Scraping ESG Scores

Getting Company and Mutual Fund ESG Scores from Yahoo Finance

Scott Burstein

2021-03-04

Step 1: Preparation

Load Packages

```
# uncomment as necessary:
#install.packages("tidyverse")
#install.packages("urltools")
#install.packages("httr")
#install.packages("robotstxt")

library(tidyverse)
library(urltools)
library(httr)
library(robotstxt)

# Load rvest/stringr/dplyr/tibble packages for node use [Lines 117-125]:
library(rvest)
library(stringr)
library(dplyr)
library(tibble)

# Load quantmod package for market cap calculation [Lines 247-252]:
#install.packages("quantmod")
library(quantmod)
```

Write Helper Functions to Extract Data

Data Parsing Function

```
fun_parse <- function(xpath, xmldoc = page.i) {
  x <- xmldoc %>%
    html_nodes(xpath = xpath) %>%
    html_text(trim = TRUE)
  if (length(x) == 0 & xpath == '//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/div[2]/div[2]/di
    return("None")
  }
```

```
  if (grepl("% AUM", x)) {
    return(as.numeric(sub("% AUM", "", sub("based on ", "", x))) / 100)
  }
  if (!grepl("\\d", x)) {
    return(trimws(x))
  } else {
    if (grepl("percentile", x)) {
      return(x %>% str_replace_all("[^0-9\\.]", "") %>% as.numeric() / 100)
    } else {
      if (grepl("updated on", x)) {
        r <- sub("Last updated on ", "", x)
        r <- paste(unlist(strsplit(r, "/"))[2], unlist(strsplit(r, "/"))[1], sep = "-")
        return(anytime::anydate(r))
      } else {
        return(as.numeric(x))
      }
    }
  }
}
```

Yahoo "Product Involvement Areas" Helper Function

```
fun_lists <- function() {
  x <- page.i %>%
    html_nodes(xpath = '//*[@id="Col2-3-InvolvementAreas-Proxy"]/section/table') %>%
    html_table() %>%
    data.frame()
  n <- sum(grepl("Yes", x[, 2]))
  if (n == 0) return(NA)
  if (n == 1) return(x[grep("Yes", x[, 2]), 1])
  if (n >= 2) return(list(x[grep("Yes", x[, 2]), 1]))
}
```

Wrapper Function for robots.txt - paths_allowed() function

```
fun_robots <- function(url = link.i) {
  base_url <- paste0(url_parse(url)$scheme, "://", domain(url))
  paths_allowed(
    paths = sub(base_url, "", link.i),
    domain = domain(url),
    bot = "*"
  )
}
```

Get Default User Agent

```
httr:::default_ua()
```

```
## [1] "libcurl/7.64.1 r-curl/4.3 httr/1.4.2"
```

```
## [1] "libcurl/7.64.1 r-curl/4.3 httr/1.4.2"
```

Establish Custom User Agent String Variable

```
var_agent <- "Scott Burstein (scott.burstein@duke.edu). Doing academic research."
```

Step 2: Create Data Tables

Create Companies Data Table

```
# Note: ^GSPC is the symbol/ticker for the S&P 500 Index
wiki_link = "https://en.wikipedia.org/wiki/List_of_S%26P_500_companies"
dat_stocks <- read_html(wiki_link) %>%
  html_nodes("table[id='constituents']") %>%
  html_table() %>%
  data.frame() %>%
  as_tibble()
```

Inspect Current Column Names

```
colnames(dat_stocks)
```

```
## [1] "Symbol"                "Security"              "SEC.filings"
## [4] "GICS.Sector"           "GICS.Sub.Industry"     "Headquarters.Location"
## [7] "Date.first.added"      "CIK"                   "Founded"
```

Rename Columns, Data Cleaning, Etc.

```
# rename columns
colnames(dat_stocks) <- c("ticker", "company", "filings", "sector", "industry", "location", "added", "c

# select columns
dat_stocks <- dat_stocks[, c("ticker", "company", "sector", "industry")]

# rename tickers
dat_stocks$ticker <- gsub("[.]", "-", dat_stocks$ticker)
```

Inspect Data Again

```r
head(dat_stocks, 5)
```

```
## # A tibble: 5 x 4
##   ticker company             sector                industry
##   <chr>  <chr>               <chr>                 <chr>
## 1 MMM    3M Company          Industrials           Industrial Conglomerates
## 2 ABT    Abbott Laboratories Health Care           Health Care Equipment
## 3 ABBV   AbbVie Inc.         Health Care           Pharmaceuticals
## 4 ABMD   Abiomed             Health Care           Health Care Equipment
## 5 ACN    Accenture           Information Technolo~ IT Consulting & Other Servic~
```

Create Placeholder Columns for ESG Data (acquired below)

```r
dat_stocks$esgRating    <- as.character(NA) # ESG Rating
dat_stocks$esgScore.tot <- as.integer(NA)   # ESG Score (Total/Overall)
dat_stocks$esgScore.env <- as.integer(NA)   # ESG Score (Environmental)
dat_stocks$esgScore.soc <- as.integer(NA)   # ESG Score (Social)
dat_stocks$esgScore.gov <- as.integer(NA)   # ESG Score (Governance)
dat_stocks$esgRank.tot  <- as.numeric(NA)   # Percentile Rank (Total/Overall)
dat_stocks$esgRank.env  <- as.numeric(NA)   # Percentile Rank (Environmental)
dat_stocks$esgRank.soc  <- as.numeric(NA)   # Percentile Rank (Social)
dat_stocks$esgRank.gov  <- as.numeric(NA)   # Percentile Rank (Governance)
dat_stocks$conRating    <- as.character(NA) # Controversy Rating
dat_stocks$conLevel     <- as.integer(NA)   # Controversy Level
dat_stocks$conAreas     <- as.character(NA) # Controversy Areas (Products)
dat_stocks$asOf         <- Sys.Date()       # Last Updated date
```

Create Mutual Funds Data Table

```r
# NEED TO LOAD THIS CSV FILE TO CORRECT LOCATION ON YOUR LOCAL COMPUTER FIRST:
# https://www.kylerudden.com/blog/scraping-esg-scores/dat_funds.csv

# Location for me:
dat_funds <- read.csv("dat_funds.csv")
```

```r
head(dat_funds)
```

```
##   X ticker     familyName                                        fundName
## 1 1  ABEMX       Aberdeen                 Aberdeen Emerging Markets Fund
## 2 2  ABALX American Funds           American Funds American Balanced Fund
## 3 3  AMRMX American Funds             American Funds American Mutual Fund
## 4 4  AEPGX American Funds          American Funds EuroPacific Growth Fund
## 5 5  AGTHX American Funds       American Funds The Growth Fund of America
## 6 6  AWSHX American Funds American Funds Washington Mutual Investors Fund
```

Create Placeholder Columns for ESG Fund Data

```
dat_funds$esgRating    <- as.character(NA) # ESG Rating
dat_funds$esgScore.tot <- as.integer(NA)   # ESS Score (Total/Portfolio)
dat_funds$esgScore.env <- as.integer(NA)   # ESG Score (Environmental)
dat_funds$esgScore.soc <- as.integer(NA)   # ESG Score (Social)
dat_funds$esgScore.gov <- as.integer(NA)   # ESG Score (Governance)
dat_funds$esgScore.aum <- as.integer(NA)   # ESG Score (% AUM basis)
dat_funds$esgScore.raw <- as.integer(NA)   # ESG Score (Raw)
dat_funds$esgScore.ded <- as.integer(NA)   # ESG Score (Controversy Deduction)
dat_funds$susMandate   <- as.character(NA) # Sustainability Mandate
dat_funds$susRank.pct  <- as.numeric(NA)   # Sustainability Rank (Percentile)
dat_funds$susRank.cat  <- as.numeric(NA)   # Sustainability Rank (Category)
dat_funds$asOf         <- Sys.Date()       # Last Updated date
```

Step 3: Download ESG Data:

Download Companies ESG Data

```
i <- 1
for (i in 1:nrow(dat_stocks)) {
  message(paste0(i, " of ", nrow(dat_stocks)))
  tryCatch({
    tick.i <- dat_stocks$ticker[i]
    link.i <- paste0("https://finance.yahoo.com/quote/", tick.i, "/sustainability")
    bots.i <- suppressMessages(fun_robots(link.i))
    if (bots.i) {
      Sys.sleep(runif(1, 0.5, 3.0))
      page.i <- GET(link.i, user_agent(var_agent)) %>% content()
      dat_stocks$esgRating[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/di
      dat_stocks$esgScore.tot[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div
      dat_stocks$esgScore.env[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div
      dat_stocks$esgScore.soc[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div
      dat_stocks$esgScore.gov[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div
      dat_stocks$esgRank.tot[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/
      dat_stocks$esgRank.env[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/
      dat_stocks$esgRank.soc[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/
      dat_stocks$esgRank.gov[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/
      dat_stocks$conRating[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/div[2]
      dat_stocks$conLevel[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/div[2]/
      dat_stocks$conAreas[i] <- fun_lists()
      dat_stocks$asOf[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[3]/span[2]/spa
    }
  }, error=function(e){})
}
dat_stocks$asOf[which(is.na(dat_stocks$esgRating))] <- NA
```

Inspect Proportion of Morningstar/Sustainalytics Data Present

```r
scales::percent(sum(!is.na(dat_stocks$esgRating)) / nrow(dat_stocks))
```

```
## [1] "54%"
```

Add Percentage Market Capitalization Data

```r
#Using the quantmod library
dat_stocks$mktCap <- suppressWarnings(
  quantmod::getQuote(dat_stocks$ticker, what = "marketCap")$marketCap
)
```

```
## downloading set: 1 , 2 , 3 , ...done
```

```r
scales::percent(sum(dat_stocks$mktCap[which(!is.na(dat_stocks$esgRating))]) / sum(dat_stocks$mktCap))
```

```
## [1] "65%"
```

Save Stocks Dataframe to a .csv File

```r
write.csv(dat_stocks, 'dat_stocks.csv')
```

Download Mutual Funds ESG Data

```r
i <- 1
for (i in 1:nrow(dat_funds)) {
  message(paste0(i, " of ", nrow(dat_funds)))
  tryCatch({
    tick.i <- dat_funds$ticker[i]
    link.i <- paste0("https://finance.yahoo.com/quote/", tick.i, "/sustainability")
    bots.i <- suppressMessages(fun_robots(link.i))
    if (bots.i) {
      Sys.sleep(runif(1, 0.5, 3.0))
      page.i <- GET(link.i, user_agent(var_agent)) %>% content()
      if (grepl("ESG", fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/h3/span'))) {
        dat_funds$esgRating[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/div/
        dat_funds$esgScore.tot[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/di
        dat_funds$esgScore.env[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/di
        dat_funds$esgScore.soc[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/di
        dat_funds$esgScore.gov[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[1]/di
        dat_funds$esgScore.aum[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/di
        dat_funds$esgScore.raw[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/di
        dat_funds$esgScore.ded[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[2]/di
        dat_funds$susMandate[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[3]/div/
        dat_funds$susRank.pct[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[3]/p[1
        dat_funds$susRank.cat[i] <- page.i %>%
```

```
          html_nodes(xpath = '//*[@id="Col1-0-Sustainability-Proxy"]/section/div[3]/p[2]/span/span') %>%
          html_text(trim = TRUE)
        dat_funds$asOf[i] <- fun_parse('//*[@id="Col1-0-Sustainability-Proxy"]/section/div[4]/span[2]/sp

      }
    }
  }, error=function(e){})
}
dat_funds$asOf[which(is.na(dat_funds$esgRating))] <- NA
```

Inspect Raw Score and Controversy Deduction

```
dat_funds$esgScore.tot - (dat_funds$esgScore.raw + dat_funds$esgScore.ded)
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

Step 4: Initial Analysis

Stocks Data Summary

```
stock_look <- subset(dat_stocks, !is.na(esgRating)) %>%
  group_by(sector) %>%
  summarise(
    esgScore.tot = ceiling(mean(esgScore.tot)),
    esgScore.env = ceiling(mean(esgScore.env)),
    esgScore.soc = ceiling(mean(esgScore.soc)),
    esgScore.gov = ceiling(mean(esgScore.gov)),
  ) %>%
  ungroup()

stock_look <- stock_look[order(stock_look$esgScore.tot, decreasing = TRUE), ]
stock_look
```

```
## # A tibble: 11 x 5
##    sector                 esgScore.tot esgScore.env esgScore.soc esgScore.gov
##    <chr>                         <dbl>        <dbl>        <dbl>        <dbl>
##  1 Energy                           34           18           10            8
##  2 Utilities                        33           17           11            7
##  3 Materials                        28           15            8            7
##  4 Industrials                      27            9           13            7
##  5 Consumer Staples                 26            9           12            6
##  6 Health Care                      23            3           13            9
##  7 Financials                       20            2            9           10
##  8 Communication Services           19            1           10            9
##  9 Consumer Discretionary           17            4            8            6
## 10 Information Technology           17            3            8            7
## 11 Real Estate                      15            5            5            6
```

Funds Data Summary

```
fund_look <- subset(dat_funds, !is.na(esgRating)) %>%
  group_by(familyName) %>%
  summarise(
    esgScore.tot = ceiling(mean(esgScore.tot)),
    esgScore.env = ceiling(mean(esgScore.env)),
    esgScore.soc = ceiling(mean(esgScore.soc)),
    esgScore.gov = ceiling(mean(esgScore.gov)),
  ) %>%
  ungroup()
fund_look <- fund_look[order(fund_look$esgScore.tot, decreasing = TRUE), ]
fund_look
```

```
## # A tibble: 12 x 5
##    familyName            esgScore.tot esgScore.env esgScore.soc esgScore.gov
##    <chr>                        <dbl>        <dbl>        <dbl>        <dbl>
##  1 Aberdeen                        24            6            9            9
##  2 American Funds                  24            5           11            8
##  3 Hartford Mutual Funds           24            5           11            8
##  4 Vanguard                        24            5           10            8
##  5 Invesco                         23            3            9            7
##  6 Jensen                          22            3           11            8
##  7 JPMorgan                        22            4           10            8
##  8 Legg Mason                      21            3           10            8
##  9 Morgan Stanley                  21            2           10            8
## 10 Putnam                          21            4            9            7
## 11 TIAA Investments                21            3            9            7
## 12 MainStay                        20            2           10            8
```

Reference Cited:

https://www.kylerudden.com/blog/scraping-esg-scores/