



h_da

HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES

D3 EINSTEIGER

Claudius Coenen

Modus

- Bei Fragen einfach sofort melden
- Bremst mich, wenn ich zu schnell werde
- JavaScript-Kenntnisse (und etwas HTML) sind voraus gesetzt
- Bitte gebt im Anschluss Feedback (via MRMCD-Fahrplan!)
- Folien, Materialien:
 - <https://goo.gl/6nsK8G> bzw.
 - <https://www.claudiuscoenen.de/mrmcd/> (eigentliche Adresse)





Claudius
amenthes



ccoenen



amenthes_de



amenthes@chaos.social



ccc@amenthes.de

VISUALISIERUNG - BEISPIELE -



„Unischanke“

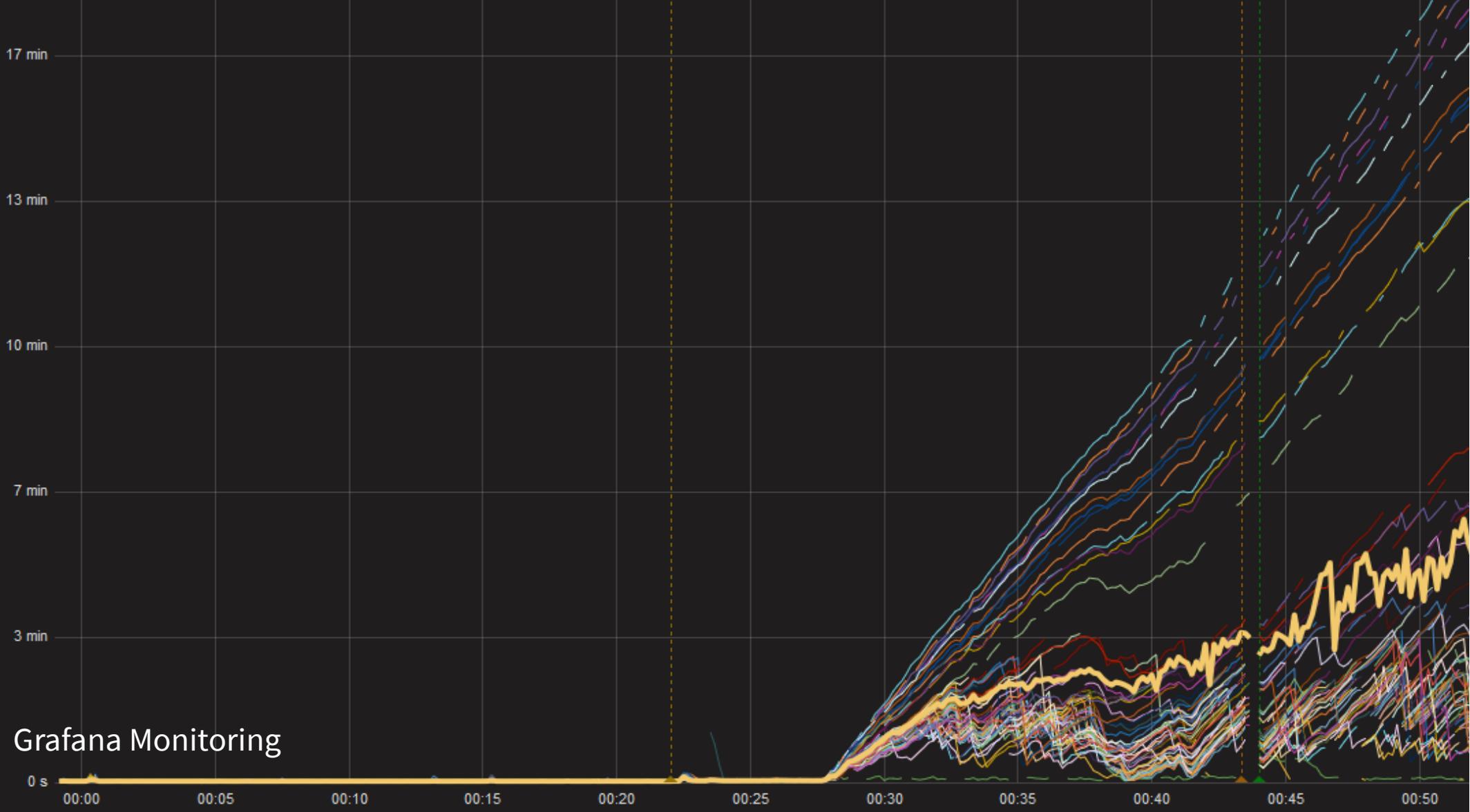
Cholera Map (1845)

Dr. John Snow

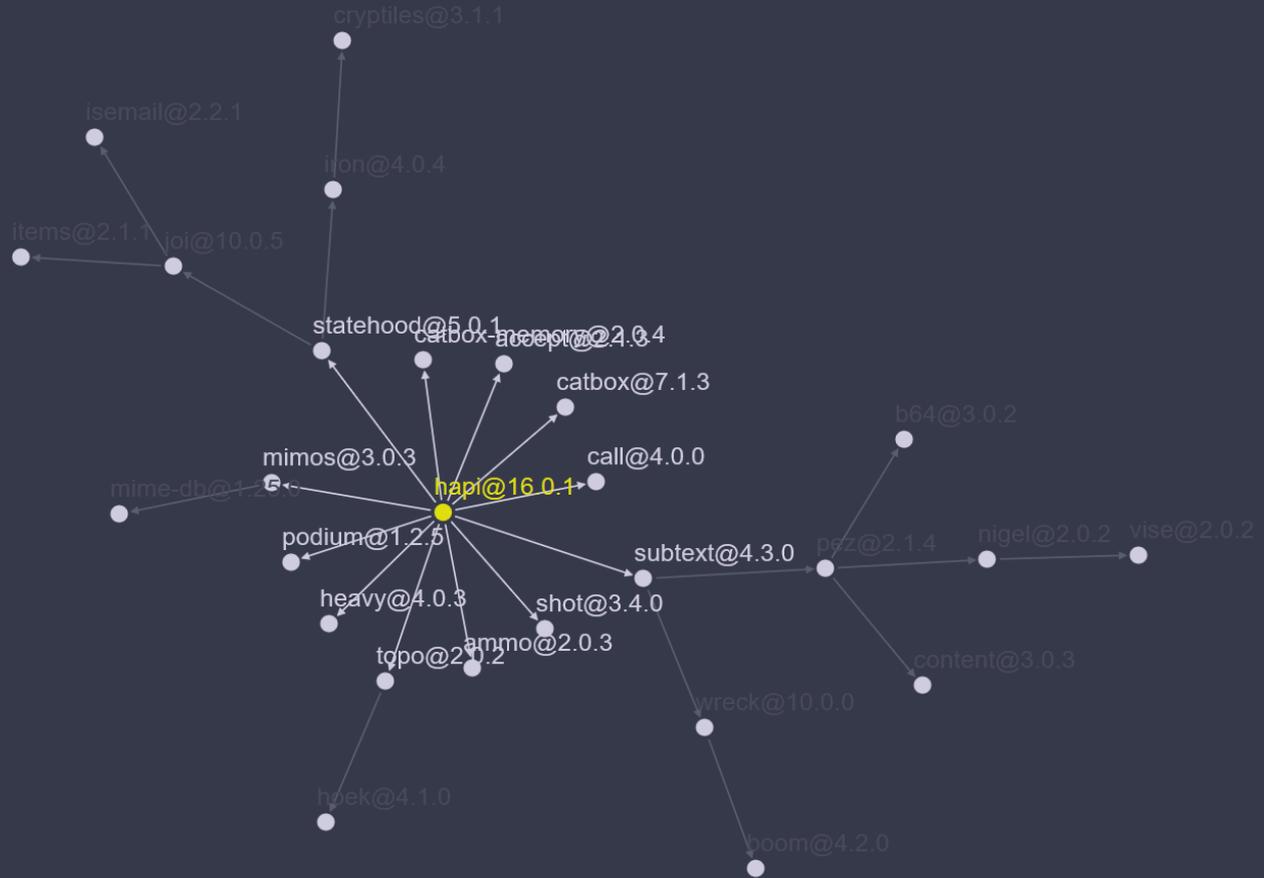
(kein Witz)



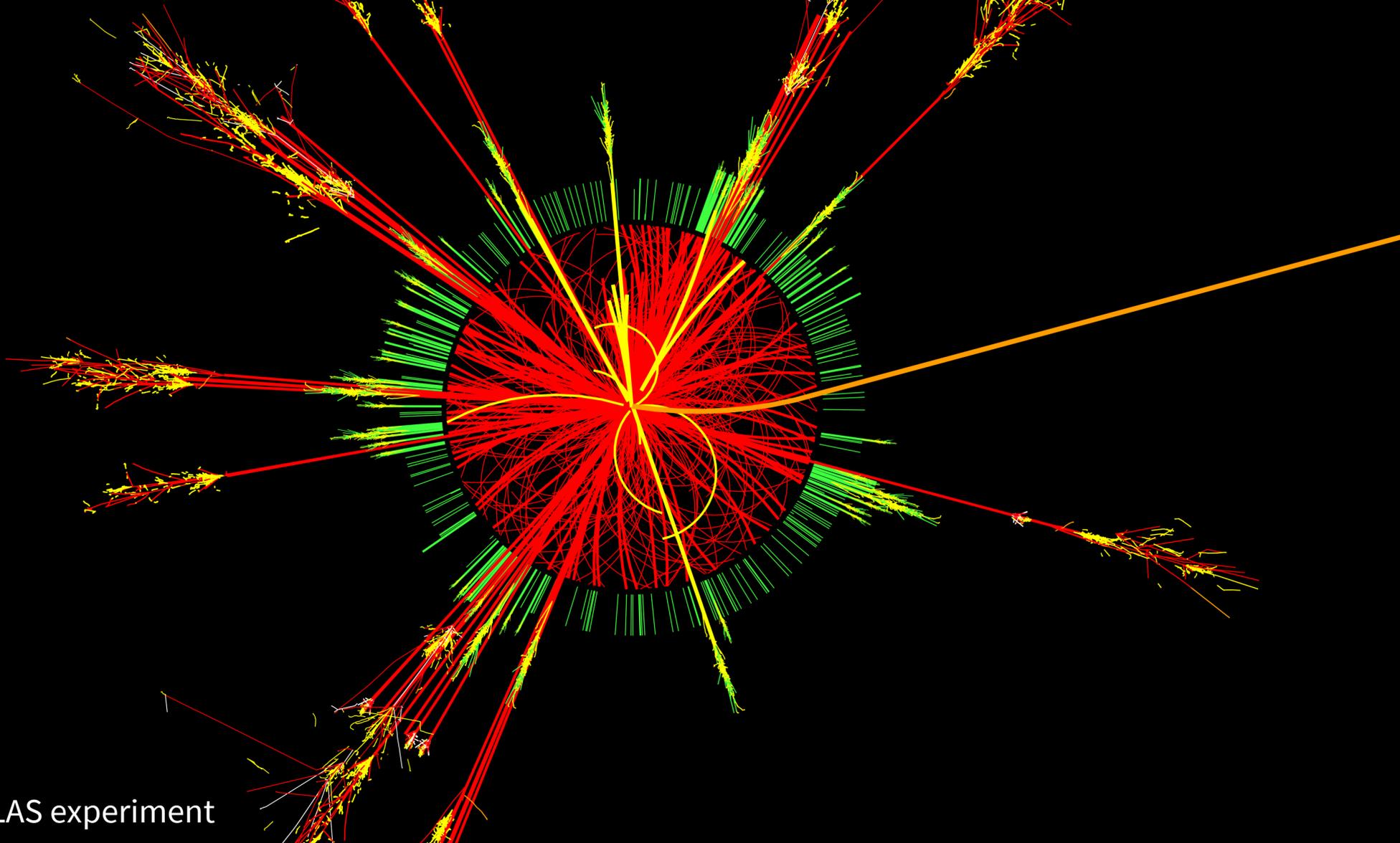
```
D, [2016-12-06T06:26:51.347102 #17394] DEBUG -- : [ci-#1627] ResponseMessage (12B) received Response Messa
D, [2016-12-06T06:26:51.353770 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:26:51.429637 #17394] DEBUG -- : timestamp: 1481005610, summary_cts (A): [0.114, 0.0, 0.0
D, [2016-12-06T06:26:51.437677 #17394] DEBUG -- : Staging request: successful.
D, [2016-12-06T06:26:59.733417 #17394] DEBUG -- : [ci-#1627] Queue status [in: 0 messages / out: 0 message
D, [2016-12-06T06:26:59.733645 #17394] DEBUG -- : [ci-#1627] queueing PingMessage (8B) {:payload=>""}
D, [2016-12-06T06:27:01.342444 #17394] DEBUG -- : [ci-#1627] ResponseMessage (12B) received Response Messa
D, [2016-12-06T06:27:01.356226 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:27:01.728317 #17394] DEBUG -- : timestamp: 1481005620, summary_cts (A): [0.142, 0.001, 0
D, [2016-12-06T06:27:01.818153 #17394] DEBUG -- : Staging request: successful.
D, [2016-12-06T06:27:11.360332 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:27:11.418610 #17394] DEBUG -- : timestamp: 1481005630, summary_cts (A): [0.14, 0.001, 0.
D, [2016-12-06T06:27:11.427897 #17394] DEBUG -- : Staging request: successful.
D, [2016-12-06T06:27:15.019493 #17394] DEBUG -- : [ci-#1627] Queue status [in: 0 messages / out: 0 message
D, [2016-12-06T06:27:15.019696 #17394] DEBUG -- : [ci-#1627] queueing PingMessage (8B) {:payload=>""}
D, [2016-12-06T06:27:21.390870 #17394] DEBUG -- : [ci-#1627] ResponseMessage (12B) received Response Messa
D, [2016-12-06T06:27:21.411308 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:27:21.934151 #17394] DEBUG -- : timestamp: 1481005640, summary_cts (A): [0.161, 0.002, 0
D, [2016-12-06T06:27:21.977745 #17394] DEBUG -- : Staging request: successful.
D, [2016-12-06T06:27:30.023546 #17394] DEBUG -- : [ci-#1627] Queue status [in: 0 messages / out: 0 message
D, [2016-12-06T06:27:30.023817 #17394] DEBUG -- : [ci-#1627] queueing PingMessage (8B) {:payload=>""}
D, [2016-12-06T06:27:31.383844 #17394] DEBUG -- : [ci-#1627] ResponseMessage (12B) received Response Messa
D, [2016-12-06T06:27:31.389388 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:27:31.456762 #17394] DEBUG -- : timestamp: 1481005650, summary_cts (A): [0.15, 0.001, 0.
D, [2016-12-06T06:27:31.470861 #17394] DEBUG -- : Staging request: successful.
D, [2016-12-06T06:27:41.326596 #17394] DEBUG -- : [ci-#1627] StreamDataMessage2 (208B) received {:timestam
D, [2016-12-06T06:27:41.657801 #17394] DEBUG -- : timestamp: 1481005660, summary_cts (A): [0.143, 0.001, 0
```



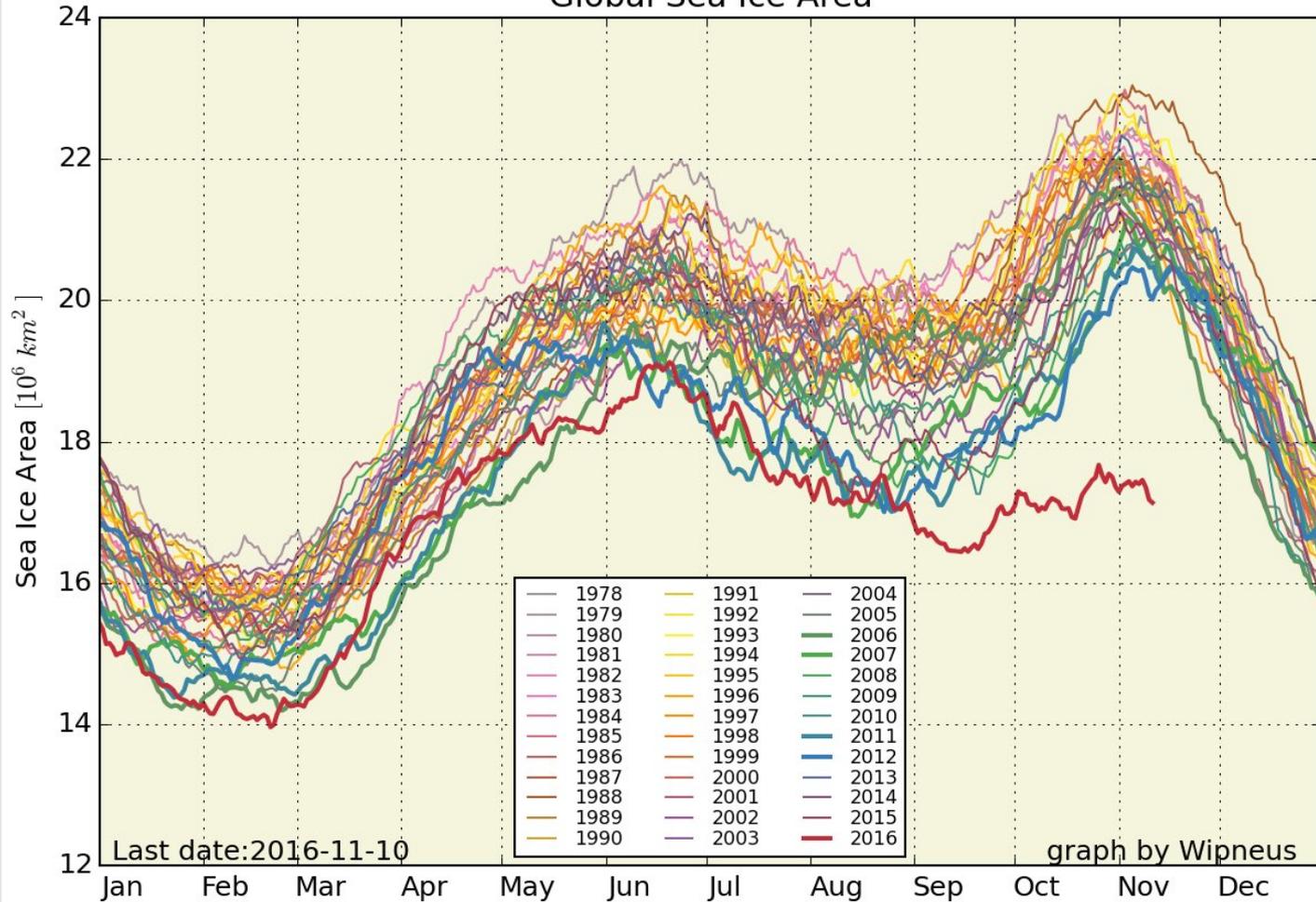
Grafana Monitoring



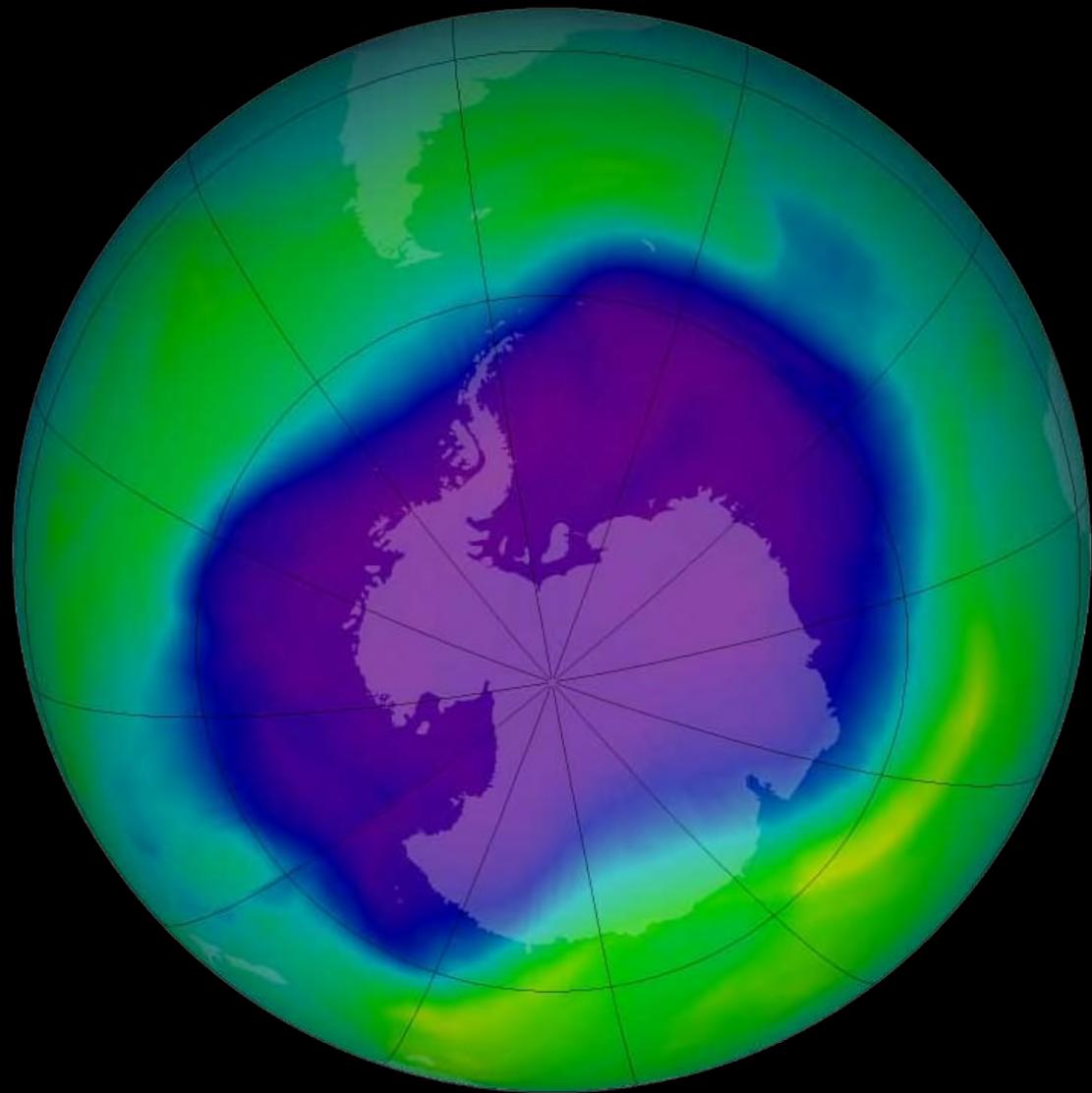
LHC ATLAS experiment



from NSIDC sea ice concentration data
Global Sea Ice Area



Globale Erwärmung



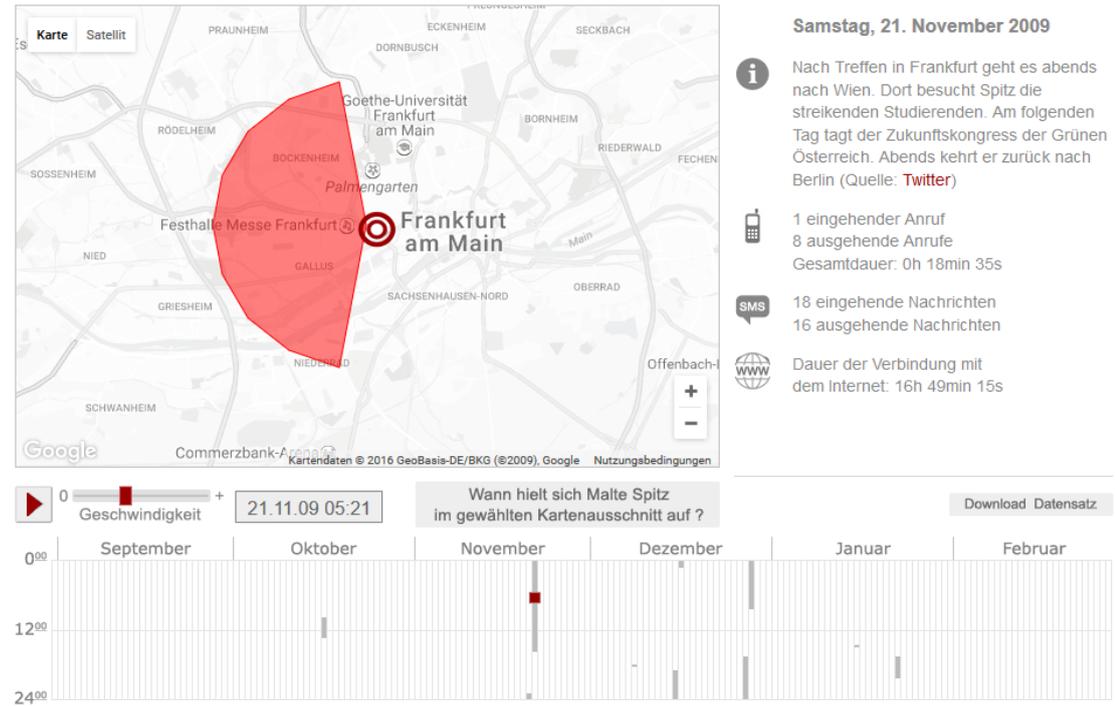
Ozonloch

Verräterisches Handy

deutsch | english

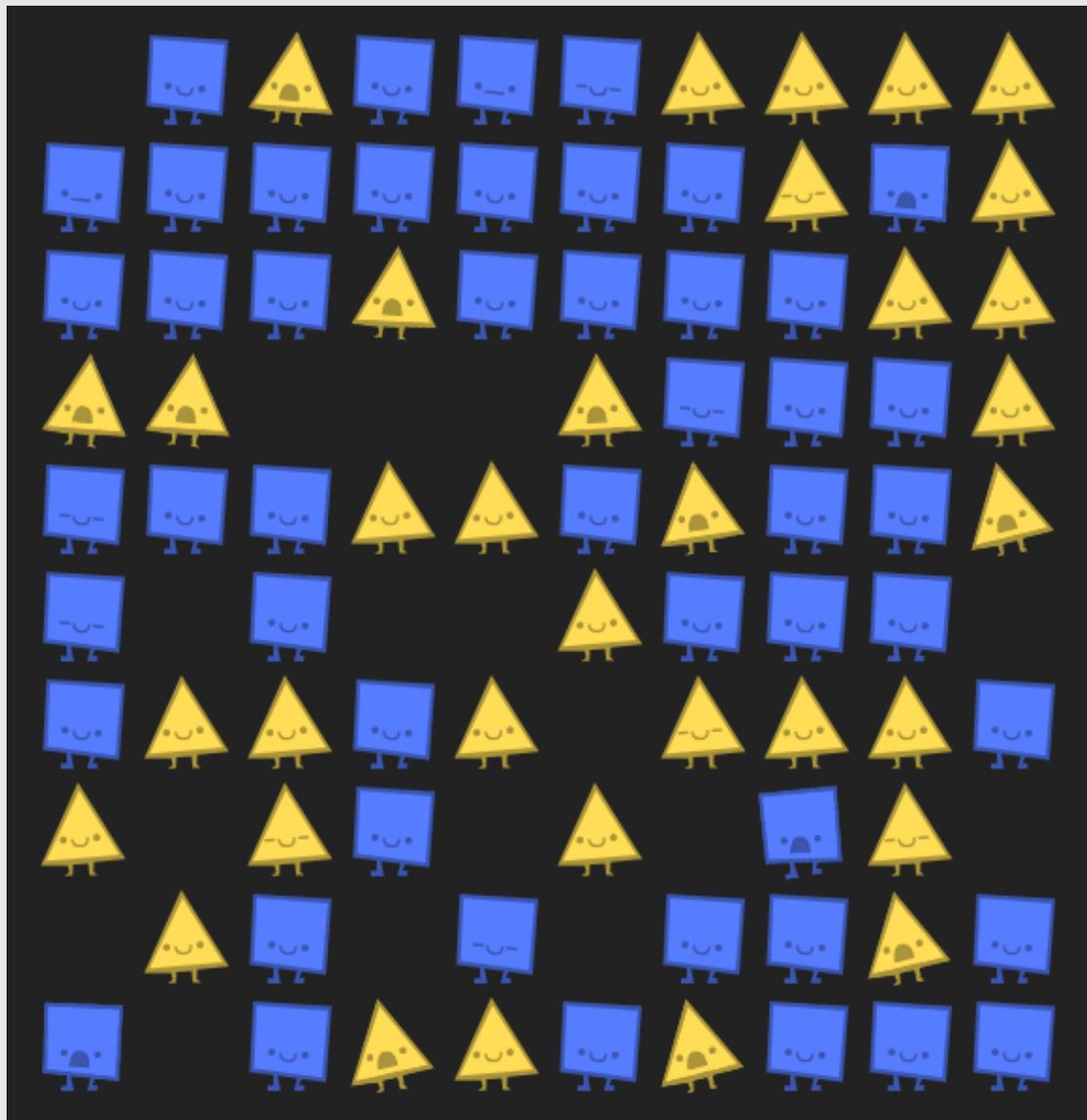
Sechs Monate seiner Vorratsdaten hat der Grünenpolitiker Malte Spitz von der Telekom eingeklagt und ZEIT ONLINE zur Verfügung gestellt. Auf Basis dieser Daten können Sie all seine Bewegungen dieser Zeit nachvollziehen. Die Geodaten haben wir zusätzlich mit frei im Netz verfügbaren Informationen aus dem Leben des Abgeordneten ([Twitter](#), [Blogeinträge](#) und [Webseiten](#)) verknüpft.

Mit der Play-Taste startet die Reise durch Malte Spitz' Leben. Über den Geschwindigkeitsregler können Sie das Tempo anpassen oder an beliebigen Punkten mit der Pause-Taste anhalten. Zusätzlich zeigt der darunter stehende Kalender, wann er noch an diesem Ort war – gleichzeitig kann darüber jeder beliebige Zeitpunkt angesteuert werden. Jede der vertikalen Spalten entspricht einem Tag.



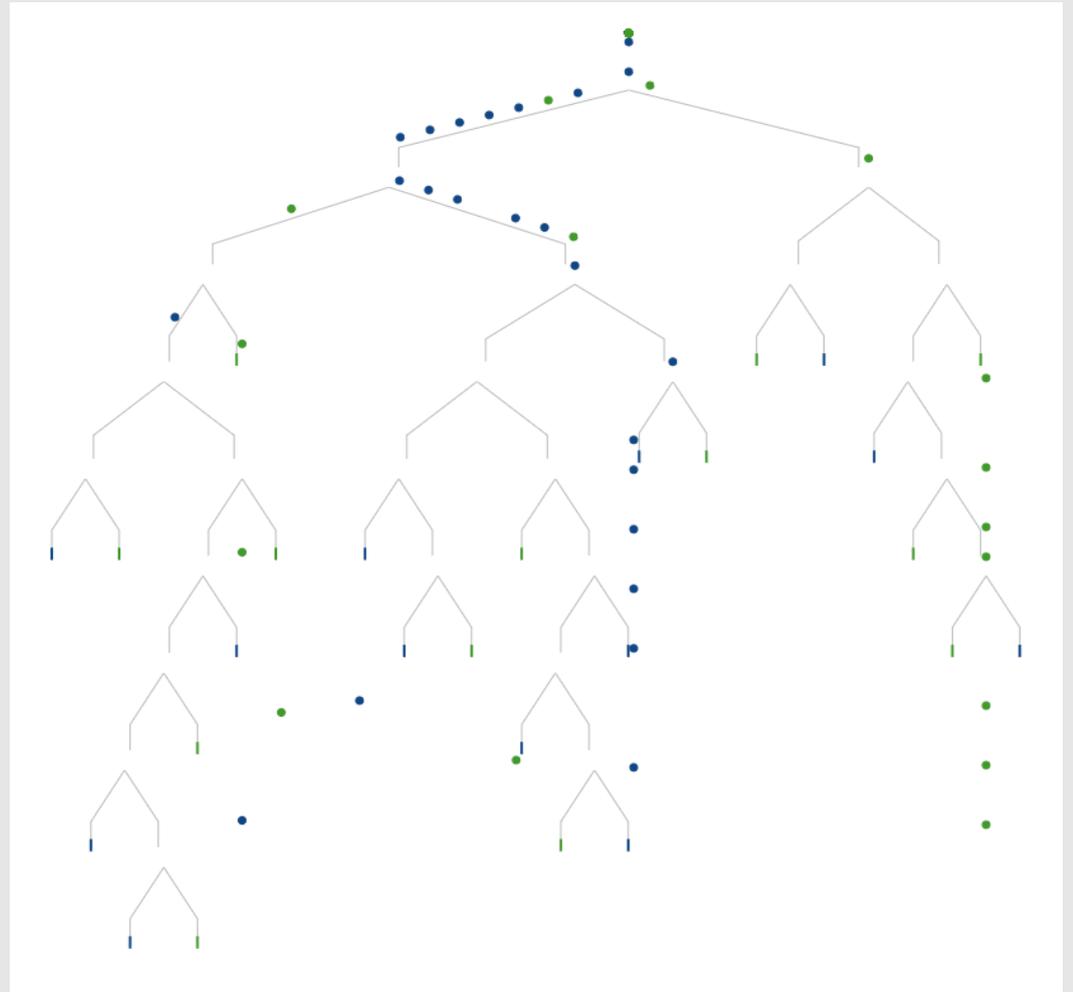
Vorratsdaten des Malte Spitz

<http://www.zeit.de/datenschutz/malte-spitz-vorratsdaten>



Parabel der Polygone
<http://ncase.me/polygons-de/>

Evolution of Trust
<http://ncase.me/trust/>

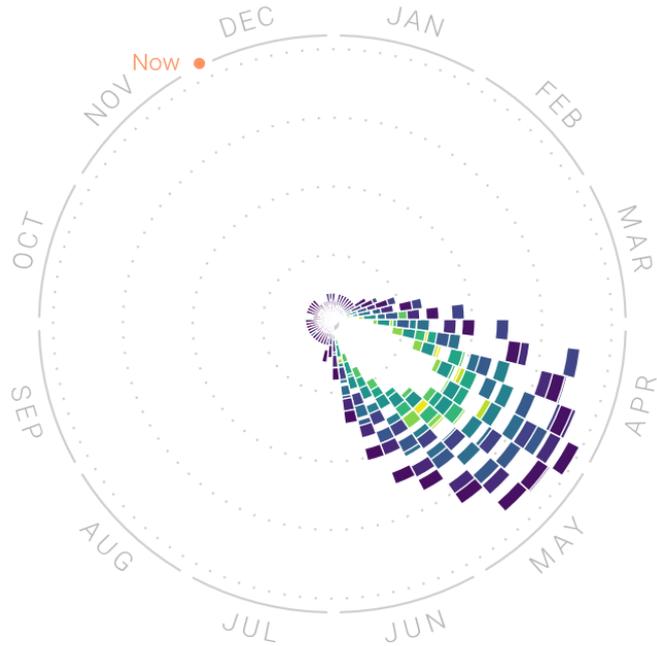


Introduction to Machine Learning

<http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>

2004

2016



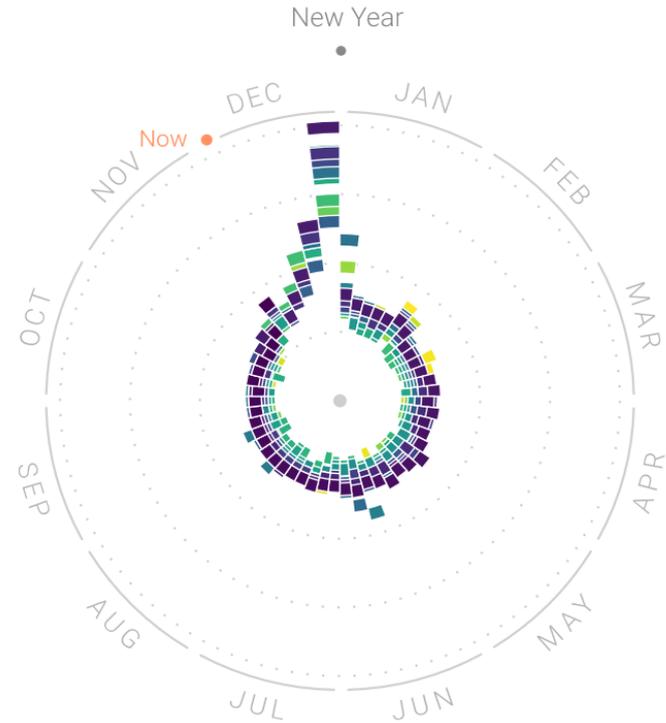
ASPARAGUS (GERMANY)

Play!



2004

2016



CHAMPAGNE

Play!

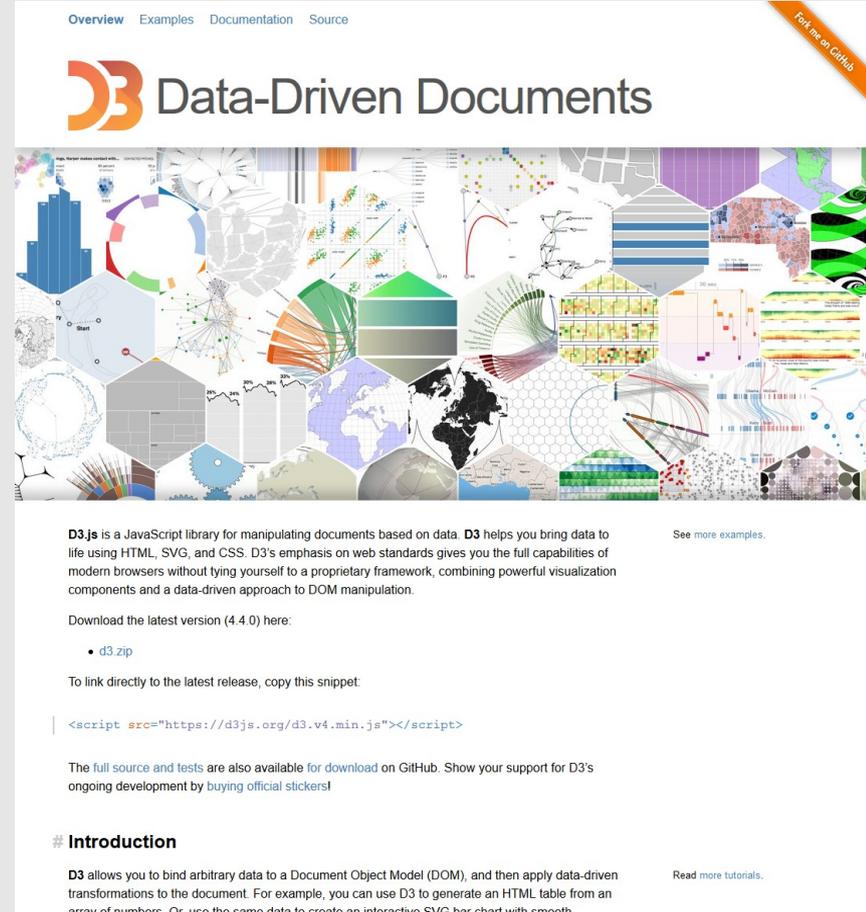


VISUALISIERUNG
- MINIPROJEKT TEIL 1 -

Benutzte Bibliothek: D3.js

- Spezialisiert auf interaktive Anwendungen, denen große Datenmengen zugrunde liegen
- Kommt aus dem Umfeld der New York Times
- Sehr mächtig, aber auch komplex
- Ausführlich dokumentiert, viele Beispiele
- <https://d3js.org/> (offizielle Website)
- <http://christopheviau.com/d3list/gallery.html> (Beispiele)

Achtung: Version 3 und Version 4 unterscheiden sich stark!



Overview Examples Documentation Source

Fork me on GitHub

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.

Download the latest version (4.4.0) here:

- [d3.zip](#)

To link directly to the latest release, copy this snippet:

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

The full source and tests are also available for download on GitHub. Show your support for D3's ongoing development by buying official stickers!

Introduction

D3 allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth

D3-API: `.select()` und `.append()`

- `select` wählt ein Element aus dem HTML- oder SVG-Dokument aus
- `selectAll` wählt mehrere gleichartige Elemente aus
- `append` fügt einem Element ein Unter-Element hinzu
- Beides returned eine „d3 selection“, mit der ich weiter arbeiten kann

```
// wir holen uns das <body>-Tag  
var firstElement = d3.select("body");
```

```
// und hängen ein neues <div>-Tag hinein  
var anotherElement = firstElement.append("div");
```

D3-API: `.text()`, `.attr()` und `.style()`

- Sobald Sie eine Selection haben, können Sie sie verändern.
- Es gibt einige Helfer, die wichtigsten sind
 - `text()` - verändert den Text-Inhalt des Elements
 - `attr()` - verändert Attribute des Elements
 - `style()` - verändert das Style-Attribut gezielt
- Alle diese Funktionen können Werte oder Callbacks entgegen nehmen.

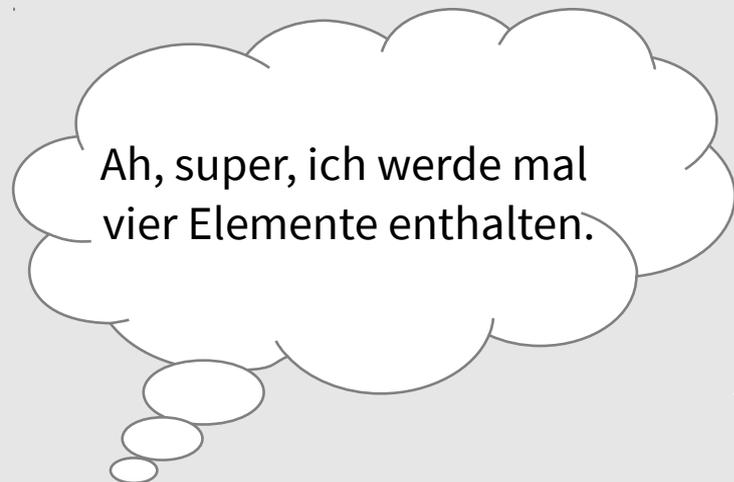


Dazu später mehr!

```
// fester Wert - ein immer gleicher Text:  
d3.select("body").text("Ohai!");
```

D3-API: `.data()` Joins

- Verknüpft Daten mit einer gesamten Selection
- Normalfall: Selection enthält Unter-Elemente für jeden Datensatz



```
var flats = [  
  {id: 1, title: "a"},  
  {id: 2, title: "b"},  
  {id: 3, title: "c"},  
  {id: 4, title: "d"}  
];
```

`var items = anotherElement.selectAll("p").data(flats);`

D3-API: `.enter()`, `.exit()`

brauchen wir noch nicht, gehört aber zusammen!

- Enter sorgt dafür, dass die richtige Anzahl Elemente vorhanden ist

```
items..enter().append("p");  
// erzeugt nicht 1 Element, sondern  
// so viele wie nötig!
```

D3-API: Daten-Joins und Daten-Callbacks

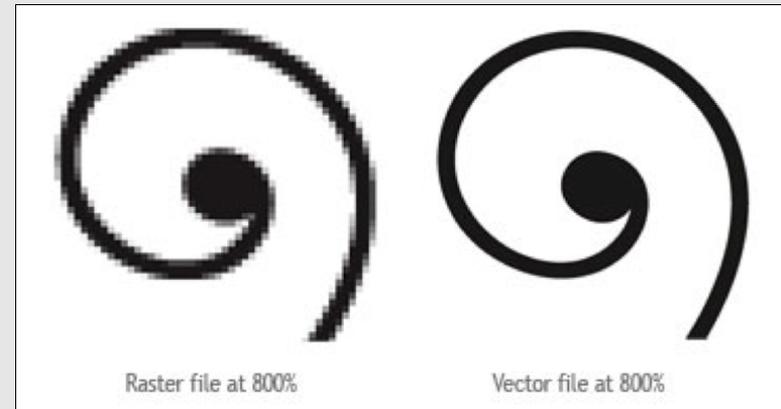
- In einer gejointen Selection kann jedes Element auf seine eigenen Daten zugreifen.
- Dazu ist ein Callback notwendig

```
items.enter().append("p")  
  .text(function (d) { return d.title; });
```

VISUALISIERUNG
- MINIPROJEKT TEIL 2 -

Was ist SVG?

- Vektor-Grafik im Browser
- Liniengrafiken, Text und einfache Formen
- Lässt sich handhaben wie herkömmliche Bilder
- Lässt sich zusätzlich interaktiv gestalten
- XML-Syntax (ähnlich wie HTML)



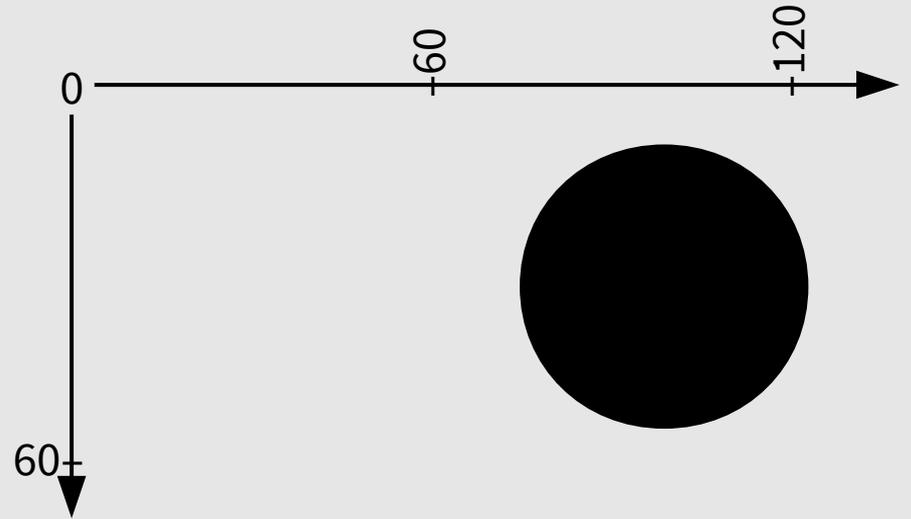
SVG-Crashkurs: `circle`-Element

- Attribute

- `cx`
- `cy`
- `r`
- (+ Standard-Attribute)

- Code-Beispiel:

```
<circle cx="90" cy="30" r="25"></circle>
```



SVG-Crashkurs: g-Element

- Gruppierung
vgl. `<div>` in HTML
vgl. Gruppen in Grafikprogrammen
- Gruppe selbst ist unsichtbar, nur der Inhalt ist sichtbar
- Code-Beispiel:

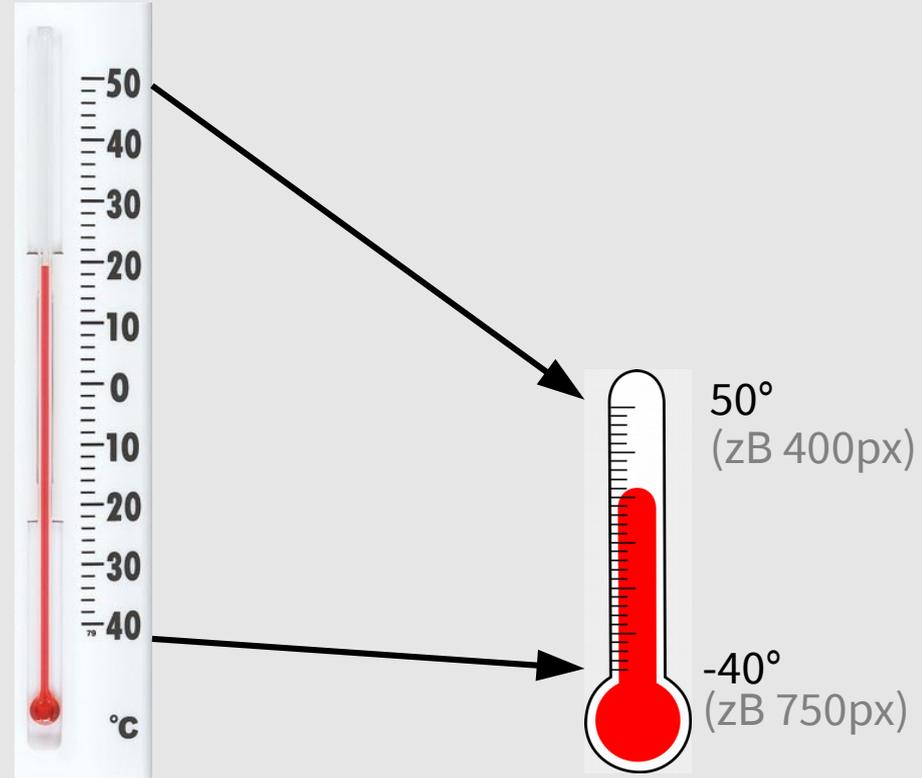
```
<g>  
  <circle cx="20" cy="15" r="15"></circle>  
  <circle cx="90" cy="14" r="25"></circle>  
  <circle cx="50" cy="75" r="50"></circle>  
</g>
```

Weitere SVG-Elemente

- Siehe „Basic Shapes“ Spezifikation
 - z.B. rect (Attribute: x, y, width, height)
- Siehe „Path“ Spezifikation
 - Komplexe Formen zeichnen mit LineTo/MoveTo
- Siehe „Text“ Spezifikation

D3 Begriffe: Scale

- Eine Scale sorgt für die Umrechnung von Daten zu Bildschirm-Koordinaten
- Viele Typen, am wichtigsten: „Linear“ und „Ordinal“
- **Domain:** Die Sprache ihrer Daten
- **Range:** Die Koordinaten auf dem Bildschirm



Echte Welt bzw. Daten

Visualisierung

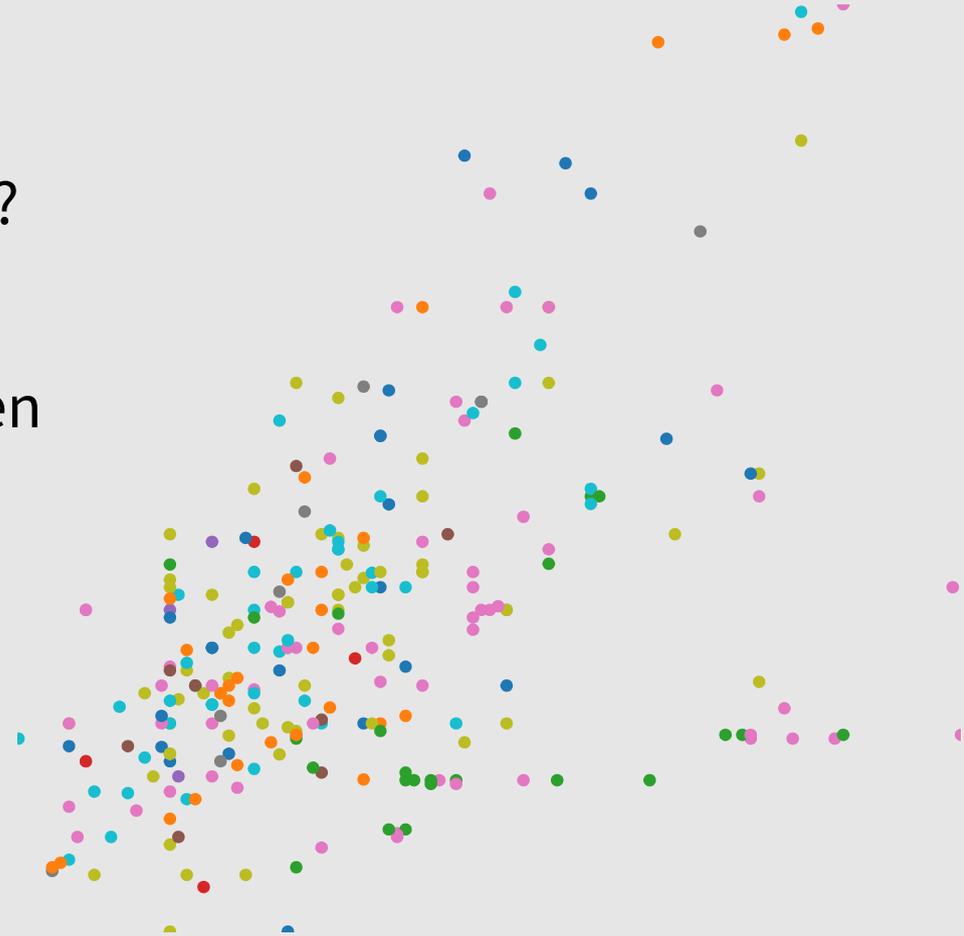
Daten: Mietwohnungen in Darmstadt

- 274 Wohnungen
- Januar/Februar 2016
- Leider teilweise unvollständig
- Eckdaten wie
 - Lage
 - Fläche, Stockwerk, Ausstattung
 - Kosten

```
{  
  id: 1,  
  title: "Schöne helle 4 Zim..",  
  district: "Nord",  
  latitude: 49.8795,  
  longitude: 8.6651,  
  rooms: "4",  
  area: 105,  
  level: "EG",  
  balcony: true,  
  bathtub: false,  
  baseRent: 1400,  
  additionalCosts: 150,  
  heatingCosts: 100000,  
}
```

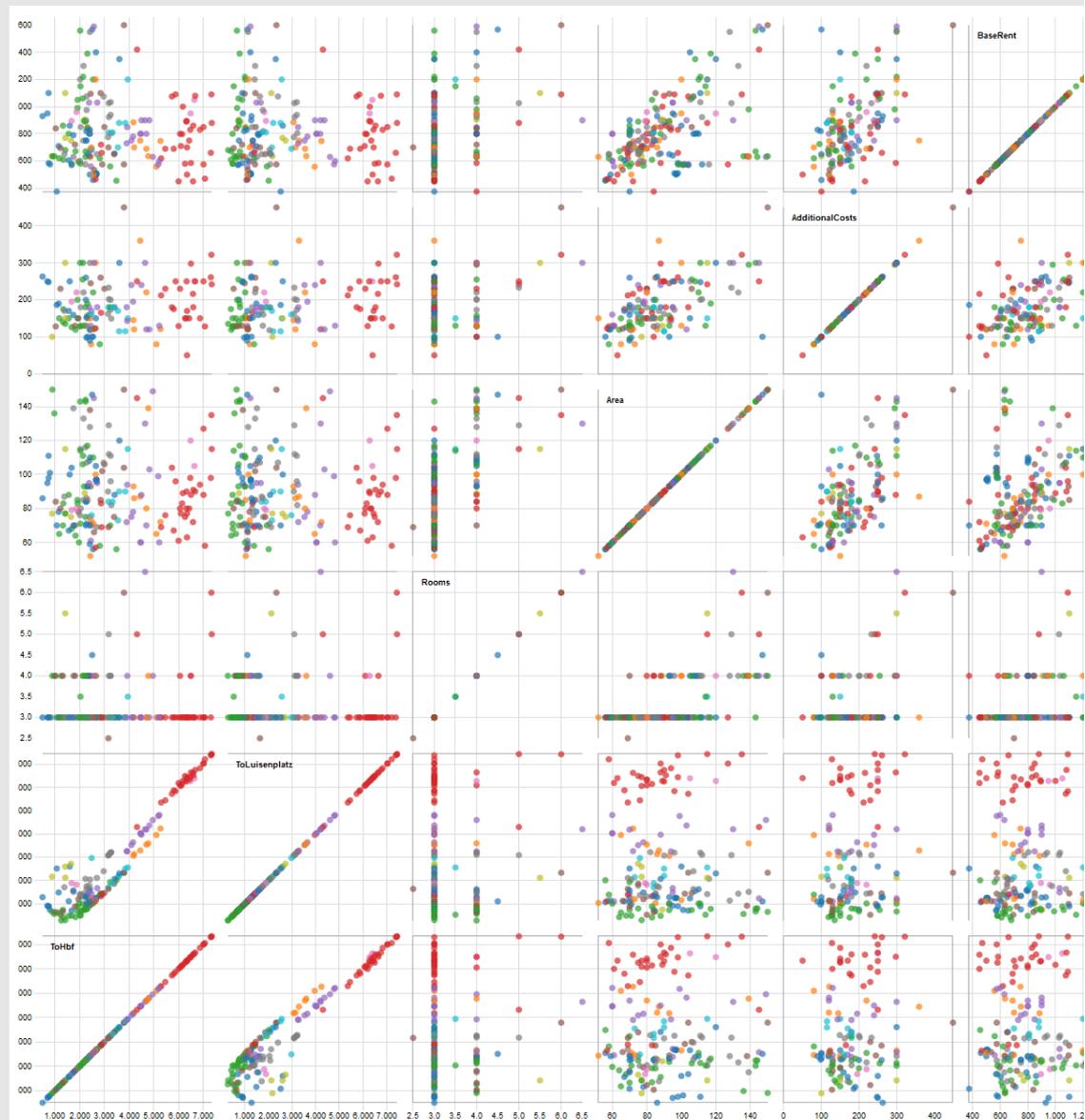
Scatterplot mit echten Daten in SVG

- Kann ich das zu SVG umbauen?
- Kann ich die Daten sinnvoll abbilden?
- Ziel 1: Scatterplot
- Ziel 2: Gefühl für die Daten bekommen



Weitere ausbaustufen

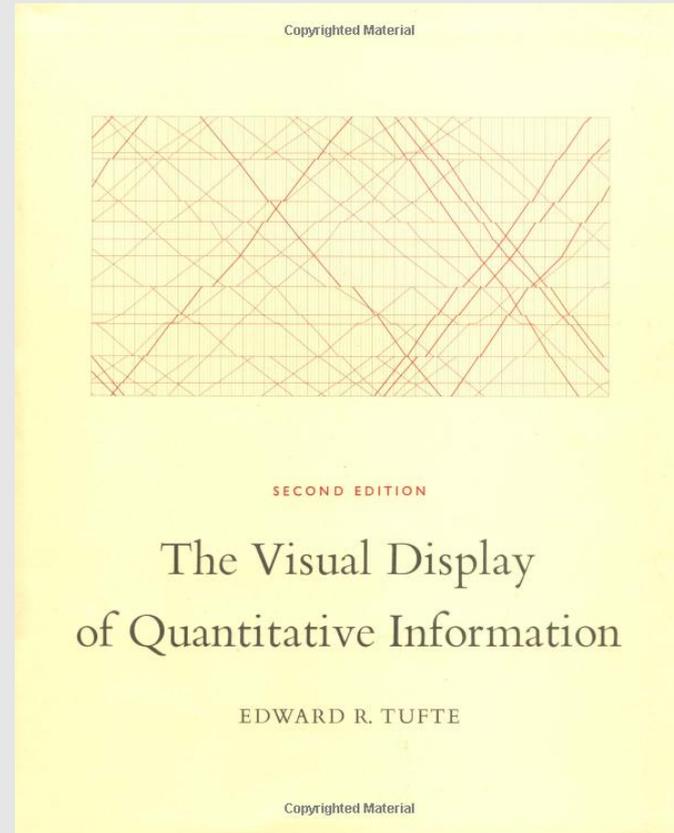
- Scatterplot-Matrix:
Schnell Korrelationen erkennen
- Wie:
<https://bl.ocks.org/mbostock/4063663>
- Warum:
Visual Intro To Machine Learning
<http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>



VISUALISIERUNG
- WEITERFÜHRENDES -

Referenzen, Weiterführendes

- <https://www.amazon.de/dp/0961392142>
- <https://d3js.org/>
- <http://christopheviau.com/d3list/gallery.html>
- <https://bost.ocks.org/mike/> (Blog des D3 Autors)
- <http://bl.ocks.org/mbostock>



Spannende Datenquellen

- „Data is Plural“ (Newsletter und Archiv):
<https://tinyletter.com/data-is-plural>
- Facebook: **Settings** → “Lade eine Kopie deiner Daten herunter”
- Twitter: **Settings** → “Your Twitter Data”
- Pocket: Optionen → **Export** -> “HTML-Datei exportieren”
- Delicious/Pinboard
- Google: Mein Konto → Meine persönlichen Daten → Inhalte kontrollieren → “Archiv erstellen” → Anweisungen folgen.
- Vorschläge willkommen!

D3-Beispiele

- Einfacher Bar-Chart:
<http://bl.ocks.org/mbostock/3885304>
- Line-Chart
<https://bl.ocks.org/mbostock/3883245>
- Line-Chart mit Zoom/Brush
<http://bl.ocks.org/mbostock/34f08d5e11952a80609169b7917d4172>
- Sleep Cycles (fortgeschritten)
<http://bl.ocks.org/mbostock/3cfa2d1dbae2162a60203b287431382c>
- Treemap (etwas komplizierter)
<http://bl.ocks.org/mbostock/8fe6fa6ed1fa976e5dd76cfa4d816fec>
- SVG Transitions
<http://bl.ocks.org/mbostock/1125997>

Aufgabe

- Nehmen Sie ein beliebiges Beispiel (oder zwei oder drei)
- Bringen Sie es auf ihrem Computer zum laufen
- Machen Sie es kaputt!
 - Finden Sie nach und nach heraus, welche Zeilen für was zuständig sind
 - Finden Sie heraus, wie sie das Ergebnis nach Ihren Wünschen ändern können
- Gängige Fehlequellen:
 - D3.v3 und D3.v4 unterscheiden sich deutlich. Viele Beispiele sind noch aus Version 3!
 - Protipp: [API Unterschiede sind hier dokumentiert](#)
 - Wenn Sie Daten nachladen möchten, müssen Sie Ihr Projekt von einem Webserver starten