

Understanding Society User Support - Support #1498

Create child id in BHPS and assign them their parents' work status

02/02/2021 09:30 AM - Mario Martinez-Jimenez

| | | | |
|---|----------|--------------------|------------|
| Status: | Resolved | Start date: | 02/02/2021 |
| Priority: | High | % Done: | 100% |
| Assignee: | | | |
| Category: | | | |
| Description Dear Sir or Madam, I have managed to create a mother, father, and children identifier using the EGOALT file in BHPS waves. However, I am struggling in how to assign for each child id their parents work status. So far, I have done this: use "\$dataout\egoaltPanelSmall", clear //Generate the partner id: g long partner_pidp=apidp //Generate the mother, and father id: g long mother_pidp=apidp if relationship_bh>=4 & relationship_bh<=7 & asex==2 g long father_pidp=apidp if relationship_bh>=4 & relationship_bh<=7 & asex==1 g long grandmother_pidp=apidp if relationship_bh==20 & asex==2 g long grandfather_pidp=apidp if relationship_bh==20 & asex==1 save "\$dataout\temp_egoalt", replace use "\$dataout\egoaltPanelSmall", clear // Only keep children of EGO keep if relationship_bh>=9 & relationship_bh<=12 count fre wave // attach age variables of ALTER keep pidp apidp relationship_bh esex asex rename pidp xpidp // xpidp: ego identifier who are parents of these children rename apidp pidp //pidp: this now are actually the id of children merge m:1 pidp using "\$dataout\indallPanelSmallVar", keepusing(pidp age_indall wave) drop if _m==2 drop _m rename pidp kpidp label var kpidp "cross-wave identifier of child" rename age_indall kage label var kage "child's age" rename asex ksex lab var ksex "child's sex" rename xpidp pidp **// Assign parent's work status variable values: Mother's: foreach v in sex_dv mastat_dv jbstat { bysort hidp mother_pidp: ge m_`v' = cond(n==1, `v'[1], `v'[2], .) /// if partnum < . lab val m_`v' `v' } Fathers': foreach v in sex_dv mastat_dv jbstat { bysort hidp father_pidp: ge f_`v' = cond(n==1, `v'[1], `v'[2], .) /// | | | |

```
if partnum < .  
lab val f'v' `v'  
} **
```

It would be something like that?

Thank you so much for your kind of answer – it helps me a lot to solve how to identify children inside a household.

Best wishes,

Mario

History

#1 - 02/02/2021 10:57 AM - Understanding Society User Support Team

- Status changed from New to In Progress

- % Done changed from 0 to 10

- Private changed from Yes to No

Dear Mario,

Many thanks for your enquiry. The Understanding Society team is looking into it and we will get back to you as soon as we can.

We aim to respond to simple queries within 48 hours and more complex issues within 7 working days. While we will aim to keep to this response times due to the current coronavirus (COVID-19) related situation it may take us longer to respond.

Best wishes,

Understanding Society User Support Team

#2 - 02/02/2021 01:57 PM - Mario Martinez-Jimenez

Hi,

Thanks for your quick answer.

I would like to add a better way to create children id:

```
foreach w in $W {                                // For each wave, I keep the dataset I need  
local n=strpos("abcdefghijklmnopqr", "`w")        // Creates a correspondence between letter and number  
di `n'  
use "$main_data/bhps_w`n'/b`w'_egoalt.dta", clear  
rename b`w'_ * *                                // Removes the prefix of the variable  
gen wave = `n'  
lab var wave "wave"  
save "$dataout\egoaltSmall`n'", replace  
}
```

```
foreach w in $W {                                // For each wave, I keep the dataset I need  
local n=strpos("abcdefghijklmnopqr", "`w")        // Creates a correspondence between letter and number  
di `n'  
use "$main_data/bhps_w`n'/b`w'_indall.dta", clear  
rename b`w'_ * *                                // Removes the prefix of the variable  
gen wave = `n'  
rename age age_indall  
lab var wave "wave"  
save "$dataout\indallSmall`n'", replace  
}
```

```
forvalues i=1/18 {  
use "$dataout\egoaltSmall`i'", clear  
keep if relationship_bh>=9 & relationship_bh<=12  
count  
keep pidp apidp relationship_bh esex asex  
rename pidp xpidp // xpidp: ego identifier who are parents of these children  
rename apidp pidp //pidp: this now are actually the id of children  
merge m:1 pidp using "$dataout\indallSmall`i'.dta", keepusing(pidp age_indall wave birthy)  
drop if _m==2  
drop _m  
rename pidp kpidp  
label var kpidp "cross-wave identifier of child"  
rename age_indall kage  
label var kage "child's age"  
rename asex ksex  
lab var ksex "child's sex"
```

```

rename xpidp pidp
rename relationship_bh krelationship_bh
label var krelationship_bh "Relationship of ego with the children"
rename birthy kbirthy
lab var kbirthy "child's year of birth"
bys pidp (kage): g k_id=_n
save "$dataout\childrenSmall`i'", replace
}

use "$dataout\childrenSmall1", clear // I append all these small datasets
forvalues n=2/$nW {
append using "$dataout\childrenSmall`n'"
}

// Compress the data to save space
describe
sort pidp wave
compress

// recode values from -1 to -9 to Stata system missing for all variables
quietly mvdecode _all, mv(-9/-1)

• Year
gen year= 1991 if wave==1
replace year= 1992 if wave==2
replace year= 1993 if wave==3
replace year= 1994 if wave==4
replace year= 1995 if wave==5
replace year= 1996 if wave==6
replace year= 1997 if wave==7
replace year= 1998 if wave==8
replace year= 1999 if wave==9
replace year= 2000 if wave==10
replace year= 2001 if wave==11
replace year= 2002 if wave==12
replace year= 2003 if wave==13
replace year= 2004 if wave==14
replace year= 2005 if wave==15
replace year= 2006 if wave==16
replace year= 2007 if wave==17
replace year= 2008 if wave==18

fre year

// Merge All EGOALT:
merge m:m pidp using "$dataout\egoaltPanelSmallVar"
drop _m

// Drop children born after or before

• drop if kbirthy<1976 | kbirthy>1991

//Save data for use:
save "$dataout\childrenPanelSmall", replace

//Drop unuseful datasets:

foreach n in $N {
erase "$dataout\egoaltSmall`n'.dta"
}
foreach n in $N {
erase "$dataout\indallSmall`n'.dta"
}

So then, I don't know how to assign each child their parents work status in the household - as I did for the age and sex of the children. I can imagine that I need the info from INDRESP and use Example 7 of the Introduction to understanding Society workshop that you have done but for the spouse of the ego, but still I don't know how to use this command for this case:

// Create within-household unique partnership identifier.
gen partnum=cond(pno < ppno, pno, ppno) if ppno>0
lab var partnum "Partnership identifier, within household"

// Drop cases who are single or partner/spouse is not present in HH.
*drop if ppno<=0

```

```
// Count the number of respondents for each partnership ID
// Those with non-responding spouses will have numinpart=1.
bysort hidp partnum: egen numinpart = sum(ppno > 0)
label variable numinpart "# respondents from partnership (max 2)"
tab numinpart

sort hidp partnum pno
list hidp partnum pno ppno numinpart in 1/50, sepby(hidp)

// Assign spouse/partner variable values
bysort hidp partnum: ge sp_age=cond(_n==2,age1,age2,.) ///
if partnum<.
lab val sp_age age

foreach v in sex mastat jbstat {
bysort hidp partnum: ge sp_`v' = cond(_n==2, `v'[1], `v'[2], .) ///
if partnum < .
lab val sp_`v' `v'
}
sort hidp partnum
list hidp partnum pno ppno numinpart sp_age sp_sex sp_mastat sp_jbstat in 1/50, sepby(hidp)

label variable sp_age "age of spouse/partner's"
label variable sp_sex "sex of spouse/partner's"
label variable sp_jbstat "current economic activity of spouse/partner"
label variable sp_mastat "de fact marital status of spouse/partner"
```

Many thanks,

Mario

#3 - 02/03/2021 08:06 AM - Understanding Society User Support Team

- File *parents_children_matching_public.do* added
 - Status changed from *In Progress* to *Feedback*
 - % Done changed from 10 to 80

Hello Mario,

If you want to link children with their parents information you can do it for the waves they were living in the same household using the attached syntax. We will be uploading this syntax file to our Syntax webpage soon.

Please let us know if this does not solve your problem or you need further clarifications.

Best wishes,
 Understanding Society User Support Team

#4 - 02/03/2021 01:09 PM - Mario Martinez-Jimenez

Hi Sir or Madam,

Many thanks for your the kind of response - the Do.file has really helpful information.

However, my aim is to do the same but using the BHPS. In the BHPS dataset, the (adopted/ biological/step) mother and (adopted/ biological/step) father id is not available in INDALL file - we just have the father and mother biological identification. However, there is some information about that in the YOUNTH file (that starts in Wave 4 / 1994), but still is just for biological and step-parents and has not included the adopted once. I know that we could use information in EGOALT to create (adopted/ biological/step) mother and (adopted/ biological/step) father id, and then, we may use them to create these mother and father variables? How can I do this using the do.file that is attached for the case of the BHPS dataset -which is slightly tricky?

Moreover, I would like to observe for each child in the household in the BHPS, their parent's employment status. In the do.file attached, we can generate the parents' employment status for each pidp. So, using the EGOALT file, I have created a kpdp for children in the BHPS, and then, I would like to use this kpdp to create their parent's employment status rather than for all the pidp (is that make sense?) - so, how I can do this using the do.file that has been attached to do that? od using INDALL pidp, I would already included the kpids?

Thank you so much for your time and help.

Best wishes,

Mario

#5 - 02/03/2021 01:20 PM - Understanding Society User Support Team

Yes,

Step 1: create the step/adopted/natural parent pidp using the egoalt files. This was covered in the workshop you attended (it was Example 8). You can also register for our MoodleX course and download the material: <https://www.understandingsociety.ac.uk/help/training/online/introduction-course>

Step 2: Then you can use the syntax I provided to match information of step/adopted/natural parents with their children.

There is a section in the syntax file which shows you how you can add additional information about parents. In the syntax file only age_dv and employ from indall are included. You can other variables from other individual files like indresp. In your case you want to include jbstat from indrep.

#6 - 02/08/2021 04:48 PM - Mario Martinez-Jimenez

- File *merge_childrenBHPS.do* added

- File *parents_children_matching.do* added

Dear Sir or Madam,

Thank you so much for your kind of response - I have been working with the codes and your advice these days.

Following Step 1, I have created step/adopted/natural parent pidp (npid_bh) as well as children id (kpidp). See do.file called "merge_childreBHPS".

Second, following your Step 2, I used the syntax to match the information of step/adopted/natural parents with their children, using indresp variables for my case.

Then, I have merged with the children id (kpidp) "egoalt" information following the syntax from Example 8 of the workshop. Doing that, I have also information of the childre's sex, age and year of birth. Please see do.file titled "parents_children_matching".

For my analysis, I just need information of the children born between 1978 and 1993, and their parents variables (work status/pay work/education). Keeping these individuals (using the kbirthy variable), I am observing few observation of the parents jbhas/jbstatus/qfedhi variables and quite non-representative since I have more parents unemployed rather than employed. Without dropping observation with regard the year of birth of the child (kbirthy), I do not have practically missing observations for the jbhas variable. Therefore, I do believe that something wrong I have done with my syntaxis when I am merging the children information.

Perhaps, you have another codes to create children variables and merged them with their parents status. Any help on this would be much appreciate since I have being trying different ways to solve this issue.

Looking forward to hearing from you - I would be able to make a quick call if it is possible.

I really appreciate all your help - which is really needed during this situation.

kind regards,

Mario

#7 - 02/11/2021 11:30 AM - Understanding Society User Support Team

- Assignee changed from Understanding Society User Support Team to Alita Nandi

#8 - 02/11/2021 11:33 AM - Understanding Society User Support Team

Hi Mario,

We have provided you with the syntax for merging parents with children. Please modify that to create the data you need - children's information to their parents' data.

Best wishes,
Understanding Society User Support Team

#9 - 02/11/2021 11:38 AM - Mario Martinez-Jimenez

Hi Alita,

Thanks for your response.

Yes, you have provided me with this syntax but there is not too much description on what is going on in each command. For instance, would you mind to explain to me what is needed the second merge (i.e. merge 1:1 pidp using "\$data/ukhls_w'i'/w'_indall", nogen)?

Thanks for your help.

Best,

Mario

#10 - 02/11/2021 12:27 PM - Understanding Society User Support Team

- File *parents_children_matching_public_v2.do* added

ok. I have added more explanation in the attached syntax file, and also slightly changed the syntax so that only those respondents living with at least one parent are in the final file. Let me know if this is easier to understand and work with.

#11 - 08/05/2021 02:01 PM - Understanding Society User Support Team

- Assignee deleted (Alita Nandi)

#12 - 10/12/2021 02:47 PM - Understanding Society User Support Team

- Status changed from Feedback to Resolved

- % Done changed from 80 to 100

Files

| | | | |
|--|---------|------------|---|
| merge_egoaltBHPS.do | 8.53 KB | 02/02/2021 | Mario Martinez-Jimenez |
| parents_children_matching_public.do | 2.08 KB | 02/03/2021 | Understanding Society User Support Team |
| merge_childrenBHPS.do | 6.95 KB | 02/08/2021 | Mario Martinez-Jimenez |
| parents_children_matching.do | 3.33 KB | 02/08/2021 | Mario Martinez-Jimenez |
| parents_children_matching_public_v2.do | 2.83 KB | 02/11/2021 | Understanding Society User Support Team |