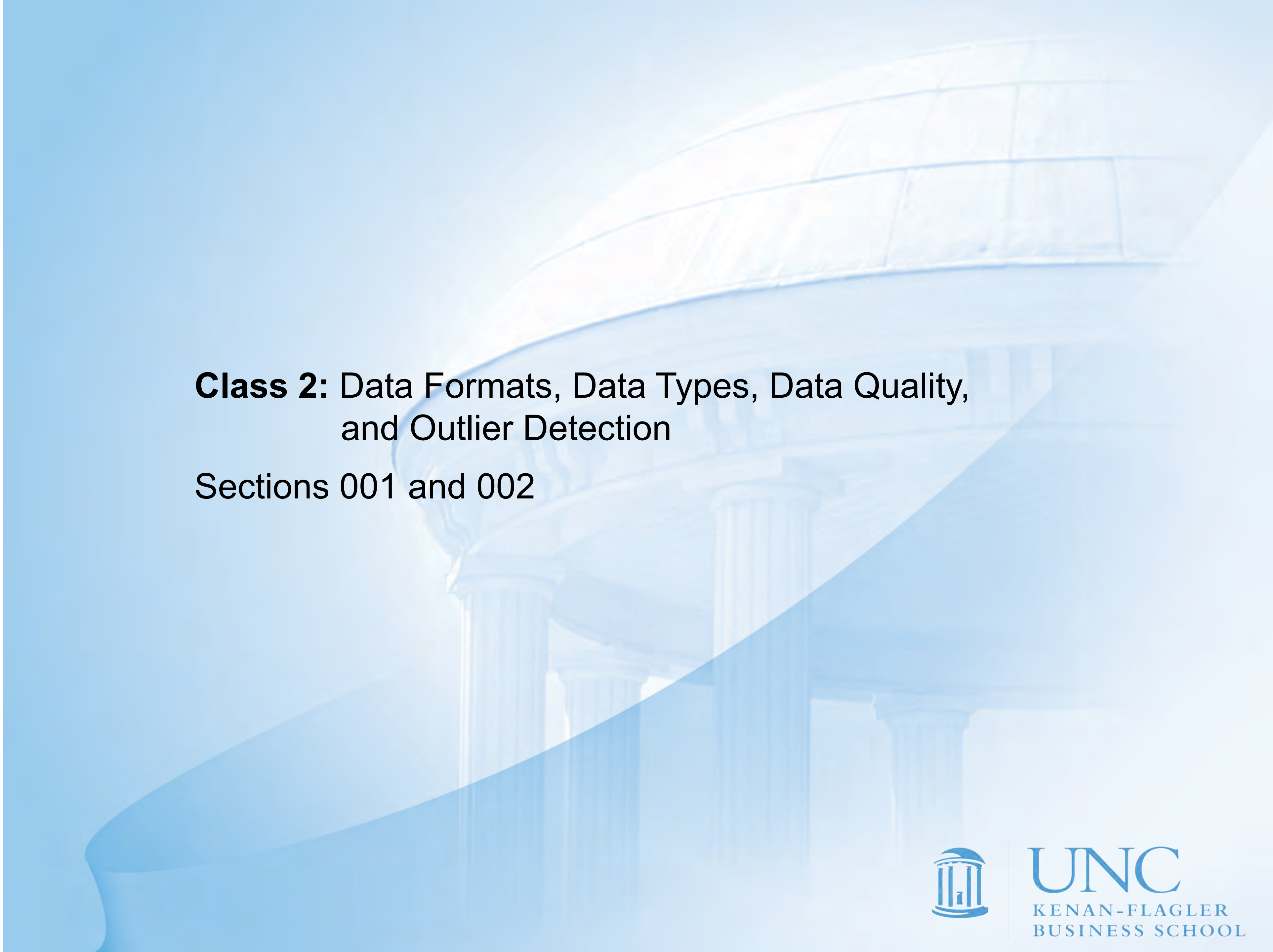
Data Science in the Business World

BUSI488 & COMP488

Daniel M. Ringel

UNC Kenan-Flagler Business School
Spring 2022

January 12th, 2023



Class 2: Data Formats, Data Types, Data Quality, and Outlier Detection

Sections 001 and 002

Today's Agenda

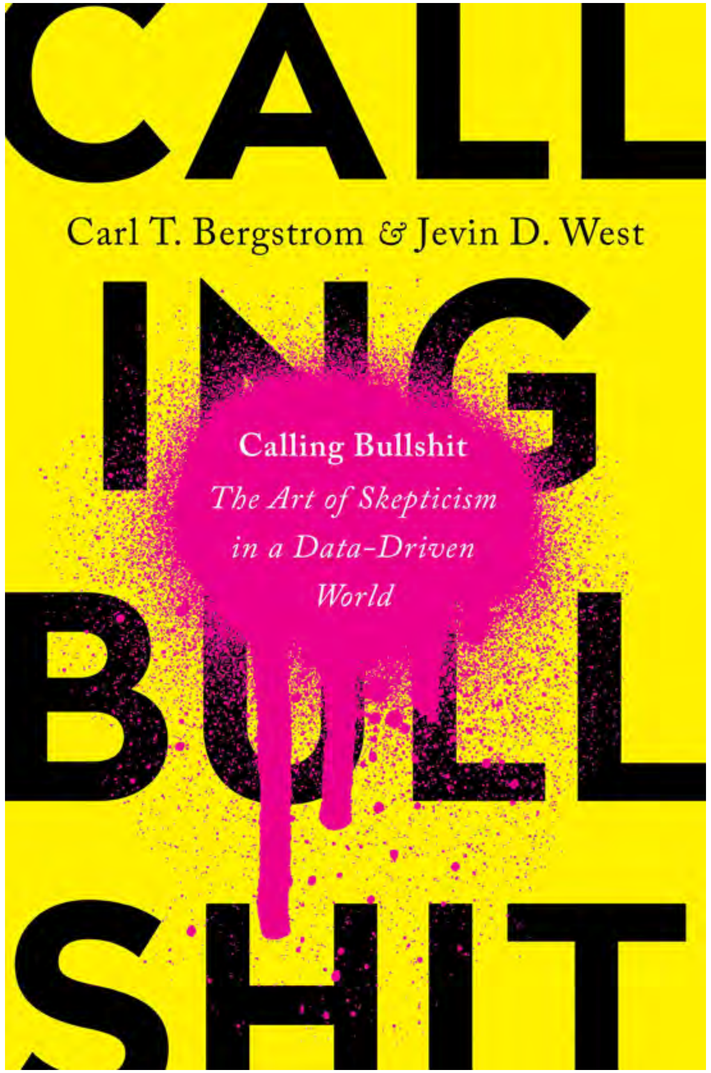
1. What are Data?
2. Data Sources of Structured vs. Unstructured Data
3. Data Storage
4. Data Types and Structure
5. Data Quality
6. Cleaning Data
7. Anomalies
8. Working with Data in Cloud Computing (Google CoLab)
9. Learning to Code in Python in this Course (DataCamp)

Prep-Check:

✓ Read Syllabus
✓ Set-up CoLab
✓ Started DataCamp HW1

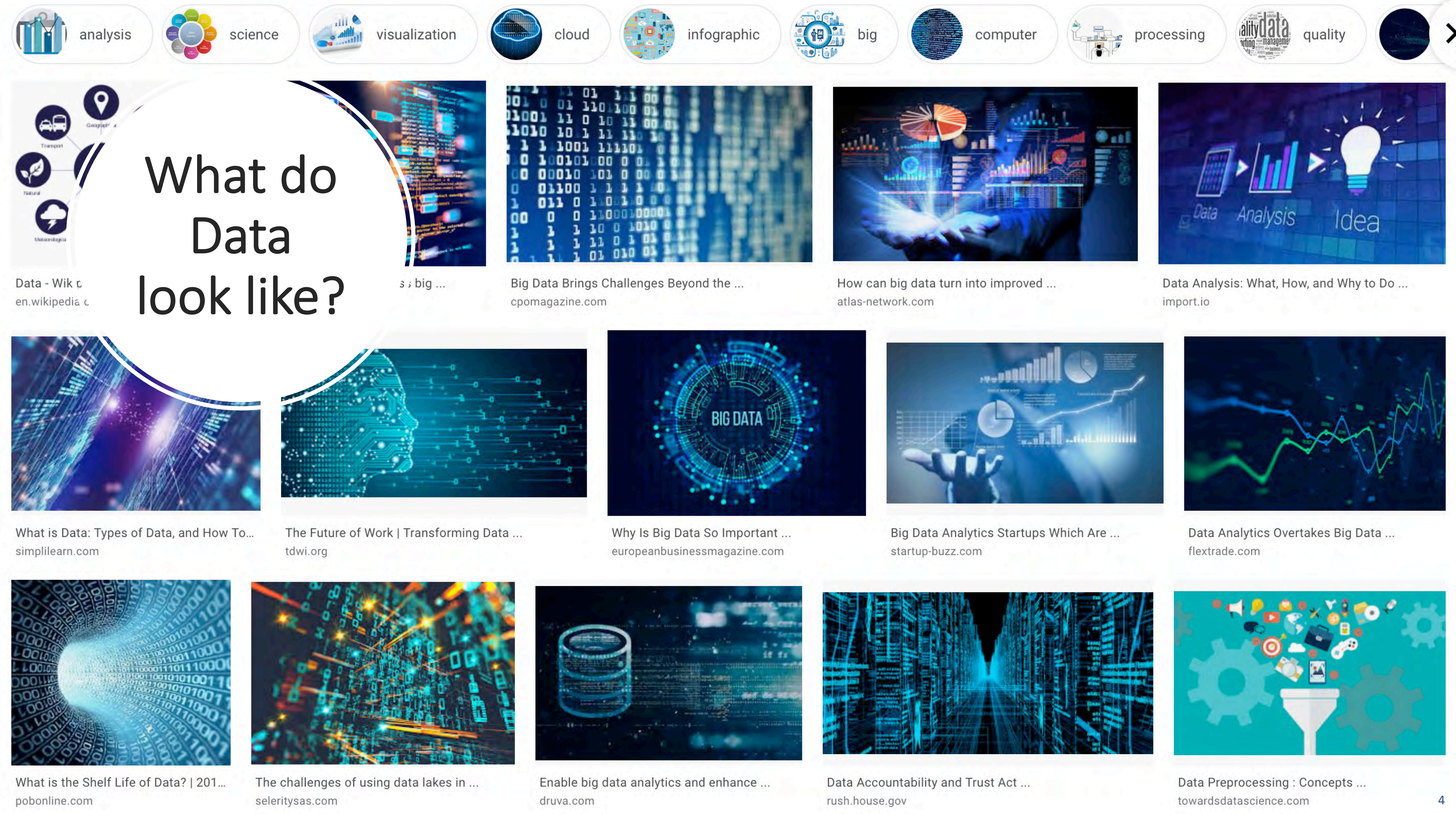
What are Data?

- Data (singular datum) are individual **units of information**
- A datum describes a **single quality or quantity** of some object or phenomenon
- In analytical processes, data are **represented by variables**
- Data are measured, collected, reported, and analyzed
- Data is **not equivalent** to **insight** or **knowledge**
- Data is the **least abstract concept**; Knowledge the most abstract


CALLING BULLSHIT
Carl T. Bergstrom & Jevin D. West
Calling Bullshit
The Art of Skepticism in a Data-Driven World

Sources:
Shannon, C.E. (1948), A mathematical theory of communication. *Bell system technical journal*, *27*(3), pp.379-423.
Wikipedia (2020), https://en.wikipedia.org/wiki/Data, accessed January 15th 2020

What do Data look like?



analysis science visualization cloud infographic big computer processing quality

Data - Wik[illegible]
en.wikipedia.[illegible]

[illegible] big ...

Big Data Brings Challenges Beyond the ...
cpomagazine.com

How can big data turn into improved ...
atlas-network.com

Data Analysis: What, How, and Why to Do ...
import.io

What is Data: Types of Data, and How To...
simplilearn.com

The Future of Work | Transforming Data ...
tdwi.org

Why Is Big Data So Important ...
europeanbusinessmagazine.com

Big Data Analytics Startups Which Are ...
startup-buzz.com

Data Analytics Overtakes Big Data ...
flextrade.com

What is the Shelf Life of Data? | 201...
pobonline.com

The challenges of using data lakes in ...
seleritysas.com

Enable big data analytics and enhance ...
druva.com

Data Accountability and Trust Act ...
rush.house.gov

Data Preprocessing : Concepts ...
towardsdatascience.com

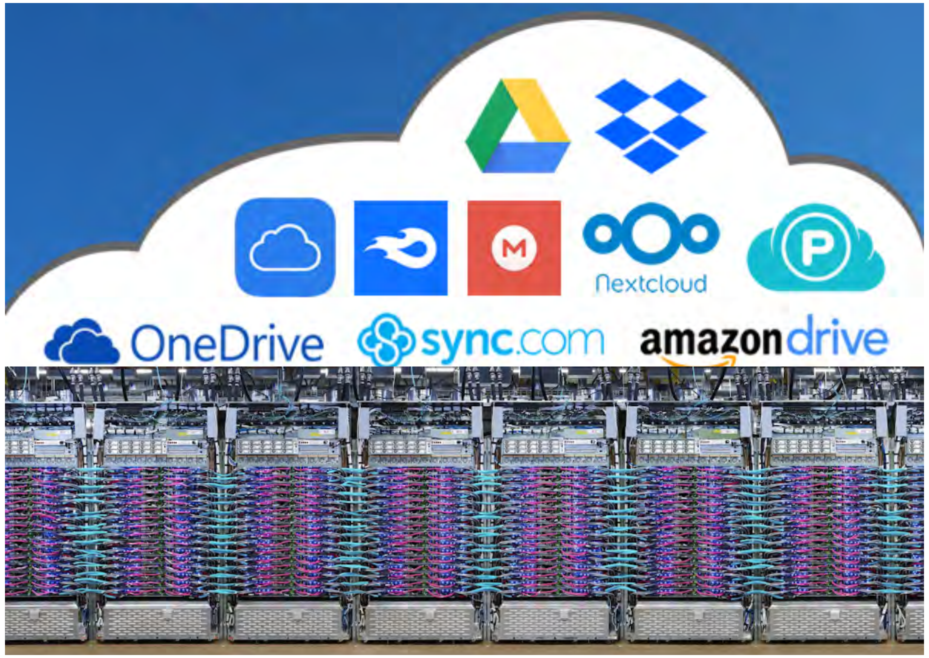
Data Sources and Data Structure

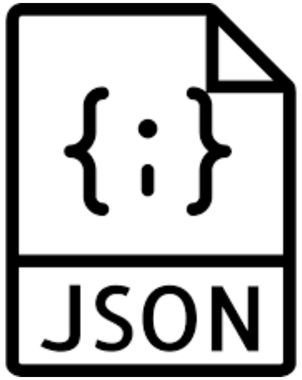
Data Structure

Data Source	Structured	Unstructured
Internal	***Human-generated*** • Surveys • Aptitude tests • Evaluations / KPIs ***Machine-generated*** • Web metrics • Purchase records • Process control measures	***Human-generated*** • E-mails, letters, reports, messages • Calls, memos, videos, illustrations • Corporate / government communication ***Machine-generated*** • AI: Chat-bots / AI-voice / Avatars • Surveillance video • Automated content aggregation
External	***Human-generated*** • Number (re)tweets, likes, searches • Rating scores • Votes ***Machine-generated*** • GPS coordinates, timestamps • Biofeedback / health metrics • Device details (IP, camera type)	***Human-generated*** • Product reviews, complaints • Comments, discussions • Photos, videos, sound ***Machine-generated*** • Content from bots / trolls (fake-news) • Automated selective surveillance • AI composed music, art, etc.

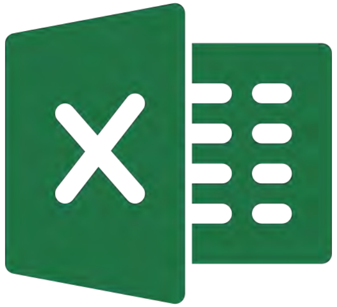
How Data are Stored 1.0

How Data are Stored 2.0



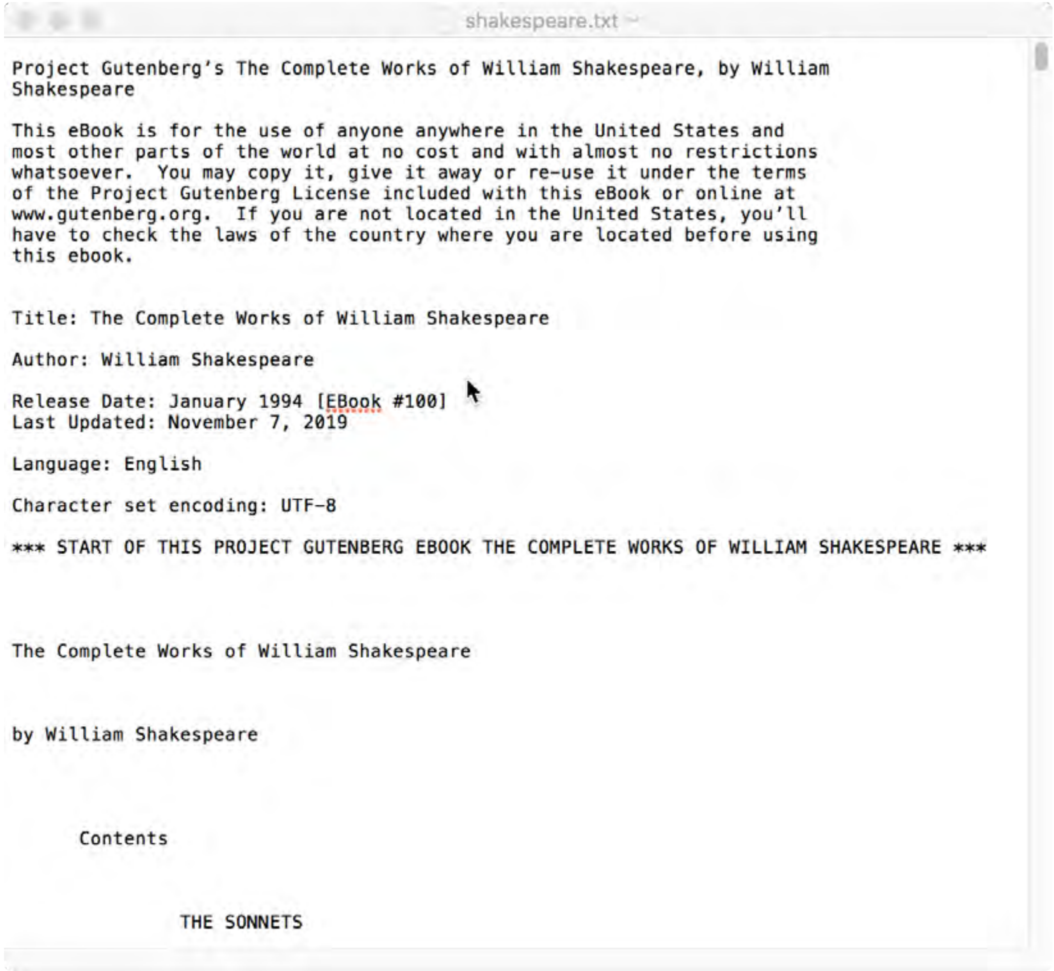


Data Formats for Storage
Some Examples



Data in Flat Text Files

Text Editor View



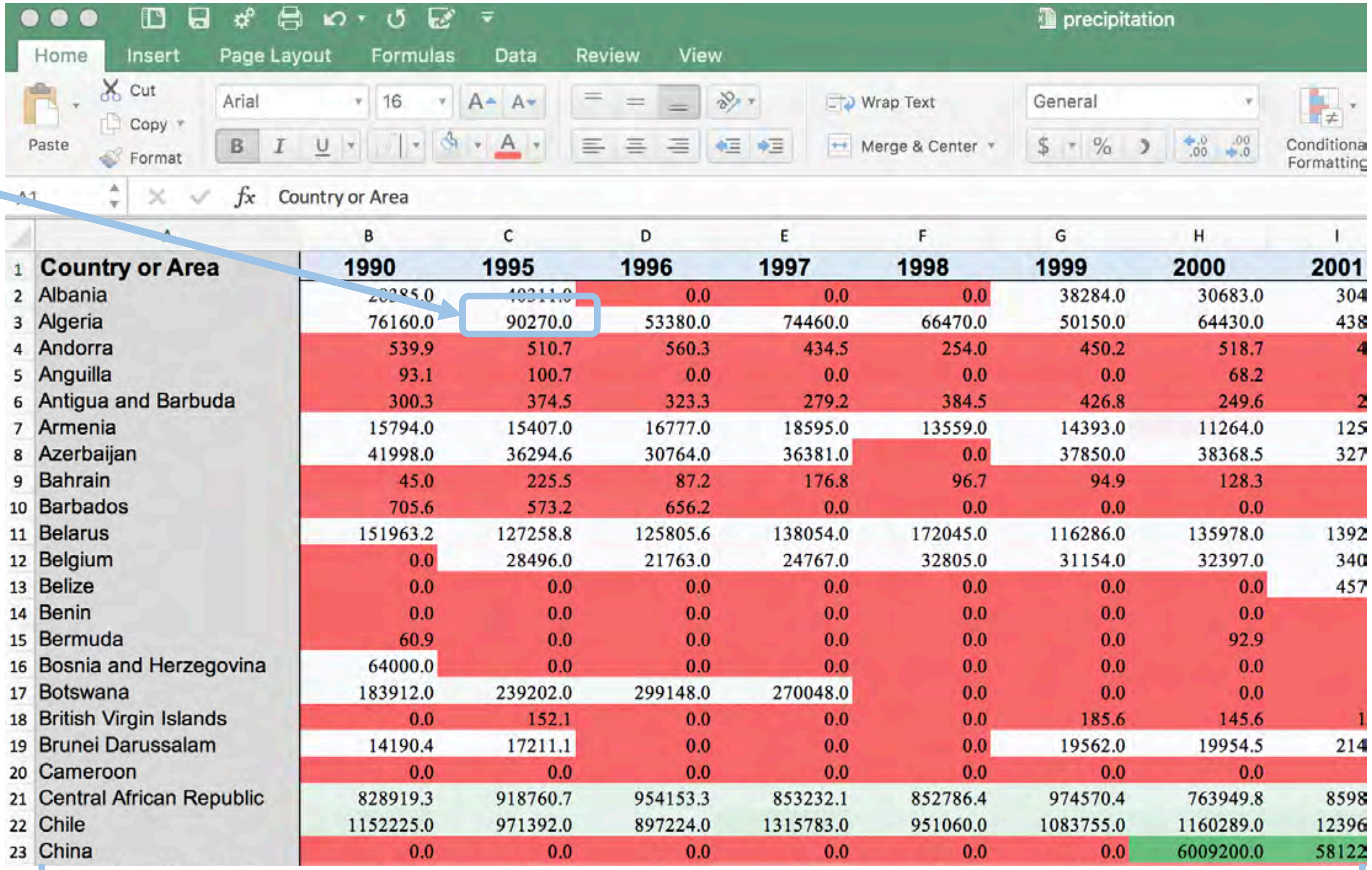
```
Project Gutenberg’s The Complete Works of William Shakespeare, by William
Shakespeare

This eBook is for the use of anyone anywhere in the United States and
most other parts of the world at no cost and with almost no restrictions
whatsoever.  You may copy it, give it away or re-use it under the terms
of the Project Gutenberg License included with this eBook or online at
www.gutenberg.org.  If you are not located in the United States, you’ll
have to check the laws of the country where you are located before using
this ebook.


Title: The Complete Works of William Shakespeare

Author: William Shakespeare

Release Date: January 1994 [EBook #100]
Last Updated: November 7, 2019

Language: English

Character set encoding: UTF-8

*** START OF THIS PROJECT GUTENBERG EBOOK THE COMPLETE WORKS OF WILLIAM SHAKESPEARE ***




The Complete Works of William Shakespeare



by William Shakespeare




      Contents



               THE SONNETS
```

Raw Text View

'\r\nProject Gutenberg’s The Complete Works of William Shakespeare, by William\r\nShakespeare\r\n\r\nThis eBook is for the use of anyone anywhere in the United States and\r\nmost other parts of the world at no cost and with almost no restrictions\r\nwhatsoever. You may copy it, give it away or re-use it under the terms\r\nof the Project Gutenberg License included with this eBook or online at\r\nwww.gutenberg.org. If you are not located in the United States, you’ll\r\nhave to check the laws of the country where you are located before using\r\nthis ebook.\r\n\r\n\r\nTitle: The Complete Works of William Shakespeare\r\n\r\nAuthor: William Shakespeare\r\n\r\nRelease Date: January 1994 [EBook #100]\r\nLast Updated: November 7, 2019\r\n\r\nLanguage: English\r\n\r\nCharacter set encoding: UTF-8\r\n\r\n*** START OF THIS PROJECT GUTENBERG EBOOK THE COMPLETE WORKS OF WILLIAM SHAKESPEARE ***\r\n\r\n\r\n\r\n\r\nThe Complete Works of William Shakespeare\r\n\r\n\r\n\r\nby William Shakespeare\r\n\r\n\r\n\r\n\r\n Contents\r\n\r\n\r\n\r\nTHE SONNETS\r\n\r\n

\r   carriage return
\n   new line

Data in Microsoft Excel



fx Country or Area

	A	B	C	D	E	F	G	H	I
1	Country or Area	1990	1995	1996	1997	1998	1999	2000	2001
2	Albania	[illegible]	[illegible]	0.0	0.0	0.0	38284.0	30683.0	304
3	Algeria	76160.0	90270.0	53380.0	74460.0	66470.0	50150.0	64430.0	438
4	Andorra	539.9	510.7	560.3	434.5	254.0	450.2	518.7	4
5	Anguilla	93.1	100.7	0.0	0.0	0.0	0.0	68.2	
6	Antigua and Barbuda	300.3	374.5	323.3	279.2	384.5	426.8	249.6	2
7	Armenia	15794.0	15407.0	16777.0	18595.0	13559.0	14393.0	11264.0	125
8	Azerbaijan	41998.0	36294.6	30764.0	36381.0	0.0	37850.0	38368.5	327
9	Bahrain	45.0	225.5	87.2	176.8	96.7	94.9	128.3	
10	Barbados	705.6	573.2	656.2	0.0	0.0	0.0	0.0	
11	Belarus	151963.2	127258.8	125805.6	138054.0	172045.0	116286.0	135978.0	1392
12	Belgium	0.0	28496.0	21763.0	24767.0	32805.0	31154.0	32397.0	340
13	Belize	0.0	0.0	0.0	0.0	0.0	0.0	0.0	457
14	Benin	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
15	Bermuda	60.9	0.0	0.0	0.0	0.0	0.0	92.9	
16	Bosnia and Herzegovina	64000.0	0.0	0.0	0.0	0.0	0.0	0.0	
17	Botswana	183912.0	239202.0	299148.0	270048.0	0.0	0.0	0.0	
18	British Virgin Islands	0.0	152.1	0.0	0.0	0.0	185.6	145.6	1
19	Brunei Darussalam	14190.4	17211.1	0.0	0.0	0.0	19562.0	19954.5	214
20	Cameroon	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
21	Central African Republic	828919.3	918760.7	954153.3	853232.1	852786.4	974570.4	763949.8	8598
22	Chile	1152225.0	971392.0	897224.0	1315783.0	951060.0	1083755.0	1160289.0	12396
23	China	0.0	0.0	0.0	0.0	0.0	0.0	6009200.0	58122

Cell

Rows

Columns

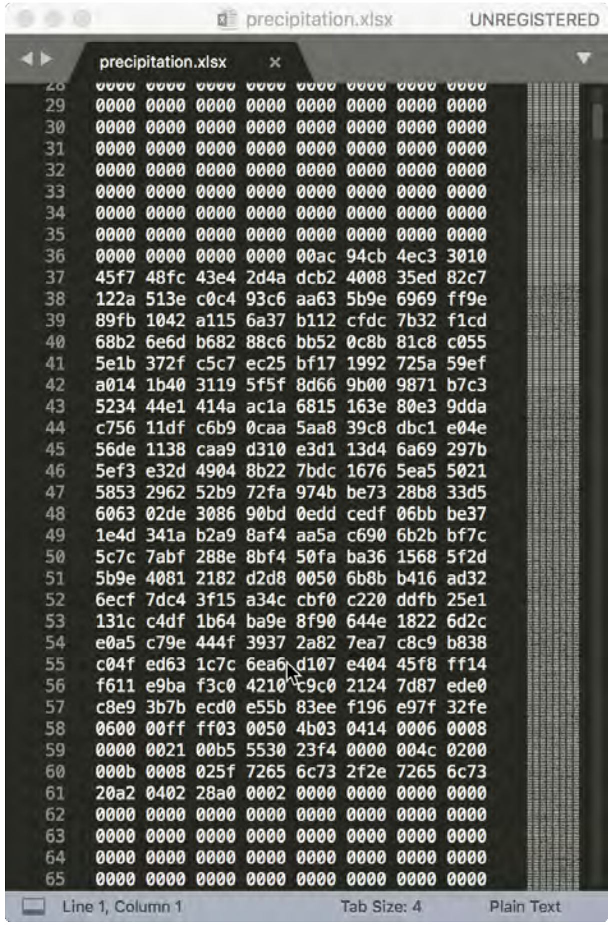
Sheet

File

...

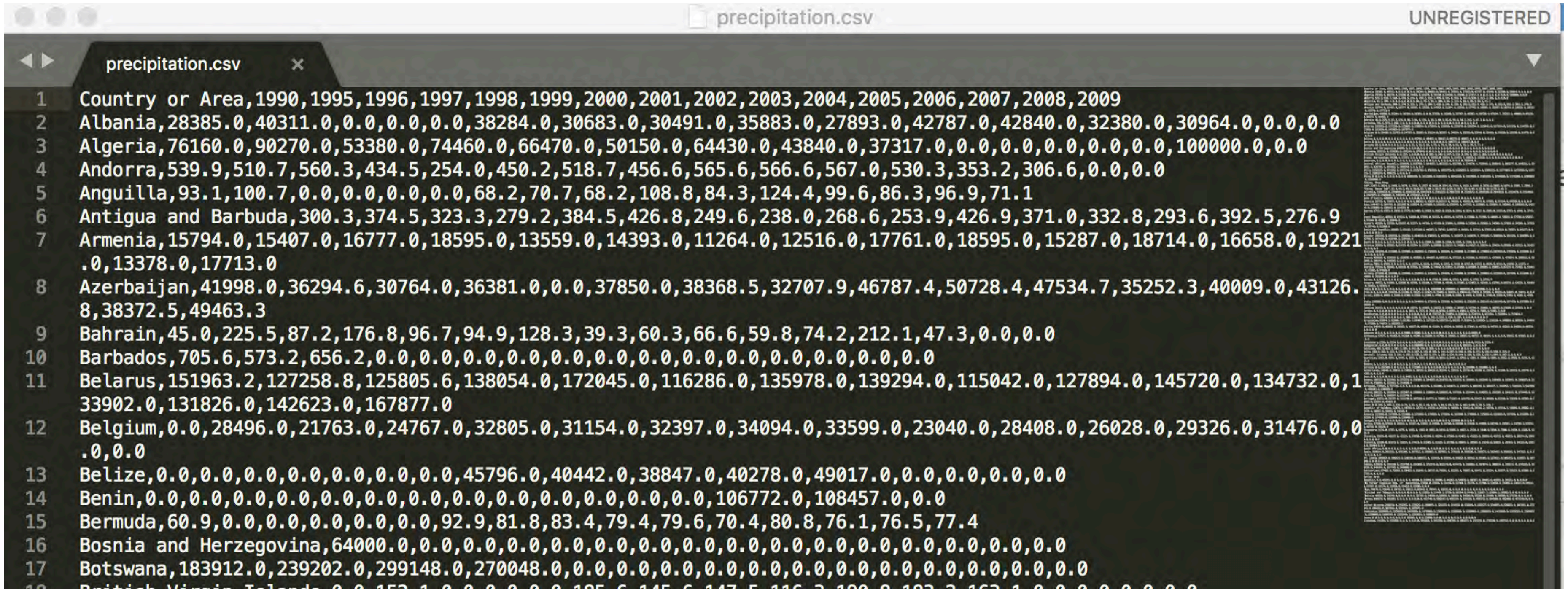
...

Raw File view



```
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 00ac 94cb 4ec3 3010
45f7 48fc 43e4 2d4a dcb2 4008 35ed 82c7
122a 513e c0c4 93c6 aa63 5b9e 6969 ff9e
89fb 1042 a115 6a37 b112 cfdc 7b32 f1cd
68b2 6e6d b682 88c6 bb52 0c8b 81c8 c055
5e1b 372f c5c7 ec25 bf17 1992 725a 59ef
a014 1b40 3119 5f5f 8d66 9b00 9871 b7c3
5234 44e1 414a ac1a 6815 163e 80e3 9dda
c756 11df c6b9 0caa 5aa8 39c8 dbc1 e04e
56de 1138 caa9 d310 e3d1 13d4 6a69 297b
5ef3 e32d 4904 8b22 7bdc 1676 5ea5 5021
5853 2962 52b9 72fa 974b be73 28b8 33d5
6063 02de 3086 90bd 0edd cedf 06bb be37
1e4d 341a b2a9 8af4 aa5a c690 6b2b bf7c
5c7c 7abf 288e 8bf4 50fa ba36 1568 5f2d
5b9e 4081 2182 d2d8 0050 6b8b b416 ad32
6ecf 7dc4 3f15 a34c cbf0 c220 ddfb 25e1
131c c4df 1b64 ba9e 8f90 644e 1822 6d2c
e0a5 c79e 444f 3937 2a82 7ea7 c8c9 b838
c04f ed63 1c7c 6ea6 d107 e404 45f8 ff14
f611 e9ba f3c0 4210 c9c0 2124 7d87 ede0
c8e9 3b7b ecd0 e55b 83ee f196 e97f 32fe
0600 00ff ff03 0050 4b03 0414 0006 0008
0000 0021 00b5 5530 23f4 0000 004c 0200
000b 0008 025f 7265 6c73 2f2e 7265 6c73
20a2 0402 28a0 0002 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000
```

Data in a Comma Separated Value (CSV) File



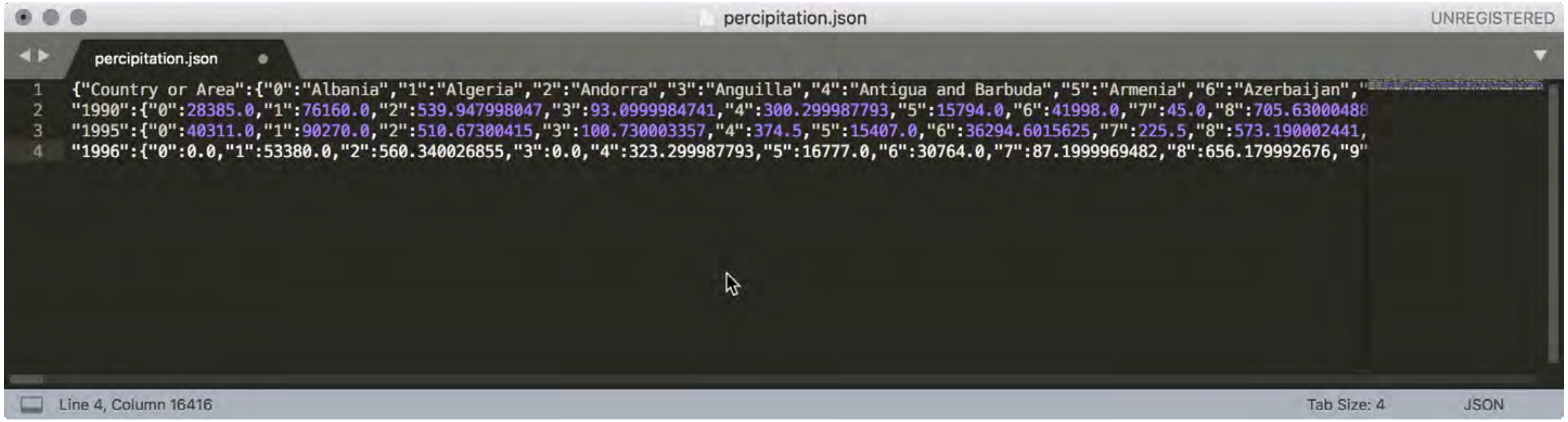
```
Country or Area,1990,1995,1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007,2008,2009
Albania,28385.0,40311.0,0.0,0.0,0.0,38284.0,30683.0,30491.0,35883.0,27893.0,42787.0,42840.0,32380.0,30964.0,0.0,0.0
Algeria,76160.0,90270.0,53380.0,74460.0,66470.0,50150.0,64430.0,43840.0,37317.0,0.0,0.0,0.0,0.0,0.0,100000.0,0.0
Andorra,539.9,510.7,560.3,434.5,254.0,450.2,518.7,456.6,565.6,566.6,567.0,530.3,353.2,306.6,0.0,0.0
Anguilla,93.1,100.7,0.0,0.0,0.0,0.0,68.2,70.7,68.2,108.8,84.3,124.4,99.6,86.3,96.9,71.1
Antigua and Barbuda,300.3,374.5,323.3,279.2,384.5,426.8,249.6,238.0,268.6,253.9,426.9,371.0,332.8,293.6,392.5,276.9
Armenia,15794.0,15407.0,16777.0,18595.0,13559.0,14393.0,11264.0,12516.0,17761.0,18595.0,15287.0,18714.0,16658.0,19221
.0,13378.0,17713.0
Azerbaijan,41998.0,36294.6,30764.0,36381.0,0.0,37850.0,38368.5,32707.9,46787.4,50728.4,47534.7,35252.3,40009.0,43126.
8,38372.5,49463.3
Bahrain,45.0,225.5,87.2,176.8,96.7,94.9,128.3,39.3,60.3,66.6,59.8,74.2,212.1,47.3,0.0,0.0
Barbados,705.6,573.2,656.2,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
Belarus,151963.2,127258.8,125805.6,138054.0,172045.0,116286.0,135978.0,139294.0,115042.0,127894.0,145720.0,134732.0,1
33902.0,131826.0,142623.0,167877.0
Belgium,0.0,28496.0,21763.0,24767.0,32805.0,31154.0,32397.0,34094.0,33599.0,23040.0,28408.0,26028.0,29326.0,31476.0,0
.0,0.0
Belize,0.0,0.0,0.0,0.0,0.0,0.0,0.0,45796.0,40442.0,38847.0,40278.0,49017.0,0.0,0.0,0.0,0.0
Benin,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,106772.0,108457.0,0.0
Bermuda,60.9,0.0,0.0,0.0,0.0,0.0,92.9,81.8,83.4,79.4,79.6,70.4,80.8,76.1,76.5,77.4
Bosnia and Herzegovina,64000.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
Botswana,183912.0,239202.0,299148.0,270048.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
```

Lines (with line break at end of each line often indicated by "\n" or by ↵)

Columns are separated by commas—other separators possible(!!!)

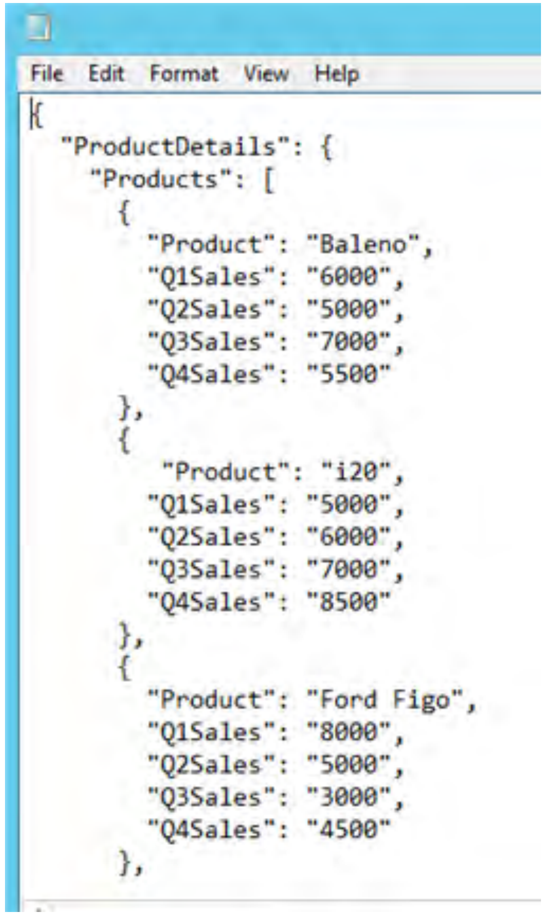
Note that the line numbering is not part of the csv file and was added for better readability by the text editor used (Sublime Text—get it at https://www.sublimetext.com/)

Data in a JavaScript Object Notation (JSON) File



```
{"Country or Area":{"0":"Albania","1":"Algeria","2":"Andorra","3":"Anguilla","4":"Antigua and Barbuda","5":"Armenia","6":"Azerbaijan","
"1990":{"0":28385.0,"1":76160.0,"2":539.947998047,"3":93.0999984741,"4":300.299987793,"5":15794.0,"6":41998.0,"7":45.0,"8":705.63000488
"1995":{"0":40311.0,"1":90270.0,"2":510.67300415,"3":100.730003357,"4":374.5,"5":15407.0,"6":36294.6015625,"7":225.5,"8":573.190002441,
"1996":{"0":0.0,"1":53380.0,"2":560.340026855,"3":0.0,"4":323.299987793,"5":16777.0,"6":30764.0,"7":87.1999969482,"8":656.179992676,"9"
```

Note: originally, there were no line breaks—everything was on a single long line. I added breaks to better illustrate how information was organized



```
{
  "ProductDetails": {
    "Products": [
      {
        "Product": "Baleno",
        "Q1Sales": "6000",
        "Q2Sales": "5000",
        "Q3Sales": "7000",
        "Q4Sales": "5500"
      },
      {
         "Product": "i20",
        "Q1Sales": "5000",
        "Q2Sales": "6000",
        "Q3Sales": "7000",
        "Q4Sales": "8500"
      },
      {
        "Product": "Ford Figo",
        "Q1Sales": "8000",
        "Q2Sales": "5000",
        "Q3Sales": "3000",
        "Q4Sales": "4500"
      },
```

- JSON is a lightweight data-interchange format
- JSON is "self-describing" and easy to understand
- JSON is language independent *
- JSON is text only: it can easily be sent to and from a server, and used as a data format by any programming language

** JSON uses JavaScript syntax, but the JSON format is text only. Text can be read and used as a data format by any programming language.*

Data in a Pickle File

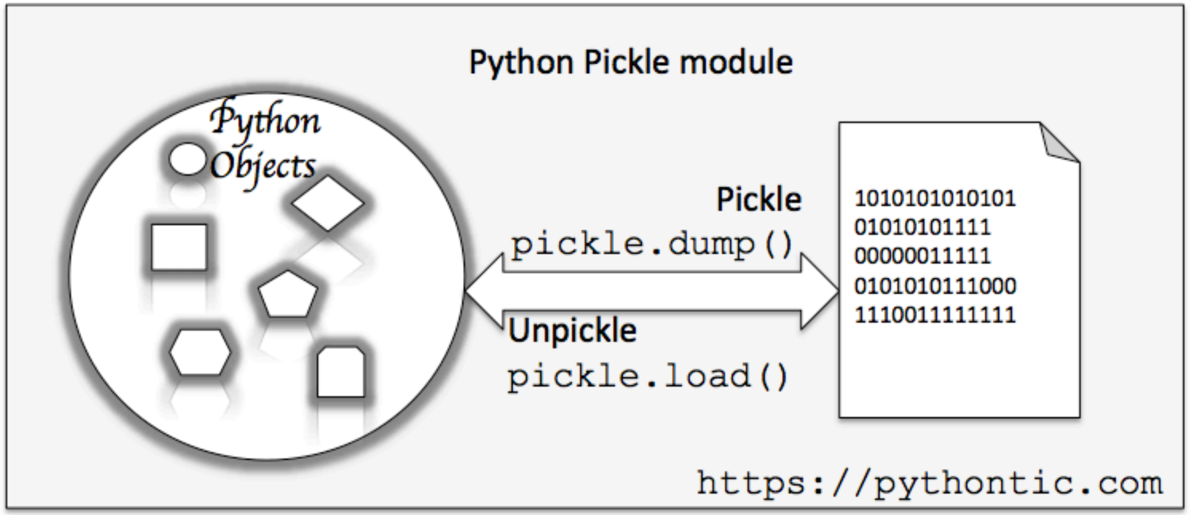
The **Python Pickle Module** implements binary protocols for serializing and de-serializing a Python object structure.

"Pickling" is the process whereby a Python object hierarchy is converted into a byte stream, and ***"unpickling"*** is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy.

Pickling (and unpickling) is alternatively known as "serialization", "marshalling," or "flattening"; however, to avoid confusion, the terms used in this course are "pickling" and "unpickling".

When Not To Use pickle

If you want to use data across different programming languages, pickle is not recommended. In contrast, JSON is standardized and language-independent. This is a serious advantage over pickle. It's also much faster than pickle.



https://pythontic.com

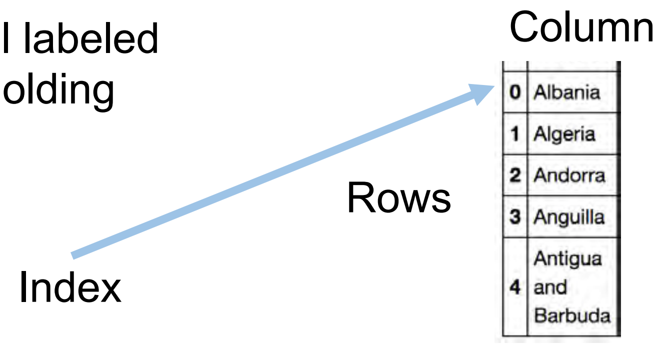
Check-out DataCamp's Tutorial

https://www.datacamp.com/community/tutorials/pickle-python-tutorial

Data in a Pandas Series and DataFrame

Series

A one-dimensional labeled array capable of holding any data type

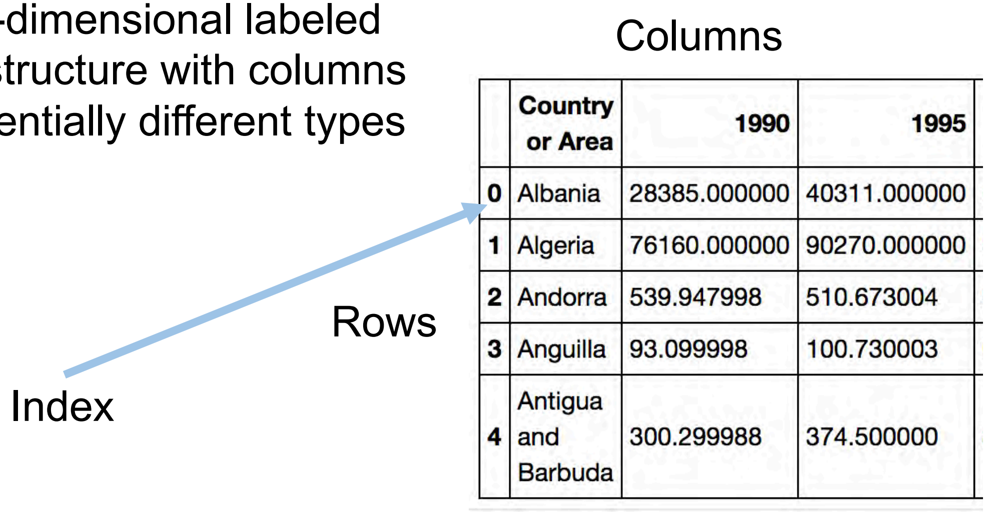


0	Albania
1	Algeria
2	Andorra
3	Anguilla
4	Antigua and Barbuda

series1 = pd.Series([‘Albania’, ‘Algeria’, ‘Andorra’, ‘Anguilla’, ‘Antigua and Barbuda’], index = [0, 1, 2, 3, 4])

DataFrame

A two-dimensional labeled data structure with columns of potentially different types



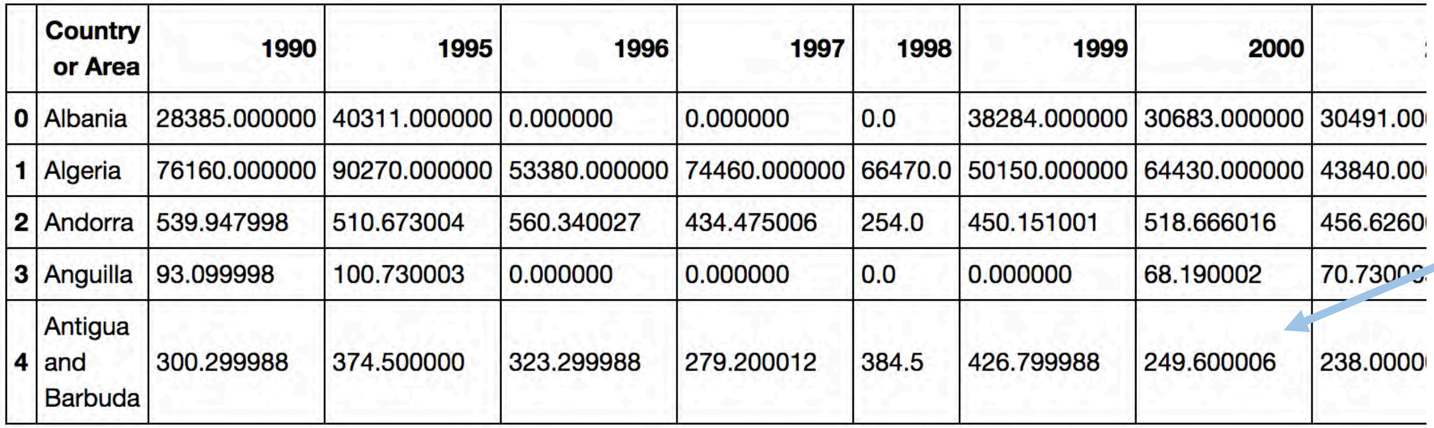
	Country or Area	1990	1995
0	Albania	28385.000000	40311.000000
1	Algeria	76160.000000	90270.000000
2	Andorra	539.947998	510.673004
3	Anguilla	93.099998	100.730003
4	Antigua and Barbuda	300.299988	374.500000

data = {‘Country or Area’: [‘Albania’, ‘Algeria’, ‘Andorra’, ‘Anguilla’, ‘Antigua and Barbuda’],
‘1990’: [28385, 76160, 539.947998, 93.099998, 300.299988],
‘1995’: [40311, 90270, 510.673004, 100.730003, 374.500000]}

df = pd.DataFrame(data, columns = [‘Country or Area’, ‘1990’, ‘1995’])

Many Names for (often) the same Thing

Dataset, File, Table, Sheet, Data Frame



	Country or Area	1990	1995	1996	1997	1998	1999	2000	
0	Albania	28385.000000	40311.000000	0.000000	0.000000	0.0	38284.000000	30683.000000	30491.00
1	Algeria	76160.000000	90270.000000	53380.000000	74460.000000	66470.0	50150.000000	64430.000000	43840.00
2	Andorra	539.947998	510.673004	560.340027	434.475006	254.0	450.151001	518.666016	456.6260
3	Anguilla	93.099998	100.730003	0.000000	0.000000	0.0	0.000000	68.190002	70.7300
4	Antigua and Barbuda	300.299988	374.500000	323.299988	279.200012	384.5	426.799988	249.600006	238.0000

Rows
Instances
Cases
Examples
Observations
Tuples

Cell,
Value

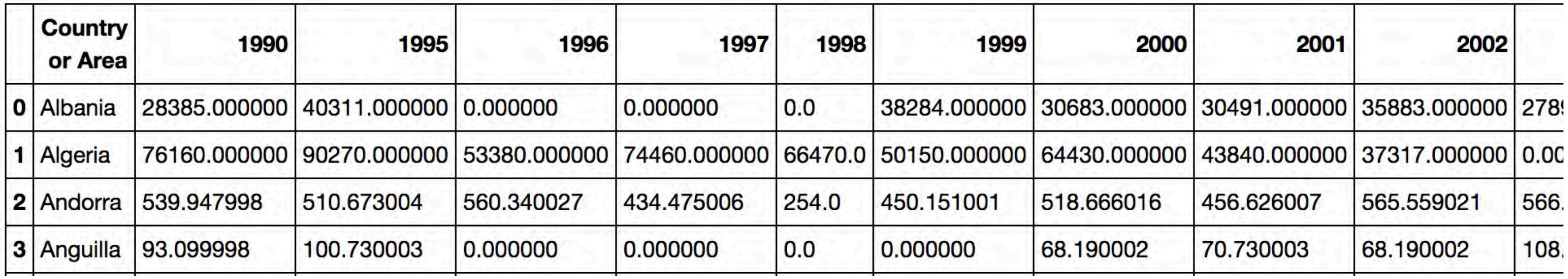
Columns, Features, Variables, Fields

- Columns generally have the same type of data, but rows can be heterogeneous
- Types tell the computer how big something is and what operations it supports
- Types help us avoid errors caused by applying the wrong operations to the data

Tidy Data Concept

“Tidy Data” are also known in statistics as a *model matrix* or *data matrix:*

- Standard method of displaying a multivariate set of data
- Rows correspond to sample individuals and columns to variables
- Entry in the i^{th} row and j^{th} column gives value of the j^{th} variate as measured or observed on the i^{th} individual



	Country or Area	1990	1995	1996	1997	1998	1999	2000	2001	2002	
0	Albania	28385.000000	40311.000000	0.000000	0.000000	0.0	38284.000000	30683.000000	30491.000000	35883.000000	278
1	Algeria	76160.000000	90270.000000	53380.000000	74460.000000	66470.0	50150.000000	64430.000000	43840.000000	37317.000000	0.00
2	Andorra	539.947998	510.673004	560.340027	434.475006	254.0	450.151001	518.666016	456.626007	565.559021	566.
3	Anguilla	93.099998	100.730003	0.000000	0.000000	0.0	0.000000	68.190002	70.730003	68.190002	108.

More recently, Wickham defined "Tidy Data" as data sets that are arranged such that each variable is a column, and each observation (or case) is a row

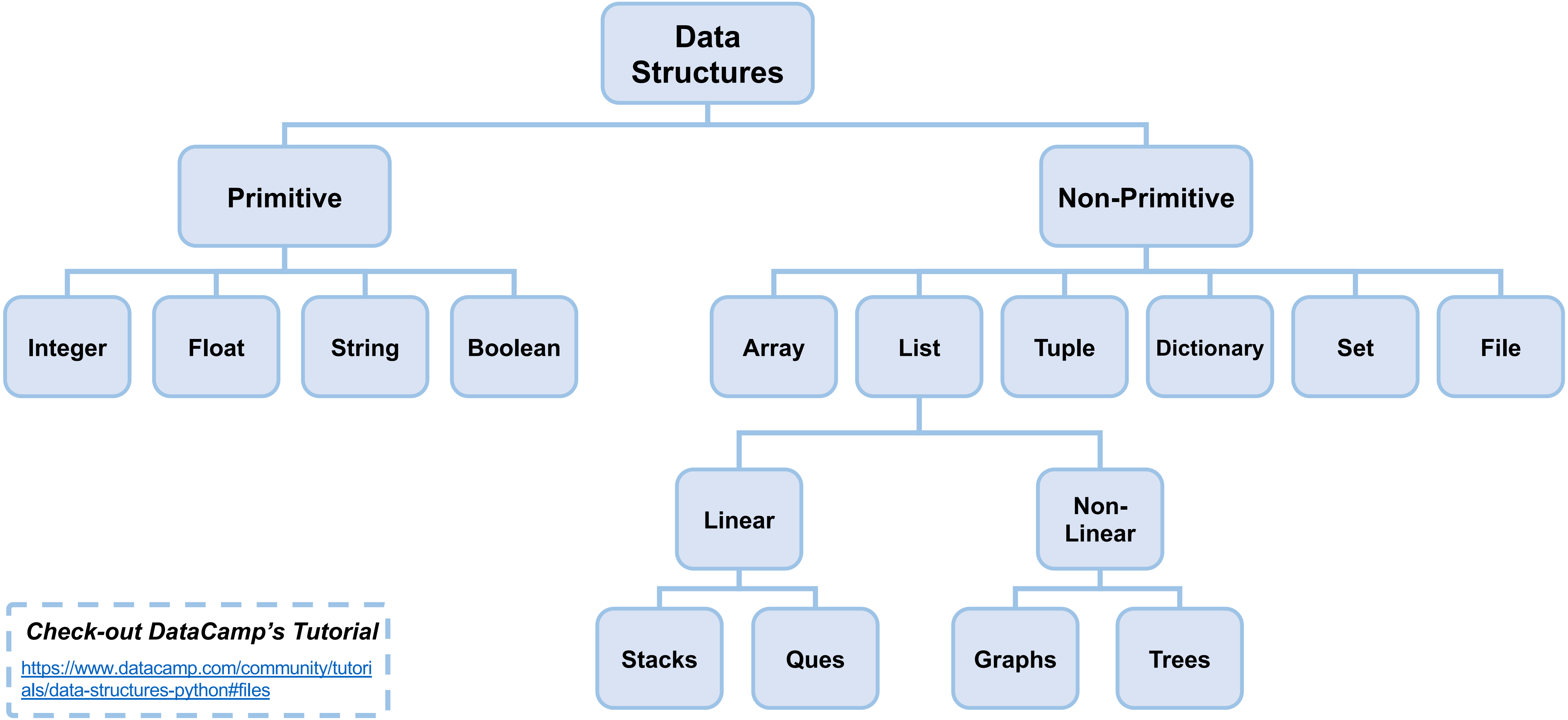
- Each column should be common type of measurement/feature
- Each row has all observations from one measurement/experiment
- Table should unite all observations for the features in common

Wickham, H., 2014. Tidy data. Journal of Statistical Software, 59(10), pp.1-23.

Knowledge Check: Types of Data

Categorical	category names may be unrelated – no median or mean
Boolean/binary	two categories
Ordinal	Can order, so median makes sense.
Interval	Evenly spaced, so mean makes sense. No zero.
Time	Interval with (daily or seasonal) patterns
Ratio	Even spacing, well-defined zero
Spatial	Multidimensional ratio coordinates

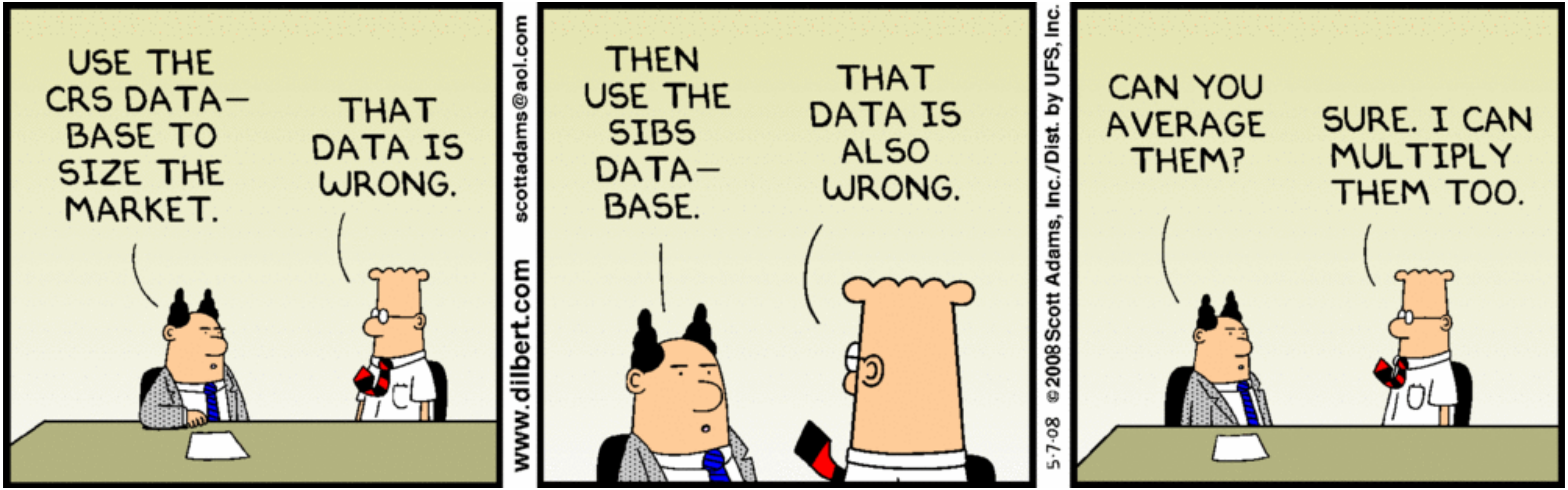
Data Structures and Types in Python



Check-out DataCamp's Tutorial

https://www.datacamp.com/community/tutorials/data-structures-python#files

Imperfect Data


USE THE CRS DATA-BASE TO SIZE THE MARKET.
THAT DATA IS WRONG.
scottadams@aol.com
www.dilbert.com
THEN USE THE SIBS DATA-BASE.
THAT DATA IS ALSO WRONG.
5-7-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.
CAN YOU AVERAGE THEM?
SURE. I CAN MULTIPLY THEM TOO.

Key Insight

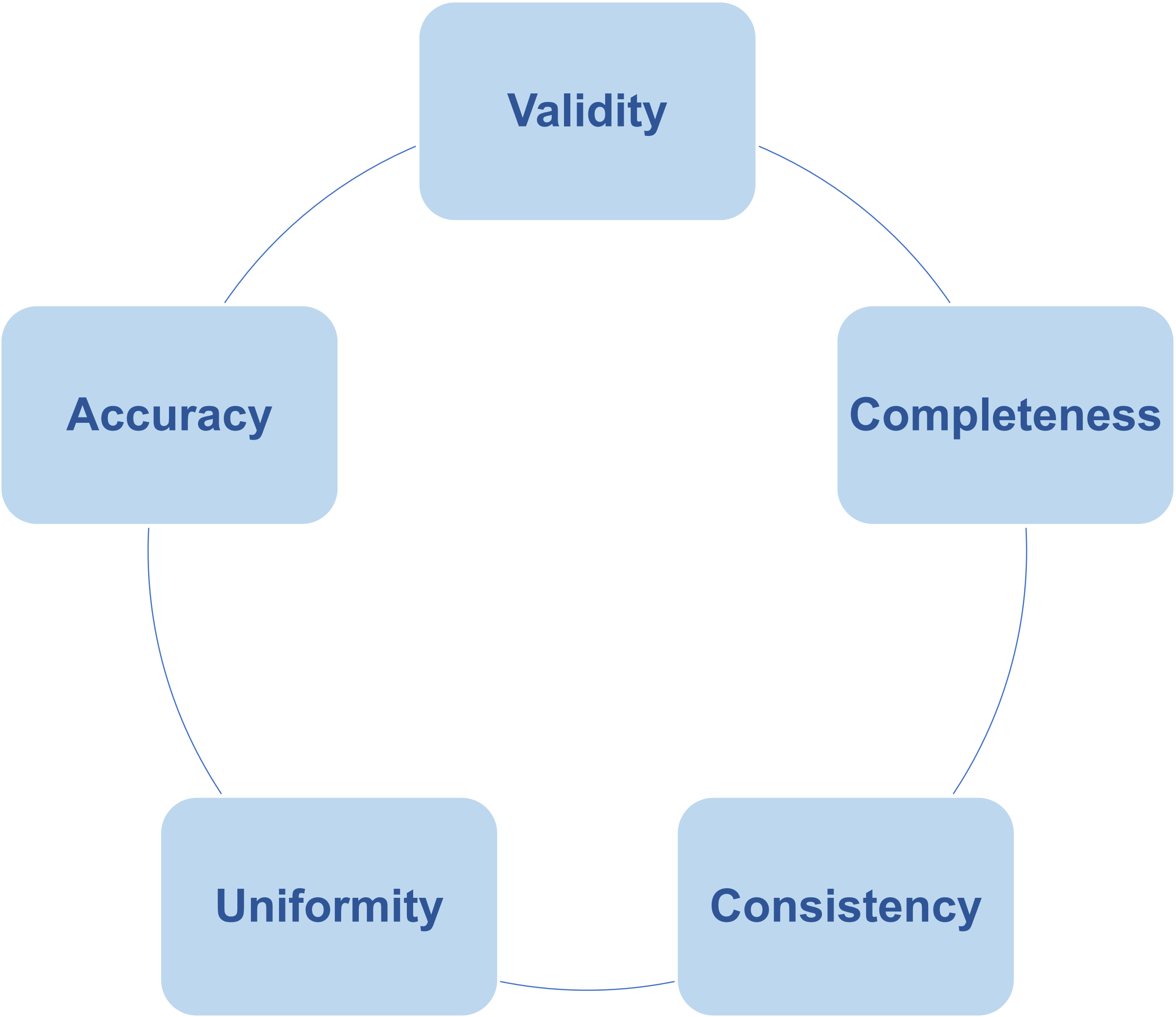
Rule #1 Garbage in, garbage out

Rule #2 Quality data beats fancy algorithms

What can go Wrong? Everything!

Each combination of
{table, records, fields} × {names, inconsistent types,
duplicates, missing, merged,
disaggregated, pivoted, …}

Data Quality



Validity

Completeness

Consistency

Uniformity

Accuracy

Validity

Data-Type Constraints: values in a column must be of same datatype (e.g., boolean, numeric, date, etc.)

Range Constraints: typically, numbers or dates should fall within a certain range

Mandatory Constraints*:* certain columns cannot be empty

Unique Constraints: a field, or a combination of fields, must be unique across a dataset (test for duplicates!)

Set-Membership Constraints**:** values of a column come from a set of discrete values (e.g., gender)

Regular Expression Patterns: text fields that have to be in a certain pattern (e.g., phone numbers)

Cross-field Validation: certain conditions that span across multiple fields must hold. For example, a patient’s date of discharge from the hospital cannot be earlier than the date of admission.

Foreign-key Constraints**:** as in relational databases, a foreign key column can’t have a value that does not exist in the referenced primary key.

Accuracy

The degree to which the data is close to the true values

First defining all possible *valid* values helps easily spot invalid values

BUT, this does not mean they are *accurate*

A *valid* ROI is 22%, but this might not be *accurate* for your firm

Note that *accuracy* is not equivalent to *precision:* Saying that Horizon Hobby (a U.S. hobby firm that makes remote control cars) is a manufacturer of electric cars is true, but not very precise.

Completeness

The degree to which all required data are known

Missing data is going to happen for various reasons

- → Error in collection process
- → Accidental deletion
- → No response
- → Not applicable
- → etc.

Mitigate the problem by collecting it again, imputing it, or assigning a meaning to it (e.g., not applicable)

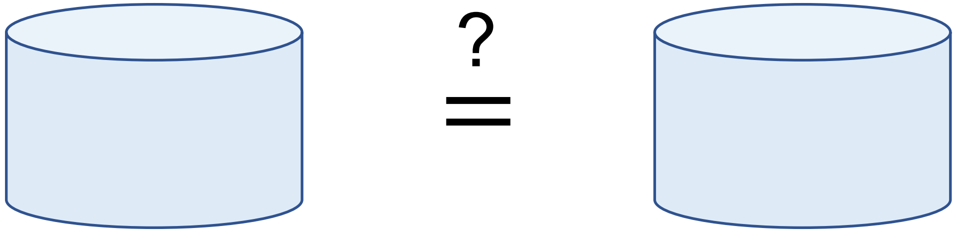
Consistency

The degree to which the data is consistent, within the same data set or across multiple data sets

Inconsistency occurs when two values in the data set contradict each other

A record in a customer database may indicate that a customer’s lifetime value (CLV) is high even though they have not purchased anything in the past 5 years and their previous purchases were only clearance items with a total value of $99.

Similarly, a client’s credit score may be 520 in one database and 789 in another.



Uniformity

The degree to which the data is specified using the same unit of measure

Revenue may be recorded either in U.S. Dollars or in Euros

Dates might be in U.S. or in European formats

→ Data should always be converted to a common measure unit

Data Cleaning Workflow

Inspect Identify unexpected, incorrect, and inconsistent data

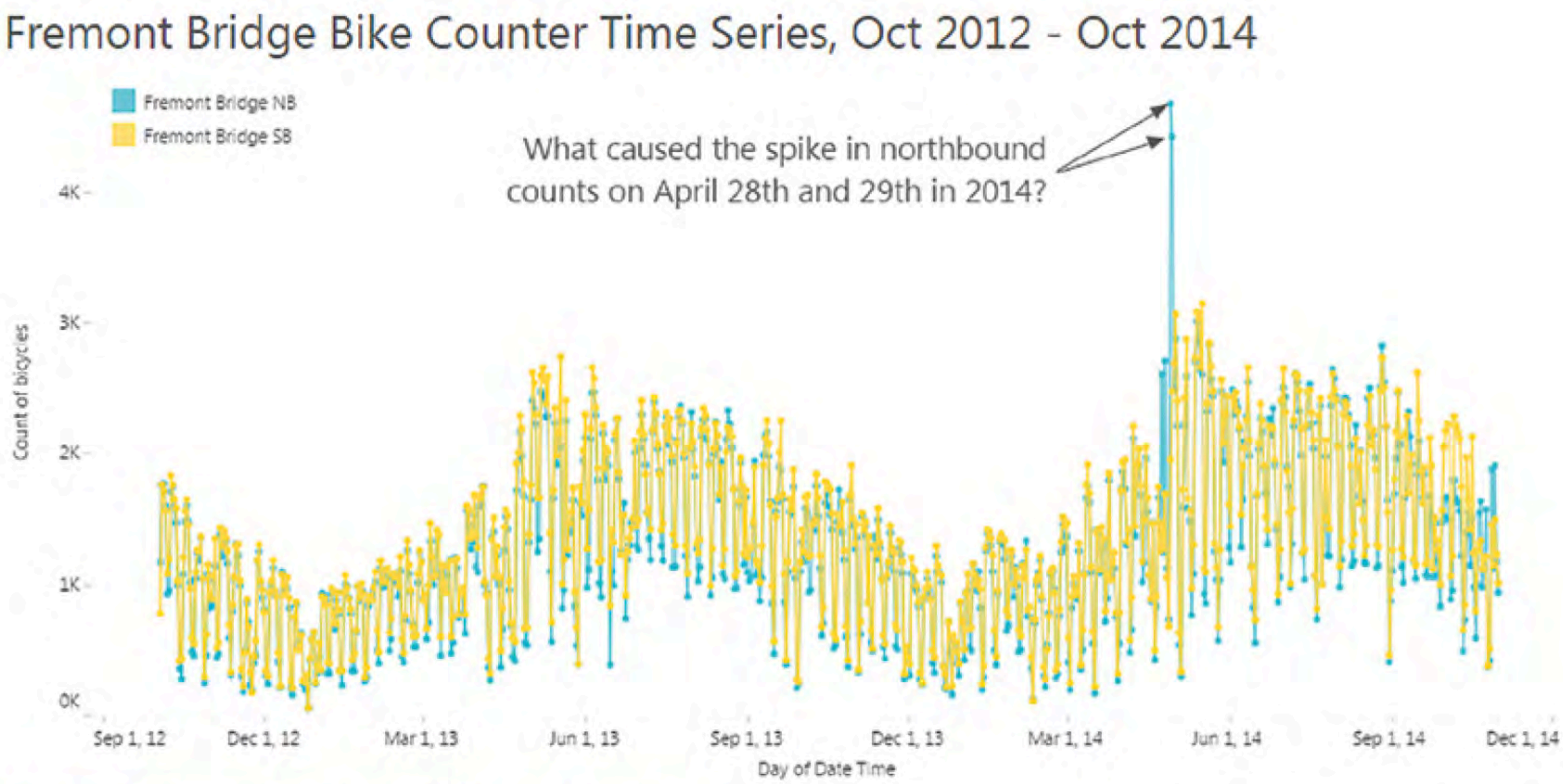
Clean Resolve data quality violations, and fix or remove identified anomalies

Verify Re-inspect to verify correctness

Report Create report that details the changes made
and the quality of the currently stored data

ANOMALY

Anomaly!?



Fremont Bridge Bike Counter Time Series, Oct 2012 - Oct 2014

What caused the spike in northbound counts on April 28th and 29th in 2014?

Data source: http://www.seattle.gov/transportation/bikecounter_fremont.htm

Anomalies in Data

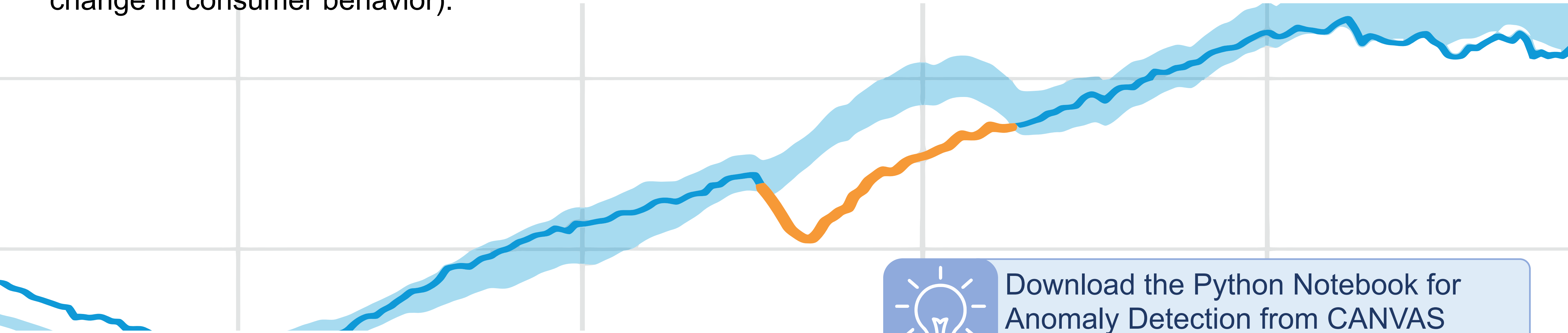
Definition of Anomaly

1. Something different, abnormal, peculiar, or not easily classified
2. Deviation from the common rule **:** irregularity

Source: Merriam-Webster

Anomaly detection is a step in data mining that identifies data points, events, and/or observations that deviate from a dataset's normal behavior.

Anomalies can **indicate** critical incidents, errors in the data collection, or potential business opportunities (e.g., a change in consumer behavior).



Download the Python Notebook for Anomaly Detection from CANVAS and put it on your Google Drive

Anomalies: Novel vs. Outlier

Humans are relatively good at finding patterns,
and they're also quite good at finding things that don't fit a pattern.

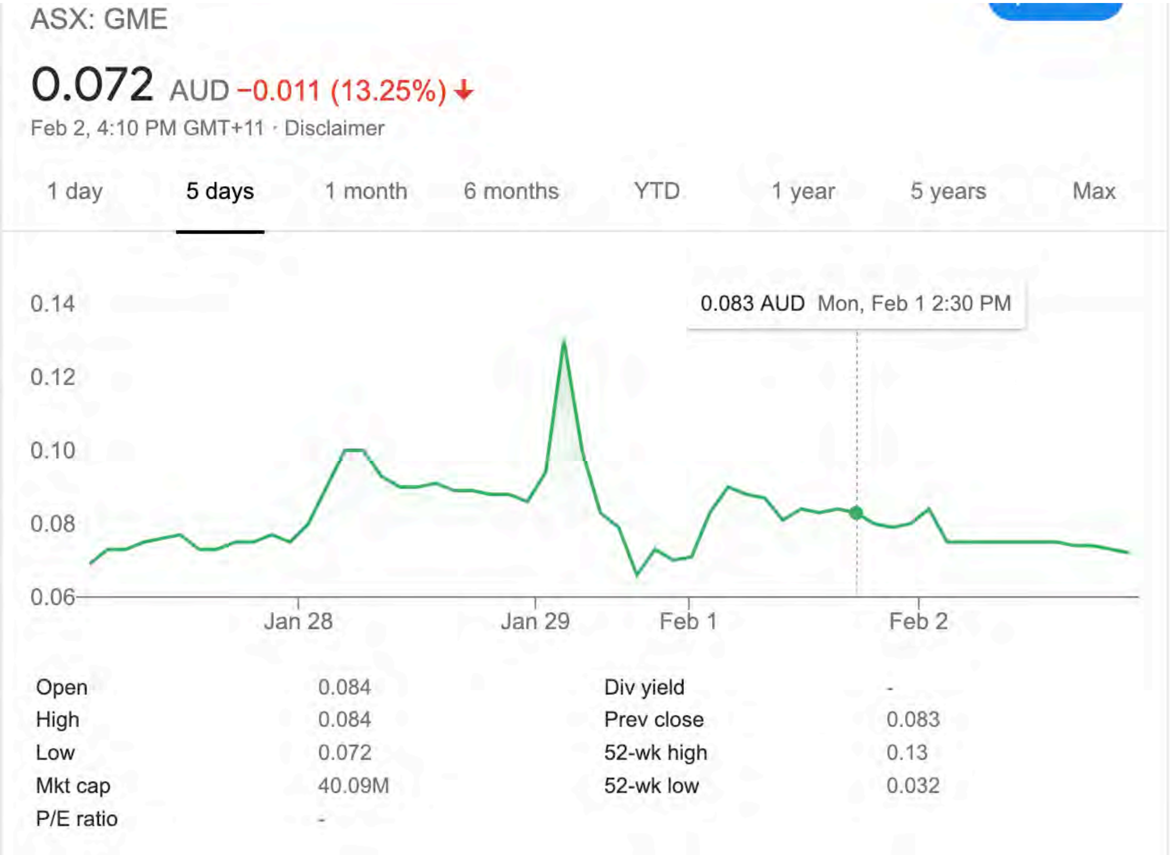
Novelty detection is the mechanism by which an intelligent organism can identify an incoming sensory pattern as being hitherto unknown.

- Consider N observations from the same distribution described by p features
- Let’s add one more observation
- Is the new observation so different from the others that we can doubt it is regular?
  (i.e., does it come from the same distribution?)

Outlier detection is the identification of rare items, events or observations which raise suspicions by differing significantly from most of the data.

- Similar to novelty detection
- Goal is to separate a core of regular observations from some polluting ones
- BUT: we don’t have a clean data set representing the population of regular observations that can be used as basis

Anomalies in Financial Markets

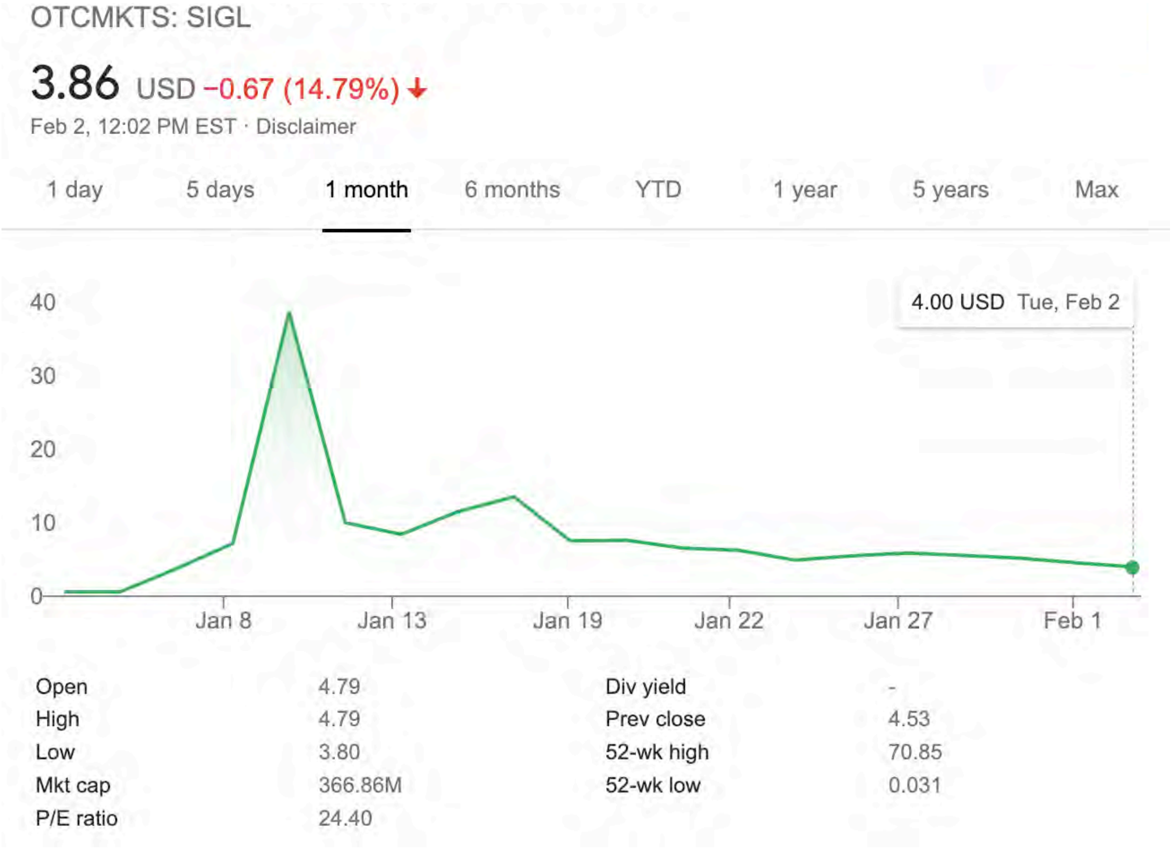


ASX: GME

0.072 AUD −0.011 (13.25%) ↓

Feb 2, 4:10 PM GMT+11 · Disclaimer

Open	0.084	Div yield	-
High	0.084	Prev close	0.083
Low	0.072	52-wk high	0.13
Mkt cap	40.09M	52-wk low	0.032
P/E ratio	-		



OTCMKTS: SIGL

3.86 USD −0.67 (14.79%) ↓

Feb 2, 12:02 PM EST · Disclaimer

Open	4.79	Div yield	-
High	4.79	Prev close	4.53
Low	3.80	52-wk high	70.85
Mkt cap	366.86M	52-wk low	0.031
P/E ratio	24.40		

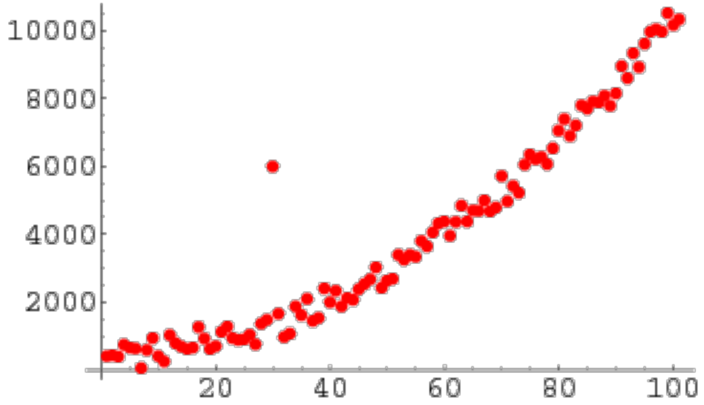
1. What is going on?
2. Novelty, Outlier, neither or both?
3. Opportunity, Error, something else?
5. What to do with it?

Outlier Detection

Threats and Opportunities

Outliers might be

- items that are so far outside the norm that they need not be considered
- errors in the data
- novel items that are worth exploring



Outliers can

- alert you to a problem or and unknown opportunity
- cause potentially severe errors in your models that lead to incorrect conclusions

Important to detect all outliers to

- Analyze them to know why you had them there in the first place
- Eliminate them if appropriate

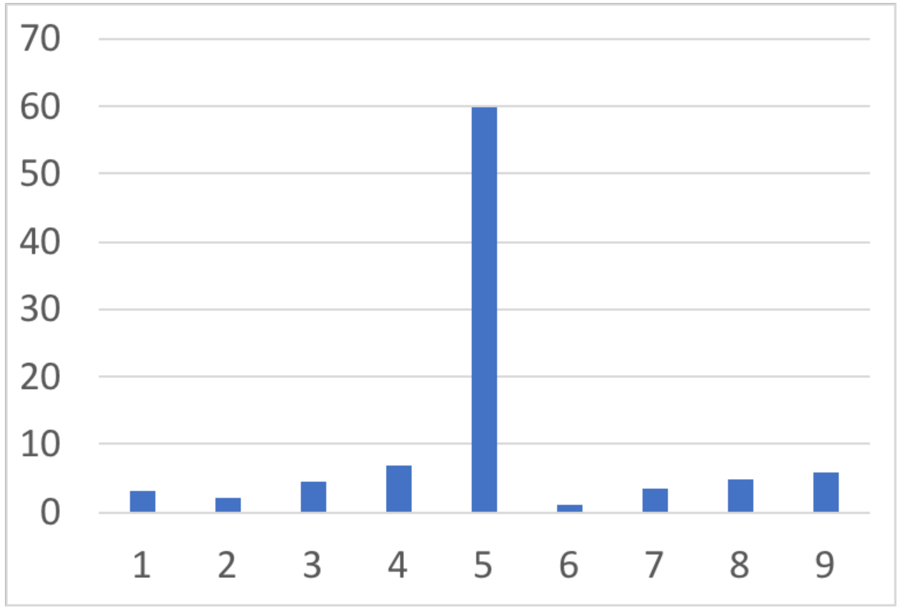
Based on Will Badr’s 2019 post: “5 Ways to Detect Outliers/Anomalies That Every Data Scientist Should Know (Python Code)”

Outlier Detection: Visual Inspection

Easy

[2.99,1.99,4.49,6.99,59.90,0.99,3.29,4.89,5.79]

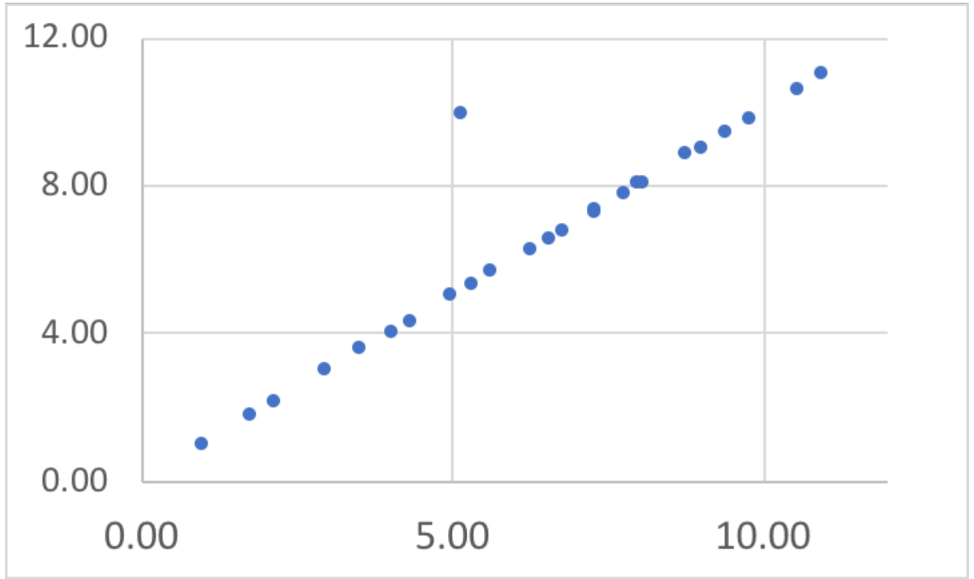
Obvious



Harder

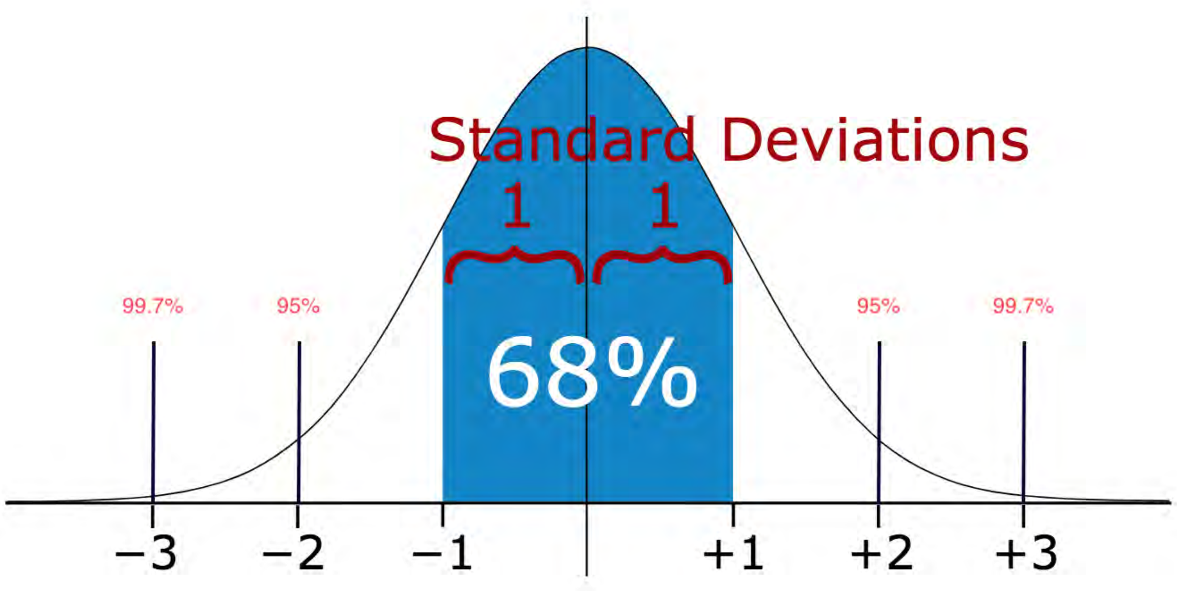
1.00	0.95
1.75	1.68
2.17	2.08
2.97	2.95
3.51	3.51
4.06	3.98
4.35	4.26
4.99	4.95
5.63	5.60
6.57	6.48
7.32	7.23
8.01	8.00
5.16	9.90
5.31	5.26
6.26	6.18
6.81	6.72
7.31	7.27
7.78	7.69
8.06	8.02
8.78	8.77
9.03	8.95
9.42	9.39
9.79	9.75
10.57	10.52
10.95	10.95

Obvious



Outlier Detection: Standard Deviation

Normal distribution:
About 68% of the data values lie within one standard deviation of the mean
About 95% are within two standard deviations
About 99.7% lie within three standard deviations

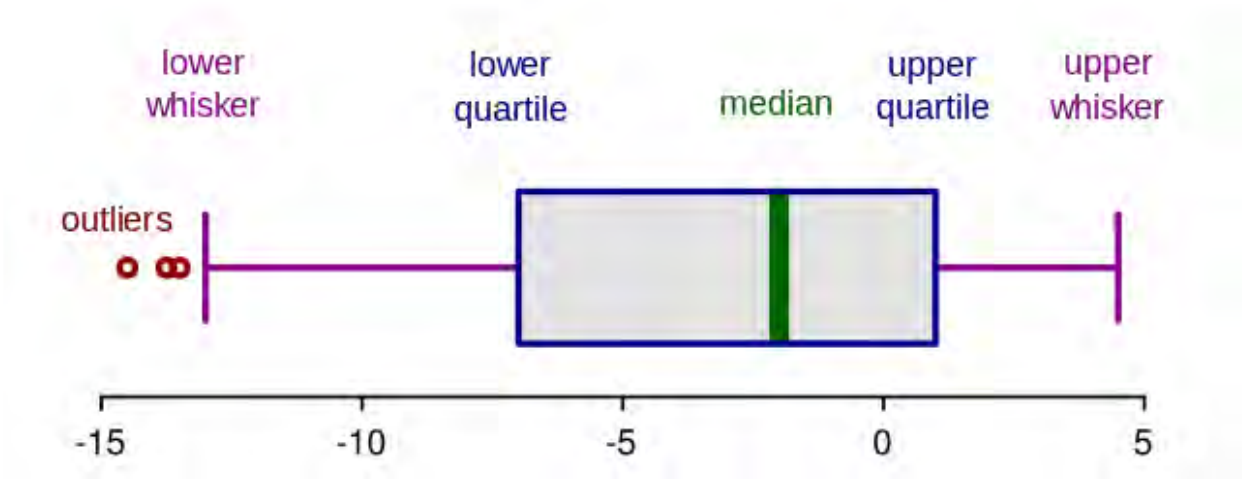


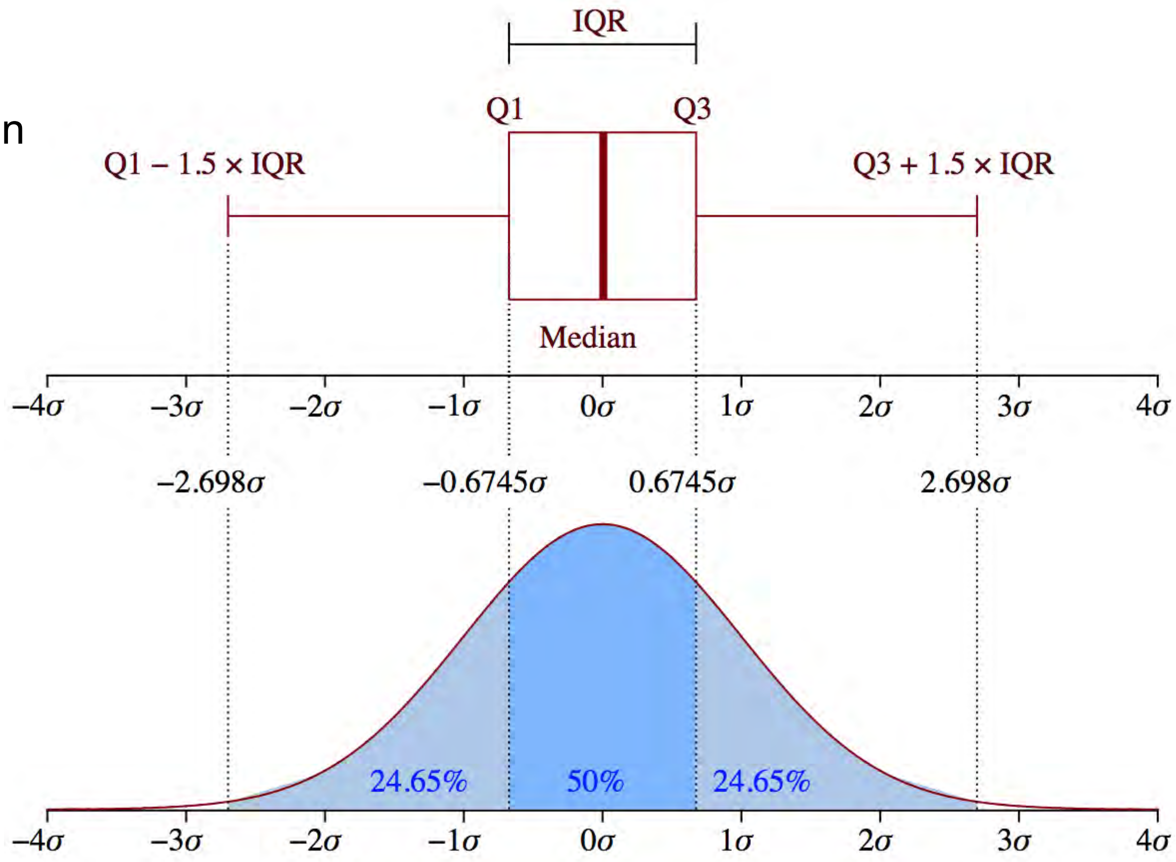
Any data point that is more than 3 times the standard deviation might be an outlier candidate

Based on Will Badr's 2019 post: "5 Ways to Detect Outliers/Anomalies That Every Data Scientist Should Know (Python Code)" and Wikipedia.org

Outlier Detection: Boxplots

- Graphical depiction of numerical data through their quantiles
- Lower and upper whiskers as the boundaries of the data distribution
- Data points outside of the whiskers can be considered outliers





Based on Will Badr's 2019 post: "5 Ways to Detect Outliers/Anomalies That Every Data Scientist Should Know (Python Code)" and Wikipedia.org

Outlier Detection: Clustering

- **DBScan** can be used as density-based anomaly detection method
- Applicable to single or multi-dimensional data
- **Identifies different "Points"** (i.e., samples, observations, objects, items)

●● **Core Points** are central samples (i.e., observations, objects, items) in a cluster

- Set the min number to form a cluster using the *hyperparameter* **min_samples**
- Set the maximum allowable distance between two samples to still be considered as being in same cluster with **eps** hyperparameter)

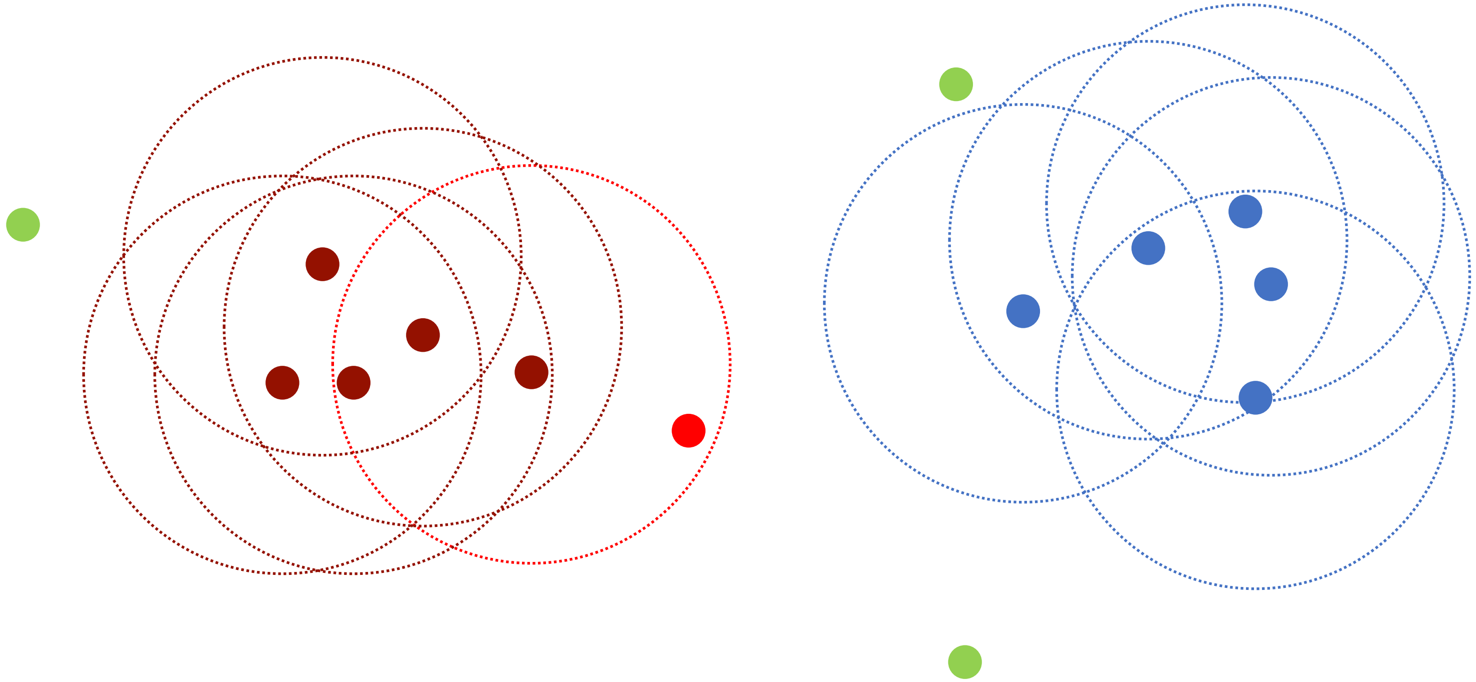
● **Border Points** are in same cluster as core points but further away from its center

● **Noise Points** are identified as not belonging to any cluster

The downside with this method is that the higher the dimension, the less accurate it becomes.

You also need to make a few assumptions like estimating the right value for eps which can be challenging.

Note that k-means and hierarchal clustering can also be used to detect outliers



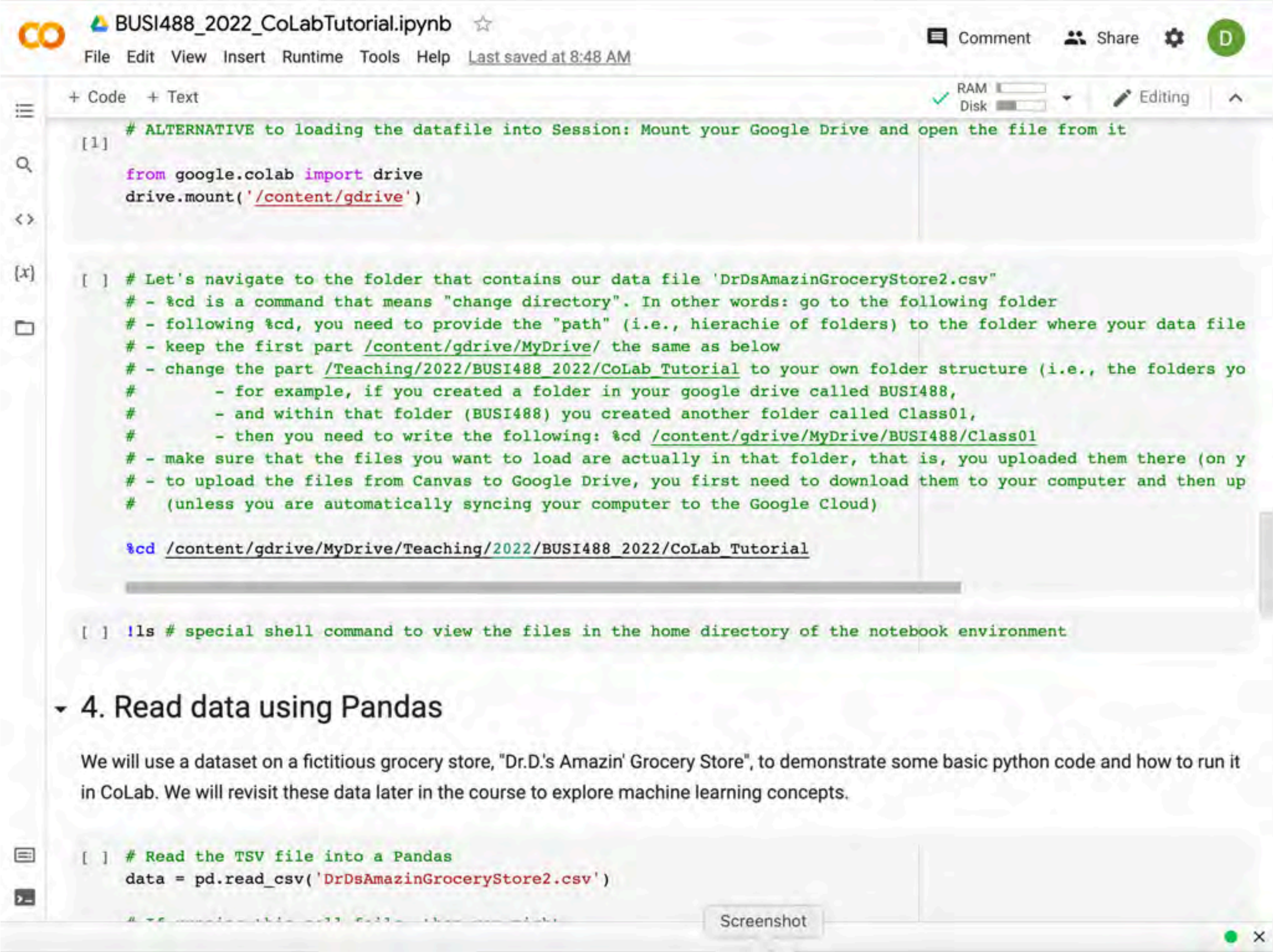
Working with Data in Cloud Computing (Google CoLab)

*Colaboratory, or "**Colab**" for short, **is** a product from **Google** Research. **CoLab** allows anybody to write and execute arbitrary python code through the browser, and **is** especially well suited to machine learning, data analysis and education.*

- **Create, open, edit, run, save python (jupyter) notebooks**
- **Upload Files and Datasets to Google Drive**
- **Export Data/Files to Google Drive**
- **Runtimes – use GPUs or even TPUs**
- **Clone Repositories from Github**

  Copy the clone link of the Github repository to CoLab
- **Terminal Commands**

  !pip install **library_name**
- **Collaborate: Share your notebook**


BUSI488_2022_CoLabTutorial.ipynb
File Edit View Insert Runtime Tools Help Last saved at 8:48 AM
Comment
Share
D
+ Code + Text
RAM
Disk
Editing
Screenshot

```
[1]  # ALTERNATIVE to loading the datafile into Session: Mount your Google Drive and open the file from it

     from google.colab import drive
     drive.mount('/content/gdrive')
```

```
[ ]  # Let's navigate to the folder that contains our data file 'DrDsAmazinGroceryStore2.csv"
     # - %cd is a command that means "change directory". In other words: go to the following folder
     # - following %cd, you need to provide the "path" (i.e., hierachie of folders) to the folder where your data file
     # - keep the first part /content/gdrive/MyDrive/ the same as below
     # - change the part /Teaching/2022/BUSI488_2022/CoLab_Tutorial to your own folder structure (i.e., the folders yo
     #       - for example, if you created a folder in your google drive called BUSI488,
     #       - and within that folder (BUSI488) you created another folder called Class01,
     #       - then you need to write the following: %cd /content/gdrive/MyDrive/BUSI488/Class01
     # - make sure that the files you want to load are actually in that folder, that is, you uploaded them there (on y
     # - to upload the files from Canvas to Google Drive, you first need to download them to your computer and then up
     #   (unless you are automatically syncing your computer to the Google Cloud)

     %cd /content/gdrive/MyDrive/Teaching/2022/BUSI488_2022/CoLab_Tutorial
```

```
[ ]  !ls # special shell command to view the files in the home directory of the notebook environment
```

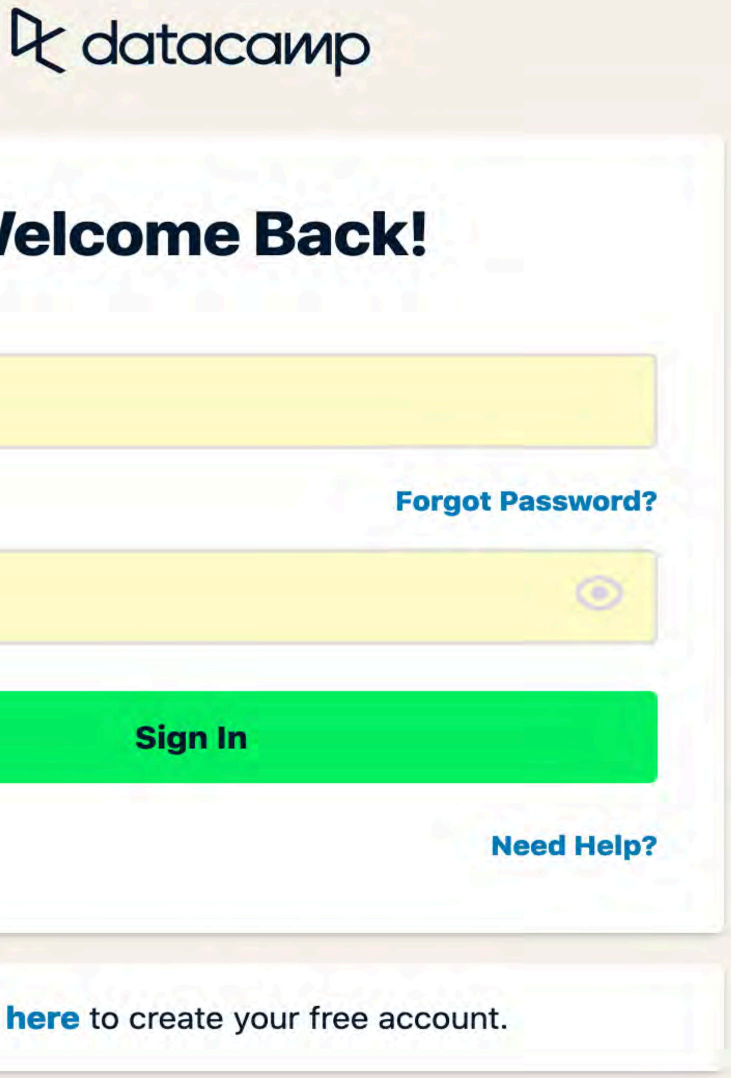
4. Read data using Pandas

We will use a dataset on a fictitious grocery store, "Dr.D.'s Amazin' Grocery Store", to demonstrate some basic python code and how to run it in CoLab. We will revisit these data later in the course to explore machine learning concepts.

```
[ ]  # Read the TSV file into a Pandas
     data = pd.read_csv('DrDsAmazinGroceryStore2.csv')
```

This course requires YOU to use CoLab

- Follow instructions on Canvas: Google CoLab
- Complete CoLab Tutorial before Class 02 (CANVAS>Files>CoLab_Tutorial)



Welcome Back!

E-mail address: dmr@unc.edu

Password: ••••••••••

□ Remember me

Or click here to create your free account.

Go to DataCamp:
www.datacamp.com

Instructions for DataCamp:
https://kenan-flagler.instructure.com/courses/3629908/pages/datacamp

Looking Ahead

Next Class: Tuesday, January 17th, 2023

Missing Data and Data Pitfalls

DataCamp Homework 1 due!

- Introduction to Python *(approximately 4 hours)*
- *Due on January 17th by 11:59pm*

Readings:

- none