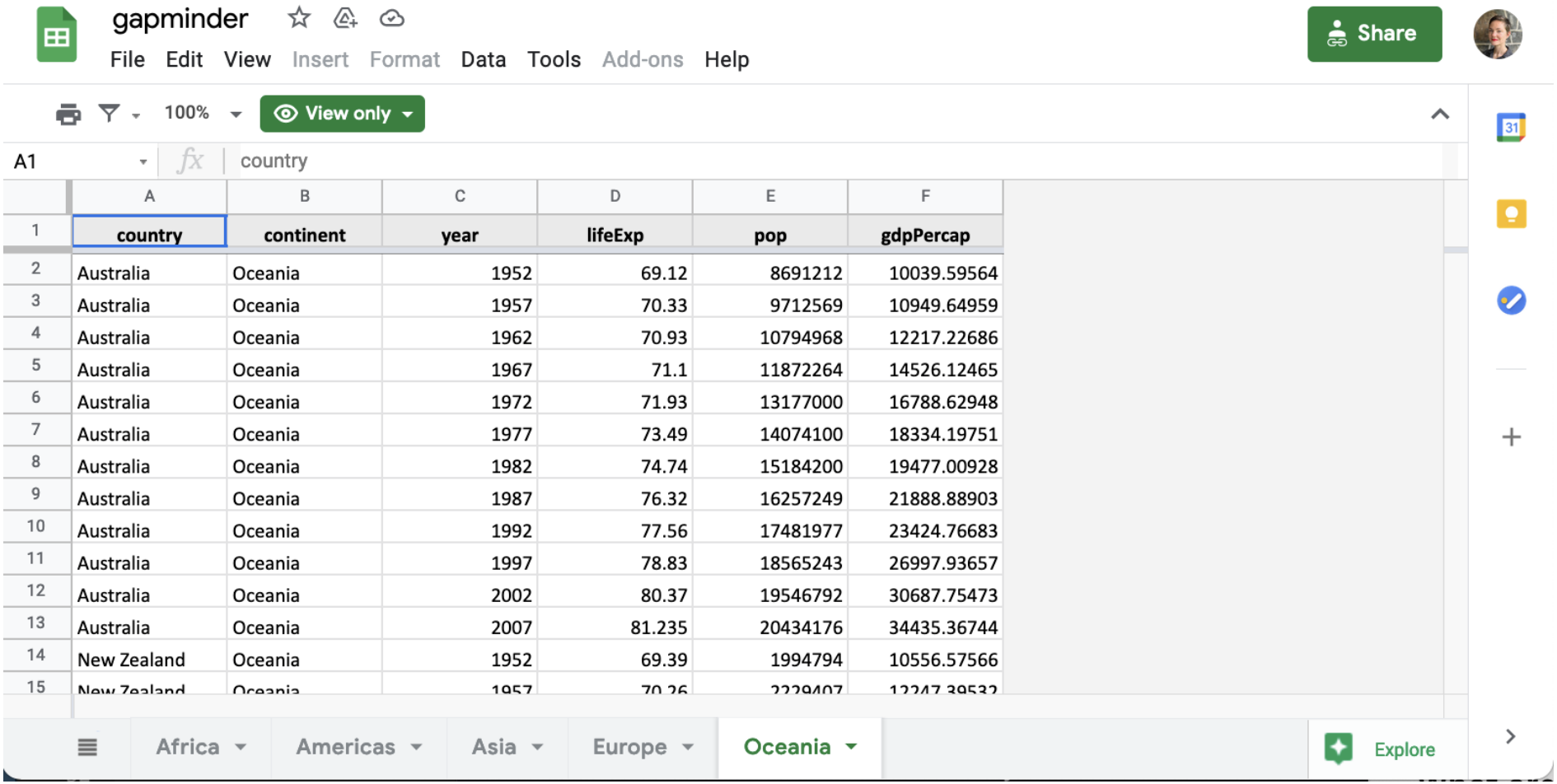
Advanced Data IO

Data Wrangling in R

Google Sheets



	A	B	C	D	E	F
1	country	continent	year	lifeExp	pop	gdpPercap
2	Australia	Oceania	1952	69.12	8691212	10039.59564
3	Australia	Oceania	1957	70.33	9712569	10949.64959
4	Australia	Oceania	1962	70.93	10794968	12217.22686
5	Australia	Oceania	1967	71.1	11872264	14526.12465
6	Australia	Oceania	1972	71.93	13177000	16788.62948
7	Australia	Oceania	1977	73.49	14074100	18334.19751
8	Australia	Oceania	1982	74.74	15184200	19477.00928
9	Australia	Oceania	1987	76.32	16257249	21888.88903
10	Australia	Oceania	1992	77.56	17481977	23424.76683
11	Australia	Oceania	1997	78.83	18565243	26997.93657
12	Australia	Oceania	2002	80.37	19546792	30687.75473
13	Australia	Oceania	2007	81.235	20434176	34435.36744
14	New Zealand	Oceania	1952	69.39	1994794	10556.57566
15	New Zealand	Oceania	1957	70.26	2229407	12247.39532

https://docs.google.com/spreadsheets/d/1U6Cf_qEOhiR9AZqTqS3mbMF3zt2db48ZP5v

Reading data with the googlesheets4 package

First, set up Google credentials

```
library(googlesheets4)
```

```
# Prompts a browser pop-up
gs4_auth()
```

```
# Once set up, you can automate this process by passing your email
gs4_auth(email = "avamariehoffman@gmail.com")
```

Reading data with the googlesheets4 package

You can also supply an authorization token directly, but make sure to add any files to your .gitignore!

```
library(googledrive)
drive_auth(email= "<email>",
           token = readRDS("google-sheets-token.rds")) # Saved in a file
```

Reading data with the googlesheets4 package

Read in using `googlesheets4::read_sheet`

```
sheet_url = paste0("https://docs.google.com/spreadsheets/d/1U6Cf_qEOhiR9",
                   "AZqTqS3mbMF3zt2db48ZP5v3rkrAEJY/edit#gid=780868077")
sheet_dat_1 = read_sheet(sheet_url)
```

```
Reading from "gapminder"

Range "Africa"
```

```
head(sheet_dat_1)
```

```
# A tibble: 6 x 6
  country continent  year lifeExp      pop gdpPercap
  <chr>   <chr>     <dbl>   <dbl>    <dbl>     <dbl>
1 Algeria Africa     1952    43.1  9279525     2449.
2 Algeria Africa     1957    45.7 10270856     3014.
3 Algeria Africa     1962    48.3 11000948     2551.
4 Algeria Africa     1967    51.4 12760499     3247.
5 Algeria Africa     1972    54.5 14760787     4183.
6 Algeria Africa     1977    58.0 17152804     4910.
```

Reading data with the googlesheets4 package

Specify the sheet name if necessary:

```
sheet_url = paste0("https://docs.google.com/spreadsheets/d/1U6Cf_qEOhiR9",
                   "AZqTqS3mbMF3zt2db48ZP5v3rkrAEJY/edit#gid=780868077")
sheet_dat_oceania = read_sheet(sheet_url, sheet = "Oceania")
```

```
Reading from "gapminder"

Range "'Oceania'"
```

```
head(sheet_dat_oceania)
```

```
# A tibble: 6 x 6
  country   continent  year lifeExp      pop gdpPercap
  <chr>     <chr>     <dbl>   <dbl>    <dbl>     <dbl>
1 Australia Oceania    1952    69.1  8691212    10040.
2 Australia Oceania    1957    70.3  9712569    10950.
3 Australia Oceania    1962    70.9 10794968    12217.
4 Australia Oceania    1967    71.1 11872264    14526.
5 Australia Oceania    1972    71.9 13177000    16789.
6 Australia Oceania    1977    73.5 14074100    18334.
```

Reading data with the googlesheets4 package

Alternatively, list out the sheet names using `sheet_names()`.

```
sheet_names(sheet_url)
```

```
[1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

Reading data with the googlesheets4 package

Iterate through the sheet names:

```
gapminder_sheets = sheet_names(sheet_url)

data_list = list()
for(g_sheet in gapminder_sheets){
  data_list[[g_sheet]] = read_sheet(sheet_url, sheet = g_sheet)
}
```

```
Reading from "gapminder"

Range "'Africa'"

Reading from "gapminder"

Range "'Americas'"

Reading from "gapminder"

Range "'Asia'"

Reading from "gapminder"

Range "'Europe'"

Reading from "gapminder"
```

Reading data with the googlesheets4 package

Iterate through the sheet names:

```
str(data_list)
```

```
List of 5
 $ Africa  : tibble [624 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ country  : chr [1:624] "Algeria" "Algeria" "Algeria" "Algeria" ...
  ..$ continent: chr [1:624] "Africa" "Africa" "Africa" "Africa" ...
  ..$ year     : num [1:624] 1952 1957 1962 1967 1972 ...
  ..$ lifeExp  : num [1:624] 43.1 45.7 48.3 51.4 54.5 ...
  ..$ pop      : num [1:624] 9279525 10270856 11000948 12760499 14760787 ...
  ..$ gdpPercap: num [1:624] 2449 3014 2551 3247 4183 ...
 $ Americas: tibble [300 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ country  : chr [1:300] "Argentina" "Argentina" "Argentina" "Argentina" .
  ..$ continent: chr [1:300] "Americas" "Americas" "Americas" "Americas" ...
  ..$ year     : num [1:300] 1952 1957 1962 1967 1972 ...
  ..$ lifeExp  : num [1:300] 62.5 64.4 65.1 65.6 67.1 ...
  ..$ pop      : num [1:300] 17876956 19610538 21283783 22934225 24779799 ...
  ..$ gdpPercap: num [1:300] 5911 6857 7133 8053 9443 ...
 $ Asia    : tibble [396 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ country  : chr [1:396] "Afghanistan" "Afghanistan" "Afghanistan" "Afghar
  ..$ continent: chr [1:396] "Asia" "Asia" "Asia" "Asia" ...
  ..$ year     : num [1:396] 1952 1957 1962 1967 1972 ...
  ..$ lifeExp  : num [1:396] 28.8 30.3 32 34 36.1 ...
  ..$ pop      : num [1:396] 8425333 9240934 10267083 11537966 13079460 ...
  ..$ gdpPercap: num [1:396] 779 821 853 836 740 ...
 $ Europe  : tibble [360 × 6] (S3: tbl_df/tbl/data.frame)
  ..$ country  : chr [1:360] "Albania" "Albania" "Albania" "Albania" ...
```

Writing data with the googlesheets4 package

```
sheet_dat_oceania = sheet_dat_oceania %>%
  mutate(lifeExp_days = lifeExp * 365)
sheet_out = gs4_create("Oceania-days",
                       sheets = list(Oceania_days = sheet_dat_oceania))
```

```
Creating new Sheet: "Oceania-days"
```

```
# Opens a browser window
gs4_browse(sheet_out)
```

Append data with the googlesheets4 package

```
sheet_append(sheet_out, data = sheet_dat_oceania, sheet = "Oceania_days")
```

```
Writing to "Oceania-days"
```

```
Appending 24 row(s) to "Oceania_days"
```

JHU Tidyverse Book

https://jhudatascience.org/tidyversecourse/get-data.html#google-sheets

googlesheets Lab

http://sisbid.github.io/Data-Wrangling/labs/advanced-io-lab.Rmd

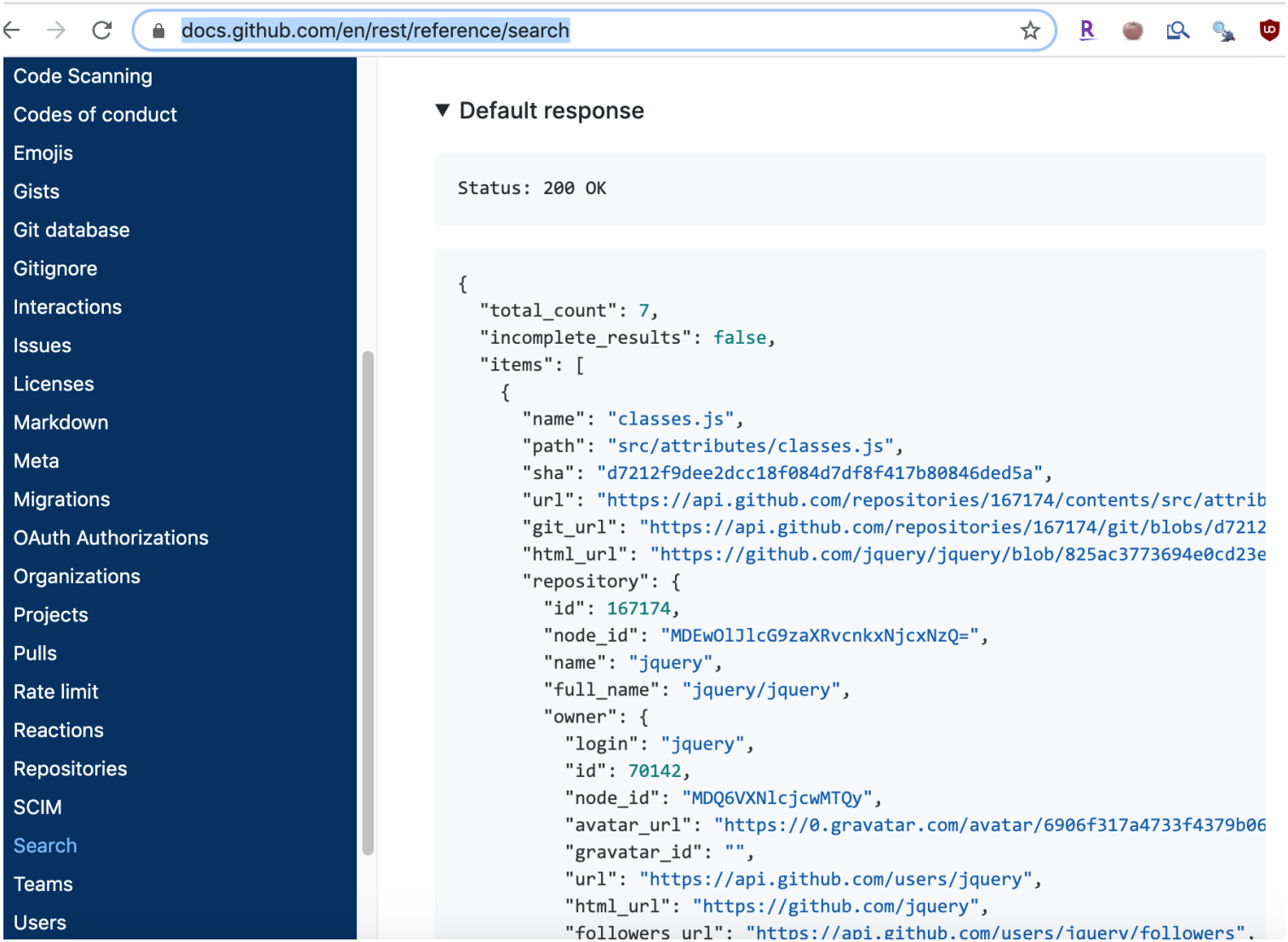
JSON: JavaScript Object Notation
Lists of stuff

Example [edit]

The following example shows a possible JSON representation describing a person.

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "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://docs.github.com/en/rest/reference/search

```
#install.packages("jsonlite")
library(jsonlite)
jsonData <- fromJSON(paste0("https://raw.githubusercontent.com/Biuni",
                            "/PokemonGO-Pokedex/master/pokedex.json"))
head(jsonData)
```

```
$pokemon
     id num                  name                                                im
1     1 001             Bulbasaur http://www.serebii.net/pokemongo/pokemon/001.pn
2     2 002               Ivysaur http://www.serebii.net/pokemongo/pokemon/002.pn
3     3 003              Venusaur http://www.serebii.net/pokemongo/pokemon/003.pn
4     4 004            Charmander http://www.serebii.net/pokemongo/pokemon/004.pn
5     5 005            Charmeleon http://www.serebii.net/pokemongo/pokemon/005.pn
6     6 006             Charizard http://www.serebii.net/pokemongo/pokemon/006.pn
7     7 007              Squirtle http://www.serebii.net/pokemongo/pokemon/007.pn
8     8 008             Wartortle http://www.serebii.net/pokemongo/pokemon/008.pn
9     9 009             Blastoise http://www.serebii.net/pokemongo/pokemon/009.pn
10   10 010              Caterpie http://www.serebii.net/pokemongo/pokemon/010.pn
11   11 011                Metapod http://www.serebii.net/pokemongo/pokemon/011.pn
12   12 012            Butterfree http://www.serebii.net/pokemongo/pokemon/012.pn
13   13 013                Weedle http://www.serebii.net/pokemongo/pokemon/013.pn
14   14 014                Kakuna http://www.serebii.net/pokemongo/pokemon/014.pn
15   15 015              Beedrill http://www.serebii.net/pokemongo/pokemon/015.pn
16   16 016                Pidgey http://www.serebii.net/pokemongo/pokemon/016.pn
17   17 017             Pidgeotto http://www.serebii.net/pokemongo/pokemon/017.pn
18   18 018               Pidgeot http://www.serebii.net/pokemongo/pokemon/018.pn
19   19 019               Rattata http://www.serebii.net/pokemongo/pokemon/019.pn
20   20 020              Raticate http://www.serebii.net/pokemongo/pokemon/020.pn
21   21 021               Spearow http://www.serebii.net/pokemongo/pokemon/021.pn
22   22 022                Fearow http://www.serebii.net/pokemongo/pokemon/022.pn
23   23 023                 Ekans http://www.serebii.net/pokemongo/pokemon/023.pn
```

Data frame structure from JSON

```
dim(jsonData$pokemon)
```

```
[1] 151  17
```

```
class(jsonData$pokemon$type) # Can be lists
```

```
[1] "list"
```

```
jsonData$pokemon$type
```

```
[[1]]
[1] "Grass"  "Poison"

[[2]]
[1] "Grass"  "Poison"

[[3]]
[1] "Grass"  "Poison"

[[4]]
[1] "Fire"

[[5]]
[1] "Fire"

[[6]]
[1] "Fire"   "Flying"
```

Data frame structure from JSON

```
class(jsonData$pokemon$next_evolution[[1]]) # Or lists of data.frames!
```

```
[1] "data.frame"
```

```
jsonData$pokemon$next_evolution
```

```
[[1]]
  num     name
1 002  Ivysaur
2 003 Venusaur

[[2]]
  num     name
1 003 Venusaur

[[3]]
NULL

[[4]]
  num       name
1 005 Charmeleon
2 006  Charizard

[[5]]
  num      name
1 006 Charizard

[[6]]
```

JSON Lab

http://sisbid.github.io/Data-Wrangling/labs/advanced-io-lab.Rmd

Web Scraping

This is data

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

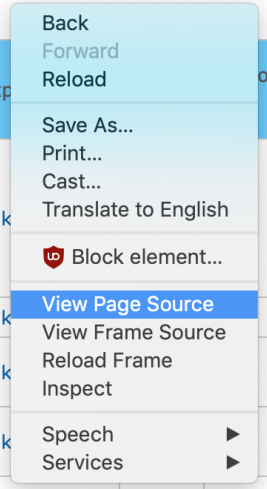
The Datasets

Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	ExpressionSet	Count table	Phenotype table	Notes
bodymap	not published, but publicly available here	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 femlaes
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
								control vs

View the source

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	Exp	otype	Notes
bodymap	not published, but publicly available here	human	19	2,197,622,796	link		Illumina Human BodyMap 2.0 - - tissue comparison
cheung	20856902	human	41	834,584,950	link		HapMap - CEU
core	19056941	human	2	8,670,342	link		lung fibroblasts
gilad	20009012	human	6	41,356,738	link		liver; males and femlaes
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

What the computer sees

Not Secure | view-source:bowtie-bio.sourceforge.net/recount/

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<script src="sorttable.js" type="text/javascript"></script>
<title>ReCount: analysis-ready RNA-seq gene count datasets</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />
<script type="text/javascript">

  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-26478269-2']);
  _gaq.push(['_trackPageview']);

  (function() {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();

</script>

</head>

<body class="c20">
<div id="wrap">
    <div id="top">
    <div class="lefts">
    <table width="100%" cellpadding="2">
      <tr><td>
        <a href="./index.shtml"><h1>ReCount</h1></a>
        <h2>A multi-experiment resource of analysis-ready RNA-seq gene count datasets</h2>
      </td><td align="right" valign="middle">
        <h1><a href="http://www.biostat.jhsph.edu/"><img border=0 src="images/jhsph_white.png"></img></a>  </h1>
      </td></tr>
    </table>
    </div>
  </div>

  <div id="subheader">
<p><b>There is now <a href="https://jhubiostatistics.shinyapps.io/recount/">a new version of recount</a> that provides processed and summarized express
data for nearly 60,000 human RNA-seq samples from the Sequence Read Archive (SRA). The <a href="https://github.com/leekgroup/recount">associated
Bioconductor package</a> provides a convenient API for querying, downloading, and analyzing the data. Each processed study consists of meta- and phenot
data, the expression levels of genes and their underlying exons and splice junctions, and corresponding genomic annotation.  See <a
```

Ways to see the source

Chrome:

1. right click on page
2. select “View Page Source”

Firefox:

1. right click on page
2. select “View Page Source”

Microsoft Edge:

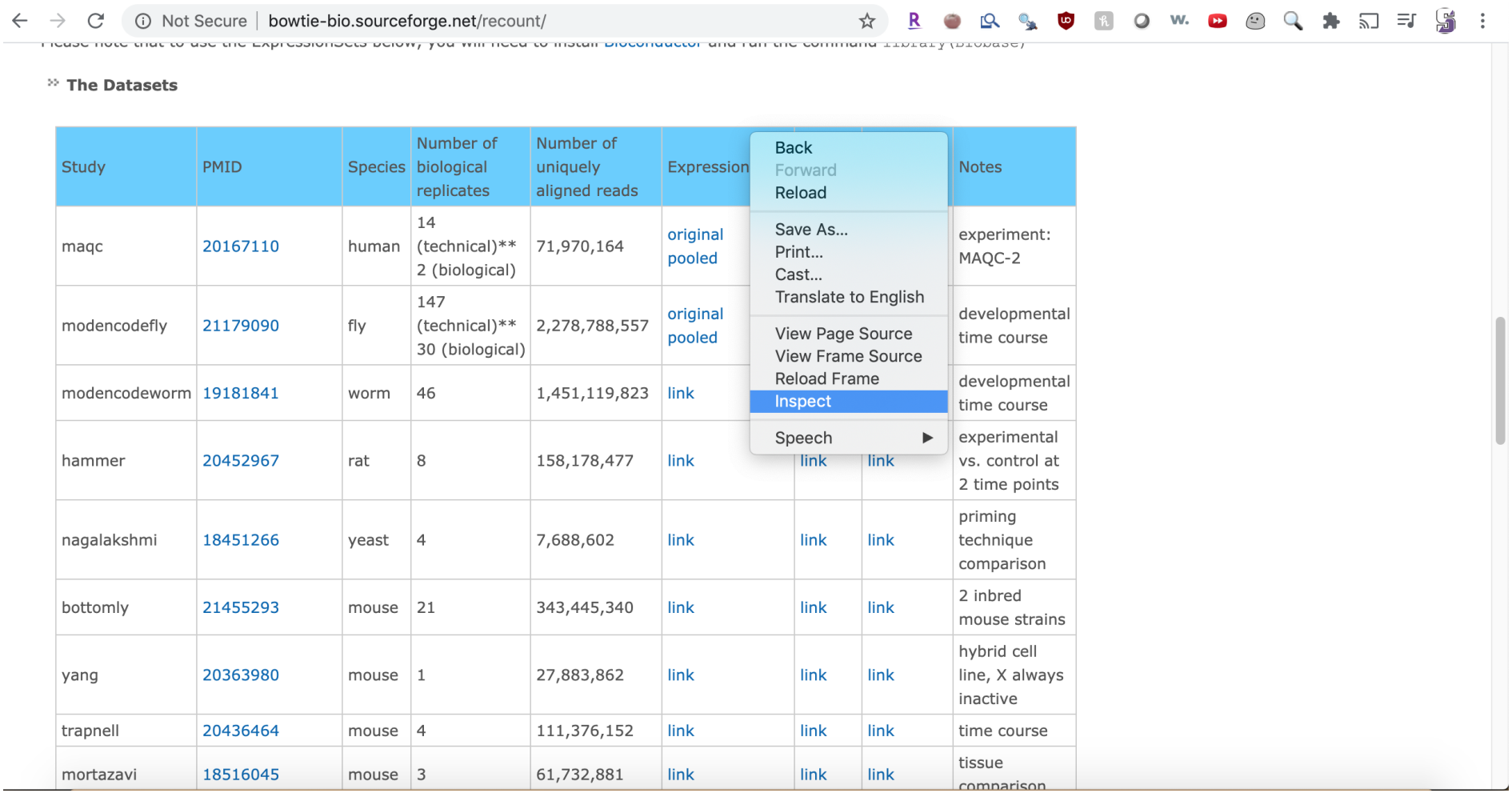
1. right click on page
2. select “view source”

Safari

1. click on “Safari”
2. select “Preferences”
3. go to “Advanced”
4. check “Show Develop menu in menu bar”
5. right click on page
6. select “View Page Source”

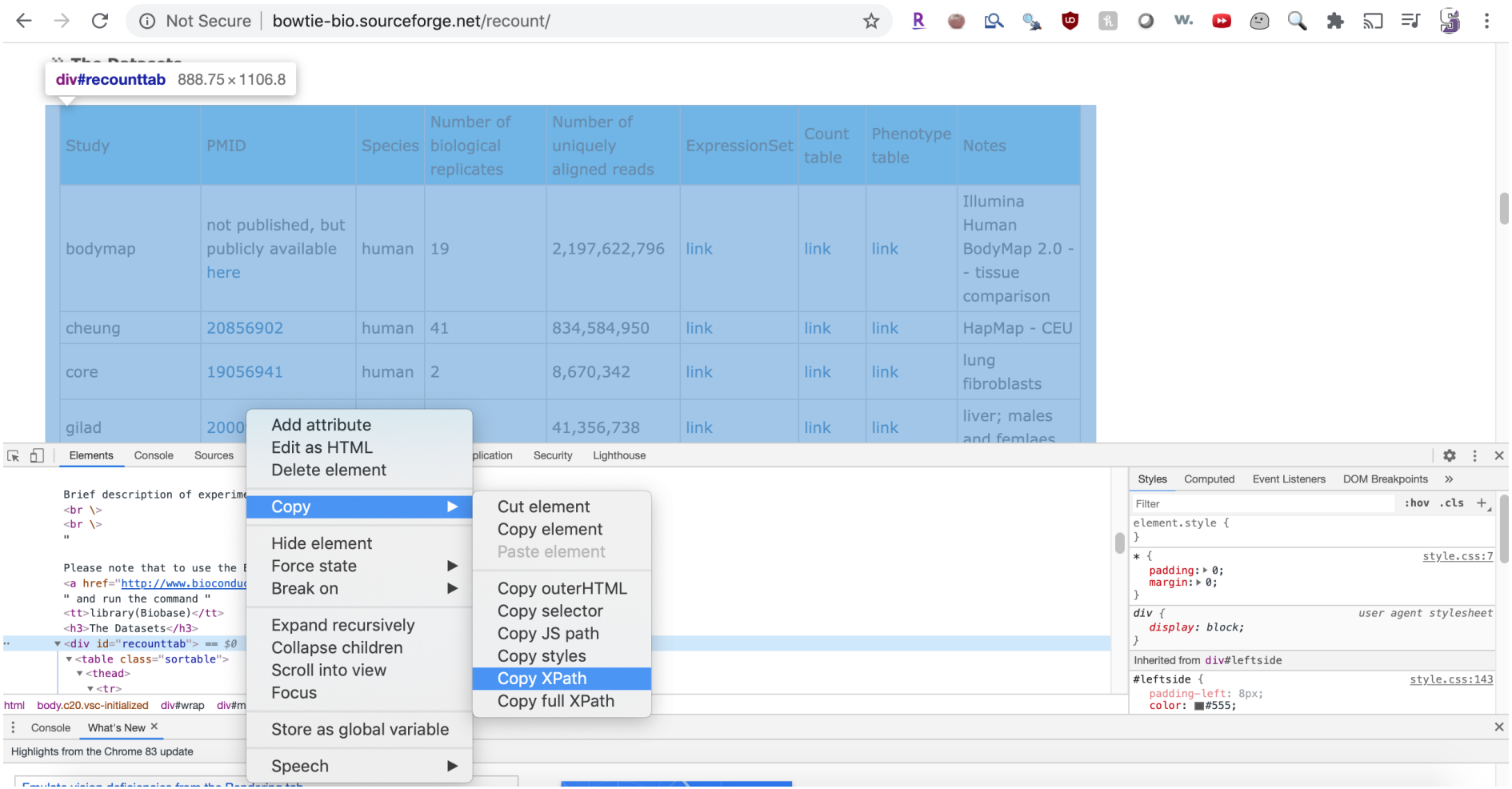
https://github.com/simonmunzert/rscraping-jsm-2016/blob/c04fd91fec711df65c838e07723125155a7f2cda/02-scraping-with-rvest.r

Inspect element



Study	PMID	Species	Number of biological replicates	Number of uniquely aligned reads	Expression			Notes
maqc	20167110	human	14 (technical)** 2 (biological)	71,970,164	original pooled			experiment: MAQC-2
modencodefly	21179090	fly	147 (technical)** 30 (biological)	2,278,788,557	original pooled			developmental time course
modencodeworm	19181841	worm	46	1,451,119,823	link			developmental time course
hammer	20452967	rat	8	158,178,477	link	link	link	experimental vs. control at 2 time points
nagalakshmi	18451266	yeast	4	7,688,602	link	link	link	priming technique comparison
bottomly	21455293	mouse	21	343,445,340	link	link	link	2 inbred mouse strains
yang	20363980	mouse	1	27,883,862	link	link	link	hybrid cell line, X always inactive
trapnell	20436464	mouse	4	111,376,152	link	link	link	time course
mortazavi	18516045	mouse	3	61,732,881	link	link	link	tissue comparison

Copy XPath


Not Secure | bowtie-bio.sourceforge.net/recount/
div#recounttab 888.75 × 1106.8
Study | PMID | Species | Number of biological replicates | Number of uniquely aligned reads | ExpressionSet | Count table | Phenotype table | Notes
bodymap | not published, but publicly available here | 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 | 2000 | | | 41,356,738 | link | link | link | liver; males and femlaes
Add attribute
Edit as HTML
Delete element
Copy
Hide element
Force state
Break on
Expand recursively
Collapse children
Scroll into view
Focus
Store as global variable
Speech
Cut element
Copy element
Paste element
Copy outerHTML
Copy selector
Copy JS path
Copy styles
Copy XPath
Copy full XPath
Elements
Console
Sources
Security
Lighthouse
Styles
Computed
Event Listeners
DOM Breakpoints
What's New
Highlights from the Chrome 83 update

Use SelectorGadget

https://rvest.tidyverse.org/articles/selectorgadget.html

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)
```

```
       X1                                                 X2       X3
1   Study                                               PMID Species
2 bodymap not published, but publicly available here    human
3  cheung                                           20856902   human
4    core                                           19056941   human
5   gilad                                           20009012   human
6    maqc                                           20167110   human
                              X4                                  X5
1  Number of biological replicates Number of uniquely aligned reads
2                               19                      2,197,622,796
3                               41                        834,584,950
4                                2                          8,670,342
5                                6                         41,356,738
6 14 (technical)**  2 (biological)                         71,970,164
               X6                X7                 X8
1   ExpressionSet     Count table Phenotype table
2            link              link              link
3            link              link              link
```

Little cleanup

```
colnames(dat) = as.character(dat[1,])
dat = dat[-1,]
head(dat)
```

```
       Study                                                PMID Species
2    bodymap not published, but publicly available here   human
3     cheung                                          20856902   human
4       core                                          19056941   human
5      gilad                                          20009012   human
6       maqc                                          20167110   human
7 montgomery                                          20220756   human
   Number of biological replicates Number of uniquely aligned reads
2                               19                      2,197,622,796
3                               41                        834,584,950
4                                2                          8,670,342
5                                6                         41,356,738
6 14 (technical)**  2 (biological)                         71,970,164
7                               60                       *886,468,054
    ExpressionSet     Count table Phenotype table
2            link            link            link
3            link            link            link
4            link            link            link
5            link            link            link
6 original pooled original pooled original pooled
7            link            link            link
                                              Notes
2 Illumina Human BodyMap 2.0 -- tissue comparison
3                                       HapMap - CEU
```

Ethics and Web Scraping

https://slate.com/culture/2020/04/whitney-museum-new-york-apartment-exhibit-creators-interview.html

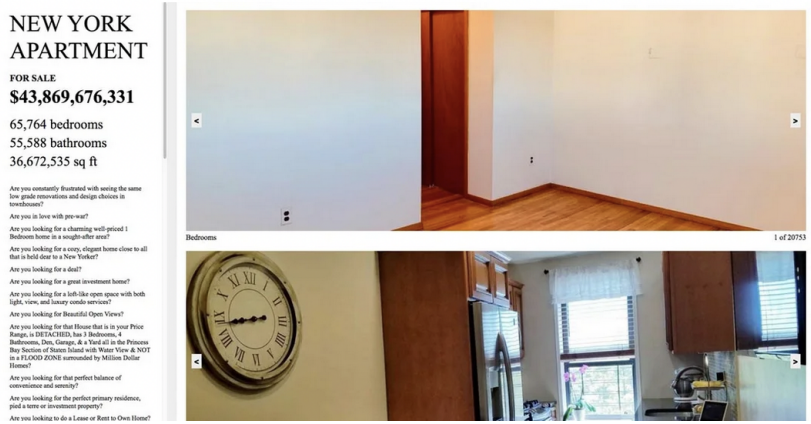
BROW BEAT

An Art Exhibit You Can Visit Without Leaving Your Couch

The creators of the Whitney Museum's *New York Apartment* explain how they combined thousands of listings into a website for one massive, $43.9 billion dwelling.

BY RACHELLE HAMPTON

APRIL 02, 2020 • 7:30 AM



Ethics and Web Scraping

https://doi.org/10.1016/j.dib.2020.106178

Contents lists available at ScienceDirect

Data in Brief

journal homepage: www.elsevier.com/locate/dib

Data Article

COVID-19: A scholarly production dataset report for research analysis

Breno Santana Santos[a,b,*], Ivanovitch Silva[a], Marcel da Câmara Ribeiro-Dantas[c], Gisliany Alves[a], Patricia Takako Endo[d], Luciana Lima[a]

[a] Universidade Federal do Rio Grande do Norte (UFRN), Rio Grande do Norte, Brazil
[b] Núcleo de Pesquisa e Prática em Inteligência Competitiva (NUPIC), Universidade Federal de Sergipe (UFS), Itabaiana, SE, Brazil
[c] Institut Curie (UMR168), Sorbonne Université (EDITE), Paris, France
[d] Universidade de Pernambuco (UPE), Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 7 July 2020
Revised 6 August 2020
Accepted 12 August 2020
Available online 19 August 2020

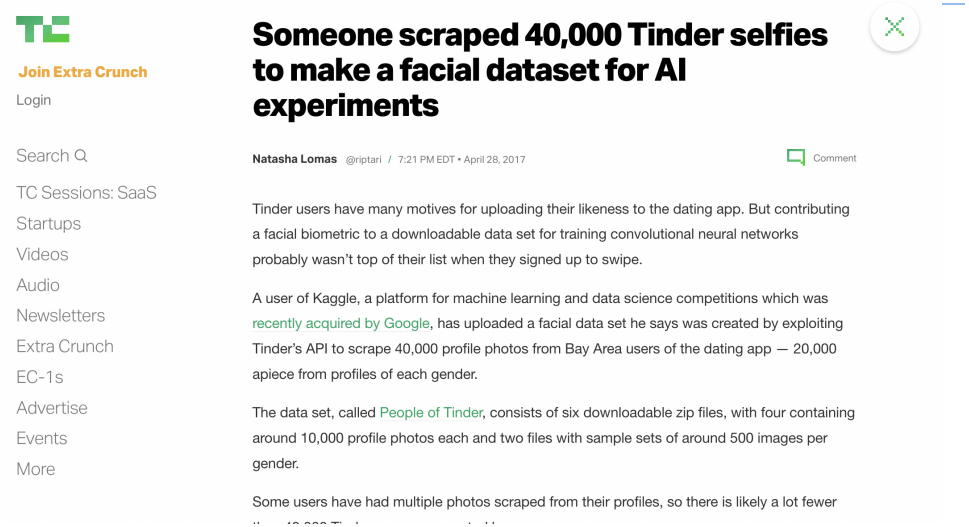
Keywords:
COVID-19
SARS-CoV-2
Pandemic
Data Science
Bibliometrics
Scientometrics

ABSTRACT

COVID-2019 has been recognized as a global threat, and several studies are being conducted in order to contribute to the fight and prevention of this pandemic. This work presents a scholarly production dataset focused on COVID-19, providing an overview of scientific research activities, making it possible to identify countries, scientists and research groups most active in this task force to combat the coronavirus disease. The dataset is composed of 40,212 records of articles' metadata collected from Scopus, PubMed, arXiv and bioRxiv databases from January 2019 to July 2020. Those data were extracted by using the techniques of Python Web Scraping and preprocessed with Pandas Data Wrangling. In addition,

Ethics and Web Scraping

https://techcrunch.com/2017/04/28/someone-scraped-40000-tinder-selfies-to-make-a-facial-dataset-for-ai-experiments/



TC

Join Extra Crunch

Login

Search

TC Sessions: SaaS
Startups
Videos
Audio
Newsletters
Extra Crunch
EC-1s
Advertise
Events
More

Someone scraped 40,000 Tinder selfies to make a facial dataset for AI experiments

Natasha Lomas @riptari / 7:21 PM EDT • April 28, 2017

Comment

Tinder users have many motives for uploading their likeness to the dating app. But contributing a facial biometric to a downloadable data set for training convolutional neural networks probably wasn't top of their list when they signed up to swipe.

A user of Kaggle, a platform for machine learning and data science competitions which was recently acquired by Google, has uploaded a facial data set he says was created by exploiting Tinder's API to scrape 40,000 profile photos from Bay Area users of the dating app — 20,000 apiece from profiles of each gender.

The data set, called People of Tinder, consists of six downloadable zip files, with four containing around 10,000 profile photos each and two files with sample sets of around 500 images per gender.

Some users have had multiple photos scraped from their profiles, so there is likely a lot fewer

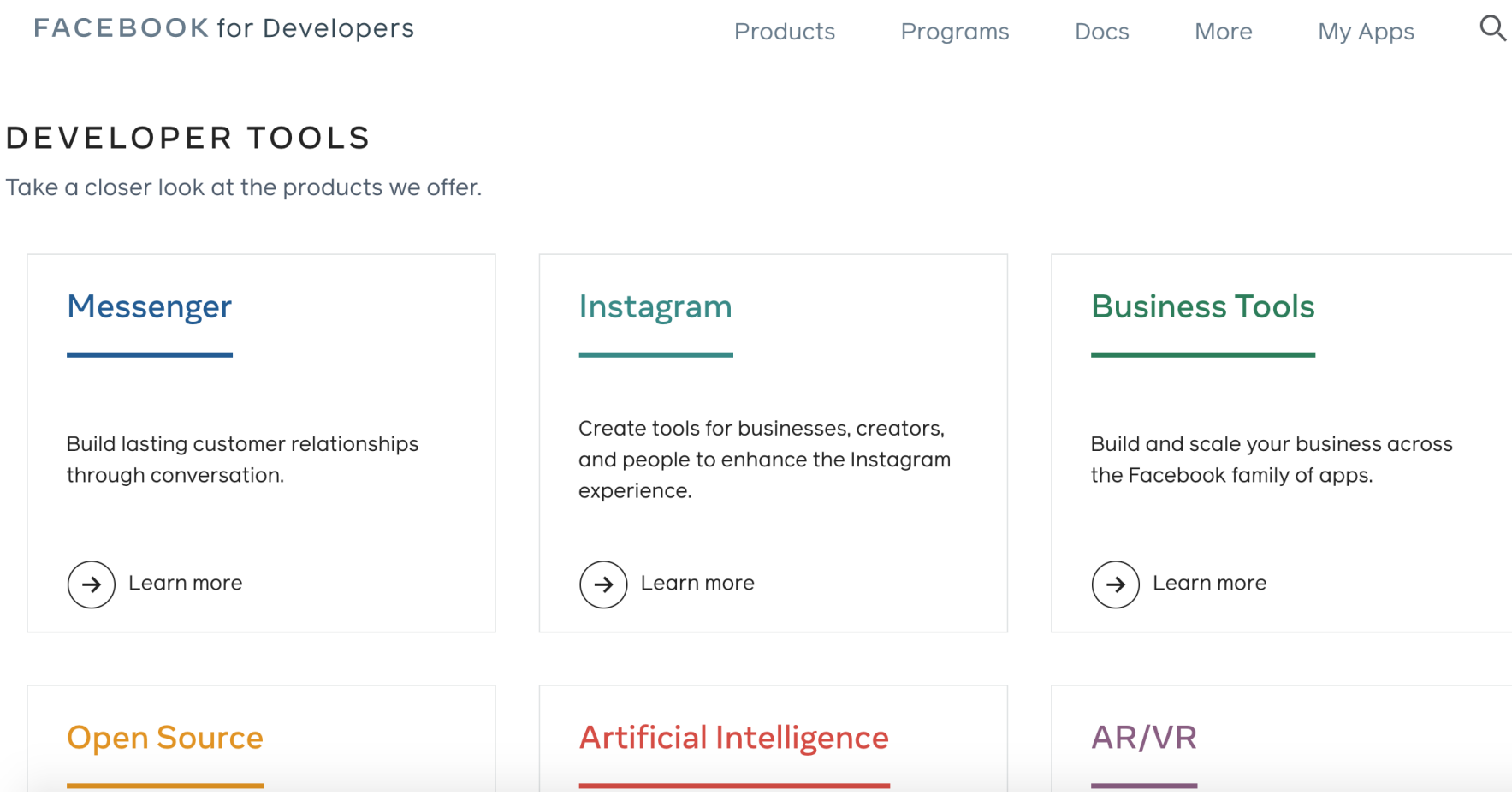
Ethics and Web Scraping

https://on.wsj.com/3hzeu9i

APIs

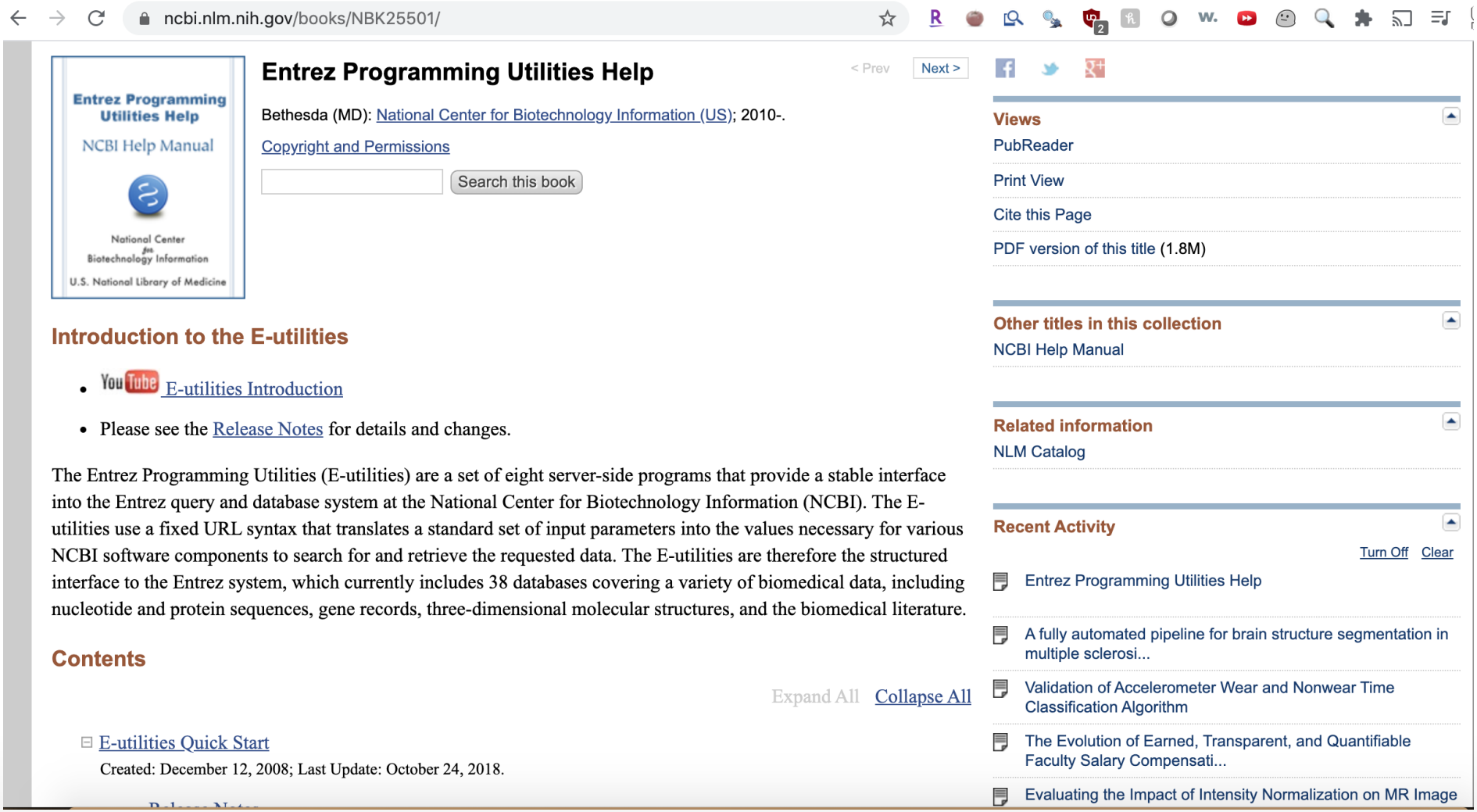
Application Programming Interfaces

https://developers.facebook.com/



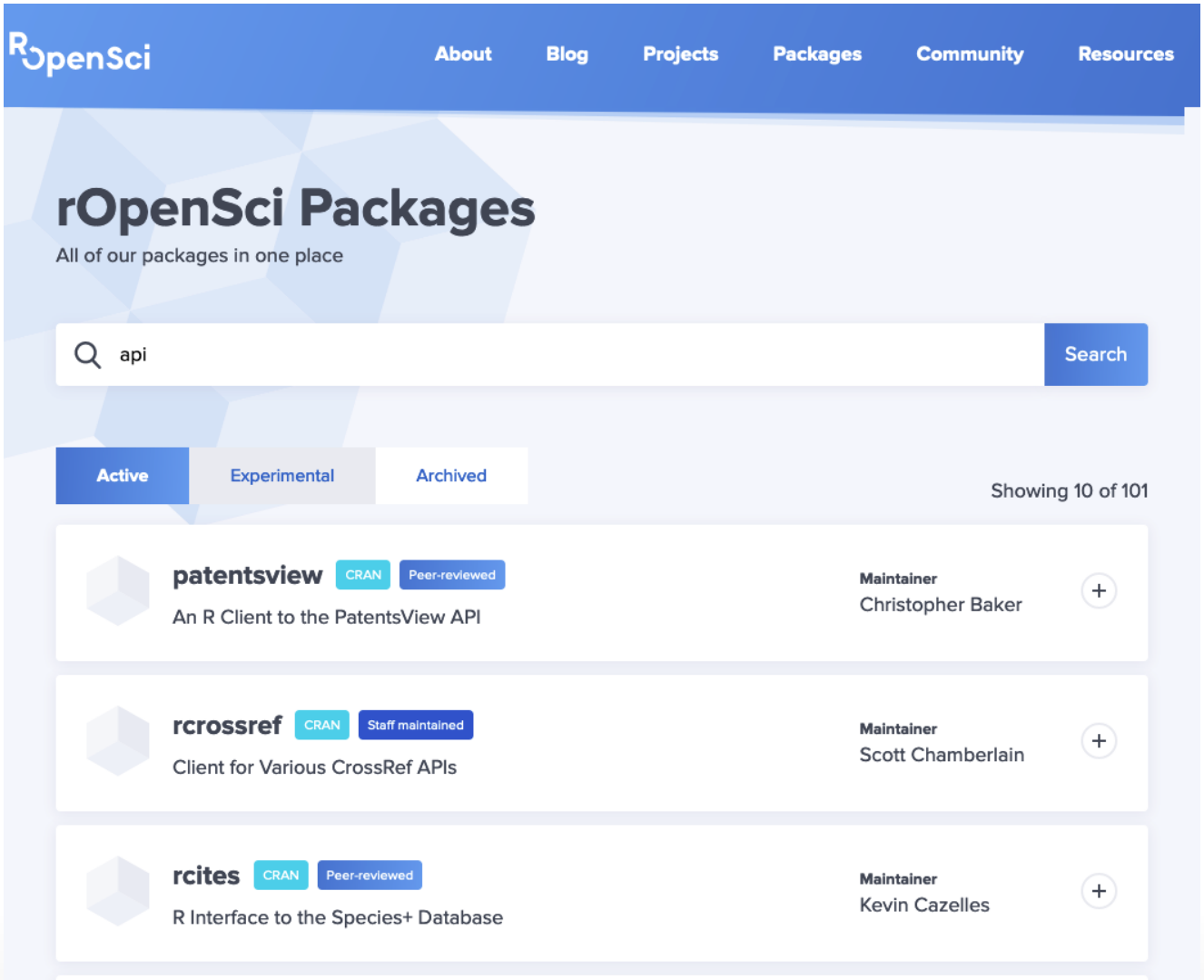
In biology too!

http://www.ncbi.nlm.nih.gov/books/NBK25501/
https://www.ncbi.nlm.nih.gov/home/develop/api/



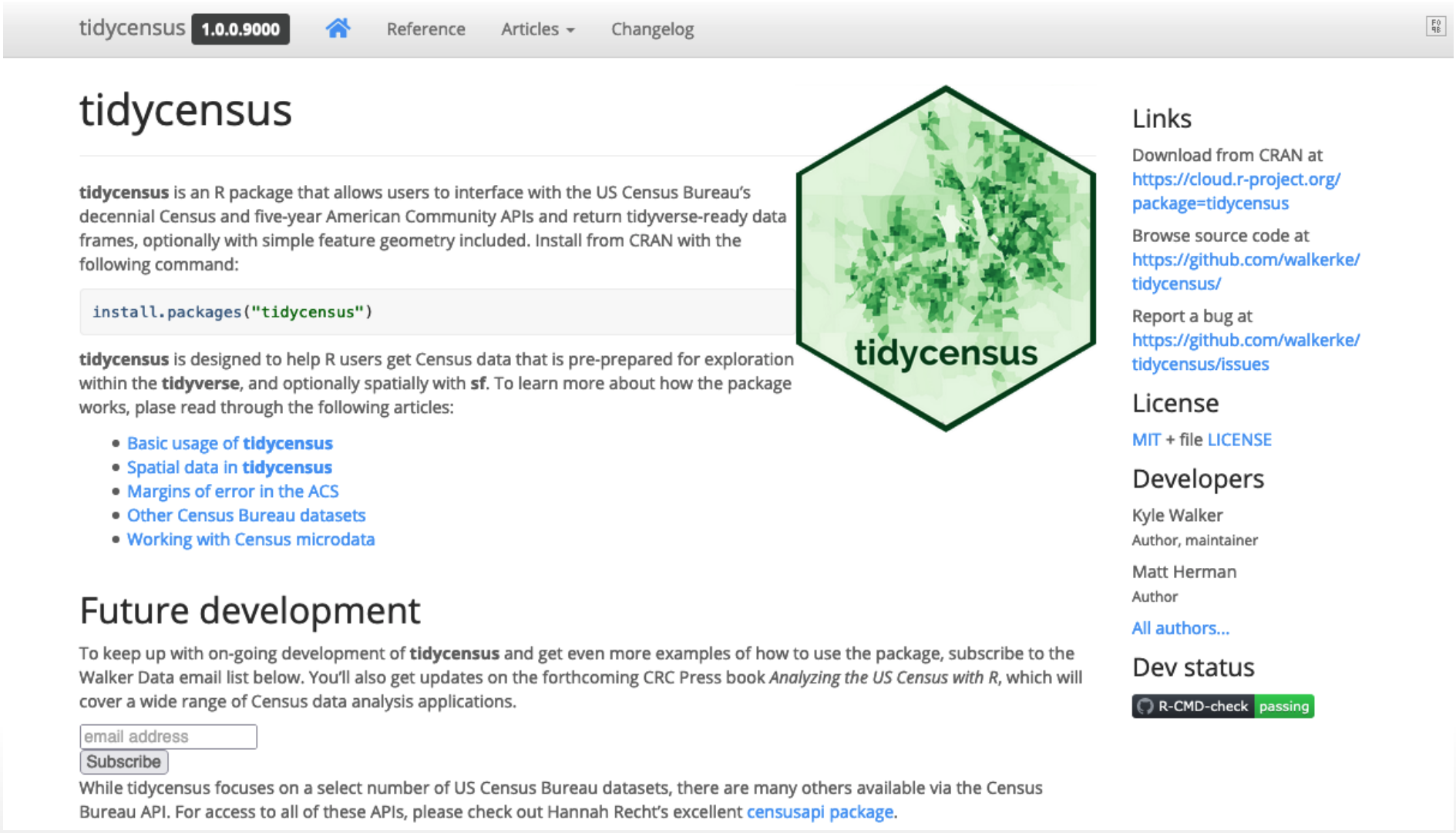
Step 0: Did someone do this already

https://ropensci.org/packages/



Step 0: Did someone do this already

`tidycensus` package: https://walker-data.com/tidycensus/articles/basic-usage.html



tidycensus 1.0.0.9000 Reference Articles Changelog

tidycensus

tidycensus is an R package that allows users to interface with the US Census Bureau's decennial Census and five-year American Community APIs and return tidyverse-ready data frames, optionally with simple feature geometry included. Install from CRAN with the following command:

```
install.packages("tidycensus")
```

tidycensus is designed to help R users get Census data that is pre-prepared for exploration within the **tidyverse**, and optionally spatially with **sf**. To learn more about how the package works, plase read through the following articles:

- Basic usage of **tidycensus**
- Spatial data in **tidycensus**
- Margins of error in the ACS
- Other Census Bureau datasets
- Working with Census microdata

Future development

To keep up with on-going development of **tidycensus** and get even more examples of how to use the package, subscribe to the Walker Data email list below. You'll also get updates on the forthcoming CRC Press book *Analyzing the US Census with R*, which will cover a wide range of Census data analysis applications.

email address
Subscribe

While tidycensus focuses on a select number of US Census Bureau datasets, there are many others available via the Census Bureau API. For access to all of these APIs, please check out Hannah Recht's excellent censusapi package.

Links

Download from CRAN at https://cloud.r-project.org/package=tidycensus

Browse source code at https://github.com/walkerke/tidycensus/

Report a bug at https://github.com/walkerke/tidycensus/issues

License

MIT + file LICENSE

Developers

Kyle Walker
Author, maintainer

Matt Herman
Author

All authors...

Dev status

R-CMD-check passing

Step 1: DIY

https://github.com/ThatCopy/catAPI/wiki/Usage

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

# Requests a random cat fact
query_url = "https://thatcopy.pw/catapi/rest/"

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

```
$id
[1] 40

$url
[1] "https://i.thatcopy.pw/cat/WxT0aQg.jpg"

$webpurl
[1] "https://i.thatcopy.pw/cat-webp/WxT0aQg.webp"

$x
[1] 63.21

$y
[1] 20.35
```

Not all APIs are "open"

https://www.rdocumentation.org/packages/twitteR/versions/1.1.9

```
# install.packages("twitteR")
library(twitteR)
# Supplied by Twitter
setup_twitter_oauth("API key", "API secret")

searchTwitter("crab cake", geocode="39.290692,-76.610221,5mi")
```

https://developer.twitter.com/en/portal/petition/academic/is-it-right-for-you

Not all APIs are "open"

https://walker-data.com/tidycensus/articles/basic-usage.html

```
# install.packages("tidycensus")
library(tidycensus)
# Supplied by census.gov
census_api_key("YOUR API KEY GOES HERE")

get_decennial(geography = "state",
                        variables = "P013001", # code for median age
                        year = 2010)
```