EXPLORATORY ANALYSIS OF WEB-SCRAPED DATASET FROM DR AND ALTINGET'S CANDIDATE-TEST

EXAM PAPER FOR AARHUS UNIVERSITY COURSE: INTRODUCTION TO CULTURAL DATA SCIENCE 147201U019

ASTRID ELMANN HANSEN
AU ID: 670438

Table of contents

Introduction

It is common knowledge that politics play a major role in shaping the society and the world we live in and thus have a direct impact on the everyday lives of people. It is thus essential for a democracy that a majority of citizens participate in elections, to ensure that the composition of elected politicians reflects the general population Although it is notoriously hard for immigrants to gain access to the Danish democracy (Garde Gräs, 2022), 84,2% of eligible voters participated in the latest election of Denmark's Folketing, the main governing body of Denmark (Danmarks Statistik, 2022).

For the people aspiring to become a part of the exclusive group of 179 politicians governing Denmark, they must be able to grab the attention and eventually the support of voters. Candidates to the Folketing are naturally aware of this. One way for candidates to reach voters, and for voters to gain information on who they can vote for is with a candidate test. These tests, also called Voting Advice Applications (VAAs), matches voters with candidates based on their answers to a number of questions.

While several candidate tests exist in Denmark, the test developed by Altinget and used by Danmarks Radio (DR) is the only one that will be focused on in this paper. DR is the Danish public service broadcaster and Altinget is an "impartial political niche media" (Altinget, 2013). Altinget has developed candidate tests for the Danish Folketing elections since 2001, with ongoing collaboration with election researchers in Denmark and the European research network ECPR Research Network on VAAs (ECPR Research Network, 2022).

The 2022 test by Altinget consists of 25 questions that all candidates running for election are invited to answer. The questions are formed as statements and answered on a Likert-type scale with 5 steps, 1: "completely disagree", 2: "disagree", 4: "agree", 5: "completely agree". It is not possible to answer 3, as it is unambiguous what this might represent, Nielsen notes that it could be interpreted as "status quo", "skip" or "don't care" amongst others. The decision to exclude 3 also serves to discourage candidates that might chose this option often to match with as many voters as possible. It is however possible to skip a question. It is also possible for the candidate to attach a short note to their answers. When the test is completed by the individual voter, the Manhattan distance formula is used to match the voter with candidates in the voter's constituency (Nielsen, 2021).

In this paper I will use web scraping in Python to acquire the answers of all candidates on Altinget and DR's candidate test for the Danish Folketing election in 2022. Consequently, I will use R to conduct an exploratory analysis of the resulting dataset. As with most exploratory analyses no hypothesis is stated, but the focus of the analysis will be to provide neat plots that will shed light on patterns that cannot be observed when just browsing through the candidate test online. Furthermore, I will collect data on which candidates were elected, and whether patterns in their answer style are linked with their success in the election. The entirety of the project can be seen in the associated GitHub repository, available at https://github.com/AddiH/CulturalDataScienceEXAM.

Software framework

The project was developed on a MacBook Pro (M1, 2020) running macOS Monterey version 12.5. Code was written in and executed from RStudio (2022.12.0+353) and Jupyter lab (3.5.0) using programming languages R (4.1.2) and Python

(3.9.5). Additional software metadata can be found in the appendix in table "Software metadata". Details, citation and version of specific libraries, packages, and modules can be found in the appendix in the table "Citations for software, modules, and packages". Examples of when modules were used are in bold in the following sections.

Data Acquisition and Processing

Sources

Data was acquired from DR from the following URLs:

Ballot: https://www.dr.dk/nyheder/politik/folketingsvalg/din-stemmeseddel

The first webpage (ballot) is an overview of all constituencies in Denmark. Each constituency contains the candidates in that constituency.

Constituency: https://www.dr.dk/nyheder/politik/folketingsvalg/din-stemmeseddel/*url key for constituency*

From the constituency page the URL-key for each candidate is available and when acquired, data on each candidate can be accessed through:

Candidate: https://www.dr.dk/nyheder/politik/folketingsvalg/din-stemmeseddel/kandidater/*url key for candidate*

As an example of a website for one individual, you can visit:

https://www.dr.dk/nyheder/politik/folketingsvalg/din-stemmeseddel/kandidater/419-sofie-lippert-troelsen

As 1014 individuals stood for election, thus 1014 pages were scraped for data. Data on elected candidates was accessed through:

Elected: https://www.dr.dk/nyheder/politik/folketingsvalg/valgte

Web-scraping

The following sections go through the .ipynb scripts used to web scrape DR's website with python, followed by an overview of two .rmd files used to clean and visualise the data.

01 Identifying urls and extracting data - Python

The initial step involved finding and scraping the relevant webpages. The following section describes the code in the file 01_scrape.ipynb, available through the GitHub repository for this project.

requests was used to get the ballot-page, and **beautiful soup** was used to navigate the HTML structure. An inspection of the source code for the ballot-page revealed that the links to each constituency was of a class named "AccordionGrid_link__cGkec" and was thus a useful indicator to find the url for each constituency. The module **urllib.parse** provided the final and clean list of url-keys.

When the url of all 92 constituencies was acquired and cleaned, it was necessary to write a function that could collect the url-key of each candidate in every party in a constituency. The process of the function is as follows:

- Request the page for the constituency
- Find the JSON with **json** by looking for " *id='__NEXT_DATA__'* "
- Find each party in the constituency by going to:
  ['props']['pageprops']['smallconstituencycandidatesbypartycode'] in the json file
- Loop through each party in the constituency and save the url-key for each candidate

This function was then used in a loop that went through all 92 constituencies, resulting in a list with the url-key for each candidate.

Finally, the actual test data and demographic information on each candidate could be scraped. As in the function, the JSON for each candidate was found by looking for " *id='__NEXT_DATA__'* " and then saved. All JSON files are available at data/kandidater in the GitHub. Although dr.dk is a large website and unlikely to come under pressure due to my requests, **random** and **time** was used to add a random delay after each iteration of the loop.

pathlib was used to ease the process of working with paths and files and **tqdm** provided loading bars when relevant.

02 Extracting data from JSON files - Python

After JSON files on every candidate was acquired, it was necessary to investigate the JSON files to find the relevant data for extraction. Demographic information such as name, gender and profession were available alongside the candidates answers and any notes to specific questions. The process of looping through all candidate files and saving the relevant information can be seen in the file 02_JSON_to_df.ipynb on GitHub.

03 Identifying elected candidates - Python

It was necessary to scrape additional data from dr.dk to identify the elected candidates. The website linked previously contained only the elected candidates, as well as the votes they received in the election. The process of acquiring the data was very similar to the previous steps; download the website, loop over all candidates, save the relevant information from the JSON file. The full code is available in the file 03_elected_candidates.ipynb on GitHub.

04 Cleaning data - R

R was used in the two final documents. With several datasets, one with demographic information on the candidates, one with all answers and notes, and one with the elected candidates, some cleaning was required. Additionally, some processing was needed to add extra variables. The extra variables added to the final dataset were whether the candidate participated in the test, how many questions they answered, how many notes they wrote, the average length of their notes and the conviction in their answers. Conviction denotes in percentage how often a candidate answered 1: "completely disagree" or 5: "completely agree". Conviction is calculated as follows: $conviction = \frac{sum\ of\ answers\ with\ 1\ or\ 5}{number\ of\ questions\ answered} *$ 100. All code is available in the file 04_cleaning.Rmd on GitHub. **Tidyverse** was used throughout the cleaning process and **pacman** was used to manage packages used with R. The metadata on the file data/fv_22_kandidat_test.csv is available in the appendix in table "File metadata". This file is one of the final products of the web-scraping and the foundation of

most of the subsequent visualisations and analysis. The two other files are data/answers.csv and data/notes.csv which are also detailed in the metadata table.

05 Visualisation – R

Finally, the visualisation were completed using mainly the package ggplot2 which is a part of **tidyverse**. **Ggridges** was used with ggplot2 to create stacked density plots. The entire code for the visualisation is available at 05_visualising.Rmd and a knitted html version of the same file is available at 05_visualising.html. This file contains additional plots than the ones picked out for the result section.

Empirical Results

In the following section I will showcase 4 plots and a simple t-test, and provide my thoughts on how I designed them, as well as a short analysis of their content.

Candidates' notes and their election status
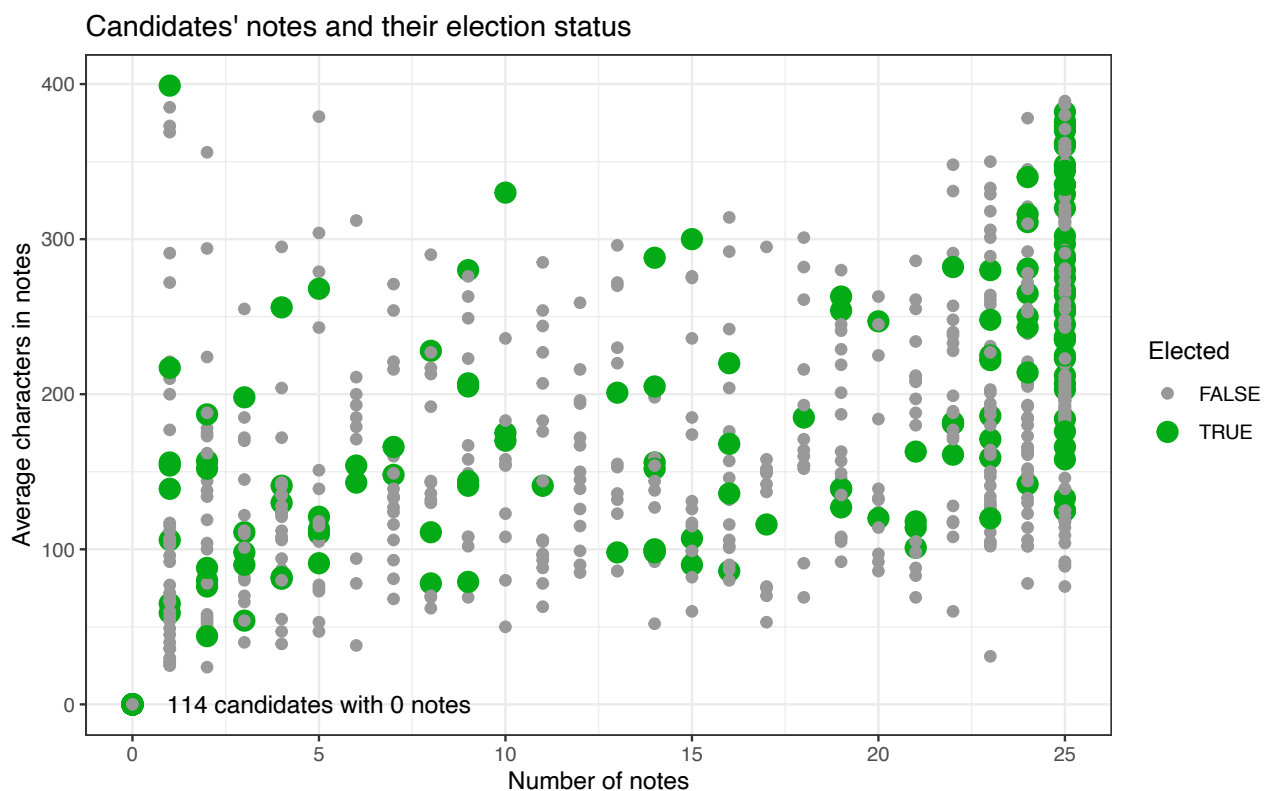
First, to investigate how candidates left notes on questions and how this might affect whether they got elected, I made the plot above. Each dot represents a candidate, and the colour and size identify whether they got elected. As only 19% of the candidates that participated in the test were elected, the plot looks a bit cluttered without the special focus on the elected candidates. By giving them a bright colour and increasing their size, it is easier to notice any patterns between notes and election status – if there were any. The plot has the disadvantage that it is very difficult to understand from the plot alone how many candidates wrote 0 notes. While I have attempted to add the information on the plot, the plot below gives a better intuitive understanding of the dispersion:

Number of notes from all candidates
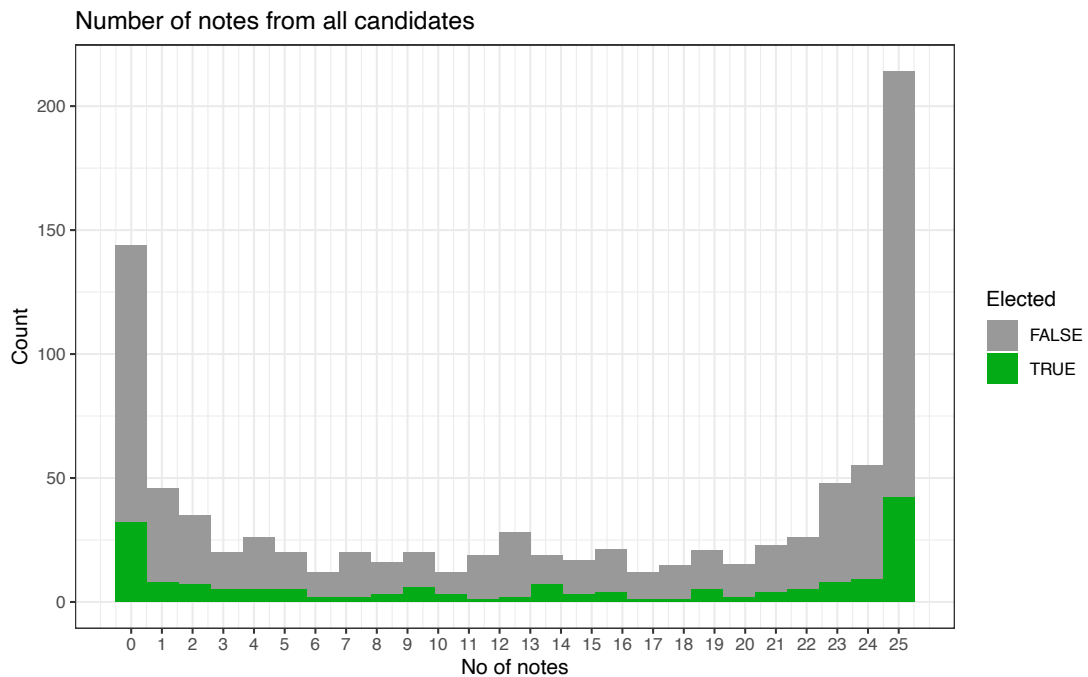
Here, it is easy to see a tendency to answer all questions, or none at all. Furthermore, this tendency does not seem to be dependent on or predictive of the election status.

Secondly, to investigate how candite's conviction might affect their election status I made the plot below:

Density plot of candidates' conviction
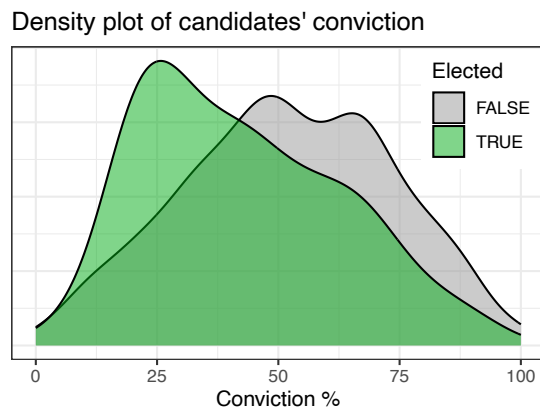
Here, a clear tendency for candidates with a lower conviction to have been elected is evident. To check whether this is statistically significant, it is possible to do a t-test. Of the 904 candidates that participated in the test, the 172 elected candidates ($M = 42, SD = 20.7$) compared to the 732 unelected candidates ($M = 52, SD = 21.7$), demonstrated significantly lower conviction, $t(267) = -5.4, p < .0001$. This result demands thorough discussion based on sound theory, as many factors go in to shaping this result. Demarcation between causation and correlation is already difficult, but without any theoretical model, it is a completely fruitless endeavor. A full reflection on the results are beyond the scope of this paper, but I will just mention: the data are not perfectly normally distributed, which weakens the robustness of the results. Furthermore, party affiliation most likely plays a major role in the candidates' conviction and their odds of election, which

is not taken into account in this plot and t-test. Below I have attempted to visualize how party and conviction might be correlated:

Density plot of parties' conviction
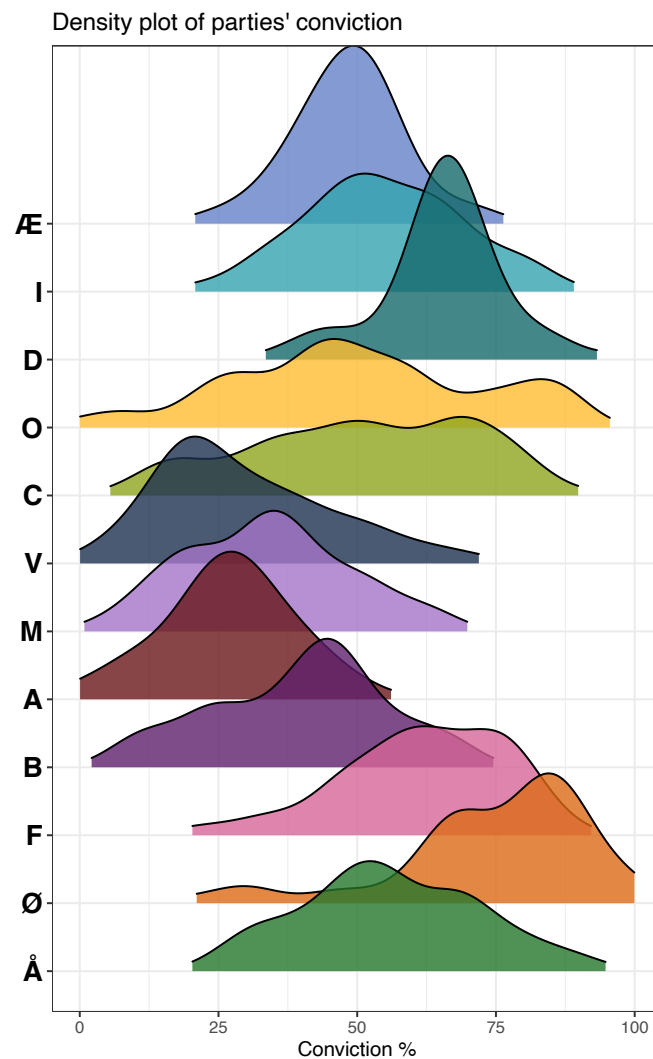
If you are familiar with Danish politics, you will recognize the official letter of the parties, as well as the color most often associated with them. It is notoriously difficult to place political parties along a one-dimensional axis, and since I am not a political science major, I have chosen to order the parties as DR does on their website, roughly with most right-leaning parties at the top and left leaning parties on the bottom. Interestingly the three parties in the government formed by the 2022 election, V, M, and A, all have a relatively low mean conviction. This ties neatly in with the fact that they define themselves as a "middle-government" as most of their candidates quite literally answer the candidate test with the middle answer options (Bugge & Pedersen, 2022). The plot was made with ggridges and I can recommend Data Novia's tutorial on it (Alboukadel, 2022).

Critical evaluation

Importantly, this exploratory analysis makes no absolute claims, and a thorough critical evaluation is not necessary. However, in terms of how representative the data is, I can confidently say that it is 89% representative, as 904 of the 1014 candidates participated in the candidate test, and only 3 elected candidates did not participate. Whether the test is representative of the candidates' actual beliefs and indeed of the policies they might vote for later, is another matter entirely.

In a larger study using this data, it might be interesting to do natural language processing analysis on the notes written by the candidates. This could be compared to candidates tweets on similar subjects, or perhaps the party's political manifesto. It is also possible to compare this data to the next candidate test and see how elected candidates might change their answering style after 4 years in the Danish Folketing.

I set a personal goal to improve my Python skills, as I have very little experience with programming scripts from scratch. The process was at times frustrating and took longer than anticipated. But I am satisfied with the final product, and I have learned a great deal in the process.

Conclusions

As no hypothesis was stated initially, no clear and final conclusions are drawn in this paper. But it is possible to sum up this exploratory analysis with the following points:

- It is possible to acquire data on the candidate test from DR and produce excellent plots
- No indication of correlation between whether a candidate was elected, and the number of notes and average length of notes they produced, was observed
- There might be a tendency for candidates to write 0 or 25 notes. Only a few candidates write between 1 and 24 notes
- A significant difference in the distribution of conviction between elected and unelected candidates was found. These results are briefly discussed in the previous section.
- It seems that candidates from parties placed in the political "middle" have less conviction in their answers than candidates from the political wings

Acknowledgements

References

GitHub repository for project:

Hansen, A. E. (2023). 2022 candidate-test in Denmark (Version 1.0.0) [Computer software]. https://github.com/AddiH/CulturalDataScienceEXAM

Alboukadel. (2022). Elegant Visualization of Density Distribution in R Using Ridgeline. *Data Novia*. https://www.datanovia.com/en/blog/elegant-visualization-of-density-distribution-in-r-using-ridgeline/

Altinget. (2013). Hvad er Altinget? *Altinget*. https://www.altinget.dk/artikel/altingetdks-formaal-maalgruppe/

Bugge, M., & Pedersen, M. L. (2022). S og V skal i regering sammen for første gang i over 40 år. *Danmarks Radio*. https://www.dr.dk/nyheder/politik/s-og-v-skal-i-regering-sammen-foerste-gang-i-over-40-aar

Danmarks Statistik. (2022). Opgørelse af folketingsvalget den 1. November 2022. *Danmarks Statistik*. https://www.dst.dk/valg/Valg1968094/other/OpgorelseFolketingsvalg2022_v2.pdf

ECPR Research Network. (2022). Voting Advice Applications ECPR Research Network. *VAA Research Network*. http://vaa-research.net/?page_id=24

Garde Gräs, M. (2022). Demokratiets festdag nærmer sig! Men Raul og en halv million andre må ikke være med. *Zetland*. https://www.zetland.dk/historie/sevgQkQN-moB3gk6P-640c2

Nielsen, J. (2021). Sådan har vi lavet kandidattesten. *Altinget*. https://www.altinget.dk/artikel/saadan-har-vi-lavet-kandidattesten

Appendices

Software metadata

Nr	Software metadata description	Details
S1	Current software version	*RStudio (2022.12.0+353), Jupyter lab (3.5.0), R (4.1.2) and Python (3.9.5).*
S2	Permanent link to Github repository where you put your script or R project	*https://github.com/AddiH/CulturalDataScienceEXAM*
S3	Legal Software License	*MIT License*
S4	Computing platform / Operating System	*MacBook Pro (M1, 2020) running macOS Monterey version 12.5*
S5	Installation requirements & dependencies for software not used in class	*Most of the software used in this paper was also used in class. Some modules in Python were not presented in class, see module table below.*
S6	If available Link to software documentation for special software	*Example http://mozart.github.io/documentation/*
S6	Support email for questions	*202006712@post.au.dk or astrid.elmann@gmail.com*

Citations for software, modules, and packages

Name	Citation	Version
R	*R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.*	4.1.2
Rstudio	*RStudio Team (2020). RStudio: Integrated Development for R. RStudio, PBC, Boston, MA URL http://www.rstudio.com/*	2022.12.0+353
Tidyverse	*Wickham et al., (2019). Welcome to the tidyverse. Journal of Open Source Software, 4(43), 1686, https://doi.org/10.21105/joss.01686*	1.3.1
ggridges	*Claus O. Wilke (2022). ggridges: Ridgeline Plots in 'ggplot2'. R package version 0.5.4. https://CRAN.R-project.org/package=ggridges*	0.5.4
pacman	*Rinker TW, Kurkiewicz D (2018). pacman: Package Management for R. version 0.5.0, http://github.com/trinker/pacman.*	0.5.1
Python	*Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.*	3.9.5
Jupyter Lab	*Kluyver, T., Ragan-Kelley, B., Fernando, Granger, B., Bussonnier, M., Frederic, J., … Willing, C. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. In F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas (pp. 87–90).*	3.5.0
tqdm	*Casper da Costa-Luis, Stephen Karl Larroque, Kyle Altendorf, Hadrien Mary, richardsheridan, Mikhail Korobov, Noam Raphael, Ivan Ivanov, Marcel Bargull, Nishant Rodrigues, Guangshuo Chen, Antony Lee, Charles Newey, CrazyPython, JC, Martin Zugnoni, Matthew D. Pagel, mjstevens777, Mikhail Dektyarev, … Max Nordlund. (2022). tqdm: A fast, Extensible Progress Bar for Python and CLI (v4.64.1). Zenodo. https://doi.org/10.5281/zenodo.7046742*	4.64.1
pandas	*McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).*	1.5.1
bs4	*Richardson, L. (2007). Beautiful soup documentation. April.*	4.8.1
The following modules are built into python, and their version follows python		
random		
urllib.parse		
time	*Van Rossum, G. (2021). The Python Library Reference, release 3.9.5. Python Software Foundation.*	
requests		
pathlib		
json		

The result of this paper is two files, detailed below. While more files are available in the GitHub, they are "raw" and uncleaned, used in intermediate steps and as they are irrelevant to the final product, they are not described in detail here.

Variable	Details
data/fv_22_kandidat_test.csv	This file contains demographic information about all candidate to the Danish X election in 2022. Additional details on their participation in DR and Altinget's candidate test is also available.
ID	Uniqe ID used for each candidate, also the url-key to dr.dk.
first_name	First name of candidate
last_name	Last name of candidate
gender	Candidate's gender, M = male, F = female
birth	Candidate's birthday in ISO 8601 time format
education	The highest level of education completed by the candidate
elected	TRUE = candidate was elected, FALSE = candidate was not elected
votes	Amount of votes each candidate received. If the candidate was not elected, no details on votes are available
participated	TRUE = candidate participated in the test, FALSE = candidate did not participate in the test
no_answered	Number of questions on the test answered
conviction	The percentage of questions where the candidate answered "completely agree" (4) or "completely disagree" (1)
no_notes	The amount of votes the candidate wrote
avg_note_length	The average length in characters of candidates notes
data/answers.csv	Contains candidates answers to DR and Altinget's candidate test on the Danish X election in 2022.
ID	Uniqe ID used for each candidate, also the url-key to dr.dk.
answer_i	The number indicates the question number. The possible answers to each question are: "skip the question" (0) "completely disagree" (1) "disagree" (2) "agree" (3) "completely agree" (4) or
data/notes.csv	Contains candidates notes/comments to answers to DR and Altinget's candidate test on the Danish X election in 2022.
ID	Uniqe ID used for each candidate, also the url-key to dr.dk.
info_i	The number indicates the question number. NAs indicate where candidates made no comment to the question.