GESIS Fall Seminar in Computational Social Science 2024
“Web Data Collection with R and/or Python”

Lecturer: Iulia Cioroianu
Affiliation: University of Bath, UK
Email: iulia.cioroianu@gmail.com

Date: September 9-13, 2024
Time: 09:00-10:30 and 13:00-15:30 (R)
11:00-12:30 and 13:30-16:00 (Python)

About the Lecturer
Iulia Cioroianu is a Senior Lecturer (Associate Professor) in the Department of Politics, Languages and International Studies at the University of Bath. She holds a Ph.D. in Political Science from New York University and an M.A. from Central European University. Before joining the University of Bath, she was a research fellow in the Q-Step Centre for Quantitative Social Sciences at the University of Exeter, and a pre-doctoral fellow in the LSE Department of Methodology. Iulia is a data scientist who studies online political communication and information exposure using a range of computational methods such as natural language processing and quantitative text analysis, agent-based modelling, machine learning and experiments. Her work was published in Electoral Studies, Social Networks, Economics Letters, Journal of Corporate Finance and AAAI conference proceedings, and was features in NCRM podcasts and Sage research methods videos.

Course Description
The exponential increase in online and social media data offers unprecedented opportunities for advancing research across a variety of fields, both within academia and outside of it. This course provides researchers the tools needed to collect and pre-process large-scale data from a range of online sources. The course will be offered both in R and in Python. Students can attend taught sessions in both programming languages in the morning, and can choose their preferred language for individual/group work and exercises in the afternoon. The content and examples used in the lecturer-led tutorials are similar across programming languages, making it easier for those interested in developing new skills in a secondary language that they may not be proficient in to do so by drawing parallels across the two sessions.

Through a combination of lectures, hands-on tutorials and individual/group exercises, participants will develop a theoretical understanding of the challenges associated with online data collection and the best methods and tools for addressing them in R and in Python, as well as the practical skills needed to collect data through Application Programming Interfaces (APIs), navigate dynamic websites and scrape data from both static and dynamic web pages. The sources used in the examples provided include social media websites, online media outlets and news aggregators, government data portals and other large-scale online data repositories.

Acknowledging that the most difficult part of a computational project involving the collection of complex and heterogenous data is often the pre-processing needed to prepare the data for subsequent analysis and link it across a variety of sources, the course also covers text-based methods for data cleaning and pre-processing. By the end of the week, participants should be able to apply the methods studied to extract and process data for their own research projects.

Organizational Structure of the Course
The course will consist of taught morning sessions in R and Python, and ample opportunities for student independent and group work in the afternoon. In each programming language, morning sessions consist of a short lecture laying out the main notions and providing an overview of the language-specific tools used (approximately 30 minutes), followed by a hands-on lecturer-led tutorial (1 hour). In the afternoon, students will have the
opportunity to solve exercises in their chosen programming language working independently or in small groups, with the support of the lecturer and one teaching assistant (2 hours). Solutions to the exercises will be provided and discussed in the final part of the day (30 minutes). The daily schedule by programming language group is presented in the table below.

<table>
<thead>
<tr>
<th>Activity</th>
<th>R</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical overview and lecturer-led tutorial</td>
<td>9:00-10:30</td>
<td>11:00-12:30</td>
</tr>
<tr>
<td>Individual or small-group exercises and solutions</td>
<td>13:00-15:30</td>
<td>13:30-16:00</td>
</tr>
</tbody>
</table>

Keywords

Web scraping, automated data collection, APIs, R, Python

Target Group

You will find the course useful if:
- You want to learn how to collect and process large amounts of data from online sources fast.
- You aim to improve your existing web scraping skills or have so far encountered difficulties trying to scrape data from online sources.
- You have a research idea for which online data might be suitable, but you are not sure of the practical implications.

Course and Learning Objectives

By the end of the course you will:
- Understand the structure and basic features of different forms of online data.
- Be able to collect data from static and dynamic websites.
- Be able to interact with APIs to access and collect data.
- Be able to parse, clean and process the data collected.
- Be able to apply the methods studied to their own research projects.

Course Prerequisites

- Working knowledge of R and/or Python, including data structures and control structures.
- Participants attending both the R and the Python sessions should have a basic level of knowledge in each of the two programming languages.
- If you lack basic knowledge of these programming languages, you are encouraged to take the Introduction to Computational Social Science with R or Introduction to Computational Social Science with Python course in week 1 and/or the introductory online workshops (Intro to R, Intro to Python).

Software and Hardware Requirements

Participants should bring their own laptops for use in the course and pre-install the following software:

**R:**
- RStudio
- required packages (final list of packages to be provided before the course): httr, rvest, RSelenium, dplyr, tidyr, stringr, quanteda

**Python:**
- Python 3
- Required packages (final list to be provided before the course): requests, lxml, BeautifulSoup, Selenium, pandas, re, stringr, NLTK
Recommended Literature to Look at in Advance

*R refresher:*

*Python refresher:*

*Or a refresher on R and Python side-by-side:*

*Other recommended readings:*

Day-to-day Schedule and Literature

**Day 1: Introduction to Web Scraping**
We will begin with an overview of web scraping, its applications in social science research, and the ethical and legal issues associated with its use. The morning sessions will provide a general idea of the structure and logic of websites and APIs and introduce the core concepts that we will be working with, as well as the language-specific libraries that we will be using. We will discuss the structure and features of different forms of online data. Participants will share their expectations and discuss their intended use of web scraping for their own research.

The afternoon sessions will ensure that participants’ Python and R setups are functional. We will also make sure that everyone has all the required credentials for working with different APIs later in the course. A series of exercises will focus on reinforcing basic programming and data manipulation skills. Participants will then make their first HTTP request and we will explore the output together in preparation for next day’s session.

*Literature (suggested)*
Day 2: Scraping Static Websites
During the second day we will cover methods of data extraction from websites for which the content is fixed and does not change based on user interaction. Building upon the final lab exercise from the previous day, we will discuss the basics of HTTP requests and how web pages are structured using HTML and CSS. We will then cover different methods for locating and extracting various data types from web pages, including text, hyperlinks, tables, images and other media, as well as metadata, and go over some practical examples using BeautifulSoup in Python and rvest in R. An introduction to regular expressions will also be provided.

In the afternoon participants will work through a series of exercises designed to test and reinforce their skills, including capturing article titles, authors, and publication dates from a news archive, and collecting indicators from the websites of governments and international organizations. Each exercise will be structured as a mini research project that participants will have to prepare and implement either in a group or individually.

Literature (suggested)
BeautifulSoup documentation: https://beautifulsoup-4.readthedocs.io
Rvest documentation: https://cran.r-project.org/web/packages/rvest/rvest.pdf

Day 3: Scraping Dynamic Websites
We will begin the third day with an overview of the unique challenges of dynamic websites that make traditional scraping techniques inadequate, and the discuss the main ways in which we can overcome them programatically. We will then demonstrate the use of Selenium in Python and RSelenium in R to automate and simulate user interactions with web pages through several examples. We will learn to handle dynamic pagination, JavaScript code and infinite scrolling, and to manage browsing sessions, requests to authenticate and rate limits.

In the afternoon, participants will have the opportunity to apply these skills in different scenarios, including collecting online financial data, automating online database queries, and collecting data from search engines and social media websites that would be otherwise difficult to access.

Literature (suggested)
Selenium documentation: https://www.selenium.dev/documentation/
Python: https://selenium-python.readthedocs.io/
R: https://cran.r-project.org/web/packages/RSelenium/RSelenium.pdf

Day 4: Working with APIs
Application Programming Interfaces (APIs) can be used to access and collect a range of data that is relevant for social science research. We will start with an introduction to APIs—what they are, how they function, and why they are important tools for researchers looking to access structured data directly from online platforms. We will learn about endpoints, requests, responses, and authentication methods, and go over several examples based on a simple but powerful political information API (Vote Smart) as well as the Reddit API.

In the afternoon sessions, participants will apply the new skills to extract data from other APIs, such as the Manifesto Project API, the YouTube Data API and the (limited access) X (Twitter) v2 API.

Literature (suggested)


Day 5: Processing, Storing and Starting to Analyze the Collected Data

The final day will provide essential skills for handling the collected web data from initial collection and storage to final analysis. The first part of the day focuses on data pre-processing and cleaning. Participants will learn the essential techniques for transforming raw data into formats that are suitable for storage, sharing and analysis. We will then explore different storage options and the types of databases suitable for large-scale web data. We will discuss relational databases such as MySQL, NoSQL databases like MongoDB, and cloud storage solutions, and evaluate the choice of database based on the project’s scale, data type, and accessibility requirements. We will briefly consider the process of writing the documentation for the resulting dataset to facilitate research sharing and replicability. Finally, we will focus on text cleaning, processing, and analysis tasks that are commonly used for web data, and provide an overview of text analysis techniques such as sentiment analysis and topic modeling.

In the afternoon sessions participants will have the opportunity to apply these skills to some of the data collected in previous days. By the end of the course participants will have a portfolio of thoroughly documented web scraping examples that can be replicated or adapted for other use cases.

Literature (suggested)

Python NLTK documentation: https://www.nltk.org/
R Quanteda documentation: http://quanteda.io/

Additional Recommended Literature

The following books are useful as reference material and for other/more specialized tasks: