Foundations of Data Science

DS 3001

Data Science Program
Department of Computer Science
Worcester Polytechnic Institute
Instructor: Prof. Kyumin Lee

Upcoming Schedule

- Project Proposal
  - https://canvas.wpi.edu/courses/18106/assignments/132329
  - Due date: Today
- HW3 will be out this Friday

Data Science: The Context

Ask question: What data needs to be recorded? or collected?

Why? What research question am I going to answer?

What do I want it to look like?

Real World

Humans behaving
Biology
Finance
Internet
Medicine
Sociology
Olympics

→ **Raw Data is Collected / Recorded**

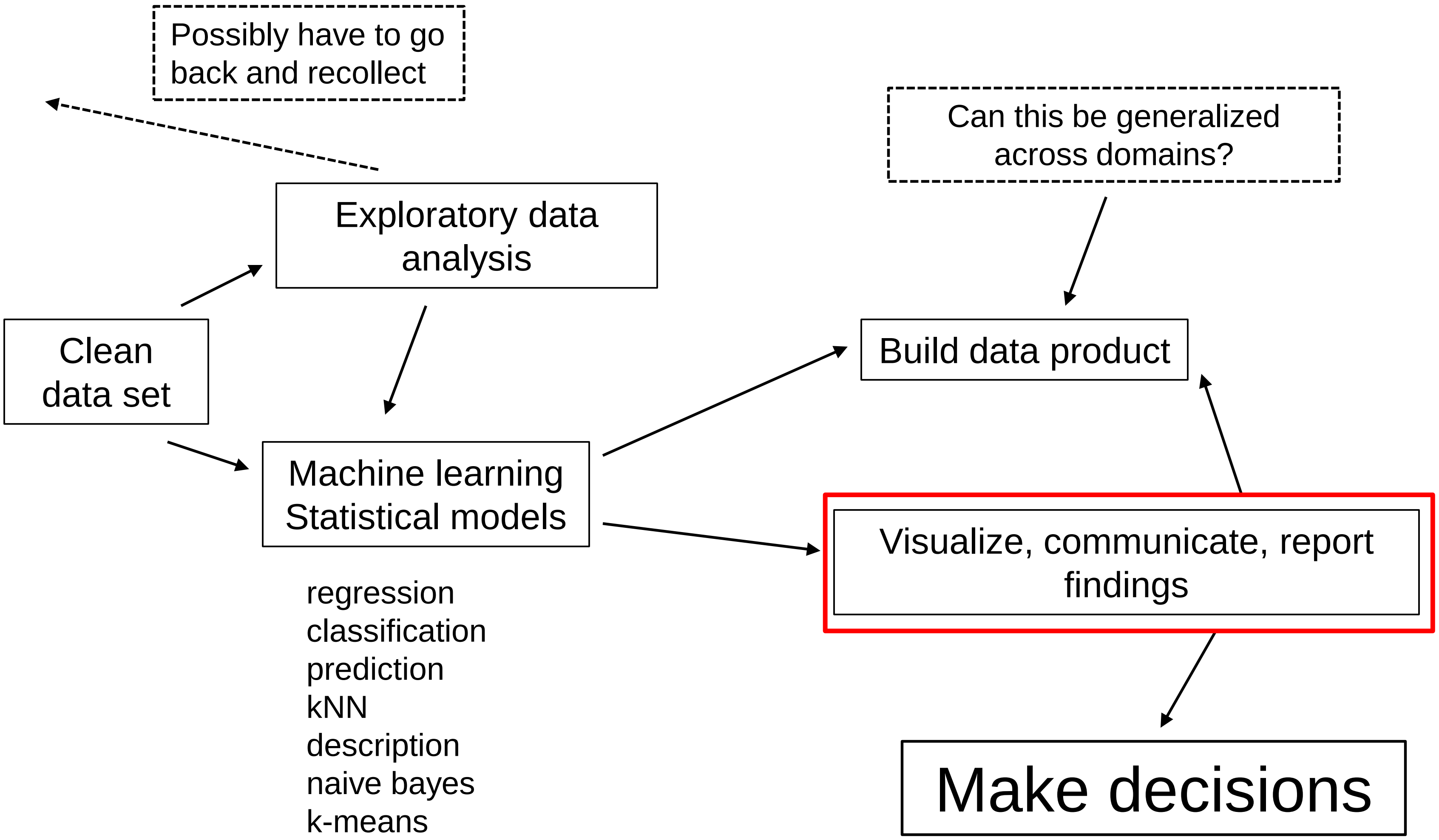
email
logs
medical records
surveys
blood drawn
(microarray)
olympic records
NYT web pages

→ **Data is Processed**

pipelines
web scraping
cleaning
munging
joining
wrangling

→ **Data Set**

“clean” table


Possibly have to go back and recollect
Can this be generalized across domains?
Exploratory data analysis
Clean data set
Build data product
Machine learning Statistical models
Visualize, communicate, report findings
regression
classification
prediction
kNN
description
naive bayes
k-means
Make decisions

Data Visualization

The Value of Visualization

- **Record** information
  - Blueprints, photographs, seismographs, …
- **Analyze** data to support reasoning
  - Develop and assess hypotheses
  - Discover errors in data
  - Expand memory
  - Find patterns
- **Communicate** information to others
  - Share and persuade
  - Collaborate and revise

Tufte: Principles of Graphical Excellence

- Graphical excellence is the well-designed presentation of interesting data – a matter of *substance*, *statistics*, and *design*

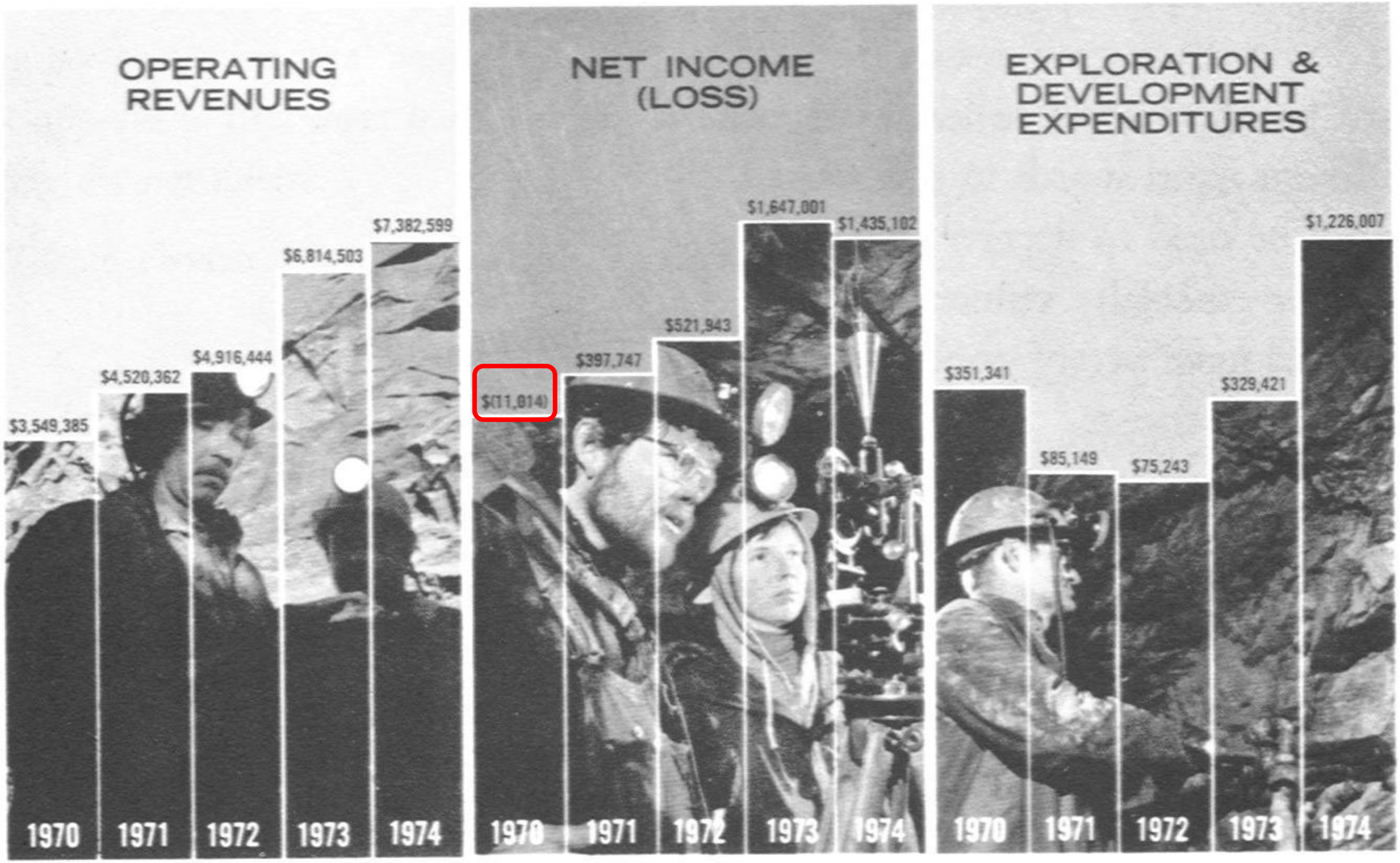
- Graphical excellence consists of complex ideas communicated with *clarity*, *precision*, and *efficiency*

Tufte: Graphical Integrity

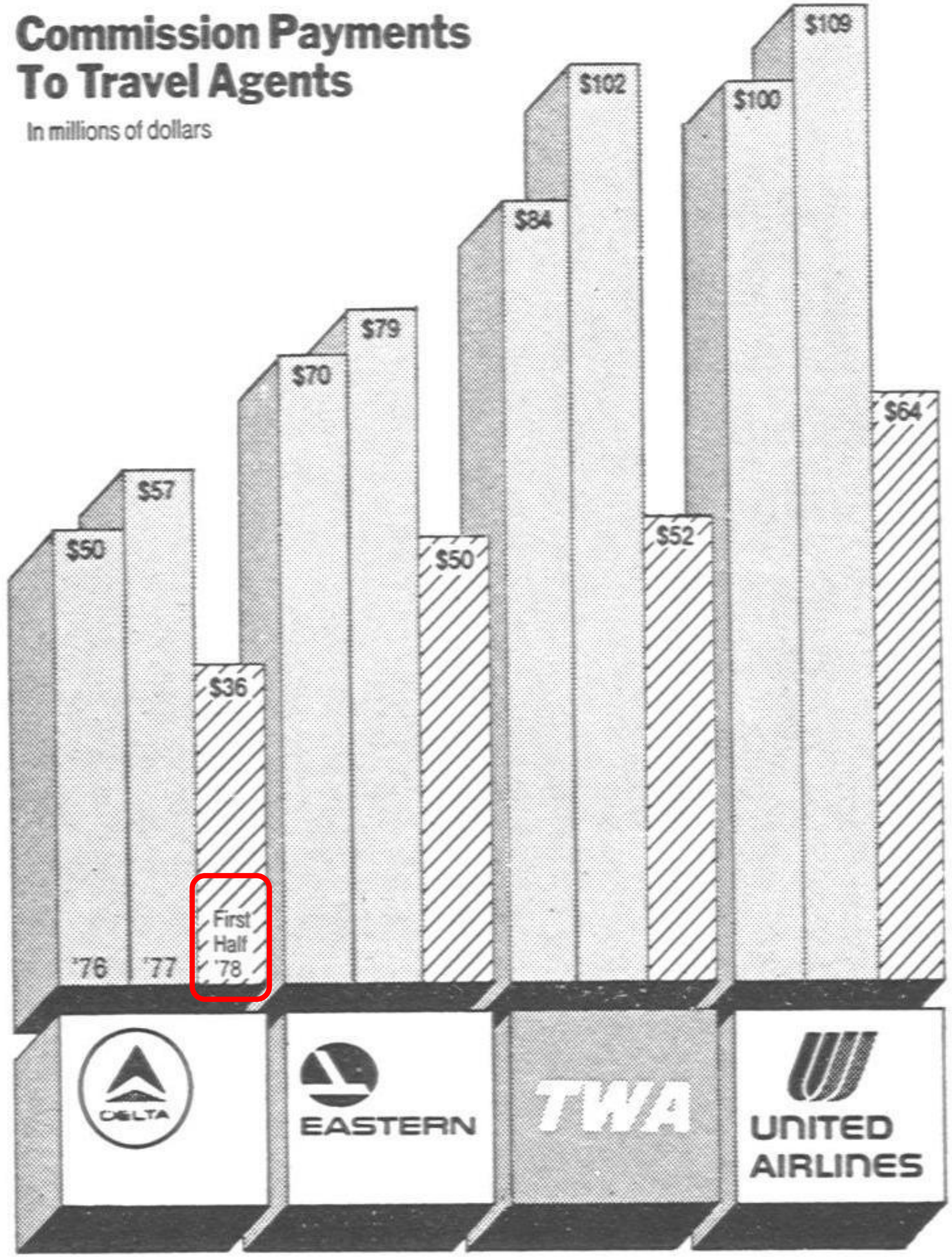
- “not lying with statistics”
- tell the truth about data

Uh oh …

Examples of Infographics lacking integrity



Day Mines, Inc., 1974 Annual Report



New York Times, 8/8/78

Comparative Annual Cost per Capita for care of Insane in Pittsburgh City Homes and Pennsylvania State Hospitals.

$147

South Mountain

$172

Pittsburgh

$198

Harrisburg

$213

Norristown

$214

Warren

Lie Factor

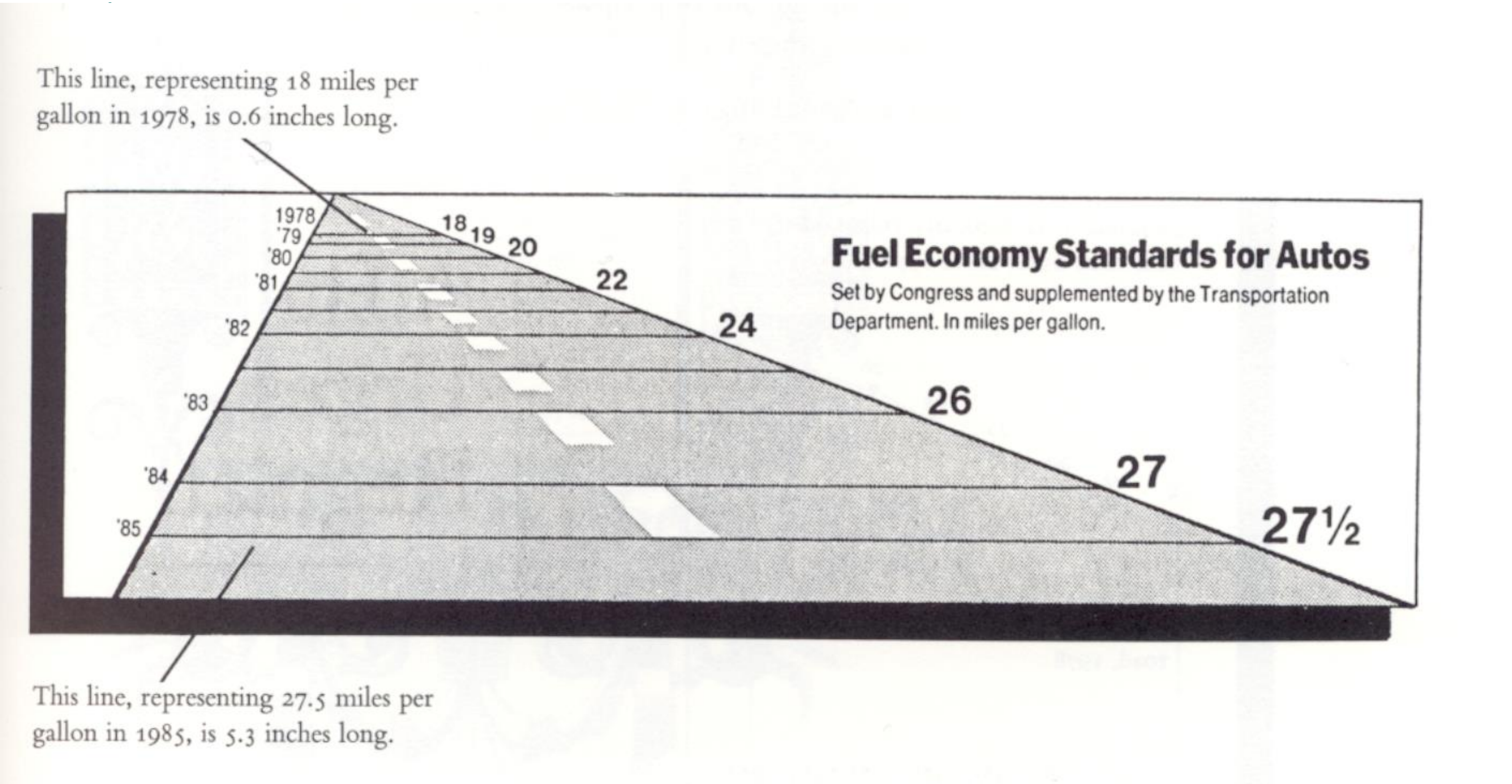
- Given perceptual difficulties – strive for uniformity (predictability) in graphics (p56)
  - ‘the representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented.’
  - ‘Clear, detailed and thorough labeling 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.

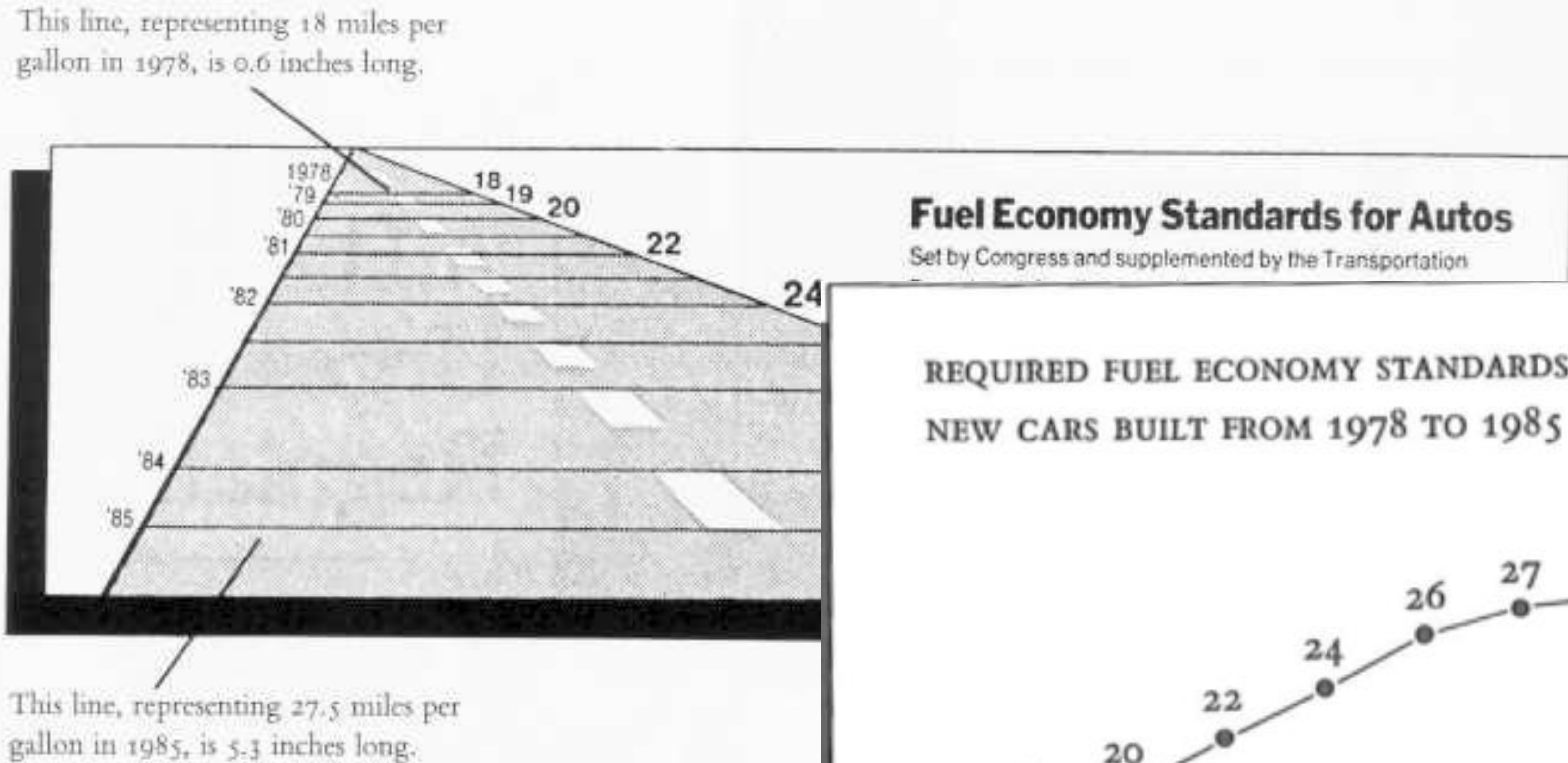
$$\text{Lie Factor} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$$

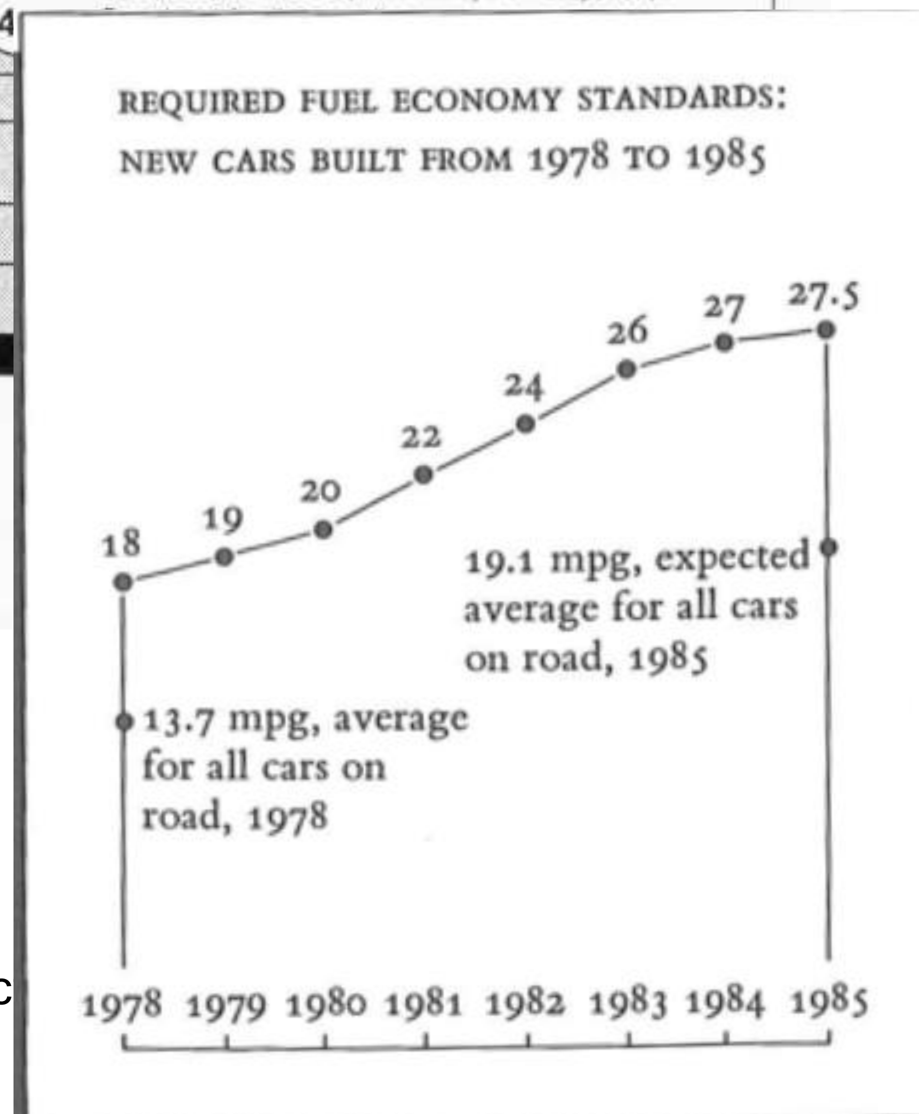
- Lie factor of 1 ➔ is desirable
- Lie factor > 1.05 or < 0.95 go beyond plotting errors

Extreme example

- Fuel economy standards for automobiles
  - 18 miles/gallon in 1978 to 27.5 miles/gallon in 1985
  - (27.5 – 18.0)/(18.0) x 100 = 53% increase







- Graphic increase
  (5.3 – 0.6)/(0.6) x 100 = 783%
- Lie Factor = 783/53 = 14.8
- Additional confounding factors
  Usually the future is in front of us
  Dates remain same size and fuel fac

Visual Area and Numerical Measure

Use of area to portray 1D data can be confusing

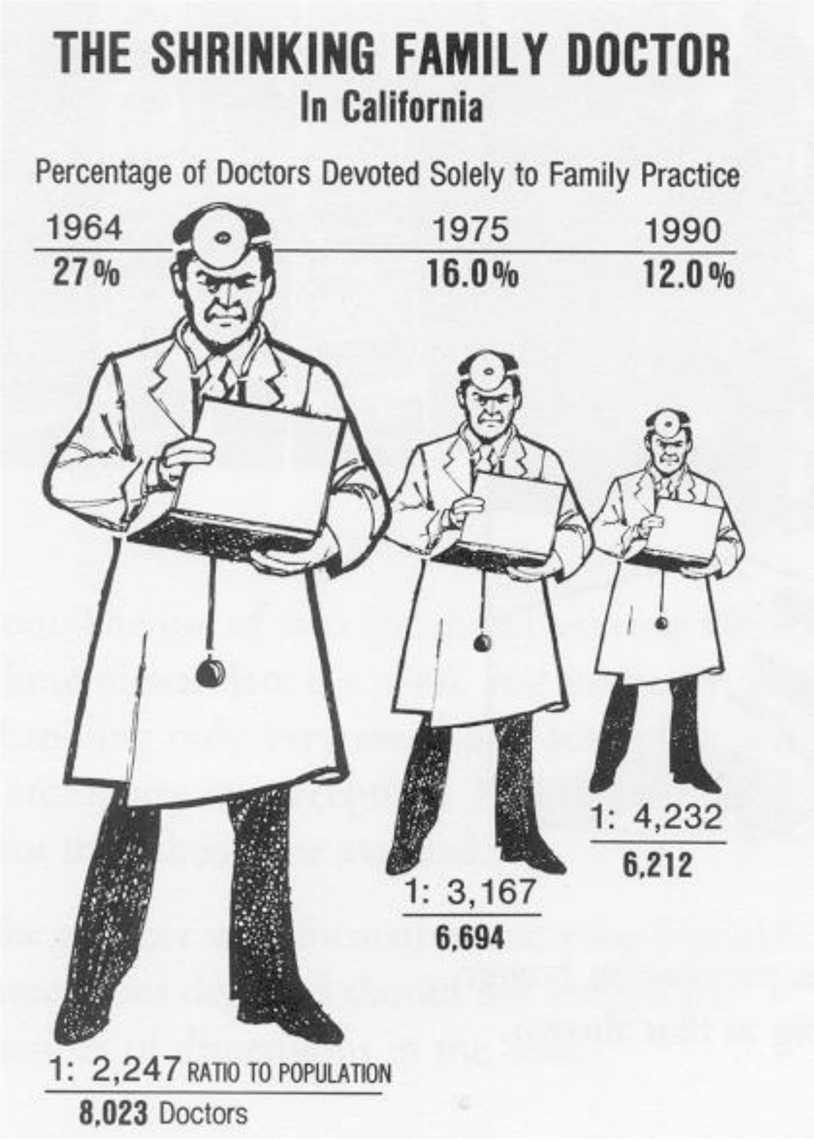
-Area has 2 dimensions

The ‘incredible’ shrinking family doctor

Lie factor of 2.8

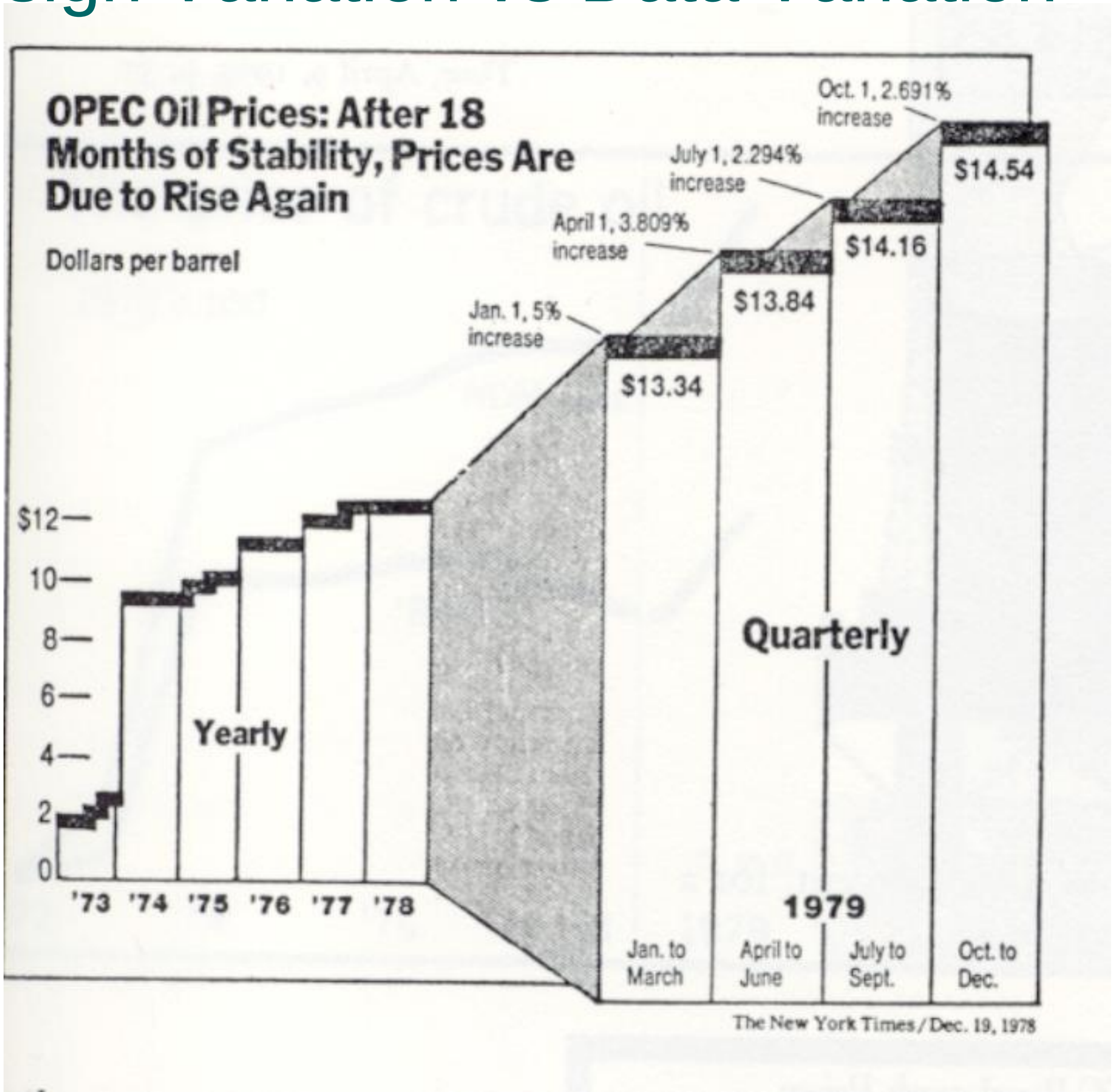
Plus incorrect horizontal spacing

Los Angeles Times, August 5, 1979 p.3, (Tufte, 1983, p69)



Design Variation vs Data Variation

New York Times, Dec. 19, 1978, p.D-7 (Tufte, 1983, p61)



Design Variation vs Data Variation

- 5 different vertical scales show price

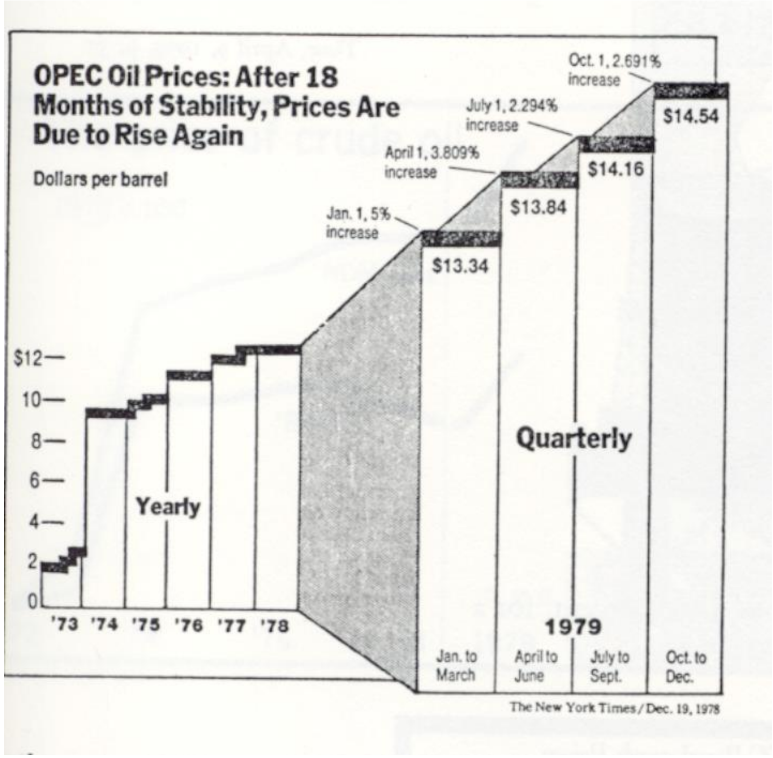
During this time	one vertical inch equals
1973 -1978	$8.00
Jan. – Mar. 1979	$4.73
Apr. – June 1979	$4.37
Jul. – Sept. 1979	$4.16
Oct. – Dec. 1979	$3.92

- 2 different horizontal scales show passage time

During this time	one horizontal inch equals
1973-1978	3.8 years
1979	0.57 years

- With both scales shifting the distortion is multiplicative

Show data variation, not design variation!



National Science Foundation, Science Indicators, 1974 (Washington D.C., 1976), p.15, (Tufte, 1983, p60)

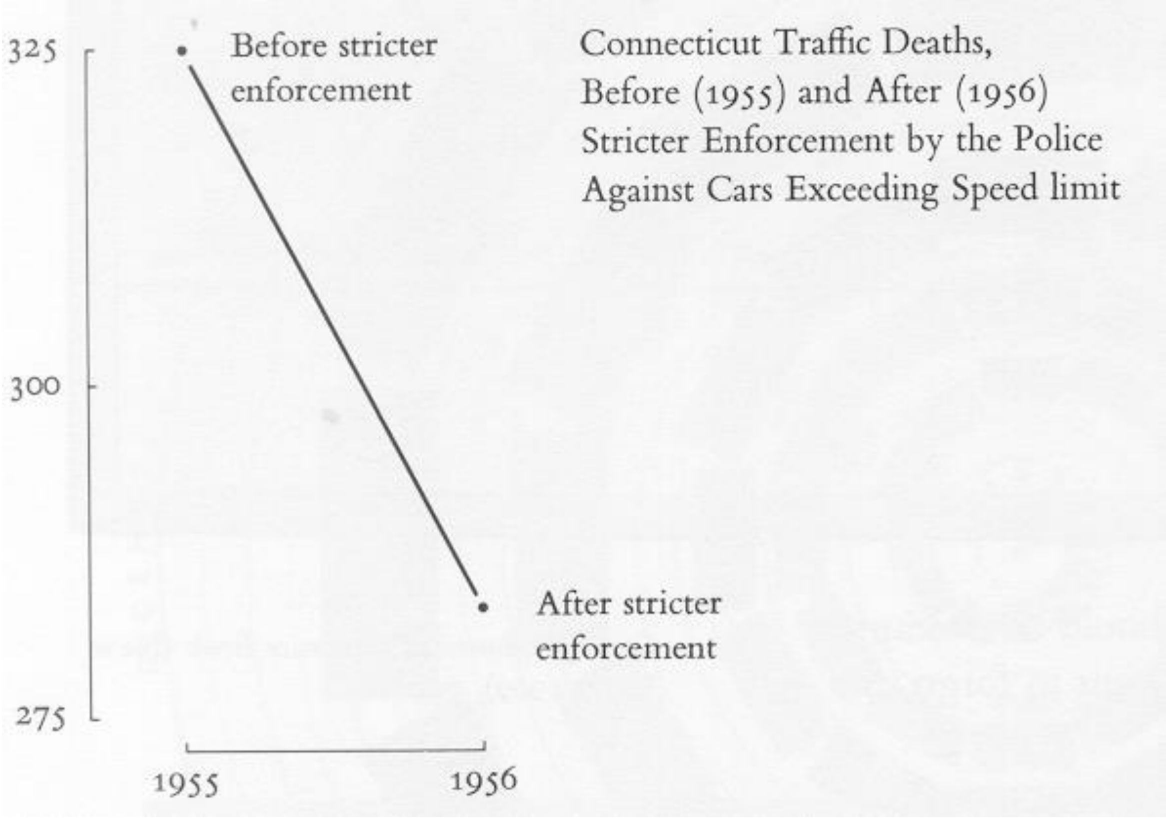
Context is Essential

Graphics must not quote data out of context

Data sparse graphics should provoke suspicion

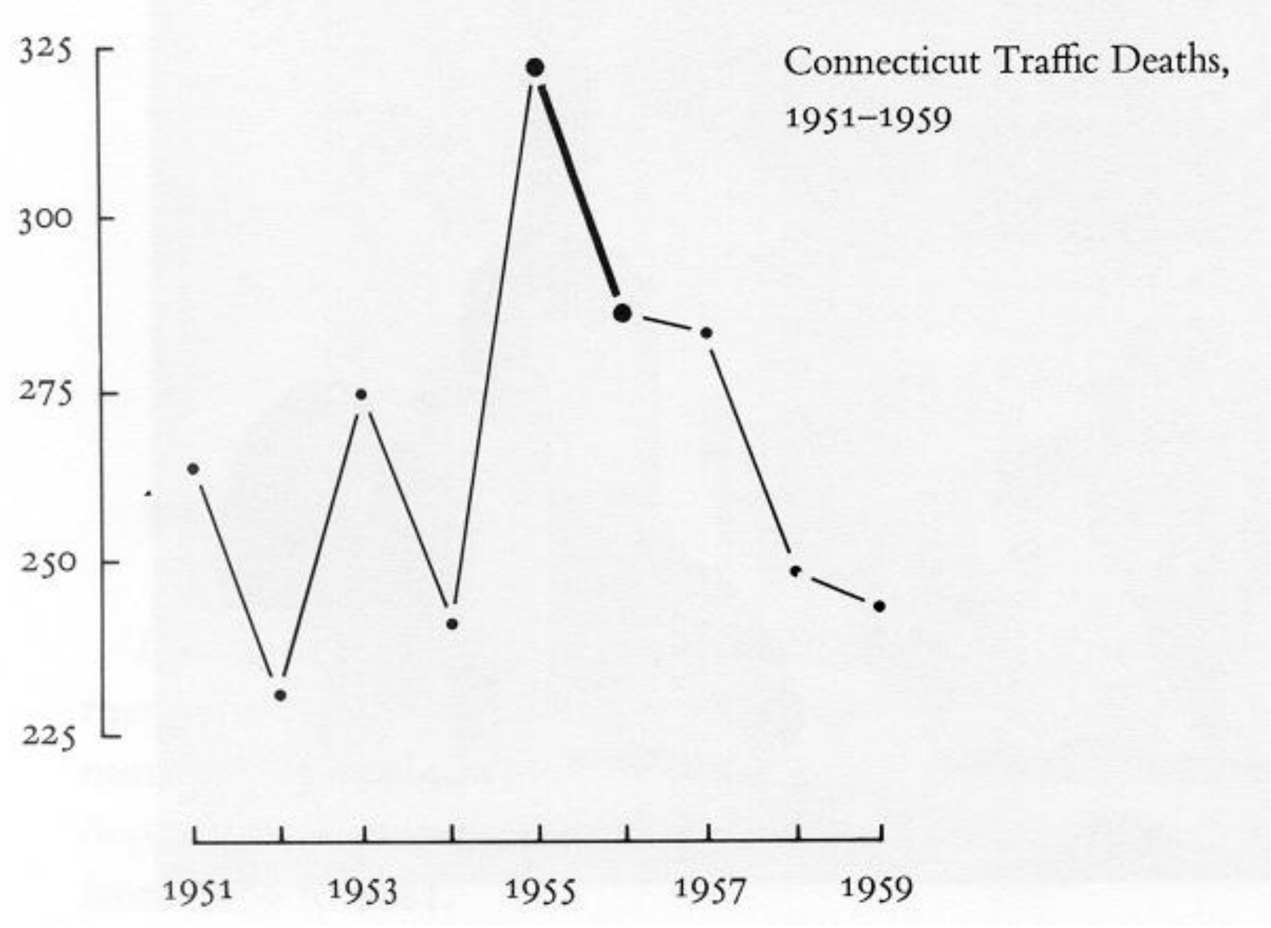
Graphics often lie by omission

Nearly all important questions are left unanswered by this graph



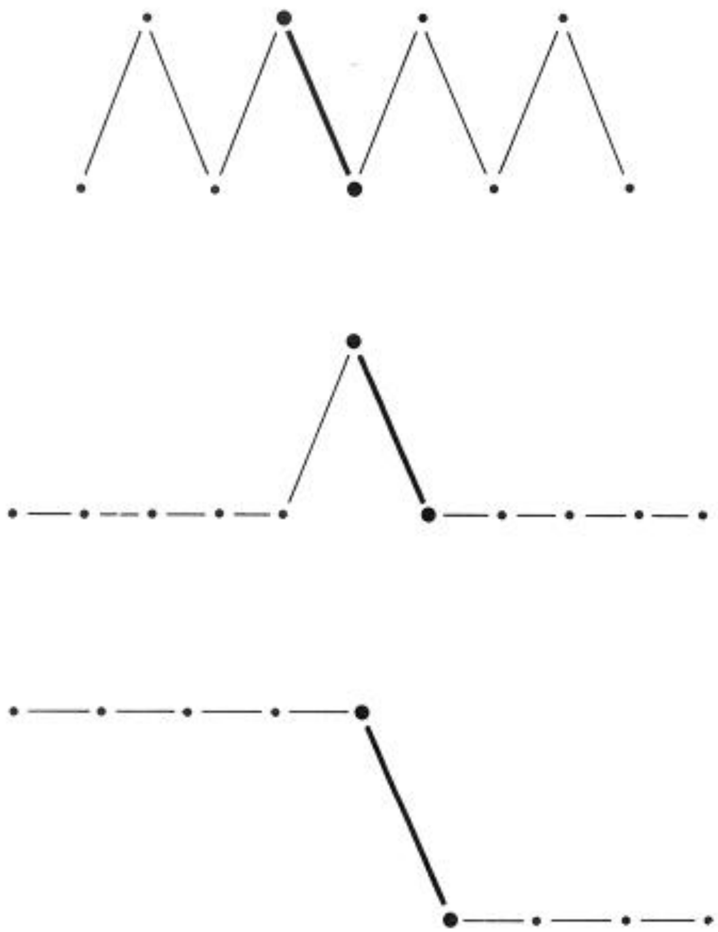
Context is Essential

A few more data points tell a more complete story



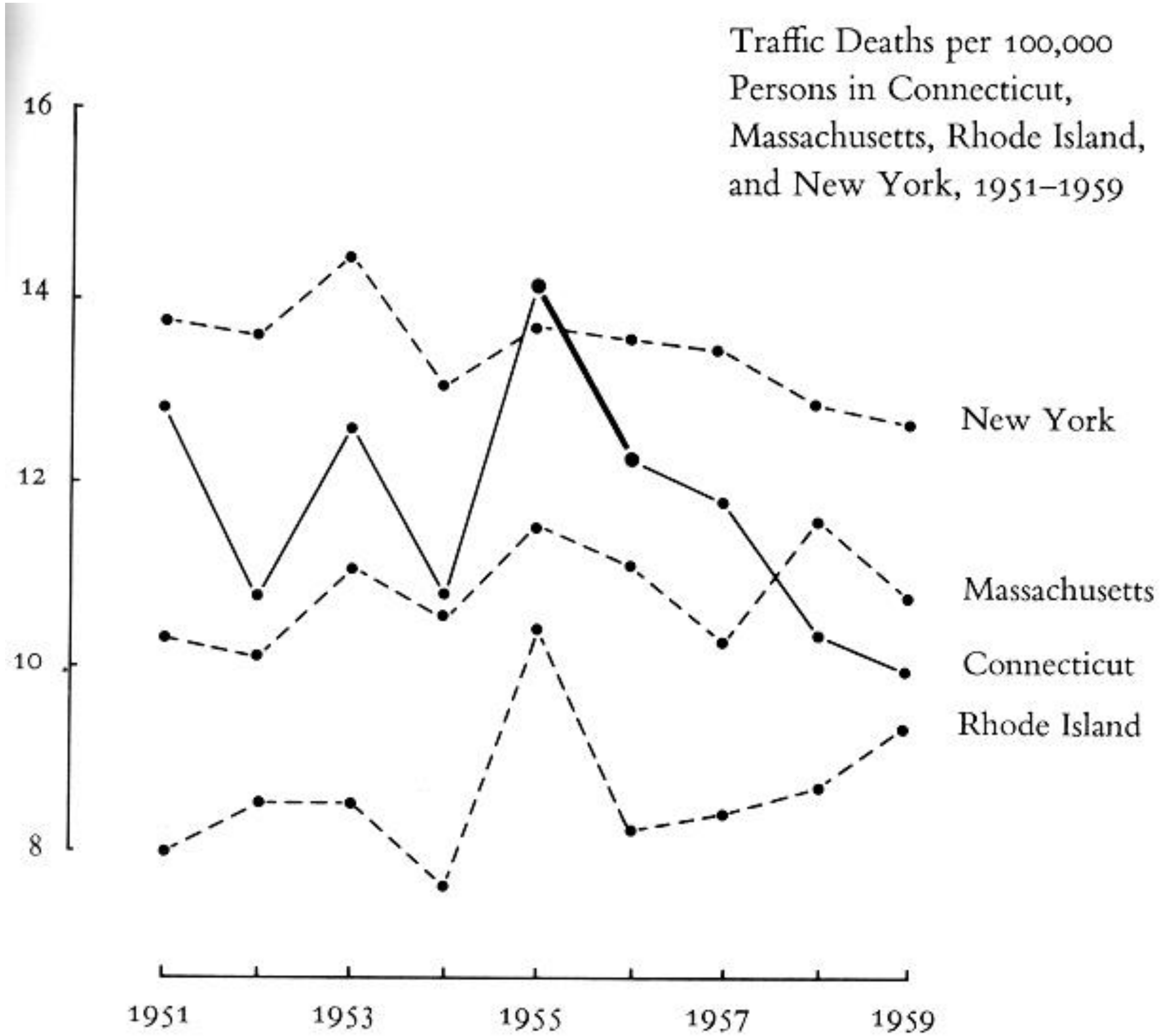
Context is Essential

Different data points would tell
a different stories



Context is Essential

Comparisons with adjacent states give more context



Donald T. Campbell and H. Laurence Ross, "The Connecticut Crackdown on Speeding: Time Series Data in Quasi-Experimental Analysis," in Edward R. Tufte, ed., *The Quantitative Analysis of Social Problems* (Reading, Mass., 1970), 110–125.

Tufte: Principles of Graphical Excellence

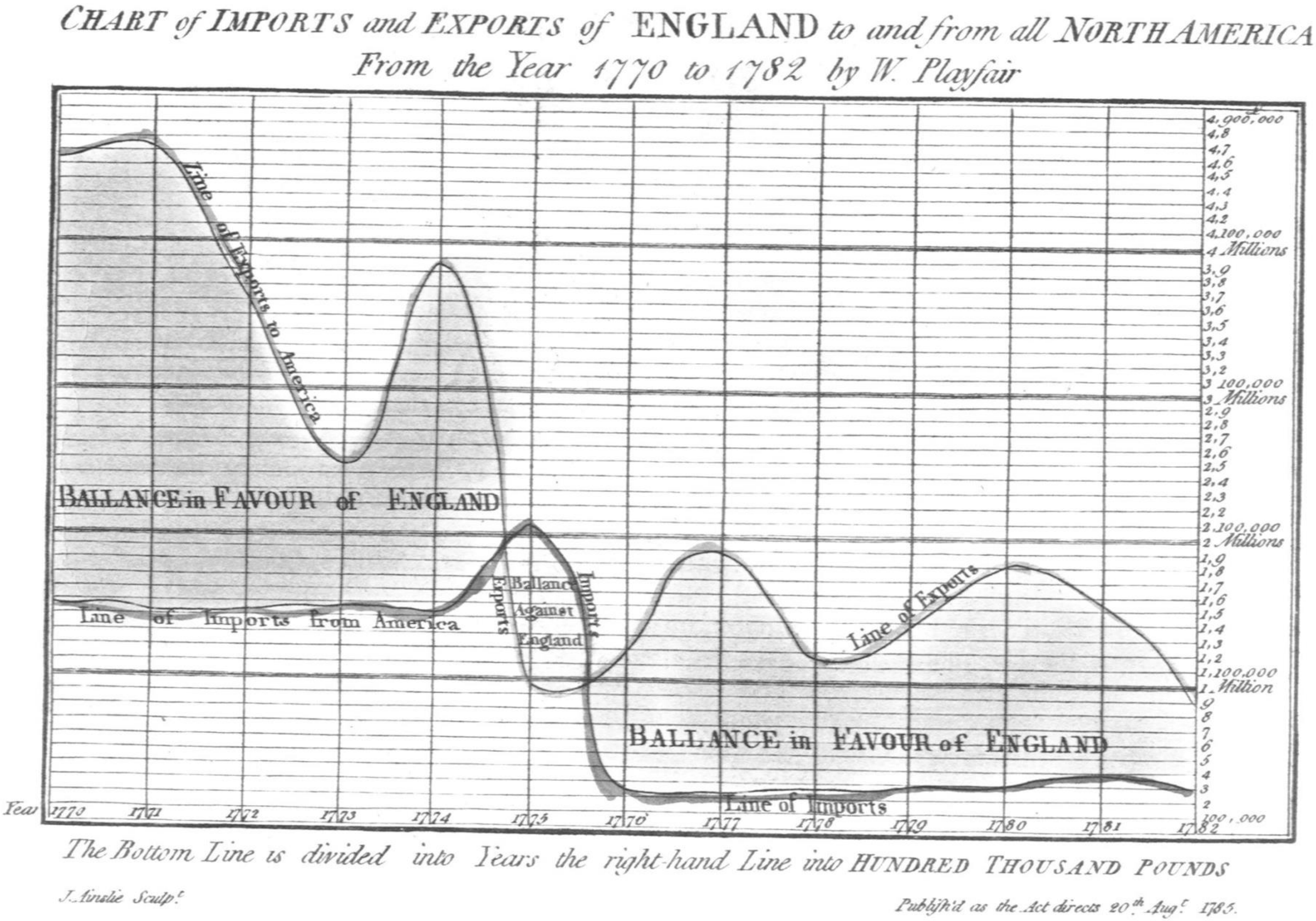
- Graphical excellence is the well-designed presentation of interesting data – a matter of *substance*, *statistics*, and *design*

- Graphical excellence consists of complex ideas communicated with *clarity*, *precision*, and *efficiency*

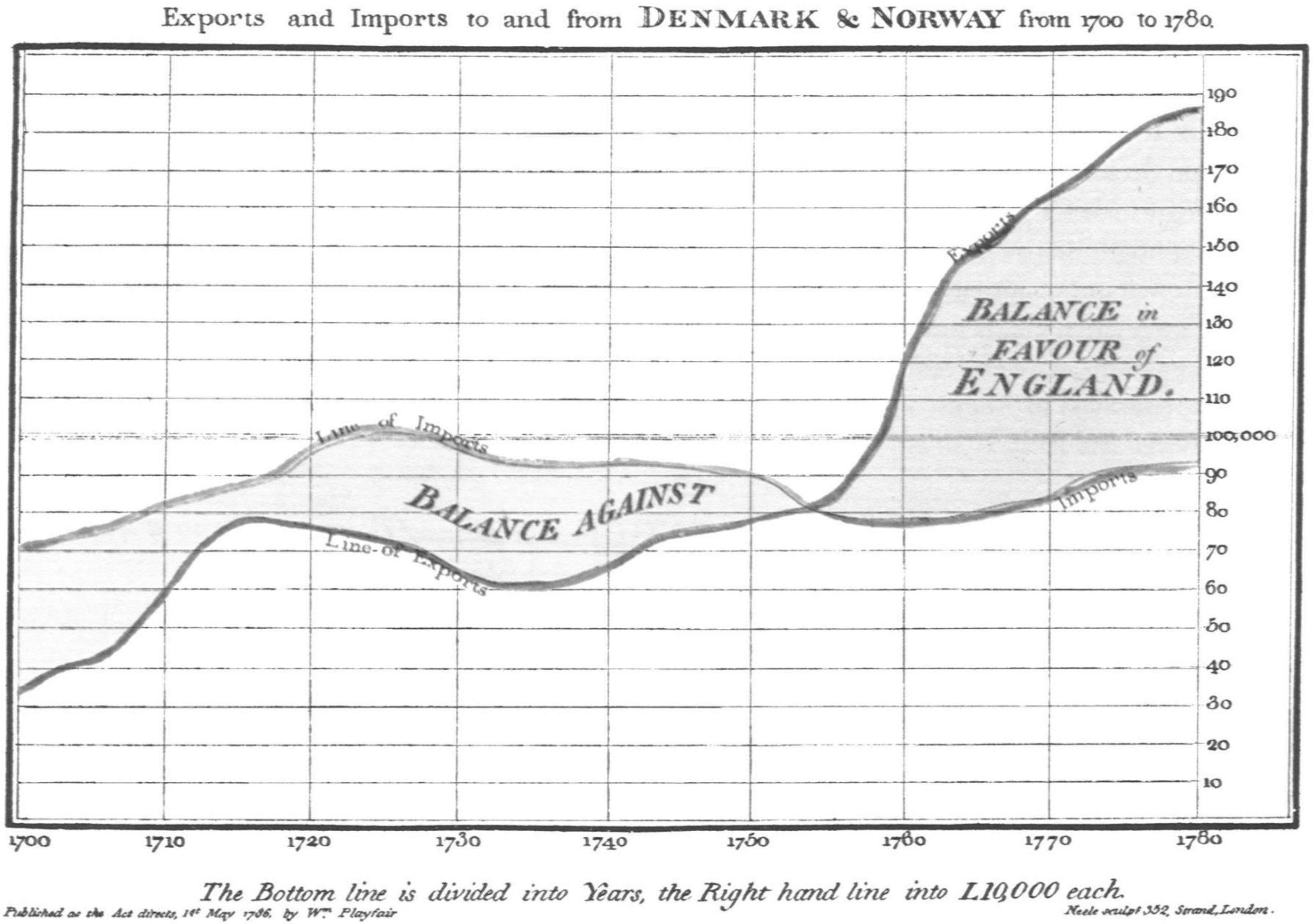
Tufte’s principles for better viz?

- Above all else, show the data
- Maximize the data-ink ratio
  - Erase non-data-ink
  - Erase redundant data-ink
- Revise and edit

Above all else, show the data



Above all else, show the data



Maximize the data-ink ratio

$$\text{Data-ink ratio} = \frac{\text{data-ink}}{\text{Total ink used to print graphic}}$$

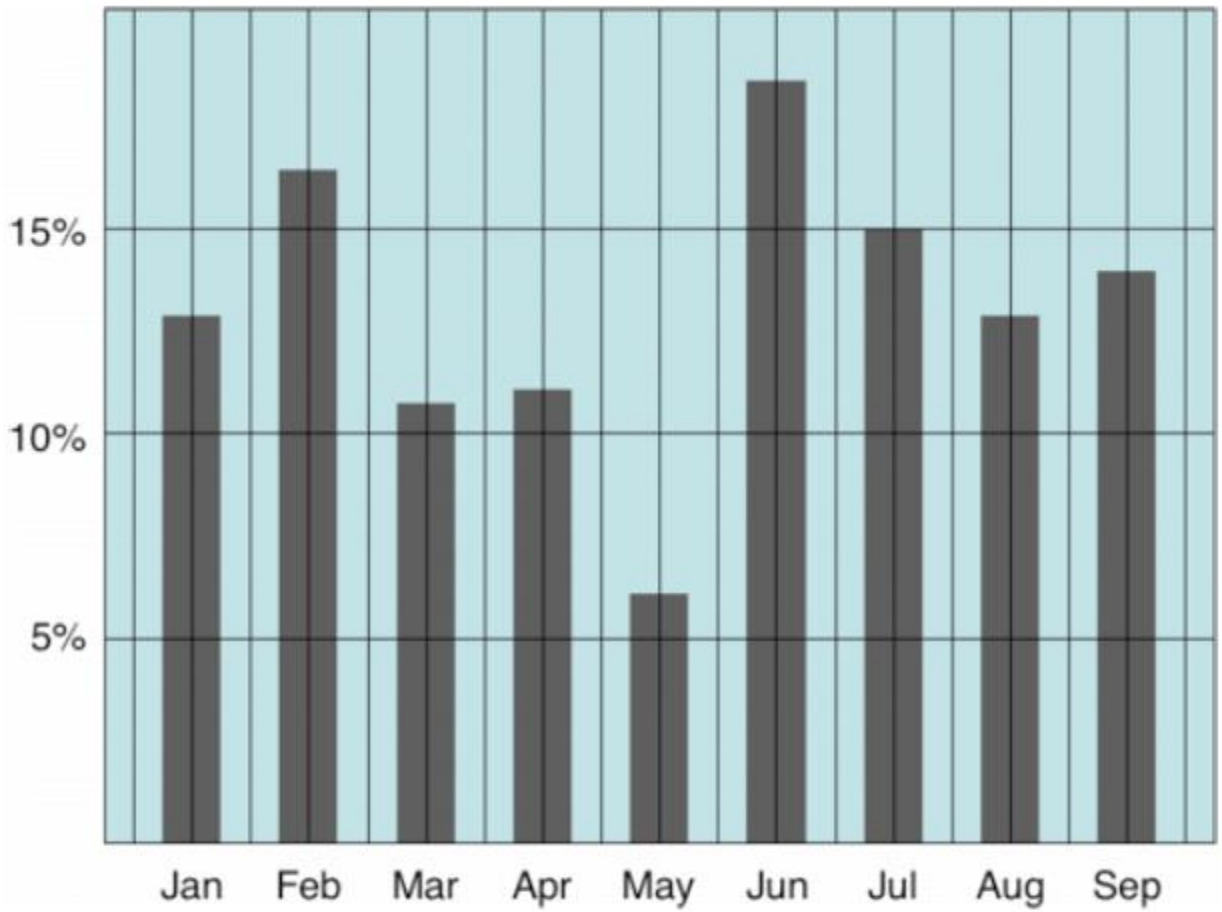
= Proportion of a graphic's ink devoted to the non-redundant display of data-information.

= 1.0 – proportion of graphic that can be erased without the loss of information

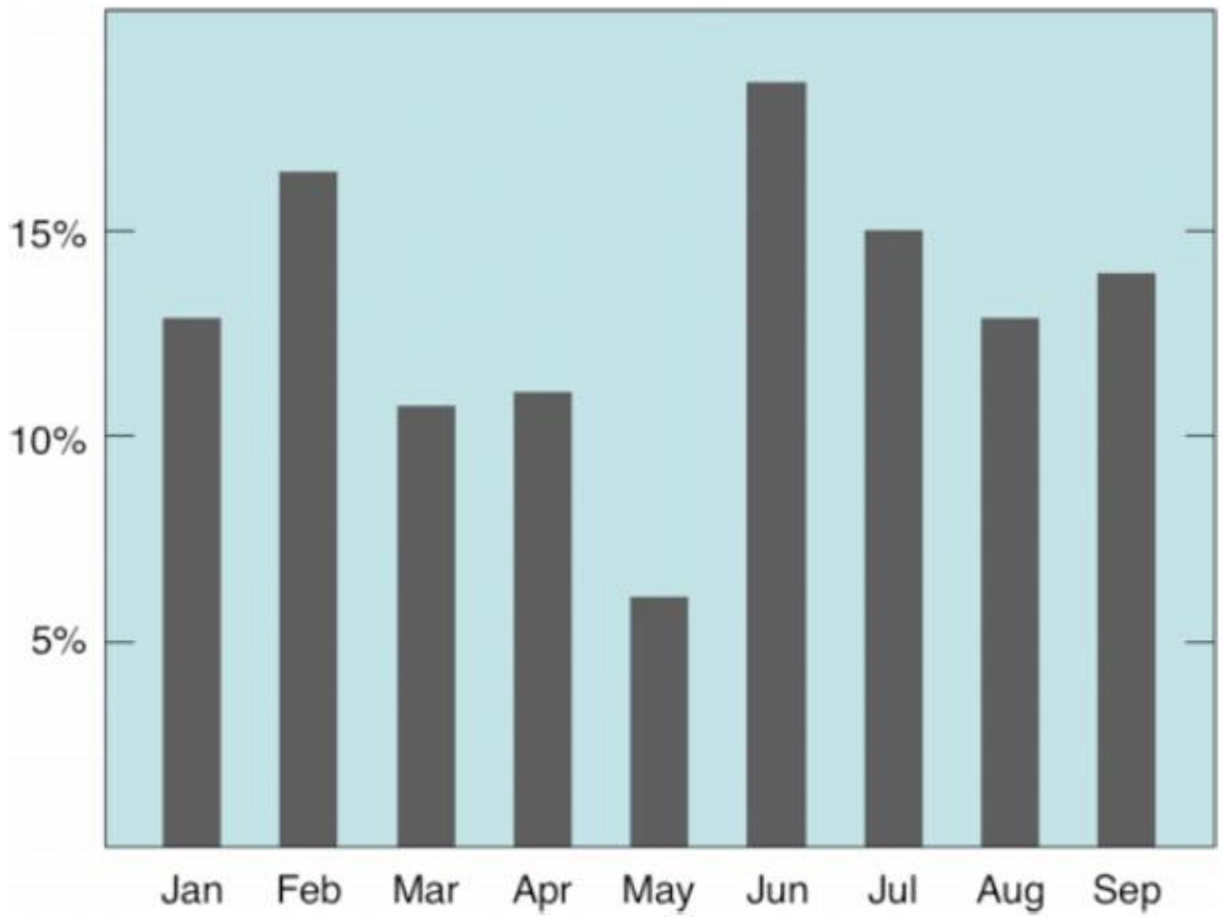
Maximize the data-ink ratio

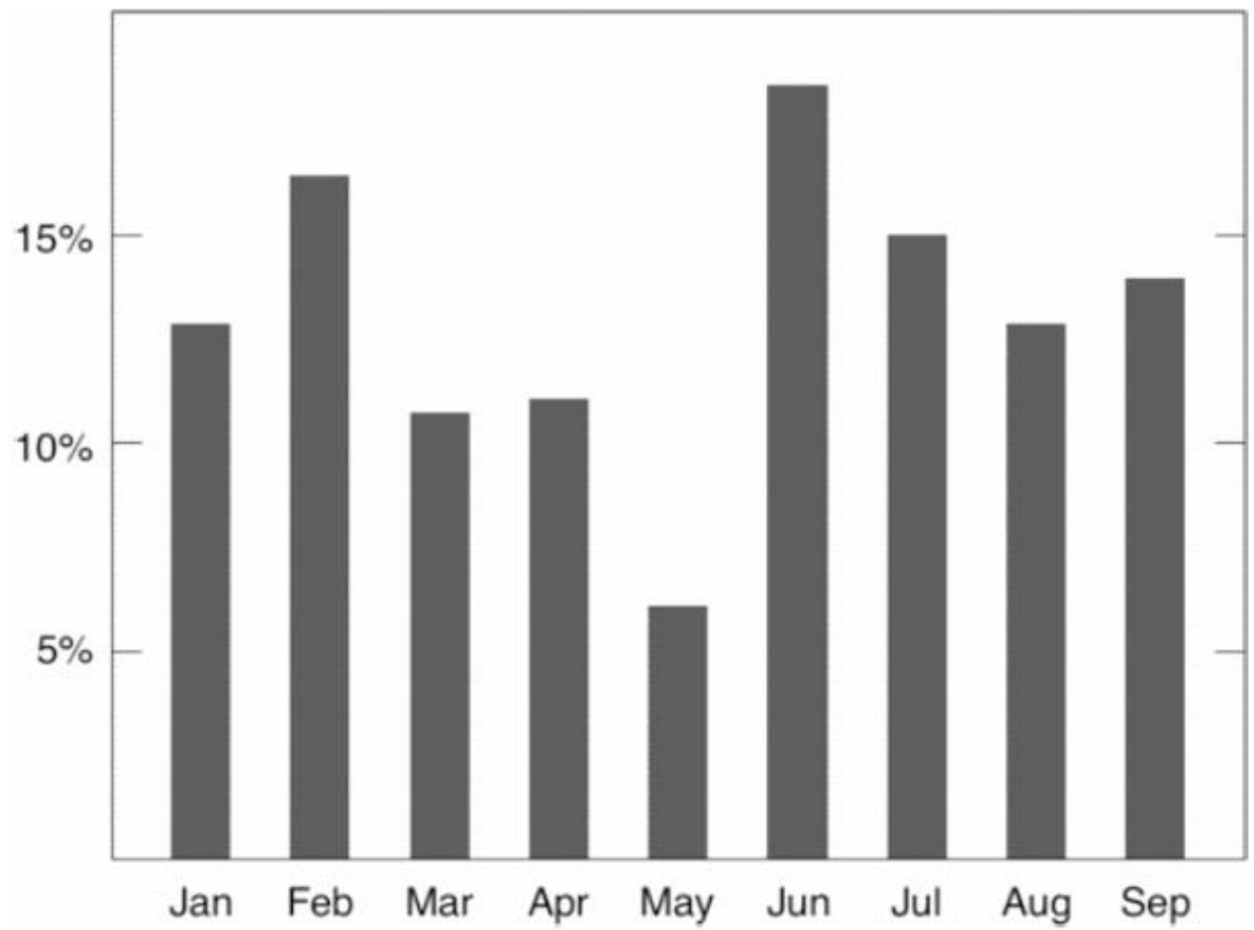
- Within reason
- In essence, you should be able to argue for every pixel

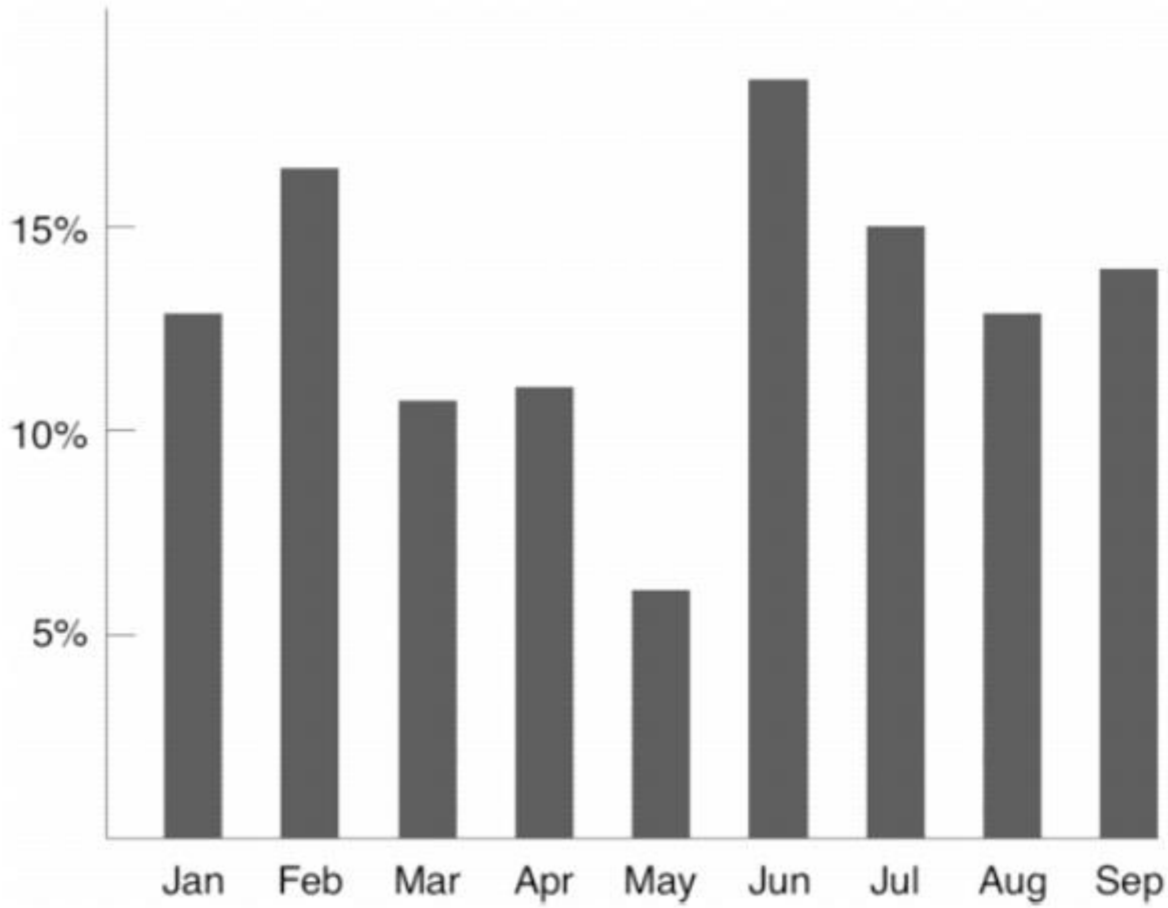
- Starting point:
  - erase non-data ink
  - erase redundant data-ink

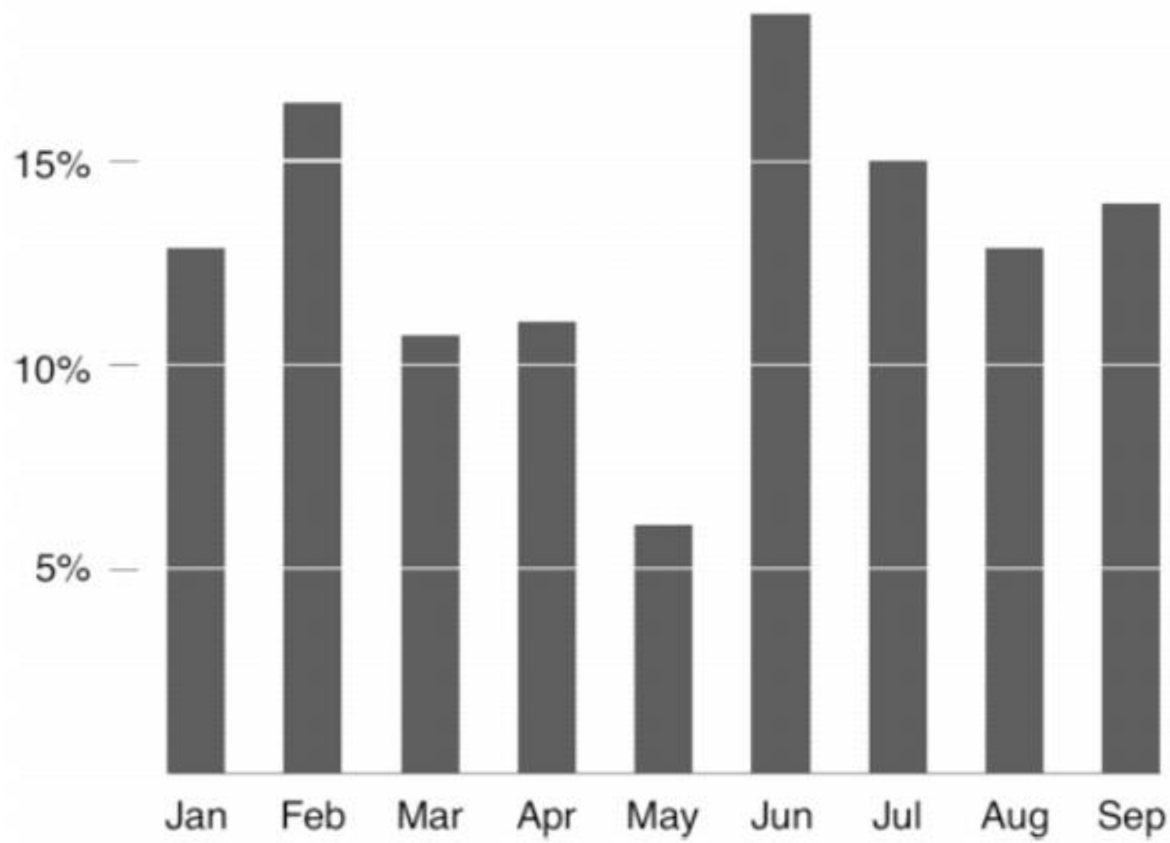


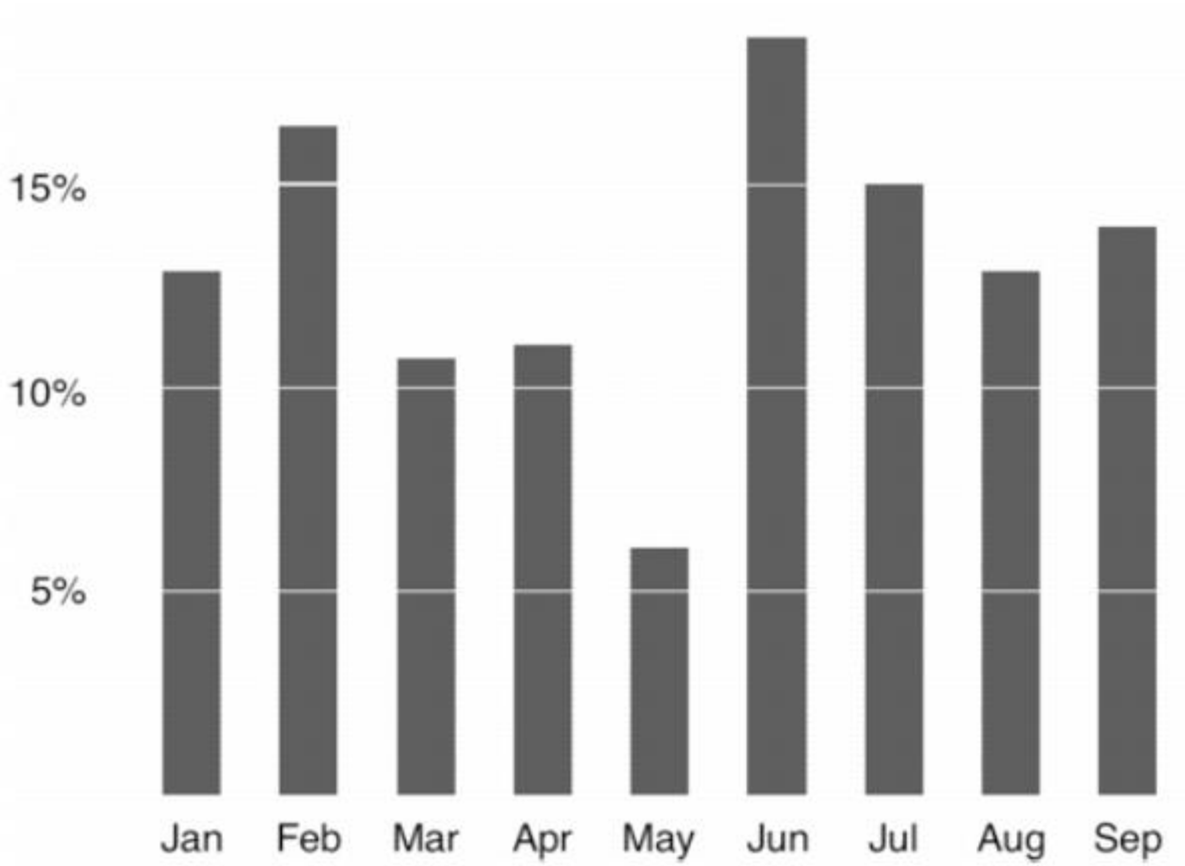
15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep


15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep


15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep


15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep


15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep


15%
10%
5%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep

Summary

- Show data variation, not design variation
- Avoid using ink for non-data items
- Avoid redundancy
- Clear and detailed labeling should be used to defeat graphical distortion
- Revise and Edit

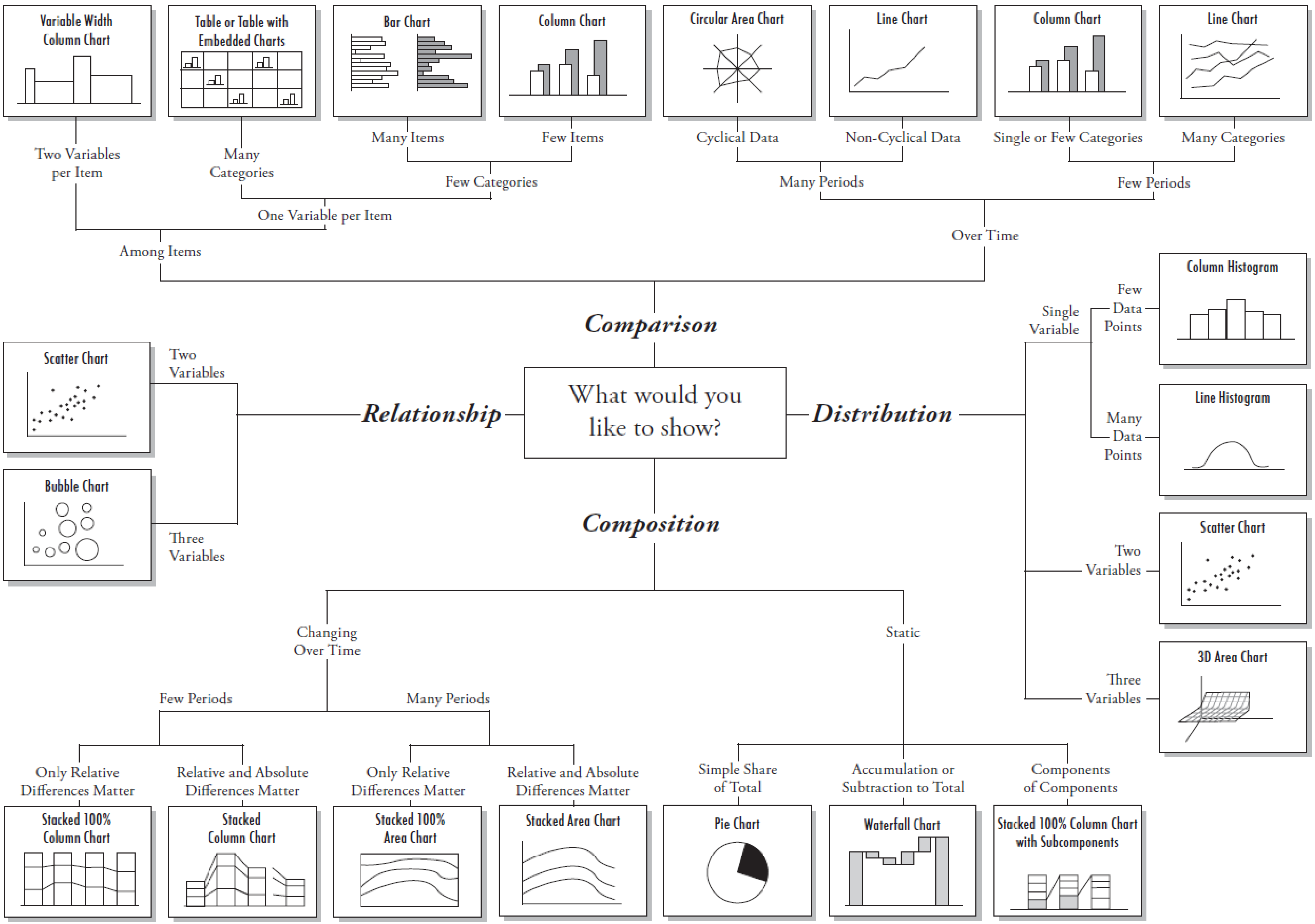
Other Examples

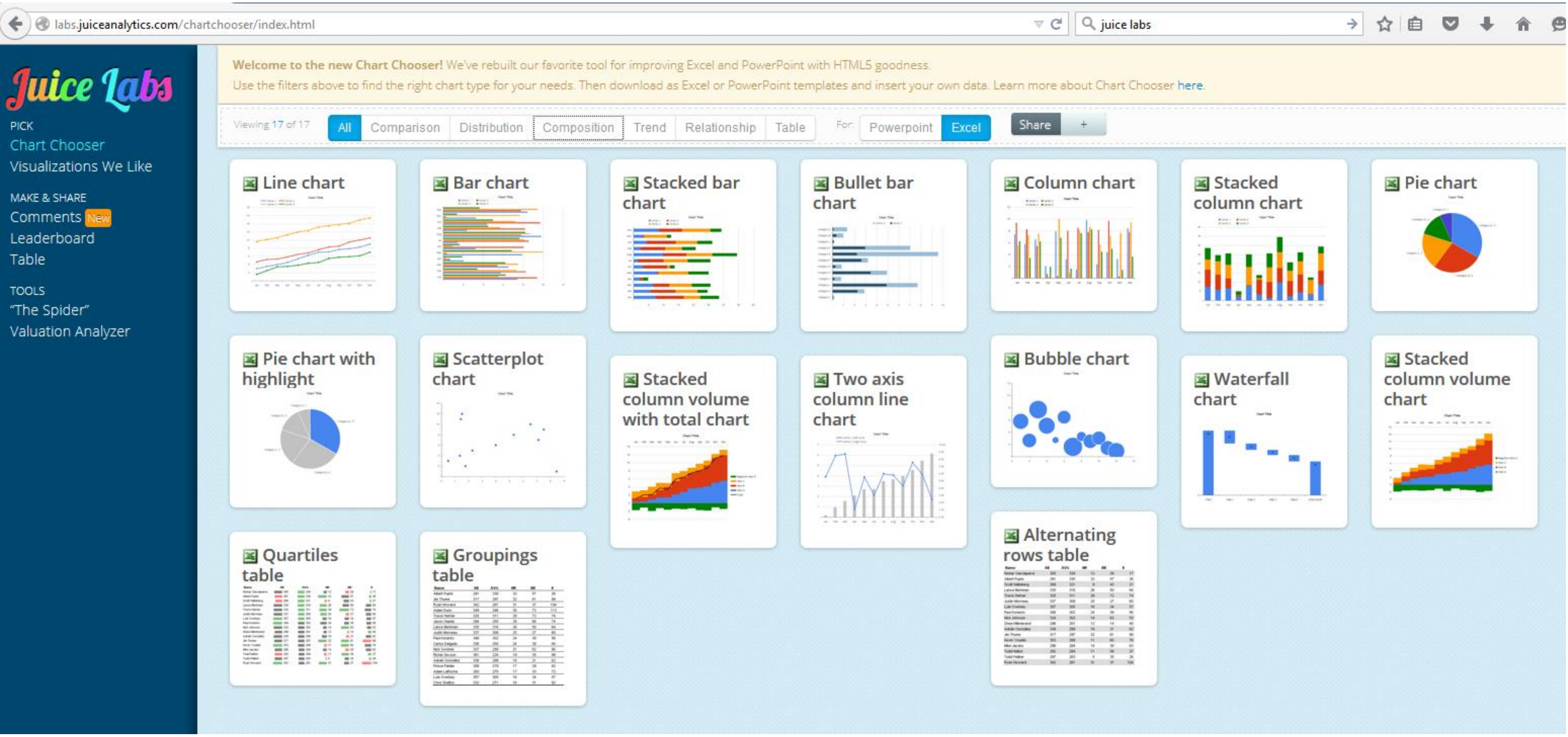
http://www.nytimes.com/interactive/2008/02/23/movies/20080223_REVENUE_GRAPHIC.html http://submarine-cable-map-2013.telegeography.com/

http://www.npr.org/sections/itsallpolitics/2012/11/01/163632378/a-campaign-map-morphed-by-money

Practical Tips

Chart Suggestions—A Thought-Starter




labs.juiceanalytics.com/chartchooser/index.html
juice labs
Juice labs
PICK
Chart Chooser
Visualizations We Like
MAKE & SHARE
Comments New
Leaderboard
Table
TOOLS
"The Spider"
Valuation Analyzer
Welcome to the new Chart Chooser! We've rebuilt our favorite tool for improving Excel and PowerPoint with HTML5 goodness.
Use the filters above to find the right chart type for your needs. Then download as Excel or PowerPoint templates and insert your own data. Learn more about Chart Chooser here.
Viewing 17 of 17
All
Comparison
Distribution
Composition
Trend
Relationship
Table
For:
Powerpoint
Excel
Share
+
Line chart
Bar chart
Stacked bar chart
Bullet bar chart
Column chart
Stacked column chart
Pie chart
Pie chart with highlight
Scatterplot chart
Stacked column volume with total chart
Two axis column line chart
Bubble chart
Waterfall chart
Stacked column volume chart
Quartiles table
Groupings table
Alternating rows table

Tools

- Tableau
  - https://www.tableau.com/solutions/gallery
- Highcharts
  - https://www.highcharts.com/demo
- Matplotlib
  - A library for plotting data in Python
  - https://matplotlib.org/examples/index.html

Data Science: The Context

Ask question: What data needs to be recorded? or collected?

Why? What research question am I going to answer?

What do I want it to look like?

Real World

Humans behaving
Biology
Finance
Internet
Medicine
Sociology
Olympics

→ Raw Data is Collected / Recorded

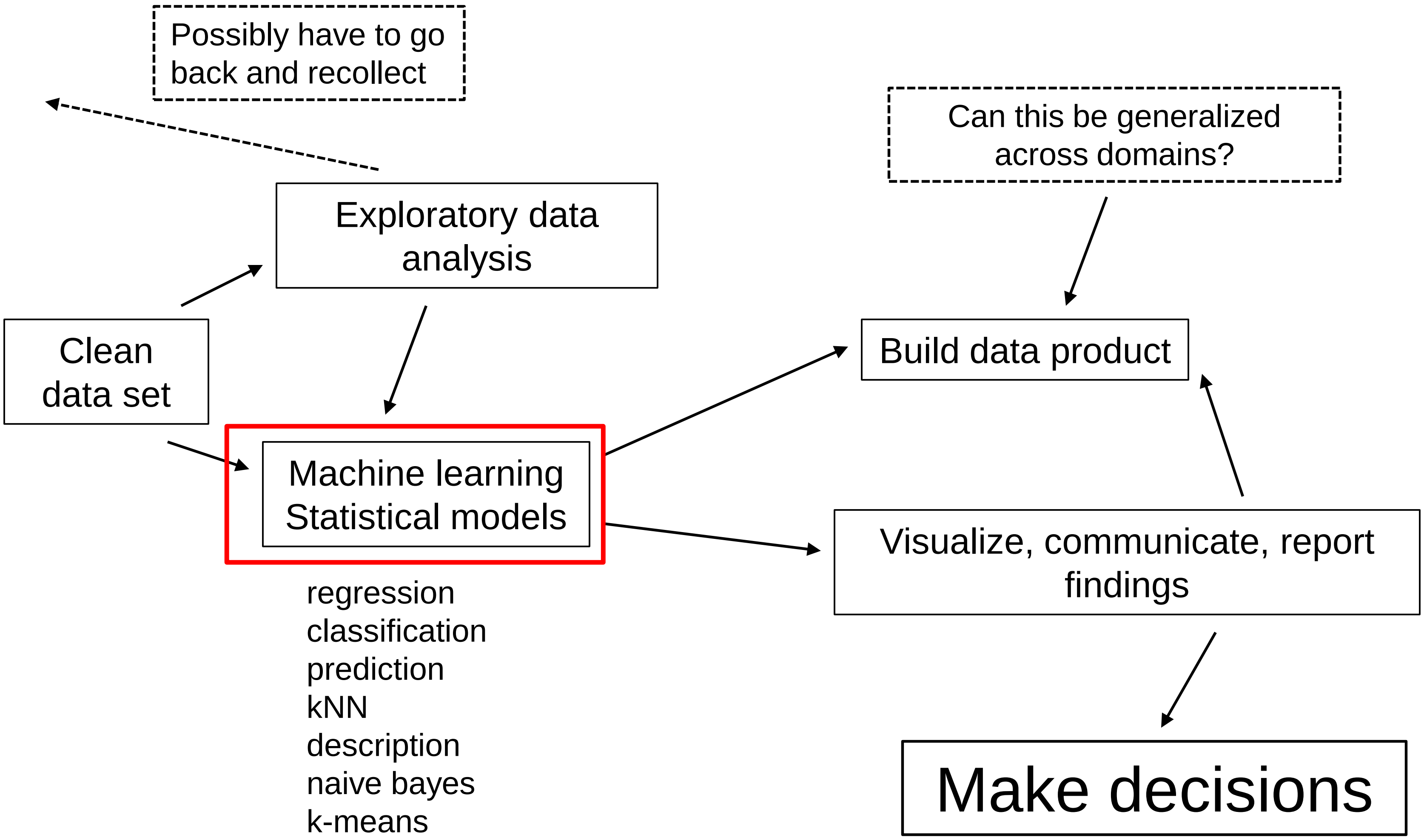
email
logs
medical records
surveys
blood drawn
(microarray)
olympic records
NYT web pages

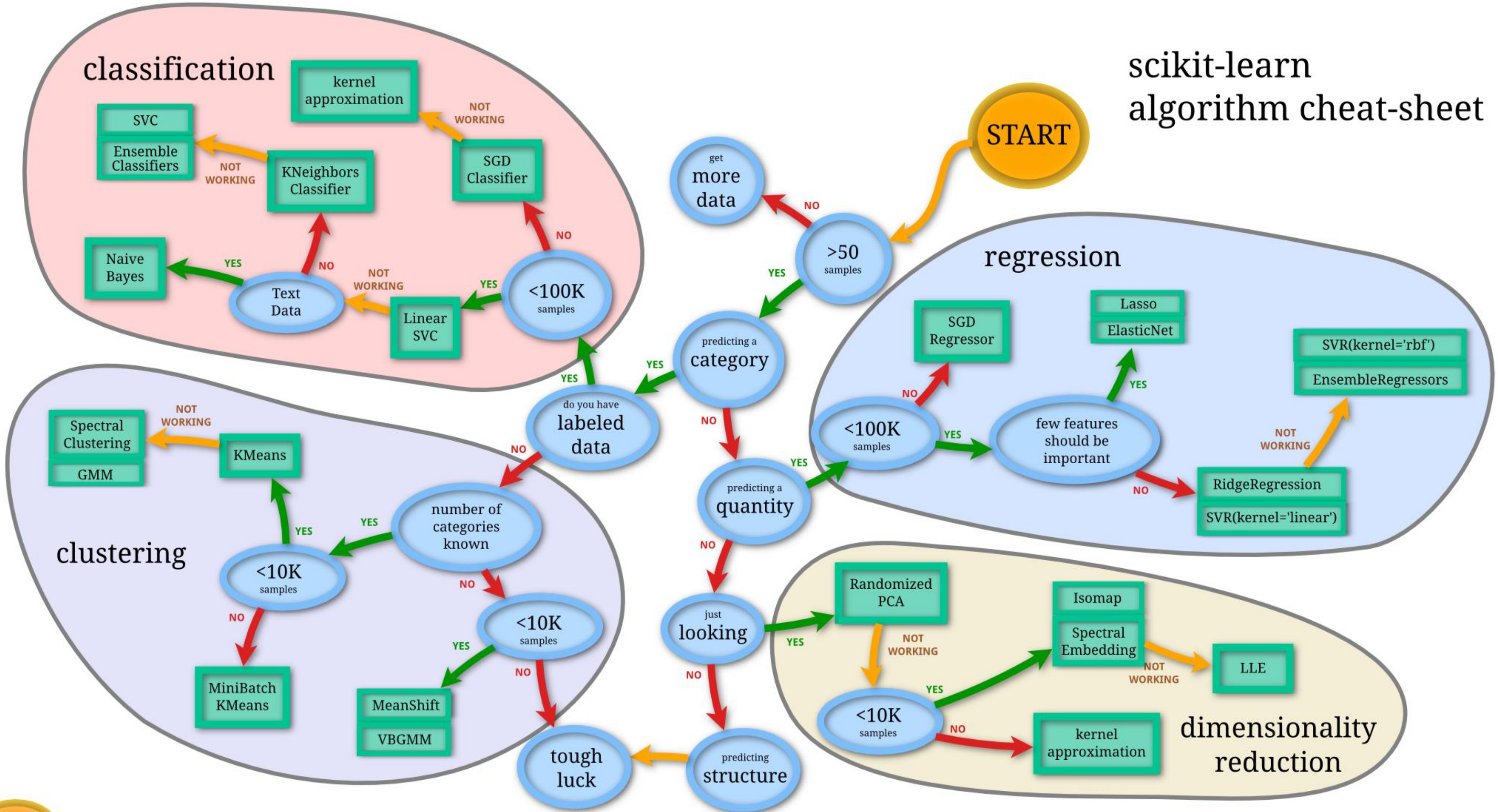
→ Data is Processed

pipelines
web scraping
cleaning
munging
joining
wrangling

→ Data Set

“clean” table


Possibly have to go back and recollect
Exploratory data analysis
Clean data set
Machine learning Statistical models
regression
classification
prediction
kNN
description
naive bayes
k-means
Can this be generalized across domains?
Build data product
Visualize, communicate, report findings
Make decisions



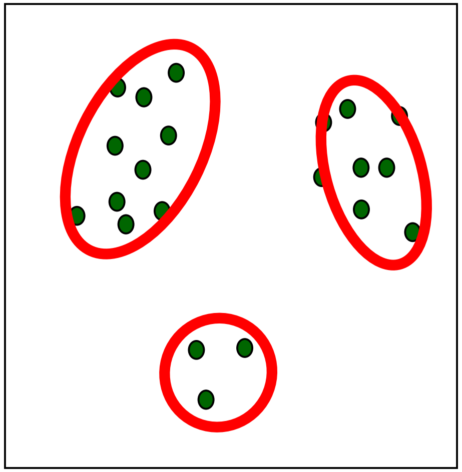
Back

https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Clustering

Unsupervised Learning

- Supervised learning
  - Predict target value (“y”) given features (“x”)
  - E.g., classification and regression



- Unsupervised learning
  - Understand patterns of data (just “x”)
  - Useful for many reasons
    - Data mining (“explain”)
    - Representation (feature generation or selection)
  - E.g., Clustering

High Dimensional Data

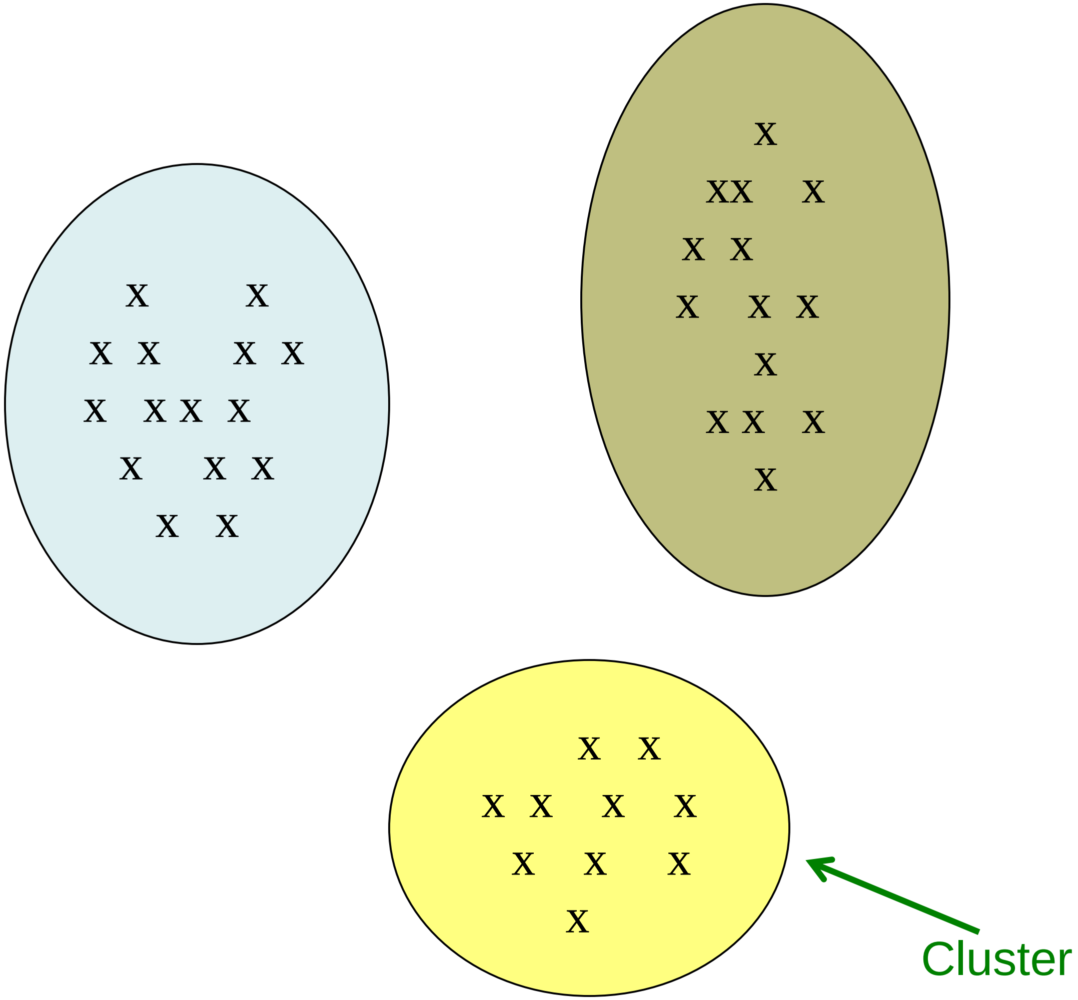
- Given a cloud of data points, we want to understand their structure

Clustering

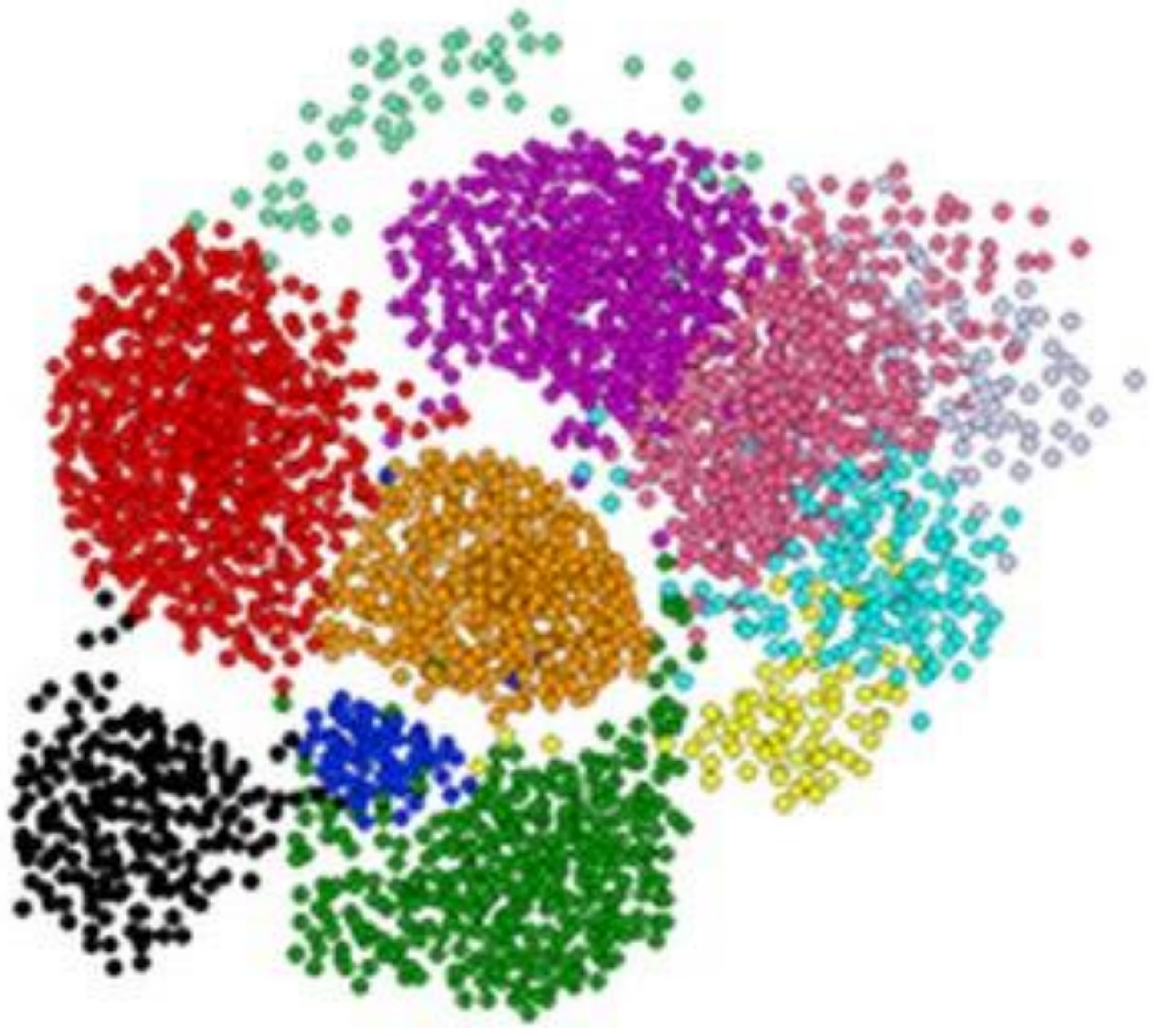
- Given a **set of points**, with a notion of **distance** between points, **group the points** into some number of *clusters*, so that
  - Members of a cluster are close/similar to each other
  - Members of different clusters are dissimilar

- Usually:
  - Points are in a high-dimensional space
  - Similarity is defined using a distance measure
    - Euclidean, Cosine, Jaccard, edit distance, …

Example Clusters



Clustering is Hard!



Why is it hard?

- Clustering in two dimensions looks easy
- Clustering small amounts of data looks easy
- And in most cases, looks are not deceiving

- Many applications involve not 2, but 10 or 10,000 dimensions
- **High-dimensional spaces look different:** Almost all pairs of points are at about the same distance

- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Example: Clustering Songs

- Intuitively: Music divides into categories, and customers prefer a few categories
  - But what are categories really?
- Represent a song by a set of customers who downloaded it
- Similar songs have similar sets of downloaders, and vice-versa

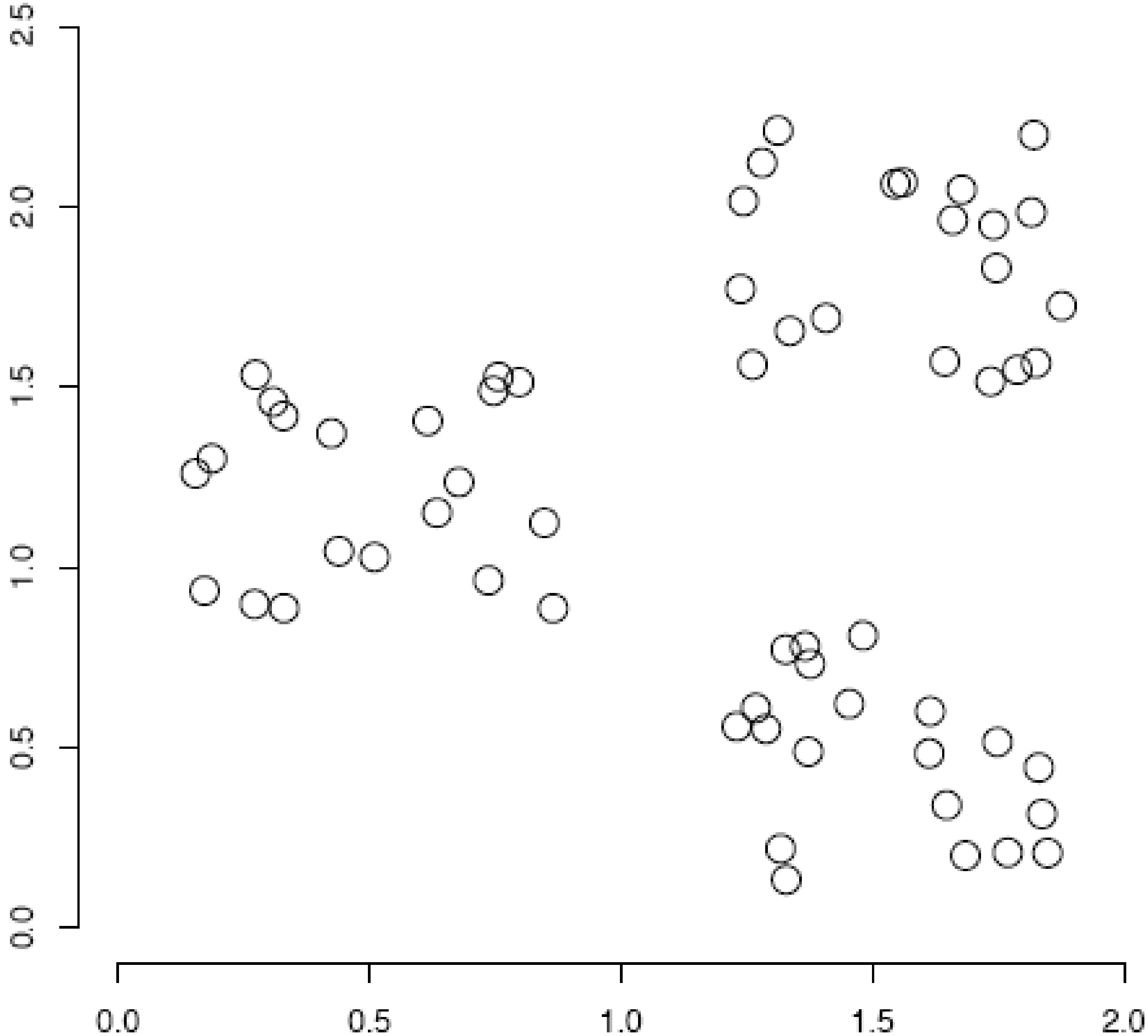
- **Goal: Find clusters of similar songs**

Challenge

- To cluster songs:
  - How do we define the problem?
  - How do we tackle it?
- Hint: Represent a song by a set of customers who downloaded it

- k-means
- k-medoids
- Naive Bayes
- EM clustering (probabilistic)

K-means



0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
1.5
2.0

K-means (in one slide!)

Input is **k** (the number of clusters), **data points** in Euclidean space

0. Initialize clusters by picking one point per cluster

Loop:

1. Place each point in the cluster whose current centroid is nearest
2. Find the new centroid for each cluster

K-means

- Objective/partitioning criterion: minimize the average squared difference from the centroid
- Assumes documents are real-valued vectors
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, ω:

$$\vec{\mu}(\omega) = \frac{1}{|\omega|} \sum_{\vec{x} \in \omega} \vec{x}$$

- We try to find the minimum average squared difference by iterating two steps:
  - reassignment: assign each vector to its closest centroid
  - recomputation: recompute each centroid as the average of the vectors that were assigned to it in reassignment

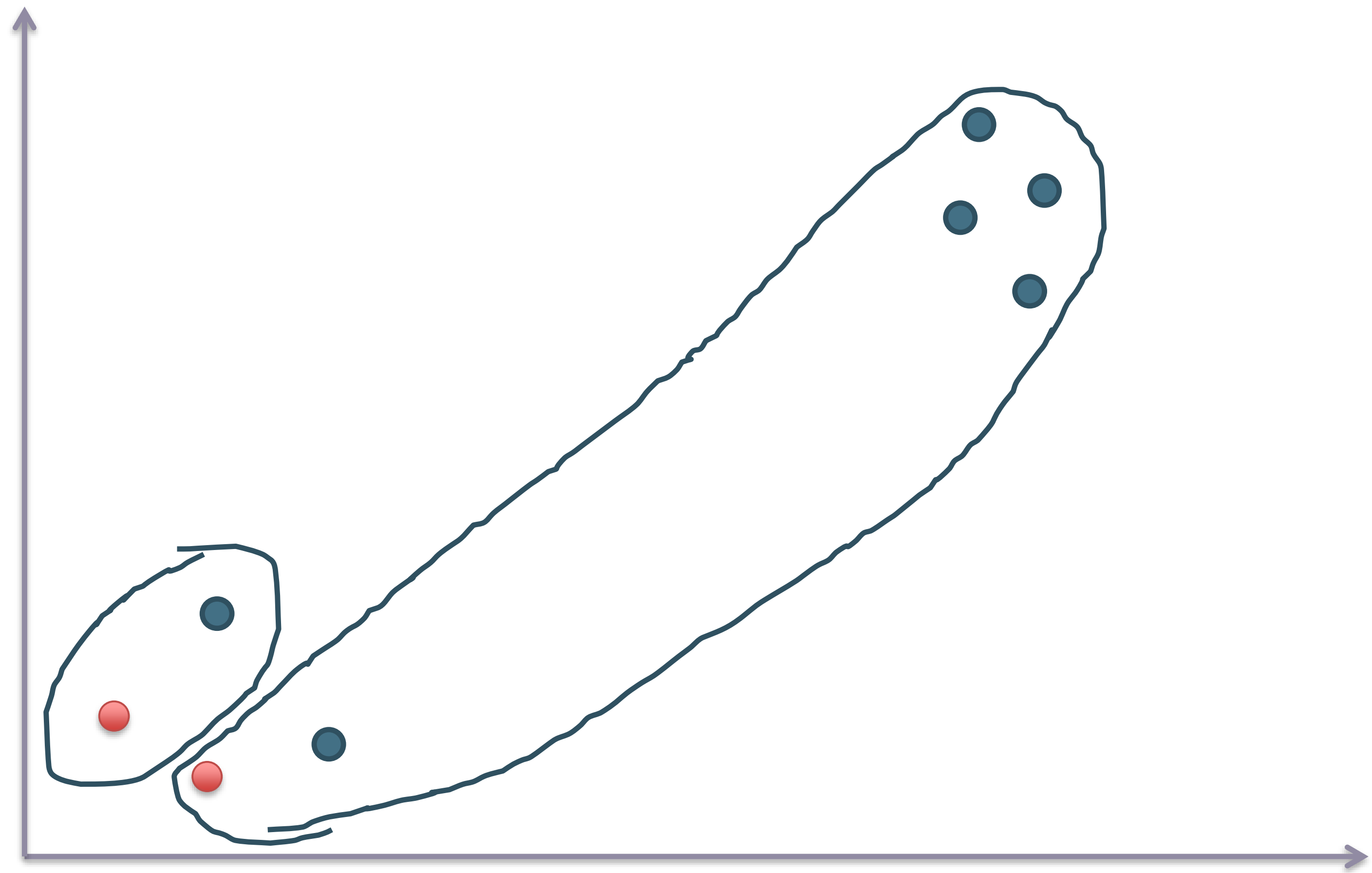
K-MEANS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$

$(\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_K) \leftarrow$ SELECTRANDOMSEEDS$(\{\vec{x}_1, \ldots, \vec{x}_N\}, K)$
for $k \leftarrow 1$ **to** K
do $\vec{\mu}_k \leftarrow \vec{s}_k$
while stopping criterion has not been met
do for $k \leftarrow 1$ **to** K
    do $\omega_k \leftarrow \{\}$
    for $n \leftarrow 1$ **to** N
    do $j \leftarrow \arg\min_{j'} |\vec{\mu}_{j'} - \vec{x}_n|$
        $\omega_j \leftarrow \omega_j \cup \{\vec{x}_n\}$ *(reassignment of vectors)*
    for $k \leftarrow 1$ **to** K
    do $\vec{\mu}_k \leftarrow \frac{1}{|\omega_k|} \sum_{\vec{x} \in \omega_k} \vec{x}$ *(recomputation of centroids)*
return $\{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$

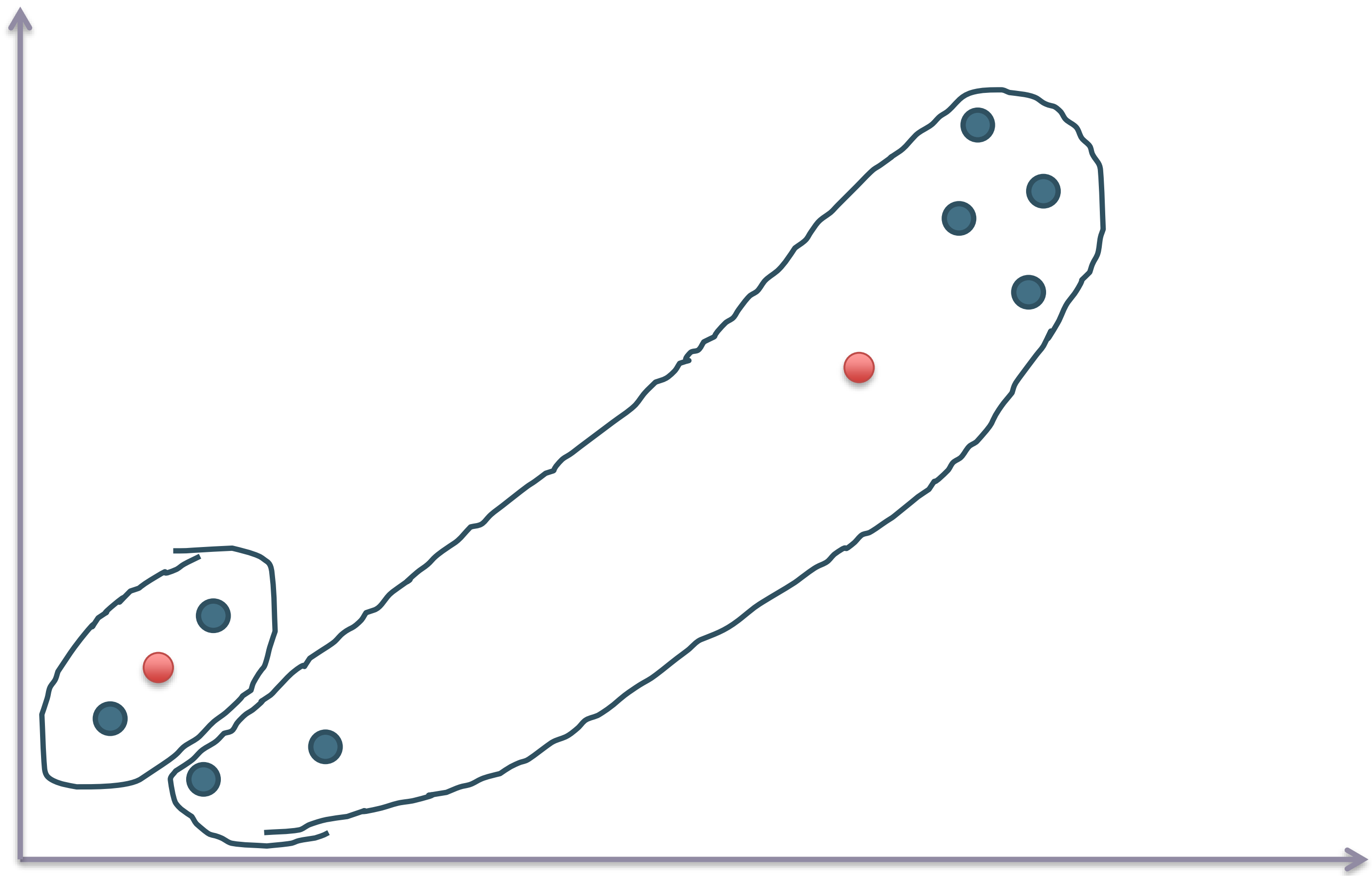
K-Means Clustering Example

K-Means Clustering Example

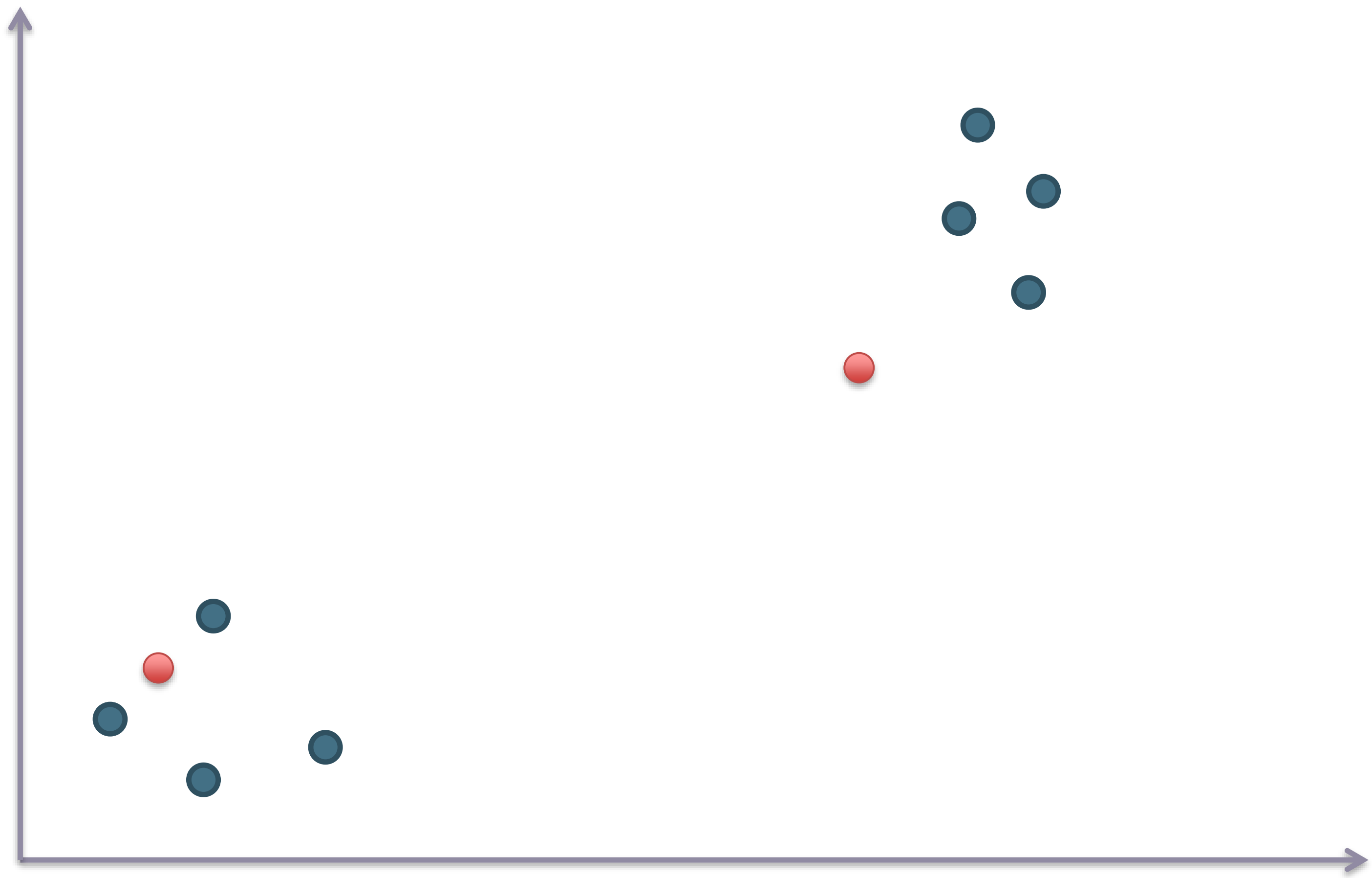
K-Means Clustering Example



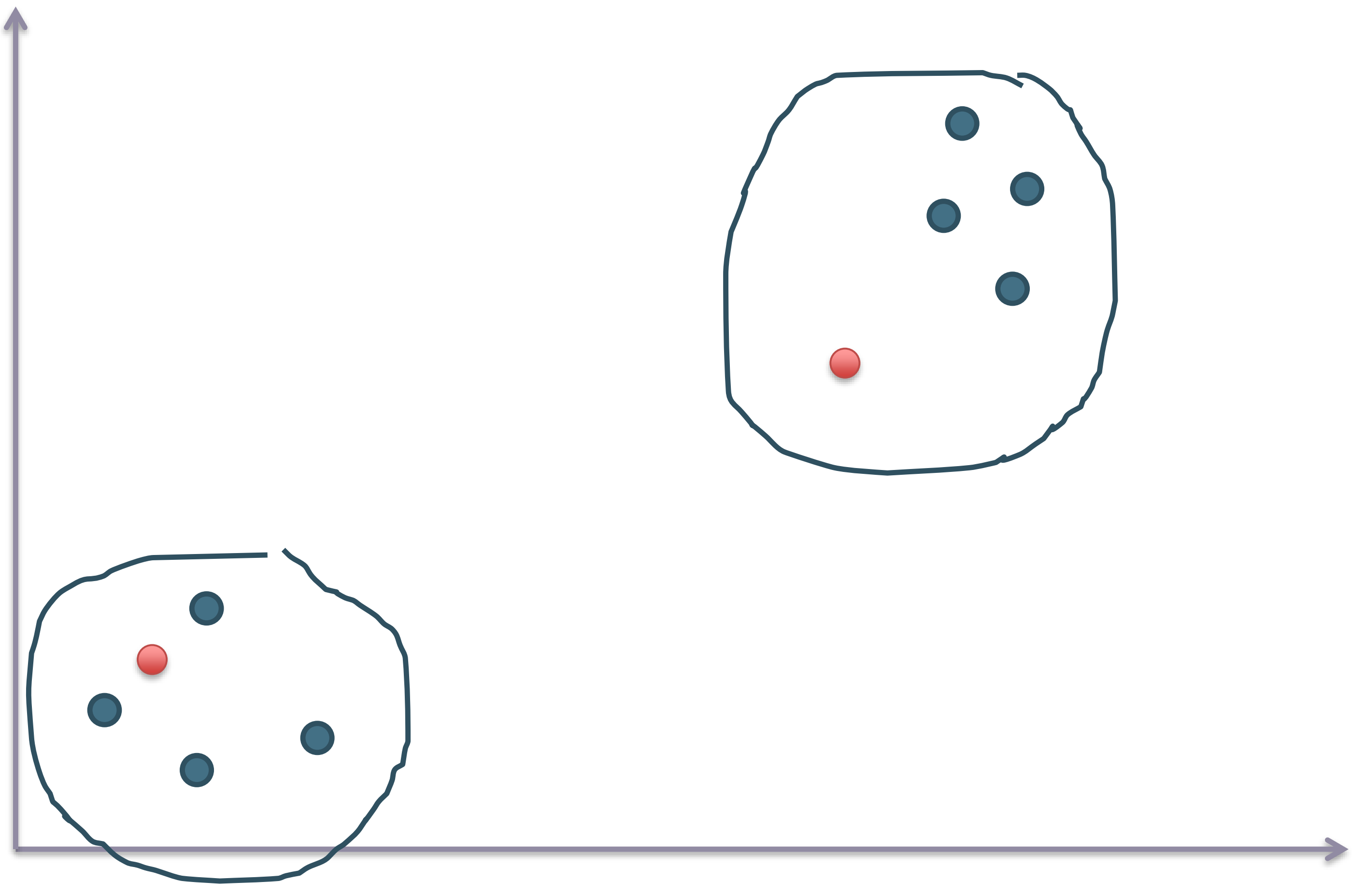
K-Means Clustering Example



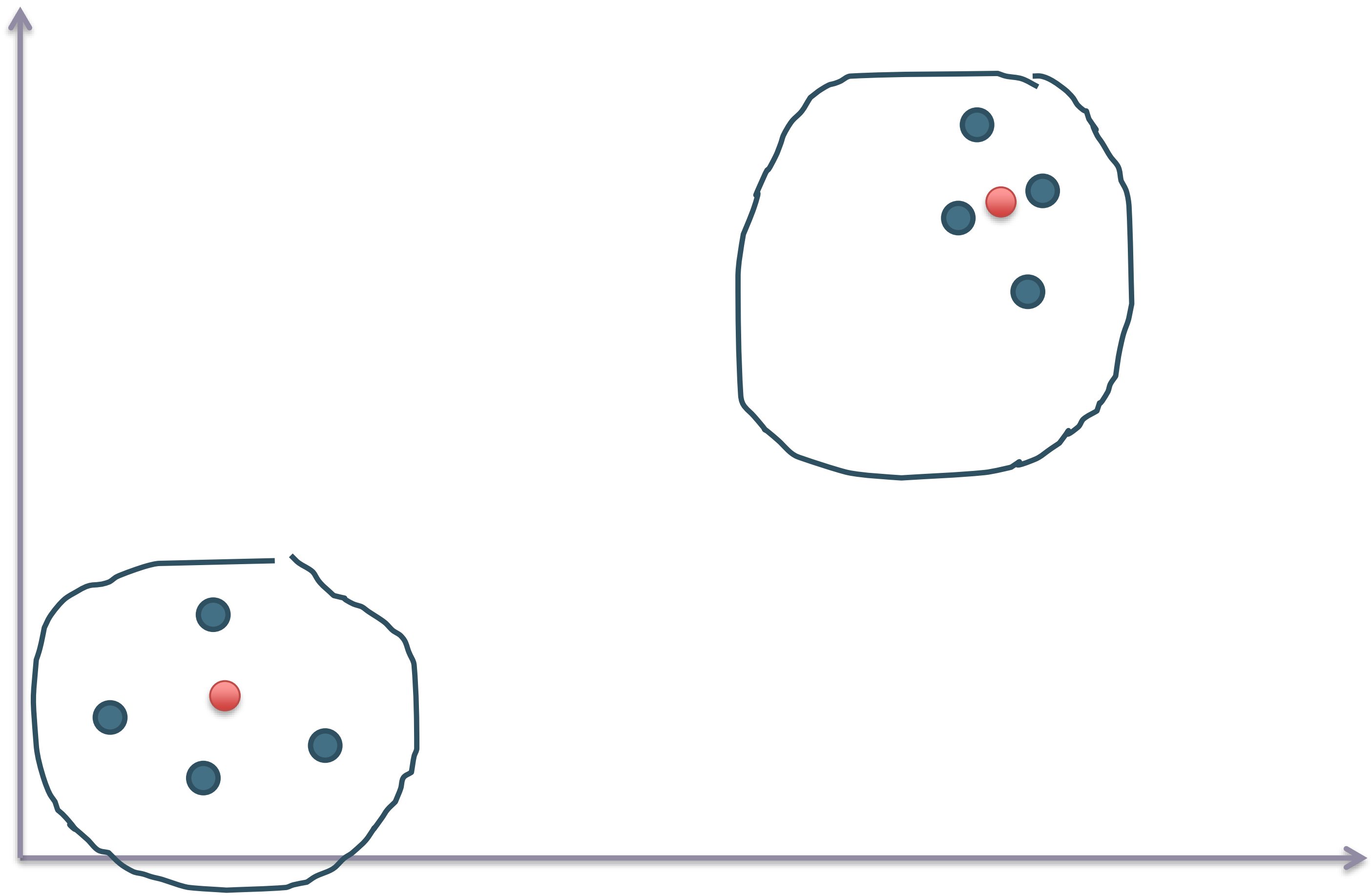
K-Means Clustering Example



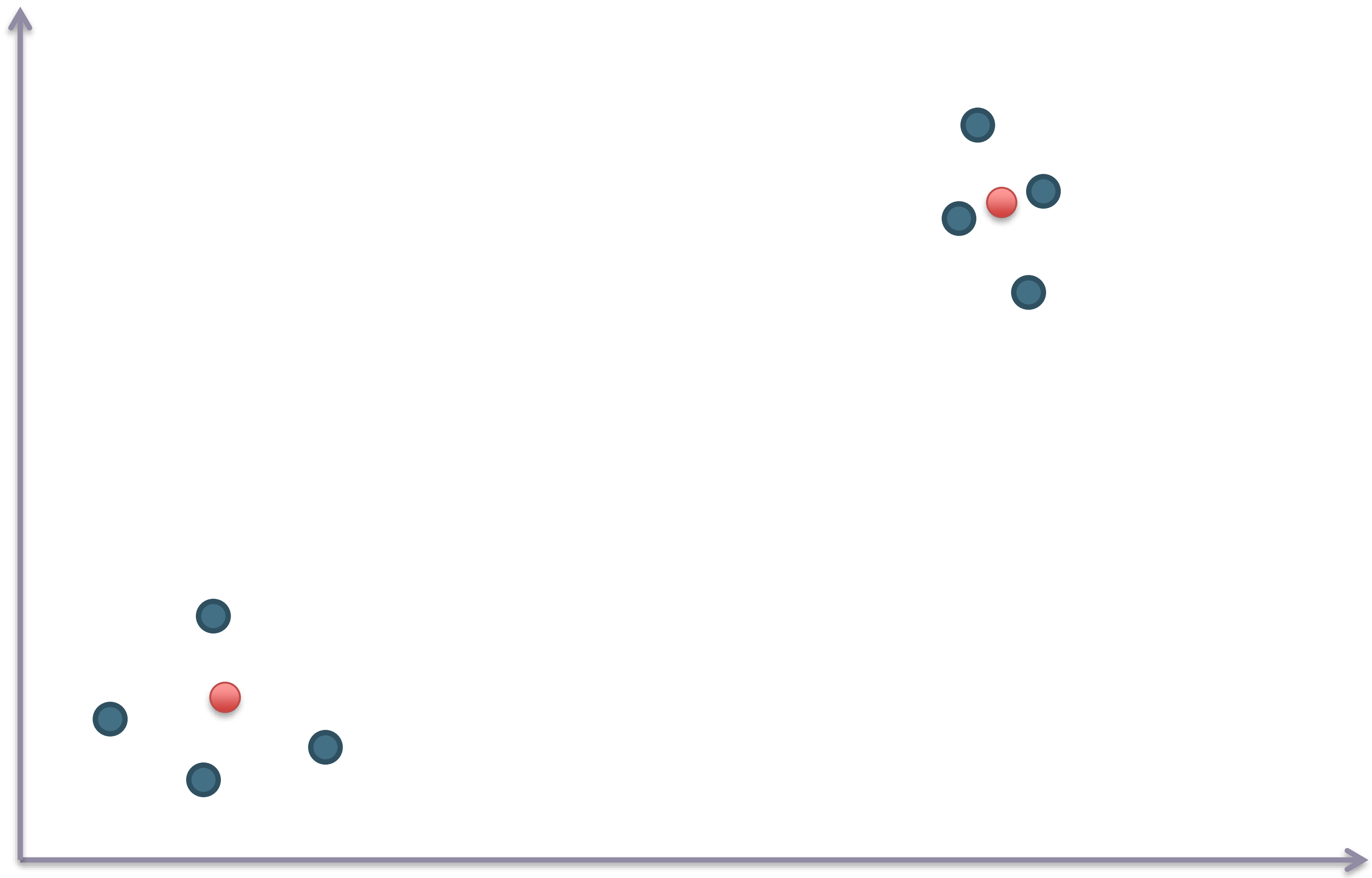
K-Means Clustering Example



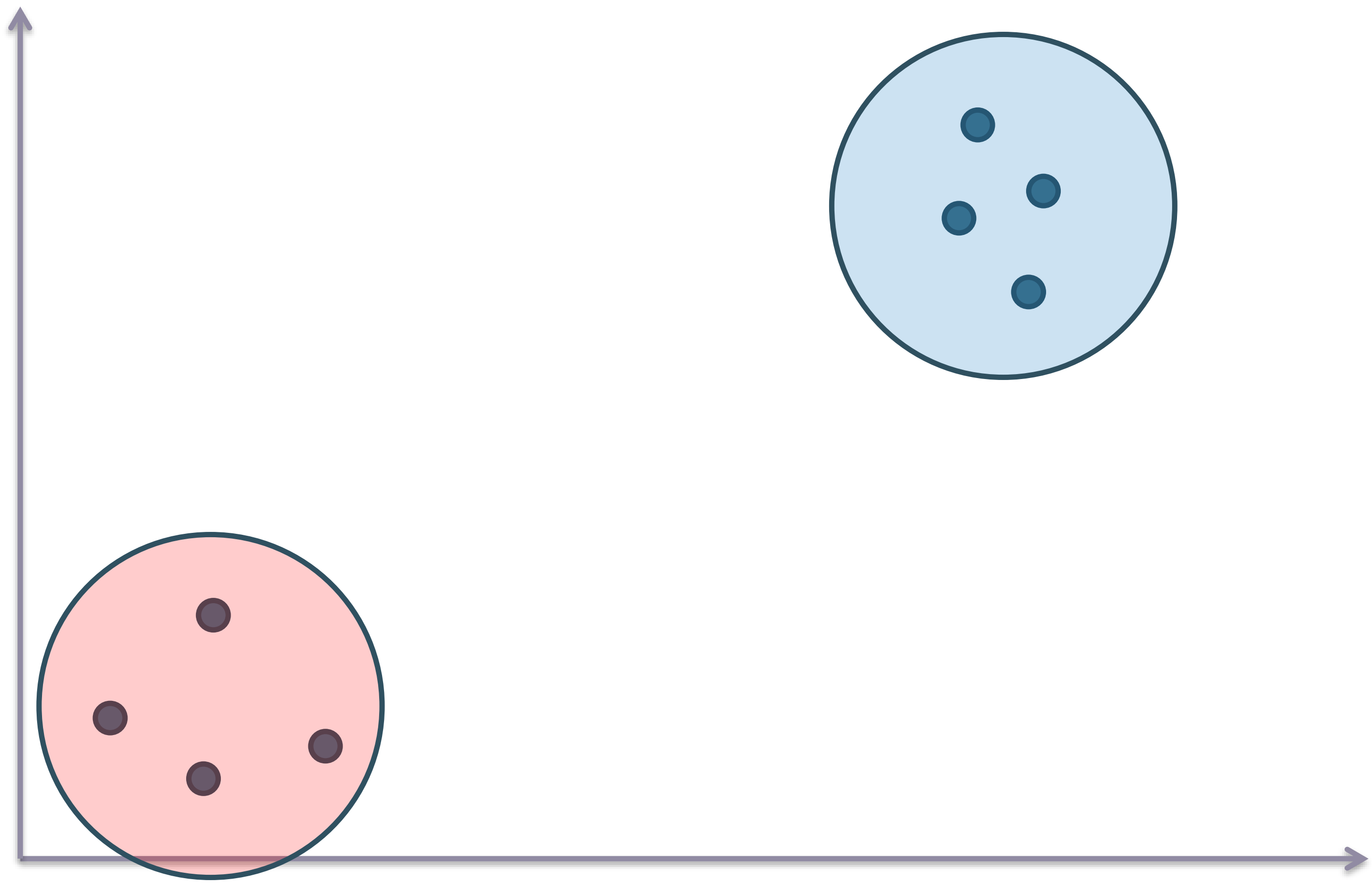
K-Means Clustering Example



K-Means Clustering Example

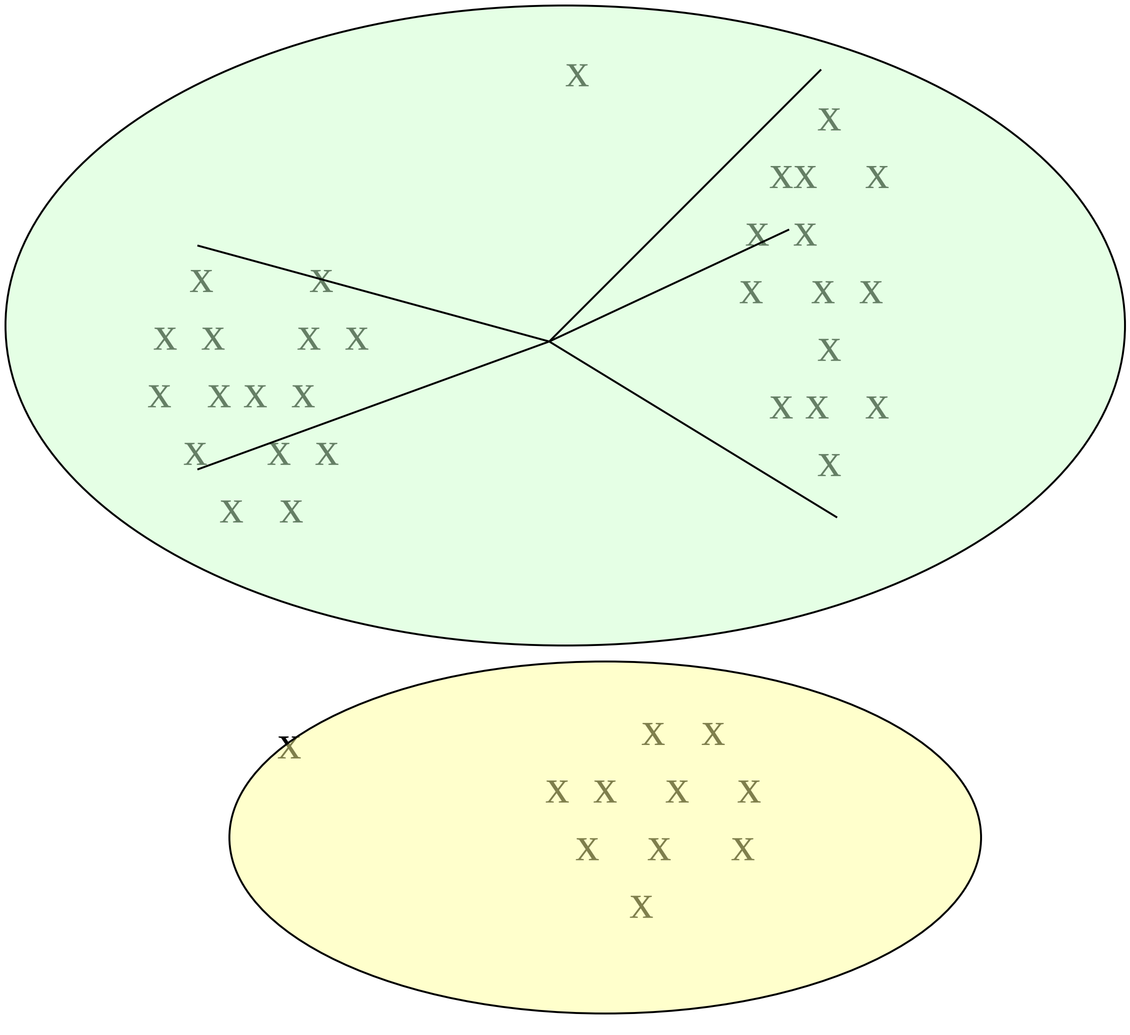


K-Means Clustering Example



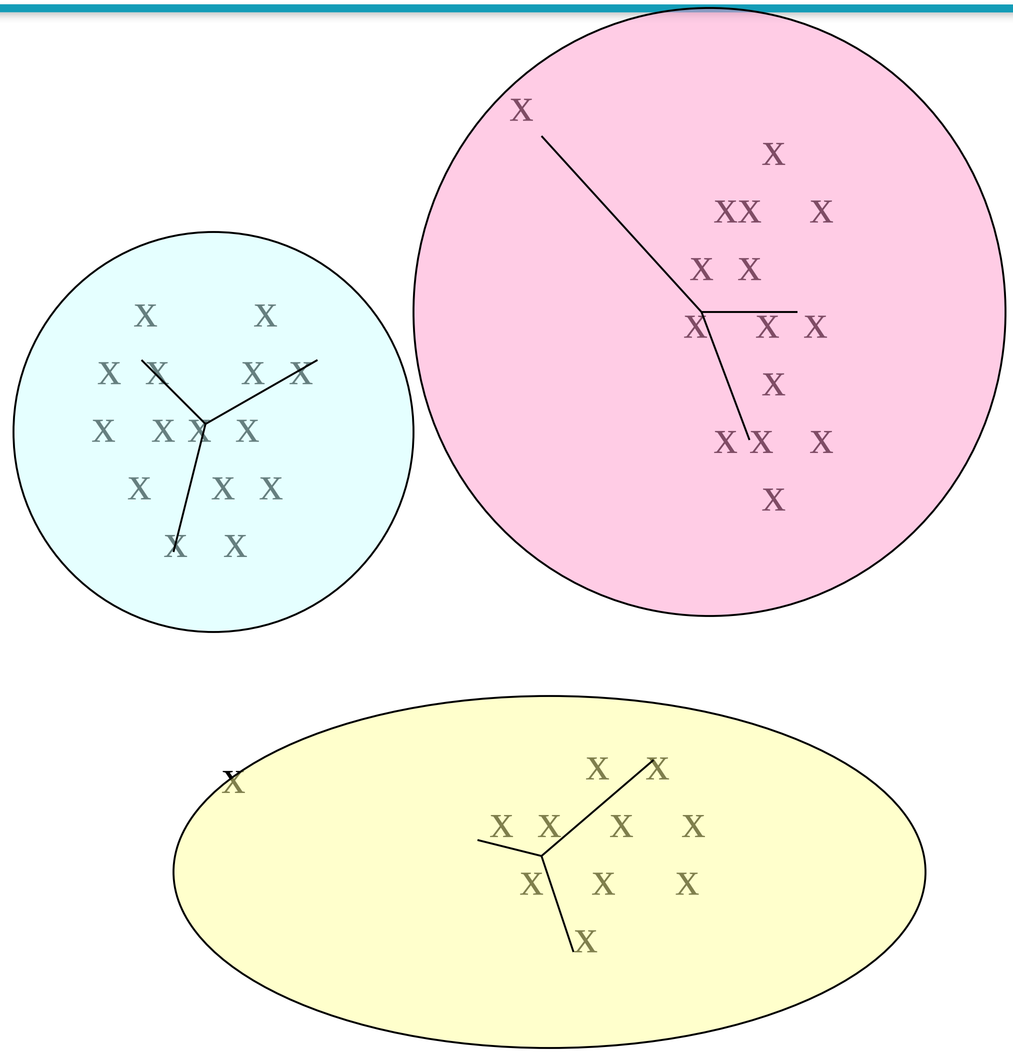
Example: Picking *k*

Too few;
many long
distances
to centroid.



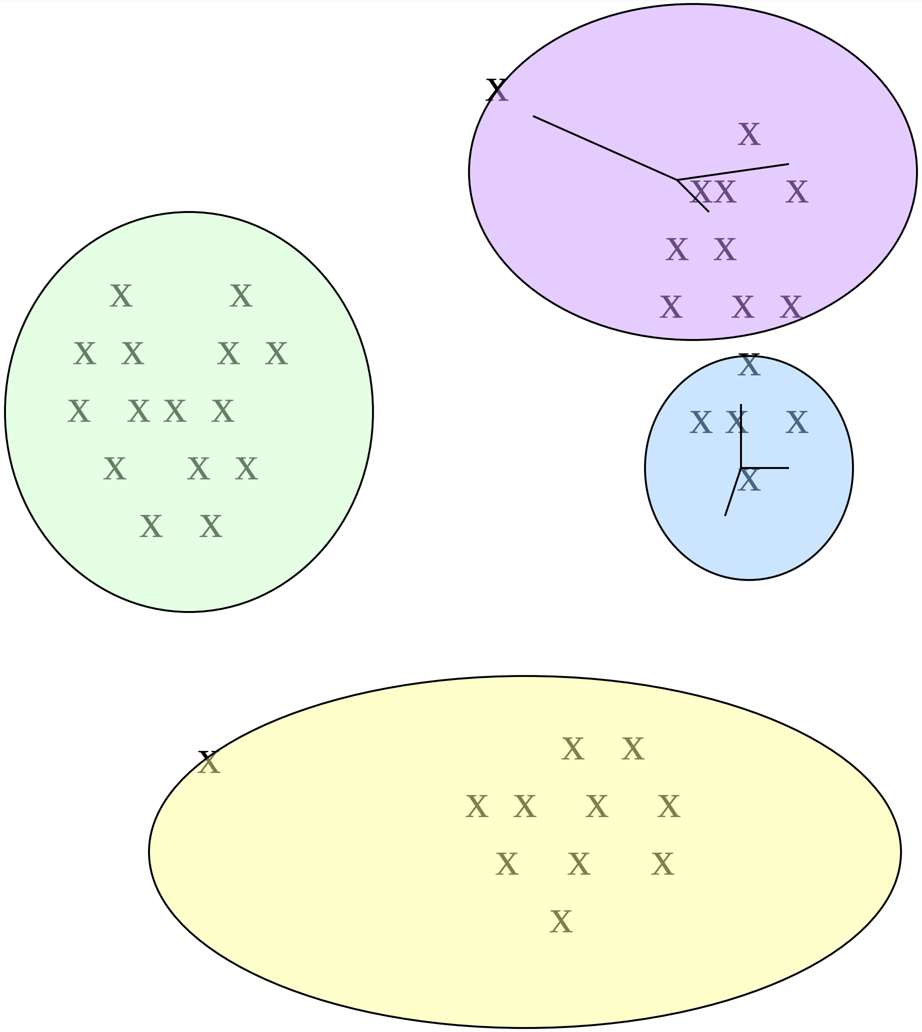
Example: Picking *k*

Just right;
distances
rather short.



Example: Picking *k*

Too many;
little improvement
in average
distance.



Convergence of K Means

- K-means converges to a fixed point in a finite number of iterations.
- Proof:
  - The sum of squared distances (RSS) decreases during reassignment.
  - (because each vector is moved to a closer centroid)
  - RSS decreases during recomputation.
  - There is only a finite number of clusterings
  - Thus: We must reach a fixed point.
- But we don’t know how long convergence will take!
- If we don’t care about a few docs switching back and forth, then convergence is usually fast (< 10-20 iterations).

Recomputation decreases average distance

- RSS = residual sum of squares (the "goodness" measure G)

$$\mathrm{RSS}_k(\vec{v}) = \sum_{\vec{x}\in\omega_k} \|\vec{v}-\vec{x}\|^2 = \sum_{\vec{x}\in\omega_k} \sum_{m=1}^{M} (v_m - x_m)^2$$

$$\frac{\partial \mathrm{RSS}_k(\vec{v})}{\partial v_m} = \sum_{\vec{x}\in\omega_k} 2(v_m - x_m) = 0$$

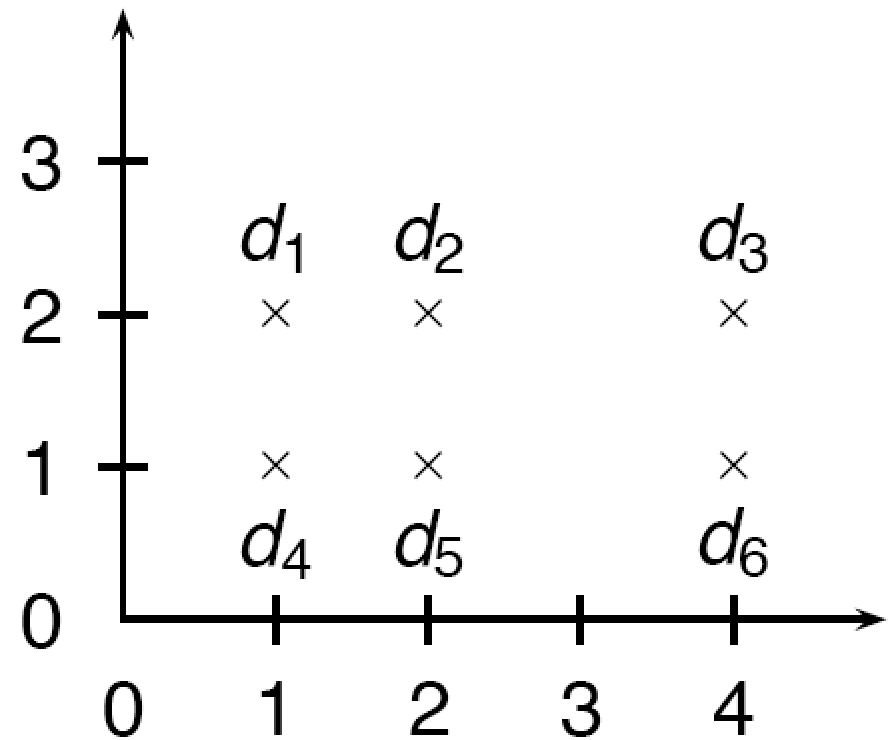
$$v_m = \frac{1}{|\omega_k|} \sum_{\vec{x}\in\omega_k} x_m$$

- The last line is the componentwise definition of the centroid! We minimize RSSk when the old centroid is replaced with the new centroid. RSS, the sum of the RSSk , must then also decrease during recomputation.

Optimality of *K*-means

- Convergence does not mean that we converge to the optimal clustering!
- This is the great weakness of K-means.
- If we start with a bad set of seeds, the resulting clustering can be horrible.

Example of suboptimal clustering!!!!



- What is the optimal clustering for K=2?
- What happens when our seeds are: d2, d5?

Initialization of *K*-means

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points
  - Initialize with the results of another method.

How many clusters?

Hmm…

- **Either: Number of clusters K is given.**
  - Then partition into K clusters
  - K might be given because there is some external constraint. Example: You cannot show more than 10–20 clusters on a screen.
- **Or: Finding the “right” number of clusters is part of the problem.**
  - Given docs, find K for which an optimum is reached.
  - How to define “optimum”?
  - Why can’t we use RSS or average distance from centroid?

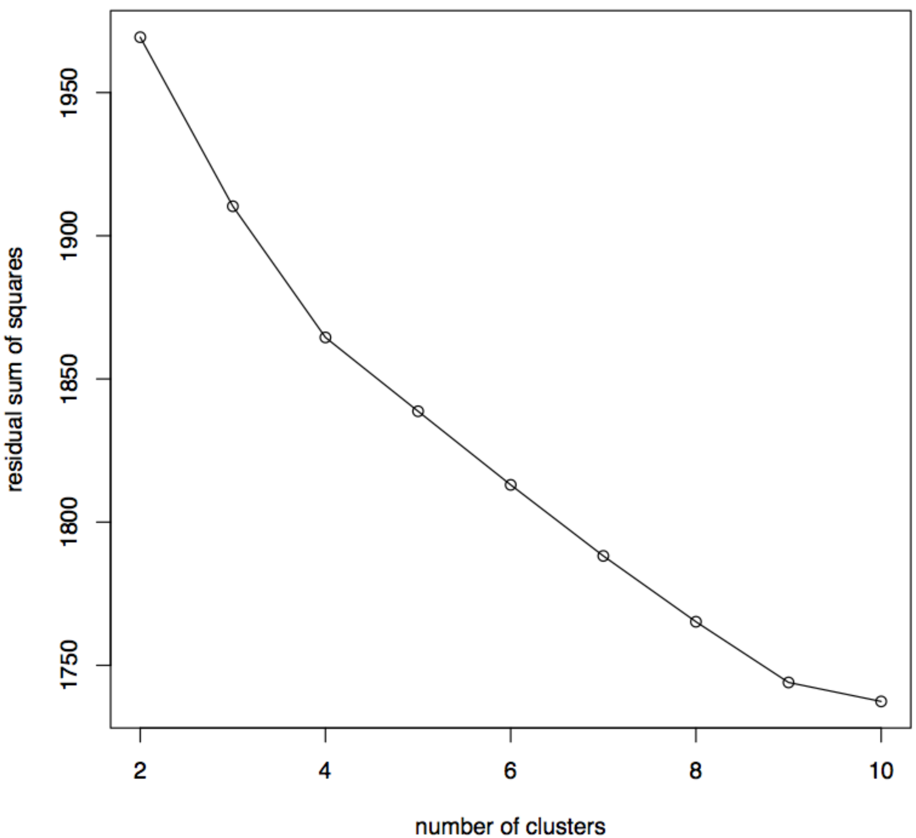
Simple objective function for *K*

- Basic idea:
  - Start with 1 cluster ($K = 1$)
  - Keep adding clusters (= keep increasing K)
  - Add a penalty for each new cluster
- Trade off cluster penalties against average squared distance from centroid
- Choose the value of K with the best tradeoff

Simple objective function for *K*

- Given a clustering, define the cost for a document as (squared) distance to centroid
- Define total distortion RSS(K) as sum of all individual document costs (corresponds to average distance)
- Then: penalize each cluster with a cost λ
- Thus for a clustering with *K* clusters, total cluster penalty is *K*λ
- Define the total cost of a clustering as distortion plus total cluster penalty: RSS(K) + *K*λ
- Select K that minimizes (RSS(K) + *K*λ)
- Still need to determine good value for λ . . .

Finding the “knee” in the curve



Pick the number of clusters where curve “flattens”. Here: 4 or 9.

What Is the Problem of the K-Means Method?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data

- K-Medoids: instead of finding a **centroid** for each cluster, use one of the points (the **medoid**) as the cluster "center"