



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 programatically 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

Jeff Leek Edit
 Assistant Professor of Biostatistics, Johns Hopkins Bloomberg School of Public Health Edit
 Statistics - Computing - Genomics - Personalized Medicine - Scientific Communication Edit
 Verified email at jhsph.edu Edit
 My profile is public Edit [Link](#) [Homepage](#) Edit

Citation indices

	All	Since 2008
Citations	1285	1146
h-index	10	10
i10-index	11	11

Citations to my articles

Select: All, None **Actions** **Show:** 20 **1-20** [Next >](#)

Title / Author	Cited by	Year
<input type="checkbox"/> Significance analysis of time course microarray experiments JD Storey, W Xiao, JT Leek, RG Tompkins, RW Davis Proceedings of the National Academy of Sciences of the United States of ...	338	2005
<input type="checkbox"/> Capturing heterogeneity in gene expression studies by surrogate variable analysis JT Leek, JD Storey PLoS Genetics 3 (9), e161	171	2007
<input type="checkbox"/> EDGE: extraction and analysis of differential gene expression JT Leek, E Monsen, AR Dabney, JD Storey Bioinformatics 22 (4), 507-508	140	2006
<input type="checkbox"/> Tackling the widespread and critical impact of batch effects in high-throughput data JT Leek, RB Scharpf, HC Bravo, D Simcha, B Langmead, WE Johnson, D Geman, K ... Nature Reviews Genetics 11 (10), 733-739	133	2010
<input type="checkbox"/> The optimal discovery procedure for large-scale significance testing, with applications to comparative microarray experiments JD Storey, JY Dai, JT Leek UW Biostatistics Working Paper Series, 260	107	2005

Systems-level dynamic analyses of fate chance in murine embryonic stem

Google scholar

My Citations - Help

Follow this author

5 Followers

Add co-authors

John D. Storey	Add
Rafael A Irizarry	Add
Ben Langmead	Add
Hector Corrada Br...	Add
wenzhong xiao	Add
W. Evan Johnson	Add
Alexander Lachm...	Add
Olga Troyanskaya	Add
Avi Ma'ayan	Add
Edoardo M Airoidi	Add

[View all co-authors](#)

Co-authors

No co-authors

☐ **Inviting co-author**

<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\"></head><body><div class=\"main-content\"><div class=\"text\">Jeff Leek</div><div class=\"text\">Google Scholar Citations</div></div></body></html>"
```

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)
```

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

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

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

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/htr/htr.pdf>

Accessing websites with passwords

```
pg2 = GET("http://httpbin.org/basic-auth/user/passwd",  
          authenticate("user", "passwd"))  
pg2
```

```
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/htr/htr.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