

[Source](#)

Welcome to Week 1 Lecture 2!

Data Science in Python &
Machine Learning



Learning Goals:

At the end of this lesson you will be able to:

1. Distinguish 'Practice' assignments from 'Core' assignments.
2. Review feedback, including assignment status:
 - a. Passed, Missing, Resubmit
3. Codealong Introduction to Packages & Numpy
4. Load data into a Pandas dataframe from a .csv or .xlsx file
5. Read error messages to debug code.

Announcements & Reminders

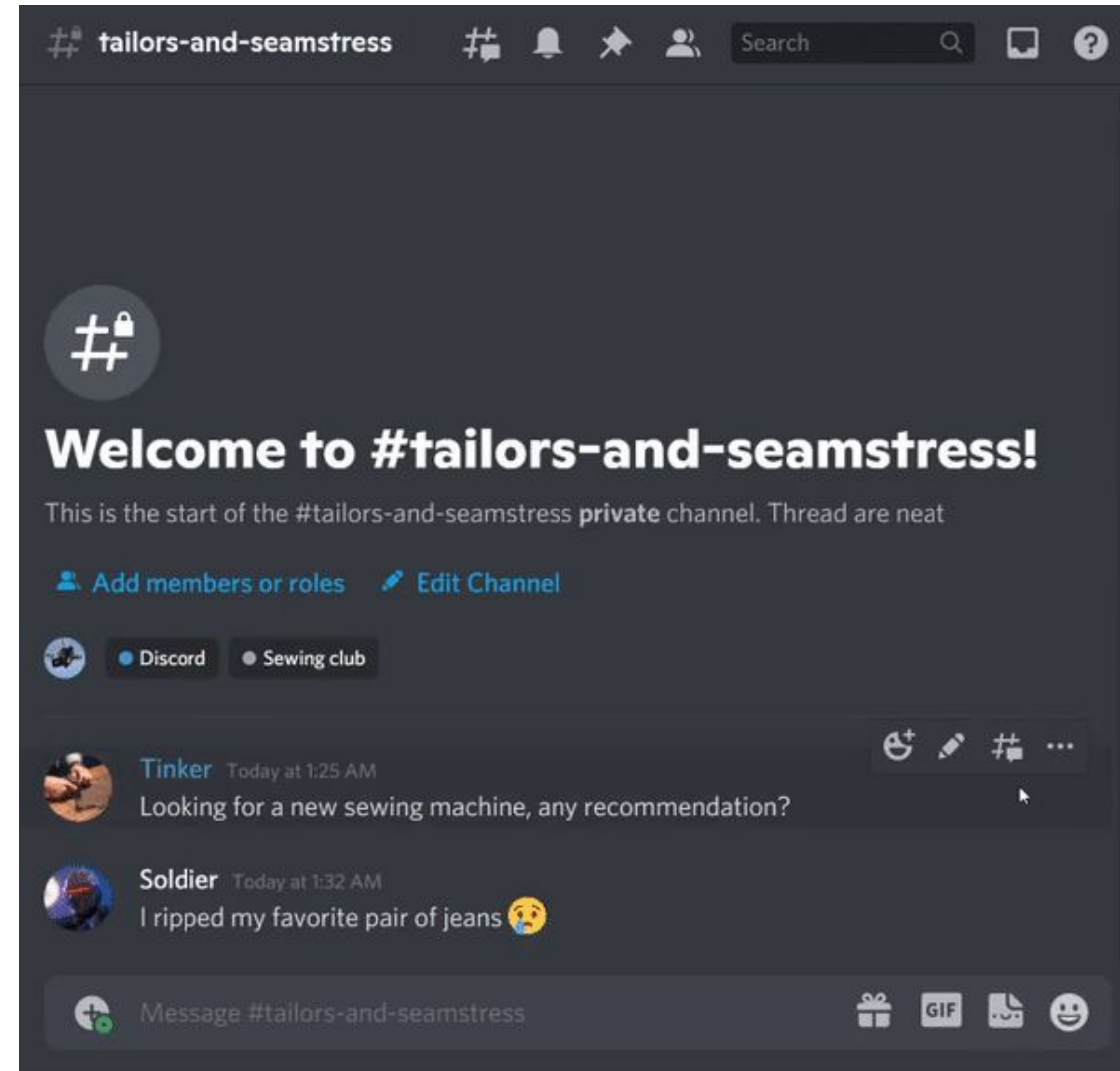
Bonus (Optional)

Lesson/Codealong on Python Basics

- **When?**
 - Tomorrow, 01/28/22 at 5 PM PST (normal lecture time)
- **What?**
 - **We will walk through the solution this week's Code Review Activity** on Python Basics:
 - Repo link: <https://github.com/sensei-jirving/code-reviews-week-1-python-basics>
- **Why?**
 - Because this will be **our final session focusing on JUST basic python.**
 - You **do not need to have completed** the activity to attend.
- Will it be recorded?
 - Yes!
- **Is attendance required?**
 - **Absolutely not!**
 - But strongly recommended.

Please Use Discord Threads Insteads of Replies

- Discord has 2 ways to respond to a message:
 - **✗ Reply**: appears on main channel, disconnected from original message.
 - **✓ Create a thread**: creates a sidebar conversation that keeps every response grouped together and accessible.
- **Why?**
 - **Easier for others to benefit** from your questions and answers.
 - Reduces clutter & intimidation



- **Core** Assignments are due Sundays
- Always double check you feedback document for which assignments are **Core**

GitHub (Practice)	Practice	Week 1 - Sun	✓		
Colab Notebook (Practice)	Practice	Week 1 - Sun	✓		
Calculating Mortgage Payment (Practice)	Practice	Week 1 - Sun	✓	Basic	30 min - 1 hr
FizzBuzz (Practice)	Practice	Week 1 - Sun	✓	Intermediate	30 min - 1 hr
Bakery NumPy Exercise (Core)	Core	Week 1 - Sun	✓	Basic	
Project 1 - Part 1 (Core)	Core	Week 1 - Sun	✓		
Distance and Time (Core)	Core	Week 1 - Sun	✓	Basic	
Interview Questions (Optional)	Optional		✓	Basic	

Your Personal Feedback Document

- A Google Sheet like this has been shared with you!
- Please verify that your name is at the top of this document
- This is where you will see if you have passed or if your task needs to be resubmitted
- Grading occurs on Monday and Tuesday. **This is a live document so check back for updates!**

A	B	C
Student #		
Assignment	Status	Feedback
Data Science Fundamentals		
Week 1		
Bakery NumPy (Core)	passed	Well done.
Project 1 - Part 1 (Core)	resubmit	10/11 Please complete all parts of this assignment
Distance Time (Core)	missing	10/11
Week 2		
Loading & Filtering Data (Core)		
Titanic Data - Pandas (Core)		
Project 1 - Part 2 (Core)		
Week 3		
Average Height Exercise (Core)		
Histograms & Boxplots (Core)		
Project 1 - Part 3 (Core)		
Week 4		
Recreate Graph Part 2 (Core)		
Graphing Exercise (Core)		
Project 1 - Part 4 (Core)		

Assignments marked 'resubmit' count as submitted for the week they are due, BUT must be corrected and resubmitted by the following Sunday or they will be retroactively considered 'missing' for the purpose of calculating Academy Probation.

Week 1 CORE Assignments

These **MUST** be submitted by 11:59 pm PST on Sunday January 30th:

- 1) Bakery NumPy Exercise
- 2) Project 1 - Part 1
- 3) Distance and Time

If a core assignment is missing, you will be placed on academic probation


- even if you have done other assignments and
- even if you have spent a lot of time learning and working

Note: If you accumulate 3 academic probations at any time during the program, you will be recommended for dismissal. Probations are cumulative and irreversible except in the case of documented mitigating circumstances.

Sample Feedback:

***Typically updated by
Wednesday morning***

Student #		Your Name Here
Assignment	Status	Feedback
Data Science Fundamentals		
Week 1		
Bakery NumPy (Core)	resubmit	Great start, but you did not get the correct final answer
Project 1 - Part 1 (Core)	missing	
Distance Time (Core)	passed	great job!
Week 2		
Loading & Filtering Data (Core)		
Titanic Data - Pandas (Core)		
Project 1 - Part 2 (Core)		



- **The ONLY way you will know you have a resubmit is opening this document!**
- Note that if the status box is empty, the assignment has not been reviewed yet.
- If the status is “passed” and the feedback box is empty, no news is good news!

Resubmission



- You will be asked to **resubmit** a task if:
 - There are errors in your code
 - Parts of the assignment are missing
 - Parts of the assignment are incorrect
 - The file or link cannot be opened
 - The work is low quality and/or difficult to read and follow
- Please don't stress if you are asked to **resubmit** an assignment! It just means that there is room for improvement, and we want to ensure you've demonstrated competency and professionalism in each task.
- You can **resubmit** on the Learn Platform in the same place that you submitted your original assignment.
- **Resubmitted** assignments will automatically be checked along with the next week's assignments.
- All assignments marked 'resubmit' must be corrected and resubmitted before the end of the stack in order to move forward.

Python Packages/Modules



- Python comes with many built-in functions [\[Built-In Functions\]](#)
 - print, type, range, etc
- **But most of the functionality we need as data scientists is not included** in base Python.
- **We can download other collections of functions and classes, called Packages** (A.K.A. Libraries A.K.A Modules)
- Packages You will Be Using in Stack 1:
 - Numpy
 - Pandas
 - Matplotlib
 - Seaborn

Intro to Packages with NumPy



- Hop over to our Colab Notebook:
 -  [Intro to Python Packages - NumPy](#)

Loading and Exploring Data with Pandas 🐼!



Getting Data into Colab

Two Options:

1. Save file in Drive
 - a. Mount drive
 - b. Open file folder in Colab on the left:
 - c. Navigate to file in Drive
 - d. Right click file and 'copy filepath'
2. Download data locally and upload to temporary Colab environment
 - a. Download data
 - b. Open file folder on the left
 - c. Drag and drop the file from your computer to upload it
 - d. Right click file and 'copy filepath'



pd.read_csv() and pd.read_excel()

Different file types need different functions to load them:

- .csv files → pd.read_csv('filepath')
- .xlsx files → pd.read_excel('filepath')
- 'filepath' can be a remote URL as long as it points directly to a .csv or .xlsx file.
- Other functions exist to load other file types, but these are the ones we will use.

```
# import pandas
import pandas as pd

# read in the excel file and name your dataframe (df is common, but unoriginal)
df = pd.read_excel('/content/drive/MyDrive/Live Class Materials/Week 01/Total_Population_By_City_Acs_2015.xlsx')

# explore the top 5 rows of dataframe (this is the head)
df.head()
```


Headers

Age	Sex	Charge	Marital Status	Children
46	male	\$124	married	2
32	male	\$65	single	0
56	female	\$75	married	1

Header is top row. (This is default assumption)

```
pd.read_excel('path')
```

```
pd.read_excel('path', header = 0)
```

46	male	\$124	married	2
32	male	\$65	single	0
56	female	\$75	married	1

No header

```
pd.read_excel('path', header = None)
```

ACME Company				
Demographics				
Age	Sex	Charge	Marital Status	Children
46	male	\$124	married	2
32	male	\$65	single	0
56	female	\$75	married	1

Header present, but not at the top

```
pd.read_excel('path', header = 2)
```

Quick Look at top and bottom of dataframe

This will give you the top 5 rows: `df.head()`

To see a different number of rows such as the first 8 rows: `df.head(8)`

To see the bottom 5 rows `df.tail()`

To see a different number of rows such as the last 11 rows `df.tail(11)`

To see random rows `df.sample(5)`

Checking data types

To get lots of info including data type, index, and Column Names `df.info()`

To get JUST the data type for all columns: `df.dtypes`

To get the data type for just one column `df['name'].dtypes`

Slicing Dataframes:

Selecting specific column(s)

To select 1 column as a pandas **series**

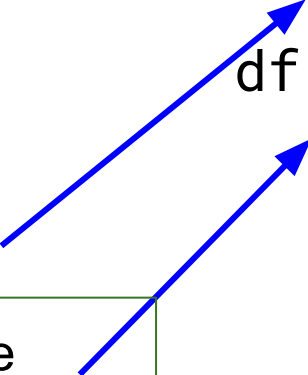
```
df[ 'name' ]
```

To select 1 column as a pandas **dataframe**

```
df[[ 'name' ]]
```

To select multiple columns as a **dataframe**

```
df[[ 'name', 'Manufacturer' ]]
```



Notice double
square brackets
for dataframe

How many rows and columns?

To get lots of info including data type, index,
and Column Names

`df.info()`

To get the shape (rows, columns)

`df.shape`

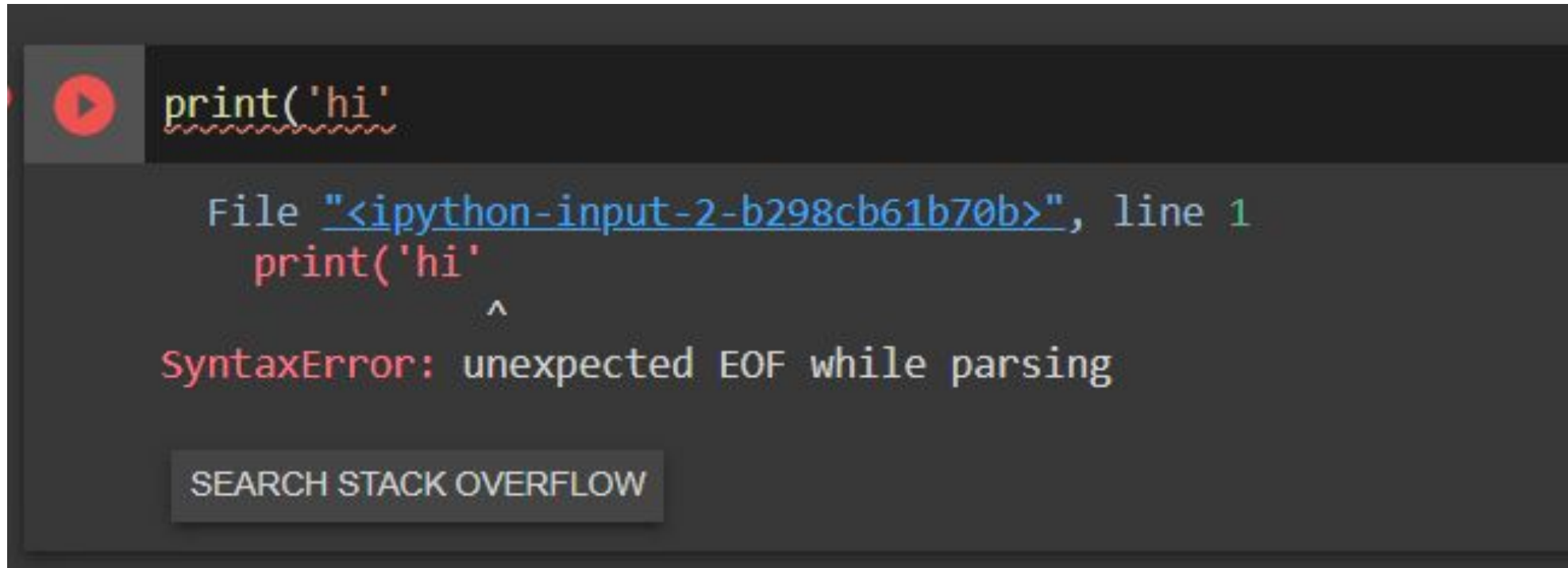
To get the number of rows

`len(df)`

To get the number of columns

`len(df.columns)`

A Note About Errors:

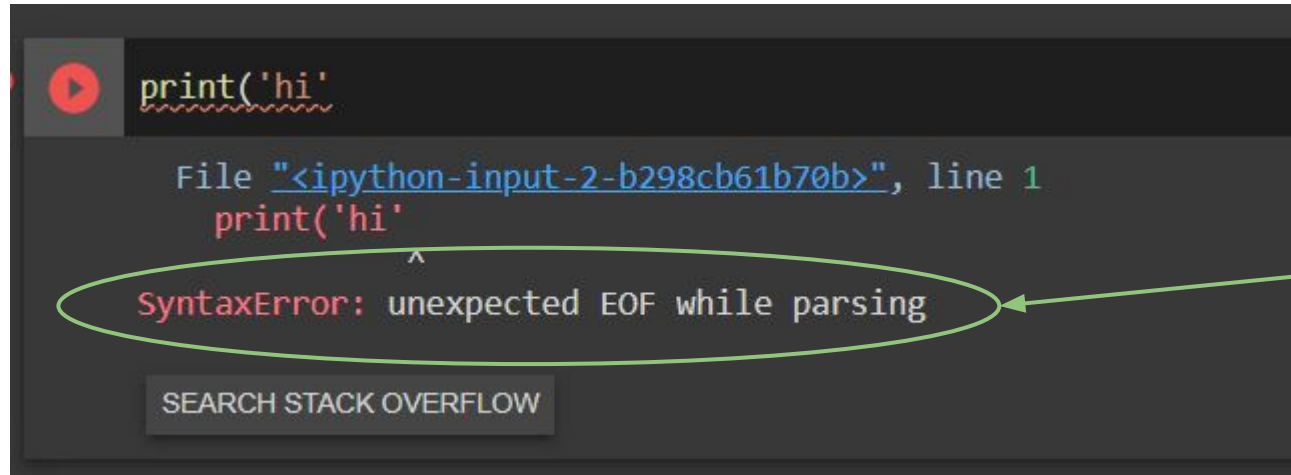
A screenshot of a Jupyter Notebook interface showing a syntax error. At the top, there is a red play button icon next to the code `print('hi'`, which has a red wavy underline. Below this, the error message is displayed: `File "<ipython-input-2-b298cb61b70b>", line 1` followed by `print('hi'` and a red caret (^) pointing to the end of the line. The error text is `SyntaxError: unexpected EOF while parsing`. At the bottom of the error message, there is a button that says "SEARCH STACK OVERFLOW".

```
print('hi'
```

File "<ipython-input-2-b298cb61b70b>", line 1
print('hi'
^
SyntaxError: unexpected EOF while parsing

SEARCH STACK OVERFLOW

Listen to your error messages!



```
print('hi'
```

File "<ipython-input-2-b298cb61b70b>", line 1
print('hi'

SyntaxError: unexpected EOF while parsing

SEARCH STACK OVERFLOW

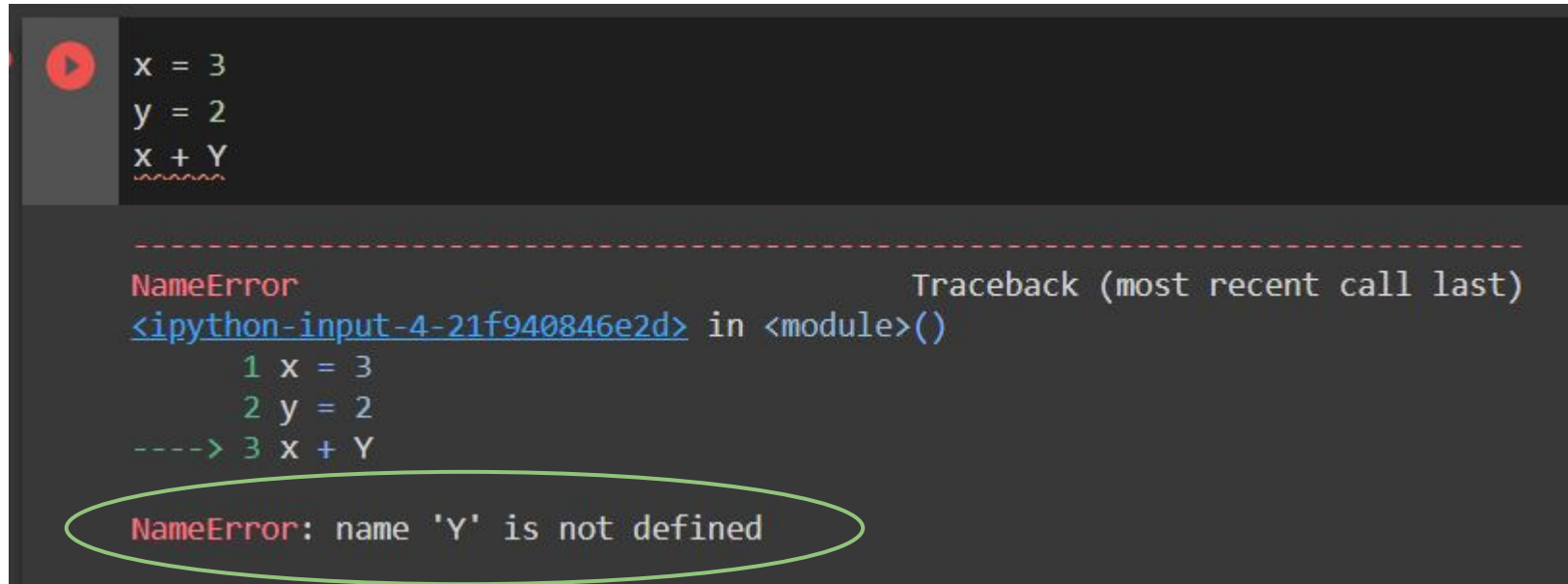
The most helpful information about the error is usually at the **BOTTOM** of the error message.

Sometimes you have to scroll down to find it.

EOF (End of File) error usually indicates you that you don't have complete (open and closed)

()
[]
{ }

Listen to your error messages!



```
x = 3
y = 2
x + Y
```

```
NameError                                Traceback (most recent call last)
<ipython-input-4-21f940846e2d> in <module>()
      1 x = 3
      2 y = 2
----> 3 x + Y

NameError: name 'Y' is not defined
```

This NameError tells you that you have not defined a value for Y

Note that lowercase was used to define `y = 2`, but variables are case sensitive!

Listen to your error messages!

```
df = pd.read_excel('/content/drive/MyDrive/Live Class Materials/Week 01/Total Population By City Acs 2015.xlsx')
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-6-9a11d933a974> in <module>()  
----> 1 df = pd.read_excel('/content/drive/MyDrive/Live Class Materials/Week 01/Total_Population_By_City_Acs_2015.xlsx')
```

```
NameError: name 'pd' is not defined
```

This says that pd is not defined which means I must not have imported the pandas library
or
maybe I did not assign it an alias (pd)
or
maybe I had a typo in the alias.

Either way, it means check that code!

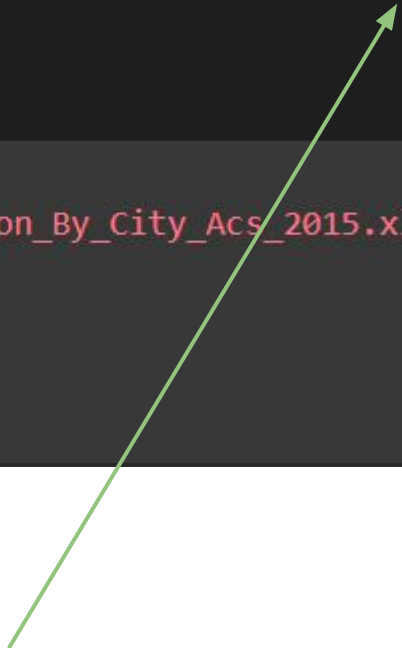
```
# import pandas
import pandas as pd

# read in the excel file and name your dataframe (df is common, but unoriginal)
df = pd.read_excel(/content/drive/MyDrive/Live Class Materials/Week 01/Total Population By City Acs 2015.xlsx')

# explore the top 5 rows of dataframe (this is the head)
df.head()
```

File "<ipython-input-8-a3e56844f2cb>", line 5
df = pd.read_excel(/content/drive/MyDrive/Live Class Materials/Week 01/Total_Population_By_City_Acs_2015.xlsx')
^
SyntaxError: invalid syntax

SEARCH STACK OVERFLOW



SyntaxError means you made a typo in the code.

Here both quotes are at the end instead of one at the beginning.

A variety of typos give syntax errors such as misspelling a keyword like 'prtn' instead of 'print'

Explore [this link](#) for more help in understanding your SyntaxErrors

Loading & Exploring a Data Set

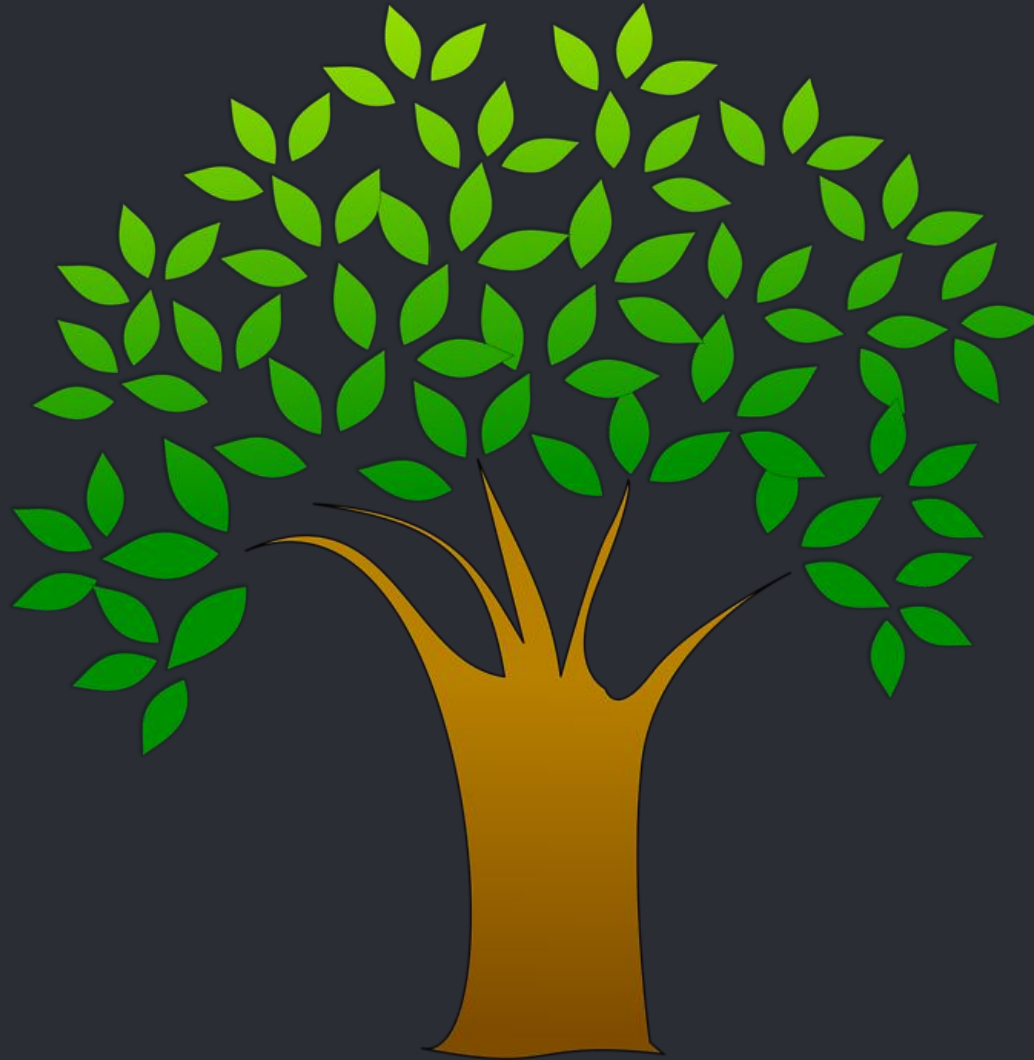
Data

[Data Set: Total Population by City](#)

[Today's Colab
Notebook](#)



Data Set
[Source](#)

We learn through failure & trial/error.



Questions?

Today's Notebooks

-  [Intro to Python Packages - NumPy](#) (Post-Class)
 - [Blank Version for Codealong](#)
-  [Loading & Exploring Data - Group Activity](#)