Assignment-3

Name: Jagadeesh Pradhan

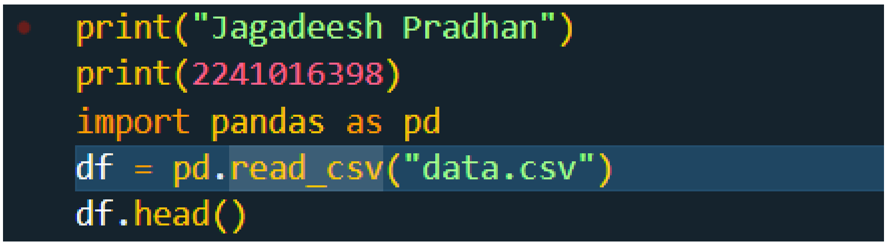
Reg No: 2241016398

Sl No:33

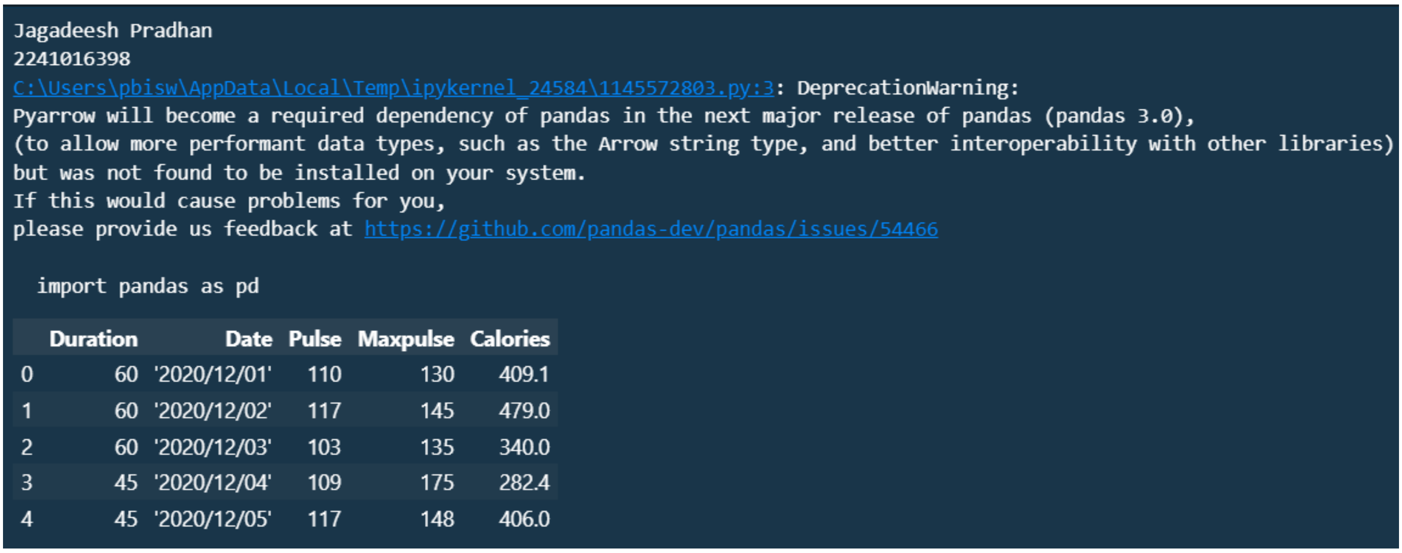
Q-1) Download the dataset given in the following link in your local repository. Read the dataset and clean it if it has some missing values, wrong data, wrong formats or duplicate values. Clean the dataset and print the 5 to 15 rows of the data. Finally, save the clean dataset in your local repository.
Dataset: data.csv

Answer)

Import The Data csv File

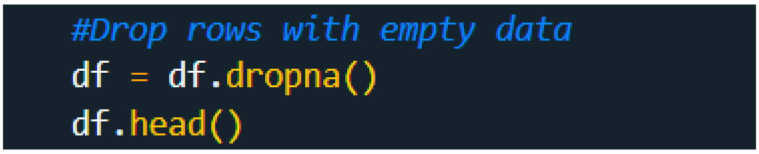


```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
df = pd.read_csv("data.csv")
df.head()
```

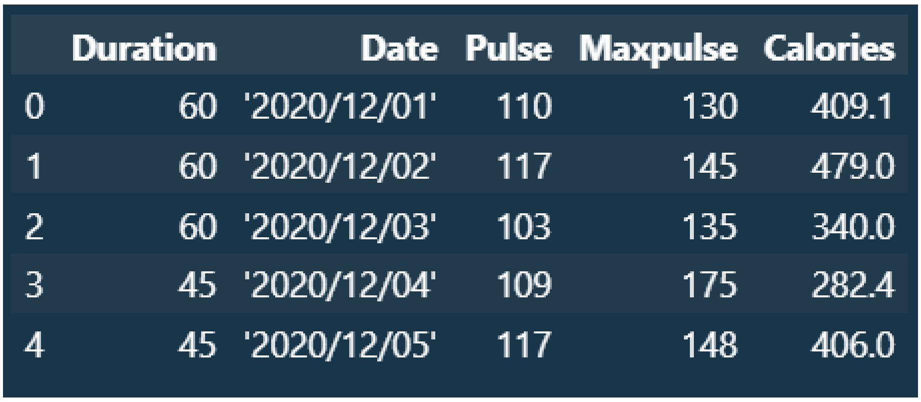


```
Jagadeesh Pradhan
2241016398
C:\Users\pbisw\AppData\Local\Temp\ipykernel_24584\1145572803.py:3: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

	Duration	Date	Pulse	Maxpulse	Calories
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0

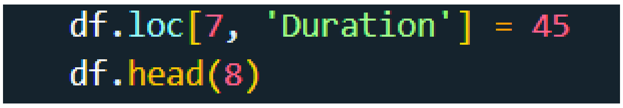


```
#Drop rows with empty data
df = df.dropna()
df.head()
```

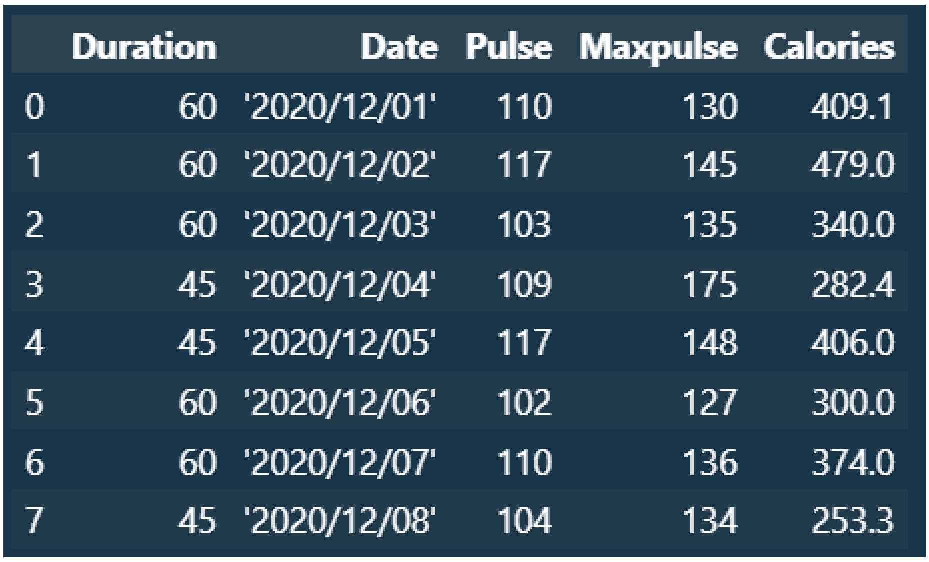


	Duration	Date	Pulse	Maxpulse	Calories
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0

Change The value 450 into 45

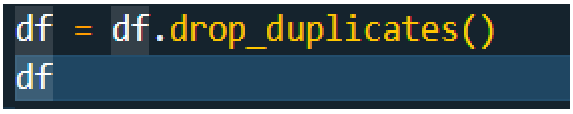


```
df.loc[7, 'Duration'] = 45
df.head(8)
```

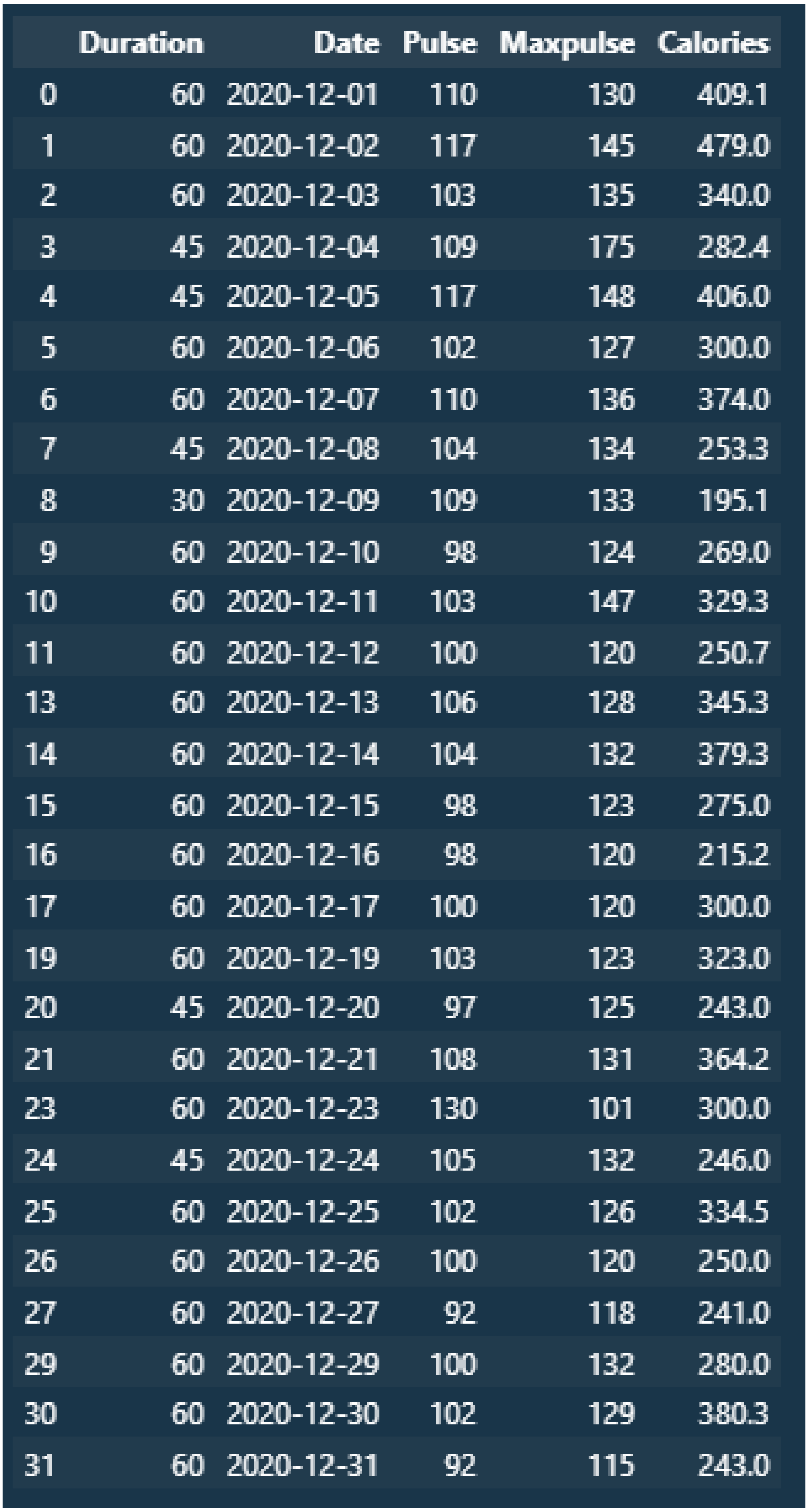


	Duration	Date	Pulse	Maxpulse	Calories
0	60	'2020/12/01'	110	130	409.1
1	60	'2020/12/02'	117	145	479.0
2	60	'2020/12/03'	103	135	340.0
3	45	'2020/12/04'	109	175	282.4
4	45	'2020/12/05'	117	148	406.0
5	60	'2020/12/06'	102	127	300.0
6	60	'2020/12/07'	110	136	374.0
7	45	'2020/12/08'	104	134	253.3

Delete The duplicates Rows

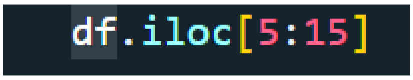


```
df = df.drop_duplicates()
df
```

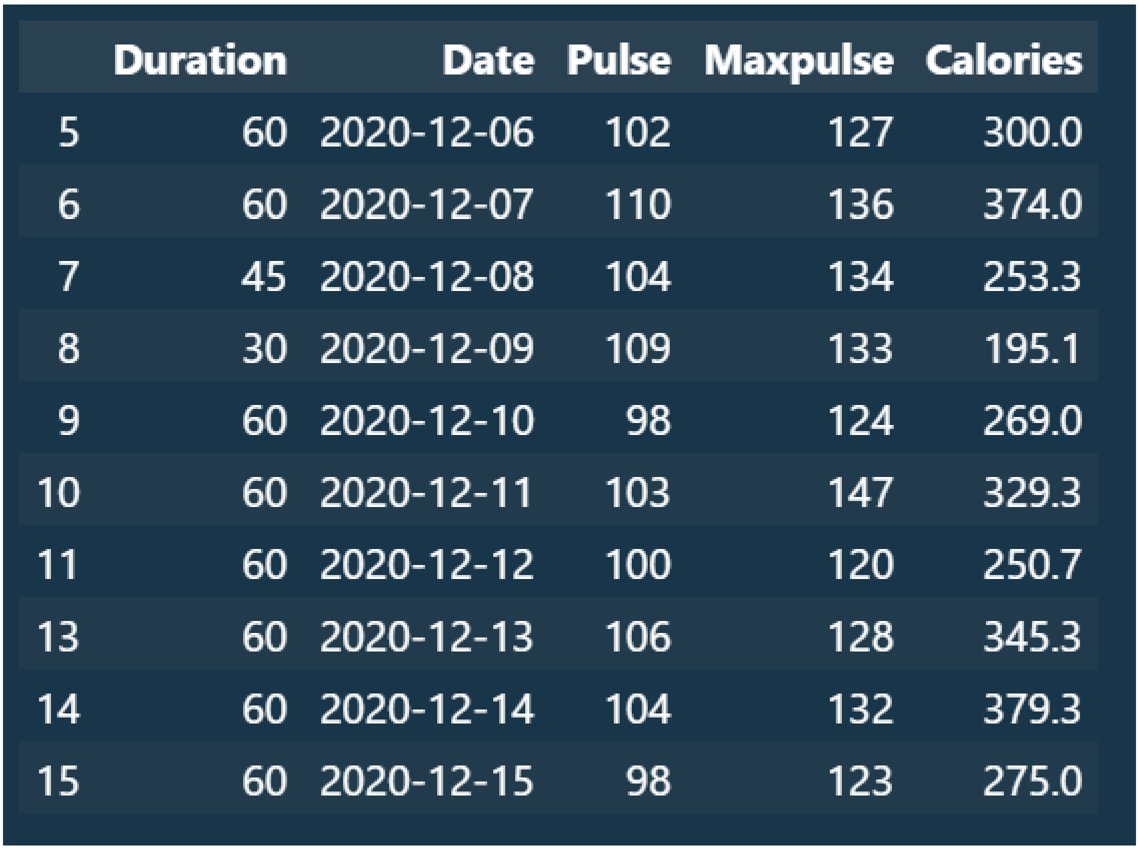


	Duration	Date	Pulse	Maxpulse	Calories
0	60	2020-12-01	110	130	409.1
1	60	2020-12-02	117	145	479.0
2	60	2020-12-03	103	135	340.0
3	45	2020-12-04	109	175	282.4
4	45	2020-12-05	117	148	406.0
5	60	2020-12-06	102	127	300.0
6	60	2020-12-07	110	136	374.0
7	45	2020-12-08	104	134	253.3
8	30	2020-12-09	109	133	195.1
9	60	2020-12-10	98	124	269.0
10	60	2020-12-11	103	147	329.3
11	60	2020-12-12	100	120	250.7
13	60	2020-12-13	106	128	345.3
14	60	2020-12-14	104	132	379.3
15	60	2020-12-15	98	123	275.0
16	60	2020-12-16	98	120	215.2
17	60	2020-12-17	100	120	300.0
19	60	2020-12-19	103	123	323.0
20	45	2020-12-20	97	125	243.0
21	60	2020-12-21	108	131	364.2
23	60	2020-12-23	130	101	300.0
24	45	2020-12-24	105	132	246.0
25	60	2020-12-25	102	126	334.5
26	60	2020-12-26	100	120	250.0
27	60	2020-12-27	92	118	241.0
29	60	2020-12-29	100	132	280.0
30	60	2020-12-30	102	129	380.3
31	60	2020-12-31	92	115	243.0

Print The row from 5 to 15:



```
df.iloc[5:15]
```



	Duration	Date	Pulse	Maxpulse	Calories
5	60	2020-12-06	102	127	300.0
6	60	2020-12-07	110	136	374.0
7	45	2020-12-08	104	134	253.3
8	30	2020-12-09	109	133	195.1
9	60	2020-12-10	98	124	269.0
10	60	2020-12-11	103	147	329.3
11	60	2020-12-12	100	120	250.7
13	60	2020-12-13	106	128	345.3
14	60	2020-12-14	104	132	379.3
15	60	2020-12-15	98	123	275.0

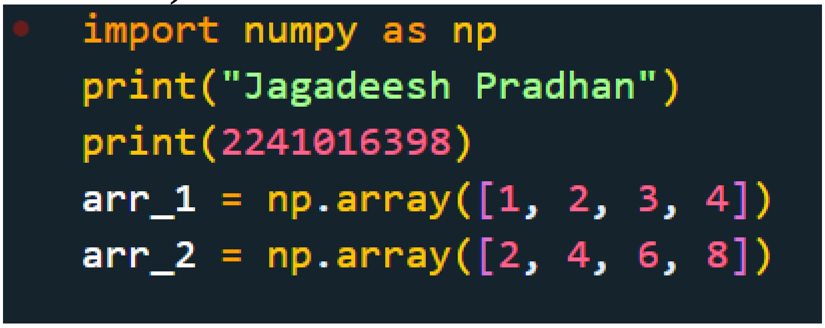
Q-2) Given two arrays, arr_1 = np.array([1, 2, 3, 4]), arr_2 = np.array([2, 4, 6, 8]). Print the results of the following operations: Add, Subtract, Multiply, Divide. Use NumPy

A. Generate a random integer 2D array with three rows and four columns named arr_3, with four values between 0 and 100.

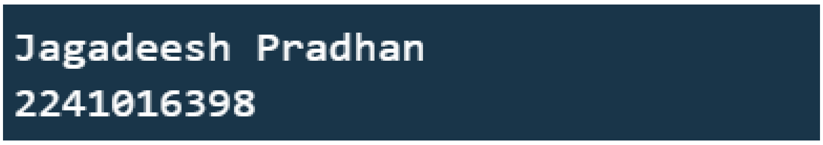
B. print the exponential and logarithmic values of all elements in array arr_1.

C. Given an array arr_4 = np.array(1.2, 2.5, 5.6, 3.4, 7.8], print the round-down (floor) and round-up (ceil) values.

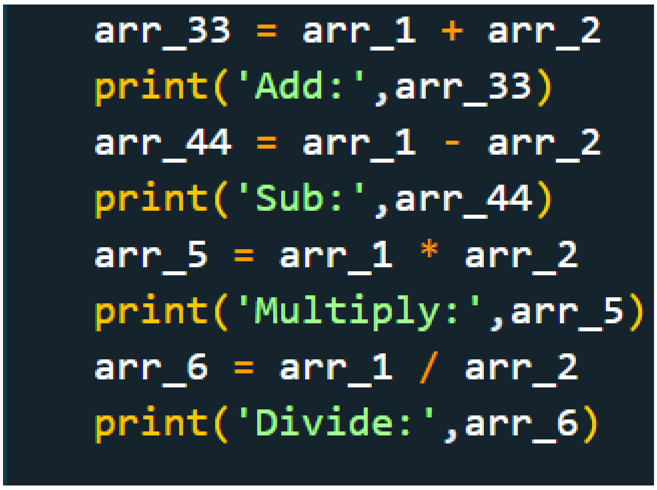
Answer)



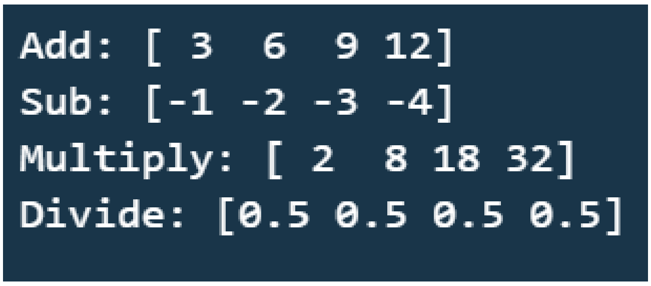
```
import numpy as np
print("Jagadeesh Pradhan")
print(2241016398)
arr_1 = np.array([1, 2, 3, 4])
arr_2 = np.array([2, 4, 6, 8])
```



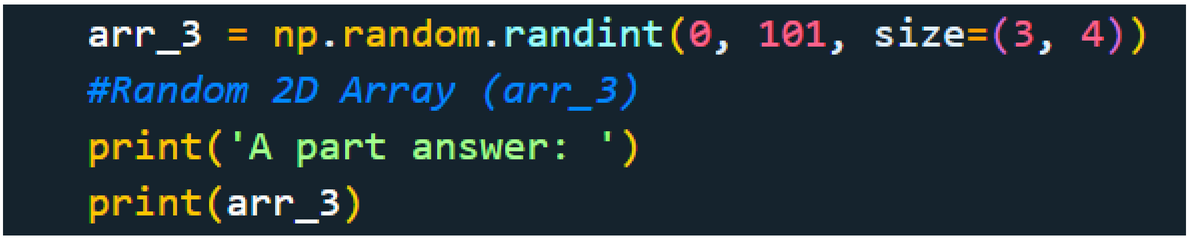
```
Jagadeesh Pradhan
2241016398
```



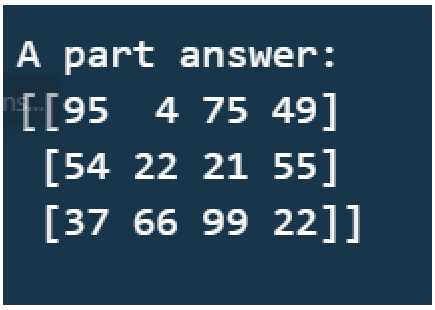
```
    arr_33 = arr_1 + arr_2
    print('Add:',arr_33)
    arr_44 = arr_1 - arr_2
    print('Sub:',arr_44)
    arr_5 = arr_1 * arr_2
    print('Multiply:',arr_5)
    arr_6 = arr_1 / arr_2
    print('Divide:',arr_6)
```



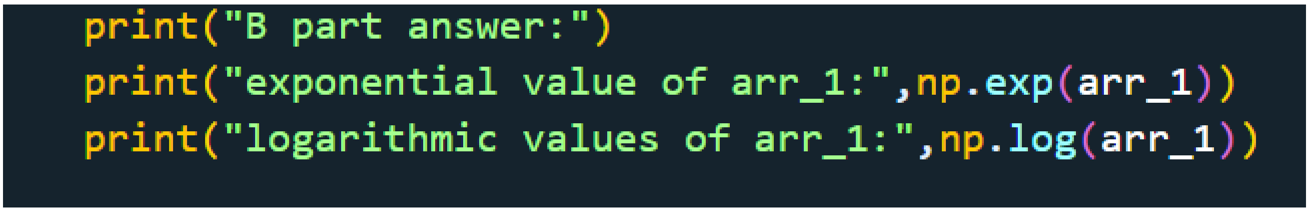
```
Add: [ 3  6  9 12]
Sub: [-1 -2 -3 -4]
Multiply: [ 2  8 18 32]
Divide: [0.5 0.5 0.5 0.5]
```



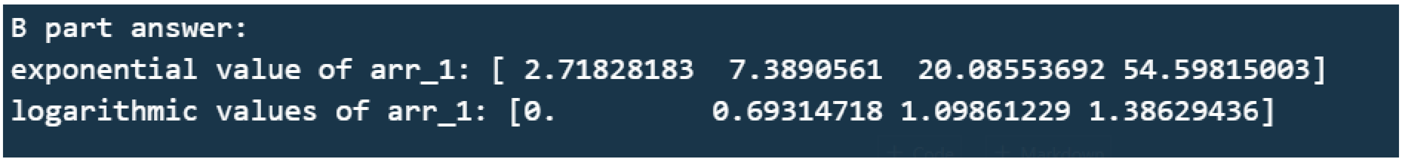
```
    arr_3 = np.random.randint(0, 101, size=(3, 4))
    #Random 2D Array (arr_3)
    print('A part answer: ')
    print(arr_3)
```



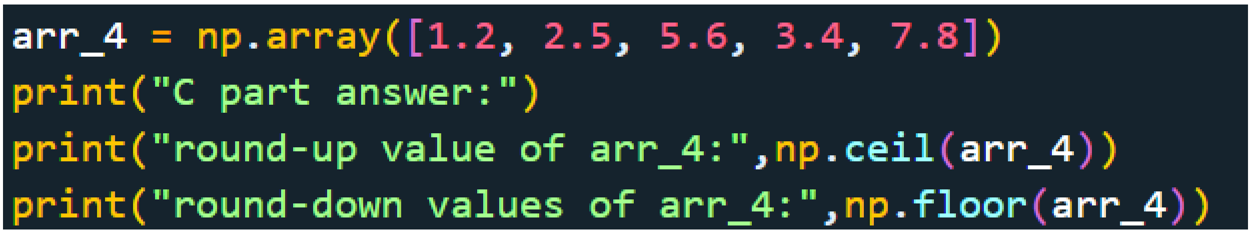
```
A part answer:
[[95  4 75 49]
 [54 22 21 55]
 [37 66 99 22]]
```



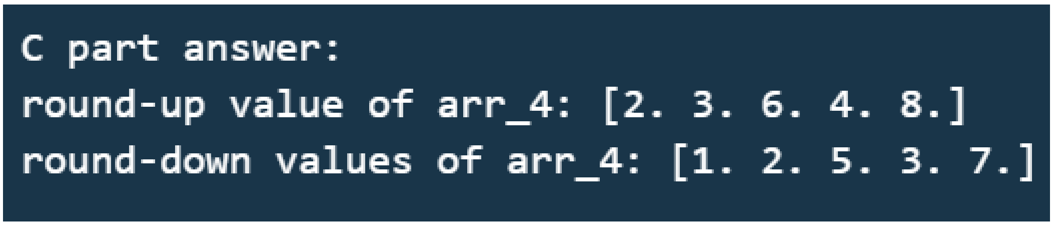
```
    print("B part answer:")
    print("exponential value of arr_1:",np.exp(arr_1))
    print("logarithmic values of arr_1:",np.log(arr_1))
```



```
B part answer:
exponential value of arr_1: [ 2.71828183  7.3890561  20.08553692 54.59815003]
logarithmic values of arr_1: [0.         0.69314718 1.09861229 1.38629436]
```



```
arr_4 = np.array([1.2, 2.5, 5.6, 3.4, 7.8])
print("C part answer:")
print("round-up value of arr_4:",np.ceil(arr_4))
print("round-down values of arr_4:",np.floor(arr_4))
```

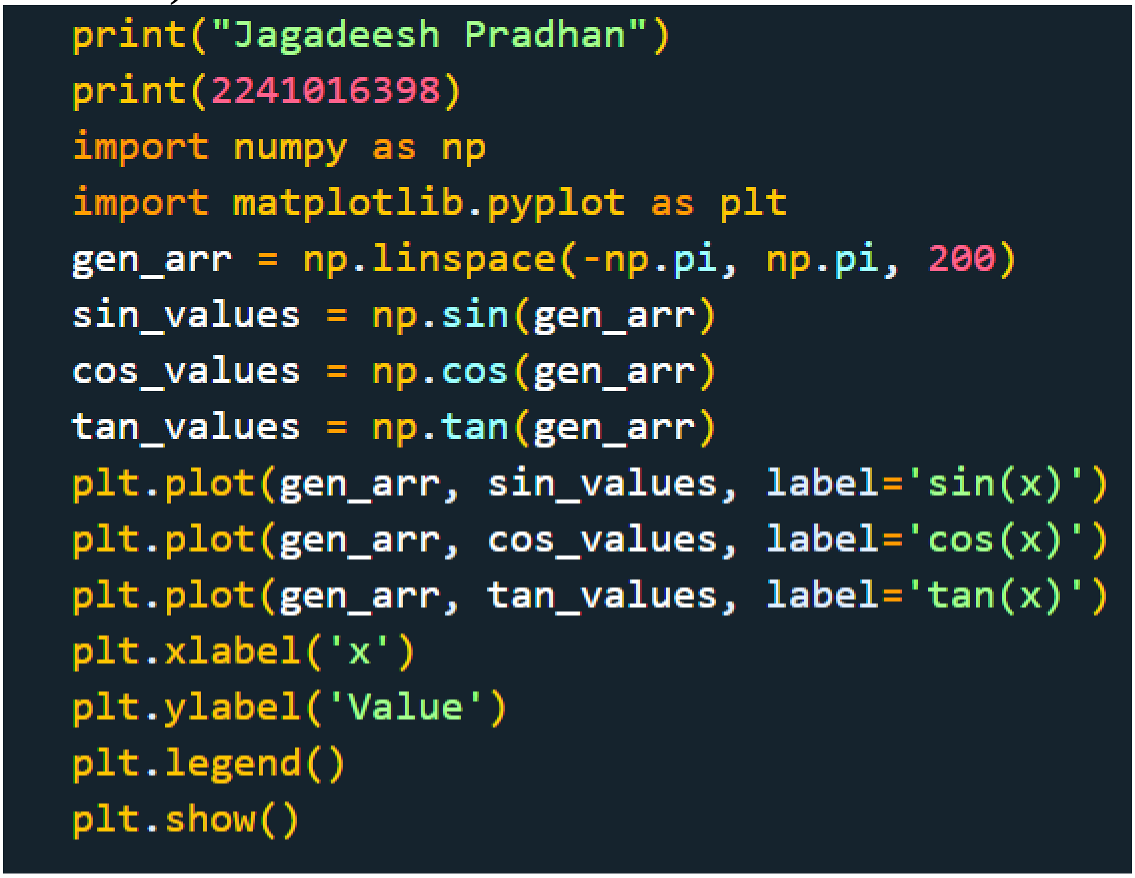


```
C part answer:
round-up value of arr_4: [2. 3. 6. 4. 8.]
round-down values of arr_4: [1. 2. 5. 3. 7.]
```

Q-3) Generate an array of 200 values between -pi & pi. Calculate the corresponding sin, cos and tan value for the generated array. Finally, plot the sin, cos and tan curves using the matplotlib library.

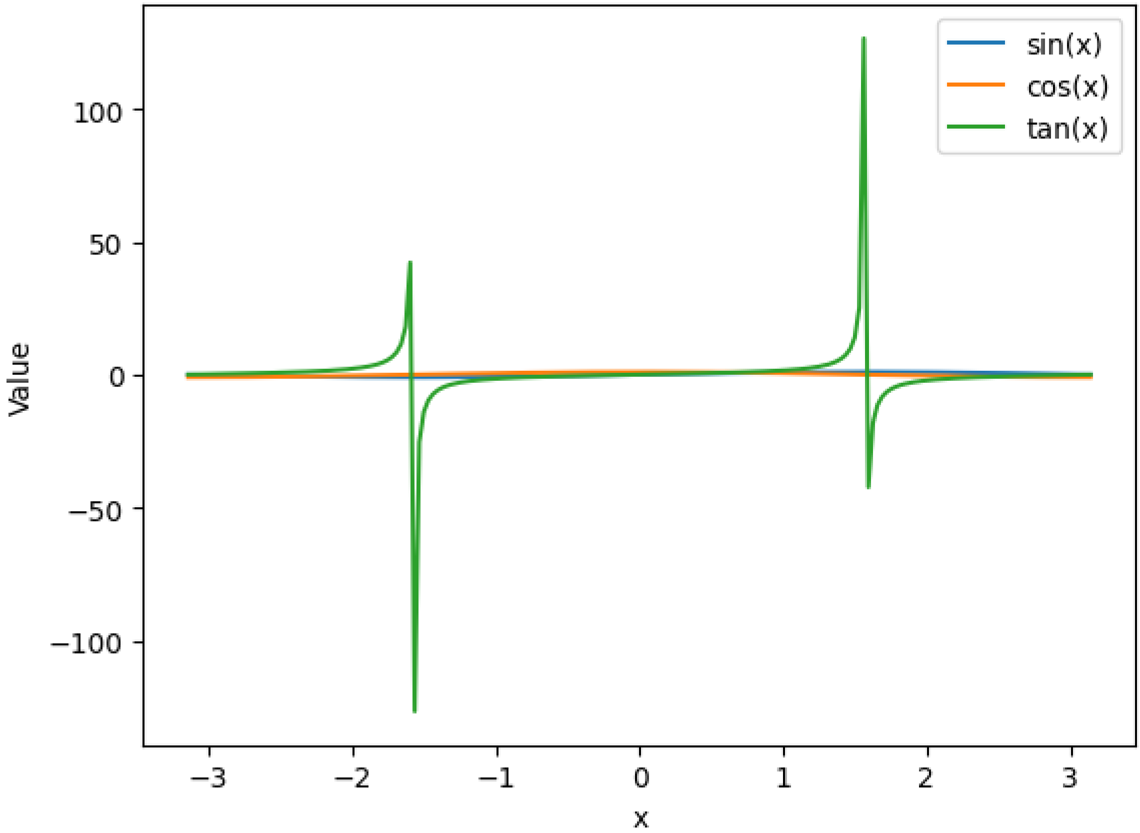
Hints: use np.linspace to generate values (gen_arr), then sin_values = np.sin(gen_arr) and so on for others and finally use plot(gen_arr, sin_values). Follow the same for cos and tan also.

Answer)



```
print("Jagadeesh Pradhan")
print(2241016398)
import numpy as np
import matplotlib.pyplot as plt
gen_arr = np.linspace(-np.pi, np.pi, 200)
sin_values = np.sin(gen_arr)
cos_values = np.cos(gen_arr)
tan_values = np.tan(gen_arr)
plt.plot(gen_arr, sin_values, label='sin(x)')
plt.plot(gen_arr, cos_values, label='cos(x)')
plt.plot(gen_arr, tan_values, label='tan(x)')
plt.xlabel('x')
plt.ylabel('Value')
plt.legend()
plt.show()
```

```
Jagadeesh Pradhan
2241016398
```



Q-4) Given two 2D arrays, arr_1 = np.array([[1, 2], [3, 4]]), arr_2 = np.array([[2, 4], [6, 9]]). Perform the following operations.

A. Matrix multiplication with Dot Product

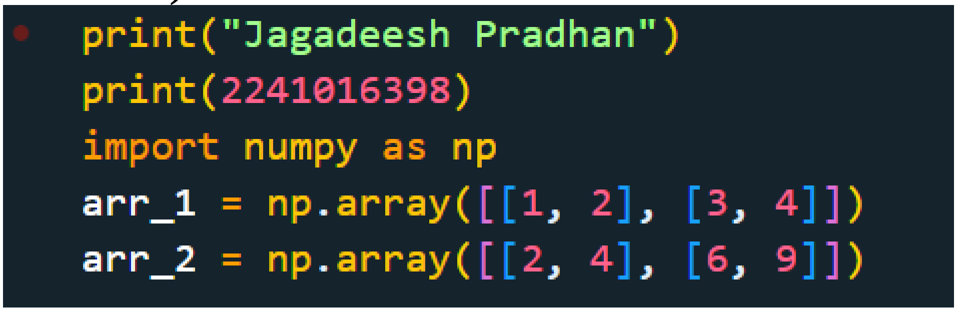
B. Compute eigenvalues and eigenvectors for both matrices.

C. Compute the determinant of both matrices.

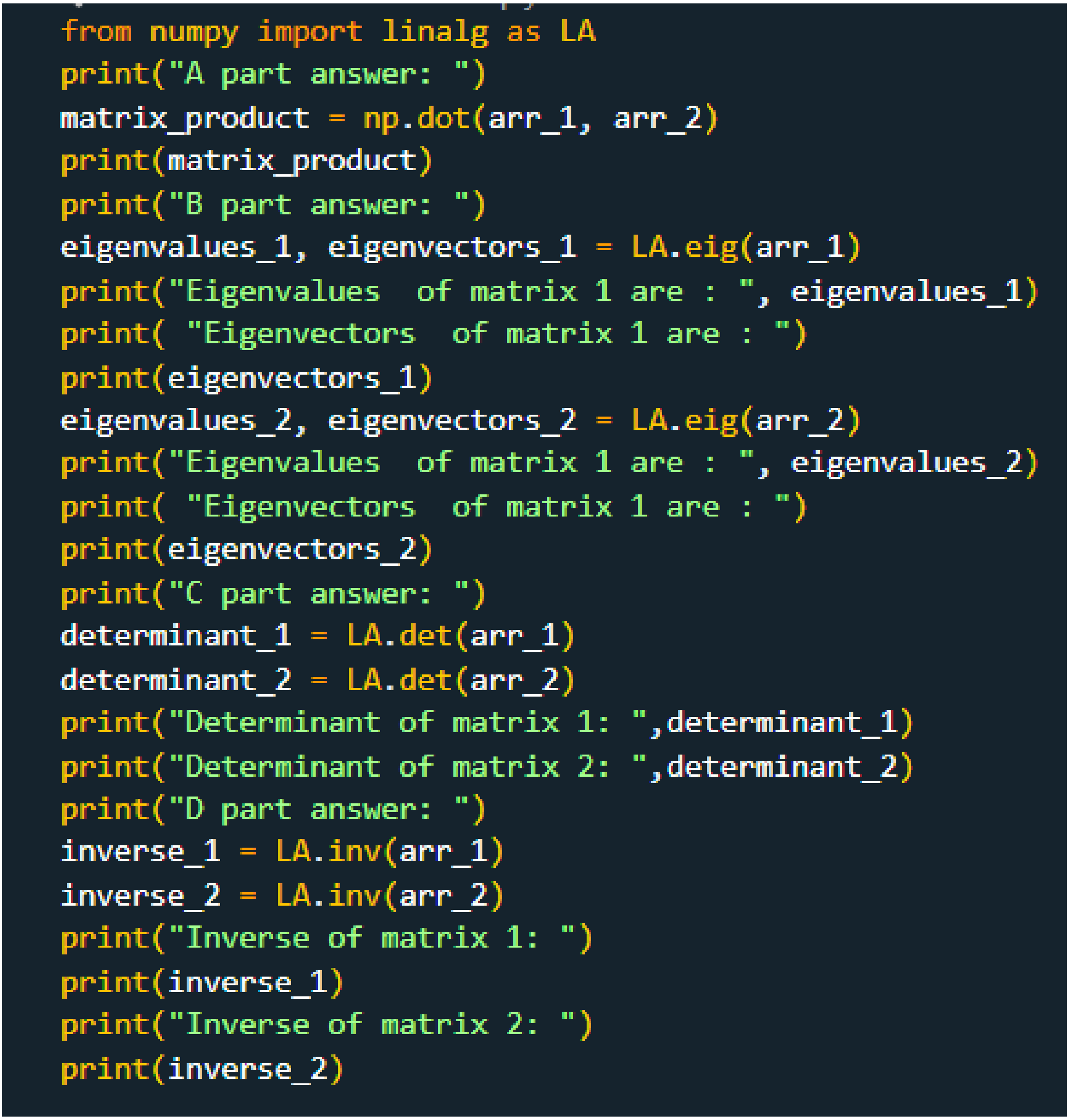
D. Compute the inverse of both the matrices.

Hint: Use from numpy import linalg as LA

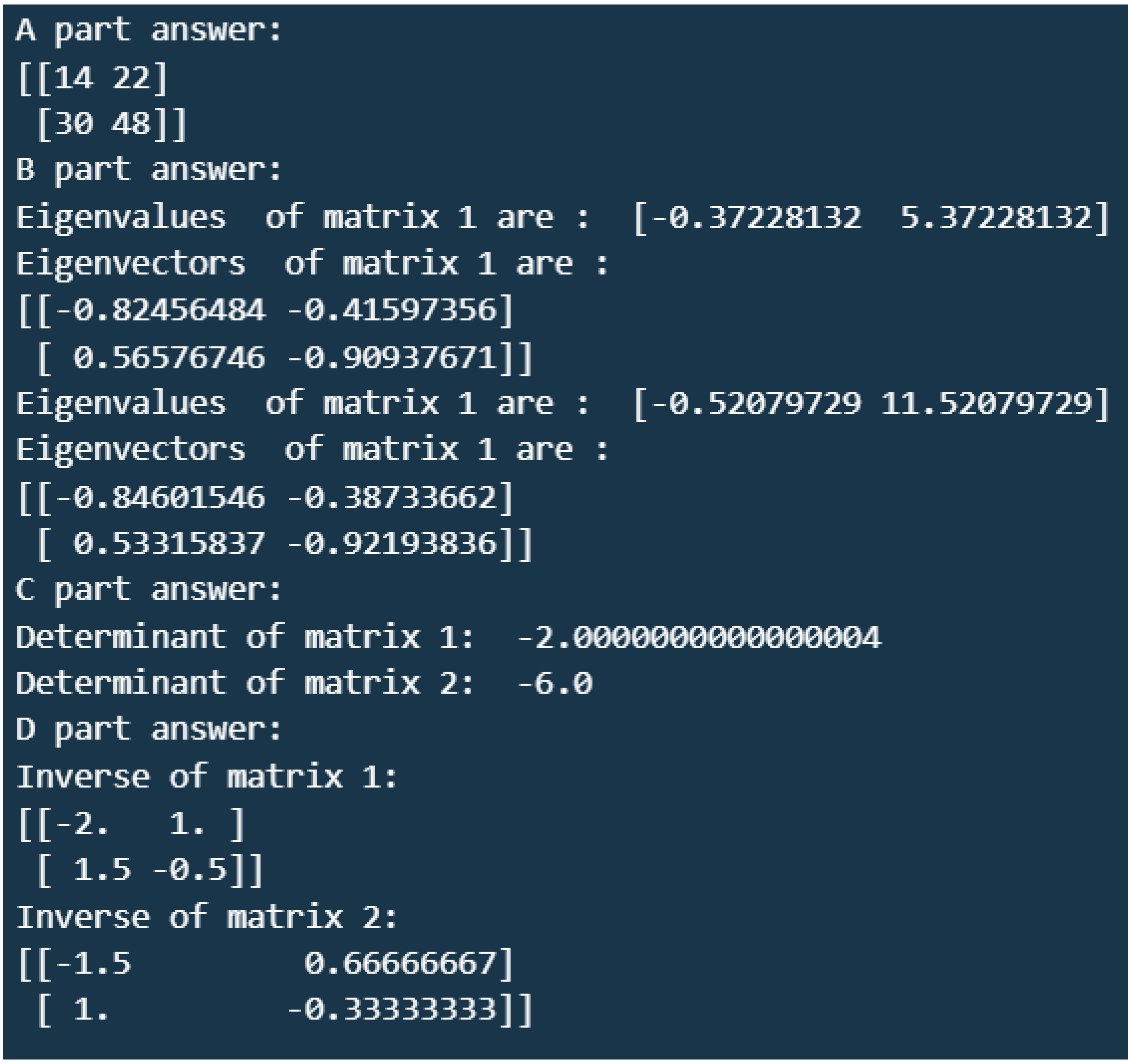
Answer)



```python
print("Jagadeesh Pradhan")
print(2241016398)
import numpy as np
arr_1 = np.array([[1, 2], [3, 4]])
arr_2 = np.array([[2, 4], [6, 9]])
```



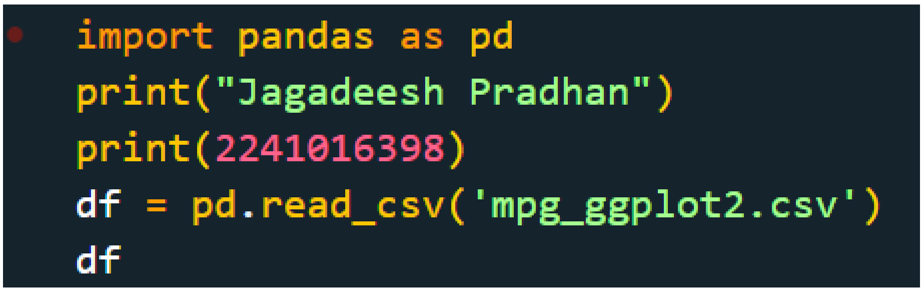
```
from numpy import linalg as LA
print("A part answer: ")
matrix_product = np.dot(arr_1, arr_2)
print(matrix_product)
print("B part answer: ")
eigenvalues_1, eigenvectors_1 = LA.eig(arr_1)
print("Eigenvalues  of matrix 1 are : ", eigenvalues_1)
print( "Eigenvectors  of matrix 1 are : ")
print(eigenvectors_1)
eigenvalues_2, eigenvectors_2 = LA.eig(arr_2)
print("Eigenvalues  of matrix 1 are : ", eigenvalues_2)
print( "Eigenvectors  of matrix 1 are : ")
print(eigenvectors_2)
print("C part answer: ")
determinant_1 = LA.det(arr_1)
determinant_2 = LA.det(arr_2)
print("Determinant of matrix 1: ",determinant_1)
print("Determinant of matrix 2: ",determinant_2)
print("D part answer: ")
inverse_1 = LA.inv(arr_1)
inverse_2 = LA.inv(arr_2)
print("Inverse of matrix 1: ")
print(inverse_1)
print("Inverse of matrix 2: ")
print(inverse_2)
```



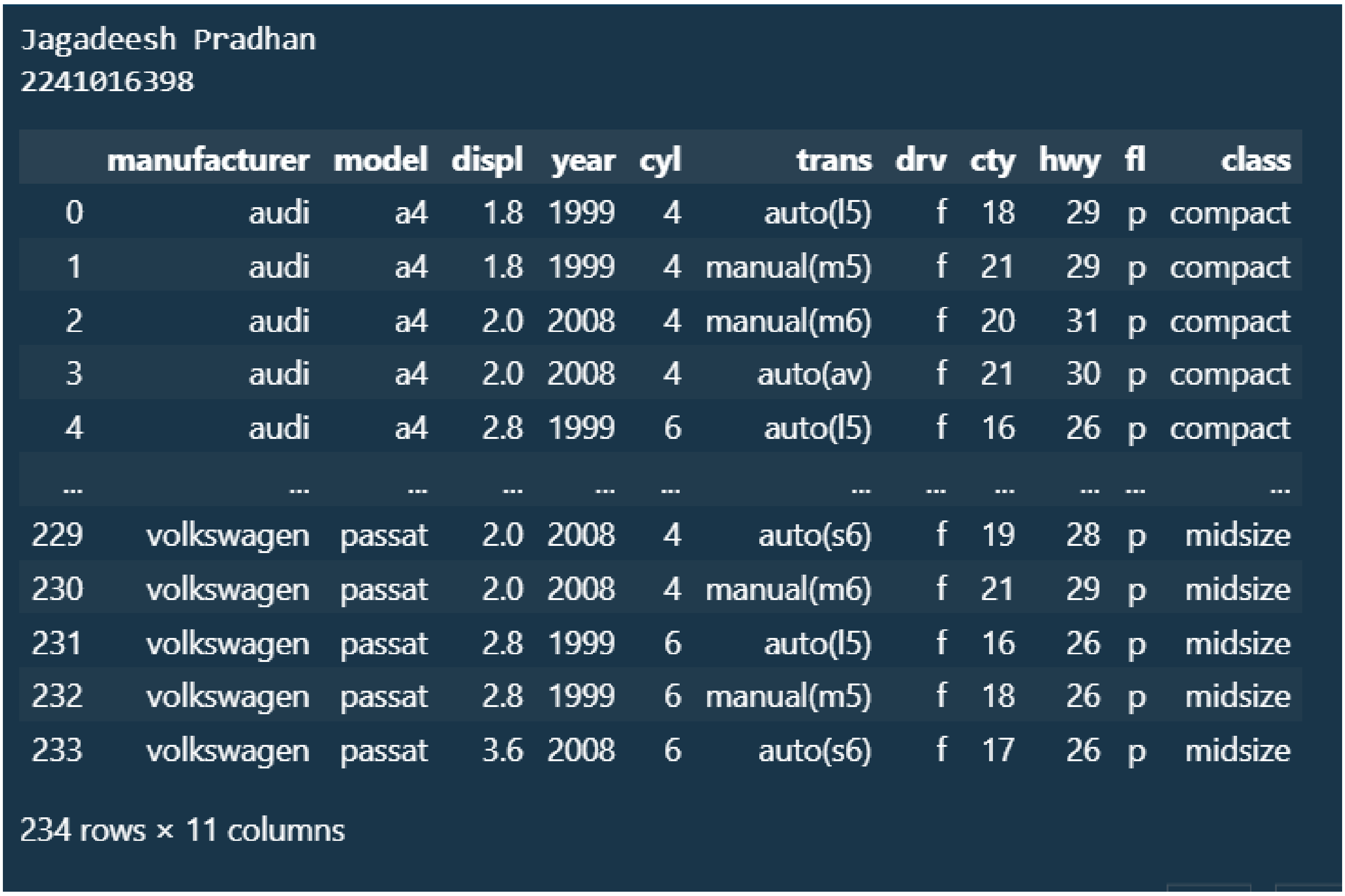
```
A part answer:
[[14 22]
 [30 48]]
B part answer:
Eigenvalues  of matrix 1 are :  [-0.37228132  5.37228132]
Eigenvectors  of matrix 1 are :
[[-0.82456484 -0.41597356]
 [ 0.56576746 -0.90937671]]
Eigenvalues  of matrix 1 are :  [-0.52079729 11.52079729]
Eigenvectors  of matrix 1 are :
[[-0.84601546 -0.38733662]
 [ 0.53315837 -0.92193836]]
C part answer:
Determinant of matrix 1:  -2.0000000000000004
Determinant of matrix 2:  -6.0
D part answer:
Inverse of matrix 1:
[[-2.   1. ]
 [ 1.5 -0.5]]
Inverse of matrix 2:
[[-1.5        0.66666667]
 [ 1.        -0.33333333]]
```

Q-5) Using the given data, use the matplotlib library to draw the Pie chart for the column class.
Dataset: Data

Answer)



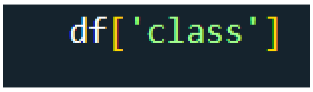
```
import pandas as pd
print("Jagadeesh Pradhan")
print(2241016398)
df = pd.read_csv('mpg_ggplot2.csv')
df
```



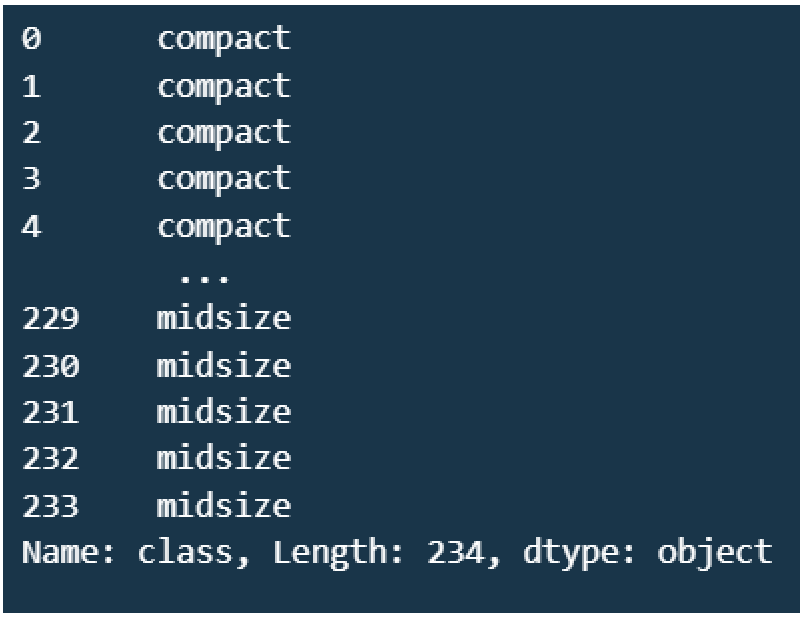
```
Jagadeesh Pradhan
2241016398
```

	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...	...	...	...	...	...	...	...	...	...	...	...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

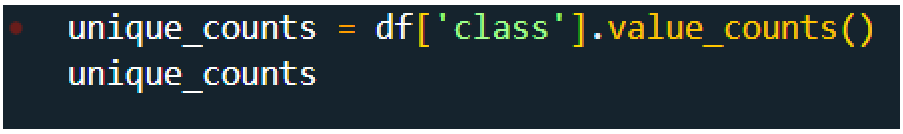
234 rows × 11 columns



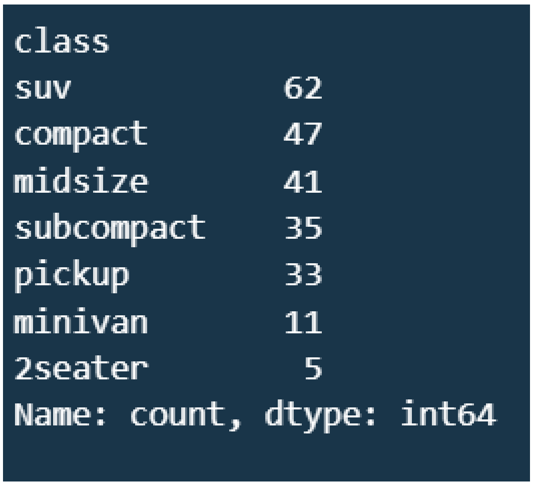
```
df['class']
```



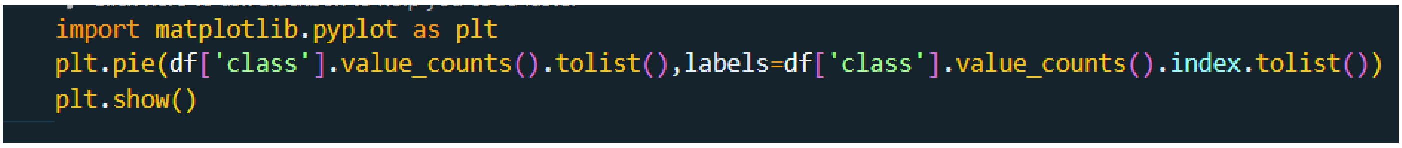
```
0      compact
1      compact
2      compact
3      compact
4      compact
        ...   
229    midsize
230    midsize
231    midsize
232    midsize
233    midsize
Name: class, Length: 234, dtype: object
```



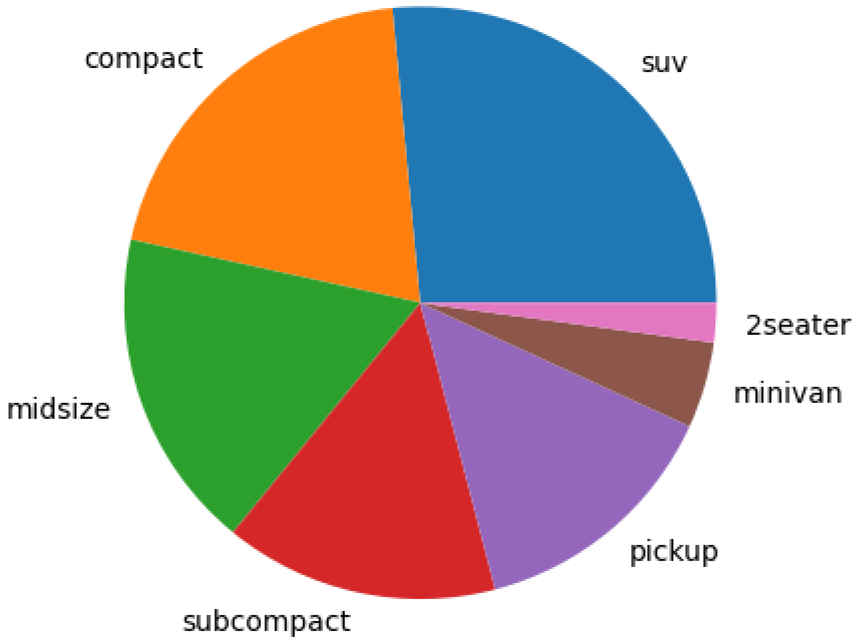
```
unique_counts = df['class'].value_counts()
unique_counts
```



```
class
suv           62
compact       47
midsize       41
subcompact    35
pickup        33
minivan       11
2seater        5
Name: count, dtype: int64
```

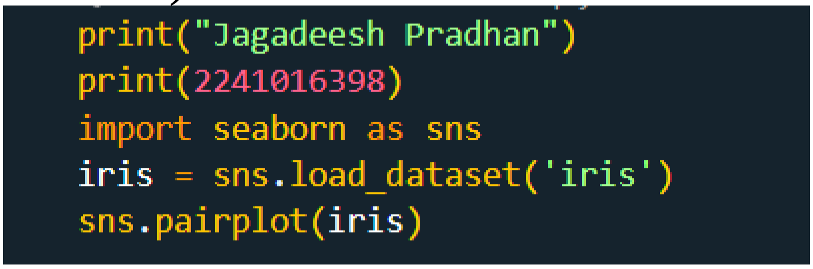


```
import matplotlib.pyplot as plt
plt.pie(df['class'].value_counts().tolist(),labels=df['class'].value_counts().index.tolist())
plt.show()
```

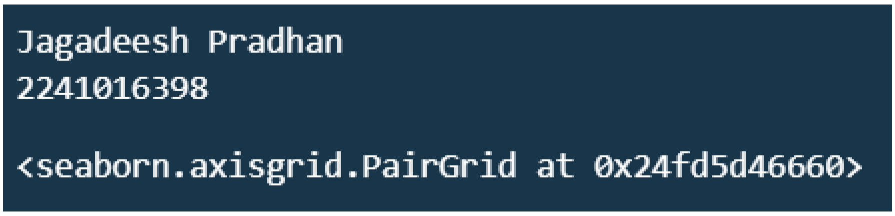


Q-6) A pairwise plot is a favourite in exploratory analysis to understand the relationship between all possible pairs of numeric variables. Use the seaborn library to load the iris dataset and then plot the pairwise plot for the iris data.

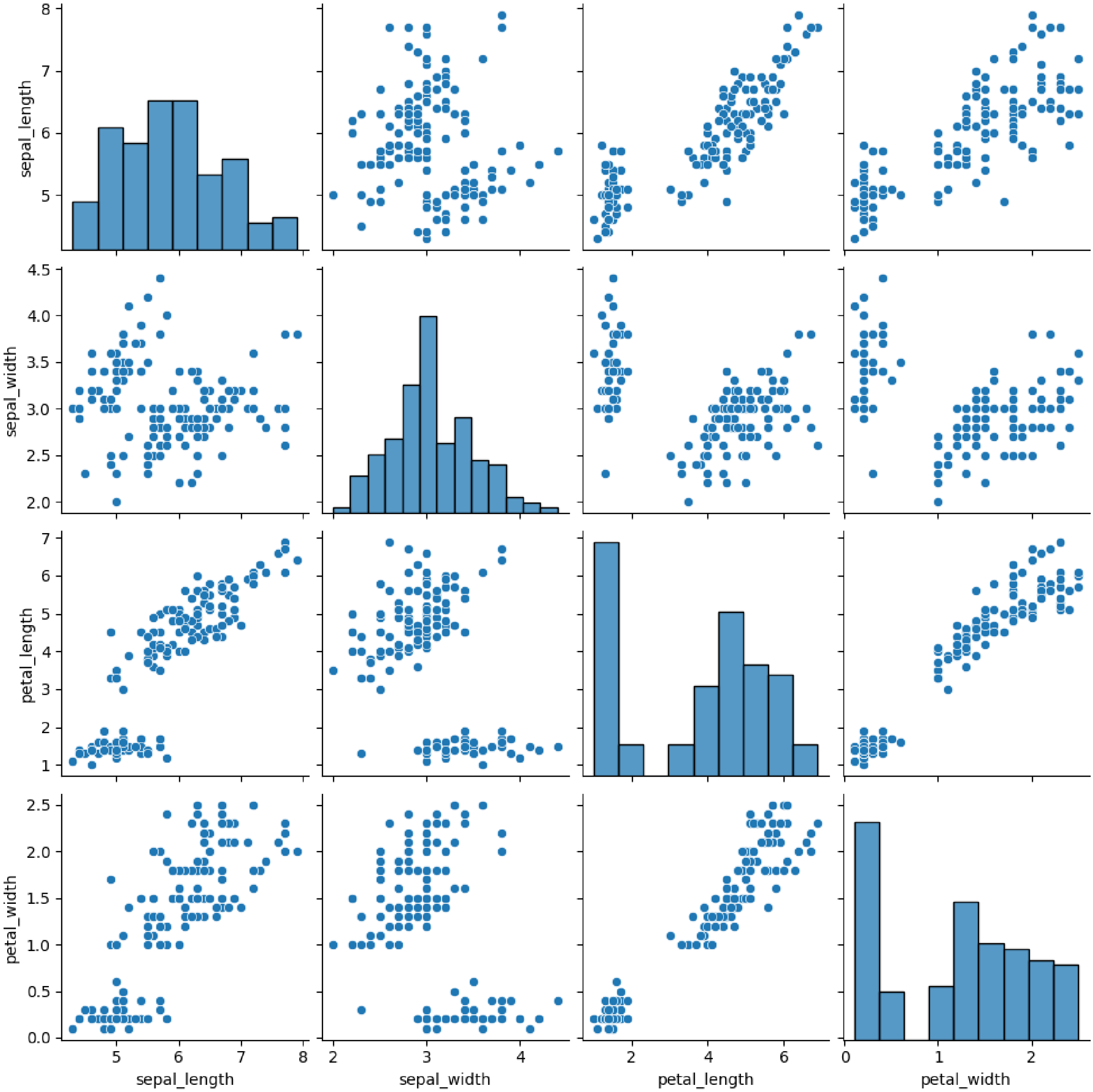
Answer)

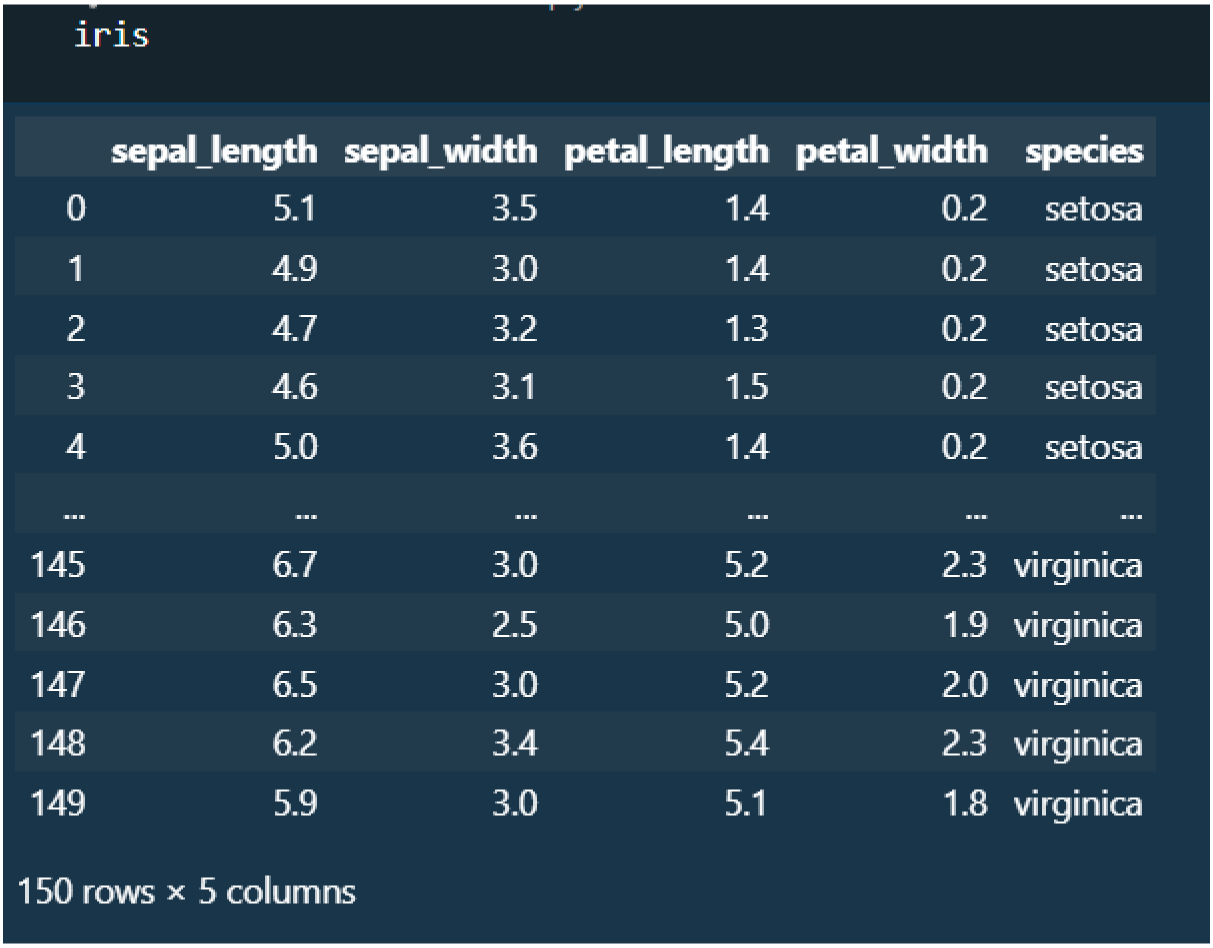


```
print("Jagadeesh Pradhan")
print(2241016398)
import seaborn as sns
iris = sns.load_dataset('iris')
sns.pairplot(iris)
```



```
Jagadeesh Pradhan
2241016398

<seaborn.axisgrid.PairGrid at 0x24fd5d46660>
```





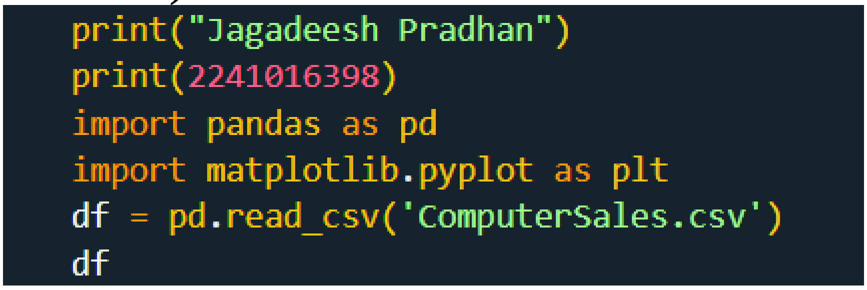
```
iris
```

	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
...	...	...	...	...	...
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

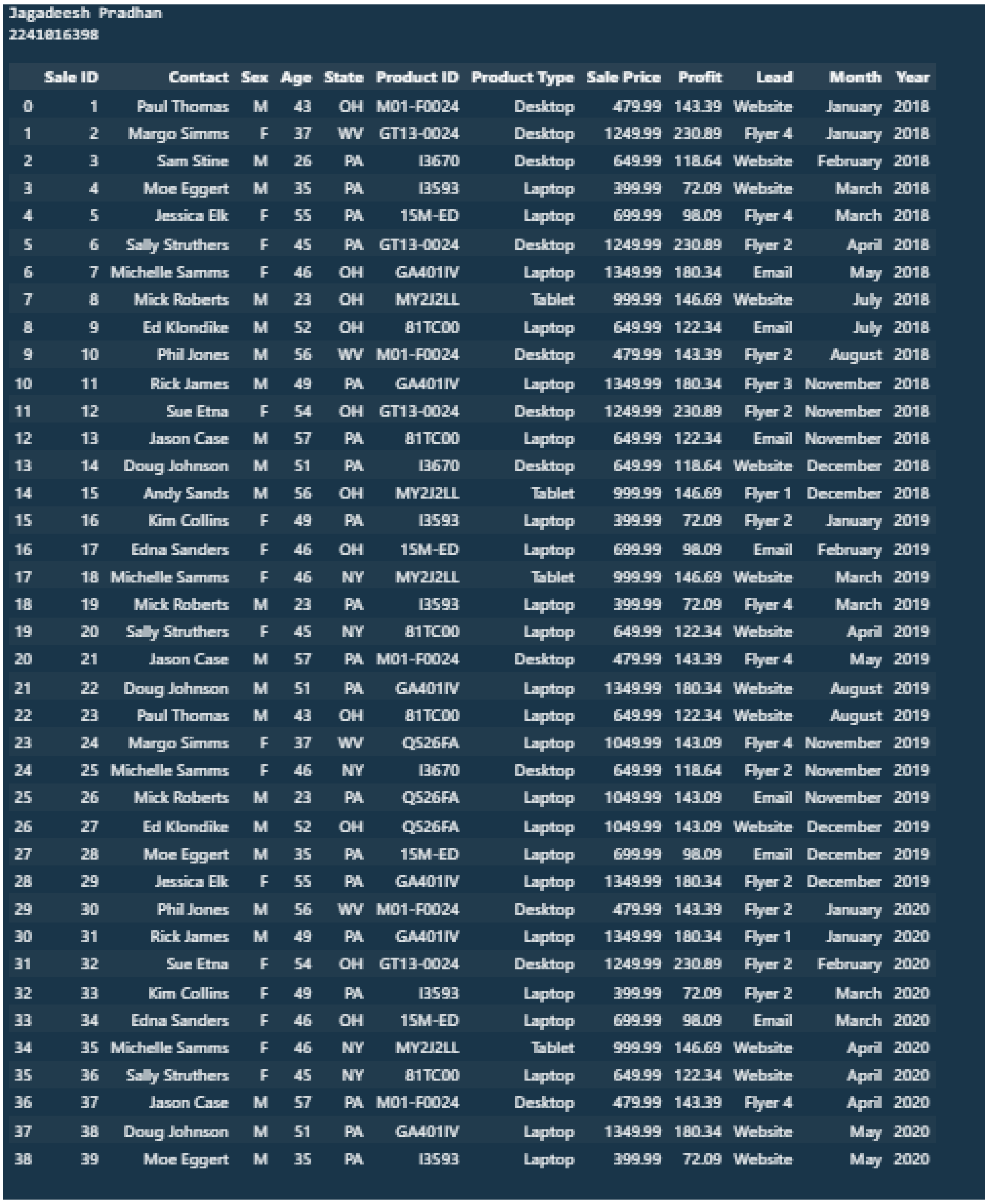
150 rows × 5 columns

Q-7) Download the given dataset and plot the histogram for integer and float columns of the
dataset.
Dataset: Data

Answer)

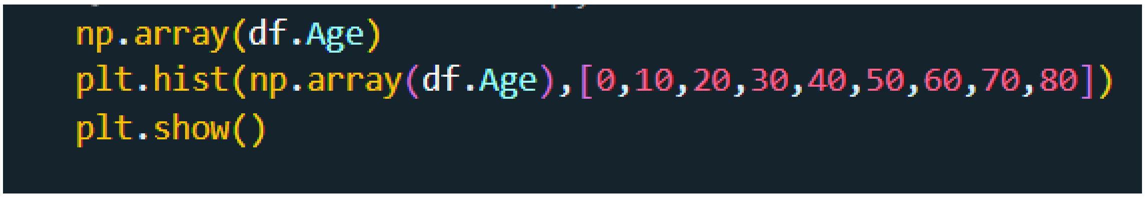


```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('ComputerSales.csv')
df
```

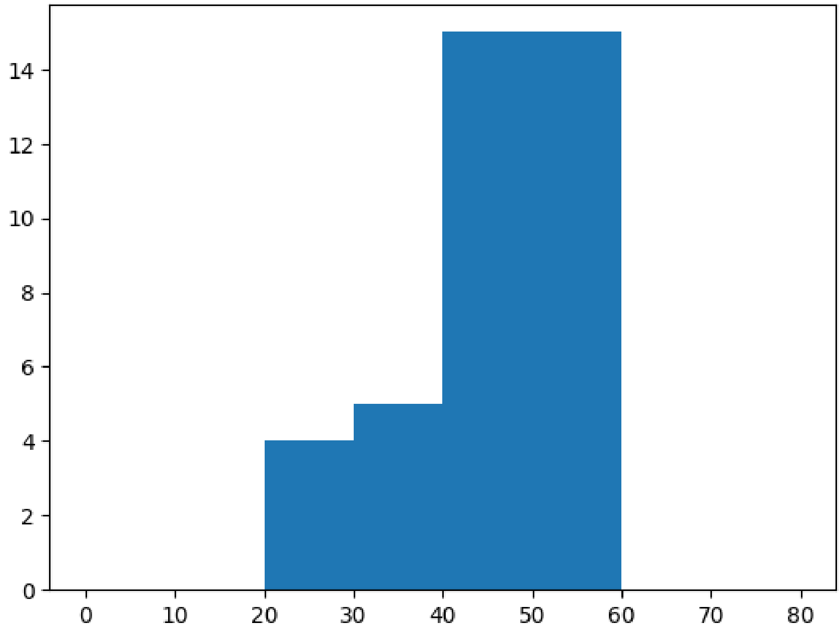


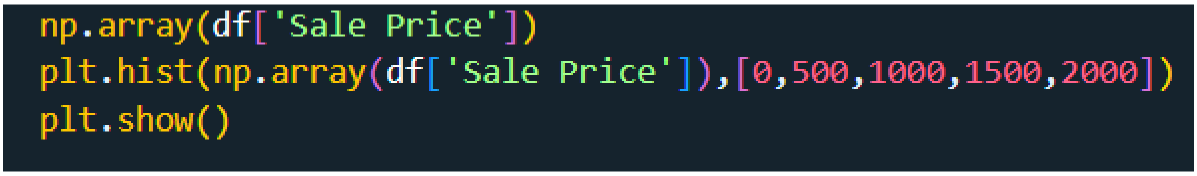
```
Jagadeesh Pradhan
2241816398
```

	Sale ID	Contact	Sex	Age	State	Product ID	Product Type	Sale Price	Profit	Lead	Month	Year
0	1	Paul Thomas	M	43	OH	M01-F0024	Desktop	479.99	143.39	Website	January	2018
1	2	Margo Simms	F	37	WV	GT13-0024	Desktop	1249.99	230.89	Flyer 4	January	2018
2	3	Sam Stine	M	26	PA	I3670	Desktop	649.99	118.64	Website	February	2018
3	4	Moe Eggert	M	35	PA	I3593	Laptop	399.99	72.09	Website	March	2018
4	5	Jessica Elk	F	55	PA	15M-ED	Laptop	699.99	98.09	Flyer 4	March	2018
5	6	Sally Struthers	F	45	PA	GT13-0024	Desktop	1249.99	230.89	Flyer 2	April	2018
6	7	Michelle Samms	F	46	OH	GA401IV	Laptop	1349.99	180.34	Email	May	2018
7	8	Mick Roberts	M	23	OH	MY2J2LL	Tablet	999.99	146.69	Website	July	2018
8	9	Ed Klondike	M	52	OH	81TC00	Laptop	649.99	122.34	Email	July	2018
9	10	Phil Jones	M	56	WV	M01-F0024	Desktop	479.99	143.39	Flyer 2	August	2018
10	11	Rick James	M	49	PA	GA401IV	Laptop	1349.99	180.34	Flyer 3	November	2018
11	12	Sue Etna	F	54	OH	GT13-0024	Desktop	1249.99	230.89	Flyer 2	November	2018
12	13	Jason Case	M	57	PA	81TC00	Laptop	649.99	122.34	Email	November	2018
13	14	Doug Johnson	M	51	PA	I3670	Desktop	649.99	118.64	Website	December	2018
14	15	Andy Sands	M	56	OH	MY2J2LL	Tablet	999.99	146.69	Flyer 1	December	2018
15	16	Kim Collins	F	49	PA	I3593	Laptop	399.99	72.09	Flyer 2	January	2019
16	17	Edna Sanders	F	46	OH	15M-ED	Laptop	699.99	98.09	Email	February	2019
17	18	Michelle Samms	F	46	NY	MY2J2LL	Tablet	999.99	146.69	Website	March	2019
18	19	Mick Roberts	M	23	PA	I3593	Laptop	399.99	72.09	Flyer 4	March	2019
19	20	Sally Struthers	F	45	NY	81TC00	Laptop	649.99	122.34	Website	April	2019
20	21	Jason Case	M	57	PA	M01-F0024	Desktop	479.99	143.39	Flyer 4	May	2019
21	22	Doug Johnson	M	51	PA	GA401IV	Laptop	1349.99	180.34	Website	August	2019
22	23	Paul Thomas	M	43	OH	81TC00	Laptop	649.99	122.34	Website	August	2019
23	24	Margo Simms	F	37	WV	Q526FA	Laptop	1049.99	143.09	Flyer 4	November	2019
24	25	Michelle Samms	F	46	NY	I3670	Desktop	649.99	118.64	Flyer 2	November	2019
25	26	Mick Roberts	M	23	PA	Q526FA	Laptop	1049.99	143.09	Email	November	2019
26	27	Ed Klondike	M	52	OH	Q526FA	Laptop	1049.99	143.09	Website	December	2019
27	28	Moe Eggert	M	35	PA	15M-ED	Laptop	699.99	98.09	Email	December	2019
28	29	Jessica Elk	F	55	PA	GA401IV	Laptop	1349.99	180.34	Flyer 2	December	2019
29	30	Phil Jones	M	56	WV	M01-F0024	Desktop	479.99	143.39	Flyer 2	January	2020
30	31	Rick James	M	49	PA	GA401IV	Laptop	1349.99	180.34	Flyer 1	January	2020
31	32	Sue Etna	F	54	OH	GT13-0024	Desktop	1249.99	230.89	Flyer 2	February	2020
32	33	Kim Collins	F	49	PA	I3593	Laptop	399.99	72.09	Flyer 2	March	2020
33	34	Edna Sanders	F	46	OH	15M-ED	Laptop	699.99	98.09	Email	March	2020
34	35	Michelle Samms	F	46	NY	MY2J2LL	Tablet	999.99	146.69	Website	April	2020
35	36	Sally Struthers	F	45	NY	81TC00	Laptop	649.99	122.34	Website	April	2020
36	37	Jason Case	M	57	PA	M01-F0024	Desktop	479.99	143.39	Flyer 4	April	2020
37	38	Doug Johnson	M	51	PA	GA401IV	Laptop	1349.99	180.34	Website	May	2020
38	39	Moe Eggert	M	35	PA	I3593	Laptop	399.99	72.09	Website	May	2020

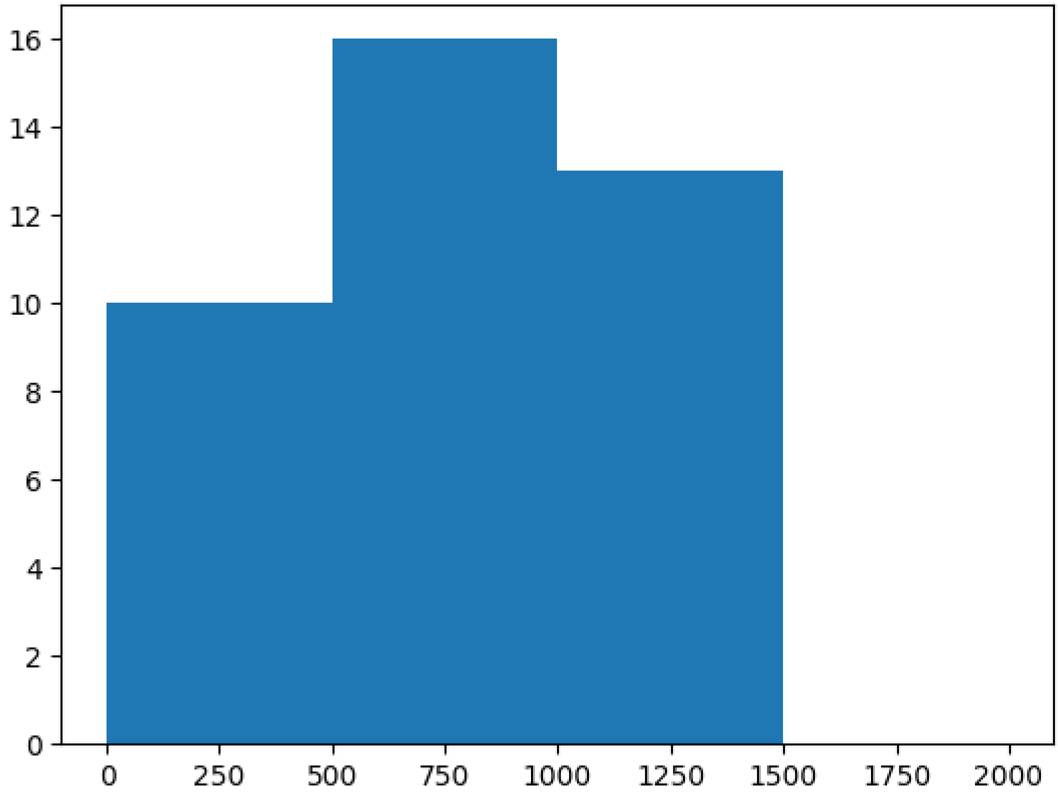


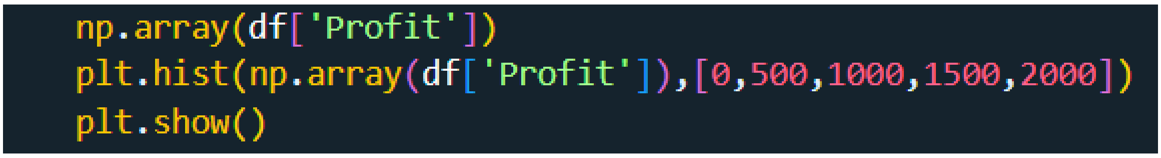
```
    np.array(df.Age)
    plt.hist(np.array(df.Age),[0,10,20,30,40,50,60,70,80])
    plt.show()
```



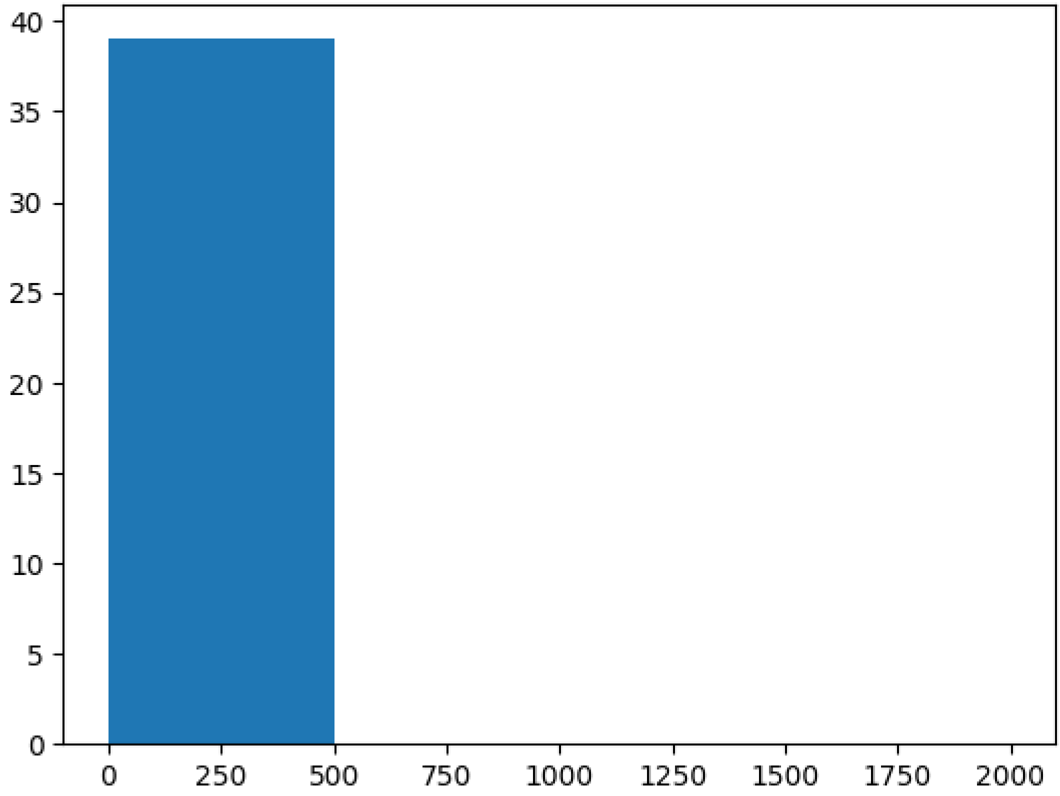


```
np.array(df['Sale Price'])
plt.hist(np.array(df['Sale Price']),[0,500,1000,1500,2000])
plt.show()
```



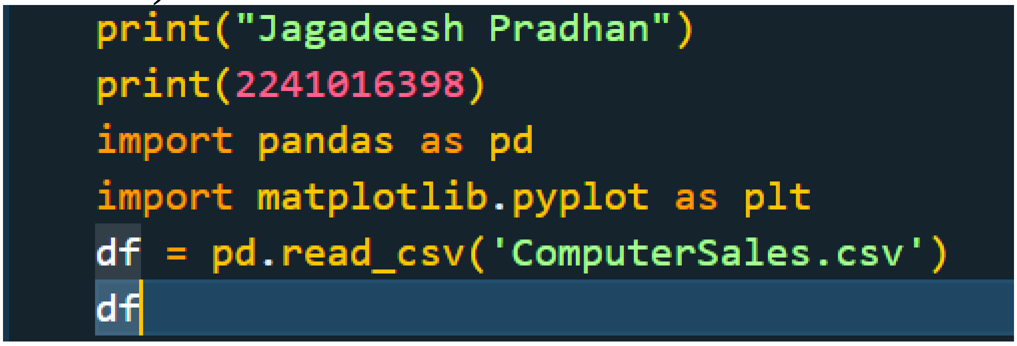


```
np.array(df['Profit'])
plt.hist(np.array(df['Profit']),[0,500,1000,1500,2000])
plt.show()
```

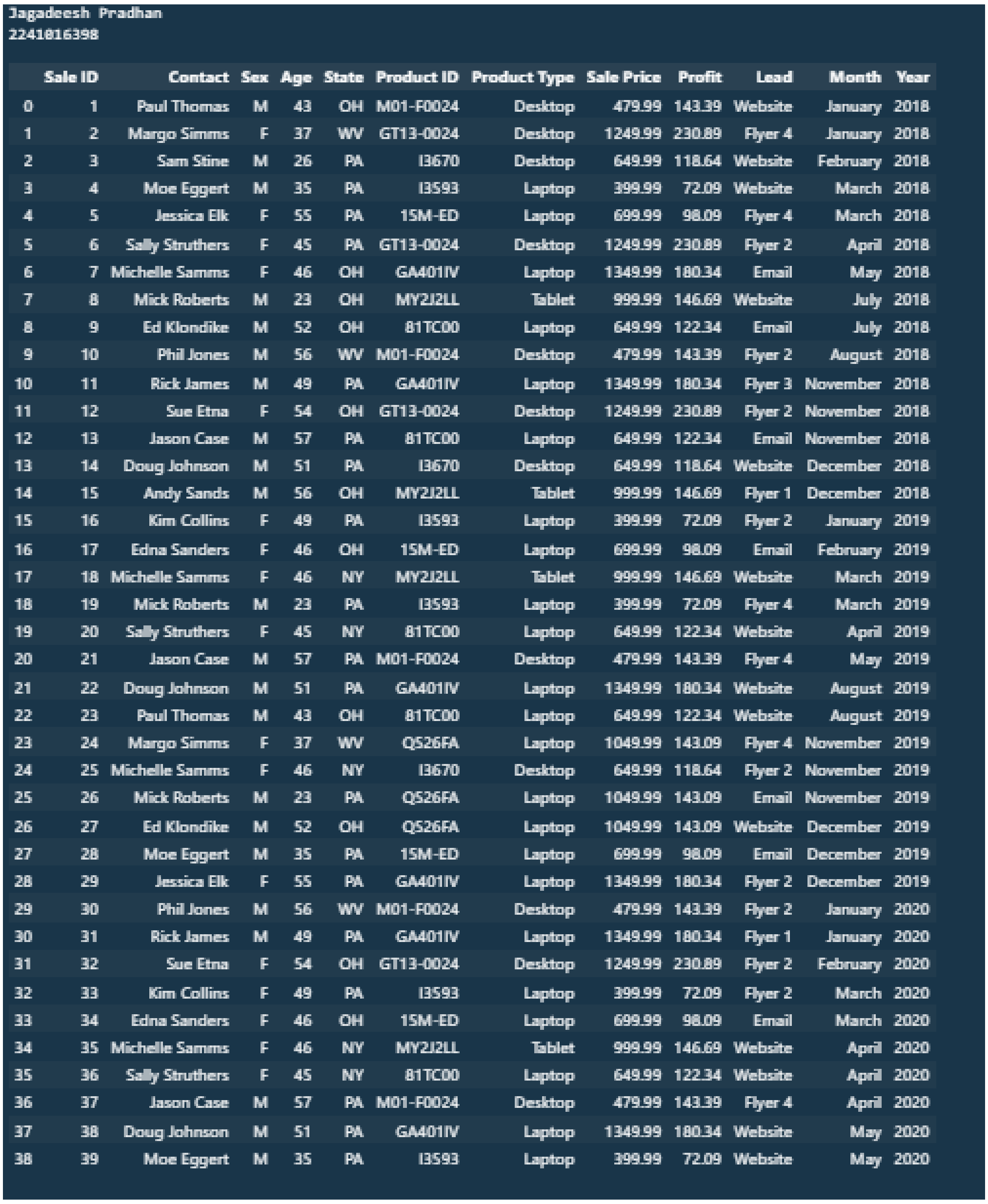


Q-8) Download the dataset and plot the scatter plot for the sale price and profit columns.
Dataset: Data

Answer)

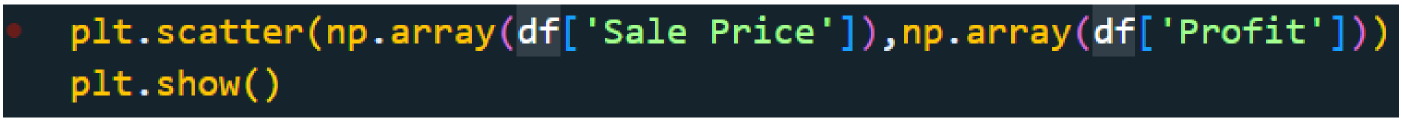


```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('ComputerSales.csv')
df
```

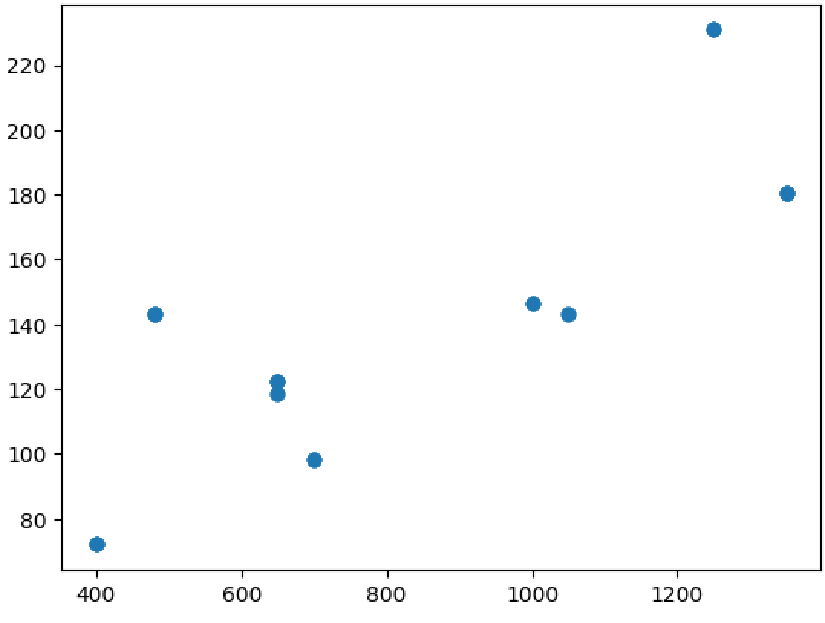


```
Jagadeesh Pradhan
2241016398
```

	Sale ID	Contact	Sex	Age	State	Product ID	Product Type	Sale Price	Profit	Lead	Month	Year
0	1	Paul Thomas	M	43	OH	M01-F0024	Desktop	479.99	143.39	Website	January	2018
1	2	Margo Simms	F	37	WV	GT13-0024	Desktop	1249.99	230.89	Flyer 4	January	2018
2	3	Sam Stine	M	26	PA	I3670	Desktop	649.99	118.64	Website	February	2018
3	4	Moe Eggert	M	35	PA	I3593	Laptop	399.99	72.09	Website	March	2018
4	5	Jessica Elk	F	55	PA	15M-ED	Laptop	699.99	98.09	Flyer 4	March	2018
5	6	Sally Struthers	F	45	PA	GT13-0024	Desktop	1249.99	230.89	Flyer 2	April	2018
6	7	Michelle Samms	F	46	OH	GA401IV	Laptop	1349.99	180.34	Email	May	2018
7	8	Mick Roberts	M	23	OH	MY2J2LL	Tablet	999.99	146.69	Website	July	2018
8	9	Ed Klondike	M	52	OH	81TC00	Laptop	649.99	122.34	Email	July	2018
9	10	Phil Jones	M	56	WV	M01-F0024	Desktop	479.99	143.39	Flyer 2	August	2018
10	11	Rick James	M	49	PA	GA401IV	Laptop	1349.99	180.34	Flyer 3	November	2018
11	12	Sue Etna	F	54	OH	GT13-0024	Desktop	1249.99	230.89	Flyer 2	November	2018
12	13	Jason Case	M	57	PA	81TC00	Laptop	649.99	122.34	Email	November	2018
13	14	Doug Johnson	M	51	PA	I3670	Desktop	649.99	118.64	Website	December	2018
14	15	Andy Sands	M	56	OH	MY2J2LL	Tablet	999.99	146.69	Flyer 1	December	2018
15	16	Kim Collins	F	49	PA	I3593	Laptop	399.99	72.09	Flyer 2	January	2019
16	17	Edna Sanders	F	46	OH	15M-ED	Laptop	699.99	98.09	Email	February	2019
17	18	Michelle Samms	F	46	NY	MY2J2LL	Tablet	999.99	146.69	Website	March	2019
18	19	Mick Roberts	M	23	PA	I3593	Laptop	399.99	72.09	Flyer 4	March	2019
19	20	Sally Struthers	F	45	NY	81TC00	Laptop	649.99	122.34	Website	April	2019
20	21	Jason Case	M	57	PA	M01-F0024	Desktop	479.99	143.39	Flyer 4	May	2019
21	22	Doug Johnson	M	51	PA	GA401IV	Laptop	1349.99	180.34	Website	August	2019
22	23	Paul Thomas	M	43	OH	81TC00	Laptop	649.99	122.34	Website	August	2019
23	24	Margo Simms	F	37	WV	Q526FA	Laptop	1049.99	143.09	Flyer 4	November	2019
24	25	Michelle Samms	F	46	NY	I3670	Desktop	649.99	118.64	Flyer 2	November	2019
25	26	Mick Roberts	M	23	PA	Q526FA	Laptop	1049.99	143.09	Email	November	2019
26	27	Ed Klondike	M	52	OH	Q526FA	Laptop	1049.99	143.09	Website	December	2019
27	28	Moe Eggert	M	35	PA	15M-ED	Laptop	699.99	98.09	Email	December	2019
28	29	Jessica Elk	F	55	PA	GA401IV	Laptop	1349.99	180.34	Flyer 2	December	2019
29	30	Phil Jones	M	56	WV	M01-F0024	Desktop	479.99	143.39	Flyer 2	January	2020
30	31	Rick James	M	49	PA	GA401IV	Laptop	1349.99	180.34	Flyer 1	January	2020
31	32	Sue Etna	F	54	OH	GT13-0024	Desktop	1249.99	230.89	Flyer 2	February	2020
32	33	Kim Collins	F	49	PA	I3593	Laptop	399.99	72.09	Flyer 2	March	2020
33	34	Edna Sanders	F	46	OH	15M-ED	Laptop	699.99	98.09	Email	March	2020
34	35	Michelle Samms	F	46	NY	MY2J2LL	Tablet	999.99	146.69	Website	April	2020
35	36	Sally Struthers	F	45	NY	81TC00	Laptop	649.99	122.34	Website	April	2020
36	37	Jason Case	M	57	PA	M01-F0024	Desktop	479.99	143.39	Flyer 4	April	2020
37	38	Doug Johnson	M	51	PA	GA401IV	Laptop	1349.99	180.34	Website	May	2020
38	39	Moe Eggert	M	35	PA	I3593	Laptop	399.99	72.09	Website	May	2020



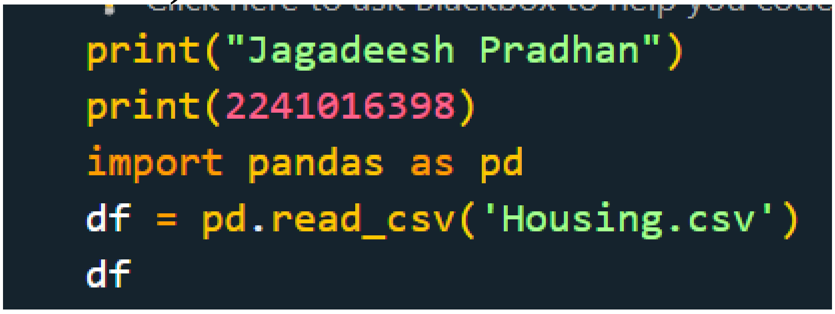
```
plt.scatter(np.array(df['Sale Price']),np.array(df['Profit']))
plt.show()
```



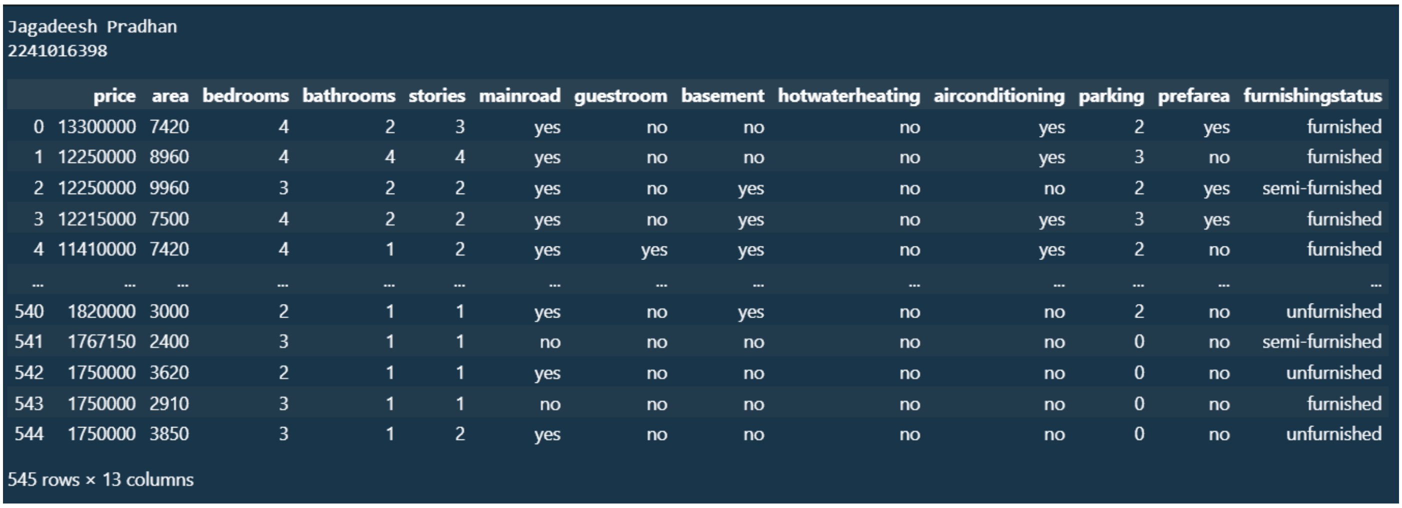
Q

Q-9) Download the given dataset and split the dataset into 70% for training and 30% for testing using the Scikit library.
Dataset: Data

Answer)



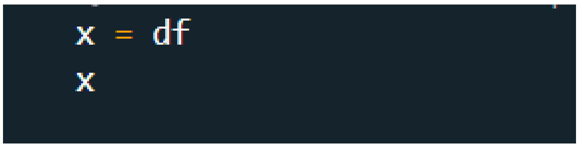
```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
df = pd.read_csv('Housing.csv')
df
```



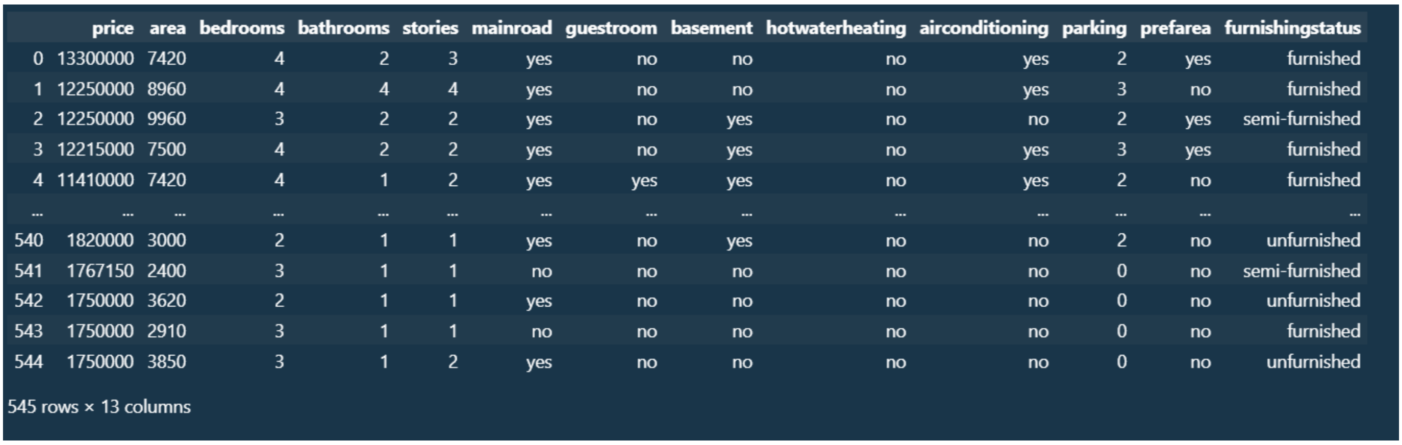
```
Jagadeesh Pradhan
2241016398
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no	unfurnished
541	1767150	2400	3	1	1	no	no	no	no	no	0	no	semi-furnished
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no	unfurnished
543	1750000	2910	3	1	1	no	no	no	no	no	0	no	furnished
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no	unfurnished

545 rows × 13 columns

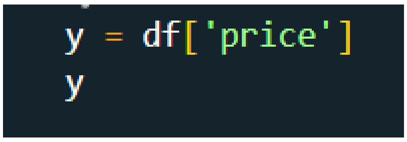


```
x = df
x
```

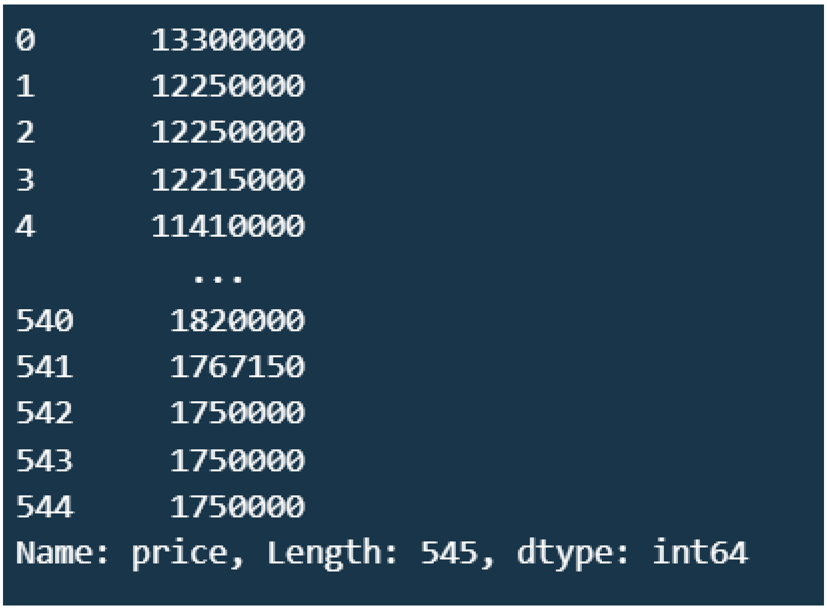


	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no	unfurnished
541	1767150	2400	3	1	1	no	no	no	no	no	0	no	semi-furnished
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no	unfurnished
543	1750000	2910	3	1	1	no	no	no	no	no	0	no	furnished
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no	unfurnished

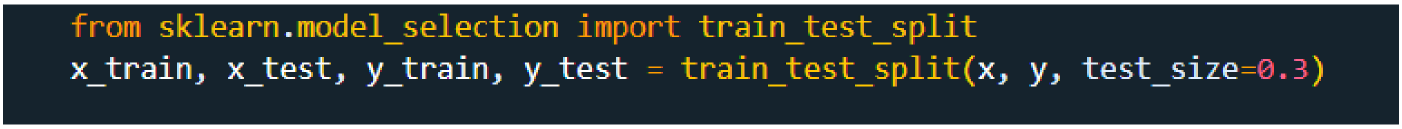
545 rows × 13 columns



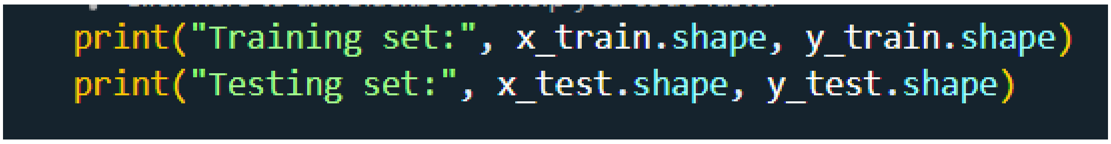
```
y = df['price']
y
```



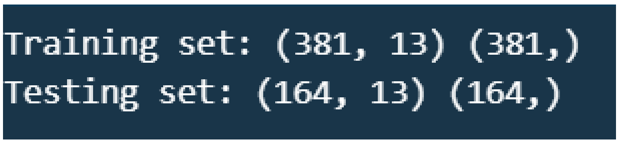
```
0      13300000
1      12250000
2      12250000
3      12215000
4      11410000
         ...
540     1820000
541     1767150
542     1750000
543     1750000
544     1750000
Name: price, Length: 545, dtype: int64
```



```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3)
```



```
print("Training set:", x_train.shape, y_train.shape)
print("Testing set:", x_test.shape, y_test.shape)
```

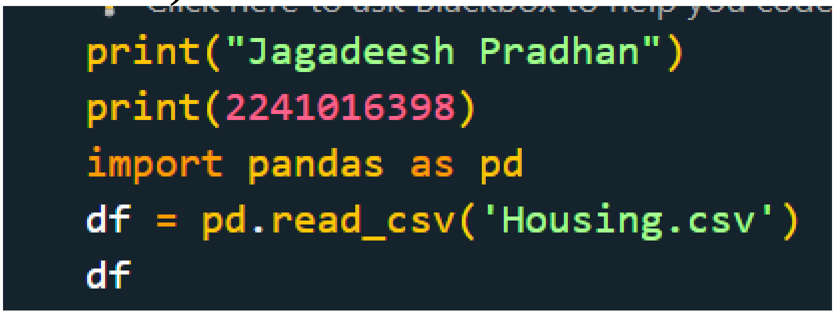


```
Training set: (381, 13) (381,)
Testing set: (164, 13) (164,)
```

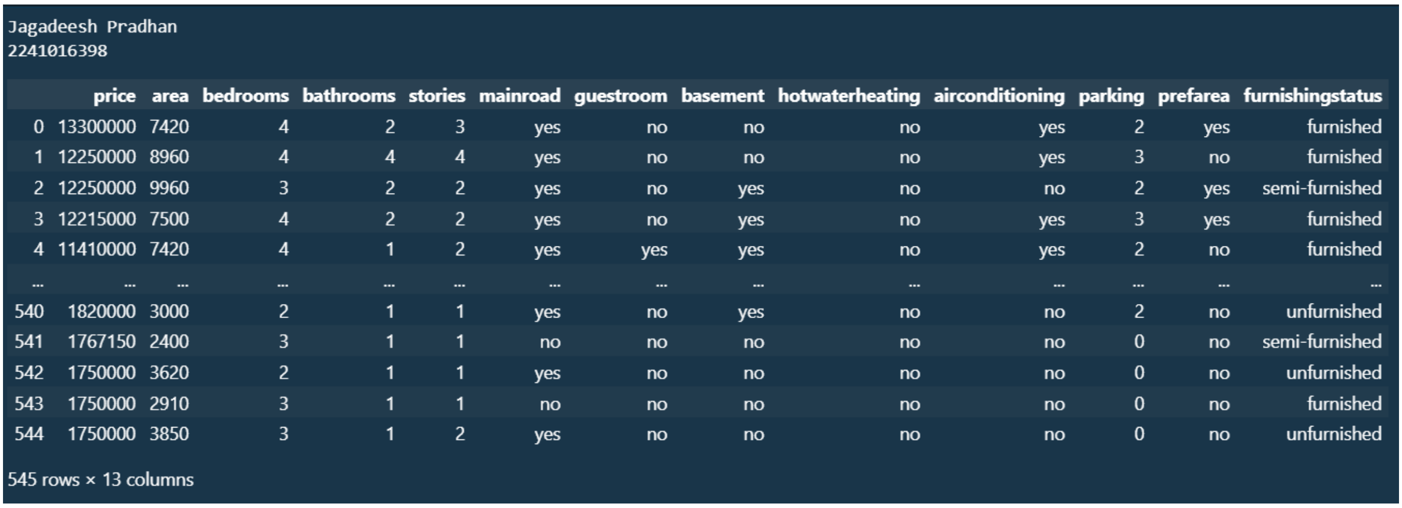
Q-10) Download the dataset given in the link, convert the categorical data into integer columns, and print the shape and head of the dataset.

Dataset: Data

Answer)



```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
df = pd.read_csv('Housing.csv')
df
```



Jagadeesh Pradhan
2241016398

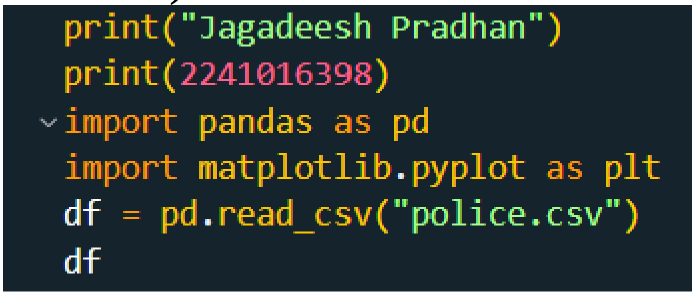
	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes	furnished
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no	furnished
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes	semi-furnished
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes	furnished
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no	furnished
...	...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no	unfurnished
541	1767150	2400	3	1	1	no	no	no	no	no	0	no	semi-furnished
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no	unfurnished
543	1750000	2910	3	1	1	no	no	no	no	no	0	no	furnished
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no	unfurnished

545 rows × 13 columns

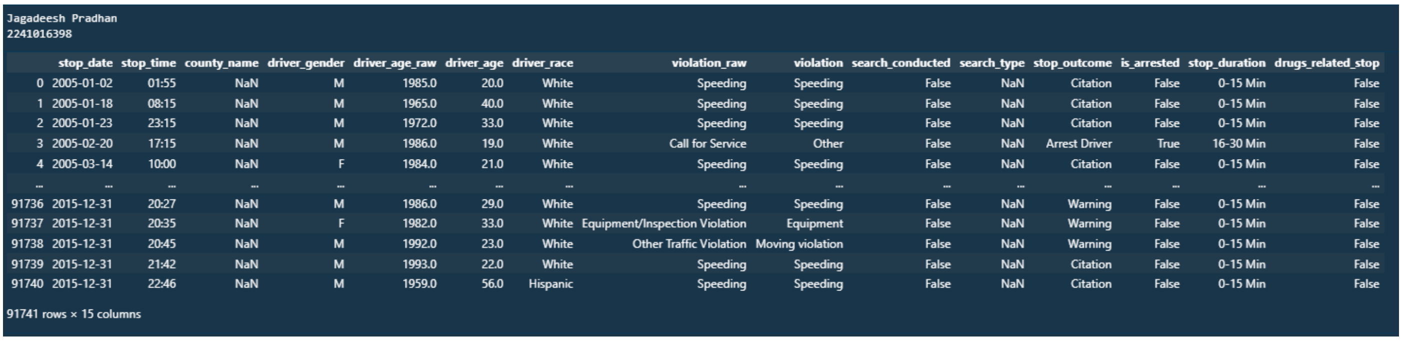
HOME ASSIGNMENT

Q-1) Read the dataset given in the following link. Print the info on the data and clean the data
if required. Visualize the data using an appropriate diagram based on your observations.
Dataset: Data

Answer)



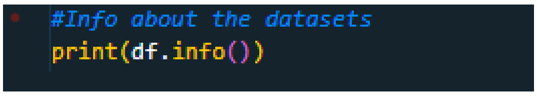
```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("police.csv")
df
```



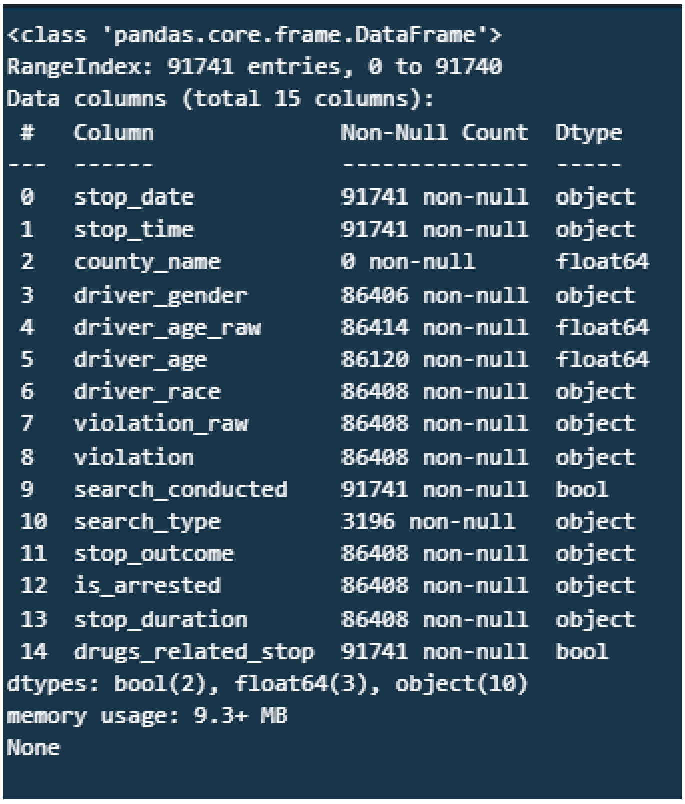
Jagadeesh Pradhan
2241016398

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted	search_type	stop_outcome	is_arrested	stop_duration	drugs_related_stop
0	2005-01-02	01:55	NaN	M	1985.0	20.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
1	2005-01-18	08:15	NaN	M	1965.0	40.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
2	2005-01-23	23:15	NaN	M	1972.0	33.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
3	2005-02-20	17:15	NaN	M	1986.0	19.0	White	Call for Service	Other	False	NaN	Arrest Driver	True	16-30 Min	False
4	2005-03-14	10:00	NaN	F	1984.0	21.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
91736	2015-12-31	20:27	NaN	M	1986.0	29.0	White	Speeding	Speeding	False	NaN	Warning	False	0-15 Min	False
91737	2015-12-31	20:35	NaN	F	1982.0	33.0	White	Equipment/Inspection Violation	Equipment	False	NaN	Warning	False	0-15 Min	False
91738	2015-12-31	20:45	NaN	M	1992.0	23.0	White	Other Traffic Violation	Moving violation	False	NaN	Warning	False	0-15 Min	False
91739	2015-12-31	21:42	NaN	M	1993.0	22.0	White	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False
91740	2015-12-31	22:46	NaN	M	1959.0	56.0	Hispanic	Speeding	Speeding	False	NaN	Citation	False	0-15 Min	False

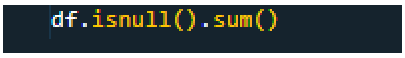
91741 rows × 15 columns



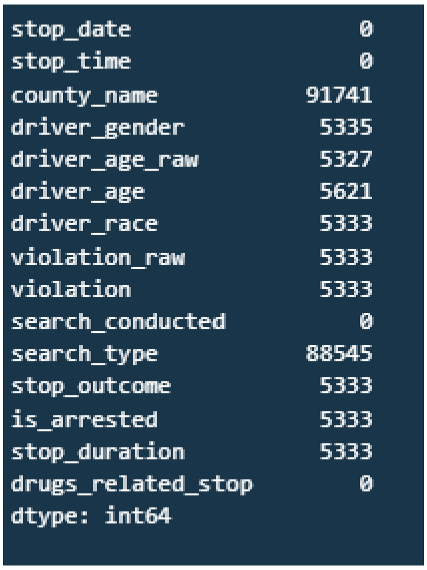
```
#Info about the datasets
print(df.info())
```



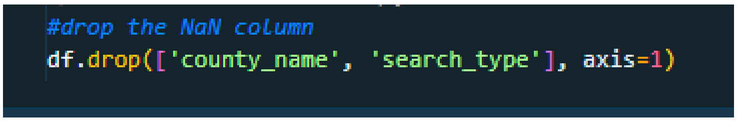
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 91741 entries, 0 to 91740
Data columns (total 15 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   stop_date           91741 non-null  object 
 1   stop_time           91741 non-null  object 
 2   county_name         0 non-null      float64
 3   driver_gender       86406 non-null  object 
 4   driver_age_raw      86414 non-null  float64
 5   driver_age          86120 non-null  float64
 6   driver_race         86408 non-null  object 
 7   violation_raw       86408 non-null  object 
 8   violation           86408 non-null  object 
 9   search_conducted    91741 non-null  bool   
 10  search_type         3196 non-null   object 
 11  stop_outcome        86408 non-null  object 
 12  is_arrested         86408 non-null  object 
 13  stop_duration       86408 non-null  object 
 14  drugs_related_stop  91741 non-null  bool   
dtypes: bool(2), float64(3), object(10)
memory usage: 9.3+ MB
None
```



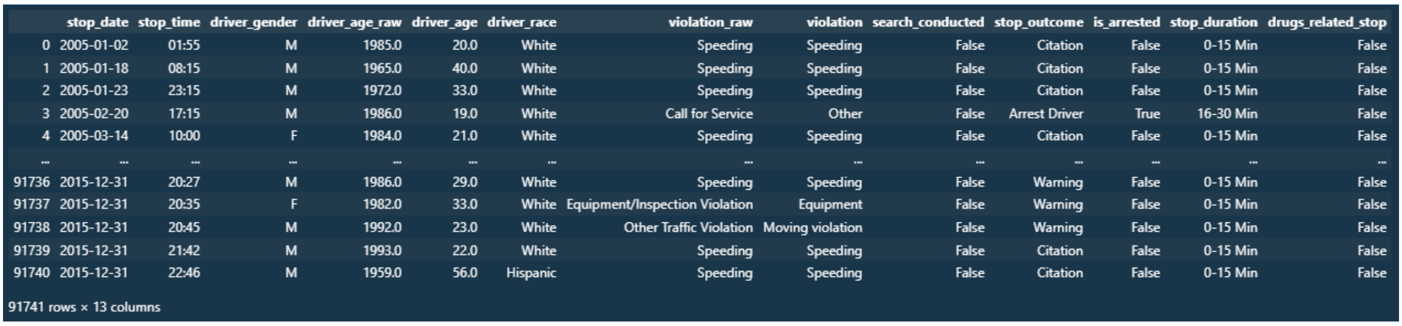
```
df.isnull().sum()
```



```
stop_date                 0
stop_time                 0
county_name           91741
driver_gender          5335
driver_age_raw         5327
driver_age             5621
driver_race            5333
violation_raw          5333
violation              5333
search_conducted          0
search_type           88545
stop_outcome           5333
is_arrested            5333
stop_duration          5333
drugs_related_stop        0
dtype: int64
```

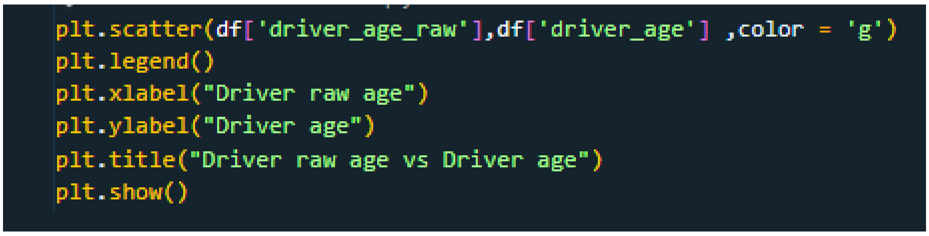


```
#drop the NaN column
df.drop(['county_name', 'search_type'], axis=1)
```

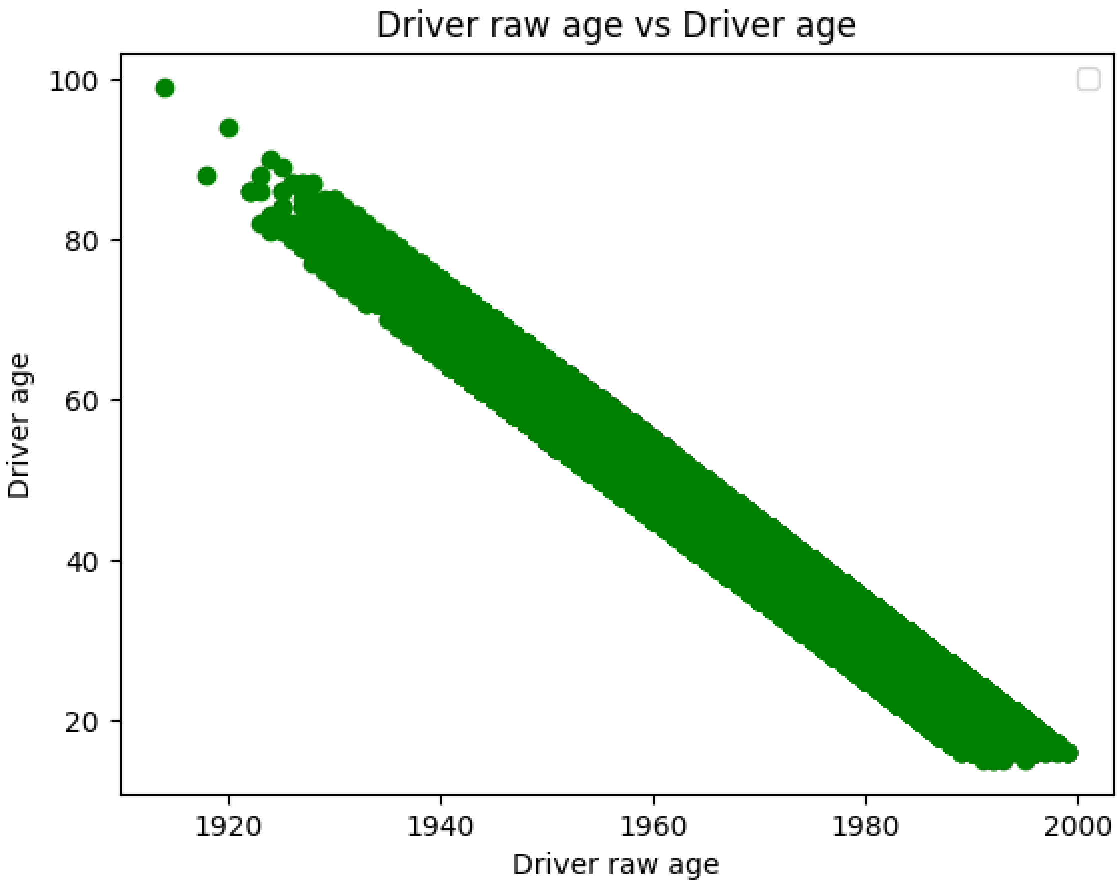


	stop_date	stop_time	driver_gender	driver_age_raw	driver_age	driver_race	violation_raw	violation	search_conducted	stop_outcome	is_arrested	stop_duration	drugs_related_stop
0	2005-01-02	01:55	M	1985.0	20.0	White	Speeding	Speeding	False	Citation	False	0-15 Min	False
1	2005-01-18	08:15	M	1965.0	40.0	White	Speeding	Speeding	False	Citation	False	0-15 Min	False
2	2005-01-23	23:15	M	1972.0	33.0	White	Speeding	Speeding	False	Citation	False	0-15 Min	False
3	2005-02-20	17:15	M	1986.0	19.0	White	Call for Service	Other	False	Arrest Driver	True	16-30 Min	False
4	2005-03-14	10:00	F	1984.0	21.0	White	Speeding	Speeding	False	Citation	False	0-15 Min	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
91736	2015-12-31	20:27	M	1986.0	29.0	White	Speeding	Speeding	False	Warning	False	0-15 Min	False
91737	2015-12-31	20:35	F	1982.0	33.0	White	Equipment/Inspection Violation	Equipment	False	Warning	False	0-15 Min	False
91738	2015-12-31	20:45	M	1992.0	23.0	White	Other Traffic Violation	Moving violation	False	Warning	False	0-15 Min	False
91739	2015-12-31	21:42	M	1993.0	22.0	White	Speeding	Speeding	False	Citation	False	0-15 Min	False
91740	2015-12-31	22:46	M	1959.0	56.0	Hispanic	Speeding	Speeding	False	Citation	False	0-15 Min	False

91741 rows × 13 columns



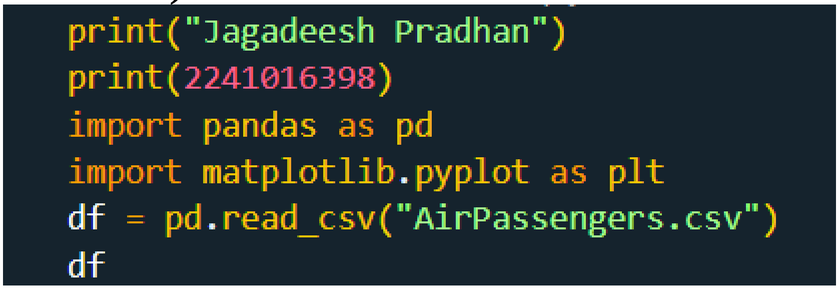
```
plt.scatter(df['driver_age_raw'],df['driver_age'] ,color = 'g')
plt.legend()
plt.xlabel("Driver raw age")
plt.ylabel("Driver age")
plt.title("Driver raw age vs Driver age")
plt.show()
```



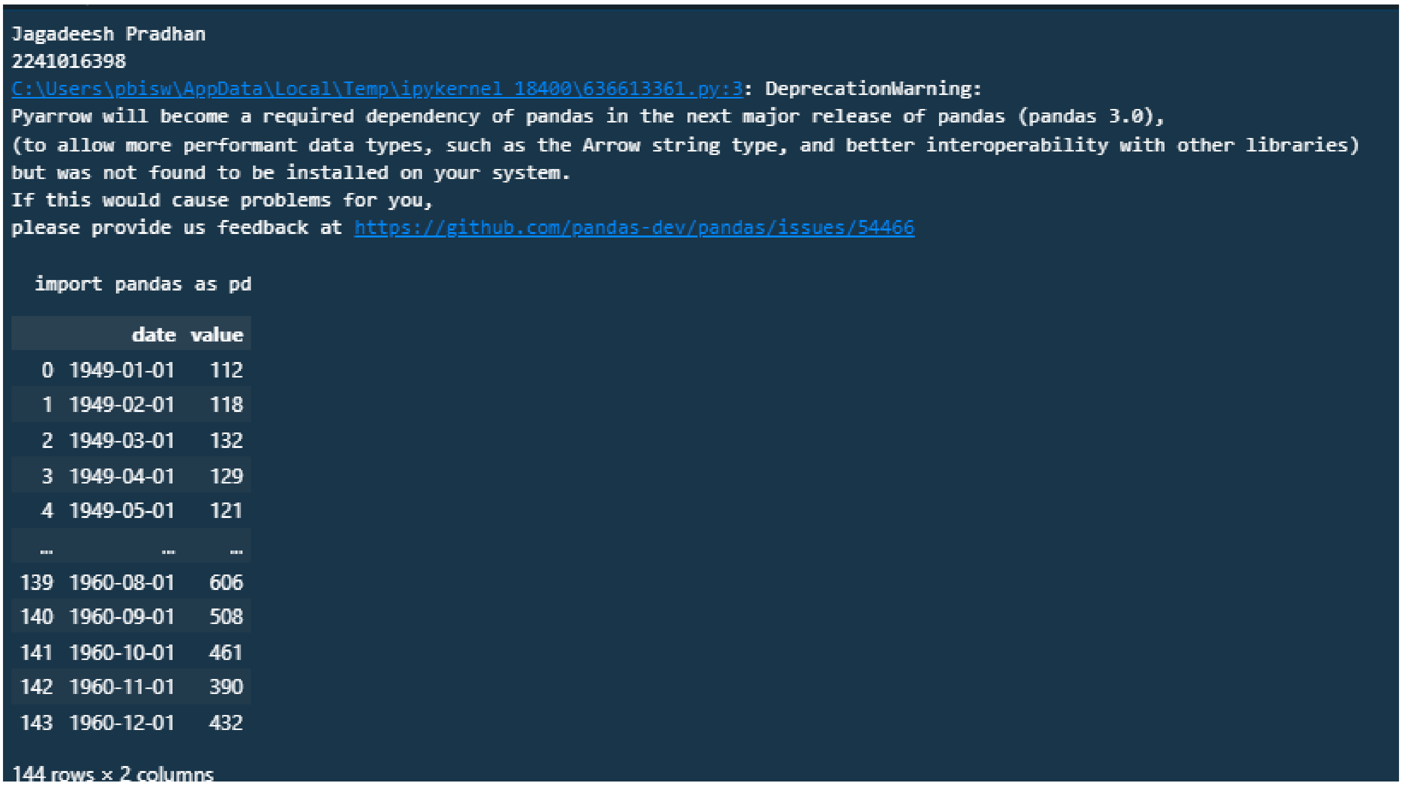
H-2) The time series plot visualises how a given metric changes over time. Use the given data and draw the time series plot to visualise how the Air Passenger traffic changed between 1949 and 1969.

Dataset: Data

Answer)



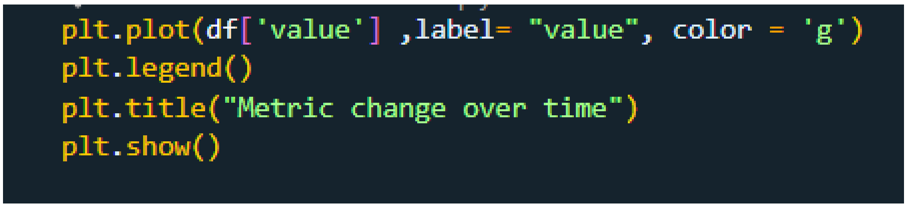
```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("AirPassengers.csv")
df
```



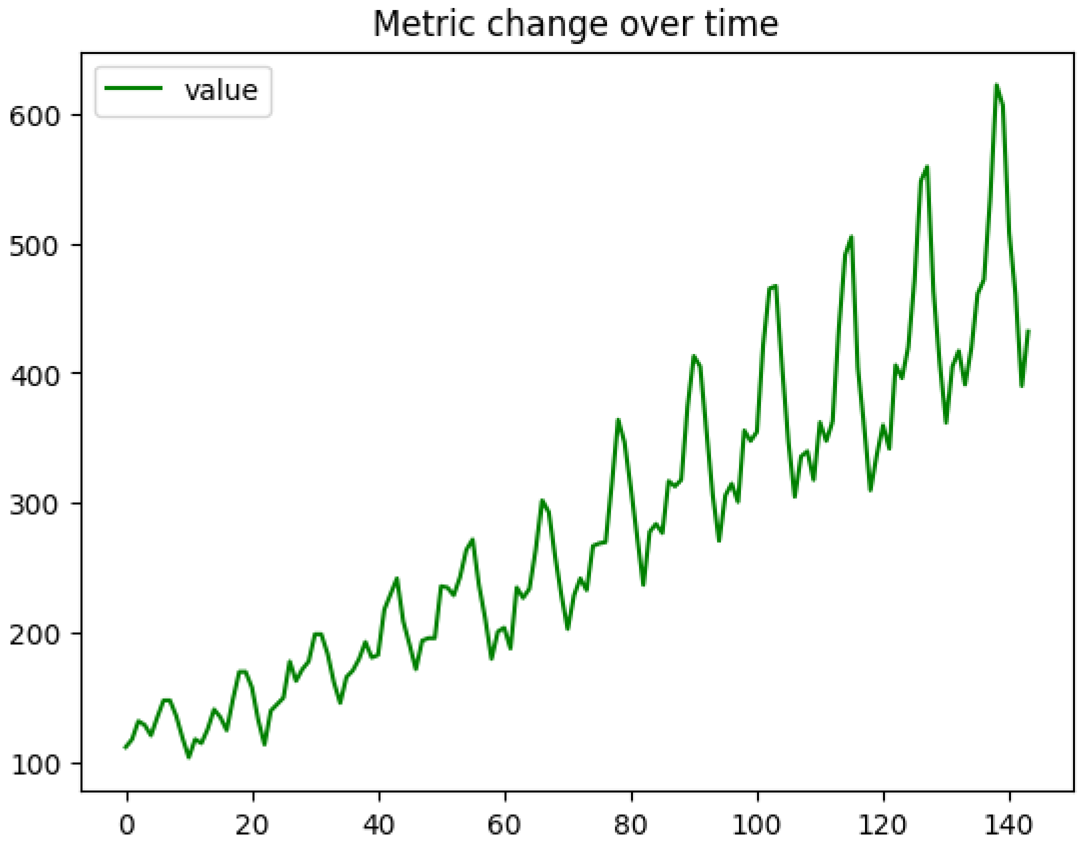
```
Jagadeesh Pradhan
2241016398
C:\Users\pbisw\AppData\Local\Temp\ipykernel_18400\636613361.py:3: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import pandas as pd
```

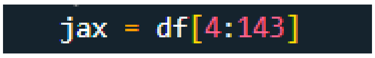
	date	value
0	1949-01-01	112
1	1949-02-01	118
2	1949-03-01	132
3	1949-04-01	129
4	1949-05-01	121
...	...	...
139	1960-08-01	606
140	1960-09-01	508
141	1960-10-01	461
142	1960-11-01	390
143	1960-12-01	432

144 rows × 2 columns



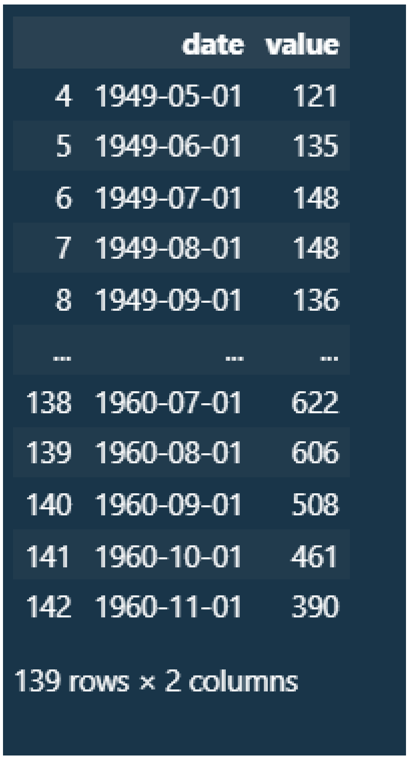
```
plt.plot(df['value'] ,label= "value", color = 'g')
plt.legend()
plt.title("Metric change over time")
plt.show()
```





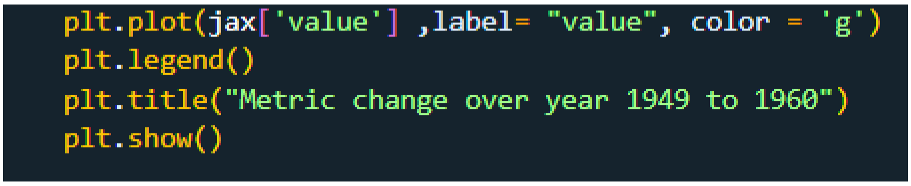
```
jax = df[4:143]
```

```
jax
```

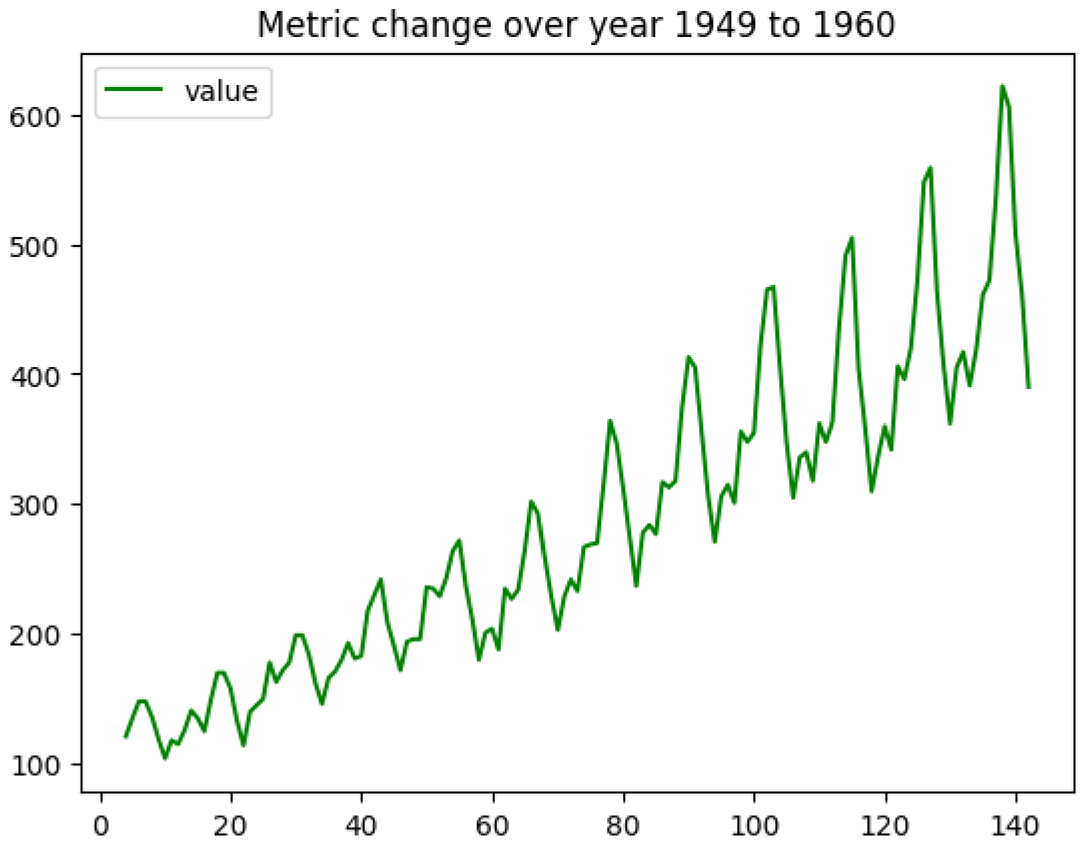


	date	value
4	1949-05-01	121
5	1949-06-01	135
6	1949-07-01	148
7	1949-08-01	148
8	1949-09-01	136
...	...	...
138	1960-07-01	622
139	1960-08-01	606
140	1960-09-01	508
141	1960-10-01	461
142	1960-11-01	390

139 rows × 2 columns

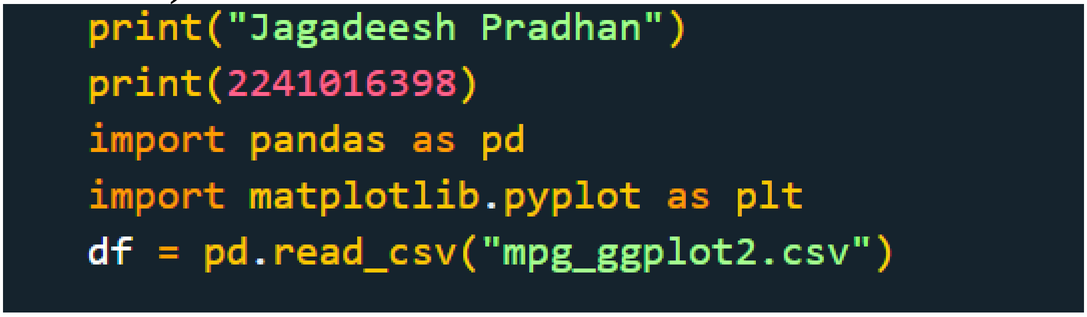


```
plt.plot(jax['value'] ,label= "value", color = 'g')
plt.legend()
plt.title("Metric change over year 1949 to 1960")
plt.show()
```

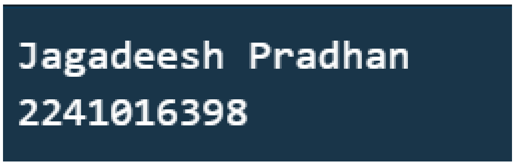


H-3) Read the dataset given in the following link. Print the info on the data and clean the data if required. Finally, draw the bar chart, where the x-axis has the ['manufacturer'] column, and the y-axis has the ['counts'] column.
Dataset: Data

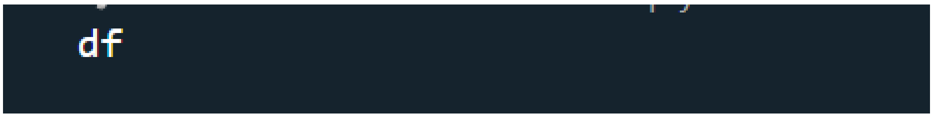
Answer)



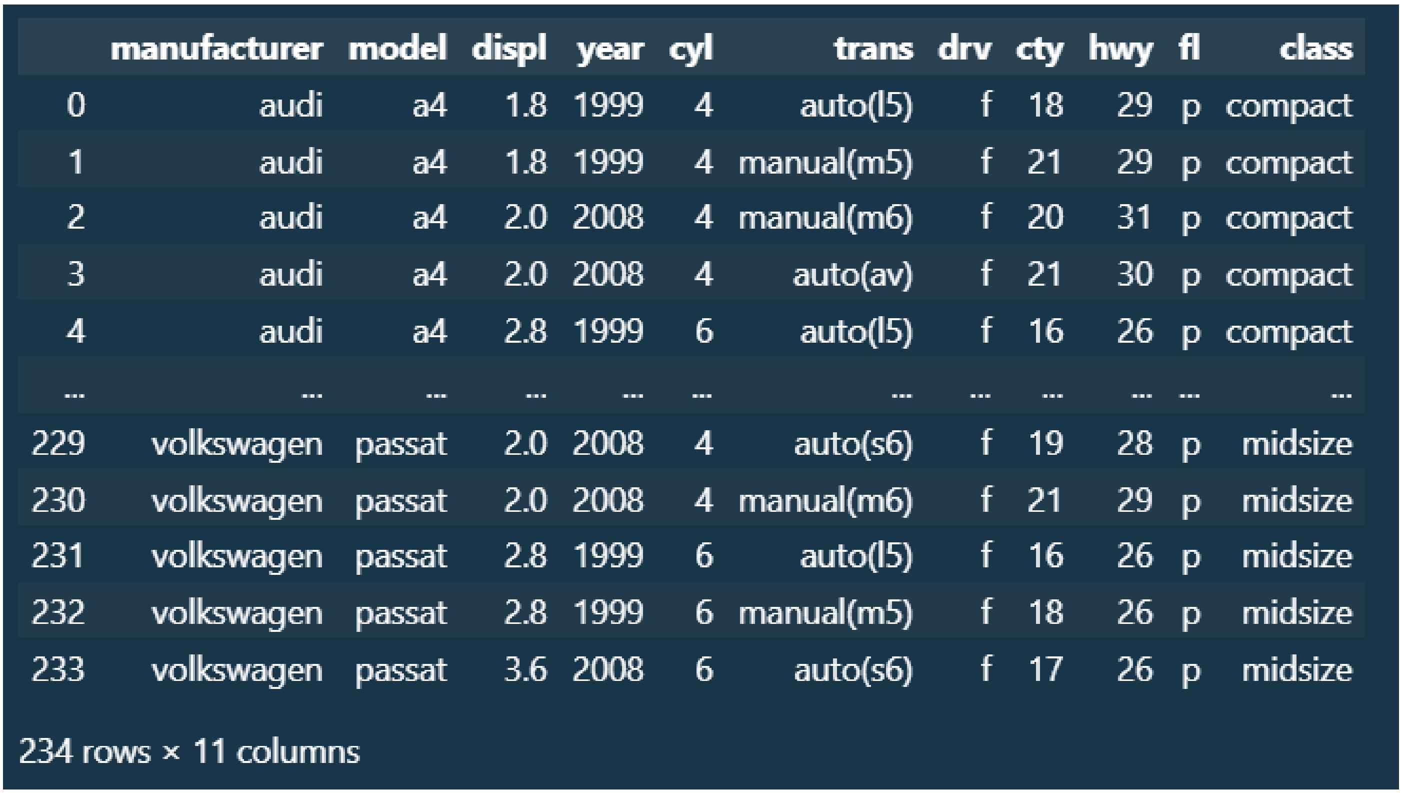
```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv("mpg_ggplot2.csv")
```



```
Jagadeesh Pradhan
2241016398
```

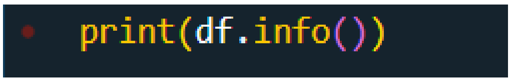


```
df
```

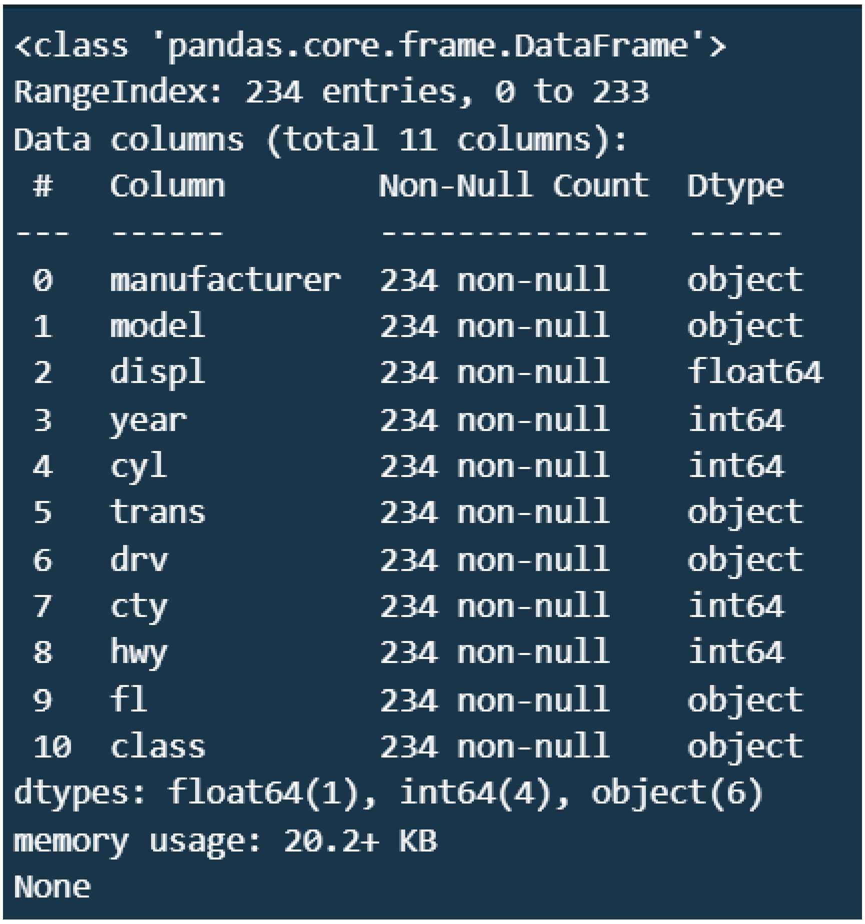


	manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
0	audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
1	audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
2	audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
3	audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
4	audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
...	...	...	...	...	...	...	...	...	...	...	...
229	volkswagen	passat	2.0	2008	4	auto(s6)	f	19	28	p	midsize
230	volkswagen	passat	2.0	2008	4	manual(m6)	f	21	29	p	midsize
231	volkswagen	passat	2.8	1999	6	auto(l5)	f	16	26	p	midsize
232	volkswagen	passat	2.8	1999	6	manual(m5)	f	18	26	p	midsize
233	volkswagen	passat	3.6	2008	6	auto(s6)	f	17	26	p	midsize

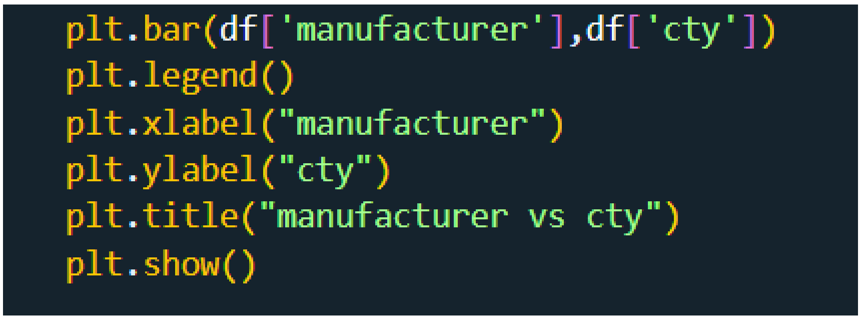
234 rows × 11 columns



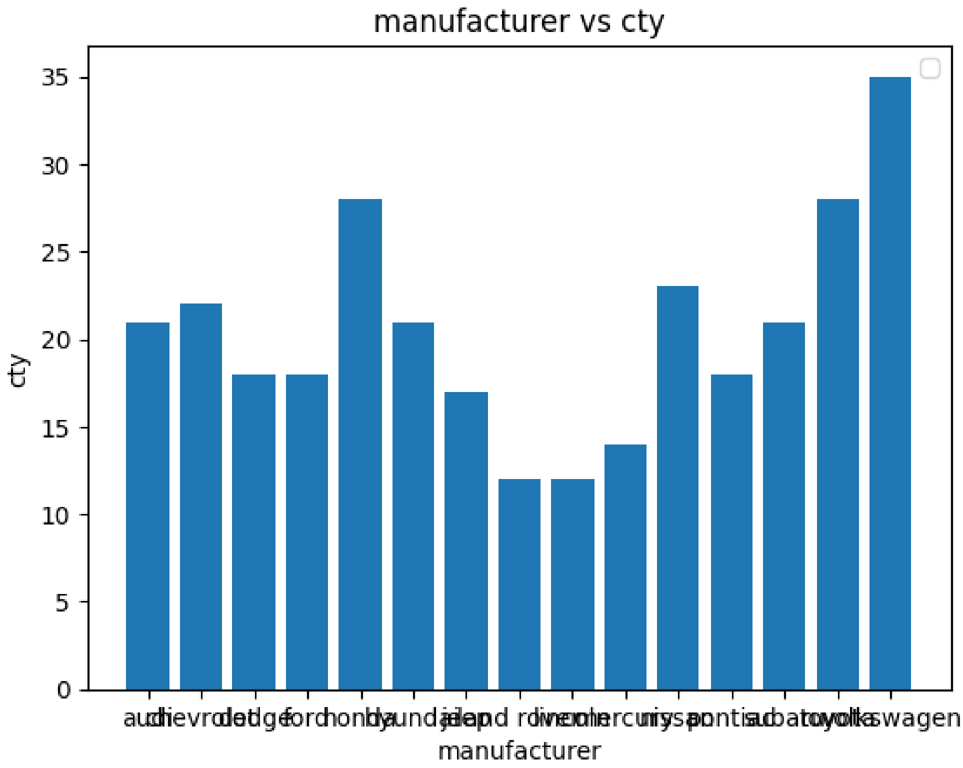
```
print(df.info())
```



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 234 entries, 0 to 233
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   manufacturer  234 non-null    object 
 1   model         234 non-null    object 
 2   displ         234 non-null    float64
 3   year          234 non-null    int64  
 4   cyl           234 non-null    int64  
 5   trans         234 non-null    object 
 6   drv           234 non-null    object 
 7   cty           234 non-null    int64  
 8   hwy           234 non-null    int64  
 9   fl            234 non-null    object 
 10  class         234 non-null    object 
dtypes: float64(1), int64(4), object(6)
memory usage: 20.2+ KB
None
```



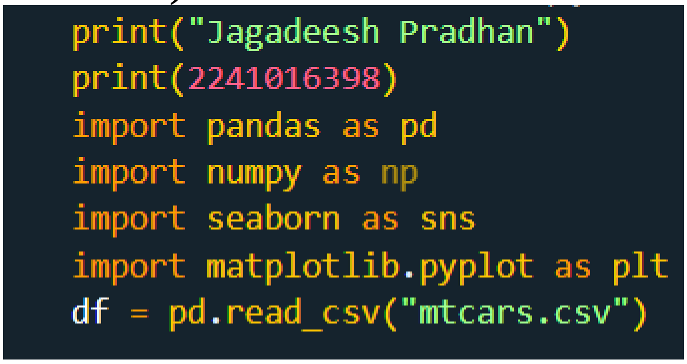
```
    plt.bar(df['manufacturer'],df['cty'])
    plt.legend()
    plt.xlabel("manufacturer")
    plt.ylabel("cty")
    plt.title("manufacturer vs cty")
    plt.show()
```



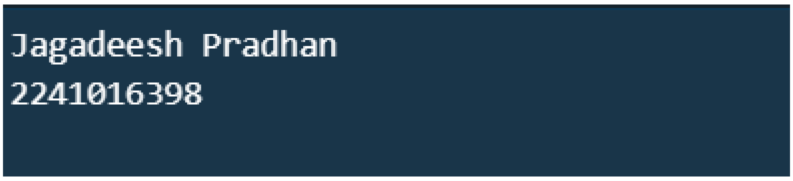
H-4) A correlogram is used to visually see the correlation metric between all possible pairs of
numeric variables in a given data frame (or 2D array). Import the dataset provided in the following link and plot the heatmap to visualize the correlation.

Dataset: Data

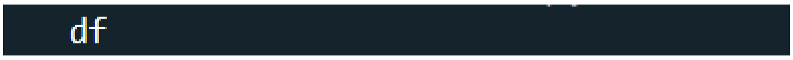
Answer)



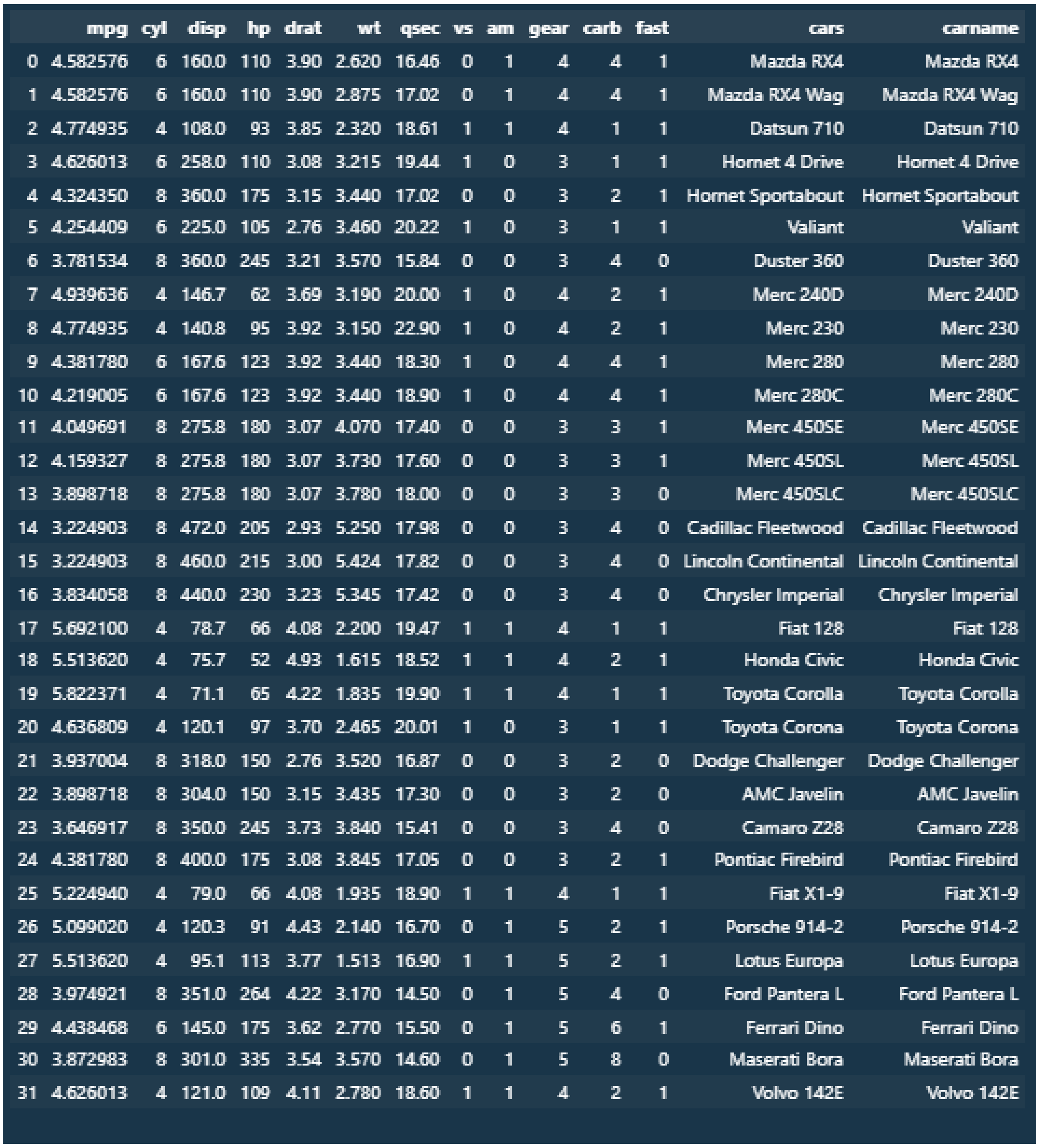
```
print("Jagadeesh Pradhan")
print(2241016398)
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv("mtcars.csv")
```



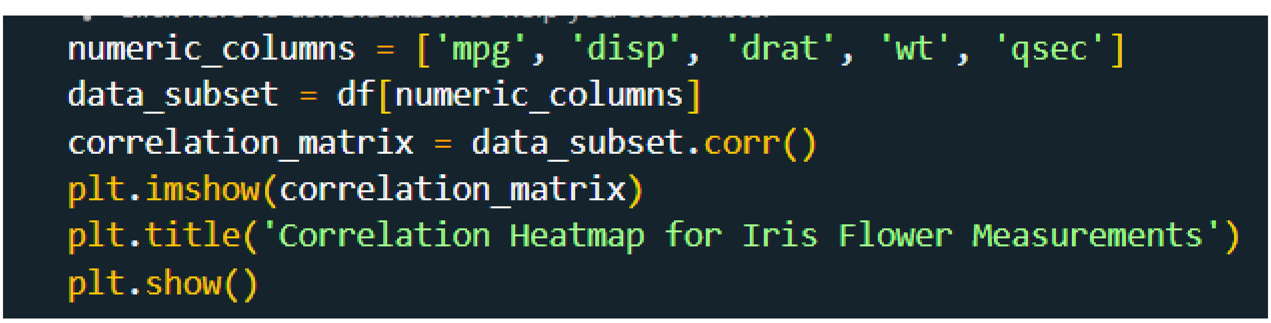
```
Jagadeesh Pradhan
2241016398
```



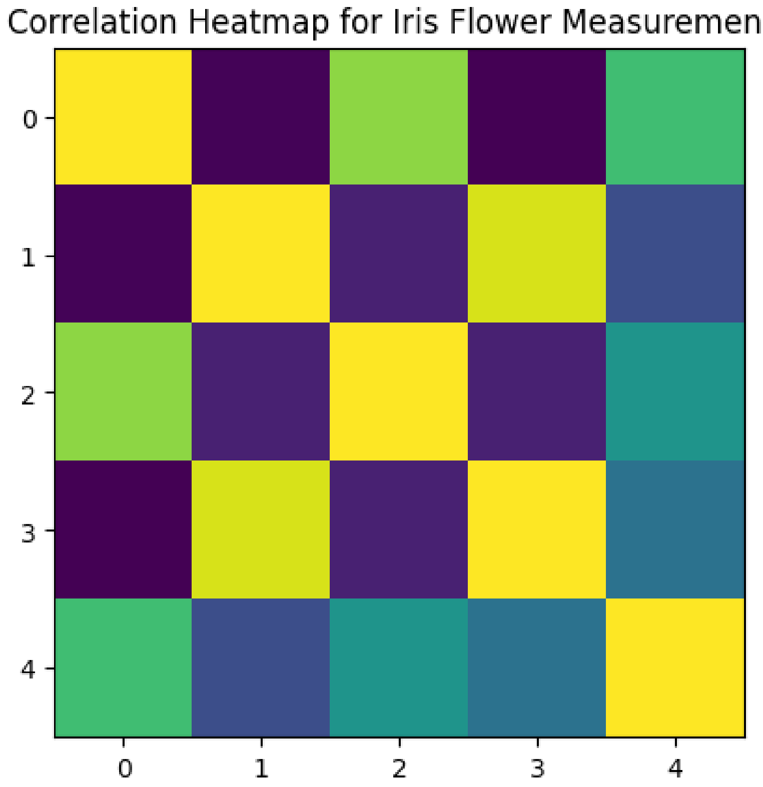
```
df
```

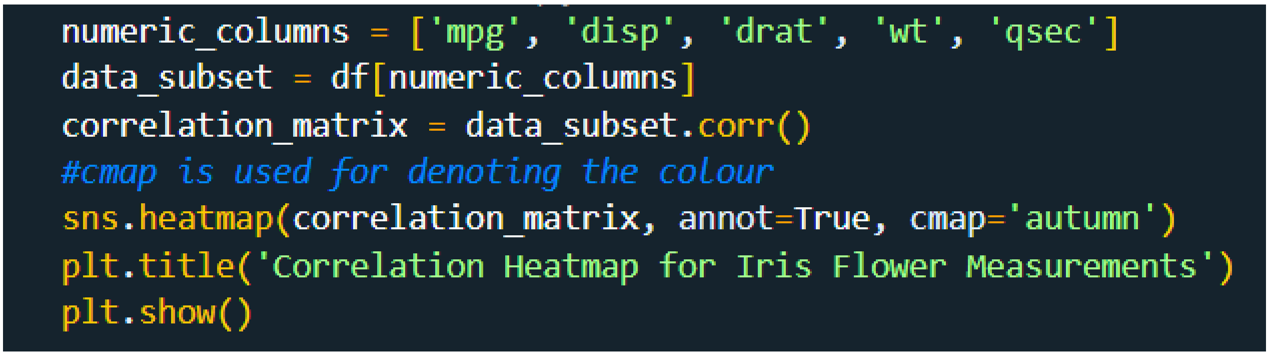


	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	fast	cars	carname
0	4.582576	6	160.0	110	3.90	2.620	16.46	0	1	4	4	1	Mazda RX4	Mazda RX4
1	4.582576	6	160.0	110	3.90	2.875	17.02	0	1	4	4	1	Mazda RX4 Wag	Mazda RX4 Wag
2	4.774935	4	108.0	93	3.85	2.320	18.61	1	1	4	1	1	Datsun 710	Datsun 710
3	4.626013	6	258.0	110	3.08	3.215	19.44	1	0	3	1	1	Hornet 4 Drive	Hornet 4 Drive
4	4.324350	8	360.0	175	3.15	3.440	17.02	0	0	3	2	1	Hornet Sportabout	Hornet Sportabout
5	4.254409	6	225.0	105	2.76	3.460	20.22	1	0	3	1	1	Valiant	Valiant
6	3.781534	8	360.0	245	3.21	3.570	15.84	0	0	3	4	0	Duster 360	Duster 360
7	4.939636	4	146.7	62	3.69	3.190	20.00	1	0	4	2	1	Merc 240D	Merc 240D
8	4.774935	4	140.8	95	3.92	3.150	22.90	1	0	4	2	1	Merc 230	Merc 230
9	4.381780	6	167.6	123	3.92	3.440	18.30	1	0	4	4	1	Merc 280	Merc 280
10	4.219005	6	167.6	123	3.92	3.440	18.90	1	0	4	4	1	Merc 280C	Merc 280C
11	4.049691	8	275.8	180	3.07	4.070	17.40	0	0	3	3	1	Merc 450SE	Merc 450SE
12	4.159327	8	275.8	180	3.07	3.730	17.60	0	0	3	3	1	Merc 450SL	Merc 450SL
13	3.898718	8	275.8	180	3.07	3.780	18.00	0	0	3	3	0	Merc 450SLC	Merc 450SLC
14	3.224903	8	472.0	205	2.93	5.250	17.98	0	0	3	4	0	Cadillac Fleetwood	Cadillac Fleetwood
15	3.224903	8	460.0	215	3.00	5.424	17.82	0	0	3	4	0	Lincoln Continental	Lincoln Continental
16	3.834058	8	440.0	230	3.23	5.345	17.42	0	0	3	4	0	Chrysler Imperial	Chrysler Imperial
17	5.692100	4	78.7	66	4.08	2.200	19.47	1	1	4	1	1	Fiat 128	Fiat 128
18	5.513620	4	75.7	52	4.93	1.615	18.52	1	1	4	2	1	Honda Civic	Honda Civic
19	5.822371	4	71.1	65	4.22	1.835	19.90	1	1	4	1	1	Toyota Corolla	Toyota Corolla
20	4.636809	4	120.1	97	3.70	2.465	20.01	1	0	3	1	1	Toyota Corona	Toyota Corona
21	3.937004	8	318.0	150	2.76	3.520	16.87	0	0	3	2	0	Dodge Challenger	Dodge Challenger
22	3.898718	8	304.0	150	3.15	3.435	17.30	0	0	3	2	0	AMC Javelin	AMC Javelin
23	3.646917	8	350.0	245	3.73	3.840	15.41	0	0	3	4	0	Camaro Z28	Camaro Z28
24	4.381780	8	400.0	175	3.08	3.845	17.05	0	0	3	2	1	Pontiac Firebird	Pontiac Firebird
25	5.224940	4	79.0	66	4.08	1.935	18.90	1	1	4	1	1	Fiat X1-9	Fiat X1-9
26	5.099020	4	120.3	91	4.43	2.140	16.70	0	1	5	2	1	Porsche 914-2	Porsche 914-2
27	5.513620	4	95.1	113	3.77	1.513	16.90	1	1	5	2	1	Lotus Europa	Lotus Europa
28	3.974921	8	351.0	264	4.22	3.170	14.50	0	1	5	4	0	Ford Pantera L	Ford Pantera L
29	4.438468	6	145.0	175	3.62	2.770	15.50	0	1	5	6	1	Ferrari Dino	Ferrari Dino
30	3.872983	8	301.0	335	3.54	3.570	14.60	0	1	5	8	0	Maserati Bora	Maserati Bora
31	4.626013	4	121.0	109	4.11	2.780	18.60	1	1	4	2	1	Volvo 142E	Volvo 142E

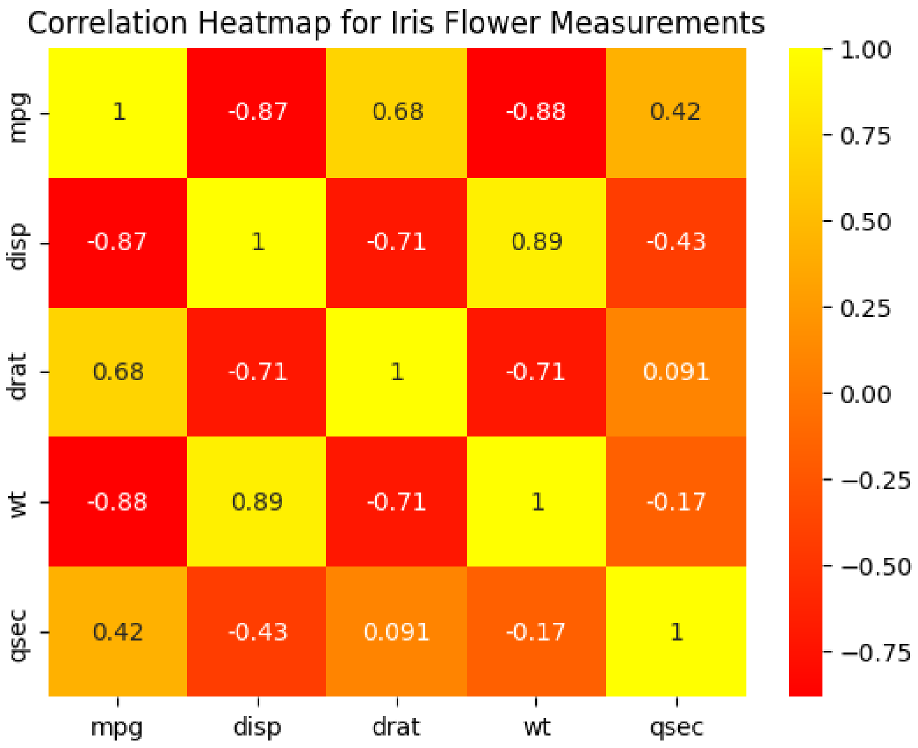


```
numeric_columns = ['mpg', 'disp', 'drat', 'wt', 'qsec']
data_subset = df[numeric_columns]
correlation_matrix = data_subset.corr()
plt.imshow(correlation_matrix)
plt.title('Correlation Heatmap for Iris Flower Measurements')
plt.show()
```





```
numeric_columns = ['mpg', 'disp', 'drat', 'wt', 'qsec']
data_subset = df[numeric_columns]
correlation_matrix = data_subset.corr()
#cmap is used for denoting the colour
sns.heatmap(correlation_matrix, annot=True, cmap='autumn')
plt.title('Correlation Heatmap for Iris Flower Measurements')
plt.show()
```


Correlation Heatmap for Iris Flower Measurements
mpg
disp
drat
wt
qsec
mpg
disp
drat
wt
qsec