Package 'psidR'

July 2, 2014

Type Package

Title Build panel data sets from PSID raw data

Version 1.2

Date 2013-09-23

Author Florian Oswald

Maintainer Florian Oswald <florian.oswald@gmail.com>

Description Makes it easy to build panel data in wide format from PSID
delivered raw data. Deals with data downloaded and pre-processed by Stata or SAS,or can optionally download directly from the PSID server using the SAScii package.
psidR takes care of merging data from each wave onto a cross-period index file,so that individuals can be followed over time. The user must specify which years
they are interested in, and the PSID variable names (e.g. ER21003) for each year
(they differ in each year). There are different panel data designs and two
popular subsetting criteria (heads only and core sample only) implemented.

URL https://github.com/floswald/psidR

Depends data.table, RCurl, foreign, SAScii

License GPL-3

Suggests survey

Collate 'build.panel.r' 'makeids.r' 'psidR-package.r'

NeedsCompilation no

Repository CRAN

Date/Publication 2013-09-10 23:56:58

R topics documented:

`build.panel` *build.panel: Build PSID panel data set*

Description

Builds a panel data set in wide format with id variables `personID` and `period` from individual PSID family files.

Usage

```
build.panel(datadir = NULL, fam.vars, ind.vars = NULL,
  SAScii = FALSE, heads.only = TRUE, core = TRUE,
  design = "balanced", verbose = FALSE)
```

Arguments

`datadir`	either `NULL`, in which case saves to tmpdir or path to directory containing family files ("FAMyyyy.xyz") and individual file ("IND2009ER.xyz") in admissible formats .xyz. Admissible are .dta, .csv, .RData, .rda. Please follow naming convention.
`fam.vars`	data.frame of variable to retrieve from family files. see example for required format.
`ind.vars`	data.frame of variables to get from individual file. In almost all cases this will be the type of survey weights you want to use. don't include id variables ER30001 and ER30002.
`SAScii`	logical TRUE if you want to directly download data into Rda format (no dependency on STATA/SAS/SPSS). may take a long time.
`heads.only`	logical TRUE if user wants household heads only. if FALSE, data contains a row with value of "relation to head" variable.
`core`	logical TRUE if user wants core sample only. if FALSE, data will oversample poverty sample.
`design`	either character "balanced" or "all" or integer. "Balanced" means only individuals who appear in each wave are considered. "All" means all are taken. An integer value stands for minimum consecutive years of participation, i.e. design=3 means present in at least 3 consecutive waves.
`verbose`	logical TRUE if you want verbose output.

Details

takes desired variables from family files for specified years in folder `datadir` and merges using the id information in `IND2011ER.xyz`, which must be in the same directory. Note that only one IND file may be present in the directory (each PSID shipping comes with a new IND file). There is an option to directly download the data from the PSID server to folder `datadir` or `tmpdir`. The user can change subsetting criteria as well as sample designs. Merge: the variables `interview number` in each family file map to the `interview number` variable of a given year in the individual file. Run `example(build.panel)` for a demonstration. Accepted input data are stata format .dta, .csv files or R data formats .rda and RData. Similar in usage to stata module `psiduse`.

Value

`data`	resulting `data.table`. the variable `pid` is the unique person identifier, constructed from ID1968 and pernum.
`dict`	data dictionary if stata data was supplied, NULL else

Examples

```
## Not run:
# specify variables from family files you want

myvars <- data.frame(year=c(2001,2003),
                     house.value=c("ER17044","ER21043"),
                     total.income=c("ER20456","ER24099"),
                     education=c("ER20457","ER24148"))

# specify variables from individual index file

indvars = data.frame(year=c(2001,2003),
                     longitud.wgt=c("ER33637","ER33740"))

# call builder
# mydir is a directory that contains FAM2001ER.dta,
# FAM2003ER.dta and IND2011ER.dta

# default
d <- build.panel(datadir=mydir,
                 fam.vars=myvars,
                 ind.vars=indvars)

# also non-heads
d <- build.panel(datadir=mydir,
                 fam.vars=myvars,
                 ind.vars=indvars,
                 heads.only=FALSE)

# non-balanced panel design
d <- build.panel(datadir=mydir,
                 fam.vars=myvars,
                 ind.vars=indvars,
                 heads.only=FALSE,
```

```
                design=2) # keep if stay 2+ periods

## End(Not run)

# ######################################
# reproducible example on artifical data.
# run this with example(build.panel).
# ######################################

## make reproducible family data sets for 2 years
## variables are: family income (Money) and age

# suppose there are N individuals in year 1 and year 2.
# zero attrition.

N <- 10

fam <- data.frame(int85 = 1:N,int86=sample(1:N),
                  Money85=rlnorm(n=N,10,1),
                  age85=sample(20:80,size=N,replace=TRUE))
fam$Money86 <- fam$Money85+rnorm(N,500,30)
fam$age86 <- fam$age85+1
fam

# separate into data.frames.
# you would download files like those two:
fam1985 <- subset(fam,select = c(int85,Money85,age85))
fam1986 <- subset(fam,select = c(int86,Money86,age86))

# assign correct PSID varname of "family interview 1985"
names(fam1985)[1] <- "V11102"
names(fam1986)[1] <- "V12502"


# construct an Individual index file: that would be IND2009ER
# needs to have a unique person number (ER30001)
# and an indicator for whether from core etc,
# as well as the interview number for each year
#
# for sake of illustration, suppose the PSID has a total
# of 2N people (i.e. N are neither in year1 nor year2,
# but in some other years)
IND2009ER <- data.frame(ER30001=sample((2*N):(4*N),size=2*N),
                        ER30002=sample(1:(2*N),size=2*N))

# if a person is observed, they have an interview number
# in both years. if not observed, it's zero.
# randomly allocate persons to ER30001.
tmp <- rbind(fam[,1:2],data.frame(int85=rep(0,N),int86=rep(0,N)))

IND2009ER <- cbind(IND2009ER,tmp[sample(1:(2*N)),])
names(IND2009ER)[3:4] <- c("ER30463","ER30498")
```

```
# also need relationship to head in each year in the index
# 50% prob of being head in year1
IND2009ER$ER30465 <- sample(c(10,20),prob=c(0.5,0.5),
                           size=2*N,replace=TRUE)
IND2009ER$ER30500 <- sample(c(10,20),prob=c(0.9,0.1),
                          size=2*N,replace=TRUE)
# and a survey weight
IND2009ER$ER30497 <- runif(20)
IND2009ER$ER30534 <- runif(20)
IND2009ER

# setup the ind.vars data.frame
indvars <- data.frame(year=c(1985,1986),ind.weight=c("ER30497","ER30534"))

# create a temporary datadir
my.dir <- tempdir()
# save those in the datadir
# notice different R formats admissible
save(fam1985,file=paste0(my.dir,"/FAM1985ER.rda"))
save(fam1986,file=paste0(my.dir,"/FAM1986ER.RData"))
save(IND2009ER,file=paste0(my.dir,"/IND2009ER.RData"))

# now famvars
famvars <- data.frame(year=c(1985,1986),
                      money=c("Money85","Money86"),
                      age=c("age85","age86"))

# call the builder
# need to set core==FALSE because person numbering indicates
# that all ids<2931 are not core.
# set heads to FALSE to have a clear count.
# data will contain column "relation.head" holding the relationship code.

d <- build.panel(datadir=my.dir,fam.vars=famvars,
                 ind.vars=indvars,core=FALSE,
                 heads=FALSE,verbose=TRUE)

# notice: all 2*N individuals are present
print(d$data[order(pid)],nrow=Inf) # check the age column

# see what happens if we drop non-heads
# only the ones who are heads in BOTH years
# are present (since design='balanced' by default)
d <- build.panel(datadir=my.dir,fam.vars=famvars,
                 ind.vars=indvars,core=FALSE,
                 heads=TRUE,verbose=FALSE)
print(d$data[order(pid)],nrow=Inf)

# change sample design to "all":
# we'll keep individuals if they are head in one year,
# and drop in the other
d <- build.panel(datadir=my.dir,fam.vars=famvars,
                 ind.vars=indvars,core=FALSE,heads=TRUE,
```

```
                verbose=FALSE,design="all")
print(d$data[order(pid)],nrow=Inf)

file.remove(paste0(my.dir,"/FAM1985ER.rda"),
            paste0(my.dir,"/FAM1986ER.RData"),
            paste0(my.dir,"/IND2009ER.RData"))

# END psidR example

# ######################################################################
# Please go to https://github.com/floswald/psidR for more example usage
# ######################################################################
```

`get.psid` *get.psid connects to PSID database and downloads into Rda*

Description

see http://www.asdfree.com/ for other usage and http://stackoverflow.com/questions/15853204/how-to-login-and-then-download-a-file-from-aspx-web-pages-with-r

Usage

```
get.psid(file, name, params, curl)
```

Arguments

`file`	string psid file number
`name`	string of filename on disc
`params`	postFormRCurl parameters
`curl`	postFormRCurl curl handle

Author(s)

Anthony Damico <ajdamico@gmail.com>

`make.char` *Convert factor to character*

Description

helper function to convert factor to character in a data.table

Usage

```
make.char(x)
```

Arguments

x	a factor

Value

a character

`makeids` *ID list for mergeing PSID*

Description

this list is taken from http://ideas.repec.org/c/boc/bocode/s457040.html

Usage

```
makeids()
```

Details

this function hardcodes the PSID variable names of "interview number" from both family and individual file for each wave, as well as "sequence number", "relation to head" and numeric value x of that variable such that "relation to head" == x means the individual is the head. Varies over time.

`psidR` *psidR*

Description

psidR is a package that helps the task of building longitudinal datasets from the Panel Study of Income Dynamics (PSID). The user must supply the PSID variable names that correspond to the variables of interest in each desired wave. The data may be in .dta, .csv format on disk. Creation of .dta or .csv datasets requires access to Stata or SAS software. There is an option to bypass this requirement by directly downloading the data from the server into a data.frame.

Index