Week 1: Exploratory Data Analysis

MAISI

Building Interpretable AI in Healthcare
Michigan Data Science Team - Winter 2026

Week 1 Agenda

01. Meet the Leads + Icebreaker!

Get to know us and start to learn about each other with a fun start-of-semester icebreaker!

02. Intro to EDA

What is Exploratory Data Analysis, which Python libraries does it utilize, and how will it be useful going forward?

03. Identifying Good Data

Using reliable training data is paramount. But how can we tell if our dataset is trustworthy and representative?

04. Hands-On Data Science!

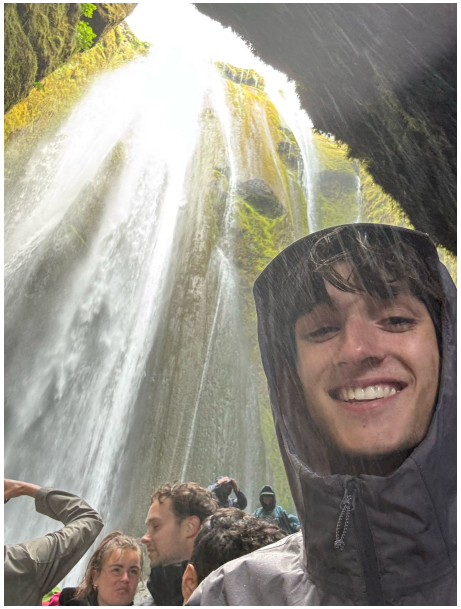
Let's apply these EDA skills to datasets of your choice and get our projects rolling!

01

Meet the Leads / Fun Icebreaker!!

Get to know us and start to learn about each other with a fun start-of-semester icebreaker!

Meet the Leads! - Will McKanna



Hometown: Rockford, MI

Major: DS and Statistics

Year: Sophomore

Ask me about: Studying abroad in Iceland, crocheting, trombone, Michigan and Detroit football, being a UMich Tour Guide

Meet the Leads! - Seena Simkani

Hometown: Grand Blanc, MI

Major: Data Science

Year: Sophomore

Ask me about: Flying planes, playing the piano, mopeds, music

Team Building - Icebreaker Bingo!

	A	B	C	D	E
1	I'm a fan of the Detroit Lions	Slept overnight at a UofM non dorm building	I can whistle	I'm part of another CS/DS club	I get the supreme slice @ Joe's
2	I'm a member of MAISI	I have season tickets to Michigan Football	Touched grass this summer (3+ outdoor activities)	I'm a Data Science major	I know the capital of Mongolia
3	I play a sport	I'm a non CS/DS major	I'm a part of MDST	I pay for guac at chipotle	I live on North
4	I'm a Computer Science major	Took Math 215 at Michigan (WCC >>)	I play an instrument	I've taken a formal statistics class (HS/college)	I've visited the Upper Peninsula
5	I'm from the state of Michigan	Skipped < half of my lectures last week	I live on Central	I've customized my VSCode	Read 3+ books this school year (since August)

02 Intro to EDA

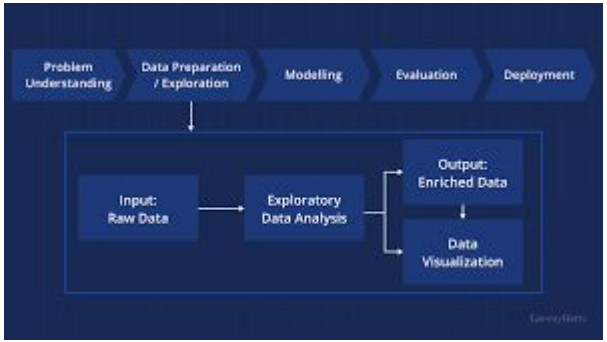
What is Exploratory Data Analysis, which Python libraries does it utilize, and how will it be useful going forward?

What is Exploratory Data Analysis (EDA)?

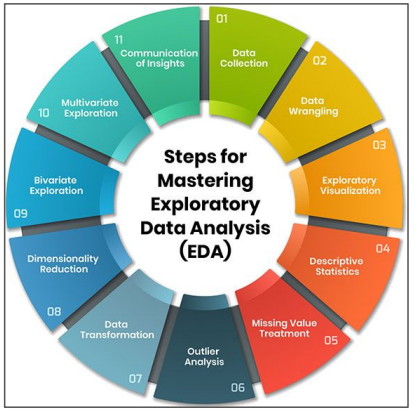
Definition: “A process used to analyze datasets to summarize their main characteristics, often using visualization tools.”

Main Goals:

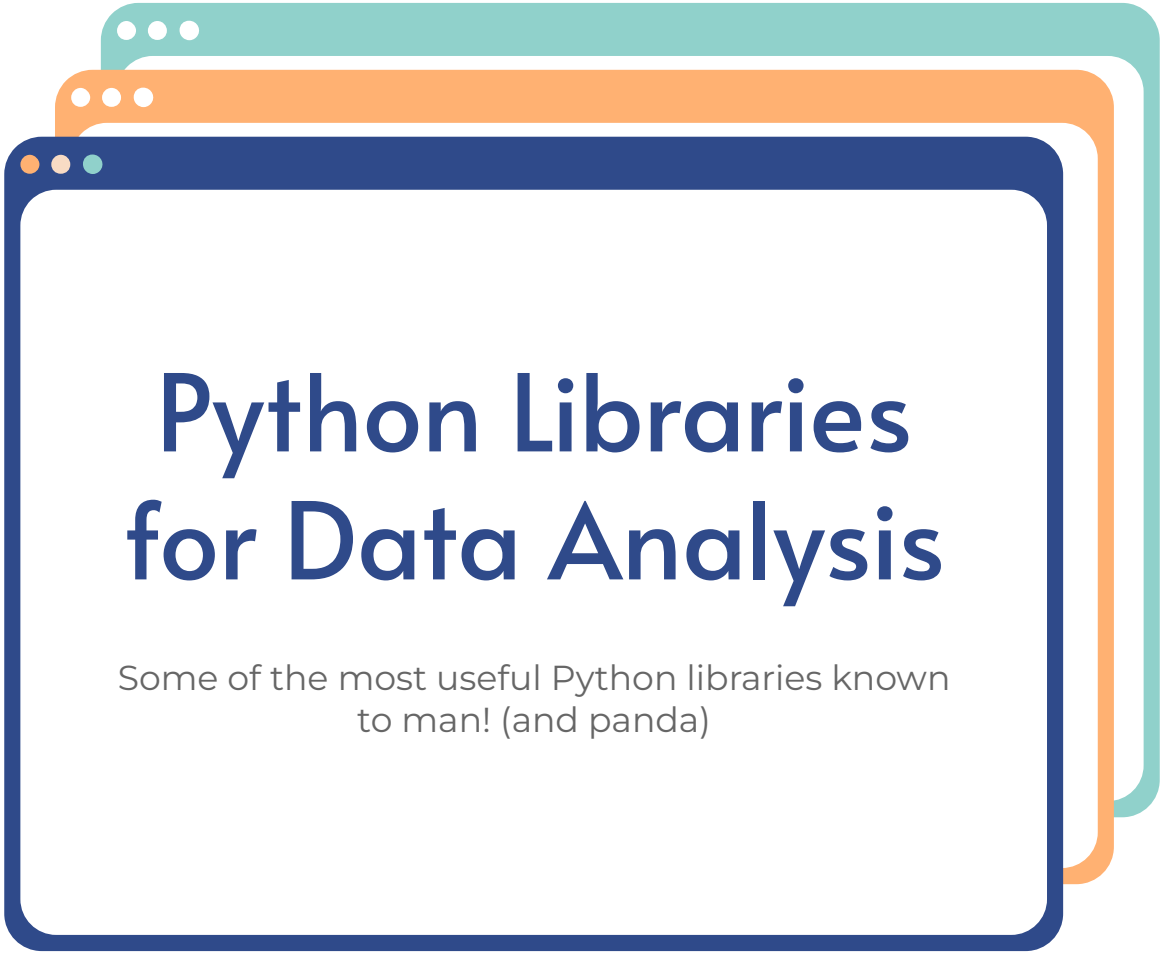
- Understand Data Structure - Inspect the dataset to understand shape and types.
- Identify Patterns and Relationships - Use statistics and plots to find correlations and trends.
- Detect Data Quality Issues - Locate missing values, outliers, and inconsistencies.
- Generate Hypotheses - Develop questions from insights that can be investigated further.



The Significance of EDA



- **EDA** is something of a philosophy of understanding data <u>without assumptions</u>. This enables us to obtain a thorough understanding of its **context, patterns,** and **limitations.**
- It reveals insights and helps in identifying **biases** and **relationships** that guide further analysis
- EDA ensures that data is **well-understood** and clean.
    - Many datasets have missing values of some sort. This can mess up your code! (and your understanding)
- This process informs **stronger hypotheses** and leads to better, **data-driven decisions**.


Python Libraries
for Data Analysis
Some of the most useful Python libraries known
to man! (and panda)

What is Pandas?

- Pandas is a Python library for data manipulation and analysis
    - The name “pandas” is derived from “Panel Data” and “Python Data Analysis” - helps us **understand data structures**
- Produces two main structures: **Series (1D)** and **DataFrame (2D)** <- Imagine a table in a database!
- Main Uses
    - Identify **missing values** and discover **patterns**
    - Enables **data cleaning and preprocessing** – crucial steps to data preparation

What is NumPy?

- NumPy stands for “Numerical Python”
  - Provides **fast, efficient arrays** and a wide range of **mathematical functions**
  - Key tool for scientific computing

- NumPy’s Relationship with pandas and Its Role in EDA
  - Foundational for pandas: pandas is **built on top** of NumPy; pandas DataFrames internally use NumPy arrays
  - Exploratory Data Analysis: NumPy enables data manipulation, statistical analysis, and complex calculations required for understanding data before visualization or modeling

03 Identifying Good Datasets

Using reliable training data is paramount. But how can we tell if our dataset is trustworthy and representative?

Why is it important to have good data?

- Data imbued with **bias** can perpetuate harmful stereotypes, lead to misguided takeaways, and mess with the training of machine learning models.

- In the context of **healthcare** , data with accuracy or interpretability deficiencies can lead to further marginalization of at-risk populations.
    - Ex. A machine learning model is improperly trained on patient data that includes financial records and comes to the conclusion that low-income patients are less likely to be readmitted and thus must be at lower risk for their condition to reemerge. (What's the problem here?)

- There are three main types of bias that we will highlight:
    - Sampling Bias
    - Feature Selection Bias
    - Models Perpetuating Stereotypes

Sampling Bias

- The strength of an ML model is proportional to the size of its training dataset
  - Like a human: more studying, more knowledge

- **Non-representative datasets** (datasets with sampling bias) have varying degrees of representation for people/instances in different groups

- Ex: In 2007, UMass-Amherst created a dataset called Labeled Faces in the Wild, an image dataset with thousands of human faces. However...
  - It was 77% male and 83% white
  - There were 530 images of just George W. Bush (the president at the time) - which is twice as many as all images of all Black women combined

Q: What can we take away from this?
A: It matters what data we train ML models on (EDA would expose these flaws!)

George Robertson (22)

George W Bush (530)

Gerhard Schroeder (109)

Feature Selection Bias

- **Feature Selection** is the process of identifying and selecting the most relevant features of a dataset to use in training an ML model.

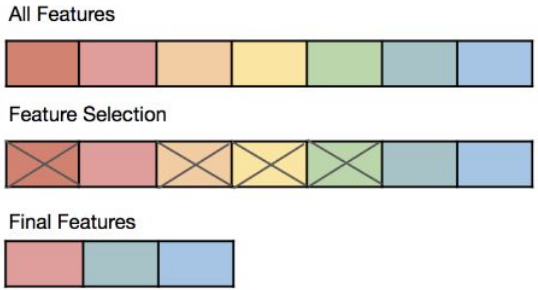
- In most ML models, this is a crucial step to reduce overfitting, improve performance, and help the model to generalize.

Unfair outcomes start to emerge when the features are selected improperly.

- If a feature reflects a certain protected group, this can lead to unfair outcomes between groups.
  - E.g. if the models trained on your dataset took a patient's race in as a feature, this can lead to undesirable outcomes and racial profiling.

- A **proxy** is a feature that is not directly reflective of a certain group but is correlated with one.

- Four not-so-obvious examples
  - Last name
  - Skin cancer risk
  - If you worked food service in HS
  - ZIP code



Data Perpetuating Stereotypes

- Society is inherently biased in uncountably many ways, some more subversive than others. But what happens when a dataset is "well-adjusted" to how we think? Is an algorithm biased if it makes decisions based off of biased behavior it learned from humans?

- Example: Natural Language Processing (NLP) model **word2vec**
  - Trained to calculate the relationship between words
    - However, a now-famous paper by Tolga Bolukbasi found that this model perpetuated some... unsatisfactory stereotypes
  - Some outputs included: "'man' is to 'computer scientist' as 'woman' is to 'homemaker'"

- Question: is this model sexist, or does it merely reflect how we perceive the world implicity?

Possible Datasets to Use

Diabetes Data from 1999-2008

Lung Cancer Data

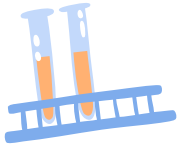
Cardiovascular Health Data

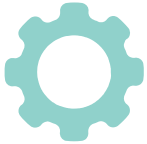
Stroke Prediction Data

Heart Attack Data

Sleep Health Data

04





References

Here are a handful of useful guides to high-utility functions, Git demos, and Colab explanations!

Project Timeline

Week 1 (01/25): Icebreaker/EDA intro, choosing a good dataset

Week 2 (02/01): Data Visualization, Further EDA

Week 3 (02/08): Feature Engineering and Intro to Machine Learning

Week 4 (02/15): Machine Learning and Model Optimization

Week 5 (02/22): Conformal Predictions, Risk Control, and Multi-Class Calibration

NO MEETING 03/01 OR 03/08 – SPRING BREAK

Week 6 (03/15): SHAP Values and Feature Importance Interpretation

Week 7 (03/22): Intro to Streamlit and Project Work Time

Weeks 8 + 9 (03/29 + 04/05): Final Project work time + Data Science Night prep!

Important Functions to Know (pt. 1)

Syntax	Description
import pandas as pd	Import the **pandas** library, using the alias **pd** for convenience
import numpy as np	Import the **NumPy** library, using alias **np** for numerical operations
import seaborn as sns	Import **Seaborn** for statistical data visualization
df = pd.read_csv('file.csv')	Load a **CSV** file into a **pandas DataFrame** for data manipulation
df.head()	Returns the **first 5 rows** of the DataFrame, useful for quickly inspecting data
df.info()	Provides a concise **summary** of the DataFrame, including **data types** and **non-null values**

Important Functions to Know (pt. 2)

Syntax	Description
df.describe()	Generates **descriptive statistics** of numerical columns (mean, median, quartiles, etc.)
df[‘column_name’]	Access a **specific column** in the DataFrame, works like a key in a dictionary
df.drop(columns=[‘col’...])	**Drops specified columns** from the DataFrame,use inplace=True to modify the original DataFrame
df.index	Returns the **index labels** of the DataFrame
df.isnull()	returns a DataFrame of **boolean values**, where each entry indicates whether the corresponding value in df is NaN (missing)

Important Functions to Know (pt. 3)

Syntax	Description
df[‘column’].value_counts()	Returns the **count** of **unique values** in a specific column
df[‘column’].mean()	Returns the **mean value** of a numerical column
df.corr()	Computes **correlation** for numerical columns to understand relationships
df.columns	Lists all **column names** in the **DataFrame**, useful for renaming or viewing dataset structure
df.shape	Returns **# of rows and columns**

Important Functions to Know (pt. 4)

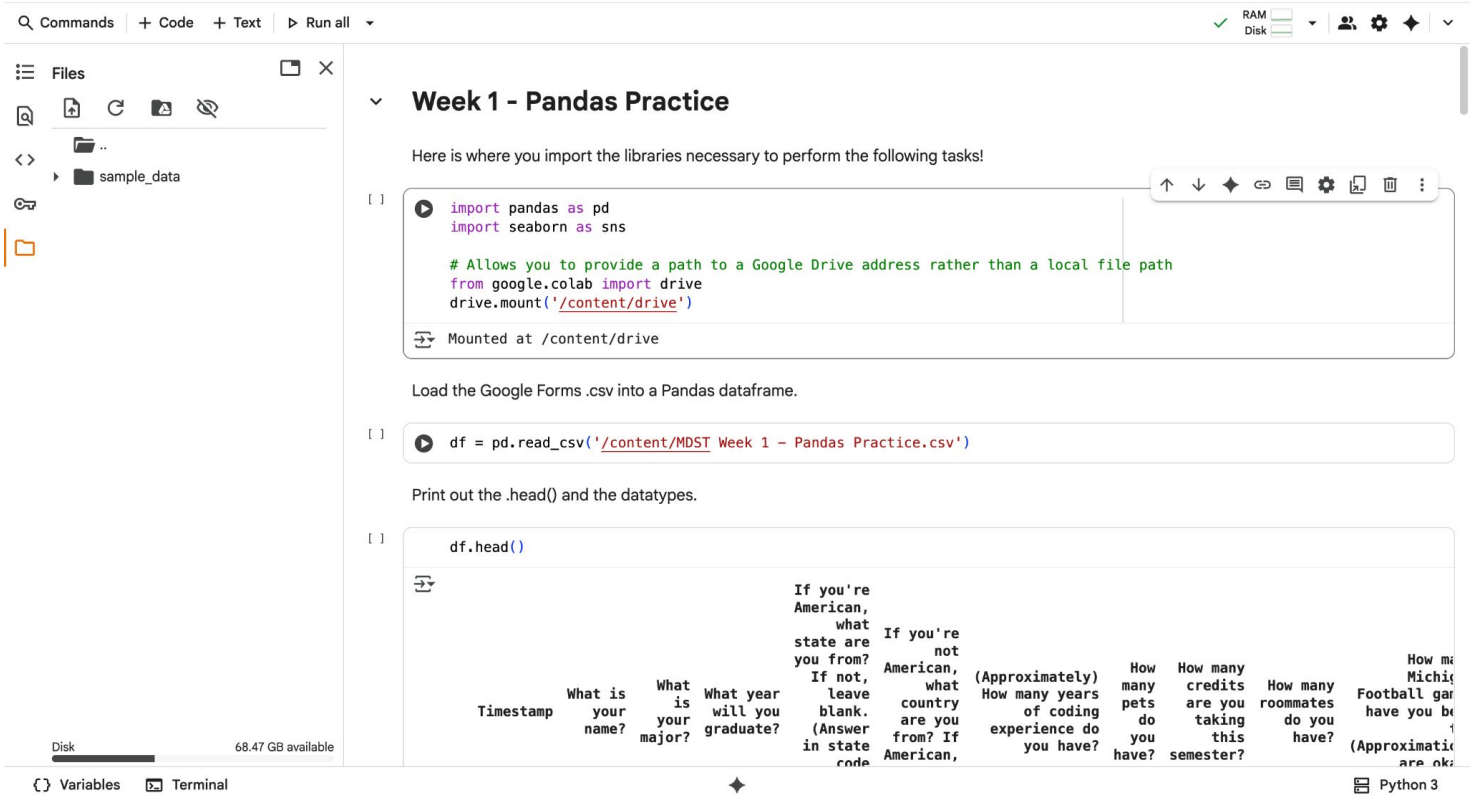
Syntax	Description
df.rename(columns={'old' : 'new'}, inplace=True)	Renames **specific columns** to new names
df.groupby(‘category’)[‘value’]	Groups the DataFrame using a specified column to perform **aggregate functions** (e.g., .sum(), .mean())
df[‘new_column’] = df[‘column’].apply(function)	**Applies** a function to each element or column/row
df.shape	Returns a **tuple** representing the **dimensions** (rows, columns) of the DataFrame
df.dropna()	Removes rows or columns containing **missing values**

Important Functions to Know (pt. 5)

Syntax	Description
df.to_numpy()	Converts Pandas DataFrame into a **NumPy array**
np.shape	Returns a **tuple** representing the **dimensions** (rows, columns) of the array (similar to df.shape)
np.reshape()	**Reshapes** an original numpy array to the specified dimensions (rows, columns)
np.array()	Creates an **N-dimensional array** (ndarray) which is more **memory efficient** than standard python lists
np.unique()	Returns the **unique elements** in a NumPy array (also returns the **counts of each element** given the keyword argument, return_counts=True)

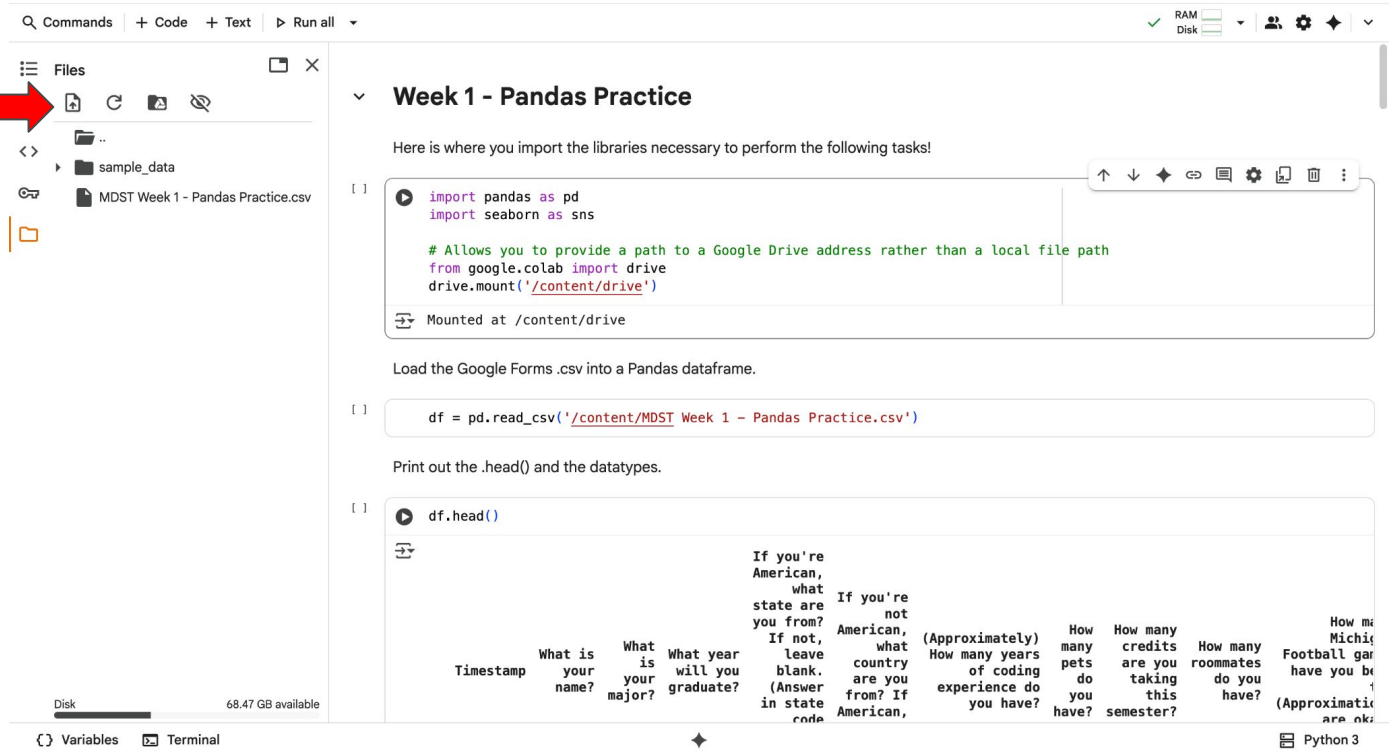
How to Upload Files into Colab

Click on the folder in the sidebar



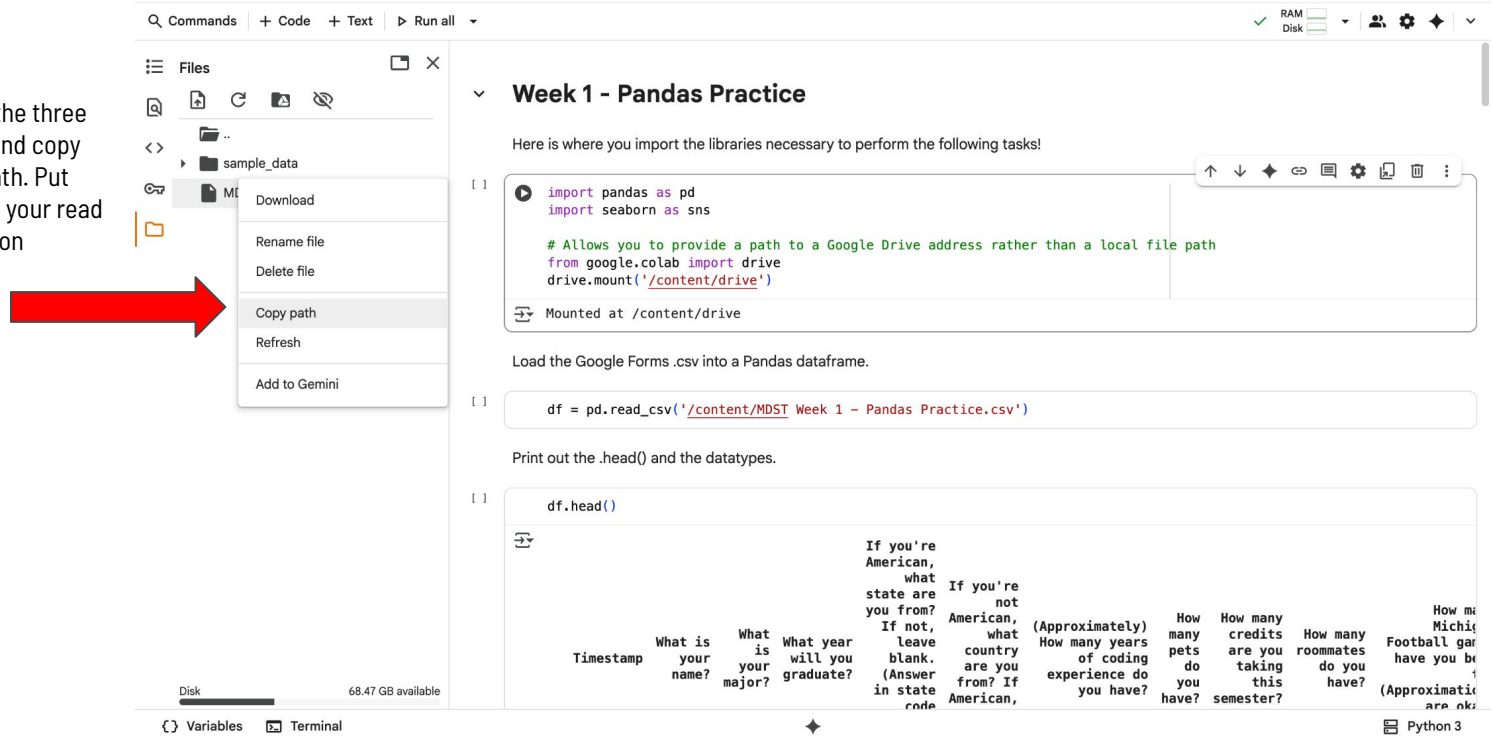
How to Upload Files into Colab

Click the upload button and select the file you want to upload



How to Upload Files into Colab

Click the three dots and copy the path. Put this in your read function



How to Mount Your Google Drive

```
from google.colab import drive

drive.mount('/content/drive')


pd.read_csv('/content/drive/MyDrive/[FILE NAME]')
```

****NOTE:** If you saved your dataset in a folder (not just loose in your MyDrive folder), you will need to add further code to the file path before the file name in the final line. For example, if I saved my data called “alzheimers.csv” in a folder called “MDST-W26,” my read_csv statement would read:

```
pd.read_csv('/content/drive/MyDrive/MDST-W26/alzheimers.csv')
```