Webscraping in R using Rvest

Loren Collingwood

To harvest text, the webscraping skillset is a must. This vignette introduces webscraping in R using the package Rvest. When I was growing up, we used Python to webscrape but now that is less necessary. In my experience, you cannot always webscrape everything, and a one-size fits all

Step 1

Download the selector gadget chrome extension.

Step 2

Install then load the Rvest package.

```
    # Install R package if you have not done so #
    #install.packages("rvest")

    # Load package into current R session #
    library("rvest")
#> Loading required package: xml2
```

Step 3

Select a website you want to scrape and investigate.

1. Investigate whether the particular information you want from the website can be scraped.
2. Is the information presented across a series of pages?
3. Is the information possibly in a table?
4. Are there multiple bits of information you want to scrape? i.e., title, author, date, main body of text.
5. Think about how you want the data structure to look like when webscraping is complete. Do you want the text in single .txt files, a single .csv file with columns for metadata?

Step 4

Open the website (if not already open). Click on the selector gadget and begin looking for unique html tags that identify what you want to scrape

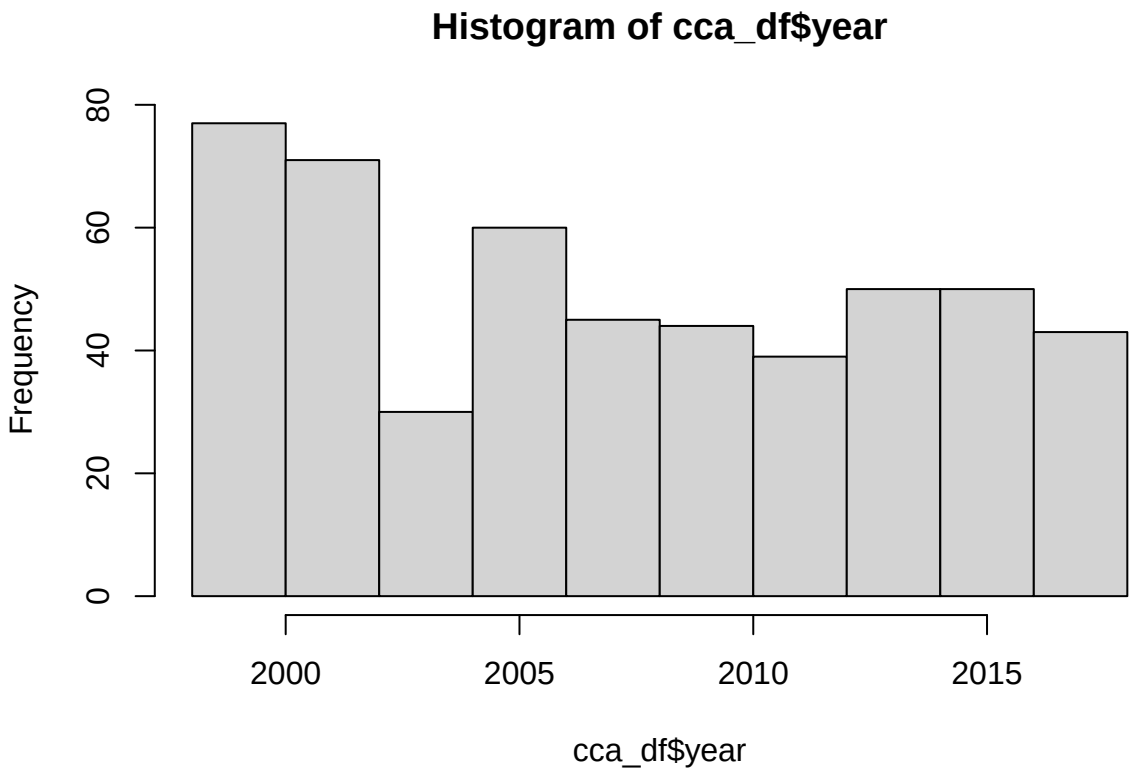
Example

Core Civic or Corrections Corporation of America (CCA) is a private corrections firm publicly traded on the NYSE. The company produces newsletters to shareholders publicly available on its website. Let's scrape these.

```
library(stringr)
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
library(data.table)
#>
#> Attaching package: 'data.table'
#> The following objects are masked from 'package:dplyr':
#>
#>     between, first, last

setwd("~/Dropbox/collingwood_research/posc_fall_20/POSC-207/data")

list.files()
#> [1] "cca_links.csv"

########################
# Read in Links Data #
########################

links <- read.csv("cca_links.csv", header=T, stringsAsFactors = F)

links <- links$link

# Initiate list holder/container #
cca_hold <- list()

n <- length(links)

#################
# Initiate Loop #
#################

for (i in 1:n){

  if (i == 1) message("Start")
  if (i == round(n*.5,0)) message("50% Done")
  if (i == n) message("Done!")

  ###############################################
  # Read in Link -- this is the main iterator #
  ###############################################
```

```
  cca <- read_html(links[i])

  #################
  # Extract Date #
  #################

  text <- html_text(html_nodes(cca, "div div")); length(text)

  # Date Regular Expression and Clean #
  dates <- str_squish(str_extract(text, "[a-zA-Z]+ [0-9]+[,]+\\s+[0-9]{4}"))

  # Further Cleaning #
  dates <- dates[!is.na(dates)][1] # take out nas and select first date

  ####################
  # Extract Headline #
  ####################

  # Collapse Text into Vector #
  text <- paste(text, collapse=" ")

  text_manip <- unlist ( str_split(text, dates) )[1] # take the first split on the date
  text_manip <- unlist(str_split(text_manip, "\n\n\n")) # Assumes \n\n\n is in all releases
  text_manip <- text_manip[length(text_manip)] # assumes the headline is last
  headline <- str_trim(text_manip) # Store into headline vector

  rm(text) # Garbage Clean #

  ######################################
  # Get Paragraph Text of News Release #
  ######################################

  text <- html_text(html_nodes(cca, "div p")); length(par)
  text <- paste(text, collapse = " ")
  text <- gsub("\r","", text)
  text <- gsub("\n","", text)

  # Place data frame into List Item #
  cca_hold[[i]] <- data.frame(dates, headline, text, stringsAsFactors = F)

}
#> Start
#> 50% Done
#> Done!

############################################
# Convert List of dataframes to dataframe #
############################################

cca_df <- rbindlist(cca_hold)

########################
# Look at Column Names #
```

```
########################

names(cca_df)
#> [1] "dates"    "headline" "text"

#################
# Look at Dates #
#################

cca_df$dates <- lubridate::mdy(cca_df$dates) # Convert dates #
cca_df$year <- lubridate::year(cca_df$dates) # Convert dates #
head(cca_df$headline)
#> [1] "CoreCivic Declares Quarterly Cash Dividend of $0.43 Per Share"
#> [2] "CoreCivic Reports Third Quarter 2018 Financial Results"
#> [3] "CoreCivic Appoints Devin I. Murphy to Its Board of Directors"
#> [4] "CoreCivic Announces 2018 Third Quarter Earnings Release and Conference Call Dates"
#> [5] "CoreCivic Enters Into New Management Contract with the State of Vermont at the Tallahatchie Cou
#> [6] "CoreCivic Announces Acquisition of 540,566 SF Social Security Administration Facility in Baltim

###########################################
# Take a quick look at Yearly Distribution #
###########################################

hist(cca_df$year)
```



Summary

This vignette has provided a brief overview of the workflow for using webscraping in R.