Рубежный контроль №1

Аброчнов Егор

Группа ИУ5Ц-83Б

Вариант 29

Задание: Для заданного набора данных постройте основные графики, входящие в этап разведочного анализа данных. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Какие графики Вы построили и почему? Какие выводы о наборе данных Вы можете сделать на основании построенных графиков?

Импортируем библиотеки:

```
import sys
sys.path
import pandas as pd
import numpy as np
np.seterr(divide='ignore', invalid='ignore')
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Загрузим данные:

```
df=pd.read_csv('Admission_Predict.csv')
```

```
df.head()
```

```
   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

   Research  Chance of Admit
```

```
0          1               0.92
1          1               0.76
2          1               0.72
3          1               0.80
4          0               0.65
```

Узнаем типы полей датасета:

```
df.dtypes
```

```
Serial No.             int64
GRE Score              int64
TOEFL Score            int64
University Rating      int64
SOP                  float64
LOR                  float64
CGPA                 float64
Research               int64
Chance of Admit      float64
dtype: object
```

Найдем пропуски:

```
for col_empty in df.columns:
    empty_count = df[df[col_empty].isnull()].shape[0]
    print('{} - {}'.format(col_empty, empty_count))
```

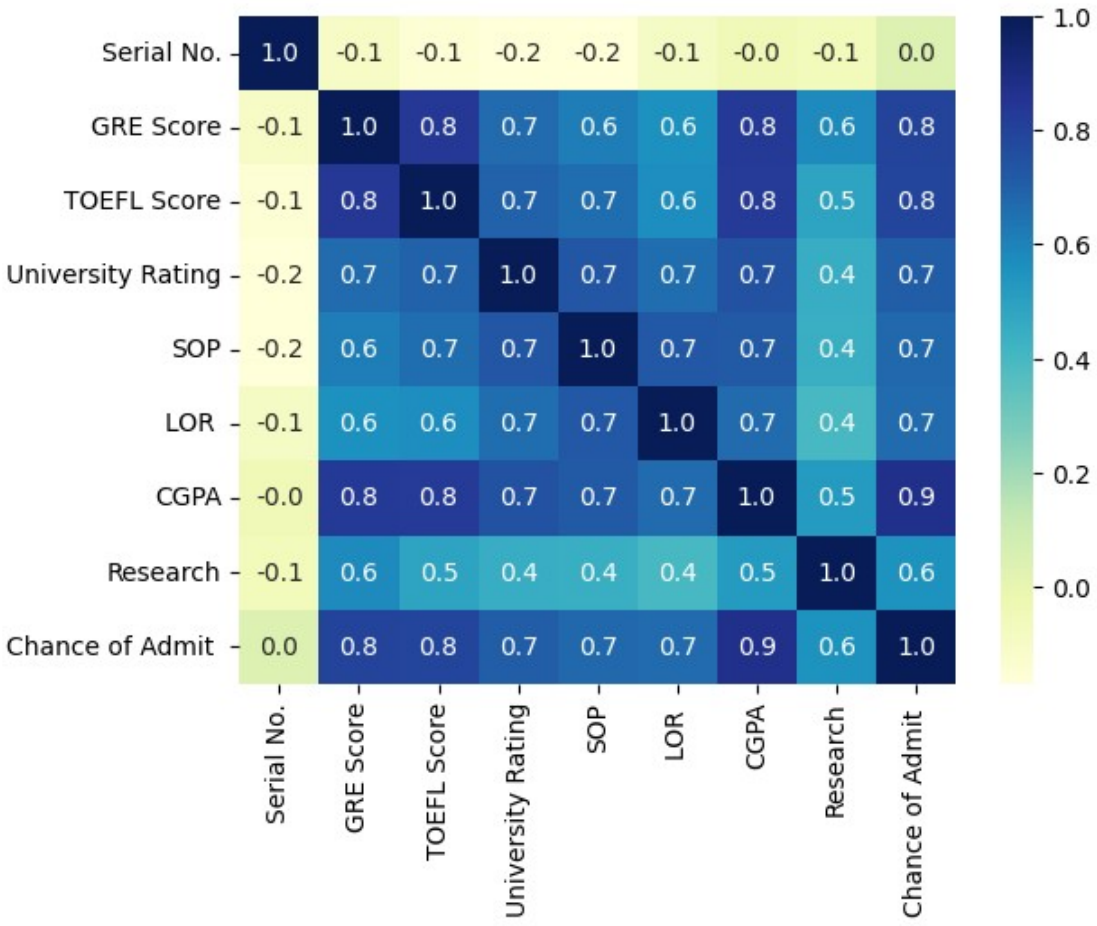
```
Serial No. - 0
GRE Score - 0
TOEFL Score - 0
University Rating - 0
SOP - 0
LOR  - 0
CGPA - 0
Research - 0
Chance of Admit  - 0
```

Пропусков не обнаружено

Построим корреляционную матрицу:

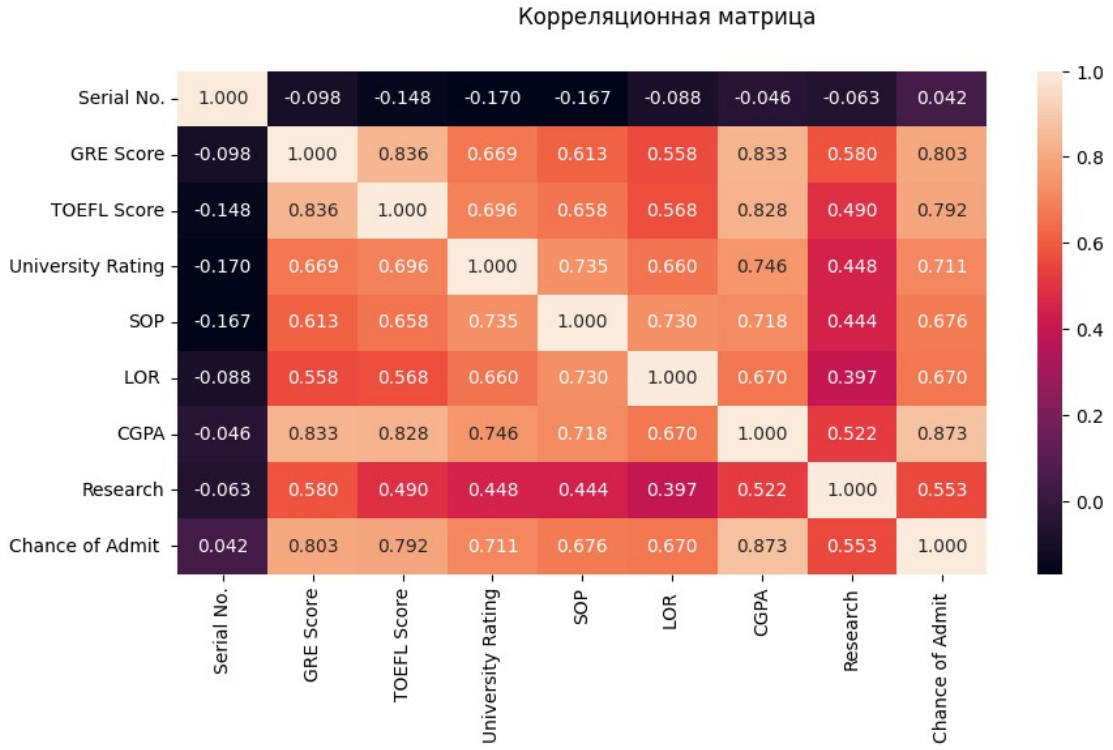
```
sns.heatmap(df.corr(), cmap='YlGnBu', annot=True, fmt='.1f')
```

```
<Axes: >
```




```
fig, ax = plt.subplots(1, 1, sharex='col', sharey='row',
figsize=(10,5))
fig.suptitle('Корреляционная матрица')
sns.heatmap(df.corr(), ax=ax, annot=True, fmt='.3f')
```

```
<Axes: >
```



Диграмма рассеяния

```
df.describe()
```

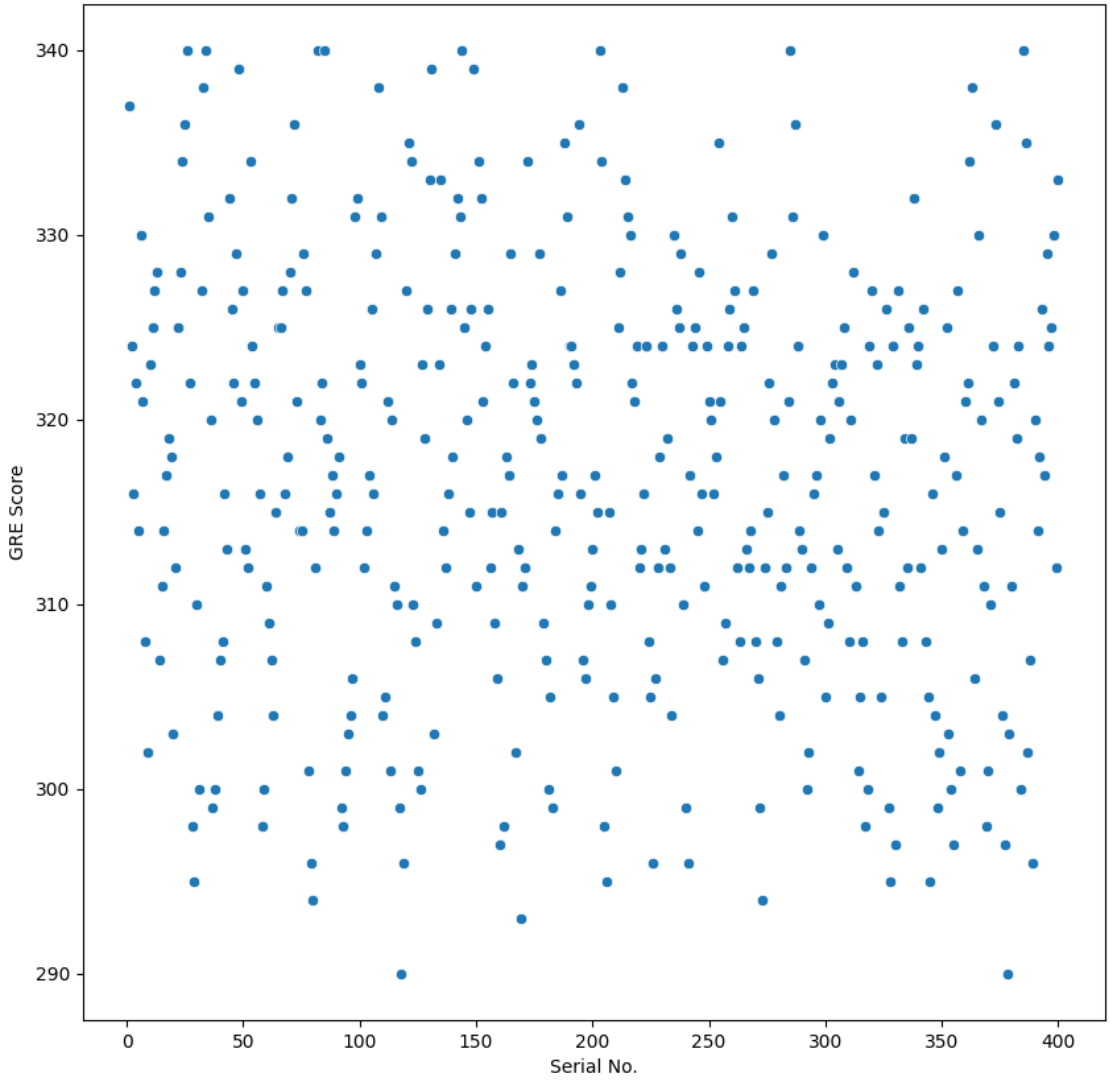
```
       Serial No.   GRE Score  TOEFL Score  University Rating
SOP
count  400.000000  400.000000   400.000000         400.000000
400.000000  \
mean   200.500000  316.807500   107.410000           3.087500
3.400000
std    115.614301   11.473646     6.069514           1.143728
1.006869
min      1.000000  290.000000    92.000000           1.000000
1.000000
25%    100.750000  308.000000   103.000000           2.000000
2.500000
50%    200.500000  317.000000   107.000000           3.000000
3.500000
75%    300.250000  325.000000   112.000000           4.000000
4.000000
max    400.000000  340.000000   120.000000           5.000000
5.000000

              LOR         CGPA    Research  Chance of Admit
count  400.000000  400.000000  400.000000       400.000000
mean     3.452500    8.598925    0.547500         0.724350
std      0.898478    0.596317    0.498362         0.142609
min      1.000000    6.800000    0.000000         0.340000
```

```
25%       3.000000    8.170000    0.000000          0.640000
50%       3.500000    8.610000    1.000000          0.730000
75%       4.000000    9.062500    1.000000          0.830000
max       5.000000    9.920000    1.000000          0.970000
```

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Serial No.', y='GRE Score', data=df)
```

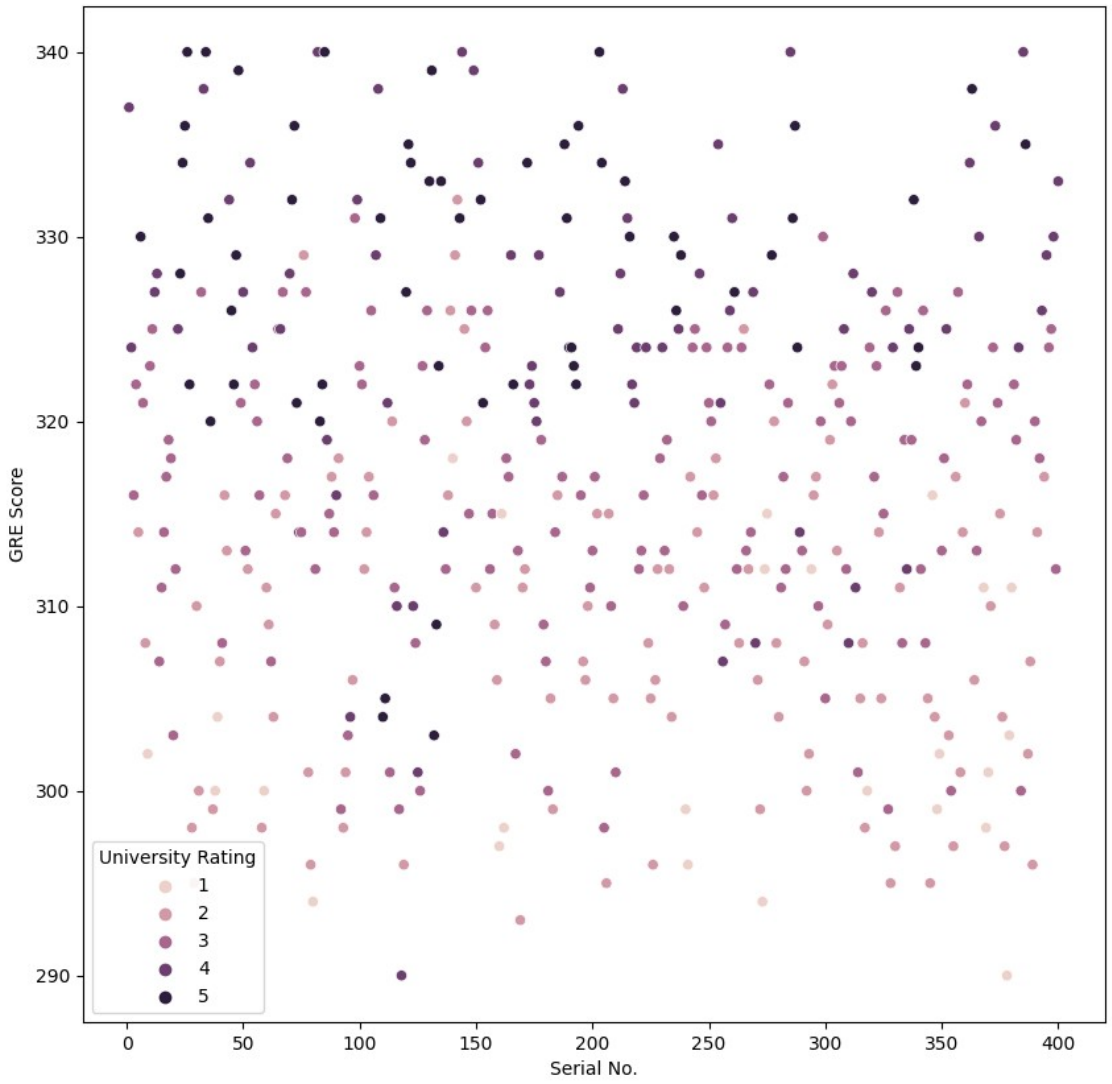
```
<Axes: xlabel='Serial No.', ylabel='GRE Score'>
```



Отсюда видно, что основная часть колледжей содержит очки GRE в промежутке 300-330

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Serial No.', y='GRE Score', data=df,
hue='University Rating')
```

```
<Axes: xlabel='Serial No.', ylabel='GRE Score'>
```

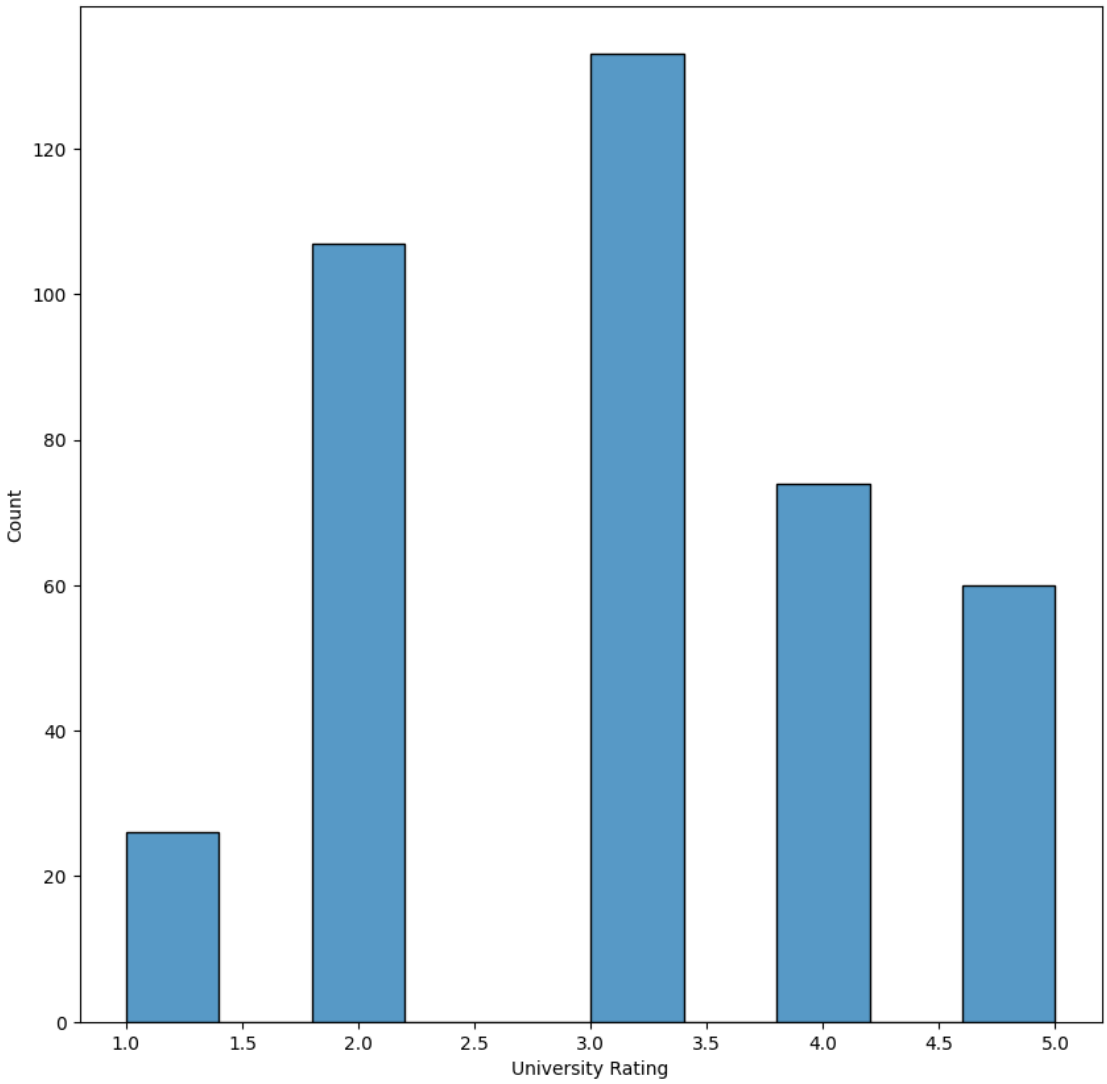


Основная часть высокооцененных колледжей имеет больше 320 очков GRE, но видно, что встречаются высокооцененные с очками ниже 310.

Гистограмма

```
fig, ax = plt.subplots(figsize=(10,10))
sns.histplot(df['University Rating'])
```

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

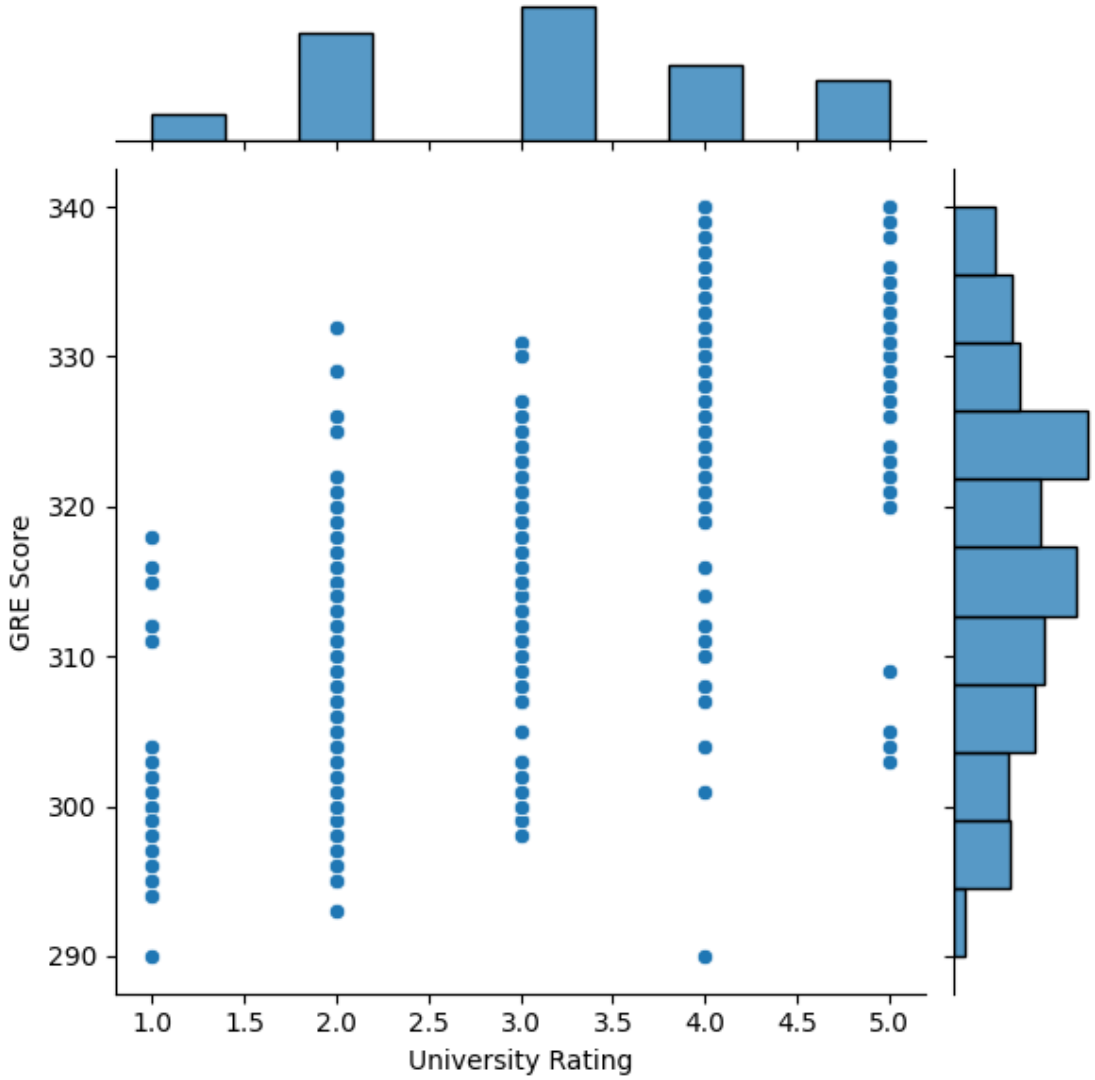


Больше всего колледжей оценено на 3.

Jointplot

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

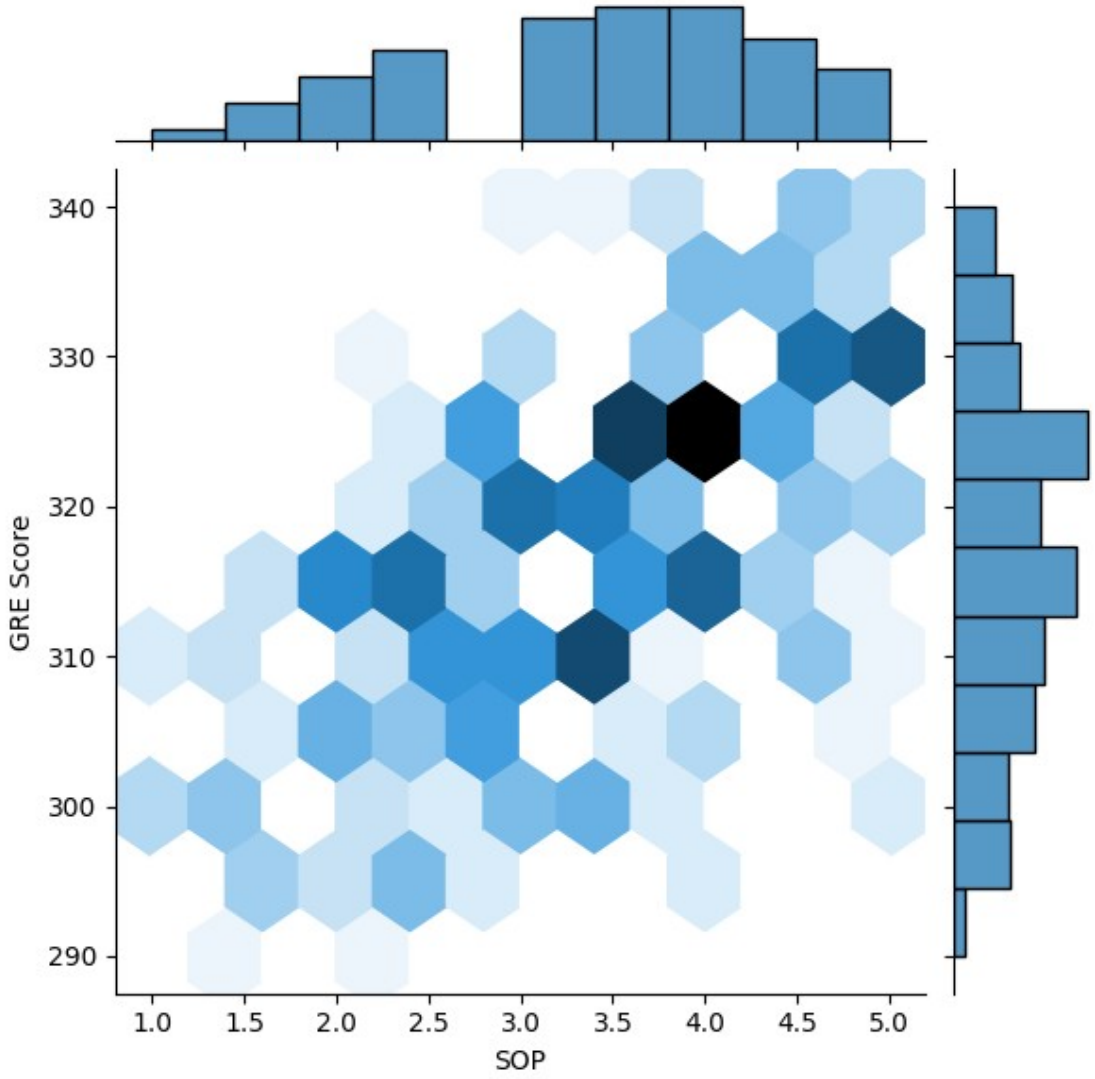
```
<seaborn.axisgrid.JointGrid at 0x2c8323e1330>
```



Комбинация диаграммы рассеивания и гистрограммы.

```
sns.jointplot(x='SOP', y='GRE Score', data=df, kind='hex')
```

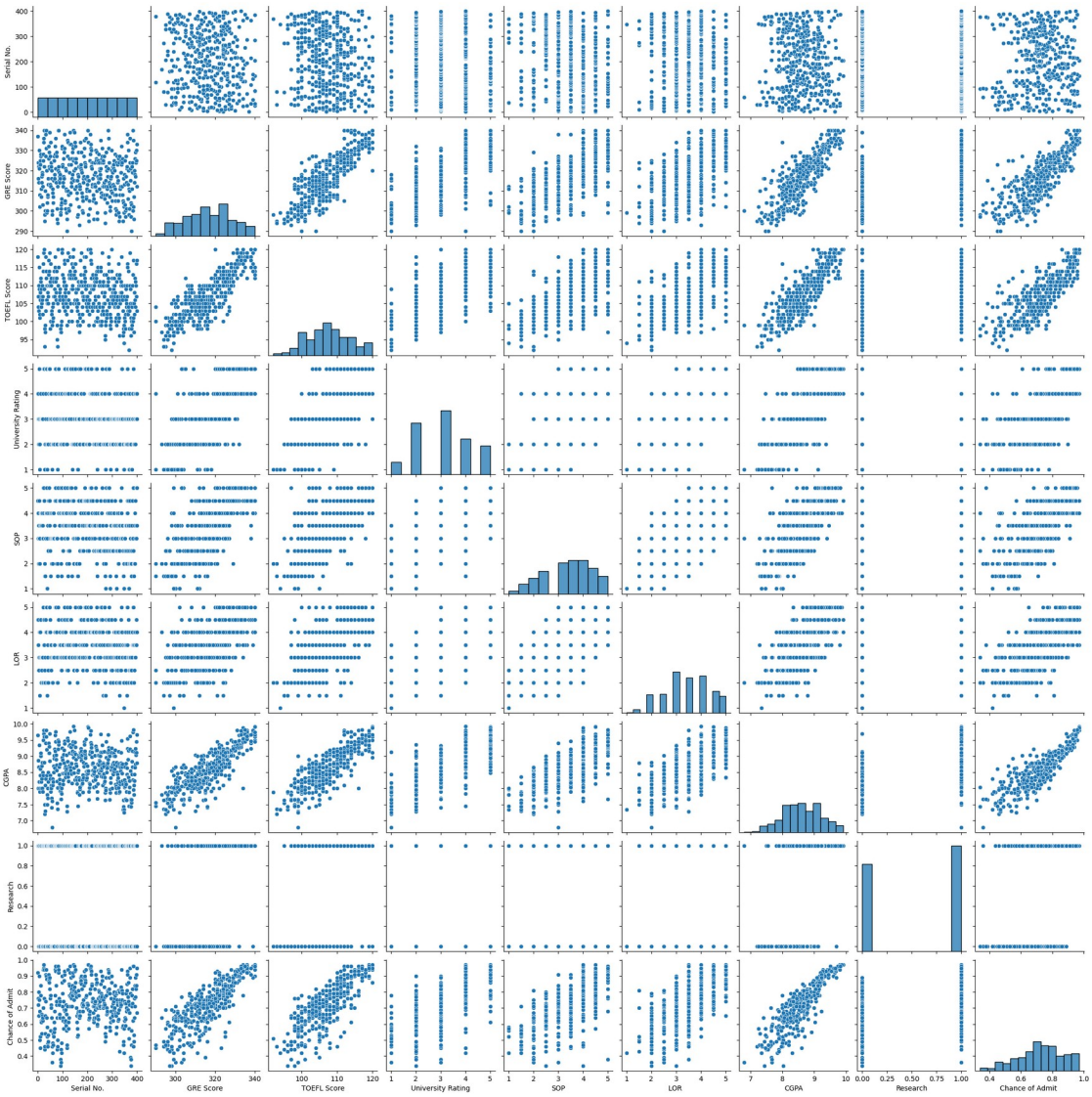
```
<seaborn.axisgrid.JointGrid at 0x2c832729390>
```



Парные диаграммы

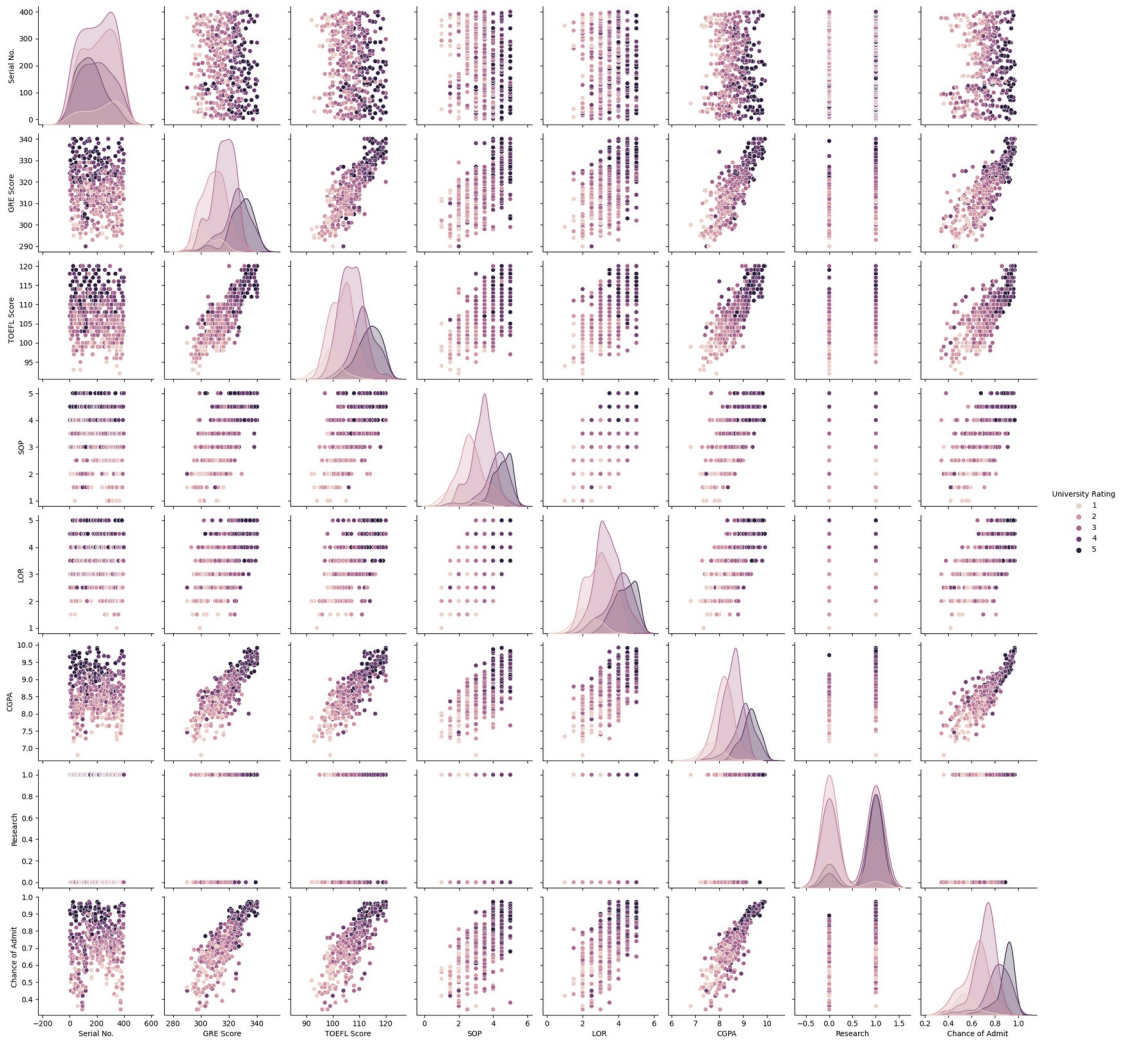
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x2c832811120>
```




```
sns.pairplot(df, hue='University Rating')
```

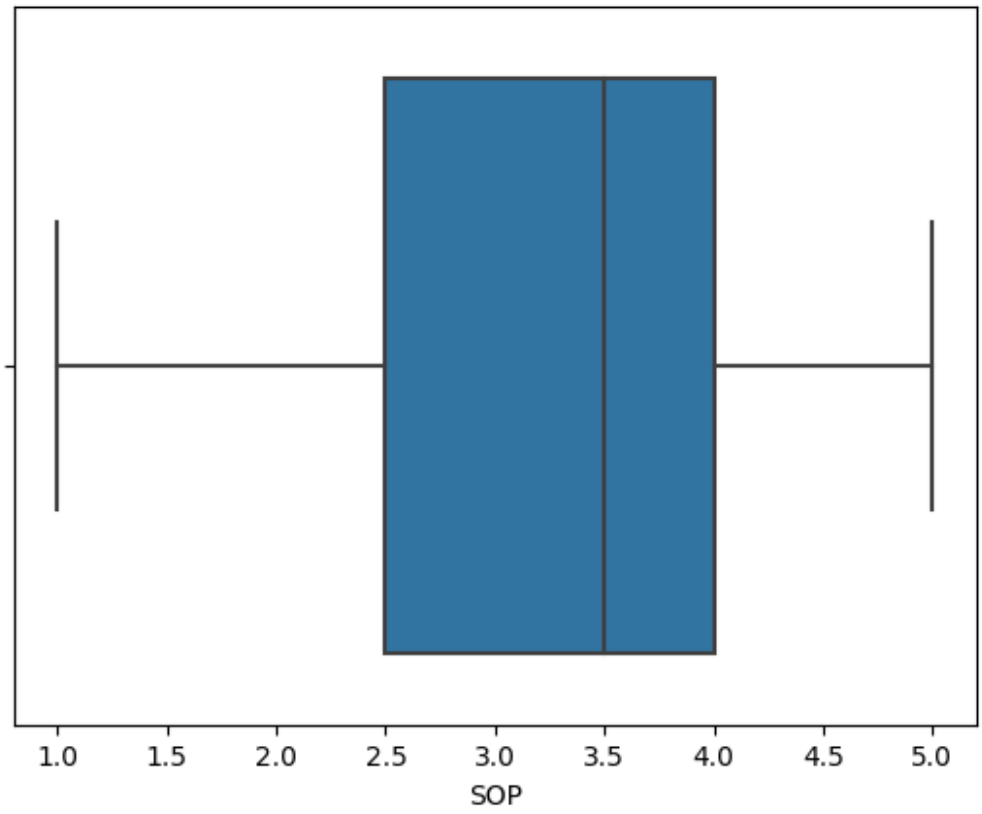
```
<seaborn.axisgrid.PairGrid at 0x2c8370d54e0>
```



Ящик с усами

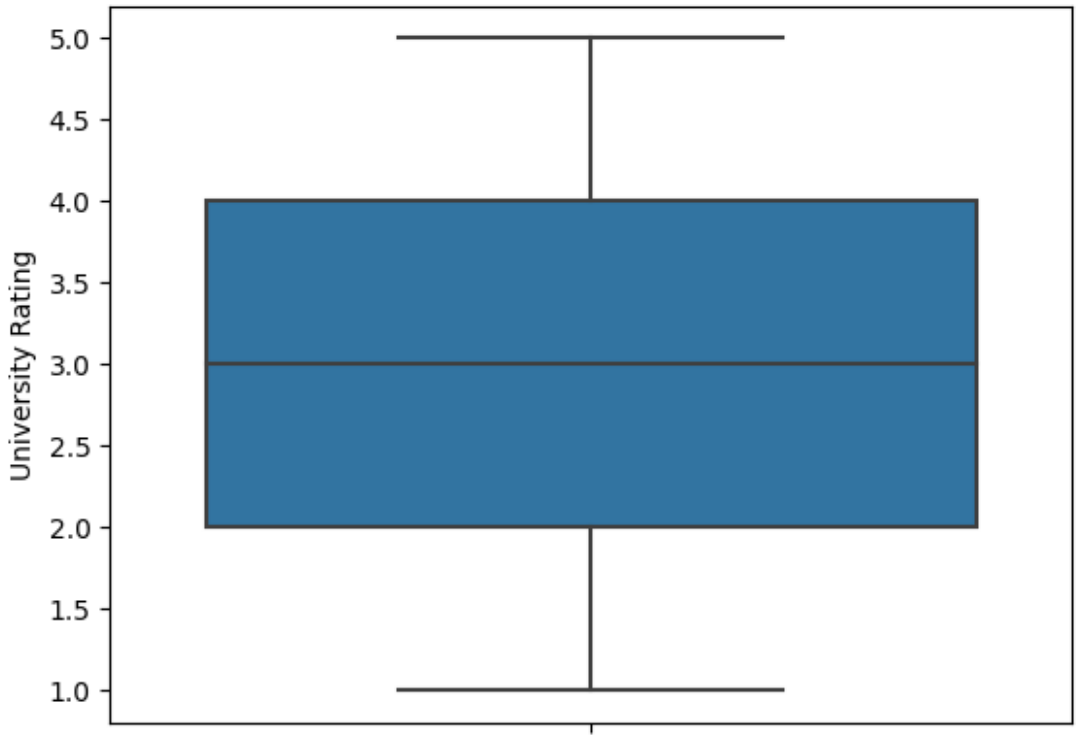
```
sns.boxplot(x=df['SOP'])
```

```
<Axes: xlabel='SOP'>
```




```
sns.boxplot(y=df['University Rating'])
```

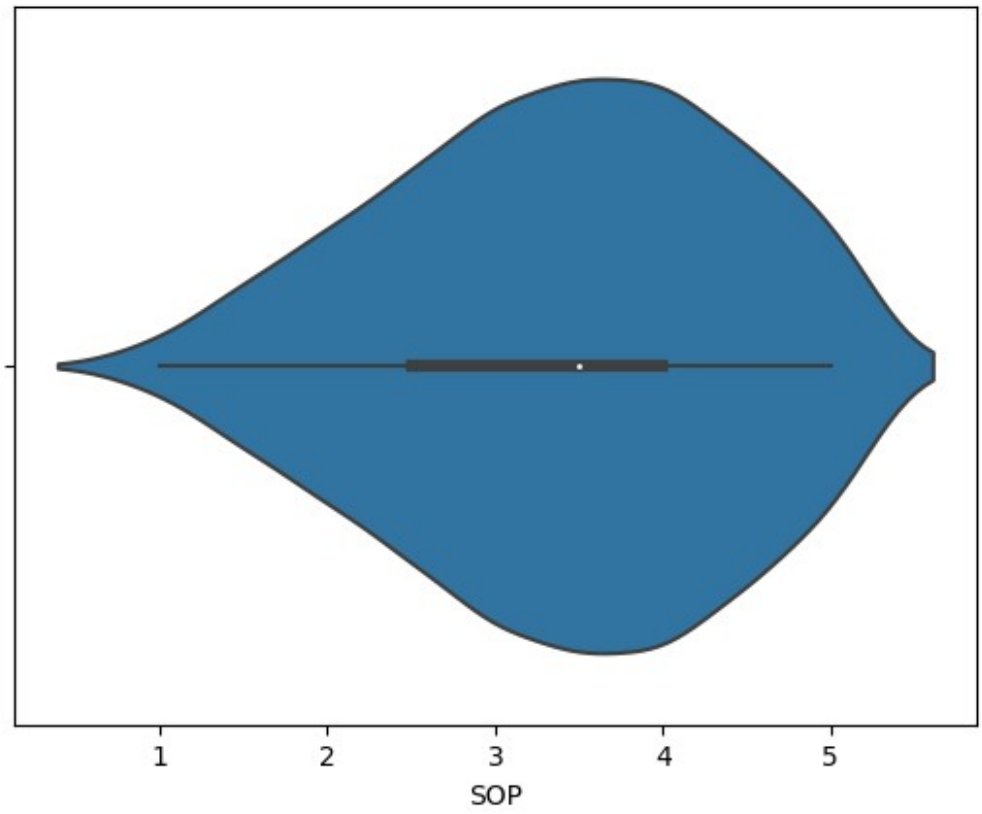
```
<Axes: ylabel='University Rating'>
```



Скрипачная диаграмма

```
sns.violinplot(x=df['SOP'])
```

```
<Axes: xlabel='SOP'>
```

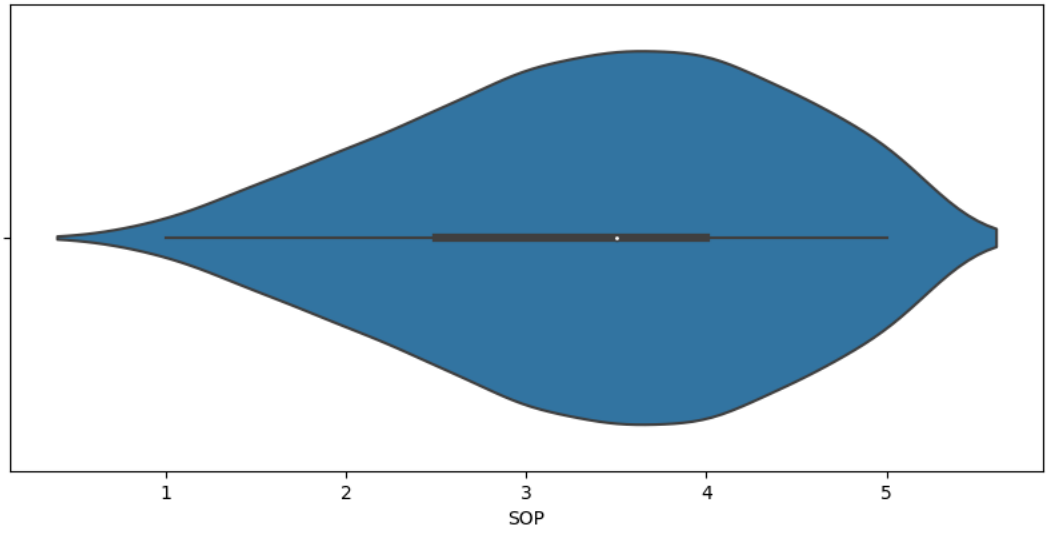


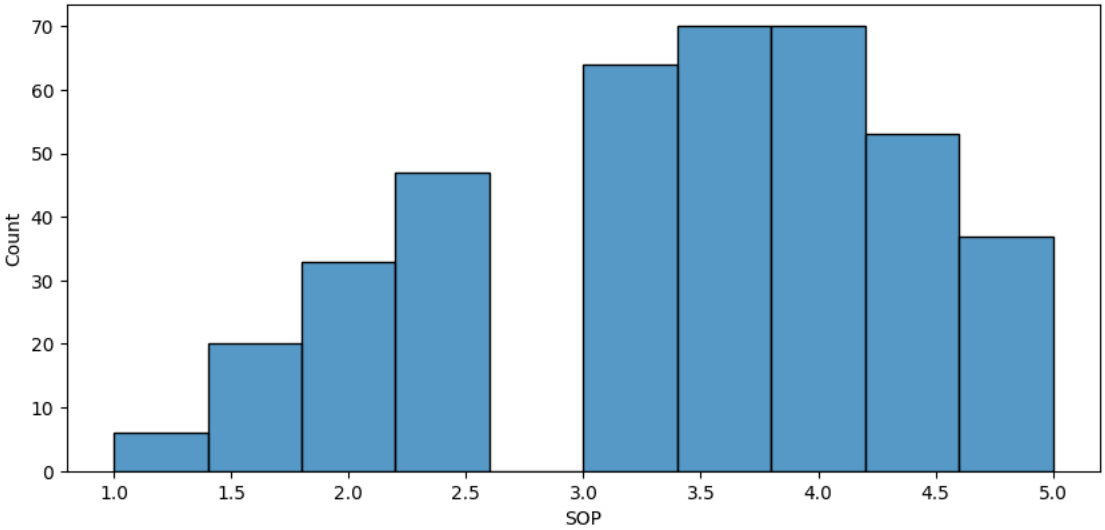
Скрипачная диаграмма показывает распределение плотности SOP очков.

Сравним с гистограммой

```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=df['SOP'])
sns.histplot(df['SOP'])
```

```
<Axes: xlabel='SOP', ylabel='Count'>
```





Из гистрограммы видно, что скрипачная показывает распределение плотности SOP очков.