Package 'HMDHFDplus'

February 3, 2023

Type Package

Title Read Human Mortality Database and Human Fertility Database Data from the Web

Version 2.0.1

Date 2023-02-03

Description Utilities for reading data from the Human Mortality Database (<https://www.mortality.org>), Human Fertility Database (<https://www.humanfertility.org>), and similar databases from the web or locally into an R session as data.frame objects. These are the two most widely used sources of demographic data to study basic demographic change, trends, and develop new demographic methods. Other supported databases at this time include the Human Fertility Collection (<https://www.fertilitydata.org>), The Japanese Mortality Database (<https://www.ipss.go.jp/p-toukei/JMD/index-en.html>), and the Canadian Human Mortality Database (<http://www.bdlc.umontreal.ca/chmd/>). Arguments and data are standardized.

URL https://github.com/timriffe/TR1

BugReports https://github.com/timriffe/TR1/issues

License GPL-2

LazyLoad yes

Encoding UTF-8

Depends R (>= 4.1),

Collate 'age2int.R' 'HFDutils.R' 'HMDutils.R' 'readHFD.R' 'readHMD.R' 'userInput.R'

Imports httr, rvest, dplyr, janitor, lubridate, readr, stringr, tidyr, rlang

Suggests RCurl

RoxygenNote 7.2.3

Language 'en-US'

NeedsCompilation no

Author Tim Riffe [aut, cre],
Carl Boe [aut],
Jason Hilton [aut],
Josh Goldstein [ctb],
Stephen Holzman [ctb]

Maintainer Tim Riffe <tim.riffe@gmail.com>

Repository CRAN

Date/Publication 2023-02-03 09:40:13 UTC

R topics documented:

age2int 2
getCHMDprovinces 3
getHFCcountries 4
getHFDcountries 4
getHFDdate 5
getHFDitemavail 5
getHMDcountries 6
getHMDitemavail 6
getJMDprefectures 7
HFCparse 7
HFDparse 8
HMDparse 9
readCHMDweb 9
readHFCweb 10
readHFD 12
readHFDweb 12
readHMD 14
readHMDweb 15
readJMDweb 16
userInput 17

Index **19**

`age2int` `age2int()` *convert the Age column from standard HMD or HFD tables to integer*

Description

Long the bane of many an HMD/HFD user is that the age column must be read into R as a factor or character vector, yet we'd like to use it as integer or numeric. This function strips symbols that are used to indicate the open age groups ("12-","55+","110+"), and coerces to integer format. This function is called by `HFDparse()` and `HMDparse()`, and so forth.

Usage

```
age2int(Age)
```

Arguments

Age	a vector of the Age column from and HMD or HFD data object that has been read directly into R. This may be a factor or character vector.

Details

This function is written for the sake of various parse functions.

Value

the same age vector as a clean integer.

Note

original function submitted by Josh Goldstein, modified by Tim Riffe.

Examples

```
AgeTest <- c("12-","13","14","55+")
(AgeNew  <- age2int(AgeTest))
AgeNew + .5 # sort of mid-interval

# also handles abrdiged ages properly:
AgeAbridged <- c("0","1-4","5-9","10-14")
age2int(AgeAbridged)
```

---

getCHMDprovinces *get a named vector of CHMD province codes*

Description

This is a helper function to get a vector of 3-character province codes.

Usage

```
getCHMDprovinces()
```

Value

a character vector of 3 character province codes.

Examples

```
## Not run:  (provs <- getCHMDprovinces())
```

getHFCcountries *getHFCcountries a function to grab all present country codes used in the Human Fertility Collection*

Description

The function returns a list of population codes used in the Human Fertility Collection (HFC). Optionally, it also can return a data.frame with both the full population name and short code.

Usage

```
getHFCcountries(names = FALSE)
```

Arguments

names	logical. Default FALSE Should a data.frame matching full country names to short codes be given?

Value

either a character vector of short codes (default) or a data.frame of country names and codes.

Examples

```
## Not run:
getHFCcountries()
getHFCcountries(names = TRUE)

## End(Not run)
```

getHFDcountries *internal function for grabbing the HFD country short codes.*

Description

This function is called by readHFDweb() and is separated here for modularity. We include both main and provisional countries in the grab.

Usage

```
getHFDcountries()
```

Value

a ‘tibble‘ with three columns ‘Country‘, ‘link‘ and ‘CNTRY‘ (the country short code)

`getHFDdate` *internal function for grabbing the date of last update for a given HFD country*

Description

called by `readHFDweb()`. This assumes that `CNTRY` is actually available in the HFD.

Usage

```
getHFDdate(CNTRY)
```

Arguments

`CNTRY`	HFD country short code.

Value

character string of eight integers representing the date as `"yyyymmdd"`.

`getHFDitemavail` *List the available data item names for a given HFD country.*

Description

called by `readHFDweb()`. This assumes that `CNTRY` is actually available in the HFD.

Usage

```
getHFDitemavail(CNTRY)
```

Arguments

`CNTRY`	HFD country short code.

Value

a tibble of all available data files for the selected country. There are several useful identifiers that can help determine the appropriate file, including the ‘measure‘ and ‘subtype‘ as detected from the html table properties, and ‘lexis‘ and ‘parity‘ as detected either from the file names or the table properties.

getHMDcountries *internal function for grabbing the HMD country short codes.*

Description

This function is called by `readHMDweb()` and is separated here for modularity. Assumes you have an internet connection.

Usage

```
getHMDcountries()
```

Value

a vector of HMD country short codes.

getHMDitemavail *internal function for grabbing the available data item names for a given country.*

Description

called by `readHMDweb()` to find file urls. This assumes that `CNTRY` is actually available in the HFD.

Usage

```
getHMDitemavail(CNTRY)
```

Arguments

`CNTRY`	character. HMD country short code.

Value

a tibble of all available data items for the selected country. There are several useful identifiers that can help determine the appropriate file, including the ‘measure‘, ‘lexis‘, ‘sex‘ and interval information, as detected from the item names.

getJMDprefectures *get a named vector of JMD prefecture codes*

Description

This is a helper function for those familiar with prefecture names but not with prefecture codes (and vice versa). It is also useful for looped downloading of data.

Usage

```
getJMDprefectures()
```

Value

a character vector of 2-digit prefecture codes. Names correspond to the proper names given in the English version of the HMD webpage.

Examples

```
## Not run:  (prefectures <- getJMDprefectures())
```

HFCparse *HFCparse internal function for modifying freshly read HCD data in its standard form*

Description

called by `readHFC()` and `readHFCweb()`. We assume there are no factors in the given data.frame and that it has been read in from the raw text files using something like: `read.csv(file = filepath, stringsAsFactors = FALSE, na.strings = ".", strip.white = TRUE)`. This function is visible to users, but is not likely needed directly.

Usage

```
HFCparse(DF)
```

Arguments

`DF`	a data.frame of HFC data, freshly read in.

Details

This parse routine is based on the subjective opinions of the author...

Value

DF same data.frame, modified so that columns are of a useful class. If there were open age categories, such as "-" or "+", this information is stored in a new dummy column called OpenInterval. Values of 99 or -99 in the AgeInterval column are replaced with "+" and "-", respectively. Year taken from Year1, and YearInterval is given, rather than Year2. Users wishing for a central time point should bear this is mind. The column Country is renamed CNTRY. Otherwise, columns in this database are kept in the data.frame, in case they may be useful.

HFDparse *internal function for modifying freshly read HFD data in its standard form*

Description

called by readHFD() and readHFDweb(). We assume there are no factors in the given data.frame and that it has been read in from the raw text files using something like: read.table(file = filepath, header = TRUE, skip = 2, na.strings = ".", as.is = TRUE). This function is visible to users, but is not likely needed directly.

Usage

```
HFDparse(DF)
```

Arguments

DF	a data.frame of HFD data, freshly read in.

Details

This parse routine is based on the subjective opinions of the author...

Value

DF same data.frame, modified so that columns are of a useful class. If there were open age categories, such as "-" or "+", this information is stored in a new dummy column called OpenInterval.

HMDparse — *internal function for modifying freshly read HMD data in its standard form*

Description

called by `readHMD()` and `readHMDweb()`. We assume there are no factors in the given data.frame and that it has been read in from the raw text files using something like: `read.table(file = filepath, header = TRUE, skip = 2, na.strings = ".", as.is = TRUE)`. This function is visible to users, but is not likely needed directly.

Usage

```
HMDparse(DF, filepath)
```

Arguments

`DF`	a data.frame of HMD data, freshly read in.
`filepath`	just to check if these are population counts from the name.

Details

This parse routine is based on the subjective opinions of the author...

Value

DF same data.frame, modified so that columns are of a useful class. If there were open age categories, such as `"-"` or `"+"`, this information is stored in a new dummy column called `OpenInterval`.

readCHMDweb — *read data from the Canadian Human Mortality Database into R*

Description

CHMD data are formatted exactly as HMD data. This function simply parses the necessary url together given a province code and data item (same nomenclature as HMD). Data is parsed using `HMDparse()`, which converts columns into useful and intuitive classes, for ready-use. See `?HMDparse` for more information on type conversions. No authentication is required for this database. Only a single item/prefecture is downloaded. Loop for more complex calls (See examples). The provID is not appended as a column, so be mindful of this if appending several items together into a single `data.frame`. Note that at the time of this writing, the finest Lexis resolution for prefectural lifetables is 5x5 (5-year, 5-year age groups). Raw data are, however, provided in 1x1 format, and deaths are also available in triangles. Note that cohort data are not produced for Canada at this time (but you could produce such data by starting with the `Deaths\_Lexis` file...).

Usage

```
readCHMDweb(provID = "can", item = "Deaths_1x1", fixup = TRUE, ...)
```

Arguments

provID	a single provID 3 character string, as returned by `getCHMDprovinces()`.
item	the statistical product you want, e.g., `"fltper_5x5"`. Only 1.
fixup	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?
...	extra arguments ultimately passed to `read.table()`. Not likely needed.

Details

This database is curated independently from the HMD/HFD family, and so file types and locations may be subject to change. If this happens, please notify the package maintainer.

Value

`data.frame` of the data item is invisibly returned

Examples

```
## Not run:
library(HMDHFDplus)
# grab province codes (including All Canada)
provs <- getCHMDprovinces()
# grab all mltper_5x5
# and stick into long data.frame:
mltper <- do.call(rbind, lapply(provs, function(provID){
               Dat      <- readCHMDweb(provID = provID, item = "mltper_5x5", fixup = TRUE)
                   Dat$provID <- provID
                   Dat
}))

## End(Not run)
```

---

readHFCweb *readHFCweb get HFC data straight from the web into R!*

Description

This concatenates the necessary url and calls `read.csv()` with all the necessary defaults to avoid annoying surprises. `Age` is given as an integer, along with an `AgeInterval`. The default behavior is to change the `AgeInterval` column to character and produce a logical indicator column, `OpenInterval`. `Year` is also given as the starting year of a `YearInterval`, rather than the original `Year1` and `Year2` columns. The column `Country` is renamed `CNTRY`. All other original columns are maintained. Output is invisibly returned, so you must assign it to take a look.

Usage

```
readHFCweb(CNTRY, item, fixup = TRUE, ...)
```

Arguments

CNTRY	character string of the HCD short code. Only one! Run `getHFCcountries(FALSE)` to see what the options are.
item	character string of the data product code, which is the base file name, but excluding the country code and file extension `.txt`. For instance, `"ASFRstand_TOT"`, `"ASFRstand_BO"`, `"TFRMAB_TOT"`, `"TFRMAB_BO"`. Only one item!
fixup	logical. Default `TRUE`. Should column classes be coerced to those more similar to HFD, HMD?
...	optional arguments passed to `read.csv()`. Not required.

Examples

```
## Not run:
DF <- readHFCweb("CZE","TFRMAB_TOT")
head(DF)
DF <- readHFCweb("CZE","ASFRstand_BO")
head(DF)

# get ASFRstand_BO for all countries where available:
Countries <- getHFCcountries()
# takes a minute to run

urls <- paste0("http://www.fertilitydata.org/data/",
               Countries,"/", Countries, "_",  "ASFRstand_BO", ".txt")

HaveBO <- RCurl::url.exists(urls)
 # we grab data for these countries:
 (Countries <- Countries[HaveBO])

# Also takes 1-15 min depending on internet connection and machine
# read in one at a time and stick together into long data.frame
allBO <- do.call(rbind,
        # this is the iteration of reading in
        lapply(Countries, function(CNTRY){
            readHFCweb(CNTRY, item = "ASFRstand_BO")
             })) # closes off the meta-rbind thingy
dim(allBO) # [1] 133237     31
unique(allBO$CNTRY)


## End(Not run)
```

readHFD `readHFD()` *reads a standard HFD .txt table as a* `data.frame`

Description

This calls `read.table()` with all the necessary defaults to avoid annoying surprises. The Age column is also stripped of `"-"` and `"+"` and converted to integer, and a logical indicator column called `OpenInterval` is added to show where these were located. Output is invisibly returned, so you must assign it to take a look. This is to avoid lengthy console printouts.

Usage

```
readHFD(filepath, fixup = TRUE, ...)
```

Arguments

`filepath`	path or connection to the HFD text file, including .txt suffix.
`fixup`	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?
`...`	other arguments passed to `read.table`, not likely needed.

Details

No details of note.

Value

data.frame of standard HFD output, except the Age column has been cleaned, and a new open age indicator column has been added.

Note

original function submitted by Josh Goldstein, modified by Tim Riffe.

readHFDweb *read an HFD data file directly from the web as an R data.frame*

Description

Read HFD data directly from the web. This function is useful for short reproducible examples, or to make code guaranteed to always use the most up to date version of a particular HFD data file. For working with the entire HFD for a comparative study, it may be more efficient to download the full HFD zip files and read in the elements using `readHFD()`. This function returns data formatted in the same way as `readHFD()`, that is, with Age columns (and others) converted to integer, and with open age group identifiers stored in a new logical column called `OpenInterval`. It is faster to specify `CNTRY` and `item` as arguments than to make the function figure out what's available. For repeated calls to this function, you can pass your username and password in as variables without having to include these in you R script by using `userInput()`– see example. The user also has the option of querying particular updates from the HFD revision history. If you wish to specify a particular update, you must know the date that a particular country was updated, in the format `"YYYYMMDD"`. These dates differ between countries, so keep a good record if you wish your work to be reproducible to that extent (as well as lightweight)!

Usage

```
readHFDweb(
  CNTRY = NULL,
  item = NULL,
  username = NULL,
  password = NULL,
  fixup = TRUE,
  Update = NULL
)
```

Arguments

`CNTRY`	character string of the HFD short code. Only one!
`item`	character string of the data product code, which is the base file name, but excluding the country code and file extension `.txt`. For instance, `"mabRR"` or `"tfrVHbo"`. If you're not sure, then leave it blank and a list will be shown. Only one item!
`username`	your HFD usernames, which is usually the email address you registered with
`password`	your HFD password. Don't make this a sensitive password, as things aren't encrypted.
`fixup`	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?
`Update`	character string of 8-digit date code of the format `"YYYYMMDD"`. Defaults to most recent update.

Details

You need to register for HFD to use this function: https://www.humanfertility.org. It is advised to pass in your credentials as named vectors rather than directly as character strings, so that they are not saved directly in your code. See examples. One option is to just save them in your Rprofile file.

Value

data.frame of the given HFD data file, modified in some friendly ways.

Examples

```
### # this will ask you to enter your login details in the R console
### DAT <- readHFDweb("JPN","tfrRR")
###
### # ---------------------------------------
### # this is a good way to reuse your login credentials without
### # having to reveal them in your R script.
### # if you want to do this in batch then I'm
### # afraid you'll have to find a clever way to
### # pass in your credentials without an interactive
### # session, such as reading them in from a system file of your own.
### myusername <- userInput()
### mypassword <- userInput()
### DAT <- readHMDweb("USA","mltper_1x1",mypassword,myusername)
###
### #---------------------------------------
### # this also works, but you'll need to make two selections,
### # plus enter data in the console twice:
### DAT <- readHFDweb()
```

---

readHMD `readHMD()` *reads a standard HMD .txt table as a* `data.frame`

Description

This calls `read.table()` with all the necessary defaults to avoid annoying surprises. The Age column is also stripped of `"+"` and converted to integer, and a logical indicator column called `OpenInterval` is added to show where these were located. If the file contains population counts, values are split into two columns for Jan 1 and Dec 31 of the year. Output is invisibly returned, so you must assign it to take a look. This is to avoid lengthy console printouts.

Usage

```
readHMD(filepath, fixup = TRUE, ...)
```

Arguments

`filepath`	path or connection to the HMD text file, including .txt suffix.
`fixup`	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?
`...`	other arguments passed to `read.table`, not likely needed.

Details

Population counts in the HMD typically refer to Jan 1st. One exception are years in which a territorial adjustment has been accounted for in estimates. For such years, 'YYYY-' refers to Dec 31 of the year before the adjustment, and 'YYYY+' refers to Jan 1 directly after the adjustment (adjustments are always made Jan 1st). In the data, it will just look like two different estimates for the same year, but in fact it is a definition change or similar. In order to remove headaches from potential territorial adjustments in the data, we simply create two columns, one for January 1st (e.g.,`"Female1"`) and another for Dec 31st (e.g.,`"Female2"`) . One can recover the adjustment coefficient for each year by taking the ratio $$Vx = P1(t+1) / P2(t)$$. In most years this will be 1, but in adjustment years there is a difference. This must always be accounted for when calculating rates and exposures. Argument `fixup` is outsourced to `HMDparse()`.

Value

data.frame of standard HMD output, except the Age column has been cleaned, and a new open age indicator column has been added. If the file is Population.txt or Population5.txt, there will be two columns each for males and females.

Note

function written by Tim Riffe.

`readHMDweb` *readHMDweb a basic HMD data grabber.*

Description

This is a basic HMD data grabber, based on Carl Boe's original `HMD2R()`. It will only grab a single HMD statistical product from a single country. Some typical R pitfalls are removed: The Age column is coerced to integer, while an AgeInterval column is created. Also Population counts are placed into two columns, for Jan. 1st and Dec. 31 of the same year, so as to remove headaches from population universe adjustments, such as territorial changes. Fewer options means less to break. To do more sophisticated data extraction, iterate over country codes or statistical items. Reformatting can be done outside this function using, e.g., `long2mat()`. Argument `fixup` is outsourced to `HMDparse()`.

Usage

```
readHMDweb(CNTRY, item, username, password, fixup = TRUE)
```

Arguments

`CNTRY`	character. HMD population letter code. If not spelled right, or not specified, the function provides a selection list. Only 1.
`item`	character. The statistical product you want, e.g., `"fltper_1x1"`. Only 1.

username	character. Your HMD user id, usually the email address you registered with the HMD under. If left blank, you'll be prompted. Do that if you don't mind the typing and prefer not to save your username in your code.
password	character. Your HMD password. If left blank, you'll be prompted. Do that if you don't mind the typing and prefer not to save your password in your code.
fixup	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?

Details

This function points to the new HMD website (from June 2022) rather than the mirror of the old site that it temporarily pointed to; If your credentials fail then a likely reason is that you need to re-register at the new HMD website https://www.mortality.org/Account/UserAgreement. As soon as you register, your new credentials should work.

Value

data.frame of the HMD product, read as as `readHMD()` would read it.

---

readJMDweb *read data from the Japan Mortality Database into R*

Description

JMD data are formatted exactly as HMD data. This function simply parses the necessary url together given a prefecture code and data item (same nomenclature as HMD). Data is parsed using `HMDparse()`, which converts columns into useful and intuitive classes, for ready-use. See `?HMDparse` for more information on type conversions. No authentication is required for this database. Only a single item/prefecture is downloaded. Loop for more complex calls (See examples). The prefID is not appended as a column, so be mindful of this if appending several items together into a single `data.frame`. Note that at the time of this writing, the finest Lexis resolution for prefectural lifetables is 5x5 (5-year, 5-year age groups). Raw data are, however, provided in 1x1 format, and deaths are also available in triangles.

Usage

```
readJMDweb(prefID = "01", item = "Deaths_5x5", fixup = TRUE, ...)
```

Arguments

prefID	a single prefID 2-digit character string, ranging from `"00"` to `"47"`.
item	the statistical product you want, e.g., `"fltper_5x5"`. Only 1.
fixup	logical. Should columns be made more user-friendly, e.g., forcing Age to be integer?
...	extra arguments ultimately passed to `read.table()`. Not likely needed.

Details

No details of note. This database in independently maintained, so file types/locations are subject to change. If this happens, please notify the package maintainer.

Value

`data.frame` of the data item is invisibly returned

Examples

```
## Not run:
library(HMDHFDplus)
# grab prefecture codes (including All Japan)
prefectures <- getJMDprefectures()
# grab all mltper_5x5
# and stick into long data.frame:
mltper <- do.call(rbind, lapply(prefectures, function(prefID){
               Dat      <- readJMDweb(prefID = prefID, item = "mltper_5x5", fixup = TRUE)
                   Dat$PrefID <- prefID
                   Dat
}))

## End(Not run)
```

`userInput` *userInput let the user type in a character string*

Description

this is useful for asking the user for a username or password, so that it goes directly to a variable and doesn't get inadvertently saved into an R script. There are no arguments. This will only return a character string. This is low key, don't bother using it for data entry. Just type characters, no need to put it in quotes, pressing enter will cause the function to return. Output will not be printed to the console, but it can be assigned directly. This is useful to have as an auxiliary function in case multiple calls to functions such as `readHMDweb()` are desired.

Usage

```
userInput(silent = FALSE)
```

Arguments

`silent`	logical should a little prompt be given, telling the user to enter text in the console?

Value

a character string, as given by the user.

Examples

```
### mypassword <- userInput()
### myusername <- userInput()
### DAT <- readHMDweb("USA","mltper_1x1",mypassword,myusername)
```

Index

age2int, 2

getCHMDprovinces, 3
getHFCcountries, 4
getHFDcountries, 4
getHFDdate, 5
getHFDitemavail, 5
getHMDcountries, 6
getHMDitemavail, 6
getJMDprefectures, 7

HFCparse, 7
HFDparse, 8
HMDparse, 9

readCHMDweb, 9
readHFCweb, 10
readHFD, 12
readHFDweb, 12
readHMD, 14
readHMDweb, 15
readJMDweb, 16

userInput, 17