Package: psidR (via r-universe)

August 27, 2024

Type Package

Title Build Panel Data Sets from PSID Raw Data

Version 2.2

Date 2024-05-29

Author Florian Oswald

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

Description Makes it easy to build panel data in wide format from Panel Survey of Income Dynamics ('PSID') delivered raw data. Downloads data 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). The package offers helper functions to retrieve variable names from different waves. There are different panel data designs and sample subsetting criteria implemented (``SRC'', ``SEO'', ``immigrant'' and ``latino'' samples).

Depends R (>= 3.5.0)

URL https://github.com/floswald/psidR, http://floswald.github.io/psidR/

Imports data.table, RCurl, foreign, SAScii, openxlsx, futile.logger

License GPL-3

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

Suggests testthat

RoxygenNote 7.2.3

Repository https://floswald.r-universe.dev

RemoteUrl https://github.com/floswald/psidr

RemoteRef HEAD

RemoteSha f31085871a873a3bf571594d5bc43cd714640d59

Contents

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

Description

Builds a panel data set with id variables `pid` (unique person identifier) and `year` from individual PSID family files and supplemental wealth files.

Usage

```
build.panel(
  datadir = NULL,
  fam.vars,
  ind.vars = NULL,
  heads.only = FALSE,
  current.heads.only = FALSE,
  sample = NULL,
  design = "balanced",
  loglevel = INFO
)
```

Arguments

`datadir`	either `NULL`, in which case saves to tmpdir or path to directory containing family files ("FAMyyyy.RData") and individual file ("IND2009ER.RData").
`fam.vars`	data.frame of variable to retrieve from family files. Can contain 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.
`heads.only`	logical TRUE if user wants household heads only. Household heads in sample year.
`current.heads.only`	logical TRUE if user wants current household heads only. Distinguishes mover outs heads.
`sample`	string indicating which sample to select: "SRC" (survey research center), "SEO" (survey for economic opportunity), "immigrant" (immigrant sample), "latino" (Latino family sample). Defaults to NULL, so no subsetting takes place.
`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.
`loglevel`	one of INFO, WARN and DEBUG. INFO by default.

Details

There are several supported approches. Approach one downloads stata data, uses stata to build each wave, then puts it together with 'psidR'. The second (recommended) approach downloads all data directly from the psid servers (no Stata needed). For this approach you need to supply the precise names of psid variables - those variable names vary by year. E.g. *total family income* will have different names in different waves. The function `getNamesPSID` greatly helps collecting names for all waves.

Value

resulting `data.table`. the variable `pid` is the unique person identifier, constructed from ID1968 and pernum

Merging

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.

Supplements

Notice that support for wealth supplements is disabled! Recent releases of the main family file have wealth data included. Earlier waves must be merged manually, again by variable `interview number` as above.

Examples

```
## Not run:
# ################################################
# Real-world example: not run because takes long.
# Build panel with income, wage, age and education
# optionally: add wealth supplements!
```

```
# ##############################################

# The package is installed with a list of variables
# Alternatively, search for names with \code{\link{getNamesPSID}}
# This is the body of function build.psid()
# (so why not call build.psid() and see what happens!)
r = system.file(package="psidR")
if (small){
  f = fread(file.path(r,"psid-lists","famvars-small.txt"))
  i = fread(file.path(r,"psid-lists","indvars-small.txt"))
} else {
  f = fread(file.path(r,"psid-lists","famvars.txt"))
  i = fread(file.path(r,"psid-lists","indvars.txt"))
}
setkey(i,"name")
setkey(f,"name")
i = dcast(i[,list(year,name,variable)],year~name)
f = dcast(f[,list(year,name,variable)],year~name)
  d = build.panel(datadir="~/datasets/psid/",fam.vars=f,
                  ind.vars=i,
                  heads.only =TRUE,sample="SRC",
                  design="all")
  save(d,file="~/psid.RData")

## 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

## Data acquisition step:
## run build.panel with sascii=TRUE

# testPSID creates artifical PSID data
td <- testPSID(N=12,N.attr=0)
fam1985 <- data.table::copy(td$famvars1985)
fam1986 <- data.table::copy(td$famvars1986)
IND2019ER <- data.table::copy(td$IND2019ER)

# 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(IND2019ER,file=paste0(my.dir,"/IND2019ER.RData"))

## end Data acquisition step.
```

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

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

# call the builder
# data will contain column "relation.head" holding the relationship code.

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

# 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,
                 heads.only=TRUE)
print(d[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,heads.only=TRUE,
                 design="all")
print(d[order(pid)],nrow=Inf)

file.remove(paste0(my.dir,"/FAM1985ER.rda"),
            paste0(my.dir,"/FAM1986ER.RData"),
            paste0(my.dir,"/IND2019ER.RData"))

# END psidR example

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

`build.psid` *Build example PSID*

Description

Builds a panel from the full PSID dataset

Usage

```
build.psid(datadr = "~/datasets/psid/", small = TRUE)
```

Arguments

datadr	string of the data directory
small	logical TRUE if only use years 2013 and 2015.

Value

a data.table with panel data

---

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

Description

see http://asdfree.com/ for other usage and https://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>

---

getNamesPSID *GetPSID variables names from various years*

Description

The user can specify one variable name from any year. This function will find that variable's correct name in any of the years specified by the user. If user does not specify the `years` variable, return will represent all years in which variable was present.

Usage

```
getNamesPSID(aname, cwf, years = NULL, file = NULL)
```

Arguments

aname	A variable name in any of the PSID years
cwf	A data.frame representation of the cross-walk file, (the psid.xlsx file).
years	A vector of years. If NULL, all years in which that variable existed are returned
file	optional file name to write csv

Details

This uses the psid.xlsx crosswalk file from UMich, which is available at http://psidonline.isr.umich.edu/help/xyr/psid.xlsx. In the example, the package openxlsx's read.xlsx is used to import the crosswalk file.

Ask for one variable at a time.

Value

A vector of names, one for each year.

Author(s)

Paul Johnson <pauljohn@ku.edu> and Florian Oswald

Examples

```
# read UMich crosswalk from installed file
r = system.file(package="psidR")
cwf = openxlsx::read.xlsx(file.path(r,"psid-lists","psid.xlsx"))

# or download directly
# cwf <- read.xlsx("http://psidonline.isr.umich.edu/help/xyr/psid.xlsx")

# then get names with
getNamesPSID("ER17013", cwf, years = 2001)
getNamesPSID("ER17013", cwf, years = 2003)
getNamesPSID("ER17013", cwf, years = NULL)
getNamesPSID("ER17013", cwf, years = c(2005, 2007, 2009))
```

---

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.

---

medium.test.ind *three year test, ind file*

Description

three year test, ind file

Usage

```
medium.test.ind(dd = NULL)
```

Arguments

dd	Data Dictionary location. If NULL, use temp dir and force download

medium.test.ind.NA *three year test, ind file and one NA variable*

Description

three year test, ind file and one NA variable

Usage

```
medium.test.ind.NA(dd = NULL)
```

Arguments

dd	Data Dictionary location. If NULL, use temp dir and force download

medium.test.ind.NA.wealth *three year test, ind file and one NA variable and wealth*

Description

three year test, ind file and one NA variable and wealth

Usage

```
medium.test.ind.NA.wealth(dd = NULL)
```

Arguments

dd	Data Dictionary location. If NULL, use temp dir and force download

medium.test.noind *three year test, no ind file*

Description

three year test, no ind file

Usage

```
medium.test.noind(dd = NULL)
```

Arguments

dd	Data Dictionary location

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. Data can be supplied via Stata, or directly downloaded from PSID servers without any need for STATA. data.frame.

---

small.test.ind *one year test, ind file*

Description

one year test, ind file

Usage

```
small.test.ind(dd = NULL)
```

Arguments

dd	Data Dictionary location. If NULL, use temp dir and force download

---

small.test.noind *one year test, no ind file*

Description

one year test, no ind file

Usage

```
small.test.noind(dd = NULL)
```

Arguments

dd	Data Dictionary location. If NULL, use temp dir and force download

testPSID *Create a test PSID dataset*

Description

makes artifical PSID data with variables age and income for two consecutive years 1985 and 1986.

Usage

```
testPSID(N = 100, N.attr = 0)
```

Arguments

N	number of people in each wave
N.attr	number of people lost to attrition

Value

list with (fake) individual index file IND2009ER and family files for 1985 and 1986

Index