\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*b1\_shape\_HRS\_and\_geo

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Outline:

\*/

set more off

global b1\_family\_kidlevel\_dta ="R:/Public/Contributions/Rand/RandFamily/2014v1/stata/randhrsfamk1992\_2014v1.dta"

global b1\_family\_reslevel\_dta ="R:/Public/Contributions/Rand/RandFamily/2014v1/stata/randhrsfamr1992\_2014v1.dta"

global b1\_longitudinal\_dta = "R:/Public/Contributions/Rand/RandHrs2018V1/stata/randhrs1992\_2018v1.dta"

global b1\_detail\_impu\_dta = "R:/Public/Contributions/Rand/RandIncWlthImp/stata/randhrsimp1992\_2018v1.dta"

global b1\_cor\_geo\_dta = "R:/Restricted/Geographic Information/xyrDet/stata/hrsxgeo18v8b\_r.dta"

global b1\_child\_geo\_dta = "R:/Restricted/Geographic Information/Czip/Stata/CHILDZIPXWAVE.dta"

if "`1'"!="function calling"{

prog drop \_all

global temp\_dir ="../temp"

global output\_dir="../output"

prog main

cd "U:\2110ChildcareMigration\1Clean\1PreliminaryDataPattern\code"

b1a\_load\_data

b1b\_shape\_cor\_to\_HH

b1b\_merge\_data

end

}

prog b1b\_shape\_cor\_to\_HH

u $temp\_dir/b1a1\_cor\_geo.dta, clear

merge 1:1 hhidpn using $temp\_dir/b1a3c\_rand\_resp\_long.dta, nogen keep(matched)

merge 1:1 hhidpn using $temp\_dir/b1a3b\_rand\_resp.dta, nogen keep(matched)

drop hhid

gen hhid = floor(hhidpn/1000)

egen pn\_gend\_count = nvals(hhidpn), by(hhid ragender)

keep if pn\_gend\_count==1

drop pn\_gend\_count hhidpn

sort hhid ragender

order hhid ragender rgeostateusps7, first

order r12wthh, last

reshape wide rgeostateusps7-r12wthh, i(hhid) j(ragender)

foreach x of varlist \*1 \*2{

disp("`x'")

local len = length("`x'")

local y = substr("`x'",1,`len'-1)

local z = substr("`x'",-1,1)

if "`z'"=="1"{

ren `x' m`y'

}

if "`z'"=="2"{

ren `x' f`y'

}

}

b1b1\_generate\_hh\_variables

save $temp\_dir/b1b\_hh.dta, replace

end

prog b1b1\_generate\_hh\_variables

forvalues x = 3/12{

gen h`x'kdcarehr=max(max(mr`x'kdcarehr,0)+max(ms`x'kdcarehr,0),max(fr`x'kdcarehr,0)+max(fs`x'kdcarehr,0),max(mr`x'kdcarehr,0)+max(fr`x'kdcarehr,0),max(ms`x'kdcarehr,0)+max(fs`x'kdcarehr,0))

}

forvalues x = 7/14{

gen hgeostateusps`x'=frgeostateusps`x'

gen hgeozipcode`x'=frgeozipcode`x'

gen hgeourbrurIII`x'=frgeourbrurIII`x'

gen hgeourbrurXIII`x'=frgeourbrurXIII`x'

replace hgeostateusps`x'=mrgeostateusps`x' if (mrgeozipcode`x'!="" & frgeozipcode`x'=="")

replace hgeozipcode`x'=mrgeozipcode`x' if (mrgeozipcode`x'!="" & frgeozipcode`x'=="")

replace hgeourbrurIII`x'=mrgeourbrurIII`x' if (mrgeozipcode`x'!="" & frgeozipcode`x'=="")

replace hgeourbrurXIII`x'=mrgeourbrurXIII`x' if (mrgeozipcode`x'!="" & frgeozipcode`x'=="")

}

gen haracem=max(fraracem,mraracem)

end

prog b1b\_merge\_data

u $temp\_dir/b1a3a\_rand\_kidresp.dta, clear

merge 1:1 kidid using $temp\_dir/b1a2\_child\_geo.dta, nogen keep(matched)

merge n:1 hhid using $temp\_dir/b1b\_hh.dta, nogen keep(matched)

drop if hasplit==1

drop if mrlink==0

drop if frlink==0

drop hasplit mrlink frlink

foreach x of varlist \_all {

local name\_last=""

if regexm("`x'", "([0-9]+)") {

local wave = regexs(1)

local name = regexr("`x'", "([0-9]+)", "w")

ren `x' `name'`wave'

}

}

order \_all, sequential

order kidid hhid, first

b1b1\_construct\_variables

save $output\_dir/b1b\_kidresp\_double\_merge.dta, replace

end

prog b1b1\_construct\_variables

forvalues x = 3/12{

egen hwkdcare`x'=sum(kwkdcare`x'), by(hhid)

gen kwkdcarehr`x'= 0

replace kwkdcarehr`x'= max(hwkdcarehr`x'/hwkdcare`x',100) if kwkdcare`x'==1

drop hwkdcarehr`x' hwkdcare`x'

}

end

prog b1a\_load\_data

b1a1\_load\_cor\_geo

b1a2\_load\_kid\_geo

b1a3\_load\_rand\_HRS

end

prog b1a1\_load\_cor\_geo

u "$b1\_cor\_geo\_dta", clear

keep hhid pn year stateusps zipcode urbrur2003 urbrur2013

gen hhidpn = hhid+pn

destring hhidpn, replace

ren urbrur2003 urbrurIII

ren urbrur2013 urbrurXIII

ren year interview\_year

drop if interview\_year<2004

gen wave = (interview\_year-2004)/2+7

drop interview\_year

order hhid pn hhidpn wave stateusps, first

order urbrurXIII, last

foreach x of varlist stateusps-urbrurXIII{

ren `x' rgeo`x'

}

reshape wide rgeostateusps-rgeourbrurXIII, i(hhidpn) j(wave)

save $temp\_dir/b1a1\_cor\_geo.dta, replace

end

prog b1a2\_load\_kid\_geo

u "$b1\_child\_geo\_dta", clear

keep hhid opn flag\* state\* city\* long\* lat\*

drop statename\*

order hhid opn city18, first

order long04, last

foreach x of varlist \_all {

if regexm("`x'", "([0-9]+)") {

local year = regexs(1)

local name = regexr("`x'", "([0-9]+)", "")

local wave = 5+`year'/2

ren `x' kgeo`name'`wave'

}

}

gen kidid = hhid+"0"+opn

collapse (firstnm) hhid-kgeolong7, by(kidid)

drop hhid

save $temp\_dir/b1a2\_child\_geo.dta, replace

end

prog b1a3\_load\_rand\_HRS

b1a3a\_load\_rand\_kidresp

b1a3b\_load\_rand\_resp

b1a3c\_load\_rand\_resp\_long

end

prog b1a3a\_load\_rand\_kidresp

u "$b1\_family\_kidlevel\_dta", clear

keep hhidpn opn kidid ///

hasplit kabyearbg kagenderbg kaeduc ///

k\*resd ///

k\*mstat k\*nkid ///

k\*kdcare k\*tcany k\*tcamt

b1a3a1\_construct\_variables

b1a3a2\_collapse\_to\_kid

save $temp\_dir/b1a3a\_rand\_kidresp.dta, replace

end

prog b1a3a2\_collapse\_to\_kid

gen hhid = floor(hhidpn/1000)

collapse (firstnm) hhid kabyearbg kagenderbg ///

(max) hasplit kaeduc k\*resd k\*mstat k\*nkid k\*kdcare k\*tcany ink\* ///

(sum) k\*tcamt ///

(min) kf\_born\_w kf\_born\_wl kfo\_born\_w, by(kidid)

end

prog b1a3c\_load\_rand\_resp\_long

u "$b1\_longitudinal\_dta", clear

keep hhidpn ragender raracem r\*sayret s\*sayret

save $temp\_dir/b1a3c\_rand\_resp\_long.dta, replace

end

prog b1a3b\_load\_rand\_resp

u "$b1\_family\_reslevel\_dta", clear

keep hhidpn rlink r\*wthh s\*kdcarehr r\*kdcarehr

save $temp\_dir/b1a3b\_rand\_resp.dta, replace

end

prog b1a3a1\_construct\_variables

forvalue x = 4/12{

local l = `x'-1

local ll = `x'-2

local lll = `x'-3

gen ink`x'nkidl= 0 if (mod(k`x'nkid,1)==0 & mod(k`l'nkid,1)==0)

gen ink`x'nkidll= 0 if (mod(k`x'nkid,1)==0 & mod(k`ll'nkid,1)==0)

gen ink`x'nkidlll= 0 if (mod(k`x'nkid,1)==0 & mod(k`lll'nkid,1)==0)

replace ink`x'nkidl= (k`x'nkid != k`l'nkid) if (mod(k`x'nkid,1)==0 & mod(k`l'nkid,1)==0)

replace ink`x'nkidll= (k`x'nkid != k`ll'nkid) if (mod(k`x'nkid,1)==0 & mod(k`ll'nkid,1)==0)

replace ink`x'nkidlll= (k`x'nkid != k`lll'nkid) if (mod(k`x'nkid,1)==0 & mod(k`lll'nkid,1)==0)

replace ink`x'nkidll=0 if ink`x'nkidl==1

gen ink`x'nkidbll=ink`x'nkidll

replace ink`x'nkidbll= 1 if ink`x'nkidl==1

replace ink`x'nkidlll=0 if ink`x'nkidbll== 1

gen ink`x'nkidblll=ink`x'nkidlll

replace ink`x'nkidblll= 1 if ink`x'nkidbll== 1

if `x'<=11{

local f = `x'+1

gen ink`x'nkidf= 0 if (mod(k`x'nkid,1)==0 & mod(k`f'nkid,1)==0)

replace ink`x'nkidf= (k`x'nkid != k`f'nkid) if (mod(k`x'nkid,1)==0 & mod(k`f'nkid,1)==0)

}

if `x'<=10{

local ff= `x'+2

gen ink`x'nkidff= 0 if (mod(k`x'nkid,1)==0 & mod(k`ff'nkid,1)==0)

replace ink`x'nkidff= (k`x'nkid != k`ff'nkid) if (mod(k`x'nkid,1)==0 & mod(k`ff'nkid,1)==0)

replace ink`x'nkidff=0 if ink`x'nkidf==1

gen ink`x'nkidbff=ink`x'nkidff

replace ink`x'nkidbff= 1 if ink`x'nkidf==1

}

}

gen kf\_born\_w=.

gen kf\_born\_wl=.

forvalue x = 2/12{

local l = `x'-1

replace kf\_born\_w=`x' if k`x'nkid>0 & k`x'nkid<10e8 & k`l'nkid==0

if `x'>2{

local ll = `x'-2

replace kf\_born\_wl=`x' if k`x'nkid>0 & k`x'nkid<10e8 & k`ll'nkid==0

}

}

gen kfo\_born\_w= .

forvalue x = 1/12{

replace kfo\_born\_w=`x' if k`x'nkid>0 & k`x'nkid<10e8 & kfo\_born\_w==.

}

end

\*Execute

if "`1'"!="function calling"{

main

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*b2\_create\_distance\_measure

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Outline:

\*/

set more off

global b1\_output\_dta = "../output/b1b\_kidresp\_double\_merge.dta"

global zip\_cord\_csv = "../data\_file/zip\_cord\_2013.csv"

if "`1'"!="function calling"{

prog drop \_all

global temp\_dir ="../temp"

global output\_dir="../output"

prog main

cd "U:\2110ChildcareMigration\1Clean\1PreliminaryDataPattern\code"

b2a\_obtain\_rgeocord

b2b\_distance\_measure

b2c\_reshape\_long

\*b2d\_summary\_stat

end

}

prog b2d\_summary\_stat

u $output\_dir/b2c\_kidresp\_double\_merge.dta, clear

egen freq = count(wave), by(kidid)

tab freq

tab kage

sum hgeolat\*

sum kgeolatw\*

sum khdist\*

sum kdistc\*

sum hdistc\*

sum khdistc\*

sum in\*

end

prog b2c\_reshape\_long

drop fpn mpn opn

local stub = ""

foreach x of varlist \*8{

local len = length("`x'")

local y = substr("`x'",1,`len'-1)

local stub = " `stub' "+" `y' "

}

reshape long `stub', i(kidid) j(wave)

drop if wave<7 | wave>12

gen kage = 2004 + (wave-7)\*2 - kabyearbg

drop if kage==.

drop if kage<25 | kage>44

gen kstage\_detail="25-29" if kage<=29

replace kstage\_detail="30-34" if kage>=30 & kage<=34

replace kstage\_detail="35-39" if kage>=35 & kage<=39

replace kstage\_detail="40-44" if kage>=40 & kage<=44

gen kstage\_num=1 if kage<=29

replace kstage\_num=2 if kage>=30 & kage<=34

replace kstage\_num=3 if kage>=35 & kage<=39

replace kstage\_num=4 if kage>=40 & kage<=44

gen kskill=0 if mod(kaeduc,1)==0

replace kskill=1 if kaeduc>=16 & kaeduc<10e8

save $output\_dir/b2c\_kidresp\_double\_merge.dta, replace

end

prog b2a\_obtain\_rgeocord

import delimited using $zip\_cord\_csv, clear

save $temp\_dir/zip\_cord.dta, replace

u $b1\_output\_dta, clear

forvalues x = 7/14{

gen zip=real(hgeozipcodew`x')

merge n:1 zip using $temp\_dir/zip\_cord.dta, keep(match master) nogen

ren lat hgeolat`x'

ren lng hgeolong`x'

drop zip

}

end

prog b2b\_distance\_measure

forvalues x = 7/14{

geodist kgeolatw`x' kgeolongw`x' hgeolat`x' hgeolong`x', generate(khdist`x') miles

}

forvalues x = 8/14{

local y = `x'-1

geodist kgeolatw`x' kgeolongw`x' kgeolatw`y' kgeolongw`y', generate(kdistc`x') miles

geodist hgeolat`x' hgeolong`x' hgeolat`y' hgeolong`y', generate(hdistc`x') miles

gen khdistc`x'= khdist`x'-khdist`y'

gen inkdistc`x'= 0 if kdistc`x'<20 & kdistc`x'!=.

replace inkdistc`x'= 1 if kdistc`x'>=20 & kdistc`x'!=.

gen inhdistc`x'= 0 if hdistc`x'<20 & hdistc`x'!=.

replace inhdistc`x'= 1 if hdistc`x'>=20 & hdistc`x'!=.

gen khdist\_pri`x'=khdist`y'

}

sum hgeolat\*

sum kgeolatw\*

sum khdist\*

sum kdistc\*

sum hdistc\*

sum khdistc\*

sum in\*

end

\*Execute

if "`1'"!="function calling"{

main

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*b3\_spatial\_analysis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Outline:

\*/

set more off

global b2\_output\_dta = "../output/b2c\_kidresp\_double\_merge.dta"

if "`1'"!="function calling"{

prog drop \_all

global temp\_dir ="../temp"

global output\_dir="../output"

prog main

cd "U:\2110ChildcareMigration\1Clean\1PreliminaryDataPattern\code"

b3a\_movement\_after\_birth

b3b\_transfer\_over\_space

end

}

prog b3b\_transfer\_over\_space

u $b2\_output\_dta, clear

keep if inkwnkidblll==1 & kwkdcarehr!=. & khdist!=.

gen kwkdcarehr\_num = kwkdcarehr

replace kwkdcarehr\_num = 1 if kwkdcarehr>=100 & kwkdcarehr<300

replace kwkdcarehr\_num = 2 if kwkdcarehr>=300 & kwkdcarehr<1000

replace kwkdcarehr\_num = 3 if kwkdcarehr>=1000 & kwkdcarehr<10e8

gen khdist\_num = 0 if khdist< 20

replace khdist\_num = 1 if khdist>=20 & khdist<50

replace khdist\_num = 2 if khdist>=50 & khdist<200

replace khdist\_num = 3 if khdist>=200 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

eststo clear

quietly eststo: reg kwtcamt khdist i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr khdist i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr\_num khdist i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr\_num khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(khdist\* \*kskill \*haracem \*kagenderbg)

eststo clear

quietly eststo: reg kwtcamt i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII i.kwkdcarehr\_num

quietly eststo: reg kwkdcarehr\_num i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr\_num i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII kwtcamt

quietly eststo: reg kwkdcarehr i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep( \*khdist\_num \*kskill \*kwkdcarehr\_num kwtcamt \*haracem)

tab kwkdcarehr\_num khdist\_num, column

end

prog b3a\_movement\_after\_birth

u $b2\_output\_dta, clear

gen inkhdistc= .

replace inkhdistc=1 if khdistc>20 & khdistc<10e8

replace inkhdistc=0 if khdistc<-20

sum inkwnkidl\* inkwnkidf\*

eststo clear

quietly eststo: reg khdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\* \*kskill)

eststo clear

quietly eststo: reg kdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg hdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg hdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\* \*kskill)

eststo clear

quietly eststo: logit inkhdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: logit inhdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: logit inkdistc inkwnkidlll inkwnkidll inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: logit inkhdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: logit inhdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: logit inkdistc inkwnkidff inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\* \*kskill)

/\*

eststo clear

quietly eststo: reg khdistc inkwnkidll i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidff i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\*)

eststo clear

quietly eststo: reg kwtcamt inkwnkidll i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwtcamt inkwnkidff i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\*)

eststo clear

quietly eststo: reg kwkdcarehr inkwnkidll i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr inkwnkidff i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\*)

eststo clear

quietly eststo: reg inhdistc inkwnkidll i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inhdistc inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inhdistc inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inhdistc inkwnkidff i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\*)

eststo clear

quietly eststo: reg inkdistc inkwnkidll i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inkdistc inkwnkidl i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inkdistc inkwnkidf i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg inkdistc inkwnkidff i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(ink\*)

\*/

end

\*Execute

if "`1'"!="function calling"{

main

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*b3B\_spatial\_analysis

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Outline:

\*/

set more off

global b2\_output\_dta = "../output/b2c\_kidresp\_double\_merge.dta"

if "`1'"!="function calling"{

prog drop \_all

global temp\_dir ="../temp"

global output\_dir="../output"

prog main

cd "U:\2110ChildcareMigration\1Clean\1PreliminaryDataPattern\code"

\*b3Ba\_cross\_sectional\_movement1

\*b3Bb\_cross\_sectional\_movement2

\*b3Bc\_longitudinal\_movement

\*b3Bd\_treatment\_analysis

\*b3Be\_subgroup\_analysis

\*b3Bf\_subgroup\_analysis2

b3Bg\_condition\_on\_moving

end

}

prog b3Bg\_condition\_on\_moving

u $b2\_output\_dta, clear

gen khdist\_num = 0 if khdist< 50

replace khdist\_num = 1 if khdist>=50 & khdist<200

replace khdist\_num = 2 if khdist>=200 & khdist<500

replace khdist\_num = 3 if khdist>=500 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

gen haskid=kwnkid

replace haskid=1 if kwnkid>0 & kwnkid<10e8

egen everhaskid=max(haskid), by(kidid)

egen khdistmax=max(khdist), by(kidid)

gen khdistmax\_num = 0 if khdistmax< 50

replace khdistmax\_num = 1 if khdistmax>=50 & khdistmax<200

replace khdistmax\_num = 2 if khdistmax>=200 & khdistmax<500

replace khdistmax\_num = 3 if khdistmax>=500 & khdistmax<1000

replace khdistmax\_num = 4 if khdistmax>=1000 & khdistmax<10e8

gen khdistend\_temp=khdist if wave==12

egen khdistend=max(khdistend\_temp), by(kidid)

drop khdistend\_temp

gen khdistend\_num = 0 if khdistend< 50

replace khdistend\_num = 1 if khdistend>=50 & khdistend<200

replace khdistend\_num = 2 if khdistend>=200 & khdistend<500

replace khdistend\_num = 3 if khdistend>=500 & khdistend<1000

replace khdistend\_num = 4 if khdistend>=1000 & khdistend<10e8

gen ktreat = 0

replace ktreat = 1 if kfo\_born\_w-1 <= wave

replace ktreat = 2 if kfo\_born\_w+3 > wave & ktreat == 1

egen kididnum=group(kidid)

gen old= 1 if kstage\_num==4 & wave==12

egen isold=max(old), by(kidid)

\*keep if isold==1

gen khdist\_pri\_num = 0 if khdist\_pri<200

replace khdist\_pri\_num = 1 if khdist\_pri>=200 & khdist\_pri<500

replace khdist\_pri\_num = 2 if khdist\_pri>=500 & khdist\_pri<1000

replace khdist\_pri\_num = 3 if khdist\_pri>=1000 & khdist\_pri<1500

replace khdist\_pri\_num = 4 if khdist\_pri>=1500 & khdist\_pri<2000

keep if kdistc>=50

\*directly

eststo clear

sort khdist\_pri\_num

quietly by khdist\_pri\_num: eststo: reg khdistc i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

esttab, p keep(\*treat \*kstage\_num \_cons) stats(r2 N)

by khdist\_pri\_num: sum khdist

end

prog b3Bf\_subgroup\_analysis2

u $b2\_output\_dta, clear

gen khdist\_num = 0 if khdist< 50

replace khdist\_num = 1 if khdist>=50 & khdist<200

replace khdist\_num = 2 if khdist>=200 & khdist<500

replace khdist\_num = 3 if khdist>=500 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

gen haskid=kwnkid

replace haskid=1 if kwnkid>0 & kwnkid<10e8

egen everhaskid=max(haskid), by(kidid)

egen khdistmax=max(khdist), by(kidid)

gen khdistmax\_num = 0 if khdistmax< 50

replace khdistmax\_num = 1 if khdistmax>=50 & khdistmax<200

replace khdistmax\_num = 2 if khdistmax>=200 & khdistmax<500

replace khdistmax\_num = 3 if khdistmax>=500 & khdistmax<1000

replace khdistmax\_num = 4 if khdistmax>=1000 & khdistmax<10e8

gen khdistend\_temp=khdist if wave==12

egen khdistend=max(khdistend\_temp), by(kidid)

drop khdistend\_temp

gen khdistend\_num = 0 if khdistend< 50

replace khdistend\_num = 1 if khdistend>=50 & khdistend<200

replace khdistend\_num = 2 if khdistend>=200 & khdistend<500

replace khdistend\_num = 3 if khdistend>=500 & khdistend<1000

replace khdistend\_num = 4 if khdistend>=1000 & khdistend<10e8

gen ktreat = 0

replace ktreat = 1 if kfo\_born\_w-1 <= wave

replace ktreat = 2 if kfo\_born\_w+3 > wave & ktreat == 1

egen kididnum=group(kidid)

gen old= 1 if kstage\_num==3 & wave==12

egen isold=max(old), by(kidid)

keep if isold==1

\*directly

eststo clear

quietly eststo: reg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

xtset kididnum wave

quietly eststo: xtreg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII, re

quietly eststo: xtreg khdist i.ktreat i.kstage\_num i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

sss

\*by kskill

forvalue x=0/1{

preserve

keep if kskill==`x'

eststo clear

quietly eststo: reg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

xtset kididnum wave

quietly eststo: xtreg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII, re

quietly eststo: xtreg khdist i.ktreat i.kstage\_num i.wave hgeourbrurXIII, fe

esttab, p keep(\*treat \*kstage\_num) stats(r2 N)

restore

}

\*by khdistend\_num

local group khdistend\_num

sort `group'

eststo clear

quietly by `group': eststo: reg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

xtset kididnum wave

sort `group'

eststo clear

quietly by `group': eststo: xtreg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII, re

esttab, p drop(\*wave) stats(r2 N)

xtset kididnum wave

sort `group'

eststo clear

quietly by `group': eststo: xtreg khdist i.ktreat i.kstage\_num i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

\*by khdistend\_num kskill

eststo clear

local group khdistend\_num

forvalue x = 0/1{

preserve

keep if kskill==`x'

sort `group'

quietly by `group': eststo: reg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

restore

}

esttab, p drop(\*wave) stats(r2 N)

forvalue x = 0/1{

preserve

keep if kskill==`x'

xtset kididnum wave

sort `group'

quietly by `group': eststo: xtreg khdist i.ktreat i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII, re

restore

}

esttab, p drop(\*wave) stats(r2 N)

eststo clear

forvalue x = 0/1{

preserve

keep if kskill==`x'

xtset kididnum wave

sort `group'

quietly by `group': eststo: xtreg khdist i.ktreat i.kstage\_num i.wave hgeourbrurXIII, fe

restore

}

esttab, p drop(\*wave) stats(r2 N)

\*by kstage\_num

local group kstage\_num

sort `group'

eststo clear

quietly by `group': eststo: reg khdist i.ktreat i.kskill i.haracem i.kagenderbg i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

end

prog b3Be\_subgroup\_analysis

u $b2\_output\_dta, clear

egen wave\_count=count(wave) , by(kidid)

tab wave\_count

keep if wave\_count==6

drop wave\_count

gen khdist\_num = 0 if khdist< 50

replace khdist\_num = 1 if khdist>=50 & khdist<200

replace khdist\_num = 2 if khdist>=200 & khdist<500

replace khdist\_num = 3 if khdist>=500 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

gen haskid=kwnkid

replace haskid=1 if kwnkid>0 & kwnkid<10e8

egen everhaskid=max(haskid), by(kidid)

egen khdistmax=max(khdist), by(kidid)

gen khdistmax\_num = 0 if khdistmax< 50

replace khdistmax\_num = 1 if khdistmax>=50 & khdistmax<200

replace khdistmax\_num = 2 if khdistmax>=200 & khdistmax<500

replace khdistmax\_num = 3 if khdistmax>=500 & khdistmax<1000

replace khdistmax\_num = 4 if khdistmax>=1000 & khdistmax<10e8

gen khdistend\_temp=khdist if wave==12

egen khdistend=max(khdistend\_temp), by(kidid)

drop khdistend\_temp

gen khdistend\_num = 0 if khdistend< 50

replace khdistend\_num = 1 if khdistend>=50 & khdistend<200

replace khdistend\_num = 2 if khdistend>=200 & khdistend<500

replace khdistend\_num = 3 if khdistend>=500 & khdistend<1000

replace khdistend\_num = 4 if khdistend>=1000 & khdistend<10e8

gen ktreat = 0

replace ktreat = 1 if kf\_born\_w-1 <= wave

replace ktreat = 2 if kf\_born\_w+3 > wave & ktreat == 1

\*by khdistmax\_num

sort khdistmax\_num

eststo clear

quietly by khdistmax\_num: eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

eststo clear

quietly eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII khdistmax

esttab, p drop(\*wave) stats(r2 N)

egen kididnum=group(kidid)

xtset kididnum wave

sort khdistmax\_num

eststo clear

quietly by khdistmax\_num: eststo: xtreg khdist i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

\*by khdistend\_num

local group khdistend\_num

sort `group'

eststo clear

quietly by `group': eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

xtset kididnum wave

sort `group'

eststo clear

quietly by `group': eststo: xtreg khdist i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

\*by kskill

local group kskill

sort `group'

eststo clear

quietly by `group': eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

xtset kididnum wave

sort `group'

eststo clear

quietly by `group': eststo: xtreg khdist i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

\*by khdistend\_num kskill

eststo clear

local group khdistend\_num

forvalue x = 0/1{

preserve

keep if kskill==`x'

sort `group'

quietly by `group': eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

restore

}

esttab, p drop(\*wave) stats(r2 N)

eststo clear

forvalue x = 0/1{

preserve

keep if kskill==`x'

xtset kididnum wave

sort `group'

quietly by `group': eststo: xtreg khdist i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

restore

}

esttab, p drop(\*wave) stats(r2 N)

\*by khdistmax\_num kskill

eststo clear

local group khdistmax\_num

forvalue x = 0/1{

preserve

keep if kskill==`x'

sort `group'

quietly by `group': eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

restore

}

esttab, p drop(\*wave) stats(r2 N)

eststo clear

forvalue x = 0/1{

preserve

keep if kskill==`x'

xtset kididnum wave

sort `group'

quietly by `group': eststo: xtreg khdist i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

restore

}

esttab, p drop(\*wave) stats(r2 N)

\*by khdistend\_num kskill; look at khdistc

eststo clear

local group khdistend\_num

forvalue x = 0/1{

preserve

keep if kskill==`x'

sort `group'

quietly by `group': eststo: reg khdistc i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

restore

}

esttab, p drop(\*wave) stats(r2 N)

eststo clear

forvalue x = 0/1{

preserve

keep if kskill==`x'

xtset kididnum wave

sort `group'

quietly by `group': eststo: xtreg khdistc i.ktreat mwsayret haskid i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

restore

}

esttab, p drop(\*wave) stats(r2 N)

end

prog b3Bd\_treatment\_analysis

u $b2\_output\_dta, clear

egen wave\_count=count(wave) , by(kidid)

tab wave\_count

keep if wave\_count==6

drop wave\_count

gen khdist\_num = 0 if khdist< 20

replace khdist\_num = 1 if khdist>=20 & khdist<50

replace khdist\_num = 2 if khdist>=50 & khdist<200

replace khdist\_num = 3 if khdist>=200 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

gen haskid=kwnkid

replace haskid=1 if kwnkid>0 & kwnkid<10e8

egen everhaskid=max(haskid), by(kidid)

gen ktreat = 0

replace ktreat = 1 if kf\_born\_w-1 <= wave

replace ktreat = 2 if kf\_born\_w+3 > wave & ktreat == 1

tab ktreat kstage\_num, col

eststo clear

quietly eststo: reg khdist inkwnkidblll mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist\_num i.ktreat mwsayret haskid everhaskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(\*ktreat \*kskill \*haracem hgeour\* kwmstat mwsayret \*kagenderbg \*kstage\_num haskid everhaskid inkwnkidblll) stats(r2 N)

egen kididnum=group(kidid)

xtset kididnum wave

eststo clear

quietly eststo: xtreg khdist inkwnkidblll haskid mwsayret i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

quietly eststo: xtreg khdist i.ktreat haskid mwsayret i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

quietly eststo: xtreg khdist\_num i.ktreat haskid mwsayret i.kstage\_num kwmstat i.wave hgeourbrurXIII, fe

esttab, p drop(\*wave) stats(r2 N)

quietly xtreg khdist i.ktreat everhaskid haskid mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII, fe

estimates store fixed

quietly xtreg khdist i.ktreat everhaskid haskid mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII, re

estimates store random

hausman fixed random

end

prog b3Bc\_longitudinal\_movement

u $b2\_output\_dta, clear

egen wave\_count=count(wave) , by(kidid)

tab wave\_count

keep if wave\_count==6

drop wave\_count

gen khdist\_num = 0 if khdist< 20

replace khdist\_num = 1 if khdist>=20 & khdist<50

replace khdist\_num = 2 if khdist>=50 & khdist<200

replace khdist\_num = 3 if khdist>=200 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen kwnkidII=kwnkid if wave==12

sort wave

collapse (firstnm) haracem kstage\_num kagenderbg kwmstat kskill hgeourbrurXIII first\_khdist\_num=khdist\_num kwnkidII (min) khdist\_num khdistc khdist (max) kwnkid inkwnkidblll inkwnkidbff, by(kidid)

sum inkwnkidblll inkwnkidbff

tab kwnkid, missing

gen haskid=kwnkid

replace haskid=1 if kwnkid>0 & kwnkid<10e8

gen haskidII=kwnkidII

replace haskidII=1 if kwnkidII>0 & kwnkidII<10e8

tab inkwnkidblll haskid

tab inkwnkidblll haskidII

tab haskid haskidII

tab inkwnkidbff haskid

eststo clear

quietly eststo: reg khdistc inkwnkidblll inkwnkidbff haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidblll haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidbff haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidblll inkwnkidbff haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidblll haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdistc inkwnkidbff haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

esttab, p keep(\*first\_khdist\_num \*kskill \*haracem inkwnkidblll inkwnkidbff \*kstage\_num haskid)

eststo clear

quietly eststo: reg khdist inkwnkidblll inkwnkidbff haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdist inkwnkidblll haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdist inkwnkidbff haskid i.first\_khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdist inkwnkidblll inkwnkidbff haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdist inkwnkidblll haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

quietly eststo: reg khdist inkwnkidbff haskid i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat hgeourbrurXIII

esttab, p keep(\*first\_khdist\_num \*kskill \*haracem inkwnkidblll inkwnkidbff \*kstage\_num haskid)

end

prog b3Bb\_cross\_sectional\_movement2

u $b2\_output\_dta, clear

keep if inkwnkidblll==1 & kwkdcarehr!=. & khdist!=.

gen kwkdcarehr\_num = kwkdcarehr

replace kwkdcarehr\_num = 1 if kwkdcarehr>=100 & kwkdcarehr<300

replace kwkdcarehr\_num = 2 if kwkdcarehr>=300 & kwkdcarehr<1000

replace kwkdcarehr\_num = 3 if kwkdcarehr>=1000 & kwkdcarehr<10e8

gen khdist\_num = 0 if khdist< 20

replace khdist\_num = 1 if khdist>=20 & khdist<50

replace khdist\_num = 2 if khdist>=50 & khdist<200

replace khdist\_num = 3 if khdist>=200 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

eststo clear

quietly eststo: reg kwkdcare khdist\_num mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr\_num khdist\_num mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg kwkdcarehr khdist\_num mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(khdist\* \*kskill \*haracem hgeour\* kwmstat mwsayret)

end

prog b3Ba\_cross\_sectional\_movement1

u $b2\_output\_dta, clear

gen mwsayret= 0 if inlist(mrwsayret,0,1,2,3) | inlist(mswsayret,0,1,2,3)

replace mwsayret=1 if mrwsayret==0 | mswsayret==0

sum inkwnkidl\* inkwnkidf\*

eststo clear

quietly eststo: reg khdist inkwnkidlll inkwnkidll inkwnkidl inkwnkidf inkwnkidff mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist inkwnkidlll mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist inkwnkidll mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist inkwnkidl mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist inkwnkidf mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

quietly eststo: reg khdist inkwnkidff mwsayret i.kskill i.haracem i.kstage\_num i.kagenderbg kwmstat i.wave hgeourbrurXIII

esttab, p keep(inkwnkid\* \*kskill \*haracem hgeour\* kwmstat mwsayret)

end

\*Execute

if "`1'"!="function calling"{

main

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*b4\_check\_fertility\_delay

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Outline:

\*/

set more off

global b2\_output\_dta = "../output/b2c\_kidresp\_double\_merge.dta"

if "`1'"!="function calling"{

prog drop \_all

global temp\_dir ="../temp"

global output\_dir="../output"

prog main

cd "U:\2110ChildcareMigration\1Clean\1PreliminaryDataPattern\code"

b4a\_analysis

end

}

prog b4a\_analysis

u $b2\_output\_dta, clear

egen wave\_count=count(wave) , by(kidid)

tab wave\_count

keep if wave\_count==6

drop wave\_count

keep if wave==12

gen khdist\_num = 0 if khdist< 50

replace khdist\_num = 1 if khdist>=50 & khdist<200

replace khdist\_num = 2 if khdist>=200 & khdist<500

replace khdist\_num = 3 if khdist>=500 & khdist<1000

replace khdist\_num = 4 if khdist>=1000 & khdist<10e8

egen kididnum=group(kidid)

gen kage\_fob=kage+2\*(kfo\_born\_w-12)

gen kage\_fb=kage+2\*(kf\_born\_w-12)

gen old= 1 if kstage\_num==3 & wave==12

replace old= 2 if kstage\_num==4 & wave==12

egen isold=max(old), by(kidid)

\*keep if isold==1

\*directly

eststo clear

quietly eststo: reg kage\_fob i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

preserve

keep if isold ==1

quietly eststo: reg kage\_fob i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

restore

preserve

keep if isold==2

quietly eststo: reg kage\_fob i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

restore

esttab, p drop(\*wave) stats(r2 N)

\*by skill

local group kskill

sort `group'

eststo clear

quietly by `group': eststo: reg kage\_fob i.khdist\_num i.kskill i.haracem i.kstage\_num i.kagenderbg i.wave hgeourbrurXIII

esttab, p drop(\*wave) stats(r2 N)

end

\*Execute

if "`1'"!="function calling"{

main

}