Study Materials for Lecture 8

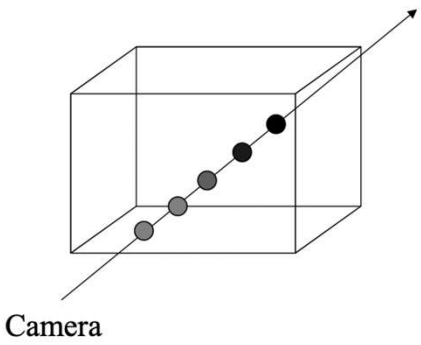
- Parallel Algorithms:
  - https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial
  - Parallel Volume Rendering Using Binary Swap Image Composition, Ma et al.
  - Parallel Volume Rendering on the IBM Blue Gene/P, Peterka et al.

Accelerating Volume Rendering

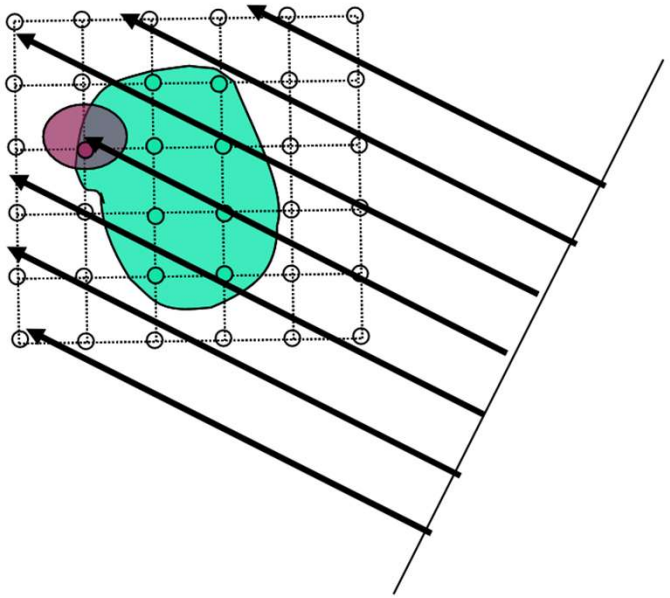
- Early ray termination
- Empty space skipping
- Adaptive sampling

Early Ray Termination

- In front to back compositing, we keep track of accumulated opacities separately
- We can stop ray traversal when accumulated alpha/opacity for a ray reaches 1 and nothing behind will be visible

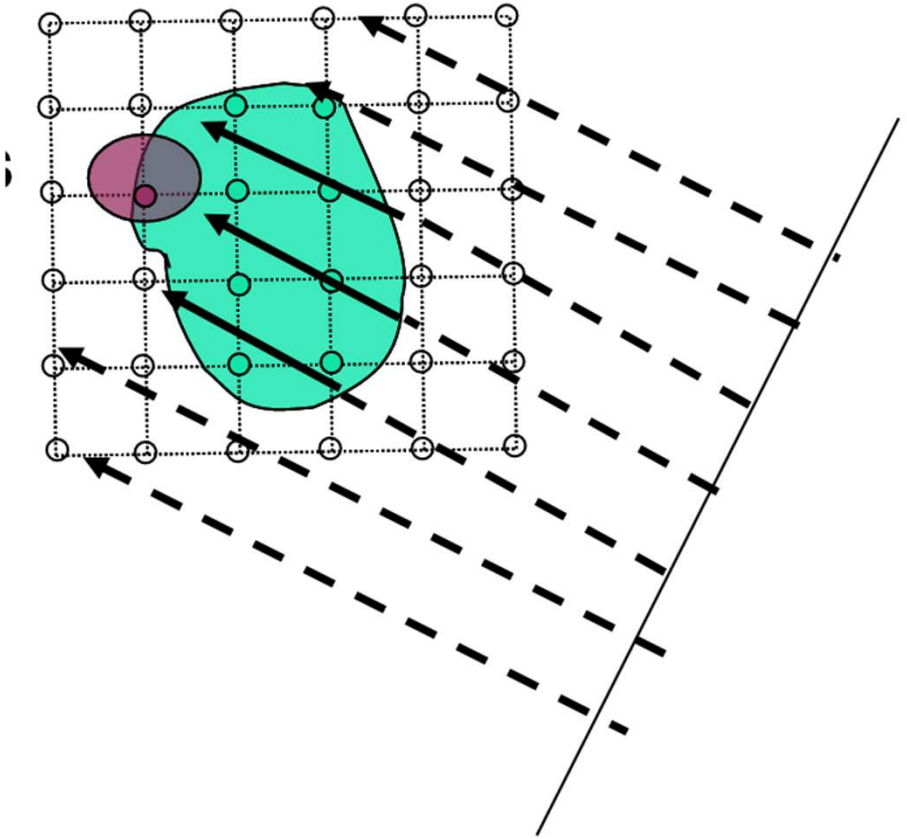


Current Alpha	Accumulated Alpha
0.2	0.2
0.4	0.4+(1- 0.4)*0.2=0.52
0.5	0.5+(1- 0.5)*0.52=0.76
0.4	0.4+(1- 0.4)*0.76=0.86
0.6	0.6+(1- 0.6)*0.86=0.94



Empty Region Skipping

- Skip Empty Cells
- Homogeneity acceleration
  - Approximate homogeneous regions with fewer sample points



Adaptive Sampling

- Increase sample density in high-gradient regions
- Decrease sample density in low-gradient/homogeneous regions

Dealing with Large-Scale Data via Parallel Data Processing and Visualization

Why Parallel Programming?

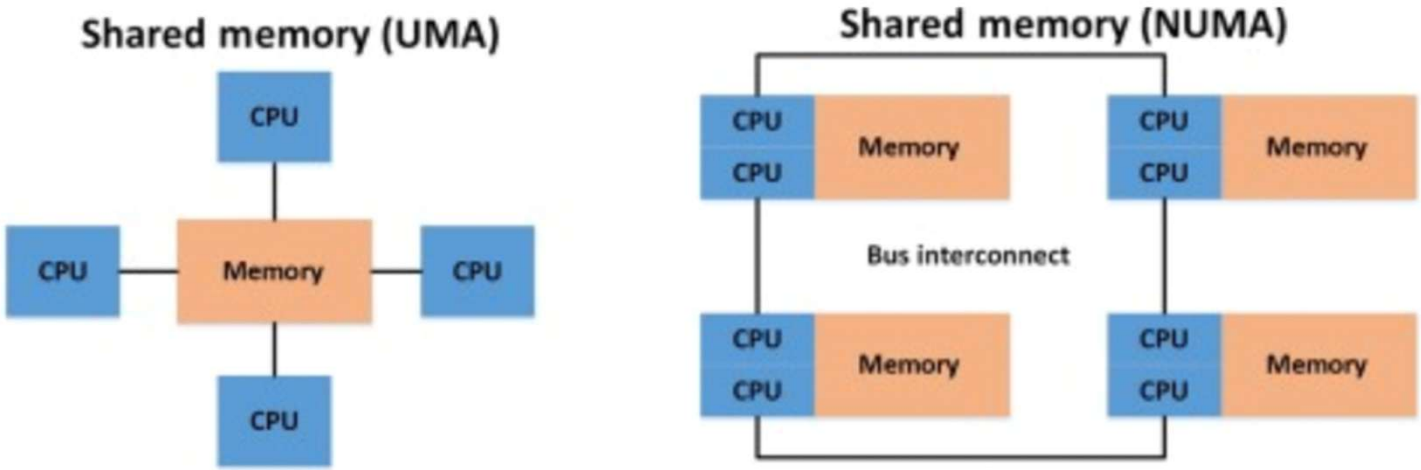
- Utilize the aggregated memory from all computing elements to store very large data sets
- Utilize the aggregated compute power to share the expensive task workload
- Utilize the aggregated I/O bandwidth, if a parallel file system is available
- Data computed from large scale simulations are typically stored in a file system connected to a supercomputer

Parallel Computing Paradigms

- Shared-memory parallelism
- Distributed-memory parallelism
- Hybrid parallelism

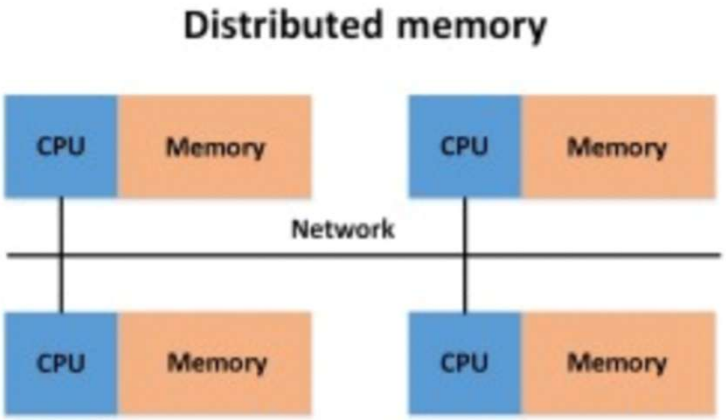
Shared Memory Parallelism

- Shared-memory parallelism
  - All PEs can access the shared memory, hence have the same view of the data
  - No explicit message passing is needed
  - Computation are done through multiple threads, or libraries such as OpenMP



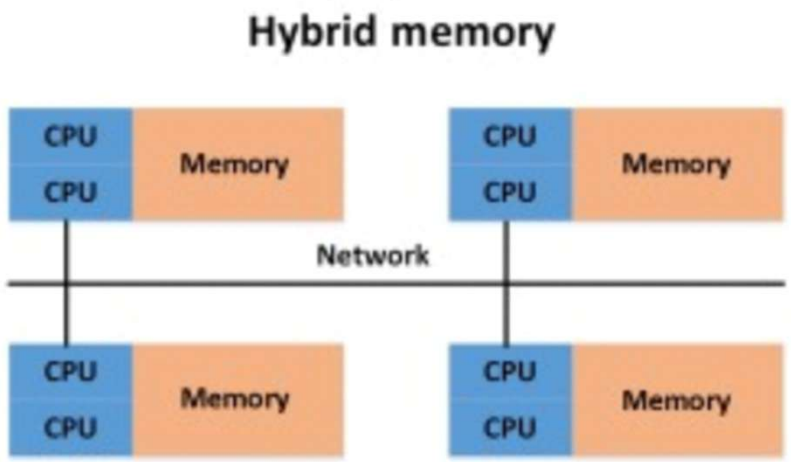
Distributed Memory Parallelism

- Distributed-memory parallelism
  - Data are distributed across the local memory of different Processing Element (PE)
  - Coordination among PEs is done through message passing using libraries such as MPI
  - Tasks are run on each core within each processing node



Hybrid Parallelism

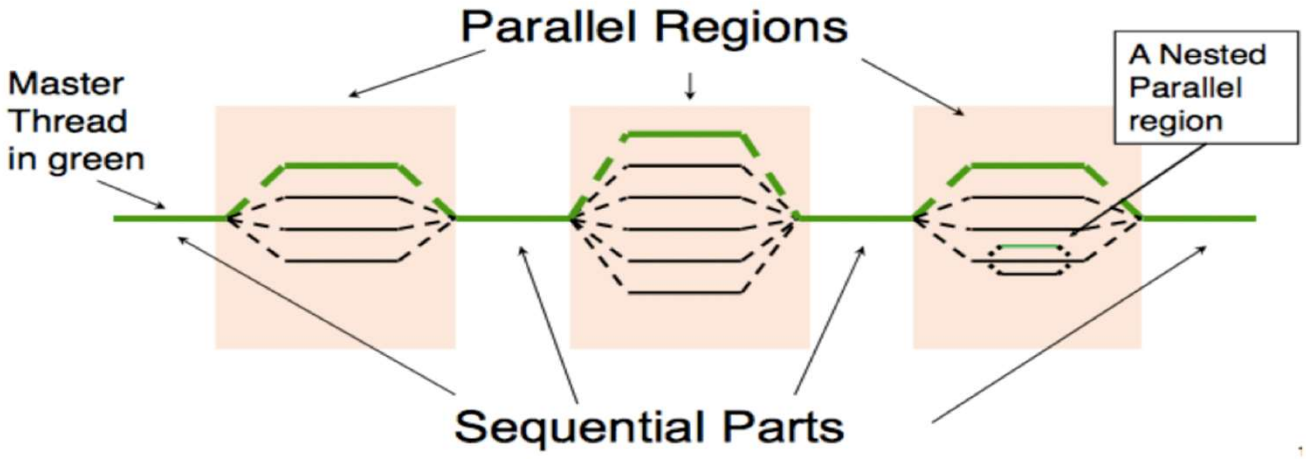
- Hybrid parallelism
  - Hybrid parallelism refers to a blend of distributed- and shared-memory parallel programming techniques within a single application
  - Each node has multiple threads running shared memory parallelism, but nodes have distributed memory and communicate with one another using message passing
  - Most modern clusters and supercomputers provide hybrid parallelism capability



Parallel Programming API – Shared Memory

- Open Multi-Processing (OpenMP) - Shared Memory Parallelism
- Latest OpenMP versions also supports GPUs

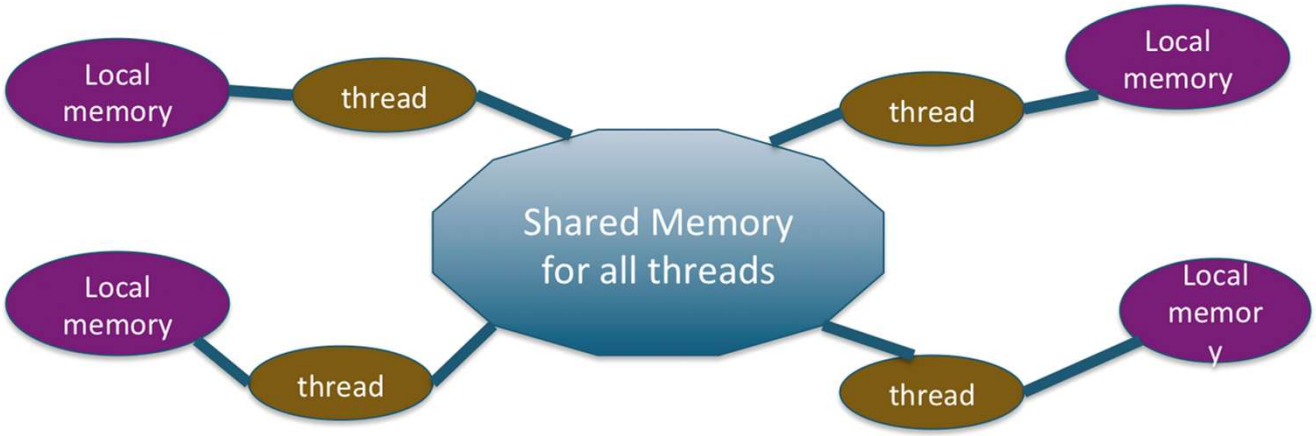
  https://www.openmp.org/



Parallel Programming API – Shared Memory

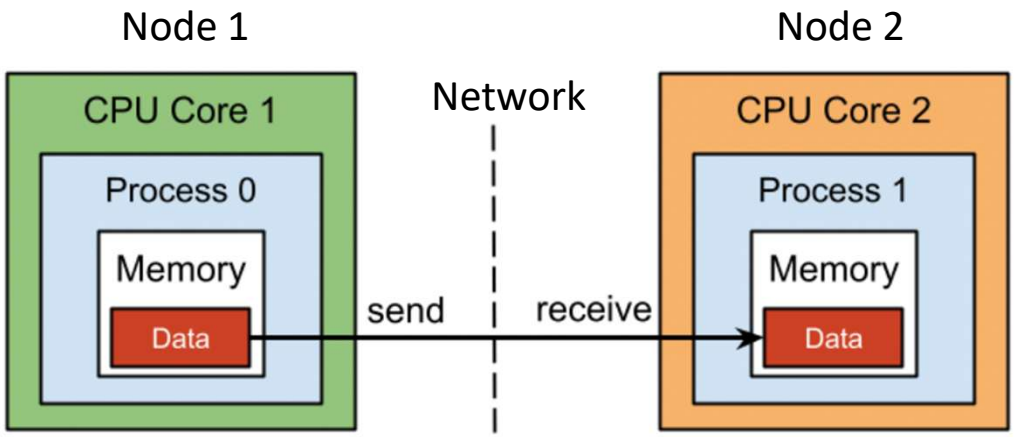
- Open Multi-Processing (OpenMP) - Shared Memory Parallelism
- Latest OpenMP versions also supports GPUs

  https://www.openmp.org/



Parallel Programming API – Distributed Memory

- Message Passing Interface (MPI) – Distributed Parallelism



Param Sanganak at IITK

Total number of nodes: 332 (20 + 312)

o Service nodes: 20 (Master+ Login+ Service+ Management)
o CPU only nodes: 150 (Total 7200 cores, 28800GB memory)
o GPU nodes: 20
o High Memory nodes:78

- Different implementations of MPI:
  - Open MPI (https://www.open-mpi.org/)
  - MVAPICH (https://mvapich.cse.ohio-state.edu/)
  - MPICH (https://www.mpich.org/)

OpenMP vs MPI

MPI	OpenMP
Available from different vendors and gets compiled on Windows, macOS, and Linux operating systems.	An add-on in a compiler such as a GNU compiler and Intel compiler.
Supports parallel computation for distributed-memory and shared-memory systems.	Supports parallel computation for shared-memory systems only.
A process-based parallelism.	A thread-based parallelism.
With MPI, each process has its own memory space and executes independently from the other processes.	With OpenMP, threads share the same resources and access shared memory.
Processes exchange data by passing messages to each other.	There is no notion of message-passing. Threads access shared memory.
Process creation overhead occurs one time.	It depends on the implementation. More overhead can occur when creating threads to join a task.

OpenMP vs MPI: Hello World

```c
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[])
{
    int nthreads, tid;

    #pragma omp parallel private(nthreads, tid)
    {
        tid = omp_get_thread_num();
        nthreads = omp_get_num_threads();
        printf("Hello World Thread %d / %d\n", tid, nthreads);


    }

    return 0;
}
```

```
Hello World Thread 1 / 2
Hello World Thread 2 / 2
```

```c
#include <mpi.h>
#include <stdio.h>

int main(int argc, char** argv) {
    // Initialize the MPI environment
    MPI_Init(NULL, NULL);

    // Get the number of processes
    int world_size;
    MPI_Comm_size(MPI_COMM_WORLD, &world_size);

    // Get the rank of the process
    int world_rank;
    MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);

    // Get the name of the processor
    char processor_name[MPI_MAX_PROCESSOR_NAME];
    int name_len;
    MPI_Get_processor_name(processor_name, &name_len);

    // Print off a hello world message
    printf("Hello world from processor %s, rank %d out of %d processors\n",
           processor_name, world_rank, world_size);

    // Finalize the MPI environment.
    MPI_Finalize();
}
```

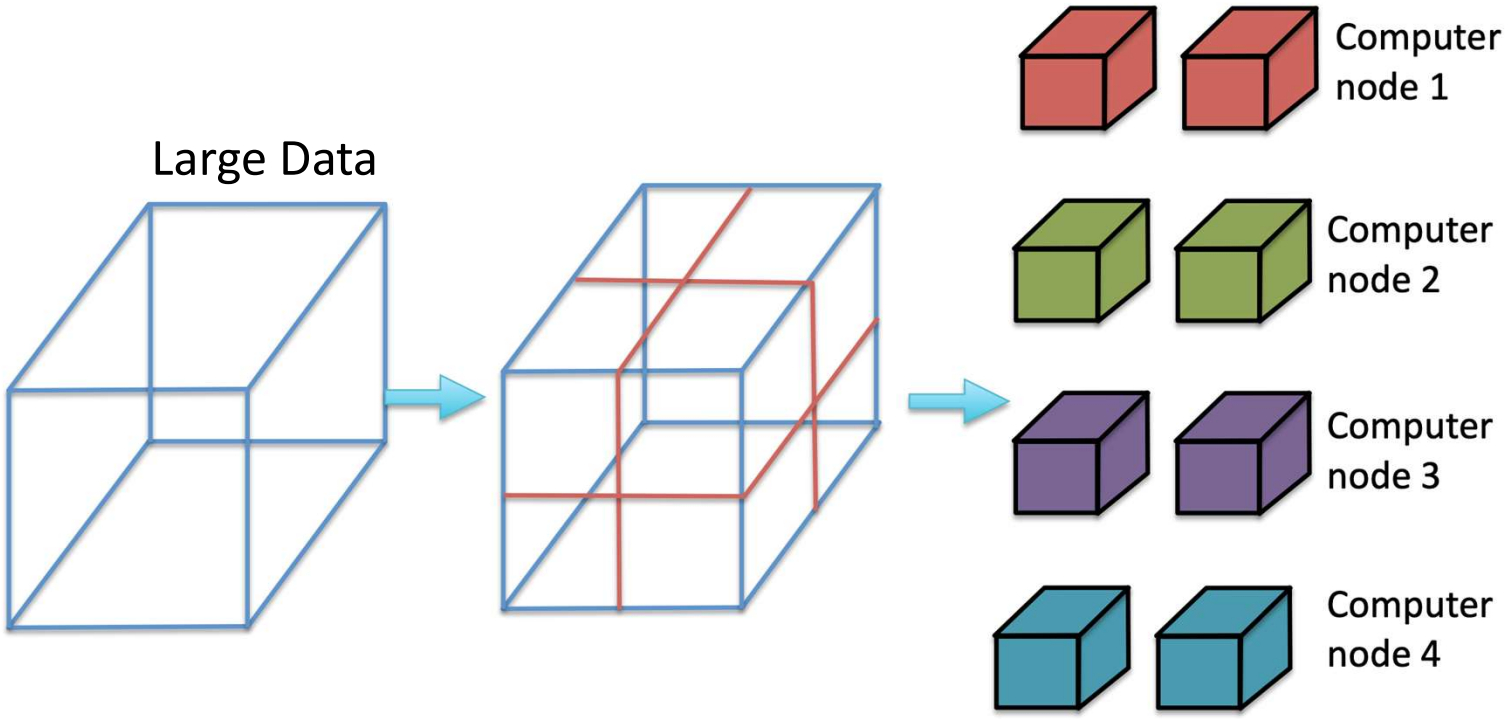
```
Hello world from processor ea72304b6e5c, rank 0 out of 4 processors
Hello world from processor ea72304b6e5c, rank 1 out of 4 processors
Hello world from processor ea72304b6e5c, rank 2 out of 4 processors
Hello world from processor ea72304b6e5c, rank 3 out of 4 processors
```

Steps for a Parallel Distributed Algorithm

- Data Decomposition
- Distribute decomposed data to processors
- Parallelly process data and apply the algorithm
- Aggregate partial results from different processors to produce the final result

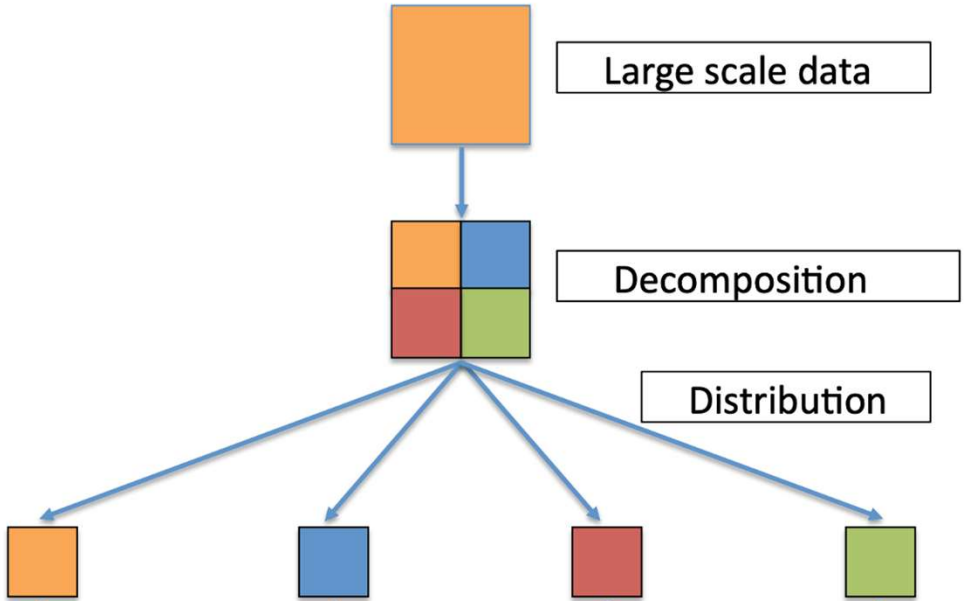
Data Decomposition

After decomposition, data blocks are distributed to processors



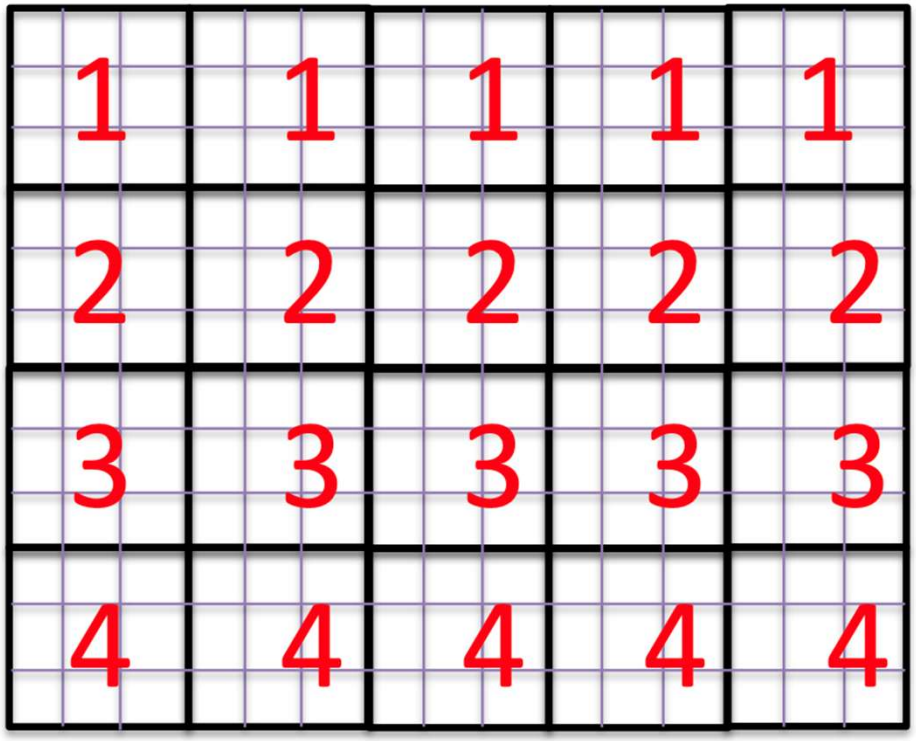
Data Distribution

- After decomposition, data blocks need to be distributed to processors
- Proper distribution is critical for the overall performance and minimizing overhead
- What overhead?
  - Load imbalance
  - Communication of data over network among processors
    - Synchronization



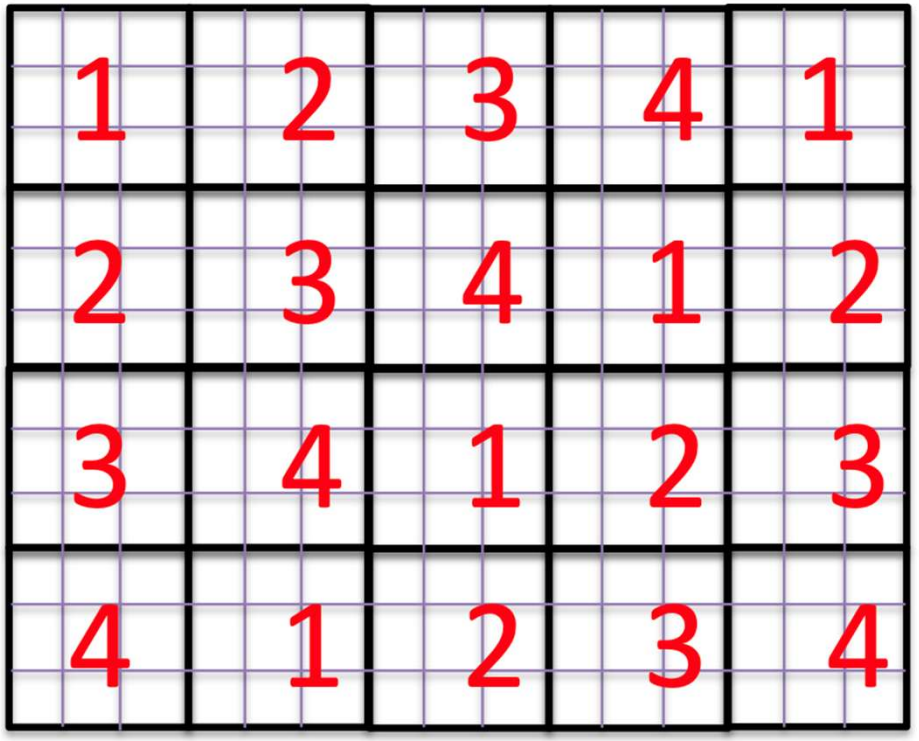
Data Distribution Techniques

- Contiguous distributions



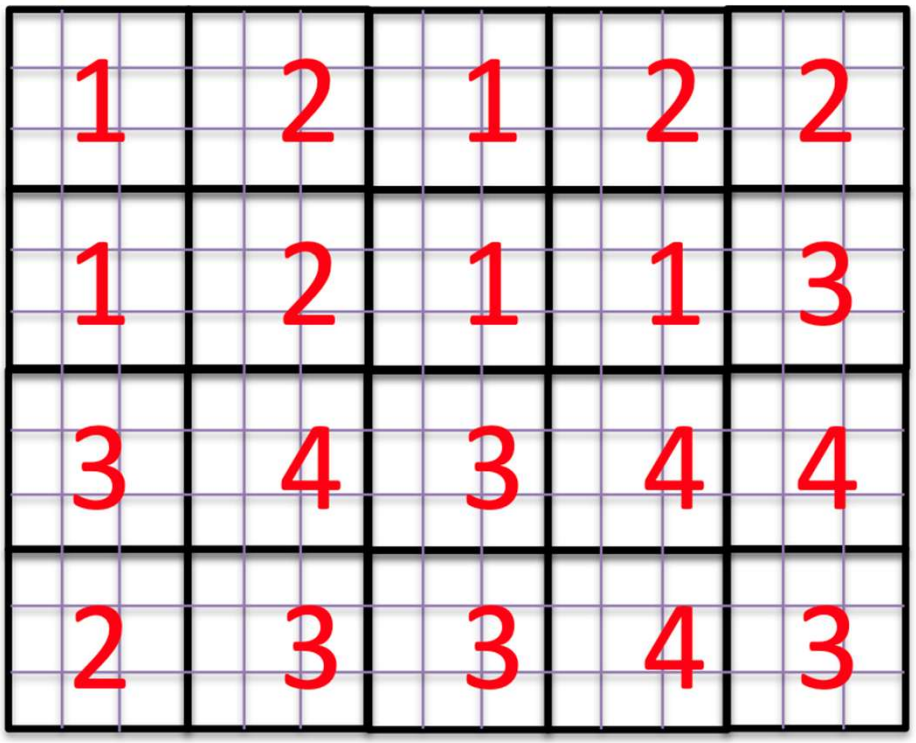
Data Distribution Techniques

- Block cyclic (round robin) distribution

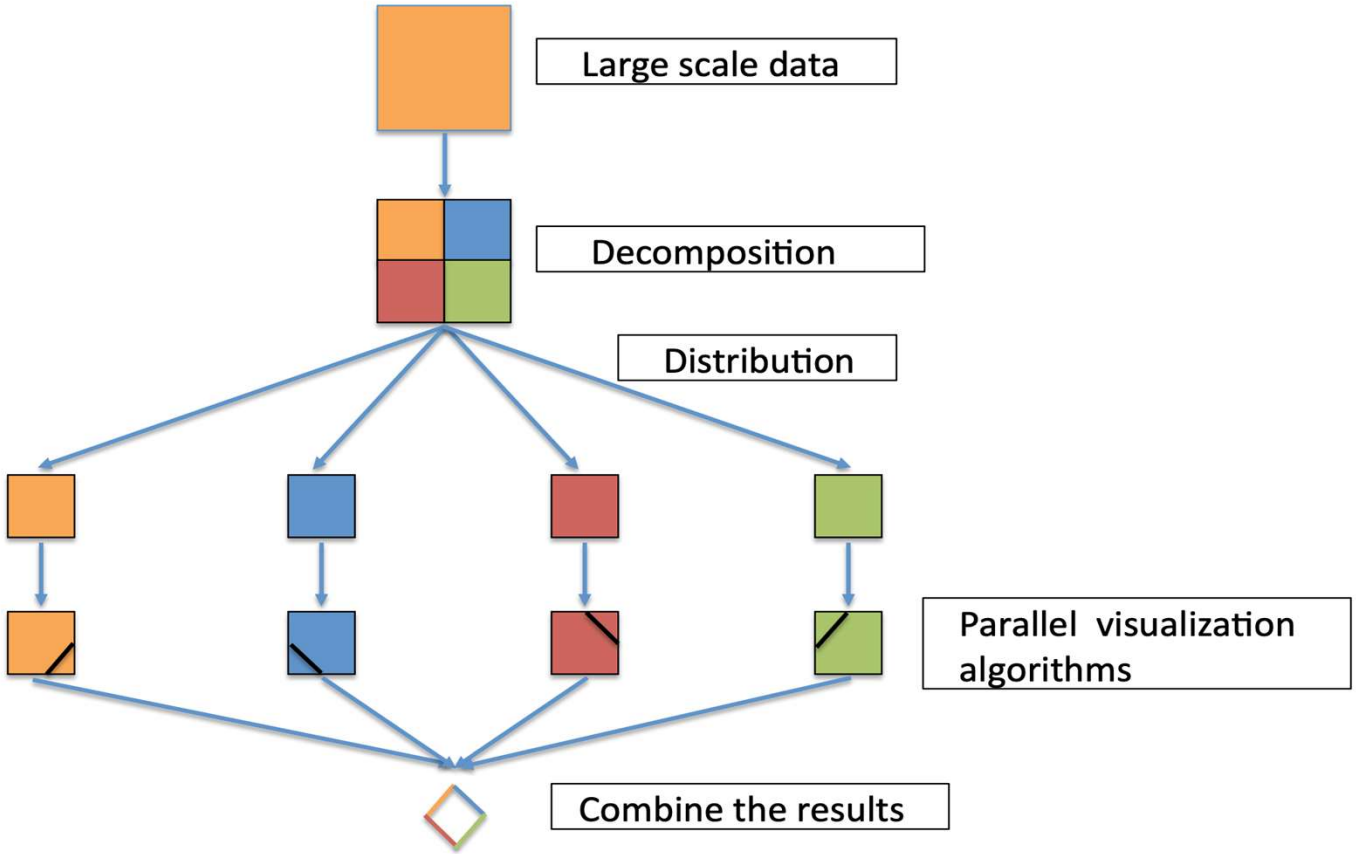


Data Distribution Techniques

- Workload aware

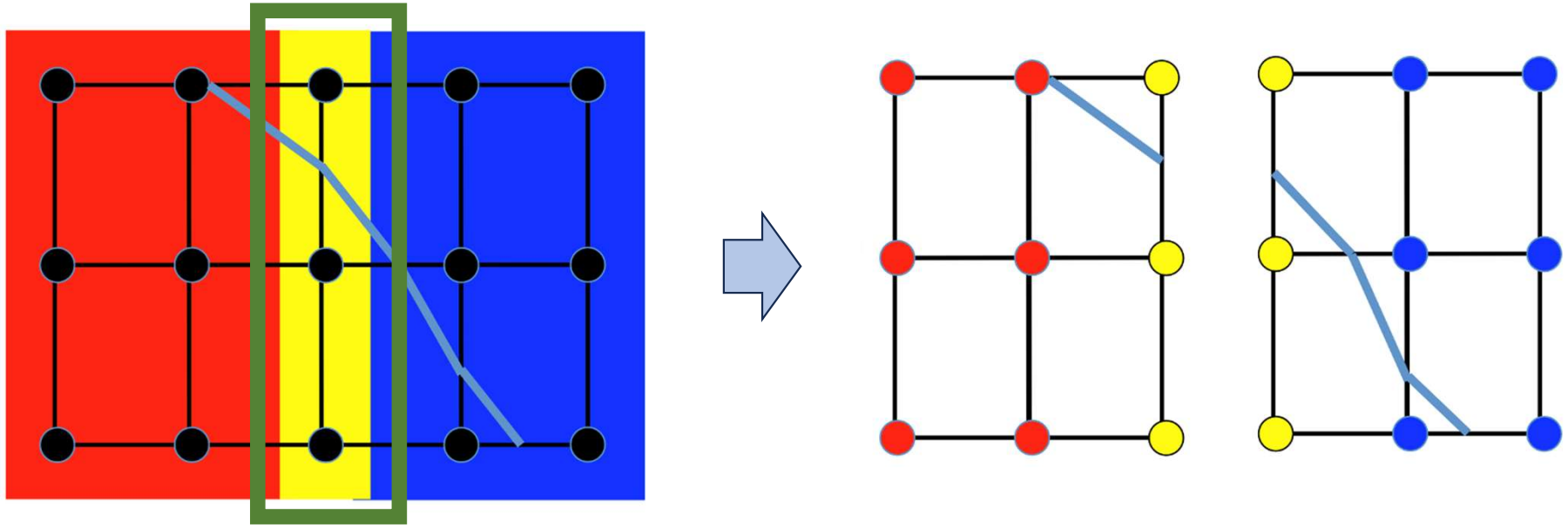


Parallel Data Processing and Aggregation of Partial Results



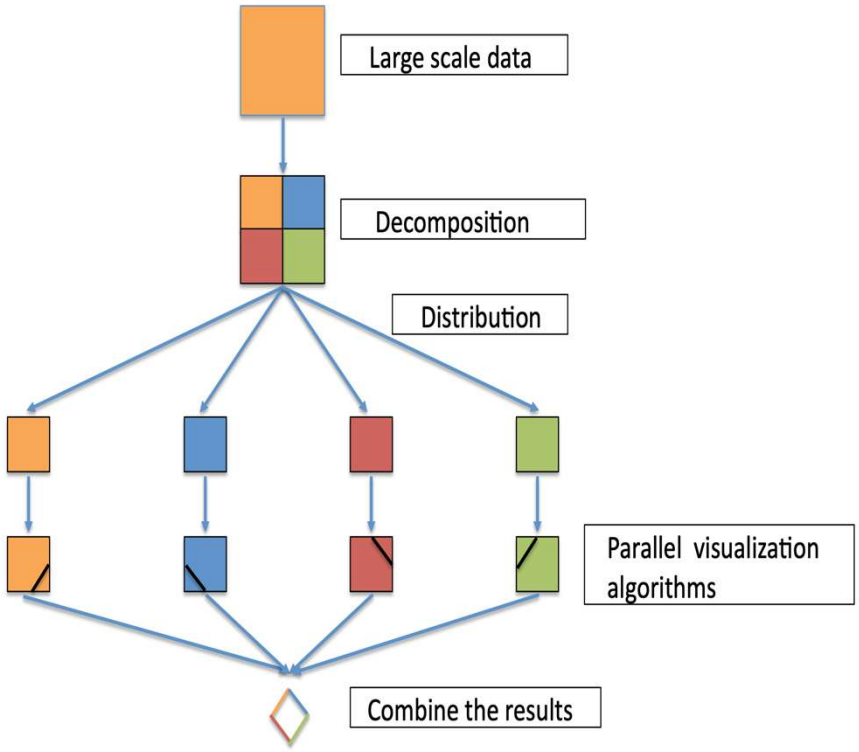
How to Ensure Continuity at Boundaries

- Need ‘Ghost cells’
  - For interpolation-based tasks, we need to replicate and share boundary data points with neighboring processors
  - Cells in the duplicated layers are called ghost cells



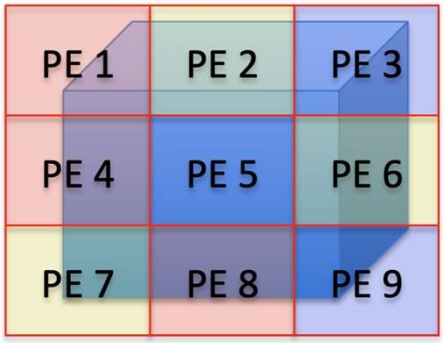
Parallel Isosurface Extraction

- Parallel Marching Cubes (Squares)
- Observation: Each cell can be processed in parallel as there is no dependency among cells
- Steps:
  - Divide the data into chunks of equal size
  - Each processing element (processor) runs Marching Cubes for the cells in the chucks that are assigned to it
  - Return the resulting triangles back to a master node, or render the triangles if parallel rendering is desired



Parallel Volume Rendering

- Parallel Volume Rendering
  - Parallel by pixel – Fits better for Shared memory approach

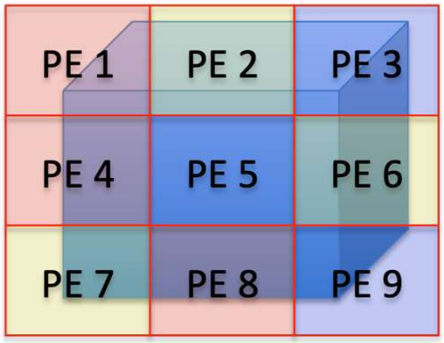


Parallel by pixel

- No image compositing needed
- Little/no communication
- Data often need to be replicated
- Difficult to scale to large data

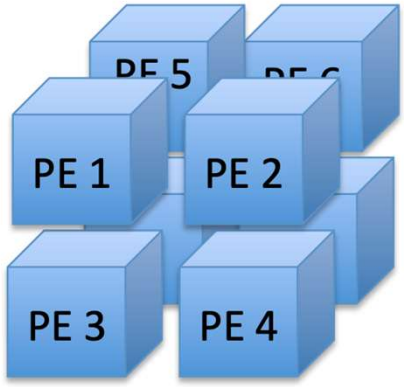
Parallel Volume Rendering

- Parallel Volume Rendering
  - Parallel by pixel – Fits better for Shared memory approach
  - Parallel by data – Fits better for Distributed Memory approach



Parallel by pixel

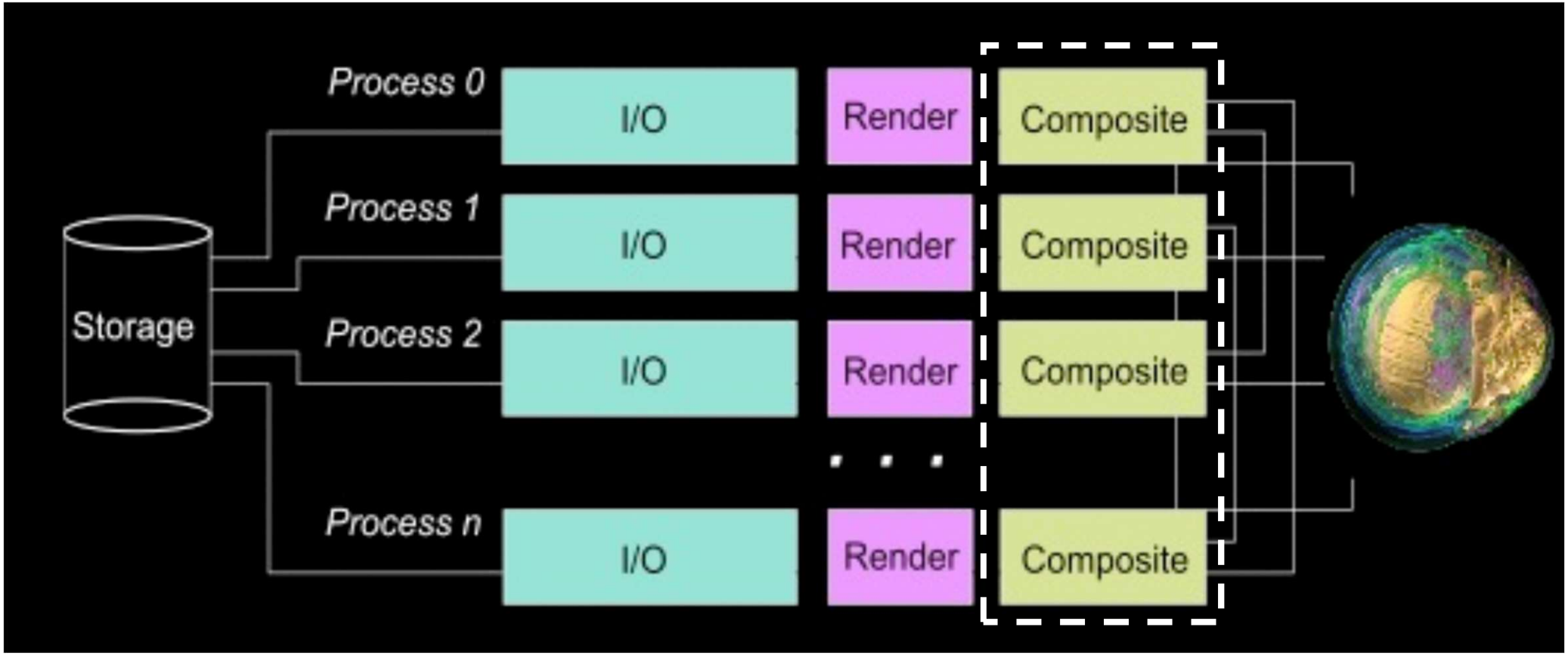
- No image compositing needed
- Data may need to be replicated
- Difficult to scale to large data



Parallel by data

- Image compositing needed
- More communication needed
- Data are distributed
- Easy to scale to large data

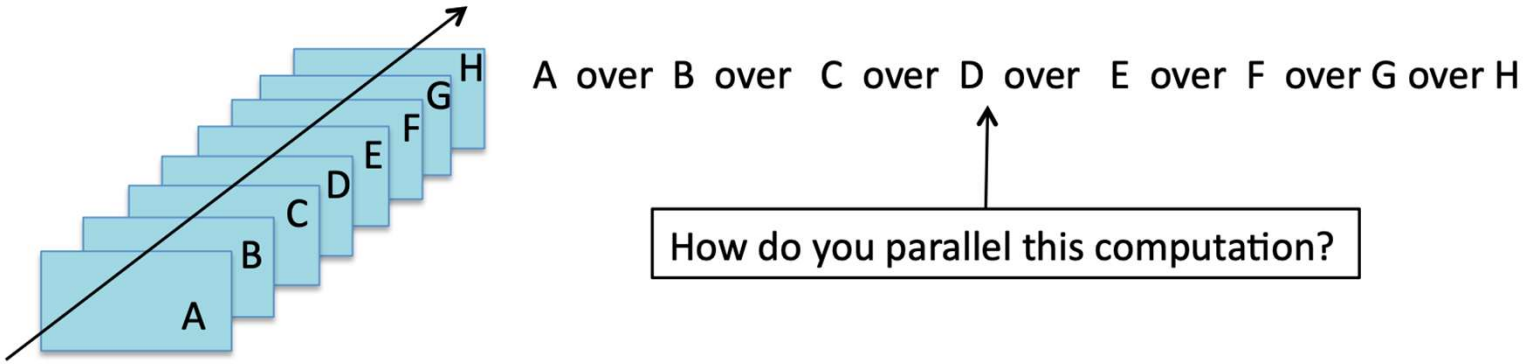
Parallel Volume Rendering



How to do this compositing in parallel efficiently?

Parallel Image Compositing for Volume Rendering

- Composite partial images generated from sub-domains
  - Minimize communication: send and receive images
- Compositing needs to follow the visibility order of ray casting technique



Parallel Image Compositing for Volume Rendering

A over B = AB | C over D = CD | E over F= EF | G over H = GH

(Requires 4 PEs)

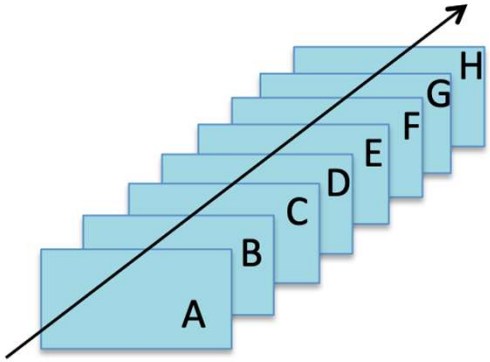
AB over CD = ABCD | EF over GH = EFGH

(Requires 2 PEs)

Load balancing is not good: Half of the processors are busy at every new stage

ABCD over EFGH = ABCDEFGH

(Requires 1 PE)

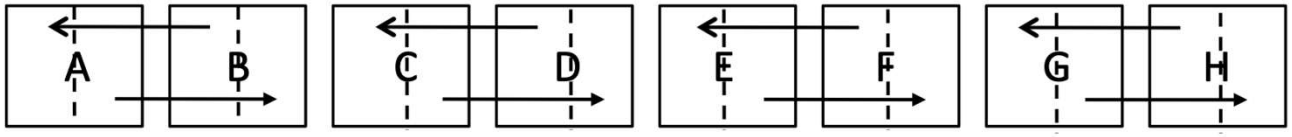


A over B over C over D over E over F over G over H

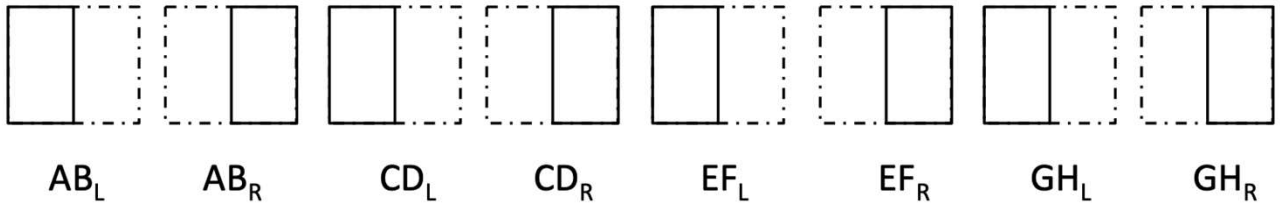
How do you parallel this computation?

Parallel Image Compositing for Volume Rendering: A better strategy

- keep every processor (PE) busy all the time
- Recursive halving

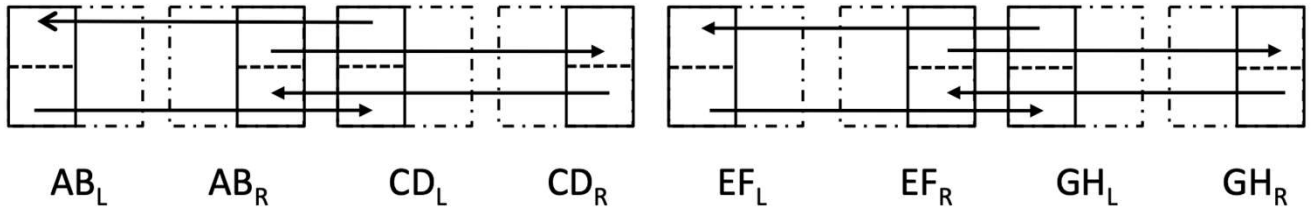


Divide each image into two halves, and give half to the other PE to composite

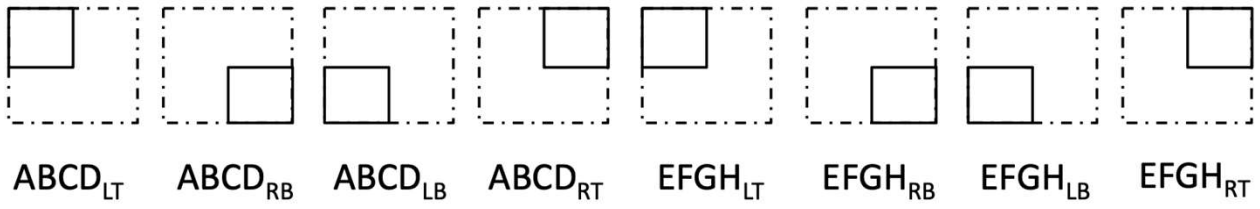


Parallel Image Compositing for Volume Rendering: A better strategy

- keep every processor (PE) busy all the time
- Recursive halving

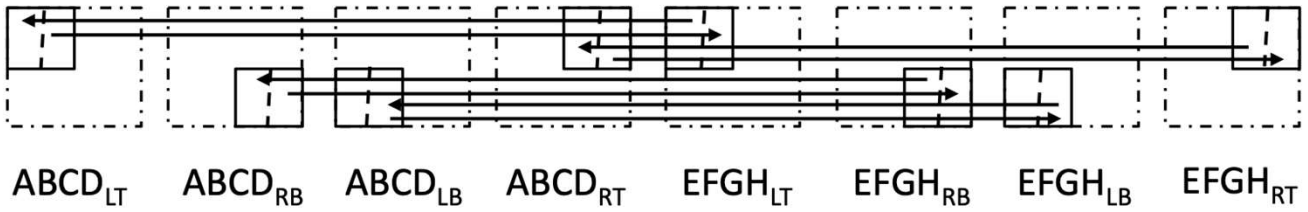


Divide the half image further into top and bottom halves, and give it to the corresponding PE of to composite



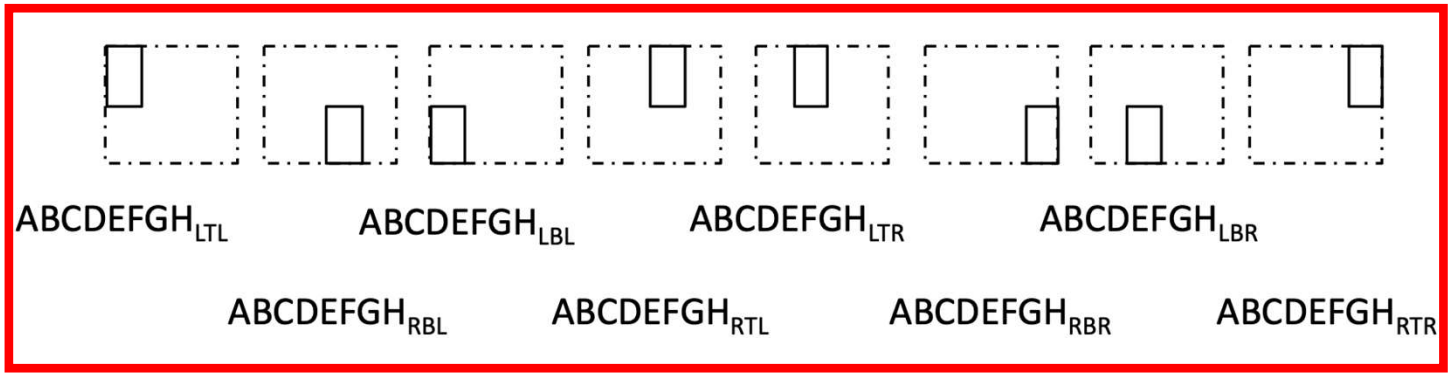
Parallel Image Compositing for Volume Rendering: A better strategy

- keep every processor (PE) busy all the time
- Recursive halving



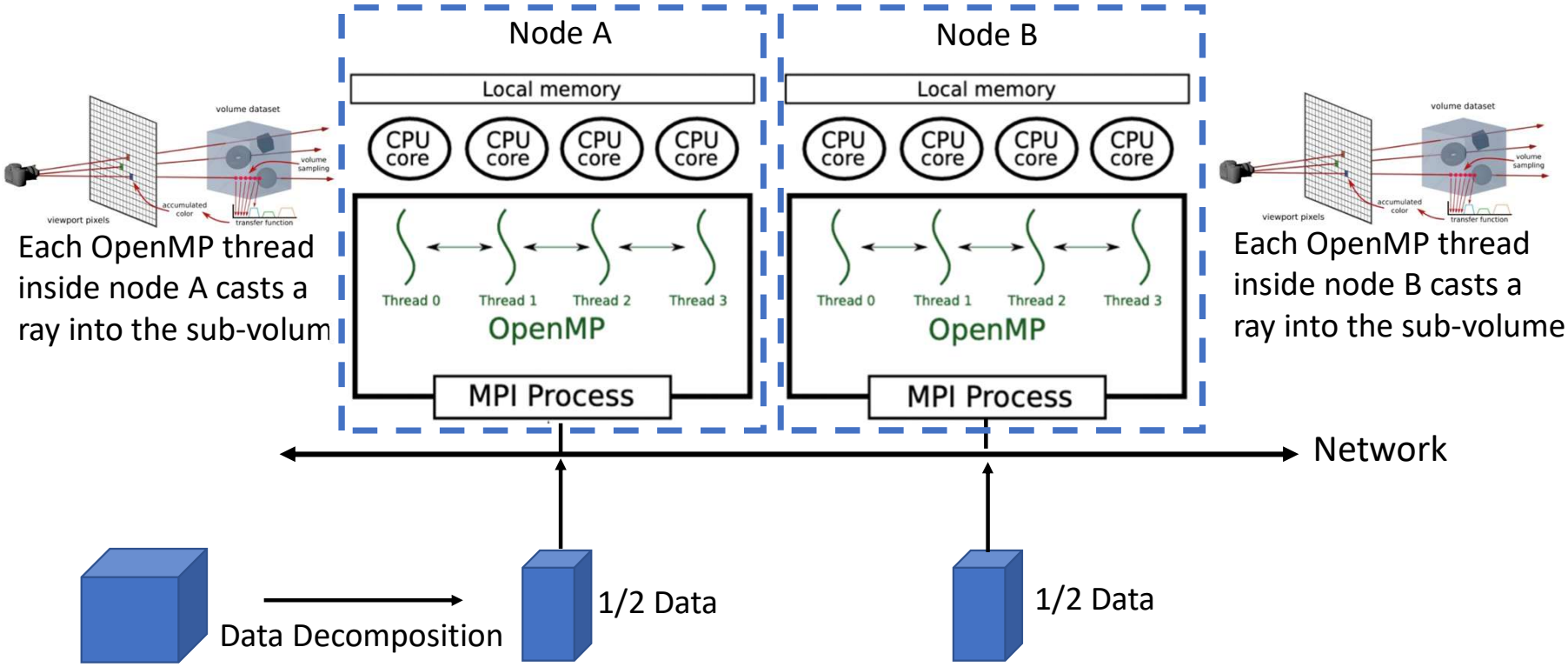
Divide the 1/4 image further into left and right halves, and give it to the corresponding PE of to composite

Load balanced approach: all workers are busy with equal workload at all the time



Hybrid Parallel Volume Rendering

- Use MPI + OpenMP together to extract fine grained parallelism for very large-scale data set



How to Say Nothing with Scientific Visualization

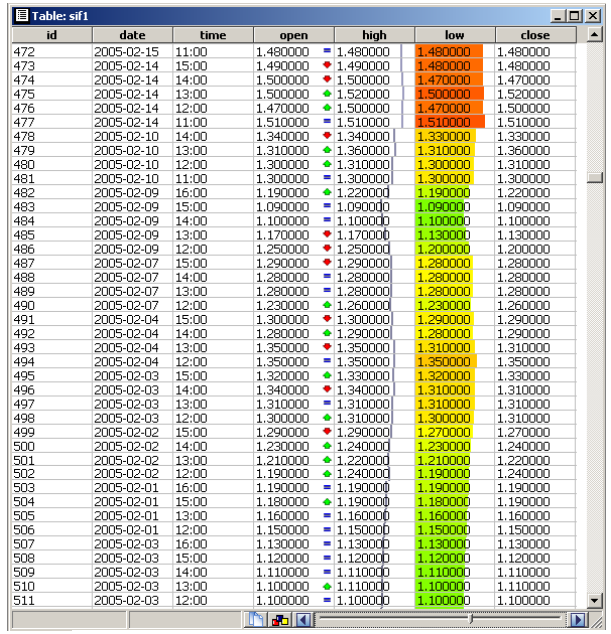
- Never include a color legend
- Avoid annotation
- Never mention error characteristics
- When in doubt, smooth
- Avoid providing performance data
- Never learn anything about the data or the discipline
- Never compare with others
- Never cite references of data
- Claim generalizability but show result on a single data
- Use view angle to hide shortcomings
- ‘This is easily extended to 3D’

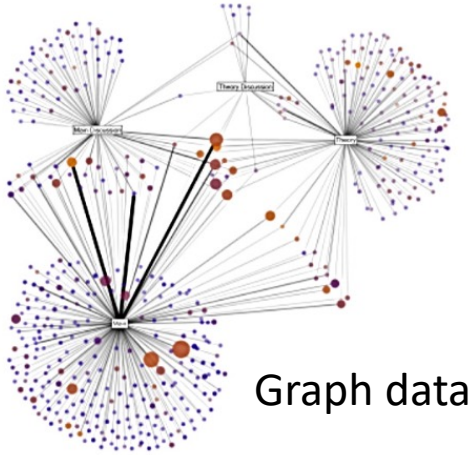
Fourteen Ways to Say Nothing with Scientific Visualization, Eric Raible, NASA, in IEEE Computer.

Information Visualization (InfoVis)

- The use of computer-supported, interactive visual representations of data to amplify cognition
  - Data is not necessarily defined on a spatial domain
  - Data is not always numerical
  - Data is inherently discrete
- The study of transforming data, information, and knowledge into interactive visual representations

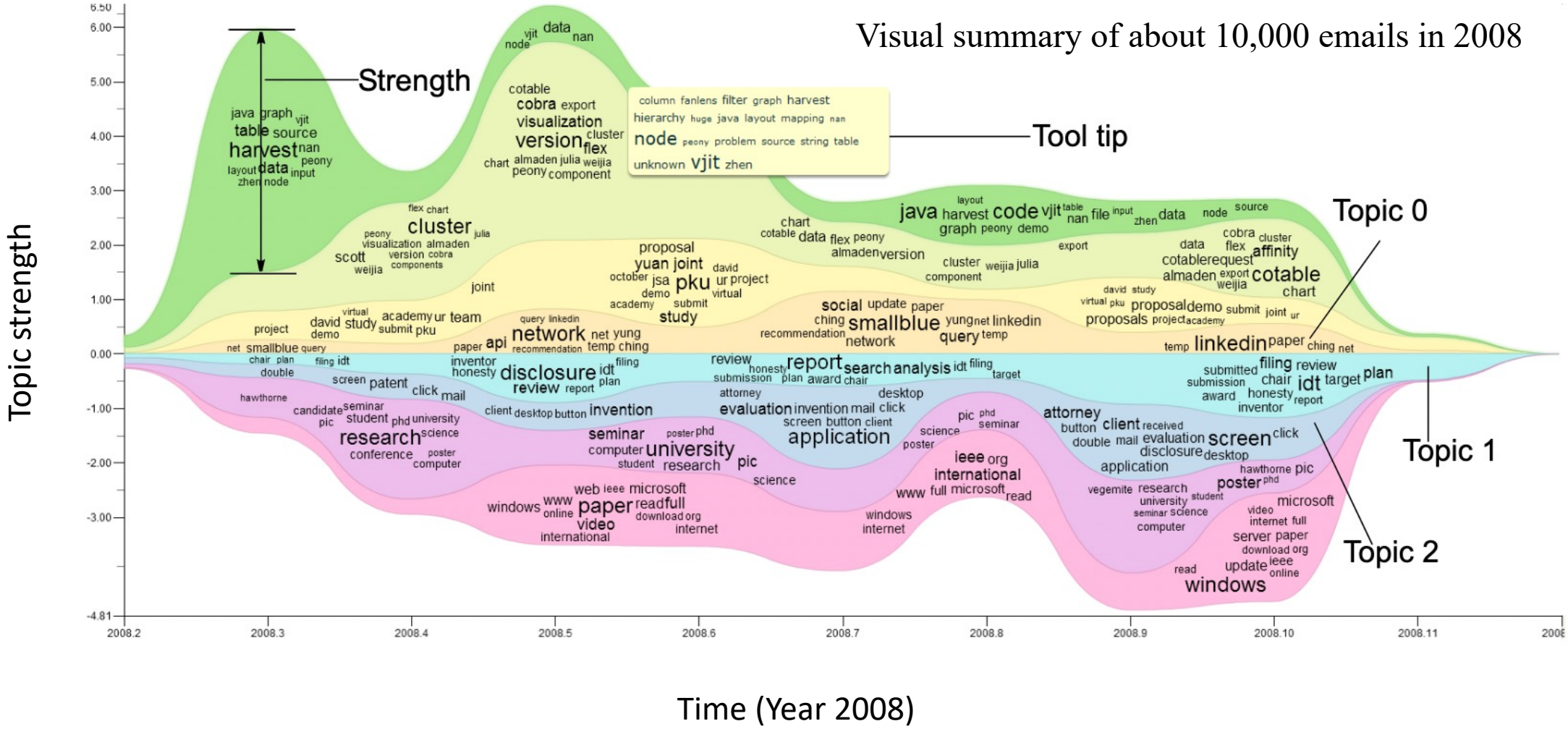
Table data



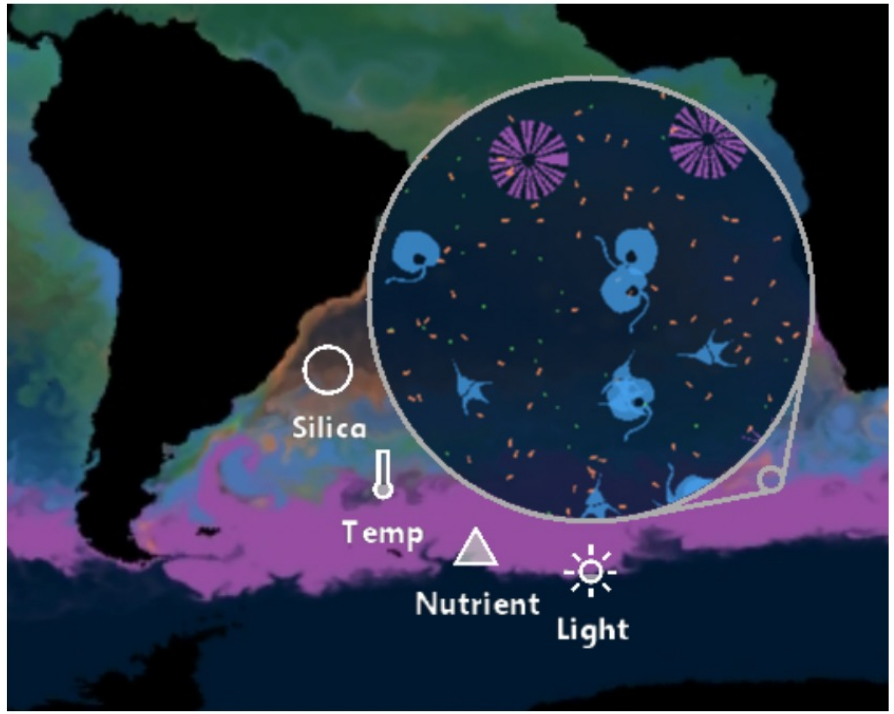


Graph data

Information Visualization for Business Data



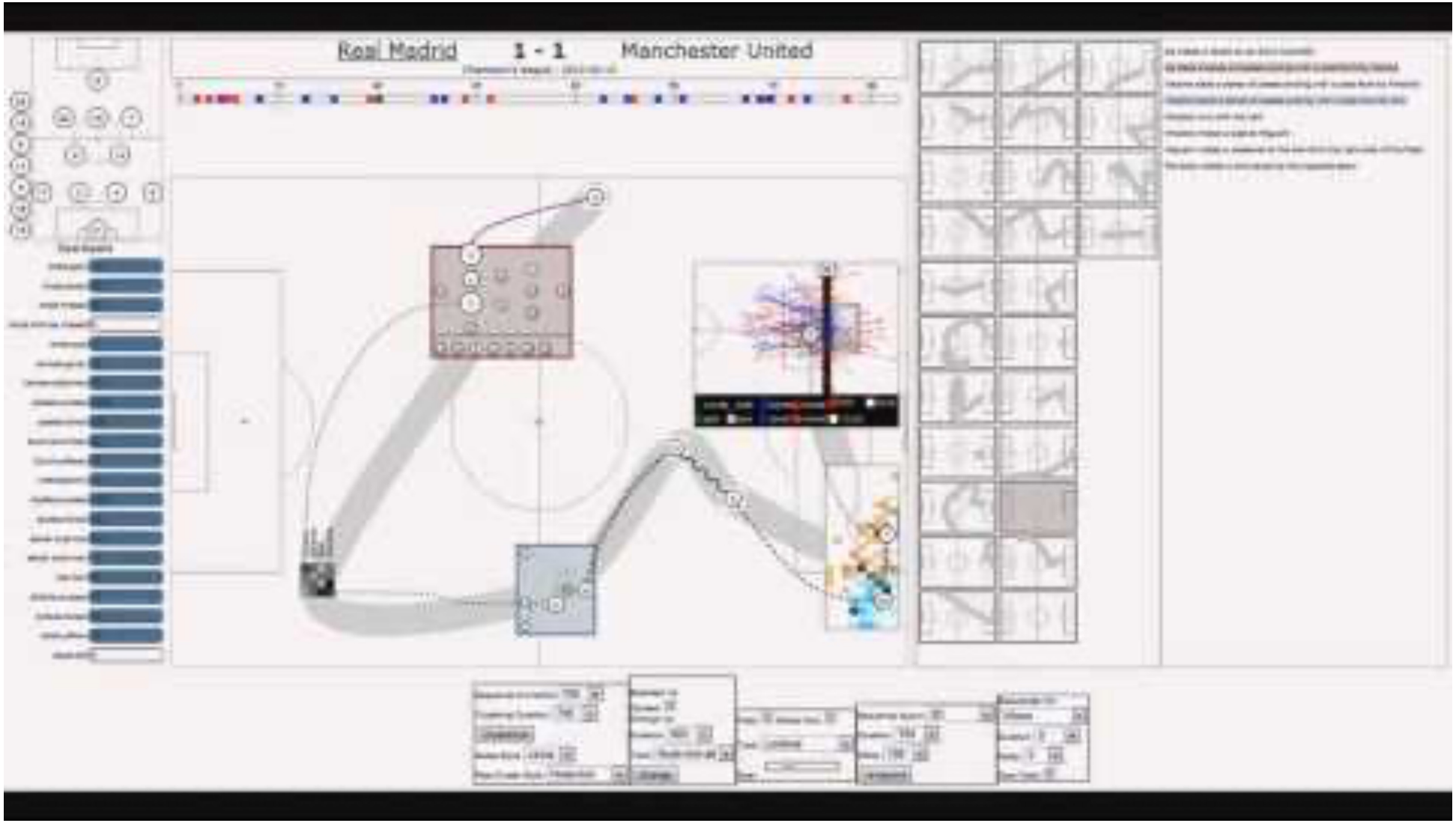
Information Visualization for Science Data



Circle viewer with indicators of environmental variables at the selected location. Silica: inorganic SiO_2 concentration; Temp: temperature; Nutrient: inorganic NO_3 concentration; Light: photosynthetically available radiation.

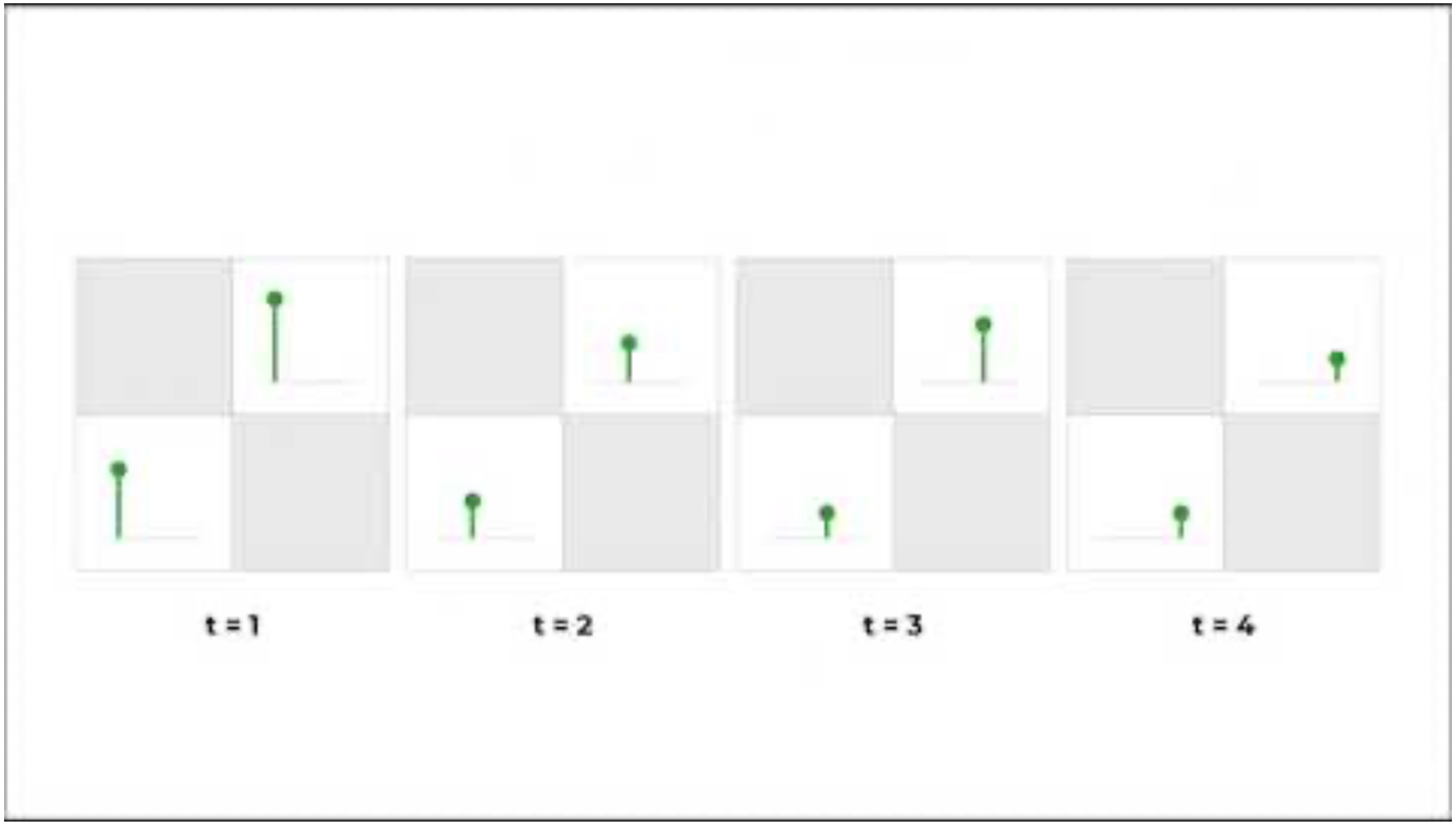
Exploratory Data Visualization Tool for Museum Visitors

Information Visualization for Soccer Data



Information Visualization for ML Classifiers

- A detailed evaluation of classifiers for model selection and debugging
- An interactive, comparative, model agnostic visualization system



Information Visualization for ML Model Explainability

A Brief Taxonomy of InfoVis Techniques

- InfoVis Techniques
  - Empirical Methods
  - Interaction
  - Frameworks
  - Applications

Empirical Methods

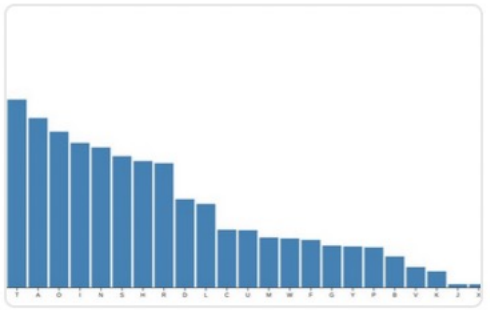
- Empirical methods are categorized as
  - Model and Evaluation
- Model
  - Visual representation model
  - Data driven model
- Evaluation
  - User studies are the most used in InfoVis and offer a scientifically sound method to measure visualization performance
    - Statistical methods

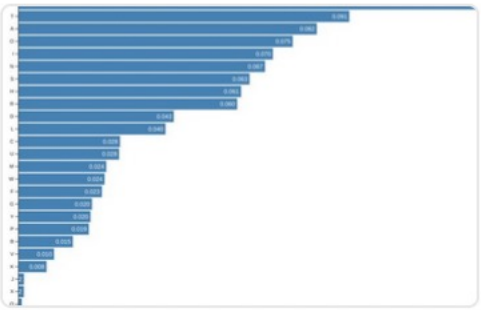
Interaction

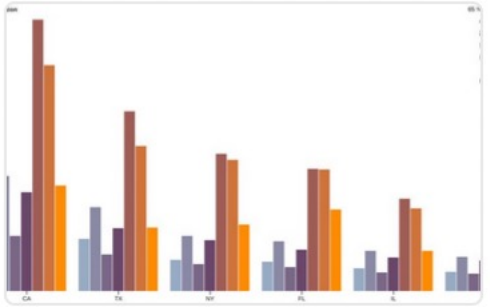
- Interaction is a fundamental aspect of InfoVis techniques
- Two Interaction categories
  - WIMP (windows, icons, mouse, pointer)
  - Post-WIMP
    - Touch interfaces
- Another operation-based categorization of interactions
  - select, explore, reconfigure, encode, abstract/elaborate, filter, and connect

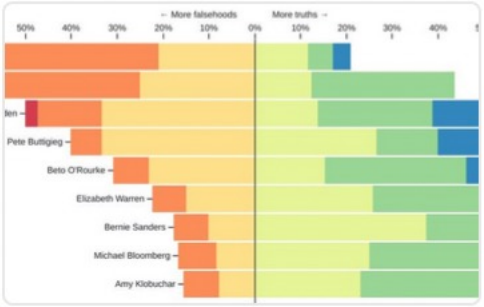
Frameworks/Systems

- Researchers have proposed a variety of visualization systems such as *Improvise*, the *InfoVis Toolkit*, and *Prefuse* to support the creation and customization of visualization applications.
- More recently, a new web-based library called Data-Driven Documents (D3) has become a very popular toolkit to construct interactive visualizations on the web
  - https://d3js.org/

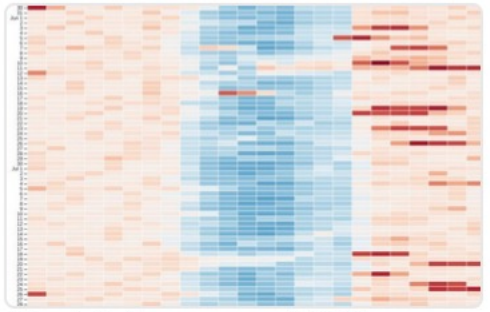

Bar chart

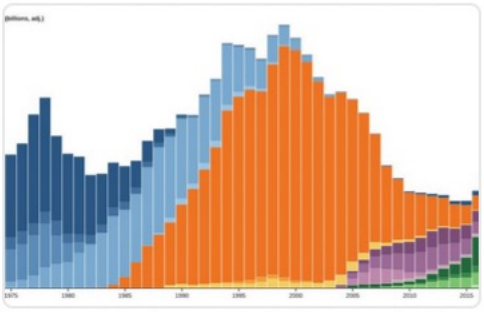

Horizontal bar chart


Grouped bar chart



Diverging stacked bar chart


Electricity usage, 2019


Revenue by music format, 1...

https://observablehq.com/@d3/galleryhttps://observablehq.com/@d3/gallery

Applications

- Four different types of data and applications
  - Graph data visualization
  - Text data visualization
  - Map data visualization
  - Multivariate data visualization

Exploratory Data Analysis

"The greatest value of a picture is when it forces us to notice what we never expected to see."

- John Tukey

InfoVis: Big Data Aspects

- Common objectives for big data visualization
  - Decision initiation or modification
  - Enhancing understanding
- Considerations for creating big data visualization systems
  - Source data
  - Information transfer to the audience
  - Design choices/scalability
- Enhance visualization by Graphical overlays
  - Highlights
  - Encodings
  - Summary statistics
  - Annotations

InfoVis: Issues and Risks

- Imprecision and Inaccuracy
  - Display information at a lower level of precision and accuracy than numerical or tabular formats
- Optical Significance
  - Viewer can interpret a difference or pattern as meaningful based on their perception, sometimes without corresponding quantitative evidence to support this interpretation
- Visualization Oversaturation
  - A dramatic increase in deficient and flawed visualizations

Libraries for Data Visualization: Matplotlib

- The most basic and Python's standard data visualization library
- A comprehensive library for creating static, animated, and interactive visualizations in Python.
- https://matplotlib.org/
- Examples: https://matplotlib.org/stable/gallery/index.html

Libraries for Data Visualization: Seaborn

- Built on top of Matplotlib but with better aesthetics and interactivity
- It provides a high-level interface for drawing attractive and informative statistical graphics.
- https://seaborn.pydata.org/
- Examples: https://seaborn.pydata.org/examples

Libraries for Data Visualization: Bokeh

- Bokeh is a Python library for creating interactive visualizations for modern web browsers.
  - Build beautiful graphics, ranging from simple plots to complex dashboards
- Create JavaScript-powered visualizations without writing any JavaScript code
- https://docs.bokeh.org/en/latest/

If you are new to Bokeh

Follow these guides to get started:

- First steps: simple tutorials that walk you through installing Bokeh and creating your first visualizations.
- User guide: explanations of all key functionalities of Bokeh and how to use them. Includes several standalone examples.

If you have some basic knowledge of Bokeh

Learn more by exploring examples:

- Gallery: a collection of examples with source code.
- Interactive tutorial notebooks: a collection of interactive notebooks to experiment with all elements of Bokeh.
- User guide: explanations of all key functionalities of Bokeh and how to use them, including examples.

If you need more advanced information

Get to know every aspect of Bokeh:

- Reference guide: detailed information about all of Bokeh's components.
- Contributor guide: information on the various ways you can contribute to the Bokeh project.

Libraries for Data Visualization: Plotly Dash

- Dash is an Open-Source Python library for creating reactive, Web-based applications
  - Built on top of Plotly.js and React.js
  - User interface library for creating analytical web applications
- https://dash.plotly.com/
- https://dash.gallery/Portal/

- **Dash is 'React' for Python**
  - React: A JavaScript library for building user interfaces

Libraries for Data Visualization: D3

- D3 - Data-Driven Documents
  - D3.js is a JavaScript library for manipulating documents based on data.
  - D3 helps you bring data to life using HTML, SVG, and CSS.
  - D3's emphasis on web standards gives you the full capabilities of modern browsers
  - Combines powerful visualization components and a data-driven approach to DOM manipulation
- https://d3js.org/

Study Materials for Lecture 10

- Reference papers and resoucres on slide footnotes
- Book: Visualization Analysis and Design by T. Munzner
  - Chapter 6: Rules of Thumb for Designing Visualizations
  - Chapter 12: Facet into Multiple Views
  - Chapter 14: Focus + Context
  - Chapter 13: Reduce Items and Attributes

Data Types

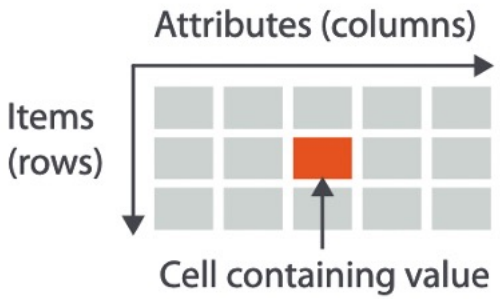
- Type = Structural or mathematical interpretation of the data

→ Items *(row, node)* → Attributes *(variable, data dimension)* → Links *(relationship)* → Positions *(spatial location)* → Grids *(sampling)*

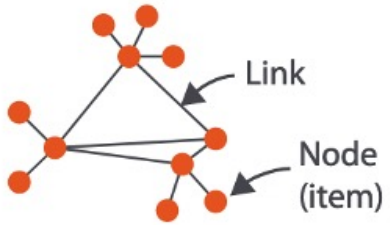
Data Types

- Dataset = collection of information/data

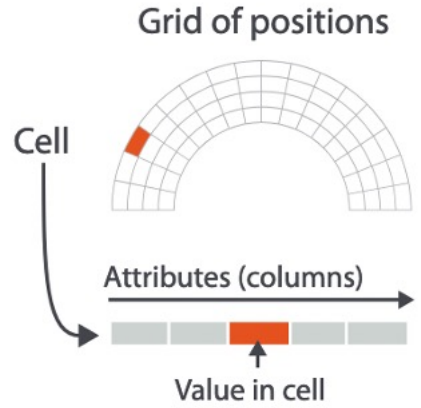
➔ Tables



➔ Networks

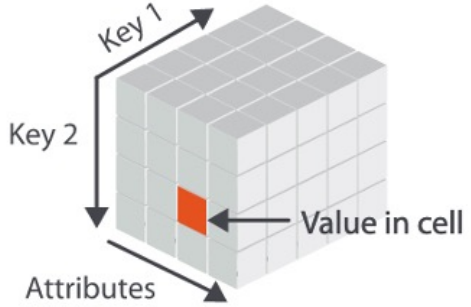


➔ Fields (Continuous)

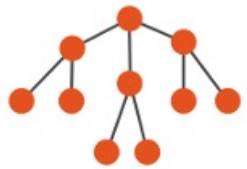


➔ Geometry (Spatial)

➔ *Multidimensional Table*



➔ *Trees*



Attribute Types

- Attribute Types:

→ Categorical *(nominal)*

e.g., fruit (apple, pear, etc.)
colleges (CCIS, CAMD, etc.)

→ Ordered

→ *Ordinal* *(ordered)*

e.g., months (J, F, M, etc.)
sizes (xs, s, m, l, xl)

→ *Quantitative* *(continuous)*

e.g., lengths (1", 2.5", 5")
population

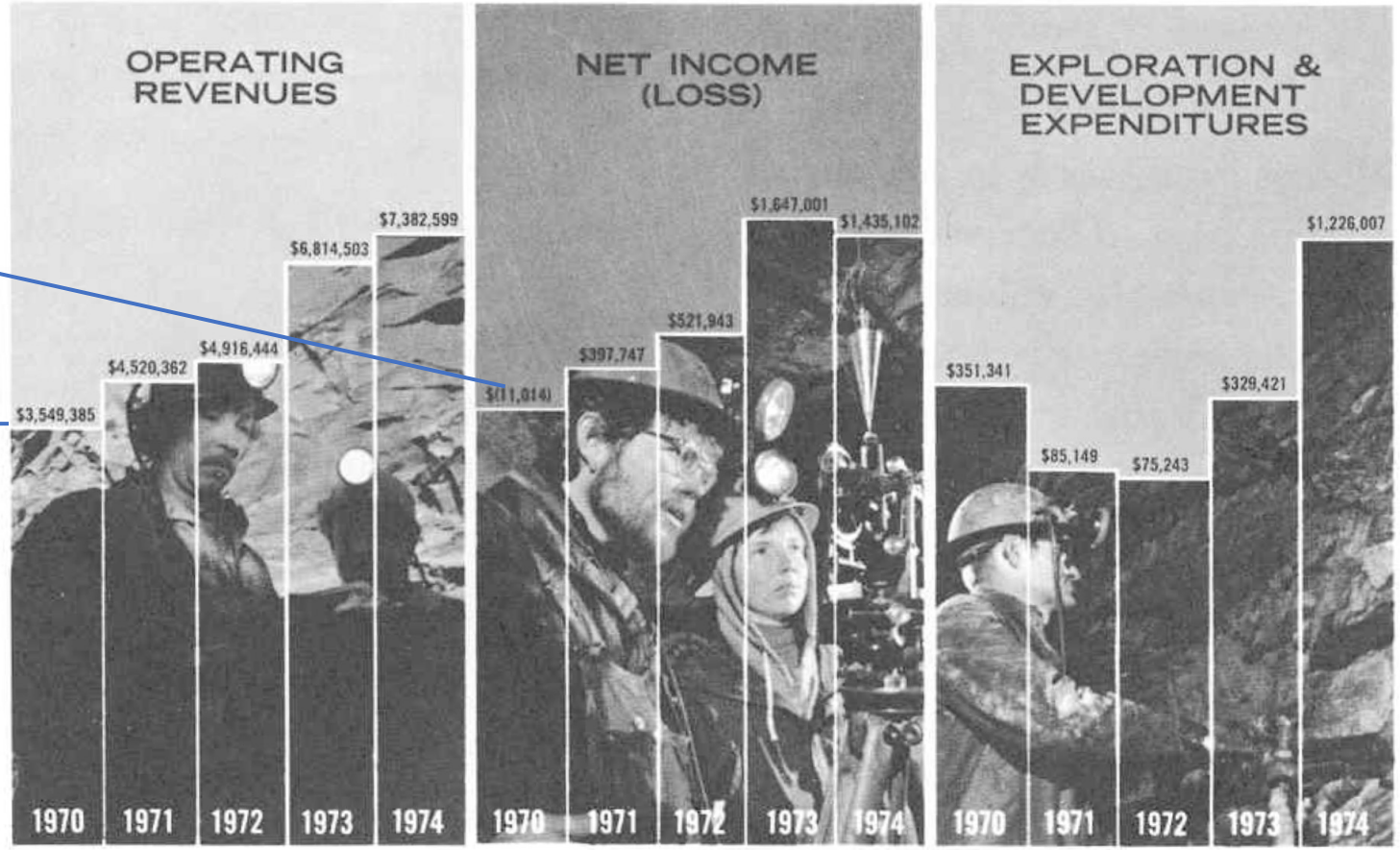
Rules of Thumb for Designing Visualizations

Graphical Integrity

"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data"

Tufte, "Visual Display of Quantitative Information" (1983)

Graphical Integrity



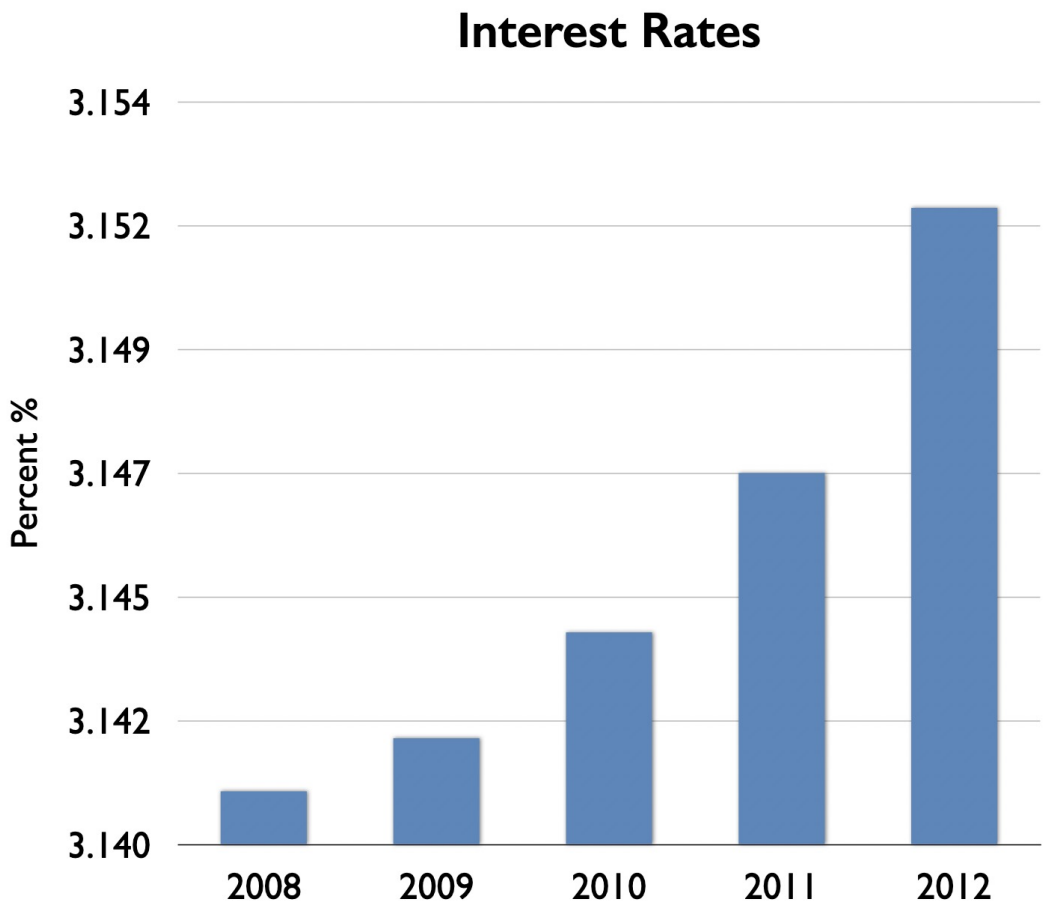
$11,014

$3,549,385

No Y-axis baseline!

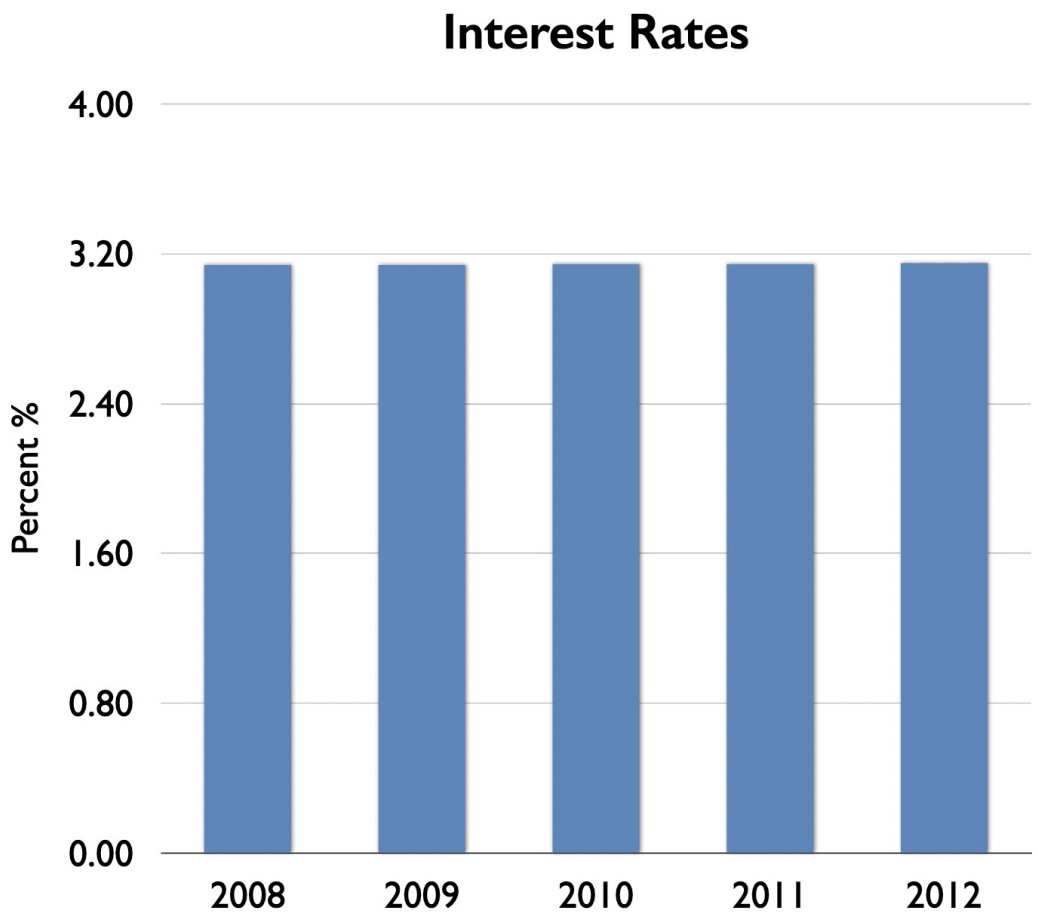
"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

Graphical Integrity



"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

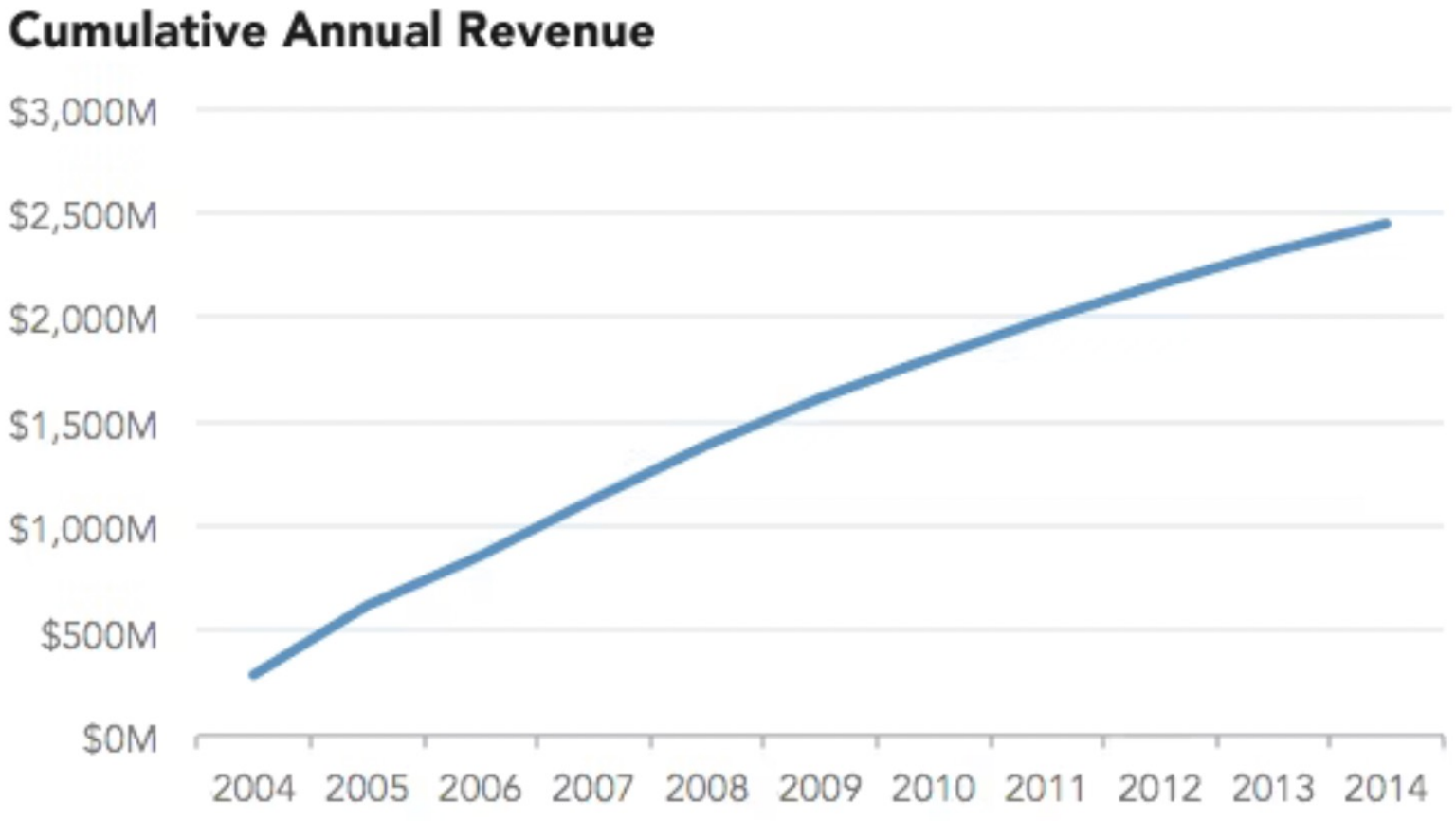
Graphical Integrity



Y-axis scale is important to show the context

"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

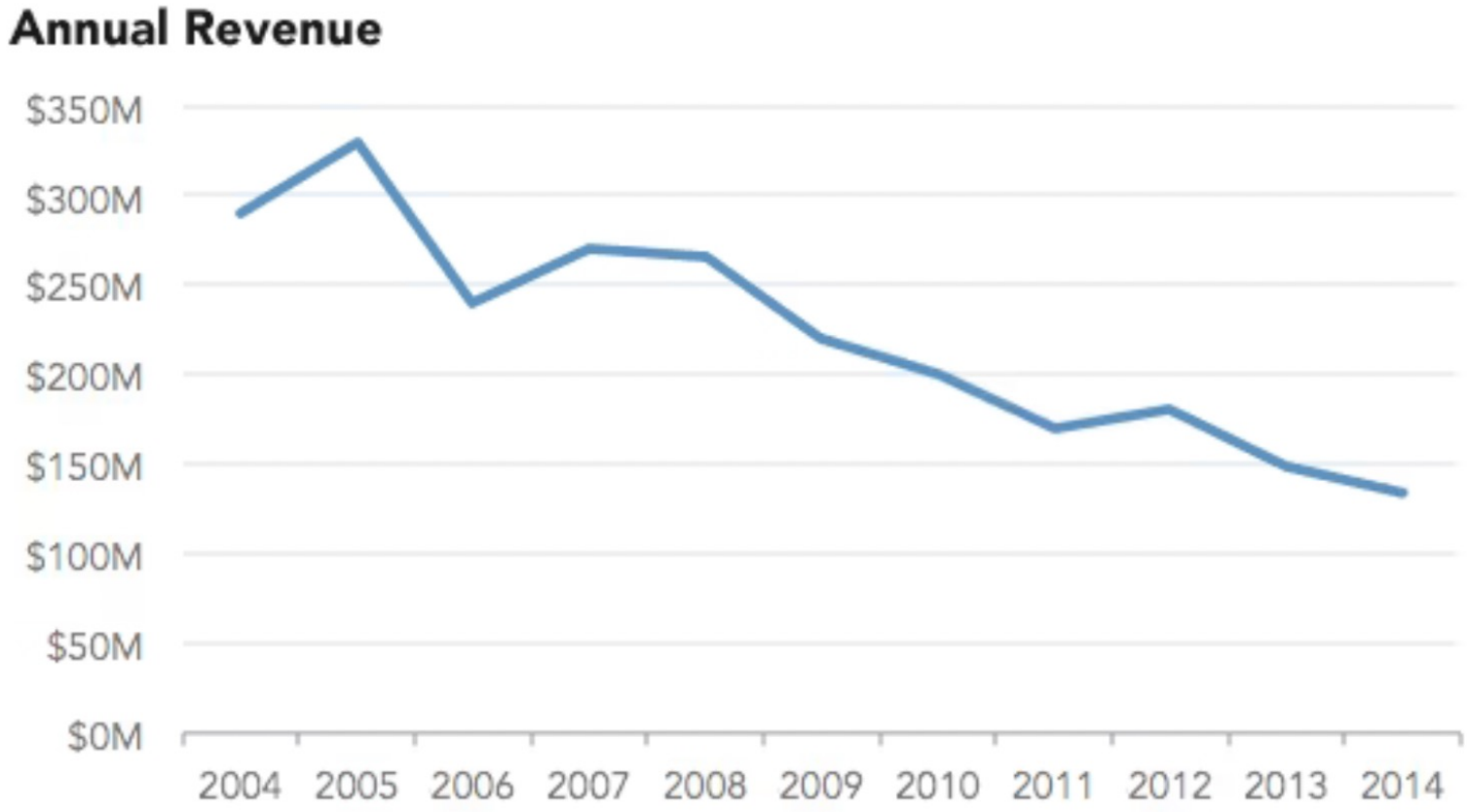
Graphical Integrity



Y-axis scale is important to show the context

"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

Graphical Integrity

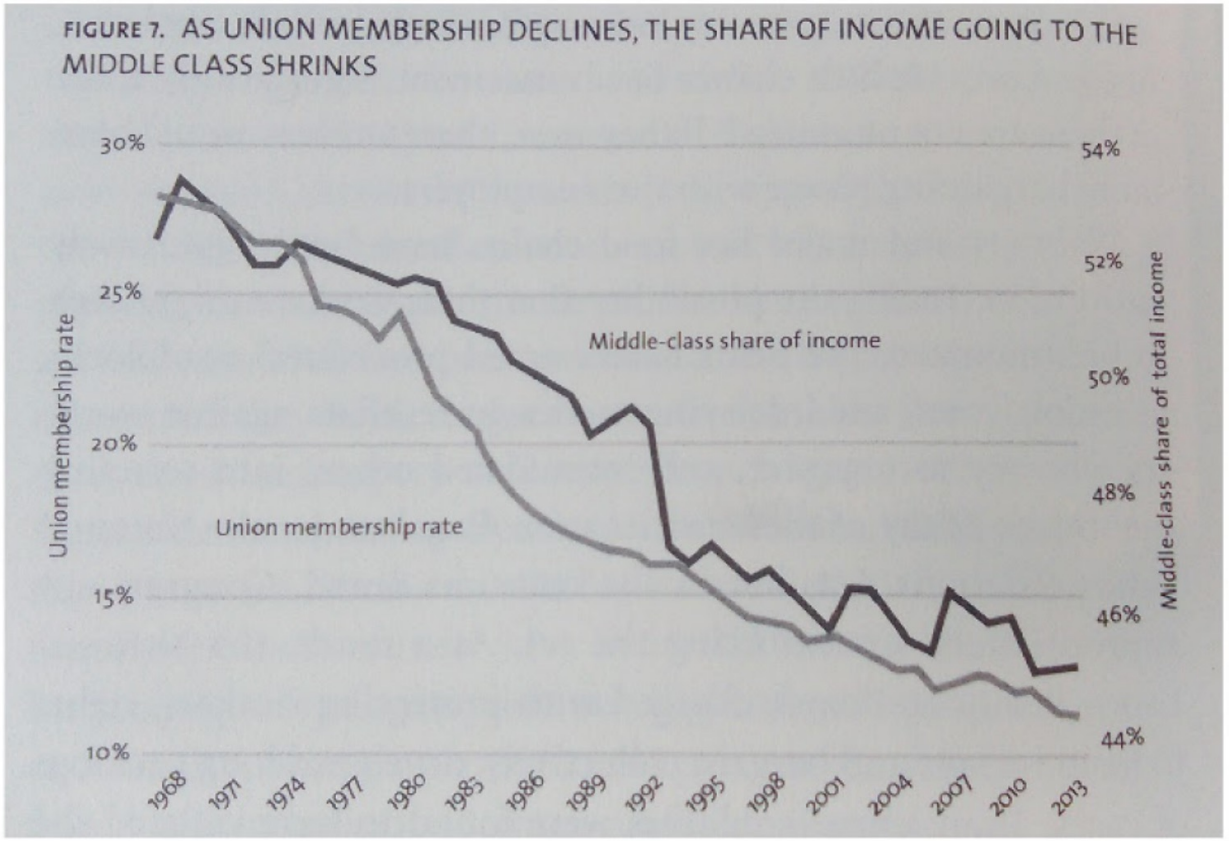


Cumulative graphs can mislead

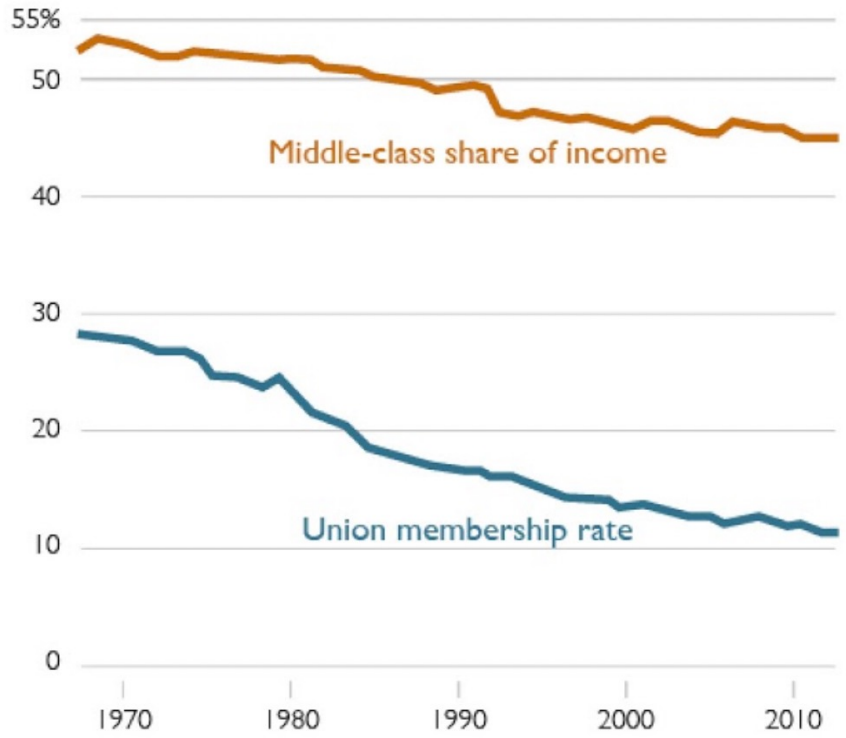
"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

Graphical Integrity

Double the axes, double the mischief

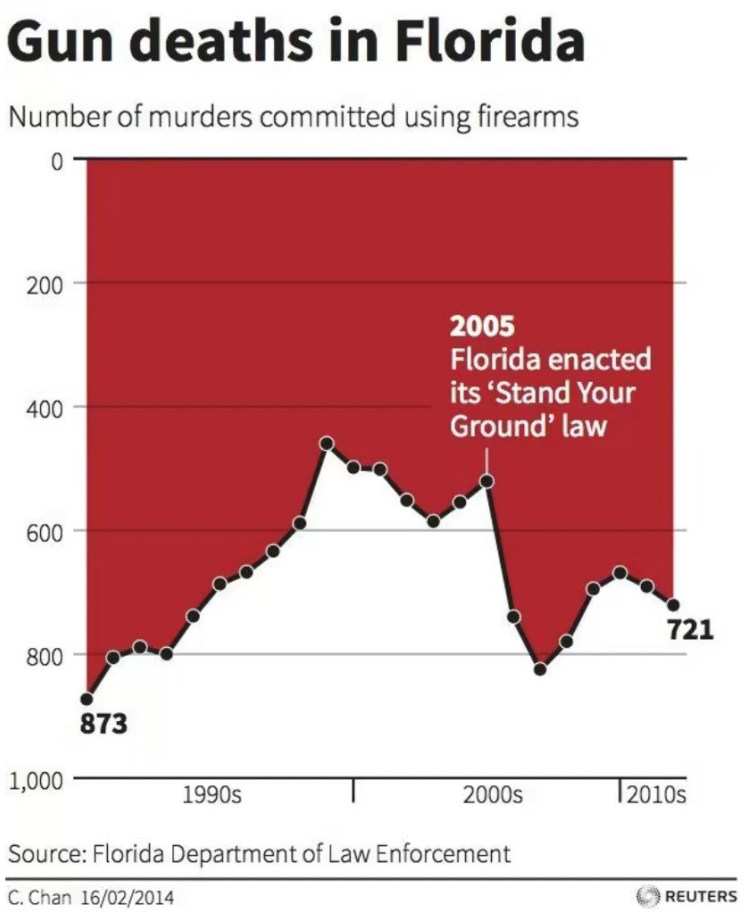


NEW VERSION



"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

Graphical Integrity



Y-axis is flipped!

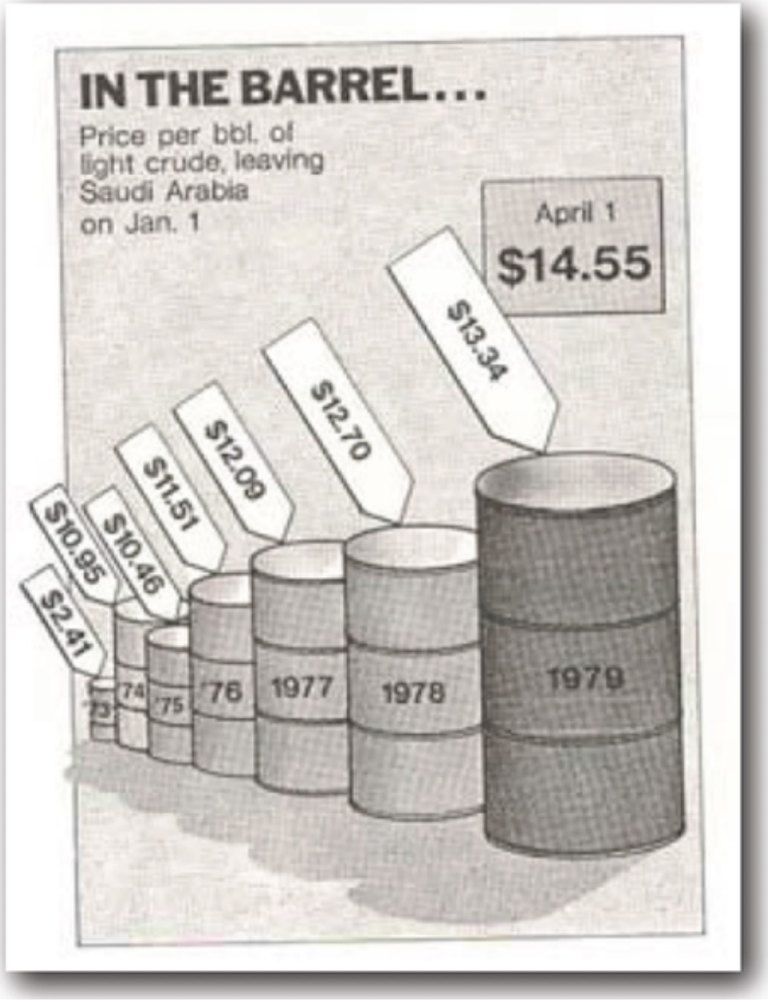
"Clear, detailed, and thorough labeling should be used to defeat graphical distortion and ambiguity. Write out explanations of the data on the graphic itself. Label important events in the data."

Graphical Integrity

"The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities measured."

Tufte, "Visual Display of Quantitative Information" (1983)

Graphical Integrity

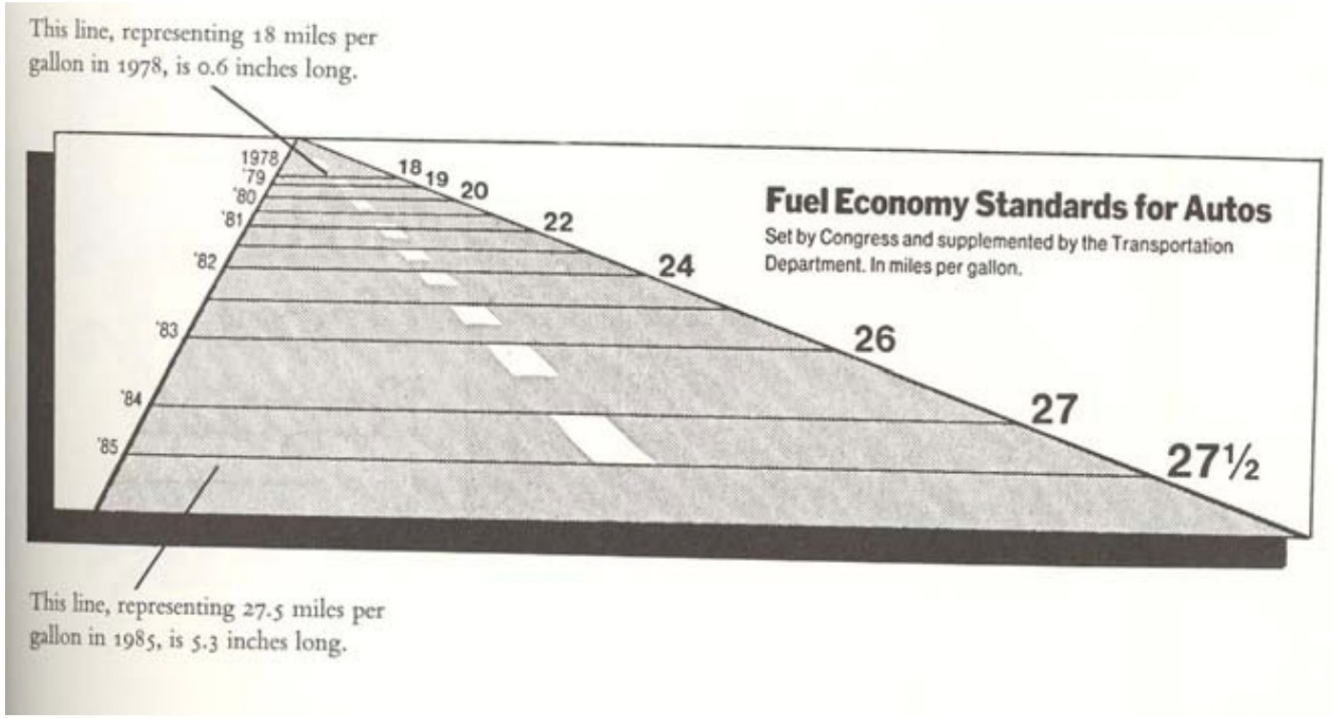


Inconsistent proportion of barrel sizes

"The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities measured."

Graphical Integrity: Lie Factor

- Lie Factor = (Size of effect in graphic)/(Size of effect in data)
- Lie Factor = >1, overstating
- Lie Factor = 1, accurate
- Lie Factor = <1, understating



Graphical Integrity: Lie Factor

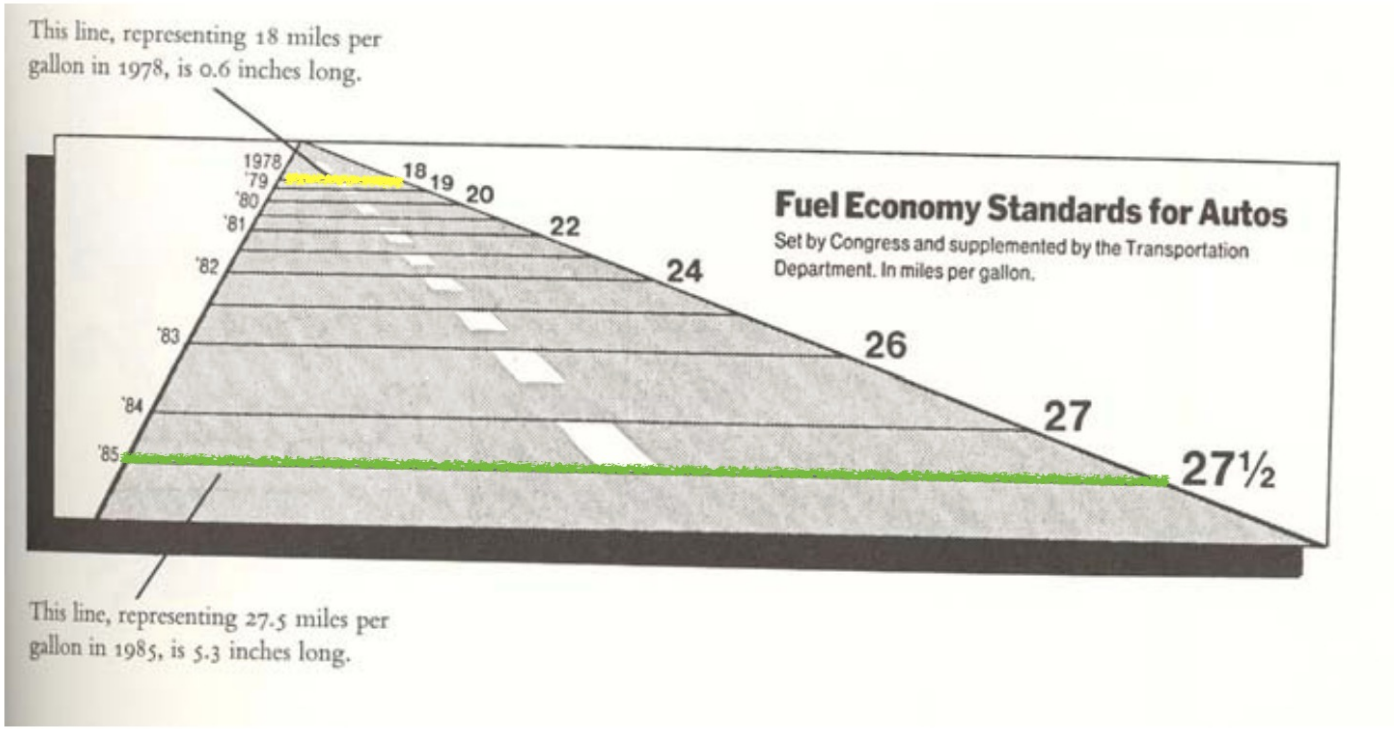


Image = $\frac{5.3" - 0.6"}{0.6"} = 7.83 = 783\%$

Data = $\frac{27.5 - 18}{18} = 0.53 = 53\%$

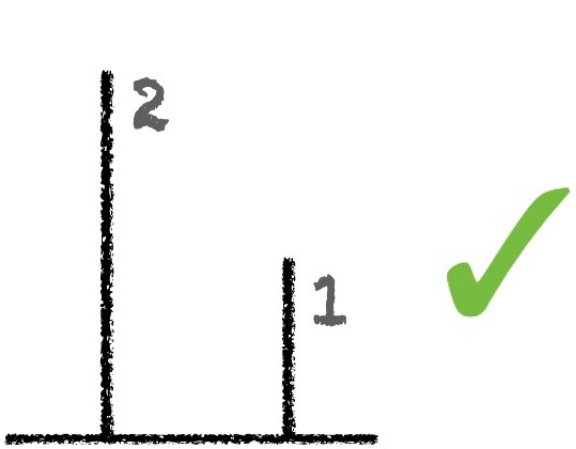
Lie Factor = $\frac{783\%}{53\%} = 14.8$

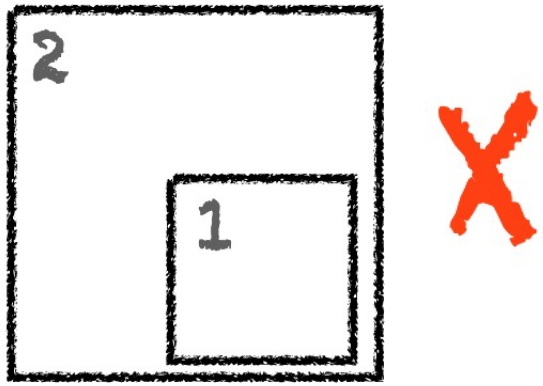
Lie Factor = >1, overstating

"The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities measured."

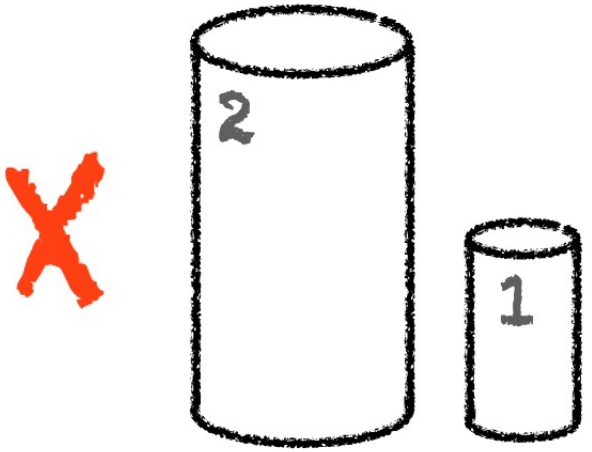
Graphical Integrity: Lie Factor

Lie Factor = (Size of effect in graphic)/(Size of effect in data)





Make sure area is proportional to data!



3D bar charts are bad

"The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities measured."

Graphical Integrity

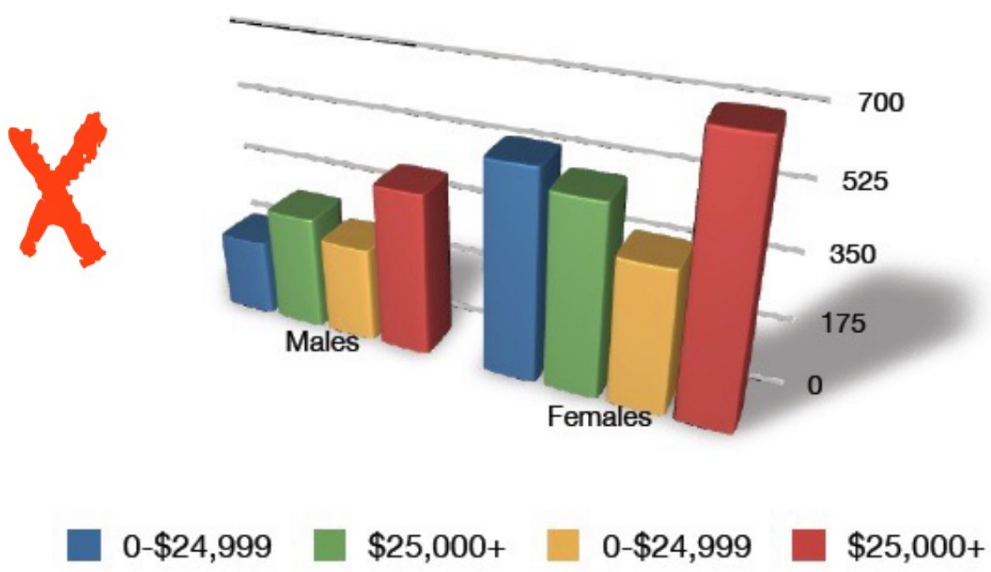
"The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data."

Tufte, "Visual Display of Quantitative Information" (1983)

Graphical Integrity

- No Unjustified 3D

Dimensions in data: 2
Dimensions in plot: 3

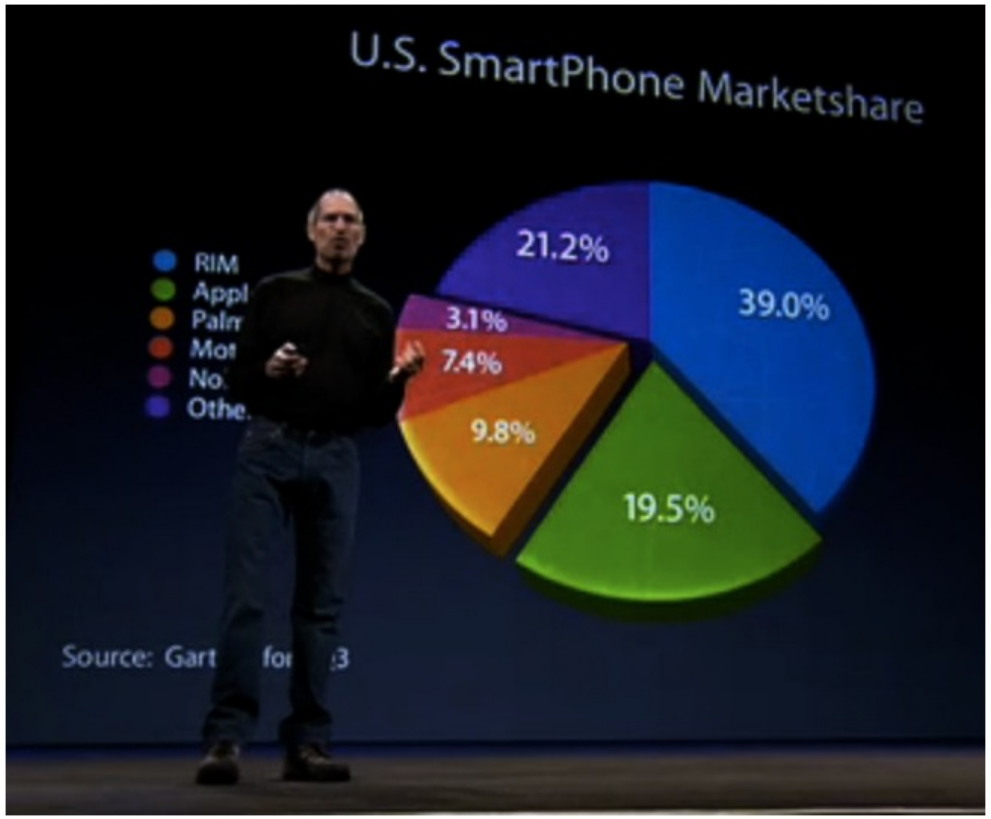


Dimensions in data: 2
Dimensions in plot: 2

"The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data."

Graphical Integrity

- No Unjustified 3D



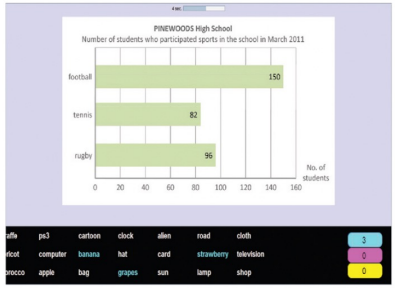
"The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data."

ChartJunk Debate

- All elements in visualization that are not necessary for interpreting the information related to the data being shown

- Heavy or dark grid lines
- Unnecessary text
- Inappropriately complex or gimmicky font faces
- Ornamented chart axes
- Backgrounds or icons within data graphs
- Ornamental shading and unnecessary dimensions

An Empirical Study on Using Visual Embellishments in Visualization

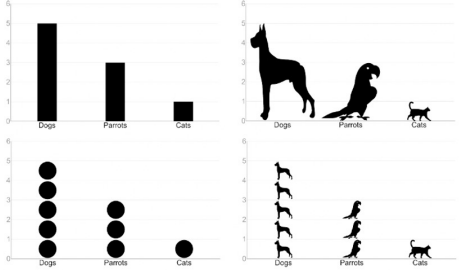


Borgo, et al. (2012)

What makes a visualization memorable?

Borkin, et al. (2013)
Borkin, et al. (2015)

ISOTYPE Visualization – Working Memory, Performance, and Engagement with Pictographs

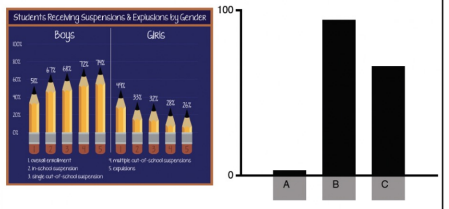


Haroz, et al. (2015)

Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts

Bateman, et al. (2010)

An Evaluation of the Impact of Visual Embellishments in Bar Charts



Skau, et al. (2015)

Tufte: Graphical displays should...

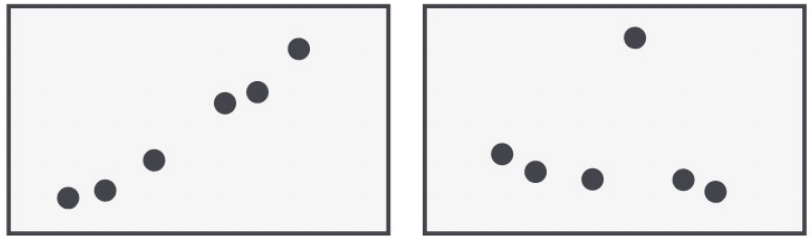
- Show the data
- Avoid distorting what the data have to say
- Encourage comparisons
- Reveal the data at several levels of detail
- Serve a reasonably clear purpose
- Be closely integrated with the statistical and verbal descriptions

Principled Approaches for InfoVis

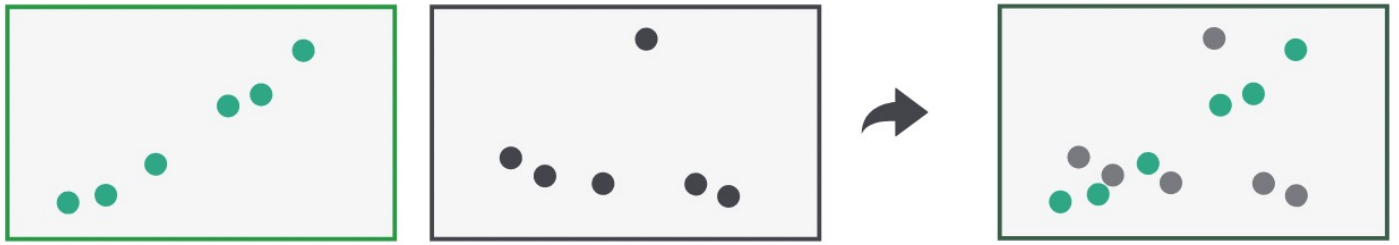
- Views & Facet & Linked Data
- Focus and Context
- Filtering and Aggregation

Views and Facet

- Split visualization plots into multiple views or separate into multiple layers
- Benefit?
  - Complexity reduction
  - Rely on vision instead of memory retrieval!



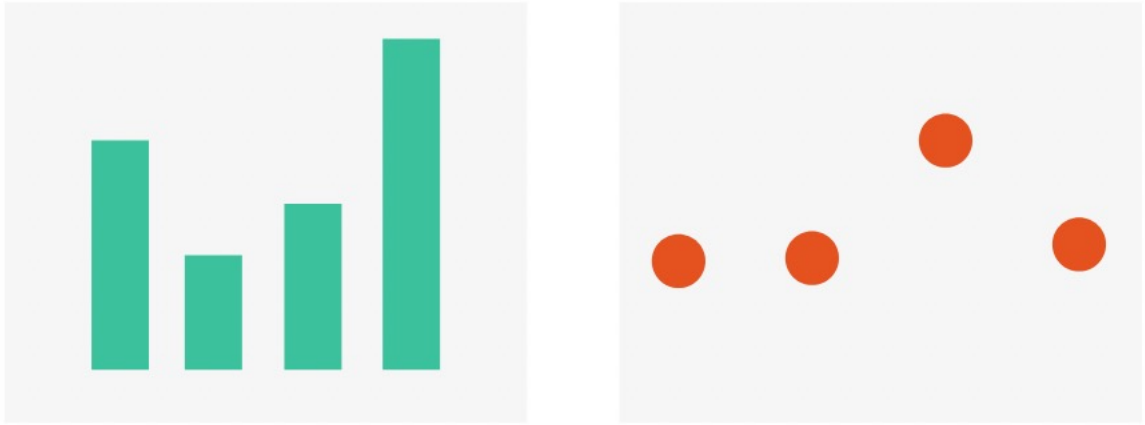
Side-by-side views



Superimpose as layers

Views and Facet: Side-by-Side

- Partition into Side-by-Side Views
  - Easy to compute and build
  - BUT: Multiple views take up more space!
- Side-by-side: Juxtapose
  - Small multiples
  - Multiform

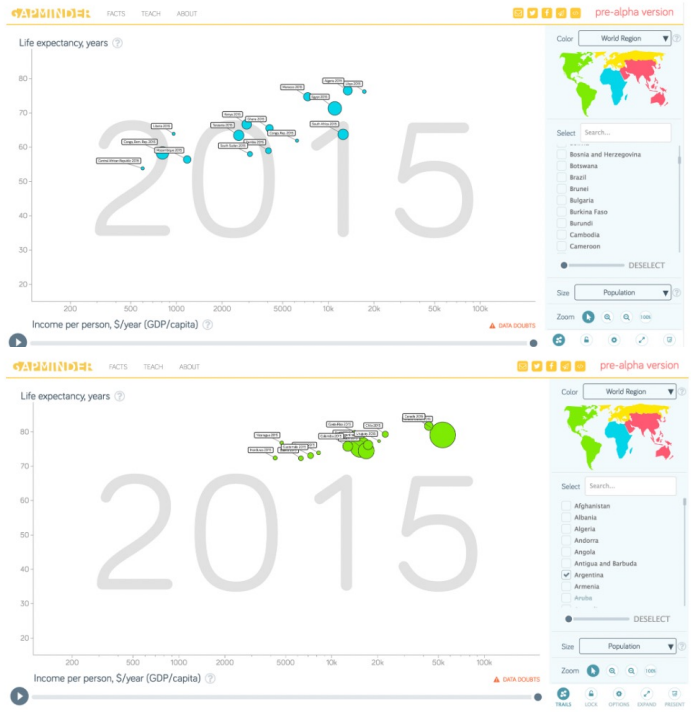


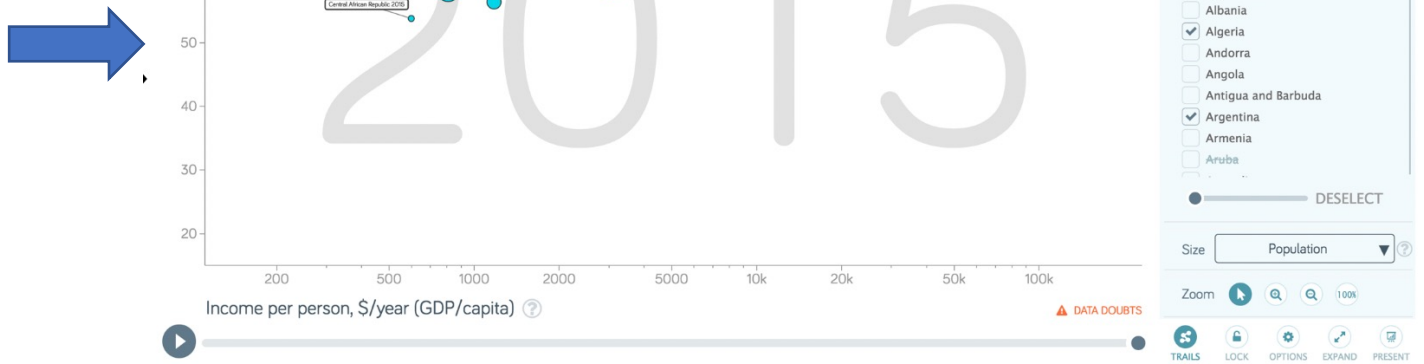
Multiform: Same data, different encoding

Small multiples

Views and Facet: Superimpose Layers

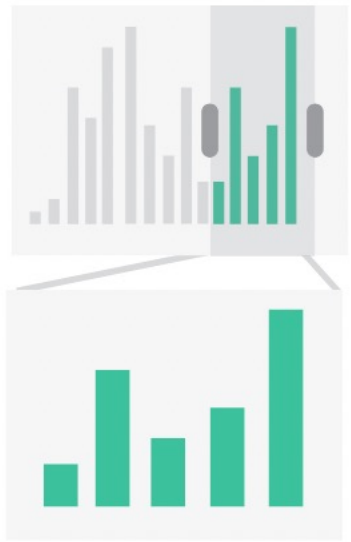
- Superimpose visualization layers to show data patterns
  - Less screen space required
  - still easy to compare
  - BUT: limits encoding options
  - Can get messy soon with multiple layers



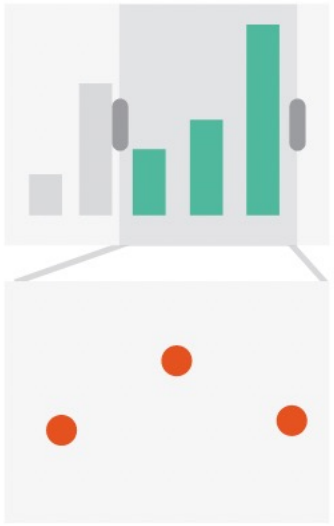


Views and Facet: Overview and Detail

- Provides detailed view of a subset
- Benefit: For large or complex data, a single view of the entire dataset cannot present fine details



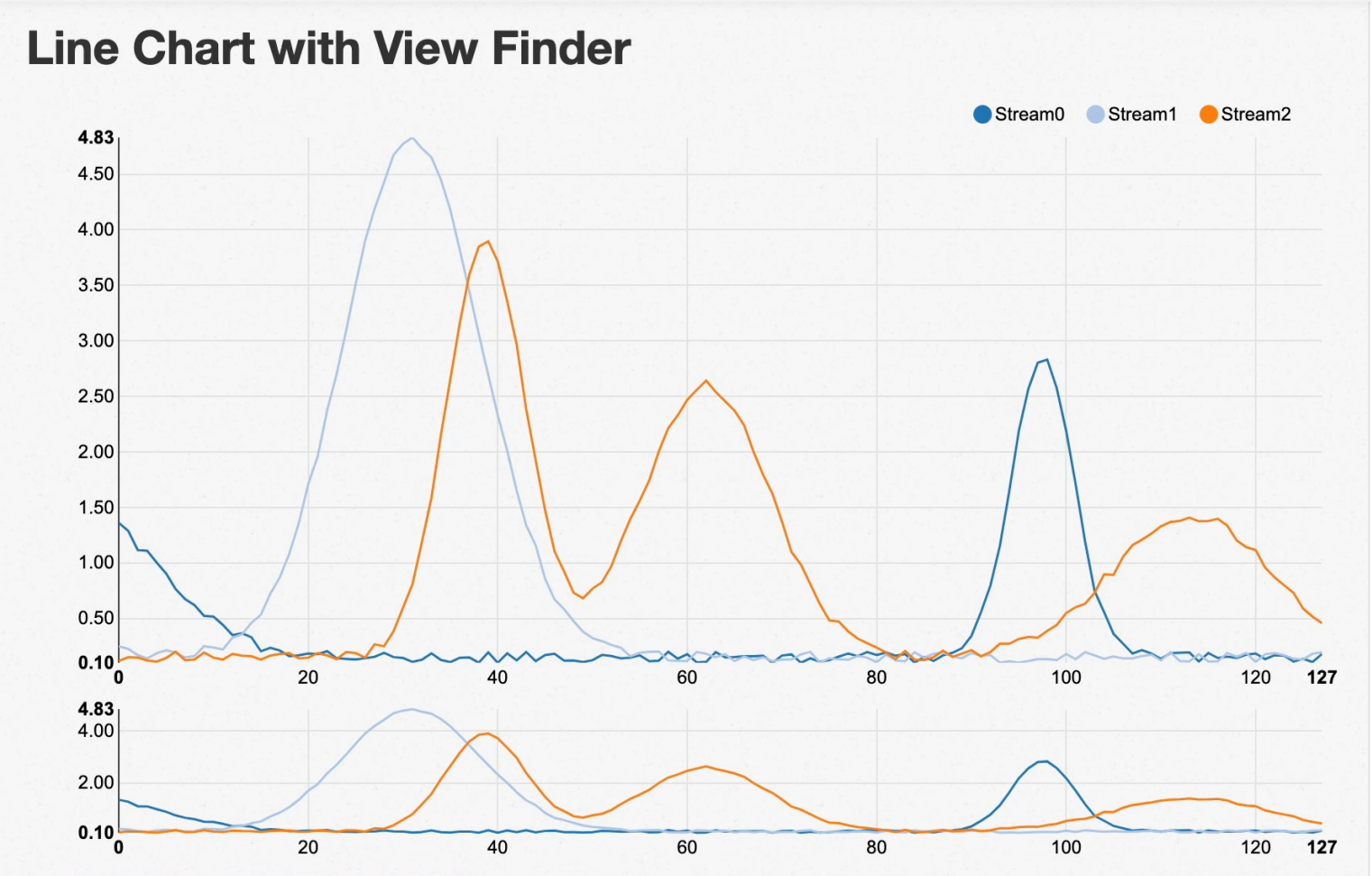
Overview/ Detail



Multiform, Overview/ Detail

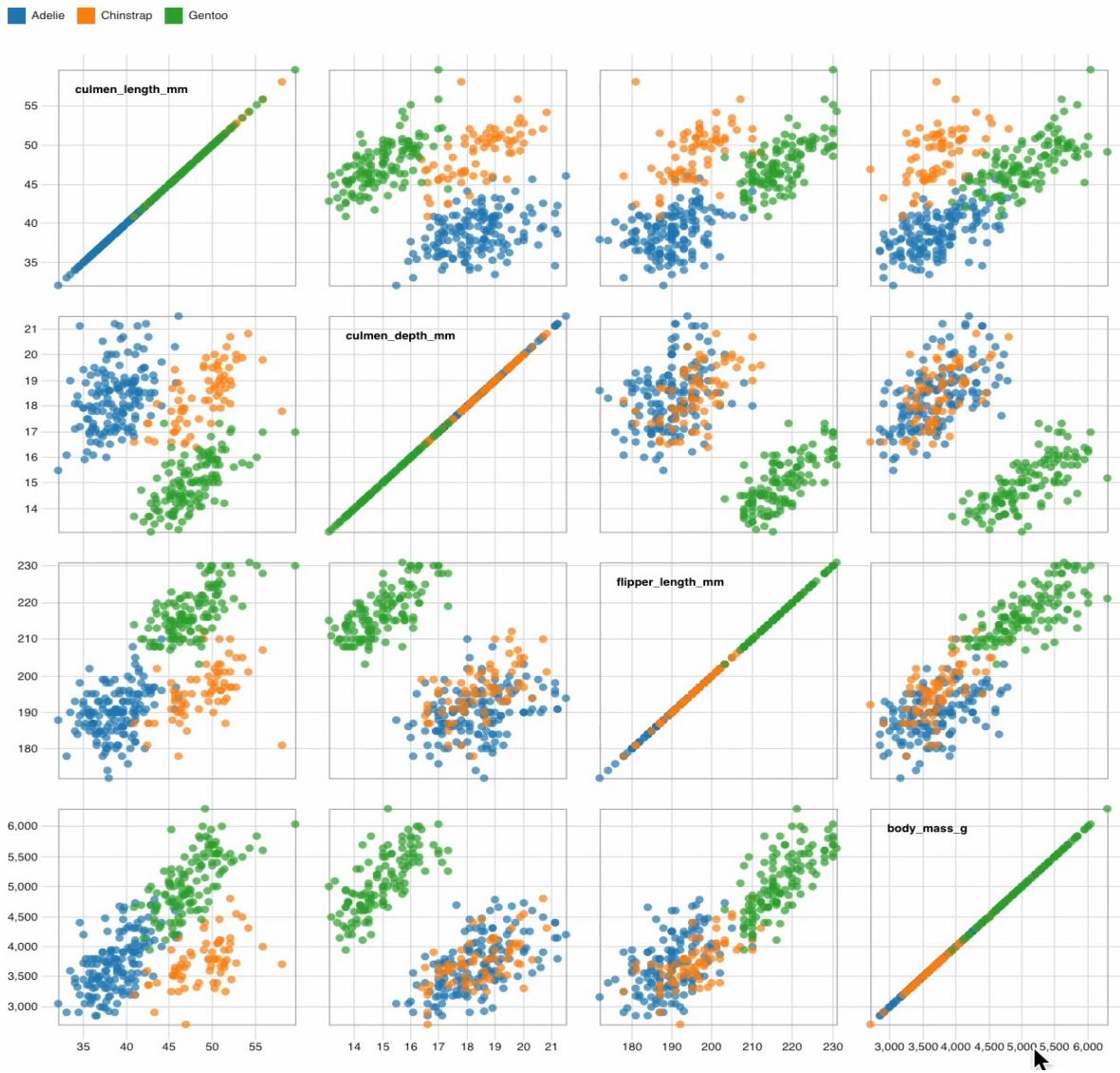
Views and Facet: Brush and Zoom

- Brush and Zoom



Views and Facet: Brush and Link

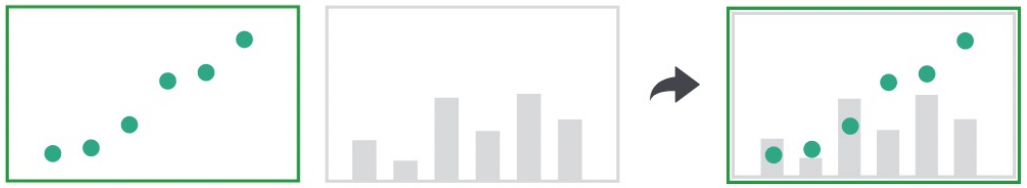
- Brush and Link
- Multiple views that are simultaneously visible and linked together such that actions in one view affect the others



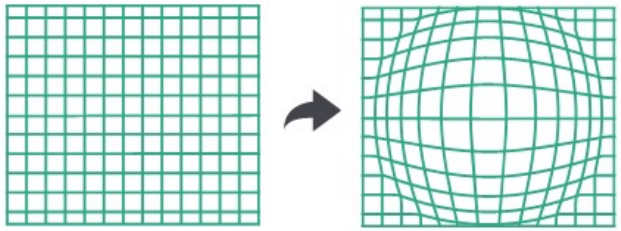
Focus and Context

- User selects region of interest (focus) through navigation or selection
- Provide context through aggregation, reduction, or layering
- Carefully pick what to show; hint at what you are not showing

Elide data



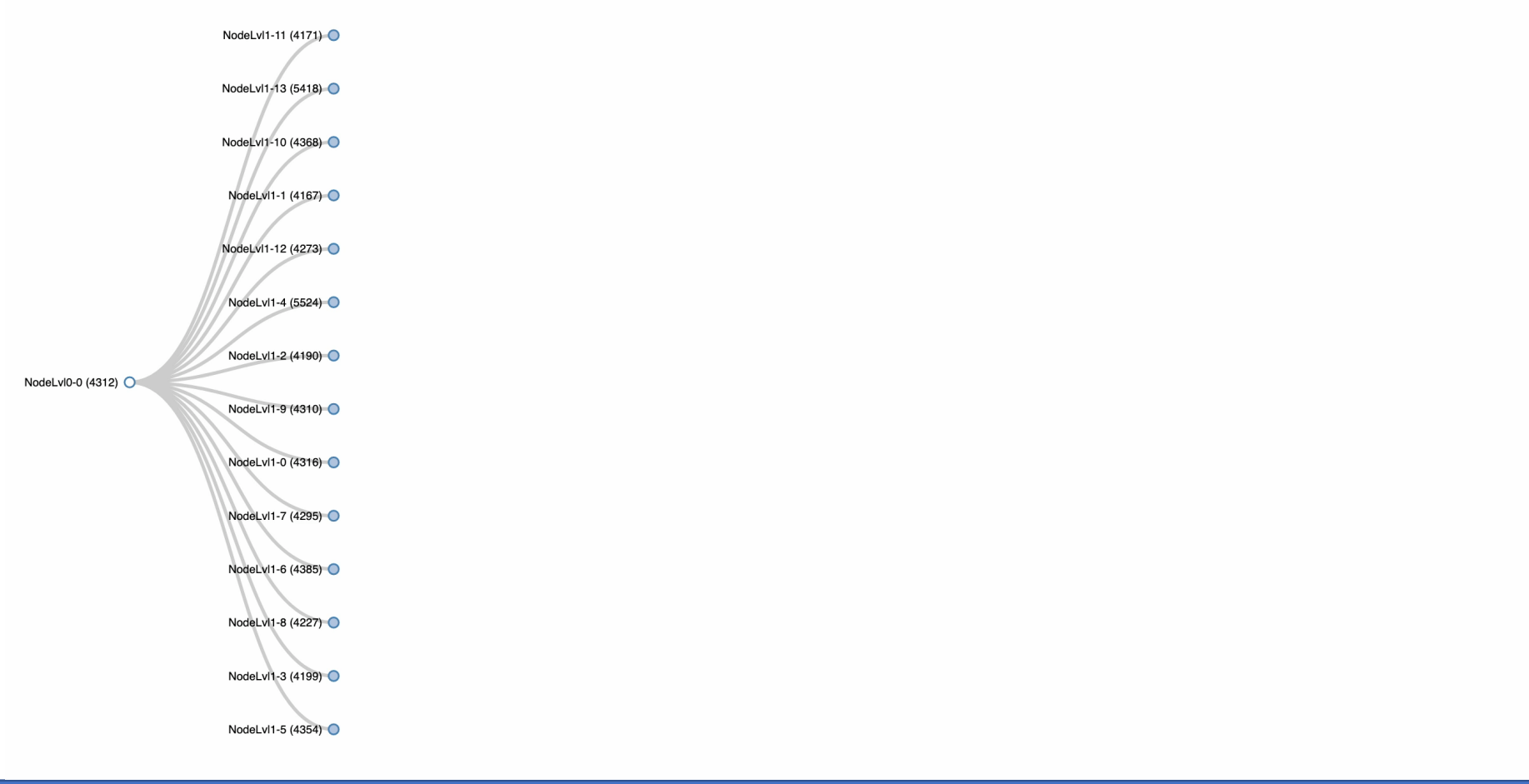
Superimpose layers



Distort geometry

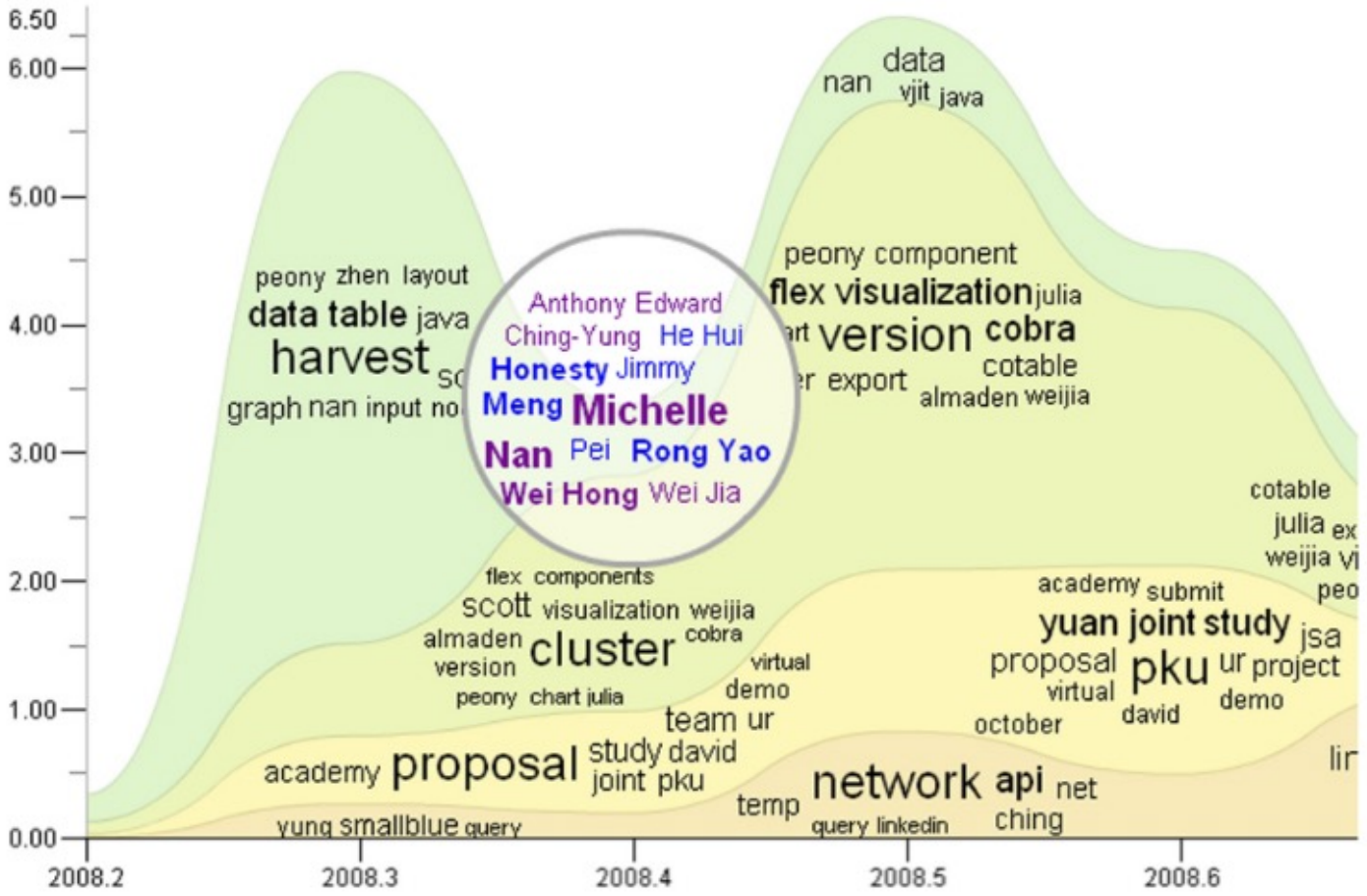
Focus and Context: Elision

- Focus items shown in detail, other items summarized for context – Collapsible tree



Focus and Context: Superimpose Layers

- Focus layer limited to a local region of view, instead of stretching across the entire view – Standard Lens



https://dl.acm.org/doi/pdf/10.1145/2089094.2089101

Focus and Context: Superimpose Layers

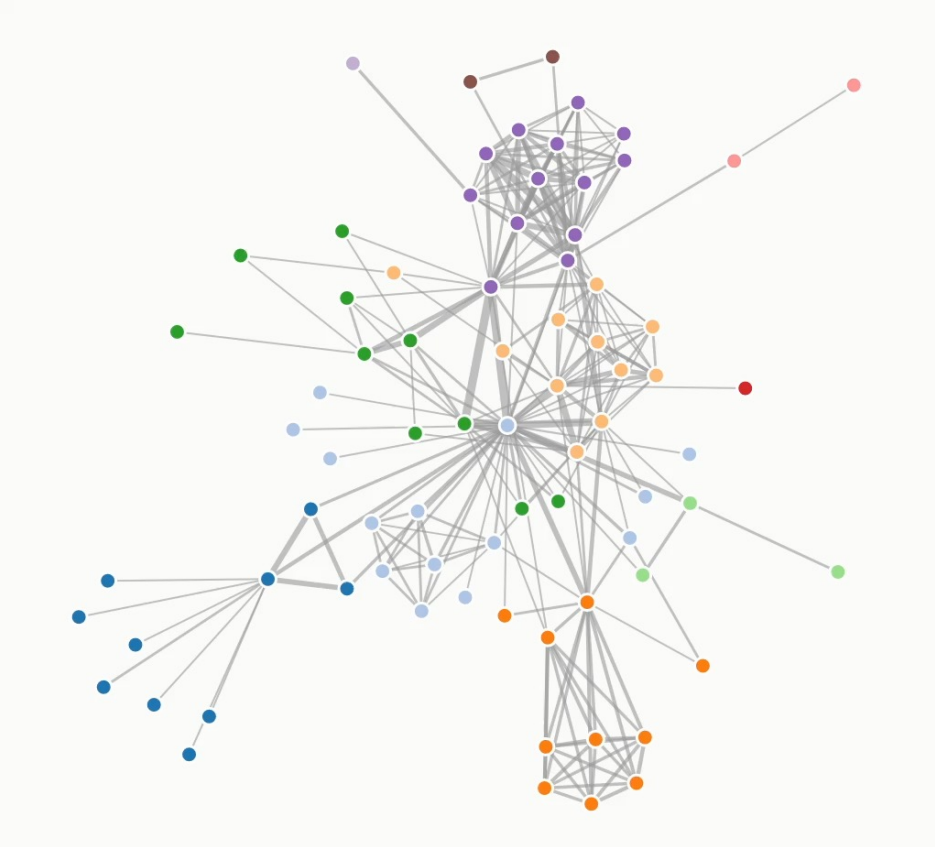
- Focus layer limited to a local region of view, instead of stretching across the entire view – Magic Lens

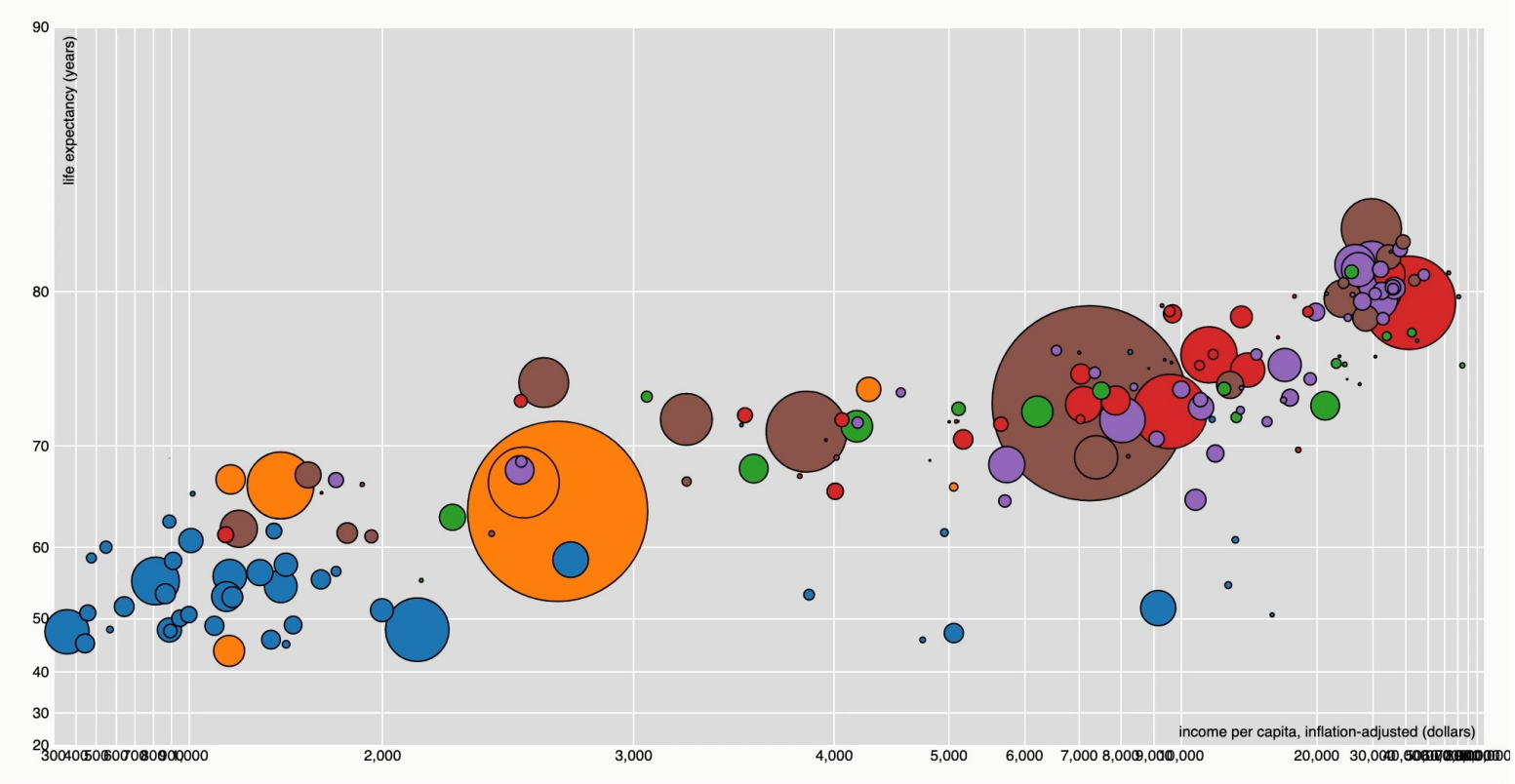
Focus and Context: Superimpose Layers

- Focus layer limited to a local region of view, instead of stretching across the entire view – FingerGlass

Focus and Context: Distort Geometry

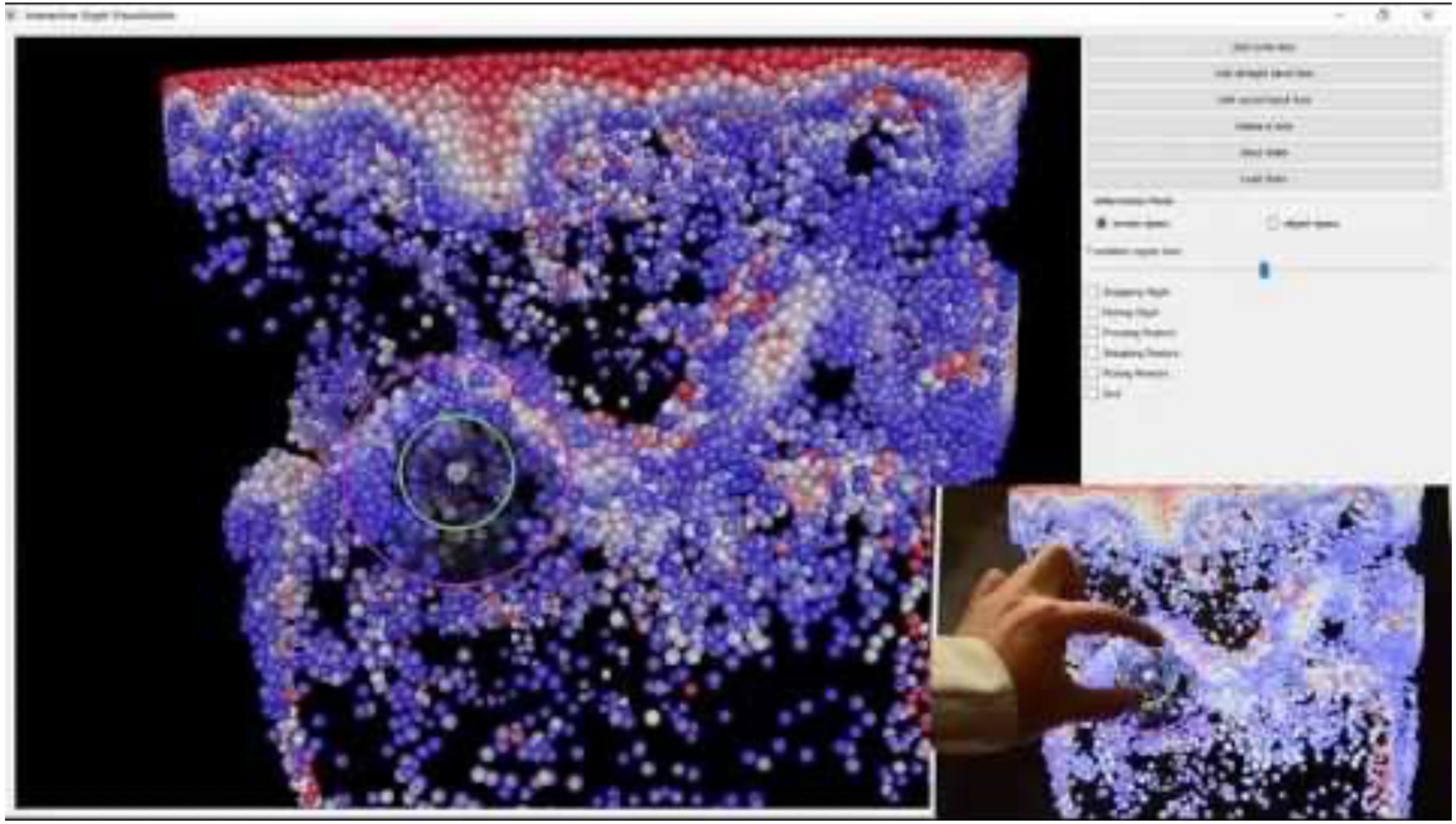
- Use geometric distortion of the contextual regions to make room for the details in the focus region(s) – Fisheye Lens





Focus and Context: Distort Geometry

- Use geometric distortion of the contextual regions to make room for the details in the focus region(s) – GlyphLens

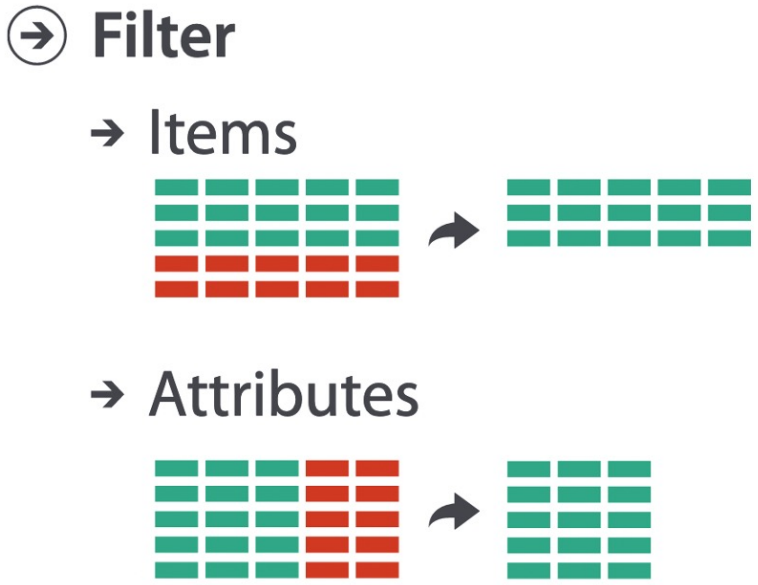


Filtering and Aggregation

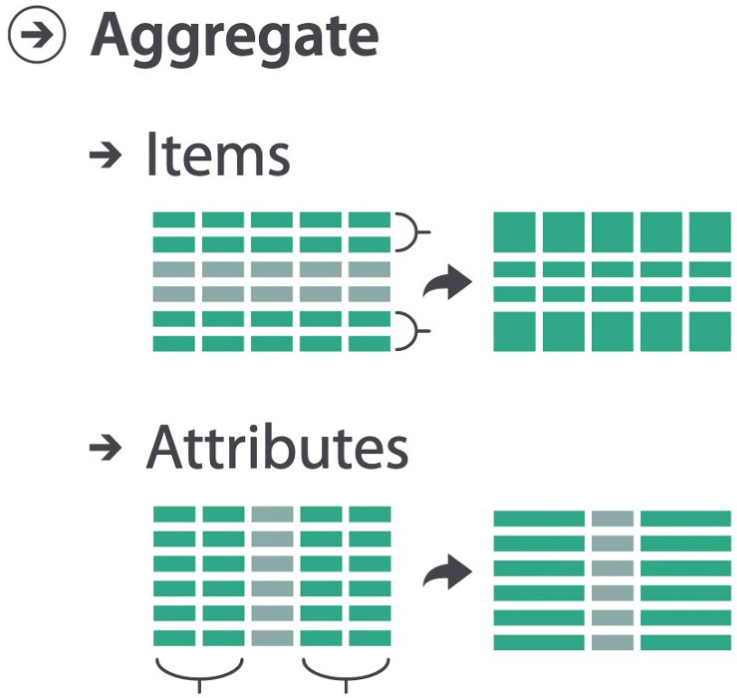
- Purpose: Complexity reduction in exploratory visual data analysis
- Reduce amount of data shown
  - Strategy for complexity reduction
  - Be careful not to hide important details
  - Can reduce items and/or attributes

Filtering vs Aggregation

- Filter: Eliminate data elements

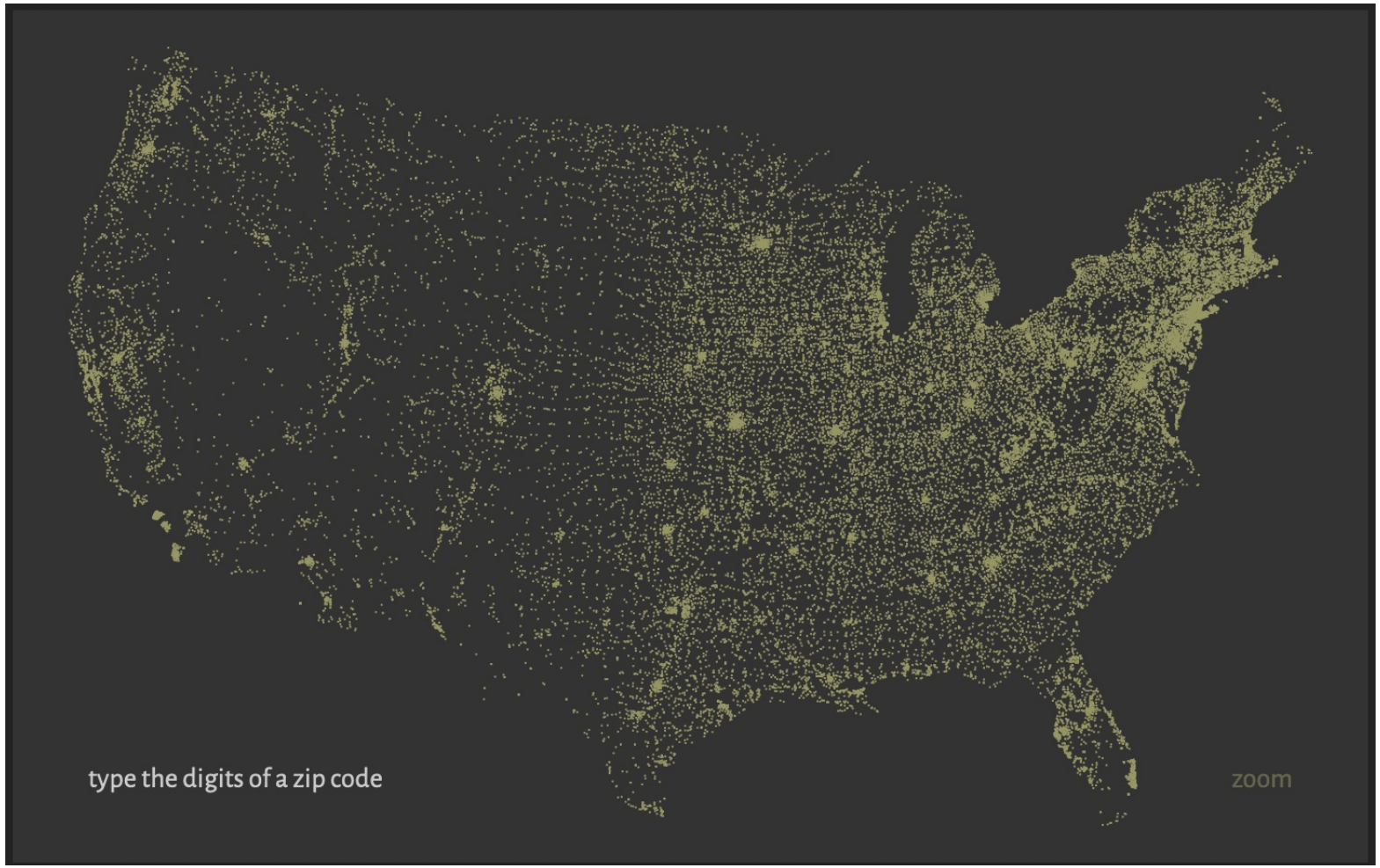


- Aggregate: Create new elements from multiple raw elements



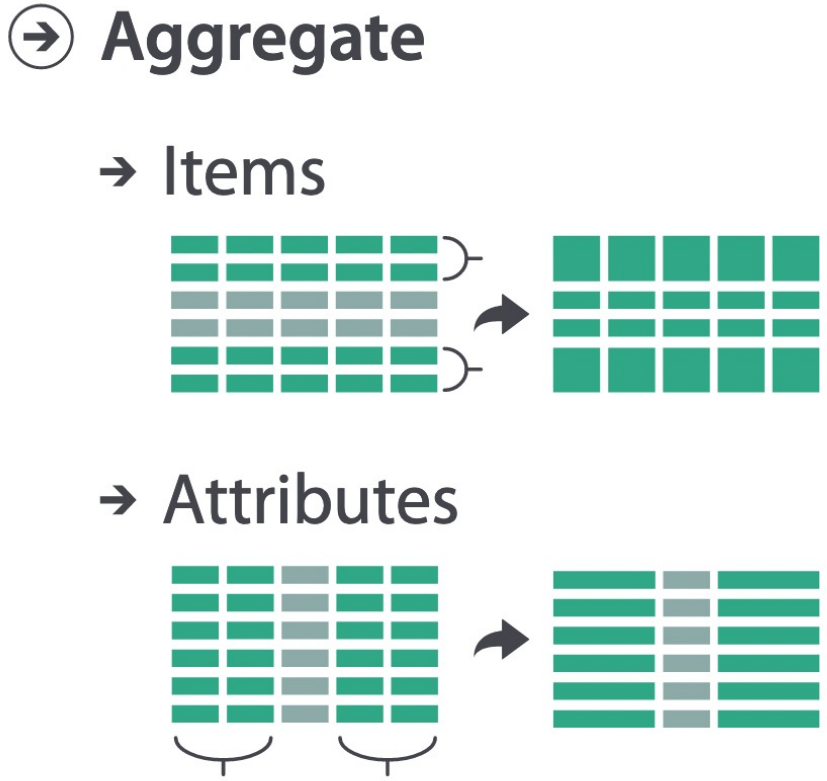
Filtering: Dynamic Query

http://benfry.com/zipdecode/



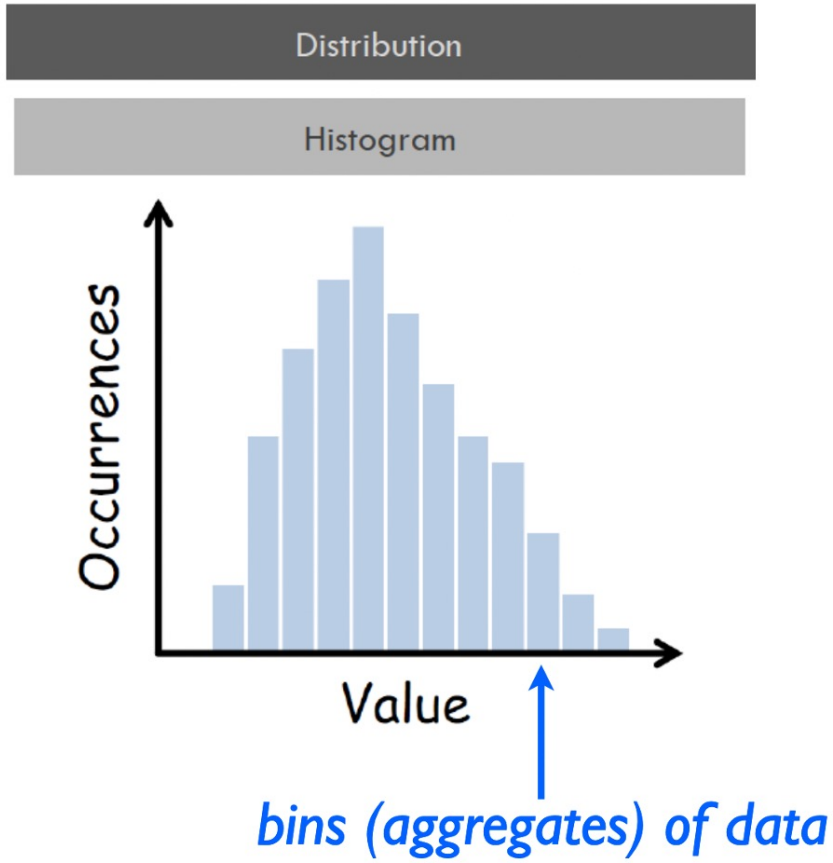
Aggregation

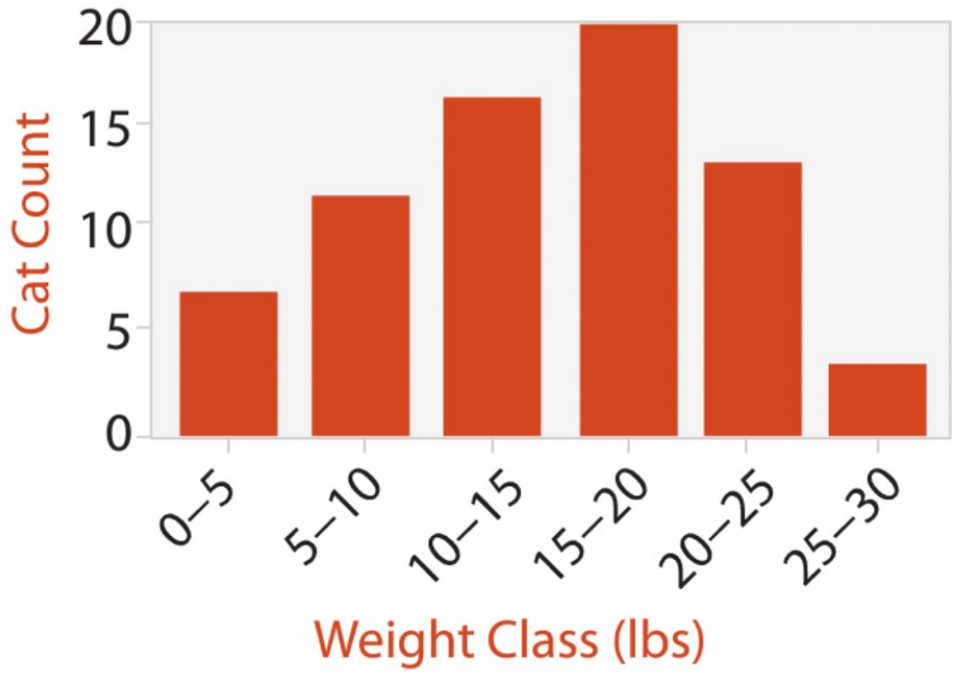
- Aggregate = Create new element representing multiple raw elements



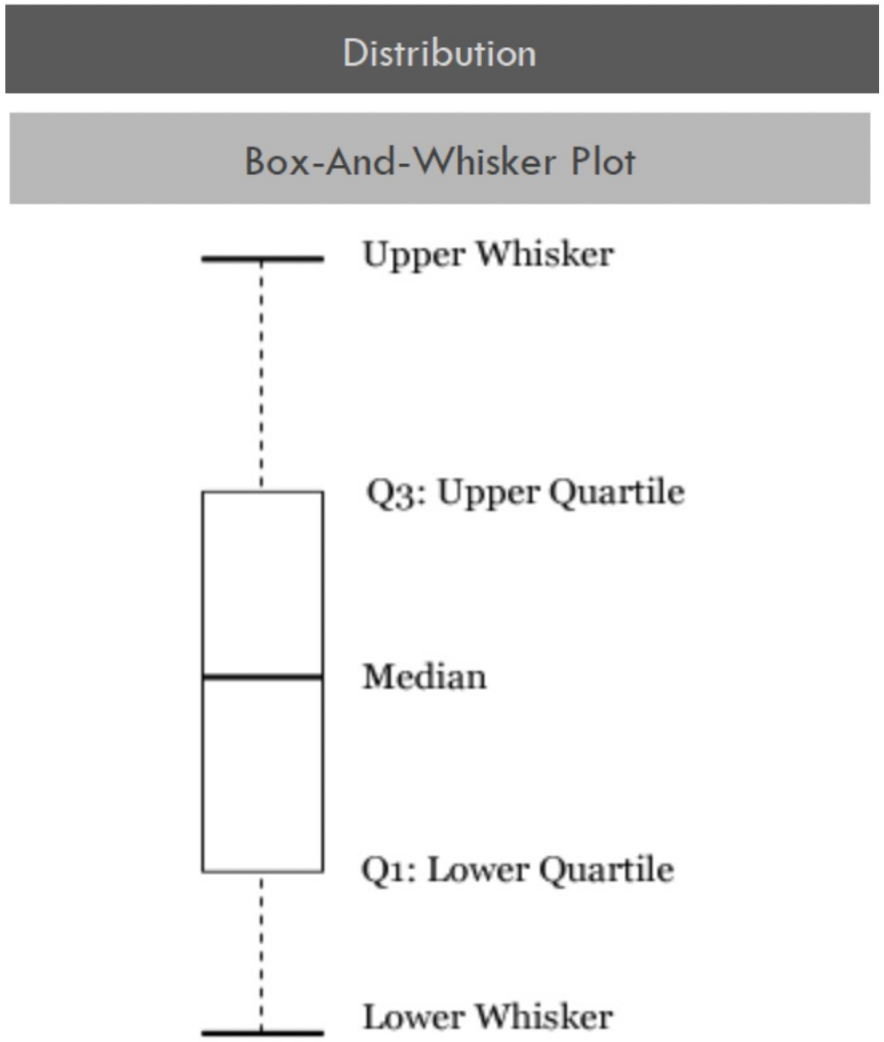
- How to Aggregate?
  - Item aggregation
  - Attribute aggregation (i.e., dimensionality reduction)
  - Spatial aggregation

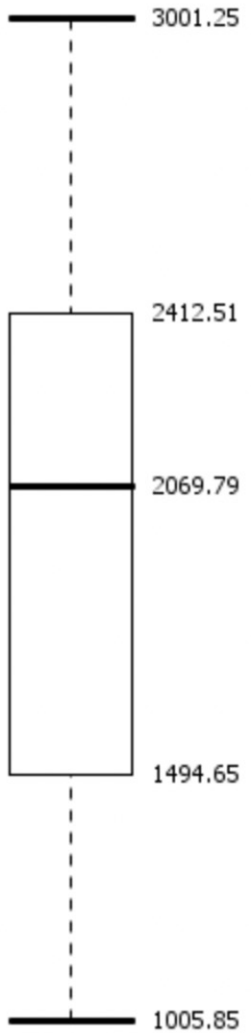
Aggregate Items: Histograms





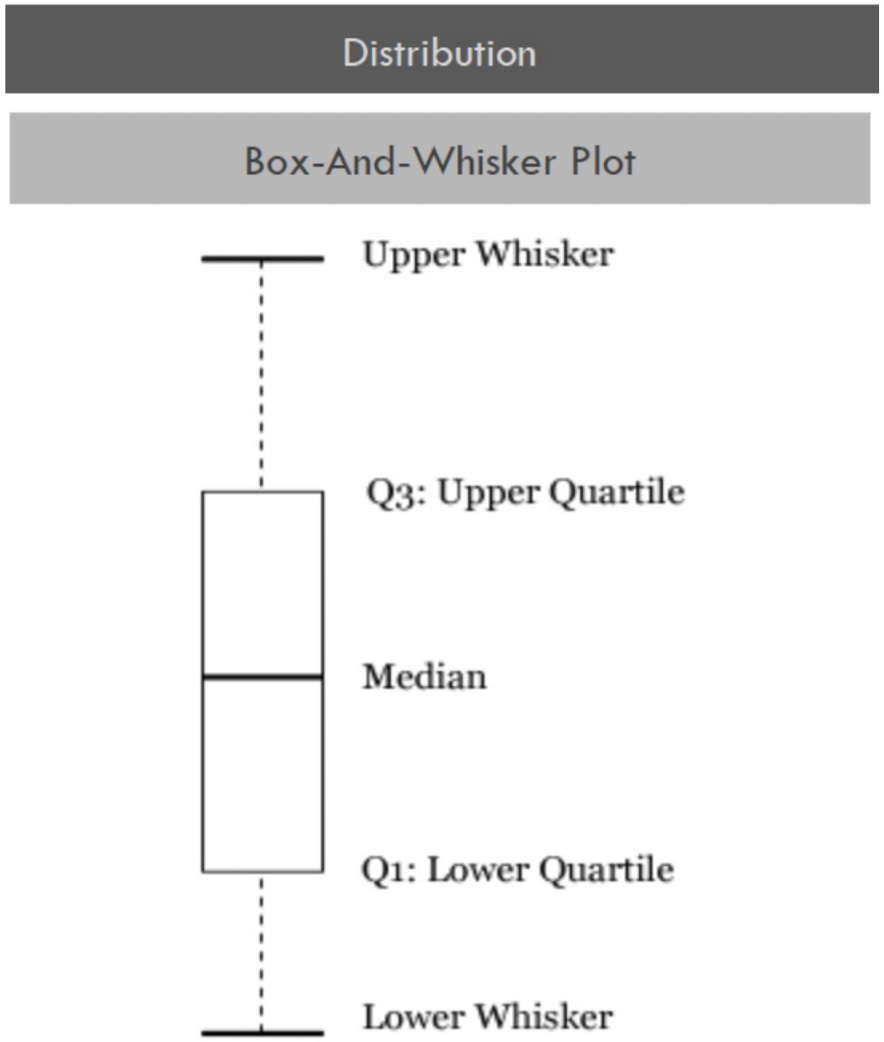
Aggregate Items: Box and Whisker Plot





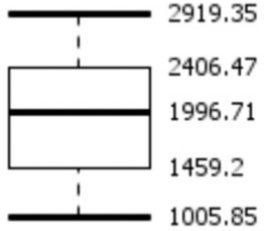
Citizen Nr.	Income	Key Value
25	3,001.25	Maximum
24	2,996.45	
23	2,919.35	
22	2,787.02	
21	2,784.72	
20	2,696.83	
19	2,412.51	Q3: 0.75 * 25 = 18.75 = 19.
18	2,400.43	
17	2,367.84	
16	2,333.37	
15	2,285.53	
14	2,214.87	
13	2,069.79	Median: 0.5 * 25 = 12.5 = 13.
12	1,923.62	
11	1,819.22	
10	1,773.34	
9	1,597.54	
8	1,589.48	
7	1,494.65	Q1: 0.25 * 25 = 6.25 = 7.
6	1,423.74	
5	1,391.92	
4	1,334.88	
3	1,184.53	
2	1,125.78	
1	1,005.85	Minimum

Aggregate Items: Box and Whisker Plot



+10345.67

Mean = 2,303.43,
Median = 1996.71



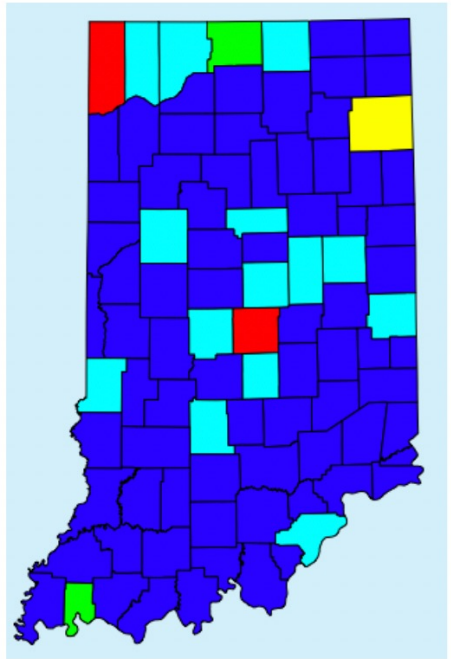
Citizen Nr.	Income	Key Value
24	10,345.67	Maximum
23	2,919.35	Upper Bound
22	2,787.02	
21	2,784.72	
20	2,696.83	
19	2,412.51	
18	2,400.43	Q3: (18. + 19.)/2 = 2,406.47
17	2,367.84	
16	2,333.37	
15	2,285.53	
14	2,214.87	
13	2,069.79	
12	1,923.62	Median: (12. + 13.) / 2 = 1,996.71
11	1,819.22	
10	1,773.34	
9	1,597.54	
8	1,589.48	
7	1,494.65	
6	1,423.74	Q1: (6. + 7.) / 2 = 1,459.2
5	1,391.92	
4	1,334.88	
3	1,184.53	
2	1,125.78	
1	1,005.85	Minimum / Lower Bound

Spatial Aggregation

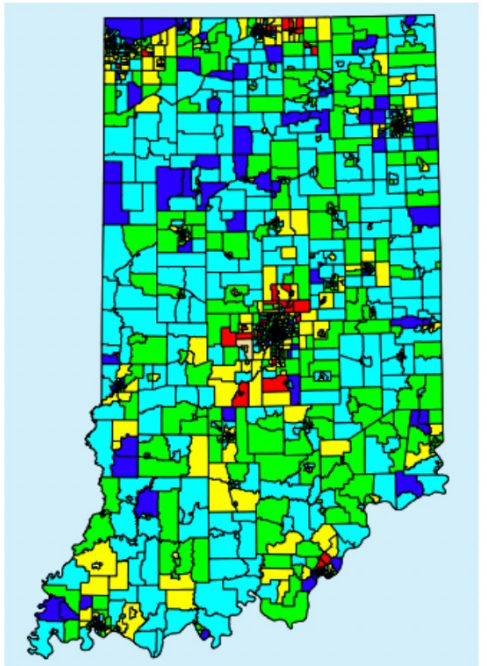
- Analysis of data using aggregated units
- Selecting the correct aggregate units of analysis is critical

Modifiable areal unit problem (MAUP): Boundary definition can dramatically change data analysis

Maps of household population in US state Indiana



By County



By Census Tract

Not as uniform as county map implies!

Study Materials for Lecture 11

- Visualizing High-Dimensional Data: Advances in the Past Decade; S. Liu et al., TVCG2016
- t-SNE: https://distill.pub/2016/misread-tsne/
- UMAP: https://pair-code.github.io/understanding-umap/

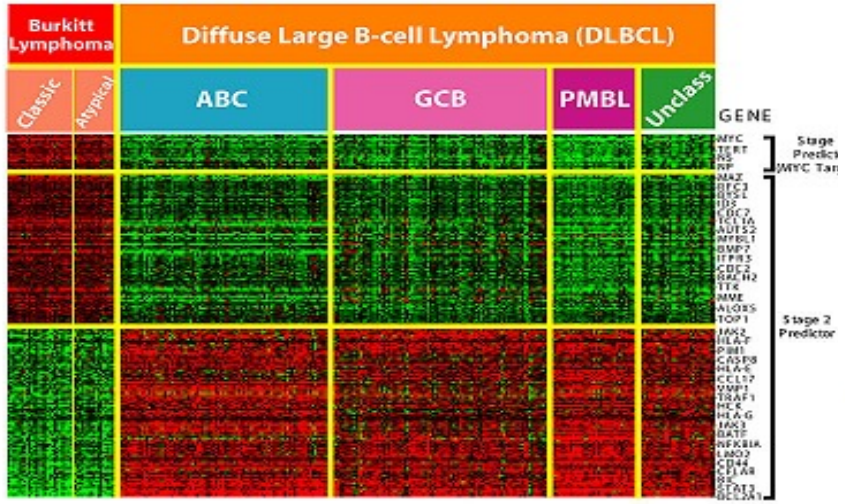
High Dimensional Data

- In statistics, high dimensional data is a data where the number of attributes (features) are larger than the number of samples
- In practice, often, when a data set has large number of attributes, it is also referred to as high dimensional data
  - Examples: biological data, gene expression data, social media user data, network data, etc.

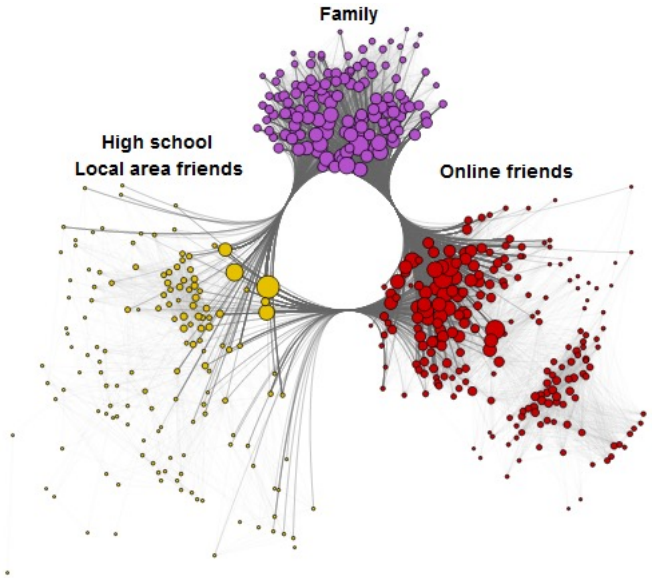
Dataset	Features	GP	DT	NB	KNNs	SVMs	RF
Adenocarcinomas(58)	50	99.83±0.009	73.33±1.6	87.66±3	86±1.3	91.33±0.7	77.67±1.5
	100	98.46±0.08	76.33±3.3	90±3.3	93.3±1.0	91.67±1.2	89.67 ± 1.5
	150	97.14±0.9	86±2.7	90±3.3	91.67±1.3	95±1.6	89.67±1.5
Oral Mucosa(79)	50	99.95±0.002	79.82±1.8	74.82±3.44	57.14±6.6	81±1.4	78.3±2.2
	100	98.57±0.08	70.89±0.16	69.46±3.8	62.14±2.1	77.32±2.6	81.25±5.9
	150	96.93±0.17	68.39±0.95	65.71±2.7	64.64±1.3	81.75±1.4	78.75±6.7
B-Cells(79)	50	99.41±0.03	77.32±2.6	74.82±3.44	82.32±2.2	85±4.7	82±5.5
	100	97.28±0.15	72.32±4.2	76.07 ±3.0	77.57±2.2	88.75±3.5	83.75±5.1
	150	96.59±0.19	71.25±9	78.57±2.2	76.25±6.7	87.5±3.9	83.75±5.1
Placenta(76)	50	99.91±0.005	77.49±2.5	68.75±9.8	70.71±4.7	84.28 ± 0.45	84.28±0.45
	100	99.30±0.03	73.39±5.1	67.5±10.7	69.46±5.1	84.28±0.45	84.28±0.45
	150	97.95±0.11	70.71±4.2	68.75±9.8	69.46±5.1	84.28±0.45	81.6±1.2
Melanoma(83)	50	97.64±0.13	88.19±4.1	94.16±1.8	95.13±2.4	91.67±1.3	96.52±1.0
	100	97.06±0.16	84.3±2.9	92.77±1.67	95.13±1.53	96.3±2.8	97.77±0.7
	150	96.37±0.51	85.5±3.3	95.2±1.4	96.3±1.14	97.63±0.7	97.77±0.7
Breast cancer(97)	50	97.87±0.12	52.77±0.8	49.44±1.9	43.22±0.3	54.66±0.2	56±0.14
	100	96.75±0.18	52.55±2.5	49.33±1.9	47.22±11	51.55±1.2	55.88±0.1
	150	96.90±0.17	51.67±2.2	51.44±1.3	52.1±2.4	52.55±0.9	57±0.45
Skeletal Muscle(110)	50	99.24±0.04	65.45±2.2	69.09±7.4	74.54±0.57	89.09±2.2	83.63±0.5
	100	98.69±0.07	71.81±5.4	66.36±2.0	83.63±5.1	98.18±0.57	82.72±0.28
	150	98.27 ±0.09	63.36±0.8	68.18±1.43	85.45±1.72	97.27±2.0	81.81±2.8
Osteoarthritis(139)	50	99.90±0.005	71.97±1.5	72.74±3.7	78.4±0.46	86.97±3.1	78.46±1.9
	100	99.23±0.04	70±9.4	69.12±2.4	84.23±2.5	90.65±0.52	81.31±1.04
	150	98.73±0.07	82.03±0.8	67.69±2.9	83.51±2.7	94.23±0.68	77.74±2.17

https://doi.org/10.1371/journal.pone.0196385.t004

Tabular data



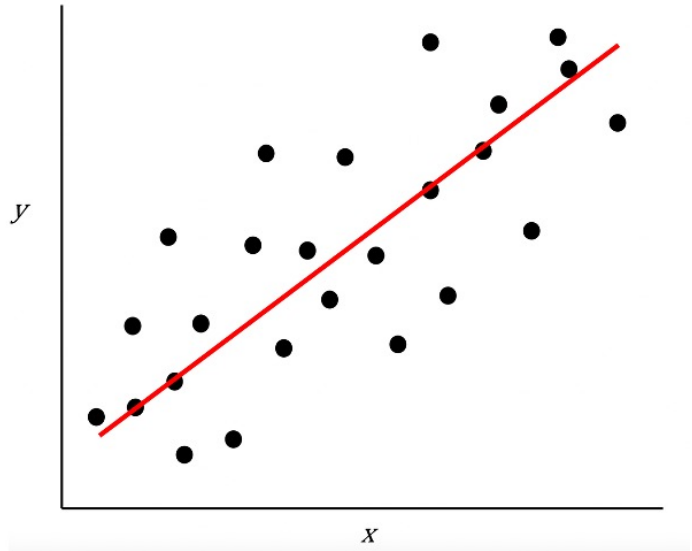
Microarray data

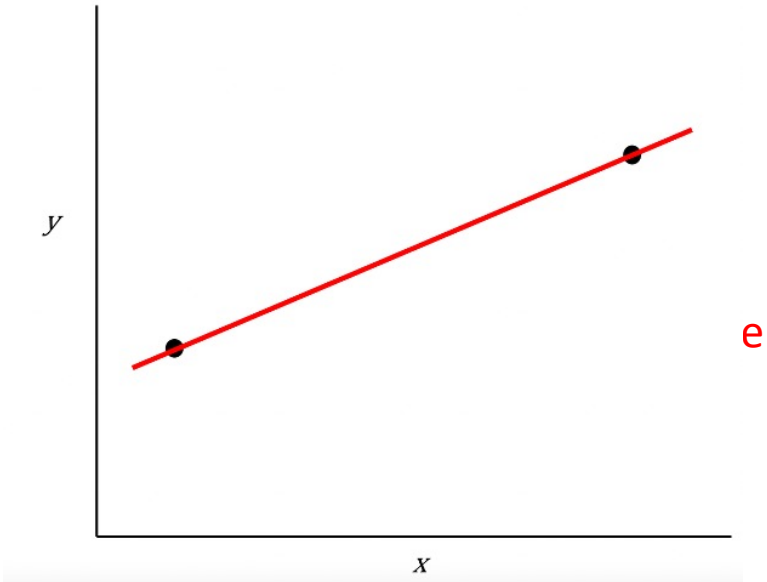


Graph data

Intuition: Why High-Dimensional Data Can Be A Problem?

- Imagine a situation in which the number of observations and features in a dataset are almost equal
- Effective number of observations per features is low
- Result: Models (Statistical or ML) can overfit and so less generalizable

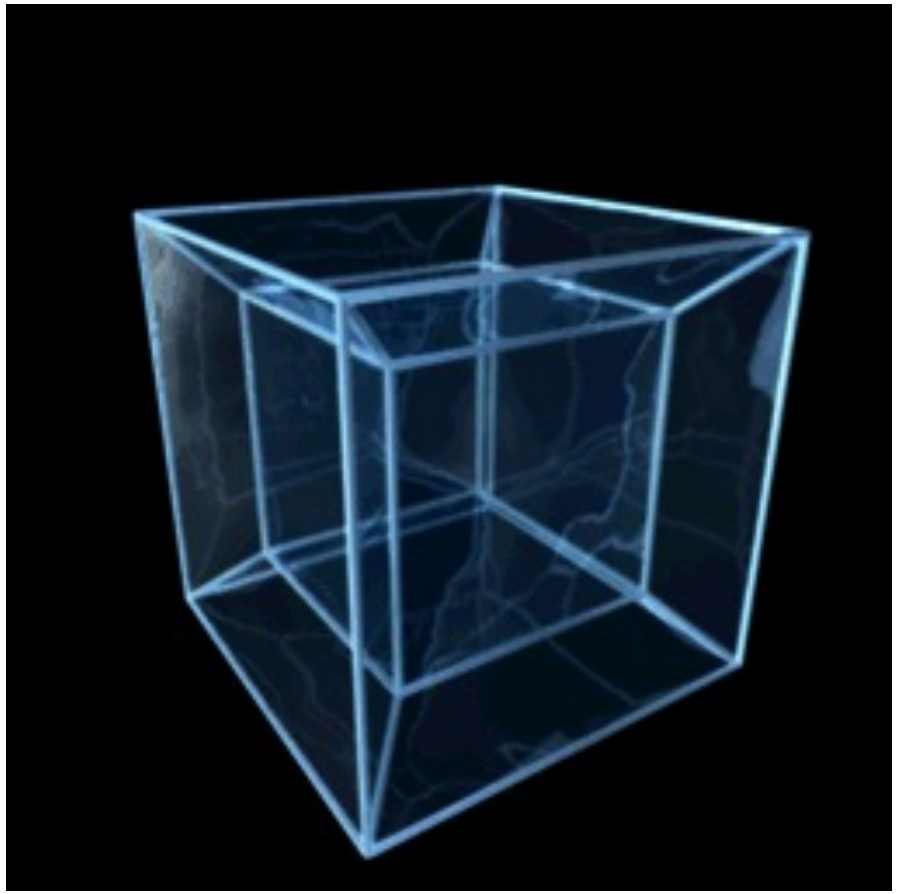




High dim. data, erfitting in regression

Understanding High-Dimensional Objects

- Feature vectors are typically high dimensional
- We do not understand such vectors well – why?
- Because we don’t learn to see high-dim objects when our vision system develops
- We only perceive 3D world!



3D projection of a 4D cube

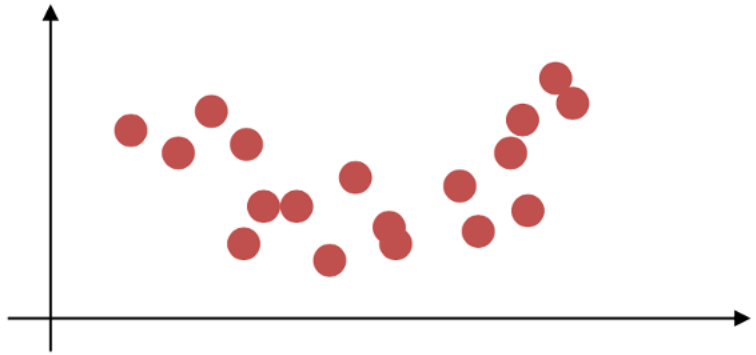
Curse of Dimensionality

- A phenomenon related with high dimensional data
  - Challenging to identify meaningful patterns while analyzing and visualizing the data
- With increasing dimensionality, the volume of the space increases rapidly, making the data sparse in high dimension
- To obtain a reliable result, the amount of data needed often grows exponentially with the dimensionality
- Distance computation between objects in high dimensional space becomes difficult

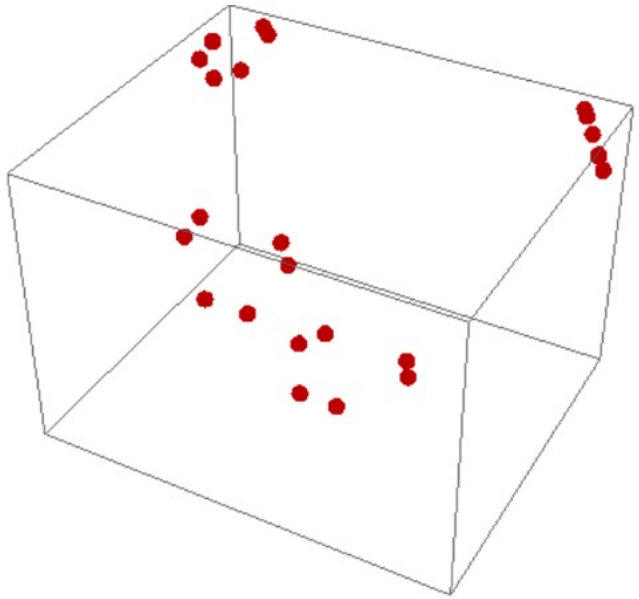
Sparseness in High Dimensional Space

- Space gets extremely sparse
  - with every extra dimension points get pulled apart further
  - distances become meaningless

1D – points are very close



2D – points spread apart

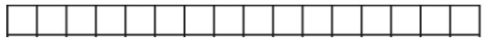


3D – getting even sparser

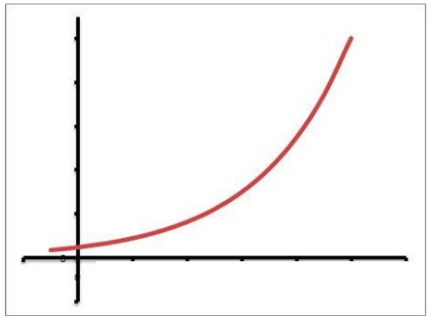
4D, 5D, … – sparseness grows further

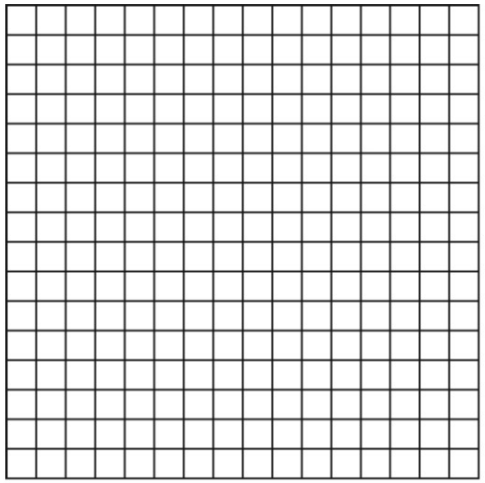
Space and Memory Management

- Indexing (and storage) gets very expensive with increasing dimensionality
  - Exponential growth in the number of dimensions

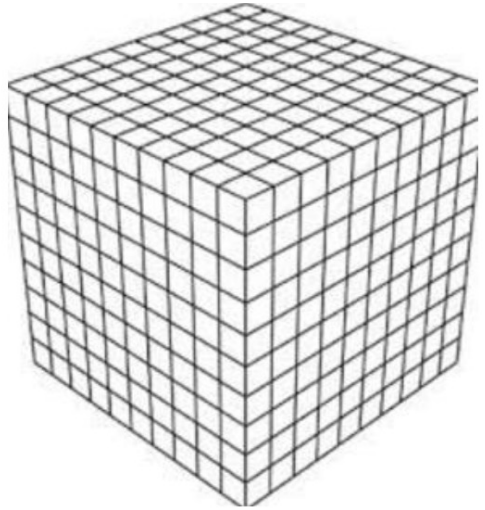


16 cells





$16^2 = 256$ cells

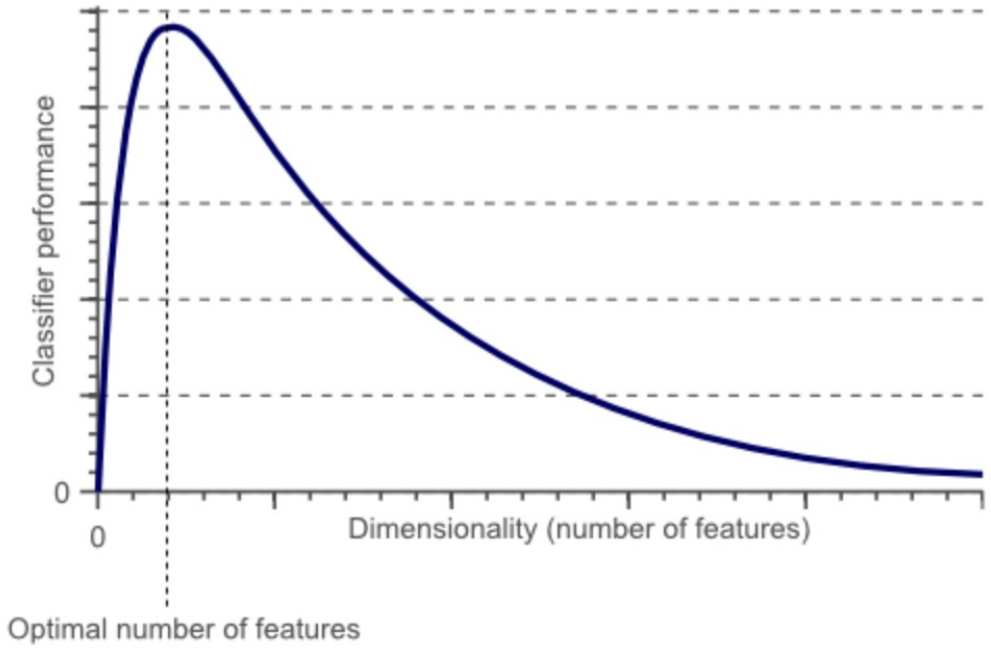


$16^3 = 4{,}096$ cells

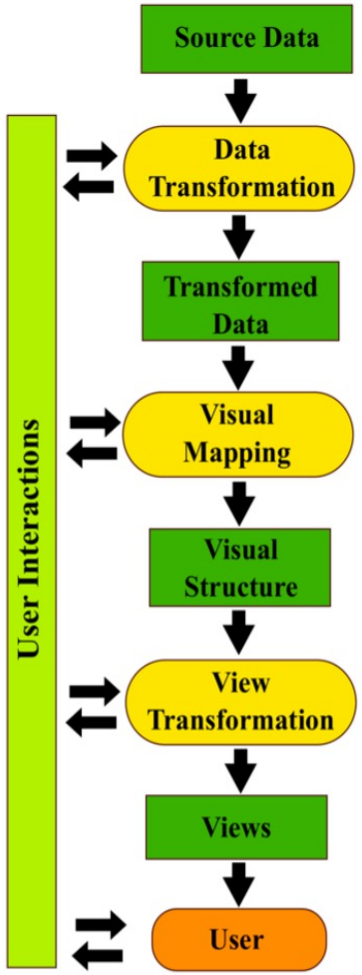
- 4D: 65k cells 5D: 1M cells 6D: 16M cells 7D: 268M cells

High Dimension: In Machine Learning

- Hughes' Phenomenon: With a fixed number of training samples, the average (expected) predictive power of a classifier or regressor first increases as the number of dimensions or features used is increased but beyond a certain dimensionality it starts deteriorating instead of improving steadily



High Dimensional Data Analysis and Visualization



Data Transformation

Dimension Reduction	Subspace Clustering	Regression Analysis	Topological Data Analysis
linear projection [21], [22],	Dimension Space Exploration	Optimization	Morse-Smale Complex
non-linear DR [23], [24],	[2], [31], [32],	Design Steering	[42], [43], [44], [45]
Control Points Projection [25], [26],	Subset of Dimension [33], [34],	[38], [39], [40],	Reeb Graph [46], [47], [48]
Distance Metric [27], [28],	Non-Axis-Parallel Subspace	Structural Summaries	Contour Tree [49], [50],
Precision Measures [29], [30]	[35], [36], [37]	[5], [41]	Topological Features [51], [52]

Visual Mapping

Axis Based	Glyphs	Pixel-Oriented	Hierarchy Based	Animation	Evaluation
Scatterplot Matrix [53],	Per-Element Glyphs	Jigsaw Map [67],	Dimension	GGobi [76],	Scatterplot Guideline
Parallel Coordinate [54],	[60], [61]	Pixel Bar Charts [68],	Hierarchy [71],	TripAdvisorND	[79], [80],
Radial Layout [55],	[62], [63],	Circle Segment [69],	Topology-based	[77],	PCPs Effectiveness [81],
Hybrid Construction	Multi-Object Glyphs	Value & Relation	Hierarchy [72], [73],	Rolling the	Radical Layout [82],
[56], [57], [58], [59]	[64], [65], [66]	Display [70]	Others [74], [75]	Dice [78]	Animation [83]

View Transformation

Illustrative Rendering	Continuous Visual Representation	Accurate Color Blending	Image Space Metrics
Illustrative PCP [84],	Continuous Scatterplot [89], [90],	Hue-Preserving	Clutter Reduction
Illuminated 3D Scatterplot [85],	Continuous Parallel Coordiante [91], [92],	Blending [95],	[97], [98],
PCP Transfer Function [86]	Splatterplots [93]	Weaving vs. Blending	Pargnostics [99],
Magic Lens [87], [88]	Splatting for PCPs [94]	[96]	Pixnostic [100]

Visualizing High-Dimensional Data: Advances in the Past Decade

Dimensionality Reduction: Why?

- Produce embedding of high dimensional data into low dimensional space
- Visual analysis of high dimensional data
- Useful for feature engineering in ML techniques
- Helps in finding redundant features from large scale data

Dimensionality Reduction Techniques

Linear methods

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

Non-linear methods

- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)
- Multidimensional Scaling (MDS)
- ISOMAP
- Locally linear embedding (LLE)
- Laplacian Eigenmap (LE)

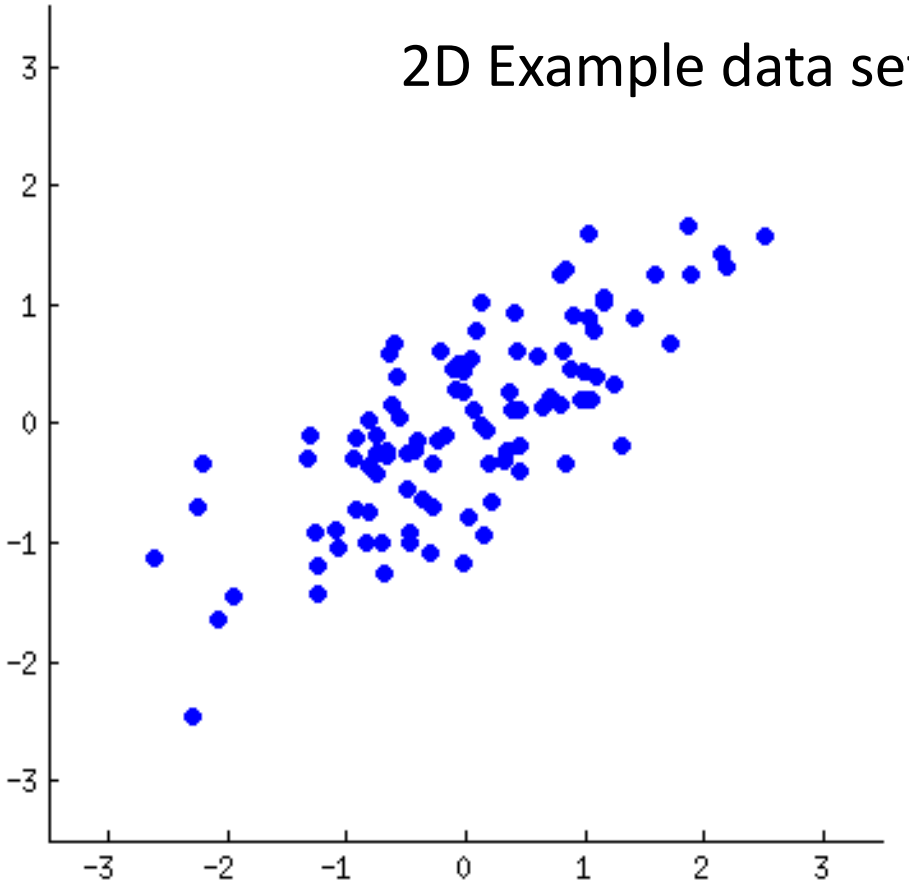
Principal Component Analysis (PCA)

- Unsupervised technique for extracting variance structure from high dimensional data to represent data in a lower dimensional space
  - An orthogonal (linear) projection of the data into a subspace so that the variance of the projected data is maximized
- Useful for:
  - Visualization
  - Further processing by machine learning algorithms
  - More efficient use of resources (e.g., time, memory, space)
  - In general: fewer dimensions → better generalization and modeling

Principal Component Analysis

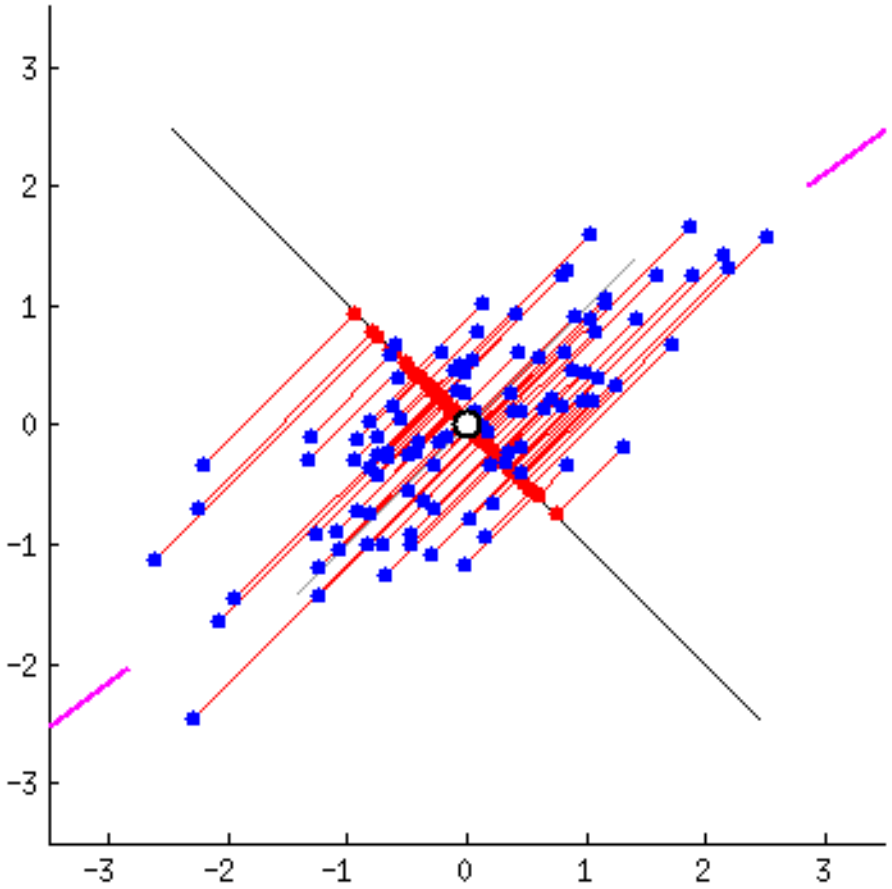
- A linear transformation that chooses a new coordinate system for the data such that greatest variance by any projection of the data comes to lie on the first axis (first principal component), the second greatest variance on the second axis, and so on….
- How to reduce data dimension with PCA?
  - Principal components are sorted in the order of their explained variance in the data
  - Eliminating later principal components

Principal Component Analysis



https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579

Principal Component Analysis



Animated view of how PCA works conceptually

https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579

Steps of PCA Algorithm

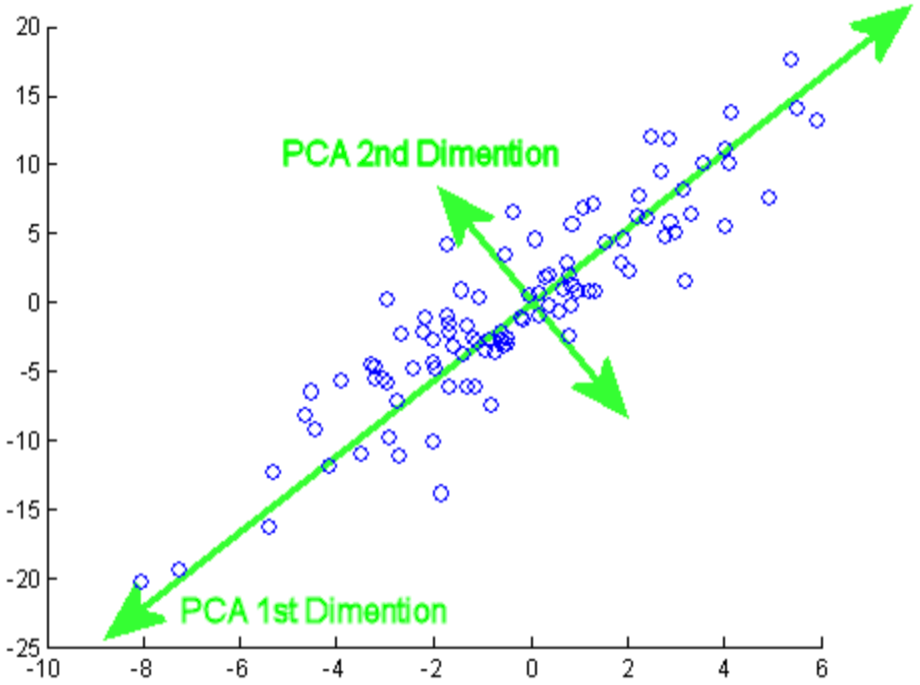
- Suppose we have a dataset with *n* records and *m* features for each record, i.e., data has *n* rows and *m* columns

1. Standardize the data: $x_{new} = \frac{x - \mu}{\sigma}$

2. Calculate the covariance matrix of the standardized data matrix
3. Calculate the eigen decomposition of the covariance matrix
   - Results in a list of eigenvalues and a list of eigenvectors
4. Sort eigenvalues in descending order to get a ranking for the eigenvectors (principal components) or axes of the new subspace

How to Project Data into a Lower Dimension?

- A total of k components (k < m) can be selected to create a projection subspace.
- The k eigenvectors are called principal components, that have the k largest eigenvalues
- Data can be projected into the k-dimensional subspace via matrix multiplication



PCA: Explained Variance

- Explained variance is a statistical measure of how much variation in a dataset can be attributed to each of the principal components
  - How much of the total variance is "explained" by each component
- Ordered (large to small) eigenvalues can help
- Explained variance for i^{th} principal component:

$$\frac{\lambda_i}{\lambda_1+\lambda_2+\ldots+\lambda_n}$$

λ_i : i^{th} eigen value of covariance matrix

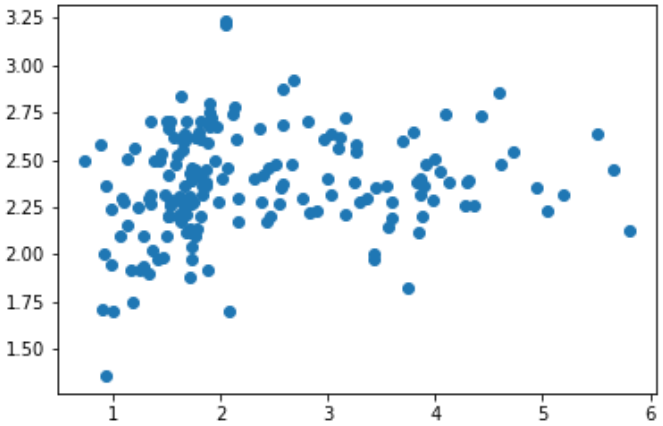
Visualization of Data using PCA

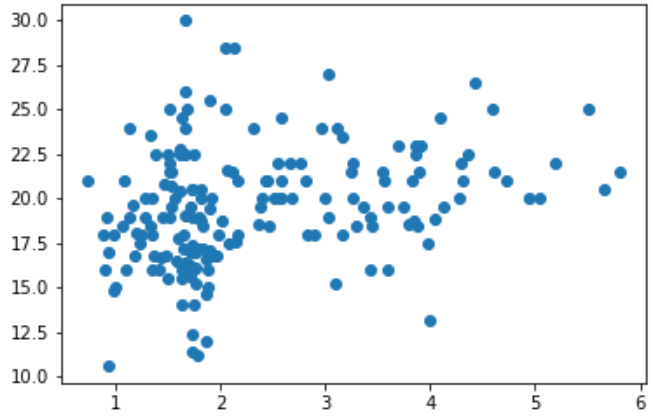
- Wine data set
  - 178 records, 13 features for each record

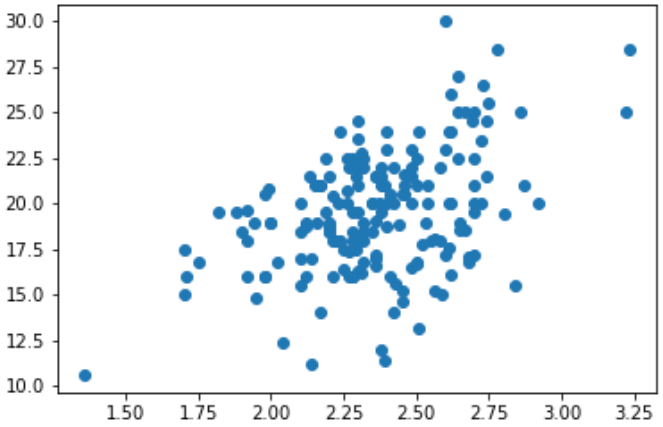
```
[[1.423e+01 1.710e+00 2.430e+00 ... 1.040e+00 3.920e+00 1.065e+03]
 [1.320e+01 1.780e+00 2.140e+00 ... 1.050e+00 3.400e+00 1.050e+03]
 [1.316e+01 2.360e+00 2.670e+00 ... 1.030e+00 3.170e+00 1.185e+03]
 ...
 [1.327e+01 4.280e+00 2.260e+00 ... 5.900e-01 1.560e+00 8.350e+02]
 [1.317e+01 2.590e+00 2.370e+00 ... 6.000e-01 1.620e+00 8.400e+02]
 [1.413e+01 4.100e+00 2.740e+00 ... 6.100e-01 1.600e+00 5.600e+02]]
```

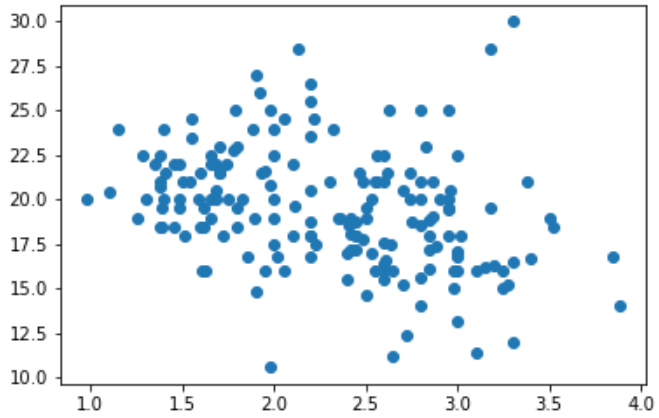
Visualization of Data using PCA

- Scatter plot for selected pairwise features (attributes)



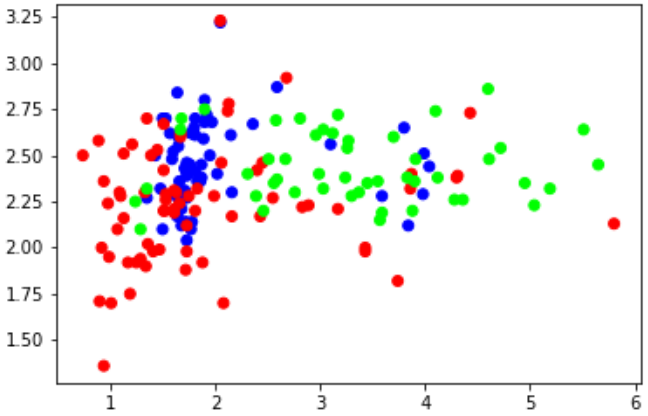


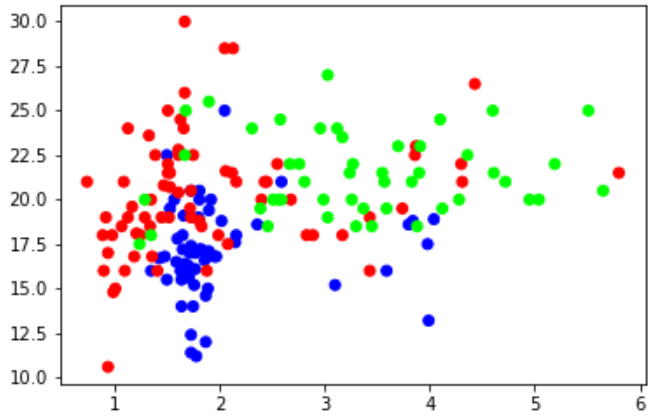


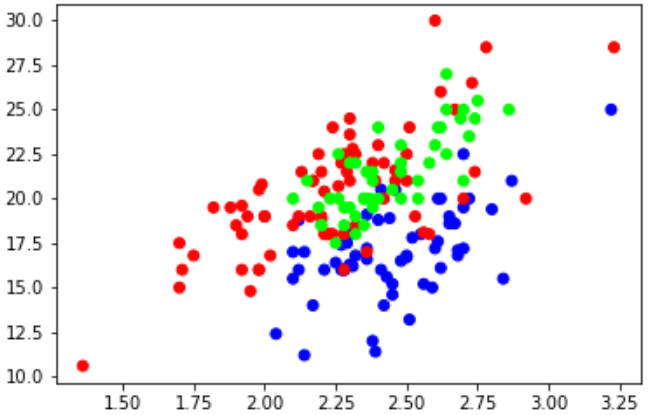


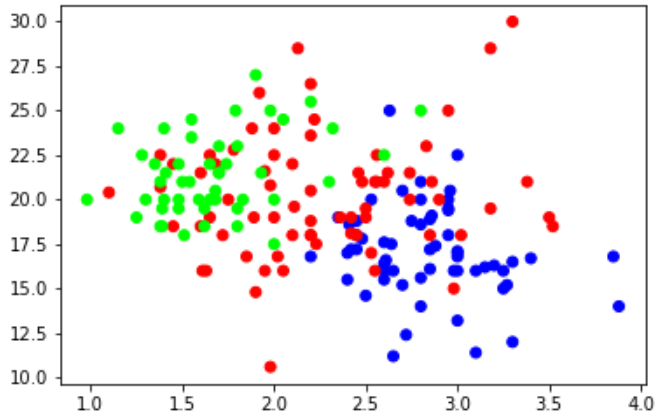
Visualization of Data using PCA

- Scatter plot for selected pairwise features (attributes)
  - Color points by class label



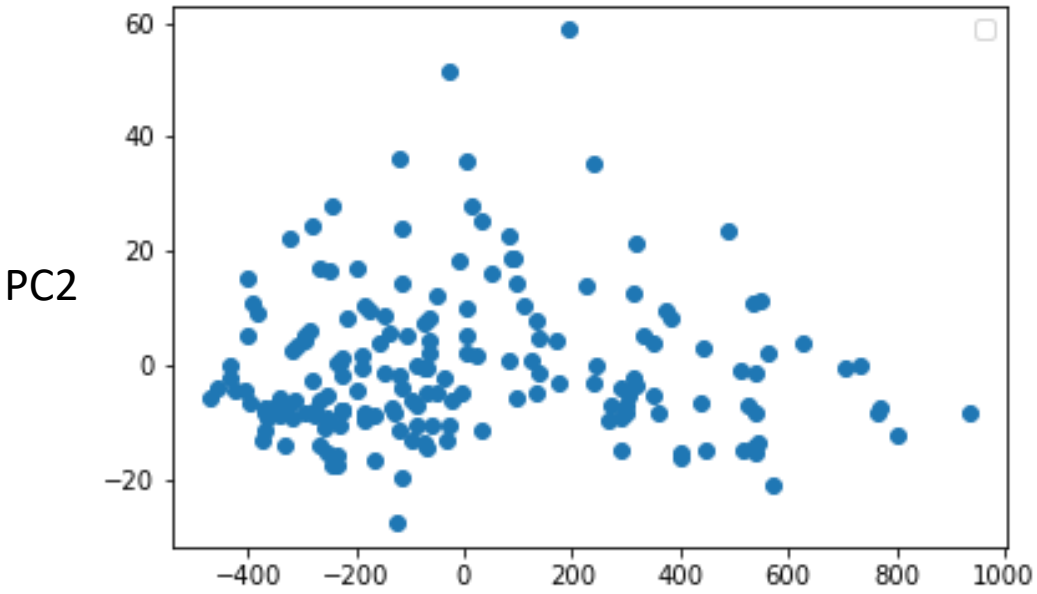




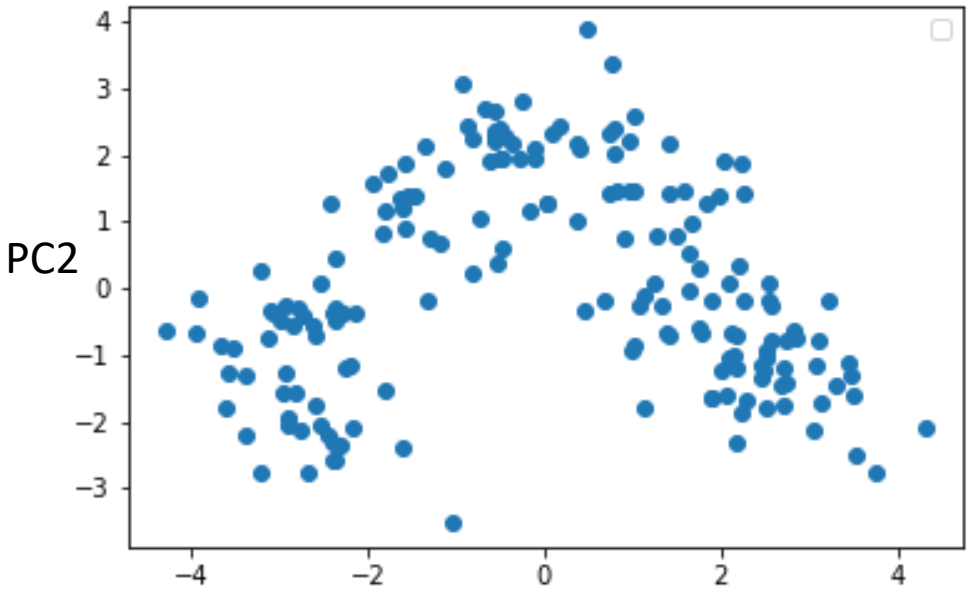


Visualization of Data using PCA

- Scatter plot using two first principal components as axes after PCA is applied



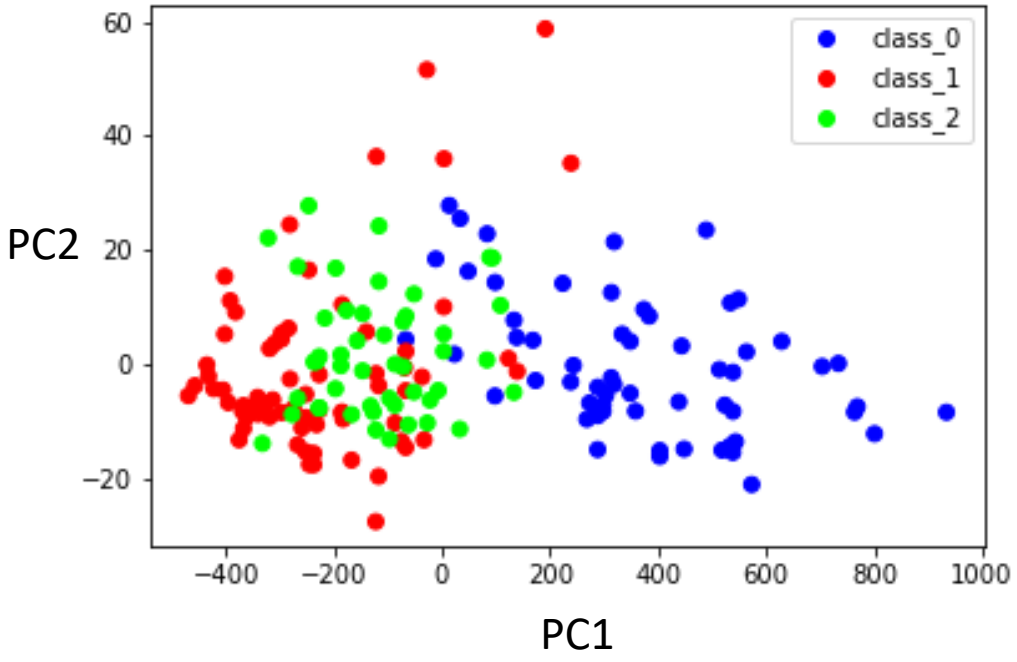
PCA with no standardization



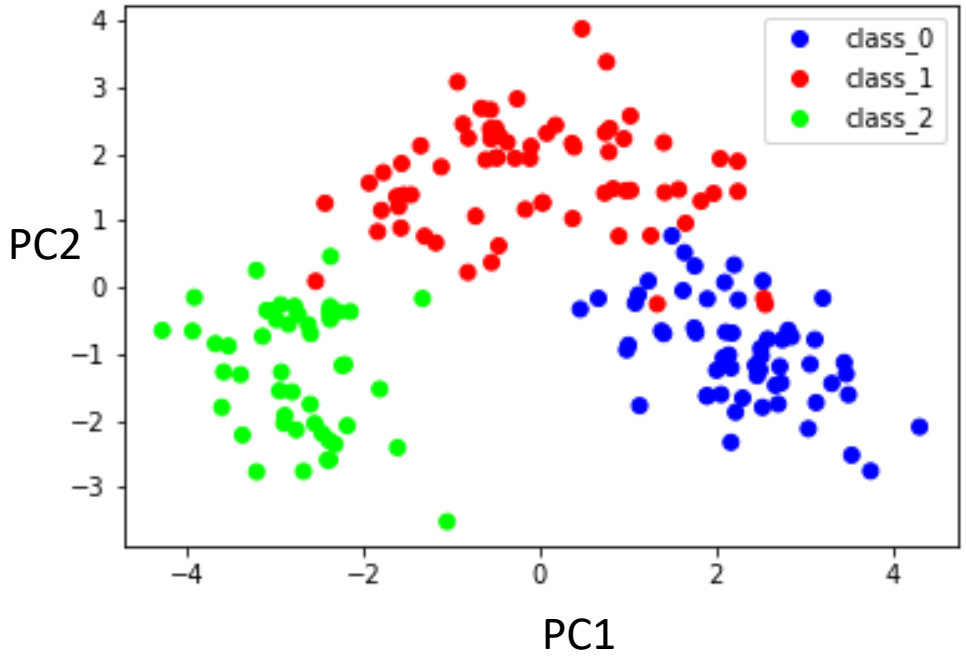
PCA with standardization

Visualization of Data using PCA

- Scatter plot using two first principal components as axes after PCA is applied
  - Color points by class label



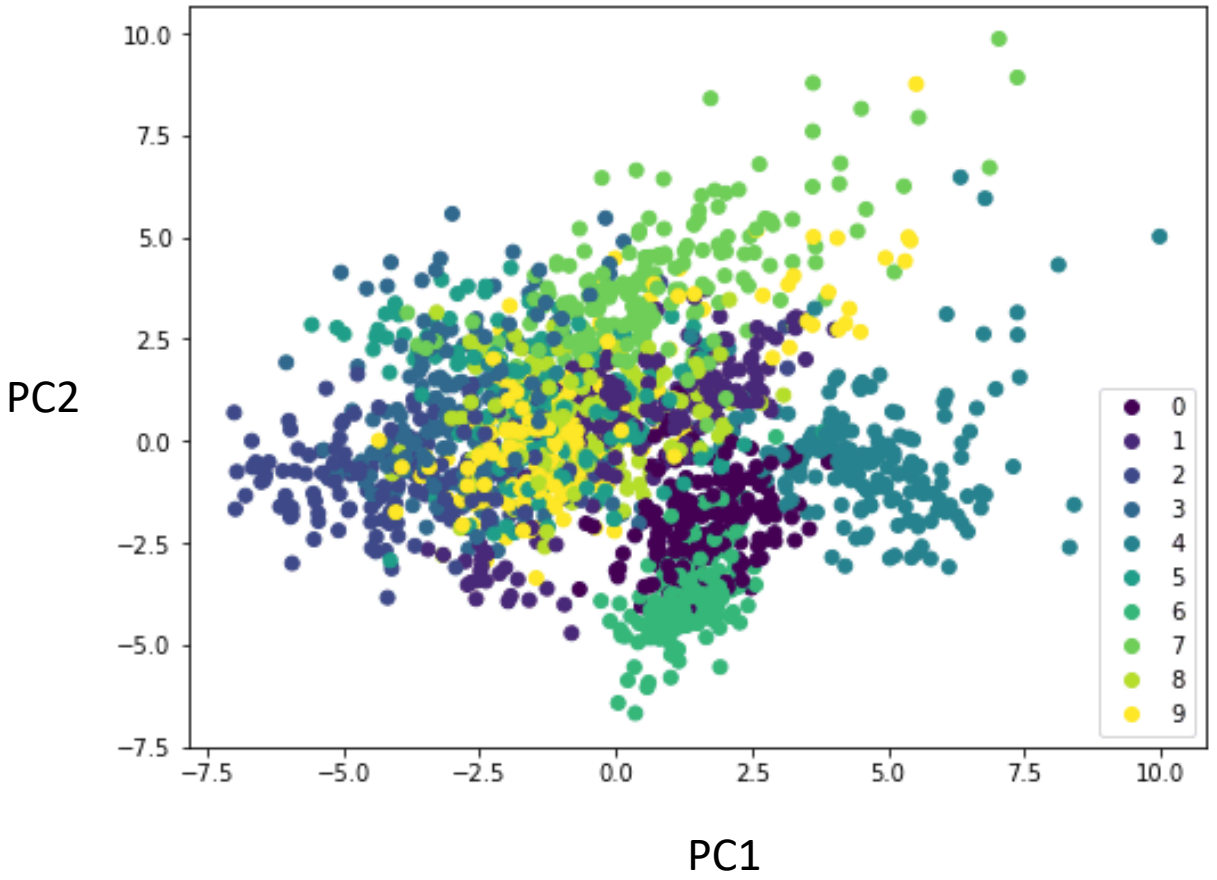
PCA with no standardization



PCA with standardization

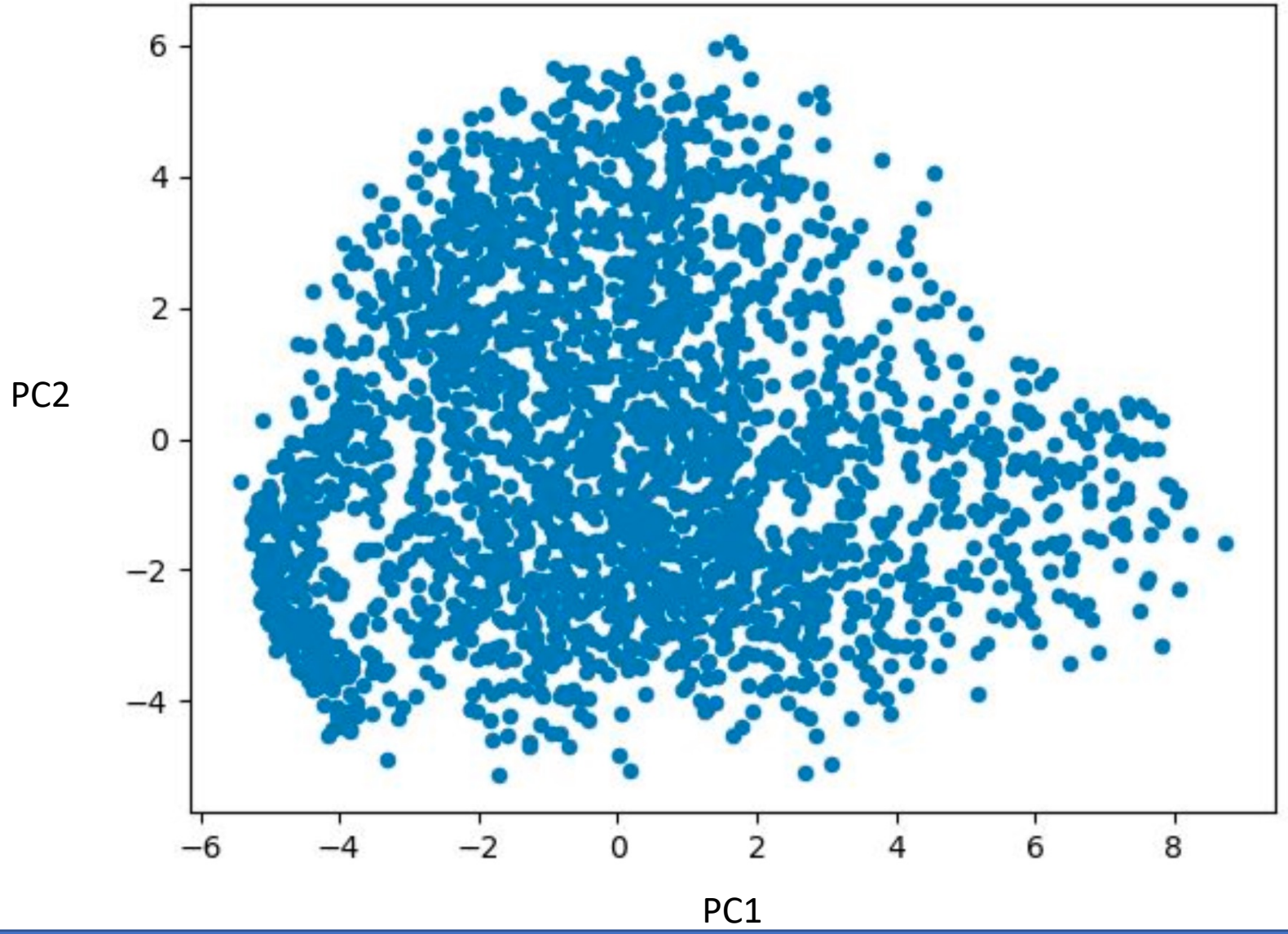
Visualization of Data using PCA

- 2D projection plot of hand-written digit data set with 10 classes
  - Not easily separable

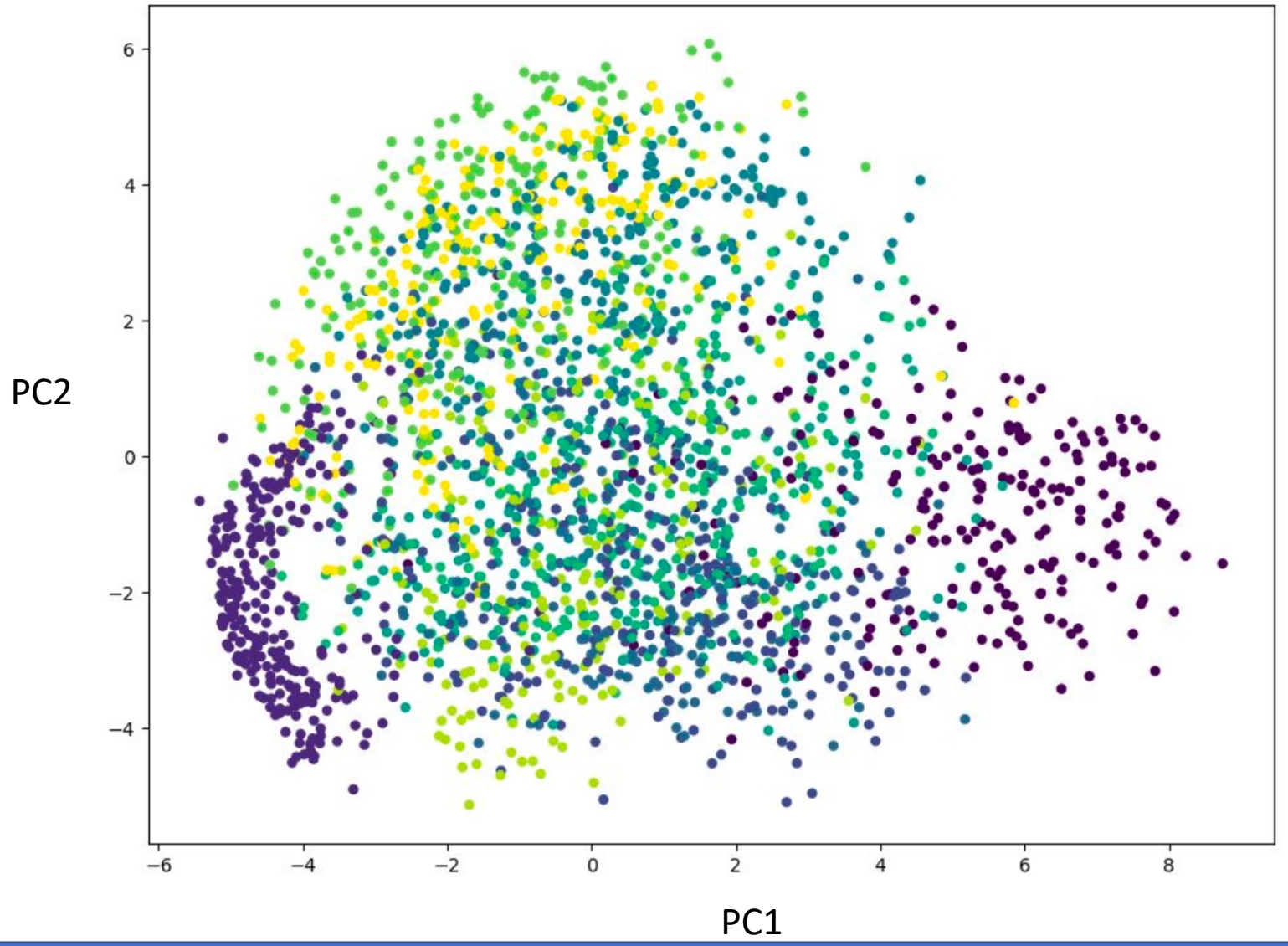


PCA on MNIST data

PCA on MNIST data



PCA on MNIST data



PCA is not able to separate classes clearly

Study Materials for Lecture 12

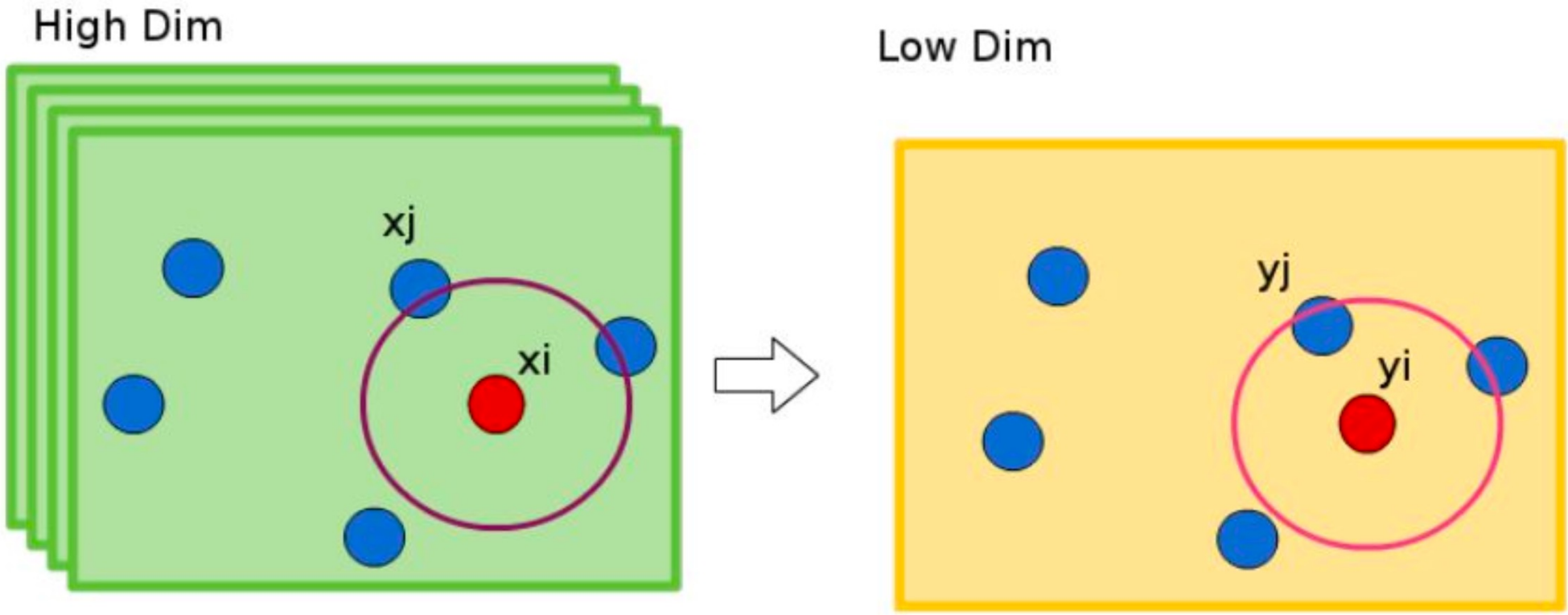
- Visualizing High-Dimensional Data: Advances in the Past Decade; S. Liu et al., TVCG2016
- t-SNE: https://distill.pub/2016/misread-tsne/
- UMAP: https://pair-code.github.io/understanding-umap/
- Footnotes in slides

t-Distributed Stochastic Neighbor Embedding (t-SNE)

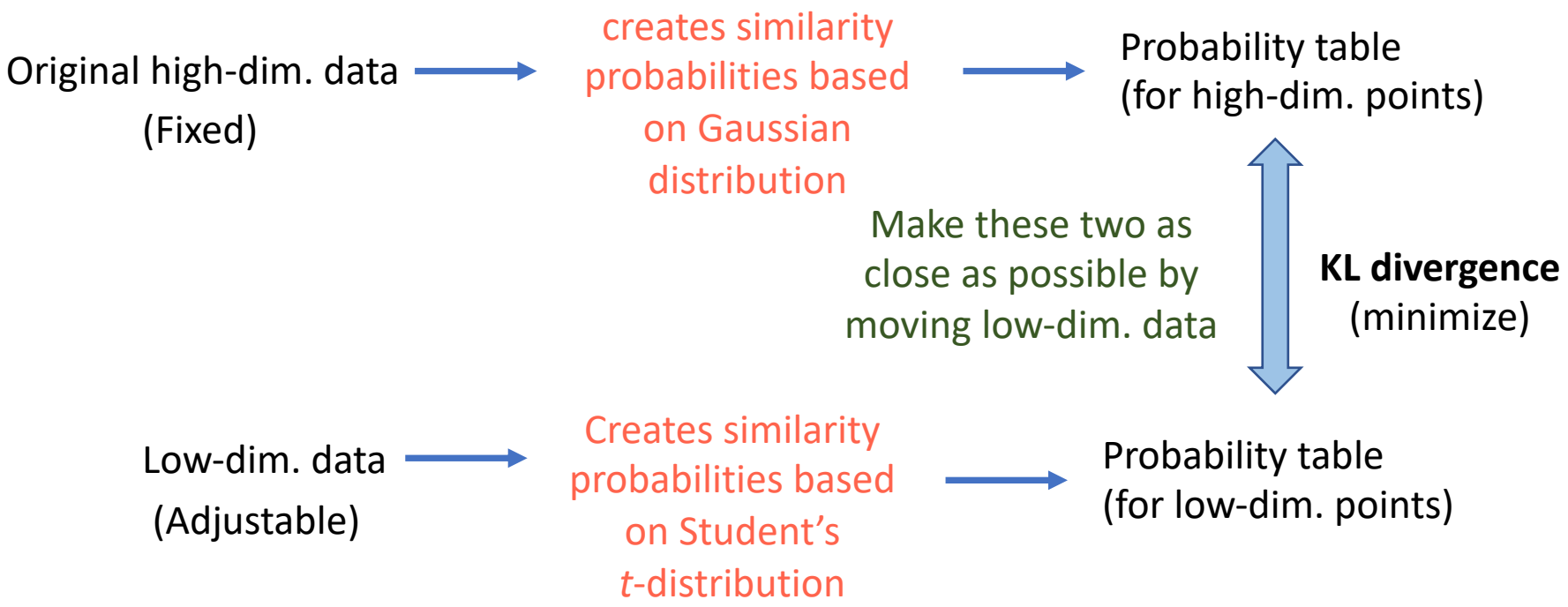
- PCA is not always effective in finding patterns in low dimensional visualization space
  - It is a linear algorithm, meaning that it cannot represent complex nonlinear relationship between features
- t-Distributed Stochastic Neighbor Embedding: A nonlinear dimensionality reduction technique
- t-SNE is specifically designed for high dimensional data visualization purposes
  - **Paper:** https://jmlr.org/papers/v9/vandermaaten08a.html
  - ~51,000 citations!!

t-SNE : Underlying Idea

Measure pairwise distances between high dimensional and low dimensional points

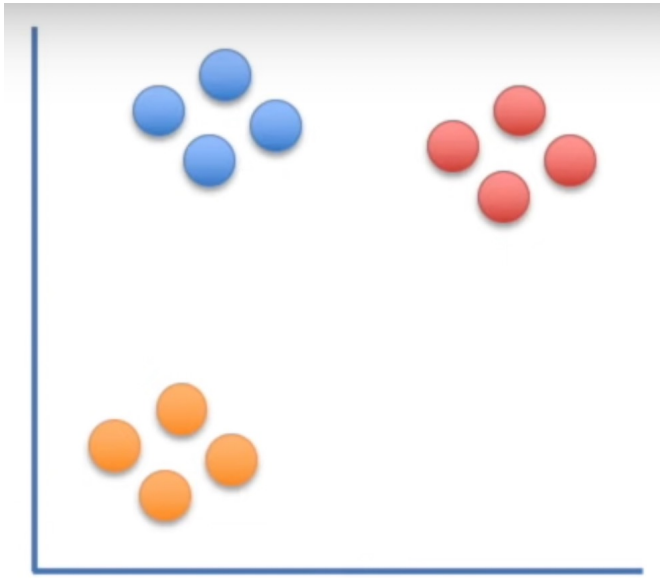


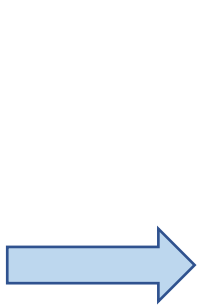
t-SNE : Underlying Idea

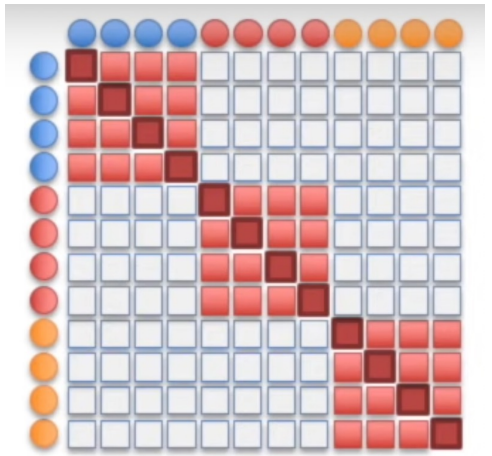


t-SNE : Underlying Idea

High Dim data

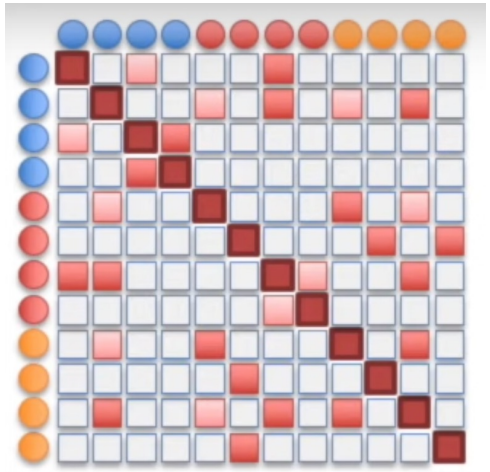






Similarity Probability table in high dimension

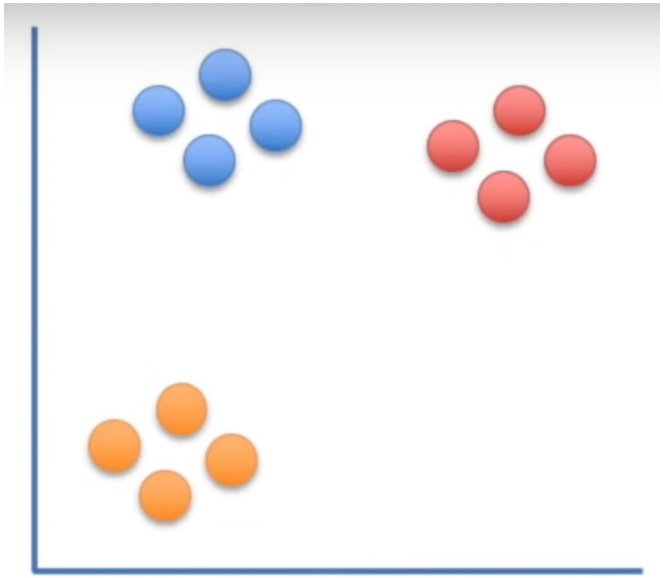
Low Dim data

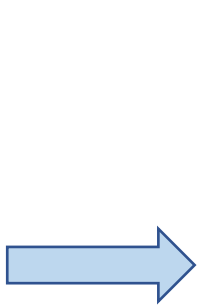


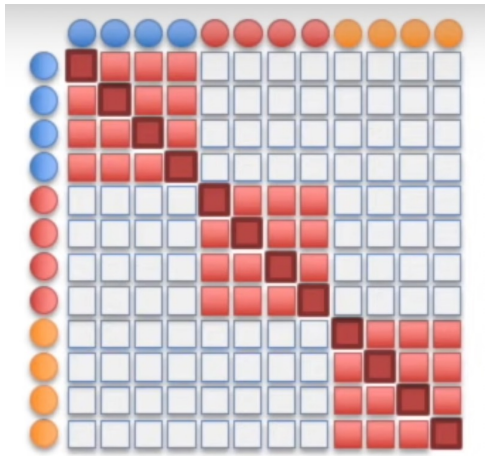
Initial Similarity Probability table in low dimension

t-SNE : Underlying Idea

High Dim data

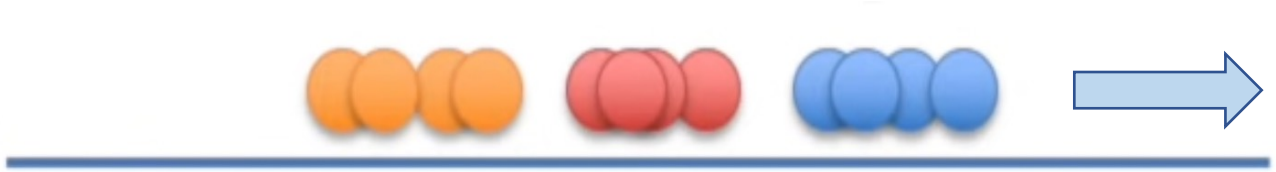


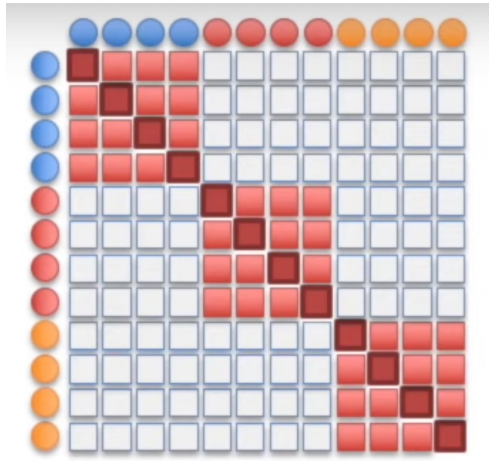




Similarity Probability table in high dimension

Low Dim data





Final Similarity Probability table in low dimension

t-SNE: Measure Distances and Optimize

- Similarity of datapoints in high dimensional space

$$p_{ij} = \frac{exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{k \neq l} exp(-||x_l - x_k||^2/2\sigma^2)}$$

- Similarity of datapoints in low dimensional space

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}$$

- Cost function: Minimize KL Divergence

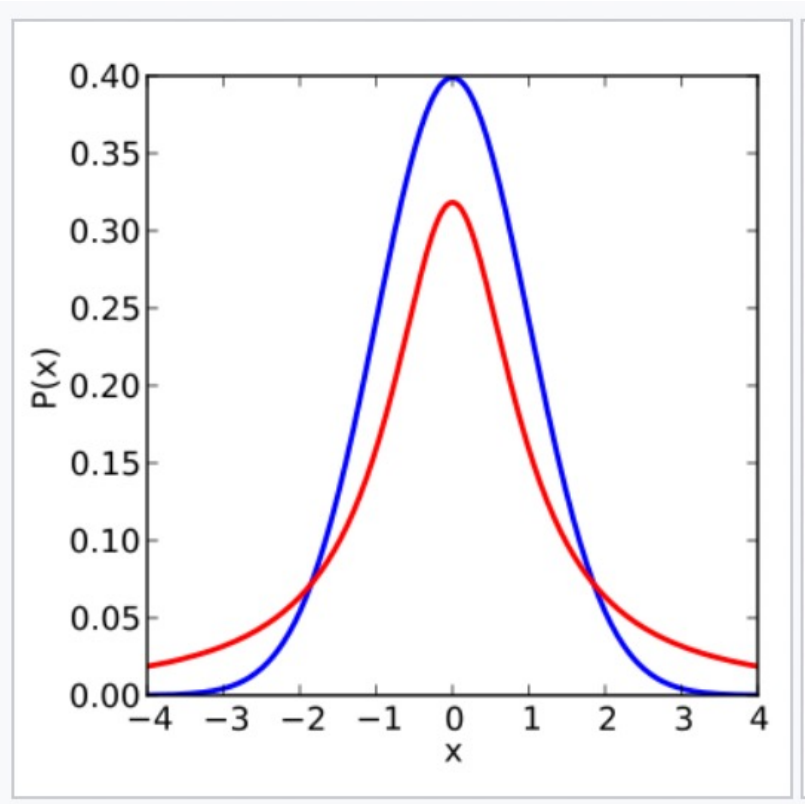
$$C = KL(P||Q) = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}}$$

KL Divergence: Kullback–Leibler divergence is a measure of how one probability distribution P is different from a second, reference probability distribution Q

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

What is the t in the t-SNE ?

- To measure distance between points in the low dimensional space, a student's **t**-distribution is used instead of a Gaussian distribution

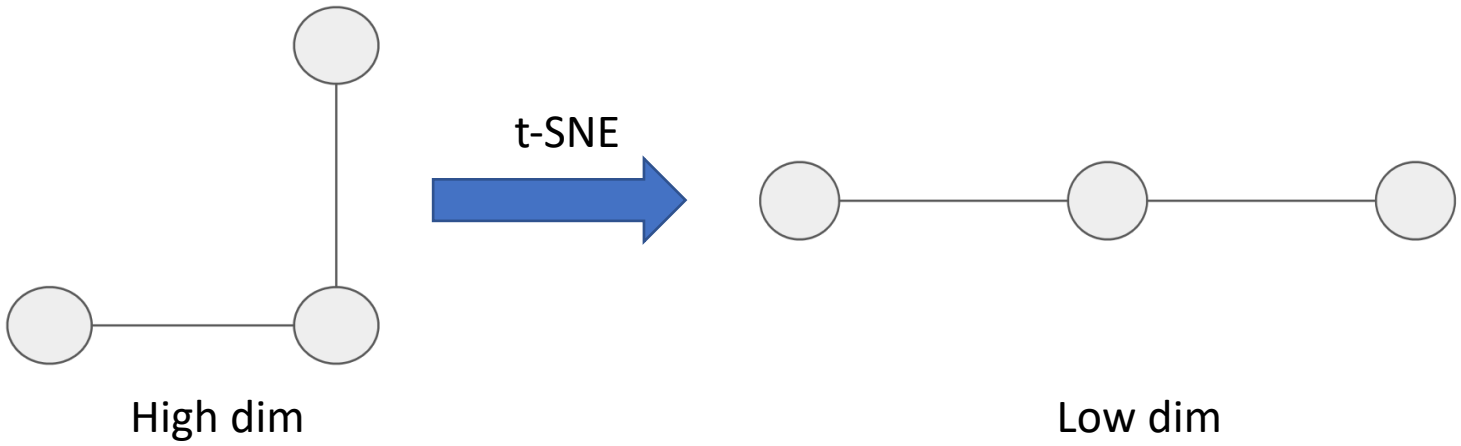


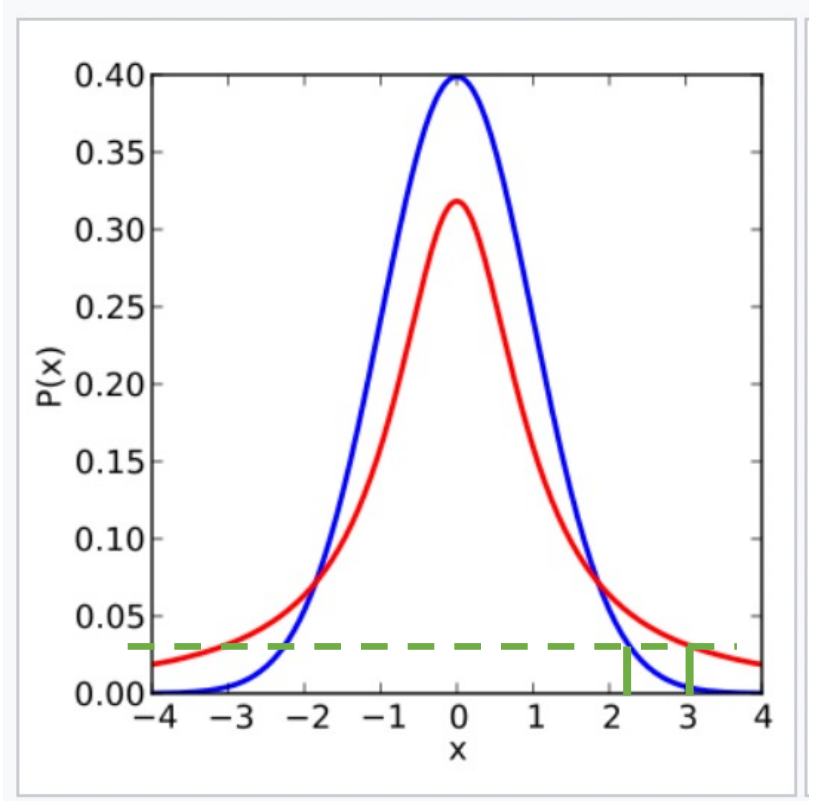
Blue: Gaussian distribution
Red: t-Distribution

- The student's t-distribution has a heavier tail compared to the Gaussian distribution

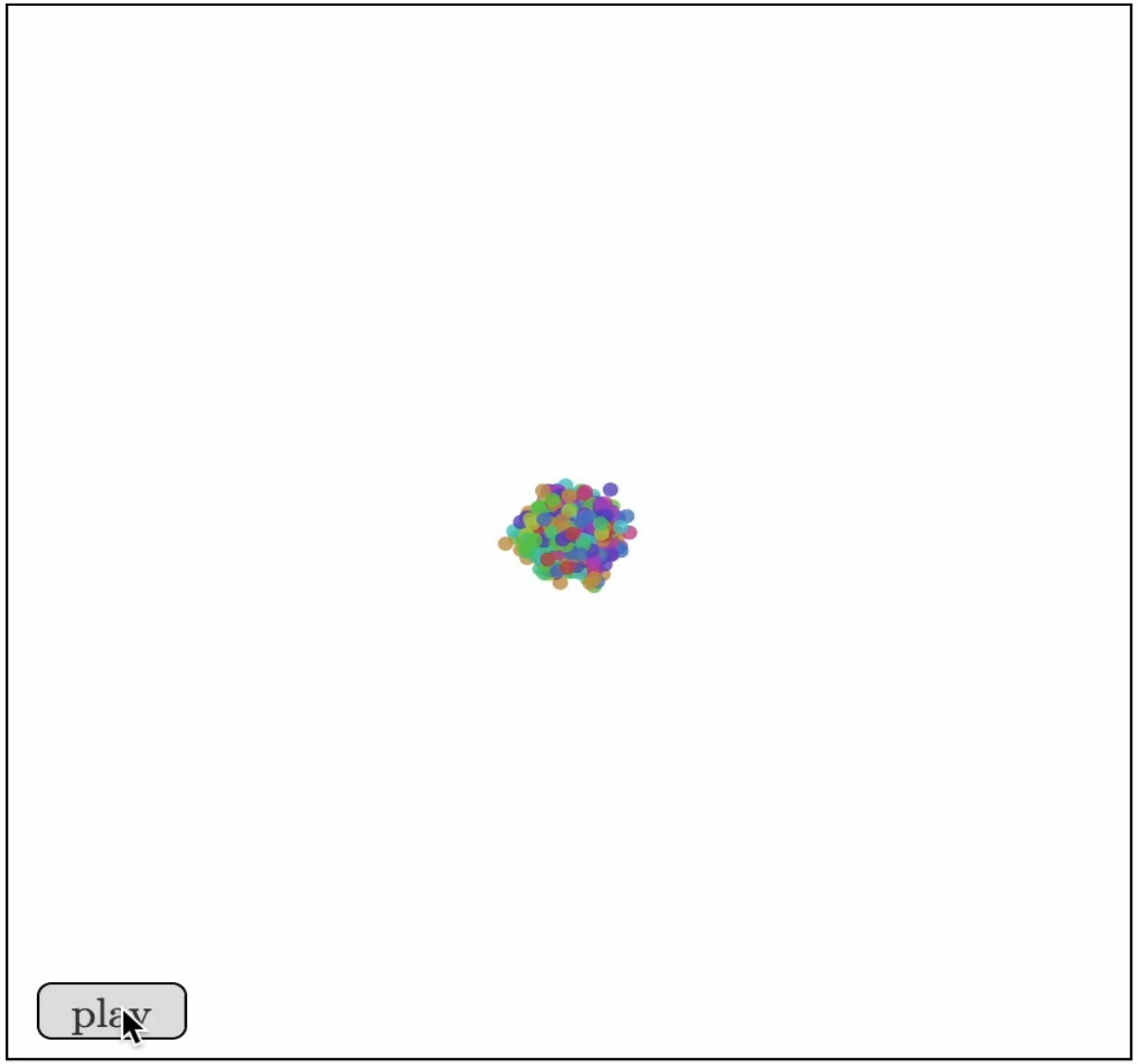
Why t-distribution? Crowding Problem

- Goal is to preserve local structure in low dimensional embedding
- Points which are far apart in high dimensional space should be far apart after projection
- The heavy tailed t-distribution helps to achieve it



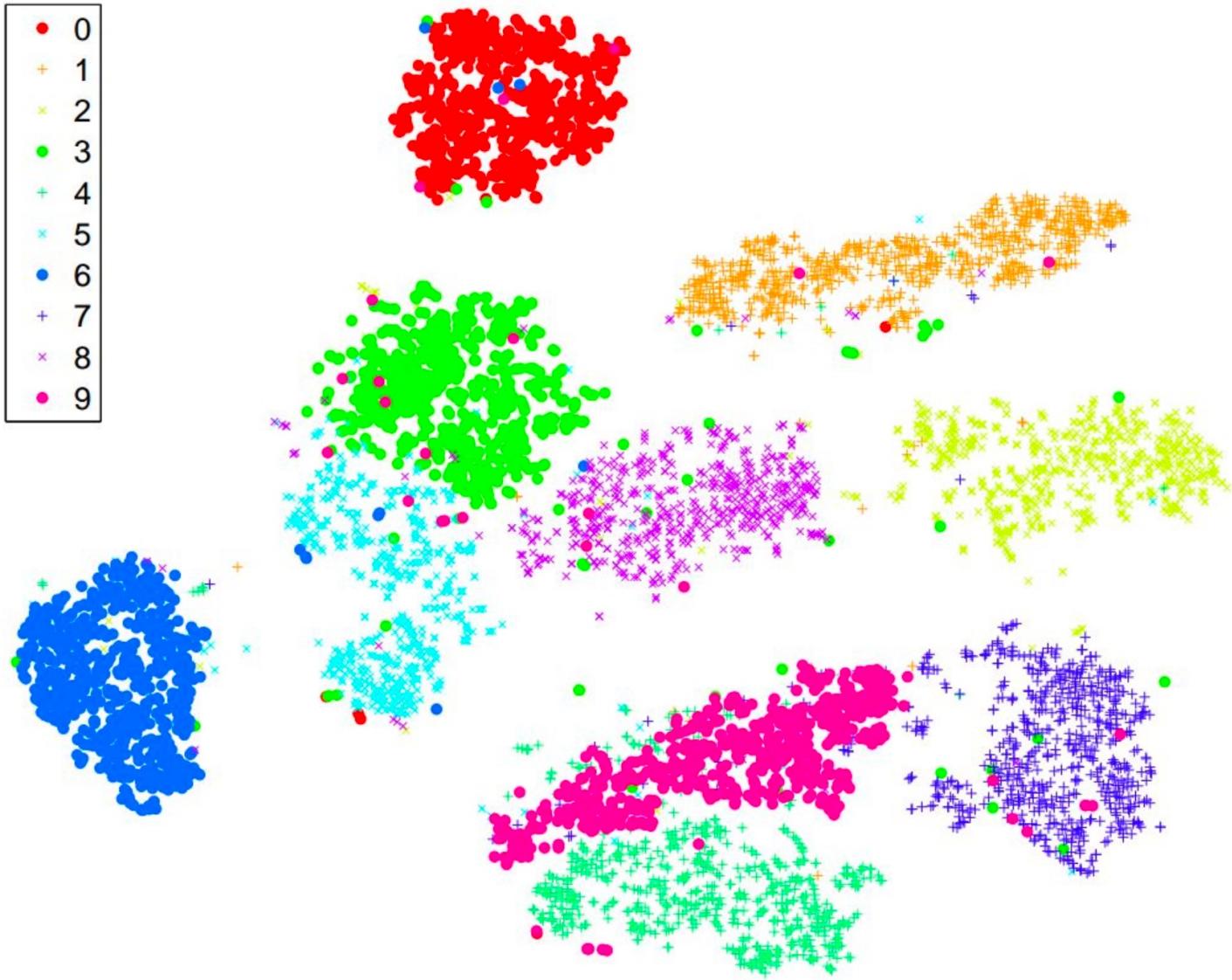


Execution of t-SNE on MNIST Data



Visualizing MNIST with t-SNE

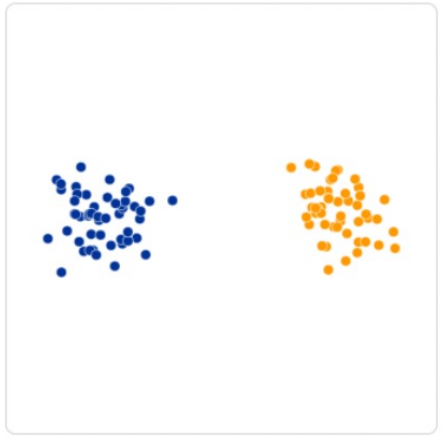
t-SNE on MNIST Data



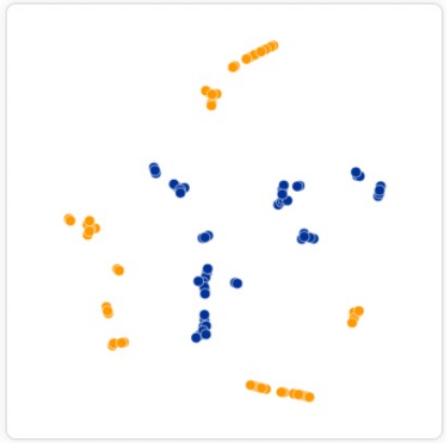
How to use t-SNE Effectively

- A key hyperparameter is perplexity
- It is a parameter that determines how to balance attention between local and global structures
  - Intuition: A guess about the number of close neighbors each point has
  - Changing this parameter has significant impact on final layout

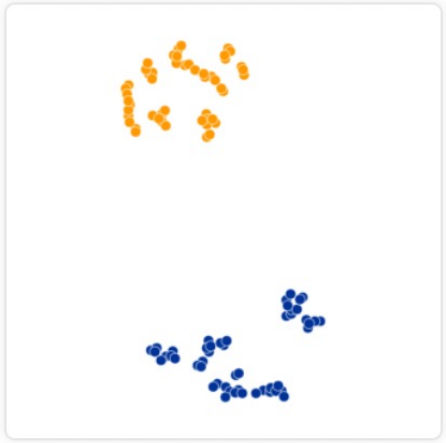
Impact of Perplexity in t-SNE



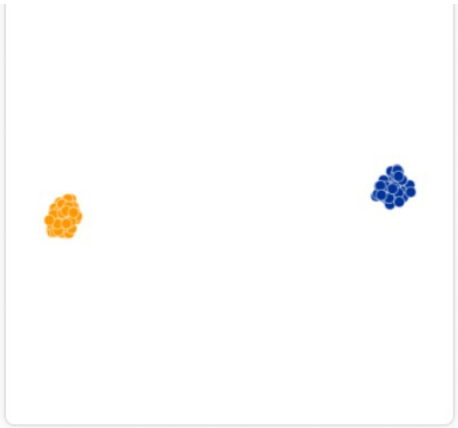
Original



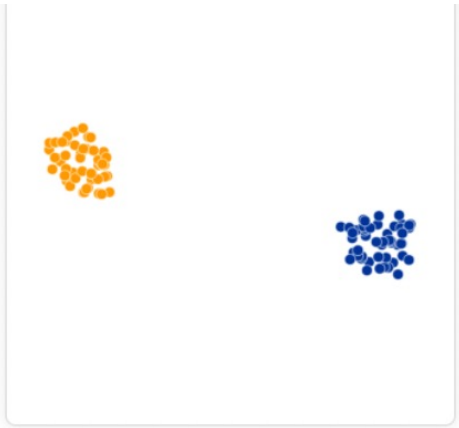
Perplexity: 2
Step: 5,000



Perplexity: 5
Step: 5,000



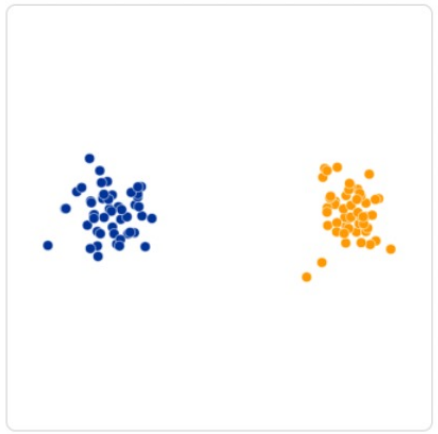
Perplexity: 30
Step: 5,000



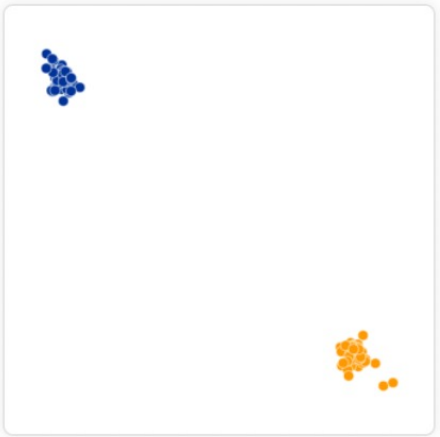
Perplexity: 50
Step: 5,000

Perplexity: 100
Step: 5,000

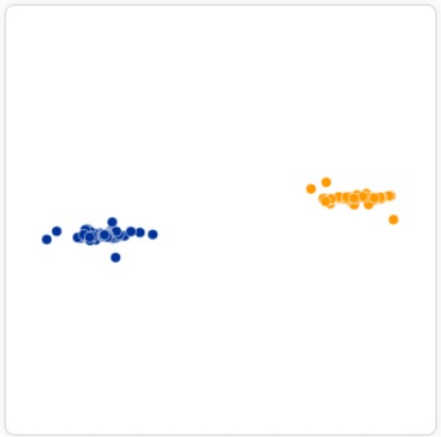
Impact of Epsilon (#iterations) in t-SNE



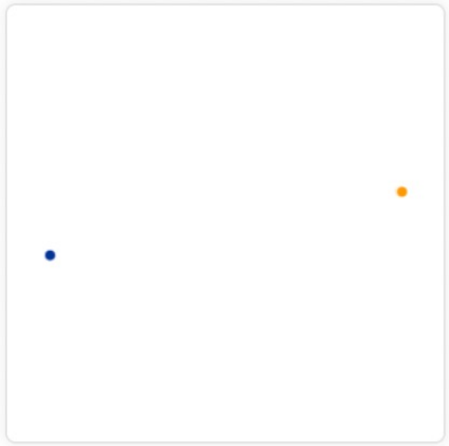
Original



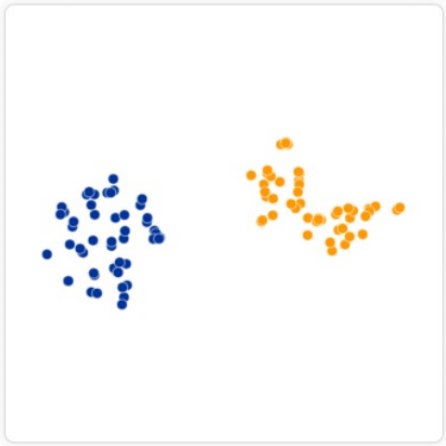
Perplexity: 30
Step: 10



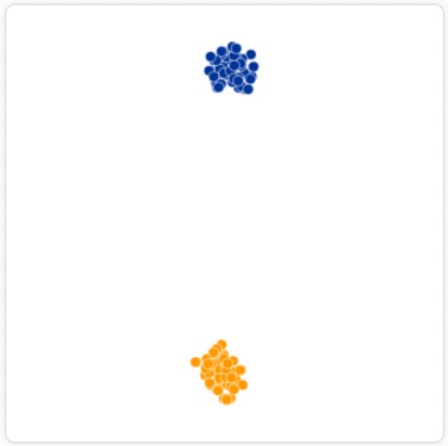
Perplexity: 30
Step: 20



Perplexity: 30
Step: 60



Perplexity: 30
Step: 120



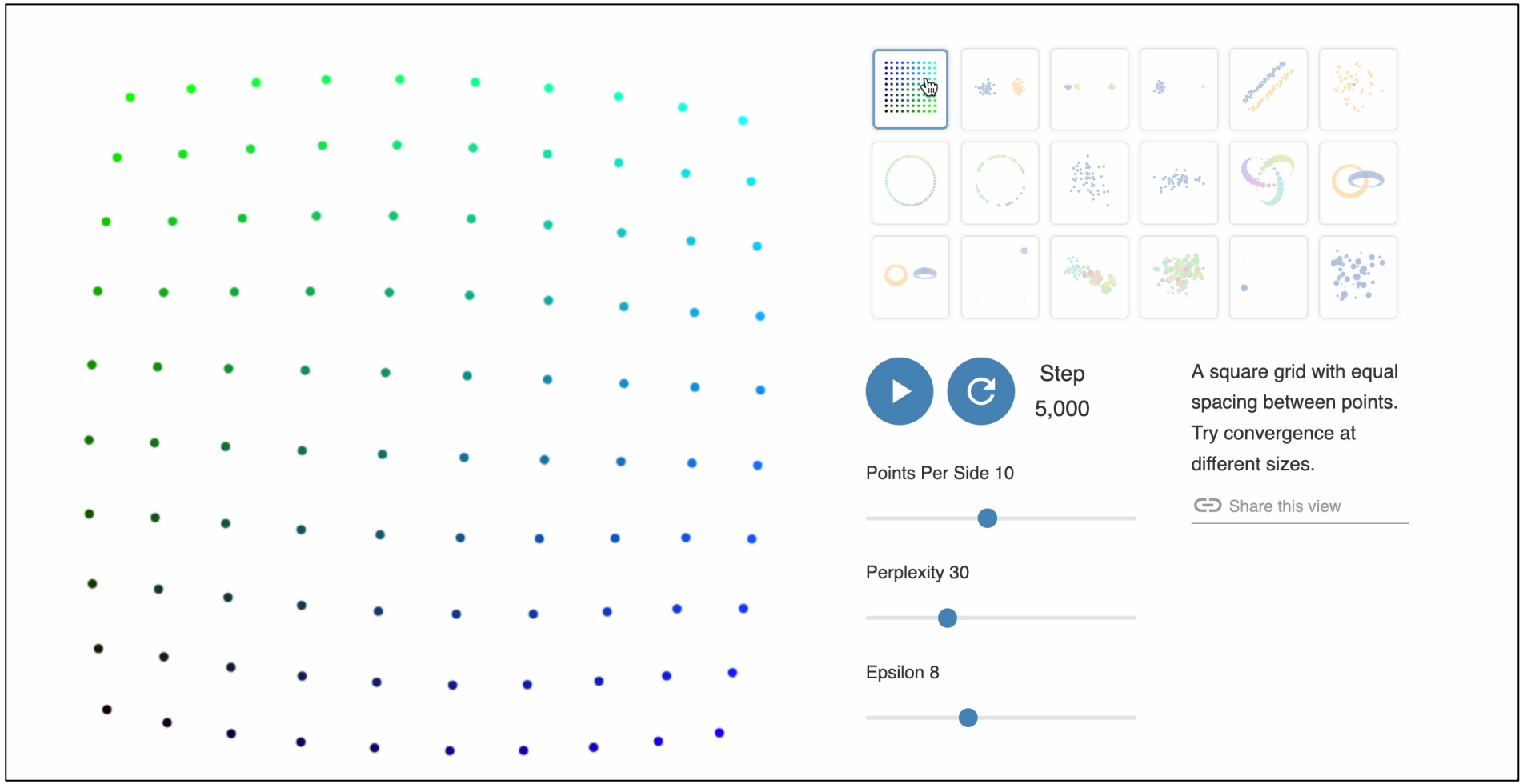
Perplexity: 30
Step: 1,000

Other Key Aspects of t-SNE

- Cluster sizes in a t-SNE plot may mean nothing
- Distances between clusters might not mean anything
- Random noise doesn’t always look random
- For topology, you may need more than one plot at different perplexity values
- Try with different number of iterations to ensure the algorithm has converged

Excellent Online Resource for Learning t-SNE

- https://distill.pub/2016/misread-tsne/

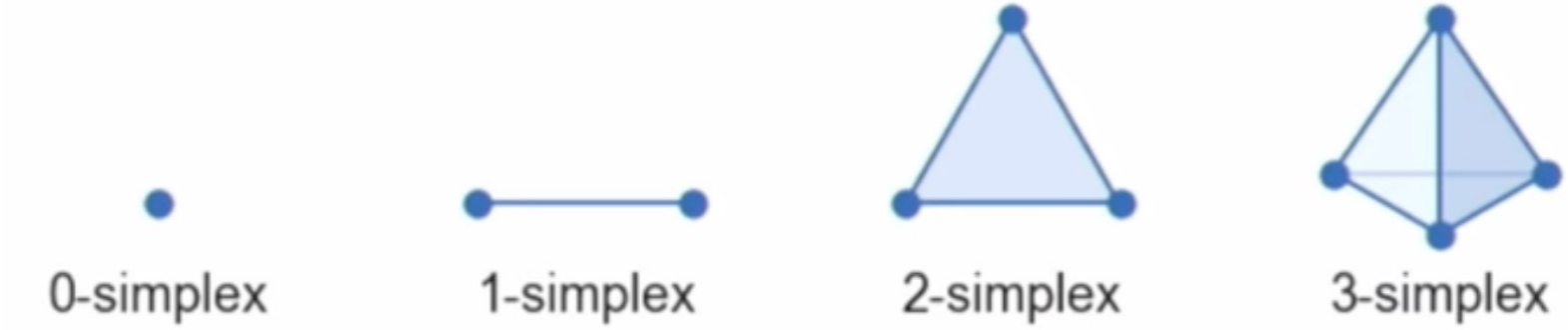


Uniform Manifold Approximation and Projection (UMAP)

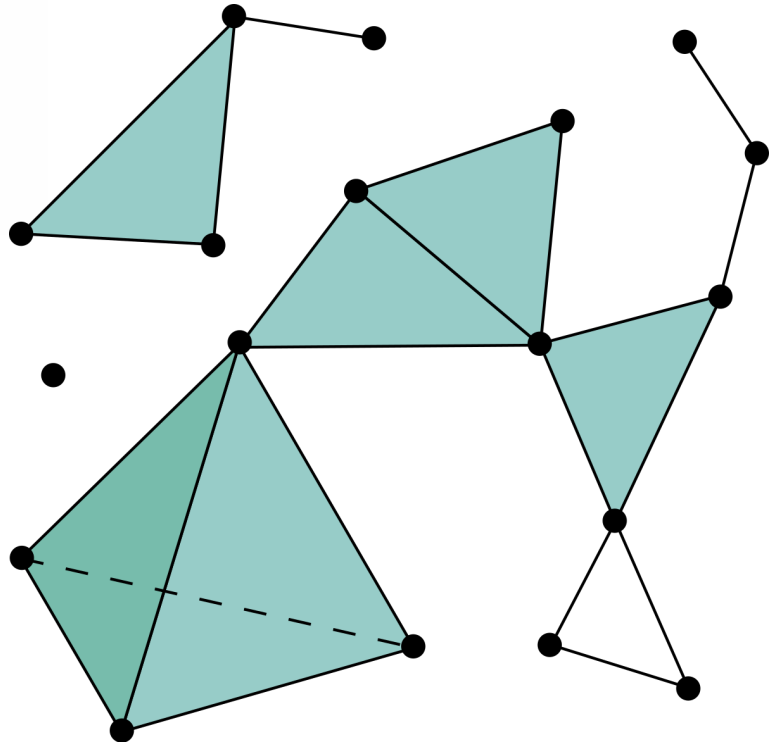
- A dimensionality reduction technique that assumes the available data samples are evenly (**uniformly**) distributed across a topological space (**manifold**), which can be **approximated** from these finite data samples and mapped (**projected**) to a lower-dimensional space.
- Key steps:
  - Learning the manifold structure in the high-dimensional space
  - Finding a low-dimensional representation of the high dim. manifold
- Given the high dimensional graph structure, UMAP projects the data into lower dimensions via a **force-directed graph layout algorithm**
- **Paper:** https://arxiv.org/abs/1802.03426
  - ~15,000 citations!!

Simplex and Simplicial Complex

- A simplex is a generalization of the notion of a triangle or tetrahedron to arbitrary dimensions



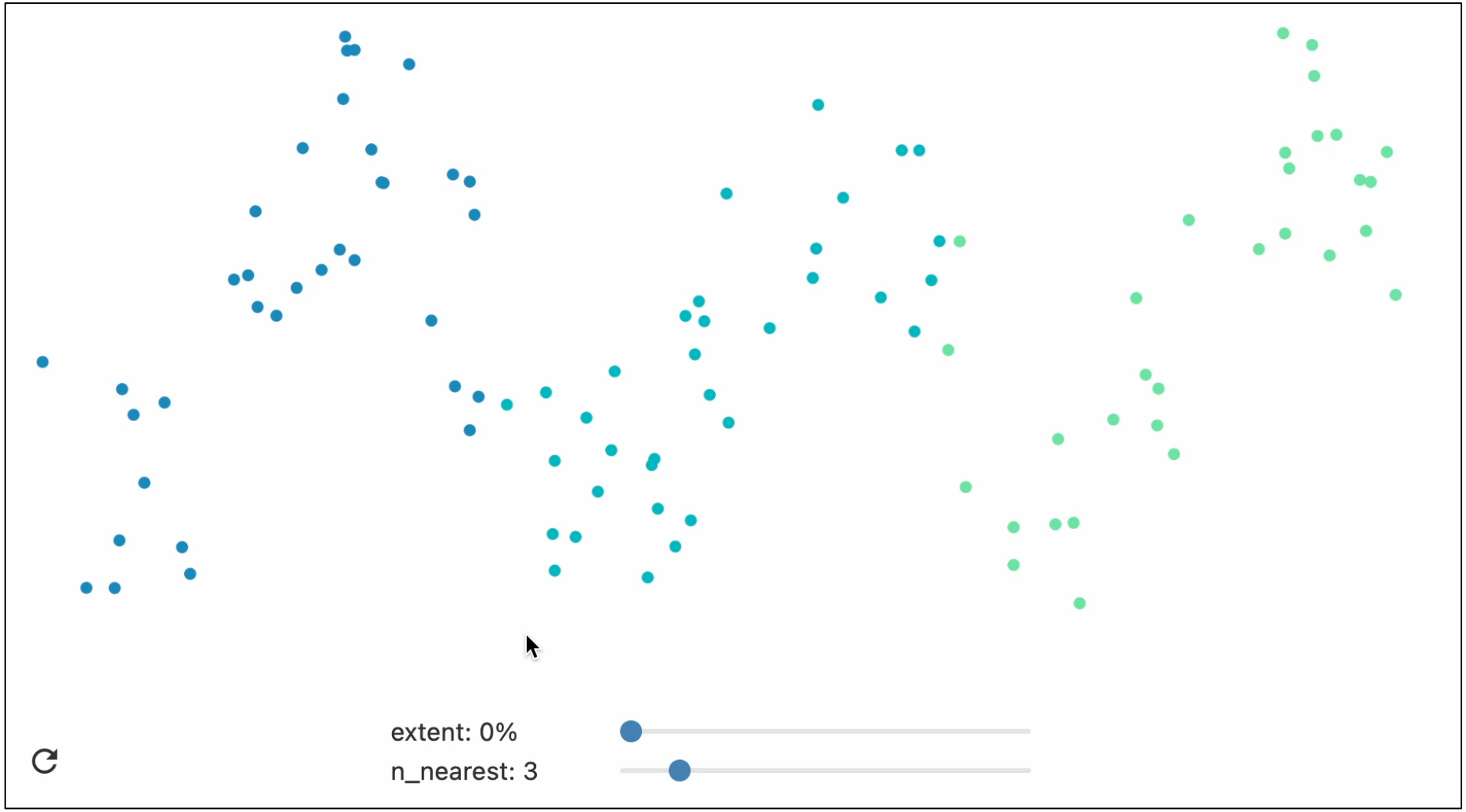
- A simplicial complex is a set composed of points, line segments, triangles, and their n-dimensional counterparts



UMAP

- Conceptually very similar to t-SNE
  - Construct a high dimensional graph representation of the data
  - Optimizes a low-dimensional graph to be as structurally similar as possible
- Idea behind constructing the high dimensional graph
  - Build a ‘fuzzy simplicial complex’
    - A weighted graph where edge weights represent likelihood that two points are connected
  - Connectedness: Grow a radius outward from each point and when it overlaps with a neighbor, connect the points
  - Then make the graph "fuzzy" by decreasing the likelihood of connection as the radius grows outward
  - Each point must be connected to at least its closest neighbor to capture local structure

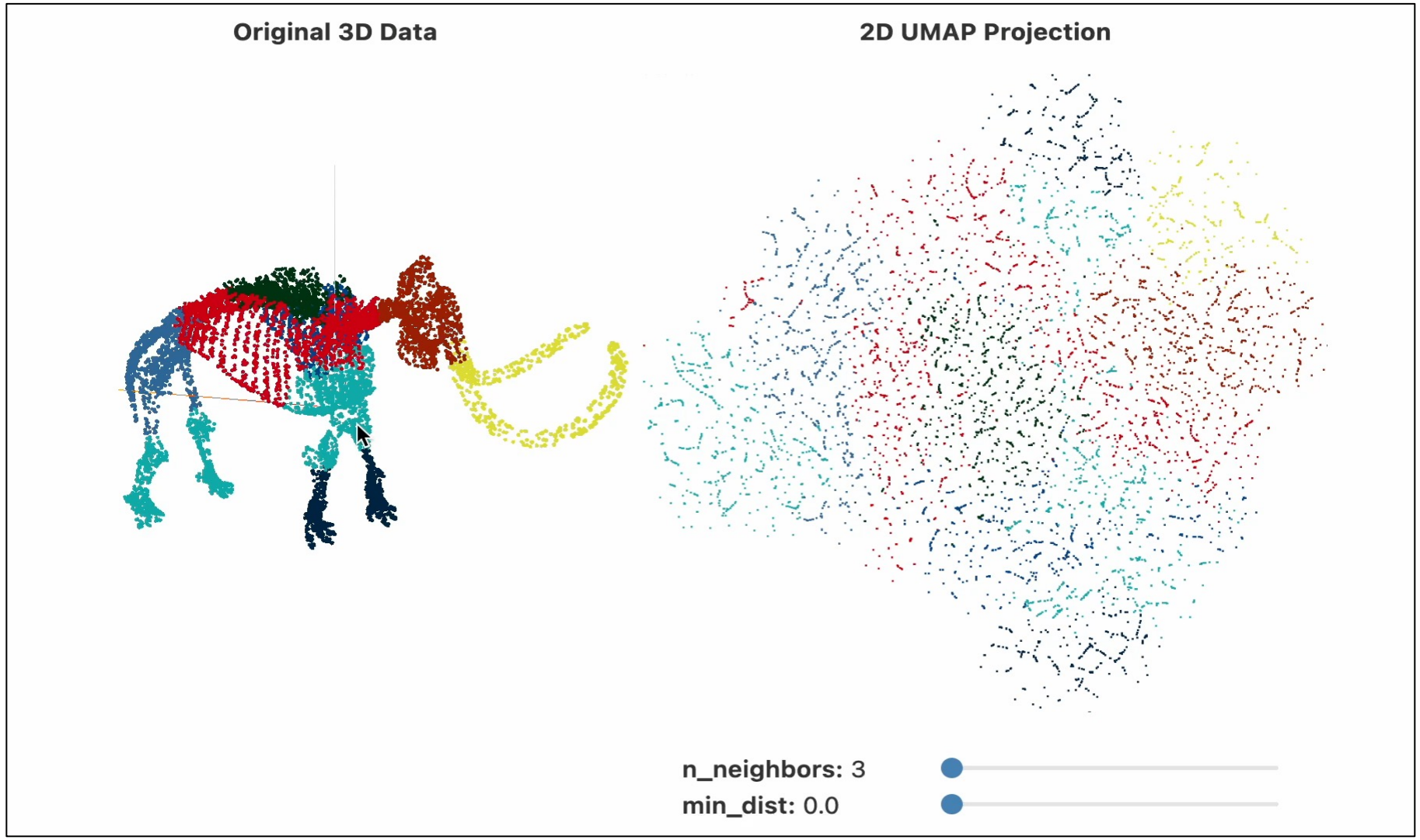
UMAP: High Dimensional Graph Construction



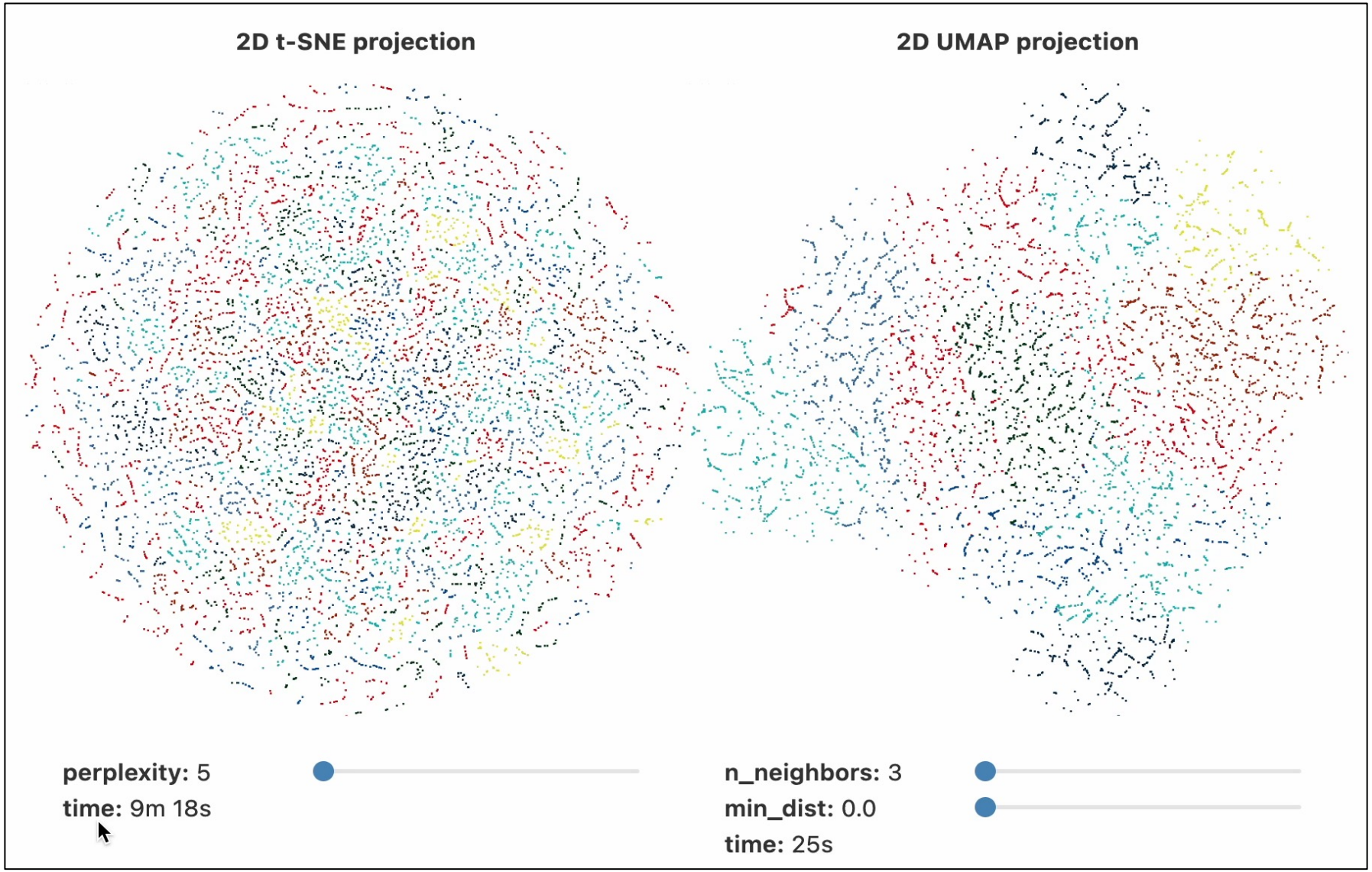
Two Important Parameters of UMAP

- Number of nearest neighbors
  - the number of approximate nearest neighbors used to construct the initial high-dimensional graph
- Minimum distance
  - the minimum distance between points in low-dimensional space controls how tightly UMAP clumps points together
    - Low values leading to more tightly packed embeddings
    - Larger values will make UMAP pack points together more loosely, focusing instead on the preservation of the broad topological structure

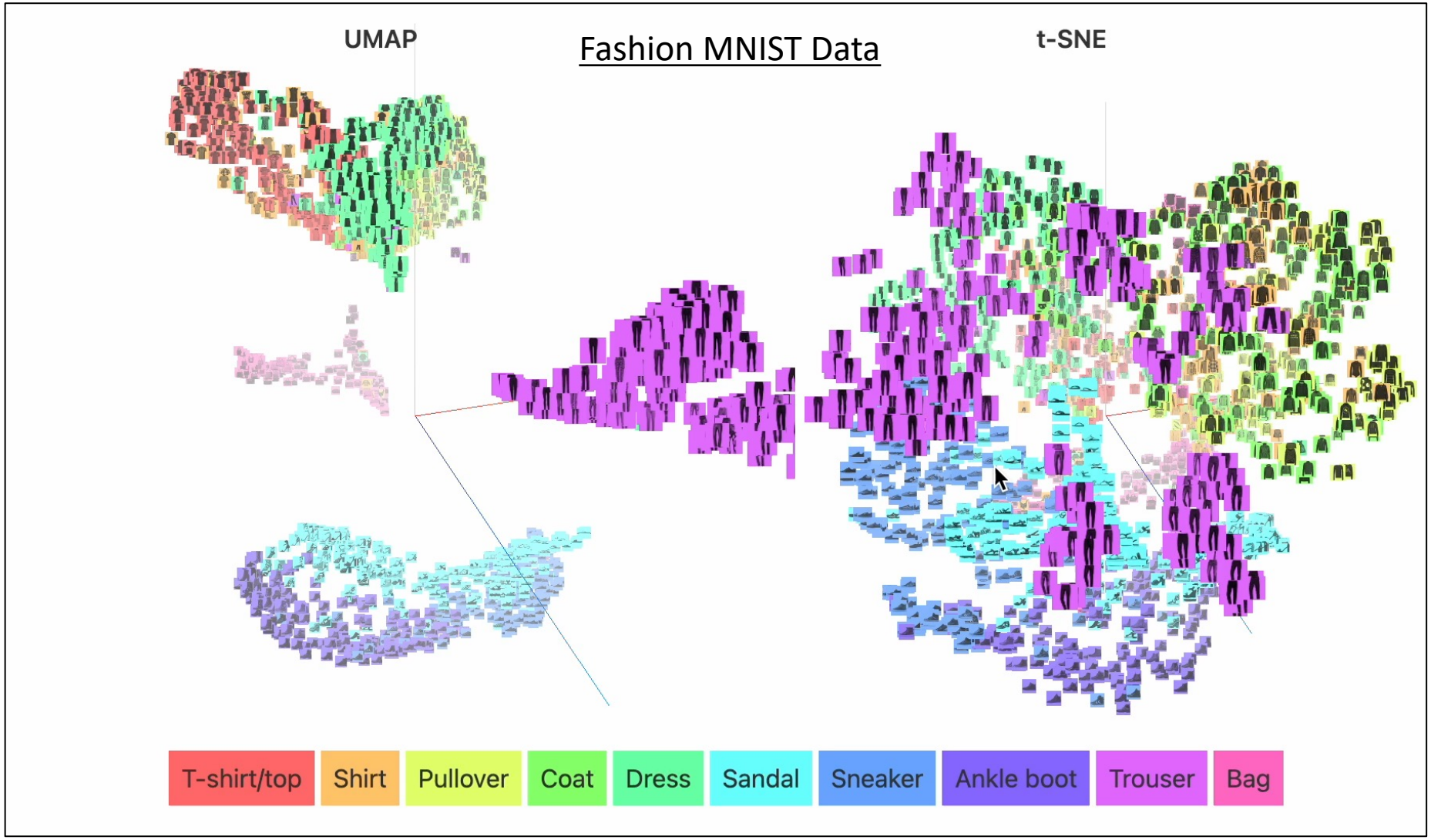
Two Important Parameters of UMAP



UMAP vs t-SNE


2D t-SNE projection
2D UMAP projection
perplexity: 5
time: 9m 18s
n_neighbors: 3
min_dist: 0.0
time: 25s

UMAP vs t-SNE


UMAP
Fashion MNIST Data
t-SNE
T-shirt/top
Shirt
Pullover
Coat
Dress
Sandal
Sneaker
Ankle boot
Trouser
Bag

Comments to both t-SNE and UMAP Methods

- Hyperparameters really matter!
- Cluster sizes may mean nothing
- Distances between clusters might not mean anything
- You may need more than one plot
- Random noise doesn’t always look random

A Comparative Study & Performance

MNIST dataset (downsampled to 2000 points)

PCA: 0.82 sec

LLE: 260 sec

Isomap: 280 sec

t-SNE: 250 sec

UMAP: 44 sec