

Reading data from the web

Jeffrey Leek Johns Hopkins Bloomberg School of Public Health

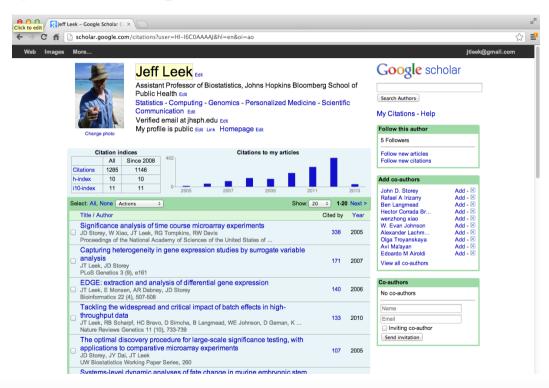
Webscraping

Webscraping: Programatically extracting data from the HTML code of websites.

- · It can be a great way to get data How Netflix reverse engineered Hollywood
- · Many websites have information you may want to programaticaly read
- · In some cases this is against the terms of service for the website
- · Attempting to read too many pages too quickly can get your IP address blocked

http://en.wikipedia.org/wiki/Web_scraping

Example: Google scholar



http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en

Getting data off webpages - readLines()

```
con = url("http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en")
htmlCode = readLines(con)
close(con)
htmlCode
```

```
[1] "<!DOCTYPE html><html><head><title>Jeff Leek - Google Scholar Citations</title><meta name=\"robots\"
```

Parsing with XML

```
library(XML)
url <- "http://scholar.google.com/citations?user=HI-I6C0AAAAJ&hl=en"
html <- htmlTreeParse(url, useInternalNodes=T)

xpathSApply(html, "//title", xmlValue)</pre>
```

```
[1] "Jeff Leek - Google Scholar Citations"
```

```
xpathSApply(html, "//td[@id='col-citedby']", xmlValue)
```

```
[1] "Cited by" "397"
                                                         "138"
                                                                    "125"
                         "259"
                                    "237"
                                               "172"
                                                                               "122"
[9] "109"
               "101"
                         "34"
                                    "26"
                                               "26"
                                                         "24"
                                                                    "19"
                                                                              "13"
                                               "6"
[17] "12"
               "10"
                         "10"
                                    "7"
```

GET from the httr package

```
library(httr); html2 = GET(url)
content2 = content(html2,as="text")
parsedHtml = htmlParse(content2,asText=TRUE)
xpathSApply(parsedHtml, "//title", xmlValue)
```

```
[1] "Jeff Leek - Google Scholar Citations"
```

Accessing websites with passwords

```
pg1 = GET("http://httpbin.org/basic-auth/user/passwd")
pg1
```

```
Response [http://httpbin.org/basic-auth/user/passwd]
Status: 401
Content-type:
```

http://cran.r-project.org/web/packages/httr/httr.pdf

Accessing websites with passwords

```
Response [http://httpbin.org/basic-auth/user/passwd]
  Status: 200
  Content-type: application/json
{
   "authenticated": true,
   "user": "user"
}
```

```
names(pg2)
```

```
[1] "url" "handle" "status_code" "headers" "cookies" "content"
[7] "times" "config"
```

Using handles

```
google = handle("http://google.com")
pg1 = GET(handle=google,path="/")
pg2 = GET(handle=google,path="search")
```

http://cran.r-project.org/web/packages/httr/httr.pdf

Notes and further resources

- · R Bloggers has a number of examples of web scraping http://www.r-bloggers.com/?
 s=Web+Scraping
- The httr help file has useful examples http://cran.r-project.org/web/packages/httr/httr.pdf
- · See later lectures on APIs