Getting started with censusapi

`censusapi` is a wrapper for the United States Census Bureau's APIs (https://www.census.gov/data/developers/data-sets.html). As of 2017 over 200 Census API endpoints are available, including Decennial Census, American Community Survey, Poverty Statistics, and Population Estimates APIs. This package is designed to let you get data from all of those APIs using the same main function— `getCensus` —and the same syntax for each dataset.

`censusapi` generally uses the APIs' original parameter names so that users can easily transition between Census's documentation and examples and this package. It also includes metadata functions to return data frames of available APIs, variables, and geographies.

API key setup

To use the Census APIs, sign up (https://api.census.gov/data/key_signup.html) for an API key. Then, if you're on a non-shared computer, add your Census API key to your .Renviron profile and call it CENSUS_KEY. `censusapi` will use it by default without any extra work on your part. Within R, run:

```
# Add key to .Renviron
Sys.setenv(CENSUS_KEY=YOURKEYHERE)
# Reload .Renviron
readRenviron("~/.Renviron")
# Check to see that the expected key is output in your R console
Sys.getenv("CENSUS_KEY")
```

In some instances you might not want to put your key in your .Renviron - for example, if you're on a shared school computer. You can always choose to specify your key within `getCensus` instead.

Finding your API

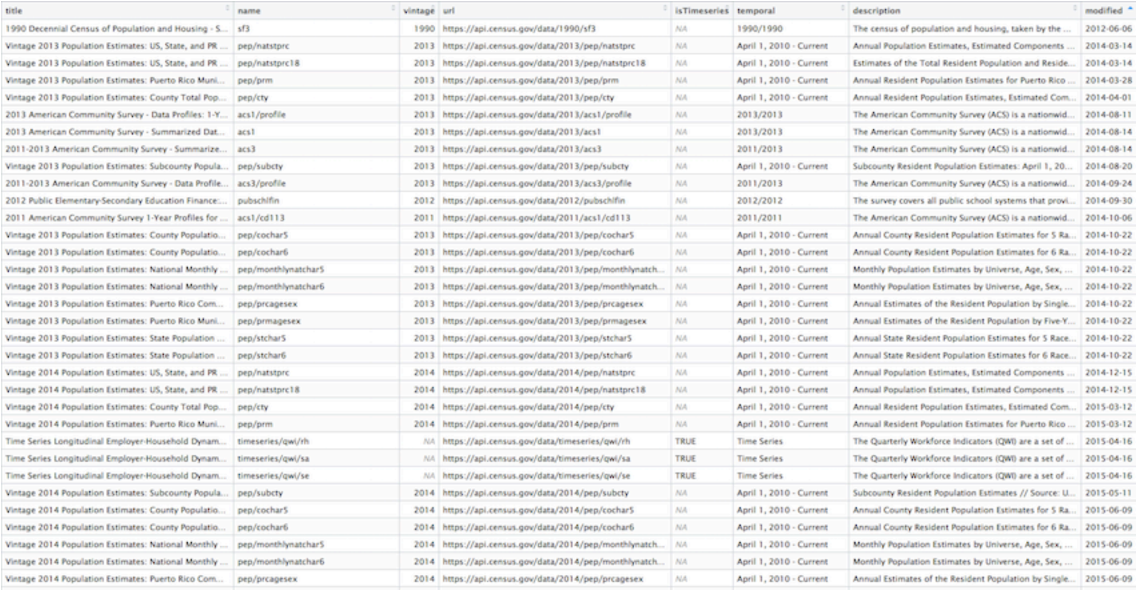
To get started, load the `censusapi` library.

```
library(censusapi)
```

The Census APIs have over 200 endpoints, covering dozens of different datasets.

To see a current table of every available endpoint, run `listCensusApis` :

```
apis <- listCensusApis()
View(apis)
```



title	name	vintage	url	isTimeseries	temporal	description	modified
1990 Decennial Census of Population and Housing - S...	sf3	1990	https://api.census.gov/data/1990/sf3	NA	1990/1990	The census of population and housing, taken by the ...	2012-06-06
Vintage 2013 Population Estimates: US, State, and PR ...	pep/natstprc	2013	https://api.census.gov/data/2013/pep/natstprc	NA	April 1, 2010 - Current	Annual Population Estimates, Estimated Components ...	2014-03-14
Vintage 2013 Population Estimates: US, State, and PR ...	pep/natstprc18	2013	https://api.census.gov/data/2013/pep/natstprc18	NA	April 1, 2010 - Current	Estimates of the Total Resident Population and Reside...	2014-03-14
Vintage 2013 Population Estimates: Puerto Rico Muni...	pep/prm	2013	https://api.census.gov/data/2013/pep/prm	NA	April 1, 2010 - Current	Annual Resident Population Estimates for Puerto Rico ...	2014-03-28
Vintage 2013 Population Estimates: County Total Pop...	pep/cty	2013	https://api.census.gov/data/2013/pep/cty	NA	April 1, 2010 - Current	Annual Resident Population Estimates, Estimated Com...	2014-04-01
2013 American Community Survey - Data Profiles: 1-Y...	acs1/profile	2013	https://api.census.gov/data/2013/acs1/profile	NA	2013/2013	The American Community Survey (ACS) is a nationwid...	2014-08-11
2013 American Community Survey - Summarized Dat...	acs1	2013	https://api.census.gov/data/2013/acs1	NA	2013/2013	The American Community Survey (ACS) is a nationwid...	2014-08-14
2011-2013 American Community Survey - Summarize...	acs3	2013	https://api.census.gov/data/2013/acs3	NA	2011/2013	The American Community Survey (ACS) is a nationwid...	2014-08-14
Vintage 2013 Population Estimates: Subcounty Popula...	pep/subcty	2013	https://api.census.gov/data/2013/pep/subcty	NA	April 1, 2010 - Current	Subcounty Resident Population Estimates: April 1, 20...	2014-08-20
2011-2013 American Community Survey - Data Profile...	acs3/profile	2013	https://api.census.gov/data/2013/acs3/profile	NA	2011/2013	The American Community Survey (ACS) is a nationwid...	2014-09-24
2012 Public Elementary-Secondary Education Finance:...	pubschlfin	2012	https://api.census.gov/data/2012/pubschlfin	NA	2012/2012	The survey covers all public school systems that provi...	2014-09-30
2011 American Community Survey 1-Year Profiles for ...	acs1/cd113	2011	https://api.census.gov/data/2011/acs1/cd113	NA	2011/2011	The American Community Survey (ACS) is a nationwid...	2014-10-06
Vintage 2013 Population Estimates: County Populatio...	pep/cochar5	2013	https://api.census.gov/data/2013/pep/cochar5	NA	April 1, 2010 - Current	Annual County Resident Population Estimates for 5 Ra...	2014-10-22
Vintage 2013 Population Estimates: County Populatio...	pep/cochar6	2013	https://api.census.gov/data/2013/pep/cochar6	NA	April 1, 2010 - Current	Annual County Resident Population Estimates for 6 Ra...	2014-10-22
Vintage 2013 Population Estimates: National Monthly ...	pep/monthlynatchar5	2013	https://api.census.gov/data/2013/pep/monthlynatch...	NA	April 1, 2010 - Current	Monthly Population Estimates by Universe, Age, Sex, ...	2014-10-22
Vintage 2013 Population Estimates: National Monthly ...	pep/monthlynatchar6	2013	https://api.census.gov/data/2013/pep/monthlynatch...	NA	April 1, 2010 - Current	Monthly Population Estimates by Universe, Age, Sex, ...	2014-10-22
Vintage 2013 Population Estimates: Puerto Rico Com...	pep/prcagesex	2013	https://api.census.gov/data/2013/pep/prcagesex	NA	April 1, 2010 - Current	Annual Estimates of the Resident Population by Single...	2014-10-22
Vintage 2013 Population Estimates: Puerto Rico Muni...	pep/prmagesex	2013	https://api.census.gov/data/2013/pep/prmagesex	NA	April 1, 2010 - Current	Annual Estimates of the Resident Population by Five-Y...	2014-10-22
Vintage 2013 Population Estimates: State Population ...	pep/stchar5	2013	https://api.census.gov/data/2013/pep/stchar5	NA	April 1, 2010 - Current	Annual State Resident Population Estimates for 5 Race...	2014-10-22
Vintage 2013 Population Estimates: State Population ...	pep/stchar6	2013	https://api.census.gov/data/2013/pep/stchar6	NA	April 1, 2010 - Current	Annual State Resident Population Estimates for 6 Race...	2014-10-22
Vintage 2014 Population Estimates: US, State, and PR ...	pep/natstprc	2014	https://api.census.gov/data/2014/pep/natstprc	NA	April 1, 2010 - Current	Annual Population Estimates, Estimated Components ...	2014-12-15
Vintage 2014 Population Estimates: US, State, and PR ...	pep/natstprc18	2014	https://api.census.gov/data/2014/pep/natstprc18	NA	April 1, 2010 - Current	Annual Population Estimates, Estimated Components ...	2014-12-15
Vintage 2014 Population Estimates: County Total Pop...	pep/cty	2014	https://api.census.gov/data/2014/pep/cty	NA	April 1, 2010 - Current	Annual Resident Population Estimates, Estimated Com...	2015-03-12
Vintage 2014 Population Estimates: Puerto Rico Muni...	pep/prm	2014	https://api.census.gov/data/2014/pep/prm	NA	April 1, 2010 - Current	Annual Resident Population Estimates for Puerto Rico ...	2015-03-12
Time Series Longitudinal Employer-Household Dynam...	timeseries/qwi/rh	NA	https://api.census.gov/data/timeseries/qwi/rh	TRUE	Time Series	The Quarterly Workforce Indicators (QWI) are a set of ...	2015-04-16
Time Series Longitudinal Employer-Household Dynam...	timeseries/qwi/sa	NA	https://api.census.gov/data/timeseries/qwi/sa	TRUE	Time Series	The Quarterly Workforce Indicators (QWI) are a set of ...	2015-04-16
Time Series Longitudinal Employer-Household Dynam...	timeseries/qwi/se	NA	https://api.census.gov/data/timeseries/qwi/se	TRUE	Time Series	The Quarterly Workforce Indicators (QWI) are a set of ...	2015-04-16
Vintage 2014 Population Estimates: Subcounty Popula...	pep/subcty	2014	https://api.census.gov/data/2014/pep/subcty	NA	April 1, 2010 - Current	Subcounty Resident Population Estimates // Source: U...	2015-05-11
Vintage 2014 Population Estimates: County Populatio...	pep/cochar5	2014	https://api.census.gov/data/2014/pep/cochar5	NA	April 1, 2010 - Current	Annual County Resident Population Estimates for 5 Ra...	2015-06-09
Vintage 2014 Population Estimates: County Populatio...	pep/cochar6	2014	https://api.census.gov/data/2014/pep/cochar6	NA	April 1, 2010 - Current	Annual County Resident Population Estimates for 6 Ra...	2015-06-09
Vintage 2014 Population Estimates: National Monthly ...	pep/monthlynatchar5	2014	https://api.census.gov/data/2014/pep/monthlynatch...	NA	April 1, 2010 - Current	Monthly Population Estimates by Universe, Age, Sex, ...	2015-06-09
Vintage 2014 Population Estimates: National Monthly ...	pep/monthlynatchar6	2014	https://api.census.gov/data/2014/pep/monthlynatch...	NA	April 1, 2010 - Current	Monthly Population Estimates by Universe, Age, Sex, ...	2015-06-09
Vintage 2014 Population Estimates: Puerto Rico Com...	pep/prcagesex	2014	https://api.census.gov/data/2014/pep/prcagesex	NA	April 1, 2010 - Current	Annual Estimates of the Resident Population by Single...	2015-06-09

This returns useful information about each endpoint, including `name` , which you'll need to make your API call.

Using `getCensus`

The main function in `censusapi` is `getCensus` , which makes an API call to a given Census API and returns a data frame of results. Each API has slightly different parameters, but there are always a few required arguments:

- `name` : the name of the API as defined by the Census, like "acs5" or "timeseries/bds/firms"
- `vintage` : the dataset year, generally required for non-timeseries APIs
- `vars` : the list of variable names to get
- `region` : the geography level to return, like state or county

Some APIs have additional required or optional arguments, like `time` , `monthly` , or `period` . Check the specific documentation (https://www.census.gov/data/developers/data-sets.html) for your API to see what options are allowed.

Let's walk through an example getting uninsured rates by income group using the Small Area Health Insurance Estimates API (https://www.census.gov/data/developers/data-sets/Health-Insurance-Statistics.html), which provides detailed annual state-level and county-level estimates of health insurance rates.

Choosing variables

`censusapi` includes a metadata function called `listCensusMetadata` to get information about an API's variable options and geography options. Let's see what variables are available in the SAHIE API:

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

name	label	concept	predicateType	group	limit	required
NIPR_LB90	Number in Demographic Group for Selected Income Range, Lower Bound for 90% Confidence Interval	Uncertainty Measure	int	N/A	0	NA
NIPR_PT	Number in Demographic Group for Selected Income Range, Estimate	Estimate	int	N/A	0	NA
AGECAT	Age Category	Demographic ID	int	N/A	6	default displayed
NIC_PT	Number Insured, Estimate	Estimate	int	N/A	0	NA
GEOID	State+County FIPS Code	Uncertainty Measure	int	N/A	0	NA
STATE	State FIPS Code	Geographic ID	int	N/A	0	NA

We'll use a few of these variables to get uninsured rates by income group:

- `IPRCAT` : Income Poverty Ratio Category
- `IPR DESC` : Income Poverty Ratio Category Description

- `PCTUI_PT` : Percent Uninsured in Demographic Group for Selected Income Range, Estimate
- `NAME` : Name of the geography returned (e.g. state or county name)

Choosing regions

We can also use `listCensusMetadata` to see which geographic levels we can get data for using the SAHIE API.

```
listCensusMetadata(
    name = "timeseries/healthins/sahie",
    type = "geography")
```

name	geoLevelId	limit	referenceDate	requires	wildcard	optionalWithWCFor
us	010	1	2015-01-01	NULL	NULL	NA
county	050	3142	2015-01-01	state	state	state
state	040	52	2015-01-01	NULL	NULL	NA

This API has three geographic levels: `us` , `county` within states, and `state` .

First, using `getCensus` , let's get uninsured rate by income group at the national level for 2017.

```
sahie_national <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "IPRCAT", "IPR_DESC", "PCTUI_PT"),
    region = "us:*",
    time = 2018)
sahie_national
```

time	us	NAME	IPRCAT	IPR_DESC	PCTUI_PT
2018	1	United States	0	All Incomes	10.4
2018	1	United States	1	<= 200% of Poverty	17.3
2018	1	United States	2	<= 250% of Poverty	16.8
2018	1	United States	3	<= 138% of Poverty	17.3
2018	1	United States	4	<= 400% of Poverty	14.5
2018	1	United States	5	138% to 400% of Poverty	13.1

We can also get this data at the state level for every state by changing `region` to `"state:*"` :

```
sahie_states <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "IPRCAT", "IPR_DESC", "PCTUI_PT"),
    region = "state:*",
    time = 2018)
head(sahie_states)
```

time	state	NAME	IPRCAT	IPR_DESC	PCTUI_PT
2018	01	Alabama	0	All Incomes	11.9
2018	01	Alabama	1	<= 200% of Poverty	19.6
2018	01	Alabama	2	<= 250% of Poverty	18.5
2018	01	Alabama	3	<= 138% of Poverty	20.6
2018	01	Alabama	4	<= 400% of Poverty	15.5
2018	01	Alabama	5	138% to 400% of Poverty	12.5

Finally, we can get county-level data. The geography metadata showed that we can choose to get county-level data within states. We'll use `region` to specify county-level results and `regionin` to request data for Alabama and Alaska.

```
sahie_counties <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "IPRCAT", "IPR_DESC", "PCTUI_PT"),
    region = "county:*",
    regionin = "state:01,02",
    time = 2018)
head(sahie_counties, n=12L)
```

time	state	county	NAME	IPRCAT	IPR_DESC	PCTUI_PT
2018	01	001	Autauga County, AL	0	All Incomes	10.0
2018	01	001	Autauga County, AL	1	<= 200% of Poverty	18.7
2018	01	001	Autauga County, AL	2	<= 250% of Poverty	17.3
2018	01	001	Autauga County, AL	3	<= 138% of Poverty	20.0
2018	01	001	Autauga County, AL	4	<= 400% of Poverty	13.6
2018	01	001	Autauga County, AL	5	138% to 400% of Poverty	10.6
2018	01	003	Baldwin County, AL	0	All Incomes	13.2
2018	01	003	Baldwin County, AL	1	<= 200% of Poverty	24.5
2018	01	003	Baldwin County, AL	2	<= 250% of Poverty	22.7
2018	01	003	Baldwin County, AL	3	<= 138% of Poverty	25.7
2018	01	003	Baldwin County, AL	4	<= 400% of Poverty	18.4
2018	01	003	Baldwin County, AL	5	138% to 400% of Poverty	15.3

Because the SAHIE API is a timeseries (as indicated in its name), we can get multiple years of data at once using the `time` argument.

```
sahie_years <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "PCTUI_PT"),
    region = "state:01",
    time = "from 2006 to 2018")
head(sahie_years)
```

time	state	NAME	PCTUI_PT
2006	01	Alabama	15.7
2007	01	Alabama	14.6
2008	01	Alabama	15.3
2009	01	Alabama	15.8
2010	01	Alabama	16.9
2011	01	Alabama	16.6

Advanced topics

This package allows access to the full range of the U.S. Census Bureau's APIs. Where the API allows it, you can specify complicated geographies or filter based on a range of parameters. Each API is a little different, so be sure to read the documentation for the specific API that you're using. Also see more examples in the example masterlist (https://hrecht.github.io/censusapi/articles/example-masterlist.html).

Miscellaneous parameters

Some of the APIs allow complex calls, including specifying a country FIPS code or age. The most commonly used parameters, including `time` , `date` , and `sic` are included as built-in options in `getCensus` , but you can also specify other parameters yourself. (Note: this generally does not apply to the popular American Community Survey and Decennial Census APIs.)

In the SAHIE API, we can filter data by the categorical variables `AGECAT` (age group), `IPRCAT` (income group), `RACECAT` (race) and `SEXCAT` (sex), in addition to geography and time. More information on those variables is available in the online documentation (https://www.census.gov/data/developers/data-sets/Health-Insurance-Statistics.html).

Here's how to get the uninsured rate (`PCTUI_PT`) for non-elderly adults (`AGECAT = 1`) with incomes of 138 to 400% of the poverty line (`IPRCAT = 5`), by race (`RACECAT`) and state.

```
sahie_nonelderly <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "PCTUI_PT", "IPR_DESC", "AGE_DESC", "RACECAT", "RACE_DESC"),
    region = "state:*",
    time = 2018,
    IPRCAT = 5,
    AGECAT = 1)
head(sahie_nonelderly)
```

time	state	NAME	PCTUI_PT	IPR_DESC	AGE_DESC	RACECAT	RACE_DESC	IPRCAT	AGECAT
2018	01	Alabama	16.0	138% to 400% of Poverty	18 to 64 years	0	All Races	5	1
2018	01	Alabama	14.4	138% to 400% of Poverty	18 to 64 years	1	White alone, not Hispanic	5	1
2018	01	Alabama	15.9	138% to 400% of Poverty	18 to 64 years	2	Black alone, not Hispanic	5	1
2018	01	Alabama	37.5	138% to 400% of Poverty	18 to 64 years	3	Hispanic (any race)	5	1
2018	02	Alaska	22.5	138% to 400% of Poverty	18 to 64 years	0	All Races	5	1
2018	02	Alaska	18.5	138% to 400% of Poverty	18 to 64 years	1	White alone, not Hispanic	5	1

Note: data by race is only returned where the population is large enough, so some states will not have rows for some race groups. Here's another example, getting national data from percent uninsured (`PCTUI_PT`) and number uninsured (`NUI_PT`), along with the associated margins of error, by race group and income group for all years.

```
sahie_nonelderly_annual <- getCensus(
    name = "timeseries/healthins/sahie",
    vars = c("NAME", "PCTUI_PT", "PCTUI_MOE", "NUI_PT", "NUI_MOE", "IPRCAT", "IPR_DESC", "AGE_DESC", "RACECAT", "
RACE_DESC"),
    region = "us:*",
    time = "from 2006 to 2018",
    AGECAT = 1)
head(sahie_nonelderly_annual)
```

time	us	NAME	PCTUI_PT	PCTUI_MOE	NUI_PT	NUI_MOE	IPRCAT	IPR_DESC	AGE_DESC	RACECAT	RACE_DESC	AGECAT
2006	1	United States	19.5	0.3	36363986	549708	0	All Incomes	18 to 64 years	0	All Races	1
2006	1	United States	39.5	0.7	19368368	440544	1	<= 200% of Poverty	18 to 64 years	0	All Races	1
2006	1	United States	36.5	0.6	23595529	455801	2	<= 250% of Poverty	18 to 64 years	0	All Races	1
2006	1	United States	13.7	0.3	17094552	364661	0	All Incomes	18 to 64 years	1	White alone, not Hispanic	1
2006	1	United States	32.0	0.8	7846458	277188	1	<= 200% of Poverty	18 to 64 years	1	White alone, not Hispanic	1
2006	1	United States	28.5	0.7	9614431	291480	2	<= 250% of Poverty	18 to 64 years	1	White alone, not Hispanic	1

Other APIs can be filtered too. For example, the International Data Base (https://www.census.gov/data/developers/data-sets/international-database.html) population projections APIs allow you to get data by age and country.

See what variables the IDB 1 year API allows:

```
listCensusMetadata(
    name = "timeseries/idb/1year",
    type = "variables")
```

name	label	concept	predicateType	group	limit	required
POP	Total mid-year population	Total Midyear Population	int	N/A	0	NA

AREA_KM2	Area in square kilometers	Geographic Characteristics	int	N/A	0	NA
FIPS	FIPS country/area code	Geographic Characteristics	string	N/A	0	NA
SEX	Sex	Age and Sex	int	N/A	0	default displayed
YR	Year	Required variable	int	N/A	0	NA
NAME	Country or area name	Geographic Characteristics	string	N/A	0	NA
AGE	Single year of age from 0-100+	Age and Sex	int	N/A	0	true

Here's a simple call getting projected population by age for all countries in 2050.

```
pop_2050 <- getCensus(name = "timeseries/idb/1year",
    vars = c("FIPS", "NAME", "AGE", "POP"),
    time = 2050)
head(pop_2050)
```

time	FIPS	NAME	AGE	POP
2050	AA	Aruba	0	1554
2050	AA	Aruba	1	1554
2050	AA	Aruba	2	1551
2050	AA	Aruba	3	1554
2050	AA	Aruba	4	1550
2050	AA	Aruba	5	1553

But we can make a much more specific call by specifying `FIPS` and `AGE` to get just the population projections for teenagers in Portugal.

```
pop_portugal <- getCensus(
    name = "timeseries/idb/1year",
    vars = c("NAME", "POP"),
    time = 2050,
    FIPS = "PO",
    AGE = "13:19")
pop_portugal
```

time	NAME	POP	FIPS	AGE
2050	Portugal	82014	PO	13
2050	Portugal	82573	PO	14
2050	Portugal	83083	PO	15
2050	Portugal	83540	PO	16
2050	Portugal	83812	PO	17
2050	Portugal	83919	PO	18
2050	Portugal	83880	PO	19

The Quarterly Workforce Indicators APIs (https://www.census.gov/data/developers/data-sets/qwi.html) allow even more specific calls. Here's one example:

```
qwi <- getCensus(
    name = "timeseries/qwi/sa",
    region = "state:02",
    vars = c("Emp", "sex"),
    year = 2012,
    quarter = 1,
    agegrp = "A07",
    ownercode = "A05",
    firmsize = 1,
    seasonadj = "U",
    industry = 21)
qwi
```

Emp	sex	year	quarter	agegrp	ownercode	firmsize	seasonadj	industry	state

52	0	2012	1	A07	A05	1	U	21	02
46	1	2012	1	A07	A05	1	U	21	02
6	2	2012	1	A07	A05	1	U	21	02

Variable groups

For some surveys, particularly the American Community Survey and Decennial Census, you can get many related variables at once using a `group` , defined by the Census Bureau. In some other data tools, like American FactFinder, this idea is referred to as a `table` .

The American Community Survey (ACS) APIs include estimates (variable names ending in “E”), annotations, margins of error, and statistical significance, depending on the data set. Read more on ACS variable types (https://www.census.gov/data/developers/data-sets/acs-5year.html) and annotation symbol meanings on the Census website.

You can retrieve these annotation variables manually, by specifying a list of variables. We’ll get the estimate, margin of error and annotations for median household income in the past 12 months for Census tracts in Alaska.

```
acs_income <- getCensus(
    name = "acs/acs5",
    vintage = 2017,
    vars = c("NAME", "B19013_001E", "B19013_001EA", "B19013_001M", "B19013_001MA"),
    region = "tract:*",
    regionin = "state:02")
head(acs_income)
```

state	county	tract	NAME	B19013_001E	B19013_001EA	B19013_001M	B19013_001MA
02	261	000300	Census Tract 3, Valdez-Cordova Census Area, Alaska	89000	NA	20435	NA
02	122	000600	Census Tract 6, Kenai Peninsula Borough, Alaska	58125	NA	5725	NA
02	122	001100	Census Tract 11, Kenai Peninsula Borough, Alaska	69028	NA	5941	NA
02	261	000100	Census Tract 1, Valdez-Cordova Census Area, Alaska	49076	NA	7165	NA
02	122	000200	Census Tract 2, Kenai Peninsula Borough, Alaska	57694	NA	6526	NA
02	122	000800	Census Tract 8, Kenai Peninsula Borough, Alaska	50904	NA	3723	NA

You can also retrieve also estimates and annotations for a group of variables in one command. Here’s the `group` call for that same table, B19013 (https://api.census.gov/data/2016/acs/acs5/groups/B19013.html).

```
# See descriptions of the variables in group B19013
group_B19013 <- listCensusMetadata(
    name = "acs/acs5",
    vintage = 2017,
    type = "variables",
    group = "B19013")
group_B19013
```

name	label	predicateType	group	limit	predicateOnly	concept
B19013_001MA	Annotation of Margin of Error!!Median household income in the past 12 months (in 2017 inflation-adjusted dollars)	string	B19013	0	TRUE	NA
B19013_001EA	Annotation of Estimate!!Median household income in the past 12 months (in 2017 inflation-adjusted dollars)	string	B19013	0	TRUE	NA
B19013_001E	Estimate!!Median household income in the past 12 months (in 2017 inflation-adjusted dollars)	int	B19013	0	TRUE	MEDIAN HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2017 INFLATION-ADJUSTED DOLLARS)
B19013_001M	Margin of Error!!Median household income in the past 12 months (in 2017 inflation-adjusted dollars)	int	B19013	0	TRUE	MEDIAN HOUSEHOLD INCOME IN THE PAST 12 MONTHS (IN 2017 INFLATION-ADJUSTED DOLLARS)

```
acs_income_group <- getCensus(
    name = "acs/acs5",
    vintage = 2017,
    vars = c("NAME", "group(B19013)"),
    region = "tract:*",
    regionin = "state:02")
```

```
#> Warning in responseFormat(raw): NAs introduced by coercion
```

```
head(acs_income_group)
```

state	county	tract	NAME	GEO_ID	B19013_001E	B19013_001M	NAME_1	B19013_001EA	B19013_001MA
02	261	000300	Census Tract 3, Valdez-Cordova Census Area, Alaska	1400000US02261000300	89000	20435	NA	NA	NA
02	122	000600	Census Tract 6, Kenai Peninsula Borough, Alaska	1400000US02122000600	58125	5725	NA	NA	NA
02	122	001100	Census Tract 11, Kenai Peninsula Borough, Alaska	1400000US02122001100	69028	5941	NA	NA	NA
02	261	000100	Census Tract 1, Valdez-Cordova Census Area, Alaska	1400000US02261000100	49076	7165	NA	NA	NA
02	122	000200	Census Tract 2, Kenai Peninsula Borough, Alaska	1400000US02122000200	57694	6526	NA	NA	NA
02	122	000800	Census Tract 8, Kenai Peninsula Borough, Alaska	1400000US02122000800	50904	3723	NA	NA	NA

Some variable groups contain many related variables and their associated annotations. As an example, we'll get the list of variables included in group B17020 (https://api.census.gov/data/2016/acs/acs5/groups/B17020.html), poverty status by age.

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

name	label	predicateType	group	limit	predicateOnly	concept
B17020_017EA	Annotation of Estimate!!Total!!Income in the past 12 months at or above poverty level!!85 years and over	string	B17020	0	TRUE	NA
B17020_016MA	Annotation of Margin of Error!!Total!!Income in the past 12 months at or above poverty level!!75 to 84 years	string	B17020	0	TRUE	NA
B17020_016EA	Annotation of Estimate!!Total!!Income in the past 12 months at or above poverty level!!75 to 84 years	string	B17020	0	TRUE	NA
B17020_015MA	Annotation of Margin of Error!!Total!!Income in the past 12 months	string	B17020	0	TRUE	NA

B17020_015MA	Annotation of Margin of Error!!Total!!Income in the past 12 months at or above poverty level!!60 to 74 years	string	B17020	0	TRUE	NA
B17020_015EA	Annotation of Estimate!!Total!!Income in the past 12 months at or above poverty level!!60 to 74 years	string	B17020	0	TRUE	NA
B17020_014EA	Annotation of Estimate!!Total!!Income in the past 12 months at or above poverty level!!18 to 59 years	string	B17020	0	TRUE	NA

Advanced geographies

Some geographies, particularly Census tracts and blocks, need to be specified within larger geographies like states and counties. This varies by API endpoint, so make sure to read the documentation for your specific API and run `listCensusMetadata` to see the available geographies.

You may want to get get data for many geographies that require a parent geography. For example, tract-level data from the 2010 Decennial Census can only be requested from one state at a time.

In this example, we use the built in `fips` list of state FIPS codes (https://www.census.gov/library/reference/code-lists/ansi.html) to request tract-level data from each state and join into a single data frame.

```
tracts <- NULL
for (f in fips) {
    stateget <- paste("state:", f, sep="")
    temp <- getCensus(
        name = "dec/sf1",
        vintage = 2010,
        vars = "P001001",
        region = "tract:*",
        regionin = stateget)
    tracts <- rbind(tracts, temp)
}
head(tracts)
```

state	county	tract	P001001
01	001	020100	1912
01	001	020500	10766
01	001	020300	3373
01	001	020400	4386
01	001	020200	2170
01	001	020600	3668

The `regionin` argument of `getCensus` can also be used with a string of nested geographies, as shown below.

The 2010 Decennial Census summary file 1 requires you to specify a state and county to retrieve block-level data. Use `region` to request block level data, and `regionin` to specify the desired state and county.

```
data2010 <- getCensus(
    name = "dec/sf1",
    vintage = 2010,
    vars = "P001001",
    region = "block:*",
    regionin = "state:36+county:027+tract:010000")
head(data2010)
```

state	county	tract	block	P001001
36	027	010000	1000	31
36	027	010000	1011	17
36	027	010000	1028	41
36	027	010000	1001	0
36	027	010000	1031	0
36	027	010000	1002	4

Troubleshooting

The APIs contain hundreds of API endpoints and dozens of datasets, each of which work a little differently. The Census Bureau also makes frequent updates, which unfortunately are not always announced in advance. If you're getting an error message or unexpected results, here are some things to check.

Variables

Use `listCensusMetadata(type = "variables")` on your API to see the table of available variables.

- Occasionally the variable names will change with data updates or API updates. The names may be different from year to year. Capitalization may also be inconsistent between datasets or years.
- The Census APIs are case-sensitive, which means that if the variable name you want is uppercase you'll need to write it uppercase in your request. Most of the APIs use uppercase variable names, but some use lowercase and some even use sentence case.

Geographies

Use `listCensusMetadata(type = "geographies")` on your dataset to check which geographies you can use.

- Each API has its own list of valid geographies and they occasionally change as the Census Bureau makes updates. If a previously available geography isn't available anymore, email cnmp.developers.list@census.gov (mailto:cnmp.developers.list@census.gov) detailing the issue.
- If you're specifying a region by FIPS code, for example `state:01`, make sure to use the full code, padded with 0s if necessary. The APIs did not always enforce this (previously, `state:1` usually worked), but now they do. See the Census reference files (https://www.census.gov/geographies/reference-files/2017/demo/popest/2017-fips.html) for valid FIPS codes.

Unexpected errors

Occasionally you might get the general error message `"There was an error while running your query.  We've logged the error and we'll correct it ASAP.  Sorry for the inconvenience."` This comes from the Census Bureau and could be caused by any number of problems, including server issues. Try rerunning your API call. If that doesn't work and you are requesting a large amount of data, try reducing the amount that you're requesting, for example getting only one state at a time. If you're still having trouble, email cnmp.developers.list@census.gov (mailto:cnmp.developers.list@census.gov). Include in your email the raw API call that's provided in your `getCensus` error message (not your R code) so that they can try to help.

Margins of error and annotations

Read the documentation (https://www.census.gov/data/developers/data-sets/acs-1year/notes-on-acs-estimate-and-annotation-values.html) to understand why you might see values like `-555555555` in your margin of error columns. These values are set by the Census Bureau, not this `censusapi` package.

Other ways to get help

- Open a Github issue (https://github.com/hrecht/censusapi/issues) for bugs or issues with this R package.
- Join the public Census Bureau Slack channel (http://links.govdelivery.com/track?
  type=click&enid=ZWFzPTEmbWFpbGluZ2lkPTIwMTgwOTE3Ljk0OTc1NjkxJm1lc3NhZ2VpZD1NREItUFJELUJVTC0yMDE4MDkxNy45NDk3NTY5MSZkYXRhYr
  eml=gd&utm_medium=email&utm_source=govdelivery) and ask your question in the R or API rooms.
- Email the Census Bureau API team at cnmp.developers.list@census.gov (mailto:cnmp.developers.list@census.gov) for questions relating to the underlying data and APIs.

Additional resources

- Census Data API User Guide (https://www.census.gov/data/developers/guidance/api-user-guide.Overview.html)

Disclaimer

This product uses the Census Bureau Data API but is not endorsed or certified by the Census Bureau.