Data Acquisition Tutorial

Election Data Science

Peter Licari

2020-09-21

What we'll be covering

1. How to read-in a csv
2. How to read-in a tsv
3. How to read-in a xlsx
4. How to acquire data from the census
5. How to download data from site directly
6. How to read-in data from a web (REST) API
7. How to scrape a table from Wikipedia

Set up

You can get the data for today at the class GitHub site. (Don't forget to set your working directory!)

```r
library(tidyverse)
library(readxl)
library(httr)
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.6.3

## Loading required package: xml2

##
## Attaching package: 'rvest'

## The following object is masked from 'package:purrr':
##
##     pluck

## The following object is masked from 'package:readr':
##
##     guess_encoding
```

Reading in a csv

The Code   Output

```r
#Make sure you have readr and/or tidyverse active!

Pres_Cands_2020 <- read_csv("fec_cands_july.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   district_number = col_double(),
##   load_date = col_datetime(format = ""),
##   first_file_date = col_date(format = ""),
##   last_file_date = col_date(format = ""),
##   last_f2_date = col_date(format = ""),
##   active_through = col_double(),
##   candidate_inactive = col_logical(),
##   inactive_election_years = col_logical()
## )

## See spec(...) for full column specifications.
```

Reading in a tsv

The Code   Output

```
#Remember, sometimes TSVs (and CSVs--and PSVs for that matter) are writ

Tyyrell_history <- read_tsv("voterhistory.txt")
```

Both `read_csv` and `read_tsv` are special implementations of the more general `read_delim` (meaning read delimited). With this function, you can specify the **delim**iter with the `delim` option allowing you to read in other kinds of delimited data.

How to read-in a xlsx

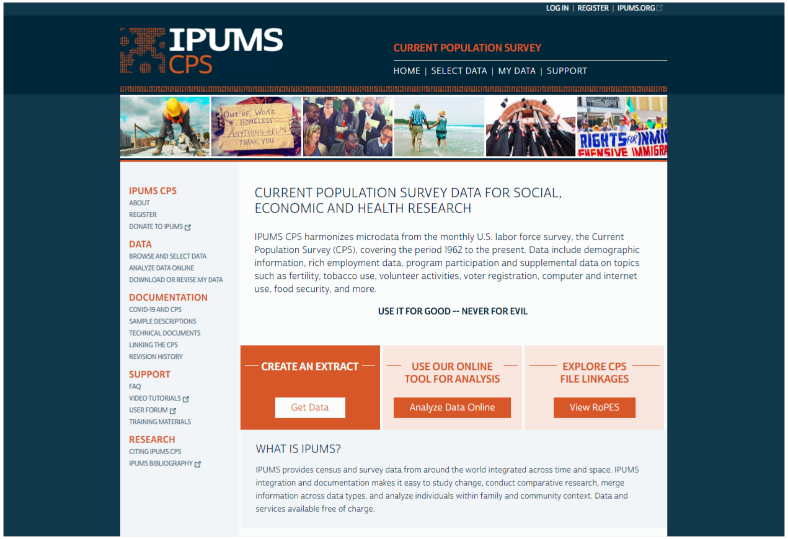
Code Output

```
#Remember, sometimes TSVs (and CSVs--and PSVs for that matter) are writ

gainesville_contributions <- read_xlsx("gainesville_pacs.xlsx")
```

How to get Census data

CPS   Other Data



CURRENT POPULATION SURVEY DATA FOR SOCIAL, ECONOMIC AND HEALTH RESEARCH

IPUMS CPS harmonizes microdata from the monthly U.S. labor force survey, the Current Population Survey (CPS), covering the period 1962 to the present. Data include demographic information, rich employment data, program participation and supplemental data on topics such as fertility, tobacco use, volunteer activities, voter registration, computer and internet use, food security, and more.

USE IT FOR GOOD -- NEVER FOR EVIL

WHAT IS IPUMS?

IPUMS provides census and survey data from around the world integrated across time and space. IPUMS integration and documentation makes it easy to study change, conduct comparative research, merge information across data types, and analyze individuals within family and community context. Data and services available free of charge.

Data directly

```r
#File URL
url1 <- "https://s3.amazonaws.com/dl.ncsbe.gov/data/ncvoter89.zip"

#Downloads file from url1 and pastes it in the current directory as vote
download.file(url1, destfile = paste0(getwd(),"/voterfile.zip"))

#Unzips the file
unzip("voterfile.zip")

#Renames (not needed, but useful for clarity)
file.rename(from = "ncvoter89.txt", to = "voterregistration.txt")
```

Scraping from an API

Code How it works Output

Let's

```
#API web address
abs_voting <- "https://api.gdeltproject.org/api/v2/tv/tv?query=%22absent

data_raw <- httr::GET(abs_voting)
abs_mentions <- httr::content(data_raw)

#Renaming columns; filtering to just get CNN, MSNBC, and FOX
abs_mentions <- abs_mentions %>%
  rename("time" = 1, "channel" = 2) %>%
  filter(channel %in% c("CNN", "MSNBC", "FOXNEWS"))
```

Scrape a table from wikipedia

Initial Scrape Raw Results Cleaning Results

```
url <- "https://en.wikipedia.org/wiki/2018_United_States_House_of_Repres

fl_web_raw <- url %>%
  read_html() %>%
  html_node(xpath = '/html/body/div[3]/div[3]/div[5]/div[1]/table[3]') %
  html_table(fill=TRUE)
```