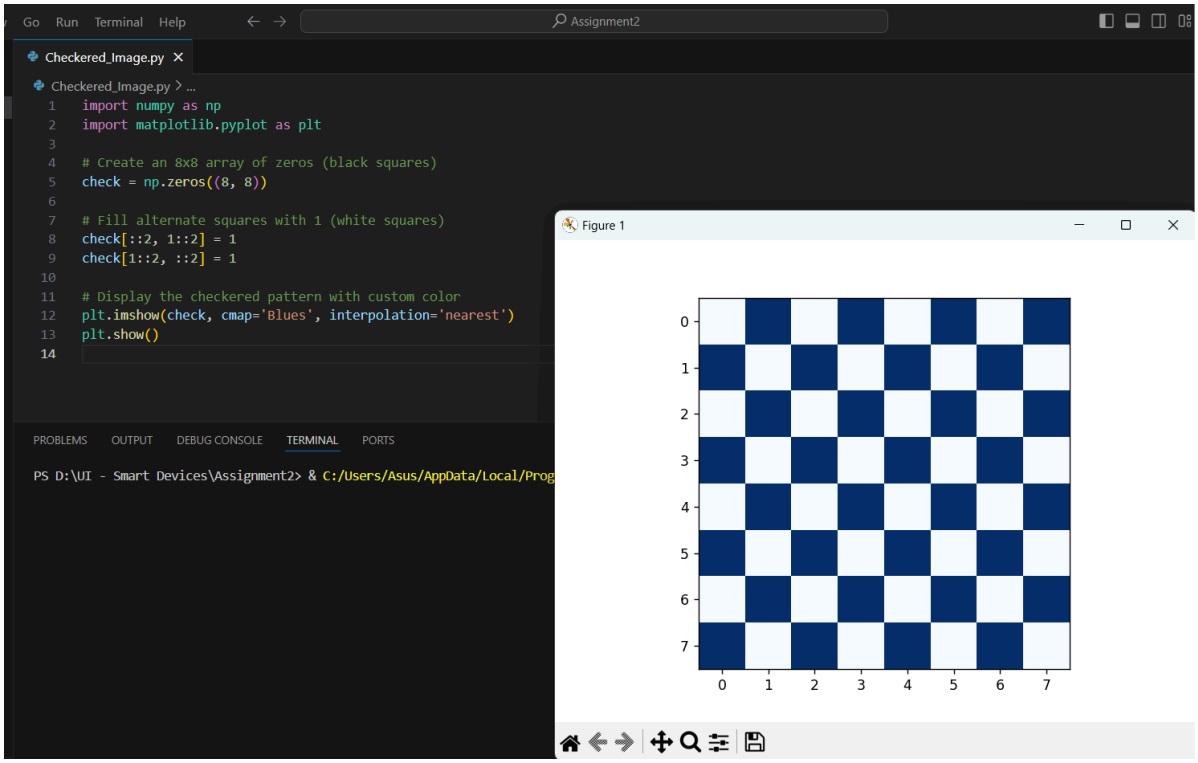
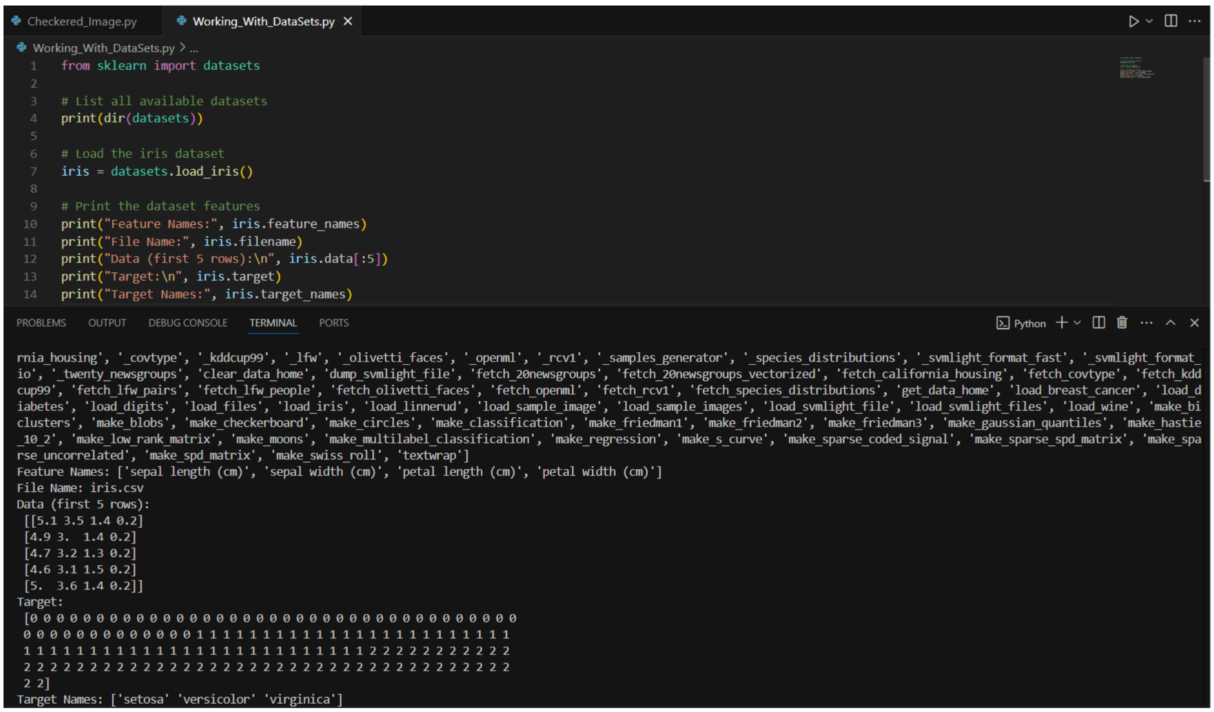
Assignment 2
24F --UI - Smart Devices (SEC. 401)
Md Abdul Kader #301358013

Activity 1: Checkered image. Print the image in the color of your choice.



SkLearn Activities:

- Display all the datasets.
- Pick up any dataset and look at the following commands to print:
    - Feature names
    - File name
    - Frame
    - Target
    - Target names



Checkered_Image.py | Working_With_DataSets.py

Working_With_DataSets.py > ...

```python
from sklearn import datasets

# List all available datasets
print(dir(datasets))

# Load the iris dataset
iris = datasets.load_iris()

# Print the dataset features
print("Feature Names:", iris.feature_names)
print("File Name:", iris.filename)
print("Data (first 5 rows):\n", iris.data[:5])
print("Target:\n", iris.target)
print("Target Names:", iris.target_names)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS | Python

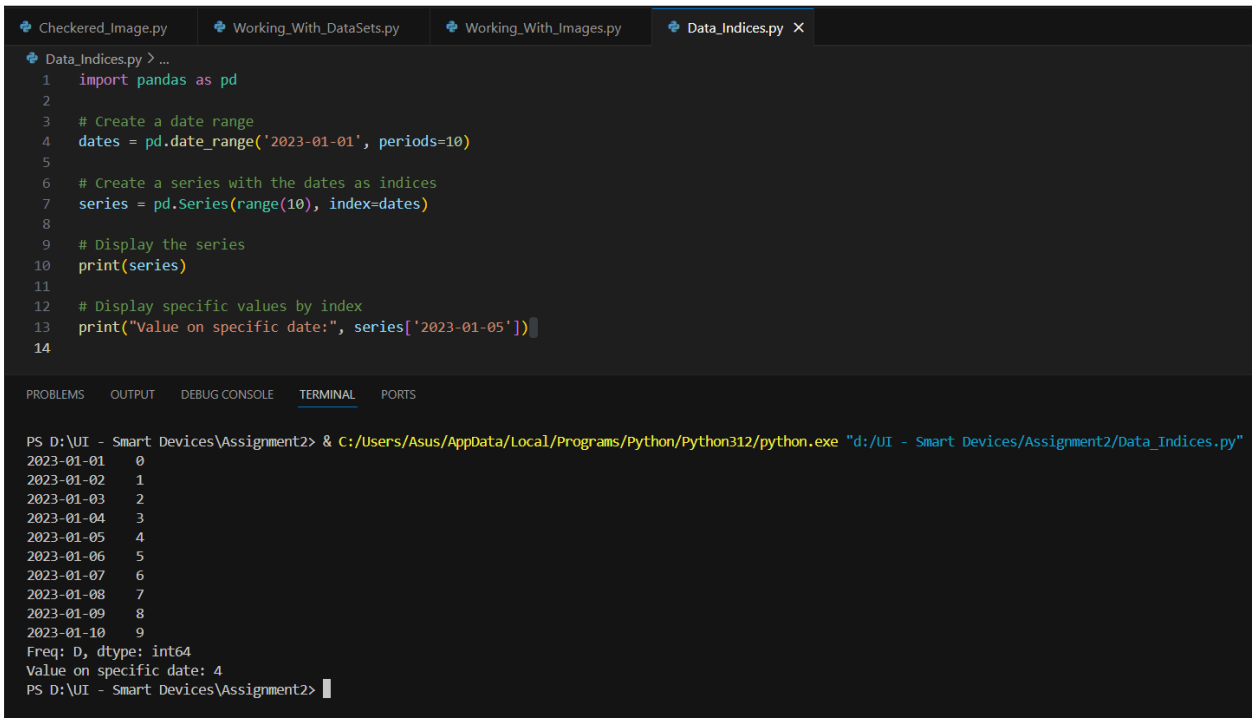
```
rnia_housing', '_covtype', '_kddcup99', '_lfw', '_olivetti_faces', '_openml', '_rcv1', '_samples_generator', '_species_distributions', '_svmlight_format_fast', '_svmlight_format_
io', '_twenty_newsgroups', 'clear_data_home', 'dump_svmlight_file', 'fetch_20newsgroups', 'fetch_20newsgroups_vectorized', 'fetch_california_housing', 'fetch_covtype', 'fetch_kdd
cup99', 'fetch_lfw_pairs', 'fetch_lfw_people', 'fetch_olivetti_faces', 'fetch_openml', 'fetch_rcv1', 'fetch_species_distributions', 'get_data_home', 'load_breast_cancer', 'load_d
iabetes', 'load_digits', 'load_files', 'load_iris', 'load_linnerud', 'load_sample_image', 'load_sample_images', 'load_svmlight_file', 'load_svmlight_files', 'load_wine', 'make_bi
clusters', 'make_blobs', 'make_checkerboard', 'make_circles', 'make_classification', 'make_friedman1', 'make_friedman2', 'make_friedman3', 'make_gaussian_quantiles', 'make_hastie
_10_2', 'make_low_rank_matrix', 'make_moons', 'make_multilabel_classification', 'make_regression', 'make_s_curve', 'make_sparse_coded_signal', 'make_sparse_spd_matrix', 'make_spa
rse_uncorrelated', 'make_spd_matrix', 'make_swiss_roll', 'textwrap']
Feature Names: ['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
File Name: iris.csv
Data (first 5 rows):
 [[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]]
Target:
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2]
Target Names: ['setosa' 'versicolor' 'virginica']
```

Scikit-image Activities:

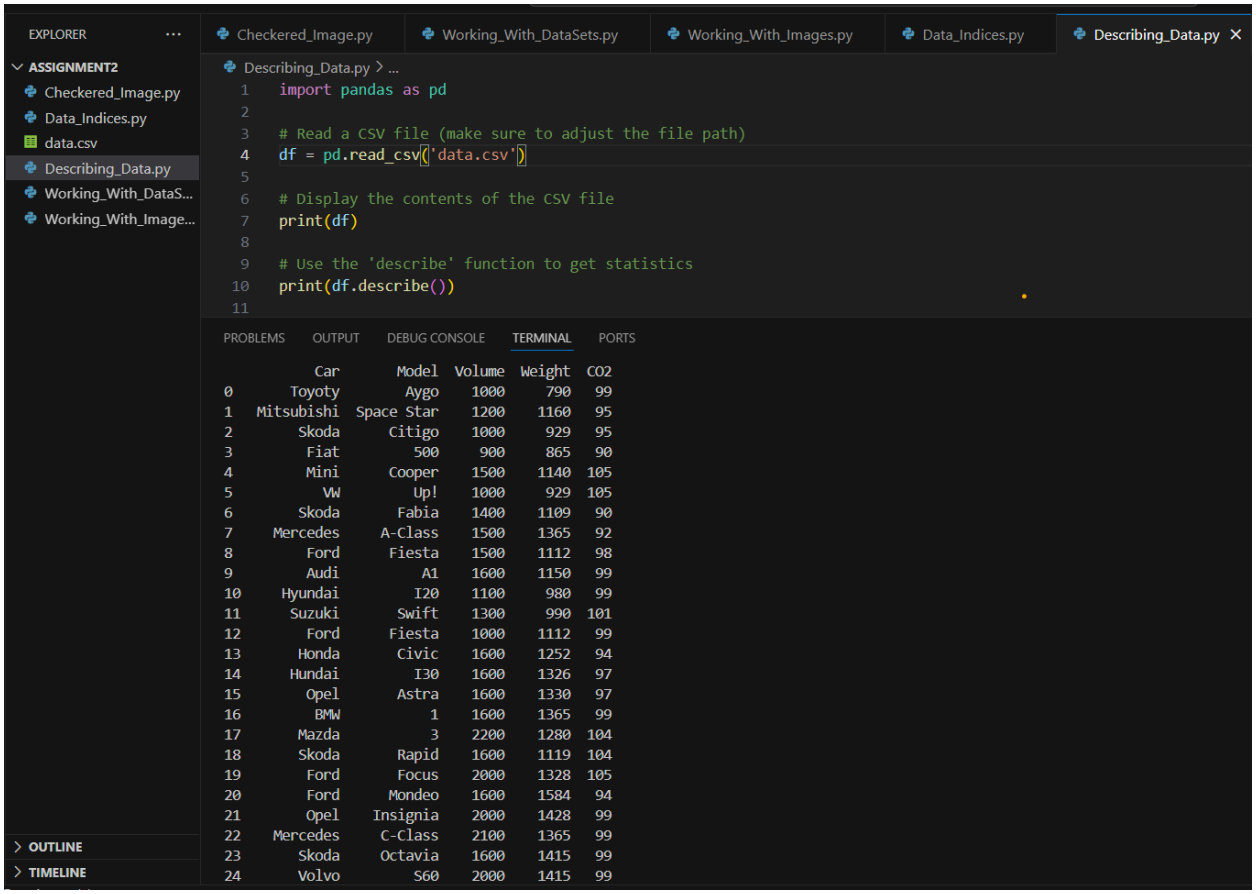
- Display all the images that come with the Scikit-image library.
- Pick up any image and look at the following commands to print:
- data
- datatype
- shape
- plot the image.
- Use the filter to plot images with gaussian noise and sigma as 5.

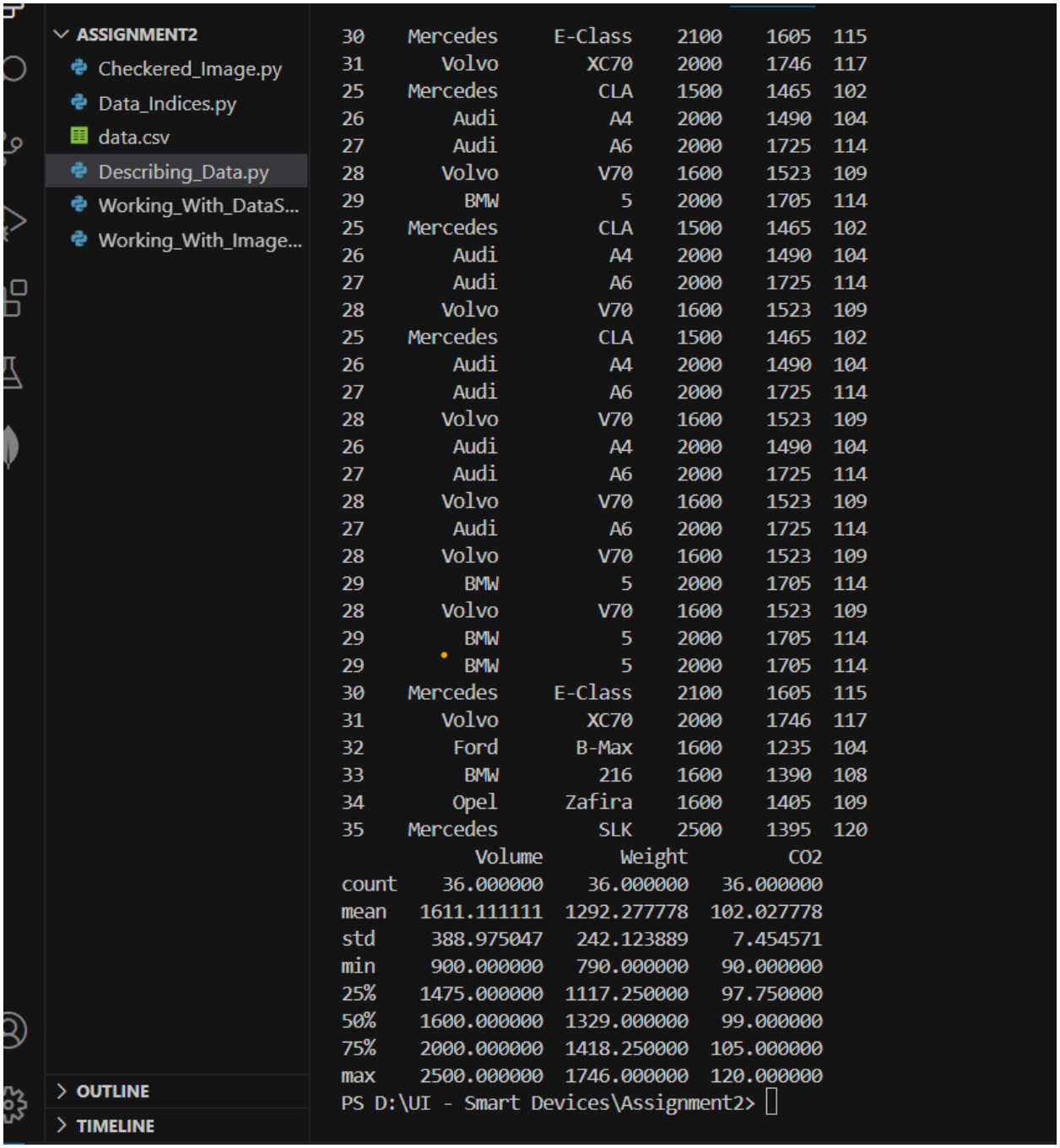
Pandas Activities:

• Create a series of 10 numbers with dates as indices. Display some specific numbers with values or indices.



• Create a .csv file, read it using pandas and display the content. Use the command describe to display the statistics of the file.


EXPLORER
ASSIGNMENT2
Checkered_Image.py
Data_Indices.py
data.csv
Describing_Data.py
Working_With_DataS...
Working_With_Image...
OUTLINE
TIMELINE
Checkered_Image.py
Working_With_DataSets.py
Working_With_Images.py
Data_Indices.py
Describing_Data.py
Describing_Data.py > ...
1 import pandas as pd
2
3 # Read a CSV file (make sure to adjust the file path)
4 df = pd.read_csv('data.csv')
5
6 # Display the contents of the CSV file
7 print(df)
8
9 # Use the 'describe' function to get statistics
10 print(df.describe())
11
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Car Model Volume Weight CO2
0 Toyoty Aygo 1000 790 99
1 Mitsubishi Space Star 1200 1160 95
2 Skoda Citigo 1000 929 95
3 Fiat 500 900 865 90
4 Mini Cooper 1500 1140 105
5 VW Up! 1000 929 105
6 Skoda Fabia 1400 1109 90
7 Mercedes A-Class 1500 1365 92
8 Ford Fiesta 1500 1112 98
9 Audi A1 1600 1150 99
10 Hyundai I20 1100 980 99
11 Suzuki Swift 1300 990 101
12 Ford Fiesta 1000 1112 99
13 Honda Civic 1600 1252 94
14 Hundai I30 1600 1326 97
15 Opel Astra 1600 1330 97
16 BMW 1 1600 1365 99
17 Mazda 3 2200 1280 104
18 Skoda Rapid 1600 1119 104
19 Ford Focus 2000 1328 105
20 Ford Mondeo 1600 1584 94
21 Opel Insignia 2000 1428 99
22 Mercedes C-Class 2100 1365 99
23 Skoda Octavia 1600 1415 99
24 Volvo S60 2000 1415 99


ASSIGNMENT2
Checkered_Image.py
Data_Indices.py
data.csv
Describing_Data.py
Working_With_DataS...
Working_With_Image...
OUTLINE
TIMELINE
30 Mercedes E-Class 2100 1605 115
31 Volvo XC70 2000 1746 117
25 Mercedes CLA 1500 1465 102
26 Audi A4 2000 1490 104
27 Audi A6 2000 1725 114
28 Volvo V70 1600 1523 109
29 BMW 5 2000 1705 114
25 Mercedes CLA 1500 1465 102
26 Audi A4 2000 1490 104
27 Audi A6 2000 1725 114
28 Volvo V70 1600 1523 109
25 Mercedes CLA 1500 1465 102
26 Audi A4 2000 1490 104
27 Audi A6 2000 1725 114
28 Volvo V70 1600 1523 109
26 Audi A4 2000 1490 104
27 Audi A6 2000 1725 114
28 Volvo V70 1600 1523 109
27 Audi A6 2000 1725 114
28 Volvo V70 1600 1523 109
29 BMW 5 2000 1705 114
28 Volvo V70 1600 1523 109
29 BMW 5 2000 1705 114
29 BMW 5 2000 1705 114
30 Mercedes E-Class 2100 1605 115
31 Volvo XC70 2000 1746 117
32 Ford B-Max 1600 1235 104
33 BMW 216 1600 1390 108
34 Opel Zafira 1600 1405 109
35 Mercedes SLK 2500 1395 120
Volume Weight CO2
count 36.000000 36.000000 36.000000
mean 1611.111111 1292.277778 102.027778
std 388.975047 242.123889 7.454571
min 900.000000 790.000000 90.000000
25% 1475.000000 1117.250000 97.750000
50% 1600.000000 1329.000000 99.000000
75% 2000.000000 1418.250000 105.000000
max 2500.000000 1746.000000 120.000000
PS D:\UI - Smart Devices\Assignment2>