CS105 Lab 1 Introduction

Taghreed Alanazi

TA information

Name: Taghreed Alanazi

Email:

- talan002@ucr.edu
- taghreed.alanazi@email.ucr.edu

OH:

- OH reservation through this link: https://calendly.com/talan002/cs-105-oh
  - Wednesday 2:00 to 4:00 pm
  - Thursday 12:00 to 2:00 pm
- Room to meet: WCH 110

If you have any question or concern, please reach me out through **my email** or **Slack** (Direct message).

Lab work and grading procedure

- Lab attendance is **required**, 7 to 8 lab assignments.
- Student are highly encouraged to **team up** to finish the lab work (no more than 2 students in the group) or **individually**.
- Each group submits one assignment. Both students will receive the same credit (unless requested otherwise).

- For gaining scores of your lab work:
  - Directly demo the work to us.
  - Submitting files for grading.

- **NOTE:** you still need to submit you file online before the due day.

- **Each lab must be demoed.** If a student (or a group) fails to submit or demo the assignment, **he/she receives a "0".**

- **Late submission → 20% off.**

Demo

- Each group need to demo during the lab time (**the due day is on Tuesday**), or can come and show the work during OHs.

- During demo, **both group members** need to show up and contribute to answer questions and explanations.

- You **still need to submit your ipynb file and your pdf file online before the due date.**

- Scores are totally dependent on your demo. **You can get feedback immediately.**

How to demo and the programming language used

You can use Anacoda (check Anaconda Installation Guide pdf).

We will use → Python programming language.

- **Widely Used:** Python is a popular language for data analysis, machine learning, AI, web development, and more.
- **Simple and Readable:** Python is known for its clear and human-readable syntax.
  - No need for semicolons to end statements.
  - Lines end by themselves, making code more readable.
- **Dynamic Typing:** No need to declare variable types upfront.
- **Indentation Matters:** Unlike many languages, Python uses indentation to denote code blocks.
  - Ensure consistent whitespace for logical blocks of code.

Basic Python concepts

- **Data Types:**
  - Integers: Whole numbers (e.g., 5, -3, 0)
  - Floats: Decimal numbers (e.g., 3.14, -0.001, 5.0)
  - Strings: Text (e.g., 'Hello', "Python")
  - Booleans: True or False values (True, False)
- **Variables**:
  - Store and reference data by a name.
  - Example:

```
age = 25
name = "Alice"
```

- **Basic Operations:**
  - Arithmetic: +, -, *, /
  - Concatenation for strings: 'Hello' + ' World!'
  - Comparison: ==, !=, <, >, <=, >=
- **Print Function:**
  - Display values to the screen.
  - Example:

```
print("Hello, World!")
```

- **Lists:** Collection of items (can be of mixed types).
  - **Example:** `fruits = ["apple", "banana", "cherry"]`
- **Tuples:** Similar to lists but immutable (can't be changed after they're created).
  - **Example:** `coordinates = (4, 5)`
- **Dictionaries:** Key-value pairs.
  - **Example:** `person = {"name": "Alice", "age": 25}`
- **Control Structures:**
  - **If-Else Statements:** Conditional execution based on a test.
- **Loops:**
  - **For Loop:** Iterate over sequences (like lists or ranges).
  - **While Loop:** Execute as long as a condition remains true.
- **Functions:** Blocks of reusable code.

Examples:

```
# If-Else Statements example
age = 22
if age < 20:
    print("You're a teenager!")
else:
    print("You're an adult!")
```

```
You're an adult!
```

```
# For loop example
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
   print(fruit)
```

```
apple
banana
cherry
```

```
# Functions example
def greet(name):
    return "Hello, " + name + "!"

print(greet('Alice'))
```

```
Hello, Alice!
```

```
# While Loop example
count = 0
while count < 5:
    print(count)
    count += 1
```

```
0
1
2
3
4
```

In data analysis, we do:

- Data Cleaning:
  - Handling missing data.
  - Data transformation and normalization.
- Exploratory Data Analysis (EDA):
  - Understanding the data's main characteristics, often using statistical graphics, plots, and information tables.
- Data Preprocessing:
  - Feature engineering.
  - One-hot encoding, normalization, etc.
- … **and more!**
- We want to introduce **some libraries and tools** that are frequently used in Data analysis field.

Libraries and tools (with examples)

Numpy:
Fundamental package for scientific computing in Python.
Provides support for large multi-dimensional arrays and matrices.

```
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
arr
```

```
array([1, 2, 3, 4, 5])
```

Pandas:
Essential for data manipulation and analysis.
Provides data structures like DataFrame for handling and analyzing structured data.

```
import pandas as pd
data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
df = pd.DataFrame(data)
```

```
data
```

```
{'Name': ['Alice', 'Bob'], 'Age': [25, 30]}
```

```
df
```

	Name	Age
0	Alice	25
1	Bob	30

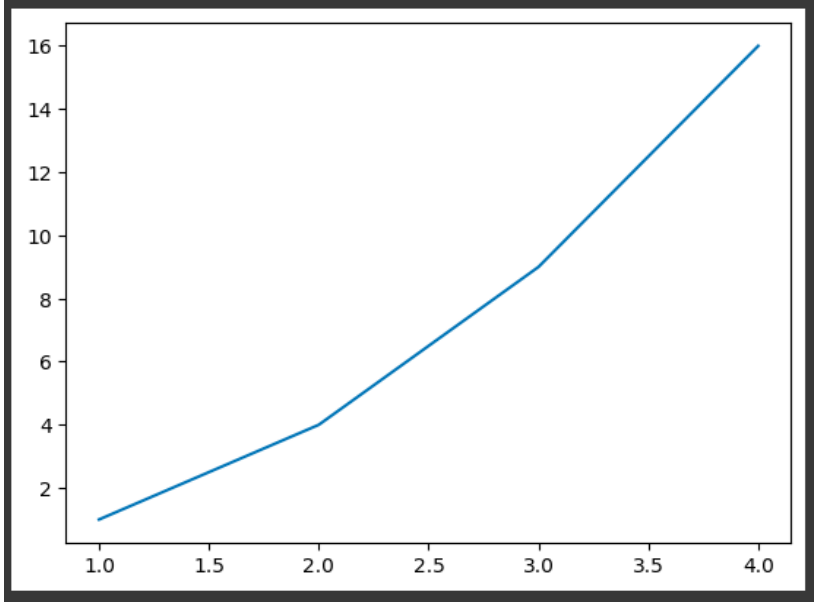
Libraries and tools (with examples)

Matplotlib & Seaborn:
Libraries for **data visualization.**
Plot graphs, histograms, scatter plots, etc.

Scipy:
Used for high-level computations.
Provides modules for optimization, integration, interpolation, eigenvalue problems, and more.

```
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()
```



Useful Python resources

You can learn more with these tutorials:

https://docs.python.org/3/tutorial/

Also, you can try Code Academy: https://www.codecademy.com/learn/learn-python-3

Hints of useful functions for Lab1

- **Question 0:** Give your best guess and explain your thought process behind your answer.
- **Question 1:** set_index(col)
- **Question 2:** Instead of using a for loop to change all the values of a column, we can use special functions offered by pandas that can modify entire columns at a time. For example, if we wanted to cast a column of floats (eg. df.col1) as integers, we might use the following line of code: df.col1 = df.col1.astype(int), also, we can visualize a series of numbers using value_counts() and plot().
- **Question 3:** Here we do the same thing as Q2, but we are extracting the last digit. An easy way to index the last digit is by using -1 (e.g. 'hello'[-1] returns o).
- **Question 4:** Here we again are doing something similar to Q2. (But which index should we be using?)

More examples about the functions please refer to the "Lab1 Examples_w23.ipynb".