Package: alspac (via r-universe)

October 4, 2024

Title Data dictionary for ALSPAC

Version 0.48.1

Description Functions to search and extract variables based on keywords from the ALSPAC data dictionary.

License Artistic-2.0

Depends R (>= 3.5.0)

Imports dplyr, haven, magrittr, parallel, plyr, readstata13, stringr, tibble

Encoding UTF-8

LazyData no

RoxygenNote 7.3.2

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

RemoteUrl https://github.com/explodecomputer/alspac

RemoteRef HEAD

RemoteSha a72a6fd2d1609d0def3ec5423e901b9c2ad8795a

Contents

`addSourcesToDictionary` *Add data sources information to the dictionary from the data/sources.csv file. See generateSourcesSpreadsheet() for details about creating this file. This information is used when decide which data values to remove for participants who have withdrawn consent.*

Description

Add data sources information to the dictionary from the data/sources.csv file. See generateSourcesSpreadsheet() for details about creating this file. This information is used when decide which data values to remove for participants who have withdrawn consent.

Usage

```
addSourcesToDictionary(dictionary)
```

Arguments

`dictionary`	The name of an existing dictionary or the dictionary itself.

`createDictionary` *Create a dictionary from ALSPAC Stata files*

Description

Create a dictionary from ALSPAC Stata files

Usage

```
createDictionary(datadir = "Current", name = NULL, quick = FALSE)
```

Arguments

`datadir`	ALSPAC data subdirectory from which to create the index (Default: "Current"). .
`name`	If not `NULL`, then the resulting dictionary will be saved to a file in the R package for use next time the package is loaded. The dictionary will be available with the given name (Default: `NULL`).
`quick`	Logical. Default `FALSE`. The function uses multiple processors using `mclapply()`. Use multiple processors by setting `mc.cores` option using `options()`.

Value

Data frame dictionary listing available variables.

`current` *Current data*

Description

A list of current datasets.

Usage

```
current
```

Format

'current' A data frame with 87,248 rows and 19 columns:

obj Stata dataset name

name variable name

lab

type

counts

cat1

cat2

cat3

cat4

path

cat5

mother

mother_clinic

mother_quest

partner_quest

partner_clinic

partner

child_based

child_completed

dictionaryGood *Checks a dictionary*

Description

Checks if all the files referred to in the dictionary are accessible given the ALSPAC data directory.

Usage

```
dictionaryGood(dictionary, max.print = 10)
```

Arguments

dictionary	The name of an existing dictionary or the dictionary itself.
max.print	The maximum number of missing files to list if any are missing (Default: 10).

Value

TRUE if all files exist, otherwise FALSE and a warning listing at most max.print missing files.

extractDataset *Extract a dataset for external ALSPAC users*

Description

Extract a dataset for external ALSPAC users

Usage

```
extractDataset(
  variable_file,
  cid_file,
  b_number = "BXXXX",
  author = "Author",
  output_format = "sav",
  output_path = ".",
  output_file = file.path(output_path, paste0(author, "_", b_number, "_",
    format(Sys.time(), "%d%b%y"), ".", output_format)),
  dictionary = "current"
)
```

Arguments

variable_file	CSV file with column "Name" containing ALSPAC variable names.
cid_file	CSV file with two columns named "ALN" and the last letter of the filename (e.g. for "ACEHDBFG.txt" the column would be named "G").
b_number	B number of the project.
author	Last name of the project author.
output_format	"sav","csv" or "dta" (Default: "sav").
output_path	File path of output file, default is the current directory (Default: ".").
output_file	Dataset file (should not already exist). Default is derived from function arguments as follows: <output_path>/<author>_<b_number>_<date>.<output_format>.
dictionary	ALSPAC dictionary to use "current" (Default: "current").

Value

Saves the output dataset to ‘output_file‘ and returns it.

Examples

```
## Not run:
library(alspac)
setDataDir("R:/Data")
dat <- extractDataset(
         variable_file="ACEHDBFG.txt",
         cid_file="Vars_from_Explore.csv",
         output_format="sav",
         b_number="B0001",
         author="Smith")
## creates a data file with a name like "Smith_B0001_12Jul21.sav"
## in the current directory

## End(Not run)
```

---

extractVars *Extract variables from data*

Description

Take the output from ‘findVars‘ as a list of variables to extract from ALSPAC data

Usage

```
extractVars(
  x,
  exclude_withdrawn = TRUE,
  core_only = TRUE,
  adult_only = FALSE,
```

```
  spss = FALSE,
  haven = FALSE
)
```

Arguments

x	Output from ‘findVars‘
exclude_withdrawn	Whether to automatically exclude withdrawn consent IDs. Default is TRUE. This is conservative, removing all withdrawn consant ALNs from all datasets. Only use FALSE here if you have a more specific list of withdrawn consent IDs for your specific variables.
core_only	Whether to automatically exclude data from participants not in the core ALSPAC dataset (Default: TRUE). This should give the same samples as the Stata/SPSS scripts in the R:/Data/Syntax folder.
adult_only	No longer supported. Parent-specific restrictions are applied automatically when child-based or child-completed variables are not requested.
spss	Logical. Default FALSE.
haven	Logical. Default FALSE.

Details

There are about 130 ALSPAC data files. Given output from ‘findVars‘, this function will retrieve all the variables from these files and collapse them into a single data frame. It will return columns for all the variables, plus columns for ‘aln‘, ‘qlet‘ and ‘mult_mum‘ or ‘mult_dad‘ if they were present in any of the files.

Suppose we extract a four variables, one for each of mothers, children, fathers and partners. This will return the variables requested, along with some other columns -

- ‘aln‘ - This is the pregnancy identifier. NOTE - this is **not** an individual identifier. For example, notice that row 4 has entries for the father variable ‘ff1a005a‘, the mother variable ‘fm1a010a‘, and the partner variable ‘pc013‘.

- ‘qlet‘ - This is the child ID for the specific pregnancy. It will take values from A-D. **All** children will have a qlet, and **only** children will have a qlet. Therefore **if qlet is not NA, that row represents an individual child**.

- ‘alnqlet‘ - this is the ALN + QLET. If the individual is a child (e.g. row 8) then they will have a different ‘alnqlet‘ compared to the ‘aln‘. Otherwise, the ‘aln‘ is the same as the ‘alnqlet‘

- ‘mult_mum‘ and ‘mult_dad‘ - Sometimes the same mother (or father) had more than one pregnancy in the 18 month recruitment period. Those individuals have two ALNs. If either of these columns is "Yes" then that means you can drop them from the results if you want to avoid individuals being duplicated. This is the guidance from the FOM2 documentation:

1.7 Important Note for all data users: Please be aware that some women may appear in the release file more than once. This is due to the way in which women were originally enrolled into the study and were assigned IDs. ALSPAC started by enrolling pregnant women and the main study ID is a pregnancy based ID. Therefore if a women enrolled with two different pregnancies (both having an expected delivery date within the recruitment period [April 1991-December 1992]), she will have two separate IDs to uniquely identify these women and their pregnancies. An indicator variable has

been included in the file, called mult_mum to identify these women. If you are carrying out mother based research that does not require you to consider repeat pregnancies for which we have data then please select mult_mum == 'No' to remove the duplicate entries. This will keep one pregnancy and randomly drop the other pregnancy. If you are matching the data included in this file to child based data or have been provided with a dataset that includes the children of the ALSPAC pregnancies, as well as the mother-based data, you need not do anything as each pregnancy (and hence each child from a separate pregnancy) has a unique identifier and a mothers data has been included/repeated here for each of her pregnancies where appropriate.

The speed at which this function runs is dependent upon how fast your connection is to the R drive and how many variables you are extracting at once.

Value

A data frame with all the variable specified in 'x'. If `exclude_withdrawn` was `TRUE`, then columns named `woc_*` indicate which samples were excluded.

Examples

```
## Not run:
# Find all variables with BMI in the description
bmi_variables <- findVars("bmi", ignore.case=TRUE)
# Extract all the variables into a data.frame:
bmi <- extractVars(bmi_variables)
# Alternatively just extract the variables for adults
bmi <- extractVars(subset(bmi_variables, cat3 %in% c("Mother", "Adult")))

## End(Not run)
```

`extractWebOutput` *Extract variables exported from the ALSPAC variable lookup web app*

Description

The variable lookup webapp allows you to browse the available variables and export a list of selected variables. This function will read that exported list and extract the individual level data for each of the selected variables.

Usage

```
extractWebOutput(filename)
```

Arguments

`filename`	Name of file exported from ALSPAC variable lookup web app

Details

More generally, this function requires a file that has at least one column with the header 'Variable' followed by a list of variable names.

The R: drive must be mounted and its path set with the `setDataDir` function.

Value

Data frame

`filterVars` *Filter duplicate variables from findVars*

Description

Filter duplicate variables from findVars

Usage

```
filterVars(x, ...)
```

Arguments

`x`	Output from `findVars()`.
`...`	Filter terms. The name corresponds to the variable name for which to remove duplicates. Each term is a named vector whose names correspond to columns in 'x'. The values provide patterns for the given column to match.

Details

`findVars()` may identify multiple variables with the same name. This function can be used to select among these duplicates.

Value

The subset of 'x' that satisfies the supplied filters or that were not provided a filter.

Examples

```
## Not run:
varnames <- c("kz021","kz011b","ype9670", "c645a")
vars <- findVars(varnames)
vars <- subset(vars, subset=tolower(name) %in% varnames)
vars <- filterVars(vars, kz021=c(obj="^kz"),
                  kz011b=c(obj="^cp", lab="Participant"),
                  c645a=c(cat2="Quest"))

## End(Not run)
```

`findVars` *Find variables*

Description

Provide a list of search terms to find variables in the data dictionary.

Usage

```
findVars(
  ...,
  logic = "any",
  ignore.case = TRUE,
  perl = FALSE,
  fixed = FALSE,
  whole.word = FALSE,
  dictionary = "current"
)
```

Arguments

`...`	Search terms
`logic`	Conditions for the search strings, can be "all", "any", or "none". Set to "any" by default.
`ignore.case`	Should search terms be case sensitive? Defaults to TRUE.
`perl`	logical. Should perl-compatible regexps be used? Defaults to FALSE.
`fixed`	logical. If 'TRUE', 'pattern' is a string to be matched as is. Overrides all conflicting arguments. Defaults to FALSE.
`whole.word`	If 'TRUE' search term "word" will be changed to "\bword\b" to only match whole words. Defaults to FALSE.
`dictionary`	Data frame or name of a data dictionary. Dictionary available by default is "current" (from the R:/Data/Current folder). New dictionaries can be created using the `createDictionary()` function. (Default: "current").

Details

The resulting data frame will have the following columns:

- obj - The name of the data file
- name - The name of the variable
- type - The type of data for the variable
- lab - A description of the label
- code - The ALSPAC dictionary code for the variable
- counts - The number of non-NA values in the variable
- cat1-4 - These columns correspond to the folder names that the objects were found in

Value

A data frame containing a list of the variables, the files they originate from, and some descripton about the files

Examples

```
## Not run:
# Find variables with BMI or height in the description (this will return a lot of results!)
bmi_variables <- findVars("bmi", "height", logic="any", ignore.case=TRUE)

## End(Not run)
```

`generateSourcesSpreadsheet` *This function was used to initially create the data/sources.csv spreadsheet which provides the source of data for each data file ('obj') in the dictionary. This information is then used to determine which bits of data to remove to satisfy exclusion lists. Sources include mother, mother_clinic, mother_quest, partner, partner_clinic, partner_quest, child_based and child_completed. Sources can for the most part be determined automatically from the 'path' information provided for each variable in the dictionary.*

Description

sources <- generateSourcesSpreadsheet() utils::write.csv(sources, file="data/sources.csv", row.names=FALSE)

Usage

```
generateSourcesSpreadsheet()
```

`getDefaultDataDir` *Guess the default data directory*

Description

The R drive will be mounted in different paths for different systems. This function guesses the path to be used as default in setDataDir

Usage

```
getDefaultDataDir()
```

readExclusions *Get list of ALNs to exclude*

Description

The exclusion lists for mothers and children are stored in .do files in the R: drive. This function reads all of these .do files and then parses out the ALNS for withdrawn consent.

Usage

```
readExclusions()
```

Value

List of ALNs for each .do file.

removeExclusions *Remove data for participants who have withdrawn consent.*

Description

The exclusion lists for mothers and children are stored in .do files in the R: drive. This function obtains ALNs to exclude from these files and then sets the variable values to missing for the appropriate participants and adds indicator variables for these participants ("woc_*").

Usage

```
removeExclusions(x)
```

Arguments

x	Data frame output from `extractVars()`.

Value

The input data frame but with appropriate values set to missing with additional variables ("woc_*") identifying participants who have withdrawn consent.

setDataDir *Set the data directory*

Description

This function is automatically called upon loading the package through 'library(alspac)' It creates a global option called 'alspac_data_dir' which is used by the `extractVars` function to locate the alspac data files. This function guesses the path to be used as default in setDataDir. The defaults are:

- Windows: R:/Data/
- Mac: /Volumes/data/
- Linux: ~/.gvfs/data/

Usage

```
setDataDir(datadir = getDefaultDataDir())
```

Arguments

`datadir`	The directory where the ALSPAC data can be found

Value

Null. Assigns the option alspac_data_dir

Examples

```
## Not run:
setDataDir() # This sets the path based on the operating system's default
setDataDir("/some/other/path/") # This is how to supply a path manually

## End(Not run)
```

updateDictionaries *Update dictionaries*

Description

Update the variable dictionaries for the ALSPAC dataset.

Usage

```
updateDictionaries()
```

Index