Tracking R&D spending by 700 Listed US Pharma Companies - Part 2

redwallanalytics.com/2020/02/18/tracking-r-d-spending-by-700-listed-us-pharma-companies/

```
# Re-load data previously stored for purposes of this blog post
pharma <-
  fread("~/Desktop/David/Projects/xbrl_investment/data/pharma_inc.csv")
```

Introduction

In A Walk Though of Accessing Financial Statements with XBRL in R - Part 1, we went through the first steps of pulling XBRL data for a single company from Edgar into R. Although an improvement over manual plugging of numbers into a Excel, there is still a way to go to having clean comparable numbers, and several of challenges were discussed. It is still not clear if XBRL will unseat old school style, manual spreadsheet analysis any time soon. The strength of using analytic software generally comes from looking at scale over cross sections and time for patterns, which might otherwise be missed.

In this post, we are going to take our analysis a step further using Financial Modeling Prep, which maintains a free API of updated and standardized financial statements for all US companies (among other data). It appears that Financial Modeling Prep is also working from Edgar, but does a bit of cleaning and standardizing. We also found that Top Foreign Stocks maintains lists of most stock groups.

We were debating the extent to which the pharma industry invests in R&D. This seemed like a subject which we could easily explore using our new XBRL tools, and so that is what we are going to do in this post.

Load Data

First, we downloaded the NYSE Major Pharmaceutical and NASDAQ Biotechnology lists from Top Foreign Stocks and stored the xlsx files on our disc. We could have just as easily scraped directly from the website, but chose to store the data in this case.

```
# Majors
major <-
  read_excel(
    "~/Desktop/David/Projects/xbrl_investment/data/major-pharma-NYSE-Jan-2020.xlsx"
  )

# Biotech list
biotech <-
  read_excel(
    "~/Desktop/David/Projects/xbrl_investment/data/biotech-NASDAQ-Feb-11-2020.xlsx"
  )

tickers <-
  c(major$Ticker, biotech$Ticker)

tickers[1:10]
```

```
 [1] "ABT"  "ABBV" "AGN"  "AMRX" "RCUS" "AZN"  "BHC"  "BHVN" "BMY"  "BMY"
```

Data Collection and Cleaning Functions

Next, we built two helper functions, first called get_sector to download the data from financialmodelingprep.com, and second called convert_dt to convert to a data.table, then clean up names and variable types.

```
# Build URL to call financialmodelingprep API for the income statement of each ticker
# Call the json from the financialmodelingprep API
# Add 5-10 second delay so as to be respectful guests

get_sector <- function(ticker){

  # Build url for that ticker with the income statement API
  url <- "https://financialmodelingprep.com/api/v3/financials/income-statement/"
  company <- paste0(url,ticker)

  # Try in case there is an issue with the data for a particular company
  income_list <- try(fromJSON(company))

  # Stagger requests
  delay <- 5:10
  wait <- sample(delay, replace = TRUE)
  Sys.sleep(wait)

  # Return income_list
  income_list
}
```

```
# Convert the list item into a data.table
# Convert variable types and clean names

convert_dt <- function(list) {

  # Take list$financials into a data.frame, then a data.table
  income_df <- list$financials
  income_dt <- setDT(income_df)

  # Convert all but first col to numeric
  num <- 2:ncol(income_dt)
  income_dt[,(num):=lapply(.SD, as.numeric),.SDcols=num]

  # Clean names with janitor package
  income_dt <- janitor::clean_names(income_dt)

  # Return data.table
  income_dt
}
```

Bulk Collection

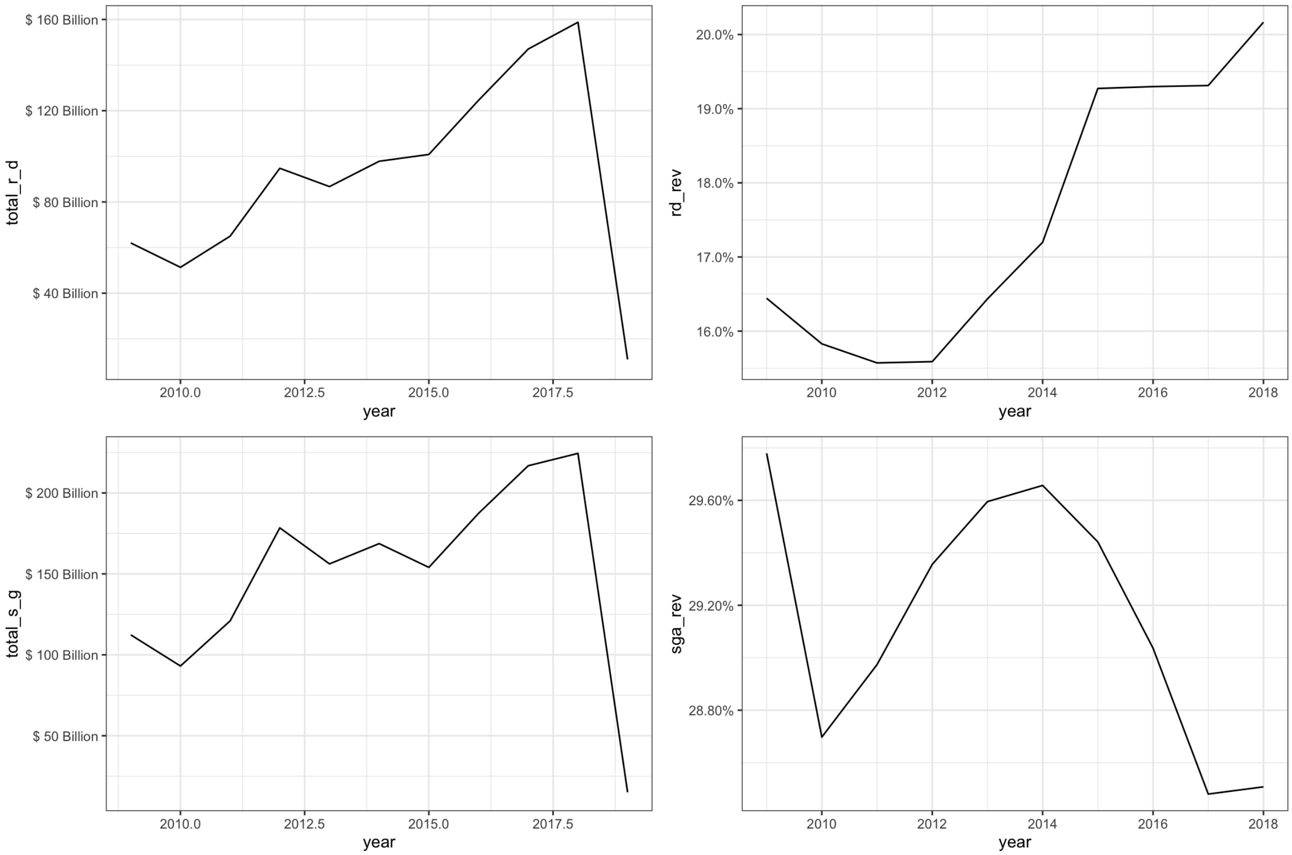
We then used our get_sector function to pull all available income statements for the ~700 companies by ticker from financialmodelingprep.com. This took about an hour to run, and income statements for about ~20 of the companies failed to download. Remember, companies merge, are bought out and go bankrupt over the long term. It is likely that we are not completely accurately reflecting corporate actions, mergers, reclassification's, etc.

```
# This chunk was run previously and the resulting parma data.table
# was saved for the purposes of this blog post

# Apply get_sector function to query financialmodelingprep API for each ticker
pharma_sector <-
  lapply(tickers, get_sector)

# Name list for ticker
names(pharma_sector) <- tickers

# Drop cases where the ticker was not available
pharma_sector <-
  pharma_sector[lengths(pharma_sector) == 2]

# Apply convert_dt function
pharma_dt <-
  lapply(pharma_sector, convert_dt)

# Merge all data.tables and add column name with tickers
pharma <-
  rbindlist(pharma_dt, use.names = TRUE, idcol = "ticker")
```

Charting the R&D and SG&A Averages

We can see in the first chart that the total spending on R&D for the group has risen by over 3x, and R&D-to-sales has risen by about 1/3 since 2010. The charts show a dip in 2019, because only about 10% of the companies have released their 10-K's for 2019 when we gathered the data last week. The bottom chart shows SG&A-to-sales which is slightly larger and has much more stable over the period moving in a surprisingly tight 1% band.

```
#Our transformation function
scaleFUN <- function(x)
  paste("$", x / 1000000000, "Billion")

# R&D totals
p <-
  pharma[, .(year = year(as.Date(date)), r_d_expenses, revenue)][
    ][, .(
      total_rev = sum(revenue, na.rm = TRUE),
      total_r_d = sum(r_d_expenses, na.rm = TRUE
      )),
      by = year][
    ][, ggplot(.SD, aes(year, total_r_d)) +
        geom_line() +
        scale_y_continuous(labels = scaleFUN) +
        theme_bw()
      ]

# R&D relative to sales
p1 <-
  pharma[, .(
    year = year(as.Date(date)),
    r_d_expenses,
    revenue
    )][
    ][, .(
      rd_rev = sum(r_d_expenses, na.rm = TRUE) / sum(revenue, na.rm = TRUE)),
      by = year][
    ][year %in% c(2009:2018),
      ggplot(.SD,aes(year,rd_rev)) +
        geom_line() +
        scale_y_continuous(labels = scales::percent) +
        theme_bw()
      ]

# SG&A totaals
p2 <-
  pharma[, .(
    year = year(as.Date(date)),
    sg_a_expense,
    revenue
    )][, .(
    total_rev = sum(revenue, na.rm = TRUE),
    total_s_g = sum(sg_a_expense, na.rm = TRUE)
  ), by = year][
    ][, ggplot(.SD, aes(year, total_s_g)) +
        geom_line() +
        scale_y_continuous(labels = scaleFUN) +
        theme_bw()
      ]

# SG&A relative to sales
p3 <-
  pharma[, .(
    year = year(as.Date(date)),
    sg_a_expense,
    revenue)][
    ][, .(
      sga_rev = sum(sg_a_expense, na.rm = TRUE) / sum(revenue, na.rm = TRUE)),
      by = year][
    ][year %in% c(2009:2018),
      ggplot(.SD, aes(year, sga_rev)) +
        geom_line() +
        scale_y_continuous(labels = scales::percent) +
        theme_bw()]

p + p1 +
  p2 + p3
```



```
# Clean up
rm(list=ls()[str_detect(ls(), "^p\\d|^p$")])
```

Drilling Down to Company R&D Efficiency

Averages can be misleading, so to make it a more interesting visualization, let’s look at R&D-to-sales by company for the first five years versus sales growth over the period. We will take out companies which had zero sales and R&D at the beginning of the period, because those threw off the chart scales. Of our 683 companies in the data set during the period, this cut our pool to 265. Then, we took out companies which grew from very low bases and those losing most of their sales, further reducing our universe to 146. This number in itself says a lot. The sales of the group doubled over the period, but less than 1/4 of the companies where even in the data at the beginning of the period, so there is a lot of churn, and the companies which were there for the whole period had R&D of only 5% of sales and grew just 63%.

The chart below shows these companies with the color and size reflected in ending sales so big companies have bigger circles. As we have often done, it is possible to hover to see the company details. The x-axis is log scale, but the majority of companies are spending below the 20% average. The big names Gilead, Merck, Bauch Health and Allergan were standouts, with moderate R&D and rapid sales growth. There were a lot of small companies which spent a lot and got nowhere, and many that did very well. There was also a cluster of very big companies which spent and didn’t grow much or even lost revenue.

```
# R&D / sales 2008-2013
rd_sales <-
  pharma[year(as.Date(date)) %in% c(2008:2013),
   .(rd_sales =
       sum(r_d_expenses, na.rm = TRUE) / sum(revenue, na.rm = TRUE)),
   ticker][
     rd_sales>0 &
       rd_sales<10
     ]

# Tickers with data for that period
#tickers <- rd_sales$ticker

# Sales for tickers active during full period
sales_year <-
  pharma[,
    .(revenue,
      year = year(as.Date(date)),
      ticker)]

# Mean annual revenues 2009-2011
sales_early <-
  sales_year[!is.na(revenue),
    .SD[year %in% c(2009:2011),
        .(ticker, revenue, .N), ticker]][
          ][N >=3][
            ][, .(
              rev_early = mean(revenue, na.rm = TRUE)
              ), ticker]

# Mean annual revenues 2016-2018
sales_late <-
  sales_year[!is.na(revenue),
    .SD[year %in% c(2016:2018),
        .(ticker,revenue,.N),ticker]][
          ][N>=3][
            ][, .(
              rev_late = mean(revenue, na.rm = TRUE)
              ), ticker]

# Join early and late mean annual revenue run rate, and calculate growth
sales_growth <-
  sales_late[sales_early, on = "ticker"][
    ][, .(growth = rev_late / rev_early - 1,
          ticker,
          rev_late
          )]

# Join sales_growth and rd_sales data
combined <-
  rd_sales[
    sales_growth, on="ticker"][
      ][growth<10 &
          growth > -1]
```

```
# ggplot two vectors
p <-
  combined[,
    ggplot(.SD,
           aes(
             rd_sales,
             growth,
             text = paste("Ticker: ", ticker),
             color = rev_late,
             size = rev_late
           )) +
      geom_point() +
      scale_x_log10(labels = scales::percent) +
      scale_y_continuous(labels = scales::percent) +
      theme_bw() +
      labs(
        title = "R&D Efficiency by Company - 2009-2018",
        caption = "Source:  Edgar",
        y = "Approx. Annual Growth over Period",
        x = "R&D-to-sales at Beginning"
      )]

# Plotly
plotly::ggplotly(p)

# Clean up
rm(list = ls()[str_detect(ls(), "sales|^p$")])
```

Conclusions

This was a warm-up for the next phase in Part 3 of the series. Now that we know how to collect financial statement date in bulk for a large number of companies, we will try to collect as many companies as possible for as long as possible, and hopefully including quarterly results. Then, we will use specific balance sheet and cash conversion metrics to mine for potential problems. The next post will take some time to deliver!