Social Security MTR Calculation

Introduction:

This document explains the process by which we calculate the marginal tax rates (MTR) for Social Security for individuals in the CPS dataset.

Methodology:

We begin by including only individuals for our MTR calculation that are in the labor force, which we define by three criteria:

1. Their reported age is between 18 and 65 inclusive

2. They're not currently enrolled as a part-time or full-time student (a_ftpt from CPS)

3. Their earned income is greater than 0

Note: We define earned income as the sum of 'wsal_val', 'semp_val', and 'frse_val' from CPS

Once we exclude those who aren't in the labor force, we use earnings function to predict the earnings in a given year of each individual. In this estimation, we include several supplementary variables in addition to the schooling and potential experience terms used in Mincer's earnings function[1.]. Previous literature has shown that earnings differ for various demographic groups; for example, women earn less than men, and blacks earn less than whites. Thus it is reasonable to include gender and race into this earnings function. In addition, the earnings literature suggests that there exists a large marital status wage gap within genders. We use the children dummy as the proxy for marital status. Children dummy is constructed using the age of the oldest child in each family. For each person, children dummy equals to 0 for the years before he has the first child, children dummy equals to 1 for the years after he has the first child. Major industry is also added as independent regressor. Thus, the final equation is given by

$$\ln(y) = \ln(y_0) + \beta_1 S + \beta_2 X + \beta_2 X^2 + \beta_3 F + \beta_4 R + \beta_5 C + \beta_5 F * C + \beta_6 I$$

where $\ln(y)$ is the logarithm of earnings, y_0 is the earnings of somebody with no education or experience, S is years of education, X is years of work experience, F is a categorical gender dummy variable for being a female, R is a categorical race dummy variable that includes the statistically significant races as dummy variables (for normal CPS: AI_HP_in, Asian_only, Black_AI_Asian, Hawaiian_in, White_only, for CPSRETS: race_1.0' , 'race_4.0','race_5.0' ,'race_6.0' ,'race_8.0','race_14.0', 'race_21.0' ,'race_22.0'), C is the dummy variable for having children, F*C is an interaction term between gender and children, and I is the dummy variable for major industry (agriculture and forestry, armed forces, construction, educational and health services, financial activities, information, leisure and hospitality, manufacturing, mining, other services, professional and business, public administration, transportation and utilities, wholesale and retail trade).

To specify *S* in our calculation, we use the variable 'a_hga' from the CPS dataset and assign each possible category of education a number, which we define as YrsPstHS in the python code, to reflect how many years beyond high school it takes to finish your education.

Degree Type	YrsPstHS or 'S'	Degree Type	YrsPstHS or 'S'
Less than high school	0	Bachelor's degree	5
High school graduate	1	Master's degree	7
Some college but no degree	2	Professional school degree	10
Associate degree	3	Doctorate degree	10

We then assume that each individual in the labor force maintains the same level of education for the remainder of their lives and began working immediately upon completing their education. Thus we define experience as

$$X = age - S - 17$$

(Example: An individual aged 34 received a master's degree. Then *X* would be 34-7-17 = 10.)

Now that these variables are clearly defined, we were able to perform a regression using the variables earned_income, YrsPstHS, experience, experience_squared, prdtrace, a_sex, child, and a_mjind to determine the coefficients. With these coefficients, we can imputed each individual's lifetime earnings for each working year of their life (we assume all begin working after completing education and work until 65).

The results of the regression are below.

```
                    OLS Regression Results
==============================================================================
Dep. Variable:        earned_income   R-squared:                 0.246
Model:                          OLS   Adj. R-squared:            0.245
Method:               Least Squares   F-statistic:               920.7
Date:              Thu, 04 May 2017   Prob (F-statistic):         0.00
Time:                      15:45:23   Log-Likelihood:           -79824.
No. Observations:             59407   AIC:                   1.597e+05
Df Residuals:                 59385   BIC:                   1.599e+05
Df Model:                        21
Covariance Type:          nonrobust
```

==

	coef	std err	t	P>|t|	[0.025	0.975]
Reg_YrsPstHS	0.1517	0.002	94.361	0.000	0.149	0.155
experience	0.0598	0.001	46.723	0.000	0.057	0.062
experienceSquared	-0.0010	2.67e-05	-37.885	0.000	-0.001	-0.001
a_sex	8.1996	0.024	339.130	0.000	8.152	8.247
child	-1.118e-15	1.16e-16	-9.619	0.000	-1.35e-15	-8.9e-16
a_sex_child	-7.02e-16	1.71e-16	-4.111	0.000	-1.04e-15	-3.67e-16
Agriculture, forestry,	0.8368	0.036	23.566	0.000	0.767	0.906
Construction	1.0392	0.024	43.701	0.000	0.993	1.086
Educational and health services	0.7401	0.021	35.850	0.000	0.700	0.781
Financial activities	1.1250	0.024	47.000	0.000	1.078	1.172
Information	1.0861	0.032	33.947	0.000	1.023	1.149
Leisure and hospitality	0.6009	0.023	25.873	0.000	0.555	0.646
Manufacturing	1.1515	0.022	51.894	0.000	1.108	1.195
Mining	1.5405	0.044	34.803	0.000	1.454	1.627
Other services	0.6507	0.026	25.187	0.000	0.600	0.701
Professional and business	0.9232	0.022	41.813	0.000	0.880	0.966
Public administration	1.1800	0.026	46.251	0.000	1.130	1.230
Transportation and utilities	1.1255	0.025	44.649	0.000	1.076	1.175
Wholesale and retail trade	0.8256	0.022	38.127	0.000	0.783	0.868
AI-HP	-1.6531	0.656	-2.520	0.012	-2.939	-0.367
Asian only	0.1610	0.019	8.382	0.000	0.123	0.199
Black-AI-Asian	-2.2400	0.928	-2.414	0.016	-4.059	-0.421
Hawaiian/Pacific Islander	0.1612	0.054	2.976	0.003	0.055	0.267
White only	0.1405	0.011	12.376	0.000	0.118	0.163

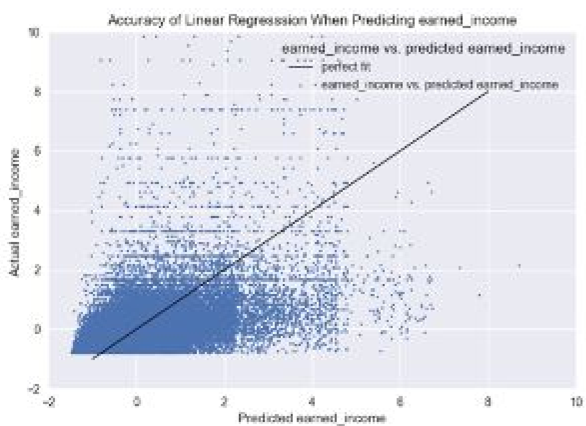
==

Omnibus:	23990.545	Durbin-Watson:	1.944
Prob(Omnibus):	0.000	Jarque-Bera (JB):	196180.005
Skew:	-1.735	Prob(JB):	0.00
Kurtosis:	11.199	Cond. No.	5.04e+20

==

Figure 1 shows the accuracy of linear regression when predicting earned_income. We standardize both actual earned income and predicted earned income in this graph.

Figure 1



We create a vector of lifetime earnings for each year of an individual's working life until the present year by plugging in the individual's education, and their work experience in any given year. We then scale the earnings vector according to an average wage index found at https://www.ssa.gov/oact/cola/AWI.html so that future earnings are not in terms of 2014 earnings. We keep pre-2014 earnings at 2014 level, and index them upon retirement accordingly

<u>For Scripts in MTR_Anypiab_SS Folder</u>

This vector of earnings can then be used to estimate the individual's estimated monthly social security benefit using the Social Security Administration's calculator, called Anypiab, located at website at https://www.ssa.gov/oact/anypia/anypiab.html . Once one downloads the anypiab.exe file into the same directory as our SS_MTR_anypia.py script found in the MTR_Anypiab_SS folder, our script uses this anypiab applet by creating .pia compatible files that contain the lifetime earnings, birthday, year of retirement, and gender of each recipient. Our script commands the anypiab applet to take in the .pia files compute the estimated monthly social security benefit. We then extract this benefit amount and repeat these steps with an added earnings adjustment to calculate MTRs.

We format the lifetime earnings vectors four different ways based on four different earnings assumptions before we plug them into the anypiab applet. First, we assume that there are no future earnings. We do this by setting all future earnings of each individual from the year 2015 until they retire equal to 0. Second, we use our regression to predict the future wages for all the working years of each individual's life (and appropriately scale them with the wage index). Third, we assume that after 2014, earnings remain constant at the 2014 pre-adjustment earnings amount. Lastly, we use the Anypiab program's future earnings projection, called the 2016 Trustees Report Alternative II (which is a moderate wage increase assumption, rather than pessimistic or optimistic). Each of these assumptions produces different SS benefit results, and subsequently different SS MTRs.

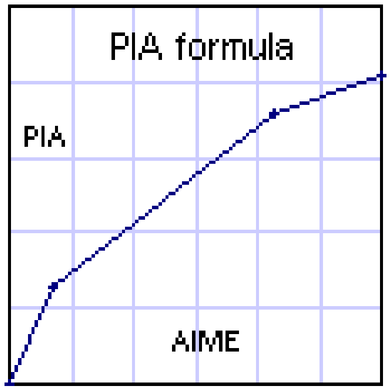
To calculate SS MTR amounts for each individual, we first extract the SS benefit amount from the anypiab output file, then we make this benefit representative of a lifetime benefit by multiplying by 12 to make it yearly, then by the number of years remaining in the individual's lifetime, which we assume to be 13 after retirement (78 years old). Once we get this total lifetime benefit, we implement a $500 adjustment/increase to the current year's (2014) earnings in the lifetime earnings vector and recalculate the monthly benefit using the applet (we use a $500 adjustment because any smaller adjustment doesn't change the AIME amount, and consequently the monthly SS benefit amount on the anypiab app).

Finally, the script takes this new benefit calculation, stored in the output file, and similarly multiply by 12 and 13 to make it a lifetime benefit, take the difference between the old and new lifetime benefit, and divide that difference by the $500 adjustment to get the marginal tax rates for each individual. We perform this set of steps for each individual in the labor force of the CPS dataset.

FOR all other MTR calculation scripts

Since the anypiab calculator produces random errors when calculating SS benefit for large datasets, we also coded up income rules by creating our own version of the anypiab calculator. Here are examples to illustrate how retirement benefits are calculated at https://www.ssa.gov/OACT/ProgData/retirebenefit1.html.

We follow the process that the Anypiab applet uses to calculate SS benefits, which are consistent with SSA benefit calculations. First, the Anypiab applet calculates the average indexed monthly earnings (AIME). AIME is defined as the sum of the indexed earnings of the thirty-five years with the highest earnings, divided by 35 (number of years considered) and twelve (months in year). It's important to note that Social Security's Old-Age, Survivors, and Disability Insurance (OASDI) program limits the amount of earnings subject to taxation for a given year. This limit, known as maximum earnings, changes each year with changes in the national average wage index. Secondly, Anypiab applet determines the Primary Insurance Amounts (PIA) that individuals receive each month by applying a specific formula to AIME. The PIA splits total AIME into three different ranges that are given by bend points. The AIME that falls within the first bend point's range is multiplied by .9. The AIME that falls within the second and third bend point is multiplied by .32, and the AIME that is greater than the third bend point is multiplied by .15. The sum of all of these products is the PIA. Bend points also change annually with changes in the national average wage index. Bend points for the year of first eligibility (the year a person attains age 62 in retirement cases) are used in the calculation.

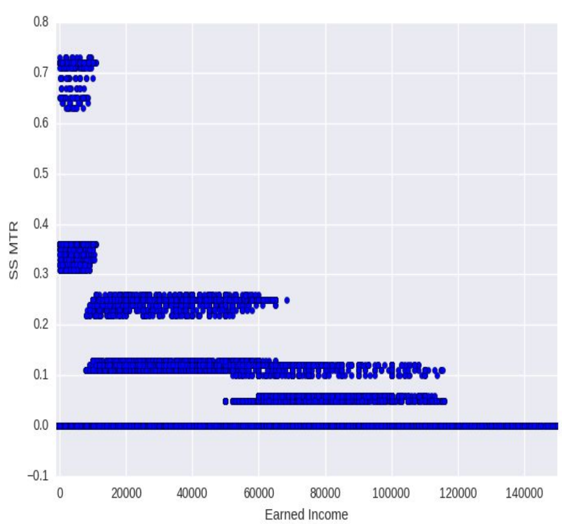


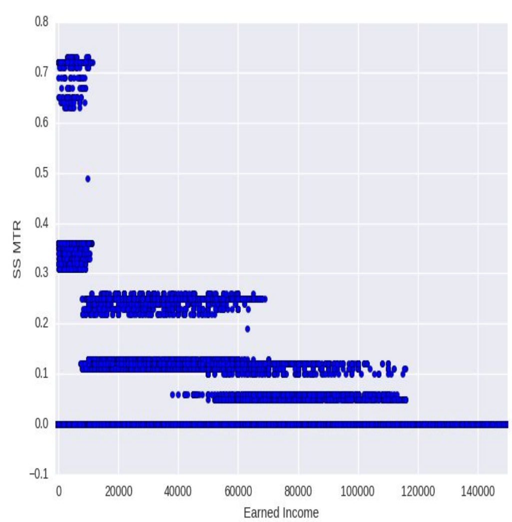
Lastly, the Anypiab applet calculates monthly retirement benefits based on PIA and age. Because in this Social Security MTR Calculation, we assume all individuals retire at age 65, each person get credit for delayed retirement. The indexing amount is determined by the benefit increase assumptions based

We then take the calculated PIA benefit and create a vector of PIA benefits over the next 13 years (until death, 78) by indexing the retirement year's PIA benefit with future CPI index corresponding to how many years after retirement the benefit is considered. We then take the vector of monthly PIA benefits corresponding to each year, and multiply them by 12 to make it a vector of yearly PIA benefits. We then sum the vector of PIA benefits, to get a lifetime benefit amount. Once we get this total lifetime benefit, we implement a $500 adjustment/increase to the current year's (2014) earnings in the lifetime earnings vector and recalculate the PIA and lifetime SS benefit in the same way again.

After we have calculated the pre- and post-adjustment lifetime SS benefits, we take the difference between the old and new lifetime benefit, and divide that difference by the $500 adjustment to get the marginal tax rates for each individual. We perform this set of steps for each individual in the labor force of the CPS dataset.

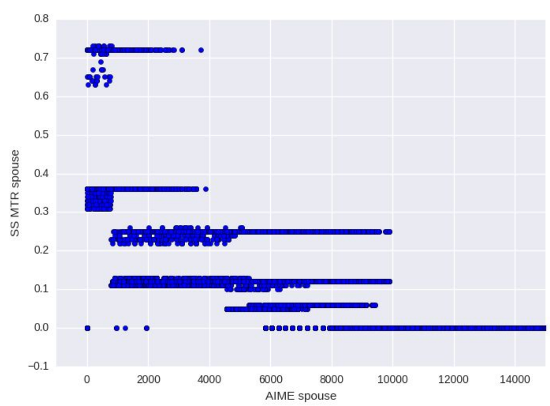
Below we give the MTRs for different individuals for the CPS using the constant future earnings and the future earnings based on our regression respectively :

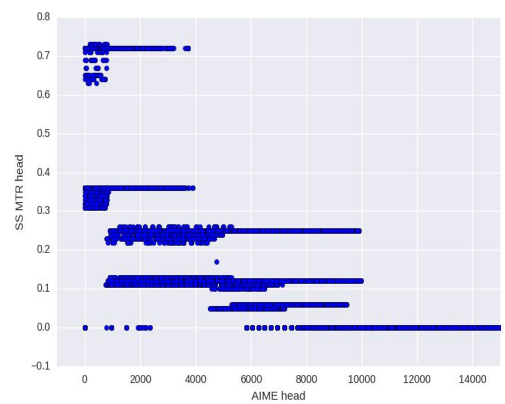




Notice that the MTRs decrease as earned income increases. Also notice that are roughly 6 different echelons of MTRs. Much of this difference is due to the bend points; however, the anypiab applet introduces rounding error at each step. First, AIME is rounded down to the nearest dollar. This introduces rounding error especially, for example, if AIME before adjustment was 4985.99, which would be rounded down to 4985, and a $500 earnings adjustment pushes it up to 4987.01, then the new AIME would be 4987. This is difference will reflect a much higher MTR than if the rounding was not done. This is why there are MTRs as high as .72. Also much of discrepancies within the echelons is due to this same rounding. Rounding is also done to the PIA amounts, since they are measured in dollars, they must be rounded down to the nearest ten cent. The indexed future CPI benefits also can compound MTRs. Note: These graphs include all individuals from CPS, not just those in the labor force. This explains the zero MTRs along entire earned income range.

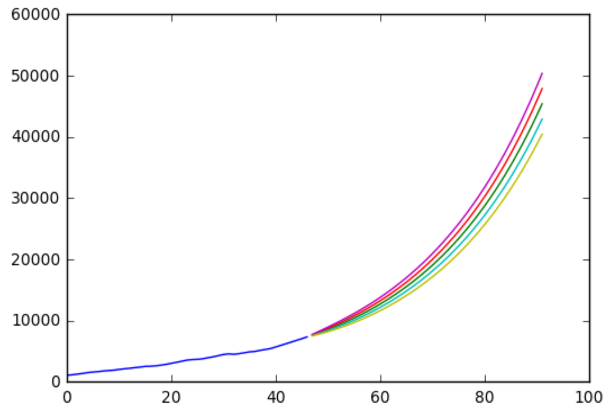
Now we show the MTRs calculated with respect to AIME instead and for the head and spouse of tax units in the CPSRETS file. These are according to constant future earnings assumptions, with indexed earnings and SS benefit.

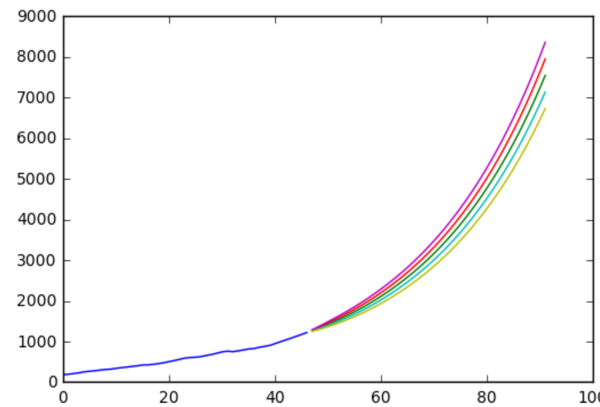


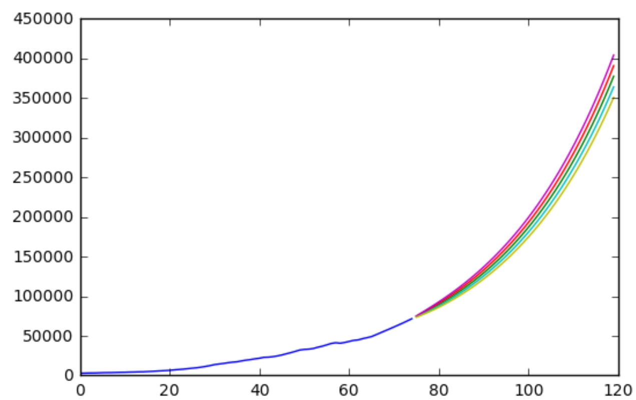


Note that all AIME that is greater than one signifies that these individuals are part of the labor force. However, there are still SS MTR values of zero for low AIME amounts due to rounding errors as well. AIME goes up but is rounded down still to nearest dollar (938.01 AIME before with 938.99 after rounds down to same integer). Also, since the top 35 years of earnings are chosen, this can offset the effect of current year adjustments (for example if last years earnings wasn't in top earnings before adjustment but gets marginally bumped into top 35 after adjustment). Also, we see that MTRs decrease still with increasing AIME.

Since the SSA 2016 Trustees report, which contains the average wages, maximum earnings, CPI index, and bend points for different years used in the Anypiab applet, only contains information up to the year 2025, we needed a way to obtain the values for these indicators from the year 2026 up to the year 2070. We used an ARMA model in state space form, with Kalman Filtering to predict these indicators. The ARMA/ Kalman code used to generate these indicators is found at https://github.com/parkerrogers/Benefits/tree/master/SS/MTR/Income_Rules_CPSnormal/arma_projections. A plot of the projections for bend points 1, 2 and averages wages respectively is given below:







The middle green line indicates the prediction line, and the others represent one and two standard deviations away from the prediction in either direction. We chose to use a more modest two standard deviation below the prediction. Apart from the bend points and average wages, we were able to obtain the maximum earnings and CPI increases from 2026-2070 from the SSA. All of these time dynamics are found in the repository given as Bendpoints.csv, Max_Earnings.csv, averagewages.csv and CPI_Intermediate.csv .

1. Mincer, Jacob (1958). "Investment in Human Capital and Personal Income Distribution". *Journal of Political Economy*. **66** (4): 281–302. doi:10.1086/258055. JSTOR 1827422.