COGS138: Neural Data Science

Lecture 8

C. Alex Simpkins, PhD
UCSD Dept. of Cognitive Science, Spring 2023
RDPRobotics, LLC
http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23

rdprobotics@gmail.com | csimpkinsjr@ucsd.edu

Plan for today

- Announcements
- Assignment 2 overview
- Review - Last time
- More on Data, Visualization, and outlier detection
- Group projects introduction

Announcements

- A1 - not everybody turned it in, if you have not please do and email me!
- A2 - due **a week from release, on data hub**
- Reading 2 - Released on canvas and in web site password protected area soon, lecture quiz due **a week from release**
- **Group formation** - check canvas for empty groups, please self-sign up
- Previous project review released tonight

Last time

Course links

Website	http://casimpkinsjr.radiantdolphinpress.com/pages/cogs138_sp23	Main face of the course and everything will be linked from here. Lectures, Readings, Handouts, Files, links
GitHub	https://github.com/drsimpkins-teaching	files/data, additional materials & final projects
datahub	https://datahub.ucsd.edu	assignment submission
Piazza	https://piazza.com/ucsd/spring2023/cogs138_sp23_a00/home (course code on canvas home page)	questions, discussion, and regrade requests
Canvas	https://canvas.ucsd.edu/courses/44897	grades, lecture videos
Anonymous Feedback	Will be able to submit via google form	If I ever offend you, use an example you are uncomfortable with, or to provide general feedback. Please remain constructive and polite

Data structures (Types, Tidy Data, Data Intuition), Data Cleaning

Neural data and structures

- Neural data science generates and processes large amounts of data
- Data must be stored in some organized way for analysis - “Structure”
    - There are three classes of data storage we will discuss - *structured*, *semi-structured*, *unstructured*

Data Structures Review

Structured data

- Can be stored in database SQL
- Tables with rows and columns
- Requires a relational key
- 5-10% of all data

Semi-structured data

- Doesn't reside in a relational database
- Has organizational properties (easier to analyze)
- CSV, XML, JSON

Unstructured

- Non-tabular data
- 80% of the world's data
- Images, text, audio, videos

Question

- Why do we do this? What do you think?
- Could we perform neural data science without understanding data structure or giving it any thought?

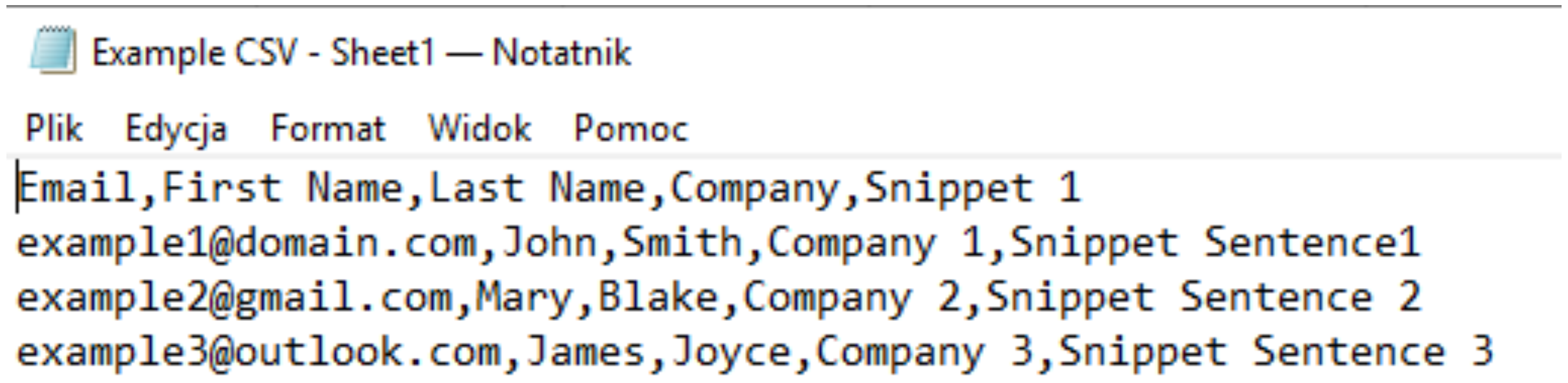
(Semi-)Structured Data

Data that is stored in such a way that it is easy to search and work with. These data are stored in a particular format that adheres to organization principles imposed by the file format. These are the data structures data scientists work with most often.

CSV files

Each column separated by a comma

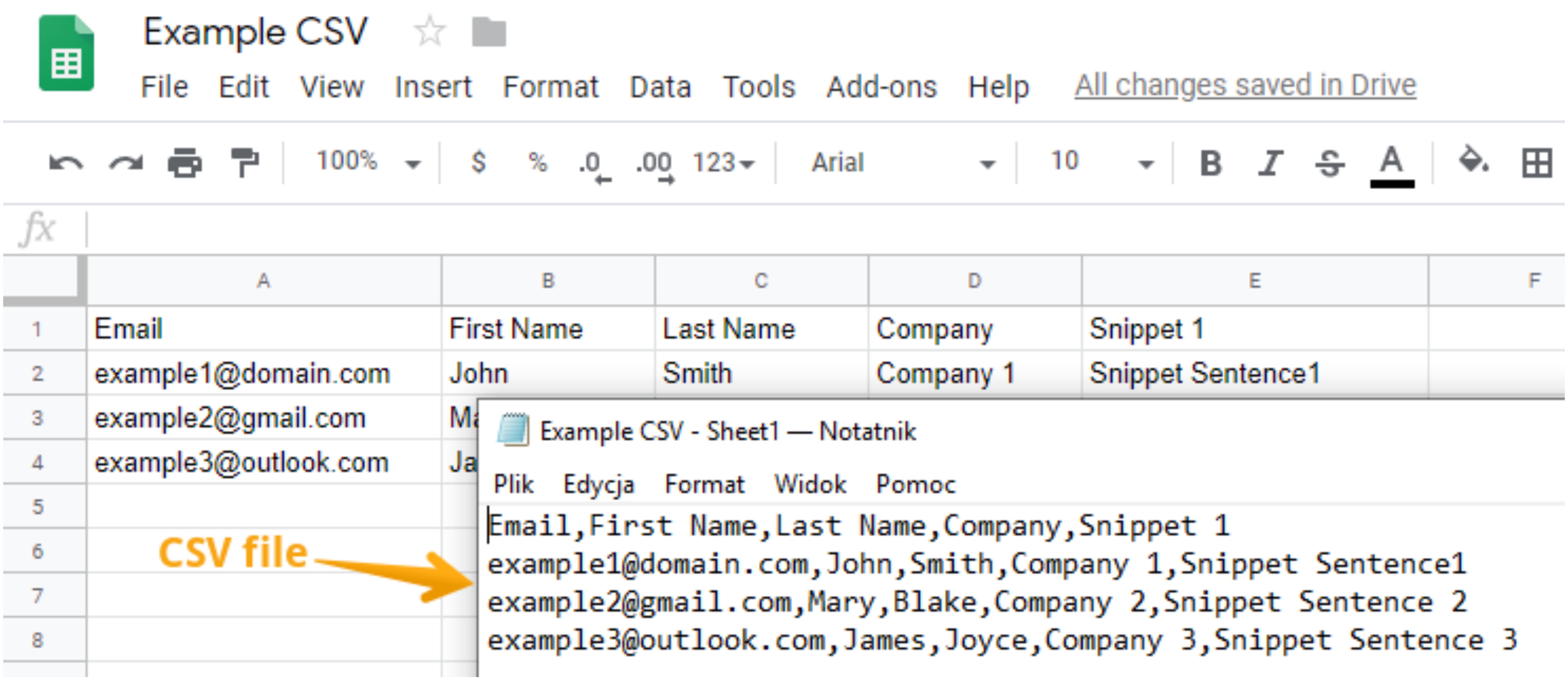
Has the extension “.csv”


Example CSV - Sheet1 — Notatnik

Plik Edycja Format Widok Pomoc

```
Email,First Name,Last Name,Company,Snippet 1
example1@domain.com,John,Smith,Company 1,Snippet Sentence1
example2@gmail.com,Mary,Blake,Company 2,Snippet Sentence 2
example3@outlook.com,James,Joyce,Company 3,Snippet Sentence 3
```

Each row is separated by a new line


Example CSV
File Edit View Insert Format Data Tools Add-ons Help All changes saved in Drive
100% $ % .0 .00 123 Arial 10 B I S A
fx
A B C D E F
1 Email First Name Last Name Company Snippet 1
2 example1@domain.com John Smith Company 1 Snippet Sentence1
3 example2@gmail.com
4 example3@outlook.com
Example CSV - Sheet1 — Notatnik
Plik Edycja Format Widok Pomoc
Email,First Name,Last Name,Company,Snippet 1
example1@domain.com,John,Smith,Company 1,Snippet Sentence1
example2@gmail.com,Mary,Blake,Company 2,Snippet Sentence 2
example3@outlook.com,James,Joyce,Company 3,Snippet Sentence 3
CSV file

CSV

JSON: key-value pairs

nested/hierarchical data

{"Name": "Isabela"}

key

value

These are all nested within attributes

```
"attributes": {
  "Take-out": true,
  "Wi-Fi": "free",
  "Drive-Thru": true,
  "Good For": {
    "dessert": false,
    "latenight": false,
    "lunch": false,
    "dinner": false,
    "breakfast": false,
    "brunch": false
  },
```

These are all nested within “Good For”

JSON

Extensible Markup Language (XML): nodes, tags, and elements

nested/hierarchical data

A **node**

An *opening* **tag**

An **element**

```
$node
<tag>
    <tag2> more content </tag2>
    <tag3> more content </tag3>
</tag>
```

A *closing* **tag**

XML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<customers>
    <customer>
        <customer_id>1</customer_id>
        <first_name>John</first_name>
        <last_name>Doe</last_name>
        <email>john.doe@example.com</email>
    </customer>
    <customer>
        <customer_id>2</customer_id>
        <first_name>Sam</first_name>
        <last_name>Smith</last_name>
        <email>sam.smith@example.com</email>
    </customer>
    <customer>
        <customer_id>3</customer_id>
        <first_name>Jane</first_name>
        <last_name>Doe</last_name>
        <email>jane.doe@example.com</email>
    </customer>
</customers>
```

XML

Relational Databases: A set of interdependent tables

1. Efficient Data Storage
2. Avoid Ambiguity
3. Increase Data Privacy

Results | Messages

	EmployeeId	FirstName	LastName	DepartmentName
1	1	Ken	Sanchez	Executive
2	2	Terri	Duffy	Engineering
3	3	Roberto	Tamburello	Engineering
4	4	Rob	Walters	Engineering
5	5	Gail	Erickson	Engineering
6	6	Jossef	Goldberg	Engineering
7	7	Dylan	Miller	Support
8	8	Diane	Margheim	Support
9	9	Gigi	Matthew	Support
10	10	Michael	Raheem	Support

relational
database

Information is stored across tables

	unique_identifier	
	AH13JK	
	JJ29JJ	
	CI21AA	

	unique_identifier	
	AH13JK	
	JJ29JJ	
	JJ29JJ	
	XJ11AS	
	CI21AA	

	unique_identifier	
	AH13JK	
	SE92FE	
	CI21AA	

entries are *related* to one another by their unique identifier

relational database

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	**JJ29JJ**	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

health inspections

id	inspection_date	inspector	score
AH13JK	2018-08-21	Sheila	97
JJ29JJ	2018-03-12	D'eonte	98
JJ29JJ	2018-01-02	Monica	66
XJ11AS	2018-12-16	Mark	43
CI21AA	2018-08-21	Anh	99

rating

id	stars
AH13JK	4.9
JJ29JJ	4.8
XJ11AS	4.2
CI21AA	4.7

relational database

restaurant

name	id	address	type
Taco Stand	AH13JK	1 Main St.	Mexican
Pho Place	**JJ29JJ**	192 Street Rd.	Vietnamese
Taco Stand	XJ11AS	18 W. East St.	Fusion
Pizza Heaven	CI21AA	711 K Ave.	Italian

health inspections

id	inspection_date	inspector	score
AH13JK	2018-08-21	Sheila	97
JJ29JJ	2018-03-12	D'eonte	98
JJ29JJ	2018-01-02	Monica	66
XJ11AS	2018-12-16	Mark	43
CI21AA	2018-08-21	Anh	99

rating

id	stars
AH13JK	4.9
JJ29JJ	4.8
XJ11AS	4.2
CI21AA	4.7

Two different restaurants with the same name will have different unique identifiers

relational database

Unstructured Data

Some datasets record information about the state of the world, but in a more heterogeneous way. Perhaps it is a large text corpus with images and links like Wikipedia, or the complicated mix of notes and test results appearing in personal medical records.

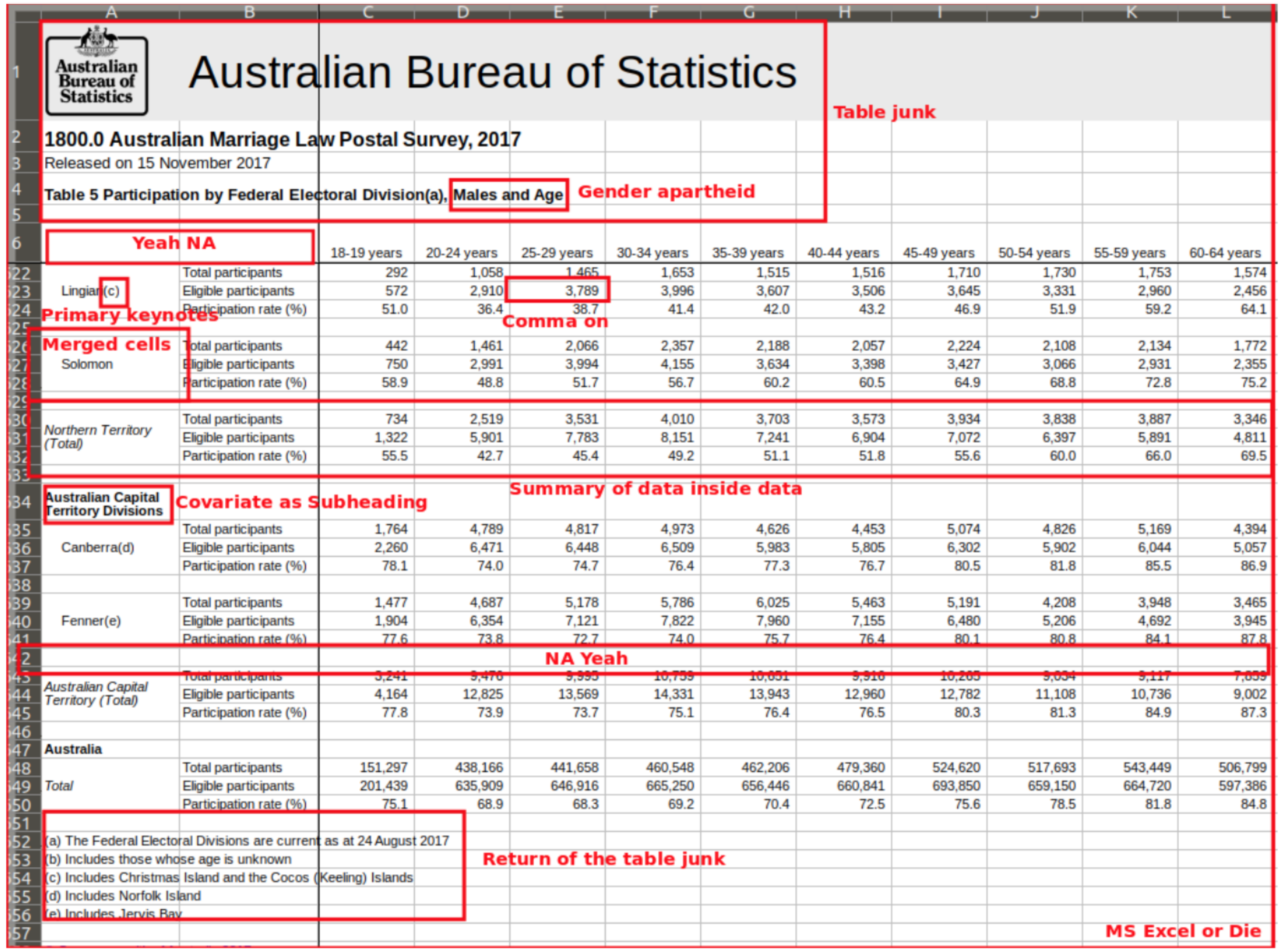
Unstructured Data Types

- Text files and documents
- Websites and applications
- Sensor data
- Image files
- Audio files
- Video files
- Email data
- Social media data

Tidy Data

"Good data scientists understand, in a deep way, that the heavy lifting of cleanup and preparation isn't something that gets in the way of solving the problem: it is the problem." - DJ Patil

untidy data



data wrangling

tidy data

	area	gender	age	State	Area (sq km)	Eligible participants	Participation rate (%)	Total participants	Total Paticipants
2	Adelaide	Female	18-19 years	SA	76	1341	83.5	1120	1120
3	Adelaide	Female	20-24 years	SA	76	4620	81.2	3750	3750
4	Adelaide	Female	25-29 years	SA	76	4897	81.8	4004	4004
5	Adelaide	Female	30-34 years	SA	76	4784	79.8	3820	3820
6	Adelaide	Female	35-39 years	SA	76	4319	79	3411	3411
7	Adelaide	Female	40-44 years	SA	76	4310	80.6	3472	3472
8	Adelaide	Female	45-49 years	SA	76	4579	81.4	3728	3728
9	Adelaide	Female	50-54 years	SA	76	4475	84.7	3791	3791
10	Adelaide	Female	55-59 years	SA	76	4622	87.3	4033	4033
11	Adelaide	Female	60-64 years	SA	76	4342	89.3	3879	3879
12	Adelaide	Female	65-69 years	SA	76	3970	90.7	3602	3602
13	Adelaide	Female	70-74 years	SA	76	3009	90.3	2716	2716
14	Adelaide	Female	75-79 years	SA	76	2156	88.5	1908	1908
15	Adelaide	Female	80-84 years	SA	76	1673	85.1	1423	1423

https://medium.com/@miles.mcbain/tidying-the-australian-same-sex-marriage-postal-survey-data-with-r-5d35cea07962

Tidy Data

1. Each variable you measure should be in a single column

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

2. Every observation of a variable should be in a different row

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

3. There should be one table for each type of data

Demographic Survey Data

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

Doctor's Office Measurements Data

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

4. If you have multiple tables, they should include a column in each *with the same column label* that allows them to be joined or merged

	A	B	C	D	E	F	G
1	ID	LastName	FirstName	Sex	City	State	Occupation
2	1004	Smith	Jane	female	Frederick	MD	Welder
3	4587	Nayef	Mohammed	male	Upper Darby	PA	Nurse
4	1727	Doe	Janice	female	San Diego	CA	Doctor
5	6879	Jordan	Alex	male	Birmingham	AL	Teacher

	A	D	E	F	G
1	ID	Height_inches	Weight_lbs	Insulin	Glucose
2	1004	65	180	0.60	163
3	4587	75	215	1.46	150
4	1727	62	124	0.72	177
5	6879	77	160	1.23	205

Tidy data == rectangular data

A

	A	B	C	D	E
1	id	sex	glucose	insulin	triglyc
2	101	Male	134.1	0.60	273.4
3	102	Female	120.0	1.18	243.6
4	103	Male	124.8	1.23	297.6
5	104	Male	83.1	1.16	142.4
6	105	Male	105.2	0.73	215.7

Broman KW, Woo KH. (2017) Data organization in spreadsheets. *PeerJ Preprints* 5:e3183v1 https://doi.org/10.7287/peerj.preprints.3183v1

Tidy Data Benefits

1. Consistent data structure
2. Foster tool development
3. Require only a small set of tools to be learned
4. Allow for datasets to be combined

Data Intuition

https://www.youtube.com/watch?v=0YzvupOX8Is

Has humanity produced enough paint to cover the entire land area of the Earth?

—Josh (Bolton, MA)

Fermi Estimation

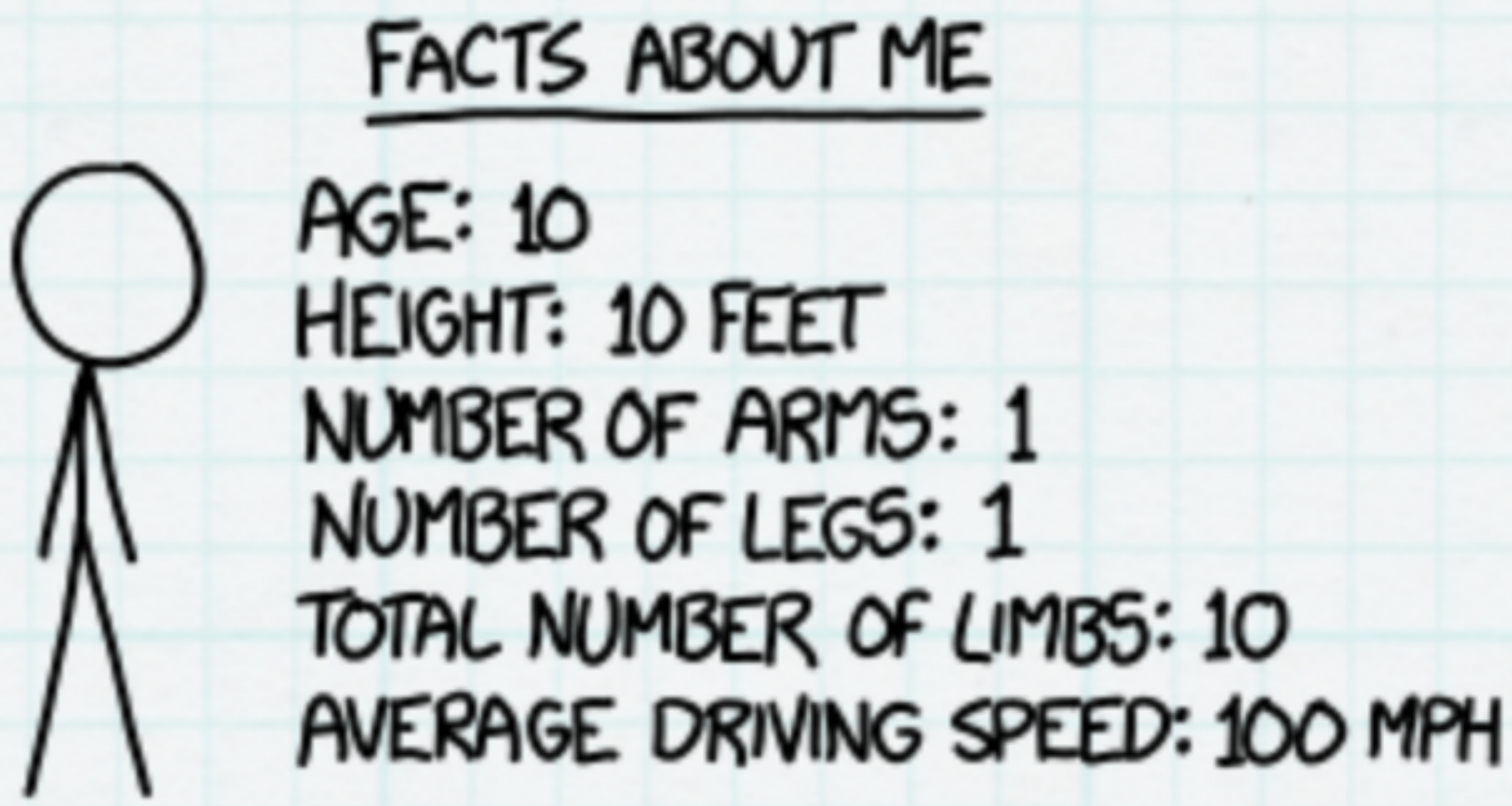
Has humanity produced enough paint to cover the entire land area of the Earth?

A
YES

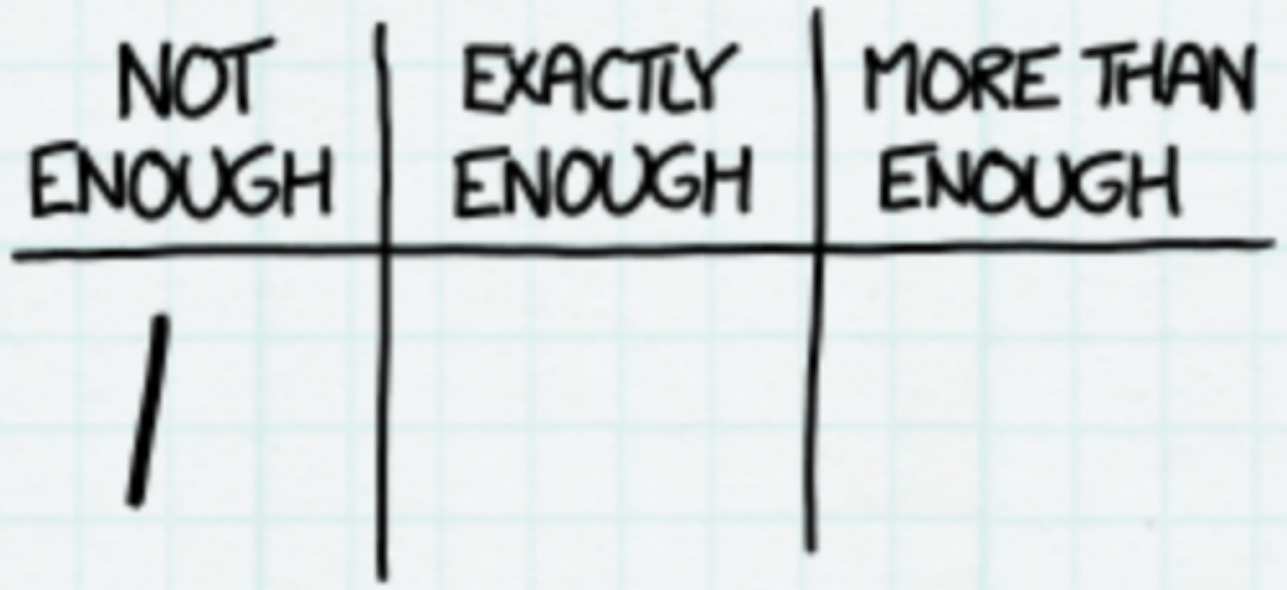
B
NO

This answer is pretty straightforward. We can look up the size of the world's paint industry, extrapolate backward to figure out the total amount of paint produced. We'd also need to make some assumptions about how we're painting the ground. Note: When we get to the Sahara desert, I recommend not using a brush.

But first, let's think about different ways we might come up with a guess for what the answer will be. In this kind of thinking—often called **Fermi estimation**—all that matters is getting in the right ballpark; that is, the answer should have about the right number of digits. In Fermi estimation, you can round [1] all your answers to the nearest order of magnitude:



Let's suppose that, on average, everyone in the world is responsible for the existence of two rooms, and they're both painted. My living room has about 50 square meters of paintable area, and two of those would be 100 square meters. 7.15 billion people times 100 square meters per person is a little under a trillion square meters—an area smaller than Egypt.



Let's make a wild guess that, on average, one person out of every thousand spends their working life painting things. If I assume it would take me three hours to paint the room I'm in,[2] and 100 billion people have ever lived, and each of them spent 30 years painting things for 8 hours a day, we come up with 150 trillion square meters ... just about exactly the land area of the Earth.

How much paint does it take to paint a house? I'm not enough of an adult to have any idea, so let's take another Fermi guess.

Based on my impressions from walking down the aisles, home improvement stores stock about as many light bulbs as cans of paint. A normal house might have about 20 light bulbs, so let's assume a house needs about 20 gallons of paint.[3] Sure, that sounds about right.

The average US home costs about $200,000. Assuming each gallon of paint covers about 300 square feet, that's a square meter of paint per $300 of real estate. I vaguely remember that the world's real estate has a combined value of something like $100 trillion, [4] which suggests there's about 300 billion square meters of paint on the world's real estate. That's about one New Mexico.

Of course, both of the building-related guesses could be overestimates (lots of buildings are not painted) or underestimates (lots of things that are not buildings [5] are painted) But from these wild Fermi estimates, my guess would be that there probably isn't enough paint to cover all the land.

So, how did Fermi do?

According to the report **The State of the Global Coatings Industry**, the world produced 34 billion liters of paints and coatings in 2012.

There's a neat trick that can help us here. If some quantity—say, the world economy—has been growing for a while at an annual rate of **n**—say, 3% (0.03)—then the most recent year's share of the whole total so far is $1 - \frac{1}{1+n}$, and the whole total so far is the most recent year's amount times $1 + \frac{1}{n}$.

https://what-if.xkcd.com/84/

If we assume paint production has, in recent decades, followed the economy and grown at about 3% per year, that means the total amount of paint produced equals the current yearly production times 34.[6] That comes out to a little over a trillion liters of paint. At 30 square meters per gallon,[7] that's enough to cover 9 trillion square meters—about the area of the United States.

So the answer is no; there's not enough paint to cover the Earth's land, and—at this rate—probably won't be enough until the year 2100.

Data Intuition

1. Think about your question and your expectations
2. Do some Fermi calculations (back of the envelope calculations)
3. Write code & look at outputs <- think about those outputs
4. Use your gut instinct / background knowledge to guide you
5. Review code & fix bugs
6. Create test cases - “Sanity checks”

What is data cleaning?

- Fixing/removing incorrect, corrupted, incorrectly formatted, duplicate, incomplete, data within a dataset
- Many issues combining data sources and types, researcher styles, standards, recording errors, etc

Consequences of poorly cleaned data

- Unreliable outcomes and algorithms
- Difficult to detect these issues
- Biased results
- Failure to process algorithms (for example NANs causing errors)

Variability in cleaning

- There is no one process to clean data
- Varies from set to set, project to project, software to software
- But can establish a ‘template’ procedure/process of ‘check-offs’ to make sure you’ve done your best to address it

Methods can be

- Interactive through ‘wrangling tools’
- Automated through scripts, programs or other software (batch processing)

On to today…

A quick overview of one possible data cleaning process example

1. View your data (EDA) - commands (‘print()’, ‘dataFrame.head()’, ‘dataFrame.shape’)

2. Compute the missing proportions of data (NANs etc)

3. View each column data type, format, content

4. Check for trailing white spaces in text, eliminate characters that are irrelevant (punctuation, symbols, etc)

5. Explore if any columns need to be split or combined

6. Check uniqueness of values (sanity check)

To the notebook overview

Visualization of neural data

- https://mne.tools/stable/auto_tutorials/evoked/20_visualize_evoked.html
- https://mne.tools/stable/auto_tutorials/inverse/70_eeg_mri_coords.html#sphx-glr-auto-tutorials-inverse-70-eeg-mri-coords-py

Outlier detection