```
*************************************************************************************
************************************
// Intergenerational transmission of unemployment - evidence for German sons
// Miriam Mäder, Steffen Müller, Regina T. Riphahn, Caroline Schwientek
*************************************************************************************
************************************

version 12.1
set more off
set matsize 5000

glo wdir                        "DATA PATH (SOEP-FILES)"
glo wdirD                       "DATA PATH (GENERATED DATA/SAMPLES)"
global results                  "RESULTS PATH"


*************************************************************************************
************************************
// Generate variables from the GSOEP
*************************************************************************************
************************************

************************************************************
***********************PGEN*********************************
************************************************************

cd "$wdir"
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc a{
use `welle'pgen.dta, clear
keep emplst lfs nace *psbil *pbbil02 persnr hhnrakt
gen welle_`welle'=1
save Temp\temppgen_`welle', replace
}
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc{
append using Temp\temppgen_`welle'.dta
}
gen jahr=1984 if welle_a==1
replace jahr=1985 if welle_b==1
replace jahr=1986 if welle_c==1
replace jahr=1987 if welle_d==1
replace jahr=1988 if welle_e==1
replace jahr=1989 if welle_f==1
replace jahr=1990 if welle_g==1
replace jahr=1991 if welle_h==1
replace jahr=1992 if welle_i==1
replace jahr=1993 if welle_j==1
replace jahr=1994 if welle_k==1
replace jahr=1995 if welle_l==1
replace jahr=1996 if welle_m==1
replace jahr=1997 if welle_n==1
replace jahr=1998 if welle_o==1
replace jahr=1999 if welle_p==1
replace jahr=2000 if welle_q==1
replace jahr=2001 if welle_r==1
replace jahr=2002 if welle_s==1
replace jahr=2003 if welle_t==1
replace jahr=2004 if welle_u==1
replace jahr=2005 if welle_v==1
replace jahr=2006 if welle_w==1
replace jahr=2007 if welle_x==1
replace jahr=2008 if welle_y==1
replace jahr=2009 if welle_z==1
replace jahr=2010 if welle_ba==1
replace jahr=2011 if welle_bb==1
```

```
replace jahr=2012 if welle_bc==1
        label variable jahr "Jahr der Befragung"

drop welle_*

gen emplst =.
gen lfs =.
gen nace=.
gen school=.
gen college=.

        label var emplst "Employment Status"
        label var lfs "Labor Force Status"
        label var nace "Nomenclature des statistiques des activités économiques
(Communauté européenne)"
        label var school "Höchster Schulabschluss"
        label var college "Höhere Bildung"

forvalues i=84/99{
rename emplst`i' emplst19`i'
rename lfs`i' lfs19`i'
rename nace`i' nace19`i'
}
foreach num in 00 01 02 03 04 05 06 07 08 09 10 11 12{
rename emplst`num' emplst20`num'
rename lfs`num' lfs20`num'
rename nace`num' nace20`num'
}
foreach num of numlist 1984/2012 {
replace emplst = emplst`num' if jahr == `num'
replace lfs = lfs`num' if jahr == `num'
replace nace = nace`num' if jahr == `num'
}

        label values emplst emplst11_EN
        label values lfs lfs11_EN
        label values nace nace11_EN

forvalues i =1984/2012{
drop lfs`i'
drop emplst`i'
drop nace`i'
}

        renvars *psbil \ school1984-school2012 , display
        renvars *pbbil02 \ college1984-college2012 , display

foreach num of numlist 1984/2012 {
replace school = school`num' if jahr == `num'
replace college = college`num' if jahr == `num'
}
        label values school apsbil_EN
        label values college bcpbbil02_EN

forvalues i =1984/2012{
drop school`i'
drop college`i'
}

cd "$wdirD"
save PGEN-Variables.dta, replace

*********************************************************
**********************HBRUTTO****************************
*********************************************************

cd "$wdir"
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc a {
```

```
use `welle'hbrutto.dta, clear
keep hhnrakt `welle'bula
gen welle_`welle'=1
save Temp\temphbrutto_`welle', replace
}
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc{
append using Temp\temphbrutto_`welle'
}
gen jahr=1984 if welle_a==1
replace jahr=1985 if welle_b==1
replace jahr=1986 if welle_c==1
replace jahr=1987 if welle_d==1
replace jahr=1988 if welle_e==1
replace jahr=1989 if welle_f==1
replace jahr=1990 if welle_g==1
replace jahr=1991 if welle_h==1
replace jahr=1992 if welle_i==1
replace jahr=1993 if welle_j==1
replace jahr=1994 if welle_k==1
replace jahr=1995 if welle_l==1
replace jahr=1996 if welle_m==1
replace jahr=1997 if welle_n==1
replace jahr=1998 if welle_o==1
replace jahr=1999 if welle_p==1
replace jahr=2000 if welle_q==1
replace jahr=2001 if welle_r==1
replace jahr=2002 if welle_s==1
replace jahr=2003 if welle_t==1
replace jahr=2004 if welle_u==1
replace jahr=2005 if welle_v==1
replace jahr=2006 if welle_w==1
replace jahr=2007 if welle_x==1
replace jahr=2008 if welle_y==1
replace jahr=2009 if welle_z==1
replace jahr=2010 if welle_ba==1
replace jahr=2011 if welle_bb==1
replace jahr=2012 if welle_bc==1
        label variable jahr "Jahr der Befragung"

drop welle_*
renvars *bula \ bula1984-bula2012

gen bula=.

        label variable bula "Bundesland"
        label define bulalb 1 "Schleswig Holstein [1]" 2 "Hamburg [2]" 3
"Niedersachsen [3]" 4 "Bremen [4]" 5 "Nordrhein Westfalen [5]" 6 "Hessen [6]" 7
"Rheinland Pfalz [7]" 8 "Baden Württemberg [8]" 9 "Bayern [9]" 10 "Saarland
[10]" 11 "Berlin [11]" 12 "Brandenburg [12]" 13 "Mecklenburg-Vorpommern [13]" 14
"Sachsen [14]" 15 "Sachsen Anhalt [15]" 16 "Thüringen [16]"
        label values bula bulalb

foreach num of numlist 1984/2012 {
replace bula            = bula`num' if jahr==`num'
}
label values bula bbbula_EN
drop bula1984-bula2012

cd "$wdirD"
save HBRUTTO-Variables.dta, replace


************************************************************
**********************PEQUIV********************************
************************************************************

cd "$wdir"
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc a{
```

```
use `welle'pequiv.dta, clear
keep  e11106 persnr hhnrakt
gen welle_`welle'=1
save Temp\temppgen_`welle', replace
}
foreach welle in b c d e f g h i j k l m n o p q r s t u v w x y z ba bb bc{
append using Temp\temppgen_`welle'.dta
}
gen jahr=1984 if welle_a==1
replace jahr=1985 if welle_b==1
replace jahr=1986 if welle_c==1
replace jahr=1987 if welle_d==1
replace jahr=1988 if welle_e==1
replace jahr=1989 if welle_f==1
replace jahr=1990 if welle_g==1
replace jahr=1991 if welle_h==1
replace jahr=1992 if welle_i==1
replace jahr=1993 if welle_j==1
replace jahr=1994 if welle_k==1
replace jahr=1995 if welle_l==1
replace jahr=1996 if welle_m==1
replace jahr=1997 if welle_n==1
replace jahr=1998 if welle_o==1
replace jahr=1999 if welle_p==1
replace jahr=2000 if welle_q==1
replace jahr=2001 if welle_r==1
replace jahr=2002 if welle_s==1
replace jahr=2003 if welle_t==1
replace jahr=2004 if welle_u==1
replace jahr=2005 if welle_v==1
replace jahr=2006 if welle_w==1
replace jahr=2007 if welle_x==1
replace jahr=2008 if welle_y==1
replace jahr=2009 if welle_z==1
replace jahr=2010 if welle_ba==1
replace jahr=2011 if welle_bb==1
replace jahr=2012 if welle_bc==1
        label variable jahr "Jahr der Befragung"

drop welle_*
gen industry =.
        label var industry "1 digit industry code"

forvalues i=84/99{
rename e11106`i' e1110619`i'
}
foreach num in 00 01 02 03 04 05 06 07 08 09 10 11 12{
rename e11106`num' e1110620`num'
}

foreach num of numlist 1984/2012 {
replace industry = e11106`num' if jahr == `num'
}
label values industry e1110611_EN
drop e111061984- e111062012

cd "$wdirD"
save PEQUIV-Variables.dta, replace

************************************************************
**********************BIOPAREN-Vater**********************
************************************************************

cd "$wdir"
use bioparen.dta, clear
keep persnr vsbil vbbil vbstell

cd "$wdirD"
```

```
save BIOPAREN-Variables_Vater.dta, replace

*************************************************************
**********************BIOPAREN-Mutter*********************
*************************************************************

cd "$wdir"
use bioparen.dta, clear
keep persnr msbil mbbil mbstell

cd "$wdirD"
save BIOPAREN-Variables_Mutter.dta, replace


*******************************************************************************
***********************************
// Definition of variables
*******************************************************************************
***********************************

* Arbeitslosigkeit.dta

cd "$wdirD"
use PGEN-Variables.dta, clear
keep persnr jahr hhnrakt emplst lfs nace school college

gen AL_1 = .
        label var AL_1 "=1 if unemployed (definition #1)"
        replace AL_1 = 0 if emplst == 1 | emplst == 2 | emplst == 3 | emplst ==
4 | emplst == 6
        replace AL_1 = 0 if emplst == 5 & lfs != 6
        replace AL_1 = 1 if empls == 5 & lfs == 6

gen AL_2 = .
        label var AL_2 "=1 if unemployed (definition #2)"
        replace AL_2 = 0 if emplst == 1 | emplst == 2 | emplst == 3 | emplst ==
4 | emplst == 6
        replace AL_2 = 0 if emplst == 5 & lfs == 3 ///
                                        | emplst == 5 & lfs == 4 ///
                                        | emplst == 5 & lfs == 5 ///
                                        | emplst == 5 & lfs == 7 ///
                                        | emplst == 5 & lfs == 8 ///
                                        | emplst == 5 & lfs == 9 ///
                                        | emplst == 5 & lfs == 10 ///
                                        | emplst == 5 & lfs == 11 ///
                                        | emplst == 5 & lfs == 12
        replace AL_2 = 1 if emplst == 5 & lfs == 1 ///
                                        | emplst == 5 & lfs == 2 ///
                                        | emplst == 5 & lfs == 6

recode school college (-3 -2 -1 =.)
save Arbeitslosigkeit.dta, replace

* Bula_HH.dta

cd "$wdirD"
use HBRUTTO-Variables.dta, clear

recode bula (-1 -2 -3=.)
sort hhnrakt jahr
save Bula_HH.dta, replace

* Vaterbildung.dta

cd "$wdirD"
use BIOPAREN-Variables_Vater, clear

mvdecode vsbil vbbil vbstell, mv(-3 -2 -1 0=.)
```

```
rename vsbil Vater_Schule
rename vbbil Vater_hBildung
rename vbstell Vater_Beruf

cd "$wdirD"
sort persnr
save Vaterbildung, replace

* Mutterbildung.dta

cd "$wdirD"
use BIOPAREN-Variables_Mutter, clear

mvdecode msbil mbbil mbstell, mv(-3 -2 -1 0=.)

rename msbil Mutter_Schule
rename mbbil Mutter_hBildung
rename mbstell Mutter_Beruf

cd "$wdirD"
sort persnr
save Mutterbildung, replace

* Vatercharakteristika.dta

cd "$wdir"
use ppfad.dta, clear

keep persnr gebjahr migback

rename gebjahr gebjahr_Vater
rename migback migback_Vater
        label var gebjahr_Vater "Geburtsjahr Vater"
        label var migback_Vater "Migrationshintergrund Vater"

cd "$wdirD"
sort persnr
save Vatercharakteristika, replace

*  Muttercharakteristika.dta

cd "$wdir"
use ppfad.dta, clear

keep persnr gebjahr migback

rename gebjahr gebjahr_Mutter
rename migback migback_Mutter
        label var gebjahr_Mutter "Geburtsjahr Mutter"
        label var migback_Mutter "Migrationshintergrund Mutter"

cd "$wdirD"
sort persnr
save Muttercharakteristika, replace

* Kindcharakteristika_Sohn.dta

cd "$wdir"
use ppfad.dta, clear
keep persnr sex gebjahr migback

        label var sex "Geschlecht Sohn"
        label var gebjahr "Geburtsjahr Sohn"
        label var migback "Migrationshintergrund Sohn"

cd "$wdirD"
sort persnr
```

```
save Kindcharakteristika_Sohn, replace

*  Geschwister.dta

cd "$wdir"
use ppfad.dta, clear
        keep persnr hhnr
        sort persnr

merge 1:1 persnr using bioparen
drop _merge

keep persnr hhnr mnr geschw nums numb
rename persnr ID
rename mnr persnr

sort persnr
merge m:1 persnr using biobirth

keep ID persnr hhnr sumkids geschw nums numb
rename persnr mnr
rename ID persnr

keep if persnr !=.
sort persnr
merge 1:1 persnr using hp
drop _merge

sort persnr
merge 1:1 persnr using mp
drop _merge

sort persnr
merge 1:1 persnr using rp
drop _merge

sort persnr
merge 1:1 persnr using tp
drop _merge

sort persnr
merge 1:1 persnr using wp
drop _merge

keep persnr hhnr mnr sumkids hp1319 hp1323 mp09h02 mp09g02 rp107h02 rp107g02
tp13703 tp13702 wp11721 wp11724 geschw nums numb
recode hp1319 hp1323 mp09h02 mp09g02 rp107h02 rp107g02 tp13703 tp13702 wp11721
wp11724 (-3 -1=.) (-2=0), copyrest

egen Schwestern1=rowmax(wp11721 tp13703 rp107h02 mp09h02 hp1323)
egen Brüder1=rowmax(wp11724 tp13702 rp107g02 mp09g02 hp1319)
gen Geschwister1=Schwestern1+Brüder1

gen Kinder = sumkids if sumkids != 0
gen Geschwister2 = Kinder -1

recode geschw (-2 -1 =.)
recode nums numb(-1=.)
gen Schwestern2=nums if nums != -2
replace Schwestern2 = 0 if geschw == 1 & nums == -2
replace Schwestern2 = 0 if geschw == 2
gen Brüder2=numb if numb != -2
replace Brüder2 = 0 if geschw == 1 & numb == -2
replace Brüder2 = 0 if geschw == 2
gen Geschwister3 = Schwestern2 + Brüder2

egen Geschwister=rowmax(Geschwister1 Geschwister2 Geschwister3)
        label var Geschwister "Anzahl Geschwister"
```

```
merge 1:1 persnr using biosib
keep persnr Geschwister num_sib
replace num_sib = num_sib - 1 if num_sib !=.

replace Geschwister = num_sib if Geschwister ==.
replace Geschwister = num_sib if num_sib > Geschwister & num_sib !=.
drop num_sib

cd "$wdirD"
sort persnr
save Geschwister, replace

* Reihenfolge_Geschwister.dta

cd "$wdir"
use biosib, clear
keep persnr pos_sib
recode pos_sib (-2=.)
rename pos_sib birthorder
        label var birthorder "Birth Order"

cd "$wdirD"
sort persnr
save Reihenfolge_Geschwister.dta, replace

* Branchen-Dummies_1DIC.dta

cd "$wdirD"
use PEQUIV-Variables.dta, clear
keep persnr jahr industry

recode industry (-2 -1 0 =.)
sort persnr jahr
save Branchen_Dummies_1DIC.dta, replace

duplicates drop persnr, force
keep persnr
expand 39
bysort persnr: gen jahr=_n
replace jahr = 1973 + jahr

merge 1:m persnr jahr using Branchen_Dummies_1DIC.dta
drop _merge
erase Branchen_Dummies_1DIC.dta

sort persnr jahr
foreach jahr in 2011 2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999
1998 1997 1996 1995 1994 1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 1983
1982 1981 1980 1979 1978 1977 1976 1975 1974{
replace industry = industry[_n+1] if jahr == `jahr' & industry == .
}

by persnr (jahr): egen perioden_branche = count(industry)
drop if perioden_branche == 0
drop perioden_branche
rename industry ind1dic

sort persnr jahr
save Initial_Industry_1DIC.dta, replace


* Branchen-Dummies_2DIC.dta

cd "$wdirD"
use PGEN-Variables.dta, clear
keep persnr jahr nace
```

```
recode nace (-3 -2 -1 0 =.)
sort persnr jahr
save Branchen_Dummies_2DIC.dta, replace

duplicates drop persnr, force
keep persnr
expand 39
bysort persnr: gen jahr=_n
replace jahr = 1973 + jahr

merge 1:m persnr jahr using Branchen_Dummies_2DIC.dta
drop _merge
erase Branchen_Dummies_2DIC.dta

sort persnr jahr
foreach jahr in 2011 2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999
1998 1997 1996 1995 1994 1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 1983
1982 1981 1980 1979 1978 1977 1976 1975 1974{
replace nace = nace[_n+1] if jahr == `jahr' & nace == .
}

by persnr (jahr): egen perioden_branche = count(nace)
drop if perioden_branche == 0
drop perioden_branche
rename nace ind2dic

sort persnr jahr
save Initial_Industry_2DIC.dta, replace

* ALrisiko_pgen.dta

cd "$wdirD"
use PGEN-Variables, clear
keep persnr jahr emplst lfs nace

qui tab1 emplst lfs nace, m
recode emplst lfs nace (-3 -2 -1 = .)

gen not_employed = .
replace not_employed = 1 if emplst == 5
replace not_employed = 0 if emplst == 1 | emplst == 2 |emplst == 3 | emplst == 4
| emplst == 6
drop if not_employed == .
        label var not_employed "Employment Status = not employed"

sort persnr jahr
by persnr (jahr): egen perioden_soep = count(jahr)
by persnr (jahr): egen perioden_branche = count(nace)
drop if perioden_branche == 0

rename nace industry
sort persnr jahr
keep persnr jahr industry not_employed
cd "$wdirD"
save risiko_2DIC, replace

use risiko_2DIC, clear

keep persnr
duplicates drop persnr, force
expand 29
sort persnr
by persnr: gen jahr=_n
replace jahr = 1983+jahr
merge 1:1 persnr jahr using risiko_2DIC.dta
drop _merge

gen ind_full = industry
```

```
bysort persnr (jahr): replace ind_full = ind_full[_n-1] if ind_full[_n] == .

sort persnr jahr
foreach jahr in 2011 2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999
1998 1997 1996 1995 1994 1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 {
bysort persnr: replace ind_full  = ind_full[_n+1] if jahr == `jahr' & ind_full
== .
}

gen jahr_ind=jahr if industry!=.
bysort persnr: egen jahr_indf_pgen = min(jahr_ind)

gen ind1=industry if not_employed==0
bysort persnr: replace ind1 = industry[_n-1] if industry[_n]==. & not_employed
==1

gen ind5=industry if not_employed==0
bysort persnr: replace ind5 = industry[_n-1] if ind5[_n]==. & industry[_n-1] !=.
 & not_employed ==1
bysort persnr: replace ind5 = industry[_n-2] if ind5[_n]==. & industry[_n-2] !=.
 & not_employed ==1
bysort persnr: replace ind5 = industry[_n-3] if ind5[_n]==. & industry[_n-3] !=.
 & not_employed ==1
bysort persnr: replace ind5 = industry[_n-4] if ind5[_n]==. & industry[_n-4] !=.
 & not_employed ==1
bysort persnr: replace ind5 = industry[_n-5] if ind5[_n]==. & industry[_n-5] !=.
 & not_employed ==1

sort ind1 jahr
bysort ind1 jahr: egen double anz1 = count(persnr) if ind1!=.  & not_employed !=
.
bysort ind1 jahr: egen double sum1 = sum(not_employed)  if ind1!=.  &
not_employed != .
gen double risk1 = sum1 / anz1 if ind1!=.  & not_employed != .

bysort ind5 jahr: egen double anz5 = count(persnr)  if ind5!=.  & not_employed
!= .
bysort ind5 jahr: egen double sum5 = sum(not_employed)  if ind5!=.  &
not_employed != .
gen double risk5 = sum5 / anz5  if ind5!=.  & not_employed != .

bysort ind_full jahr: egen AL1_pgen = max(risk1) if jahr!= 1984
bysort ind_full jahr: egen AL5_pgen = max(risk5) if jahr!= 1984

label var industry "nace industry code"
label var AL1_pgen "AL-Risiko - 1 Jahr Definition - 2 digit industry code"
label var industry "nace industry code"
label var AL5_pgen "AL-Risiko - 5 Jahr Definition - 2 digit industry code"

rename ind_full ind_pgen
keep AL1_pgen AL5_pgen persnr jahr jahr_indf_pgen ind_pgen
sort persnr jahr
save ALrisiko_pgen, replace

* ALrisiko_pequiv.dta

cd "$wdirD"
use PEQUIV-Variables, clear
keep persnr jahr industry
recode industry (-3 -2 -1 0= .)

sort persnr jahr
merge m:m persnr jahr using PGEN-Variables
keep if _merge == 3
drop _merge nace
recode emplst lfs (-3 -2 -1 = .)

gen not_employed = .
```

```
replace not_employed = 1 if emplst == 5
replace not_employed = 0 if emplst == 1 | emplst == 2 |emplst == 3 | emplst == 4
| emplst == 6
drop if not_employed == .
        label var not_employed "Employment Status = not employed"

sort persnr jahr
by persnr (jahr): egen perioden_soep = count(jahr)
by persnr (jahr): egen perioden_branche = count(industry)
drop if perioden_branche == 0

sort persnr jahr
keep persnr jahr industry not_employed
save risiko_1DIC, replace

use risiko_1DIC, clear

keep persnr
duplicates drop persnr, force
expand 29
sort persnr
by persnr: gen jahr=_n
replace jahr = 1983+jahr
merge 1:1 persnr jahr using risiko_1DIC.dta
drop _merge

gen ind_full = industry
bysort persnr (jahr): replace ind_full = ind_full[_n-1] if ind_full[_n] == .

sort persnr jahr
foreach jahr in 2011 2010 2009 2008 2007 2006 2005 2004 2003 2002 2001 2000 1999
1998 1997 1996 1995 1994 1993 1992 1991 1990 1989 1988 1987 1986 1985 1984 {
bysort persnr: replace ind_full  = ind_full[_n+1] if jahr == `jahr' & ind_full
== .
}

gen jahr_ind=jahr if industry!=.
bysort persnr: egen jahr_indf_pequiv = min(jahr_ind)

sort persnr jahr
gen ind1=industry if not_employed==0
bysort persnr : replace ind1 = industry[_n-1] if industry[_n]==. &
not_employed==1

gen ind5=industry if not_employed==0
bysort persnr : replace ind5 = industry[_n-1] if ind5[_n]==. & industry[_n-1]
!=. & not_employed==1
bysort persnr : replace ind5 = industry[_n-2] if ind5[_n]==. & industry[_n-2]
!=. & not_employed==1
bysort persnr : replace ind5 = industry[_n-3] if ind5[_n]==. & industry[_n-3]
!=. & not_employed==1
bysort persnr : replace ind5 = industry[_n-4] if ind5[_n]==. & industry[_n-4]
!=. & not_employed==1
bysort persnr : replace ind5 = industry[_n-5] if ind5[_n]==. & industry[_n-5]
!=. & not_employed==1

sort ind1 jahr
bysort ind1 jahr: egen double anz1 = count(persnr) if ind1!=.  & not_employed!=.
bysort ind1 jahr: egen double sum1 = sum(not_employed)  if ind1!=. &
not_employed!=.
gen double risk1 = sum1 / anz1 if ind1!=. & not_employed!=.

bysort ind5 jahr: egen double anz5 = count(persnr)  if ind5!=. & not_employed!=.
bysort ind5 jahr: egen double sum5 = sum(not_employed)  if ind5!=. &
not_employed!=.
gen double risk5 = sum5 / anz5  if ind5!=. & not_employed!=.

bysort ind_full jahr: egen AL1_pequiv = max(risk1) if jahr!= 1984
```

```
bysort ind_full jahr: egen AL5_pequiv = max(risk5) if jahr!= 1984

label var AL1_pequiv "AL-Risiko - 1 Jahr Definition - 1 digit industry code"
label var industry "nace industry code"
label var AL5_pequiv "AL-Risiko - 5 Jahr Definition - 1 digit industry code"

rename ind_full ind_equiv
keep AL1_pequiv AL5_pequiv persnr jahr jahr_indf_pequiv ind_equiv
sort persnr jahr
save ALrisiko_pequiv, replace

*************************************************************************************
************************************
// Information on sons
*************************************************************************************
************************************

cd "$wdirD"
use Arbeitslosigkeit.dta, clear

merge m:1 persnr using Kindcharakteristika_Sohn.dta
keep if _merge==3
drop _merge

recode gebjahr (-1=.)
gen alter=jahr - gebjahr
        label var alter "Alter Sohn"
        renvars emplst lfs AL_1 AL_2 gebjahr migback alter nace   \ emplst_Sohn
lfs_Sohn AL_1_Sohn AL_2_Sohn gebjahr_Sohn migback_Sohn alter_Sohn nace_Sohn

keep if alter >=17 & alter <= 24
keep if sex == 1
drop sex

sort persnr jahr
drop if AL_1 ==. | AL_2 ==.

bysort persnr: egen Count = count(persnr)
bysort persnr: egen ANZ_1_Sohn = sum(AL_1)
bysort persnr: egen ANZ_2_Sohn = sum(AL_2)
bysort persnr: gen Miss_Sohn = 8 - Count

label var ANZ_1_Sohn "Number of years unemployed son - Def #1"
label var ANZ_2_Sohn "Number of years unemployed son - Def #2"
label var Miss_Sohn "Number of years not responded between age 17 and 24"

keep persnr jahr hhnrakt emplst_Sohn lfs_Sohn AL_1_Sohn AL_2_Sohn gebjahr_Sohn
migback_Sohn alter_Sohn ANZ_1_Sohn ANZ_2_Sohn Miss_Sohn nace_Sohn school college

save Sohninfo , replace

sort persnr
merge m:1 persnr using Geschwister.dta
keep if _merge == 3
drop _merge

sort persnr
merge m:1 persnr using Reihenfolge_Geschwister.dta
drop if _merge == 2
drop _merge

sort hhnrakt jahr
merge m:1 hhnrakt jahr using Bula_HH.dta
drop if _merge == 2
drop _merge

glo X Geschwister birthorder hhnrakt bula  college school
foreach var in $X{
```

```
rename `var' `var'_Sohn
}

save Sohninfo.dta , replace

*************************************************************************************
***********************************
// Combinations father and sons
*************************************************************************************
***********************************

cd "$wdirD"
use Sohninfo.dta, clear

duplicates drop persnr, force
keep persnr gebjahr

cd "$wdir"
merge 1:1 persnr using bioparen , keepusing(vnr)
keep if _merge==3
drop _merge

rename gebjahr gebjahr_kind
rename persnr ID
rename vnr persnr

keep if persnr != -3 & persnr != -2 & persnr != -1
count
cd "$wdirD"
save Vater_Sohn_Kombinationen.dta, replace

cd "$wdirD"
use Vater_Sohn_Kombinationen.dta, clear
sort persnr gebjahr_kind

gen Gebjahr_Kind1=.
replace Gebjahr_Kind1=gebjahr_kind if persnr[_n] != persnr[_n-1]

gen Gebjahr_Kind2=.
replace Gebjahr_Kind2=gebjahr_kind if Gebjahr_Kind1[_n]==. & Gebjahr_Kind1[_n-1]
!=.

gen Gebjahr_Kind3=.
replace Gebjahr_Kind3=gebjahr_kind if Gebjahr_Kind1[_n]==. &
Gebjahr_Kind2[_n]==. & Gebjahr_Kind2[_n-1]!=.

gen Gebjahr_Kind4=.
replace Gebjahr_Kind4=gebjahr_kind if Gebjahr_Kind1[_n]==. &
Gebjahr_Kind2[_n]==. & Gebjahr_Kind3[_n]==. & Gebjahr_Kind3[_n-1]!=.

gen Gebjahr_Kind5=.
replace Gebjahr_Kind5=gebjahr_kind if Gebjahr_Kind1[_n]==. &
Gebjahr_Kind2[_n]==. &  Gebjahr_Kind3[_n]==. &  Gebjahr_Kind4[_n]==. &
Gebjahr_Kind4[_n-1]!=.

gen Gebjahr_Kind6=.
replace Gebjahr_Kind6=gebjahr_kind if Gebjahr_Kind1[_n]==. &
Gebjahr_Kind2[_n]==. &  Gebjahr_Kind3[_n]==. & Gebjahr_Kind4[_n]==. &
Gebjahr_Kind5[_n]==. & Gebjahr_Kind5[_n-1]!=.

gen Gebjahr_Kind7=.
replace Gebjahr_Kind7=gebjahr_kind if Gebjahr_Kind1[_n]==. &
Gebjahr_Kind2[_n]==. &  Gebjahr_Kind3[_n]==. & Gebjahr_Kind4[_n]==. &
Gebjahr_Kind5[_n]==. &  Gebjahr_Kind6[_n]==. & Gebjahr_Kind6[_n-1]!=.


forvalues i = 1/7{
recode Gebjahr_Kind`i' (. = 0)
```

```
}
forvalues i=1/7{
bysort persnr (ID): egen K`i' = max(Gebjahr_Kind`i')
        label var K`i' "Geburtsjahr Kind #`i'"
}
forvalues i=1/7{
gen ID_Kind`i' =.
replace ID_Kind`i' = ID if Gebjahr_Kind`i' !=0
}
forvalues i = 1/7{
recode ID_Kind`i' (. = 0)
}
forvalues i=1/7{
bysort persnr (ID): egen ID_K`i' = max(ID_Kind`i')
        label var ID_K`i' "Personennummer Kind #`i'"
}
duplicates drop persnr , force

keep persnr K1-K7 ID_K1-ID_K7
recode K1 - K7 (0 = .)
recode ID_K1 - ID_K7 (0 = .)

renvars K1-K7 \ gebjahr_kind1 gebjahr_kind2 gebjahr_kind3 gebjahr_kind4
gebjahr_kind5 gebjahr_kind6 gebjahr_kind7
renvars ID_K1-ID_K7 \ persnr_kind1 persnr_kind2 persnr_kind3 persnr_kind4
persnr_kind5 persnr_kind6 persnr_kind7

cd "$wdirD"
sort persnr
save Vater_Sohn_Kombinationen2, replace

forvalues i=1/7{
use Vater_Sohn_Kombinationen2, clear
keep persnr gebjahr_kind`i' persnr_kind`i'
drop if gebjahr_kind`i'==.
save Vater_Sohn`i', replace
}
erase Vater_Sohn_Kombinationen2.dta

*************************************************************************************
***********************************
// State of residence father (son aged 10)
*************************************************************************************
***********************************

cd "$wdirD"
forval i =1/7{
use Vater_Sohn`i', clear
sort persnr
merge 1:m persnr using PGEN-Variables.dta
keep if _merge == 3
gen alter_kind=jahr-gebjahr_kind`i'
keep if alter_kind ==10
sort persnr jahr

drop _merge nace lfs emplst school college gebjahr_kind alter_kind
sort hhnrakt jahr

merge 1:1 hhnrakt jahr using Bula_HH.dta
keep if _merge == 3
drop _merge  hhnr hhnrakt jahr

save BulaVater_Sohn_10`i'.dta, replace
}

use BulaVater_Sohn_101.dta, clear
forval i=2/7{
append using BulaVater_Sohn_10`i'.dta
```

```
}

forval i=1/7{
erase BulaVater_Sohn_10`i'.dta
 }

 drop persnr
 gen persnr=.
 forval i=1/7{
 replace persnr = persnr_kind`i' if persnr ==.
 drop persnr_kind`i'
 }

sort persnr
save BulaVater_Sohn10.dta, replace


*******************************************************************************
***********************************
// Unemployment father t0
*******************************************************************************
***********************************

cd "$wdirD"
forvalues i=1/7{
use Vater_Sohn`i', clear
sort persnr
merge 1:m persnr using Arbeitslosigkeit.dta
keep if _merge==3
drop _merge hhnrakt emplst lfs nace school college

rename persnr nr_vater
sort nr_vater jahr
gen alter_kind=jahr-gebjahr_kind`i'
        label var alter_kind "Alter des Kindes"
keep if alter_kind >=10 & alter_kind <= 15

forvalues u=1/2{
bysort nr_vater (jahr): egen Vater_AL_`u' = max(AL_`u')
        label var Vater_AL_`u' "Vater jemals arbeitslos (AL `u')? als Kind
zwischen 10 und 15?"
        }
bysort nr_vater: egen Count_Vater_before = count(nr_vater)
bysort nr_vater: egen ANZ_1_Vater_before = sum(AL_1)
bysort nr_vater: egen ANZ_2_Vater_before = sum(AL_2)
bysort nr_vater: gen Miss_Vater_before = 6 - Count

label var ANZ_1_Vater_before "Number of years unemployed father before- Def #1"
label var ANZ_2_Vater_before "Number of years unemployed father before - Def #2"
label var Miss_Vater_before "Number of years not responded father before"


duplicates drop nr_vater, force
keep nr_vater persnr_kind`i' gebjahr_kind`i' Vater_AL_1 Vater_AL_2
persnr_kind`i' ANZ_1_Vater_before ANZ_2_Vater_before Miss_Vater_before
Count_Vater_before

rename gebjahr_kind`i' gebjahr_Sohn
rename persnr_kind`i' persnr_Sohn
save AL_Vater_Sohn`i', replace
}
use AL_Vater_Sohn1, clear
forval i=2/7{
append using AL_Vater_Sohn`i'
}

forval i=1/7{
erase AL_Vater_Sohn`i'.dta
```

```
}
label var persnr_Sohn "Personennr. Sohn"

sort nr_vater
save Vaterarbeitslosigkeit_Sohn.dta, replace

*******************************************************************************
***********************************
// Unemployment father t2
*******************************************************************************
***********************************

cd "$wdirD"
forvalues i=1/6{

use Vater_Sohn`i', clear
sort persnr
merge 1:m persnr using Arbeitslosigkeit.dta
keep if _merge==3
drop _merge hhnrakt emplst lfs nace school college

rename persnr nr_vater
sort nr_vater jahr
gen alter_kind=jahr-gebjahr_kind`i'
        label var alter_kind "Alter des Kindes"
keep if alter_kind >=25 & alter_kind <= 30

forvalues u=1/2{
bysort nr_vater (jahr): egen Vater_Kind`i'_AL_`u' = max(AL_`u')
        label var Vater_Kind`i'_AL_`u' "Vater jemals arbeitslos (AL `u', Kind
`i') als Kind im Alter zwischen 25 und 30?"
}
bysort nr_vater: egen Count_Vater_after = count(nr_vater)
bysort nr_vater: egen ANZ_1_Vater_after = sum(AL_1)
bysort nr_vater: egen ANZ_2_Vater_after = sum(AL_2)
bysort nr_vater: gen Miss_Vater_after = 6 - Count

label var ANZ_1_Vater_after "Number of years unemployed father after- Def #1"
label var ANZ_2_Vater_after "Number of years unemployed father after - Def #2"
label var Miss_Vater_after "Number of years not responded father after"

duplicates drop nr_vater, force
keep nr_vater persnr_kind`i' Vater_Kind`i'_AL_1 Vater_Kind`i'_AL_2
persnr_kind`i' ANZ_1_Vater_after ANZ_2_Vater_after Miss_Vater_after
Count_Vater_after
rename persnr_kind`i' persnr_Sohn
save AL_Vater_Sohn`i'_after, replace
}
use AL_Vater_Sohn1_after, clear
forval i=2/6{
append using AL_Vater_Sohn`i'_after
}

forval i=1/6{
erase AL_Vater_Sohn`i'_after.dta
}
label var persnr_Sohn "persnr Sohn"
drop Vater_Kind1_AL_1 Vater_Kind1_AL_2 Vater_Kind2_AL_1-Vater_Kind6_AL_2
gen persnr = persnr_Sohn
sort persnr
save Vaterarbeitslosigkeit_Sohn_after.dta, replace

*******************************************************************************
***********************************
// Unemployment father (robustness test - son age 10-13)
*******************************************************************************
***********************************
```

```
cd "$wdirD"
forvalues i=1/7{
use Vater_Sohn`i', clear
sort persnr
merge 1:m persnr using Arbeitslosigkeit.dta
keep if _merge==3
drop _merge hhnrakt emplst lfs nace school college

rename persnr nr_vater
sort nr_vater jahr
gen alter_kind=jahr-gebjahr_kind`i'
        label var alter_kind "Alter des Kindes"
keep if alter_kind >=10 & alter_kind <= 13

forvalues u=1/2{
bysort nr_vater (jahr): egen Vater_AL_`u' = max(AL_`u')
        label var Vater_AL_`u' "Vater jemals arbeitslos (AL `u')? als Kind
zwischen 10 und 13?"
        }
bysort nr_vater: egen Count_Vater_before = count(nr_vater)
bysort nr_vater: egen ANZ_1_Vater_before = sum(AL_1)
bysort nr_vater: egen ANZ_2_Vater_before = sum(AL_2)
bysort nr_vater: gen Miss_Vater_before = 4 - Count

label var ANZ_1_Vater_before "Number of years unemployed father before- Def #1"
label var ANZ_2_Vater_before "Number of years unemployed father before - Def #2"
label var Miss_Vater_before "Number of years not responded father before"

duplicates drop nr_vater, force
keep nr_vater persnr_kind`i' gebjahr_kind`i' Vater_AL_1 Vater_AL_2
persnr_kind`i' ANZ_1_Vater_before ANZ_2_Vater_before Miss_Vater_before
Count_Vater_before

rename gebjahr_kind`i' gebjahr_Sohn
rename persnr_kind`i' persnr_Sohn
save AL_Vater_Sohn`i'_10_13, replace
}
use AL_Vater_Sohn1_10_13, clear
forval i=2/7{
append using AL_Vater_Sohn`i'_10_13
}

forval i=1/7{
erase AL_Vater_Sohn`i'_10_13.dta
}
label var persnr_Sohn "Personennr. Sohn"

sort nr_vater
save Vaterarbeitslosigkeit_Sohn_10_13.dta, replace

*************************************************************************************
************************************
// Merge
*************************************************************************************
************************************

cd "$wdirD"
use Vaterarbeitslosigkeit_Sohn, clear

gen persnr=persnr_Sohn
sort persnr
merge 1:1 persnr using Vaterbildung.dta
keep if _merge == 3
drop _merge
sort persnr
merge 1:1 persnr using BulaVater_Sohn10.dta
drop if _merge == 2
drop _merge
```

```
sort persnr
merge 1:1 persnr using Vaterarbeitslosigkeit_Sohn_after.dta
drop if _merge ==2
drop _merge

replace persnr=nr_vater
sort persnr
merge m:1 persnr using Vatercharakteristika.dta
drop if _merge == 2
drop _merge

gen jahr = gebjahr_Sohn
sort persnr jahr
merge m:1 persnr jahr using Initial_Industry_1DIC.dta
drop if _merge == 2
drop _merge

sort persnr jahr
merge m:1 persnr jahr using Initial_Industry_2DIC.dta
drop if _merge == 2
drop _merge

drop jahr

renvars  bula\  bula_Vater

gen jahr = gebjahr_Sohn + 8

sort persnr jahr
merge m:1 persnr jahr using ALrisiko_pgen
drop if _merge == 2
drop _merge

sort persnr jahr
merge m:1 persnr jahr using ALrisiko_pequiv
drop if _merge == 2
drop _m

drop jahr

replace persnr=persnr_Sohn

sort persnr
merge m:m persnr using Sohninfo.dta
keep if _merge == 3
drop _merge

qui tab school, m gen(D_School_)

forval i=1/8{
bysort persnr_Sohn: egen MAX_School_`i' =max(D_School_`i')
drop D_School_`i'
}
gen SchoolMAX =.
replace SchoolMAX = 4 if MAX_School_4 == 1  & school !=.
replace SchoolMAX = 3 if MAX_School_3 == 1 & MAX_School_4 != 1 & school !=.
replace SchoolMAX = 2 if MAX_School_2 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & school !=.
replace SchoolMAX = 1 if MAX_School_1 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & school !=.
replace SchoolMAX = 5 if MAX_School_5 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & school !=.
replace SchoolMAX = 7 if MAX_School_7 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & MAX_School_5 !=1 & school !=.
replace SchoolMAX = 6 if MAX_School_6 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & MAX_School_5 !=1 & MAX_School_7 !=1 &
school !=.
```

```
label values SchoolMAX apsbil_EN

*******************************************************************************
***********************************
// Final sample / definition variables
*******************************************************************************
***********************************

* Education son
gen sonbil = school_Sohn
recode sonbil (4 = 3) (5 6 . = 4) (7=5), copyrest
label define SBIL 1 "Lower secondary school degree (Hauptschulabschluss)"  2
"Intermediate school degree (Mittlere Reife)" ///
                        3 "Upper secondary school degree (Abitur)/Technical
secondary school degree (Fachhochschule)" 4 "Other degree / No school degreee /
Missing information" 5 "Currently in school"
label values sonbil SBIL

*  Siblings
gen sibs = Geschwister_Sohn
replace sibs = 4 if sibs > 4

* Birth order
gen birtho = birthorder_Sohn
replace birtho = 4 if birthorder_Sohn >3 & birthorder_Sohn!= .
replace birtho = 99 if birthorder_Sohn ==.

* Fathers postsecondary education
gen fathbil = 1 if Vater_hBildung == 10 | Vater_hBildung >= 50
replace fathbil = 2 if Vater_hBildung > = 20 & Vater_hBildung < 24
replace fathbil = 2 if Vater_hBildung == 40
replace fathbil = 3 if Vater_hBildung > = 24 & Vater_hBildung < 27
replace fathbil = 4 if Vater_hBildung > = 27 &  Vater_hBildung < 30
replace fathbil = 5 if Vater_hBildung > = 30 & Vater_hBildung < 40
replace fathbil = 99 if Vater_hBildung == .
label define VBIL 1 "No postsecondary education" 2 "Vocational training" 3
"Industrial/commercial/health care apprenticeship" 4 "Technical college, Civil
servant training" 5 "University" 99 "Missing"
label values fathbil VBIL

* Fathers secondary education
gen fathschool = Vater_Schule
replace fathschool = 99 if Vater_Schule == .
replace fathschool = 6 if Vater_Schule > 5 & Vater_Schule<99
label values fathschool vsbil_EN

* Fathers year of birth
gen fathgeb = 1 if gebjahr_Vater < 1940
replace fathgeb = 2 if gebjahr_Vater < 1940 + 10 & gebjahr_Vater >=  1940
replace fathgeb = 3 if gebjahr_Vater < 1940 + 20 & gebjahr_Vater >=  1940 + 10
replace fathgeb = 4 if gebjahr_Vater < 1940 + 30 & gebjahr_Vater >=  1940 + 20
replace fathgeb = 5 if gebjahr_Vater < 1940 + 40 & gebjahr_Vater >=  1940 + 30
label define fcat 1 "1929-1939"  2 "1940-1949" 3 "1950-1959" 4 "1960-1969" 5
"1970-1975"
label values fathgeb fcat

* Sons year of birth
gen songeb = 1 if gebjahr_Sohn < 1979
replace songeb = 2 if gebjahr_Sohn < 1979 + 2 &  gebjahr_Sohn >= 1979
replace songeb = 3 if gebjahr_Sohn < 1979 + 4 &  gebjahr_Sohn >= 1979 + 2
replace songeb = 4 if gebjahr_Sohn < 1979 + 6 &  gebjahr_Sohn >= 1979 + 4
replace songeb = 5 if gebjahr_Sohn < 1979 + 8 &  gebjahr_Sohn >= 1979 + 6
replace songeb = 6 if gebjahr_Sohn < 1979 + 10 &  gebjahr_Sohn >= 1979 + 8
replace songeb = 7 if gebjahr_Sohn < 1979 + 12 &  gebjahr_Sohn >= 1979 + 10
replace songeb = 8 if gebjahr_Sohn < 1979 + 14 &  gebjahr_Sohn >= 1979 + 12
replace songeb = 9 if gebjahr_Sohn < 1979 + 16 &  gebjahr_Sohn >= 1979 + 14
replace songeb = 10 if gebjahr_Sohn < 1979 + 18 &  gebjahr_Sohn >= 1979 + 16
```

```
label define scat 1 "1977-1978" 2 "1979-1980" 3 "1981-1982" 4 "1983-1984" 5
"1985-1986" 6 "1987-1988" 7 "1989-1990" 8 "1991-1992" 9 "1993-1994" 10 "1995"
label values songeb scat

* State Dummies
replace bula_Vater = bula_Sohn if bula_Vater == .
gen east = bula_Vater > 10
label var east "Father lived in East Germany at sons age 10"
gen state = bula_Vater

* Migback
replace migback_Vater = migback_Sohn if migback_Vater < 0 | migback_Vater == 4
label values migback_Vater migback_EN

foreach var in songeb fathgeb fathschool fathbil sibs birtho migback_Vater state
sonbil nace_Sohn {
qui tab `var', gen(`var'_) m
}
 rename   ANZ_1_Vater_before ALbef_fath
 rename   ANZ_1_Vater_after ALaft_fath
 rename   ANZ_2_Sohn AL_son

 global varlabel                    ///
 fathbil_1 "No postsecondary education"    ///
 fathbil_2 "Other vocational training"  ///
 fathbil_3 "Industrial/commercial/health care apprenticeship"   ///
 fathbil_4 "Technical college, civil servant training"  ///
 fathbil_5 "University degree"  ///
 fathbil_6 "Missing information"  ///
 fathgeb_1 "1929-1939"       ///
 fathgeb_2 "1940-1949"  ///
 fathgeb_3 "1950-1959"  ///
 fathgeb_4 "1960-1969"  ///
 fathgeb_5 "1970-1975"  ///
 songeb_1 "1977-1978"   ///
 songeb_2 "1979-1980"   ///
 songeb_3 "1981-1982"   ///
 songeb_4 "1983-1984"   ///
 songeb_5 "1985-1986"   ///
 songeb_6 "1987-1988"   ///
 songeb_7 "1989-1990"   ///
 songeb_8 "1991-1992"   ///
 songeb_9 "1993-1994"   ///
 songeb_10 "1995"  ///
 fathschool_1 "Lower secondary school degree (Hauptschulabschluss)"  ///
 fathschool_2 "Intermediate school degree (Mittlere Reife)"  ///
 fathschool_3 "Technical school degree (Fachhochschulreife)"  ///
 fathschool_4 "Upper secondary school degree (Abitur)"  ///
 fathschool_5 "Other degree"  ///
 fathschool_6 "No school degree"   ///
 fathschool_7 "Missing information"  ///
 sonbil_1 "Lower secondary school degree (Hauptschulabschluss)"  ///
 sonbil_2 "Intermediate school degree (Mittlere Reife)"  ///
 sonbil_3 "Upper secondary school degree (Abitur)"  ///
 sonbil_4 "Other degree / No school degreee / Missing information"  ///
 sonbil_5 "Currently in school"  ///
 birtho_1 "1st born"  ///
 birtho_2 "2nd born"  ///
 birtho_3 "3rd born"  ///
 birtho_4 "4th bornh"  ///
 birtho_5 "Missing information"  ///
 sibs_1 "No siblings"  ///
 sibs_2 "1 sibling"  ///
 sibs_3 "2 siblings"  ///
 sibs_4 "3 siblings"  ///
 sibs_5 ">4 siblings"  ///
 migback_Vater_1 "No migration background" ///
```

```
 migback_Vater_2 "Direct migration background" ///
 migback_Vater_3 "Indirect migration background" ///
 Miss_Sohn "Number of years son not observed age 17-24" ///
 Miss_Vater_before "Number of years father not observed (son aged 10-15)" ///
 east "Father lived in East Germany at sons' age 10" ///
 ALbef_fath  "Years father unemployed while son aged 10-15" ///
 ALaft_fath  "Years father unemployed while son aged 25-30" ///
 AL_son "Number of years son workless" ///
gebjahr_Sohn "Sons' year of birth" ///
birthorder_Sohn "Sons' birthorder" ///
Geschwister_Sohn "Sons' number of siblings" ///
gebjahr_Vater "Fathers' year of birth"

  label variable gebjahr_Vater "Fathers' year of birth"
 label variable gebjahr_Sohn "Sons' year of birth"
 label variable birthorder_Sohn "Sons' birthorder"
 label variable Geschwister_Sohn "Sons' number of siblings"
 label variable migback_Vater "Fathers' migration background"

drop MAX_School* Vater_AL_2 ANZ_2_Vater_before ANZ_2_Vater_after persnr_Sohn
ind1dic ind2dic ind_pgen jahr_indf_pgen emplst_Sohn lfs_Sohn college_Sohn ///
 ind_equiv jahr_indf_pequiv hhnrakt_Sohn AL_1_Sohn ANZ_1_Sohn Vater_Schule
Vater_Beruf alter_Sohn bula_Sohn  migback_Sohn bula_Vater

save panel.dta , replace

*************************************************************************************
************************************
// Generate cross-section data set (final sample)
*************************************************************************************
************************************

sort persnr jahr
duplicates drop persnr , force

drop school SchoolMAX
replace AL1_pgen = AL1_pequiv if AL1_pgen==. & AL1_pequiv!=.
drop AL1_pe AL5_pe

save finalsample.dta , replace

*************************************************************************************
************************************
// Alternative data set for robustness test (son aged 10-13)
*************************************************************************************
************************************

cd "$wdirD"
use Vaterarbeitslosigkeit_Sohn_10_13, clear

gen persnr=persnr_Sohn
sort persnr
merge 1:1 persnr using Vaterbildung.dta
keep if _merge == 3
drop _merge
sort persnr
merge 1:1 persnr using BulaVater_Sohn10.dta
drop if _merge == 2
drop _merge

sort persnr
merge 1:1 persnr using Vaterarbeitslosigkeit_Sohn_after.dta
drop if _merge ==2
drop _merge

replace persnr=nr_vater
sort persnr
merge m:1 persnr using Vatercharakteristika.dta
```

```
drop if _merge == 2
drop _merge

gen jahr = gebjahr_Sohn
sort persnr jahr
merge m:1 persnr jahr using Initial_Industry_1DIC.dta
drop if _merge == 2
drop _merge

sort persnr jahr
merge m:1 persnr jahr using Initial_Industry_2DIC.dta
drop if _merge == 2
drop _merge

drop jahr
renvars  bula\  bula_Vater
gen jahr = gebjahr_Sohn + 8

sort persnr jahr
merge m:1 persnr jahr using ALrisiko_pgen
drop if _merge == 2
drop _merge

sort persnr jahr
merge m:1 persnr jahr using ALrisiko_pequiv
drop if _merge == 2
drop _merge

drop jahr
replace persnr=persnr_Sohn

sort persnr
merge m:m persnr using Sohninfo.dta
keep if _merge == 3
drop _merge
qui tab school, m gen(D_School_)

forval i=1/8{
bysort persnr_Sohn: egen MAX_School_`i' =max(D_School_`i')
drop D_School_`i'
}
gen SchoolMAX =.
replace SchoolMAX = 4 if MAX_School_4 == 1  & school !=.
replace SchoolMAX = 3 if MAX_School_3 == 1 & MAX_School_4 != 1 & school !=.
replace SchoolMAX = 2 if MAX_School_2 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & school !=.
replace SchoolMAX = 1 if MAX_School_1 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & school !=.
replace SchoolMAX = 5 if MAX_School_5 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & school !=.
replace SchoolMAX = 7 if MAX_School_7 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & MAX_School_5 !=1 & school !=.
replace SchoolMAX = 6 if MAX_School_6 == 1 & MAX_School_4 != 1 & MAX_School_3 !=
1 & MAX_School_2 != 1 & MAX_School_1 !=1 & MAX_School_5 !=1 & MAX_School_7 !=1 &
school !=.

label values SchoolMAX apsbil_EN
sort persnr_Sohn jahr
duplicates drop persnr_Sohn , force

drop school
rename SchoolMAX school_Sohn
replace AL1_pgen = AL1_pequiv if AL1_pgen==. & AL1_pequiv!=.

* Education son
gen sonbil = school_Sohn
recode sonbil (4 = 3) (5 6 . = 4) (7=5), copyrest
label define SBIL 1 "Lower secondary school degree (Hauptschulabschluss)"  2
```

```
"Intermediate school degree (Mittlere Reife)" ///
                        3 "Upper secondary school degree (Abitur)/Technical
secondary school degree (Fachhochschule)" 4 "Other degree / No school degreee /
Missing information" 5 "Currently in school"
label values sonbil SBIL

*  Siblings
gen sibs = Geschwister_Sohn
replace sibs = 4 if sibs > 4

* Birth order
gen birtho = birthorder_Sohn
replace birtho = 4 if birthorder_Sohn >3 & birthorder_Sohn!= .
replace birtho = 99 if birthorder_Sohn ==.

* Fathers postsecondary education
gen fathbil = 1 if Vater_hBildung == 10 | Vater_hBildung >= 50
replace fathbil = 2 if Vater_hBildung > = 20 & Vater_hBildung < 24
replace fathbil = 2 if Vater_hBildung == 40
replace fathbil = 3 if Vater_hBildung > = 24 & Vater_hBildung < 27
replace fathbil = 4 if Vater_hBildung > = 27 &  Vater_hBildung < 30
replace fathbil = 5 if Vater_hBildung > = 30 & Vater_hBildung < 40
replace fathbil = 99 if Vater_hBildung == .
label define VBIL 1 "No postsecondary education" 2 "Vocational training" 3
"Industrial/commercial/health care apprenticeship" 4 "Technical college, Civil
servant training" 5 "University" 99 "Missing"
label values fathbil VBIL

* Fathers secondary education
gen fathschool = Vater_Schule
replace fathschool = 99 if Vater_Schule == .
replace fathschool = 6 if Vater_Schule > 5 & Vater_Schule<99
label values fathschool vsbil_EN

* Fathers year of birth
gen fathgeb = 1 if gebjahr_Vater < 1940
replace fathgeb = 2 if gebjahr_Vater < 1940 + 10 & gebjahr_Vater >=  1940
replace fathgeb = 3 if gebjahr_Vater < 1940 + 20 & gebjahr_Vater >=  1940 + 10
replace fathgeb = 4 if gebjahr_Vater < 1940 + 30 & gebjahr_Vater >=  1940 + 20
replace fathgeb = 5 if gebjahr_Vater < 1940 + 40 & gebjahr_Vater >=  1940 + 30
label define fcat 1 "1929-1939"  2 "1940-1949" 3 "1950-1959" 4 "1960-1969" 5
"1970-1975"
label values fathgeb fcat

* Sons year of birth
gen songeb = 1 if gebjahr_Sohn < 1979
replace songeb = 2 if gebjahr_Sohn < 1979 + 2 &  gebjahr_Sohn >= 1979
replace songeb = 3 if gebjahr_Sohn < 1979 + 4 &  gebjahr_Sohn >= 1979 + 2
replace songeb = 4 if gebjahr_Sohn < 1979 + 6 &  gebjahr_Sohn >= 1979 + 4
replace songeb = 5 if gebjahr_Sohn < 1979 + 8 &  gebjahr_Sohn >= 1979 + 6
replace songeb = 6 if gebjahr_Sohn < 1979 + 10 &  gebjahr_Sohn >= 1979 + 8
replace songeb = 7 if gebjahr_Sohn < 1979 + 12 &  gebjahr_Sohn >= 1979 + 10
replace songeb = 8 if gebjahr_Sohn < 1979 + 14 &  gebjahr_Sohn >= 1979 + 12
replace songeb = 9 if gebjahr_Sohn < 1979 + 16 &  gebjahr_Sohn >= 1979 + 14
replace songeb = 10 if gebjahr_Sohn < 1979 + 18 &  gebjahr_Sohn >= 1979 + 16

label define scat 1 "1977-1978" 2 "1979-1980" 3 "1981-1982" 4 "1983-1984" 5
"1985-1986" 6 "1987-1988" 7 "1989-1990" 8 "1991-1992" 9 "1993-1994" 10 "1995"
label values songeb scat

* State Dummies
replace bula_Vater = bula_Sohn if bula_Vater == .
gen east = bula_Vater > 10
label var east "Father lived in East Germany at sons age 10"
gen state = bula_Vater

* Migback
replace migback_Vater = migback_Sohn if migback_Vater < 0 | migback_Vater == 4
```

```
label values migback_Vater migback_EN

foreach var in songeb fathgeb fathschool fathbil sibs birtho migback_Vater state
sonbil nace_Sohn {
qui tab `var', gen(`var'_) m
}
 rename   ANZ_1_Vater_before ALbef_fath
 rename   ANZ_1_Vater_after ALaft_fath
 rename   ANZ_2_Sohn AL_son

 global varlabel                    ///
 fathbil_1 "No postsecondary education"    ///
 fathbil_2 "Other vocational training"  ///
 fathbil_3 "Industrial/commercial/health care apprenticeship"   ///
 fathbil_4 "Technical college, civil servant training"  ///
 fathbil_5 "University degree"  ///
 fathbil_6 "Missing information"  ///
 fathgeb_1 "1929-1939"       ///
 fathgeb_2 "1940-1949"  ///
 fathgeb_3 "1950-1959"  ///
 fathgeb_4 "1960-1969"  ///
 fathgeb_5 "1970-1975"  ///
 songeb_1 "1977-1978"   ///
 songeb_2 "1979-1980"   ///
 songeb_3 "1981-1982"   ///
 songeb_4 "1983-1984"   ///
 songeb_5 "1985-1986"   ///
 songeb_6 "1987-1988"   ///
 songeb_7 "1989-1990"   ///
 songeb_8 "1991-1992"   ///
 songeb_9 "1993-1994"   ///
 songeb_10 "1995"  ///
 fathschool_1 "Lower secondary school degree (Hauptschulabschluss)"  ///
 fathschool_2 "Intermediate school degree (Mittlere Reife)"  ///
 fathschool_3 "Technical school degree (Fachhochschulreife)"  ///
 fathschool_4 "Upper secondary school degree (Abitur)"  ///
 fathschool_5 "Other degree"  ///
 fathschool_6 "No school degree"   ///
 fathschool_7 "Missing information"  ///
 sonbil_1 "Lower secondary school degree (Hauptschulabschluss)"  ///
 sonbil_2 "Intermediate school degree (Mittlere Reife)"  ///
 sonbil_3 "Upper secondary school degree (Abitur)"  ///
 sonbil_4 "Other degree / No school degreee / Missing information"  ///
 sonbil_5 "Currently in school"  ///
 birtho_1 "1st born"  ///
 birtho_2 "2nd born"  ///
 birtho_3 "3rd born"  ///
 birtho_4 "4th bornh"  ///
 birtho_5 "Missing information"  ///
 sibs_1 "No siblings"  ///
 sibs_2 "1 sibling"  ///
 sibs_3 "2 siblings"  ///
 sibs_4 "3 siblings"  ///
 sibs_5 ">4 siblings"  ///
 migback_Vater_1 "No migration background" ///
 migback_Vater_2 "Direct migration background" ///
 migback_Vater_3 "Indirect migration background" ///
 Miss_Sohn "Number of years son not observed age 17-24" ///
 Miss_Vater_before "Number of years father not observed (son aged 10-15)" ///
 east "Father lived in East Germany at sons' age 10" ///
 ALbef_fath  "Years father unemployed while son aged 10-15" ///
 ALaft_fath  "Years father unemployed while son aged 25-30" ///
 AL_son "Number of years son workless" ///
gebjahr_Sohn "Sons' year of birth" ///
birthorder_Sohn "Sons' birthorder" ///
Geschwister_Sohn "Sons' number of siblings" ///
gebjahr_Vater "Fathers' year of birth"
```

```
 label variable gebjahr_Vater "Fathers' year of birth"
 label variable gebjahr_Sohn "Sons' year of birth"
 label variable birthorder_Sohn "Sons' birthorder"
 label variable Geschwister_Sohn "Sons' number of siblings"
 label variable migback_Vater "Fathers' migration background"

 drop MAX_School* Vater_AL_2 ANZ_2_Vater_before ANZ_2_Vater_after persnr_Sohn
ind1dic ind2dic ind_pgen jahr_indf_pgen emplst_Sohn lfs_Sohn college_Sohn ///
 ind_equiv jahr_indf_pequiv hhnrakt_Sohn AL_1_Sohn ANZ_1_Sohn Vater_Schule
Vater_Beruf alter_Sohn bula_Sohn  migback_Sohn bula_Vater

keep if  AL1_pgen !=.
save sons_10_13.dta, replace

*************************************************************************************
************************************
// Erase unnecessary data
*************************************************************************************
************************************

cd "$wdirD"
erase ALrisiko_pequiv.dta
erase ALrisiko_pgen.dta
erase Arbeitslosigkeit.dta
erase BIOPAREN-Variables_Mutter.dta
erase BIOPAREN-Variables_Vater.dta
erase Bula_HH.dta
erase BulaVater_Sohn10.dta
erase Geschwister.dta
erase HBRUTTO-Variables.dta
erase Initial_industry_1DIC.dta
erase Initial_Industry_2DIC.dta
erase Kindcharakteristika_Sohn.dta
erase Mutterbildung.dta
erase Muttercharakteristika.dta
erase PEQUIV-Variables.dta
erase PGEN-Variables.dta
erase Reihenfolge_Geschwister.dta
erase risiko_1DIC.dta
erase risiko_2DIC.dta
erase Sohninfo.dta
erase Vater_Sohn_Kombinationen.dta
erase Vater_Sohn1.dta
erase Vater_Sohn2.dta
erase Vater_Sohn3.dta
erase Vater_Sohn4.dta
erase Vater_Sohn5.dta
erase Vater_Sohn6.dta
erase Vater_Sohn7.dta
erase Vaterarbeitslosigkeit_Sohn.dta
erase Vaterarbeitslosigkeit_Sohn_10_13.dta
erase Vaterarbeitslosigkeit_Sohn_after.dta
erase Vaterbildung.dta
erase Vatercharakteristika.dta

*************************************************************************************
************************************
// Globals
*************************************************************************************
************************************

global var0 Miss_Sohn Miss_Vater_before
global var1 songeb_* fathgeb_* fathschool_* fathbil_* sibs_* birtho_*
global var2 migback_Vater_*
global var3 state_*

global var0het Miss_Sohn Miss_Vater_before
global var1het ib1.songeb ib1.fathgeb ib1.fathschool ib1.fathbil ib1.sibs
```

```
ib1.birtho
global var2het ib1.migback_Vater
global var3het ib1.state

global se cluster(nr_vater)

global descr AL_son  ALbef_fath  Miss_Sohn Miss_Vater_before ///
                        gebjahr_Sohn sonbil_* Geschwister_Sohn birthorder_Sohn
migback_Vater_*  ///
                        gebjahr_Vater east fathschool_* fathbil_*

*************************************************************************************
************************************
// Descriptives
*************************************************************************************
************************************

cd "$wdirD"
use finalsample, clear
keep if  AL1_pgen !=.

count if Vater_AL_1 ==0
count if Vater_AL_1 ==1
count


tabstat $descr , by(Vater_AL_1) stats( mean) save c(s)

tabstat $descr , by(Vater_AL_1) stats( sd) save c(s)

tabstat $descr ,  stats( mean) save c(s)

tabstat $descr , stats( sd) save c(s)



*************************************************************************************
************************************
// OLS and IV
*************************************************************************************
************************************

cd "$wdirD"
use finalsample , clear
keep if  AL1_pgen !=.


foreach ins in    AL1_pgen AL5_pgen {
 ivreg2 AL_son (ALbef_fath =`ins' )  ///
                , $se   first  savefirst

estimates restore _ivreg2_ALbef_fath
test `ins'


 ivreg2 AL_son $var0 $var1 $var2 $var3 (ALbef_fath =`ins' )  ///
                , $se   first  savefirst
estimates restore _ivreg2_ALbef_fath
test `ins'
}


 reg AL_son  ALbef_fath  , $se
 reg  AL_son  ALbef_fath   $var0 $var1 migback_Vater_2 migback_Vater_3 $var3  ,
$se
```

```
*************************************************************************************
************************************
// Gottschalk method
*************************************************************************************
************************************

*Panel A
cd "$wdirD"
use finalsample , clear

keep if Count_Vater_after !=.

 reg  AL_son  ALbef_fath  ALaft_fath  , $se
        nlcom _b[ALbef_fath]-_b[ALaft_fath], post



 reg  AL_son  ALbef_fath  ALaft_fath  Miss_Vater_after $var0 $var1 $var2 $var3
, $se
        nlcom _b[ALbef_fath]-_b[ALaft_fath], post


gen Vater_before =  ALbef_fath
 reg  AL_son  Vater_before if  ALaft_fath!=.  , $se
 reg  AL_son  Vater_before   $var0 $var1 $var2 $var3    if  ALaft_fath!=.  , $se




*Panel B
cd "$wdirD"
use finalsample, clear

keep if Count_Vater_after >=3 &  Count_Vater_after !=. & Count_Vater_before >= 3

 reg  AL_son  ALbef_fath  ALaft_fath  , $se
        nlcom _b[ALbef_fath]-_b[ALaft_fath], post


 reg  AL_son  ALbef_fath  ALaft_fath  Miss_Vater_after $var0 $var1 $var2 $var3
, $se
        nlcom _b[ALbef_fath]-_b[ALaft_fath], post



gen Vater_before =  ALbef_fath
 reg  AL_son  Vater_before if  ALaft_fath!=.  , $se
 reg  AL_son  Vater_before   $var0 $var1 $var2 $var3    if  ALaft_fath!=.  , $se




*************************************************************************************
************************************
// Heterogeneities
*************************************************************************************
************************************

cd "$wdirD"
use finalsample, clear
keep if  AL1_pgen !=.

gen west = 1-east
```

```
gen nomig=migback_Vater==1
gen mig=1-nomig
gen ledu = Vater_hBildung < = 10
replace ledu = 1 if Vater_hBildung>=40 & Vater_hBildung !=.
gen medu = (Vater_hBildung > = 20 & Vater_hBildung < = 26)
gen hedu = (Vater_hBildung > = 27 & Vater_hBildung < = 32)
replace ledu =. if Vater_hBildung ==.
replace medu =. if Vater_hBildung ==.
replace hedu =. if Vater_hBildung ==.


foreach region in east west {
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if  `region'==1, $se
 reg  AL_son  ALbef_fath   $var0het $var1het   if nomig==1 & `region'==1, $se
 reg  AL_son  ALbef_fath   $var0het $var1het    if mig==1 & `region'==1, $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if ledu==1 &
`region'==1, $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if medu==1 &
`region'==1, $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if hedu==1 &
`region'==1, $se
}

 reg  AL_son  ALbef_fath  $var0het $var1het $var2het $var3het  , $se
 reg  AL_son  ALbef_fath   $var0het $var1het   if nomig==1 , $se
 reg  AL_son  ALbef_fath   $var0het $var1het    if mig==1 , $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if ledu==1 , $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if medu==1 , $se
 reg  AL_son  ALbef_fath   $var0het $var1het $var2het   if hedu==1 , $se



*************************************************************************
***********************************
// Transmission mechanisms
*************************************************************************
***********************************

cd "$wdirD"
use finalsample, clear
keep if  AL1_pgen !=.

global t1  songeb_* state_*   Miss_Sohn Miss_Vater_before
global t2  fathgeb_* fathschool_* fathbil_*
global t3  sibs_* birtho_* migback_Vater_*
global t4  sonbil_* nace_Sohn_*

*Panel A
reg AL_son  ALbef_fath $t1 , $se


testparm ${t1}


reg AL_son  ALbef_fath $t1 $t2, $se
forvalues t=1/2 {
testparm ${t`t'}
}

reg AL_son  ALbef_fath $t1 $t2 $t3, $se
forvalues t=1/3 {
testparm ${t`t'}
}

reg AL_son  ALbef_fath $t1 $t2 $t3 $t4, $se
forvalues t=1/4 {
testparm ${t`t'}
}
```

```
*Panel B
reg AL_son  ALbef_fath $t1 , $se
testparm ${t1}


reg AL_son  ALbef_fath $t1 $t2, $se

forvalues t=1/2 {
testparm ${t`t'}
}

reg AL_son  ALbef_fath $t1  $t3, $se

foreach t in 1 3 {
testparm ${t`t'}
}

reg AL_son  ALbef_fath $t1 $t4, $se

foreach t in 1 4 {
testparm ${t`t'}
}



*********************************************************************************
***********************************
// Robustness tests
*********************************************************************************
***********************************

*Panel A
cd "$wdirD"
use panel.dta , clear

drop school
rename SchoolMAX school_Sohn
replace AL1_pgen = AL1_pequiv if AL1_pgen==. & AL1_pequiv!=.
keep if  AL1_pgen !=.


foreach ins in    AL1_pgen AL5_pgen  {
 ivreg2 AL_2_Sohn (ALbef_fath =`ins' )  ///
                , $se   first  savefirst
estimates restore _ivreg2_ALbef_fath
qui test `ins'


 ivreg2  AL_2_Sohn $var0 $var1 $var2 $var3 (ALbef_fath =`ins' )  ///
                , $se   first  savefirst

estimates restore _ivreg2_ALbef_fath
qui test `ins'

}

 reg  AL_2_Sohn ALbef_fath  , $se
 reg  AL_2_Sohn ALbef_fath   $var0 $var1 $var2 $var3  , $se
```

```
*Panel B
cd "$wdirD"
use finalsample, clear
keep if  AL1_pgen !=.

foreach ins in    AL1_pgen AL5_pgen  {
 ivreg2  AL_2_Sohn (  Vater_AL_1 =`ins' )  ///
                , $se   first  savefirst
estimates restore _ivreg2_Vater_AL_1
qui test `ins'


 ivreg2  AL_2_Sohn $var0 $var1 $var2 $var3 (Vater_AL_1 =`ins' )  ///
                , $se   first  savefirst
estimates restore _ivreg2_Vater_AL_1
qui test `ins'
}

reg  AL_2_Sohn  Vater_AL_1  , $se
reg   AL_2_Sohn  Vater_AL_1   $var0 $var1 $var2 $var3  , $se



*Panel C
cd "$wdirD"
use sons_10_13.dta, clear

foreach ins in    AL1_pgen AL5_pgen {
 ivreg2 AL_son (ALbef_fath =`ins' )  ///
                , $se   first  savefirst

estimates restore _ivreg2_ALbef_fath
qui test `ins'


 ivreg2 AL_son $var0 $var1 $var2 $var3 (ALbef_fath =`ins' )  ///
                , $se   first  savefirst
estimates restore _ivreg2_ALbef_fath
qui test `ins'

}

reg AL_son  ALbef_fath  , $se

reg  AL_son  ALbef_fath   $var0 $var1 migback_Vater_2 migback_Vater_3 $var3  ,
$se
```