

PK

Импорт библиотек

```
In [1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
import warnings
from sklearn import datasets
from sklearn.datasets import load_wine
from sklearn import linear_model
from sklearn.cluster import KMeans
from sklearn import metrics
from pandas import DataFrame
%pylab inline

Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: data = pd.read_csv('Admission_Predict.csv', sep = ',' )
```

```
In [3]: data.head()
```

Out[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

```
In [4]: data.info()

<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
```

```
In [5]: data.describe()
```

Out[5]:

	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

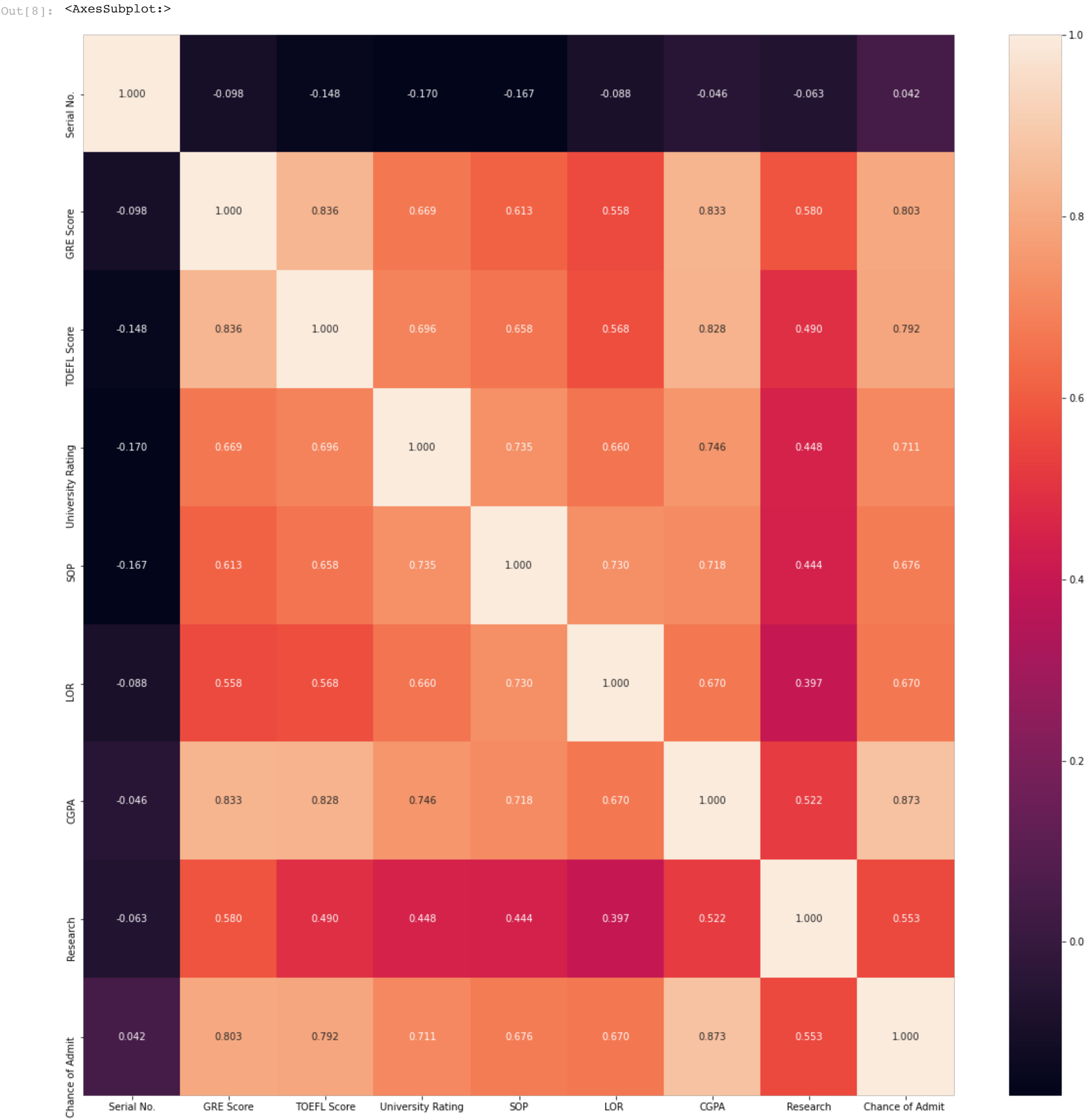
```
In [6]: ## Корр. анализ
corr_matrix = data.corr()
```

```
In [7]: corr_matrix['CGPA']
```

Out[7]:

```
Serial No.      -0.045608
GRE Score       0.833060
TOEFL Score     0.828417
University Rating 0.746479
SOP             0.718144
LOR            -0.670211
CGPA            1.000000
Research        0.521654
Chance of Admit  0.873289
Name: CGPA, dtype: float64
```

```
In [8]: plt.figure(figsize=(20,20))
sns.heatmap(corr_matrix, annot=True, fmt='.3f')
```



```
In [9]: # Увеличенные диаграммы рассеяния
sns.jointplot(x = "CGPA", y = "SOP", kind="scatter", data = data)
```

