SICSS- Istanbul 2020 Pre-SICSS Tasks

WELCOME

WARNING: *All of the recommendations are based on our experience; therefore they can be highly subjective. We do not argue that the packages&approaches we are suggesting in this document are the best ones. However, we are using some of them for a long time, and they have been handy.*

You could find SICSS material following these links: Lecture Materials link1 link2

Coding social science by Matti: link3

Data Project Steps

Every data project generally has four critical milestones. Which are:

- Data collection
- Data cleaning
- Data analysis
- Reporting

Tasks

Theory and practise rarely meet in the real world. Therefore instead of writing tutorials which are long, fancy and full with nice codes, we are going to suggest three introductory guides and two abstract case studies, and provide guidance every week till SICSS via zoom and Slack.

The tasks cover data collection, cleaning and reporting. We are leaving the analysis to SICSS week.

When you face a bug or difficulty, instead of immediately asking the TAs, we strongly suggest that you should try to solve your problems by searching on StackOverflow and instead of copy-pasting the correct answer you should type it!

If you still have problems, please post your questions at least one hour before zoom meetings. Ahmet will be able to organize zoom meetings on Wednesday and Friday between 19:30 - 20:30 (Turkish time) and Yunus Emre will organize zoom meetings on Slack. We will try to solve anything you ask, but we will have to prioritize due to time constraints.

Data collection (Monday - Tuesday)

General Information

In computational science, collecting data is more accessible than conventional social science methods. You can literally gather data while you are sleeping! Internet is full of data and waiting for you to collect them.

There are two standard techniques you can use:

- Using Application Programming Interface[1][2][3]. Most of the social networking platforms provides with an API (Twitter, YouTube, Facebook). To use an API, you should obtain credentials and comply with the terms of service.
- Using web scraping. Web scraping is the "free" way of obtaining data from the Internet. However, still, you should be careful about ethical and legal issues. You should leave some time between page requests to prevent an unintended DoS attack.

Here are some tutorials to collect data with R:

- Twitter API
- YouTube API
- Web Scraping with rvest

What you should do in this task?

- First of all, it is always quite exciting to start a new project, so congratulations! We expect you to collect data from Twitter API and scrape at least five HTML pages in this task. **You should be able to finish this task in two days.**
- **Twitter:** If you don't have Twitter API, please apply from here as soon as possible. Download at least 20K tweets and store them in a data.table.
- **HTML Scrape:** Go to a page you are interested in and scrape it by using rvest. Assign each `<p>` element to a row in a data.table. Create date, group, title and URL variables in your data. Repeat this step for at least 5 different pages of the web site and combine all data into one single dataset.
- Save your data.

Data Cleaning (Wednesday - Thursday)

General Information

There are two popular libraries to handle data in the R environment. The first one is data.table package and the second one is dplyr package. I am not stressing pros/cons of these two packages, but you should read this post before learning either one. In my view, you should give both a chance and stick with the one that comes natural to you. I can disclose my preference and why I am using it in "real" big data projects during our zoom meetings!

If you are a social scientist, one of the most crucial variables in your projects would be text. To apply a fancy text analysis, you must have the ability to clean text. Text analysis is a long and adventurous journey, and it always starts with basic string operations. You should start learning string functions with stringr and stringi packages.

On the other hand, you should learn a little bit regular expressions and test your expressions before using them. You could develop very complex expressions to solve your problem quickly when string libraries are not enough.

Once you finished with strings in your dataset, you go beyond and discover your data. Our number one suggestions would be quanteda and quanteda.textmodels packages. It is quite easy and fun working with these packages.

What you should do in this task?

- Read your Twitter and web datasets.
- Subset your data by using a keyword, a time interval, a user name for the Twitter dataset.
- Order your data alphabetically and by weekday! You can use lubridate for date-time operations.
- Understand what is a merge operation and merge your datasets based on a textual trait.
- Create a quanteda corpus object based on Twitter dataset text column. Assign date and tweet ID as metadata
- Create a document feature matrix from your corpus and find top 10 frequent terms.
- Weight your dfm by using tf-idf scheme

You should be able to compl

Reporting (Friday)

One of the most dominant sides of the R is reporting the data. ggplot2 package would provide diverse options from basic scatterplots to drawing complex maps and networks. Once you understand how to build a basic plot with ggplot, it is quite easy to solve any problem and produce classy graphics. You should learn how to use it!

The second essential skill you should learn is working with markdowns. You could effortlessly produce documents like this one.

Last but not least, if you are fluent with basic R, you could produce great data products with Shiny. It is not the most straightforward interface library, but you should give it a try once you are ready!

What you should do in this task?

- Load you datasets you created in Task 1 and Task 2.
- Create a new Rmd document in RStudio
- Display your results in an HTML document

You should be able to complete this step in one day.

Twitter textual network analysis
(Second week - Monday - Tuesday)

Package suggestions: rtweet, data.table, quanteda, ggplot2

In this task, you should register Twitter as a developer.

We provide a frame below, and you should fill the gaps.

- Sign in to your Twitter developer account and create a new application you later use to crawl tweets.
- Authenticate your account
- Read here what are the limitations of Twitter API's
- Decide on some interesting hashtag and use is as search term crawl 10K tweets. You should decide whether you want to use Streaming API or Search API and understand the difference!
- Clean all punctuation except hashtags and mentions in the text variable. It is quite easy and well documented on the Internet; however, I suggest you should think about the problem. Instead of thinking it as a whole, divide it into small pieces and start bit by bit! For example, think about removing only one type of character and create a pattern.
- Following this quanteda tutorial create a document feature matrix base on your text variable.
- Display your results in a markdown document with explanations. In addition to quanteda's pretty plots, you could include histograms or even a network of tweeters!

You should be able to complete this step in two days.

Web Scraping (Second week - Wednesday - Thursday)

Packages: rvest, data.table, ggplot2,

In this task, you should get data from your favourite website. You should read the terms of the website before scraping data and, be careful; you should allocate enough time between new requests while scraping web!

To understand web scraping and be successful at it, you should know at least basic HTML structure. In addition, you need to learn what are a tag - HTML element- , css selector and xpath. Note you don't have to write xpath, you can simply copy it from your browser. However, you should open the inspector!

- Go to your favourite web page.
- Copy the URL address and start crawling the page.
- After you downloaded one page inspect it with rvest functions
- Write a while loop to trace all related pages.
- Extract information from the pages.
- Analyze the text as you did in Task 1

You should be able to complete this step in two days.

READING LIST

- Grimmer, J. (2015). We Are All Social Scientists Now: How Big Data, Machine Learning, and Causal Inference Work Together. PS: Political Science & Politics, 48(01), 80–83. https://doi.org/10.1017/S1049096514001784
- Kitchin, R. (2014). Big Data, new epistemologies and paradigm shifts. Big Data & Society, 1(1), https://doi.org/10.1177/2053951714528481
- Granovetter, M. S. (1973). The Strength of Weak Ties. American Journal of Sociology 78(6). https://www.jstor.org/stable/2776392
- Coleman, J. S. (1986). Social Theory, Social Research, and a Theory of Action. American Journal of Sociology 91(6). http://www.jstor.org/stable/2779798
- Tufekci, Z. (2014). Engineering the public: Big data, surveillance and computational politics. First Monday, 19(7). https://doi.org/10.5210/fm.v19i7.4901
- Unver, A. (2019). Internet, Social Media and Conflict Studies Can Greater Interdisciplinarity Solve the Field's Analytical Deadlocks? St. Anthony's International Review Vol. 14, No. 3. https://arxiv.org/abs/1905.01777
- Unver, A. (2019). Computational IR: What Coding, Programming and Internet Research Can Do for the Discipline? All Azimuth: A Journal of Foreign Policy and Peace Vol. 8, No. 2. https://doi.org/10.20991/allazimuth.476433
- Edelmann, A., Wolff, T., Montagne, D., & Bail, C. A. (2020). Computational Social Science and Sociology. Annual Review of Sociology, 46(1). https://doi.org/10.1146/annurev-soc-121919-054621