A Visualization of Swedish Election Data

Torsten Gustafsson

Abstract— Visualization of election data is useful for many things. This report describes an application that visualizes the voting data of Sweden for the years 2002-2014. It was developed using the Javascript library D3.js.

Index Terms—Visualization, Election data, Sweden.

✦

1 INTRODUCTION

Swedish elections contains a lot of data in the form of votes. These votes may be filtered into the party they are for, and where the vote took place. This report describes an application that visualizes this data, using the vote data concerning each municipality and the vote percentage for each party in each of these municipalities.

2 BACKGROUND AND RELATED WORK

There exist several visualizations of the Swedish elections, and the data can be visualized in many different ways. An example is the visualization found in [2], which shows how the vote percentages have changed for each party during the years since 1911. Such an example does not focus on where the votes took place though, instead it takes the country as a whole into consideration. While this is interesting in and of itself, one may want to get more details of for example in which municipalities the votes were made and how many people lives there compared to the municipalities vote percentages for each party.

This application visualizes the geographical locations of each municipality, as well as their vote percentages. This means that, unlike the example where all the voting data for Sweden was visualized, it focuses on the scale of municipalities to visualize data. It also show detailed statistics of the municipalities, that might be interesting when looking for correlations in the data.

3 DATA

The data contains information of the Swedish elections over the time-period 2002-2014.

The application uses seven different datasets. Four of them contains the election data, while the other three contains more detailed statistical information about each municipality. The different datasets are listed in Table 1. They are all taken from "Statistiska Centalbyrån" or "SCB" [1].

Table 1: The datasets used

Data	Description
Swedish_Election_2002.csv	Election data for the year 2002. Contains 3202 lines, with municipality, party, and the percentage as its variables.
Swedish_Election_2006.csv	Same as above but for 2006.
Swedish_Election_2010.csv	Same as above but for 2010.
Swedish_Election_2014.csv	Same as above but for 2014.
population.csv	Contains data of how many people lives in each municipality for each (relevant) year, between 25-64 years old.
education.csv	Contains data of how many people that are highly educated in each municipality, between 25-64 years old.
income.csv	Contains data of the average income for each municipality, between 25-64 years old.

- *Torsten Gustafsson is a student at Linköping University, Sweden, e-mail: torgu529@student.liu.se.*

4 METHOD

An application was developed to achieve this kind of visualization. By running the application from a web-browser, the user may select which election year to visualize, and choose between a couple of different views. One of these views will show a map of Swedens municipalities, colored according to the majority party for each municipality using a fixed color scheme for the parties. The other view will let the user select which party to visualize, and color all the municipalities with that partys color, but with a higher color intensity for those with a high percentage for that party.

The map have a color legend that show different information depending on which view is currently active, as shown in figure 1 and 2.
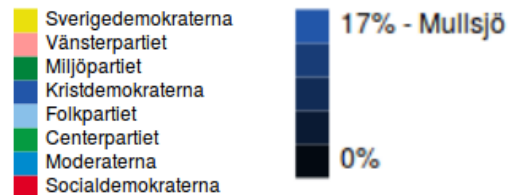

Fig. 1: Legend used when the show majority view is active. Shows what the different colors represents.

Fig. 2: Legend used when the filter by party view is active. Shows how the color grade scales from lowest- to highest intensity.

It was decided that the application should be written in Swedish since it will most likely mainly attract a Swedish audience.

5 IMPLEMENTATION

The application was developed for a web-browser, specifically Firefox (the other browsers does not have full support for all of its features). Its main focus is to visualize the voting data for Swedens last four elections.

This was done mostly through the javascript library D3.js, which can handle data and easily show it in the form of charts. It can also connect a map to the data if you use a .json file containing geographical data.

A big reason why D3 was used was simply because the developement team had some recent experience with it, and it felt like an obvious choice for visualizing this kind of data.

6 RESULTS

The work resulted in an application where the user may use a set of functions to visualize the voting data in different ways. Figure 3 shows how they look to the user. They include a search field where a specific municipality may be found. By dragging the first slider, the user may change which voting year to visualize. The two views are selected by choosing one of the two radiobuttons available. If the "Filter by Party" view is selected, the user may also choose a specific party to visualize. The second slider may be used as a complement to both views. All municipalities that does not fullfill the threshold chosen will be colored

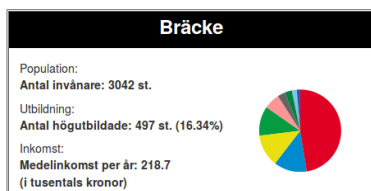Fig. 3: The different functions avaliable.

Fig. 4: The information box shows additional details of the selected municipality.

black. This may be useful when looking for the municipalities that have a clear leading party, since the other ones will be filtered out.

The user may select a municipality either by searching for it or by clicking on it. When a municipality is selected, it will show up in the detailed information-box, shown in Figure 4, where more detailed statistics about that municipality is shown.
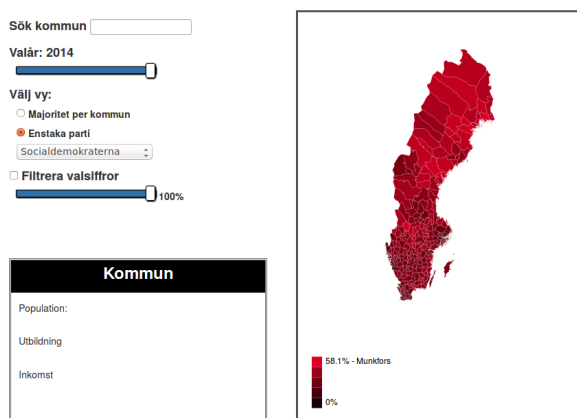

Fig. 5: The final application.

Figure 5 shows the final application. This is how it looks when it has just been started. No municipality is selected, the latest election year (2014) is shown, and the active view is the "Filter by a Specific Party" view, with Socialdemokraterna selected.

7 CONCLUSIONS AND FUTURE WORK

The application does what it is intended to do. A map of Swedens municipalities successfully visualize the voting data in a way that is simple to understand and navigate through.

The implemented functions all work as intended and the purpose of the functions are mostly very clear. Using this application, a user may find detailed information about a municipality, or get a visualization of how the votes are distributed across Sweden, in some differing, yet relevant ways.

Some things to improve is primarily the design. The application is designed with HTML and CSS. By defining a concrete plan for the design and then start implementing that design using some kind of front-end framework, the application could possibly look really good. As it is now, it is a working application with a pretty dull appearance.

Some optimization could be done as well. The search function uses a function that performs a new search every time the user lifts a key. This means that the search will happen dynamically while the user is typing, but because the search is slow, it causes some lag if the user is a fast typer. This problem can be solved by either calling the search function only when the user hits the enter key, or by improving the speed of the search itself.

There is also some problem with certain functions being called more often than necessary. For example, when the user selects a municipality by clicking it, the function responsible for drawing the detailed information-box is called twice at the moment. This could be prevented by a simple refactoring of the code.

REFERENCES

[1] Statistiska centralbyrån. `http://www.scb.se`. Accessed: 2016-03-09.
[2] Historical swedish election results. `http://ejoh.se/historical-swedish-riksdag-election-results/`. Accessed: 2016-03-09.