# Collecting digital trace data

Shuning Lu (Ph.D.)
Assistant Professor
Department of Communication
North Dakota State University

Email: shuning.lu@ndsu.edu
Twitter: @shuning_lu

# Today's agenda

**Morning:**

- Logistics  9:30-10:00
- Overview of digital trace data 10:00-10:30
- Data collection tutorial 10:40-11:30
- Research speed-dating 11:30-noon

**Noon:**

- Lunch and guest speaker (Alvin Zhou)

**Afternoon:**

- Group activity & Report 13:30-16:00

# What is digital trace data?

Social media

Geo-spatial data

News websites

Digital archives

Discussion: other kinds of data?

# Strengths and weaknesses of digital trace data

Strengths: Always-on, unobtrusive, capture relationships

Weaknesses: proprietary, non-representativeness, drift, algorithmic confounding, unstructured, other bias

How to leverage the strengths?

How to alleviate the weakness?

# Ways of collecting digital trace data

Downloading existing dataset

Screen/web scraping

API

# Downloading existing dataset

https://datasetsearch.research.google.com

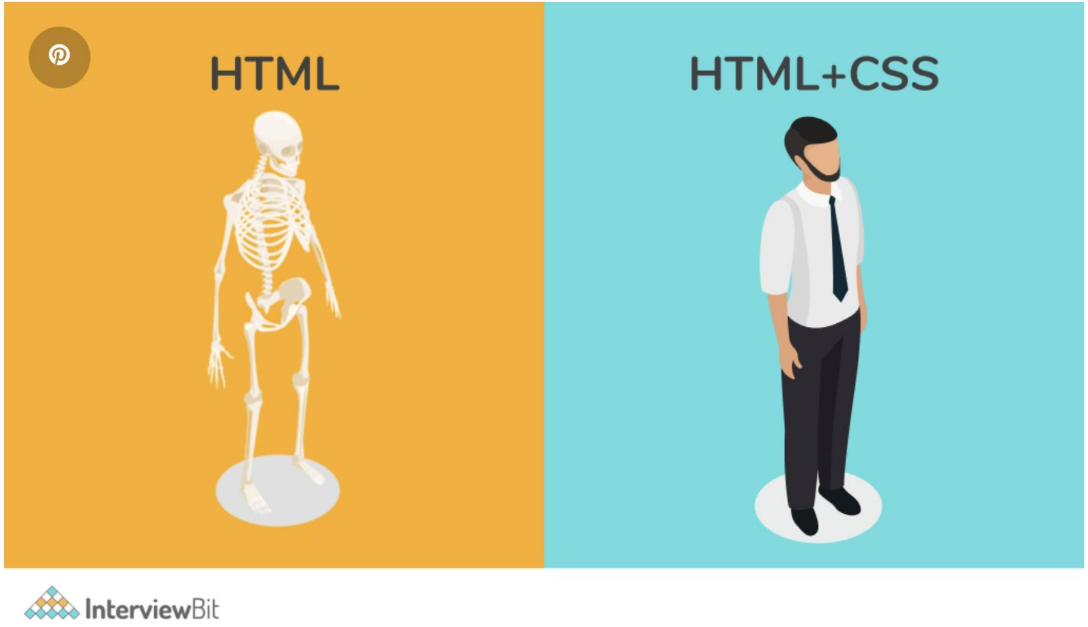
Macro-level data: World Bank, OECD, government, etc.

Traditional micro-level data: GSS, Pew Research, etc.

Public accessible platform data: Ad data, legal request data, special topics, etc.

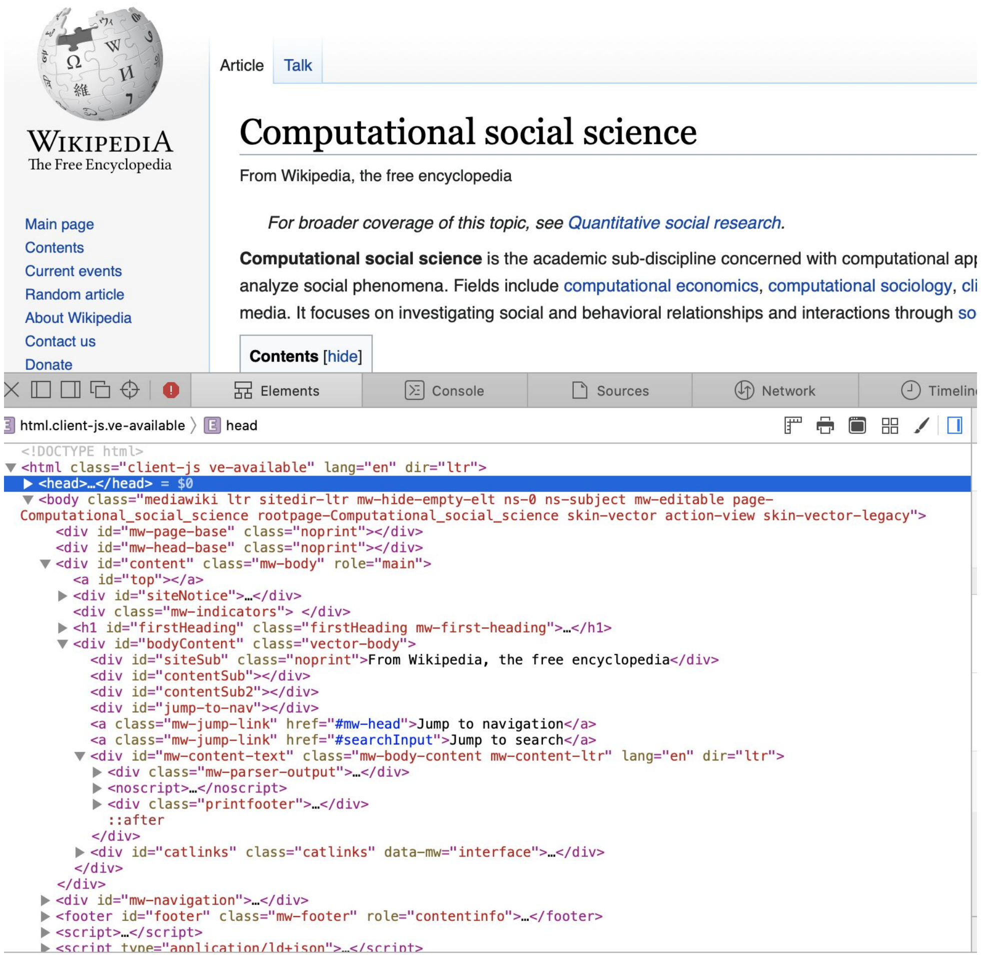
Research data: GitHub, Open Science Framework, Harvard Dataverse, ICPSR, etc.

# Screen/web scraping



# Developer tool

Inspect html/css codes



Computational social science

From Wikipedia, the free encyclopedia

*For broader coverage of this topic, see Quantitative social research.*

**Computational social science** is the academic sub-discipline concerned with computational ap
analyze social phenomena. Fields include computational economics, computational sociology, cli
media. It focuses on investigating social and behavioral relationships and interactions through so

```
<!DOCTYPE html>
▼<html class="client-js ve-available" lang="en" dir="ltr">
  ▶<head>…</head> = $0
  ▼<body class="mediawiki ltr sitedir-ltr mw-hide-empty-elt ns-0 ns-subject mw-editable page-
Computational_social_science rootpage-Computational_social_science skin-vector action-view skin-vector-legacy">
      <div id="mw-page-base" class="noprint"></div>
      <div id="mw-head-base" class="noprint"></div>
    ▼<div id="content" class="mw-body" role="main">
        <a id="top"></a>
      ▶<div id="siteNotice">…</div>
        <div class="mw-indicators"> </div>
      ▶<h1 id="firstHeading" class="firstHeading mw-first-heading">…</h1>
      ▼<div id="bodyContent" class="vector-body">
          <div id="siteSub" class="noprint">From Wikipedia, the free encyclopedia</div>
          <div id="contentSub"></div>
          <div id="contentSub2"></div>
          <div id="jump-to-nav"></div>
          <a class="mw-jump-link" href="#mw-head">Jump to navigation</a>
          <a class="mw-jump-link" href="#searchInput">Jump to search</a>
        ▼<div id="mw-content-text" class="mw-body-content mw-content-ltr" lang="en" dir="ltr">
          ▶<div class="mw-parser-output">…</div>
          ▶<noscript>…</noscript>
          ▶<div class="printfooter">…</div>
            ::after
          </div>
        ▶<div id="catlinks" class="catlinks" data-mw="interface">…</div>
        </div>
      </div>
    ▶<div id="mw-navigation">…</div>
    ▶<footer id="footer" class="mw-footer" role="contentinfo">…</footer>
    ▶<script>…</script>
    ▶<script type="application/ld+json">…</script>
```

# Screen/web scraping workflow

Installing/loading  R package: RVest, tidyverse, ggplot

Choosing webpage

Extracting content

Basic cleaning

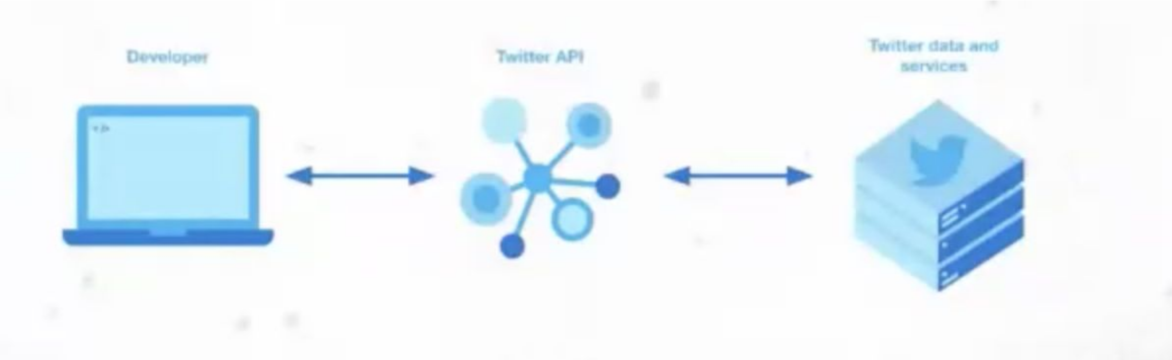
Exploratory data analysis

# Tool time: web/screen scraping

Code along!

# Using API

Application Programming Interface: customized requests of data from the server/owner



# Available APIs

Reddit API

Twitter API

NYT API

Spotify API

...

https://docs.google.com/spreadsheets/d/1ZEr3okdlb0zctmX0MZKo-gZKPsq5WGn1nJOxPV7al-Q/edit#gid=0

# Strengths and weakness of API

Strengths: efficiency, personalization, automation, integration, broad scope, structured

Weaknesses: availability, query limitation, affordability, maintenance

Social media companies like Reddit and Twitter are chasing the almighty dollar at the expense of its users and its own platforms. Credit: Avishek Das/SOPA Images/LightRocket via Getty Images

# API workflow

Read API documentation

*Sign up for a developer account

*Once approved, set up an App

*Save your credentials (keys and tokens)

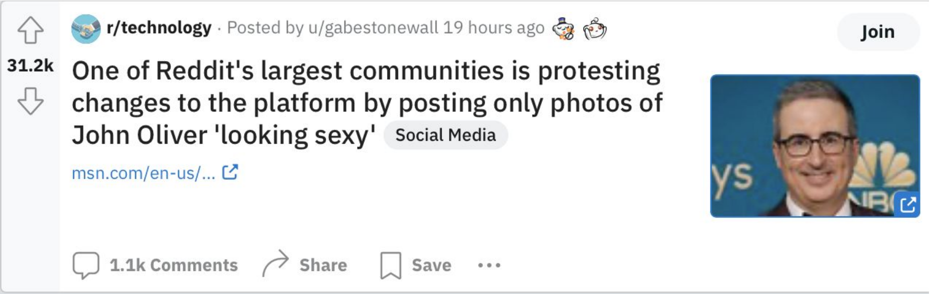
*Set up user authentication information

Get your R environment set up

Start your first API request

Data wrangling

# Reddit



r/technology · Posted by u/gabestonewall 19 hours ago

One of Reddit's largest communities is protesting changes to the platform by posting only photos of John Oliver 'looking sexy'

msn.com/en-us/...

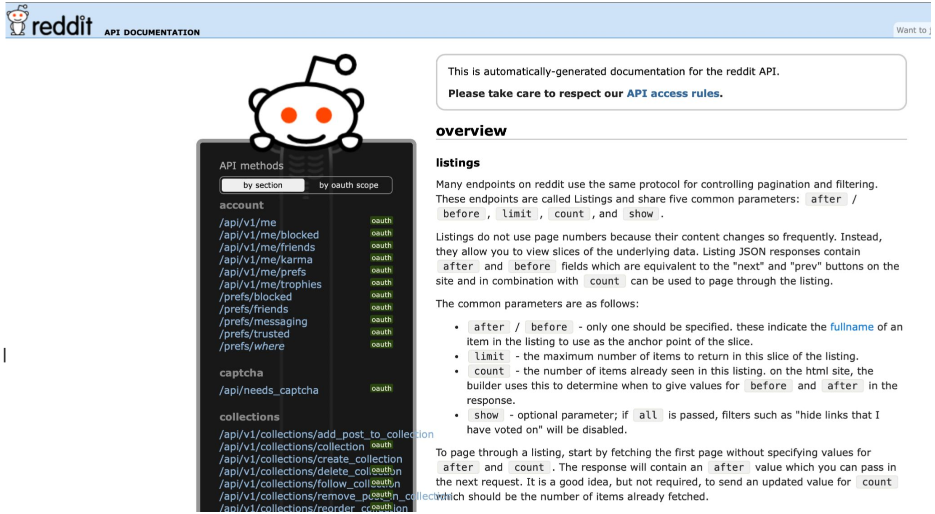
- 100,000+ active communities
- Over 57M daily active uniques
- Over 50,000 daily active moderators (mods)
- More than 80% of the top 5,000 communities (by DAU) are open

Post Subreddit User Comment Upvotes

# Reddit API

The Reddit API allows programmatic control of virtually every fu
user can perform on the site.

- gather and analyze data for academic research
- gather and analyze data for brand management
- monitor and moderate content on the platform
- build recommendation systems based on trends and user interests
- build bots for automated tasks (posting)
- …



API methods

by section | by oauth scope

**account**

/api/v1/me oauth
/api/v1/me/blocked oauth
/api/v1/me/friends oauth
/api/v1/me/karma oauth
/api/v1/me/prefs oauth
/api/v1/me/trophies oauth
/prefs/blocked oauth
/prefs/friends oauth
/prefs/messaging oauth
/prefs/trusted oauth
/prefs/*where* oauth

**captcha**

/api/needs_captcha oauth

**collections**

/api/v1/collections/add_post_to_collection
/api/v1/collections/collection oauth
/api/v1/collections/create_collection
/api/v1/collections/delete_collection oauth
/api/v1/collections/follow_collection oauth
/api/v1/collections/remove_post_in_collection oauth
/api/v1/collections/reorder_collection oauth

This is automatically-generated documentation for the reddit API.

**Please take care to respect our API access rules.**

## overview

### listings

Many endpoints on reddit use the same protocol for controlling pagination and filtering. These endpoints are called Listings and share five common parameters: `after` / `before` , `limit` , `count` , and `show` .

Listings do not use page numbers because their content changes so frequently. Instead, they allow you to view slices of the underlying data. Listing JSON responses contain `after` and `before` fields which are equivalent to the "next" and "prev" buttons on the site and in combination with `count` can be used to page through the listing.

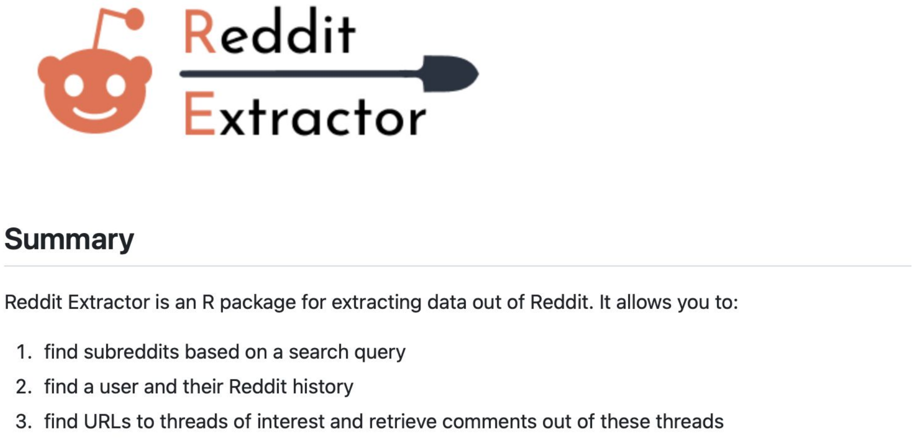
The common parameters are as follows:

- `after` / `before` - only one should be specified. these indicate the fullname of an item in the listing to use as the anchor point of the slice.
- `limit` - the maximum number of items to return in this slice of the listing.
- `count` - the number of items already seen in this listing. on the html site, the builder uses this to determine when to give values for `before` and `after` in the response.
- `show` - optional parameter; if `all` is passed, filters such as "hide links that I have voted on" will be disabled.

To page through a listing, start by fetching the first page without specifying values for `after` and `count` . The response will contain an `after` value which you can pass in the next request. It is a good idea, but not required, to send an updated value for `count` which should be the number of items already fetched.

# R package for interacting with Reddit API

RedditExtractoR



## Summary

Reddit Extractor is an R package for extracting data out of Reddit. It allows you to:

1. find subreddits based on a search query
2. find a user and their Reddit history
3. find URLs to threads of interest and retrieve comments out of these threads

# Tool time: Reddit API

Code along!