APIs and Web Scraping Lab

David Gerard

2020-09-23

Learning Objectives

- Obtain data from an API.
- Scrape data from the web.
- Intense data cleaning.

Semantic Scholar

Semantic Scholar is like Google Scholar, but they claim to return more useful connections between journal articles and authors. Recently, I've found myself using both, whereas before I spent most of my time searching for articles on Google Scholar.

Semantic Scholar has a really nice API to practice on: https://api.semanticscholar.org/

I actually used this API in real-life when I was preparing my reappointment files to check out and plot citation counts. Let's try and reproduce what I did.

Just like Google Scholar, each author as their own page. Here is mine: https://www.semanticscholar.org/author/David-Gerard/145899953

1. Consider the following vector of DOIs of my publications:

```
gerard_dois <- c("10.1186/s12859-020-3450-9",
                 "10.1093/bioinformatics/btz852",
                 "10.1080/07391102.2019.1679666",
                 "10.1093/biostatistics/kxy029",
                 "10.1534/genetics.118.301468",
                 "10.1214/17-EJS1330",
                 "10.5705/ss.202018.0345",
                 "10.1016/j.laa.2016.04.033",
                 "10.1016/j.jmva.2015.01.020",
                 "10.1007/s11084-013-9331-8",
                 "10.1186/1471-2148-11-291")
```

   Use the API to download the paper information for each of my publications listed above. Save the content of this information as a list. So, for example, each element of this list is itself a list with the following elements:

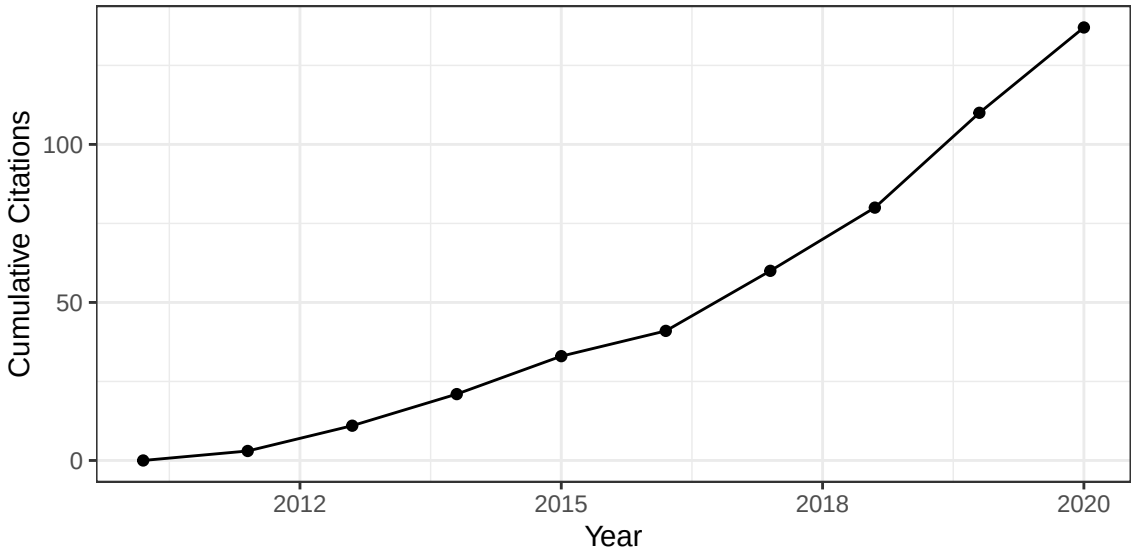
```
##  [1] "abstract"                "arxivId"
##  [3] "authors"                 "citationVelocity"
##  [5] "citations"               "corpusId"
##  [7] "doi"                     "fieldsOfStudy"
##  [9] "influentialCitationCount" "is_open_access"
## [11] "is_publisher_licensed"   "paperId"
## [13] "references"              "title"
```

```
## [15] "topics"                    "url"
## [17] "venue"                     "year"
```

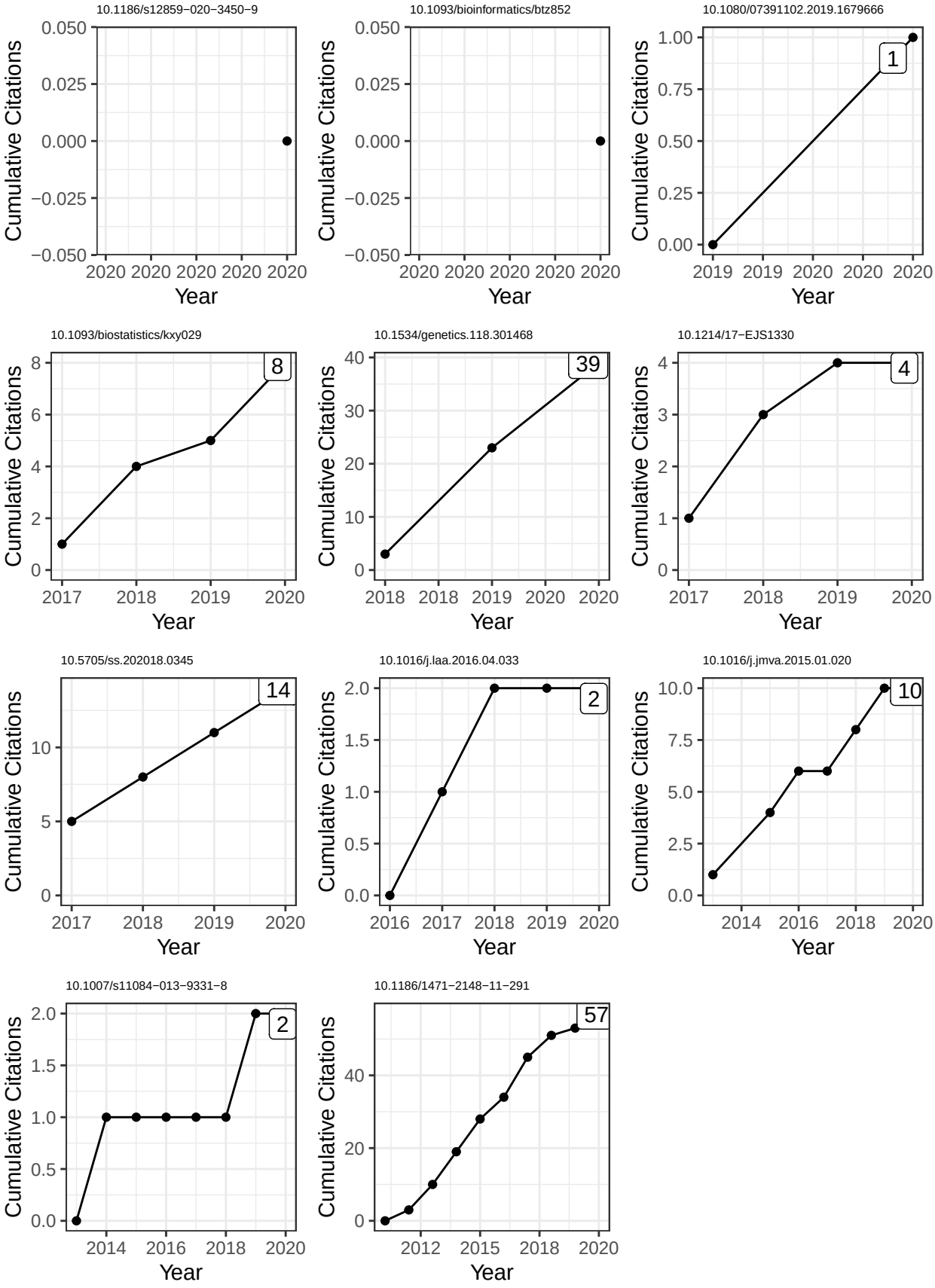
2. Tidy these data. Extract the DOI's, title, and authors from each paper. Also, for each year, calculate the number of citations each paper received. So your final data frame should look like this:

```
## Rows: 49
## Columns: 5
## $ year          <dbl> 2020, 2020, 2019, 2020, 2018, 2019, 2020, 2017, 2018,...
## $ count         <dbl> 0, 0, 0, 1, 3, 1, 3, 1, 3, 20, 16, 1, 2, 1, 0, 5, 3, ...
## $ doi           <chr> "10.1186/s12859-020-3450-9", "10.1093/bioinformatics/...
## $ title         <chr> "Data-based RNA-seq simulations by binomial thinning"...
## $ year_released <int> 2020, 2020, 2019, 2019, 2018, 2018, 2018, 2018, 2018,...
```

3. Plot the cumulative number of citations over time, aggregating over all papers. Your plot should look like this:



4. For each paper, make a plot of cumulative summations (use a for-loop). Your plots should look like this:


10.1186/s12859−020−3450−9
10.1093/bioinformatics/btz852
10.1080/07391102.2019.1679666
10.1093/biostatistics/kxy029
10.1534/genetics.118.301468
10.1214/17−EJS1330
10.5705/ss.202018.0345
10.1016/j.laa.2016.04.033
10.1016/j.jmva.2015.01.020
10.1007/s11084−013−9331−8
10.1186/1471−2148−11−291
Cumulative Citations
Year
1
8
39
4
14
2
10
2
57

Film Remakes

Consider the list of film remakes from Wikipedia: https://en.wikipedia.org/wiki/List_of_film_remakes_(A-M) and https://en.wikipedia.org/wiki/List_of_film_remakes_(N-Z)

1. Download the html file and save it as a variable. You can also load the "remakes_1.html" and "remakes_2.html" files in the data folder.
2. Extract the "table.wikitable" elements from both files.
3. Now obtain a single list of all of the table elements.
4. Create a single data frame with two columns — `Remakes` and `Original version`.
5. Create a data frame that contains the year of the remake, the year of the original, and the name of the original. Note that there are many films with multiple remakes.

   Your final data frame should look like this:

```
## # A tibble: 973 x 3
##    year_rm name_ov                          year_ov
##      <dbl> <chr>                              <dbl>
##  1    2010 13 Tzameti                          2005
##  2    1951 Le Corbeau                          1943
##  3    1996 One Hundred and One Dalmatians      1961
##  4    2005 Two Thousand Maniacs!               1964
##  5    1948 The Three Godfathers                1916
##  6    1936 The Three Godfathers                1916
##  7    1930 The Three Godfathers                1916
##  8    1921 The Three Godfathers                1916
##  9    1919 The Three Godfathers                1916
## 10    2017 3 Idiots                            2009
## # ... with 963 more rows
```

6. There are two films remade in the same year as the original version. What were they?
7. Find the 5 movies that were remade the most number of times. You should get:

```
## # A tibble: 5 x 2
##   name_ov                n
##   <chr>              <int>
## 1 Oliver Twist           9
## 2 Jane Eyre              8
## 3 Munna Bhai M.B.B.S.    8
## 4 Robin Hood             8
## 5 Treasure Island        8
```

8. Plot a step function for these movies by year. Your final plot should look like this: