Package ‘censusapi’

October 12, 2022

Version 0.8.0
Title Retrieve Data from the Census APIs
Description A wrapper for the U.S. Census Bureau APIs that returns data frames of Census data and metadata. Available datasets include the Decennial Census, American Community Survey, Small Area Health Insurance Estimates, Small Area Income and Poverty Estimates, Population Estimates and Projections, and more.
URL https://www.hrecht.com/censusapi/,
https://github.com/hrecht/censusapi
BugReports https://github.com/hrecht/censusapi/issues
Depends R (>= 3.0.0)
License GPL-3
LazyData true
RoxygenNote 7.2.1
Imports httr, jsonlite, utils
Suggests knitr, rmarkdown
Encoding UTF-8
NeedsCompilation no
Author Hannah Recht [aut, cre]
Maintainer Hannah Recht <censusapi.rstats@gmail.com>
Repository CRAN
Date/Publication 2022-08-09 15:30:16 UTC

R topics documented:

fips ................................................................. 2
getCensus ......................................................... 2
listCensusApis .................................................. 4
listCensusMetadata .............................................. 5
makeVarlist ...................................................... 7

Index 8
**fips**  
*List of state fips codes - 50 states plus DC*

**Description**

Some small geographies in some Census APIs can only be used under a state hierarchy. This is a list of fips codes that may be looped over to retrieve data for all states.

**Usage**

```r
fips
getCensus
```

**Format**

A list of fips codes for 50 states and the District of Columbia

**Examples**

```r
fips
getCensus
```
getCensus

pscode = NULL,
naics2012 = NULL,
naics2007 = NULL,
naics2002 = NULL,
naics1997 = NULL,
sic = NULL,
...
)

Arguments

name The programmatic name of your dataset, e.g. ‘timeseries/poverty/saipe’ or ‘acs/acs5’. See ‘listCensusApis()’ for options.
vintage Vintage (year) of dataset, e.g. 2014. Not required for timeseries APIs.
key Your Census API key, obtained at https://api.census.gov/data/key_signup.html. This function will default to a ‘CENSUS_KEY’ stored in your .Renviron if available.
vars List of variables to get. Required.
region Geography to get.
regionin Optional hierarchical geography to limit region.
time, year, date, period, monthly Optional arguments used for some time series APIs.
show_call List the underlying API call that was sent to the Census Bureau.
convert_variables Convert likely numeric variables into numeric data. Default is true. If false, results will be characters, which is the type returned by the Census Bureau.
... Other valid arguments to pass to the Census API. Note: the APIs are case sensitive.

Examples

## Not run:
# Get total population and median household income for places (cities, towns, villages)
# in one state from the 5-year ACS.
acs_simple <- getCensus(
  name = "acs/acs5",
  vintage = 2020,
  vars = c("NAME", "B01001_001E", "B19013_001E"),
  region = "place:*",
  regionin = "state:01")
head(acs_simple)

# Get all data from the B19013 variable group.
# This returns estimates as well as margins of error and annotation flags.
acs_group <- getCensus(
name = "acs/acs5",
vintage = 2020,
vars = c("B01001_001E", "group(B19013)",
region = "place:*",
regionin = "state:01")
head(acs_group)

# Retreive 2010 Decennial Census block-level data within a specific tract,
# using the regionin argument to precisely specify the Census tract.
decennial_2010 <- getCensus(
  name = "dec/sf1",
vintage = 2010,
vars = c("NAME","P001001"),
region = "block:*",
regionin = "state:36+county:027+tract:010000")
head(decennial_2010)

# Get poverty rates for children and for people of all ages over time using the
# Small Area Income and Poverty Estimates API
saipe <- getCensus(
  name = "timeseries/poverty/saipe",
vars = c("NAME", "SAEPOVRT0_17_PT", "SAEPOVRTALL_PT"),
region = "state:01",
year = "2000:2019")
head(saipe)

# Get County Business Patterns data for a specific NAICS sector.
cbp_2016 <- getCensus(
  name = "cbp",
vintage = "2016",
vars = c("EMP", "ESTAB", "NAICS2012_TTL", "GEO_TTL"),
region = "state:*",
naiscs2012 = "23")
head(cbp_2016)

## End(Not run)

---

**listCensusApis**  
*Get useful dataset metadata on all available APIs as a data frame*

**Description**

Scrapes https://api.census.gov/data.json and returns a dataframe that includes: title, description, name, vintage, url, dataset type, and other useful fields.

**Usage**

listCensusApis()
### Examples

```r
## Not run:
apis <- listCensusApis()
head(apis)

## End(Not run)
```

---

**listCensusMetadata**  
*Get information about a specific API as a data frame*

---

**Description**

Get information about a specific API as a data frame

**Usage**

```r
listCensusMetadata(
  name,
  vintage = NULL,
  type = "variables",
  group = NULL,
  variable_name = NULL,
  include_values = FALSE
)
```

**Arguments**

- **name**: API programmatic name - e.g. `acs/acs5`. See list of names with `listCensusApis()`.
- **vintage**: Vintage (year) of dataset. Not required for timeseries APIs.
- **type**: Type of metadata to return. Options are: *‘variables’* (default) - list of variable names and descriptions for the dataset. *‘geographies’* - available geographies. *‘groups’* - available variable groups. Not used for all datasets. *‘values’* - encoded value labels for a given variable. Pair with ‘variable_name’. Not used for all datasets.
- **group**: An optional variable group code, used to return metadata for a specific group of variables only. Variable groups are not used for all APIs.
- **variable_name**: A name of a specific variable used to return value labels for that variable. Value labels are not published for all APIs.
- **include_values**: Use with ‘type = "variables"’. Include value metadata for all variables in a dataset if value metadata exists. Default is "FALSE".
## Not run:

# List the variables available in the Small Area Health Insurance Estimates.

```r
sahie_variables <- listCensusMetadata(
  name = "timeseries/healthins/sahie",
  type = "variables"
)
head(sahie_variables)
```

# List the geographies available in the 5-year 2020 American Community Survey.

```r
acs_geographies <- listCensusMetadata(
  name = "acs/acs5",
  vintage = 2020,
  type = "geographies"
)
head(acs_geographies)
```

# List the variable groups available in the 5-year 2020 American Community Survey.

```r
acs_groups <- listCensusMetadata(
  name = "acs/acs5",
  vintage = 2020,
  type = "groups"
)
head(acs_groups)
```

# Create a data dictionary with all variable names and encoded values for
# a microdata API.

```r
cbp_dict <- listCensusMetadata(
  name = "cbp",
  vintage = 2020,
  type = "variables",
  include_values = TRUE)
head(cbp_dict)
```

# List the value labels of the NAICS2017 variable in the 2020 County
# Business Patterns dataset.

```r
cbp_naics_values <- listCensusMetadata(
  name = "cbp",
  vintage = 2020,
  type = "values",
  variable = "NAICS2017")
head(cbp_naics_values)
```

# List of variables that are included in the B17020 group in the
# 5-year American Community Survey.

```r
group_B17020 <- listCensusMetadata(
  name = "acs/acs5",
  vintage = 2017,
  type = "variables",
  group = "B17020")
head(group_B17020)
```

## End(Not run)
**makeVarlist**

Use variable metadata to find variables containing a given string.

**Description**

Return a list of variable names or data frame of variable metadata containing a given string. This can be used create a list of variables to later pass to getCensus, or a data frame documenting variables used in a given project.

**Usage**

```
makeVarlist(name, vintage = NULL, find, varsearch = "all", output = "list")
```

**Arguments**

- `name`: API name - e.g. acs5. See list at https://api.census.gov/data.html
- `vintage`: Year of dataset, e.g. 2014 - not required for timeseries APIs
- `find`: A string to find in the variable metadata
- `varsearch`: Optional argument specifying which fields to search. Default is "all". Options are "all", "name", "label", or "concept".
- `output`: Optional argument, specifying output to "list" or "dataframe". Default is "list".

**Examples**

# Return a list, and then use getCensus function to retrieve those variables

```r
## Not run:
myvars <- makeVarlist(name = "timeseries/poverty/saipe",
                     find = "Ages 0-4",
                     varsearch = "label")
myvars
saipe_dt <- getCensus(name = "timeseries/poverty/saipe",
                      time = 2016,
                      vars = myvars,
                      region = "state:*")
head(saipe_dt)
```

## End(Not run)
Index

* `api`
  * `getCensus`, 2

* `datasets`
  * `fips`, 2

* `metadata`
  * `listCensusApis`, 4
  * `listCensusMetadata`, 5

`fips`, 2

`getCensus`, 2

`listCensusApis`, 4

`listCensusMetadata`, 5

`makeVarlist`, 7