Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №1
по дисциплине
«Методы машинного обучения»
на тему
«Методы обработки данных»

Выполнил:
студент группы ИУ5-24М
Зубаиров В. А.

Москва — 2020 г.

1. Зубаиров Валерий, ИУ5-24М

1.1. РК №1 по курсу ММО

1.1.1. МГТУ им. Н. Э. Баумана, Москва

1.2. Вариант

Вариант №6

1.3. Задача:

- Для заданного набора данных построить основные графики, входящие в этап разведочного анализа данных с использованием библиотек Matplotlib и Seaborn.
- В случае наличия пропусков в данных удалить строки или колонки, содержащие пропуски.
- Провести корреляционный анализ.
- Сделать выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.
- Построить Violin plot для одного из параметров

1.4. Датасет

This dataset is created for prediction of Graduate Admissions from an Indian perspective.
The dataset contains several parameters which are considered important during the application for Masters Programs.
The parameters included are :
GRE Scores (out of 340)
TOEFL Scores (out of 120)
University Rating (out of 5)
Statement of Purpose and Letter of Recommendation Strength (out of 5)
Undergraduate GPA (out of 10)
Research Experience (either 0 or 1)
Chance of Admit (ranging from 0 to 1)

[0]:
```
import numpy as np
import pandas as pd
import seaborn as sns

import matplotlib.pyplot as plt
import matplotlib.mlab as mlab
import matplotlib
plt.style.use('ggplot')
from matplotlib.pyplot import figure

%matplotlib inline
matplotlib.rcParams['figure.figsize'] = (12,8)
```

[0]:
```
filename =  "./V.csv"
```

[0]:
```
data = pd.read_csv(filename)
```

[8]:
```
data.info()
```

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

[9]:
```
data.describe()
```

[9]:

	Serial No.	GRE Score	…	Research	Chance of Admit
count	400.000000	400.000000	…	400.000000	400.000000
mean	200.500000	316.807500	…	0.547500	0.724350
std	115.614301	11.473646	…	0.498362	0.142609
min	1.000000	290.000000	…	0.000000	0.340000
25%	100.750000	308.000000	…	0.000000	0.640000
50%	200.500000	317.000000	…	1.000000	0.730000
75%	300.250000	325.000000	…	1.000000	0.830000
max	400.000000	340.000000	…	1.000000	0.970000

[8 rows x 9 columns]

[10]:
```
data.corr()
```

[10]:

	Serial No.	GRE Score	…	Research	Chance of Admit
Serial No.	1.000000	-0.097526	…	-0.063138	0.042336
GRE Score	-0.097526	1.000000	…	0.580391	0.802610
TOEFL Score	-0.147932	0.835977	…	0.489858	0.791594
University Rating	-0.169948	0.668976	…	0.447783	0.711250
SOP	-0.166932	0.612831	…	0.444029	0.675732
LOR	-0.088221	0.557555	…	0.396859	0.669889
CGPA	-0.045608	0.833060	…	0.521654	0.873289
Research	-0.063138	0.580391	…	1.000000	0.553202
Chance of Admit	0.042336	0.802610	…	0.553202	1.000000

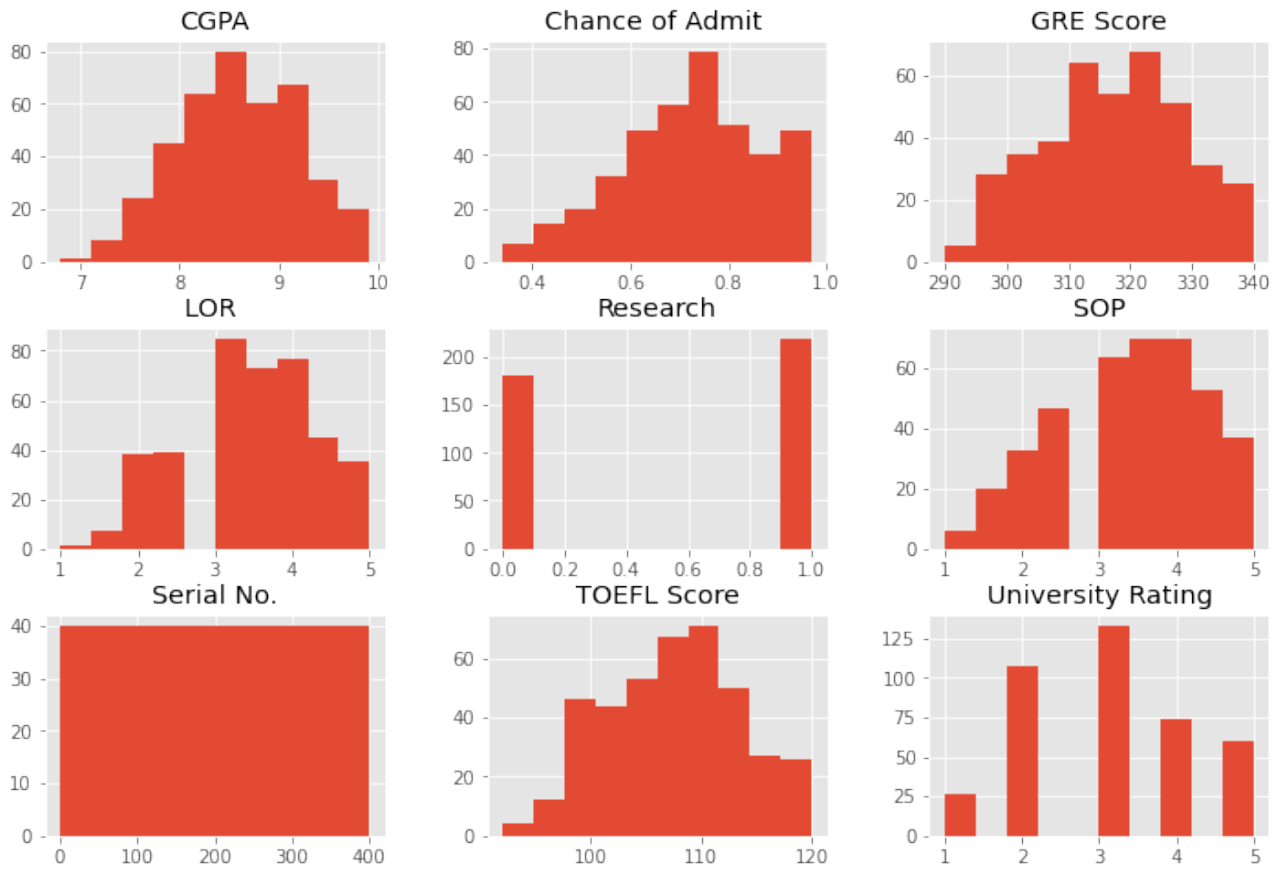
[9 rows x 9 columns]

[11]:
```
data.hist()
```

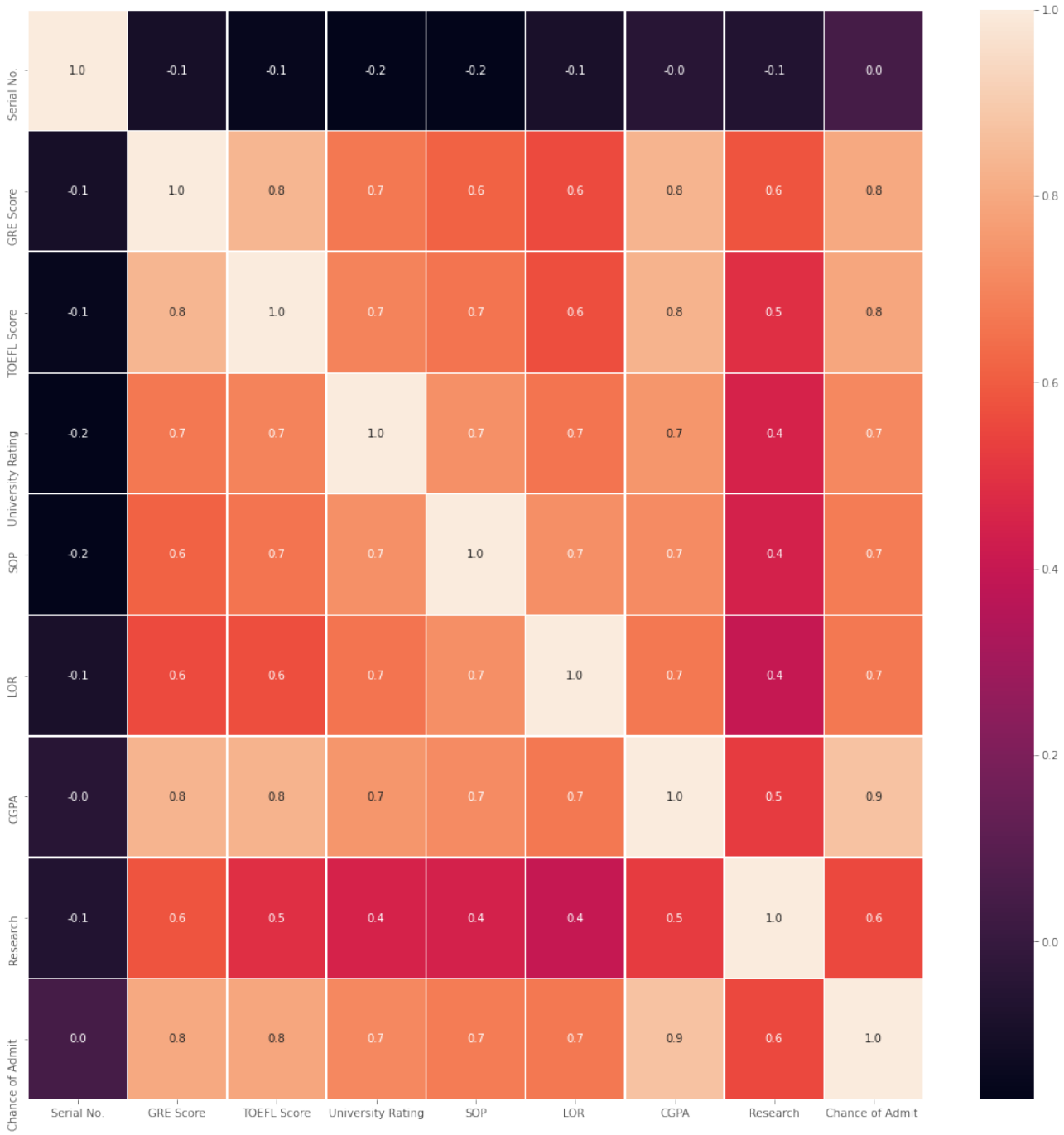
```
/usr/local/lib/python3.6/dist-packages/pandas/plotting/_matplotlib/tools.py:307:
MatplotlibDeprecationWarning:
```

```
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
/usr/local/lib/python3.6/dist-packages/pandas/plotting/_matplotlib/tools.py:307:
MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  layout[ax.rowNum, ax.colNum] = ax.get_visible()
/usr/local/lib/python3.6/dist-packages/pandas/plotting/_matplotlib/tools.py:313:
MatplotlibDeprecationWarning:
The rowNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().rowspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
/usr/local/lib/python3.6/dist-packages/pandas/plotting/_matplotlib/tools.py:313:
MatplotlibDeprecationWarning:
The colNum attribute was deprecated in Matplotlib 3.2 and will be removed two
minor releases later. Use ax.get_subplotspec().colspan.start instead.
  if not layout[ax.rowNum + 1, ax.colNum]:
```

[11]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fb061291940>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb06123c828>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb0611eca90>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb06119fcf8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb06114ff60>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb061108208>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fb06113c470>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb0610ea6a0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fb0610ea710>]],
      dtype=object)
```



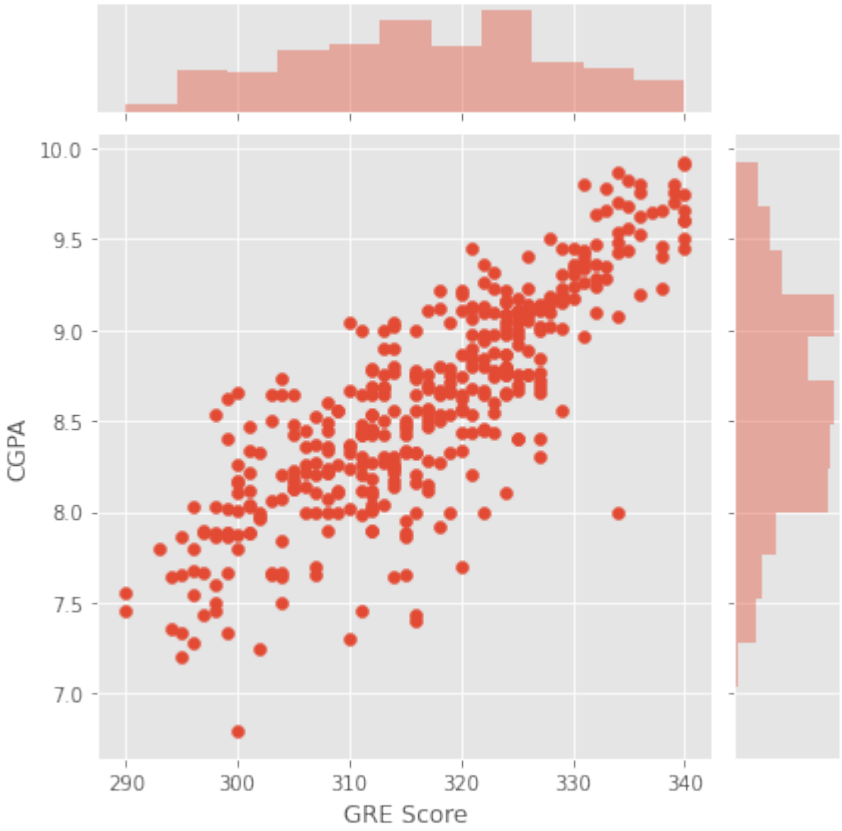
[13]:
```
f,ax = plt.subplots(figsize=(18, 18))
sns.heatmap(data.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```

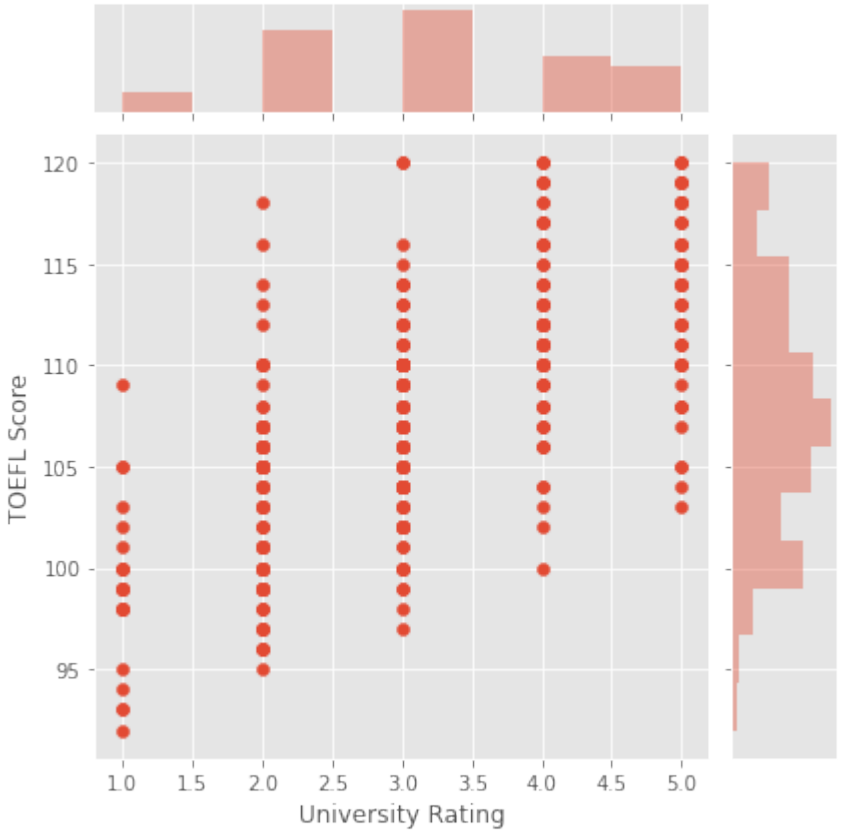


[24]:
```
sns.jointplot(x='GRE Score', y='CGPA', data=data)

sns.jointplot(x='University Rating', y='TOEFL Score', data=data)
```

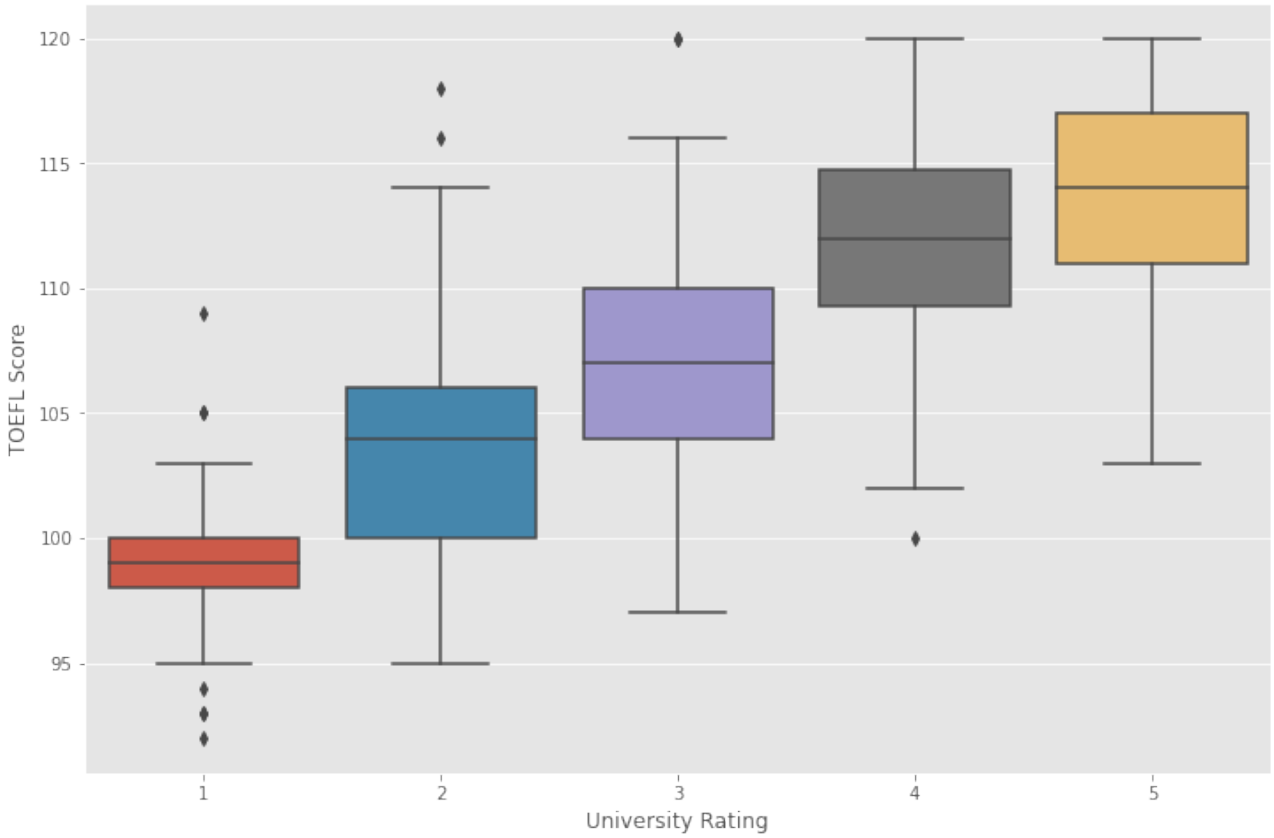
[24]: <seaborn.axisgrid.JointGrid at 0x7fb05a8df320>


CGPA
GRE Score
10.0
9.5
9.0
8.5
8.0
7.5
7.0
290
300
310
320
330
340


TOEFL Score
University Rating
120
115
110
105
100
95
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

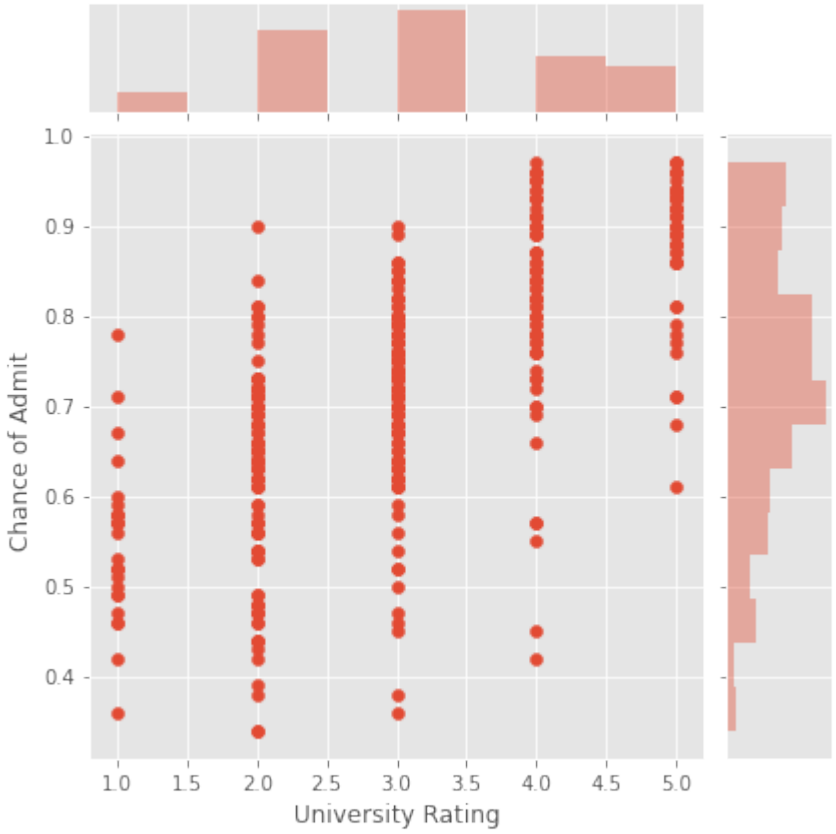
[25]:
```
sns.boxplot(x=data["University Rating"], y=data["TOEFL Score"])
```

[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb0609c7710>



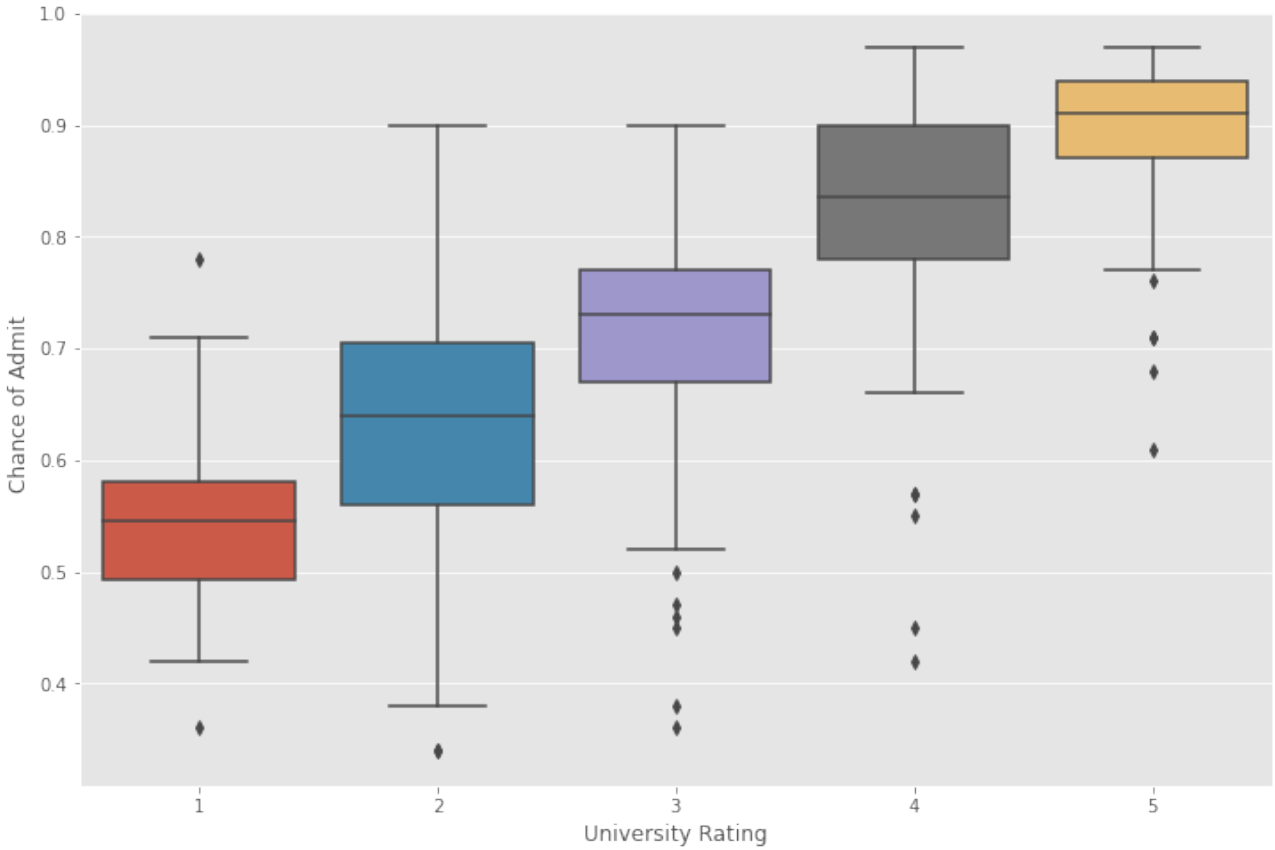
[30]:
```
sns.jointplot(x='University Rating', y='Chance of Admit ', data=data)
```

[30]: <seaborn.axisgrid.JointGrid at 0x7fb05b109a90>



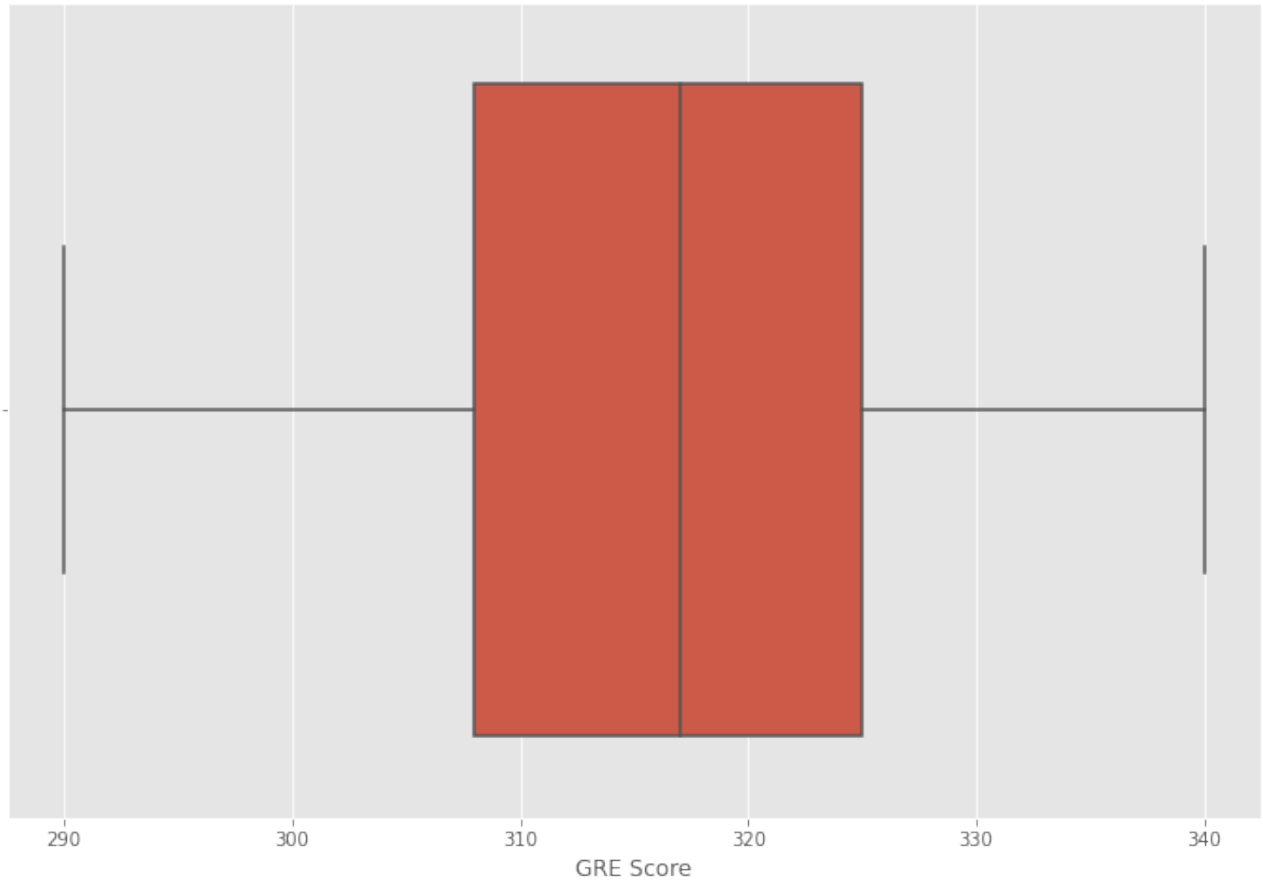
[32]:
```
sns.boxplot(x=data["University Rating"], y=data["Chance of Admit "])
```

[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05a36c278>



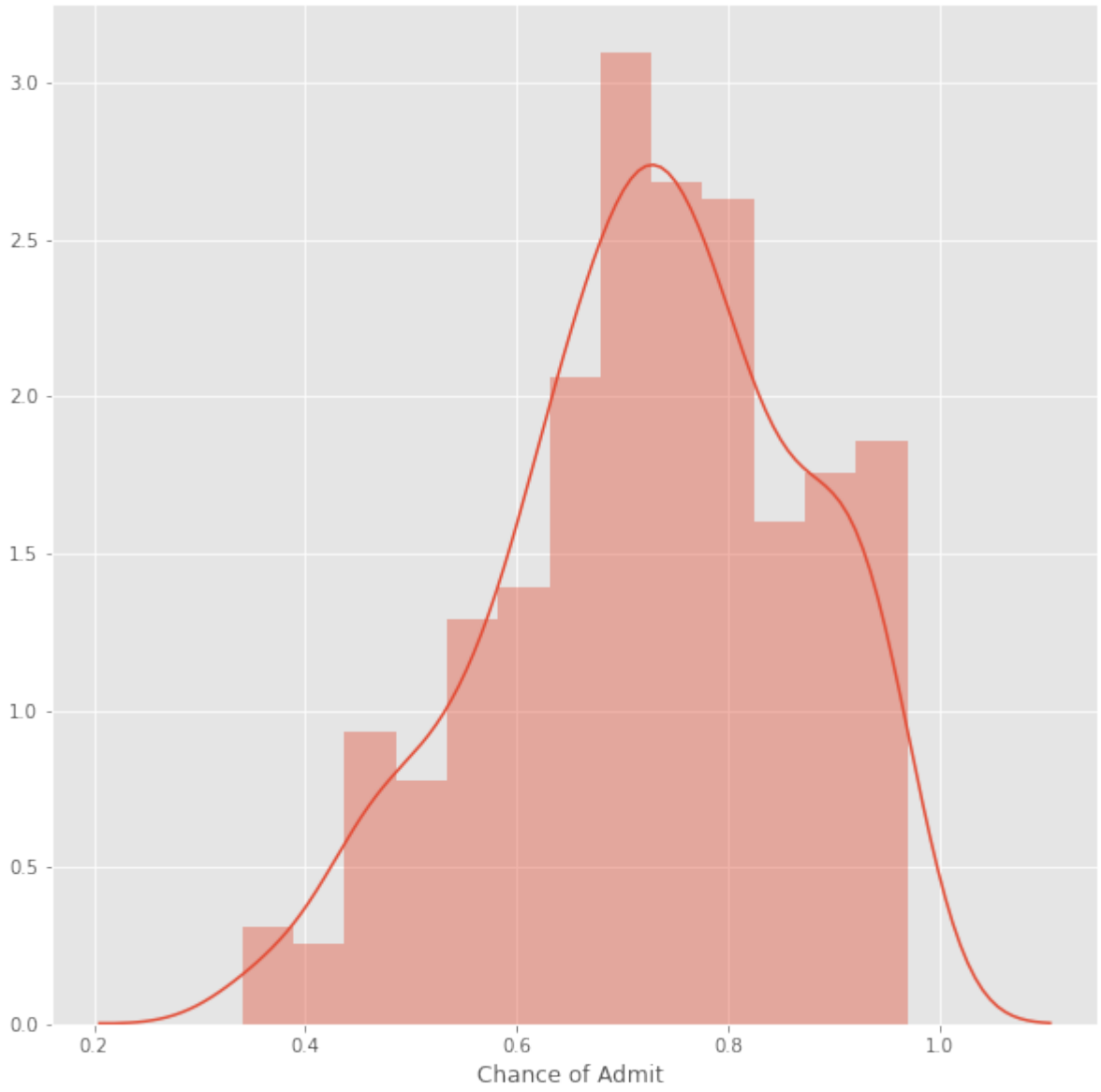
[23]:
```
sns.boxplot(x=data["GRE Score"])
```

[23]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05a976ac8>



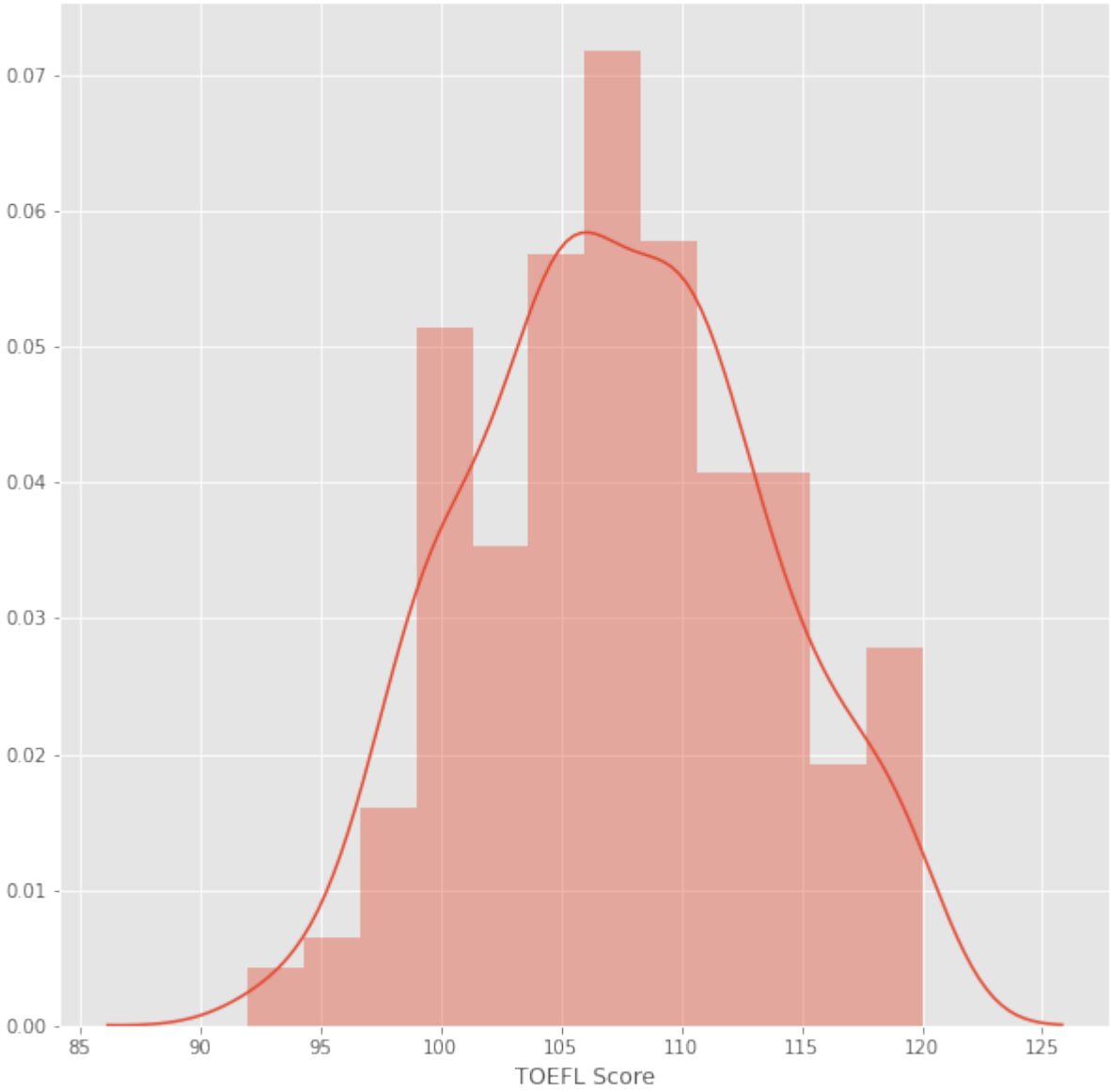
[33]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Chance of Admit '])
```

[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05a209588>



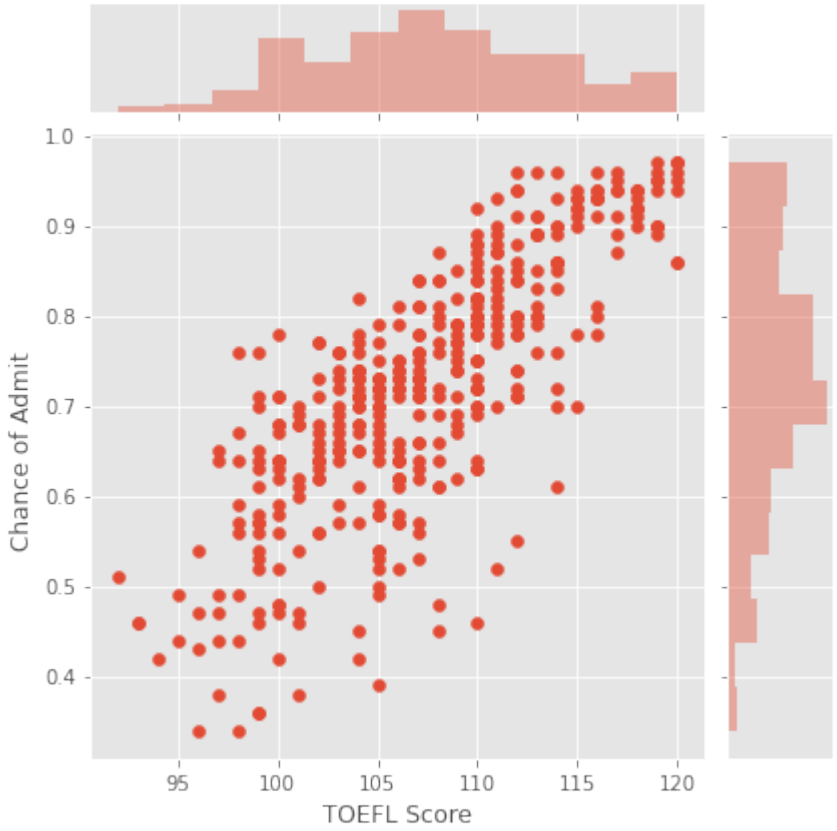
[34]:
```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['TOEFL Score'])
```

[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05a205828>



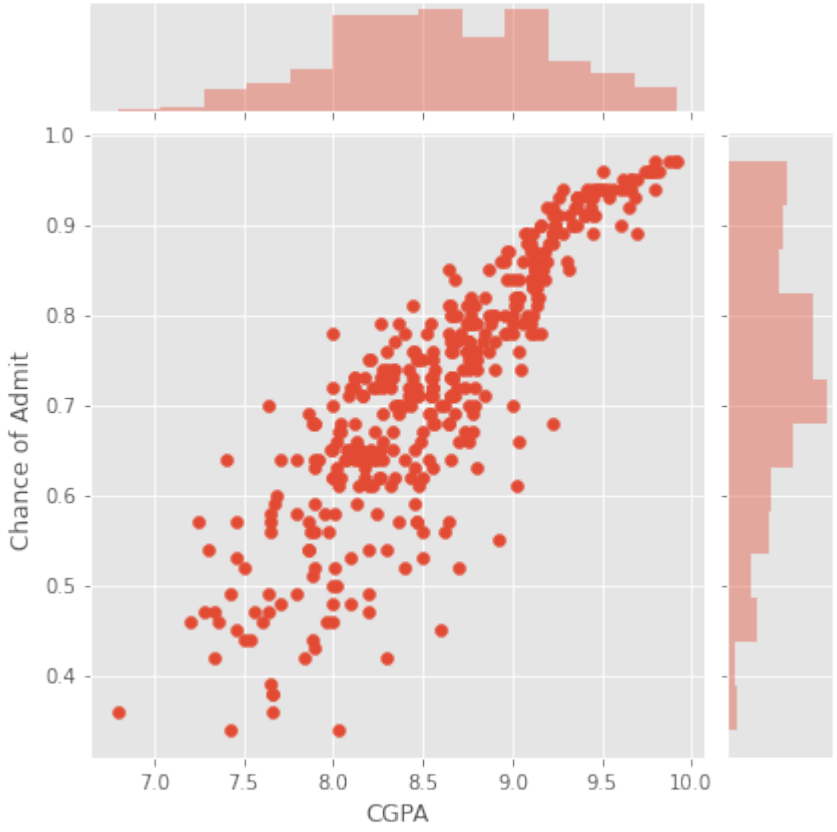
[39]:
```
sns.jointplot(x='TOEFL Score', y='Chance of Admit ', data=data)
```

[39]: <seaborn.axisgrid.JointGrid at 0x7fb05a50cc18>



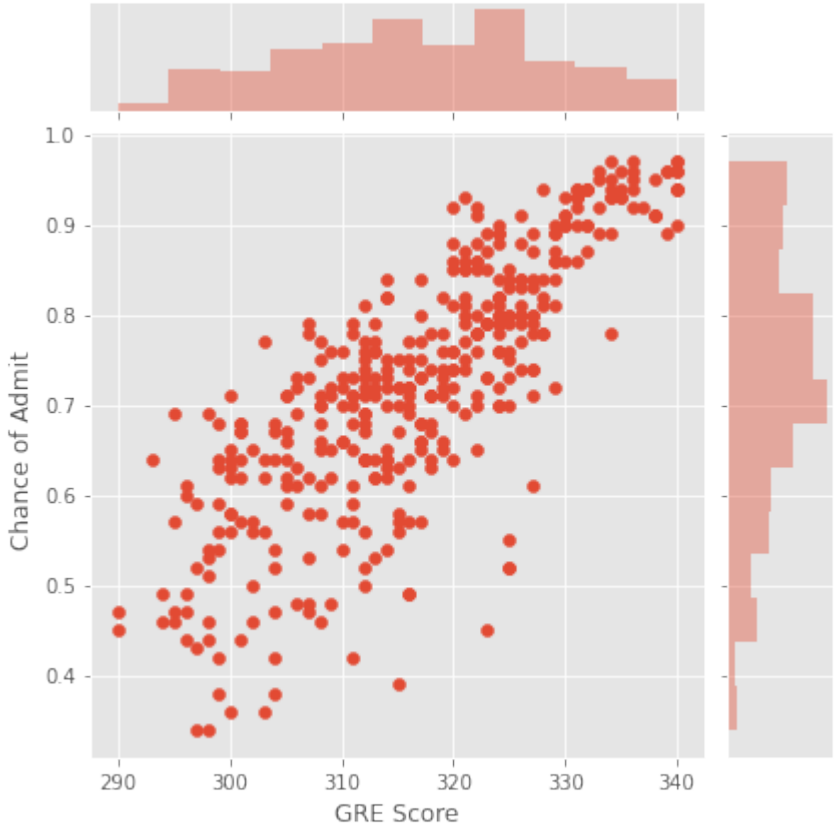
[40]:
```
sns.jointplot(x='CGPA', y='Chance of Admit ', data=data)
```

[40]: <seaborn.axisgrid.JointGrid at 0x7fb05b07d668>



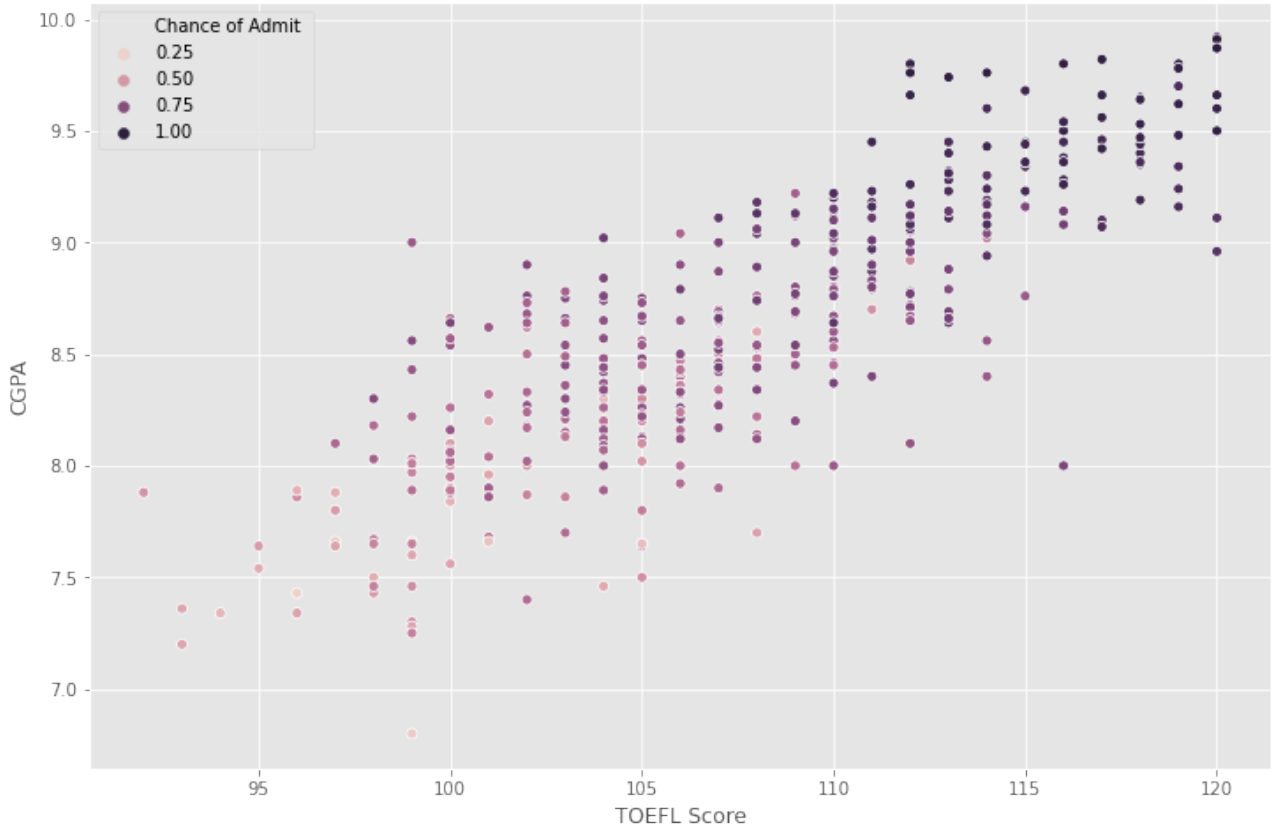
[41]:
```
sns.jointplot(x='GRE Score', y='Chance of Admit ', data=data)
```

[41]: <seaborn.axisgrid.JointGrid at 0x7fb05a685ba8>



[43]:
```
sns.scatterplot(x='TOEFL Score', y='CGPA', data=data, hue='Chance of Admit ')
```

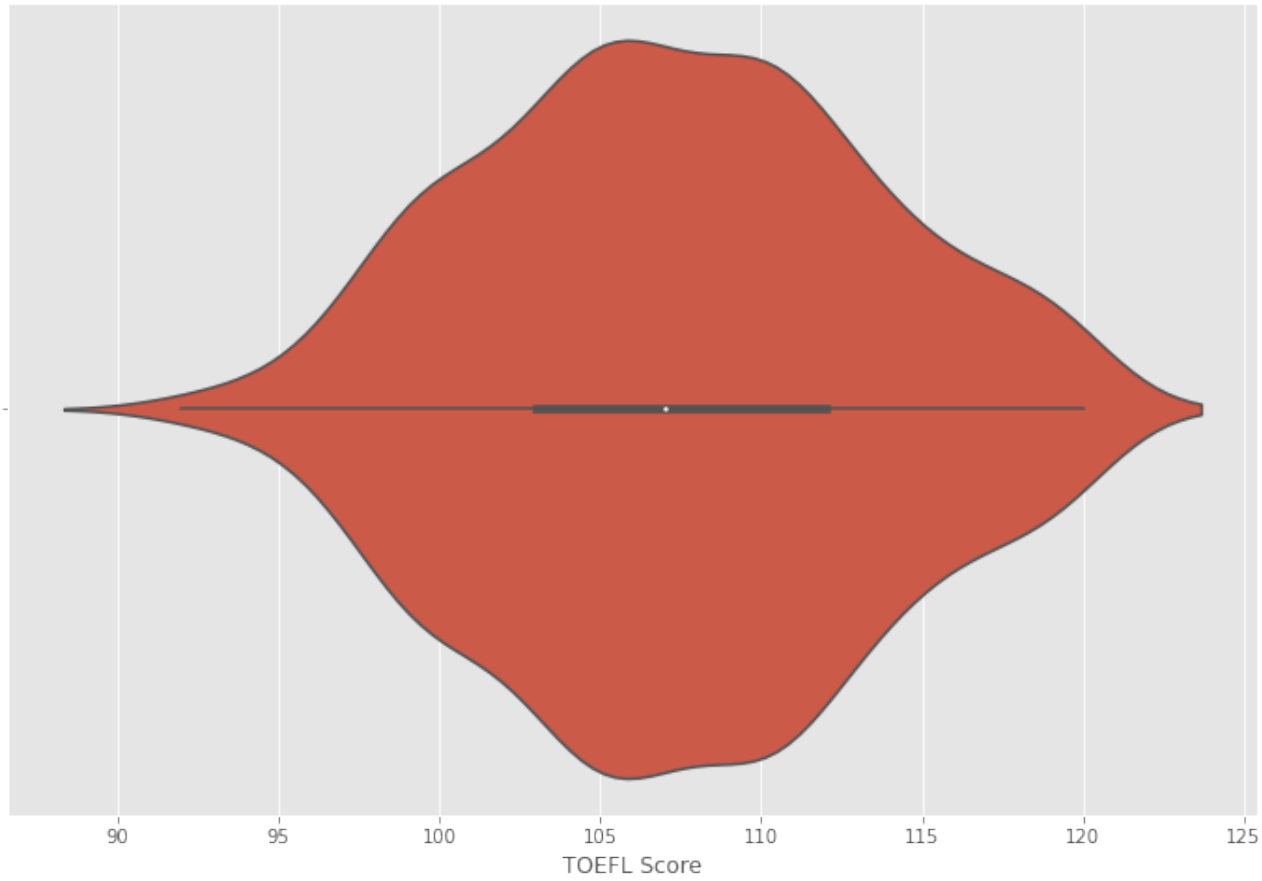
[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05a685b70>



[0]:

[38]:
```
sns.violinplot(x=data["TOEFL Score"])
```

[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb05cbe5d30>



В данном датасете целевой признак - Chance of Admit. Видно, что на него влияют другие признаки - University Rating, TOEFL Score, GRE Score, CGPA, LOR, SOP. Часть этих признаков имеет довольно большую корреляцию между собой, поэтому много признаков можно удалить без потери точности предсказания целевого признака. Можно построить модель линейной регрессии, она будет достаточно точна

[0]: