Visualization of Open Data

Michael Gruber*

Student

1 MOTIVATION

2 MOTIVATION

I am a big fan and proponent of open governance, open source and open data. There is a substantial amount already out there, and its becoming more every day.

2.1 The Problem

Open data is, as the name suggests, freely available for everyone. But many Institutions don't invest very much in their open data programs, especially early on. Data is made available, which is a good first step, but is often not so easily explored. Usually open data is released in an open format for which free tools exist. Occationally, it is even coupled with online-services (tough the quality of those varies greatly as well).

Very often the data's origin is from some proprietary software with its own formats, and handled in special ways. So data is simply exported from those programs. This can lead to situations where the released files don't fully adhear to the format-specification, or the data has to be preprocessed due to structural issues to make use of it. There can also be inconsistencies in data-format within one institution, eg. with multiple departments releasing data from their own tools. Even the locale-setting of the software can make things harder, eg. number and date-formats.

This makes it next to impossible to create a reasonable standard program/interface to visualize those data-sets. And by extension, makes it hard to "grab a file" and just make use of it. Some integrated solution, based on, and provided for a certain platform, using as standard as possible techniques, would be highly desirable.

2.2 Data & source

I decided to work with the data provided by "data.wien.gv.at" open data platform. The scope of my work is time-related data, eg population development. As the data within this scope is mostly available in csv-format, i chose to limit myself to only support csv as file-format.

2.3 Goal

My goal is to visually representation at least a subset of the available data. The visualization and interface should be held rather simple, so it is accessable to the "average user". But it should also be useful to more experienced users, so I want to integrate an "expert-mode" to provide the features less interesting/suitable for the laymen.

2.4 Users

2.4.1 A

Citizens that are curious about open data, and want to explore the data that is available. They might be just generally taking a look around, having heard about open data in the media. Or they might be interested in looking at official data as a source of information (maybe looking for a potentially more trustworthy source than eg. "Die Krone"[7], or wikipedia[**?**]).

2.4.2 B

For example students looking for references or statistics regarding official numbers. Category-B users are expected to have (at least limited) knowledge about statistics and higher "visual literacy"

3 IMPLEMENTATION

3.1 Brief description of how the system was implemented (toolkits, languages, platforms)

3.1.1 Overview

- Javascript[4] & Css[1]
- D3[2] as base platform
- Rickshaw[10], built on D3, for the interactive Visualization.
- jQuery[5] & extensions
  jQueryUI[6] for certain interface-features
  jQuery.parse/papa-parse[8], csv-parser-plugin)
- handsontable[3], for table-creation/interaction
- PHP[9] for serverside file-handling

Except for serverside file-handling (download, storing, loading) in PHP, everything has been implemented based on Javascript/Css and libraries based on those.

3.2 Implementation challenges

3.2.1 file-format

Originally, the intend was to use d3 and its integrated filehandling-capabilities to read in data from csv, but due to limitations it could not handle the format in which it is provided by "data.wien.gv.at". After some research, and trial-and-error, my choice fell on the papa-parse-plugin for jQuery, which seemed the most reasonable alternative. Still, some effort had to be taken to deal with the non-standard use of embedded information in the csv-tables.

3.2.2 file-content

Csv is used for many types of data on "data.wien.gv.at". Most of it does not contain data suitable for my visualization platform (due to my focus on time-related data. Also the processing of the structure proved to be challenging. Values for attributes that are in context of certain other attributes are embedded as additional rows, so the context of the dependent attribute has to be derived from their values, which turned out to be rather error-prone and convoluted to solve. Example illustrating the problem:

```
regioncode 1 sex=1 value 2010
regioncode 1 sex=2 value 2010

⋮

regioncode N sex=1 value 2010
regioncode N sex=2 value 2010
```

*e-mail: michi.gruber@gmail.com

⋮

```
regioncode 1 sex=1 value 2011
regioncode 1 sex=2 value 2011
```

In this example, “value” is actually in context of both “regioncode” and “sex”. It is not an easy question as to how to display such data (especially in an automated fashion). A simple approach would be to detect such occurences and “flatten” the data.
This would result in multiplying the number of attributes by (regioncode-count)*(sex-count). In some files, there were more than 10 regioncodes and other dependent attributes, and potentially as little as just two different years, making this approach rather unpractically.

An alternative would be to give the user the choice as to which attributes to derive from others, resulting in a much more complicated interface, and probably requires a reasonable knowledgeable or experienced person.

With only one such dependency, eg. sex and year, it is possible to create multiple plots, each for its own subset of the resulting values, but this approach makes it harder to compare values eg men vs. women.

3.3 value ranges

A common occourence is that a file contains values with vastly different ranges. A good example of that is the “Economic indicators”-dataset, containing GDP per capita (≈ 60000) and GDP groth rate ($\approx 4\%$). Having a single plot can only display a subset of attributes at a time, hiding the others.

4 Results

4.1 Scenarios

At first, a visitor lands on the Tab for data-selection.

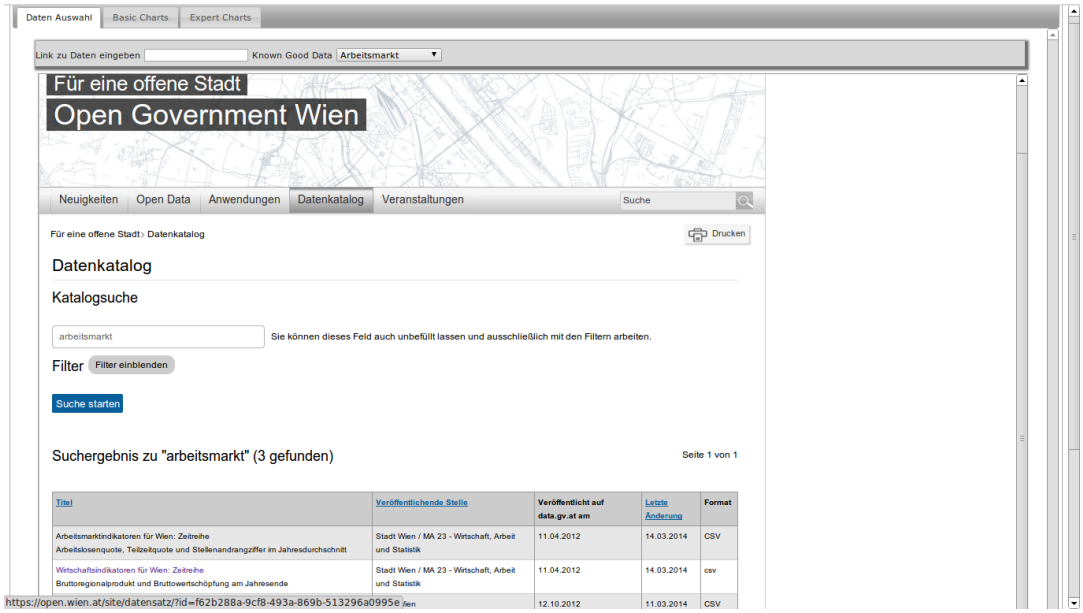

Figure 1: Data Selection Tab

The user has two choices:

1. choose from a couple of pre-selected datasets
2. browse the embedded “data.wien.gv.at”-page for interesting datasets

Option 1 is realized in form of a dropdown-box of selected datasets in the header. (Remark: The set of datasets in the handin is chosen to illustrate both good data, as well as some challanges.)
Option 2: After the user found what he is interested in, he pastes the link to it (drag&drop won’t work, would be cross-domain..) in the textfield. He will automatically land on the charts-tab.

5 References

References

[1] Css tutorial, 2014.
[2] D3 - data driven documents, 2014.
[3] Handsontable data grid editor, 2014.
[4] Javascript tutorial, 2014.
[5] jquery, 2014.
[6] jquery user interface, 2014.
[7] Kronenzeitung, 2014.
[8] Papa parse, 2014.
[9] Php tutorial, 2014.
[10] Rickshaw - interactive time series graphs, 2014.