

# Advanced Data IO

Data Wrangling in R

# Google Sheets



gapminder



File Edit View Insert Format Data Tools Add-ons Help

Share



100% View only

A1 country

	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	12217.22686



Africa

Americas

Asia

Europe

Oceania



Explore

[https://docs.google.com/spreadsheets/d/1U6Cf\\_qEOhiR9AZqTqS3mbMF3zt2db48ZP5v3rkrAEJY/edit#gid=78](https://docs.google.com/spreadsheets/d/1U6Cf_qEOhiR9AZqTqS3mbMF3zt2db48ZP5v3rkrAEJY/edit#gid=78)

# 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 never to make these visible online!

```
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 `read_sheet()`

```
sheet_url <-  
  "https://docs.google.com/spreadsheets/d/1U6Cf_qE0hiR9AZqTqS3mbMF3zt2db48ZP5v3rkrAEJY/edit#gid=780868077"  
sheet_dat_1 <- read_sheet(sheet_url)
```

- ✓ Reading from "gapminder".
- ✓ Range 'Africa'.

```
head(sheet_dat_1)
```

```
# A tibble: 6 × 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_dat_oceania <- read_sheet(sheet_url, sheet = "Oceania")
```

- ✓ Reading from "gapminder".
- ✓ Range ' 'Oceania' '.

```
head(sheet_dat_oceania)
```

```
# A tibble: 6 × 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.
```

## Pull in a subset of data: rows

```
read_sheet(sheet_url, sheet = "Oceania", range = cell_rows(1:4))
```

- ✓ Reading from "gapminder".
- ✓ Range "'Oceania'!1:4'".

```
# A tibble: 3 × 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.
```

## Pull in a subset of data: columns

```
read_sheet(sheet_url, sheet = "Oceania", range = cell_cols("A:B"))
```

- ✓ Reading from "gapminder".
- ✓ Range ' 'Oceania' !A:B'.

```
# A tibble: 24 × 2
  country continent
  <chr>    <chr>
1 Australia Oceania
2 Australia Oceania
3 Australia Oceania
4 Australia Oceania
5 Australia Oceania
6 Australia Oceania
7 Australia Oceania
8 Australia Oceania
9 Australia Oceania
10 Australia Oceania
# i 14 more rows
```

# Reading data with the `googlesheets4` package

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 with `purrr`:

```
gapminder_sheets <- sheet_names(sheet_url)  
data_list <- map(gapminder_sheets, ~ read_sheet(sheet_url, sheet = .x))
```

- ✓ Reading from "gapminder".
- ✓ Range 'Africa'.
- ✓ Reading from "gapminder".
- ✓ Range 'Americas'.
- ✓ Reading from "gapminder".
- ✓ Range 'Asia'.
- ✓ Reading from "gapminder".
- ✓ Range 'Europe'.
- ✓ Reading from "gapminder".
- ✓ Range 'Oceania'.

# Reading data with the `googlesheets4` package

Check out the list:

```
str(data_list)
```

```
List of 5
$ : 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 ...
$ : 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 ...
$ : tibble [396 × 6] (S3:tbl_df/tbl/data.frame)
..$ country : chr [1:396] "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
..$ 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 ...
$ : tibble [360 × 6] (S3:tbl_df/tbl/data.frame)
..$ country : chr [1:360] "Albania" "Albania" "Albania" "Albania" ...
```

# Reading data with the googlesheets4 package

Pull out sheets as needed:

```
data_list[[{sheet}]]  
# OR  
data_list %>% pluck({sheet})  
# OR (if named)  
data_list${sheet}
```

The screenshot shows an RStudio code editor with the following code:

```
105  
106 data_list$ ← Tibbles stored in the list  
107 ...  
108  
109 ... ## Reading  
110  
111 Pull out  
112
```

A tooltip box with an orange border and rounded corners is positioned over the dollar sign (\$) in the code. The text "Tibbles stored in the list" is displayed inside the box, with an orange arrow pointing from the left side of the box towards the dollar sign. Below the tooltip, a dropdown menu is open, listing five categories: Africa, Americas, Asia, Europe, and Oceania, each preceded by a small colored square icon.

# Writing data with the `googlesheets4` package

```
sheet_dat_oceania <- data_list[[5]]  
  
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 rows to 'Oceania\_days'.

# JHU Tidyverse Book

<https://jhubdatascience.org/tidyversecourse/get-data.html#google-sheets>

# 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

The screenshot shows a web browser displaying the GitHub REST API documentation at [docs.github.com/en/rest/reference/search](https://docs.github.com/en/rest/reference/search). The left sidebar contains a list of API endpoints, and the main content area shows a "Default response" example with a status of 200 OK and a JSON payload.

```
{  
  "total_count": 7,  
  "incomplete_results": false,  
  "items": [  
    {  
      "name": "classes.js",  
      "path": "src/attributes/classes.js",  
      "sha": "d7212f9dee2dcc18f084d7df8f417b80846ded5a",  
      "url": "https://api.github.com/repos/167174/contents/src/attribut  
      "git_url": "https://api.github.com/repos/167174/git/blobs/d7212  
      "html_url": "https://github.com/jquery/jquery/blob/825ac3773694e0cd23e  
      "repository": {  
        "id": 167174,  
        "node_id": "MDEwOlJlcG9zaXRvcnkxNjcxNzQ=",  
        "name": "jquery",  
        "full_name": "jquery/jquery",  
        "owner": {  
          "login": "jquery",  
          "id": 70142,  
          "node_id": "MDQ6VXNlcjcwMTQy",  
          "avatar_url": "https://0.gravatar.com/avatar/6906f317a4733f4379b06  
          "gravatar_id": "",  
          "url": "https://api.github.com/users/jquery",  
          "html_url": "https://github.com/jquery",  
          "followers_url": "https://api.github.com/users/iauerv/followers".  
        }  
      }  
    }  
  ]  
}
```

<https://docs.github.com/en/rest/reference/search>

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

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

# Data frame structure from JSON

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

```
[1] 151 17
```

```
class(jsonData$pokemon)
```

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

```
jsonData$pokemon %>% filter(type == "Fire") %>% select(!(img))
```

		id	num	name	type	height	weight	candy	candy_count
1		4	004	Charmander	Fire	0.61 m	8.5 kg	Charmander	Candy
2		5	005	Charmeleon	Fire	1.09 m	19.0 kg	Charmander	Candy
3		37	037	Vulpix	Fire	0.61 m	9.9 kg	Vulpix	Candy
4		38	038	Ninetales	Fire	1.09 m	19.9 kg	Vulpix	Candy
5		58	058	Growlithe	Fire	0.71 m	19.0 kg	Growlithe	Candy
6		59	059	Arcanine	Fire	1.91 m	155.0 kg	Growlithe	Candy
7		77	077	Ponyta	Fire	0.99 m	30.0 kg	Ponyta	Candy
8		78	078	Rapidash	Fire	1.70 m	95.0 kg	Ponyta	Candy
9		126	126	Magmar	Fire	1.30 m	44.5 kg		None
10		136	136	Flareon	Fire	0.89 m	25.0 kg	Eevee	Candy

		egg	spawn_chance	avg_spawns	spawn_time	multipliers
1		2 km	0.2530	25.30	08:45	1.65
2	Not in	Eggs	0.0120	1.20	19:00	1.79
3		5 km	0.2200	22.00	13:43	2.74, 2.81
4	Not in	Eggs	0.0077	0.77	01:32	NULL
5		5 km	0.9200	92.00	03:57	2.31, 2.36
6	Not in	Eggs	0.0170	1.70	03:11	NULL

## Going deeper..

```
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"
```

```
[[7]]
```

```
[1] "Water"
```

```
[[8]]
```

```
[1] "Water"
```

## 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]]
```

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

# Lab

# Extra Slides: Web Scraping and APIs

# This is data

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

← → ⌂ ⓘ Not Secure | bowtie-bio.sourceforge.net/recount/ ☆ R 🔍 🍎 🔎 🛡️ 🗃

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

## » 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 <a href="#">here</a>	human	19	2,197,622,796	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	Illumina Human BodyMap 2.0 -- tissue comparison
cheung	<a href="#">20856902</a>	human	41	834,584,950	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	HapMap - CEU
core	<a href="#">19056941</a>	human	2	8,670,342	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	lung fibroblasts
gilad	<a href="#">20009012</a>	human	6	41,356,738	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	liver; males and females
maqc	<a href="#">20167110</a>	human	14 (technical)** 2 (biological)	71,970,164	<a href="#">original pooled</a>	<a href="#">original pooled</a>	<a href="#">original pooled</a>	experiment: MAQC-2
montgomery	<a href="#">20220756</a>	human	60	*886,468,054	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	HapMap - CEU
pickrell	<a href="#">20220758</a>	human	69	*886,468,054	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	HapMap - YRI
sultan	<a href="#">18599741</a>	human	4	6,573,643	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	cell type comparison
wang	<a href="#">18978772</a>	human	22	223,929,919	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	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	Type	Notes	
bodymap	not published, but publicly available <a href="#">here</a>	human	19	2,197,622,796	<a href="#">link</a>	<a href="#">Save As...</a> <a href="#">Print...</a> <a href="#">Cast...</a> <a href="#">Translate to English</a> <a href="#">Block element...</a>	Illumina Human BodyMap 2.0 - - tissue comparison	
cheung	<a href="#">20856902</a>	human	41	834,584,950	<a href="#">link</a>	<a href="#">View Page Source</a> <a href="#">View Frame Source</a> <a href="#">Reload Frame</a> <a href="#">Inspect</a>	HapMap - CEU	
core	<a href="#">19056941</a>	human	2	8,670,342	<a href="#">link</a>	<a href="#">Speech Services</a>	lung fibroblasts	
gilad	<a href="#">20009012</a>	human	6	41,356,738	<a href="#">link</a>	<a href="#">▶</a> <a href="#">▶</a>	liver; males and females	
maqc	<a href="#">20167110</a>	human	14 (technical)** 2 (biological)	71,970,164	<a href="#">original pooled</a>	<a href="#">original pooled</a>	<a href="#">original pooled</a>	experiment: MAQC-2
montgomery	<a href="#">20220756</a>	human	60	*886,468,054	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	HapMap - CEU
pickrell	<a href="#">20220758</a>	human	69	*886,468,054	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	HapMap - YRI
sultan	<a href="#">18599741</a>	human	4	6,573,643	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	cell type comparison
wang	<a href="#">18978772</a>	human	22	223,929,919	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	tissue comparison
katz.mouse	<a href="#">21057496</a>	mouse	4	14,368,471	<a href="#">link</a>	<a href="#">link</a>	<a href="#">link</a>	control vs. CUG-BP1

# What the computer sees

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <script src="sorttable.js" type="text/javascript"></script>
5 <title>ReCount: analysis-ready RNA-seq gene count datasets</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <link rel="stylesheet" type="text/css" href="css/style.css" media="screen" />
8 <script type="text/javascript">
9
10 var _gaq = _gaq || [];
11 _gaq.push(['_setAccount', 'UA-26478269-2']);
12 _gaq.push(['_trackPageview']);
13
14 (function() {
15   var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
16   ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
17   var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
18 })();
19
20 </script>
21
22 </head>
23
24 <body class="c20">
25 <div id="wrap">
26   <div id="top">
27     <div class="lefts">
28       <table width="100%" cellpadding="2">
29         <tr><td>
30           <a href=".index.shtml"><h1>ReCount</h1></a>
31           <h2>A multi-experiment resource of analysis-ready RNA-seq gene count datasets</h2>
32         </td><td align="right" valign="middle">
33           <h1><a href="http://www.biostat.jhsph.edu/"></a>&ampnbsp&ampnbsp</h1>
34         </td></tr>
35       </table>
36     </div>
37   </div>
38
39   <div id="subheader">
40     <p><b>There is now <a href="https://jhbiostatistics.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 href="https://jhbiostatistics.shinyapps.io/recount/">the documentation</a> for more information.</b></p>

```

# 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

Not Secure | bowtie-bio.sourceforge.net/recount/

Please note that to use the expressionsets below, you will need to install Bioculator and run the command library(Biocore).

» The Datasets

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

# Copy XPath

The screenshot shows a web browser window with the URL [Not Secure | bowtie-bio.sourceforge.net/recount/](http://bowtie-bio.sourceforge.net/recount/). A context menu is open over a table row with the ID `recounttab`. The table has columns for Study, PMID, Species, Number of biological replicates, Number of uniquely aligned reads, ExpressionSet, Count table, Phenotype table, and Notes. The rows contain data for bodymap, cheung, core, and gilad.

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 females

The context menu options include:

- Add attribute
- Edit as HTML
- Delete element
- Copy** (highlighted)
- Cut element
- Copy element
- Paste element
- Copy outerHTML
- Copy selector
- Copy JS path
- Copy styles
- Copy XPath** (highlighted)
- Copy full XPath
- Store as global variable
- Speech

The DevTools sidebar shows the element tree, and the bottom status bar indicates "Console What's New".

## 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)**	71,970,164	2 (biological)
	X6	X7	X8
1	ExpressionSet	Count table	Phenotype table
2	link	link	link
3	link	link	link
4	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	
4		lung fibroblasts	

# 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

### NEW YORK APARTMENT

FOR SALE  
**\$43,869,676,331**

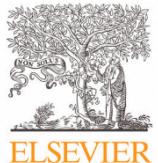
65,764 bedrooms  
55,588 bathrooms  
36,672,535 sq ft

Are you constantly frustrated with seeing the same low grade renovations and design choices in homes? Are you in love with your-wait? Are you looking for a charming well-groomed bedroom home in a sought-after area? Are you looking for a cozy, elegant home close to all the action in the city? Are you looking for a doll? Are you looking for a doll? Are you looking for a great investment home? Are you looking for a loft-like open space with both light and privacy? Are you looking for Beautiful Open Views? Are you looking for the House that is in your Price Range, is DATA-CHEED, has 3 Bedrooms, 4 Bathrooms, is located in the heart of the Princess Bay Section of Staten Island with Water Views & NOT IN A FLOOD ZONE surrounded by Millions Dollar Homes? Are you looking for the perfect balance of urban and suburban living? Are you looking for the perfect primary residence, pied à terre or investment property? Are you looking to do a Lease or Rent to Own Home?



# Ethics and Web Scraping

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



Contents lists available at ScienceDirect

Data in Brief

ELSEVIER

journal homepage: [www.elsevier.com/locate/dib](http://www.elsevier.com/locate/dib)



## Data Article

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



Breno Santana Santos<sup>a,b,\*</sup>, Ivanovitch Silva<sup>a</sup>,  
Marcel da Câmara Ribeiro-Dantas<sup>c</sup>, Gisliany Alves<sup>a</sup>,  
Patricia Takako Endo<sup>d</sup>, Luciana Lima<sup>a</sup>

<sup>a</sup> Universidade Federal do Rio Grande do Norte (UFRN), Rio Grande do Norte, Brazil

<sup>b</sup> Núcleo de Pesquisa e Prática em Inteligência Competitiva (NUPIC), Universidade Federal de Sergipe (UFS), Itabaiana, SE, Brazil

<sup>c</sup> Institut Curie (UMR168), Sorbonne Université (EDITE), Paris, France

<sup>d</sup> 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/>



[Join Extra Crunch](#)

[Login](#)

Search Q

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>

THE WALL STREET JOURNAL.

Subscribe | Sign In

English Edition | Print Edition | Video | Podcasts | [Latest Headlines](#)

Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate WSJ. Magazine Sports

SHARE

WORLD | ASIA | CHINA

## Alibaba Falls Victim to Chinese Web Crawler in Large Data Leak

Software developer scrapes 1.1 billion pieces of user data, including IDs and phone numbers, over eight months

Ant, Alibaba Show How China Rises in Big Tech Faster Than Other Countries

In less than six months, China's tech giant Ant went from planning a blockbuster IPO to restructuring in response to pressure from the central bank. As the U.S. also takes aim at big tech, here's how China is moving faster. Photo illustration: Sharon Shi

### MOST POPULAR NEWS

1. Rich Americans Borrow to Live Off Their Paper Wealth
2. Richard Branson's Virgin Galactic Flight Opens Door to Space Tourism
3. Biden Order Takes Aim at Tractor Repair
4. Why Aren't Millions of Unemployed Americans Finding Jobs?

# APIs

# Application Programming Interfaces

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

FACEBOOK for Developers

Products

Programs

Docs

More

My Apps



## DEVELOPER TOOLS

Take a closer look at the products we offer.

### Messenger

Build lasting customer relationships through conversation.



Learn more

### Instagram

Create tools for businesses, creators, and people to enhance the Instagram experience.



Learn more

### Business Tools

Build and scale your business across the Facebook family of apps.



Learn more

### Open Source

### Artificial Intelligence

### AR/VR

# In biology too!

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

<https://www.ncbi.nlm.nih.gov/home/develop/api/>

The screenshot shows a web browser displaying the "Entrez Programming Utilities Help" page. The URL in the address bar is [ncbi.nlm.nih.gov/books/NBK25501/](http://www.ncbi.nlm.nih.gov/books/NBK25501/). The page content includes a sidebar with the title "Entrez Programming Utilities Help", the "NCBI Help Manual" logo, and the "National Center for Biotechnology Information" logo. The main content area features a heading "Entrez Programming Utilities Help", a copyright notice, and a search bar. To the right, there are social sharing icons (Facebook, Twitter, Google+), a "Views" section with links to "PubReader", "Print View", "Cite this Page", and a "PDF version of this title (1.8M)". Below this is a section titled "Other titles in this collection" with a link to "NCBI Help Manual". Further down is a "Related information" section linking to "NLM Catalog". A "Recent Activity" sidebar on the right lists several recent updates, including "Entrez Programming Utilities Help", "A fully automated pipeline for brain structure segmentation in multiple sclerosis...", "Validation of Accelerometer Wear and Nonwear Time Classification Algorithm", "The Evolution of Earned, Transparent, and Quantifiable Faculty Salary Compensation...", and "Evaluating the Impact of Intensity Normalization on MR Image". At the bottom, there are "Contents" links for "E-utilities Quick Start" and "E-utilities API", along with a note about the last update.

# Step 0: Did someone do this already

<https://ropensci.org/packages/>

The screenshot shows the rOpenSci Packages page. At the top, there is a navigation bar with links for About, Blog, Projects, Packages, Community, and Resources. Below the navigation bar, the title "rOpenSci Packages" is displayed, followed by the subtitle "All of our packages in one place". A search bar contains the query "api", and a blue "Search" button is to its right. Below the search bar, there are three filter buttons: "Active" (which is selected), "Experimental", and "Archived". To the right of these filters, it says "Showing 10 of 101". The main content area displays three package cards. Each card includes the package name, CRAN status, maintainer information, and a plus sign icon for adding the package. The first package is "patentsview" (CRAN, Peer-reviewed), maintained by Christopher Baker. The second is "rcrossref" (CRAN, Staff maintained), maintained by Scott Chamberlain. The third is "rcites" (CRAN, Peer-reviewed), maintained by Kevin Cazelles.

Package	Status	Maintainer
patentsview	CRAN Peer-reviewed	Christopher Baker
rcrossref	CRAN Staff maintained	Scott Chamberlain
rcites	CRAN Peer-reviewed	Kevin Cazelles

# Step 0: Did someone do this already

**tidycensus** package: <https://walker-data.com/tidycensus/articles/basic-usage.html>

The screenshot shows the homepage of the **tidycensus** website. At the top, there's a navigation bar with links for "tidycensus 1.0.0.9000", "Reference", "Articles ▾", "Changelog", and a small "F0 4B" icon. Below the navigation, the title "tidycensus" is displayed in a large, bold font. A green hexagonal logo featuring a map of the United States and the word "tidycensus" is centered on the page. To the right of the logo, there's a "Links" section with links to CRAN, GitHub, and issues. Below the logo, there's a "Basic usage" section with a code snippet for installing the package, a "Future development" section with a newsletter sign-up form, and a note about other Census Bureau datasets.

**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, please 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.

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://catfact.ninja/fact"

req <- GET(query_url)
content(req)
```

```
$fact
[1] "Phoenician cargo ships are thought to have brought the first domesticated"
```

```
$length
[1] 105
```

## 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)
```