

Media Transparency Database Visualization

188.943 Praktikum aus Visual Computing

Abstract

The “media transparency database” contains the accumulated amount of money spent by governmental organizations on media companies. This data can be explored as a multimodal dynamic network. Existing web projects already present solutions to visualize the dataset, but to analyse the data further a user needs more interaction methods. This project implements a task-tailored dashboard with multiple connected views, which implements brushing and linking to enable the user to analyse the dataset in an easy to use matter.

Related Work

The “media transparency database” is available since the third quarter of 2012. Since then several visualizations got presented:

- [\[F. Lang\]](#) presents a visualization that uses grouping of the media entities to reduce the screen space and complexity of the visualization. It only uses one quarter of one year of the total data. It is possible to interact with the visualization and ungroup the media entities.
- [\[M. Hametner\]](#) presents a static visualization with bar charts and a line plot as a visualization for time oriented data by the Austrian newspaper “Der Standard”.
- [\[M. Schrempf\]](#) implemented a force-directed node link diagram. The user of this visualization is able to interact with the data and filter it with different queries. But the force-directed node link diagram was too slow for the huge database.
- [\[P. Salhofer et. al\]](#) implemented a website to get an overview of the media dataset. It features multiple visualizations which are all interactive but not connected to one dashboard.

The first two visualizations are presentations of an analysis of the data. But the last two approaches are visualizations that support the user to analyze and investigate into the data. The force directed node link diagram has the problem that it is too slow to render a nice overview of the dynamic network. Additionally it is hard to interpret the payment flow, because it is visually encoded in the size of the nodes of the diagram.

The visualization of [\[P. Salhofer et. al\]](#) is stable, easy and fast to interact. But the visualizations are distributed onto 4 different web pages, which makes it hard to combine the insight of the user from one visualization with the others. Additionally the payment flow visualization is restricted to only 800 relations.

Technologies

The following technologies are used:

- *JavaScript*: a script language to create dynamic client side webpage.
- *Data-Driven Documents*: A JavaScript library for manipulating documents based on data.
- *jQuery*: A fast, small, and feature-rich JavaScript library.
- *Bootstrap*: A framework for developing responsive, mobile first projects on the web.
- *Brunch*: A [node.js](#) build tool to compile scripts and styles and to concatenate scripts and styles.
- *clean-css*: Is a [node.js](#) library for minifying CSS files.
- *uglify-js*: Is a [node.js](#) library for minifying JavaScript files.
- *crossfilter*: Is a JavaScript library to explore multivariate datasets with coordinated views.
- *DC*: Is a JavaScript library with native crossfilter support to create charts for multidimensional data exploration
- *Git*: Is used as version control system.

Problem Description

Governmental advertisement in media and sponsorships are a possible way to influence press opinion. Therefore, the Austrian parliament passed a law that made it mandatory for governmental organizations to disclose their expenses for advertisements in different media (TV, radio, print, as well as online).

This so-called “media transparency database” is made publicly available by the Austrian Regulatory Authority for Broadcasting and Telecommunications ([RTR](#)) via the Austrian open government data portal.

It contains the accumulated amount of money transferred in a certain quarter of the year for each governmental organization and media company. This database can be explored as a multimodal dynamic network.

Data Structure

The media transparency database is structured as a Table with each row containing a relation from one governmental organization (*Rechtsträger*) to one media company (*Medium*). This relation contains the amount of transferred money (*Euro*), the quarter of the year (*Jahr Quartal*) and the law of the reason of the payment (*Gesetz*).

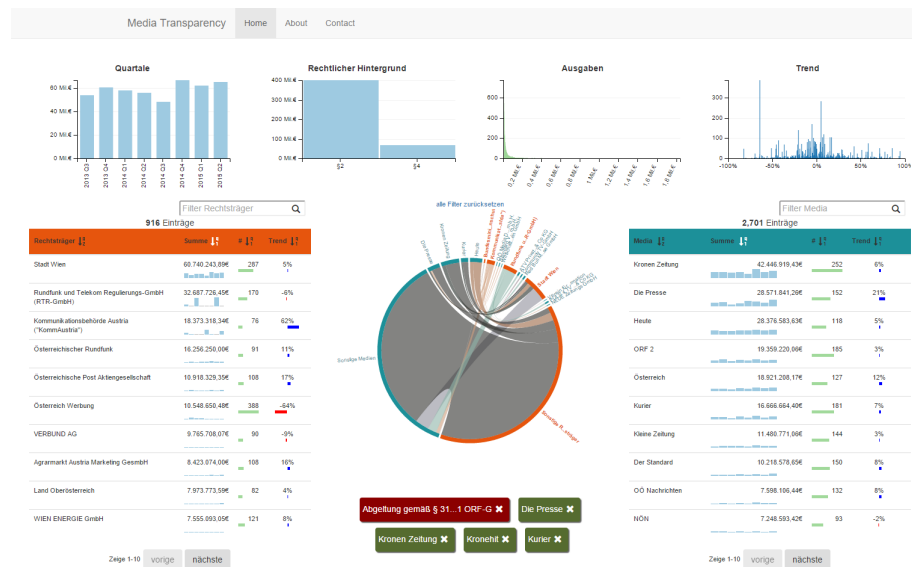
The table contains over 145000 entries over 12 quarters. So that one quarter contains 13000 data entries. There are over 1000 governmental organizations and media companies.

The data quality of the database is not sufficient enough for some data entries. These entries include spelling mistakes or are just differently formatted.

Examples

Dashboard

(Click on the image to go the the video)



Filters

Flow Visualization

Details

Source Code Documentation

Install

The release build from the last tag is in the subfolder [public](#). Copy it onto a webserver and it should work. Or visit my [server](#)

This project depends on [Brunch](#). Install it with for example [node.js](#). All depending libraries should be managed by brunch. If something is missing you have to install it manually with [node.js](#).

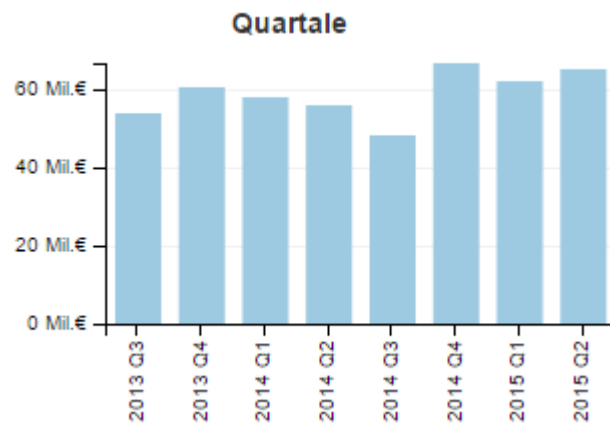


Figure 1:

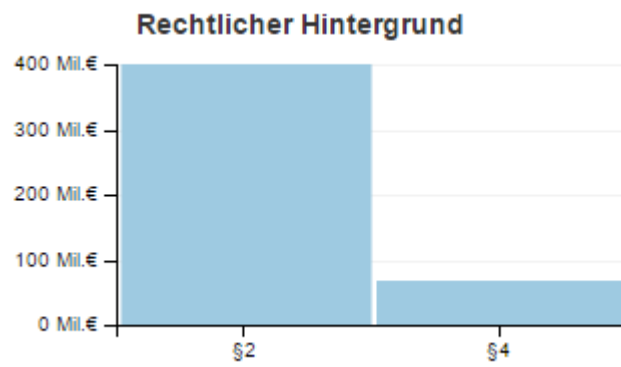


Figure 2:

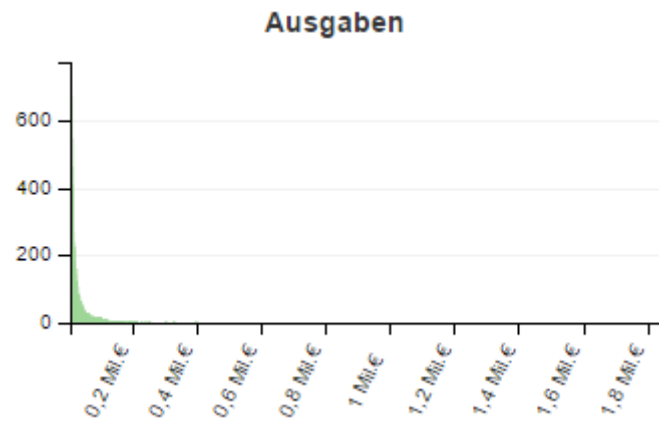


Figure 3:

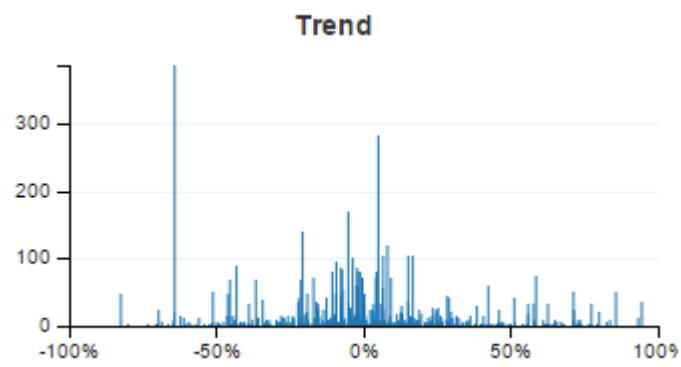


Figure 4:

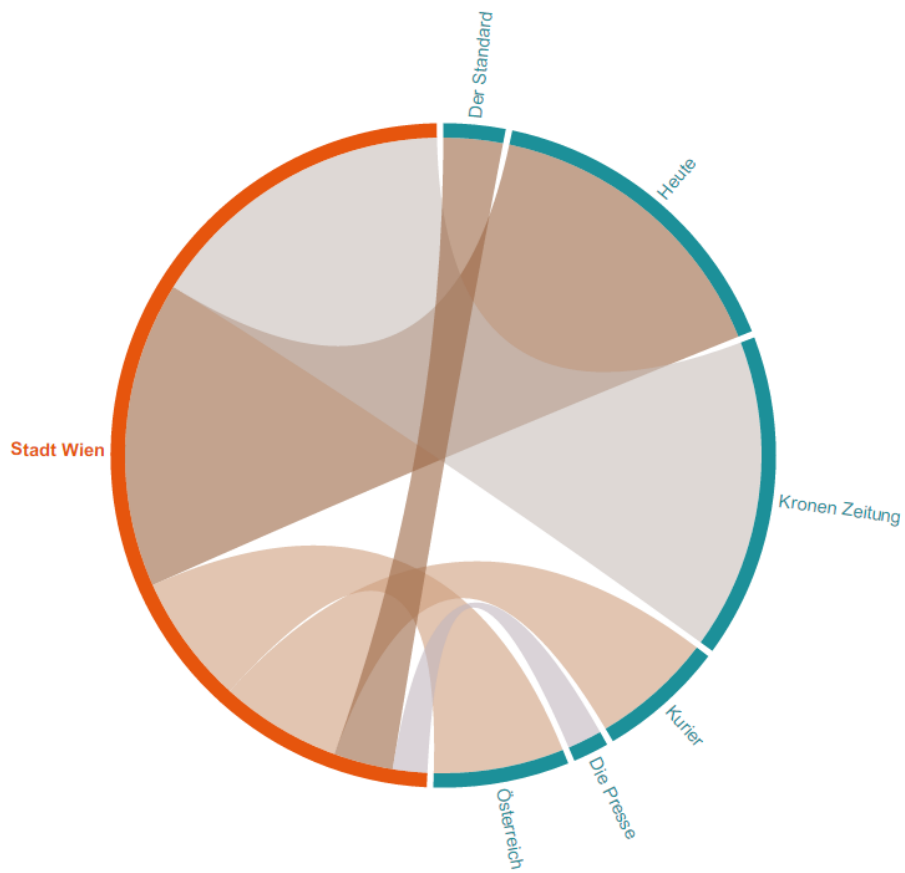


Figure 5:

2,735 Einträge

Media ↓ ²	Summe ↓ ¹	# ↓ ¹	Trend ↓ ¹
Kronen Zeitung	83.124.010,22€ 	250 	7%
Die Presse	33.530.638,89€ 	149 	9%
Kurier	31.629.341,12€ 	171 	8%
Heute	28.376.583,63€ 	118 	5%
ORF 2	19.359.220,06€ 	185 	3%
Österreich	18.921.208,17€ 	127 	12%
Abgeltung gemäß § 31 Abs. 11 ORF-G	15.000.000,00€ 	1 	-100%
Kleine Zeitung	11.480.771,06€ 	144 	3%
Der Standard	10.260.775,04€ 	150 	8%
OÖ Nachrichten	7.598.106,44€ 	132 	8%

Zeige 1-10

vorige
nächste

Figure 6:

Filter Rechtsträger

400 Einträge

Rechtsträger	Summe	#	Trend
Stadt Wien	28.105.836,84€ 	6 	5%
Österreichischer Rundfunk	8.753.679,00€ 	6 	23%
Bundeskanzleramt	3.879.237,29€ 	6 	4%
Österreichische Bundesbahnen-Holding Aktiengesellschaft	3.856.094,00€ 	6 	-19%
Wirtschaftskammer Wien	3.725.776,45€ 	6 	3%
Österreichische Post Aktiengesellschaft	3.405.237,82€ 	6 	4%
Bundesministerium für Verkehr, Innovation und Technologie	3.176.485,98€ 	6 	3%
Land Niederösterreich	3.161.014,63€ 	6 	-13%
VERBUND AG	2.605.745,25€ 	6 	-9%
WIEN ENERGIE GmbH	2.169.284,54€ 	6 	24%

Zeige 1-10

vorige

nächste

Figure 7:

1. install [node.js](#)
2. install brunch globally:

```
npm install -g brunch
```

3. install local packages (if necessary):

```
npm install --save-dev brunch javascript-brunch css-brunch sass-brunch  
queue-async
```

Run

develop standard run

```
brunch build
```

develop run with server

```
brunch watch --server
```

The standard port for the brunch server is 3333 (`localhost:3333`)

release build

```
brunch build --production
```

Documentation of global variables

The source code documentation of global variables is in a separate file: [application.js](#) I created a documentation file with [jsdoc](#) it is in [documentation/jsdoc](#) or online on my server: <https://pfahler.at/mtdb2/doc/>

Structure

The source code of this project is in the subfolder [app](#) this folder is structured as follow:

- [assets](#) the assets files that the project depends on. For example the `index.html`, the csv data, the used fonts, ...
- [styles](#) the style files of the project. Currently `less` is used to translate the style files to a single css file. The filenames of the single files have to start with an `_` to be ignored by the `brunch` compiler. And in the file [structure](#) the single files are linked together.
- [views](#) This folder contains the files that describe the dom structure of the view.
- [about.jade](#) contains a documentation of the project in markdown code

- `charts.jade` contains the overall structure of the dashboard
- `chordChart.jade` contains the structure of the chord chart: the tooltip and the chart itself.
- `contact.jade` contains the contact information of the project
- `filters.jade` contains the structure of the filters of the dashboard. For example the quartals bar chart, the legal background bar chart, ...
- `nav.jade` contains the structure of the fixed navbar
- `table.jade` contains the structure of the two table widgets (legal and media entities), their count widgets and includes the `chordChart.jade` between them.
- `application.js` contains the definition and documentation of all global variables.
- `charts.js` initializes the application.
- `chordChart.js` initializes the chord chart and implements the method for updating it.
- `data.js` is responsible for initializing the data.
 - 1) if a local storage is available it is used.
 - 2) also the combined objects from a previous session is restored.
- `dataparse.js` parses the media transparence database data.
- `filterCharts.js` methods for the initialization of the 4 filter charts.
- `globalFunction.js` initializes the global functions of the application
- `listeners.js` initializes the listeners for tooltips or the table.
- `dimensions.js` initializes the dimensions and the grouping of the dimensions.
- `matrixFactory.js` creates the matrix for the chord chart.
- `tables.js` initializes the tables and the functions for updating the tables