

1.Ans

```
In [4]: # importing pandas

import pandas as ps

# pandas Version

print(ps.__version__)
```

0.25.1

2.Ans

```
In [9]: # Creating Series from a numpy array

import numpy as np

# Creating numpy array

arr=np.array([1,3,6,8,10,11])

type(arr)
```

Out[9]: numpy.ndarray

```
In [35]: # Converting numpy array into Series

pan_ser = ps.Series(arr)

pan_ser
```

Out[35]:

0	1
1	3
2	6
3	8
4	10
5	11

dtype: int32

```
In [17]: type(pan_ser)
```

Out[17]: pandas.core.series.Series

3.Ans

```
In [21]: # Creating Series along with index value

ser=ps.Series([1,2,3,4,5],index=['a','b','c','d','e'])

ser
```

```
Out[21]: a    1
         b    2
         c    3
         d    4
         e    5
         dtype: int64
```

```
In [39]: print(ser[3], ser['a'], type(ser))

4 1 <class 'pandas.core.series.Series'>
```

```
In [34]: # Converting the index of a series into a column of a dataframe

df=ps.DataFrame(ser,columns = ["value"])

df
```

```
Out[34]:
```

	value
a	1
b	2
c	3
d	4
e	5

```
In [38]: type(df)
```

```
Out[38]: pandas.core.frame.DataFrame
```

4.Ans

```
In [52]: # Importing seaborn

import seaborn as sb

# Listing all the datasets in seaborn library

sb.get_dataset_names()
```

```
Out[52]: ['anagrams',
          'anscombe',
          'attention',
          'brain_networks',
          'car_crashes',
          'diamonds',
          'dots',
          'exercise',
          'flights',
          'fmri',
          'gammas',
          'geyser',
          'iris',
          'mpg',
          'penguins',
          'planets',
          'tips',
          'titanic']
```

```
In [57]: # Loading mpg dataset

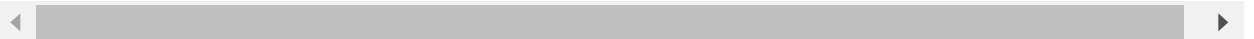
mpg=sb.load_dataset('mpg')

mpg
```

Out[57]:

	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 s10

398 rows × 9 columns



5.Ans

```
In [59]: # Country of origin

mpg['origin'].unique()
```

Out[59]: array(['usa', 'japan', 'europa'], dtype=object)

6.Ans

```
In [64]: # Extracting the part of the dataframe which contains cars belonging to 'usa'

mpg1=mpg[mpg['origin'].str.contains("usa")]
```

```
In [65]: mpg1
```

```
Out[65]:
```

	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

