CSE 564
Visualization & Visual Analytics

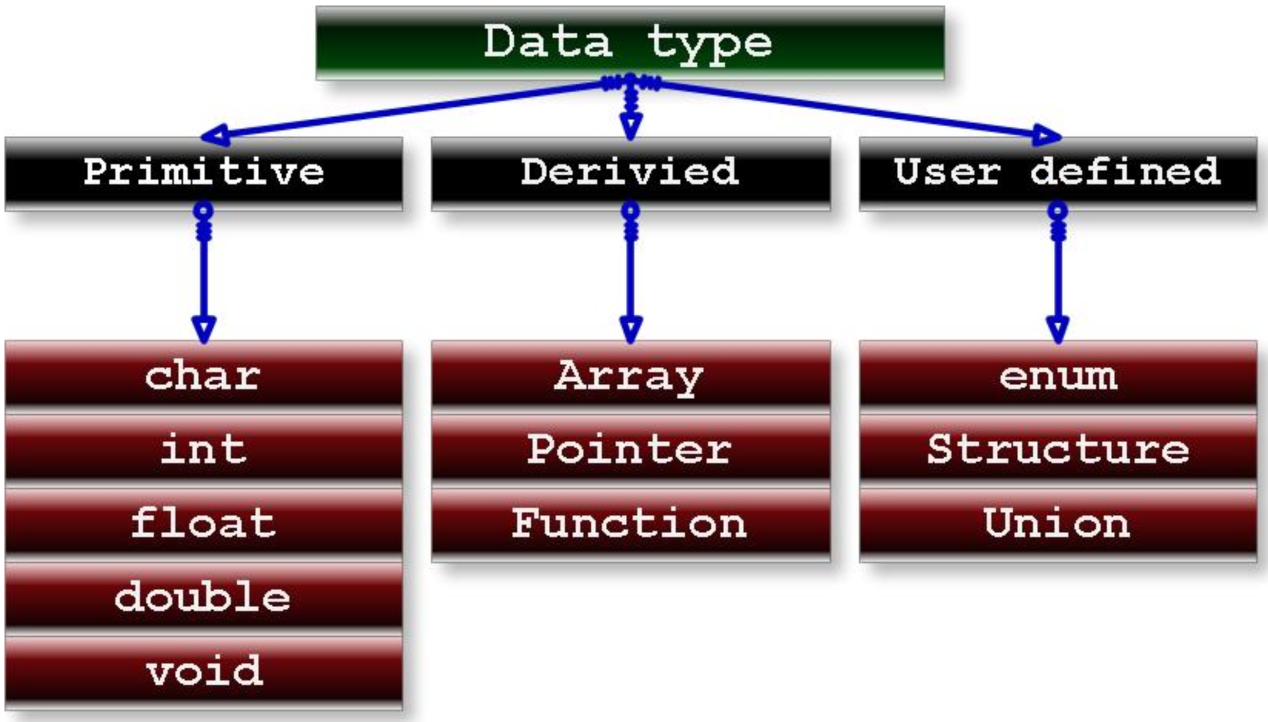
Applications and Basic Tasks

Klaus Mueller

Computer Science Department
Stony Brook University

Lecture	Topic	Projects
1	Intro, schedule, and logistics	
2	Applications of visual analytics	
3	Basic tasks, data types	Project #1 out
4	Data assimilation and preparation	
5	Introduction to D3	
6	Bias in visualization	
7	Data reduction and dimension reduction	
8	Visual perception	Project #2(a) out
9	Visual cognition	
10	Visual design and aesthetics	
11	Cluster analysis: numerical data	
12	Cluster analysis: categorical data	Project #2(b) out
13	High-dimensional data visualization	
14	Dimensionality reduction and embedding methods	
15	Principles of interaction	
16	Midterm #1	
17	Visual analytics	Final project proposal call out
18	The visual sense making process	
19	Maps	
20	Visualization of hierarchies	Final project proposal due
21	Visualization of time-varying and time-series data	
22	Foundations of scientific and medical visualization	
23	Volume rendering	Project 3 out
24	Scientific and medical visualization	Final Project preliminary report due
25	Visual analytics system design and evaluation	
26	Memorable visualization and embellishments	
27	Infographics design	
28	Midterm #2	

Data Types Every CS Person Knows



Data Types in Visual Analytics

Numeric

Categorical

Text

Time series

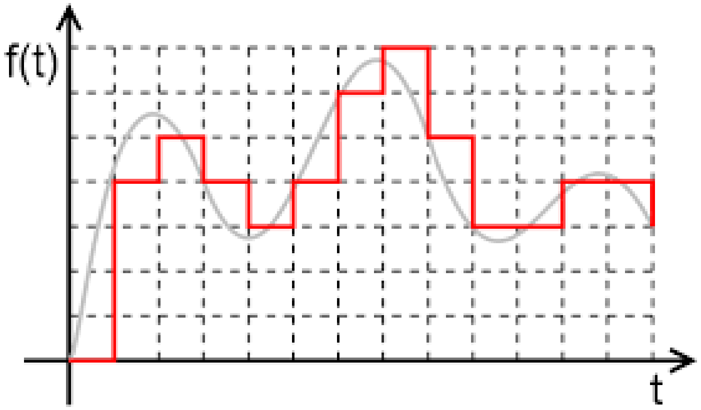
Graphs and networks

Hierarchies

Variables in Statistics

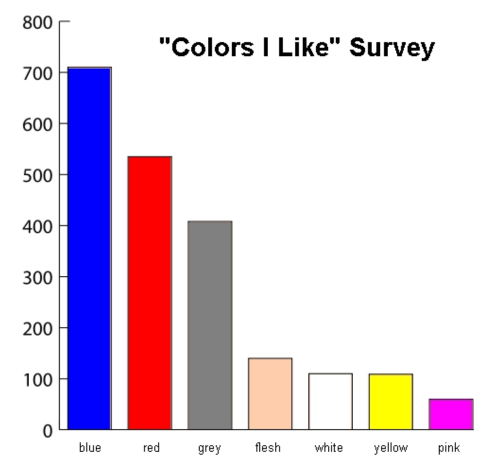
Numeric variables

- measure a **quantity** as a number
- like: ‘how many' or 'how much'
- can be continuous (grey curve)
- or discrete (red steps)



Categorical variables

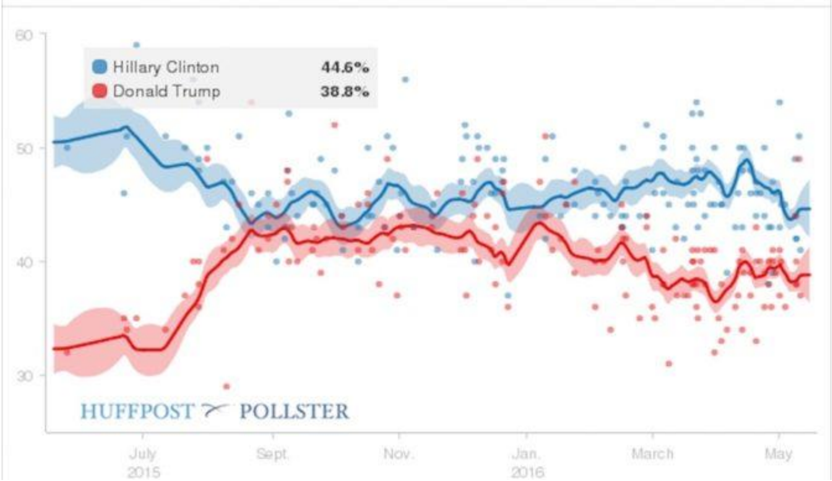
- describe a **quality** or characteristic
- like: 'what type' or 'which category’



Numeric Variables

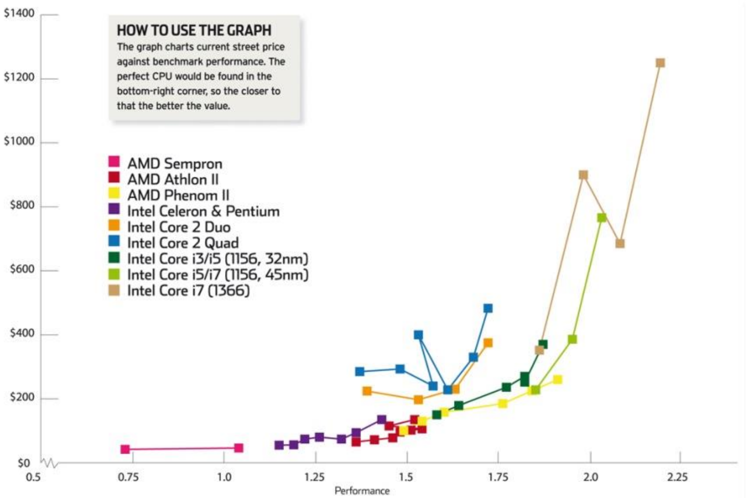
Most often the x-axis is 'time'

- provides an intuitive & innate ordering of the data values
- the majority of people expect the x-axis to be 'time'



But 'time' is not the only option

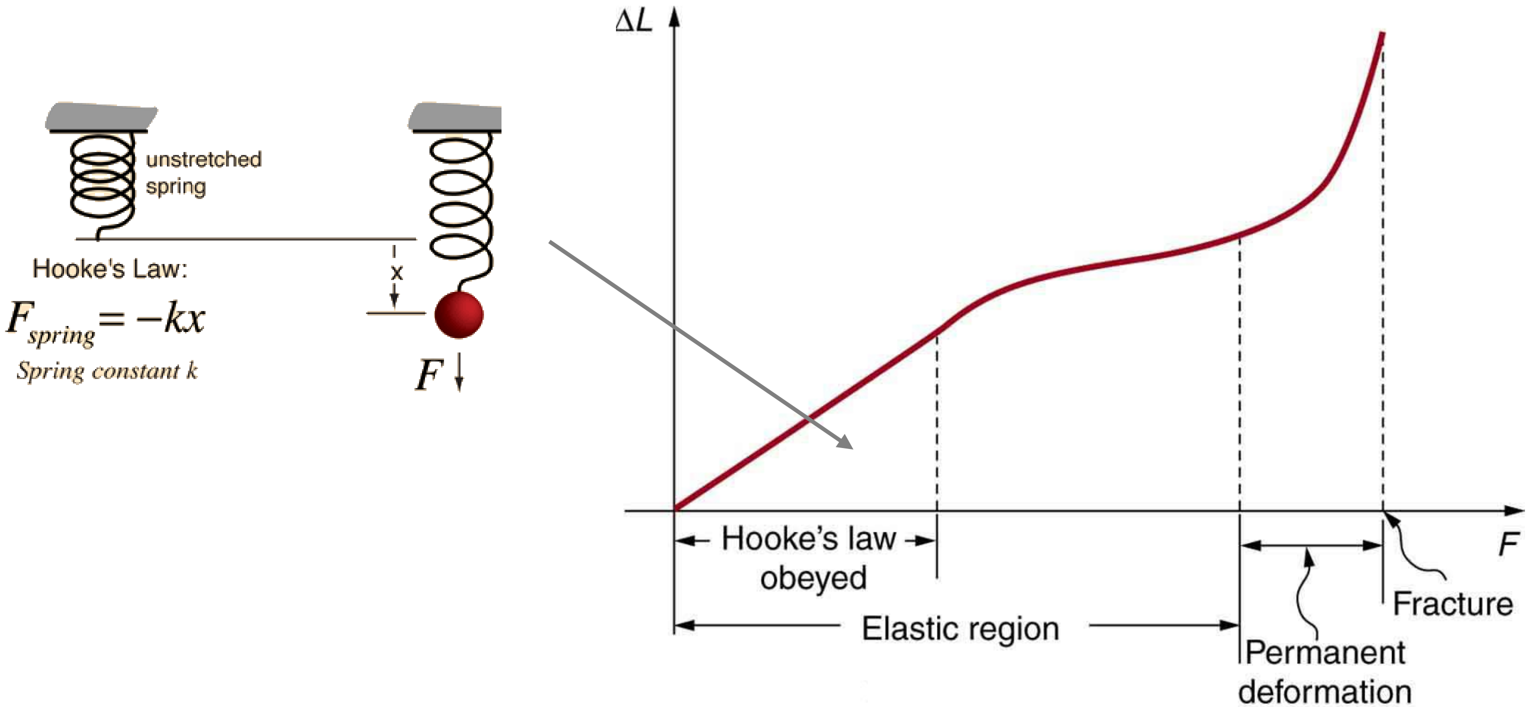
- engineers, statisticians, etc. will be receptive to this idea
- can you think of an example?



Numeric Variables

Another plot where 'time' is not the x-axis

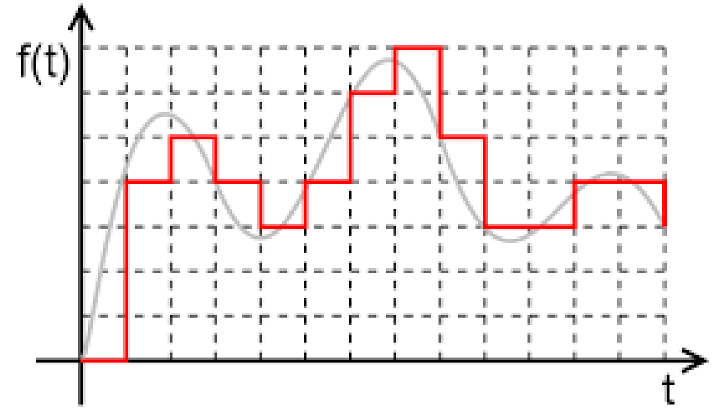
- from the engineering / physics domain
- in some sense, it tells a story



VARIABLES IN STATISTICS

Numeric variables

- measure a **quantity** as a number
- like: ‘how many' or 'how much'
- can be continuous (grey curve)
- or discrete (red steps)

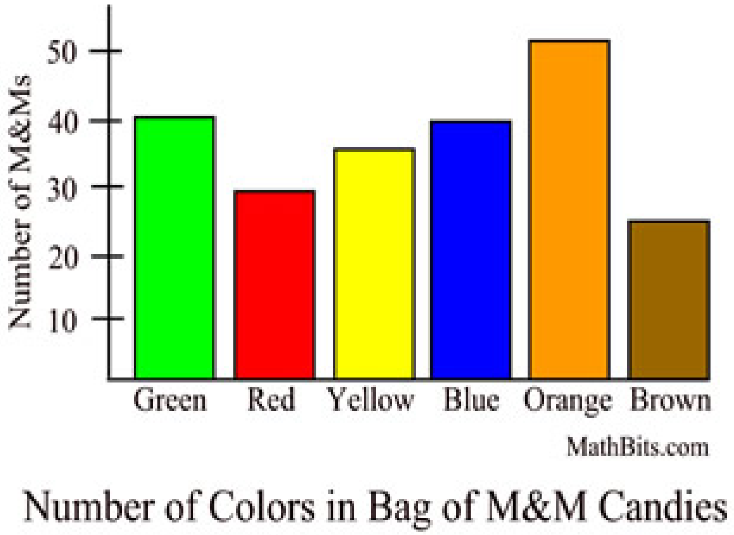


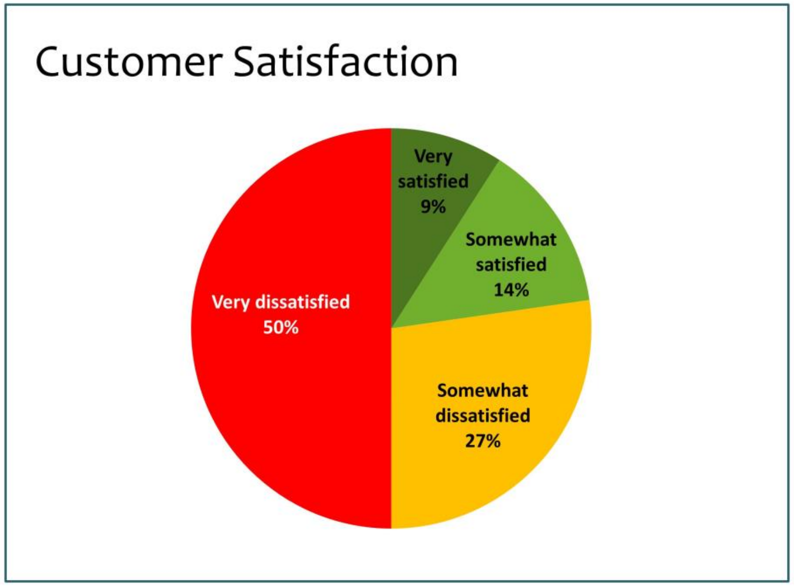
Categorical variables

- describe a **quality** or characteristic
- like: 'what type' or 'which category’
- can be ordinal = ordered, ranked (distances need not be equal)
  – clothing size, academic grades, levels of agreement
- or nominal = not organized into a logical sequence
  – gender, business type, eye color, brand

Categorical Variables

Usually plotted as bar charts or pie charts





??

??

nominal

ordinal

but of course you can plot either of them in either of these two representations

Numbers are Good

But not everything is expressed in numbers

- images
- video
- text
- web logs
- …

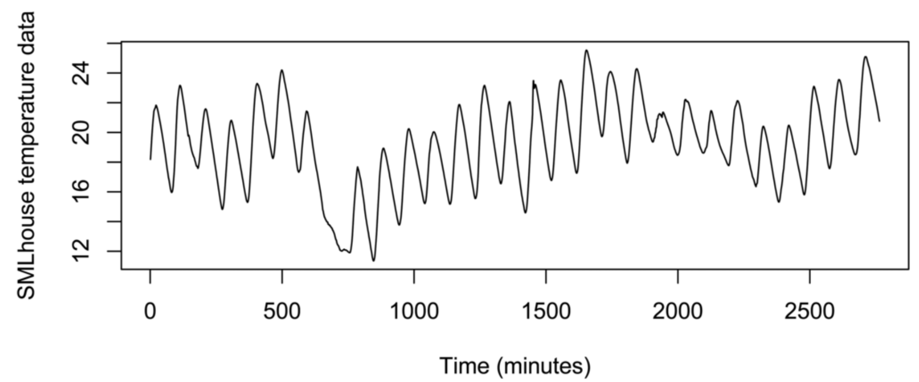
Do feature analysis to turn these abstract things into numbers

- then apply your analysis as usual
- but keep the reference to the original data so you can return to the native domain where the analysis problem originated

Sensor Data

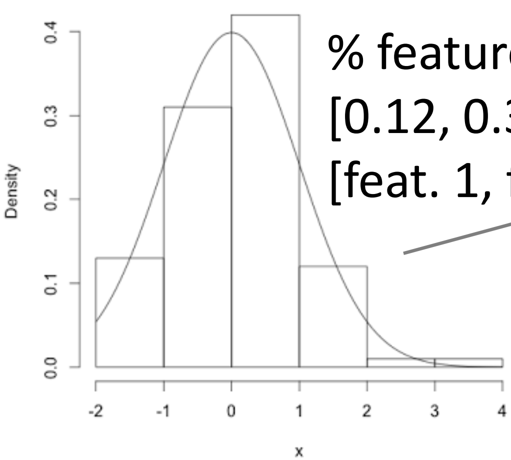
Characteristics

- often large scale
- time series

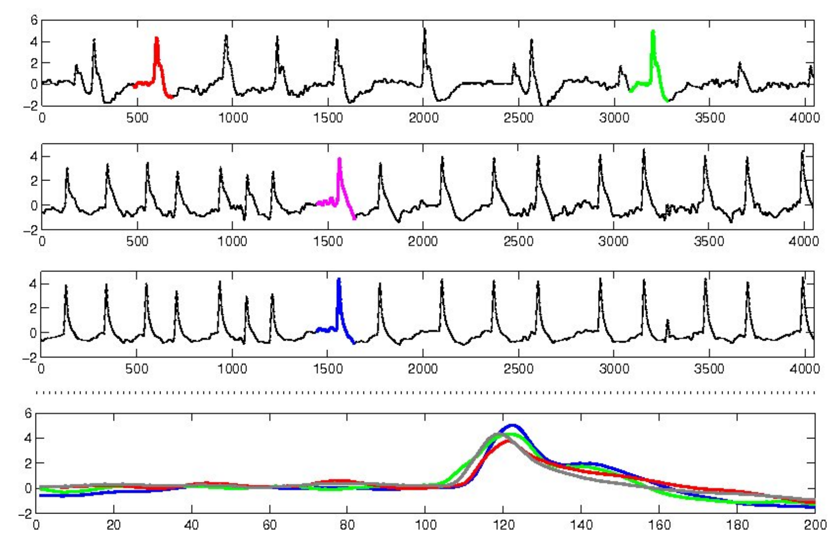


Feature Analysis

- example: Motif discovery
- encode into 5D data vector



% features discovered in stream
[0.12, 0.3, 0.41, 0.12, 0.05]
[feat. 1, feat. 2, .., feat. 5]

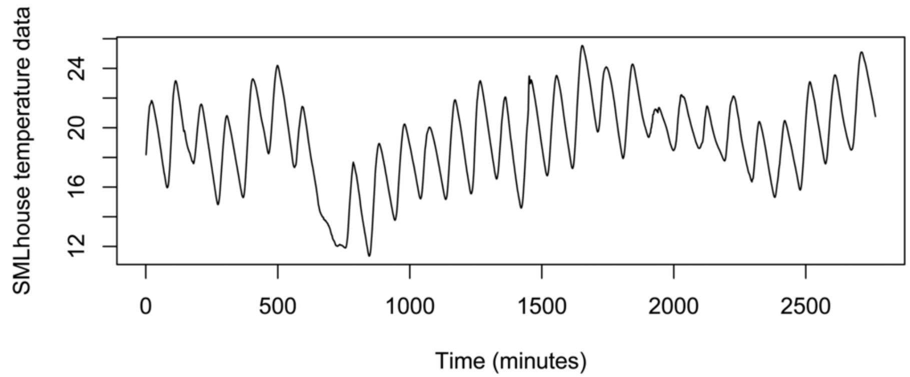


Motif discovery

Sensor Data

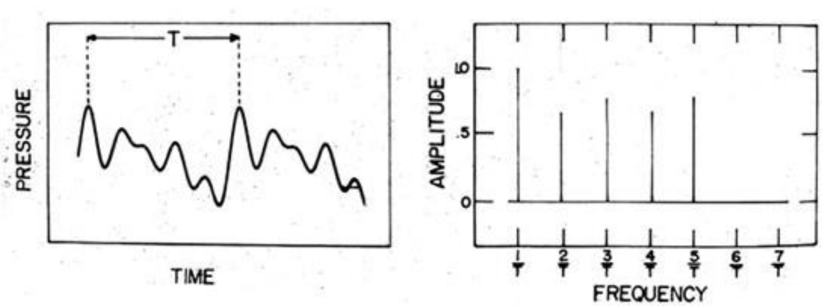
Characteristics

- often large scale
- time series



Feature Analysis

- Fourier transform (FT, FFT)
- Wavelet transform (WT, FWT)



Fourier transform

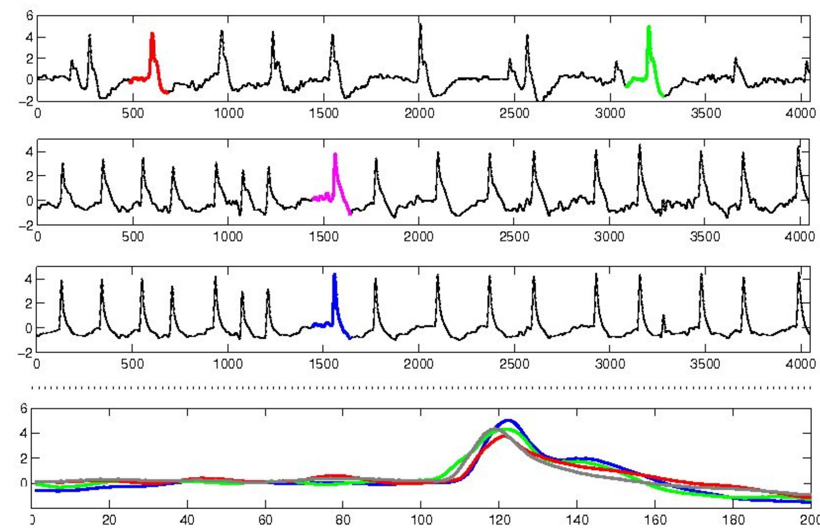


Image Data

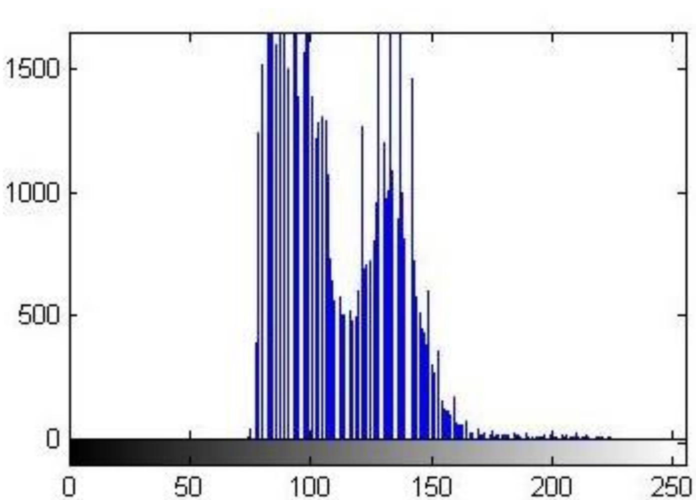
histograms

Characteristics

- array of pixels

Feature Analysis

- value histograms
- encode into a 256-D vector



[0, 0, 0, …., 10, …, 1200, …..]

Image Data

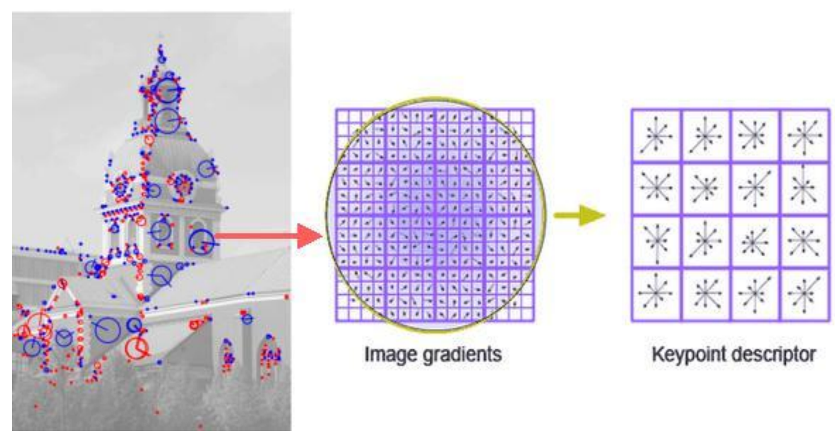
Characteristics

- array of pixels

Feature Analysis

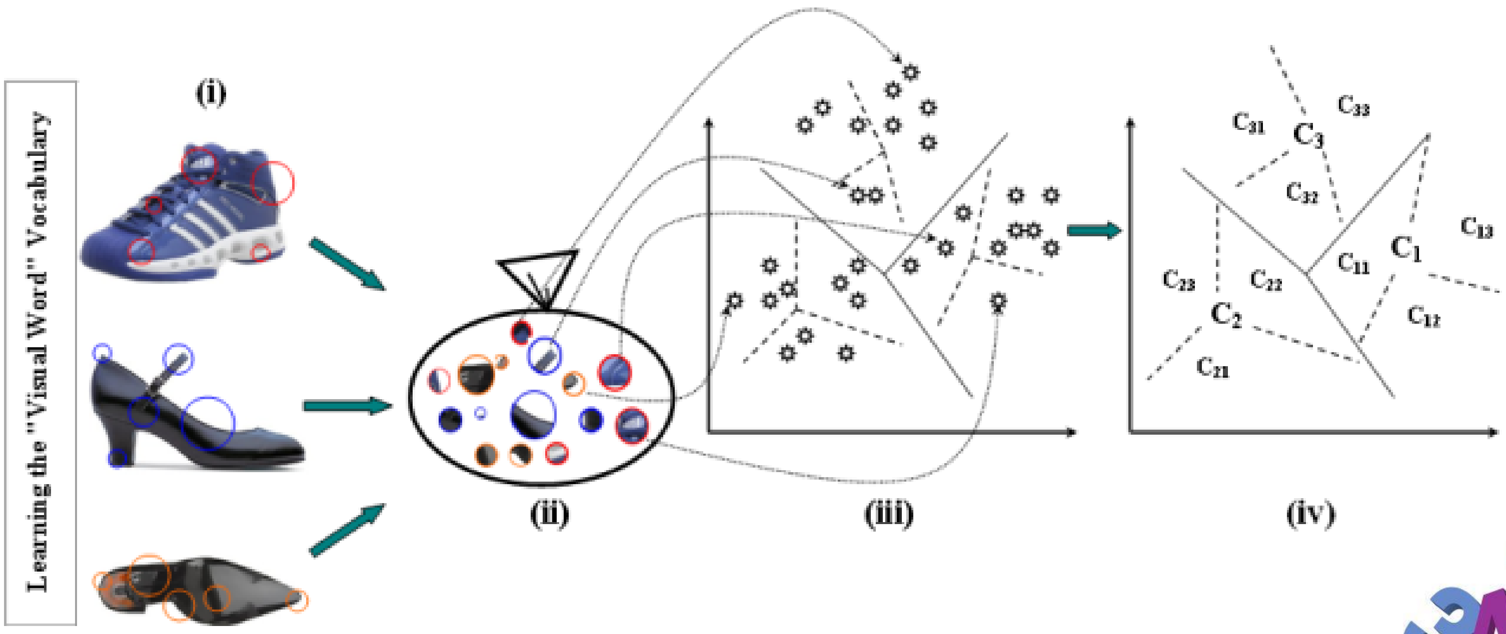
- value histograms
- gradient histograms
- FFT, FWT
- Scale Invariant Feature Transform (SIFT)
- Bag of Features (BoF)
- visual words

histograms



SIFT

Bag of Features (BoF)



Bag of Features (BoF)

1. Obtain the set of bags of features
   (i) Select a large set of images
   (ii) Extract the SIFT feature points of all the images in the set and obtain the SIFT descriptor for each feature point extracted from each image
   (iii) Cluster the set of feature descriptors for the amount of bags we defined and train the bags with clustered feature descriptors
   (iv) Obtain the visual vocabulary

2. Obtain the BoF descriptor for a given image/video frame
   (v) Extract SIFT feature points of the given image
   (vi) Obtain SIFT descriptor for each feature point
   (vii) Match the feature descriptors with the vocabulary we created in the first step
   (viii) Build the histogram

More information

Video data

Characteristics

- essentially a time series of images

Feature Analysis

- many of the above techniques apply albeit extension is non-trivial

Text Data

Characteristics

- often raw and unstructured

Feature analysis

- first step is to remove stop words and stem the data
- perform **named-entity recognition** to gain atomic elements
  - identify names, locations, actions, numeric quantities, relations
  - understand the structure of the sentence and complex events
- example:
  - Jim bought 300 shares of Acme Corp. in 2006.
  - [Jim]$_{\text{Person}}$ bought [300 shares] $_{\text{Quantity}}$ of [Acme Corp.]$_{\text{Organiz.}}$ in [2006]$_{\text{Time}}$
- distinguish between
  - application of grammar rules (old style, need experienced linguists)
  - statistical models (Google etc., need big data to build)

Text to Numeric Data

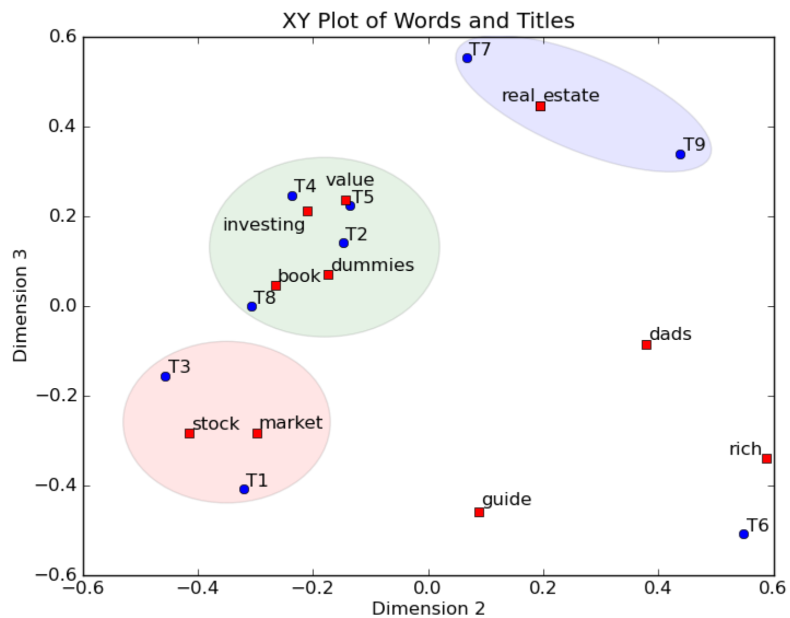
Create a term-document matrix

- turns text into a high-dimensional vector which can be compared
- use Latent Semantic Analysis (LSA) to derive a visualization

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

Term-Document Matrix

LSA

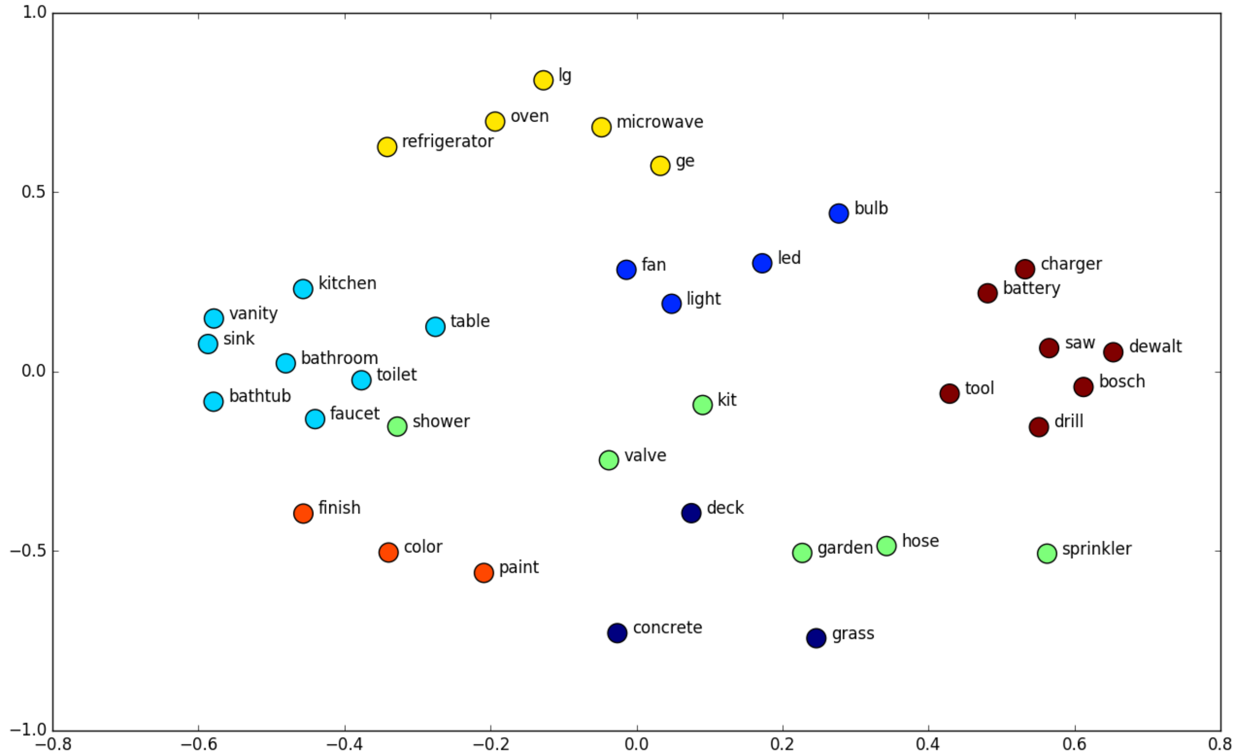


Word/document cluster

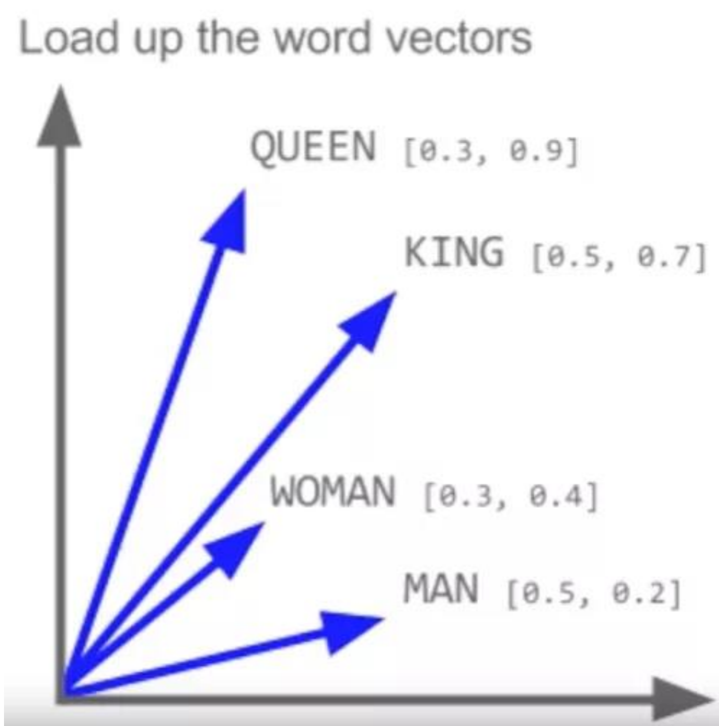
Word Embedding

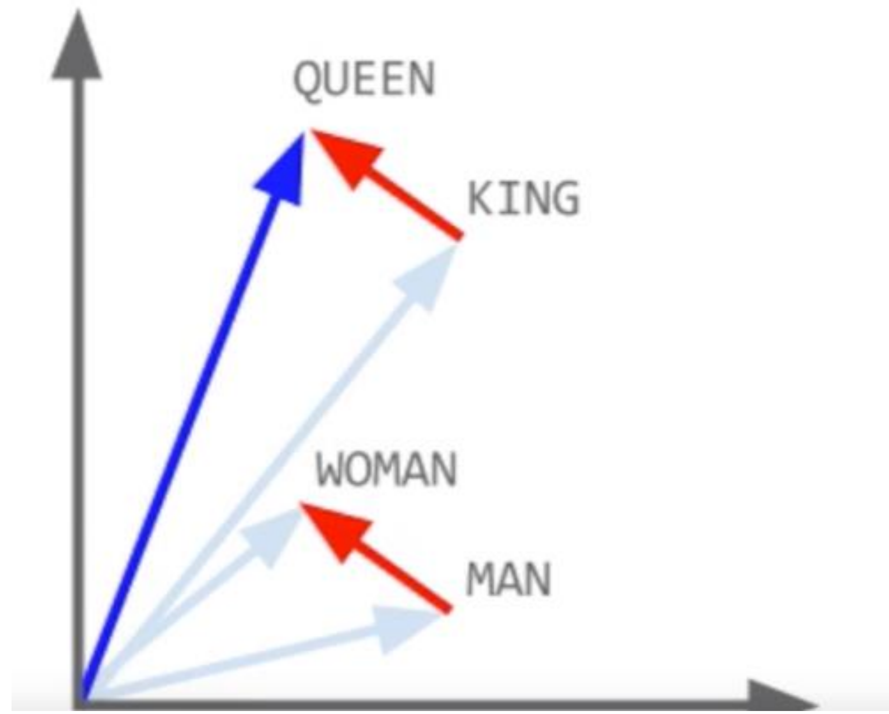
Train a shallow neural network (NN) on a corpus of text

- the NN weight vectors encode word similarity as a high-D vector
- use a 2D embedding technique to display



Word embedding Algebra





gender = WOMAN – MAN
QUEEN = KING + gender

QUEEN = KING – MAN + WOMAN

Word Cloud

Maps the frequency of words in a corpus to size

https://www.jasondavies.com/wordcloud/

OTHER DATA

Weblogs

- typically represented as text strings in a pre-specified format
- this makes it easy to convert them into multidimensional representation of categorical and numeric attributes

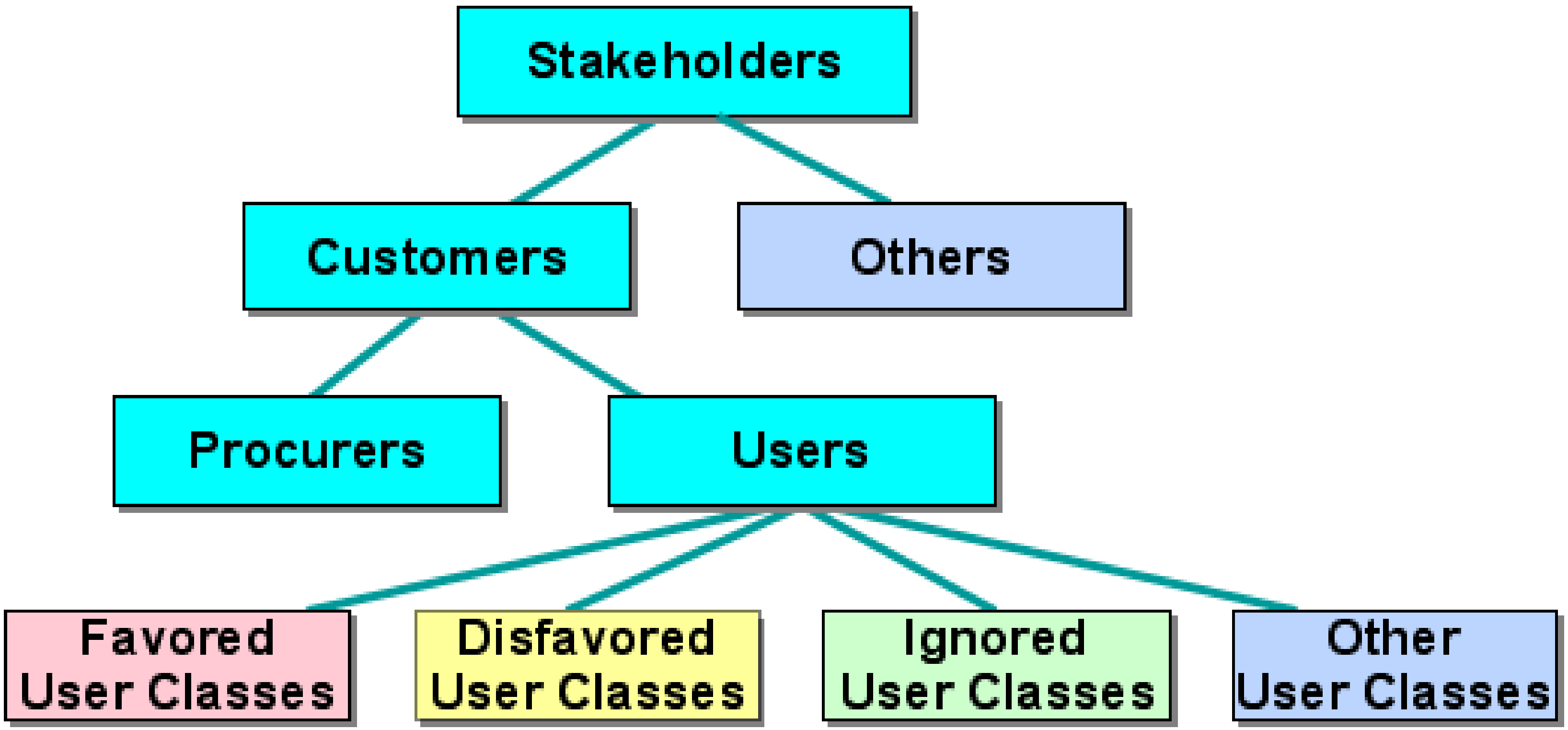
Network traffic

- characteristics of the network packets are used to analyze intrusions or other interesting activity
- a variety of features may be extracted from these packets
  - the number of bytes transferred
  - the network protocol used
  - IP ports used

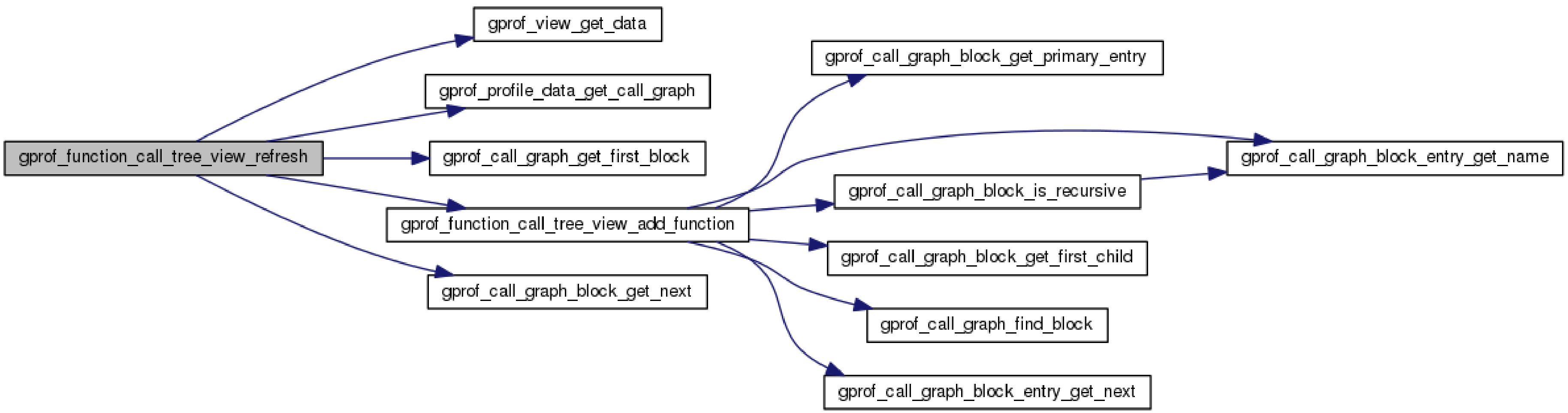
Let's Look at Some Essential Graphical Representations

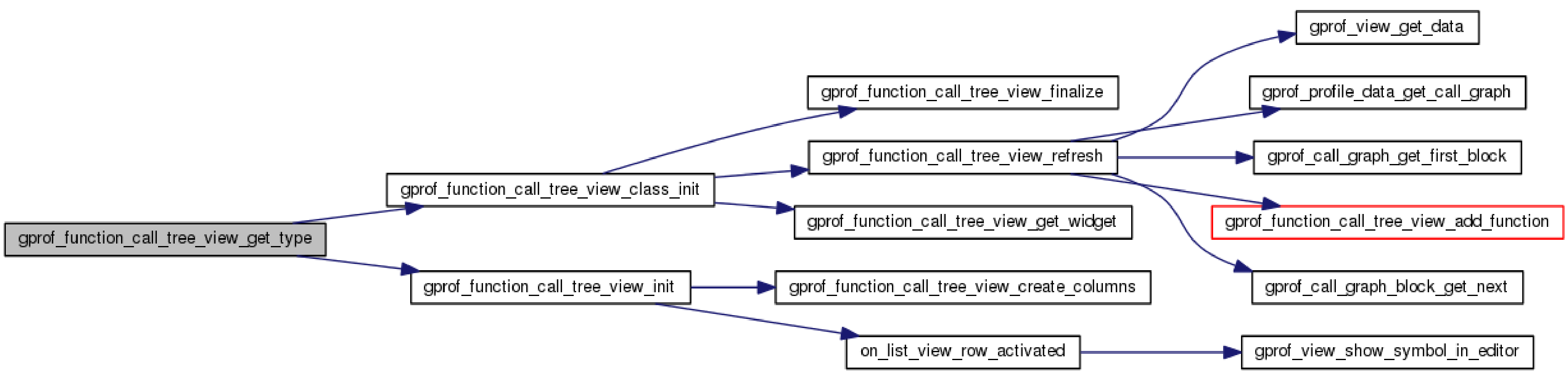
And Do Some Advertising for D3

Stakeholder Hierarchy

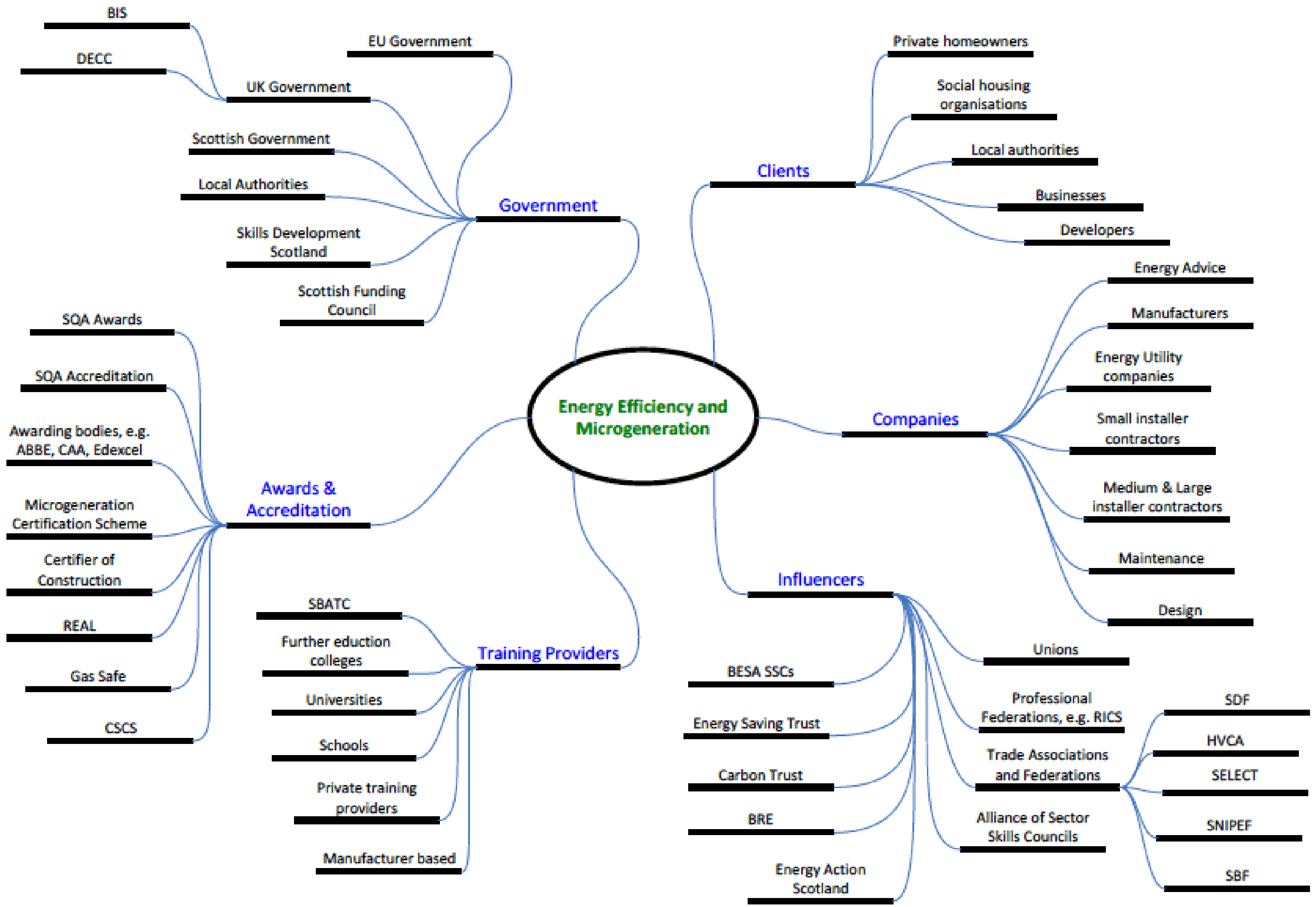


Function Call Tree





More Complex Stakeholder Hierarchy



Hierarchies

Questions you might have

- how large is each group of stakeholders (or function)?
  – tree with quantities
- what fraction is each group with respect to the entire group?
  – partition of unity
- how is information disseminated among the stakeholders (or functions)?
  – information flow
- how close (or distant) are the individual stakeholders (functions) in terms of some metric?
  – force directed layout

Invoke Nature

More scalable tree, and natural with some randomness

http://animateddata.co.uk/lab/d3-tree/

Collapsible Tree

A standard tree, but one that is scalable to large hierarchies

http://mbostock.github.io/d3/talk/20111018/tree.html

Zoomable Partition Layout

A tree that is scalable and has partial partition of unity

http://mbostock.github.io/d3/talk/20111018/partition.html

Sunburst

More space efficient since it's radial, has partial partition of unity

https://observablehq.com/@kerryrodden/sequences-sunburst

Bubble Charts

No hierarchy information, just quantities

https://observablehq.com/@d3/bubble-chart

Circle Packing

Quantities and containment, but not partition of unity

http://mbostock.github.io/d3/talk/20111116/pack-hierarchy.html

Treemap

Quantities, containment, and full partition of unity

http://mbostock.github.io/d3/talk/20111018/treemap.html

Chord Diagram

Relationships among group fractions, not necessarily a tree

https://observablehq.com/@d3/chord-diagram

Hierarchical Edge Bundling

Relationships of individual group members, also in terms of quantitative measures such as information flow

http://mbostock.github.io/d3/talk/20111116/bundle.html

Collapsible Force Layout

Relationships within organization members expressed as distance and proximity

http://mbostock.github.io/d3/talk/20111116/force-collapsible.html

Voronoi Tessellation

Shows the closest point on the plane for a given set of points… and a new point via interaction

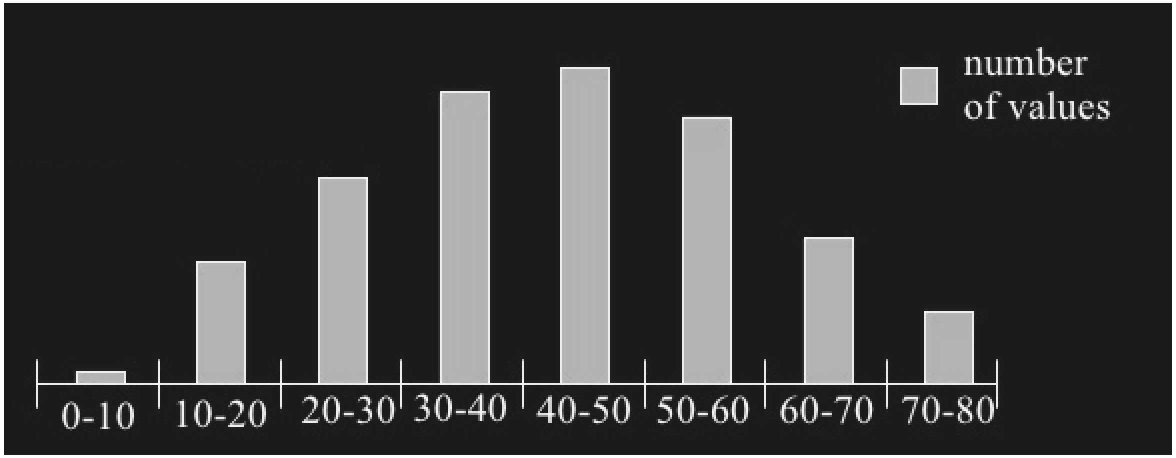
https://observablehq.com/collection/@d3/d3-delaunay

Data Type Conversions and Transformation

Numeric to Categorical Data: Discretization (1)

Solution 1:

- divide the numeric attribute values into φ **equi-width** ranges
- each range/bucket has the same width
- example: customer age

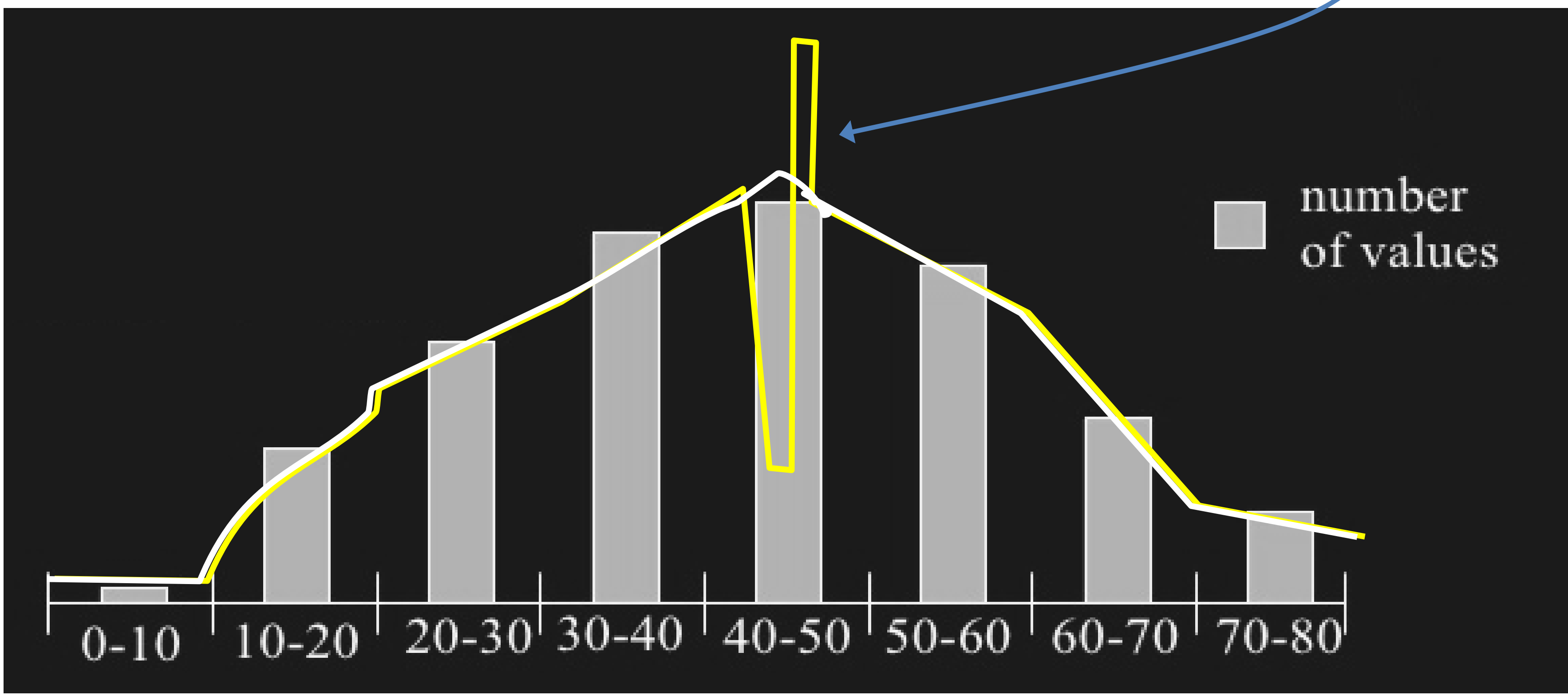


- what is lost here?

Problem With Equi-Width Histogram

Age ranges of customers could be unevenly distributed within a bin

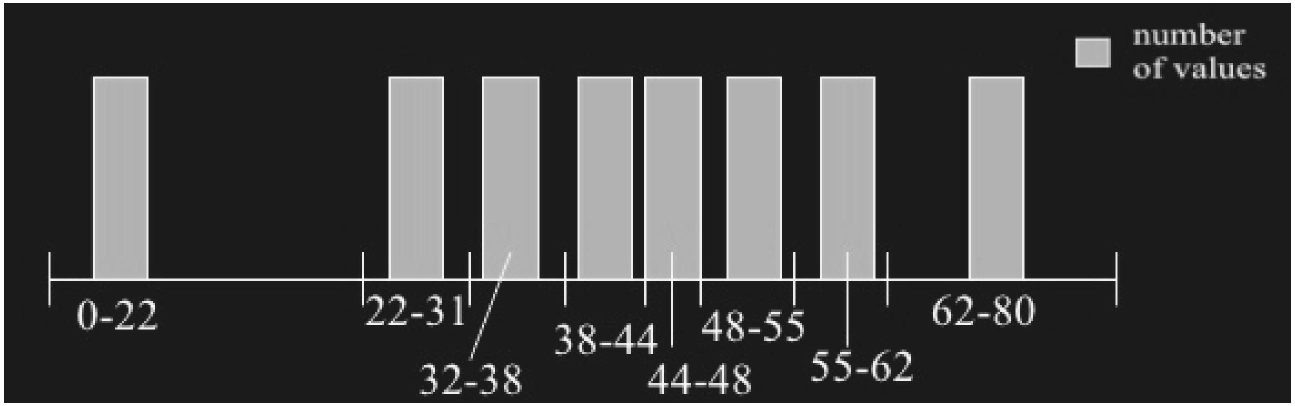
- this could be an interesting anomaly



Numeric to Categorical Data: Discretization (2)

Solution 2:

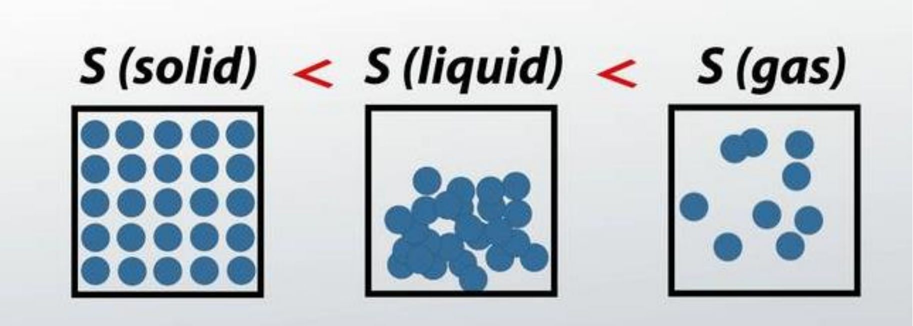
- divide the numeric attribute values into φ **equi-depth** ranges
- same number of samples in each bin
- (again) example: customer age:



- what is the disadvantage here?
- extra storage needed: must store the start/end value for each bin

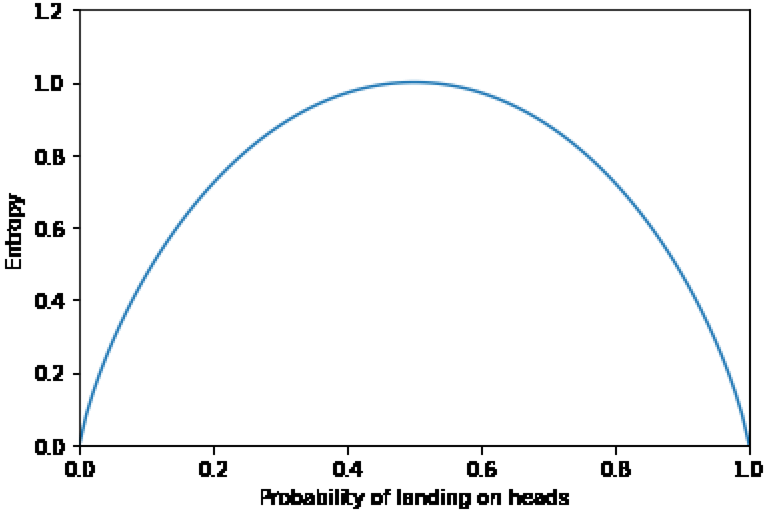
One More Binning Method

Entropy-based binning



Entropy is the amount of surprise to make a certain observation

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i)$$



The Data

from
https://www.saedsayad.com/supervised_binning.htm

O-Ring Failure	Temperature
Y	53
Y	56
Y	57
N	63
N	66
N	67
N	67
N	67
N	68
N	69
N	70
Y	70
Y	70
Y	70
N	72
N	73
N	75
Y	75
N	76
N	76
N	78
N	79
N	80
N	81

ENTROPY BASED BINNING (EBB)

Aim:

- find the best split so that the bins are as pure as possible that is the majority of the values in a bin correspond to have the same class label
- formally, it is characterized by finding the split with the maximal information gain.

Step 1: Calculate "Entropy" for the target.

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

O-Ring Failure	
Y	N
7	17

E (Failure) = E(7, 17) = E(0.29, .71) = -0.29 x $\log_2$(0.29) - 0.71 x $\log_2$(0.71) = **0.871**

Entropy Based Binning (EBB)

Step 2: Calculate "Entropy" for the target given a bin.

$$E(S,A) = \sum_{v \in A} \frac{|S_v|}{|S|} E(S_v)$$

		O-Ring Failure	
		Y	N
Temperature	<= 60	3	0
	> 60	4	17

E (Failure,Temperature) = P(<=60) x E(3,0) + P(>60) x E(4,17) = 3/24 x 0 + 21/24 x 0.7= **0.615**

Step 3: Calculate "Information Gain" given a bin.

$$\textbf{\textit{Information Gain}} = E(S) - E(S,A)$$

Information Gain (Failure, Temperature) = **0.256**

Entropy Based Binning (EBB)

[<=60, >60] turns out to be the best split

Iterate for further splits for bins with highest entropies

Gain = 0.256

		O-Ring Failure Y	O-Ring Failure N
Temperature	<= 60	3	0
	> 60	4	17

Gain = 0.101

		O-Ring Failure Y	O-Ring Failure N
Temperature	<= 70	6	8
	> 70	1	9

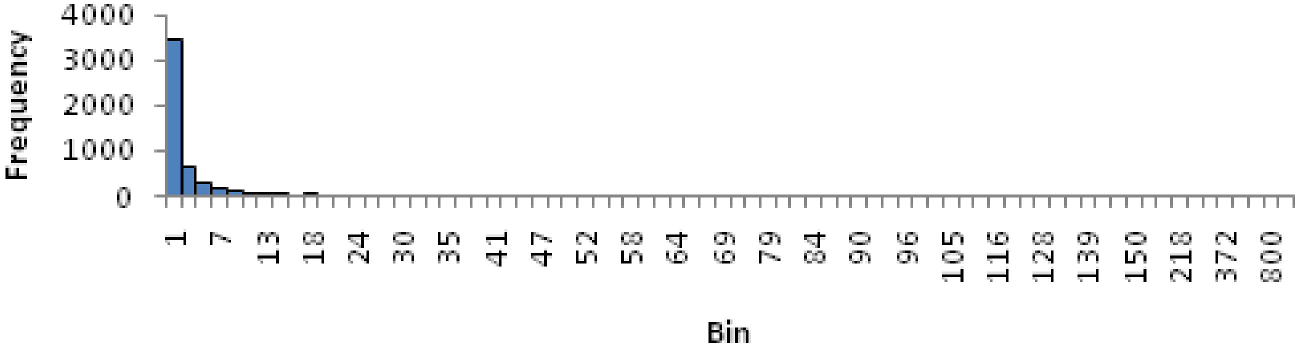
Gain = 0.148

		O-Ring Failure Y	O-Ring Failure N
Temperature	<= 75	7	11
	> 75	0	6

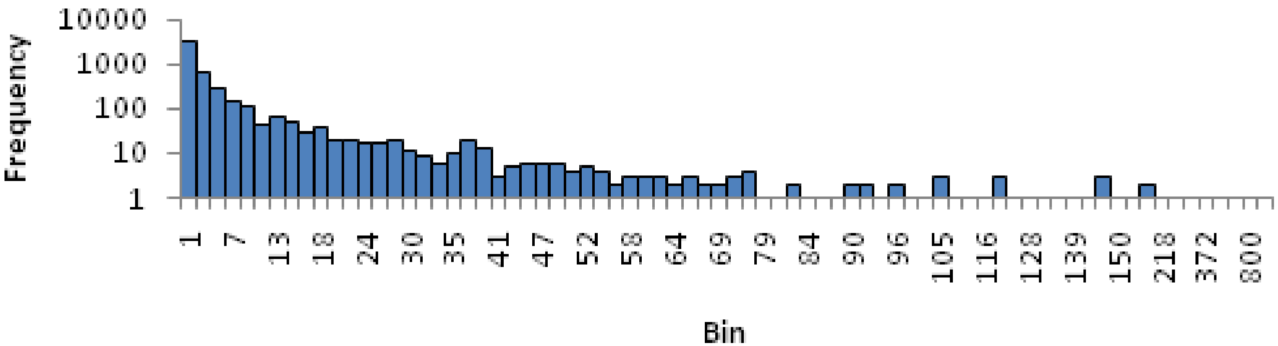
Numeric to Categorical Data: Discretization (3)

Solution 3:

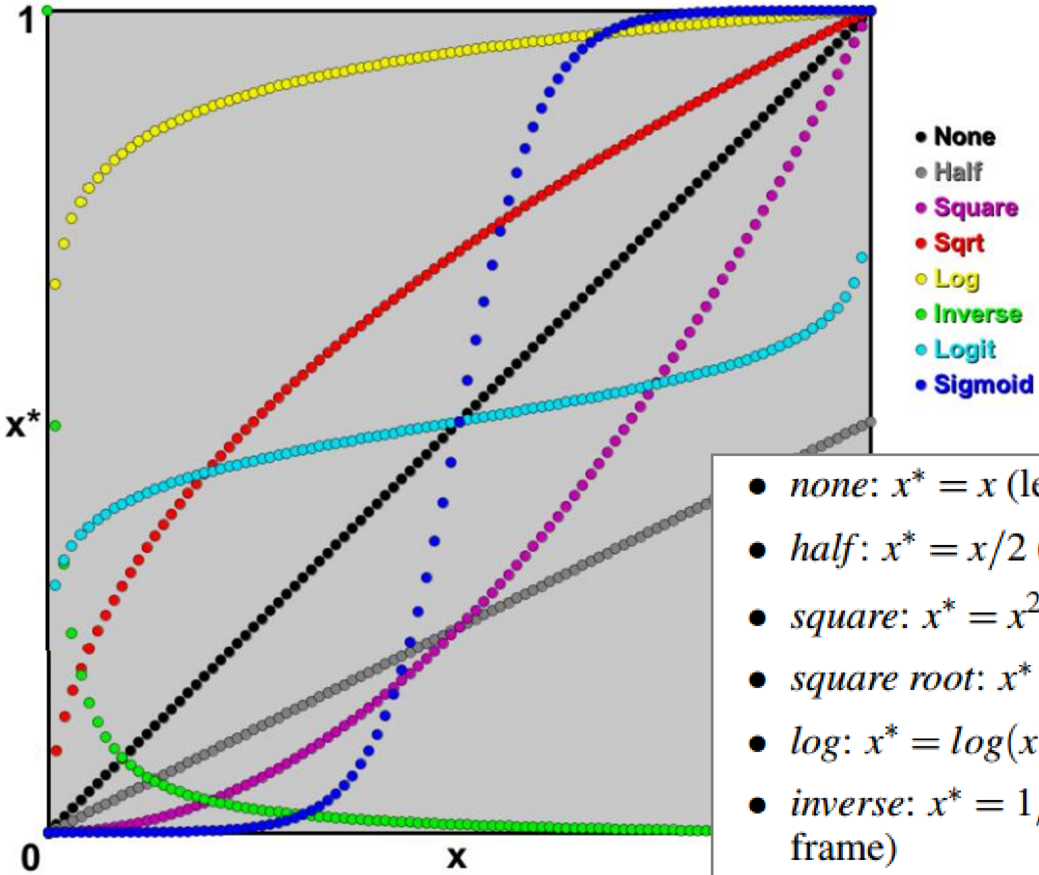
- what if all the bars have seemingly the same height
- or are dominated by one large peak



- switch to log scaling of the y-value



Other Transformations



- *none*: $x^* = x$ (leaves points unchanged)
- *half*: $x^* = x/2$ (squeezes all points together)
- *square*: $x^* = x^2$ (pulls points toward left of frame)
- *square root*: $x^* = \sqrt{x}$ (mildly pulls points toward right of frame)
- *log*: $x^* = log(x)$ (strongly pulls points toward right of frame)
- *inverse*: $x^* = 1/x$ (reverses scale and squeezes points into left of frame)
- *logit*: $x^* = (log(x/(1-x)) + 10)/20$ (squeezes points toward middle of frame)
- *sigmoid*: $x^* = 1/(1 + exp(-20x + 10))$ (expands points away from middle of frame)

Dang and Wilkinson, "Transforming Scagnostics to Reveal Hidden Features", TVCG 2014

Data Representation

Ever tried to reduce the size of an image and you got this?

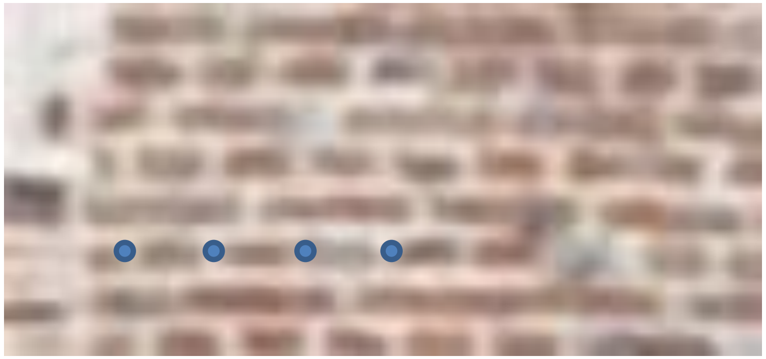
This is aliasing

Data Representation

But what you really wanted is this:

This is *anti-aliasing*

Why Is This Happening?



The smaller image resolution cannot represent the image detail captured at the higher resolution

- skipping this small detail leads to these undesired artifacts

What Is Anti-Aliasing

Procedure

- either sample at a higher rate
- or smooth the signal before sampling it
- the latter is called *filtering*

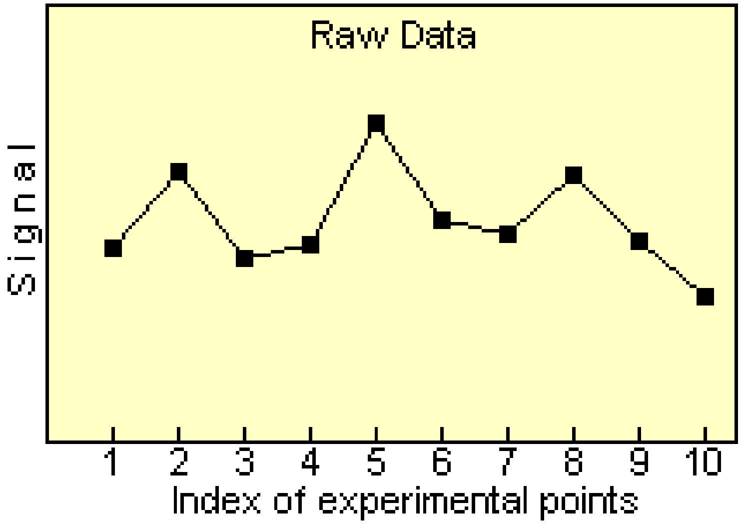
Anti-Aliasing Via Smoothing

Anti-Aliasing Via Smoothing

What is Smoothing?

Slide a window across the signal

- stop at each discrete sample point
- average the original data points that fall into the window
- store this average value at the sample point
- move the window to the next sample point
- repeat



Anti-Aliasing Via Smoothing: Tradeoffs

looks sharper, but has "jaggies"

a bit blurred, but no more jaggies

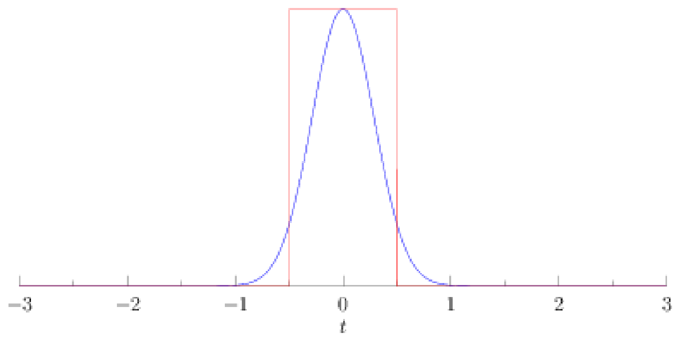
Filters

What is the filter we just used called?

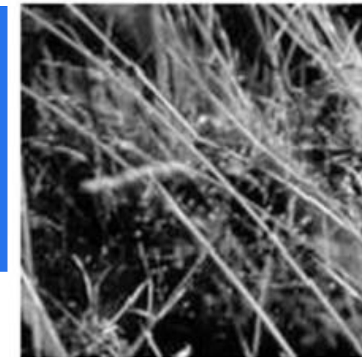
- it's called a *box filter*

There are other filters

- for example, Gaussian filter
- yields a smoother result
- box filtering is simplest

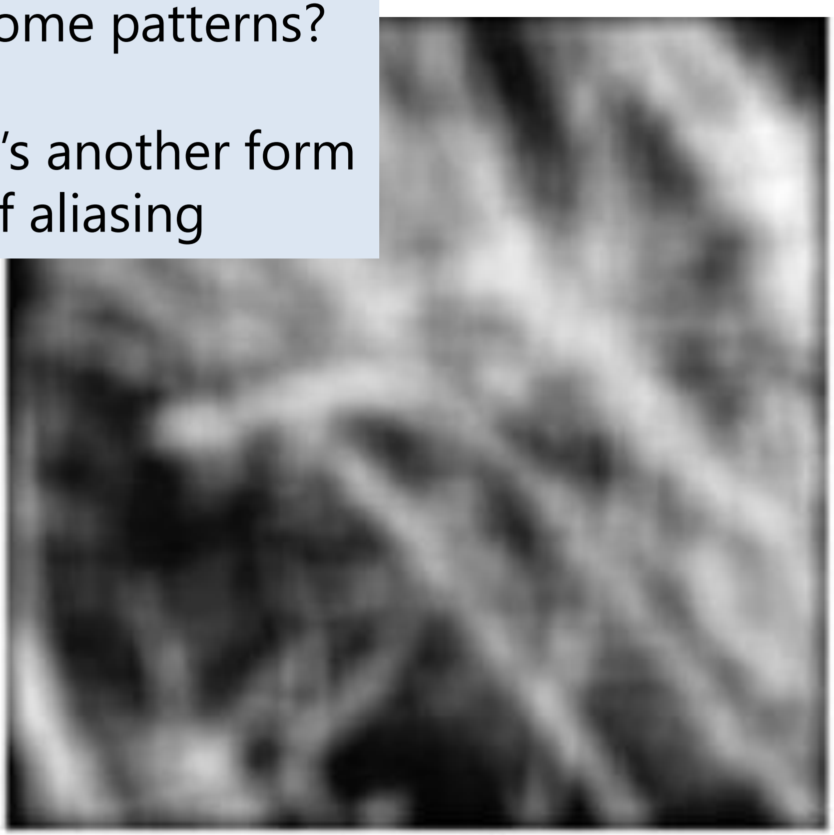


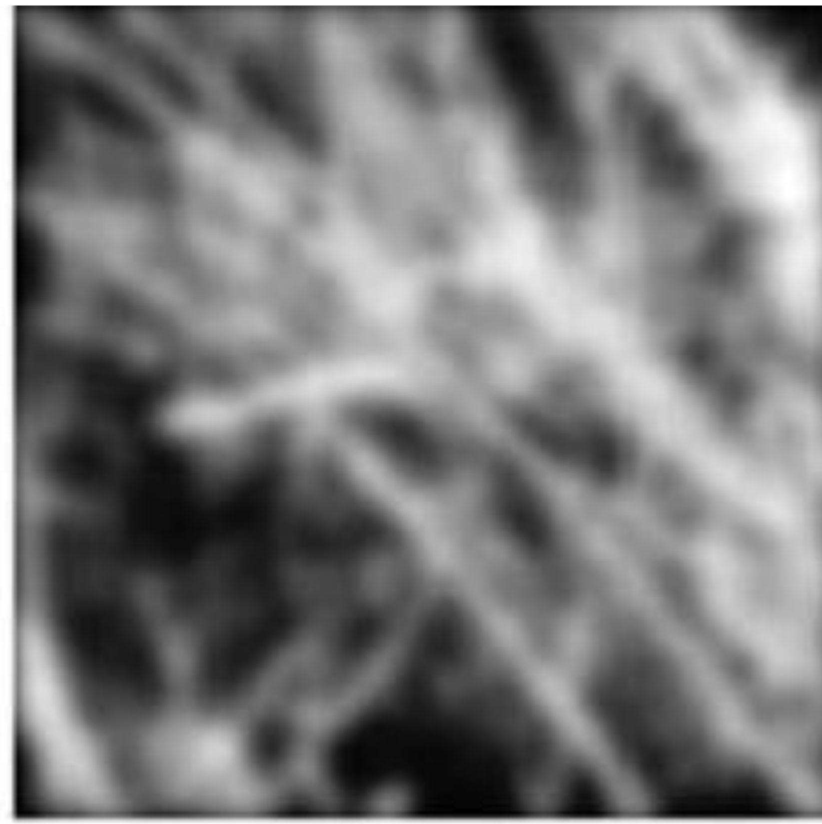
Box Filter vs. Gaussian Filter



Can you see some patterns?

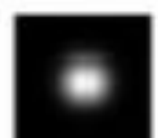
It's another form of aliasing





2D box

2D Gaussian



The Solution

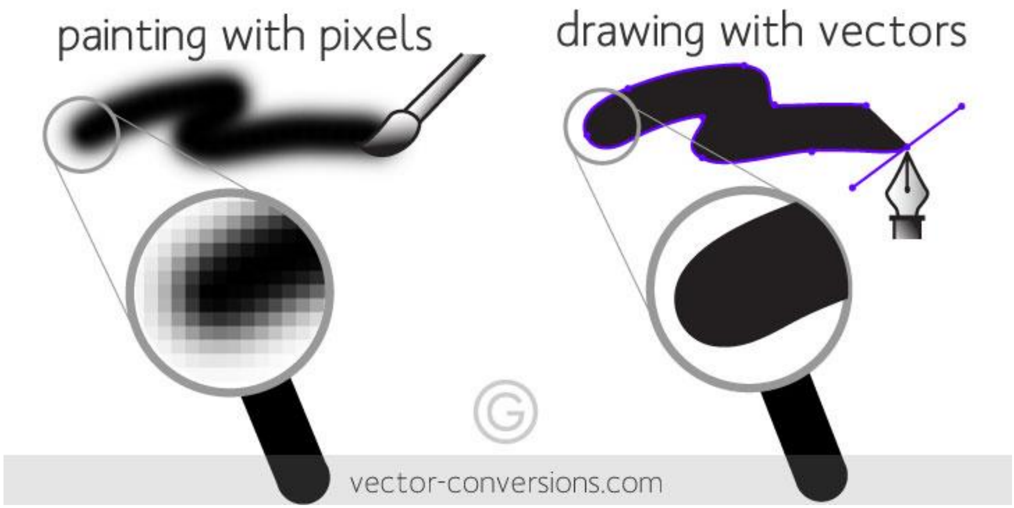
What's the underlying problem?

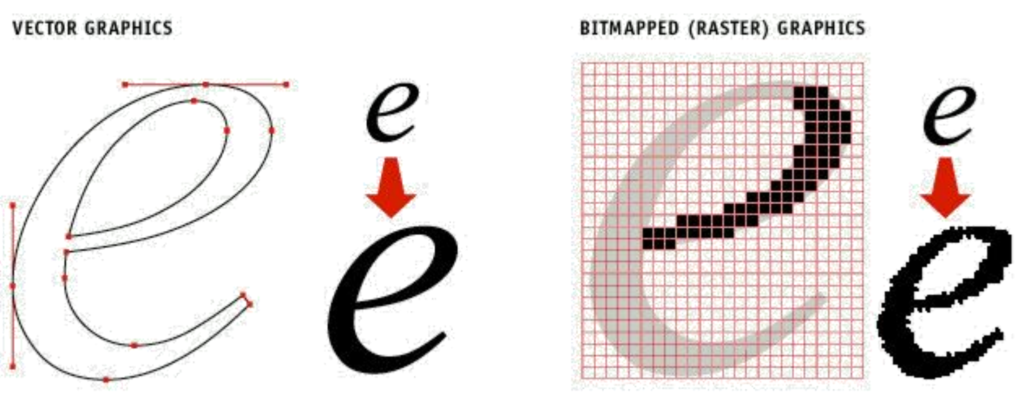
- detail can't be refined upon zoom
- can just be replicated or blurred

The solution…

- represent detail as a function that can be mathematically refined
- replace raster graphics by vector graphics

Scalable Vector Graphics (SVG)





Photographs and Images in SVG

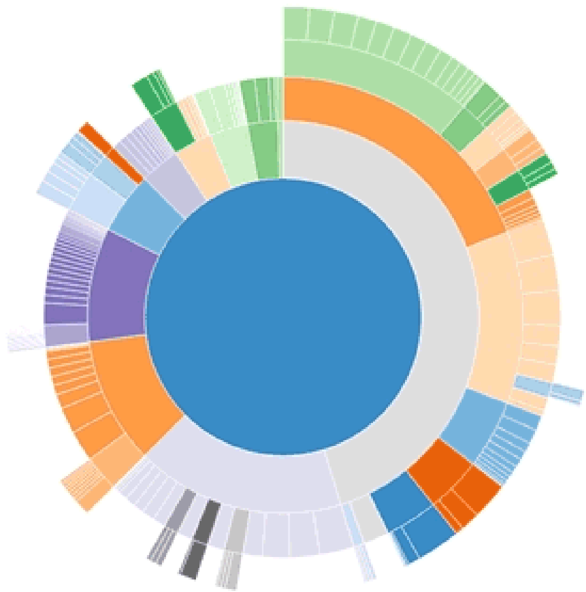
Vector graphics tends to have an “cartoonish” look

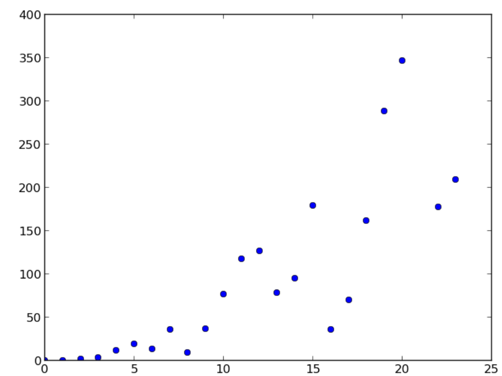
raster graphics

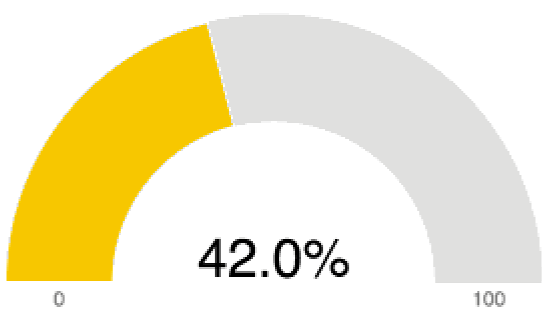
vector graphics

Photographs and Images in SVG

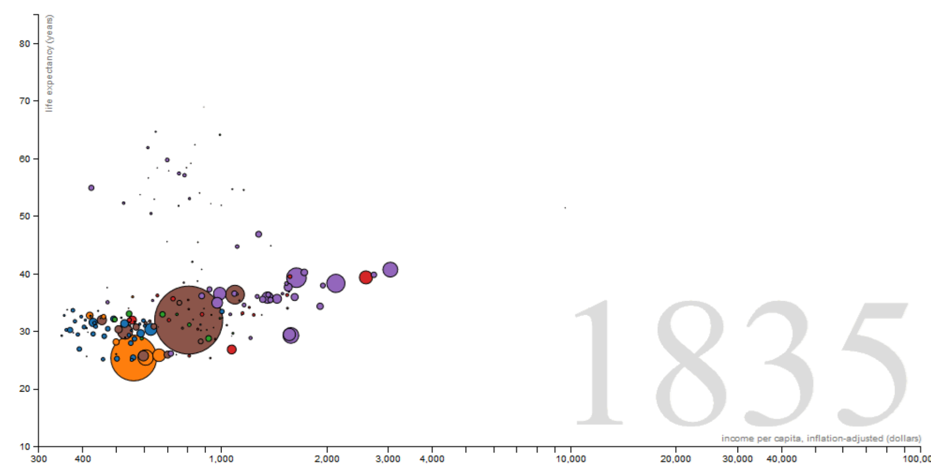
D3 Uses SVG





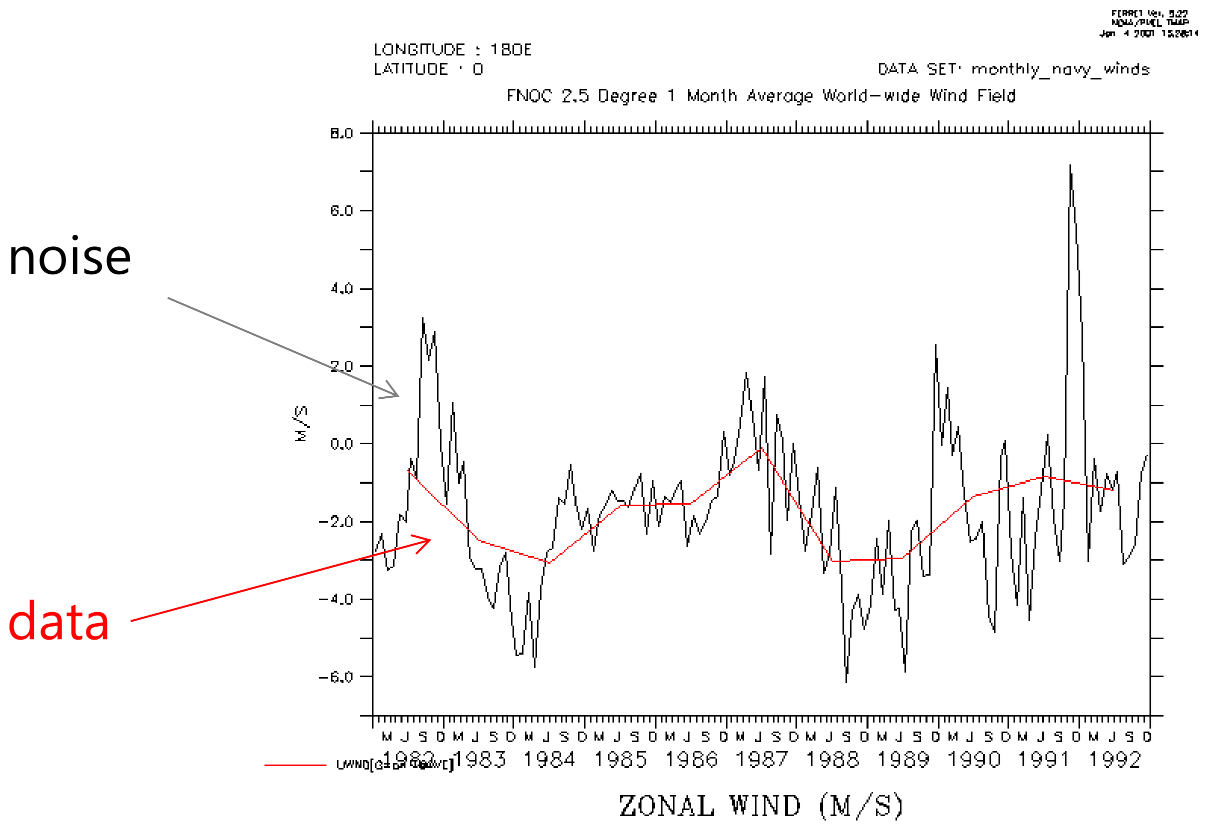


The Wealth & Health of Nations



Smoothing for De-Noising

Filtering/smoothing also eliminates noise in the data



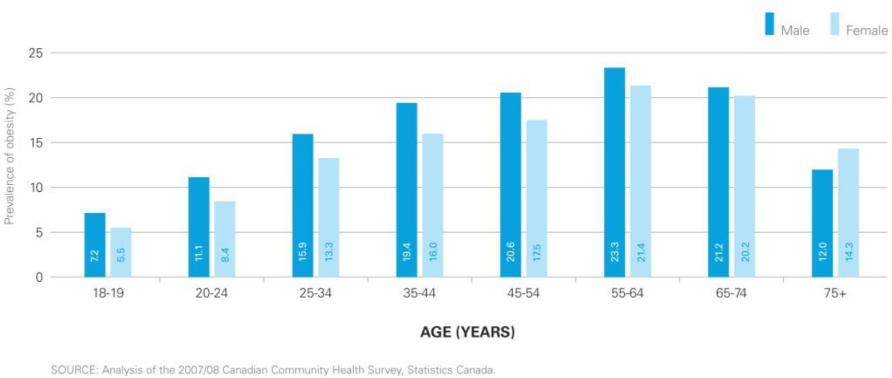
Back to Bar Charts

In some ways, bar charts reduce noise and uncertainties in the data

- the bins do the smoothing

Example:

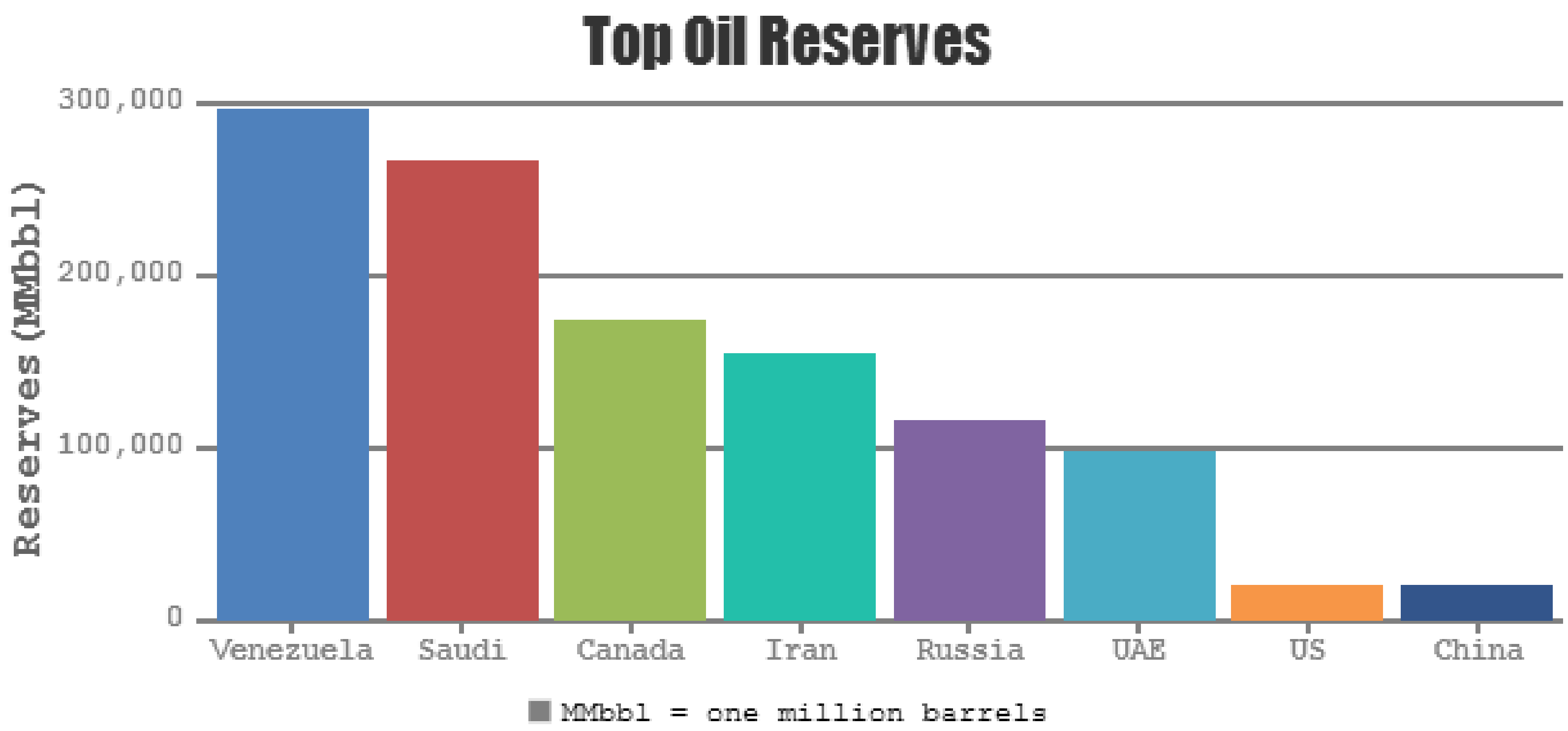
- obesity over age (group)



Bar Charts

Of course, bar charts can also hold categorical data

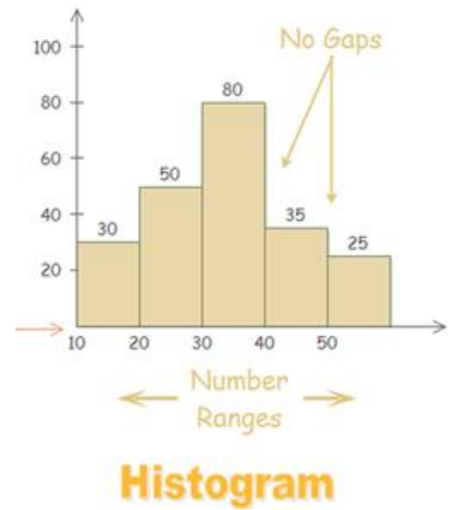
- smoothing by semantic grouping
- for example, Europe vs. {France, Spain, Italy, Germany, …}



Bar Charts vs. Histograms

Histograms

- bars show the frequency of numerical data
- quantitative data
- elements are grouped together, so that they are considered as ranges
- bars cannot be reordered
- width of bars need not be the same



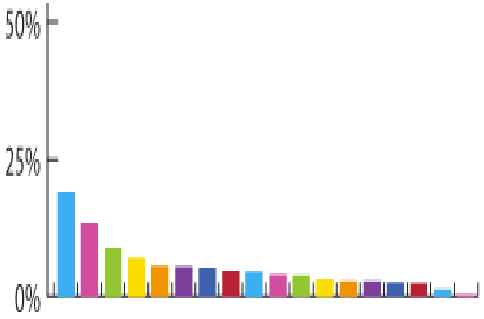
Bar charts

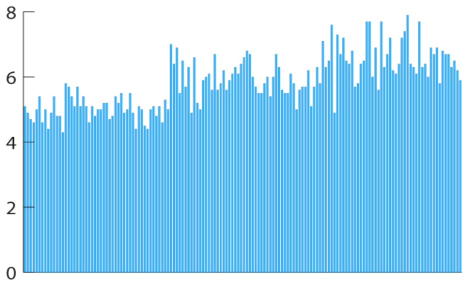
- uses bars to compare different categories of data
- comparison of discrete variables
- elements are taken as individual entities
- bars can be reordered
- width of bars need to be the same

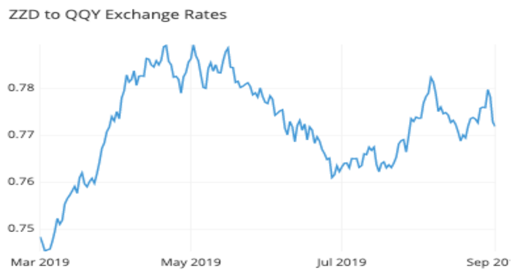
How many Bars in a Bar Chart

How many bars are too many (in a chart)

- if individual categories are the focus? 12 is a good rule
- if the overall trend is the important factor? 50 or even more
- eventually you can switch to a line chart







- sort bars by height and use ‘other’ to aggregate the bar chart tails into a single bar
- find a grouping that can semantically aggregate bars, for example aggregate countries into continents

[more information]

Bar Charts in D3

https://observablehq.com/@d3/bar-chart

Working with bar charts and histograms is the topic of Lab 1

- the next two slides offer some help with calculations

Histogram Calculations – Binning

Determine bin size

- min(data) is optional, can also use 0 or some reasonable value
- max(data) is optional, can also use some reasonable value

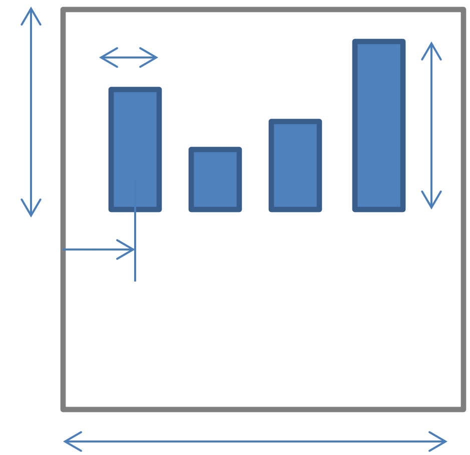
$$bin\ size = \frac{\max(data) - \min(data)}{number\ of\ bins}$$

Given a data value *val* increment (++) the bin value

- but first initialize bin val array to 0

$$bin\ val\ array\left[\left\lfloor\frac{val - \min(data)}{bin\ size}\right\rfloor\right]++$$

Histogram Calculations – Plotting



Determine bin size on the screen

$$bin\ size\ on\ screen = \frac{chart\ width}{number\ of\ bins}$$

Center of a bar for bin with index *bin index*

$$bar\ center\ on\ screen = (bin\ index\ \cdot bin\ size\ on\ screen) + 0.5$$

Height of the bar for a bin with index *bin index*

$$bar\ height(bin\ index) = bin\ val\ array(bin\ index) \cdot \frac{chart\ height}{\max(bin\ val\ array)}$$

Do not forget that the origin of a web page is the top left corner