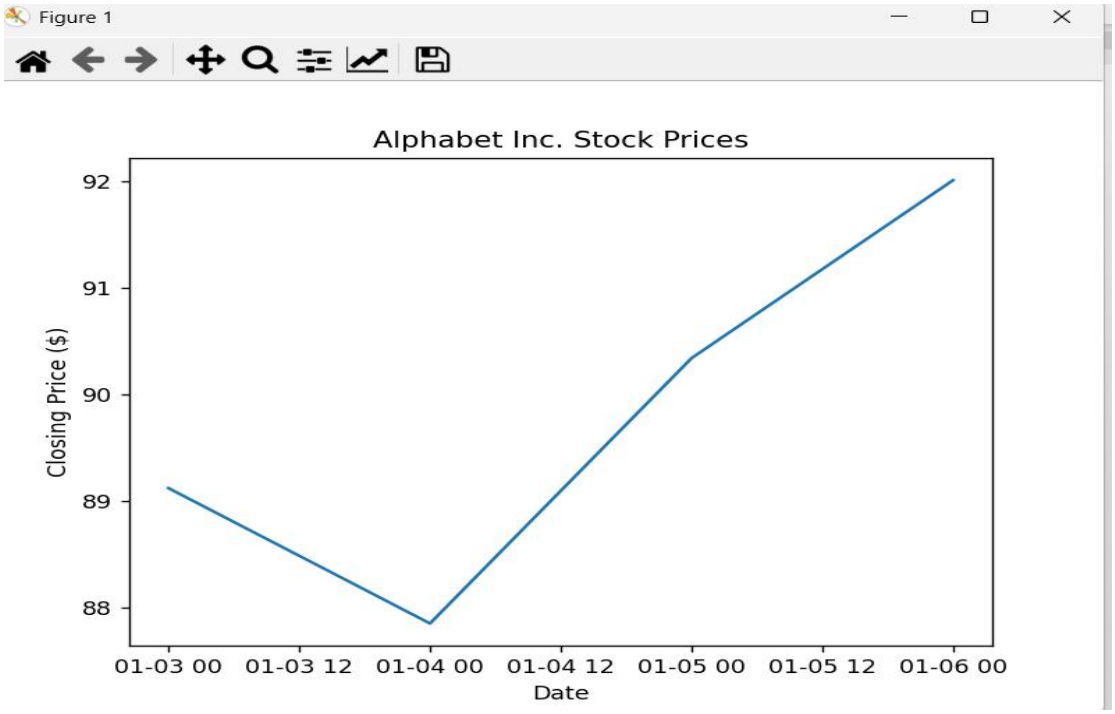
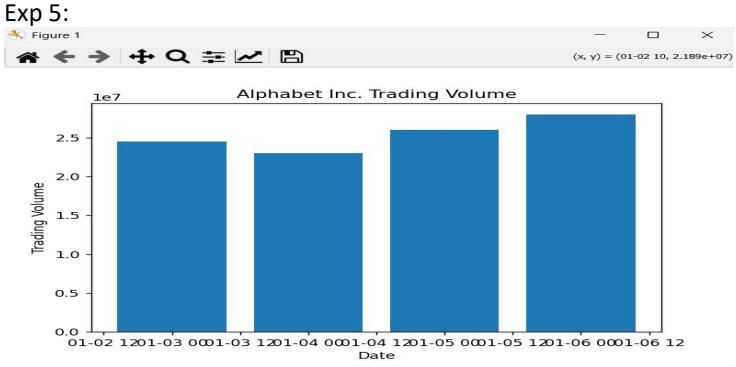
OUPUTS

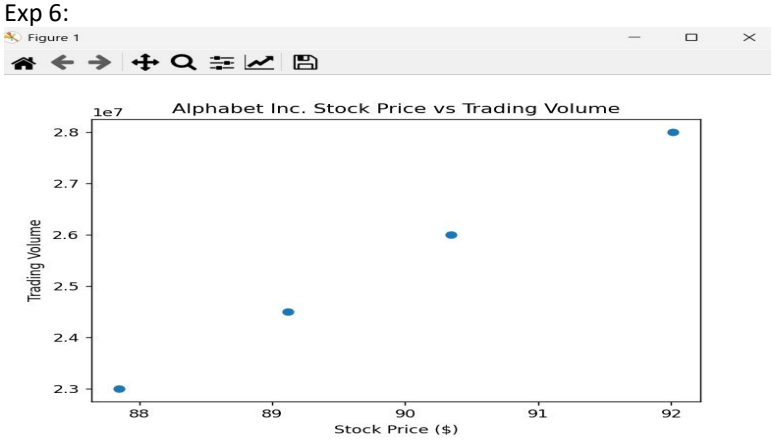
```
Python 3.13.9 (tags/v3.13.9:8183fa5, Oct 14 2025, 14:09:13) [MSC v.1944 64 bit (AMD64)] on win32
Enter "help" below or click "Help" above for more information.

========= RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 1.py ========
Distinct Department IDs:
[ 10  20  30  40  50  60  70  80  90 100 110 120 130 140 150 160 170 180
 190 200 210 220 230 240 250 260 270]

========= RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 2.py ========
Employee IDs who have done two or more jobs:
[101, 176, 200]

========= RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 3.py ========
        JOB_ID                      JOB_TITLE  MIN_SALARY  MAX_SALARY
11      ST_MAN                  Stock Manager        5500        8500
12    ST_CLERK                    Stock Clerk        2008        5000
13    SH_CLERK                 Shipping Clerk        2500        5500
8       SA_REP           Sales Representative        6000       12008
7       SA_MAN                  Sales Manager       10000       20080
9       PU_MAN              Purchasing Manager        8000       15000
10    PU_CLERK                Purchasing Clerk        2500        5500
18      PR_REP  Public Relations Representative        4500       10500
6   AC_ACCOUNT               Public Accountant        4200        9000
14     IT_PROG                      Programmer        4000       10000
0      AD_PRES                       President       20080       40000
16      MK_REP       Marketing Representative        4000        9000
15      MK_MAN              Marketing Manager        9000       15000
17      HR_REP  Human Resources Representative        4000        9000
3       FI_MGR                 Finance Manager        8200       16000
1        AD_VP  Administration Vice President       15000       30000
2      AD_ASST       Administration Assistant        3000        6000
5       AC_MGR              Accounting Manager        8200       16000
4   FI_ACCOUNT                      Accountant        4200        9000

========= RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 4.py ========
```



Exp 5:



Exp 6:



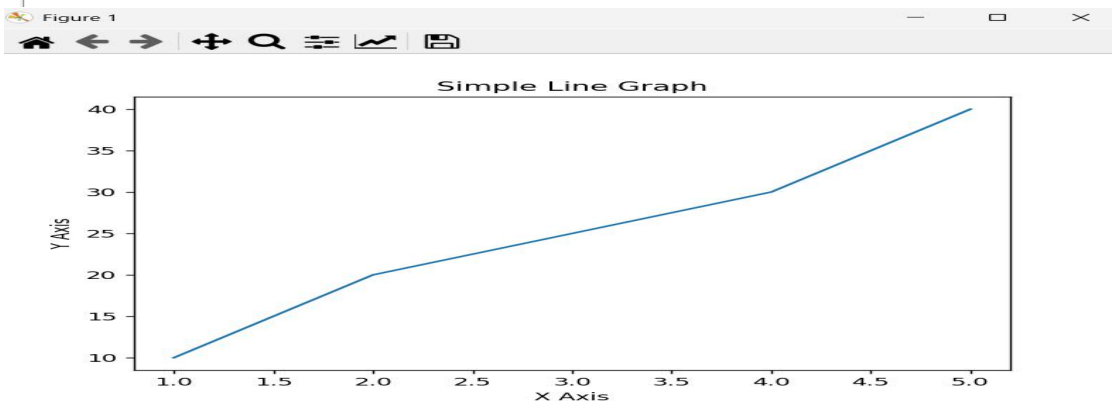

```
========== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 7.py ==========
                max        min
         Sale_Value Sale_Value
Item
Notebook       2000       1800
Pen            1500       1200
Pencil          950        800

========== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 8.py ==========
          Units_Sold
Item
Notebook         200
Pen              270
Pencil           380

========== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 9.py ==========
                          Sale_amt
Region  Manager SalesMan
Central Douglas John           250
        Hermann Luis         150948
                Shelli        25000
                Sigal        121820
        Martha  Steven        89850
        Timothy David          6075
East    Douglas Karen         40500
        Martha  Alexander    231076
                Diana         14500
West    Douglas Michael       38336
        Timothy Stephen       67088
```

```
======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 10.py ========
Original DataFrame:

      A     B   C     D
0  81.0  33.0  41  17.0
1  43.0  75.0  36  23.0
2  71.0   NaN  80  37.0
3  82.0   3.0  10  45.0
4  28.0   3.0  35  59.0
5  53.0  85.0  96   NaN
6  98.0  93.0  88  57.0
7   NaN  51.0  56  31.0
8   6.0  31.0  31  46.0
9  87.0   7.0   1  72.0

NaN Positions (True indicates NaN):

       A      B      C      D
0  False  False  False  False
1  False  False  False  False
2  False   True  False  False
3  False  False  False  False
4  False  False  False  False
5  False  False  False   True
6  False  False  False  False
7   True  False  False  False
8  False  False  False  False
9  False  False  False  False

DataFrame with NaN values marked:

      A     B   C     D
0  81.0  33.0  41  17.0
1  43.0  75.0  36  23.0
2  71.0   NaN  80  37.0
3  82.0   3.0  10  45.0
4  28.0   3.0  35  59.0
5  53.0  85.0  96   NaN
6  98.0  93.0  88  57.0
7   NaN  51.0  56  31.0
8   6.0  31.0  31  46.0
9  87.0   7.0   1  72.0

================================================================ RESTART: C:\Users\M
 AYESHA\OneDrive\Desktop\Practice\exp 11.py ======================================
==========================
Original DataFrame:

   Col1  Col2  Col3  Col4
0  17.0    35  84.0  32.0
1  26.0    97   NaN  53.0
2  52.0    67  94.0  50.0
3  59.0    40  25.0  79.0
4   NaN    86  16.0  34.0
5  84.0    19   5.0  78.0
6  87.0    10  33.0  96.0
7  50.0     6   9.0   NaN
8   5.0    75  82.0  37.0
9  95.0    21  59.0  46.0

NaN Positions (True = NaN):

    Col1   Col2   Col3   Col4
0  False  False  False  False
1  False  False   True  False
2  False  False  False  False
3  False  False  False  False
4   True  False  False  False
5  False  False  False  False
6  False  False  False  False
7  False  False  False   True
8  False  False  False  False
9  False  False  False  False
Python 3.13.9 (tags/v3.13.9:8183fa5, Oct 14 2025, 14:09:13) [MSC v.1944 64 bit (
AMD64)] on win32
Enter "help" below or click "Help" above for more information.

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 12.py ========
DataFrame values:

          A         B         C         D
0  1.881615  0.308648 -2.794185 -1.652725
1 -1.327564  0.635468 -0.091019  0.765807
2 -1.099691  0.999901  0.570137  0.936598
3  0.690882  1.174607 -0.483486 -0.587894
4  0.087374  1.296866 -0.448272  0.237075
5 -1.750025  1.119646  0.032516 -1.991299
6 -0.645123  2.648821  1.010596  1.625086
7 -0.795778  1.179783  0.501092  0.703723
8  0.905231  1.449572 -1.128669 -0.394412
9 -0.151958  0.903341 -0.438605  1.017578

Rounded DataFrame:

      A     B     C     D
0  1.88  0.31 -2.79 -1.65
1 -1.33  0.64 -0.09  0.77
2 -1.10  1.00  0.57  0.94
3  0.69  1.17 -0.48 -0.59
4  0.09  1.30 -0.45  0.24
5 -1.75  1.12  0.03 -1.99
6 -0.65  2.65  1.01  1.63
7 -0.80  1.18  0.50  0.70
8  0.91  1.45 -1.13 -0.39
9 -0.15  0.90 -0.44  1.02

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 13.py ========
       A      B      C
0  False  False   True
1  False   True  False
2   True  False  False
3  False  False  False
```

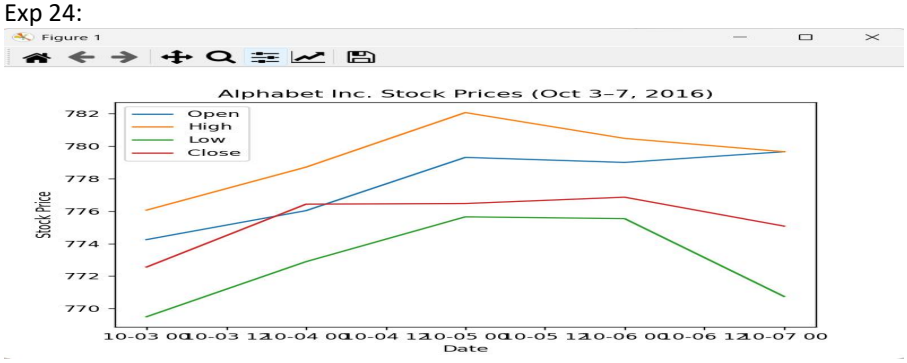
```
======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 14.py ========
Original DataFrame:
      A     B     C
0  10.0   5.0   NaN
1  20.0   NaN   8.0
2   NaN  15.0  12.0
3  40.0  20.0  16.0

DataFrame after replacing missing values:
      A     B     C
0  10.0   5.0   0.0
1  20.0   0.0   8.0
2   0.0  15.0  12.0
3  40.0  20.0  16.0

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 15.py ========
Original DataFrame:
      A     B     C     D
0  10.0   NaN   NaN   5.0
1   NaN   NaN   8.0   NaN
2   NaN  15.0   NaN  12.0
3  40.0  20.0  16.0   NaN
4   NaN   NaN   NaN  25.0

Rows with at least 2 NaN values:
      A     B    C     D
0  10.0   NaN  NaN   5.0
1   NaN   NaN  8.0   NaN
2   NaN  15.0  NaN  12.0
4   NaN   NaN  NaN  25.0

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 16.py ========
Original DataFrame:
  school_code class student_name
0        S001     V         Alex
1        S002    VI         John
2        S001     V       Martha
3        S003   VII        Steve
4        S002    VI         Lucy
5        S001   VII        David

Data grouped by School Code:

School Code: S001
  school_code class student_name
0        S001     V         Alex
```

```
Type of GroupBy object:
<class 'pandas.core.groupby.generic.DataFrameGroupBy'>

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 17.py ========
Original DataFrame:
  school_code class    name  age
0        S001     V    Alex   11
1        S002    VI    John   12
2        S001     V  Martha   10
3        S003   VII   Steve   13
4        S002    VI    Lucy   12
5        S001   VII   David   14

Mean, Min and Max age for each school:
                mean  min  max
school_code
S001       11.666667   10   14
S002       12.000000   12   12
S003       13.000000   13   13

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 18.py ========
Original DataFrame:
  school_code class    name  age
0        S001     V    Alex   11
1        S002    VI    John   12
2        S001     V  Martha   10
3        S003   VII   Steve   13
4        S002    VI    Lucy   12
5        S001   VII   David   14

Data grouped by School Code and Class:

Group: ('S001', 'V')
  school_code class    name  age
0        S001     V    Alex   11
2        S001     V  Martha   10

Group: ('S001', 'VII')
  school_code class   name  age
5        S001   VII  David   14

Group: ('S002', 'VI')
  school_code class  name  age
```

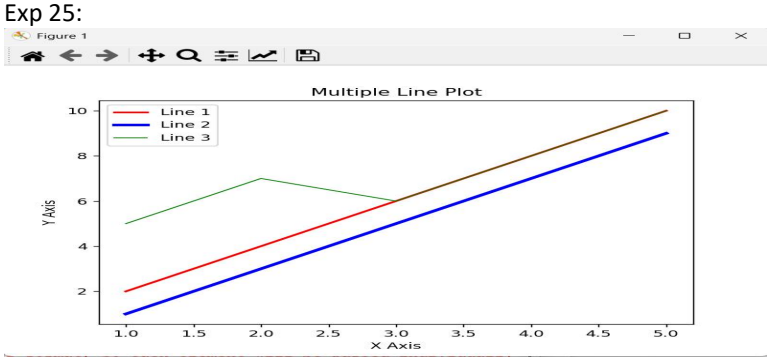
```
Group: ('S002', 'VI')
  school_code class  name  age
1        S002    VI  John   12
4        S002    VI  Lucy   12

Group: ('S003', 'VII')
  school_code class   name  age
3        S003   VII  Steve   13
>

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 19.py ========
Dataset Shape (Rows, Columns):
(5, 5)

Column Names:
['Year', 'WHO region', 'Country', 'Beverage Types', 'Display Value']
>

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 20.py ========
Index of rows containing substring 'Beer':
[3, 4]
>

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\EXP 21.py ========
Original DataFrame:
      Name       City
0     Alex   New York
1   mArThA     LoNDoN
2     JOHN      PaRiS
3     luCy     BeRlIn

DataFrame after swapping case of Name column:
      Name       City
0     aLEX   New York
1   MaRtHa     LoNDoN
2     john      PaRiS
3     LUcY     BeRlIn
>

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 22.py ========
```



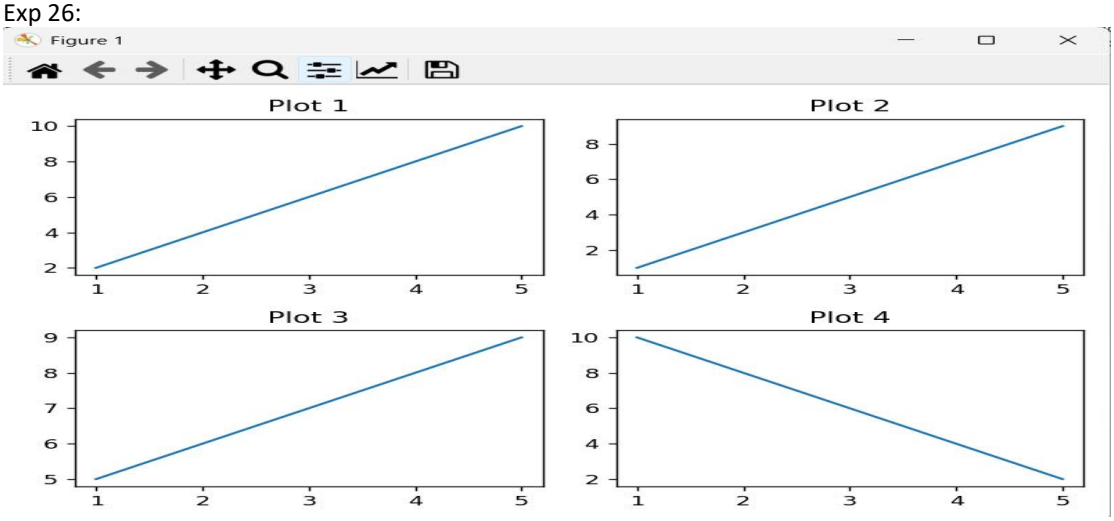
Exp 24:



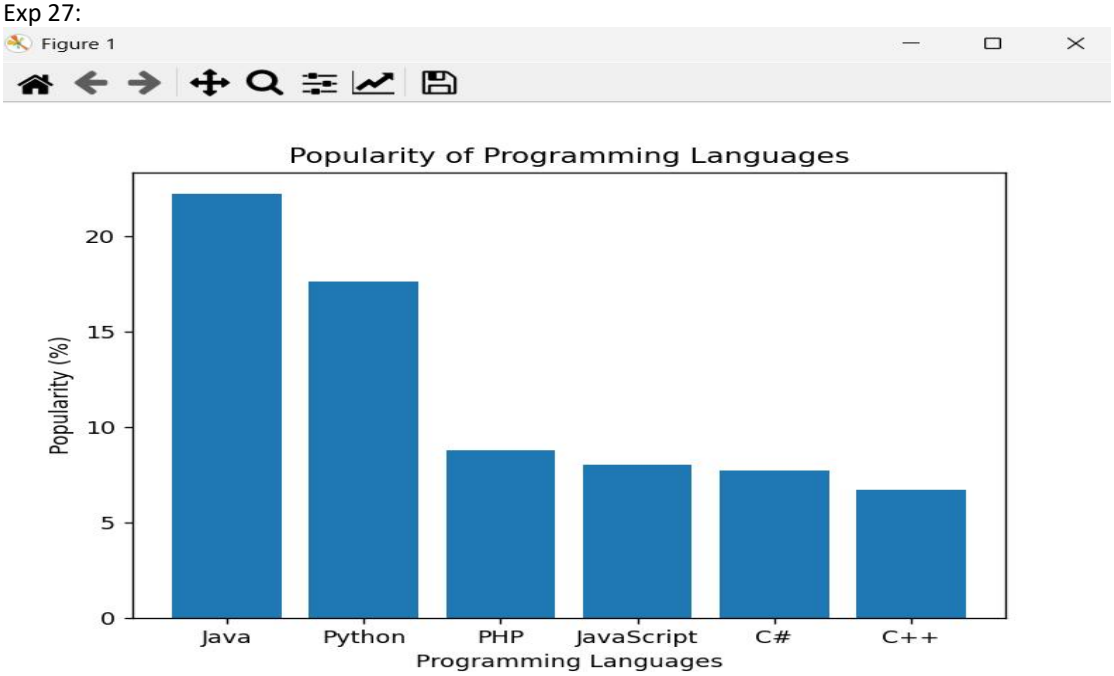
Exp 25:



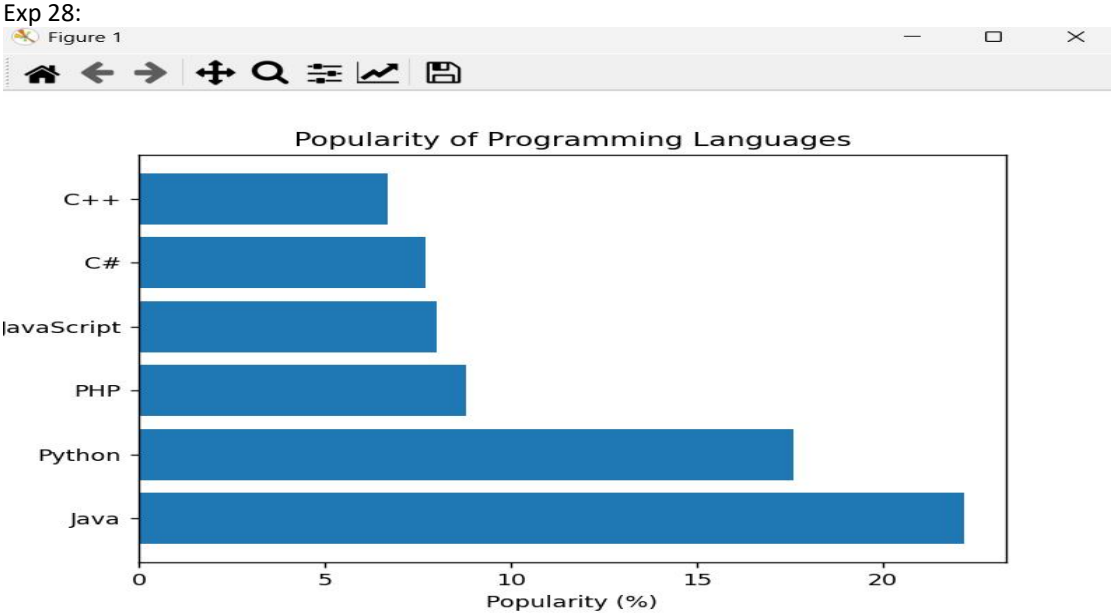
Exp 26:



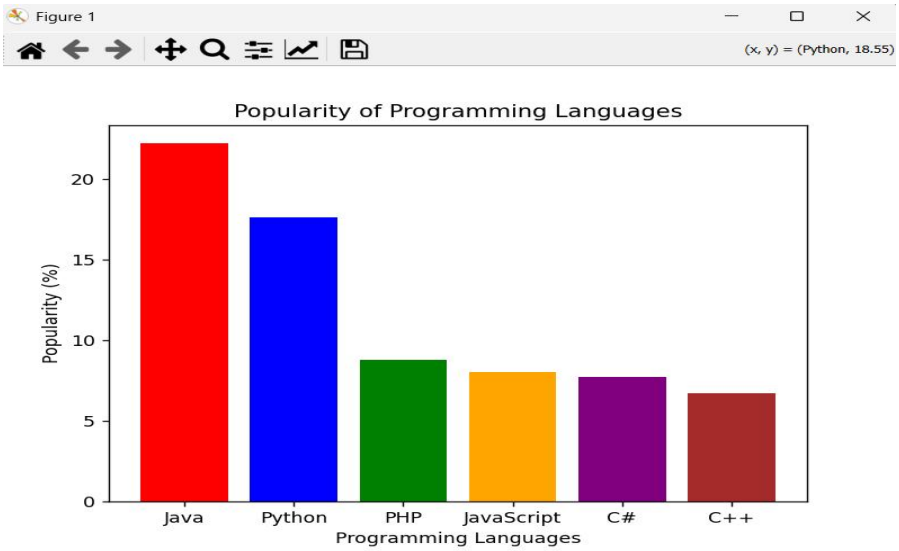
Exp 27:



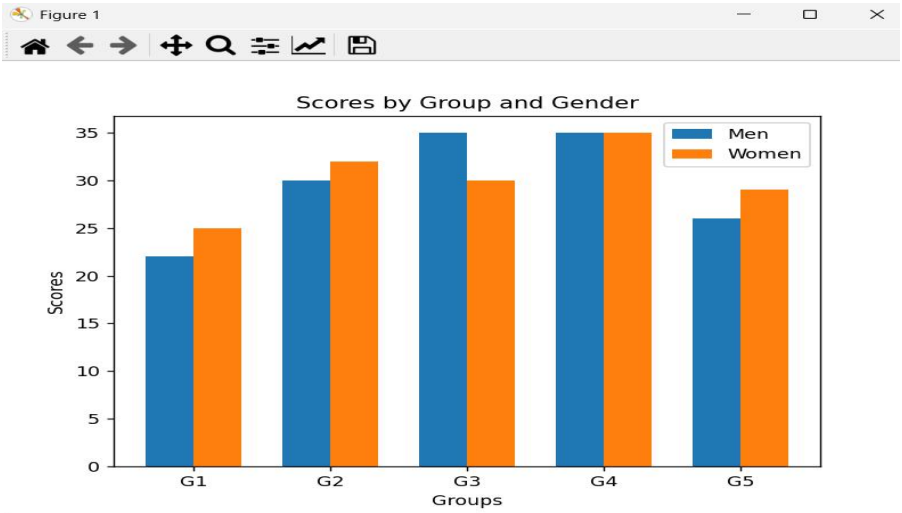
Exp 28:



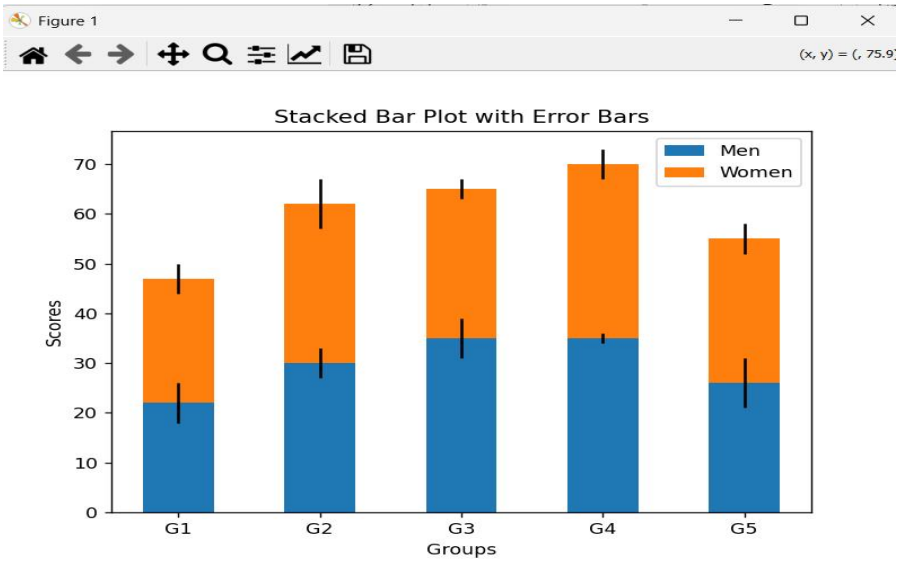
Exp 29:



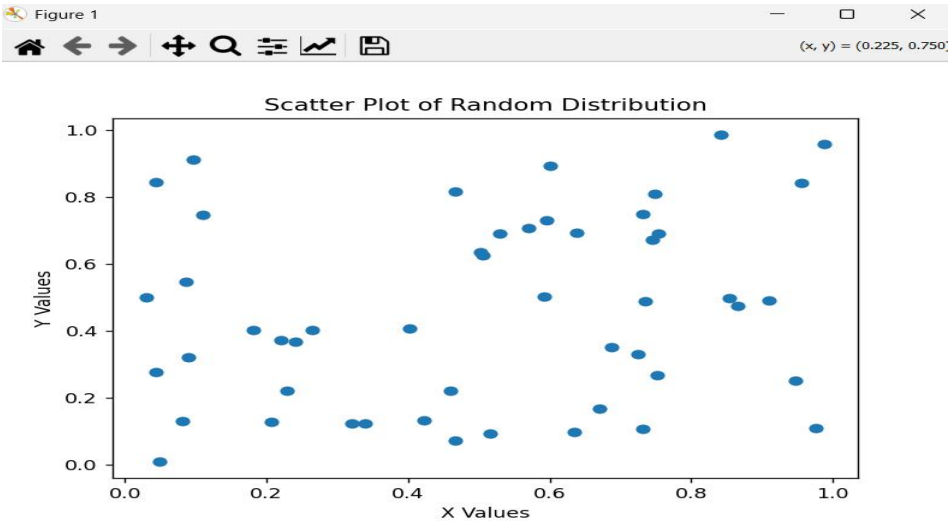
Exp 30:



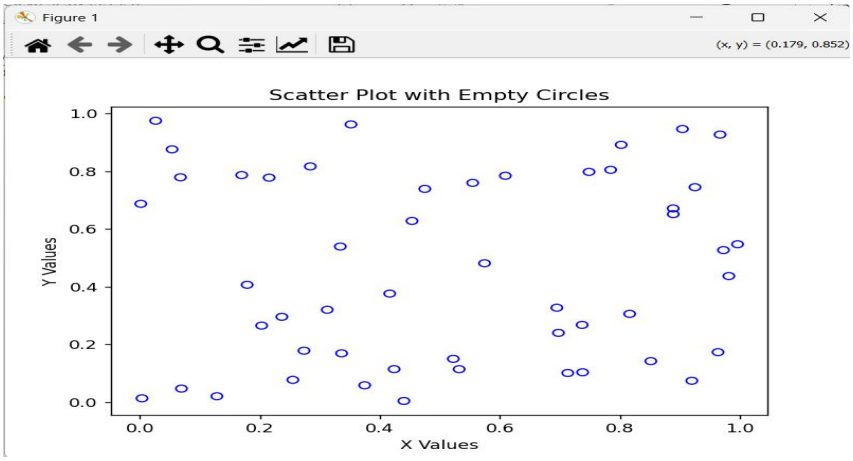
Exp 31:



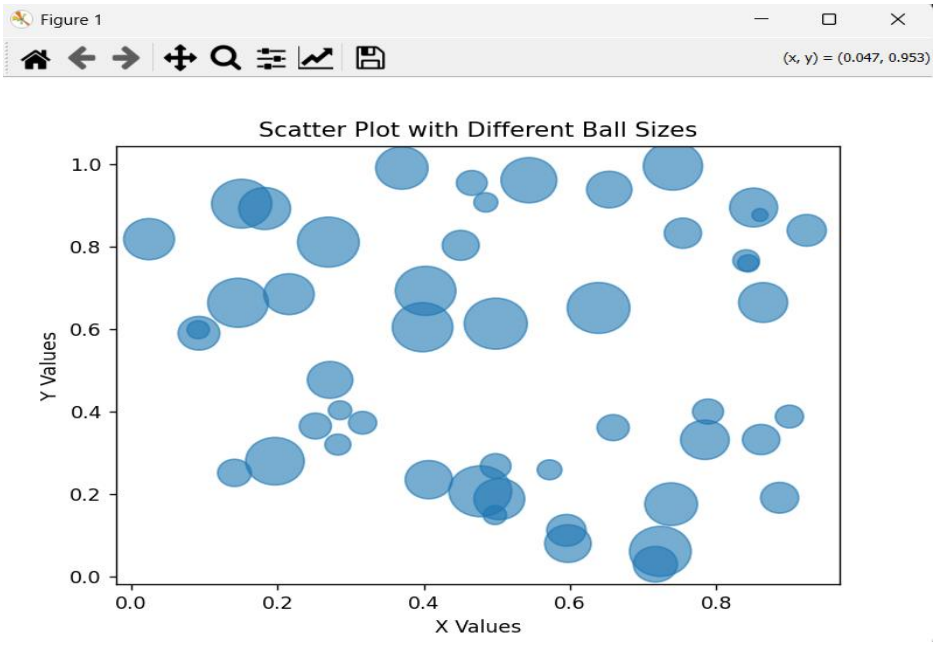
Exp 32:



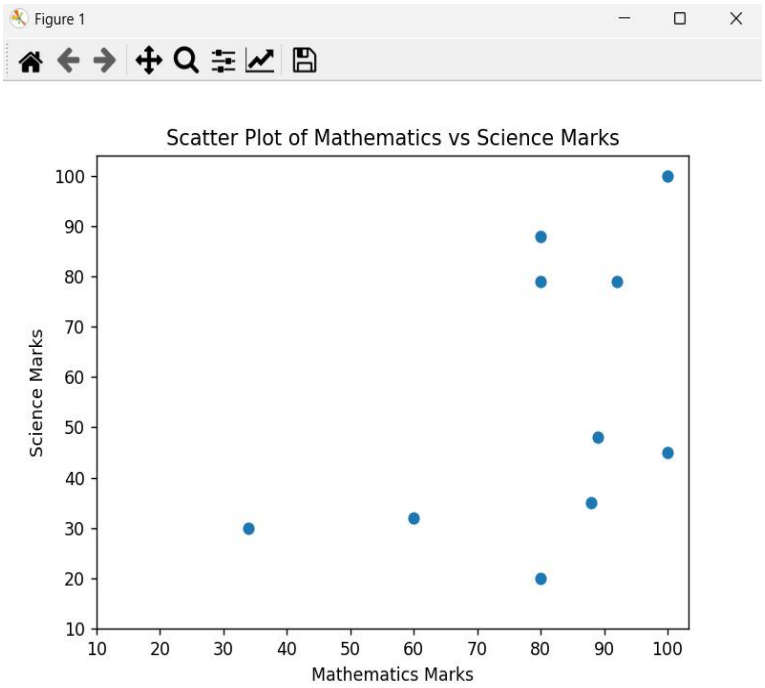
Exp 33:



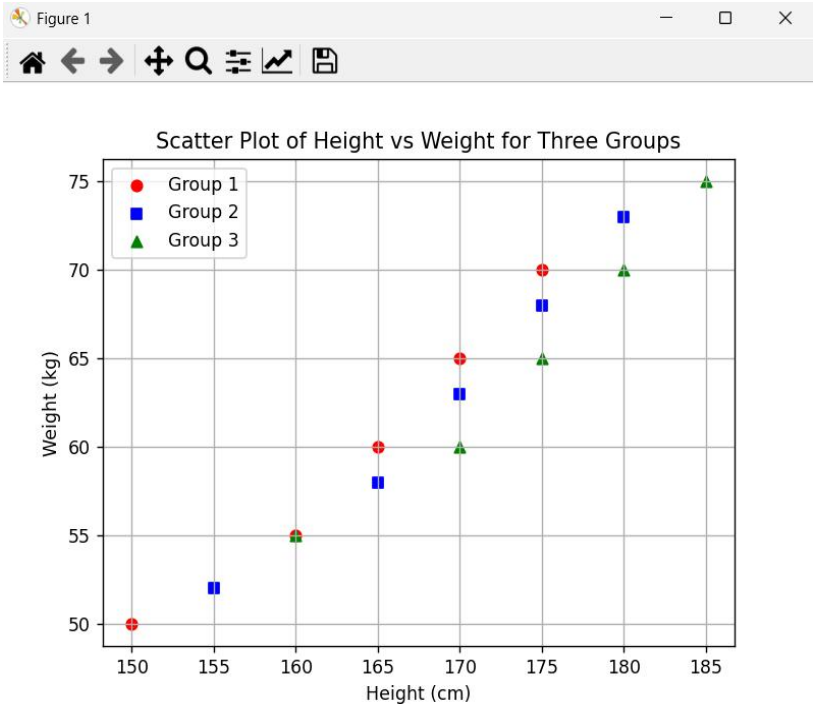
Exp 34:



Exp 35:



Exp 36:



Exp 37:

```
======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 37.py ========
DataFrame:
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
```

```
======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 38.py ========
DataFrame with specified index labels:
       name  score  attempts qualify
a  Anastasia   12.5         1     yes
b      Dima    9.0         3      no
c  Katherine   16.5         2     yes
d     James    NaN         3      no
e     Emily    9.0         2      no
f   Michael   20.0         3     yes
g   Matthew   14.5         1     yes
h     Laura    NaN         1      no
i     Kevin    8.0         2      no
j     Jonas   19.0         1     yes

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 39.py ========
First 3 rows of the DataFrame:
       name  score  attempts qualify
a  Anastasia   12.5         1     yes
b      Dima    9.0         3      no
c  Katherine   16.5         2     yes

======== RESTART: C:\Users\M AYESHA\OneDrive\Desktop\Practice\exp 40.py ========
Selected columns 'name' and 'score':
       name  score
a  Anastasia   12.5
b      Dima    9.0
c  Katherine   16.5
d     James    NaN
e     Emily    9.0
f   Michael   20.0
g   Matthew   14.5
h     Laura    NaN
i     Kevin    8.0
j     Jonas   19.0
```