Package 'usdarnass'

October 12, 2022

Type Package

Title USDA NASS Quick Stats API

Version 0.1.0

Description An alternative for downloading various United States Department of
Agriculture (USDA) data from <https://quickstats.nass.usda.gov/> through R.
You must sign up for an API token from the mentioned website in order for
this package to work.

URL <https://github.com/rdinter/usdarnass>

BugReports <https://github.com/rdinter/usdarnass/issues>

License GPL-3

Encoding UTF-8

LazyData true

Imports httr, jsonlite, methods, utils, readr

RoxygenNote 6.1.1

Suggests knitr, rmarkdown, tidyverse, dplyr, tidyr

VignetteBuilder knitr

NeedsCompilation no

Author Robert Dinterman [cre, aut] (<https://orcid.org/0000-0002-9055-6082>),
Jonathan Eyer [aut]

Maintainer Robert Dinterman <robert.dinterman@gmail.com>

Repository CRAN

Date/Publication 2019-06-21 08:50:03 UTC

R topics documented:

---

nass_count   *Get number of observations from Quick Stats query*

Description

Checks the number of observations that will be returned in a data request. All queries to the Quick Stats are limited to 50,000 observations. This is a helpful function in determining how to structure a data request to fit within the 50,000 limit.

Usage

```
nass_count(source_desc = NULL, sector_desc = NULL, group_desc = NULL,
  commodity_desc = NULL, short_desc = NULL, domain_desc = NULL,
  domaincat_desc = NULL, agg_level_desc = NULL,
  statisticcat_desc = NULL, state_name = NULL, asd_desc = NULL,
  county_name = NULL, region_desc = NULL, zip_5 = NULL,
  watershed_desc = NULL, year = NULL, freq_desc = NULL,
  reference_period_desc = NULL, token = NULL, ...)
```

Arguments

source_desc	"Program" - Source of data ("CENSUS" or "SURVEY"). Census program includes the Census of Ag as well as follow up projects. Survey program includes national, state, and county surveys.
sector_desc	"Sector" - Five high level, broad categories useful to narrow down choices. ("ANIMALS & PRODUCTS", "CROPS", "DEMOGRAPHICS", "ECONOMICS", or "ENVIRONMENTAL")
group_desc	"Group" - Subsets within sector (e.g., under sector_desc = "CROPS", the groups are "FIELD CROPS", "FRUIT & TREE NUTS", "HORTICULTURE", and "VEGETABLES").
commodity_desc	"Commodity" - The primary subject of interest (e.g., "CORN", "CATTLE", "LABOR", "TRACTORS", "OPERATORS").
short_desc	"Data Item" - A concatenation of six columns: commodity_desc, class_desc, prodn_practice_desc, util_practice_desc, statisticcat_desc, and unit_desc.
domain_desc	"Domain" - Generally another characteristic of operations that produce a particular commodity (e.g., "ECONOMIC CLASS", "AREA OPERATED", "NAICS CLASSIFICATION", "SALES"). For chemical usage data, the domain describes the type of chemical applied to the commodity. The domain_desc = "TOTAL" will have no further breakouts; i.e., the data value pertains completely to the short_desc.
domaincat_desc	"Domain Category" - Categories or partitions within a domain (e.g., under domain_desc = "SALES", domain categories include $1,000 TO $9,999, $10,000 TO $19,999, etc).

agg_level_desc	"Geographic Level" - Aggregation level or geographic granularity of the data. ("AGRICULTURAL DISTRICT", "COUNTY", "INTERNATIONAL", "NATIONAL", "REGION : MULTI-STATE", "REGION : SUB-STATE", "STATE", "WATERSHED", or "ZIP CODE")
statisticcat_desc	
	"Category" - The aspect of a commodity being measured (e.g., "AREA HARVESTED", "PRICE RECEIVED", "INVENTORY", "SALES").
state_name	"State" - State full name.
asd_desc	"Ag District" - Ag statistics district name.
county_name	"County" - County name.
region_desc	"Region" - NASS defined geographic entities not readily defined by other standard geographic levels. A region can be a less than a state (SUB-STATE) or a group of states (MULTI-STATE), and may be specific to a commodity.
zip_5	"Zip Code" - US Postal Service 5-digit zip code.
watershed_desc	"Watershed" - Name assigned to the HUC.
year	"Year" - The numeric year of the data and can be either a character or numeric vector. Conditional values are also possible, for example a character vector of ">=1999" of "1999<=" will give years greater than or equal to 1999. Right now the logical values can either be greater/less than or equal to with the logical at either the beginning or end of a string with the year.
freq_desc	"Period Type" - Length of time covered ("ANNUAL", "SEASON", "MONTHLY", "WEEKLY", "POINT IN TIME"). "MONTHLY" often covers more than one month. "POINT IN TIME" is as of a particular day.
reference_period_desc	
	"Period" - The specific time frame, within a freq_desc.
token	API key, default is to use the value stored in .Renviron which is stored from the nass_set_key function. If there is no API key stored in the environment, a character string can be provided.
\dots	Not used.

Value

Number of observations.

Examples

```
## Not run:
# Determine all the observations in NASS
nass_count()

## End(Not run)

## Not run:
# Find the number of observations for Wake County in North Carolina
nass_count(state_name = "NORTH CAROLINA", county_name = "WAKE")

## End(Not run)
```

---

nass_data   *Get data from the Quick Stats query*

Description

Sends query to Quick Stats API from given parameter values. Data request is limited to 50,000 records per the API. Use [nass_count](#) to determine number of records in query.

Usage

```
nass_data(source_desc = NULL, sector_desc = NULL, group_desc = NULL,
  commodity_desc = NULL, short_desc = NULL, domain_desc = NULL,
  domaincat_desc = NULL, agg_level_desc = NULL,
  statisticcat_desc = NULL, state_name = NULL, asd_desc = NULL,
  county_name = NULL, region_desc = NULL, zip_5 = NULL,
  watershed_desc = NULL, year = NULL, freq_desc = NULL,
  reference_period_desc = NULL, token = NULL, format = c("CSV",
  "JSON", "XML"), numeric_vals = FALSE)
```

Arguments

source_desc	"Program" - Source of data ("CENSUS" or "SURVEY"). Census program includes the Census of Ag as well as follow up projects. Survey program includes national, state, and county surveys.
sector_desc	"Sector" - Five high level, broad categories useful to narrow down choices. ("ANIMALS & PRODUCTS", "CROPS", "DEMOGRAPHICS", "ECONOMICS", or "ENVIRONMENTAL")
group_desc	"Group" - Subsets within sector (e.g., under sector_desc = "CROPS", the groups are "FIELD CROPS", "FRUIT & TREE NUTS", "HORTICULTURE", and "VEGETABLES").
commodity_desc	"Commodity" - The primary subject of interest (e.g., "CORN", "CATTLE", "LABOR", "TRACTORS", "OPERATORS").
short_desc	"Data Item" - A concatenation of six columns: commodity_desc, class_desc, prodn_practice_desc, util_practice_desc, statisticcat_desc, and unit_desc.
domain_desc	"Domain" - Generally another characteristic of operations that produce a particular commodity (e.g., "ECONOMIC CLASS", "AREA OPERATED", "NAICS CLASSIFICATION", "SALES"). For chemical usage data, the domain describes the type of chemical applied to the commodity. The domain_desc = "TOTAL" will have no further breakouts; i.e., the data value pertains completely to the short_desc.
domaincat_desc	"Domain Category" - Categories or partitions within a domain (e.g., under domain_desc = "SALES", domain categories include $1,000 TO $9,999, $10,000 TO $19,999, etc).

agg_level_desc	"Geographic Level" - Aggregation level or geographic granularity of the data. ("AGRICULTURAL DISTRICT", "COUNTY", "INTERNATIONAL", "NATIONAL", "REGION : MULTI-STATE", "REGION : SUB-STATE", "STATE", "WATERSHED", or "ZIP CODE")
statisticcat_desc	
	"Category" - The aspect of a commodity being measured (e.g., "AREA HARVESTED", "PRICE RECEIVED", "INVENTORY", "SALES").
state_name	"State" - State full name.
asd_desc	"Ag District" - Ag statistics district name.
county_name	"County" - County name.
region_desc	"Region" - NASS defined geographic entities not readily defined by other standard geographic levels. A region can be a less than a state (SUB-STATE) or a group of states (MULTI-STATE), and may be specific to a commodity.
zip_5	"Zip Code" - US Postal Service 5-digit zip code.
watershed_desc	"Watershed" - Name assigned to the HUC.
year	"Year" - The numeric year of the data and can be either a character or numeric vector. Conditional values are also possible, for example a character vector of ">=1999" of "1999<=" will give years greater than or equal to 1999. Right now the logical values can either be greater/less than or equal to with the logical at either the beginning or end of a string with the year.
freq_desc	"Period Type" - Length of time covered ("ANNUAL", "SEASON", "MONTHLY", "WEEKLY", "POINT IN TIME"). "MONTHLY" often covers more than one month. "POINT IN TIME" is as of a particular day.
reference_period_desc	
	"Period" - The specific time frame, within a freq_desc.
token	API key, default is to use the value stored in `.Renviron` which is stored from the [nass_set_key](#) function. If there is no API key stored in the environment, a character string can be provided.
format	Output format from API call. Defaults to CSV as it is typically the smallest sized call. Other options are JSON and XML but these are not recommended. XML currently not supported.
numeric_vals	Optional to convert the year, Value, and coefficient of variation (CV %) to numerics as opposed to defaulted character values. Default is to FALSE as some Values have a suppression code. Converting to numeric will result in suppressed values to be NA.

Value

A data frame containing query to API.

Examples

```
## Not run:
# Get state values in 2012 for all of the values of agricultural land
```

```
nass_data(agg_level_desc = "STATE", year = "2012",
commodity_desc = "AG LAND", domain_desc = "VALUE")

## End(Not run)

## Not run:
# Get county level values in 2012 for the specific data item
  nass_data(year = 2012, agg_level_desc = "COUNTY",
    short_desc = "AG LAND, INCL BUILDINGS - ASSET VALUE, MEASURED IN $")

## End(Not run)
```

---

nass_param   *Get all possible values of a parameter*

Description

All possible values of a parameter for a given query. Helps to understand the columns of data.frame
from nass_data.

Usage

```
nass_param(param = NULL, source_desc = NULL, sector_desc = NULL,
  group_desc = NULL, commodity_desc = NULL, short_desc = NULL,
  domain_desc = NULL, domaincat_desc = NULL, agg_level_desc = NULL,
  statisticcat_desc = NULL, state_name = NULL, asd_desc = NULL,
  county_name = NULL, region_desc = NULL, zip_5 = NULL,
  watershed_desc = NULL, year = NULL, freq_desc = NULL,
  reference_period_desc = NULL, token = NULL)
```

Arguments

param	A valid parameter value. Available names are: source_desc, sector_desc, group_desc, commodity_desc, short_desc, domain_desc, domaincat_desc, agg_level_desc, statisticcat_desc, state_name, asd_desc, county_name, region_desc, zip_5, watershed_desc, year, freq_desc, and reference_period_desc.
source_desc	"Program" - Source of data ("CENSUS" or "SURVEY"). Census program includes the Census of Ag as well as follow up projects. Survey program includes national, state, and county surveys.
sector_desc	"Sector" - Five high level, broad categories useful to narrow down choices. ("ANIMALS & PRODUCTS", "CROPS", "DEMOGRAPHICS", "ECONOMICS", or "ENVIRONMENTAL")
group_desc	"Group" - Subsets within sector (e.g., under sector_desc = "CROPS", the groups are "FIELD CROPS", "FRUIT & TREE NUTS", "HORTICULTURE", and "VEGETABLES").
commodity_desc	"Commodity" - The primary subject of interest (e.g., "CORN", "CATTLE", "LABOR", "TRACTORS", "OPERATORS").

short_desc	"Data Item" - A concatenation of six columns: commodity_desc, class_desc, prodn_practice_desc, util_practice_desc, statisticcat_desc, and unit_desc.
domain_desc	"Domain" - Generally another characteristic of operations that produce a particular commodity (e.g., "ECONOMIC CLASS", "AREA OPERATED", "NAICS CLASSIFICATION", "SALES"). For chemical usage data, the domain describes the type of chemical applied to the commodity. The domain_desc = "TOTAL" will have no further breakouts; i.e., the data value pertains completely to the short_desc.
domaincat_desc	"Domain Category" - Categories or partitions within a domain (e.g., under domain_desc = "SALES", domain categories include $1,000 TO $9,999, $10,000 TO $19,999, etc).
agg_level_desc	"Geographic Level" - Aggregation level or geographic granularity of the data. ("AGRICULTURAL DISTRICT", "COUNTY", "INTERNATIONAL", "NATIONAL", "REGION : MULTI-STATE", "REGION : SUB-STATE", "STATE", "WATERSHED", or "ZIP CODE")
statisticcat_desc	
	"Category" - The aspect of a commodity being measured (e.g., "AREA HARVESTED", "PRICE RECEIVED", "INVENTORY", "SALES").
state_name	"State" - State full name.
asd_desc	"Ag District" - Ag statistics district name.
county_name	"County" - County name.
region_desc	"Region" - NASS defined geographic entities not readily defined by other standard geographic levels. A region can be a less than a state (SUB-STATE) or a group of states (MULTI-STATE), and may be specific to a commodity.
zip_5	"Zip Code" - US Postal Service 5-digit zip code.
watershed_desc	"Watershed" - Name assigned to the HUC.
year	"Year" - The numeric year of the data and can be either a character or numeric vector. Conditional values are also possible, for example a character vector of ">=1999" of "1999<=" will give years greater than or equal to 1999. Right now the logical values can either be greater/less than or equal to with the logical at either the beginning or end of a string with the year.
freq_desc	"Period Type" - Length of time covered ("ANNUAL", "SEASON", "MONTHLY", "WEEKLY", "POINT IN TIME"). "MONTHLY" often covers more than one month. "POINT IN TIME" is as of a particular day.
reference_period_desc	
	"Period" - The specific time frame, within a freq_desc.
token	API key, default is to use the value stored in .Renviron which is stored from the nass_set_key function. If there is no API key stored in the environment, a character string can be provided.

Value

Character vector of all possible parameter values.

Examples

```
## Not run:
# Return the program sources for data
nass_param("source_desc")

## End(Not run)

## Not run:
# Return the group categories available in the CROPS sector
nass_param("group_desc", sector_desc = "CROPS")

## End(Not run)
```

---

nass_set_key   *Set a Quick Stats API key*

Description

This function will add your Quick Stats API key to your `.Renviron` file so it can be called securely without being stored in your code. After you have installed your key, it can be called any time by typing `Sys.getenv("NASS_KEY")` and can be used in package functions by simply typing NASS_KEY. If you do not have an `.Renviron` file, the function will create on for you. If you already have an `.Renviron` file, the function will append the key to your existing file, while making a backup of your original file for disaster recovery purposes.

Usage

```
nass_set_key(key = NULL, overwrite = FALSE)
```

Arguments

key	The API key provided to you from NASS formatted in quotes. A key can be acquired at https://quickstats.nass.usda.gov/api
overwrite	If this is set to TRUE, it will write an existing NASS_KEY that you already have in your `.Renviron` file.

Examples

```
## Not run:
set_nass_key("abcd01234567890123456789012345678789")
# First time, relead your enviornment so you can use the key without
# restarting R.
readRenviron("~/.Renviron")
# You can check it with:
Sys.getenv("NASS_KEY")
```

```
## End(Not run)

## Not run:
# If you need to overwrite an existing key:
nass_set_key("abcd012345678901234567890123456789", overwrite = TRUE)
# First time, relead your enviornment so you can use the key without
# restarting R.
readRenviron("~/.Renviron")
# You can check it with:
Sys.getenv("NASS_KEY")

## End(Not run)
```

Index