

# Рубежный контроль 1

Выполнил Гусев Сергей ИУ5Ц-82Б

## Вариант №2

**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?

```
In [5]:

import pandas as pd
import sklearn
import numpy as np
from sklearn.datasets import load_wine
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [6]:

wine = load_wine()
```

```
In [7]:

print(wine.data.shape)

(178, 13)
```

```
In [8]:

data=pd.DataFrame(wine.data)
```

```
In [9]:

data.head()
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0

Проверим датасет на наличие пропусков

```
In [10]:

data.isnull().sum()
```

Out[10]:

```
0      0
1      0
2      0
3      0
4      0
5      0
6      0
7      0
8      0
9      0
10     0
11     0
12     0
dtype: int64
```

Основные статистические характеристики набора данных

```
In [11]:

data.describe()
```

Out[11]:

	0	1	2	3	4	5	6	7	8	9	10	11	12
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.058090	0.957449	2.611685	746.893258
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.318286	0.228572	0.709990	314.907474
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.280000	0.480000	1.270000	278.000000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.220000	0.782500	1.937500	500.500000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.690000	0.965000	2.780000	673.500000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.200000	1.120000	3.170000	985.000000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.000000	1.710000	4.000000	1680.000000

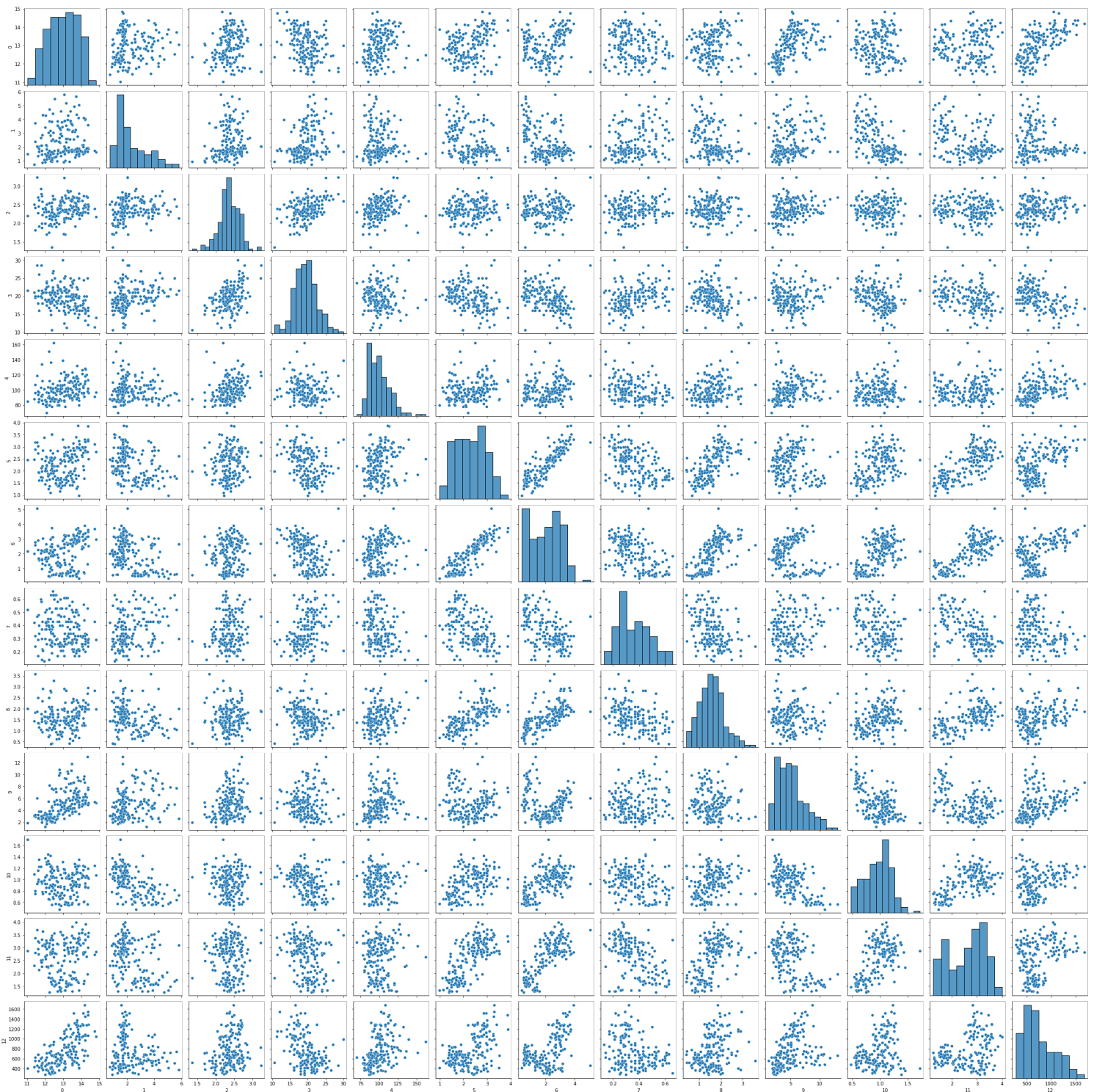
Построим парные диаграммы для всего датасета

```
In [14]:
```

```
sns.pairplot(data)
```

```
Out[14]:
```

```
<seaborn.axisgrid.PairGrid at 0x2ca2c85ee20>
```



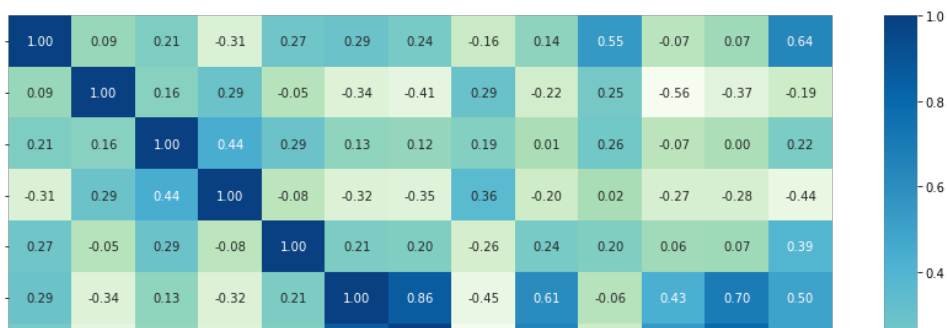
Построим корреляционную диаграмму

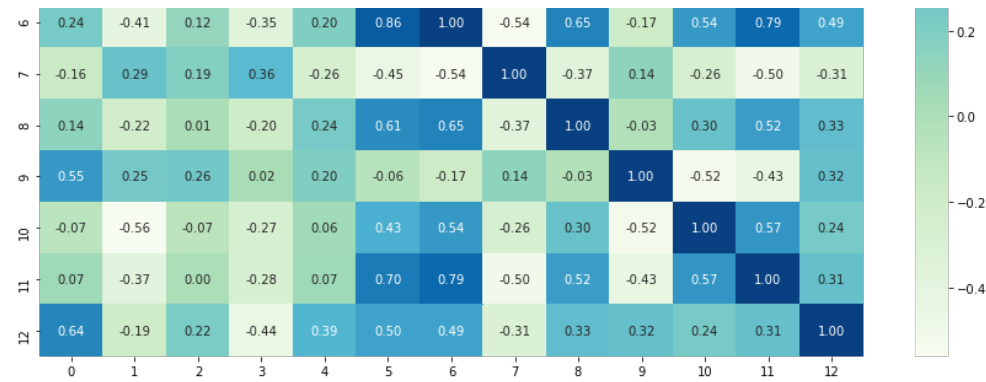
```
In [15]:
```

```
fig, ax = plt.subplots(figsize=(15,10))
sns.heatmap(data.corr(), annot=True, fmt='.2f', cmap='GnBu')
```

```
Out[15]:
```

```
<AxesSubplot:>
```





Выводы: в матрице признаки хорошо коррелируют между собой. Это значит, что на их основании можно будет построить в дальнейшем обучающую модель.

In [ ]: