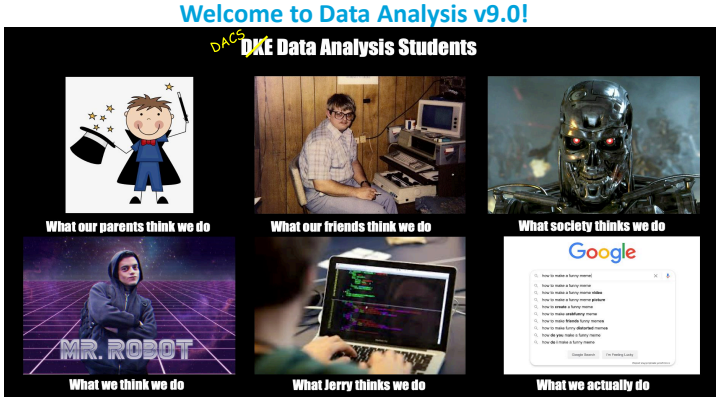
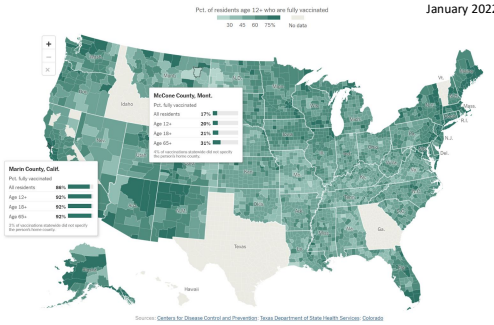
Welcome to Data Analysis v9.0!



Why?

NYTimes
Visualization:
January 2022

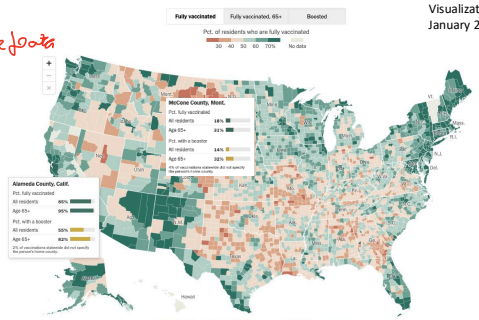


8

Why?

Belief is social

NYTimes
Visualization:
January 2023

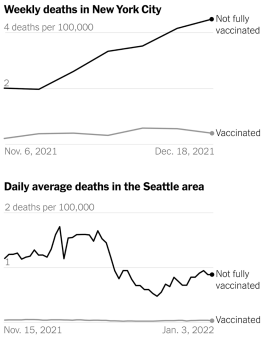


9

Why?

Data is a tool to navigate the complexity of our reality

Link



10

Why?

But data is easy to abuse and/or misinterpret

William Makis MD
@MakisMD

Stunning numbers from Denmark:

56% double vaccinated are catching 65% of "other variants" & 79% of Omicron

Most vulnerable group to Omicron BY FAR

25% boosted still catch 10% of Omicron cases, while unvaccinated catch 8.5%

This is worse than vaccine failure. This is damage.

1:08 AM · Dec 20, 2021 · Twitter Web App

805 Retweets 136 Quote Tweets

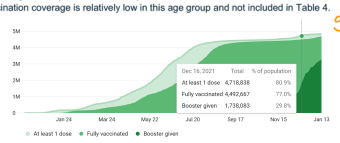
1,146 Likes

Table 4. Vaccination status for individuals ≥12 years infected with Omicron compared to other variants, data included in the table are from 22 November to 16 December 2021

Tabel 4. Vaccinationsstatus for personer ≥12 år med omikron-infektion sammenlignet med andre varianter i perioden fra og med 22. november 2021 til og med 16. december 2021

Vaccination status (12+ year olds)	Other variants (No. of cases)	Other variants (%)	Omicron (No. of cases)	Omicron (%)
Booster vaccinated	8,866	8.6	1,851	10.8
Fully vaccinated	67,034	65.3	13,546	79.0
Not vaccinated	23,492	22.9	1,454	8.5
Received first dose	3,216	3.1	304	1.8
Total	102,608	99.9	17,155	100.0

Individuals aged 5-11 years have recently been invited for COVID-19 vaccination, hence the vaccination coverage is relatively low in this age group and not included in Table 4.



11

Why?

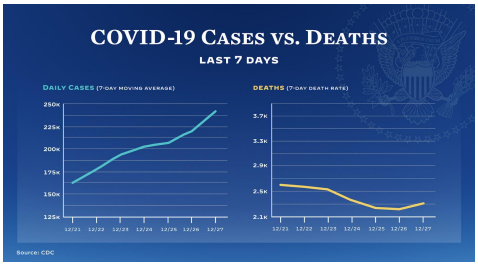
Even important entities communicate poorly

The White House
@WhiteHouse

Omicron cases are on the rise, but it's clear that vaccines and boosters are making a difference. Vaccines and boosters help prevent severe illness and death — if you haven't already, go get your vaccine and booster.

4:56 PM · Dec 29, 2021 · The White House

Actual Data: Link



12

Why?

Data science involves ***data-focused***, ***computational*** and ***inferential*** approaches to:

- Explore and understand the world (aka science)
- Address and solve practical challenges (aka engineering)

After this course, you should be able to take data and produce useful insights on the world's most challenging and ambiguous problems.

What?

- Convey basic skills about all steps in data analysis (DA)

Theory	Practice	Impact
• Key Statistical/ML concepts • Important metrics for evaluation of the model • Extracting insights from analysis of the models	• Implement algorithms using python libraries • Use online tools and DA resources • Handle different kinds of data	• Solve real-life problems using DA • Evaluate the social impact of DA • Communicate & visualize outcomes in a meaningful way

KEN3450
Data Analysis

Chapter I:
Introduction,
Exploratory Data Analysis &
Effective Visualizations

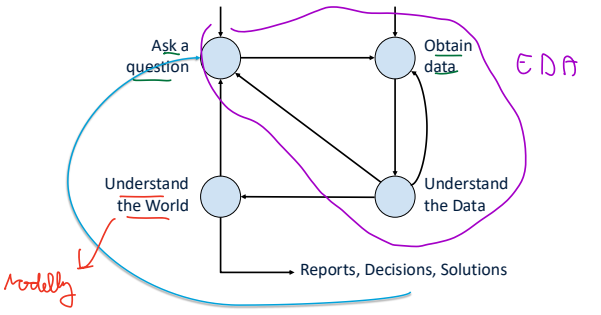
Gerasimos (Jerry) Spanakis, PhD
http://jerryspan.github.io

Maastricht University

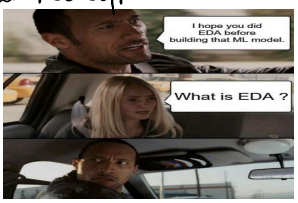
Learning goals

- Question the source of a dataset
- Practice simple pre-processing techniques
- Use EDA as a tool to further expand the analysis
- Understand why visualization/plotting is important
- Learn aspects that tend to make visualizes effective and ineffective
- Feel comfortable designing plots that best convey your message
- Gain experience in producing plots with Python

Data science lifecycle



This week: EDA & Visualizations



EDA key properties

More on the engineering side:
- Data sources (also relates to science)
- Data formats
- Data models (databases: not our focus)
- Data storage engines & processing (MLops: not our focus)

More on the science side:
- Granularity
- Scope
- Temporality
- Faithfulness

Data sources

- User generated
- Systems generated
- Internal databases:
  users, inventory, customer relationships
- Third-party data

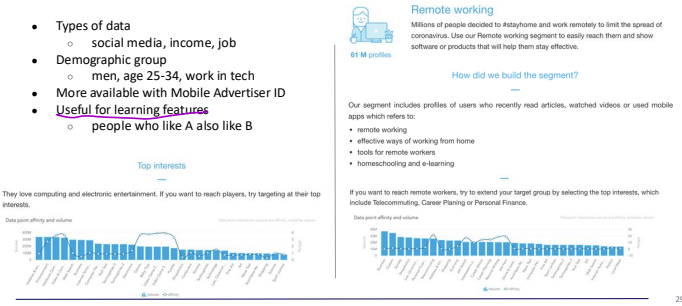
Data sources

Users generated data	Systems generated data
User inputs	Logs, metadata, predictions
Easily mal-formatted	Easier to standardize
Need to be processed ASAP	OK to process periodically (unless to detect problems ASAP)

Users' behavioral data (clicks, time spent, etc.) is often system-generated but is considered **user data**

Third-party data: creepy but fascinating

- Types of data
  - social media, income, job
- Demographic group
  - men, age 25-34, work in tech
- More available with Mobile Advertiser ID
- Useful for learning features
  - people who like A also like B

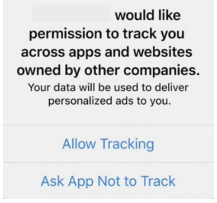


onaudience.com/audience-data

The end of tracking IDs ...

EDITORS' PICK | Jun 24, 2020, 12:38am EDT | 685,400 views

Apple Just Crippled IDFA, Sending An $80 Billion Industry Into Upheaval



In response, the state-backed China Advertising Association, which has 2,000 members, has launched a new way to track and identify iPhone users called CAID, which is being widely tested by tech companies and advertisers in the country.

TikTok wants to keep tracking iPhone users with state-backed workaround | Ars Technica

Data formats

	Format	Binary/Text	Human-readable	Example use cases
Row-major	JSON	Text	Yes	Everywhere
Row-major	CSV	Text	Yes	Everywhere
Column-major	Parquet	Binary	No	Hadoop, Amazon Redshift
Column-major	Avro	Binary	No	Hadoop
Column-major	Protobuf	Binary	No	Google, TensorFlow (TFRecord)
Column-major	Pickle	Binary	No	Python, PyTorch serialization

How to store multimodal data?

```
{'image': [[200,155,0], [255,255,255], ...], 'label': 'car', 'id': 1}
```

What are the access patterns
- How frequently the data will be accessed?
- Which hardware will we use (e.g. complex ML models run on CPU/GPU/TPU)

Row-major vs. column-major

Column-major:
- stored and retrieved column-by-column
- good for accessing features

Row-major:
- stored and retrieved row-by-row
- good for accessing samples

	Column 1	Column 2	Column 3
Sample 1	...	...	...
Sample 2	...	...	...
Sample 3	...	...	...

Row-major vs. column-major: DataFrame vs. ndarray

Pandas DataFrame: column-major
- accessing a row much slower than accessing a column and NumPy

```
# Get the column `date`, 1000 loops
%timeit -n1000 df["Date"]

# Get the first row, 1000 loops
%timeit -n1000 df.iloc[0]
```

```
1.78 µs ± 167 ns per loop (mean ± std. dev. of 7 runs, 1000 loops each)
145 µs ± 9.41 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

NumPy ndarray: row-major by default
- can specify to be column-based

```
df_np = df.to_numpy()
%timeit -n1000 df_np[0]
%timeit -n1000 df_np[:,0]
```

```
147 ns ± 1.54 ns per loop (mean ± std. dev. of 7 runs, 1000 loops each)
204 ns ± 0.678 ns per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

https://github.com/chiphuyen/just-pandas-things

Text vs. binary formats

	Text files	Binary files
Examples	CSV, JSON	Parquet
Pros	Human readable	Compact
Store the number *1000000*?	7 characters -> 7 bytes	If stored as int32, only 4 bytes

You can unload the result of an Amazon Redshift query to your Amazon S3 data lake in Apache Parquet, an efficient open columnar storage format for analytics. Parquet format is up to 2x faster to unload and consumes up to 6x less storage in Amazon S3, compared with text formats. This enables you to save data transformation and enrichment you have done in

Data models

- Describe how data is represented
  - Structured vs. unstructured
- For structured data: two main paradigms:
  - Relational model (similar to the SQL model)
  - NoSQL (gaves us the .json format)

```
# Document 1: harry_potter.json
{
    "Title": "Harry Potter",
    "Author": "J.K. Rowling",
    "Publisher": "Banana Press",
    "Country": "UK",
    "Sold as": [
        {"Format": "Paperback", "Price": "$20"},
        {"Format": "E-book", "Price": "$10"}
    ]
}

# Document 2: sherlock_holmes.json
{
    "Title": "Sherlock Holmes",
    "Author": "Conan Doyle",
    "Publisher": "Guava Press",
    "Country": "US",
    "Sold as": [
        {"Format": "Paperback", "Price": "$30"},
        {"Format": "E-book", "Price": "$15"}
    ]
}

# Document 3: the_hobbit.json
{
    "Title": "The Hobbit",
    "Author": "J.R.R. Tolkien",
    "Publisher": "Banana Press",
    "Country": "UK",
    "Sold as": [
        {"Format": "Paperback", "Price": "$30"},
    ]
}
```


Structured vs. unstructured data

Structured	Unstructured
Schema clearly defined	Whatever
Easy to search and analyze	Fast arrival (e.g. no need to clean up first)
Can only handle data with specific schema	Can handle data from any source
Schema changes will cause a lot of trouble	No need to worry about schema changes
Data warehouses	Data lakes

Data Storage Engines & Processing

Databases optimized for

Transactional processing (OLTP)

- Transactions: tweeting, ordering an Uber, uploading a new model, etc.
- Operations:
  - Insert when generated
  - Occasional update/delete

Analytical processing (OLAP)

- How to get aggregated information from a large amount of data?
  - e.g. what's the average ride price last month for riders at Uber?
- Operations:
  - Mostly SELECT

OLTP & OLAP are outdated terms

Decoupling storage & processing

- OLTP & OLAP: how data is stored is also how it's processed
  - Same data being stored in multiple databases
  - Each uses a different processing engine for different query types
- New paradigm: storage is decoupled from processing
  - Data can be stored in the same place
  - A processing layer on top that can be optimized for different query types

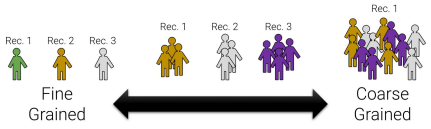
From ETL (Extract, Transform, Load) to ELT (Extract, Load, Transform)

Structured	-> unstructured (want more flexibility)	-> structured (tools & infra standardized)
ETL	-> ELT	-> ETL

Transform: the "meaty"/"juicy" part (the "heart" of EDA)

- cleaning, validating, transposing, deriving values, joining from multiple sources, deduplicating, splitting, aggregating, etc.

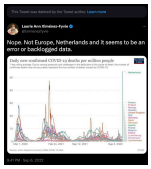
EDA - Granularity



- What does each record represent?
  - Purchases, persons, trees, …
- Do all records capture granularity on the same level?
- If the data is coarse, how were records aggregated?

EDA - Scope

- Does my data cover my area of interest?
  - E.g. I am studying crime data in Limburg but I only have data from Maastricht
- Is my data too broad?
  - E.g. *Sampling* protocol might lead to poor coverage
- Does my data cover the right time frame?
  - E.g. COVID19 test reporting



EDA - Temporality

- Data changes – when was the data collected?
- What is the meaning of time/date fields?
  - e.g. when the event "happened"
  - e.g. when data was collected vs. entered
- Time depends on "where"
  - Python library `datetime` is a life-saver
- Are there strange *null* values?
  - e.g. January 1st 1970 or 1900
- What about periodicity?

EDA - Faithfulness

- Do I trust this data?
- Does my data contain unrealistic or "incorrect" values
  - Dates in the future for events in the past
  - Locations that don't exist
  - Negative counts
  - Large outliers
  - …
- Does my data violate obvious dependences?
  - Age and birthday don't match
  - …
- Was the data entered by hand?
  - Expect spelling errors
- Are there any obvious signs of data falsification?
  - Repeated names, fake looking email addresses

Common feature engineering operations

1. Handling missing values
2. Scaling
3. Discretization
4. Categorical features
5. Feature crossing

Handling missing values

- Not all missing values are equal
  - Missing not at random (MNAR)
  - Missing at random (MAR)
  - Missing completely at random (MCAR)

Handling missing values

Missing not at random – when a value is missing due to the value itself

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	($350,000?)		2	Engineer	Yes
5	35	B	($350,000?)	Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

Missing at random – when a value is missing due to another observed variable

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

Missing completely at random – there is no pattern to which values are missing

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Deletion – removing data with missing entries
- Imputation – filling missing fields with certain values

Many people prefer deletion not because it's better, but it's easier to do

Handling missing values

- Deletion
  - Column deletion – remove columns with too many missing entries
    - drawbacks – even if half the values are missing, the remaining data still potentially useful information for predictions
    - e.g. even if over half the column for 'Marital status' is missing, marital status is still highly correlated with house purchasing
  - Row deletion

Marital status
Married
Single
Single

Handling missing values

- Deletion
  - Column deletion
  - Row deletion

Handling missing values

- Row deletion
  - Good for: data missing completely at random (MCAR) and few values missing

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1	39	A	150,000	Married	1	Engineer	No
2	27	B	50,000	Single	0	Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	75,000	Married	2	Engineer	Yes
5	35	B	35,000	Single	0	Doctor	Yes
6	32	A	50,000	Married	0	Teacher	No
7	33	B	60,000	Single	2	Teacher	No
8	20	B	10,000	Single	1	Student	No

Handling missing values

- Row deletion
  - Bad when many examples have missing fields

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Row deletion
  - Bad for: missing values are not at random (MNAR)
  - Missing information is information itself

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B	($350,000?)		2	Engineer	Yes
5	35	B	($350,000?)	Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Handling missing values

- Row deletion
  - Bad for: missing data at random (MAR)
  - Can potentially bias data – we've accidentally removed all examples with gender 'A'

ID	Age	Gender	Annual income	Marital status	Number of children	Job	Buy?
1		A	150,000		1	Engineer	No
2	27	B	50,000			Teacher	No
3		A	100,000	Married	2		Yes
4	40	B			2	Engineer	Yes
5	35	B		Single	0	Doctor	Yes
6		A	50,000		0	Teacher	No
7	33	B	60,000	Single		Teacher	No
8	20	B	10,000			Student	No

Imputation

- Fill missing fields with certain values
  - Defaults
    - e.g. 0, or the empty string, etc.
  - Statistical measures – mean, median, mode
    - e.g. if a day in July is missing its temperature value, fill it with the median temperature in July

Depending on the dataset and the application, there might be a different solution applicable!

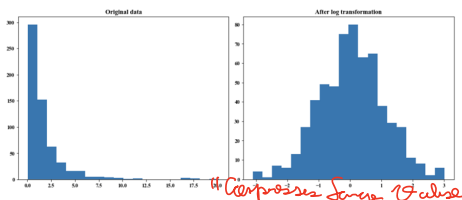
Scaling

scaling type	use case
min/max normalization	Any -- no assumptions about variables
z-score normalization	When variables follow a normal distribution
log scaling	When variables follow an exponential distribution

- scaling can be a common source of data leakage (more on this later in course)
- scaling variables requires global statistics

Log scaling

- Help with skewed data
- Often gives performance gain



Discretization

- Turning a continuous feature into a discrete feature (quantization)
- Create buckets for different ranges
  - Incorporate knowledge/expertise about each variable by constructing specific buckets
- Examples
  - Income
    - Lower income: x < $35,000
    - Middle income: $35,000 <= x < $100,000
    - High income: x >= $100,000
  - Age
    - Minors: x < 18
    - College: 18 <= x < 22
    - Young adult: 22 <= x < 30
    - 30 <= x < 40
    - 40 <= x < 65
    - Seniors: x >= 65

Encoding Categorical Features

- Example: you want to build a recommendation system for Amazon
  - There are over 2 million brands that we need to recommend

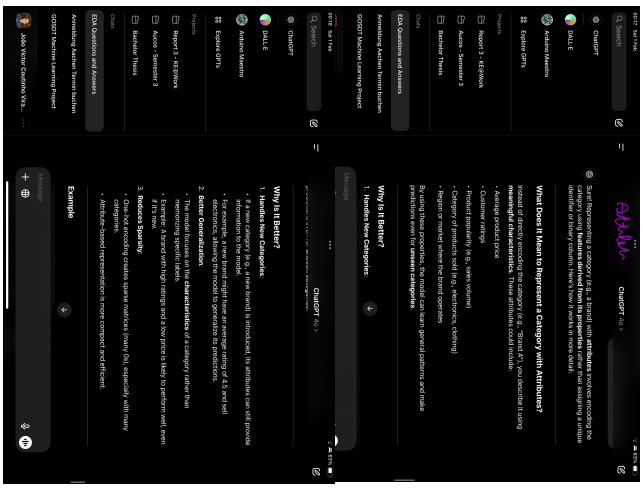
How do we encode the different brands/vendors?

Encoding Categorical Features

- one-hot encoding!
- → encode unseen brands with "UNKNOWN"
- → Group bottom 1% of brands and newcomers into "UNKNOWN" category
- Potential further issues: All newcomers are treated the same as unpopular brands in the platform
  - Solution 1: represent each category with its attributes
  - Solution 2: Hashing (widely used in industry, considered "hacky")

How do we handle a new brand that wants to join Amazon?

Problem! "UNKNOWN" was not seen during training, so none of the products in this category are being recommended



Feature Crossing

- Combine two or more features to create a new feature

Marriage	Single	Married	Single	Single	Married
Children	0	2	1	0	1
Marriage & children	Single, 0	Married, 2	Single, 1	Single, 0	Married, 1

- Helps models learn non-linear relationships between variables
- Warning – feature crossing can blow up your feature space
  - e.g. Feature A and B both have 100 categories → Feature A x B will have 10,000 categories
  - Apart from scalability issues, increases the risk of overfitting

Data types

- **Nominal:** categorical data, no ordering
  - Example: Pet: {dog, cat, rabbit, ...}
  - Operations: =, ≠
- **Ordinal:** categorical data, with ordering
  - Example: Rating: {1,2,3,4,5}
  - Operations: =, ≠, ≥, ≤, >, <
- **Interval:** numerical data, zero doesn't mean zero "quantity"
  - Example: Temperature Fahrenheit
  - Operations: =, ≠, ≥, ≤, >, <, +, -
- **Ratio:** numerical data, zero has meaning related to zero "quantity"
  - Example: Weight, Temperature Kelvin
  - Operations: =, ≠, ≥, ≤, >, <, +, -, /

Data types: Descriptive statistics (and plots)

- We are going to focus on the **type** and **dimensionality** of the underlying data
- (Not an exhaustive list)
  - 1D: bar chart, pie chart, histogram, boxplot
  - 2D: scatter plot, line plot, stacked box & whisker plots, heatmap
  - 3D+: scatter matrix, bubble chart

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```


Interval and ratio data: Descriptives

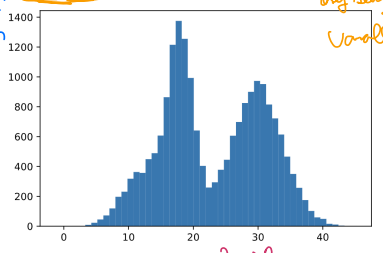
- Min, max, median, mean (or average)

```
In [12]:
# now use numpy to compute descriptive statistics for ages
print(len(ages))
print(np.min(ages))
print(np.max(ages))
print(np.mean(ages))
print(np.median(ages))
```

```
32561
17
90
38.58164675532078
37.0
```

Histogram

	Data
Nominal	✗
Ordinal	✗
Interval	✓
Ratio	✓

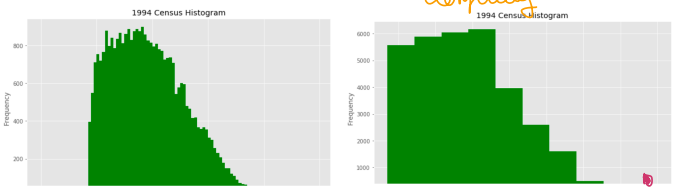


Data →
Count per bin (range of values)
→Plot with bar chart

```
plt.hist(x_values, density=True)
```

```
sns.histplot(data=df, x="x_column", stat="density")
```

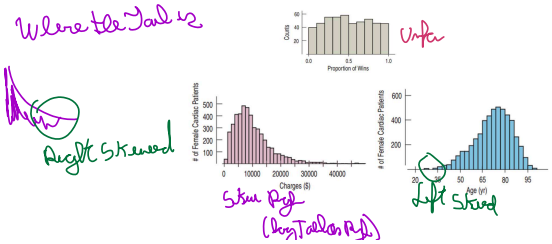

Different bins tell different stories...



- Note: We can use histogram to detect (potential) outliers

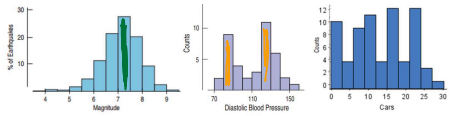
What do we look for in a histogram?

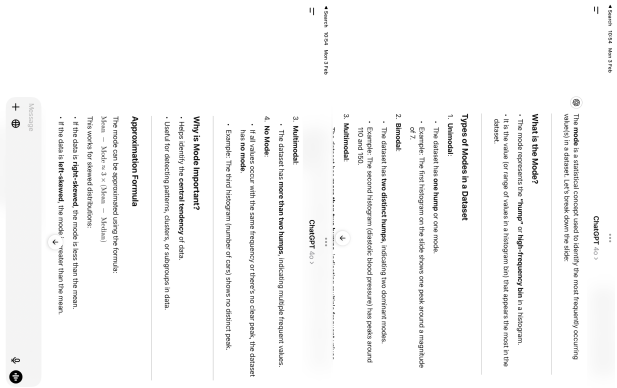
- Distribution (Uniform, Skewed Right / Skewed Left)

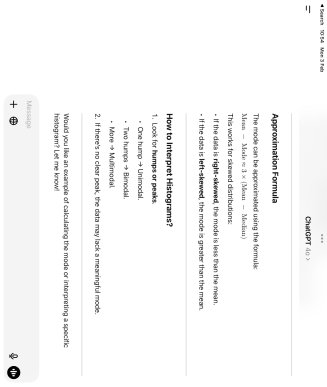


What do we look for in a histogram, cont.?

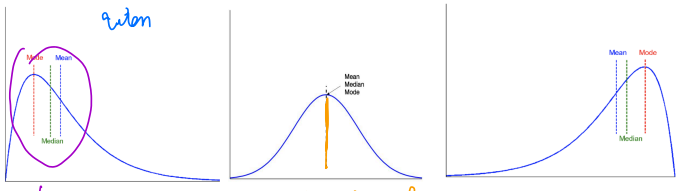
- Mode: A "hump" or a high-frequency bin
  - unimodal, bimodal, multimodal
  - approx. by *mean-mode = 3 x (mean-median)*







Intuition of mean, median, mode



Range, Variance, Standard Deviation

$$Range = Maximum - Minimum$$

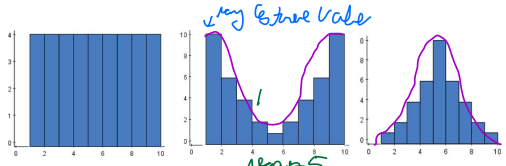
$$\text{Mean, } \mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\text{Variance} = \sigma^2 = \left(\frac{1}{n}\right)\sum_{i=1}^{n}(x_i - \mu)^2$$

$$\text{Std. dev.} = \sigma$$

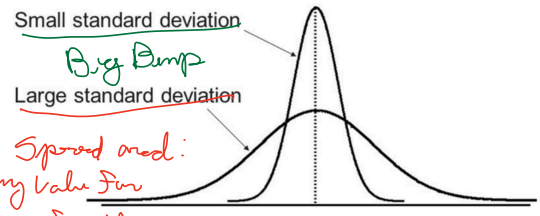
Histograms and standard deviation

Order the histograms below from smallest standard deviation to largest standard deviation.



A. A, B, C
B. B, A, C
C. C, A, B
D. A, C, B
E. B, C, A

Histograms and standard deviation



Boxplot


```
sns.boxplot(data=wb, y="Gross domestic product: % growth : 2016")
```


Nominal and ordinal data types

- Don't try to compute any numerical summary
- *Only* (relative) frequency tables

Class	Count
First	325
Second	285
Third	706
Crew	885

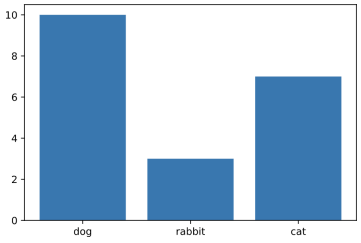
Class	%
First	14.77
Second	12.95
Third	32.08
Crew	40.21

```
In [20]:
print(race.value_counts()) #frequency table
print(race.value_counts()/len(race)) #relative frequency table
```

```
White                27816
Black                 3124
Asian-Pac-Islander    1039
Amer-Indian-Eskimo     311
Other                  271
```

Bar chart

	Data
Nominal	✓
Ordinal	✓
Interval	✗
Ratio	✗

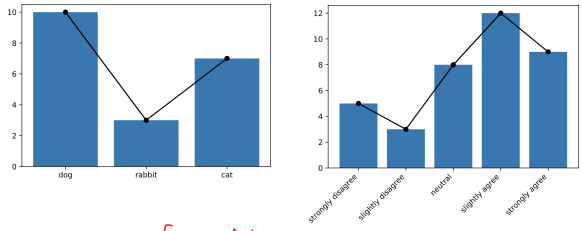



```
animals = wb["Animals"].value_counts()
plt.bar(animals.index, animals.values);

wb["Animals"].value_counts().plot(kind='bar')
```

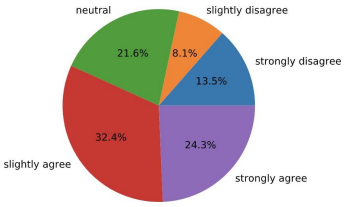

Bar chart (bad)

Don't use lines within a bar chart for categorial or ordinal features!



Pie chart (mostly bad)

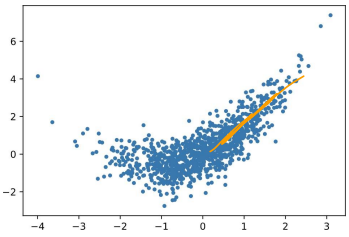
	Data
Nominal	✗
Ordinal	✗
Interval	✗
Ratio	✗



Scatterplot

	Dim 1	Dim 2
Nominal	✗	✗
Ordinal	✗	✗
Interval	✓	✓
Ratio	✓	✓

Why not ordinal data in first dimension?



```
plt.scatter(x_values, y_values)

sns.scatterplot(data=df, x="x_column", \
                y="y_column", hue="hue_column")
```


Correlation

- The coefficient of two variables x and y is given by

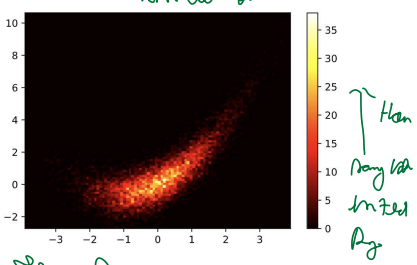
$$corr(x,y) = \frac{cov(x,y)}{\sigma_x \sigma_y} \qquad cov(x,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y).$$

- Assumptions, conditions, properties
  - Correlation assumes a **linear** relationship between x and y
  - Outliers can strongly affect the value of correlation
  - *corr > 0* → positive association, *corr<0* → negative association
  - -1 <= *corr* <=1
  - Interchanging x and y does not change the correlation
  - *Corr* has no units
  - Changing the units of x or y does not affect *corr*

http://guessthecorrelation.com/

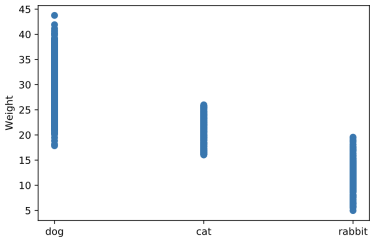
Heatmap (density or 2D histogram)

	Dim 1	Dim 2
Nominal	✗	✗
Ordinal	✗	✗
Interval	✓	✓
Ratio	✓	✓



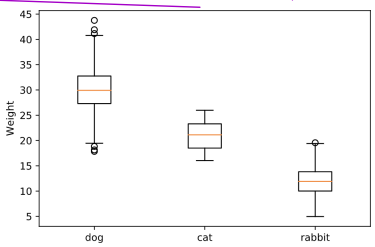
Scatterplot (very very bad)

	Dim 1	Dim 2
Nominal	✗	✗
Ordinal	✗	✗
Interval	✓	✓
Ratio	✓	✓



side-by-side boxplot

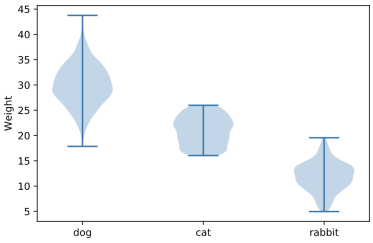
	Dim 1	Dim 2
Nominal	✓	✗
Ordinal	✓	✗
Interval	✗	✓
Ratio	✗	✓




```
sns.boxplot(data=wb, x="animals", y="weight");
```


side-by-side violin plot

	Dim 1	Dim 2
Nominal	✓	✗
Ordinal	✓	✗
Interval	✗	✓
Ratio	✗	✓



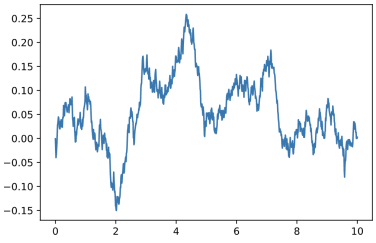

```
sns.violinplot(data=wb, x="animals", y="weight");
```


Line plot

When Dim 1 is **temporal**

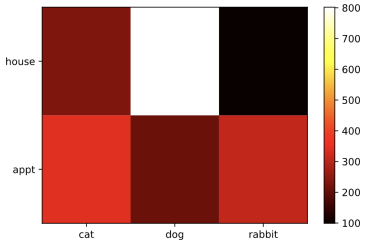
	Dim 1	Dim 2
Nominal	✗	✗
Ordinal	✗	✗
Interval	✓	✓
Ratio	✓	✓

Why not ordinal data in first dimension?



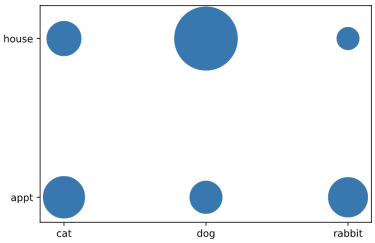
Heatmap (matrix)

	Dim 1	Dim 2
Nominal	✓	✓
Ordinal	✓	✓
Interval	✗	✗
Ratio	✗	✗



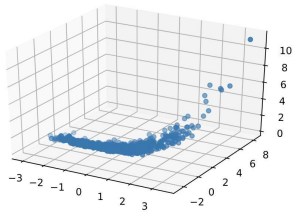
Bubbleplot

	Dim 1	Dim 2
Nominal	✓	✓
Ordinal	✓	✓
Interval	✗	✗
Ratio	✗	✗



3D scatterplot

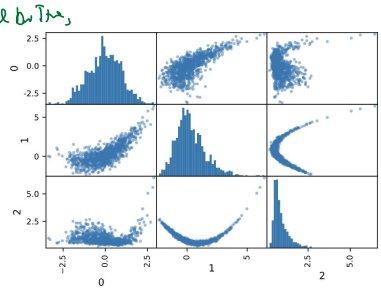
	Dim 1	Dim 2	Dim 3
Nominal	✗	✗	✗
Ordinal	✗	✗	✗
Interval	✗	✗	✗
Ratio	✗	✗	✗



Avoid 3D!

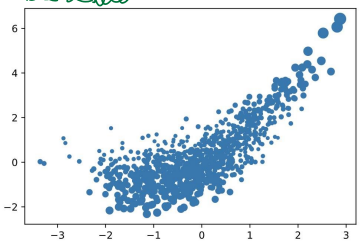
Scatterplot matrix

	Dim 1	Dim 2	Dim 3
Nominal	✗	✗	✗
Ordinal	✗	✗	✗
Interval	✓	✓	✓
Ratio	✓	✓	✓



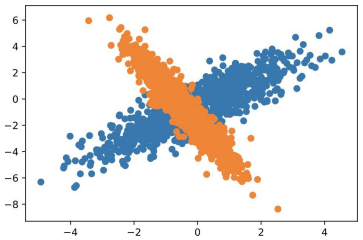
Bubbleplot ("light" 3D)

	Dim 1	Dim 2	Dim 3
Nominal	✗	✗	✗
Ordinal	✗	✗	✗
Interval	✓	✓	✓
Ratio	✓	✓	✓



Color scatterplot ("light" 3D)

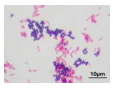
	Dim 1	Dim 2	Dim 3
Nominal	✗	✗	✓
Ordinal	✗	✗	✓
Interval	✓	✓	✗
Ratio	✓	✓	✗



HARD DONTs

- Don't make histograms for nominal/ordinal data!
- Don't compute numerical summaries for nominal/ordinal data!
  - e.g. mean student i-number does not anything
- Don't look for shape, spread etc. in a bar chart
- Choose an appropriate bin for histograms
- Beware of outliers
- Not everything might be possible to plot/visualize

What questions to ask?



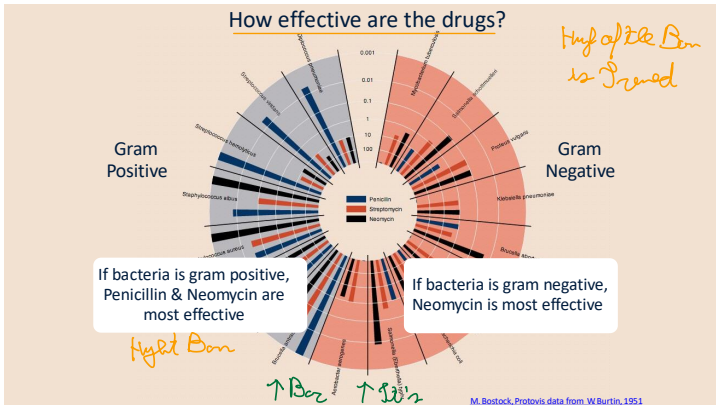
Genus, *Species*

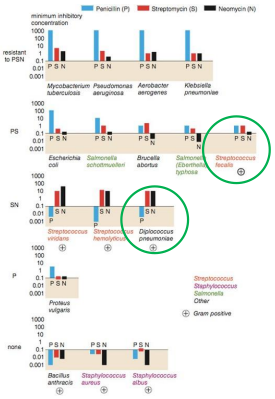
Table 1: Burtin's data.

Bacteria	Antibiotic: Penicillin	Streptomycin	Neomycin	Gram Staining
Aerobacter *aerogenes*	870	1	1.6	negative
Brucella *abortus*	1	2	0.02	negative
Brucella *anthracis*	0.001	0.01	0.007	positive
Diplococcus *pneumoniae*	0.005	11	10	positive
Escherichia *coli*	100	0.4	0.1	negative
Klebsiella *pneumoniae*	850	1.2	1	negative
Mycobacterium *tuberculosis*	800	5	2	negative
Proteus *vulgaris*	3	0.1	0.1	negative
Pseudomonas *aeruginosa*	850	2	0.4	negative
Salmonella (Ebertheila) *typhosa*	1	0.4	0.008	negative
Salmonella *schottmuelleri*	10	0.8	0.09	negative
Staphylococcus *albus*	0.007	0.1	0.001	positive
Staphylococcus *aureus*	0.03	0.03	0.001	positive
Streptococcus *fecalis*	1	1	0.1	positive
Streptococcus *hemolyticus*	0.001	14	10	positive
Streptococcus *viridans*	0.005	10	40	positive

Min.Inhibitory Concentration [ml/g]

W Burtin, (1951)





How resistant are the bacteria?

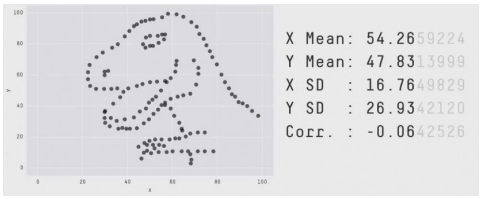
Not a streptococcus!
(realized ~30 years later)

Really a streptococcus!
(realized ~20 years later)

Wainer & Lysen, "That's funny..." American Scientist, 2009
Adapted from Brian Schmotzer

Why don't we just use numbers? Datasaurus

https://www.autodesk.com/research/publications/same-stats-different-graphs

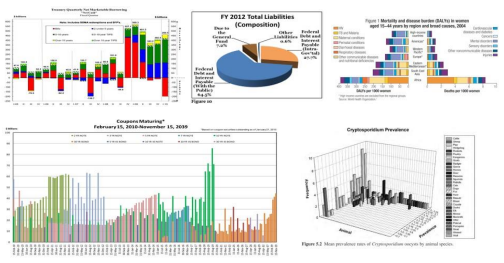


Visualization motivation

Visualizations help us to analyze and explore the data.
They help to:

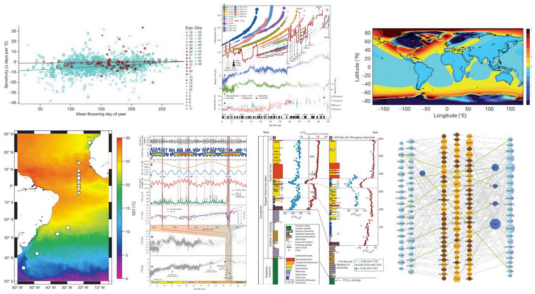
- Identify hidden patterns and trends
- Formulate/test hypotheses
- Communicate any modeling results
  - Present information and ideas succinctly
  - Provide evidence and support
  - Influence and persuade
- Determine the next step in analysis/modeling

But there are many bad examples...



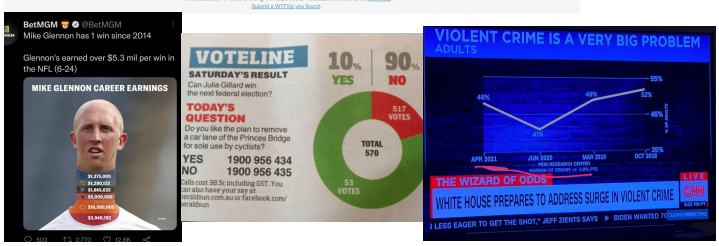
Sources: US Treasury and WHO reports

Even more...



Sources: Nature

http://viz.wtf



Effective visualizations

Simple but effe

Data Data

1. Have graphical integrity
2. Keep it simple…enough
3. Use sensible design
4. Use the right display

1. Graphical integrity

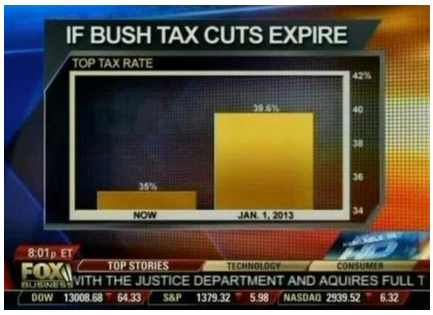
- Show data variation, not design variation
- Clear, detailed and thorough labeling and appropriate scales
- Plot all the data
- Be proportional

Should be Standard

↓

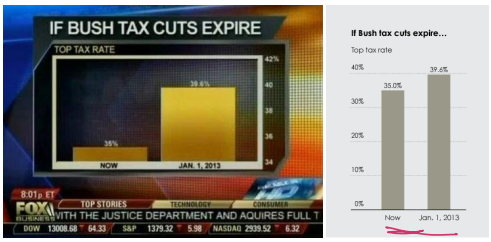
Statue may add Should still make Sense

What's the problem here?



Should Start at 0

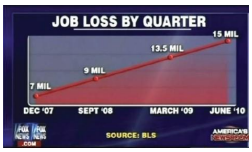
Problem: Scale distortions

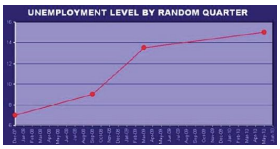


Stay at 0

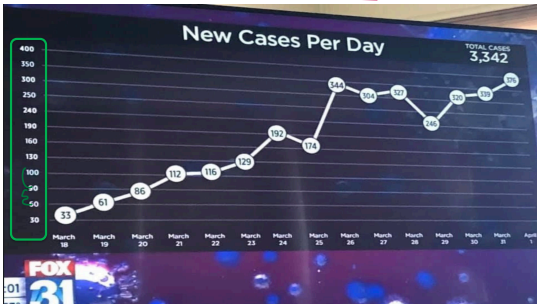
Another example

Problem: Scale distortions





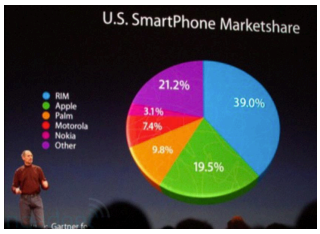
COVID-19 gave us some really nice plots…



Scale Spyg

https://qz.com/1872980/how-bad-covid-19-data-visualizations-mislead-the-public

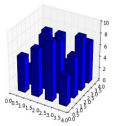
What are the problems here?

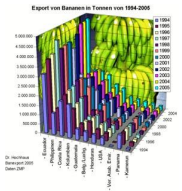


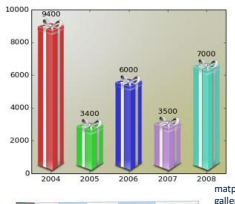
- Violates the Area Principle

- Add up the percents!

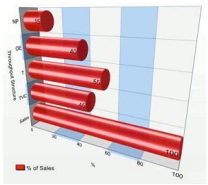
2. Keep it simple







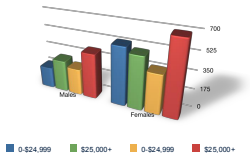
matplotlib gallery

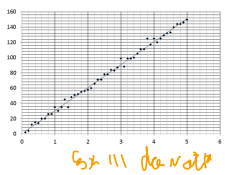


Excel Charts Blog

2. Keep it simple

- Size of the graphic effect should be directly proportional to the numerical quantities (“lie factor”)



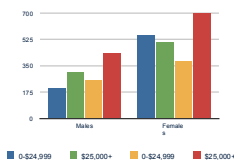


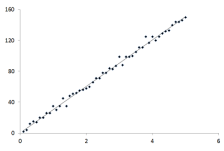
Est III da nalt

Maximize data-ink ratio

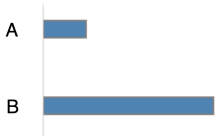
$$\text{Data-Ink Ratio} = \frac{\text{Data ink}}{\text{Total ink used in graphic}}$$

We can Improve



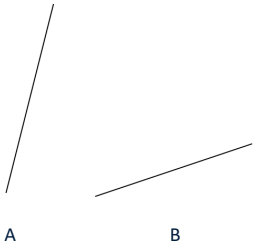


How much longer?



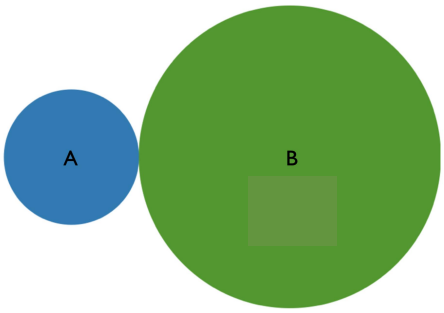
A. 2
B. 3
C. 4
D. 5
E. 6

How much steeper slope?



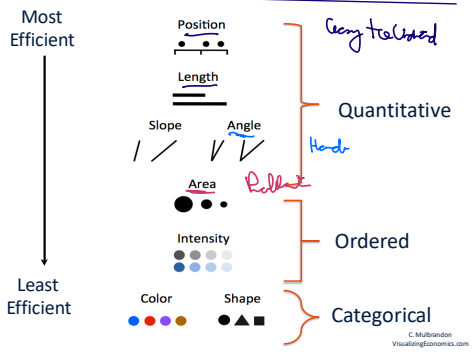
A. 2
B. 3
C. 4
D. 5
E. 6

How much larger area?

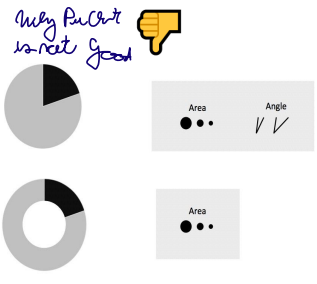


A. 2
B. 3
C. 4
D. 5
E. 6

30 years of visualization research in 1 slide



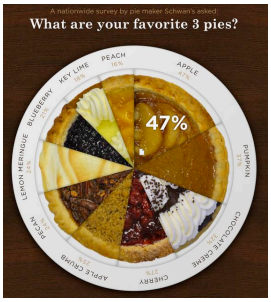
Most vs. less effective

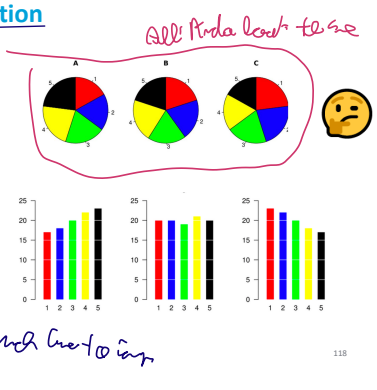


VisualizingEconomics.com

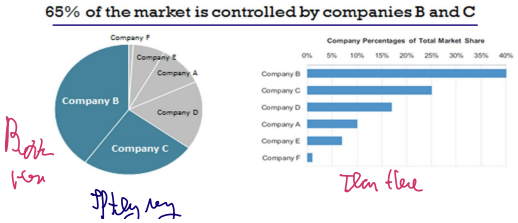
Pie charts have bad reputation

http://eagerpies.com/





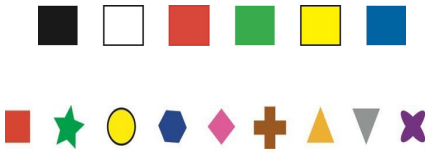
...but sometimes they are useful



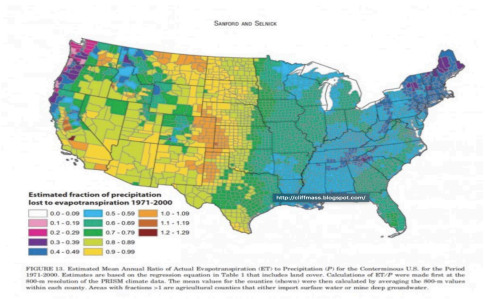
3. Use a sensible design

Use color strategically

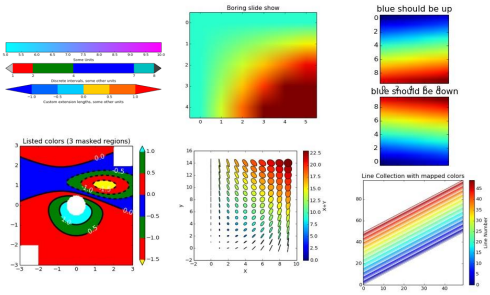
- For quantitative variables: Do not use color
- For ordinal data: Try varying luminance & saturation
- For categorical data: Do not use more than 5-8 colors



Least effective

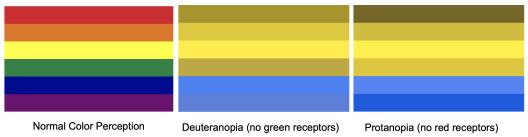


Avoid Rainbow Colors!



Color blindness

- 10% of males, 1% of females
- Most common: red-green weakness / blindness



- There are packages that take care of this!

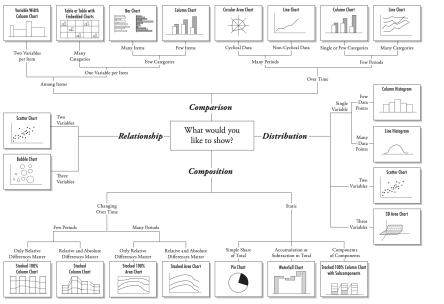
4. Choose the right display

What do you want your visualization to show about your data?

- **Distribution**: how a variable or variables in the dataset distribute over a range of possible values.
- **Relationship**: how the values of multiple variables in the dataset relate
- **Composition**: how the dataset breaks down into subgroups
- **Comparison**: how trends in multiple variable or datasets compare

4. Choose the right display

Chart Suggestions—A Thought-Starter



http://extremepresentation.typepad.com/blog/files/choosing_a_good_chart.p

Summary

- Reality check on your dataset
- Per variable:
  - Consistency check, missing data, outliers, noise…
  - Histograms, barplots, boxplots, etc.
  - Transformations/engineering possible
- Per pairs of variables (or more):
  - Associations (conditional tables)
  - Correlations (linearity check) for quantitative variables
- Tell your story via scientific, verbal and visual means