# Introduction to Big Data Analysis

# 3803ICT course structure

**W1.** Introduction to Big Data Analysis

Data Preparation and Preprocessing

**W2.** Data Preparation & Preprocessing

Data Analysis and Interpretation

**W3.** Exploratory Data Analysis

**W4.** Statistical Data Analysis

**W5.** Predictive Data Analysis

Visualization

**W6.** Visual Data Analysis

Analysis of special types of data

**W7.** Time Series

**W8.** Textual Data

**W9.** Graph Data

Analysis with Big Data infrastructures

**W10.** Distributed Data Analysis

**W11.** Cloud-based Data Analysis

**W12.** Revision

# Unit Logistics: Reading Materials

- Recommended books:
  - Doing Data Science. Cathy O'Neil, Rachel Schutt. O'Reilly Media, October 2013
  - Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications. Igual, Laura, Seguí, Santi. Springer Verlag
  - Algorithms for Data Science. Steele, Brian, Chandler, John, Reddy, Swarna. Springer Verlag
- Programming environment:
  - Python, Jupyter notebook
  - Python packages: pandas, scikit-learn, etc.

# Introduction to Big Data Analysis

I. Data Science
II. Data Analysis and Applications
III. Data Types
IV. Recap: Statistics
V. Recap: Relational Database
VI. Tools

# I. Data Science

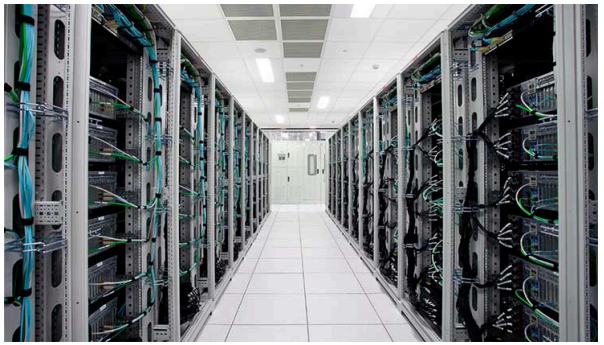
# From Science to Data Science

- Thousand years ago: science was **empirical**
  - *Describing natural phenomena*
- Last few hundred years: **theoretical branch**
  - *Using models, generalizations*

$$E=mc^2$$

- Last few decades: **computational** branch
  - *Simulating complex phenomena*
- Today: **data exploration** (eScience)
  - *Unify theory, experiment, and simulation*
    - Building several data centers to capture the data from devices , simulators, human, etc

**→Scientist analyzes the data to extract actionable insights**



The Four Paradigm: Data-Intensive Scientific Discovery, 2009

# What is Data Science

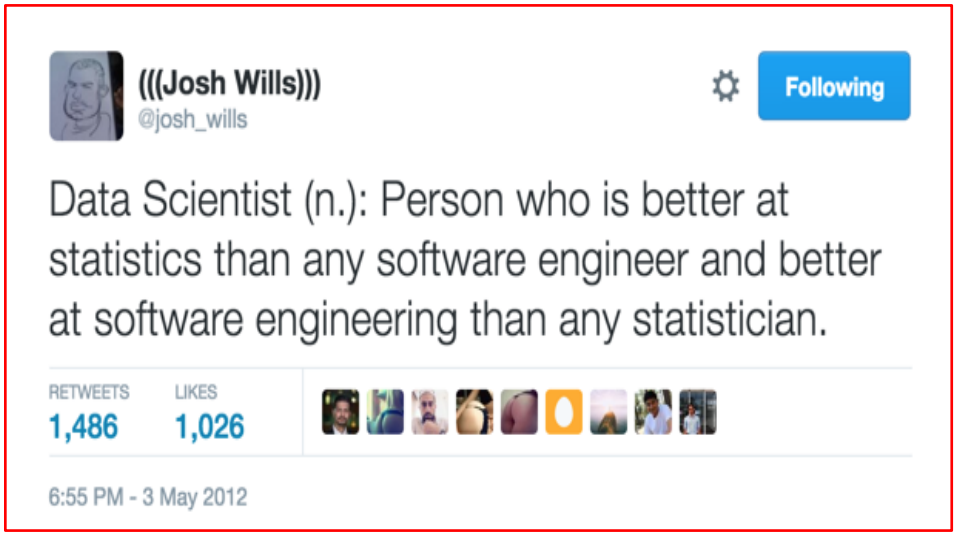
There isn't a definition agreed by all yet!

| | |
|---|---|
| Wikipedia | "Data Science is the extraction of knowledge from large volumes of data that are structured or unstructured" |
| **NIST, 2015** | **"Data science is the empirical synthesis of actionable knowledge from raw data through the data lifecycle process"** |
| Dhar, 2013 | "Data science is the study of generalizable knowledge from data" |
| Peter Naur, 1974 | "[data science is] The science of dealing with data, once they have been established, while the relation of the data to what they represent is delegated to other fields and sciences." |

# Data Scientists

"A data scientist is someone who can obtain, scrub, explore, model, and interpret data, blending hacking, statistics, and machine learning. Data scientists not only are adept at working with data, but appreciate data itself as a first-class product."

*Hilary Mason, chief scientist at bit.ly*




*Josh Wills, Data Scientist at Slack*

# Data Scientists in Job Markets

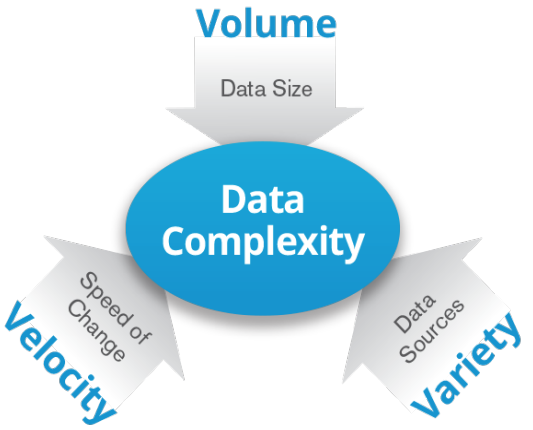
- Salary: very competitive payroll


https://au.indeed.com/jobs?q=data+scientist&l=

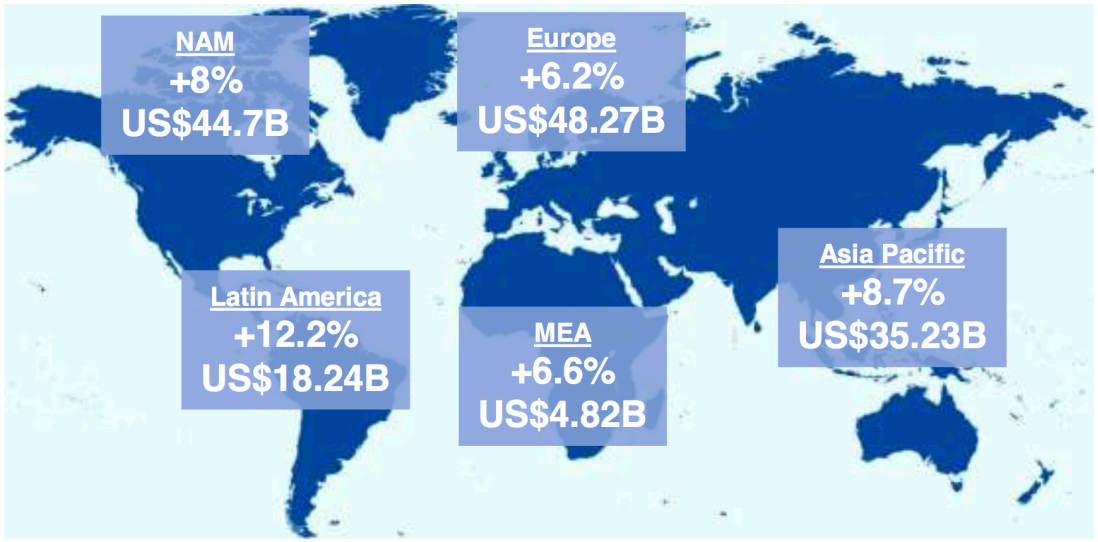
- Trend:
  - Businesses Will Need One Million Data Scientists by 2018

https://www.kdnuggets.com/2016/01/businesses-need-one-million-data-scientists-2018.html

# Why Data Science?




- Data becomes “Big”
  - More than 2.7 zettabytes of data exist today
  - Projected to grow to 180 zettabytes in 2025
  - 1 zettabytes = 1 billion terabytes (TB) = 1 trillion GB
- Data center becomes the brain of a company
  - 151.3 billion USD investment
  - 8% increase every year




https://steemit.com/indonesia/@slempase/big-data-data-is-the-new-gold-eng-big-data-data-adalah-emas-baru-ina-201796t161933938z
https://www.slideshare.net/beldeninc/data-center-trends-2014
https://www.simplilearn.com/data-science-vs-big-data-vs-data-analytics-article

# Why Data Science?

- Big Data open-ups cutting-edge technologies for Computer Science
  Usecase: Google Translation
  - Traditional approach:
    - Build a dictionary

| | |
|---|---|
| I | je |
| am | suis |
| student | étudiant |

    - Build a linguistic model to identify subject, verb, object
    - Perform 1-1 translation.

S V O

I am a student

je suis un étudiant

Brants, Thorsten, et al. "Large language models in machine translation." Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007.

# Why Data Science?

- Usecase: Google Translation
  - Google approach (using Big Data technique):
    - Do not build a dictionary

| | |
|---|---|
| I | |
| am | suis |
| student | étudiant |

    - Build a Big Data repository to store all possible transcripts

| | |
|---|---|
| I am a student | Je suis un étudiant |
| I am a doctor | Je suis médecin |
| … | … |

    - Given an input sentence, find the most similar sentence in the repository
  - Why it works?
    - Data contains models itself → If data is big enough, it contains all possible models
    - The repository is big enough → Can find similar sentences → Can translate all possible sentences

Brants, Thorsten, et al. "Large language models in machine translation." Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). 2007.

# Why Data Science?

- Add significant values for businesses:
  - Empower management and officers to make better decisions
  - Direct the actions based on trends which in turn help in defining goals
  - Challenge the staff to adopt best practices and focus on issues that matter
  - Make decisions with quantifiable, data-driven evidence
  - Identify and refine of target audiences
  - Recruit the right talent for the organization (e.g. linkedin.com, seek.com.au)

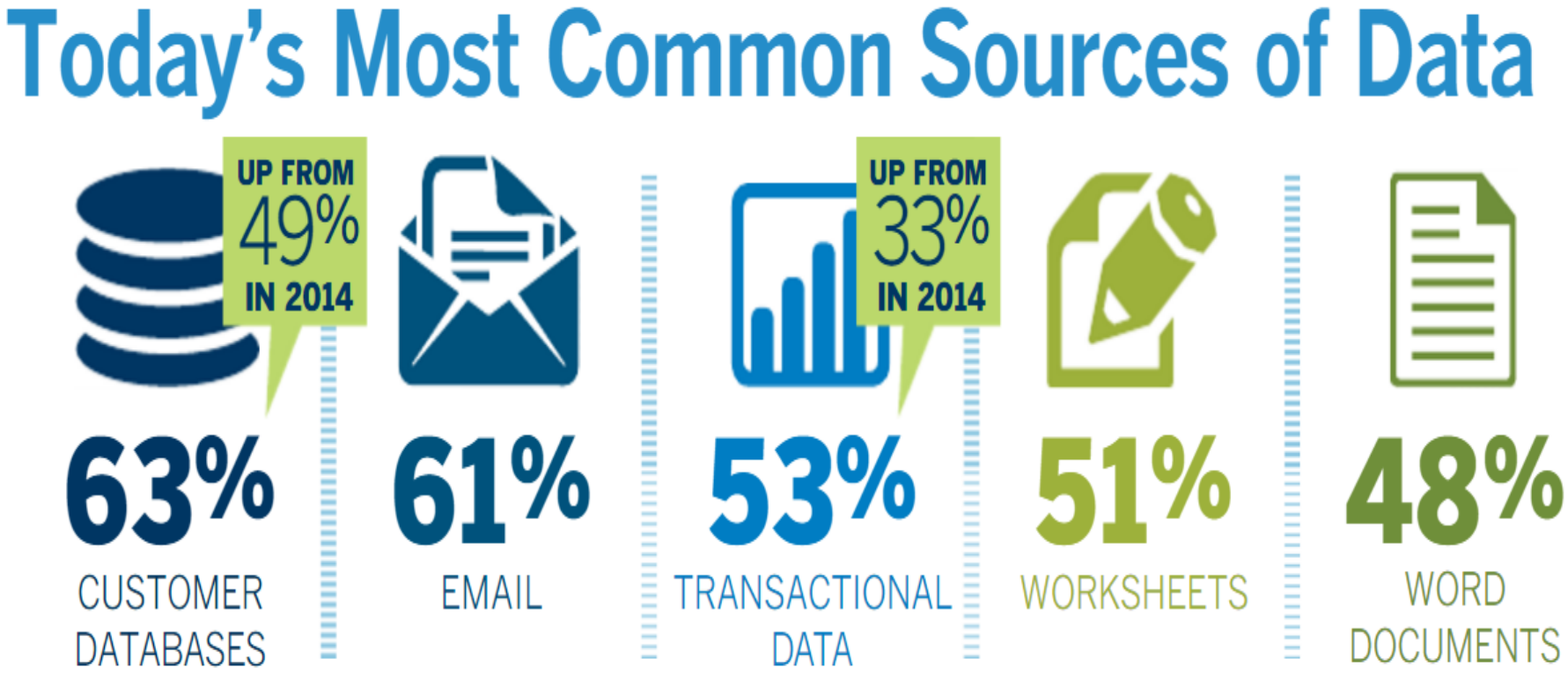
https://www.simplilearn.com/why-and-how-data-science-matters-to-business-article

# Where is data coming from?

- E.g. Tourism department wants to know the travel patterns of tourists when they visit Australia.
  - They have to buy data from telecommunication companies (e.g. Optus, Telstra)
    - Tourists often buy a SIM card with 4G when they arrive
  - Data scientists analyze the data to find patterns

# Where is data coming from?

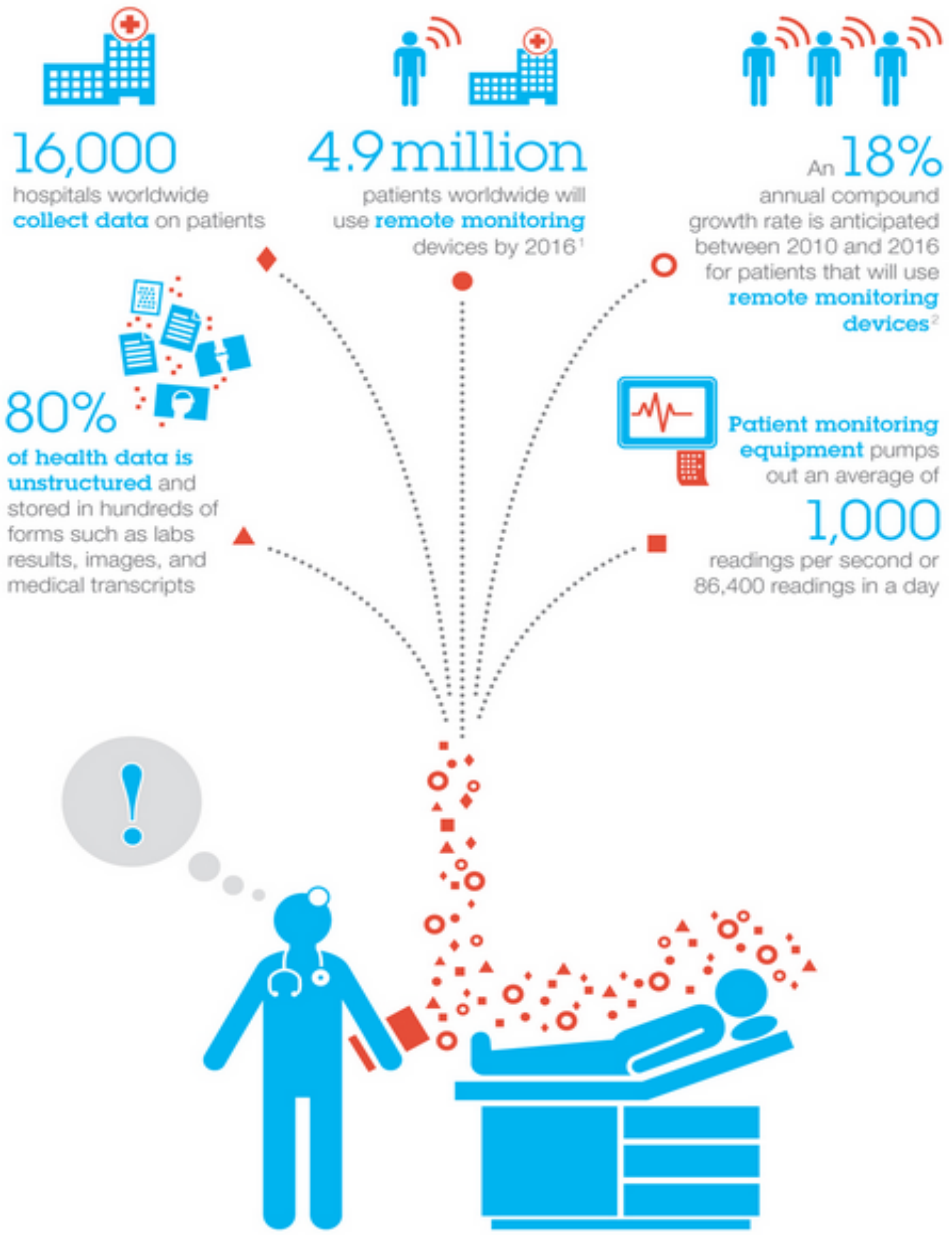
- **Businesses**




https://www.idgenterprise.com/resource/marketing-tools/big-data-and-analytics-the-big-picture-infographic/

# Where is data coming from?

- **Healthcare**




http://www.ibmbigdatahub.com/infographic/big-data-healthcare-tapping-new-insight-save-lives

# Where is data coming from?

- **Government**


Left-side is about data sources. Right-side is about analytics tool to enrich the data

Olugbenga Jola-Olukoya https://www.linkedin.com/pulse/big-data-enables-development-smart-cities-olugbenga-jola-olukoya

# Where is the data coming from?

- **World Wide Web**


- *Web 1.0*
- Documents
- Online shops

- *Web 2.0*
- Social media networks
- User-generated content

- *Web 3.0*
- Internet of things (mobile phones, cars, smart grids…)
- Semantic Web

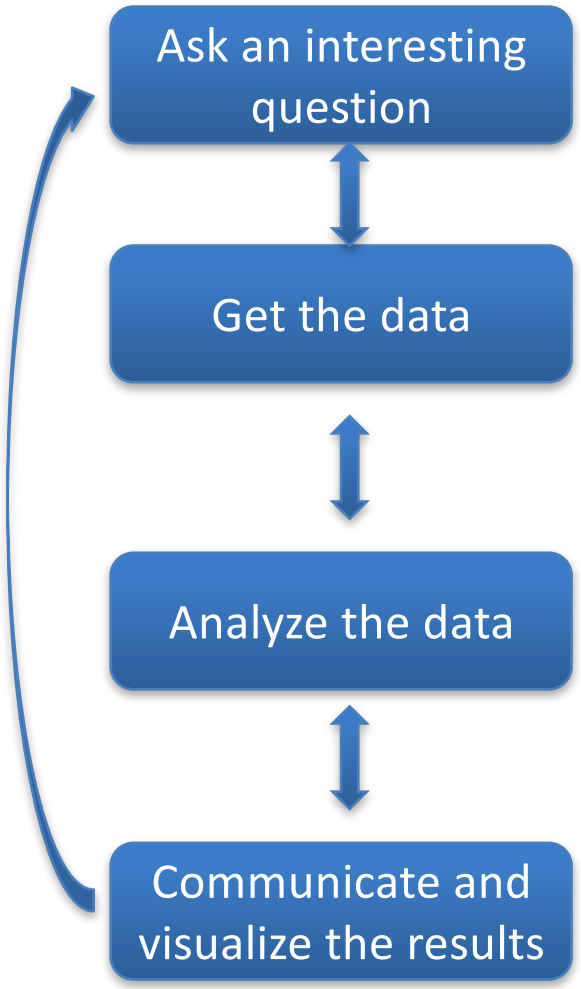
# Data Science pipeline

**1. Ask an interesting question:**

- What is the goal?
- What would you do if you had all the data?
- What do you want to predict or estimate?

**2. Get the data:**

- How were the data sampled?
- Which data are relevant?
- Are there privacy issues?



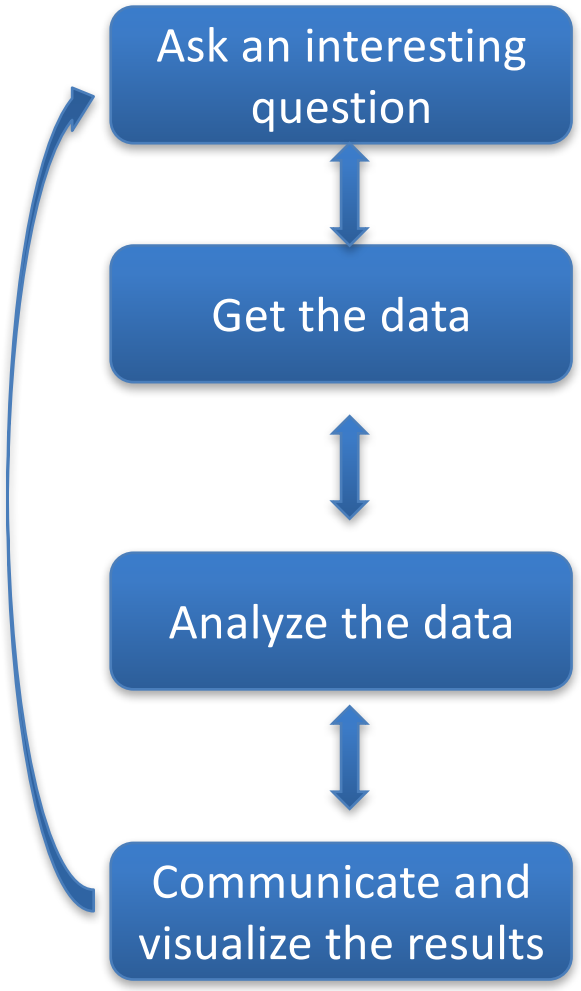
# Data Science pipeline

**3. Analyze the data:**

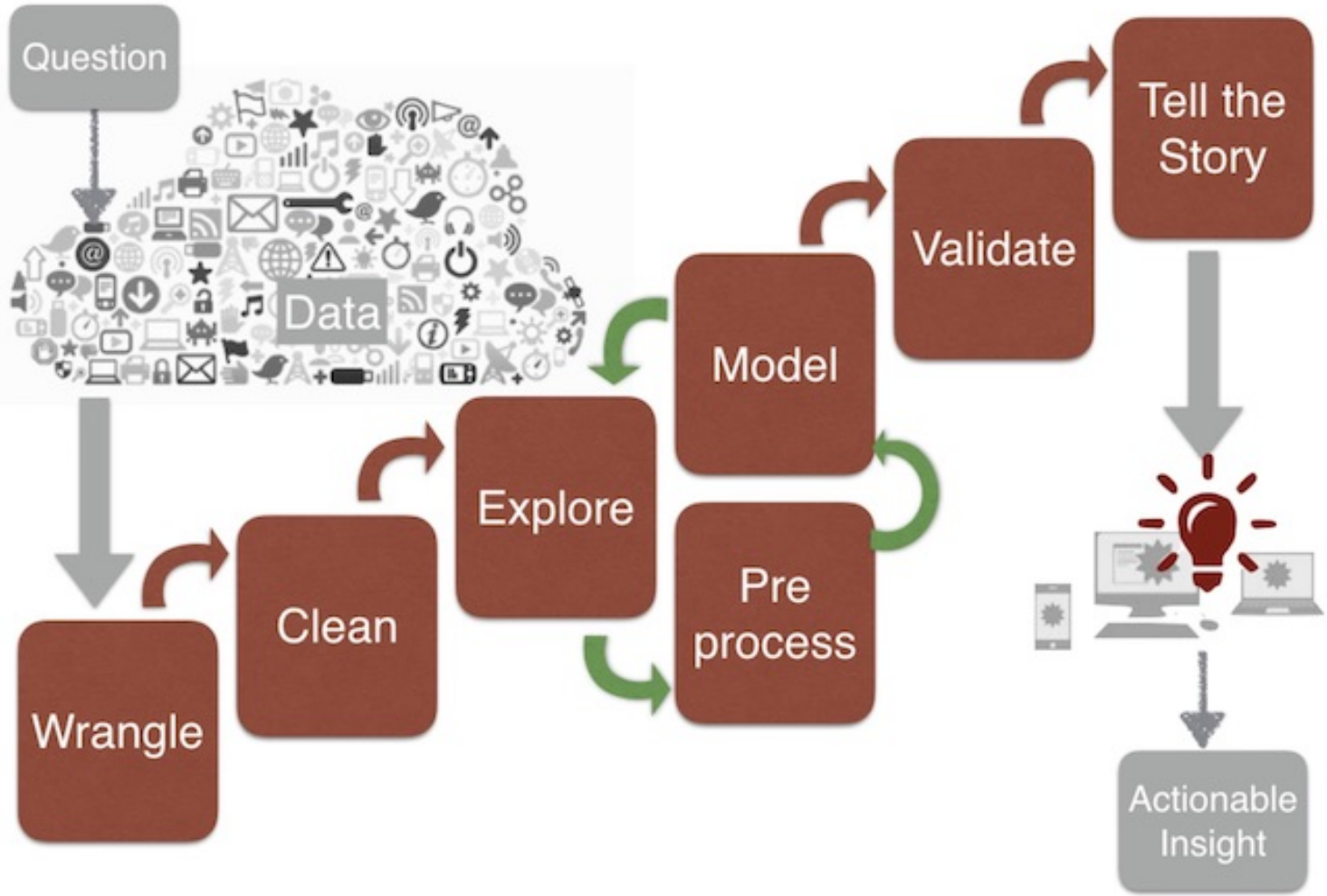
- Are there anomalies?
- Are there patterns?
- Are there trends?

**4. Communicate and visualize the results**

- What did we learn?
- Do the results make sense?
- Can we tell a story?

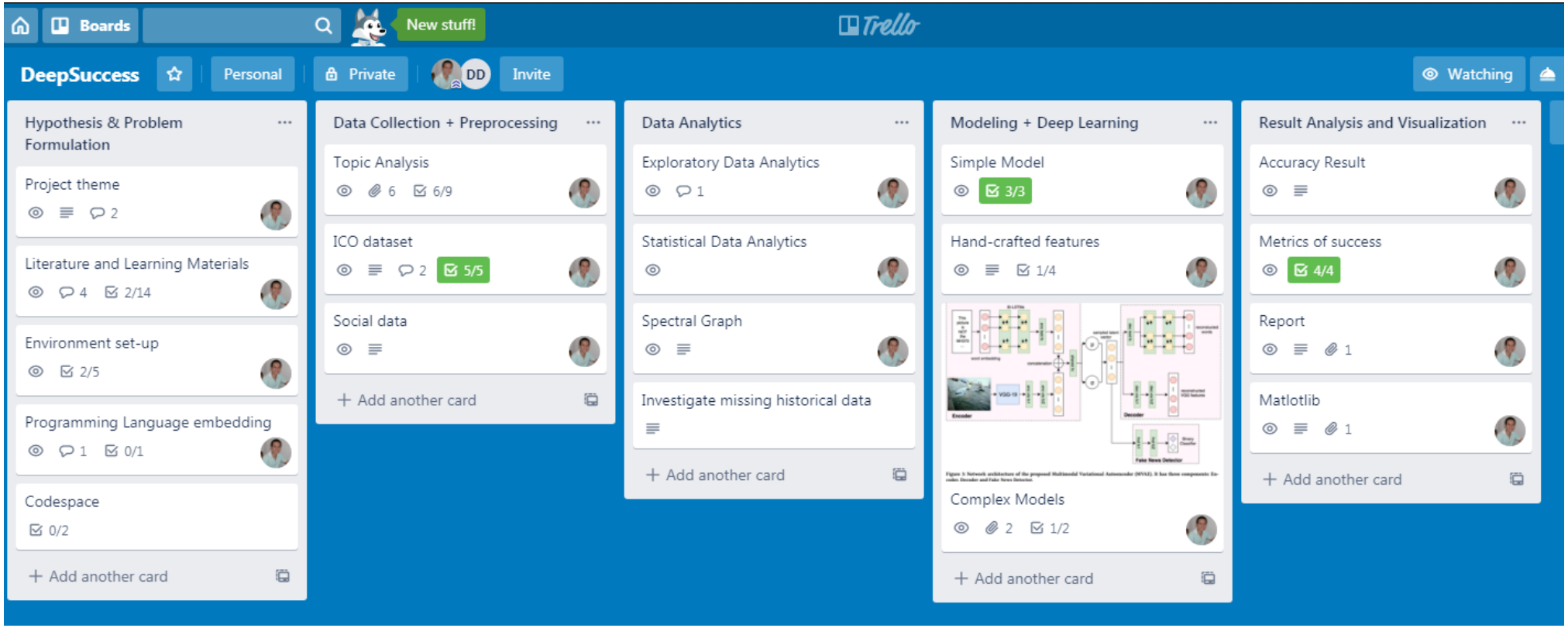


# Another presentation of Data Science Pipeline



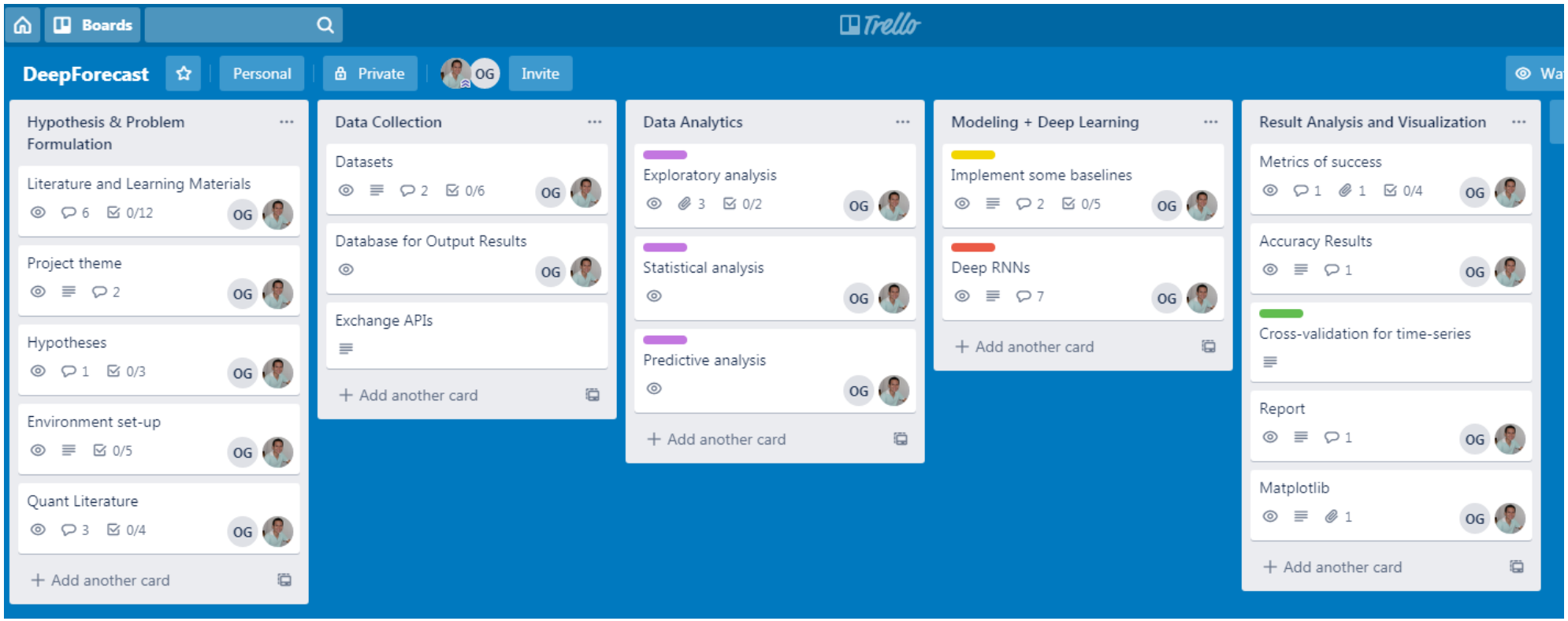
# Data Science Projects

- Project dashboard: Trello
- Codespace: Github
- Dataspace: Google Drive
- Communication: Slack



# Data Science Projects

- Free tools
- Suitable for research purpose



# Skill Set of Data Science

- A data scientist requires a large body of skill sets …
  - Data manipulation:
    - Modern programming languages: Python, R, C#, Scala, etc.
    - Crawling, cleaning, parsing, representing, scrapping, etc.
  - **Data analytics: → the focus of this course**
    - Ask the right questions, construct hypothesis
    - Find dependency, correlation, perform statistical analysis, exploratory data analysis
    - Algorithms and models: from classical tools such as logistic regression to modern machine learning (graphical models, SVM, gradient boosted tree)
  - Communication of results
    - Visualization: charting, graphing, interactive graphics, tools,
    - Present your analysis, results, talk to business partners, etc.

# II. Data Analytics and Applications

- "Data Analytics is the process of inspecting, cleaning, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making."

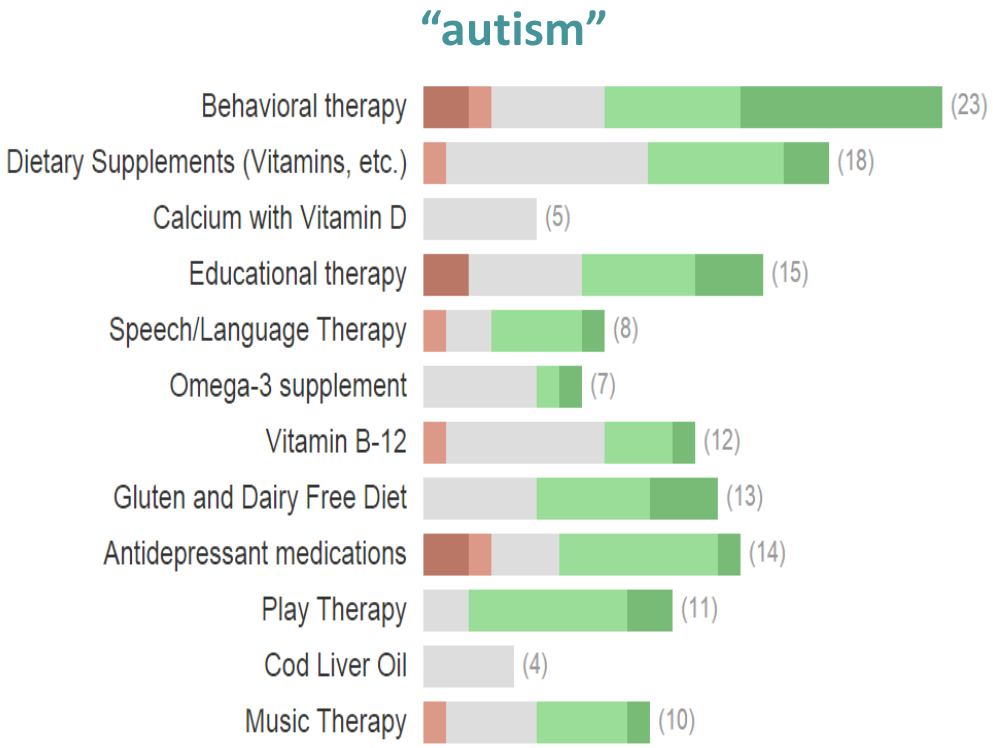
→ To gain insights into data through computation, statistics, and visualization

# Application: collective intelligence

- Patients come together to share quantitative medical data
  - Patients vote on effectiveness of each treatment
  - Use data from the crowd to derive new insights

CureTogether

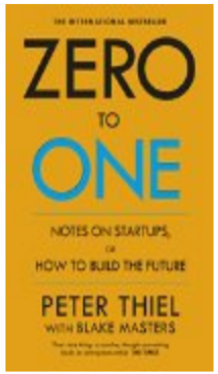
curetogether.com



# Application: product recommendation

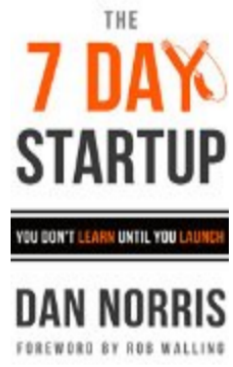
**Related to Items You've Viewed**

You viewed




Zero to One: Notes on Start Ups, or...
Blake Masters, Peter Thiel
(6)
Kindle Price: **$12.93**

Customers who viewed this also viewed




The 7 Day Startup: You Don't Learn...
Dan Norris, Rob Walling
(16)
Kindle Price: **$4.56**


The Lean Startup: How Constant...
Eric Ries
(9)
Kindle Price: **$16.14**

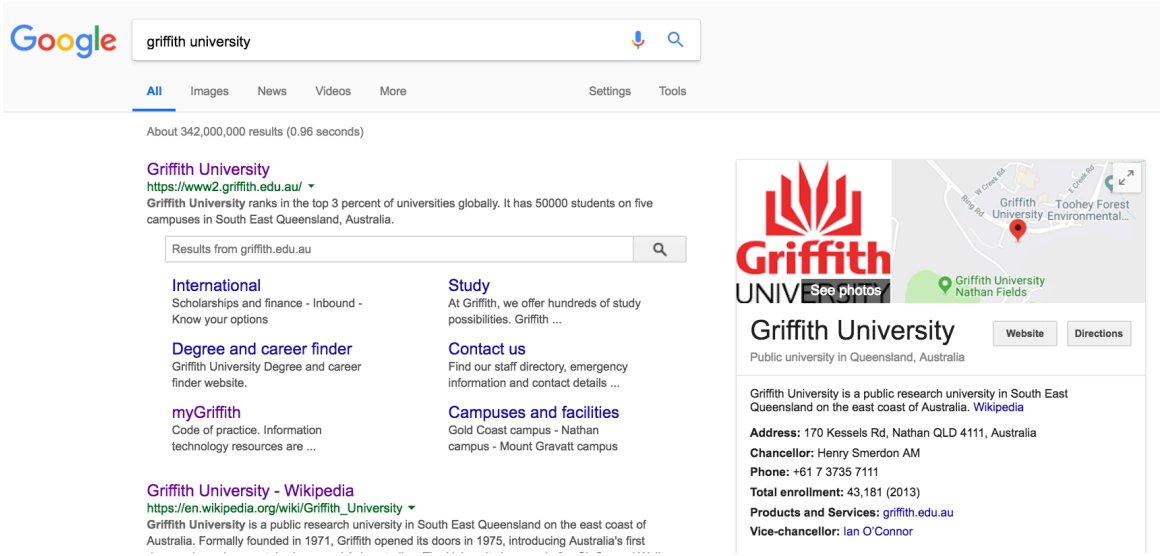

Elon Musk: How the Billionaire CEO of...
Ashlee Vance
(2)
Kindle Price: **$17.09**

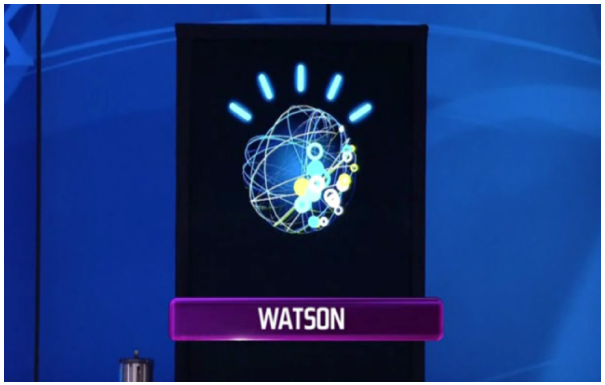
amazon.com®
and you're done.™

- Amazon uses data to build recommender system
  - Apply data analytics to understand customer behaviours
  - Recommend similar products from users with similar behaviours

# Application: knowledge base

- Apply data analytics to derive new knowledge



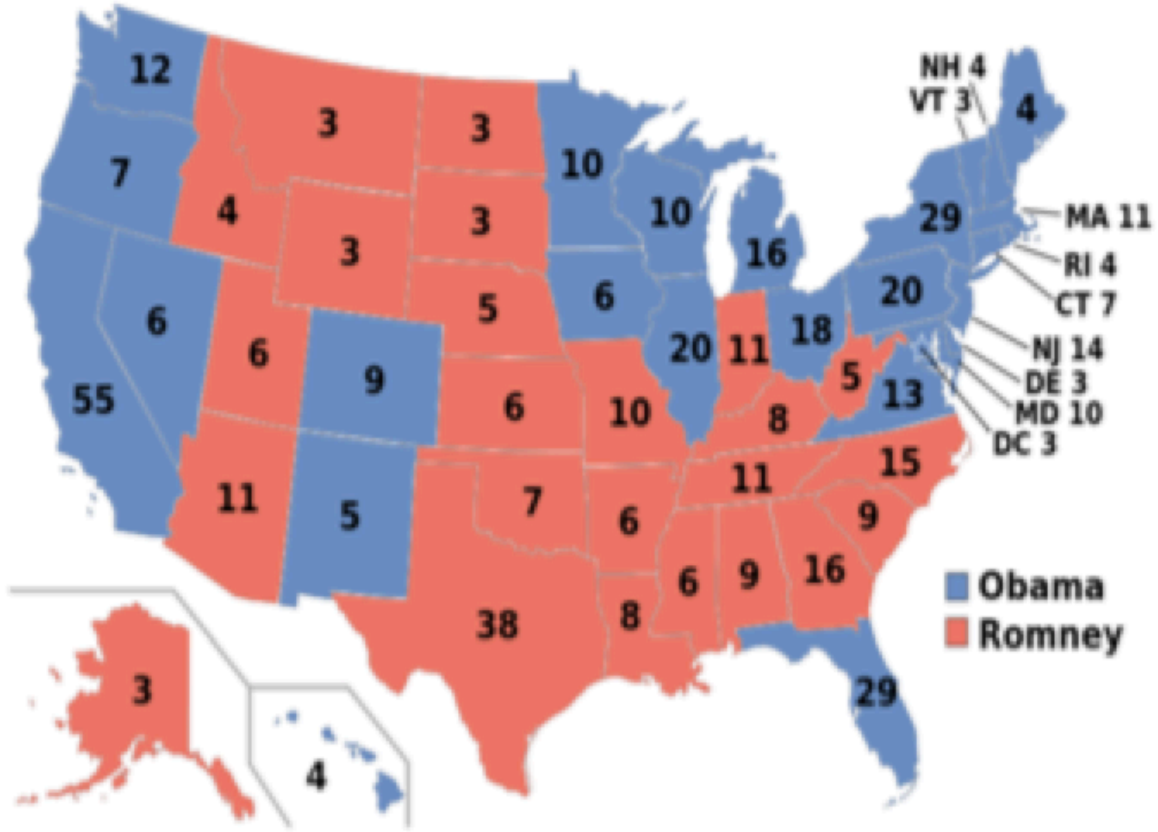

Google Knowledge Graph: provide description of all entities such as people, places, and organizations

IBM Watson – Q&A service: automatically answer a textual question

# Application: predict elections

Silver, who made his name by using cold hard math (historical data) to predict elections correctly in 49 out of 50 states in the 2008 and all 50 states in 2012

http://www.slate.com/articles/news_and_politics/politics/2016/01/nate_silver_said_donald_trump_had_no_shot_where_did_he_go_wrong.html

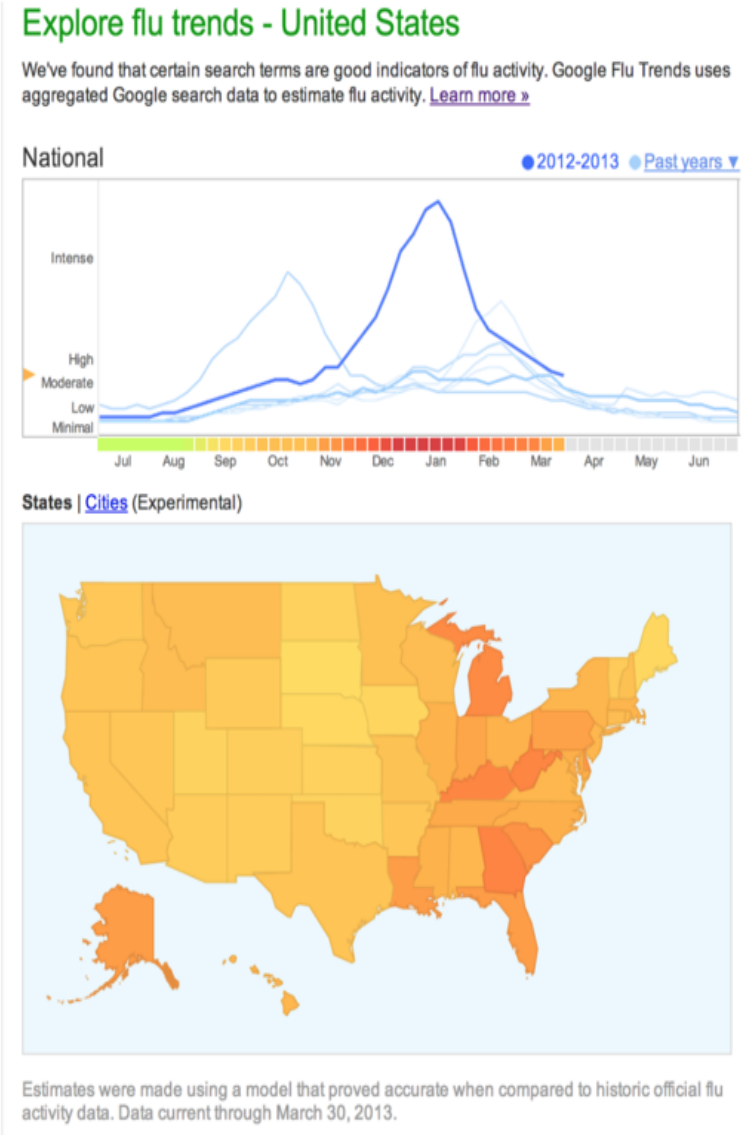



https://en.wikipedia.org/wiki/United_States_presidential_election,_2012

# Application: Flu Monitoring

Google Flu Trend: provide estimates of influenza activity for more than 25 countries

- Use the search queries related to flu on Google
- Users tend to search information for potential flu outbreaks
- Predict flu at a location based on the number of queries and their IP location



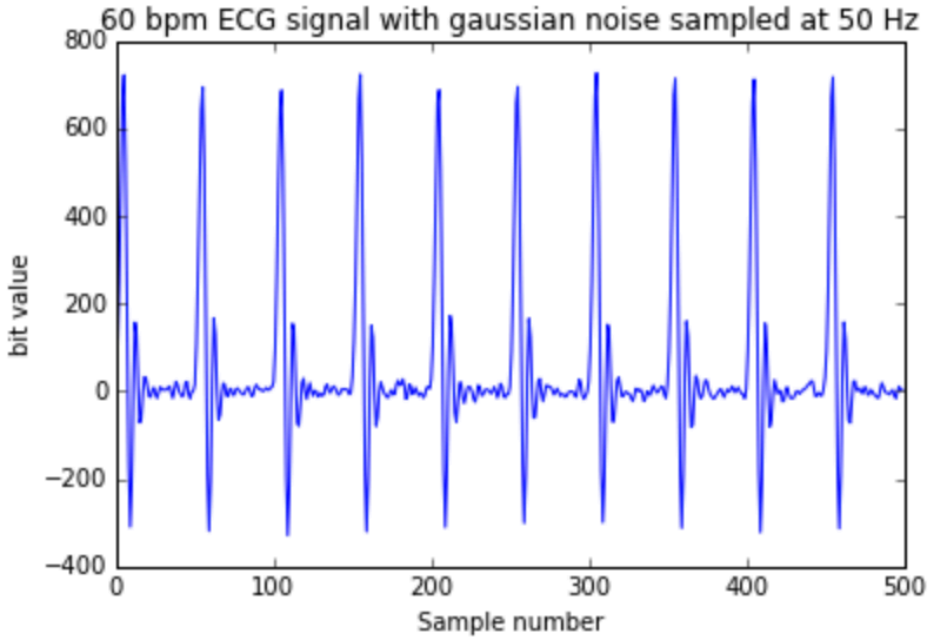

https://en.wikipedia.org/wiki/Google_Flu_Trends

# III. Data Types

- Numerical data
- Categorical data
- Ordinal data
- Textual data
- Graph data

# Numerical

- Represents quantitative measurement
  - Heights of people, page load times, stock prices, etc.
- Two types:
  - Discrete Data: integer based; often counts of some events
    - How many purchases did a customer make in a year?
    - How many times did I flip “heads”?
  - Continuous Data: has an infinite number of possible values
    - How much time did it take for a user to check out?
    - How much rain fell on a given day?

# Categorical

- Qualitative data without inherent mathematical meaning
  - Gender, yes/no (binary data), race, state of residence, product category, political party, etc.
- Compact representation: assign numbers to categories
  - Without mathematical meaning

# Ordinal

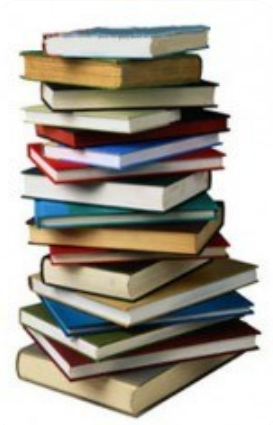
- A mixture of numerical and categorical
- Ordinal data that has mathematical meaning
- Example: movie ratings on a 1-5 scale
  - Ratings must be 1,2,3,4, or 5
  - These values have mathematical meaning
    - 1 means it's a worse movie than a 2.

# Textual Data

- Huge amount of diverse textual data produced every second for wide range of sources


Social media streams

Emails



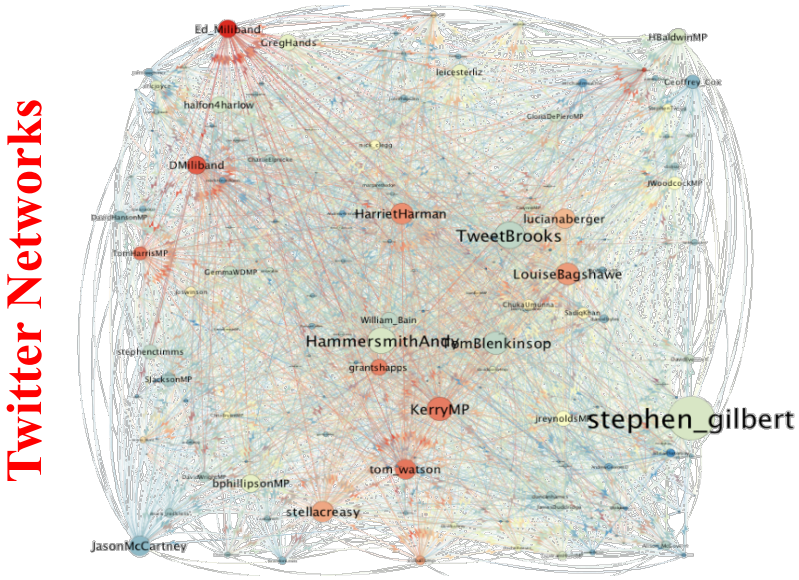
Books

Newspapers

Research articles

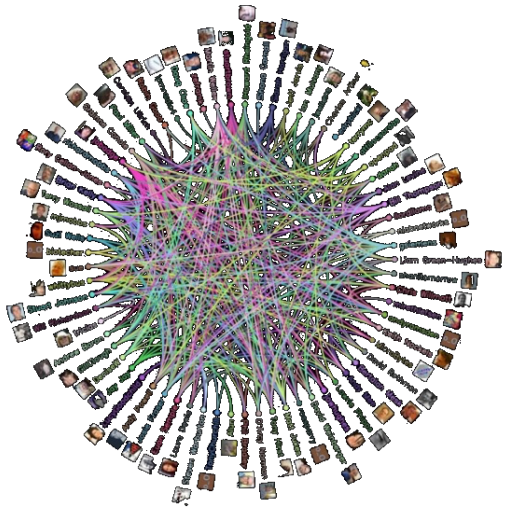
# Graph Data

- E.g. social networks




UK Members of Parliament's Twitter Network

http://blog.ouseful.info/2010/09/13/first-pass-quick-look-at-the-uk-mps-twitter-network/



Follower network of a UK photographer

http://scienceoftheinvisible.blogspot.com/2008/05/network-arithmetic.html

# IV. Recap: Statistics

- Mean, Median, Mode
- Variance and Standard Deviation
- Probability Density Function
- Probability Mass Function
- Percentiles

# Mean

- AKA Average
- Sum / number of samples
- Example:
  - Samples of student grades in Griffith University:

    6, 5, 4, 5, 7, 6, 6, 5, 6
  - The MEAN is (6+5+4+5+7+6+6+5+6) / 9 = 5.56

# Median

- Sort the values, and take the value at the mid point
- Example:

6, 5, 4, 5, 7, 6, 6, 5, 6

Sort it:

4, 5, 5, 5, 6, 6, 6, 6, 7

Median = 6

# Median (cont'd)

- If you have an even number of samples, take the average of the two in the middle
- Median is less sensitive to outliers than the mean
  - Example: mean household income in the US is $72,641, but the median is only $51,939, because the mean is skewed by a handful of billionaires
  - Median better represents the "typical" American in this example

# Mode

- The most common value in a dataset
  - Not relevant to continuous numerical data
- Back to our student grades example:

6, 5, 4, 5, 7, 6, 6, 5, 6

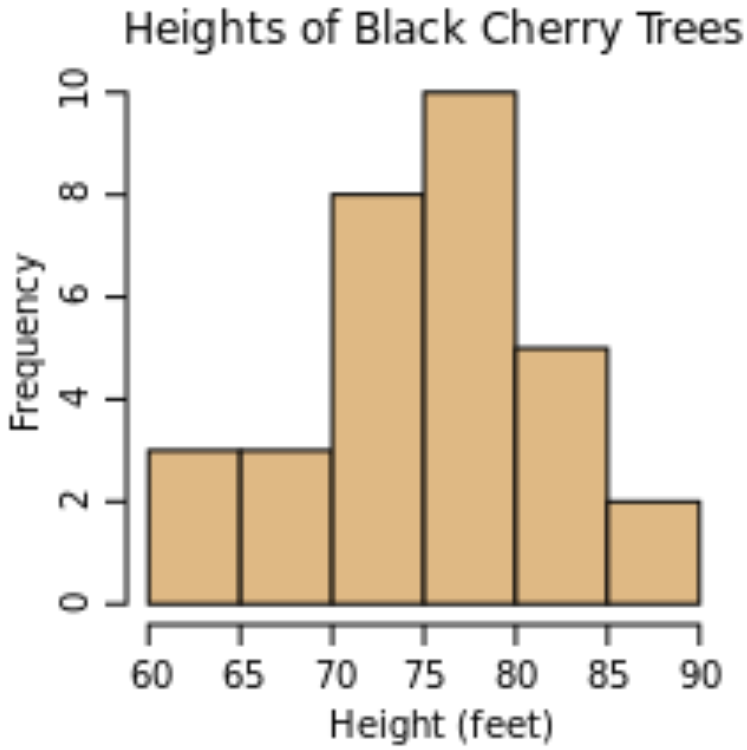
How many of each value are there?

4: 1, 5: 3, 6: 4, 7: 1

The MODE is 6

# Variance and Standard Deviation

- We want to know the spread of a data, i.e. the shape of the distribution of a dataset.




https://en.wikipedia.org/wiki/Histogram

# Variance measures how "spread-out" the data is

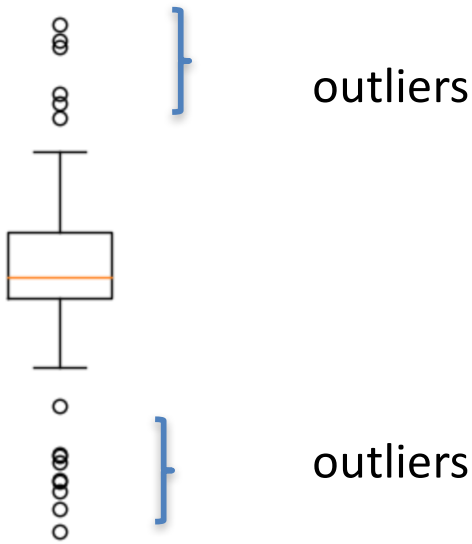
- Variance $\sigma^2$
  - is simply the **average of the squared differences from the mean**
- Example: what is the variance of the dataset (6, 5, 4, 5, 7, 6, 6, 5, 6)?
  - First find the mean: (6, 5, 4, 5, 7, 6, 6, 5, 6)/9 = 5.56
  - Now find the differences from the mean:

    (0.44, -0.56, -1.56, -0.56, 1.44, 0.44, 0.44, -0.56, 0.44)
  - Find the squared differences:

    (0.19, 0.31, 2.43, 0.31, 2.07, 0.19, 0.19, 0.31, 0.19)
  - Find the average of the squared differences:
    - $\sigma^2$ = (0.19+0.31+2.43+0.31+2.07+0.19+0.19+0.31+0.19) / 9 = 0.69

# Standard Deviation is just the square root of the variance

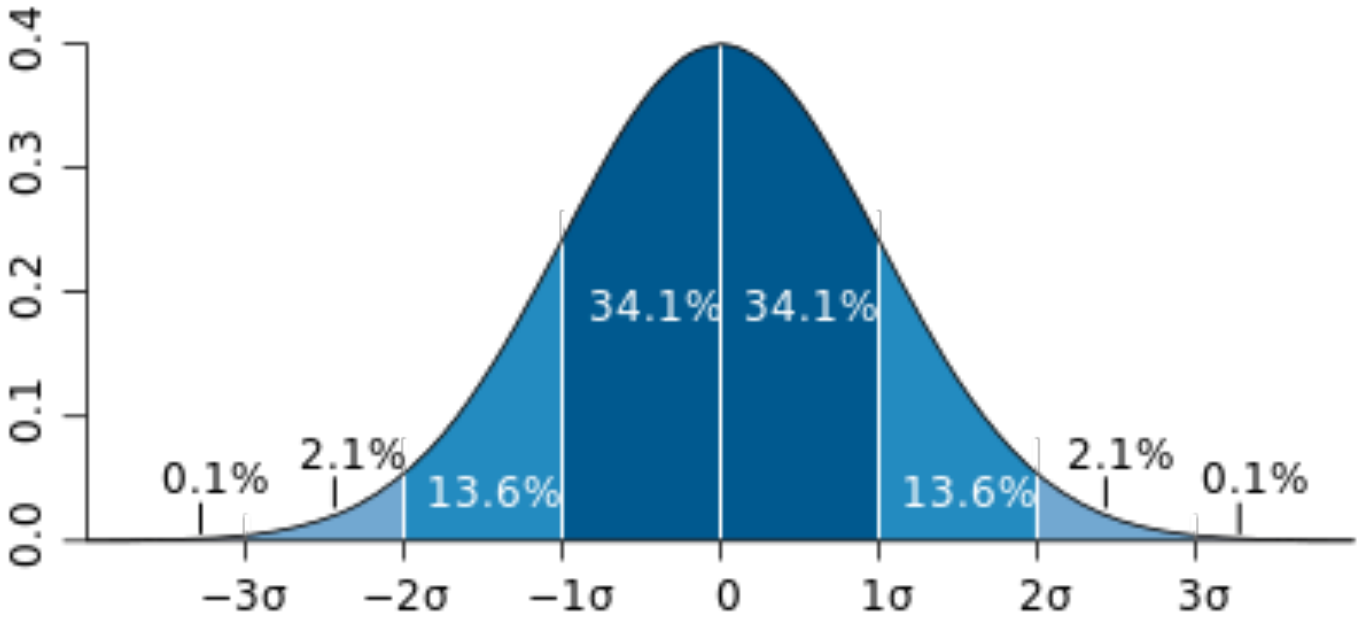
- $\sigma^2$ = 0.69
- $\sigma$ = sqrt(0.69) = 0.83
- So the standard deviation of (6, 5, 4, 5, 7, 6, 6, 5, 6) is 0.83
- Often used to identify outliers
  - E.g. Data points that lie more than one standard deviation from the mean are considered outliers




https://matplotlib.org/examples/pylab_examples/boxplot_demo.html

# Probability Density Function

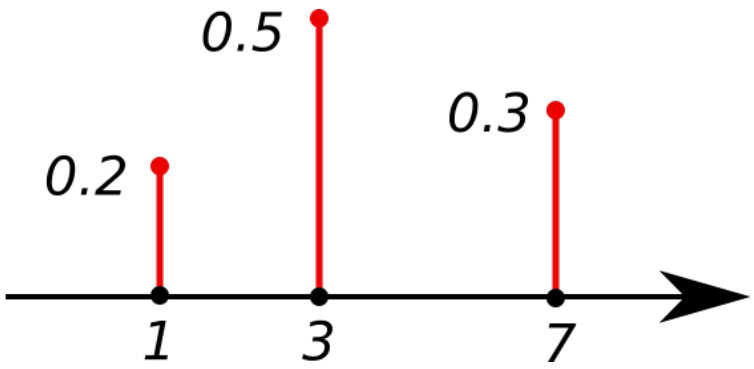
- Gives you the probability of a data point falling within some given range of a given value
- Often used for continuous data
- Example: a “normal distribution”



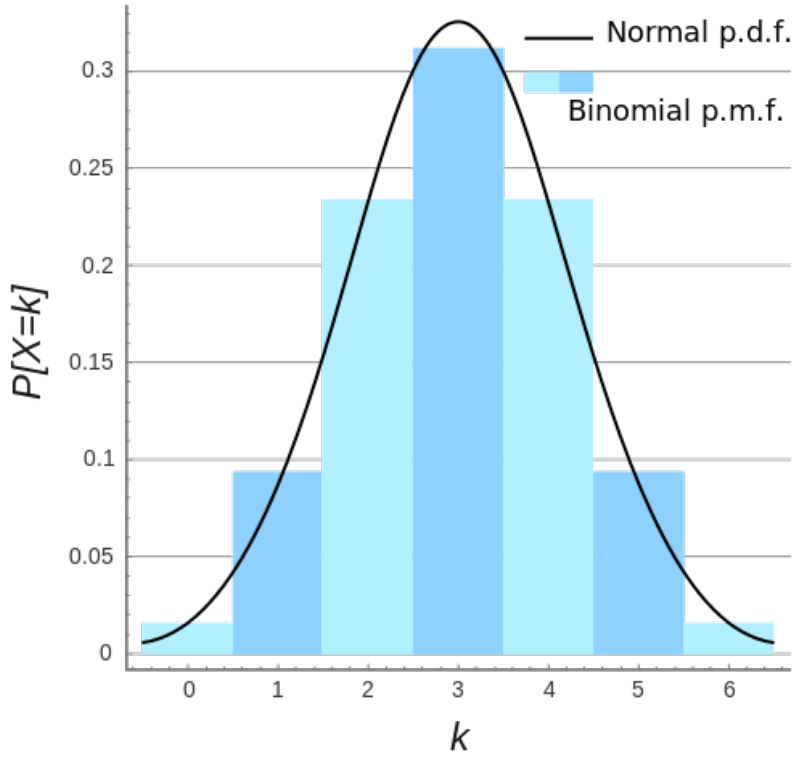

https://en.wikipedia.org/wiki/Standard_deviation

# Probability Mass Function

- Gives you the probability of a data point is exactly equal to some value
- Often used for discrete data




PMF representation



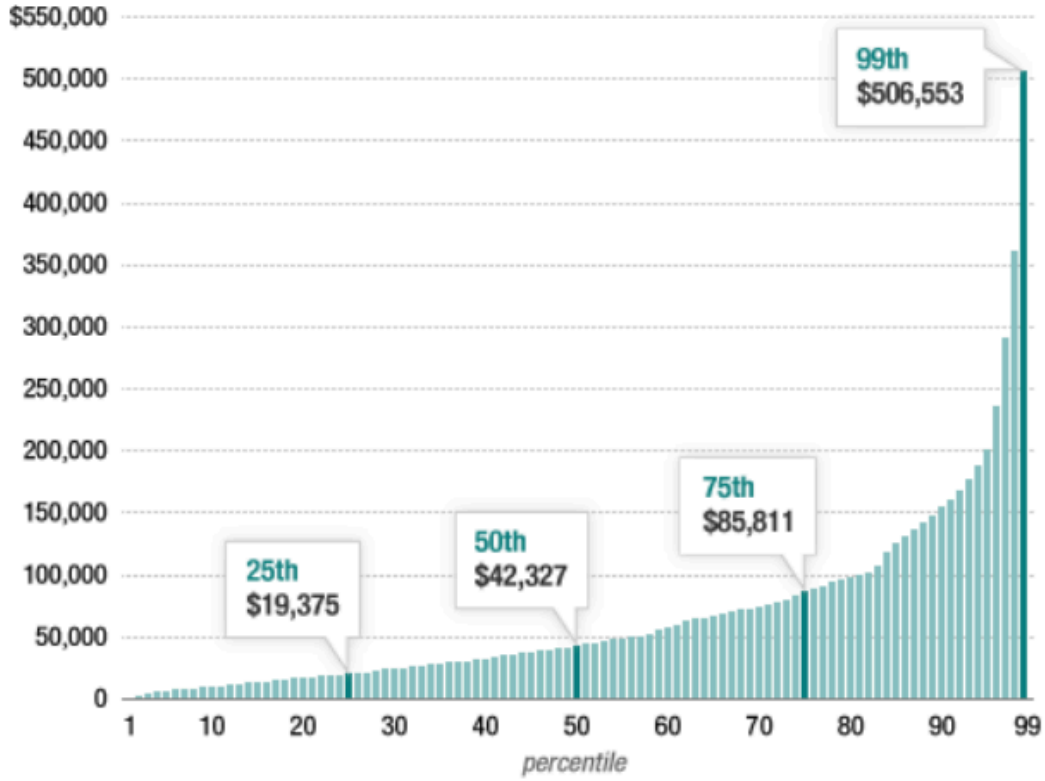

Binomial PMF can be approximated by a normal PDF → easier to calculate some statistics

https://en.wikipedia.org/wiki/Probability_mass_function

https://en.wikipedia.org/wiki/Binomial_distribution

# Percentiles

- X-th percentile = the data point below which X% of the values may be found
- Example: income distribution
  - 25-th percentile = $19,375
  - 75-th percentile = $85,811




https://en.wikipedia.org/wiki/Percentile

# V. Recap: Relational Database

- Use table to represent data
- A data row is referred to as a tuple
- Column name is referred to as an attribute

Staff Branch

| staffNo | sName | position | salary | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 | 22 Deer Rd, London |
| SG37 | Ann Beech | Assistant | 12000 | B003 | 163 Main St, Glasgow |
| SG14 | David Ford | Supervisor | 18000 | B003 | 163 Main St, Glasgow |
| SA9 | Mary Howe | Assistant | 9000 | B007 | 16 Argyll St, Aberdeen |
| SG5 | Susan Brand | Manager | 24000 | B003 | 163 Main St, Glasgow |
| SL41 | Julie Lee | Assistant | 9000 | B005 | 22 Deer Rd, London |

# Relational Model

- No duplicate rows in a table
- No duplicate columns in a table
- Every table has a unique name
- Every column within a table has a unique name

Staff Branch

| staffNo | sName | position | salary | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 | 22 Deer Rd, London |
| SG37 | Ann Beech | Assistant | 12000 | B003 | 163 Main St, Glasgow |
| SG14 | David Ford | Supervisor | 18000 | B003 | 163 Main St, Glasgow |
| SA9 | Mary Howe | Assistant | 9000 | B007 | 16 Argyll St, Aberdeen |
| SG5 | Susan Brand | Manager | 24000 | B003 | 163 Main St, Glasgow |
| SL41 | Julie Lee | Assistant | 9000 | B005 | 22 Deer Rd, London |

# Relational Model

- Primary key: uniquely identify all other attributes in a given row

Staff Branch

| staffNo | sName | position | salary | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 | 22 Deer Rd, London |
| SG37 | Ann Beech | Assistant | 12000 | B003 | 163 Main St, Glasgow |
| SG14 | David Ford | Supervisor | 18000 | B003 | 163 Main St, Glasgow |
| SA9 | Mary Howe | Assistant | 9000 | B007 | 16 Argyll St, Aberdeen |
| SG5 | Susan Brand | Manager | 24000 | B003 | 163 Main St, Glasgow |
| SL41 | Julie Lee | Assistant | 9000 | B005 | 22 Deer Rd, London |

# Data inside a table

- Atomic value in each cell
  - Atomic: single, non-decomposable
- A special value Null can be used if the value is unknown or does not exist

staff

| ID | name | office | phone | Works at |
|---|---|---|---|---|
| S001 | Lisa | G17_1.45 | 28673 | Griffith uni |
| S002 | Luke | G39_2.48 | Null | Griffith uni |

# Why Relational Model

- Avoid data redundancy

Staff Branch

| staffNo | sName | position | salary | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 | 22 Deer Rd, London |
| SG37 | Ann Beech | Assistant | 12000 | B003 | 163 Main St, Glasgow |
| SG14 | David Ford | Supervisor | 18000 | B003 | 163 Main St, Glasgow |
| SA9 | Mary Howe | Assistant | 9000 | B007 | 16 Argyll St, Aberdeen |
| SG5 | Susan Brand | Manager | 24000 | B003 | 163 Main St, Glasgow |
| SL41 | Julie Lee | Assistant | 9000 | B005 | 22 Deer Rd, London |

Data as a single table

Staff

| staffNo | sName | position | salary | branchNo |
|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 |
| SG37 | Ann Beech | Assistant | 12000 | B003 |
| SG14 | David Ford | Supervisor | 18000 | B003 |
| SA9 | Mary Howe | Assistant | 9000 | B007 |
| SG5 | Susan Brand | Manager | 24000 | B003 |
| SL41 | Julie Lee | Assistant | 9000 | B005 |

Branch

| branchNo | bAddress |
|---|---|
| B005 | 22 Deer Rd, London |
| B007 | 16 Argyll St, Aberdeen |
| B003 | 163 Main St, Glasgow |

Relational model: two tables

- No need to write bAddress for every staff

# Why Relational Model?

- Store multi-value data

staff

| ID | name | office | phone | Works at |
|---|---|---|---|---|
| S001 | Lisa | G17_1.45 | 28673 | Griffith uni |
| S001 | Lisa | G17_1.45 | 25678 | Griffith uni |
| S002 | Luke | G39_2.48 | Null | Griffith uni |


staff

| ID | name | office | Works at |
|---|---|---|---|
| S001 | Lisa | G17_1.45 | Griffith uni |
| S002 | Luke | G39_2.48 | Griffith uni |

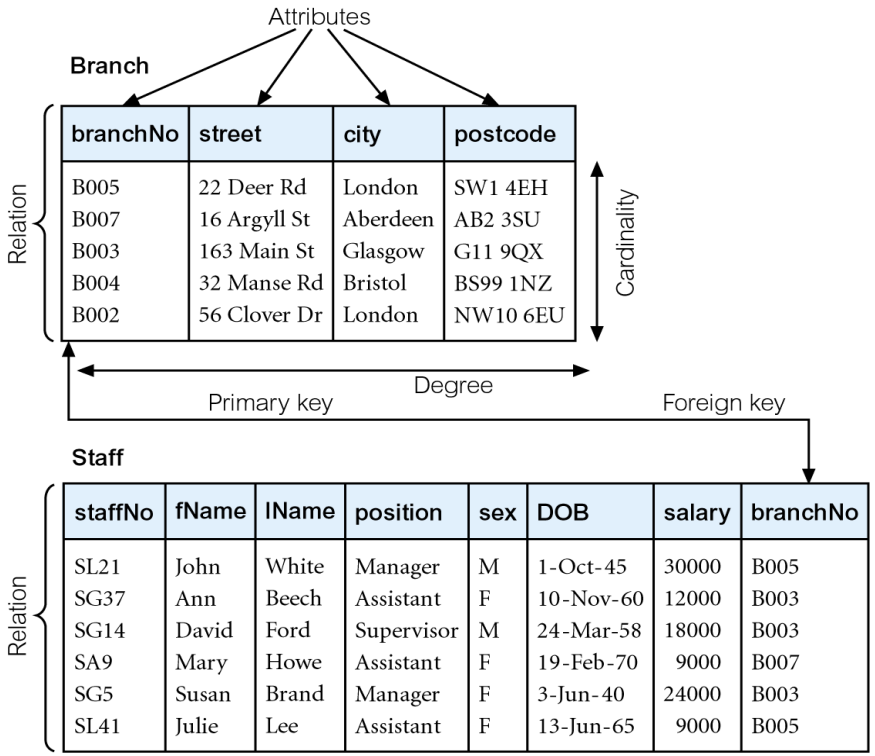
staff-phone

| ID | phone |
|---|---|
| S001 | 28673 |
| S001 | 25678 |
| S002 | Null |

# Relational Model

- A relational model is a collection of tables/relations(entities)
- The relations in a database are inter-related somehow, e.g., via *foreign keys*
  - Primary key is used to identify a unique row in a table
  - Foreign key is used to link a unique row in one table to a row in another table

- Relational database management system :
  - MySQL, PostgreSQL, Oracle, DB2, SQLite, …




| branchNo | street | city | postcode |
|---|---|---|---|
| B005 | 22 Deer Rd | London | SW1 4EH |
| B007 | 16 Argyll St | Aberdeen | AB2 3SU |
| B003 | 163 Main St | Glasgow | G11 9QX |
| B004 | 32 Manse Rd | Bristol | BS99 1NZ |
| B002 | 56 Clover Dr | London | NW10 6EU |

| staffNo | fName | lName | position | sex | DOB | salary | branchNo |
|---|---|---|---|---|---|---|---|
| SL21 | John | White | Manager | M | 1-Oct-45 | 30000 | B005 |
| SG37 | Ann | Beech | Assistant | F | 10-Nov-60 | 12000 | B003 |
| SG14 | David | Ford | Supervisor | M | 24-Mar-58 | 18000 | B003 |
| SA9 | Mary | Howe | Assistant | F | 19-Feb-70 | 9000 | B007 |
| SG5 | Susan | Brand | Manager | F | 3-Jun-40 | 24000 | B003 |
| SL41 | Julie | Lee | Assistant | F | 13-Jun-65 | 9000 | B005 |

# Query data in relation model

- SQL provides language for core data manipulations
- SQL is a declarative language: you think about what you want, not how to compute it

# Query data in relation model

Imperative

```
//dogs = [{name: 'Fido', owner_id: 1}, {...}, ... ]
//owners = [{id: 1, name: 'Bob'}, {...}, ...]

var dogsWithOwners = []
var dog, owner

for(var di=0; di < dogs.length; di++) {
  dog = dogs[di]

  for(var oi=0; oi < owners.length; oi++) {
    owner = owners[oi]
    if (owner && dog.owner_id == owner.id) {
      dogsWithOwners.push({
        dog: dog,
        owner: owner
      })
    }
  }}
}
```

Declarative

```
SELECT * from dogs
INNER JOIN owners
WHERE dogs.owner_id = owners.id
```

From: http://latentflip.com/imperative-vs-declarative/

# SQL: Selection

Format

```
SELECT   target-list
FROM     relation-list
WHERE    condition
```

Example

```
SELECT   sName
FROM     staff
WHERE    salary >= 10000
```

*relation-list*: A list of relation names

*target-list*: A list of attributes of tables in *relation-list*

*condition*: Comparisons combined using AND, OR and NOT.

- Comparisons are Attr *op* const or Attr1 *op* Attr2, where *op* is one of =≠<>≤≥

# SQL: Selection

```
SELECT    sName
FROM      staff
WHERE     salary >= 10000
```

Staff

| | staffNo | sName | position | salary | branchNo |
|---|---|---|---|---|---|
| ✓ | SL21 | John White | Manager | 30000 | B005 |
| ✓ | SG37 | Ann Beech | Assistant | 12000 | B003 |
| ✓ | SG14 | David Ford | Supervisor | 18000 | B003 |
| | SA9 | Mary Howe | Assistant | 9000 | B007 |
| ✓ | SG5 | Susan Brand | Manager | 24000 | B003 |
| | SL41 | Julie Lee | Assistant | 9000 | B005 |

# SQL: Joins and Inference

Chaining relations together is the basic inference method in relational DBs. It produces new relations (effectively new facts) from the data:

```
SELECT      S.sName, B.bAddress
 FROM       Staff S, Branch B
 WHERE      S.BranchNo=B.BranchNo
```

**Staff**

| staffNo | sName | salary | branchNO |
|---|---|---|---|
| SL21 | John White | 30000 | B005 |
| SA9 | Mary Howe | 9000 | B007 |
| SL41 | Julie Lee | 9000 | B005 |

**Branch**

| branchNO | bAddress |
|---|---|
| B005 | 123 Queen Street |
| B007 | 321 King Street |

# SQL: Joins and Inference

```
SELECT    S.sName, B.bAddress
FROM      Staff S, Branch B
WHERE     S.BranchNo=M.BranchNo
```

Staff

| staffNo | sName | salary | branchNO |
|---|---|---|---|
| SL21 | John White | 30000 | B005 |
| SA9 | Mary Howe | 9000 | B007 |
| SL41 | Julie Lee | 9000 | B005 |

Branch

| branchNO | bAddress |
|---|---|
| B005 | 123 Queen Street |
| B007 | 321 King Street |

Step 1

| staffNo | sName | salary | branchNO | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | 30000 | B005 | B005 | 123 Queen Street |
| SL21 | John White | 30000 | B005 | B007 | 321 King Street |
| SA9 | Mary Howe | 9000 | B007 | B005 | 123 Queen Street |
| SA9 | Mary Howe | 9000 | B007 | B007 | 321 King Street |
| SL41 | Julie Lee | 9000 | B005 | B005 | 123 Queen Street |
| SL41 | Julie Lee | 9000 | B005 | B007 | 321 King Street |

Step 2

| staffNo | sName | salary | branchNO | branchNo | bAddress |
|---|---|---|---|---|---|
| SL21 | John White | 30000 | B005 | B005 | 123 Queen Street |
| SA9 | Mary Howe | 9000 | B007 | B007 | 321 King Street |
| SL41 | Julie Lee | 9000 | B005 | B005 | 123 Queen Street |

Step 3

| sName | bAddress |
|---|---|
| John White | 123 Queen Street |
| Mary Howe | 321 King Street |
| Julie Lee | 123 Queen Street |

# SQL: Aggregations and GroupBy

- One of the most common operations on data tables is aggregation (**count, sum, average, min, max,**…).
- They provide a means to see high-level patterns in the data, to make summaries of it, etc.
- You need ways of specifying which columns are being aggregated over, which is the role of a **GroupBy** operator.

# SQL: Aggregations and GroupBy

Staff

| staffNo | sName | position | salary | branchNo |
|---|---|---|---|---|
| SL21 | John White | Manager | 30000 | B005 |
| SG37 | Ann Beech | Assistant | 12000 | B003 |
| SG14 | David Ford | Supervisor | 18000 | B003 |
| SA9 | Mary Howe | Assistant | 9000 | B007 |
| SG5 | Susan Brand | Manager | 24000 | B003 |
| SL41 | Julie Lee | Assistant | 9000 | B005 |

```
SELECT    branchNo, AVG(salary)
FROM      Staff
GROUP BY    branchNo
```

| branchNo | AVG(salary) |
|---|---|
| B005 | 19500 |
| B003 | 18000 |
| B007 | 9000 |

# SQL implementations



etc.

```
#!/usr/bin/python

import MySQLdb

# Open database connection
db = MySQLdb.connect("localhost","testuser","test123","TESTDB" )

# prepare a cursor object using cursor() method
cursor = db.cursor()

sql = "SELECT * FROM EMPLOYEE \
       WHERE INCOME > '%d'" % (1000)
try:
   # Execute the SQL command
   cursor.execute(sql)
   # Fetch all the rows in a list of lists.
   results = cursor.fetchall()
   for row in results:
      fname = row[0]
      lname = row[1]
      age = row[2]
      sex = row[3]
      income = row[4]
      # Now print fetched result
      print "fname=%s,lname=%s,age=%d,sex=%s,income=%d" % \
             (fname, lname, age, sex, income )
except:
   print "Error: unable to fecth data"

# disconnect from server
db.close()
```

# VI. Tools

1. Anaconda
2. Jupyter Notebook
3. Python Basics
4. Python libraries for data analytics

# Anaconda: our Python environment

- Will be used in lab sessions
- You can do it on your laptop:
  - https://www.anaconda.com/download
  - Version: Python 3.6!

# Install Anaconda

- This course uses Python
  - Make sure you have a Python development environment set up on your personal computer (we will do it for you in laboratory computers).
  - We will walk through installing a package called Anaconda which has both the development environment and all the Python packages you need pre-installed. It makes life really easy.
- **Anaconda Distribution by Continuum Analytics:**
  1. Go to the website: https://www.anaconda.com/download/
  2. Download the installer (Python 3.6 version) for your OS
  3. Run the installer

# Jupyter Notebook

- Jupyter Notebook is a web application for interactive data science
- Create documents that combine live-code with narrative text, equations, images, videos, and visualizations.
- Reproducible record of computations to share on GitHub, Dropbox, and Jupyter Notebook Viewer.
- Shareable: can be exported to PDF, HTML, etc.
- Interactive Widgets: code can produce rich output such as images, videos, LaTeX, and Javascript. Interactive widgets can be used to manipulate and visualize data in realtime.

# Python Basics

- We will revisit basics of Python programming
- This course is meant for students with some programming experience
- If you are completely new to programming, this course may be **too advanced** for you

# Python Basics: Topics Covered

- Data Types
  - Numbers
  - Strings
  - Print
  - Formatting
  - Lists
  - Dictionaries
  - Booleans
  - Tuples and Sets
- Comparison Operators
- If, elif, and else Statements
- For Loops
- While Loops
- range()
- List Comprehension
- Functions
- Lambda Expressions
- Map and Filter

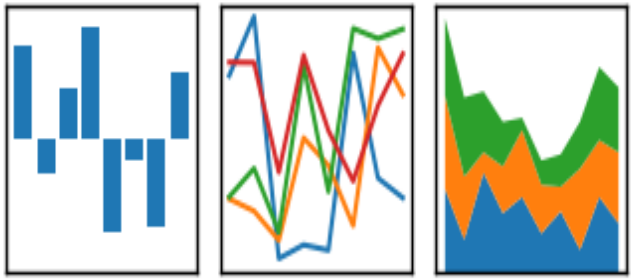
# Python libraries for Data Analytics

https://seaborn.pydata.org/



pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Numpy: great for handling numbers, vectors, matrices
- Scipy: great for numerical optimizations
- Pandas: great for handling tabular/relational data
- Scikit Learn: great for data analytics techniques

# References

[1] https://www.slideshare.net/e2m/introduction-to-ipython-jupyter-notebooks

[2] https://www.slideshare.net/mbussonn/jupyter-a-platform-for-data-science-at-scale