

Data Wrangling in R

Advanced data io

Google Sheets



jtleek@gmail.com

File Edit View Insert Format Data Tools Add-ons Help

Comments

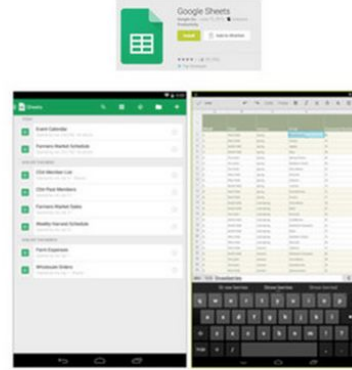
 Share f_x

<https://docs.google.com/spreadsheets>

iOS



Android



enter data from a phone
enter data w/o WiFi

<https://speakerdeck.com/jennybc/googlesheets-talk-at-user2015>

Reading data with the googlesheets package

```
install.packages("googlesheets")
```

```
library(googlesheets)
```



data_wrangling_background_2019

File Edit View Insert Format Data Tools Add-ons Help

Working...



Sh

fx											
	A	B	C	D	E	F	G	H	I	J	
1	Git	Github	R	Rstudio	Reproducible Research	R markdown	Data import	Web scraping	Data cleaning	dplyr	Bioco
2	9	9	9	9	9	9	10	10	9	8	
3	1	1	5	5	7	5	4	2	6	1	
4	0	0	2	3	3	0	3	0	2	0	
5	0	1	1	1	1	1	1	0	0	1	
6	0	2	7	7	7	5	8	0	6	6	
7	2	2	3	2	0	0	1	1	2	0	
8	0	0	4	5	7	5	5	2	4	3	
9	1	2	3	3	5	0	0	0	7	0	
10	1	1	3	3	4	3	3	0	3	1	
11	1	1	5	5	6	4	7	1	5	2	
12	1	2	3	3	3	2	2	1	1	1	
13	1	1	4	4	3	3	5	2	4	6	
14	0	1	5	6	0	0	1	1	5	0	
15	0	0	1	1	9	0	0	0	0	0	
16	1	2	7	8	5	6	5	0	6	5	
17	1	1	3	3	9	1	3	0	3	1	
18	1	1	5	5	5	2	6	0	3	2	
19	1	1	1	2	1	3	3	1	5	0	
20	0	1	2	3	2	0	3	4	4	0	
21	0	0	2	0	5	0	1	0	5	0	
22	0	0	3	3	5	3	3	1	3	1	
23	0	0	4	4	2	3	2	0	3	0	
24	1	1	6	6	5	3	6	0	5	6	
25	5	7	6	6	6	1	4	0	2	1	
26	1	1	4	5	9	4	5	0	3	4	
27	2	2	5	6	3	3	7	2	6	4	
28	0	0	1	1	2	1	2	1	2	0	
29	0	0	5	5	5	0	5	0	5	4	
30	1	1	1	1	0	1	0	0	1	0	



data_wrangling_background_2019

File Edit View Insert Format Data Tools Add-ons Help [Last edit was yesterday at 8:50 AM](#)

Share...

New

Open Ctrl+O

Import

Make a copy

Download as

Email as attachment...

Make available offline

Version history

Rename

Move to...

Move to trash

Publish to the web...

Email collaborators

Document details...

Spreadsheet settings

 Print Ctrl+P

	C	D	E	F	G	H	I
		Rstudio	Reproducible Research	R markdown	Data import	Web scraping	Data
1	9	9	9	9	10	10	
2	5	5	7	5	4	2	
3	2	3	3	0	3	0	
4	1	1	1	1	1	0	
5	7	7	7	5	8	0	
6	3	2	0	0	1	1	
7	4	5	7	5	5	2	
8	3	3	5	0	0	0	
9	3	3	4	3	3	0	
10	5	5	6	4	7	1	
11	3	3	3	2	2	1	
12	4	4	3	3	5	2	
13	5	6	0	0	1	1	
14	1	1	9	0	0	0	
15	7	8	5	6	5	0	
16	3	3	9	1	3	0	
17	5	5	5	2	6	0	
18	1	2	1	3	3	1	
19	2	3	2	0	3	4	
20	2	0	5	0	1	0	
21	3	3	5	3	3	1	
22	4	4	2	3	2	0	
23	6	6	5	3	6	0	
24	6	6	6	1	4	0	
25	4	5	9	4	5	0	
26	2	6	3	3	7	2	
27	0	1	2	1	2	1	
28	0	5	5	0	5	0	

```
sheets_url =  
"https://docs.google.com/spreadsheets/d/1WBrH655fxqKW1Q  
qvD5hnqvEWIvRzDJcKEgjjFeYxeM/edit?usp=sharing"
```

```
gsurl1 = gs_url(sheets_url)
```

```
dat = gs_read(gsurl1)
```

```
date_read = date()
```

```
date_read
```




	A	B	C	D	E	F
1	UW?	R user (yes or no)	Years of R experience	Github User	Years of Github Experience	Why are you taking this module (free text)
2	no	yes		5 yes		5 learn more about health related application in big data and statistics
3	no	yes		4 sometimes		1 learn best practices
4	no	yes		2 no		0 learn more R skills
5	no	yes		1 aspiring		0 new tips and tricks
6	no	yes		4 sparsely		4 have intermediate knowledge, would love to learn more
7	No	YEs but sparse		2 no		0 learn about big data and analyses
8	no	yes		1 no		0 need it for research, self-teaching inefficient
9	No	Yes	<2 months	No		0
10	No	Yes	3+ Years	kinda		2 Boss wants me to!
11						
12	no	not regularly		1 no		0 as prep for machine learning course
13	yes	meh	2 months	meh	1 month	learn to integrate large multimodal (and nested) data at multiple levels
14	no	yes		5 no	no	i'm pretty much self-taught, so need skill honing and best practices.
15	yes	yes	3 years	nah		want to expand my R knowledge, break out of typical coding patterns
16	no	no		0 no		0 i need to learn to how to analyze large data sets
17						
18	once	yes	48 months	nope	1 month	need to get back into R and start git
19						
20	alumna	no		0.5 no		0 really improve my R skills to feel more comfortable using it, get an intr
21	Yes	Yes		3 no		0 Help with research and advisor recommended it
22	no	no	<1	no		0 need to learn R for work
23	no	future user	<1	no		0 need to learn how become more adept/proficient at using R for my rese
24						
25	no	yes	<1	yes	<1	Learn about health data science
26						
27	no	yes		1 not really		0.1 I work in an environment where SAS is mainstream though I have been
28	no	yes	<1	no		0 to utilize R for RNA seq data
29	yes	yes	<1	no		0 I am doing omics
30	no	user-ish	<1	no		0 getting more R experience in processing data versus what I use. - perl

```
sheets_url =  
"https://docs.google.com/spreadsheets/d/1j9vbv8MrVV7EK1  
5vyz-rnhjiXhRkmIFEHgdv1_p1cCc/edit?usp=sharing"
```

```
# Only necessary on rstudio.cloud
```

```
options(httr_oob_default=TRUE)
```

```
# Will ask you to log in
```

```
gs_auth()
```

```
gsurl1 = gs_url(sheets_url)
```

```
dat = gs_read(gsurl1)
```

Google Sheets

<https://bit.ly/1Cgzjxb>

JSON

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 25,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    }
  ],
  "children": [],
  "spouse": null
}
```

Why JSON matters

<https://developer.github.com/v3/search/>



When searching for repositories, you can get text match metadata for the **name** and **description** fields. (See the section on [text match metadata](#) for full details.)

Here's an example response:

```
{
  "text_matches": [
    {
      "object_url": "https://api.github.com/repositories/3081286",
      "object_type": "Repository",
      "property": "name",
      "fragment": "Tetris",
      "matches": [
        {
          "text": "Tetris",
          "indices": [
            0,
            6
          ]
        }
      ]
    },
    {
      "object_url": "https://api.github.com/repositories/3081286",
      "object_type": "Repository",
      "property": "description",
      "fragment": "A C implementation of Tetris using Pennsim through LC4",
      "matches": [
        {
          "text": "Tetris",
          "indices": [
            22,
            28
          ]
        }
      ]
    }
  ]
}
```

```
github_url = "https://api.github.com/users/jtleek/repos"

#install.packages("jsonlite")
library(jsonlite)
jsonData <- fromJSON(github_url)

date_read = date()

date_read
```

Data frame structure from JSON

```
dim(jsonData)
```

```
jsonData$name
```

```
#One of the columns is a data frame!
```

```
table(apply(jsonData,class))
```

```
dim(jsonData$owner)
```

```
names(jsonData$owner)
```

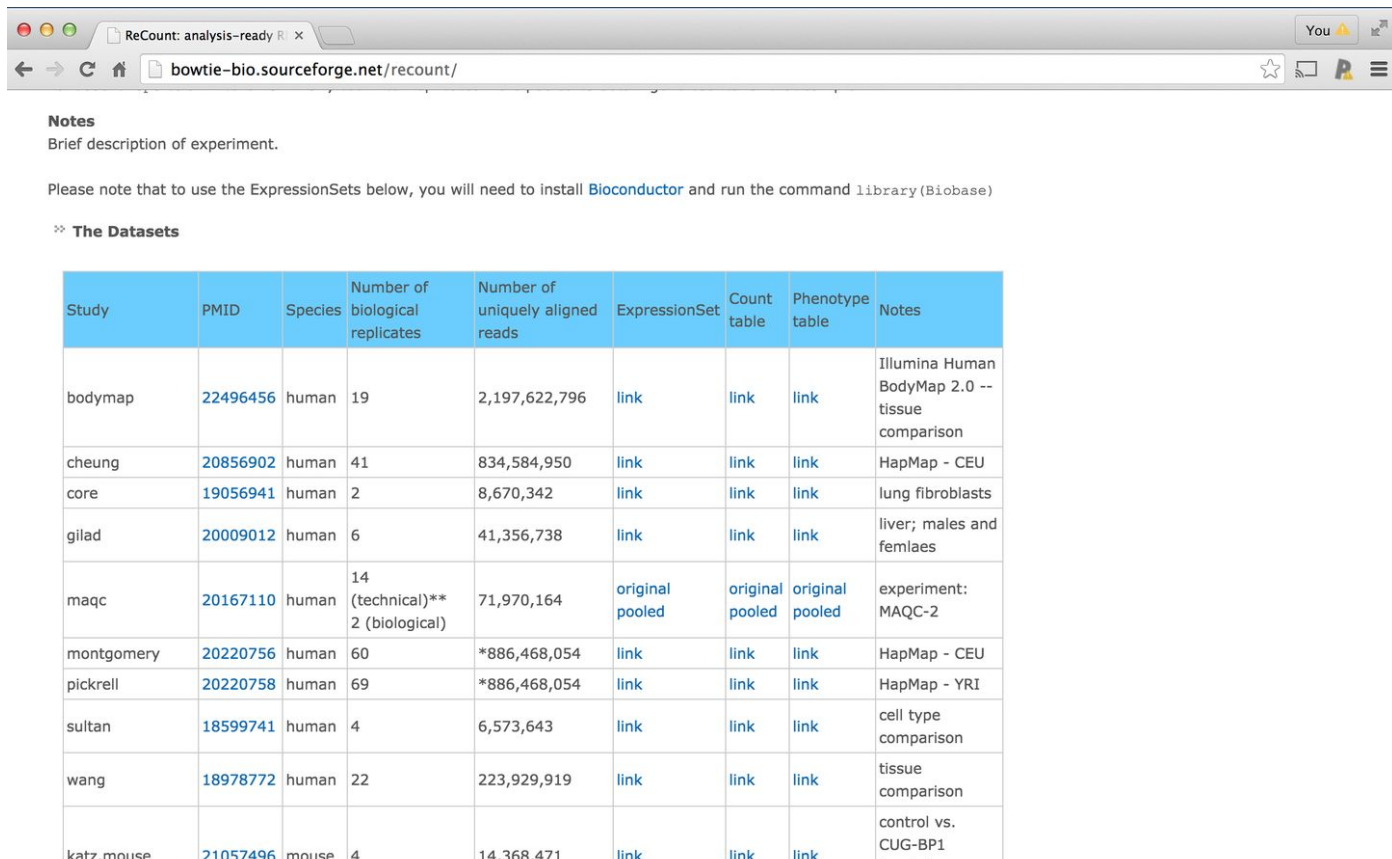

JSON Lab

<https://bit.ly/2JNLUil>

Web Scraping

This is data

<http://bowtie-bio.sourceforge.net/recount/>



Notes
Brief description of experiment.

Please note that to use the ExpressionSets below, you will need to install [Bioconductor](#) and run the command `library(Biobase)`

✧ **The Datasets**

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and females
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	link	HapMap - YRI
sultan	18599741	human	4	6,573,643	link	link	link	cell type comparison
wang	18978772	human	22	223,929,919	link	link	link	tissue comparison
katz mouse	21057496	mouse	4	14,368,471	link	link	link	control vs. CUG-BP1

View the source



Notes

Brief description of experiment.

Please note that to use the ExpressionSets below, you will need to install the `library(Biobase)` and run the command `library(Biobase)`

✧ The Datasets

Study	PMID	Species	Number of biological replicates	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19		link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	liver; males and females
maq	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	HapMap - CEU
montgomery	20220756	human	60	*886,468,054	link	link	HapMap - CEU

What the computer sees

Brief description of experiment.<br \><br \>

Please note that to use the ExpressionSets below, you will need to install Bioconductor and run the command
<tt>library(Biobase)</tt>

```
<h3>The Datasets</h3>
<div id="recounttab">
<table class="sortable"><tbody>
<tr>
<td>Study</td>
<td>PMID</td>
<td>Species</td>
<td>Number of biological replicates</td>
<td>Number of uniquely aligned reads</td>
<td>ExpressionSet</td>
<td>Count table</td>
<td>Phenotype table</td>
<td>Notes</td>
</tr>

<tr>
<td>bodymap</td>
  <td><a href="http://www.ncbi.nlm.nih.gov/pubmed/22496456">22496456</a></td>
  <td>human</td>
  <td>19</td>
  <td>2,197,622,796</td>
  <td><a href="./ExpressionSets/bodymap_eset.RData">link </a></td>
  <td><a href="./countTables/bodymap_count_table.txt">link</a></td>
  <td><a href="./phenotypeTables/bodymap_phenodata.txt">link</a></td>
  <td>Illumina Human BodyMap 2.0 -- tissue comparison</td></tr>

<tr>
<td>cheung</td>
  <td><a href="http://www.ncbi.nlm.nih.gov/pubmed?term=20856902">20856902</a></td>
```

Ways to see the source

Chrome:

1. right click on page

2. select "view source"

Firefox:

1. right click on page

2. select "view source"

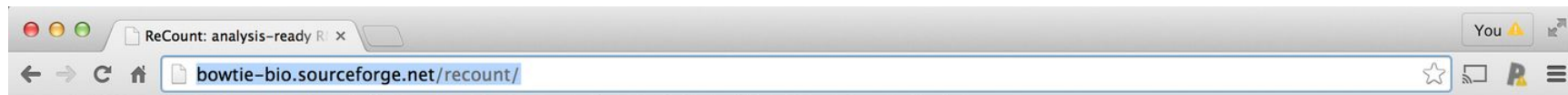
Microsoft Edge:

1. right click on page

2. select "view source"

<https://github.com/simonmunzert/rscraping-jsm-2016/blob/c04fd91fec711df65c838e07723125155a7f2cda/02-scraping-with-rvest.r>

Inspect element



Notes

Brief description of experiment.

Please note that to use the ExpressionSets below, you will need to install the `library(Biobase)` and run the command `library(Biobase)`

» The Datasets

Study	PMID	Species	Number of biological replicates	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19		link	link	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	20856902	human	41	834,584,950	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	liver; males and females
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled	original pooled	experiment: MAQC-2
montgomery	20220756	human	60	*886,468,054	link	link	HapMap - CEU
pickrell	20220758	human	69	*886,468,054	link	link	HapMap - YRI

Back

Forward

Reload

Save As...

Print...

Translate to English

View Page Source

View Page Info

P Paperpile ▶

Inspect Element

Copy XPath

ReCount: analysis-ready R | x

bowtie-bio.sourceforge.net/recount/

Please note that to use the ExpressionSets below, you will need to install [Bioconductor](#) and run the command `library(Biobase)`

The Datasets 828px x 993px

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	22496456	human	19	2,197,622,796	link	link	link	Illumina Human BodyMap 2.0 -- tissue comparison
bottomly	21455293	mouse	21	343,445,340	link	link	link	2 inbred mouse strains
cheung	20856902	human	41	834,584,950	link	link	link	HapMap - CEU
core	19056941	human	2	8,670,342	link	link	link	lung fibroblasts
gilad	20009012	human	6	41,356,738	link	link	link	liver; males and females

Elements Network Sources Timeline Profiles Resources Audits Console

Please note that to use the ExpressionSets below, you will need to install "Bioconductor" and run the command "library(Biobase)".

Copy XPath

Styles Computed Event Listeners »

```
element.style {  
}  
  
media="screen" bowtie-bio.sourceforge.net/style.css:226  
#recounttab table {  
  margin: 1em;  
  margin-top: 15px;  
  border-collapse: collapse;
```


rvest package

```
recount_url = "http://bowtie-bio.sourceforge.net/recount/"  
# install.packages("rvest")  
library(rvest)  
htmlfile = read_html(recount_url)  
  
nds = html_nodes(htmlfile,  
  xpath='//*[@id="recounttab"]/table')  
dat = html_table(nds)  
dat = as.data.frame(dat)  
head(dat)
```

70,000 OkCupid Users Just Had Their Data Published

WRITTEN BY JOSEPH COX

May 12, 2016 // 12:44 PM EST



A student and a co-researcher have publicly released a dataset on nearly 70,000 users of the dating site OkCupid, including their sexual turn-ons, orientation, usernames and more. And critics say it may be possible to work out users' real identities from the published data.

The situation is raising questions about what type of data researchers should be allowed to collect en masse, repackage and perhaps distribute.

Information posted to OkCupid is semi-public: you can discover some profiles with a Google search if you type in a person's username, and see some of the information they've provided, but not all of it. In order to do that, you need to log into the site.

<http://motherboard.vice.com/read/70000-okcupid-users-just-had-their-data-published>

Peer review
and scientific
publishing

Text mining: what do publishers have against this hi-tech research tool?

Researchers push for end to publishers' default ban on computer scanning of tens
of thousands of papers to find links between genes and diseases



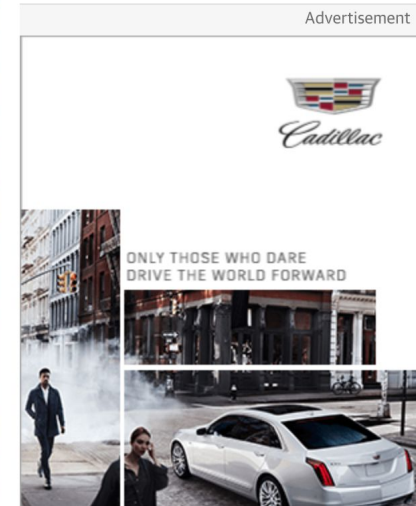
Alok Jha, Science
correspondent

Wednesday 23 May 2012
11.27 EDT



Shares
4

Comments
62

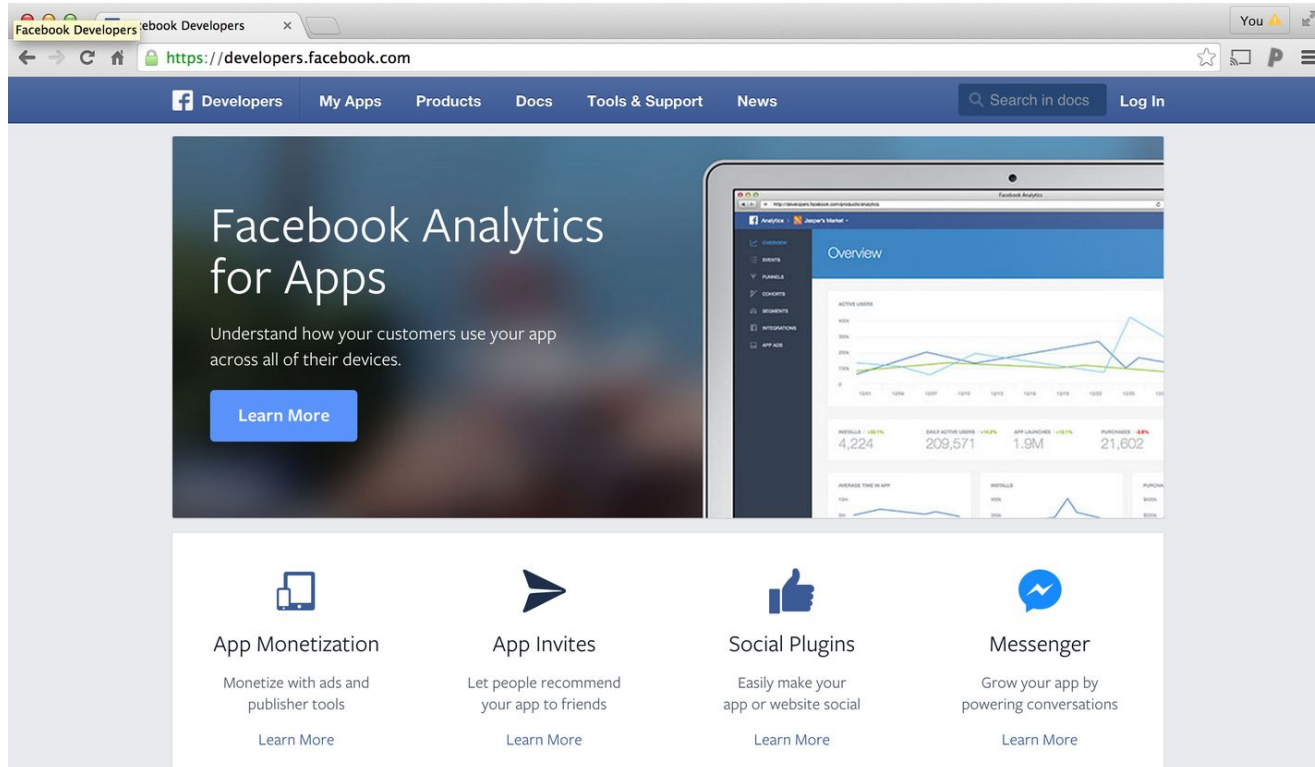


<https://www.theguardian.com/science/2012/may/23/text-mining-research-tool-forbidden>

APIs

Application Programming Interfaces

<https://developers.facebook.com/>



The screenshot shows the Facebook Developers website. The browser's address bar displays <https://developers.facebook.com/>. The navigation bar includes links for Developers, My Apps, Products, Docs, Tools & Support, and News, along with a search bar and a Log In button. The main content area features a large banner for "Facebook Analytics for Apps" with the text "Understand how your customers use your app across all of their devices." and a "Learn More" button. To the right of the banner is a preview of the Facebook Analytics dashboard, which includes a line chart for "ACTIVE USERS" and a table of key metrics.




Metric	Value	Change
INSTALLS	4,224	+15.2%
DAILY ACTIVE USERS	209,571	+15.2%
APP LAUNCHES	1.9M	+15.2%
PURCHASES	21,602	+15.2%

Below the banner, there are four featured sections, each with an icon, a title, a description, and a "Learn More" link:

- App Monetization**: Monetize with ads and publisher tools. [Learn More](#)
- App Invites**: Let people recommend your app to friends. [Learn More](#)
- Social Plugins**: Easily make your app or website social. [Learn More](#)
- Messenger**: Grow your app by powering conversations. [Learn More](#)

In biology too!

<http://www.ncbi.nlm.nih.gov/books/NBK25501/>

 NCBI [Resources](#)  [How To](#) 

[jleek2@era](#) [My NCBI](#) [Sign Out](#)

Bookshelf

Books 

Search

[Browse Titles](#) [Limits](#) [Advanced](#)[Help](#)

Entrez Programming Utilities Help

Bethesda (MD): [National Center for Biotechnology Information \(US\)](#); 2010-.

[Copyright and Permissions](#)

Search this book

< Prev

Next >

PubReader format:
click here to try

Views 

[PubReader](#)

[Print View](#)

[Cite this Page](#)

Other titles in this collection 

[NCBI Help Manual](#)

Related information 

[NLM Catalog](#)

Recent Activity 

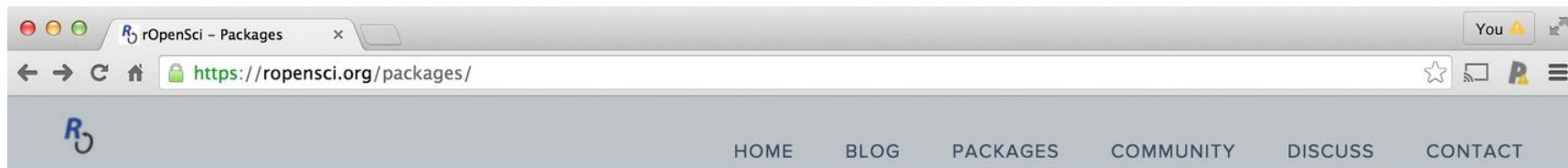
Introduction to the E-utilities

-  [E-utilities Introduction](#)
- Please see the [Release Notes](#) for details and changes.


The Entrez Programming Utilities (E-utilities) are a set of eight server-side programs that provide a stable interface into the Entrez query and database system at the National Center for Biotechnology Information (NCBI). The E-utilities use a fixed URL syntax that translates a standard set of input parameters into the values necessary for various

Step 0: Did someone do this already

<https://ropensci.org/>

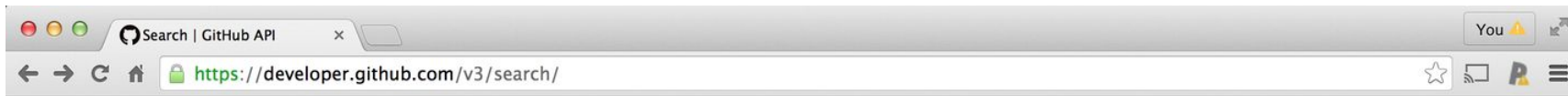


rOpenSci packages

This is a complete list of all available rOpenSci packages. Packages are grouped by ones that acquire [data](#), [full-text of journal articles](#), [altmetrics](#), [data-publication](#), [reproducibility](#) and [data visualization](#). Packages with a  sign are stable versions that you can quickly install from your nearest mirror using `install.packages("PACKAGE_NAME")`. Others are in various stages of development (bleeding edge packages are not listed here) and you can learn more by following our [GitHub organization page](#). All of our software packages are open source. Please see package description files for more information on specific licenses. We also have a [package status dashboard](#) that updates periodically.

[Data Publication](#) | [Data Access](#) | [Literature](#) | [Altmetrics](#) | [Reproducibility](#) | [Databases](#) | [Data Visualization](#) |

Do it yourself



API

Reference Webhooks Guides Libraries

Search

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search

► Overview

► Activity

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

► Repositories

▼ Search

Read the docs

<https://developer.github.com/v3/>

API

Reference

Webhooks

Guides

Libraries

Search

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview

► Activity

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

► Repositories

▼ Search

[Repositories](#)

Read the docs

API

Reference

Webhooks

Guides

Libraries

Rate limit

The Search API has a custom rate limit. For requests using [Basic Authentication](#), [OAuth](#), or [client ID and secret](#), you can make up to 30 requests per minute. For unauthenticated requests, the rate limit allows you to make up to 10 requests per minute.

See the [rate limit documentation](#) for details on determining your current rate limit status.

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Pull Requests

► Repositories

▼ Search

Repositories

Read the docs

Example

Suppose you want to find the definition of the `addClass` function inside `jQuery`. Your query would look something like this:

```
https://api.github.com/search/code?q=addClass+in:file+language:js+repo:jquery/jquery
```

Here, we're searching for the keyword `addClass` within a file's contents. We're making sure that we're only looking in files where the language is JavaScript. And we're scoping the search to the `repo:jquery/jquery` repository.

Example

Suppose you want to find the definition of the `addClass` function inside [jQuery](#). Your query would look something like this:

```
https://api.github.com/search/code?q=addClass+in:file+language:js+repo:jquery/jquery
```

Here, we're searching for the keyword `addClass` within a file's contents. We're making sure that we're only looking in files where the language is JavaScript. And we're scoping the search to the `repo:jquery/jquery` repository.

A dissected example

The image is a screenshot of the GitHub API documentation page for the Search endpoint. The page has a dark header with navigation links: API, Reference, Webhooks, Guides, and Libraries. The main heading is 'Search'. Below it, there is a list of search types: i. Search repositories, ii. Search code, iii. Search issues, iv. Search users, and v. Text match metadata. A section titled 'About the Search API' explains that the API is optimized for finding specific items and provides up to 1,000 results per search. On the right side, there is a sidebar with a table of contents listing various API endpoints: Overview, Gists, Git Data, Issues, Miscellaneous, Organizations, Pull Requests, Repositories, and Search. The 'Search' endpoint is currently selected, and its sub-section 'Repositories' is visible below it.

API

Reference Webhooks Guides Libraries

Search

[i. Search repositories](#)

[ii. Search code](#)

[iii. Search issues](#)

[iv. Search users](#)

[v. Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

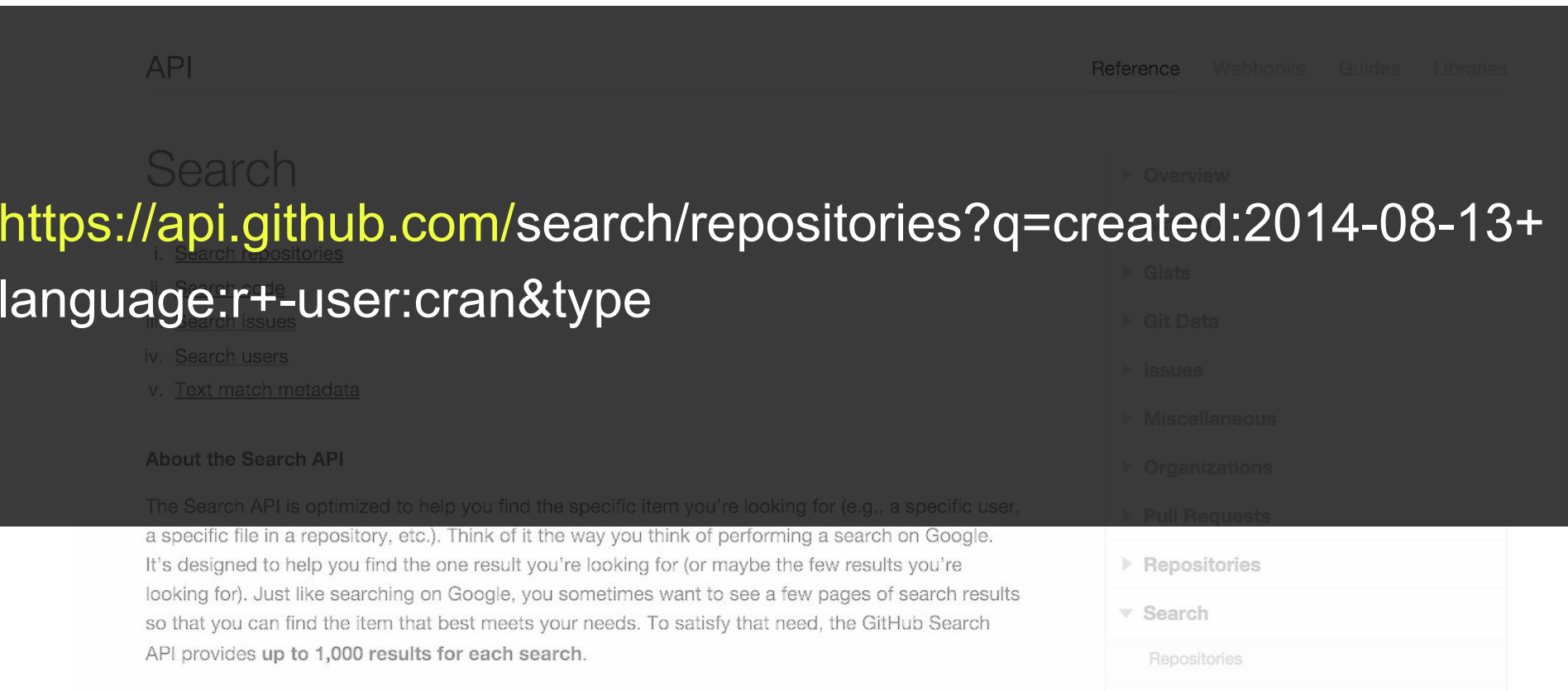
► Repositories

▼ Search

Repositories

<https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type>

The base URL



The image is a screenshot of the GitHub API documentation page for the Search endpoint. The page has a dark header with navigation links: API, Reference, Webhooks, Guides, and Libraries. The main content area is titled 'Search' and lists five search types: i. Search repositories, ii. Search code, iii. Search issues, iv. Search users, and v. Text match metadata. Below this is a section 'About the Search API' which explains that the API is optimized for finding specific items and provides up to 1,000 results per search. On the right side, there is a sidebar with a table of contents listing various API endpoints: Overview, Gists, Git Data, Issues, Miscellaneous, Organizations, Pull Requests, Repositories, and Search. The 'Search' entry is expanded, showing 'Repositories' as a sub-item. Overlaid on the center of the screenshot is a large URL in yellow and white text: `https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type`.

API Reference Webhooks Guides Libraries

Search

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

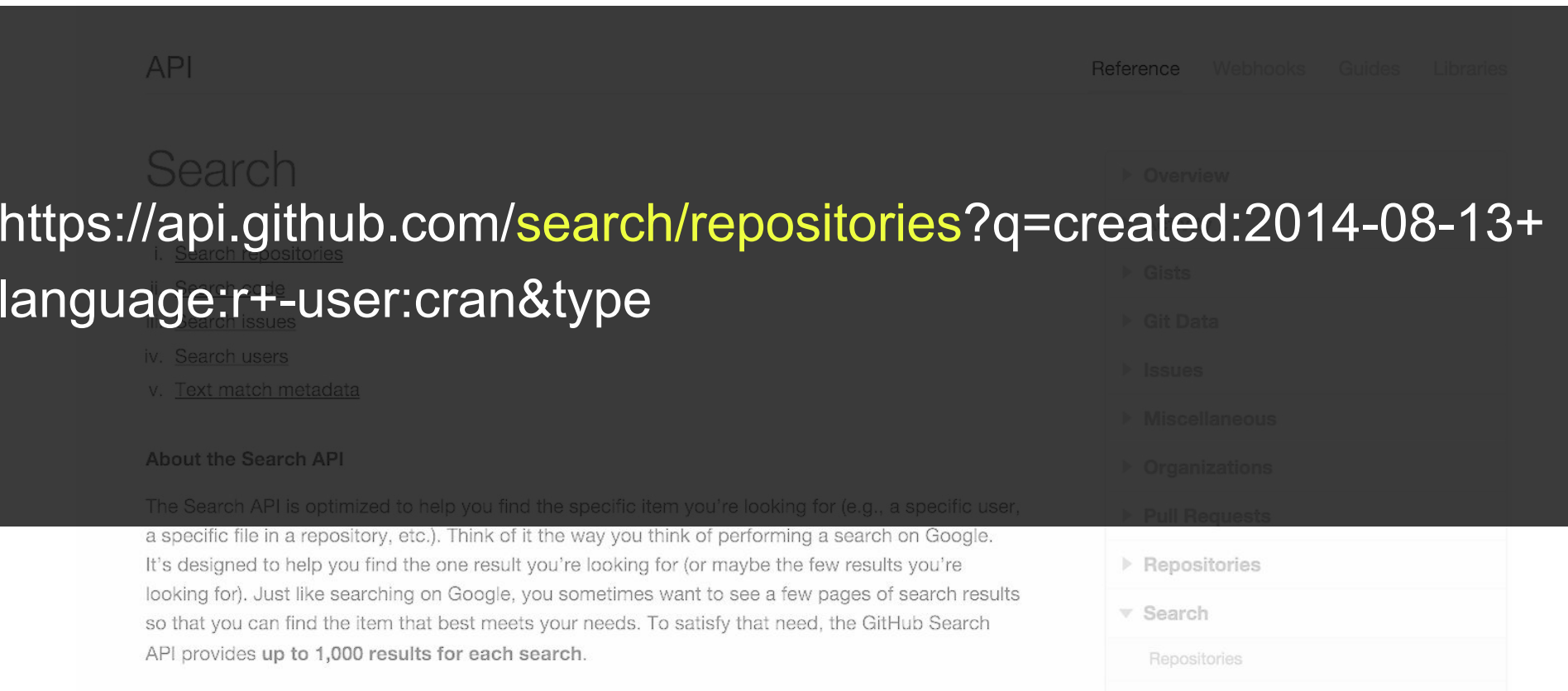
About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview
► Gists
► Git Data
► Issues
► Miscellaneous
► Organizations
► Pull Requests
► Repositories
▼ Search
Repositories

`https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type`

Search repositories



The image is a screenshot of the GitHub API documentation page for the Search repositories endpoint. The page has a dark header with navigation links: API, Reference, Webhooks, Guides, and Libraries. The main content area is divided into two columns. The left column contains the endpoint URL, a list of search filters, and a section titled 'About the Search API'. The right column contains a table of contents with links to Overview, Gists, Git Data, Issues, Miscellaneous, Organizations, Pull Requests, Repositories, and Search. The URL is `https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type`. The search filters are: i. Search repositories, ii. Search code, iii. Search issues, iv. Search users, and v. Text match metadata. The 'About the Search API' section explains that the API is optimized to help find specific items and provides up to 1,000 results for each search.

API

Reference Webhooks Guides Libraries

Search

`https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type`

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

- ▶ [Overview](#)
- ▶ [Gists](#)
- ▶ [Git Data](#)
- ▶ [Issues](#)
- ▶ [Miscellaneous](#)
- ▶ [Organizations](#)
- ▶ [Pull Requests](#)
- ▶ [Repositories](#)
- ▼ [Search](#)
 - [Repositories](#)

Create a query

The image is a screenshot of the GitHub API documentation page for the Search endpoint. The page has a dark header with navigation links: API, Reference, Webhooks, Guides, and Libraries. The main content area is titled 'Search' and lists five search types: i. Search repositories, ii. Search code, iii. Search issues, iv. Search users, and v. Text match metadata. Below this is a section 'About the Search API' which explains that the Search API is optimized to help find specific items and provides up to 1,000 results for each search. On the right side, there is a sidebar with a table of contents listing various API endpoints: Overview, Gists, Git Data, Issues, Miscellaneous, Organizations, Pull Requests, Repositories, and Search. The 'Search' endpoint is currently selected, and its sub-section 'Repositories' is visible below it.

API Reference Webhooks Guides Libraries

Search

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

Search

- Repositories

https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type

Date repo was created

The image is a screenshot of the GitHub API documentation page for the Search endpoint. The page has a dark theme. At the top, there are navigation links: API, Reference, Webhooks, Guides, and Libraries. The main heading is "Search". Below it, there is a list of search types: i. Search repositories, ii. Search code, iii. Search issues, iv. Search users, and v. Text match metadata. To the right of the main content, there is a sidebar with a list of search types: Overview, Gists, Git Data, Issues, Miscellaneous, Organizations, and Pull Requests. Below the sidebar, there is a section for "Repositories" and "Search". The "Search" section is expanded, showing "Repositories".

API

Reference Webhooks Guides Libraries

Search

- i. [Search repositories](#)
- ii. [Search code](#)
- iii. [Search issues](#)
- iv. [Search users](#)
- v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

► Repositories

▼ Search

Repositories

<https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type>

Language repo is in

<https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type>

API

Reference

Webhooks

Guides

Libraries

Search

i. [Search repositories](#)

ii. [Search code](#)

iii. [Search issues](#)

iv. [Search users](#)

v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

► Repositories

▼ Search

Repositories

Ignore repos from "cran"

<https://api.github.com/search/repositories?q=created:2014-08-13+language:r+-user:cran&type>

API

Reference

Webhooks

Guides

Libraries

Search

i. [Search repositories](#)

ii. [Search code](#)

iii. [Search issues](#)

iv. [Search users](#)

v. [Text match metadata](#)

About the Search API

The Search API is optimized to help you find the specific item you're looking for (e.g., a specific user, a specific file in a repository, etc.). Think of it the way you think of performing a search on Google. It's designed to help you find the one result you're looking for (or maybe the few results you're looking for). Just like searching on Google, you sometimes want to see a few pages of search results so that you can find the item that best meets your needs. To satisfy that need, the GitHub Search API provides **up to 1,000 results for each search**.

► Overview

► Gists

► Git Data

► Issues

► Miscellaneous

► Organizations

► Pull Requests

► Repositories

▼ Search

Repositories

```
#install.packages("httr")
```

```
library(httr)
```

```
query_url =
```

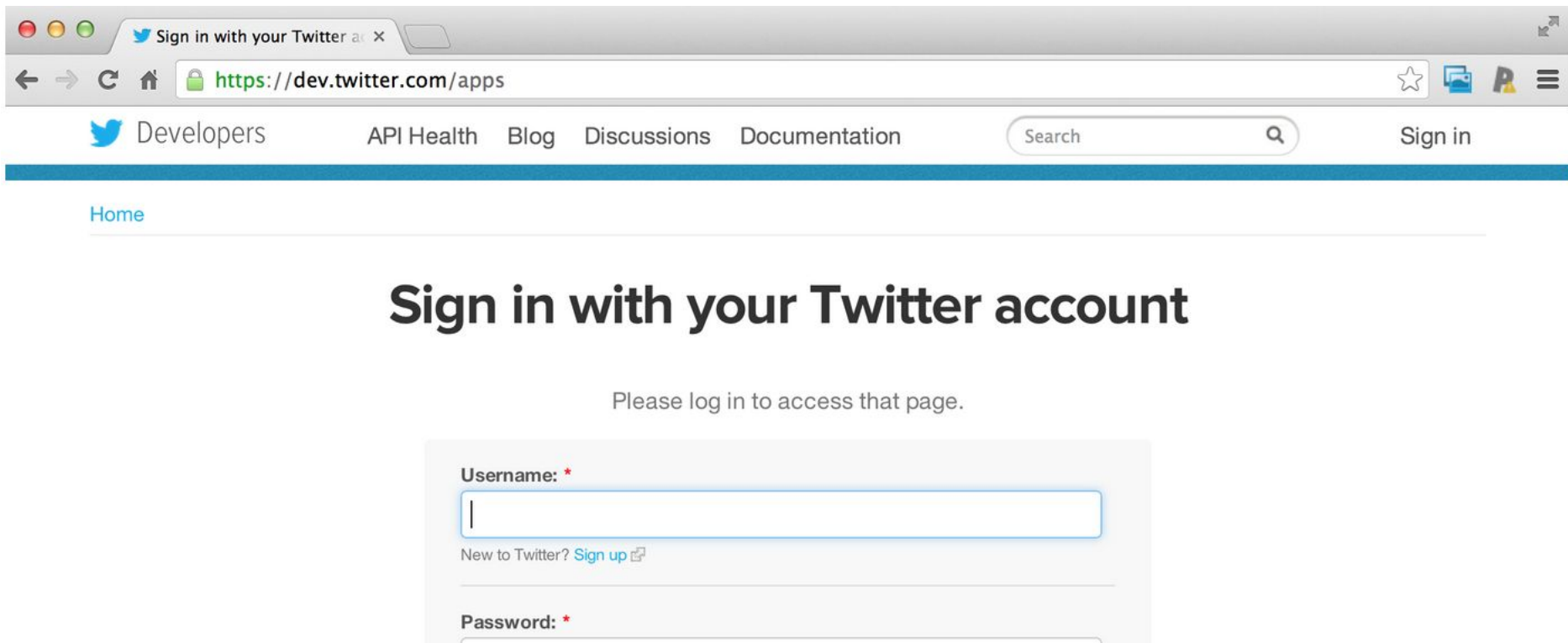
```
"https://api.github.com/search/repositories?q=created:2014-08-13+language  
:r+-user:cran"
```

```
req = GET(query_url)
```

```
names(content(req))
```

Not all APIs are “open”

<https://apps.twitter.com/>



The screenshot shows a web browser window with the address bar displaying `https://dev.twitter.com/apps`. The page header includes navigation links: Developers, API Health, Blog, Discussions, and Documentation, along with a search bar and a 'Sign in' button. The main content area features a large heading 'Sign in with your Twitter account' and a message 'Please log in to access that page.' Below this is a sign-in form with fields for 'Username' and 'Password', both marked with a red asterisk. A link for 'New to Twitter? Sign up' is also present.

Sign in with your Twitter account

Please log in to access that page.

Username: *

New to Twitter? [Sign up](#)

Password: *

```
myapp = oauth_app("twitter",  
                  key="yourConsumerKeyHere",secret="yourConsumerSecretHere")  
sig = sign_oauth1.0(myapp,  
                    token = "yourTokenHere",  
                    token_secret = "yourTokenSecretHere")  
homeTL = GET("https://api.twitter.com/1.1/statuses/home_timeline.json", sig)
```

But you can get cool data

```
json1 = content(homeTL)
json2 = jsonlite::fromJSON(toJSON(json1))
json2[1,1:4]
```

	created_at	id	id_str
1	Mon Jan 13 05:18:04 +0000 2014	4.225984e+17	422598398940684288

```
text
1 Now that P. Norvig's regex golf IPython notebook hit Slashdot, let's see
if our traffic spike tops the previous one: http://t.co/Vc6JhZXOo8
```

Web + APIs lab

<https://bit.ly/2Jlwllt>