

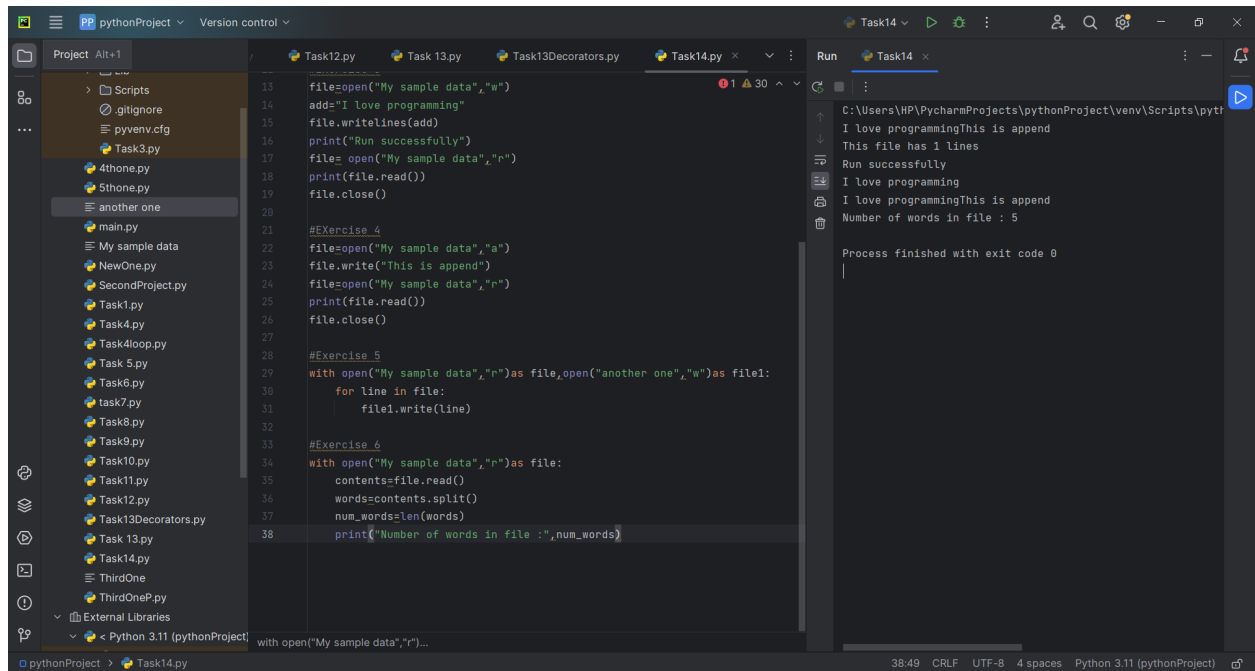
The screenshot displays a Python IDE interface with three main panels:

- Project Explorer (Left):** Shows a project structure with a folder named `Scripts` containing several files: `gitignore`, `pyvenv.cfg`, `Task3.py`, `4thone.py`, `5thone.py`, `another one`, `main.py`, `My sample data`, `NewOne.py`, `SecondProject.py`, `Task1.py`, `Task4.py`, `Task4loop.py`, `Task 5.py`, `Task6.py`, `task7.py`, `Task8.py`, `Task9.py`, `Task10.py`, `Task11.py`, `Task12.py`, `Task13Decorators.py`, `Task 13.py`, `Task14.py`, `ThirdOne`, and `ThirdOneP.py`. Below the project files, there are sections for `External Libraries` and `Python 3.11 (pythonProject)`.
- Editor (Middle):** Displays the content of `Task14.py`. The code includes comments for exercises and uses `file.open()` to read and write data. It also includes a `with open()` block for file handling.
- Run Console (Right):** Shows the output of the script execution. The output indicates that the file was successfully opened, read, and written, and that the process finished with exit code 0.

```
1 #File_handling
2 #Exercise 1
3 file=open("My sample data", "r")
4 print(file.read())
5 file.close()
6
7
8 #exercise 2
9 with open("My sample data","r") as file:
10     lines=len(file.readlines())
11     print("This file has",lines,"lines")
12
13 #Exercise 3
14 file=open("My sample data","w")
15 add="I love programming"
16 file.writelines(add)
17 print("Run successfully")
18 file=open("My sample data","r")
19 print(file.read())
20 file.close()
21
22 #Exercise 4
23 file=open("My sample data","a")
24 file.write("This is append")
25 file=open("My sample data","r")
26 print(file.read())
27 file.close()
28
29 #Exercise 5
30 with open("My sample data","r") as file,open("another one","w") as file1:
31     for line in file:
32         file1.write(line)
```

Run Console Output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe
I love programmingThis is append
This file has 1 lines
Run successfully
I love programming
I love programmingThis is append
Number of words in file : 5
Process finished with exit code 0
```



Task15

Task for class on 27-10-2023

Exception Handling

Exercise 1

Write a Python program that prompts the user to input a list of integers and raises an exception if any of the integers in the list are negative.

Exercise 2

Write a Python program that prompts the user to input a list of integers and computes the average of those integers. Use try-except blocks to handle any exceptions that might occur. Use the finally clause to print a message indicating that the program has finished running.

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'Lib', 'Scripts', 'gitignore', 'pyvenv.cfg', 'Task3.py', '4thone.py', '5thone.py', 'another one', 'main.py', 'My sample data', 'NewOne.py', 'SecondProject.py', 'Task1.py', 'Task4.py', 'Task4loop.py', and 'Task 5.py'. The main editor displays 'Task 15.py' with the following code:

```
1 try:
2     while True:
3         n=int(input("Enter an integer:"))
4         if n<0:
5             raise ValueError("Enter a positive integer")
6
7 except ValueError as e:
8     print(f"error :{e}")
```

The Run console at the bottom shows the execution of 'Task 15.py' with the following output:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\pythonProject\Task 15.py"
Enter an integer:12
Enter an integer:2
Enter an integer:55
Enter an integer:14
Enter an integer:-1
error :Enter a positive integer
Process finished with exit code 0
```

The screenshot shows the PyCharm IDE with a project named 'pythonProject'. The file explorer on the left shows a directory structure with files like 'pythonProject', 'venv', 'Lib', 'Scripts', 'gitignore', 'pyvenv.cfg', 'Task3.py', '4thone.py', '5thone.py', 'another one', 'main.py', 'My sample data', 'NewOne.py', 'SecondProject.py', 'Task1.py', and 'Task4.py'. The main editor displays 'Task 15 2.py' with the following code:

```
1 #Exercise 2
2 try:
3     sum=0
4     count=1
5     while True:
6         n=int(input("Enter an integer(press 0 to exit):"))
7         count+=1
8         if n==0:
9             break
10        sum+=n
11    avg=sum/count
12    print("Average of integers:",avg)
13 except:
14     print("error")
15 finally:
16     print("Program has finished running")
17
```

The Run console at the bottom shows the execution of 'Task 15 2.py' with the following output:

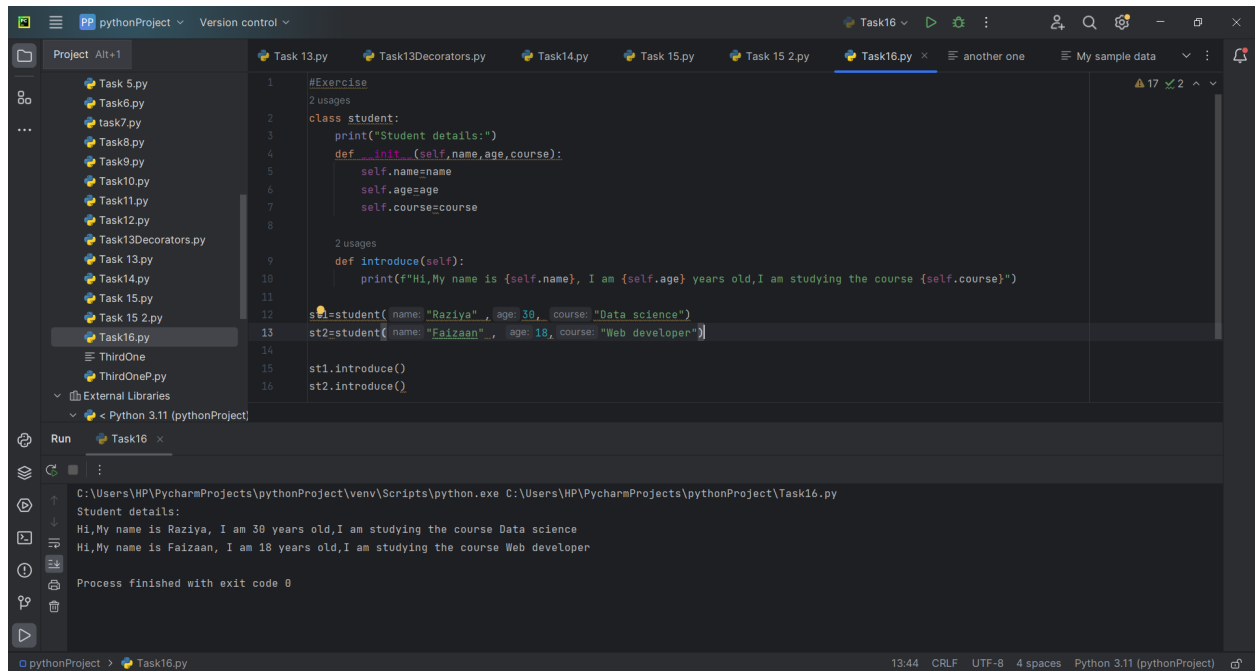
```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe "C:\Users\HP\PycharmProjects\pythonProject\Task 15 2.py"
Enter an integer(press 0 to exit):12
Enter an integer(press 0 to exit):45
Enter an integer(press 0 to exit):67
Enter an integer(press 0 to exit):78
Enter an integer(press 0 to exit):65
Enter an integer(press 0 to exit):0
Average of integers: 53.4
Program has finished running
Process finished with exit code 0
```

Task 16

Task for class on 30-10-2023

Classes and Objects:

Create a Python class called Student with a constructor that initializes attributes for a student's name, age, and course. Then, write a method that allows a student to introduce themselves. Finally, create two instances of the Student class and demonstrate how to use them.



Task17

Task for class conducted on 06-11-2023:

Regex:

- 1 Write a Python program to calculate the square root of a number using the math module.
- 2 Write a Python program to generate a random number between 1 and 10 using the random module.
- 3 Write a Python program to convert a string to uppercase using the string module.
- 4 Write a Python program to calculate the factorial of a number using the math module.
- 5 Write a regular expression to find all occurrences of the word "fox" in the given text
- 6 Write a regular expression to find all occurrences of the word "quick" in the given text
- 7 Write a regular expression to find all occurrences of words that start with the letter "c" in the given text.
- 8 Write a regular expression to find all occurrences of the word "lazy" followed by any one character in the given text.
- 9 Use match() to find if the text starts with "The"
- 10 Use search() to find the first occurrence of the word "fox"

The screenshot shows the PyCharm IDE with the file `Task17.py` open. The code contains six exercises related to regular expressions:

```
1 #Regular Expressions
2 #Exercise 1
3 import math
4 n=int(input("Enter a Number:"))
5 sqrt=math.sqrt(n)
6 print("Square root of number",n,"is",sqrt)
7
8 #Exercise 2
9 import random
10 rannum=random.randint(1,10)
11 print("The random number between 1 and 10 is:",rannum)
12
13 #Exercise 3
14 import string
15 nstring="Hi all, How are you"
16 trans_table=string.maketrans(string.ascii_lowercase,string.ascii_uppercase)
17 lstring=nstring.translate(trans_table)
18 print("the string",nstring,"in uppercase is:",lstring)
19
20 #Exercise 4
21 num=int(input("Enter a number"))
22 facnum=math.factorial(num)
23 print("Factorial of number",num,"is",facnum)
24
25 #Exercise 5
26 import re
27 txt="fox is an animal,fox lived in jungle,fox is brown in color"
28 x=re.findall(pattern="fox",txt)
29 print(x)
30
31 #Exercise 6
```

The Run console shows the output of the program:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\
Enter a Number:12
Square root of number 12 is 3.4641016151377544
The random number between 1 and 10 is: 5
the string Hi all, How are you in uppercase is: HI ALL, HOW ARE YOU
Enter a number10
Factorial of number 10 is 3628800
['fox', 'fox', 'fox']
['quick', 'quick', 'quick']
['can', 'carrying', 'cat']
['lazy ']
Text is starting with 'The'
The word 'fox' found

Process finished with exit code 0
```

The screenshot shows the PyCharm IDE with the file `Task17.py` open. The code contains five exercises related to regular expressions:

```
31 #Exercise 6
32 import re
33 txt="nick quick tick pick quick tick quick"
34 x=re.findall(pattern="quick",txt)
35 print(x)
36
37 #Exercise 7
38 import re
39 txt="a man can carrying a cat"
40 x=re.findall(pattern=r"\b(c|w)",txt)
41 print(x)
42
43 #Exercise 8
44 import re
45 txt="lazy become lazy"
46 x=re.findall(pattern=r"\blazy",txt)
47 print(x)
48
49 #Exercise 9
50 import re
51 txt="The laptop is heavy"
52 x=re.match(pattern="The",txt)
53 if x:
54     print("Text is starting with 'The'")
55 else:
56     print("Text is not starting with 'The'")
57
58 #Exercise 10
59 import re
60 txt="fox is a wild animal, it has a baby fox"
61 x=re.search(pattern=r"\bfox\b",txt)
```

The Run console shows the output of the program:

```
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\
Enter a Number:12
Square root of number 12 is 3.4641016151377544
The random number between 1 and 10 is: 5
the string Hi all, How are you in uppercase is: HI ALL, HOW ARE YOU
Enter a number10
Factorial of number 10 is 3628800
['fox', 'fox', 'fox']
['quick', 'quick', 'quick']
['can', 'carrying', 'cat']
['lazy ']
Text is starting with 'The'
The word 'fox' found

Process finished with exit code 0
```

```
pythonProject Version control
Project Alt+1
2.py
Task16.py
MyBankingProject.py
Task17.py
another one
My sample data
MyBankingProje
NewOne.py
SecondProject
Task1.py
Task4.py
Task4loop.py
Task 5.py
Task6.py
task7.py
Task8.py
Task9.py
Task10.py
Task11.py
Task12.py
Task13Decorat
Task14.py
Task 15 2.py
Task16.py
Task17.py
ThirdOne
ThirdOneP.py
External Libraries
Python 3.11
Binary Skele else

40 x=re.findall( pattern: r"\b(c|w)",txt)
41 print(x)
42
43 #Exercise 8
44 import re
45 txt="lazy become lazy"
46 x=re.findall( pattern: r"\blazy",txt)
47 print(x)
48
49 #Exercise 9
50 import re
51 txt="The laptop is heavy"
52 x=re.match( pattern: "The",txt)
53 if x:
54     print("Text is starting with 'The'")
55 else:
56     print("Text is not starting with 'The'")
57
58 #Exercise 10
59 import re
60 txt="fox is a wild animal, it has a baby fox"
61 x=re.search( pattern: r"\bfox\b",txt)
62 if x:
63     print("The word 'fox' found")
64 else:
65     print("not found")

Run Task17
C:\Users\HP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\
Enter a Number:12
Square root of number 12 is 3.4641016151377544
The random number between 1 and 10 is: 5
the string Hi all, How are you in uppercase is: HI ALL, HOW ARE YOU
Enter a number:10
Factorial of number 10 is 3628800
['fox', 'fox', 'fox']
['quick', 'quick', 'quick']
['can', 'carrying', 'cat']
['lazy ']
Text is starting with 'The'
The word 'fox' found

Process finished with exit code 0
```

Project

Task for class conducted on 03-11-2023:

OOPs:

We have covered core python, now its time to test your coding and logical skills, also to know how confident you are with the syntax, i'm giving you a small project:
Write a python program to replicate a Banking system. The following features are mandatory:

- 1.Account login
2. Amount Depositing
3. Amount Withdrawal

```
pythonProject Version control BankingProject Task15 2.py Task16.py Task17.py another on Run BankingProject
> Lib
> Scripts
> gitignore
> pyvenv.cfg
Task3.py
4thone.py
5thone.py
another one
BankingProject
main.py
My sample data
MyBankingProj
NewOne.py
SecondProject
Task1.py
Task4.py
Task4loop.py
Task 5.py
Task6.py
task7.py
Task8.py
Task9.py
Task10.py
Task11.py
Task12.py
Task13Decorati
TaskV12 nv

1 usage
class Bank:
    def __init__(self, accno, accname, username, password, balance=0):
        self.accNo=accno
        self.accholder=accname
        self.uname=username
        self.pwd=password
        self.balance=balance

    1 usage
    def Login(self,un,pw):
        if((self.uname==un)and(self.pwd==pw)):
            return True
        else:
            print("Login unsuccessful")

    1 usage
    def deposit(self,depAmount):
        if(depAmount<0):
            print("Invalid amount")
        else:
            self.balance=self.balance+depAmount
            print(depAmount,"successfully deposited")

    1 usage
    def withdraw(self,withamount):
        if(withamount>self.balance):
            print("Insufficient balance")
        else:

Problems 57 Project Errors 6 Server-Side Analysis Vulnerable Dependencies
> BankingProject.py C:\Users\HP\PycharmProjects\pythonProject\57 problems
pythonProject BankingProject.py 34:1 CRLF UTF-8 4 spaces Python 3.11 (pythonProject)
```

```
pythonProject Version control BankingProject Task15 2.py Task16.py Task17.py another on Run BankingProject
Project Alt+1 sk 15.py Task15 2.py Task16.py Task17.py another on Run BankingProject
> Lib
> Scripts
> gitignore
> pyvenv.cfg
Task3.py
4thone.py
5thone.py
another one
BankingProject
main.py
My sample data
MyBankingProj
NewOne.py
SecondProject
Task1.py
Task4.py
Task4loop.py
Task 5.py
Task6.py
Task7.py
Task8.py
Task9.py
Task10.py
Task11.py
Task12.py
Task13Decorati
TaskV12 nv

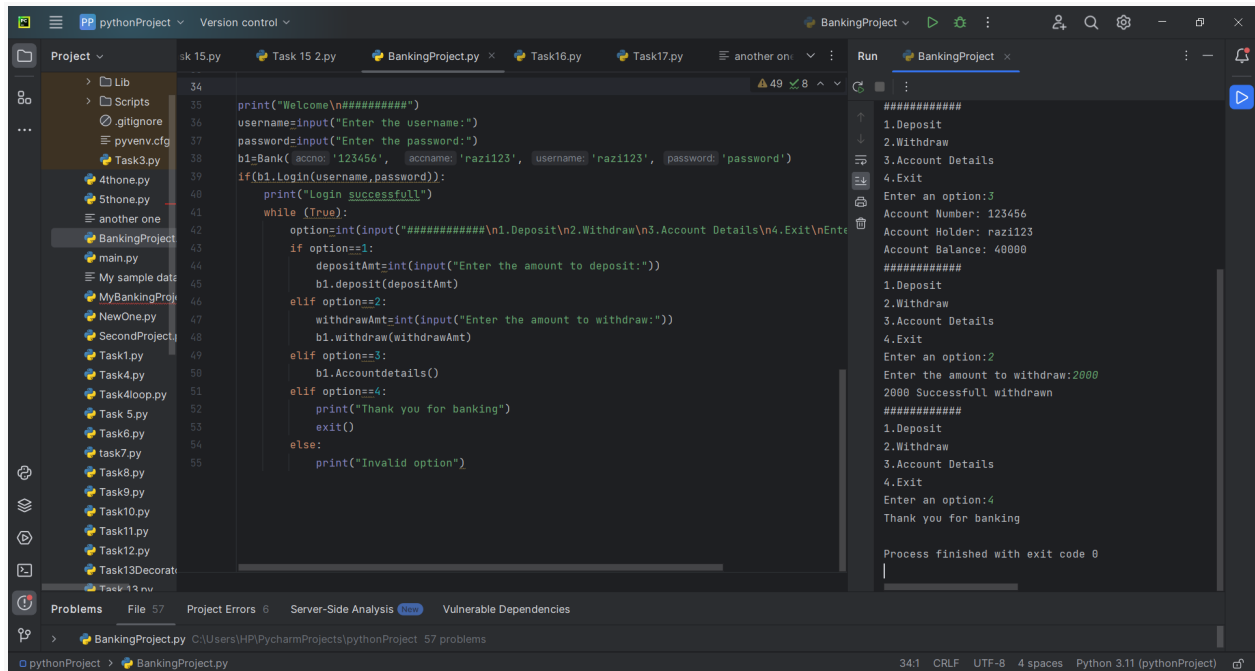
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

else:
    self.balance=self.balance-withamount
    print(withamount,"Successful withdrawn")

1 usage
def Accountdetails(self):
    print("Account Number:",self.accNo)
    print("Account Holder:",self.accholder)
    print("Account Balance:",self.balance)
    return

print("Welcome\n#####")
username=input("Enter the username:")
password=input("Enter the password:")
b1=Bank( accno: '123456', accname: 'razi123', username: 'razi123', password: 'password')
if(b1.Login(username,password)):
    print("Login successful")
    while (True):
        options=int(input("#####\n1.Deposit\n2.Withdraw\n3.Account Details\n4.Exit\nEnter an option:"))
        if option==1:
            depositAmt=int(input("Enter the amount to deposit:"))
            b1.deposit(depositAmt)
        elif option==2:
            withdrawAmt=int(input("Enter the amount to withdraw:"))
            b1.withdraw(withdrawAmt)
        elif option==3:
            b1.Accountdetails()

Problems 57 Project Errors 6 Server-Side Analysis Vulnerable Dependencies
> BankingProject.py C:\Users\HP\PycharmProjects\pythonProject\57 problems
pythonProject BankingProject.py 34:1 CRLF UTF-8 4 spaces Python 3.11 (pythonProject)
```



Task 18 Numpy

Task for class on 14-10-2023:

Numpy:

Exercise 1:

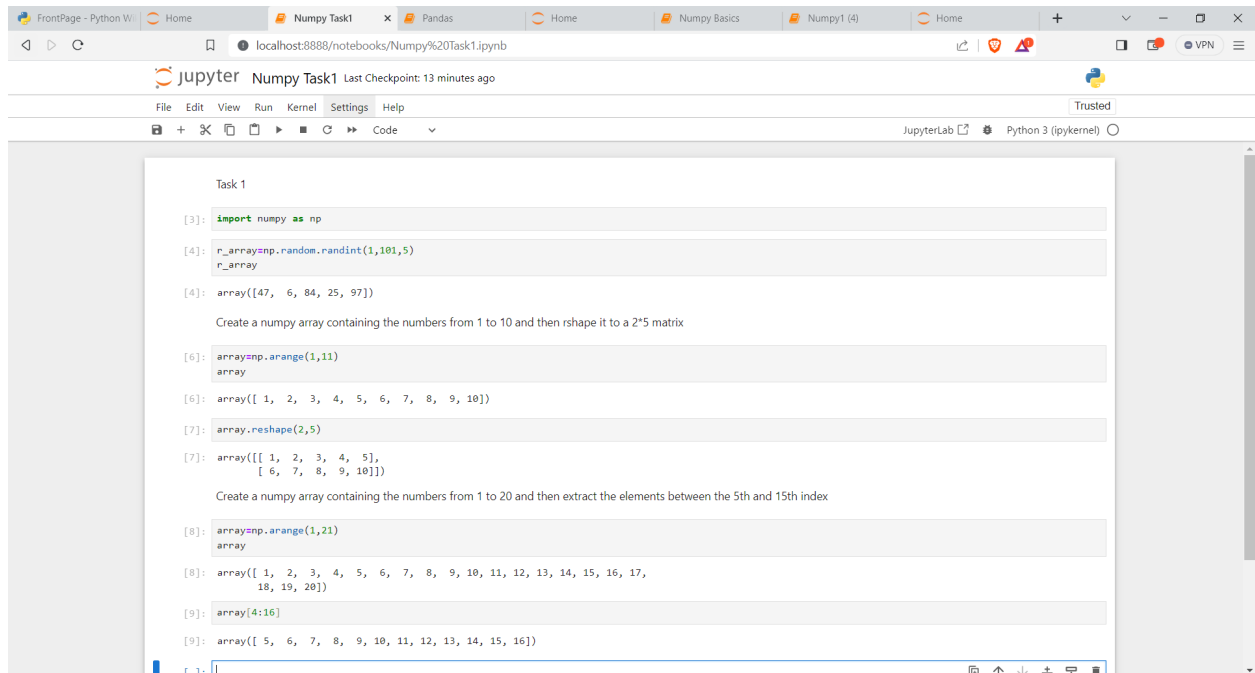
Create a numpy array with 5 elements consisting of random integers between 1 and 100.

Exercise 2:

Create a numpy array containing the numbers from 1 to 10, and then reshape it to a 2x5 matrix.

Exercise 3:

Create a numpy array containing the numbers from 1 to 20, and then extract the elements between the 5th and 15th index.



The screenshot shows a Jupyter Notebook interface with the title 'Numpy Task1'. The notebook contains several code cells and text annotations. The code cells are as follows:

```
[3]: import numpy as np
```

```
[4]: r_array=np.random.randint(1,101,5)
r_array
```

```
[4]: array([47,  6, 84, 25, 97])
```

Create a numpy array containing the numbers from 1 to 10 and then reshape it to a 2*5 matrix

```
[6]: array=np.arange(1,11)
array
```

```
[6]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

```
[7]: array.reshape(2,5)
```

```
[7]: array([[ 1,  2,  3,  4,  5],
          [ 6,  7,  8,  9, 10]])
```

Create a numpy array containing the numbers from 1 to 20 and then extract the elements between the 5th and 15th index

```
[8]: array=np.arange(1,21)
array
```

```
[8]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
          18, 19, 20])
```

```
[9]: array[4:16]
```

```
[9]: array([ 5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16])
```

Task 19 Pandas

Task for class on 17-10-2022:

Pandas:

Series

Exercise 1:

Create a pandas series containing the top 5 programming languages used in 2021, along with their respective popularity index (in descending order). Then, extract the third item from the series.

Exercise 2:

Create a Pandas series with the following data: [1, 2, 3, 4, 5]. Then, calculate the sum, mean, and standard deviation of the series.

Exercise 3:

Create a Pandas series with the following data: {'apples': 3, 'bananas': 2, 'oranges': 1}. Then, add a new item to the series with the key 'pears' and the value 4.

Task 19

Create a pandas series containing the top 5 programming languages used in 2021, along with their respective popularity index (in descending order). Then, extract the third item from the series.

```
[2]: import pandas as pd
```

```
[6]: lang=pd.Series({'Python':5,'c#':4,'JavaScript':3,'Java':2,'c++':1})
lang
```

```
[6]: Python    5
     c#        4
     JavaScript 3
     Java      2
     c++       1
     dtype: int64
```

```
[7]: #Extracting third element
     third_element=lang.index[2]
     third_element
```

```
[7]: 'JavaScript'
```

Create a Pandas series with the following data: [1, 2, 3, 4, 5]. Then, calculate the sum, mean, and standard deviation of the series.

```
[8]: series_data=pd.Series([1,2,3,4,5])
     series_data
```

```
[8]: 0    1
     1    2
     2    3
     3    4
     4    5
```

```
4    5
dtype: int64
```

```
[10]: sum=series_data.sum()
     sum
```

```
[10]: 15
```

```
[11]: mean=series_data.mean()
     mean
```

```
[11]: 3.0
```

```
[12]: Standard_deviation=series_data.std()
     Standard_deviation
```

```
[12]: 1.5811388300841898
```

Create a Pandas series with the following data: {'apples': 3, 'bananas': 2, 'oranges': 1}. Then, add a new item to the series with the key 'pears' and the value 4.

```
[13]: fruit_series=pd.Series({'apple':3,'banana':2,'oranges':1})
     fruit_series
```

```
[13]: apple    3
     banana  2
     oranges  1
     dtype: int64
```

```
[14]: fruit_series['pears']=4
     fruit_series
```

```
[14]: apple    3
     banana  2
     oranges  1
     pears    4
     dtype: int64
```

Task 20 Data Frame

Tasks for class on 21-11-2023:

Dataframe

Exercise 1:

Create a dataframe with the following columns: name, age, and gender. The dataframe should have 10 rows of data. Make a sample data.

Exercise 2:

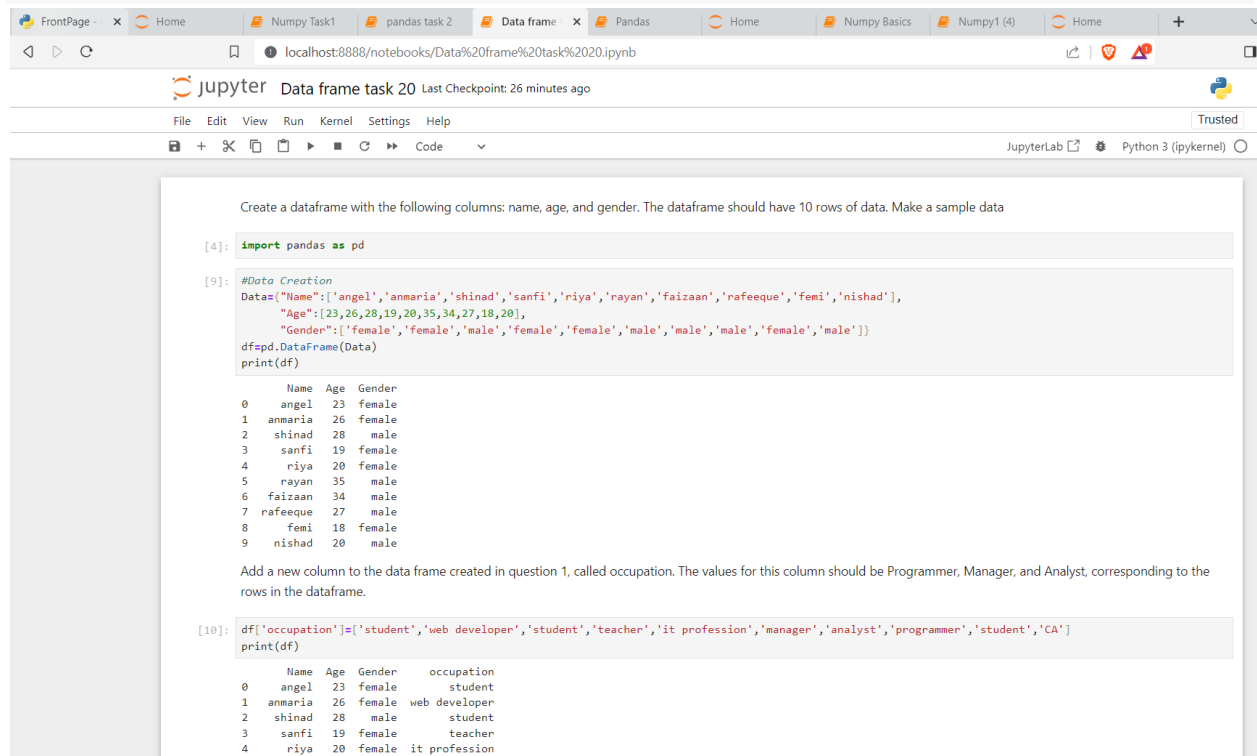
Add a new column to the data frame created in question 1, called occupation. The values for this column should be Programmer, Manager, and Analyst, corresponding to the rows in the dataframe.

Exercise 3:

Select the rows of the dataframe where the age is greater than or equal to 30.

Exercise 4:

Sort the data frame by age in descending order



The screenshot shows a JupyterLab interface with a notebook titled "Data frame task 20". The notebook contains two code cells. The first cell, labeled [4], imports pandas as pd and creates a DataFrame with 10 rows of sample data. The second cell, labeled [10], adds a new column named 'occupation' to the DataFrame with values: 'student', 'web developer', 'student', 'teacher', 'it profession', 'manager', 'analyst', 'programmer', 'student', 'CA'. The output of the first cell shows the DataFrame with columns Name, Age, and Gender. The output of the second cell shows the DataFrame with the new 'occupation' column added.

```
[4]: import pandas as pd

[9]: #Data Creation
Data={"Name":["angel","anmaria","shinad","sanfi","riya","rayan","faizaan","rafeeqe","femi","nishad"],
      "Age":[23,26,28,19,20,35,34,27,18,20],
      "Gender":["female","female","male","female","female","male","male","male","female","male"]}
df=pd.DataFrame(Data)
print(df)
```

	Name	Age	Gender
0	angel	23	female
1	anmaria	26	female
2	shinad	28	male
3	sanfi	19	female
4	riya	20	female
5	rayan	35	male
6	faizaan	34	male
7	rafeeqe	27	male
8	femi	18	female
9	nishad	20	male

Create a dataframe with the following columns: name, age, and gender. The dataframe should have 10 rows of data. Make a sample data

```
[10]: df['occupation']=['student','web developer','student','teacher','it profession','manager','analyst','programmer','student','CA']
print(df)
```

	Name	Age	Gender	occupation
0	angel	23	female	student
1	anmaria	26	female	web developer
2	shinad	28	male	student
3	sanfi	19	female	teacher
4	riya	20	female	it profession

```
2  shinad  28  male  student
3  sanfi   19  female teacher
4  riya    20  female it profession
5  rayan   35  male  manager
6  faizaan 34  male  analyst
7  rafeeqe 27  male  programmer
8  femi    18  female student
9  nishad  20  male  CA

Select the rows of the dataframe where the age is greater than or equal to 30.

[13]: rows=df[df['Age']>=30]
      print(rows)

      Name  Age  Gender  occupation
5  rayan   35  male  manager
6  faizaan 34  male  analyst

Sort the data frame by age in descending order

[14]: df_sort=df.sort_values(by='Age',ascending=False)
      print(df_sort)

      Name  Age  Gender  occupation
5  rayan   35  male  manager
6  faizaan 34  male  analyst
2  shinad  28  male  student
7  rafeeqe 27  male  programmer
1  anmaria 26  female web developer
0  angel   23  female student
4  riya    20  female it profession
9  nishad  20  male  CA
3  sanfi   19  female teacher
8  femi    18  female student
```

Task 21 Matplotlib

Task for class on 28-11-2023:

Matplotlib:

Exercise 1:

Create a line plot using matplotlib pyplot that displays the population of four different cities over time. Each city should have its own line, and the x-axis should represent years (e.g. 2010, 2011, 2012, etc.) while the y-axis should represent the population.

The data for the four cities is provided below:

City A: [500000, 550000, 600000, 650000, 700000, 750000, 800000]

City B: [800000, 850000, 900000, 950000, 1000000, 1050000, 1100000]

City C: [1000000, 1050000, 1100000, 1150000, 1200000, 1250000, 1300000]

City D: [1200000, 1250000, 1300000, 1350000, 1400000, 1450000, 1500000]

Exercise 2:

Create a scatter plot using matplotlib pyplot that shows the relationship between the number of hours studied and the test scores obtained by a group of students. Use the following data:

Hours Studied: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

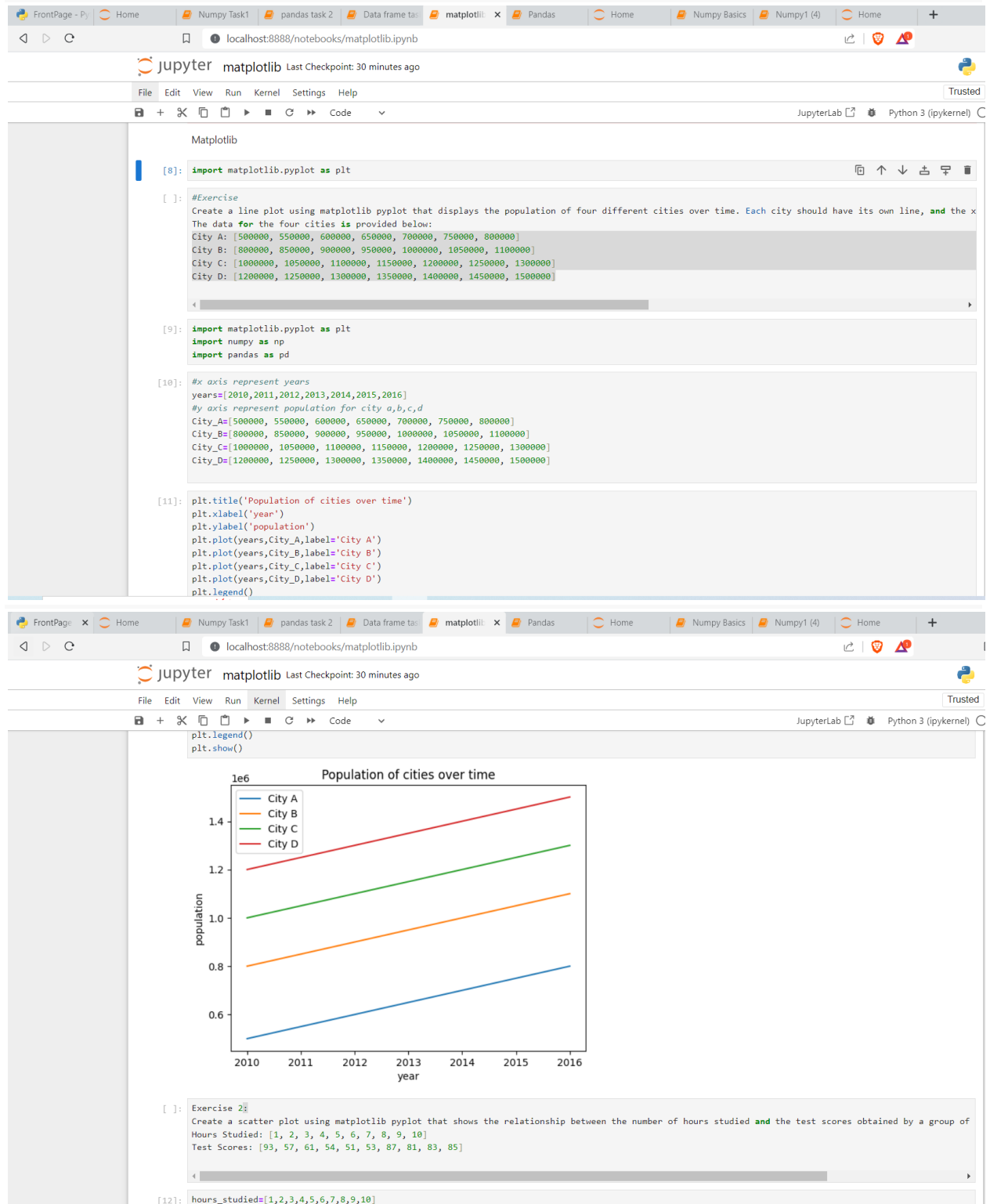
Test Scores: [93, 57, 61, 54, 51, 53, 87, 81, 83, 85]

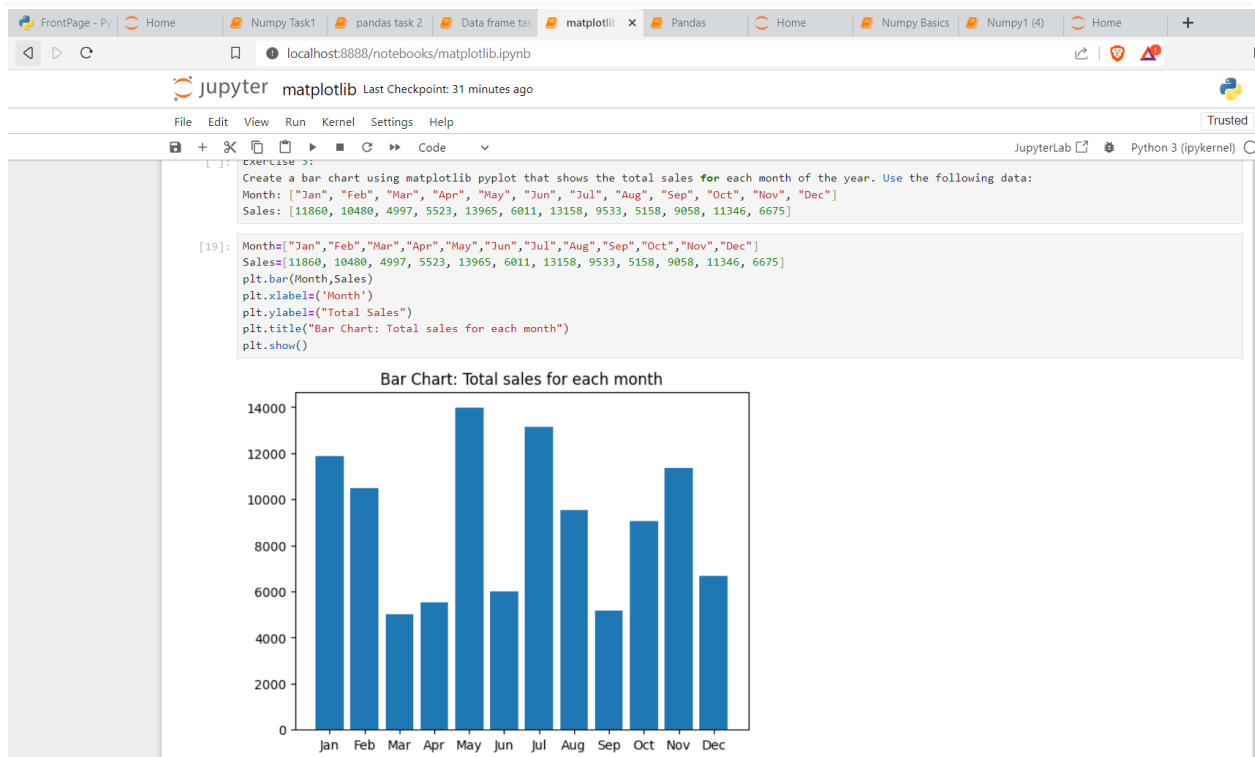
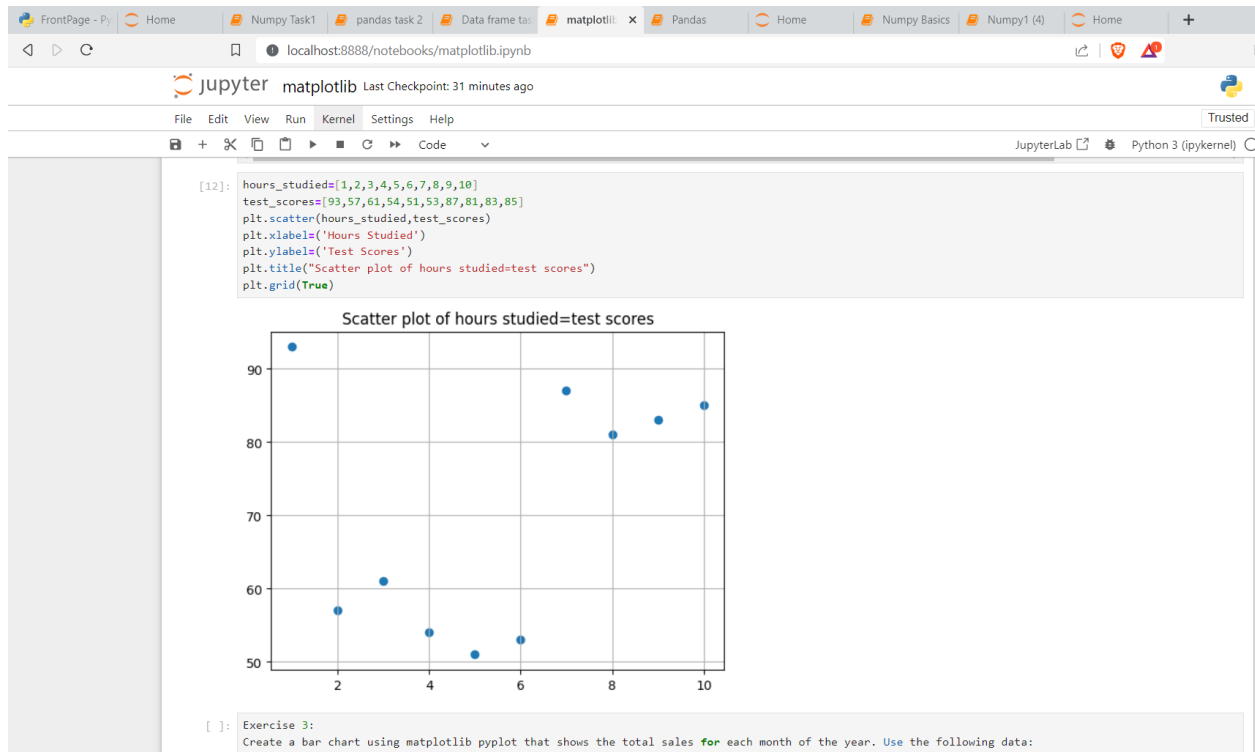
Exercise 3:

Create a bar chart using matplotlib pyplot that shows the total sales for each month of the year. Use the following data:

Month: ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]

Sales: [11860, 10480, 4997, 5523, 13965, 6011, 13158, 9533, 5158, 9058, 11346, 6675]





Task 22 Matplotlib and Seaborn

Task for class on 05-12-2023:

Matplotlib and Seaborn:

Using the below dataset of fifa players dataset. Perform Exploratory data analysis and find the following insights:

- 1.Which country has the most number of players
- 2.Plot a bar chart of 5 top countries with most number of players
- 3.Which player has the highest salary?
- 4.Plot a histogram to get the salary range of the players.
- 5.Who is the tallest player in the fifa players?
- 6.Which club has most number of players?
- 7.Which foot is most preferred by the players?Draw a bar chart for preferred foot.

https://drive.google.com/file/d/10oyIT1KPdwUqeU9-2LX0xE5-ZytNn9su/view?usp=share_link

FrontPage - Python Wiki Home Matplotlib and seaborn x fifa_data.csv - Google Drive New Tab fifa_data.csv - Google Drive

localhost:8888/notebooks/Matplotlib%20and%20seaborn.ipynb

Jupyter Matplotlib and seaborn Last Checkpoint: 23 hours ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[10]: import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

[18]: fifa_data=pd.read_csv("C:\\jupyternotebook\\fifa_data.csv")
fifa_data

[18]:
```

	Unnamed: 0	ID	Name	Age	Photo	Nationality	Flag	Overall	Potential	Club	
	0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentina	https://cdn.sofifa.org/flags/52.png	94	94	FC Barcelona
	1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portugal	https://cdn.sofifa.org/flags/38.png	94	94	Juventus
	2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Brazil	https://cdn.sofifa.org/flags/54.png	92	93	Paris Saint-Germain
	3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spain	https://cdn.sofifa.org/flags/45.png	91	93	Manchester United
	4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgium	https://cdn.sofifa.org/flags/7.png	91	92	Manchester City

	18202	18202	238813	J. Lundstram	19	https://cdn.sofifa.org/players/4/19/238813.png	England	https://cdn.sofifa.org/flags/14.png	47	65	Crewe Alexandra
	18203	18203	243165	N. Christofferson	19	https://cdn.sofifa.org/players/4/19/243165.png	Sweden	https://cdn.sofifa.org/flags/46.png	47	63	Trelleborgs FF
	18204	18204	241638	B. Worman	16	https://cdn.sofifa.org/players/4/19/241638.png	England	https://cdn.sofifa.org/flags/14.png	47	67	Cambridge United
	18205	18205	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	England	https://cdn.sofifa.org/flags/14.png	47	66	Tranmere Rovers
	18206	18206	246268	D. Walker-Rice	17	https://cdn.sofifa.org/players/4/19/246268.png	England	https://cdn.sofifa.org/flags/14.png	47	66	Tranmere Rovers

FrontPage - Python Wiki Home Matplotlib and seaborn x fifa_data.csv - Google Drive New Tab fifa_data.csv - Google Drive

localhost:8888/notebooks/Matplotlib%20and%20seaborn.ipynb

Jupyter Matplotlib and seaborn Last Checkpoint: 23 hours ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Which country has the most numbers of players

```
[19]: Most_players=fifa_data["Nationality"].value_counts().idxmax()
Most_players

[19]: 'England'
```


Plot a bar chart of 5 top countries with most numbers of players

```
[20]: top5_countries=fifa_data['Nationality'].value_counts().head()
top5_countries

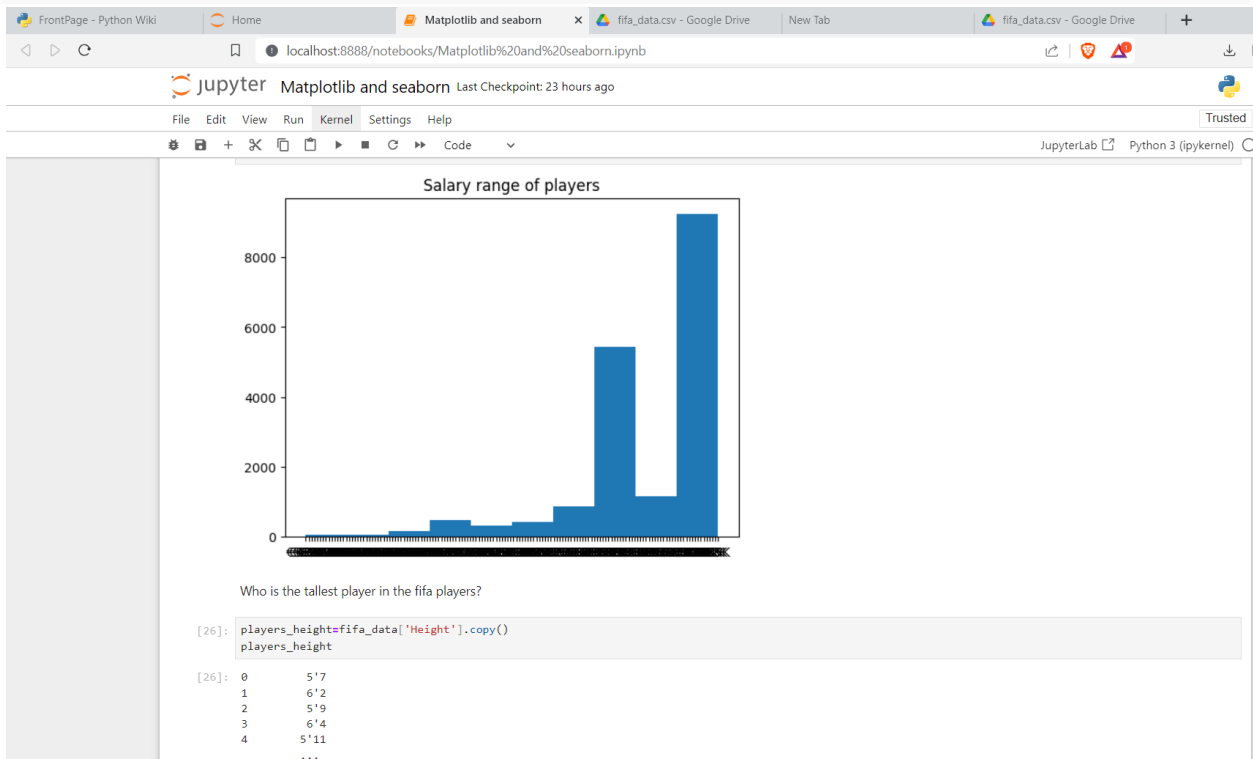
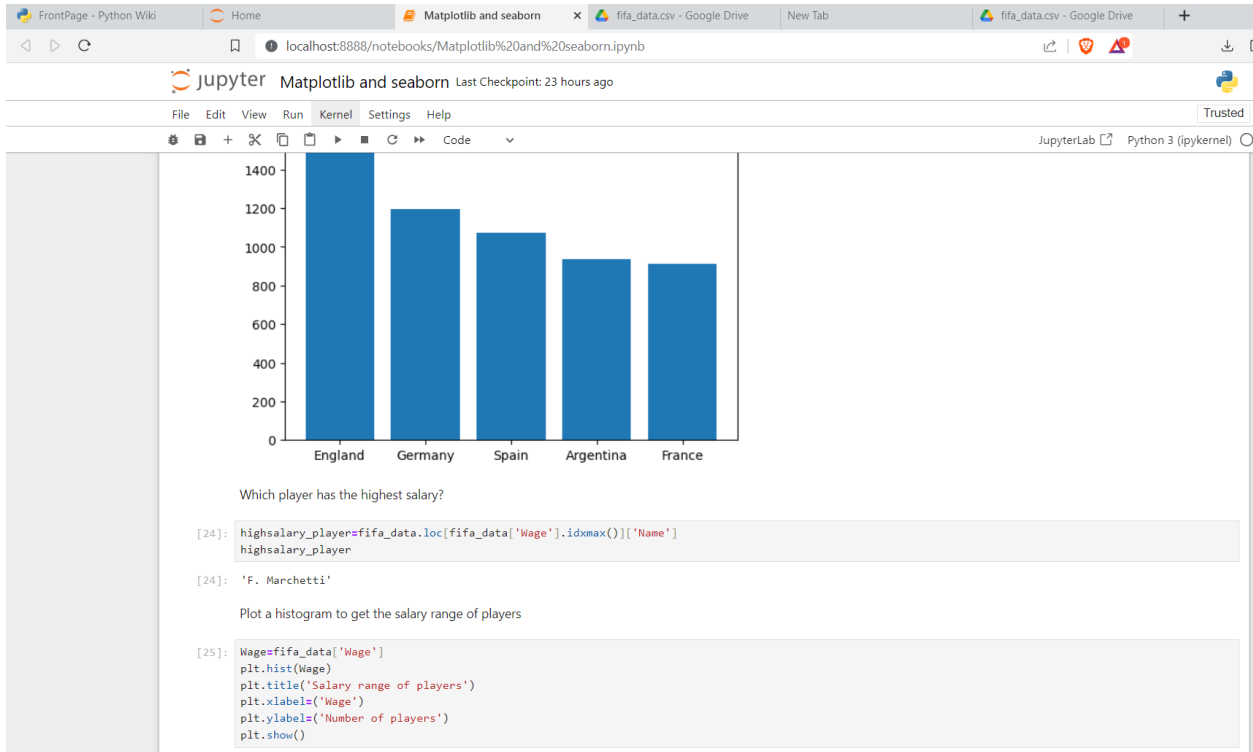
[20]: Nationality
England    1662
Germany   1198
Spain     1072
Argentina  937
France    914
Name: count, dtype: int64

[21]: top_countries=['England','Germany','Spain','Argentina','France']
players=[1662,1198,1072,937,914]
plt.bar(top_countries,players)
plt.title('Top 5 countries with most number of players')
plt.xlabel('Top 5 countries')
plt.ylabel('No. of players')
plt.show()
```

Top 5 countries with most number of players



Country	No. of players
England	1662
Germany	1198
Spain	1072
Argentina	937
France	914



```

3         6'4
4         5'11
...
18202     5'9
18203     6'3
18204     5'8
18205     5'10
18206     5'10
Name: Height, Length: 18207, dtype: object

```

```

[28]: def feetto_cm(height):
      if isinstance(height,str):
          feet,inches=map(int,height.split("'"))
          return feet*12+inches
      else:
          return height
      fifa_data['Height(inches)']=fifa_data['Height'].apply(feetto_cm)
      fifa_data[['Height','Height(inches)']]

```

```

[28]:
      Height  Height(inches)
0         5'7             67.0
1         6'2             74.0
2         5'9             69.0
3         6'4             76.0
4         5'11            71.0
...      ...              ...
18202     5'9             69.0
18203     6'3             75.0
18204     5'8             68.0
18205     5'10            70.0

```

```

18205     5'10            70.0
18206     5'10            70.0

```

18207 rows × 2 columns

To find the tallest player

```

[30]: tallest_players=fifa_data.loc[fifa_data['Height(inches)'].idxmax()]
      tallest_player[["Name","Height"]]

```

```

[30]: Name      T. Holý
      Height      6'9
      Name: 11614, dtype: object

```

Which club has most number of players?

```

[31]: most_num_players=fifa_data['Club'].value_counts().max()
      club_most_no_players=fifa_data['Club'].value_counts()[fifa_data['Club'].value_counts()==most_num_players]
      print(club_most_no_players)

```

```

Club
FC Barcelona      33
Valencia CF       33
Fortuna Düsseldorf 33
Cardiff City      33
Rayo Vallecano    33
CD Leganés        33
Frosinone         33
Newcastle United  33
Southampton      33
Burnley          33
Eintracht Frankfurt 33
Wolverhampton Wanderers 33
TSG 1899 Hoffenheim 33
Everton          33
AS Monaco         33

```

```
Frosinone          33
Newcastle United   33
Southampton        33
Burnley            33
Eintracht Frankfurt 33
Wolverhampton Wanderers 33
TSG 1899 Hoffenheim 33
Everton            33
AS Monaco          33
RC Celta           33
Empoli             33
Manchester City     33
Manchester United   33
Borussia Dortmund  33
Real Madrid        33
Atlético Madrid    33
Tottenham Hotspur  33
Chelsea            33
Liverpool          33
Arsenal            33
Name: count, dtype: int64
```

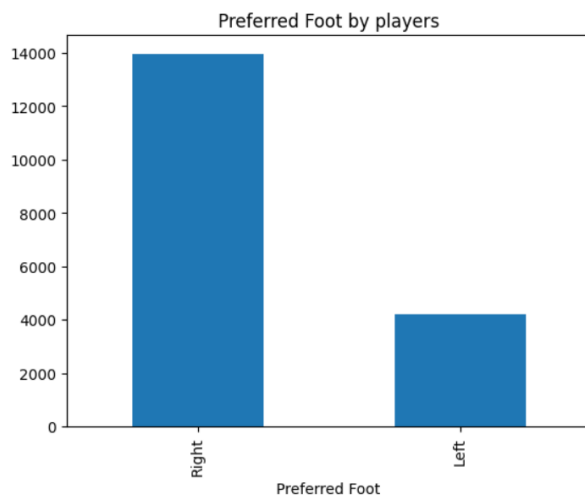
Which foot is most preferred by the players? Draw a bar chart for preferred foot.

```
[32]: preferred_foot=fifa_data.value_counts('Preferred Foot')
      preferred_foot
```

```
[32]: Preferred Foot
      Right    13948
      Left     4211
      Name: count, dtype: int64
```

```
[37]: preferred_foot.plot.bar()
      plt.title('Preferred Foot by players')
      plt.xlabel('Preferred foot')
      plt.ylabel('Number of players')
      plt.show()
```

```
plt.ylabel('Number of players')
plt.show()
```



[]:

