MINI PROJECT 1

Name -Sameer Sanjay Barde

College-National Institute of Food Entrepreneurship Management

Year-M-Tech 2nd year

URL of dataset-
https://www.kaggle.com/datasets/akshaydattatraykhare/data-for-admission-in-the-university

```
#create dataframe
import pandas as pd
df=pd.read_csv('/content/adm_data (1).csv')
df
```

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

400 rows × 9 columns

```
df.info
```

```
<bound method DataFrame.info of      Serial No.  GRE Score  TOEFL Score  University
Rating  SOP  LOR   CGPA  \
0             1        337          118                  4  4.5   4.5  9.65
1             2        324          107                  4  4.0   4.5  8.87
2             3        316          104                  3  3.0   3.5  8.00
3             4        322          110                  3  3.5   2.5  8.67
4             5        314          103                  2  2.0   3.0  8.21
..          ...        ...          ...                ...  ...   ...   ...
395         396        324          110                  3  3.5   3.5  9.04
396         397        325          107                  3  3.0   3.5  9.11
397         398        330          116                  4  5.0   4.5  9.45
398         399        312          103                  3  3.5   4.0  8.78
399         400        333          117                  4  5.0   4.0  9.66

     Research  Chance of Admit
0           1             0.92
1           1             0.76
2           1             0.72
3           1             0.80
4           0             0.65
..        ...              ...
395         1             0.82
```

```
     396         1              0.84
     397         1              0.91
     398         0              0.67
     399         1              0.95

     [400 rows x 9 columns]>
```

```
df.shape
```

```
     (400, 9)
```

```
df.size #Total number of elements in the dataframe
```

```
     3600
```

```
#TO officially check the null values or missing values
df.isnull().sum()
```

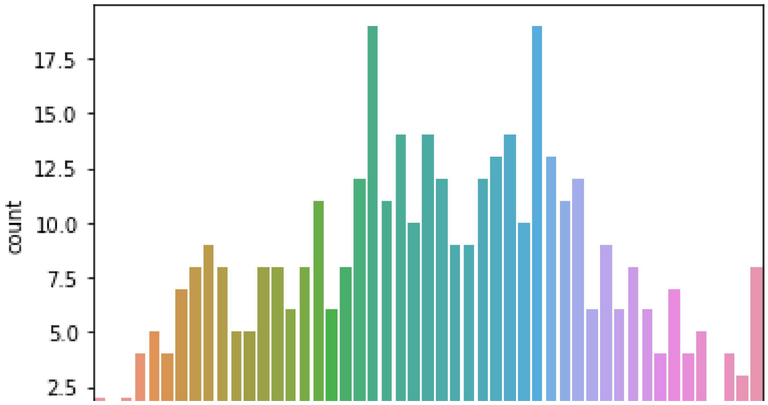
```
     Serial No.           0
     GRE Score            0
     TOEFL Score          0
     University Rating    0
     SOP                  0
     LOR                  0
     CGPA                 0
     Research             0
     Chance of Admit      0
     dtype: int64
```

```
#I want to find out the exact count of unique elements in each and every column
df.nunique()
```

```
     Serial No.           400
     GRE Score             49
     TOEFL Score           29
     University Rating      5
     SOP                    9
     LOR                    9
     CGPA                 168
     Research               2
     Chance of Admit       60
     dtype: int64
```

```
#VISUALISATION
import seaborn as sns
sns.countplot(x = 'GRE Score',data = df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fab2751b410>
```



```
#To find how many gets more same GRE score
df.groupby('GRE Score').size()
```

```
GRE Score
290     2
293     1
294     2
295     4
296     5
297     4
298     7
299     8
300     9
301     8
302     5
303     5
304     8
305     8
306     6
307     8
308    11
309     6
310     8
311    12
312    19
313    11
314    14
315    10
316    14
317    12
318     9
319     9
320    12
321    13
322    14
323    10
324    19
325    13
326    11
327    12
328     6
329     9
```

```
     330     6
     331     8
     332     6
     333     4
     334     7
     335     4
     336     5
     337     1
     338     4
     339     3
     340     8
     dtype: int64
```

```
#To find mostly repeated CGPA
df.groupby('CGPA').size()
```

```
     CGPA
     6.80    1
     7.20    1
     7.25    1
     7.28    1
     7.30    1
            ..
     9.80    3
     9.82    1
     9.87    1
     9.91    1
     9.92    1
     Length: 168, dtype: int64
```

```
#to find repeated GRE score and CGPA
a=df.groupby(["GRE Score","CGPA"]).size()
a
```

```
     GRE Score  CGPA
     290        7.46    1
                7.56    1
     293        7.80    1
     294        7.36    1
                7.64    1
                       ..
     340        9.60    2
                9.66    1
                9.74    1
                9.91    1
                9.92    1
     Length: 385, dtype: int64
```

```
import numpy as np
a
```

```
     GRE Score  CGPA
     290        7.46    1
```

```
             7.56    1
     293     7.80    1
     294     7.36    1
             7.64    1
                    ..
     340     9.60    2
             9.66    1
             9.74    1
             9.91    1
             9.92    1
     Length: 385, dtype: int64
```

```
#to find maximum CGPA
np.max(df['CGPA'])
```

```
     9.92
```

```
#to find maximum GRE
np.max(df['GRE Score'])
```

```
     340
```

```
np.min(df['GRE Score'])
```

```
     290
```

```
np.min(df['CGPA'])
```

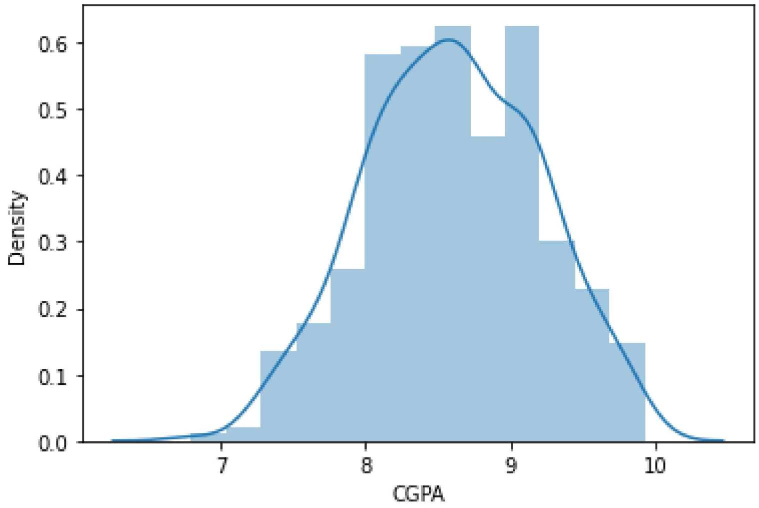
```
     6.8
```

```
#To find  total toppers
Toppers = np.sum((df['CGPA']>=9.92))
Toppers
```

```
     1
```

```
#DISTRIBUTION PLOT
sns.distplot(df['CGPA'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning: `d
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fab26e79650>
```



0s completed at 9:27 PM