Package 'gender'

September 4, 2015

Type Package

Title Predict Gender from Names Using Historical Data

Version 0.5.1

Date 2015-09-03

Description Encodes gender based on names and dates of birth using historical datasets. By using these datasets instead of lists of male and female names, this package is able to more accurately guess the gender of a name, and it is able to report the probability that a name was male or female.

URL https://github.com/ropensci/gender

Depends R (>= 3.0.0), utils, stats

Imports dplyr (>= 0.4.2), httr (>= 1.0.0), jsonlite (>= 0.9.16)

Suggests genderdata (>= 0.5.0), ggplot2 (>= 1.0.0), knitr (>= 1.11), testthat (>= 0.10.0)

Additional_repositories http://packages.ropensci.org

LazyData yes

License MIT + file LICENSE

VignetteBuilder knitr

BugReports https://github.com/ropensci/gender/issues

NeedsCompilation no

Author Lincoln Mullen [aut, cre],
Cameron Blevins [ctb],
Ben Schmidt [ctb]

Maintainer Lincoln Mullen <lincoln@lincolnmullen.com>

Repository CRAN

Date/Publication 2015-09-04 07:42:15

R topics documented:

gender-package 2
check_genderdata_package 2
gender 3
gender_df 4
install_genderdata_package 5

Index **6**

`gender-package` *Gender: predict gender by name from historical data*

Description

Gender: predict gender from names using historical data

Details

Encodes gender based on names and dates of birth, using U.S. Census or Social Security data sets. Requires separate downlaod of datasets, which should be done automatically and can be done manually by running `install_genderdata_package()`.

Author(s)

`<lincoln@lincolnmullen.com>`

`check_genderdata_package` *Check whether to install data for gender function and install if necessary*

Description

If the genderdata package is not installed, install it from GitHub using devtools. If it is not up to date, reinstall it.

Usage

```
check_genderdata_package()
```

gender *Predict gender from first names using historical data*

Description

This function predicts the gender of a first name given a year or range of years in which the person was born. The prediction can use one of several data sets suitable for different time periods or geographical regions. See the package vignette for suggestions on using this function with multiple names and for a discussion of which data set is most suitable for your research question. When using certains methods, the `genderdata` data package is required; you will be prompted to install it if it is not already available.

Usage

```
gender(names, years = c(1932, 2012), method = c("ssa", "ipums", "napp",
  "kantrowitz", "genderize", "demo"), countries = c("United States", "Canada",
  "United Kingdom", "Germany", "Iceland", "Norway", "Sweden"))
```

Arguments

`names`	First names as a character vector. Names are case insensitive.
`years`	The birth year of the name whose gender is to be predicted. This argument can be either a single year, a range of years in the form `c(1880, 1900)`. If no value is specified, then for the `"ssa"` method it will use the period 1932 to 2012; acceptable years for the SSA method range from 1880 to 2012, but for years before 1930 the IPUMS method is probably more accurate. For the `"ipums"` method the default range is the period 1789 to 1930, which is also the range of acceptable years. For the `"napp"` method the default range is the period 1758 to 1910, which is also the range of acceptable years. If a year or range of years is specified, then the names will be looked up for that period.
`method`	This value determines the data set that is used to predict the gender of the name. The `"ssa"` method looks up names based from the U.S. Social Security Administration baby name data. (This method is based on an implementation by Cameron Blevins.) The `"ipums"` method looks up names from the U.S. Census data in the Integrated Public Use Microdata Series. (This method was contributed by Ben Schmidt.) The `"kantrowitz"` method uses the Kantrowitz corpus of male and female names. The `"genderize"` method uses the Genderize.io <http://genderize.io/> API, which is based on "user profiles across major social networks." The `"demo"` method is uses the top 100 names in the SSA method; it is provided only for demonstration purposes when the `genderdata` package is not installed and it is not suitable for research purposes.
`countries`	The countries for which datasets are being used. For the `"ssa"` and `"ipums"` methods, the only valid option is `"United States"` which will be assumed if no argument is specified. For the `"napp"` method, you may specify a character vector with any of the following countries: `"Canada"`, `"United Kingdom"`, `"Germany"`, `"Iceland"`, `"Norway"`, `"Sweden"`. For the `"kantrowitz"` and `"genderize"` methods, no country should be specified.

Value

Returns a data frame containing the results of predicting the gender. The exact components of the returned list will depend on the specific method used. They include the following:

`name`	The name for which the gender has been predicted.
`proportion_male`	The proportion of male names for the given range of years.
`proportion_female`	The proportion of female names for the given range of years.
`gender`	The predicted gender based on the proportion of male and female names. Possible values are `"male"` and `"female"` for proportions above `0.5`, `"either"` for proportions that are exactly `0.5`, and `NA` for combinations of names and years for which a gender cannot be predicted using the given method.
`year_min`	The lower bound (inclusive) of the year range used for the prediction.
`year_max`	The upper bound (inclusive) of the year range used for the prediction.

Examples

```
gender("madison", method = "demo", years = 1985)
gender("madison", method = "demo", years = c(1900, 1985))
# SSA method
## Not run: gender("madison", method = "demo", years = c(1900, 1985))
# IPUMS method
## Not run: gender("madison", method = "ipums", years = 1860)
# NAPP method
## Not run: gender("madison", method = "napp", countries = c("Sweden", "Denmark"))
```

---

gender_df *Use gender prediction with data frames*

Description

In a common use case for gender prediction, you have a data frame with a column for first names and a column for birth years (or, two columns specifying a minimum and a maximum potential birth year). This function wraps the `gender` function to efficiently apply it to such a data frame. The result is a data frame with one prediction of the gender for each unique combination of first name and birth year. The resulting data frame can then be merged back into your original data frame.

Usage

```
gender_df(data, name_col = "name", year_col = "year", method = c("ssa",
  "ipums", "napp", "demo"))
```

Arguments

data	A data frame containing first names and birth year or range of potential birth years.
name_col	A string specifying the name of the column containing the first names.
year_col	Either a single string specifying the birth year associated with the first name, or character vector with two elements: the names of the columns with the minimum and maximum years for the range of potential birth years.
method	One of the historical methods provided by this package: "ssa", "ipums", "napp", or "demo". See gender for details.

Value

A data frame with columns from the output of the gender function, and one row for each unique combination of first names and birth years.

See Also

gender

Examples

```
library(dplyr)
demo_df <- data_frame(names = c("Hillary", "Hillary", "Hillary",
                                "Madison", "Madison"),
                      birth_year = c(1930, 2000, 1930, 1930, 2000),
                      min_year = birth_year - 1,
                      max_year = birth_year + 1,
                      stringsAsFactors = FALSE)

# Using the birth year for the predictions.
# Notice that the duplicate value for Hillary in 1930 is removed
gender_df(demo_df, method = "demo",
          name_col = "names", year_col = "birth_year")

# Using a range of years
gender_df(demo_df, method = "demo",
          name_col = "names", year_col = c("min_year", "max_year"))
```

---

install_genderdata_package

Install the genderdata package after checking with the user

Description

Install the genderdata package after checking with the user

Usage

```
install_genderdata_package()
```

Index

check_genderdata_package, 2

gender, 3, *4, 5*
gender-package, 2
gender_df, 4

install_genderdata_package, 5