

**Data Manual for the Construction of Variables from the German Family Panel:
Child Residence, Father-Child Contact and Past Union Histories**

Katja Köppen, Michaela Kreyenfeld und Heike Trappe

Summary

This manual is a background paper for the research paper “Loose ties: Determinants of father child-child contact in Germany” written by Katja Köppen, Michaela Kreyenfeld and Heike Trappe published in the *Journal of Marriage and Family* 2018. It explains how we cleaned the partnership and fertility histories. We furthermore show how we extracted information from each panel wave (on e.g. father-child contact, child residence) and how we merged that to the partnership and fertility histories. In order to elucidate each step in the programming process, we use the person with the ID 816873000. The STATA code that was used to construct the variables can be found in the Appendix to this document.

1. Cleaning of partnership and fertility histories

1.1. Partnership histories

The starting point of the analysis is the partnership history “biopart” (see Table 1). Based on the starting and ending dates of a relationship, a cohabitation and a marriage, we transferred the data into an episode data set as displayed in Table 2. In the original data set, time is measured as **century months**. We transferred this information into **months since birth of the respondent**. In order to be able to include a date of censoring, we checked in the panel data set (anchor1, anchor2 etc.) when the person was surveyed for the last time and inserted this date as the censoring date (t=521 in Table 2). After we cleaned the histories, we generated a variable that indicates the current family status (FAM). In order to construct this variable, we needed to prioritize states. We prioritized marriage over cohabitation and cohabitation over living apart together relationships. In some cases, respondents were still married when they started a new partnership. In order to be able to identify the end and beginning of a new partnership, we constructed a variable that indicates the order of the current relationship (UNIONORDER).

A couple of assumptions were made when we cleaned the histories:

- We disregarded breaks in the partnership.
- In case of an overlap of partnerships, we prioritized the latter over the first partnership.
- If no dates were reported in the partnership history, we assumed that the person was single until censoring.
- In case the starting or ending dates of an episode were missing, we flagged the entire case and removed it from the final analysis.

Table 1: Partnership history in biopart.dta

Relbeg	Relend	Cohbeg	Cohend	Marbeg	Marend	DOB
1160	1357	1167	1364	1225	-99 Ongoing	872
1356	-99 Ongoing	-3 Does not apply	-3 Does not apply	-3 Does not apply	-3 Does not apply	872

Table 2: Cleaned partnership history

_t0	_t	RELATION	COHAB	MARRIAGE	UNIONORDER	FAM
0	288	No	No	No	0	Single
288	295	Yes	No	No	1	LAT
295	353	Yes	Yes	No	1	Cohab
353	484	Yes	Yes	Yes	1	Married
484	492	Yes	Yes	Yes	2	Married
492	521	Yes	No	Yes	2	Married

2. Cleaning of fertility histories

From the fertility history (biochild.dta), we know that the respondent had a first child at “1186” in century months (not shown in table). This information was transferred to months since birth of the respondent and merged on to the cleaned partnership history. Also key control variables are constructed in this step. These include the family status at first birth (FAM01) and a time-varying covariate that indicates whether the person is a separated parent (SEP01). The data set is shown in Table 3. A difficulty in the fertility history (biochild.dta) is that all children, including foster, step- and adopted children are recorded. If a person has a foster child, an adopted or step-child that is older than his/her biological child, this child is assigned the order “1”. As we are interested in the relationship of the father to his first born children, we had to rearrange the fertility histories for fathers with adopted and foster children.

Table 3: Fertility and partnership history

_t0	_t	FAM	FAM01	KID1	SEP01
0	288	Single	Cohabiting	Childless	-2
288	295	LAT	Cohabiting	Childless	-2
295	314	Cohabiting	Cohabiting	Childless	-2
314	338	Cohabiting	Cohabiting	Age 0-1	0
338	353	Cohabiting	Cohabiting	Age 2-3	0
353	358	Married	Cohabiting	Age 2-3	0
358	362	Married	Cohabiting	Age 2-3	0
362	382	Married	Cohabiting	Age 4-5	0
382	386	Married	Cohabiting	Age 4-5	0
386	406	Married	Cohabiting	Age 6-7	0
406	410	Married	Cohabiting	Age 6-7	0
410	430	Married	Cohabiting	Age 8-9	0
430	434	Married	Cohabiting	Age 8-9	0
434	454	Married	Cohabiting	Age 10+	0
454	478	Married	Cohabiting	Age 10+	0
478	484	Married	Cohabiting	Age 10+	0
484	485	Married	Cohabiting	Age 10+	1
485	521	Married	Cohabiting	Age 10+	1

3. The panel information: Contact and more

We have appended all eight panel waves in the next step. The data is displayed in Table 4. We used this data to construct a variable that indicates the contact frequency between the separated father and the child (CONTACT). Note that this variable was not surveyed in wave 1 yet. Further variables are the custody arrangement (SORGE). Note here that we have imputed missing information, assuming that the custody arrangement is relatively stable across time. There is information on the date that the interview in the respective year was conducted (Intdat). We used this information to construct a variable that indicates the age of the respondent in months since birth (_t). This is also the time dimension of the combined fertility and partnership histories that we constructed earlier (see Table 3). We merged the latter data and the panel data based on the id and the date of interview (_t). The final data set is displayed in Table 5. The final data set includes all variables that are available in the panel as well as in the fertility and partnership histories, such as the time since the separation from the partner of the first born child (DUR_SEP01).

Table 4: Panel data

wave	CONTACT	SORGE	SORGE_IMP	Intdat	_t
1	.	.	.	1322	450
3	.	.	.	1333	461
4	.	.	.	1344	472
5	.	.	.	1355	483
6	2	.	.	1369	497
7	2	1	1	1380	508
8	.	.	1	1393	521

Table 5: Panel data

Wave	CONTACT	SORGE	SORGE_IMP	Intdat	_t	DUR_SEP01
1	.	.	.	1322	450	.
3	.	.	.	1333	461	.
4	.	.	.	1344	472	.
5	.	.	.	1355	483	.
6	2	.	.	1369	497	12
7	2	1	1	1380	508	23
8	.	.	1	1393	521	36

4. Sample Size

Original sample size	13891
After reduction to...	
wave 2-8 (wave 1, contact not surveyed)	10827
...male respondents	5210
... valid partnership/fertility histories	5170
...	
... first child is under 18	1805
...separated father	379
... does not live with child or does not have sole custody	313
... valid information on contact	285

5. Appendix: STATA-Code

```
*DATA: Merge all panel files
use "$SOURCE\anchor1", clear
append using "$SOURCE\anchor1_DD" "$SOURCE\anchor2" "$SOURCE\anchor3" "$SOURCE\anchor4"
"$SOURCE\anchor5" "$SOURCE\anchor6" "$SOURCE\anchor7" "$SOURCE\anchor8", nolabel
sort id wave
generate intdat=(inty-1900)*12+intm
bys id: egen INTDAT=max(intdat)
g DOB= ((doby_gen-1900)*12+dobm_gen)-1
save "$SAVE\PANEL", replace

*DATA: Partnership Biographies (time varying)
use "$SAVE\PANEL", clear
bysort id: g SPELL=_n
keep if SPELL==1
keep id INTDAT DOB
merge 1:m id using "$SOURCE\biopart"

*VAR: Missings/inconsistencies in partnership bio
g TEST02=0
replace TEST02=1 if relbeg==-7 | relend==-7 | cohbeg==-7 | cohend==-7 | marbeg==-7
| marend==-7
bys id: egen EVER_MISSING=min(TEST02)
lab var EVER_MISSING "something missing in partnership history"

*Episode Data
g BEGIN=relbeg-DOB
g END=relend-DOB
g BEGIN01=cohbeg-DOB
g END01=cohend-DOB
g BEGIN02=marbeg-DOB
g END02=marend-DOB
g TIME=INTDAT-DOB
replace END=9999 if relend==-99 | relend==.
replace END01=9999 if cohend==-3 | cohend==-99 | cohend==.
replace END02=9999 if marend==-3 | marend==-99 | marend==.
replace BEGIN01=9999 if cohbeg==-3 | relend==. | cohend==. | marend==.
replace BEGIN02=9999 if marbeg==-3 | relend==. | cohend==. | marend==.
keep id BEGIN* END* TIME EVER_MISSING
sort id BEGIN
bysort id: ge SPELL=_n
reshape wide BEGIN* END* , i(id) j(SPELL)
replace END1=BEGIN2 if END1>BEGIN2 & END1!=. & BEGIN2!=.
replace END2=BEGIN3 if END2>BEGIN3 & END2!=. & BEGIN3!=.
replace END3=BEGIN4 if END3>BEGIN4 & END3!=. & BEGIN4!=.
replace END4=BEGIN5 if END4>BEGIN5 & END4!=. & BEGIN5!=.
replace END5=BEGIN6 if END5>BEGIN6 & END5!=. & BEGIN6!=.
replace END6=BEGIN7 if END6>BEGIN7 & END6!=. & BEGIN7!=.
replace END7=BEGIN8 if END7>BEGIN8 & END7!=. & BEGIN8!=.
replace END8=BEGIN9 if END8>BEGIN9 & END8!=. & BEGIN9!=.
replace END9=BEGIN10 if END9>BEGIN10 & END9!=. & BEGIN10!=.
replace END10=BEGIN11 if END10>BEGIN11 & END10!=. & BEGIN11!=.
replace END11=BEGIN12 if END11>BEGIN12 & END11!=. & BEGIN12!=.
replace END12=BEGIN13 if END12>BEGIN13 & END12!=. & BEGIN13!=.
replace BEGIN1=0 if BEGIN1<0
replace BEGIN2=0 if BEGIN2<0
replace END1=BEGIN2 if (END1<0 & BEGIN2==0) | (END1<0 & BEGIN2>0)
replace END2=BEGIN3 if END2<0 & BEGIN3>0

*VAR: Relationship
g EVER=0
stset TIME, fail(EVER) id(id)
local i=0
while `i'<10 {
local INDEX =`i'+1
stsplitt REL_B`INDEX' if BEGIN`INDEX'!=., after at(0) (time=BEGIN`INDEX')
stsplitt REL_E`INDEX' if END`INDEX'!=., after at(0) (time=END`INDEX')
local i=`i'+1
}

generate RELATION=.
replace RELATION=0 if REL_B1==-1 & REL_E1==-1
```

```

replace RELATION=0 if REL_B1==0 & REL_E1==0
local i=0
while `i'<10 {
local INDEX =`i'+1
replace RELATION=1 if REL_B`INDEX'==0 & REL_E`INDEX'==-1
replace RELATION=. if BEGIN`INDEX'<0 | END`INDEX'<0
replace RELATION=0 if BEGIN`INDEX'==0 & END`INDEX'==0
replace RELATION=0 if BEGIN`INDEX'==0 & END`INDEX'>0
local i=`i'+1
}

*VAR: Order of union
generate TEST=id if RELATION==1
sort TEST _t0
by TEST: g UNIONORDER= n
replace UNIONORDER=. if TEST==.
lab var UNIONORDER "Order of partnership"

local i=0
while `i'<10 {
local INDEX =`i'+1
stsplint COH_B`INDEX' if BEGIN`INDEX'!=., after at(0) (time=BEGIN01`INDEX')
stsplint COH_E`INDEX' if END`INDEX'!=., after at(0) (time=END01`INDEX')
stsplint MAR_B`INDEX' if BEGIN`INDEX'!=., after at(0) (time=BEGIN02`INDEX')
stsplint MAR_E`INDEX' if END`INDEX'!=., after at(0) (time=END02`INDEX')
local i=`i'+1
}

*VAR: Marriage
generate MARRIAGE=.
replace MARRIAGE=0 if MAR_B1==-1 & MAR_E1==-1
replace MARRIAGE=0 if MAR_B1==0 & MAR_E1==0
local i=0
while `i'<10 {
local INDEX =`i'+1
replace MARRIAGE=1 if MAR_B`INDEX'==0 & MAR_E`INDEX'==-1
replace MARRIAGE=. if BEGIN01`INDEX'<0 | END01`INDEX'<0
local i=`i'+1
}

*VAR: Cohabitation
generate COHAB=.
replace COHAB=0 if COH_B1==-1 & COH_E1==-1
replace COHAB=0 if COH_B1==0 & COH_E1==0
local i=0
while `i'<10 {
local INDEX =`i'+1
replace COHAB=1 if COH_B`INDEX'==0 & COH_E`INDEX'==-1
replace COHAB=. if BEGIN02`INDEX'<0 | END02`INDEX'<0
local i=`i'+1
}

*VAR: Family status (time-varying)
generate FAM=4
replace FAM=3 if RELATION==1
replace FAM=2 if COHAB==1
replace FAM=1 if MARRIAGE==1
bys id: egen TEST01=max(FAM)
replace FAM=-1 if TEST01==. | RELATION==.
lab var FAM "Family status: Time varying"
lab def FAM 1 "Married" 2 "Cohabiting" 3 "LAT" 4 "Single"
lab val FAM FAM
sort id _t0
keep id FAM UNIONORDER EVER_MISSING _ * RELATION COHAB MARRIAGE
save "$SAVE\EVENT_UNION", replace

*DATA: Fertility histories(time-varying)
use "$SOURCE\biochild.dta", clear
keep if statusk==1
bys id number: g SPELL=_n
keep if SPELL==1
generate DUR_KID=dobk-dob
replace DUR_KID=-1 if dobk<0
bys id: g SPELL01=_n
keep id DUR_KID SPELL01
reshape wide DUR_KID, i(id) j(SPELL01)
keep id DUR_KID*

```

```

merge 1:m id using "$SAVE\EVENT_UNION"
mvencode _all, mv(9998)
local i=0
while `i'<10 {
local INDEX =`i'+1
stsplitt KID`INDEX', after at(0 24 48 72 96 120) (time=DUR_KID`INDEX')
local i=`i'+1
}
local i=0
while `i'<10 {
local INDEX =`i'+1
recode KID`INDEX' -1=0 0=1 24=2 48=3 72=4 96=5 120=6
replace KID`INDEX'=-9 if DUR_KID`INDEX'<0 | DUR_KID`INDEX'==.
lab var KID`INDEX' "Kid `INDEX'"
lab def KID`INDEX' 0 "childless" 1 "Age 0-1" 2 "Age 2-3" 3 "Age 4-5" 4 "Age 6-7" 5
"Age 8-9" 6 "Age 10+"
lab val KID`INDEX' KID`INDEX'
local i=`i'+1
}

*VAR: Family status at first birth
bys id: generate TEST=FAM if KID1==1 & KID1[_n-1]==0 & KID1[_n-1]!=.
bys id: egen FAM01=min(TEST)
lab var FAM01 "Family status at first birth"
lab val FAM01 FAM

*VAR: Separated parent
g SEP01=0
bys id: replace SEP01=1 if KID1!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP01=1 if KID1!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP01=1 if SEP01[_n-1]==1
local i=`i'+1
}
lab var SEP01 "Separated from first child (over partnership histories)"

*VAR: Time since separation of parent of first child
stsplitt TESTX, every(1)
sort id _t0
by id: g TEST10=_t0 if SEP01==1 & SEP01[_n-1]==0
bys id: egen TEST11=max(TEST10)
g DUR_SEP01=_t0-TEST11
lab var DUR_SEP01 "Duration since separation of parent of first kid (over partnership
histories)"

generate TRENN_C=99
replace TRENN_C=1 if DUR_SEP01>=0
replace TRENN_C=2 if DUR_SEP01>=36
replace TRENN_C=3 if DUR_SEP01>=72
replace TRENN_C=4 if DUR_SEP01>=108
replace TRENN_C=5 if DUR_SEP01>=144
replace TRENN_C=. if DUR_SEP01==.
lab var TRENN_C "Time since separation (categorical)"
lab def TRENN_C 1 "0-2 years" 2 "3-5 years" 3 "6-8 years" 4 "9-11years" 5 "12+ years"
99 "before"
lab val TRENN_C TRENN_C

*VAR: Family status at separation
by id: g TEST05=FAM if SEP01==1 & SEP01[_n-1]==0
bys id: egen FAM03=max(TEST05)
lab var FAM03 "Family status at separation"
lab val FAM03 FAM

*VAR: Age of child (in months) at separation
g TESTK01=int((_t0-DUR_KID1)/12) if DUR_SEP01==0
bys id: egen AGEK_SEP=max(TESTK01)
lab var AGEK_SEP "Age of child at separation"

*VAR: Age (in months) at separation of parent of first kid
g TEST06=_t0 if DUR_SEP01==0
bys id: egen TEST01=max(TEST06)
g AGE_SEP01=int((TEST01)/12)
lab var AGE_SEP01 "Age at separation"

*VAR: Further separated kids

```



```

g SEP02=0
bys id: replace SEP02=1 if KID2!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP02=1 if KID2!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP02=1 if SEP02[_n-1]==1
local i=`i'+1
}
lab var SEP02 "Separated from 2nd child (over partnership histories)"

g SEP03=0
bys id: replace SEP03=1 if KID3!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP03=1 if KID3!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP03=1 if SEP03[_n-1]==1
local i=`i'+1
}
lab var SEP03 "Separated from 3rd child (over partnership histories)"

g SEP04=0
bys id: replace SEP04=1 if KID4!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP04=1 if KID4!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP04=1 if SEP04[_n-1]==1
local i=`i'+1
}
lab var SEP04 "Separated from 4th child (over partnership histories)"

g SEP05=0
bys id: replace SEP05=1 if KID5!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP05=1 if KID5!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP05=1 if SEP05[_n-1]==1
local i=`i'+1
}
lab var SEP05 "Separated from 5th child (over partnership histories)"

g SEP06=0
bys id: replace SEP06=1 if KID6!=0 & UNIONORDER!=UNIONORDER[_n-1]
replace SEP06=1 if KID6!=0 & FAM==4
local i=0
while `i'<10 {
local INDEX =`i'+1
by id: replace SEP06=1 if SEP06[_n-1]==1
local i=`i'+1
}
lab var SEP06 "Separated from 6th child (over partnership histories)"

g SEP_FURTHER=0
replace SEP_FURTHER=1 if SEP02==1 | SEP03==1 | SEP04==1 | SEP05==1 | SEP06==1
lab var SEP_FURTHER "Further separated children (over partnership histories)"

g NEW_KID=0
replace NEW_KID=1 if KID2>0 & SEP02==0
replace NEW_KID=1 if KID3>0 & SEP03==0
replace NEW_KID=1 if KID4>0 & SEP04==0
replace NEW_KID=1 if KID5>0 & SEP05==0
replace NEW_KID=1 if KID6>0 & SEP06==0
lab var NEW_KID "Kid from ongoing union"

stjoin
sort id _t
keep id_* FAM* KID* DUR_KID* UNIONORDER SEP01 SEP_FURTHER DUR_SEP01 EVER_MISSING
TRENN_C NEW_KID AGE_SEP01 AGEK_SEP
drop _merge
save "$SAVE\EVENT_ALL", replace

```

```

*DATA: PANEL DATA
use "$SOURCE\biochild.dta", clear
drop if index<0
keep id index currliv sexk statusk pid parentid dobk pno wave number
reshape wide currliv sexk statusk pid parentid dobk pno number, i(id wave) j(index)

merge wave id using "$SAVE\PANEL", sort
save "$SAVE\TEST.dta", replace

use "$SAVE\TEST.dta", clear

*VAR: East (place of residence)
generate EAST=0 if bula>=0 & bula<=10
replace EAST=1 if bula>=11 & bula<=16
lab def EAST 0 "West" 1 "East"
lab val EAST EAST

*VAR: Employment status
gen EMP=4
replace EMP=1 if lfs==9 | lfs==10 | lfs==11 | lfs==12
replace EMP=2 if lfs == 4
replace EMP=3 if isced==0|lfs==1|lfs==2|lfs==3|lfs==5|lfs==6|lfs==7|lfs==8|lfs==13|lfs==7|lfs==. if
capture label drop EMP
label var EMP "Employment status"
label def EMP 1 "employed" 2 "unemployed" 3 "education/other/missing"
label value EMP EMP

*VAR: Education
generate EDU=9
replace EDU=1 if isced==1 | isced==2 | isced==3 | isced==0
replace EDU=2 if isced==4 | isced==5 | isced==6
replace EDU=3 if isced==7 | isced==8
label var EDU "Education"
label def EDU 1 "low" 2 "medium" 3 "high" 9"missing"
label value EDU EDU

*VAR: Age of first child
recode klage -7=. -3=.
gen ALTER_KIND=99
replace ALTER_KIND=1 if klage<6
replace ALTER_KIND=2 if klage>=6 & klage<12
replace ALTER_KIND=3 if klage>=12 & klage<18
label var ALTER_KIND "Age first child"
label def ALTER_KIND 1 "<6years" 2 "6-11 years" 3"12-17 years"
label value ALTER_KIND ALTER_KIND

**VAR: Contact
local i=0
while `i'<10 {
local INDEX =`i'+1
g CONTACT`INDEX'=.
replace CONTACT`INDEX'=1 if crn16k`INDEX'==1 & number`INDEX'==1
replace CONTACT`INDEX'=2 if crn16k`INDEX'==2 & number`INDEX'==1
replace CONTACT`INDEX'=3 if crn16k`INDEX'==3 & number`INDEX'==1
replace CONTACT`INDEX'=4 if crn16k`INDEX'==4 & number`INDEX'==1
replace CONTACT`INDEX'=5 if crn16k`INDEX'==5 & number`INDEX'==1
replace CONTACT`INDEX'=6 if crn16k`INDEX'==6|crn16k`INDEX'==7|crn16k`INDEX'==8|crn16k`INDEX'==9) & number1==1 if
lab def CONTACT`INDEX' 1 "Daily" 2 "several time per week" 3 "once per week" 4 "1-3 per
month" 5 "several times per year" 6 "less/no contact"
lab val CONTACT`INDEX' CONTACT`INDEX'
local i=`i'+1
}

*VAR: Contact with first kid
generate CONTACT_BIO1=.
replace CONTACT_BIO1=CONTACT1 if CONTACT1 !=. & statusk1==1
replace CONTACT_BIO1=CONTACT2 if statusk1==3 | statusk1==4
replace CONTACT_BIO1=CONTACT3 if (statusk1==3 | statusk1==4) & (statusk2==3 |
statusk2==4)
lab def CONTACT_BIO1 1 "Daily" 2 "several time per week" 3 "once per week" 4 "1-3 per
month" 5 "several times per year" 6 "less/no contact"
lab val CONTACT_BIO1 CONTACT_BIO1
lab var CONTACT_BIO1 "Contact with first biological child"

*VAR: Contact bivariat
generate CONTACT=.

```

```

replace CONTACT=1 if CONTACT_BIO1==1 | CONTACT_BIO1==2 | CONTACT_BIO1==3 |
CONTACT_BIO1==4
replace CONTACT=0 if CONTACT_BIO1==5 | CONTACT_BIO1==6
label var CONTACT "Contact frequency based on non-resident fathers information"
label def CONTACT 1 "contact" 0 "no contact"
label value CONTACT CONTACT

*VAR: Custody (Sorgerechtsregelung)
local i=0
while `i'<10 {
local INDEX =`i'+1
g SURGE`INDEX'=.
replace SURGE`INDEX'=1 if crn18k`INDEX'==1
replace SURGE`INDEX'=2 if crn18k`INDEX'==2
replace SURGE`INDEX'=3 if crn18k`INDEX'==3
lab def SURGE`INDEX' 1 "joint" 2 "Anchor" 3 "Partner"
lab val SURGE`INDEX' SURGE`INDEX'
local i=`i'+1
}

*VAR: Custody kid 1 (Sorgerechtsregelung)
generate SURGE_BIO1=.
replace SURGE_BIO1=SURGE1 if statusk1==1
replace SURGE_BIO1=SURGE2 if statusk1==3 | statusk1==4
replace SURGE_BIO1=SURGE3 if (statusk1==3 | statusk1==4) & (statusk2==3 | statusk2==4)
lab def SURGE_BIO1 1 "joint" 2 "anchor" 3 "Partner"
lab val SURGE_BIO1 SURGE_BIO1
lab var SURGE_BIO1 "Custody (Sorgerechtsregelung)"

*VAR: Custody (imputed)
g SURGE_IMP=SURGE_BIO1
sort id wave
local i=0
while `i'<10 {
local INDEX =`i'+1
bys id: replace SURGE_IMP=SURGE_BIO1[_n-`INDEX'] if SURGE_BIO1[_n-`INDEX']!=. &
SURGE_IMP==.
local i=`i'+1
}
recode SURGE_IMP .=9

*VAR: Alimony (Kindesunterhalt)
gen UNTERHALT=999
replace UNTERHALT=1 if inc22i2==1
replace UNTERHALT=0 if inc22i2==0|inc22i2==7|inc22i2<0
label var UNTERHALT "Pays Alimony for children?"
label def UNTERHALT 1 "yes" 0 "no"
label value UNTERHALT UNTERHALT

**VAR: Residence with first kid
generate LIVE_KID1=.
replace LIVE_KID1=currliv1 if statusk1==1
replace LIVE_KID1=currliv2 if (statusk1==3 | statusk1==4) & statusk2==1
replace LIVE_KID1=currliv3 if (statusk1==3 | statusk1==4) & (statusk2==3 | statusk2==4)
& statusk3==1
lab var LIVE_KID1 "Residence kid 1"
lab val LIVE_KID1 currliv_bc
tab LIVE_KID1

*VAR: Familienstatus aktuell
capture drop FAM_NOW
generate FAM_NOW=9
replace FAM_NOW=3 if relstat==4 | relstat==5
replace FAM_NOW=2 if relstat==3 | relstat==8 | relstat==11
replace FAM_NOW=1 if relstat==2 | relstat==7 |relstat==10
replace FAM_NOW=0 if relstat==1 | relstat==6 | relstat==9
capture label drop FAM_NOW
lab def FAM_NOW 0 "single" 1 "LAT" 2 "Coh" 3 "Married" 9"missing"
lab val FAM_NOW FAM_NOW
tab FAM_NOW

*VAR: AGE
g _t=intdat-DOB
g AGE=int(_t/12)

```

```

*MERGE WITH PANEL DATA
sort id wave
keep id wave _t UNTERHALT  SORGE_IMP AGE* EAST EMP ALTER_KIND klage CONTACT* nkidsbio
sex_gen EDU LIVE_KID1 LIVES_KID1 FAM_NOW dlcalweight
keep if wave!=.
sort id _t
save "$SAVE\PANEL01", replace
use "$SAVE\PANEL01", clear
merge id _t using "$SAVE\EVENT_ALL" , sort
save "$SAVE\FINAL", replace

```

```

*MULTIVARIAT ANALYSIS: CONTACT
use "$SAVE\FINAL", clear
keep if wave!=.
drop if wave==1
keep if sex_gen==1
drop if DUR_KID1==--1
drop if DUR_KID2==--1
drop if EVER==1
keep if FAM01>=0
keep if nkidsbio>0
drop if klage>17
keep if SEP01==1
drop if LIVE_KID1==1
drop if SORGE_IMP==2
keep if CONTACT!=.

```