National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Electrical Engineering

Faculty Member: Ma'am Neelma Naz

Date: October 8, 2025

Semester: 7th

Group: Gp-02

CS471 Machine Learning

Lab 4: Introduction to Pandas and MatplotLib

		PLO4	**PLO5**	**PLO5**	**PLO8**	**PLO9**
		CLO4	CLO5	CLO5	CLO6	CLO7
Student Name	**Reg. No**	**Viva / Quiz / Demo** 5 Marks	**Analysis of Data in Report** 5 Marks	**Modern Tool Usage** 5 Marks	**Ethics** 5 Marks	**Individual and Teamwork** 5 Marks
Hanzla Sajjad	403214					
Irfa Farooq	412564					

Introduction

This laboratory exercise is focused on handling and visualizing datasets for machine learning purposes. In any machine learning task, we are working with data. For dataset handling, we use the Pandas library which can load .csv files into a data frame. During machine learning, we also need to make plots. For this, we make use of the PyPlot submodule in the MatplotLib library.

Objectives

The following are the main objectives of this lab:

- Load dataset into a python program environment
- Analyze dataset using the Pandas module
- Perform any needed cleaning of the dataset
- Draw line plots in python for dataset analysis
- Draw scatter plots in python for dataset analysis

Lab Conduct

- Respect faculty and peers through speech and actions
- The lab faculty will be available to assist the students. In case some aspect of the lab experiment is not understood, the students are advised to seek help from the faculty.
- In the tasks, there are commented lines such as #YOUR CODE STARTS HERE# where you have to provide the code. You must put the code between the #START and #END parts of these commented lines. Do NOT remove the commented lines.
- Use the tab key to provide the indentation in python.
- Upon completing the lab, you must delete the manual from the lab computer

Theory

Pandas (panel data) is a library that can load tabular data from .csv files and store into a NumPy compatible table known as a “Pandas Data Frame”. Each column in a data frame is of a “Pandas Series” type. Aside from loading datasets, pandas also enables us to perform basic mean, mode, median operations as well as clean up incomplete or duplicate data.

MatplotLib is another library focused on data visualization. It contains many functions for displaying plots, subplots, scatter plots etc. Line plots are used widely for monitoring training accuracies and losses. Scatter plots are used mainly for modeling the feature space of the dataset.

A brief summary of the list functions in python is provided below:

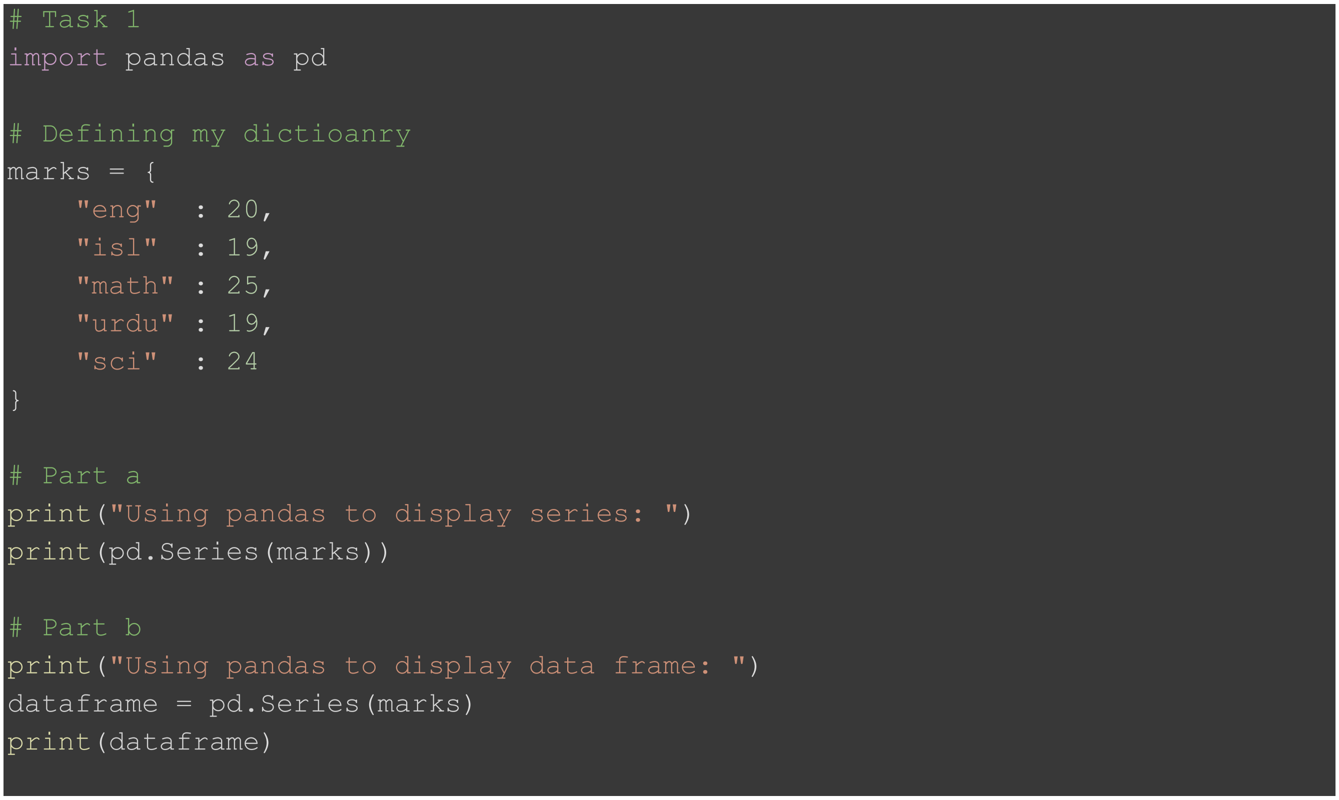
append(I)	append item I to the end of the list
insert(i, I)	insert item I at i position of the list
extend(L)	extend/concatenate a second list L
remove(I)	remove a specified item I from a list
pop(i)	remove item at specific index i in the list
count(I)	return total number of a specific item I from a list
index(I)	return index of first occurrence of a specific item I
reverse	reverse the items of the list

For this lab, you will be provided with some dataset files (in .csv format) which you will need for the tasks. Additionally, for the final task, you will need to arrange your own dataset by downloading it from the internet. You will need to import pandas and matplotlib.pyplot for this lab.

Lab Task 1 – Pandas Series and Dataframes

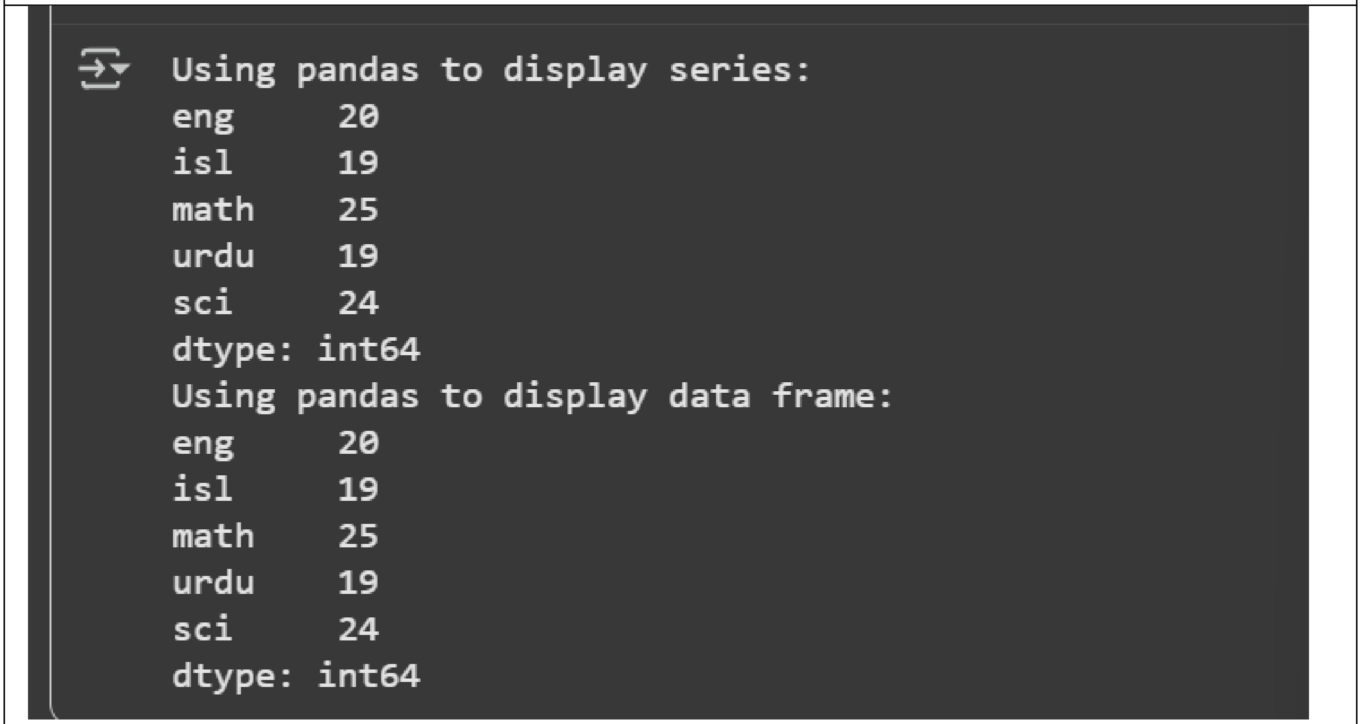
a) Create a Pandas series using a dictionary and display the output.
b) Create a Pandas dataframe using a dictionary and display the output.
Provide all of the codes and screenshots of the final outputs.

Code



```
# Task 1
import pandas as pd

# Defining my dictioanry
marks = {
    "eng"  : 20,
    "isl"  : 19,
    "math" : 25,
    "urdu" : 19,
    "sci"  : 24
}

# Part a
print("Using pandas to display series: ")
print(pd.Series(marks))

# Part b
print("Using pandas to display data frame: ")
dataframe = pd.Series(marks)
print(dataframe)
```

Output Console



```
Using pandas to display series:
eng     20
isl     19
math    25
urdu    19
sci     24
dtype: int64
Using pandas to display data frame:
eng     20
isl     19
math    25
urdu    19
sci     24
dtype: int64
```

Lab Task 2 – CSV Files

Load dataset 1 into a dataframe and perform the following

a) Print the dataset using the head and tail functions

b) Print any 3 rows from the dataset

c) Print any 5 elements from the dataset

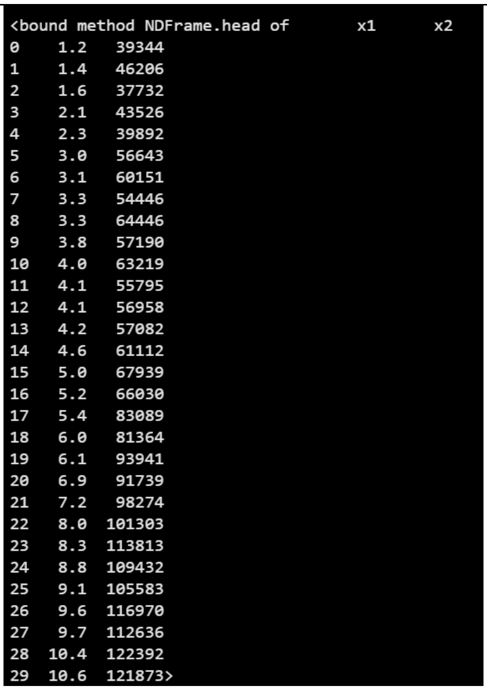
d) Use the mean, mode and median functions for each column in the dataset

Provide all of the codes and screenshots of the final output.

Code

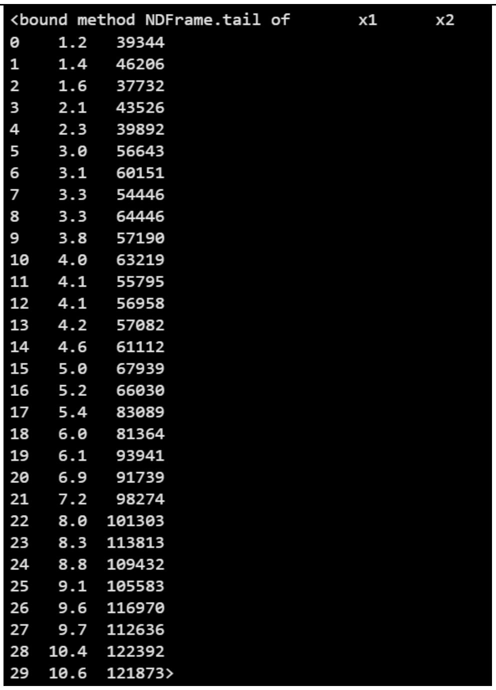
```
# Task 2
import pandas as pd

# Laoding csv files
df = pd.read_csv('lab4_dataset1 (1).csv')

# Part a
print(df.head)
print(df.tail)

# Part b
print(df.head(3))

# Part c
print(df.head(5))

# Pard d
print("Mean of column x1: ", df["x1"].mean())
print("Median of column x2: ", df["x2"].median())
print("Mode of column x1: ", df["x1"].mode())
```

Output Console



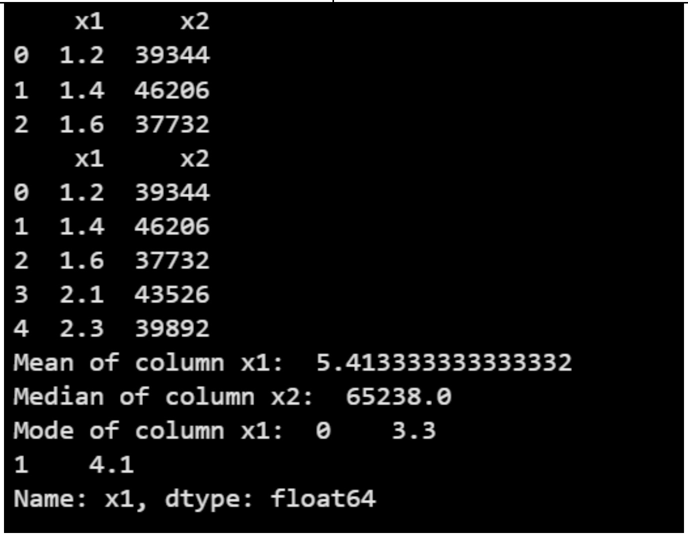
```
<bound method NDFrame.head of       x1      x2
0    1.2   39344
1    1.4   46206
2    1.6   37732
3    2.1   43526
4    2.3   39892
5    3.0   56643
6    3.1   60151
7    3.3   54446
8    3.3   64446
9    3.8   57190
10   4.0   63219
11   4.1   55795
12   4.1   56958
13   4.2   57082
14   4.6   61112
15   5.0   67939
16   5.2   66030
17   5.4   83089
18   6.0   81364
19   6.1   93941
20   6.9   91739
21   7.2   98274
22   8.0  101303
23   8.3  113813
24   8.8  109432
25   9.1  105583
26   9.6  116970
27   9.7  112636
28  10.4  122392
29  10.6  121873>
```

Figure 1: Printing through head function



```
<bound method NDFrame.tail of       x1      x2
0    1.2   39344
1    1.4   46206
2    1.6   37732
3    2.1   43526
4    2.3   39892
5    3.0   56643
6    3.1   60151
7    3.3   54446
8    3.3   64446
9    3.8   57190
10   4.0   63219
11   4.1   55795
12   4.1   56958
13   4.2   57082
14   4.6   61112
15   5.0   67939
16   5.2   66030
17   5.4   83089
18   6.0   81364
19   6.1   93941
20   6.9   91739
21   7.2   98274
22   8.0  101303
23   8.3  113813
24   8.8  109432
25   9.1  105583
26   9.6  116970
27   9.7  112636
28  10.4  122392
29  10.6  121873>
```

Figure 2: Printing through tail function



```
    x1     x2
0  1.2  39344
1  1.4  46206
2  1.6  37732
    x1     x2
0  1.2  39344
1  1.4  46206
2  1.6  37732
3  2.1  43526
4  2.3  39892
Mean of column x1:  5.413333333333332
Median of column x2:  65238.0
Mode of column x1:  0    3.3
1    4.1
Name: x1, dtype: float64
```

Figure 3: Printing elements and mean, median, mode

Lab Task 3 – Dataset Cleaning

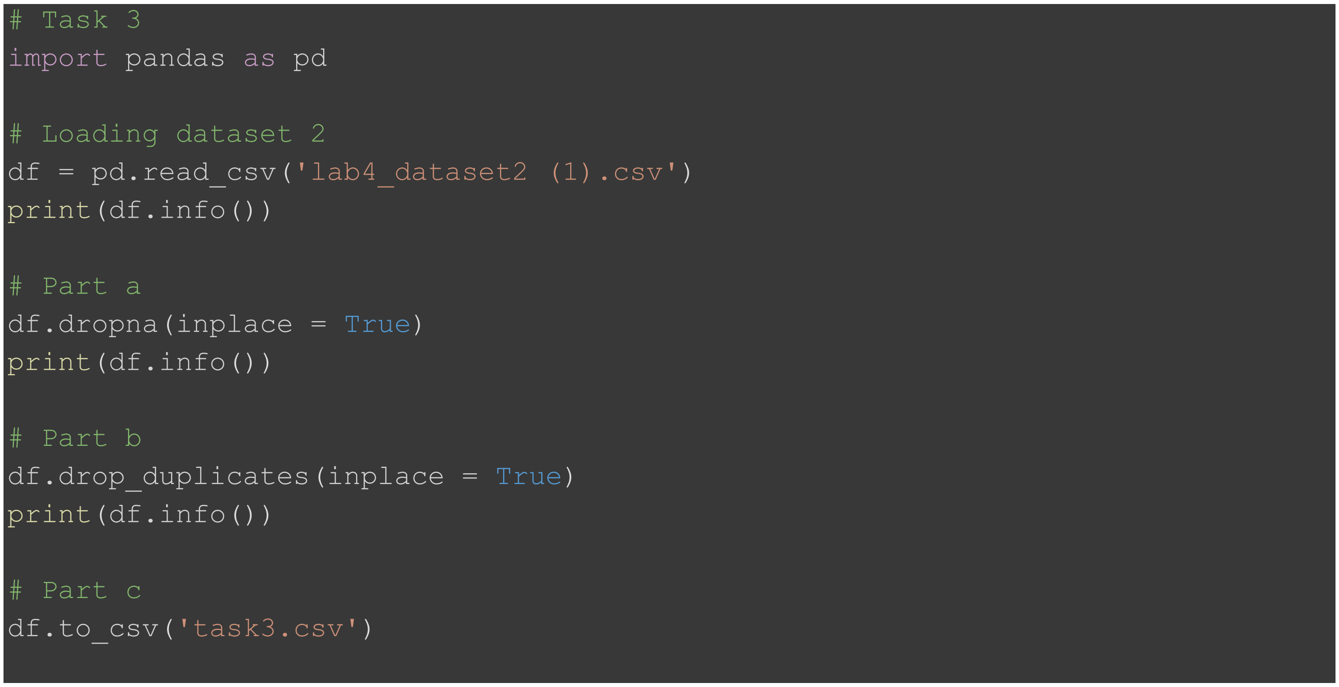
Load dataset 2 into a dataframe.

a) Write code to remove the incomplete rows from the dataset

b) Write code to remove the duplicated rows from the dataset

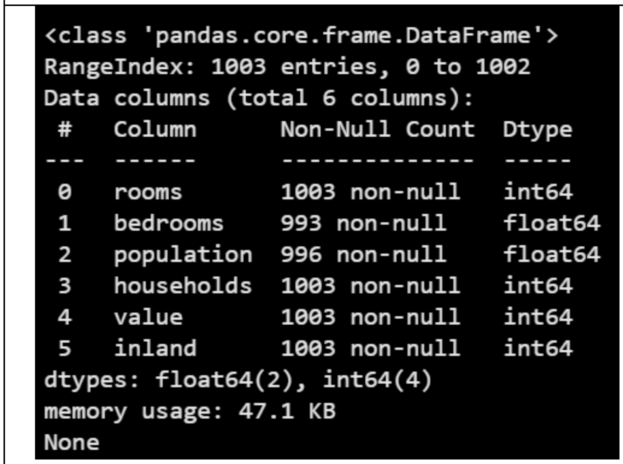
c) Save the cleaned dataset into a dataframe. You need to attach this cleaned dataset file (renamed to task3.csv) in your lab submission.

Code



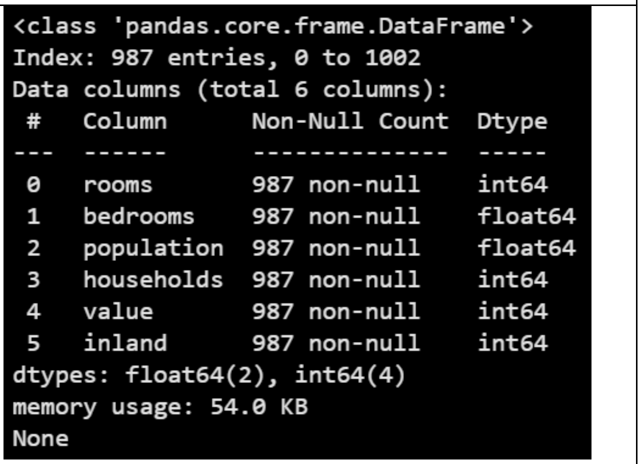
```python
# Task 3
import pandas as pd

# Loading dataset 2
df = pd.read_csv('lab4_dataset2 (1).csv')
print(df.info())

# Part a
df.dropna(inplace = True)
print(df.info())

# Part b
df.drop_duplicates(inplace = True)
print(df.info())

# Part c
df.to_csv('task3.csv')
```

Output Console



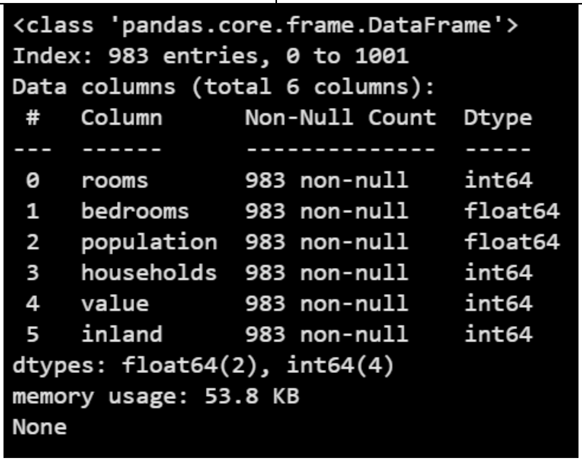
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1003 entries, 0 to 1002
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   rooms       1003 non-null   int64
 1   bedrooms    993 non-null    float64
 2   population  996 non-null    float64
 3   households  1003 non-null   int64
 4   value       1003 non-null   int64
 5   inland      1003 non-null   int64
dtypes: float64(2), int64(4)
memory usage: 47.1 KB
None
```

Figure 4: After loading the dataset



```
<class 'pandas.core.frame.DataFrame'>
Index: 987 entries, 0 to 1002
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   rooms       987 non-null    int64
 1   bedrooms    987 non-null    float64
 2   population  987 non-null    float64
 3   households  987 non-null    int64
 4   value       987 non-null    int64
 5   inland      987 non-null    int64
dtypes: float64(2), int64(4)
memory usage: 54.0 KB
None
```

Figure 5: After dropping empthy columns



```
<class 'pandas.core.frame.DataFrame'>
Index: 983 entries, 0 to 1001
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   rooms       983 non-null    int64
 1   bedrooms    983 non-null    float64
 2   population  983 non-null    float64
 3   households  983 non-null    int64
 4   value       983 non-null    int64
 5   inland      983 non-null    int64
dtypes: float64(2), int64(4)
memory usage: 53.8 KB
None
```

Figure 6: After removing duplicate columns

Lab Task 4 – Line and Scatter Plots

For this task, you will need to use datasets 1 and 2. You will also require the matplotlib.pyplot module for plotting. Perform the following.

a) Make line plots of the following equations for x = 1 to 100. You will need to make use of NumPy arrays for this part.

   i. $y = 2x + 1$

   ii. $y = 3x^2$

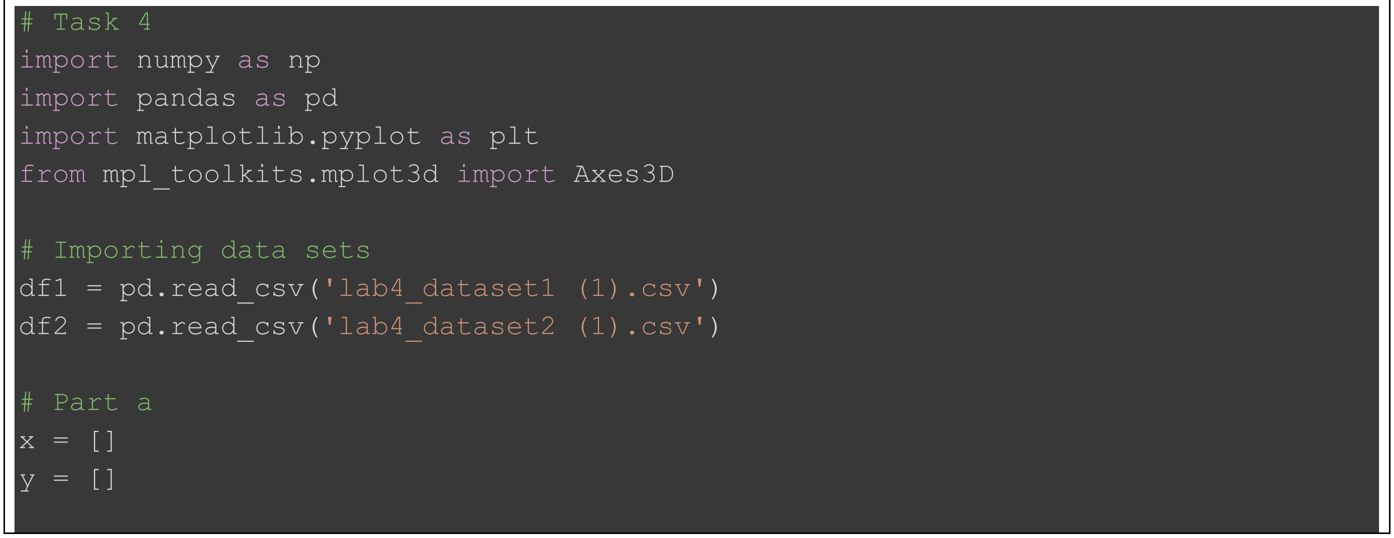
   iii. $y = \cos(x) + 2\sin(x-45)$

b) Load dataset 1 and make a scatter plot (axes x1 and x2)

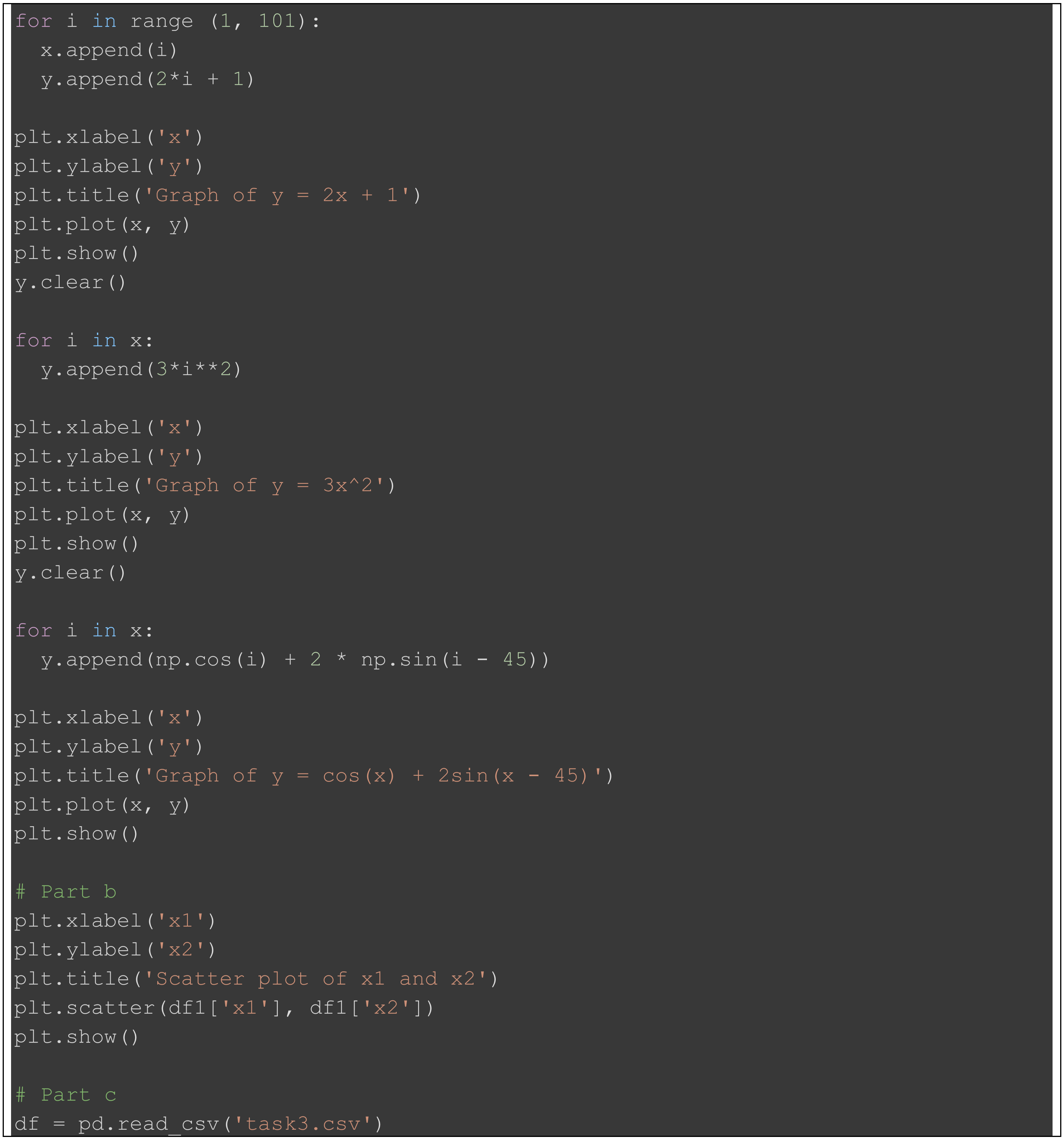
c) Load dataset 2 (cleaned version) and make a scatter plot (2 columns as axes). You need to use markers for the labels (y) such that 0 corresponds to a red circle and 1 corresponds to a blue square. The label y is the “inland” column. For x1 and x2, choose any 2 columns from the dataset and also mention the columns that you are using.

d) Load dataset 2 (cleaned version) and make a 3-D scatter plot between any three features in the dataset (axes x1, x2, x3). Specify the features that you use in your plot.

Code



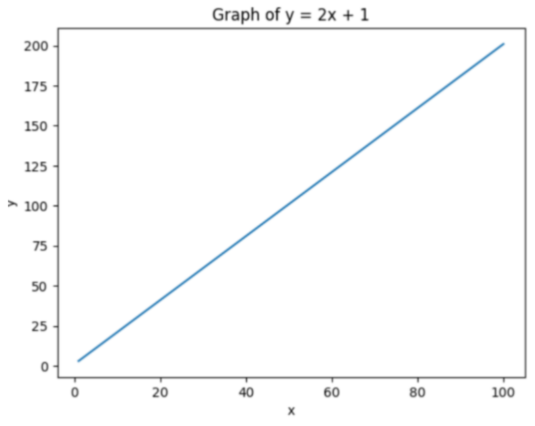
```
# Task 4
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Importing data sets
df1 = pd.read_csv('lab4_dataset1 (1).csv')
df2 = pd.read_csv('lab4_dataset2 (1).csv')

# Part a
x = []
y = []
```

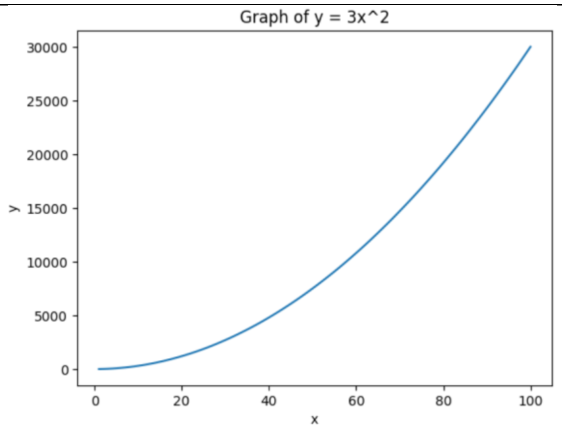


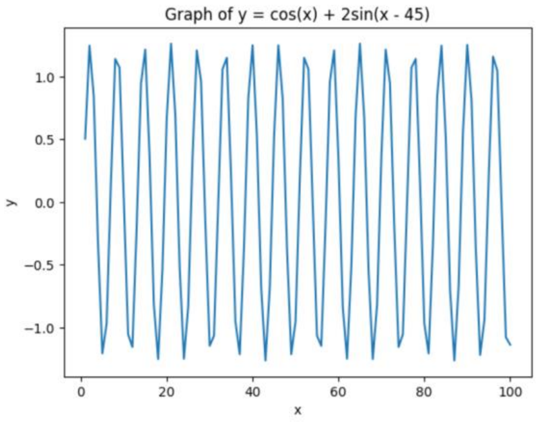
```
for i in range (1, 101):
  x.append(i)
  y.append(2*i + 1)

plt.xlabel('x')
plt.ylabel('y')
plt.title('Graph of y = 2x + 1')
plt.plot(x, y)
plt.show()
y.clear()

for i in x:
  y.append(3*i**2)

plt.xlabel('x')
plt.ylabel('y')
plt.title('Graph of y = 3x^2')
plt.plot(x, y)
plt.show()
y.clear()

for i in x:
  y.append(np.cos(i) + 2 * np.sin(i - 45))

plt.xlabel('x')
plt.ylabel('y')
plt.title('Graph of y = cos(x) + 2sin(x - 45)')
plt.plot(x, y)
plt.show()

# Part b
plt.xlabel('x1')
plt.ylabel('x2')
plt.title('Scatter plot of x1 and x2')
plt.scatter(df1['x1'], df1['x2'])
plt.show()

# Part c
df = pd.read_csv('task3.csv')
```

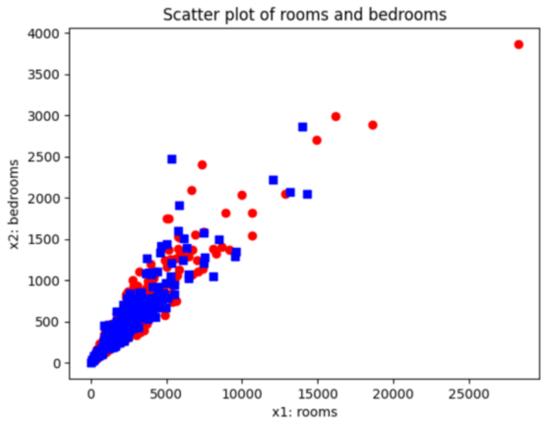
```
plt.xlabel('x1: rooms')
plt.ylabel('x2: bedrooms')
plt.title('Scatter plot of rooms and bedrooms')
plt.scatter(df[df["inland"] == 0]["rooms"],
            df[df["inland"] == 0]["bedrooms"],
            color = 'red', marker = 'o', label = 'inland = 0')
plt.scatter(df[df["inland"] == 1]["rooms"],
            df[df["inland"] == 1]["bedrooms"],
            color = 'blue', marker = 's', label = 'inland = 1')
plt.show()

# Part d
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

ax.scatter(df["rooms"], df["bedrooms"], df["households"], color='green', 
marker='o')

ax.set_xlabel('Rooms')
ax.set_ylabel('Bedrooms')
ax.set_zlabel('Households')
plt.title('Scatter plot of Rooms, Bedrooms, and Households')
```

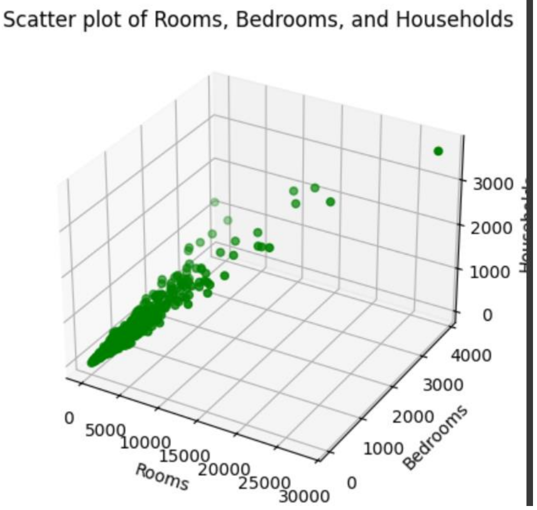
Output Console







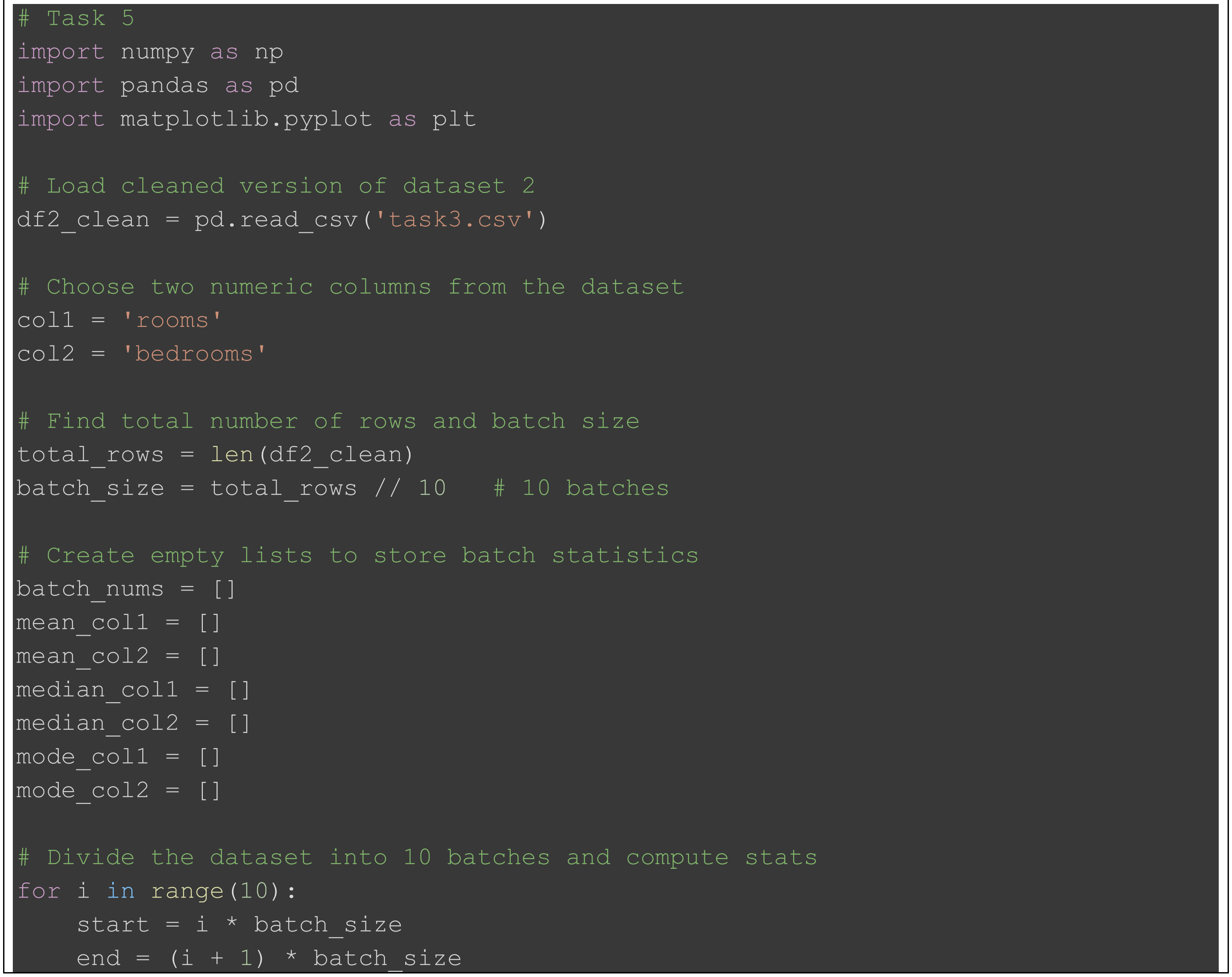




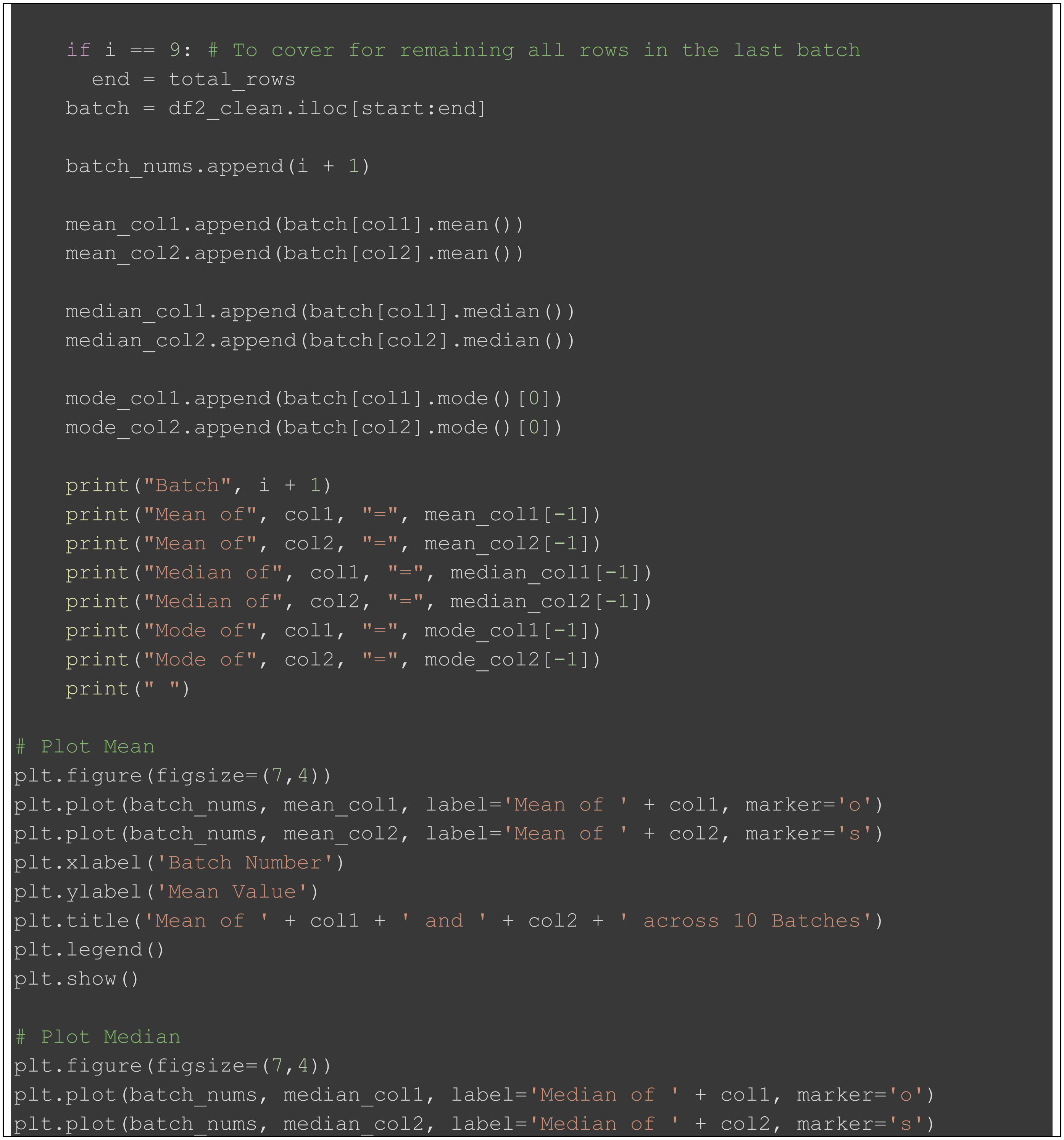
Lab Task 5 – Dataset Batches

Load the cleaned version of dataset 2 into a dataframe. For this task, you will divide the dataset examples into 10 batches. For each individual batch, calculate the mean, mode and median for any two columns of the dataset. Finally, make line plots showing the batch number on the x-axis and the mean, mode and median on the y-axis.

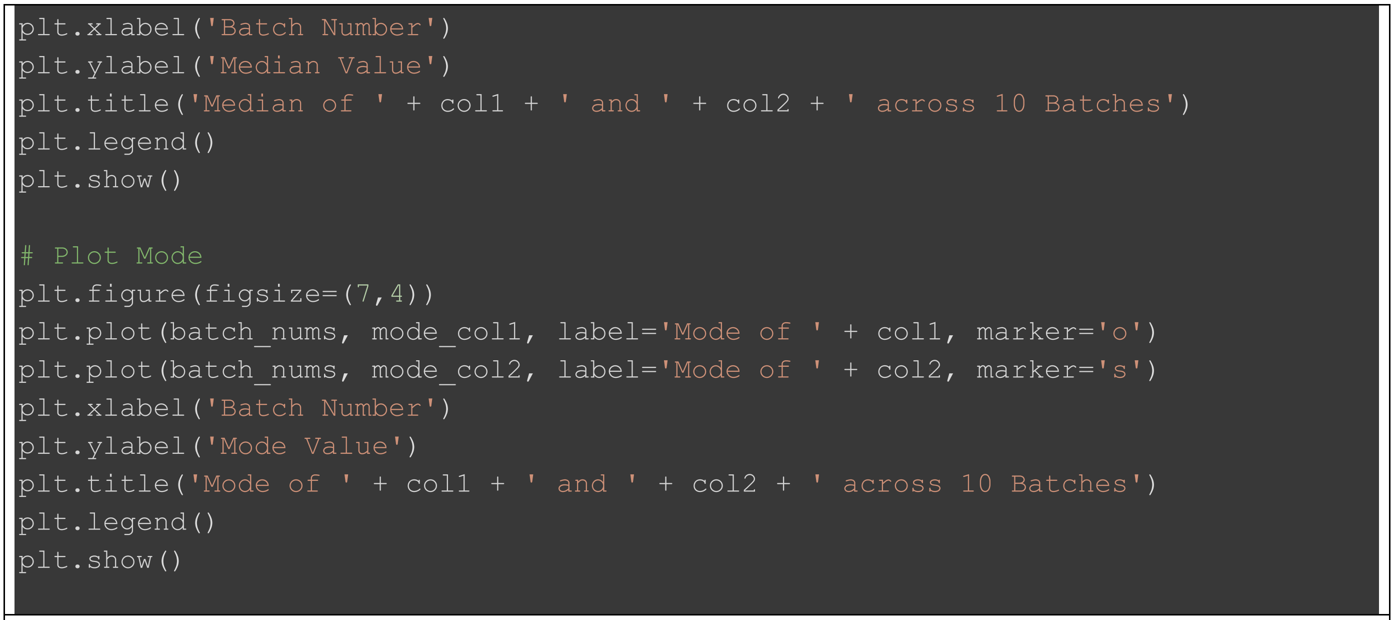
Code



```python
# Task 5
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Load cleaned version of dataset 2
df2_clean = pd.read_csv('task3.csv')

# Choose two numeric columns from the dataset
col1 = 'rooms'
col2 = 'bedrooms'

# Find total number of rows and batch size
total_rows = len(df2_clean)
batch_size = total_rows // 10   # 10 batches

# Create empty lists to store batch statistics
batch_nums = []
mean_col1 = []
mean_col2 = []
median_col1 = []
median_col2 = []
mode_col1 = []
mode_col2 = []

# Divide the dataset into 10 batches and compute stats
for i in range(10):
    start = i * batch_size
    end = (i + 1) * batch_size
```

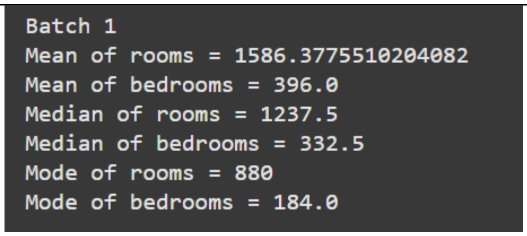


```
    if i == 9: # To cover for remaining all rows in the last batch
      end = total_rows
    batch = df2_clean.iloc[start:end]

    batch_nums.append(i + 1)

    mean_col1.append(batch[col1].mean())
    mean_col2.append(batch[col2].mean())

    median_col1.append(batch[col1].median())
    median_col2.append(batch[col2].median())

    mode_col1.append(batch[col1].mode()[0])
    mode_col2.append(batch[col2].mode()[0])

    print("Batch", i + 1)
    print("Mean of", col1, "=", mean_col1[-1])
    print("Mean of", col2, "=", mean_col2[-1])
    print("Median of", col1, "=", median_col1[-1])
    print("Median of", col2, "=", median_col2[-1])
    print("Mode of", col1, "=", mode_col1[-1])
    print("Mode of", col2, "=", mode_col2[-1])
    print(" ")

#  Plot Mean
plt.figure(figsize=(7,4))
plt.plot(batch_nums, mean_col1, label='Mean of ' + col1, marker='o')
plt.plot(batch_nums, mean_col2, label='Mean of ' + col2, marker='s')
plt.xlabel('Batch Number')
plt.ylabel('Mean Value')
plt.title('Mean of ' + col1 + ' and ' + col2 + ' across 10 Batches')
plt.legend()
plt.show()

#  Plot Median
plt.figure(figsize=(7,4))
plt.plot(batch_nums, median_col1, label='Median of ' + col1, marker='o')
plt.plot(batch_nums, median_col2, label='Median of ' + col2, marker='s')
```

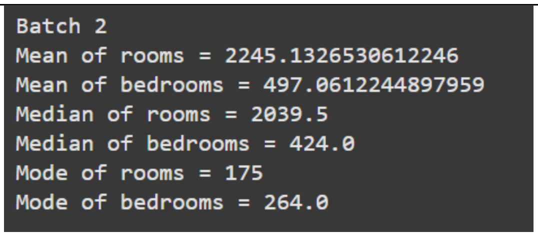


```
plt.xlabel('Batch Number')
plt.ylabel('Median Value')
plt.title('Median of ' + col1 + ' and ' + col2 + ' across 10 Batches')
plt.legend()
plt.show()

# Plot Mode
plt.figure(figsize=(7,4))
plt.plot(batch_nums, mode_col1, label='Mode of ' + col1, marker='o')
plt.plot(batch_nums, mode_col2, label='Mode of ' + col2, marker='s')
plt.xlabel('Batch Number')
plt.ylabel('Mode Value')
plt.title('Mode of ' + col1 + ' and ' + col2 + ' across 10 Batches')
plt.legend()
plt.show()
```

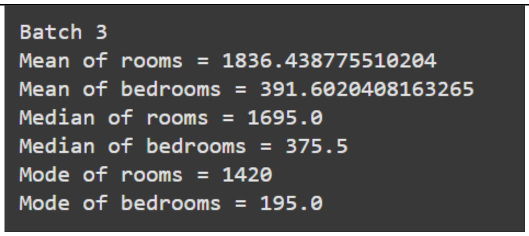
Output Console



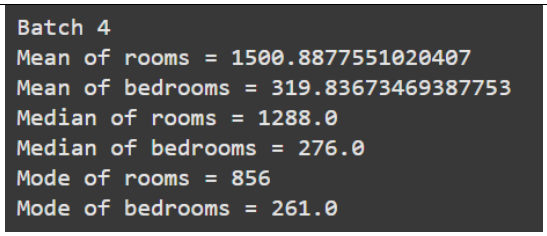
```
Batch 1
Mean of rooms = 1586.3775510204082
Mean of bedrooms = 396.0
Median of rooms = 1237.5
Median of bedrooms = 332.5
Mode of rooms = 880
Mode of bedrooms = 184.0
```



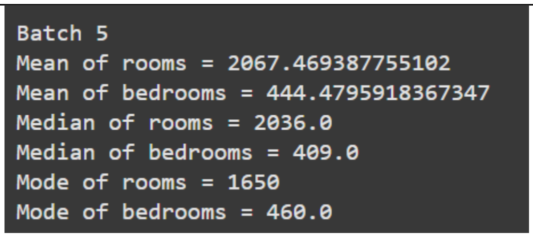
```
Batch 2
Mean of rooms = 2245.1326530612246
Mean of bedrooms = 497.0612244897959
Median of rooms = 2039.5
Median of bedrooms = 424.0
Mode of rooms = 175
Mode of bedrooms = 264.0
```



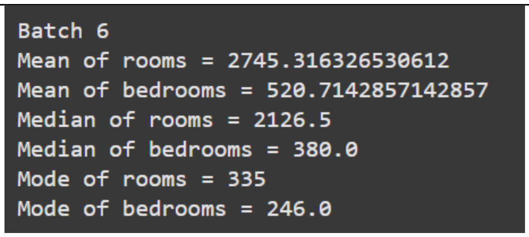
```
Batch 3
Mean of rooms = 1836.438775510204
Mean of bedrooms = 391.6020408163265
Median of rooms = 1695.0
Median of bedrooms = 375.5
Mode of rooms = 1420
Mode of bedrooms = 195.0
```



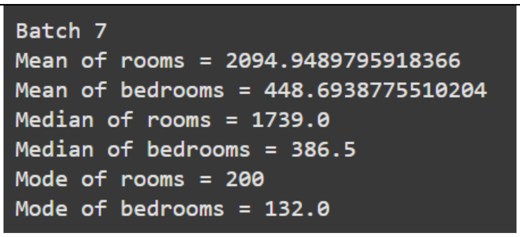
```
Batch 4
Mean of rooms = 1500.8877551020407
Mean of bedrooms = 319.83673469387753
Median of rooms = 1288.0
Median of bedrooms = 276.0
Mode of rooms = 856
Mode of bedrooms = 261.0
```



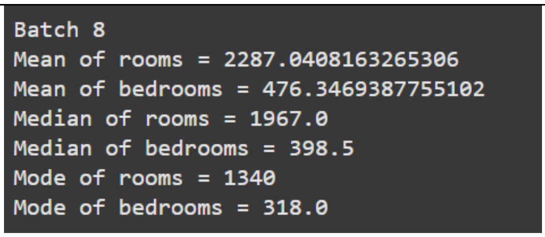
```
Batch 5
Mean of rooms = 2067.469387755102
Mean of bedrooms = 444.4795918367347
Median of rooms = 2036.0
Median of bedrooms = 409.0
Mode of rooms = 1650
Mode of bedrooms = 460.0
```



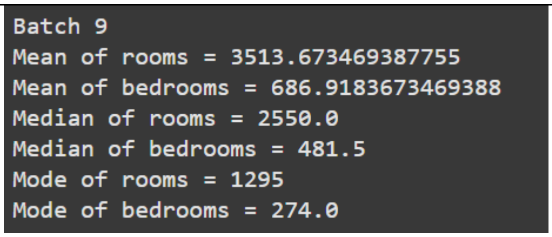
```
Batch 6
Mean of rooms = 2745.316326530612
Mean of bedrooms = 520.7142857142857
Median of rooms = 2126.5
Median of bedrooms = 380.0
Mode of rooms = 335
Mode of bedrooms = 246.0
```



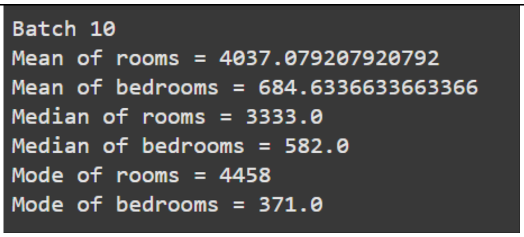
```
Batch 7
Mean of rooms = 2094.9489795918366
Mean of bedrooms = 448.6938775510204
Median of rooms = 1739.0
Median of bedrooms = 386.5
Mode of rooms = 200
Mode of bedrooms = 132.0
```



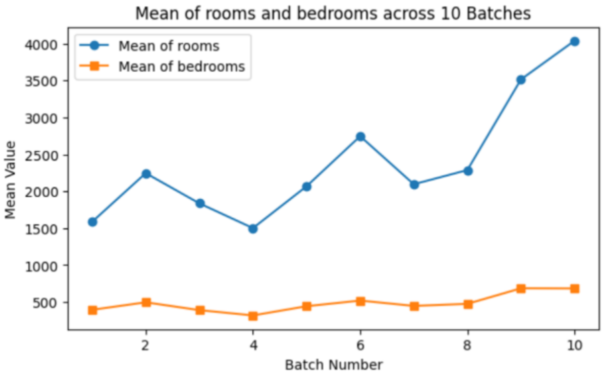
```
Batch 8
Mean of rooms = 2287.0408163265306
Mean of bedrooms = 476.3469387755102
Median of rooms = 1967.0
Median of bedrooms = 398.5
Mode of rooms = 1340
Mode of bedrooms = 318.0
```

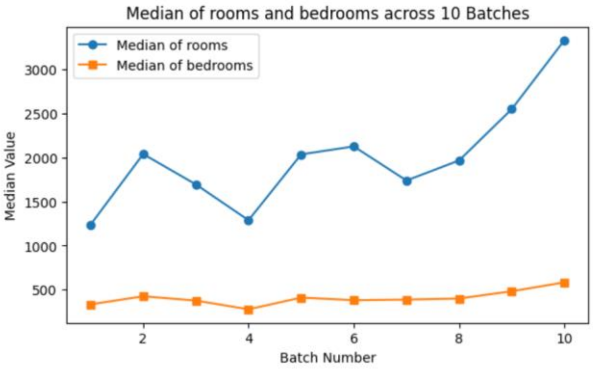


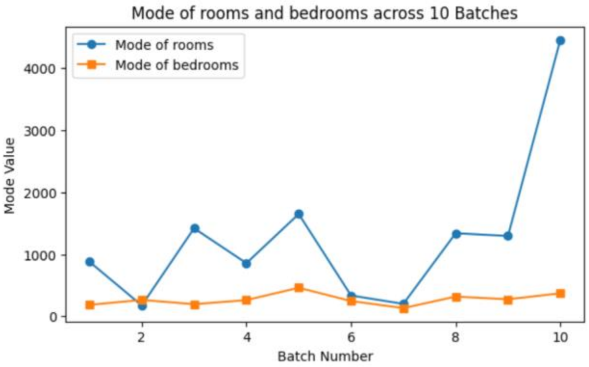
```
Batch 9
Mean of rooms = 3513.673469387755
Mean of bedrooms = 686.9183673469388
Median of rooms = 2550.0
Median of bedrooms = 481.5
Mode of rooms = 1295
Mode of bedrooms = 274.0
```



```
Batch 10
Mean of rooms = 4037.079207920792
Mean of bedrooms = 684.6336633663366
Median of rooms = 3333.0
Median of bedrooms = 582.0
Mode of rooms = 4458
Mode of bedrooms = 371.0
```



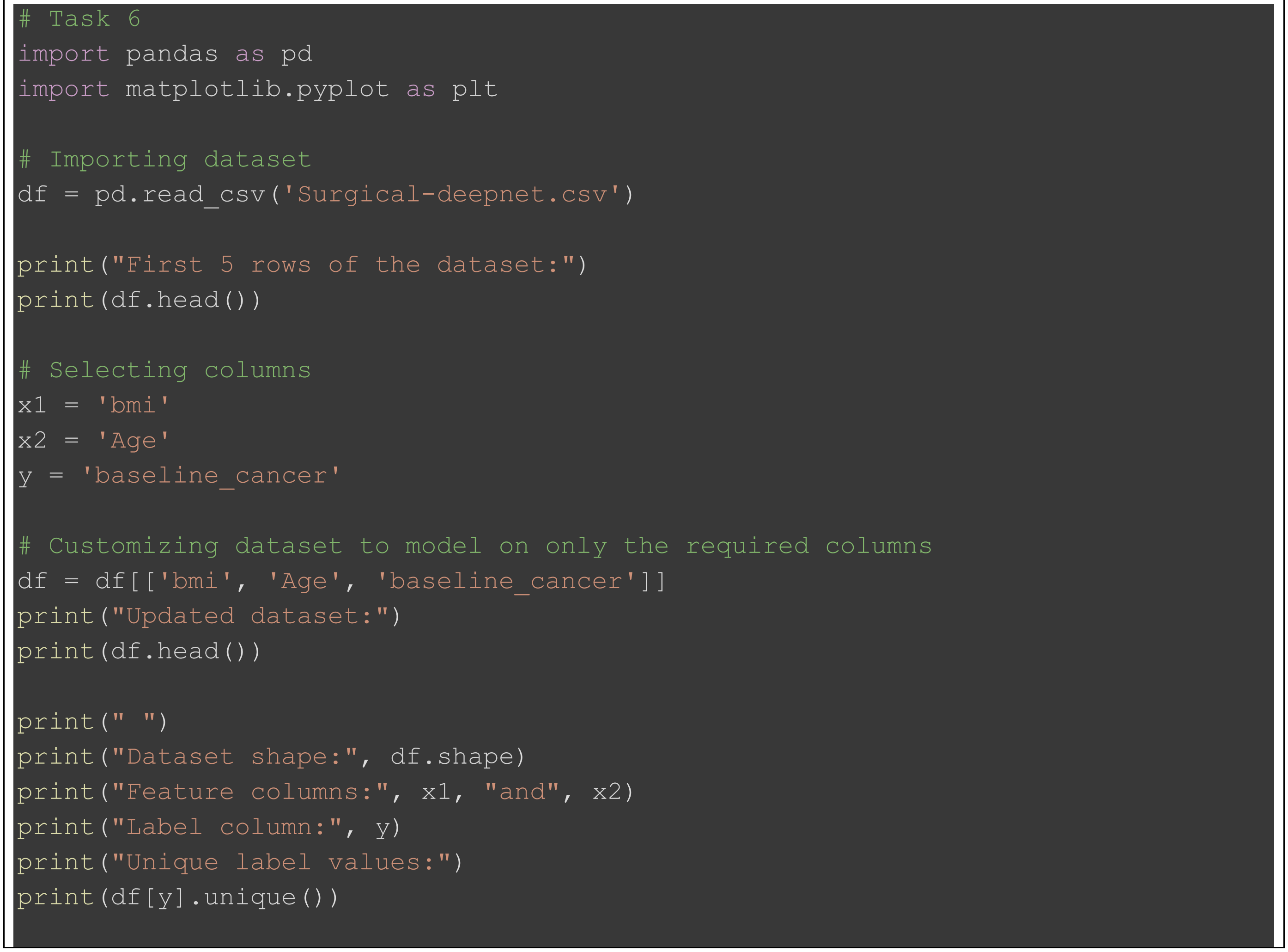




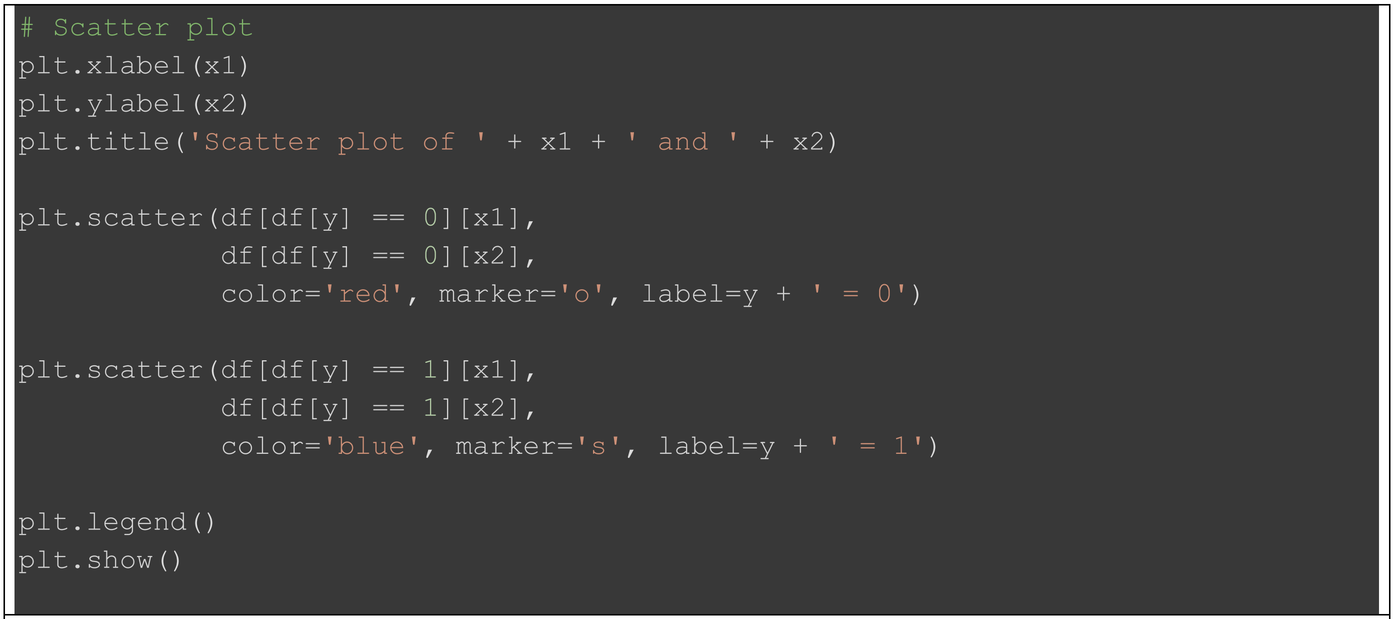
Lab Task 6 – Your Own Dataset

Download your own CSV dataset from the internet (e.g. Kaggle). Your dataset must have at least 500 rows and at least 2 feature columns. Your dataset must also have a labels column with classification data (0/1). Make a scatter plot between the feature axes and show the labels with different markers. Provide all of the codes and screenshots of the plots. You will also need to submit the downloaded dataset with your report. Note that no two submitted datasets must be exactly the same.

Code

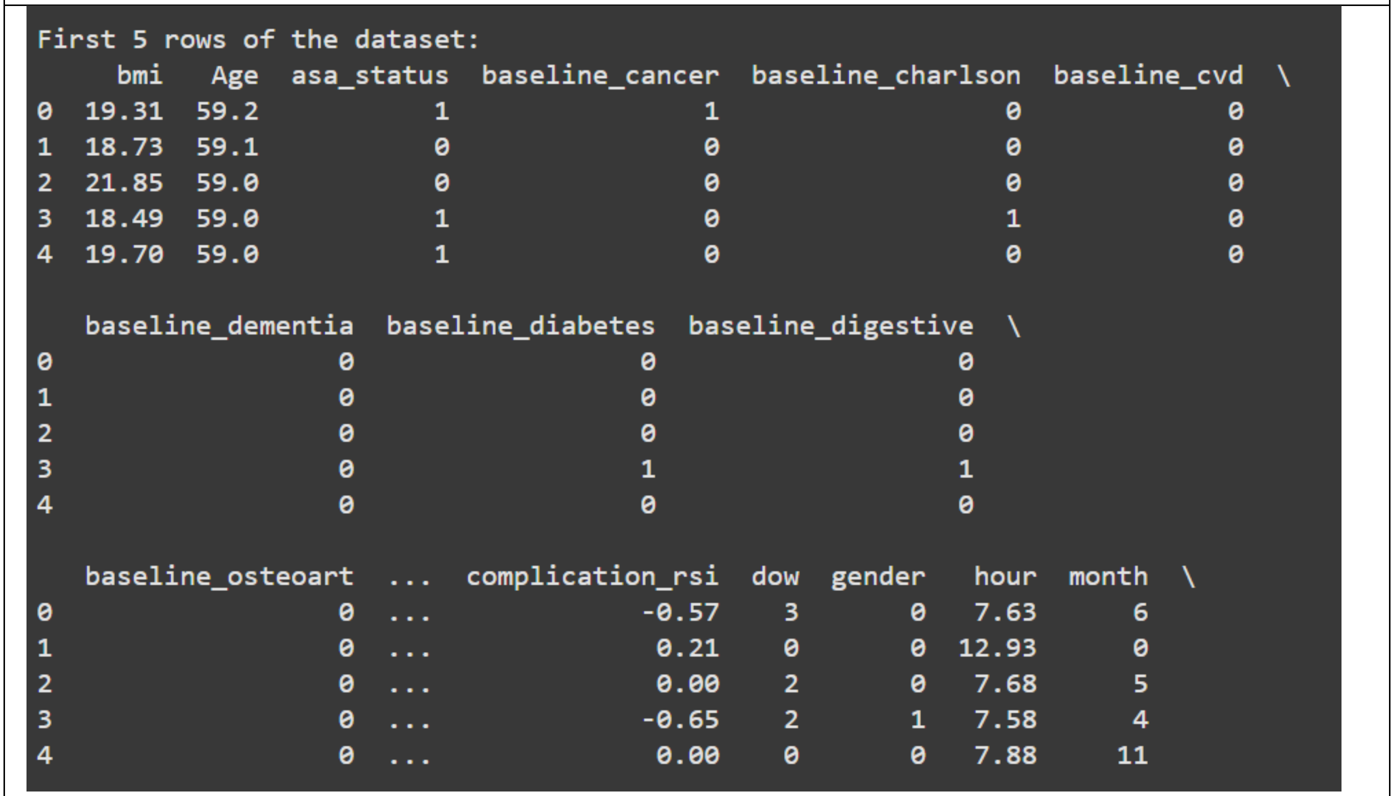


```
# Task 6
import pandas as pd
import matplotlib.pyplot as plt

# Importing dataset
df = pd.read_csv('Surgical-deepnet.csv')

print("First 5 rows of the dataset:")
print(df.head())

# Selecting columns
x1 = 'bmi'
x2 = 'Age'
y = 'baseline_cancer'

# Customizing dataset to model on only the required columns
df = df[['bmi', 'Age', 'baseline_cancer']]
print("Updated dataset:")
print(df.head())

print(" ")
print("Dataset shape:", df.shape)
print("Feature columns:", x1, "and", x2)
print("Label column:", y)
print("Unique label values:")
print(df[y].unique())
```

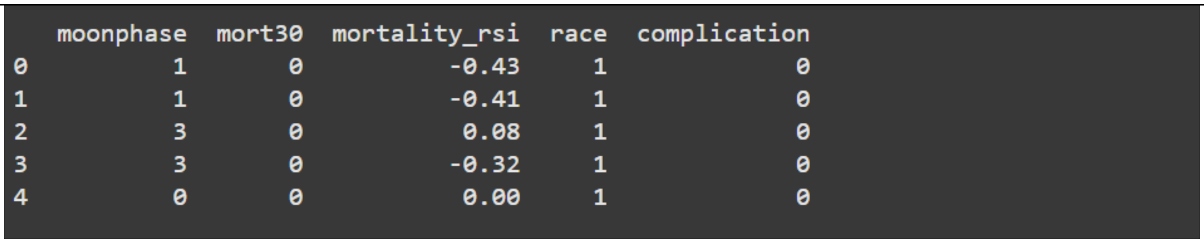


```
# Scatter plot
plt.xlabel(x1)
plt.ylabel(x2)
plt.title('Scatter plot of ' + x1 + ' and ' + x2)

plt.scatter(df[df[y] == 0][x1],
            df[df[y] == 0][x2],
            color='red', marker='o', label=y + ' = 0')

plt.scatter(df[df[y] == 1][x1],
            df[df[y] == 1][x2],
            color='blue', marker='s', label=y + ' = 1')

plt.legend()
plt.show()
```

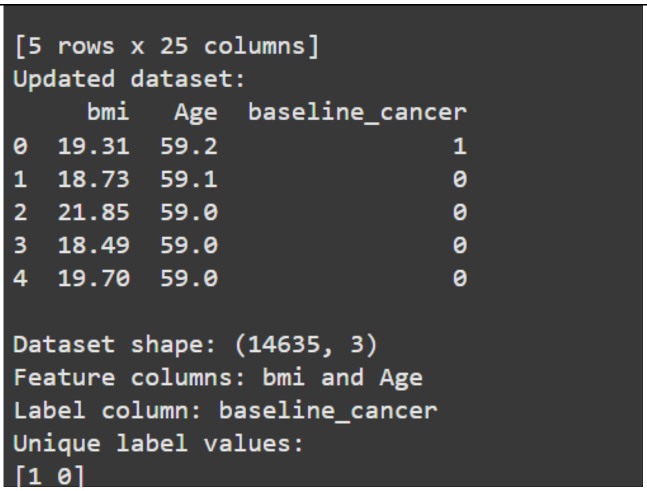
Output Console



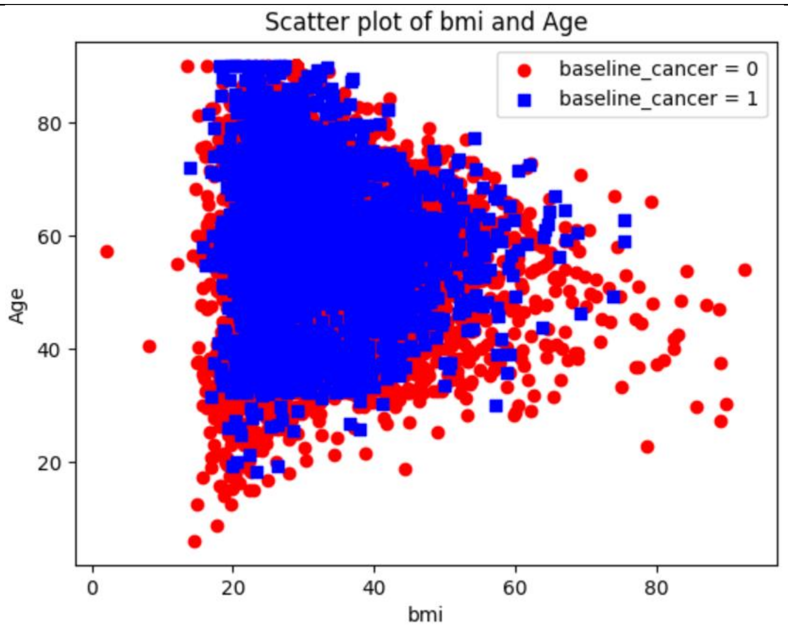
```
First 5 rows of the dataset:
     bmi   Age  asa_status  baseline_cancer  baseline_charlson  baseline_cvd  \
0  19.31  59.2           1                1                  0             0
1  18.73  59.1           0                0                  0             0
2  21.85  59.0           0                0                  0             0
3  18.49  59.0           1                0                  1             0
4  19.70  59.0           1                0                  0             0

   baseline_dementia  baseline_diabetes  baseline_digestive  \
0                  0                  0                   0
1                  0                  0                   0
2                  0                  0                   0
3                  0                  1                   1
4                  0                  0                   0

   baseline_osteoart  ...  complication_rsi  dow  gender   hour  month  \
0                  0  ...             -0.57    3       0   7.63      6
1                  0  ...              0.21    0       0  12.93      0
2                  0  ...              0.00    2       0   7.68      5
3                  0  ...             -0.65    2       1   7.58      4
4                  0  ...              0.00    0       0   7.88     11
```



```
   moonphase  mort30  mortality_rsi  race  complication
0          1       0          -0.43     1             0
1          1       0          -0.41     1             0
2          3       0           0.08     1             0
3          3       0          -0.32     1             0
4          0       0           0.00     1             0
```



```
[5 rows x 25 columns]
Updated dataset:
     bmi   Age  baseline_cancer
0  19.31  59.2                1
1  18.73  59.1                0
2  21.85  59.0                0
3  18.49  59.0                0
4  19.70  59.0                0

Dataset shape: (14635, 3)
Feature columns: bmi and Age
Label column: baseline_cancer
Unique label values:
[1 0]
```


Scatter plot of bmi and Age
● baseline_cancer = 0
■ baseline_cancer = 1
Age
bmi