Package 'seqSHP'

May 25, 2024

Type Package

Title Building Sequences from SHP Waves

Version 0.1.1

Maintainer Gilbert Ritschard <gilbert.ritschard@unige.ch>

Description Based on the structure of the SPSS version of the Swiss
Household Panel (SHP) data, provides a function seqFromWaves()
that seeks the data of variables specified by the user in each
of the wave files and collects them as sequences. The function also
matches the sequences with variables from other files such as the
master files of persons (MP) and households (MH), and social origins (SO).
It can also match with activity calendar data (CA).

Imports haven, dplyr

Suggests TraMineR

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Gilbert Ritschard [aut, cre, cph]
(<https://orcid.org/0000-0001-7776-0903>)

Repository CRAN

Date/Publication 2024-05-25 17:30:02 UTC

R topics documented:

---

getColumnIndex	*Column indexes of protoname in the table returned by extractSeqFromWaves*

Description

Column indexes in `data` of the variable fitting the provided protoname.

Usage

```
getColumnIndex(data, protoname)
```

Arguments

data	Tibble or data.frame. Table returned by seqFromWaves.
protoname	String. Protoname of a variable.

Details

Returns the column indexes in `data` of the variable fitting the provided protoname. Examples in help page of seqFromWaves.

Value

Vector of indexes of the columns of `data` corresponding to the protoname.

See Also

seqFromWaves

---

seqFromWaves	*Extracting sequences from SHP waves*

Description

Based on the structure of the 'SPSS' version of the Swiss Household Panel (SHP) data, the function seeks the data of variables specified by the user in each of the wave files and collects them as sequence data in a table. The function can also match the sequences with variables from other files such as the master files of persons (MP) and households (MH) and social origins (SO). It can also match with activity calendar data (CA). In addition it can extract user specified covariates from a specific wave.

Usage

```
seqFromWaves(
  wavedir = NULL,
  datadir = NULL,
  shpdir = NULL,
  pvarseq = NULL,
  hvarseq = NULL,
  MPvar = c("SEX", "BIRTHY"),
  SOvar = NULL,
  LJvar = NULL,
  CAvar = NULL,
  PLWvar = NULL,
  HLWvar = NULL,
  waves = NULL,
  covw = max(waves),
  maxMissing = length(waves) - 1,
  maxMissingCA = length(CAvar) - 1
)
```

Arguments

wavedir	String. Path to the SPSS SHP wave data. If NULL, wavedir is built from shpdir.
datadir	String. Path to the SPSS WA (All Waves) data. If NULL, datadir is built from shpdir.
shpdir	String. Root path of the SHP data. The path should end with the two-digits number of the last wave, e.g., "C:/shp23".
pvarseq	Vector of strings. Protoname(s) ($$ for year) of the wanted sequence(s) of personal data.
hvarseq	Vector of strings. Protoname(s) ($$ for year) of the wanted sequence(s) of household data.
MPvar	Vector of strings. Variables to be extracted from the person master (MP) file.
SOvar	Vector of strings. Variables to be extracted from the social origin (SO) file.
LJvar	Vector of strings. Variables to be extracted from the last job (LJ) file.
CAvar	Vector of strings. Variables to be extracted from the activity calendar (CA) file.
PLWvar	Vector of strings. Variables to be extracted from the covw wave personal file.
HLWvar	Vector of strings. Variables to be extracted from the covw wave household file.
waves	Vector of integers. Selected waves (waves id number, not year!)
covw	Integer. Id number of wave from which to extract PLWvar and HLWvar covariates.
maxMissing	Integer. Maximum allowed missing states in yearly sequences (pvarseq and hvarseq).
maxMissingCA	Integer. Maximum allowed missing states in monthly sequences (CAvar).

Details

SHP data are available for free from FORS (https://forscenter.ch/projects/swiss-household-panel/data/) but require the user to accept the usage contract.

The function extracts the columns corresponding to the protonames provided from the successive wave files and collects them in a tibble table. From this table, it is then, for example, straightforward to create state sequence objects for 'TraMineR'.

When using the shpdir argument, the shpdir path must end with the two-digits number xx of the last wave. The path wavedir is then set as shpdir/SHP-Data-W1-Wxx-SPSS/ and datadir as shpdir/SHP-Data-WA-SPSS/. For example, with shpdir="C:/SHP/shp23", wavedir will be set as "C:/SHP/shp23/SHP-Data-W1-W23-SPSS/" and datadir as "C:/SHP/shp23/SHP-Data-WA-SPSS/".

The list of variable names pvarseq and hvarseq must be provided as protonames with $$ standing for the last two digits of the year.

maxMissing is set by default as length(waves) - 1, which drops cases for which one of the yearly sequence defined by pvarseq and hvarseq is empty (i.e., has no valid state). Likewise, maxMissingCA is set by default as length(CAvar) - 1 to exclude cases with an empty monthly activity calendar sequence.

The package is based on a function written in 2012 by Matthias Studer.

Value

A tibble with the selected sequence data and covariates.

Author(s)

Gilbert Ritschard

References

Swiss Household Panel documentation at https://forscenter.ch/projects/swiss-household-panel/

See Also

getColumnIndex

Examples

```
## Setting paths to SHP data files. Adapt to your local folders!
## It should be something like
## wavedir <- "C:/SwissHPanel/shp23/SHP-Data-W1-W23-SPSS/"
## datadir <- "C:/SwissHPanel/shp23/SHP-Data-WA-SPSS/"

## Consider first the example of 3 waves and a MP file
##  shipping with the package
wavedir <- paste0(system.file(package="seqSHP"),"/extdata/")
datadir <- wavedir

####### Working status
```

```
first.w <- 1
last.w  <- 3
waves <- first.w:last.w
maxMissing <- 2

## Sequence of categorical variables
##  WSTAT$$ is working status (WS)
shp <- seqFromWaves(wavedir, datadir,
                 pvarseq="WSTAT$$",
                 waves=waves, maxMissing=maxMissing)

## Retrieve WS labels
attr(shp$WSTAT00,"labels")

## Creating WS sequence object
library(TraMineR)
ws.shortlab <- c("AO","UN","NL")
ws.longlab <- c("Active Occupied","Unemployed","Not in Labor Force")
ws.alph <- c(1,2,3)
xtlab <- (1998+first.w):(1998+last.w)

wsvar <- getColumnIndex(shp, "WSTAT$$")
ws.seq <- seqdef(shp[, wsvar], right=NA,
                 alphabet=ws.alph, states=ws.shortlab, labels=ws.longlab,
                 cnames=xtlab)

## plotting first 100 sequences
seqIplot(ws.seq[1:100,], sort="from.start")




## Not run:
#####################################################
## To run the full examples below, you must first install SHP data
## in an accessible folder
##
## Adapt to your local folders!
wavedir <- "C:/SwissHPanel/shp23/SHP-Data-W1-W23-SPSS/"
datadir <- "C:/SwissHPanel/shp23/SHP-Data-WA-SPSS/"

####### Working status

first.w <- 2
last.w  <- 23
waves <- first.w:last.w
maxMissing <- 10

## Sequence of categorical variables
##  WSTAT$$ is working status (WS) and
##  P$$C44 satisfaction with life
shp <- seqFromWaves(wavedir, datadir,
                 pvarseq=c("WSTAT$$","P$$C44"),
                 waves=waves, maxMissing=maxMissing)
```

```
## Retrieve WS labels
attr(shp$WSTAT00,"labels")

## Creating WS sequence object
library(TraMineR)
ws.shortlab <- c("AO","UN","NL")
ws.longlab <- c("Active Occupied","Unemployed","Not in Labor Force")
ws.alph <- c(1,2,3)
xtlab <- (1998+first.w):(1998+last.w)

wsvar <- getColumnIndex(shp, "WSTAT$$")
ws.seq <- seqdef(shp[, wsvar], right=NA,
                 alphabet=ws.alph, states=ws.shortlab, labels=ws.longlab,
                 cnames=xtlab, xtstep=2, tick.last=TRUE)

seqIplot(ws.seq, sort="from.start")


######### Activity calendar from sep99 to dec2021

month.short.names <- tolower(sapply(month.name, substr, 1, 3))
xtlab.ca <- c("sep99","oct99","nov99","dec99")
for (t in 00:21) {
 xtlab.ca <- c(xtlab.ca,paste0(month.short.names, formatC(t,width=2,flag=0)))
}
names(xtlab.ca) <- xtlab.ca
ca.var <- toupper(xtlab.ca) ## SPSS variable names are uppercase

CAseqdata <- seqFromWaves(wavedir, datadir, CAvar=ca.var, maxMissingCA=36)

attr(CAseqdata$SEP99, "labels")
## First 3 columns are IDPERS, SEX, and BIRTHY. Sequences from the other columns
seqCA <- seqdef(CAseqdata[,-(1:3)], cnames=xtlab.ca, right=NA, xtstep=6, tick.last=TRUE)
seqdplot(seqCA, border=NA, with.missing=TRUE)


## End(Not run)
```

Index