

Вардунян А.Т. ИУ5-61Б

1 Оглавление

2. [Задание](#)
3. [Импорт библиотек](#)
4. [Загрузка и первичный анализ данных](#)
5. [Визуализация](#)
6. [Корреляционный анализ](#)

2 Задание ([к оглавлению](#))

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов [здесь](#).
- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных, например из Scikit-learn.
- Пример преобразования датасетов Scikit-learn в Pandas Dataframe можно посмотреть [здесь](#).

Для лабораторных работ не рекомендуется выбирать датасеты большого размера.

- Создать ноутбук, который содержит следующие разделы:
 1. Текстовое описание выбранного Вами набора данных.
 2. Основные характеристики датасета.
 3. Визуальное исследование датасета.
 4. Информация о корреляции признаков.

Сформировать отчет и разместить его в своем репозитории на github.

3 Импорт библиотек ([к оглавлению](#))

```
Ввод [1]: import numpy as np
import pandas as pd

from sklearn.datasets import load_diabetes

import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

4 Загрузка и первичный анализ данных ([к оглавлению](#))

```
Ввод [2]: X, y = load_diabetes(return_X_y=True)
print(load_diabetes()["DESCR"])
```

```
.. _diabetes_dataset:
```

```
Diabetes dataset
```

```
-----
```

Ten baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements were obtained for each of n = 442 diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

****Data Set Characteristics:****

:Number of Instances: 442

:Number of Attributes: First 10 columns are numeric predictive values

:Target: Column 11 is a quantitative measure of disease progression one year after baseline

:Attribute Information:

- age age in years
- sex
- bmi body mass index
- bp average blood pressure
- s1 tc, total serum cholesterol
- s2 ldl, low-density lipoproteins
- s3 hdl, high-density lipoproteins
- s4 tch, total cholesterol / HDL
- s5 ltg, possibly log of serum triglycerides level
- s6 glu, blood sugar level

Note: Each of these 10 feature variables have been mean centered and scaled by the standard deviation times `n_samples` (i.e. the sum of squares of each column totals 1).

Source URL:

<https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html> (<https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html>)

For more information see:

Bradley Efron, Trevor Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regression," Annals of Statistics (with discussion), 407-499.
(https://web.stanford.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)

```
Ввод [3]: df1 = pd.DataFrame(X, columns=["age", "sex", "bmi", "bp", "tc", "ldl", "hdl", "tch", "ltg", "glu"])
df1
```

```
Out[3]:
```

	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641
...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018118	0.044485
439	0.041708	0.050680	-0.015906	0.017282	-0.037344	-0.013840	-0.024993	-0.011080	-0.046879	0.015491
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044528	-0.025930
441	-0.045472	-0.044642	-0.073030	-0.081414	0.083740	0.027809	0.173816	-0.039493	-0.004220	0.003064

442 rows × 10 columns

```
Ввод [4]: df2 = pd.DataFrame(y, columns=["disease_progression"])
df2
```

```
Out[4]:
```

	disease_progression
0	151.0
1	75.0
2	141.0
3	206.0
4	135.0
...	...
437	178.0
438	104.0
439	132.0
440	220.0
441	57.0

442 rows × 1 columns

```
Ввод [5]: df = pd.merge(df1,df2, left_index=True, right_index=True)
df
```

```
Out[5]:
```

	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu	disease_progression
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019908	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068330	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005671	-0.045599	-0.034194	-0.032356	-0.002592	0.002864	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022692	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031991	-0.046641	135.0
...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018118	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017282	-0.037344	-0.013840	-0.024993	-0.011080	-0.046879	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044528	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081414	0.083740	0.027809	0.173816	-0.039493	-0.004220	0.003064	57.0

442 rows × 11 columns

```
Ввод [6]: df.describe()
```

```
Out[6]:
```

	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu	disease_progression
count	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02	4.420000e+02
mean	-3.634285e-16	1.308343e-16	-8.045349e-16	1.281655e-16	-8.835316e-17	1.327024e-16	-4.574646e-16	3.777301e-16	-3.830854e-16	-3.4121e-16	135.0
std	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	4.761905e-02	135.0
min	-1.072256e-01	-4.464164e-02	-9.027530e-02	-1.123996e-01	-1.267807e-01	-1.156131e-01	-1.023071e-01	-7.639450e-02	-1.260974e-01	-1.3771e-01	57.0
25%	-3.729927e-02	-4.464164e-02	-3.422907e-02	-3.665645e-02	-3.424784e-02	-3.035840e-02	-3.511716e-02	-3.949338e-02	-3.324879e-02	-3.3171e-02	135.0
50%	5.383060e-03	-4.464164e-02	-7.283766e-03	-5.670611e-03	-4.320866e-03	-3.819065e-03	-6.584468e-03	-2.592262e-03	-1.947634e-03	-1.0771e-03	135.0
75%	3.807591e-02	5.068012e-02	3.124802e-02	3.564384e-02	2.835801e-02	2.984439e-02	2.931150e-02	3.430886e-02	3.243323e-02	2.791701e-02	135.0
max	1.107267e-01	5.068012e-02	1.705552e-01	1.320442e-01	1.539137e-01	1.987880e-01	1.811791e-01	1.852344e-01	1.335990e-01	1.356111e-01	135.0

```
Ввод [7]: df.shape
```

```
Out[7]: (442, 11)
```

Ввод [8]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   age                   442 non-null   float64
1   sex                   442 non-null   float64
2   bmi                   442 non-null   float64
3   bp                    442 non-null   float64
4   tc                    442 non-null   float64
5   ldl                   442 non-null   float64
6   hdl                   442 non-null   float64
7   tch                   442 non-null   float64
8   ltg                   442 non-null   float64
9   glu                   442 non-null   float64
10  disease_progression    442 non-null   float64
dtypes: float64(11)
memory usage: 38.1 KB
```

Ввод [9]:

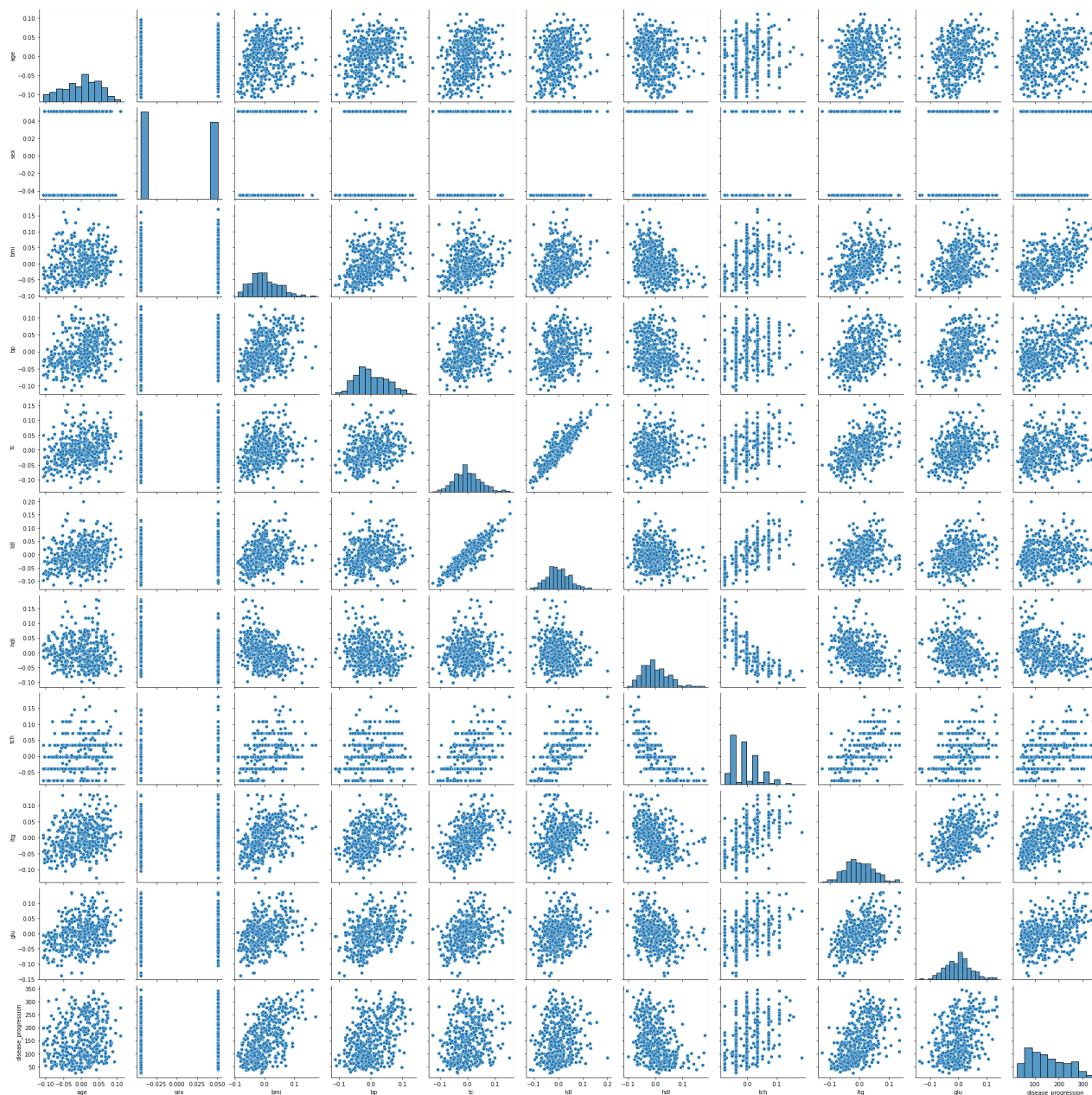
```
# Количество пустых значений
total_count = df.shape[0]
for col in df.columns:
    temp_null_count = df[df[col].isnull()].shape[0]
    temp_perc = round((temp_null_count / total_count) * 100.0, 2)
    print('Колонка {} - {}, {}%'.format(col, temp_null_count, temp_perc))
```

```
Колонка age - 0, 0.0%
Колонка sex - 0, 0.0%
Колонка bmi - 0, 0.0%
Колонка bp - 0, 0.0%
Колонка tc - 0, 0.0%
Колонка ldl - 0, 0.0%
Колонка hdl - 0, 0.0%
Колонка tch - 0, 0.0%
Колонка ltg - 0, 0.0%
Колонка glu - 0, 0.0%
Колонка disease_progression - 0, 0.0%
```

5 Визуализация ([к оглавлению](#))

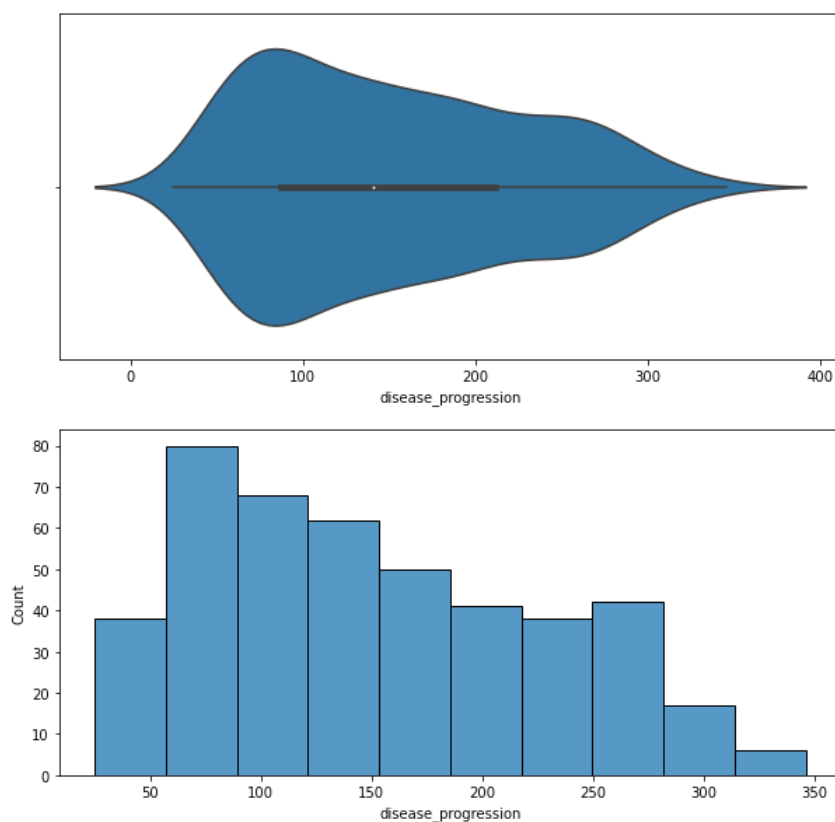
```
Ввод [10]: sns.pairplot(df)
```

```
Out[10]: <seaborn.axisgrid.PairGrid at 0x7f9d3e27deb0>
```



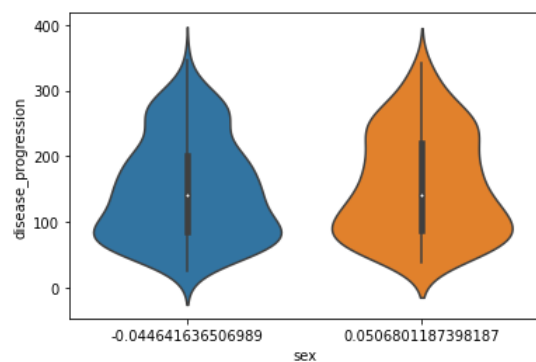
```
Ввод [11]: fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=df['disease_progression'])
sns.histplot(df['disease_progression'], ax=ax[1])
```

Out[11]: <AxesSubplot:xlabel='disease_progression', ylabel='Count'>

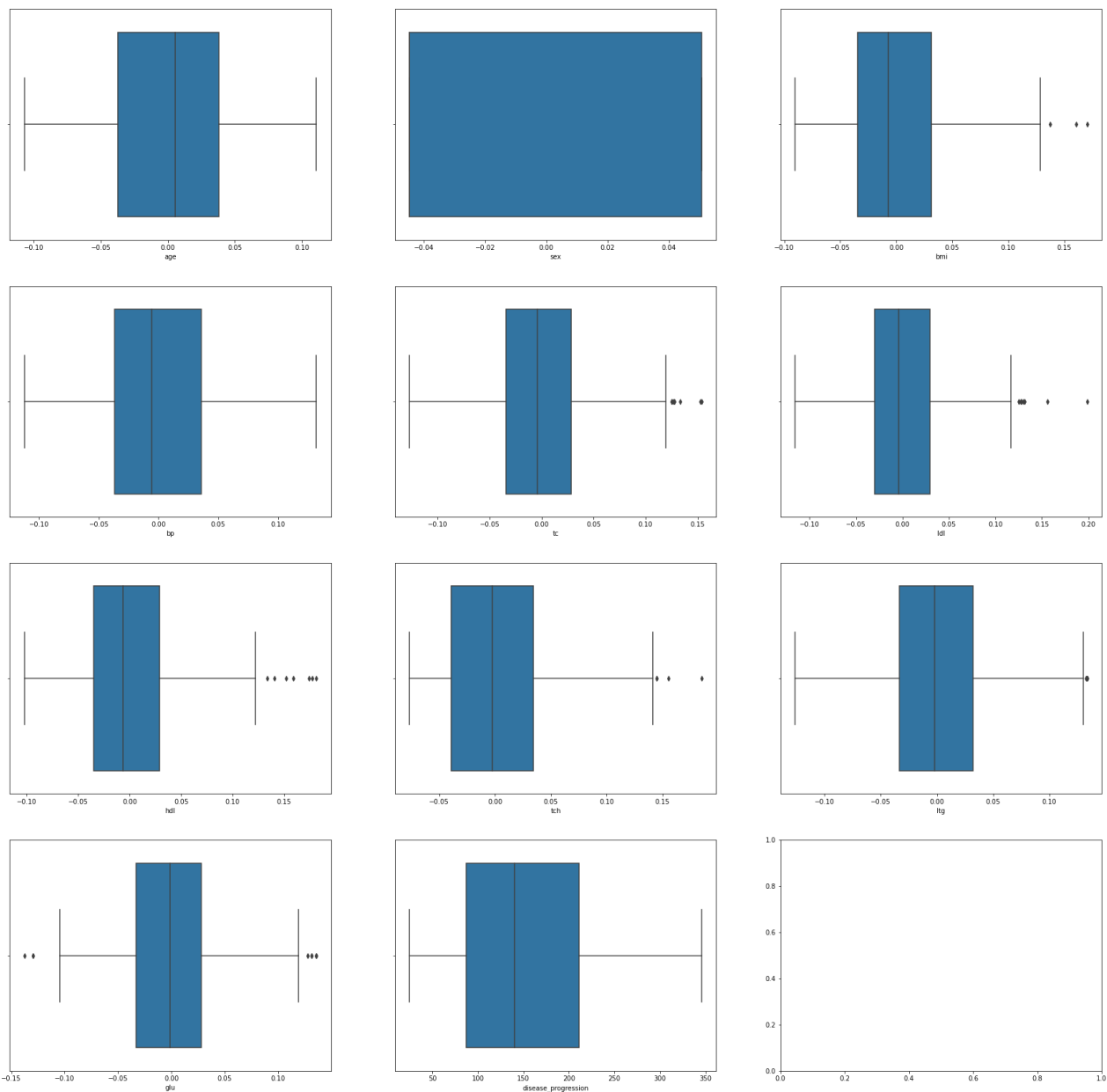


```
Ввод [12]: # Распределение параметра disease_progression сгруппированные по sex.
sns.violinplot(x='sex', y='disease_progression', data=df)
```

Out[12]: <AxesSubplot:xlabel='sex', ylabel='disease_progression'>



```
Ввод [13]: height = 4
width = 3
fig, ax = plt.subplots(height, width, figsize=(30,30))
for i in range(height):
    for j in range(width):
        if i * width + j != 11:
            sns.boxplot(x=df[df.columns[i * width + j]], ax=ax[i, j])
```



6 Корреляционный анализ ([к оглавлению](#))

Ввод [14]: `df.corr()`

Out[14]:

	age	sex	bmi	bp	tc	ldl	hdl	tch	ltg	glu	disease_progression
age	1.000000	0.173737	0.185085	0.335427	0.260061	0.219243	-0.075181	0.203841	0.270777	0.301731	0.187889
sex	0.173737	1.000000	0.088161	0.241013	0.035277	0.142637	-0.379090	0.332115	0.149918	0.208133	0.043062
bmi	0.185085	0.088161	1.000000	0.395415	0.249777	0.261170	-0.366811	0.413807	0.446159	0.388680	0.586450
bp	0.335427	0.241013	0.395415	1.000000	0.242470	0.185558	-0.178761	0.257653	0.393478	0.390429	0.441484
tc	0.260061	0.035277	0.249777	0.242470	1.000000	0.896663	0.051519	0.542207	0.515501	0.325717	0.212022
ldl	0.219243	0.142637	0.261170	0.185558	0.896663	1.000000	-0.196455	0.659817	0.318353	0.290600	0.174054
hdl	-0.075181	-0.379090	-0.366811	-0.178761	0.051519	-0.196455	1.000000	-0.738493	-0.398577	-0.273697	-0.394789
tch	0.203841	0.332115	0.413807	0.257653	0.542207	0.659817	-0.738493	1.000000	0.617857	0.417212	0.430453
ltg	0.270777	0.149918	0.446159	0.393478	0.515501	0.318353	-0.398577	0.617857	1.000000	0.464670	0.565883
glu	0.301731	0.208133	0.388680	0.390429	0.325717	0.290600	-0.273697	0.417212	0.464670	1.000000	0.382483
disease_progression	0.187889	0.043062	0.586450	0.441484	0.212022	0.174054	-0.394789	0.430453	0.565883	0.382483	1.000000

Ввод [15]: `df.corr()['disease_progression']`

Out[15]:

```

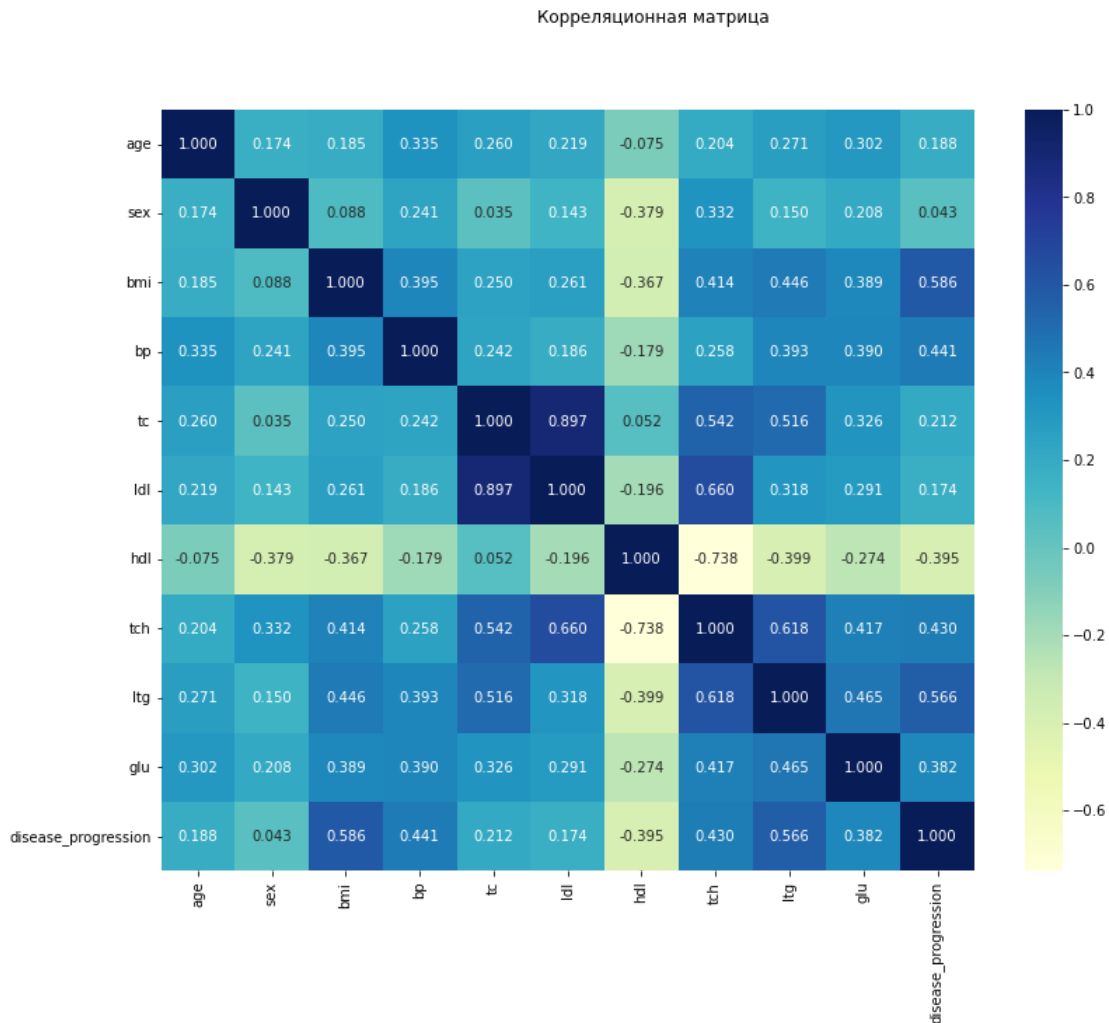
age          0.187889
sex          0.043062
bmi          0.586450
bp           0.441484
tc           0.212022
ldl          0.174054
hdl         -0.394789
tch          0.430453
ltg          0.565883
glu          0.382483
disease_progression  1.000000
Name: disease_progression, dtype: float64

```



```
Ввод [16]: fig, ax = plt.subplots(1, 1, sharex='col', sharey='row', figsize=(13,10))
fig.suptitle('Корреляционная матрица')
sns.heatmap(df.corr(), ax=ax, annot=True, fmt='.3f', cmap='YlGnBu')
```

Out[16]: <AxesSubplot:>



На основе корреляционной матрицы можно сделать следующие выводы.

Лучше всего с целевым признаком `disease_progression` коррелируют следующие признаки:

Признак	Корреляция
bmi	0.586
bp	0.441
tch	0.430
ltg	0.566

При этом признак `sex` вообще не коррелирует с целевым признаком. Признаки `ldl` и `tc` сильно коррелируют между собой (0.897), следовательно, необходимо избавиться от одного из них (от `ldl`, т.к он меньше коррелирует с целевым признаком).