**Introduction**

Every company traded in the US stock market must report its quarterly and yearly documents to the SEC and the public in general. This includes its accounting statements (10-K, 10-K) and any other corporate event that is relevant to investors.

[Edgar](https://www.sec.gov/edgar/searchedgar/companysearch.html) is the interface where we can search for a company’s filling information. By looking up a company’s CIK code, one can find all previous filling information. A complete list of available forms can be found in this [link](https://www.sec.gov/forms).

Package GetEdgarData allows the user import the financial documents from such fillings directly into R. Unlike other packages, the information is not taken from the filling’s xml files, but the [structured datasets](https://www.sec.gov/dera/data) at the DERA (Division of Economic and Risk Analysis) section . This means we can import a large amount of structured financial data very quickly. The downside is that the available data starts at 2009.

Like many other packages I’ve wrote for data grabbing, the queries are saved locally using package memoise. This means that the second time you ask for a particular year of data, the function will load a local copy, and will not download the data again.

Both new packages, GetEdgarData and GetQuandlData ([blog post](https://www.msperlin.com/blog/post/2019-10-01-new-package-getquandldata/)) are going to be part of the second edition of my book “Analyzing Financial Data with R” (see first edition [here](https://www.msperlin.com/pafdR/)). My expectation is to publish the new book in early 2020.

**Installation**

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

#install.packages('GetEdgarData')

# from github

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

**Example 01 – Apples Quarterly Net Profit**

The first step in using GetEdgarData is finding information about available companies:

library(GetEdgarData)

library(tidyverse)

my\_year <- 2018

type\_form <- '10-K'

df\_info <- get\_info\_companies(years = my\_year,

type\_data = 'yearly',

type\_form = type\_form)

glimpse(df\_info)

## Observations: 450

## Variables: 13

## $ current\_name "AIR PRODUCTS & CHEMICALS INC /DE/", "ALICO INC…

## $ former\_name NA, "ALICO LAND DEVELOPMENT CO", "SKYWORKS SOLU…

## $ change\_date\_name NA, 19740219, 20020627, NA, 19730319, 19920703,…

## $ sic\_code 2810, 100, 3674, 3674, 3674, 3590, 1311, 3841, …

## $ country "US", "US", "US", "US", "US", "US", "US", "US",…

## $ state "PA", "FL", "MA", "MA", "CA", "WI", "HI", "NJ",…

## $ city "ALLENTOWN", "FT. MYERS,", "WOBURN", "NORWOOD",…

## $ cik 2969, 3545, 4127, 6281, 6951, 6955, 10048, 1079…

## $ id\_file "0000002969-18-000044", "0000003545-18-000108",…

## $ form "10-K", "10-K", "10-K", "10-K", "10-K", "10-K",…

## $ year 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018,…

## $ quarter "FY", "FY", "FY", "FY", "FY", "Q4", "FY", "FY",…

## $ sic\_desc "INDUSTRIAL INORGANIC CHEMICALS", "AGRICULTURAL…

We find information about 450 companies for the 10-K documents in the year of 2018. Digging deeper we find that the official name of Apple is ‘APPLE INC’. Let’s use it to download the financial information since 2009.

my\_company <- 'APPLE INC'

my\_years <- 2009:2018

type\_data <- 'quarterly'

df\_fin\_reports <- get\_edgar\_fin\_data(companies = my\_company,

years = my\_years,

type\_data = type\_data)

glimpse(df\_fin\_reports)

## Observations: 752

## Variables: 17

## $ current\_name "APPLE INC", "APPLE INC", "APPLE INC", "APPLE I…

## $ former\_name "APPLE COMPUTER INC", "APPLE COMPUTER INC", "AP…

## $ change\_date\_name 19970808, 19970808, 19970808, 19970808, 1997080…

## $ sic\_code 3571, 3571, 3571, 3571, 3571, 3571, 3571, 3571,…

## $ cik 320193, 320193, 320193, 320193, 320193, 320193,…

## $ id\_file "0000320193-18-000145", "0000320193-18-000145",…

## $ form "10-K", "10-K", "10-K", "10-K", "10-K", "10-K",…

## $ year 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018,…

## $ quarter "FY", "FY", "FY", "FY", "FY", "FY", "FY", "FY",…

## $ tag "NetIncomeLoss", "NetIncomeLoss", "NetIncomeLos…

## $ version "us-gaap/2018", "us-gaap/2018", "us-gaap/2018",…

## $ ref\_date 2018-03-31, 2018-06-30, 2018-09-30, 2018-03-31…

## $ unit\_ref "USD", "USD", "USD", "USD", "USD", "USD", "USD"…

## $ value\_ref 1.3822e+10, 1.1519e+10, 1.4125e+10, 2.3422e+10,…

## $ qtrs 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…

## $ coreg NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,…

## $ sic\_desc "ELECTRONIC COMPUTERS", "ELECTRONIC COMPUTERS",…

And now we filter for the net income (id tag = ‘NetIncomeLoss’) and plot the resulting dataframe:

net\_income <- df\_fin\_reports %>%

filter(tag == 'NetIncomeLoss')

p <- ggplot(net\_income,

aes(x = ref\_date, y = value\_ref)) +

geom\_col() +

labs(title = 'APPLE Quarterly Net Income (10-Q)',

subtitle = paste0(min(my\_years), ' - ', max(my\_years)),

x = '',

y = 'Net Income ($)',

caption = paste0('Data from EDGAR ', '\n',

'Downloaded with package GetEdgarData') )

print(p)

**Example 02 – Quarterly Net Profit of Many Companies**

The package is really handy for fetching information for many companies. This is due to the fact that the SEC/DERA stores data of all companies by year and the package creates a local cache of the resulting data. This means that, by fetching data for one company, we indirectly have information for all companies.

Let’s see an example by selecting four random companies and creating the same previous graph:

set.seed(5)

my\_companies <- sample(df\_info$current\_name, 4)

my\_years <- 2009:2018

type\_data <- 'quarterly'

net\_income <- get\_edgar\_fin\_data(companies = my\_companies,

years = my\_years,

type\_data = type\_data) %>%

filter(tag == 'NetIncomeLoss')

p <- ggplot(net\_income,

aes(x = ref\_date, y = value\_ref)) +

geom\_col() + facet\_wrap(~current\_name, scales = 'free') +

labs(title = 'Quarterly Net Income for Four Random companies',

subtitle = paste0(min(my\_years), ' - ', max(my\_years)),

x = '',

y = 'Net Income ($)',

caption = paste0('Data from EDGAR ', '\n',

'Downloaded in R with package GetEdgarData') )

print(p)