

UKHLS exploration (weighed kin) 4.20

Reading and Cleaning data

```
#housekeeping
#Clear objects already in the environment - start with a clean slate
rm(list=ls())

#loading libraries
library(tidyverse)
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.2      v readr      2.1.4
v forcats    1.0.0      v stringr    1.5.0
v ggplot2    3.4.3      v tibble     3.2.1
v lubridate  1.9.2      v tidyr      1.3.0
v purrr      1.0.2

-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
library(svyVGAM)
```

```
Loading required package: VGAM
Loading required package: stats4
Loading required package: splines
Loading required package: survey
Loading required package: grid
Loading required package: Matrix
```

Attaching package: 'Matrix'

The following objects are masked from 'package:tidyr':

expand, pack, unpack

Loading required package: survival

Attaching package: 'survey'

The following object is masked from 'package:VGAM':

calibrate

The following object is masked from 'package:graphics':

dotchart

```
library(sjlabelled)
```

Warning: package 'sjlabelled' was built under R version 4.3.3

Attaching package: 'sjlabelled'

The following object is masked from 'package:forcats':

as_factor

The following object is masked from 'package:dplyr':

as_label

The following object is masked from 'package:ggplot2':

as_label

```
library(desctable)
```

Warning: package 'desctable' was built under R version 4.3.3

Loading required package: pander

Warning: package 'pander' was built under R version 4.3.3

Attaching package: 'desctable'

The following objects are masked from 'package:stats':

chisq.test, fisher.test, IQR

```
library(summarytools)
```

Warning: package 'summarytools' was built under R version 4.3.3

Attaching package: 'summarytools'

The following object is masked from 'package:sjlabelled':

unlabel

The following object is masked from 'package:tibble':

view

```
library(naniar)
```

Warning: package 'naniar' was built under R version 4.3.3

```
library(survey)
library(svrep)
library(Hmisc)
```

Attaching package: 'Hmisc'

The following objects are masked from 'package:summarytools':

label, label<-

The following object is masked from 'package:survey':

deff

The following objects are masked from 'package:dplyr':

src, summarize

The following objects are masked from 'package:base':

format.pval, units

```
library(srvyr)
```

Attaching package: 'srvyr'

The following object is masked from 'package:Hmisc':

summarize

The following object is masked from 'package:stats':

filter

```
library(marginaleffects)
library(haven)
```

Attaching package: 'haven'

The following objects are masked from 'package:sjlabelled':

as_factor, read_sas, read_spss, read_stata, write_sas, zap_labels

```
library(catregs)
library(margins)
library(modelsummary)
```

Attaching package: 'modelsummary'

The following object is masked from 'package:Hmisc':

Mean

The following object is masked from 'package:VGAM':

Max

```
library(zoo)
```

Attaching package: 'zoo'

The following objects are masked from 'package:base':

as.Date, as.Date.numeric

```
library(mice)
```

Warning: package 'mice' was built under R version 4.3.3

Attaching package: 'mice'

The following object is masked from 'package:stats':

filter

The following objects are masked from 'package:base':

cbind, rbind

```
library(stargazer)
```

Please cite as:

Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
R package version 5.2.3. <https://CRAN.R-project.org/package=stargazer>

```
library(texreg)
```

Warning: package 'texreg' was built under R version 4.3.3

Version: 1.39.3
Date: 2023-11-09
Author: Philip Leifeld (University of Essex)

Consider submitting praise using the `praise` or `praise_interactive` functions.
Please cite the JSS article in your publications -- see `citation("texreg")`.

Attaching package: 'texreg'

The following object is masked from 'package:tidyr':

`extract`

```
library(VIM)
```

Warning: package 'VIM' was built under R version 4.3.3

Loading required package: colorspace
VIM is ready to use.

Suggestions and bug-reports can be submitted at: <https://github.com/statistikat/VIM/issues>

Attaching package: 'VIM'

The following object is masked from 'package:VGAM':

```
wine
```

The following object is masked from 'package:datasets':

```
sleep
```

```
library(lattice)
library(ggplot2)
library(plyr)
```

You have loaded plyr after dplyr - this is likely to cause problems.
If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
library(plyr); library(dplyr)

Attaching package: 'plyr'

The following objects are masked from 'package:srvyr':

```
mutate, rename, summarise, summarize
```

The following objects are masked from 'package:Hmisc':

```
is.discrete, summarize
```

The following objects are masked from 'package:dplyr':

```
arrange, count, desc, failwith, id, mutate, rename, summarise,
summarize
```

The following object is masked from 'package:purrr':

```
compact
```

```
library(dplyr)
```

Wave 1-5 egoalt (8-6) hhresp (8-6)

```
inpath<-"D:/r git projects/ox-R/final essay/UKHLS n BHPS stata/UKDA-6614-stata/stata/stata  
missval <- c(-9, -8, -7, -2, -1)
```

Wave 8~13 egoalt (9-13) hhresp (9-13)

```
#data=====
#wave 8
h_indresp <- read_dta(file=paste0(inpath, "ukhls/h_indresp.dta"))
h_indresp<-h_indresp%>%dplyr::select(pidp, h_hidp, h_mastat_dv,
                                     h_age_dv, h_sex_dv, h_hiqqual_dv,
                                     h_mastat_dv, h_racel_dv,h_fimnnet_dv,h_nchild_dv,h_urban_dv,h_gor_dv,h_jbstat,h_s

#wave 9
i_indresp <- read_dta(file=paste0(inpath, "ukhls/i_indresp.dta"))
i_indresp<-i_indresp%>%dplyr::select(pidp, i_hidp, i_mastat_dv,
                                     i_age_dv, i_sex_dv, i_hiqqual_dv,
                                     i_mastat_dv, i_racel_dv,i_fimnnet_dv,i_nchild_dv,i_urban_dv,i_gor_dv,i_jbstat,i_s

#wave 10
j_indresp <- read_dta(file=paste0(inpath, "ukhls/j_indresp.dta"))
j_indresp<-j_indresp%>%dplyr::select(pidp, j_hidp, j_mastat_dv,
                                     j_age_dv, j_sex_dv, j_hiqqual_dv,
                                     j_mastat_dv, j_racel_dv,j_fimnnet_dv,j_nchild_dv,j_urban_dv,j_gor_dv,j_jbstat,j_s

#wave 11
k_indresp <- read_dta(file=paste0(inpath, "ukhls/k_indresp.dta"))
k_indresp<-k_indresp%>%dplyr::select(pidp, k_hidp, k_mastat_dv,
                                     k_age_dv, k_sex_dv, k_hiqqual_dv,
                                     k_mastat_dv, k_racel_dv,k_fimnnet_dv,k_nchild_dv,k_urban_dv,k_gor_dv,k_jbstat,k_s

#wave 12
l_indresp <- read_dta(file=paste0(inpath, "ukhls/l_indresp.dta"))
l_indresp<-l_indresp%>%dplyr::select(pidp, l_hidp, l_mastat_dv,
                                     l_age_dv, l_sex_dv, l_hiqqual_dv,
                                     l_mastat_dv, l_racel_dv,l_fimnnet_dv,l_nchild_dv,l_urban_dv,l_gor_dv,l_jbstat,l_s

#wave 13
```



```

m_indresp <- read_dta(file=paste0(inpath, "ukhls/m_indresp.dta"))

m_indresp<-m_indresp%>%dplyr::select(pidp, m_hidp, m_mastat_dv,
  m_age_dv, m_sex_dv, m_hiqual_dv,
  m_mastat_dv, m_racel_dv,m_fimmnet_dv,m_nchild_dv,m_urban_dv,m_gor_dv,m_jbstat,m_s

#egoalt=====
h_egoalt<-read_dta(file=paste0(inpath, "ukhls/h_egoalt.dta"))

i_egoalt<-read_dta(file=paste0(inpath, "ukhls/i_egoalt.dta"))

j_egoalt<-read_dta(file=paste0(inpath, "ukhls/j_egoalt.dta"))

k_egoalt<-read_dta(file=paste0(inpath, "ukhls/k_egoalt.dta"))

l_egoalt<-read_dta(file=paste0(inpath, "ukhls/l_egoalt.dta"))

m_egoalt<-read_dta(file=paste0(inpath, "ukhls/m_egoalt.dta"))

#hhresp=====
h_hhresp<-read_dta(file=paste0(inpath, "ukhls/h_hhresp.dta")) %>%
  dplyr::select(h_hidp,h_tenure_dv)

i_hhresp<-read_dta(file=paste0(inpath, "ukhls/i_hhresp.dta")) %>%
  dplyr::select(i_hidp,i_tenure_dv)

j_hhresp<-read_dta(file=paste0(inpath, "ukhls/j_hhresp.dta")) %>%
  dplyr::select(j_hidp,j_tenure_dv)

k_hhresp<-read_dta(file=paste0(inpath, "ukhls/k_hhresp.dta")) %>%
  dplyr::select(k_hidp,k_tenure_dv)

l_hhresp<-read_dta(file=paste0(inpath, "ukhls/l_hhresp.dta")) %>%
  dplyr::select(l_hidp,l_tenure_dv)

m_hhresp<-read_dta(file=paste0(inpath, "ukhls/m_hhresp.dta")) %>%
  dplyr::select(m_hidp,m_tenure_dv)

```

```

##remove all missing values in indresp
for (i in 1:5) {
  h_indresp<- h_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))
}
for (i in 1:5) {
  i_indresp<- i_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))
}
for (i in 1:5) {
  j_indresp<- j_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))
}
for (i in 1:5) {
  k_indresp<- k_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))
}
for (i in 1:5) {
  l_indresp<- l_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))}

for (i in 1:5) {
  m_indresp<- m_indresp %>%
    mutate_all(., list(~na_if(., missval[i])))}

```

Getting the youths not living with parents in the next wave

Wave 1~5

Wave 8~12

Wave 8

```

mothers <- h_egoalt %>%
  filter(h_relationship_dv %in% c(4) & h_asex==2) %>%
  select(pidp, apidp)
mothers$mother_pidp=mothers$apidp

h_egoalt <- h_egoalt %>%

```

```

    full_join(mothers, by=c("pidp", "apidp"))

rm(mothers)

fathers <- h_egoalt %>%
  filter(h_relationship_dv %in% c(4) & h_asex==1) %>%
  select(pidp, apidp)
fathers$father_pidp=fathers$apidp

h_egoalt <- h_egoalt %>%
  full_join(fathers, by=c("pidp", "apidp"))

rm(fathers)

other_relatives <- h_egoalt %>%
  filter(!(h_relationship_dv%in% c(1,2,3,4,9)) ) %>%
  select(pidp, apidp)
other_relatives$other_relatives_pidp=other_relatives$apidp

h_egoalt <- h_egoalt %>%
  full_join(other_relatives, by=c("pidp", "apidp"))

rm(other_relatives)

#at least two relatives in the household structure who are not the focal individual's parent
test<-h_egoalt%>%aggregate(other_relatives_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","relativenum")
h_egoalt<-join_all(list(h_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent live with a child
child <- h_egoalt %>%
  filter(h_relationship_dv%in% c(9) ) %>%
  select(pidp, apidp)
child$child_pidp=child$apidp
h_egoalt <- h_egoalt %>%
  full_join(child, by=c("pidp", "apidp"))
rm(child)

#count number of respondent children
test<-h_egoalt%>%aggregate(child_pidp~pidp,FUN=length)

```

```

colnames(test)<-c("pidp","childnum")
h_egoalt<-join_all(list(h_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(marriage)
mar <- h_egoalt %>%
  filter(h_relationship_dv%in% c(1,2) ) %>%
  select(pidp, apidp)
mar$mar_pidp=mar$apidp
h_egoalt <- h_egoalt %>%
  full_join(mar, by=c("pidp", "apidp"))
rm(mar)
#count number of respondent married partner
test<-h_egoalt%>%aggregate(mar_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_husbandwife")
h_egoalt<-join_all(list(h_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(cohabitation)
cohab <- h_egoalt %>%
  filter(h_relationship_dv%in% c(3) ) %>%
  select(pidp, apidp)
cohab$cohab_pidp=cohab$apidp
h_egoalt <- h_egoalt %>%
  full_join(cohab, by=c("pidp", "apidp"))
rm(cohab)
#count number of cohabitee
test<-h_egoalt%>%aggregate(cohab_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_cohabitee")
h_egoalt<-join_all(list(h_egoalt,test),by="pidp",type="left")
rm(test)

# leave: not live with mothers
h_egoalt$separated_m <- labelled(
  ifelse(!is.na(h_egoalt$mother_pidp) & h_egoalt$h_enwstat==5), -8,
  ifelse(!is.na(h_egoalt$mother_pidp) & h_egoalt$h_anwstat==5), 3,
  ifelse(!is.na(h_egoalt$mother_pidp) &
    h_egoalt$h_anwstat==2 & h_egoalt$h_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(h_egoalt$mother_pidp) &

```

```

      h_egoalt$h_anwstat %in% c(3,4) & h_egoalt$h_enwstat==2), 2,
ifelse(!(is.na(h_egoalt$mother_pidp) &
      h_egoalt$h_anwstat==2 & h_egoalt$h_enwstat==2), 1,
ifelse(!(is.na(h_egoalt$mother_pidp) & h_egoalt$h_anwstat==1), 0,
      ifelse(!(is.na(h_egoalt$mother_pidp)), -9, NA)))))),

label = "Whether living with biological mother at the next wave",
labels = c("missing" = -9,
      "ego not alive at next wave" = -8,
      "living together at next wave" = 0, "separated_m at next wave" = 1,
      "possibly separated_m at next wave" = 2,
      "mother not alive at next wave" = 3))

#leave: not live with fathers
h_egoalt$separated_f <- labelled(
  ifelse(!(is.na(h_egoalt$father_pidp) & h_egoalt$h_enwstat==5), -8,
  ifelse(!(is.na(h_egoalt$father_pidp) & h_egoalt$h_anwstat==5), 3,
  ifelse(!(is.na(h_egoalt$father_pidp) &
      h_egoalt$h_anwstat==2 & h_egoalt$h_enwstat %in% c(3,4)), 2,
  ifelse(!(is.na(h_egoalt$father_pidp) &
      h_egoalt$h_anwstat %in% c(3,4) & h_egoalt$h_enwstat==2), 2,
  ifelse(!(is.na(h_egoalt$father_pidp) &
      h_egoalt$h_anwstat==2 & h_egoalt$h_enwstat==2), 1,
  ifelse(!(is.na(h_egoalt$father_pidp) & h_egoalt$h_anwstat==1), 0,
      ifelse(!(is.na(h_egoalt$father_pidp)), -9, NA))))))),

label = "Whether living with biological father at the next wave",
labels = c("missing" = -9,
      "ego not alive at next wave" = -8,
      "living together at next wave" = 0, "separated at next wave" = 1,
      "possibly separated at next wave" = 2,
      "father not alive at next wave" = 3))

#fill the separate indicator
h_egoalt<-h_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_m,.direction="updown")
h_egoalt<-h_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_f,.direction="updown")

#find out who separate from both parents
#not live with both parents

```

```

h_egoalt$separated<- labelled(
  ifelse((h_egoalt$separated_m%in%c(1)&h_egoalt$separated_f%in%c(1)),1,

  ifelse((h_egoalt$separated_m%in%c(1)& is.na(h_egoalt$separated_f)),1,#one
  ifelse((h_egoalt$separated_f%in%c(1)& is.na(h_egoalt$separated_m)),1,

  ifelse((h_egoalt$separated_f%in%c(1)& h_egoalt$separated_m%in%c(3)),1,#death
  ifelse((h_egoalt$separated_m%in%c(1)& h_egoalt$separated_f%in%c(3)),1,

  #possibly separated
  ifelse((h_egoalt$separated_m%in%c(2)&h_egoalt$separated_f%in%c(2)),2,
  ifelse((h_egoalt$separated_m%in%c(1)&h_egoalt$separated_f%in%c(2)),2,
  ifelse((h_egoalt$separated_m%in%c(2)&h_egoalt$separated_f%in%c(1)),2,

  ifelse((h_egoalt$separated_m%in%c(2)& is.na(h_egoalt$separated_f)),2,#one
  ifelse((h_egoalt$separated_f%in%c(2)& is.na(h_egoalt$separated_m)),2,

  ifelse((h_egoalt$separated_f%in%c(2)& h_egoalt$separated_m%in%c(3)),1,#death
  ifelse((h_egoalt$separated_m%in%c(2)& h_egoalt$separated_f%in%c(3)),1,

  #living together
  ifelse((h_egoalt$separated_m%in%c(0)|h_egoalt$separated_f%in%c(0)),0,
  #respondents dead
  ifelse((h_egoalt$separated_m%in%c(-8)|h_egoalt$separated_f%in%c(-8)),-8,
    ifelse((!is.na(h_egoalt$separated_m)|!is.na(h_egoalt$separated_f)),-9,NA)))))))))

label = "Whether living with at least one biological parent at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together with at least one parent at next wave" = 0,
  "separated from both parents at next wave" = 1,
  "possibly separated from both parents at next wave" = 2))

#living with both parents indicator
h_egoalt<-h_egoalt %>%group_by(pidp) %>%tidyr::fill(mother_pidp,.direction="updown")
h_egoalt<-h_egoalt %>%group_by(pidp) %>%tidyr::fill(father_pidp,.direction="updown")

#flags for living with mother or father this wave
h_egoalt<-h_egoalt%>%group_by(pidp)%>%mutate(livewithmother=ifelse(!is.na(h_egoalt$mother_

```

```

h_egoalt<-h_egoalt%>%group_by(pidp)%>%mutate(livewithfather=ifelse(!is.na(h_egoalt$father_

#living with both parents this wave
h_egoalt$liveboth<-c(0)
h_egoalt$liveboth[h_egoalt$livewithmother==1 & h_egoalt$livewithfather==1]<-1

#a column for living with mother, father, or both this wave
h_egoalt$lvpa<-c(0)
h_egoalt$lvpa[h_egoalt$livewithmother==1]<-1
h_egoalt$lvpa[h_egoalt$livewithfather==1]<-2
h_egoalt$lvpa[h_egoalt$liveboth==1]<-3
h_egoalt$lvpa[h_egoalt$lvpa==0]<-NA

h_egoalt$lvpa<-factor(h_egoalt$lvpa, levels=c(1,2,3),labels=c("live with mother","live wit

#select only for those who live with at least one parent this wave and at risk of leaving
h_egoalt<-h_egoalt[which(h_egoalt$lvpa%in%c("live with mother","live with father","live wi

#merge datafiles
h_indresp<-h_indresp%>%dplyr::select(pidp, h_hidp, h_mastat_dv,
    h_age_dv, h_sex_dv, h_hiqal_dv,
    h_mastat_dv, h_racel_dv,h_fimnnet_dv,h_nchild_dv,h_urban_dv,h_gor_dv,h_jbstat,h_s

h_egoaltt<-h_egoalt%>%dplyr::select(pidp,mother_pidp,father_pidp,separated_m,separated_f,s

#child levels
h_indrespc<-join_all(list(h_indresp,h_egoaltt), by='pidp', type='left')
h_indrespc<-h_indrespc %>% distinct_at(vars(pidp), .keep_all = TRUE)

h_indrespc<-h_indrespc[(which(!is.na(h_indrespc$mother_pidp))|!is.na(h_indrespc$father_pidp

#add parent levels
h_indresppt <-h_indresp

#join parental data by pidp
father<-as.data.frame(h_indrespc$father_pidp)
mother<-as.data.frame(h_indrespc$mother_pidp)

```

```

colnames(father)[1]<-"pidp"
colnames(mother)[1]<-"pidp"

#attached parental values
h_indresppf<-join_all(list(father,h_indresppt), by='pidp', type='left')
h_indresppm<-join_all(list(mother,h_indresppt), by='pidp', type='left')

#for housing tenure
h_indresppf<-join_all(list(h_indresppf,h_hhresp), by='h_hidp', type='left')
h_indresppm<-join_all(list(h_indresppm,h_hhresp), by='h_hidp', type='left')

#rename var to differentiate parents
h_indresppf <-
  rename_at(h_indresppf, vars(starts_with("h_")), ~str_replace(., "h_", "ip_"))
h_indresppm <-
  rename_at(h_indresppm, vars(starts_with("h_")), ~str_replace(., "h_", "im_"))

#join to the child-level data by parent's pidp
colnames(h_indresppf)[1]<-"father_pidp"
colnames(h_indresppm)[1]<-"mother_pidp"

h_indresppf<-join_all(list(h_indrespc,h_indresppf), by="father_pidp", type='left')
h_indresppm<-join_all(list(h_indrespc,h_indresppm), by="mother_pidp", type='left')

h_indrespp<-join_all(list(h_indresppm,h_indresppf), by="pidp", type='left')

h_indrespp<-h_indrespp %>% distinct_at(vars(pidp), .keep_all = TRUE)

rm(father,mother,h_egoaltt,h_indrespc,h_indresppf,h_indresppm,h_indresppt)

#drop duplicate columns
h_indrespp <- h_indrespp %>% select(-contains(".1"))

## add cross wave identifier
h_indrespp$from_wave_h<-1

```

Wave 9


```

mothers <- i_egoalt %>%
  filter(i_relationship_dv %in% c(4) & i_asex==2) %>%
  select(pidp, apidp)
mothers$mother_pidp=mothers$apidp

i_egoalt <- i_egoalt %>%
  full_join(mothers, by=c("pidp", "apidp"))

rm(mothers)

fathers <- i_egoalt %>%
  filter(i_relationship_dv %in% c(4) & i_asex==1) %>%
  select(pidp, apidp)
fathers$father_pidp=fathers$apidp

i_egoalt <- i_egoalt %>%
  full_join(fathers, by=c("pidp", "apidp"))

rm(fathers)

other_relatives <- i_egoalt %>%
  filter(!(i_relationship_dv%in% c(1,2,3,4,9)) ) %>%
  select(pidp, apidp)
other_relatives$other_relatives_pidp=other_relatives$apidp

i_egoalt <- i_egoalt %>%
  full_join(other_relatives, by=c("pidp", "apidp"))

rm(other_relatives)

#at least two relatives in the household structure who are not the focal individual's parent
test<-i_egoalt%>%aggregate(other_relatives_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","relativenum")
i_egoalt<-join_all(list(i_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent live with a child
child <- i_egoalt %>%
  filter(i_relationship_dv%in% c(9) ) %>%
  select(pidp, apidp)

```

```

child$child_pidp=child$apidp
i_egoalt <- i_egoalt %>%
  full_join(child, by=c("pidp", "apidp"))
rm(child)
#count number of respondent children
test<-i_egoalt%>%aggregate(child_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","childnum")
i_egoalt<-join_all(list(i_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(marriage)
mar <- i_egoalt %>%
  filter(i_relationship_dv%in% c(1,2) ) %>%
  select(pidp, apidp)
mar$mar_pidp=mar$apidp
i_egoalt <- i_egoalt %>%
  full_join(mar, by=c("pidp", "apidp"))
rm(mar)
#count number of respondent married partner
test<-i_egoalt%>%aggregate(mar_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_husbandwife")
i_egoalt<-join_all(list(i_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(cohabitation)
cohab <- i_egoalt %>%
  filter(i_relationship_dv%in% c(3) ) %>%
  select(pidp, apidp)
cohab$cohab_pidp=cohab$apidp
i_egoalt <- i_egoalt %>%
  full_join(cohab, by=c("pidp", "apidp"))
rm(cohab)
#count number of cohabitee
test<-i_egoalt%>%aggregate(cohab_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_cohabitee")
i_egoalt<-join_all(list(i_egoalt,test),by="pidp",type="left")
rm(test)

# leave: not live with mothers

```

```

i_egoalt$separated_m <- labelled(
  ifelse(!is.na(i_egoalt$mother_pidp) & i_egoalt$i_enwstat==5), -8,
  ifelse(!is.na(i_egoalt$mother_pidp) & i_egoalt$i_anwstat==5), 3,
  ifelse(!is.na(i_egoalt$mother_pidp) &
    i_egoalt$i_anwstat==2 & i_egoalt$i_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(i_egoalt$mother_pidp) &
    i_egoalt$i_anwstat %in% c(3,4) & i_egoalt$i_enwstat==2), 2,
  ifelse(!is.na(i_egoalt$mother_pidp) &
    i_egoalt$i_anwstat==2 & i_egoalt$i_enwstat==2), 1,
  ifelse(!is.na(i_egoalt$mother_pidp) & i_egoalt$i_anwstat==1), 0,
  ifelse(!is.na(i_egoalt$mother_pidp)), -9, NA))))),

label = "Whether living with biological mother at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated_m at next wave" = 1,
  "possibly separated_m at next wave" = 2,
  "mother not alive at next wave" = 3))

#leave: not live with fathers
i_egoalt$separated_f <- labelled(
  ifelse(!is.na(i_egoalt$father_pidp) & i_egoalt$i_enwstat==5), -8,
  ifelse(!is.na(i_egoalt$father_pidp) & i_egoalt$i_anwstat==5), 3,
  ifelse(!is.na(i_egoalt$father_pidp) &
    i_egoalt$i_anwstat==2 & i_egoalt$i_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(i_egoalt$father_pidp) &
    i_egoalt$i_anwstat %in% c(3,4) & i_egoalt$i_enwstat==2), 2,
  ifelse(!is.na(i_egoalt$father_pidp) &
    i_egoalt$i_anwstat==2 & i_egoalt$i_enwstat==2), 1,
  ifelse(!is.na(i_egoalt$father_pidp) & i_egoalt$i_anwstat==1), 0,
  ifelse(!is.na(i_egoalt$father_pidp)), -9, NA))))),

label = "Whether living with biological father at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated at next wave" = 1,
  "possibly separated at next wave" = 2,
  "father not alive at next wave" = 3))

#fill the separate indicator

```

```

i_egoalt<-i_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_m,.direction="updown")
i_egoalt<-i_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_f,.direction="updown")

#find out who separate from both parents
#not live with both parents
i_egoalt$separated<- labelled(
  ifelse((i_egoalt$separated_m%in%c(1)&i_egoalt$separated_f%in%c(1)),1,

  ifelse((i_egoalt$separated_m%in%c(1)& is.na(i_egoalt$separated_f)),1,#one
  ifelse((i_egoalt$separated_f%in%c(1)& is.na(i_egoalt$separated_m)),1,

  ifelse((i_egoalt$separated_f%in%c(1)& i_egoalt$separated_m%in%c(3)),1,#death
  ifelse((i_egoalt$separated_m%in%c(1)& i_egoalt$separated_f%in%c(3)),1,

  #possibly separated
  ifelse((i_egoalt$separated_m%in%c(2)&i_egoalt$separated_f%in%c(2)),2,
  ifelse((i_egoalt$separated_m%in%c(1)&i_egoalt$separated_f%in%c(2)),2,
  ifelse((i_egoalt$separated_m%in%c(2)&i_egoalt$separated_f%in%c(1)),2,

  ifelse((i_egoalt$separated_m%in%c(2)& is.na(i_egoalt$separated_f)),2,#one
  ifelse((i_egoalt$separated_f%in%c(2)& is.na(i_egoalt$separated_m)),2,

  ifelse((i_egoalt$separated_f%in%c(2)& i_egoalt$separated_m%in%c(3)),1,#death
  ifelse((i_egoalt$separated_m%in%c(2)& i_egoalt$separated_f%in%c(3)),1,

  #living together
  ifelse((i_egoalt$separated_m%in%c(0)|i_egoalt$separated_f%in%c(0)),0,
  #respondents dead
  ifelse((i_egoalt$separated_m%in%c(-8)|i_egoalt$separated_f%in%c(-8)),-8,
    ifelse((!is.na(i_egoalt$separated_m)|!is.na(i_egoalt$separated_f)),-9,NA)))))))))

label = "Whether living with at least one biological parent at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together with at least one parent at next wave" = 0,
  "separated from both parents at next wave" = 1,
  "possibly separated from both parents at next wave" = 2))

#living with both parents indicator

```

```

i_egoalt<-i_egoalt %>%group_by(pidp) %>%tidyr::fill(mother_pidp,.direction="updown")
i_egoalt<-i_egoalt %>%group_by(pidp) %>%tidyr::fill(father_pidp,.direction="updown")

#flags for living with mother or father this wave
i_egoalt<-i_egoalt%>%group_by(pidp)%>%mutate(livewithmother=ifelse(!is.na(i_egoalt$mother_
i_egoalt<-i_egoalt%>%group_by(pidp)%>%mutate(livewithfather=ifelse(!is.na(i_egoalt$father_

#living with both parents this wave
i_egoalt$liveboth<-c(0)
i_egoalt$liveboth[i_egoalt$livewithmother==1 & i_egoalt$livewithfather==1]<-1

#a column for living with mother, father, or both this wave
i_egoalt$lvpa<-c(0)
i_egoalt$lvpa[i_egoalt$livewithmother==1]<-1
i_egoalt$lvpa[i_egoalt$livewithfather==1]<-2
i_egoalt$lvpa[i_egoalt$liveboth==1]<-3
i_egoalt$lvpa[i_egoalt$lvpa==0]<-NA

i_egoalt$lvpa<-factor(i_egoalt$lvpa, levels=c(1,2,3),labels=c("live with mother","live wit

#select only for those who live with at least one parent this wave and at risk of leaving
i_egoalt<-i_egoalt[which(i_egoalt$lvpa%in%c("live with mother","live with father","live wi

#merge datafiles
i_indresp<-i_indresp%>%dplyr::select(pidp, i_hidp, i_mastat_dv,
  i_age_dv, i_sex_dv, i_hiqal_dv,
  i_mastat_dv, i_racel_dv,i_fimmnet_dv,i_nchild_dv,i_urban_dv,i_gor_dv,i_jbstat,i_s

i_egoaltt<-i_egoalt%>%dplyr::select(pidp,mother_pidp,father_pidp,separated_m,separated_f,s

#child levels
i_indrespc<-join_all(list(i_indresp,i_egoaltt), by='pidp', type='left')
i_indrespc<-i_indrespc %>% distinct_at(vars(pidp), .keep_all = TRUE)

i_indrespc<-i_indrespc[(which(!is.na(i_indrespc$mother_pidp))|!is.na(i_indrespc$father_pidp)

```

```

#add parent levels
i_indresppt <-i_indresp

#join parental data by pidp
father<-as.data.frame(i_indrespc$father_pidp)
mother<-as.data.frame(i_indrespc$mother_pidp)

colnames(father)[1]<-"pidp"
colnames(mother)[1]<-"pidp"

#attached parental values
i_indresppf<-join_all(list(father,i_indresppt), by='pidp', type='left')
i_indresppm<-join_all(list(mother,i_indresppt), by='pidp', type='left')

#for housing tenure
i_indresppf<-join_all(list(i_indresppf,i_hhresp), by='i_hidp', type='left')
i_indresppm<-join_all(list(i_indresppm,i_hhresp), by='i_hidp', type='left')

#rename var to differentiate parents
i_indresppf <-
  rename_at(i_indresppf, vars(starts_with("i_")), ~str_replace(., "i_", "ip_"))
i_indresppm <-
  rename_at(i_indresppm, vars(starts_with("i_")), ~str_replace(., "i_", "im_"))

#join to the child-level data by parent's pidp
colnames(i_indresppf)[1]<-"father_pidp"
colnames(i_indresppm)[1]<-"mother_pidp"

i_indresppf<-join_all(list(i_indrespc,i_indresppf), by="father_pidp", type='left')
i_indresppm<-join_all(list(i_indrespc,i_indresppm), by="mother_pidp", type='left')

i_indrespp<-join_all(list(i_indresppm,i_indresppf), by="pidp", type='left')

i_indrespp<-i_indrespp %>% distinct_at(vars(pidp), .keep_all = TRUE)

rm(father,mother,i_egoaltr,i_indrespc,i_indresppf,i_indresppm,i_indresppt)

#drop duplicate columns
i_indrespp <- i_indrespp %>% select(-contains(".1"))

```

```
## add cross wave identifier
i_indrespp$from_wave_i<-1
```

Wave 10

```
mothers <- j_egoalt %>%
  filter(j_relationship_dv %in% c(4) & j_asex==2) %>%
  select(pidp, apidp)
mothers$mother_pidp=mothers$apidp

j_egoalt <- j_egoalt %>%
  full_join(mothers, by=c("pidp", "apidp"))

rm(mothers)

fathers <- j_egoalt %>%
  filter(j_relationship_dv %in% c(4) & j_asex==1) %>%
  select(pidp, apidp)
fathers$father_pidp=fathers$apidp

j_egoalt <- j_egoalt %>%
  full_join(fathers, by=c("pidp", "apidp"))

rm(fathers)

other_relatives <- j_egoalt %>%
  filter(!(j_relationship_dv%in% c(1,2,3,4,9)) ) %>%
  select(pidp, apidp)
other_relatives$other_relatives_pidp=other_relatives$apidp

j_egoalt <- j_egoalt %>%
  full_join(other_relatives, by=c("pidp", "apidp"))

rm(other_relatives)

#at least two relatives in the household structure who are not the focal individual's parent
test<-j_egoalt%>%aggregate(other_relatives_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","relativenum")
j_egoalt<-join_all(list(j_egoalt,test),by="pidp",type="left")
rm(test)
```

```

#whether respondent live with a child
child <- j_egoalt %>%
  filter(j_relationship_dv%in% c(9) ) %>%
  select(pidp, apidp)
child$child_pidp=child$apidp
j_egoalt <- j_egoalt %>%
  full_join(child, by=c("pidp", "apidp"))
rm(child)

#count number of respondent children
test<-j_egoalt%>%aggregate(child_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","childnum")
j_egoalt<-join_all(list(j_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(marriage)
mar <- j_egoalt %>%
  filter(j_relationship_dv%in% c(1,2) ) %>%
  select(pidp, apidp)
mar$mar_pidp=mar$apidp
j_egoalt <- j_egoalt %>%
  full_join(mar, by=c("pidp", "apidp"))
rm(mar)

#count number of respondent married partner
test<-j_egoalt%>%aggregate(mar_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_husbandwife")
j_egoalt<-join_all(list(j_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(cohabitation)
cohab <- j_egoalt %>%
  filter(j_relationship_dv%in% c(3) ) %>%
  select(pidp, apidp)
cohab$cohab_pidp=cohab$apidp
j_egoalt <- j_egoalt %>%
  full_join(cohab, by=c("pidp", "apidp"))
rm(cohab)

#count number of cohabitee
test<-j_egoalt%>%aggregate(cohab_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_cohabitee")
j_egoalt<-join_all(list(j_egoalt,test),by="pidp",type="left")

```



```

rm(test)

# leave: not live with mothers
j_egoalt$separated_m <- labelled(
  ifelse(!is.na(j_egoalt$mother_pidp) & j_egoalt$j_enwstat==5), -8,
  ifelse(!is.na(j_egoalt$mother_pidp) & j_egoalt$j_anwstat==5), 3,
  ifelse(!is.na(j_egoalt$mother_pidp) &
    j_egoalt$j_anwstat==2 & j_egoalt$j_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(j_egoalt$mother_pidp) &
    j_egoalt$j_anwstat %in% c(3,4) & j_egoalt$j_enwstat==2), 2,
  ifelse(!is.na(j_egoalt$mother_pidp) &
    j_egoalt$j_anwstat==2 & j_egoalt$j_enwstat==2), 1,
  ifelse(!is.na(j_egoalt$mother_pidp) & j_egoalt$j_anwstat==1), 0,
  ifelse(!is.na(j_egoalt$mother_pidp)), -9, NA))))))

label = "Whether living with biological mother at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated_m at next wave" = 1,
  "possibly separated_m at next wave" = 2,
  "mother not alive at next wave" = 3))

#leave: not live with fathers
j_egoalt$separated_f <- labelled(
  ifelse(!is.na(j_egoalt$father_pidp) & j_egoalt$j_enwstat==5), -8,
  ifelse(!is.na(j_egoalt$father_pidp) & j_egoalt$j_anwstat==5), 3,
  ifelse(!is.na(j_egoalt$father_pidp) &
    j_egoalt$j_anwstat==2 & j_egoalt$j_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(j_egoalt$father_pidp) &
    j_egoalt$j_anwstat %in% c(3,4) & j_egoalt$j_enwstat==2), 2,
  ifelse(!is.na(j_egoalt$father_pidp) &
    j_egoalt$j_anwstat==2 & j_egoalt$j_enwstat==2), 1,
  ifelse(!is.na(j_egoalt$father_pidp) & j_egoalt$j_anwstat==1), 0,
  ifelse(!is.na(j_egoalt$father_pidp)), -9, NA))))))

label = "Whether living with biological father at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated at next wave" = 1,
  "possibly separated at next wave" = 2,

```

```

    "father not alive at next wave" = 3))

#fill the separate indicator
j_egoalt<-j_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_m,.direction="updown")
j_egoalt<-j_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_f,.direction="updown")

#find out who separate from both parents
#not live with both parents
j_egoalt$separated<- labelled(
  ifelse((j_egoalt$separated_m%in%c(1)&j_egoalt$separated_f%in%c(1)),1,

  ifelse((j_egoalt$separated_m%in%c(1)& is.na(j_egoalt$separated_f)),1,#one
  ifelse((j_egoalt$separated_f%in%c(1)& is.na(j_egoalt$separated_m)),1,

  ifelse((j_egoalt$separated_f%in%c(1)& j_egoalt$separated_m%in%c(3)),1,#death
  ifelse((j_egoalt$separated_m%in%c(1)& j_egoalt$separated_f%in%c(3)),1,

  #possibly separated
  ifelse((j_egoalt$separated_m%in%c(2)&j_egoalt$separated_f%in%c(2)),2,
  ifelse((j_egoalt$separated_m%in%c(1)&j_egoalt$separated_f%in%c(2)),2,
  ifelse((j_egoalt$separated_m%in%c(2)&j_egoalt$separated_f%in%c(1)),2,

  ifelse((j_egoalt$separated_m%in%c(2)& is.na(j_egoalt$separated_f)),2,#one
  ifelse((j_egoalt$separated_f%in%c(2)& is.na(j_egoalt$separated_m)),2,

  ifelse((j_egoalt$separated_f%in%c(2)& j_egoalt$separated_m%in%c(3)),1,#death
  ifelse((j_egoalt$separated_m%in%c(2)& j_egoalt$separated_f%in%c(3)),1,

  #living together
  ifelse((j_egoalt$separated_m%in%c(0)|j_egoalt$separated_f%in%c(0)),0,
  #respondents dead
  ifelse((j_egoalt$separated_m%in%c(-8)|j_egoalt$separated_f%in%c(-8)),-8,
    ifelse((!is.na(j_egoalt$separated_m)|!is.na(j_egoalt$separated_f)),-9,NA)))))))))

label = "Whether living with at least one biological parent at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together with at least one parent at next wave" = 0,
  "separated from both parents at next wave" = 1,

```

```

    "possibly separated from both parents at next wave" = 2))

#living with both parents indicator
j_egoalt<-j_egoalt %>%group_by(pidp) %>%tidyr::fill(mother_pidp,.direction="updown")
j_egoalt<-j_egoalt %>%group_by(pidp) %>%tidyr::fill(father_pidp,.direction="updown")

#flags for living with mother or father this wave
j_egoalt<-j_egoalt%>%group_by(pidp)%>%mutate(livewithmother=ifelse(!is.na(j_egoalt$mother_
j_egoalt<-j_egoalt%>%group_by(pidp)%>%mutate(livewithfather=ifelse(!is.na(j_egoalt$father_

#living with both parents this wave
j_egoalt$liveboth<-c(0)
j_egoalt$liveboth[j_egoalt$livewithmother==1 & j_egoalt$livewithfather==1]<-1

#a column for living with mother, father, or both this wave
j_egoalt$lvpa<-c(0)
j_egoalt$lvpa[j_egoalt$livewithmother==1]<-1
j_egoalt$lvpa[j_egoalt$livewithfather==1]<-2
j_egoalt$lvpa[j_egoalt$liveboth==1]<-3
j_egoalt$lvpa[j_egoalt$lvpa==0]<-NA

j_egoalt$lvpa<-factor(j_egoalt$lvpa, levels=c(1,2,3),labels=c("live with mother","live with
#select only for those who live with at least one parent this wave and at risk of leaving
j_egoalt<-j_egoalt[which(j_egoalt$lvpa%in%c("live with mother","live with father","live wi

#merge datafiles
j_indresp<-j_indresp%>%dplyr::select(pidp, j_hidp, j_mastat_dv,
    j_age_dv, j_sex_dv, j_hiqua_dv,
    j_mastat_dv, j_race1_dv,j_fimmnet_dv,j_nchild_dv,j_urban_dv,j_gor_dv,j_jbstat,j_s

j_egoaltt<-j_egoalt%>%dplyr::select(pidp,mother_pidp,father_pidp,separated_m,separated_f,s

#child levels
j_indrespc<-join_all(list(j_indresp,j_egoaltt), by='pidp', type='left')
j_indrespc<-j_indrespc %>% distinct_at(vars(pidp), .keep_all = TRUE)

```

```

j_indrespc<-j_indrespc[(which(!is.na(j_indrespc$mother_pidp))|!is.na(j_indrespc$father_pidp))
]

#add parent levels
j_indresppt <-j_indresp

#join parental data by pidp
father<-as.data.frame(j_indrespc$father_pidp)
mother<-as.data.frame(j_indrespc$mother_pidp)

colnames(father)[1]<-"pidp"
colnames(mother)[1]<-"pidp"

#attached parental values
j_indresppf<-join_all(list(father,j_indresppt), by='pidp', type='left')
j_indresppm<-join_all(list(mother,j_indresppt), by='pidp', type='left')

#for housing tenure
j_indresppf<-join_all(list(j_indresppf,j_hhresp), by='j_hidp', type='left')
j_indresppm<-join_all(list(j_indresppm,j_hhresp), by='j_hidp', type='left')

#rename var to differentiate parents
j_indresppf <-
  rename_at(j_indresppf, vars(starts_with("j_")), ~str_replace(., "j_", "ip_"))
j_indresppm <-
  rename_at(j_indresppm, vars(starts_with("j_")), ~str_replace(., "j_", "im_"))

#join to the child-level data by parent's pidp
colnames(j_indresppf)[1]<-"father_pidp"
colnames(j_indresppm)[1]<-"mother_pidp"

j_indresppf<-join_all(list(j_indrespc,j_indresppf), by="father_pidp", type='left')
j_indresppm<-join_all(list(j_indrespc,j_indresppm), by="mother_pidp", type='left')

j_indrespp<-join_all(list(j_indresppm,j_indresppf), by="pidp", type='left')

j_indrespp<-j_indrespp %>% distinct_at(vars(pidp), .keep_all = TRUE)

rm(father,mother,j_egoaltt,j_indrespc,j_indresppf,j_indresppm,j_indresppt)

```

```
#drop duplicate columns
j_indrespp <- j_indrespp %>% select(-contains(".1"))

j_indrespp$from_wave_j<-1
```

Wave 11

```
mothers <- k_egoalt %>%
  filter(k_relationship_dv %in% c(4) & k_asex==2) %>%
  select(pidp, apidp)
mothers$mother_pidp=mothers$apidp

k_egoalt <- k_egoalt %>%
  full_join(mothers, by=c("pidp", "apidp"))

rm(mothers)

fathers <- k_egoalt %>%
  filter(k_relationship_dv %in% c(4) & k_asex==1) %>%
  select(pidp, apidp)
fathers$father_pidp=fathers$apidp

k_egoalt <- k_egoalt %>%
  full_join(fathers, by=c("pidp", "apidp"))

rm(fathers)

other_relatives <- k_egoalt %>%
  filter(!(k_relationship_dv%in% c(1,2,3,4,9)) ) %>%
  select(pidp, apidp)
other_relatives$other_relatives_pidp=other_relatives$apidp

k_egoalt <- k_egoalt %>%
  full_join(other_relatives, by=c("pidp", "apidp"))

rm(other_relatives)

#at least two relatives in the household structure who are not the focal individual's parent
test<-k_egoalt%>%aggregate(other_relatives_pidp~pidp,FUN=length)
```

```

colnames(test)<-c("pidp","relativenum")
k_egoalt<-join_all(list(k_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent live with a child
child <- k_egoalt %>%
  filter(k_relationship_dv%in% c(9) ) %>%
  select(pidp, apidp)
child$child_pidp=child$apidp
k_egoalt <- k_egoalt %>%
  full_join(child, by=c("pidp", "apidp"))
rm(child)
#count number of respondent children
test<-k_egoalt%>%aggregate(child_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","childnum")
k_egoalt<-join_all(list(k_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(marriage)
mar <- k_egoalt %>%
  filter(k_relationship_dv%in% c(1,2) ) %>%
  select(pidp, apidp)
mar$mar_pidp=mar$apidp
k_egoalt <- k_egoalt %>%
  full_join(mar, by=c("pidp", "apidp"))
rm(mar)
#count number of respondent married partner
test<-k_egoalt%>%aggregate(mar_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_husbandwife")
k_egoalt<-join_all(list(k_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(cohabitation)
cohab <- k_egoalt %>%
  filter(k_relationship_dv%in% c(3) ) %>%
  select(pidp, apidp)
cohab$cohab_pidp=cohab$apidp
k_egoalt <- k_egoalt %>%
  full_join(cohab, by=c("pidp", "apidp"))

```

```

rm(cohab)
#count number of cohabitee
test<-k_egoalt%>%aggregate(cohab_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_cohabitee")
k_egoalt<-join_all(list(k_egoalt,test),by="pidp",type="left")
rm(test)

# leave: not live with mothers
k_egoalt$separated_m <- labelled(
  ifelse(!is.na(k_egoalt$mother_pidp) & k_egoalt$k_enwstat==5), -8,
  ifelse(!is.na(k_egoalt$mother_pidp) & k_egoalt$k_anwstat==5), 3,
  ifelse(!is.na(k_egoalt$mother_pidp) &
    k_egoalt$k_anwstat==2 & k_egoalt$k_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(k_egoalt$mother_pidp) &
    k_egoalt$k_anwstat %in% c(3,4) & k_egoalt$k_enwstat==2), 2,
  ifelse(!is.na(k_egoalt$mother_pidp) &
    k_egoalt$k_anwstat==2 & k_egoalt$k_enwstat==2), 1,
  ifelse(!is.na(k_egoalt$mother_pidp)& k_egoalt$k_anwstat==1), 0,
  ifelse(!is.na(k_egoalt$mother_pidp)), -9, NA))))),

label = "Whether living with biological mother at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated_m at next wave" = 1,
  "possibly separated_m at next wave" = 2,
  "mother not alive at next wave" = 3))

#leave: not live with fathers
k_egoalt$separated_f <- labelled(
  ifelse(!is.na(k_egoalt$father_pidp) & k_egoalt$k_enwstat==5), -8,
  ifelse(!is.na(k_egoalt$father_pidp) & k_egoalt$k_anwstat==5), 3,
  ifelse(!is.na(k_egoalt$father_pidp) &
    k_egoalt$k_anwstat==2 & k_egoalt$k_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(k_egoalt$father_pidp) &
    k_egoalt$k_anwstat %in% c(3,4) & k_egoalt$k_enwstat==2), 2,
  ifelse(!is.na(k_egoalt$father_pidp) &
    k_egoalt$k_anwstat==2 & k_egoalt$k_enwstat==2), 1,
  ifelse(!is.na(k_egoalt$father_pidp) & k_egoalt$k_anwstat==1), 0,
  ifelse(!is.na(k_egoalt$father_pidp)), -9, NA))))),

```

```

label = "Whether living with biological father at the next wave",
labels = c("missing" = -9,
           "ego not alive at next wave" = -8,
           "living together at next wave" = 0, "separated at next wave" = 1,
           "possibly separated at next wave" = 2,
           "father not alive at next wave" = 3))

#fill the separate indicator
k_egoalt<-k_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_m,.direction="updown")
k_egoalt<-k_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_f,.direction="updown")

#find out who separate from both parents
#not live with both parents
k_egoalt$separated<- labelled(
  ifelse((k_egoalt$separated_m%in%c(1)&k_egoalt$separated_f%in%c(1)),1,

  ifelse((k_egoalt$separated_m%in%c(1)& is.na(k_egoalt$separated_f)),1,#one
  ifelse((k_egoalt$separated_f%in%c(1)& is.na(k_egoalt$separated_m)),1,

  ifelse((k_egoalt$separated_f%in%c(1)& k_egoalt$separated_m%in%c(3)),1,#death
  ifelse((k_egoalt$separated_m%in%c(1)& k_egoalt$separated_f%in%c(3)),1,

  #possibly separated
  ifelse((k_egoalt$separated_m%in%c(2)&k_egoalt$separated_f%in%c(2)),2,
  ifelse((k_egoalt$separated_m%in%c(1)&k_egoalt$separated_f%in%c(2)),2,
  ifelse((k_egoalt$separated_m%in%c(2)&k_egoalt$separated_f%in%c(1)),2,

  ifelse((k_egoalt$separated_m%in%c(2)& is.na(k_egoalt$separated_f)),2,#one
  ifelse((k_egoalt$separated_f%in%c(2)& is.na(k_egoalt$separated_m)),2,

  ifelse((k_egoalt$separated_f%in%c(2)& k_egoalt$separated_m%in%c(3)),1,#death
  ifelse((k_egoalt$separated_m%in%c(2)& k_egoalt$separated_f%in%c(3)),1,

  #living together
  ifelse((k_egoalt$separated_m%in%c(0)|k_egoalt$separated_f%in%c(0)),0,
  #respondents dead
  ifelse((k_egoalt$separated_m%in%c(-8)|k_egoalt$separated_f%in%c(-8)),-8,
         ifelse((!is.na(k_egoalt$separated_m)|!is.na(k_egoalt$separated_f)),-9,NA)))))))))

```



```

label = "Whether living with at least one biological parent at the next wave",
labels = c("missing" = -9,
           "ego not alive at next wave" = -8,
           "living together with at least one parent at next wave" = 0,
           "separated from both parents at next wave" = 1,
           "possibly separated from both parents at next wave" = 2))

#living with both parents indicator
k_egoalt<-k_egoalt %>%group_by(pidp) %>%tidyr::fill(mother_pidp,.direction="updown")
k_egoalt<-k_egoalt %>%group_by(pidp) %>%tidyr::fill(father_pidp,.direction="updown")

#flags for living with mother or father this wave
k_egoalt<-k_egoalt%>%group_by(pidp)%>%mutate(livewithmother=ifelse(!is.na(k_egoalt$mother_
k_egoalt<-k_egoalt%>%group_by(pidp)%>%mutate(livewithfather=ifelse(!is.na(k_egoalt$father_

#living with both parents this wave
k_egoalt$liveboth<-c(0)
k_egoalt$liveboth[k_egoalt$livewithmother==1 & k_egoalt$livewithfather==1]<-1

#a column for living with mother, father, or both this wave
k_egoalt$lvpa<-c(0)
k_egoalt$lvpa[k_egoalt$livewithmother==1]<-1
k_egoalt$lvpa[k_egoalt$livewithfather==1]<-2
k_egoalt$lvpa[k_egoalt$liveboth==1]<-3
k_egoalt$lvpa[k_egoalt$lvpa==0]<-NA

k_egoalt$lvpa<-factor(k_egoalt$lvpa, levels=c(1,2,3),labels=c("live with mother","live wit

#select only for those who live with at least one parent this wave and at risk of leaving
k_egoalt<-k_egoalt[which(k_egoalt$lvpa%in%c("live with mother","live with father","live wi

#merge datafiles
k_indresp<-k_indresp%>%dplyr::select(pidp, k_hidp, k_mastat_dv,
                                     k_age_dv, k_sex_dv, k_hiqal_dv,
                                     k_mastat_dv, k_racel_dv,k_fimmnet_dv,k_nchild_dv,k_urban_dv,k_gor_dv,k_jbstat,k_s

k_egoaltt<-k_egoalt%>%dplyr::select(pidp,mother_pidp,father_pidp,separated_m,separated_f,s

```

```

#child levels
k_indrespc<-join_all(list(k_indresp,k_egoaltt), by='pidp', type='left')
k_indrespc<-k_indrespc %>% distinct_at(vars(pidp), .keep_all = TRUE)

k_indrespc<-k_indrespc[(which(!is.na(k_indrespc$mother_pidp))|!is.na(k_indrespc$father_pidp))

#add parent levels
k_indresppt <-k_indresp

#join parental data by pidp
father<-as.data.frame(k_indrespc$father_pidp)
mother<-as.data.frame(k_indrespc$mother_pidp)

colnames(father)[1]<-"pidp"
colnames(mother)[1]<-"pidp"

#attached parental values
k_indresppf<-join_all(list(father,k_indresppt), by='pidp', type='left')
k_indresppm<-join_all(list(mother,k_indresppt), by='pidp', type='left')

#for housing tenure
k_indresppf<-join_all(list(k_indresppf,k_hhresp), by='k_hidp', type='left')
k_indresppm<-join_all(list(k_indresppm,k_hhresp), by='k_hidp', type='left')

#rename var to differentiate parents
k_indresppf <-
  rename_at(k_indresppf, vars(starts_with("k_")), ~str_replace(., "k_", "ip_"))
k_indresppm <-
  rename_at(k_indresppm, vars(starts_with("k_")), ~str_replace(., "k_", "im_"))

#join to the child-level data by parent's pidp
colnames(k_indresppf)[1]<-"father_pidp"
colnames(k_indresppm)[1]<-"mother_pidp"

k_indresppf<-join_all(list(k_indrespc,k_indresppf), by="father_pidp", type='left')
k_indresppm<-join_all(list(k_indrespc,k_indresppm), by="mother_pidp", type='left')

k_indrespp<-join_all(list(k_indresppm,k_indresppf), by="pidp", type='left')

```

```

k_indrespp<-k_indrespp %>% distinct_at(vars(pidp), .keep_all = TRUE)

rm(father,mother,k_egoaltt,k_indrespc,k_indresppf,k_indresppm,k_indresppt)

#drop duplicate columns
k_indrespp <- k_indrespp %>% select(-contains(".1"))

k_indrespp$from_wave_k<-1

```

Wave 12

```

mothers <- l_egoalt %>%
  filter(l_relationship_dv %in% c(4) & l_asex==2) %>%
  select(pidp, apidp)
mothers$mother_pidp=mothers$apidp

l_egoalt <- l_egoalt %>%
  full_join(mothers, by=c("pidp", "apidp"))

rm(mothers)

fathers <- l_egoalt %>%
  filter(l_relationship_dv %in% c(4) & l_asex==1) %>%
  select(pidp, apidp)
fathers$father_pidp=fathers$apidp

l_egoalt <- l_egoalt %>%
  full_join(fathers, by=c("pidp", "apidp"))

rm(fathers)

other_relatives <- l_egoalt %>%
  filter(!(l_relationship_dv%in% c(1,2,3,4,9)) ) %>%
  select(pidp, apidp)
other_relatives$other_relatives_pidp=other_relatives$apidp

l_egoalt <- l_egoalt %>%
  full_join(other_relatives, by=c("pidp", "apidp"))

```

```

rm(other_relatives)

#at least two relatives in the household structure who are not the focal individual's parent
test<-l_egoalt%>%aggregate(other_relatives_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","relativenum")
l_egoalt<-join_all(list(l_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent live with a child
child <- l_egoalt %>%
  filter(l_relationship_dv%in% c(9) ) %>%
  select(pidp, apidp)
child$child_pidp=child$apidp
l_egoalt <- l_egoalt %>%
  full_join(child, by=c("pidp", "apidp"))
rm(child)
#count number of respondent children
test<-l_egoalt%>%aggregate(child_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","childnum")
l_egoalt<-join_all(list(l_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(marriage)
mar <- l_egoalt %>%
  filter(l_relationship_dv%in% c(1,2) ) %>%
  select(pidp, apidp)
mar$mar_pidp=mar$apidp
l_egoalt <- l_egoalt %>%
  full_join(mar, by=c("pidp", "apidp"))
rm(mar)
#count number of respondent married partner
test<-l_egoalt%>%aggregate(mar_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_husbandwife")
l_egoalt<-join_all(list(l_egoalt,test),by="pidp",type="left")
rm(test)

#whether respondent cohabit with a partner(cohabitation)
cohab <- l_egoalt %>%
  filter(l_relationship_dv%in% c(3) ) %>%

```

```

    select(pidp, apidp)
cohab$cohab_pidp=cohab$apidp
l_egoalt <- l_egoalt %>%
  full_join(cohab, by=c("pidp", "apidp"))
rm(cohab)

#count number of cohabitee
test<-l_egoalt%>%aggregate(cohab_pidp~pidp,FUN=length)
colnames(test)<-c("pidp","live_with_cohabitee")
l_egoalt<-join_all(list(l_egoalt,test),by="pidp",type="left")
rm(test)

# leave: not live with mothers
l_egoalt$separated_m <- labelled(
  ifelse(!is.na(l_egoalt$mother_pidp) & l_egoalt$l_enwstat==5), -8,
  ifelse(!is.na(l_egoalt$mother_pidp) & l_egoalt$l_anwstat==5), 3,
  ifelse(!is.na(l_egoalt$mother_pidp) &
    l_egoalt$l_anwstat==2 & l_egoalt$l_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(l_egoalt$mother_pidp) &
    l_egoalt$l_anwstat %in% c(3,4) & l_egoalt$l_enwstat==2), 2,
  ifelse(!is.na(l_egoalt$mother_pidp) &
    l_egoalt$l_anwstat==2 & l_egoalt$l_enwstat==2), 1,
  ifelse(!is.na(l_egoalt$mother_pidp)& l_egoalt$l_anwstat==1), 0,
  ifelse(!is.na(l_egoalt$mother_pidp)), -9, NA))))),

label = "Whether living with biological mother at the next wave",
labels = c("missing" = -9,
  "ego not alive at next wave" = -8,
  "living together at next wave" = 0, "separated_m at next wave" = 1,
  "possibly separated_m at next wave" = 2,
  "mother not alive at next wave" = 3))

#leave: not live with fathers
l_egoalt$separated_f <- labelled(
  ifelse(!is.na(l_egoalt$father_pidp) & l_egoalt$l_enwstat==5), -8,
  ifelse(!is.na(l_egoalt$father_pidp) & l_egoalt$l_anwstat==5), 3,
  ifelse(!is.na(l_egoalt$father_pidp) &
    l_egoalt$l_anwstat==2 & l_egoalt$l_enwstat %in% c(3,4)), 2,
  ifelse(!is.na(l_egoalt$father_pidp) &
    l_egoalt$l_anwstat %in% c(3,4) & l_egoalt$l_enwstat==2), 2,
  ifelse(!is.na(l_egoalt$father_pidp) &

```

```

        l_egoalt$l_anwstat==2 & l_egoalt$l_enwstat==2), 1,
    ifelse(!is.na(l_egoalt$father_pidp) & l_egoalt$l_anwstat==1), 0,
    ifelse(!is.na(l_egoalt$father_pidp)), -9, NA))))),

label = "Whether living with biological father at the next wave",
labels = c("missing" = -9,
           "ego not alive at next wave" = -8,
           "living together at next wave" = 0, "separated at next wave" = 1,
           "possibly separated at next wave" = 2,
           "father not alive at next wave" = 3))

#fill the separate indicator
l_egoalt<-l_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_m,.direction="updown")
l_egoalt<-l_egoalt %>%group_by(pidp) %>%tidyr::fill(separated_f,.direction="updown")

#find out who separate from both parents
#not live with both parents
l_egoalt$separated<- labelled(
  ifelse((l_egoalt$separated_m%in%c(1)&l_egoalt$separated_f%in%c(1)),1,

  ifelse((l_egoalt$separated_m%in%c(1)& is.na(l_egoalt$separated_f)),1,#one
  ifelse((l_egoalt$separated_f%in%c(1)& is.na(l_egoalt$separated_m)),1,

  ifelse((l_egoalt$separated_f%in%c(1)& l_egoalt$separated_m%in%c(3)),1,#death
  ifelse((l_egoalt$separated_m%in%c(1)& l_egoalt$separated_f%in%c(3)),1,

  #possibly separated
  ifelse((l_egoalt$separated_m%in%c(2)&l_egoalt$separated_f%in%c(2)),2,
  ifelse((l_egoalt$separated_m%in%c(1)&l_egoalt$separated_f%in%c(2)),2,
  ifelse((l_egoalt$separated_m%in%c(2)&l_egoalt$separated_f%in%c(1)),2,

  ifelse((l_egoalt$separated_m%in%c(2)& is.na(l_egoalt$separated_f)),2,#one
  ifelse((l_egoalt$separated_f%in%c(2)& is.na(l_egoalt$separated_m)),2,

  ifelse((l_egoalt$separated_f%in%c(2)& l_egoalt$separated_m%in%c(3)),1,#death
  ifelse((l_egoalt$separated_m%in%c(2)& l_egoalt$separated_f%in%c(3)),1,

  #living together
  ifelse((l_egoalt$separated_m%in%c(0)|l_egoalt$separated_f%in%c(0)),0,

```

```

#respondents dead
ifelse((l_egoalt$separated_m%in%c(-8)|l_egoalt$separated_f%in%c(-8)),-8,
       ifelse(!is.na(l_egoalt$separated_m)|!is.na(l_egoalt$separated_f)),-9,NA)))))))))

label = "Whether living with at least one biological parent at the next wave",
labels = c("missing" = -9,
           "ego not alive at next wave" = -8,
           "living together with at least one parent at next wave" = 0,
           "separated from both parents at next wave" = 1,
           "possibly separated from both parents at next wave" = 2))

#living with both parents indicator
l_egoalt<-l_egoalt %>%group_by(pidp) %>%tidyr::fill(mother_pidp,.direction="updown")
l_egoalt<-l_egoalt %>%group_by(pidp) %>%tidyr::fill(father_pidp,.direction="updown")

#flags for living with mother or father this wave
l_egoalt<-l_egoalt%>%group_by(pidp)%>%mutate(livewithmother=ifelse(!is.na(l_egoalt$mother_
l_egoalt<-l_egoalt%>%group_by(pidp)%>%mutate(livewithfather=ifelse(!is.na(l_egoalt$father_

#living with both parents this wave
l_egoalt$liveboth<-c(0)
l_egoalt$liveboth[l_egoalt$livewithmother==1 & l_egoalt$livewithfather==1]<-1

#a column for living with mother, father, or both this wave
l_egoalt$lvpa<-c(0)
l_egoalt$lvpa[l_egoalt$livewithmother==1]<-1
l_egoalt$lvpa[l_egoalt$livewithfather==1]<-2
l_egoalt$lvpa[l_egoalt$liveboth==1]<-3
l_egoalt$lvpa[l_egoalt$lvpa==0]<-NA

l_egoalt$lvpa<-factor(l_egoalt$lvpa, levels=c(1,2,3),labels=c("live with mother","live with
#select only for those who live with at least one parent this wave and at risk of leaving
l_egoalt<-l_egoalt[which(l_egoalt$lvpa%in%c("live with mother","live with father","live wi

#merge datafiles
l_indresp<-l_indresp%>%dplyr::select(pidp, l_hidp, l_mastat_dv,
                                     l_age_dv, l_sex_dv, l_hiqua_dv,

```

```

l_mastat_dv, l_racel_dv, l_fimmnet_dv, l_nchild_dv, l_urban_dv, l_gor_dv, l_jbstat, l_s

l_egoaltt<-l_egoalt%>%dplyr::select(pidp,mother_pidp,father_pidp,separated_m,separated_f,s

#child levels
l_indrespc<-join_all(list(l_indresp,l_egoaltt), by='pidp', type='left')
l_indrespc<-l_indrespc %>% distinct_at(vars(pidp), .keep_all = TRUE)

l_indrespc<-l_indrespc[(which(!is.na(l_indrespc$mother_pidp))|!is.na(l_indrespc$father_pidp

#add parent levels
l_indresppt <-l_indresp

#join parental data by pidp
father<-as.data.frame(l_indrespc$father_pidp)
mother<-as.data.frame(l_indrespc$mother_pidp)

colnames(father)[1]<-"pidp"
colnames(mother)[1]<-"pidp"

#attached parental values
l_indresppf<-join_all(list(father,l_indresppt), by='pidp', type='left')
l_indresppm<-join_all(list(mother,l_indresppt), by='pidp', type='left')

#for housing tenure
l_indresppf<-join_all(list(l_indresppf,l_hhresp), by='l_hidp', type='left')
l_indresppm<-join_all(list(l_indresppm,l_hhresp), by='l_hidp', type='left')

#rename var to differentiate parents
l_indresppf <-
  rename_at(l_indresppf, vars(starts_with("l_")), ~str_replace(., "l_", "ip_"))
l_indresppm <-
  rename_at(l_indresppm, vars(starts_with("l_")), ~str_replace(., "l_", "im_"))

#join to the child-level data by parent's pidp
colnames(l_indresppf)[1]<-"father_pidp"
colnames(l_indresppm)[1]<-"mother_pidp"

```



```

l_indresppf<-join_all(list(l_indrespc,l_indresppf), by="father_pidp", type='left')
l_indresppm<-join_all(list(l_indrespc,l_indresppm), by="mother_pidp", type='left')

l_indrespp<-join_all(list(l_indresppm,l_indresppf), by="pidp", type='left')

l_indrespp<-l_indrespp %>% distinct_at(vars(pidp), .keep_all = TRUE)

rm(father,mother,l_egoaltt,l_indrespc,l_indresppf,l_indresppm,l_indresppt)

#drop duplicate columns
l_indrespp <- l_indrespp %>% select(-contains(".1"))

l_indrespp$from_wave_l<-1

```

Merging all

```

#remove all the prefixes
h_indrespp <-
  rename_at(h_indrespp, vars(starts_with("h_")), ~str_replace(., "h_", ""))
i_indrespp <-
  rename_at(i_indrespp, vars(starts_with("i_")), ~str_replace(., "i_", ""))
j_indrespp <-
  rename_at(j_indrespp, vars(starts_with("j_")), ~str_replace(., "j_", ""))
k_indrespp <-
  rename_at(k_indrespp, vars(starts_with("k_")), ~str_replace(., "k_", ""))
l_indrespp <-
  rename_at(l_indrespp, vars(starts_with("l_")), ~str_replace(., "l_", ""))

indrespb<-rbind.fill(h_indrespp,i_indrespp,j_indrespp,k_indrespp,l_indrespp)

```

Coding variables

child level

```

#child level

#level of education completed
indrespb$edu<-c(0)

```

```

indrespb$edu[indrespb$hiqual_dv%in%c(9)]<-1 #low (everything below)
indrespb$edu[indrespb$hiqual_dv%in%c(3,4,5)]<-2 #middle (completed A-level or secondary hi
indrespb$edu[indrespb$hiqual_dv%in%c(1,2)]<-3 #high (have a degree)
indrespb$edu[indrespb$edu==0]<-NA

```

```

print(attr(indrespb$hiqual_dv,"labels"))

```

missing	inapplicable	refusal	don't know
-9	-8	-2	-1
Degree Other higher degree	A-level etc	GCSE etc	
1	2	3	4
Other qualification	No qualification		
5	9		

```

indrespb$edu<-factor(indrespb$edu,levels=c(1,2,3),labels=c('low',"middle","high"))

```

```

#two-category education

```

```

indrespb$edu2<-c(0)

```

```

indrespb$edu2[indrespb$hiqual_dv%in%c(1,2)]<-1 #high (have a degree)

```

```

indrespb$edu2[is.na(indrespb$hiqual_dv)]<-NA

```

```

#family structure

```

```

indrespb$family_status<-c(0)

```

```

#coresident partner and child

```

```

indrespb$family_status[indrespb$childnum>=1 & indrespb$live_with_husbandwife>=1]<-1

```

```

indrespb$family_status[indrespb$childnum>=1 & indrespb$live_with_cohabitee>=1]<-1

```

```

#coresident partner

```

```

indrespb$family_status[indrespb$live_with_cohabitee>=1 & indrespb$childnum==0]<-2

```

```

indrespb$family_status[indrespb$live_with_husbandwife>=1 & indrespb$childnum==0]<-2

```

```

indrespb$family_status[indrespb$live_with_cohabitee>=1 & is.na(indrespb$childnum)]<-2

```

```

indrespb$family_status[indrespb$live_with_husbandwife>=1 & is.na(indrespb$childnum)]<-2

```

```

#coresident child

```

```

indrespb$family_status[indrespb$childnum>=1 & indrespb$live_with_husbandwife==0 & indrespb$

```

```

indrespb$family_status[indrespb$childnum>=1 & is.na(indrespb$live_with_husbandwife) & indres

```

```

indrespb$family_status[indrespb$childnum>=1 & indrespb$live_with_husbandwife==0 & is.na(ind

```

```

indrespb$family_status[indrespb$childnum>=1 & is.na(indrespb$live_with_husbandwife) & is.na

```

```

#no coresident partner or child=0

```

```

indrespb$family_status<-factor(indrespb$family_status,levels=c(0,1,2,3),labels=c("No cores

```

```

#age centered
indrespb$age_centered<-indrespb$age_dv-16
#age squared
indrespb$age_sq<-indrespb$age_centered**2

#logarithm of income
indrespb$fimnnet_dv[indrespb$fimnnet_dv<0]<-0
#add 1 to all income values to avoid -inf in log transformation
indrespb$incomeadd<-indrespb$fimnnet_dv+1
indrespb$lnincome<-log(indrespb$incomeadd)
indrespb<-indrespb%>%dplyr::select(-incomeadd)

#racel groups
indrespb$racel<-c(0)
indrespb$racel[indrespb$racel_dv%in%c(1,2,3,4)]<-1 #Whites
indrespb$racel[indrespb$racel_dv%in%c(14,15)]<-2 #African Caribbean
indrespb$racel[indrespb$racel_dv%in%c(9,10,11)]<-3 #Indiani,Pakistani, Bangladeshi
indrespb$racel[indrespb$racel_dv%in%c(12,13)]<-4 #Other Asian
indrespb$racel[indrespb$racel==0]<-5 #Other

indrespb$racel<-factor(indrespb$racel,levels=c(1,2,3,4,5),labels=c("Whites","African Carib

indrespb$racel[is.na(indrespb$racel_dv)]<-NA

##two-categories
indrespb$racel2<-c(0)
indrespb$racel2[indrespb$racel_dv%in%c(1,2,3,4)]<-1 #Whites
indrespb$racel2[!(indrespb$racel_dv%in%c(1,2,3,4))]<-2 #non-Whites
indrespb$racel2[is.na(indrespb$racel_dv)]<-NA

indrespb$racel2<-factor(indrespb$racel2,levels=c(1,2),labels=c("Whites","non-Whites"))

#education status
indrespb$student<-c(0)
indrespb$student[indrespb$jbststat%in%c(7)]<-1 #full-time student
indrespb$student[is.na(indrespb$jbststat)]<-NA

#employment status
indrespb$unemployed<-c(0)

```

```

indrespb$unemployed[indrespb$jbststat%in%c(3)]<-1
indrespb$unemployed[is.na(indrespb$jbststat)]<-NA

#housing prices
indrespb$houseprices<-c(0)
indrespb$houseprices[indrespb$gor_dv%in%c(1,10,11,12)]<-1 #low
indrespb$houseprices[indrespb$gor_dv%in%c(2,3,4,5)]<-2 #middle
indrespb$houseprices[indrespb$gor_dv%in%c(6,7,8,9)]<-3 #high
indrespb$houseprices[is.na(indrespb$gor_dv)]<-NA

indrespb$houseprices<-factor(indrespb$houseprices,levels=c(1,2,3),labels=c("low","middle",

#urban,sex binary
indrespb$sex<-c(0)
indrespb$sex[indrespb$sex_dv==2]<-1 #females
indrespb$sex[is.na(indrespb$sex_dv)]<-NA

indrespb$urban<-c(0)
indrespb$urban[indrespb$urban_dv==1]<-1 #urban areas
indrespb$urban[is.na(indrespb$urban_dv)]<-NA

indrespb<-indrespb%>%mutate(health=coalesce(sf1,scsf1))

```

Warning: `..1` and `..2` have conflicting value labels.
i Labels for these values will be taken from `..1`.
x Values: 5

```

#living with both biological parents
indrespb$liveboth<-c(0)
indrespb$liveboth[indrespb$lvpa=="live with both parents"]<-1
indrespb$liveboth[is.na(indrespb$lvpa)]<-NA

#relative num for family structure
indrespb$relative_num<-c(0)
indrespb$relative_num[indrespb$relativenum>=2]<-1
indrespb$relative_num<-factor(indrespb$relative_num,levels=c(0,1),labels=c("small hh","lar

```

parental level

```

#level of education completed
#Fathers:

```

```

indrespb$paedu<-c(0)
indrespb$paedu[indrespb$ip_higual_dv%in%c(9)]<-1 #low (everything below)
indrespb$paedu[indrespb$ip_higual_dv%in%c(3,4,5)]<-2 #middle (completed A-level or seconda
indrespb$paedu[indrespb$ip_higual_dv%in%c(1,2)]<-3 #high (have a degree)
indrespb$paedu[indrespb$paedu==0]<-NA

print(attr(indrespb$ip_higual_dv,"labels"))

```

missing	inapplicable	refusal	don't know
-9	-8	-2	-1
Degree Other	higher degree	A-level etc	GCSE etc
1	2	3	4
Other qualification	No qualification		
5	9		

```

indrespb$paedu<-factor(indrespb$paedu,levels=c(1,2,3),labels=c('low',"middle","high"))

#Mothers:
indrespb$maedu<-c(0)
indrespb$maedu[indrespb$im_higual_dv%in%c(9)]<-1 #low (everything below)
indrespb$maedu[indrespb$im_higual_dv%in%c(3,4,5)]<-2 #middle (completed A-level or seconda
indrespb$maedu[indrespb$im_higual_dv%in%c(1,2)]<-3 #high (have a degree)
indrespb$maedu[indrespb$maedu==0]<-NA

indrespb$maedu<-factor(indrespb$maedu,levels=c(1,2,3),labels=c('low',"middle","high"))

#the highest education level between two parents
indrespb<-transform(indrespb, parentedu = pmin(im_higual_dv, ip_higual_dv))
indrespb$parentedu<-ifelse(is.na(indrespb$parentedu), indrespb$im_higual_dv, indrespb$pare
indrespb$parentedu<-ifelse(is.na(indrespb$parentedu), indrespb$ip_higual_dv, indrespb$pare

indrespb$parent_edu<-c(0)
indrespb$parent_edu[indrespb$parentedu%in%c(9)]<-1 #low (everything below, the category "o
indrespb$parent_edu[indrespb$parentedu%in%c(3,4,5)]<-2 #middle (complete A-level or have s
indrespb$parent_edu[indrespb$parentedu%in%c(1,2)]<-3 #high (have a degree)
indrespb$parent_edu[indrespb$parentedu==0]<-NA

indrespb$parent_edu<-factor(indrespb$parent_edu,levels=c(1,2,3),labels=c('low',"middle","h

#two-category education

```

```

indrespb$parent_edu2<-c(0)
indrespb$parent_edu2[indrespb$parentedu%in%c(1,2)]<-1 #high (have a degree)
indrespb$parent_edu2[is.na(indrespb$parentedu)]<-NA

indrespb<-indrespb%>%dplyr::select(-parentedu)

#general health of parents**
indrespb<-indrespb%>%mutate(health_father=coalesce(ip_sf1, ip_scsf1))

```

Warning: `..1` and `..2` have conflicting value labels.
i Labels for these values will be taken from `..1`.
x Values: 5

```

indrespb<-indrespb%>%mutate(health_mother=coalesce(im_sf1, im_scsf1))

```

Warning: `..1` and `..2` have conflicting value labels.
i Labels for these values will be taken from `..1`.
x Values: 5

```

#least healthy parent
indrespb<-transform(indrespb, parenthealth = pmax(health_father, health_mother))
indrespb$parenthealth<-ifelse(is.na(indrespb$parenthealth), indrespb$health_father, indrespb$health_mother)
indrespb$parenthealth<-ifelse(is.na(indrespb$parenthealth), indrespb$health_mother, indrespb$health_father)

#parent's age (youngest)
indrespb <- transform(indrespb, parentage = pmin(ip_age_dv
, im_age_dv))
indrespb$parentage<-ifelse(is.na(indrespb$parentage), indrespb$ip_age_dv, indrespb$im_age_dv)
indrespb$parentage<-ifelse(is.na(indrespb$parentage), indrespb$im_age_dv, indrespb$ip_age_dv)

#parent's income (sum and logged)
indrespb$palnincome<-indrespb$ip_fimnnet_dv+indrespb$im_fimnnet_dv
indrespb$palnincome[indrespb$palnincome<0]<-0
#add 1 to all income values to avoid -inf in log transformation
indrespb$paincomeadd<-indrespb$palnincome+1
indrespb$palnincome<-log(indrespb$paincomeadd)
indrespb<-indrespb%>%dplyr::select(-paincomeadd)

#parent's income (quartiles)

```

```
indrespb$paincome<-indrespb$ip_fimnnet_dv+indrespb$im_fimnnet_dv
print(quantile(indrespb$paincome, prob=c(.25,.5,.75), type=1,na.rm=T))
```

```
      25%      50%      75%
1928.18 2946.67 4192.75
```

```
indrespb$paincomeq<-c(0)
indrespb$paincomeq[indrespb$paincome>=0 & indrespb$paincome<1928.18]<-1
indrespb$paincomeq[indrespb$paincome>=1928.18 & indrespb$paincome<2946.67]<-2
indrespb$paincomeq[indrespb$paincome>=2946.67 & indrespb$paincome<4192.75 ]<-3
indrespb$paincomeq[indrespb$paincome>=4192.75]<-4
```

```
#parental tenure
indrespb$mature<-c(0)
indrespb$mature[indrespb$im_tenure_dv%in%c(1,2)]<-1 #ownership
indrespb$mature[indrespb$im_tenure_dv%in%c(3,4,5)]<-2 #social rent
indrespb$mature[indrespb$im_tenure_dv%in%c(6,7,8)]<-3 #private rent n other
indrespb$mature[indrespb$mature==0]<-NA
```

```
indrespb$mature<-factor(indrespb$mature,levels=c(1,2,3),labels=c("ownership","social rent"))
```

```
indrespb$pature<-c(0)
indrespb$pature[indrespb$ip_tenure_dv%in%c(1,2)]<-1 #ownership
indrespb$pature[indrespb$ip_tenure_dv%in%c(3,4,5)]<-2 #social rent
indrespb$pature[indrespb$ip_tenure_dv%in%c(6,7,8)]<-3 #private rent n other
indrespb$pature[indrespb$pature==0]<-NA
```

```
indrespb$pature<-factor(indrespb$pature,levels=c(1,2,3),labels=c("ownership","social rent"))
```

```
#the parents who have the most adv housing
indrespb <- transform(indrespb, housingtenure = pmin(ip_tenure_dv, im_tenure_dv))
indrespb$housingtenure<-ifelse(is.na(indrespb$housingtenure), indrespb$ip_tenure_dv, indrespb$im_tenure_dv)
indrespb$housingtenure<-ifelse(is.na(indrespb$housingtenure), indrespb$im_tenure_dv, indrespb$ip_tenure_dv)
```

```
indrespb$housing_tenure<-c(0)
indrespb$housing_tenure[indrespb$housingtenure%in%c(1,2)]<-1 #ownership
indrespb$housing_tenure[indrespb$housingtenure%in%c(3,4,5)]<-2 #social rent
indrespb$housing_tenure[indrespb$housingtenure%in%c(6,7,8)]<-3 #private rent n other
indrespb$housing_tenure[indrespb$housing_tenure==0]<-NA
indrespb$housing_tenure<-factor(indrespb$housing_tenure,levels=c(1,2,3),labels=c("ownership","social rent"))
```

```

indrespb<-indrespb%>%dplyr::select(-housingtenure)

#cross_wave n interaction variables
#age * parental lnincome
indrespb$paincomeXage<-indrespb$palnincome*indrespb$age_centered
#age * parental lnincome quartiles
indrespb$paincomeqXage<-indrespb$paincomeq*indrespb$age_centered

#cross wave identifiers
indrespb$wave<-c(0)
indrespb$wave[indrespb$from_wave_h==1]<-1
indrespb$wave[indrespb$from_wave_i==1]<-2
indrespb$wave[indrespb$from_wave_j==1]<-3
indrespb$wave[indrespb$from_wave_k==1]<-4
indrespb$wave[indrespb$from_wave_l==1]<-5

indrespb$wave<-factor(indrespb$wave,levels=c(1,2,3,4,5),labels=c("wave 8","wave 9","wave 10"))

#indrespb<-indrespb%>%dplyr::select(-from_wave_h,-from_wave_i,-from_wave_j,-from_wave_k,-from_wave_l)

```

Dependent variables

```

indrespb$leave<-c(0)
indrespb$leave[indrespb$separated%in%c(1,2)]<-1

#wave h leavers
hleave<-indrespb[which(indrespb$leave==1&indrespb$from_wave_h==1),]
#join with i indresp for information
hleave<-join_all(list(hleave,i_indresp), by='pidp', type='left')

hleave$out<-c(0)
#leaving for marriage
hleave$out[hleave$i_mastat_dv%in%c(2,3)& hleave$i_jbstat!=7]<-1

#leaving for cohabitation
hleave$out[hleave$i_mastat_dv%in%c(10) & hleave$i_jbstat!=7]<-2

#leaving as a full-time student
hleave$out[hleave$i_jbstat%in%c(7)]<-3

```



```

#leaving for other reasons=0

#wave i leavers
ileave<-indrespb[which(indrespb$leave==1&indrespb$from_wave_i==1),]
#join with j indresp for information
ileave<-join_all(list(ileave, j_indresp), by='pidp', type='left')

ileave$out<-c(0)
#leaving for marriage
ileave$out[ileave$j_mastat_dv%in%c(2,3) & ileave$j_jbstat!=7]<-1

#leaving for cohabitation
ileave$out[ileave$j_mastat_dv%in%c(10) & ileave$j_jbstat!=7]<-2

#leaving as a full-time student
ileave$out[ileave$j_jbstat%in%c(7)]<-3

#leaving for other reasons=0

#wave j leavers
jleave<-indrespb[which(indrespb$leave==1&indrespb$from_wave_j==1),]
#join with k indresp for information
jleave<-join_all(list(jleave, k_indresp), by='pidp', type='left')

jleave$out<-c(0)
#leaving for marriage
jleave$out[jleave$k_mastat_dv%in%c(2,3) & jleave$k_jbstat!=7]<-1

#leaving for cohabitation
jleave$out[jleave$k_mastat_dv%in%c(10) & jleave$k_jbstat!=7]<-2

#leaving as a full-time student
jleave$out[jleave$k_jbstat%in%c(7)]<-3

#leaving for other reasons=0

#wave k leavers
kleave<-indrespb[which(indrespb$leave==1&indrespb$from_wave_k==1),]
#join with l indresp for information

```

```

kleave<-join_all(list(kleave, l_indresp), by='pidp', type='left')

kleave$out<-c(0)
#leaving for marriage
kleave$out[kleave$l_mastat_dv%in%c(2,3) & kleave$l_jbstat!=7]<-1

#leaving for coiabitation
kleave$out[kleave$l_mastat_dv%in%c(10) & kleave$l_jbstat!=7]<-2

#leaving as a full-time student
kleave$out[kleave$l_jbstat%in%c(7)]<-3

#leaving for other reasons=0

#wave 1 leavers
lleave<-indrespb[which(indrespb$leave==1&indrespb$from_wave_1==1),]
#join with m indresp for information
lleave<-join_all(list(lleave, m_indresp), by='pidp', type='left')

lleave$out<-c(0)
#leaving for marriage
lleave$out[lleave$m_mastat_dv%in%c(2,3) & lleave$m_jbstat!=7]<-1

#leaving for coiabitation
lleave$out[lleave$m_mastat_dv%in%c(10) & lleave$m_jbstat!=7]<-2

#leaving as a full-time student
lleave$out[lleave$m_jbstat%in%c(7)]<-3

#leaving for other reasons=0
out<-rbind.fill(hleave,ileave,jleave,kleave,lleave)
out<-out%>%dplyr::select(pidp,wave,out)
indrespb<-join_all(list(indrespb,out), by=c("pidp", "wave"), type='left')

#mar
indrespb$mar<-c(0)
indrespb$mar[indrespb$out==1]<-1

#cohab
indrespb$cohab<-c(0)
indrespb$cohab[indrespb$out==2]<-1

```

```
#student
indrespb$student<-c(0)
indrespb$student[indrespb$out==3]<-1

#other
indrespb$other<-c(0)
indrespb$other[indrespb$out==0]<-1
```

Model building

Wave 1-5

Wave 8-12

```
indrespbby<-indrespb[which(indrespb$age_centered%in%c(0:14)),]
indrespbbyf<-indrespb[which(indrespb$age_centered%in%c(0:14)&indrespb$sex==1),]
indrespbmy<-indrespb[which(indrespb$age_centered%in%c(0:14)&indrespb$sex==0),]

mar_bf<-glm(mar~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))

cohab_bf<-glm(cohab~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))

student_bf<-glm(student~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

Warning: glm.fit: algorithm did not converge

```
all_bf<-glm(leave~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))

#paincomeXage effects more strongly for men
mar_bm<-glm(mar~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

```
cohab_bm<-glm(cohab~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
student_bm<-glm(student~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

Warning: glm.fit: algorithm did not converge

```
all_bm<-glm(leave~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
mar_b<-glm(mar~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
cohab_b<-glm(cohab~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
student_b<-glm(student~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

Warning: glm.fit: algorithm did not converge

```
other_b<-glm(other~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
all_b<-glm(leave~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure))
```

```
#how many boomerang
```

```
counts <- indrespb%>%dplyr::count(pidp,separated)
```

```
counts2<-data.frame(unclass(table(counts$pidp,counts$separated)))
```

```
counts2<-counts2%>%mutate(leave=coalesce(X1, X2))
```

```
counts2<-counts2%>%mutate(missing_dead=coalesce(X.9 , X.8))
```

```
test<-counts2$X0+counts2$leave
```

```
prop.table(table(test))
```

```
test
```

```
      0      1      2  
0.13213945 0.79771114 0.07014941
```

```
0.2086468
```

```
[1] 0.2086468
```

Plotting results

```
convertModel <- function(model) {
  tr <- createTexreg(
    coef.names = names(model$coef),
    coef = as.numeric(summary(model)$coefficients[,1]),
    se = as.numeric(summary(model)$coefficients[,2]),
    pvalues = as.numeric(summary(model)$coefficients[,4]),
    gof.names = c("N","AIC"),
    gof = c(nobs(all_b),AIC(model)),
    gof.decimal = c(F,F)

  )
}

#compare males and females in wave 8-12

p1<-huxtablereg(lapply(list(all_bf,all_bm,mar_bf,mar_bm,cohab_bf,cohab_bm,student_bf,student_bm),
  #full and nested models for all_b
  nest1<-glm(leave~age_centered,data=indrespby,family=binomial(link="logit"))

  nest2<-glm(leave~age_centered+factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure),data=indrespby,family=binomial(link="logit"))

  p2<-huxtablereg(lapply(list(nest1,nest2,all_b), convertModel))

  nest1<-glm(leave~age_centered,data=indrespby,family=binomial(link="logit"))

  nest2<-glm(leave~age_centered+factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure),data=indrespby,family=binomial(link="logit"))

  all_b<-glm(leave~factor(paincomeq)+paincomeqXage+factor(parent_edu2)+factor(housing_tenure),data=indrespby,family=binomial(link="logit"))

  Summarize all leavers (all, men and women)

  print(summary(all_b))
```

Call:

```
glm(formula = leave ~ factor(paincomeq) + paincomeqXage + factor(parent_edu2) +
```

```

factor(housing_tenure) + parenthealth + parentage + factor(liveboth) +
age_centered + factor(sex) + lnincome + factor(edu2) + factor(family_status) +
factor(racel2) + factor(student) + factor(unemployed) + factor(urban) +
factor(houseprices) + factor(wave), family = binomial(link = "logit"),
data = indrespby)

```

Coefficients:

	Estimate	Std. Error	z value
(Intercept)	-2.356393	0.292362	-8.060
factor(paincomeq)1	-0.402442	0.156161	-2.577
factor(paincomeq)2	-0.401644	0.156386	-2.568
factor(paincomeq)3	-0.410963	0.165258	-2.487
factor(paincomeq)4	-0.274153	0.178886	-1.533
paincomeqXage	0.022434	0.004958	4.525
factor(parent_edu2)1	0.151985	0.064109	2.371
factor(housing_tenure)social rent	-0.078708	0.086685	-0.908
factor(housing_tenure)private rent n other	0.216275	0.117673	1.838
parenthealth	-0.102420	0.030417	-3.367
parentage	-0.012507	0.005530	-2.262
factor(liveboth)1	-0.234701	0.119145	-1.970
age_centered	0.111682	0.012476	8.951
factor(sex)1	0.210578	0.059393	3.545
lnincome	0.067006	0.014339	4.673
factor(edu2)1	0.437977	0.067080	6.529
factor(family_status)coresident partner and child	0.735003	0.290754	2.528
factor(family_status)coresident partner	0.517030	0.179192	2.885
factor(family_status)coresident child	0.535401	0.190600	2.809
factor(racel2)non-Whites	-1.026362	0.088813	-11.556
factor(student)1	18.417807	152.783828	0.121
factor(unemployed)1	0.011589	0.109376	0.106
factor(urban)1	-0.183862	0.069373	-2.650
factor(houseprices)middle	0.236792	0.077460	3.057
factor(houseprices)high	0.010229	0.077154	0.133
factor(wave)wave 9	-0.175781	0.087754	-2.003
factor(wave)wave 10	-0.174380	0.087982	-1.982
factor(wave)wave 11	-0.236152	0.091840	-2.571
factor(wave)wave 12	-0.039302	0.089298	-0.440
	Pr(> z)		
(Intercept)	7.64e-16	***	
factor(paincomeq)1	0.009963	**	
factor(paincomeq)2	0.010220	*	
factor(paincomeq)3	0.012890	*	
factor(paincomeq)4	0.125384		

```

paincomeqXage                6.03e-06 ***
factor(parent_edu2)1         0.017753 *
factor(housing_tenure)social rent 0.363887
factor(housing_tenure)private rent n other 0.066072 .
parenthealth                 0.000759 ***
parentage                   0.023721 *
factor(liveboth)1           0.048852 *
age_centered                 < 2e-16 ***
factor(sex)1                 0.000392 ***
lnincome                    2.97e-06 ***
factor(edu2)1               6.61e-11 ***
factor(family_status)coresident partner and child 0.011474 *
factor(family_status)coresident partner 0.003910 **
factor(family_status)coresident child 0.004969 **
factor(race12)non-Whites    < 2e-16 ***
factor(student)1           0.904049
factor(unemployed)1        0.915618
factor(urban)1             0.008041 **
factor(houseprices)middle  0.002236 **
factor(houseprices)high    0.894526
factor(wave)wave 9         0.045166 *
factor(wave)wave 10        0.047479 *
factor(wave)wave 11        0.010131 *
factor(wave)wave 12        0.659849

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 10123 on 17966 degrees of freedom
Residual deviance: 8771 on 17938 degrees of freedom
(3505 observations deleted due to missingness)
AIC: 8829

Number of Fisher Scoring iterations: 14

```
print(summary(all_bf))
```

Call:

```
glm(formula = leave ~ factor(paincomeq) + paincomeqXage + factor(parent_edu2) +
```

```

factor(housing_tenure) + parenthealth + parentage + factor(liveboth) +
age_centered + lnincome + factor(edu2) + factor(family_status) +
factor(racel2) + factor(student) + factor(unemployed) + factor(urban) +
factor(houseprices) + factor(relative_num) + factor(wave),
family = binomial(link = "logit"), data = indrespbfy)

```

Coefficients:

	Estimate	Std. Error	z value
(Intercept)	-1.914644	0.410905	-4.660
factor(paincomeq)1	-0.532449	0.204472	-2.604
factor(paincomeq)2	-0.400107	0.202321	-1.978
factor(paincomeq)3	-0.524362	0.215743	-2.430
factor(paincomeq)4	-0.357552	0.233980	-1.528
paincomeqXage	0.019623	0.006757	2.904
factor(parent_edu2)1	0.189051	0.087185	2.168
factor(housing_tenure)social rent	-0.135790	0.117232	-1.158
factor(housing_tenure)private rent n other	0.196210	0.158309	1.239
parenthealth	-0.086504	0.041103	-2.105
parentage	-0.020240	0.007692	-2.631
factor(liveboth)1	-0.010996	0.149699	-0.073
age_centered	0.139613	0.017085	8.172
lnincome	0.056882	0.019623	2.899
factor(edu2)1	0.437052	0.091059	4.800
factor(family_status)coresident partner and child	1.153586	0.401672	2.872
factor(family_status)coresident partner	0.588224	0.235833	2.494
factor(family_status)coresident child	0.594785	0.197580	3.010
factor(racel2)non-Whites	-0.939593	0.121385	-7.741
factor(student)1	18.368027	177.697805	0.103
factor(unemployed)1	0.194617	0.154065	1.263
factor(urban)1	-0.230512	0.093359	-2.469
factor(houseprices)middle	0.110194	0.103210	1.068
factor(houseprices)high	-0.098233	0.103754	-0.947
factor(relative_num)large hh	0.018360	0.089281	0.206
factor(wave)wave 9	-0.043029	0.116494	-0.369
factor(wave)wave 10	-0.308088	0.123243	-2.500
factor(wave)wave 11	-0.285484	0.126067	-2.265
factor(wave)wave 12	0.022144	0.119082	0.186
	Pr(> z)		
(Intercept)	3.17e-06	***	
factor(paincomeq)1	0.00921	**	
factor(paincomeq)2	0.04798	*	
factor(paincomeq)3	0.01508	*	
factor(paincomeq)4	0.12648		


```

paincomeqXage                                0.00368 **
factor(parent_edu2)1                          0.03013 *
factor(housing_tenure)social rent             0.24674
factor(housing_tenure)private rent n other    0.21519
parenthealth                                  0.03533 *
parentage                                      0.00850 **
factor(liveboth)1                             0.94144
age_centered                                 3.04e-16 ***
lnincome                                       0.00375 **
factor(edu2)1                                 1.59e-06 ***
factor(family_status)coresident partner and child 0.00408 **
factor(family_status)coresident partner        0.01262 *
factor(family_status)coresident child          0.00261 **
factor(race12)non-Whites                      9.89e-15 ***
factor(student)1                             0.91767
factor(unemployed)1                           0.20651
factor(urban)1                                0.01355 *
factor(houseprices)middle                     0.28567
factor(houseprices)high                       0.34375
factor(relative_num)large hh                  0.83707
factor(wave)wave 9                           0.71186
factor(wave)wave 10                           0.01243 *
factor(wave)wave 11                           0.02354 *
factor(wave)wave 12                           0.85248

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 5590.2 on 9201 degrees of freedom
Residual deviance: 4727.3 on 9173 degrees of freedom
(1831 observations deleted due to missingness)
AIC: 4785.3

Number of Fisher Scoring iterations: 14

```
print(summary(all_bm))
```

Call:

```
glm(formula = leave ~ factor(paincomeq) + paincomeqXage + factor(parent_edu2) +
```

```

factor(housing_tenure) + parenthealth + parentage + factor(liveboth) +
age_centered + lnincome + factor(edu2) + factor(family_status) +
factor(racel2) + factor(student) + factor(unemployed) + factor(urban) +
factor(houseprices) + factor(relative_num) + factor(wave),
family = binomial(link = "logit"), data = indrespbmy)

```

Coefficients:

	Estimate	Std. Error	z value
(Intercept)	-2.486324	0.460937	-5.394
factor(paincomeq)1	-0.176593	0.250522	-0.705
factor(paincomeq)2	-0.314968	0.253015	-1.245
factor(paincomeq)3	-0.234535	0.265526	-0.883
factor(paincomeq)4	-0.119077	0.285112	-0.418
paincomeqXage	0.025759	0.007370	3.495
factor(parent_edu2)1	0.147720	0.095943	1.540
factor(housing_tenure)social rent	-0.018444	0.130174	-0.142
factor(housing_tenure)private rent n other	0.285033	0.177757	1.603
parenthealth	-0.123966	0.045540	-2.722
parentage	-0.006569	0.008545	-0.769
factor(liveboth)1	-0.590391	0.202333	-2.918
age_centered	0.080510	0.018533	4.344
lnincome	0.081096	0.021108	3.842
factor(edu2)1	0.418824	0.100647	4.161
factor(family_status)coresident partner and child	0.441990	0.453697	0.974
factor(family_status)coresident partner	0.445541	0.283465	1.572
factor(family_status)coresident child	-13.279130	364.914609	-0.036
factor(racel2)non-Whites	-1.090580	0.136672	-7.980
factor(student)1	18.645759	300.727126	0.062
factor(unemployed)1	-0.158447	0.156248	-1.014
factor(urban)1	-0.087908	0.105153	-0.836
factor(houseprices)middle	0.407995	0.119357	3.418
factor(houseprices)high	0.153327	0.117696	1.303
factor(relative_num)large hh	-0.183542	0.098733	-1.859
factor(wave)wave 9	-0.336046	0.135445	-2.481
factor(wave)wave 10	-0.007020	0.126459	-0.056
factor(wave)wave 11	-0.158245	0.134755	-1.174
factor(wave)wave 12	-0.096663	0.136490	-0.708
	Pr(> z)		
(Intercept)	6.89e-08 ***		
factor(paincomeq)1	0.480874		
factor(paincomeq)2	0.213184		
factor(paincomeq)3	0.377081		
factor(paincomeq)4	0.676203		

paincomeqXage	0.000474 ***
factor(parent_edu2)1	0.123642
factor(housing_tenure)social rent	0.887325
factor(housing_tenure)private rent n other	0.108824
parenthealth	0.006486 **
parentage	0.442071
factor(liveboth)1	0.003524 **
age_centered	1.40e-05 ***
lnincome	0.000122 ***
factor(edu2)1	3.16e-05 ***
factor(family_status)coresident partner and child	0.329959
factor(family_status)coresident partner	0.116004
factor(family_status)coresident child	0.970972
factor(race12)non-Whites	1.47e-15 ***
factor(student)1	0.950561
factor(unemployed)1	0.310546
factor(urban)1	0.403152
factor(houseprices)middle	0.000630 ***
factor(houseprices)high	0.192663
factor(relative_num)large hh	0.063031 .
factor(wave)wave 9	0.013099 *
factor(wave)wave 10	0.955729
factor(wave)wave 11	0.240269
factor(wave)wave 12	0.478821

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 4510.4 on 8764 degrees of freedom

Residual deviance: 3992.5 on 8736 degrees of freedom

(1674 observations deleted due to missingness)

AIC: 4050.5

Number of Fisher Scoring iterations: 14

Check assumptions

```
view(dfSummary(indrespb))
```

Switching method to 'browser'

Output file written: C:\Users\Think\AppData\Local\Temp\RtmpoRPwqW\file4dec18e9659d.html