Package 'wru'

September 1, 2017

Version 0.1-7

Date 2017-8-31

Title Who are You? Bayesian Prediction of Racial Category Using Surname and Geolocation

Author Kabir Khanna [aut, cre], Kosuke Imai [aut, cre], Hubert Jin [ctb]

Maintainer Kabir Khanna <kkhanna@princeton.edu>

Description Predicts individual race/ethnicity using surname, geolocation, and other attributes, such as gender and age. The method utilizes the Bayes' Rule to compute the posterior probability of each racial category for any given individual. The package implements methods described in Imai and Khanna (2015) ``Improving Ecological Inference by Predicting Individual Ethnicity from Voter Registration Records" <DOI:10.1093/pan/mpw001>.

URL https://github.com/kosukeimai/wru

BugReports https://github.com/kosukeimai/wru/issues

Depends R (>= 3.2.0), utils

Imports devtools (>= 1.10.0)

Suggests testthat

LazyLoad yes

LazyData yes

License GPL (>= 3)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-09-01 08:03:35 UTC

R topics documented:

`census_geo_api` *Census Data download function.*

Description

`census_geo_api` retrieves U.S. Census geographic data for a given state.

Usage

```
census_geo_api(key, state, geo = "tract", age = FALSE, sex = FALSE,
  retry = 0)
```

Arguments

`key`	A required character object. Must contain user's Census API key, which can be requested here.
`state`	A required character object specifying which state to extract Census data for, e.g., `"NJ"`.
`geo`	A character object specifying what aggregation level to use. Use `"county"`, `"tract"`, `"block"`, or `"place"`. Default is `"tract"`. Warning: extracting block-level data takes very long.
`age`	A `TRUE/FALSE` object indicating whether to condition on age or not. If `FALSE` (default), function will return Pr(Geolocation | Race). If `TRUE`, function will return Pr(Geolocation, Age | Race). If *sex* is also `TRUE`, function will return Pr(Geolocation, Age, Sex | Race).
`sex`	A `TRUE/FALSE` object indicating whether to condition on sex or not. If `FALSE` (default), function will return Pr(Geolocation | Race). If `TRUE`, function will return Pr(Geolocation, Sex | Race). If *age* is also `TRUE`, function will return Pr(Geolocation, Age, Sex | Race).
`retry`	The number of retries at the census website if network interruption occurs.

Details

This function allows users to download U.S. Census 2010 geographic data, at either the county, tract, block, or place level, for a particular state.

Value

Output will be an object of class `list`, indexed by state names. It will consist of the original user-input data with additional columns of Census geographic data.

References

Relies on get_census_api, get_census_api_2, and vec_to_chunk functions authored by Nicholas Nagle, available here.

Examples

```
## Not run: census_geo_api(key = "...", states = c("NJ", "DE"), geo = "block")
## Not run: census_geo_api(key = "...", states = "FL", geo = "tract", age = TRUE, sex = TRUE)
```

`census_helper` *Census helper function.*

Description

`census_helper` links user-input dataset with Census geographic data.

Usage

```
census_helper(key, voter.file, states = "all", geo = "tract", age = FALSE,
  sex = FALSE, census.data = NA, retry = 0)
```

Arguments

`key`	A required character object. Must contain user's Census API key, which can be requested here.
`voter.file`	An object of class `data.frame`. Must contain field(s) named *county*, *tract*, *block*, and/or *place* specifying geolocation. These should be character variables that match up with U.S. Census categories. County should be three characters (e.g., "031" not "31"), tract should be six characters, and block should be four characters. Place should be five characters if it is included.
`states`	A character vector specifying which states to extract Census data for, e.g. `c("NJ", "NY")`. Default is `"all"`, which extracts Census data for all states contained in user-input data.
`geo`	A character object specifying what aggregation level to use. Use `"county"`, `"tract"`, or `"block"`. Default is `"tract"`. Warning: extracting block-level data takes very long.
`age`	A `TRUE`/`FALSE` object indicating whether to condition on age or not. If `FALSE` (default), function will return Pr(Geolocation | Race). If `TRUE`, function will return Pr(Geolocation, Age | Race). If *sex* is also `TRUE`, function will return Pr(Geolocation, Age, Sex | Race).

sex	A TRUE/FALSE object indicating whether to condition on sex or not. If FALSE (default), function will return Pr(Geolocation | Race). If TRUE, function will return Pr(Geolocation, Sex | Race). If *age* is also TRUE, function will return Pr(Geolocation, Age, Sex | Race).
census.data	A optional census object of class list containing pre-saved Census geographic data. Can be created using get_census_data function. If *census.data* is provided, the *age* element must have the same value as the *age* option specified in this function (i.e., TRUE in both or FALSE in both). Similarly, the *sex* element in the object provided in *census.data* must have the same value as the *sex* option here. If *census.data* is missing, Census geographic data will be obtained via Census API.
retry	The number of retries at the census website if network interruption occurs.

Details

This function allows users to link their geocoded dataset (e.g., voter file) with U.S. Census 2010 data. The function extracts Census Summary File data at the county, tract, or block level using the 'UScensus2010' package. Census data calculated are Pr(Geolocation | Race) where geolocation is county, tract, or block.

Value

Output will be an object of class data.frame. It will consist of the original user-input data with additional columns of Census data.

Examples

```
## Not run: census_helper(key = "...", voter.file = voters, states = "nj", geo = "block")
## Not run: census_helper(key = "...", voter.file = voters, states = "all", geo = "tract",
age = TRUE, sex = TRUE)
## End(Not run)
```

---

get_census_api *Census API function.*

Description

get_census_api obtains U.S. Census data via the public API.

Usage

```
get_census_api(data_url, key, vars, region, retry = 0)
```

Arguments

data_url	URL root of the API, including the question mark, e.g., "https://api.census.gov/data/2010/sf1?".
key	A required character object containing user's Census API key, which can be requested here.
vars	A character vector of variables to get, e.g., c("P0050003","P0050004","P0050005", "P0050006"). If there are more than 50 variables, then function will automatically split variables into separate queries.
region	Character object specifying which region to obtain data for. Must contain "for" and possibly "in", e.g., "for=block:1213&in=state:47+county:015+tract:*".
retry	The number of retries at the census website if network interruption occurs.

Details

This function obtains U.S. Census data via the public API. User can specify the variables and region(s) for which to obtain data.

Value

If successful, output will be an object of class data.frame. If unsuccessful, function prints the URL query that caused the error.

References

Based on code authored by Nicholas Nagle, which is available here.

Examples

```
## Not run: get_census_api(data_url = "https://api.census.gov/data/2010/sf1?", key = "...",
vars = c("P0050003","P0050004","P0050005", "P0050006"), region = "for=county:*&in=state:34")
## End(Not run)
```

---

get_census_api_2 *Census API URL assembler.*

Description

get_census_api_2 assembles URL components for get_census_api.

Usage

```
get_census_api_2(data_url, key, get, region, retry = 0)
```

Arguments

data_url	URL root of the API, including the question mark, e.g., "https://api.census.gov/data/2010/sf1?".
key	A required character object containing user's Census API key, which can be requested here.
get	A character vector of variables to get, e.g., c("P0050003","P0050004","P0050005", "P0050006"). If there are more than 50 variables, then function will automatically split variables into separate queries.
region	Character object specifying which region to obtain data for. Must contain "for" and possibly "in", e.g., "for=block:1213&in=state:47+county:015+tract:*".
retry	The number of retries at the census website if network interruption occurs.

Details

This function assembles the URL components and sends the request to the Census server. It is used by the get_census_api function. The user should not need to call this function directly.

Value

If successful, output will be an object of class data.frame. If unsuccessful, function prints the URL query that was constructed.

References

Based on code authored by Nicholas Nagle, which is available here.

Examples

```
## Not run: get_census_api_2(data_url = "https://api.census.gov/data/2010/sf1?", key = "...",
get = c("P0050003","P0050004","P0050005", "P0050006"), region = "for=county:*&in=state:34")
## End(Not run)
```

---

get_census_data *Multilevel Census data download function.*

Description

get_census_data returns county-, tract-, and block-level Census data for specified state(s). Using this function to download Census data in advance can save considerable time when running predict_race and census_helper.

Usage

```
get_census_data(key, states, age = FALSE, sex = FALSE,
  census.geo = "block", retry = 0)
```

Arguments

key	A required character object containing a valid Census API key, which can be requested here.
states	which states to extract Census data for, e.g., c("NJ", "NY").
age	A TRUE/FALSE object indicating whether to condition on age or not. If FALSE (default), function will return Pr(Geolocation | Race). If TRUE, function will return Pr(Geolocation, Age | Race). If *sex* is also TRUE, function will return Pr(Geolocation, Age, Sex | Race).
sex	A TRUE/FALSE object indicating whether to condition on sex or not. If FALSE (default), function will return Pr(Geolocation | Race). If TRUE, function will return Pr(Geolocation, Sex | Race). If *age* is also TRUE, function will return Pr(Geolocation, Age, Sex | Race).
census.geo	An optional character vector specifying what level of geography to use to merge in U.S. Census 2010 geographic data. Currently "county", "tract", "block", and "place" are supported.
retry	The number of retries at the census website if network interruption occurs.

Value

Output will be an object of class list indexed by state. Output will contain a subset of the following elements: state, age, sex, county, tract, block, and place.

Examples

```
## Not run: get_census_data(key = "...", states = c("NJ", "NY"), age = TRUE, sex = FALSE)
```

---

merge_surnames *Surname probability merging function.*

Description

merge_surnames merges surnames in user-input dataset with corresponding race/ethnicity probabilities from U.S. Census Surname List and Spanish Surname List.

Usage

```
merge_surnames(voter.file, surname.year = 2010, clean.surname = T,
  impute.missing = T)
```

Arguments

voter.file	An object of class data.frame. Must contain a field named 'surname' containing list of surnames to be merged with Census lists.
surname.year	An object of class numeric indicating which year Census Surname List is from. Accepted values are 2010 and 2000. Default is 2010.

`clean.surname` A TRUE/FALSE object. If TRUE, any surnames in *voter.file* that cannot initially be matched to surname lists will be cleaned, according to U.S. Census specifications, in order to increase the chance of finding a match. Default is TRUE.

`impute.missing` A TRUE/FALSE object. If TRUE, race/ethnicity probabilities will be imputed for unmatched names using race/ethnicity distribution for all other names (i.e., not on Census List). Default is TRUE.

Details

This function allows users to match surnames in their dataset with the U.S. Census Surname List (from 2000 or 2010) and Spanish Surname List to obtain Pr(Race | Surname) for each of the five major racial groups.

By default, the function matches surnames to the Census list as follows: 1) Search raw surnames in Census surname list; 2) Remove any punctuation and search again; 3) Remove any spaces and search again; 4) Remove suffixes (e.g., Jr) and search again; 5) Split double-barreled surnames into two parts and search first part of name; 6) Split double-barreled surnames into two parts and search second part of name; 7) For any remaining names, impute probabilities using distribution for all names not appearing on Census list.

Each step only applies to surnames not matched in a previous ste. Steps 2 through 7 are not applied if `clean.surname` is FALSE.

Note: Any name appearing only on the Spanish Surname List is assigned a probability of 1 for Hispanics/Latinos and 0 for all other racial groups.

Value

Output will be an object of class `data.frame`. It will consist of the original user-input data with additional columns that specify the part of the name matched with Census data (*surname.match*), and the probabilities Pr(Race | Surname) for each racial group (*p_whi* for White, *p_bla* for Black, *p_his* for Hispanic/Latino, *p_asi* for Asian and Pacific Islander, and *p_oth* for Other/Mixed).

Examples

```
data(voters)
merge_surnames(voters)
```

---

names.all *Dataset containing Census Surname List and Spanish Surname Lists.*

Description

A dataset containing Census Surname List, which is augmented with Census Spanish Surname List. Variables are as follows:

- surname
- p_whi (i.e., Pr(White | Surname)

- p_bla (i.e., Pr(Black | Surname)
- p_his (i.e., Pr(Hispanic/Latino | Surname)
- p_asi (i.e., Pr(Asian | Surname)
- p_oth (i.e., Pr(Other | Surname)

For any surnames appearing only on Spanish Surname List, Pr(Hispanic/Latino | Surname) = 1, and remaining probabilities are set to zero.

Format

A data frame with 157,728 rows and 6 variables.

`predict_race` *Race prediction function.*

Description

`predict_race` makes probabilistic estimates of individual-level race/ethnicity.

Usage

```
predict_race(voter.file, census.surname = TRUE, surname.only = FALSE,
  surname.year = 2010, census.geo, census.key, census.data = NA,
  age = FALSE, sex = FALSE, party, retry = 0)
```

Arguments

`voter.file`	An object of class `data.frame`. Must contain a row for each individual being predicted, as well as a field named *surname* containing each individual's surname. If using geolocation in predictions, *voter.file* must contain a field named *state*, which contains the two-character abbreviation for each individual's state of residence (e.g., `"nj"` for New Jersey). If using Census geographic data in race/ethnicity predictions, *voter.file* must also contain at least one of the following fields: *county*, *tract*, *block*, and/or *place*. These fields should contain character strings matching U.S. Census categories. County is three characters (e.g., `"031"` not `"31"`), tract is six characters, and block is four characters. Place is five characters. See below for other optional fields.
`census.surname`	A `TRUE/FALSE` object. If `TRUE`, function will call `merge_surnames` to merge in Pr(Race | Surname) from U.S. Census Surname List (2000 or 2010) and Spanish Surname List. If `FALSE`, `voter.file` object must contain additional fields specifying Pr(Race | Surname), named as follows: *p_whi* for Whites, *p_bla* for Blacks, *p_his* for Hispanics/Latinos, *p_asi* for Asians, and/or *p_oth* for Other. Default is `TRUE`.
`surname.only`	A `TRUE/FALSE` object. If `TRUE`, race predictions will only use surname data and calculate Pr(Race | Surnname). Default is `FALSE`.

`surname.year` A number to specify the year of the census surname statistics. These surname statistics is stored in the data, and will be automatically loaded. The default value is `2010`, which means the surname statistics from the 2010 census will be used. Currently, the other available choice is `2000`.

`census.geo` An optional character vector specifying what level of geography to use to merge in U.S. Census 2010 geographic data. Currently `"county"`, `"tract"`, `"block"`, and `"place"` are supported. Note: sufficient information must be in user-defined *voter.file* object. If *census.geo* `= "county"`, then *voter.file* must have column named `county`. If *census.geo* `= "tract"`, then *voter.file* must have columns named `county` and `tract`. And if *census.geo* `= "block"`, then *voter.file* must have columns named `county`, `tract`, and `block`. If *census.geo* `= "place"`, then *voter.file* must have column named `place`. Specifying *census.geo* will call `census_helper` function to merge Census geographic data at specified level of geography.

`census.key` A character object specifying user's Census API key. Required if *census.geo* is specified, because a valid Census API key is required to download Census geographic data.

`census.data` A list indexed by two-letter state abbreviations, which contains pre-saved Census geographic data. Can be generated using `get_census_data` function.

`age` An optional `TRUE/FALSE` object specifying whether to condition race predictions on age (in addition to surname and geolocation). Default is `FALSE`. Must be same as *age* in *census.data* object. May only be set to `TRUE` if `census.geo` option is specified. If `TRUE`, *voter.file* should include a numerical variable *age*.

`sex` optional `TRUE/FALSE` object specifying whether to condition race predictions on sex (in addition to surname and geolocation). Default is `FALSE`. Must be same as *sex* in *census.data* object. May only be set to `TRUE` if `census.geo` option is specified. If `TRUE`, *voter.file* should include a numerical variable *sex*, where *sex* is coded as 0 for males and 1 for females.

`party` An optional character object specifying party registration field in *voter.file*, e.g., *party* `= "PartyReg"`. If specified, race/ethnicity predictions will be conditioned on individual's party registration (in addition to geolocation). Whatever the name of the party registration field in *voter.file*, it should be coded as 1 for Democrat, 2 for Republican, and 0 for Other.

`retry` The number of retries at the census website if network interruption occurs.

Details

This function implements the Bayesian race prediction methods outlined in Imai and Khanna (2015). The function produces probabilistic estimates of individual-level race/ethnicity, based on surname, geolocation, and party.

Value

Output will be an object of class `data.frame`. It will consist of the original user-input data with additional columns with predicted probabilities for each of the five major racial categories: *pred.whi* for White, *pred.bla* for Black, *pred.his* for Hispanic/Latino, *pred.asi* for Asian/Pacific Islander, and *pred.oth* for Other/Mixed.

Examples

```
data(voters)
predict_race(voters, surname.only = TRUE)
predict_race(voter.file = voters, surname.only = TRUE)
## Not run: predict_race(voter.file = voters, census.geo = "tract", census.key = "...")
## Not run: predict_race(voter.file = voters, census.geo = "tract", census.key = "...", age = T)
## Not run: predict_race(voter.file = voters, census.geo = "place", census.key = "...", sex = T)
## Not run: CensusObj <- get_census_data("...", state = c("NY", "DC", "NJ"));
predict_race(voter.file = voters, census.geo = "tract", census.data = CensusObj, party = "PID")
## End(Not run)
## Not run: CensusObj2 <- get_census_data(key = "...", state = c("NY", "DC", "NJ"), age = T, sex = T);
predict_race(voter.file = voters, census.geo = "tract", census.data = CensusObj2, age = T, sex = T)
## End(Not run)
## Not run: CensusObj3 <- get_census_data(key = "...", state = c("NY", "DC", "NJ"), census.geo = "place");
predict_race(voter.file = voters, census.geo = "place", census.data = CensusObj3)
## End(Not run)
```

`surnames2000` *Census Surname List (2000).*

Description

Census Surname List from 2000 with race/ethnicity probabilities by surname.

Usage

```
surnames2000
```

Format

A data frame with 157,728 rows and 6 variables:

surname Surname

p_whi Pr(White | Surname)

p_bla Pr(Black | Surname)

p_his Pr(Hispanic/Latino | Surname)

p_asi Pr(Asian/Pacific Islander | Surname)

p_oth Pr(Other | Surname) #'

Examples

```
data(surnames2000)
```

surnames2010 *Census Surname List (2010).*

Description

Census Surname List from 2010 with race/ethnicity probabilities by surname.

Usage

```
surnames2010
```

Format

A data frame with 167,613 rows and 6 variables:

surname Surname

p_whi Pr(White | Surname)

p_bla Pr(Black | Surname)

p_his Pr(Hispanic/Latino | Surname)

p_asi Pr(Asian/Pacific Islander | Surname)

p_oth Pr(Other | Surname) #'

Examples

```
data(surnames)
```

vec_to_chunk *Variable vector into chunks.*

Description

vec_to_chunk takes a list of variables and collects them into 50-variable chunks.

Usage

```
vec_to_chunk(x)
```

Arguments

x	Character vector of variable names.

Details

This function takes a list of variable names and collects them into chunks with no more than 50 variables each. This helps to get around requests with more than 50 variables,because the API only allows queries of 50 variables at a time. The user should not need to call this function directly.

Value

Object of class `list`.

References

Based on code authored by Nicholas Nagle, which is available here.

Examples

```
vec_to_chunk(x = c(paste("P012F0", seq(10:49), sep = ""), paste("P012I0", seq(10, 49), sep = "")))
```

---

voters *Example voter file.*

Description

An example dataset containing voter file information.

Usage

```
voters
```

Format

A data frame with 10 rows and 12 variables:

VoterID Voter identifier (numeric)

surname Surname

state State of residence

CD Congressional district

county Census county (three-digit code)

tract Census tract (six-digit code)

block Census block (four-digit code)

precinct Voting precinct

place Voting place

age Age in years

sex 0=male, 1=female

party Party registration (character)

PID Party registration (numeric) #'

Examples

```
data(voters)
```

Index