Assignment 0 – Introduction to Python programming

ESO208 – Computational methods in engineering
Instructor: Balaji Devaraju [dbalaji]

2024-25 I
Issue: 31.07.2024
Deadline: 08.08.2024

Introduction

Python programming is a versatile and widely used programming both in scientific and non-scientific communities. It is a mature programming language and is open source. In this assignment, you will get to know about the fundamentals of Python programming. This includes defining variables, arithmetic calculations, loops, condition statements, basic matrix operations and visualization. You can refer to one of the Python programming books for further help (Langtangen, 2016; Kong et al., 2020; Kiusalaas, 2014).

Google Colab

The programming exercises should be executed on Google Colab. Please register at the colab.google website if you are new to Google Colab. The Colab platform allows you to generate *notebooks*, which is where you will be writing the code. Once you are done with the exercises given in the assignment sheet, please share the link of the notebook in the mooKIT platform. A detailed description of this will be discussed in the class by your tutors.

Basics

Declaring variables

1. How do you print `Hello World` in Python?

2. Declare the variables `x`, `y`, and `z` as having the values `5`, `Thomas`, and `3.14`.

3. What do you infer about their data types?

Functions

1. Write a function to add two numbers.

2. Write a function to print out the product of three numbers given as inputs.

3. Write a function to print out the bill of three commodities which are given as 500 g of oranges costing Rs.300, 250 g of grapes costing Rs.230 and 1 kg of apples costing Rs.215.

Conditional statements

1. Take two numbers $a = 10$ and $b = 20$. Applying `if-else` conditional statements, print that b is greater than a.

2. If $a = 10$ and $b = 10$, print if a is greater than or equal to b.

3. If $a = 10$, $b = 20$ and $c = 30$, using nested if statements, print that c is greater than both a and b.

Loops

For loops

1. Create a variable `dimensions` that contains a list of the three words: length, breadth and height.
2. Loop through the letters in the word `breadth` and print the letters on the screen to form the word again.
3. Assign the values as 10 cm for length, 5 cm for breadth and 3 cm for height.
4. In a given array of numbers from 1 to 30, print out all numbers with a gap of 3 numbers using for loop.
5. Using nested for loops, create an array of any three fruit and another array assigning adjectives to the three given fruit.
6. Using nested for loops, create a multiplication table list of all natural numbers from 1 to 10 (i.e multiplication table for 1 from 1 to 10, multiplication table for 2 from 1 to 10, multiplication table for 3 from 1 to 10, and so on up till 10).

While loops

1. Print variable i as long as i is less than 6.
2. Exit the loop when i is 3.
3. If a ball is thrown from a height of 10m and it continuously bounces reducing by a factor of 0.5 (i.e. 10m, 5m, 2.5m,1.25m and so on), then using while loop, write a program till the height is above 0.10 m.

Data frames and Appending

Data frames

1. Generate a data frame with the following data:
   - 400, 358, 263, 128, 100 mm of rainfall for 5 days consecutively
   - 43, 39, 29, 33, 40 °C being the corresponding temperature
   - 15, 22, 28, 30, 25 ms^{-1} being the corresponding wind speed
2. Generate a data frame to read the file: rain_data.csv. Generate a line diagram plot with dates on the x-axis and the rainfall amount on the y-axis.

Appending

Take four variables: Precipitation, Wind Speed, Temperature, and Soil moisture. Generate an array of these four variables. Then add another variable in the array named as `gravity_anomalies`. Lastly, assign the values 300 mm, 25 ms^{-1}, 30 °C, 18 $\mathrm{m^3/m^3}$ and 0.015 Gal (a unit of gravity) and visualize them in a data frame.

Matrix operations

1. Create a list of sizes 100 ranging from 0 to 1000. Convert this list in numpy array and then form a row and column vectors of size 1×100 and 100×1 from this array, respectively and display the transpose of each of them.
2. Create a matrix of size 100×100. Extract two sub-matrices of size 50×50 and 25×25.

3. From the above two sub-matrices, get the third row and fourth column from one matrix and sort the other matrix both row and column-wise.

4. Find the rank and condition number of any sub-matrix and then save the rank and condition number in a dictionary. Call rank and condition numbers from this dictionary and add them.

5. Find minimum, maximum, mean, and standard deviation and vectorize a matrix.

6. Create two random matrices, with values ranging from 1: 500, of size 1000×1000 and 100×100. Use a for loop to compute the square of each element greater than 100 and the cube if the element is less than 100.

7. Write a function for factorial using *while* loop. Find the factorial of the prime number between 0 to 100 and store it in a list.

8. The Singular Value Decomposition of $m \times n$ matrix A is given by the formula:

   $$A = U\Sigma V^T$$

   where:

   - U: $m \times n$ matrix of the orthonormal eigenvectors of AA^T.
   - V^T: transpose of a $n \times n$ matrix containing the orthonormal eigenvectors of A^TA.
   - Σ: $n \times n$ diagonal matrix of the singular values which are the square roots of the eigenvalues of A^TA.

   Define a 3×4 matrix of random numbers between 0 and 100. Perform SVD Analysis and print U, V^T, and Σ. Check the orthogonality of U and V. Compute the pseudo-inverse of A from the SVD.

Data analysis and visualization

Primary objective

The main objective of this exercise is to read the data from a given file, make plots of the data and compute the relevant statistics.

Datasets

In this exercise, you are given average global land temperature data from 1750 to 2021 produced by the Berkeley Averaging method. This data contains monthly average surface temperature anomalies and uncertainties. It also contains corresponding values for year, five-year, ten-year, and twenty-year moving averages.

1. Access the land temperature time series data using the following link: Complete_TAVG_complete_Copy.csv.

2. Parse the csv file and get the header information from the data set.

3. Compute mean, median, mode, standard deviation and variance for all the time-series.

4. Plot the time series of monthly, annual, five year, and ten year temperature anomaly data. Also, plot all the time series plots as subplots.

5. Plot the time series of anomalies with their uncertainties with error bars.

6. Plot histogram, box plot, and violin plot for monthly and annual anomaly.

7. Regional temperature data is also available on Berkeley Earth website. The following link can be used to download temperature anomaly data for New Delhi: Delhi_TAVG_Trend_Copy.csv. Plot a scatter plot of the New Delhi time series with the global time series using monthly and annual data.

8. Visualize the monthly average of temperature data for the last 100 years for the two different cities - Kanpur and New Delhi (using their corresponding csv files). Compare the statistics of the two cities. Use the following link for accessing the data: Kanpur and New Delhi.

References

Kiusalaas, J (2014). *Numerical Methods in Engineering with Python 3.* Cambridge University Press.

Kong, Q, Siauw, T and Bayen, A (2020). *Python Programming and Numerical Methods: A Guide for Engineers and Scientists.* Elsevier Science.

Langtangen, HP (2016). *A primer on scientific programming with Python.* Texts in Computational Science and Engineering, 5th edn. Springer, Berlin, Germany.