Introduction to Computational Social Science

(Week 13: "Other" CSS skills)

Aliakbar Akbaritabar[1]

[1]Max Planck Institute for Demographic Research, Rostock, Germany;
akbaritabar@demogr.mpg.de

Intro to CSS course outline and syllabus

Universität Rostock

- Week 1 – Introduction ... What is "Computational Social Science"? Inductive and deductive research; Big data revolution
- Week 2 – Open Science and Reproducibility ... Reproducibility crisis; Pre-registration; Version control, Git, GitHub, Accessing course materials
- Week 3 – Digital Trace Data ... Observational data; Available vs. designed data; APIs and web scrapping; Representativeness
- Week 4 – Mobility and Migration ... Computational approaches to migration research; The Scholarly Migration Database
- Week 5 – Network Analysis ... Migration networks; Tie formation mechanisms in social networks; Violence of independence of observations
- Week 6 – Science of Science ... Robert K. Merton; Sociology of scientific knowledge vs Bibliometrics/ Scientometrics
- Week 7 – Text as Data ... Natural Language Processing; Topic modelling; LDA; Structural Topic Models
- Week 8 – Ethics in Computational Social Science ... Informed consent; Personal data; GDPR
- Week 9 – Social Simulation ... Agent-based modelling for social scientists; Micro-Macro link and Coleman's boat
- Week 10 – Machine Learning ... Supervised and unsupervised use of observational data; Feature learning
- Week 11 – "Other" CSS Skills ... Parallelization; Functional vs Object-Oriented; Graph databases; DuckDB; SnakeMake Workflows
- Week 12 – Limitations of Computational Social Science ... Pitfalls of digital trace data and computational approaches; Representativeness
- Week 13 – Conclusions ... Thick vs Big data, Survey experiments; Linked data; Future of CSS + LIST OF FINAL ASSIGNMENTS
- Week 14 – Final semester evaluation ... 6 weeks to prepare a short essay and empirical analysis on one of the CSS topics/skills

Intro to CSS course outline and syllabus (2025's updated timing)

Universität Rostock

- Week 1 — Introduction ... What is "Computational Social Science"? Inductive and deductive research; Big data revolution
- Week 2,3 — Open Science and Reproducibility ... Reproducibility crisis; Pre-registration; Version control, Git, GitHub, Accessing course materials
- Week 4,5 — Digital Trace Data ... Observational data; Available vs. designed data; APIs and web scrapping; Representativeness
- Week 6,7 — Mobility and Migration ... Computational approaches to migration research; The Scholarly Migration Database
- Week 8,9 — Network Analysis ... Migration networks; Tie formation mechanisms in social networks; Violence of independence of observations
- Week 10 — Science of Science ... Robert K. Merton; Sociology of scientific knowledge vs Bibliometrics/ Scientometrics
- Week 11 — Text as Data ... Natural Language Processing; Topic modelling; LDA; Structural Topic Models
- Week 12 — Ethics in Computational Social Science ... Informed consent; Personal data; GDPR
- ~~Week 00~~ — Social Simulation ... Agent-based modelling for social scientists; Micro-Macro link and Coleman's boat
- ~~Week 00~~ — Machine Learning ... Supervised and unsupervised use of observational data; Feature learning
- Week 13 — "Other" CSS Skills ... Parallelization; Functional vs Object-Oriented; Graph databases; DuckDB; SnakeMake Workflows
- Week 13 — Limitations of Computational Social Science ... Pitfalls of digital trace data and computational approaches; Representativeness
- Week 13 — Conclusions ... Thick vs Big data, Survey experiments; Linked data; Future of CSS + LIST OF FINAL ASSIGNMENTS
- ~~Week 00~~ — Final semester evaluation ... 6 weeks to prepare a short essay and empirical analysis on one of the CSS topics/skills

DISCLAIMER:

I will only share leads, links, and show you “example” scripts in R, Python, and SQL in this introductory short session; you would need to extend them on your own.

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;

Variables/attributes

Respondents/ Observations

ID	Name	Age	Political view	Education
1	Tom	24	left	NA
2	Sara	22	right	BA
3	Bill	30	neutral	MA
4	Margaret	31	NA	PhD

Example variable by observation table. Rows are independent!

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;

An example of the NDJSON format.

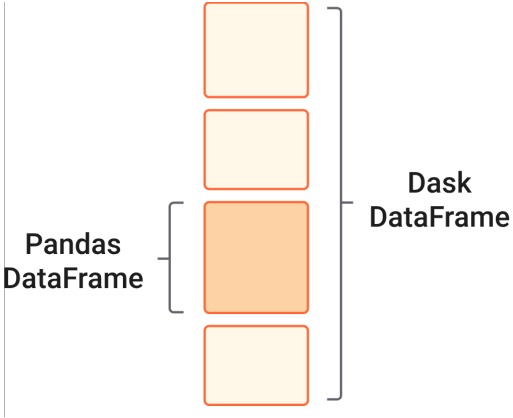
ndjson-sample.json

```
{"url":"https://www.yelp.com/search?find_desc=Desserts&find_loc=San+Jose,
{"url":"https://www.yelp.com/search?find_desc=Desserts&find_loc=San+Jose,
{"url":"https://www.yelp.com/search?find_desc=Desserts&find_loc=San+Jose,
{"url":"https://www.yelp.com/search?find_desc=Desserts&find_loc=San+Jose,
{"url":"https://www.yelp.com/search?find_desc=Desserts&find_loc=San+Jose,
```

Example unstructured text data, ndjson (same as jsonl) from: https://gist.github.com/rfmcnally/0a5a16e09374da7dd478ffbe6ba52503 Lines are independent.

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;



Example Dask data frame from:
https://docs.dask.org/en/stable/dataframe.html
to show independence of rows even in a table!

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;

Load Data | Data Processing | Machine Learning

Pandas and Dask have the same API, and so switching from one to the other is straightforward.

```
>>> import pandas as pd

>>> df = pd.read_parquet('s3://mybucket/myfile.parquet')
>>> df.head()
0  1  a
1  2  b
2  3  c
```

```
>>> import dask.dataframe as dd

>>> df = dd.read_parquet('s3://mybucket/myfile.*.parquet')
>>> df.head()
0  1  a
1  2  b
2  3  c
```

Example Dask data frame syntax vs Pandas from: https://docs.dask.org/en/stable/dataframe.html.

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?

towards data science

DATA SCIENCE

Saving Pandas DataFrames Efficiently and Quickly – Parquet vs Feather vs ORC vs CSV

Speed, RAM, size and convenience. Which storage method is best?

Mike Clayton
Nov 27, 2024 • 15 min read

Example comparison of parquet, CSV, feather etc.:
https://towardsdatascience.com/saving-pandas-dataframes-efficiently-and-quickly-parquet-vs-feather-vs-orc-v

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;

Example

Insert a function that prints a greeting, and execute it on the p1 object:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)

p1 = Person("John", 36)
p1.myfunc()
```

Example Python class and methods from: https://www.w3schools.com/python/python_classes.asp.

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;

Add Methods

Example

Add a method called `welcome` to the `Student` class:

```
class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year

  def welcome(self):
    print("Welcome", self.firstname, self.lastname, "to the
class of", self.graduationyear)
```

Try it Yourself »

Example Python class inheritance from: https://www.w3schools.com/python/python_inheritance.asp.

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);

Variables/attributes

Respondents/ Observations

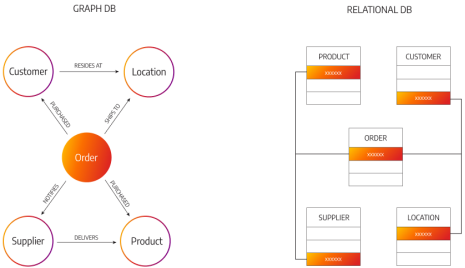
ID	Name	Age	Political view	Education
1	Tom	24	left	NA
2	Sara	22	right	BA
3	Bill	30	neutral	MA
4	Margaret	31	NA	PhD

Example violation of independence of observations, use a clustering framework or network graph/modeling?!

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);



Example difference of tabular/relational versus graph DB from: https://memgraph.com/blog/graph-database-vs-relational-database.

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;

"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;
- In general, check DuckDB's community extensions https://duckdb.org/community_extensions/list_of_extensions
- If you want to run Graph analytics, see "duckpgq" https://duckdb.org/community_extensions/extensions/duckpgq.html

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;
- DuckDB's interface to R, Python, etc, to manage I/O and ETL tasks (Needham et al., 2024)

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;
- DuckDB's interface to R, Python, etc, to manage I/O and ETL tasks (Needham et al., 2024)
- Using Dask to parallelize familiar data constructs, Pandas DF, Numpy array (Daniel, 2019)

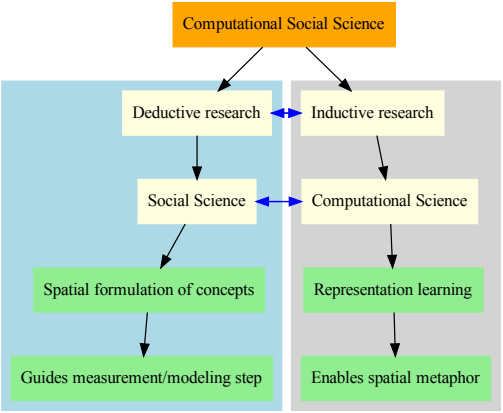
"Other" useful skills to have as a Computational Social Scientist

Universität Rostock

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;
- DuckDB's interface to R, Python, etc, to manage I/O and ETL tasks (Needham et al., 2024)
- Using Dask to parallelize familiar data constructs, Pandas DF, Numpy array (Daniel, 2019)
- Use workflow managers, e.g., SnakeMake, for reproducibility (Mölder et al., 2021)

"Other" useful skills to have as a Computational Social Scientist

- Parallelization of extract, transform, load (ETL) tasks in Python and R;
- Faster Input/output (I/O) w/ file formats such as Parquet, feather, versus row-based CSV; All columns needed? Strings?
- Functional programming versus Object-Oriented Programming;
- Tabular versus relational databases versus graph databases (Besta et al., 2021);
- In-memory databases for faster and parallelized ETL tasks such as DuckDB;
- DuckDB's interface to R, Python, etc, to manage I/O and ETL tasks (Needham et al., 2024)
- Using Dask to parallelize familiar data constructs, Pandas DF, Numpy array (Daniel, 2019)
- Use workflow managers, e.g., SnakeMake, for reproducibility (Mölder et al., 2021)



Check other discipline's skill sets, e.g., see more: Akbaritabar, 2024;

Parallelised analysis of large-scale bibliometric data w/ Dask
Using the example of ORCID 2019 XML files

Universität Rostock

- Python users see: paralleliza-tion_with_dask_precompiled.html
- Video of a tutorial I gave on this, https://youtu.be/pYDVrBcluYI, including:
  - 00:00 - 02:45; Introduction
  - 2:45 - 11:04; Requirements and installation
  - 11:05 - 38:00; Steps in using Dask/Python and results

Parallelised analysis of large-scale bibliometric data with Dask in Python, DuckDB/DBeaver in SQL

Aliakbar Akbaritabar
40 subscribers

Analytics | Edit video | 8 | Share | Promote

423 views Feb 21, 2022

Parallelised analysis of large-scale bibliometric data (with Dask in Python, DuckDB and DBeaver in SQL) Using example of ORCID 2019 XML files

Repository of codes/data: https://github.com/akbaritabar/dask-d...

30 minutes for the introduction and tutorial, 9 minutes for required installation that can be skipped if they followed tutorial's instructions provided on the repository (https://github.com/akbaritabar/dask-d....

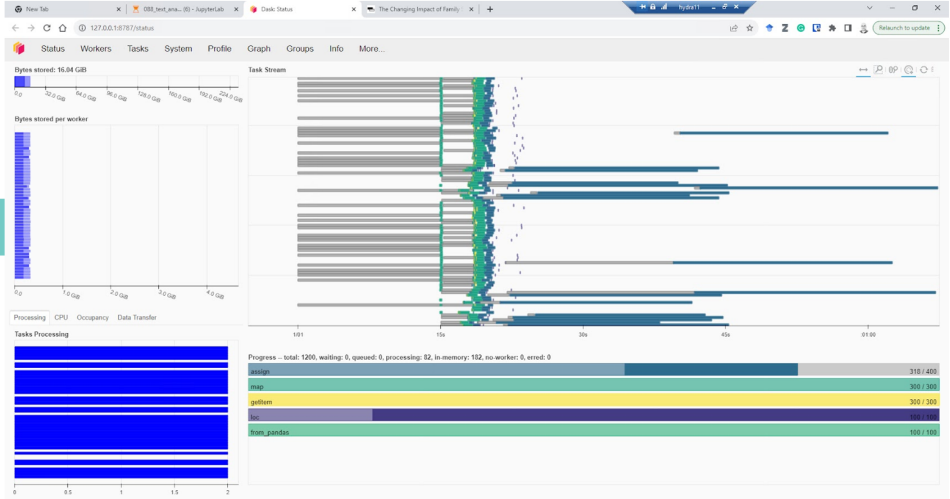
- From the beginning up to 02:45 – Introduction

Parallelization saves you a lot of time!

Parallelization saves you a lot of time!

Universität Rostock

Dask dashboard



See more on Dask: Daniel, 2019

For R

- Purrr package to run map-like queries not sequentially (Wickham et al., 2023), see: https://purrr.tidyverse.org/reference/pmap.html
- If you liked Purrr's idea, then check out Furrr: https://furrr.futureverse.org/
- Or use "data.table": https://rdatatable.gitlab.io/data.table/

- Or leave I/O to DuckDB in R
- Do "install.packages("duckdb")" and see examples in hands-on scripts

For any platform (CLI)

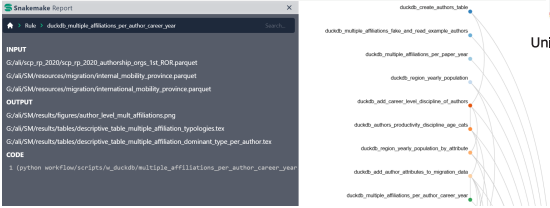
- DuckDB Command Line Interface (CLI): https://duckdb.org/docs/installation/
- Or browser shell (note resource limitations, i.e., 4GB RAM, 1 core): https://shell.duckdb.org/
- Check their book Needham et al., 2024

For Python

- Leave I/O to DuckDB in Python
- Do “pip install duckdb” and see examples in hands-on scripts
- Check User-defined-functions (UDFs) for more advanced jobs: https://duckdb.org/docs/stable/clients/python/function.html
- Check Ibis: writing Python with DuckDB backend, no SQL needed (but possible): https://ibis-project.org/tutorials/basics
- "Workflow management using SnakeMake" (e.g., Targets in R): https://snakemake.readthedocs.io/en/stable/ (Mölder et al., 2021)

For Python

- Leave I/O to DuckDB in Python
- Do “pip install duckdb” and see examples in hands-on scripts
- Check User-defined-functions (UDFs) for more advanced jobs: https://duckdb.org/docs/stable/clients/python/function.html
- Check Ibis: writing Python with DuckDB backend, no SQL needed (but possible): https://ibis-project.org/tutorials/basics
- "Workflow management using SnakeMake" (e.g., Targets in R): https://snakemake.readthedocs.io/en/stable/ (Mölder et al., 2021)



workflow > rules > snakefile_map_figures.smk

You, last month | 1 author (You)

```
## File name to use in search: snakefile_map_figures.smk ##

# ===========
## NMR and MEI MAPS for all time_span combinations ##
# ===========

rule plot_map_NMR_AND_MEI_MAPS:
    input:
        rules.prepare_data_for_mapping.output
    output:
        NMR_AND_MEI_MAPS
    log:
        NMR_AND_MEI_MAPS_LOG
    shell:
        "(python workflow/scripts/generic_src_mapping_figures.
        py --input {input} --MEASURE_MAPPED {wildcards.
        measure_mapped} --MIGRATION_SYSTEM {wildcards.
        migration_system} --GEO_REGION {wildcards.geo_region}
        --TIME_SPAN {wildcards.time_span} --output {output})
        2> {log}"
```

See: Akbaritabar and Dańko, 2025

Hands-on part

Needed installations (or checking requirements)

For the Hands-on part of this session, I used R, Python, and SQL with DuckDB. See the Readme file for this week.

Needed: Python (Vanilla or Miniforge), and virtualenv environments; R and Rstudio, DuckDB CLI

The example script shows 1) a toy example using these three languages, 2) a heavier example, and 3) a two for the price of one in Python, as I show how to use DuckDB inside SnakeMake workflows.

Where to learn basics of R and Python?

Universität Rostock

To start with Python

Check this website first:
https://www.w3schools.com/python/python_intro.asp

Check this repository by Vincent Traag and others, for an introductory course and code:
https://github.com/vtraag/intro-python

Or this one by Data Carpentry:
https://datacarpentry.org/python-ecology-lesson/

To start with R

Check this website first:
https://www.w3schools.com/r/default.asp

This "very short introduction to R" is a good start:
https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

Or this course by Data Carpentry:
https://datacarpentry.org/R-genomics/index.html

Reading materials for today

Key reading material

- Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021, April 19). Sustainable data analysis with Snakemake. 10:33. https://doi.org/10.12688/f1000research.29032.2

Additional reading materials

- Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023, June). R for Data Science. "O'Reilly Media, Inc."
- Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2021).Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. arXiv:1910.09017 [cs].
- Daniel, J. (2019, July). Data Science with Python and Dask. Manning Publications.
- Needham, M., Hunger, M., & Simons, M. (2024, August). Duckdb in Action. Manning Publications.

Thanks for your attention!

Questions and comments are welcome!

Aliakbar Akbaritabar

Research Scientist

Max Planck Institute for Demographic Research (MPIDR), Rostock, Germany.

To contact, please use Email: "aliakbar.akbaritabar@uni-rostock.de" and include in the subject line: "CSS-SoSe-2025"

LinkedIn/BlueSky/Twitter/Mastodon: @akbaritabar

References

- Akbaritabar, A. (2024).Thinking spatially in computational social science. EPJ Data Science, 13(1), 1–12. https://doi.org/10.1140/epjds/s13688-023-00443-0
- Akbaritabar, A., & Dańko, M. J. (2025, March). Scripts, data, and replication materials for "Global subnational estimates of migration of scientists reveal large disparities in internal and international flows". https://doi.org/10.5281/zenodo.15047102
- Besta, M., Peter, E., Gerstenberger, R., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2021).Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. arXiv:1910.09017 [cs].
- Daniel, J. (2019, July). Data Science with Python and Dask. Manning Publications.
- Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster, J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S., Nahnsen, S., & Köster, J. (2021, April 19). Sustainable data analysis with Snakemake. 10:33. https://doi.org/10.12688/f1000research.29032.2
- Needham, M., Hunger, M., & Simons, M. (2024, August). Duckdb in Action. Manning Publications.
- Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023, June). R for Data Science. "O'Reilly Media, Inc."