Introduction to Data Science and Analytics (DSC510)

Handling Data

George Pallis

Cooking with data

Part 2:
Data sources

Part 1:
Data models

Part 3:
Data wrangling

Cooking with data

Part 2:
Data sources

Part 1:
Data models

Part 3:
Data wrangling

Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

Data model

From Wikipedia, the free encyclopedia

A **data model** (or **datamodel**)[1][2][3][4][5] is an abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities. For instance, a data model may specify that the data element representing a car be composed of a number of other elements which, in turn, represent the color and size of the car and define its owner.

The term **data model** can refer to two distinct but closely related concepts. Sometimes it refers to an abstract formalization of the objects and relationships found in a particular application domain: for example the customers, products, and orders found in a manufacturing organization. At other times it refers to the set of concepts used in defining such formalizations: for example concepts such as entities, attributes, relations, or

A data model specifies how you think about the world

4 Topics
- 4.1 Data architecture
- 4.2 Data modeling
- 4.3 Data properties

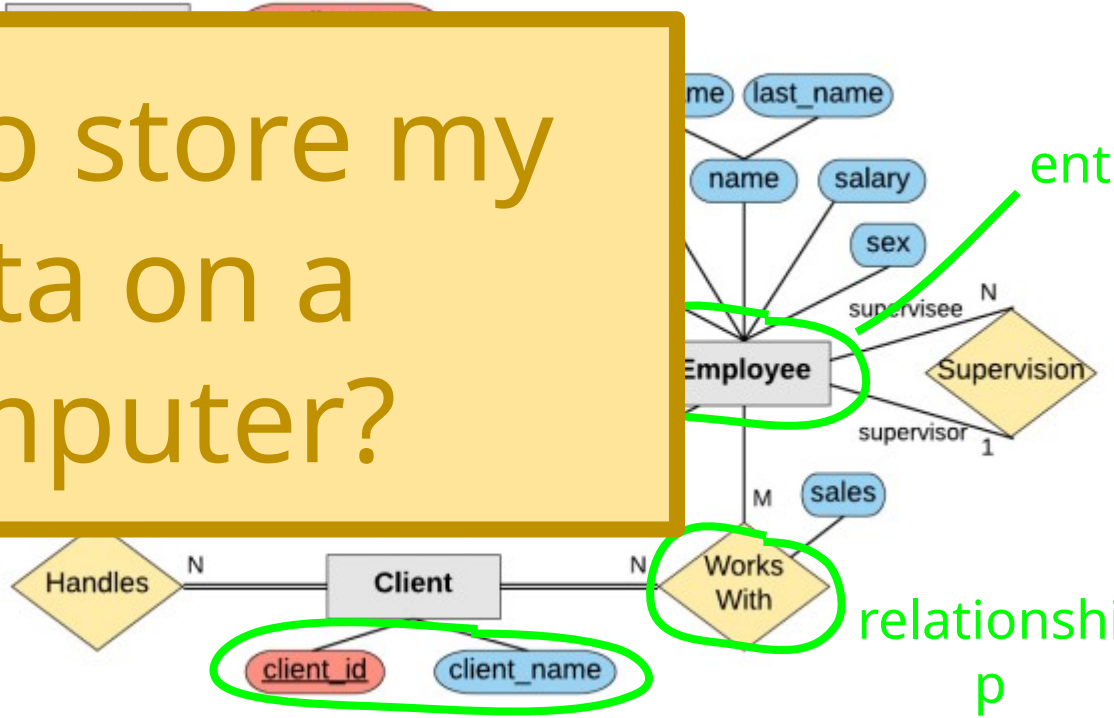
Data
Physical Model
O Data ructures

ed on int. A d, and of tion of a

functional specification to aid a computer software make-or-buy decision. The figure is an example of the interaction between process and data models.[6]

Q: “How do you think about the world?”
A: “See my entity–relationship diagram!”

How to store my data on a computer?

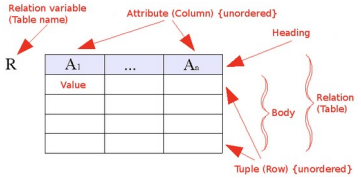


Q1: “How should I store my data on a computer?”
Q2: “How do I think about the world?”

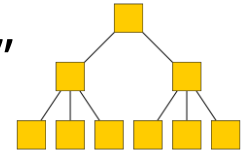
- “The world is simple: one type of entity, all with the same attributes”
  → **Flat model**

```
66.249.65.107 - -  08/Oct/2007:04:54:20 -0400] "GET /support.html
HTTP/1.1" 200 1117  "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

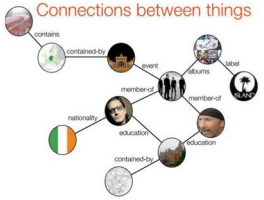
- “The world contains many types of entities, connected by relationships”
  → **Relational model**



- “The world is a hierarchy of entities”
  → **Document model**



- “The world is a complex network of entities”
  → **Network model**



Flat model

- Example: log files; e.g., Apache web server (`httpd`)
  - Entities = requests from clients to server

```
66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html HTTP/1.1"
200 11179 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"

111.111.111.111 - - [08/Oct/2007:11:17:55 -0400] "GET / HTTP/1.1" 200 10801
"http://www.google.com/search?
q=in+love+with+ada+lovelace+what+to+do&ie=utf-8&oe=utf-
8&aq=t&rls=org.mozilla:en-US:official&client=firefox-a" "Mozilla/5.0
(Windows; U; Windows NT 5.2; en-US; rv:1.8.1.7) Gecko/20070914
Firefox/2.0.0.7"
```

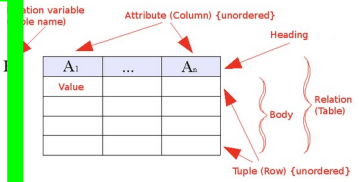
- Another common format: CSV (“comma-separated vector”)

Q1: “How should I store my data on a computer?”
Q2: “How do I think about the world?”

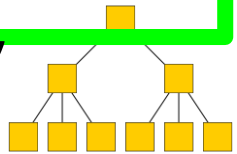
- “The world is simple: one type of entity, all with the same attributes”
  → **Flat model**

```
66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html
HTTP/1.1" 200 11179 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

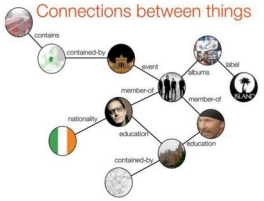
- “The world contains many types of entities, connected by relationships”
  → **Relational model**



- “The world is a hierarchy of entities”
  → **Document model**



- “The world is a complex network of entities”
  → **Network model**



Relational model

- "The world contains many types of entities, connected by relationships"
- The relational model is ubiquitous:
  - MySQL, PostgreSQL, Oracle, DB2, SQLite, …
  - You use it many times every day
- Data represented as tables ("relations") describing
  - entities,
  - relationships between entities
- Most of the data we will use can be "reduced" to the relational model

id	name
1	Bush
2	Trump
3	Obama

president	successor
1	3
3	2

Processing data in the relational model: SQL

- Declarative language for core data manipulations
- You think about what you want, not how to com

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

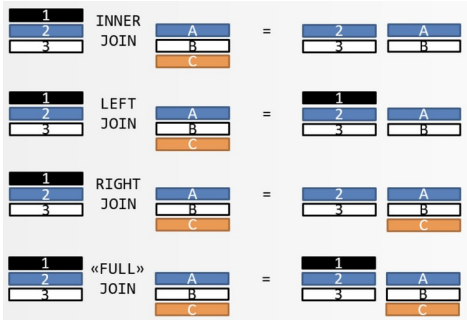
```
SELECT * from dogs
INNER JOIN owners
WHERE dogs.owner_id = owners.id
```

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

SQL

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

- Need a refresher? → Watch/do online tutorials!
- Key concepts:
  - Select (!), update, delete
  - Unique keys
  - Joins (inner, left outer, right outer, full)
  - Sorting
  - Aggregation (group by, count, min, max, avg, etc.)



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()
```

“SQL”: Pandas (Python library)

- Similar to SQL (declarative), with additional elements of functional programming (map(), filter(), etc.)
- SQL “table” ←→ Pandas “DataFrame”

Pandas vs. SQL

+ Pandas is lightweight and fast.

+ Natively Python, i.e., full SQL expressiveness plus the expressiveness of Python, especially for function evaluation.

+ Integration with plotting functions like Matplotlib.

- In Pandas, tables must fit into memory.

- No post-load indexing functionality: indices are built when a table is created.

- No transactions, journaling, etc. (matters for parallel applications)

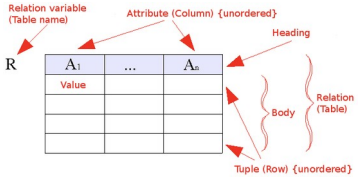
- Large, complex joins are slower.

Q1: “How should I store my data on a computer?”
Q2: “How do I think about the world?”

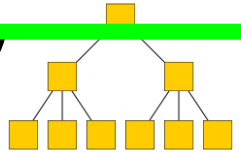
- “The world is simple: one type of entity, all with the same attributes”
  → **Flat model**

```
66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html
HTTP/1.1" 200 11179 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

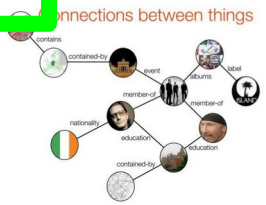
- “The world contains many types of entities, connected by relationships”
  → **Relational model**



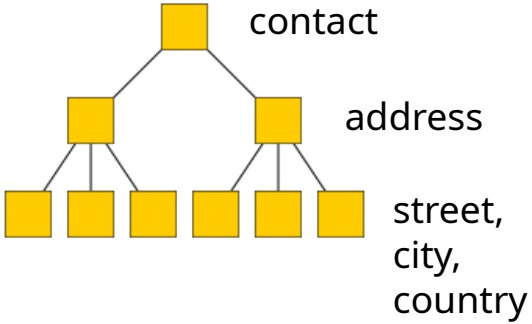
- “The world is a hierarchy of entities”
  → **Document model**



- “The world is a complex network of entities”
  → **Network model**



Document model



- "The world is a hierarchy of entities"
- XML format:

```
<contact>
   <id>656</id>
   <firstname>Chuck</firstname>
   <lastname>Smith</lastname>
   <phone>(123) 555-0178</phone>
   <phone>(890) 555-0133</phone>
   <address>
     <street>Rue de l'Ale 8</street>
     <city>Lausanne</city>
     <zip>1007</zip>
     <country>CH</country>
   </address>
</contact>
```

- JSON format:

```
contact: {
   id: 656,
   firstname: "Chuck",
   lastname: "Smith",
   phones: ["(123) 555-0178",
            "(890) 555-0133"],
   address: {
     street: "Rue de l'Ale 8",
     city: "Lausanne",
     zip: 1007,
     country: "CH"
   }
}
```

- Document model

```
<contact>
  <id>656</id>
  <firstname>Chuck</firstname>
  <lastname>Smith</lastname>
  <phone>(123) 555-0178</phone>
  <phone>(890) 555-0133</phone>
  <address>
    <street>Rue de l'Ale 8</street>
    <city>Lausanne</city>
    <zip>1007</zip>
    <country>CH</country>
  </address>
</contact>
```

Think for 1 minute:
If we want to use a relational DB (e.g., MySQL) instead of XML, how can we store 2 phone numbers for the same person?

Solution to puzzle

- Document model

```
<contact>
  <id>656</id>
  <firstname>Chuck</firstname>
  <lastname>Smith</lastname>
  <phone>(123) 555-0178</phone>
  <phone>(890) 555-0133</phone>
  <address>
    <street>Rue de l'Ale 8</street>
    <city>Lausanne</city>
    <zip>1007</zip>
    <country>CH</country>
  </address>
</contact>
```

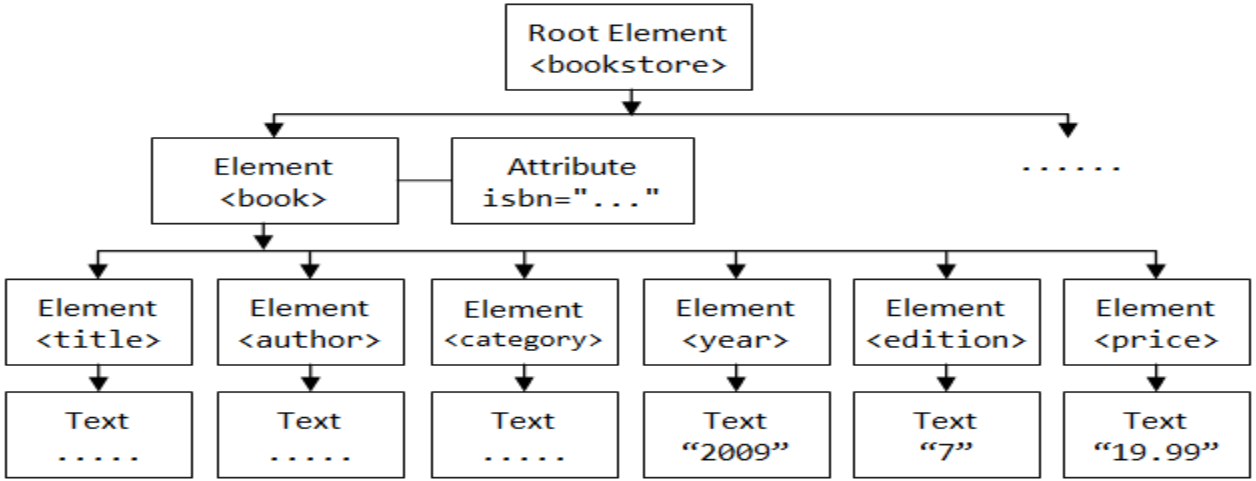
- Same in relational model

id	first name	...
656	Chuck	...
...	...	...

id	phone
656	(123) 555-0178
656	(890) 555-0133
...	...

Processing XML and JSON

- Document structure = tree
- Processing via tree traversal (depth- or breadth-first search)
- Or use proper query language, such as implemented by jq

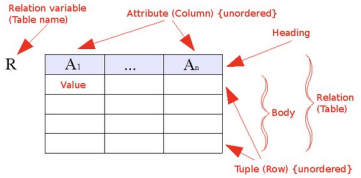


Q1: “How should I store my data on a computer?”
Q2: “How do I think about the world?”

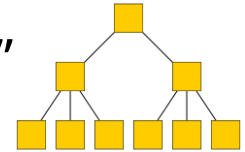
- “The world is simple: one type of entity, all with the same attributes”
  → **Flat model**

```
66.249.65.107 - - [08/Oct/2007:04:54:20 -0400] "GET /support.html
HTTP/1.1" 200 11179 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
```

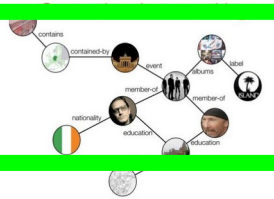
- “The world contains many types of entities, connected by relationships”
  → **Relational model**



- “The world is a hierarchy of entities”
  → **Document model**

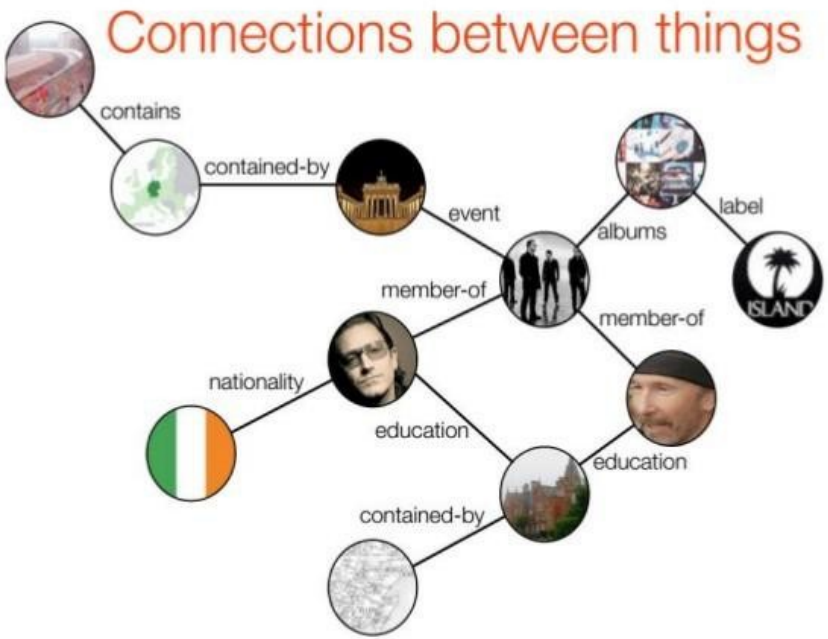


- “The world is a complex network of entities”
  → **Network model**



Network model

- “The world is a complex network of entities”



“How should I store my data on a computer?” —A word (or two) on binary formats

- Binary format often the key to performance, **avoiding expensive parsing**
- Modern binary formats support nested structures, various levels of schema enforcement, compression, etc.
- Python pickle, Java Serializable, Protocol Buffers (Google), Avro (supports schema evolution), Parquet (column-oriented), etc.

→ Consider converting to a binary format at the beginning of your processing pipeline (especially when using “big data”)

Cooking with data

Part 2:
Data sources

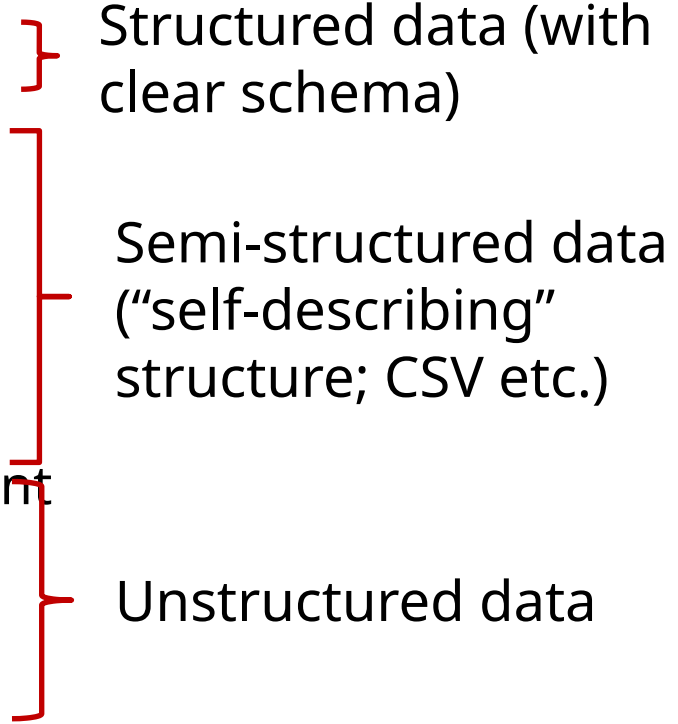
Part 1:
Data models

Part 3:
Data wrangling

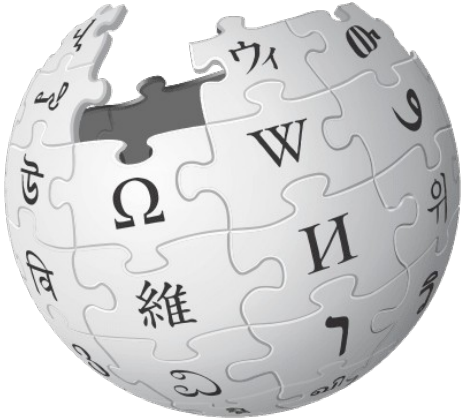
Data sources at Web companies

Examples from Facebook

- Application databases
- Web server logs
- Event logs
- API server logs
- Ad server logs
- Search server logs
- Advertisement landing page content
- Wikipedia
- Images and video



Another example: Wikipedia



- 200+ languages
- Over 50 million entities
- Mind-boggling richness of data

Not logged in Talk Contributions Create account Log in

Article Talk | Read Edit View history | Search Wikipedia

San Francisco

From Wikipedia, the free encyclopedia
(Redirected from San Francisco, California)

Coordinates: 37°47′N 122°25′W

This article is about the city and county in California. For other uses, see San Francisco (disambiguation).

San Francisco (initials **SF**[17]) (/ˌsæn frənˈsɪskoʊ/, Spanish for Saint Francis; Spanish: [san franˈsisko]), officially the **City and County of San Francisco**, is the cultural, commercial, and financial center of Northern California. The consolidated city-county covers an area of about 47.9 square miles (124 km²)[18] at the north end of the San Francisco Peninsula in the San Francisco Bay Area. It is the fourth-most populous city in California, and the 13th-most populous in the United States, with a 2016 census-estimated population of 870,887.[13] The population is projected to reach 1 million by 2033.[19]

San Francisco was founded on June 29, 1776, when colonists from Spain established Presidio of San Francisco at the Golden Gate and Mission San Francisco de Asís a few miles away, all named for St. Francis of Assisi.[1] The California Gold Rush of 1849 brought rapid growth, making it the largest city on the West Coast at the time. San Francisco became a consolidated city-county in 1856.[20] After three-quarters of the city was destroyed by the 1906 earthquake and fire,[21] San Francisco was quickly rebuilt, hosting the Panama-Pacific International Exposition nine years later. In World War II, San Francisco was a major port of embarkation for service members shipping out to the Pacific Theater.[22] It then became the birthplace of the United Nations in 1945.[23][24][25] After the war, the confluence of returning servicemen, massive immigration, liberalizing attitudes, along with the rise of the "hippie" counterculture, the Sexual Revolution, the Peace Movement growing from opposition to United States involvement in the Vietnam War, and other factors led to the Summer of Love and the gay rights movement, cementing San Francisco as a center of liberal activism in the United States. Politically, the city votes strongly along liberal Democratic Party lines.

A popular tourist destination,[26] San Francisco is known for its cool summers, fog, steep rolling hills, eclectic mix of architecture, and landmarks, including the Golden Gate Bridge, cable cars, the former Alcatraz Federal Penitentiary, Fisherman's Wharf, and its Chinatown district. San Francisco is also the headquarters of five major banking institutions and various other companies such as Levi Strauss & Co., Gap Inc., Fitbit, Salesforce.com, Dropbox, Reddit, Square, Inc., Dolby, Airbnb, Weebly, Pacific Gas and Electric Company, Yelp, Pinterest, Twitter, Uber, Lyft, Mozilla, Wikimedia Fo
home to number of educational and cultural institutions, such as the Univer
Museum, the San Francisco Museum of Modern Art, and the California Aca

San Francisco has several nicknames, including "The City by the Bay", "Gc
City", and as well as older ones like "The City that Knows How", "Baghdad
City".[17] As of 2017, San Francisco is ranked high on world liveability ranki

Contents [hide]

San Francisco, California
Consolidated city-county
City and County of San Francisco

San Francisco and the Golden Gate Bridge from Marin Headlands

Flag

Seal

Location of San Francisco in California

Coordinates: 37°47′N 122°25′W

Country	United States
State	California

link

Learning resources from Wikiversity

Places adjacent to San Francisco [show]

v · t · e **City and County of San Francisco** [show]

Articles relating to the City and County of San Francisco [show]

Authority control WorldCat Identities · VIAF: 143700861 · LCCN: n79018452 · ISNI: 0000 0004 0461 8991 · GND: 4051520-5 · SUDOC: 040776433 · NDL: 00628542

Categories: San Francisco | 1850 establishments in California | California counties | Cities in the San Francisco Bay Area | Consolidated city-counties in the United States | Counties in the San Francisco Bay Area | County seats in California | Hudson's Bay Company trading posts | Incorporated cities and towns in California | Populated coastal places in California | Populated places established in 1776 | Port cities and towns of the West Coast of the United States | Spanish mission settlements in North America

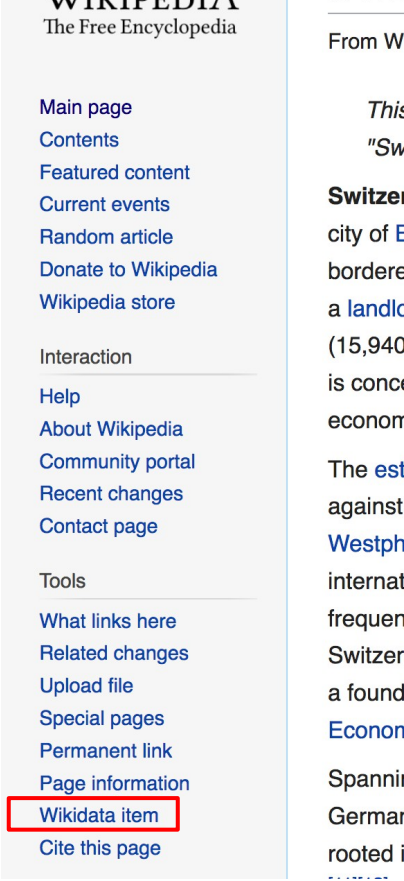
Wikipedia

How to work with Wikipedia?

- XML dumps with wiki markup, SQL database dumps
- Issues: Unicode, size, recency, etc.
- To make your life easier:
  (1) Find projects on GitHub to help you
  (2) Use more structured versions

Wikidata

- “Database version” of Wikipedia
- {fr:Suisse, de:Schweiz, it:Svizzera, en:Switzerland, …} → Q39
- Both API access and full database dumps
- Available as
  - JSON (document model)
  - RDF (network model)



English Not logged in

WIKIDATA

Item Discussion Read View history Search Wikidata

Switzerland (Q39)

federal republic in Western Europe edit

Swiss Confederation | CH | SUI | Suisse | Schweiz | Svizzera |

Main page
Community portal
Project chat
Create a new item
Recent changes
Random item
Query Service
Nearby
Help
Donate

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

In more languages Configure

Language	Label	Description	Also known as
English	Switzerland	federal republic in Western Europe	Swiss Confederation CH SUI Suisse Schweiz Svizzera
German	Schweiz	Staat in Mitteleuropa	Schweizerische Eidgenossensc… Eidgenossenschaft CH SUI
Swiss German	Schwyz	No description defined	
French	Suisse	pays d'Europe	

All entered languages

Statements

instance of	sovereign state 1 reference
	country start time 12 September 1848 *Gregorian*

Crawling and processing webpages: HTML

Plenty of bulk-downloadable HTML data:

- Common Crawl dataset, about 1.82 billion web pages -- huge!
- (… but less than 0.1% of Google's Web crawl, as of 2015)
- 145 TB, hosted on Amazon S3, also available for download

… but if you need a specific website: use a **crawler/"spider"**: Apache Nutch, Storm, Heritrix 3, Scrapy, etc. (or simply wget…)

Useful HTML tools

Requests http://docs.python-requests.org/en/master/
An elegant and simple HTTP library for Python

Scrapy https://scrapy.org/
An open-source framework to build Web crawlers

Beautiful Soup
http://www.crummy.com/software/BeautifulSoup/
A Python API for handling real HTML

Plain ol' /regular/expres*ion/s…

Schema.org: microformats for Web pages

- Nuggets of structured information embedded in (semantically) unstructured HTML

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
  Director: <span itemprop="name">James Cameron</span>
(born <time itemprop="birthDate" datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>
</div>
```

Web services

- Most large web sites today actively discourage screen-scraping to get their content
- Instead: Web service APIs, for interoperable machine-to-machine interaction over a network
- The preferred way to get data from online sources
- Most common framework: REST
  - You request a URL from the server via HTTP
  - The server responds with a text file (e.g., JSON, XML, plain text)

REST example

```
{
  "user": {
    "name": "Jane",
    "gender": "female",
    "location": {
      "href":
"http://www.example.org/
us/ny/new_york",
      "text": "New York"
    }
  }
}
```

← This resource is a description of a user named Jane

- Requested by sending GET request for the resource's URL, e.g., via curl:

  `curl http://www.example.org/users/jane/`

- If they need to modify the resource, they GET it, modify it, and PUT it back
- The href to the location resource allows savvy clients to get more information with another simple GET request
- Implication: Clients cannot be too "thin"; need to understand resource formats!

Cooking with data

Part 2:
Data sources

Part 1:
Data models

Part 3:
Data wrangling

Working with raw data sucks

Data comes in all shapes and sizes

– CSV files, PDFs, SQL dumps, .jpg, …

Different files have different formatting

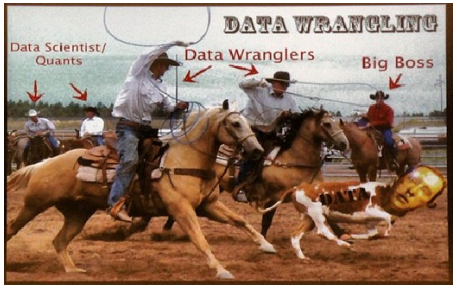
– Empty string or space instead of NULL, extra header rows, character encoding, …

“Dirty” data

– Unwanted anomalies, duplicates

Raw data without thinking:
A recipe for disaster!

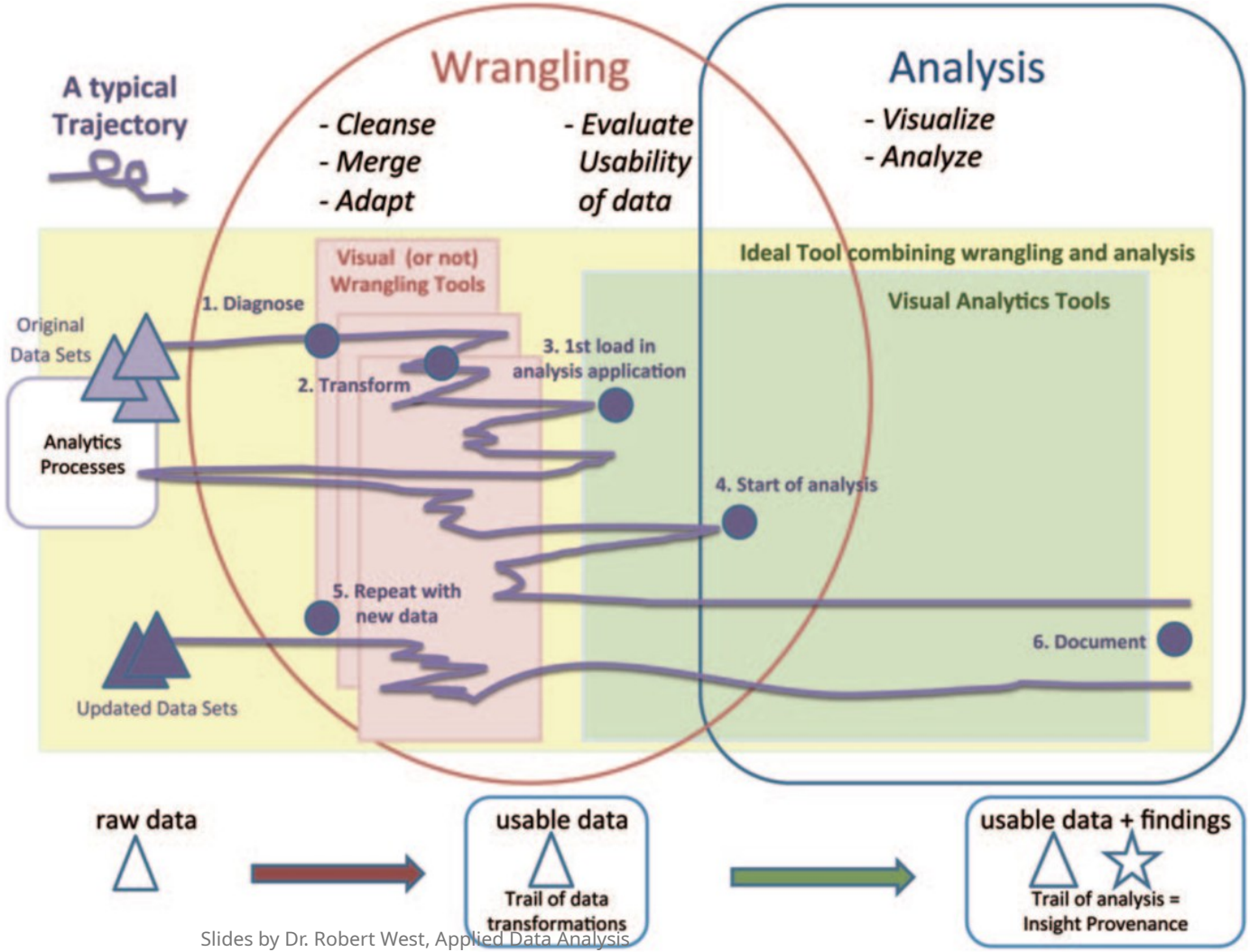
What is data wrangling?



- **Goal**: extract and standardize the raw data
  - Combine multiple data sources
  - Clean data anomalies
- **Strategy**: Combine automation with interactive visualizations to aid in cleaning
- **Outcome**: Improve efficiency and scale of data importing

Wrangling takes **between 50% and 80% of your time**…

[Source]



Types of data problems

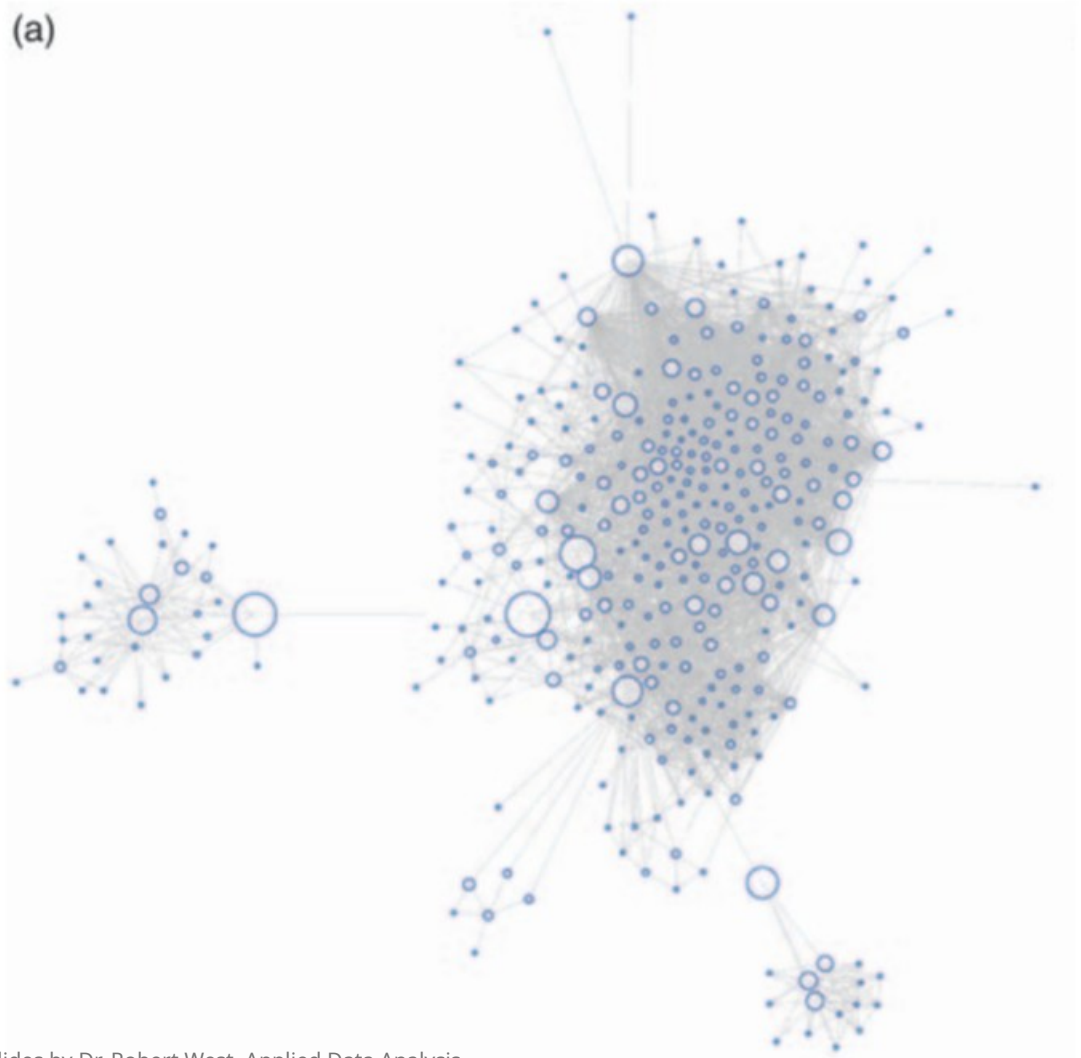
- Missing data
- Incorrect data
- Inconsistent representations of the same data
- About 75% of data problems require human intervention (e.g., experts, crowdsourcing, etc.)
- Tradeoff between cleaning data vs. over-sanitizing data

link

Diagnosing data problems

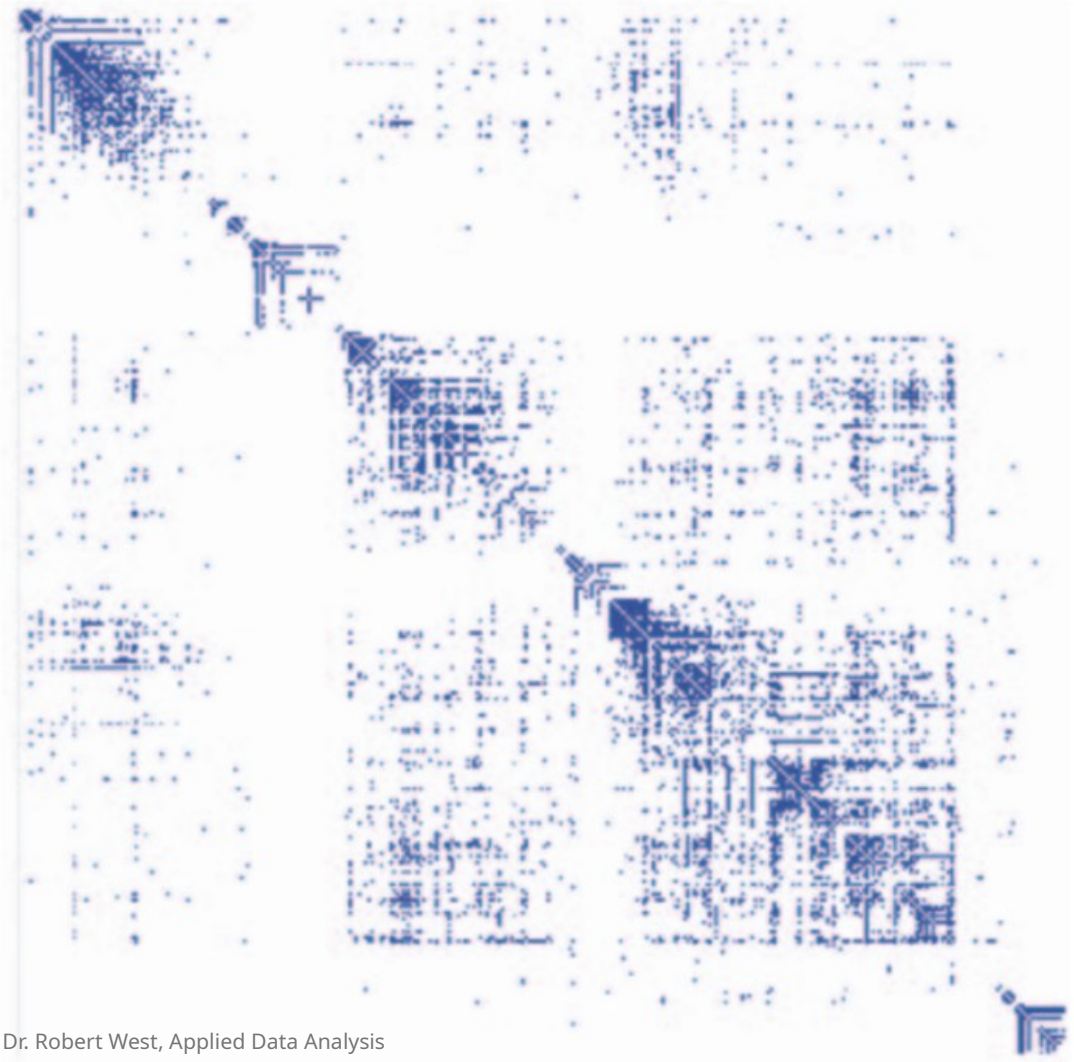
- Visualizations and basic stats can convey issues in “raw” data
- Different representations highlight different types of issues:
  - Outliers often stand out in the right kind of plot
  - Missing data will cause gaps or zero values in the right kind of plot
- Becomes increasingly difficult as data gets larger
  - Sampling to the rescue!

Facebook graph



Matrix view (1)

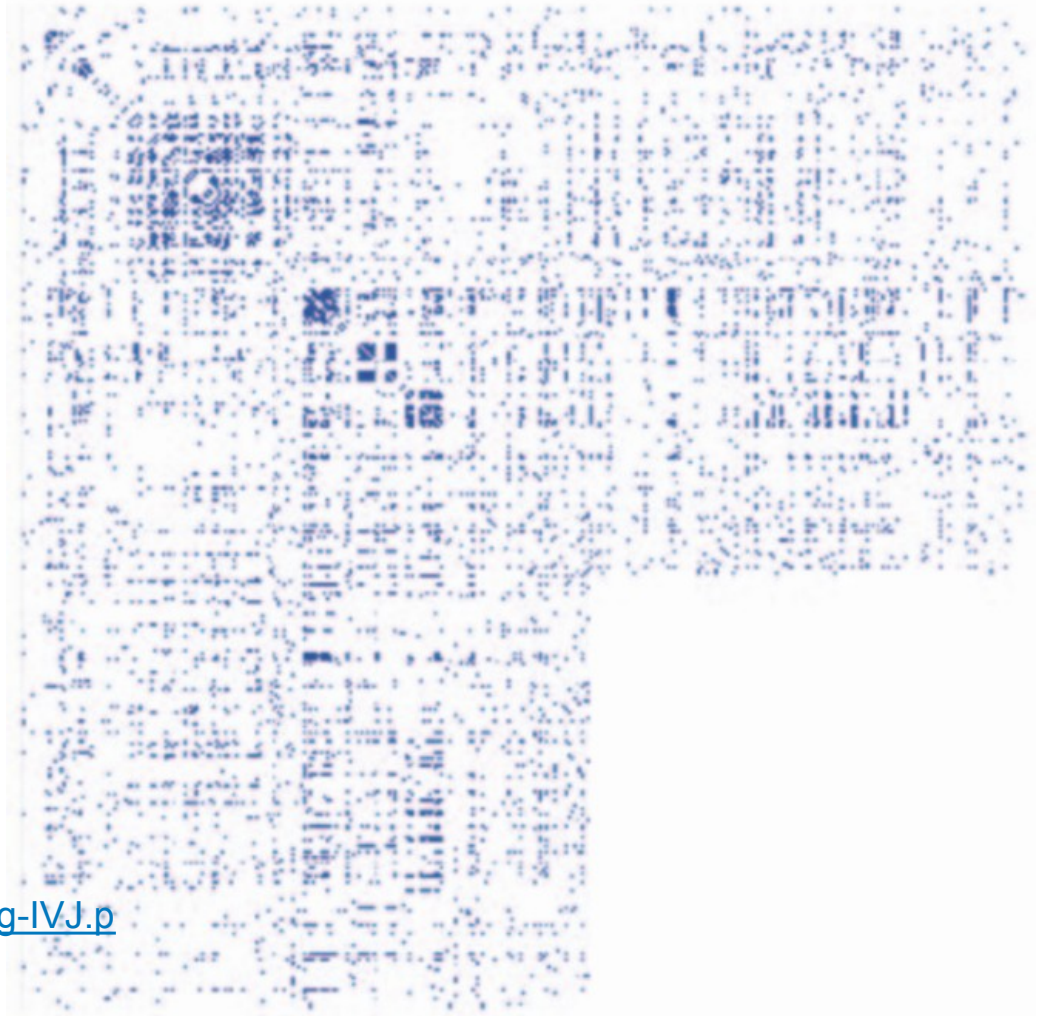
Automatic permutation of rows and columns to highlight patterns of connectivity



Matrix view (2)

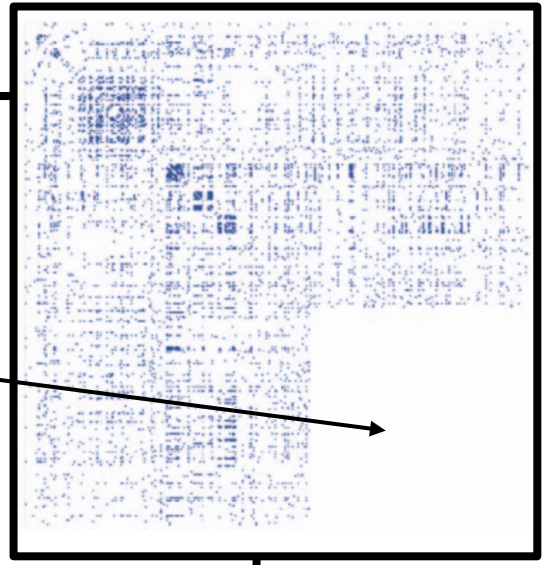
Rows and columns sorted in the order in which data was retrieved via the Facebook API

Can you guess what's going on here?

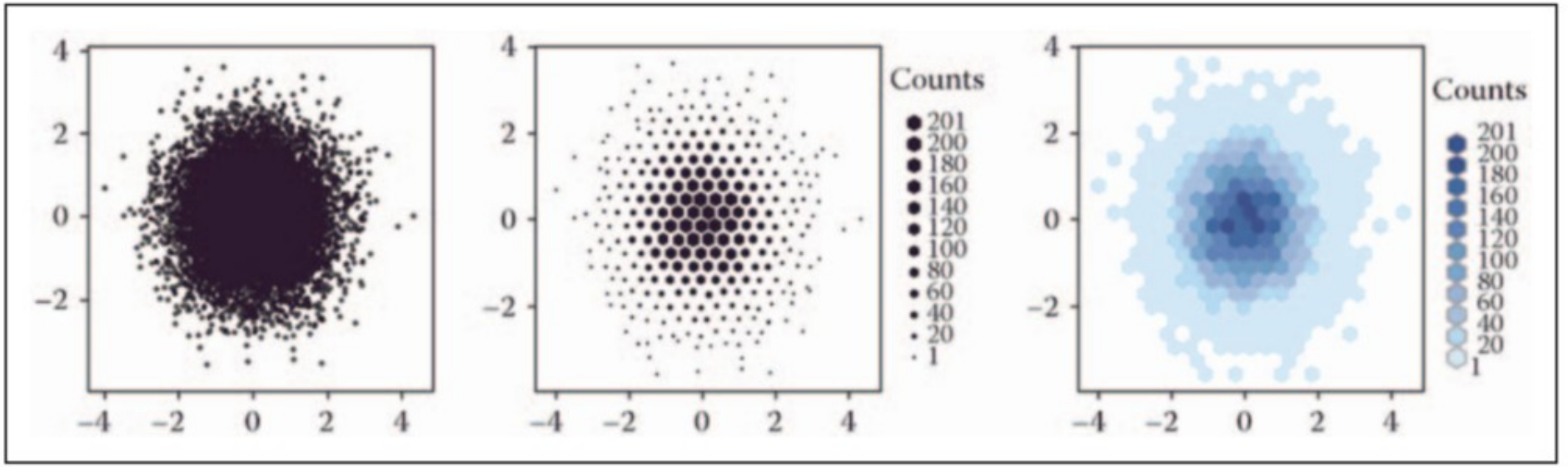


More info:
http://vis.stanford.edu/files/2011-DataWrangling-IVJ.pdf

Think for 1 minute:
What causes the white block in the adjacency matrix?

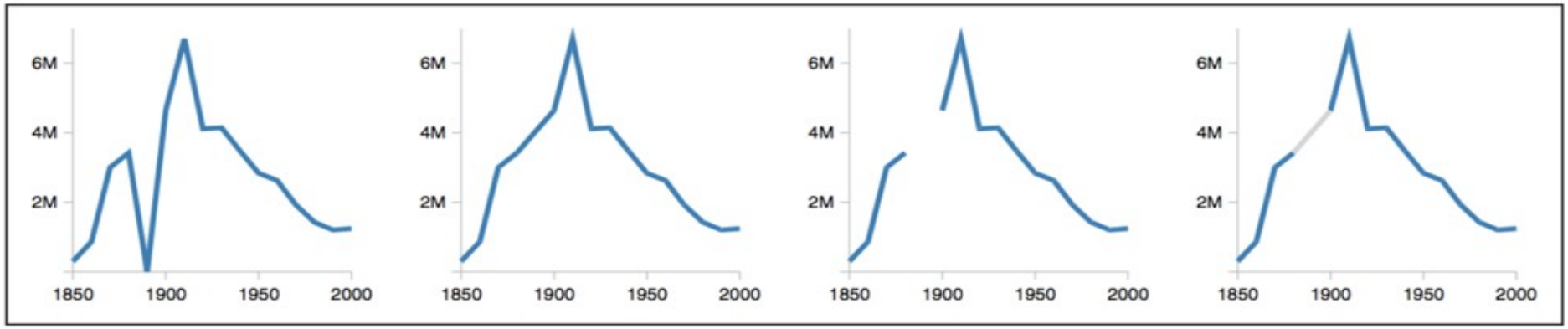


Viz at scale? Careful!



Dealing with missing data

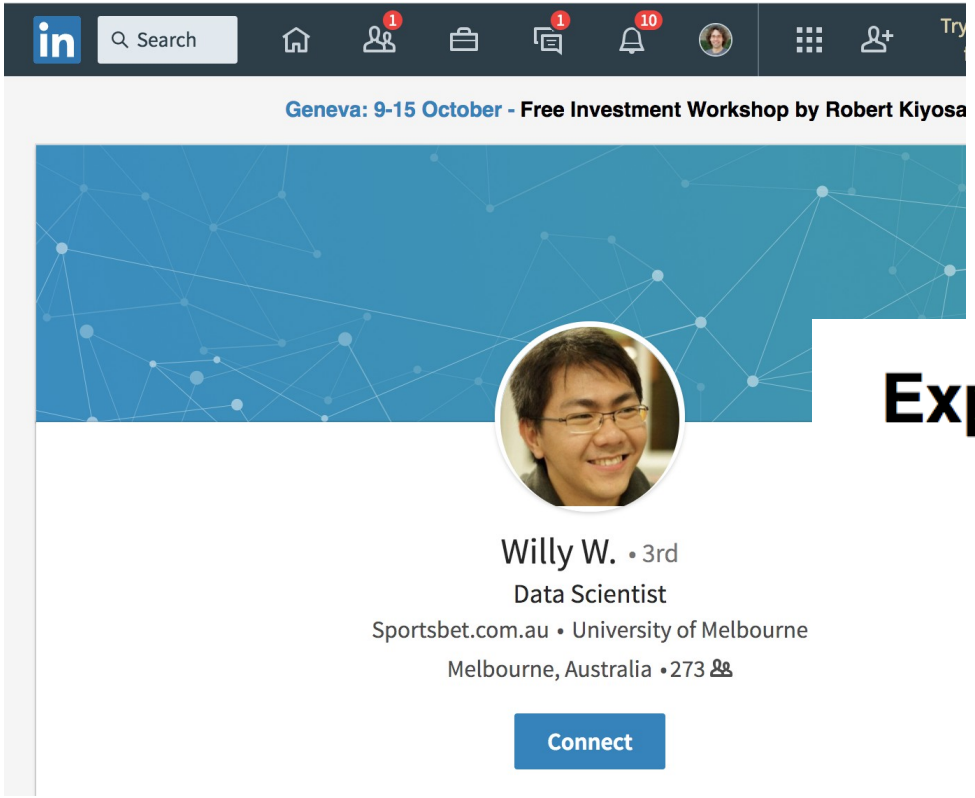
U.S. census counts of people working as “farm laborers”; values from 1890 are **missing due to records being burned in a fire**



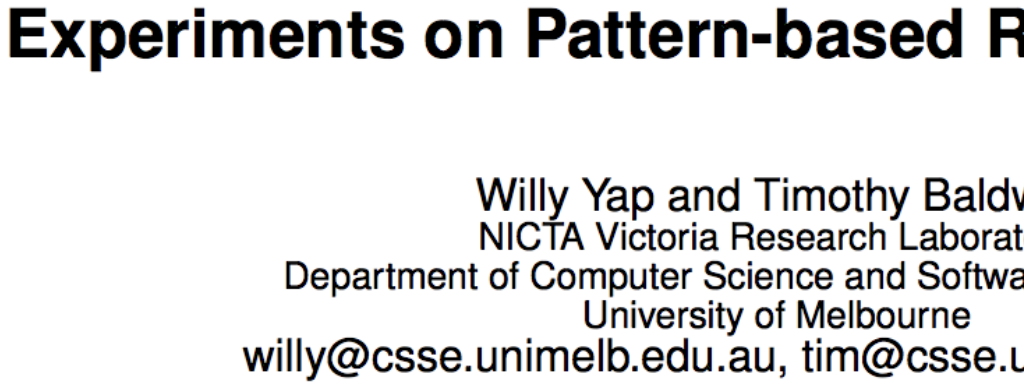
- Set values to zero?
- Interpolate based on existing data?
- Omit missing data?

Knowledge about domain and data collection should drive your choice!

Inconsistent data: “My name is Willy”



First name	Last name
Willy	NULL
…	…



Before you start analyzing your data

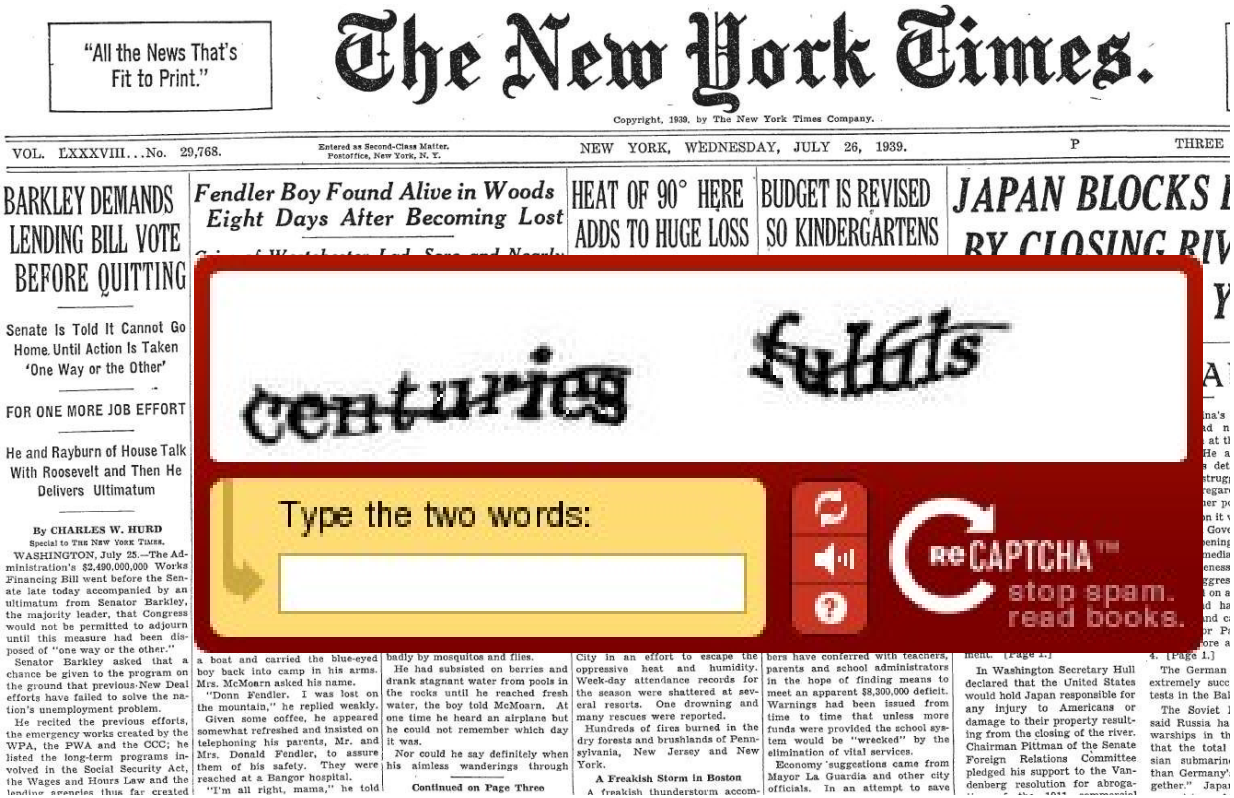
- “Do I have **missing data**?” “If data were missing, how could I know?”
- “Do I have **corrupted data**?” (May arise from measurement errors, wrong sampling strategies, etc.)
- **Parse/transform data** into appropriate format for your specific analysis
- Don’t be surprised if you need to come back to this

Desiderata – Things that are desired

It’s always ideal if you can put your hands on the **code/documentation about the dataset** you are analyzing (provenance)

It’s always ideal if the provided **data format is nicely parseable** (otherwise you need regexes, or maybe even pay humans)

Highly non-parseable data



"All the News That's Fit to Print."

The New York Times.

Copyright, 1939, by The New York Times Company.

VOL. LXXXVIII...No. 29,768. Entered as Second-Class Matter, Postoffice, New York, N. Y. NEW YORK, WEDNESDAY, JULY 26, 1939. P THREE

BARKLEY DEMANDS LENDING BILL VOTE BEFORE QUITTING

Senate Is Told It Cannot Go Home Until Action Is Taken 'One Way or the Other'

FOR ONE MORE JOB EFFORT

He and Rayburn of House Talk With Roosevelt and Then He Delivers Ultimatum

By CHARLES W. HURD

Special to The New York Times.

WASHINGTON, July 25.—The Administration's $2,490,000,000 Works Financing Bill went before the Senate late today accompanied by an ultimatum from Senator Barkley, the majority leader, that Congress would not be permitted to adjourn until this measure had been disposed of "one way or the other."

Senator Barkley asked that a chance be given to the program on the ground that previous New Deal efforts have failed to solve the nation's unemployment problem.

He recited the previous efforts, the emergency works created by the WPA, the PWA and the CCC; he listed the long-term programs involved in the Social Security Act, the Wages and Hours Law and the lending agencies thus far created

Fendler Boy Found Alive in Woods Eight Days After Becoming Lost

a boat and carried the blue-eyed boy back into camp in his arms. Mrs. McMoarn asked his name.

"Donn Fendler. I was lost on the mountain," he replied weakly.

Given some coffee, he appeared somewhat refreshed and insisted on telephoning his parents, Mr. and Mrs. Donald Fendler, to assure them of his safety. They were reached at a Bangor hospital.

"I'm all right, mama," he told

badly by mosquitos and flies.

He had subsisted on berries and drank stagnant water from pools in the rocks until he reached fresh water, the boy told McMoarn. At one time he heard an airplane but he could not remember which day it was.

Nor could he say definitely when his aimless wanderings through

Continued on Page Three

HEAT OF 90° HERE ADDS TO HUGE LOSS

City in an effort to escape the oppressive heat and humidity. Week-day attendance records for the season were shattered at several resorts. One drowning and many rescues were reported.

Hundreds of fires burned in the dry forests and brushlands of Pennsylvania, New Jersey and New York.

A Freakish Storm in Boston

A freakish thunderstorm accom-

BUDGET IS REVISED SO KINDERGARTENS

bers have conferred with teachers, parents and school administrators in the hope of finding means to meet an apparent $8,300,000 deficit. Warnings had been issued from time to time that unless more funds were provided the school system would be "wrecked" by the elimination of vital services.

Economy suggestions came from Mayor La Guardia and other city officials. In an attempt to save

JAPAN BLOCKS I BY CLOSING RIV

ment. [Page 1.]

In Washington Secretary Hull declared that the United States would hold Japan responsible for any injury to Americans or damage to their property resulting from the closing of the river. Chairman Pittman of the Senate Foreign Relations Committee pledged his support to the Vandemberg resolution for abroga-

4. [Page 1.]

The German extremely succ tests in the Bal

The Soviet said Russia ha warships in th that the total sian submarin than Germany' gether." Japa

centuries

exhibits

Type the two words:

reCAPTCHA™ stop spam. read books.

Entire NY Times archive (since 1851) digitized as of 2015

Q: What do

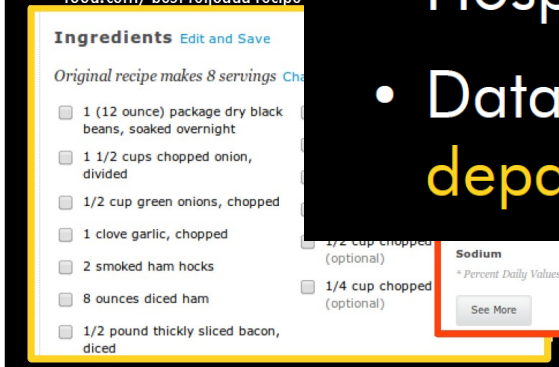
Our method: I

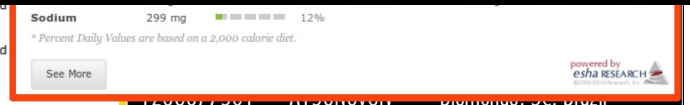
Consenting IE users, 18 mo

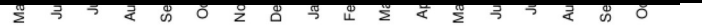
URL	Referrer	Tin
yahoo.com?q=the+onion	yahoo.com	128
theonion.com	yahoo.com?q=the+onion	128
theonion.com/Area-Man-Sad		
bing.com		
bing.com?q=feijoada+recipe		
allrecipes.com/tasty-feijoada		
food.com/best-feijoada-recipe		

- Find Amazon add-to-cart events heuristically in logs:
  - Referrer: http://www.amazon.com/Forks-Over-Knives-Plant-Based-Health/dp/1615190457
  - URL: http://www.amazon.com/gp/cart/view-upsell.html?...
- Get product info for product id using Amazon API
- Consider all add-to-cart events for category "Diets & Weight Loss"

- Collaborated with clinician at Washington Hospital Center, Washington, D.C.
- Data: All CHF admissions to emergency department for time period of our browsing logs

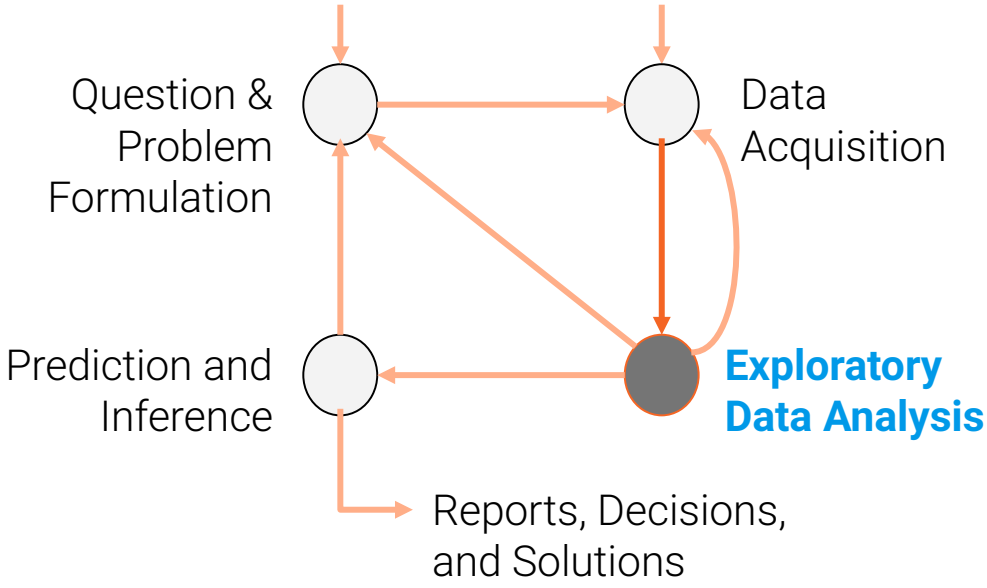


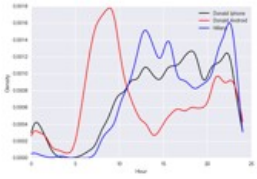




Slides by Dr. Robert West, Applied Data Analysis 12

[Paper with results and plots]





Exploring and Cleaning Tabular Data

Data Science in Practice

EDA, Data Cleaning, Text processing (regular expressions)

Tuberculosis – United States, 2021

Summary

What is already known about this topic?

The number of reported U.S. tuberculosis (TB) cases decreased sharply in 2020, possibly related to multiple factors associated with the COVID-19 pandemic.

What is added by this report?

Reported TB incidence (cases per 100,000 persons) increased 9.4%, from 2.2 during 2020 to 2.4 during 2021 but was lower than incidence during 2019 (2.7). Increases occurred among both U.S.-born and non–U.S.-born persons.

What are the implications for public health practice?

Factors contributing to changes in reported TB during 2020–2021 likely include an actual reduction in TB incidence as well as delayed or missed TB diagnoses. Timely evaluation and treatment of TB and latent tuberculosis infection remain critical to achieving U.S. TB elimination.

CDC Morbidity and Mortality Weekly Report (MMWR) 03/25/2022.

What is **incidence**? Why use it here?

How was "9.4% increase" computed?

Question: Can we **reproduce** these numbers using government data?

CDC: **source**, 2021

Defining incidence

From the CDC report: **TB incidence** is computed as the number of "cases per 100,000 persons using mid-year population estimates from the U.S. Census Bureau."

- Incidence is useful when comparing case rates across differently sized populations.

$$\textbf{TB incidence} = \frac{\#\text{ TB cases in population}}{\#\text{ groups in population}}$$

(group: 100,000 people)

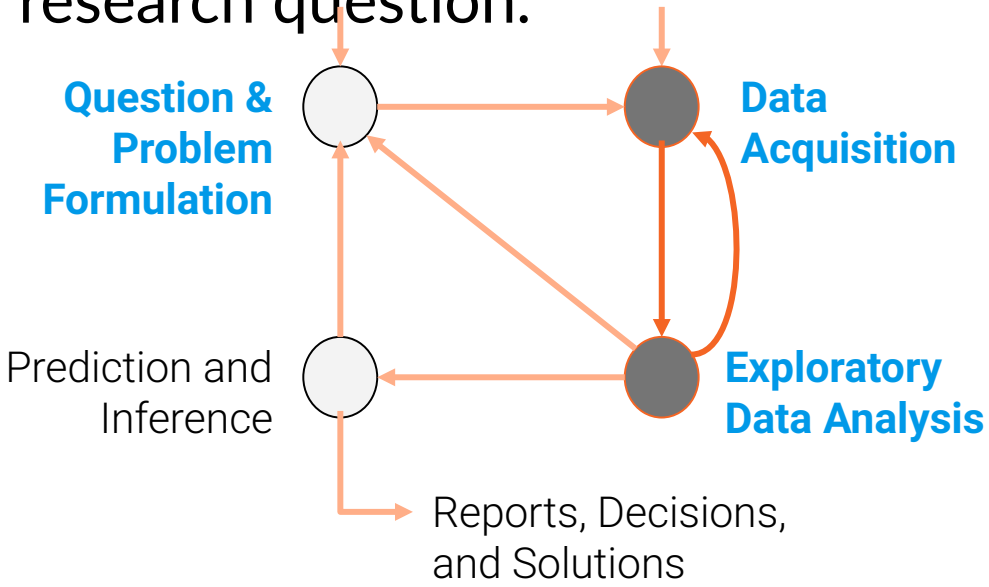
$$= \frac{\#\text{ TB cases}}{(\text{population}/100{,}000)}$$

$$= \frac{\#\text{ TB cases}}{\text{population}} \times 100{,}000$$

We don't have U.S. Census population data in our DataFrame.
We need to acquire it to verify incidence!

The Data Science Lifecycle is a Cycle

In practice, EDA informs whether you need more data to address your research question.



Key Data Properties to Consider in EDA

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

Tabular Data

- **EDA, Part I**
  - **Structure: Tabular Data**
  - Granularity
  - Structure: Variable Types
  - Multiple Files

Key Data Properties to Consider in EDA

File Format

Variable Type

Multiple files
(Primary and Foreign Keys)

Structure -- the “shape” of a data file

Granularity -- how fine/coarse is each datum

Scope -- how (in)complete is the data

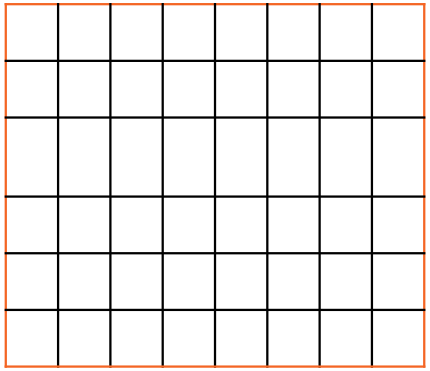
Temporality -- how is the data situated in time

Faithfulness -- how well does the data capture “reality”

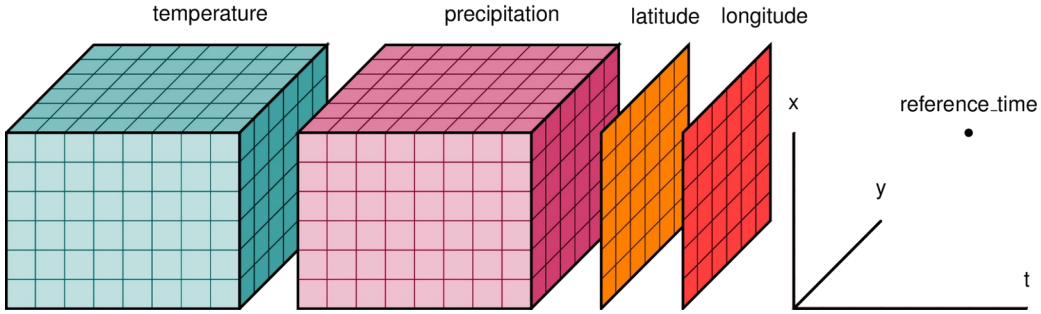
Rectangular and Non-rectangular Data

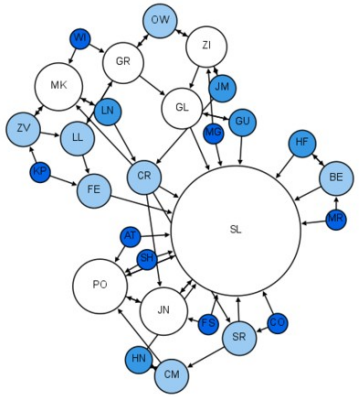
Data come in many different shapes.

Rectangular data



Non-rectangular data

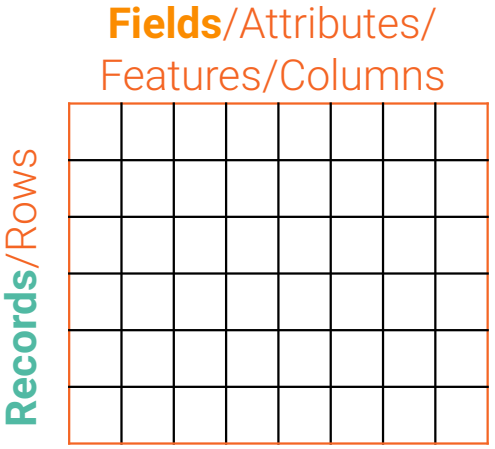




Rectangular Data

We prefer rectangular data for data analysis (why?)

- Regular structures are easy to manipulate and analyze
- A big part of data cleaning is about transforming data to be more rectangular



Two kinds of rectangular data: **Tables** and **Matrices.**

Tables (a.k.a. dataframes in R/Python and relations in SQL)

- Named columns with different types
- Manipulated using data transformation languages (map, filter, group by, join, …)

Matrices

- Numeric data of the same type (float, int, etc.)
- Manipulated using linear algebra

Demo Slides

CSV: Comma-Separated Values

Tuberculosis in the US [CDC **source**].

CSV is a very common **tabular file format**.

- **Records** (rows) are delimited by a newline: '\n', "\r\n"
- **Fields** (columns) are delimited by commas: ','

Pandas: **pd.read_csv**(header=...)

Fields/Attributes/Features/Columns

Records/Rows

	U.S. jurisdiction	TB cases 2019	...
0	Total	8,900	...
1	Alabama	87	...

Granularity

- **EDA, Part I**
  - Structure: Tabular Data
  - **Granularity**
  - Structure: Variable Types
  - Multiple Files

Granularity: ~~How fine/coarse is each datum?~~

What does each **record** represent?

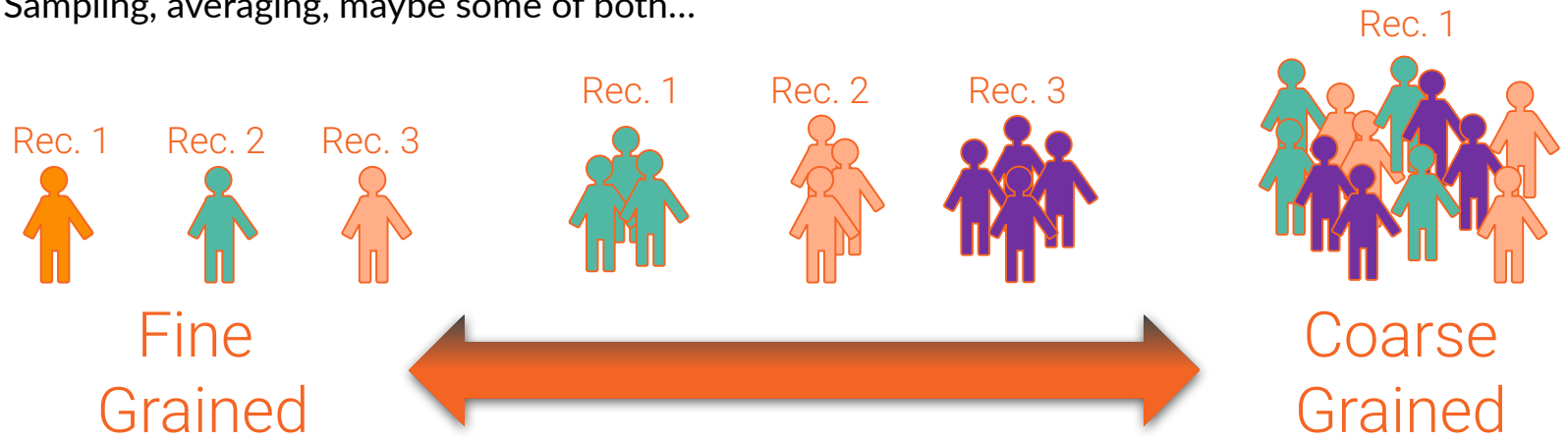
- Examples: a purchase, a person, a group of users

Do all records capture granularity at the same level?

- Some data will include summaries (aka **rollups**) as records.

If the data are **coarse**, how were the records aggregated?

- Sampling, averaging, maybe some of both…



Variable Types

- **EDA, Part I**
  - Structure: Tabular Data
  - Granularity
  - **Structure: Variable Types**
  - Multiple Files

Variables are Columns

Let's look at records with the same granularity.

What does each **column** represent?

A **variable** is a **measurement** of a particular concept.

	U.S. jurisdiction	TB cases 2019	...
1	Alabama	87	...
2	Alaska	58	...
...	...	...	...

The U.S. Jurisdiction **variable**

It has two common properties:

- **Datatype/Storage type**:
  How each variable value is stored in memory. df[colname].dtype
  - integer, floating point, boolean, object (string-like), etc.

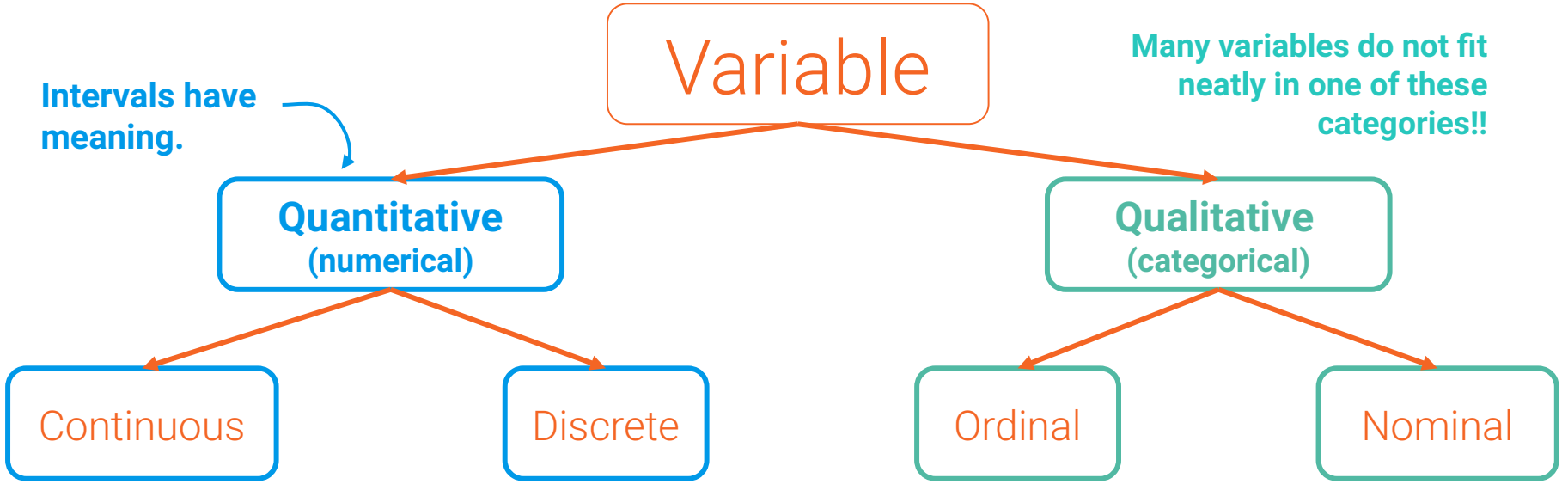
  Affects which pandas functions you use.

- **Variable type/Feature type**:
  Conceptualized measurement of information (and therefore what values it can take on).
  - Use expert knowledge
  - Explore data itself
  - Consult data codebook (if it exists).

  Affects how you visualize and interpret the data.

⚠️In this class, "variable types" are conceptual!!

Variable Feature Types

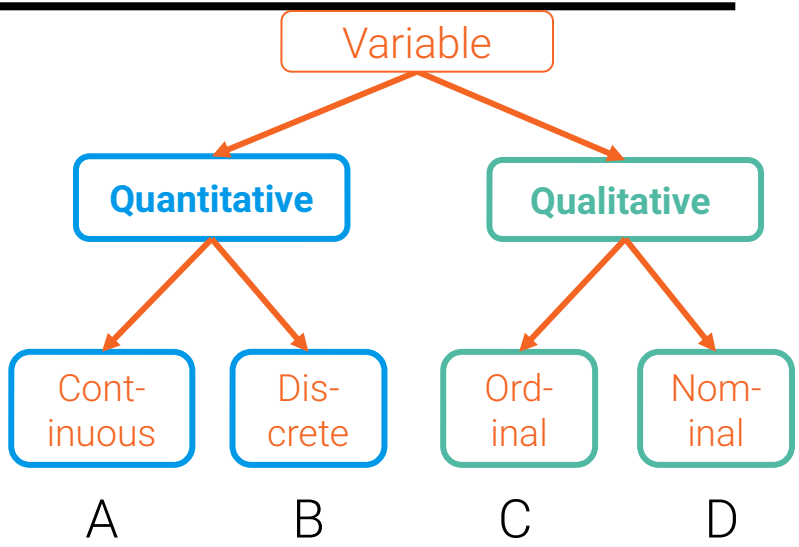


Note that **qualitative variables** could have numeric levels; conversely, **quantitative variables** could be stored as strings!

Variable Types

What is the feature type (i.e., variable type) of each variable?

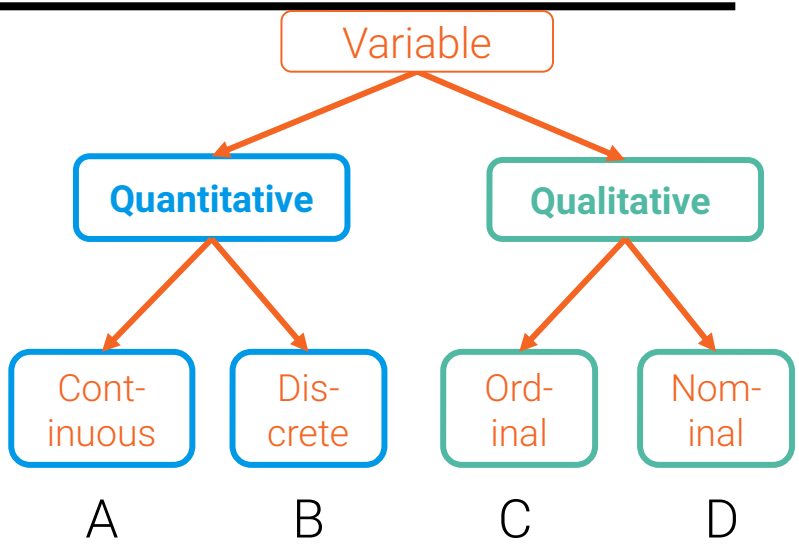
Q	Variable	Feature Type
1	CO_2 level (ppm)	
2	TB cases in 2021	
3	GPA	
4	Income bracket (low, med, high)	
5	Race/Ethnicity	
6	Number of years of education	
7	Yelp Rating	



Variable Types

What is the feature type of each variable?

Q	Variable	Feature Type
1	CO_2 level (ppm)	**A. Quantitative Cont.**
2	Number of siblings	**B. Quantitative Discrete**
3	GPA	**A. Quantitative Cont.**
4	Income bracket (low, med, high)	**C. Qualitative Ordinal**
5	Race/Ethnicity	**D. Qualitative Nominal**
6	Number of years of education	**B. Quantitative Discrete**
7	Yelp Rating	**C. Qualitative Ordinal**



Many of these examples show how "shaggy" these categories are!!
We will revisit variable types when we learn how to visualize variables.

Multiple Files

Structure

- **Multiple Files**
- More File Formats

Scope and Temporality

Faithfulness (and Missing Values)

Structure: Primary Keys and Foreign Keys

Sometimes your data comes in multiple files:

- Often data will reference other pieces of data.
- Alternatively, you will collect multiple pieces of related data.

Use pd.merge to **join** data on **keys**.

Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

Orders.csv

OrderNum	CustID	Date
1	171345	8/21/2017
2	281139	8/30/2017

Products.csv

ProdID	Cost
42	3.14
999	2.72

Purchases.csv

OrderNum	ProdID	Quantity
1	42	3
1	999	2
2	42	1

Structure: Primary Keys and Foreign Keys

Sometimes your data comes in multiple files:

- Often data will reference other pieces of data.
- Alternatively, you will collect multiple pieces of related data.

Use pd.merge to join data on **keys**.

Primary key: the column or set of columns in a table that *uniquely* determine the values in the remaining columns

- Primary keys are unique, but could be tuples.
- Examples: SSN, ProductIDs, …

Primary Key

Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

Primary Key

Orders.csv

OrderNum	CustID	Date
1	171345	8/21/2017
2	281139	8/30/2017

Products.csv

ProdID	Cost
42	3.14
999	2.72

Primary Key

Purchases.csv

OrderNum	ProdID	Quantity
1	42	3
1	999	2
2	42	1

Structure: Primary Keys and Foreign Keys

Sometimes your data comes in multiple files:

- Often data will reference other pieces of data.
- Alternatively, you will collect multiple pieces of related data.

Use pd.merge to join data on **keys**.

Primary key: the column or set of columns in a table that determine the values of the remaining columns

- Primary keys are unique, but could be tuples.
- Examples: SSN, ProductIDs, …

Foreign keys: the column or sets of columns that reference primary keys in other tables.

Primary Key

Customers.csv

CustID	Addr
171345	Harmon..
281139	Main ..

Foreign Key

Orders.csv

OrderNum	CustID	Date
1	171345	8/21/2017
2	281139	8/30/2017

Products.csv

ProdID	Cost
42	3.14
999	2.72

Purchases.csv

OrderNum	ProdID	Quantity
1	42	3
1	999	2
2	42	1

More File Formats

Structure

- Multiple Files
- **More File Formats**

Scope and Temporality

Faithfulness (and Missing Values)

Are the data in a standard format or encoding?

- Tabular data: CSV, TSV, Excel, SQL
- Nested data: JSON or XML

Are the data organized in **records** or nested?

- Can we define records by parsing the data?
- Can we reasonably un-nest the data?

Does the data reference other data?

- Can we join/merge the data?
- Do we need to?

What are the **fields** in each record?

- How are they encoded? (e.g., strings, numbers, binary, dates ...)
- What is the type of the data?

Structure -- the "shape" of a data file

Granularity -- how fine/coarse is each datum

Summary

You will do the most data wrangling when analyzing the structure of your data.

Faithfulness -- how well does the data capture "reality"

Scope and Temporality

Structure

- Multiple Files
- More File Formats

Scope and Temporality

Faithfulness (and Missing Values)

Scope

Will my data be enough to answer my question?

- **Example**: I am interested in studying crime in California but I only have Berkeley crime data.
- **Solution**: collect more data/change research question

Is my data too expansive?

- **Example**: I am interested in student grades for DSC510 but have student grades for all Data Science classes.
- **Solution:** **Filtering** ⇒ Implications on sample?
  - If the data is a sample I may have poor coverage after filtering (More on this next week)

Does my data cover the right time frame?

- Which brings us to **Temporality**

"Scope" questions are defined by your question/problem and inform if you need better-scoped data.

Temporality

Data changes – when was the data collected/last updated?

Periodicity — Is there periodicity? Diurnal (24-hr) patterns?

What is the meaning of the time and date fields? A few options:

- When the "event" happened?
- When the data was collected or was entered into the system?
- Date the data was copied into a database? (look for many matching timestamps)

Time depends on where! (**time zones** & daylight savings)

- Learn to use **datetime** Python library and Pandas **dt** accessors
- Regions have different datestring representations: 07/08/09?

Are there strange null values?

- E.g., **January 1st 1970**, January 1st 1900…?

Temporality: ~~Unix / POSIX Time~~

Time measured in seconds since **January 1st 1970 UTC**

- Minus leap seconds ...

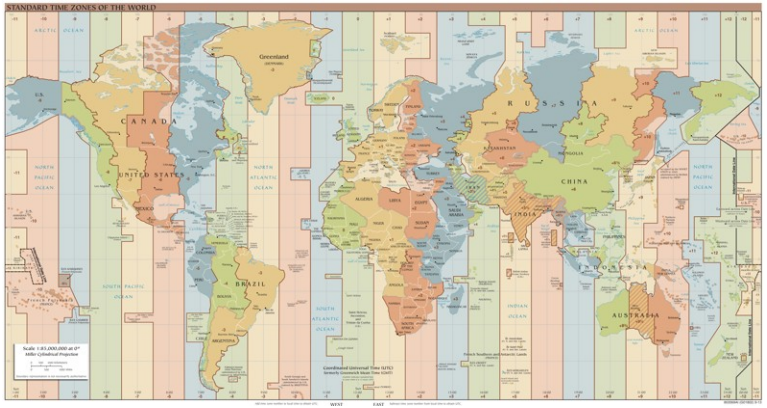
UTC is Coordinated Universal Time

- International time standard
- Measured at 0 degrees latitude
  - Similar to Greenwich Mean Time (GMT)
- No daylight savings

Time Zones:

- San Francisco (**UTC-7**) with daylight savings

Jun 27, 2023 5:00pm PDT
1687910400



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

Faithfulness (and Missing Values)

Structure

- Multiple Files
- More File Formats

Scope and Temporality

Faithfulness (and Missing Values)

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
- Misspellings of names
- Large outliers

Does my data violate **obvious dependencies**?

- E.g., age and birthday don't match

Was the data **entered by hand**?

- Spelling errors, fields shifted …
- Did the form require all fields or provide default values?

Are there obvious signs of **data falsification**?

- Repeated names, fake looking email addresses, repeated use of uncommon names or fields.

Signs that your data may not be faithful (and proposed solutions)

Truncated data
Early Microsoft Excel limits: 65536 Rows, 255 Columns

Duplicated Records or Fields
Identify and eliminate (use primary key).

Spelling Errors
Apply corrections or drop records not in a dictionary

Units not specified or consistent
Infer units, check values are in reasonable ranges for data

Time Zone Inconsistencies
Convert to a common timezone (e.g., UTC)

- Be aware of consequences in analysis when using data with inconsistencies.
- Understand the potential implications for how data were collected.

Missing Data???

Examples

" "	1970, 1900
0, -1	NaN
999, 12345	Null

NaN: "Not a Number"

Missing Data/Default Values: Solutions

A. Drop records with missing values

- Probably most common
- **Caution:** check for biases induced by dropped values
  - Missing or corrupt records might be related to something of interest

B. Keep as NaN

C. Imputation/Interpolation: Inferring missing values

- **Average Imputation:** replace with an average value
  - Which average? Often use closest related subgroup mean.
- **Hot deck imputation:** replace with a random value
- **Regression imputation:** replace with a predicted value, using some model
- **Multiple imputation:** replace with multiple random values.

Missing Data/Default Values: Solutions

A. Drop records with missing values

- Probably most common
- **Caution:** check for biases induced by dropped values
  - Missing or corrupt records might be related to something of interest

B. Keep as NaN

C. Imputation/Interpolation: Inferring missing values

- **Average Imputation:** replace with an average value
  - Which average? Often use closest related subgroup mean.
- **Hot deck imputation:** replace with a random value
- **Regression imputation:** replace with a predicted value, using some model
- **Multiple imputation:** replace with multiple random values.

} (beyond this course)

Choice affects bias and uncertainty quantification (large statistics literature)

Essential question: why are the records missing?

Summary: How do you do EDA/Data Wrangling?

Examine **data and metadata**:

- What is the date, size, organization, and structure of the data?

Examine each **field/attribute/dimension** individually

Examine **pairs of related dimensions**

- Stratifying earlier analysis: break down grades by major …

Along the way:

- **Visualize**/summarize the data
- **Validate assumptions** about data and collection process. Pay particular attention to when data were collected.
- Identify and **address anomalies**
- Apply data transformations and corrections
- **Record everything you do**! (why?)
  - Developing in Jupyter Notebooks promotes reproducibility of your own work.