



Übung Open Data:

Kapitel 4 und 5: Einführung D3.js und Daten einbinden in D3.js

Termin 4, 17. März 2016

Dr. Matthias Stürmer und Prof. Dr. Thomas Myrach

Forschungsstelle Digitale Nachhaltigkeit

Institut für Wirtschaftsinformatik

Universität Bern

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

17. März 2016

Vorlesung

1. Entstehung und Anwendung des Öffentlichkeitsgesetzes
2. Gastreferat von *Martin Stoll, Journalist im RechercheDesk von SonntagsZeitung und Le Matin Dimanche*: Journalismus im Umgang mit dem Öffentlichkeitsgesetz

Übung

1. Einführung D3.js und Daten einbinden in D3.js
2. Gastreferat von *Benjamin Wiederkehr, Interactive Things*

Vertiefte Einführung in die Programmierung

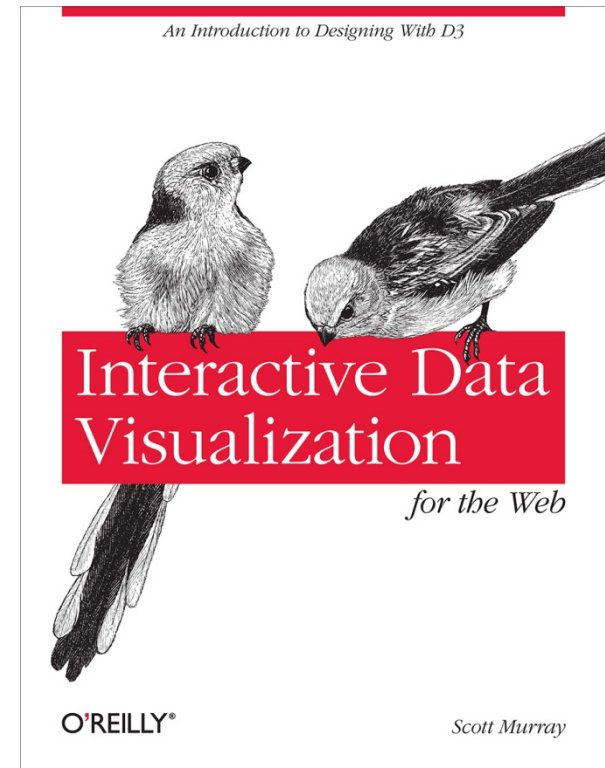
1. Einführung in Programmierkonzepte anhand von Javascript



Interactive Data Visualization for the Web

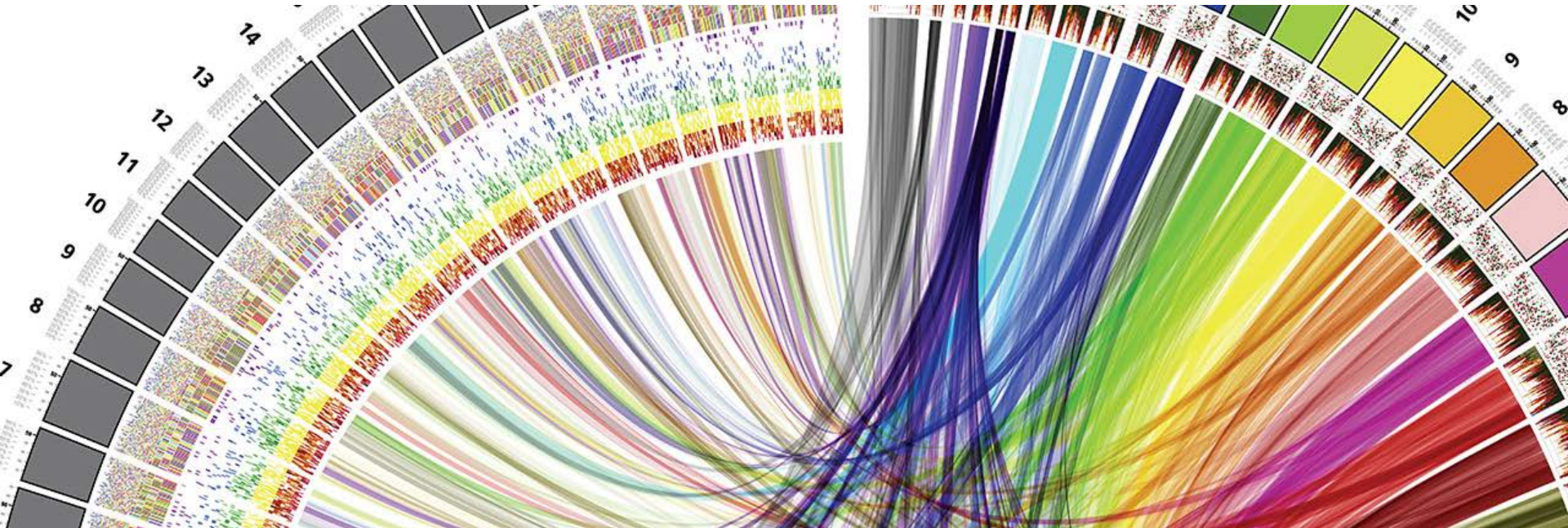
Quelle:

- > 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, 10 Kapitel davon werden in den Übungen behandelt



Agenda

1. **Kapitel 4: Setup D3.js**
2. Kapitel 5: Daten einbinden in D3.js
3. Top Down Approach: Visualisieren von Daten mit Code Snippet



D3.js Download



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](#).

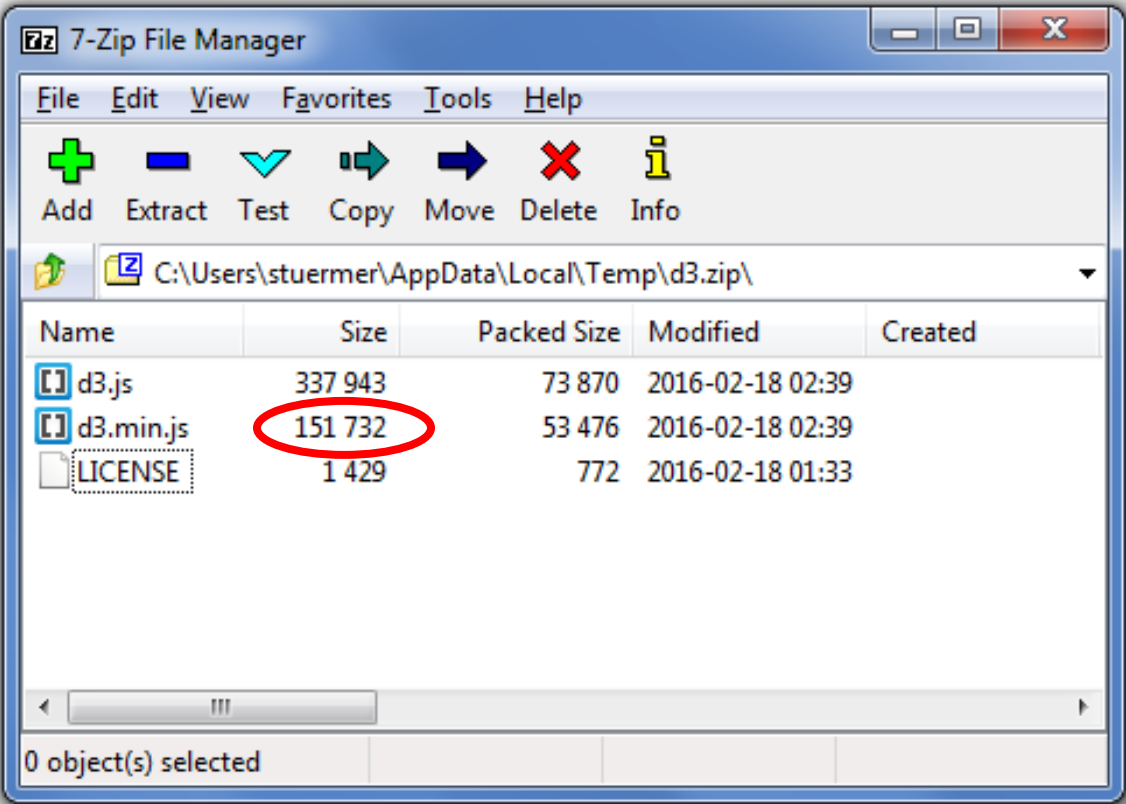
Link: <https://d3js.org>

Download the latest version (3.5.16) here:

• [d3.zip](#)



D3.js Download



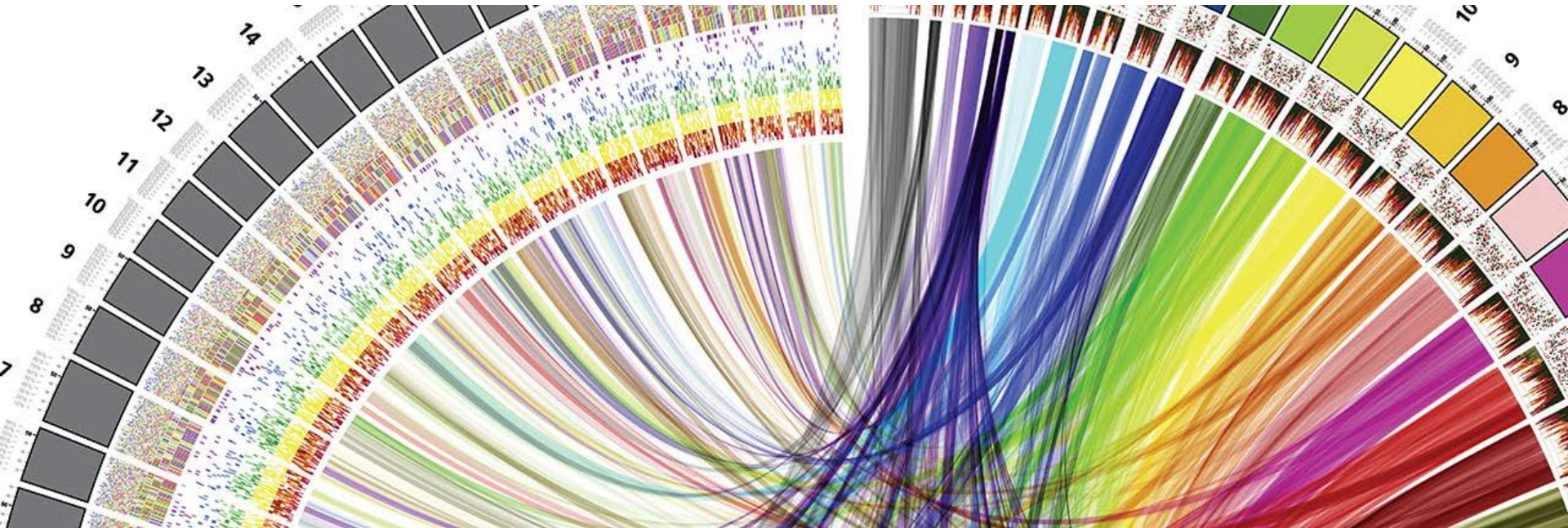
D3.js Konfiguration

1. Neuen Ordner “D3js Projekte” erstellen
2. Unterordner “d3” erstellen
3. Komprimierte d3.min.js Bibliothek reinkopieren
4. D3.js Bibliothek in HTML-Dokument einbinden:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>D3 Page Template</title>
    <script type="text/javascript" src="d3/d3.min.js"></script>
  </head>
  <body>
    <script type="text/javascript">
      // Your beautiful D3 code will go here
    </script>
  </body>
</html>
```


Agenda

1. Kapitel 4: Setup D3.js
2. **Kapitel 5: Daten einbinden in D3.js**
3. Top Down Approach: Visualisieren von Daten mit Code Snippet



Chaining Methods

```
d3.select("body").append("p").text("New paragraph!");
```

ist der gleiche Code wie

```
d3.select("body")  
  .append("p")  
  .text("New paragraph!");
```

ist der gleiche Code wie

```
var body = d3.select("body");  
var p = body.append("p");  
p.text("New paragraph!");
```

Chaining Methods

```
d3.select("body").append("p").text("New paragraph!");
```

D3	References the D3 object, so we can access its methods. Our D3 adventure begins here.
.select("body")	Give the select() method a CSS selector as input, and it will return a reference to the first element in the DOM that matches.
.append("p")	append() creates whatever new DOM element you specify and appends it to the end (but just inside) of whatever selection it's acting on. Finally, append() hands off a reference to the new element it just created.
.text("New paragraph!")	text() takes a string and inserts it between the opening and closing tags of the current selection.
;	The all-important semicolon indicates the end of this line of code. Chain over.

Binding Data

```
d3.select("body").selectAll("p")  
  .data(dataset)  
  .enter()  
  .append("p")  
  .text("New paragraph!");
```

`d3.select("body")`

Finds the body in the DOM and hands off a reference to the next step in the chain.

`.selectAll("p")`

Selects all paragraphs in the DOM. Because none exist yet, this returns an empty selection. Think of this empty selection as representing the paragraphs that will soon exist.

`.data(dataset)`

Counts and parses our data values. There are five values in our array called dataset, so everything past this point is executed five times, once for each value.

`.enter()`

To create new, data-bound elements, you must use `enter()`. This method looks at the current DOM selection, and then at the data being handed to it. If there are more data values than corresponding DOM elements, then `enter()` creates a new placeholder element on which you can work your magic. It then hands off a reference to this new placeholder to the next step in the chain.

`.append("p")`

Takes the empty placeholder selection created by `enter()` and appends a `p` element into the DOM. Hooray! Then it hands off a reference to the element it just created to the next step in the chain.

`.text("New paragraph!")`

Takes the reference to the newly created `p` and inserts a text value.

Using Your Data

Die Testdaten:

```
var dataset = [ 5, 10, 15, 20, 25 ];
```

Anzeigen der Testdaten:

```
d3.select("body").selectAll("p")  
  .data(dataset)  
  .enter()  
  .append("p")  
  .text(function(d) {return "I can count up to " + d; });
```

Nur die letzte Zeile anpassen:

```
  .style("color", function(d) {  
    if (d > 15) { //Threshold of 15  
      return "red";  
    } else {  
      return "black";  
    }  
  });
```


D3.js API Reference

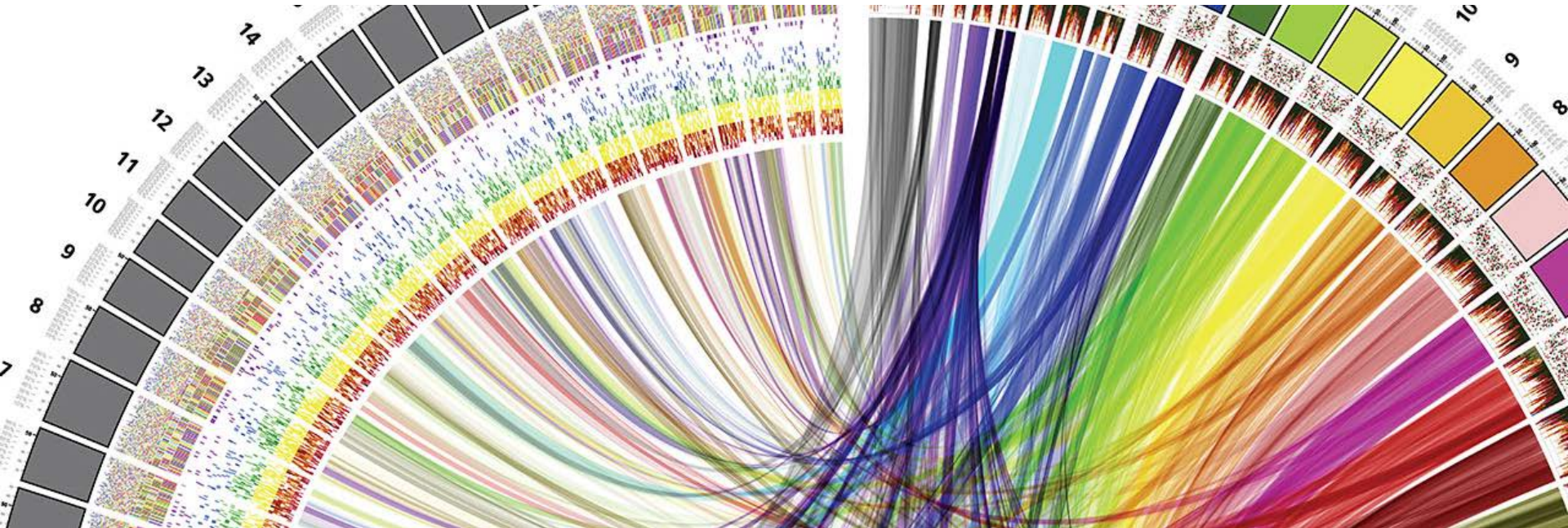
Application Programming Interface von D3.js:

<https://github.com/mbostock/d3/wiki/API-Reference>

- > [Selections](#) - manipulate elements in the current document.
- > [Transitions](#) - interpolate attributes and styles smoothly over time.
- > [Arrays](#) - manipulate arrays and objects with ease.
- > [Requests](#) - load external data.
- > [Formatting](#) - convert numbers, dates and other objects to strings.
- > [Localization](#) - control locale-specific behavior, such as number formatting.
- > [Colors](#) - parse and manipulate colors; work with color spaces.
- > [Namespaces](#) - extend D3's support for XML namespaces.
- > [Math](#) - miscellaneous mathematical functions.
- > [Internals](#) - sundry utilities for extending D3.

Agenda

1. Kapitel 4: Setup D3.js
2. Kapitel 5: Daten einbinden in D3.js
3. **Top Down Approach: Visualisieren von Daten mit Code Snippet**




Anpassen bestehender D3.js Snippets

Vorgehen:

1. Welche Daten werden visualisiert? -> Bevölkerungswachstum
2. Welche Visualisierungsart macht Sinn? -> z.B. D3.js Show Real
3. Wie muss ich die Daten anpassen? -> Transponieren
4. Was muss ich am Code Snippet anpassen -> Pfade
5. Was läuft schief? -> Debugging

Daten holen



Schweizerische Eidgenossenschaft

Confédération suisse

Confederazione Svizzera

Confederaziun svizra

Statistik Schweiz

Aktuell

Themen

Regional

International

Infothek

Dienstleistungen

Institutionen

ChronoStat

Suche

Suchen

01 - Bevölkerung

Übersicht

Panorama

Die wichtigsten Zahlen

Das Thema von A bis Z

Neues im Thema

Agenda

Bevölkerungsstand - struktur

Bevölkerungsbewegung

Migration und Integration

Zukünftige Bevölkerungsentwicklung

Daten, Indikatoren - Schweiz Szenarien

Daten, Indikatoren - Kantonale Szenarien

Daten, Indikatoren - Haushaltsszenarien

Analysen

Familien, Haushalte

Sprachen, Religionen

Zum Nachschlagen

Medienmitteilungen

Newsletter

Publikationen

Demos

Demografisches Porträt

Ausländerinnen und Ausländer in der Schweiz

Statistisches Lexikon

Karten, Atlanten

Online-Datenrecherche

Statistische Grundlagen

Erhebungen, Quellen

Nomenklaturen, Inventare

Definitionen

Bundesamt für Statistik > Themen > 01 - Bevölkerung > Zukünftige Bevölkerungsentwicklung > Daten, Indikatoren - Kantonale Szenarien > Übersicht

Seite drucken

Share

Zukünftige Bevölkerungsentwicklung – Daten, Indikatoren - Kantonale Szenarien



Übersicht

Szenarien zur Bevölkerungsentwicklung der Kantone der Schweiz 2010-2035

Das Bundesamt für Statistik (BFS) hat zum dritten Mal Szenarien zur Bevölkerungsentwicklung der Kantone berechnet, die sich auf die aktuellsten Szenarien für die gesamte Schweiz abstützen. Die Bedeutung dieser Szenarien besteht darin, dass es sich dabei um eine Regionalisierung der nationalen Szenarien (bzw. deren Methoden und Hypothesen) handelt und nicht einfach um eine Verknüpfung verschiedener kantonalen Szenarien oder eine Verteilung der Bestände der nationalen Szenarien auf die verschiedenen Kantone. So wurden die kantonalen Szenarien zur Bevölkerungsentwicklung gleichzeitig für jeden einzelnen Kanton berechnet und dann Schritt für Schritt auf die Ergebnisse der Vorausschätzungen zur Bevölkerungsentwicklung der gesamten Schweiz abgestimmt. Damit ist gewährleistet, dass die Summe der kantonalen Ergebnisse für jedes Jahr des Beobachtungszeitraums exakt mit der gesamtschweizerischen Bevölkerungsbilanz übereinstimmt.

Drei Szenarien wurden berechnet. Das mittlere Szenario (AR-00-2010) ist das Referenzszenario und führt die im Laufe der letzten Jahrzehnte beobachteten Entwicklungen fort. Das hohe Szenario (BR-00-2010) kombiniert die Hypothesen, welche das Bevölkerungswachstum begünstigen. Das tiefe Szenario (CR-00-2010) geht von Hypothesen aus, die dem Bevölkerungswachstum weniger förderlich sind.

Die detaillierten Ergebnisse der kantonalen Bevölkerungsszenarien stehen Ihnen in der Datenbank [STAT-TAB](#) zur Verfügung.

Dokument / Objekt	Titel	Perioden
	67 KB Szenarien zur Bevölkerungsentwicklung der Kantone der Schweiz 2010-2035 - Ständige Wohnbevölkerung nach Kantonen gemäss 3 Szenarien (je-d-01.03.02.01)	2010-2035
	Bundesamt für Statistik BFS	
	67 KB Szenarien zur Bevölkerungsentwicklung der Kantone der Schweiz 2010-2035 - Ständige Wohnbevölkerung nach Altersklassen und Geschlecht gemäss dem mittleren Szenario AR-00-2010	2010-2035

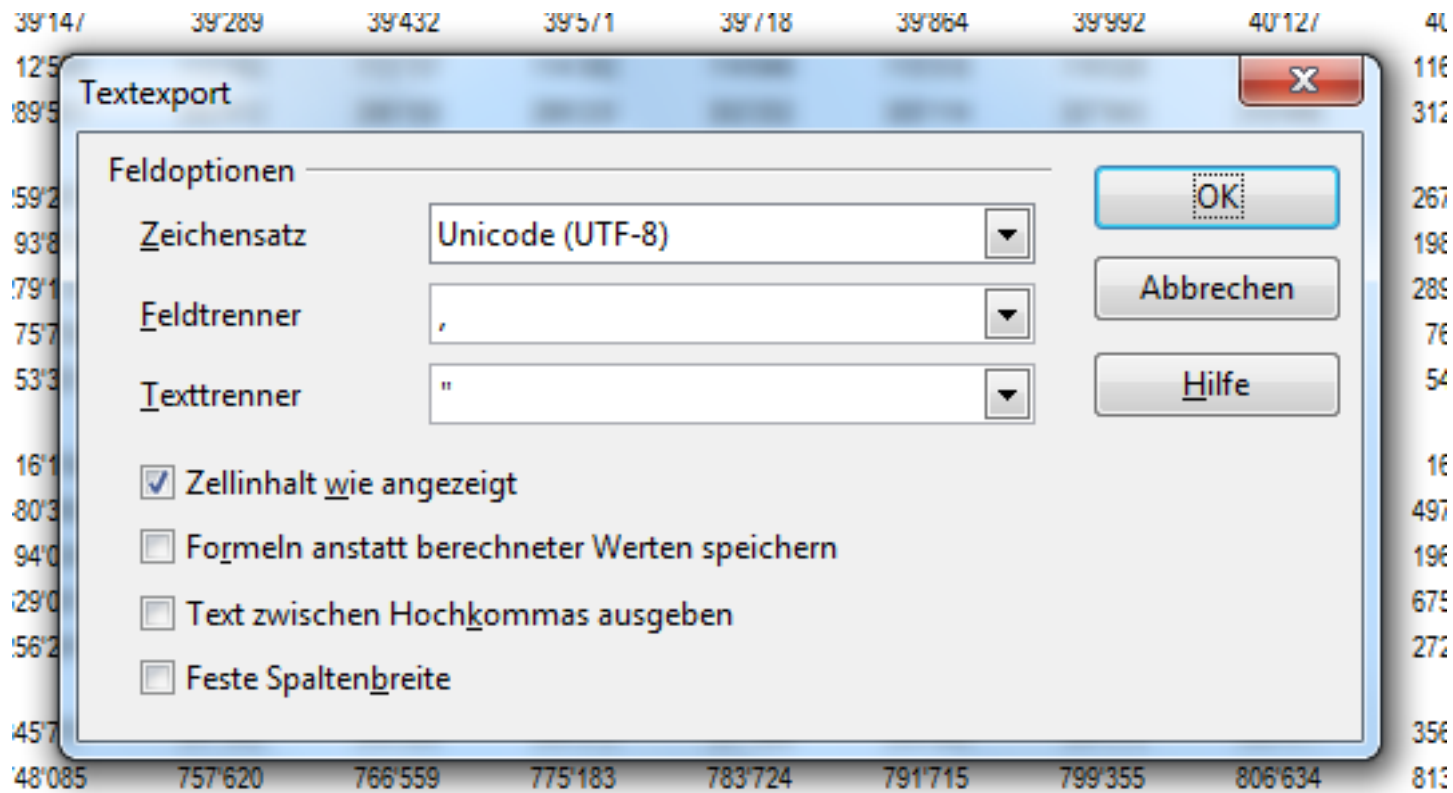
Link: http://www.bfs.admin.ch/bfs/portal/de/index/themen/01/03/blank/key_kant/01.html

FS 2016

17

je-d-01.04.02.00.01.ods - LibreOffice Calc																		
Datei	Bearbeiten	Ansicht	Einfügen	Format	Extras	Daten	Fenster	Hilfe										
<div> <div> </div> <div> </div> </div>																		
<div> <div> <div>Arial Narrow</div> <div>8</div> <div> </div> </div> <div> <div> </div> </div> </div>																		
R38																		
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	je-d-01.04.02.00.01 Ständige Wohnbevölkerung nach Kantonen gemäss dem "mittleren" Szenario AR-00-2010, 2010-2035																	
3		2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026
4	Schweiz	7'856'600	7'922'090	7'988'333	8'047'559	8'102'292	8'155'127	8'207'402	8'260'073	8'309'704	8'358'859	8'401'941	8'444'913	8'485'802	8'524'627	8'561'368	8'595'882	
5	Zürich	1'368'669	1'382'998	1'392'745	1'402'327	1'413'343	1'424'093	1'434'815	1'445'692	1'455'988	1'465'920	1'475'482	1'484'599	1'493'302	1'501'572	1'509'404	1'516'809	
6	Bern	977'690	980'541	983'815	986'664	989'301	992'083	995'044	998'164	1'001'074	1'003'781	1'006'245	1'008'477	1'010'528	1'012'358	1'013'991	1'015'403	
7	Luzern	377'028	380'487	383'765	387'451	390'843	394'152	397'393	400'613	403'699	406'700	409'566	412'312	414'947	417'473	419'849	422'106	
8	Uri	35'473	35'553	35'670	35'730	35'754	35'783	35'808	35'861	35'902	35'950	36'011	36'067	36'135	36'185	36'227	36'265	
9	Schwyz	145'749	146'185	147'008	147'605	148'350	148'968	149'691	150'387	151'089	151'729	152'386	153'000	153'633	154'228	154'815	155'369	
10	Obwalden	35'426	35'801	36'172	36'592	36'900	37'188	37'484	37'808	38'077	38'350	38'642	38'931	39'185	39'438	39'670	39'882	
11	Nidwalden	40'912	41'035	41'231	41'422	41'646	41'868	42'106	42'345	42'594	42'838	43'059	43'263	43'456	43'646	43'828	44'011	
12	Glarus	38'628	38'739	38'965	39'147	39'289	39'432	39'571	39'718	39'864	39'992	40'127	40'248	40'369	40'485	40'597	40'714	
13	Zug	111'164	111'416	112'025	112'569	113'162	113'757	114'362	114'946	115'518	116'028	116'515	116'969	117'404	117'807	118'189	118'559	
14	Freiburg	277'297	281'412	285'670														

Export der Daten als UTF-8 CSV-Datei



FileEditViewNavigateDebugHelp

Working Files

html_demo.html

demo.html

index.html

cantons.json

my_app_Complete.js

my_app.js

bevoelkerungswachstum.csv

D3 Programmierung

css

style.css

js

vendor

d3.min.js

jquery-1.11.0.min.js

underscore-min.js

my_app.js

my_app_Complete.js

bevoelkerungswachstum.csv

cantons.json

demo.html

flare.json

html_demo.html

index.html

svg_demo.html

bevoelkerungswachstum.csv - Brackets

je-d-01.04.02.00.01,"Ständige Wohnbevölkerung nach Kantonen gemäss dem ""mittleren"" Szenario AR-00-2010, 2010-2035"

2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,2029,2030,2031,2032,2033

Schweiz,7'856'600,7'922'090,7'988'333,8'047'559,8'102'292,8'155'127,8'207'402,8'260'073,8'309'704,8'358'444,8'403'193,8'448'802,8'494'512,8'540'227,8'586'000,8'631'776,8'677'552,8'723'328,8'769'104,8'814'880,8'860'656,8'906'432,8'952'208,8'997'984,9'043'760,9'089'536,9'135'312,9'181'088,9'226'864,9'272'640,9'318'416,9'364'192,9'409'968,9'455'744,9'501'520,9'547'296,9'593'072,9'638'848,9'684'624,9'730'400,9'776'176,9'821'952,9'867'728,9'913'504,9'959'280,10'005'056,10'050'832,10'096'608,10'142'384,10'188'160,10'233'936,10'279'712,10'325'488,10'371'264,10'417'040,10'462'816,10'508'592,10'554'368,10'600'144,10'645'920,10'691'696,10'737'472,10'783'248,10'829'024,10'874'800,10'920'576,10'966'352,11'012'128,11'057'904,11'103'680,11'149'456,11'195'232,11'241'008,11'286'784,11'332'560,11'378'336,11'424'112,11'469'888,11'515'664,11'561'440,11'607'216,11'652'992,11'698'768,11'744'544,11'790'320,11'836'096,11'881'872,11'927'648,11'973'424,12'019'200,12'064'976,12'110'752,12'156'528,12'202'304,12'248'080,12'293'856,12'339'632,12'385'408,12'431'184,12'476'960,12'522'736,12'568'512,12'614'288,12'660'064,12'705'840,12'751'616,12'797'392,12'843'168,12'888'944,12'934'720,12'980'496,13'026'272,13'072'048,13'117'824,13'163'600,13'209'376,13'255'152,13'300'928,13'346'704,13'392'480,13'438'256,13'484'032,13'529'808,13'575'584,13'621'360,13'667'136,13'712'912,13'758'688,13'804'464,13'850'240,13'896'016,13'941'792,13'987'568,14'033'344,14'079'120,14'124'896,14'170'672,14'216'448,14'262'224,14'308'000,14'353'776,14'399'552,14'445'328,14'491'104,14'536'880,14'582'656,14'628'432,14'674'208,14'719'984,14'765'760,14'811'536,14'857'312,14'903'088,14'948'864,14'994'640,15'040'416,15'086'192,15'131'968,15'177'744,15'223'520,15'269'296,15'315'072,15'360'848,15'406'624,15'452'400,15'498'176,15'543'952,15'589'728,15'635'504,15'681'280,15'727'056,15'772'832,15'818'608,15'864'384,15'910'160,15'955'936,16'001'712,16'047'488,16'093'264,16'139'040,16'184'816,16'230'592,16'276'368,16'322'144,16'367'920,16'413'696,16'459'472,16'505'248,16'551'024,16'596'800,16'642'576,16'688'352,16'734'128,16'779'904,16'825'680,16'871'456,16'917'232,16'963'008,17'008'784,17'054'560,17'100'336,17'146'112,17'191'888,17'237'664,17'283'440,17'329'216,17'374'992,17'420'768,17'466'544,17'512'320,17'558'096,17'603'872,17'649'648,17'695'424,17'741'200,17'786'976,17'832'752,17'878'528,17'924'304,17'970'080,18'015'856,18'061'632,18'107'408,18'153'184,18'198'960,18'244'736,18'290'512,18'336'288,18'382'064,18'427'840,18'473'616,18'519'392,18'565'168,18'610'944,18'656'720,18'702'496,18'748'272,18'794'048,18'839'824,18'885'600,18'931'376,18'977'152,19'022'928,19'068'704,19'114'480,19'160'256,19'206'032,19'251'808,19'297'584,19'343'360,19'389'136,19'434'912,19'480'688,19'526'464,19'572'240,19'618'016,19'663'792,19'709'568,19'755'344,19'801'120,19'846'896,19'892'672,19'938'448,19'984'224,20'029'992,20'075'768,20'121'544,20'167'320,20'213'096,20'258'872,20'304'648,20'350'424,20'396'200,20'441'976,20'487'752,20'533'528,20'579'304,20'625'080,20'670'856,20'716'632,20'762'408,20'808'184,20'853'960,20'899'736,20'945'512,20'991'288,21'037'064,21'082'840,21'128'616,21'174'392,21'220'168,21'265'944,21'311'720,21'357'496,21'403'272,21'449'048,21'494'824,21'540'592,21'586'368,21'632'144,21'677'920,21'723'696,21'769'472,21'815'248,21'861'024,21'906'800,21'952'576,21'998'352,22'044'128,22'089'904,22'135'680,22'181'456,22'227'232,22'273'008,22'318'784,22'364'560,22'410'336,22'456'112,22'501'888,22'547'664,22'593'440,22'639'216,22'684'992,22'730'768,22'776'544,22'822'320,22'868'096,22'913'872,22'959'648,23'005'424,23'051'200,23'096'976,23'142'752,23'188

Daten bereinigen: Kommentare und doppelte Zwischenräume löschen

• bevoelkerungswachstum.csv - Brackets

File Edit View Navigate Debug Help

Working Files

html_demo.html

demo.html

index.html

cantons.json

my_app_Complete.js

my_app.js

• bevoelkerungswachstum.csv

D3 Programmierung

css

style.css

js

vendor

d3.min.js

jquery-1.11.0.min.js

underscore-min.js

my_app.js

my_app_Complete.js

bevoelkerungswachstum.csv

cantons.json

demo.html

flare.json

html_demo.html

index.html

svg_demo.html

702 results

Aa . *

1

Region,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020,2021,2022,2023,2024,2025,2026,2027,2028,202

2

Schweiz,7'856'600,7'922'090,7'988'333,8'047'559,8'102'292,8'155'127,8'207'402,8'260'073,8

3

Zürich,1'368'669,1'382'998,1'392'745,1'402'327,1'413'343,1'424'093,1'434'815,1'445'692,1'

4

Bern,977'690,980'541,983'815,986'664,989'301,992'083,995'044,998'164,1'001'074,1'003'78

5

Luzern,377'028,380'487,383'765,387'451,390'843,394'152,397'393,400'613,403'699,406'700

6

Uri,35'473,35'553,35'670,35'730,35'754,35'783,35'808,35'861,35'902,35'950,36'011,36

7

Schwyz,145'749,146'185,147'008,147'605,148'350,148'968,149'691,150'367,151'089,151'729

8

Obwalden,35'426,35'801,36'172,36'592,36'900,37'188,37'484,37'808,38'077,38'350,38'642

9

Nidwalden,40'912,41'035,41'231,41'422,41'646,41'868,42'106,42'345,42'594,42'838,43'05

10

Glarus,38'628,38'739,38'965,39'147,39'289,39'432,39'571,39'718,39'864,39'992,40'127

11

Zug,111'164,111'416,112'025,112'569,113'162,113'757,114'362,114'946,115'518,116'028,1

12

Freiburg,277'297,281'412,285'670,289'501,292'912,296'150,299'235,302'252,305'114,307'84

13

Solothurn,254'170,255'956,257'755,259'267,260'490,261'572,262'658,263'692,264'681,265'6

14

Basel-Stadt,188'962,190'413,192'406,193'859,194'829,195'633,196'317,196'953,197'418,197

15

Basel-Landschaft,274'387,275'900,277'578,279'139,280'439,281'761,283'063,284'390,285'696

16

Schaffhausen,75'737,75'716,75'748,75'769,75'835,75'932,76'077,76'236,76'364,76'477,76

17

Appenzell A.Rh.,53'019,53'041,53'203,53'351,53'469,53'611,53'744,53'906,54'055,54'204

18

Appenzell I.Rh.,15'808,15'964,16'052,16'154,16'254,16'358,16'429,16'515,16'591,16'678

19

St. Gallen,476'374,477'380,478'706,480'360,482'432,484'758,487'260,489'742,491'932,493'

20

Graubünden,192'570,193'209,193'710,194'024,194'150,194'336,194'577,194'898,195'216,195'

Daten bereinigen: Hochkommas löschen

• bevoelkerungswachstum.csv - Brackets

File Edit View Navigate Debug Help

Working Files

html_demo.html

demo.html

index.html

cantons.json

my_app_Complete.js

my_app.js

• bevoelkerungswachstum.csv

D3 Programmierung

css

style.css

js

vendor

d3.min.js

jquery-1.11.0.min.js

underscore-min.js

my_app.js

my_app_Complete.js

bevoelkerungswachstum.csv

cantons.json

demo.html

flare.json

html_demo.html

index.html

svg_demo.html

772 results

Aa

.

*

◀

▶

8'524'627,8'561'368,8'595'882,8'628'351,8'658'886,8'687'312,8'713'831,8'738'477,8'761'401,8'782'695,8'802'409,8'820'501'572,1'509'404,1'516'809,1'523'809,1'530'507,1'536'801,1'542'776,1'548'413,1'553'805,1'558'908,1'563'769,1'568'403,1'015'403,1'016'603,1'017'633,1'018'416,1'019'014,1'019'388,1'019'561,1'019'552,1'019'397,1'019'061,1'018'614

5 Luzern,377'028,380'487,383'765,387'451,390'843,394'152,397'393,400'613,403'699,406'700,409'566,412'312,414'947,417'236,426'253,428'125,429'871,431'465,432'945,434'290,435'545,436'699,437'775

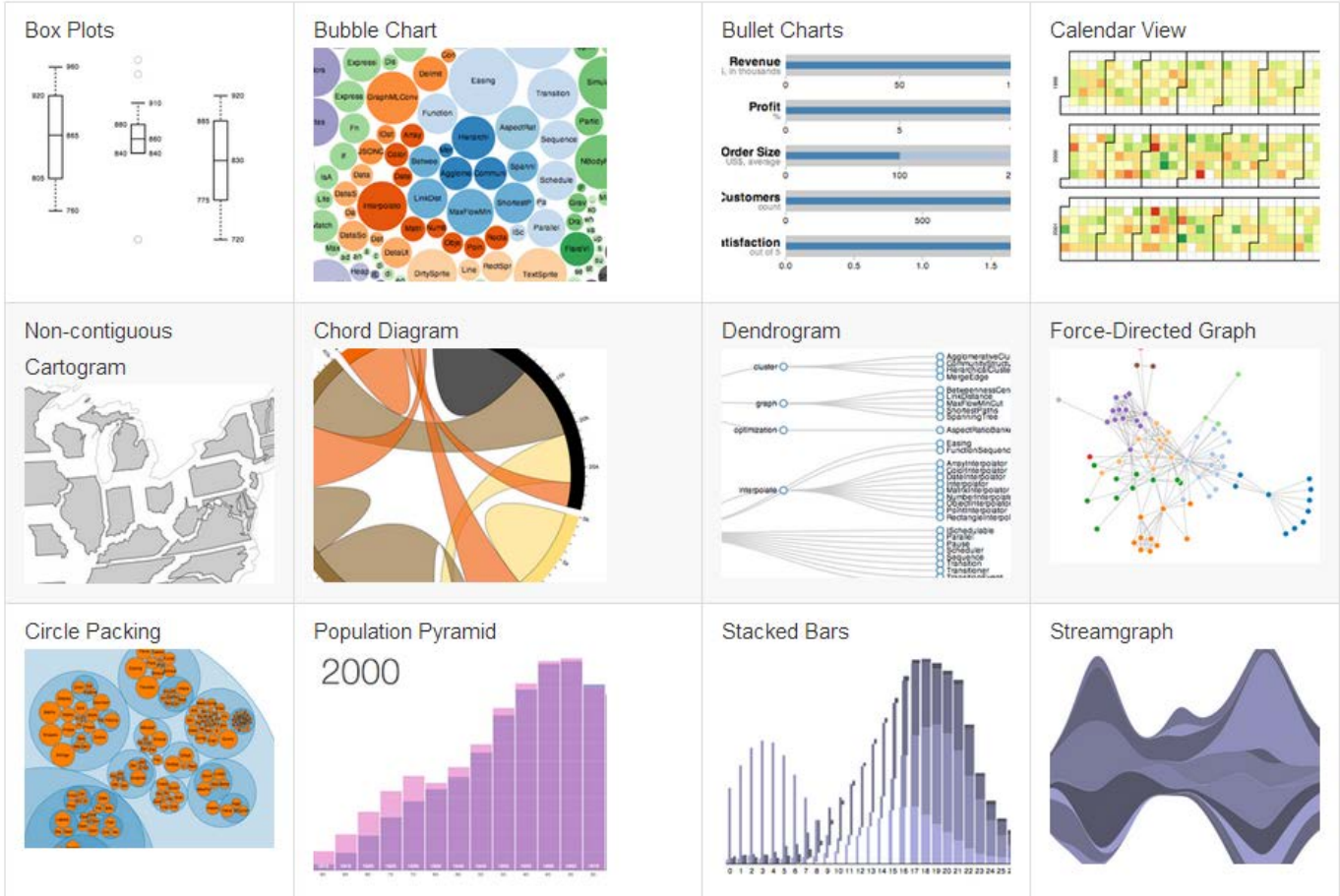
6 Uri,35'473,35'553,35'670,35'730,35'754,35'783,35'808,35'861,35'902,35'950,36'011,36'067,36'135,36'185,36'227,36'265,36'316,36'366,36'429,36'486,36'543,36'599,36'656,36'713,36'770,36'827,36'884,36'941,36'998,37'055,37'112,37'169,37'226,37'283,37'340,37'397,37'454,37'511,37'568,37'625,37'682,37'739,37'796,37'853,37'910,37'967,38'024,38'081,38'138,38'195,38'252,38'309,38'366,38'423,38'480,38'537,38'594,38'651,38'708,38'765,38'822,38'879,38'936,38'993,39'050,39'107,39'164,39'221,39'278,39'335,39'392,39'449,39'506,39'563,39'620,39'677,39'734,39'791,39'848,39'905,39'962,40'019,40'076,40'133,40'190,40'247,40'304,40'361,40'418,40'475,40'532,40'589,40'646,40'703,40'760,40'817,40'874,40'931,40'988,41'045,41'102,41'159,41'216,41'273,41'330,41'387,41'444,41'501,41'558,41'615,41'672,41'729,41'786,41'843,41'900,41'957,42'014,42'071,42'128,42'185,42'242,42'299,42'356,42'413,42'470,42'527,42'584,42'641,42'698,42'755,42'812,42'869,42'926,42'983,43'040,43'097,43'154,43'211,43'268,43'325,43'382,43'439,43'496,43'553,43'610,43'667,43'724,43'781,43'838,43'895,43'952,44'009,44'066,44'123,44'180,44'237,44'294,44'351,44'408,44'465,44'522,44'579,44'636,44'693,44'750,44'807,44'864,44'921,44'978,45'035,45'092,45'149,45'206,45'263,45'320,45'377,45'434,45'491,45'548,45'605,45'662,45'719,45'776,45'833,45'890,45'947,46'004,46'061,46'118,46'175,46'232,46'289,46'346,46'403,46'460,46'517,46'574,46'631,46'688,46'745,46'802,46'859,46'916,46'973,47'030,47'087,47'144,47'201,47'258,47'315,47'372,47'429,47'486,47'543,47'600,47'657,47'714,47'771,47'828,47'885,47'942,48'000,48'057,48'114,48'171,48'228,48'285,48'342,48'399,48'456,48'513,48'570,48'627,48'684,48'741,48'798,48'855,48'912,48'969,49'026,49'083,49'140,49'197,49'254,49'311,49'368,49'425,49'482,49'539,49'596,49'653,49'710,49'767,49'824,49'881,49'938,50'000,50'057,50'114,50'171,50'228,50'285,50'342,50'399,50'456,50'513,50'570,50'627,50'684,50'741,50'798,50'855,50'912,50'969,51'026,51'083,51'140,51'197,51'254,51'311,51'368,51'425,51'482,51'539,51'596,51'653,51'710,51'767,51'824,51'881,51'938,51'995,52'052,52'109,52'166,52'223,52'280,52'337,52'394,52'451,52'508,52'565,52'622,52'679,52'736,52'793,52'850,52'907,52'964,53'021,53'078,53'135,53'192,53'249,53'306,53'363,53'420,53'477,53'534,53'591,53'648,53'705,53'762,53'819,53'876,53'933,53'990,54'047,54'104,54'161,54'218,54'275,54'332,54'389,54'446,54'503,54'560,54'617,54'674,54'731,54'788,54'845,54'902,54'959,55'016,55'073,55'130,55'187,55'244,55'301,55'358,55'415,55'472,55'529,55'586,55'643,55'700,55'757,55'814,55'871,55'928,55'985,56'042,56'099,56'156,56'213,56'270,56'327,56'384,56'441,56'498,56'555,56'612,56'669,56'726,56'783,56'840,56'897,56'954,57'011,57'068,57'125,57'182,57'239,57'296,57'353,57'410,57'467,57'524,57'581,57'638,57'695,57'752,57'809,57'866,57'923,57'980,58'037,58'094,58'151,58'208,58'265,58'322,58'379,58'436,58'493,58'550,58'607,58'664,58'721,58'778,58'835,58'892,58'949,59'006,59'063,59'120,59'177,59'234,59'291,59'348,59'405,59'462,59'519,59'576,59'633,59'690,59'747,59'804,59'861,59'918,60'000,60'057,60'114,60'171,60'228,60'285,60'342,60'399,60'456,60'513,60'570,60'627,60'684,60'741,60'798,60'855,60'912,60'969,61'026,61'083,61'140,61'197,61'254,61'311,61'368,61'425,61'482,61'539,61'596,61'653,61'710,61'767,61'824,61'881,61'938,61'995,62'052,62'109,62'166,62'223,62'280,62'337,62'394,62'451,62'508,62'565,62'622,62'679,62'736,62'793,62'850,62'907,62'964,63'021,63'078,63'135,63'192,63'249,63'306,63'363,63'420,63'477,63'534,63'591,63'648,63'705,63'762,63'819,63'876,63'933,63'990,64'047,64'104,64'161,64'218,64'275,64'332,64'389,64'446,64'503,64'560,64'617,64'674,64'731,64'788,64'845,64'902,64'959,65'016,65'073,65'130,65'187,65'244,65'301,65'358,65'415,65'472,65'529,65'586,65'643,65'700,65'757,65'814,65'871,65'928,65'985,66'042,66'099,66'156,66'213,66'270,66'327,66'384,66'441,66'498,66'555,66'612,66'669,66'726,66'783,66'840,66'897,66'954,67'011,67'068,67'125,67'182,67'239,67'296,67'353,67'410,67'467,67'524,67'581,67'638,67'695,67'752,67'809,67'866,67'923,67'980,68'037,68'094,68'151,68'208,68'265,68'322,68'379,68'436,68'493,68'550,68'607,68'664,68'721,68'778,68'835,68'892,68'949,69'006,69'063,69'120,69'177,69'234,69'291,69'348,69'405,69'462,69'519,69'576,69'633,69'690,69'747,69'804,69'861,69'918,70'000,70'057,70'114,70'171,70'228,70'285,70'342,70'399,70'456,70'513,70'570,70'627,70'684,70'741,70'798,70'855,70'912,70'969,71'026,71'083,71'140,71'197,71'254,71'311,71'368,71'425,71'482,71'539,71'596,71'653,71'710,71'767,71'824,71'881,71'938,71'995,72'052,72'109,72'166,72'223,72'280,72'337,72'394,72'451,72'508,72'565,72'622,72'679,72'736,72'793,72'850,72'907,72'964,73'021,73'078,73'135,73'192,73'249,73'306,73'363,73'420,73'477,73'534,73'591,73'648,73'705,73'762,73'819,73'876,73'933,73'990,74'047,74'104,74'161,74'218,74'275,74'332,74'389,74'446,74'503,74'560,74'617,74'674,74'731,74'788,74'845,74'902,74'959,75'016,75'073,75'130,75'187,75'244,75'301,75'358,75'415,75'472,75'529,75'586,75'643,75'700,75'757,75'814,75'871,75'928,75'985,76'042,76'099,76'156,76'213,76'270,76'327,76'384,76'441,76'498,76'555,76'612,76'669,76'726,76'783,76'840,76'897,76'954,77'011,77'068,77'125,77'182,77'239,77'296,77'353,77'410,77'467,77'524,77'581,77'638,77'695,77'752,77'809,77'866,77'923,77'980,78'037,78'094,78'151,78'208,78'265,78'322,78'379,78'436,78'493,78'550,78'607,78'664,78'721,78'778,78'835,78'892,78'949,79'006,79'063,79'120,79'177,79'234,79'291,79'348,79'405,79'462,79'519,79'576,79'633,79'690,79'747,79'804,79'861,79'918,80'000,80'057,80'114,80'171,80'228,80'285,80'342,80'399,80'456,80'513,80'570,80'627,80'684,80'741,80'798,80'855,80'912,80'969,81'026,81'083,81'140,81'197,81'254,81'311,81'368,81'425,81'482,81'539,81'596,81'653,81'710,81'767,81'824,81'881,81'938,81'995,82'052,82'109,82'166,82'223,82'280,82'337,82'394,82'451,82'508,82'565,82'622,82'679,82'736,82'793,82'850,82'907,82'964,83'021,83'078,83'135,83'192,83'249,83'306,83'363,83'420,83'477,83'534,83'591,83'648,83'705,83'762,83'819,83'876,83'933,83'990,84'047,84'104,84'161,84'218,84'275,84'332,84'389,84'446,84'503,84'560,84'617,84'674,84'731,84'788,84'845,84'902,84'959,85'016,85'073,85'130,85'187,85'244,85'301,85'358,85'415,85'472,85'529,85'586,85'643,85'700,85'757,85'814,85'871,85'928,85'985,86'042,86'099,86'156,86'213,86'270,86'327,86'384,86'441,86'498,86'555,86'612,86'669,86'726,86'783,86'840,86'897,86'954,87'011,87'068,87'125,87'182,87'239,87'296,87'353,87'410,87'467,87'524,87'581,87'638,87'695,87'752,87'809,87'866,87'923,87'980,88'037,88'094,88'151,88'208,88'265,88'322,88'379,88'436,88'493,88'550,88'607,88'664,88'721,88'778,88'835,88'892,88'949,89'006,89'063,89'120,89'177,89'234,89'291,89'348,89'405,89'462,89'519,89'576,89'633,89'690,89'747,89'804,89'861,89'918,90'000,90'057,90'114,90'171,90'228,90'285,90'342,90'399,90'456,90'513,90'570,90'627,90'684,90'741,90'798,90'855,90'912,90'969,91'026,91'083,91'140,91'197,91'254,91'311,91'368,91'425,91'482,91'539,91'596,91'653,91'710,91'767,91'824,91'881,91'938,91'995,92'052,92'109,92'166,92'223,92'280,92'337,92'394,92'451,92'508,92'565,92'622,92'679,92'736,92'793,92'850,92'907,92'964,93'021,93'078,93'135,93'192,93'249,93'306,93'363,93'420,93'477,93'534,93'591,93'648,93'705,93'762,93'819,93'876,93'933,93'990,94'047,94'104,94'161,94'218,94'275,94'332,94'389,94'446,94'503,94'560,94'617,94'674,94'731,94'788,94'845,94'902,94'959,95'016,95'073,95'130,95'187,95'244,95'301,95'358,95'415,95'472,95'529,95'586,95'643,95'700,95'757,95'814,95'871,95'928,95'985,96'042,96'099,96'156,96'213,96'270,96'327,96'384,96'441,96'498,96'555,96'612,96'669,96'726,96'783,96'840,96'897,96'954,97'011,97'068,97'125,97'182,97'239,97'296,97'353,97'410,97'467,97'524,97'581,97'638,97'695,97'752,97'809,97'866,97'923,97'980,98'037,98'094,98'151,98'208,98'265,98'322,98'379,98'436,98'493,98'550,98'607,98'664,98'721,98'778,98'835,98'892,98'949,99'006,99'063,99'120,99'177,99'234,99'291,99'348,99'405,99'462,99'519,99'576,99'633,99'690,99'747,99'804,99'861,99'918,100'000,100'057,100'114,100'171,100'228,100'285,100'342,100'399,100'456,100'513,100'570,100'627,100'684,100'741,100'798,100'855,100'912,100'969,101'026,101'083,101'140,101'197,101'254,101'311,101'368,101'425,101'482,101'539,101'596,101'653,101'710,101'767,101'824,101'881,101'938,101'995,102'052,102'109,102'166,102'223,102'280,102'337,102'394,102'451,102'508,102'565,102'622,102'679,102'736,102'793,102'850,102'907,102'964,103'021,103'078,103'135,103'192,103'249,103'306,103'363,103'420,103'477,103'534,103'591,103'648,103'705,103'762,103'819,103'876,103'933,103'990,104'047,104'104,104'161,104'218,104'275,104'332,104'389,104'446,104'503,104'560,104'617,104'674,104'731,104'788,104'845,104'902,104'959,105'016,105'073,105'130,105'187,105'244,105'301,105'358,105'415,105'472,105'529,105'586,105'643,105'700,105'757,105'814,105'871,105'928,105'985,106'042,106'099,106'156,106'213,106'270,106'327,106'384,106'441,106'498,106'555,106'612,106'669,106'726,106'783,106'840,106'897,106'954,107'011,107'068,107'125,107'182,107'239,107'296,107'353,107'410,107'467,107'524,107'581,107'638,107'695,107'752,107'809,107'866,107'923,107'980,108'037,108'094,108'151,108'208,108'265,108'322,108'379,108'436,108'493,108'550,108'607,108'664,108'721,108'778,108'835,108'892,108'949,109'006,109'063,109'120,109'177,109'234,109'291,109'348,109'405,109'462,109'519,109'576,109'633,109'690,109'747,109'804,109'861,109'918,110'000,110'057,110'114,110'171,110'228,110'285,110'342,110'399,110'456,110'513,110'570,110'627,110'684,110'741,110'798,110'855,110'912,110'969,111'026,111'083,111'140,111'197,111'254,111'311,111'368,111'425,111'482,111'539,111'596,111'653,111'710,111'767,111'824,111'881,111'938,111'995,112'052,112'109,112'166,112'223,112'280,112'337,112'394,112'451,112'508,112'565,112'622,112'679,112'736,112'793,112'850,112'907,112'964,113'021,113'078,113'135,113'192,113'249,113'306,113'363,113'420,113'477,113'534,113'591,113'648,113'705,113'762,113'819,113'876,113'933,113'990,114'047,114'104,114'161,114'218,114'275,114'332,114'389,114'446,114'503,114'560,114'617,114'674,114'731,114'788,114'845,114'902,114'959,115'016,115'073,115'130,115'187,115'244,115'301,115'358,115'415,115'472,115'529,115'586,115'643,115'700,115'757,115'814,115'871,115'928,115'985,116'042,116'099,116'156,116'213,116'270,116'327,116'384,116'441,116'498,116'555,116'612,116'669,116'726,116'783,116'840,116'897,116'954,117'011,117'068,117'125,117'182,117'239,117'296,117'353,117'410,117'467,117'524,117'581,117'638,117'695,117'752,117'809,117'866,117'923,117'980,118'037,118'094,118'151,118'208,118'265,118'322,118'379,118'436,118'493,118'550,118'607,118'664,118'721,118'778,118'835,118'892,118'949,119'006,119'063,119'120,119'177,119'234,119'291,119'348,119'405,119'462,119'519,119'576,119'633,119'690,119'747,119'804,119'861,119'918,120'000,120'057,120'114,120'171,120'228,120'285,120'342,120'399,120'456,120'513,120'570,120'627,120'684,120'741,120'798,120'855,120'912,120'969,121'026,121'083,121'140,121'197,121'254,121'311,121'368,121'425,121'482,121'539,121'596,121'653,121'710,121'767,121'824,121'881,121'938,121'995,122'052,122'109,122'166,122'223,122'280,122'337,122'394,122'451,122'508,122'565,122'622,122'679,122'736,122'793,122'850,122'907,122'964,123'021,123'078,123'135,123'192,123'249,123'306,123'363,123'420,123'477,123'534,123'591,123'648,123'705,123'762,123'819,123'876,123'933,123'990,124'047,124'104,124'161,124'218,124'275,124'332,124'389,124'446,124'503,1

Anpassen bestehender D3.js Snippets

Vorgehen:

1. Welche Daten werden visualisiert? -> Bevölkerungswachstum
2. **Welche Visualisierungsart macht Sinn? -> D3.js Show Real**
3. Wie muss ich die Daten anpassen? -> Transponieren
4. Was muss ich am Code Snippet anpassen -> Pfade
5. Was läuft schief? -> Debugging

D3.js Code Snippets

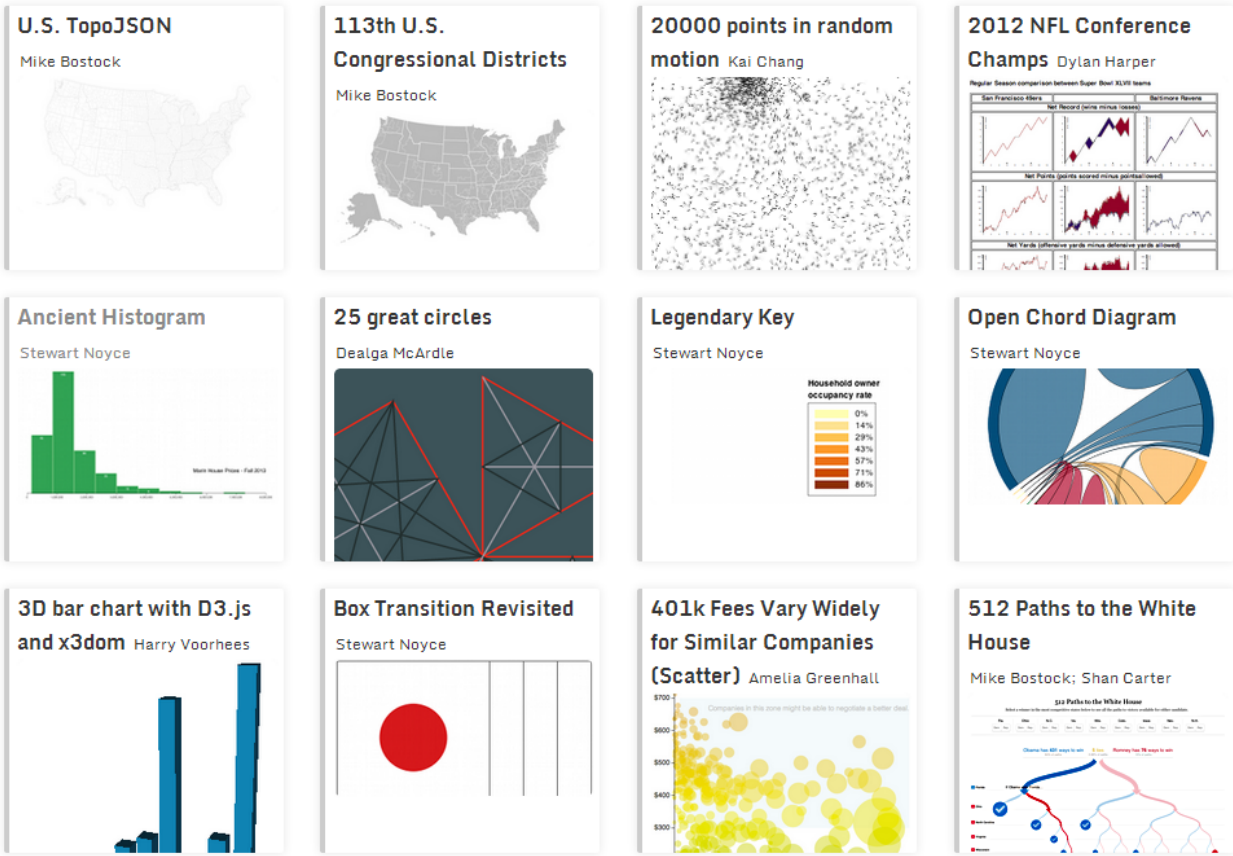


Link: <https://github.com/mbostock/d3/wiki/Gallery>



D3.js Code Snippets

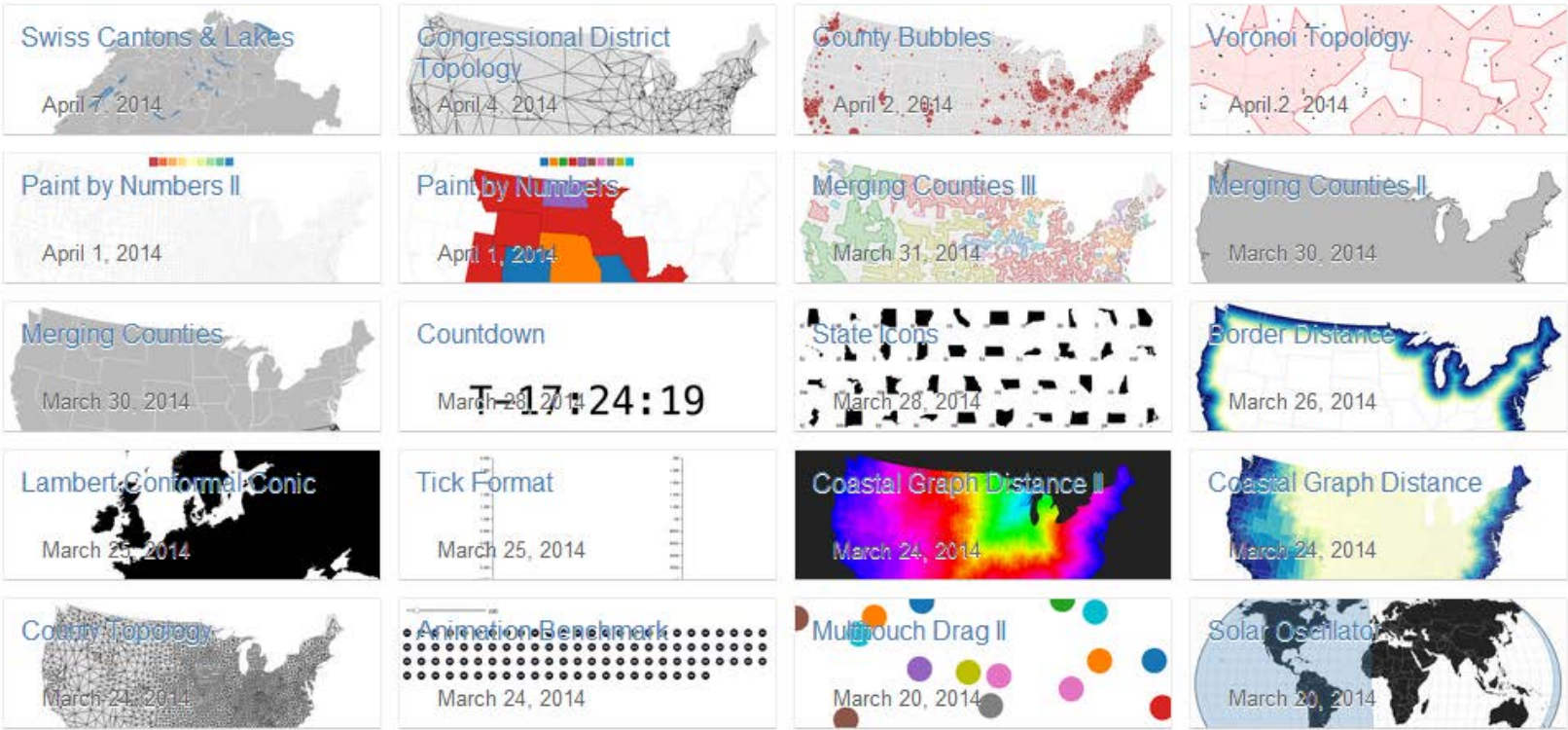
- Author
- Chart Type
- Title
-
- Untagged 1122
- Map 236
- Network 69
- Reusable 65
- Bar Chart 58
- Line Chart 49
- Math 46
- Scatterplot 39
- Bubble Chart 26
- Area Chart 25
- Tree 23
- Voronoi 17
- Pie Chart 15
- Parallel Coordinates 15
- Chord Diagram 15
- Choropleth 14
- Sankey 14
- Stacked Bar Chart 14
- Experiment 12



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

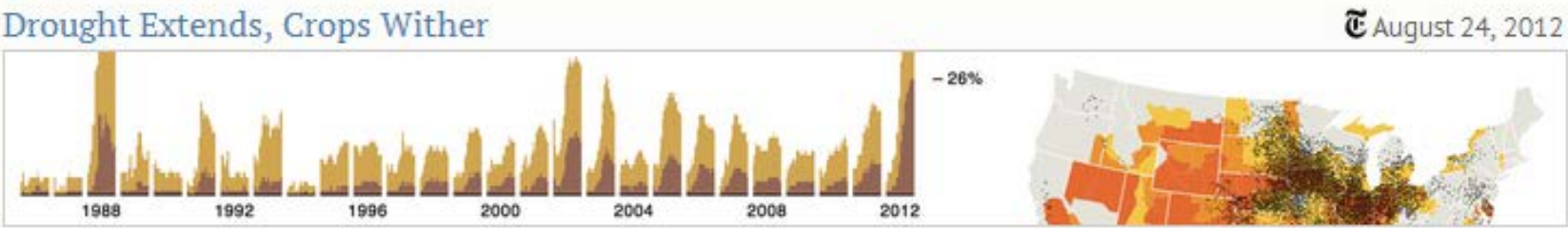
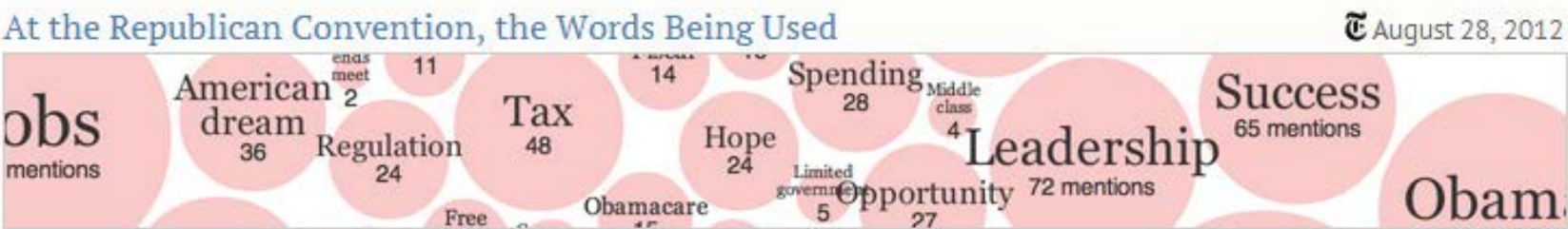
D3.js Code Snippets

mbostock's blocks



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

D3.js Code Snippets



Link: <http://bost.ocks.org/mike/>

Es gibt viele D3 Snippets im Internet...

The screenshot shows a Google search interface with the query 'd3 examples'. Below the search bar, there are tabs for 'Web', 'Bilder', 'Videos', 'Shopping', 'Maps', 'Mehr', and 'Suchoptionen'. The search results are displayed below the tabs, showing approximately 31,000,000 results in 0.30 seconds. The first result is 'Gallery · mbostock/d3 Wiki · GitHub' with a link to 'https://github.com/mbostock/d3/wiki/Gallery'. The second result is 'D3.js - Data-Driven Documents' with a link to 'd3js.org/'. The third result is 'The Big List of D3.js Examples' with a link to 'christopheviau.com/d3list/'. The fourth result is 'D3.js Gallery' with a link to 'biovisualize.github.io/d3visualization/'.

Google d3 examples

Web Bilder Videos Shopping Maps Mehr ▾ Suchoptionen

Ungefähr 31'000'000 Ergebnisse (0.30 Sekunden)

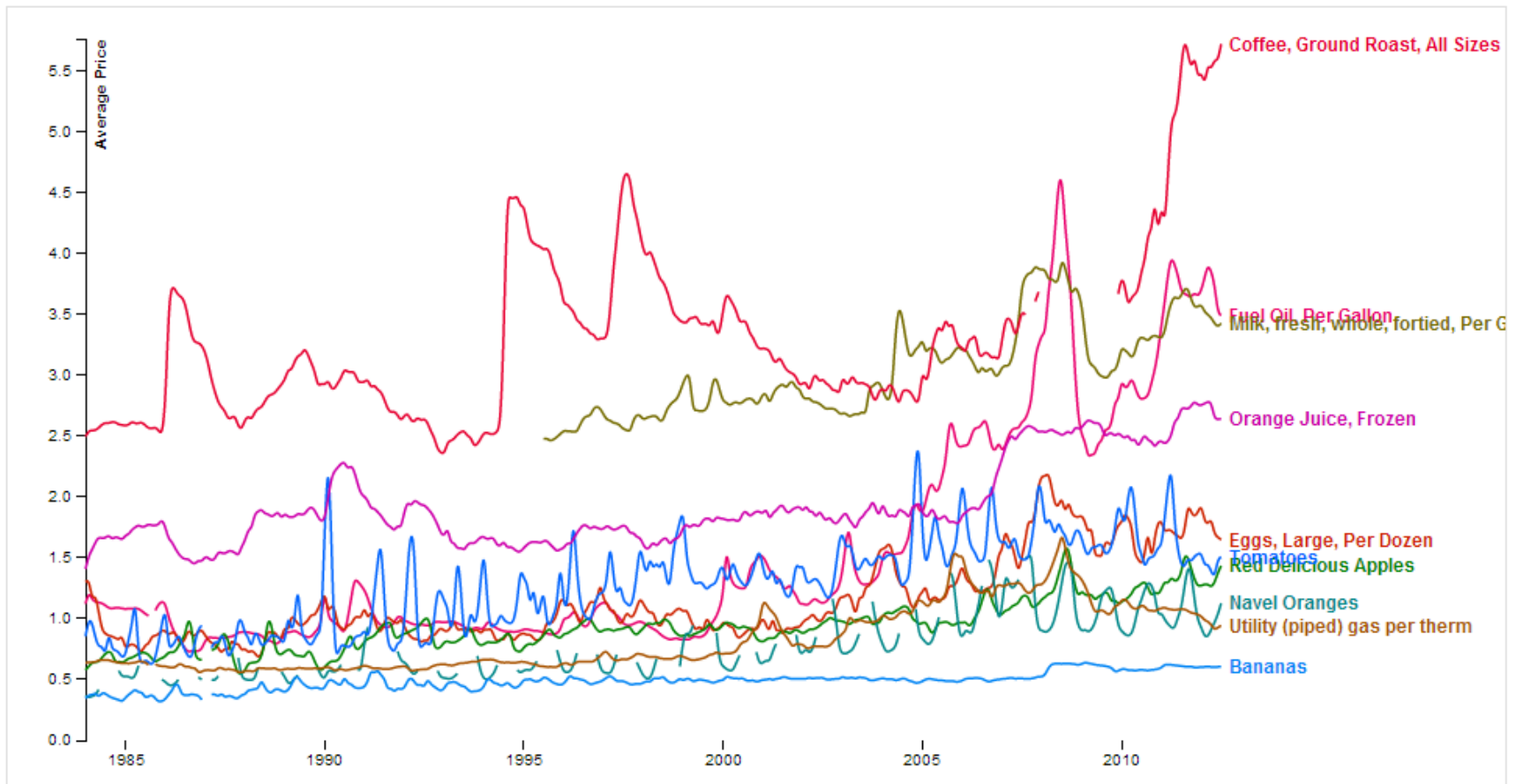
Gallery · mbostock/d3 Wiki · GitHub
<https://github.com/mbostock/d3/wiki/Gallery> ▾ Diese Seite übersetzen
Welcome to the D3 gallery! Feel free to add links to your work! More **examples** are available on bl.ocks.org/mbostock. If you want to share an **example** and don't ...
Sie haben diese Seite 2 Mal aufgerufen. Letzter Besuch: 08.04.14

D3.js - Data-Driven Documents
d3js.org/ ▾ Diese Seite übersetzen
A JavaScript library for manipulating documents based on data using HTML, SVG and CSS. Site includes overview, **examples**, documentation, and source ...
Sie haben diese Seite 5 Mal aufgerufen. Letzter Besuch: 08.04.14

The Big List of D3.js Examples
christopheviau.com/d3list/ ▾ Diese Seite übersetzen
Building a lightweight, flexible D3.js dashboard · Building a tree diagram <http://nowherenearthaca.blogspot.com/2012/06/annotating-d3-example-with-docco>.

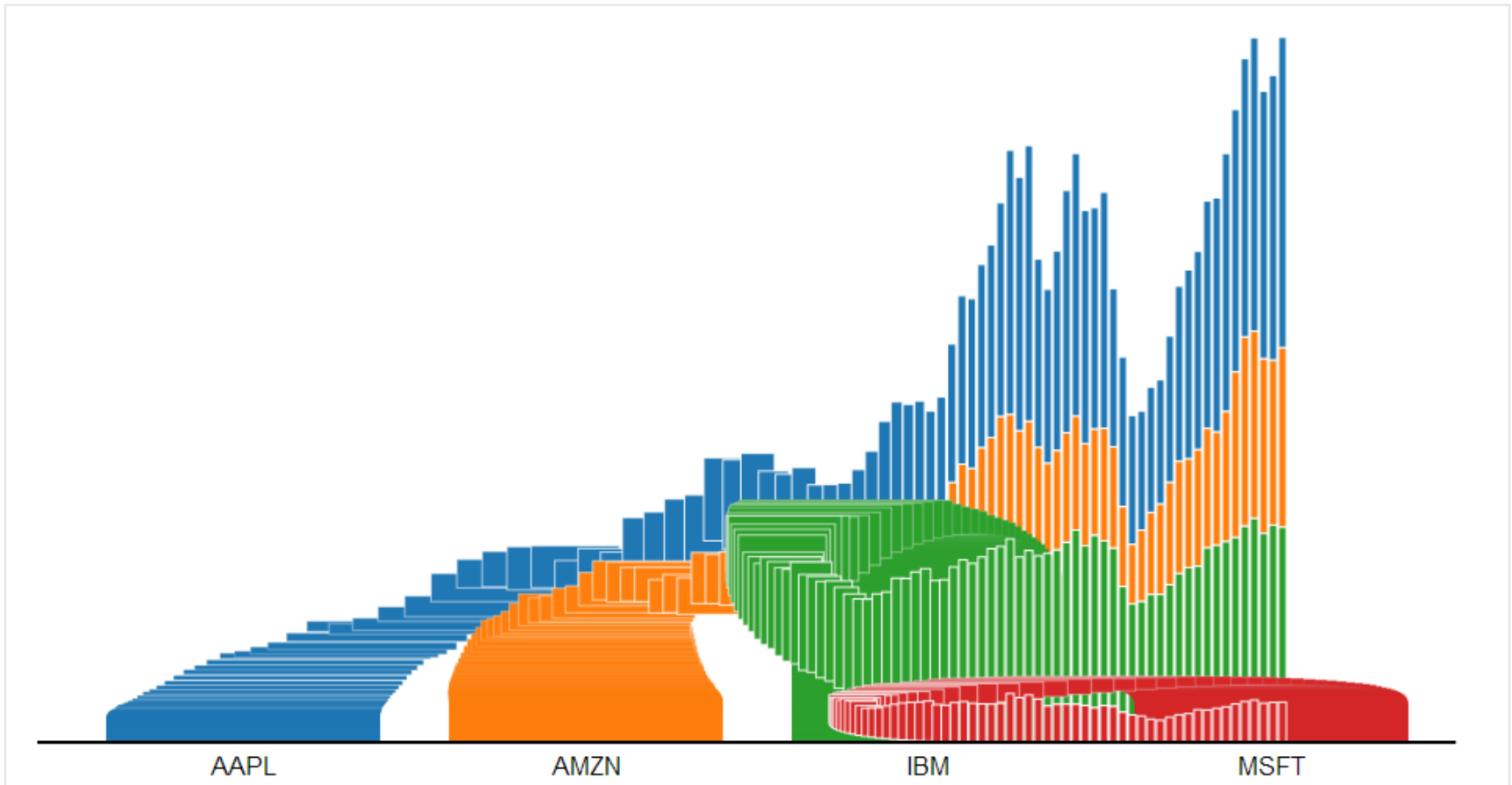
D3.js Gallery
biovisualize.github.io/d3visualization/ ▾ Diese Seite übersetzen
D3.js Gallery | About | The next version of this gallery is here with 1900 D3.js **examples**!
x. You can contribute to this gallery on the Google Spreadsheet or ...

Auswahl einer passenden Visualisierung



Link: <http://bl.ocks.org/syntagmatic/3891711>

Auswahl einer passenden Visualisierung: D3.js Show Reel



Link: <http://bl.ocks.org/mbostock/1256572>

Code und Datenformat von D3.js Show Reel analysieren

index.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<style type="text/css">

svg {
  font-family: "Helvetica Neue", Helvetica;
}

.line {
  fill: none;
  stroke: #000;
  stroke-width: 2px;
}

</style>
<body>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script>

var m = [20, 20, 30, 20],
    w = 960 - m[1] - m[3],
    h = 500 - m[0] - m[2];

var x,
    y,
    duration = 1500,
    delay = 500;
```

stocks.csv

```
symbol,date,price
MSFT,Jan 2000,39.81
MSFT,Feb 2000,36.35
MSFT,Mar 2000,43.22
MSFT,Apr 2000,28.37
MSFT,May 2000,25.45
MSFT,Jun 2000,32.54
MSFT,Jul 2000,28.4
MSFT,Aug 2000,28.4
MSFT,Sep 2000,24.53
MSFT,Oct 2000,28.02
MSFT,Nov 2000,23.34
MSFT,Dec 2000,17.65
MSFT,Jan 2001,24.84
MSFT,Feb 2001,24
MSFT,Mar 2001,22.25
MSFT,Apr 2001,27.56
MSFT,May 2001,28.14
MSFT,Jun 2001,29.7
MSFT,Jul 2001,26.93
MSFT,Aug 2001,23.21
MSFT,Sep 2001,20.82
MSFT,Oct 2001,23.65
MSFT,Nov 2001,26.12
MSFT,Dec 2001,26.95
MSFT,Jan 2002,25.92
```

Link: <http://bl.ocks.org/mbostock/1256572>

Anpassen bestehender D3.js Snippets

Vorgehen:

1. Welche Daten werden visualisiert? -> Bevölkerungswachstum
2. Welche Visualisierungsart macht Sinn? -> D3.js Show Real
3. **Wie muss ich die Daten anpassen? -> Transponieren**
4. Was muss ich am Code Snippet anpassen -> Pfade
5. Was läuft schief? -> Debugging



Daten in LibreOffice öffnen

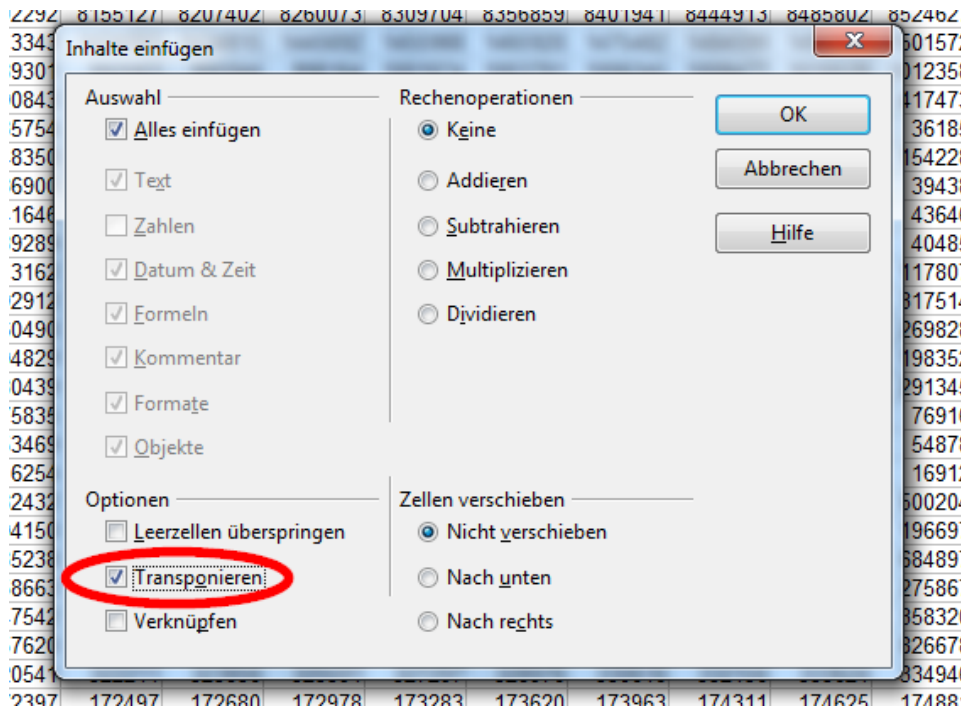
bevoelkerungswachstum.csv - LibreOffice Calc

Dati Bearbeiten Ansicht Einfügen Format Extras Daten Fenster Hilfe

Arial 10

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Region	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	
2	Schweiz	7856600	7922090	7988333	8047559	8102292	8155127	8207402	8260073	8309704	8356859	8401941	844
3	Zürich	1368669	1382998	1392745	1402327	1413343	1424093	1434815	1445692	1455988	1465920	1475482	148
4	Bern	977690	980541	983815	986664	989301	992083	995044	998164	1001074	1003781	1006245	100
5	Luzern	377028	380487	383765	387451	390843	394152	397393	400613	403699	406700	409566	41
6	Uri	35473	35553	35670	35730	35754	35783	35808	35861	35902	35950	36011	3
7	Schwyz	145749	146185	147008	147605	148350	148968	149691	150367	151089	151729	152386	15
8	Obwalden	35426	35801	36172	36592	36900	37188	37484	37808	38077	38350	38642	3
9	Nidwalden	40912	41035	41231	41422	41646	41868	42106	42345	42594	42838	43059	4
10	Glarus	38628	38739	38965	39147	39289	39432	39571	39718	39864	39992	40127	4
11	Zug	111164	111416	112025	112569	113162	113757	114362	114946	115518	116028	116515	11
12	Freiburg	277297	281412	285670	289501	292912	296150	299235	302252	305114	307843	310446	31
13	Solothurn	254170	255956	257755	259267	260490	261572	262658	263692	264681	265667	266690	26
14	Basel-Stadt	188962	190413	192406	193859	194829	195633	196317	196953	197418	197781	198017	19
15	Basel-Landschaft	274387	275900	277578	279139	280439	281761	283063	284390	285696	286920	288097	28
16	Schaffhausen	75737	75716	75748	75769	75835	75932	76077	76236	76364	76477	76592	7
17	Appenzell A.Rh.	53019	53041	53203	53351	53469	53611	53744	53906	54055	54204	54370	5

Daten kopieren und transponieren



bevoelkerungswachstum_showreal.csv

Datei Bearbeiten Ansicht Einfügen

Arial 10

F32

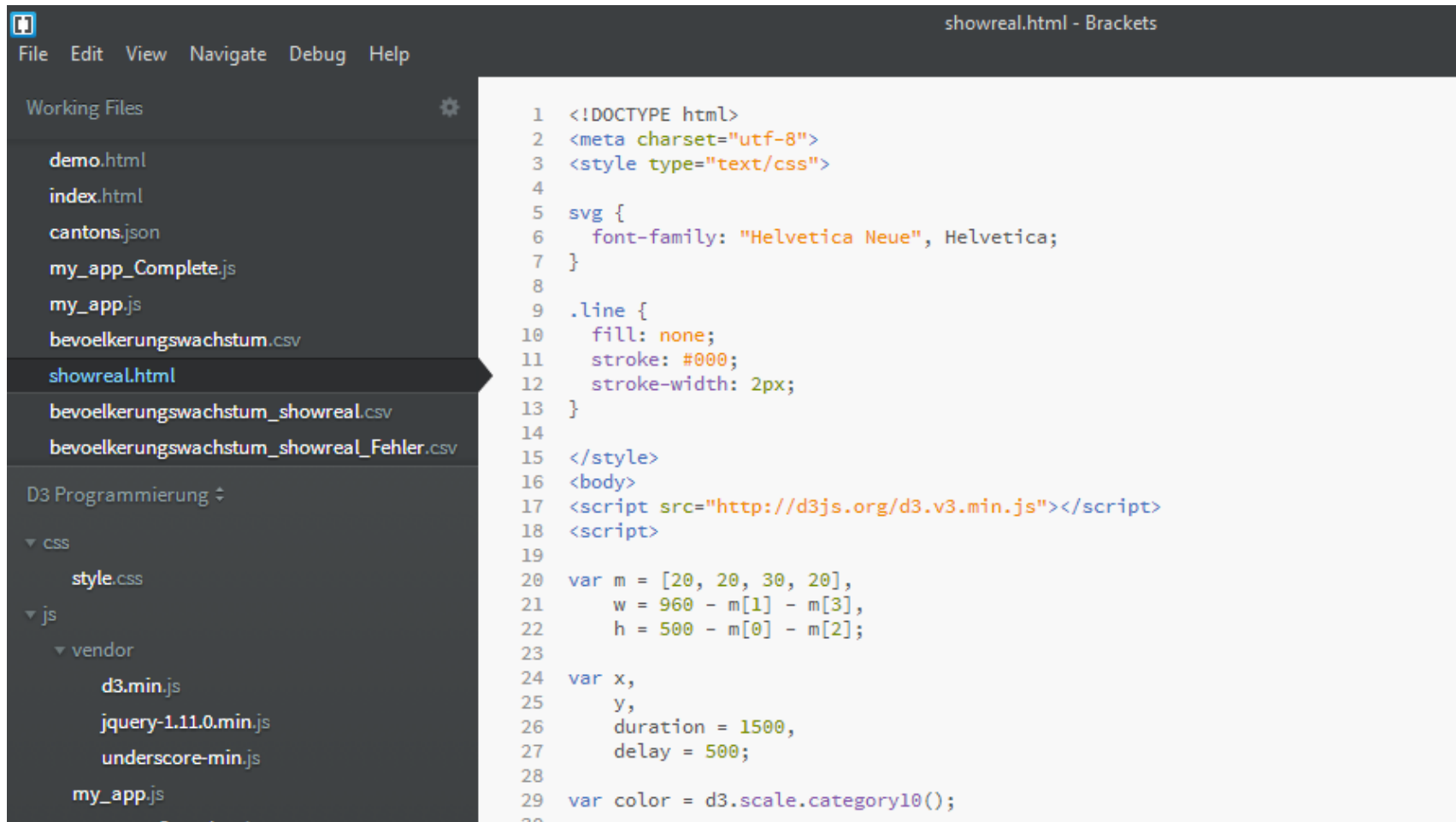
	A	B	C
1	Kanton	Jahr	Bevölkerung
2	Zürich	2010	1368669
3	Zürich	2011	1382998
4	Zürich	2012	1392745
5	Zürich	2013	1402327
6	Zürich	2014	1413343
7	Zürich	2015	1424093
8	Zürich	2016	1434815
9	Zürich	2017	1445692
10	Zürich	2018	1455988
11	Zürich	2019	1465920
12	Zürich	2020	1475482
13	Zürich	2021	1484599
14	Zürich	2022	1493302
15	Zürich	2023	1501572
16	Zürich	2024	1509404
17	Zürich	2025	1516809
18	Zürich	2026	1523809
19	Zürich	2027	1530507
20	Zürich	2028	1536801
21	Zürich	2029	1542776
22	Zürich	2030	1548413
23	Zürich	2031	1553805
24	Zürich	2032	1558908
25	Zürich	2033	1563769
26	Zürich	2034	1568397
27	Zürich	2035	1572841
28	Bern	2010	977690
29	Bern	2011	980541
30	Bern	2012	983815

Anpassen bestehender D3.js Snippets

Vorgehen:

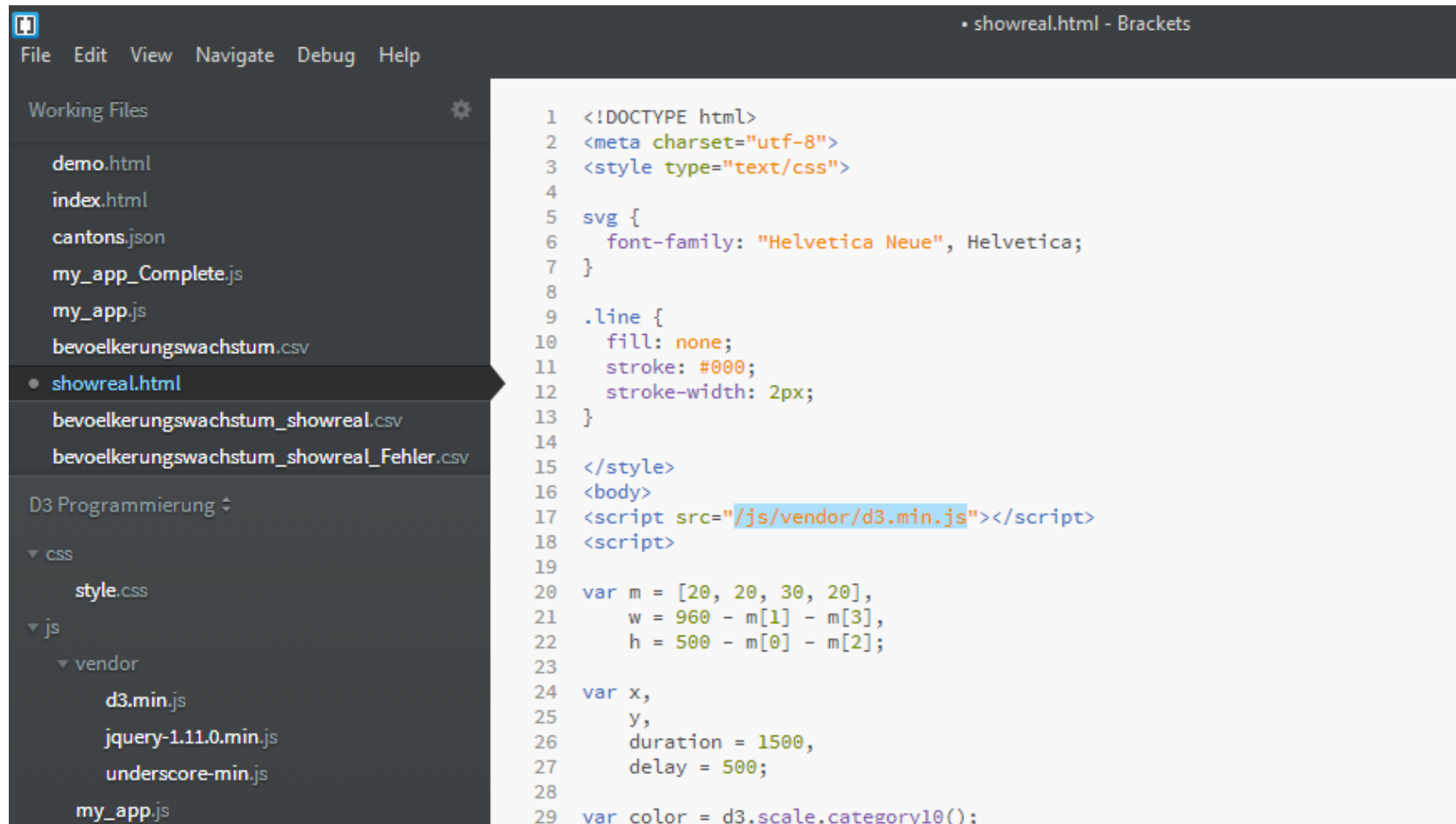
1. Welche Daten werden visualisiert? -> Bevölkerungswachstum
2. Welche Visualisierungsart macht Sinn? -> D3.js Show Real
3. Wie muss ich die Daten anpassen? -> Transponieren
4. **Was muss ich am Code Snippet anpassen -> Pfade**
5. Was läuft schief? -> Debugging

Code Snippet in Bracket kopieren



```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <style type="text/css">
4
5   svg {
6     font-family: "Helvetica Neue", Helvetica;
7   }
8
9   .line {
10     fill: none;
11     stroke: #000;
12     stroke-width: 2px;
13   }
14
15 </style>
16 <body>
17 <script src="http://d3js.org/d3.v3.min.js"></script>
18 <script>
19
20   var m = [20, 20, 30, 20],
21       w = 960 - m[1] - m[3],
22       h = 500 - m[0] - m[2];
23
24   var x,
25       y,
26       duration = 1500,
27       delay = 500;
28
29   var color = d3.scale.category10();
30
```

Pfad zu D3.js Bibliothek anpassen



```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <style type="text/css">
4
5   svg {
6     font-family: "Helvetica Neue", Helvetica;
7   }
8
9   .line {
10     fill: none;
11     stroke: #000;
12     stroke-width: 2px;
13   }
14
15 </style>
16 <body>
17 <script src="/js/vendor/d3.min.js"></script>
18 <script>
19
20   var m = [20, 20, 30, 20],
21       w = 960 - m[1] - m[3],
22       h = 500 - m[0] - m[2];
23
24   var x,
25       y,
26       duration = 1500,
27       delay = 500;
28
29   var color = d3.scale.category10();
```

Pfad zu CSV-Daten anpassen

The screenshot shows the Brackets code editor interface. On the left, the 'Working Files' sidebar lists several files, including 'showreal.html' which is currently selected. Below this, a file explorer shows the project structure with folders for 'css' and 'js', and various files like 'style.css', 'd3.min.js', and 'my_app.js'. The main editor area displays a search for 'CSV' with '2 results' found. The code snippet shown is a JavaScript file using D3.js to load and parse CSV data from 'stocks.csv' and generate an area chart. The code includes comments and function calls for data loading, parsing, and visualization.

```

52 // A area generator, for the dark stroke.
53 var area = d3.svg.area()
54   .interpolate("basis")
55   .x(function(d) { return x(d.date); })
56   .y1(function(d) { return y(d.price); });
57
58 d3.csv("stocks.csv", function(data) {
59   var parse = d3.time.format("%b %Y").parse;
60
61   // Nest stock values by symbol.
62   symbols = d3.nest()
63     .key(function(d) { return d.symbol; })
64     .entries(stocks = data);
65
66   // Parse dates and numbers. We assume values are sorted by date.
67   // Also compute the maximum price per symbol, needed for the y-domain.
68   symbols.forEach(function(s) {
69     s.values.forEach(function(d) { d.date = parse(d.date); d.price = +d.price;
70     s.maxPrice = d3.max(s.values, function(d) { return d.price; });
71     s.sumPrice = d3.sum(s.values, function(d) { return d.price; });
72   });
73
74   // Sort by maximum price, descending.
75   symbols.sort(function(a, b) { return b.maxPrice - a.maxPrice; });
76
77   var g = svg.selectAll("g")
78     .data(symbols)
  
```


Pfad zu CSV-Daten anpassen

```

49     .x(function(d) { return x(d.date); })
50     .y(h);
51
52 // A area generator, for the dark stroke.
53 var area = d3.svg.area()
54     .interpolate("basis")
55     .x(function(d) { return x(d.date); })
56     .y1(function(d) { return y(d.price); });
57
58 d3.csv("bevoelkerungswachstum_showreal.csv", function(data) {
59     var parse = d3.time.format("%b %Y").parse;
60
61     // Nest stock values by symbol.
62     symbols = d3.nest()
63         .key(function(d) { return d.symbol; })
64         .entries(stocks = data);
65
66     // Parse dates and numbers. We assume values are sorted by date.
67     // Also compute the maximum price per symbol, needed for the y-domain.
68     symbols.forEach(function(s) {
69         s.values.forEach(function(d) { d.date = parse(d.date); d.price = +d.price;
70         s.maxPrice = d3.max(s.values, function(d) { return d.price; });
71         s.sumPrice = d3.sum(s.values, function(d) { return d.price; });
72     });
73
74     // Sort by maximum price, descending.
75     symbols.sort(function(a, b) { return b.maxPrice - a.maxPrice; });
76
77     var g = svg.selectAll("g")
78         .data(symbols)
  
```

Anpassen bestehender D3.js Snippets

Vorgehen:

1. Welche Daten werden visualisiert? -> Bevölkerungswachstum
2. Welche Visualisierungsart macht Sinn? -> D3.js Show Real
3. Wie muss ich die Daten anpassen? -> Transponieren
4. Was muss ich am Code Snippet anpassen -> Pfade
5. **Was läuft schief? -> Debugging**

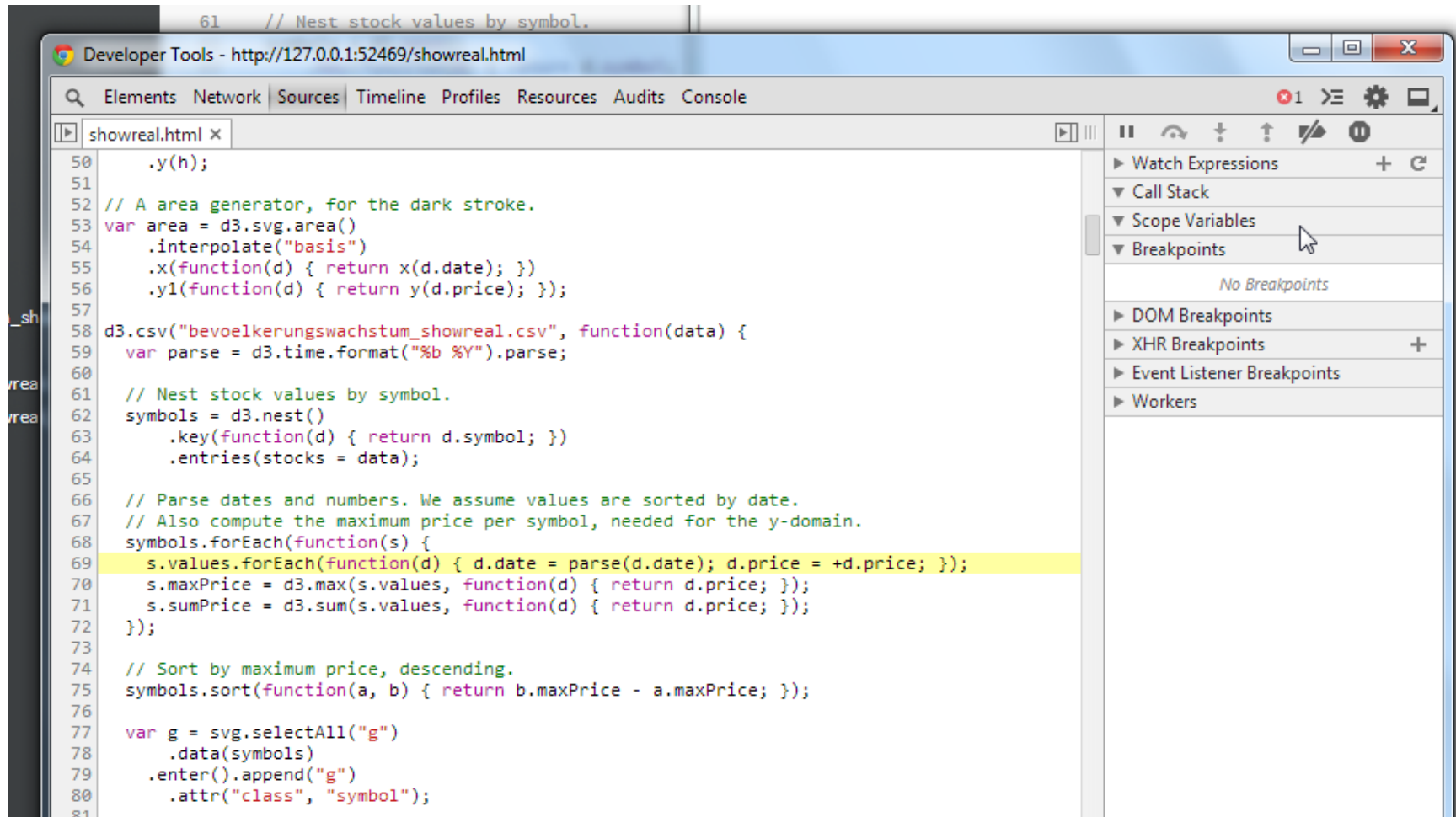
A) Gar keine Darstellung -> Console

The screenshot shows a web browser window at `127.0.0.1:52469/showreal.html` and its developer tools. The code editor on the left shows the `showreal.html` file with the following JavaScript code:

```
42 .interpolate("basis")
43 .x(function(d) { return x(d.date); })
44 .y(function(d) { return y(d.price); })
45
46 // A line generator, for the dark stroke.
47 var axis = d3.svg.line()
48   .interpolate("basis")
49   .x(function(d) { return x(d.date); })
50   .y(h);
51
52 // A area generator, for the dark stroke.
53 var area = d3.svg.area()
54   .interpolate("basis")
55   .x(function(d) { return x(d.date); })
56   .y1(function(d) { return y(d.price); })
57
58 d3.csv("bevoelkerungswachstum_showreal.csv",
59       function(parse) {
60         var parse = d3.time.format("%b %Y").parse
61         // Nest stock values by symbol.
```

The developer tools console shows an error: `Uncaught TypeError: Cannot read property 'length' of undefined`. The stack trace indicates the error occurred in `d3.min.js:1` at line 69, specifically in the `t.parse` function. The console also shows the source file `http://127.0.0.1:52469/showreal.html:69`.

Fehler in der Beschriftung der Spalten



```
61 // Nest stock values by symbol.
62
63 d3.csv("bevoelkerungswachstum_showreal.csv", function(data) {
64   var parse = d3.time.format("%b %Y").parse;
65
66   // Nest stock values by symbol.
67   symbols = d3.nest()
68     .key(function(d) { return d.symbol; })
69     .entries(stocks = data);
70
71   // Parse dates and numbers. We assume values are sorted by date.
72   // Also compute the maximum price per symbol, needed for the y-domain.
73   symbols.forEach(function(s) {
74     s.values.forEach(function(d) { d.date = parse(d.date); d.price = +d.price; });
75     s.maxPrice = d3.max(s.values, function(d) { return d.price; });
76     s.sumPrice = d3.sum(s.values, function(d) { return d.price; });
77   });
78
79   // Sort by maximum price, descending.
80   symbols.sort(function(a, b) { return b.maxPrice - a.maxPrice; });
81
82   var g = svg.selectAll("g")
83     .data(symbols)
84     .enter().append("g")
85     .attr("class", "symbol");
```

Fehler in der Beschriftung der Spalten

bevoelkerungswachstum_showreal.csv -

Datei Bearbeiten Ansicht Einfügen

Arial 10

F32

	A	B	C
	Kanton	Jahr	Bevölkerung
2	Zürich	2010	1388669
3	Zürich	2011	1382998
4	Zürich	2012	1392745
5	Zürich	2013	1402327
6	Zürich	2014	1413343
7	Zürich	2015	1424093
8	Zürich	2016	1434815
9	Zürich	2017	1445692
10	Zürich	2018	1455988
11	Zürich	2019	1465920
12	Zürich	2020	1475482



bevoelkerungswachstum_showreal

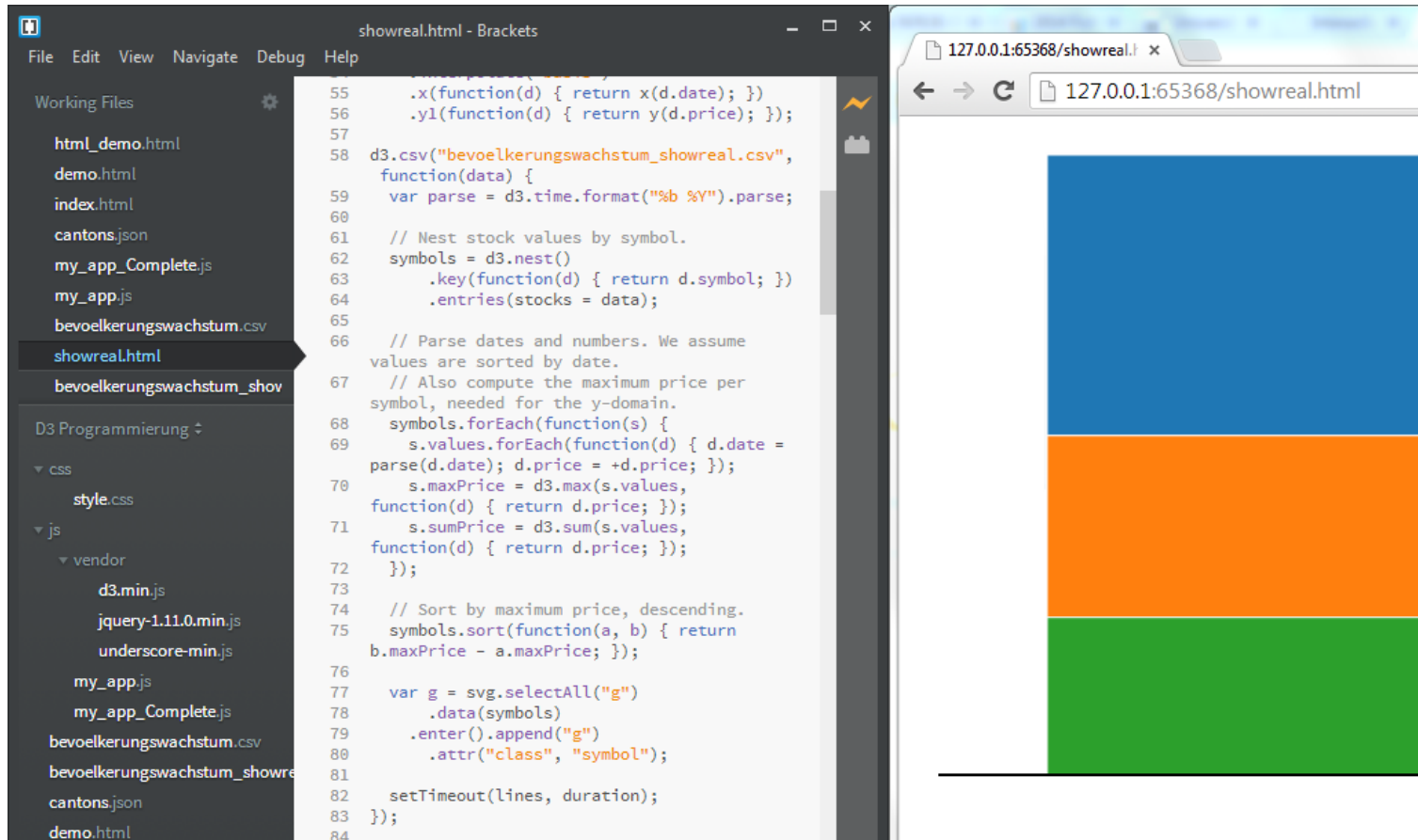
Datei Bearbeiten Ansicht Einfügen

Arial 10

J39

	A	B	C
1	symbol	date	price
2	Zürich	2010	1388669
3	Zürich	2011	1382998
4	Zürich	2012	1392745
5	Zürich	2013	1402327
6	Zürich	2014	1413343
7	Zürich	2015	1424093
8	Zürich	2016	1434815
9	Zürich	2017	1445692
10	Zürich	2018	1455988
11	Zürich	2019	1465920
12	Zürich	2020	1475482

B) Fehlerhafte Darstellung



Falsches Datumsformat

```
55     .x(function(d) { return x(d.date); });
56     .y1(function(d) { return y(d.price); });
57
58     d3.csv("bevoelkerungswachstum_chowreal.csv", function(data) {
59         var parse = d3.time.format("%b %Y").parse;
60
61         // Nest stock values by symbol.
62         symbols = d3.nest()
63             .key(function(d) { return d.symbol; })
64             .entries(stocks = data);
65
66         // Parse dates and numbers. We assume values are sorted by date.
67         // Also compute the maximum price per symbol, needed for the y-domain.
68         symbols.forEach(function(s) {
```

Korrektes Datumsformat

```
56     .y1(function(d) { return y(d.price); });
57
58     d3.csv("bevoelkerungswachstum_show.html.csv", function(data) {
59         var parse = d3.time.format("%Y").parse;
60
61         // Nest stock values by symbol.
62         symbols = d3.nest()
63             .key(function(d) { return d.symbol; })
64             .entries(stocks = data);
65
```

Jetzt klappts!

