Package ‘blscrapeR’

October 12, 2022

Type Package

Title An API Wrapper for the Bureau of Labor Statistics (BLS)

Version 3.2.2

Description Scrapes various data from <https://www.bls.gov/>. The U.S. Bureau of Labor Statistics is the statistical branch of the United States Department of Labor. The package has additional functions to help parse, analyze and visualize the data.

Depends R (>= 3.5.0)

Imports httr, jsonlite, ggplot2, magrittr, utils, stats, grDevices, dplyr, purrr, tibble, stringr

Suggests testthat, knitr, rmarkdown

License MIT + file LICENSE

URL https://github.com/keberwein/blscrapeR

BugReports https://github.com/keberwein/blscrapeR/issues

LazyData true

VignetteBuilder knitr

RoxygenNote 7.2.1

NeedsCompilation no

Author Kris Eberwein [aut, cre]

Maintainer Kris Eberwein <kris.eberwein@gmail.com>

Repository CRAN

Date/Publication 2022-09-16 22:36:13 UTC

R topics documented:

  bls_api ................................................................. 2
  bls_map_county ...................................................... 4
  bls_map_state ........................................................ 5
  county_map_data ..................................................... 6
  dateCast ............................................................ 8
**bls_api**  

*Basic Request Mechanism for BLS Tables*

**Description**

Return data frame from one or more requests via the US Bureau of Labor Statistics API. Provided arguments are in the form of BLS series ids.

**Usage**

```r
bls_api(
  seriesid,
  startyear = NULL,
  endyear = NULL,
  registrationKey = NULL,
  catalog = FALSE,
  calculations = FALSE,
  annualaverage = FALSE,
  ...
)
```

**Arguments**

- **seriesid**: The BLS id of the series your trying to load. A common format would be 'LAUCN040010000000005'. **WARNING**: All seriesIDs must contain the same time resolution. For example, monthly data sets can not be combined with annual or semi-annual data. If you need help finding seriesIDs, check the BLS website or the BLS Data Finder–links below.

- **startyear**: The first year in your data set.
bls_api

endyear  The last year in your data set.
registrationKey  The API key issued to you from the BLS website.
catalog  Series description information available only for certain data sets.
calculations  Returns year-over-year calculations if set to TRUE.
annualaverage  Returns an annual average if set to TRUE.
...  additional arguments

Value
A tibble from the BLS API.

See Also
https://www.bls.gov/data/
https://beta.bls.gov/dataQuery/search

Examples

## API Version 1.0 R Script Sample Code
## Single Series request
df <- bls_api("LAUCN040010000000005")

## Not run:
## API Version 1.0 R Script Sample Code
## Multiple Series request with date params.
df <- bls_api(c("LAUCN040010000000005", "LAUCN040010000000006"),
startyear = "2010", endyear = "2012")

## API Version 1.0 R Script Sample Code
## Multiple Series request with date params.
df <- bls_api(c("LAUCN040010000000005", "LAUCN040010000000006"),
startyear = "2010", endyear = "2012")

## API Version 2.0 R Script Sample Code
## Multiple Series request with full params allowed by v2.
df <- bls_api(c("LAUCN040010000000005", "LAUCN040010000000006"),
startyear = "2010", endyear = "2012",
registrationKey = "BLS_KEY",
calculations = TRUE, annualaverage = TRUE, catalog = TRUE)

## End(Not run)
Description

Return a ggplot object to render a choropleth map with county outlines. The map files contain 2015 FIPS codes and can be used with any data set containing county and state FIPS codes. They can not be used with the leaflet package but the shape files can be downloaded from the Census website or with the tigris package. See the "Mapping BLS Data" vignette for this package.

Usage

```r
bls_map_county(
  map_data,
  fill_rate = NULL,
  labtitle = NULL,
  stateName = NULL,
  projection = NULL,
  lowFill = "green",
  highFill = "red"
)
```

Arguments

- **map_data**: Dataframe to be used as the map's measures. Usually a result of calls to the `get_bls_county()` or `get_bls_state()` functions, but other dataframes, which include FIPS codes may be used as well.
- **fill_rate**: Column name from the dataframe that you want to use as a fill value, in quotes. NOTE: This argument is mandatory!
- **labtitle**: The main title label for your map passed as a string. The default is no title.
- **stateName**: Optional argument if you only want to map a single state or a group of selected states. The argument accepts state full state names in quotes.
- **projection**: Choices of map projection are "lambert" or "mercator". By default, the function selects Mercator for single states and Lambert for nationwide views.
- **lowFill**: The fill color of the lower values being mapped. The default color is green, but can be changed to any color accepted by `ggplot2::scale_fill_gradient`.
- **highFill**: The fill color of the higher values being mapped. The default color is green, but can be changed to any color accepted by `ggplot2::scale_fill_gradient`.

Value

A ggplot2 object based on information provided by the `map_data` argument.

See Also

[https://cran.r-project.org/package=tigris](https://cran.r-project.org/package=tigris)
Examples

```r
## Not run:
# Download the most current month unemployment statistics on a county level.
df <- get_bls_county()

# Map the unemployment rate by county.
bls_gg <- bls_map_county(map_data = df, fill_rate = "unemployed_rate",
                         labtitle = "Unemployment Rate")
bls_gg

# Map the unemployment rate for Florida and Alabama.

df <- get_bls_county(stateName = c("Florida", "Alabama"))

bls_gg <- bls_map_county(map_data = df, fill_rate = "unemployed_rate",
                         stateName = c("Florida", "Alabama"))

bls_gg
## End(Not run)
```

---

**bls_map_state**

*choropleth mapping of BLS data*

**Description**

Return a ggplot object to render a choropleth map with state outlines. The map files contain 2015 FIPS codes and can be used with any data set containing state FIPS codes. They can not be used with the leaflet package but the shape files can be downloaded from the Census website or with the tigris package. See the "Mapping BLS Data" vignette for this package.

**Usage**

```r
bls_map_state(
  map_data,
  fill_rate = NULL,
  labtitle = NULL,
  lowFill = "green",
  highFill = "red"
)
```

**Arguments**

- `map_data` Dataframe to be used as the map’s measures. Usually a result of calls to the `get_bls_state()` function but other dataframes, which include FIPS codes may be used as well.
fill_rate Column name from the dataframe that you want to use as a fill value.
labtitle The main title label for your map passed as a string. The default is no title.
lowFill The fill color of the lower values being mapped. The default color is green, but can be changed to any color accepted by `ggplot2::scale_fill_gradient`.
highFill The fill color of the higher values being mapped. The default color is green, but can be changed to any color accepted by `ggplot2::scale_fill_gradient`.

Value

A ggplot2 object based on information provided by the map_data argument.

See Also

https://cran.r-project.org/package=tigris

Examples

```r
## Not run:
# Download employment statistics for April 2016.
df <- get_bls_state("April 2016", seasonality = TRUE)

# Map the unemployment rate from data set.
bls_gg <- bls_map_state(map_data = df, fill_rate = "unemployed_rate",
                        labtitle = "Unemployment Rate")
bls_gg
## End(Not run)
```

county_map_data Dataset for mapping U.S. counties with a Mercator projection

Description

A fortified data set that includes U.S. counties and is suitable for making maps with ggplot2. The county FIPS codes and boundary lines from the most recent TIGER release from the U.S. Census Bureau.

- long: County longitude
- lat: County latitude
- order: Polygon order
- hole: hole
- piece: Polygon piece
- id: FIPS Code
A fortified data set that includes U.S. counties and is suitable for making maps with `ggplot2`. The county FIPS codes and boundary lines from the most recent TIGER release from the U.S. Census Bureau.

- **long**: County longitude
- **lat**: County latitude
- **order**: Polygon order
- **hole**: hole
- **piece**: Polygon piece
- **id**: FIPS Code
- **group**: group

**Usage**

```r
data(county_map_merc)
data(county_map_data)
```

**Format**

A data frame with 55,413 rows and 7 variables

**Details**

Dataset with the lat. / long. of county FIPS codes used for mapping.

Built-in dataset for use with the `bls_map_county()` function. To access the data directly, issue the command `data(county_map_merc)`.

**Note**

Last updated 2017-01-26
### dateCast

*Cast a date column to data frame returned by the bls_api() function*

**Description**

A helper function to create a continuous date from Year and Period columns.

**Usage**

```r
dateCast(api_df = NULL, dt_format = NULL)
```

**Arguments**

- `api_df`: The data frame you wish to cast a date column to. Be sure the data frame contains `year` and `period` columns as returned by the `bls_api()` function.
- `dt_format`: A character string containing a valid date format. The default will return the ISO 8601 date format.

**Value**

A tibble from the source `api_df` with an additional date column based on the date format given in `dt_format`.

**Examples**

```r
## Cast a date column to data frame returned by the bls_api() function.
df <- bls_api("LAUCN040010000000005") %>%
dateCast()
```

### get_bls_county

*A function that returns county-level labor statistics*

**Description**

A function to download and format state employment data. Due to limitations in the data source, the function can only return data from the last 12 months. NOTE: Unlike many other BLS data sets, these data are never estimated, meaning the most current data may be as much as 60 days behind the current data. The county data are also never seasonally adjusted.

**Usage**

```r
get_bls_county(date_mth = NULL, stateName = NULL, ...)
```
get_bls_state

Arguments

- **date_mth**: The month you would like data for. Accepts full month names and four-digit year. If NULL, it will return the most recent month in the database.

- **stateName**: is an optional argument if you only want data for certain state(s). The argument is NULL by default and will return data for all 50 states.

... additional arguments

Value

A tibble from the BLS API.

Examples

```r
## Not run:
# Most recent month in the data set.
get_bls_county()

# A specific month
df <- get_bls_county("May 2017")

# Multiple months
df <- get_bls_county(c("April 2017","May 2017"))

# A specific state
df <- get_bls_county(stateName = "Florida")

# Multiple states, multiple months
df<- get_bls_county(date_mth = "April 2017",
                     stateName = c("Florida", "Alabama"))

## End(Not run)
```

get_bls_state

A function that returns state-level labor statistics

Description

A function to download and format state employment data. These data begin on January 1976 to current. NOTE: The most current data will always be at least 30 days behind the current date, and depending on the day of your query, those numbers may be estimates.

Usage

get_bls_state(date_mth = NULL, seasonality = TRUE, ...)

Arguments

- **date_mth**: The month or months you would like data for. Accepts full month names and four-digit year.
- **seasonality**: TRUE or FALSE. The default value is TRUE.
- ... additional arguments

Value

A tibble from the BLS API.

Examples

```r
## Not run:
# Single series
get_bls_state(date_mth = "May 2016", seasonality = TRUE)

# Multiple series
get_bls_state(date_mth = c("April 2016", "May 2016"), seasonality = FALSE)
## End(Not run)
```

---

**inflation_adjust**

*Convert the Value of a US Dollar to a Given Year on or before 1913.*

Description

Returns a data frame that uses data from the Consumer Price Index (All Goods) to convert the value of a US Dollar ([$1.00 USD]) over time.

Usage

```r
inflation_adjust(base_year = NA, ...)
```

Arguments

- **base_year**: A string or integer argument to represent the base year that you would like dollar values converted to. For example, if you want to see the value of a 2007 dollar in 2015, you would select 2015 as a base year and find 2007 in the table.
- ... additional arguments

Value

A tibble from the BLS API.
Examples

```r
## Not run:
## Get historical USD values based on a 2010 dollar.
values <- inflation_adjust(base_year = 2015)
## End(Not run)
```

Description

Return a ggplot object to render a choropleth map with state and/or county outlines. The map files contain 2016 FIPS codes and can be used with any data set containing county and state FIPS codes. They can not be used with the leaflet package but the shape files can be downloaded from the Census website or with the tigris package. See the "Mapping BLS Data" vignette for this package.

Usage

```r
map_bls(
  map_data,
  fill_rate = NULL,
  labtitle = NULL,
  stateName = NULL,
  projection = NULL,
  lowFill = "green",
  highFill = "red",
  ...
)
```

Arguments

- **map_data**: Dataframe to be used as the map’s measures. Usually a result of calls to the `get_bls_county()` or `get_bls_state()` functions, but other dataframes, which include FIPS codes may be used as well.
- **fill_rate**: Column name from the dataframe that you want to use as a fill value, in quotes. NOTE: This argument is mandatory!
- **labtitle**: The main title label for your map passed as a string. The default is no title.
- **stateName**: Optional argument if you only want to map a single state or a group of selected states. The argument accepts state full state names in quotes.
- **projection**: Choices of map projection are "lambert" or "mercator". By default, the function selects Lambert for county data and Mercator for single states. and Lambert for nationwide views.
- **lowFill**: The fill color of the lower values being mapped. The default color is green, but can be changed to any color accepted by `ggplot2::scale_fill_gradient`.
Value

A ggplot2 object based on information provided by the map_data argument.

See Also

https://cran.r-project.org/package=tigris

Examples

## Not run:
# Download the most current month unemployment statistics on a county level.
df <- get_bls_county()

# Map the unemployment rate by county.
bls_gg <- map_bls(map_data = df, fill_rate = "unemployed_rate",
                  labtitle = "Unemployment Rate")
bls_gg

# Map the unemployment rate for Florida and Alabama.
df <- get_bls_county(stateName = c("Florida", "Alabama"))

bls_gg <- map_bls(map_data=df, fill_rate = "unemployed_rate",
                  stateName = c("Florida", "Alabama"))

bls_gg

# Download state employment statistics for April 2016.
df <- get_bls_state("April 2016", seasonality = TRUE)

# Map the unemployment rate from data set.
bls_gg <- map_bls(map_data = df, fill_rate = "unemployed_rate",
                  labtitle = "Unemployment Rate")
bls_gg

## End(Not run)
qcew_api

Description

Return data from the QCEW API. This is separate from the main BLS API and returns quarterly
data sliced by industry, area or size. Industry is identified by NIACS code and area is identified by
FIPS code. A key is not required for the QCEW API.

Usage

qcew_api(
  year = c(format(Sys.Date(), "%Y")),
  qtr = "1",
  slice = NULL,
  sliceCode = NULL,
  ...
)

Arguments

year These data begin in 2012 and go to the most recent complete quarter. The argument can be entered as an integer or a character. The default is 2012.

qtr Quarter: This can be any integer between 1 and 4, or "A" for annual. The argument can be entered as an integer or a character. The default is 1, which returns the first quarter.

slice The slice should be one of the three data slices offered by the API; "industry", "area", or "size."

sliceCode The slice codes depend on what slice you select. For example, if you select the "area" slice, your sliceCode should be a FIPS code. If you select "industry," your sliceCode should be a NIACS code. There are three internal data sets containing acceptable slice codes to help with selections; blscraper::niacs contains industry codes and descriptions, blscraper::area_titles contains FIPS codes and area descriptions, and blscraper::size_titles contains industry size codes. These codes can be used for the sliceCode argument.

... additional arguments

Value

A tibble from the BLS API.

See Also

https://data.bls.gov/cew/doc/access/csv_data_slices.htm

Examples

## Not run:
# A request for the employment levels and wages for NIACS 5112: Software Publishers.
dat <- qcew_api(c(format(Sys.Date(), "%Y")), qtr="1", slice="industry", sliceCode=10)
quick_employed_level  Quick employed level

Description
Returns the employment level. SeriesID: LNS12000000 If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage
quick_employed_level()

Value
A tibble from the BLS API.

Examples
## Not run:
df <- quick_employed_level()

## End(Not run)

quick_employed_rate  Quick employed rate

Description
Returns the "employment to population ratio." SeriesID: LNS12300000 If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage
quick_employed_rate()

Value
A tibble from the BLS API.
**quick_laborForce_level**

**Examples**
```r
## Not run:
df <- quick_employed_rate()

## End(Not run)
```

**Description**
Returns the civilian labor force level. SeriesID: LNS11000000. If you installed a BLS_KEY with the `set_bls_key()` function, it will detect it and use your key. This counts against your daily query limit.

**Usage**
`quick_laborForce_level()`

**Value**
A tibble from the BLS API.

**Examples**
```r
## Not run:
df <- quick_laborForce_level()

## End(Not run)
```

**quick_laborForce_rate** *Quick Civilian Labor Force Rate*

**Description**
Returns the civilian labor force participation rate. SeriesID: LNS11300000. If you installed a BLS_KEY with the `set_bls_key()` function, it will detect it and use your key. This counts against your daily query limit.

**Usage**
`quick_laborForce_rate()`
**quick_nonfarm_employed**

**Value**

A tibble from the BLS API.

**Description**

Returns the Total Nonfarm Payroll Employment, seasonally adjusted. BLS id CES0000000001. If you installed a BLS_KEY with the `set_bls_key()` function, it will detect it and use your key. This counts against your daily query limit.

**Usage**

`quick_nonfarm_employed()`

**Value**

A tibble from the BLS API.

**Examples**

```r
## Not run:
df <- quick_laborForce_rate()

## End(Not run)
```
quick_unemp_level

Quick unemployment level function

Description

Returns the unemployment level. SeriesID: LNS13000000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

quick_unemp_level()

Value

A tibble from the BLS API.

Examples

```r
## Not run:
df <- quick_unemp_level()
## End(Not run)
```

quick_unemp_rate

Quick unemployment rate function

Description

Returns the "official" unemployment rate. That is, seasonally adjusted, 16 year and over, or the "U-3" rate. SeriesID: LNS14000000. If you installed a BLS_KEY with the set_bls_key() function, it will detect it and use your key. This counts against your daily query limit.

Usage

quick_unemp_rate()

Value

A tibble from the BLS API.

Examples

```r
## Not run:
df <- quick_unemp_rate()
## End(Not run)
```
search_ids

Search the internal series_id data set.

Description

Search the internal series_id data set.

Usage

search_ids(keyword = NULL, periodicity_code = NULL, ...)

Arguments

keyword The keyword you want to search. This can be a fuzzy search of multiple keywords. For example "unemployment women".

periodicity_code The period of time of the surveys you are interested in. This is usually monthly, quarterly or annually. You can type full words or beginning letters. For example, periodicity_code = "m" or periodicity_code = "monthly".

... additional arguments

Value

A tibble from the internal data set bls_ids

Examples

## Not run:
# Search for monthly Unemployment Rates for Women
ids <- search_ids(keyword = c("Unemployment Rate", "Women"), periodicity_code = "M")

## End(Not run)

set_bls_key

Install a BLS API Key in Your .Renviron File for Repeated Use

Description

This function will add your BLS 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("BLS_KEY") and can be used in package functions by simply typing BLS_KEY. If you do not have an .Renviron file, the function will create one 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.
state_map_data

Usage

set_bls_key(key = NA, overwrite = NA)

Arguments

key

The API key provided to you from the BLS formatted in quotes. A key can be acquired at https://data.bls.gov/registrationEngine/

overwrite

If this is set to TRUE, it will overwrite an existing BLS_KEY that you already have in your .Renviron file.

Value

No return value. A convenience function used for API Key configuration.

Examples

## Not run:
set_bls_key("111111abc")
# First time, reload your environment so you can use the key without restarting R.
readRenviron("~/.Renviron")
# You can check it with:
Sys.getenv("BLS_KEY")

## End(Not run)

## Not run:
# If you need to overwrite an existing key:
set_bls_key("111111abc", overwrite = TRUE)
# First time, reload your environment so you can use the key without restarting R.
readRenviron("~/.Renviron")
# You can check it with:
Sys.getenv("BLS_KEY")

## End(Not run)

state_map_data

Dataset for mapping U.S. states

Description

A fortified data set that includes U.S. states and is suitable for making maps with ggplot2. The county FIPS codes and boundary lines from the most recent TIGER release from the U.S. Census Bureau.

- long: State longitude
- lat: State latitude
- order: Polygon order
• hole: hole
• piece: Polygon piece
• id: FIPS Code
• group: group

Usage

data(state_map_data)

Format

A data frame with 13,660 rows and 7 variables

Details

Dataset with the lat. / long. of county FIPS codes used for mapping.
Built-in dataset for use with the bls_map_state() function. To access the data directly, issue the command data(state_map_data).

Note

Last updated 2017-01-26

---

urlExists

Description

A utility function to run a tryCatch on a URL.

Usage

urlExists(target)

Arguments

target url

Value

A logical of TRUE or FALSE.
Index

* api
  - bls_api, 2
  - bls_map_county, 4
  - bls_map_state, 5
  - inflation_adjust, 10
  - map_bls, 11
  - qcew_api, 12

* bls
  - bls_api, 2
  - bls_map_county, 4
  - bls_map_state, 5
  - inflation_adjust, 10
  - map_bls, 11
  - qcew_api, 12

* cpi
  - bls_api, 2
  - inflation_adjust, 10
  - qcew_api, 12

* datasets
  - county_map_data, 6
  - state_map_data, 19

* economics
  - bls_api, 2
  - bls_map_county, 4
  - bls_map_state, 5
  - inflation_adjust, 10
  - map_bls, 11
  - qcew_api, 12

* geography
  - bls_map_county, 4
  - bls_map_state, 5
  - map_bls, 11

* geo
  - bls_map_county, 4
  - bls_map_state, 5
  - map_bls, 11

* inflation
  - bls_api, 2
  - inflation_adjust, 10

  - qcew_api, 12

* map
  - bls_map_county, 4
  - bls_map_state, 5
  - map_bls, 11

* nonfarm employed
  - quick_nonfarm-employed, 16

* quick
  - quick_employed_level, 14
  - quick_employed_rate, 14
  - quick_laborForce_level, 15
  - quick_laborForce_rate, 15
  - quick_nonfarm-employed, 16
  - quick_unemp_level, 17
  - quick_unemp_rate, 17

* rate
  - quick_employed_level, 14
  - quick_employed_rate, 14
  - quick_laborForce_level, 15
  - quick_laborForce_rate, 15
  - quick_unemp_level, 17
  - quick_unemp_rate, 17

* unemployment
  - bls_api, 2
  - bls_map_county, 4
  - bls_map_state, 5
  - inflation_adjust, 10
  - map_bls, 11
  - qcew_api, 12
  - quick_employed_level, 14
  - quick_employed_rate, 14
  - quick_laborForce_level, 15
  - quick_laborForce_rate, 15
  - quick_unemp_level, 17
  - quick_unemp_rate, 17

bls_api, 2
bls_map_county, 4
bls_map_state, 5
county_map_data, 6
county_map_merc (county_map_data), 6
dateCast, 8
get_bls_county, 8
get_bls_state, 9
inflation_adjust, 10
map_bls, 11
qcew_api, 12
quick_employed_level, 14
quick_employed_rate, 14
quick_laborForce_level, 15
quick_laborForce_rate, 15
quick_nonfarm_employed, 16
quick_unemp_level, 17
quick_unemp_rate, 17
search_ids, 18
set_bls_key, 18
state_map_data, 19
urlExists, 20