РТ5-61Б, Забурунов Л. В.

Технологии Машинного Обучения

Лабораторная Работа №1

"Разведочный Анализ Данных"

1. Текстовое описание набора данных

В лабораторной работе используется датасет "Diabetes dataset" из числа "игрушечных" от Scikit-Learn.

Имеются следующие анонимизированные данные пациентов:

1. Возраст (AGE);
2. Пол (SEX);
3. Индекс Массы Тела (BMI);
4. Кровяное давление (BP);
5. Белые кровяные тельца (S1);
6. Липопротеины низкой плотности (S2);
7. Липопротеины высокой плотности (S3);
8. Тиреотропный гормон (S4);
9. Ламотригин (S5);
10. Уровень сахара в крови (S6);
11. (здесь мне необходим ликбез) Численная мера прогрессирования заболевания через год (Y, целевой признак)

Загрузим выбранный датасет

Преобразование в PandasDataFrame уже встроено как одна из опций при получении датасета

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = pd.read_table("https://www4.stat.ncsu.edu/~boos/var.select/diabetes.tab.txt")

data.head()
```

Out[2]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59	2	32.1	101.0	157	93.2	38.0	4.0	4.8598	87	151
1	48	1	21.6	87.0	183	103.2	70.0	3.0	3.8918	69	75
2	72	2	30.5	93.0	156	93.6	41.0	4.0	4.6728	85	141
3	24	1	25.3	84.0	198	131.4	40.0	5.0	4.8903	89	206
4	50	1	23.0	101.0	192	125.4	52.0	4.0	4.2905	80	135

2. Основные характеристики набора данных

In [6]:

```
# Посмотрим на общий размер данных

print(data.shape)
```

```
(442, 11)
```

In [7]:

```
# Посмотрим на список атрибутов и типы их данных

print(data.dtypes)
```

```
AGE      int64
SEX      int64
BMI    float64
BP     float64
S1       int64
S2     float64
S3     float64
S4     float64
S5     float64
S6       int64
Y        int64
dtype: object
```

In [8]:

```
# Проверяем наличие пропусков

for attribute in data.columns:
    #print(data[attribute])
    #print(data[attribute].isnull())
    #print(data[data[attribute].isnull()])
    #print(data[data[attribute].isnull()].shape)
    #print(data[data[attribute].isnull()].shape[0])

    null_counter = data[data[attribute].isnull()].shape[0]
    if null_counter > 0:
        print("Встречено {} пропусков в столбце {}. \nНабор данных не рекомендован к использованию".format(nul
        break
else:
    print("Пропуски не обнаружены")
```

```
Пропуски не обнаружены
```

In [9]:

```
# Посмотрим на основные статистические показатели данных

data.describe()
```

Out[9]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
count	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000
mean	48.518100	1.468326	26.375792	94.647014	189.140271	115.439140	49.788462	4.070249	4.641411	91.260181	152.133484
std	13.109028	0.499561	4.418122	13.831283	34.608052	30.413081	12.934202	1.290450	0.522391	11.496335	77.093005
min	19.000000	1.000000	18.000000	62.000000	97.000000	41.600000	22.000000	2.000000	3.258100	58.000000	25.000000
25%	38.250000	1.000000	23.200000	84.000000	164.250000	96.050000	40.250000	3.000000	4.276700	83.250000	87.000000
50%	50.000000	1.000000	25.700000	93.000000	186.000000	113.000000	48.000000	4.000000	4.620050	91.000000	140.500000
75%	59.000000	2.000000	29.275000	105.000000	209.750000	134.500000	57.750000	5.000000	4.997200	98.000000	211.500000
max	79.000000	2.000000	42.200000	133.000000	301.000000	242.400000	99.000000	9.090000	6.107000	124.000000	346.000000

In [10]:

```
# Посмотрим на множество значений целевого признака

data['Y'].unique()
```

Out[10]:

```
array([151,  75, 141, 206, 135,  97, 138,  63, 110, 310, 101,  69, 179,
       185, 118, 171, 166, 144, 168,  68,  49, 245, 184, 202, 137,  85,
       131, 283, 129,  59, 341,  87,  65, 102, 265, 276, 252,  90, 100,
        55,  61,  92, 259,  53, 190, 142, 155, 225, 104, 182, 128,  52,
        37, 170,  71, 163, 150, 160, 178,  48, 270, 111,  42, 200, 113,
       143,  51, 210, 134,  98, 164,  96, 162, 279,  83, 302, 198,  95,
       232,  81, 246, 297, 258, 229, 275, 281, 173, 180,  84, 121, 161,
        99, 109, 115, 268, 274, 158, 107, 103, 272, 280, 336, 317, 235,
        60, 174, 126, 288,  88, 292, 197, 186,  25, 195, 217, 172, 214,
        70, 220, 152,  47,  74, 295, 127, 237,  64,  79,  91, 116,  86,
       122,  72,  39, 196, 222, 277,  77, 191,  73, 263, 248, 296,  78,
        93, 208, 108, 154, 124,  67, 257, 262, 177, 187, 125, 215, 303,
       243, 153, 346,  89,  50, 308, 145,  45, 264, 241,  66,  94, 230,
       181, 156, 233, 219,  80, 332,  31, 236, 253,  44, 114, 147, 242,
       249, 192, 244, 199, 306, 216, 139, 148,  54, 221, 311, 321,  58,
       123, 167, 140,  40, 132, 201, 273,  43, 175, 293, 189, 209, 136,
       261, 146, 212, 120, 183,  57], dtype=int64)
```

Видим, что целевой признак принимает различные дискретные значения

3. Визуальное исследование набора данных

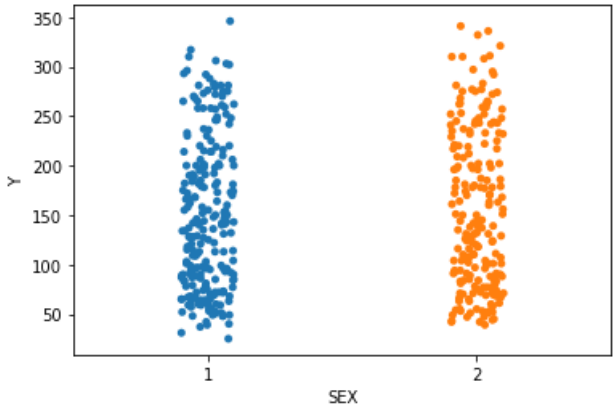
Половая статистика распределения показателя болезни

In [11]:

```
sns.stripplot(x='SEX', y='Y', data=data)
```

Out[11]:

```
<AxesSubplot:xlabel='SEX', ylabel='Y'>
```



Здесь сложно проследить какие-либо отличия по "уровню болезни" в зависимости от пола пациента.

Зависимость прогрессии заболевания от уровня сахара в крови

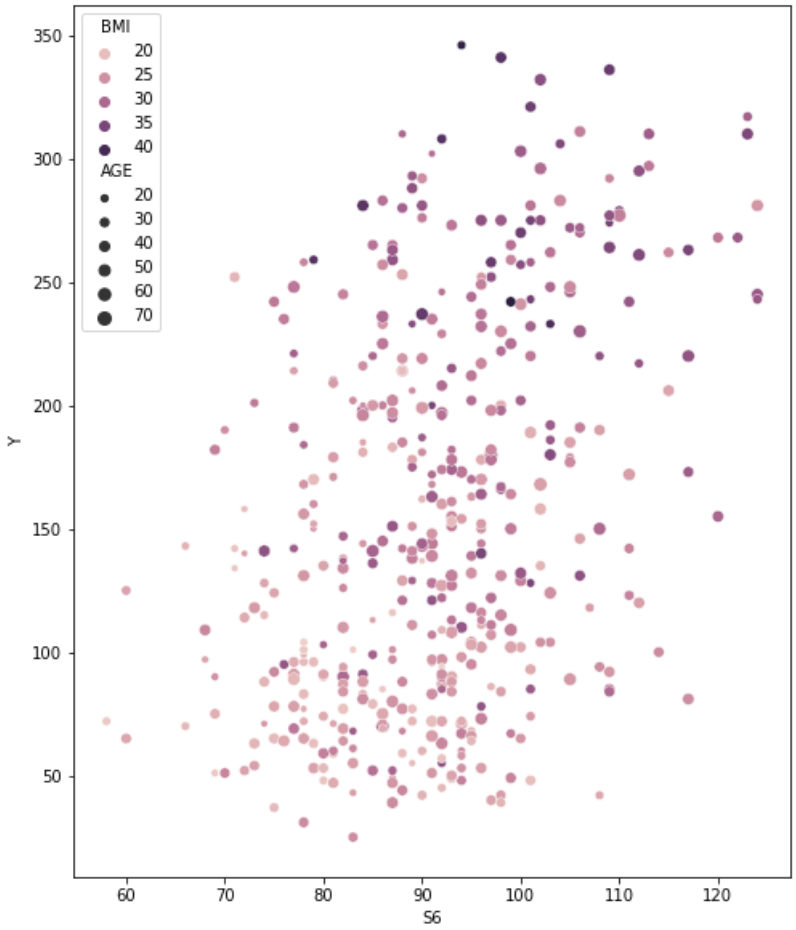
Точки графика также имеют оттенок в зависимости индекса массы тела и размер в зависимости от возраста пациента.

In [35]:

```
fig, ax = plt.subplots(figsize=(8, 10))
sns.scatterplot(ax=ax, x='S6', y='Y', hue='BMI', size="AGE", data=data)
```

Out[35]:

```
<AxesSubplot:xlabel='S6', ylabel='Y'>
```



Несмотря на значительное рассеивание точек, можно выделить линейную зависимость двух параметров (предполагаемый коэффициент корреляции до рассмотрения корр. матрицы - 0.3 или 0.4). Также можно отметить, что по мере продвижения по прямой, характеризующей эту зависимость, точки становятся темнее и больше; то есть, заболевание гораздо серьёзнее у людей, входящих в старшую возрастную группу и имеющих избыточный вес.

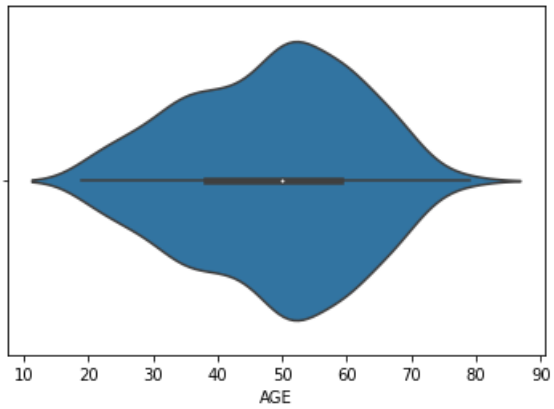
Пропорция по возрасту

In [62]:

```
sns.violinplot(data=data, x="AGE")
```

Out[62]:

```
<AxesSubplot:xlabel='AGE'>
```



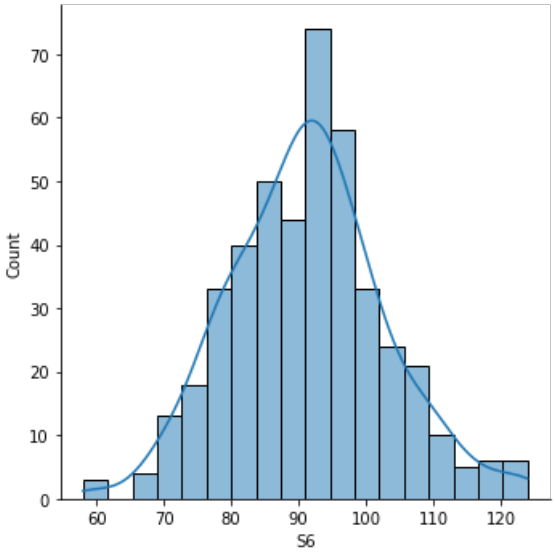
Пропорция по уровню сахара

In [5]:

```
sns.displot(data=data, x="S6", kde = True)
```

Out[5]:

```
<seaborn.axisgrid.FacetGrid at 0x2170e3ed1f0>
```



Пропорция по индексу массы тела

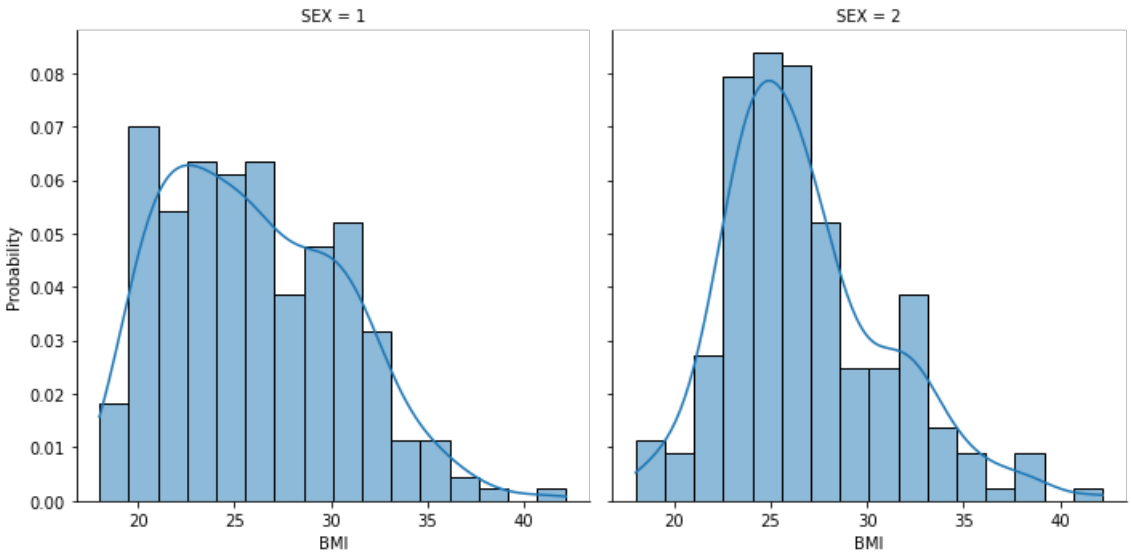
(с разделением по полу)

In [19]:

```
sns.displot(data=data, x="BMI", kde = True, col="SEX", kind="hist", stat="probability")
```

Out[19]:

```
<seaborn.axisgrid.FacetGrid at 0x2179660ec40>
```



На двух данных графиках (и предыдущем) видим практически образцовое нормальное распределение, за исключением того, что называется проблемой "длинного хвоста".

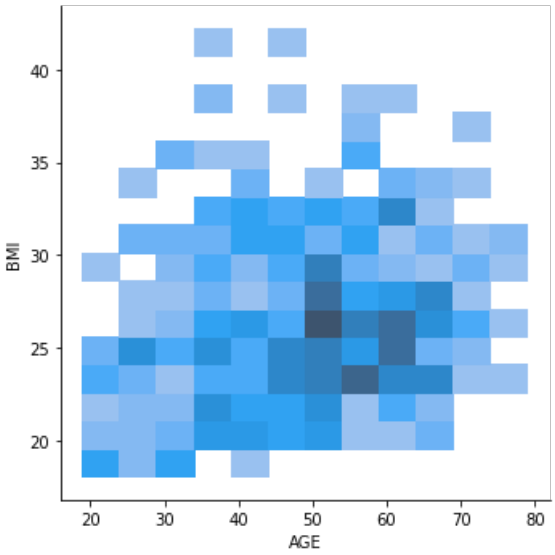
Карта плотностей по возрасту и весу

In [43]:

```
sns.displot(data=data, x="AGE", y="BMI")
```

Out[43]:

```
<seaborn.axisgrid.FacetGrid at 0x1e1b49b6cd0>
```



Из данного графика можно вычленить группу риска

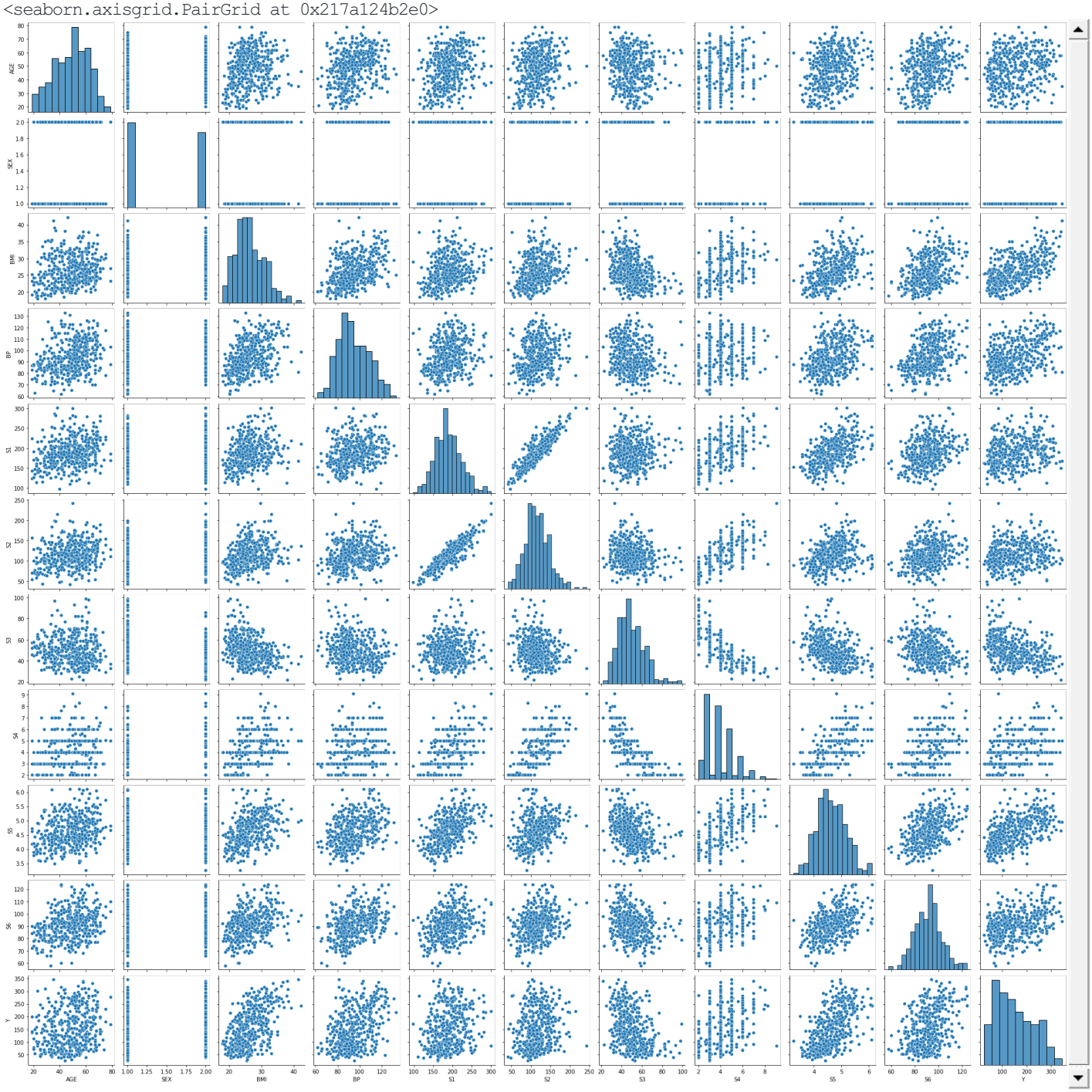
Парные диаграммы

In [22]:

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

Out[22]:

```
<seaborn.axisgrid.PairGrid at 0x217a124b2e0>
```



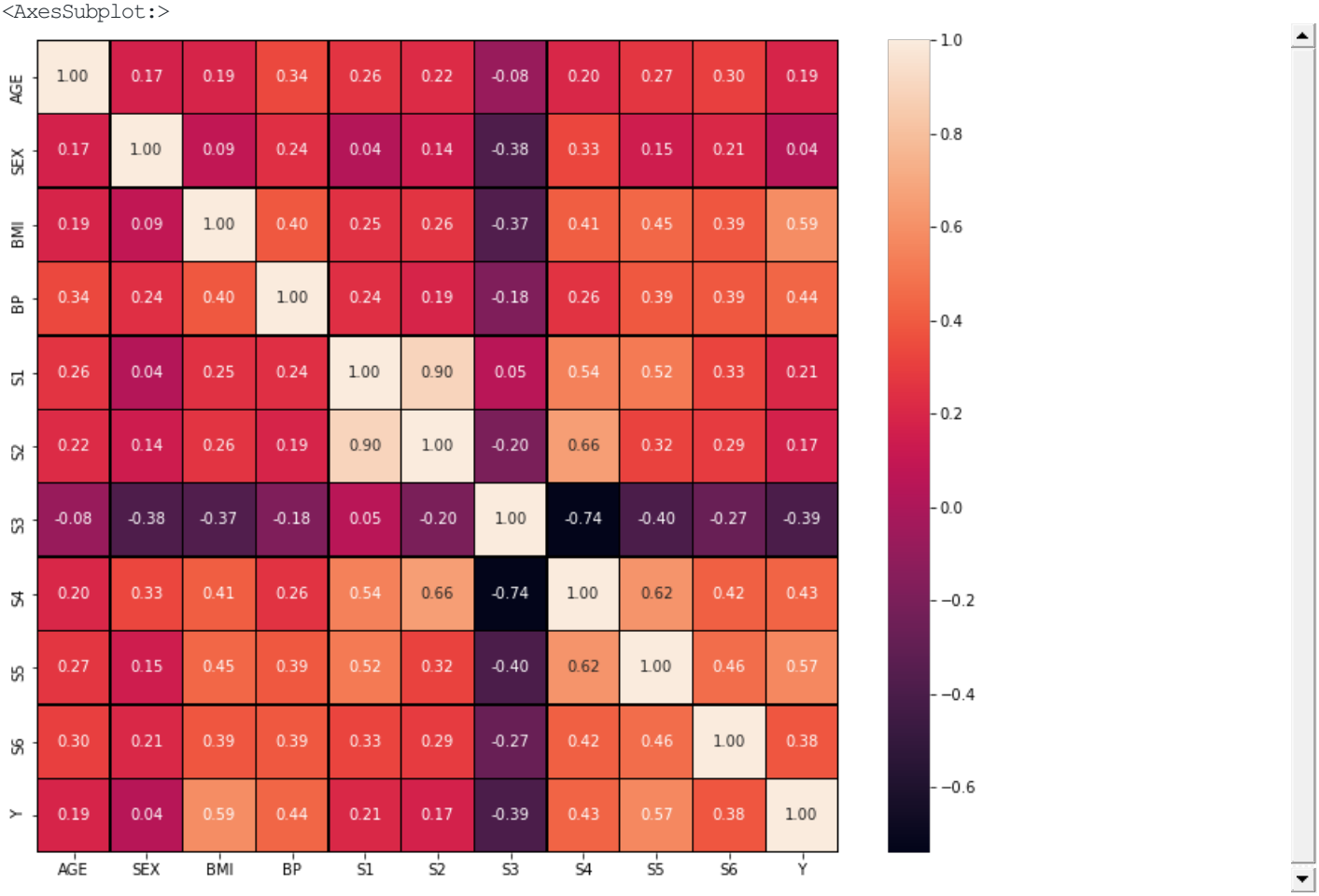
4. Информация о корреляции признаков

In [35]:

```
fig, ax = plt.subplots(figsize=(12, 10))
sns.heatmap(data.corr(), ax=ax, annot=True, fmt=".2f", linewidths=0.3, linecolor="black")
```

Out[35]:

```
<AxesSubplot:>
```



Предположение, сделанное насчёт к-та корреляции Y и S6, оказалось верным!

In [ ]: