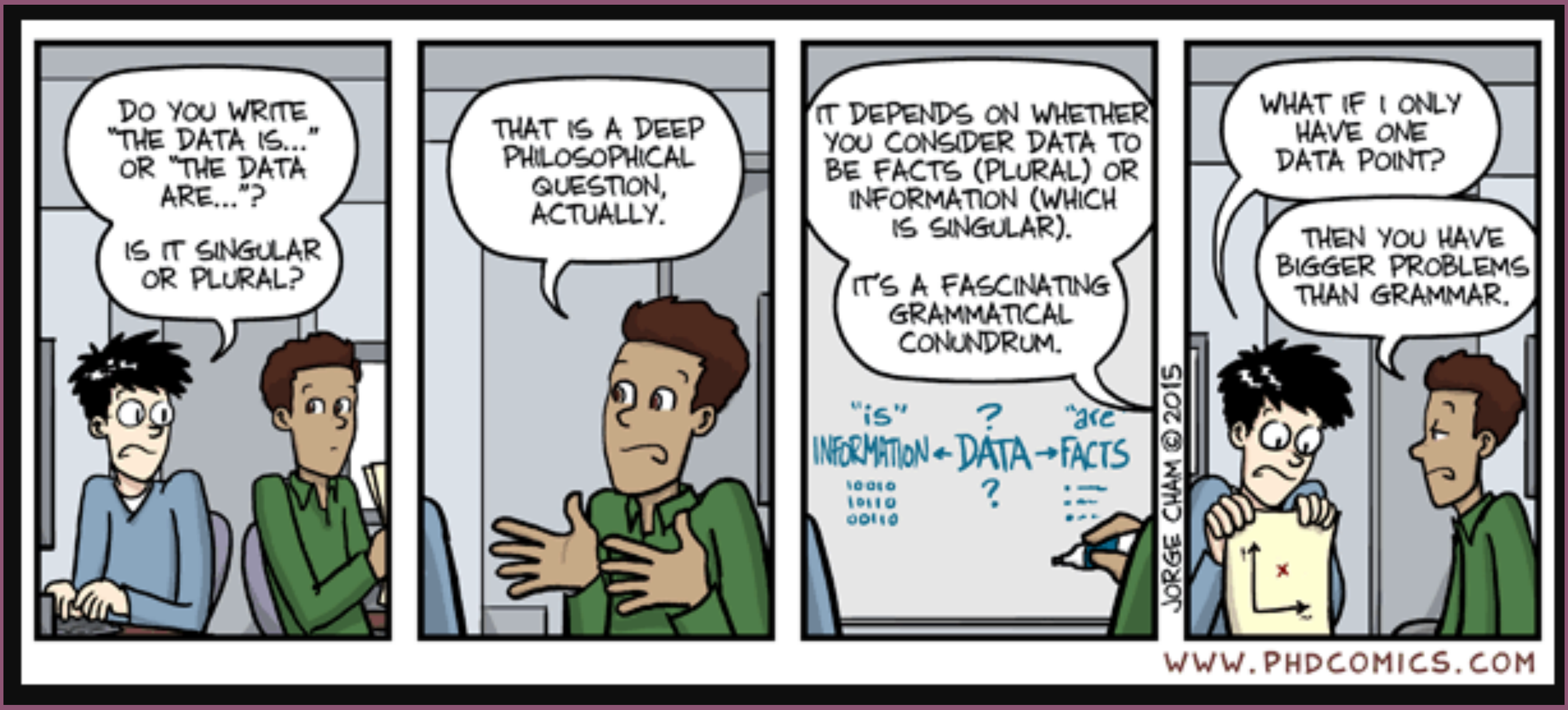
Introduction to Computation for the Humanities and Social Sciences

CS 3

Chris Tanner


DO YOU WRITE "THE DATA IS..." OR "THE DATA ARE..."?
IS IT SINGULAR OR PLURAL?
THAT IS A DEEP PHILOSOPHICAL QUESTION, ACTUALLY.
IT DEPENDS ON WHETHER YOU CONSIDER DATA TO BE FACTS (PLURAL) OR INFORMATION (WHICH IS SINGULAR).
IT'S A FASCINATING GRAMMATICAL CONUNDRUM.
"is"
?
"are"
INFORMATION ← DATA → FACTS
?
WHAT IF I ONLY HAVE ONE DATA POINT?
THEN YOU HAVE BIGGER PROBLEMS THAN GRAMMAR.
JORGE CHAM © 2015
WWW.PHDCOMICS.COM

Lecture 3

"I like Big Data and I cannot lie"

— Sir Mix-a-Lot

FIRST: SOME REMINDERS

1. The course website is at http://cs.brown.edu/courses/csci0030/ and has all assignments, resources, lectures, etc

2. Submit homework via Canvas (I will soon create the HW week 2 assignment to which you can submit)

3. Homework is due **Monday @ 11:59pm**

4. TAs (and myself) have office hours today:

    1. me from 1-3pm (CIT 527)

    2. Anna and Caroline from 5-7pm (CIT 203)

5. Further TA Hours are on Sunday and Monday.

6. I will issue override codes today

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

What is Data Collection?

- Data collection is the process of collecting information (ideally in a digestible format)

- Either you collect, process, and format the data yourself

- Or you can find a dataset someone else already produced

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Considerations When Choosing a Dataset

- What data is necessary to answer our question?
- How difficult is it to analyze a dataset?
- Comprehensive data vs sampled data?
- What is the allowed usage of data under its license?
- Who collected the data?
- When was the data collected?
- How was the data collected?
- How is the data formatted?
- Does your data collection procedures need to be approved by an IRB?
- Confidentiality Concerns

Considerations When Choosing a Dataset: Difficulty

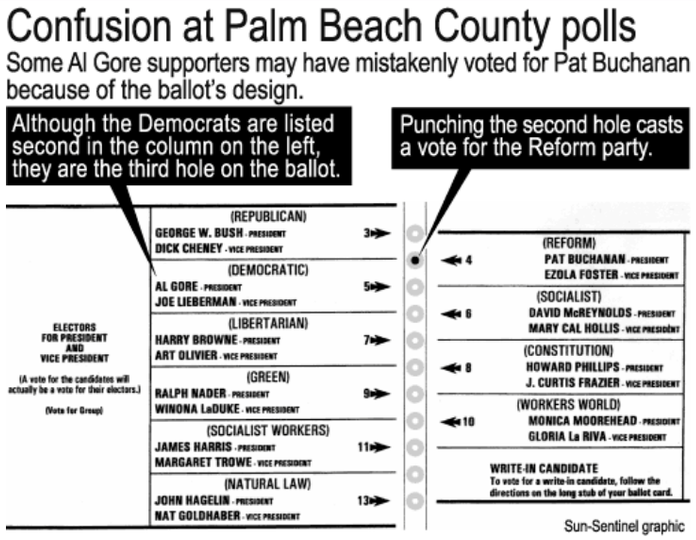
Easy for people

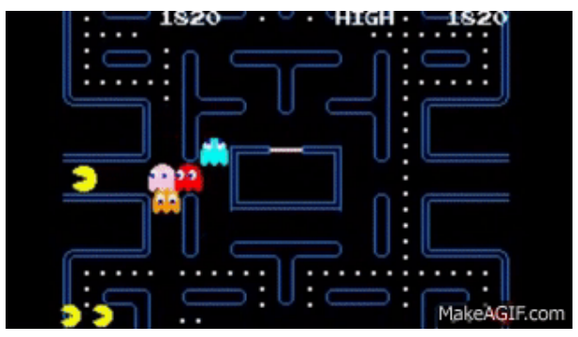
Difficult for people

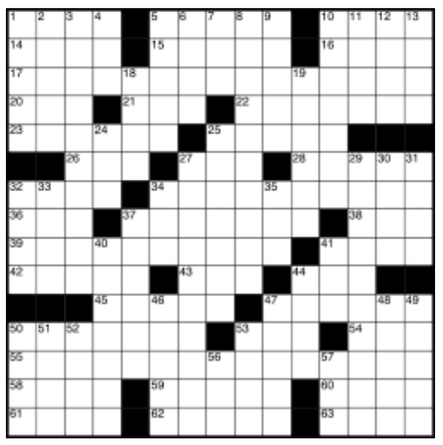
Easy for computers

Difficult for computers

	A	B	C
1	**name**	**age**	**height**
2	Michael	46	5'9"
3	Jim	31	6'0"
4	Pam	29	5'7"
5	Meredith	53	5'6"
6	Dwight	35	5'10"







Considerations When Choosing a Dataset: Difficulty

- We want data that's easy for the computer to analyze, but often the data we are interested in is easy for humans to analyze

Considerations When Choosing a Dataset: Difficulty

- We want data that's easy for the computer to analyze, but often the data we are interested in is easy for humans to analyze
- Often we can suitably answer our questions from data pulled from these items

Speech transcripts

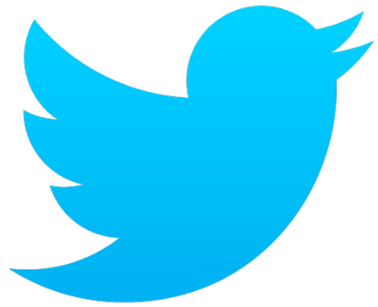
Scripts

Transcribed Lyrics

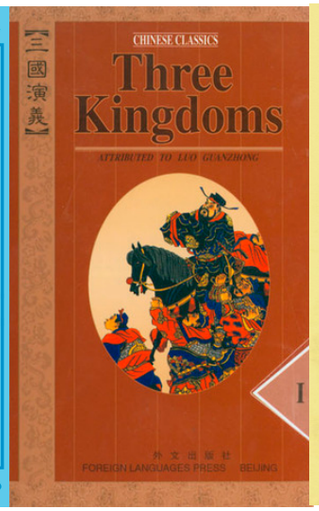
Considerations When Choosing a Dataset: Text

- Text data exists in many different forms

- Analyzing the text can provide insights into the intent and the language of the author



The New York Times



Considerations When Choosing a Dataset: Size

Senate votes per year	~15,000 votes	15KB
The Office (1 episode)	~3,000 words	13KB
Alice in Wonderland	~26,000 words	170KB
The English Bible	~800,000 words	4.2MB
Trump's Tweets	~800,000 words	4MB
Encyclopedia Britannica	~40M words	200MB
English Wikipedia	~2.9B words	14.5GB

Spreadsheets

Ideally suited for programming language

Analysis difficult

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Comprehensive Data

- We have access to all the data points that exist, which is usually a lot

- Collected and digitized as part of general procedures of an institution

The New York Times

13 million articles

~500 Million tweets per day

CONGRESS.GOV

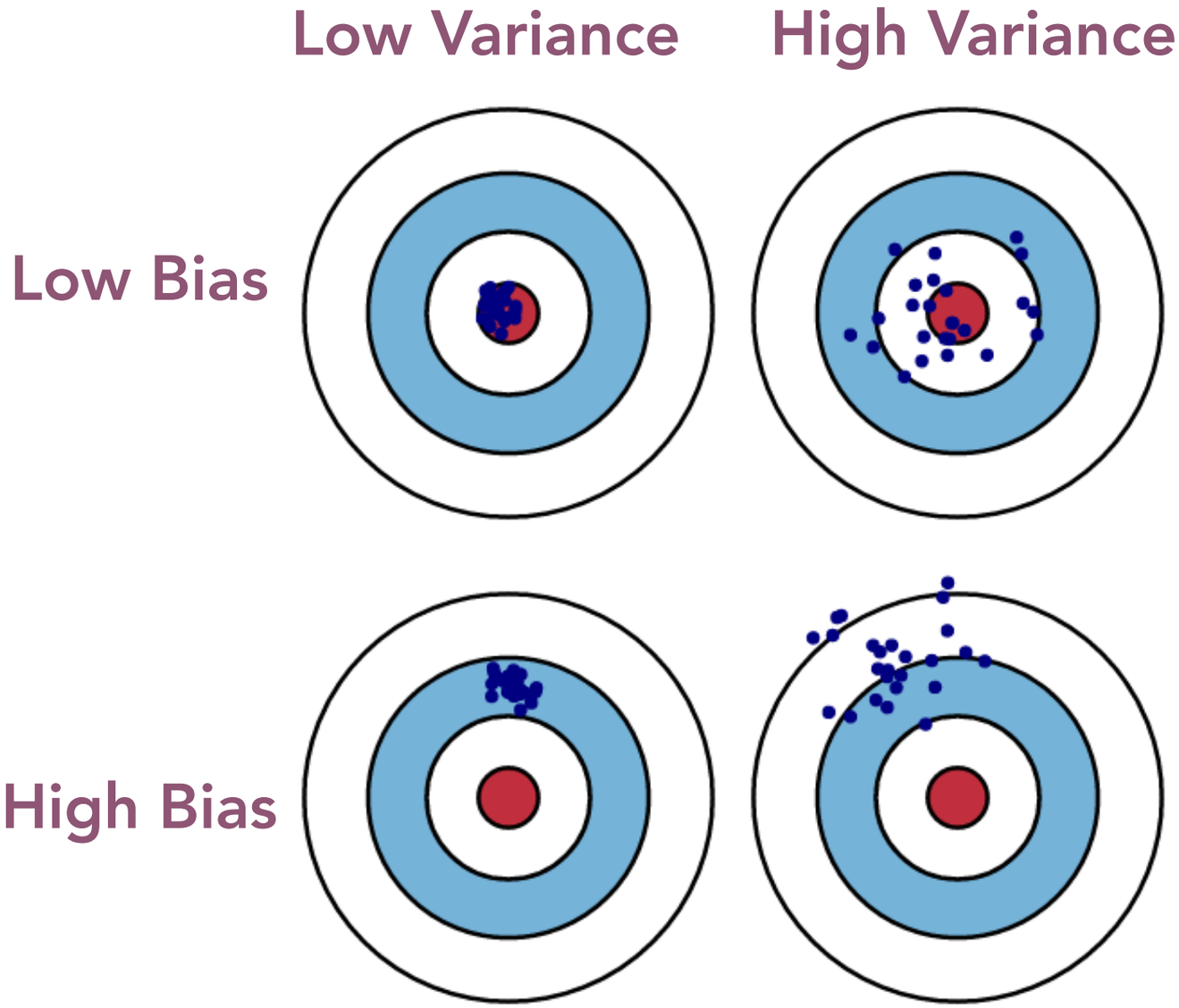
100,000s votes per year

Sampled Data

- When collecting individual data is relatively expensive
- Only a portion of the population is sampled
- Not just restricted to polling or surveys

Error vs Bias

- Error is the difference between the sampled population and the population as a whole
- Error can be systematic and non-systematic
- Systematic Error is also known as bias
- Non-systematic error is also referred to as the variance



Biases in Sampled Data

- **A bias in sampled data occurs when a procedure causes the sample to overrepresent a subpopulation**

- They may not necessarily be intentional

- Even if you don't think overrepresentation of a subpopulation will bias the dataset with regard to your question, it's still a bias

- Always strive to minimize any biases in your data collection procedures

Biases in Sampled Data

Gallup

- Randomly calls two groups of ~500 people a day by sampling among all possible phone numbers
- For landlines, asks for household member who has the next birthday
- Calls people living in all 50 states
- Tries to assure 70% cellphone, 30% landlines
- Weights data to reflect the demographics of the general population

Biases in Sampled Data

IMDb

- Registered users rate films and TV shows on a 1-10 star rating

- Registered users are an overrepresented subpopulation relative to the general population

- Registered users who rate movies in their free time further over represents a specific segment of the general population

- “Men Are Sabotaging The Online Reviews Of TV Shows Aimed At Women” - fivethirtyeight.com

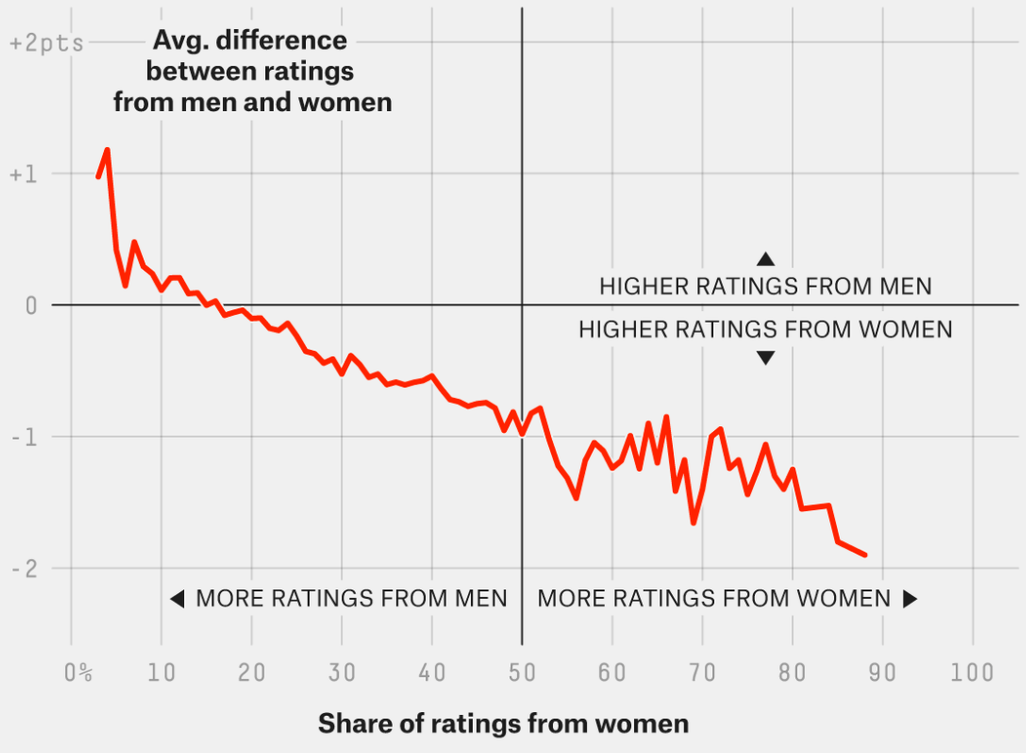
    - 60% who rated Sex in the City were women. Women gave it a 8.1, men gave it 5.8.

Biases in Sampled Data

IMDb

Men tank the ratings of shows aimed at women

Average difference between IMDb ratings of TV shows from men and women by share of ratings from women

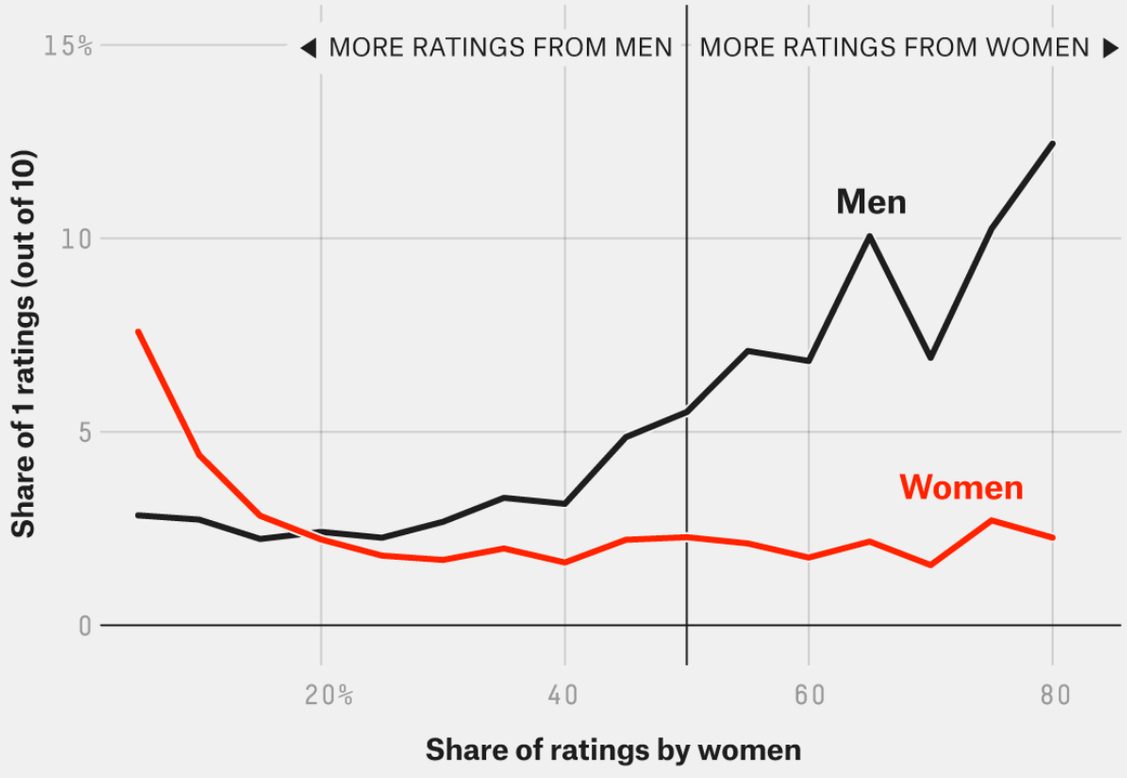


For English language shows with 1,000 or more ratings

FIVETHIRTYEIGHT BASED ON DATA FROM IMDB

Men are more likely to give the crappiest rating

Share of IMDb ratings of 1 (out of 10) for shows with at least 10,000 ratings by share of ratings from women*



*Rounded to nearest 5 percent

FIVETHIRTYEIGHT BASED ON DATA FROM IMDB

Biases in Sampled Data

Yelp

- Registered users rate businesses on a 1-5 star scale
- Registered users tend to represent a certain subset of the population (those who are more social media inclined and opinionated)
- Customers with extreme experiences are more likely to voice their opinions

Biases in Sampled Data

Yelp

3. CVS Pharmacy

12 reviews

$$ · Drugstores

College Hill

291 Thayer St
Providence, RI 02906
(401) 331-1970

The location of this CVS right on Thayer St is very convenient for shoppers, diners and students. You might be needing something from the drugstore after having a very large meal,… read more

4. CVS Pharmacy

16 reviews

Pharmacy

481 Angell St
Providence, RI 02906
(401) 521-4340

This is the best pharmacy in Providence , where managers Alex and Nicole extremely caring and helpful for customers.. Alex always takes time and explains in details everything we… read more

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Data Formats

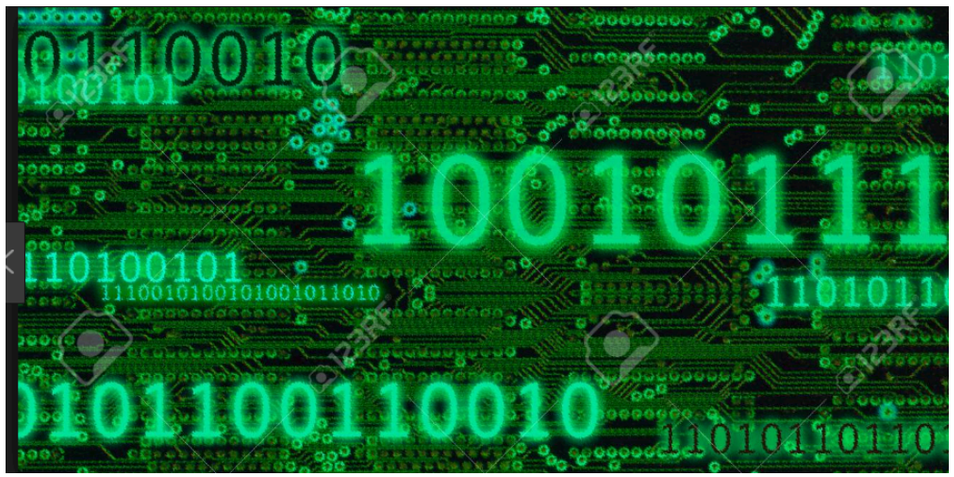
Bits and Bytes

- Data is stored to the computer in **binary**

- The smallest unit in binary is a **bit** (**bi**nary + digi**t**)

- Every bit is a 1 or 0

- 8 consecutive bits are known as a **byte:**



10111010 01001110

byte **byte**

Data Formats

Bits and Bytes

- Data is stored to the computer in **binary**

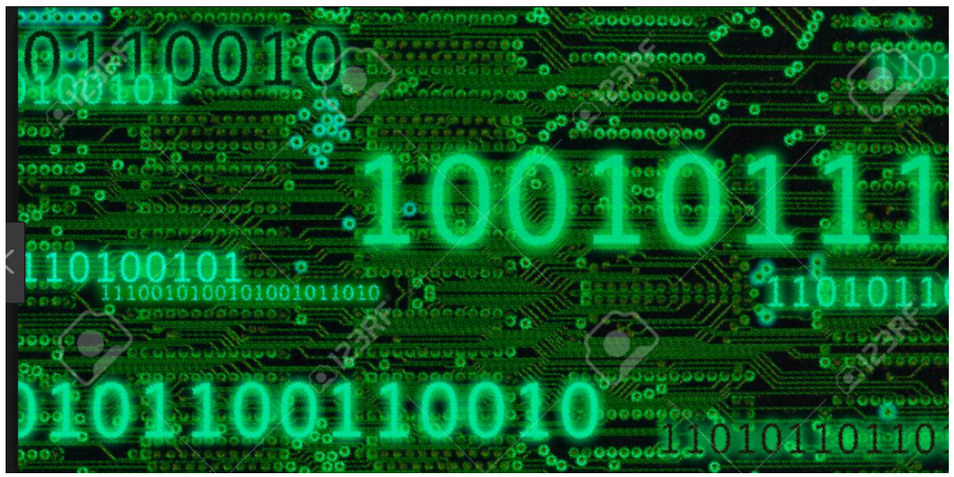
- The smallest unit in binary is a **bit** (**bi**nary + digi**t**)

- Every bit is a 1 or 0

- 8 consecutive bits are known as a **byte:**

10111010 01001110

byte **byte**



Binary vs Plain Text

- **All data stored to the computer is fundamentally represented in binary**

- Data stored to a file that can be completely interpreted as unformatted text is called **plain text**

- Binary data is simply data that cannot be interpreted as plain text

- Generally speaking, if your data file needs to be opened by a specific program, it is probably stored as binary data

- If you can read your data, as is, on *any* text editor, it is likely plain text

Data Formats

Plain Text Examples

- Plain text, what people really mean when they say plain text
- Ends in .txt (generally)
- No formatting, font type, font size, color, etc.
- Text position is provided by whitespace characters (space, tab, return)

```
ALICE'S ADVENTURES IN WONDERLAND

Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0



CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of
sitting by her sister on the bank, and of
having nothing to do: once or twice she
had peeped into the book her sister was
reading, but it had no pictures or
conversations in it, 'and what is the use
of a book,' thought Alice 'without
pictures or conversations?'
```

Plain Text Examples

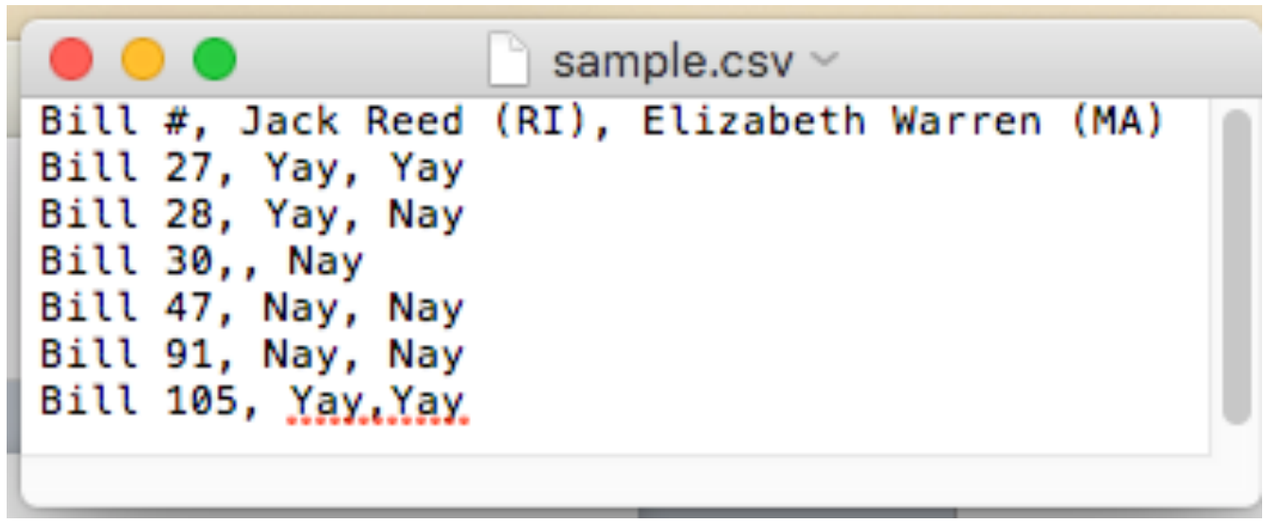
- XML (.xml)
- These colors ——> aren't actually stored in the file, the editor just adds them on your screen to help make it look prettier

```
<roll_call_vote>
  <congress>115</congress>
  <session>1</session>
  …
  <members>
    <member>
      <member_full>Alexander (R-TN)</
member_full>
      <last_name>Alexander</last_name>
      <first_name>Lamar</first_name>
      <party>R</party>
      <state>TN</state>
      <vote_cast>Yea</vote_cast>
      …
    </member>
  </members>
</roll_call_vote>
```

Data Formats

Plain Text Examples

- CSV (.csv)
- Tab-separated (.tsv)
- **Delimiter**: The character that separates each value



Data Formats

Plain Text Examples

- JSON (.json)
- **J**ava**S**cript **O**bject **N**otation
- Like XML, data is annotated
- A nesting of key-value pairs
- When whitespace is removed, can be more space efficient than XML

```
{
  "firstName": "John",
  "lastName": "Smith",
  "isAlive": true,
  "age": 27,
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021-3100"
  },
  "phoneNumbers": [
    {
      "type": "home",
      "number": "212 555-1234"
    },
    {
      "type": "office",
      "number": "646 555-4567"
    },
    {
      "type": "mobile",
      "number": "123 456-7890"
    }
  ],
  "children": [],
  "spouse": null
}
```

Data Formats

Plain Text Examples

- YAML (.yaml)
- **Y**AML **A**in't **M**arkup **L**anguage
- Annotated similarly to JSON
- Programs that can read YAML can read JSON

```
invoice: 34843
date   : 2001-01-23
bill-to: &id001
    given  : Chris
    family : Dumars
address:
    lines: |
        458 Walkman Dr.
        Suite #292
    city    : Royal Oak
    state   : MI
```

Data Formats

Plain Text vs XML vs JSON vs YAML

- They can all express the same content
- Plain Text doesn't have structure, but is universally robust
- XML is the most verbose, harder to parse
- JSON doesn't have </stuff_here> end tags
- JSON is more succinct than XML (easier to parse)
- YAML is a tad more robust than JSON, and is even more human-readable

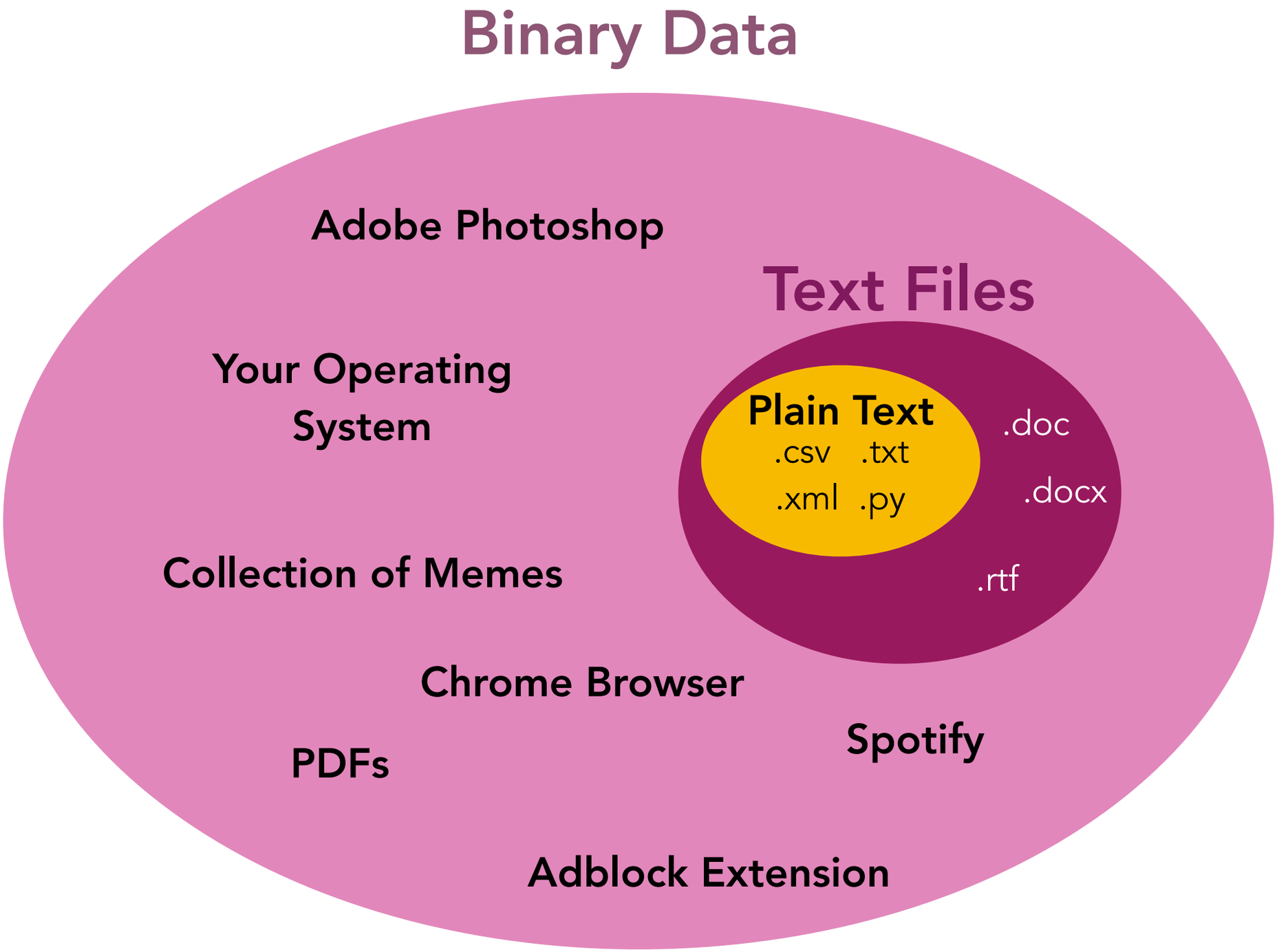
File Extensions

- File extensions are the characters that follow the '.' in a filename (hello.**txt**)

- Sometimes your computer will only allow programs to open files with a specific file extension

- Even if many programs can read a plain text data file, file extensions are used to communicate the type of data inside (.xml, .csv, .yaml, .json)

Graph of Everything on your Computer



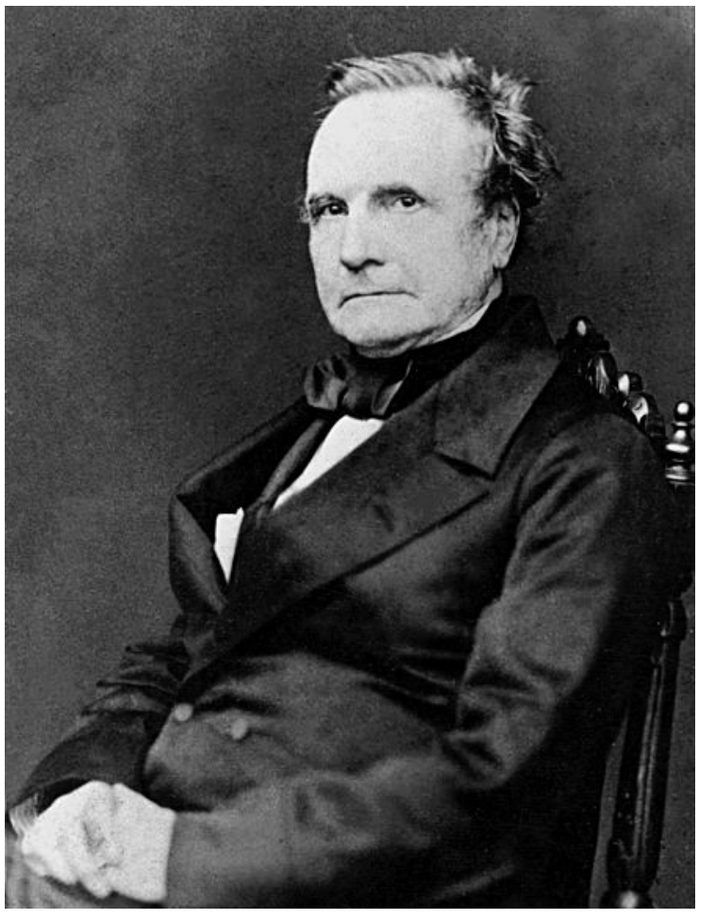
Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Old School



Charles Babbage

- **1837** — Invented the digital programmable computer the "Analytical Engine".
- Included:
  - arithmetic logic unit
  - control flow (which lines of code to execute when)
  - conditional branching (e.g., "if blah then do blah")
  - loops (keep doing some code for a certain # of times)
- He never actually built it
  (funding issues + co-worker drama)

Old School

Ada Lovelace

- **1843** — Wrote the first computer program ever!
- It was to compute Bernoulli Numbers on Babbage's *Analytical Engine*
- She was precocious:

> **[The Analytical Engine] might act upon other things besides *number*, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations,** and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine...Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data. ${}^1V_1$ ○ 0 0 1 [1]	Data. ${}^1V_2$ ○ 0 0 2 [2]	Data. ${}^1V_3$ ○ 0 0 4 [n]	Working Variables. ${}^0V_4$ ○ 0 0 0 [ ]	${}^0V_5$ ○ 0 0 0 [ ]	${}^0V_6$ ○ 0 0 0 [ ]	${}^0V_7$ ○ 0 0 0 [ ]	${}^0V_8$ ○ 0 0 0 [ ]	${}^0V_9$ ○ 0 0 0 [ ]	${}^0V_{10}$ ○ 0 0 0 [ ]	${}^0V_{11}$ ○ 0 0 0 [ ]	${}^0V_{12}$ ○ 0 0 0 [ ]	${}^0V_{12}$ ..... ○ 0 0 0 [ ]	Result Variables. ${}^1V_{21}$ B_1 in a decimal ○ fraction. [B_1]	${}^1V_{22}$ B_3 in a decimal ○ fraction. [B_3]	${}^1V_{23}$ B_5 in a decimal ○ fraction. [B_5]	${}^0V_{24}$... ○ 0 0 0 [B_7]
1	×	${}^1V_2 \times {}^1V_3$	${}^1V_4, {}^1V_5, {}^1V_6$	${}^1V_2 = {}^1V_2$, ${}^1V_3 = {}^1V_3$	$= 2n$	...	2	n	$2n$	$2n$	$2n$											
2	−	${}^1V_4 - {}^1V_1$	${}^2V_4$	${}^1V_4 = {}^2V_4$, ${}^1V_1 = {}^1V_1$	$= 2n-1$	1	...	...	$2n-1$													
3	+	${}^1V_5 + {}^1V_1$	${}^2V_5$	${}^1V_5 = {}^2V_5$, ${}^1V_1 = {}^1V_1$	$= 2n+1$	1	...	...	...	$2n+1$												
4	÷	${}^2V_5 \div {}^2V_4$	${}^1V_{11}$	${}^2V_5 = {}^0V_5$, ${}^2V_4 = {}^0V_4$	$= \frac{2n-1}{2n+1}$	...	...	...	0	0	...	...	...	...	...	$\frac{2n-1}{2n+1}$						
5	÷	${}^1V_{11} \div {}^1V_2$	${}^2V_{11}$	${}^1V_{11} = {}^2V_{11}$, ${}^1V_2 = {}^1V_2$	$= \frac{1}{2} \cdot \frac{2n-1}{2n+1}$	...	2	...	...	...	...	...	...	...	...	$\frac{1}{2} \cdot \frac{2n-1}{2n+1}$						
6	−	${}^0V_{13} - {}^2V_{11}$	${}^1V_{13}$	${}^2V_{11} = {}^0V_{11}$, ${}^0V_{13} = {}^1V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$	...	...	...	...	...	...	...	...	...	...	0	.....	$-\frac{1}{2} \cdot \frac{2n-1}{2n+1} = A_0$				
7	−	${}^1V_3 - {}^1V_1$	${}^1V_{10}$	${}^1V_3 = {}^1V_3$, ${}^1V_1 = {}^1V_1$	$= n-1\ (=3)$	1	...	n	...	...	...	...	...	...	$n-1$							
8	+	${}^1V_2 + {}^0V_7$	${}^1V_7$	${}^1V_2 = {}^1V_2$, ${}^0V_7 = {}^1V_7$	$= 2 + 0 = 2$	...	2	...	...	...	...	2										
9	÷	${}^1V_6 \div {}^1V_7$	${}^3V_{11}$	${}^1V_6 = {}^1V_6$, ${}^0V_{11} = {}^3V_{11}$	$= \frac{2n}{2} = A_1$	...	...	...	...	...	$2n$	2	...	...	...	$\frac{2n}{2} = A_1$						
10	×	${}^1V_{21} \times {}^3V_{11}$	${}^1V_{12}$	${}^1V_{21} = {}^1V_{21}$, ${}^3V_{11} = {}^3V_{11}$	$= B_1 \cdot \frac{2n}{2} = B_1 A_1$	...	...	...	...	...	...	...	...	...	...	$\frac{2n}{2} = A_1$	$B_1 \cdot \frac{2n}{2} = B_1 A_1$	.....	B_1			
11	+	${}^1V_{12} + {}^1V_{13}$	${}^2V_{13}$	${}^1V_{12} = {}^0V_{12}$, ${}^1V_{13} = {}^2V_{13}$	$= -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2}$ ...	...	...	...	...	...	...	...	...	...	...	.....	0	$\left\{ -\frac{1}{2} \cdot \frac{2n-1}{2n+1} + B_1 \cdot \frac{2n}{2} \right\}$				
12	−	${}^1V_{10} - {}^1V_1$	${}^2V_{10}$	${}^1V_{10} = {}^2V_{10}$, ${}^1V_1 = {}^1V_1$	$= n-2\ (=2)$	1	...	...	...	...	...	...	...	...	$n-2$							
13	−	${}^1V_6 - {}^1V_1$	${}^2V_6$	${}^1V_6 = {}^2V_6$, ${}^1V_1 = {}^1V_1$	$= 2n-1$	1	...	...	...	...	$2n-1$											
14	+	${}^1V_1 + {}^1V_7$	${}^2V_7$	${}^1V_1 = {}^1V_1$, ${}^1V_7 = {}^2V_7$	$= 2 + 1 = 3$	1	...	...	...	...	...	3										
15	÷	${}^2V_6 \div {}^2V_7$	${}^1V_8$	${}^2V_6 = {}^2V_6$, ${}^2V_7 = {}^2V_7$	$= \frac{2n-1}{3}$	...	...	...	...	...	$2n-1$	3	$\frac{2n-1}{3}$									
16	×	${}^1V_8 \times {}^3V_{11}$	${}^4V_{11}$	${}^1V_8 = {}^0V_8$, ${}^3V_{11} = {}^4V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3}$	...	...	...	...	...	...	...	0	...	...	$\frac{2n}{2} \cdot \frac{2n-1}{3}$						
17	−	${}^2V_6 - {}^1V_1$	${}^3V_6$	${}^2V_6 = {}^3V_6$, ${}^1V_1 = {}^1V_1$	$= 2n-2$	1	...	...	...	...	$2n-2$											
18	+	${}^1V_1 + {}^2V_7$	${}^3V_7$	${}^2V_7 = {}^3V_7$, ${}^1V_1 = {}^1V_1$	$= 3 + 1 = 4$	1	...	...	...	...	...	4										
19	÷	${}^3V_6 \div {}^3V_7$	${}^1V_9$	${}^3V_6 = {}^3V_6$, ${}^3V_7 = {}^3V_7$	$= \frac{2n-2}{4}$	...	...	...	...	...	$2n-2$	4	...	$\frac{2n-2}{4}$	...	$\left\{ \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = A_3 \right\}$						
20	×	${}^1V_9 \times {}^4V_{11}$	${}^5V_{11}$	${}^1V_9 = {}^0V_9$, ${}^4V_{11} = {}^5V_{11}$	$= \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{4} = A_3$	...	...	...	...	...	...	...	...	0								
21	×	${}^1V_{22} \times {}^5V_{11}$	${}^0V_{12}$	${}^1V_{22} = {}^1V_{22}$, ${}^0V_{12} = {}^2V_{12}$	$= B_3 \cdot \frac{2n}{2} \cdot \frac{2n-1}{3} \cdot \frac{2n-2}{3} = B_3 A_3$	...	...	...	...	...	...	...	...	...	...	0	$B_3 A_3$	.....	.....	B_3		
22	+	${}^2V_{12} + {}^2V_{13}$	${}^3V_{13}$	${}^2V_{12} = {}^0V_{12}$, ${}^2V_{13} = {}^3V_{13}$	$= A_0 + B_1 A_1 + B_3 A_3$	...	...	...	...	...	...	...	...	...	...	.....	0	$\left\{ A_3 + B_1 A_1 + B_3 A_3 \right\}$				
23	−	${}^2V_{10} - {}^1V_1$	${}^3V_{10}$	${}^2V_{10} = {}^3V_{10}$, ${}^1V_1 = {}^1V_1$	$= n-3\ (=1)$	1	...	...	...	...	...	...	...	...	$n-3$							

Here follows a repetition of Operations thirteen to twenty-three.

24	+	${}^4V_{13} + {}^0V_{24}$	${}^1V_{24}$	${}^4V_{13} = {}^0V_{13}$, ${}^0V_{24} = {}^1V_{24}$	$= B_7$	...	...	...	...	...	...	...	...	...	...	.....	.....	.....	.....	.....	.....	B_7
25	+	${}^1V_1 + {}^1V_3$	${}^1V_3$	${}^1V_1 = {}^1V_1$, ${}^1V_3 = {}^1V_3$, ${}^5V_6 = {}^0V_6$, ${}^5V_7 = {}^0V_7$	$= n+1 = 4+1 = 5$ ..... by a Variable-card. by a Variable card.	1	...	$n+1$	...	...	0	0										

Definition

- A **programming language** is a formal language to describe tasks to a machine or computer

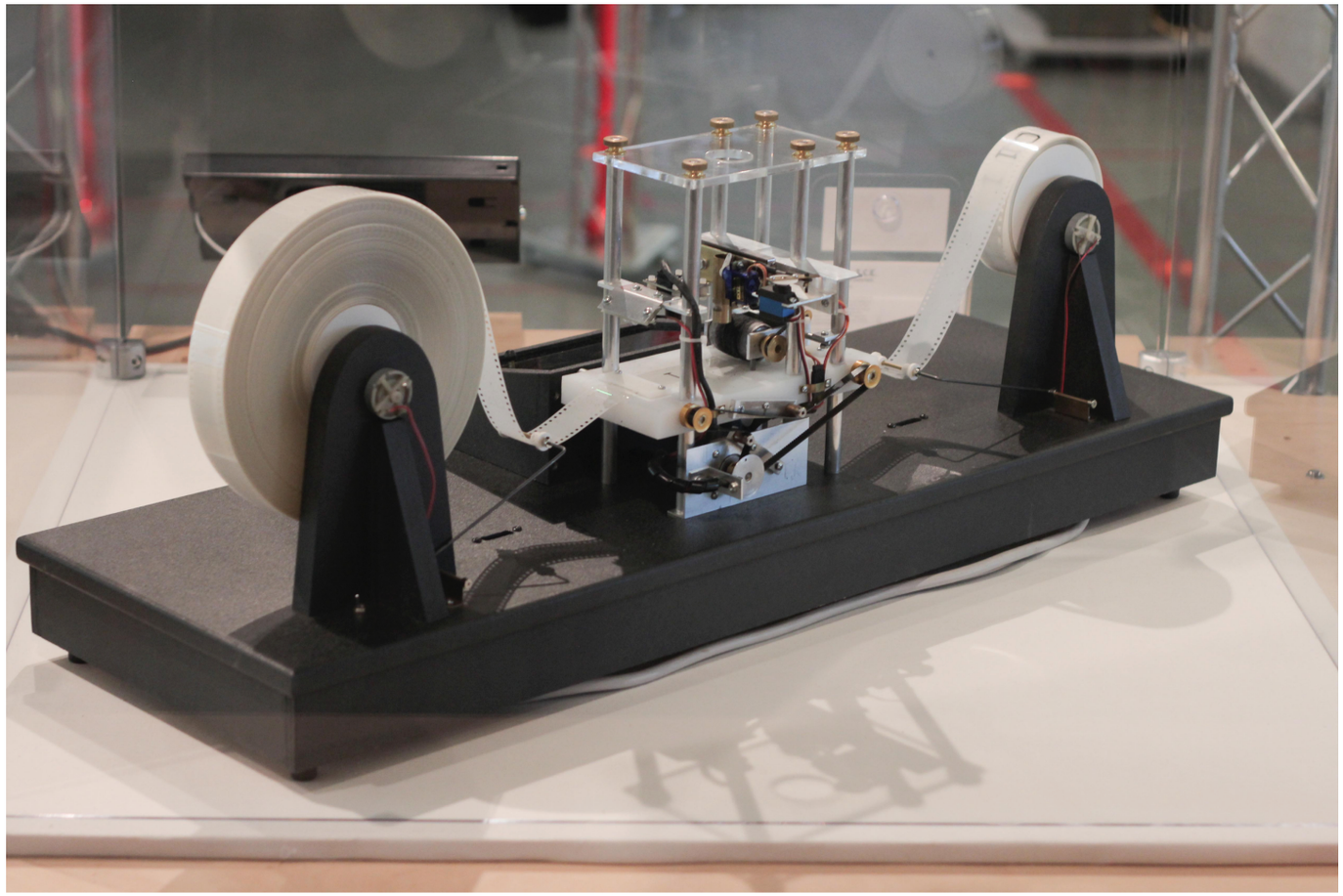
- A set of instructions used to control the behavior of a computer/machine is called a **program**

Alan Turing's Universal Turing Machine (1936)

- A mathematical description of a general purpose programmable computer

- Foundational for modern analytical methods in computer science

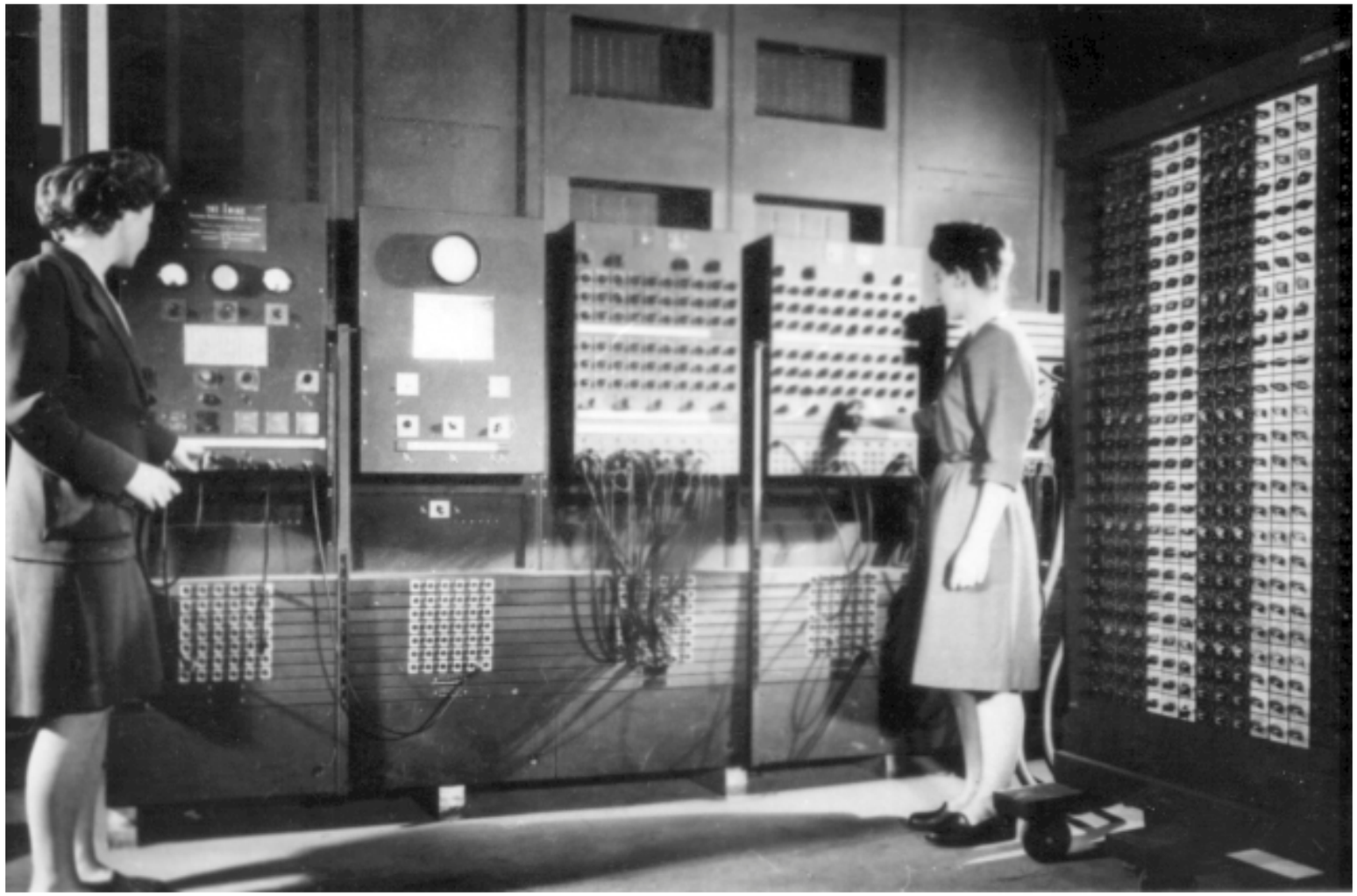
- Programmer writes instructions, and the execution is done via the tape moving and writing symbols to the tape.



History of Programming Languages

ENIAC (1946)

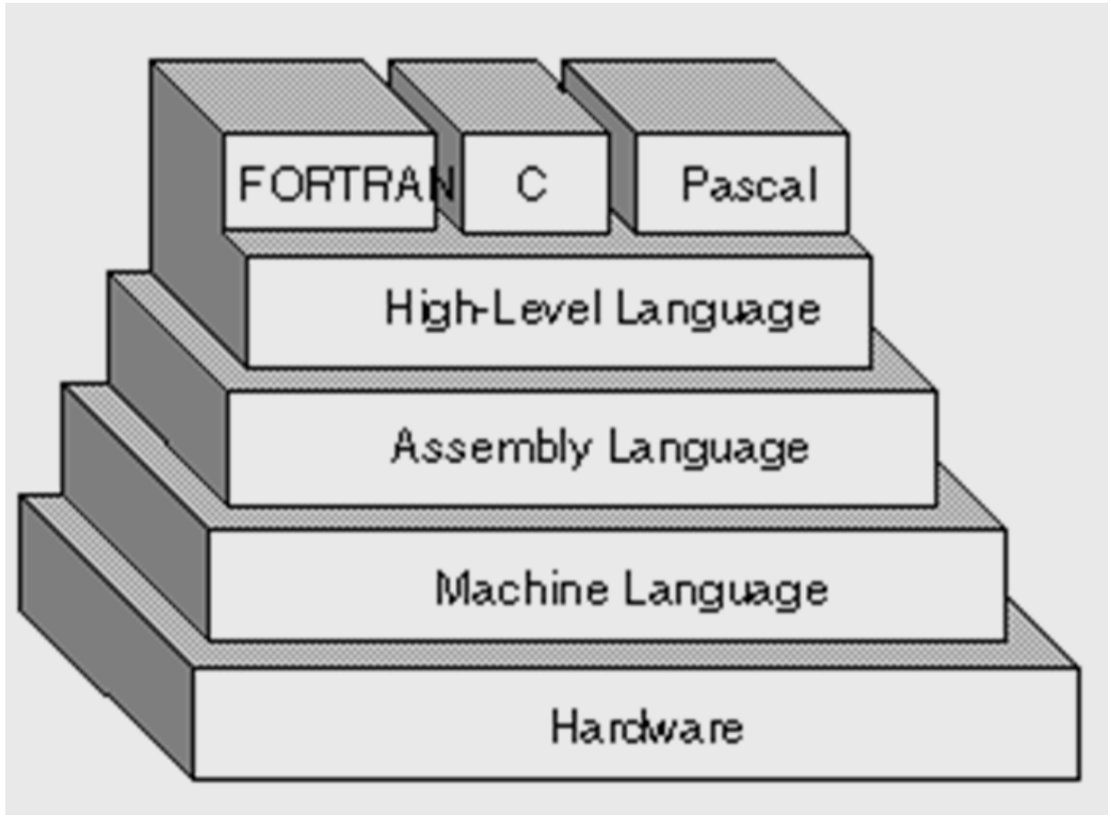
- One of first electronic general-purpose computers
- Used primarily for ballistic research
- Had 10,000 capacitors (stores data)
- First programmers were all female: Kay McNulty, Betty Jennings, Betty Snyder, Marlyn Meltzer, Fran Bilas, Ruth Lichterman



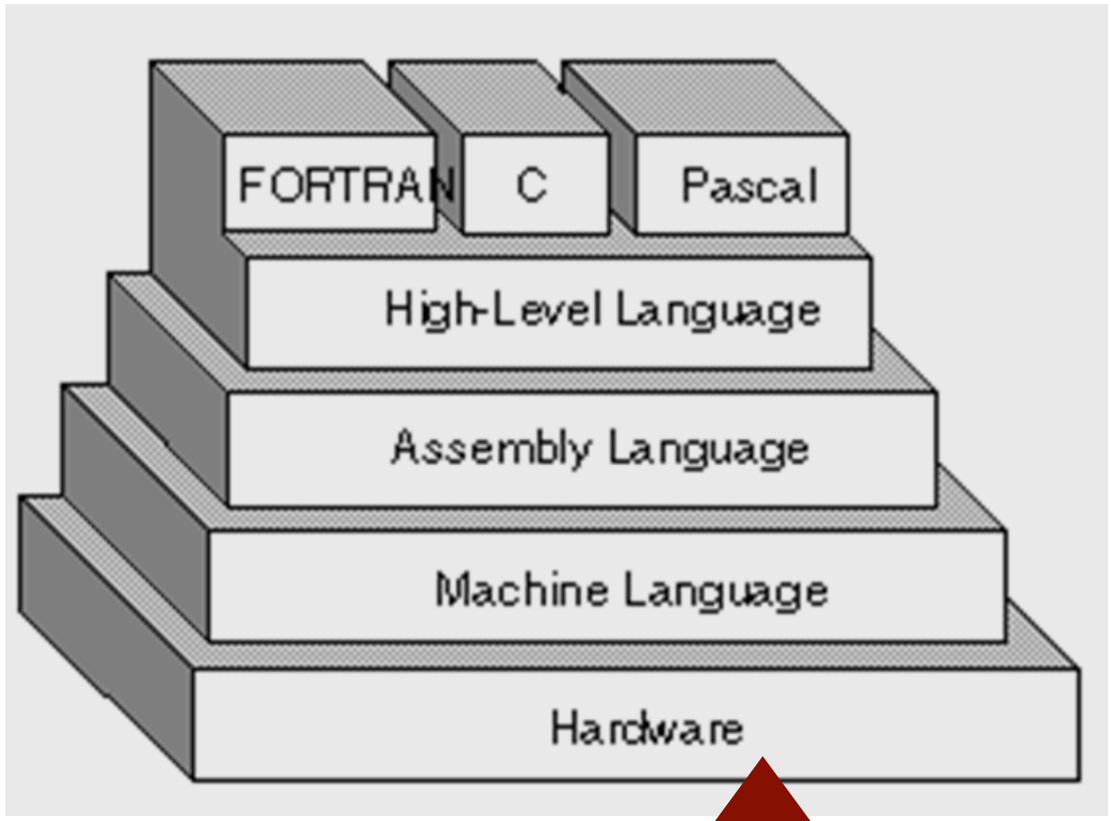
Lecture 3

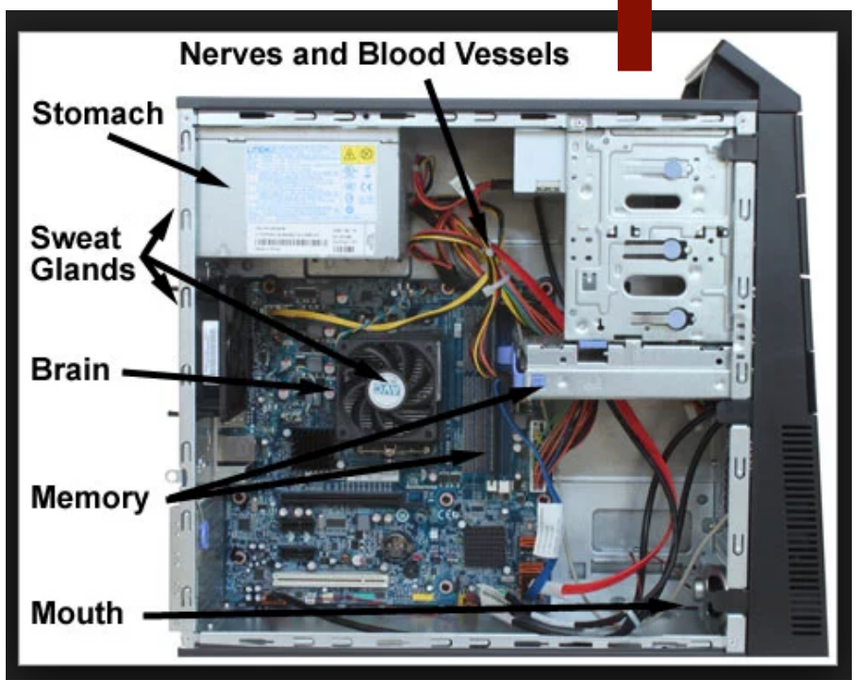
- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Types of Programming Languages

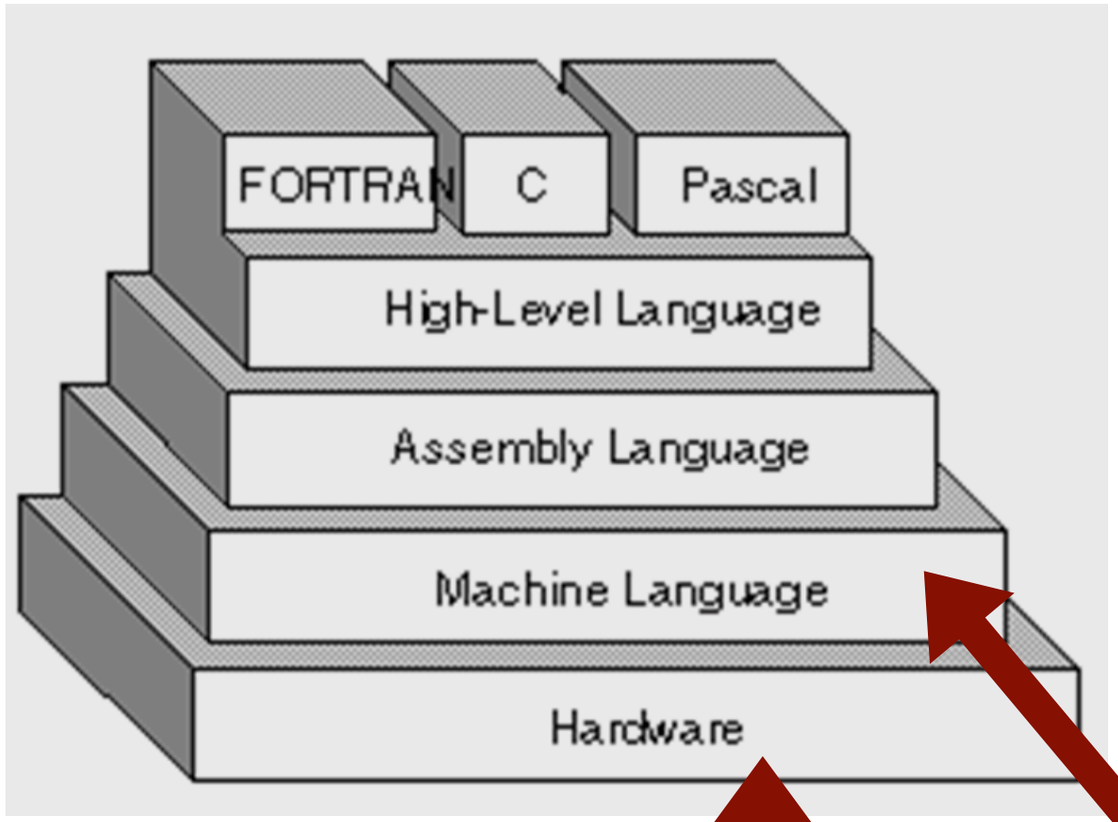


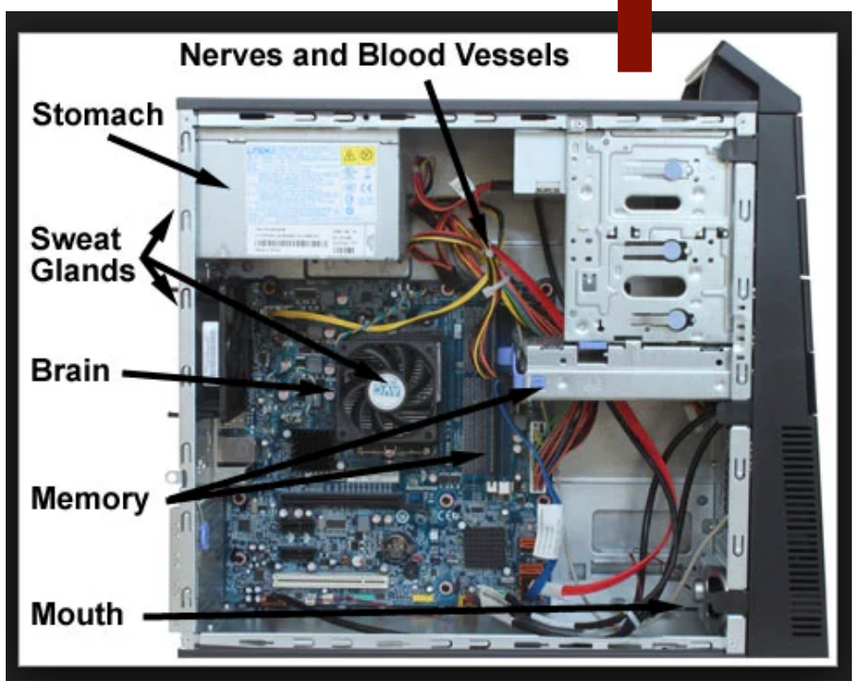
Types of Programming Languages





Types of Programming Languages

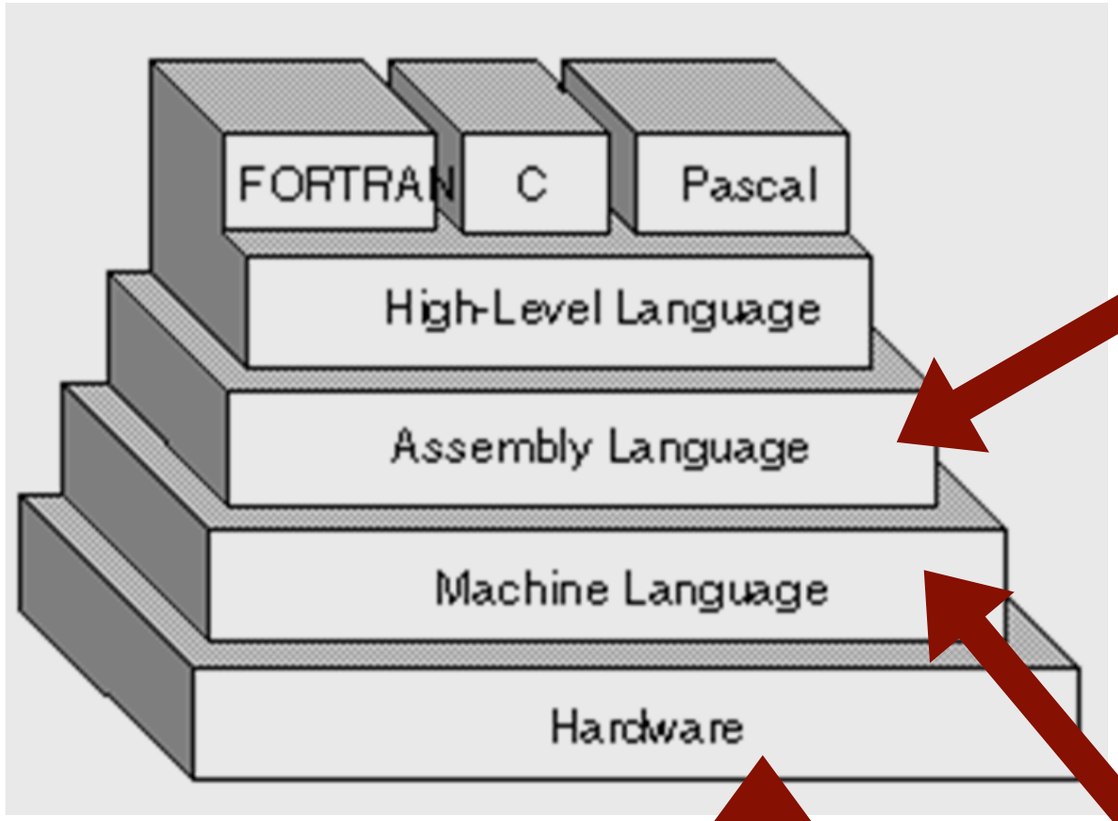




Machine Language (aka Machine Code):
The only code a machine can understand.
Nobody can program in this language.

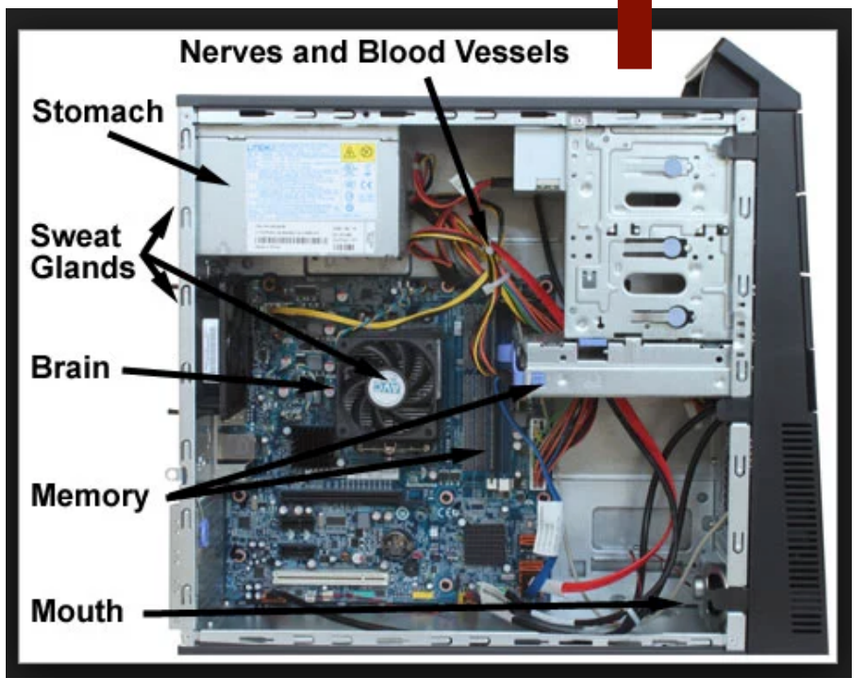
10101010010101011010101
01011010101010100010101
10111011111011010101010

Types of Programming Languages



Assembly Language:
Lowest level language to program in;
code is translated into Machine Language.

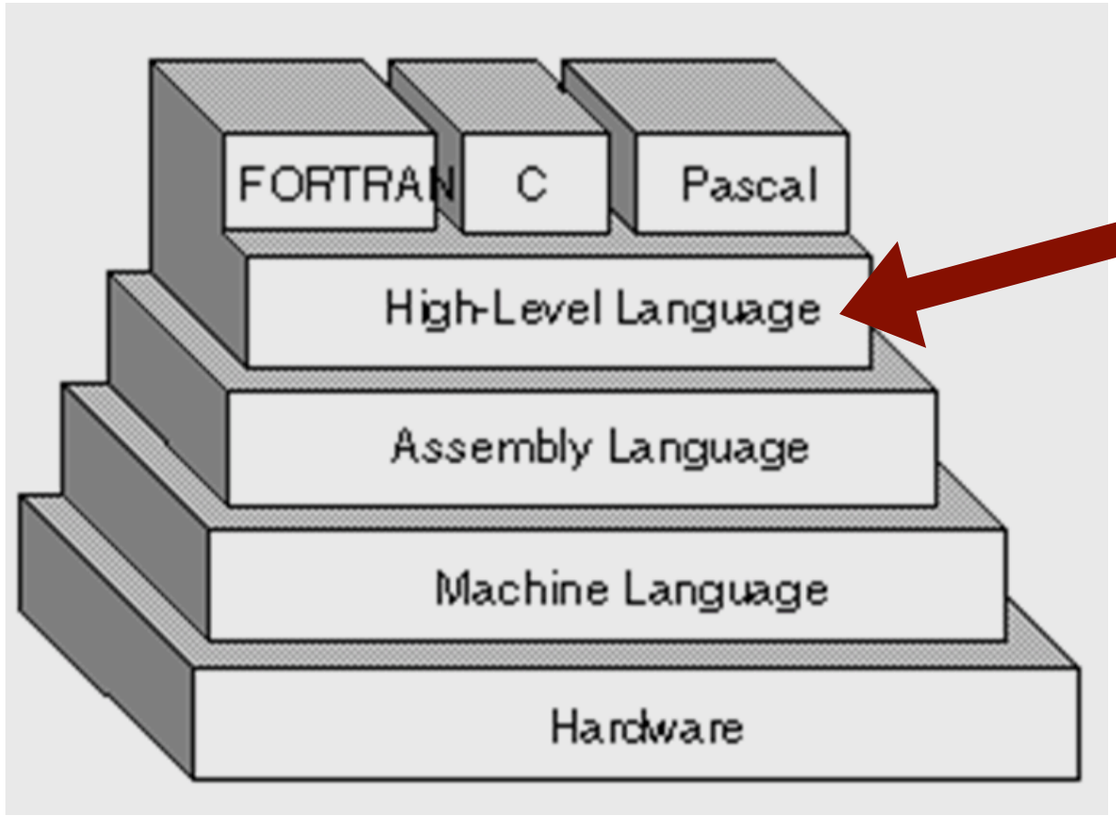
```
mov ecx, 16
mov ebp, ecx
sub esp, 4
push edi
push ecx
add [ebp-4], edi
add eax, [ebp-4]
```



Machine Language (aka Machine Code):
The only code a machine can understand.
Nobody can program in this language.

```
10101010010101011010101
01011010101010100010101
10111011111011010101010
```

Types of Programming Languages



High-Level Language:
(e.g., Python, FORTRAN, C, etc)
Allows humans to easily write and read code which the computer will automatically translate to lower-level code and execute.

```
1  import math
2
3  def is_palindrome(word):
4      for i in range(len(word)/2):
5          if word[i] != word[-(1+i)]:
6              return False
7      return True
```

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Lecture 3

- What is Data Collection?
- Considerations When Choosing a Dataset
- Comprehensive vs Sampled Data
- Data Formats
- History of Programming Languages
- Types of Programming Languages
- Python
- Pseudocode

Python

- Python is a **high-level** programming language: single lines of Python code generally specify more than one instruction to the computer

- This means that your Python code is **compiled** to **low-level** instructions called **byte-code** before your code is actually run

- The byte-code is then **interpreted** line-by-line into instructions the computer understands (*machine code*) and executed simultaneously

- Python is **general purpose**, and is not particularly specialized to a single task, unlike a language like R, Matlab or Stata

- Python is **open-source**, and a large community supports the code

Python

- You will be writing programs in the Python language
- Often times, these programs will require code written by others known as **libraries**
- In python, a library is a collection of **modules**, and your program will reference the individual modules.

Lecture 3: Last Words

Next lecture, we dive into Python and start to build a programming foundation

Lab Time!