Get Eds Data

Grant Esparza

September 24, 2018

Problem

During lecture we often use data from the class website to demonstrate new topics. I found it annoying to have to go to the website, copy the link to the data I need, and then paste it into `read.csv()`. What if there was an easier way?

Goal

Ultimately I need a function that takes the name of the data I want and returns a dataframe so I can begin using the data. While I am aware that R has the ability to webscrape using various libararies, my experience with webscraping is with Python, so I'll need a way of using Python code to grab the data from the MATH 385 website.

Python

Thankfully Rstudio has created a library reticulate that allows the user to call a script and use the returned value. Therefore the first step is to create a python script that grabs the html for the class website and finds the url to the dataset in question. I'll need a couple pythone modules:

```
import requests ## Retrieving html
from bs4 import BeautifulSoup ## Extract using html tags
```

Python (cont.)

The python code here is pretty simple thanks to the BeautifulSoup module. I retrieve the html page and convert the result into a soup object. Then I use call find() on the object to create a list of divs with id="MATH385data. Since I need the title of the data and the link I want to refine this list to only contain li elements. I then use a for loop to check if the current item is contains the substring the user specified. The function then returns the link concatenated with the rest of the url.

```python
def retrieve(target):
    page = requests.get(eds_url + '/teaching')
    soup = BeautifulSoup(page.text, 'html.parser')
    data_list = soup.find("div", id="MATH385data")
    data_list_items = data_list.find_all('li')
    for data in data_list_items:
        if target in data.contents[0]:
            res = data.find('a')
            return eds_url + res.get('href')
```

R

Since most of the work was done in `python`, the `R` function is mostly just ensuring that the enviroment has the library loaded and knows which python modules are going to be imported. I also used a default argument of `hospital` simply because we've been using it often. After the script returns, I use the returned string as the url to pass into `read.csv()`:

```r
library(reticulate)
get_eds_data <- function(target = 'hospital') {
  os <- import("requests")
  os <- import("bs4")
  source_python("grab_data.py")
  read.csv(retrieve(target))
}
```

Result

Now that the function is declared in R it can be used:

```
email <- get_eds_data("speed")
pander(head(email, 3))
```

system	mips	year	cores
UNIVAC I	0.002	1951	1
IBM System/370 158	0.64	1972	1
Intel 8080	0.29	1974	1