Stocks

Chelsea Ursaner

9/11/2018

### Setup

### Exploring tidyquant - <https://cran.r-project.org/web/packages/tidyquant/vignettes/TQ01-core-functions-in-tidyquant.html>

# looking at key ratios for Tesla  
# tq\_get lets you pull prices, financials, etc.  
tq\_get\_options()

## [1] "stock.prices" "stock.prices.google" "stock.prices.japan"   
## [4] "financials" "key.ratios" "dividends"   
## [7] "splits" "economic.data" "exchange.rates"   
## [10] "metal.prices" "quandl" "quandl.datatable"   
## [13] "alphavantager" "rblpapi"

# pulling key ratios returns a two-column dataset. The 'data' column is a list  
tesla\_ratios\_all <- tq\_get("TSLA", get = "key.ratios")  
tesla\_ratios\_all

## # A tibble: 7 x 2  
## section data   
## <chr> <list>   
## 1 Financials <tibble [150 × 5]>  
## 2 Profitability <tibble [170 × 5]>  
## 3 Growth <tibble [160 × 5]>  
## 4 Cash Flow <tibble [50 × 5]>   
## 5 Financial Health <tibble [240 × 5]>  
## 6 Efficiency Ratios <tibble [80 × 5]>   
## 7 Valuation Ratios <tibble [40 × 5]>

## Inspecting lists and practicing purrr

# Working with lists  
tesla\_ratios\_list <- tesla\_ratios\_all$data # this saves the list column as a new object - a list of 7  
tesla\_Financial <- tesla\_ratios\_all$data[1] # this creates a list of just the first sub section - a list of 1  
tesla\_Financial2 <- tesla\_ratios\_all$data[[1]] # this unnests the list. A data frame - 150 rows and 5 columns  
  
# inspecting a list element  
# look at the variable names in the raw data  
names(tesla\_Financial2)

## [1] "sub.section" "group" "category" "date" "value"

# save it as a factor  
tesla\_Financial2$sub.section <- as.factor(tesla\_Financial2$sub.section)  
# look at the unique values in sub section  
levels(tesla\_Financial2$sub.section)

## [1] "Financials"

# the ratios in Financial are not split into sub-sections  
  
# inspecting Growth ratios  
tesla\_Growth <- tesla\_ratios\_all$data[[2]]  
levels(as.factor(tesla\_Growth$sub.section))

## [1] "Margin of Sales %" "Profitability"

levels(as.factor(tesla\_Growth$category)) # see all of the ratios

## [1] "Asset Turnover (Average)" "COGS"   
## [3] "EBT Margin" "Financial Leverage (Average)"  
## [5] "Gross Margin" "Interest Coverage"   
## [7] "Net Int Inc & Other" "Net Margin %"   
## [9] "Operating Margin" "Other"   
## [11] "R&D" "Return on Assets %"   
## [13] "Return on Equity %" "Return on Invested Capital %"  
## [15] "Revenue" "SG&A"   
## [17] "Tax Rate %"

glimpse(tesla\_Growth) # see how the data is formatted

## Observations: 170  
## Variables: 5  
## $ sub.section <chr> "Margin of Sales %", "Margin of Sales %", "Margin ...  
## $ group <dbl> 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 17, 17, 17...  
## $ category <chr> "Revenue", "Revenue", "Revenue", "Revenue", "Reven...  
## $ date <date> 2008-12-01, 2009-12-01, 2010-12-01, 2011-12-01, 2...  
## $ value <dbl> 100.00, 100.00, 100.00, 100.00, 100.00, 100.00, 10...

# Creating named lists  
# Step 1 - do it for one  
tesla\_Growth\_levels <- levels(as.factor(tesla\_Growth$sub.section))  
  
# Step 2 - make it a recipe  
# I want to get a list of the distinct ratios for each section  
# first try with just map  
subsection\_levels <- map(tesla\_ratios\_all$data, ~levels(as.factor(.x$sub.section))) #yay!!!  
  
# Main list I want - all categories of ratios  
tesla\_ratios\_all$categories <- map(tesla\_ratios\_all$data, ~levels(as.factor(.x$category)))   
tesla\_ratios\_all # now it's a list column

## # A tibble: 7 x 3  
## section data categories  
## <chr> <list> <list>   
## 1 Financials <tibble [150 × 5]> <chr [15]>  
## 2 Profitability <tibble [170 × 5]> <chr [17]>  
## 3 Growth <tibble [160 × 5]> <chr [4]>   
## 4 Cash Flow <tibble [50 × 5]> <chr [5]>   
## 5 Financial Health <tibble [240 × 5]> <chr [24]>  
## 6 Efficiency Ratios <tibble [80 × 5]> <chr [8]>   
## 7 Valuation Ratios <tibble [40 × 5]> <chr [4]>

# make a new named list using the section names   
categories <- map(tesla\_ratios\_all$data, ~levels(as.factor(.x$category))) %>%  
 set\_names(tesla\_ratios\_all$section)   
# YAYY!!! So much easier than I realized  
  
# export list to look at  
print(categories[1:7])

## $Financials  
## [1] "Book Value Per Share \* USD" "Cap Spending USD Mil"   
## [3] "Dividends USD" "Earnings Per Share USD"   
## [5] "Free Cash Flow Per Share \* USD" "Free Cash Flow USD Mil"   
## [7] "Gross Margin %" "Net Income USD Mil"   
## [9] "Operating Cash Flow USD Mil" "Operating Income USD Mil"   
## [11] "Operating Margin %" "Payout Ratio % \*"   
## [13] "Revenue USD Mil" "Shares Mil"   
## [15] "Working Capital USD Mil"   
##   
## $Profitability  
## [1] "Asset Turnover (Average)" "COGS"   
## [3] "EBT Margin" "Financial Leverage (Average)"  
## [5] "Gross Margin" "Interest Coverage"   
## [7] "Net Int Inc & Other" "Net Margin %"   
## [9] "Operating Margin" "Other"   
## [11] "R&D" "Return on Assets %"   
## [13] "Return on Equity %" "Return on Invested Capital %"  
## [15] "Revenue" "SG&A"   
## [17] "Tax Rate %"   
##   
## $Growth  
## [1] "10-Year Average" "3-Year Average" "5-Year Average" "Year over Year"   
##   
## $`Cash Flow`  
## [1] "Cap Ex as a % of Sales" "Free Cash Flow Growth % YOY"   
## [3] "Free Cash Flow/Net Income" "Free Cash Flow/Sales %"   
## [5] "Operating Cash Flow Growth % YOY"  
##   
## $`Financial Health`  
## [1] "Accounts Payable" "Accounts Receivable"   
## [3] "Accrued Liabilities" "Cash & Short-Term Investments"  
## [5] "Current Ratio" "Debt/Equity"   
## [7] "Financial Leverage" "Intangibles"   
## [9] "Inventory" "Long-Term Debt"   
## [11] "Net PP&E" "Other Current Assets"   
## [13] "Other Long-Term Assets" "Other Long-Term Liabilities"   
## [15] "Other Short-Term Liabilities" "Quick Ratio"   
## [17] "Short-Term Debt" "Taxes Payable"   
## [19] "Total Assets" "Total Current Assets"   
## [21] "Total Current Liabilities" "Total Liabilities"   
## [23] "Total Liabilities & Equity" "Total Stockholders' Equity"   
##   
## $`Efficiency Ratios`  
## [1] "Asset Turnover" "Cash Conversion Cycle"   
## [3] "Days Inventory" "Days Sales Outstanding"  
## [5] "Fixed Assets Turnover" "Inventory Turnover"   
## [7] "Payables Period" "Receivables Turnover"   
##   
## $`Valuation Ratios`  
## [1] "Price to Book" "Price to Cash Flow" "Price to Earnings"   
## [4] "Price to Sales"

### More practice - making a tibble

# Pulling multiple stocks from tidyquant  
test <- tq\_get(c("TSLA", "DATA"), get = "key.ratios")  
# This results in a tibble with the stock repeated 7 times for each section  
names(test)

## [1] "symbol" "section" "data"

# Making a tibble  
stocks\_tbl <- tibble(  
 name = levels(as.factor(test$symbol)),  
 Ratios = "this shall be a list?")

### Test some code - pulling and visualizing indicators for TSLA

# Pull the indicators I care about for one stock - TSLA  
# Pull an entire section  
tesla\_ratios <- tesla\_ratios\_all %>%  
 filter(section == "Valuation Ratios") %>%  
 unnest()   
  
# Visualize the different ratios within that section   
tesla\_ratios %>%  
 ggplot(aes(x = date, y = value)) +  
 geom\_line(aes(col = factor(category, levels = c("Price to Earnings", "Price to Cash Flow", "Price to Book", "Price to Sales"))),   
 size = 1) +   
 labs(title = "10-Year Historical Valuation Ratios for TSLA", x = "",   
 y = "", col = "") +  
 theme\_tq() +  
 scale\_color\_tq()

## Applying purrr

### Make my Portfolio Dataset with All the Data

# Pull key ratios   
# specify the stock - this will be the .x  
stock = "TSLA"  
  
key\_ratios <- tq\_get(stock, get = "key.ratios") # results in a tibble

### Run first recipe - pulling all the tidyquant information for multiple stocks

# Read TD Ameritrade stocks as a list  
stocks <- (str\_split("AMZN  
ANSS  
ASCMA  
BDFFX  
BEXFX  
C  
CS  
DAL  
DATA  
DDD  
GLIBA  
KGFHY  
KROTY  
KW  
LEXEA  
LGF.B  
LSXMA  
LSXMK  
LTRPA  
MANH  
PATI  
PRMTX  
RSTAY  
SMCI  
SSP  
SSYS  
SU  
TM  
TSLA  
UA  
UAA  
WCN", pattern="\n"))  
  
# make a smaller subset  
stocks\_subset <- head(stocks[[1]])  
glimpse(stocks)

## List of 1  
## $ : chr [1:32] "AMZN" "ANSS" "ASCMA" "BDFFX" ...

# turn it into a recipe - ex. map(people, ~ length(.x$starships))  
# remember I can hover over map to see the arguments. map(.x, .f)  
# for the [list, vector, column, dataframe!.. hence .x], do [.f]   
# map(stocks\_subset, ~ tq\_get(.x, "key\_ratios"))  
# test my recipe!  
key\_ratios2 <- map(stocks\_subset, ~ tq\_get(.x, "key\_ratios")) # results in a list of six  
  
# inspect my new list!   
key\_ratios2[1:2] #look at the first two list items

## [[1]]  
## # A tibble: 7 x 2  
## section data   
## <chr> <list>   
## 1 Financials <tibble [150 × 5]>  
## 2 Profitability <tibble [170 × 5]>  
## 3 Growth <tibble [160 × 5]>  
## 4 Cash Flow <tibble [50 × 5]>   
## 5 Financial Health <tibble [240 × 5]>  
## 6 Efficiency Ratios <tibble [80 × 5]>   
## 7 Valuation Ratios <tibble [40 × 5]>   
##   
## [[2]]  
## # A tibble: 7 x 2  
## section data   
## <chr> <list>   
## 1 Financials <tibble [150 × 5]>  
## 2 Profitability <tibble [170 × 5]>  
## 3 Growth <tibble [160 × 5]>  
## 4 Cash Flow <tibble [50 × 5]>   
## 5 Financial Health <tibble [240 × 5]>  
## 6 Efficiency Ratios <tibble [80 × 5]>   
## 7 Valuation Ratios <tibble [40 × 5]>

# not helpful  
# let's name it something that makes sense. What did I do/what was I going for with this list?  
my\_portfolio\_subset <- key\_ratios2 %>%  
 set\_names(stocks\_subset)  
# That worked! my\_portfolio\_subset is now a named list of 6 stocks  
# Each stock is a tibble   
# Each tibble has a list column called data that is a data frame of all the ratios over time

### Make and run second recipe - Extract a key metric from each stock - Return on Investment

# start with one - TSLA  
# 1. extract a filtered dataframe of all the ROIs over time  
tesla\_ROI <- as.data.frame(tesla\_ratios\_all$data[2]) %>%  
 filter(category == "Return on Invested Capital %") %>%  
 select(-1,-2) %>%  
 rename(key\_ratio = category)  
   
glimpse(levels(as.factor(tesla\_ROI$category)))  
  
# Take the average   
tesla\_avg\_ROI <- mean(tesla\_ROI$value, na.rm=T)  
# Save it as a percentage  
tesla\_avg\_ROI <- round((mean(tesla\_ROI$value, na.rm=T))/100,2)  
  
# condense it  
avg\_ROI <- round(tesla\_avg\_ROI/100,2)  
  
# Make my recipe  
# map(.x, ~f) - for each stock, calculate its return on invested capital  
my\_portfolio\_Profitability <- map(my\_portfolio\_subset, ~ (.x$data[[2]]))   
  
my\_portfolio\_ROI <- map(my\_portfolio\_Profitability, ~filter(.x, category == "Return on Invested Capital %"))  
  
my\_portfolio\_avg\_ROI <- map(my\_portfolio\_ROI, ~round((mean(.x$value, na.rm=T))/100,2))  
my\_portfolio\_avg\_ROI

### Make third recipe - Pull key indicators and compile them into a tibble

# Now I have a list of stocks in the format I want it to look like.   
# Time to make the data what I want - extract key indicators  
  
# Start with one - TSLA   
tesla\_ratios\_all  
  
## Indicator group 1 - Financials  
# Share Price, Revenue, and Earnings per Share most recent values  
tesla\_i1 <- tesla\_ratios\_all %>%  
 filter(section == "Financials") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 filter(group %in% c(1, 6, 10)) %>%   
 select(category, value)   
tesla\_i1$value <- as.numeric(as.character(tesla\_i1$value))  
  
## Next section - Profitability  
# Bunch of indicators  
tesla\_i2 <- tesla\_ratios\_all %>%  
 filter(section == "Profitability") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 arrange(group) %>%  
 filter(group %in% c(18, 20, 22, 25, 26, 27:31)) %>%   
 select(category, value)  
  
# Next section- growth. All the indicators!  
tesla\_i3 <- tesla\_ratios\_all %>%  
 filter(section == "Growth") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 select(category, value)  
  
# Last group of indicators- Valuation Ratio  
# Just Price to Earnings for now  
tesla\_i4 <- tesla\_ratios\_all %>%  
 filter(section == "Valuation Ratios") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 filter(group == 86) %>%   
 select(category, value)  
  
# Check price to earnings just for fun  
tesla\_i1[1,2]/tesla\_i1[2,2]  
tesla\_i4[1,2] # Hm.. doesn't match. Oh well  
  
# Make final tibble for TSLA  
TSLA <- rbind(tesla\_i1, tesla\_i2, tesla\_i3, tesla\_i4)  
TSLA$category <- as.character(TSLA$category)  
TSLA$value <- as.numeric(TSLA$value)  
TSLA

### Compile tibbles into a named list - My Indicators

# Make my recipe  
  
# Fist for one section  
test <- map(my\_portfolio\_subset, ~ (.x %>%  
 filter(section == "Financials") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 filter(group %in% c(1, 6, 10)) %>%   
 select(category, value) %>%  
 mutate(Indicator = as.character(category), Value = as.numeric(value)) %>%  
 select(Indicator, Value)))  
glimpse(test[1])  
  
# Now for the whole thing  
my\_indicators <- map(my\_portfolio\_subset, ~ rbind((.x %>%  
 filter(section == "Financials") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 filter(group %in% c(1, 6, 10)) %>%   
 select(category, value)),  
 (.x %>%  
 filter(section == "Profitability") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 arrange(group) %>%  
 filter(group %in% c(18, 20, 22, 25, 26, 27:31)) %>%   
 select(category, value)),  
 (.x %>%  
 filter(section == "Growth") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 select(category, value)),  
 (.x %>%  
 filter(section == "Valuation Ratios") %>%  
 unnest() %>%   
 filter(!is.na(date)) %>%   
 group\_by (category) %>%  
 slice(which.max(date)) %>%  
 filter(group == 86) %>%   
 select(category, value))  
))

## Done! View List

print(my\_indicators)