



# Foundations of Social and Cultural Data Analysis


Dr. Nanne van Noord & Dr. Melvin Wevers



# Assignment

- Make sure that the code runs!
  - Restart the kernel and re-run all cells
  - Include git cloning, pip imports, and untar the data!
- Read the assignments carefully!

Each record, then, should be a **list** with four elements: (i) the year of publication, (ii) the title, (iii) the name of the author, and (iv) the name of the publisher.



```
('year', 'title', 'author', 'publisher')
('1642', 'Hiervsaem verwoest. Trevrspel.', 'Joost van den Vondel 1587-1679', 'Matthijsz, Paulus Amsterdam')
('1641', 'Gysbrecht van Aemstel, d'ondergangh van zijn stad en zijn ballingschap. Treurspel.', 'Joost van den Vondel 1587-1679', 'Houthaeck, Dirck Cornelisz Amsterdam')
('1720', 'Joseph in Egypten. Trevrspel.', 'Joost van den Vondel 1587-1679', 'Oosterwyk, Johannes van Amsterdam')
('17XX', 'Lucifer. Treurspel.', 'Joost van den Vondel 1587-1679', 'Wees, Abraham de I Amsterdam')
```

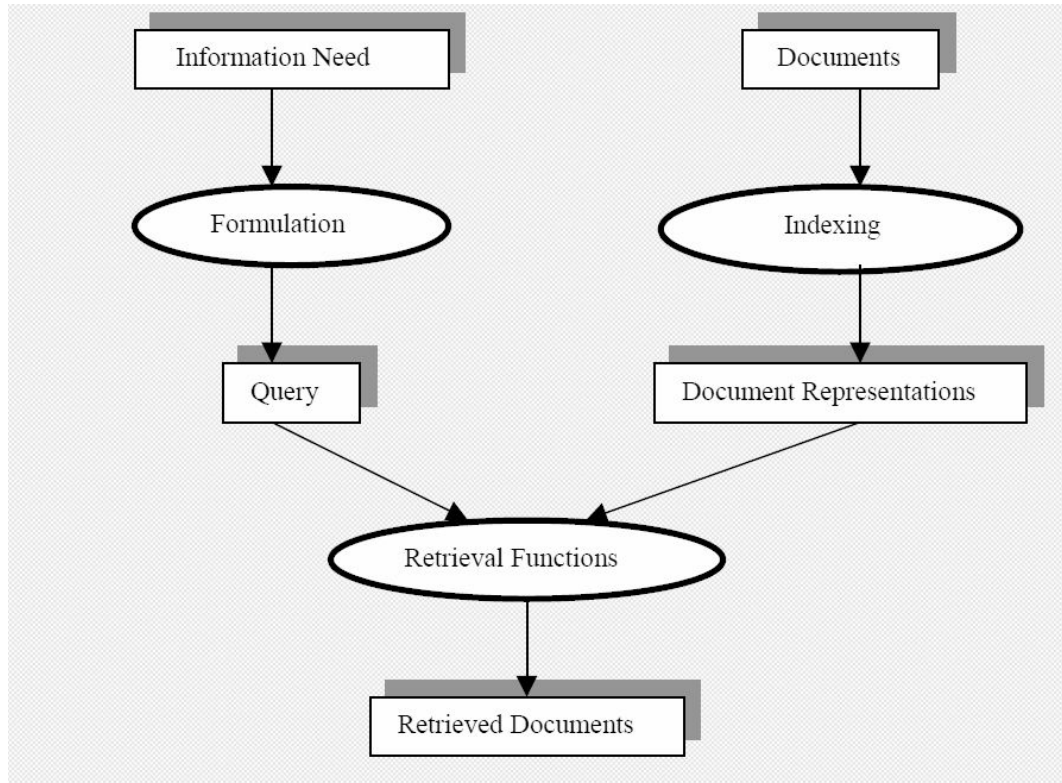
- Are there additional questions to be answered?
- Try to avoid redundant or needless code.
  - Functions can be very useful to reduce redundancy and increase readability!



# Recap

- Data search/retrieval
- Variable types / measurement scales

# Sharing/Search Relation



# Representation/Encoding

Machine/Human readable is a representation choice

- Possible to have parallel representations

Structured  
Data



Unstructured  
Data



Documents

Documents

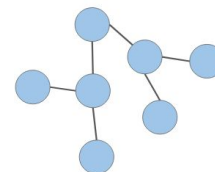
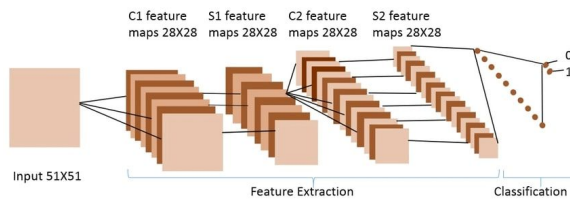
We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring ...



Vector-space  
representation

	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

Term-document matrix



Features / Attributes

Node	Attr 1	Attr 2	...	Attr n
N-1	-	-	-	-
N-2	-	-	-	-
...	-	-	-	-
N-m	-	-	-	-

# Indexing

To aid search we may want to construct a representation of the data that is specifically tailored to enable search.

## Index

Entry titles are printed in small capitals. Bold page-numbers indicate a sustained discussion of a topic, whether or not it features as an entry.

AESTHETICS 18, **31-5**, 123, 251-3; and  
 ETHICS 28, 31-2, 107  
 ability *see* disposition  
 abstractionism 314  
 acquaintance 43, 102, 160, 308, 212,  
 254, 269-70, 277, 299, 310, 348-9,  
 383  
 agreement 128-9, 135-6, 328, 368  
 ambiguity/synonymy 40, 122, 240  
 analytic/synthetic 18, 20, 131, 199-200,  
 202, 353, 356-7  
 analytic definition 26, 33, 35, **113-14**,  
**120-4**, 152  
 ancestral relation 266  
 'and so on' 149, 265, 328  
 Anscombe, G.E.M. 29, 74, 75  
 anti-realism 95, 382, 384  
 ANTHROPOLOGY **35-6**, 126, 128, 236; *see*  
 also HUMAN BEING  
 a priori *see* analytic/synthetic; philosophy;  
 synthetic a priori  
 Aquinas, T. 323  
 argument *see* function  
 Aristotle 29, 43, 124, 199, 212, 220, 226,  
 241, 292, 294, 300, 318, 340, 354, 362  
 arithmetic 20, 24, 234  
 aspect-blindness 39  
 aspect-dawning 36-9  
 ASPECT-PERCEPTION 27, 34, **36-40**, 57,  
 120, 170; continuous 40  
 assertion **60-3**, 301-2  
 assumption 61-3  
 Augustine 25, 41, 242, 277, 285, 295  
 AUGUSTINIAN PICTURE OF LANGUAGE 25,  
**41-5**, 144, 175, 195, 211, 238, 255-6,  
 274, 277, 310, 376

397

**Doc 1:**  
 I did enact Julius  
 Caesar: I was killed  
 i' the Capitol; Brutus  
 killed me.

⇒  
 Tokenisation

**Doc 2:**  
 So let it be with  
 Caesar. The noble  
 Brutus hath told  
 you Caesar was  
 ambitious.

⇒  
 Tokenisation

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

⇒  
 Sorting

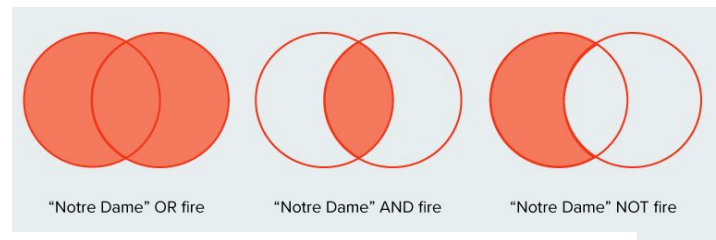
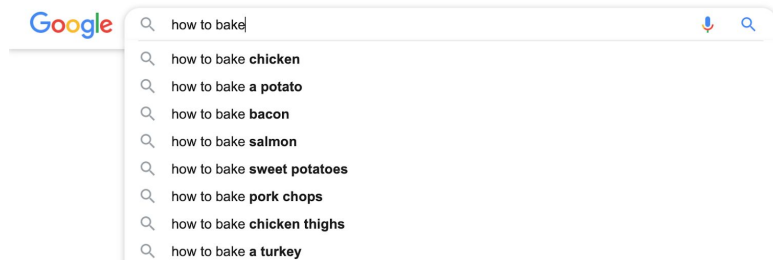
Term (sorted)	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	2
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	2
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
you	2
was	1
was	2
with	2

# Querying

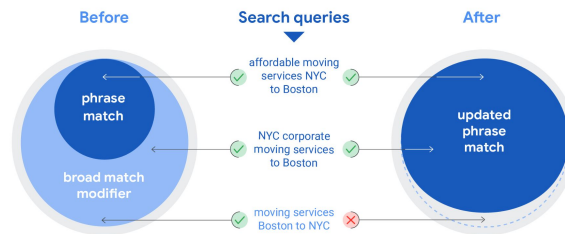
How to enable querying depends on needs and expertise of users.

Various types of queries:

- Keyword queries
  - Most common
  - Keywords implicitly connected by AND
- Boolean queries
  - Allow range of logical operators (AND, OR, NOT)
- Phrase queries
  - Search for exact multi-word match



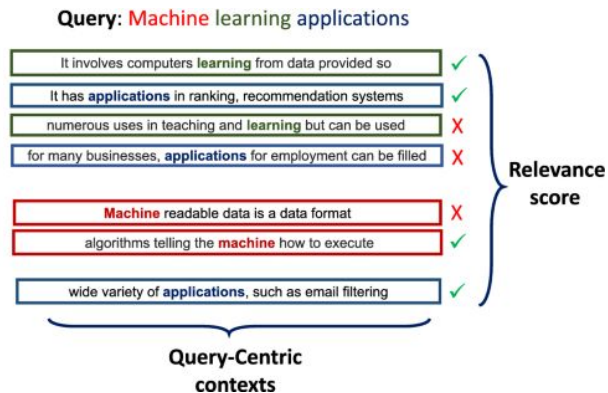
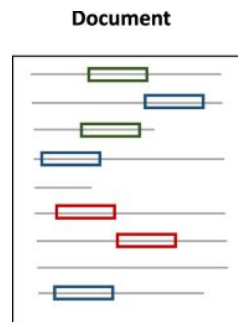
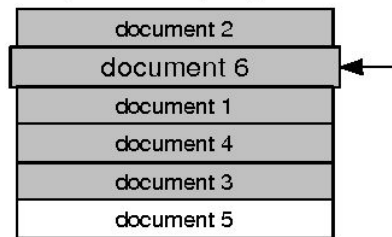
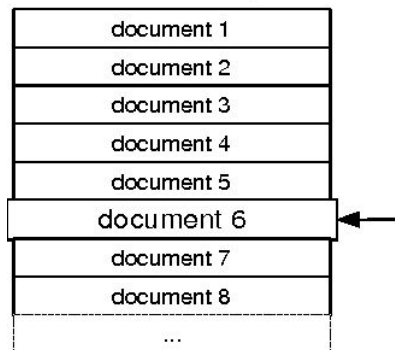
Keyword= "moving services NYC to Boston" or +moving +services +NYC +to +Boston



\*Note that these circles are not to scale

# Ranking

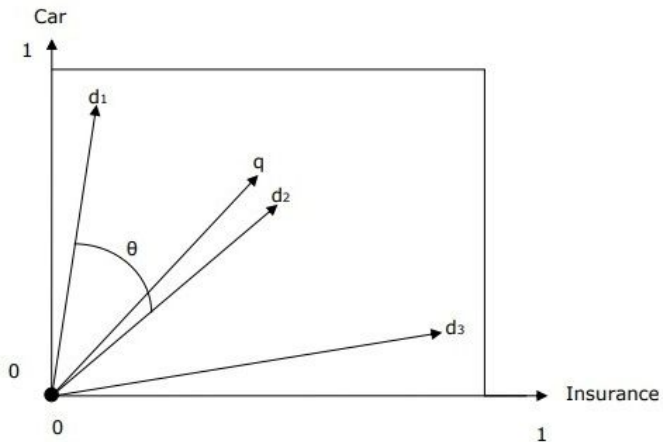
Give a score to each indexed document based on query and return in order



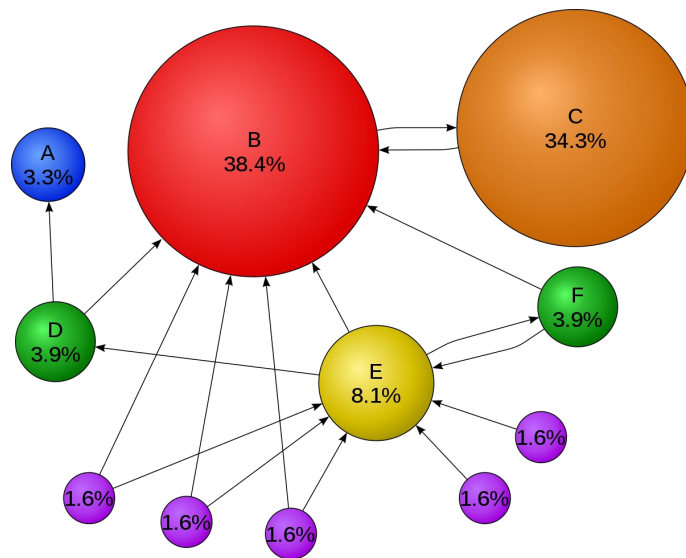


# Ranking - Relevance Score

Vector Space Model



PageRank





# Measurement Scales

- . Another aspect of a variable we can use to describe it is the measurement scale
- . The measurement scale tells us:
  - How to interpret the values on the scale in relation to the other values
  - How to compare the values

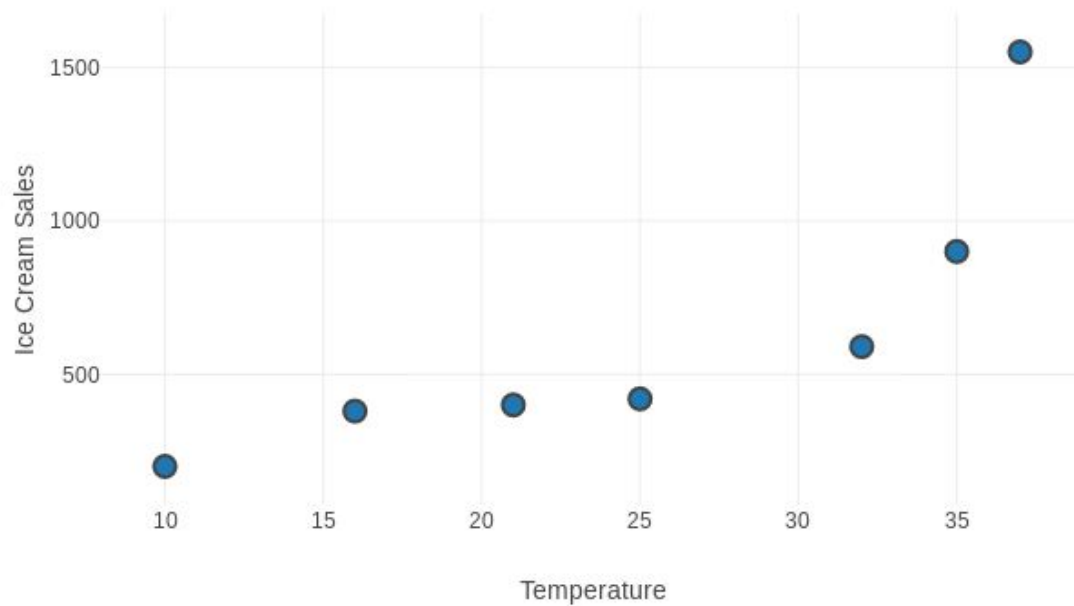
---

# Relations between variables

# Association



Temperature versus Ice Cream Sales





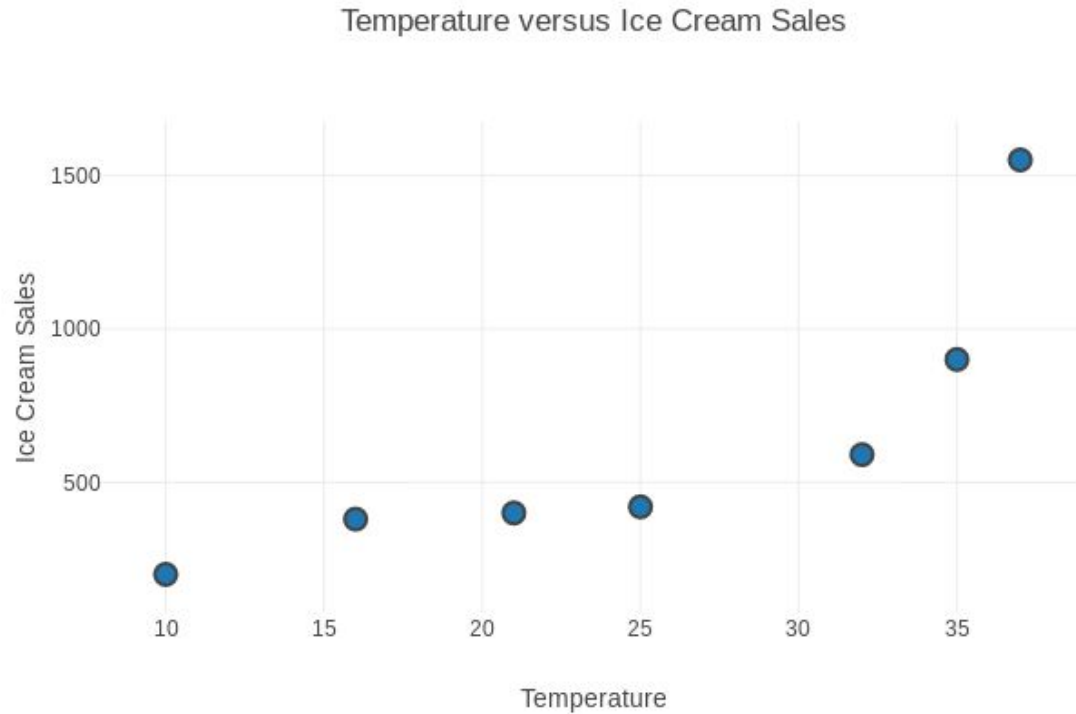
# Covariance

- Measure of joint variability of two variables
  - If  $X_i$  and  $Y_i$  are both high, or both low, then covariance is positive
  - If  $X_i$  is high and  $Y_i$  is low, or other way around, then covariance is negative
  - Magnitude depends on values of variables

- Covariance:

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^N (X_i - \mu_X)(Y_i - \mu_Y)}{N}$$

# Correlation



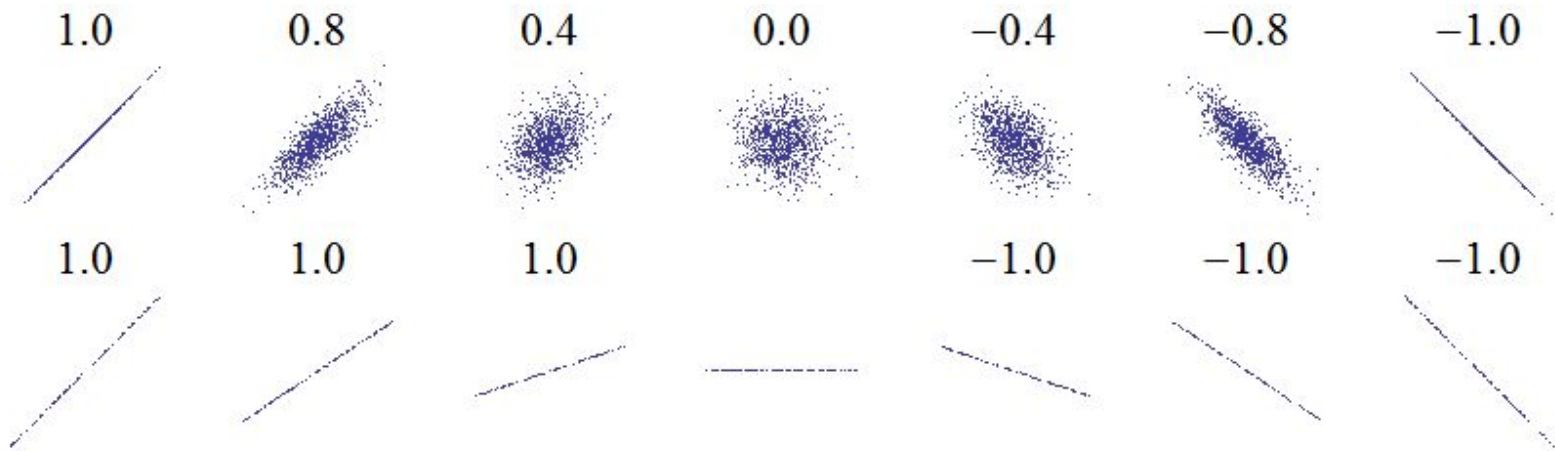


## Correlation

- Statistical relationship between two variables
- Normalized covariance: magnitude shows strength of relationship
- Popular measure is (sample) Pearson's r correlation coefficient:

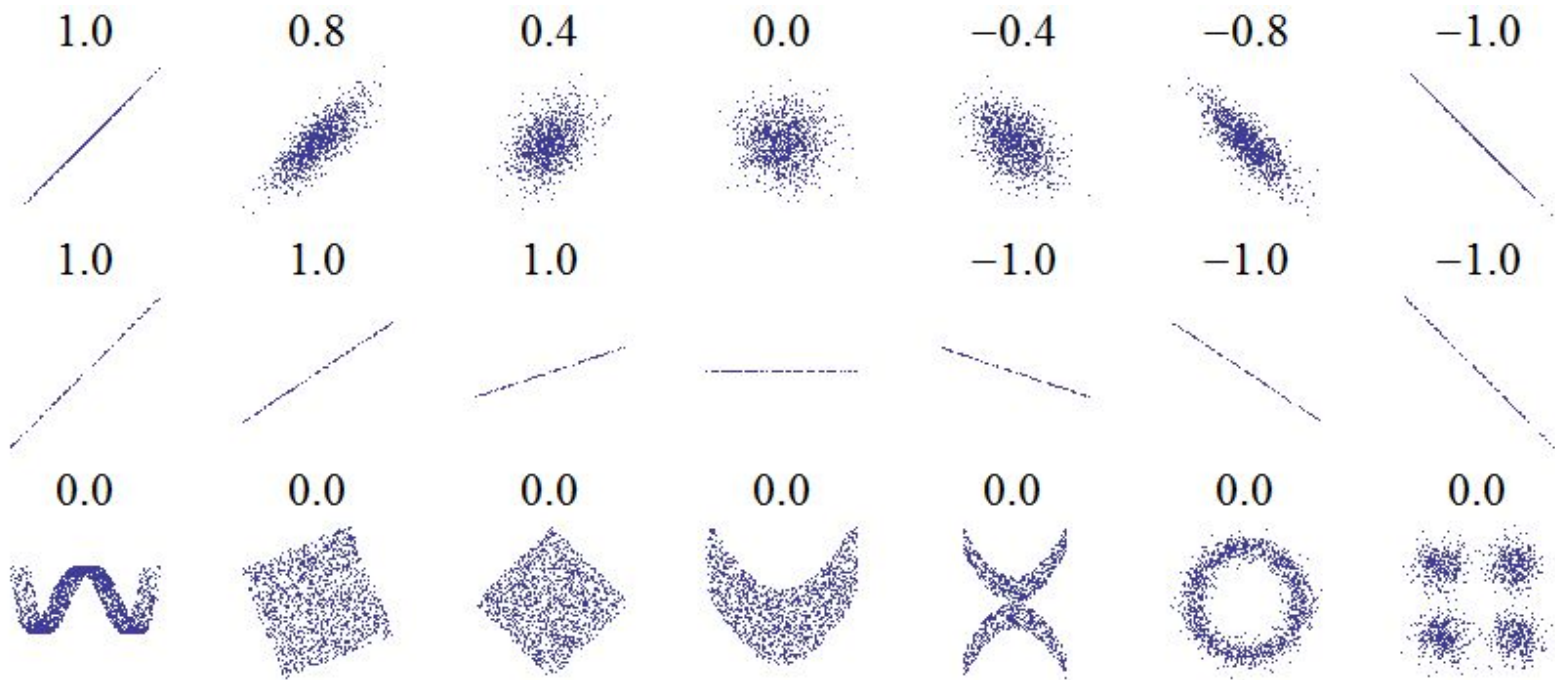
$$r_{xy} = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^N (X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^N (X_i - \mu_X)^2} \sqrt{\sum_{i=1}^N (Y_i - \mu_Y)^2}}$$

# Visualising correlations



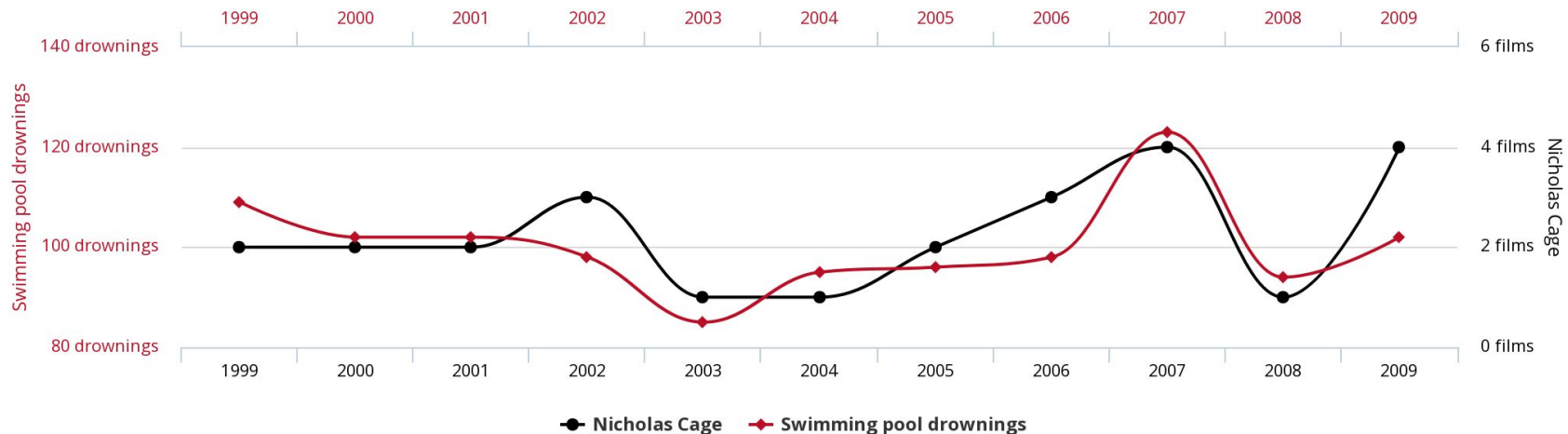


# Visualising correlations



# Spurious correlations

**Number of people who drowned by falling into a pool**  
correlates with  
**Films Nicolas Cage appeared in**

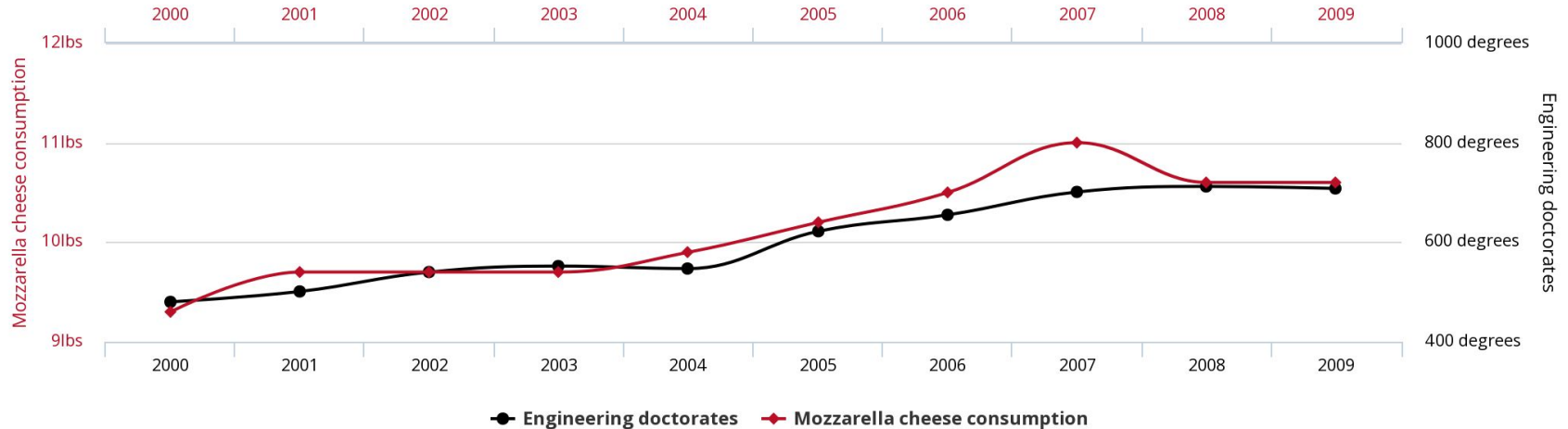


tylervigen.com

<http://www.tylervigen.com/spurious-correlations>

# Spurious correlations

**Per capita consumption of mozzarella cheese**  
correlates with  
**Civil engineering doctorates awarded**

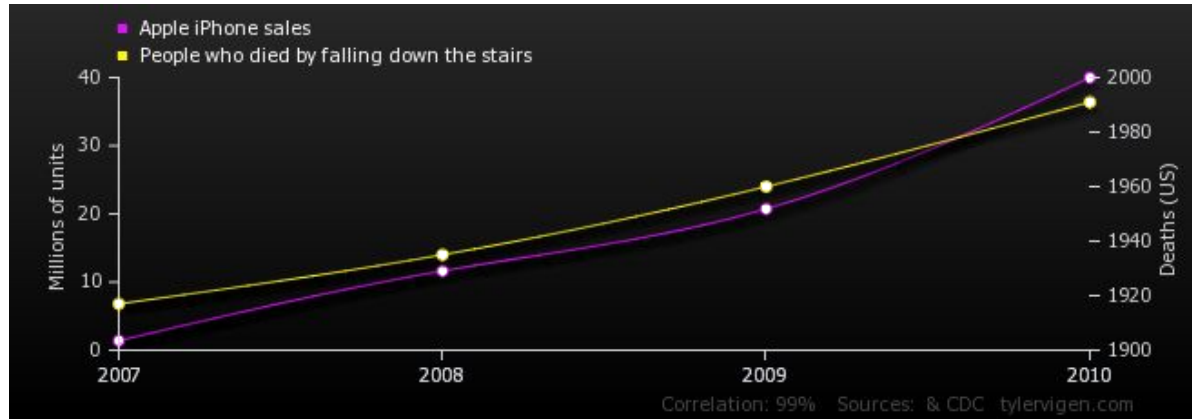


tylervigen.com

<http://www.tylervigen.com/spurious-correlations>

# Spurious correlations

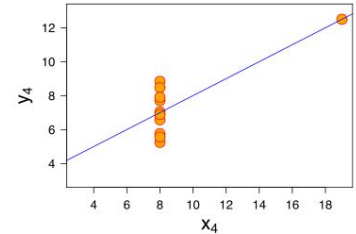
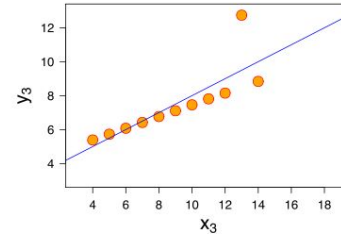
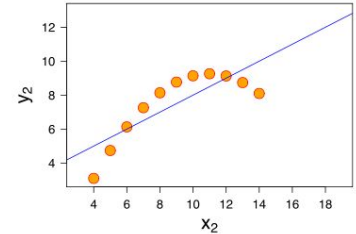
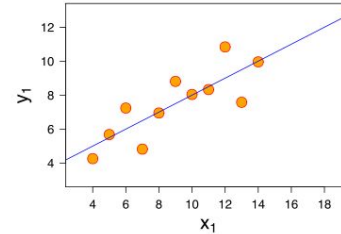
Correlation: 0.994751



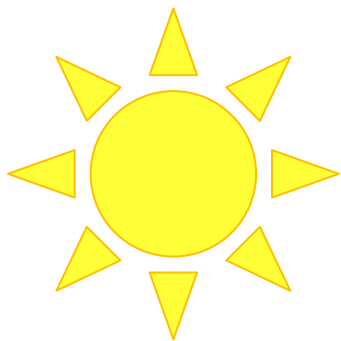
[https://tylervigen.com/view\\_correlation?id=28669](https://tylervigen.com/view_correlation?id=28669)

# Linear relation

- Correlation describes a linear relationship between variables
- But can be high even if relation isn't linear
- Can be low even if there is an obvious relationship



# Causation



Causation



Correlation



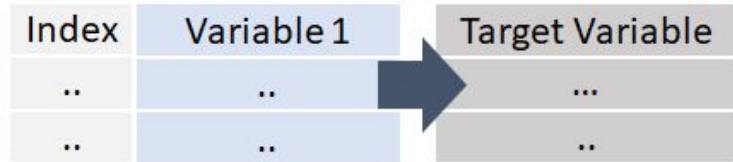
---

# Relations between instances

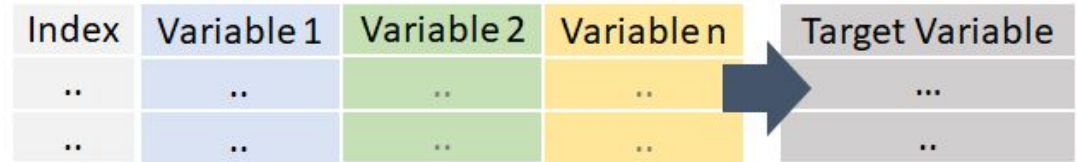


## How to compare rows?

Univariate



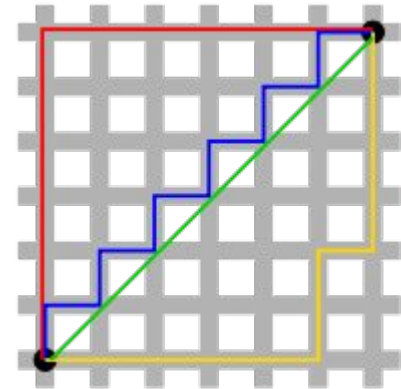
Multivariate





## Distance Metrics

- Basic properties
  - Positive separation
    - $D(x, y) > 0, \forall x \neq y$
    - $D(x, y) = 0$ , i.f.f.,  $x = y$
  - Symmetry
    - $D(x, y) = D(y, x)$
  - Triangle inequality
    - $D(x, y) \leq D(x, z) + D(z, y)$



## Dot product

$$\mathbf{a} = [a_1, a_2, \dots, a_n]$$

$$\mathbf{b} = [b_1, b_2, \dots, b_n]$$

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

```
a = [1, 2, 3, 4]
b = [4, 5, 6, 7]
```

[1] ✓ 0.0s

```
result = 0
for i in range(len(a)):
    result += a[i] * b[i]
display(result)
```

[2] ✓ 0.0s

...

60

```
result = 0
for ai, bi in zip(a, b):
    result += ai * bi
display(result)
```

[4] ✓ 0.0s

...

60

```
result = sum([ai * bi for ai, bi in zip(a,b)])
display(result)
```

[5] ✓ 0.0s

...

60

```
import numpy as np
np.dot(a,b)
```

[7] ✓ 0.0s

...

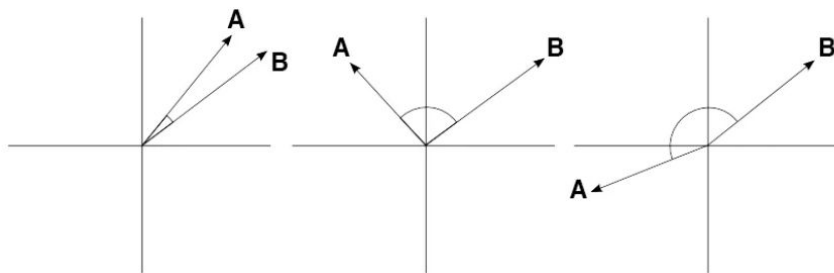
60

# Cosine

$$\text{cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}},$$

```
np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b))  
[15] ✓ 0.0s  
... 0.9759000729485332  
  
a_norm = a / np.linalg.norm(a)  
b_norm = b / np.linalg.norm(b)  
np.dot(a_norm, b_norm)  
[16] ✓ 0.0s  
... 0.9759000729485331  
  
import scipy.spatial.distance as dist  
  
dist.cosine(a,b)  
[17] ✓ 0.0s  
... 0.024099927051466796  
  
1 - np.dot(a, b) / (np.linalg.norm(a) * np.linalg.norm(b))  
[18] ✓ 0.0s  
... 0.024099927051466796
```

# Angle and magnitude



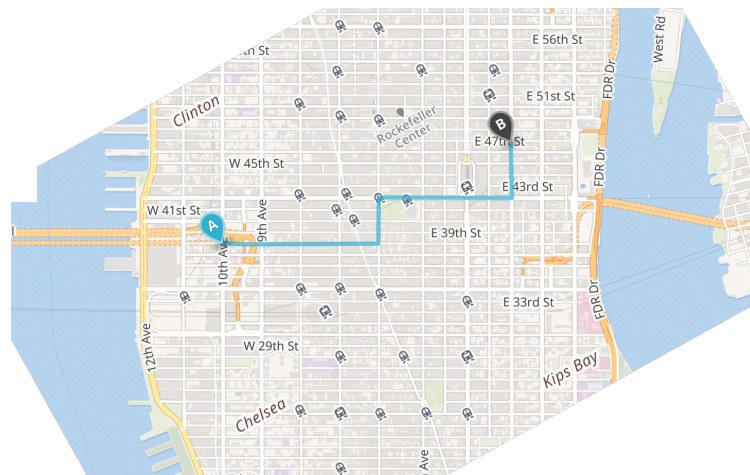
```
▶ a = [10, 20, 30, 40]  
[21] ✓ 0.0s
```

```
[25] import numpy as np  
      np.dot(a,b)  
... 600
```

```
import scipy.spatial.distance as dist  
  
dist.cosine(a,b)  
[28] ✓ 0.0s  
... 0.024099927051466907
```

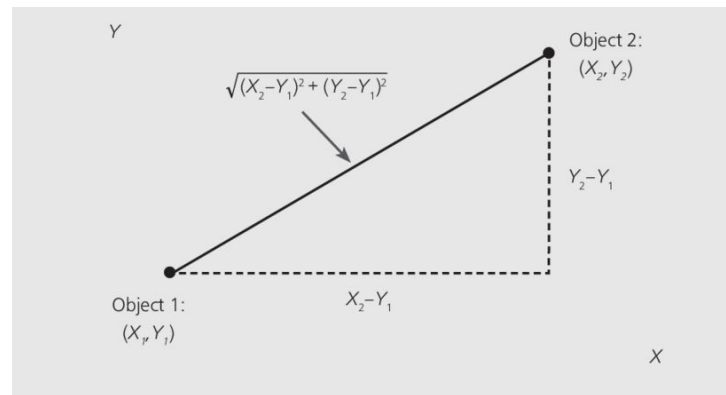
# Manhattan

$$d_T(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_T = \sum_{i=1}^n |p_i - q_i|$$











# Euclidean

$$d_2(\vec{a}, \vec{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$



# Representation Learning

Unstructured data

 Text files and documents	 Server, website and application logs	 Sensor data	 Images
 Video files	 Audio files	 Emails	 Social media data

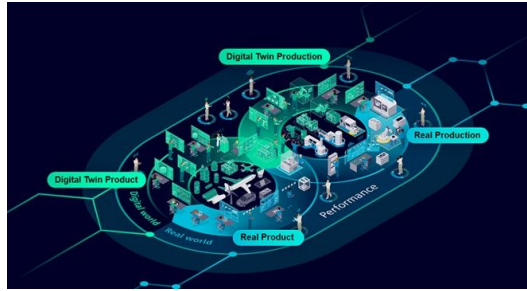
- Cannot simply measure the distance
- Need to build a model for how to represent

# Modelling

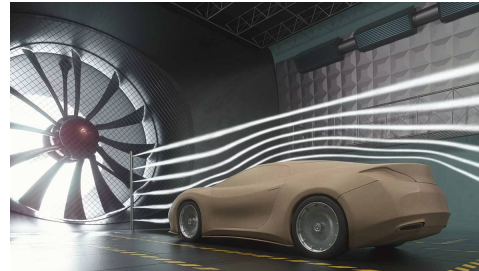
S = System

M = Model

E = Experiment



$$E(S) \approx E(M)$$



shutterstock.com · 2134374559





# What gets counted?

- Which data we count/store already depends on a model
  - What we count reflects how we see the world
- A data model is less nuanced than reality
  - Requires assumptions and abstractions

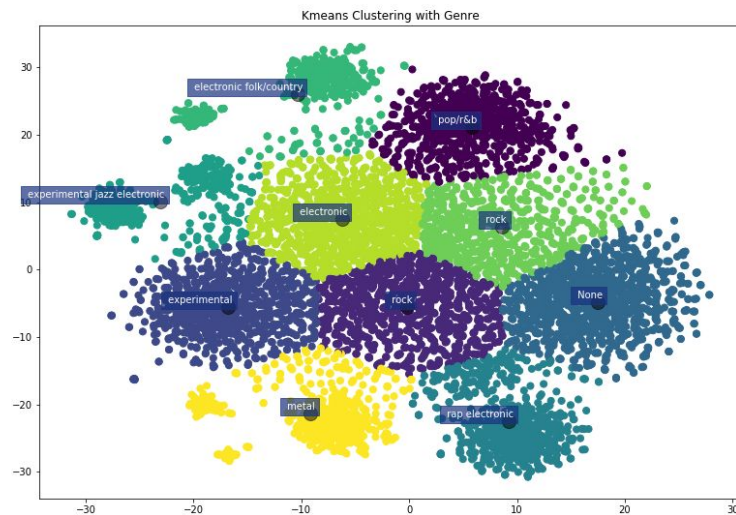
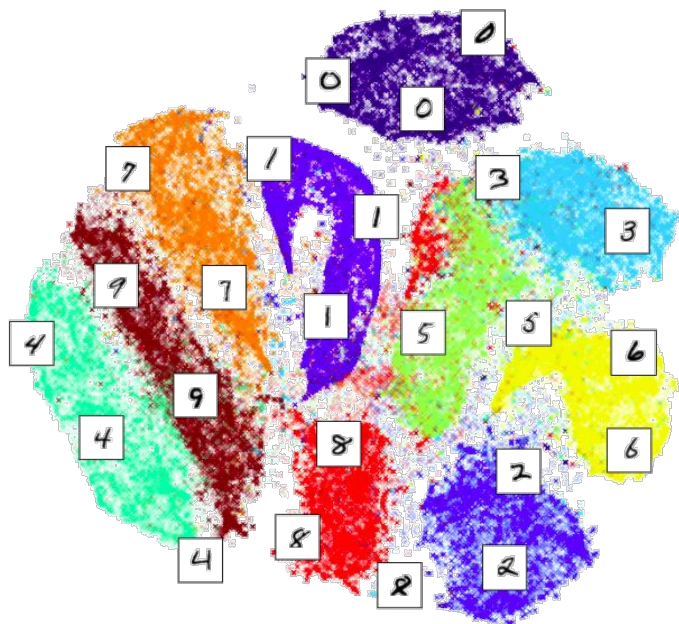
if

$S = \text{Gender}$

$M = \text{Binary}$

Then  $E(S) \approx E(M)$  ?

# Data Model



# Bag of Words

Document D1	<i>The child makes the dog happy</i> the: 2, dog: 1, makes: 1, child: 1, happy: 1
Document D2	<i>The dog makes the child happy</i> the: 2, child: 1, makes: 1, dog: 1, happy: 1



	child	dog	happy	makes	the	BoW Vector representations
D1	1	1	1	1	2	[1,1,1,1,2]
D2	1	1	1	1	2	[1,1,1,1,2]

# What is the model?

```
from collections import Counter

corpus = [
    "the child makes the dog happy",
    "the dog makes the child happy",
    "my child is happy when playing with another child"
]

tokenized_corpus = [d.split() for d in corpus]

vocabulary = Counter()
for document in tokenized_corpus:
    vocabulary.update(document)
vocabulary = sorted(vocabulary)

def BoW(doc, vocabulary):
    bow = [0] * len(vocabulary)
    for word, cnt in Counter(doc).items():
        bow[vocabulary.index(word)] = cnt
    return bow

bows = [BoW(d, vocabulary) for d in tokenized_corpus]
```

[34] ✓ 0.0s

```
dot_distances = dist.squareform(dist.pdist(bows, metric=np.dot))
np.fill_diagonal(dot_distances, np.inf)
display(dot_distances)
```

[35] ✓ 0.0s

```
... array([[inf, 8., 3.],
          [ 8., inf, 3.],
          [ 3., 3., inf]])
```

```
cos_distances = dist.squareform(dist.pdist(bows, metric='cosine'))
np.fill_diagonal(cos_distances, np.inf)
display(cos_distances)
```

[36] ✓ 0.0s

```
... array([[          inf, 2.22044605e-16, 6.80198925e-01],
          [2.22044605e-16,          inf, 6.80198925e-01],
          [6.80198925e-01, 6.80198925e-01,          inf]])
```

```
dot_distances.argmax(1), cos_distances.argmax(1)
```

[37] ✓ 0.0s

```
... (array([2, 2, 0]), array([1, 0, 0]))
```



# Working with data is modelling

- How we see the world influences how we model it
- How we model the world influences how we see it