Lab Assignment 29

Student Name: Chauhan Vandana Ramdayal

Student Id: AF0411629

Topic: Scipy Cluster and constant

In the context of SciPy, clustering and constants are two distinct areas. Here's a brief overview of both:

1. SciPy Clustering

SciPy provides various clustering algorithms under the scipy.cluster module. Some commonly used clustering techniques include:

- **K-means Clustering**: A method to partition a dataset into K clusters, where each data point belongs to the cluster with the nearest mean.
- **Hierarchical Clustering**: This includes methods for agglomerative clustering, which builds a hierarchy of clusters.
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**: A clustering algorithm that groups together points that are close to each other based on a distance measurement and a minimum number of points.

2. SciPy Constants

The scipy.constants module provides a collection of physical and mathematical constants, which can be used in scientific computations. Some of the constants included are:

- **Speed of light (c)**: The speed of light in vacuum.
- **Gravitational constant (G)**: The constant of gravitation.
- **Planck's constant (h)**: A fundamental constant in quantum mechanics.

Questions:

Q1: Convert inches to centimetre

https://raw.githubusercontent.com/AnudipAE/DANLC/master/people_heights.csv

Code:

```
import pandas as pd
from scipy.constants import inch, centi

# Load the dataset
df = pd.read_csv('https://raw.githubusercontent.com/AnudipAE/DANLC/master/people_heights.csv')

# Display the original DataFrame
print(df)

# Convert Height (inches) to Height (cm) and create a new column
df['Height (cm)'] = df['Height (inches)'] * inch / centi

# Display the updated DataFrame
print(df)
```

Output:

```
          Name  Height (inches)
0     Person 1            60.03
1     Person 2            49.51
2     Person 3            82.97
3     Person 4            64.19
4     Person 5            54.42
..         ...              ...
95   Person 96            76.69
96   Person 97            68.06
97   Person 98            57.89
98   Person 99            63.56
99  Person 100            81.85

[100 rows x 2 columns]
          Name  Height (inches)  Height (cm)
0     Person 1            60.03     152.4762
1     Person 2            49.51     125.7554
2     Person 3            82.97     210.7438
3     Person 4            64.19     163.0426
4     Person 5            54.42     138.2268
..         ...              ...          ...
95   Person 96            76.69     194.7926
96   Person 97            68.06     172.8724
97   Person 98            57.89     147.0406
98   Person 99            63.56     161.4424
99  Person 100            81.85     207.8990

[100 rows x 3 columns]
```

Q2: Convert Giga Byte to Mega Byte

https://raw.githubusercontent.com/AnudipAE/DANLC/master/file_size.csv

Code:

```
import pandas as pd
from scipy.constants import giga, mega

# Load the dataset
df = pd.read_csv('https://raw.githubusercontent.com/AnudipAE/DANLC/master/file_size.csv')

# Display the first few rows of the DataFrame
print(df.head())

# Convert Size (GB) to Size (MB) and create a new column
df['Size (MB)'] = df['Size (GB)'] * giga / mega

# Display the updated DataFrame
print(df)
```

Output:

```
     Filename  Size (GB)
0  file_1.txt       9.72
1  file_2.txt       9.81
2  file_3.txt       5.61
3  file_4.txt       4.58
4  file_5.txt       5.52
        Filename  Size (GB)  Size (MB)
0     file_1.txt       9.72     9720.0
1     file_2.txt       9.81     9810.0
2     file_3.txt       5.61     5610.0
3     file_4.txt       4.58     4580.0
4     file_5.txt       5.52     5520.0
..           ...        ...        ...
95   file_96.txt       1.29     1290.0
96   file_97.txt       7.11     7110.0
97   file_98.txt       4.86     4860.0
98   file_99.txt       7.89     7890.0
99  file_100.txt       5.52     5520.0

[100 rows x 3 columns]
```