Practical No.2

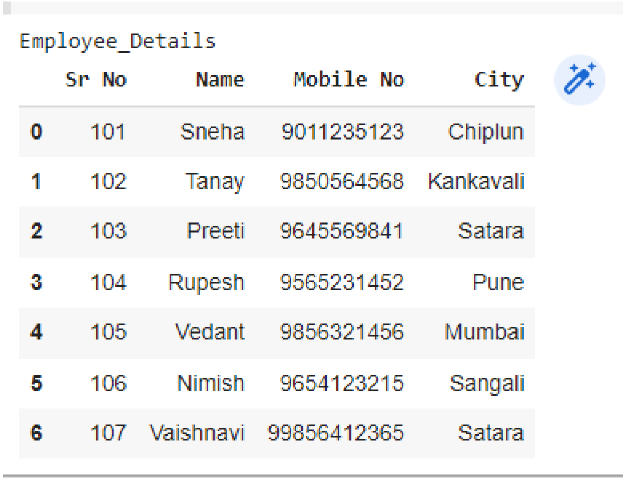
1. Create Pandas DataFrame to enter employee database which includes Sr No, Name, Mobile No, City

Implementation:

- **Program:**

```
#pandas dataframe
#Employee
print("Employee_Details")
import pandas as pd
pd.DataFrame({"Sr No":[101,102,103,104,105,106,107],
              "Name":['Sneha','Tanay','Preeti','Rupesh','Vedant','Nimis
h','Vaishnavi'],
              "Mobile
No":[9011235123,9850564568,9645569841,9565231452,9856321456,9654123215,
99856412365],
              "City":['Chiplun','Kankavali','Satara','Pune','Mumbai','S
angali','Satara']})
```

- **Output:**



```
Employee_Details
   Sr No       Name   Mobile No       City
0    101      Sneha  9011235123    Chiplun
1    102      Tanay  9850564568  Kankavali
2    103     Preeti  9645569841     Satara
3    104     Rupesh  9565231452       Pune
4    105     Vedant  9856321456     Mumbai
5    106     Nimish  9654123215    Sangali
6    107  Vaishnavi 99856412365     Satara
```

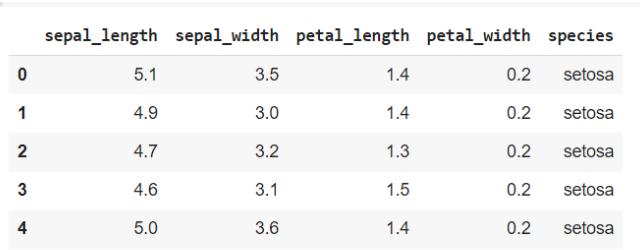
2. Use Iris Dataset from Github and perform all basic operations on Iris Dataset

Implementation:

- **Program:**

```
url='https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv'
iris2=pd.read_csv(url)
iris2.head()
```

- **Output:**

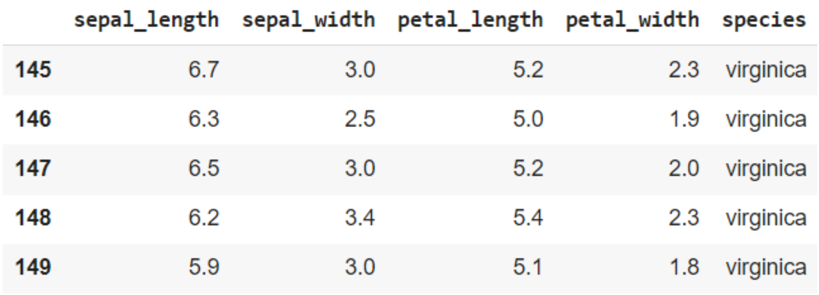


	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

- **Program:**

```
url='https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv'
iris3=pd.read_csv(url)
iris3.tail()
```

- **Output:**

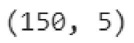


	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

- **Program:**

```
iris2.shape
```

- **Output:**

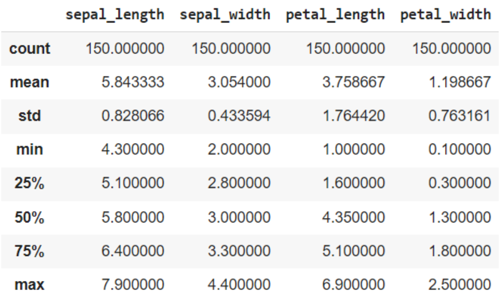


```
(150, 5)
```

- **Program:**

```
iris2.describe()
```

- **Output:**

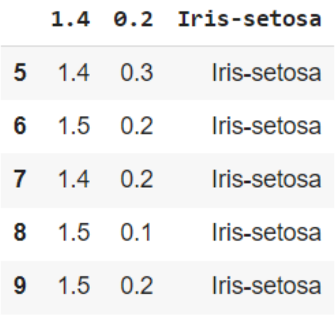


	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

- **Program:**

```
iris1.iloc[5:10,2:]
```

- **Output:**

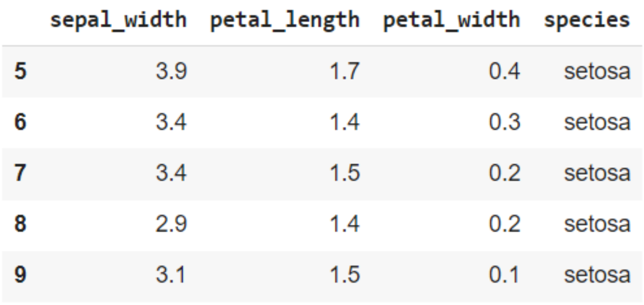


	1.4	0.2	Iris-setosa
5	1.4	0.3	Iris-setosa
6	1.5	0.2	Iris-setosa
7	1.4	0.2	Iris-setosa
8	1.5	0.1	Iris-setosa
9	1.5	0.2	Iris-setosa

- **Program:**

```
iris2.iloc[5:10,1:]
```

- **Output:**

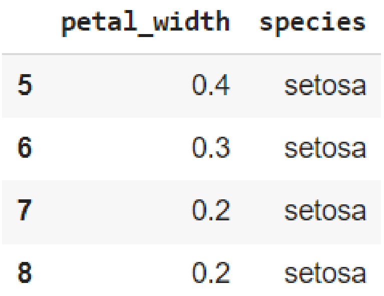


	sepal_width	petal_length	petal_width	species
5	3.9	1.7	0.4	setosa
6	3.4	1.4	0.3	setosa
7	3.4	1.5	0.2	setosa
8	2.9	1.4	0.2	setosa
9	3.1	1.5	0.1	setosa

- **Program:**

```
iris2.iloc[5:9,3:]
```

- **Output:**

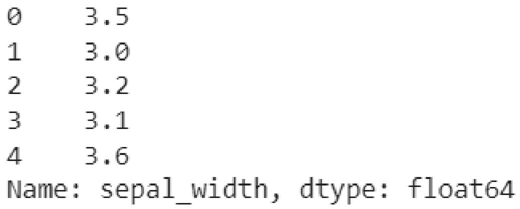


	petal_width	species
5	0.4	setosa
6	0.3	setosa
7	0.2	setosa
8	0.2	setosa

- **Program:**

```
iris2.loc[0:4,"sepal_width"]
```

- **Output:**

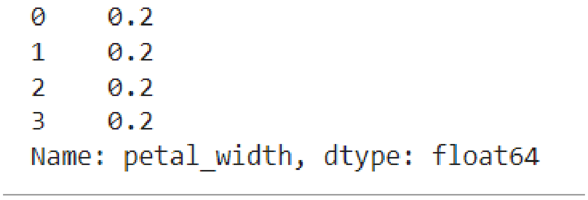


```
0    3.5
1    3.0
2    3.2
3    3.1
4    3.6
Name: sepal_width, dtype: float64
```

- **Program:**

```
iris2.loc[0:3,"petal_width"]
```

- **Output:**

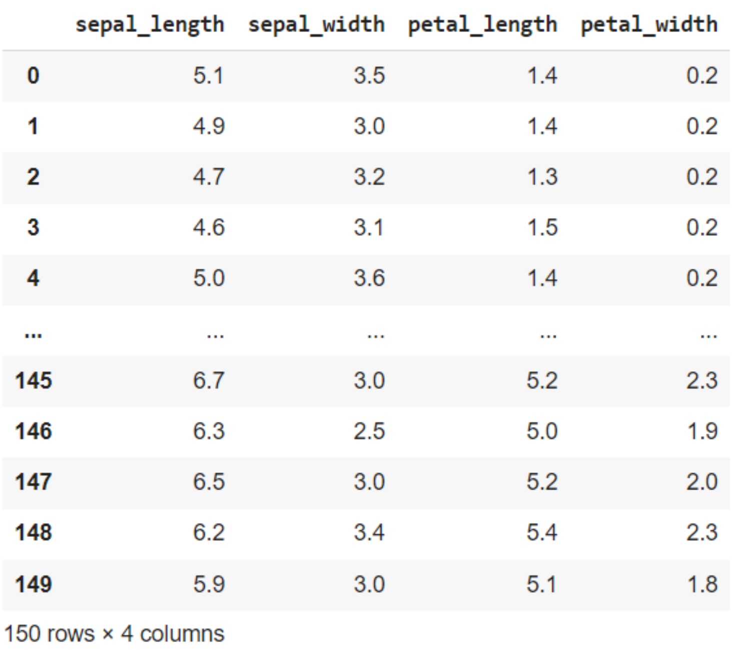


```
0    0.2
1    0.2
2    0.2
3    0.2
Name: petal_width, dtype: float64
```

- **Program:**

```
iris2.drop('species',axis=1)
```

- **Output:**



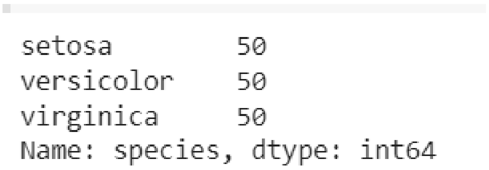
	sepal_length	sepal_width	petal_length	petal_width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

- **Program:**

```
iris2['species'].value_counts()
```

- **Output:**



```
setosa        50
versicolor    50
virginica     50
Name: species, dtype: int64
```

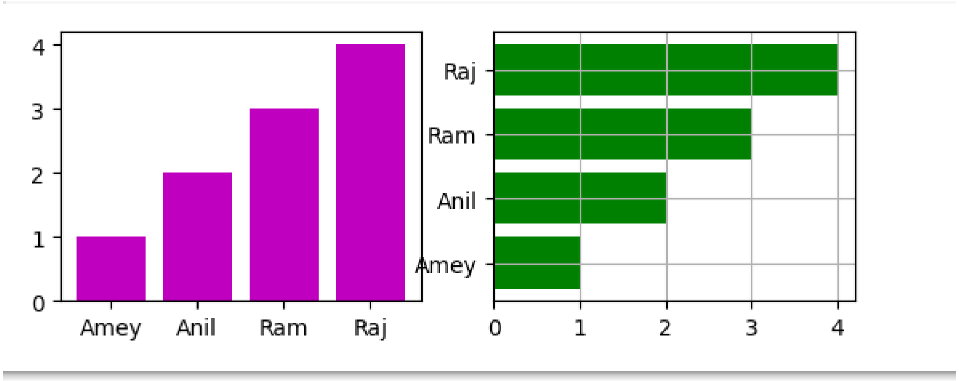
3. Use matplotlib to plot bar chart using dictionary

Implementation:

- **Program:**

```
#Bar Chart
student={"Amey":1,"Anil":2,"Ram":3,"Raj":4}

name=list(student.keys())
rollno=list(student.values())

plt.subplot(2,2,1)
plt.bar(name,rollno,color='m')

plt.subplot(2,2,2)
plt.barh(name,rollno,color='g')

plt.grid(True)
plt.show()
```

- **Output:**



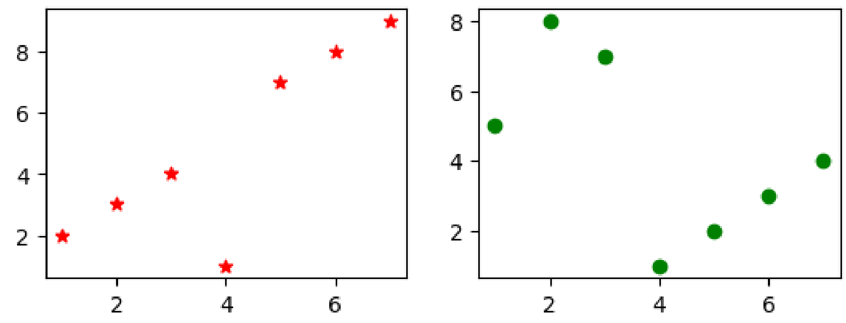
4. Use matplotlib to scatter graph and histogram

Implementation:

- **Program:**

```
#Scatter Graph
x=[1,2,3,4,5,6,7]
y=[2,3,4,1,7,8,9]
y1=[5,8,7,1,2,3,4]

plt.subplot(2,2,1)
plt.scatter(x,y,color='r',marker='*')

plt.subplot(2,2,2)
plt.scatter(x,y1,color='g')
plt.show()
```

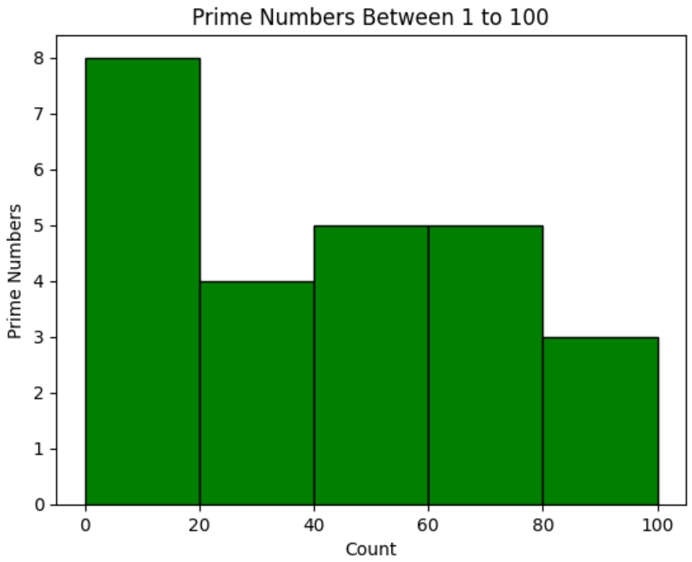
- **Output:**



- **Program:**

```
#Histogram
primeNo=np.array([2,3,5,7,11,13,17,19,23,29,37,31,41,43,47,53,59,61,67,
71,73,79,83,89,97])

bins=[0,20,40,60,80,100]
plt.title("Prime Numbers Between 1 to 100")
plt.ylabel("Prime Numbers")
plt.xlabel("Count")

plt.hist(primeNo,bins,color='g',edgecolor='k')
plt.show()
```

- **Output:**


Prime Numbers Between 1 to 100
Prime Numbers
Count