

MY472 – Data for Data Scientists

Week 8: APIs

Friedrich Geiecke

15 November 2022

<https://lse-my472.github.io/>

Course outline

1. Introduction
2. Tabular data
3. Data visualisation
4. Textual data
5. HTML, CSS, and scraping static websites
6. (Reading week)
7. XML, RSS, and scraping non-static website
8. Working with web APIs
9. Creating and managing databases
10. Interacting with online databases
11. Cloud computing

Plan for today

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

Outline

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

JSON

- ▶ API responses are very often in JSON format (JavaScript Object Notation)
- ▶ JSON is a lightweight and flexible format to store and transmit data
- ▶ JSON data can e.g. be read/parsed into R with the 'fromJSON' function from the `jsonlite` package
- ▶ Yet, many packages have their own functions to read data in JSON format into R, e.g. the 'content(r, ...)' function from the `httr` package which we will use for querying APIs

JSON

- ▶ JSON objects are key-value pairs: "someKey": 42
- ▶ Keys have to be strings with double quotes
- ▶ Values can be one of the following types:
 - ▶ JSON object ({})
 - ▶ String ("hello")
 - ▶ Number (42, 3.141)
 - ▶ Array ([])
 - ▶ Boolean (true, false)
 - ▶ null
- ▶ Nested structure

Reference:

https://www.w3schools.com/js/js_json_syntax.asp

Example (1/3)

```
{
  "name": "Bob",
  "courseWork": [
    "Assignment",
    "Final"
  ],
  "grades": [
    65,
    73
  ],
  "supervisor": {
    "name": "Alice",
    "department": "Mathematics"
  },
  "currentlyEnrolled": false
}
```

Example (2/3)

```
{
  "date": [
    "2020-10-01",
    "2020-10-17",
    "2020-10-24"
  ],
  "section": [
    "Economics",
    "Politics",
    "Sports"
  ],
  "headline": [
    "Covid recession",
    "New polls",
    "Liverpool wins"
  ],
  "lead_paragraph": [
    "The recession triggered by the pandemic ...",
    null,
    "In their game on Saturday, Liverpool FC ..."
  ]
}
```


Example (3/3)

```
{
  "MT": [
    {
      "code": "MY472",
      "title": "Data for Data Scientists",
      "description": "A course about collecting, processing, and storing data.",
      "units": 0.5,
      "offeredThisYear": true,
      "registeredStudents": []
    },
    {
      "code": "MY470",
      "title": "Computer Programming",
      "description": "An introduction to programming.",
      "units": 0.5,
      "offeredThisYear": true,
      "registeredStudents": []
    }
  ],
  "LT": [
    {
      "code": "MY459",
      "title": "Special Topics in Quantitative Analysis: Quantitative Text Analysis",
      "description": "A course about text analysis.",
      "units": 0.5,
      "offeredThisYear": true,
      "registeredStudents": []
    }
  ]
}
```

Outline

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

APIs

- ▶ API: Application Programming Interface
- ▶ In web APIs, a set of structured HTTP requests can return data in a lightweight format e.g. JSON or XML
- ▶ The API user sends a request to the API (e.g. with a software such as R) and the API returns data from the API provider's database
- ▶ APIs are widely used to communicate between applications

See also e.g. Munzert et al., 2014, Chapter 9

APIs

Types of APIs

1. **RESTful APIs**: Queries for static information at current moment (e.g. user profiles, posts, etc.)
2. **Streaming APIs**: Changes in users' data in real time (e.g. new tweets, weather alerts...)

APIs generally have extensive **documentation**

- ▶ Written for developers
- ▶ What to look for: **Endpoints** and **parameters**

Most APIs are **rate-limited**

- ▶ Restrictions on number of API calls by user/IP address and period of time
- ▶ Commercial APIs may impose a monthly fee

An example: Google Maps API

Constructing an API call

- ▶ Baseline URL **endpoint**:

`https://maps.googleapis.com/maps/api/geocode/json`

- ▶ Parameters: `?address=london`

- ▶ Authentication token: `&key=XXXXX`

From R, use `httr` package to make GET request:

```
library(httr)
r <- GET(
  "https://maps.googleapis.com/maps/api/geocode/json",
  query=list(address="london", key="XXXXX"))
```

If request was successful, returned code will be 200, where 4xx indicates client errors and 5xx indicates server errors.

If you need to attach data, use POST request.

```

{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "London",
          "short_name" : "London",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "London",
          "short_name" : "London",
          "types" : [ "postal_town" ]
        }
      ],
      "formatted_address" : "London, UK",
      "geometry" : {
        "bounds" : {
          "northeast" : {
            "lat" : 51.6723432,
            "lng" : 0.148271
          },
          "southwest" : {
            "lat" : 51.384940099999999,
            "lng" : -0.3514683
          }
        },
        "location" : {
          "lat" : 51.5073509,
          "lng" : -0.1277583
        }
      },
      ...
    }
  ]
}

```

```
{
...
    "location_type" : "APPROXIMATE",
    "viewport" : {
        "northeast" : {
            "lat" : 51.6723432,
            "lng" : 0.148271
        },
        "southwest" : {
            "lat" : 51.384940099999999,
            "lng" : -0.3514683
        }
    },
    "place_id" : "ChIJdd4hrwug2EcRmSrV3Vo6llI",
    "types" : [ "locality", "political" ]
},
"status" : "OK"
}
```

Authentication

- ▶ Many APIs require an access keys or tokens
- ▶ This is also the case for both examples in this lecture, the New York Times and Twitter APIs

R packages

Before starting a new project, it is worth checking whether there is already an R package specifically for this API, or whether to use e.g. `httr`. Where to look?

- ▶ **CRAN Web Technologies Task View** (yet, only packages released in CRAN)
- ▶ GitHub (including unreleased packages and most recent versions of packages)
- ▶ **rOpenSci Consortium**

Also see this **great list of APIs** in case you need inspiration

Why APIs?

Advantages

- ▶ Cleaner data collection: Avoid malformed HTML, no legal issues, clear data structures, more trust in data collection...
- ▶ Standardised data access procedures: Transparency, replicability
- ▶ Robustness: Benefits from “wisdom of the crowds”

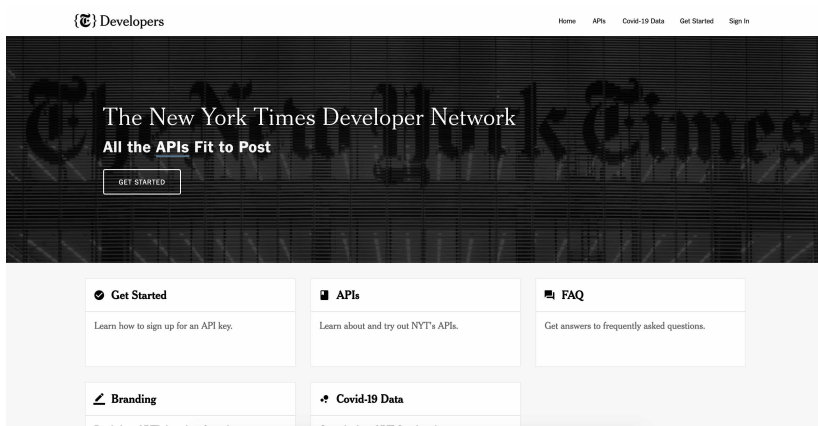
Disadvantages

- ▶ Not always available
- ▶ Dependency on API providers
- ▶ Rate limits

Outline

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

New York Times API



Website: <https://developer.nytimes.com/>

New York Times API

- ▶ The New York Times (NYT) offers a range of APIs
- ▶ We will use:
 - ▶ The Article Search API to search for keywords in articles
 - ▶ The Archive API to download the full data for a given month
- ▶ While we cannot download full articles in the public version of the Archive API, we can obtain headlines, abstracts, snippets, and/or lead paragraphs since 1851

Access to the New York Times APIs

- ▶ To obtain a key, follow the instructions here:
<https://developer.nytimes.com/get-started>
- ▶ When specifying access rights of your key, make sure to tick the boxes for Article Search and Archive API
- ▶ If you are interested in the topic, check out other APIs such as e.g. the RSS Feed API (not covered here)

Outline

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

Access to the Twitter API

- ▶ Application for a developer account can be made via <https://developer.twitter.com>
- ▶ Afterwards, apply for Elevated Access (we work with the package 'rtweet' which in its current version uses an endpoint that requires this access)
- ▶ When your application is approved, you can create an app within a project or a standalone app
- ▶ Create a bearer token and store it locally on your computer (do not upload it to GitHub or share it etc.)
- ▶ We will use this token with 'rtweet' later

Addendum: Academic Research and the Twitter API

- ▶ Recently, an “Academic Research product track” for the Twitter API was introduced
- ▶ Among other features, it can be used to access significant amounts of historical tweets for free if the application is approved
- ▶ Applications can be made via <https://developer.twitter.com/en/products/twitter-api/academic-research>
- ▶ Non-commercial use only and requires a clearly defined research objective
- ▶ There also exists an R package specifically for this type of API **academictwitterR**
- ▶ To start in R, install package with `install.packages("academictwitterR")` and run `vignette("academictwitterR-auth")`

Anatomy of a tweet



Barack Obama ✓

@BarackObama



Follow

Four more years.



RETWEETS

756,411

FAVORITES

288,867



11:16 PM - 6 Nov 2012

Anatomy of a tweet

Tweets are stored in JSON format:

```
{ "created_at": "Wed Nov 07 04:16:18 +0000 2012",  
  "id": 266031293945503744,  
  "text": "Four more years. http://t.co/bAJE6Vom",  
  "source": "web",  
  "user": {  
    "id": 813286,  
    "name": "Barack Obama",  
    "screen_name": "BarackObama",  
    "location": "Washington, DC",  
    "description": "This account is run by Organizing for Action staff.  
    Tweets from the President are signed -bo.",  
    "url": "http://t.co/8aJ56Jcemr",  
    "protected": false,  
    "followers_count": 54873124,  
    "friends_count": 654580,  
    "listed_count": 202495,  
    "created_at": "Mon Mar 05 22:08:25 +0000 2007",  
    "time_zone": "Eastern Time (US & Canada)",  
    "statuses_count": 10687,  
    "lang": "en" },  
  "coordinates": null,  
  "retweet_count": 756411,  
  "favorite_count": 288867,  
  "lang": "en"  
}
```

Twitter APIs

Two different methods to collect Twitter data

1. REST API

- ▶ Queries for specific information about users and tweets
- ▶ Search recent tweets
- ▶ Examples: User profile, list of followers and friends, tweets generated by a given user (“timeline”), users lists, etc.

2. Streaming API

- ▶ Connect to the “stream” of tweets as they are being published
- ▶ Three streaming APIs:
 - 2.1 Sample stream: 1% random sample of tweets
 - 2.2 Filter stream: tweets filtered by keywords (when volume reaches 1% of all tweets, it will also return a random sample)
 - 2.3 Geo stream: tweets filtered by location

Limitation: With usual access, larger amounts of tweets can just be downloaded in real time. Historical data can only be obtained from the last 7 days or otherwise from user timelines up to $\sim 3,200$ most recent tweets. Important exception: Academic research access

Can the sampling be biased?

[Morstatter](#) et al, 2013, *ICWSM*, “Is the Sample Good Enough? Comparing Data from Twitter’s Streaming API with Twitter’s Firehose”:

- ▶ 1% random sample from Streaming API is not truly random
- ▶ Less popular hashtags, users, topics... less likely to be sampled
- ▶ But for keyword-based samples, bias is not as important

[González-Bailón](#) et al, 2014, *Social Networks*, “Assessing the bias in samples of large online networks”:

- ▶ Small samples collected by filtering with a subset of relevant hashtags can be biased
- ▶ Central, most active users are more likely to be sampled
- ▶ Data collected via search (REST) API more biased than those collected with Streaming API

More biases in social media data

SOCIAL SCIENCES

Social media for large studies of behavior

Large-scale studies of human behavior in social media need to be held to higher methodological standards

By Derek Ruths^{1*} and Jürgen Pfeffer²

On 3 November 1948, the day after Harry Truman won the United States presidential elections, the *Chicago Tribune* published one of the most famous erroneous headlines in newspaper history: “Dewey Defeats Truman” (1, 2). The headline was informed by telephone surveys, which had inadver-

different social media platforms (8). For instance, Instagram is “especially appealing to adults aged 18 to 29, African-American, Latinos, women, urban residents” (9) whereas Pinterest is dominated by females, aged 25 to 34, with an average annual household income of \$100,000 (10). These sampling biases are rarely corrected for (if even acknowledged).

Proprietary algorithms for public data. Platform-specific sampling problems, for example, the highest-volume source of pub-

The rise of “embedded research” with providers that give them access to platform-specific data, algorithms, and resources) is creating a divided social media research community. Such efforts, for example, can see a platform’s internal workings and make accommodations, but may not be able to reveal their color or the data used to generate their findings.

Ruths and Pfeffer, 2015, “Social media for large studies of behavior”, *Science*

More biases in social media data

Sources of bias (Ruths and Pfeffer, 2015; Lazer et al, 2017)

- ▶ **Population bias**
 - ▶ Sociodemographic characteristics are correlated with presence on social media
- ▶ **Self-selection within samples**
 - ▶ Partisans more likely to post about politics (Barberá & Rivero, 2014)
- ▶ **Proprietary algorithms for public data**
 - ▶ Twitter API does not always return 100% of publicly available tweets (Morstatter et al, 2014)
- ▶ **Human behavior and online platform design**
 - ▶ e.g. *Google Flu* (Lazer et al, 2014)

More biases in social media data

Reducing biases and flaws in social media data

DATA COLLECTION

- 1. Quantifies platform-specific biases (platform design, user base, platform-specific behavior, platform storage policies)
- 2. Quantifies biases of available data (access constraints, platform-side filtering)
- 3. Quantifies proxy population biases/mismatches

METHODS

- 4. Applies filters/corrects for nonhuman accounts in data
- 5. Accounts for platform and proxy population biases
 - a. Corrects for platform-specific and proxy population biases
 - OR
 - b. Tests robustness of findings
- 6. Accounts for platform-specific algorithms
 - a. Shows results for more than one platform
 - OR
 - b. Shows results for time-separated data sets from the same platform
- 7. For new methods: compares results to existing methods on the same data
- 8. For new social phenomena or methods or classifiers: reports performance on two or more distinct data sets (one of which was not used during classifier development or design)

Issues in evaluating data from social media. Large-scale social media studies of human behavior should i address issues listed and discussed herein (further discussion in supplementary materials).

Ruths and Pfeffer, 2015, “Social media for large studies of behavior”,
Science

Outline

- ▶ JSON
- ▶ APIs
- ▶ API Examples
 - ▶ New York Times
 - ▶ Twitter
- ▶ Guided coding session

Guided coding session

01-json-in-r.Rmd

02-nytimes-api.Rmd

03-twitter-rest-api.Rmd