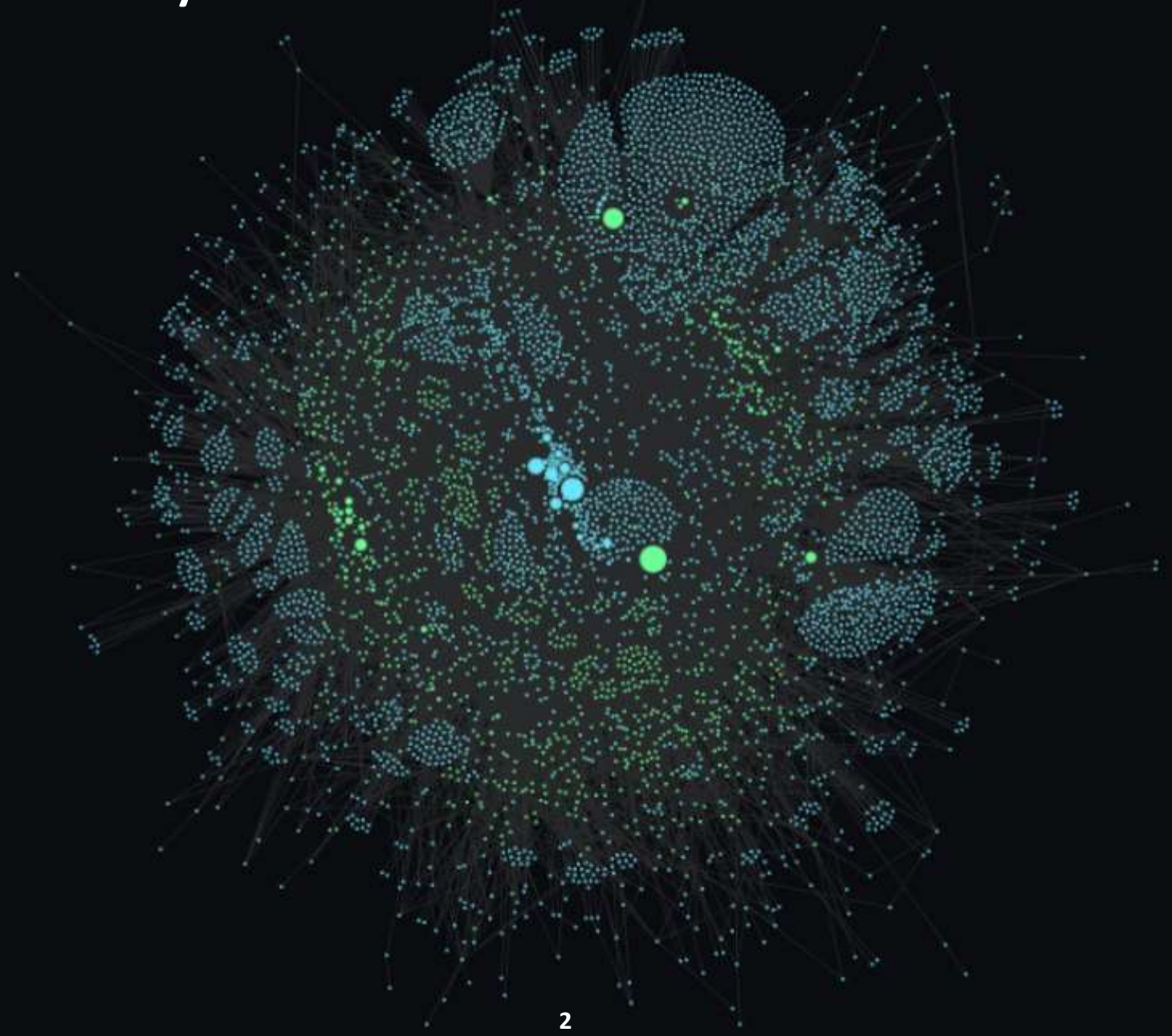
Introduction to Visual Data Science
Exploratory Visual Analysis

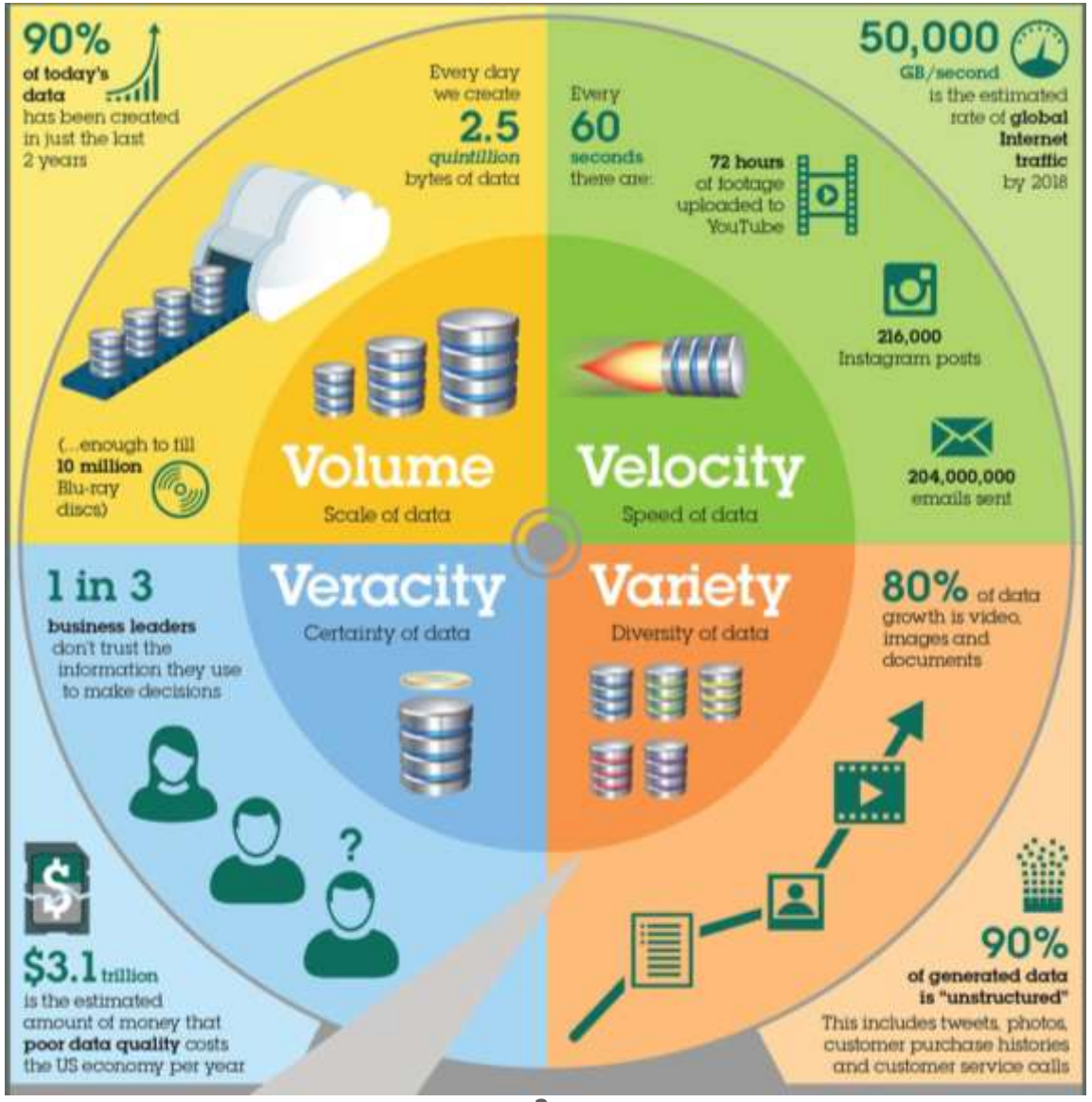
Manuela Waldner

Institute of Visual Computing & Human-Centered Technology, TU Wien, Austria

Why Do We Need Data Science?



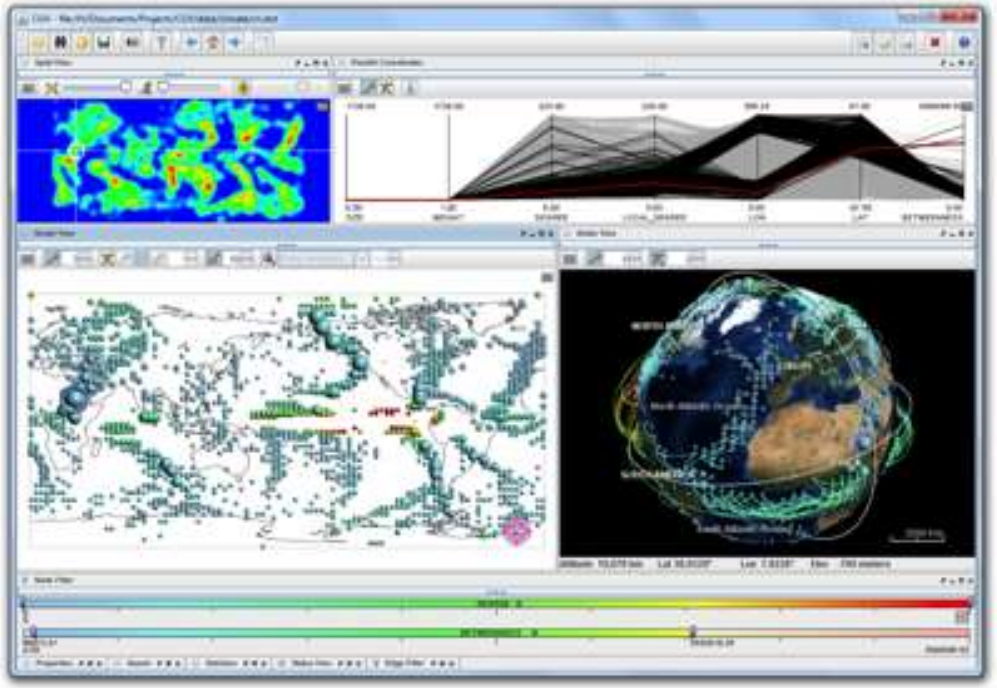
Big Data



Source: IBM

Application Areas

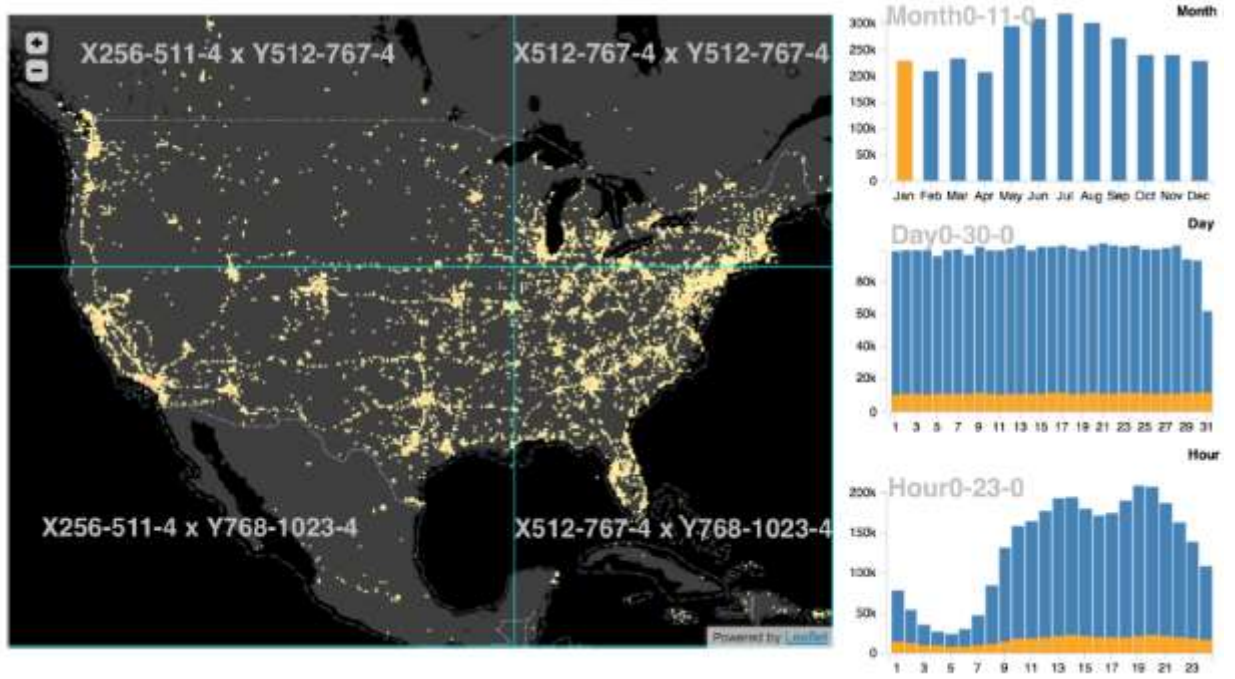
- Processing and analyzing large information spaces
  - Astronomy
  - Physics
  - Climate research
  - Biology (e.g., genome-wide association studies)
  - Medicine (e.g., cohort studies)
  - Security
  - Commerce
  - Social sciences
  - …



http://www.vismaster.eu/book/chapter-2-visual-analytics/

Example: Social Networks

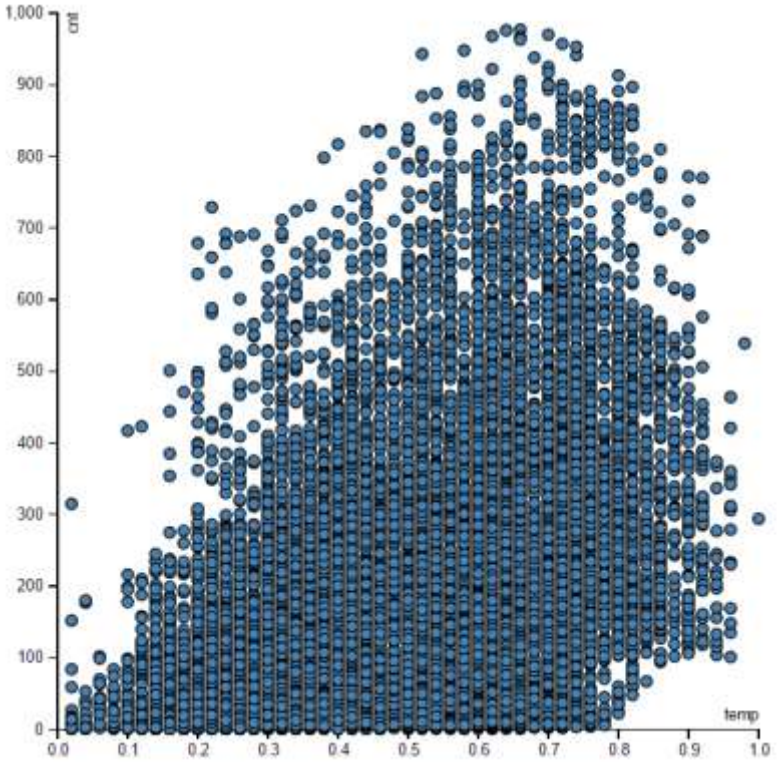
- Example: Brightkite
  - Millions of location „check-in‘s“
  - When do people go where?



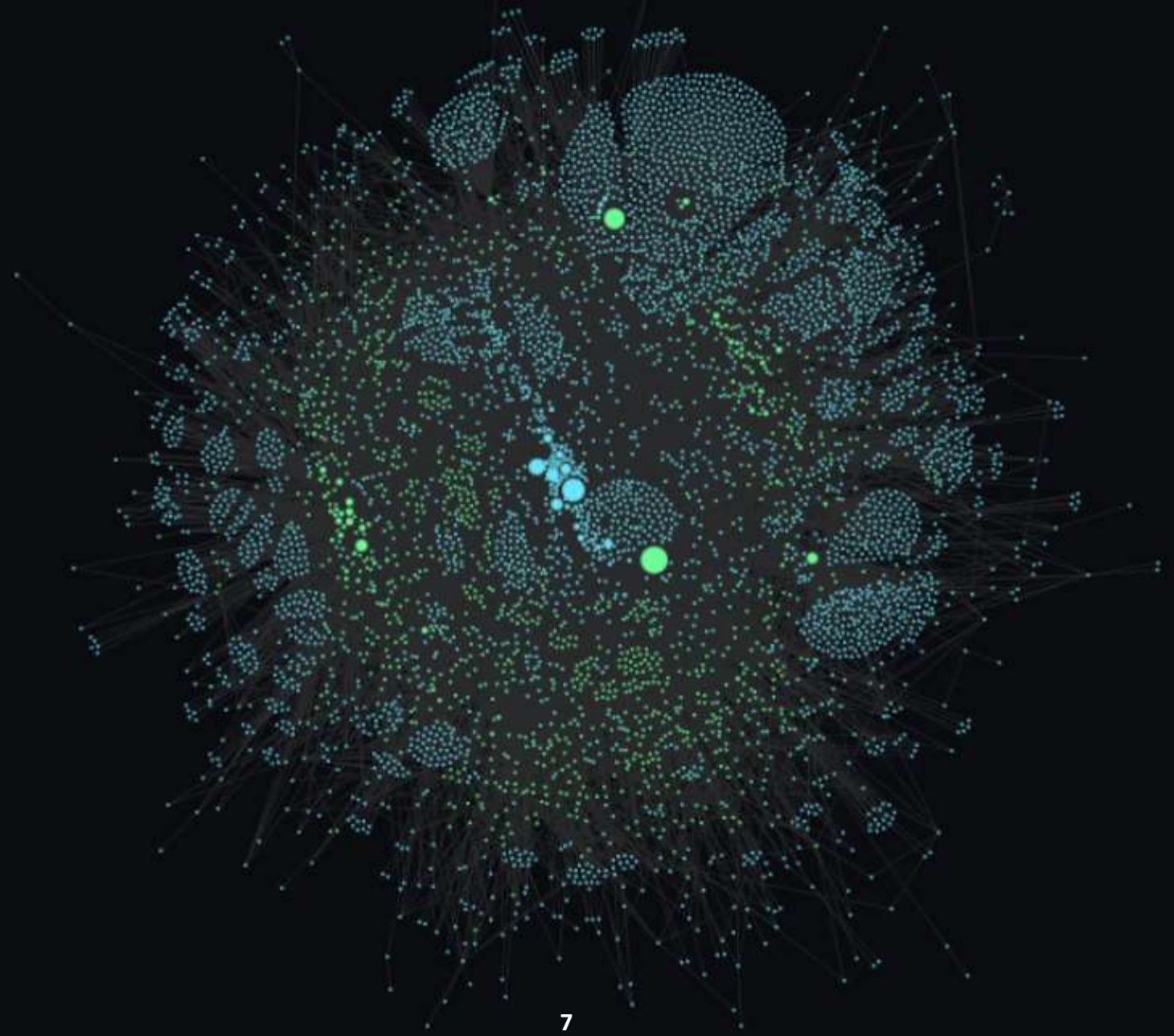
[Lui, Jiang & Heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013]

Example: Bike Rental Data

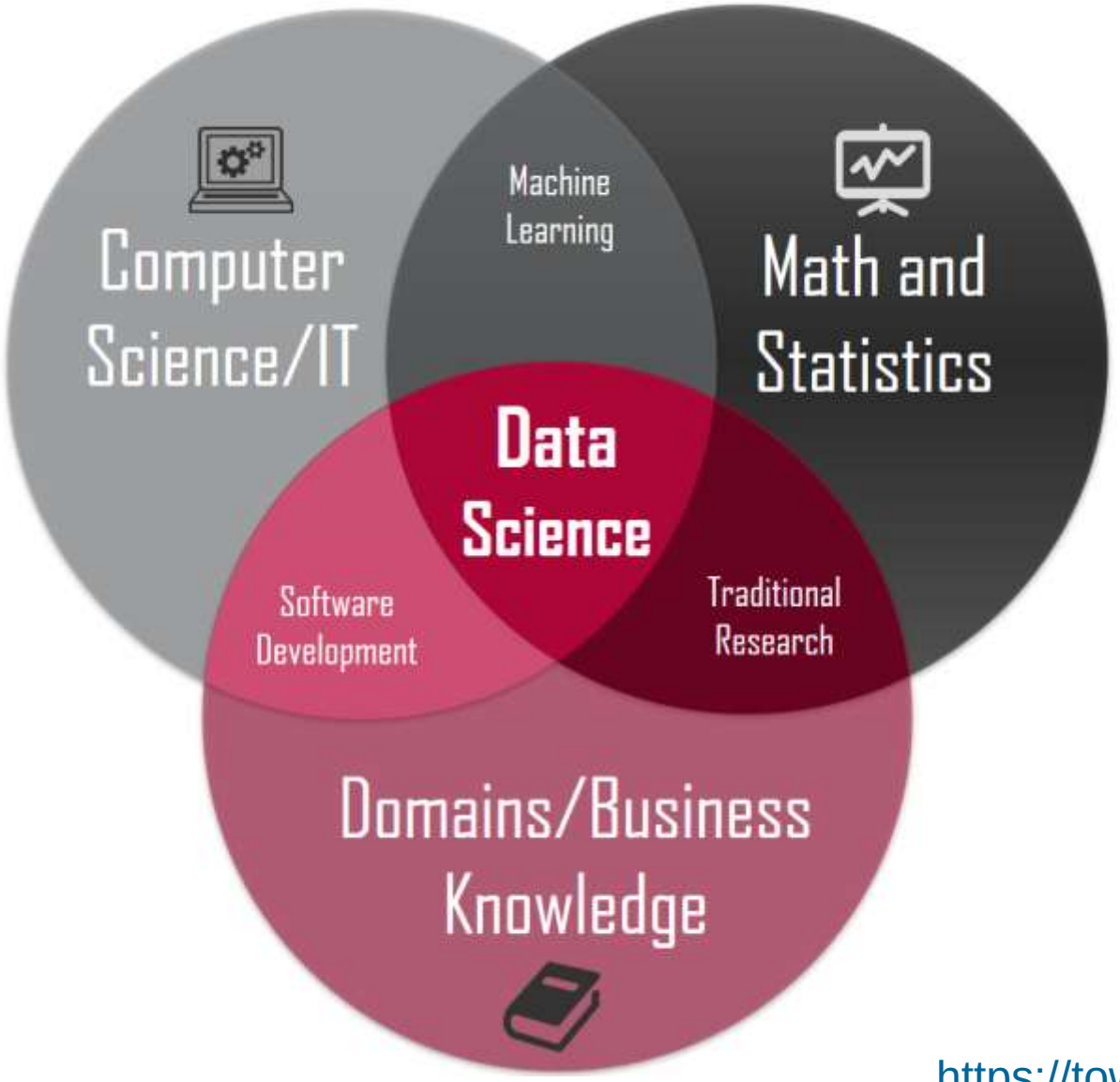
- Tens of thousands of bike rental counts
- 16 attributes
  - Number of bike rentals
  - Date
  - Week day
  - Temperature
  - Wind speed
  - ...
- When do people tend to rent a lot of bikes?



What is Data Science?



Data Science

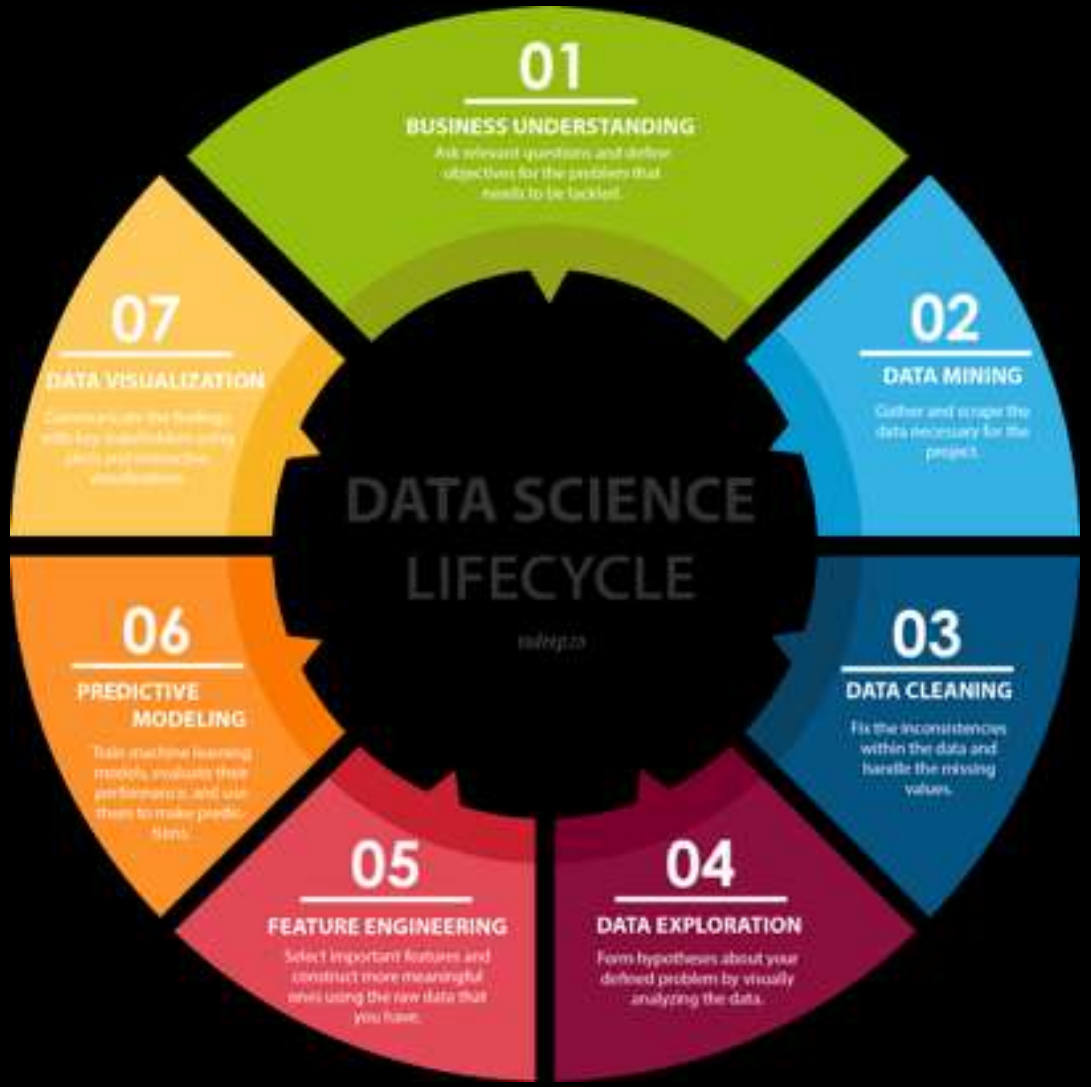


https://towardsdatascience.com

The Data Science Process

prediction
accuracy?
appropriateness?

Trust!



data =
<u>useful</u>
information?

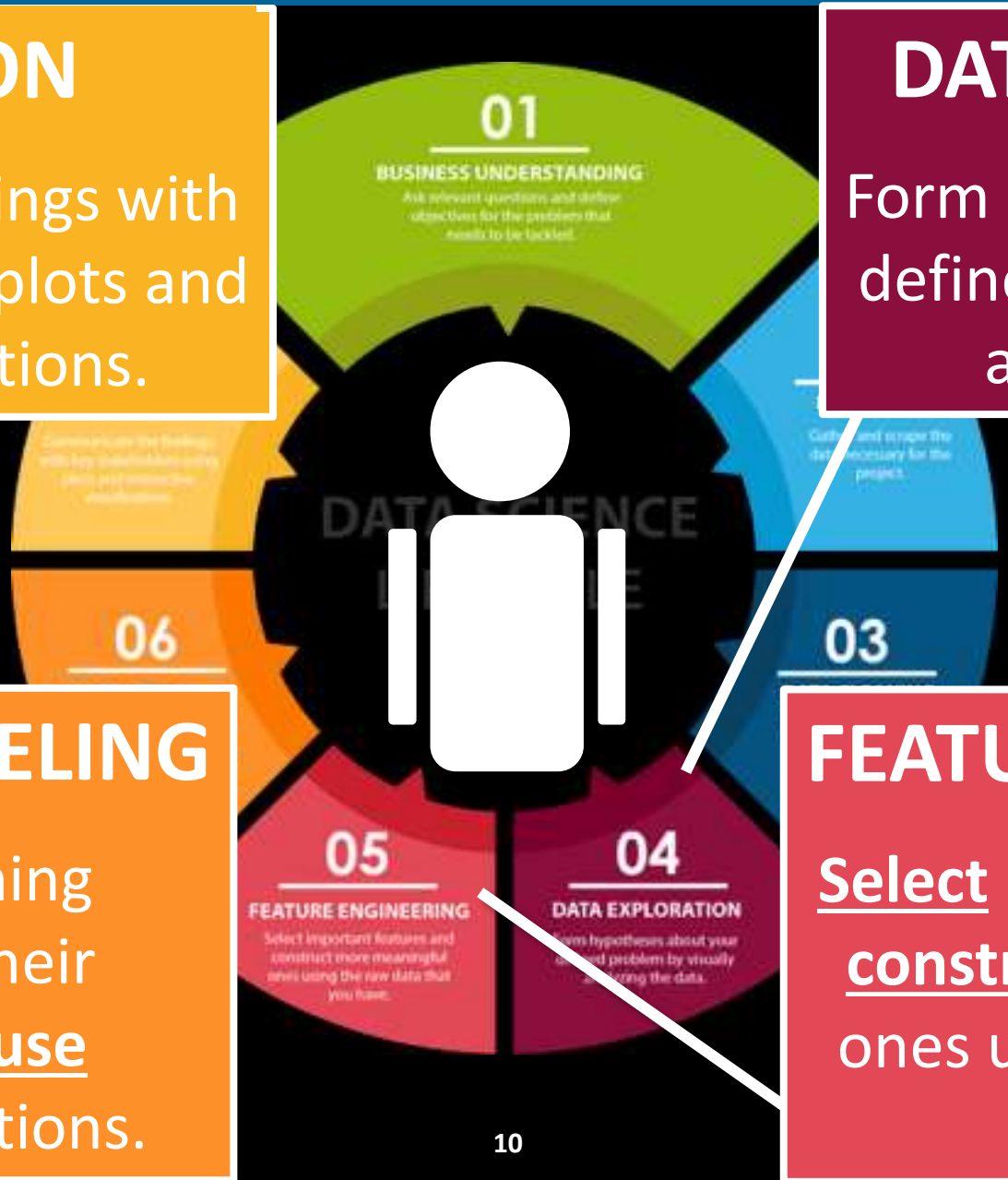
The Visual Data Science Process

PRESENTATION

Communicate the findings with key stakeholders using plots and interactive visualizations.

DATA EXPLORATION

Form hypotheses about your defined problem by **visually** analyzing the data.



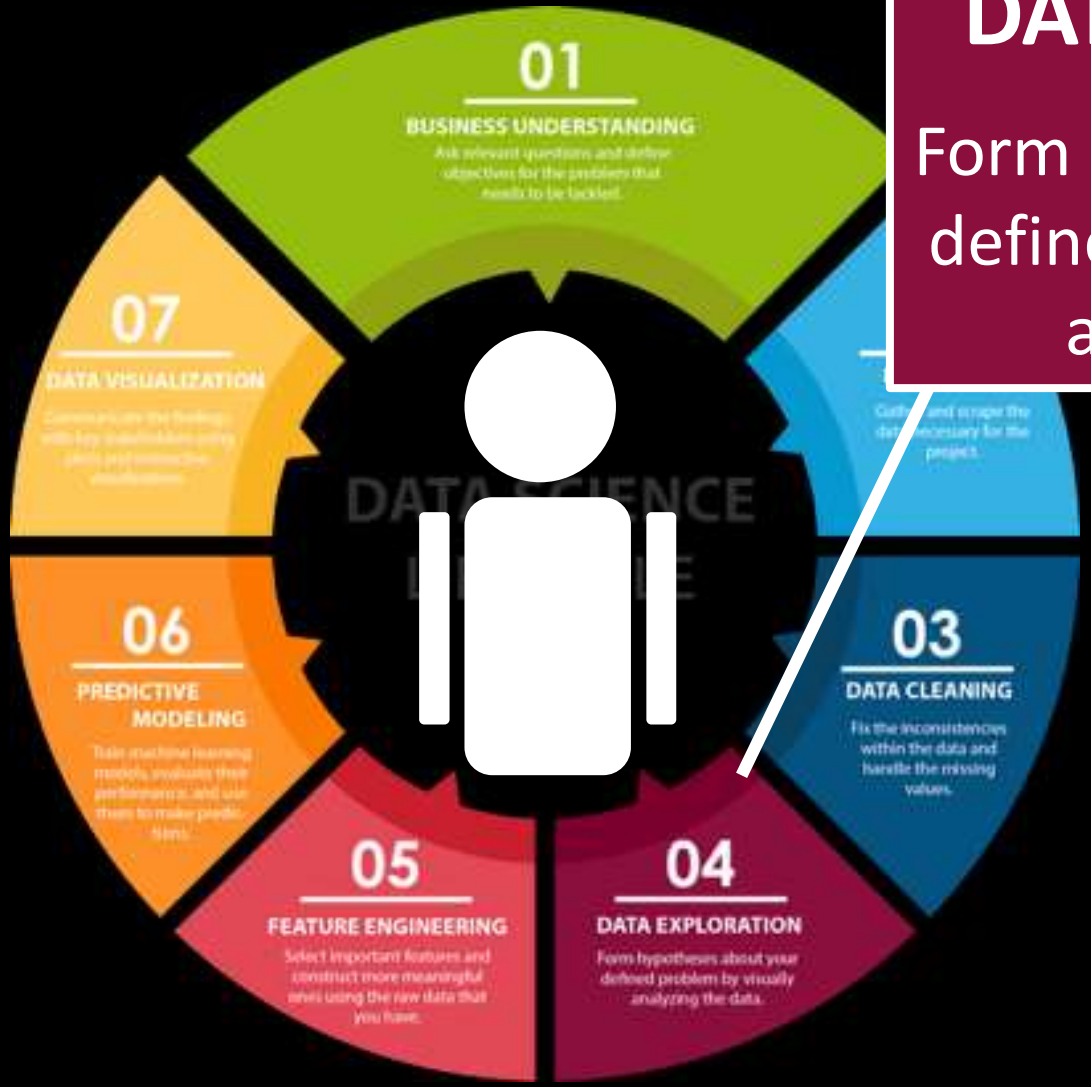
PREDICTIVE MODELING

Train machine learning models, **evaluate** their performance, and **use** them to make predictions.

FEATURE ENGINEERING

Select important features and **construct** more meaningful ones using the raw data that you have.

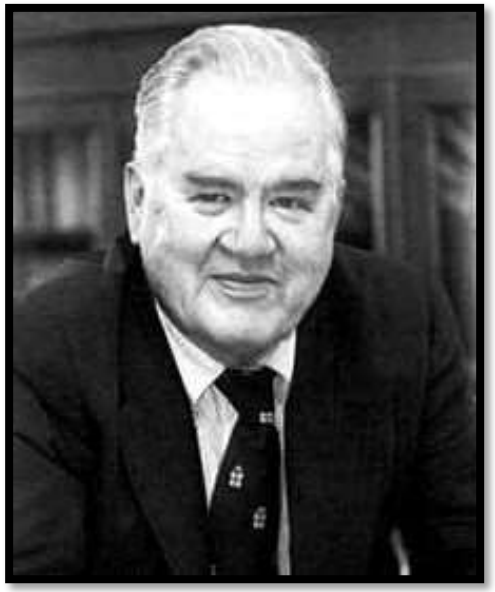
The Data Science Process



DATA EXPLORATION

Form hypotheses about your defined problem by **visually** analyzing the data.

Data Exploration



"Finding the question is often more important than finding the answer"

[John W. Tukey]

- What is big data?

Key	Value
Key 1	Value 1
Key 2	Value 2
Key 3	Value 3
...	...

.... billions of records

→ **tall** data

Key	Variable 1	Variable 2	...
Key 1	Value 1	Value 1	...
Key 2	Value 2	Value 2	...
Key 3	Value 3	Value 3	...
...	...	...	...

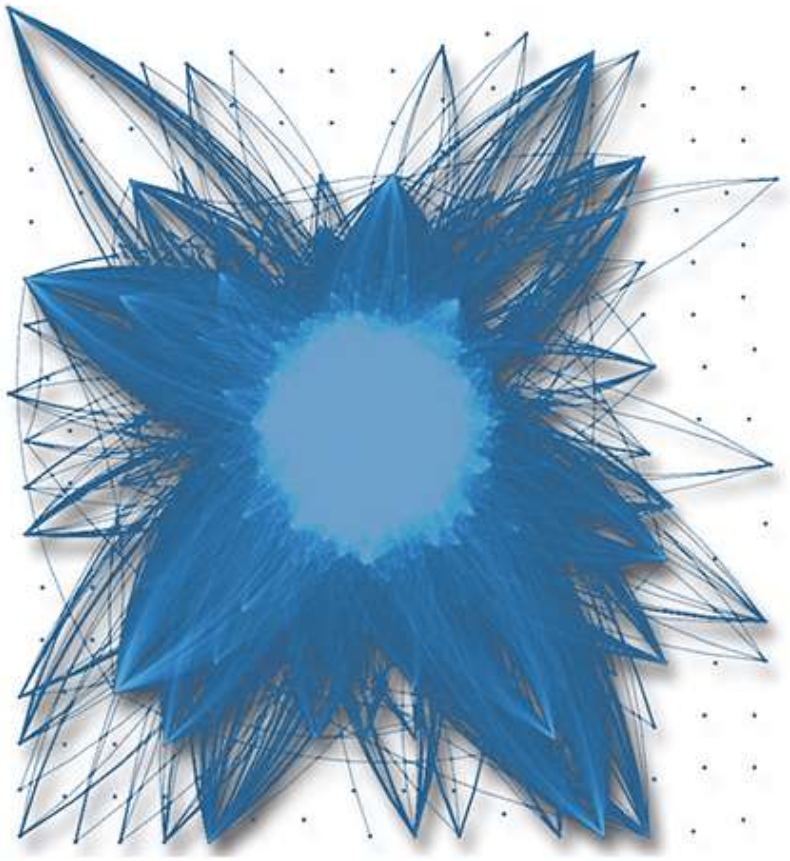
.... thousands of variables

→ **wide** data

[Heer & Kandel, Interactive Analysis of Big Data, ACM XRDS 2012]

Exploratory Analysis of Big Data

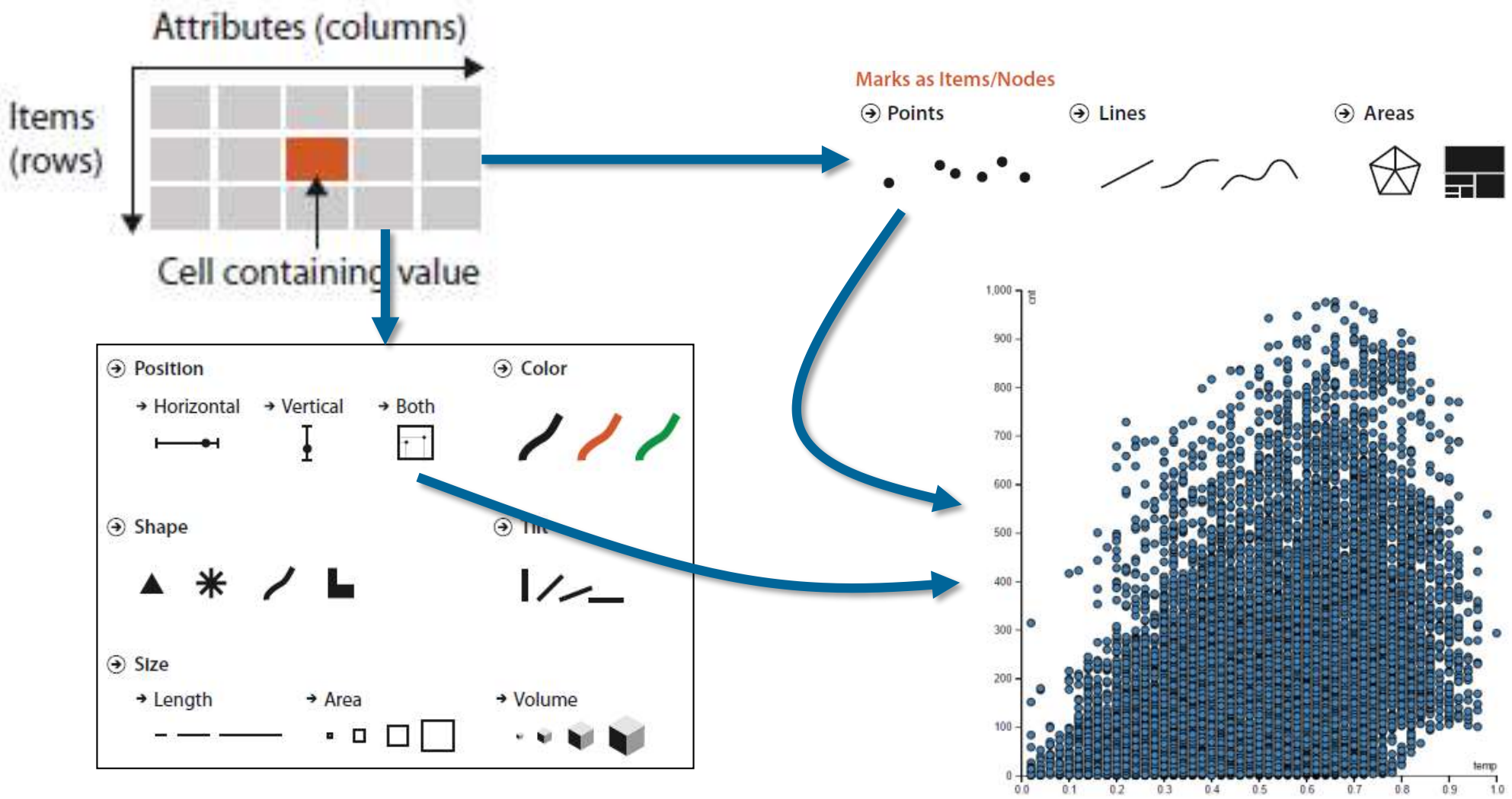
- What is big data?



[van den Elzen and van Wijk, TVCG 2014]

... Hundrets of thousands to millions of nodes and links

Recap: Data → Visual Structures

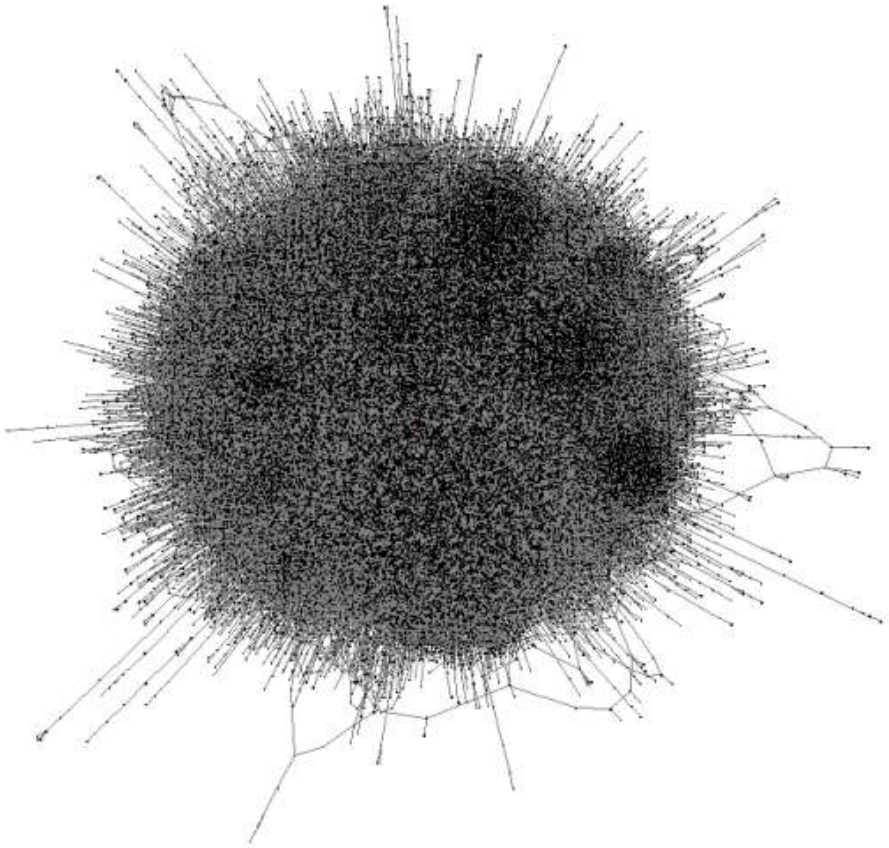


Visualization Challenges

Example: “Parallel coordinates” with hundred thousands of items and hundreds of dimensions:

Visualization Challenges

Graph “Hairball”



https://linkurio.us/blog/network-management-and-impact-analysis-with-neo4j/

Visual Exploration of Big Data

- Challenges:
  - Perceptual scalability
    - Display resolution < number of data points
    - Perceptual limitations
  - Interactive scalability
    - Inefficient database queries
    - Inefficient data processing
    - Rendering performance

[Geymayer & Schmalstieg, 2016]

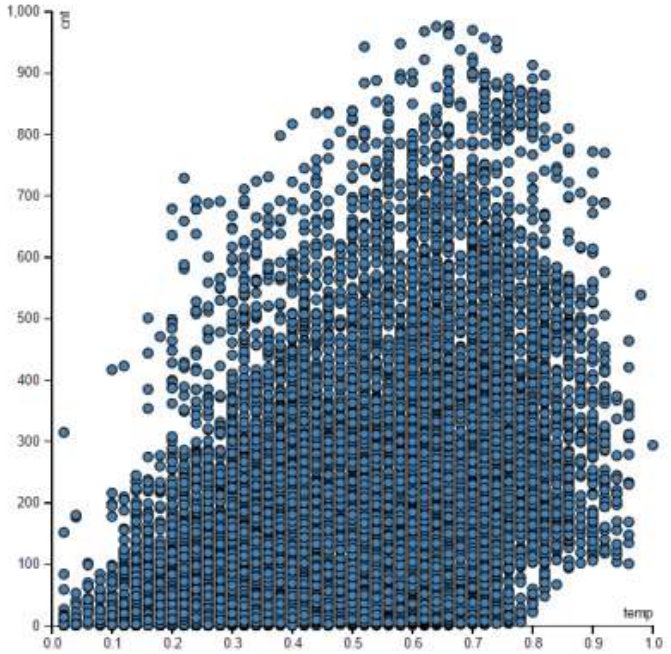
[Liu et al., imMens: Real-time Visual Querying of Big Data, EuroVis 2013]

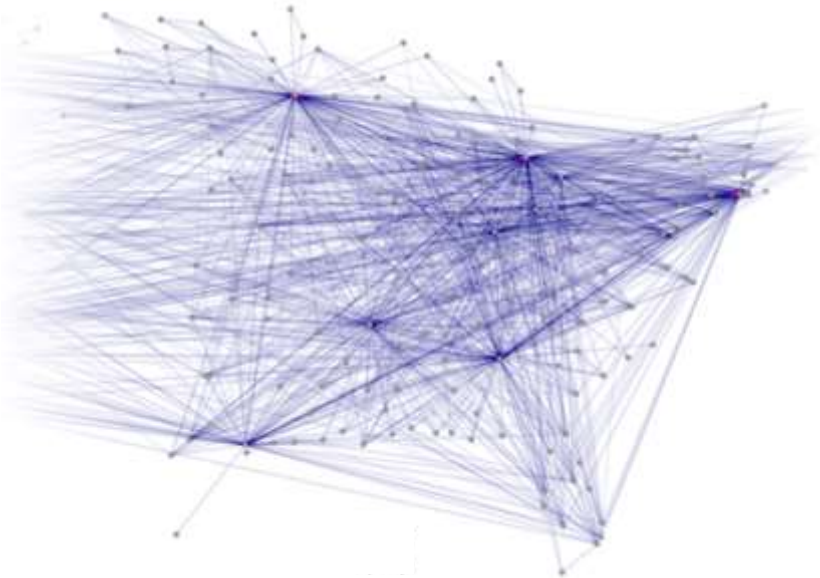
Factors Affecting Visual Scalability

- Human perception
  - Precision of eye and ability of human mind
- Monitor resolution
  - Physical size and pixel resolution
- Visual metaphors
  - Selection of metaphor and visual channels
- Interactivity
  - Panning, zooming, ...
- Data structures and algorithms
  - E.g., graph layout algorithms, data cubes,...
- Computational infrastructure
  - CPU, GPU, network

Overplotting

- Each data record is mapped to visual item
- Problems with „big data“:
  - occlusion
  - clutter
  - readability
  - waste of resources

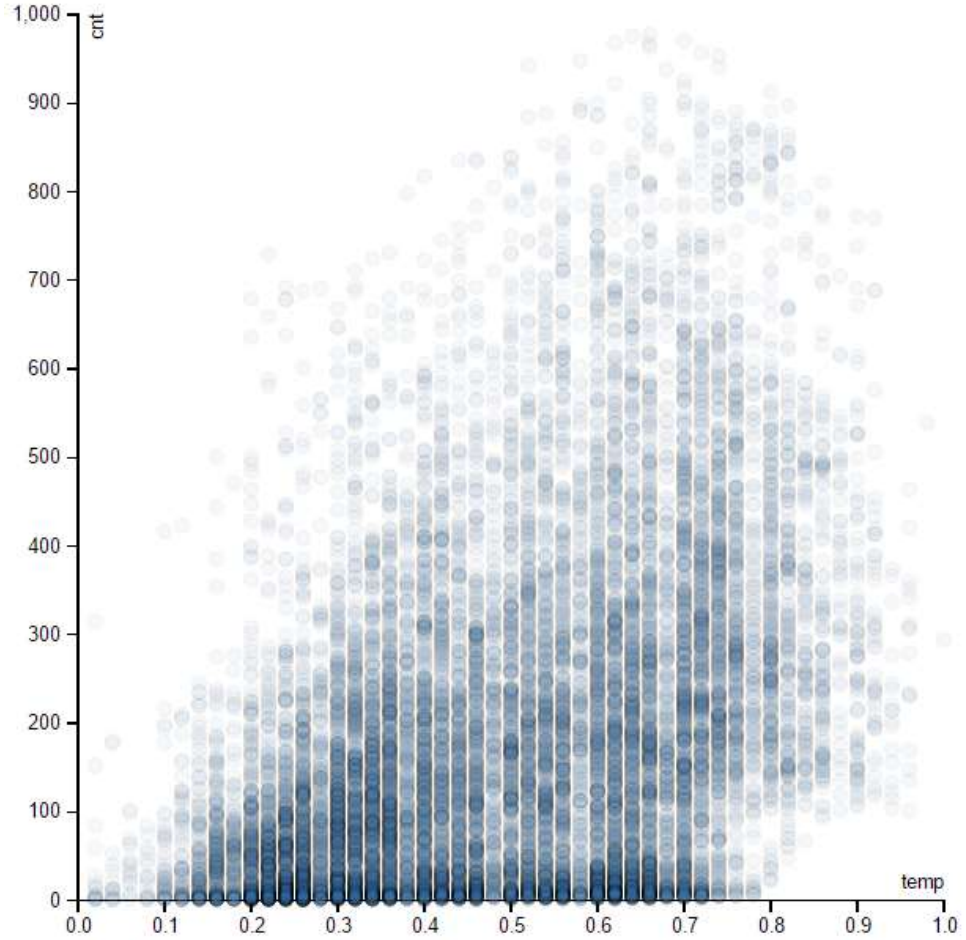




[Holten & van Wijk 2009]

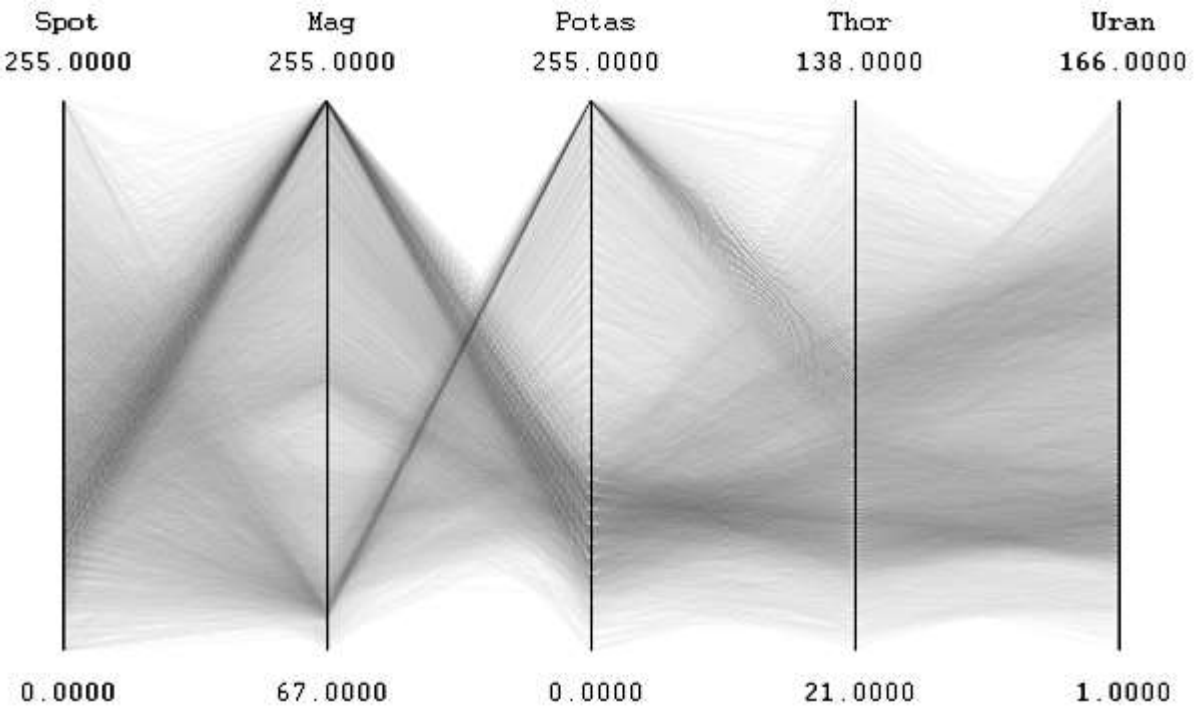
Alpha Blending

- Aggregation in image space
- Advantage:
  - Dense regions become visible
- Disadvantages:
  - Outliers may be omitted
  - Performance



Alpha Blending

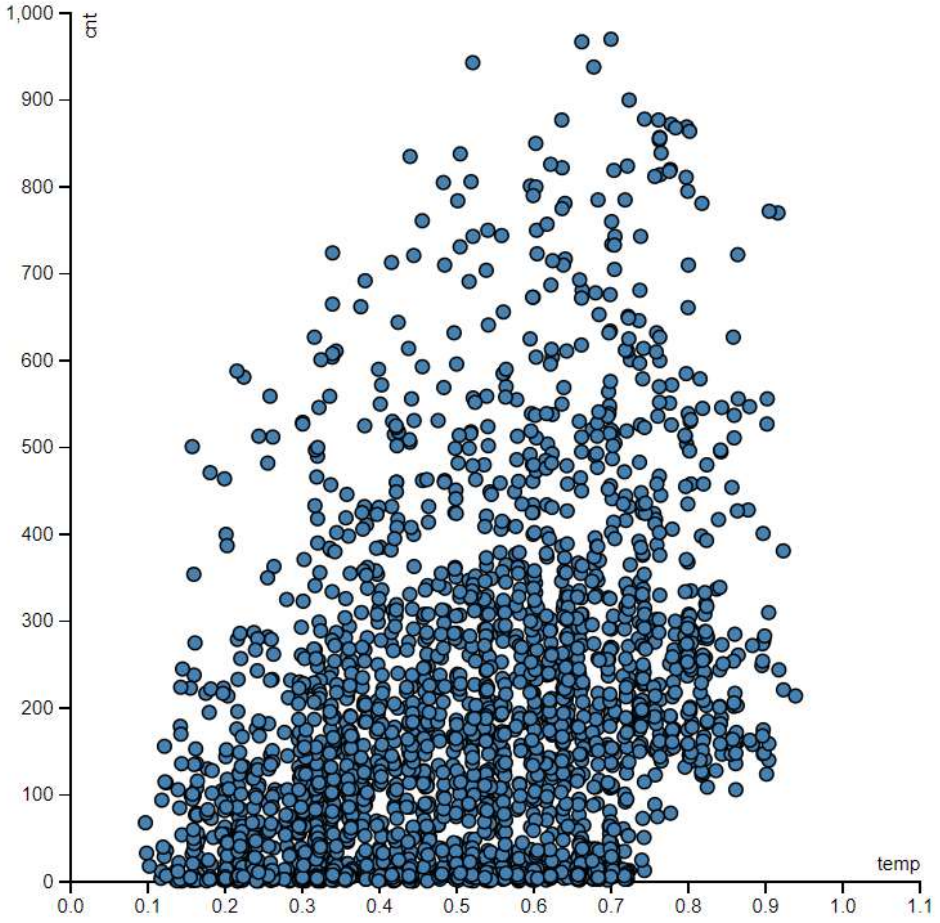
- Hundreds of thousands of semi-transparent lines



[Florek & Novotny, Interactive Information Visualization using Graphics Hardware, SCCG 2006]

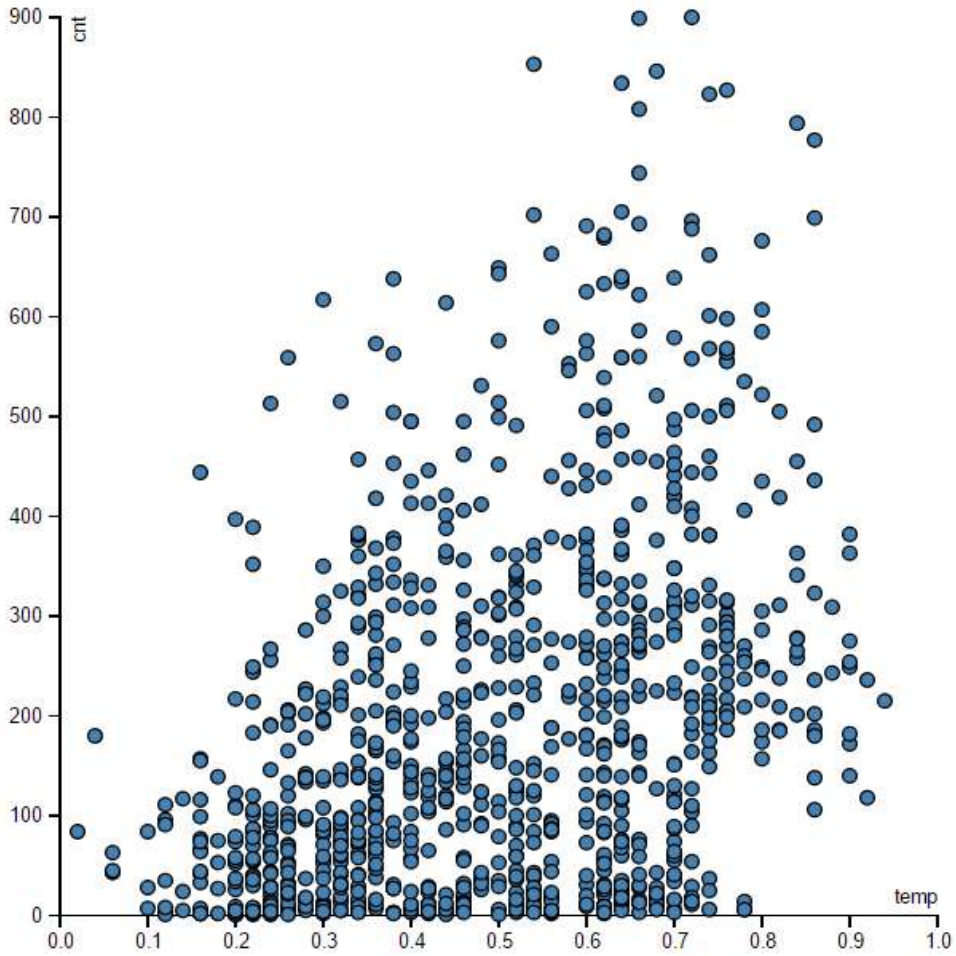
Filtering

- Selection of data subset that satisfies a user-specified criteria
- Ideal if the user has an idea of what might be ‘interesting’
- Example: Tuesday only



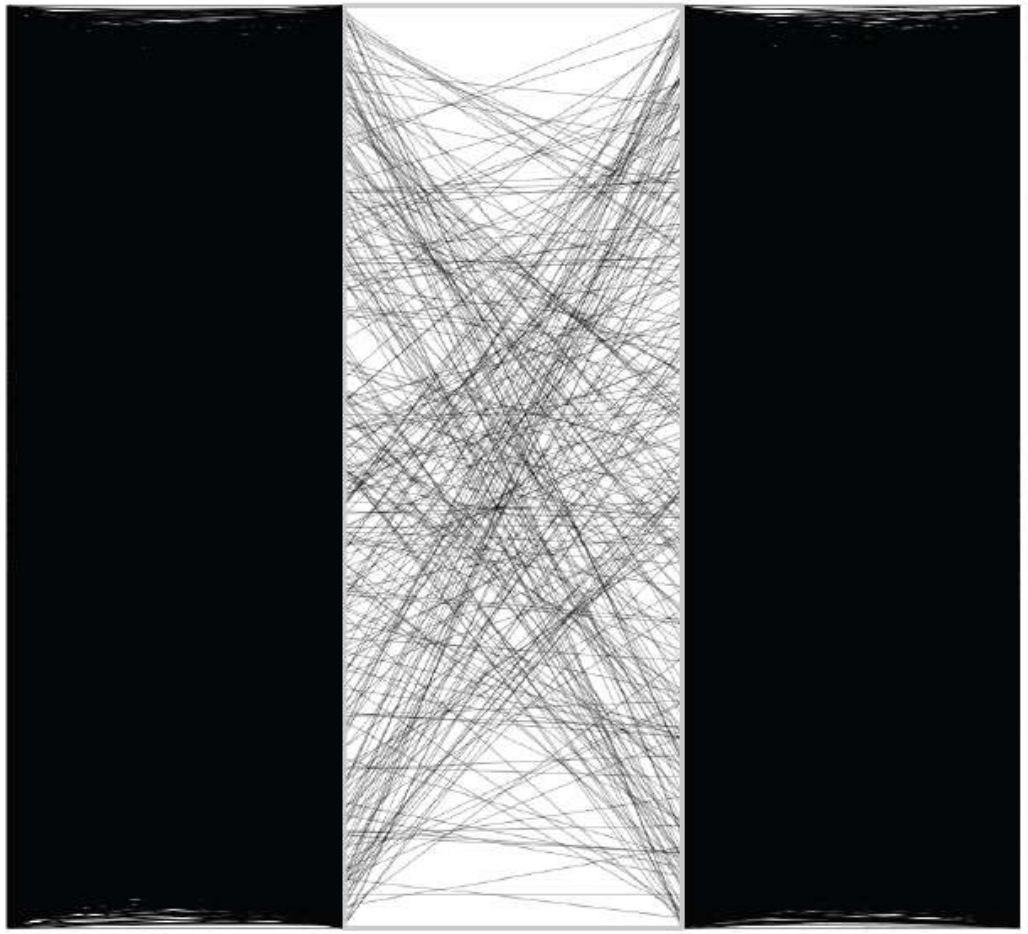
Sampling

- Random selection of subset of data
- Random sampling: every data point has same probability of being selected
- Problem: data outliers may be omitted
- Example: 1000 dots



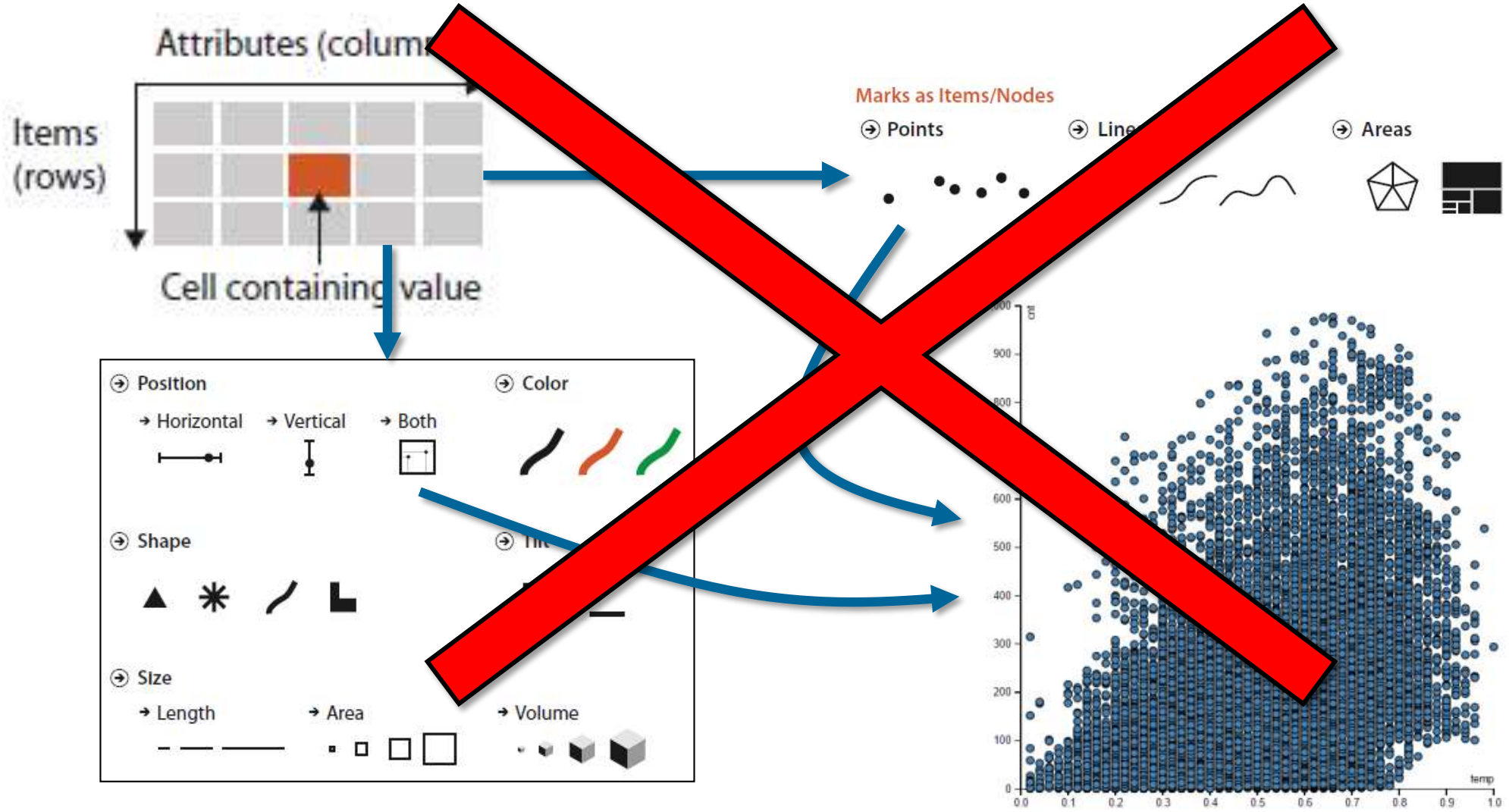
Sampling

- Parallel Coordinates: 1% of 30,000 items



[Ellis & Dix, Automatic Clutter Reduction in Parallel Coordinates, CHI-EA 2005]

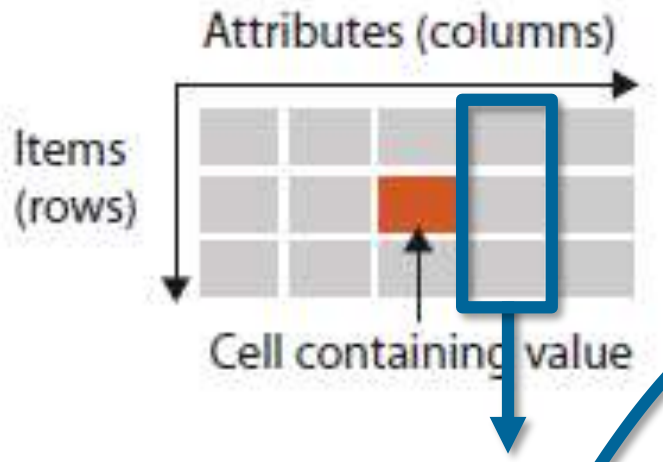
Data → Visual Structures



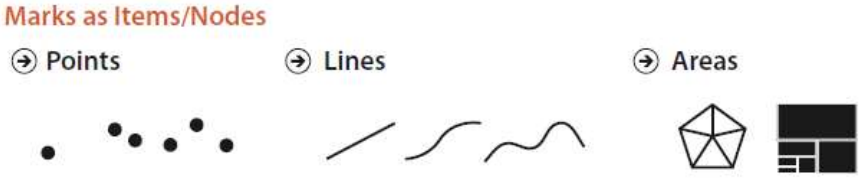
Scalable Visualizations

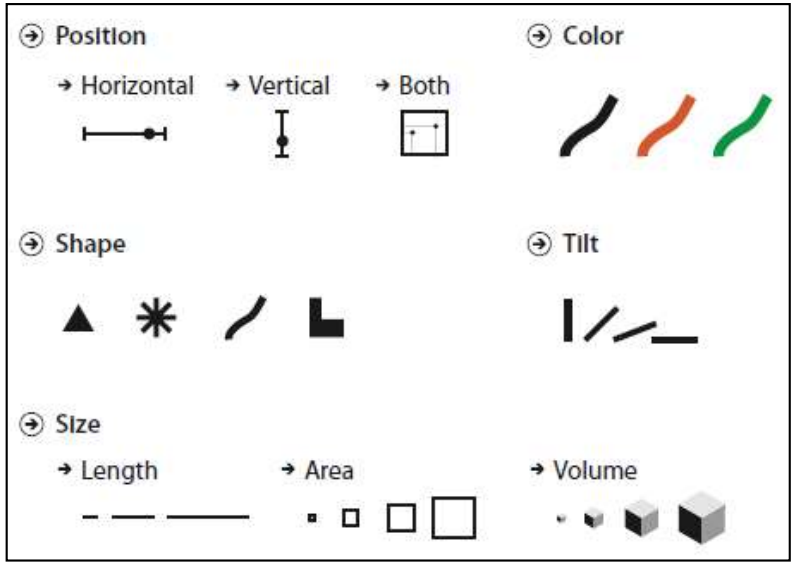
Visualize models or aggregates of the data instead of individual data points

Visualization of Summary Statistics



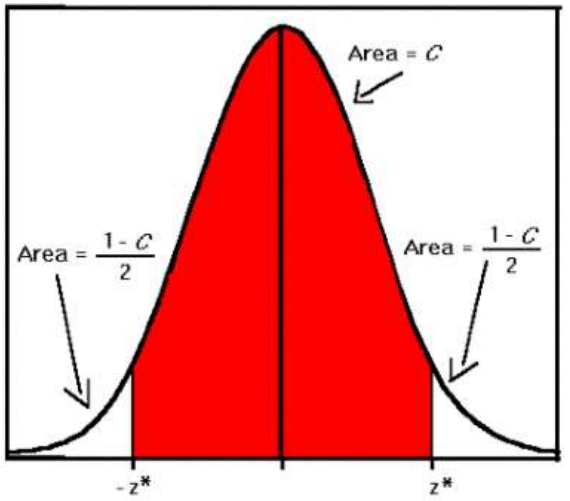
- Descriptive statistics
  - Mean
  - Median
  - Standard deviation
  - Quartiles
  - …





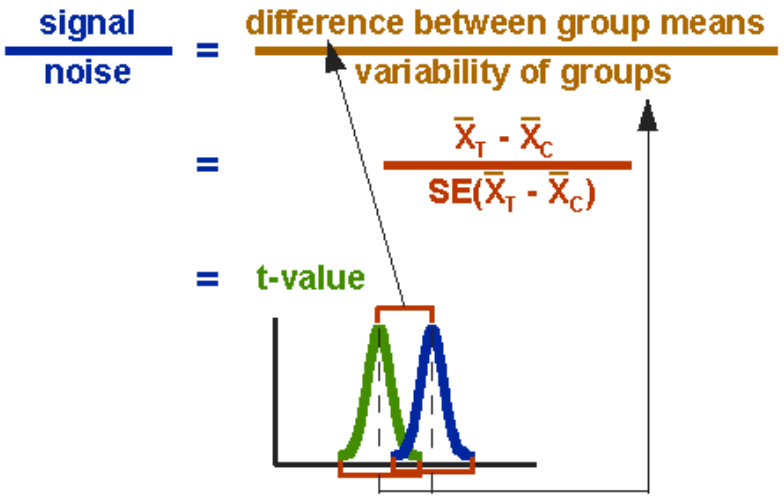
Summary Statistics

- Sample mean
  - $\bar{x} = \frac{\sum x_i}{n}$
- Sample standard deviation
  - $s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$
- Confidence interval
  - Range containing actual value with given probability
  - Normal distribution
  - Sample mean follows t distribution
  - $c = \bar{x} \pm t\frac{s}{\sqrt{n}}$



Summary Statistics

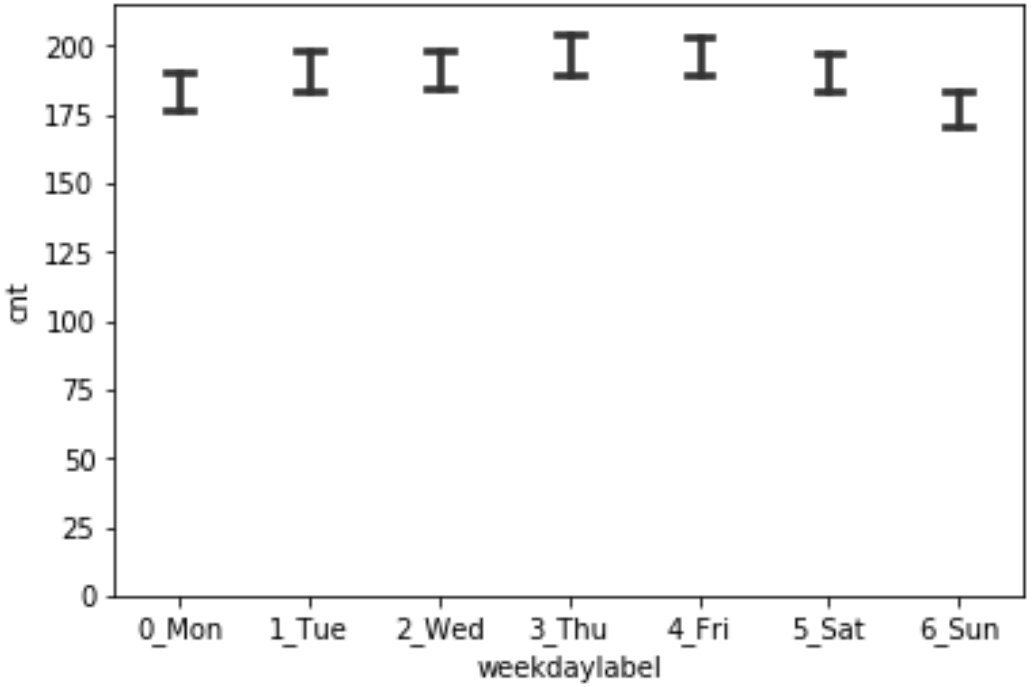
- Non-overlapping confidence intervals of two statistics
  → the two distributions are significantly different

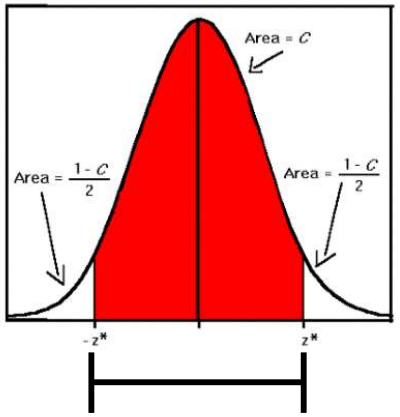


- Overlapping confidence intervals may still be different!

Model-Based Abstractions

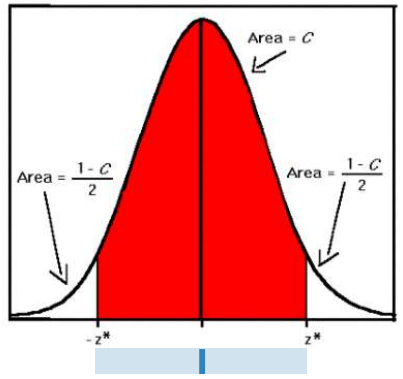
- Visualization of summary statistics
  - Confidence intervals
    - Mean → y position
    - Confidence intervals → length
  - Example: number of bike rentals per day of the week

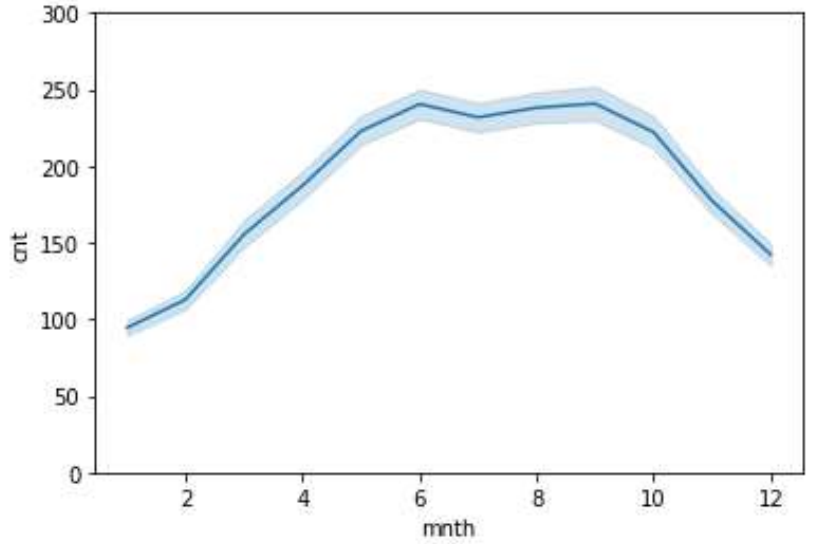


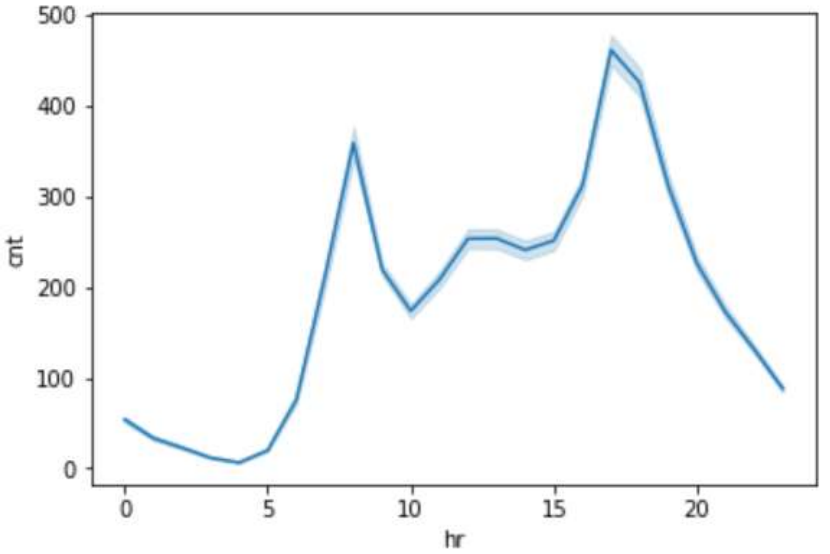


Model-Based Abstractions

- Visualization of summary statistics
  - Confidence intervals
    - Mean → y position
    - Confidence intervals → width
  - Example: number of bike rentals per month and per hour of the day

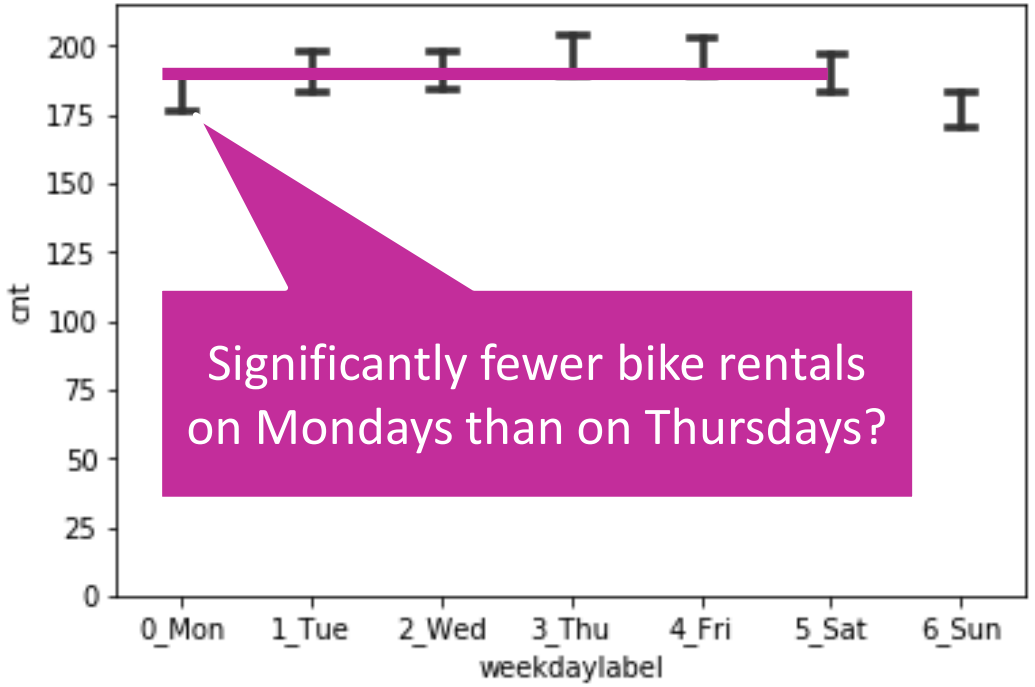






Model-Based Abstractions

- Visualization of summary statistics
  - Confidence intervals
    - Mean → y position
    - Confidence intervals → length
  - Example: number of bike rentals per day of the week



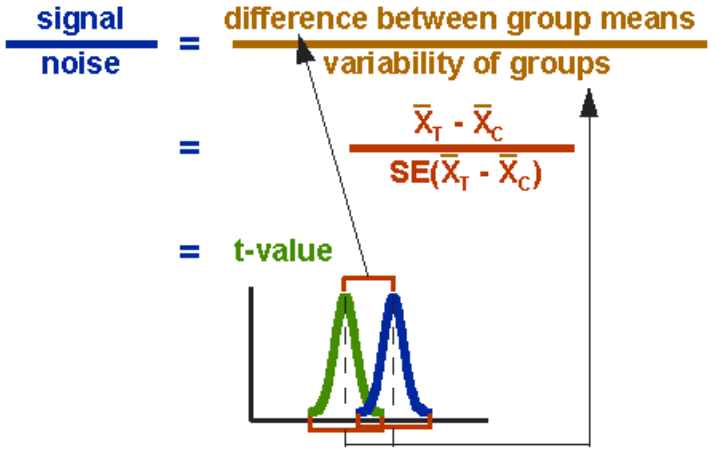
Problem I: Multiple Comparisons Problem

- Null hypothesis = there is no difference
- Null hypothesis is rejected if $p < \alpha$ (where typically $\alpha = .05$)

→ $p = .05$ means:
there is a 5% chance to get the observed results if the null hypothesis is true

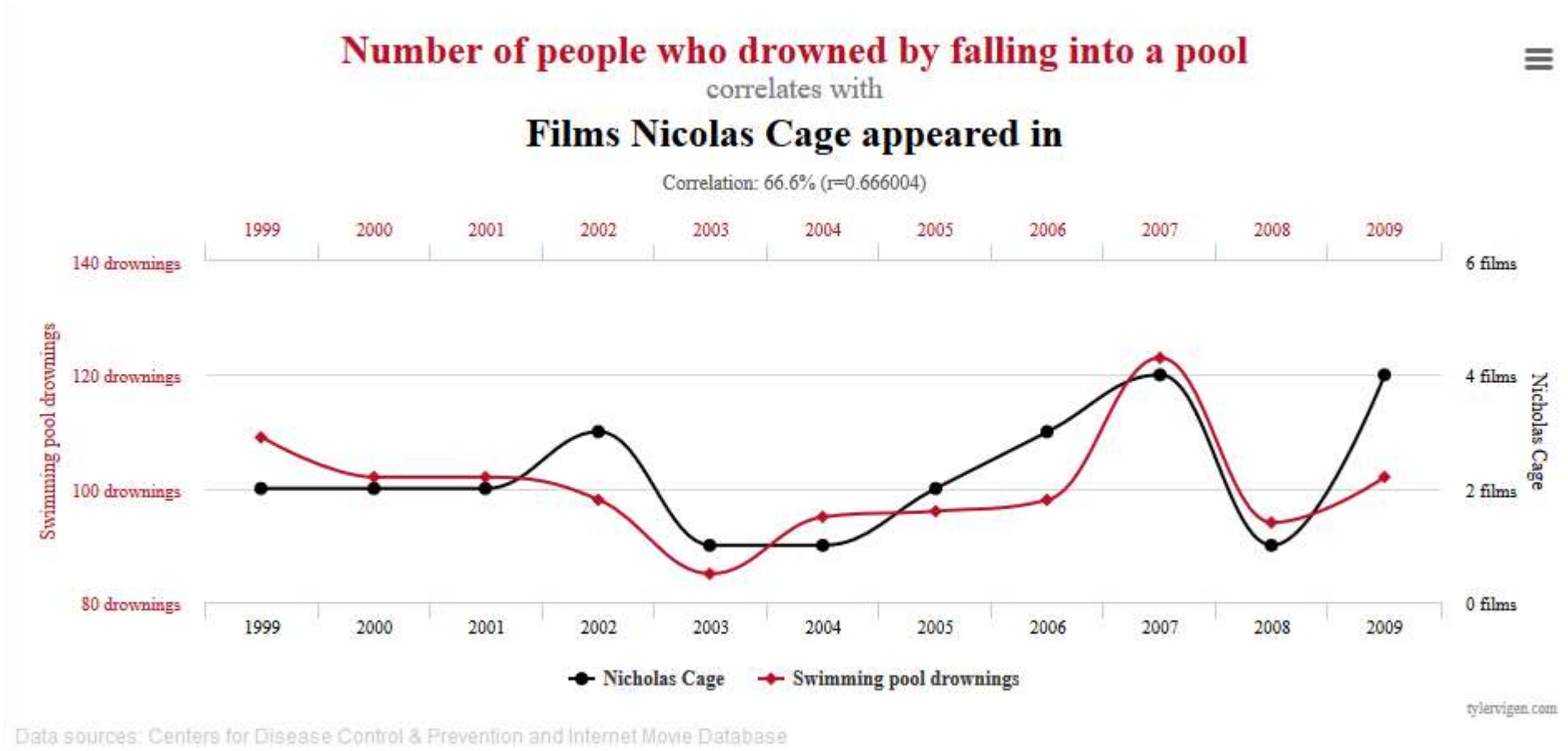
→ 5% of tests will show false positives

→ the more tests are conducted,
the more likely we will observe false positives!



http://www.socialresearchmethods.net/kb/stat_t.php

Example: Spurious Correlations



http://tylervigen.com/spurious-correlations

Adjustment for Multiple Comparisons

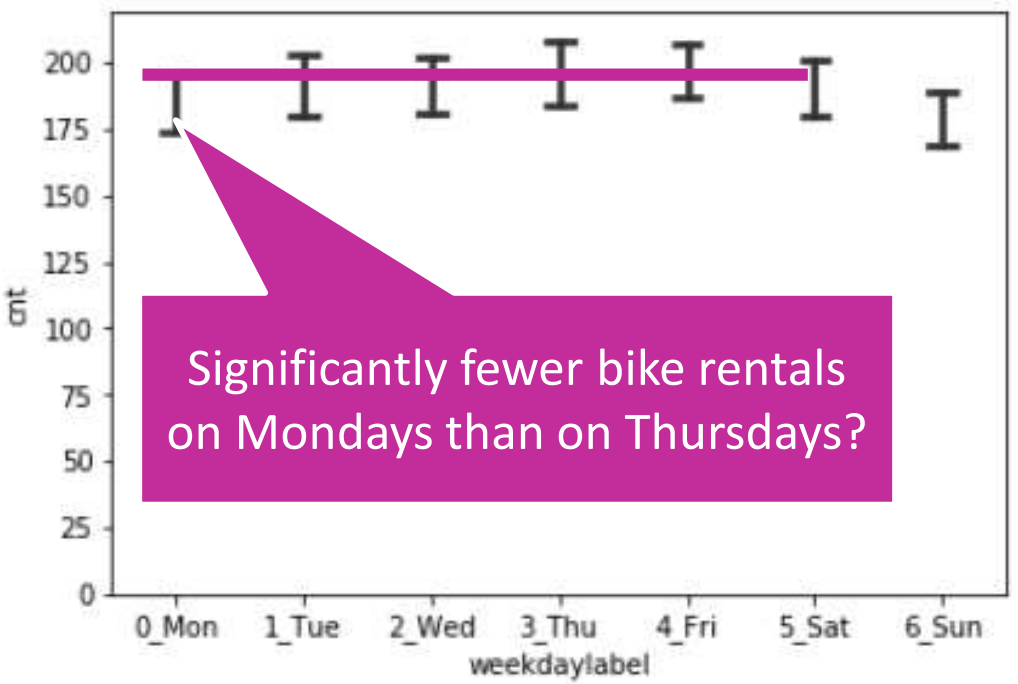
- Null hypothesis is rejected if $p < \alpha$ (where typically $\alpha = .05$)

- Adjust α to the number of comparisons:
  - Comparing 7 days: $m = \frac{n^2 - n}{2} = 21$ pair-wise comparisons
  - Bonferroni adjustment: $\alpha = \frac{0.05}{m} = 0.002$
  - Confidence interval: $c = \bar{x} \pm t \frac{s}{\sqrt{n}}$
    - t depends on α!

Degrees of freedom (ν)	Amount of area in one tail (α)							
	0.0005	0.001	0.005	0.010	0.025	0.050	0.100	0.200
1	636.6192	318.3088	63.65674	31.82052	12.70620	6.313752	3.077684	1.376382
2	31.59905	22.32712	9.924843	6.964557	4.302653	2.919986	1.885618	1.060660
3	12.92398	10.21453	5.840909	4.540703	3.182446	2.353363	1.637744	0.978472
4	8.610302	7.173182	4.604095	3.746947	2.776445	2.131847	1.533206	0.940965
5	6.868827	5.893430	4.032143	3.364930	2.570582	2.015048	1.475884	0.919544
6	5.958816	5.207626	3.707428	3.142668	2.446912	1.943180	1.439756	0.905703
7	5.407883	4.785290	3.499483	2.997952	2.364624	1.894579	1.414924	0.896030
8	5.041305	4.500791	3.355387	2.896459	2.306004	1.859548	1.396815	0.888890
9	4.780913	4.296806	3.249836	2.821438	2.262157	1.833113	1.383029	0.883404
10	4.586894	4.143700	3.169273	2.763769	2.228139	1.812461	1.372184	0.879058
11	4.436979	4.024701	3.105807	2.718079	2.200985	1.795885	1.363430	0.875530
12	4.317791	3.929633	3.054540	2.680998	2.178813	1.782288	1.356217	0.872609
13	4.220832	3.851982	3.012276	2.650309	2.160369	1.770933	1.350171	0.870152
14	4.140454	3.787390	2.976843	2.624494	2.144787	1.761310	1.345030	0.868055
15	4.072765	3.732834	2.946713	2.602480	2.131450	1.753050	1.340606	0.866245
16	4.014996	3.686155	2.920782	2.583487	2.119905	1.745884	1.336757	0.864667
17	3.965126	3.645767	2.898231	2.566934	2.109816	1.739607	1.333379	0.863279

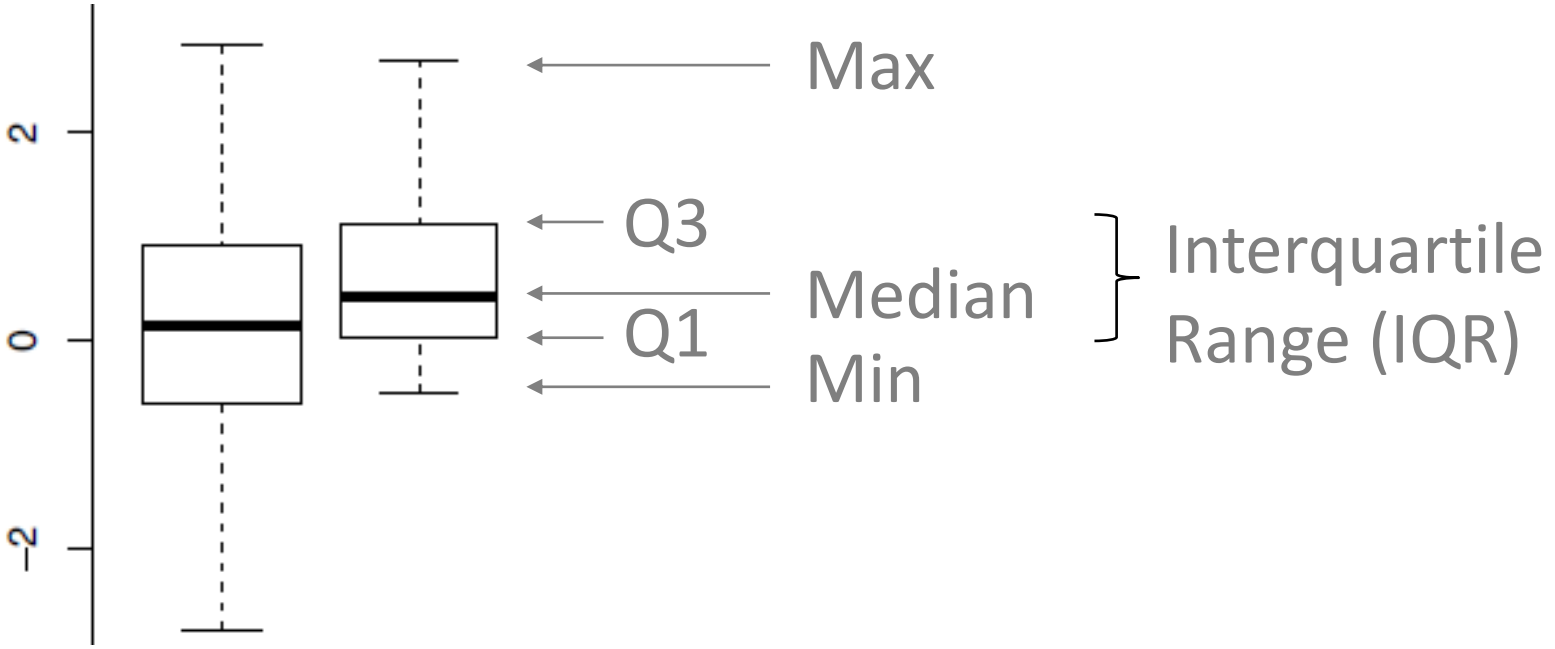
Model-Based Abstractions

- Visualization of summary statistics
  - Confidence intervals
    - Mean → y position
    - Confidence intervals → length
  - Example: number of bike rentals per day of the week with 99.8% confidence intervals



Model-Based Abstractions

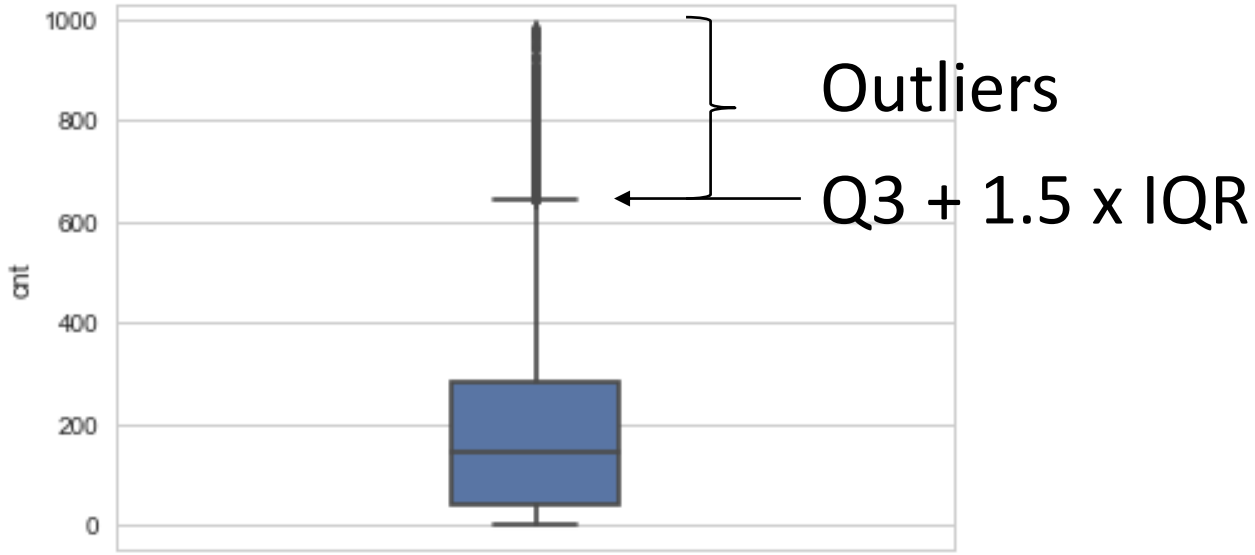
- Visualization of summary statistics:
  - Box plots



[Tukey, 1977]
[Munzner, 2014]
[Wickham and Stryjewski, 40 years of boxplots, 2012]

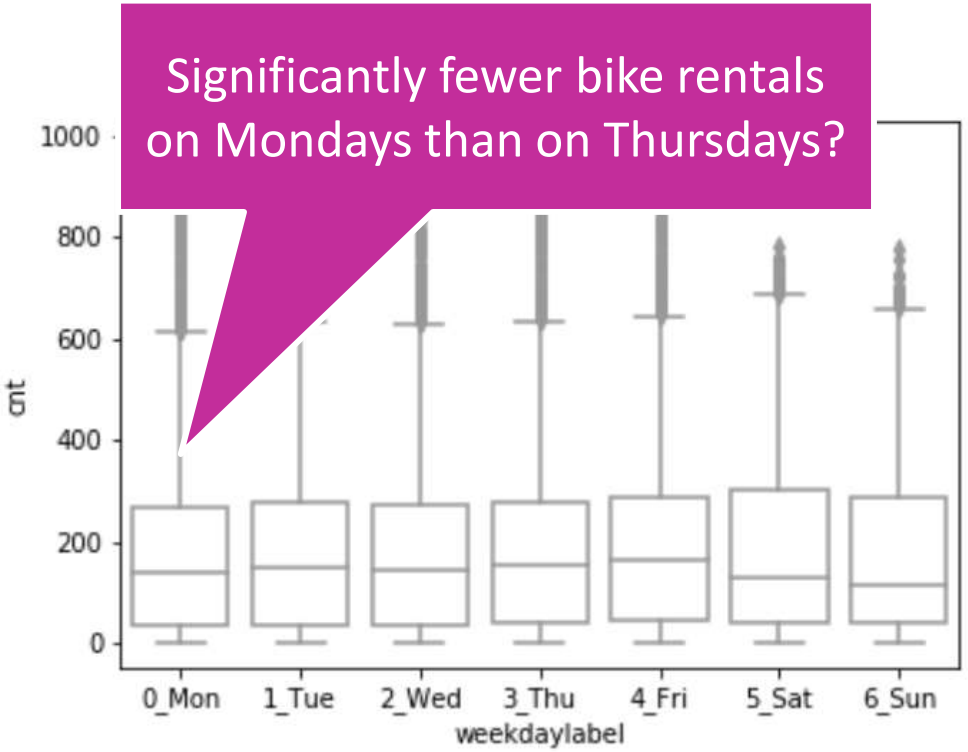
Model-Based Abstractions

- Visualization of summary statistics:
  - Box plots
  - Example: number of bike rentals



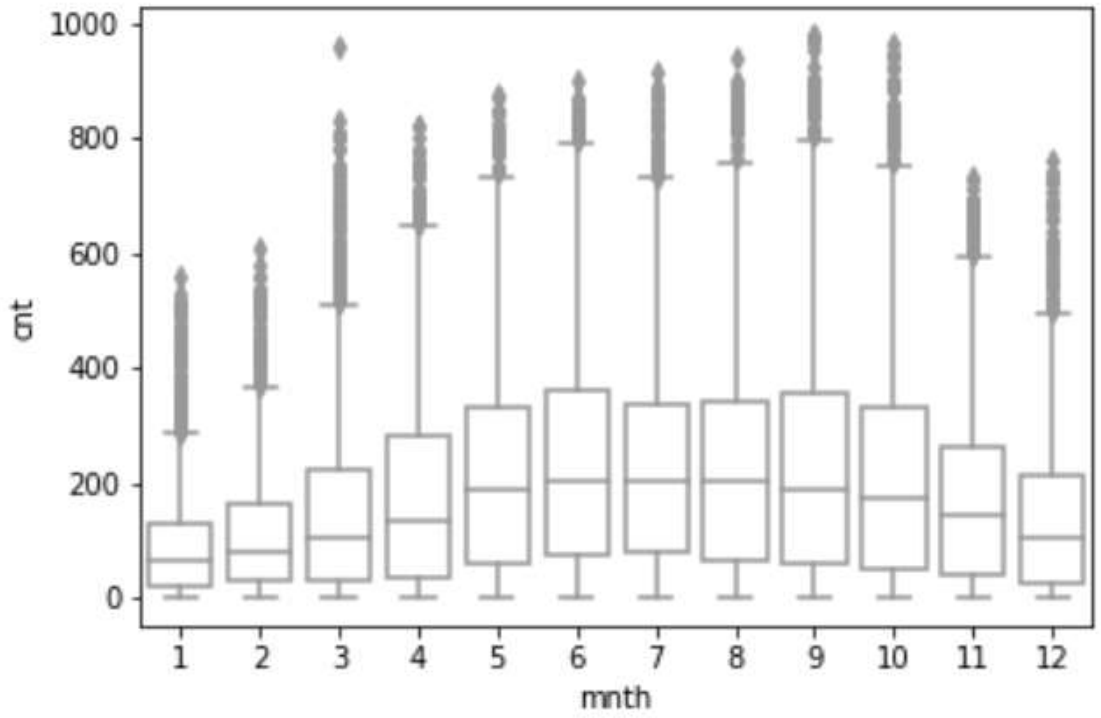
Model-Based Abstractions

- Visualization of summary statistics:
  - Box plots
  - Example: number of bike rentals per day of the week



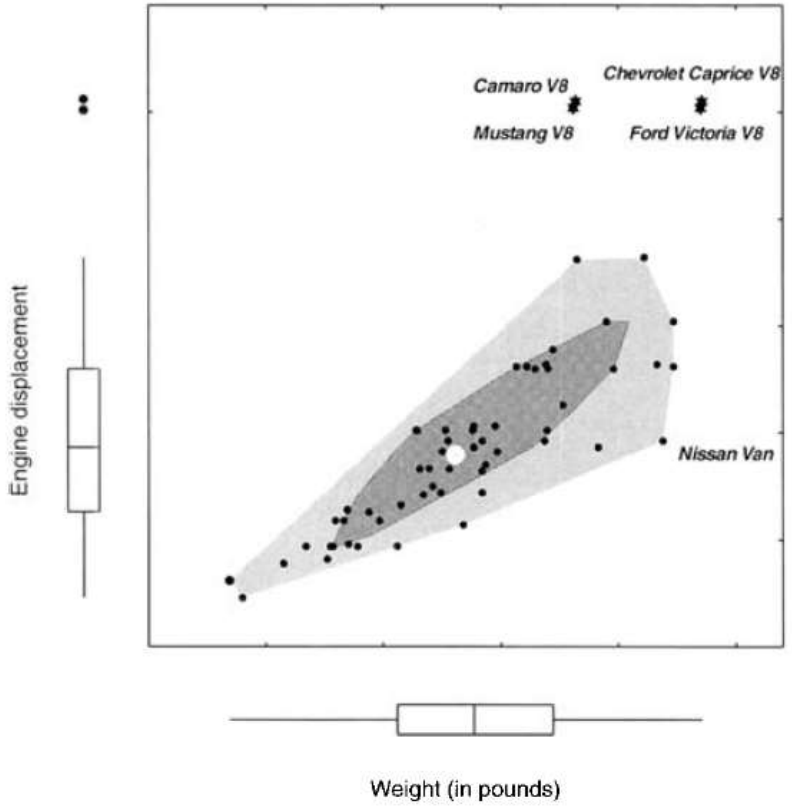
Model-Based Abstractions

- Visualization of summary statistics:
  - Box plots
  - Example: number of bike rentals per month



Model-Based Abstractions

- Visualization of summary statistics:
  - Bag plots



[Rousseeuw et al., The Bagplot: A Bivariate Boxplot, The American Statistician, 1999]

Aggregate Visualizations

- Visualize aggregates instead of individual data values
  - Data binning
    (original data → densities)
  - Transfer function (visual mapping)
    (densities → visual variables)
    - Area / Length / Position
    - Color / Intensity
    - Shape
- Also called density or binned visualizations

Recap: Attribute Types

- Nominal / Categorical
  - No implicit ordering
  - Can be same (apples) or different (apples and oranges)

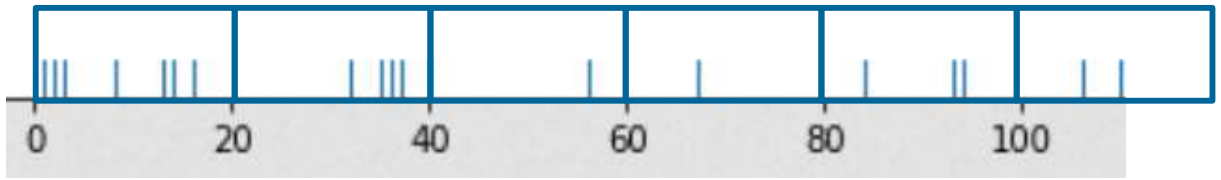
- Ordered
  - Data has implicit ordering
  - Ordinal
    - Well-defined ordering, rankings

  - Quantitative
    - Measurement of magnitude
    - Supports arithmetic comparison

[Munzner, 2014]

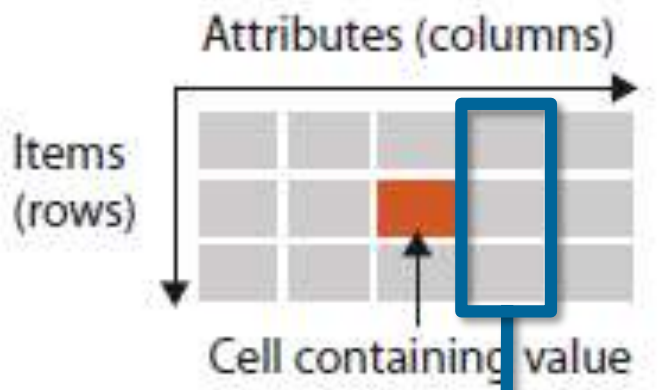
Binning

- Quantitative and ordinal attributes:
  - Separation of data range into regular intervals



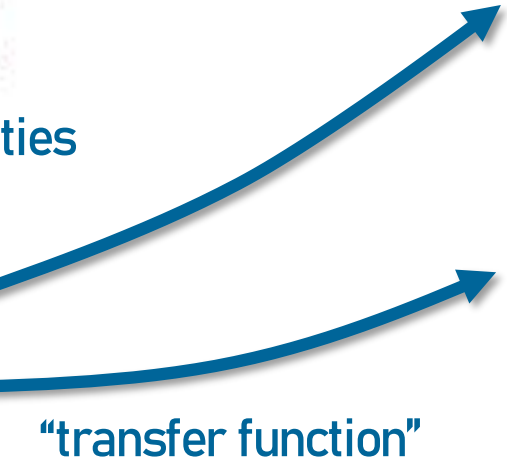
- Nominal / Categorical attributes:
  - Bins correspond to categories
  - Example: weekday
- Densities: number of items per bin
  - [0-20): 7, [20-40): 4, [40-60): 1, [60-80): 1, [80-100): 3, [100-120]: 2

Aggregate Visualizations

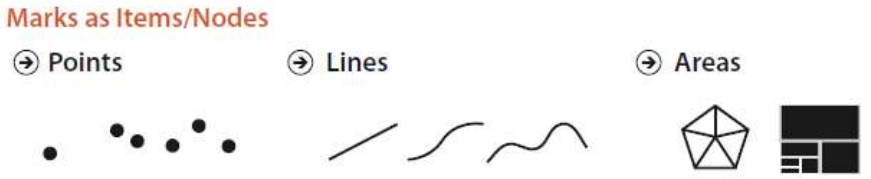


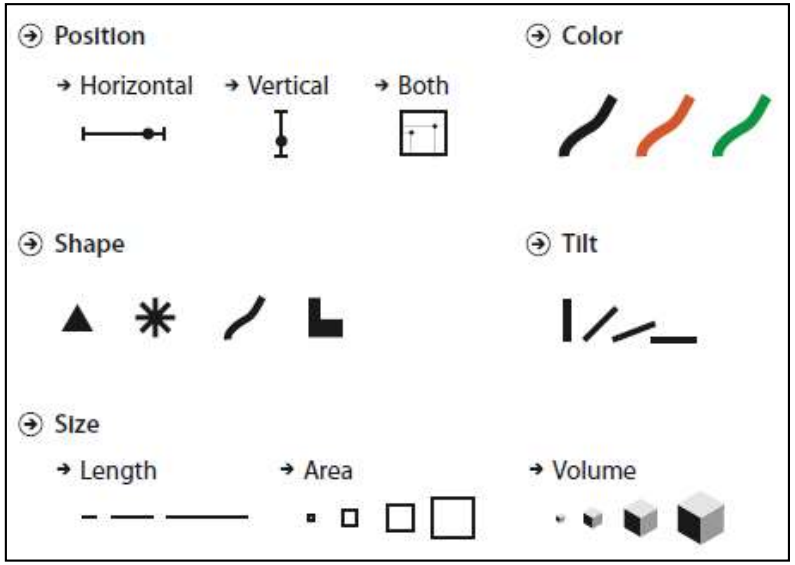
densities

- Bins
  - 1
  - 2
  - 3
  - 4
  - …



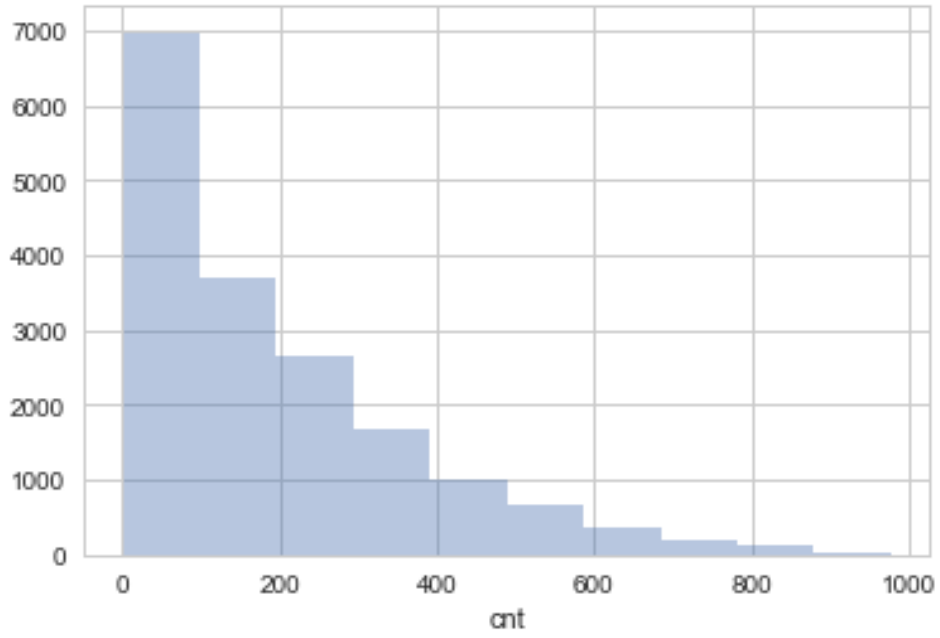
"transfer function"



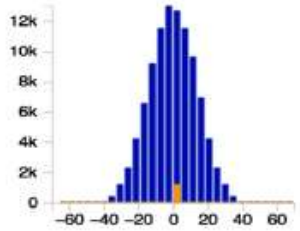
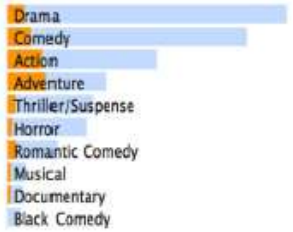
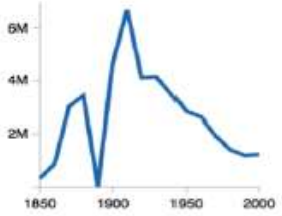
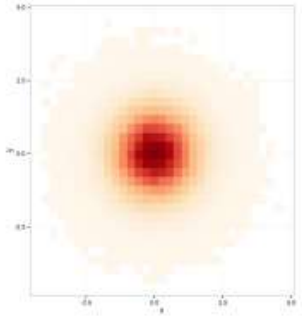
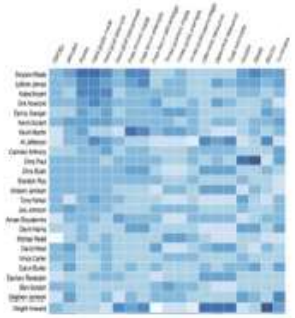
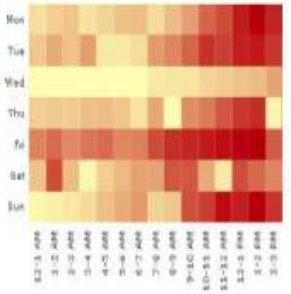


Histogram

- Split continuous data into bins with uniform width
- Height encodes number of data items in a bin (frequency)
- Examples: number of bike rentals per hour



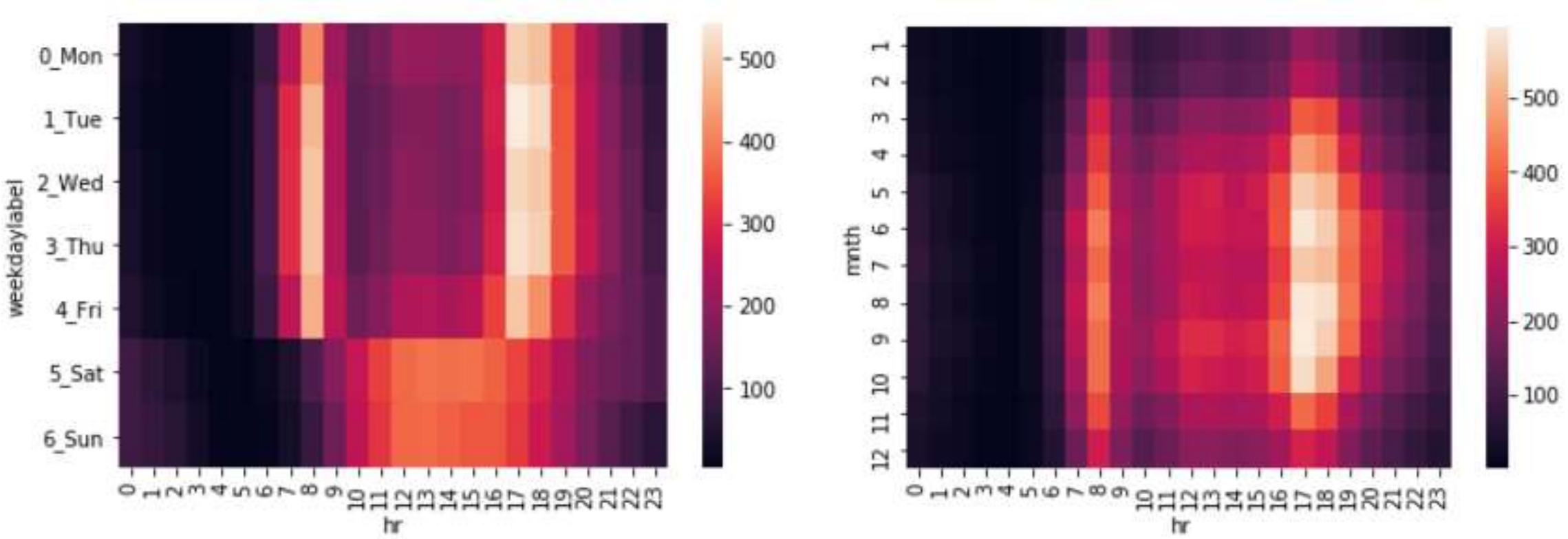
Density / Aggregate / Binned Visualizations

	Numeric	Ordinal	Temporal	Geographic
1D	 Histogram	 Bar Chart	 Line Graph / Area Chart	 Choropleth Map
2D	 Binned Scatter Plot	 Heatmap	 Temporal Heatmap	 Geographic Heatmap

[Lui et al., EuroVis 2013]

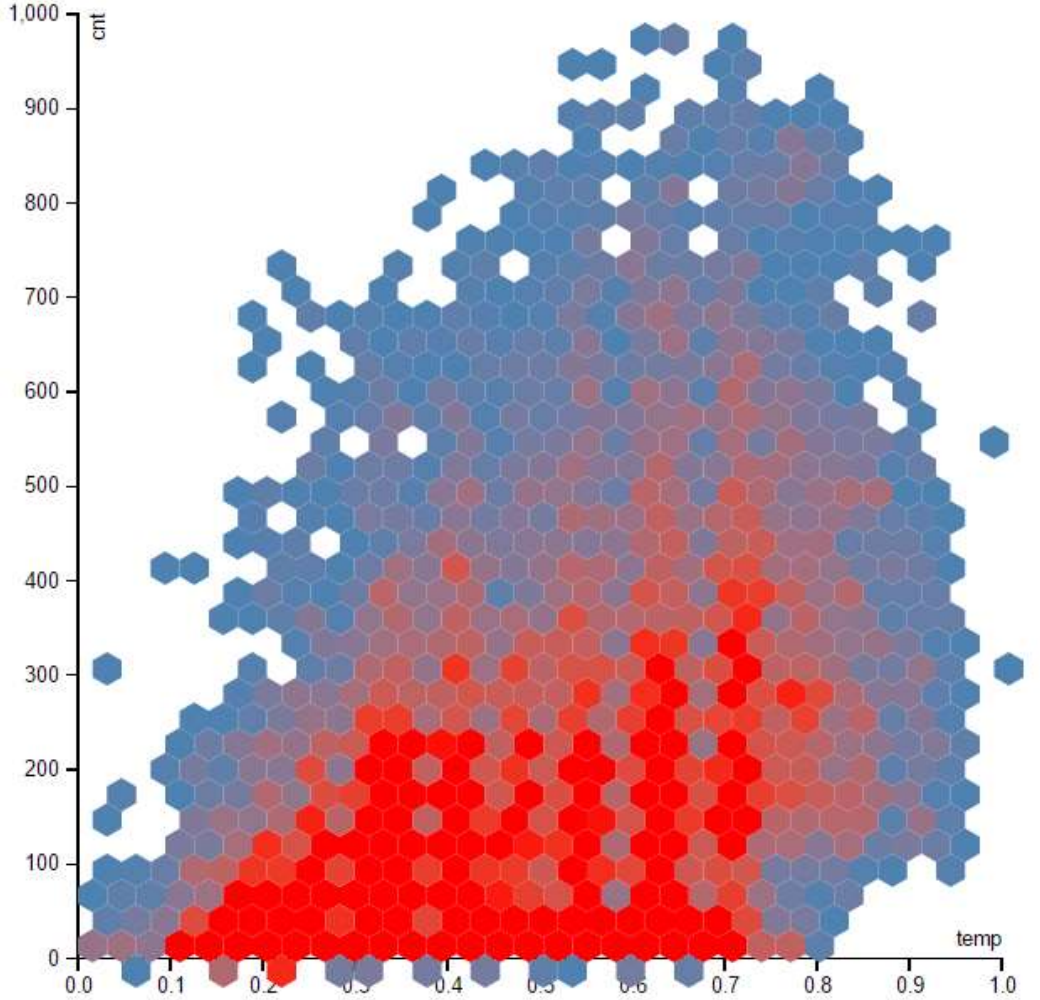
Temporal Heatmaps

- Examples: number of bike rentals per weekday / month and hour of the day



Hexagon Binning

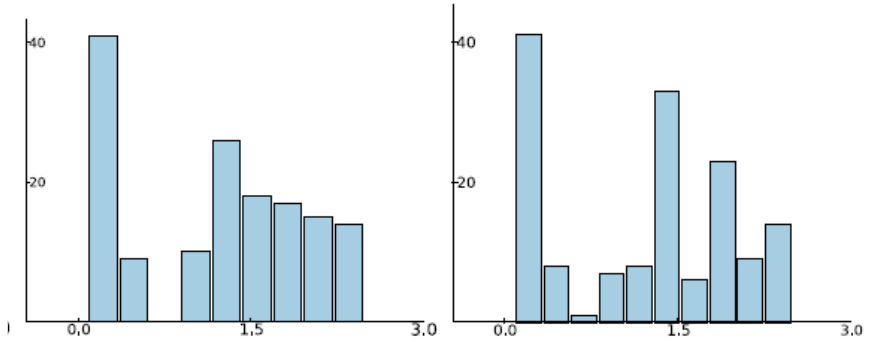
- Binning into hexagons
- Number of points per hexagon shown as color or area



[Carr et al., Scatterplot Matrix for Large N, Journal of American Statistical Association, 1987]

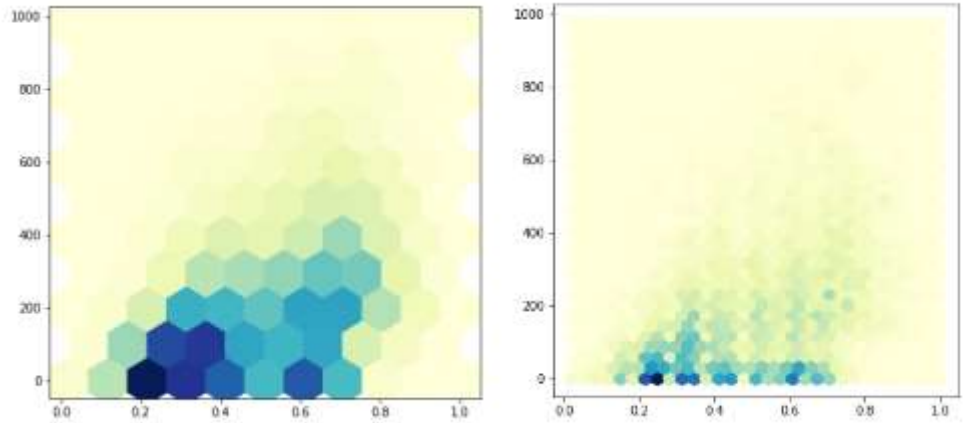
Histogram

- Can cause aliasing effects depending on bin size



9 bins 10 bins

[Lampe and Hauser, Interactive Visualization of Streaming Data with Kernel Density Estimation, PacificVis 2011]



10x10 bins 30x30 bins

Kernel Density Estimation

- Non-parametric estimation of probability density function (pdf) of random variable
- Weighting distances of observations $(x_1, x_2, \dots, x_n)$ from point x

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

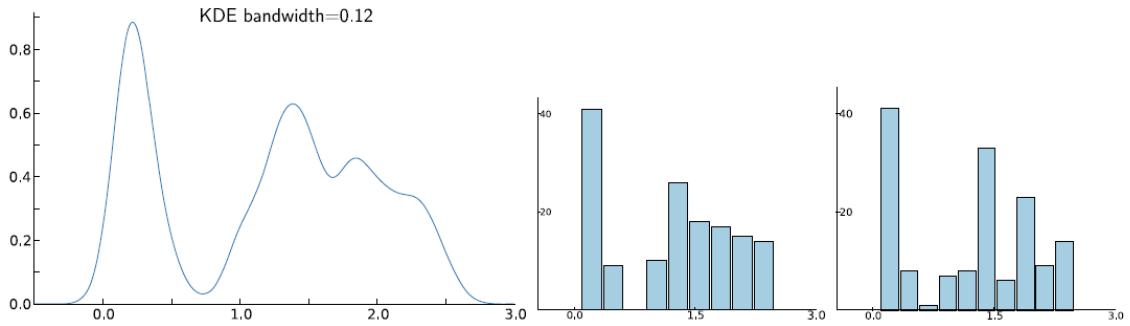
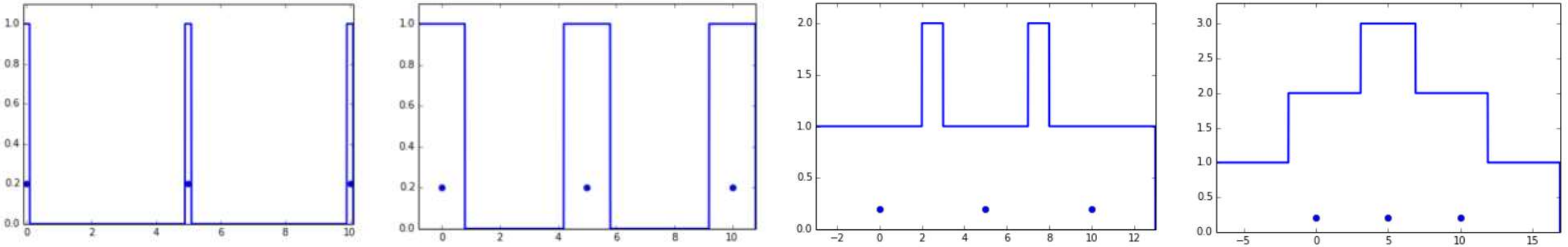


Figure 2: A kernel density estimation of *petal width* in the Iris dataset [13] and two corresponding histograms, one with 9 bins and the other with 10 bins.

[Lampe and Hauser, Interactive Visualization of Streaming Data with Kernel Density Estimation, PacificVis 2011]

Kernel Density Estimation

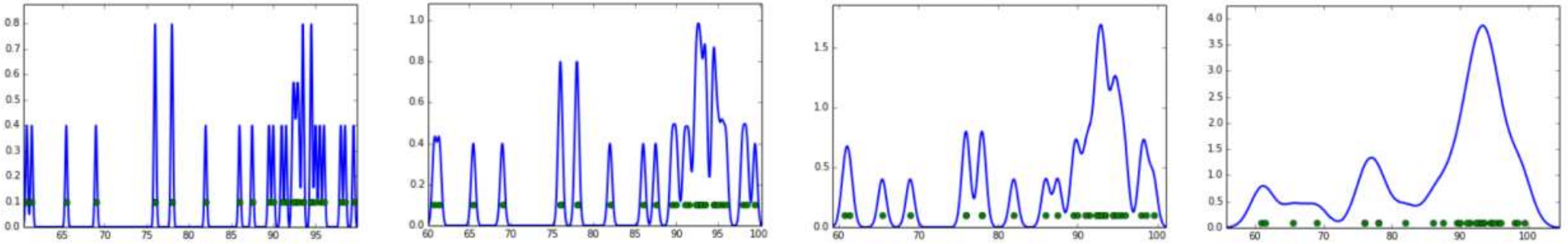
- Interactive demo: http://www.mglerner.com/blog/?p=28
- Rectangular kernel:



Kernel width

Kernel Density Estimation

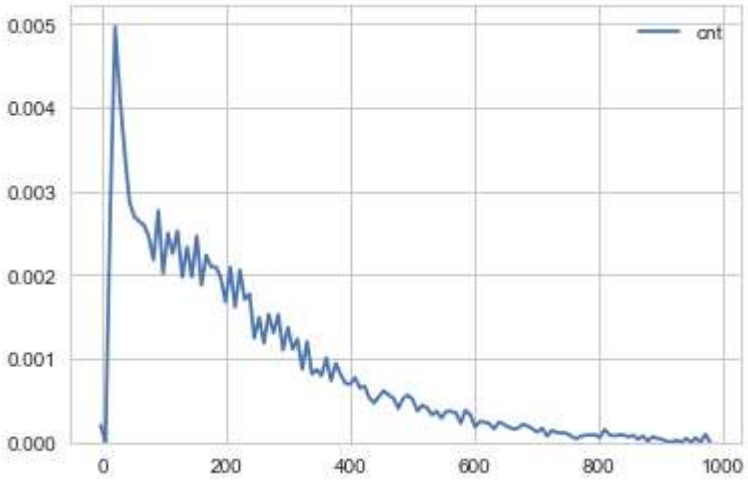
- Interactive demo: http://www.mglerner.com/blog/?p=28
- Gaussian kernel:

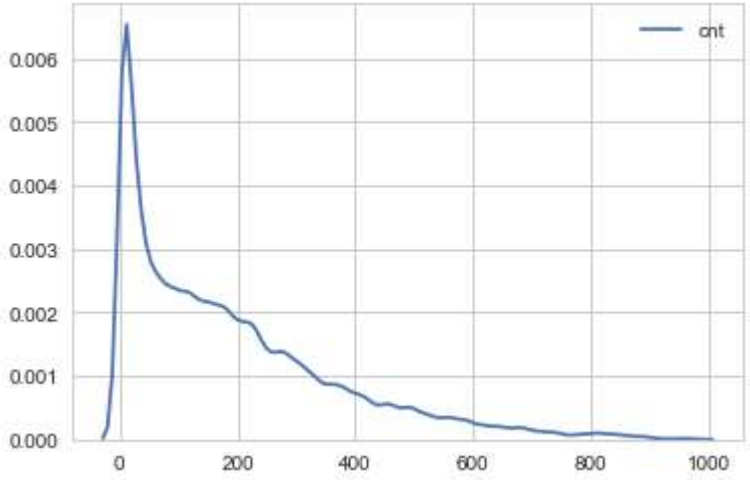


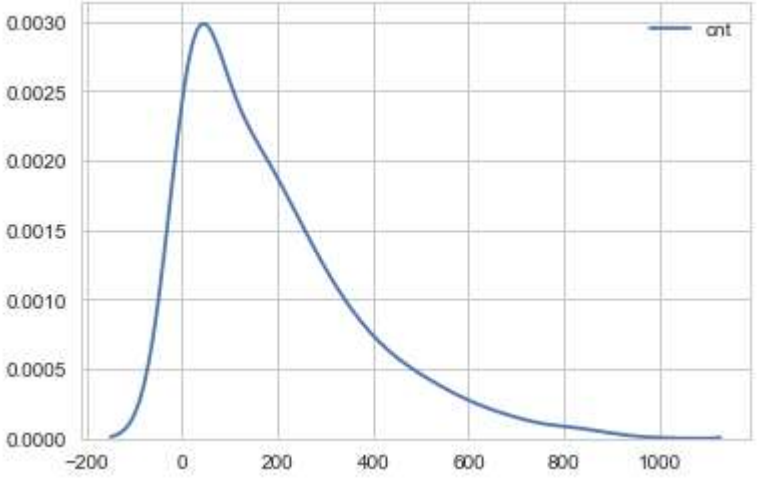
Kernel width

Kernel Density Estimation

- Bandwidth
  - Examples: bandwidth = 1, 10, 50

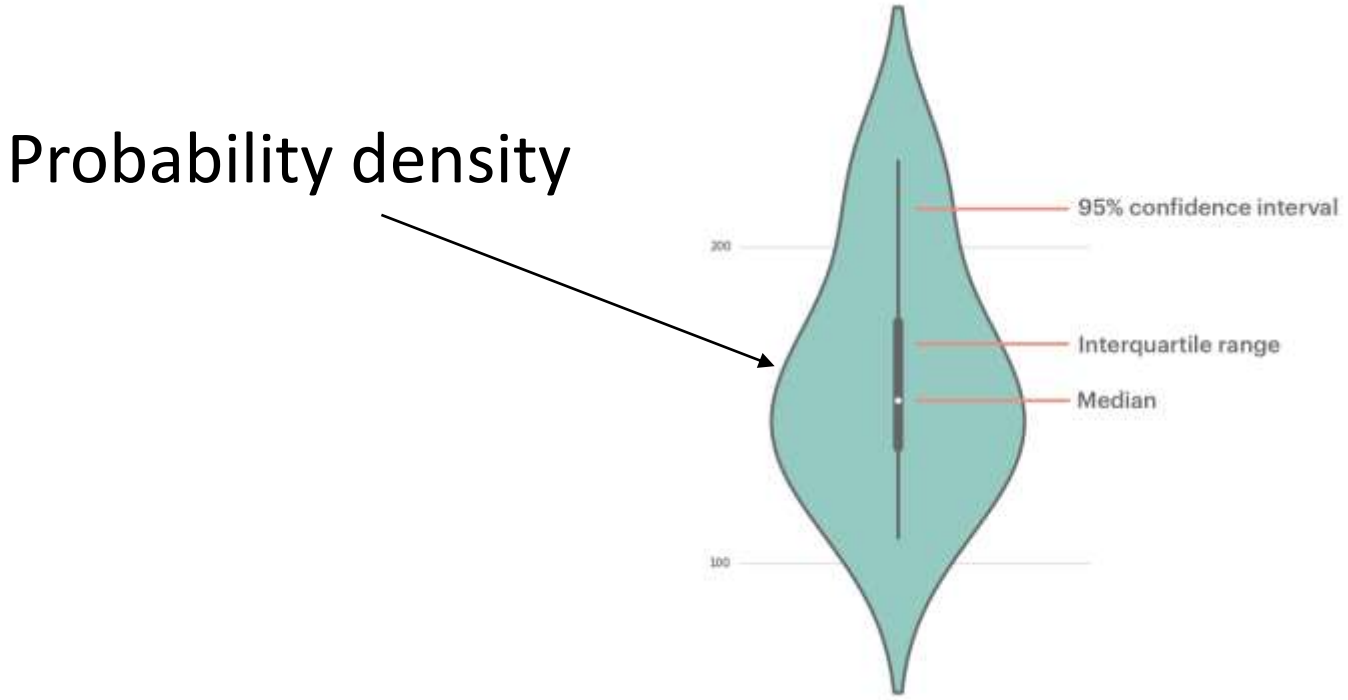






Kernel Density Estimation

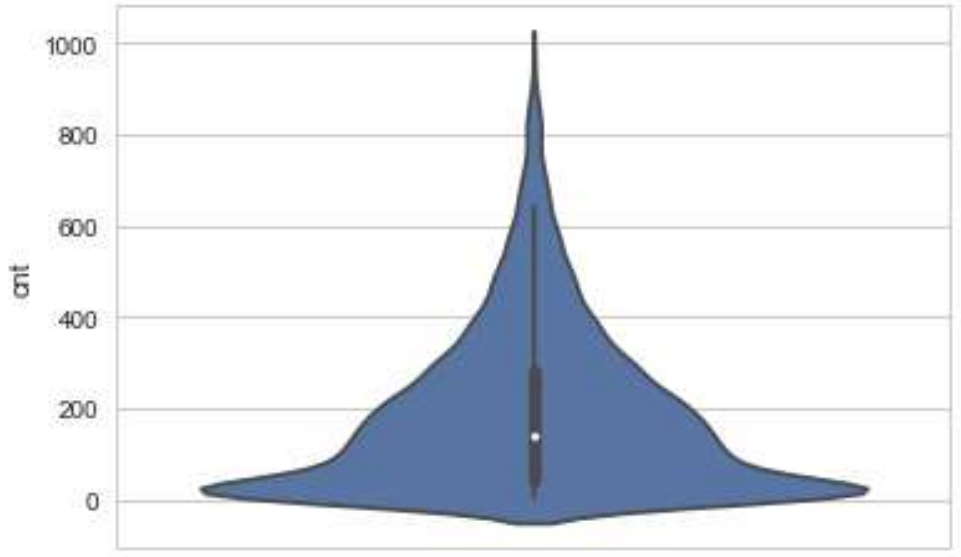
- Visualization of summary statistics & probability density:
  - Violin plots



https://blog.modeanalytics.com/violin-plot-examples/

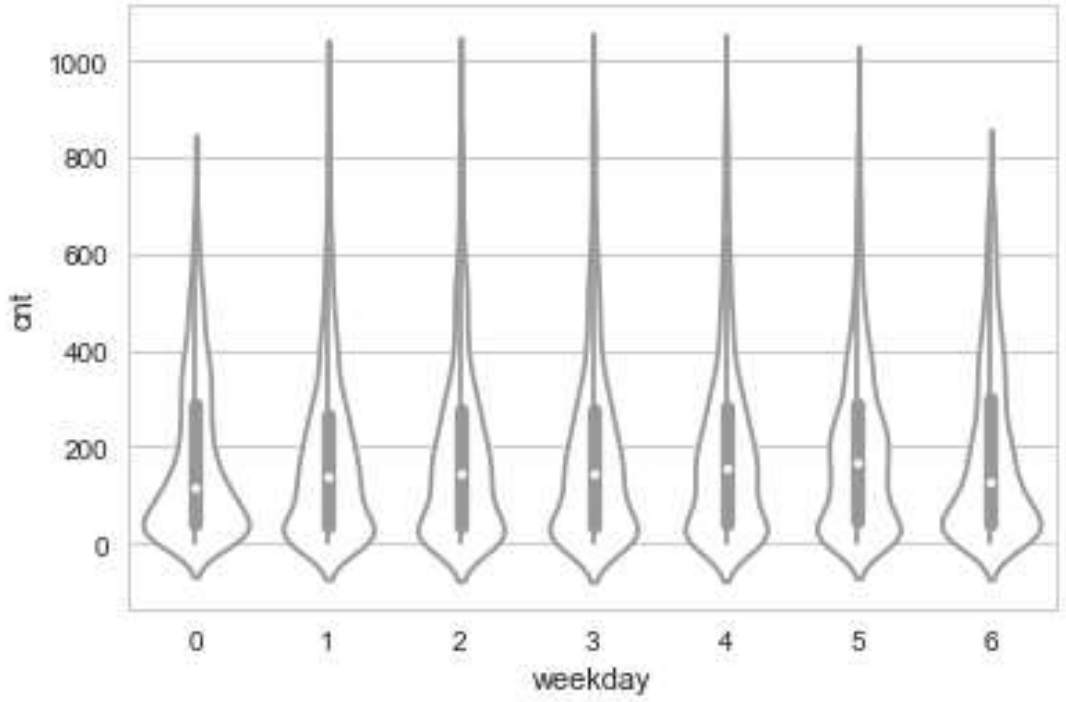
Kernel Density Estimation

- Visualization of summary statistics & probability density:
  - Violin plots
  - Example: number of bike rentals



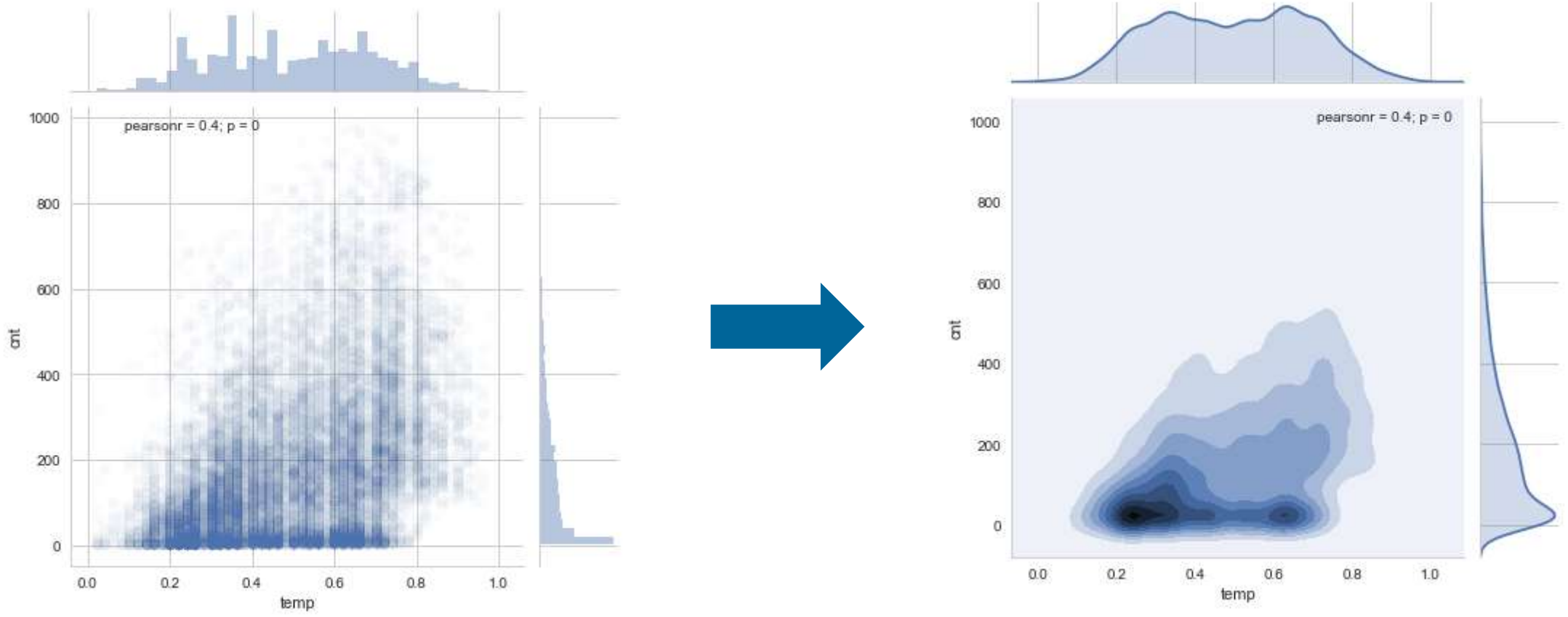
Kernel Density Estimation

- Visualization of summary statistics & probability density:
  - Violin plots
  - Example: number of bike rentals per day of the week



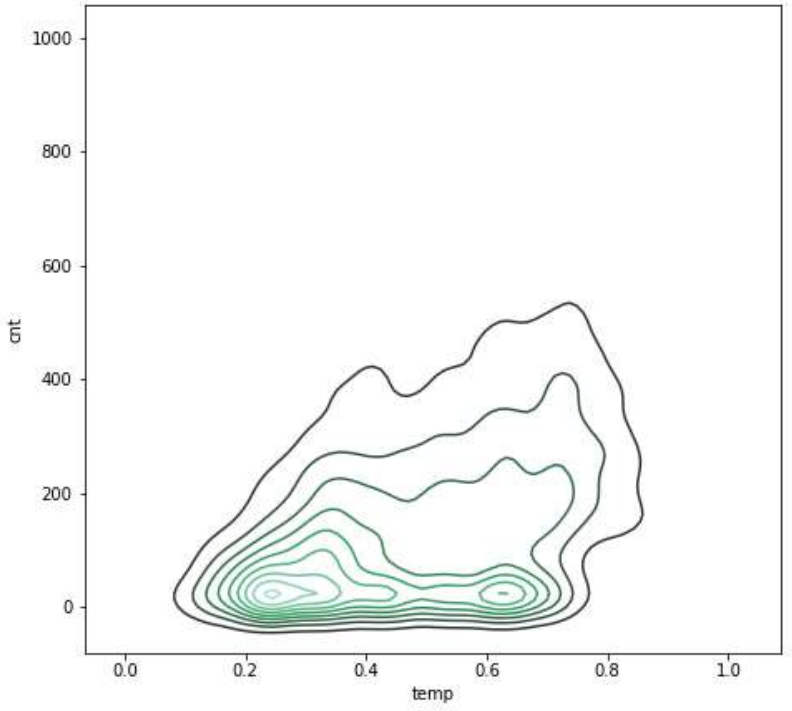
Kernel Density Estimation

- KDE scatterplot: 2D kernel density estimation



Kernel Density Estimation

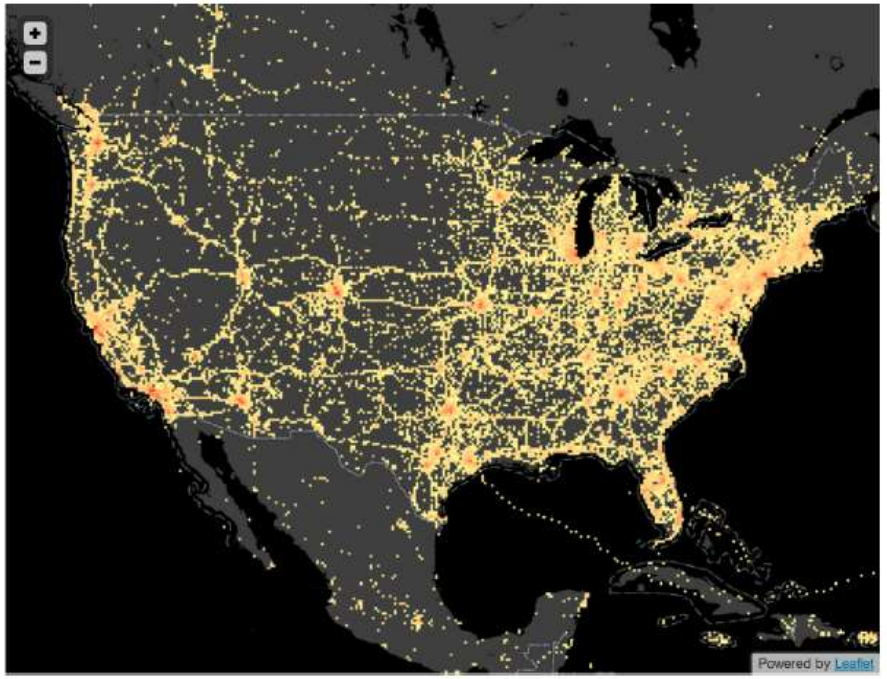
- Contour plot
  - Contours are isoline boundaries of constant densities
  - Same encoding principle as terrain maps



Bergfex

Comparison Sampling - Heatmap

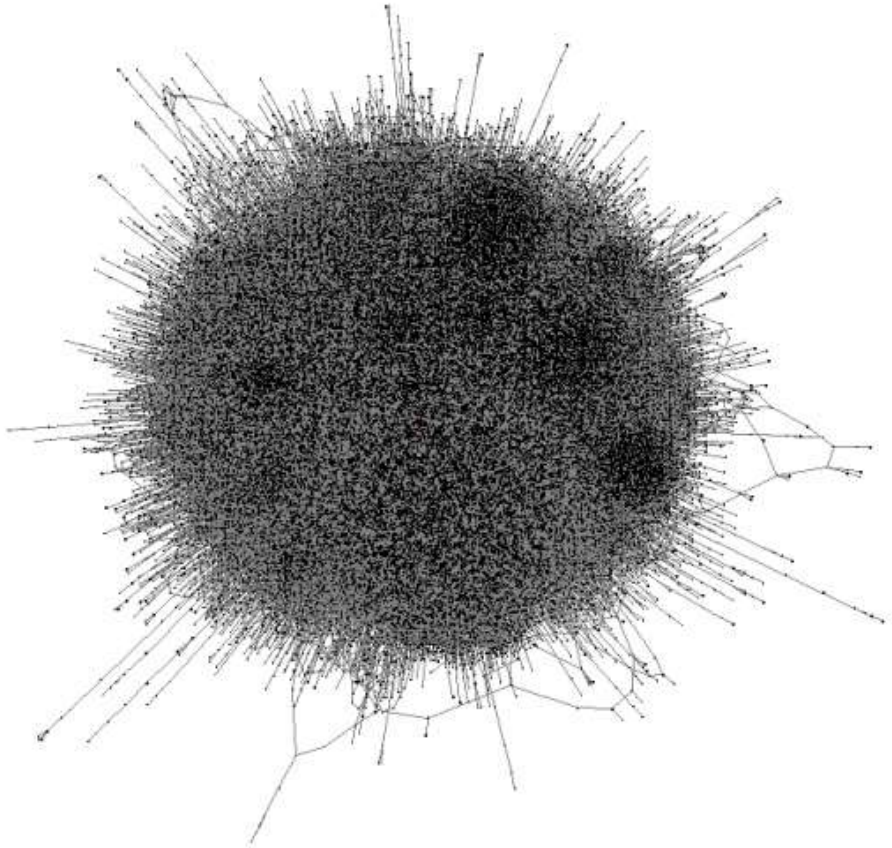
- Local overplotting
- Missing outliers
- Global patterns
- Local outliers



[Lui, Jiang & heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013]

Aggregate Visualization of Large Graphs

- Aggregation of
  - Edges
  - Nodes

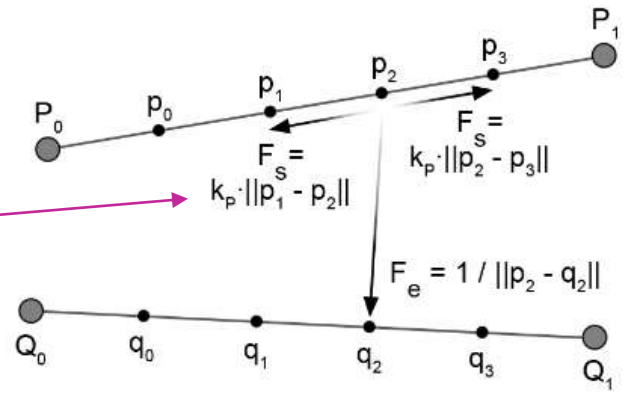


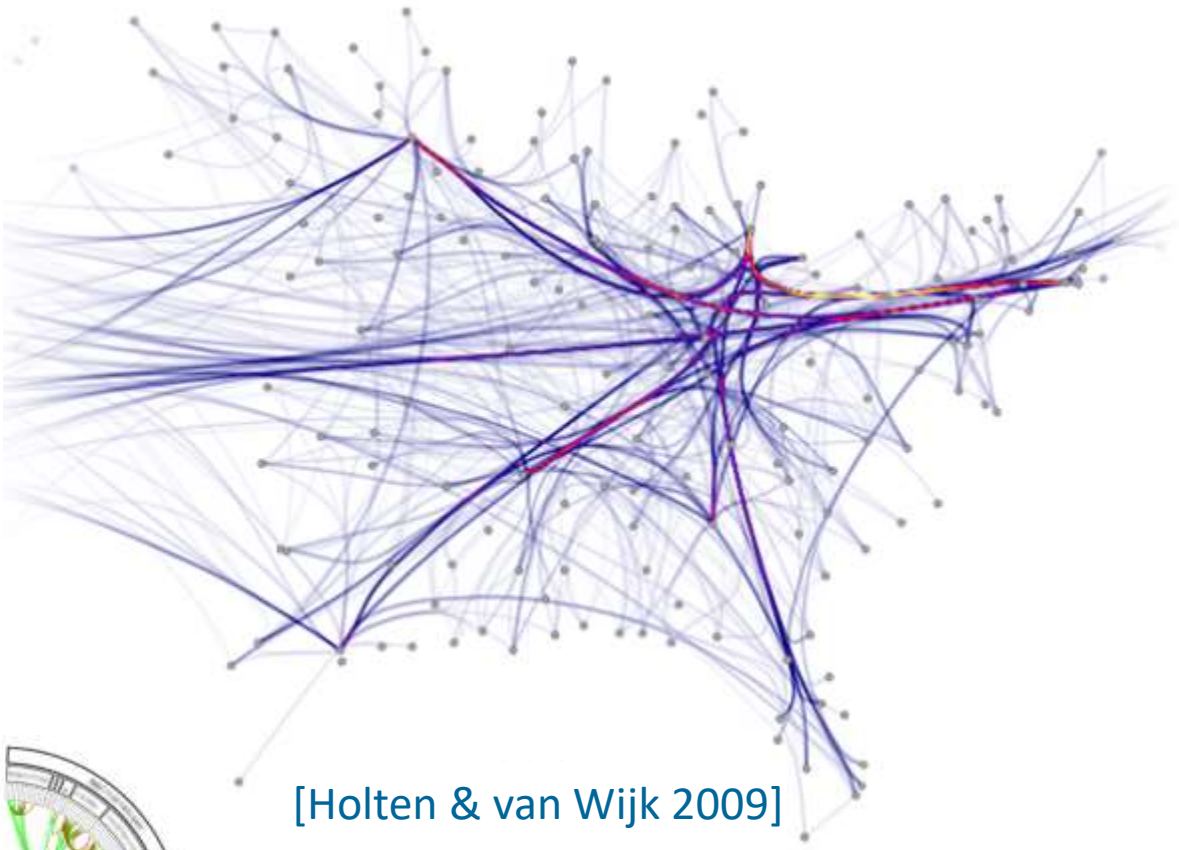
https://linkurio.us/blog/network-management-and-impact-analysis-with-neo4j/

Edge Bundling

- Aggregating edges based on:
  - Forces:

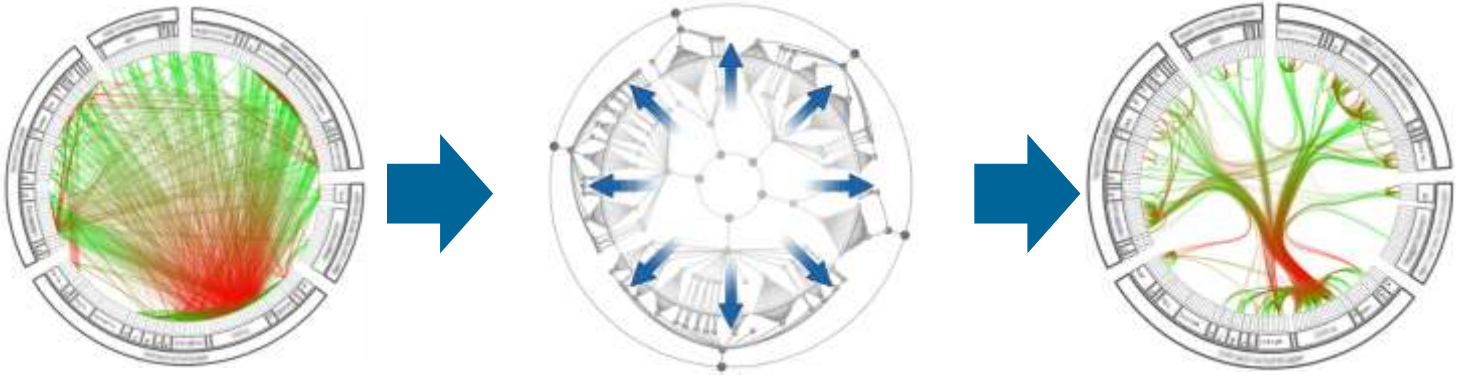
Global spring constant controls edge stiffness



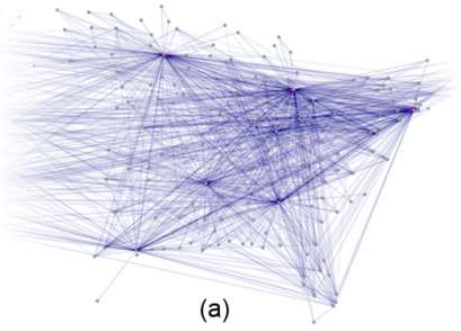


[Holten & van Wijk 2009]

  - Inherent hierarchies (compound graphs):



[Holten 2006]

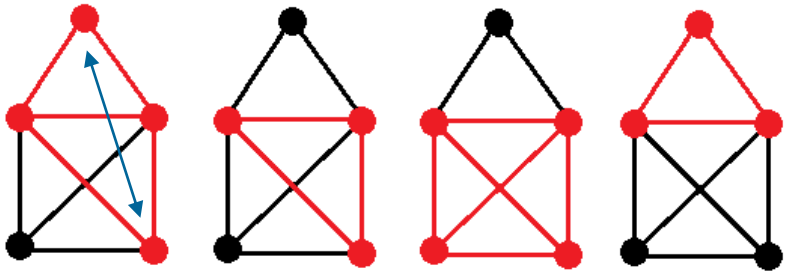


Node Aggregation

- Nodes and edges merged together → super-nodes / meta-nodes
- Aggregation of nodes based on:
  - Pre-defined node hierarchies (compound graph)
  - Aggregation according to node attributes
  - Graph topology

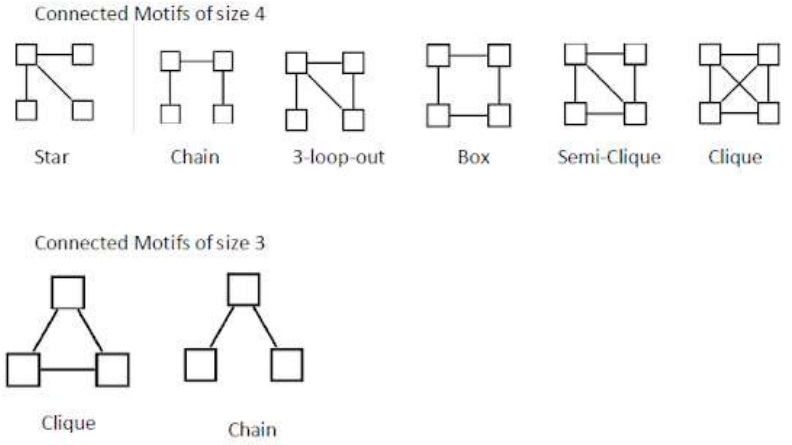
Node Aggregation Based On Graph Topology

- Graph Cliques:
  - Complete subgraphs of undirected graphs
  - All vertex pairs are adjacent



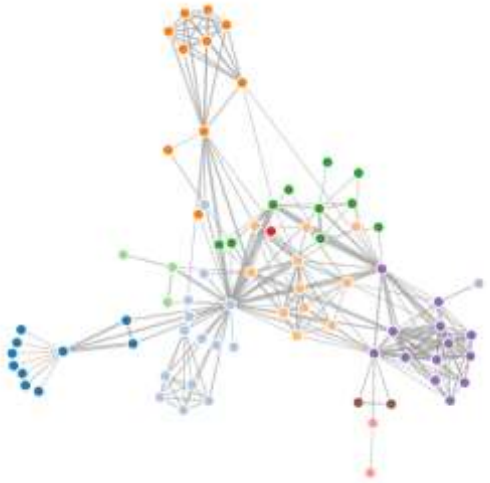
https://i.stack.imgur.com/MrlSG.png

- Graph Motifs:
  - Statistically significant sub-graphs
  - Defined by particular pattern



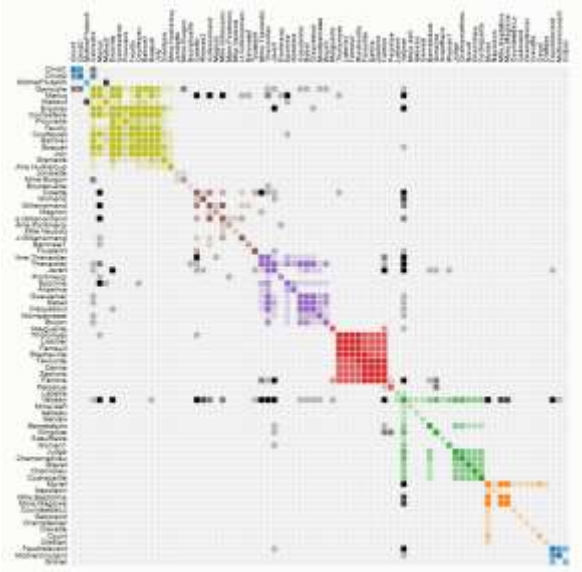
http://cnt13.blogspot.com/2013/04/motifs-fingerprints-of-graph-wha-t-are.html

Recap: Graph Visualization Techniques



Node-link diagram

https://bl.ocks.org/mbostock/4062045



Adjacency matrix

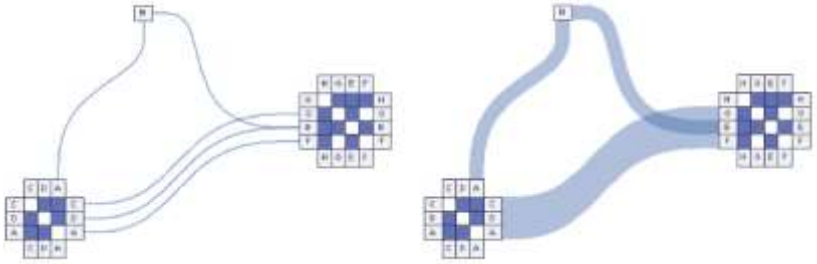
https://bost.ocks.org/mike/miserables/

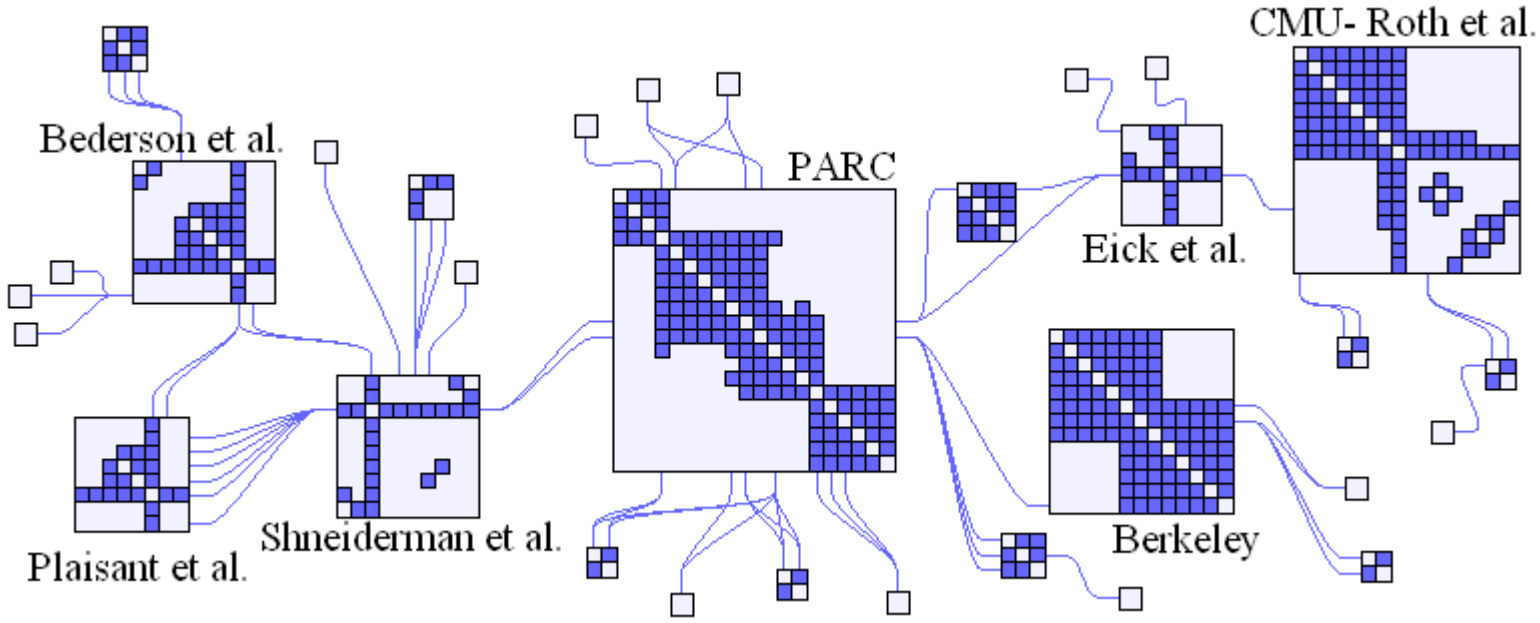
Efficient for sparse networks

Efficient for dense networks

NodeTrix

- Many networks are globally sparse but locally dense
- Cliques are adjacency matrices
- Matrices are nodes connected by links

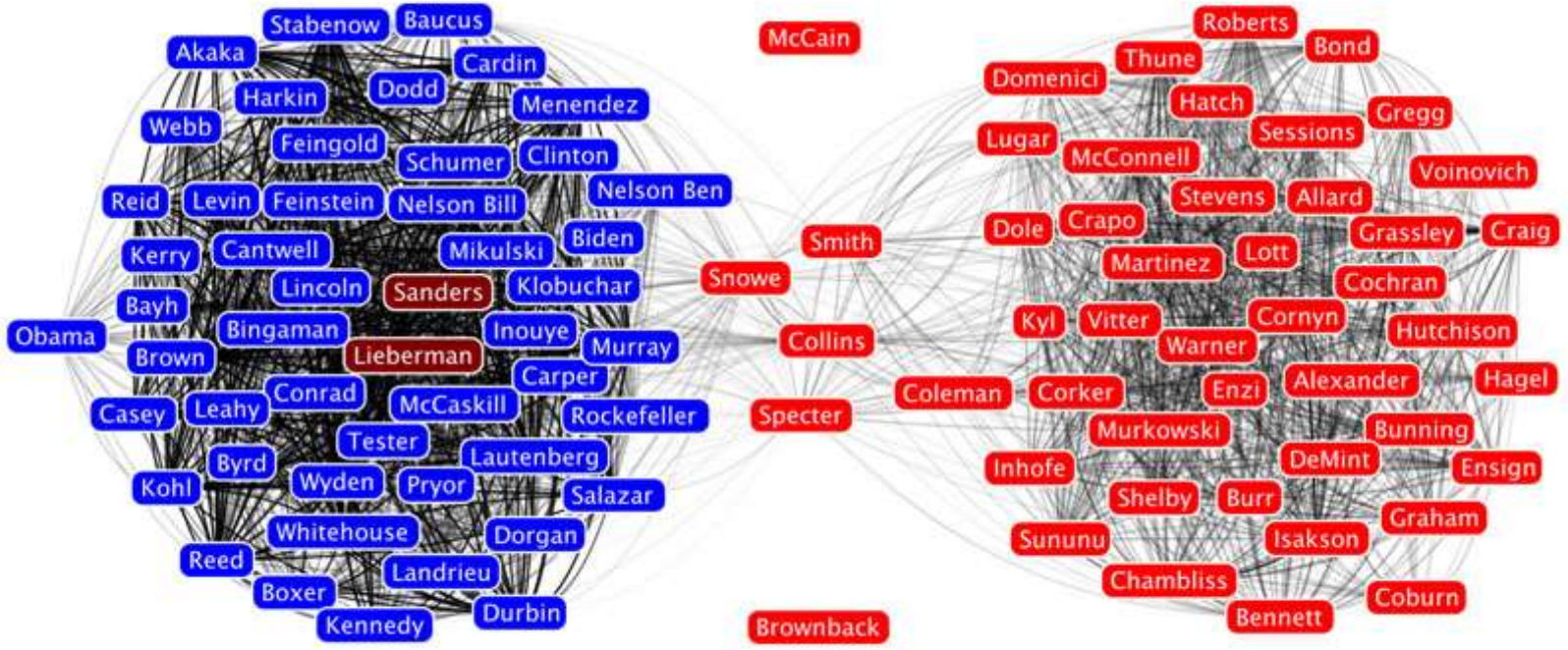




[Henry et al., TVCG 2007]

Motif Simplification

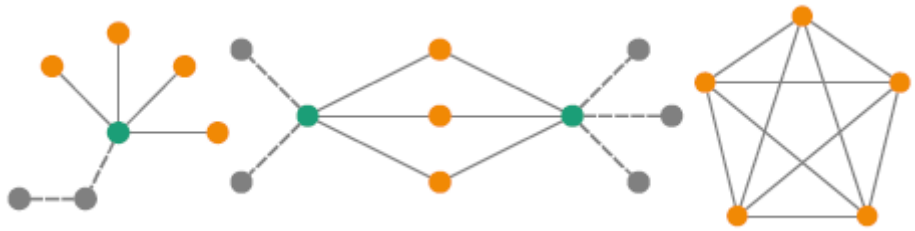
- Example: U.S. Senate 2007 co-voting network



http://www.cs.umd.edu/hcil/science20/

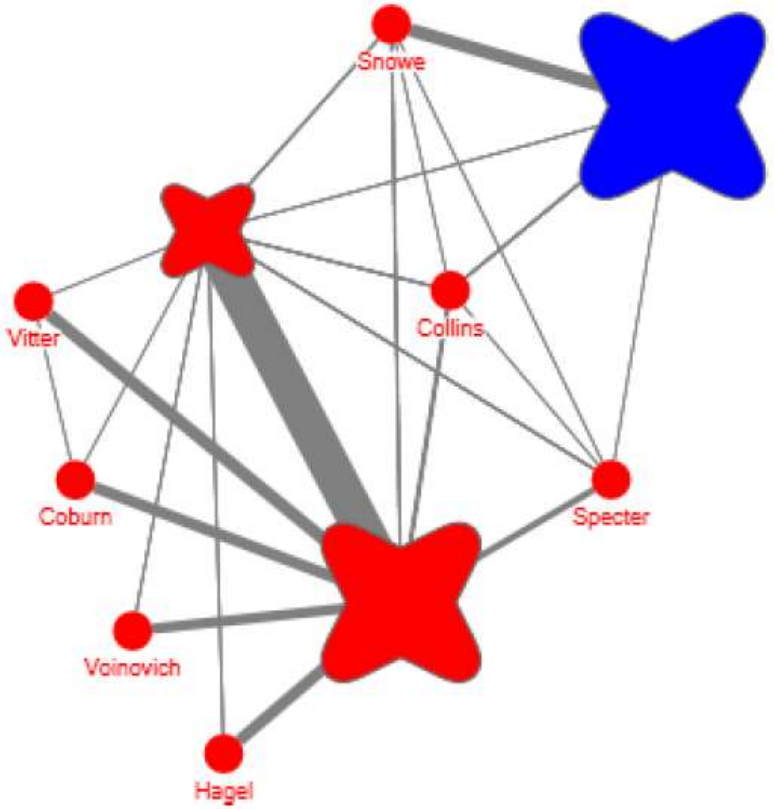
Motif Simplification

- Motifs:



fan connector clique

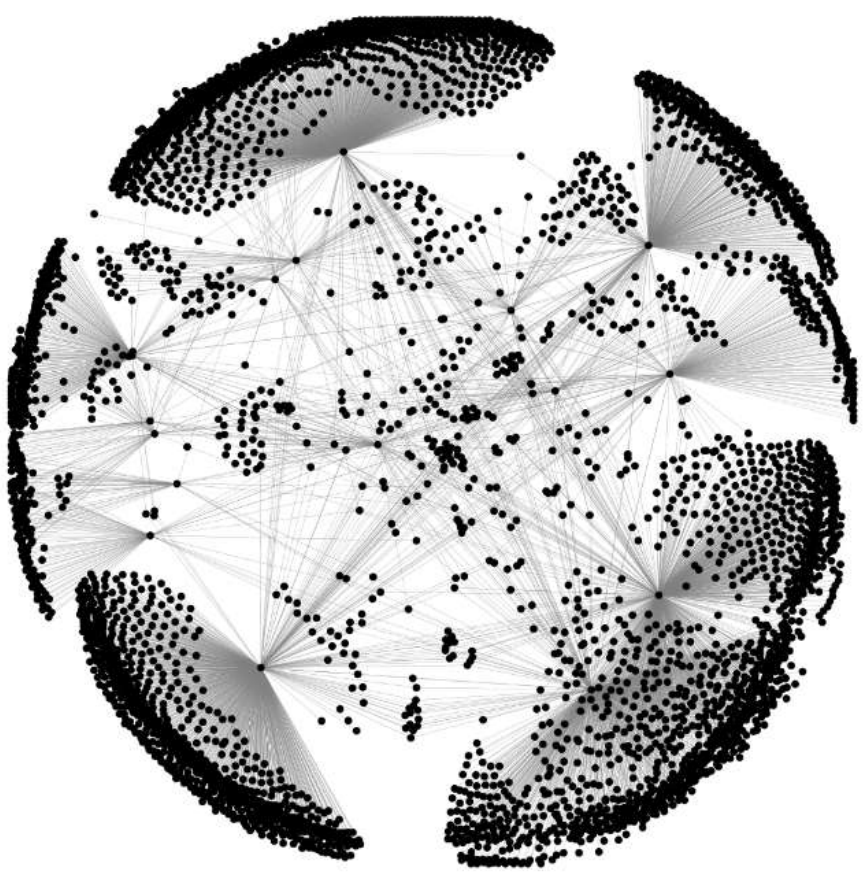
- Common motifs replaced by glyphs
- Example: clique motif glyphs in co-voting network

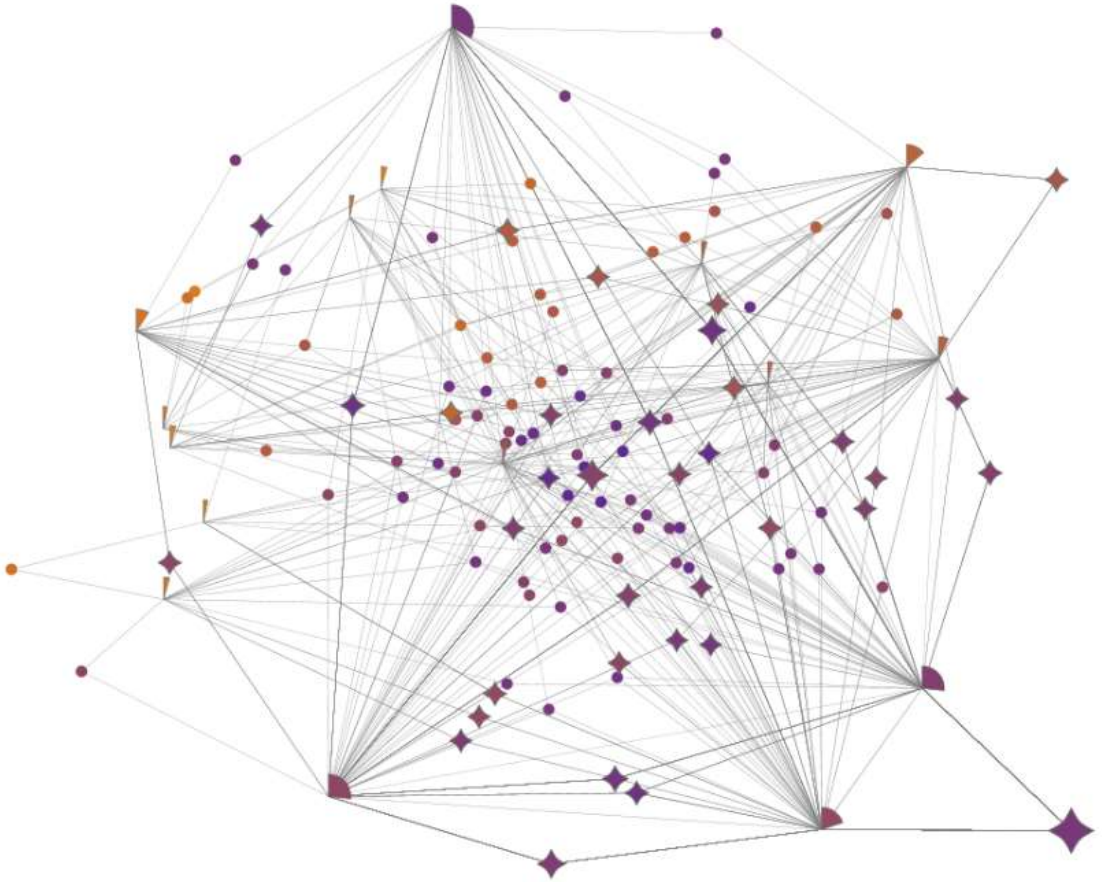


[Dunne et al., Motif simplification, CHI 2013]

Motif Simplification

- Example: Web crawl of ~4000 web pages and ~4000 links





[Dunne et al., Motif simplification, CHI 2013]

Recap: Information Seeking Mantra

Overview First

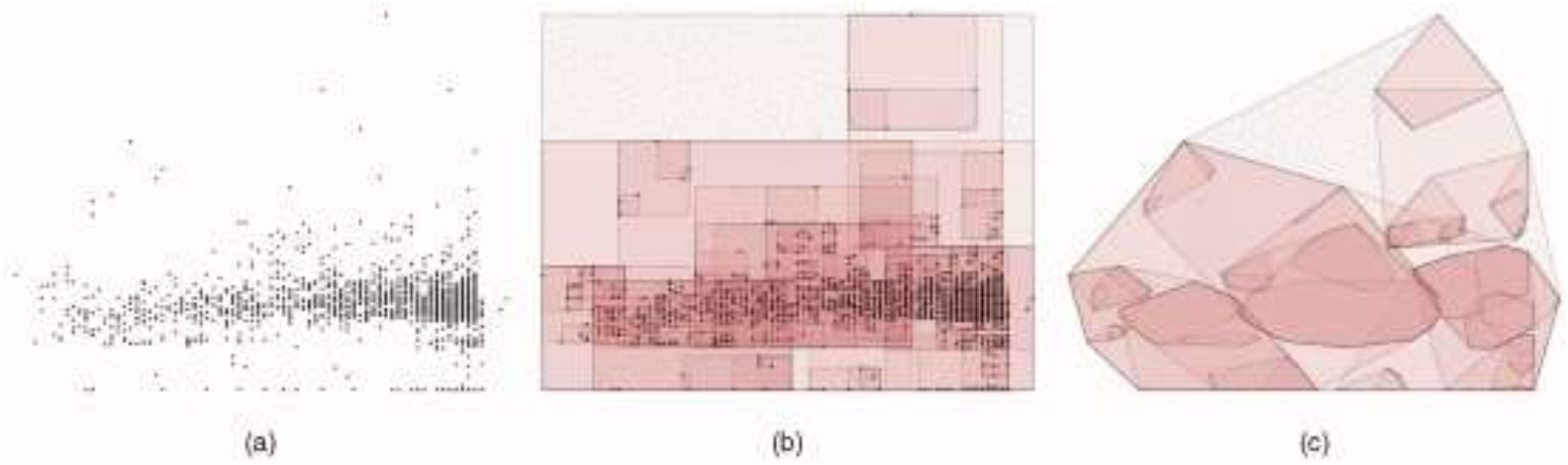
Zoom and Filter

Details on Demand

[Shneiderman 1996]

Hierarchical Aggregation

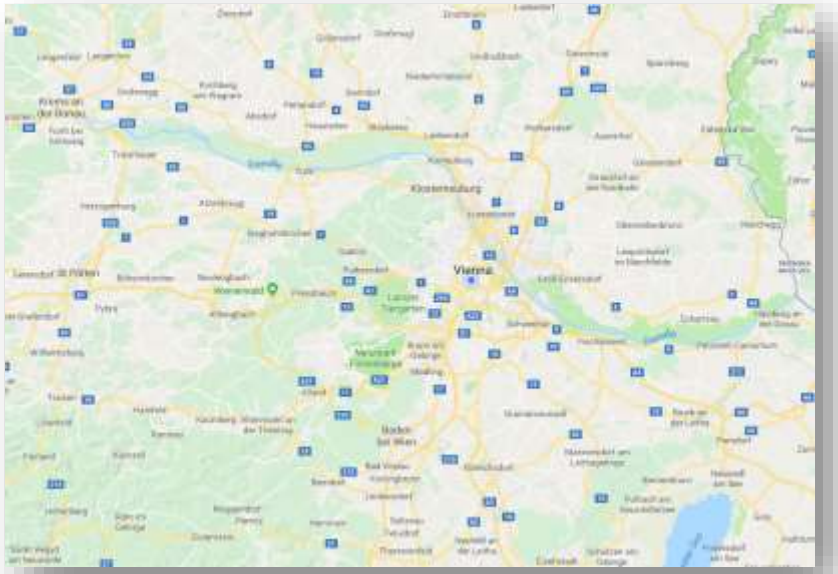
- Hierarchical aggregation of data points
  - Based on a given hierarchy
  - Based on hierarchical clustering
- Visual aggregates for scatterplots: landmark points, boxes, hulls, …
  - Encoding: count, average, extents, median, percentiles,…

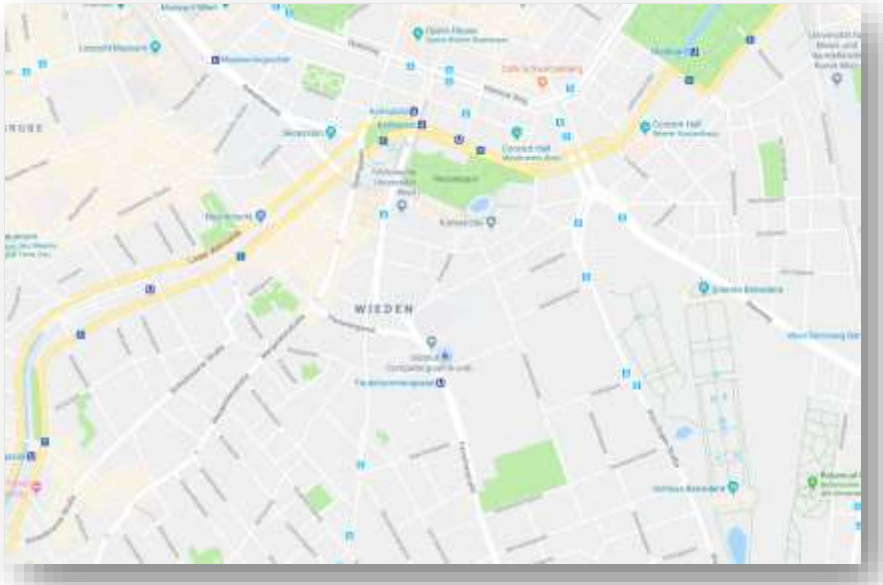


[Elmqvist & Fekete, TVCG 2009]

Semantic Zoom

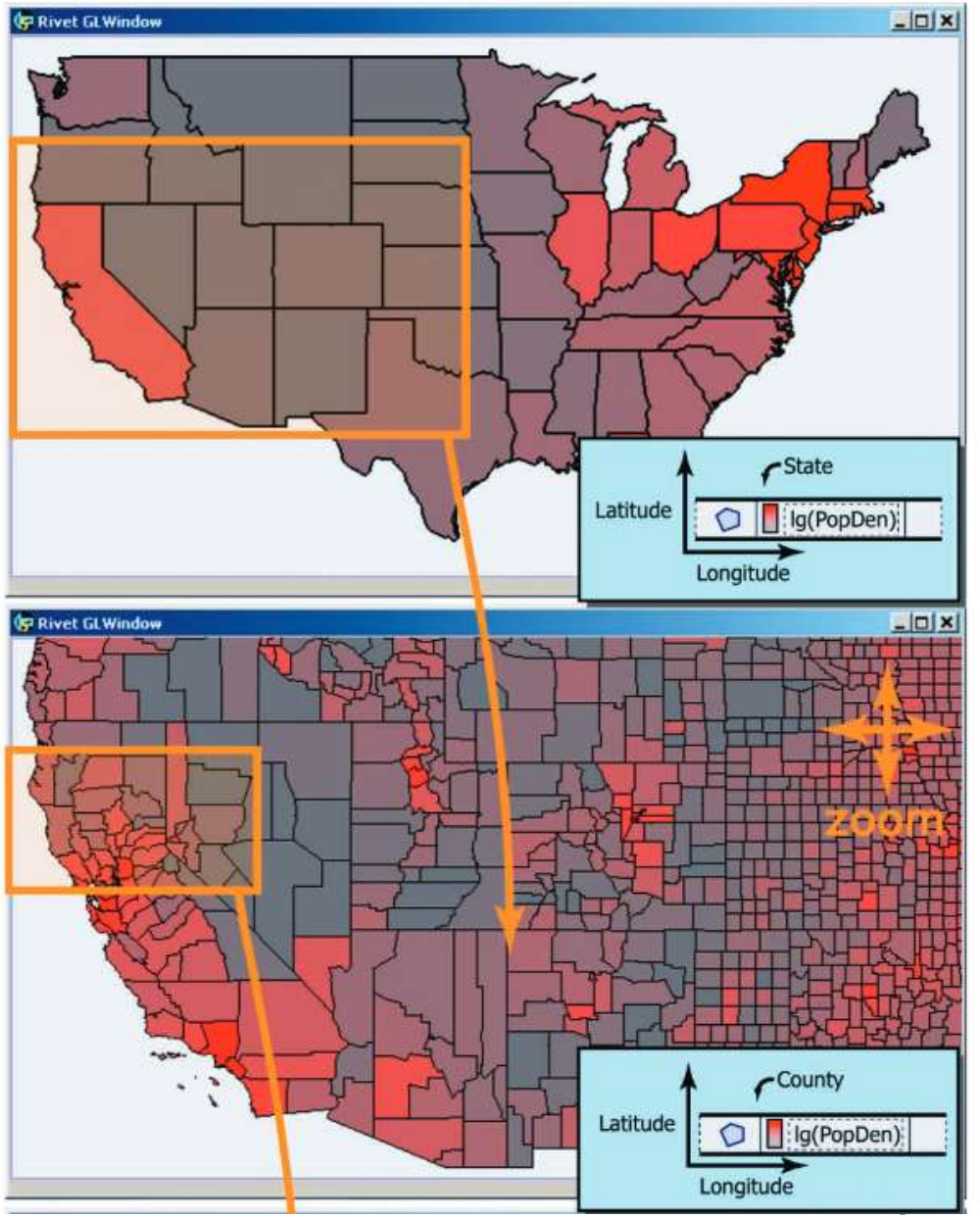
Representation of objects adapts to the number of pixels available





Zooming Choropleth Maps

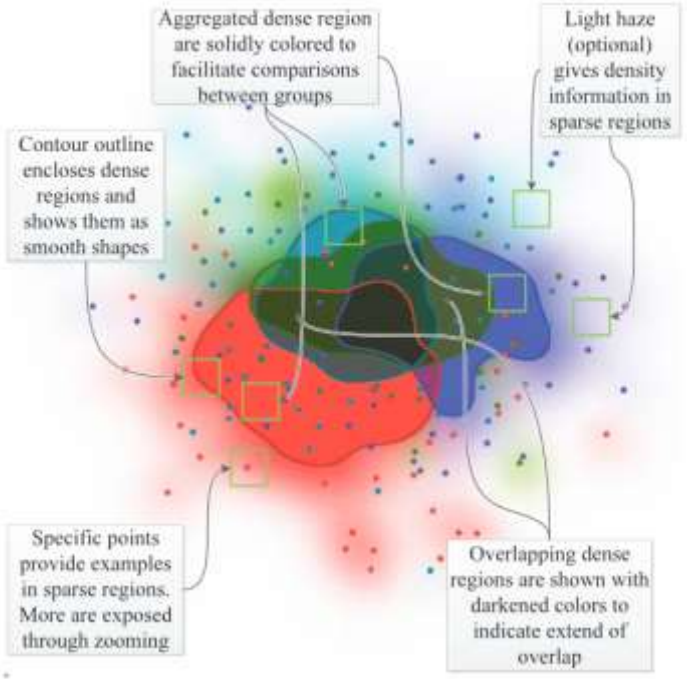
- Example: population density in the US
- Aggregation of county values into states
- Choropleth map

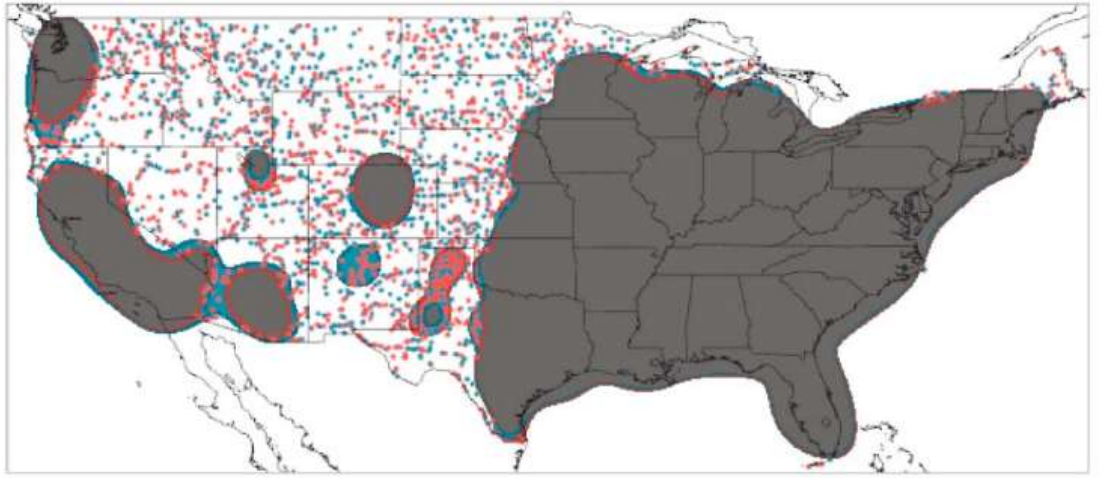


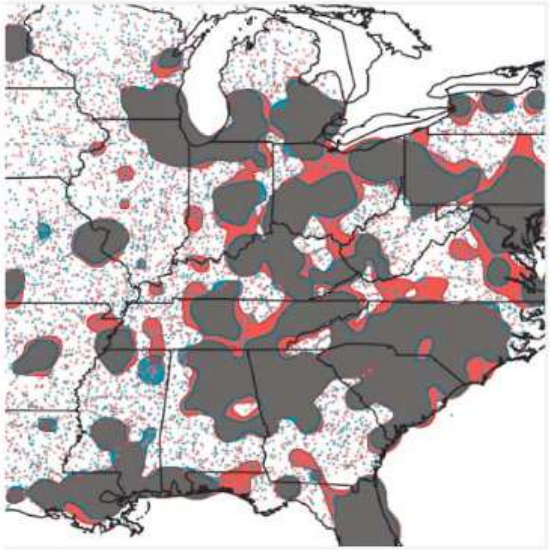
[Stolte et al., TVCG 2003]

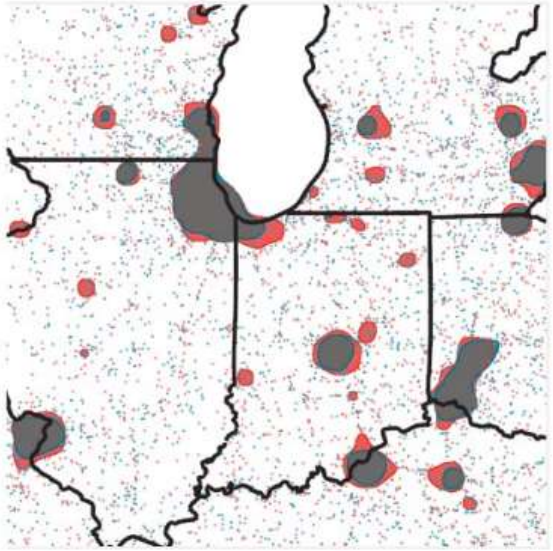
Splatterplots

- KDE → thresholding to identify dense regions
- Example: car crashes (2005 = blue, 2010 = red)





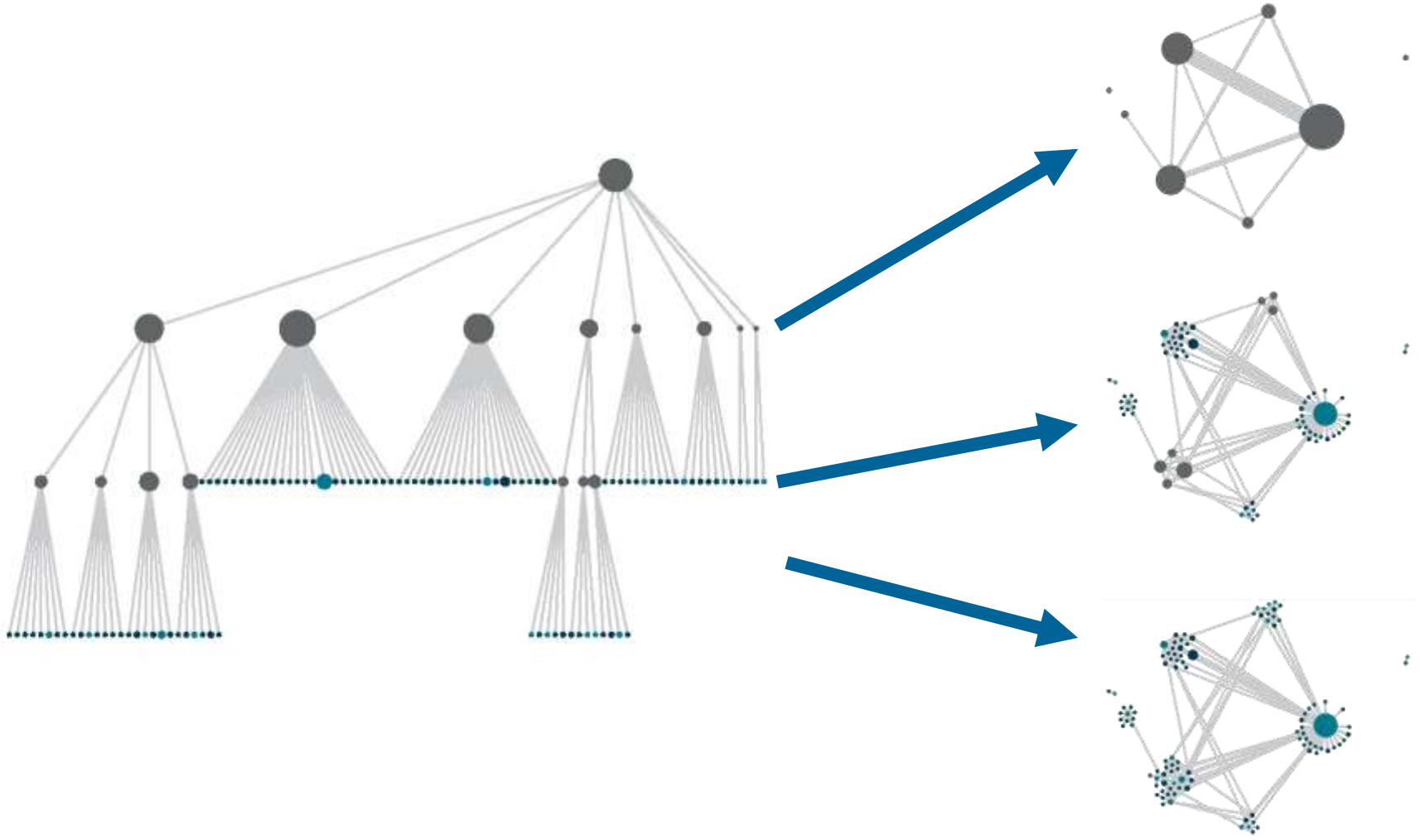




[Mayorga and Gleicher, Splatterplots: Overcoming Overdraw in Scatter Plots, TVCG 2013]

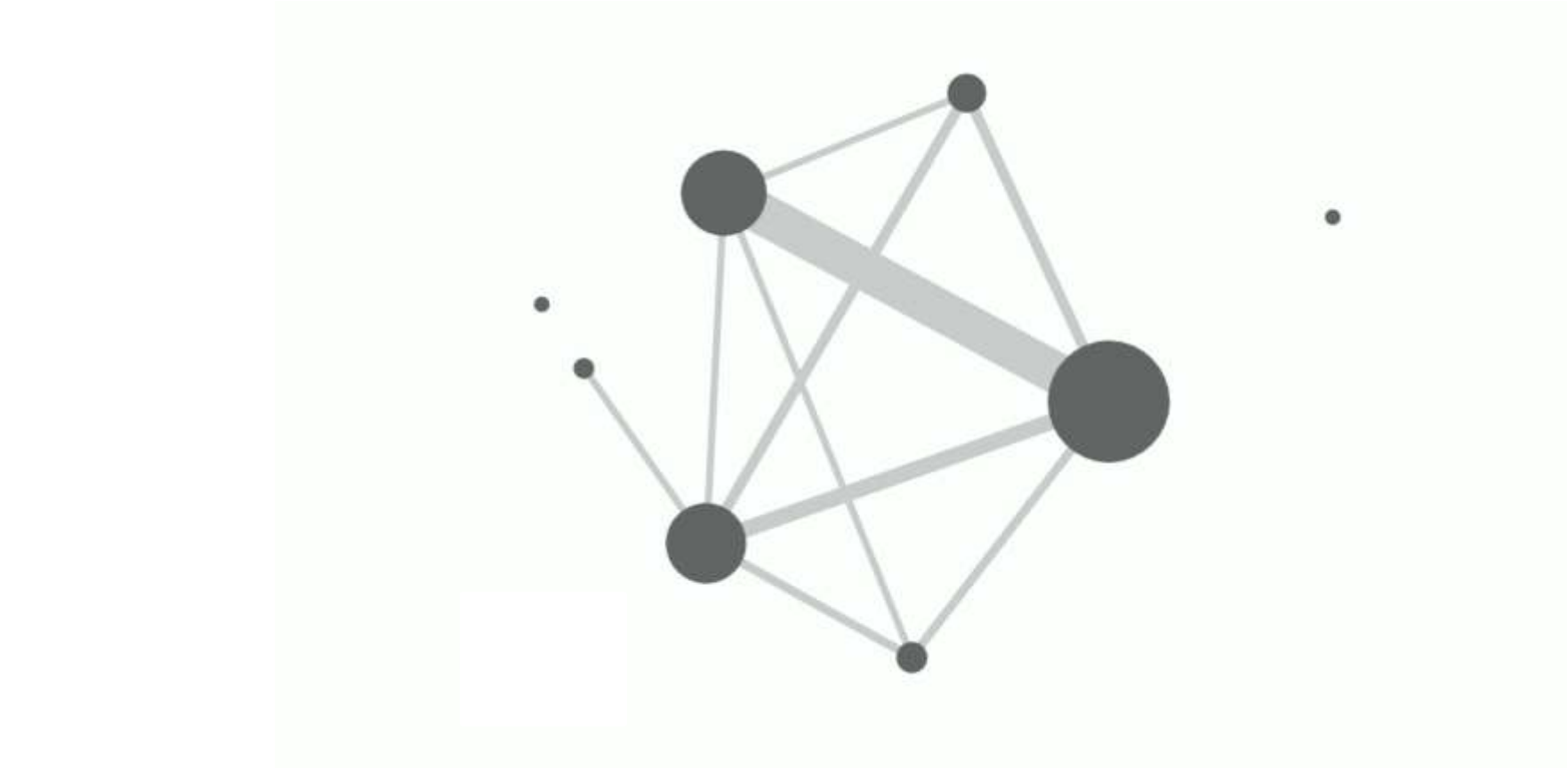
Firework Plots

- Hierarchical Aggregation: Agglomerative Clustering

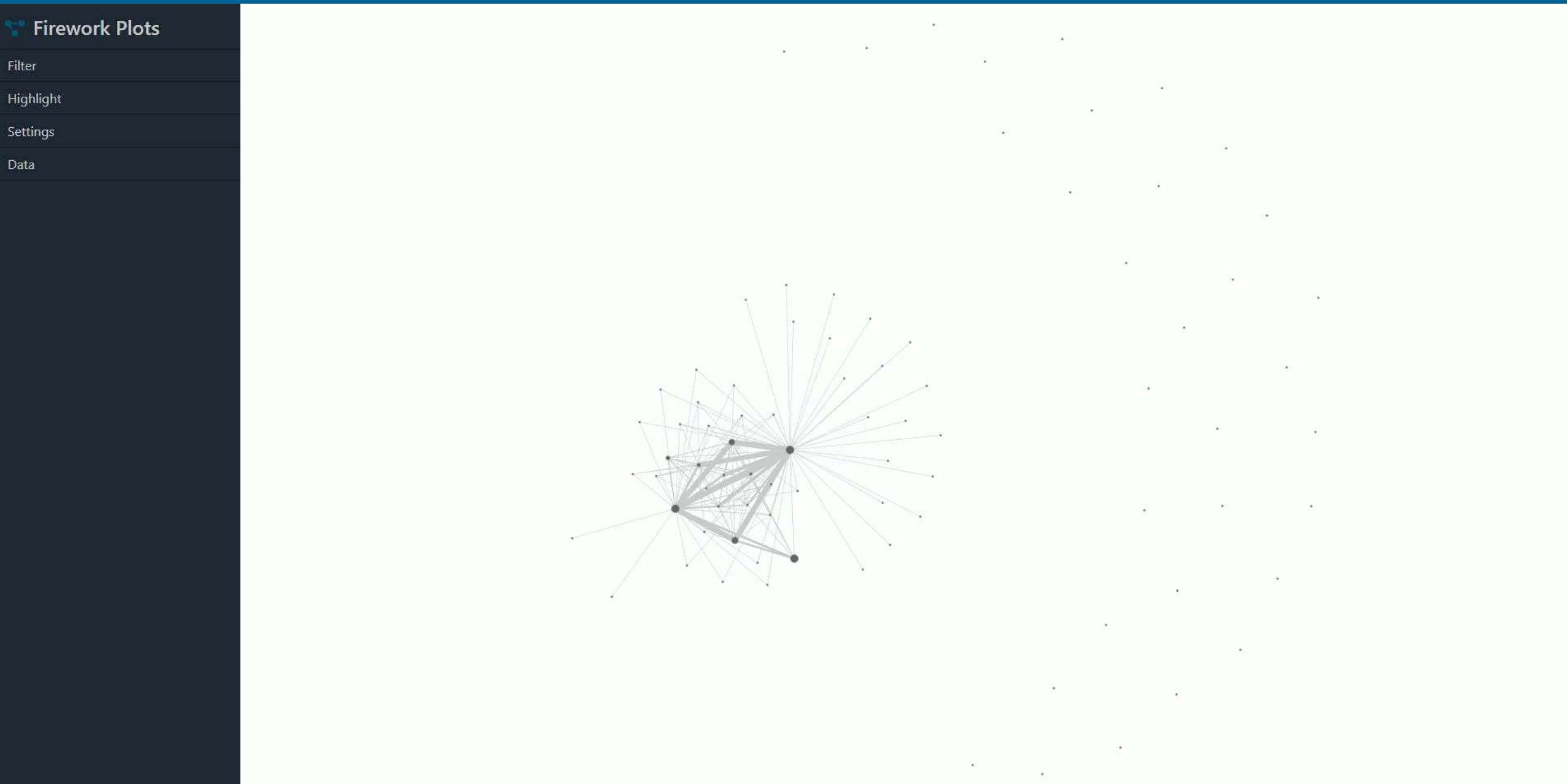


Firework Plots

- Semantic Zoom

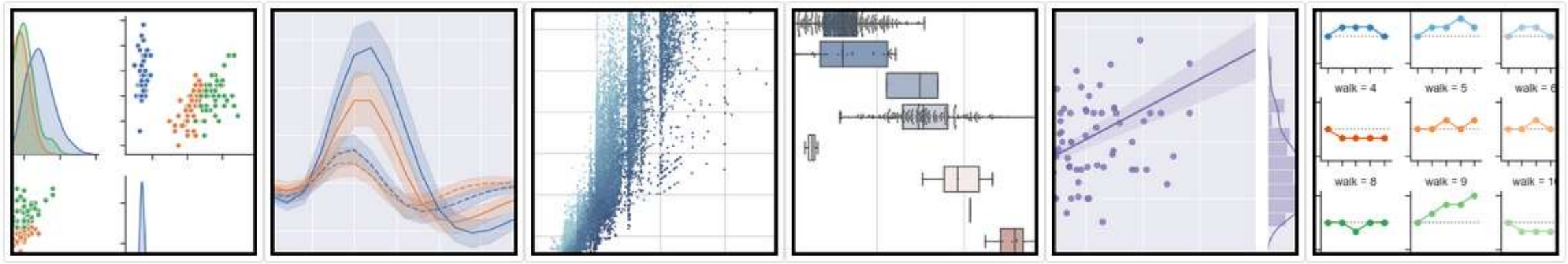


Use Case

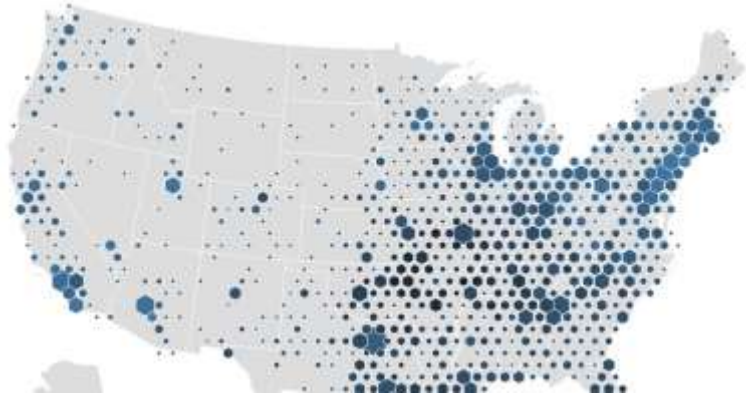


Useful Tools

- Seaborn: https://seaborn.pydata.org/



- d3-hexbin: https://github.com/d3/d3-hexbin
- d3-zoom: https://github.com/d3/d3-zoom



Interactive Scalability

- Aggregate visualizations scale well with big data
- New challenge: real-time response rate of interactive visualizations
- 100 ms: interactive response time limit
  - (Loading)
  - Brushing and linking / dynamic queries
    → filter & new aggregation
  - Panning / zooming
    → change of bin resolution / number of samples

Sources of Latency

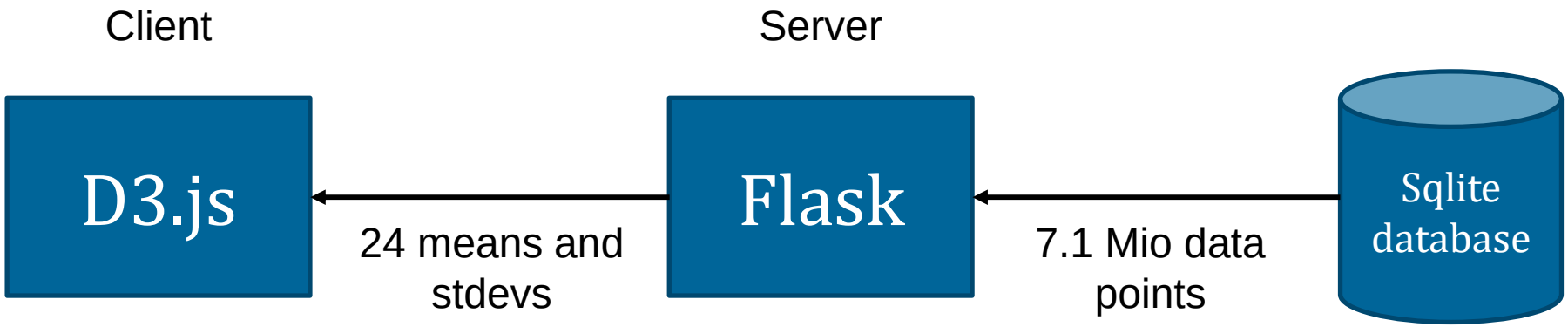
- Data query
- Data processing
- Visualization / rendering

Example: Flight Data

- ~7 Mio flights
- Departure time (hours of day) and departure delays (min) in 2008

Example: Flight Data

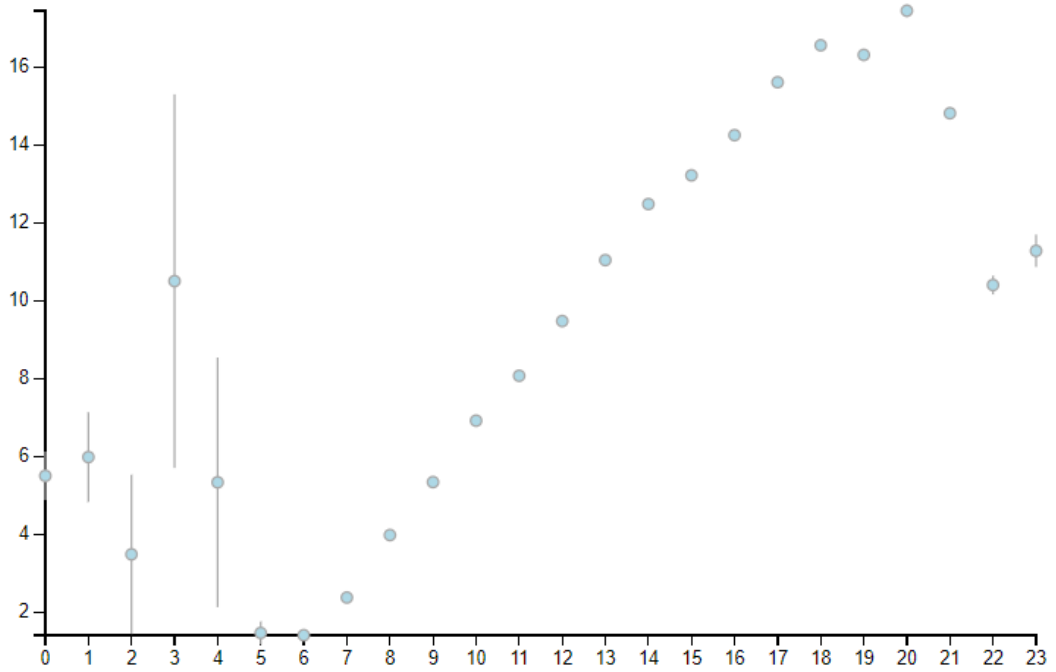
- ~7 Mio flights
- Departure time (hours of day) and departure delays (min) in 2008
- Visualize means and 95% confidence intervals for all hours of a day



- Flask: RESTful web framework for Python
  http://flask.pocoo.org/

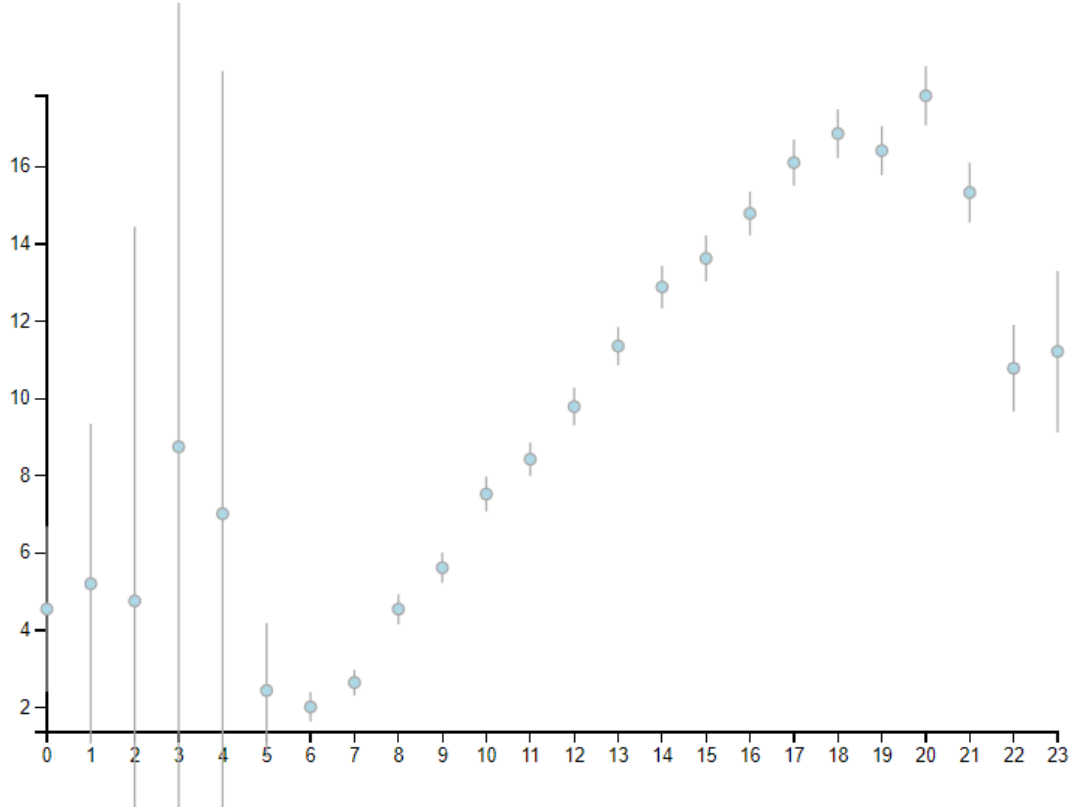
Example: Flight Data

- ~7 Mio flights: mean and confidence intervals per hour of the day
- Takes several seconds - minutes to compute on server
- Precomputation limits interactivity!



Example: Flight Data

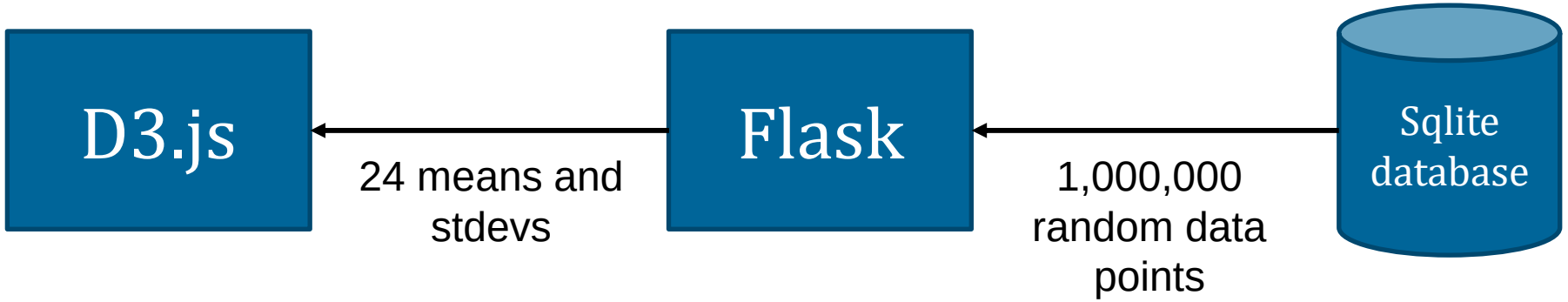
- Random sample
- ~ 1 Mio flights



1 Mio samples

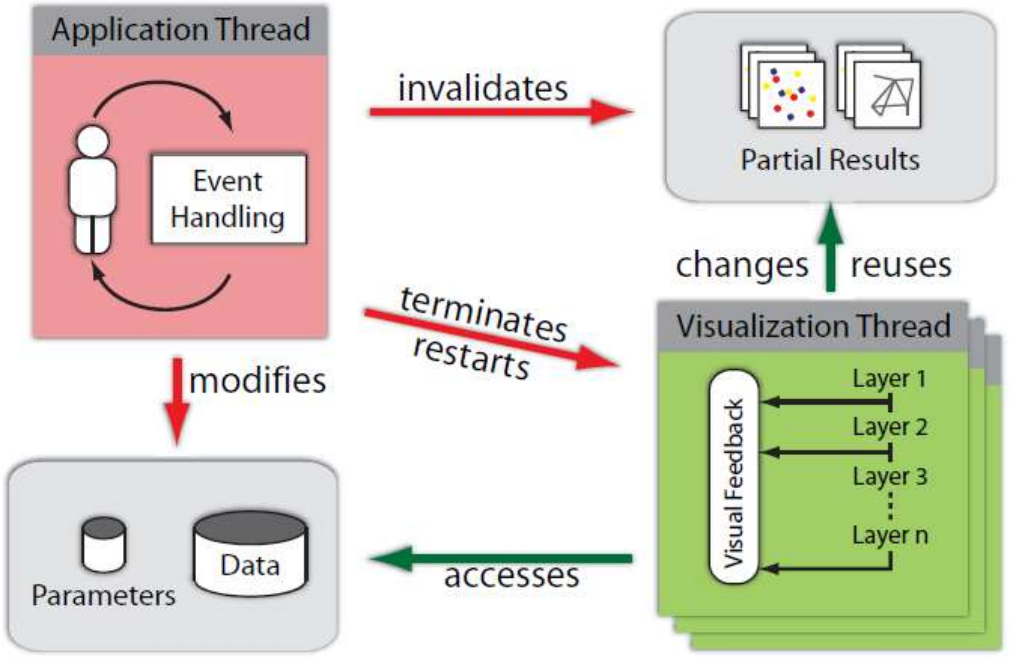
Client

Server



Progressive Analytics

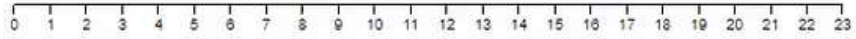
- Always produces partial results of computations under user-specified bounds that converge to the final result
- User can interactively change parameters



[Piringer et al., TVCG 2009]

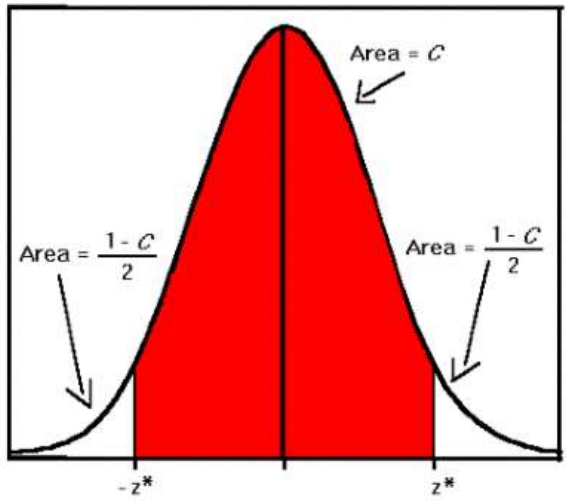
Demo

- ~ 7 Mio flights
- Incrementally refine means and confidence intervals



Confidence Intervals

- Sample mean
  - $\bar{x} = \frac{\sum x_i}{n}$
- Sample standard deviation
  - $s = \sqrt{\frac{\sum(x_i - \bar{x})^2}{n-1}}$
- Confidence interval
  - Range containing actual value with given probability
  - Normal distribution
  - Sample mean follows t distribution
  - $c = \bar{x} \pm t\frac{s}{\sqrt{n}}$



Online Aggregation

- Estimates of an aggregate query online
- Random sampling

- Online update of mean:

$$\bar{x}_n = \frac{(n-1)\,\bar{x}_{n-1} + x_n}{n} = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n}$$

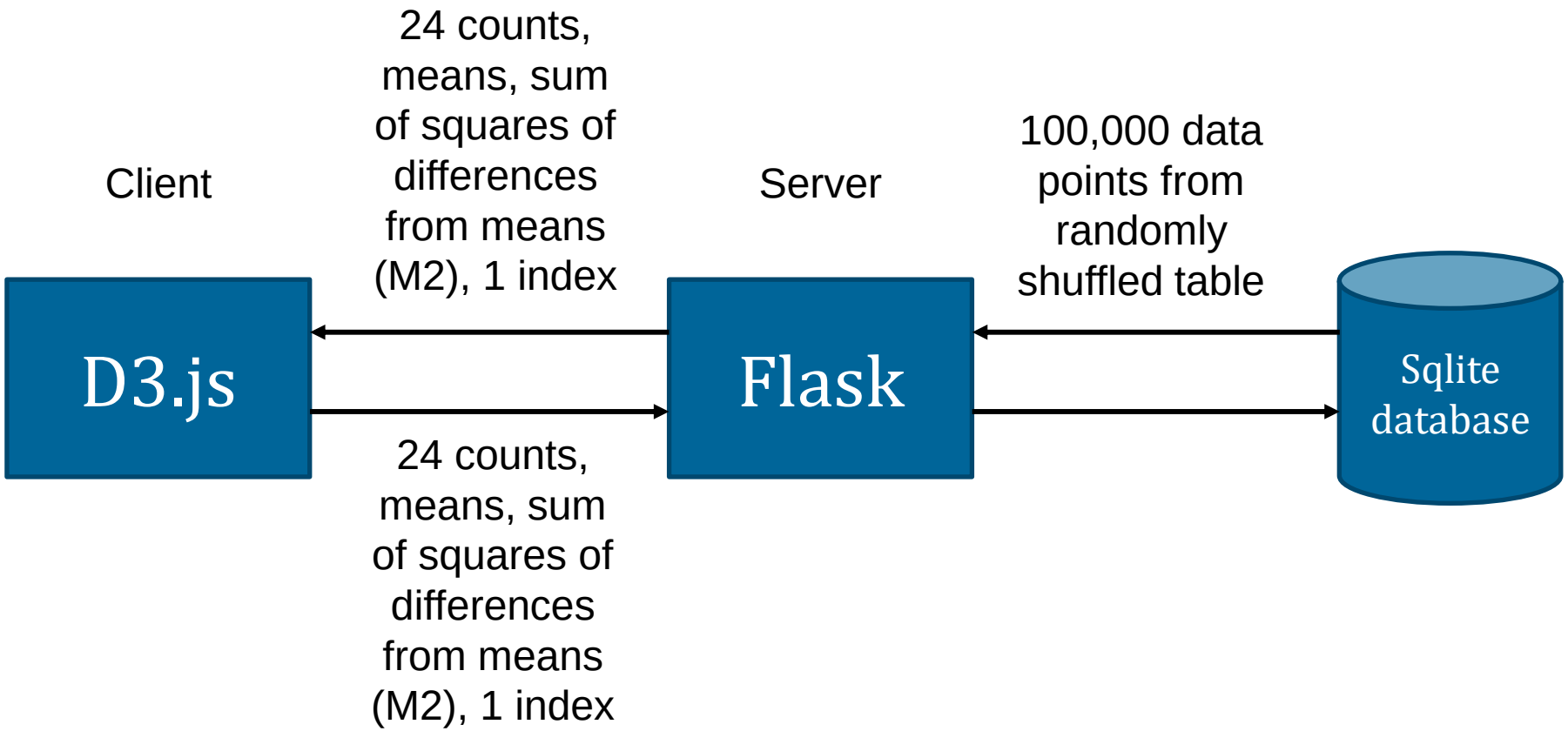
- Online update of variance based on sum of squares of differences from current mean ($M_{2,n}$):

$$M_{2,n} = M_{2,n-1} + (x_n - \bar{x}_{n-1})(x_n - \bar{x}_n)$$

$$s_n^2 = \frac{M_{2,n}}{n-1}$$

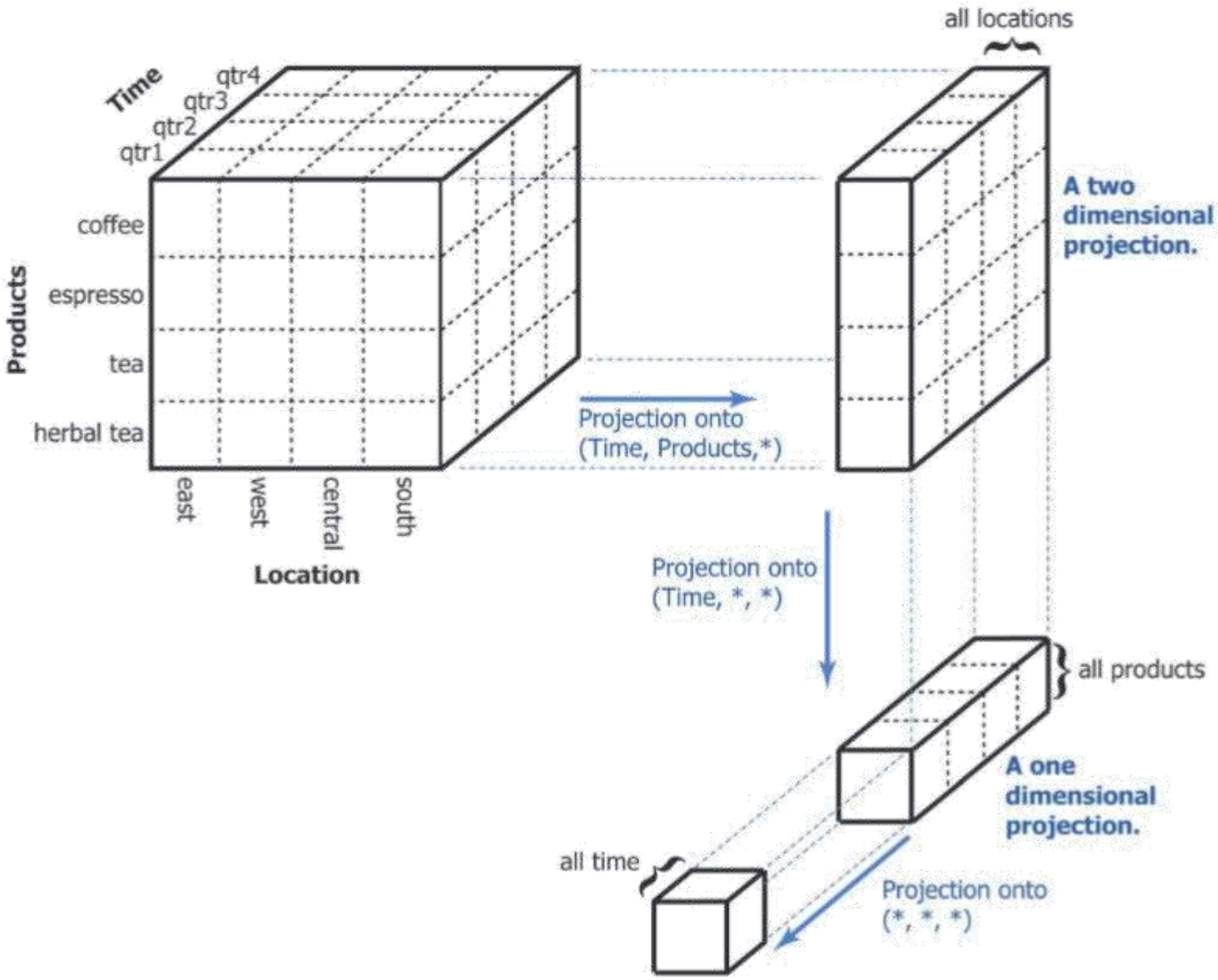
https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance

Example: Online Aggregated Flight Data



Aggregate Queries

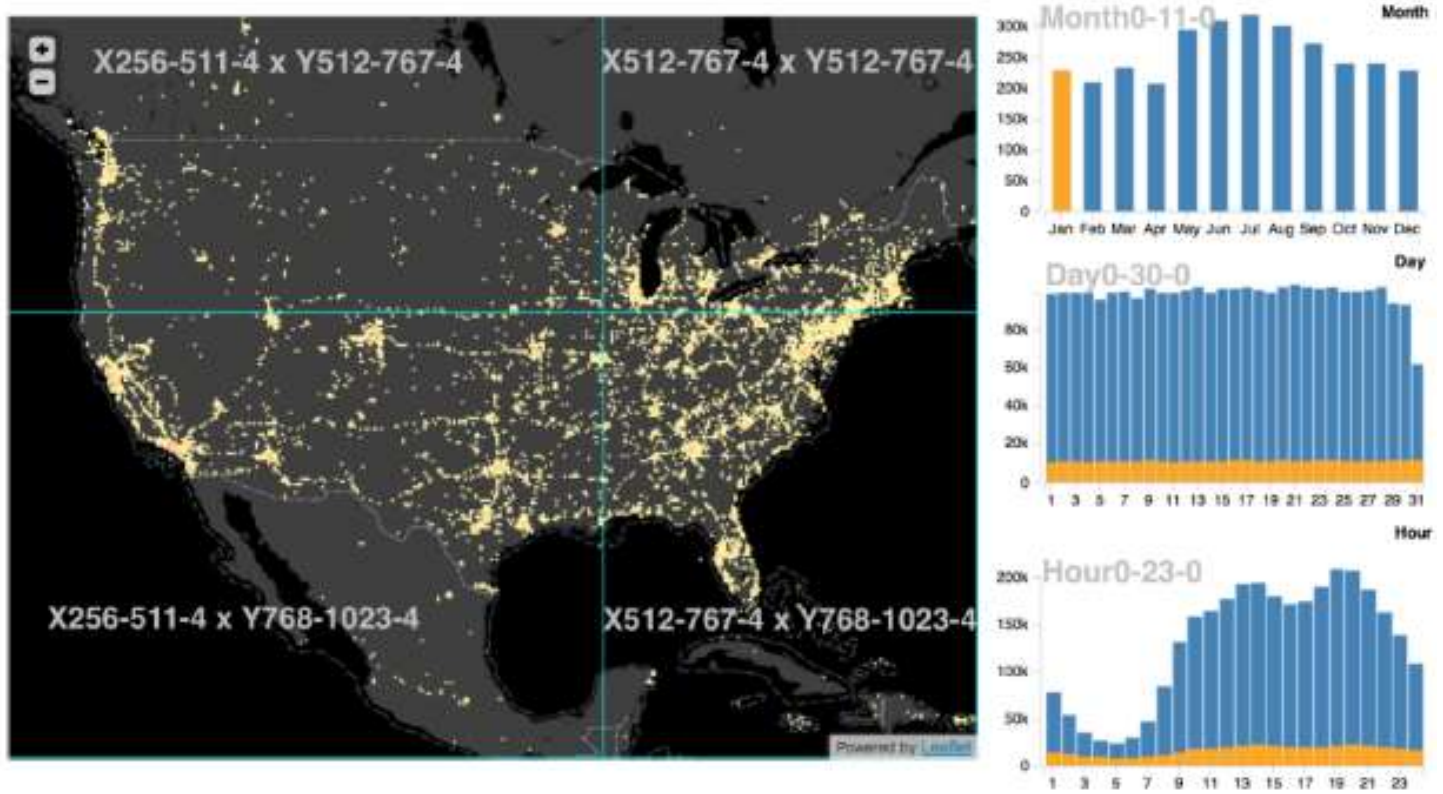
- Aggregations: count, sum, max, min,...
- Data Cubes:
  - Multi-dimensional arrays
  - Quick summaries of data
  - Information split into
    - Independent variables (dimensions)
    - Dependent variables
  - Aggregations are projections



[Stolte et al., TVCG 2003]

Brushing and Linking

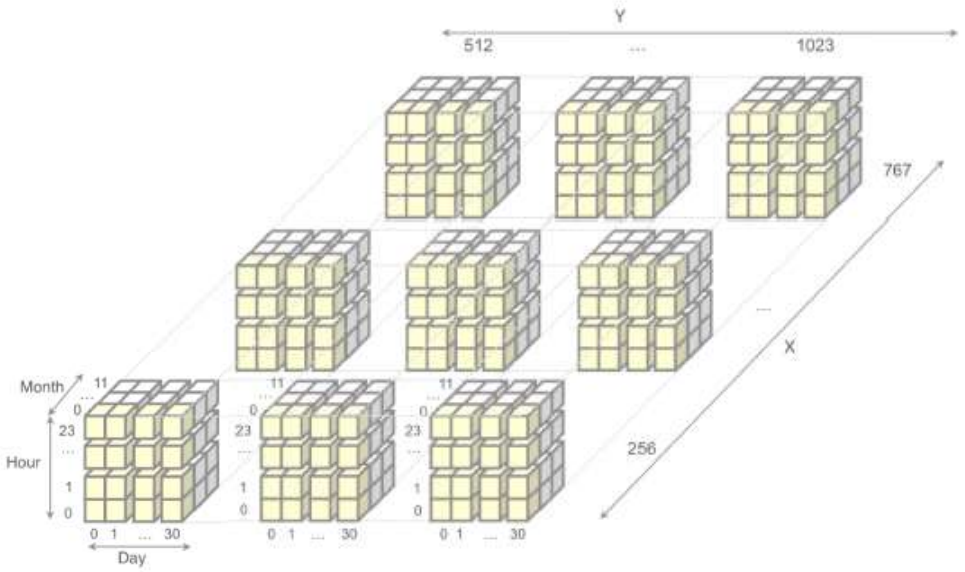
- Example: aggregate visualizations for January

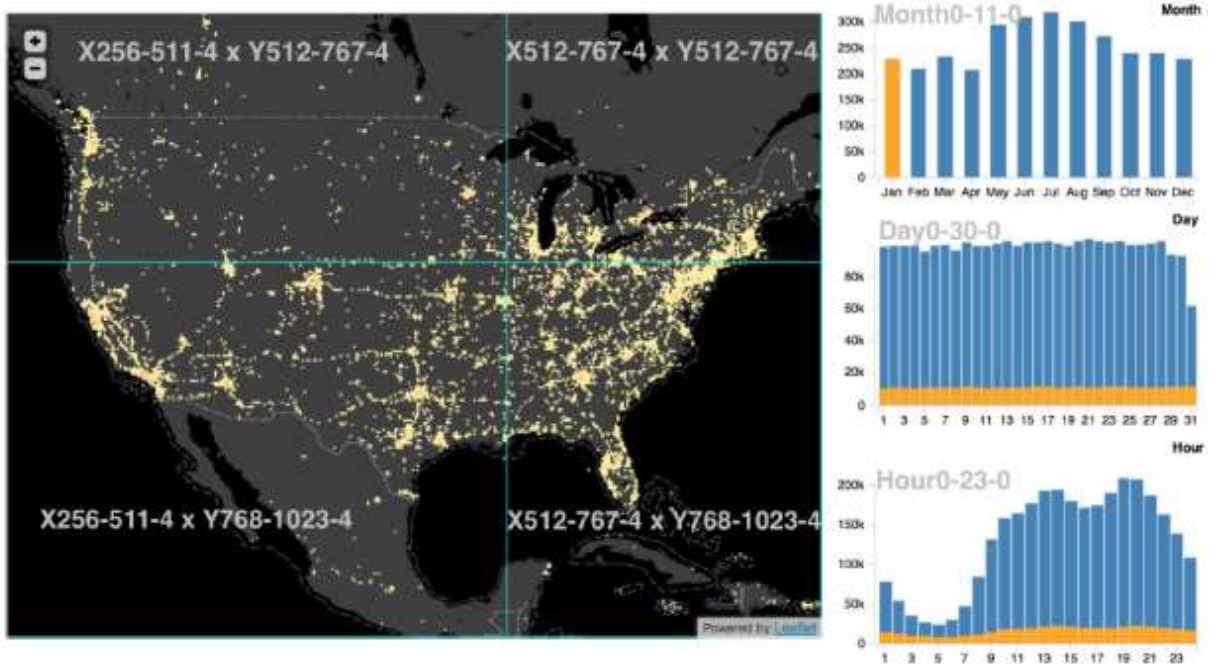


[Lui, Jiang & heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013]

Data Cubes

- Example: (hyper)cube



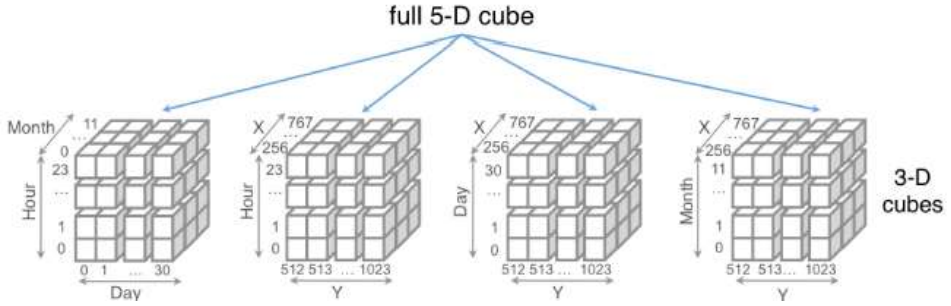


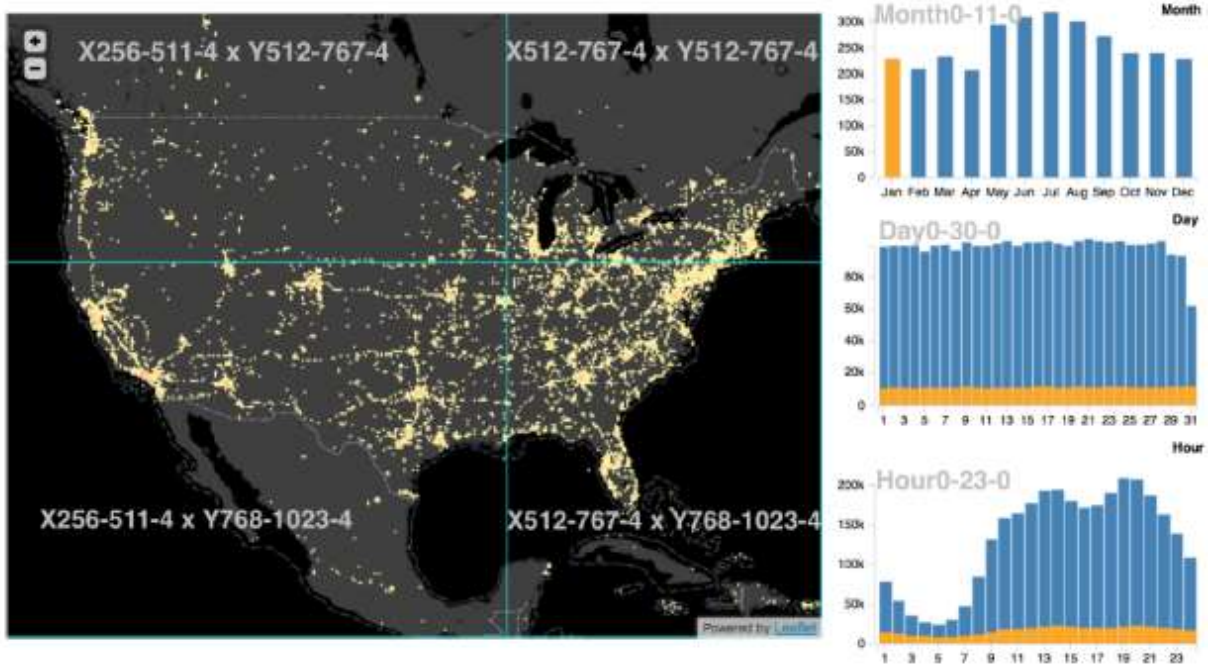
- Size of cube (with bin count b): b^5

[Lui, Jiang & heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013]

Data Cubes

- 1D or 2D aggregated views → 3D or 4D data cubes sufficient
- 3D cube projections:



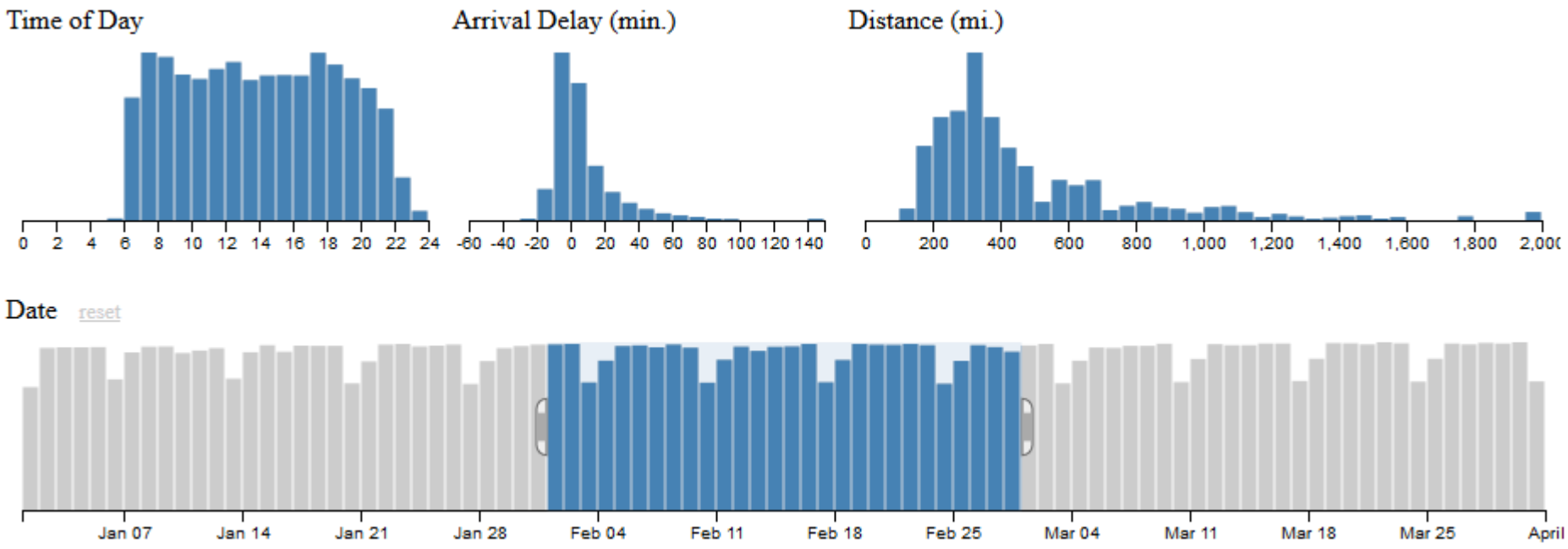


- Size of cubes (with bin count b): $4b^3$

[Lui, Jiang & heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013]

Crossfilter

- http://square.github.io/crossfilter/
- JavaScript library for efficient grouping, filtering, and aggregating

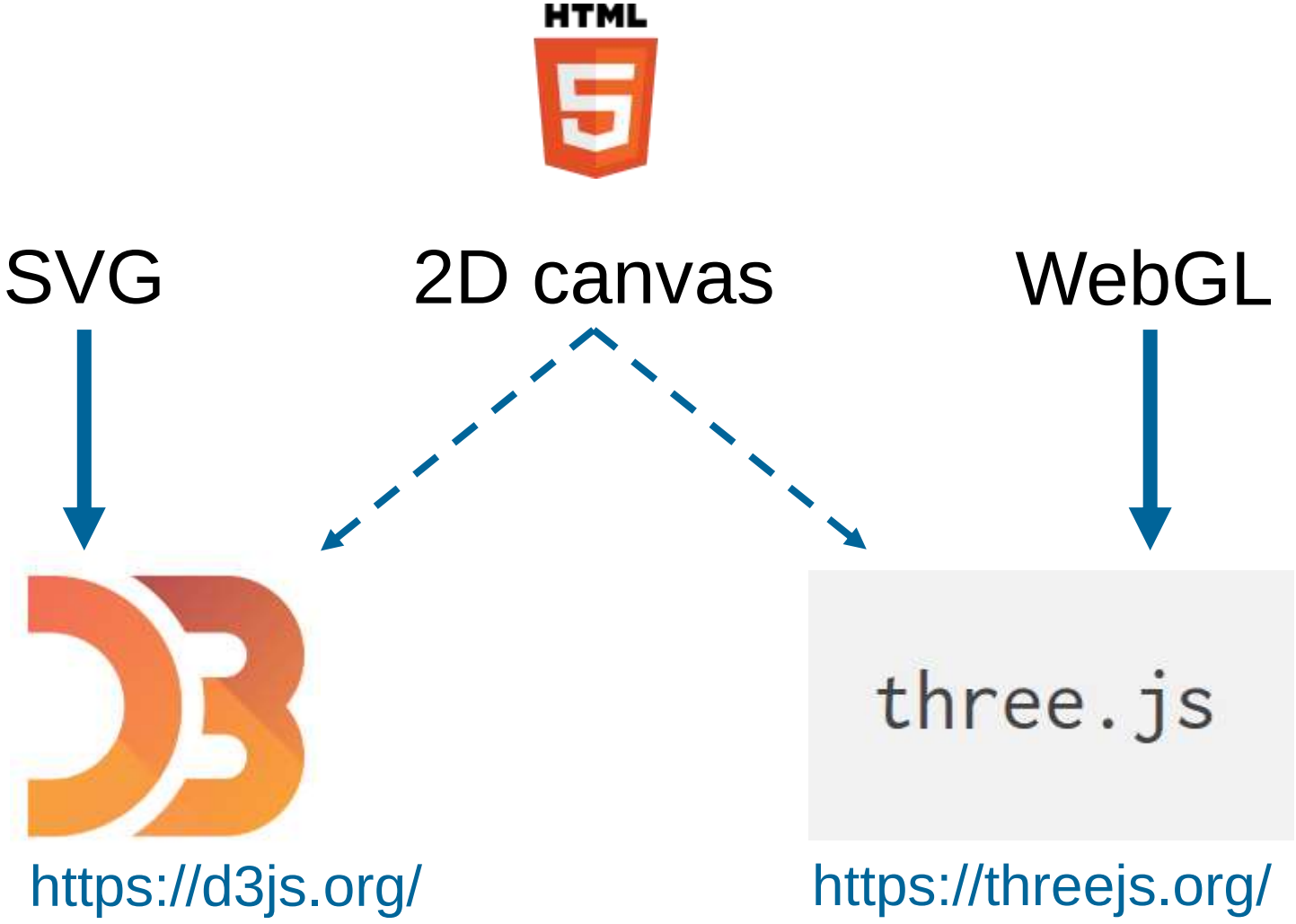


Pandas

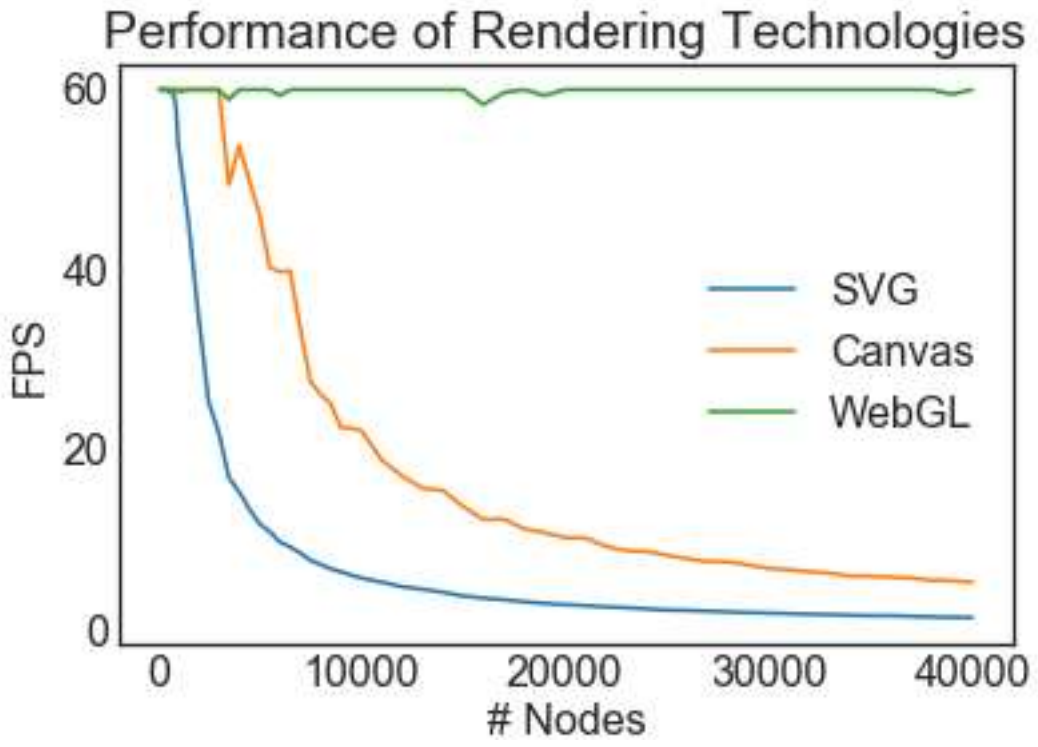
- Python data frames

- Recommendable for machine learning projects in combination with scikit-learn

HTML5 Visualization



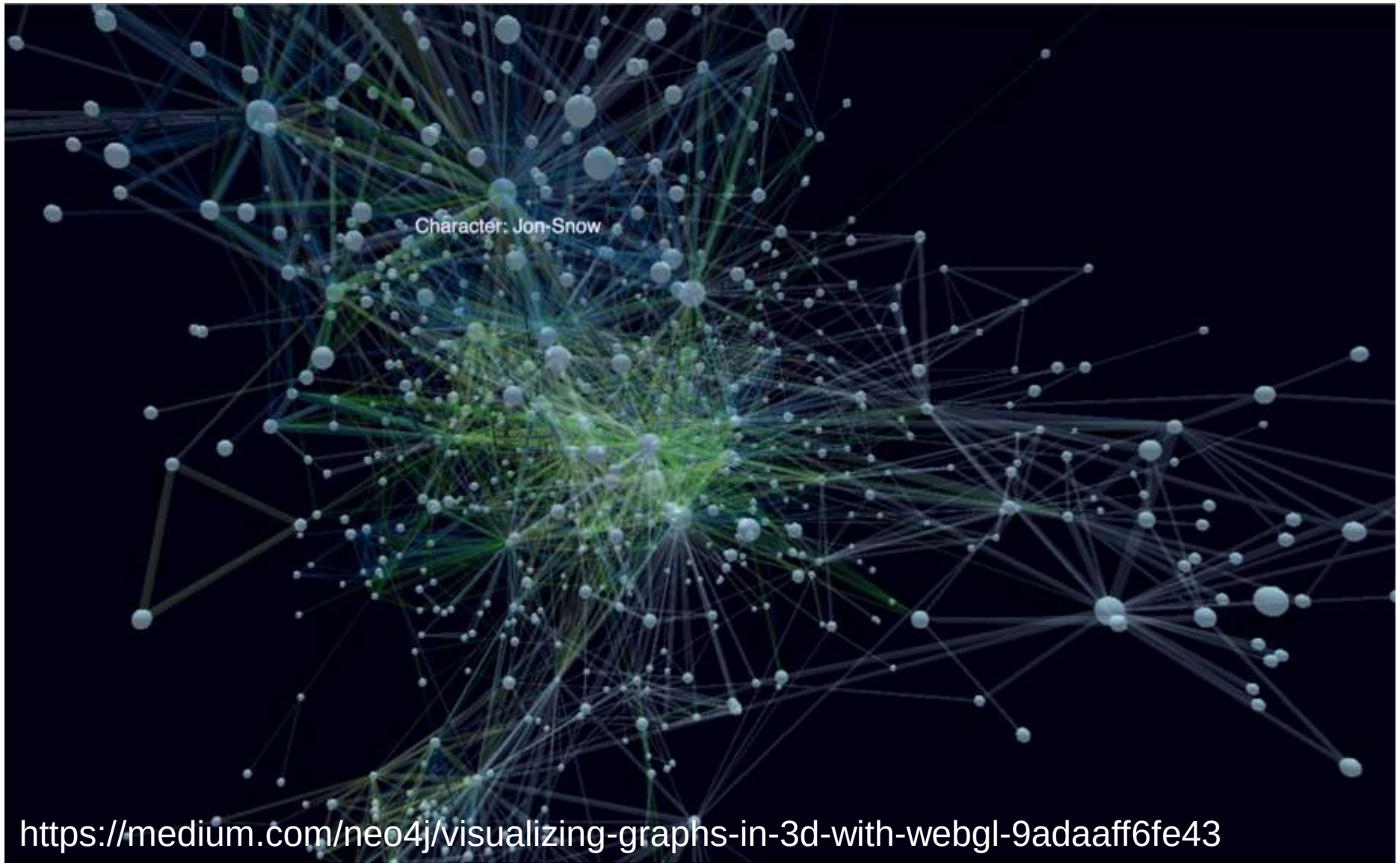
SVG vs Canvas vs WebGL



sigma.js

- JavaScript library for graph drawing
- Canvas 2D and WebGL renderers

3D Force-Directed Graph Library

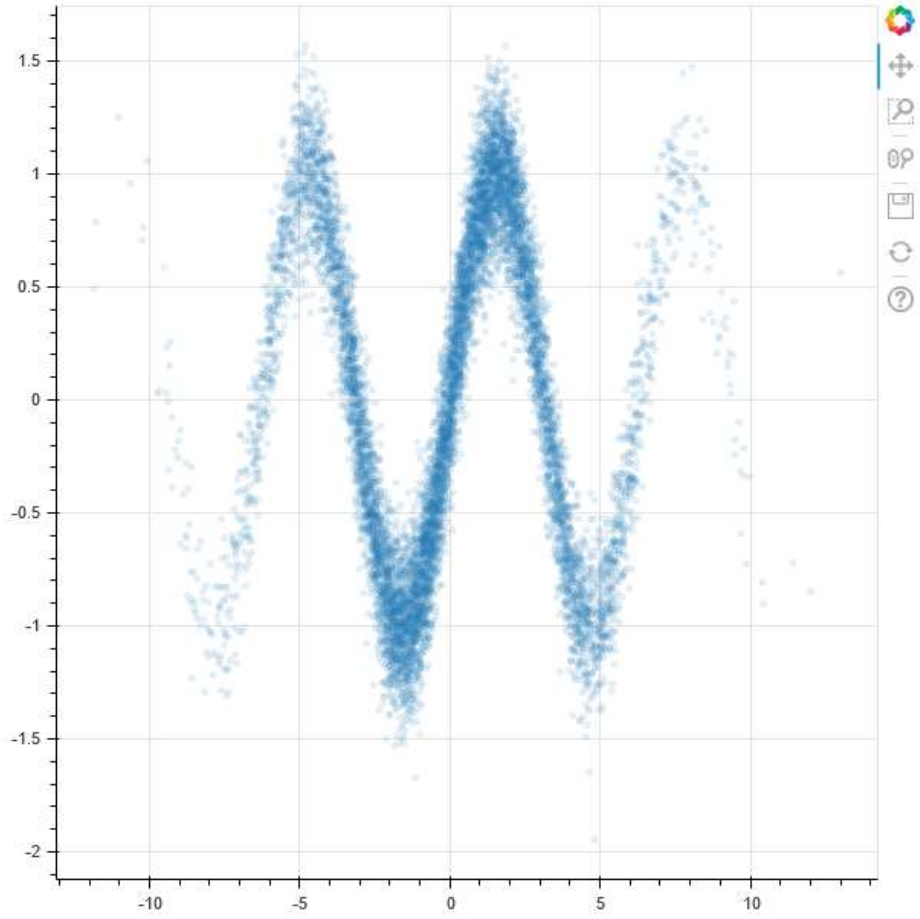


WebGL Rendering with Python

- Bokeh
  - Renders to HTML5 2D canvas
  - Partial support for WebGL rendering

[https://bokeh.pydata.org/en/latest/docs/user_guide/webgl.html]

  - Example: 10,000 scatter dots

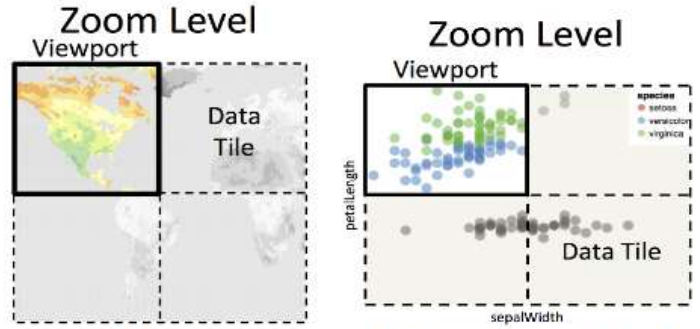


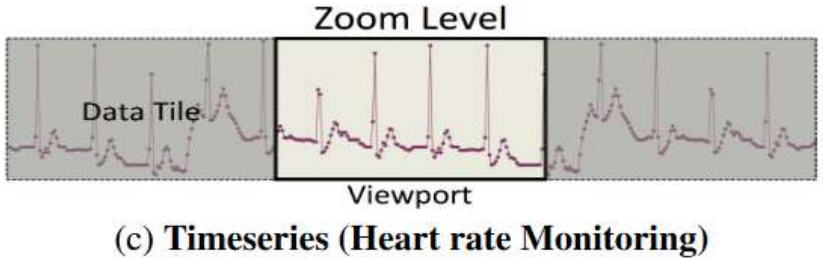
- Plotly
  - SVG and WebGL rendering

Data Tiles

- Views created offline for each zoom level
- Each zoom level separated into equal-size blocks (tiles)
- Interactive map demo: http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/

[Fisher, Hotmap: Looking at Geographic Attention, TVCG 2007]





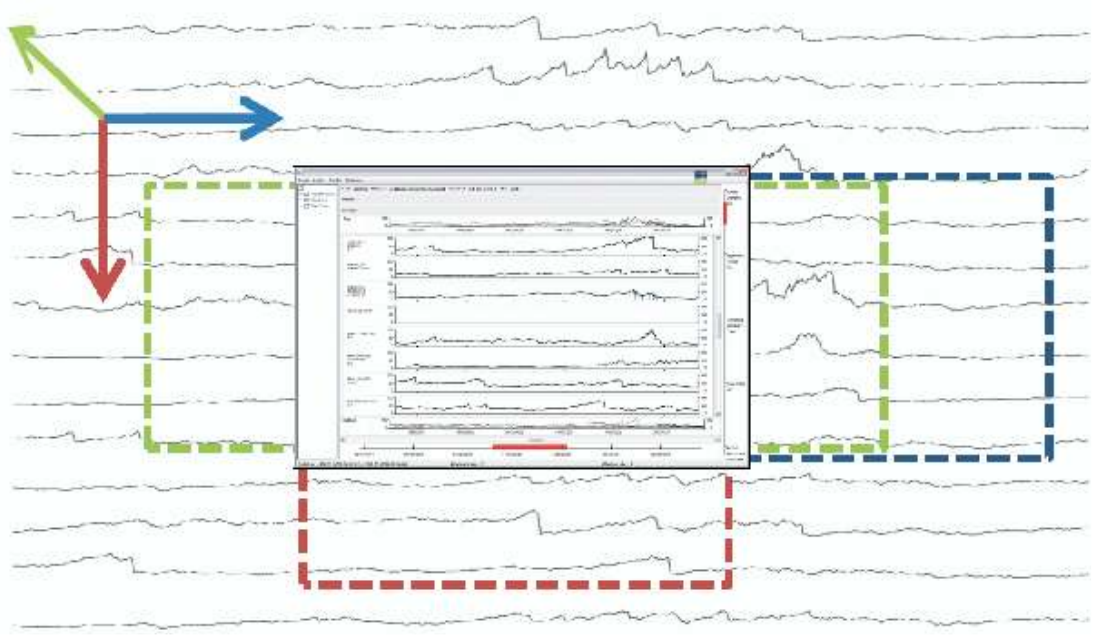
[Battle et al., Dynamic Prefetching of Data Tiles for Interactive Visualization, SIGMOD 2016]

Quadtree for Map Tiles

- Mercator projection
- Tree data structure (4 children)
- Interactive map demo:
  http://www.maptiler.org/google-maps-coordinates-tile-bounds-projection/

Data Tiles: Dynamic Prefetching

- Predicting which data tiles to load next
- Based on user interaction (panning, zooming, scrolling)
- Local caching of predicted data tiles



[Chan et al., VAST 2008]

Summary

- Perceptual scalability
  - Problem: Overplotting, visual clutter
  - Solution: Render models or aggregations instead of individual items
- Interactive scalability
  - Data queries
    - Problem: too much data to be queried with interactive response times
    - Solutions: incremental queries, aggregate queries and aggregate visualization
  - Rendering & pre-processing
    - Problem: too many graphical marks to be rendered for interactive frame rates
    - Solutions: data tiles with dynamic prefetching, HTML5 Canvas rendering

Further Reading

- Lui, Jiang & Heer: imMens: Real-Time Visual Querying of Big Data, EuroVis 2013
- Mwalongo et al.: State-of-the-Art Report in Web-based Visualization, EuroVis 2016
- Lampe and Hauser, Interactive Visualization of Streaming Data with Kernel Density Estimation, PacificVis 2011
- Mayorga and Gleicher, Splatterplots: Overcoming Overdraw in Scatter Plots, TVCG 2013
- Novotny and Hauser, Outlier-preserving Focus+Context Visualization in Parallel Coordinates, TVCG 2006
- Florek & Novotny, Interactive Information Visualization using Graphics Hardware, SCCG 2006
- Dix and Ellis: by chance enhancing interaction with large data sets through statistical sampling, AVI 2002
- Ellis and Dix: Enabling automatic clutter reduction in parallel coordinate plots, TVCG 2006
- Carr et al., Scatterplot Matrix for Large N, Journal of American Statistical Association, 1987
- Lins et al.: Nanocubes for real-time exploration of spatiotemporal datasets, TVCG 2013
- Wang et al.: Gaussian Cubes: Real-Time Modeling for Visual Exploration of Large Multidimensional Datasets, TVCG 2016
- Chan et al.: Maintaining interactivity while exploring massive time series, VAST 2008
- Piringer et al.: A multi-threading architecture to support interactive visual exploration, TVCG 2009
- McDonnel and Elmqvist: Towards Utilizing GPUs in Information Visualization: A Model and Implementation of Image-Space Operations, TVCG 2009
- Ren et al., Stardust: Accessible and Transparent GPU Support for Information Visualization Rendering, EuroVis 2017
- Battle et al., Dynamic Prefetching of Data Tiles for Interactive Visualization, SIGMOD 2016

Adapted from Real-Time Visualization VU, TU Wien

HTML5 Visualization

- HTML5 `<canvas>` element
  - Container for graphics
  - Graphics generated by JavaScript
- HTML markup:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- 2D context example (circle)

```
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
ctx.stroke();
```

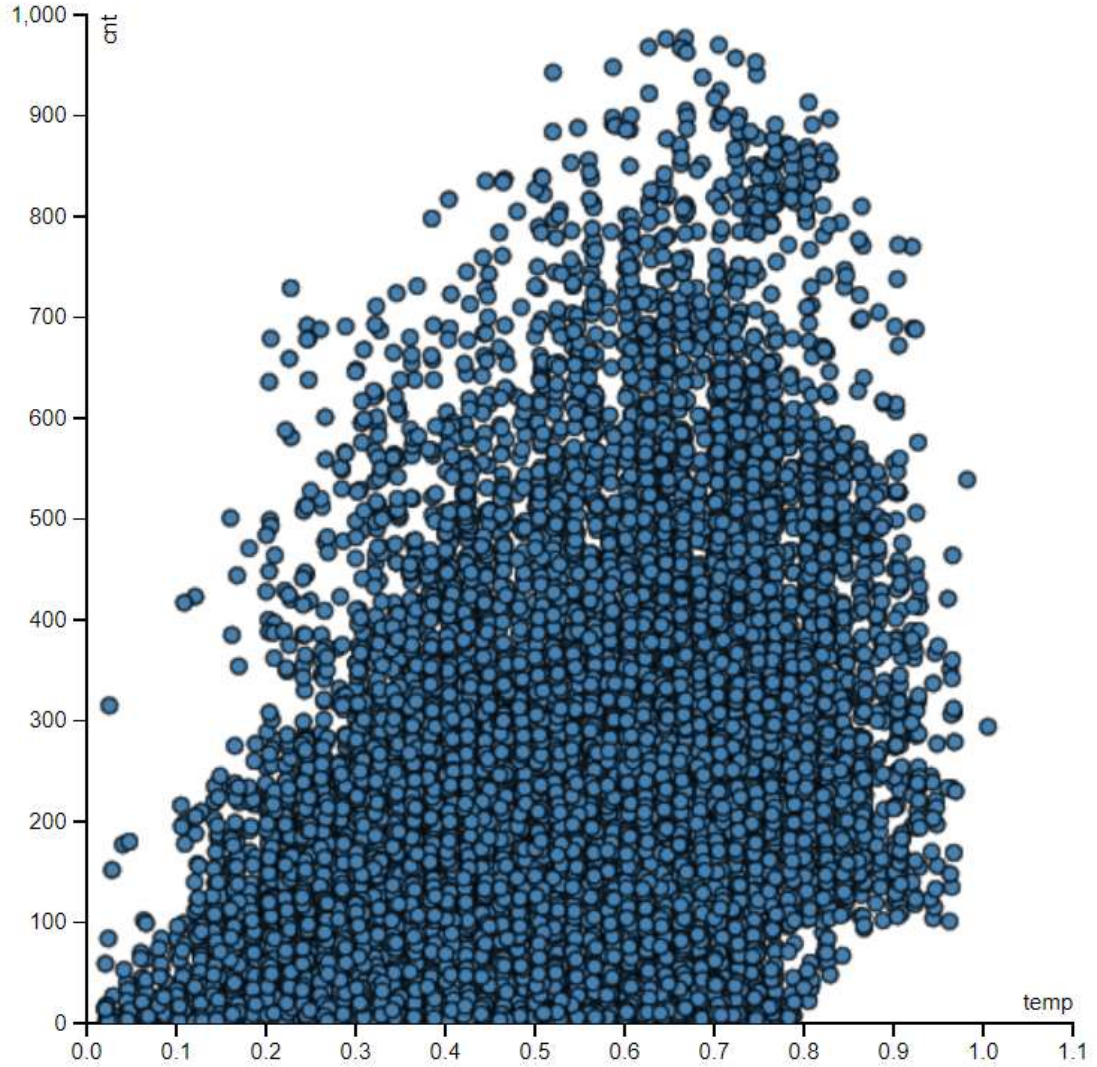

Example: Bike Rental Data

```
function renderDotsCanvas2D(){
    context.clearRect(0, 0, width, height);
    context.fillStyle = "steelblue";
    context.strokeWidth = 1;
    context.strokeStyle = "black";

    data.forEach(function(d){
        var cx = x(xDim(d));
        var cy = y(yDim(d));

        context.beginPath();
        context.arc(cx, cy, radius, 0, 2 * Math.PI);
        context.closePath();
        context.fill();
        context.stroke();
    });
}
```

- Looping over all items required
- SVG axes → alignment of SVG and canvas required
- No event item-based handlers



Example: Bike Rental Data - Tooltips

CSS

```css
div.tooltip {
    position: absolute;
    text-align: center;
    width: 120px;
    height: 30px;
    padding: 2px;
    font: 12px sans-serif;
    background: lightsteelblue;
    border: 0px;
    border-radius: 8px;
    pointer-events: none;
}
```

d3

```
var tooltip = d3.select("body").append("div")
    .attr("class", "tooltip")
    .style("opacity", 0);
```

Event listener

Set visibility, text, and position

d3 + SVG

```
dot.attr("class", "dot")
    .attr("r", 3.5)
    .attr("cx", function(d) { return x(xDim(d)); })
    .attr("cy", function(d) { return y(yDim(d)); })
    .style("fill", "steelblue")
    .on("mouseover", function(d){
        tooltip.transition().duration(50)
            .style("opacity", 0.9);
        tooltip.html("Season: " + d['season'] + "<br> Windspeed: " + d['windspeed'])
            .style("left", (d3.event.pageX) + "px")
            .style("top", (d3.event.pageY - 28) + "px");
    })
      .on("mouseout", function(d) {
        tooltip.transition()
            .duration(50)
            .style("opacity", 0);
      });
```

Example: Bike Rental Data - Tooltips

```
var clickData = data.map(function(d, i) {
      return {
          x: x(xDim(d)),
          y: y(yDim(d)),
          d: d
      };
  });
quadTree = d3.geom.quadtree(clickData);
canvas.on("click", onClick);
```

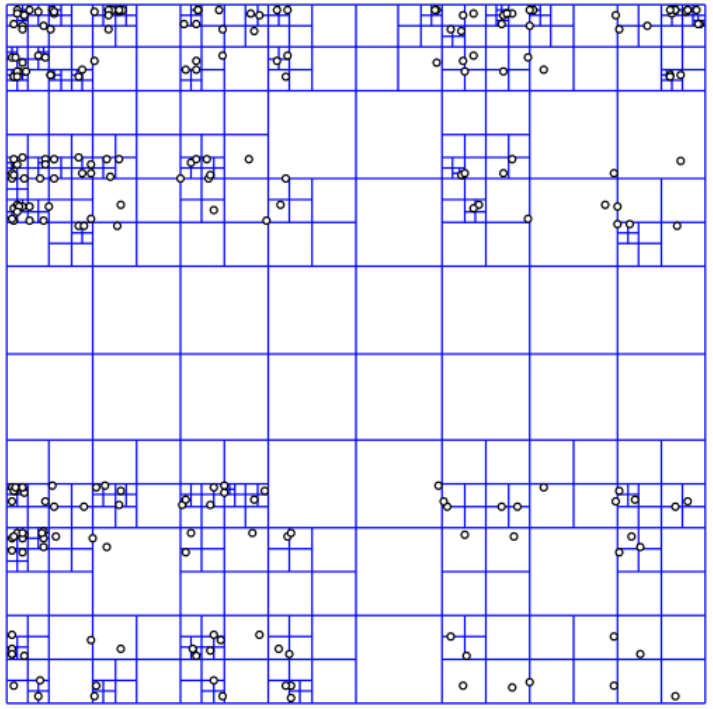
```
function onClick() {
    var mouse = d3.mouse(this);

    // find the closest point in the dataset to the clicked point
    var closest = quadTree.find([mouse[0], mouse[1]]);

    // register the click if the clicked point is in the radius of the point
    var distance = euclideanDistance(mouse[0], mouse[1], closest.x, closest.y);

    if(distance < radius){
        tooltip.transition().duration(50)
               .style("opacity", 0.9);
          tooltip.html("Season: " + closest.d['season'] + "<br> Windspeed: " + closest.d['windspeed'])
                 .style("left", (d3.event.pageX) + "px")
                 .style("top", (d3.event.pageY - 28) + "px");
    }
    else{
        tooltip.transition()
                 .duration(50)
                 .style("opacity", 0);
    }
}
```

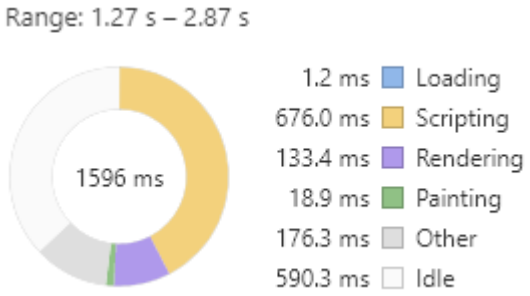
https://bl.ocks.org/ejb/e2da5a23e9a09d494bd532803d8db61c



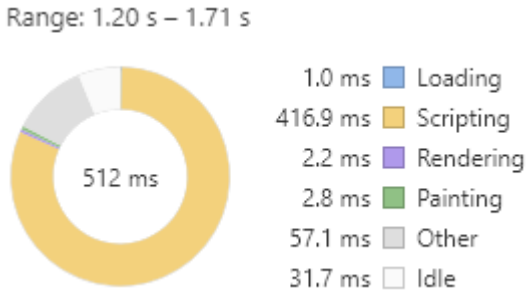
[Wikipedia]

Example: Bike Rental Data

SVG



Canvas 2D



[Chrome 70, d3.v3]

HTML5 Visualization

- HTML5 `<canvas>` element
  - Container for graphics
  - Graphics generated by JavaScript
- HTML markup:

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

- WebGL example

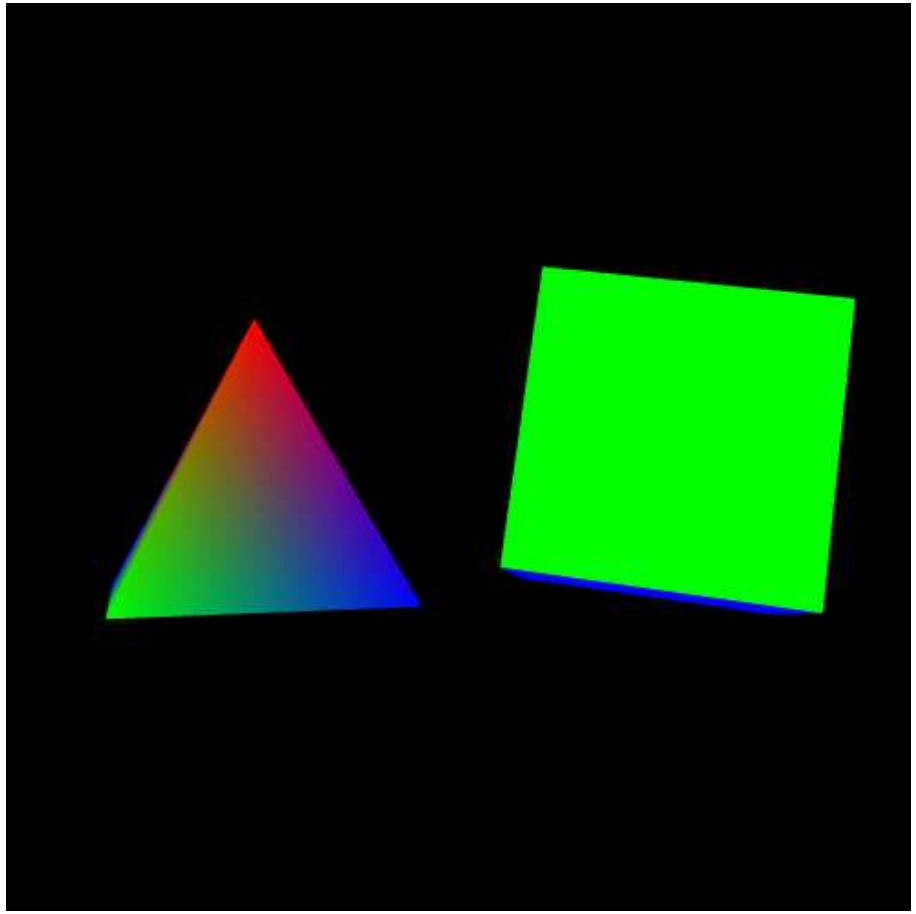
```
var c = document.getElementById("myCanvas");
var gl = c.getContext(„webgl");
gl.clearColor(0.0, 0.0, 0.0, 1.0);
...
```


WebGL

- Low-level graphics API for the web
- JavaScript implementation of OpenGL ES 2.0
- Shader-based (GLSL)
- Exposed through HTML5 canvas element
- Plugin-free supported by Chrome, Firefox, Opera & Safari

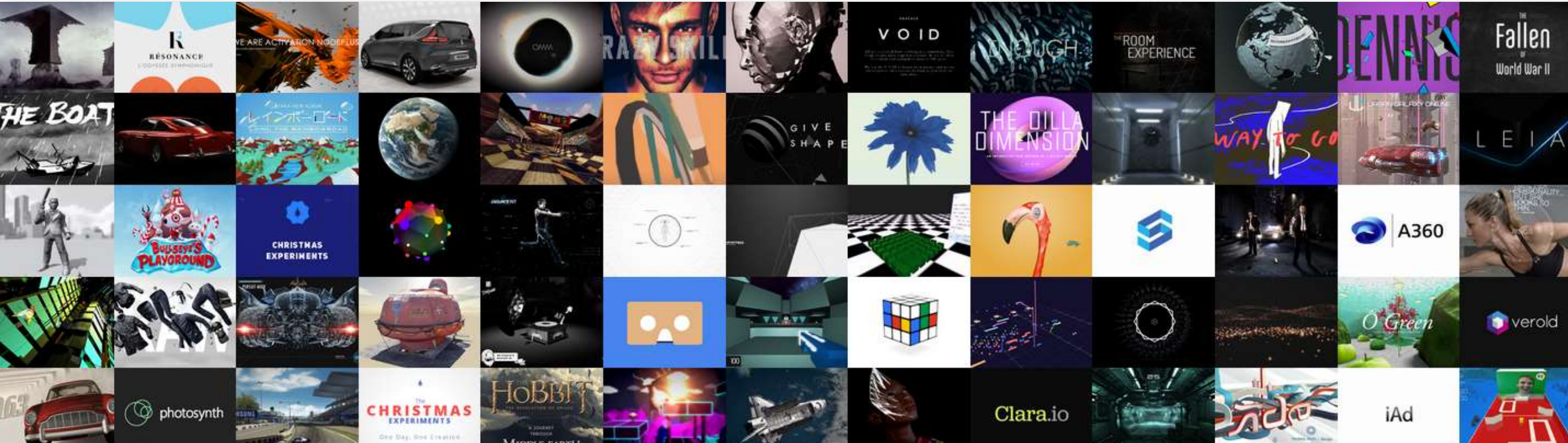
WebGL

- Example: http://learningwebgl.com/blog/?p=370

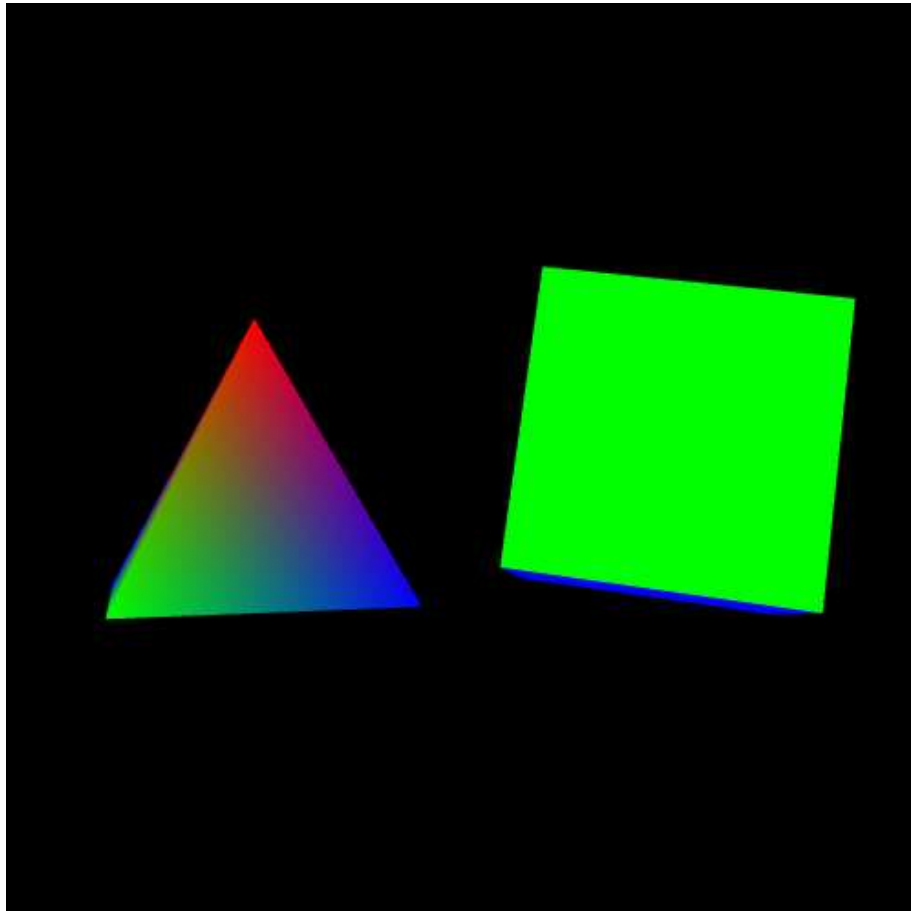


THREE.js

- Lightweight JavaScript 3D library
- Canvas, SVG, CSS3D, and **WebGL** renderers
- Scene graph
- http://threejs.org/



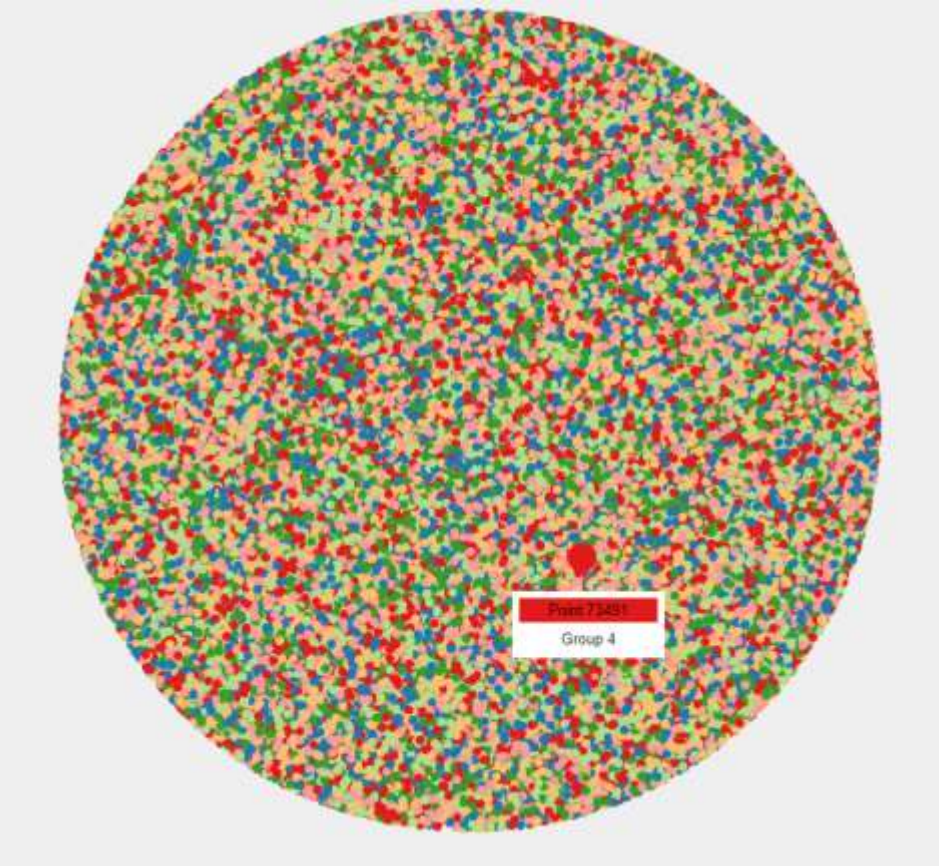
Example adopted from: http://www.johannes-raida.de/tutorials/three.js/tutorial05/tutorial05.htm#liveexample



Example: 100,000s points in THREE.js

- Circular sprites (texture with alpha channel)
- Zooming using THREE.PerspectiveCamera and d3-zoom
- Picking using THREE.Raycaster

https://beta.observablehq.com/@grantcuster/using-three-js-for-2d-data-visualization



https://codepen.io/GrantCuster/full/QQXRqj/