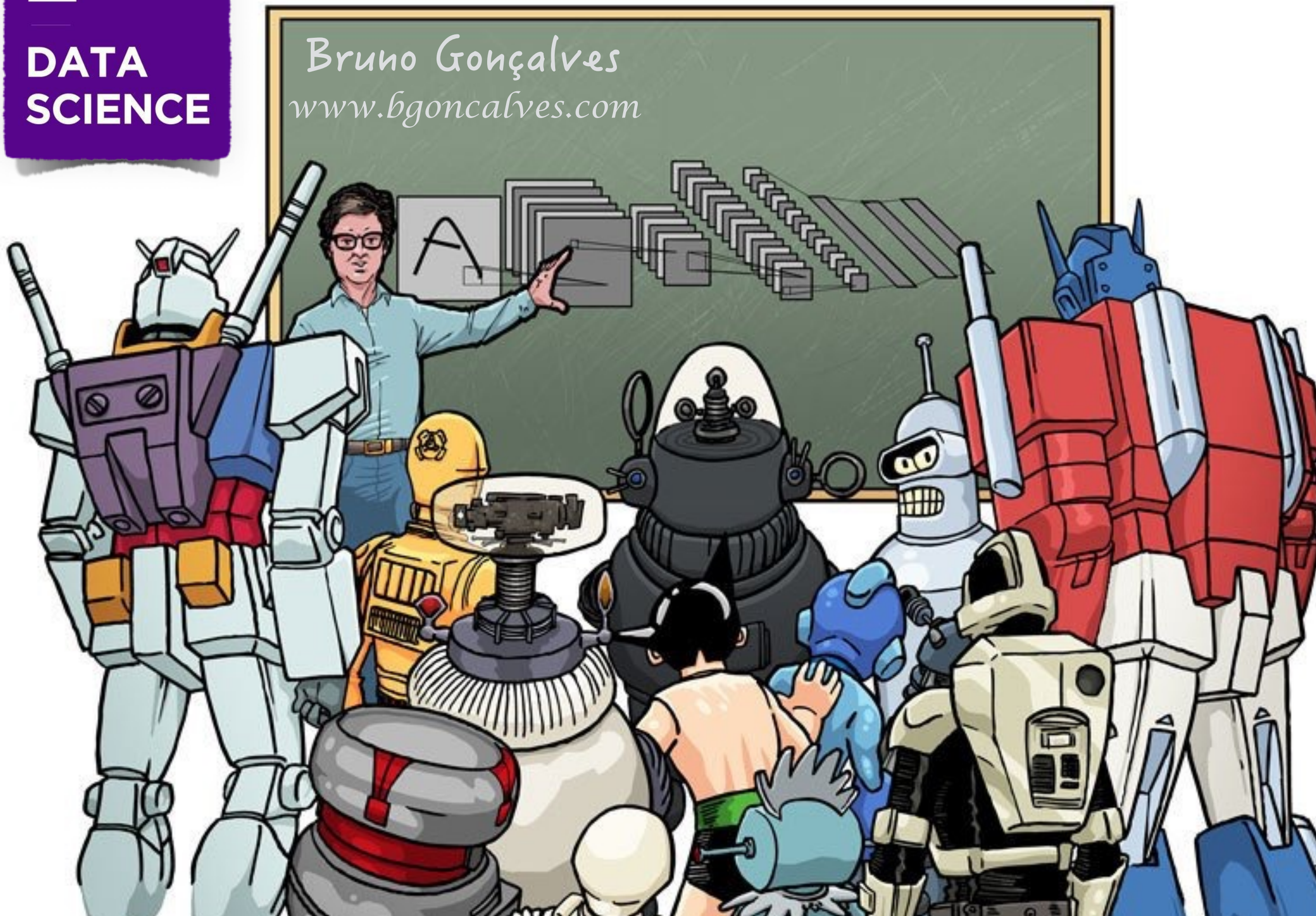
NYU

DATA SCIENCE

Machine(s) Learning and Data Science

Bruno Gonçalves
www.bgoncalves.com

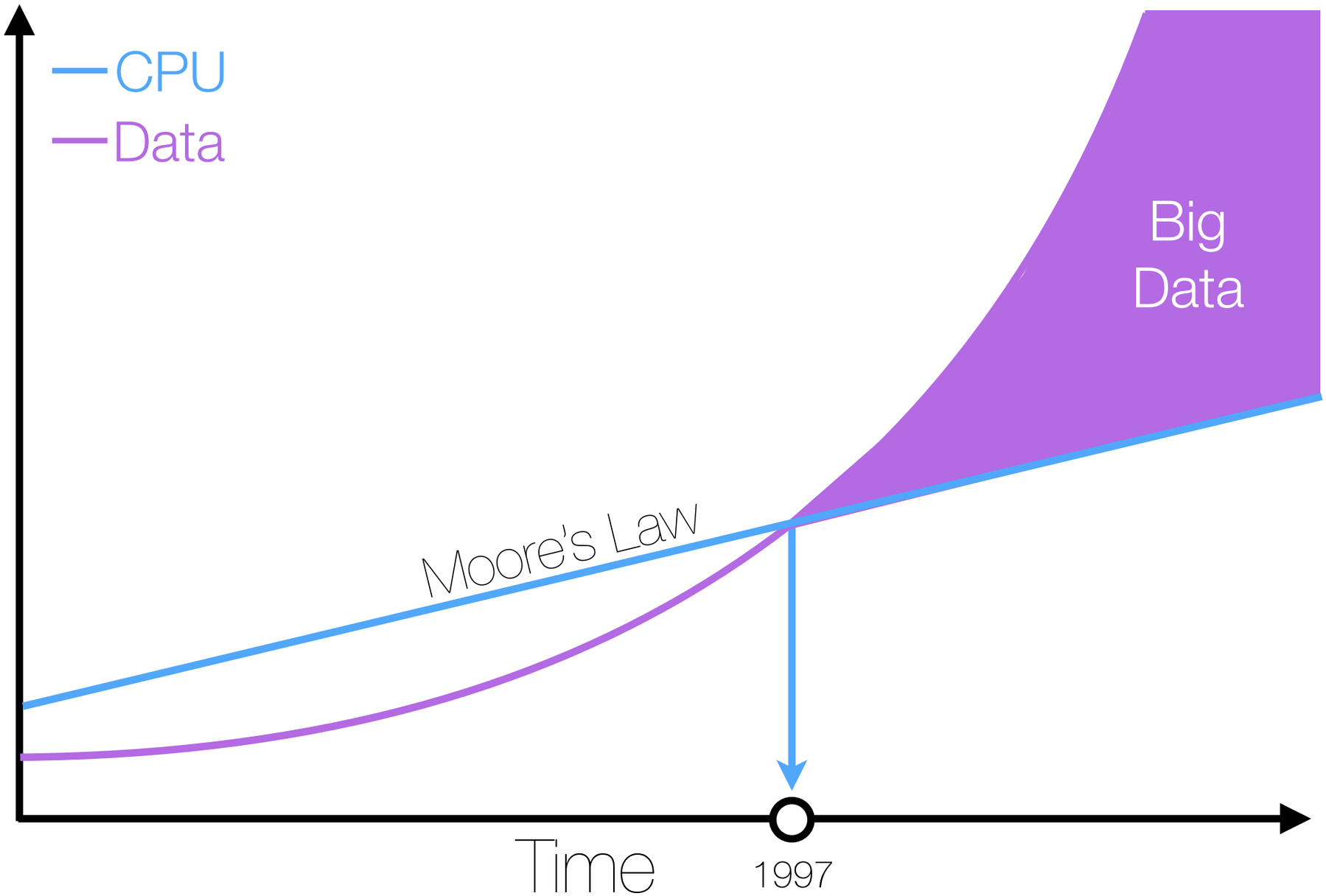


Requirements

https://bmtgoncalves.github.io/IFISC2017/

BIG DATA

Big Data



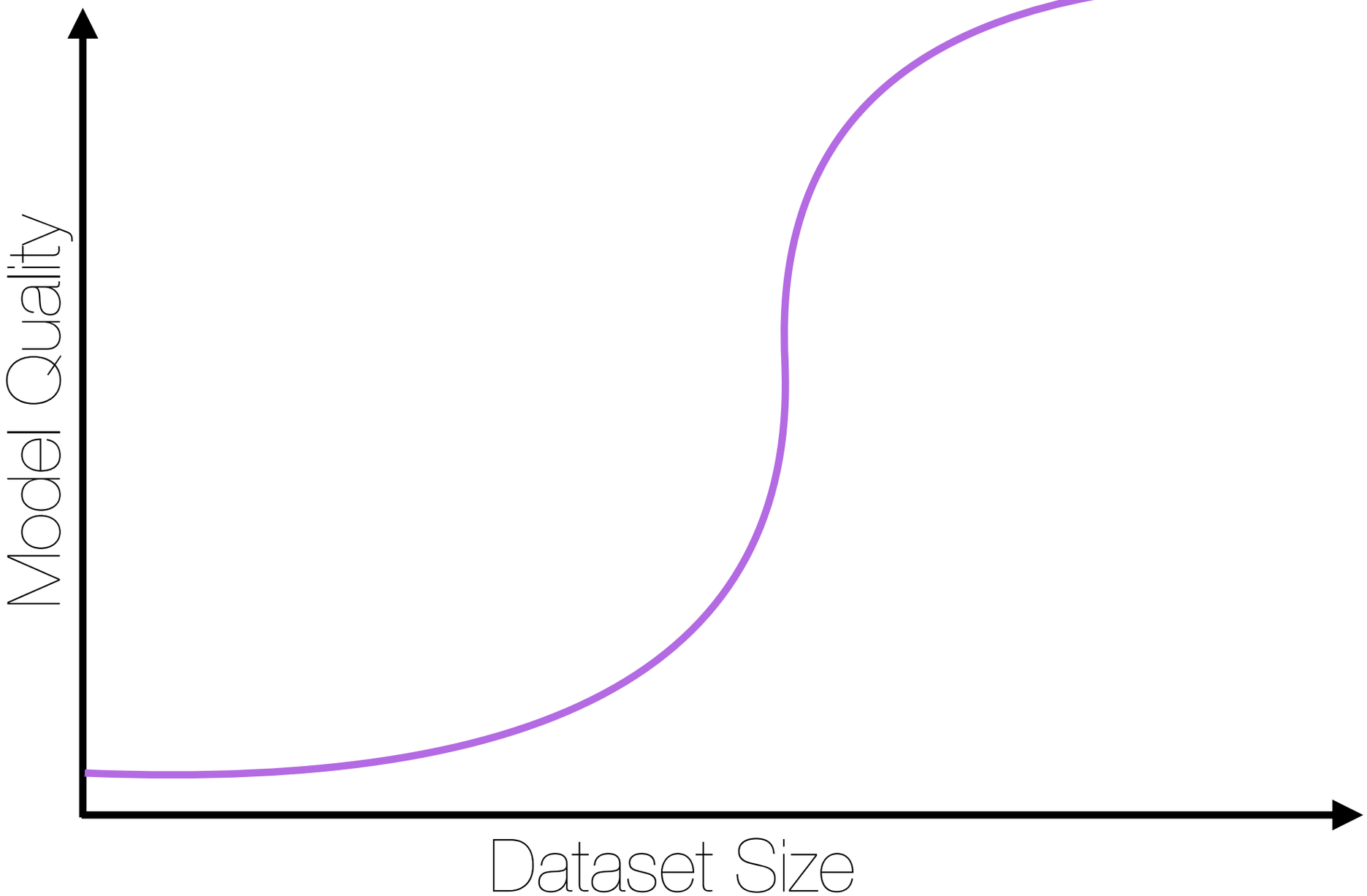
EXPERT OPINION

Contact Editor: **Brian Brannon,** bbrannon@computer.org

The Unreasonable Effectiveness of Data

Alon Halevy, Peter Norvig, and Fernando Pereira, *Google*

Big Data



https://www.wired.com/2008/06/pb-theory/

CHRIS ANDERSON SCIENCE 06.23.08 12:00 PM

THE END OF THEORY: THE DATA DELUGE MAKES THE SCIENTIFIC METHOD OBSOLETE

Illustration: Marian Bantjes

"**All models are wrong**, but some are useful."

Data Scientist:

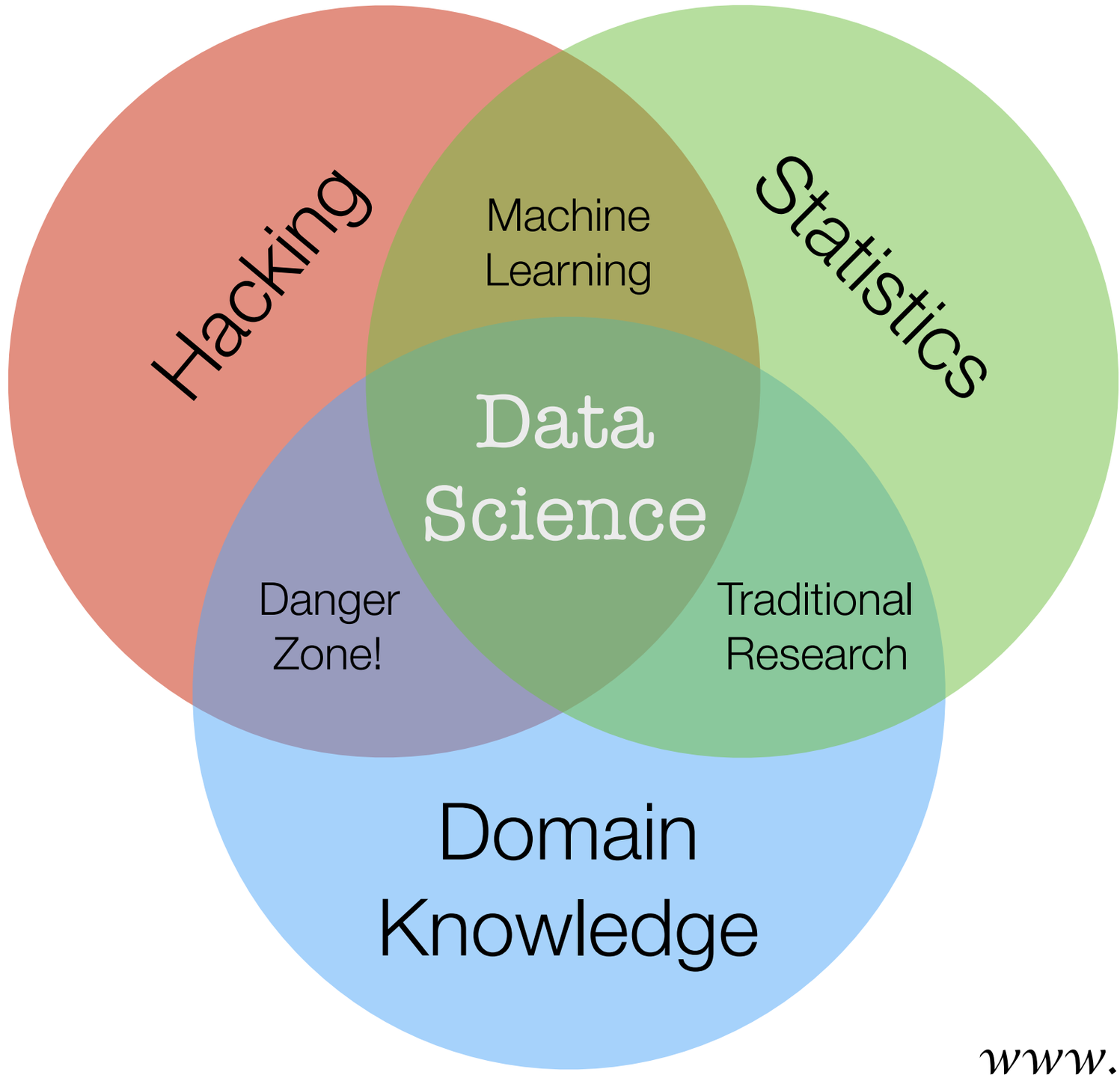
The Sexiest Job of the 21st Century

Meet the people who can coax treasure out of messy, unstructured data.

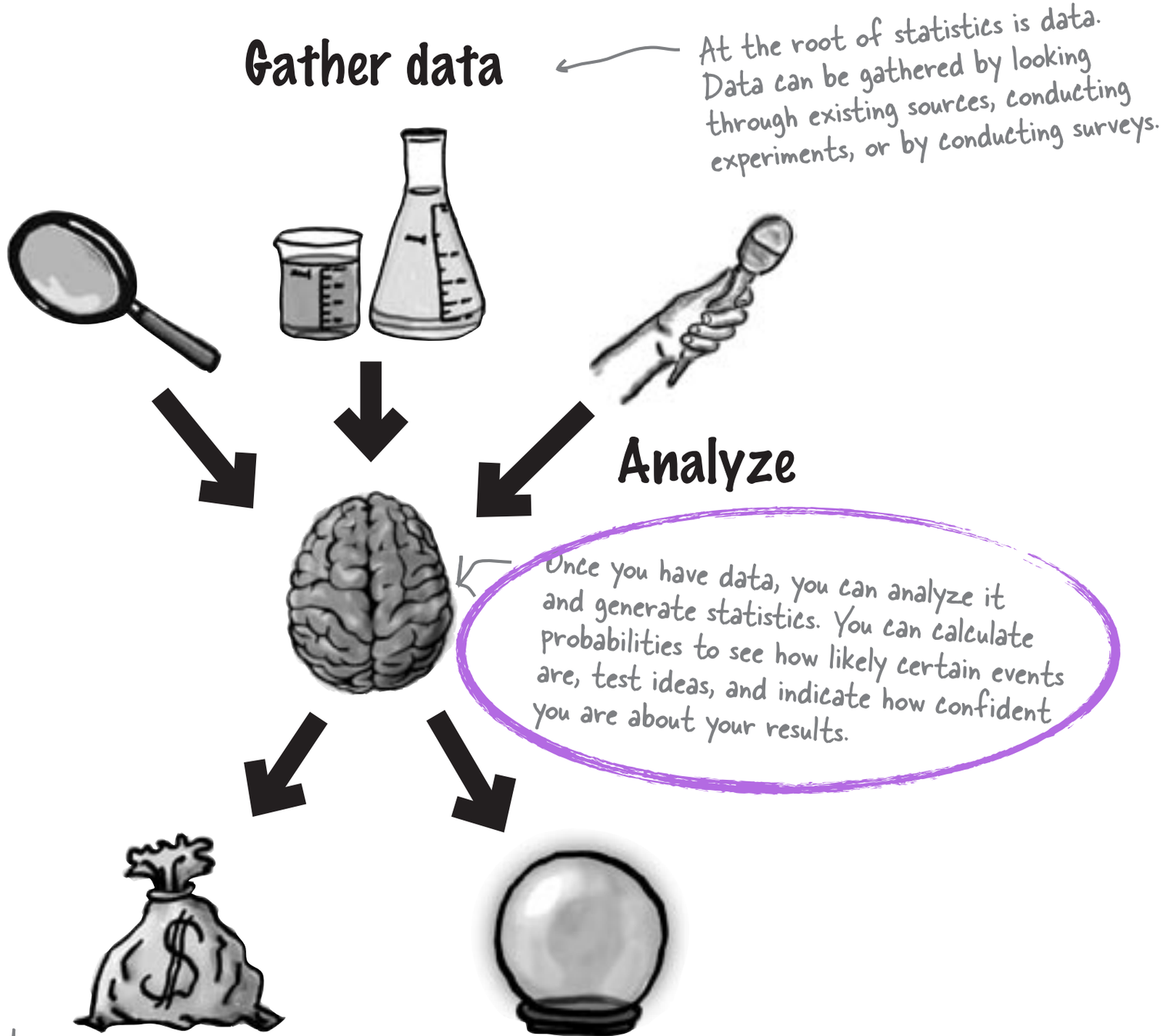
by Thomas H. Davenport and D.J. Patil

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know anyone. So you just stand in the corner sipping your drink—and you probably leave early."

Data Science



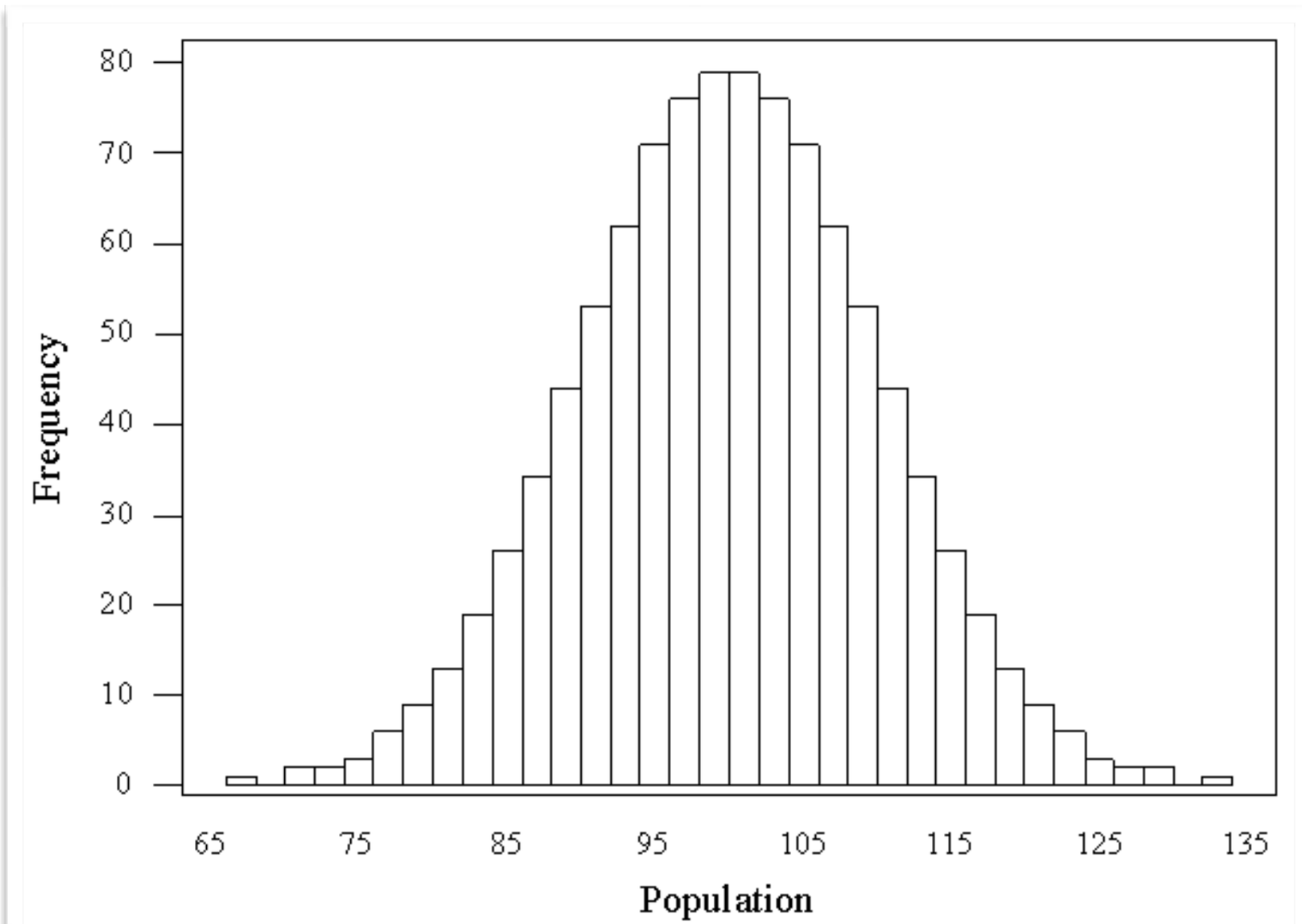
From Data To Information



Count!

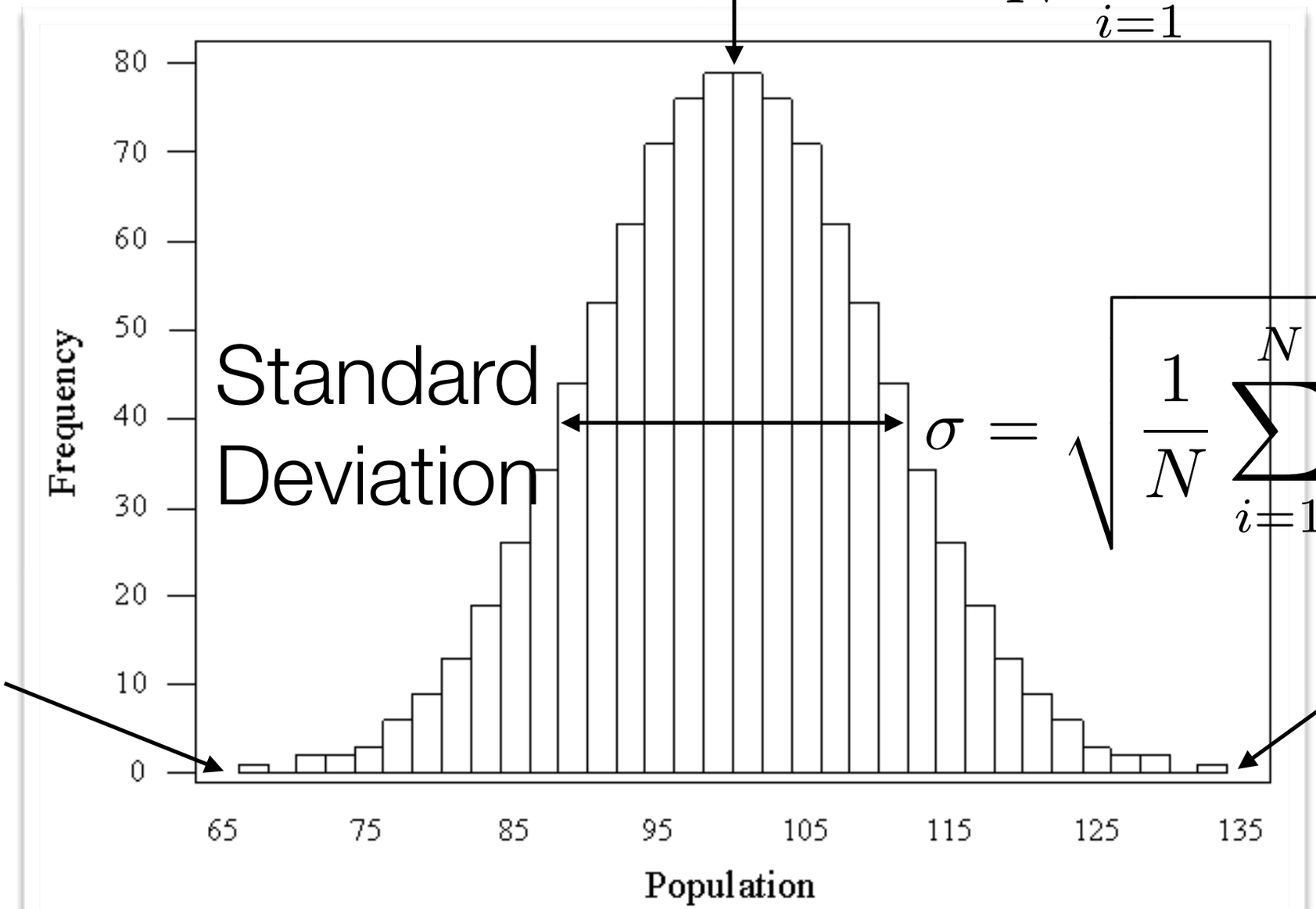
- How many items do we have?

"Zero is the most natural number"
(E. W. Dijkstra)



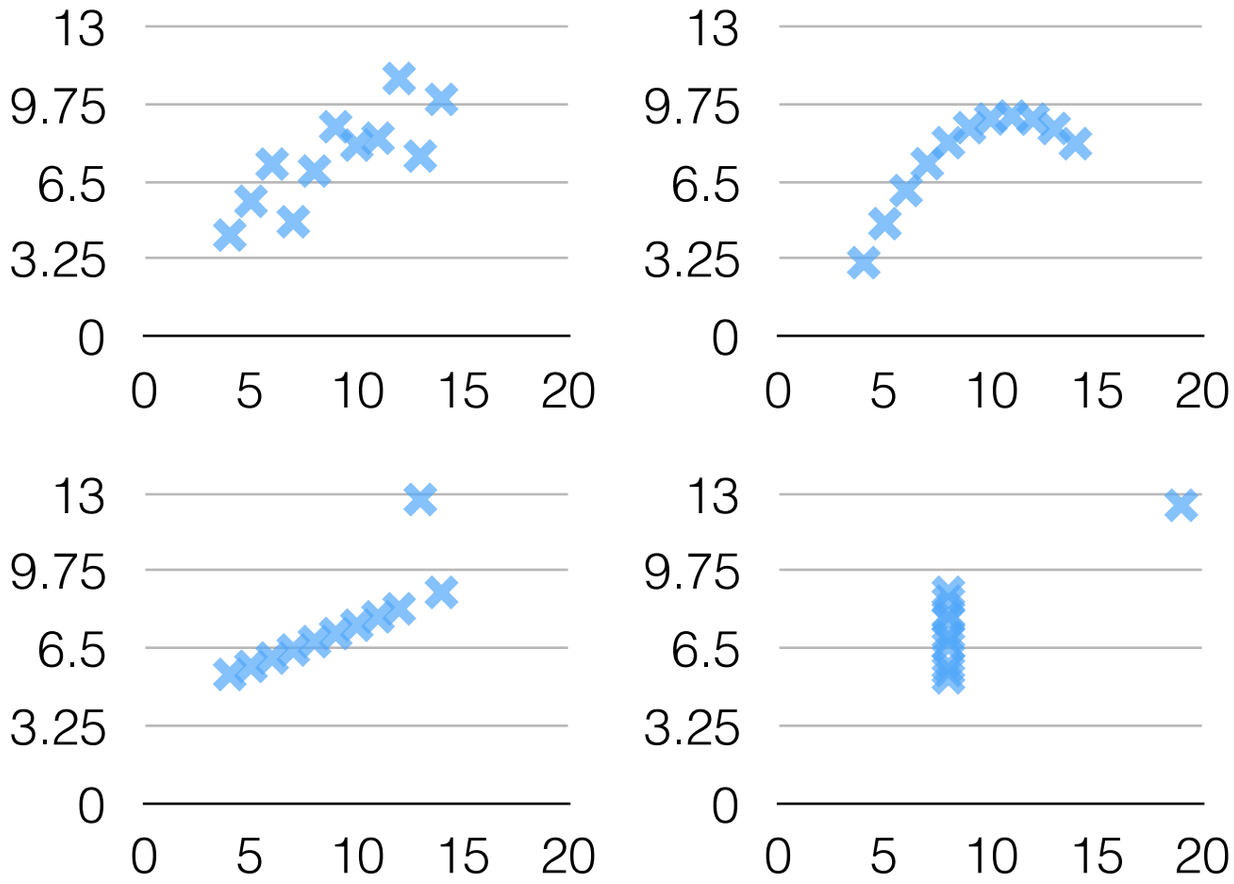
Descriptive Statistics

Mean $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$



Anscombe's Quartet

x1	y1	x2	y2	x3	y3	x4	y4
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89



μ_x	9
σ_x	11
μ_y	7.50
σ_y	~4.125
ρ	0.816
fit	y=3+0.5x

Outliers

- "Bill Gates walks into a bar and on average every patron is a millionaire…"

> Median - "the value that separates the lower 50% of the distribution from the higher 50%"

- …but the median remains the same"

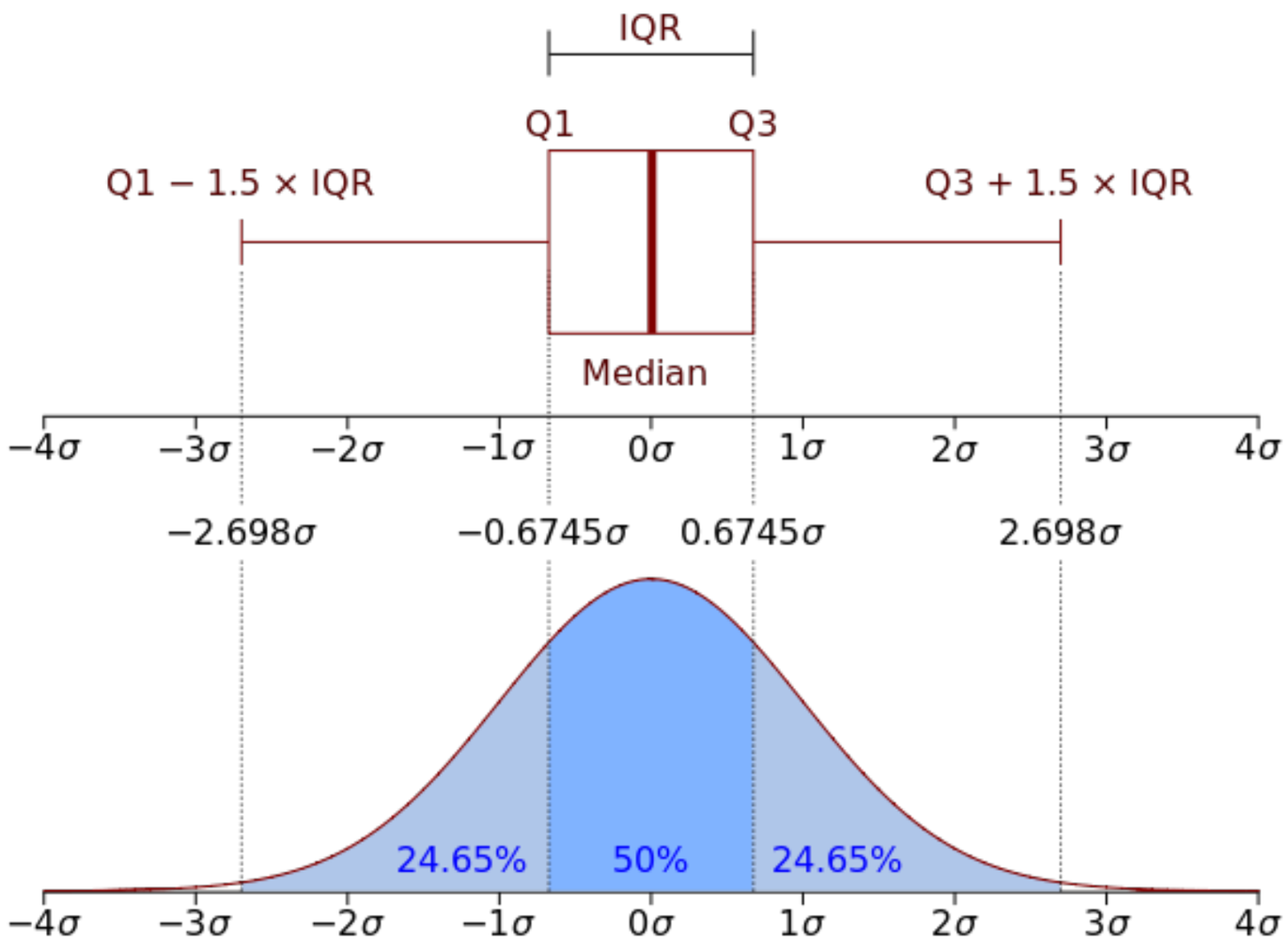
1 1 1 2 2 2 1000

- Mean = 144.14
  Median = 2

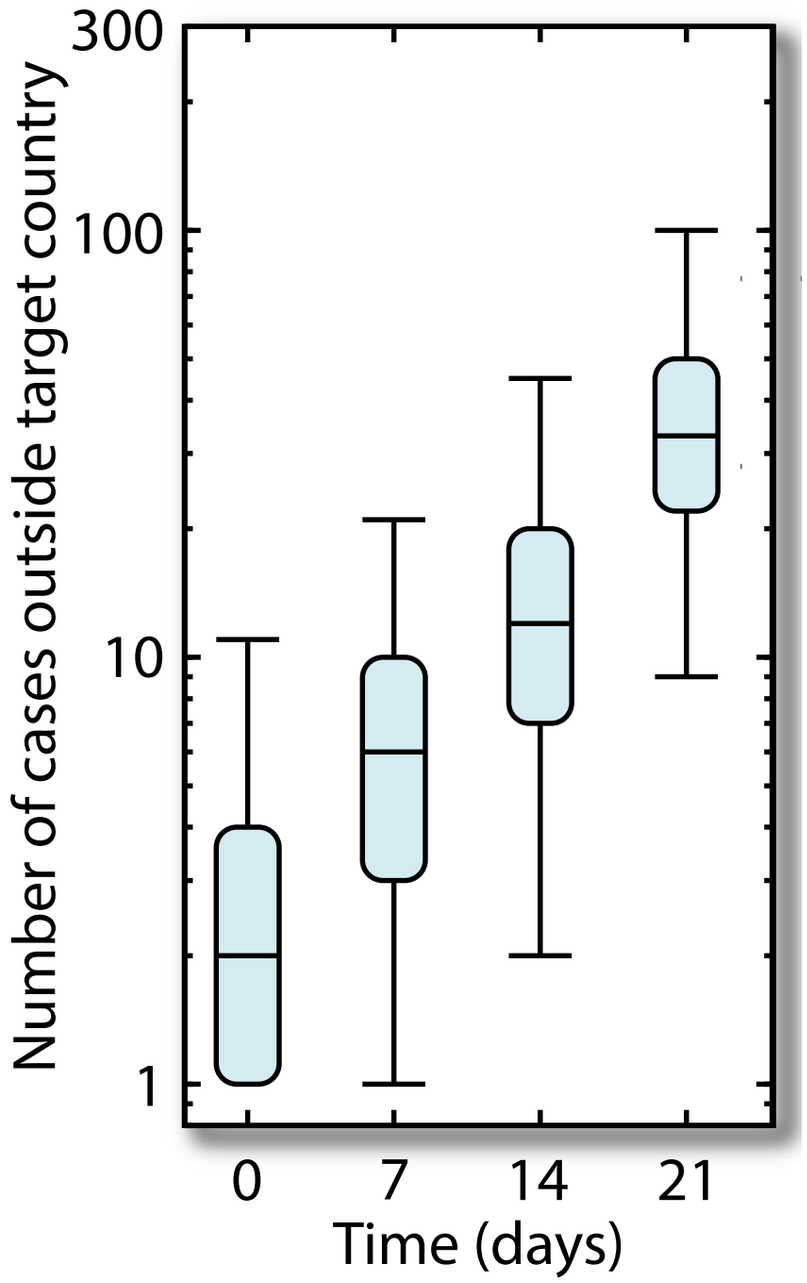
Quantiles

- Quantiles - Points taken at regular intervals of the cumulative distribution function

- Quartiles - Ranked set of points that divide the range in 4 equal intervals (25%, 50%, 75% quantiles)



Box and whiskers plot



- Show the variation of the data for each bin.
- More informative than just averages or medians.
- Useful to summarize experimental measurements, simulation results, natural variations, etc… when fluctuations are important

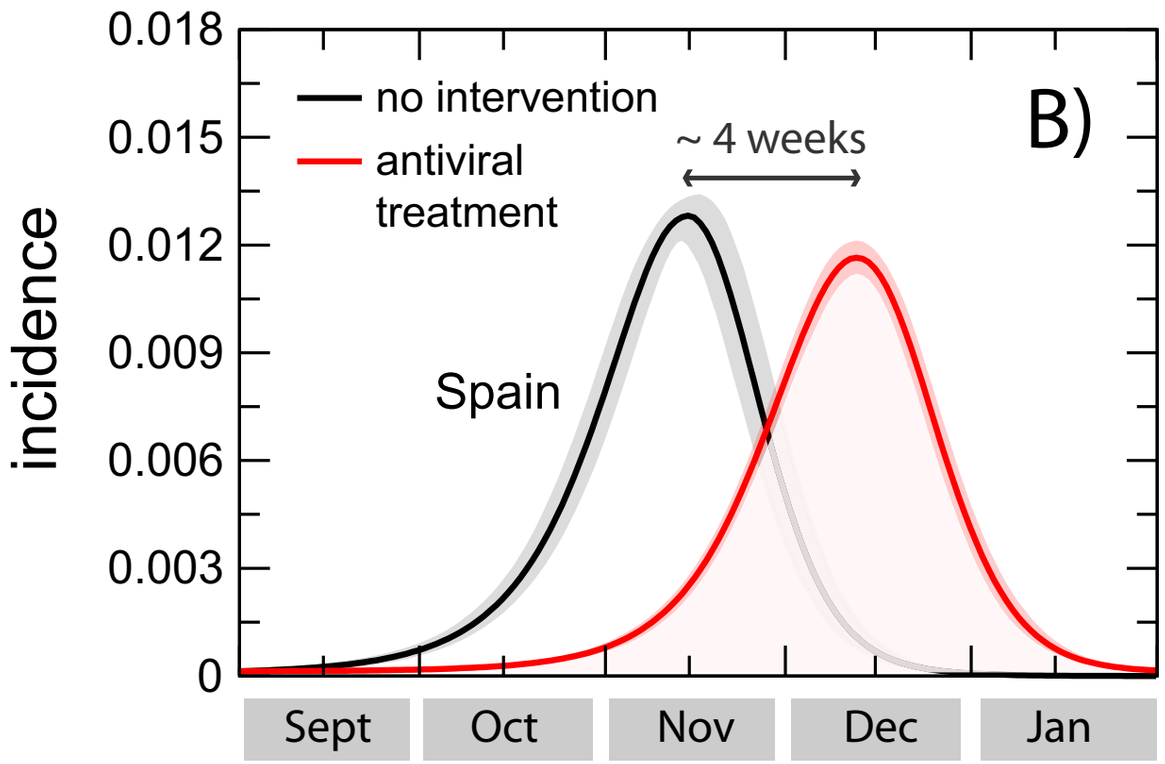
Reference Range

97.5% Percentile

95% RR — Median

2.5% Percentile

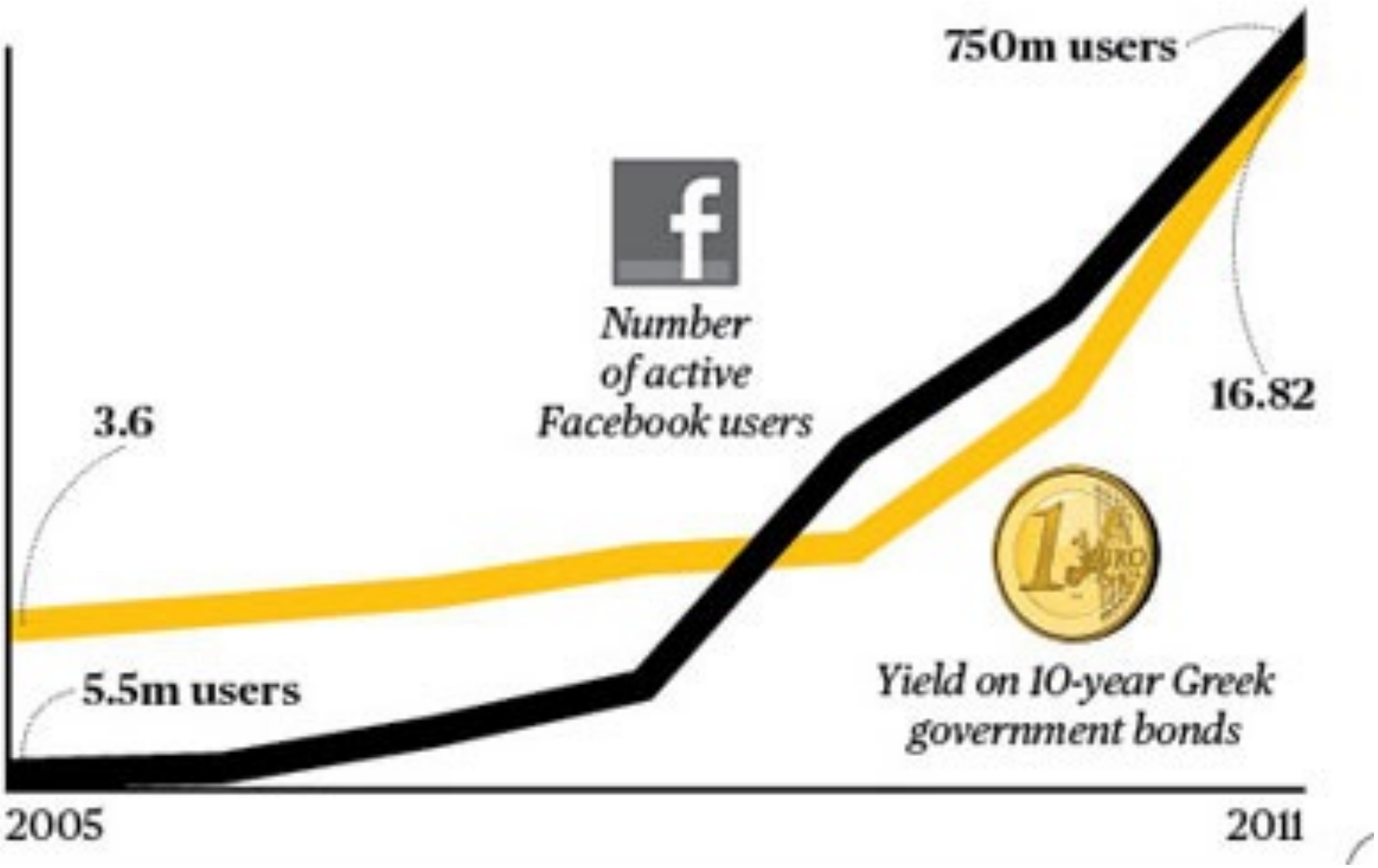
- Useful for continuous curves
- Indicates level of certainty
  "95% of the cases are in this range"



Tools For Statistical Analysis

Name	Advantages	Disadvantages	Open Source
R	Library support and Visualization	Steep learning curve	Yes
Matlab	Native matrix support, Visualization	Expensive, incomplete statistics support	No
Scientific Python	Ease and Simplicity	Heavy development	Yes
Excel	Easy, Visual, Flexible	Large datasets	No
SAS	Large Datasets	Expensive, outdated programming language	No
Stata	Easy Statistical Analysis		No
SPSS	Like Stata but more expensive and less flexible		

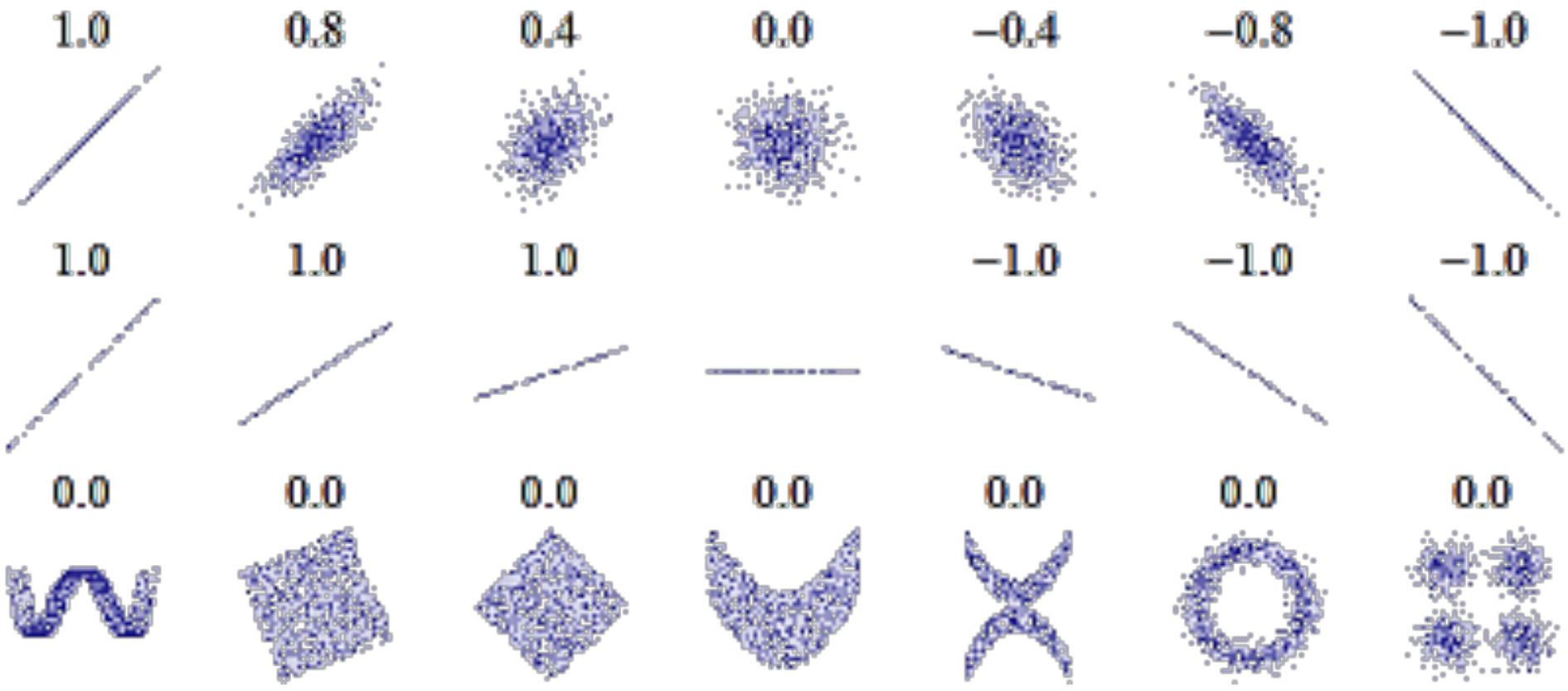
Correlations



Pearson (Linear) Correlation Coefficient

$$\rho = \frac{\sum_{i=1}^{N} (x_i - \mu_X)(y_i - \mu_Y)}{\sigma_X \sigma_Y}$$

- Does increasing one variable also increase the other?



R^2

- The square of the Person correlation between the data and the fit.

- The amount of variance of the data that is explained by the "model".

Spearman Rank Correlation

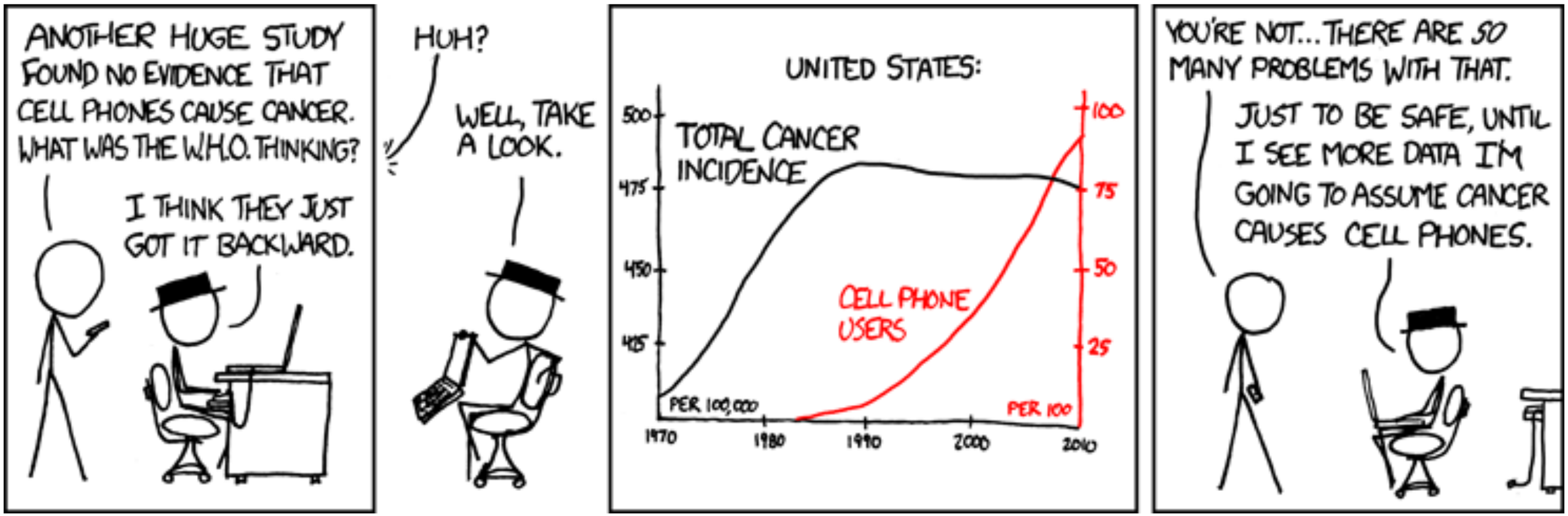
$$\rho = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N\left(N^2 - 1\right)}$$

- Equivalent to the Pearson Correlation Coefficient of the ranked variables

- d_i^2 squared difference in ranks

- less sensitive to outliers as values are limited by rank

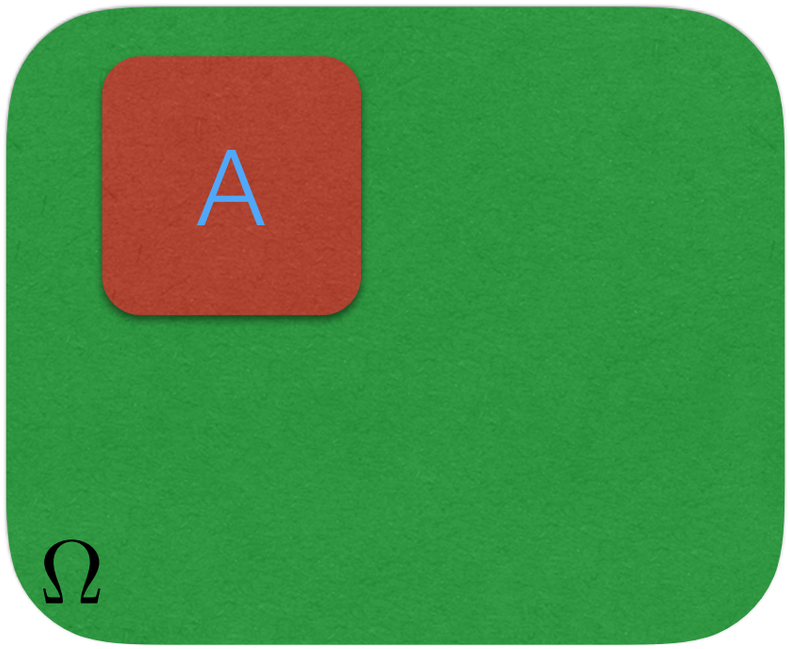
Causation


ANOTHER HUGE STUDY FOUND NO EVIDENCE THAT CELL PHONES CAUSE CANCER. WHAT WAS THE W.H.O. THINKING?
I THINK THEY JUST GOT IT BACKWARD.
HUH?
WELL, TAKE A LOOK.
UNITED STATES:
TOTAL CANCER INCIDENCE
CELL PHONE USERS
500
475
450
425
100
75
50
25
PER 100,000
PER 100
1970
1980
1990
2000
2010
YOU'RE NOT... THERE ARE SO MANY PROBLEMS WITH THAT.
JUST TO BE SAFE, UNTIL I SEE MORE DATA I'M GOING TO ASSUME CANCER CAUSES CELL PHONES.

Probability

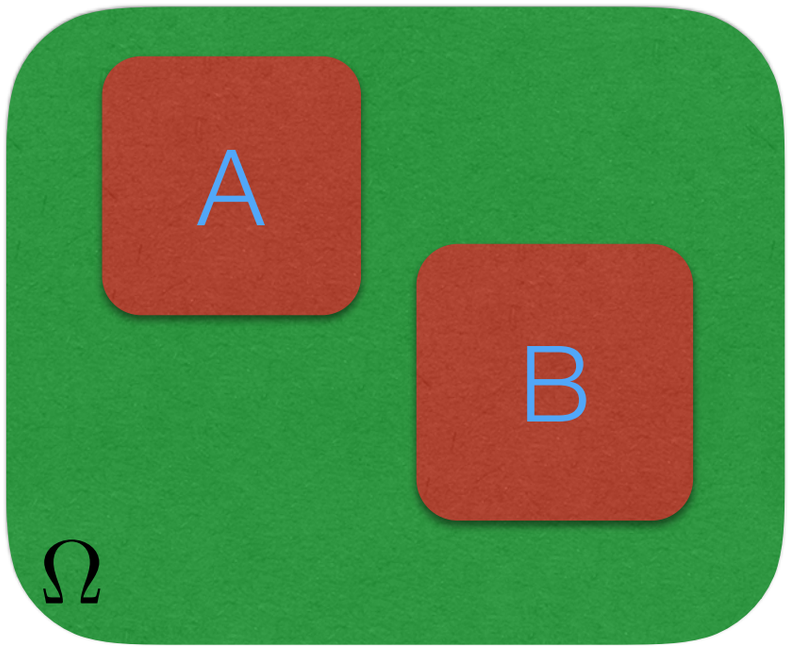
Probability

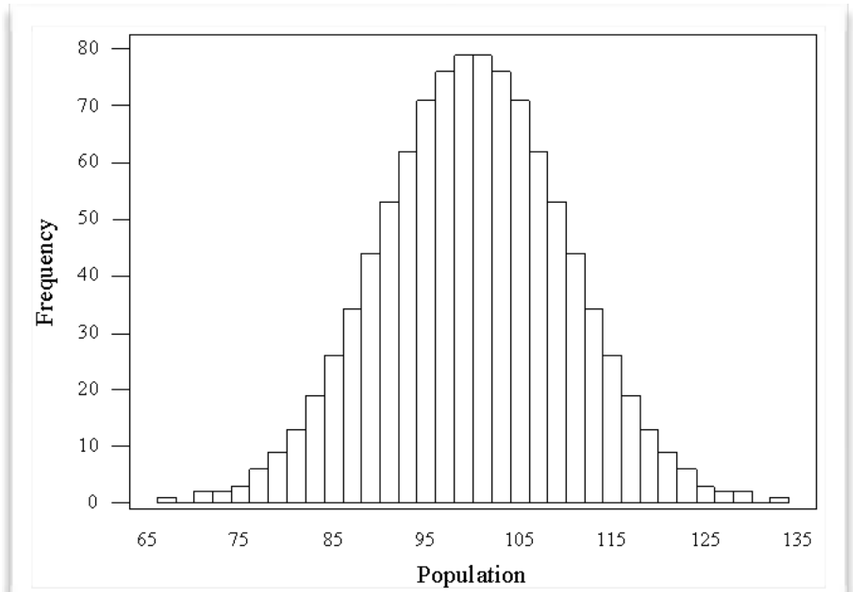
$$P(A) = \text{"Area" of A}$$
$$P(\Omega) = 1 \text{ (Normalization)}$$



Probability

$$P(A) = \text{"Area" of A}$$
$$P(\Omega) = 1 \text{ (Normalization)}$$



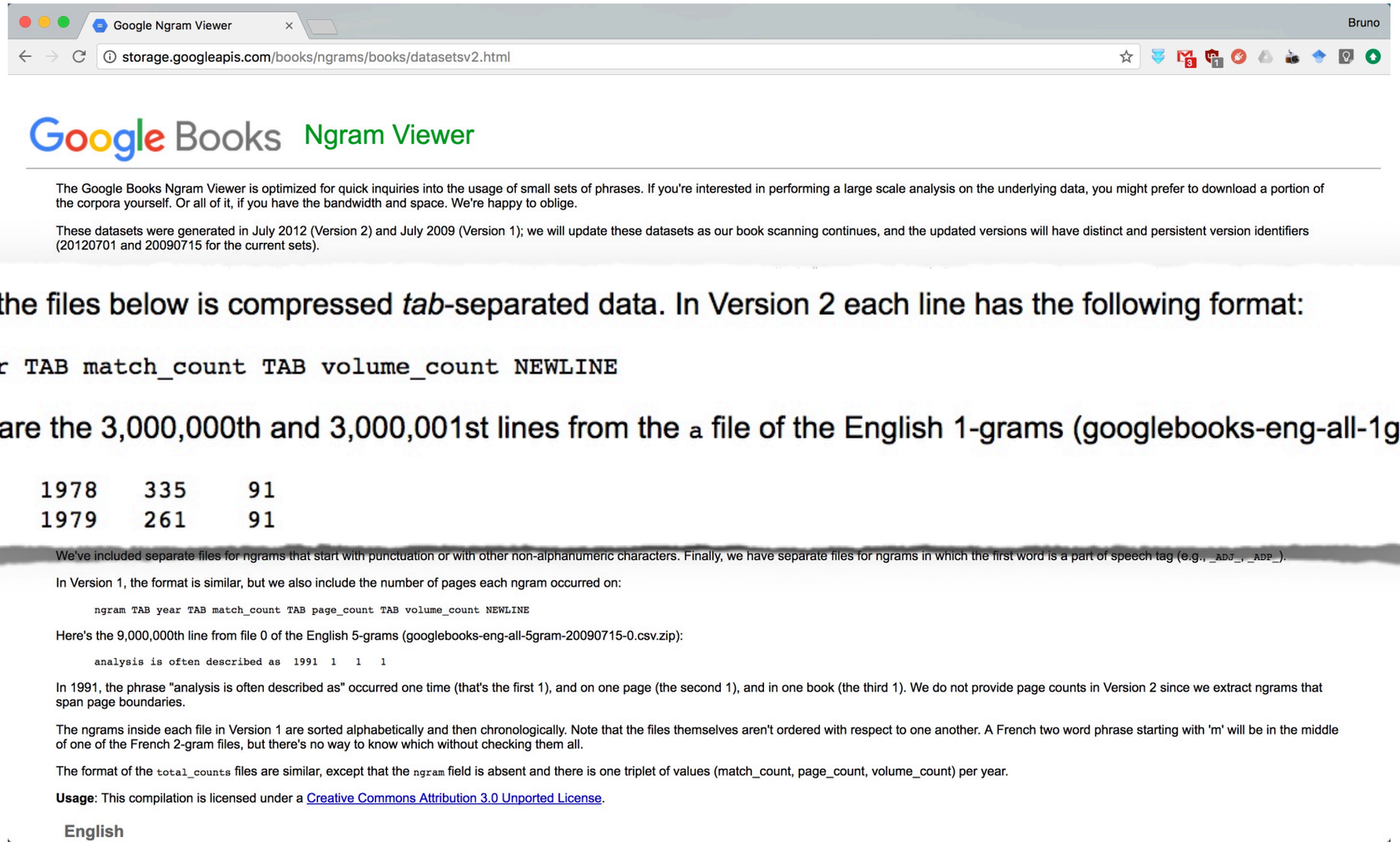


$$P(A \text{ or } B) = P(A) + P(B)$$

Character Probabilities

http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

As an example, let's calculate the probability of each letter occurring in the English language using Google Books 1-gram dataset.



Google Ngram Viewer

storage.googleapis.com/books/ngrams/books/datasetsv2.html

Bruno

Google Books Ngram Viewer

The Google Books Ngram Viewer is optimized for quick inquiries into the usage of small sets of phrases. If you're interested in performing a large scale analysis on the underlying data, you might prefer to download a portion of the corpora yourself. Or all of it, if you have the bandwidth and space. We're happy to oblige.

These datasets were generated in July 2012 (Version 2) and July 2009 (Version 1); we will update these datasets as our book scanning continues, and the updated versions will have distinct and persistent version identifiers (20120701 and 20090715 for the current sets).

File format: Each of the files below is compressed *tab*-separated data. In Version 2 each line has the following format:

```
ngram TAB year TAB match_count TAB volume_count NEWLINE
```

As an example, here are the 3,000,000th and 3,000,001st lines from the `a` file of the English 1-grams (googlebooks-eng-all-1gram-20120701-a.gz):

```
circumvallate   1978   335    91
circumvallate   1979   261    91
```

We've included separate files for ngrams that start with punctuation or with other non-alphanumeric characters. Finally, we have separate files for ngrams in which the first word is a part of speech tag (e.g., `_ADJ_`, `_ADP_`).

In Version 1, the format is similar, but we also include the number of pages each ngram occurred on:

```
ngram TAB year TAB match_count TAB page_count TAB volume_count NEWLINE
```

Here's the 9,000,000th line from file 0 of the English 5-grams (googlebooks-eng-all-5gram-20090715-0.csv.zip):

```
analysis is often described as  1991  1   1   1
```

In 1991, the phrase "analysis is often described as" occurred one time (that's the first 1), and on one page (the second 1), and in one book (the third 1). We do not provide page counts in Version 2 since we extract ngrams that span page boundaries.

The ngrams inside each file in Version 1 are sorted alphabetically and then chronologically. Note that the files themselves aren't ordered with respect to one another. A French two word phrase starting with 'm' will be in the middle of one of the French 2-gram files, but there's no way to know which without checking them all.

The format of the `total_counts` files are similar, except that the `ngram` field is absent and there is one triplet of values (match_count, page_count, volume_count) per year.

Usage: This compilation is licensed under a Creative Commons Attribution 3.0 Unported License.

English

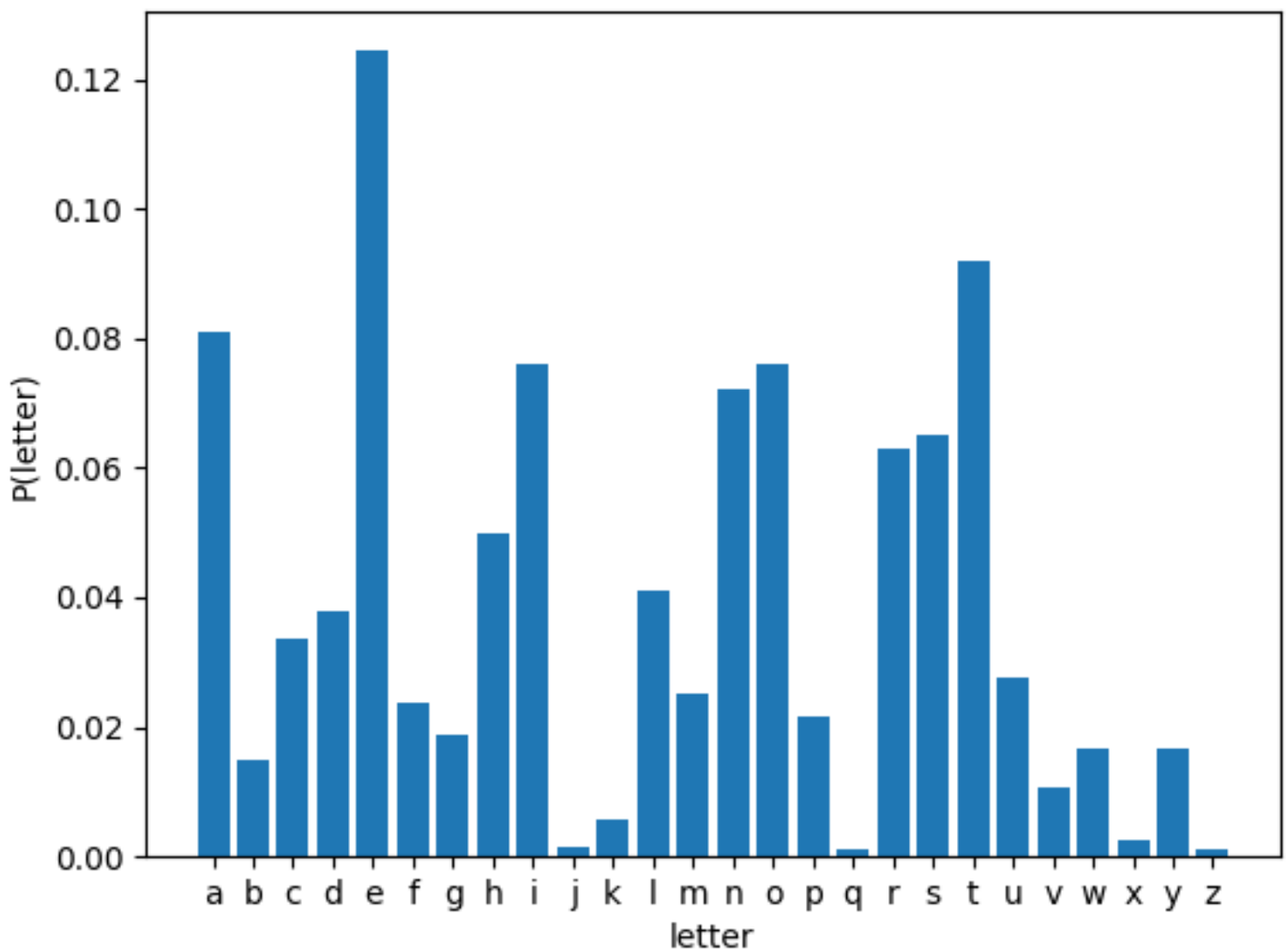
Character Probabilities

```
pos = dict(zip(characters, range(len(characters))))
counts = np.zeros(len(characters), dtype='uint64')

line_count = 0

for filename in sys.argv[1:]:
    for line in gzip.open(filename, "rt"):
        fields = line.lower().strip().split()

        count = int(fields[2])
        word = fields[0]

        if "_" in word:
            continue

        letters = letter_regex.findall(word)

        if len(letters) != len(word):
            continue

        for letter in letters:
            if letter not in pos:
                continue

            counts[pos[letter]] += count

total = np.sum(counts)
pos = list(pos.items())
pos.sort(key=lambda x: x[1])

for key, value in enumerate(pos):
    print(value[0], counts[key]/total)
```

build_lang_table.py

Character Probabilities

```
pos = dict(zip(characters, range(len(characters))))
counts = np.zeros(len(characters), dtype='uint64')
```

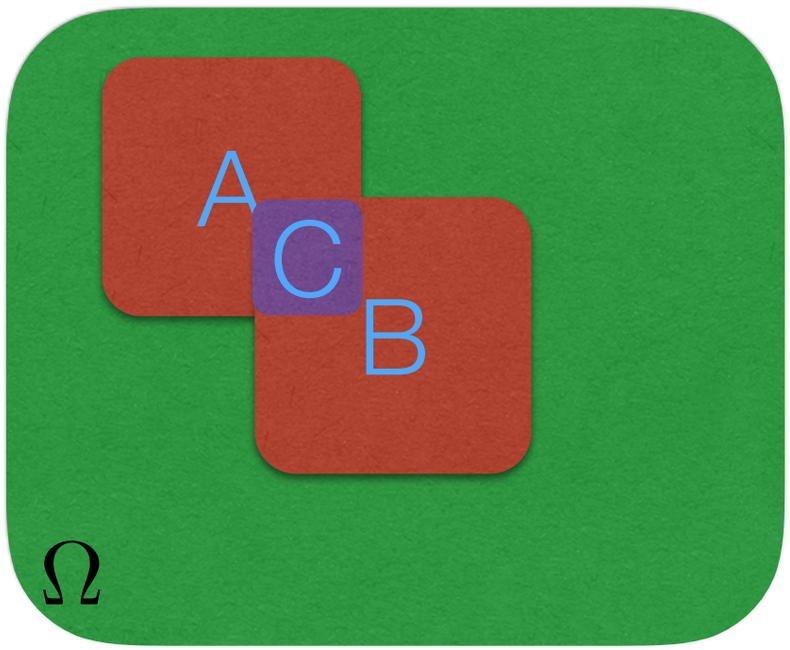



```
for key, value in enumerate(pos):
    print(value[0], counts[key]/total)
```

build_lang_table.py

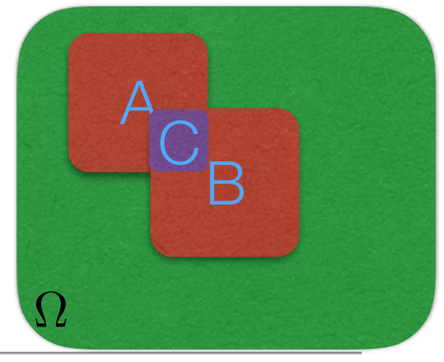
Probability

$$P(A) = \text{"Area" of A}$$
$$P(\Omega) = 1 \text{ (Normalization)}$$



$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$
$$P(C) = P(A \text{ and } B) = \text{overlap of A and B}$$

Probability



$$P(A) = \text{"Area" of A}$$
$$P(\Omega) = 1 \text{ (Normalization)}$$
$$P(A \text{ or } B) = P(A) + P(B) - P(A \text{ and } B)$$
$$P(C) = P(A \text{ and } B) = \text{overlap of A and B}$$

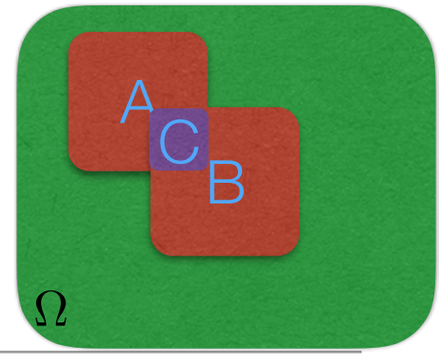
What's the probability that I'm in B given that I'm in A?

What fraction of A is occupied by B?

$$P(B|A) = \frac{P(C)}{P(A)} \rightarrow P(C) = P(B|A)\,P(A)$$

$$P(C) = P(C)$$

$$P(C) = P(C)$$



$$P(B|A) = \frac{P(C)}{P(A)} \rightarrow P(C) = P(B|A)\,P(A)$$

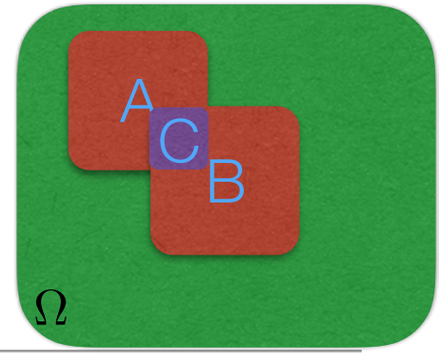
$$P(A|B) = \frac{P(C)}{P(A)} \rightarrow P(C) = P(A|B)\,P(B)$$

$$P(B|A)\,P(A) = P(A|B)\,P(B)$$

$$P(B|A) = \frac{P(A|B)\,P(B)}{P(A)}$$

Bayes Theorem

Medical Tests



Your doctor thinks you might have a rare disease that affects 1 person in 10,000. A test that is 99% accurate comes out positive. What's the probability of you having the disease?

Bayes Theorem:

$$P\left(disease|positive\ test\right) = \frac{P\left(positive\ test|disease\right)P\left(disease\right)}{P\left(positive\ test\right)}$$

Total Probability:

$$P\left(positive\ test\right) = P\left(positive\ test|disease\right)P\left(disease\right) + P\left(positive\ test|no\ disease\right)P\left(no\ disease\right)$$

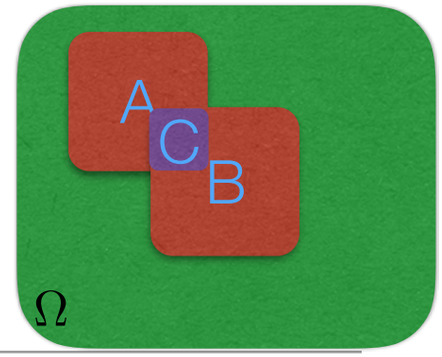
Finally:

$$P\left(disease|positive\ test\right) = 0.0098$$

Base Rate Fallacy

Low Base Rate Value
+
Non-zero False Positive Rate

Medical Tests

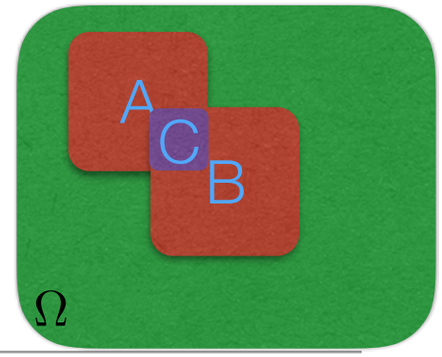


Consider a population of 1,000,000 individuals. The numbers we should expect are:

Marginals

	disease	no disease	
positive	99	9,999	10,098
negative	1	989,901	989,902
Marginals →	100	999,900	1,000,000

Medical Tests



Consider a population of 1,000,000 individuals. The numbers we should expect are:

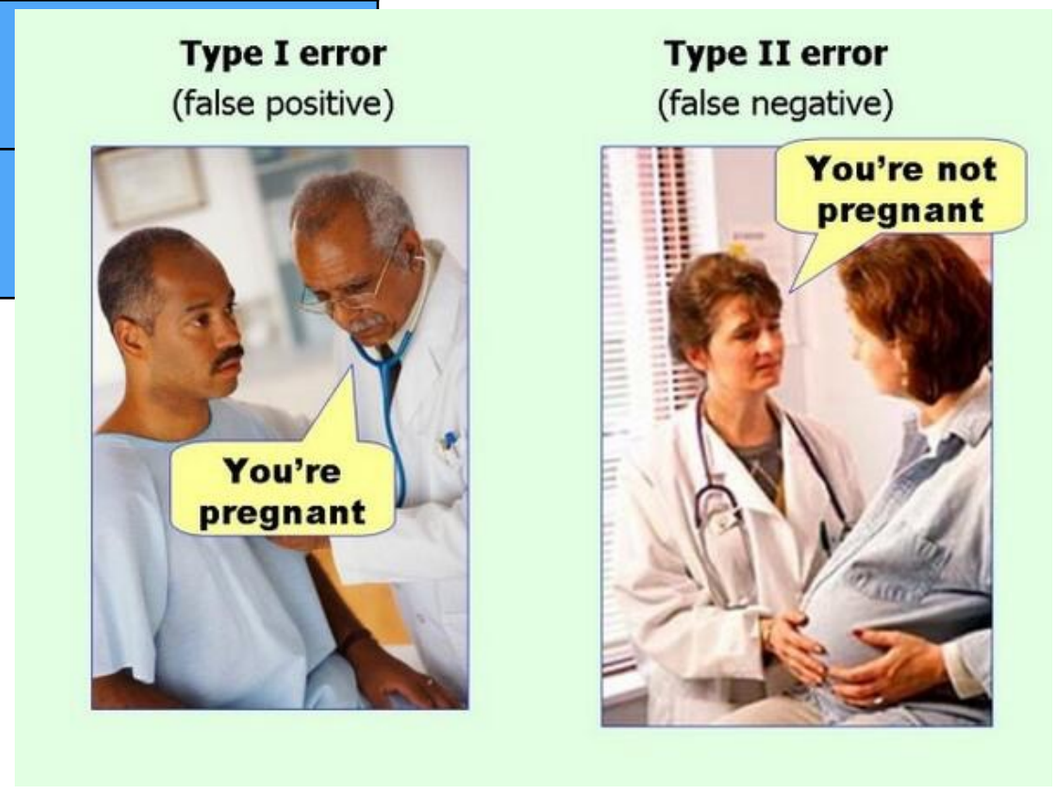
Marginals

	disease	no disease	
positive	99	9,999	10,098
negative	1	989,901	
Marginals →	100	999,900	

$$P\left(disease|positive\ test\right) = \frac{TP}{TP + FP} = 0.0098$$

$$P\left(no\ disease|negative\ test\right) = \frac{TN}{TN + FN} = 0.99999$$

@bgoncalves



(Confusion Matrix)

Feature / Test	positive	negative
positive	TP	FP
negative	FN	TN

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$specificity = \frac{TN}{FP + TN}$$

$$precision = \frac{TP}{TP + FP}$$

$$sensitivity = \frac{TP}{TP + FN}$$

harmonic mean

$$F1 = \frac{2TP}{2TP + FP + FN}$$

A second Test

Bayes Theorem still looks the same:

$$P\left(disease|positive\ test\right) = \frac{P\left(positive\ test|disease\right)P\left(disease\right)}{P\left(positive\ test\right)}$$

but now the probability that we have the disease has been updated:

$$P^{\dagger}\left(disease\right) = 0.0098$$

So this time we find:

$$P^{\dagger}\left(disease|positive\ test\right) = 0.4949$$

Each test is providing new evidence, and Bayes theorem is simply telling us how to use it to update our beliefs.