

Package ‘censusr’

July 22, 2025

Type Package

Title Collect Data from the Census API

Version 0.0.4

Date 2018-01-23

Description Use the US Census API to collect summary data tables
for SF1 and ACS datasets at arbitrary geographies.

URL <https://github.com/transportfoundry/censusr>

BugReports <https://github.com/transportfoundry/censusr/issues>

License MIT + file LICENSE

LazyData TRUE

Depends dplyr (>= 0.4.3), httr (>= 1.0.0)

Suggests knitr, rmarkdown, xml2

VignetteBuilder knitr

RoxygenNote 6.0.1

Imports stringr

NeedsCompilation no

Author Greg Macfarlane [cre, aut],
Josie Kressner [aut]

Maintainer Greg Macfarlane <greg@transportfoundry.com>

Repository CRAN

Date/Publication 2018-01-25 16:40:14 UTC

Contents

aggregate_moe	2
append_geoid	3
call_api_once	4
call_census_api	4
call_geolocator	5

2

aggregate_moe

call_geolocator_latlon

6

censusr

6

get_geo_url

7

Index

8

aggregate_moe	<i>Aggregated margin of error across multiple geographies</i>
---------------	---

Description

Aggregated margin of error across multiple geographies

Usage

aggregate_moe(x)

Arguments

x A numeric vector containing margins of error for estimates in multiple geographies.

Details

Applies a root sum of squared errors. See page A-14 of this guide: <https://www.census.gov/library/publications/2009/acs/rese>

Value

The aggregated margin of error for the geographies.

Author(s)

Josie Kressner

Examples

```
x <- c(3, 5, 12, 4)
aggregate_moe(x)
data_frame(x = x, group = c(1, 1, 2, 2)) %>%
  group_by(group) %>%
  summarise(moe = aggregate_moe(x))
```

append_geoid	<i>Retrieve GEOID from the Census Geocoder by address</i>
--------------	---

Description

Returns GEOID for 2010 geographies.

Usage

```
append_geoid(address, geoid_type = "bl")
```

Arguments

address	A tibble/data frame with (at a minimum, others can be present) either character columns street, city, and state OR numeric columns lat and lon. Lat/lon columns take priority.
geoid_type	GEOID level to return, c('co', 'tr', 'bg', 'bl'). Defaults to block.

Details

It is possible in some situations for the geolocator API to return a 500 error. Repeating the call should avoid this.

Value

the original tibble with GEOIDs appended as a new column called geoid.

Examples

```
## Not run:
airports <- dplyr::data_frame(
  street = "700 Catalina Dr", city = "Daytona Beach", state = "FL"
)
append_geoid(airports, 'tr')

## End(Not run)
```

call_api_once	<i>Call Census API for a set of variables</i>
---------------	---

Description

This is an internal function and is not intended for users. See instead [call_census_api](#).

Usage

```
call_api_once(variables_to_get, geoid, allgeos, data_source, year, period,
              api_key)
```

Arguments

variables_to_get	A character vector of the desired variable names for the Census API call, defined at https://www.census.gov/data/developers/data-sets.html
geoid	A character string with a FIPS code, between 2 and 15 digits long.
allgeos	(optional) A string identifying the type of geography for which to collect data within the the requested geoids. Must be one of c('co', 'tr', 'bg', 'bl'). For instance, if allgeos = "bg", will return all block groups within the given geoids.
data_source	A string identifying whether the SF1 (decennial census) or ACS data is desired.
year	If data_source = "acs", the final year of the summary period. Default is 2013.
period	If data_source = "acs", the length of aggregation period. Default is 5, or a 5-year aggregation table.
api_key	The user's Census API key (as a character string). You can get a free key from [Census](http://api.census.gov/data/key_signup.html). See vignette('censusr', package = 'censusr') to setup a default key as an environment variable.

Value

A `codedata.frame` with the requested variables at the requested geography.

call_census_api	<i>Retrieve data from the Census API</i>
-----------------	--

Description

Returns Census data for the 2010 SF1 or ACS 2013-2015 1-, 3-, and 5-Yr aggregations for requested variables and geographies.

Usage

```
call_census_api(variables_to_get, names = NULL, geoids, allgeos = NULL,
  data_source = c("sf1", "acs"), year = 2013, period = 5,
  api_key = NULL)
```

Arguments

<code>variables_to_get</code>	A character vector of the desired variable names for the Census API call, defined at https://www.census.gov/data/developers/data-sets.html
<code>names</code>	A character vector of the same length as <code>variables_to_get</code> giving the user-defined names for the variables (optional). Defaults to raw API names.
<code>geoids</code>	A character vector of FIPS codes; must be at least to the county (5-digit) level, and can accept down to blocks (15-digit).
<code>allgeos</code>	(optional) A string identifying the type of geography for which to collect data within the the requested geoids. Must be one of <code>c('co', 'tr', 'bg', 'bl')</code> . For instance, if <code>allgeos = "bg"</code> , will return all block groups within the given geoids.
<code>data_source</code>	A string identifying whether the SF1 (decennial census) or ACS data is desired.
<code>year</code>	If <code>data_source = "acs"</code> , the final year of the summary period. Default is 2013.
<code>period</code>	If <code>data_source = "acs"</code> , the length of aggregation period. Default is 5, or a 5-year aggregation table.
<code>api_key</code>	The user's Census API key (as a character string). You can get a free key from [Census](http://api.census.gov/data/key_signup.html). See <code>vignette('censusr', package = 'censusr')</code> to setup a default key as an environment variable.

Details

See `vignette('censusr', package = 'censusr')` for examples.

Value

a `data_frame` with each requested variable at each requested geography.

<code>call_geolocator</code>	<i>Call geolocator for one address</i>
------------------------------	--

Description

Call geolocator for one address

Usage

```
call_geolocator(street, city, state)
```

Arguments

street	A character string indicating a street name and number
city	A character string indicating a city
state	A two-digit character string with a state postal code

Value

A character string representing the Census block of the supplied address.
importFrom utils URLencode importFrom httr GET stop_for_status

call_geolocator_latlon	<i>Call geolocator for one address with lat/lon</i>
------------------------	---

Description

Call geolocator for one address with lat/lon

Usage

call_geolocator_latlon(lat, lon)

Arguments

lat	A numeric value
lon	A numeric value

Value

A character string representing the Census block of the supplied lat/lon.
importFrom utils URLencode importFrom httr GET stop_for_status

censusr	<i>censusr: A package to download Census API data</i>
---------	---

Description

The censusr package provides principally one main function: call_census_api allows users to pass a list of variables and a list of geographies, and returns a data_frame with the requested data.

`get_geo_url`*Construct a geography request string from a FIPS Code*

Description

Construct a geography request string from a FIPS Code

Usage

```
get_geo_url(geoid, allgeos)
```

Arguments

<code>geoid</code>	A character string with a FIPS code, between 2 and 15 digits long.
<code>allgeos</code>	(optional) A string identifying the type of geography for which to collect data within the the requested geoids. Must be one of c('co', 'tr', 'bg', 'bl'). For instance, if <code>allgeos = "bg"</code> , will return all block groups within the given geoids.

Value

A string with the FIPS formatted for an API request.

Index

`aggregate_moe`, [2](#)
`append_geoid`, [3](#)

`call_api_once`, [4](#)
`call_census_api`, [4](#), [4](#)
`call_geolocator`, [5](#)
`call_geolocator_latlon`, [6](#)
`censusr`, [6](#)
`censusr-package (censusr)`, [6](#)

`get_geo_url`, [7](#)