Statiswiss

Process Book

2024-05-31

Nicolas Ettlin, Fabrice Nemo, Mina Tang

EPFL

I. Overview

One of the main features of Swiss democracy is *the ability to vote* on a wide range of issues through referenda and initiatives. In Switzerland, the citizens have a *direct say* in the laws implemented in our country. The votes are divided into three categories:

1. Optional referenda, where citizens can challenge a law passed by parlament
2. Mandatory referenda, which usually happen when the Parliament wants to modify the Consitution
3. Initiatives, through which Swiss citizens may request a change in the constitution.

The core concept of our project is to visualize the results of Swiss federal votes dating back 40 years. We want to show the results on a federal and cantonal level, as well as the recommendations of the different political parties and their strength in each canton. For anyone curious about Swiss democracy, this would be a fun way to learn more about the results of every vote.

II. Starting Out

II.1. The idea

The first step of our project was choosing what our visualization was about. We knew that we wanted to have our project be about politics, so after some discussion, we settled on a website that would *visualize the results of federal votes* in Switzerland. Since Swiss citizens vote every few months on a wide variety of topics, we thought it would be useful to have a website that compiles all the results and visualizes them in an easy to understand manner.

II.2. The data

An essential aspect of our project was of course finding the relevant data. At first, we used `opendata.swiss` to gather the relevant data (vote name, vote results, party recommendations, federal election results). While doing research regarding the implementation of our website, we found the website `swissvotes.ch` , which contained a lot of the relevant information we were looking for. The data was already preprocessed, and contained links to the voting brochures as well as information about the main themes of each vote. As such, the datasets we ended up using came from both `opendata.swiss` and `swissvotes.ch` .

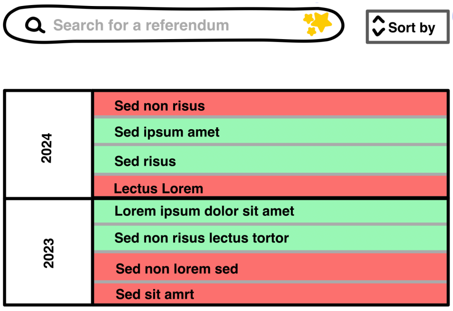
III. Designing the visualization

III.1. The concepts

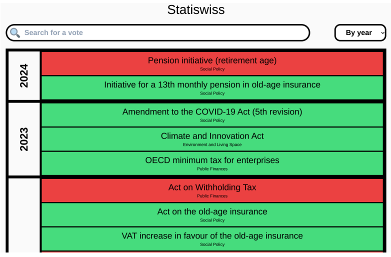
Once we had a general idea of what we wanted to do, we sketched out the core design of the website in our sketchbook. There were three fundemental concepts for our website that we wanted to implement:

Search feature

We needed a *search feature* to be able to find a given vote. Since there are over 300 votes, this feature is absolutely necessary to be able to easily find information on the website.



The idea



Statiswiss

Search for a vote

By year

2024
- Pension initiative (retirement age) — Social Policy
- Initiative for a 13th monthly pension in old-age insurance — Social Policy

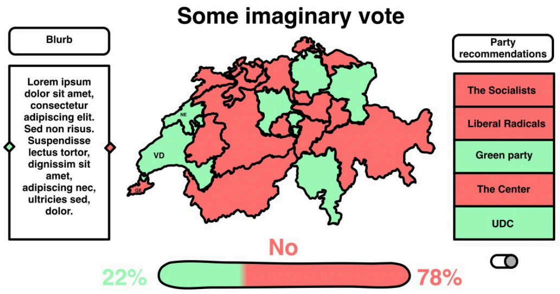
2023
- Amendment to the COVID-19 Act (5th revision) — Social Policy
- Climate and Innovation Act — Environment and Living Space
- OECD minimum tax for enterprises — Public Finances

- Act on Withholding Tax — Public Finances
- Act on the old-age insurance — Social Policy
- VAT increase in favour of the old-age insurance — Social Policy

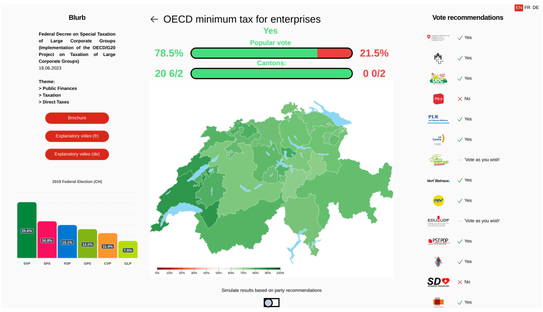
The execution

Showing the results

We needed a way to show the results: The most intuitive way is of course through a *map*. As such, we wanted a map that showed the election results by canton/commune. We also wanted a way to see the total result at a glance: we chose a *progress bar*.


Some imaginary vote
Blurb
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor.
NE
VD
Party recommendations
The Socialists
Liberal Radicals
Green party
The Center
UDC
No
22%
78%

The idea



Federal Decree on Special Taxation of Large Corporate Groups (Implementation of the OECD/G20 Project on Taxation of Large Corporate Groups)
18.06.2023

Theme:
> Public Finances
> Taxation
> Direct Taxes

Brochure

Explanatory video (fr)

Explanatory video (de)

← OECD minimum tax for enterprises

Yes

Popular vote

78.5% 21.5%

Cantons:

20 6/2 0 0/2

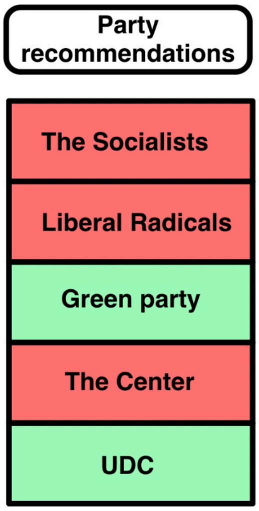
Simulate results based on party recommendations

✓ Yes
✓ Yes
✓ Yes
✕ No
✓ Yes
✓ Yes
'Vote as you wish'
✓ Yes
✓ Yes
'Vote as you wish'
✓ Yes
✓ Yes
✕ No
✓ Yes

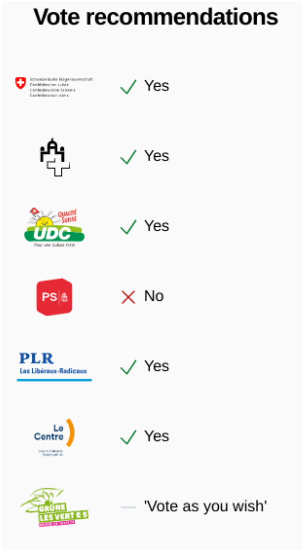
The execution

Influence of political parties

Finally, we wanted to show the ways the different political parties influenced the vote. There are two aspects: first, each party's *recommended vote*; second, the difference between the theoretical results (if the votes went according to party recommendations) and the real results.



The idea



Vote recommendations

✓ Yes

✓ Yes

UDC ✓ Yes

PS ✕ No

PLR Les Libéraux-Radicaux ✓ Yes

Le Centre ✓ Yes

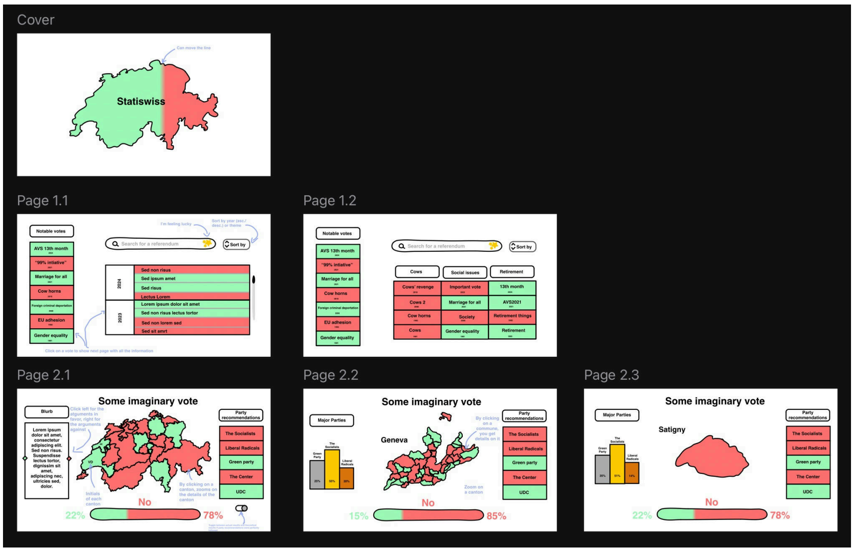
GRÜNE LES VERT-E-S — 'Vote as you wish'

The execution

III.2. The sketchbook

Our original Sketchbook can be seen in Fig. 4. The initial Sketchbook captures the idea of our project, and shows well the different aspects we wanted to implement. However, the overall design was a bit outdated, so we updated it slightly for the final website.

Fig. 4 — The sketchbook



IV. Implementing the website

Now that we knew what we wanted to do, it was time to implement it. The first prototype of the website did not have many interactable elements, it was more like a realisation of our sketchbook. Once that was done, we needed to make it truely interactable.

IV.1. The map

We created a custom map of Switzerland using D3.js. We had to implement custom behavior for the map, so that when a canton is clicked the map zooms on it and the user can see the detailed results for each municipality. Close to the map, the user can also find a bar graph showing the results of the last federal elections. This chart initially shows the countrywide results, and is updated when the user focuses on a canton to show the results in this canton.

IV.2. The search function

We implemented a function that allows the user to search a votation through semantic search, rather than a simple textual search. This is done using embeddings, a natural language processing technique which converts text to a vector in a way such that strings that are semantically similar are converted to vectors that are close. We converted every vote to such a vector, and we embed the search query of the user when they type something, in order to perform a vector search of the search query.

IV.3. The results and recommendations

Second, we needed to implement everything related to the actual data. This meant going through `swissvotes.ch` dataset and determining what the information represented was (as the information was originally in German). After this, we converted all our data to a `.json` file using a scripted coded with the `pandas` library in Python. We still needed the information from the federal election results as well as the results of votes by comune. This data was taken from `opendata.swiss` and converted to `.json` . For some of the votes, there were no english translations, so we used ChatGPT. Furtherermore, the breakdown per canton was not available for the latest votes on `swissvotes.ch` , so we used a python script to extract them from our `opendata.swiss` dataset. Once the data was in the desired format, we used plain JavaScript to extract the relevant data from the json files.

There is one json file (`all.json`) that contains basic information about votes (id, title, date, themes, results at the federal and canton level), this file is used to show general information about each vote, generate the list of votes on the index page of the website. Plain JavaScript is used to extract the information and generate HTML to show it to the user. This information is also used to compute the theoretical results of votes, had voters accurately followed the recommendations by parties they voted for.

Then, there are also 370 json files for detailed results of each vote. They contain results at the cantonal and communal level and are used to show the results on the map.

IV.4. The theoretical results

For the theoretical results, we implemented a similar process as for the real results. However, this time, instead of taking the data of the results, we computed the recommendations of the parties, weighted by the amount of seats they have in parliament. For each canton, we weighted the number by the total population so that the results on the federal level are accurate. A togle is used to switch from the theoretical to

IV.5. The tools

- We used Python to extract, clean, and format the data in the way we desired.

Tailwind CSS was used to style the website.

We used html/css/jss for the implementation of the website itself.

D3.js was used to implement the interactive map used for our website.

IV.6. The improvements

What we did improve

We made our website multilingual, providing not only English, but also the main languages of Switzerland, German and French. We also put links to the official brochure as well as explanatory videos in French and German from `easyvote` when available.

What could be done in the future

We could adjust the layout so it works on different sized screens, and also tweak the design.

V. Facing challenges

Data challenges

In the dataset taken from `swissvotes.ch` , there were thousands of different columns, and the documentation was in German, so we had to go through it by hand to write down what was what and whether it was important. Furthermore, the data had many gaps, that we had to make up for using our other data sets from `opendata.swiss` , as well as ChatGPT for translation.

Map challenges

Merging the various datasets we are using for the map visualization proved to be challenging. In some cases, the names of the votes in the dataset with the precise results for each canton/ municipality did not match the name of the votes in the main dataset. To fix this, we compared votes during the same day using a computation of the longest common substring between the two names.

VI. Peer assessment

VI.1. Nicolas

Nicolas was the one who implemented the interactive map as well as the search engine. He was also in charge of creating the screencast video and ensured the final design of the website.

VI.2. Fabrice

Fabrice was in charge of making the initial skeleton of the website and integrating the data that was not related to the maps (list of votes, information about votes, election charts, party recommendations), and tooltips that show local results on the map. He also translated the interface of the website into French and German and gathered the party logos and names. He wrote the code for computing the simulated results based on party recommendations.

VI.3. Mina

Mina processed the data into a suitable format and completed missing information from the main dataset. She was also in charge of the base design of the website, as well as writing the READMEs and process book.

VII. Useful links:

Our website: Statiswiss
Our repository: Github
Datasets used: Swissvotes, OpenData (elections), OpenData (commune geometry), OpenData (commune results),