dist Yr: gen DCount08 = 1 if Day == 1 & Yr08 == 1 & \_n == 1

bys Dist Yr: gen DCount09 = 1 if Day == 1 & Yr09 == 1 & \_n == 1

bys Dist Yr: gen DCount10 = 1 if Day == 1 & Yr10 == 1 & \_n == 1

bys Dist Yr: gen DCount11 = 1 if Day == 1 & Yr11 == 1 & \_n == 1

bys Dist Yr: gen DCount12 = 1 if Day == 1 & Yr12 == 1 & \_n == 1

bys Dist Yr: gen DCount13 = 1 if Day == 1 & Yr13 == 1 & \_n == 1

\*Mem 4Day by IQR

gen MemDayIQR1 = Mem if DCount == 1 & MemIQR1 == 1

/\*gen Day13IQR1 = Mem \* DCount13 if MemIQR1 == 1\*/

bys Yr: sum MemDayIQR1

gen MemDayIQR2 = Mem if DCount == 1 & MemIQR2 == 1

gen MemDayIQR3 = Mem if DCount == 1 & MemIQR3 == 1

gen MemDayIQR4 = Mem if DCount == 1 & MemIQR4 == 1

\*Membership by 4day and IQR1

bys Yr: sum MemDayIQR1

bys Yr: sum MemDayIQR1 if missing(DCount)

\*Membership by 4day and IQR2

bys Yr: sum MemDayIQR2

bys Yr: sum MemDayIQR2 if missing(DCount)

\*Membership by 4day and IQR3

bys Yr: sum MemDayIQR3

bys Yr: sum MemDayIQR3 if missing(DCount)

\*Membership by 4day and IQR3

bys Yr: sum MemDayIQR3

bys Yr: sum MemDayIQR3 if missing(DCount)

\*AllPass

gen Pass = alla + allp

\*AllFail

gen Fail = allb + allbb

\*Ratios

gen FedRatioExp = Fed / AnnExp

gen InfFedRatioExp = InfFed / InfExp

gen StateRatioExp = State / AnnExp

gen InfStateRatioExp = InfState / InfExp

gen NonLclRatioExp = NonLclExp / AnnExp

gen InfNonLclRatioExp = InfNonLclExp / InfExp

\*Other Variables

\*Interacted Binaries

gen InfState2007 = InfStatePP\*Yr07

gen InfState2008 = InfStatePP\*Yr08

gen InfState2009 = InfStatePP\*Yr09

gen InfState2010 = InfStatePP\*Yr10

gen InfState2011 = InfStatePP\*Yr11

gen InfState2012 = InfStatePP\*Yr12

gen InfState2013 = InfStatePP\*Yr13

gen Interact = Pov \* FRL