Introduction to blsr

The Bureau of Labor Statistics (BLS) provides publicly available data on different aspects of the U.S. economy.

The blsr package makes getting this data fast and easy:

- It provides simple functions for remotely retrieving data from the most commonly used BLS databases.
- By constraining your options to a few key databases, it cleans the messy data that is extracted from the BLS's API.
- It provides complete cross-functionality with the package blscrapeR so that users can customize their usage if need arises.

This document introduces you to blsr's basic set of download tools. Because blsr is a wrapper for the blscrapeR package (which is a wrapper for the BLS API), users should read the documentation for blscrapeR before proceeding.

Getting an API key

You will want to acquire a free API key from the BLS. The API key increases your daily query limits and expands your data access. For more information on the API key, please see the documentation for blscrapeR.

The BLS databases

The BLS provides data on several different aspects of the U.S. economy. A full list of their databases can be found here.

BLS data items are identified by objects known as series ID strings. In general, a series ID string is a combination of identifiers for:

- The database that contains the data item, e.g. CE for the Current Employment Statistics database.
- The industry and geographic area for which to summarize the data item.
- The type of data item, e.g. the unemployment rate.

Unfortunately, the precise format of series ID strings varies by the underlying database (see here for a list). The blsr package resolves this problem with datbase-specific functions and metadata tools.

Current employment statistics

The Current Employment Statistics (CES) survey produces monthly estimates of employment, hours, and earnings.

CES estimates are available from blsr at the national and state levels across all industries. Estimates span from January 1939 to present day. Seasonally and non-seasonally adjusted are available.

National estimates

National estimates for the CES belong to the CE database. To view the series ID string format for this database, examine the `series_id_map` data.frame within the loaded `ces_national_codes_list` list:

Table 1: Series ID string for CES national database

string_postions	field_name	example_value	example_translation
1-2	Prefix	CE	Current employment statistics
3	Seasonal adjustment	U	Unadjusted

string_postions	field_name	example_value	example_translation
4-11	Industry supersector	00000000	Total nonfarm
12-13	Data type	01	All employees in thousands

The `ces_national_codes_list` list provides possible values for each field that goes into the series ID. For example, the `data_types` data.frame documents the available data items:

Table 2: Data items available for CES national database

data_type_code	description
01	All employees in thousands
02	Average weekly hours of all employees (private sector industries only)
03	Average hourly earnings of all employees (private sector industries only)
11	Average weekly earnings of all employees (private sector industries only)
56	Aggregate weekly hours of all employees in thousands (private sector industries only)
57	Aggregate weekly payrolls of all employees in thousands (private sector industries only)

Note that wage and payroll data can only be run at a seasonally unadjusted level as well.

The `ces_download` function is used to download and clean the data for the series IDs. Feed this function your chosen values for each field of the series ID string:

```r
# Download data for total nonfarm payroll
ces_df = ces_download(bls_key = Sys.getenv("BLS_KEY"),
                      start_year = 2010,
                      end_year = 2015,
                      adjustment = "S",
                      industries = "00000000",
                      data_types = "01")
#> REQUEST_SUCCEEDED

# Inspect output
head(ces_df %>% data.frame())
#>   archive month period      seriesID              variable_name  value
#> 1  201512    12    M12 CES0000000001 All employees in thousands 143125
#> 2  201511    11    M11 CES0000000001 All employees in thousands 142845
#> 3  201510    10    M10 CES0000000001 All employees in thousands 142610
#> 4  201509     9    M09 CES0000000001 All employees in thousands 142271
#> 5  201508     8    M08 CES0000000001 All employees in thousands 142138
#> 6  201507     7    M07 CES0000000001 All employees in thousands 142016
#>         seasonal_code industry_name industry_level private_sector_flag
#> 1 seasonally adjusted Total nonfarm              0                   1
#> 2 seasonally adjusted Total nonfarm              0                   1
#> 3 seasonally adjusted Total nonfarm              0                   1
#> 4 seasonally adjusted Total nonfarm              0                   1
#> 5 seasonally adjusted Total nonfarm              0                   1
#> 6 seasonally adjusted Total nonfarm              0                   1
#>   naics_sector_code naics_supersector_code naics_supersector_name
#> 1                00                     NA                   <NA>
#> 2                00                     NA                   <NA>
#> 3                00                     NA                   <NA>
#> 4                00                     NA                   <NA>
#> 5                    00                      NA                     <NA>
#> 6                    00                      NA                     <NA>
```

The download functions are capable of handling multiple parameter choices for the same fields. By default, it performs a Cartesian of the choices:

```
# Download data for multiple series
ces_df = ces_download(bls_key = Sys.getenv("BLS_KEY"),
                      start_year = 2013,
                      end_year = 2015,
                      adjustment = "S",
                      industries = (ces_national_codes_list$indu_codes %>%
                                      filter(level == 2, private_sector_flag == 1))$industry_code,
                      data_types = c("01", "03", "11"))
```

In some cases you may not want to Cartesian the argument values. In this case, just use the base `bls_download` function and apply the `clean_ces_national` function:

```
# Series IDs to download
series_ids = c("CES0000000001", "CEU0000000001")

# Download the data
ces_df = bls_download(seriesid = series_ids,
                      start_year = 2010,
                      end_year = 2015,
                      bls_key = Sys.getenv("BLS_KEY"))

# Clean the data
ces_df = ces_df %>% clean_ces_national()
```

Note that the BLS API does not allow you to download data on more than 50 series IDs in a single request. The CES download function will prohibit you from doing so.

State estimates

State estimates for the CES belong to the SM database. The series ID string format is within the `ces_state_codes_list` object:

Table 3: Series ID string for CES state database

string_postions	field_name	example_value	example_translation
1-2	Prefix	SM	Current employment statistics for states
3	Seasonal adjustment	U	Unadjusted
4-10	State code	1900000	Iowa
11-18	Industry supersector	00000000	Total nonfarm
19-20	Data type	01	All employees in thousands

The same `ces_download` function can be used for state data by populating the `state` argument:

```
# Download data for total nonfarm payroll
ces_df = ces_download(bls_key = Sys.getenv("BLS_KEY"),
                      start_year = 2010,
                      end_year = 2015,
                      adjustment = "U",
                      industries = "05000000",
                      data_types = c("01", "03", "11"),
                      states = "1900000")
#> REQUEST_SUCCEEDED

# Inspect output
head(ces_df %>% data.frame())
#>   archive month period             seriesID state_name state_id
#> 1  201512    12    M12 SMU19000000500000001       Iowa       IA
#> 2  201511    11    M11 SMU19000000500000001       Iowa       IA
#> 3  201510    10    M10 SMU19000000500000001       Iowa       IA
#> 4  201509     9    M09 SMU19000000500000001       Iowa       IA
#> 5  201508     8    M08 SMU19000000500000001       Iowa       IA
#> 6  201507     7    M07 SMU19000000500000001       Iowa       IA
#>                variable_name  value seasonal_code industry_name
#> 1 All employees in thousands 1312.9    unadjusted Total private
#> 2 All employees in thousands 1317.9    unadjusted Total private
#> 3 All employees in thousands 1317.0    unadjusted Total private
#> 4 All employees in thousands 1312.3    unadjusted Total private
#> 5 All employees in thousands 1318.2    unadjusted Total private
#> 6 All employees in thousands 1320.3    unadjusted Total private
#>   industry_level private_sector_flag naics_sector_code
#> 1              1                   1                05
#> 2              1                   1                05
#> 3              1                   1                05
#> 4              1                   1                05
#> 5              1                   1                05
#> 6              1                   1                05
#>   naics_supersector_code naics_supersector_name
#> 1                     NA                   <NA>
#> 2                     NA                   <NA>
#> 3                     NA                   <NA>
#> 4                     NA                   <NA>
#> 5                     NA                   <NA>
#> 6                     NA                   <NA>
```

Other functionality remains the same as national series. Note that certain data series are available at the national level may not available at the state level from the BLS.

Job openings and labor turnover survey

The Job Openings and Labor Turnover Survey (JOLTS) produces monthly estimates of job openings, hires, quits, layoffs and discharges, and other separations.

JOLTS estimates are available from blsr at the national and regional level across all industries. The data is available via API from December 2000 to present day.

Use the `jolts_codes_list` object to view the series ID string format for the JOLTS database:

Table 4: Series ID string for JOLTS database

string_postions	field_name	example_value	example_translation
1-2	Prefix	JT	Jolts
3	Seasonal adjustment	U	Unadjusted
4-9	Industry	000000	Total nonfarm

string_postions	field_name	example_value	example_translation
10-11	Region	00	Total US
12-13	Data Element	HI	Hires
14	Rate or Level	L	Level - in thousands

For example, to retrive the monthly number of seasonally adjusted non-farm hires, we need to use the series ID `JTS00000000HIL`.

National estimates

Use the `jolts_download` function to download national estimates for JOLTS:

```
# Download the data
jolts_df = jolts_download(bls_key = Sys.getenv("BLS_KEY"),
                          start_year = 2010,
                          end_year = 2015,
                          adjustment = "S",
                          industries = "000000",
                          data_types = "HI",
                          data_levels = "L")
#> REQUEST_SUCCEEDED

# View the data
head(jolts_df)
#> # A tibble: 6 x 15
#>   archive month period seriesID region_name region_code variable_name value
#>     <dbl> <dbl> <chr>  <chr>    <chr>       <chr>       <chr>         <dbl>
#> 1  201512    12 M12    JTS0000~ Total US    00          Hires          5546
#> 2  201511    11 M11    JTS0000~ Total US    00          Hires          5341
#> 3  201510    10 M10    JTS0000~ Total US    00          Hires          5321
#> 4  201509     9 M09    JTS0000~ Total US    00          Hires          5260
#> # ... with 2 more rows, and 7 more variables: seasonal_code <chr>,
#> #   level_name <chr>, industry_name <chr>, industry_level <dbl>,
#> #   private_sector_flag <dbl>, naics_supersector_code <int>,
#> #   naics_supersector_name <chr>
```

Regional estimates

JOLTS produces data for the four geographic regions of the United States. Only the total nonfarm estimates (industry 000000) are available at the regional level.

To download regional data, populate the regions field of the `jolts_download` function:

```
# Download the data
jolts_df = jolts_download(bls_key = Sys.getenv("BLS_KEY"),
                          start_year = 2010,
                          end_year = 2015,
                          adjustment = "S",
                          industries = "000000",
                          data_types = "HI",
                          data_levels = "L",
                          regions = c("MW", "NE", "SO", "WE"))
#> Warning! Only total nonfarm series ID is available for JOLTS regional data. Making adjustments if ne
#> REQUEST_SUCCEEDED
```

Local area unemployment statistics

The Local Area Unemployment Statistics (LAUS) program produces monthly estimates of employment, unemployment, and labor force size.

LAUS estimates are available at the state level from blsr. The data is available from January 1976 to present day.

Use the `laus_codes_list` object to view the series ID string format for LAUS:

Table 5: Series ID string for LAUS database

string_postions	field_name	example_value	example_translation
1-2	Prefix	LA	LAUS
3	Seasonal adjustment	U	Unadjusted
4-18	Area code	ST0100000000000	Alabama statewide
19-20	Data type	03	Unemployment rate

For example, we can retrieve the seasonally adjusted unemployment rate for Alabama using the series ID `LASST010000000000003`.

State estimates

Use the `laus_download` function to download state estimates for the LAUS:

```
# Download the data
laus_df = laus_download(bls_key = Sys.getenv("BLS_KEY"),
                        start_year = 2010,
                        end_year = 2015,
                        adjustment = "S",
                        states = c("ST0100000000000", "ST0200000000000"),
                        data_types = c("03"))
#> REQUEST_SUCCEEDED

# View the data
head(laus_df)
#> # A tibble: 6 x 9
#>   archive month period seriesID state_name state_id variable_name value
#>     <dbl> <dbl> <chr>  <chr>    <chr>      <chr>    <chr>         <dbl>
#> 1  201512    12 M12    LASST01~ Alabama    AL       unemployment~   6  
#> 2  201511    11 M11    LASST01~ Alabama    AL       unemployment~   6  
#> 3  201510    10 M10    LASST01~ Alabama    AL       unemployment~   6  
#> 4  201509     9 M09    LASST01~ Alabama    AL       unemployment~   6.1
#> # ... with 2 more rows, and 1 more variable: seasonal_code <chr>
```

Quarterly census of employment and wages