Exploratory Data Analysis and Visualization

4. Visualization and Data processing

Li Xinke

DS

City University of Hong Kong

How do we get from data to visualization?

We need to understand:

- Properties of the data
- Properties of the image
- The rules mapping data to image

How many 3's?

248721840123874092165901476098560932472091256290650985265904827582985680960986309584390564095878950374509284750989475092984

How many 3's?

This is because color is **pre-attentively processed**.

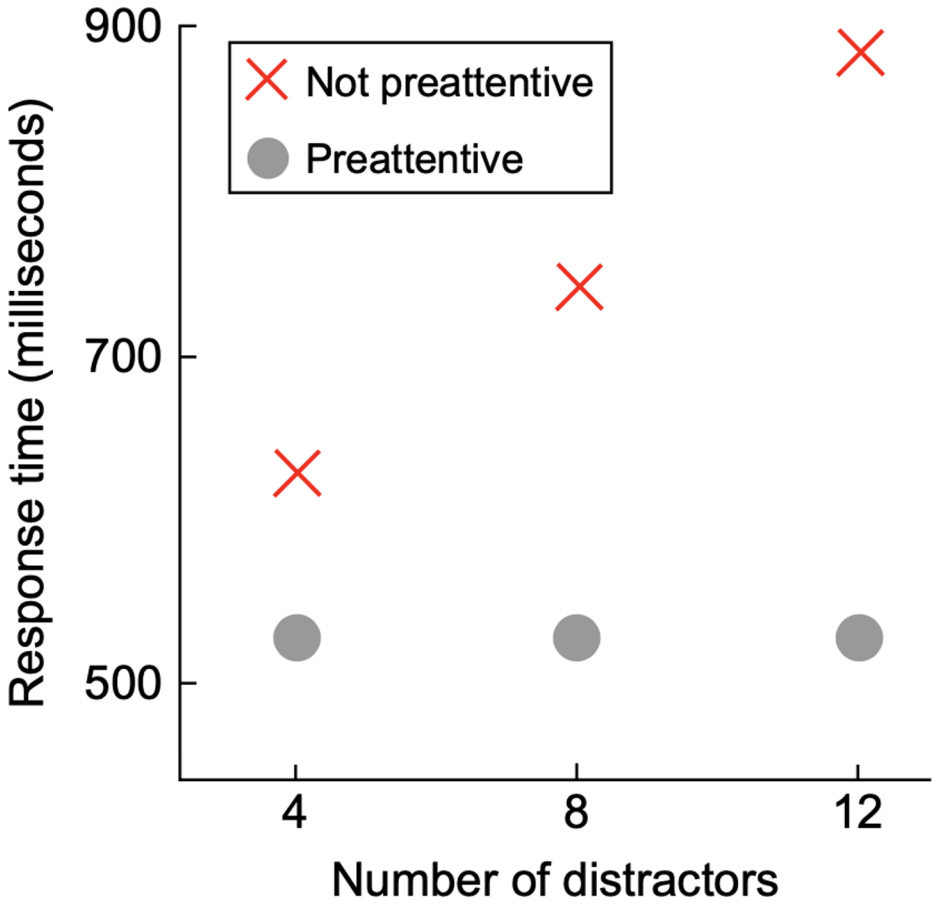
Visual perception and data visualization

- Pre-attentive processing: the subconscious accumulation of information from the environment

- A human can distinguish **differences in line length, shape, orientation, and color (hue)** readily without significant processing effort

- Effective graphics take advantage of pre-attentive processing attributes and the relative strength of these attributes

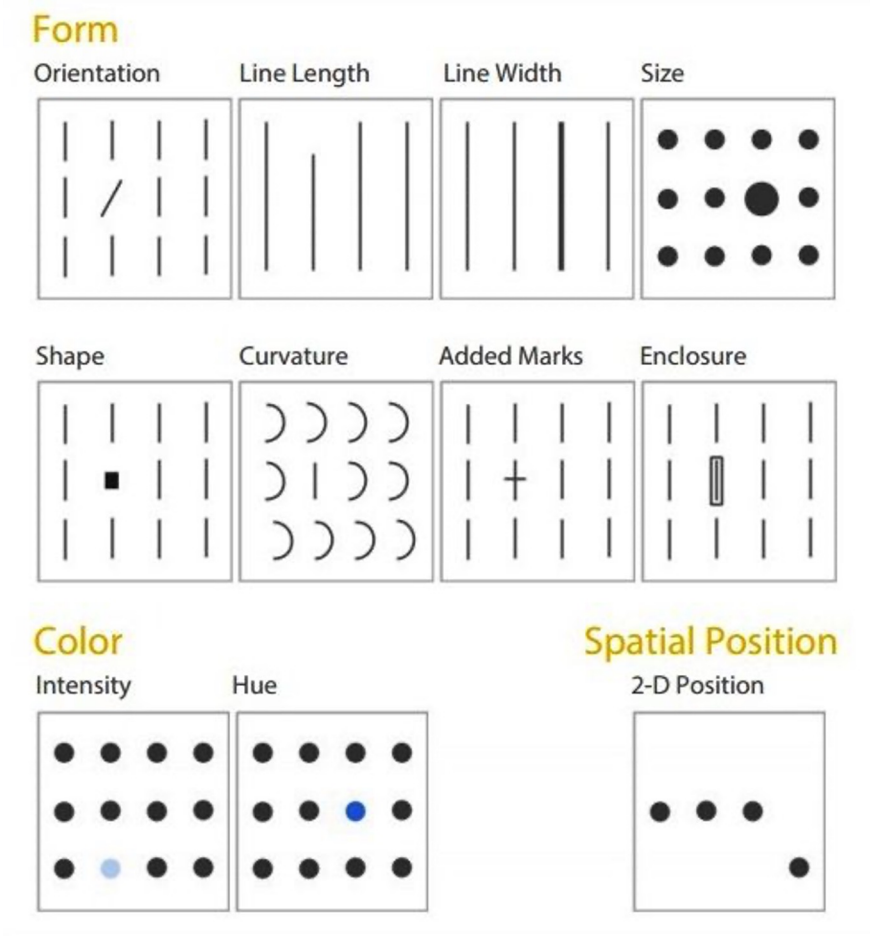
Pre-attentive Processing



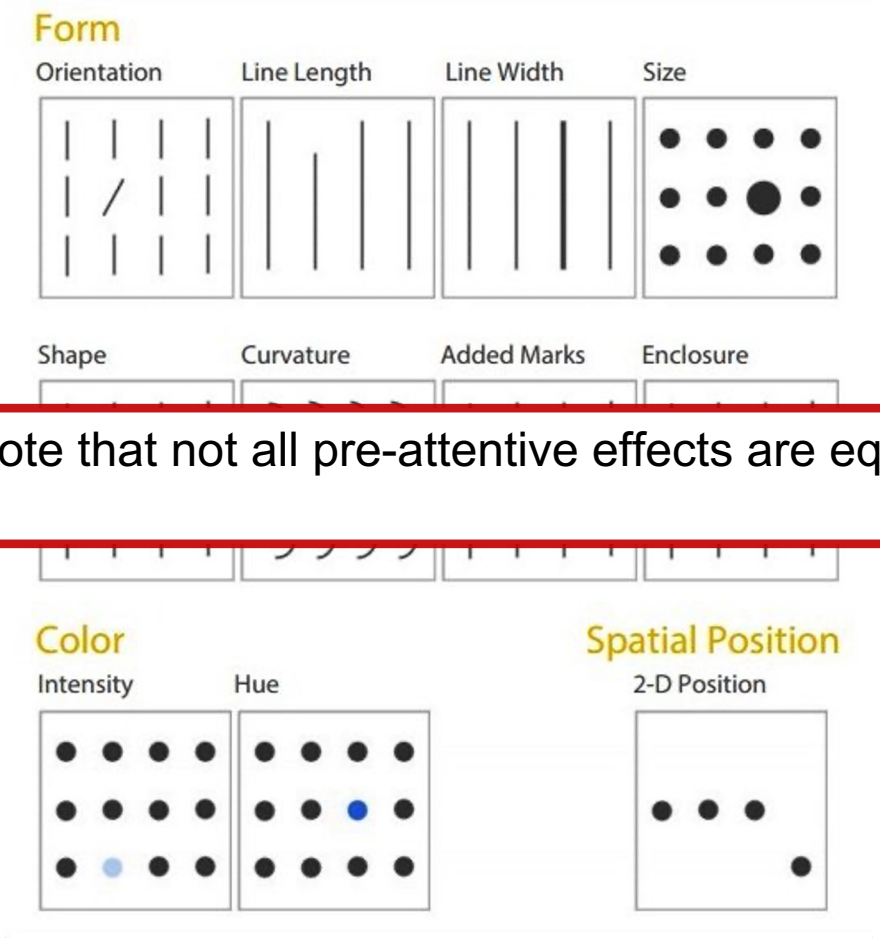
Pre-attentive Processing

- When properties of symbolic data are mapped to **visual properties**, humans can browse through large amounts of data efficiently.
- It is estimated that 2/3 of the brain's neurons can be involved in visual processing.
- Used for
  - Target detection
  - boundary detection
  - counting / estimation
  - …

Pre-attentive Attributes



Pre-attentive Attributes



It is also important to note that not all pre-attentive effects are equally strong.

What's wrong with Google's new design

- https://twitter.com/cesifoti/status/1326157710426050560?s=21

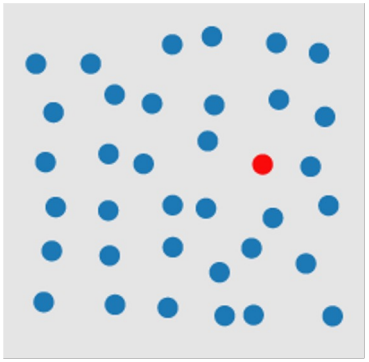
What's wrong with Google's new design

- https://twitter.com/cesifoti/status/1326157710426050560?s=21

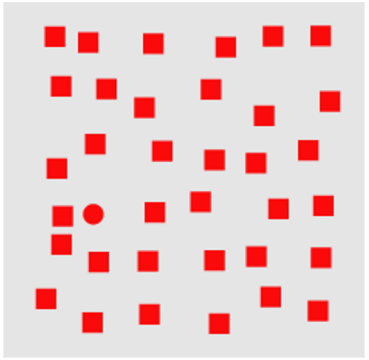
Pre-attentive Processing

- Very fast: < 200-250 ms (response time for color)
- What matters most is the contrast defined by a unique feature

Color



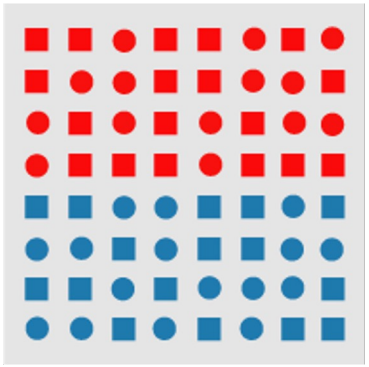
Shape



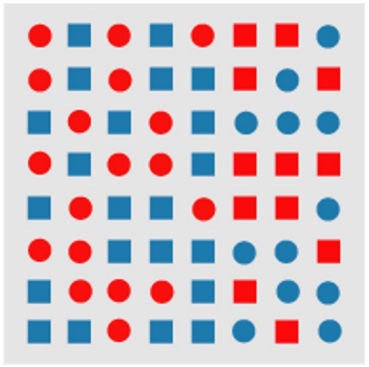
Pre-attentive Processing

- Combing pre-attentive features does not always work
- Example: a boundary defined by …

a unique feature hue



a conjunction of features



Gestalt Principles

- Gestalt school of psychology (1912)
- Understand pattern perception
- “gestalt” German for “pattern” or “form, configuration”
- “Gestalt refers to the patterns that you perceive when presented with a few graphical elements.”
- Principles of Pattern Recognition
  - Original proposed mechanisms turned out to be wrong
  - Rules themselves are still useful

https://medium.com/@Elijah_Meeks/gestalt-principles-for-data-visualization-59f18f20bd40

Slide adapted from Tamara Munzner

Gestalt Properties

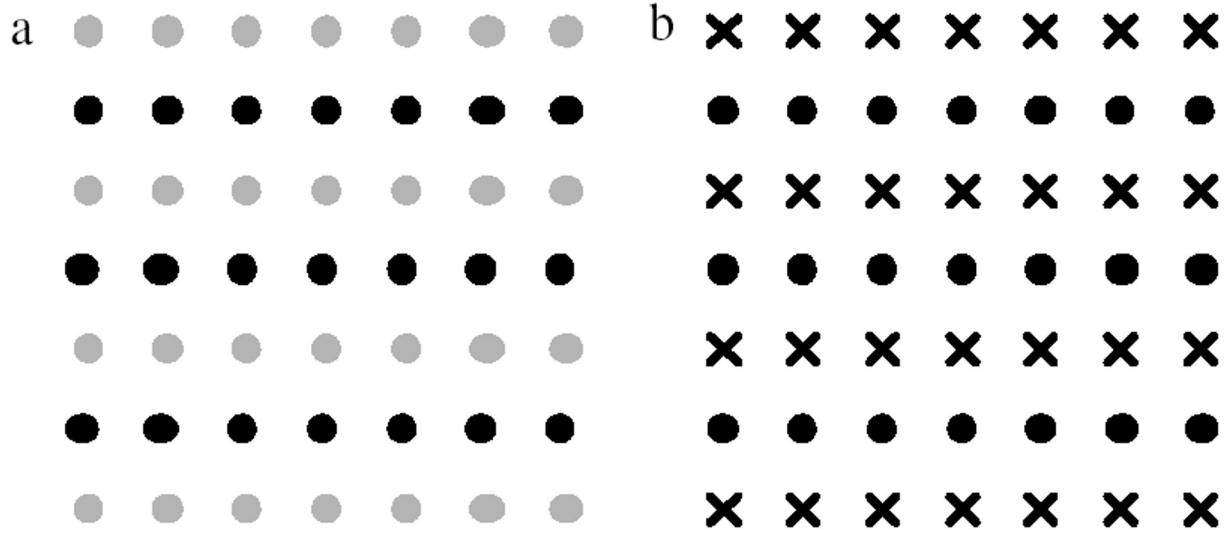
- **Spatial proximity** is a powerful perceptual organizing principle and one of the most useful in design.
- Things that are close together are perceptually grouped together.

Gestalt Properties

- Proximity

Gestalt Properties

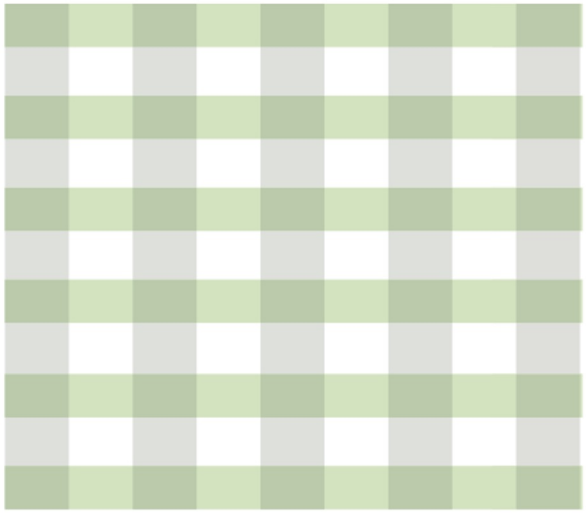
- **Similarity**
- Similar elements tend to be grouped together.



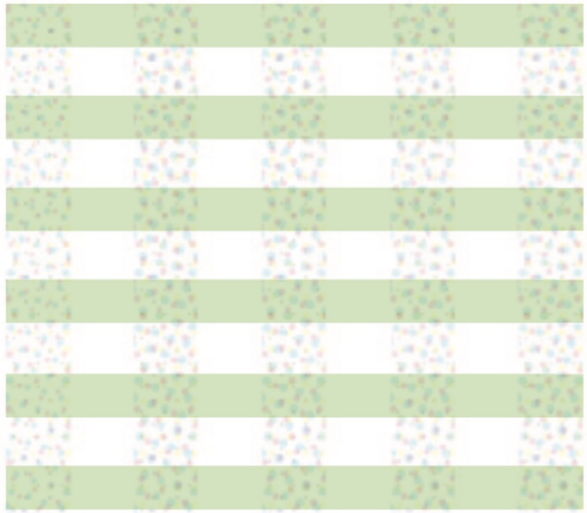
Slide adapted from Tamara Munzner

Gestalt Properties

- Similarity



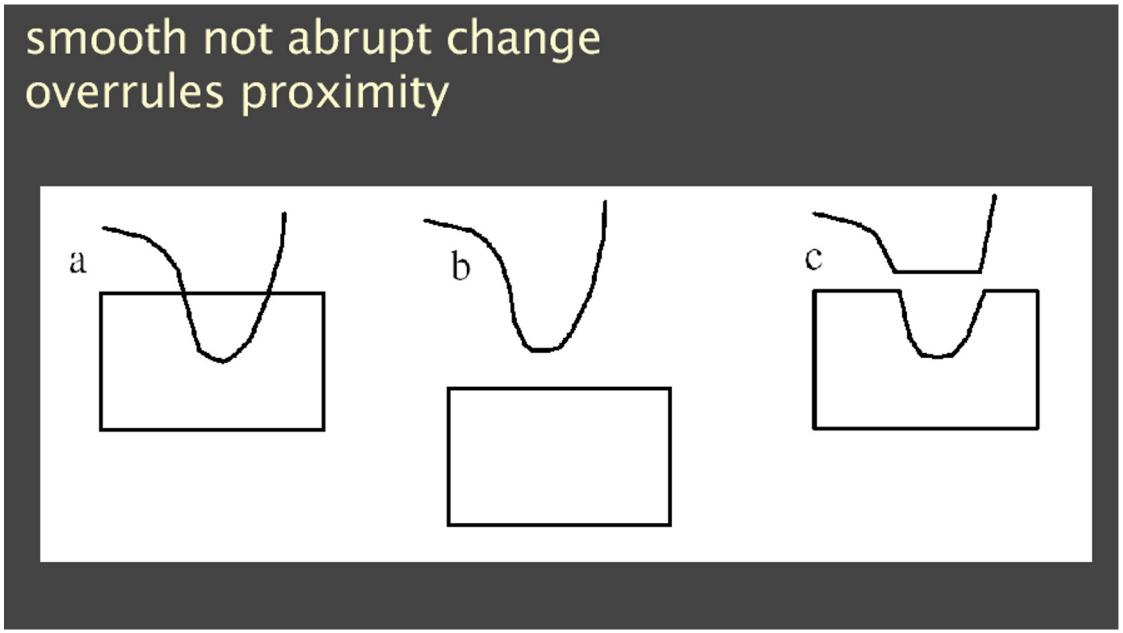
(c)



(d)

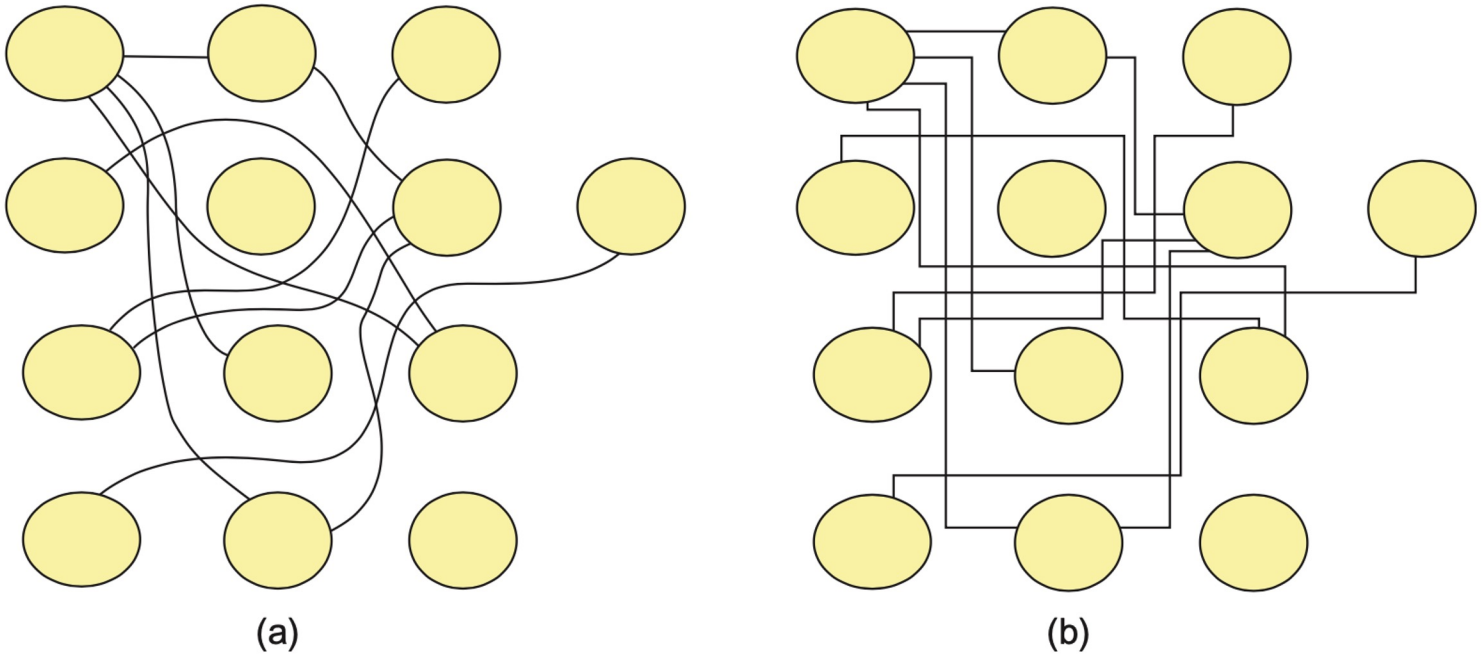
Gestalt Properties

- **Continuity**
- We are more likely to construct visual entities out of visual elements that are smooth and continuous.



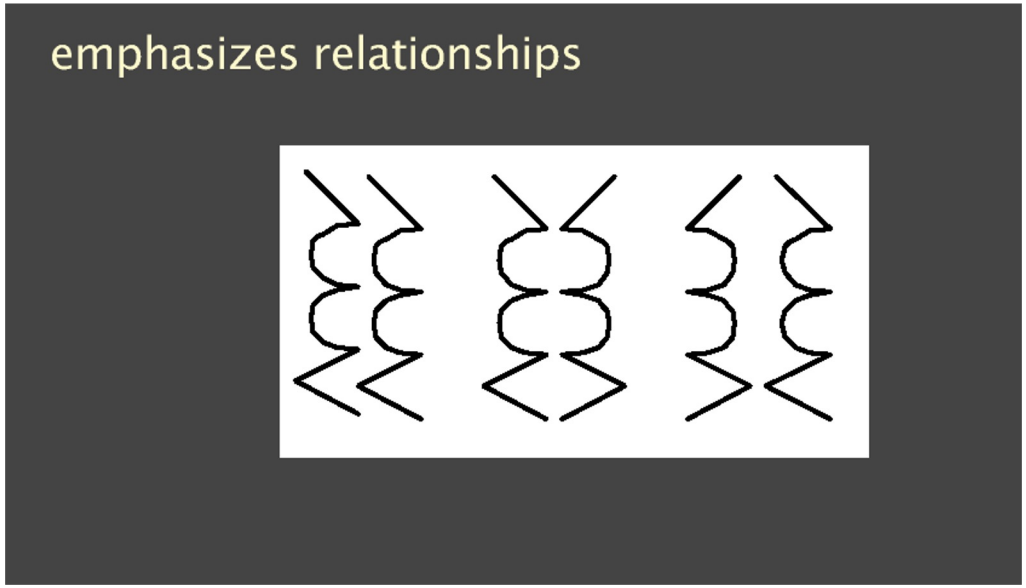
Gestalt Properties

- Continuity



Gestalt Properties

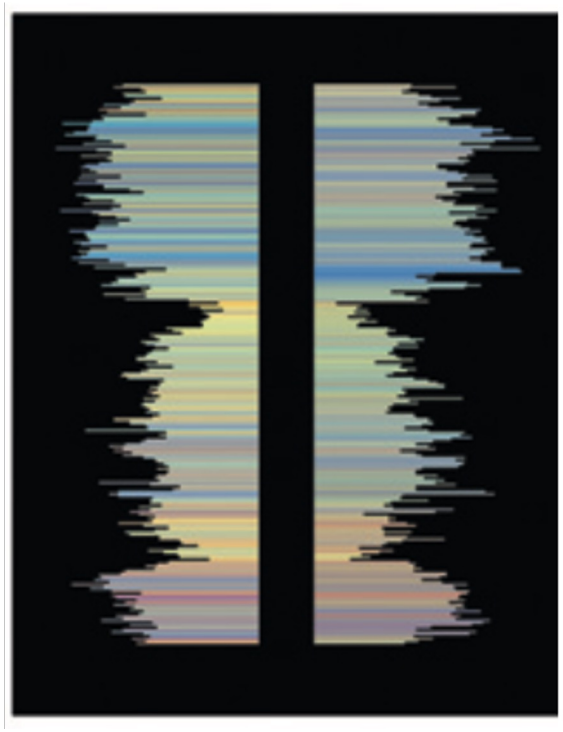
- Symmetry



Slide adapted from Tamara Munzner

Gestalt Properties

- Symmetry

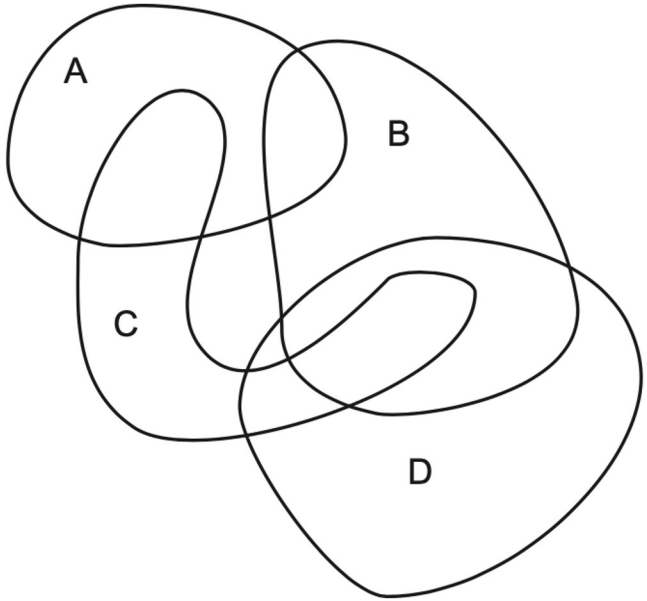


Gestalt Properties

- **Closure**
- There is a perceptual tendency to close

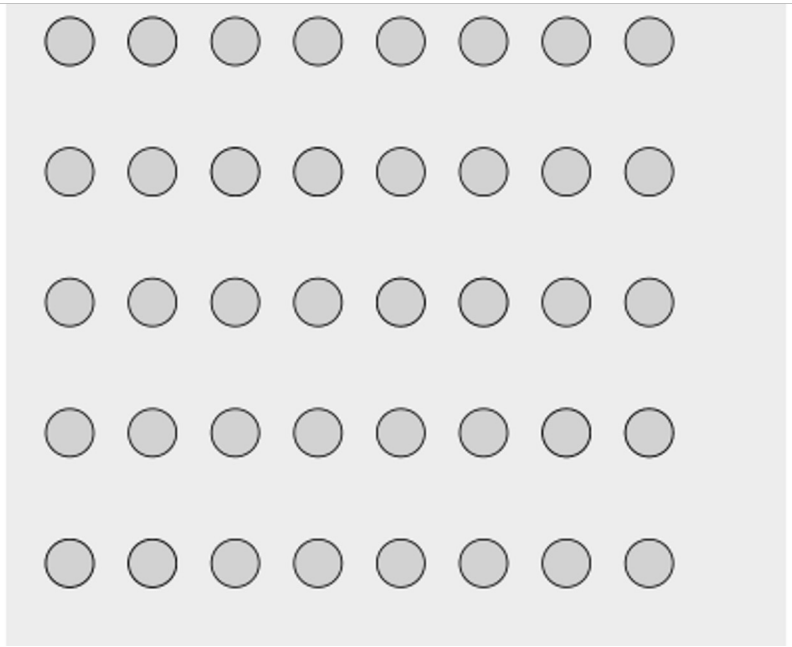
Gestalt Properties

- Closure

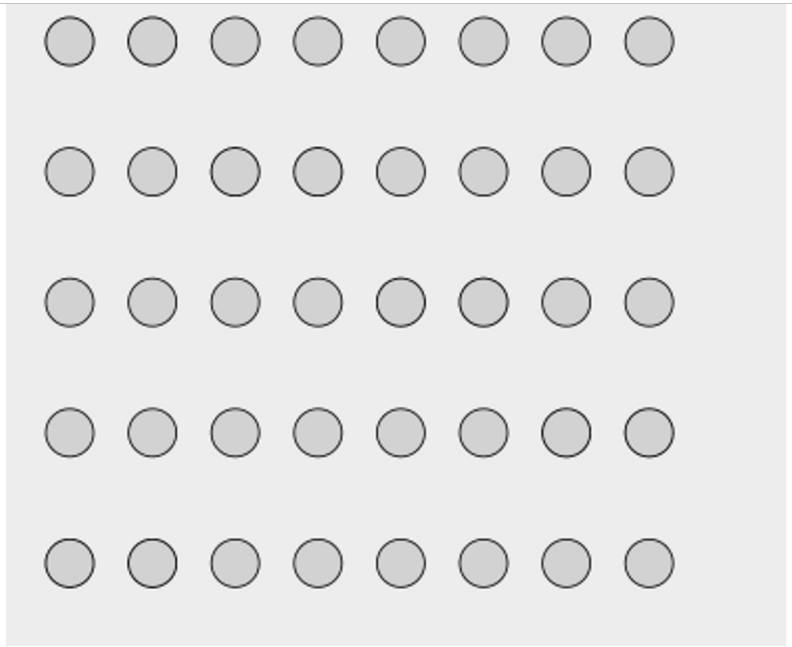


Example: Venn diagrams

Which gestalt principles you've observed in this animated figure?

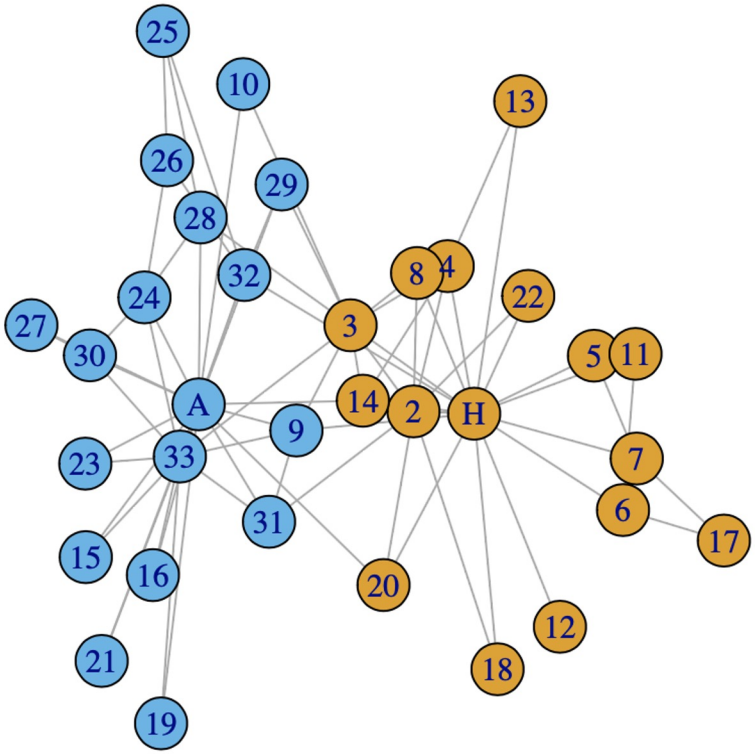


Which gestalt principles you've observed in this animated figure?



Similarity, Proximity & Enclosure

Which gestalt principles you've observed in this figure?

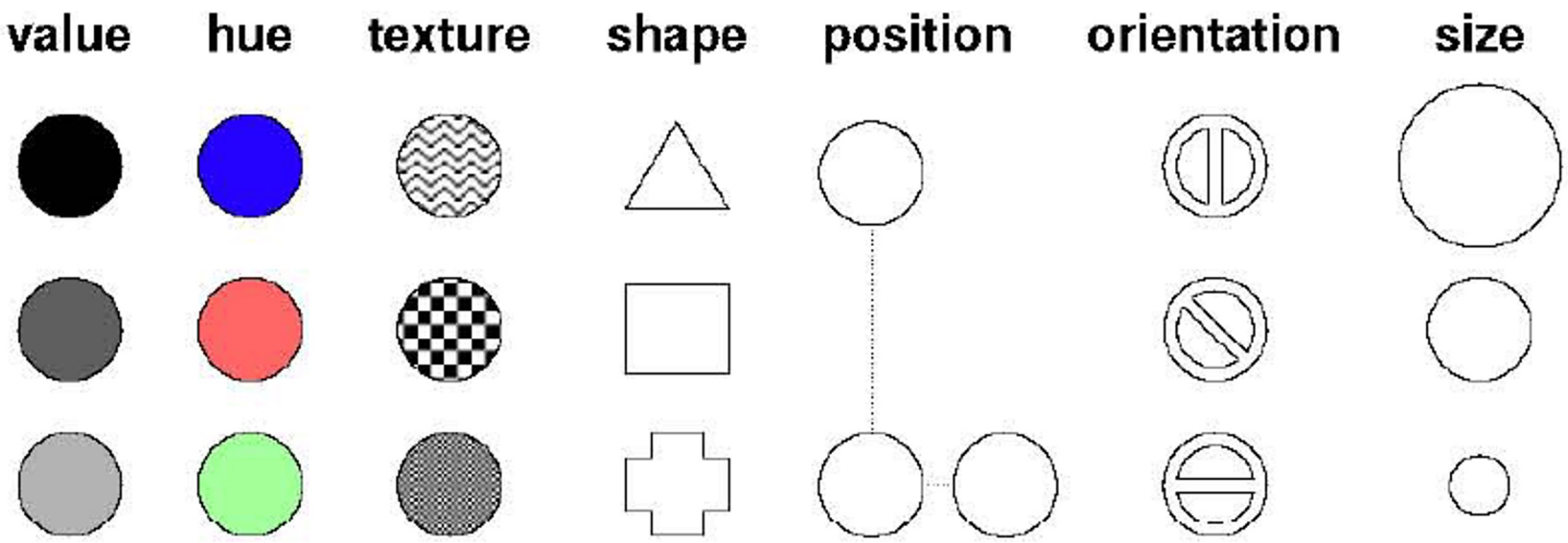


The karate club network

Sensory vs. Arbitrary Symbols

- Sensory:
  - Understanding without training
  - Resistance to instructional bias
  - Sensory immediacy
    - Hard-wired and fast
  - Cross-cultural validity
- Arbitrary
  - Hard to learn
  - Easy to forget
  - Embedded in culture and applications

Visual variables



Semiology of graphics

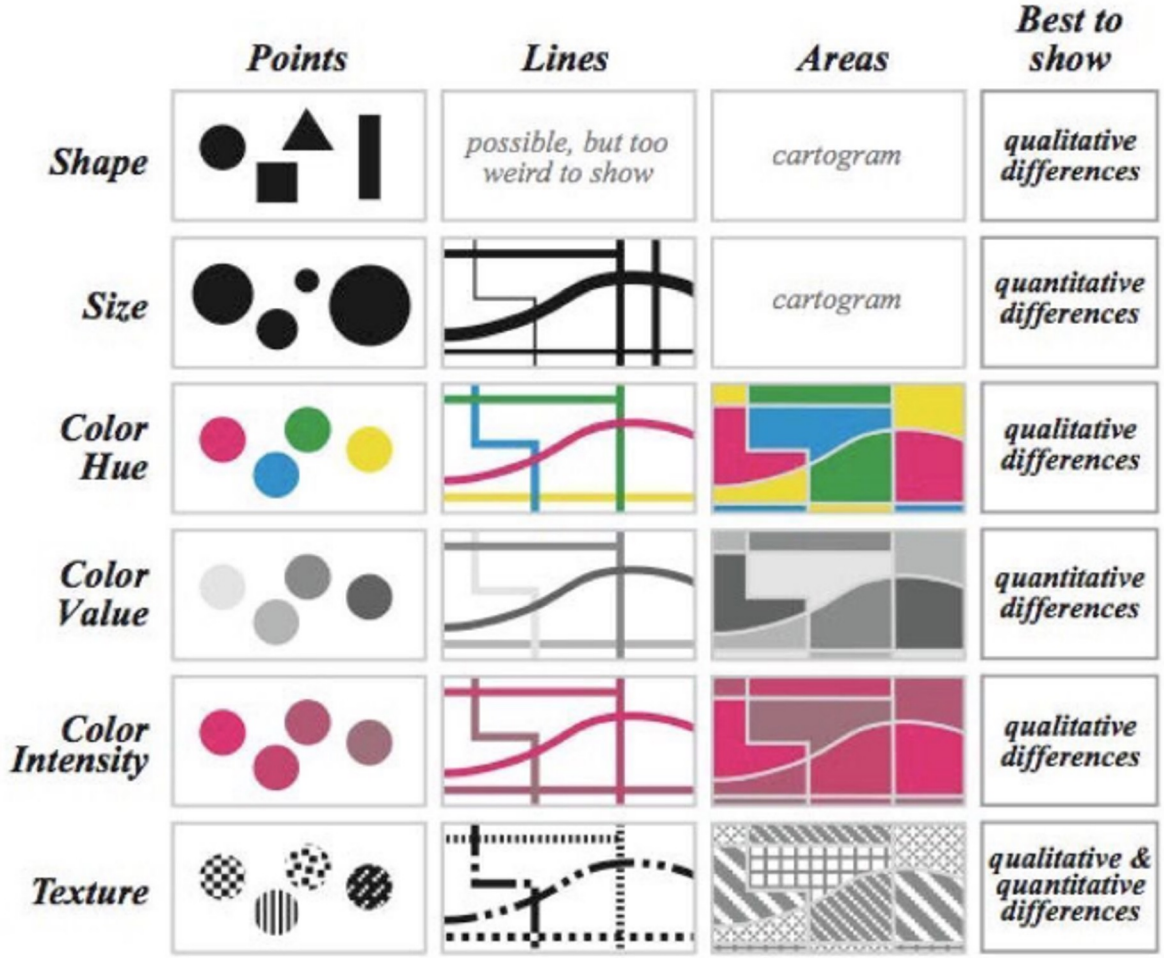
- semiology = study of signs and sign processes, likeness, analogy, metaphor, symbolism, signification, and communication (Wikipedia)

- Jacques Bertin, Semiology of Graphics, 1983

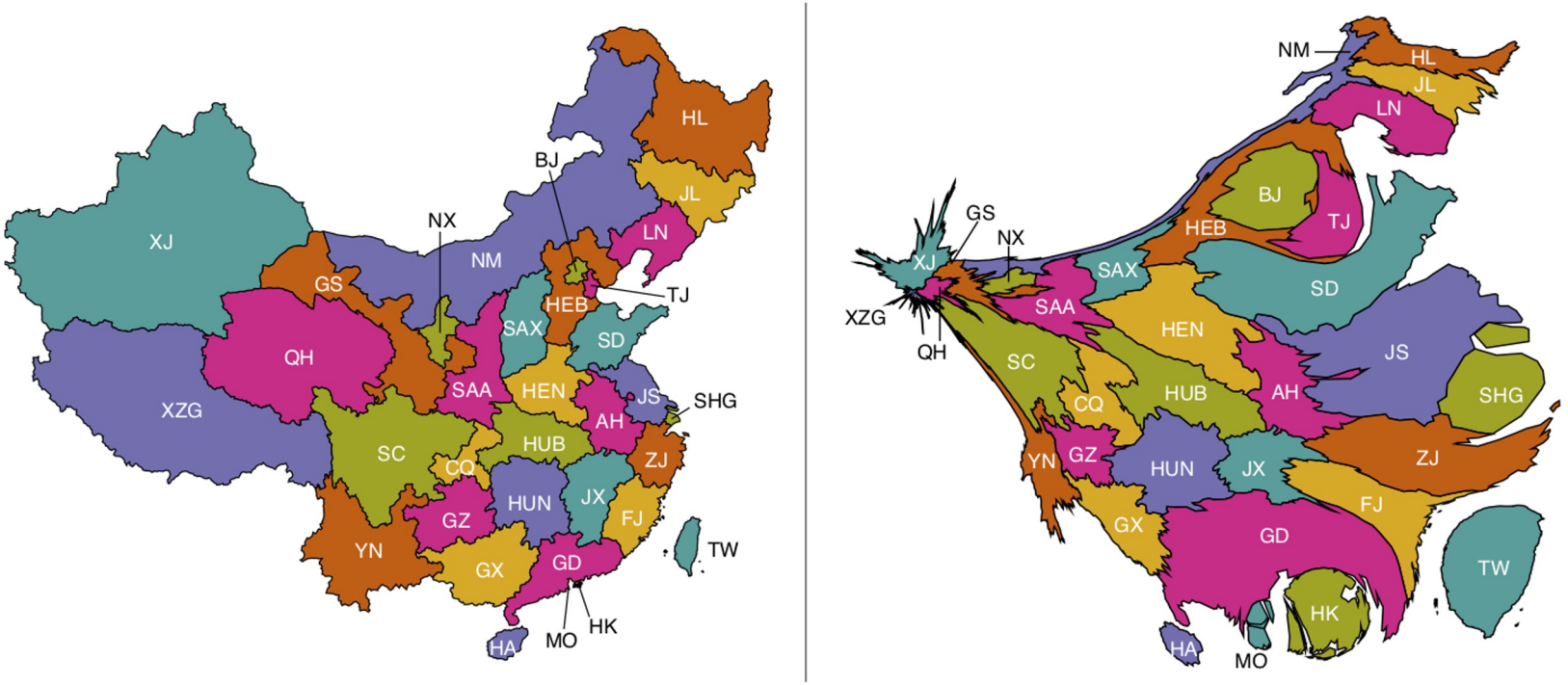
- The theoretical foundation of data visualization.

- Guides for visual encoding (the process of transforming data into graphical elements):
  - What – points, lines, area (patterns, trees / networks, grids)
  - Where – positional: XY (1D, 2D, 3D)
  - How – retinal: Z (size, lightness, texture, color, orientation, shape)
  - When – temporal: animation

Jacques Bertin “Semiology of Graphics”

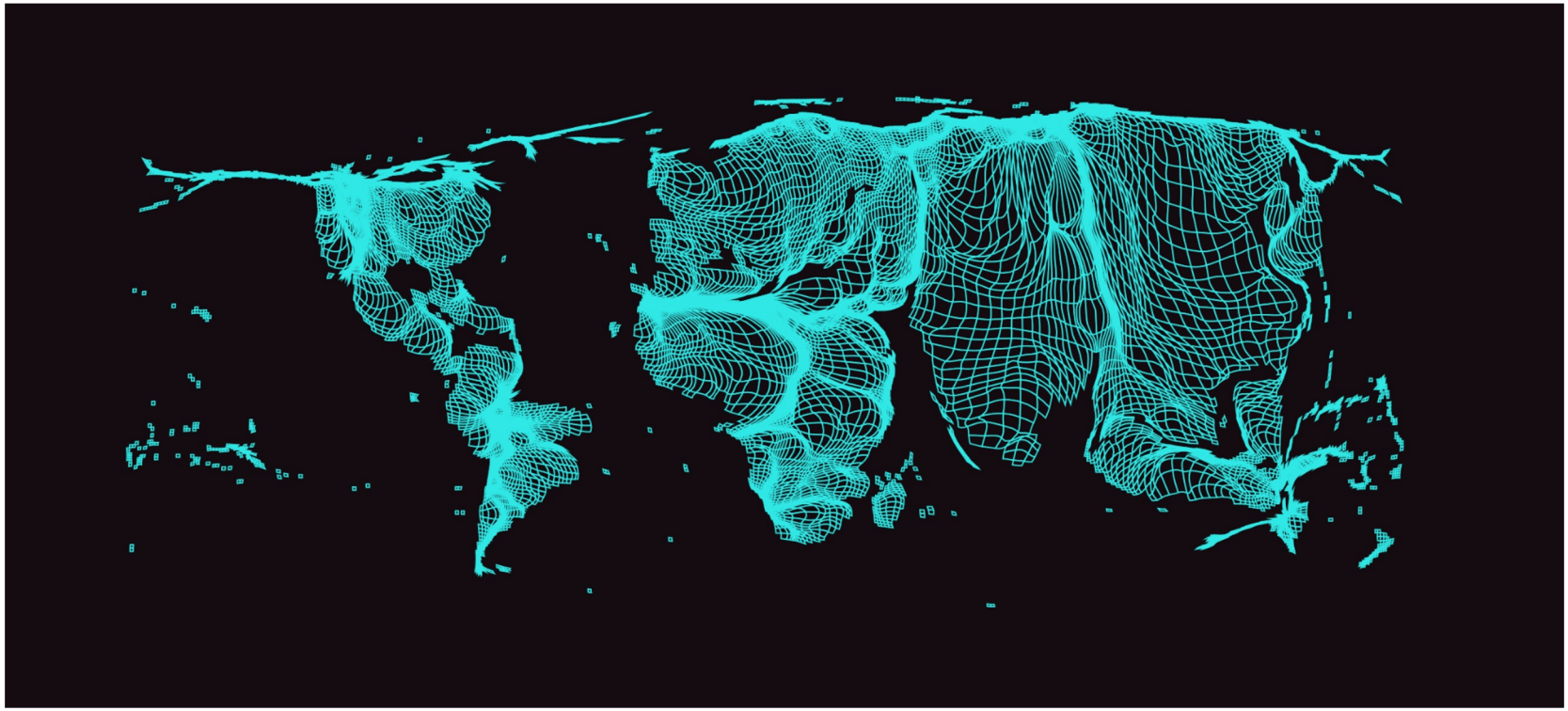


Cartogram (areas are proportion to GDP)



https://www.pnas.org/content/115/10/E2156

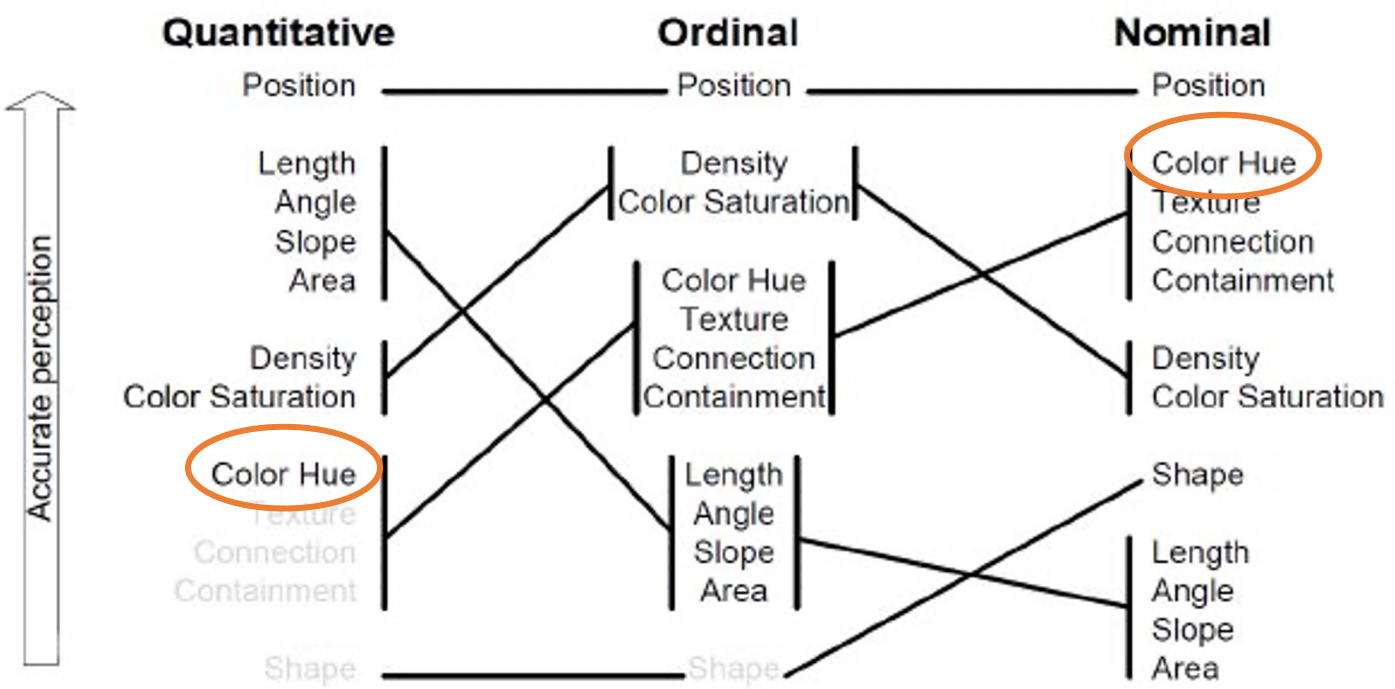
Cartogram (proportion to population)



https://www.pnas.org/content/115/10/E2156

The Mackinlay ranking of perceptual task

Effectiveness of encoding to assess alternative designs



How much (quantitative)

What / where (qualitative)

Color

- How many colors should be used?

  One of the best practices when it comes to data visualization design is **maintaining a single color** as standard for most charts.

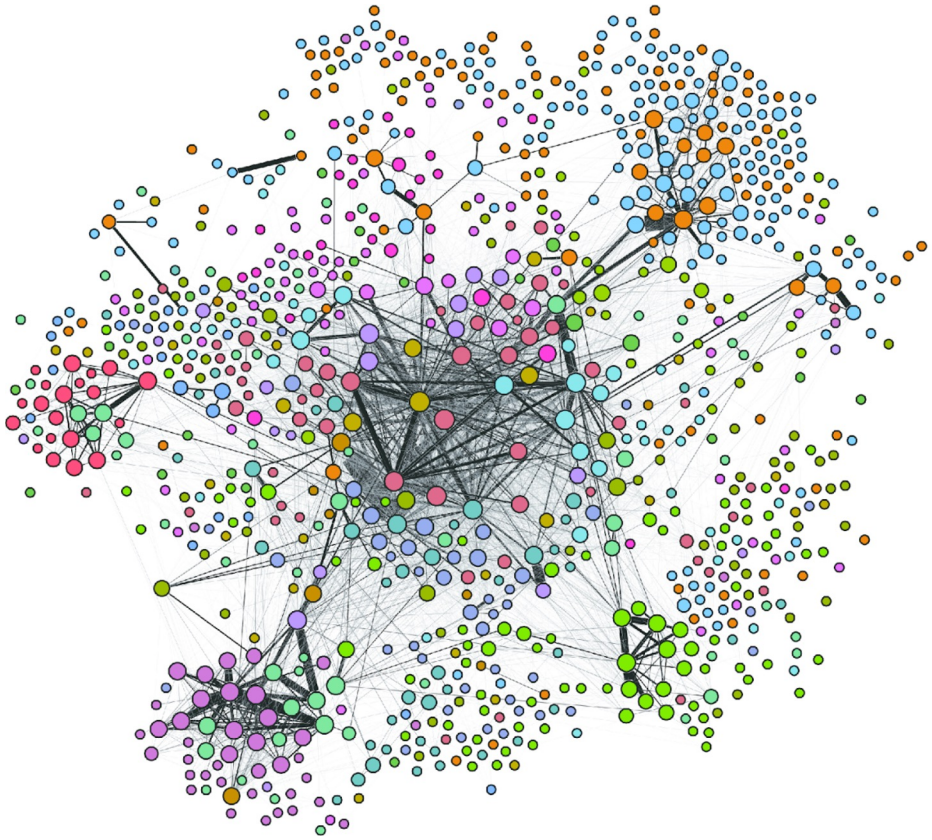
- The best time to introduce different colors?

  To differentiate between specific groups.

  - People can name ~9 colors

    but can distinguish gradations much easier
  - Research shows no more than ~6 should be used

Example: Are the colors good?

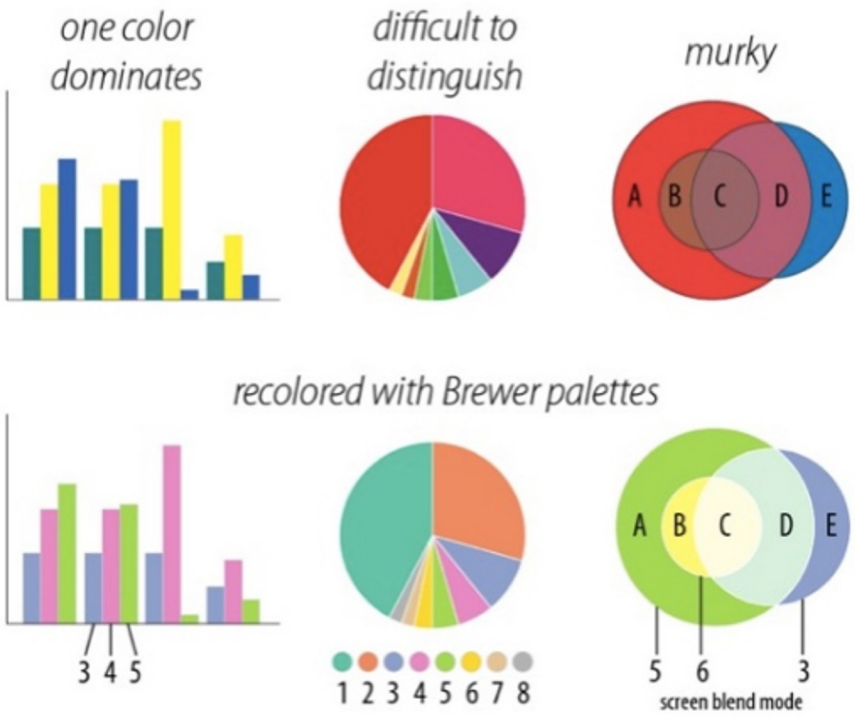


Diseases and Injuries Tabular Index

1. INFECTIOUS AND PARASITIC DISEASES (001-139)
2. NEOPLASMS (140-239)
3. ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES, AND IMMUNITY DISORDERS (240-279)
4. DISEASES OF THE BLOOD AND BLOOD-FORMING ORGANS (280-289)
5. MENTAL DISORDERS (290-319)
6. DISEASES OF THE NERVOUS SYSTEM AND SENSE ORGANS (320-389)
7. DISEASES OF THE CIRCULATORY SYSTEM (390-459)
8. DISEASES OF THE RESPIRATORY SYSTEM (460-519)
9. DISEASES OF THE DIGESTIVE SYSTEM (520-579)
10. DISEASES OF THE GENITOURINARY SYSTEM (580-629)
11. COMPLICATIONS OF PREGNANCY, CHILDBIRTH, AND THE PUERPERIUM (630-679)
12. DISEASES OF THE SKIN AND SUBCUTANEOUS TISSUE (680-709)
13. DISEASES OF THE MUSCULOSKELETAL SYSTEM AND CONNECTIVE TISSUE (710-739)
14. CONGENITAL ANOMALIES (740-759)
15. CERTAIN CONDITIONS ORIGINATING IN THE PERINATAL PERIOD (760-779)
16. SYMPTOMS, SIGNS, AND ILL-DEFINED CONDITIONS (780-799)
17. INJURY AND POISONING (800-999)
18. SUPPLEMENTARY CLASSIFICATION OF FACTORS INFLUENCING HEALTH STATUS AND CONTACT WITH HEALTH SERVICES (V01-V89)
19. SUPPLEMENTARY CLASSIFICATION OF EXTERNAL CAUSES OF INJURY AND POISONING (E800-E999)

https://www.sciencedirect.com/science/article/pii/S0165032720326501

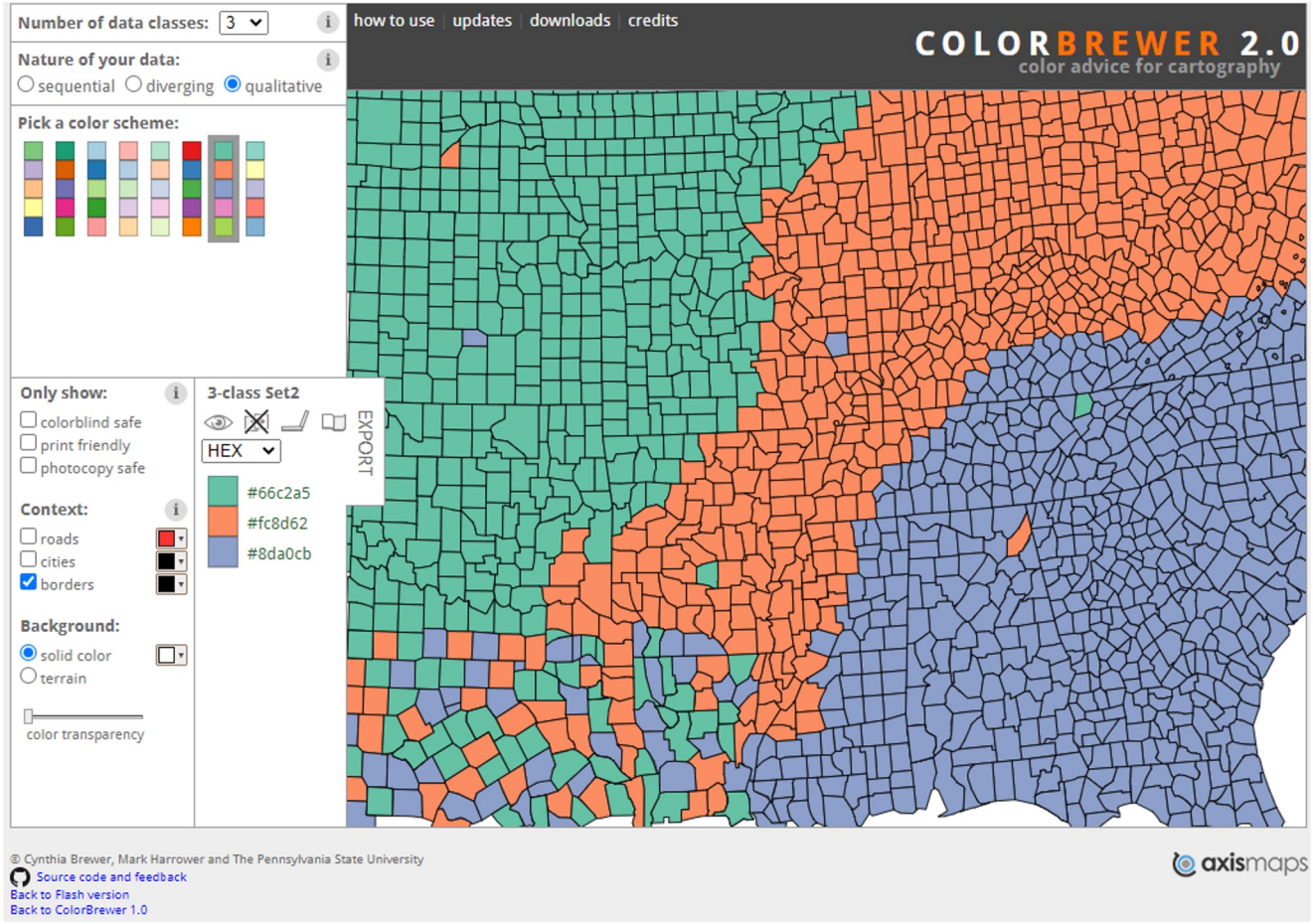
Color schemes / combination



Color – other notes

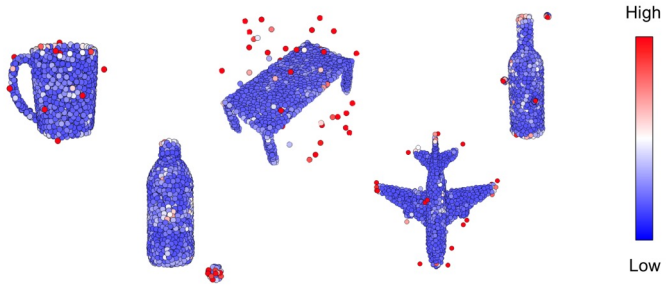
- Color blindness
  - use Colorblind-friendly palettes
- Because there are fewer blue cones, blue should be avoided for small objects in a graph, but is an effective background color.
- Becomes worse with age

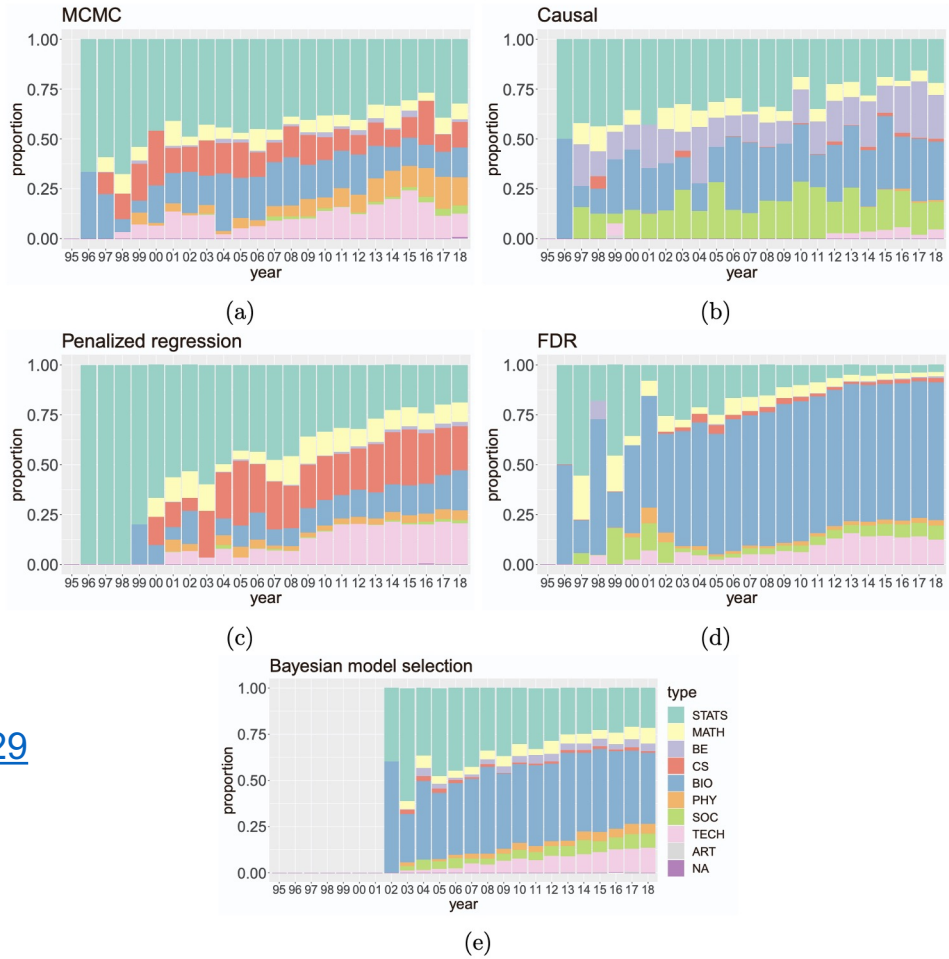
Color advice: ColorBrewer



https://colorbrewer2.org/#type=qualitative&scheme=Set2&n=3

Color



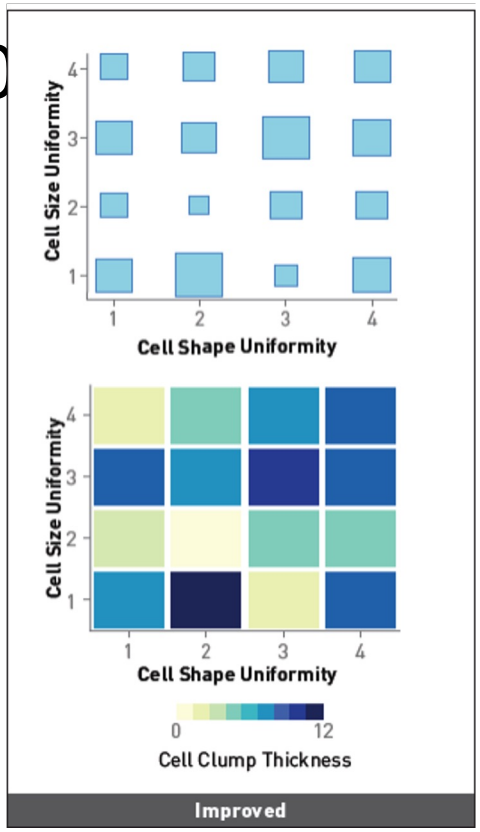


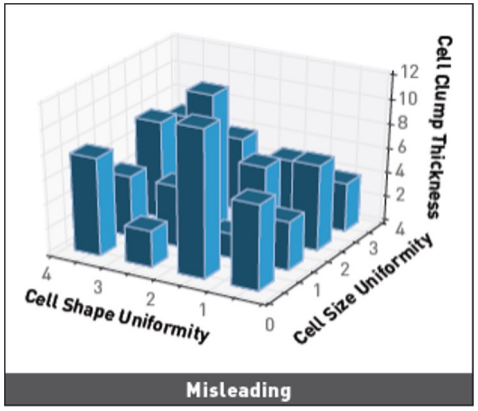
https://ojs.aaai.org/index.php/AAAI/article/view/30129

https://www.cell.com/patterns/pdfExtended/S2666-3899(22)00129-5

Dangers of Depth (3D)

- We do NOT see in 3D; we see in 2.0
- interaction complexity
- perspective distortion

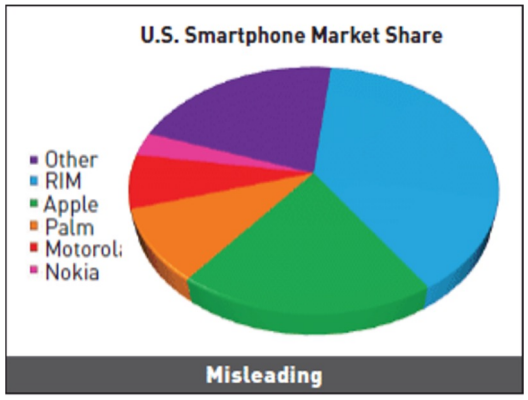


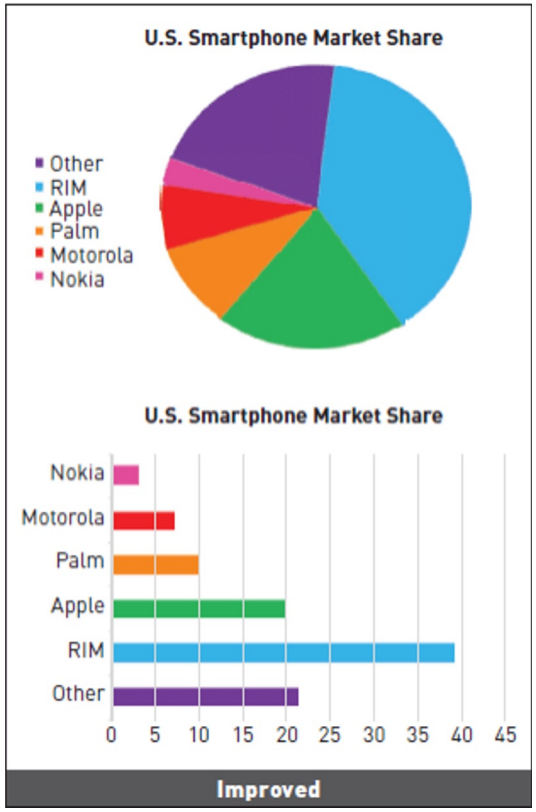


Dangers of Depth (3D)

- Distortion due to projection in the 3D pie chart causes the green wedge to represent a far larger market share than the data supports.

$$\text{“lie factor”} = \frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$$





Dangers of Depth (3D)

- To avoid occlusion and ambiguity in visualizations, use 3D only when absolutely necessary
- Instead of representing the third dimension of your data using depth, try using alternative visual variables like **color or size**
- Some kinds of data, like molecular surfaces or architectural structures, have inherent 3D shapes. In these cases, 3D can provide important contextual information
- Pairing 2D summary representations with 3D structures can help overcome these limitations, even for complex geometries and inherently spatial data
- https://www.imaios.com/en/e-Anatomy/Head-and-Neck/Head-and-neck-CT

Mental model

Clever or more “efficient” designs cannot violate mental models

Adhere to data presentation standards in your field

- Judged by those often familiar with research field
- Expected presentations of data in that field
- When review scientific articles,
  - how is data presented?
    - Are there graphs?
    - What kind?
    - What statistics are used?
    - Does the journal have a style guide?

Adhere to data presentation standards in your field

- See the difference?

代码	名称	涨幅%	总市值	流通市值	现价	涨跌	买价
600614	*ST鹏起	-5.13	12.97亿	11.19亿	0.74	-0.04	-
000820	*ST节能	+1.11	5.80亿	2.62亿	0.91	+0.01	0.90
000587	*ST金洲	+2.22	19.54亿	11.16亿	0.92	+0.02	0.91
600856	*ST中天	+0.00	12.85亿	12.63亿	0.94	+0.00	0.94
002210	*ST飞马	-2.06	15.70亿	12.59亿	0.95	-0.02	0.95
600634	*ST富控	-4.76	5.76亿	5.76亿	1.00	-0.05	-
600255	*ST梦舟	-0.93	18.76亿	18.76亿	1.06	-0.01	1.06
600010	包钢股份	-1.80	496.9亿	345.3亿	1.09	-0.02	1.09
600122	*ST宏图	-3.45	12.97亿	12.97亿	1.12	-0.04	1.12
600978	*ST宜生	-4.17	17.05亿	17.05亿	1.15	-0.05	1.15
600687	*ST刚泰	+0.00	17.27亿	17.27亿	1.16	+0.00	1.15
600290	*ST华仪	-4.88	8.89亿	8.89亿	1.17	-0.06	-
002256	*ST兆新	+2.63	22.02亿	16.32亿	1.17	+0.03	1.17
002359	*ST北讯	-4.69	13.26亿	10.15亿	1.22	-0.06	-
600568	*ST中珠	+0.00	24.51亿	19.25亿	1.23	+0.00	1.22
002122	*ST天马	-1.56	14.97亿	14.97亿	1.26	-0.02	1.25
000816	ST慧业	+2.42	18.02亿	16.97亿	1.27	+0.03	1.26

Data Preprocessing

- Why preprocess the data?
- How to clean the data?
- Data integration and transformation
- Data reduction
- Discretization and concept hierarchy generation

Why

- Data in the real world is dirty
  - incomplete: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data;
    - Very often you don’t know why certain values are missing.
  - noisy: containing errors or outliers
  - inconsistent: discrepancies in codes or names

Why

- No quality data, no quality mining results!
  - Quality decisions must be based on quality data
  - Data warehouse needs consistent integration of quality data – this is often the most difficult part!

Example: Inconsistency

Mars Probe Lost Due to Simple Math Error

October 01, 1999 | ROBERT LEE HOTZ | TIMES SCIENCE WRITER

NASA lost its $125-million Mars Climate Orbiter because spacecraft engineers failed to convert from English to metric measurements when exchanging vital data before the craft was launched, space agency officials said Thursday.

A navigation team at the Jet Propulsion Laboratory used the metric system of millimeters and meters in its calculations, while Lockheed Martin Astronautics in Denver, which designed and built the spacecraft, provided crucial acceleration data in the English system of inches, feet and pounds.

As a result, JPL engineers mistook acceleration readings measured in English units of pound-seconds for a metric measure of force called newton-seconds.

In a sense, the spacecraft was lost in translation.

"That is so dumb," said John Logsdon, director of George Washington University's space policy institute. "There seems to have emerged over the past couple of years a systematic problem in the space community of insufficient attention to detail."

Remember the Mars Climate Orbiter incident from 1999?

Major Tasks

- Data cleaning
  - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
  - Example: filling in missing values in blood test
- Data integration
  - Integration of multiple databases, data cubes, or files
  - Example: integrating electronic health record with nursing home data
- Data transformation
  - Normalization and aggregation
  - Example: age, weight, height, income
- Data reduction (More discussion later)
  - Obtains reduced representation in volume but produces the same or similar analytical results
  - Example: under-sample in tumor detection; embedding method in deep learning
- Data discretization
  - Part of data reduction but with particular importance
  - Example: young, middle-age, elderly

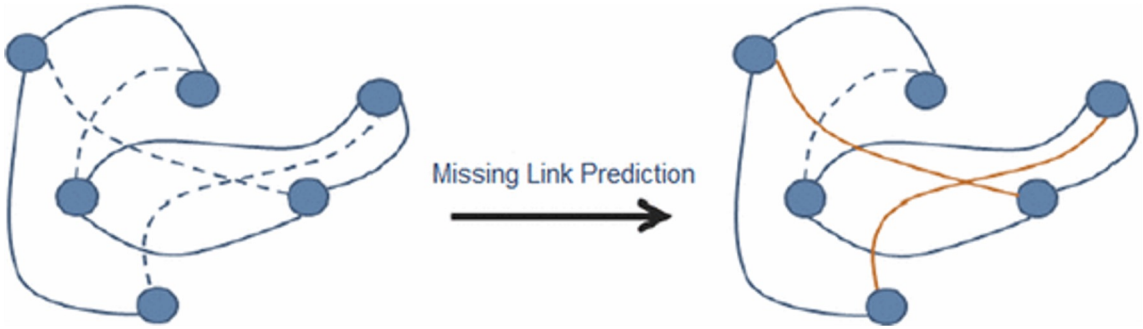
Data Cleaning

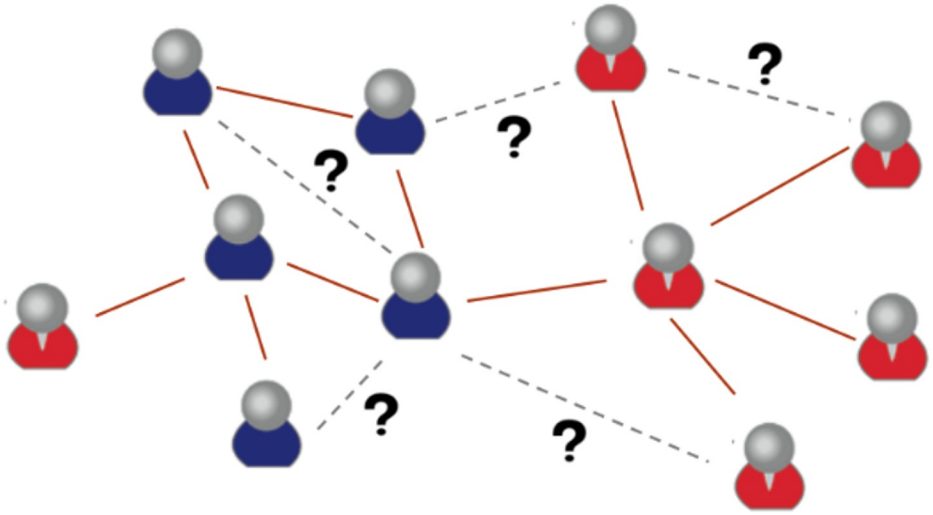
- Data cleaning tasks
  - Fill in missing values
  - Identify outliers and smooth out noisy data
  - Correct inconsistent data

Missing Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to many reasons, e.g.,
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data sometimes is not random.

Example: Link Prediction





How to handle missing data?

- Method 1: Ignore the tuple: usually done when **class label** is missing
  - assuming the tasks in classification—not effective when the percentage of missing values per attribute varies considerably
- Method 2: Fill in the value
  - Fill in the missing value manually: tedious + infeasible?
  - Use a global constant to fill in the missing value: e.g., “unknown”, a new class?!
  - Use the attribute mean to fill in the missing value
  - Use the most probable value (e.g., mode) to fill in the missing value

Missing data detection in Python

Using package pandas

```
[1]: df
```

```
     A     B     C    D
0  1.0   2.0   3.0  4.0
1  5.0   6.0   NaN  8.0
2 10.0  11.0  12.0  NaN
```

```
[2]: df.isnull()
```

```
       A      B      C      D
0  False  False  False  False
1  False  False   True  False
2  False  False  False   True
```

```
[3]: df.notnull()
```

```
      A     B      C     D
0  True  True   True  True
1  True  True  False  True
2  True  True   True  False
```

Missing data detection in Python

Using the sum method, we can then return the number of missing values per column as follows:

```
[4]: df.isnull().sum()
```

```
[4]: A      0
     B      0
     C      1
     D      1
     dtype: int64
```

(True = 1, False = 0)

Remove missing data in python

```
df
```

[1]:

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	NaN	8.0
2	10.0	11.0	12.0	NaN

simply remove the corresponding features (columns) or observations (rows) from the dataset entirely.

Drop rows

```
[6]: df.dropna(axis=0)
```

[6]:

	A	B	C	D
0	1.0	2.0	3.0	4.0

Drop columns

```
[7]: df.dropna(axis=1)
```

[7]:

	A	B
0	1.0	2.0
1	5.0	6.0
2	10.0	11.0

Remove missing data in python

[8]:
```
# only drop rows where all columns are NaN
# (returns the whole array here since we don't
# have a row with where all values are NaN
df.dropna(how='all')
```

[8]:

	A	B	C	D
0	1.0	2.0	3.0	4.0
1	5.0	6.0	NaN	8.0
2	10.0	11.0	12.0	NaN

[9]:
```
# drop rows that have less than 4 real values
df.dropna(thresh=4)
```

[9]:

	A	B	C	D
0	1.0	2.0	3.0	4.0

[10]:
```
# only drop rows where NaN appear in specific columns (here: 'C')
df.dropna(subset=['C'])
```

[10]:

	A	B	C	D
0	1.0	2.0	3.0	4.0
2	10.0	11.0	12.0	NaN

Imputing missing data in python

```
[5]: df.values

[5]: array([[ 1.,  2.,  3.,  4.],
            [ 5.,  6., nan,  8.],
            [10., 11., 12., nan]])
```

Note: scikit-learn was developed for working with NumPy arrays.
We can always access the NumPy array of a DataFrame via the values attribute.

Replace the missing value with the mean value of the entire feature column.

```
[11]: import numpy as np
      from sklearn.impute import SimpleImputer
      imp = SimpleImputer(missing_values = np.nan, strategy = 'mean')
      imp = imp.fit(df.values)
      imputed_data = imp.transform(df.values)
      imputed_data

[11]: array([[ 1. ,  2. ,  3. ,  4. ],
            [ 5. ,  6. ,  7.5,  8. ],
            [10. , 11. , 12. ,  6. ]])
```

Imputing missing data in python

Replace the missing value with constant of the entire feature column. Default value is 0.

```
In [5]: imp = SimpleImputer(missing_values = np.nan, strategy = "constant")
        imp = imp.fit(df.values)
        imputed_data = imp.transform(df.values)
        imputed_data

Out[5]: array([[ 1.,  2.,  3.,  4.],
               [ 5.,  6.,  0.,  8.],
               [10., 11., 12.,  0.]])
```

Replace the missing value with 1000 of the entire feature column.

```
In [5]: imp = SimpleImputer(missing_values = np.nan, strategy = "constant",fill_value = 1000)
        imputed_data = imp.fit_transform(df.values)
        imputed_data

Out[5]: array([[   1.,    2.,    3.,    4.],
               [   5.,    6., 1000.,    8.],
               [  10.,   11.,   12., 1000.]])
```

Imputing missing data by k-Nearest Neighbors

```
import numpy as np
from sklearn.impute import KNNImputer
X = [[1, 2, np.nan], [3, 4, 3], [np.nan, np.nan, 5], [8, 8, 7]]
df = pd.DataFrame(X, columns=['A', 'B', 'C'])
df
```

	A	B	C
0	1.0	2.0	NaN
1	3.0	4.0	3.0
2	NaN	NaN	5.0
3	8.0	8.0	7.0

The KNNImputer class provides imputation for filling in missing values using **the k-Nearest Neighbors** approach

```
imputer = KNNImputer(n_neighbors=2, weights="uniform")
imputer.fit_transform(X) # you will get a np array
```

```
array([[1. , 2. , 5. ],
       [3. , 4. , 3. ],
       [5.5, 6. , 5. ],
       [8. , 8. , 7. ]])
```

Imputing missing data by k-Nearest Neighbors

- By default, a euclidean distance metric that supports missing values, nan_euclidean_distances, is used to find the nearest neighbors.
  https://scikit-learn.org/dev/modules/generated/sklearn.metrics.pairwise.nan_euclidean_distances.html

- Each observation's missing values are imputed using **the mean values from n_neighbors nearest neighbors** found in the training data sets**.**

- Two observations are close if the features that neither is missing are

```
imputer = KNNImputer(n_neighbors=2, weights="uniform")
imputer.fit_transform(X) # you will get a np array
```

```
array([[1. , 2. , 5. ],
       [3. , 4. , 3. ],
       [5.5, 6. , 5. ],
       [8. , 8. , 7. ]])
```

We set n_neighbors to 2, which means that the number of neighboring observations to use for imputation is 2. n_neighbors' default value is 5.

Imputing missing data in python

One example about “distance” (nan_euclidean_distances) between two observations mentioned above is computed as the following way:

the distance between $[3, \text{np.nan}, \text{np.nan}, 6]$ and $[1, \text{np.nan}, 4, 5]$

$$\sqrt{\frac{4}{2}\left((3-1)^2+(6-5)^2\right)}$$

Exercise

- The distance between the following two observations.

[10, 4, 11, 9, 5]

[NaN, 2, 10, NaN, 6]

Noise

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data noise problems which require data cleaning
  - duplicate records
  - incomplete data
  - inconsistent data

How to handle noisy data?

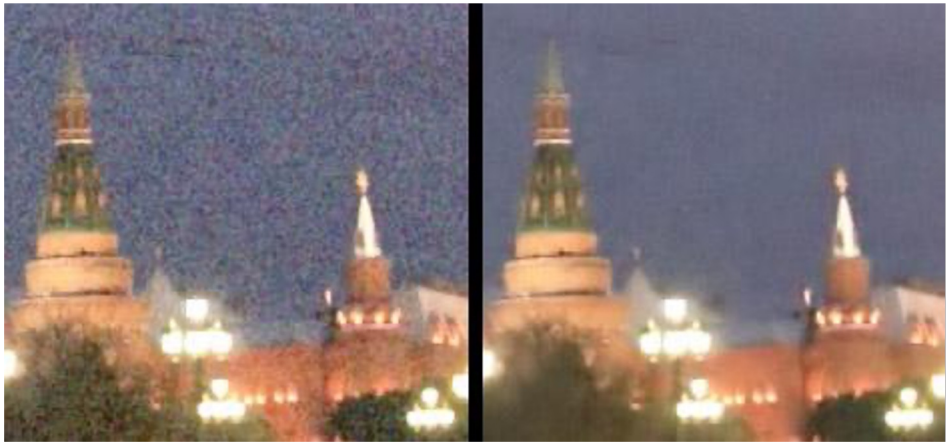
- Binning method:
  - first sort data and partition into (equal-depth) bins
  - then smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Clustering
  - detect and remove outliers
- Combined computer and human inspection
  - detect suspicious values and check by human
- Regression
  - smooth by fitting the data into regression functions

Data binning

- Equal-width (distance) partitioning:
  - It divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: W = (B-A)/N.
  - The most straightforward
  - But outliers may dominate presentation. Why?
  - Skewed data is not handled well.
- Equal-depth (frequency) partitioning:
  - It divides the range into *N* intervals, each containing approximately same number of samples
  - Good data scaling
  - Managing categorical attributes can be tricky.

Data binning

(a) No binning (b) Kodak PIXELUX (c) Phase One



Data binning

* Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34
* Partition into 3 (equal-depth) bins:
  - **Bin 1:** 4, 8, 9, 15
  - **Bin 2:** 21, 21, 24, 25
  - **Bin 3:** 26, 28, 29, 34
* Smoothing by bin means:
  - **Bin 1:** 9, 9, 9, 9
  - **Bin 2:** 23, 23, 23, 23
  - **Bin 3:** 29, 29, 29, 29
* Smoothing by bin (closer) boundaries:
  - **Bin 1:** 4, 4, 4, 15
  - **Bin 2:** 21, 21, 21, 25
  - **Bin 3:** 26, 26, 26, 34

Data integration

- Data integration:
  - combines data from multiple sources into a coherent store
- Schema integration
  - integrate metadata from different sources
  - Entity identification problem: identify real world entities from multiple data sources
- Detecting and resolving data value conflicts
  - for the same real world entity, attribute values from different sources are different
  - possible reasons: different representations, different scales, e.g., metric vs. British units

Data integration

- Redundant data occur often when integration of multiple databases
  - The same attribute may have different names in different databases
  - Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Data transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
  - min-max normalization
  - z-score normalization
  - normalization by decimal scaling
  - Log transformation normalization

Normalization

- min-max normalization

$$v' = \frac{v - min_A}{max_A - min_A}$$

- z-score normalization

$$v' = \frac{v - mean_A}{stand\_dev_A}$$

- normalization by decimal scaling

$$v' = \frac{v}{10^j}$$ Where j is the smallest integer such that Max($|v'|$)<1

- Log transformation normalization

v' = log(v)

Normalization

marks
8
10
15
20

Mean = 13.25

Standard deviation = 5.37

Min:

The minimum value of the given attribute. Here Min is **8**

Max:

The maximum value of the given attribute. Here Max is **20**

Try to separately min-max and Z-score normalize the values in the attribute.

Discretization

- Three types of attributes:
  - Nominal — values from an unordered set
  - Ordinal — values from an ordered set
  - Continuous — real numbers
- Discretization:
  - divide the range of a continuous attribute into intervals
  - Some classification algorithms only accept categorical attributes.
  - Reduce data size by discretization
  - Prepare for further analysis

Discretization

- Discretization
  - reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. Interval labels can then be used to replace actual data values.
- Concept hierarchies
  - reduce the data by collecting and replacing low level concepts (such as numeric values for the attribute age) by higher level concepts (such as young, middle-aged, or senior).

Discretization

- Methods
  - Binning
  - Histogram analysis
  - Clustering analysis