

```

1 import numpy as np
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 import seaborn as sns
5 %matplotlib inline
6
7

```

+ Code

+ Text

```

[ ] 1 data=pd.read_csv('Admission_predict.csv')
2 data
3

```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

0s completed at 7:18 PM

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67
399	400	333	117	4	5.0	4.0	9.66	1	0.95

400 rows x 9 columns

```

1 data.info()
2
3

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   Serial No.            400 non-null   int64
1   GRE Score              400 non-null   int64
2   TOEFL Score            400 non-null   int64
3   University Rating      400 non-null   int64
4   SOP                    400 non-null   float64
5   LOR                    400 non-null   float64
6   CGPA                   400 non-null   float64
7   Research               400 non-null   int64
8   Chance of Admit        400 non-null   float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB

```

```

1 data.isnull().any()
2

```

```
1 sns.distplot(data['GRE Score'])
```

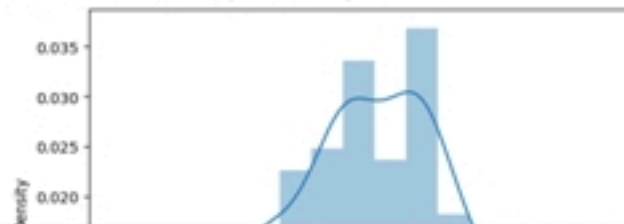
<ipython-input-32-64e93544a305>:1: UserWarning:

'distplot' is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either 'displot' (a figure-level function with similar flexibility) or 'histplot' (an axes-level function for histograms).

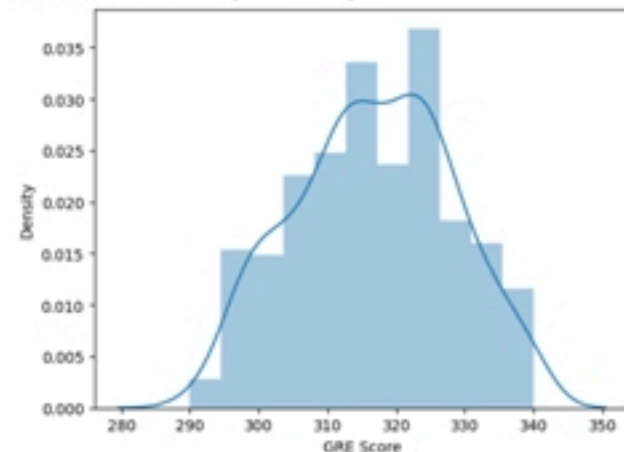
For a guide to updating your code to use the new functions, please see <https://gist.github.com/mwaskom/de66167ec2974657ad6172750bbe5751>

```
sns.distplot(data['GRE Score'])  
<Axes: xlabel='GRE Score', ylabel='Density'>
```



0s completed at 7:18 PM

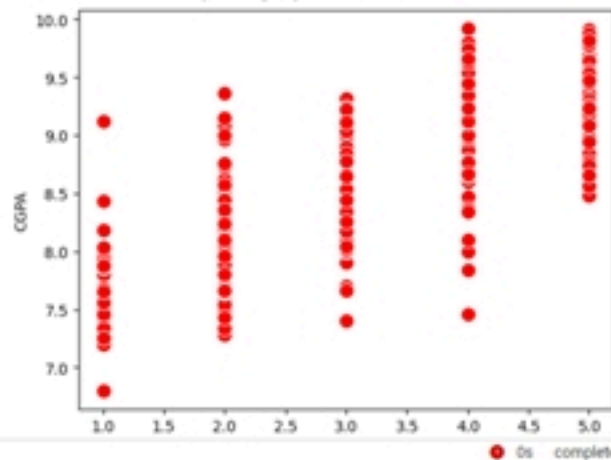
<Axes: xlabel='GRE Score', ylabel='Density'>



0s completed at 7:18 PM

```
1 sns.scatterplot(x='University Rating',y='CGPA',data=data,color='Red',s=100)
```

```
<Axes: xlabel='University Rating', ylabel='CGPA'>
```



0s completed at 7:18 PM

```
[ ] 1 x=data.iloc[:,0:7].values
    2 x
    3
```

```
array([[ 1.,  337.,  118., ...,  4.5,  4.5,  9.65],
       [ 2.,  324.,  107., ...,  4.,  4.5,  8.87],
       [ 3.,  316.,  104., ...,  3.,  3.5,  8. ],
       ...,
       [398.,  330.,  116., ...,  5.,  4.5,  9.45],
       [399.,  312.,  103., ...,  3.5,  4.,  8.78],
       [400.,  333.,  117., ...,  5.,  4.,  9.66]])
```

```
1 y=data.iloc[:,7:].values
2 y
```

```
<array([[1., 0.92],
       [1., 0.76],
       [1., 0.72],
       [1., 0.8 ],
       [0., 0.65],
       [1., 0.9 ],
       [1., 0.75],
       [0., 0.68],
       [0., 0.5 ],
       [0., 0.45],
       ...])
```

0s completed at 7:18 PM

```
1 data.info()
2
3
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Serial No.            400 non-null   int64
1   GRE Score              400 non-null   int64
2   TOEFL Score            400 non-null   int64
3   University Rating      400 non-null   int64
4   SOP                    400 non-null   float64
5   LOR                    400 non-null   float64
6   CGPA                   400 non-null   float64
7   Research               400 non-null   int64
8   Chance of Admit        400 non-null   float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```

```
1 data.isnull().any()
2
```

```
Serial No.      False
GRE Score       False
TOEFL Score     False
University Rating False
SOP             False
LOR             False
CGPA            False
Research        False
Chance of Admit False
dtype: bool
```

```
[ ] 1 #let us rename the column Chance of Admit becuse it has training space
    2 data=data.rename (columns = {'Chance of Admit ':' Chance of Admit'})
```

```
1 data.describe()
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	1.000000	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	100.750000	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	200.500000	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	300.250000	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	400.000000	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000



```
0. , 0.45 ,  
1. , 0.52 ,  
1. , 0.84 ,  
1. , 0.78 ,  
1. , 0.62 ,  
1. , 0.61 ,  
0. , 0.54 ,  
0. , 0.66 ,  
1. , 0.65 ,  
0. , 0.63 ,  
0. , 0.62 ,  
1. , 0.64 ,  
0. , 0.7 ,  
1. , 0.94 ,  
1. , 0.95 ,  
1. , 0.97 ,  
1. , 0.94 ,  
0. , 0.76 ,  
1. , 0.44 ,  
0. , 0.46 ,  
0. , 0.54 ,  
1. , 0.65 ,  
1. , 0.74 ,  
1. , 0.91 ,  
1. , 0.9 ,  
1. , 0.94 ,  
1. , 0.88 ,
```

0s completed at 7:18 PM

