

Week 3 – Systems and Toolkits

Yasmin AlNoamany

L&S 88-2

University of California, Berkeley

Announcements

- Assignment 1 grades posted
- No lecture next time
 - readings and discussions
 - assignment 3
- Semester Project - [Link](#)

Previous lecture

- Types of Web data
- Introduction to Web Mining
 - Web content mining
 - Web usage mining
 - Web structure mining
- Getting data from the Web
 - Working with APIs
 - Climate change API
 - NYT API
 - Twitter API

And what if there is no
available API?

And what if there is no
available API?

Scraping!

Outlines

- Getting Data from the Web
 - Web scrapping
- Choosing Tools to Visualize Data
- Data preprocessing
 - Data Cleaning
- Tools for visualizing data
 - Intro to R and RStudio
 - D3 and Tableau, next lecture.

Resources

- Data Analysis and Visualization in R from Data Carpentry lessons
- Producing Simple Graphs with R
<https://www.harding.edu/fmccown/r/>
- Visualize This, Ch 2 (Data Scraping)
- Web Scraping in Python ([notebook](#))
- [What I Learned Recreating One Chart Using 24 Tools](#), by Lisa Charlotte

Objectives

- Identify how to scrape data from Web
- Understand how HTML works with your browser to display a website
- Identify HTML tags and attributes
- Differentiate between different visualization tools and describe the type of tasks for which each tool might be most appropriate
- Use R and RStudio for data analysis and visualization

Web Scraping

- Extracting data from websites by extracting it directly from the HTML source code
- Web scraping a Web page involves
 - fetching a Web page (Web crawling)
 - Extracting data from it

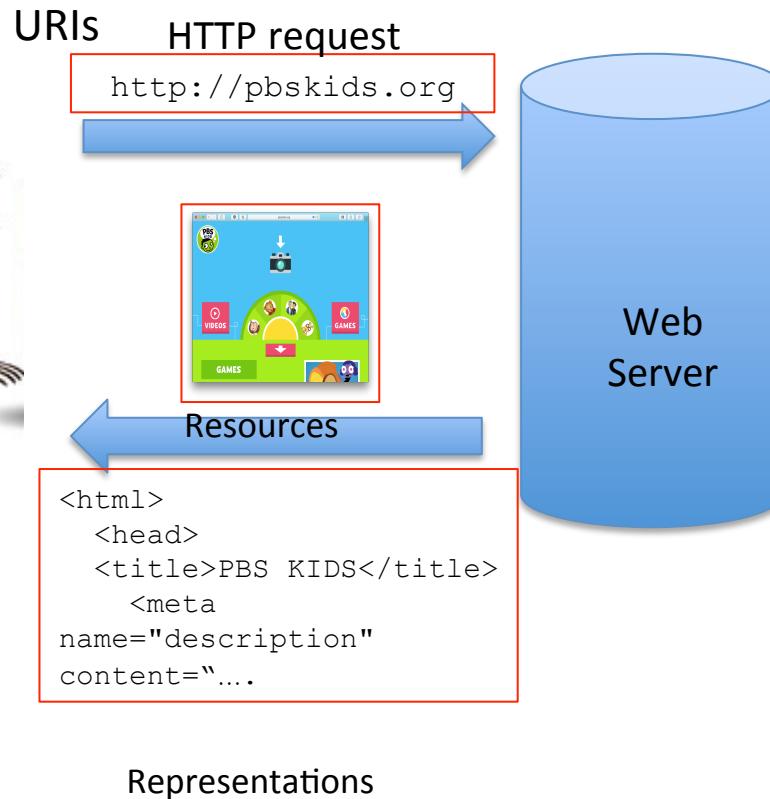
Why Web scrape

Tons of Web data useful for social scientists and humanists

- social media
- news media
- government publications
- organizational records

Basic Process

- Make a GET request
- Identify the patterns and parts of the HTML response
 - web page layout
- Iterate
 - visit each web page and grab the relevant data
 - if there are a lot of pages, this might take some time
- Store the data



Some disclaimers

- Check a site's terms and conditions before scraping
 - Some sites limit the rate of web requests you can make from a single IP
 - Be nice - don't hammer the site's server

You don't want to get blocked!

- Web scraping etiquette
 - Only one connection should be open to any given host at a time
 - A waiting time of a few seconds should occur between successive requests to a host

Robot.txt

- Robots exclusion standard
- Web site owners use the /robots.txt file to give instructions about their site to Web robots; this is called The Robots Exclusion Protocol.
 - <http://www.robotstxt.org/>
- Examples:
 - <https://www.facebook.com/robots.txt>
 - <http://www.cnn.com/robots.txt>
 - <http://www.ebay.com/robots.txt>

```
User-agent: *
Disallow: /
The "User-agent: *" means this section applies to all robots. The
"Disallow: /" tells the robot that it should not visit any pages on
the site.
```

Web scraping benefits and challenges

- Web scraping benefits
 - Any content that can be viewed on a webpage can be scraped.
 - No API needed
 - No rate-limiting or authentication (usually)
- Web scraping challenges
 - Sometimes the web site structure is difficult to parse
 - might want to download the HTML pages to a local disk before parsing
 - Sites change their layout all the time. Your scraper will break.

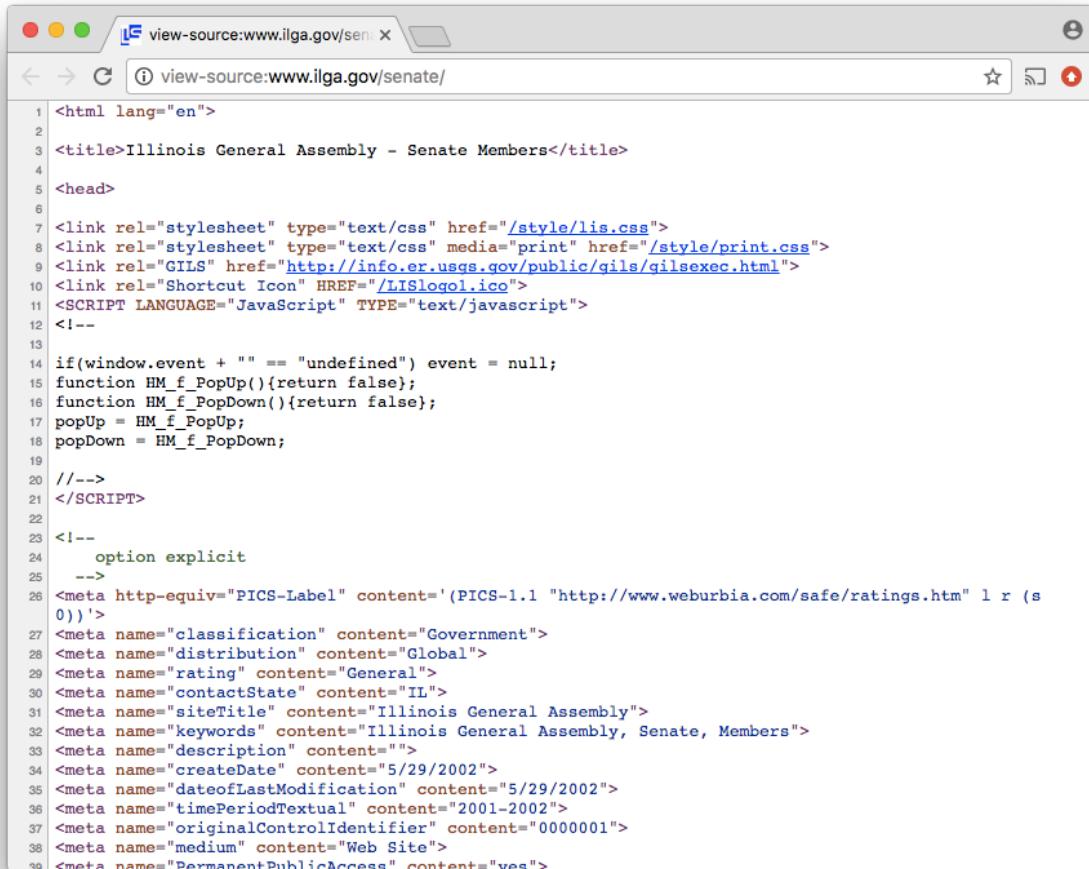
Web scraping benefits and challenges

- Web scraping benefits
 - Any content that can be viewed on a webpage can be scraped.
 - No API needed
 - No rate-limiting or authentication (usually)
- Web scraping challenges
 - Sometimes the web site structure is difficult to parse
 - might want to download the HTML pages to a local disk before parsing
 - Sites change their layout all the time. Your scraper will break.

Check for API first. If not available, scrape!

What is a Website

- Some combination of codebase, database
- The "front end" product is HTML + Cascading Style Sheets (CSS) + javascript



A screenshot of a web browser window displaying the source code of a website. The title bar shows 'view-source:www.ilga.gov/senate'. The code is a standard HTML document with some inline JavaScript and meta tags.

```
1 <html lang="en">
2
3 <title>Illinois General Assembly - Senate Members</title>
4
5 <head>
6
7 <link rel="stylesheet" type="text/css" href="/style/lis.css">
8 <link rel="stylesheet" type="text/css" media="print" href="/style/print.css">
9 <link rel="GILS" href="http://info.er.usgs.gov/public/gils/gilsexec.html">
10 <link rel="Shortcut Icon" HREF="LISLogo1.ico">
11 <SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
12 <!--
13
14 if(window.event + "" == "undefined") event = null;
15 function HM_f_PopUp(){return false;};
16 function HM_f_PopDown(){return false;};
17 popUp = HM_f_PopUp;
18 popDown = HM_f_PopDown;
19
20 //-->
21 </SCRIPT>
22
23 <!--
24   option explicit
25   -->
26 <meta http-equiv="PICS-Label" content='(PICS-1.1 "http://www.weburbia.com/safe/ratings.htm" l r (s 0))'>
27 <meta name="classification" content="Government">
28 <meta name="distribution" content="Global">
29 <meta name="rating" content="General">
30 <meta name="contactState" content="IL">
31 <meta name="siteTitle" content="Illinois General Assembly">
32 <meta name="keywords" content="Illinois General Assembly, Senate, Members">
33 <meta name="description" content="">
34 <meta name="createDate" content="5/29/2002">
35 <meta name="dateofLastModification" content="5/29/2002">
36 <meta name="timePeriodTextual" content="2001-2002">
37 <meta name="originalControlIdentifier" content="0000001">
38 <meta name="medium" content="Web Site">
39 <meta name="PermanentPublicAccess" content="yes">
```

<http://www.ilga.gov/senate/>

What is a Website

- Your browser turns that into a tidy layout

The screenshot shows a web browser window for the Illinois General Assembly Senate. The URL in the address bar is www.ilga.gov/senate/. The page title is "Illinois General Assembly". A sidebar on the left lists navigation links: Home, Legislation & Laws, Senate, House, My Legislation, Site Map, Members, Committees, Schedules, Journals, Transcripts, Rules, and Live Audio/Video. The main content area displays a table titled "Current Senate Members 100th General Assembly". The table has columns for Senator, Bills, Committees, District, and Party. The data shows 22 Republicans and 37 Democrats. The table includes links for each senator's name.

Senator	Bills	Committees	District	Party
Pamela J. Althoff	Bills	Committees	32	R
Neil Anderson	Bills	Committees	36	R
Omar Aquino	Bills	Committees	2	D
Jason A. Barickman	Bills	Committees	53	R
Scott M. Bennett	Bills	Committees	52	D
Jennifer Bertino-Tarrant	Bills	Committees	49	D
Daniel Biss	Bills	Committees	9	D
Tim Bivins	Bills	Committees	45	R
William E. Brady	Bills	Committees	44	R
Melinda Bush	Bills	Committees	31	D
Cristina Castro	Bills	Committees	22	D
James F. Clayborne, Jr.	Bills	Committees	57	D
Jacqueline Y. Collins	Bills	Committees	16	D
Michael Connelly	Bills	Committees	21	R
John J. Cullerton	Bills	Committees	6	D
Thomas Cullerton	Bills	Committees	23	D
Bill Cunningham	Bills	Committees	18	D
John F. Curran	Bills	Committees	41	R

Basic strategy of Web scraping:

- Find out what kind of HTML element your data is in (use your browser’s “inspector”)
- Think about how you can differentiate those elements from other, similar elements in the webpage using CSS
- Use Python and add-on modules like BeautifulSoup to extract just that data

HTML: Basic structure

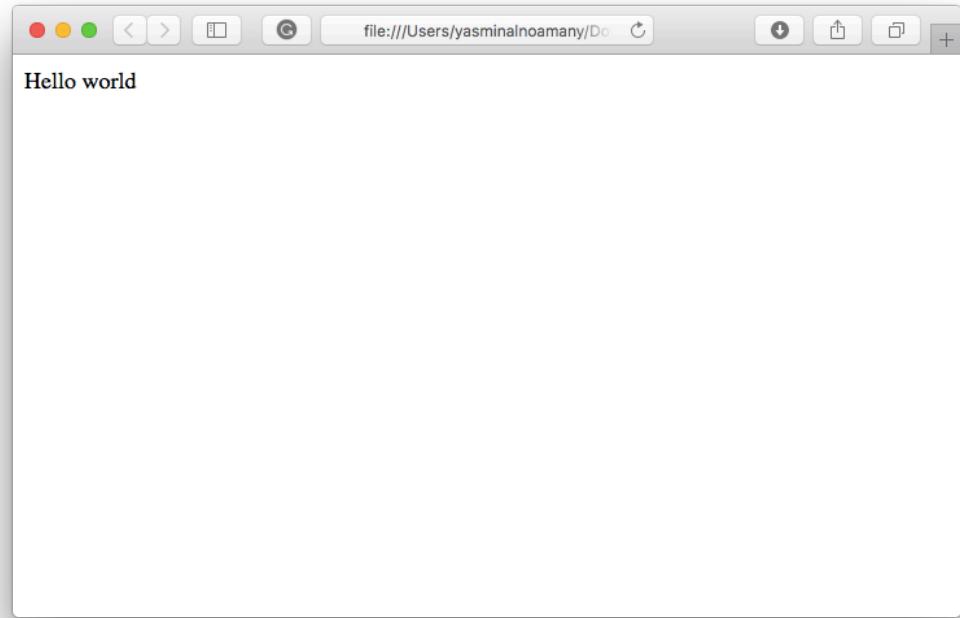
```
<DOCTYPE html>
<html>
  <head>
    <title>Page title</title>
  </head>

  <body>
    <p>Hello world</p>
  </body>
</html>
```

HTML: Basic structure

```
<DOCTYPE html>
<html>
  <head>
    <title>Page title</title>
  </head>

  <body>
    <p>Hello world</p>
  </body>
</html>
```



HTML Elements

HTML element has three components:

- Tags (starting and ending the element)
- Attributes (giving information about the element)
- Text, or Content (the text inside the element)



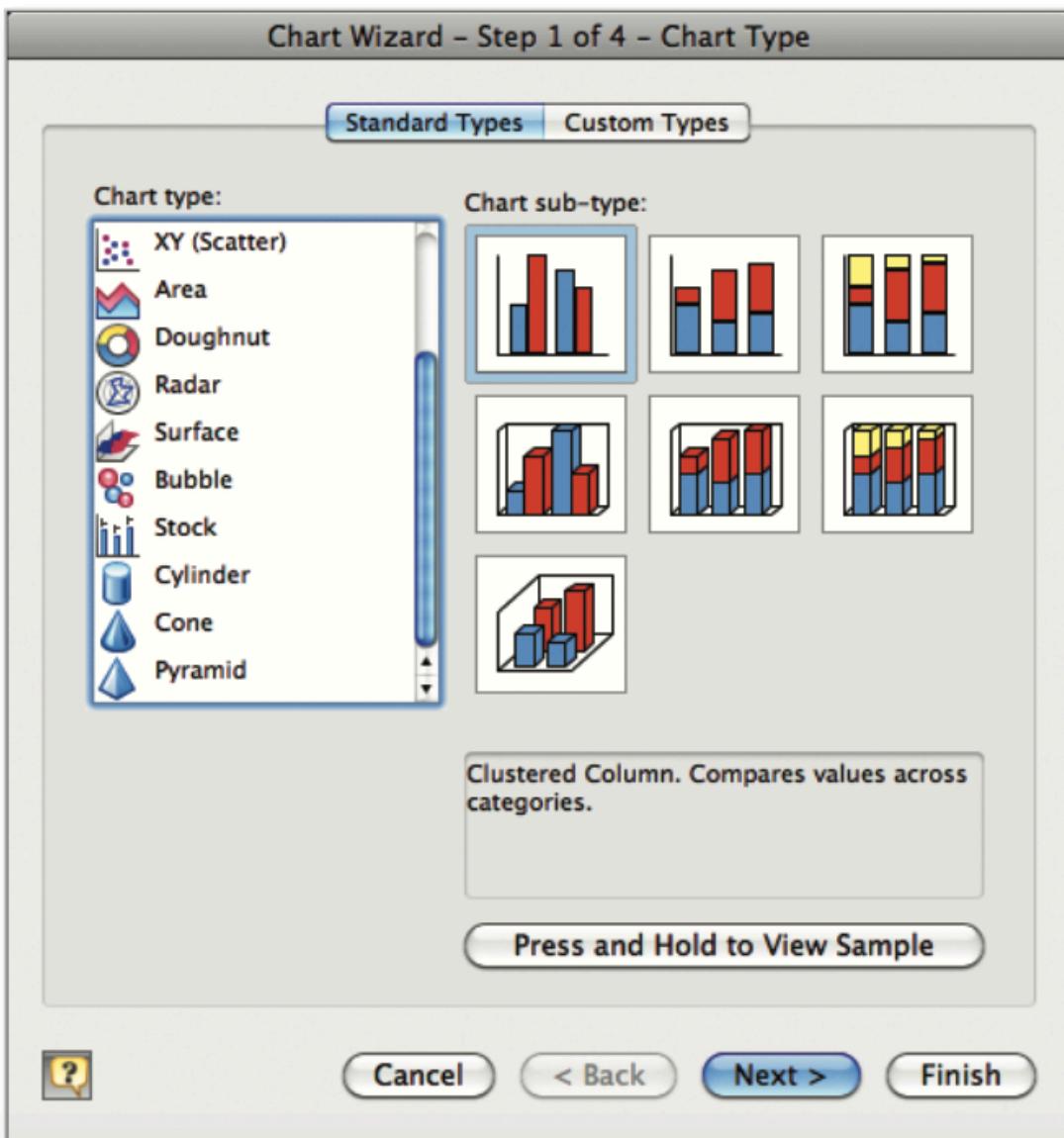
Web Scraping Example - Check Nathan Yau, Visualize This, Ch 2

- Let's grab data from the Weather Underground
 - <http://wunderground.com>
- Want temperature data for Berkeley for 2011
 - starting point: <http://www.wunderground.com/history/airport/ORF/2011/1/1/DailyHistory.html>
- Tools
 - Python
 - Beautiful Soup - Python script for reading web pages (<http://www.crummy.com/software/BeautifulSoup>)

Example

Data Visualization Tools

Excel



R

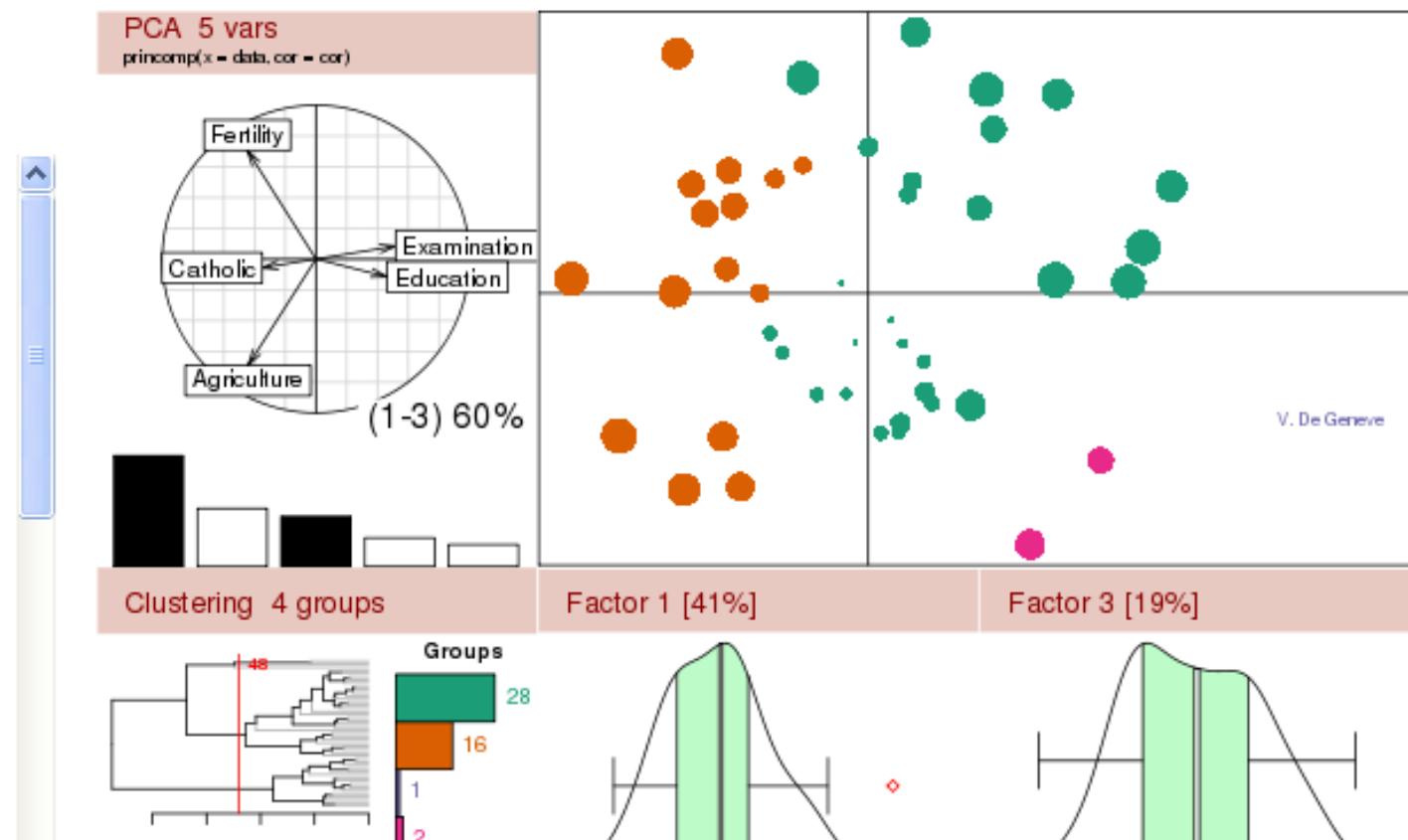


[About R](#)
[What is R?](#)
[Contributors](#)
[Screenshots](#)
[What's new?](#)

[Download,](#)
[Packages](#)
[CRAN](#)

[R Project](#)
[Foundation](#)
[Members &](#)
[Donors](#)
[Mailing Lists](#)
[Bug Tracking](#)

The R Project for Statistical Computing



Python - Matplotlib



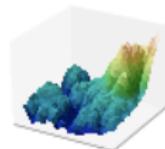
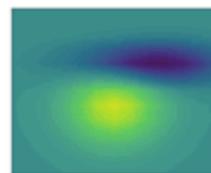
Fork me on GitHub

[home](#) | [examples](#) | [gallery](#) | [pyplot](#) | [docs](#) »

[modules](#) | [index](#)

Introduction

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.



Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code. For a sampling, see the [screenshots](#), [thumbnail gallery](#), and [examples](#) directory

[Depsy](#) 100th percentile

Travis-CI: [build](#) [error](#)

[Support matplotlib](#)

[Support NumFOCUS](#)

Quick search

Go

D3



Data-Driven Documents

A small orange rectangular button with white text that reads "Follow me on GitHub".



Tableau



Products Solutions Learning Community Support About

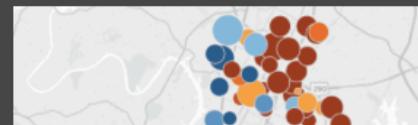
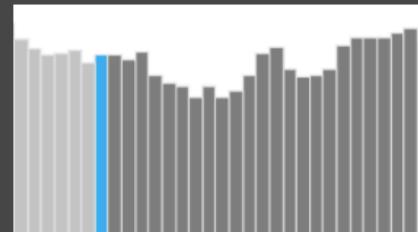
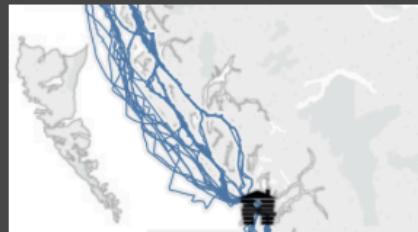
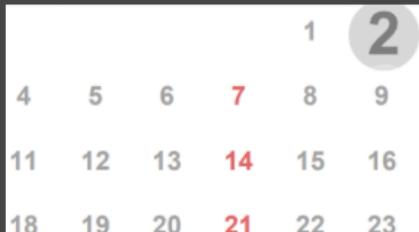
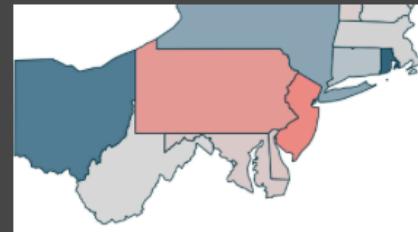
PRICING SIGN IN

TRY NOW



Tableau Viz Gallery

You can create almost any type of visualization with Tableau. See what's possible, or [try for yourself](#).



Many, many others

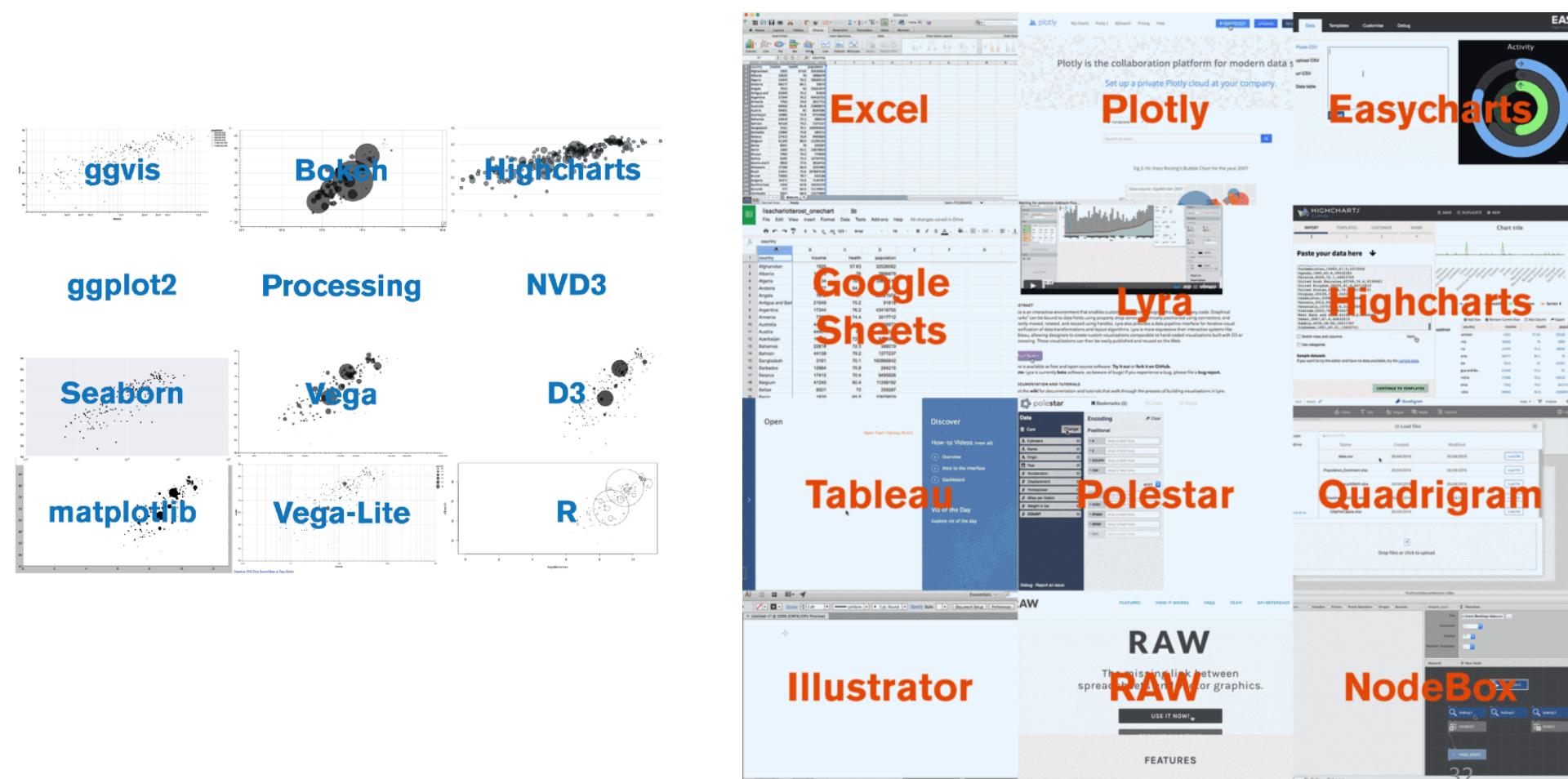
<http://selection.dataviz.ch>

The screenshot shows a web browser displaying a curated list of data visualization tools. The page has a red header bar with the URL "selection.dataviz.ch". Below the header, there's a navigation bar with tabs: "All", "Maps", "Charts", "Data", and "Color".

The main content area is a grid of tool cards, each containing a thumbnail, the tool's name, and a brief description. The tools listed include:

- Arbor.js**: A library of force-directed layout algorithms plus abstractions for graph organization and refresh handling.
- CartoDB**: A web service for mapping, analyzing and building applications with data.
- Chroma.js**: An interactive color space explorer that allows to preview a set of linear interpolated equidistant colors.
- Circos**: A software package for visualizing data in a circular layout.
- Cola.js**: A library for arranging networks using constraint-based optimization techniques.
- ColorBrewer**: A web tool for selecting colors for maps.
- Cubism.js**: A library for creating interactive time series and horizon graphs based on D3.js.
- Cytoscape**: An application for visualizing complex networks and integrating these with any type of data.
- D3.js**: An small, flexible and efficient library to create and manipulate interactive documents based on data.
- Dance.js**: A simple data-driven visualization framework based on Data.js and Underscore.js.
- Data.js**: A data representation framework providing a uniform interface to domain data.
- DataWrangler**: An interactive web application for data cleaning and transformation.
- Degrafa**: A powerful declarative graphics framework for rich user interfaces, data visualizations and animations.
- Envision.js**: A library for creating fast, dynamic and interactive time series visualizations.
- Flare**: A set of software tools for creating rich interactive data visualizations in ActionScript.
- GeoCommons**: A public community and set of tools to access, visualize and analyze data with compelling map visualizations.
- Gephi**: A visualization and exploration platform for networks with dynamic and hierarchical graphs.
- Google Chart Tools**: A collection of simple to use, customizable and free to use interactive charts and data tools.
- Google Fusion Tables**: A web application that makes it easy to host, manage, collaborate on, visualize, and publish data.
- I Want Hue**: A web application to generate and refine palettes of optimally distinct colors.
- JavaScript InfoVis Toolkit**: A JavaScript library that provides tools for creating interactive data visualizations.
- Kartograph**: A simple and lightweight framework for creating beautiful, interactive vector maps.
- Leaflet**: A lightweight JavaScript library for making tile-based interactive maps for desktop and mobile browsers.
- Many Eyes**: A web application to build, share and discuss graphic representations of user uploaded data.
- MapBox**: A web platform for hosting custom designed map tiles and a set of open source tools to build interactive maps.
- Miso**: A toolkit to expedite the creation of interactive storytelling and data visualisation content.
- Modest Maps**: A display and interaction library for tile-based maps in Flash, JavaScript and Python.
- Mr. Data Converter**: A simple console that converts Excel data into web-friendly formats, including HTML, JSON and XML.
- Mr. Nester**: A simple console for learning and experimenting with d3.js data nesting.
- NVD3.js**: A collection of re-usable charts and chart components for d3.js.
- NodeBox**: A desktop application that lets you create generative, static, animated or interactive visualizations.
- OpenRefine**: A tool for working with data, cleaning it up, reformating it or extending it with web programming.
- Paper.js**: A vector graphics scripting framework in a well designed, consistent and clean programming language.
- Polymaps**: A library for making dynamic, interactive maps with image- and vector-based tiles.
- Processing**: An open source programming language and environment to create images.
- Processing.js**: The sister project of Processing that makes projects work using web standards and without Java.
- Protovis**: A library that composes custom views of data with simple marks such as bars and dots.
- Quadrigram**: A visual programming language aimed to gather, process and visualize information.
- R**: A software environment for statistical computing and graphical techniques.
- Raphaël**: A small library that simplifies working with vector graphics on the web.
- Raw**: An application to create custom vector-based visualizations on top of D3.js.
- Recline.js**: A simple but powerful library for building data applications in pure JavaScript and HTML.
- Rickshaw**: A library for creating interactive time series graphs based on D3.js.

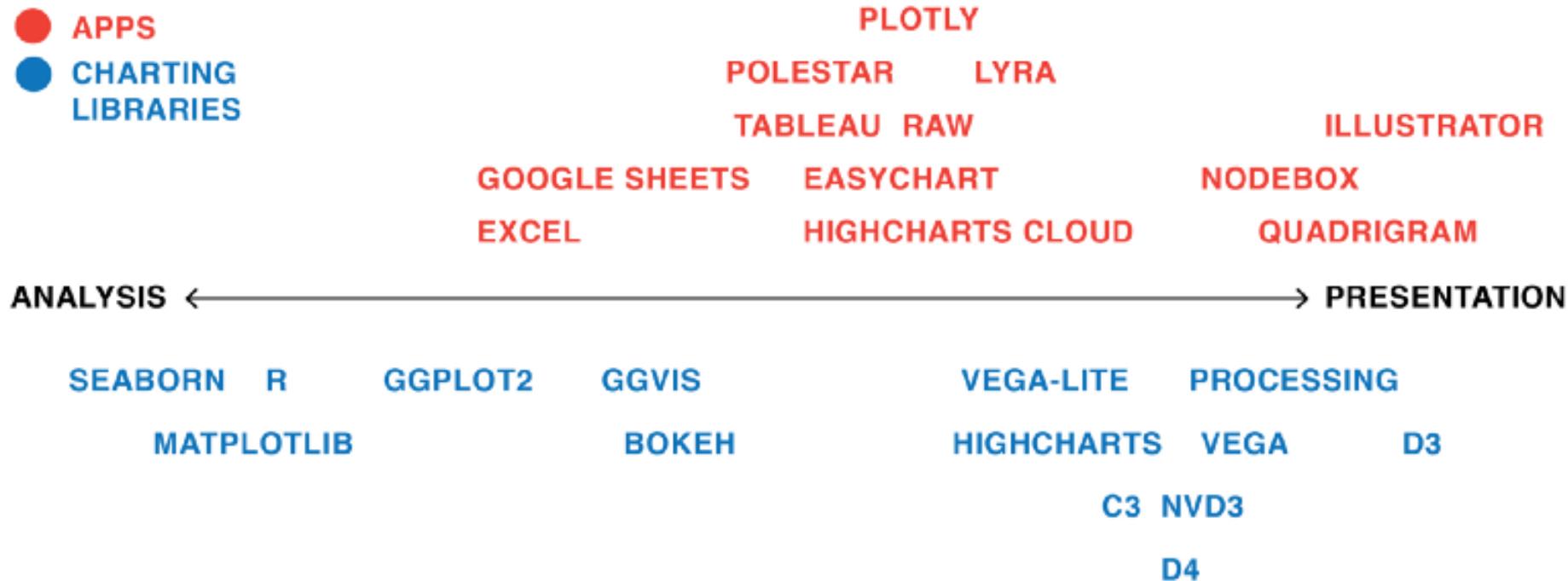
What I Learned Recreating One Chart Using 24 Tools



Finding the best tool means thinking hard about your goals and needs.

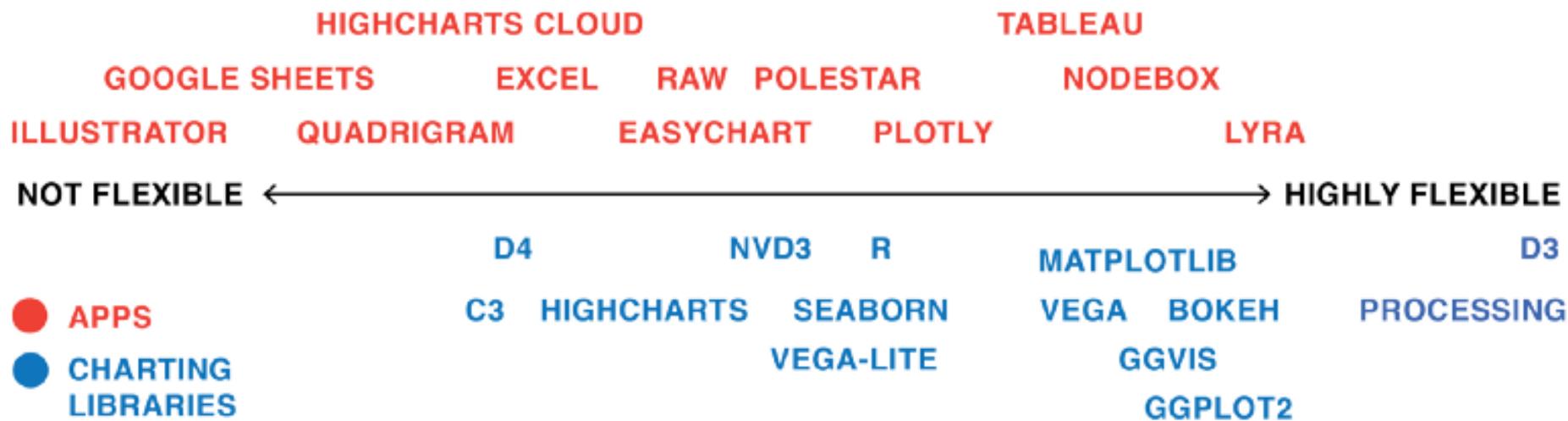
Tools for Analysis vs. Presentation

- APPS
- CHARTING LIBRARIES



Do you want to use your tool to explore the data (R, Python) or do you want to build visualizations for the public (D3.js, Illustrator)?

Flexibility of Tools



Do you just need basic chart types like a bar chart or line chart (Highcharts, Excel); or do you want to create crazy chart magic (D3.js)?

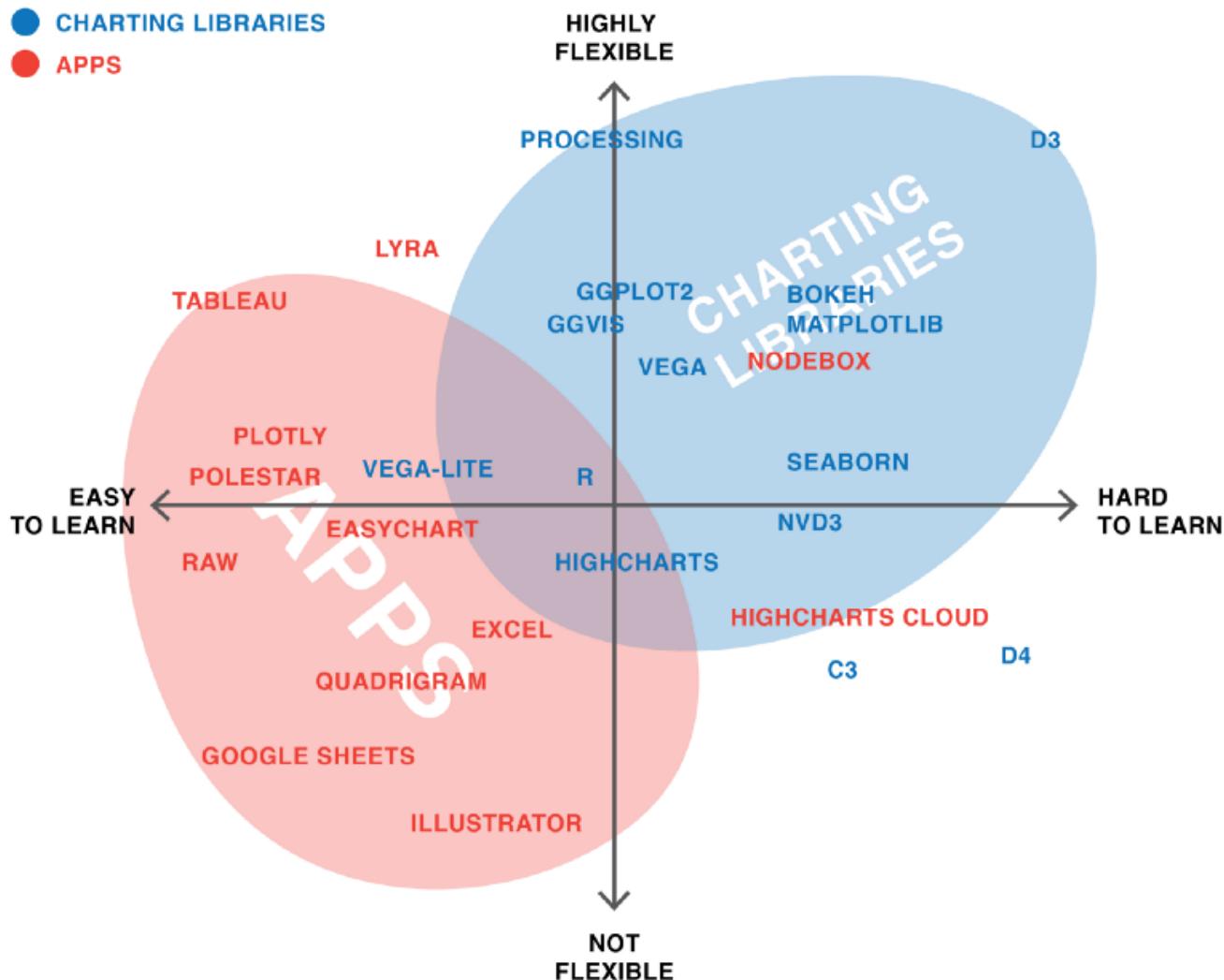
Data Management

- Will you change your data (eg. changing one or all values or adding rows or columns) while creating a data visualization?
 - The least convenient in some apps (like Illustrator) you need to start building the graph all over again when changing the data only slightly
 - The medium-convenient way is to change the data outside of the tool, then import it (again) and update the visualization with the new data. D3.js is an example for this way
 - Other apps make it even more convenient than code to deal with data. Once you import your data, you can change it or add new, transformed columns directly in the tool (examples: Plotly and Lyra)

Interactivity vs. static

	STATIC	WEB - INTERACTIVE
APPS	ILLUSTRATOR, NODEBOX, EXCEL, POLESTAR, RAW	HIGHCHARTS CLOUD, QUADRIGRAM, EASYCHRT, DATAWRAPPER, TABLEAU, PLOTLY, GOOGLE SHEETS
CHARTING LIBRARIES	GGPLOT2, MATPLOTLIB, R, SEABORN, BOKEH, PROCESSING	D3, D4, C3, NVD3, GGVIS, HIGHCHARTS, SHINY, VEGA, VEGA-LITE

We Still Live in an “Apps Are for the Easy Stuff, Code Is for the Good Stuff” World



There Are No Perfect Tools, Just Good Tools for People with Certain Mindsets

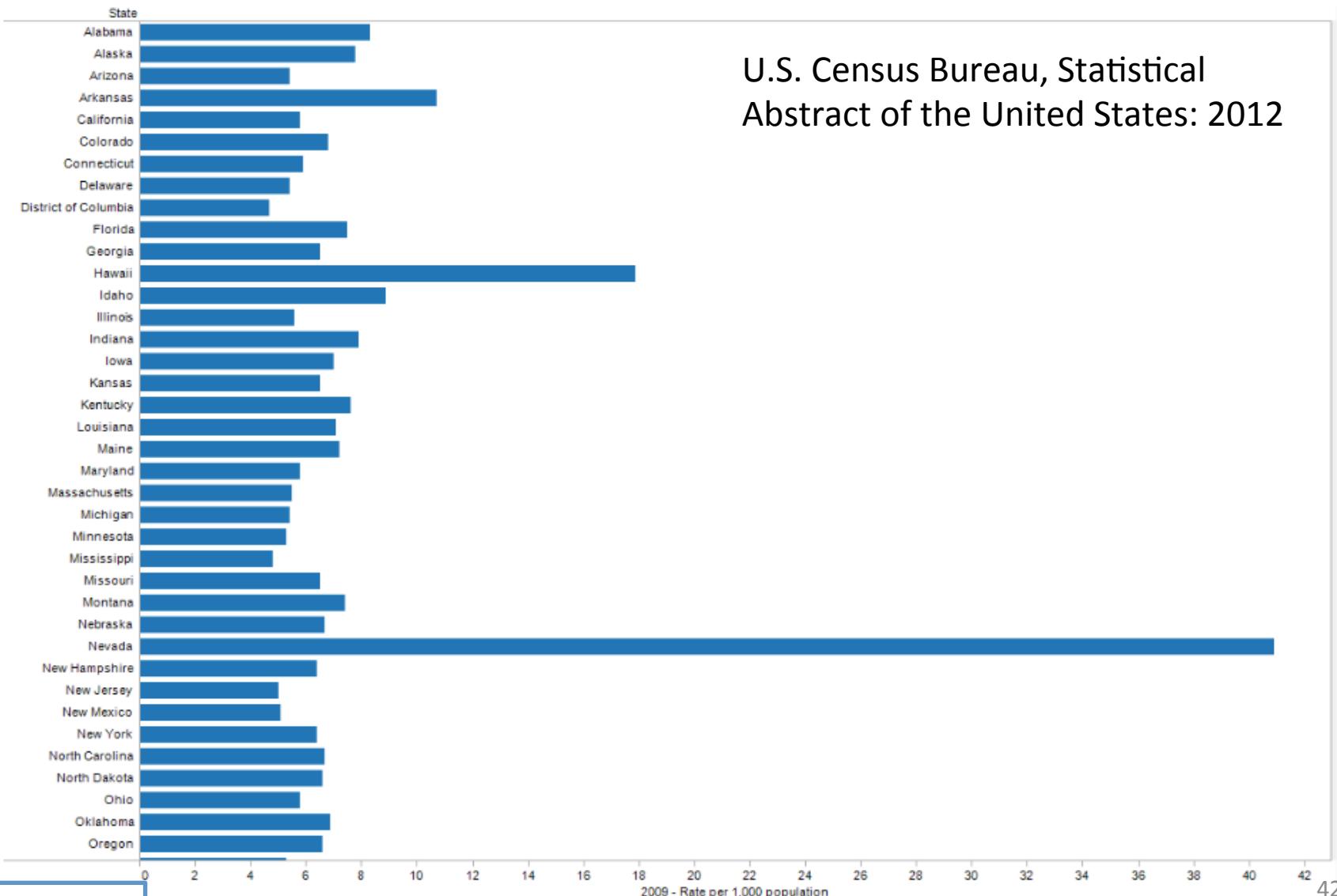
Data Cleaning

Data cleaning

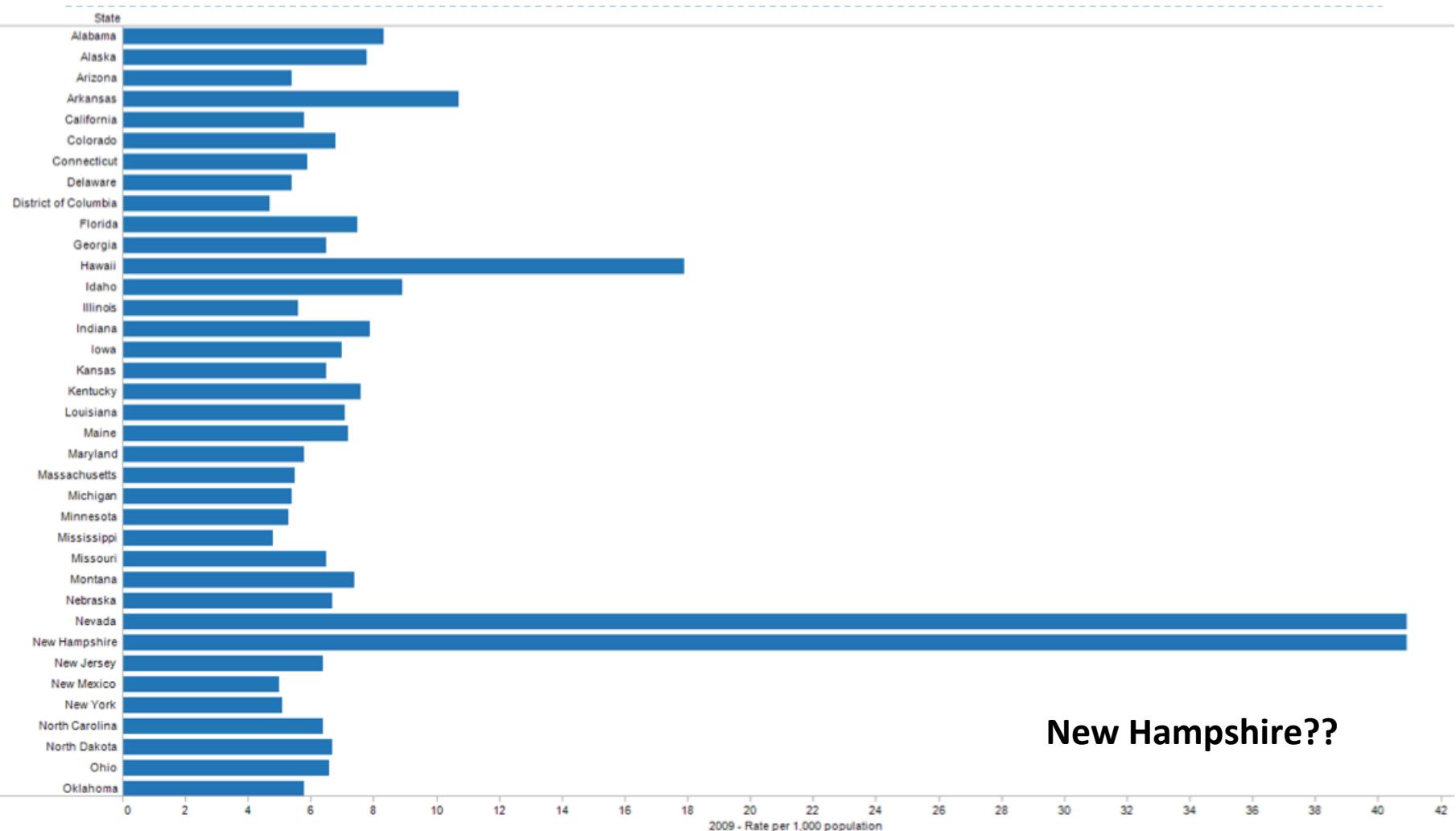
- Data in the real world tend to be:
 - Noisy: containing errors or outliers
 - Incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - Inconsistent: containing discrepancies in codes or names
- Data cleaning tend to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in data
- Before you can visualize something, you have to get the data into a form that you can work with

Marriage Rate per State

U.S. Census Bureau, Statistical
Abstract of the United States: 2012



Graphing the raw data



Now, let's look at the PDF

	Number (1,000)			Rate per 1,000 population ²		
	1990	2000	2009	1990	2000	2009
Maryland	46.3	40.0	32.4	9.7	7.5	5.8
Massachusetts	47.7	37.0	36.7	7.9	5.8	5.5
Michigan	76.1	66.4	53.1	8.2	6.7	5.4
Minnesota	33.7	33.4	28.4	7.7	6.8	5.3
Mississippi	24.3	19.7	14.5	9.4	6.9	4.8
Missouri	49.1	43.7	39.8	9.6	7.8	6.5
Montana	6.9	6.6	7.1	8.6	7.3	7.4
Nebraska	12.6	13.0	12.5	8.0	7.6	6.7
Nevada	120.6	144.3	108.2	99.0	72.2	40.9
New Hampshire	10.5	11.6	8.5	9.5	9.4	6.4
New Jersey	58.7	50.4	46.3	7.6	6.0	5.0
New Mexico ⁵	13.3	14.5	10.2	8.8	8.0	5.1
New York ⁵	154.8	162.0	120.1	8.6	7.1	6.4
North Carolina	51.9	65.6	65.8	7.8	8.2	6.7
North Dakota	4.8	4.6	4.3	7.5	7.2	6.6
Ohio	98.1	88.5	64.8	9.0	7.8	5.8
Oklahoma	33.2	15.6	23.5	10.6	(NA)	6.9
Oregon	25.3	26.0	23.5	8.9	7.6	6.6
Pennsylvania	84.9	73.2	64.2	7.1	6.0	5.3
Rhode Island	8.1	8.0	6.5	8.1	7.6	5.9

Marriage Rate per State

	Rate 1,000 population \2						
	1990	2000	2005	2006	2007	2008	2009
Maryland	9.7	5.8	6.9	6.6	6.5	6.0	5.8
Massachusetts	7.9	6.7	6.2	5.9	5.9	5.7	5.5
Michigan	8.2	6.8	6.0	5.9	5.9	5.5	5.4
Minnesota	7.7	6.9	5.9	6.0	6.0	5.5	5.3
Mississippi	9.4	7.8	5.8	5.7	5.7	5.2	4.8
Missouri	9.6	7.3	7.0	6.9	6.9	6.8	6.5
Montana	8.6	7.6	7.4	7.5	7.5	7.7	7.4
Nebraska	8.0	72.2	7.0	6.8	6.8	6.9	6.7
Nevada	99.0	9.4	57.8	52.6	52.6	43.1	40.9
New Hampshire	9.5	6.0	7.2	7.1	7.1	6.8	40.9
New Jersey	7.6	6.0	5.7	5.5	5.5	5.4	6.4
New Mexico \5	8.8	8.0	6.6	6.9	6.9	4.0	5.0
New York \5	8.6	7.1	6.8	6.8	6.8	6.5	5.1
North Carolina	7.8	8.2	7.3	7.3	7.3	7.0	6.4
North Dakota	7.5	7.2	6.9	6.8	6.8	6.7	6.7
Ohio	9.0	7.8	6.5	6.3	6.3	6.0	6.6
Oklahoma	10.6	(NA)	7.3	7.3	7.3	7.1	5.8
Oregon	8.9	7.6	7.3	7.2	7.2	6.9	6.9
Pennsylvania	7.1	6.0	5.8	5.7	5.7	5.6	5.3
Rhode Island	8.1	7.6	7.0	6.5	6.5	6.2	5.9

<http://www2.census.gov/library/publications/2011/compendia/statab/131ed/tables/12s0133.xls>

Bottom Line

- If you see something weird in your graph that you can't explain, go back and double-check your data
- Even if you didn't make an error, maybe someone else did

How to clean data

- Manually!
- Interactively with data wrangling tools
- Batch processing through scripting

Data Wrangling Tools

- **Open Refine (was Google refine)**
 - <http://openrefine.org>
 - video: http://www.youtube.com/watch?v=yNccGtn3Wb0&feature=player_embedded
- **Mr. People**
 - <http://people.erickson.net/>
 - formats lists of names
- **Mr. Data Converter**
 - http://shancarter.com/data_converter/
- **Data Wrangler**
 - <http://vis.stanford.edu/wrangler/>
 - video: <http://vimeo.com/19185801>
- **Data Science Toolkit**
 - <http://www.datasciencetoolkit.org/>
 - lots of quick conversion tools

Detect inconsistency in WDV survey data

- Look at the [data](#) carefully
 1. Identify all of the inconsistencies in the major column.
 2. Identify other problem you see in the data.
 3. What problems might these inconsistencies/incompleteness/noise cause when trying to analyze or visualize the data?
 4. If the complete dataset was only these 40 entries, how would you go about cleaning it? (Assuming that you had either a CSV or Excel file.) Would your strategy change if you had the full dataset of 2128 entries?
- Add your answer in this [document](#)
<https://goo.gl/ejsCac>

Data visualization in R

References

- Data Analysis and Visualization in R from Data Carpentry lessons

<http://www.datacarpentry.org/R-ecology-lesson/index.html>

- Producing Simple Graphs with R

<https://www.harding.edu/fmccown/r/>