GESIS Fall Seminar in Computational Social Science 2022

Syllabus for week 2:
"Automated Web Data Collection with R"

Lecturers:	Dr. Theresa Gessler	Dr. Hauke Licht
Affiliation:	University of Zurich	University of Cologne
Email:	gessler@ipz.uzh.ch	hauke.licht@wiso.uni-koeln.de

Date: September 12-16, 2022
Time: 09:30-12:30 and 13:30-16:30

About the Lecturers

Theresa Gessler is a postdoctoral researcher at the Digital Democracy Lab and the Department of Political Science of the University of Zurich. In her research, she uses text analysis and computational methods, based on data collected from different online and offline sources. Besides her interest in computational social science, Theresa works on (digital) democracy, immigration, and gender issues.

Hauke Licht is a postdoctoral researcher at the Cologne Centre for Comparative Politics, University of Cologne, and has received his PhD from the University of Zurich. text-as-data methods to study political competition and democratic representation with a strong focus on cross-lingual applications. In this research, he frequently relies on collecting textual data at scale by applying different web scraping techniques.

Course Description

The increasing availability of large amounts of data on the internet enables new lines of research in the social sciences. Although it has become easier to find data online that is relevant to social science research, such as social media content, election results, or organizations' press statements, extracting these data and bringing it into formats ready for downstream analyses can be challenging. Web data collection is thus an essential skill for researchers.

The goal of this course is to enable participants to collect web data and process it in R for their research. Course participants will learn about the characteristics of web data and their use in social science research, how to harvest content from different types of webpages, and how to collect social media data from *application programming interfaces* (APIs), such as the Twitter API.

We will cover tools and techniques that enable participants to collect web data relevant to their research and focus on two common scenarios in particular: (i) automating the collection of data presented on multiple web pages (e.g., several pages) of both static and dynamic websites (with RSelenium), and (ii) interacting with APIs to, for example, collect social media data or datasets from institutions, companies, and organizations. In addition, we will cover advanced topics such as using web sessions, interacting with HTML forms (e.g., login), managing user agents, error handling, and headless browsing.

The course is hands-on, with daily lectures followed by exercises where participants can practice their newly learned skills.

Keywords

Web data, APIs, data harvesting, automated data collection, web scraping

Course Prerequisites

- Basic knowledge of the R programming language.
- Willingness to engage with different web technologies
- Knowledge of `tidyverse` R packages (recommended)

Target Group

Participants will find the course useful if they want to

- collect larger amounts of web data from APIs or webpages
- learn about best practices in automated web data collection
- improve their existing web scraping skills by deepening their understanding of common web technologies and learning more about the process of developing robust web scrapers

Course and Learning Objectives

By the end of the course participants will:

- Know the most important characteristics of web data, including webpage content and social media data
- Gain an understanding of a variety of scraping scenarios: APIs, static pages, dynamic pages
- Be able to write reproducible and robust code for web scraping tasks
- Be able to parse, clean, and process data collected from the web

Organizational Structure of the Course

The course will be organized as a mixture of lectures (morning sessions) and exercises (afternoon sessions). In the lectures, we will focus on explaining core concepts and methods in web scraping. In exercise sessions, participants will apply their newly acquired knowledge while the instructors will be available for individual consultations and support work on assignments.

Software and hardware requirements

Participants should bring their own laptops and pre-install the following software:

- RStudio (or a comparable R interface/IDE)
- the *Google Chrome* web browser
- suggested R packages (an updated list of packages will be provided before the course)
    - for web scraping: `rvest`, `httr`, `RSelenium`, `rtweet`
    - for data processing: `dplyr`, `tidyr`, `purrr`

Prerequisites

Course materials

A list of required and suggested readings and online resources will be distributed four weeks before the course. All data that will be used throughout the course will be provided by the instructors before course sessions.

While we work with prepared examples and data throughout the course, we encourage participants to develop ideas how they want to use web data in their own research and present these ideas in class. For this purpose, we recommend the following reading as preparation: Salganik, Matthew (2017) *Bit by Bit: Social Research in the Digital Age.*

Pre-course Survey

Participants will be asked to indicate their prior experience with web scraping, their research interests and potential web scraping-related project ideas in a pre-course survey. Based on this survey, the instructors will attempt to include examples in the afternoon tutorial sessions that match participants' research interests and project ideas.

R Programming skills

The course is targeted to participants with basic R programming knowledge. Participants should make sure *before the course* that they are familiar with the following R programming concepts and techniques:

- primary data object classes (vectors, lists and data frames)
- data wrangling (manipulating vectors, lists and data frames; reshaping/pivoting data frames),
- `for` loops and (ideally) functions in the apply/map families (`map_*` in the `purrr` package)
- writing simple functions.

We will briefly recap these topics in the afternoon session of the first day of the course. However, if participants are not yet familiar with these topics, we recommend taking the corresponding free online short tutorials in the SICSS R Bootcamp: https://sicss.io/boot_camp

Recommended Literature to look at in advance

Salganik, Matthew (2017) *Bit by Bit: Social Research in the Digital Age.* **Princeton University Press. (Chapters 1 & 2)**

Day-to-day schedule

Day 1: Introduction

We will cover what web scraping is and how it can be used in social science and digital humanities research. Participants will be asked to share their expectations of the course and how they plan to use web scraping in their research. We will then introduce most fundamental concepts including APIs, the XML and HTML formats, and how websites are commonly organized.

In the afternoon tutorial session, we will first ensure that all participants have a working setup (incl. a Twitter developer account). We will then have series of coding exercises designed to ensure that all participants are comfortable with basic R programming concepts and techniques (see *Prerequisites* section above).

Day 2: Social media data & APIs

Building on the content discussed on Day 1, we will deepen participants' understanding of APIs, discussing common APIs for data sharing, as well as social media APIs. Using the Twitter API as an example, we will then show how to use the `rtweet` package to query Twitter data.

To enable participants to potentially also interact with APIs for which no R package exists (yet), we will *(i)* show how to send requests to APIs using the `httr` R package and *(ii)* explain the JSON format – the data format commonly returned by APIs. This part of the session will also include a primer on authentication, pagination, and API rate limits.

In the afternoon tutorial session, participants will learn to apply this knowledge with a small project on the Wikipedia API.

Day 3: Scraping static websites

On day 3, we will introduce how to web scrape *static* websites. Building on our general discussion of HTML (Day 1), we will cover how to systematically extract web data by introducing the *CSS selector* and *xpath* methods.

In practical applications, we will use the `rvest` R package to show how to *(i)* extract data (text, hyperlinks, tables, images and other media, as well as meta data) from web pages and *(ii)* how to automatically navigate between and scrape multiple pages of a websites.

In the afternoon tutorial session, participants will learn how to apply this knowledge to different webpages.

Day 4: Scraping dynamic websites

On the third day of the course, we will go one step further and discuss how to scrape *dynamic* websites. We will first explain what makes a page "dynamic" and show how to recognize dynamic web elements in the wild.

We will then introduce the `RSelenium` package and show how it enables systematic interaction with dynamic web elements. This will include how to setup a web driver in R (Google Chrome), how to click on web elements (e.g., to unfold/collapse drop-down elements) in an automated way, how to navigate dynamic elements (e.g., accordion elements), how to switch between windows (e.g., a main page and a pop-up), and how to automatically download files. In the afternoon, participants will have the opportunity to practice these skills.

Day 5: Advanced topics and web scraping ethics

On the last day we will begin with a recap of what we have learned during the previous four days. We will then cover the topics of Ethics in web scraping by collecting and discussing the best practices that have been taught during the first four days.

The second part of the day will focus on advanced topics in web scraping, including web sessions, user agents, proxies, login, iframes, web crawling, and other topics participants might be interested in. We will also discuss tools for the advanced parsing of webpage content, including regular expressions.