

```

cap cd "Z:\\"
* Make sure cutedpdf is the default printer
cap log close
log using Gallup_micro.log, text replace
clear
set mem 2200m
set more off
use gallup_work_inc.dta, clear /*Don't use the original "file.dta" provided by Gallup */
replace age=-1 if age==.
gen age_missing=(age===-1)
replace sex=-1 if sex==.
gen sex_missing=(sex===-1)
for X in num 1/4: gen ageX=age^X \ gen fageX=(sex==2)*ageX
global controls "age1-age4 fage1-fage4 sex age_missing sex_missing"
global sat "sat_current sat_past sat_future"
global feelings "enjoyment pain worry sadness boredom depression anger love"
global yesterday "moredays wellrested respect choose smile proud lean goodfood"

* Table 1
oprobit sat_current lgdp [pw=wt], robust cluster(cty)
oprobit sat_current lgdp $controls [pw=wt], robust cluster(cty)

* Table 2
xi: oprobit sat_current ln_inc i.cty [pw=wt], robust
xi: oprobit sat_current ln_inc $controls i.cty [pw=wt], robust
set matsize 2000
xi: reg ln_inc i.cty*i.educ $controls [pw=wt], robust
predict ln_inc_hat2
predict ln_inc_res2, res
xi: oprobit sat_current ln_inc_hat2 ln_inc_res2 $controls i.cty [pw=wt], robust

* Table 6
xi i.cty
foreach v of global feelings {
    summ feelings_`v' if lgdp~= . [w=wt]
    probit feelings_`v' lgdp $controls [pw=wt], robust cluster(cty)
    probit feelings_`v' ln_inc $controls _I* [pw=wt], robust
}
foreach v of global yesterday {
    summ yesterday_`v' if lgdp~= . [w=wt]
    probit yesterday_`v' lgdp $controls [pw=wt], robust cluster(cty)
    probit yesterday_`v' ln_inc $controls _I* [pw=wt], robust
}

* Appendix table 1
xi: oprobit sat_current i.cty [pw=wt]
predict hat if sat_current~= ., xb
summ hat [w=wt]
local mn=r(mean)
global range=10
range z -$range $range 1000
gen fz=normalden(z)
gen zfz=z*fz
gen sat_val=.
qui foreach i of numlist 0/10 {
    local j=`i'+1

    if `i'==0 {
        local lower= -$range
        local upper=_b[/cut1]
    }
    else if `i'==10 {
        local lower=_b[/cut10]
        local upper=$range
    }
    else {

```

```

        local lower=_b[/cut`i']
        local upper=_b[/cut`j']
    }
    local lower=`lower'-'`mn'
    local upper=`upper'-'`mn'
    integ fz z if z<`upper' & z>`lower'
    local fz=r(integral)
    integ zفز z if z<`upper' & z>`lower'
    local zفز=r(integral)
    local m=`zفز'/`fz'
    replace sat_val=`m' if sat_current==`i'
    di ("Cutpoint `i' ranges from `lower' to `upper' and has mean = `m'")
}
summ sat_val [w=wt]
summ sat_current [w=wt]
gen z_sat_current=(sat_current-r(mean))/r(sd)
table sat_current, c(m sat_val m z_sat_current)

table cty sat_current, c(sum wt) format(%9.2f) cseewidth(1)
xi: oprobit sat_current i.cty [pw=wt]
predict sat_hat if sat_current~., xb
table cty, c(n wt m sat_hat m sat_current) format(%9.2f) cseewidth(1)
sort cty
gen str1 x="XXX"
list cty country x countryname x wp5 if t==1, clean
drop sat_hat

*** Figure 10
for X in any b se n rsq: gen X_inc=.
levelsof cty, local(countries)
foreach c of local countries {
    display("Country: `c'")
    summ inc if cty=="`c'"
    if r(N)>0 {
        oprobit sat_current ln_inc $controls [pw=wt] if cty=="`c'", robust
        replace b_inc=_b[ln_inc] if cty=="`c'"
        replace se_inc=_se[ln_inc] if cty=="`c'"
        replace n_inc=e(N) if cty=="`c'"
        replace rsq_inc=e(r2_p) if cty=="`c'"
    }
}
table cty, c(m b_inc m se_inc m rsq_inc)

preserve
keep if t==1
keep b_inc se_inc cty
egen brank=rank(b_inc)
gen b05=round(b_inc+.025,.05)-.025
egen brank2=rank(brank), by(b05)
sort brank2
summ b_inc, det
local m=round(r(mean),.01)
local m2=`m'-.07
#delimit ;
twoway
    (bar brank2 b05, barwidth(.05) base(-.2) fintensity(50) fcolor(ltblue))
    (scatter brank2 b05, mlabel(cty) msymbol(none) mlabpos(6) mlabgap(.00) mlabsize
(vsmall) mlabcolor(black))
,
    ytitle("Number of countries", size(small))
    ylabel(0(2)18, angle(horizontal))
    xtitle("Estimated life satisfaction-income gradient")
    xlabel(-.1(0.1)0.9)
    xline(`m', lcolor(black) lwidth(thick))
    text(17.5 `m2' "Mean=`m'")
    legend(off)

```

```

        xsize(10) ysize(7.5)
        name(fig10, replace)
;
#delimit cr
graph save fig10, replace
graph export fig10.eps, fontface(Times) replace
graph export fig10.tif, replace

summ b_inc, det
summ se_inc, det
restore

* Figure 11
gen ylow=exp(lgdp-.25) if b_inc~= .
gen yhigh=exp(lgdp+.25) if b_inc~= .
xi: oprobit sat_current i.cty [pw=wt]
predict sat_current_hat if sat_current~= ., xb
summ sat_current_hat if t==1
replace sat_current_hat=sat_current_hat-r(mean)
gen satlow=sat_current_hat-b_inc*.25
gen sathigh=sat_current_hat+b_inc*.25

reg sat_current_hat lgdp if t==1
predict sat_current_fit

#delimit ;
twoway
    (line sat_current_fit gdp, sort lpattern(longdash) lcolor(gray) lwidth(thick))
    (scatter sat_current_hat gdp, mlabel(cty) mlabpos(6) mcolor(navy) mlabcolor(navy))
    (pcarrow satlow ylow sathigh yhigh, lcolor(black) mcolor(black))
    if t==1
,
    xscale(log)
    xlabel(500 ".5" 1000 "1" 2000 "2" 4000 "4" 8000 "8" 16000 "16" 32000 "32")
    xtitle("Real GDP per capita (thousands of dollars, log scale)", size(small))
    ytitle("Life satisfaction (Ordered probit index)", size(small))
    ylabel(-1.5(0.5)1.5, angle(horizontal) format(%4.1f))
    legend (
        order(2 "Country-year aggregates" 3 "Within-country wellbeing gradient" 1
"Between-country wellbeing gradient")
        rowgap(tiny) margin(tiny) bmargin(tiny)
        region(fcolor(none))
        size(small)
        rows(3) pos(11) ring(0)
    )
    xsize(10) ysize(7.5)
    name(fig11, replace)
;
#delimit cr
graph save fig11, replace
graph export fig11.eps, fontface(Times) replace
graph export fig11.tif, replace

* Figure 9
clear
set mem 2000m
set more off
/* use gallup_work_inc.dta */ *Don't use the original "file.dta" provided by Gallup */
egen group=group(ln_inc cty)
set matsize 2000
xi: oprobit sat_current i.group [pw=wt]
predict sat_inc if ln_inc~= . & sat_current~= ., xb
xi: reg sat_inc i.cty [pw=wt]
predict sat_cty

```

```

gen relsat=sat_inc-sat_cty
xi: reg ln_inc i.group [pw=wt]
predict lninc_inc if ln_inc~= . & sat_current~= .
xi: reg ln_inc i.cty [pw=wt]
predict lninc_cty
gen relinc=lninc_inc-lninc_cty
egen n=count(sat_current+ln_inc), by(group)
egen t_group=tag(group)
    This prevents having to rerun the code above*/
use temp_fig9data.dta, clear
reg relsat relinc if t_group==1 & abs(relinc)<4 [fw=n], robust
predict relsat_hat if relsat~= . & relinc~= .
global a=string(_b[_cons], "%4.3f")
global b=string(_b[relinc], "%4.3f")
global se=string(_se[relinc], "%4.3f")
#delimit ;
twoway
    (scatter relsat relinc [fw=n], msymbol(circle_hollow) msize(small) mlwidth(vvvthin)
mcolor(navy))
    (line relsat_hat relinc, sort lcolor(black) lpattern(dash))
    if n>1 & t_group==1 & abs(relinc)<3 & abs(relsat)<1.5
,
    ytitle("Life satisfaction, less country average (Ordered probit index)", size(small)
)
    ylabel(-1.5(.5)1.5, angle(horizontal) format(%4.1f))
    xtitle("Log(Household Income), less country average", size(small))
    xlabel(-3(1)3)
    legend(off)
    note("Relative well-being = $b * relative income [se=$se]", ring(0) pos(5))
    xsize(10) ysize(7.5)
    name(fig9, replace)
;
#delimit cr
graph save fig9, replace
graph export fig9.eps, fontface(Times) replace
graph export fig9.tif, replace
save temp_fig9data, replace

```

* Figure 22

```

xi i.cty
foreach v of global feelings {
    display("Feeling: `v'")
    probit feelings_`v' _I* [pw=wt], robust cluster(cty)
    predict feelings_`v'_yes
    probit feelings_`v' lgdp [pw=wt], robust cluster(cty)
    predict feelings_`v'_yes_fit
    levelsof cty if feelings_`v'~=., local(ctys)
    gen b_`v'=.
    foreach c of local ctys {
        summ feelings_`v' if cty=="`c'" & ln_inc~= .
        if r(N)>0 {
            dprobit feelings_`v' ln_inc if cty=="`c'"
            replace b_`v'=_b[ln_inc] if cty=="`c'"
        }
    }
    gen ylow=exp(lgdp-.1) if b_`v'~= .
    gen yhigh=exp(lgdp+.1) if b_`v'~= .
    gen fellow=feelings_`v'_yes-b_`v'*.1
    gen feelhigh=feelings_`v'_yes+b_`v'*.1

    local lb: var lab feelings_`v'_yes
    local lb=subinstr("`lb'", "Percent Feeling ", "", 1)
    lowess feelings_`v'_yes lgdp if t==1, nograph gen(`v'_yes_low)
#delimit ;
twoway
    (scatter feelings_`v'_yes gdp, mlabpos(0) msymbol(circle) mcolor(navy))
    (line feelings_`v'_yes_fit gdp, sort lpattern(dash) lcolor(black))

```

```

        (pcarrow fellow ylow feelhigh yhigh, lcolor(cranberry) mcolor(cranberry)
lwidth(vvthin))
        if t==1
        ,
            title("`lb'", ring(0))
            xtick(500 1000 2000 4000 8000 16000 32000)
            xlabel(500 "0.5" 2000 "2" 8000 "8" 32000 "32")
            xtitle("")
            xscale(log)
            legend(off)
            ylabel(0(.2).8, angle(horizontal))
            ytick(0(.2)1)
            xsize(10) ysize(7.5)
            name(`v', replace)
        ;
        #delimit cr
        drop ylow yhigh fellow feelhigh
    }

#delimit ;
graph combine $feelings,
    imargin(zero)
    rows(2)
    bltitle("Real GDP per capita (thousands of dollars, log scale)", size(small))
    lltitle("Percent reporting indicated feeling", size(small))
    xsize(10) ysize(7.5)
    name(fig22_micro, replace)
;
#delimit cr
graph save fig22_micro, replace
graph export fig22_micro.eps, fontface(Times) replace
graph export fig22_micro.tif, replace

* Figure 23
xi i.cty
foreach v of global yesterday {
    display("Feeling: `v'")
    probit yesterday_`v' _I* [pw=wt], robust cluster(cty)
    predict yesterday_`v'_yes
    probit yesterday_`v' lgdp [pw=wt], robust cluster(cty)
    predict yesterday_`v'_yes_fit
    levelsof cty if yesterday_`v'~=., local(ctys)
    gen b_`v'=.
    foreach c of local ctys {
        summ yesterday_`v' if cty=="`c'" & ln_inc~=.
        if r(N)>0 {
            dprobit yesterday_`v' ln_inc if cty=="`c'"
            replace b_`v'=_b[ln_inc] if cty=="`c'"
        }
    }
    gen ylow=exp(lgdp-.1) if b_`v'~=.
    gen yhigh=exp(lgdp+.1) if b_`v'~=.
    gen fellow=yesterday_`v'_yes-b_`v'*.1
    gen feelhigh=yesterday_`v'_yes+b_`v'*.1

    local lb: var lab yesterday_`v'_yes
    local lb=subinstr("`lb'", "Percent Feeling ", "", 1)
    local lbwrds=wordcount("`lb'")
    local lbwrds2=int(`lbwrds'/2)
    foreach i of numlist 1/`lbwrds2' {
        local lb1=""lb1'+word("`lb'", `i')+ " "
    }
    local lbwrds2=`lbwrds2'+1
    foreach i of numlist `lbwrds2'/`lbwrds' {
        local lb2=""lb2'+word("`lb'", `i')+ " "
    }
    lowess yesterday_`v'_yes lgdp if t==1, nograph gen(`v'_yes_low)

```

```

#delimit ;
twoway
    (scatter yesterday_`v'_yes gdp, mlabpos(0) msymbol(circle) mcolor(navy))
    (line yesterday_`v'_yes_fit gdp, sort lpattern(dash) lcolor(black))
    (pcarrow fellow ylow feelhigh yhigh, lcolor(cranberry) mcolor(cranberry)
lwidth(vvthin))
    if t==1
,
    title("`lb1'" "`lb2'", size(medsmall) span)
    xtick(500 1000 2000 4000 8000 16000 32000)
    xlabel(500 "0.5" 2000 "2" 8000 "8" 32000 "32")
    xtitle("")
    xscale(log)
    legend(off)
    ylabel(0(.2).8, angle(horizontal))
    ytick(0(.2)1)
    xsize(10) ysize(7.5)
    name(`v', replace)
;
#delimit cr
drop ylow yhigh fellow feelhigh
local lb1=""
local lb2=""
}

#delimit ;
graph combine $yesterday,
    imargin(zero)
    rows(2)
    bltitle("Real GDP per capita (thousands of dollars, log scale)", size(small))
    lltitle("Percent reporting indicated feeling", size(small))
    xsize(10) ysize(7.5)
    name(fig23_micro, replace)
;
#delimit cr
graph save fig23_micro, replace
graph export fig23_micro.eps, fontface(Times) replace
graph export fig23_micro.tif, replace

* Appendix Figure A.3
cap program drop labgraph
program define labgraph
    syntax varlist [if]
    marksample touse
    correl `1' `2' `if'
    global rho=string(r(rho),"%4.3f")
    reg `1' `2' `if'
    global b=string(_b[`2'], "%3.2f")
    global a=string(_b[_cons], "%3.2f")
    global se=string(_se[`2'], "%3.2f")
end

use gallup_work_inc.dta, clear
* Ordered probit
xi: oprobit sat_current i.cty [pw=wt]
predict sat_hat if sat_current~=., xb
egen av=mean(sat_hat), by(cty)
replace sat_hat=av
drop av
egen tag=tag(cty)
summ sat_hat if tag==1
replace sat_hat=sat_hat-r(mean)

* Simple mean
xi: reg sat_current i.cty [pw=wt]
predict sat_hat_simple if sat_current~=., xb
egen av=mean(sat_hat_simple), by(cty)

```

```

replace sat_hat_simple=av
drop av
lowess sat_hat_simple sat_hat if tag==1, nograph generate(sat_hat_simple_low)
labgraph sat_hat_simple sat_hat if tag==1
#delimit ;
twoway
    (scatter sat_hat_simple sat_hat, mcolor(navy))
    (line sat_hat_simple_low sat_hat, sort lpattern(dot) lcolor(cranberry))
    (lfit sat_hat_simple sat_hat, sort lpattern(longdash) lcolor(black))
    if tag==1
,
    title("Simple mean (0-10 response)", ring(0))
    xtitle("")
    xlabel(-1.5(.5)1.5, format(%4.1f))
    ylabel(2(1)9, angle(horizontal))
    legend(off)
    note("y= $a+$b*x [se=$se]" "Correlation=$rho", ring(0) pos(5))
    xsize(10) ysize(7.5)
    name(mean, replace)
;
#delimit cr

*Ordered logit
xi: ologit sat_current i.cty [pw=wt]
predict sat_hat_logit if sat_current~=. , xb
egen av=mean(sat_hat_logit), by(cty)
replace sat_hat_logit=av
drop av
summ sat_hat_logit if tag==1
replace sat_hat_logit=sat_hat_logit-r(mean)
lowess sat_hat_logit sat_hat if tag==1, nograph generate(sat_hat_logit_low)
labgraph sat_hat_logit sat_hat if tag==1
#delimit ;
twoway
    (scatter sat_hat_logit sat_hat, mcolor(navy))
    (line sat_hat_logit_low sat_hat, sort lpattern(dot) lcolor(cranberry))
    (lfit sat_hat_logit sat_hat, sort lpattern(longdash) lcolor(black))
    if tag==1
,
    title("Ordered logit", ring(0))
    ylabel(-3(1)3, angle(horizontal))
    xtitle("")
    xlabel(-1.5(.5)1.5, format(%4.1f))
    legend(off)
    note("y= $a+$b*x [se=$se]" "Correlation=$rho", ring(0) pos(5))
    xsize(10) ysize(7.5)
    name(ologit, replace)
;
#delimit cr

gen vsat=100*(sat_current>7)
xi: reg vsat i.cty [pw=wt]
predict vsat_hat if vsat~=.
egen av=mean(vsat), by(cty)
replace vsat=av
drop av
lowess vsat_hat sat_hat if tag==1, nograph generate(vsat_hat_low)
labgraph vsat_hat sat_hat if tag==1
reg vsat_hat sat_hat if tag==1
predict vsat_hat_hat
#delimit ;
twoway
    (scatter vsat_hat sat_hat, mcolor(navy))
    (line vsat_hat_low sat_hat, sort lpattern(dot) lcolor(cranberry))
    (line vsat_hat_hat sat_hat if vsat_hat_hat>0, sort lpattern(longdash) lcolor(black))
    if tag==1
,
    title("Percent reporting satisfaction > 7", ring(0))

```

```

        ylabel(0(20)80, angle(horizontal))
        ytick(100, notick)
        xtitle("")
        ytitle("")
        xlabel(-1.5(.5)1.5, format(%4.1f))
        legend(off)
        note("y= $a+$b*x [se=$se]" "Correlation=$rho", ring(0) pos(5))
        xsize(10) ysize(7.5)
        name(pc7, replace)
;
#delimit cr

cap program drop myoprobit0to10
program define myoprobit0to10
    version 6
    args lnf theta sd c1 c2 c3 c4 c5 c6 c7 c8 c9 c10
    local c0=-9999999
    local c11=9999999
    quietly replace `lnf'=ln(normprob((`theta'-'c0')/exp(`sd')))-normprob((`theta'-'
`c1')/exp(`sd')) if $ML_y1==0
    quietly replace `lnf'=ln(normprob((`theta'-'c1')/exp(`sd')))-normprob((`theta'-'
`c2')/exp(`sd')) if $ML_y1==1
    quietly replace `lnf'=ln(normprob((`theta'-'c2')/exp(`sd')))-normprob((`theta'-'
`c3')/exp(`sd')) if $ML_y1==2
    quietly replace `lnf'=ln(normprob((`theta'-'c3')/exp(`sd')))-normprob((`theta'-'
`c4')/exp(`sd')) if $ML_y1==3
    quietly replace `lnf'=ln(normprob((`theta'-'c4')/exp(`sd')))-normprob((`theta'-'
`c5')/exp(`sd')) if $ML_y1==4
    quietly replace `lnf'=ln(normprob((`theta'-'c5')/exp(`sd')))-normprob((`theta'-'
`c6')/exp(`sd')) if $ML_y1==5
    quietly replace `lnf'=ln(normprob((`theta'-'c6')/exp(`sd')))-normprob((`theta'-'
`c7')/exp(`sd')) if $ML_y1==6
    quietly replace `lnf'=ln(normprob((`theta'-'c7')/exp(`sd')))-normprob((`theta'-'
`c8')/exp(`sd')) if $ML_y1==7
    quietly replace `lnf'=ln(normprob((`theta'-'c8')/exp(`sd')))-normprob((`theta'-'
`c9')/exp(`sd')) if $ML_y1==8
    quietly replace `lnf'=ln(normprob((`theta'-'c9')/exp(`sd')))-normprob((`theta'-'
`c10')/exp(`sd')) if $ML_y1==9
    quietly replace `lnf'=ln(normprob((`theta'-'c10')/exp(`sd')))-normprob((`theta'-'
`c11')/exp(`sd')) if $ML_y1==10
end

xi i.cty
ml model lf myoprobit0to10 (sat_current=_I*, noconst) (sat_current=_I*, noconst) /c1 /c2
/c3 /c4 /c5 /c6 /c7 /c8 /c9 /c10 [pw=wt]
local initmu
local initsd
foreach v of varlist _I* {
    local initmu `initmu' 0
    local initsd `initsd' 0
}
local lengthmu: list sizeof initmu
local lengthsd: list sizeof initsd
di("Length of mu: `lengthmu' Length of SD: `lengthsd'")
ml init `initmu' `initsd' 1 2 3 4 5 6 7 8 9 10, copy
ml max
gen sat_oprobit_mu=.
gen sat_oprobit_sd=.
gen sat_oprobit_mu_se=.
gen sat_oprobit_sd_se=.

qui foreach v of varlist _I* {
    replace sat_oprobit_mu=[eq1]_b[`v'] if `v'==1
    replace sat_oprobit_mu_se=[eq1]_se[`v'] if `v'==1
    replace sat_oprobit_sd=exp([eq2]_b[`v']) if `v'==1
    replace sat_oprobit_sd_se=[eq2]_se[`v'] if `v'==1 /* Fix this*/
}
for X in var sat_oprobit_mu sat_oprobit_mu_se sat_oprobit_sd sat_oprobit_sd_se: egen

```



```

av=mean(X), by(cty) \ replace X=av \ drop av

summ sat_oprobit_mu if tag==1
replace sat_oprobit_mu=sat_oprobit_mu-r(mean)
lowess sat_oprobit_mu sat_hat if tag==1, nograph generate(sat_oprobit_mu_low)
labgraph sat_oprobit_mu sat_hat if tag==1
#delimit ;
twoway
    (scatter sat_oprobit_mu sat_hat, mcolor(navy))
    (line sat_oprobit_mu_low sat_hat, sort lpattern(dot) lcolor(cranberry))
    (lfit sat_oprobit_mu sat_hat, sort lpattern(longdash) lcolor(black))
    if tag==1
,
    title("Heteroscedastic ordered probit index", ring(0))
    ylabel(-2(1)2, angle(horizontal))
    ytick(2.5, notick)
    xtitle("")
    xlabel(-1.5(.5)1.5, format(%4.1f))
    legend(off)
    note("y = $a+$b*x [se=$se]" "Correlation=$rho", ring(0) pos(5))
    xsize(10) ysize(7.5)
    name(hoprobit, replace)
;
#delimit cr

#delimit ;
graph combine
    mean ologit pc7 hoprobit,
    xcommon
    bltitle("Life satisfaction, ordered probit index", size(small))
    rows(2)
    imargin(zero)
    xsize(10) ysize(7.5)
    name(figA3, replace)
;
#delimit cr
graph save figA3_micro, replace
graph export figA3_micro.eps, fontface(Times) replace
graph export figA3_micro.tif, replace

log close

```

```

*** Below this is working paper stuff
/*
* Figures
for X in any b se n rsq: for Y in num 1/2: gen XY_inc=.
levelsof cty, local(countries)
foreach c of local countries {
    display("Country: `c'")
    summ inc if cty=="`c'"
    if r(N)>0 {
        oprobit sat_current ln_inc [pw=wt] if cty=="`c'", robust
        replace b1_inc=b[ln_inc] if cty=="`c'"
        replace se1_inc=_se[ln_inc] if cty=="`c'"
        replace n1_inc=e(N) if cty=="`c'"
        replace rsq1_inc=e(r2_p) if cty=="`c'"
        oprobit sat_current ln_inc $controls [pw=wt] if cty=="`c'", robust
        replace b2_inc=b[ln_inc] if cty=="`c'"
        replace se2_inc=_se[ln_inc] if cty=="`c'"
    }
}

```

```

        replace n2_inc=e(N) if cty=="`c'"
        replace rsq2_inc=e(r2_p) if cty=="`c'"
    }
}
table cty, c(m b1_inc m sel_inc m rsq1_inc m b2_inc m se2_inc)

xi i.cty
qui foreach v of varlist sat_* feelings_* yesterday_* {
    desc `v'
    oprobit `v' lgdp [pw=wt], cluster(cty)
    local crosscountry=round(_b[lgdp],.001)
    local secrosscountry=round(_se[lgdp],.001)
    oprobit `v' ln_inc _I* [pw=wt]
    local withincountry=round(_b[ln_inc],.001)
    local sewithincountry=round(_se[ln_inc],.001)
    local lb: var label `v'
    summ `v' if ln_inc~= . [aw=wt]
    local mn=round(r(mean),.001)
    local sd=round(r(sd),.001)
    noisily display("Variable `lb' Mean: `mn' SD: `sd' Macro: `crosscountry'
(`secrosscountry') Micro: `withincountry' (`sewithincountry')")
}

gen ylow=exp(lgdp-.25) if b1_inc~= .
gen yhigh=exp(lgdp+.25) if b1_inc~= .
xi: oprobit sat_current i.cty [pw=wt]
predict sat_current_hat if sat_current~=., xb
summ sat_current_hat if t==1
replace sat_current_hat=sat_current_hat-r(mean)
gen satlow=sat_current_hat-b1_inc*.25
gen sathigh=sat_current_hat+b1_inc*.25

reg sat_current_hat lgdp if t==1
predict sat_current_fit

#delimit ;
twoway
    (scatter sat_current_hat gdp, mlabel(cty) mlabpos(6) mcolor(navy) mlabcolor(navy))
    (pcarrow satlow ylow sathigh yhigh)
    (line sat_current_fit gdp, sort lpattern(longdash) lcolor(green) lwidth(thick))
    if t==1
,
    title("Income and Life Satisfaction")
    subtitle("Comparing Within-Country and Between-Country Estimates")
    xscale(log)
    xlabel(500 1000 2000 4000 8000 16000 32000)
    xttitle("Real GDP per Capita at PPP (log scale)")
    ytitle("Life Satisfaction Index (Ordered Probit)")
    ylabel(-1.5(0.5)1, angle(horizontal))
    legend (
        order(1 "Country-year aggregates" 2 "Within-country wellbeing gradient" 3
"Between-country wellbeing gradient")
        rowgap(tiny) margin(tiny) bmargin(tiny)
        region(fcolor(none))
        size(small)
        rows(3) pos(11) ring(0)
    )
    note("Source: Gallup World Poll, 2006")
    xsize(10) ysize(7.5)
    name(sat_micromacro, replace)
    saving(sat_micromacro.gph, replace)
;
#delimit cr

preserve
keep if t==1
keep b1_inc sel_inc cty
egen brank=rank(b1_inc)

```

```

gen b05=round(b1_inc+.025,.05)-.025
egen brank2=rank(brank), by(b05)
sort brank2
summ b1_inc, det
local m=round(r(mean),.01)
#delimit ;
twoway
    (bar brank2 b05, barwidth(.05) base(-.2) fintensity(50) fcolor(ltblue))
    (scatter brank2 b05, mlabel(cty) msymbol(none) mlabpos(6) mlabgap(.00) mlabcolor
(black))
,
    title("Within-Country Life Satisfaction-Income Gradient")
    subtitle("Distribution of Estimates Across Countries")
    ytitle("Number of Countries")
    ylabel(0(4)16, angle(horizontal))
    xtitle("Estimate of Satisfaction-Income Gradient by country" "From country-specific
ordered probit regression of satisfaction on log household income")
    xlabel(-.1(0.1)0.7)
    xline(`m', lcolor(black) lwidth(thick))
    text(17.5 .273 "Mean=`m'")
    legend(off)
    note("Source: Gallup World Poll, 2006")
    xsize(10) ysize(7.5)
    saving(sat_microhisto.gph, replace)
;
#delimit cr

summ b1_inc, det
summ sel_inc, det
restore

clear
set mem 2000m
set more off
use gallup_work_inc.dta /*Don't use the original "file.dta" provided by Gallup */

egen group=group(ln_inc cty)
set matsize 2000
xi: oprobit sat_current i.group [pw=wt]
predict sat_inc if ln_inc~= . & sat_current~= ., xb
xi: reg sat_inc i.cty [pw=wt]
predict sat_cty
gen relsat=sat_inc-sat_cty
xi: reg ln_inc i.group [pw=wt]
predict lninc_inc if ln_inc~= . & sat_current~= .
xi: reg ln_inc i.cty [pw=wt]
predict lninc_cty
gen relinc=lninc_inc-lninc_cty
egen n=count(sat_current+ln_inc), by(group)
egen t_group=tag(group)
reg relsat relinc if t_group==1 & abs(relinc)<4 [fw=n], robust
predict relsat_hat if relsat~= . & relinc~= .
global a=string(_b[_cons], "%4.3f")
global b=string(_b[relinc], "%4.3f")
global se=string(_se[relinc], "%4.3f")
#delimit ;
twoway
    (scatter relsat relinc [fw=n], msymbol(circle_hollow) msize(small) mlwidth(vvvthin))
    (line relsat_hat relinc, sort lcolor(black) lpattern(dash))
    if n>1 & t_group==1 & abs(relinc)<4 & abs(relsat)<2.5
,
    title("Within-Country Relationship Between Well-Being & Income")
    subtitle("Each point aggregates satisfaction in a country*income category", ring(0))
    ytitle("Satisfaction ladder: Ordered probit index"
        "Estimated for each income category*country"
        "Well-being in income category, less country average"
    )
    ylabel(-2(1)2, angle(horizontal))

```

```
xtitle("Log(Household Income), less country average")
xlabel(-4(1)4)
legend(off)
note("Relative well-being = $b * relative income [se=$se]", ring(0) pos(5))
caption("Source: Gallup World Poll, 2006", size(small) span)
xsize(10) ysize(7.5)
name(gallup_micro, replace)

;

log close
*/
```