

```
In [6]: #Question 1:How to import pandas and check the version?
import pandas as pd
print(pd.__version__)
```

0.25.1

```
In [7]: #Question 2:How to create a series from a numpy array?
import pandas as pd
import numpy as np
data = np.array(['a', 'e', 'i', 'o', 'u'])
s = pd.Series(data)
print(s)
```

```
0    a
1    e
2    i
3    o
4    u
dtype: object
```

```
In [21]: #Question 3: How to convert the index of a series into a column of a dataframe?
population_dict={
    "TUMKUR":989898,
    "BANGALORE":1666666,
    "MYSORE":4444444,
    "HUBLI":7777777,
}
print(population_dict)
print(type(population_dict))
population=pd.Series(population_dict)
population
type(population)
pd.DataFrame(population)
```

```
{'TUMKUR': 989898, 'BANGALORE': 1666666, 'MYSORE': 4444444, 'HUBLI': 7777777}
<class 'dict'>
```

Out[21]:

	0
TUMKUR	989898
BANGALORE	1666666
MYSORE	4444444
HUBLI	7777777

```
In [8]: # Question 4: Write the code to list all the datasets available in seaborn library
import seaborn as sbn
sbn.get_dataset_names()
```

C:\Users\ADMIN\Anaconda3\lib\site-packages\seaborn\utils.py:376: UserWarning: No parser was explicitly specified, so I'm using the best available HTML parser for this system ("lxml"). This usually isn't a problem, but if you run this code on another system, or in a different virtual environment, it may use a different parser and behave differently.

The code that caused this warning is on line 376 of the file C:\Users\ADMIN\Anaconda3\lib\site-packages\seaborn\utils.py. To get rid of this warning, pass the additional argument 'features="lxml"' to the BeautifulSoup constructor.

```
gh_list = BeautifulSoup(http)
```

```
Out[8]: ['anagrams',
        'anscombe',
        'attention',
        'brain_networks',
        'car_crashes',
        'diamonds',
        'dots',
        'exercise',
        'flights',
        'fmri',
        'gammas',
        'geyser',
        'iris',
        'mpg',
        'penguins',
        'planets',
        'tips',
        'titanic']
```

```
In [9]: data=sbn.load_dataset('mpg')
data
```

Out[9]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel s/s
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino
...
393	27.0	4	140.0	86.0	2790	15.6	82	usa	ford mustang
394	44.0	4	97.0	52.0	2130	24.6	82	europa	vw pickup
395	32.0	4	135.0	84.0	2295	11.6	82	usa	dodge rampage
396	28.0	4	120.0	79.0	2625	18.6	82	usa	ford range
397	31.0	4	119.0	82.0	2720	19.4	82	usa	chevy s-10

398 rows × 9 columns



```
In [10]: # Question 5: Which country origin cars are a part of this dataset?
data['origin'].unique()
```

Out[10]: array(['usa', 'japan', 'europa'], dtype=object)

In [11]: *# Questions 6: Extract the part of the dataframe which contains cars belonging to*

```
data[data['origin']=='usa']
```

Out[11]:

	mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
0	18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
1	15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
2	18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
3	16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
4	17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino
...
392	27.0	4	151.0	90.0	2950	17.3	82	usa	chevrolet camaro
393	27.0	4	140.0	86.0	2790	15.6	82	usa	ford mustang gl
395	32.0	4	135.0	84.0	2295	11.6	82	usa	dodge rampage
396	28.0	4	120.0	79.0	2625	18.6	82	usa	ford ranger
397	31.0	4	119.0	82.0	2720	19.4	82	usa	chevy s- 10

249 rows × 9 columns



In []: