Data visualization

Volodymyr Miz

for A Network Tour of Data Science (Fall 2019)
by Prof. Pierre Vandergheynst and Prof. Pascal Frossard

Some examples are taken from Dr. Kirell Benzi's class Data Visualization (2018)

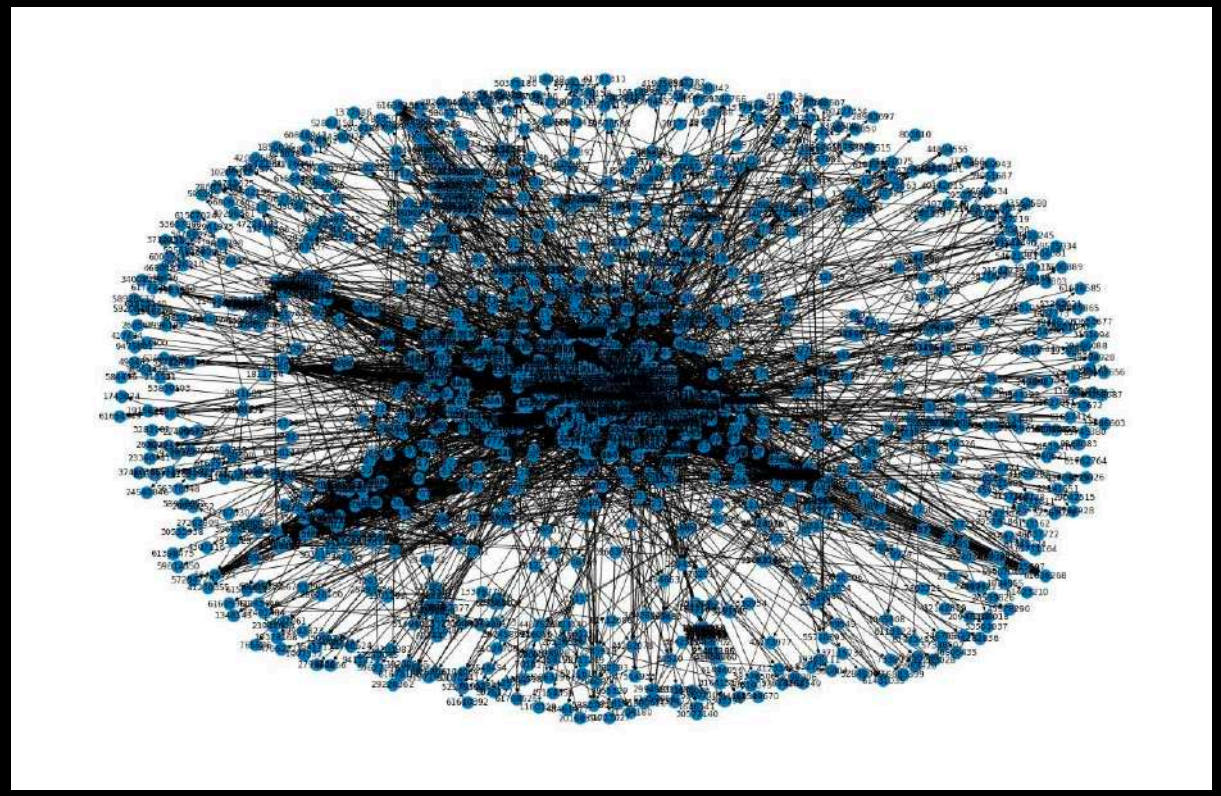
Outline

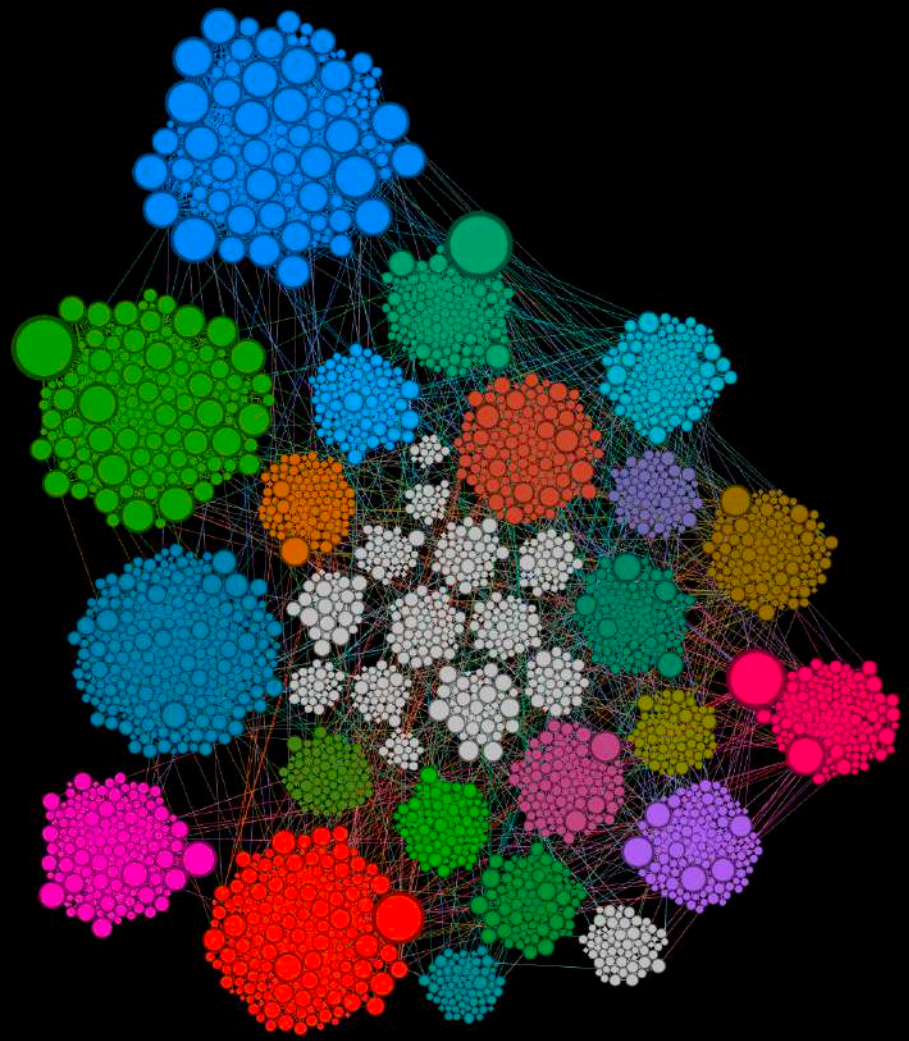
- Visualization and storytelling
  - Designing your visualization
- DOs and DON'Ts
  - Best practices
  - Misinformation
- Tools
- **Graph visualization**
- **Tutorials**
  - Graph layouts
  - Interactive web-based visualization

EPFL

LTS2

Motivation





EPFL

LTS2

1. Visualization and Storytelling

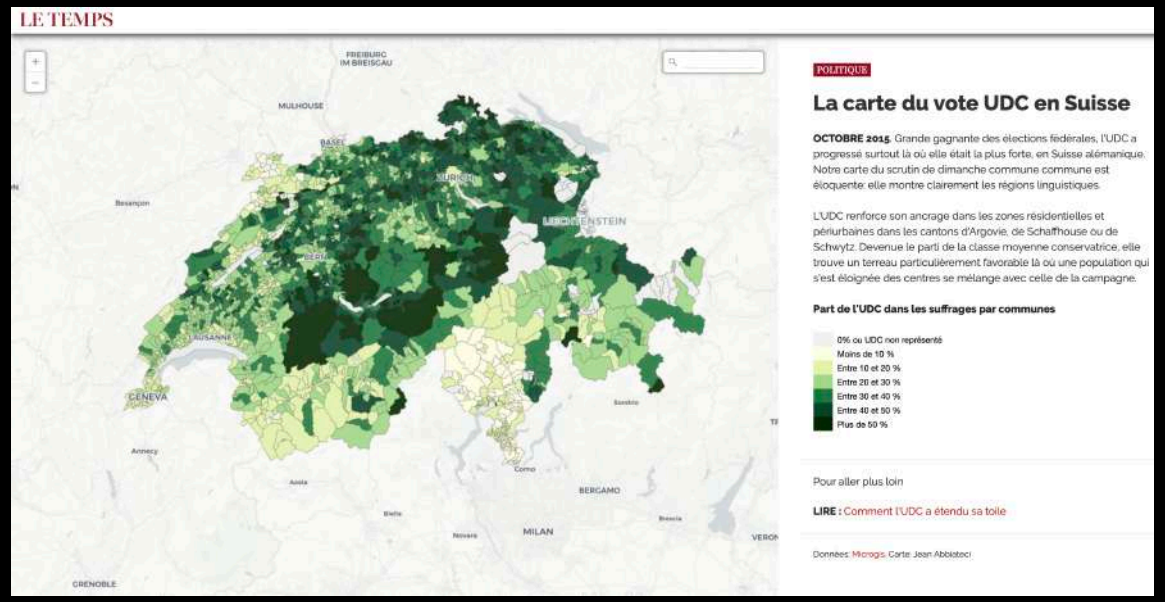
Visualization and Storytelling

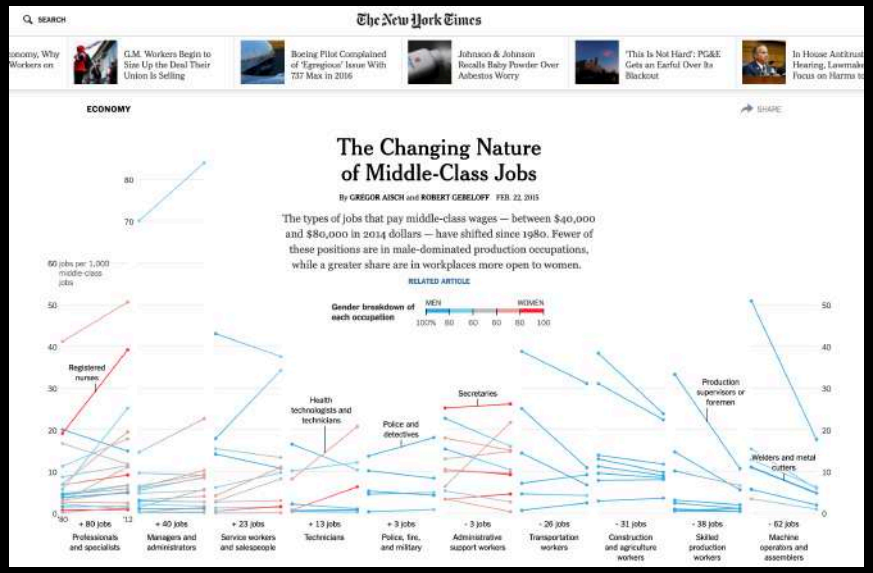
- Data journalism
- Designing your visualizations
  - What?
  - Why?
  - How?

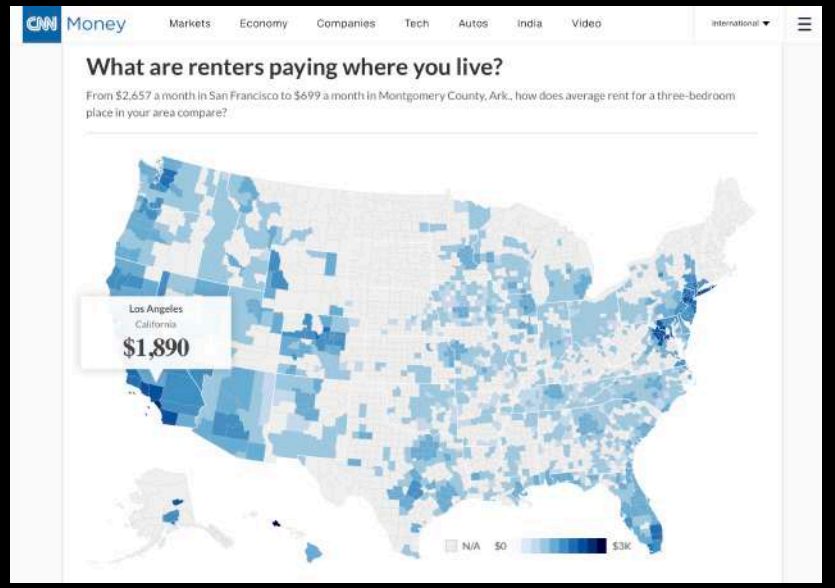
EPFL

LTS2

Data journalism. News



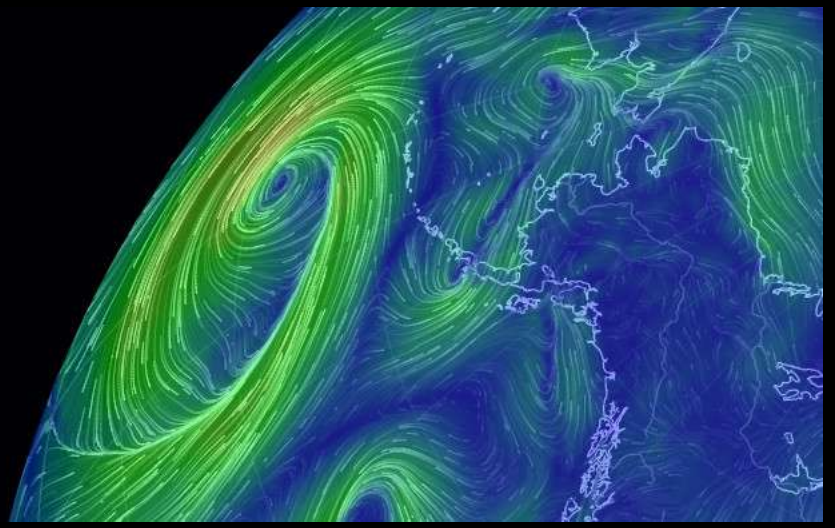


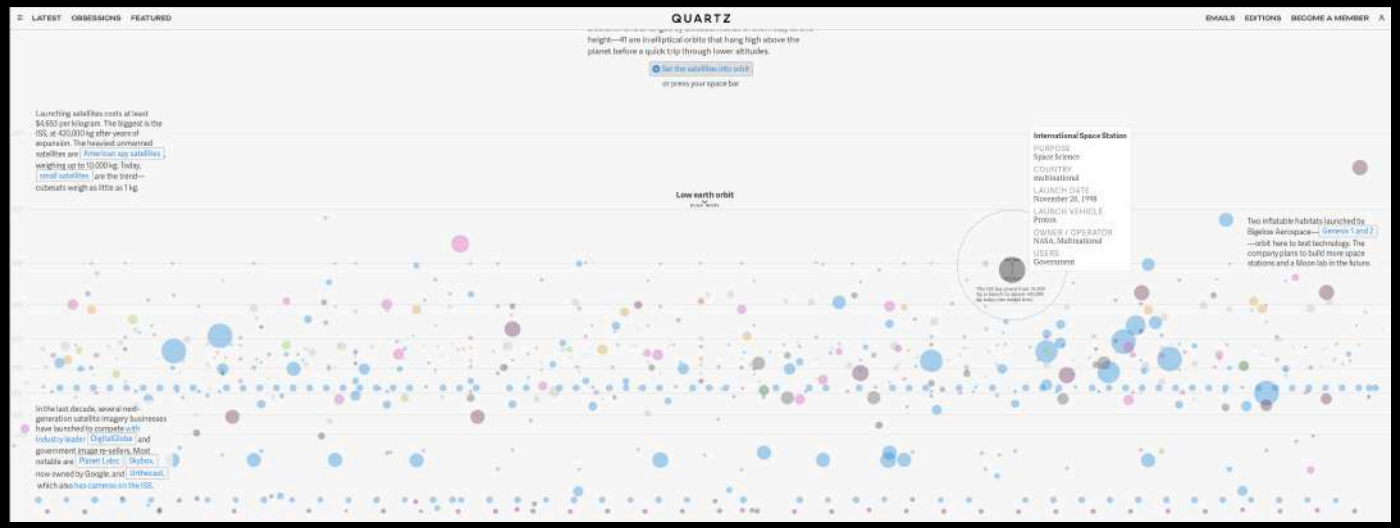


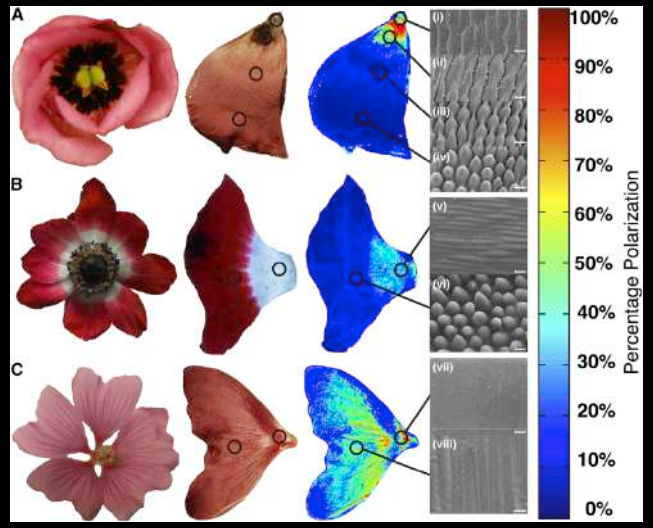
EPFL

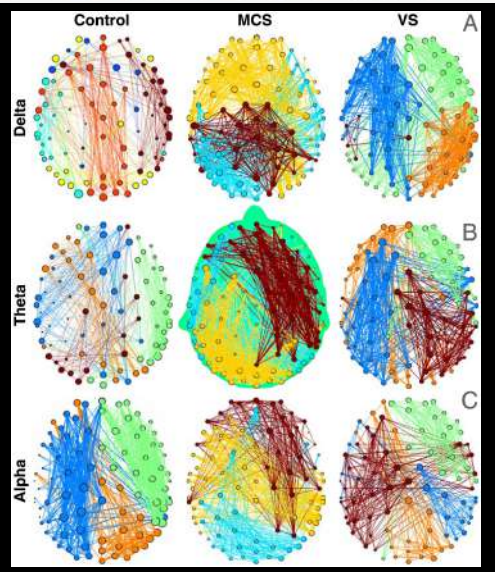
LTS2

Data journalism. Science Communication





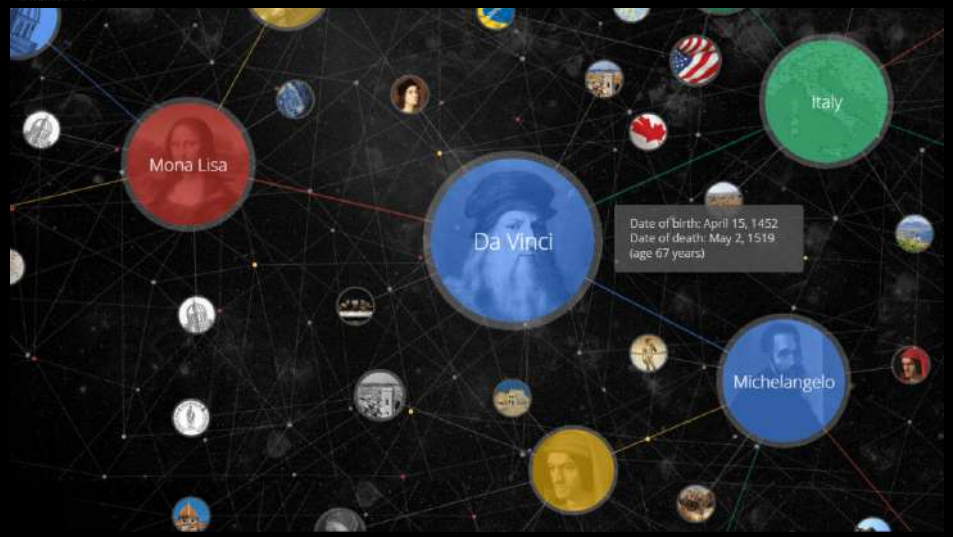


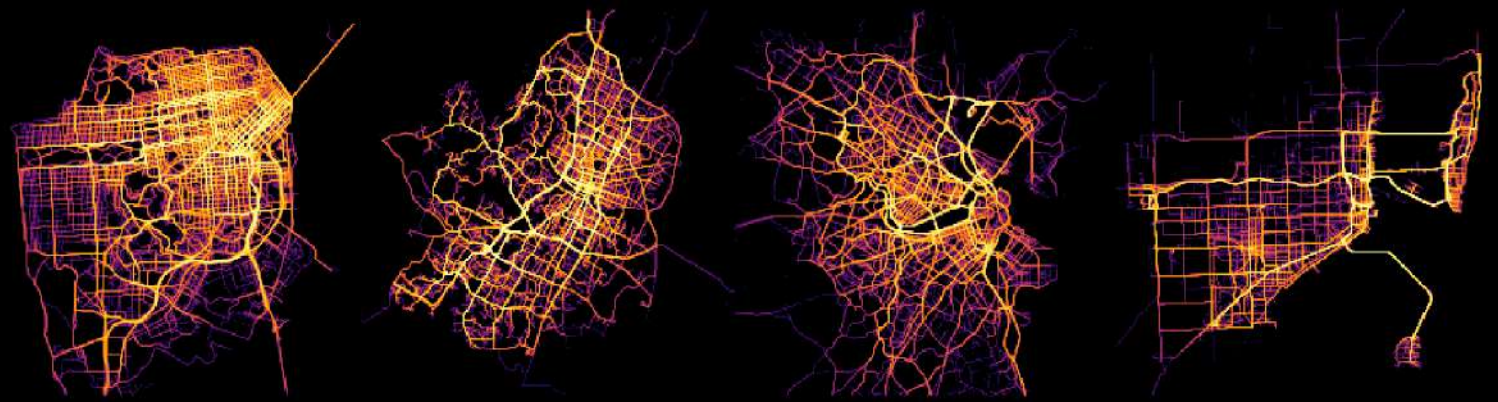


EPFL

LTS2

Data journalism. Tech and Marketing

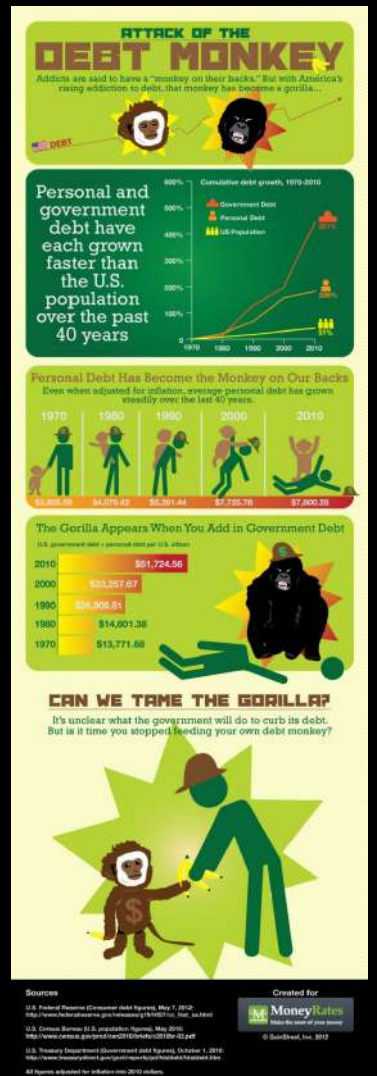


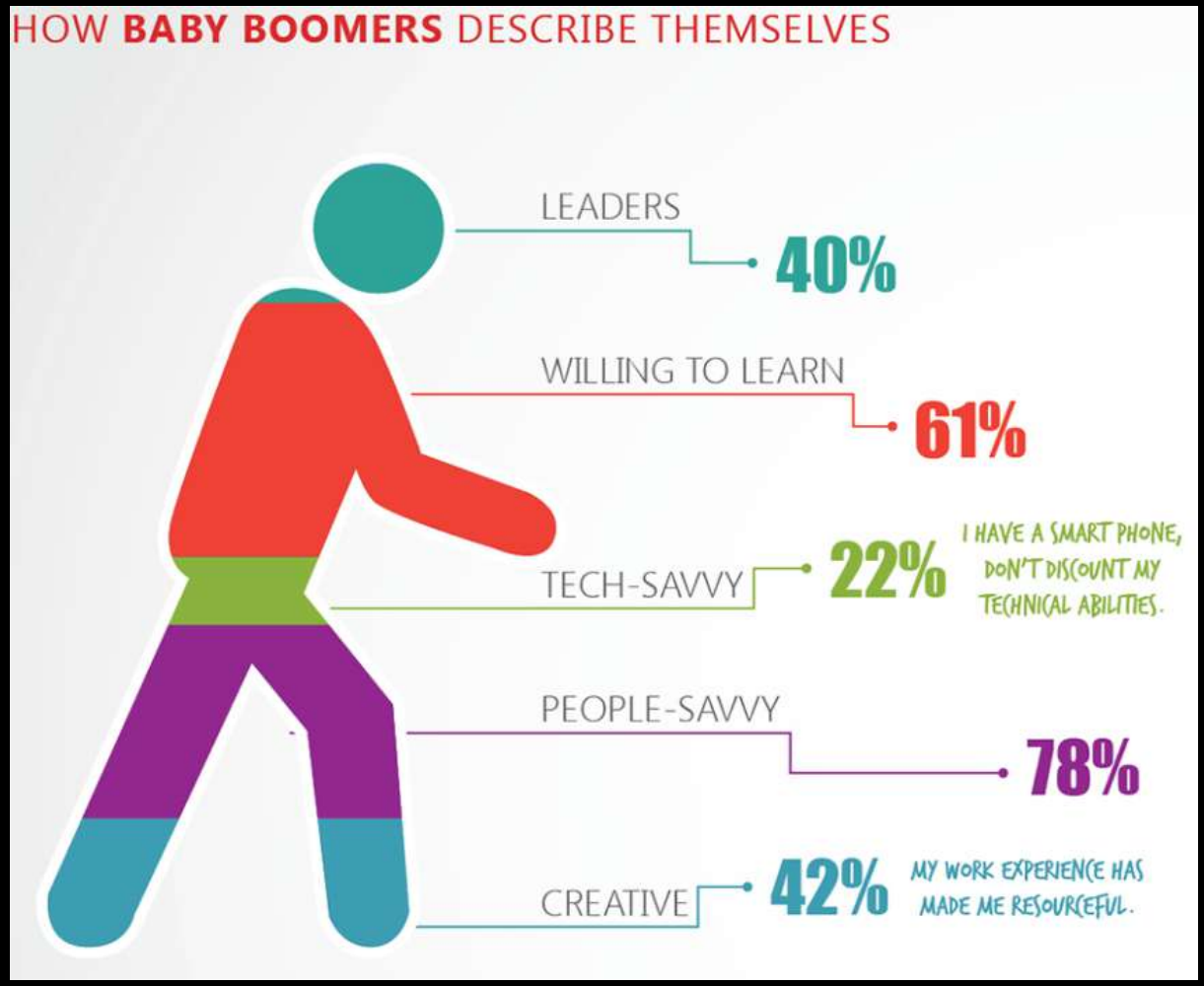


EPFL

LTS2

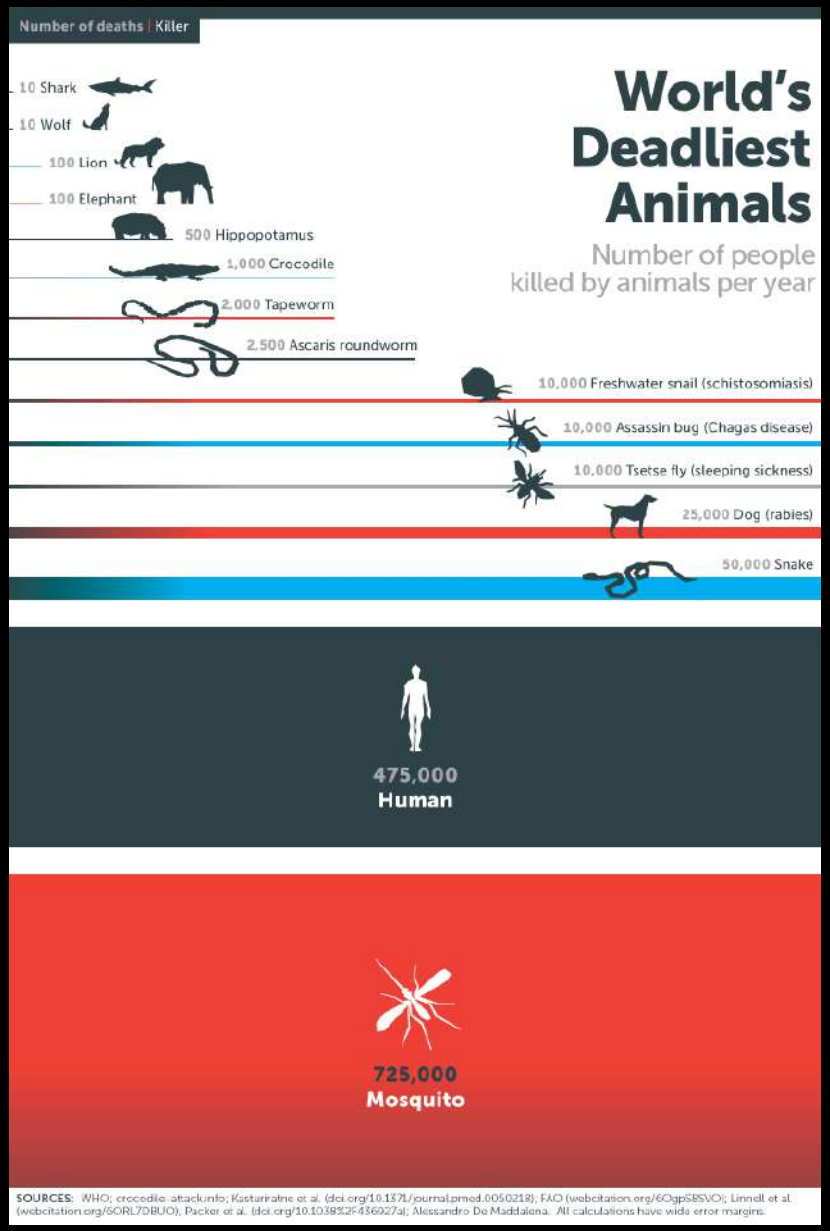
Bad Visualization Kills Good Content



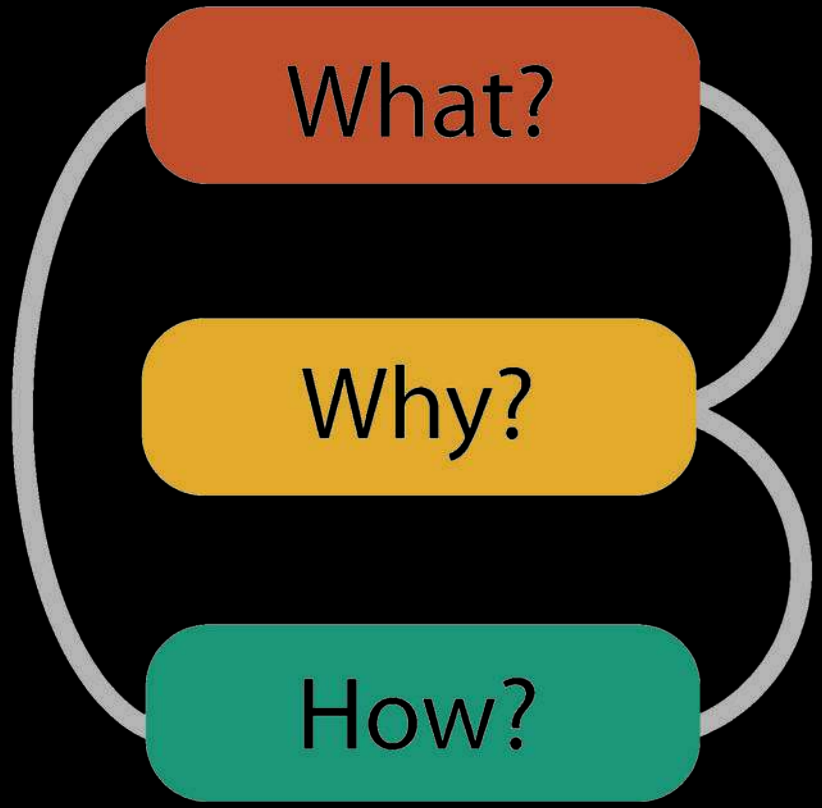


Source: https://www.copypress.com/blog/bad-data-visualization-is-killing-your-content/

Good Visualization Breaks Stereotypes



What? Why? How?



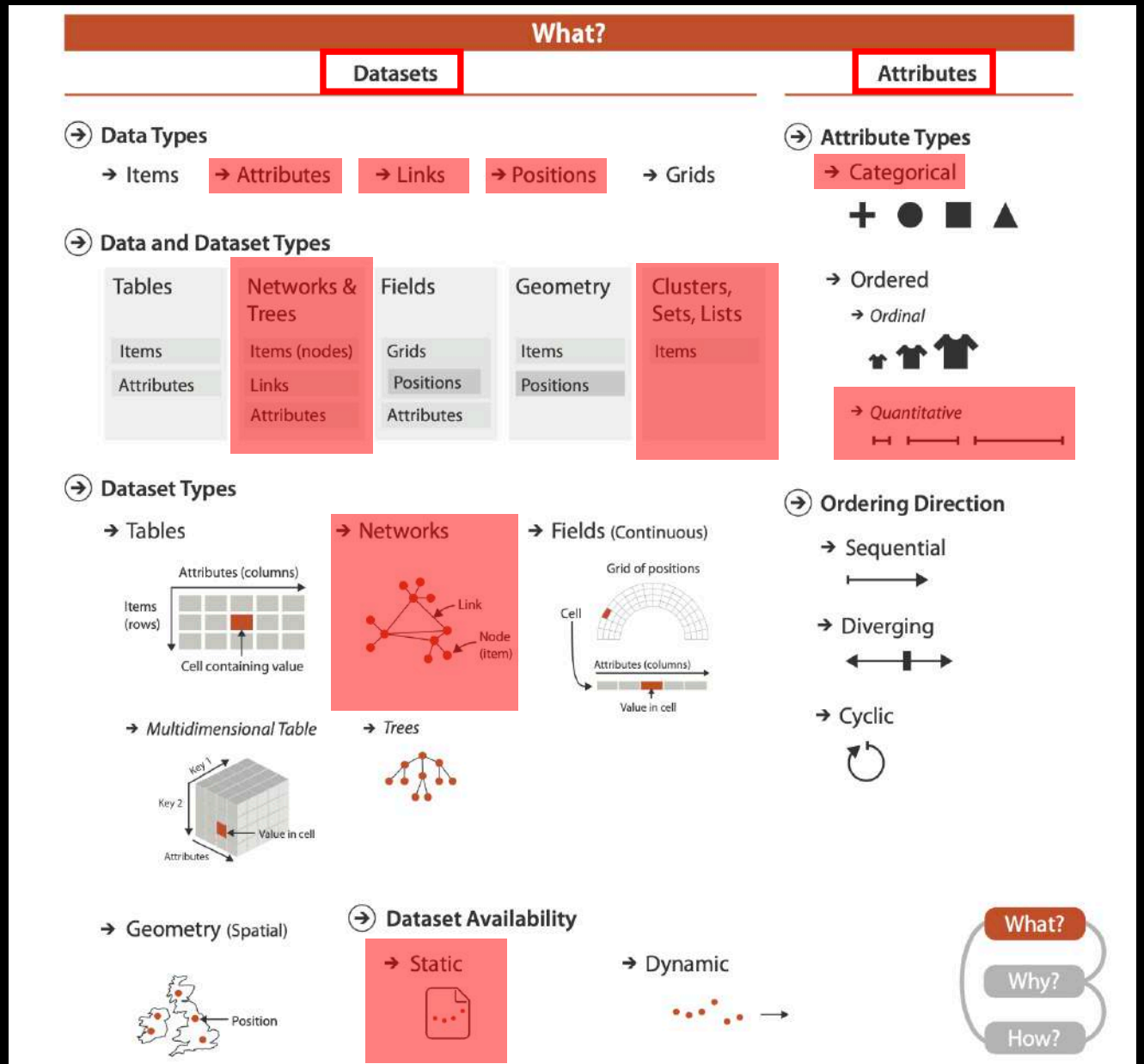
What data are you going to show?

Why are you showing visualization?

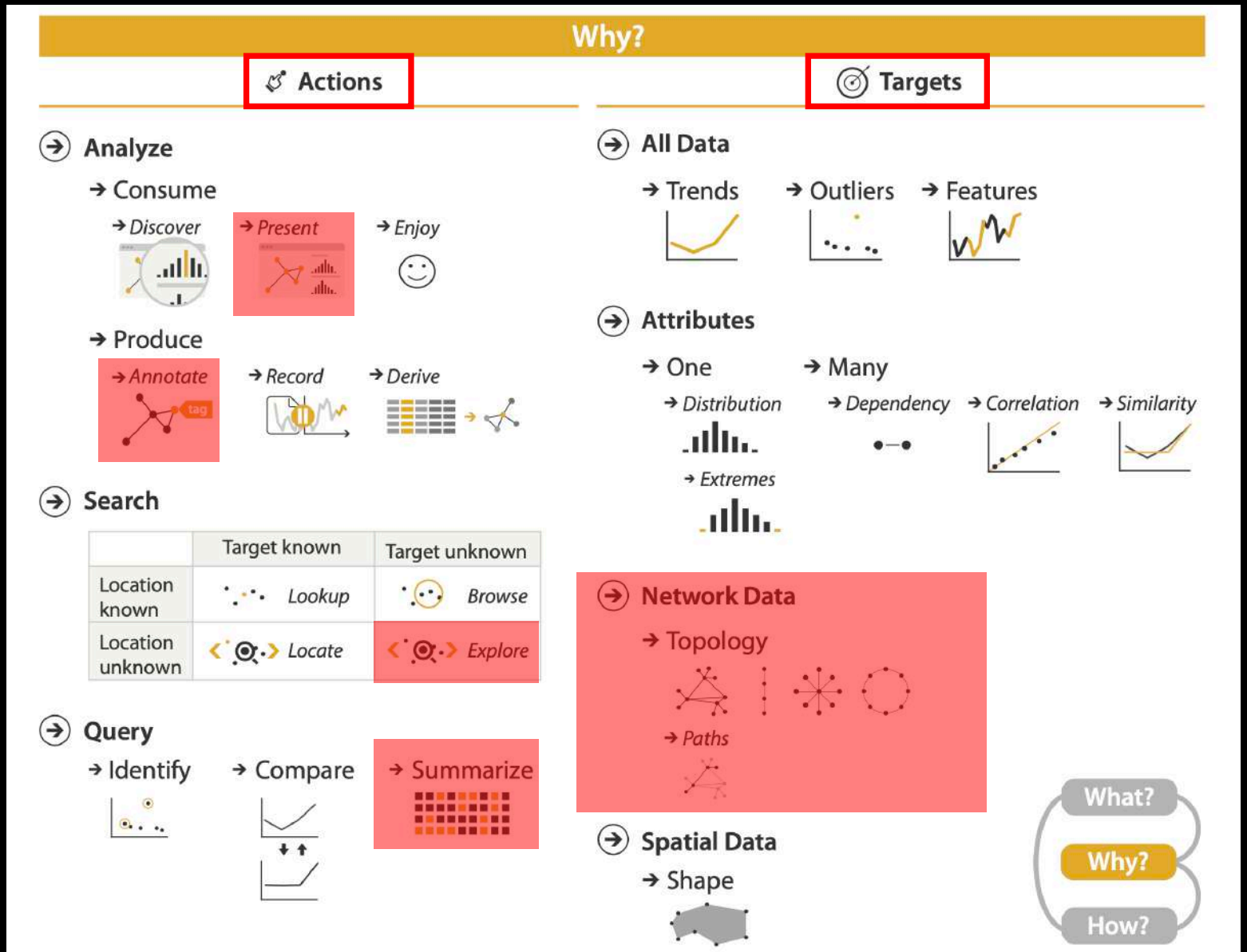
How are you going to construct visualization?

Source: Visualization. Analysis and Design by Tamara Munzner

What can we visualize?



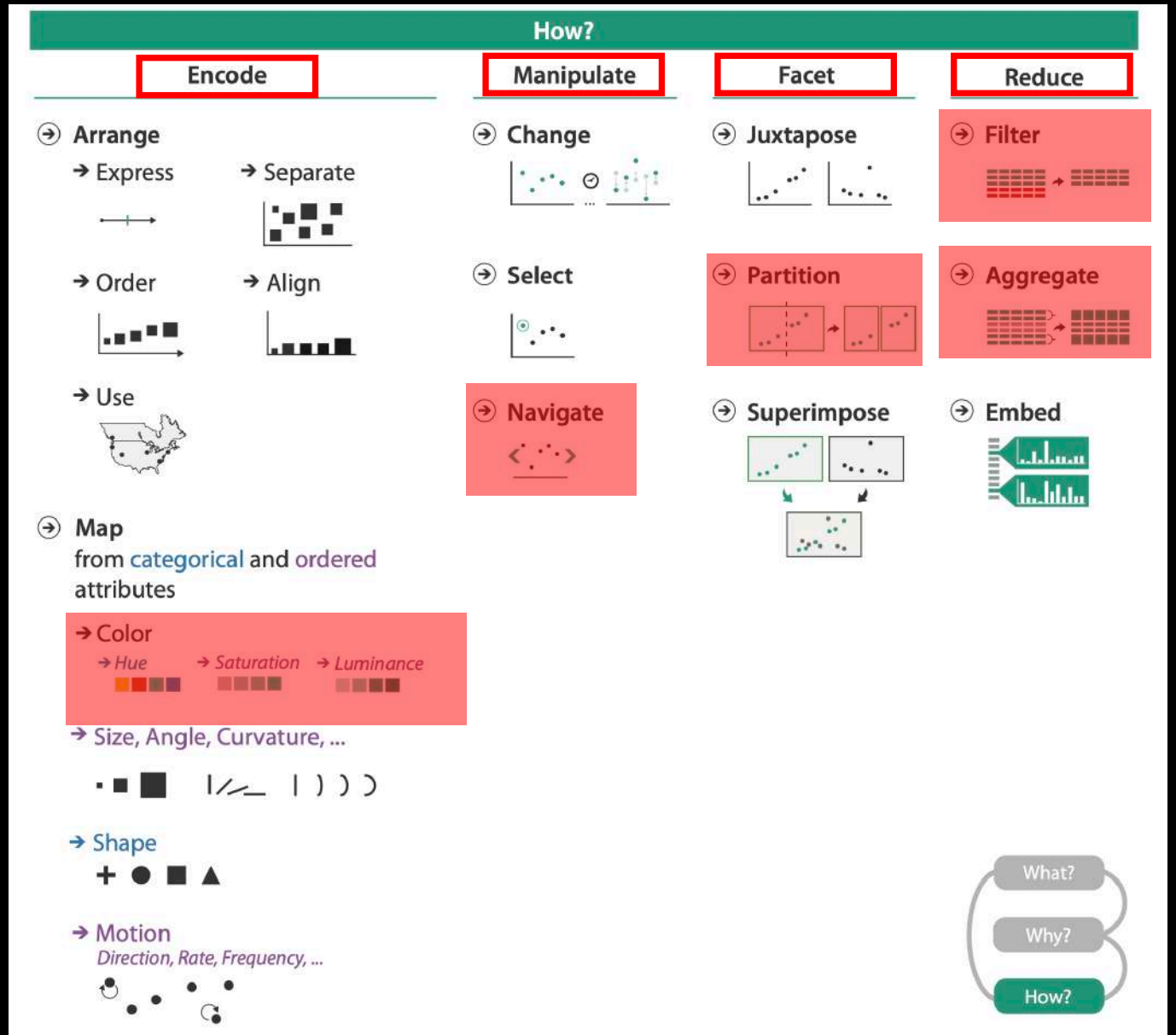
Why do people use our visualization?



EPFL

LTS2

How to design our visualization?



EPFL

LTS2

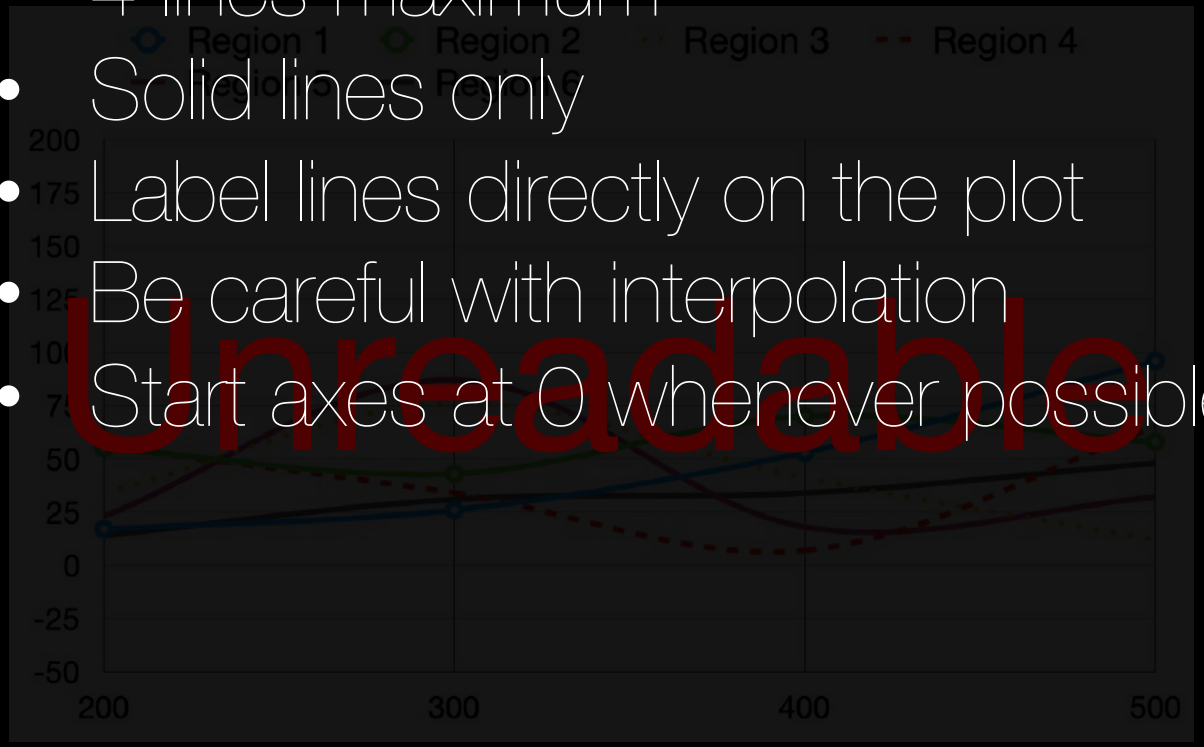
2. DOs and DON’Ts

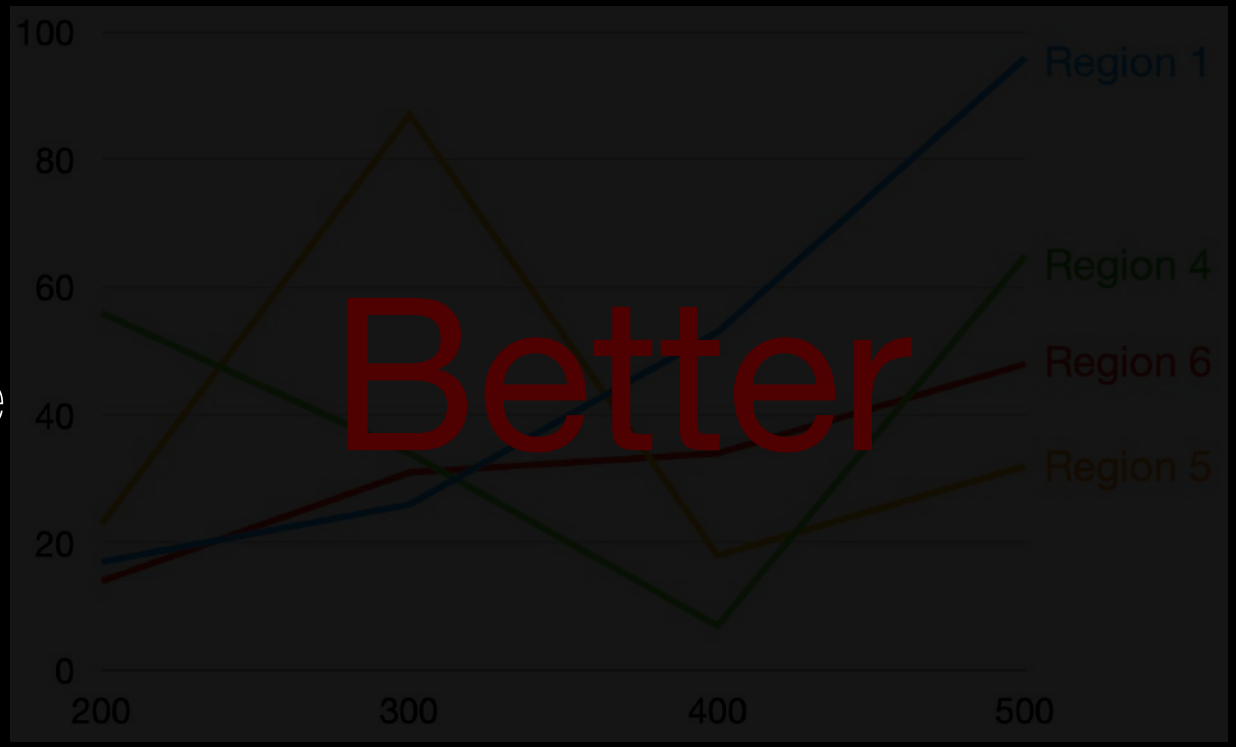
DOs and DON'Ts

- Best practices. Tips and Tricks
- Misinformation
  - Manipulation
  - Misleading visualization
  - False insights

Line Charts

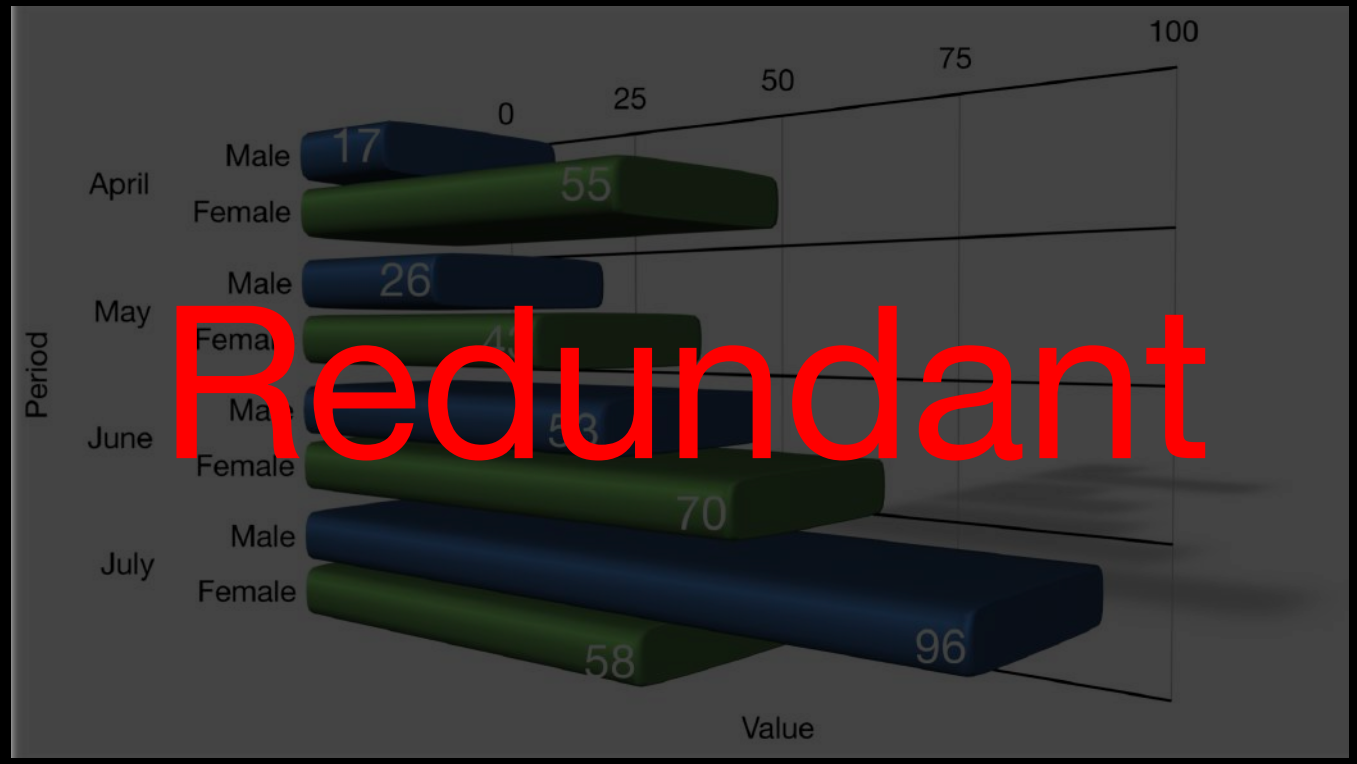
- 4 lines maximum
- Solid lines only
- Label lines directly on the plot
- Be careful with interpolation
- Start axes at 0 whenever possible





Source: Data Visualization class 2019 by Kirell Benzi

3D or not 3D. Bar charts



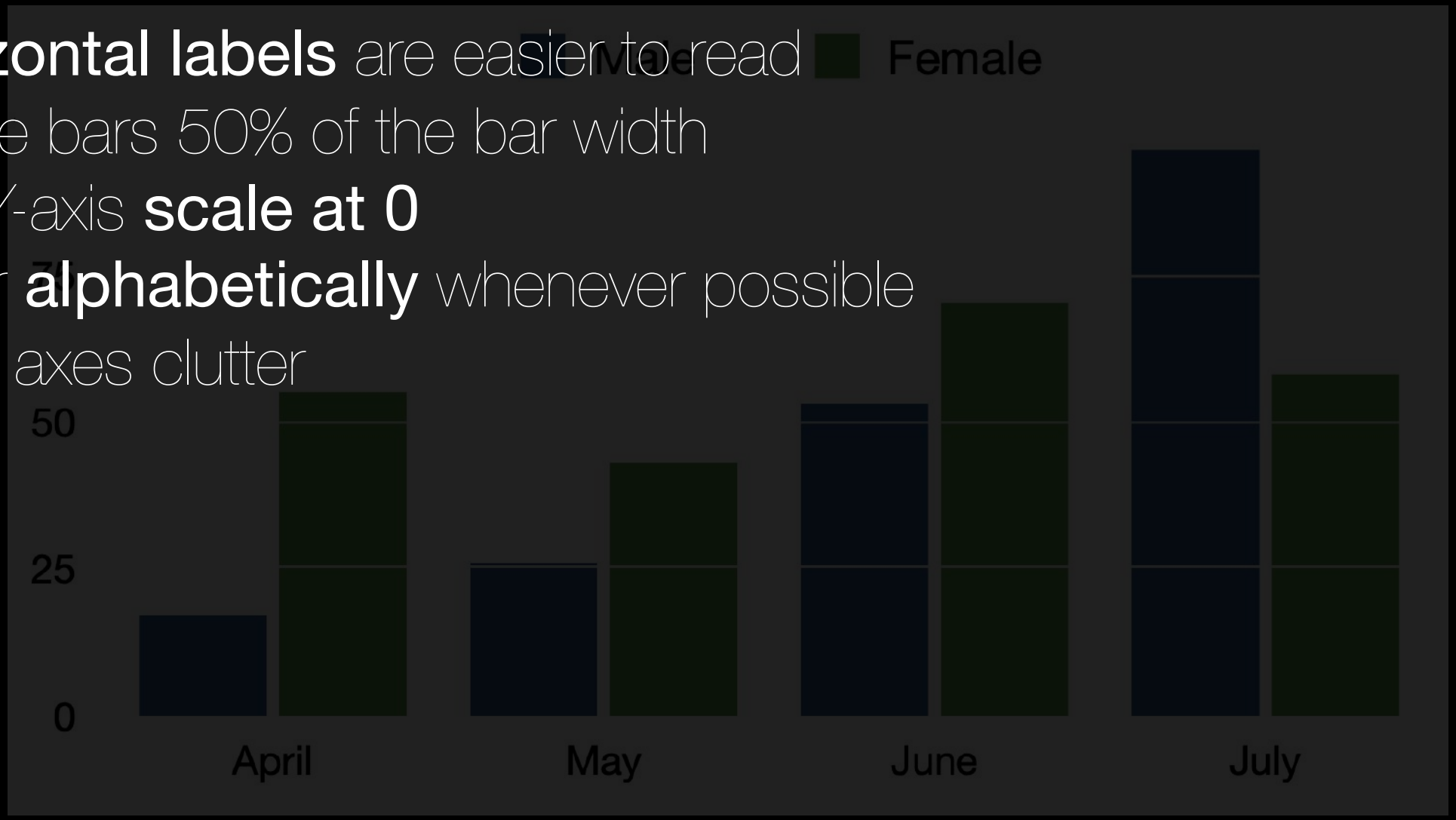
Redundant

Better

Source: Data Visualization class 2019 by Kirell Benzi

Bar Charts

- **Horizontal labels** are easier to read
- Space bars 50% of the bar width
- Star Y-axis **scale at 0**
- Order **alphabetically** whenever possible
- Avoid axes clutter



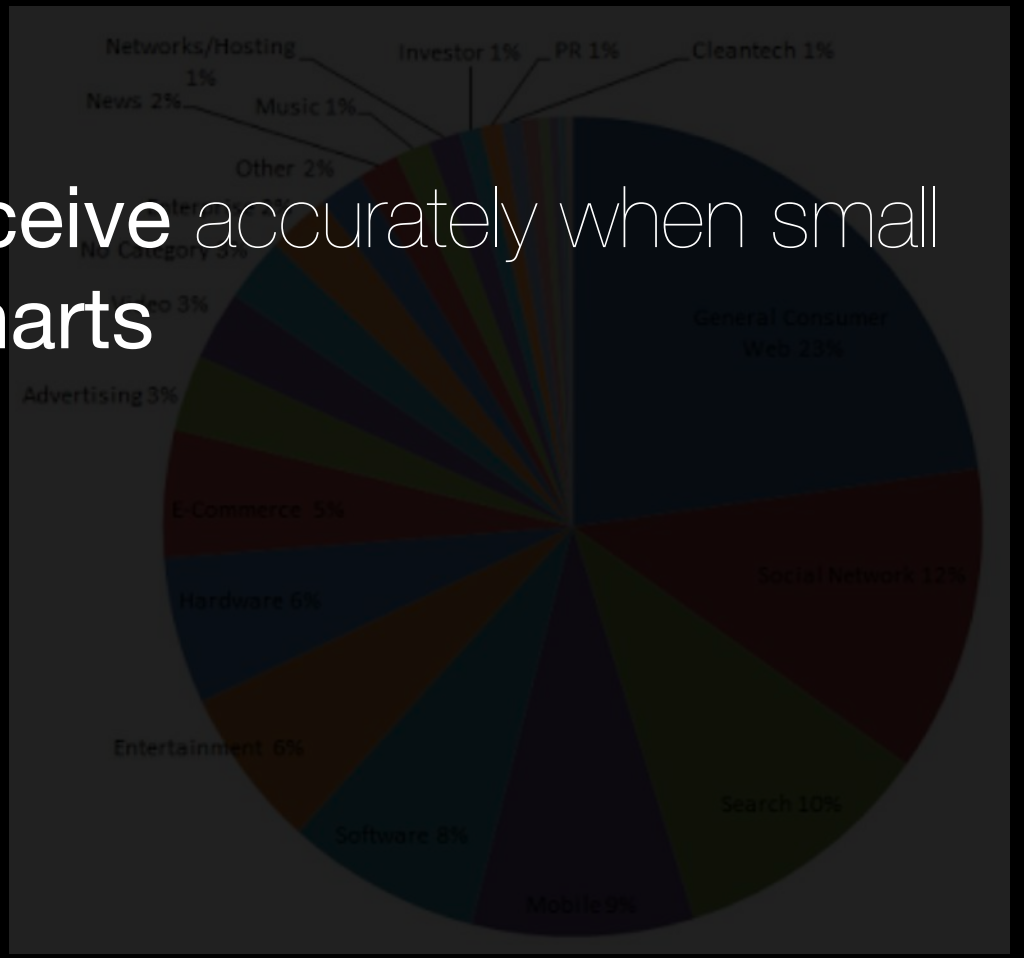
EPFL

Source: Data Visualization class 2019 by Kirell Benzi

LTS2

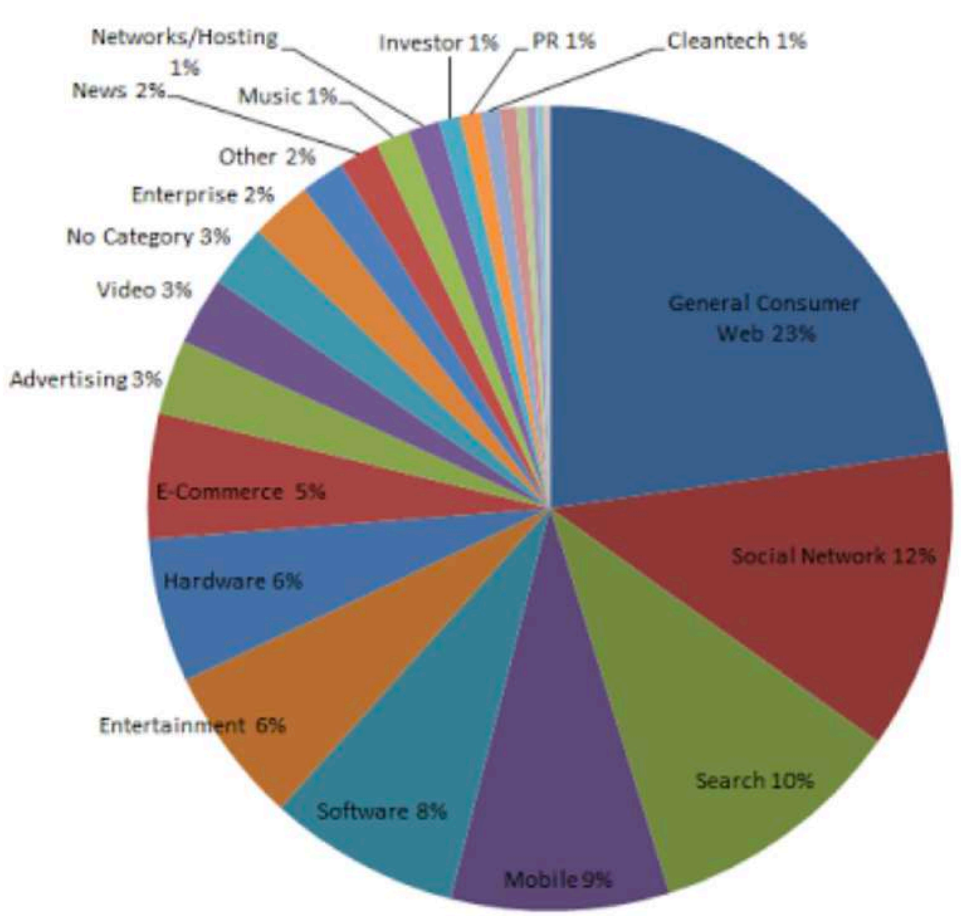
Pie Charts

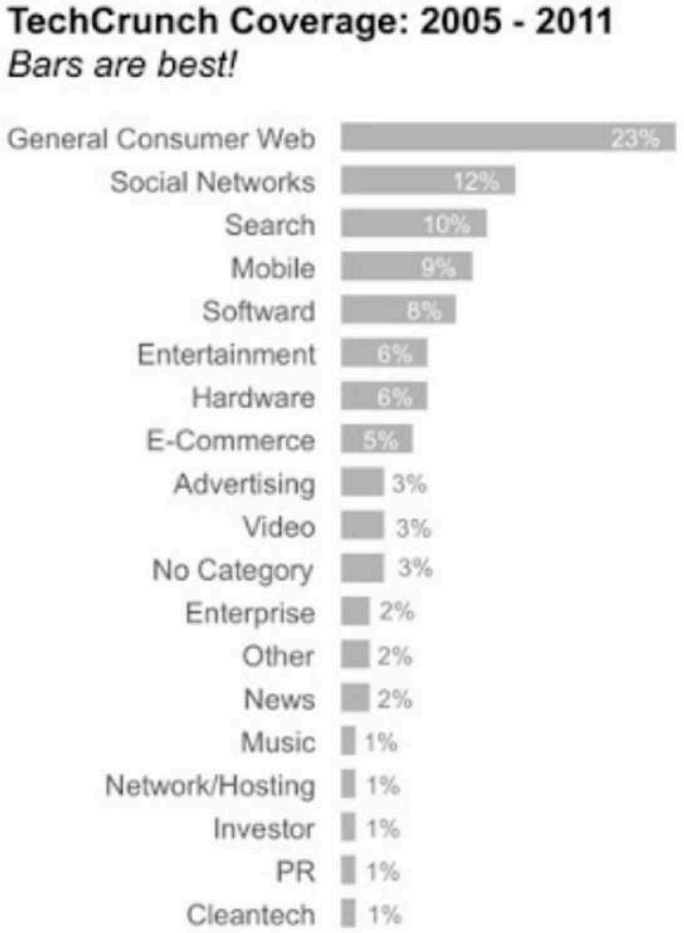
- Never ever use pie charts
- Pie charts are **misleading**
- Angles and areas are **difficult to perceive** accurately when small
- Always **convert pie charts to bar charts**



Source: Data Visualization class 2019 by Kirell Benzi

Convert pie to bar charts





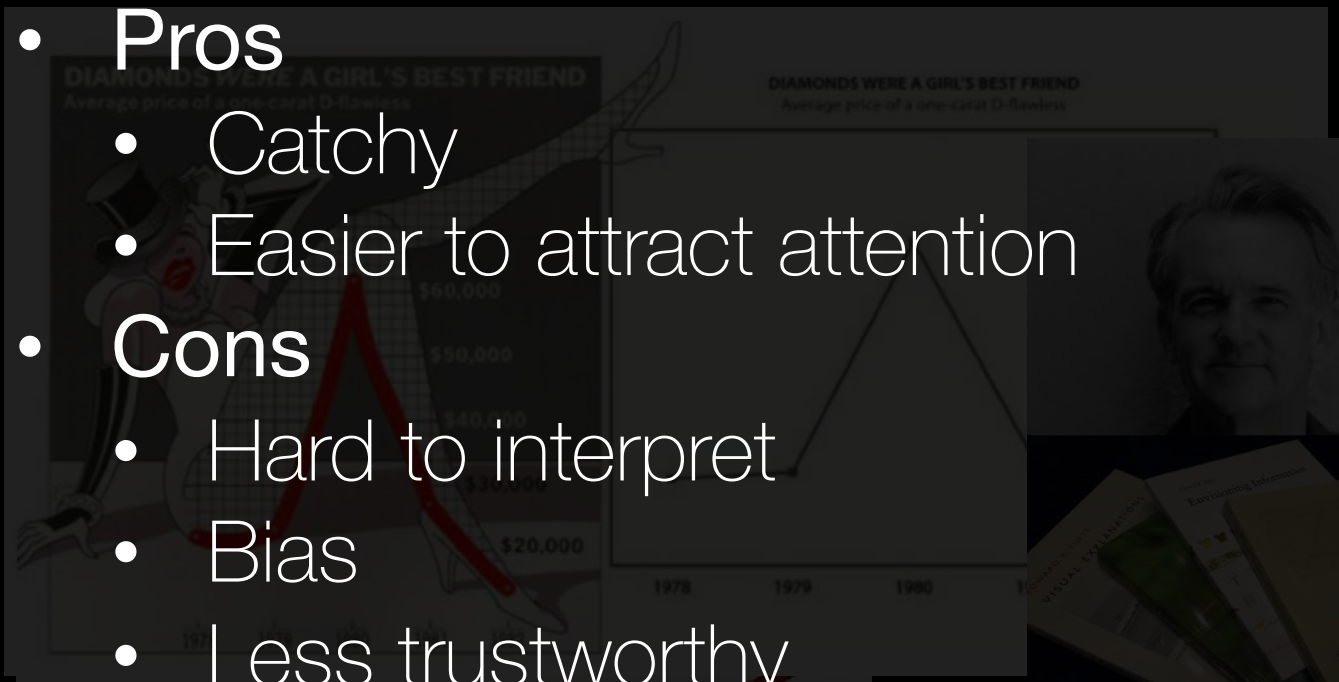
EPFL

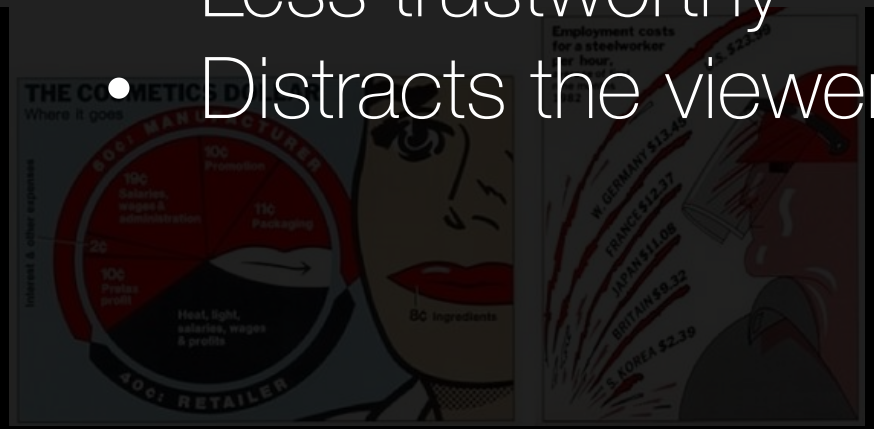
Source: Data Visualization class 2019 by Kirell Benzi

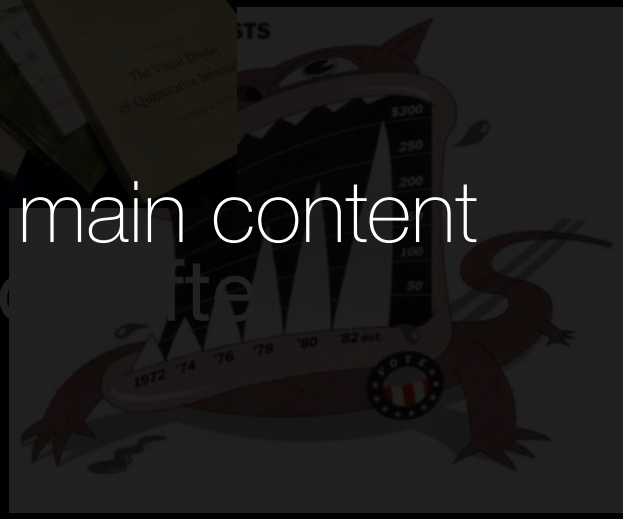
LTS2

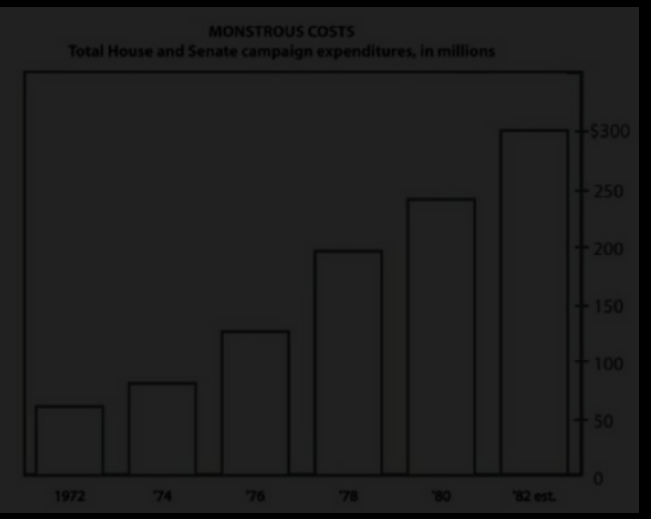
Chartjunk

- **Pros**
  - Catchy
  - Easier to attract attention
- **Cons**
  - Hard to interpret
  - Bias
  - Less trustworthy
  - Distracts the viewer from the main content



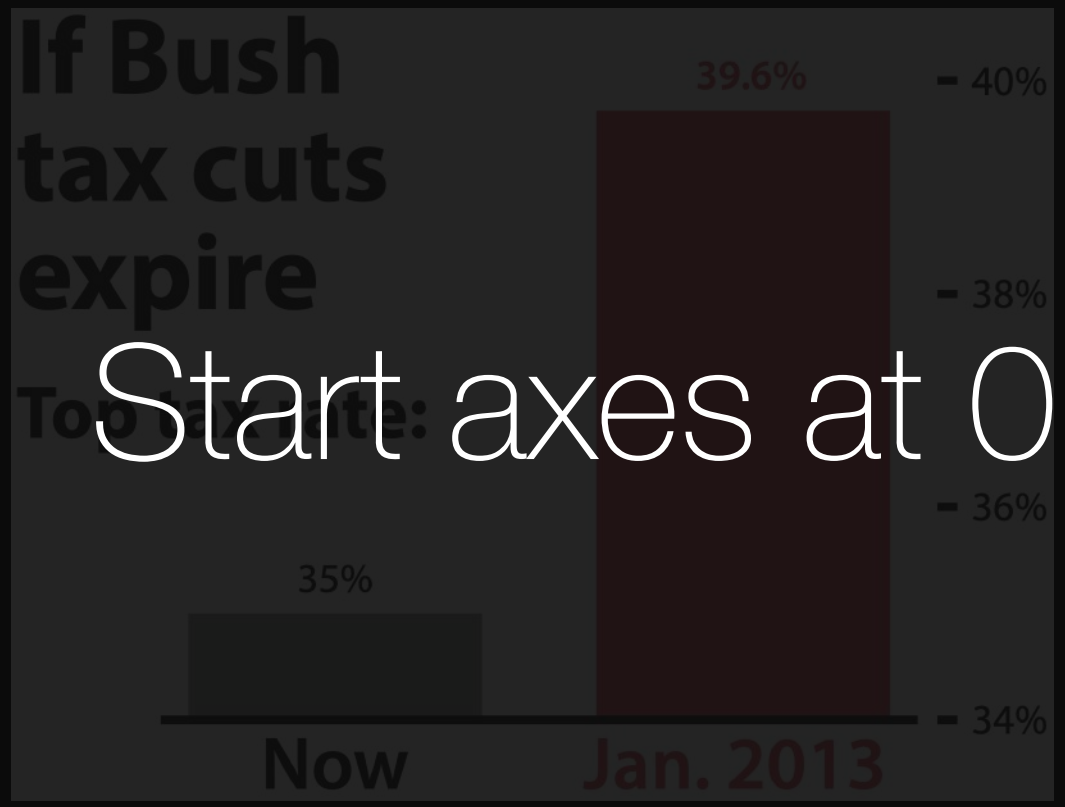


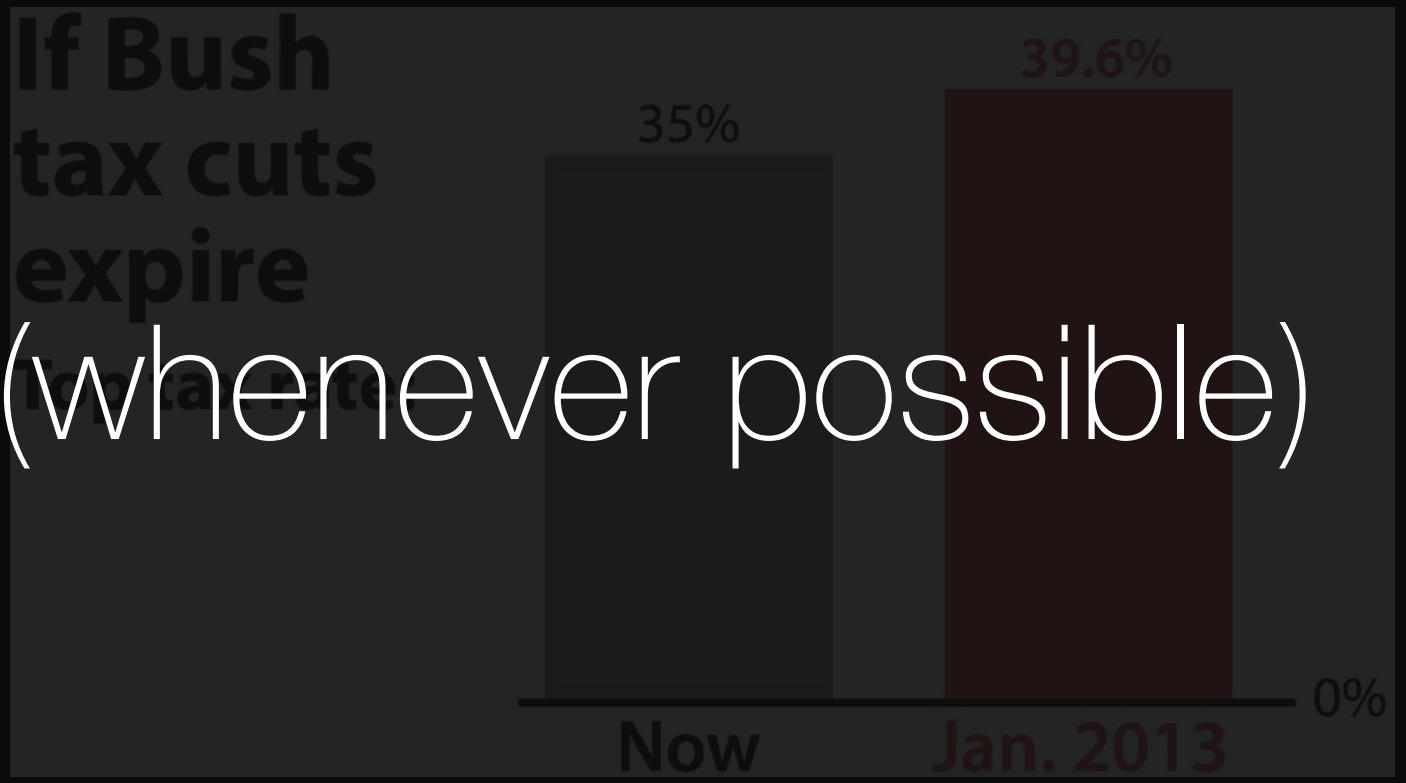




Source: Data Visualization class 2019 by Kirell Benzi

Misleading scale

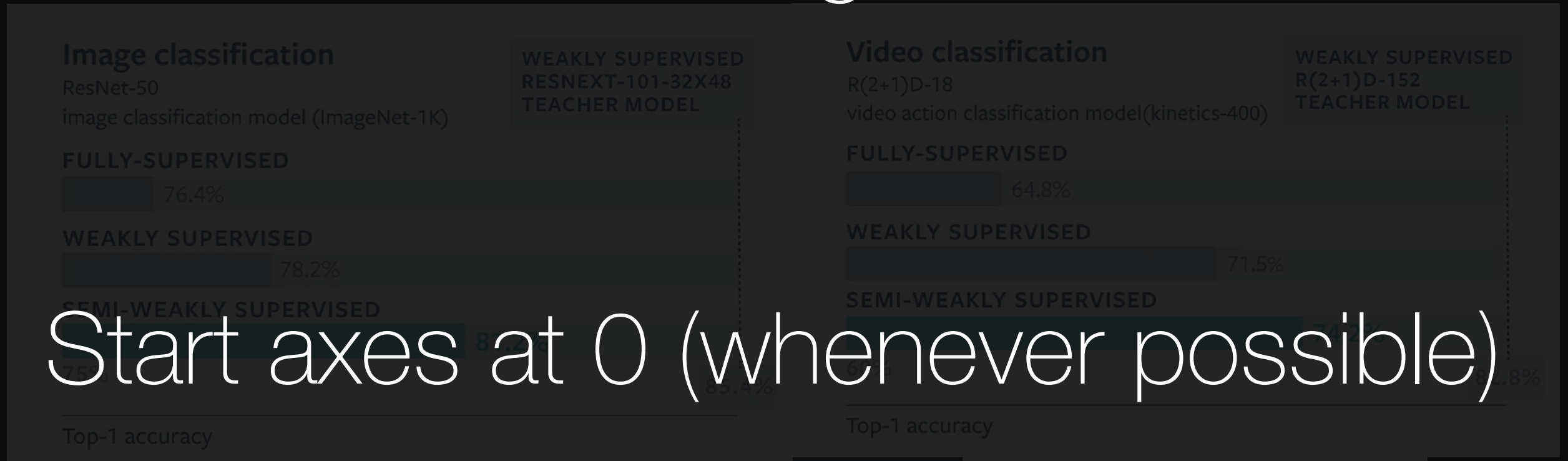




Start axes at 0 (whenever possible)

Source: How Charts Lie by Alberto Cairo

Misleading scale



Start axes at 0 (whenever possible)

Accurate image and video classification is important for a wide range of computer vision applications, from identifying harmful content, to making products more accessible to the visually impaired, to helping people more easily buy and sell things on products like Marketplace. Facebook AI is developing alternative ways to train our AI systems so that we can do more with less labeled training data overall, and also deliver accurate results even when large, high-quality labeled data sets are simply not available. Today, we are sharing details on a versatile new model training technique that delivers state-of-the-art accuracy for image and video classification systems.

This approach, which we call semi-weak supervision, is a new way to combine the merits of two different training methods: semi-supervised learning and weakly supervised learning. It opens the door the door to creating more accurate, efficient production classification models by using a teacher-student model training paradigm and billion-scale weakly supervised data sets. If the weakly supervised data sets (such as the hashtags associated with publicly available photos) are not available for the target classification task, our method can also make use of unlabeled data sets to produce highly accurate semi-supervised models.

Related posts

Creating a data set and a challenge for deepfakes

September 05, 2019

Mapping roads through deep learning and weakly supervised training

July 23, 2019

RoBERTa: An optimized method for

EPFL

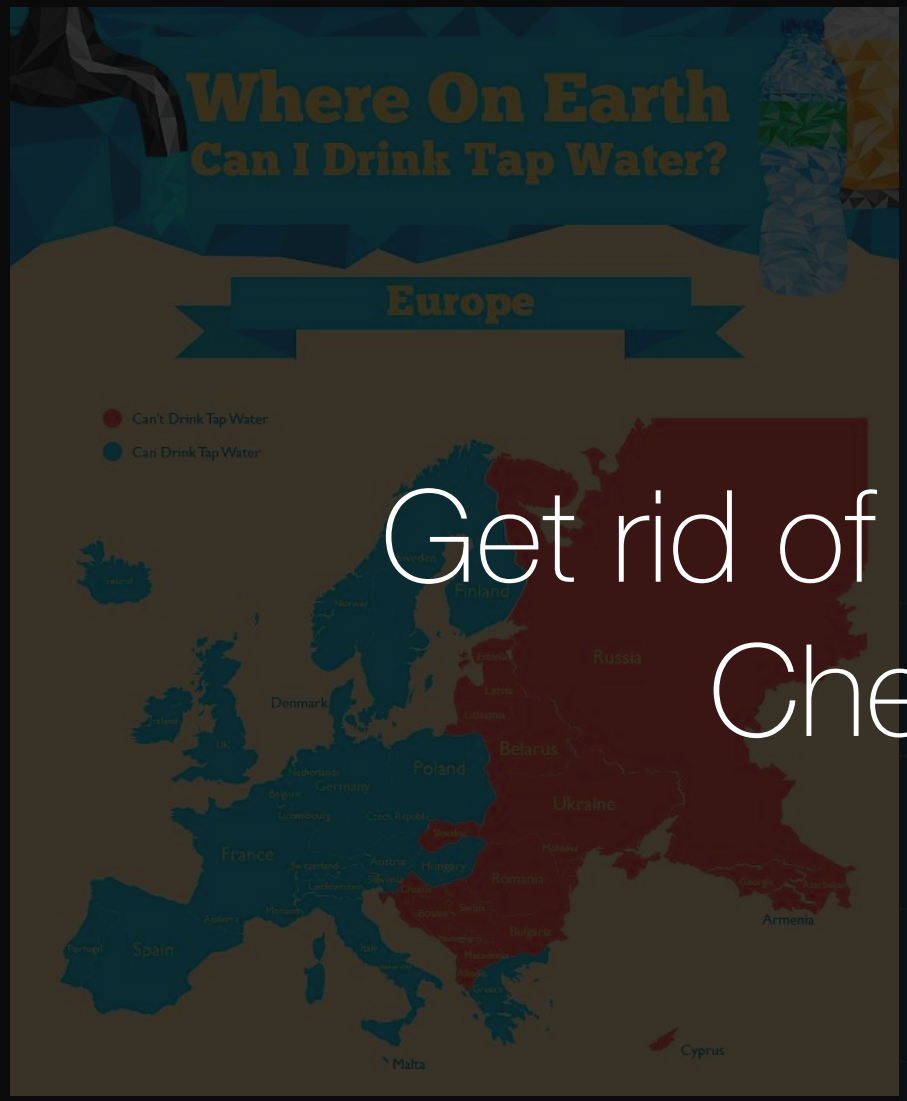
LTS2

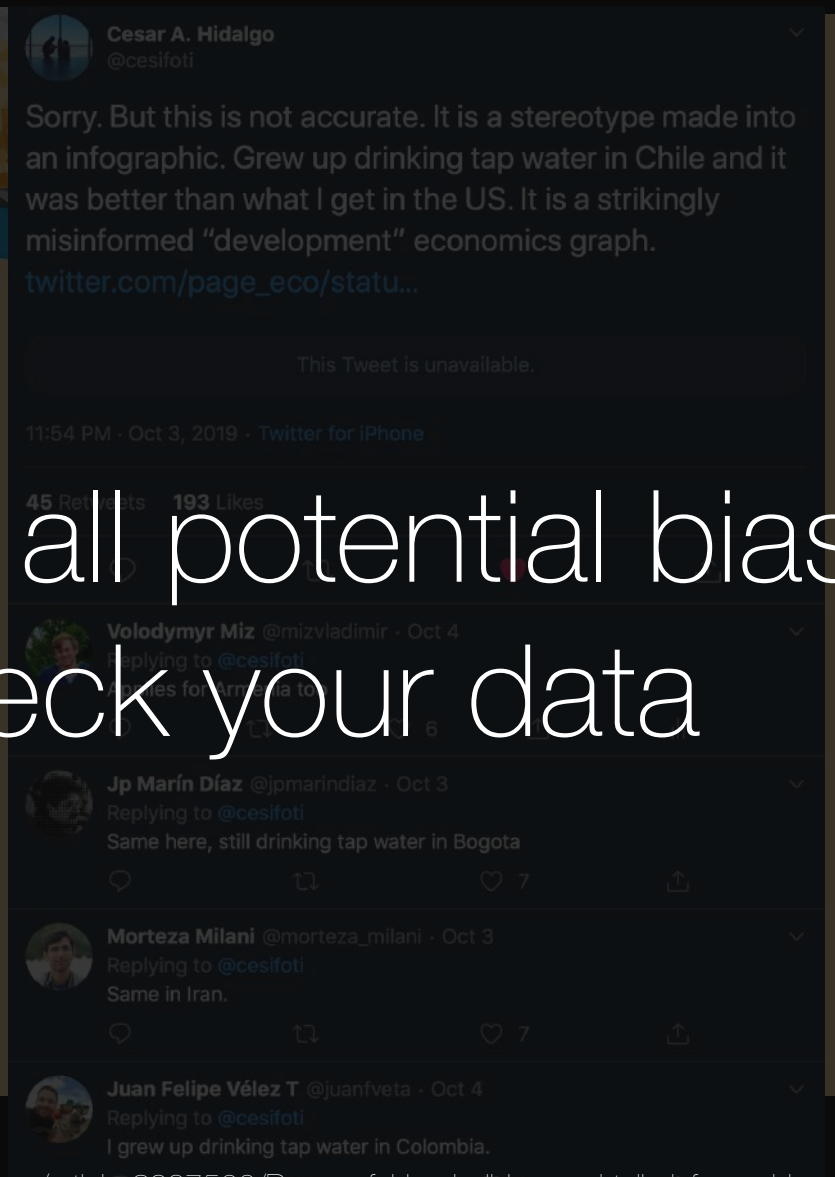
Misleading attributes

Attributes can be misleading.
Explore all attributes and know unique aspects of the data

Source: How Charts Lie by Alberto Cairo

Misinformation. Stereotypes





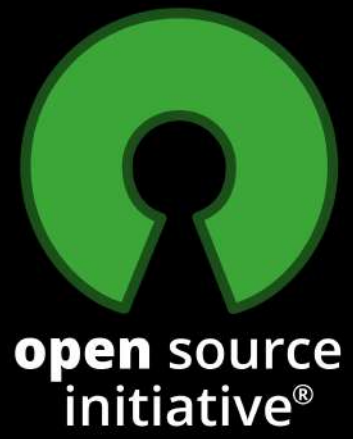
Get rid of all potential biases.
Check your data

Source: https://www.dailymail.co.uk/travel/travel_news/article-3227566/Be-careful-ice-holiday-cocktails-Infographic-reveals-world-safe-drink-tap-water.html

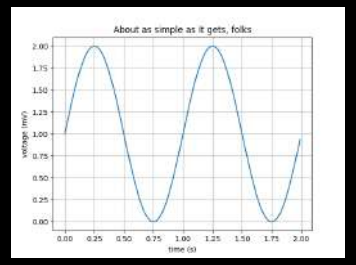
3. Tools

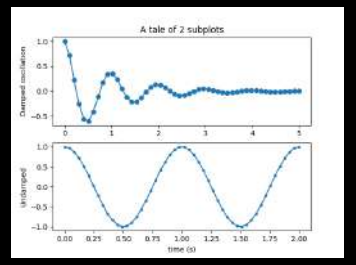
Tools

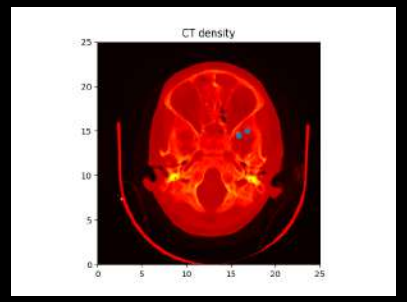
- **Open Source** and commercial products
- Python. **Scientific visualization**
  - Matplotlib
  - Bokeh
  - Datashader
- JavaScript. Publishing **interactive visualization online**
  - D3JS
- **No code** tools
  - RAWgraphs

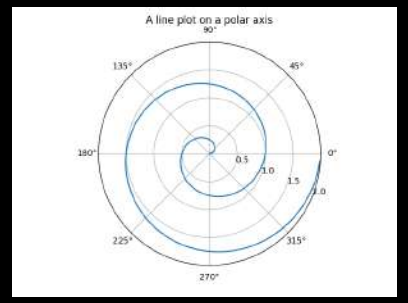


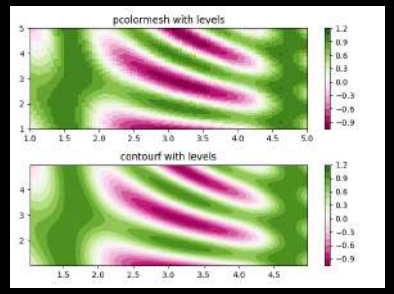
Matplotlib

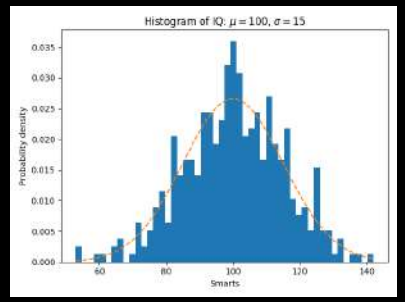


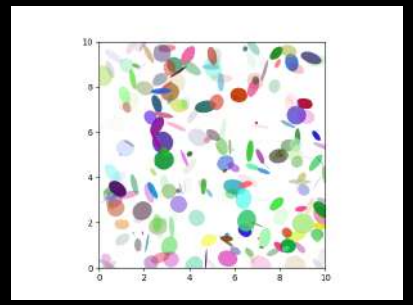


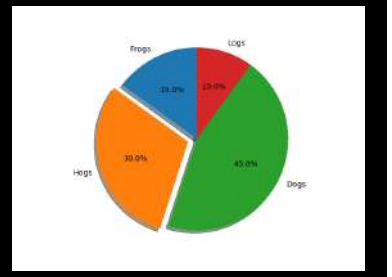


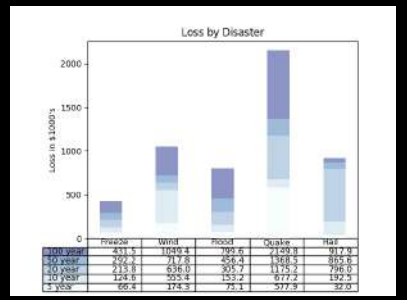


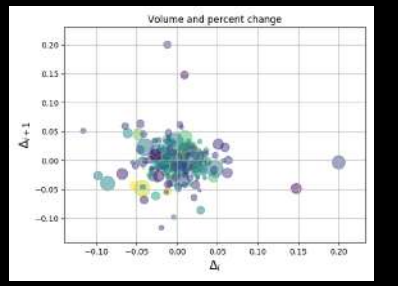


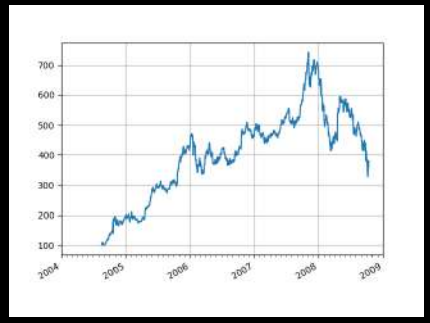


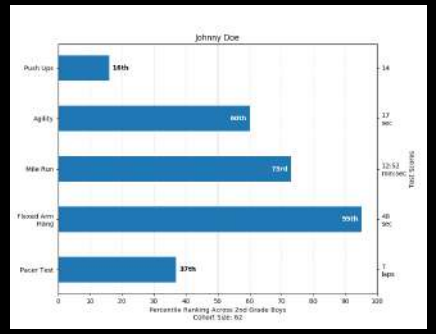


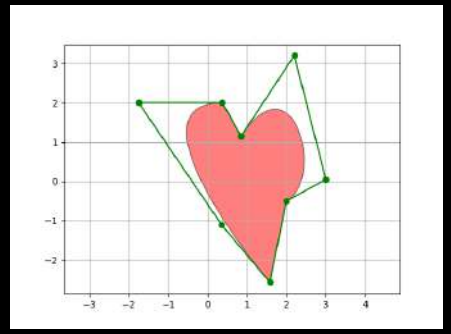


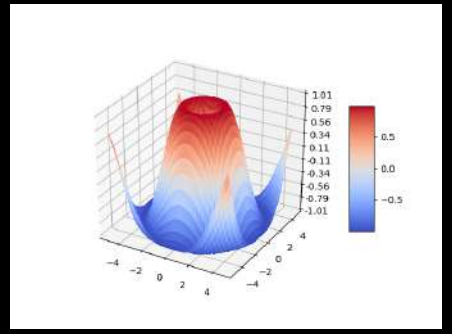


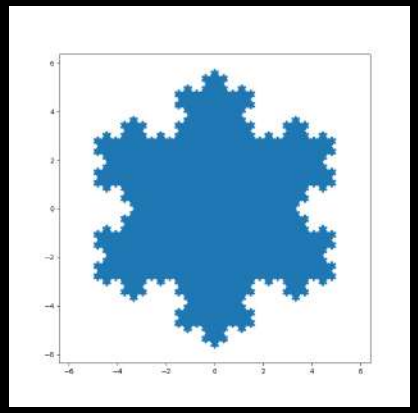


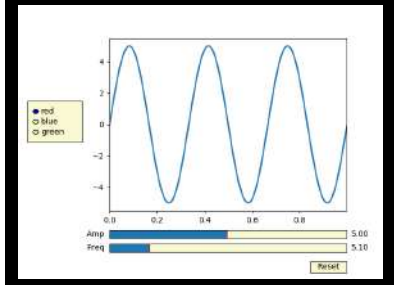


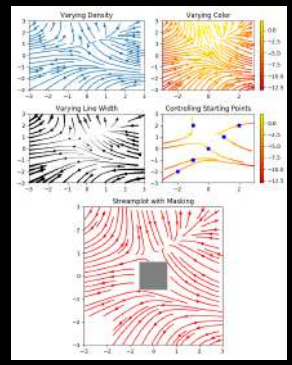






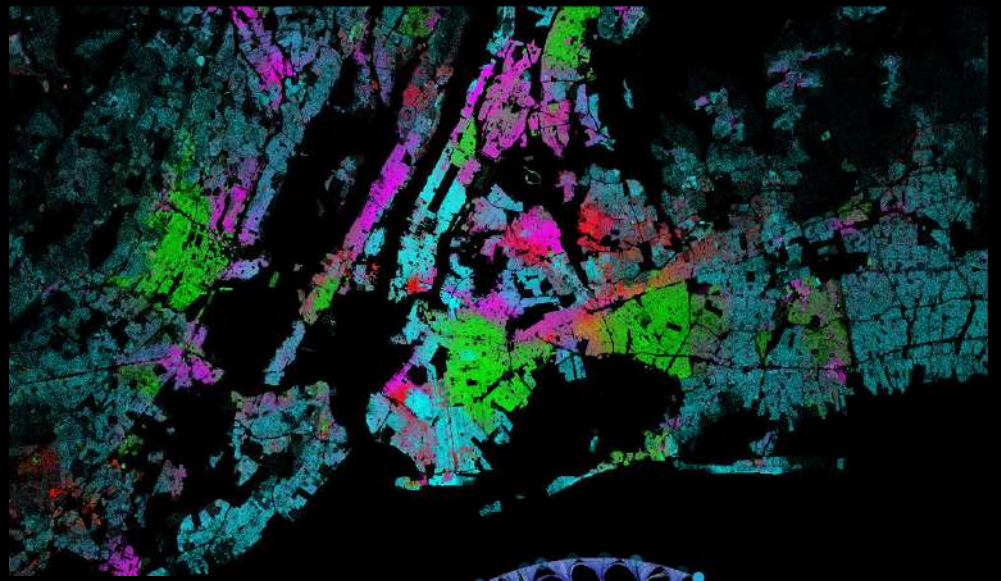


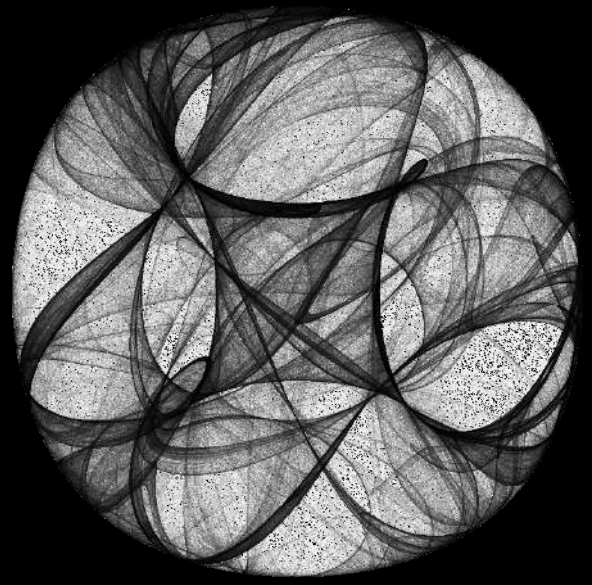


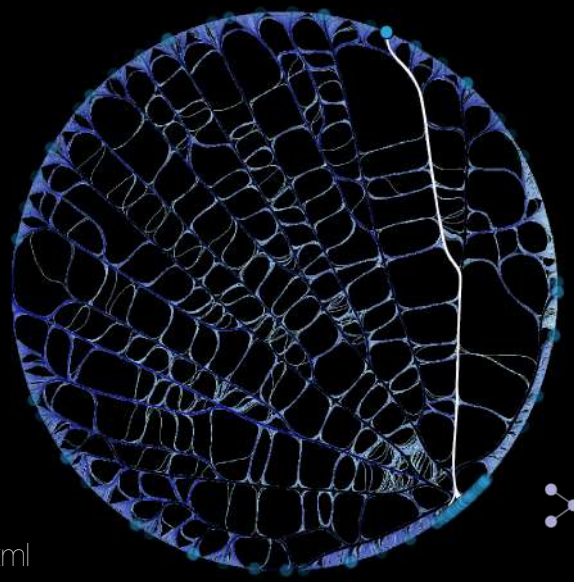


Source: https://matplotlib.org/3.1.1/tutorials/introductory/sample_plots.html

Datashader

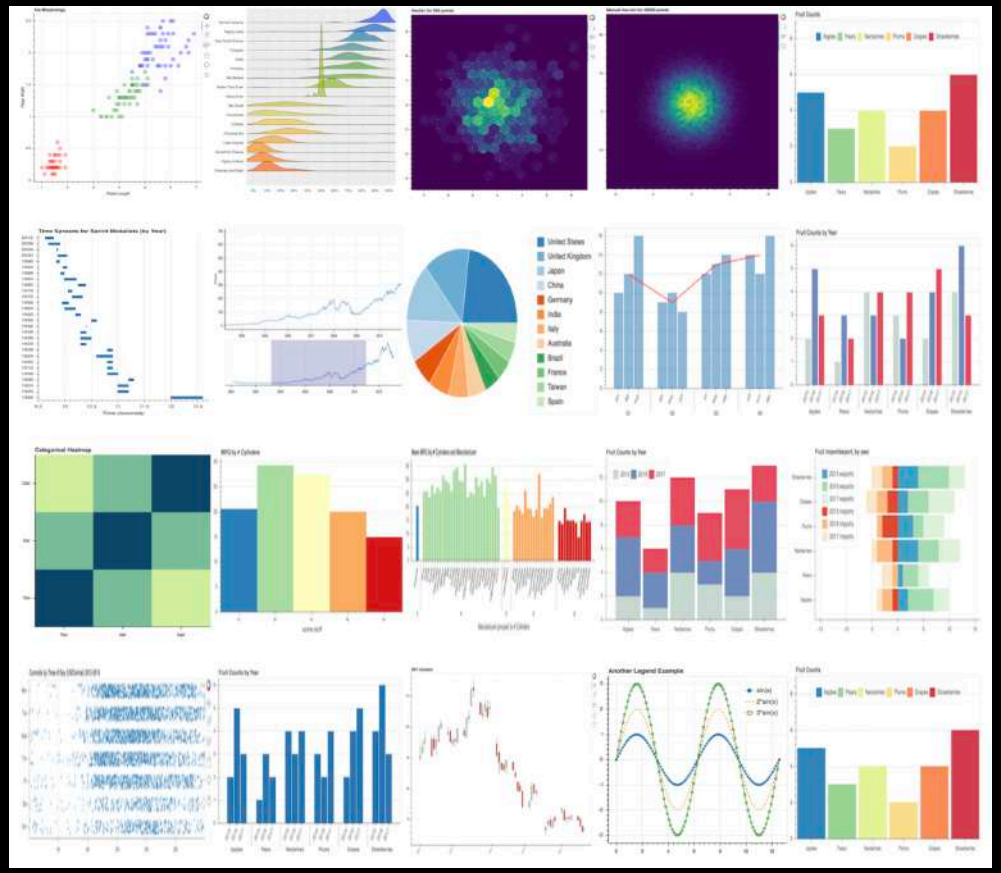


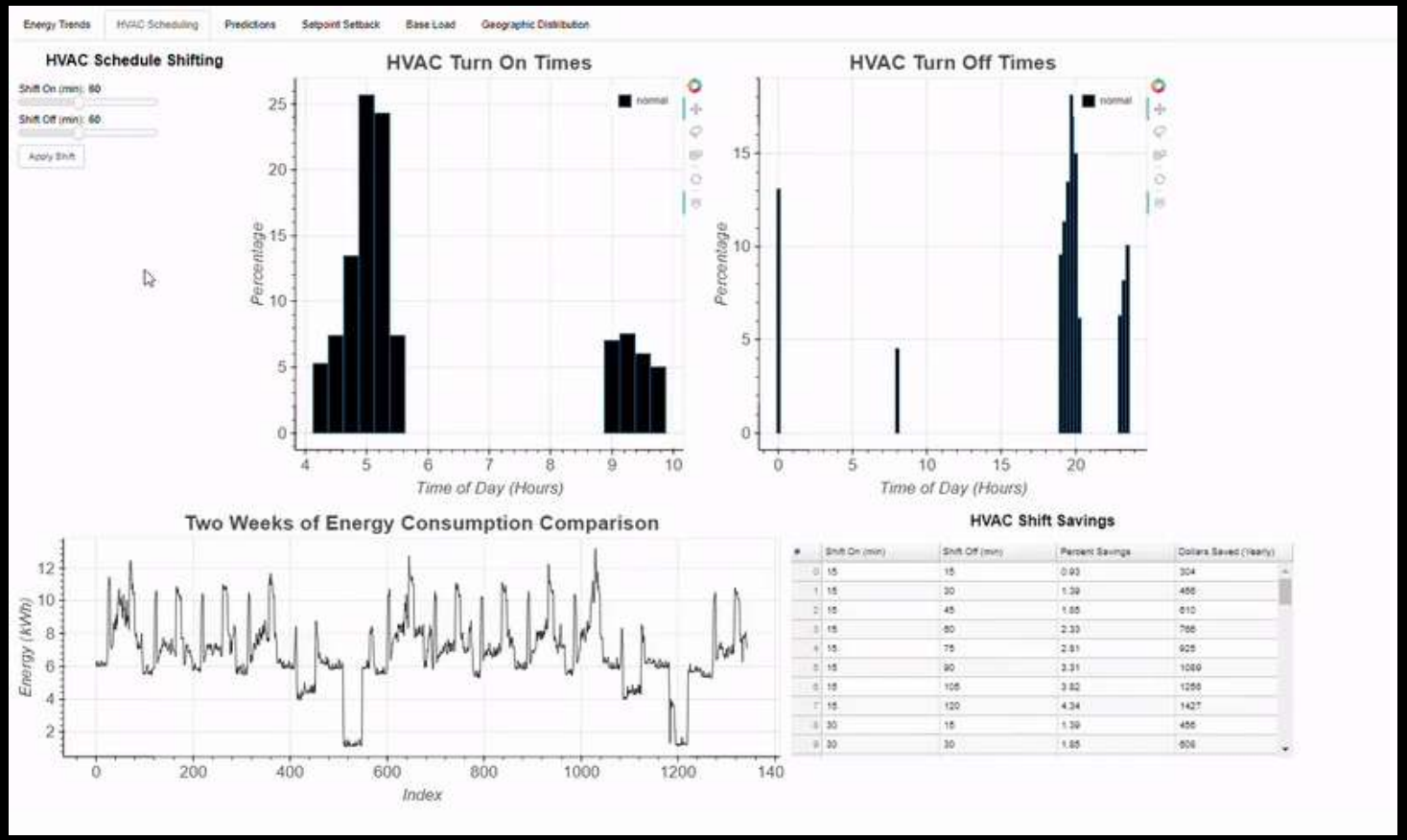




Source: https://datashader.org/index.html

Bokeh

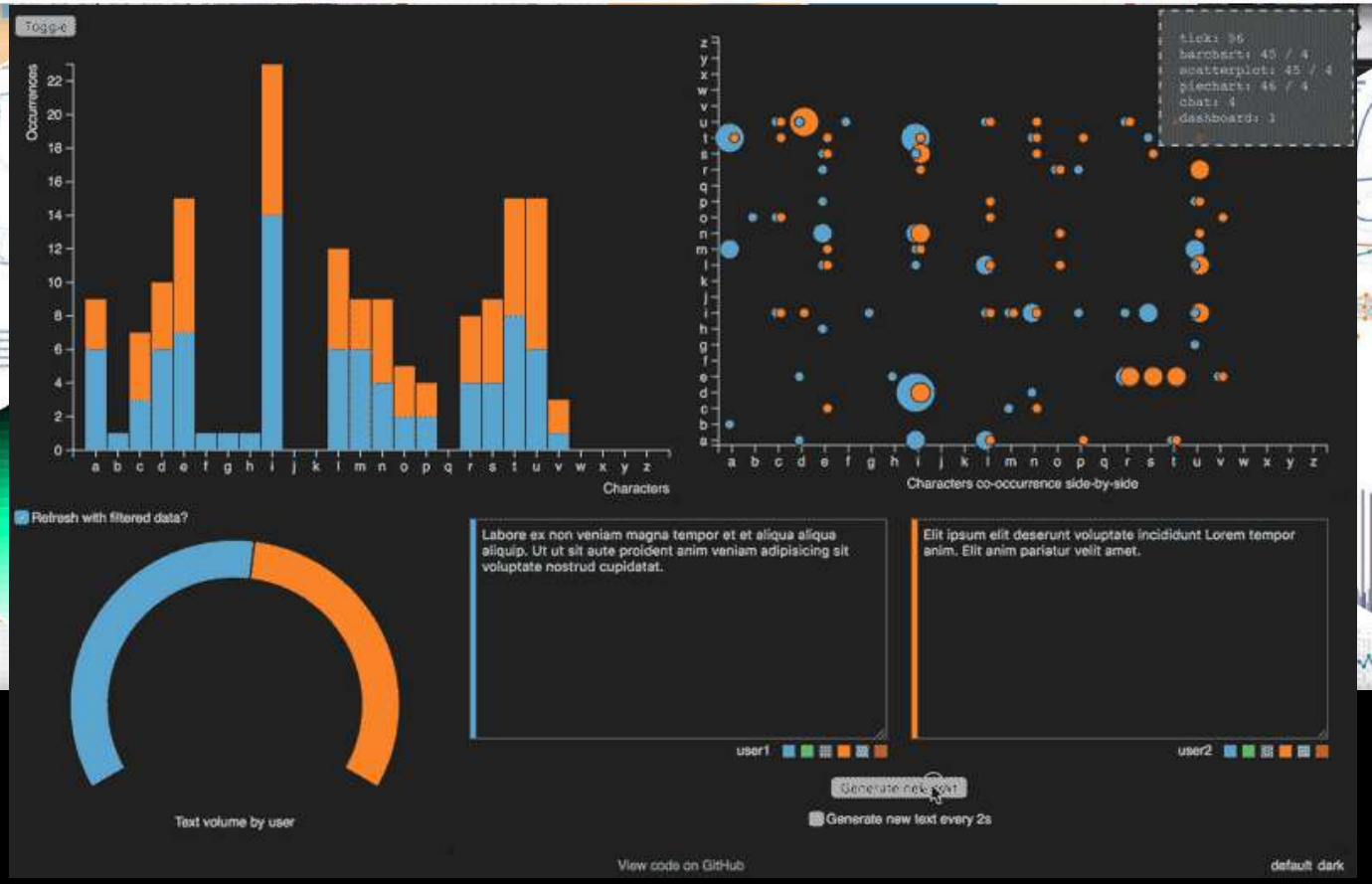


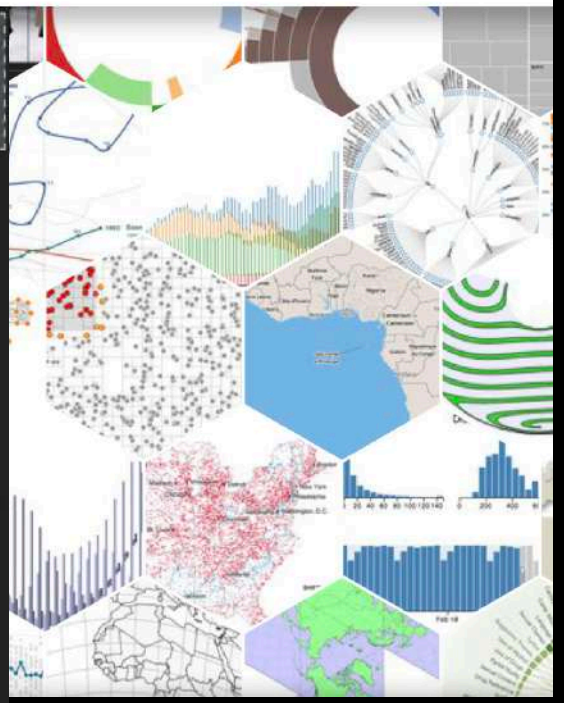


Source: https://plot.ly/python/

JavaScript. D3JS

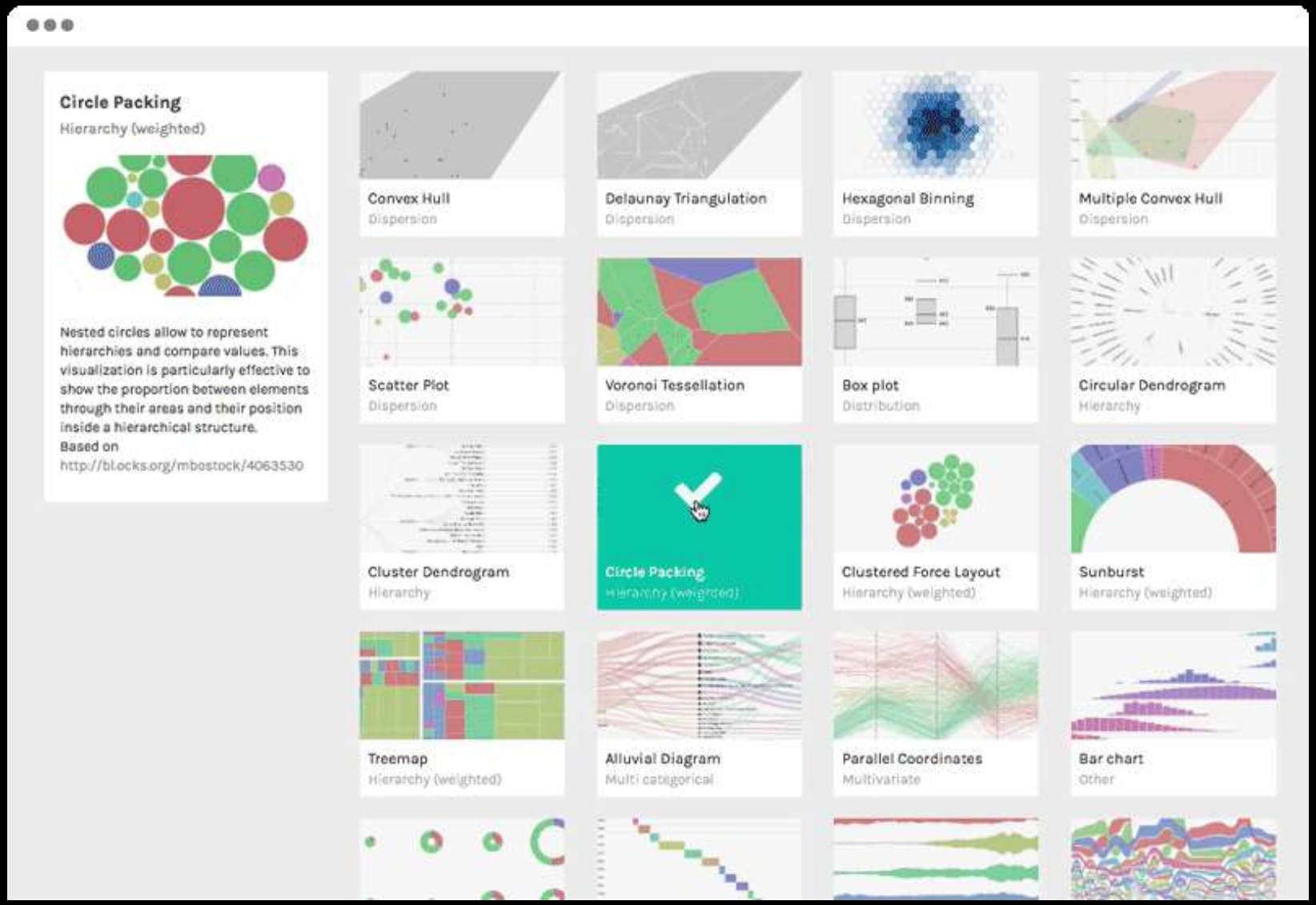
Data-Driven Documents





Source: https://github.com/d3/d3/wiki/Gallery

RAWgraphs



Source: https://rawgraphs.io

4. Graph Visualization

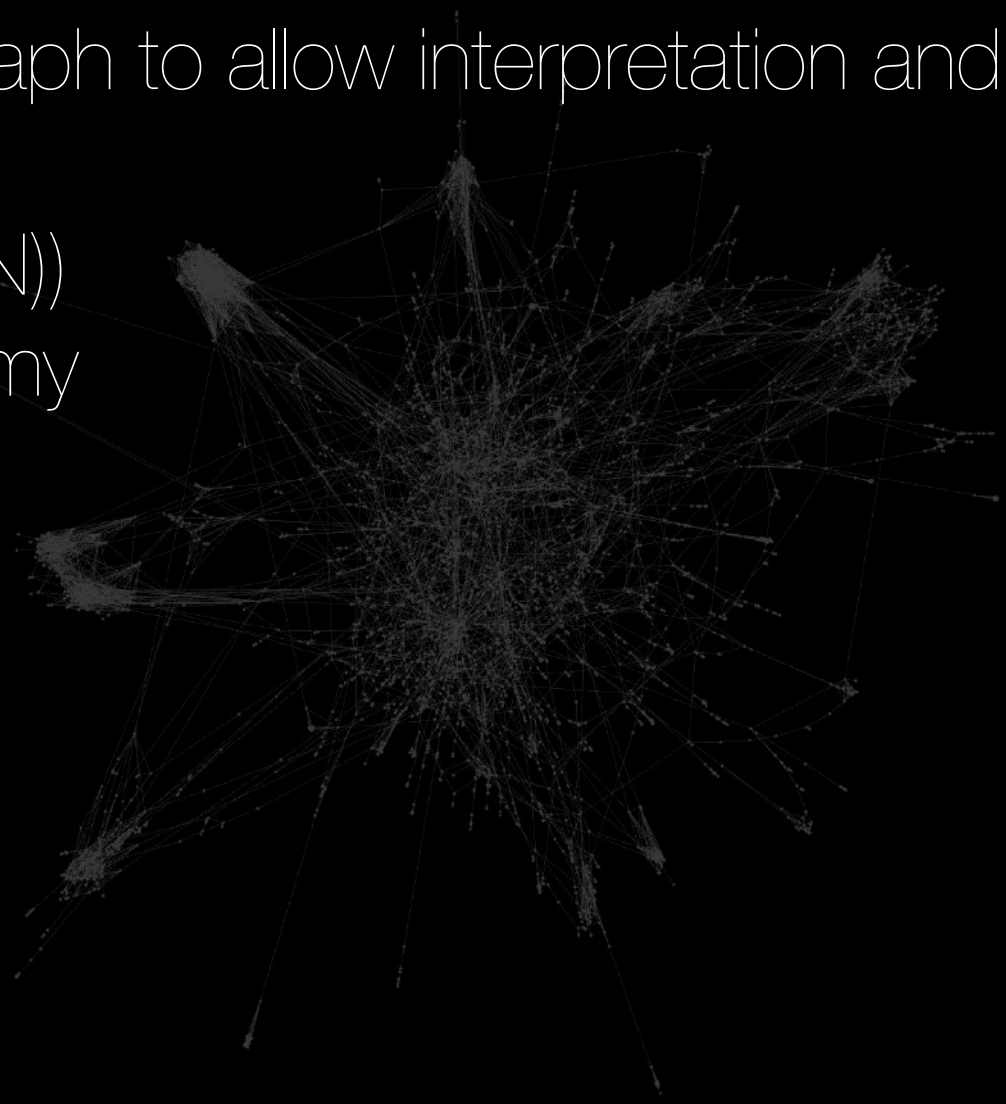
Some information appearing in this section is taken from Gephi Tutorial on Layouts by Gephi contributors https://gephi.org/about/people/

Graph Visualization

- **Layouts**. Getting **visual insights** about **the structure**
  - Force-directed
  - Circular (radial)
  - Splitters
- **Attributes**. Highlighting graph **properties, attributes and clusters**
  - Community detection
  - Graph metrics
  - Dynamics

Layouts. Force Atlas (and variations)

- **Purpose**: Spatialize graph to allow interpretation and good readability
- **Kind**: Force-directed
- **Complexity**: O(N log(N))
- **Author**: Mathieu Jacomy

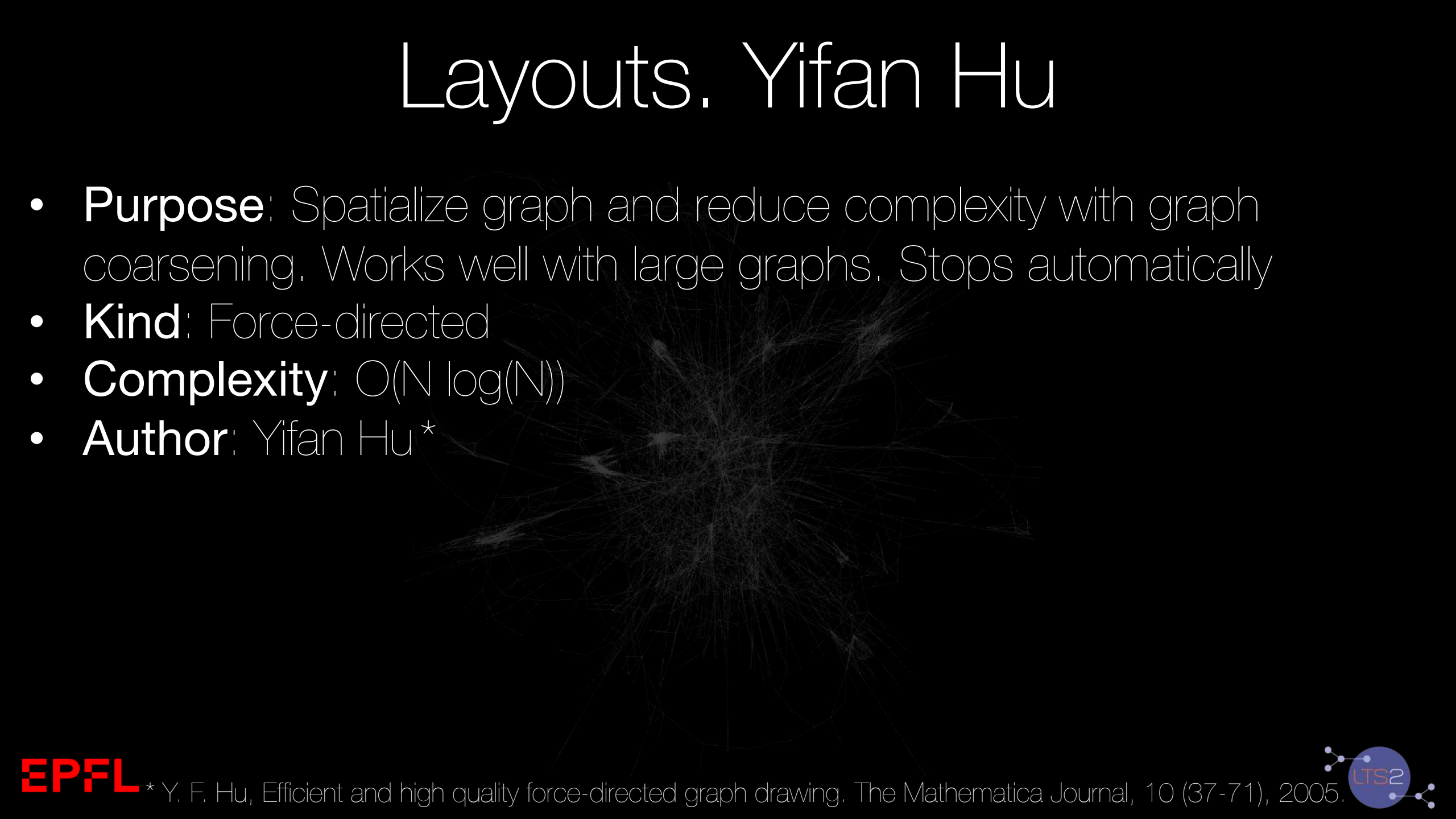


EPFL

LTS2

Layouts. Yifan Hu

- **Purpose**: Spatialize graph and reduce complexity with graph coarsening. Works well with large graphs. Stops automatically
- **Kind**: Force-directed
- **Complexity**: O(N log(N))
- **Author**: Yifan Hu *



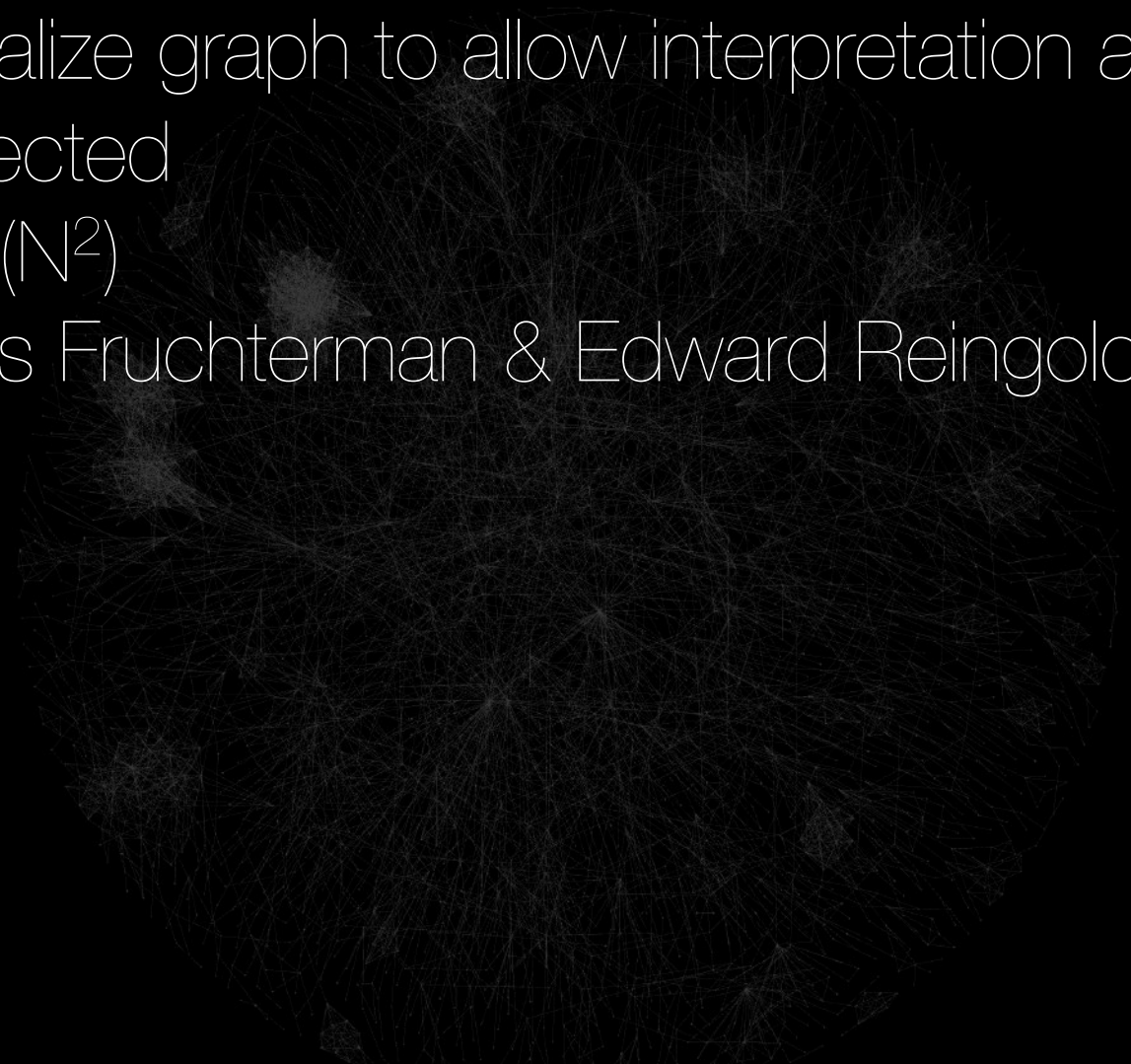
EPFL

* Y. F. Hu, Efficient and high quality force-directed graph drawing. The Mathematica Journal, 10 (37-71), 2005.

LTS2

Layouts. Fruchterman-Reingold

- **Purpose**: Spatialize graph to allow interpretation and good readability
- **Kind**: Force-directed
- **Complexity**: $O(N^2)$
- **Author**: Thomas Fruchterman & Edward Reingold *



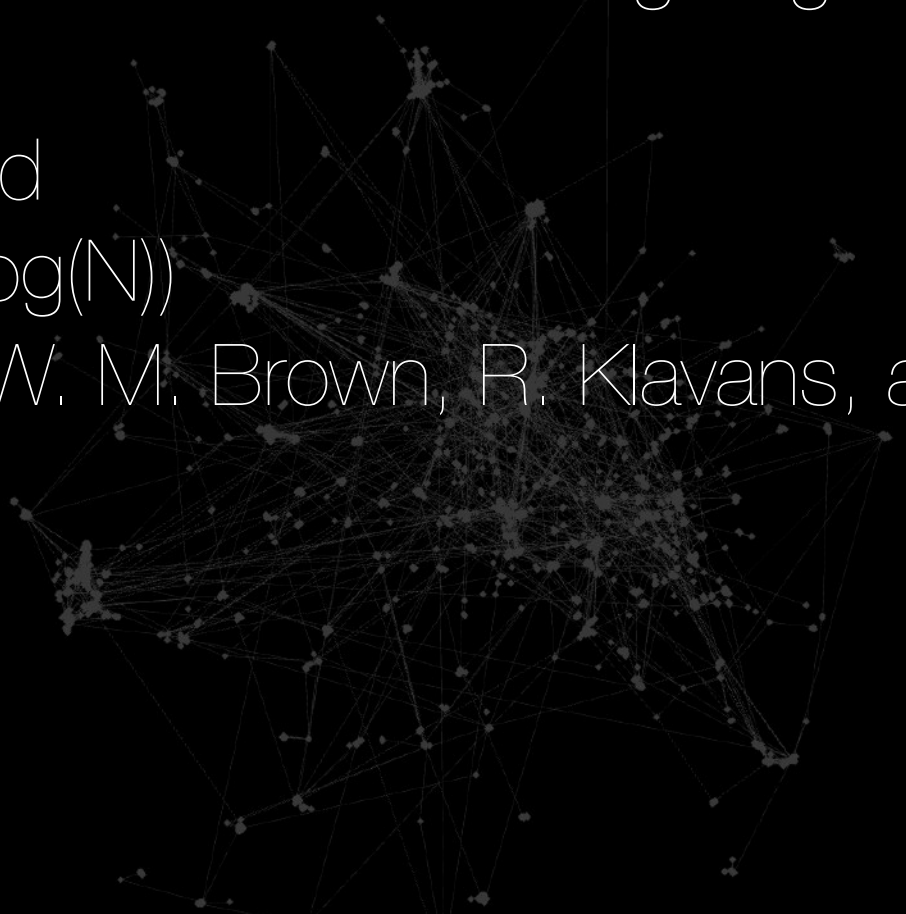
EPFL

* Fruchterman, T. M. J., & Reingold, E. M. (1991). Graph Drawing by Force-Directed Placement.

LTS2

Layouts. OpenOrd

- **Purpose**: Distinguish clusters. Cut long edges to separate clusters better
- **Kind**: Force-directed
- **Complexity**: O(N log(N))
- **Author**: S. Martin, W. M. Brown, R. Klavans, and K. Boyack *



EPFL

* S. Martin, W. M. Brown, R. Klavans, and K. Boyack, "OpenOrd: An Open-Source Toolbox for Large Graph Layout," SPIE Conference on Visualization and Data Analysis (VDA)., 2011

LTS2

Attributes. Community detection

- **Purpose**: Detect strongly-connected communities.
- **Kind**: Community detection algorithm
- **Complexity**: O(N logN)
- **Author**: Louvain * (Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre), Leiden ** (V.A. Traag, L. Waltman, and N.J. van Eck)

* Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

** V.A. Traag, L. Waltman, and N.J. van Eck, From Louvain to Leiden: guaranteeing well-connected communities, Nature Scientific Reports, 2019

EPFL

LTS2

Attributes. Node size

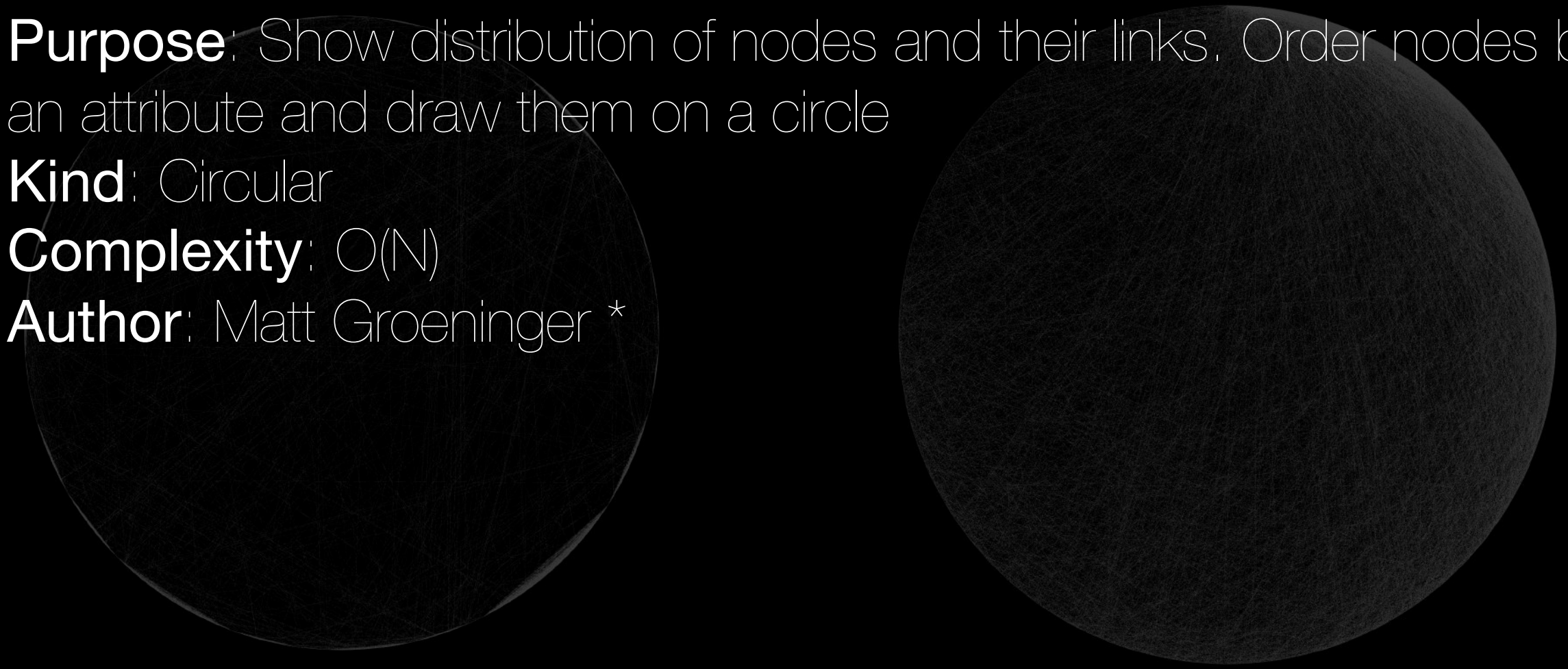
- Degree
- Centrality metrics
  - Betweenness centrality
  - Bridging centrality
- Clustering coefficient
- Page rank

EPFL

LTS2

Layouts. Circular

- **Purpose**: Show distribution of nodes and their links. Order nodes by an attribute and draw them on a circle
- **Kind**: Circular
- **Complexity**: O(N)
- **Author**: Matt Groeninger *



EPFL

* http://gephi.org/plugins/circular-layout/

LTS2

Layouts. Radial Axis

- **Purpose**: Study homophily by showing distributions of nodes inside groups. Axes radiate from the central circle. Group nodes by an attribute and draw them in axes
- **Kind**: Circular
- **Complexity**: O(N)
- **Author**: Matt Groeninger *

EPFL

* http://gephi.org/plugins/circular-layout/

LTS2

Layouts. Circle pack

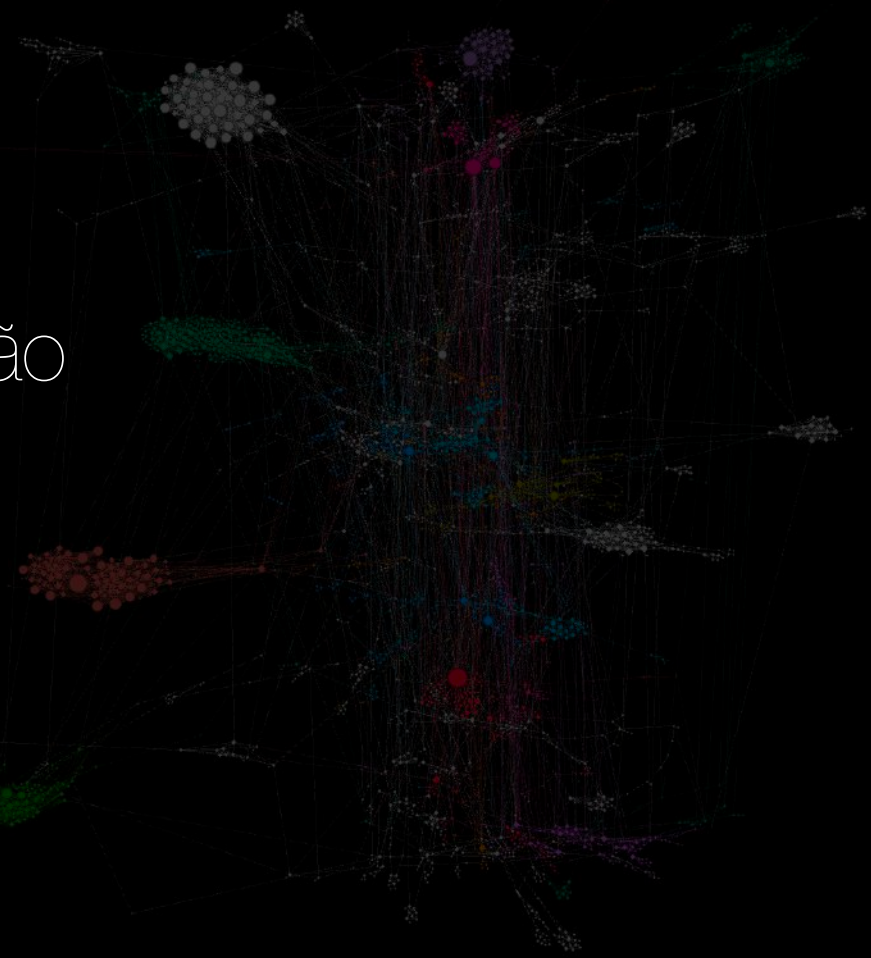
- **Purpose**: Spatialize graph using attributes (e.g. community)
- **Kind**: Splitter
- **Complexity**: O(1)
- **Author**: Mike Bostock, Patrick Murray, Ken Goulding

EPFL

LTS2

Layouts. 3D Network Splitter

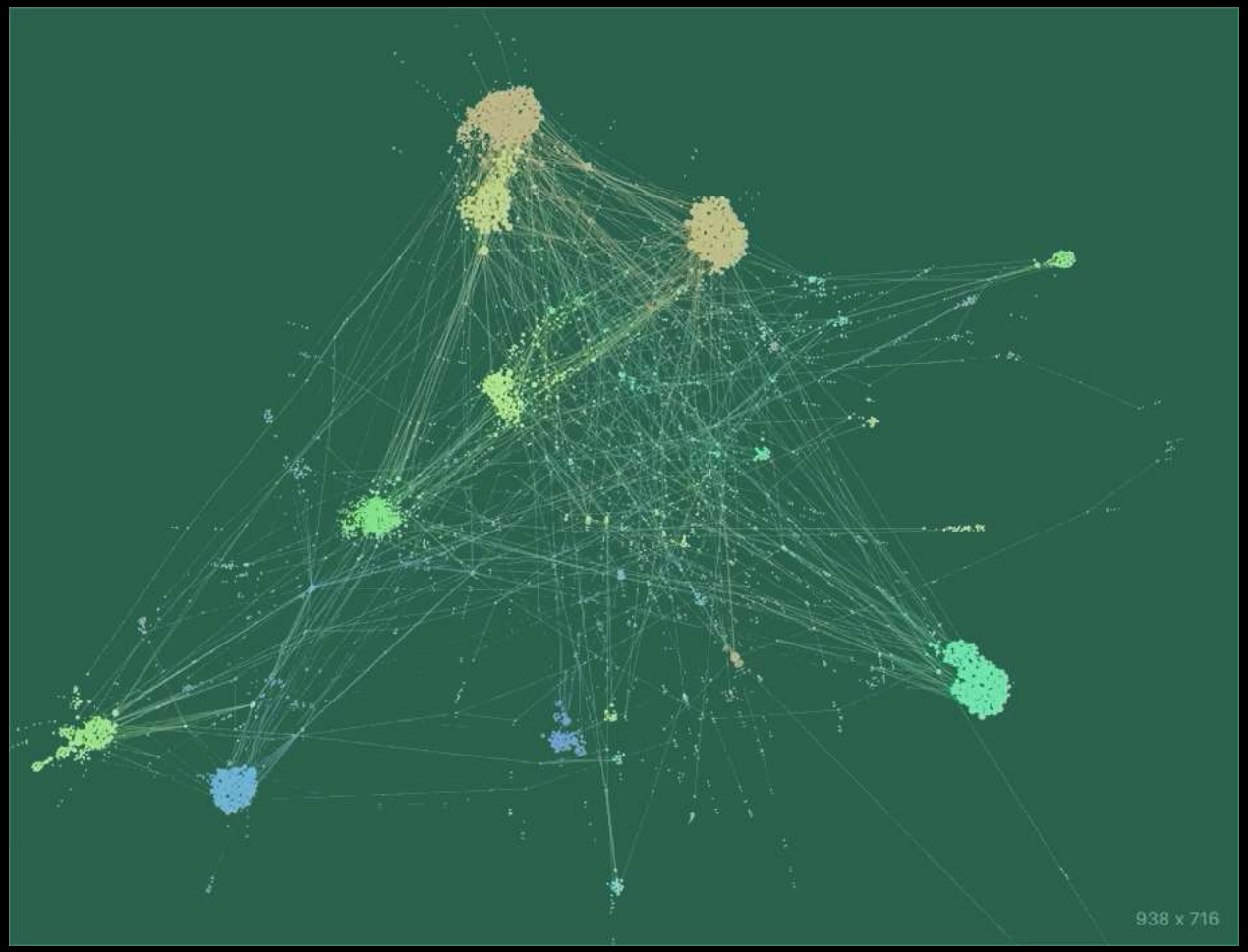
- **Purpose**: Unfold graph in layers based on user-defined attributes (e.g. communities)
- **Kind**: Splitter
- **Complexity**: O(1)
- **Author**: Alexandre Barão



EPFL

LTS2

Attributes. Dynamics



5. Tutorials

Graph Visualization Tutorials

- Overview of graph visualization tools
  - GraphTool
  - Cytoscape
  - Gephi
- Gephi Tutorials

EPFL

LTS2

Tools. GraphTool

graph-tool | Efficient network analysis

Download | Documentation | Mailing List | Git | Issues

What is graph-tool?

Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (a.k.a. networks). Contrary to most other python modules with similar functionality, the core data structures and algorithms are implemented in C++, making extensive use of template metaprogramming, based heavily on the Boost Graph Library. This confers it a level of performance that is comparable (both in memory usage and computation time) to that of a pure C/C++ library.

Download version 2.29

See Instructions | See Changelog

It is *Fast!*

Despite its nice, soft outer appearance of a regular python module, the core algorithms and data structures of graph-tool are written in C++, with performance in mind. Most of the time, you can expect the algorithms to run just as fast as if graph-tool were a pure C/C++ library. See a performance comparison.

OpenMP Support

Many algorithms are implemented in parallel using OpenMP, which provides excellent performance on multi-core architectures, without degrading it on single-core machines.

Extensive Features

An extensive array of features is included, such as support for arbitrary vertex, edge or graph properties, efficient "on the fly" filtering of vertices and edges, powerful graph I/O using the GraphML, GML and dot file formats, graph pickling, graph statistics (degree/property histogram, vertex correlations, average shortest distance, etc.), centrality measures, standard topological algorithms (isomorphism, minimum spanning tree, connected components, dominator tree, maximum flow, etc.), generation of random graphs with arbitrary degrees and correlations, detection of modules and communities via statistical inference, and much more.

Powerful Visualization

Conveniently draw your graphs, using a variety of algorithms and output formats (including to the screen). Graph-tool has its own layout algorithms and versatile, interactive drawing routines based on cairo and GTK+, but it can also work as a very comfortable interface to the excellent graphviz package.

Fully Documented

Every single function in the module is documented in the docstrings and in the online documentation, which is full of examples.

EPFL

Source: https://graph-tool.skewed.de

LTS2

Tools. Cytoscape

Source: https://cytoscape.org/what_is_cytoscape.html

Tools: Gephi

open source initiative®

Gephi
makes graphs handy

Download | Blog | Wiki | Forum | Support | Bug tracker

Home | Features | Learn | Develop | Plugins | Services | Consortium

The Open Graph Viz Platform

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

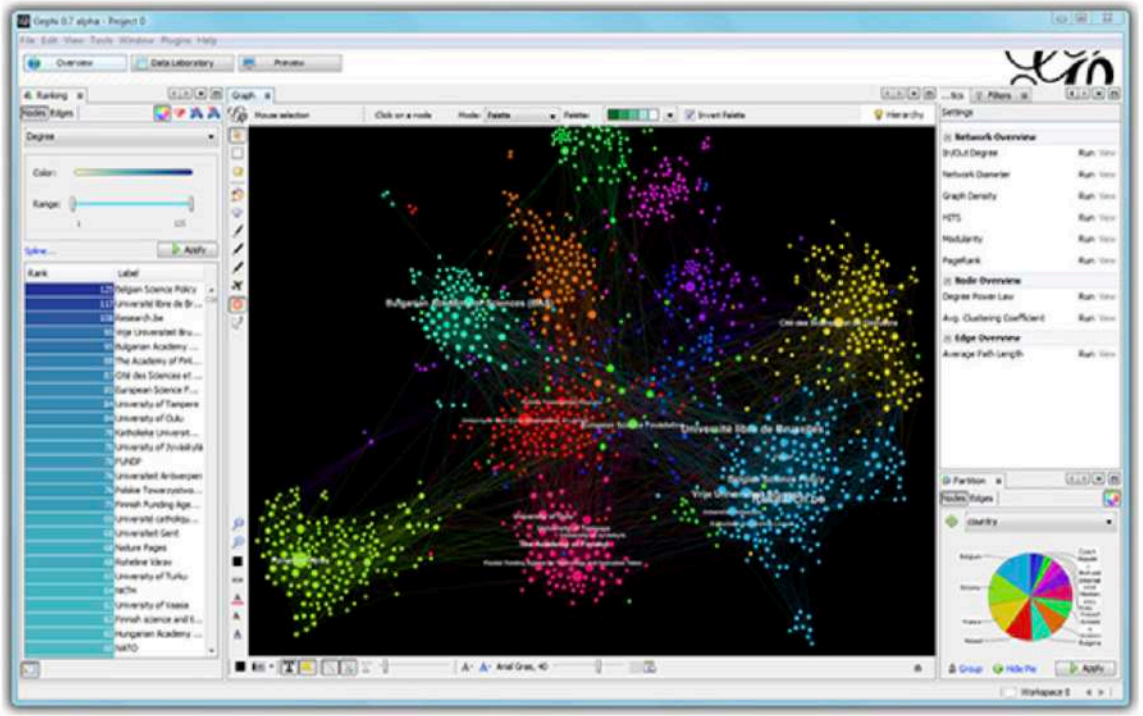
Runs on Windows, Mac OS X and Linux.

Learn More on Gephi Platform »

Download FREE
Gephi 0.9.2

Release Notes | System Requirements

- Features
- Quick start
- Screenshots
- Videos



EPFL

LTS2

Source: https://gephi.org

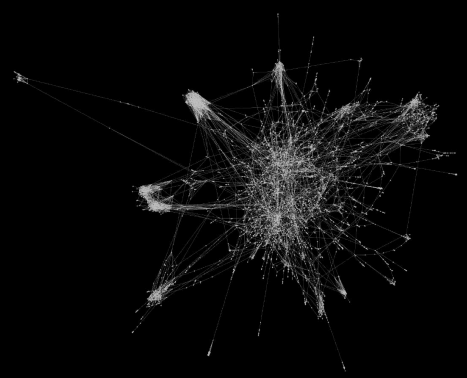
Gephi pipeline

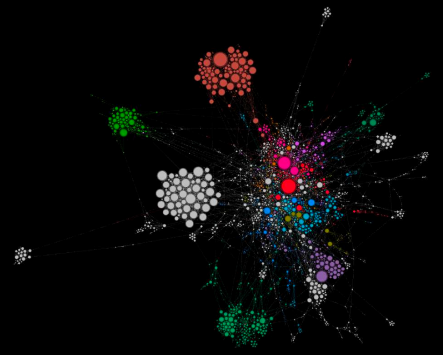
Load graph

gexf, graphml, gdf, csv etc.

→ Compute layout → Compute attributes → Design visualization → Publish online

GitHub pages





GitHub Pages

Tutorials: https://github.com/mizvol/gephi-tutorials

EPFL

LTS2

Questions?