**Introduction**

[Quandl](https://www.quandl.com/search) is one of the best platforms for finding and downloading financial and economic time series. The collection of free databases is solid and I’ve used it intensively in my research and class material.

But, a couple of things from the native package Quandl always bothered me:

* Multiple data is always returned in the wide (column oriented) format (why??);
* No local caching of data;
* No control for importing error and status;
* Not easy to work within the tidyverse collection of packages

As you suspect, I decided to tackle the problem over the weekend. The result is package GetQuandlData. This is what it does differently:

* It uses the json api (and not the Quandl native function), so that some metadata is also returned;
* The resulting dataframe is always returned in the long format, even for multiple series;
* Users can set custom names for input series. This is very useful when using along ggplot or making tables;
* Uses package memoise to set a local caching system. This means that the second time you ask for a particular time series, it will grab it from your hard drive (and not the internet);
* Always compares the requested dates against dates available in the platform.

**Installation**

# not in CRAN yet (need to test it further)

#install.packages('GetQuandlData')

# from github

devtools::install\_github('msperlin/GetQuandlData')

**Example 01 – Inflation in the US**

Let’s download and plot information about inflation in the US:

library(GetQuandlData)

library(tidyverse)

my\_id <- c('Inflation USA' = 'RATEINF/INFLATION\_USA')

my\_api <- readLines('~/Dropbox/.quandl\_api.txt') # you need your own API (get it at <https://www.quandl.com/sign-up-modal?defaultModal=showSignUp>>)

first\_date <- '2000-01-01'

last\_date <- Sys.Date()

df <- get\_Quandl\_series(id\_in = my\_id,

api\_key = my\_api,

first\_date = first\_date,

last\_date = last\_date,

cache\_folder = tempdir())

glimpse(df)

## Observations: 236

## Variables: 4

## $ series\_name "Inflation USA", "Inflation USA", "Inflation USA", "…

## $ ref\_date 2019-08-31, 2019-07-31, 2019-06-30, 2019-05-31, 201…

## $ value 1.750, 1.811, 1.648, 1.790, 1.996, 1.863, 1.520, 1.5…

## $ id\_quandl "RATEINF/INFLATION\_USA", "RATEINF/INFLATION\_USA", "R…

As you can see, the data is in the long format. Let’s plot it:

p <- ggplot(df, aes(x = ref\_date, y = value/100)) +

geom\_col() +

labs(y = 'Inflation (%)',

x = '',

title = 'Inflation in the US') +

scale\_y\_continuous(labels = scales::percent)

p

Beautiful!

**Example 02 – Inflation for many countries**

Next, lets have a look into a more realistic case, where we need inflation data for several countries:

First, we need to see what are the available datasets from database RATEINF:

library(GetQuandlData)

library(tidyverse)

db\_id <- 'RATEINF'

my\_api <- readLines('~/Dropbox/.quandl\_api.txt') # you need your own API

df <- get\_database\_info(db\_id, my\_api)

knitr::kable(df)

| **code** | **name** | **description** | **refreshed\_at** | **from\_date** | **to\_date** | **quandl\_code** | **quandl\_db** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CPI\_ARG | Consumer Price Index – Argentina | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:58 | 1988-01-31 | 2013-12-31 | RATEINF/CPI\_ARG | RATEINF |
| CPI\_AUS | Consumer Price Index – Australia | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1948-09-30 | 2019-06-30 | RATEINF/CPI\_AUS | RATEINF |
| CPI\_CAN | Consumer Price Index – Canada | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1989-01-31 | 2019-08-31 | RATEINF/CPI\_CAN | RATEINF |
| CPI\_CHE | Consumer Price Index – Switzerland | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:58 | 1983-01-31 | 2019-08-31 | RATEINF/CPI\_CHE | RATEINF |
| CPI\_DEU | Consumer Price Index – Germany | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1991-01-31 | 2019-08-31 | RATEINF/CPI\_DEU | RATEINF |
| CPI\_EUR | Consumer Price Index – Euro Area | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1990-01-31 | 2019-08-31 | RATEINF/CPI\_EUR | RATEINF |
| CPI\_FRA | Consumer Price Index – France | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1990-01-31 | 2019-08-31 | RATEINF/CPI\_FRA | RATEINF |
| CPI\_GBR | Consumer Price Index – UK | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:58 | 1988-01-31 | 2019-08-31 | RATEINF/CPI\_GBR | RATEINF |
| CPI\_ITA | Consumer Price Index – Italy | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 2001-01-31 | 2019-08-31 | RATEINF/CPI\_ITA | RATEINF |
| CPI\_JPN | Consumer Price Index – Japan | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1970-01-31 | 2019-08-31 | RATEINF/CPI\_JPN | RATEINF |
| CPI\_NZL | Consumer Price Index – New Zealand | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1988-03-31 | 2019-06-30 | RATEINF/CPI\_NZL | RATEINF |
| CPI\_RUS | Consumer Price Index – Russia | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1995-01-31 | 2019-07-31 | RATEINF/CPI\_RUS | RATEINF |
| CPI\_USA | Consumer Price Index – USA | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1913-01-31 | 2019-08-31 | RATEINF/CPI\_USA | RATEINF |
| INFLATION\_ARG | Inflation YOY – Argentina | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:58 | 1989-01-31 | 2013-12-31 | RATEINF/INFLATION\_ARG | RATEINF |
| INFLATION\_AUS | Inflation YOY – Australia | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1949-03-31 | 2019-06-30 | RATEINF/INFLATION\_AUS | RATEINF |
| INFLATION\_CAN | Inflation YOY – Canada | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1990-01-31 | 2019-08-31 | RATEINF/INFLATION\_CAN | RATEINF |
| INFLATION\_CHE | Inflation YOY – Switzerland | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1984-01-31 | 2019-08-31 | RATEINF/INFLATION\_CHE | RATEINF |
| INFLATION\_DEU | Inflation YOY – Germany | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1992-01-31 | 2019-08-31 | RATEINF/INFLATION\_DEU | RATEINF |
| INFLATION\_EUR | Inflation YOY – Euro Area | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1991-01-31 | 2019-08-31 | RATEINF/INFLATION\_EUR | RATEINF |
| INFLATION\_FRA | Inflation YOY – France | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1991-01-31 | 2019-08-31 | RATEINF/INFLATION\_FRA | RATEINF |
| INFLATION\_GBR | Inflation YOY – UK | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1989-01-31 | 2019-08-31 | RATEINF/INFLATION\_GBR | RATEINF |
| INFLATION\_ITA | Inflation YOY – Italy | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 2002-01-31 | 2019-08-31 | RATEINF/INFLATION\_ITA | RATEINF |
| INFLATION\_JPN | Inflation YOY – Japan | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1971-01-31 | 2019-08-31 | RATEINF/INFLATION\_JPN | RATEINF |
| INFLATION\_NZL | Inflation YOY – New Zealand | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 2001-03-31 | 2019-06-30 | RATEINF/INFLATION\_NZL | RATEINF |
| INFLATION\_RUS | Inflation YOY – Russia | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1996-01-31 | 2019-07-31 | RATEINF/INFLATION\_RUS | RATEINF |
| INFLATION\_USA | Inflation YOY – USA | Please visit <http://www.rateinflation.com/inflation-information/calculate-inflation> for more information. | 2019-09-28 02:19:59 | 1914-01-31 | 2019-08-31 | RATEINF/INFLATION\_USA | RATEINF |

Nice. Now we only need to filter the series with YOY inflation:

idx <- stringr::str\_detect(df$name, 'Inflation YOY')

df\_series <- df[idx, ]

and grab the data:

my\_id <- df\_series$quandl\_code

names(my\_id) <- df\_series$name

first\_date <- '2010-01-01'

last\_date <- Sys.Date()

df\_inflation <- get\_Quandl\_series(id\_in = my\_id,

api\_key = my\_api,

first\_date = first\_date,

last\_date = last\_date)

glimpse(df\_inflation)

## Observations: 897

## Variables: 4

## $ series\_name "Inflation YOY - Argentina", "Inflation YOY - Argent…

## $ ref\_date 2013-12-31, 2013-11-30, 2013-10-31, 2013-09-30, 201…

## $ value 10.95, 10.54, 10.55, 10.49, 10.55, 10.61, 10.46, 10.…

## $ id\_quandl "RATEINF/INFLATION\_ARG", "RATEINF/INFLATION\_ARG", "R…

And, finally, an elegant plot:

p <- ggplot(df\_inflation, aes(x = ref\_date, y = value/100)) +

geom\_col() +

labs(y = 'Inflation (%)',

x = '',

title = 'Inflation in the World',

subtitle = paste0(first\_date, ' to ', last\_date)) +

scale\_y\_continuous(labels = scales::percent) +

facet\_wrap(~series\_name)