



# Übung Open Data: Informationen zu den Übungen, App-Demos & Einführung in Tools

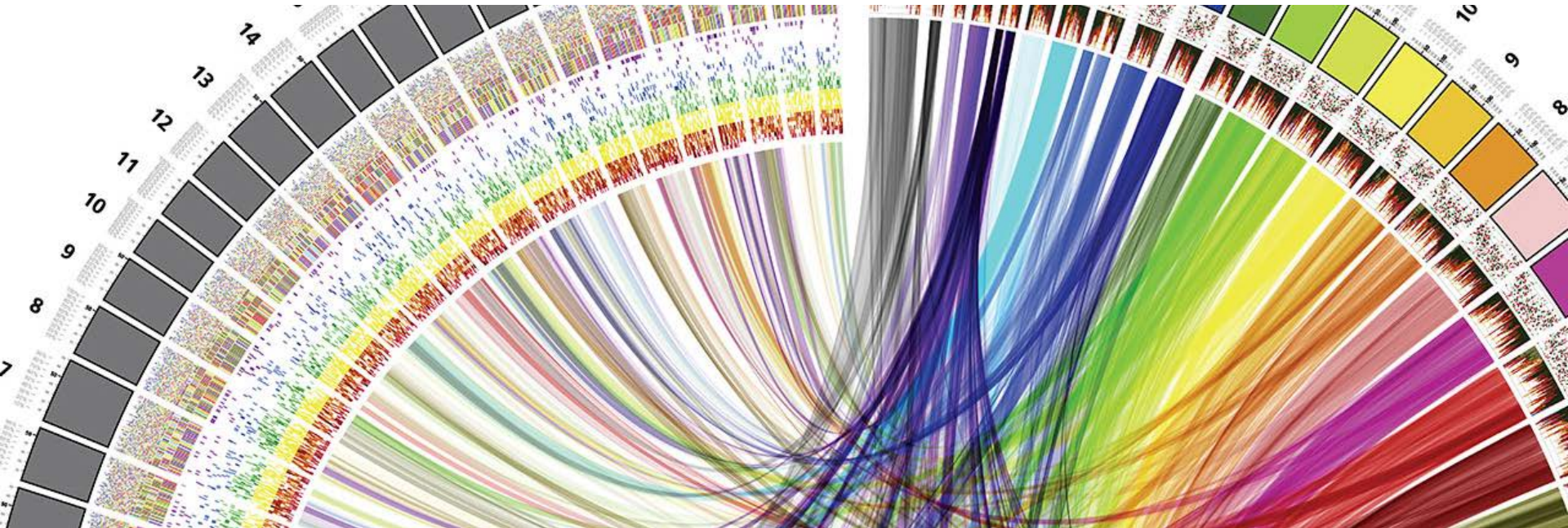
**Termin 1, 25. Februar 2016**

Dr. Matthias Stürmer und Prof. Dr. Thomas Myrach

Universität Bern, Institut für Wirtschaftsinformatik  
Abteilung Informationsmanagement  
Forschungsstelle Digitale Nachhaltigkeit

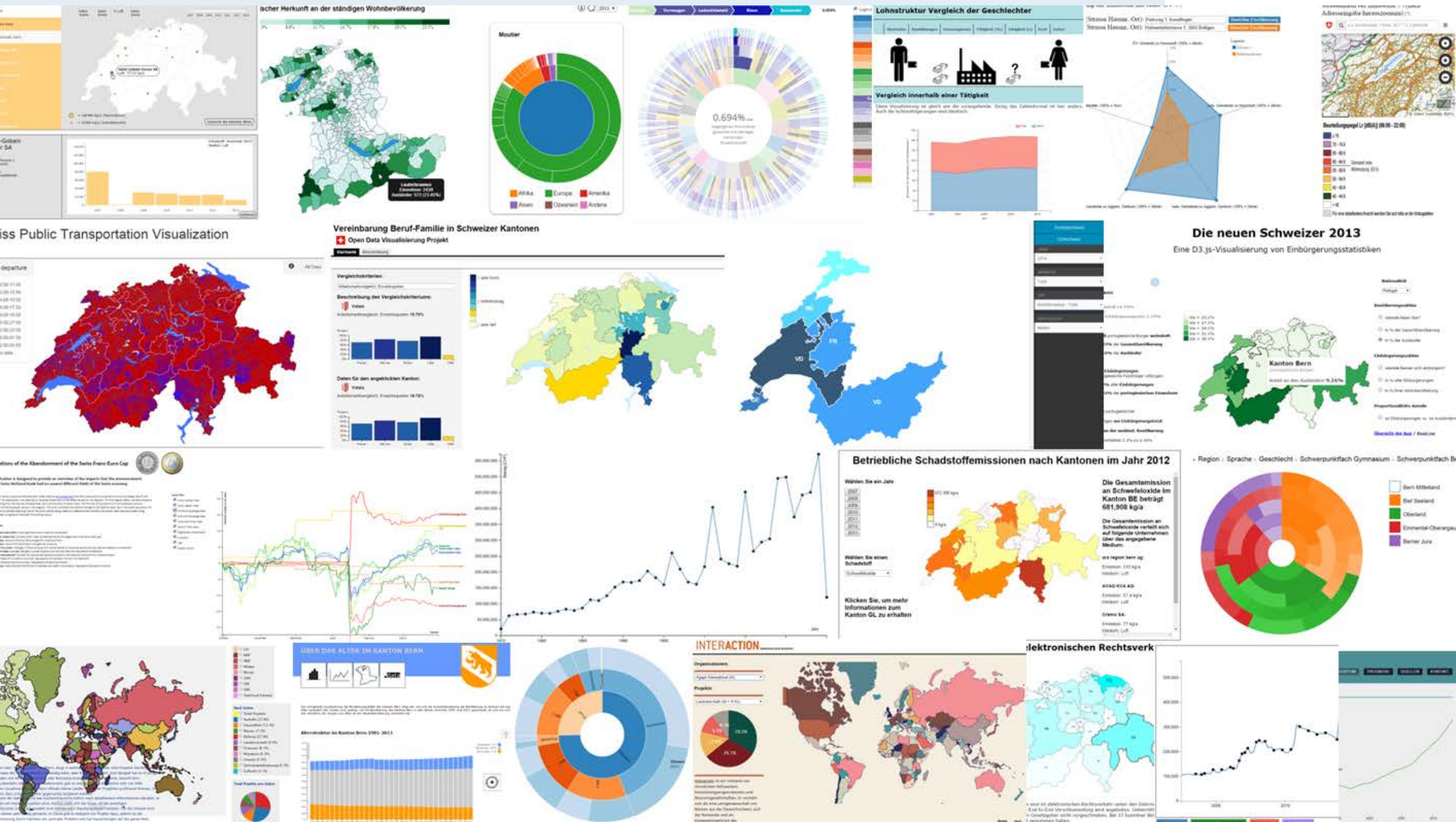
# Agenda

1. Organisatorisches zur App-Entwicklung
2. Interaktive Visualisierungen
3. Einführung Entwickler-Tools
4. App-Demo von zwei ehemaligen Studierenden
5. Fragen und Antworten





# Ziel: Entwicklung einer Open Data App



# Terminübersicht Übungen

- > 25.02.2016: Informationen zu den Übungen, App-Demos & Einführung in Tools
- > 03.03.2016: Einführung Web-Programmierung
- > **10.03.2016: Open Data Speed Dating**
- > 17.03.2016: Einführung D3.js & Daten einbinden in D3.js
- > 24.03.2016: Anpassen von bestehenden Apps & Bibliotheken die D3.js verwenden
- > 31.03.2016: Osterferien
- > 07.04.2016: Daten visualisieren & Layouts
- > 14.04.2016: Skalen und Achsen & Responsive Design
- > 21.04.2016: User Experience, Usability Patterns
- > **28.04.2016: Zwischenpräsentation** & Datenaktualisierung und Transitionen
- > 05.05.2016: Auffahrt
- > 12.05.2016: Interactivity & Geomapping
- > 19.05.2016: 3D Web-Programmierung mit Three.js & Programming Coaching
- > **26.05.2016: Abschlusspräsentationen**
- > 02.06.2016: frei

# Vertiefte Einführung in die Programmierung

Durchgeführt von **Zeilenwerk**, Agentur für Webentwicklung in Bern, [www.zeilenwerk.ch](http://www.zeilenwerk.ch):  
**Raphael Reber, Cyril Nusko, Lukas Diener** (v.l.n.r.)

- > 25.02.2016: Einführung in HTML
- > 03.03.2016: Einführung in CSS
- > 10.03.2016: HTML <3 CSS
- > 17.03.2016: Einführung in Programmierkonzepte anhand von Javascript
- > 24.03.2016: Weiterführende Informationen und Übungen in Javascript



Jeweils **15:15 – 17:00 Uhr** Raum **003** am Institut für Wirtschaftsinformatik IWI,  
Engenhaldenstrasse 08

# Programming Coaching

Oscar Meier  
Hilfsassistent IWI  
Forschungsstelle Digitale Nachhaltigkeit



Janik Endtner  
Hilfsassistent IWI  
Forschungsstelle Digitale Nachhaltigkeit



Kontaktadresse für Fragen betreffend der Applikation:  
[opendata@iwi.unibe.ch](mailto:opendata@iwi.unibe.ch)

# Wichtige Informationen zur App-Entwicklung

- > Studierende visualisieren **vorhandene Daten**, Daten sammeln (data scraping) ist **sekundär**
- > **Klein beginnen**, immer mehr dazu entwickeln (iterativ)
- > **Rasch beginnen**, nicht auf Vorlesung und Übung warten
- > **Open Data Speed Dating** am Donnerstag, **10. März 2016**
- > Folgende **Datenquellen** sind möglich:
  - a) Daten von Data Coaches
  - b) Daten von Open Data Portalen
  - c) Eigene Daten
- > **Abschlusspräsentationen** aller Studierenden-Apps am Donnerstag, **26. Mai 2016** zwischen 13h und 16h, danach Apéro
- > **Beurteilung** durch Jury (Gastdozenten und Data Coaches), **Benotung** durch Matthias Stürmer und Thomas Myrach

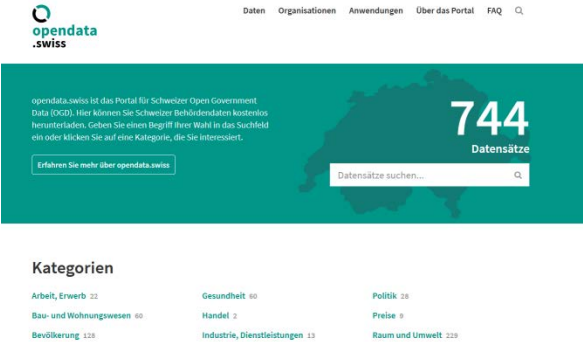
# Variante A) Daten von Data Coaches

1. **Christian Trachsel, SBB:** Zugzahlen der Schweiz und Haltestellenliste
2. **Ulrich Lantermann, Wikimedia.ch:** Wikidata
3. **Tobias Schalit, Bildungsplanung Bildungsstatistik Kanton Zürich:** Bildungsdaten Volks- und Mittelschulen aus dem Kanton Zürich
4. **Matthias Mazenauer, Statistisches Amt Zürich:** Gemeindefinanzen im Kanton Zürich, Nettoaufwendungen nach Aufgabenstellen
5. **David Oesch, Geoportal des Bundes:** Windenergieanlagen und Gesamtenergiestatistiken
6. **Beat Estermann, Berner Fachhochschule:** Wikimedia REST API
7. **Michael Fichter, Deloitte AG:** Klassifikationssysteme im Gesundheitswesen
8. **Hans Ulrich Wiedmer, Opendata.ch**
9. **Fabio Walti, Bernmobil:** Fahrgastzahlen, Fahrzeugpositionsdaten, Störungsmeldungen und Wetterbedingungen
10. **Philipp Lutz und Jonas Nakonz, Foraus:** öffentliche Daten des BFS zum Thema Migration
11. **Marco Sieber, Opendata Zürich:** Velozählungen
12. **Erich Helwin, Post:** Fahrplandaten
13. **Hansueli Pestalozzi, BAFU:** Umweltforschung in der Schweiz

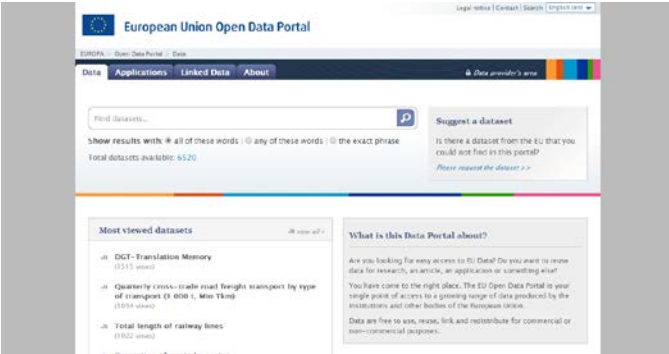


# Variante B) Daten von Open Data Portalen

## opendata.swiss



## open-data.europa.eu

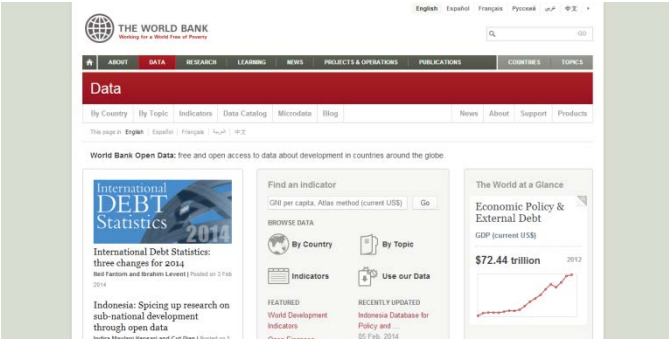


u.v.m.!

## data.un.org



## data.worldbank.org



## Variante C) Eigene Daten

- > **Selber Daten organisieren**
- > **Quellen** müssen angegeben werden
- > **Rohdaten** müssen verlinkt werden
- > Daten müssen Open Data sein (siehe **Definition Open Data**)
- > Wenn unsicher ob eigene Daten sinnvoll sind **bitte nachfragen**

# Beurteilung der Open Data Apps (BA und MA)

Open Data Apps werden nach folgenden 5 Kriterien bewertet:

1. **Funktionsumfang:** Welche Darstellungsmöglichkeiten und interaktiven Funktionen beinhaltet die Open Data App?
2. **Qualität:** Wie benutzerfreundlich, verständlich und gut dokumentiert ist die Open Data App?
3. **Komplexität:** Wie anspruchsvoll sind die visualisierten Daten und der behandelte Themenkomplex als ganzes?
4. **Impact:** Wie hoch ist die Bedeutung und die Aussagekraft der Datenvisualisierung und der Open Data App als gesamtes?
5. **Kreativität:** Wie neuartig und attraktiv sind die Visualisierung der Daten und technische Implementierung der Open Data App?

# Minimalanforderungen an Open Data App

1. **Lauffähig in Webbrowser und Verwenden von D3.js Bibliothek**, Anwendung weiterer Bibliotheken (AngularJS, NVD3.js etc.) und Datenbanken sind fakultativ
2. **Neuartige Daten** aufbereiten und visualisieren (mit Data Coach, von Datenportal etc.)
3. **Kreative Visualisierung**, nicht bloss Balken und Kreise
4. **Mindestens eine interaktive Funktion** (Mouse Click, Scroll Wheel, Buttons etc.)
5. Daten sowohl **grafisch (Visualisierung)** als auch als Zahlen bzw. Texte anzeigen (**Tooltip etc.**)
6. **Open Data Rohdaten** und **Quellenangaben** müssen verfügbar sein
7. **Übersicht der App**: Titel, Kurzbeschreibung (340 Zeichen), Namen der Autorinnen und Autoren, Emailadresse, Daten, Source Code

# Durchführung eines Open Data Projekts (nur MA)

Die selben Anforderung an Open Data App wie bei BA und **zusätzlich:**

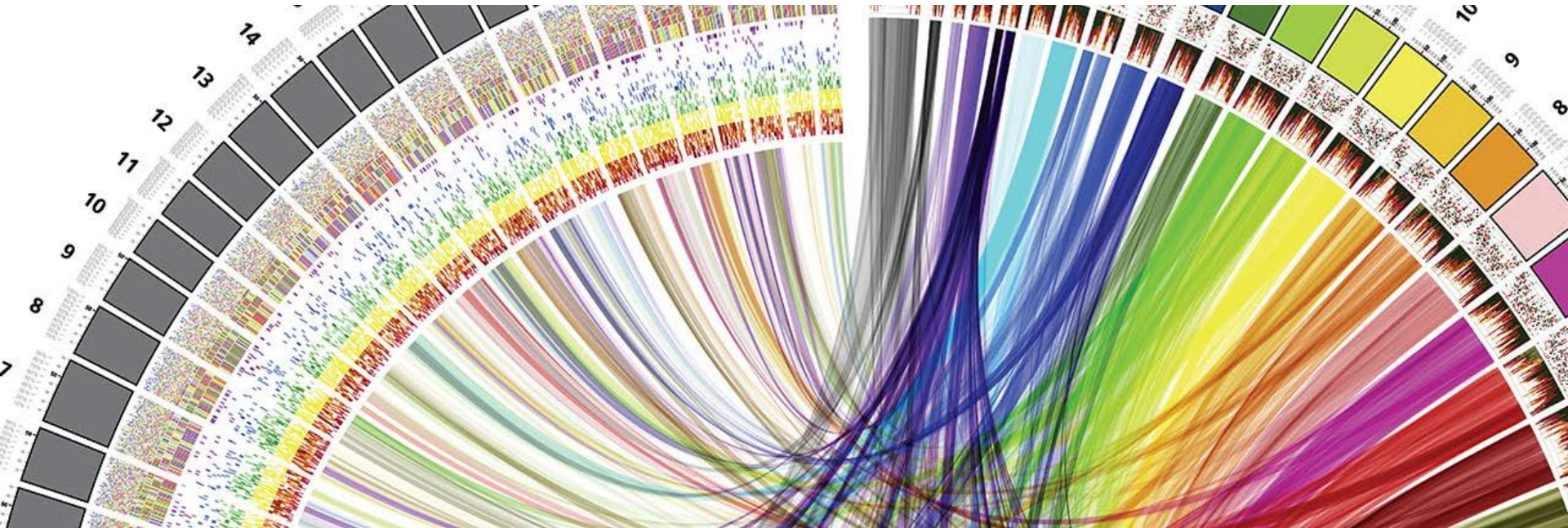
**Interpretation im Rahmen einer ausführlichen Anleitung zur Nutzung der Datenvisualisierung:**

1. **Beschreibung** der Open Data App (kleine Bedienungsanleitung)
  2. Erläuterungen der **Erkenntnisse** aus der Datenvisualisierung
  3. **Weiterführende Informationen**, Schlussfolgerungen etc., die mittels der Datenvisualisierung ermöglicht wurden
- 
- > **Zusatzaufwand** von rund 30 bis 40 Stunden pro Person muss ersichtlich sein.
  - > **Benotung:** Beschreibung zählt 25%, App-Umsetzung zählt 75%



# Agenda

1. Organisatorisches zur App-Entwicklung
2. **Interaktive Visualisierungen**
3. Einführung Entwickler-Tools
4. App-Demo von zwei ehemaligen Studierenden
5. Fragen und Antworten



# Umfrage zur Open Data Übung

- > **Kurze Umfrage** zu Studienrichtung, Übung und Programmiererfahrung
- > **Bis HEUTE ABEND, 25. Februar 2016** Umfrage auf ILIAS ausfüllen!
- > **Link:** [https://ilias.unibe.ch/goto\\_ilias3\\_unibe\\_svy\\_953127.html](https://ilias.unibe.ch/goto_ilias3_unibe_svy_953127.html)

ILIAS Universität Bern

**u<sup>b</sup>** Persönlicher Schreibtisch ▾ Magazin ▾ Support ▾ Portale ▾

Magazin » Wirtschafts- und Sozialwissenschaftliche Fakultät » Betriebswirtschaftslehre » FS2016 » Vorlesung » 398430-FS2016: Open Data Grundlagen und Übung » Umfrage zur Open Data Übung

## Umfrage zur Open Data Übung

Status: offline (nicht aktiviert)

20%

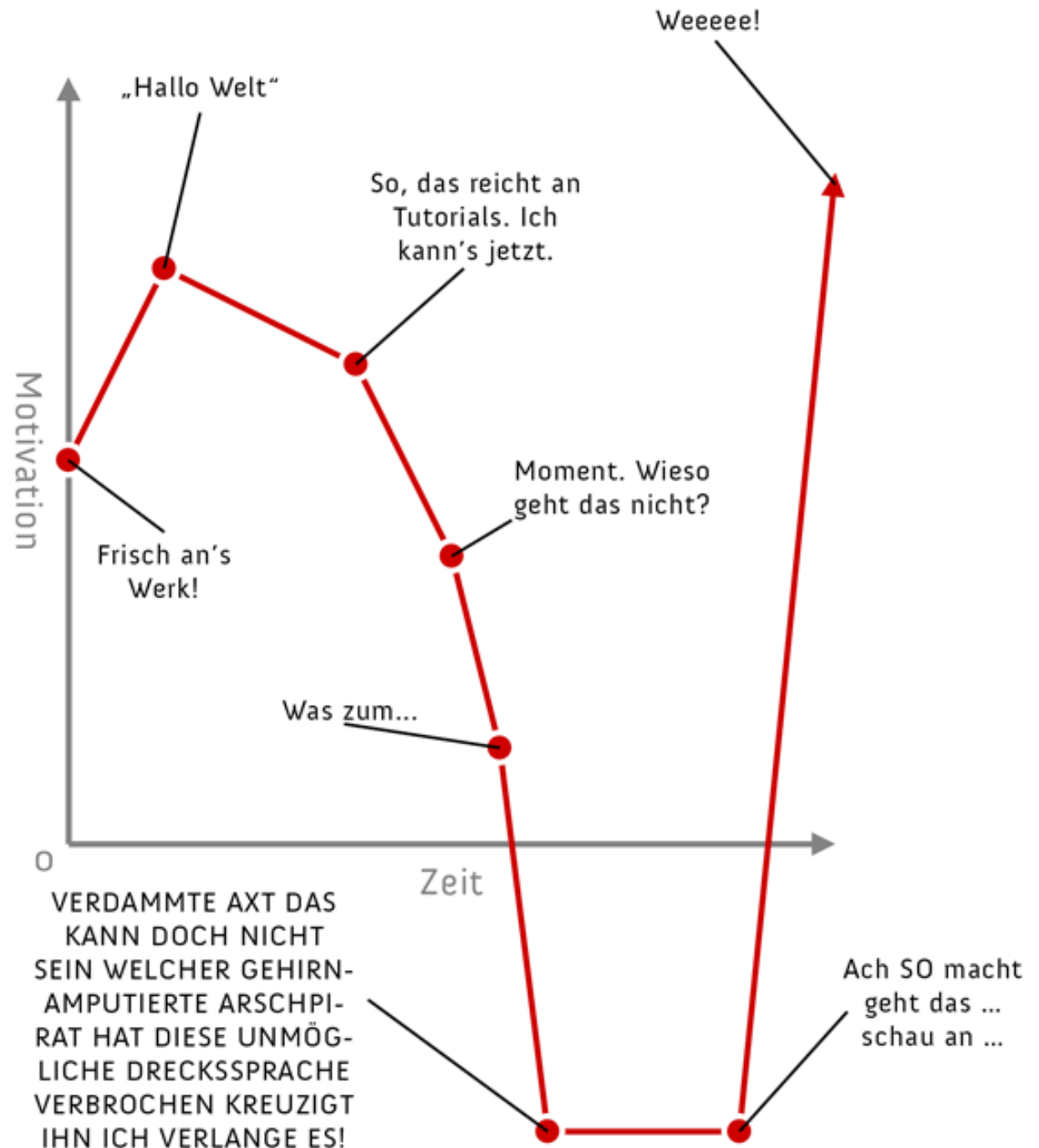
<< Zur Startseite Vorschau abbrechen Weiter >>

### Teilnahme an der Open Data Übung

Ich nehme an der Open Data Übung teil. \*

- ☐ Ja, ganz sicher
- ☐ Wahrscheinlich ja
- ☐ Eventuell, weiss noch nicht
- ☐ Nein

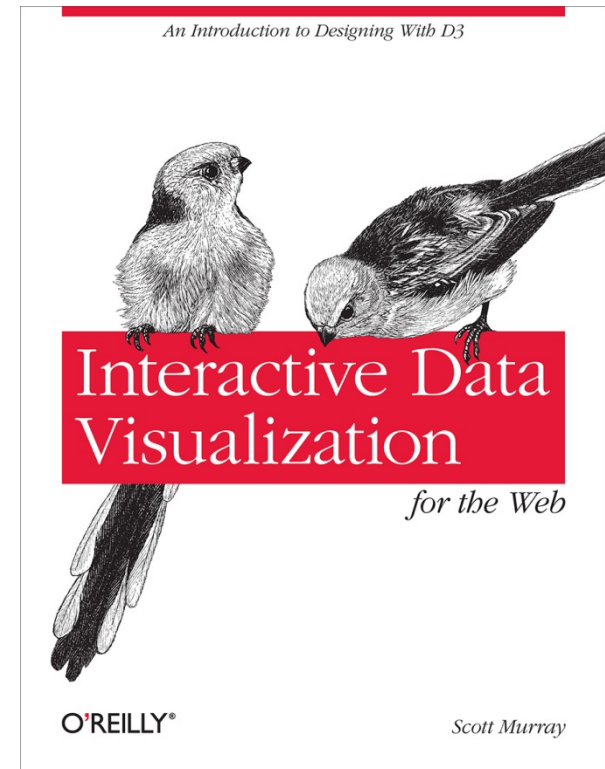
# Eine neue Programmiersprache lernen:



# Interactive Data Visualization for the Web

## Quelle:

- > O'Reilly Media, von Scott Murray
- > März 2013, 272 Seiten, Englisch
- > ISBN-10: 1449339735
- > **Gratis online als ebook**
- > Auf Amazon.de für CHF 22.50
- > „Create and publish your own interactive data visualization projects on the Web—even if you have little or no experience with data visualization or web development.”
- > Total 13 Kapitel, 12 Kapitel davon werden in den Übungen behandelt



# Gratis, interaktives Online-Buch

<http://chimera.labs.oreilly.com/books/12300000000345/index.html>

Interactive Data Visualization for the Web

fluent

Enjoy this interactive version of *Interactive Data Visualization for the Web*. Purchase and download the DRM-free ebook on [oreilly.com](http://oreilly.com).  
Learn more about the O'Reilly Ebook Advantage.

Buy the Ebook

## Interactive Data Visualization for the Web

Scott Murray

Copyright © 2013 Scott Murray

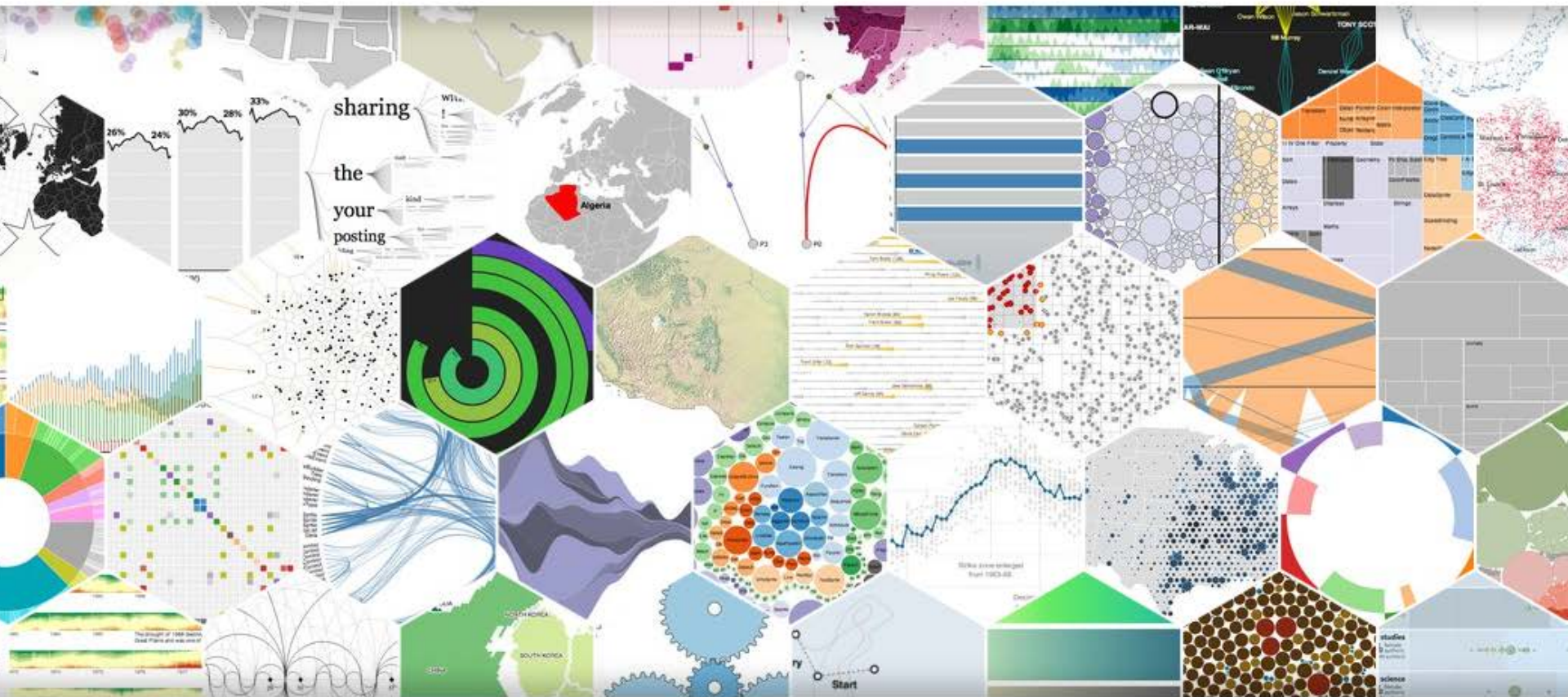
Table of Contents

The ⚡ icon indicates sections that contain interactive examples.





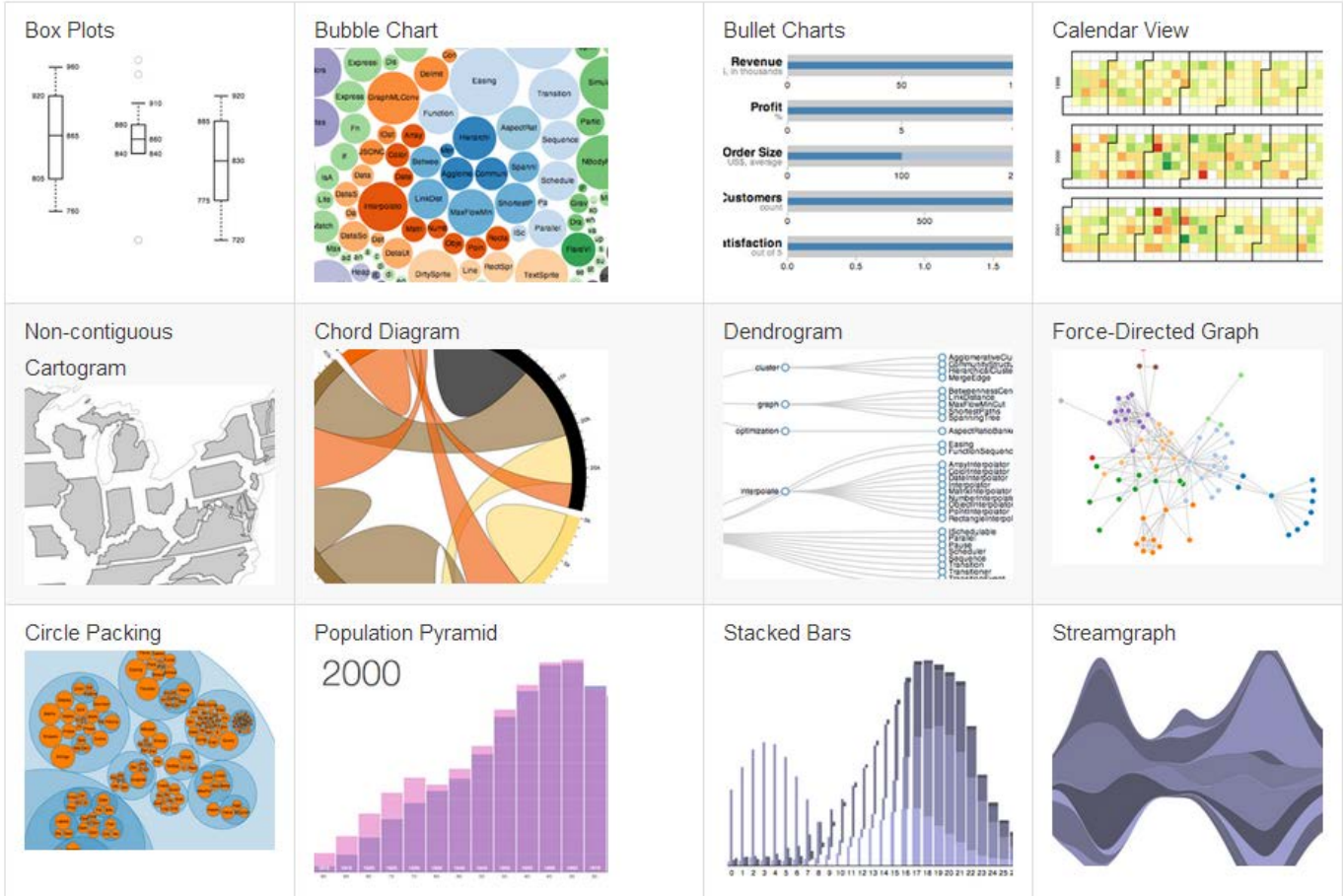
# Data-Driven Documents



**D3.js** is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

See [more examples](#).

# D3.js Code Snippets



Link: <https://github.com/mbostock/d3/wiki/Gallery>



# D3.js Code Snippets

Author

Chart Type

Title

Untagged 1122

Map 236

Network 69

Reusable 65

Bar Chart 58

Line Chart 49

Math 46

Scatterplot 39

Bubble Chart 26

Area Chart 25

Tree 23

Voronoi 17

Pie Chart 15

Parallel Coordinates 15

Chord Diagram 15

Choropleth 14

Sankey 14

Stacked Bar Chart 14

Experiment 12

U.S. TopoJSON

Mike Bostock

113th U.S. Congressional Districts

Mike Bostock

20000 points in random motion

Kai Chang

2012 NFL Conference Champs

Dylan Harper

Ancient Histogram

Stewart Noyce

25 great circles

Dealgia McArdle

Legendary Key

Stewart Noyce

Open Chord Diagram

Stewart Noyce

3D bar chart with D3.js and x3dom

Harry Voorhees

Box Transition Revisited

Stewart Noyce

401k Fees Vary Widely for Similar Companies (Scatter)

Amelia Greenhall

512 Paths to the White House

Mike Bostock; Shan Carter

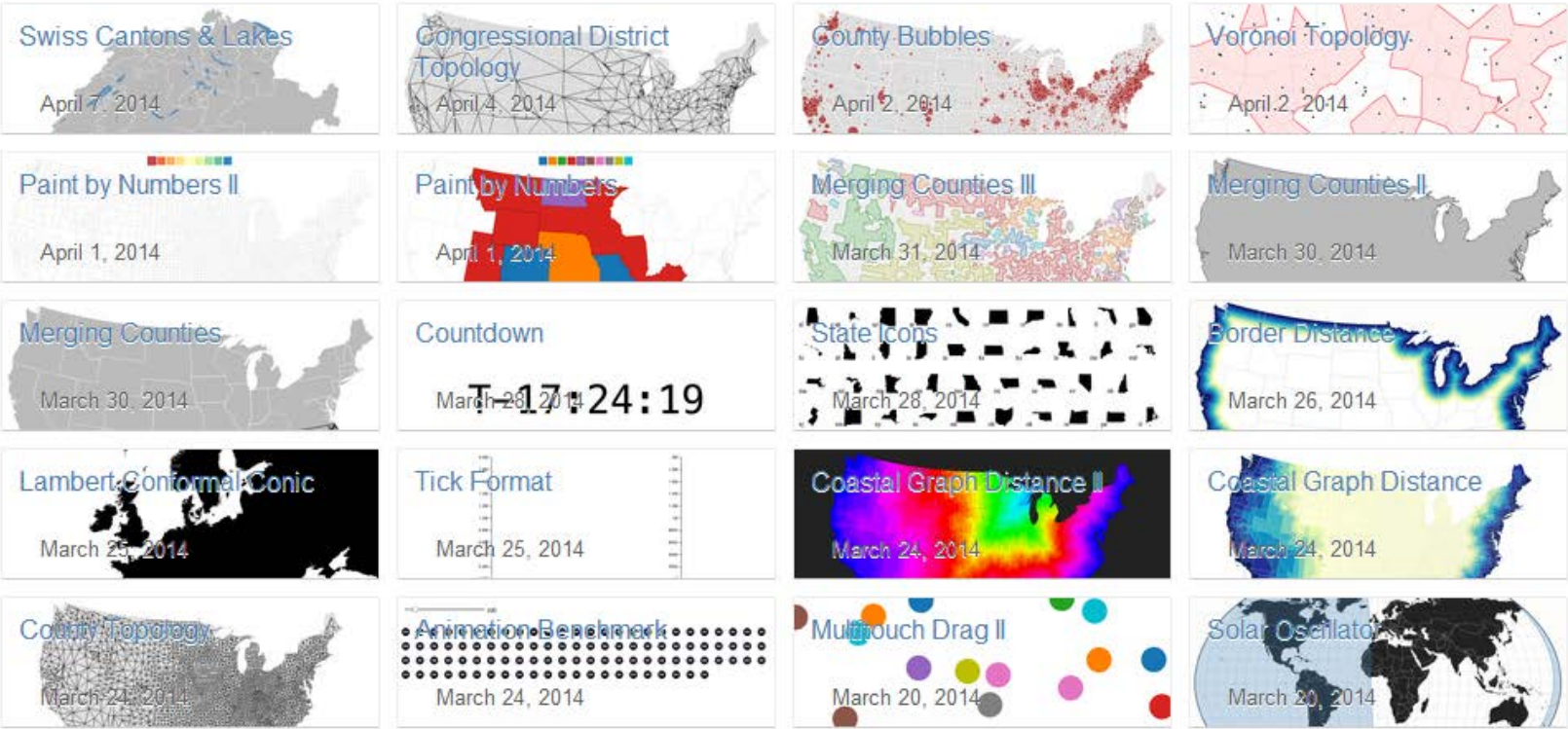
Link: <http://christopheviau.com/d3list/gallery.html>





# D3.js Code Snippets

## mbostock's blocks



Link: <http://bl.ocks.org/mbostock>

# D3.js: Cushion Treemaps

To be presented at the IEEE Symposium on Information Visualization (INFOVIS 99), San Francisco, October 25-26, 1999

**Cushion Treemaps: Visualization of Hierarchical Information**

Jarke J. van Wijk   Huub van de Wetering  
Eindhoven University of Technology  
Dept. of Mathematics and Computing Science  
P.O. Box 513, 5600 MB Eindhoven, The Netherlands  
{vanwijk, wstahw}@win.tue.nl

**Abstract**

A new method is presented for the visualization of hierarchical information, such as directory structures and organization structures. Cushion treemaps inherit the elegance of standard treemaps: compact, space-filling displays of hierarchical information, based on recursive subdivision of a rectangular image space. Intuitive shading is used to provide insight in the hierarchical structure. During the subdivision ridges are added per rectangle, which are rendered with a simple shading model. The result is a surface that consists of recursive cushions. The method is efficient, effective, easy to use and implement, and has a wide applicability.

**2 Background**

Many methods exist to display and browse through hierarchical information structures, or, for short, trees. File browsers are the best known example. Usually a listing of the files and directories is used, where the levels in the hierarchy are shown by means of indentation. The number of files and directories that can be shown simultaneously is limited, which is no problem if one knows what to search for. However, if we want to get an overview, or want to answer a more global question, such as "Why is my disk full?", scrolling, and opening and closing of subdirectories have to be used intensively. During this process it is hard to form a mental image of the overall structure [3].

**1 Introduction**

Hierarchical structures of information are ubiquitous: family trees, directory structures, organization structures, catalogues, computer programs, etcetera. Small hierarchical structures are very effective to locate information, but the content and organization of large structures is much harder to grasp.


We present a new visualization method for such large hierarchical structures: Cushion Treemaps. The method is based on treemaps, developed by Shneiderman and Johnson [11, 8]. Treemaps are efficient and compact displays, which are particularly effective to show the size of the final elements in the structure. Cushion Treemaps provide shading as a strong extra cue to emphasize the hierarchical structure. For a quick impression, figure 2 and 3 show treemaps, figure 5 and 6 show the corresponding cushion treemaps.

In section 2 we discuss existing methods to visualize hierarchical structures. The new method is presented in section 3. The embedding of the method in an interactive system for tree visualization is described in section 4. Finally, we discuss extensions and alternatives in section 5, and we summarize the results in section 6.

**Figure 1. Tree representations**

Many techniques have been proposed to show such structures more effectively. An important category are node and link diagrams (fig. 1). Elements are shown as nodes, relations are shown as links from parent to child nodes. Sophisticated techniques have been presented to improve the efficiency and aesthetic qualities of such diagrams, both in 2D and in 3D [9, 7, 1, 2, 10]. Such diagrams are very effective for small trees, but usually fall short when more than a couple of hundred elements have to be visualized simultaneously. The main reason for this limitation is simply that node and link diagrams use the display space inefficiently: Most of the pixels are used as background.

Treemaps [11, 8] were developed to remedy this problem.



**Figure 2. Treemap of file system**

The full display space is used to visualize the contents of the tree. Here we present an overview of the concept, an in depth treatment is given in the original references. Figure 1 shows an example.

Each node (as shown in the node and link diagram) has a name (a letter) and an associated size (a number). The size of leaves may represent for instance the size of individual files, the size of non-leaf nodes is the sum of the sizes of its children. The treemap is constructed via recursive subdivision of the initial rectangle. The size of each sub-rectangle is proportional to the size of the node. The direction of subdivision alternates per level: first horizontally, next vertically, etcetera. As a result, the initial rectangle is partitioned into smaller rectangles, such that the size of each rectangle reflects the size of the leaf. The structure of the tree is also reflected in the treemap, as a result of its construction. Color and annotation can be used to give extra information about the leaves.

Treemaps are very effective when size is the most important feature to be displayed. Figure 2 shows an overview of a file system: 1400 files are shown and one can effortlessly determine the largest ones. Labels are not shown here, but can be shown interactively by pointing at the areas of interest.

However, treemaps have limitations [4]. The problem addressed here is that treemaps often fall short to visualize the structure of the tree. The worst case is a balanced tree, where each parent has the same number of children and each leaf has the same size. The tree-map degenerates here into a regular grid. Indeed, leaves that are close in the tree are also close on the screen, but the reverse is not always true. As an example, figure 3 shows an (artificial) organization chart, modelled after the structure of our university. The university has seven faculties, subdivided into departments, which in turn are divided into sections. Each section is divided into units. Each unit contains four different types of staff members (full, associate, and assistant professor, PhD-student). The final 3060 rectangles denote individual employees. Questions such as "What is the largest section?" or "Is the division into units balanced?" are hard to answer from such an image.

Nested treemaps [8] are a partial remedy. During the subdivision process instead of the initial rectangle a slightly smaller rectangle is used, such that each group of siblings is enclosed by a margin. However, this consumes screen space and the visual interpretation, especially for deeply nested trees, requires effort from the viewer. The variation of properties of the surrounding lines is another option. However, the number of steps in linewidth or intensity that can be discerned without effort is small, and also here the user is required to trace lines in a maze-like image. Coloring the rectangles would not work either. Color does not provide a natural hierarchical structure, and furthermore, we want to use color to show other attributes of the elements.

One alternative has not been exploited yet: The use of shading to visualize the structure. In the remainder of this paper we will show how this can be done.

**3 Cushion treemaps**

**3.1 Method**

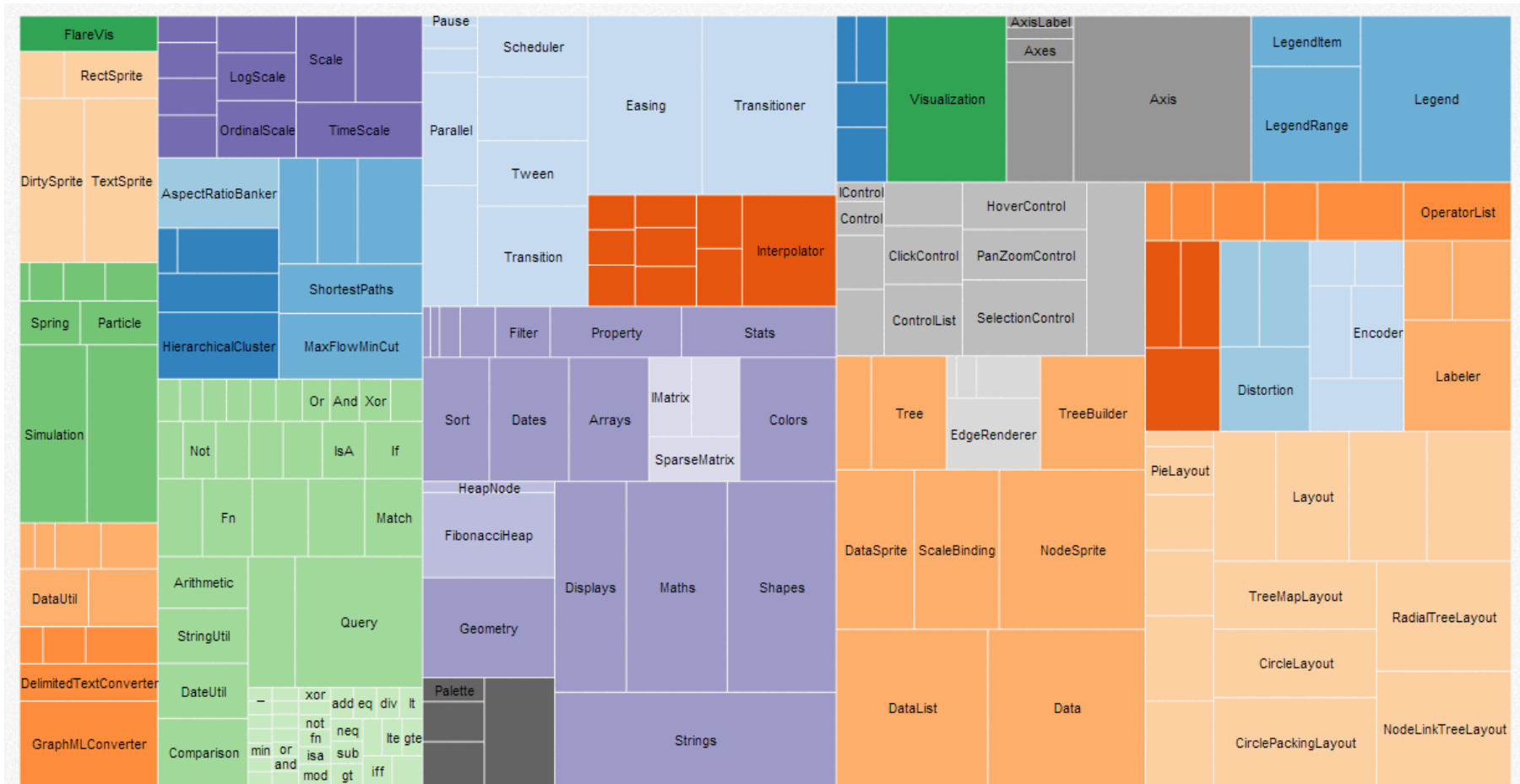
How can we use shading to show the tree structure? The human visual system is trained to interpret variations in shade as illuminated surfaces [6]. Hence, we can answer the question by constructing a surface which shape encodes the tree structure.

We introduce our solution with a simple one-dimensional example: binary subdivision of an interval. First, we subdivide the interval and add a bump to each of the two sub-

Quelle: Jarke J. van Wijk and Huub van de Wetering, Cushion Treemaps: Visualization of Hierarchical Information, 1999  
<http://www.win.tue.nl/~vanwijk/ctm.pdf>



# D3.js: Zoomable Treemaps



Quelle: <http://mbostock.github.io/d3/talk/20111018/treemap.html>

# D3.js: Obama's 2012 budget proposal

The New York Times | Politics

## Obama's 2012 Budget Proposal: How \$3.7 Trillion is Spent

Explore every nook and cranny of President Obama's budget proposal.

RELATED ARTICLE  
[Obama Budget Pivots From Stimulus to Deficit Cuts](#)

Rectangles are sized according to the proposed spending. Color shows severity of cut or increase from 2010.

-15% -10% 0 +10% +15%

The president has proposed a five-year freeze of discretionary spending, excluding national security spending. This type spending accounts for about one fifth of all spending.  
[Isolate discretionary spending.](#)

Mandatory spending, which includes entitlement programs like Medicare and Social Security, is expected to continue to rise. [Isolate mandatory spending.](#)

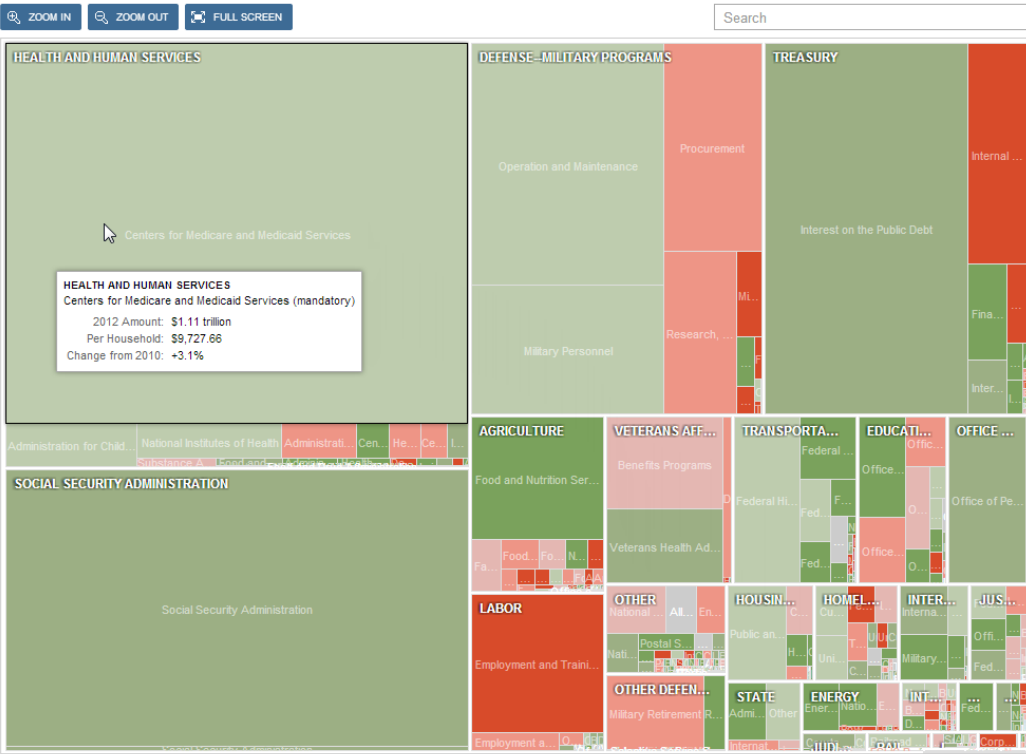
The proposal includes cuts to low income home energy assistance and community service block grants. [Zoom in.](#)

Cuts in the Environmental Protection Agency's budget include reducing funds restore the Great Lakes' environmental health. [Zoom in.](#)

Mr. Obama wants to spend more to train math, science and engineering teachers and to expand effective programs. [Zoom in.](#)

The Energy Department's budget is 12 percent higher than it was in 2010, including increases for clean energy programs. [Zoom in.](#)

Note: Chart shows funds authorized to be spent during fiscal year. It does not show "off-setting receipts"— items like postage stamp sales, park fees, Medicare prescription drug premiums and federal

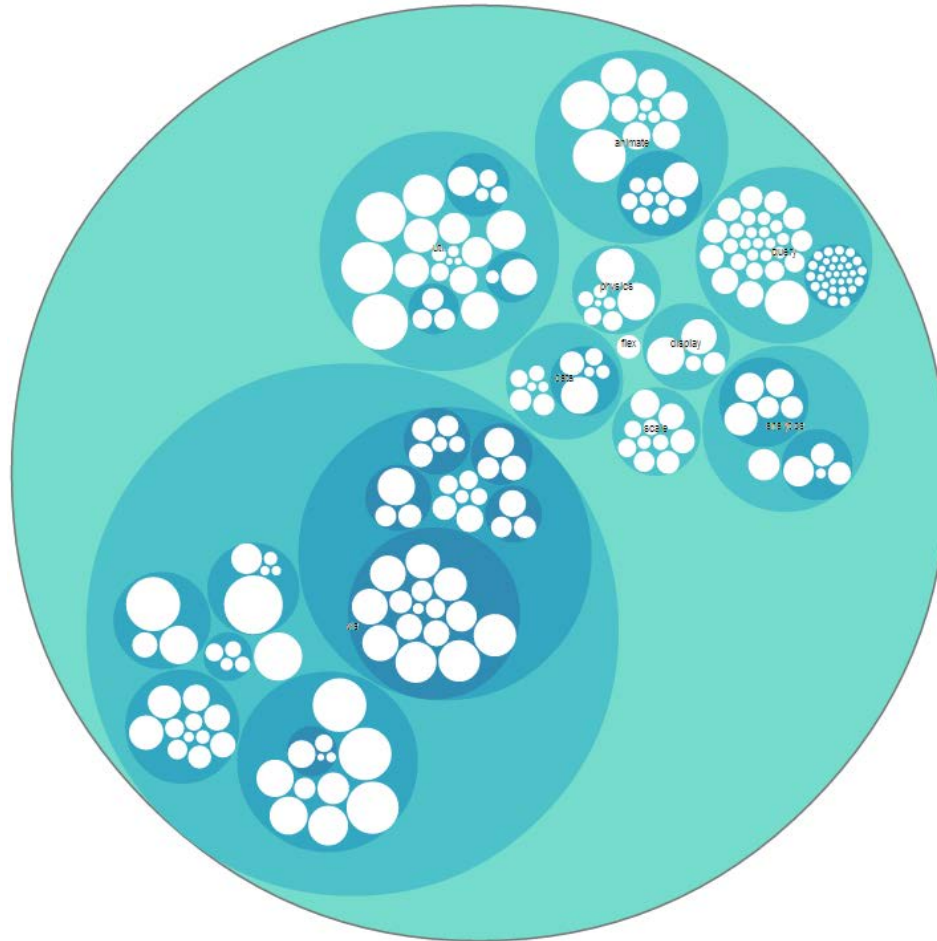


Published: February 14, 2011 | By SHAN CARTER and AMANDA COX | Source: Office of Management and Budget

[TWITTER](#) [FACEBOOK](#)

Quelle: <http://www.nytimes.com/packages/html/newsgraphics/2011/0119-budget>

## D3.js: Zooma

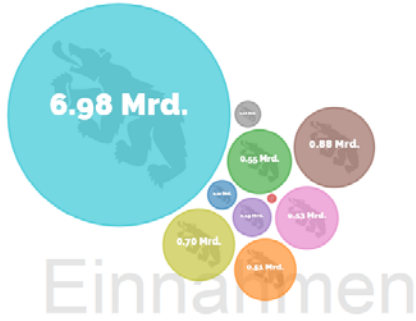


Quelle: <http://bl.ocks.org/mbostock/7607535/>

# D3.js: Finanzen Kanton Bern

Finanzdirektion des Kantons Bern

2014



## Die Finanzen des Kantons Bern

[Alle Einnahmen und Ausgaben](#)


Die Angaben enthalten die Aufwendungen und Erträge der Laufenden Rechnung sowie die Ausgaben und Einnahmen der Investitionsrechnung.

Kategorie	Einnahmen 2014 in Fr.	Ausgaben 2014 in Fr.
<a href="#">Allgemeine Verwaltung</a>	104'300'504	650'780'239
<a href="#">Öffentliche Ordnung und Sicherheit, Verteidigung</a>	507'881'767	1'046'111'184
<a href="#">Bildung</a>	552'435'826	2'551'240'316
<a href="#">Kultur, Sport und Freizeit, Kirche</a>	6'183'820	148'240'194
<a href="#">Gesundheit</a>	189'585'317	1'777'105'310
<a href="#">Soziale Sicherheit</a>	884'802'977	2'123'080'054
<a href="#">Verkehr</a>	528'216'491	844'364'528
<a href="#">Umweltschutz und Raumordnung</a>	80'961'117	125'175'744
<a href="#">Volkswirtschaft</a>	695'617'051	740'432'532
<a href="#">Finanzen und Steuern</a>	6'978'409'598	474'425'977
<b>Total</b>	<b>10'528'394'446</b>	<b>10'460'954'078</b>

Für weitere Informationen zu den Ausgabekategorien auf die Informationssymbole klicken.

Quelle: [http://files.be.ch/fin/gs/ausgaben\\_ktbe/index.html#/2014/](http://files.be.ch/fin/gs/ausgaben_ktbe/index.html#/2014/)

# Sankey Diagram




**WIKIPEDIA**  
The Free Encyclopedia

Main page  
Contents  
Featured content  
Current events  
Random article  
Donate to Wikipedia  
Wikimedia Shop

Interaction  
Help  
About Wikipedia  
Community portal  
Recent changes  
Contact page

Tools  
Print/export

Languages 

العربية  
Deutsch  
Español  
Français  
Nederlands  
日本語  
Polski  
Suomi


Article


Talk

Read

Edit

View history





Search



 Maemst 0 [Talk](#) [Sandbox](#) [Preferences](#) [Beta](#) [Watchlist](#) [Contributions](#) [Log out](#)

## Sankey diagram

From Wikipedia, the free encyclopedia

**Sankey diagrams** are a specific type of [flow diagram](#), in which the width of the arrows is shown proportionally to the flow quantity. They are typically used to visualize [energy](#) or material or [cost](#) transfers between processes.

Contents [hide]

1 Application

1.1 Historical Examples

1.2 Active Examples

2 See also

3 References

4 External links

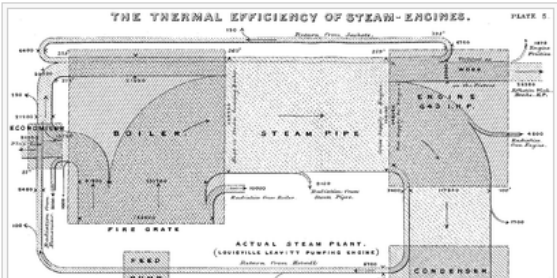
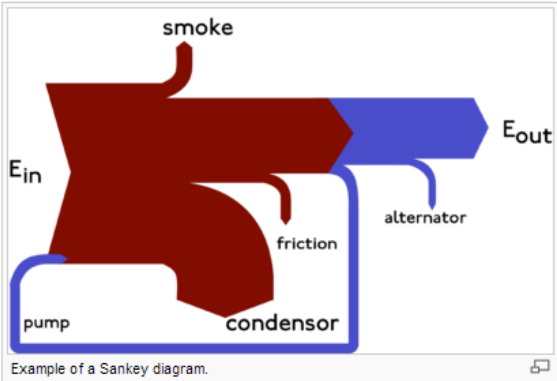
### Application [edit]

They are also commonly used to visualize the energy accounts or material flow accounts on a regional or national level. Sankey diagrams put a visual emphasis on the major transfers or flows within a system. They are helpful in locating dominant contributions to an overall flow. Often, Sankey diagrams show conserved quantities within defined system boundaries.

### Historical Examples [edit]

Sankey diagrams are named after Irish Captain [Matthew Henry Phineas Riall Sankey](#), who used this type of diagram in 1898 in a classic figure (see panel on right) showing the [energy efficiency](#) of a [steam engine](#). While the first charts in black and white were merely used to display one type of flow (e.g. steam), using colors for different types of flows has added more degrees of freedom to Sankey diagrams.

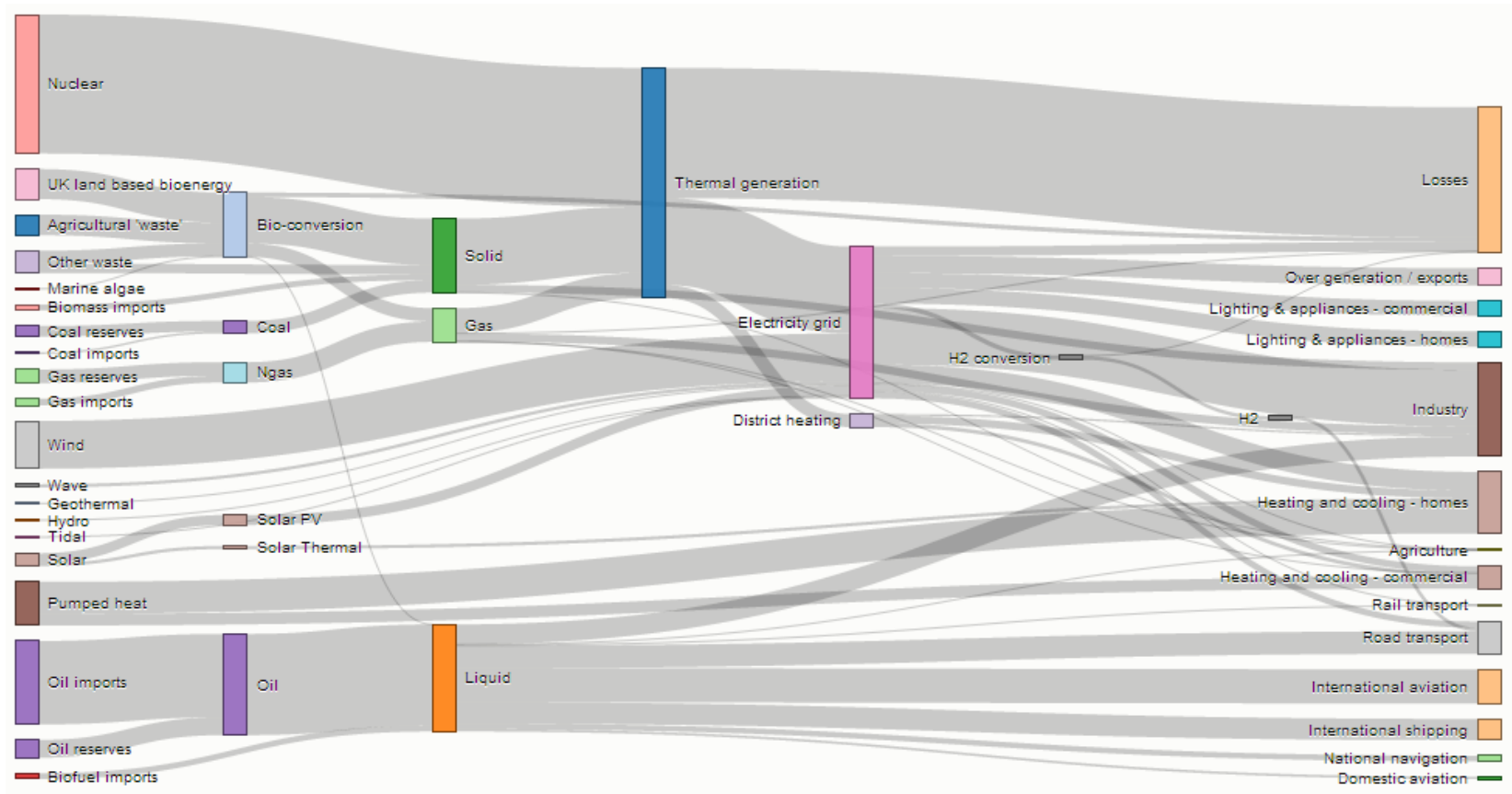
One of the most famous Sankey diagrams is [Charles Minard's](#) Map of Napoleon's Russian Campaign of 1812. It is a [flow map](#), overlaying a Sankey diagram onto a geographical map. It was created in 1869, so it actually predates Sankey's first Sankey diagram of 1898.



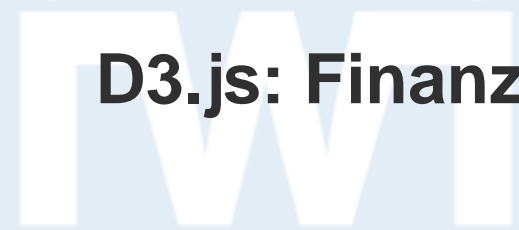
Quelle: [https://en.wikipedia.org/wiki/Sankey\\_diagram/](https://en.wikipedia.org/wiki/Sankey_diagram/)



# D3.js: Sankey Diagrams

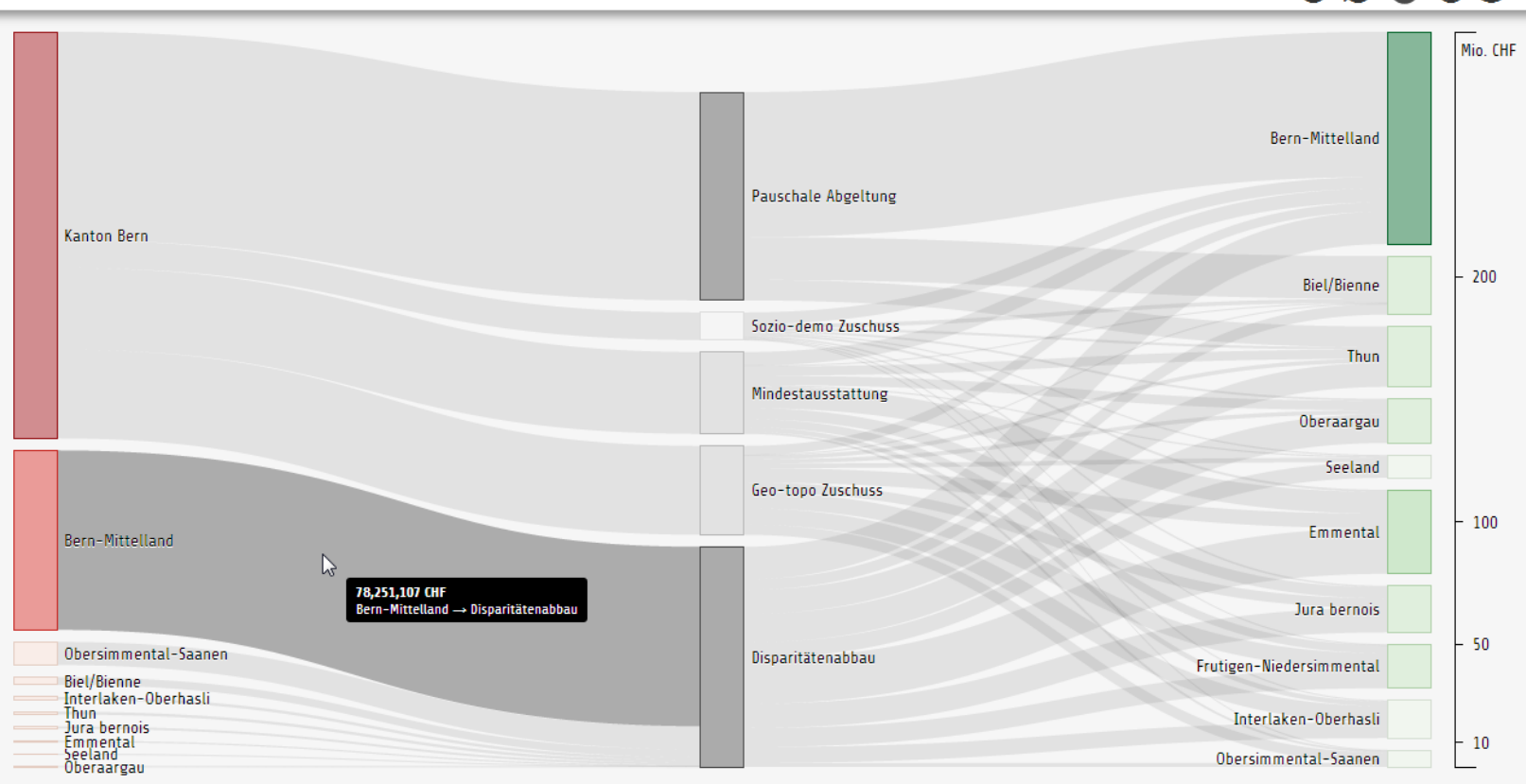


Quelle: <http://bost.ocks.org/mike/sankey/>



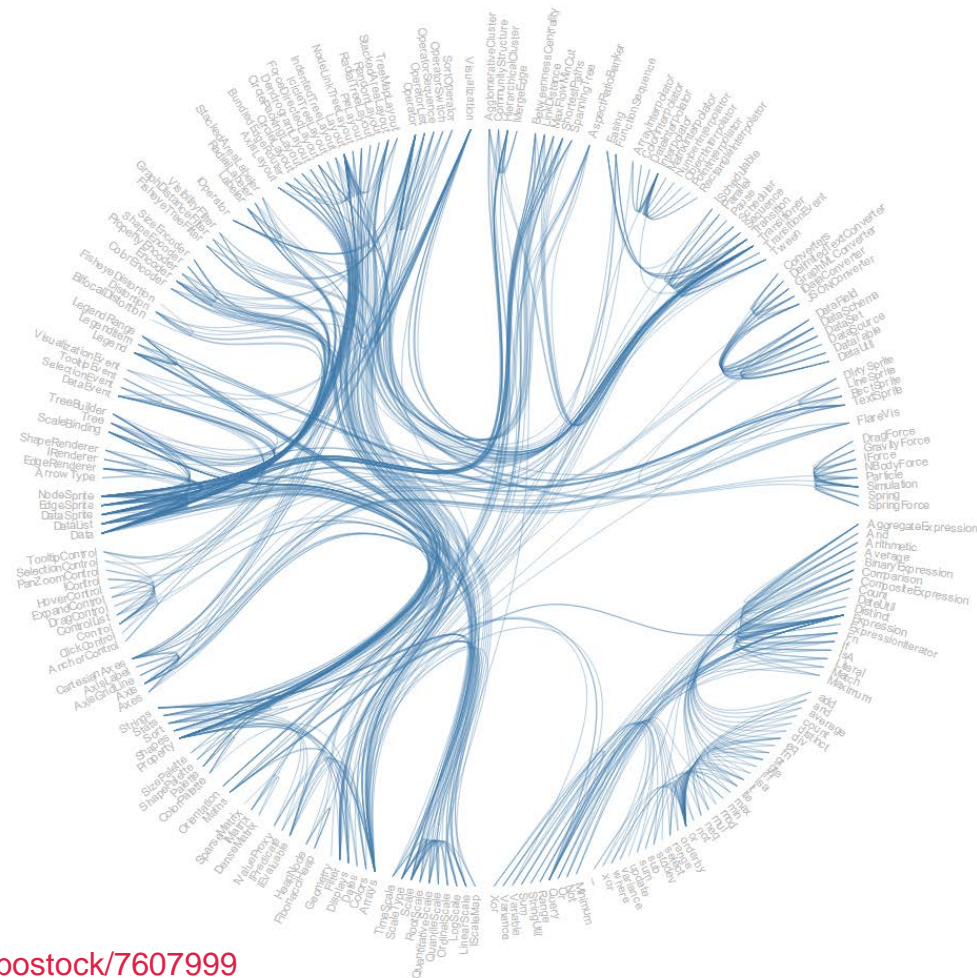
# D3.js: Finanzausgleich im Kanton Bern

Finanzausgleich im Kanton Bern (Vollzug 2012)



Quelle: <http://be-fa.budget.opendata.ch/>

## D3.js: Hierarchical



Quelle: <http://bl.ocks.org/mbostock/7607999>

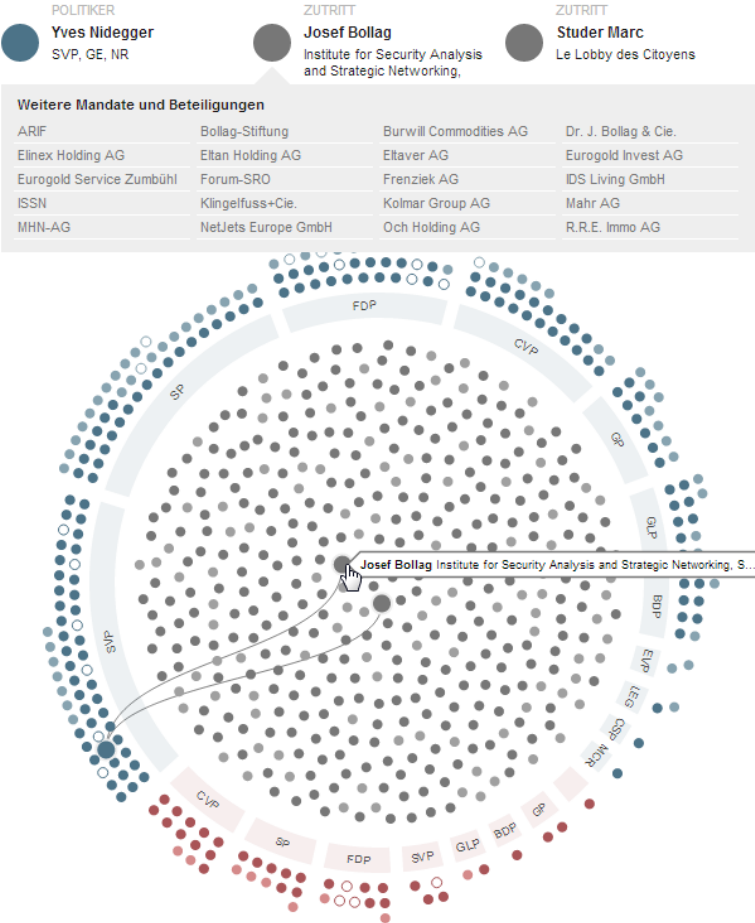
# D3.js: Visualisierte Interessenbindungen im Schweizer Parlament

## In der Wandelhalle

Wer ermöglicht Chris von Rohr den Zugang zum Bundeshaus? Welche Interessen vertritt dort Raymond Loretan? Und wie viele Vertreter von Economiesuisse gehen in Bundesbern ein und aus? Die interaktive Visualisierung gibt einen Überblick über die Zutrittsberechtigten und die Interessen, die sie vertreten.

Jeder Parlamentarier kann zwei Personen für den Zugang nominieren. Im äusseren Kreis der Visualisierung sind die Nationalräte in Blau und die Ständeräte in Rot dargestellt. Im inneren Kreis finden die Zutrittsberechtigten Platz. Mit einem Klick erhalten Sie weitere Selektionsoptionen und können so das Netzwerk aus persönlichen Mitarbeitern, Interessens- und Unternehmensvertretern in Bundesbern erkunden.

Weiter



Quelle: NZZ, 4. März 2014

<http://www.nzz.ch/aktuell/schweiz/die-daten-hinter-der-visualisierung-1.18255344>

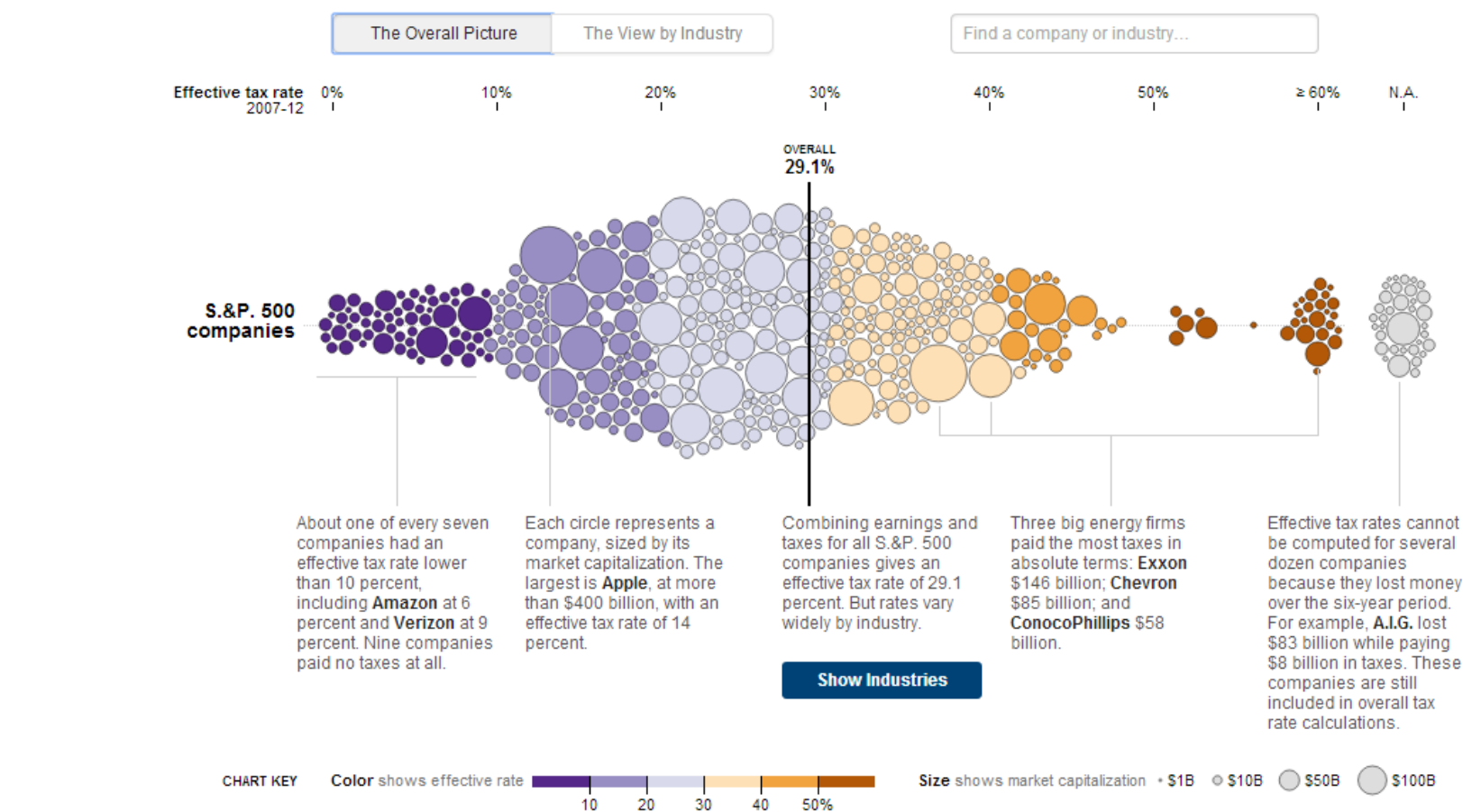
# D3.js: Clustered Force Layout I



Quelle: <http://bl.ocks.org/mbostock/1747543>

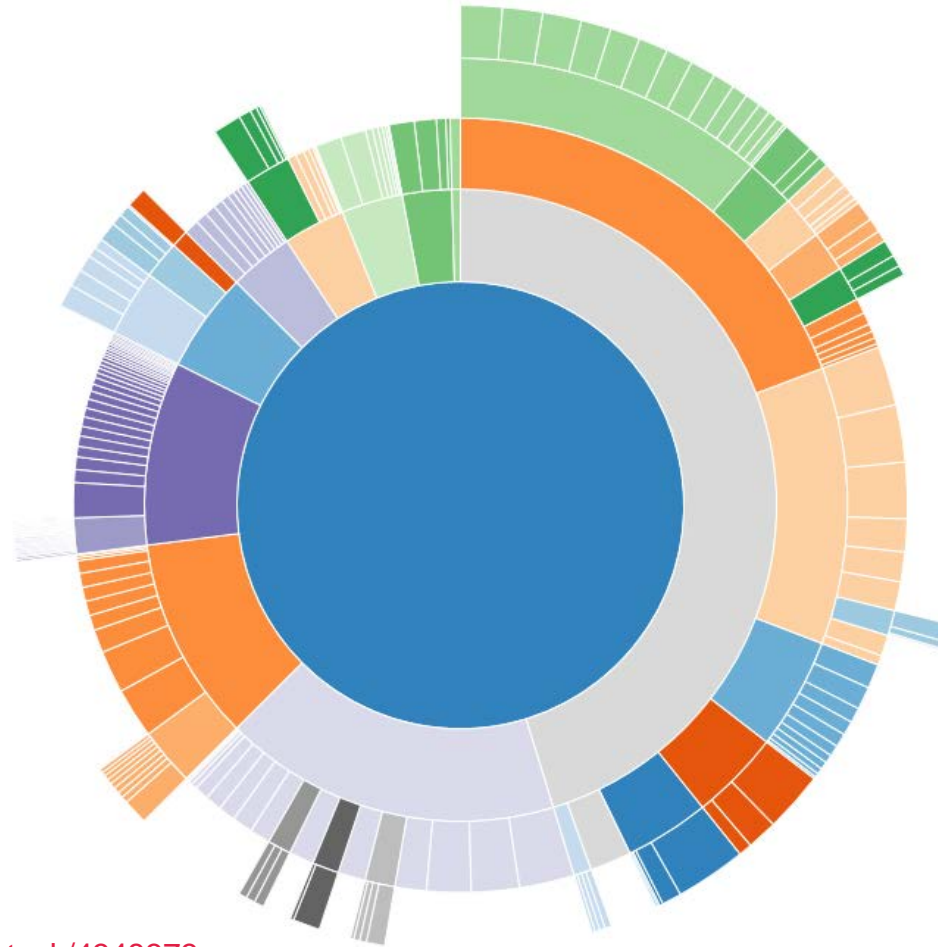


# D3.js: Tax Rates of U.S. Companies



Quelle: [http://www.nytimes.com/interactive/2013/05/25/sunday-review/corporate-taxes.html?\\_r=0](http://www.nytimes.com/interactive/2013/05/25/sunday-review/corporate-taxes.html?_r=0)

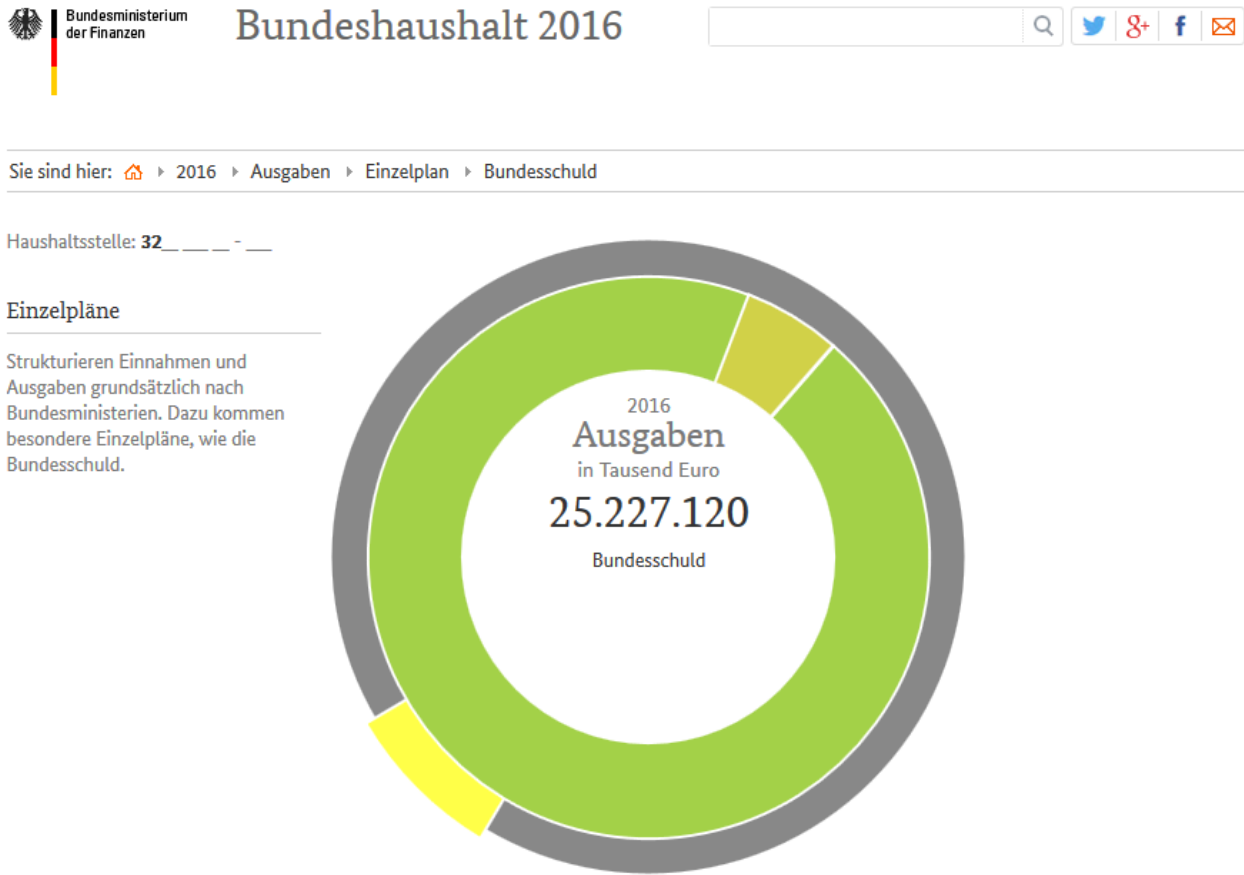
## D3.js: Zoomable Sunburst



Quelle: <http://bl.ocks.org/mbostock/4348373>



# D3.js: Finanzen Deutschland



Sie sind hier: &gt; 2016 &gt; Ausgaben &gt; Einzelplan &gt; Bundesschuld

Haushaltsstelle: 32\_\_ \_\_ - \_\_

Einzelpläne

Strukturieren Einnahmen und Ausgaben grundsätzlich nach Bundesministerien. Dazu kommen besondere Einzelpläne, wie die Bundesschuld.

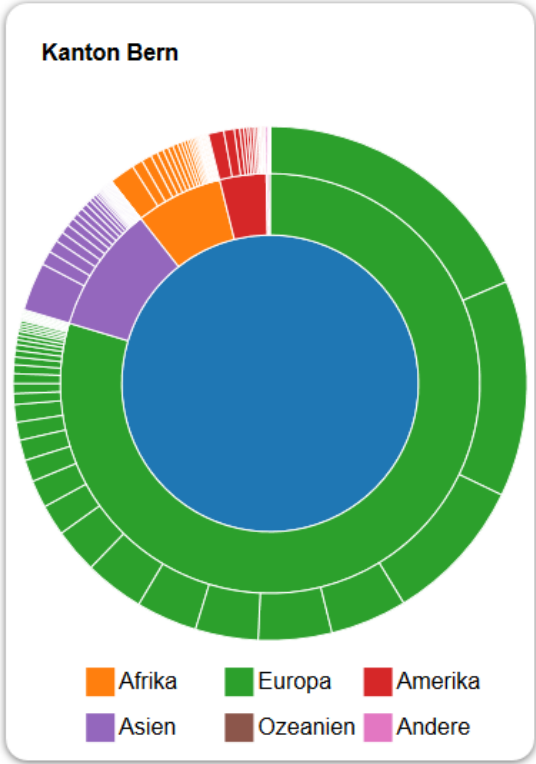
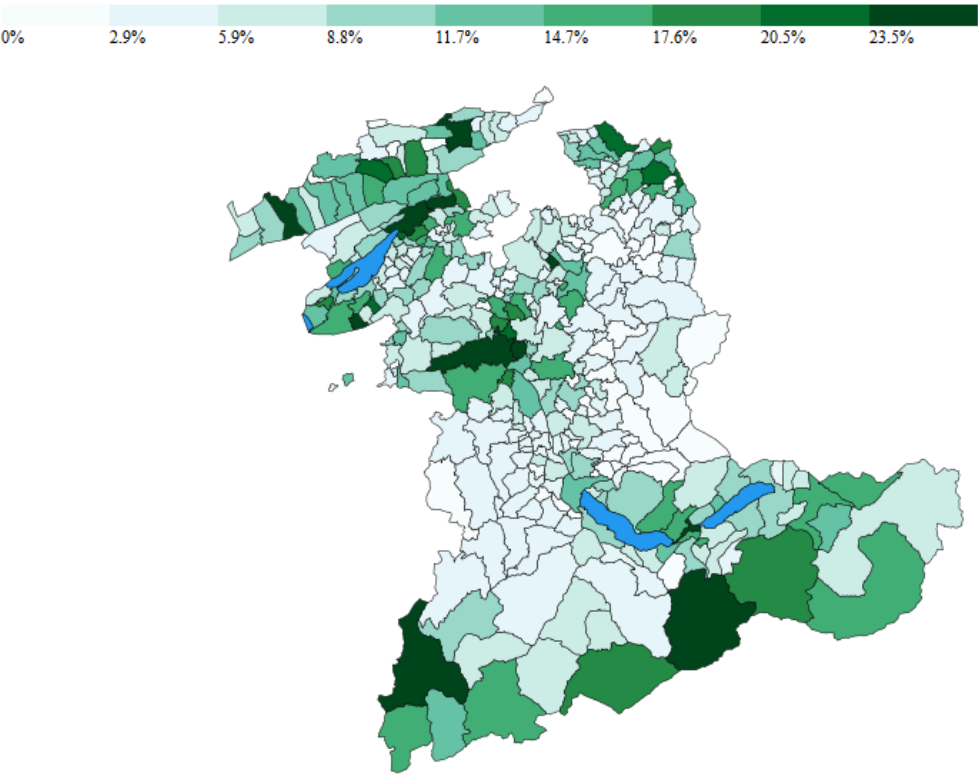


Quelle: <http://www.bundeshaushalt-info.de/#/2016/soll/ausgaben/einzelplan/32.html>

# D3.js: Ausländische Wohnbevölkerung

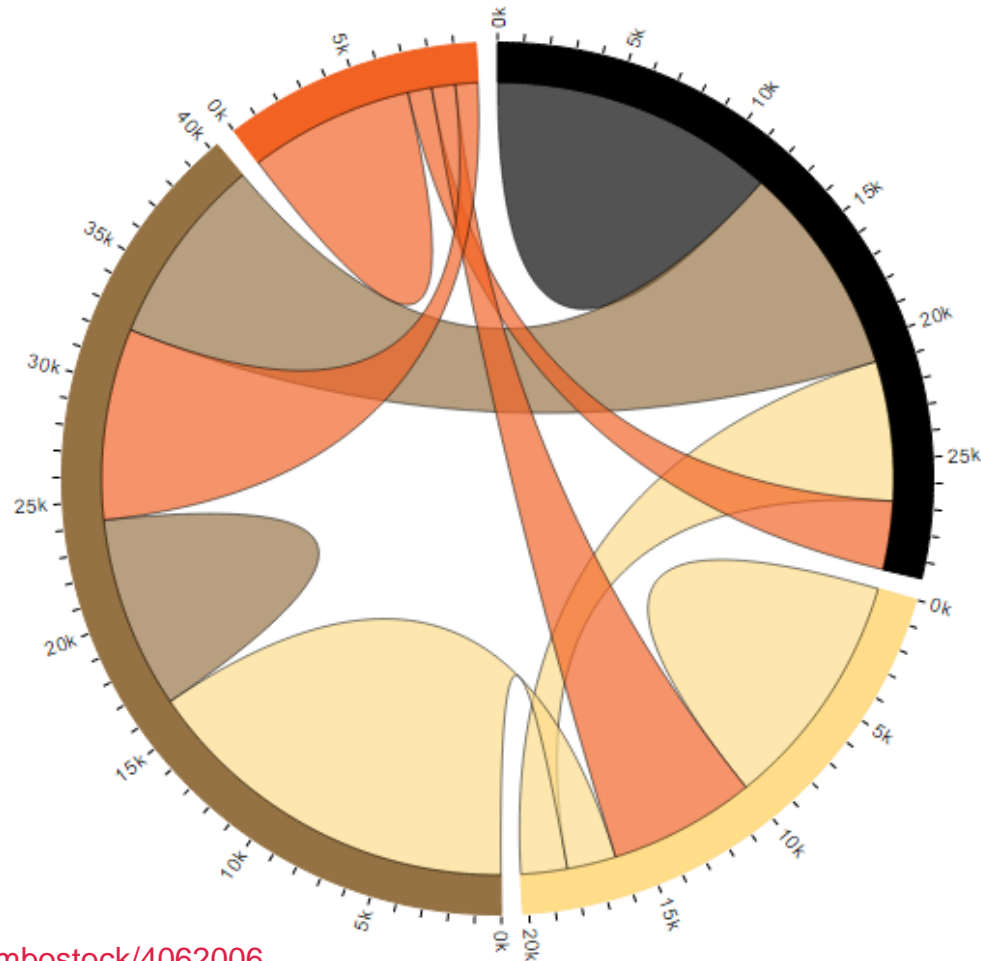
Anteil Bürger ausländischer Herkunft an der ständigen Wohnbevölkerung

  2013 ▾



Quelle: <http://lh10p414.sandbox.iwi.unibe.ch/>

# D3.js: Chord Diagram



Quelle: <http://bl.ocks.org/mbostock/4062006>

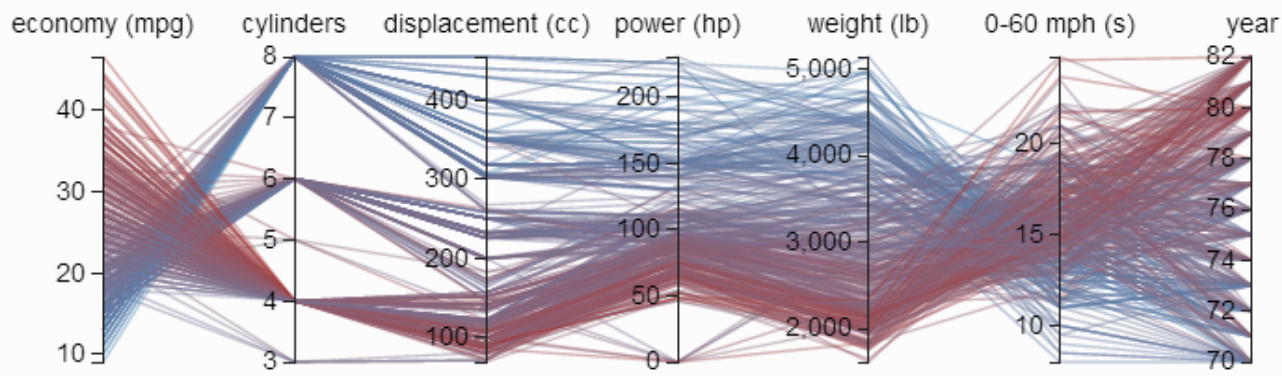


# D3.js: Eurozone debt



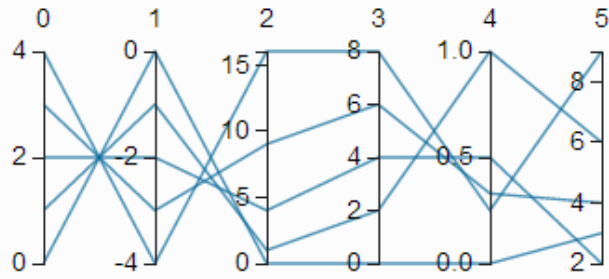
Quelle: <http://www.bbc.co.uk/news/business-15748696>

# D3.js: Parallel Coordinates



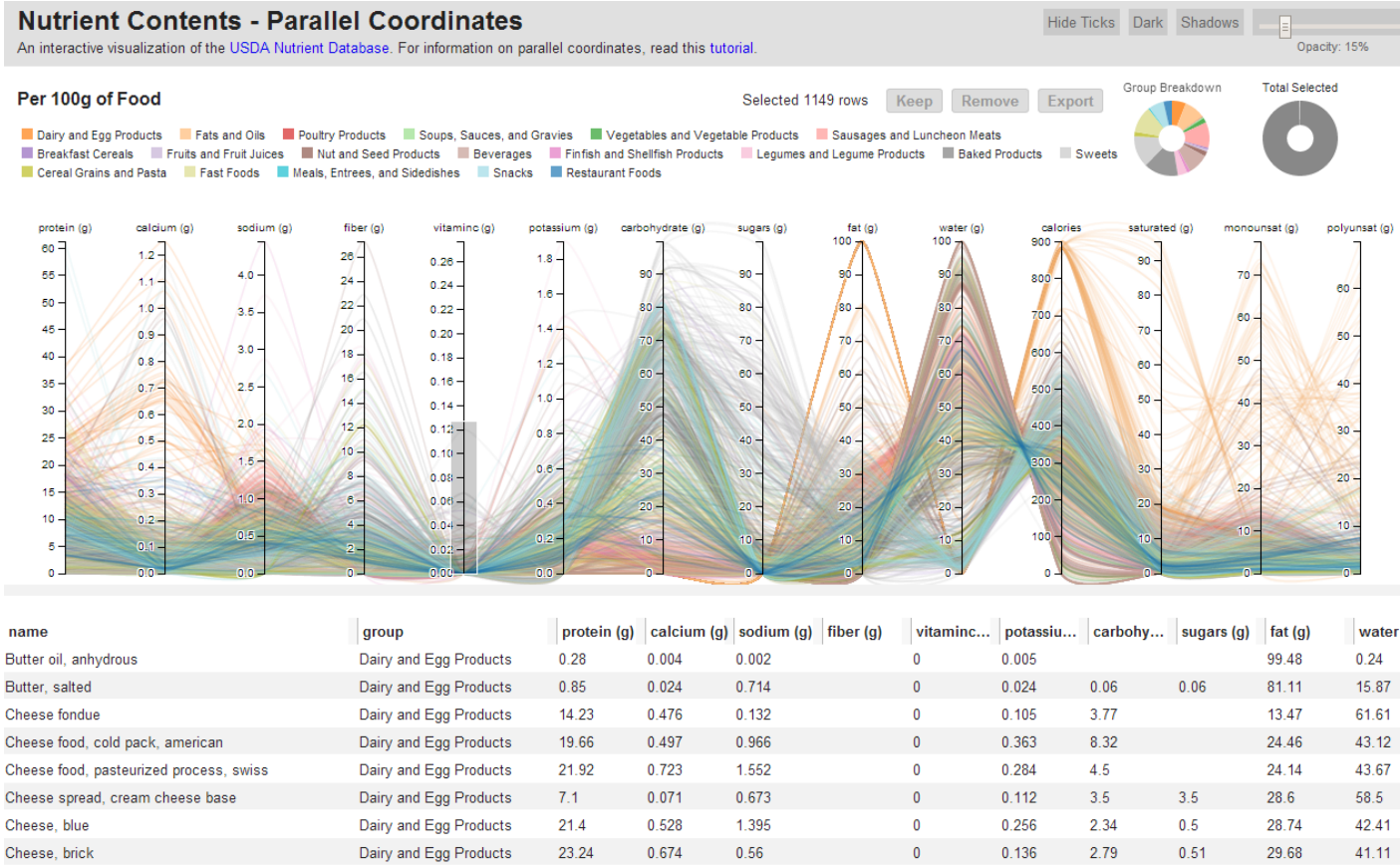
[Show code](#)

## Minimal Example



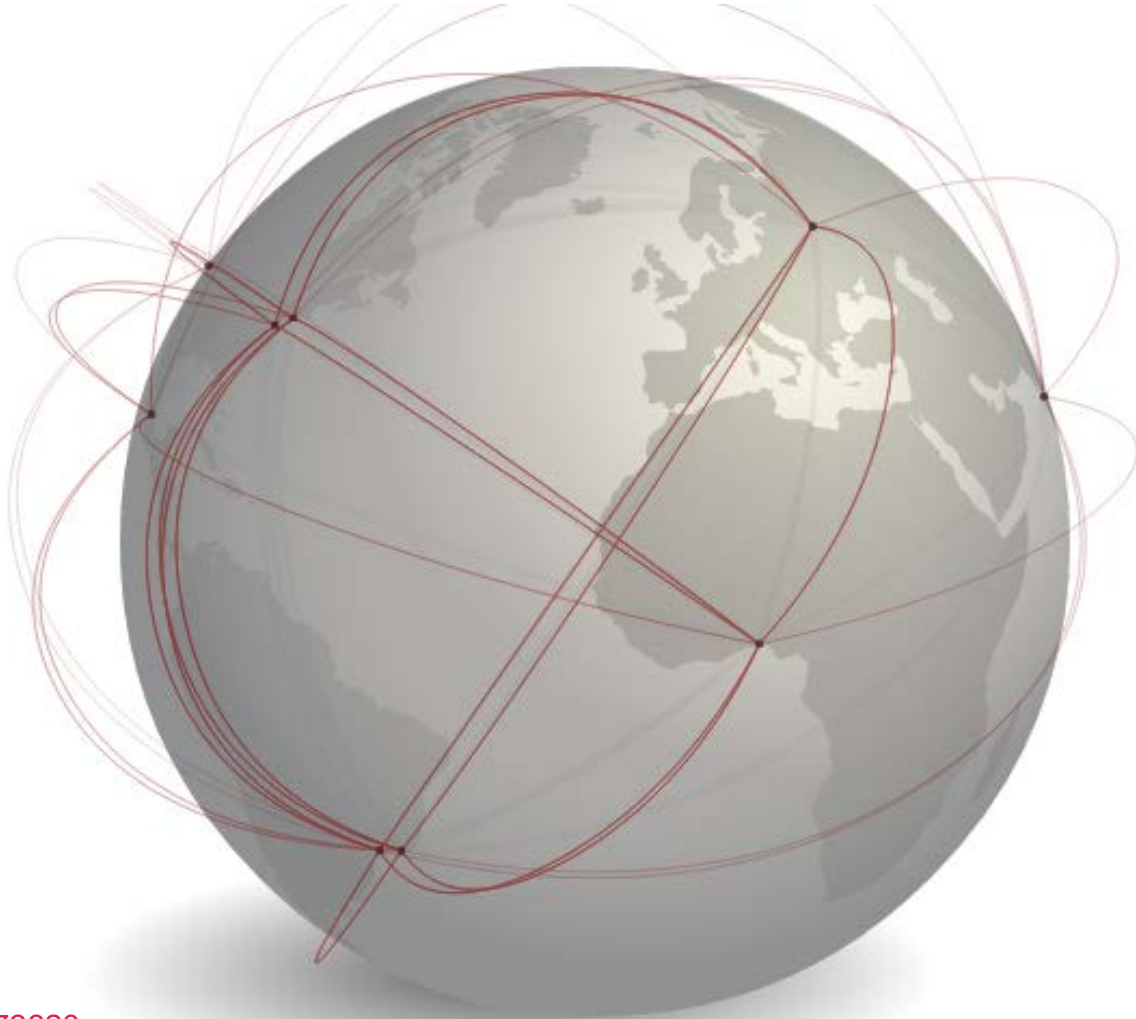
Quelle: <http://syntagmatic.github.io/parallel-coordinates/>

# D3: Nutrient Contents



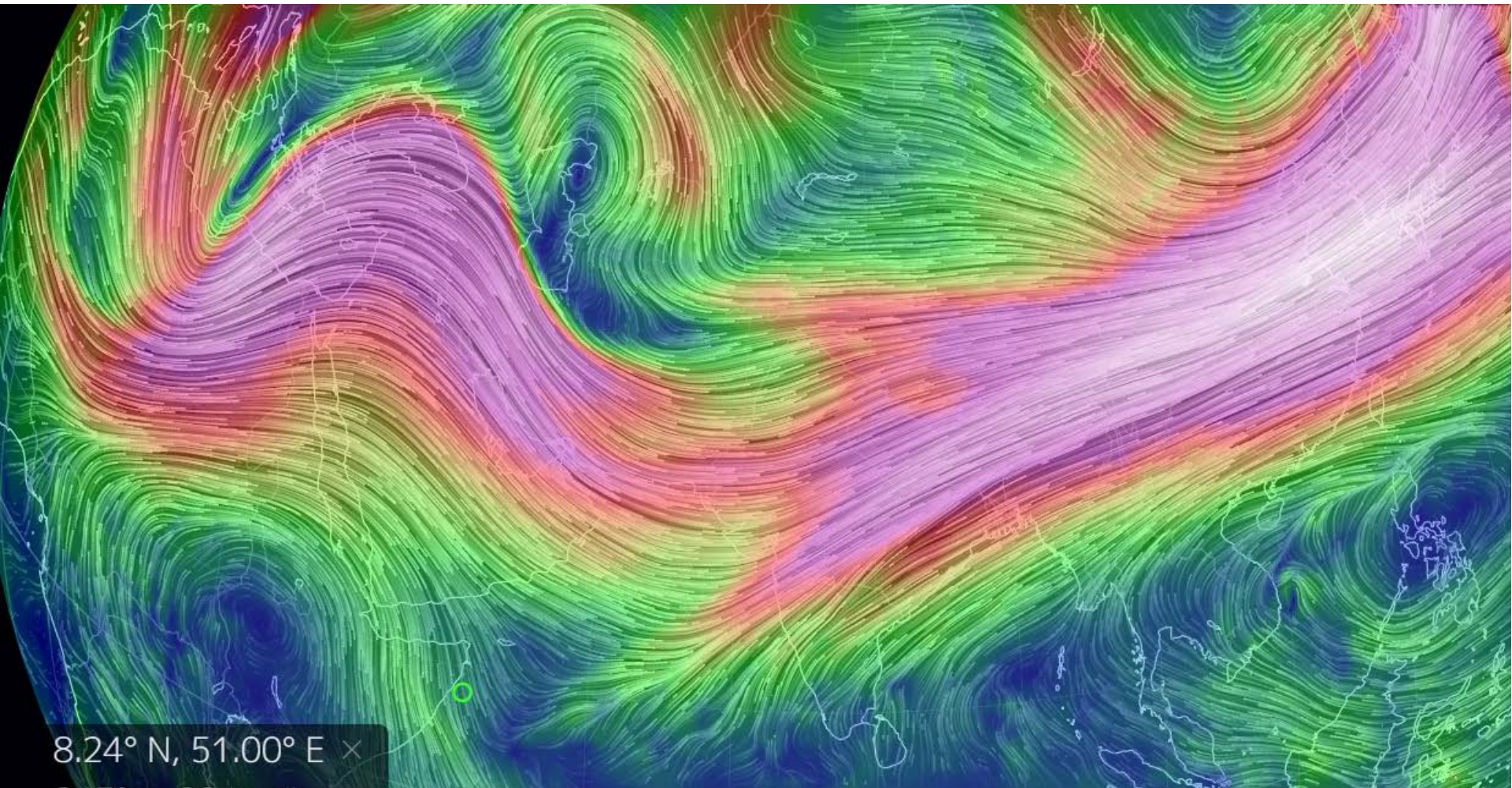
Quelle: <http://exposedata.com/parallel/>

# D3.js: Faux-3D Arcs





# Earth



8.24° N, 51.00° E ×  
345° @ 39 km/h



<http://www.three.js>



## A360

The  
**CHRISTMAS**  
EXPERIMENTS

One Day, One Creation

THE  
**HOBBIT**  
THE DESOLATION OF SMAUG  
A JOURNEY  
THROUGH  
**MIDDLE-EARTH**

iAd

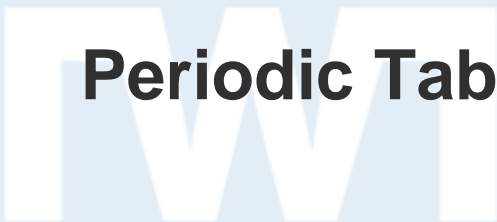
## London Olympics



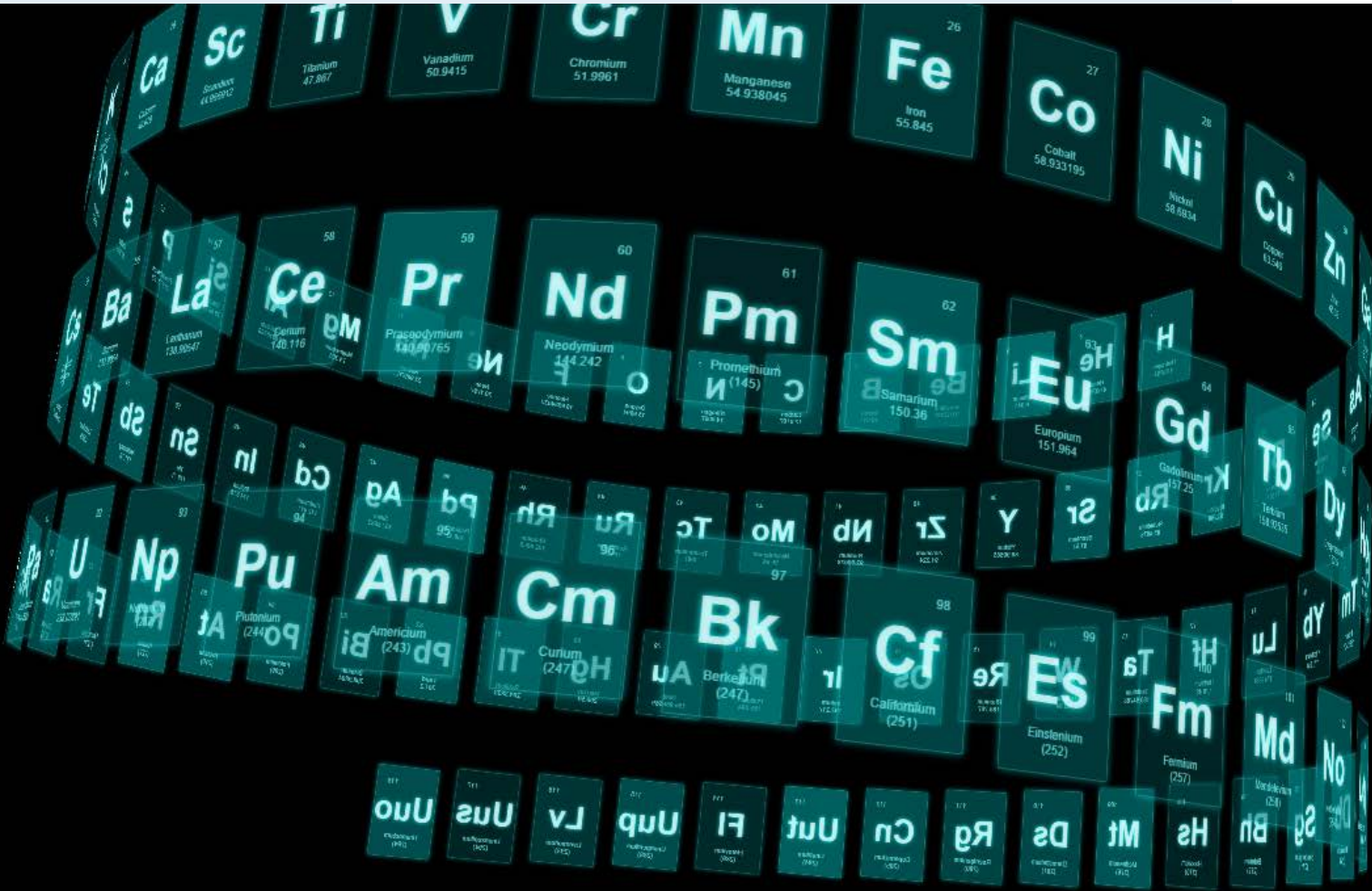
Ô Green

A photograph of the Space Shuttle Challenger in orbit above Earth. The shuttle is oriented vertically, with its nose pointing towards the top right. The orbiter is attached to the external tank and solid rocket boosters. The Earth's surface is visible in the background, showing a blue and white horizon. The shuttle's name, "CHALLENGER", is visible on the side of the orbiter. The image is a black and white photograph.





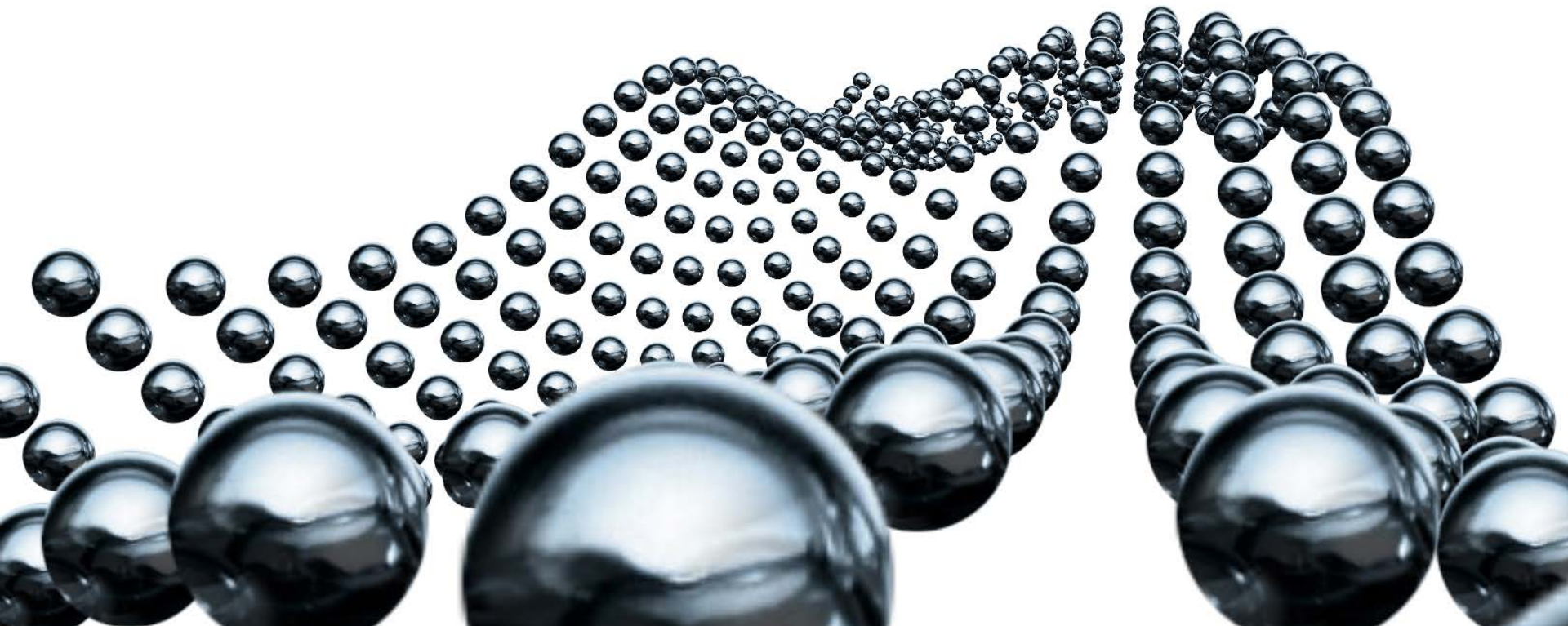
# Periodic Table





# Sprites

[http://threejs.org/examples/css3d\\_sprites.html](http://threejs.org/examples/css3d_sprites.html)



# Ocean Shaders



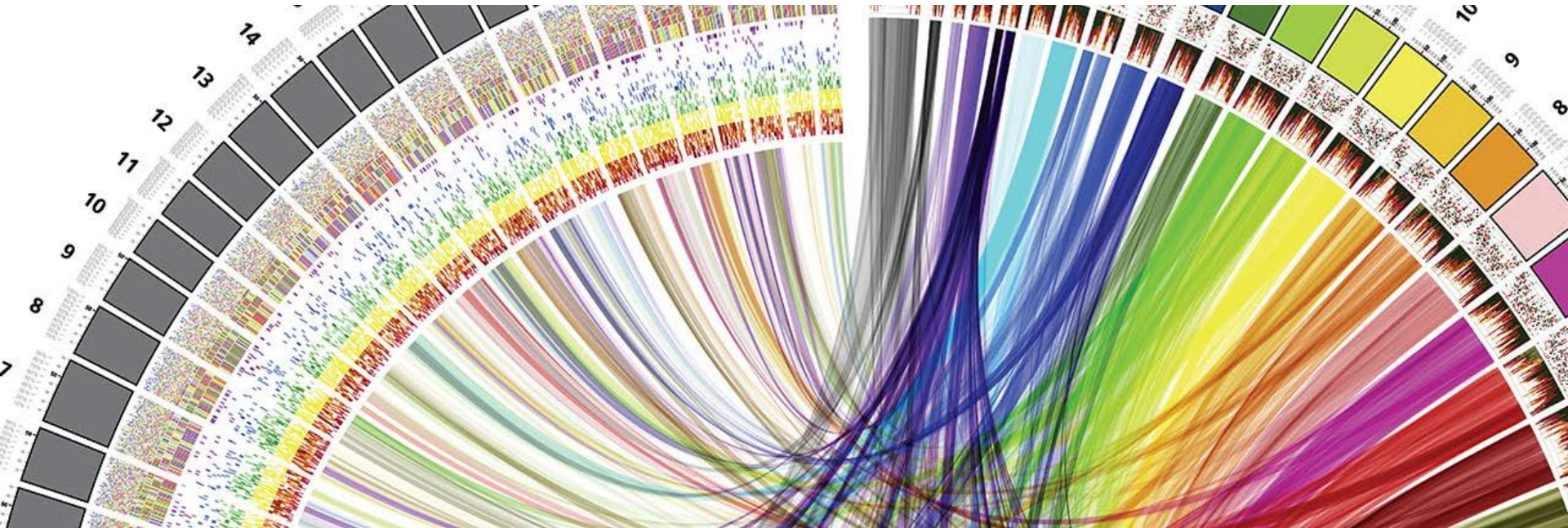
# Dynamic Terrain





# Agenda

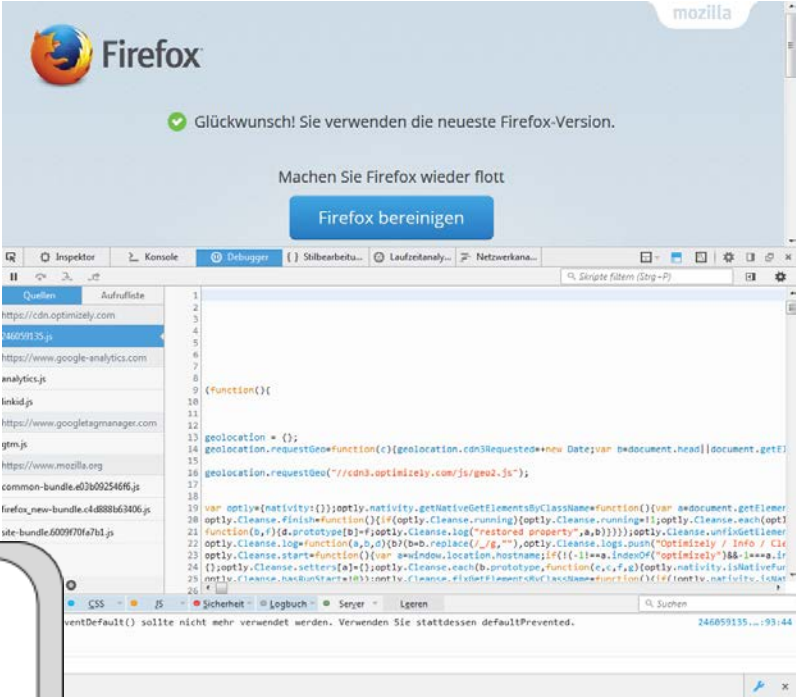
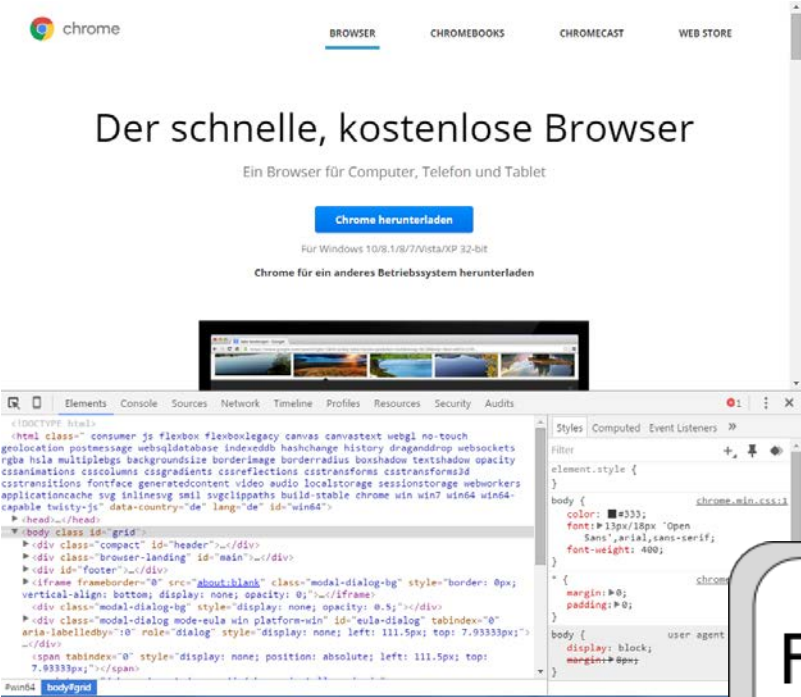
1. Organisatorisches zur App-Entwicklung
2. Interaktive Visualisierungen
3. **Einführung Entwickler-Tools**
4. App-Demo von zwei ehemaligen Studierenden
5. Fragen und Antworten



# Browser: Chrome oder Firefox

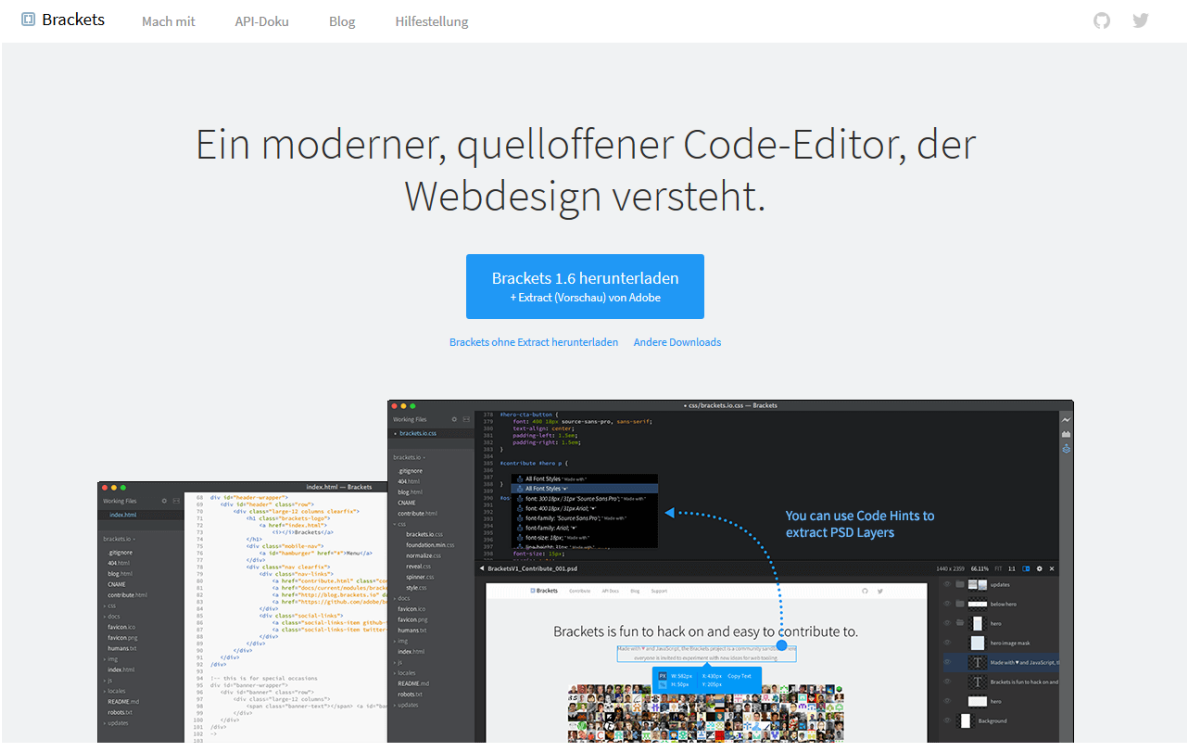
<https://www.google.com/chrome>

<https://www.mozilla.org/de/firefox/>



# Editor: Brackets

<http://brackets.io>



# IWI Sandbox

<https://sandbox.iwi.unibe.ch>

## Welcome to the IWI sandbox

Stürmer, Matthias (IWI)

Ready to deploy your application?

[Help?](#)

[Advanced users](#)

Sample applications:

- [d3-beatbox.zip](#)
- [d3-jeopardy.zip](#)
- [d3-switzerland.zip](#)
- [d3-worldmap-with-metadata.zip](#)
- [d3-worldmap.zip](#)
- [swissparliament.zip](#)

Drag'n Drop ZIP file here, or click to upload one

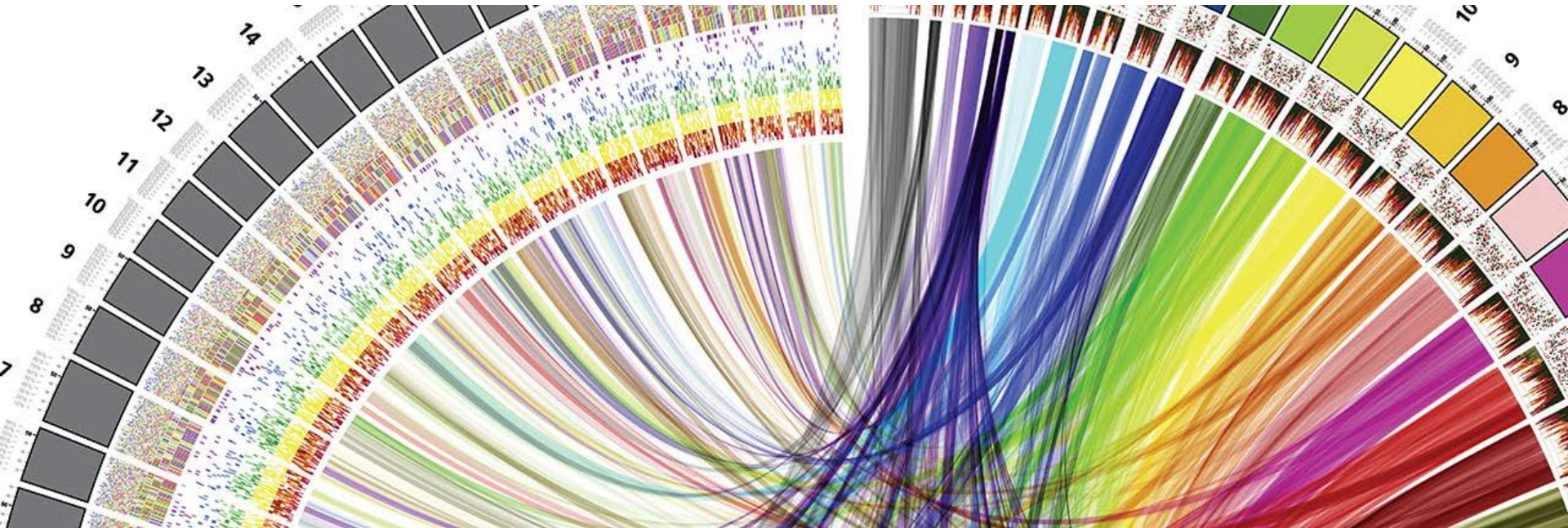
k-deploy - the worlds easiest to use [PaaS](#). by [Khôi Tran](#)

Based [dokku](#) and [buildstep](#). Implemented in [node.js](#). Inspired by [Heroku](#)



# Agenda

1. Organisatorisches zur App-Entwicklung
2. Interaktive Visualisierungen
3. Einführung Entwickler-Tools
4. **App-Demo von zwei ehemaligen Studierenden**
5. Fragen und Antworten



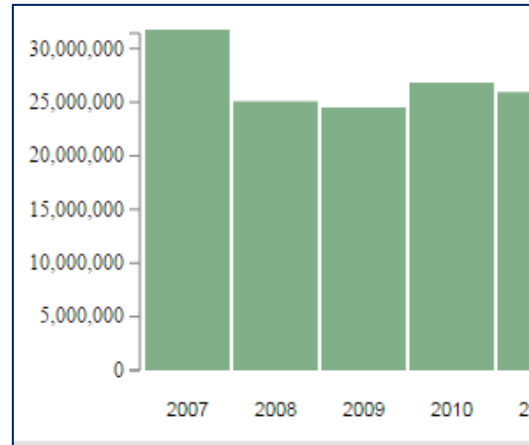
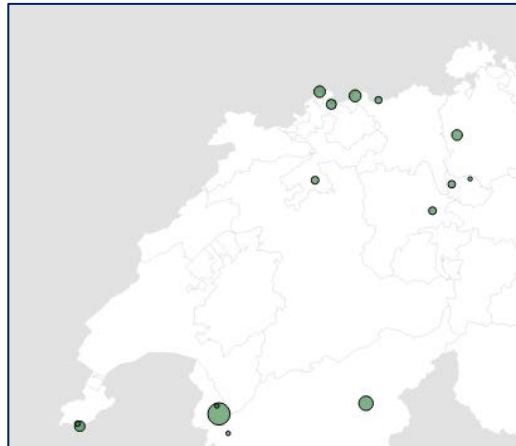


# Beispielprozess der App Entwicklung: Ziel

- > Zusammenarbeit mit dem BAFU
- > Visualisierung des Schadstoffregisters der Schweiz  
<https://www.prtr.admin.ch/>
- > «Grösste Schadstoffsünder» sichtbar machen
- > Partnerarbeit

# Beispielprozess der App Entwicklung: Vorgehen

1. Lernen von D3.js anhand der Vorlesung und über das [Onlinebuch](#) von Oreilly
2. Wählen einer geeigneten Visualisierung:

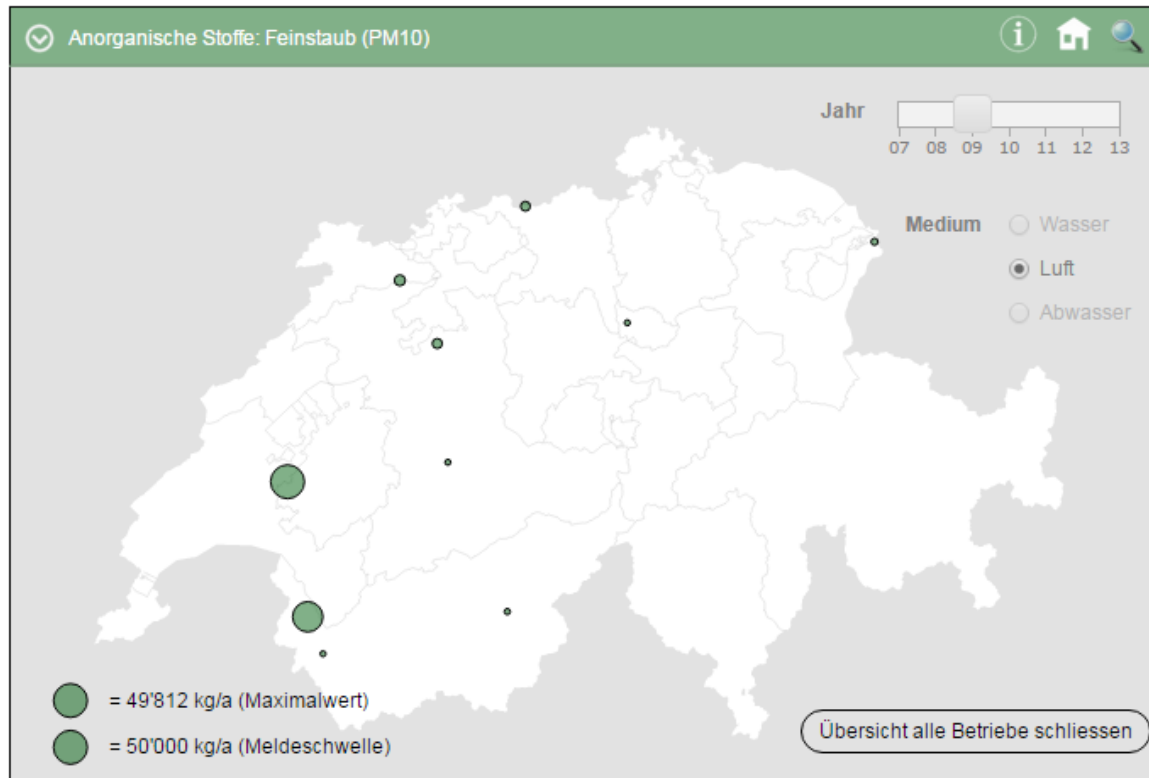


3. Programmierung und Präsentation der Applikation

# Beispielprozess der App Entwicklung: Weiterer Verlauf

- > BAFU zeigte auch danach Interesse an der Applikation
- > Anstellung an der Uni um finale Version des Schadstoffregisters zu erstellen
- > Anfang März 2016: Live-Schaltung auf der BAFU Webseite

# Schadstoffregister



> <http://schadstoffregister.opendata.iwi.unibe.ch/>

# Agenda

1. Organisatorisches zur App-Entwicklung
2. Interaktive Visualisierungen
3. Einführung Entwickler-Tools
4. App-Demo von zwei ehemaligen Studierenden
5. **Fragen und Antworten**

