```
clear
use "/Users/asherdvir-djerassi/Desktop/OneDrive/Div III/Div III_/Empirical Piece/IPUMS/CPS March
Full Data/2008.dta"
replace inctot=0 if inctot<0

*RESTRICT AGE RANGE
*-----------------------------------------------------------------
keep if age>=18
keep if age<65

*GENERATE SSDI DROP and Drop IF ON SOCIAL SECURITY
*-----------------------------------------------------------------
drop if whyss1==1 & whyss2!=2
drop if whyss1>2 & whyss2!=2
rename incss incdi
label var incdi "Income from SSDI"

*Employed
*-----------------------------------------------------------------
gen employed=1 if incwage>0
replace employed=0 if incwage<=0

*Earnings Quintiles
*-----------------------------------------------------------------
xtile quintile_earnings= incwage, nq(5)

*REMOVING MISSING VALUES
*-----------------------------------------------------------------
replace incssi=0 if incssi==99999
replace incwelfr=0 if incwelfr==99999
replace incwage=0 if incwage>=9999998
replace eitcred=0 if eitcred==9999
replace fedtax=0 if fedtax==999999
replace fedtaxac=0 if fedtaxac==999999
replace incunemp=0 if incunemp>=90000


*SSI income split up between spouses
*-----------------------------------------------------------------------------
gen SSI_spouse_equal  = incssi-incssi_sp if incssi>0 & incssi_sp>0
sum SSI_spouse_equal  if SSI_spouse_equal==0
sum incssi incssi_sp if SSI_spouse_equal==0
replace incssi= incssi/2 if SSI_spouse_equal==0
sum incssi incssi_sp if SSI_spouse_equal==0

*SNAP
*-----------------------------------------------------------------------------
*Fraction of the year on SNAP
gen fraction_yr_snap= stampmo/12
replace fraction_yr_snap=1 if fraction_yr_snap==0

*Each individual in the household is reported to have the same value of food stamps.
*This value must be broken up in equal part between each adult member of the household. Each
child should have a reported food stamp value of zero.

gen over18=1 if age>=18
replace over18=0 if over18==.
egen numover18= sum(over18), by (serial)

replace stampval=0 if stampval>=9996

gen incsnap = (stampval/numover18)
label var incsnap "SNAP Benefits"

*Fraction of the year on TANF
*-----------------------------------------------------------------------------
gen fraction_yr_tanf= mthwelfr/12
replace fraction_yr_tanf=1 if fraction_yr_tanf==0

*Transfer Dummy Variables
*-----------------------------------------------------------------------------
```

```
gen EITC=1 if eitcred>0
replace EITC=0 if EITC==.

gen SNAP=1 if stampval>0
replace SNAP=0 if SNAP==.

gen UI=1 if incunemp>0
replace UI=0 if UI==.

gen TANF=1 if incwelfr>0
replace TANF=0 if TANF==.

gen SSI=1 if incssi>0
replace SSI=0 if SSI==.

gen SSDI=1 if incdi>0
replace SSDI=0 if incdi==0

*Inctransfer
*-------------------------------------------------------------------------------

gen inctransfer = incssi+incwelfr+incsnap+eitcred+incdi+incunemp+actccrd
label var inctransfer "Total Transfer Income"

gen transfer = 1 if inctransfer>0
replace transfer=0 if inctransfer==0
label var transfer "Transfer Income Recipient"

gen inctransfer_noeitc = incssi+incwelfr+incsnap+incdi+incunemp
label var inctransfer_noeitc "Total Transfer Income, Excluding EITC"

*RACE
*-------------------------------------------------------------------
*Drop other_race and add those races to existing groups. This allows the earnings regression to
work.

gen hispanic = 1 if hispan>1
gen white=1 if race==100 | race>=810 | race==804
gen black=1 if race==200 | race>=805 | race<=807
gen american_indian=1 if race==300 | race==802 | race==808
gen asian=1 if race==651 | race==803 | race==809

replace hispanic = 0 if hispan==0
replace white=0 if race!=100 | hispanic==1
replace black=0 if race!=200 | hispanic==1
replace american_indian=0 if race!=300 | hispanic==1
replace asian=0 if race !=651 | hispanic==1

replace hispanic = 0 if hispan==0
replace white=0 if race!=100 | hispanic==1 | white==.
replace black=0 if race!=200 | hispanic==1 | black==.
replace american_indian=0 if race!=300 | hispanic==1 | american_indian==.
replace asian=0 if race !=651 | hispanic==1 | asian==.

*AGE
*-------------------------------------------------------------------
gen age_squared = age^2


*COHABITATING PARTNERS
*-------------------------------------------------------------------
gen CohabitingPartner=1 if pecohab>0
replace CohabitingPartner=0 if pecohab==0


*MARTIAL STATUS --- IF COHABITATING WILL BE CONSIDERED MARRIED
*-------------------------------------------------------------------
gen married=1 if marst<=2
replace married=1 if CohabitingPartner==1
replace married=0 if married!=1
```

```
*Sex
*-------------------------------------------------------------------
gen male=1 if sex==1
replace male=0 if sex==2
gen female=1 if sex==2
replace female=0 if sex==1

gen female_married=female*married


*EDUCATION
*-------------------------------------------------------------------
gen no_highschool = 1 if educ<40 & educ>0
replace no_highschool= 0 if no_highschool==.

gen high_school_no_diploma = 1 if educ<73 & educ>=40
replace high_school_no_diploma = 0 if high_school_no_diploma==.

gen high_school_diploma = 1 if educ==73
replace high_school_diploma= 0 if high_school_diploma==.

gen some_college_but_no_degree =1 if educ==81
replace some_college_but_no_degree = 0 if some_college_but_no_degree==.

gen associates = 1 if educ==92
replace associates = 0 if associates==.

gen bachelors= 1 if educ==111
replace bachelors=0 if bachelors==.

gen masters=1 if educ==123
replace masters=0 if master==.

gen Professional_Degree = 1 if educ==124
replace Professional_Degree = 0 if Professional_Degree==.

gen PhD = 1 if educ==125
replace PhD=0 if PhD==.

gen college=1 if educ>81
replace college=0 if college==.

*CHILDREN
*-------------------------------------------------------------------
gen children_under_5=nchlt5
gen children_over_5= nchild - children_under_5


*Interactions
*-------------------------------------------------------------------
gen married_child_under_5 = married* children_under_5
gen married_child_over_5 = married* children_over_5

gen welfare_children_under_5 =incwelfr * children_under_5
gen welfare_children_over_5 =incwelfr * children_over_5

gen ssi_children_under_5 =incssi * children_under_5
gen ssi_children_over_5 =incssi * children_over_5

* HOURS WORKED AND WAGE RATE
*-------------------------------------------------------------------------------------------------------
------------------
*Generate Annual Hours worked.
gen annualhours=uhrswork*wkswork1
label var annualhours "Annual Hours Worked"

*wage rate
gen wagerate = incwage/annualhours

*Percentage of Poverty line
*-------------------------------------------------------------------------------
```

```
gen half_poverty=1 if incwage<offcutoff/2
label var half_poverty "50% Povertly Line"

gen poverty_100=1 if incwage>=offcutoff/2 & incwage<offcutoff
label var poverty_100 "100% Povertly Line"

gen poverty_150=1 if incwage>=offcutoff & incwage<offcutoff*1.5
label var poverty_150 "150% Povertly Line"

gen poverty_200=1 if offcutoff*2 & incwage>=offcutoff*1.5
label var poverty_200 "200% Povertly Line"

gen tanf_tax_rate = 11.4 if half_poverty==1
replace tanf_tax_rate =18.9 if poverty_100==1
replace tanf_tax_rate =51.1 if poverty_150==1
replace tanf_tax_rate =47.33 if poverty_200==1

*Set Potential UI WBA
*-------------------------------------
replace wksunem1=0 if wksunem1==99

gen weeks_employed = 52 - wksunem1
label var weeks_employed "Number of weeks employed"

gen weekly_wage = incwage/weeks_employed
label var weekly_wage "Weekly Wages and Salaries"

gen wba_divisor_to_approx = divisor_bpw/52
label var wba_divisor_to_approx "Divisor to find WBA using weekly income"

gen wba_weekly_wage = weekly_wage/wba_divisor_to_approx
label var wba_weekly_wage "WBA using weekly wages, no min no max"

gen max_wba=incwage/divisor_bpw
label var max_wba "Max WBA"

* Min and  Max WBA
replace max_wba = min_high_wba if max_wba<min_high_wba
replace max_wba = max_high_wba if max_wba>max_high_wba

*-------------------------------------
*Determining whether qualifies for UI
*-------------------------------------
*ELIGIBILITY USING MINIMUM BPW
gen eligible_UI=0
label var eligible_UI "Eligible by number of weeks employed or by earned income"
replace eligible_UI=1 if incwage>=bp_min__det_elig

*Annual Max UI income
*----------------------------------
gen max_UI = max_wba*26
replace max_UI = 0 if eligible==0

*-------------------------------------------------------------------------
*QUINTILES OF DIFFERENT POTENTIAL WBA AND TOTAL BENEFIT AMOUNT IF ELIGIBLE
*-------------------------------------------------------------------------
*Low min and low max
mean max_wba max_UI[iw=wtsupp], over(quintile_earnings)

*-------------------------------------------------------------------------------
*SEE WHAT SHARE OF FIRST WEEK UI RECIPIENTS QUALIFY FOR UI TO CHECK THE QUALITY OF THESE THREE
CRITERION
*-------------------------------------------------------------------------------

gen weeks_on_UI = incunemp/max_wba
label var weeks_on_UI "Number of weeks on UI using estimated WBA without min or max"
mean weeks_on_UI [iw=wtsupp] if weeks_on_UI>0
sum weeks_on_UI if weeks_on_UI>0

*-------------------------------------------------------------------------------
*Income Effects
```

```
*-------------------------------------------------------------------------------
xtile decile_income= inctot, nq(10)
label var decile_income "Income Deciles"

gen net_benefit=521 if decile_income==1
replace net_benefit=384 if decile_income==2
replace net_benefit=286 if decile_income==3
replace net_benefit=185 if decile_income==4
replace net_benefit=115 if decile_income==5
replace net_benefit=34 if decile_income==6
replace net_benefit=-60 if decile_income==7
replace net_benefit=-178 if decile_income==8
replace net_benefit=-390 if decile_income==9
replace net_benefit=-897 if decile_income==10

*Elasticities
xtile quintile_income= inctot, nq(5)

gen cbo_inc = -.1 if married==1 & female==1 & quintile_income==1
replace cbo_inc = -.1 if married==1 & female==1 & quintile_income==2
replace cbo_inc = -.05 if married==1 & female==1 & quintile_income==3
replace cbo_inc = -.01 if married==1 & female==1 & quintile_income==4
replace cbo_inc = -.01 if married==1 & female==1 & quintile_income==5

replace cbo_inc = -.1 if (male==1 | (female==1 & married==0))& quintile_income==1
replace cbo_inc = -.1 if  (male==1 | (female==1 & married==0)) & quintile_income==2
replace cbo_inc = -.05 if  (male==1 | (female==1 & married==0)) & quintile_income==3
replace cbo_inc = -.01 if  (male==1 | (female==1 & married==0)) & quintile_income==4
replace cbo_inc = -.01 if  (male==1 | (female==1 & married==0)) & quintile_income==5

gen cbo_part = .1 if (male==1 | (female==1 & married==0))& quintile_income==1
replace cbo_part = .1 if  (male==1 | (female==1 & married==0)) & quintile_income==2
replace cbo_part = .05 if  (male==1 | (female==1 & married==0)) & quintile_income==3
replace cbo_part = .01 if  (male==1 | (female==1 & married==0)) & quintile_income==4
replace cbo_part = .01 if  (male==1 | (female==1 & married==0)) & quintile_income==5

replace cbo_part = .3 if  married==1 & female==1 & quintile_income==1
replace cbo_part = .3 if married==1 & female==1 & quintile_income==2
replace cbo_part = .2 if married==1 & female==1 & quintile_income==3
replace cbo_part = .1 if married==1 & female==1 & quintile_income==4
replace cbo_part = .1 if married==1 & female==1 & quintile_income==5

*Income Effects
gen income_effects = (net_benefit/inctot)*annualhours*cbo_inc
label var income_effects "Income Effects"

gen change_earnings_income = income_effects * wagerate
label var change_earnings_income "Change in Earnings due to Income Effect"

total change_earnings_income [iw=wtsupp]

gen net_change_income = change_earnings_income + net_benefit
label var net_change_income "Net Change in Income"

*Intensive Table
mean inctot [iw=wtsupp] if employed==1, over (decile_income)
mean income_effects [iw=wtsupp] if employed==1, over (decile_income)
mean annualhours [iw=wtsupp] if employed==1, over (decile_income)
mean change_earnings_income [iw=wtsupp] if employed==1, over (decile_income)
total change_earnings_income [iw=wtsupp] if employed==1, over (decile_income)
mean incwage [iw=wtsupp] if employed==1, over (decile_income)
mean net_change_income [iw=wtsupp] if employed==1, over (decile_income)

gen new_hours_worked = annualhours+income_effects
replace new_hours_worked=0 if new_hours_worked<0
mean new_hours_worked [iw=wtsupp] if employed==1, over (decile_income)

gen dif_hours_worked = new_hours_worked-annualhours
replace dif_hours_worked=0 if dif_hours_worked<0
mean dif_hours_worked [iw=wtsupp] if employed==1, over (decile_income)
```

```
*Extensive Margin
reg inctransfer_noeitc age  age_squared white  american_indian black asian  hispanic
no_highschool high_school_no_diploma high_school_diploma some_college_but_no_degree associates
bachelors masters Professional_Degree PhD children_under_5 children_over_5  married_child_under_5
married_child_over_5 female_married  [iw=earnwt] if incwage==0 & inctransfer_noeitc>0, nocons
predict potential_inctransfer_pre
replace potential_inctransfer_pre=(potential_inctransfer_pre/2)+max_UI if incwage>0 & max_UI>0

reg employed age age_squared white  american_indian black asian  hispanic  no_highschool
high_school_no_diploma high_school_diploma some_college_but_no_degree associates bachelors
masters Professional_Degree PhD children_under_5 children_over_5  married_child_under_5
married_child_over_5 female_married  [iw=earnwt], nocons
predict LFP_pre

reg inctot age  age_squared white  american_indian black asian  hispanic  no_highschool
high_school_no_diploma high_school_diploma some_college_but_no_degree associates bachelors
masters Professional_Degree PhD children_under_5 children_over_5  married_child_under_5
married_child_over_5 female_married  [iw=earnwt] if incwage>0, nocons
predict inctot_predict

reg change_earnings_income age  age_squared white  american_indian black asian  hispanic
no_highschool high_school_no_diploma high_school_diploma some_college_but_no_degree associates
bachelors masters Professional_Degree PhD children_under_5 children_over_5  married_child_under_5
married_child_over_5 female_married  [iw=earnwt] if incwage>0, nocons
predict change_earnings_income_predict

gen extensive = (cbo_part*(LFP_pre/(inctot_predict-
potential_inctransfer_pre)))*(change_earnings_income_predict)
mean extensive [iw=wtsupp], over(decile_income)

*Summary Stats Table
mean LFP_pre employed potential_inctransfer_pre inctransfer_noeitc inctot_predict inctot
change_earnings_income change_earnings_income_predict [iw=wtsupp], over(decile_income)

*Change in Employment and National Income
gen change_output = extensive *incwage
mean change_output extensive [iw=wtsupp], over(decile_income)
total change_output extensive [iw=wtsupp], over(decile_income)
mean change_output extensive [iw=wtsupp]
total change_output extensive [iw=wtsupp]


xtile decile_incwage= incwage, nq(10)
mean change_output extensive [iw=wtsupp], over(decile_incwage)
total change_output extensive [iw=wtsupp], over(decile_incwage)

mean incwage[iw=wtsupp], over(decile_incwage)
total incwage [iw=wtsupp], over(decile_incwage)
```