1) Текстовое описание набора данных

В качестве набора данных мы будем использовать набор данных болезни сердца - [https://www.kaggle.com/ronitf/heart-disease-uci (https://www.kaggle.com/ronitf/heart-disease-uci)](https://www.kaggle.com/ronitf/heart-disease-uci)

1. age (1 = male; 0 = female)
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. ldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

In [15]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks", rc={'figure.figsize': (10,10)})
```

In [16]:

```
data = pd.read_csv("C:/Users/VTsapiy/Desktop/лаба1/heart.csv")
```

In [33]:

```
data.head()
```

Out[33]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

2) Основные характеристики датасета

In [34]:

```
data.shape
```

Out[34]:

```
(303, 14)
```

In [35]:

```
total_count = data.shape[0]
print("Всего строк {}".format(total_count))
```

```
Всего строк 303
```

In [36]:

```
data.columns
```

Out[36]:

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

In [37]:

```
data.dtypes
```

Out[37]:

```
age           int64
sex           int64
cp            int64
trestbps      int64
chol          int64
fbs           int64
restecg       int64
thalach       int64
exang         int64
oldpeak     float64
slope         int64
ca            int64
thal          int64
target        int64
dtype: object
```

In [38]:

```
for col in data.columns:
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
age - 0
sex - 0
cp - 0
trestbps - 0
chol - 0
fbs - 0
restecg - 0
thalach - 0
exang - 0
oldpeak - 0
slope - 0
ca - 0
thal - 0
target - 0
```

In [39]:

```
data.describe()
```

Out[39]:

	age	sex	cp	trestbps	chol	fbs	restecg	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	30
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	14
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	2
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	7
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	13
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	15
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	16
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	20

In [41]:

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

Out[41]:

```
array([1, 0], dtype=int64)
```

3) Визуальное исследование датасета

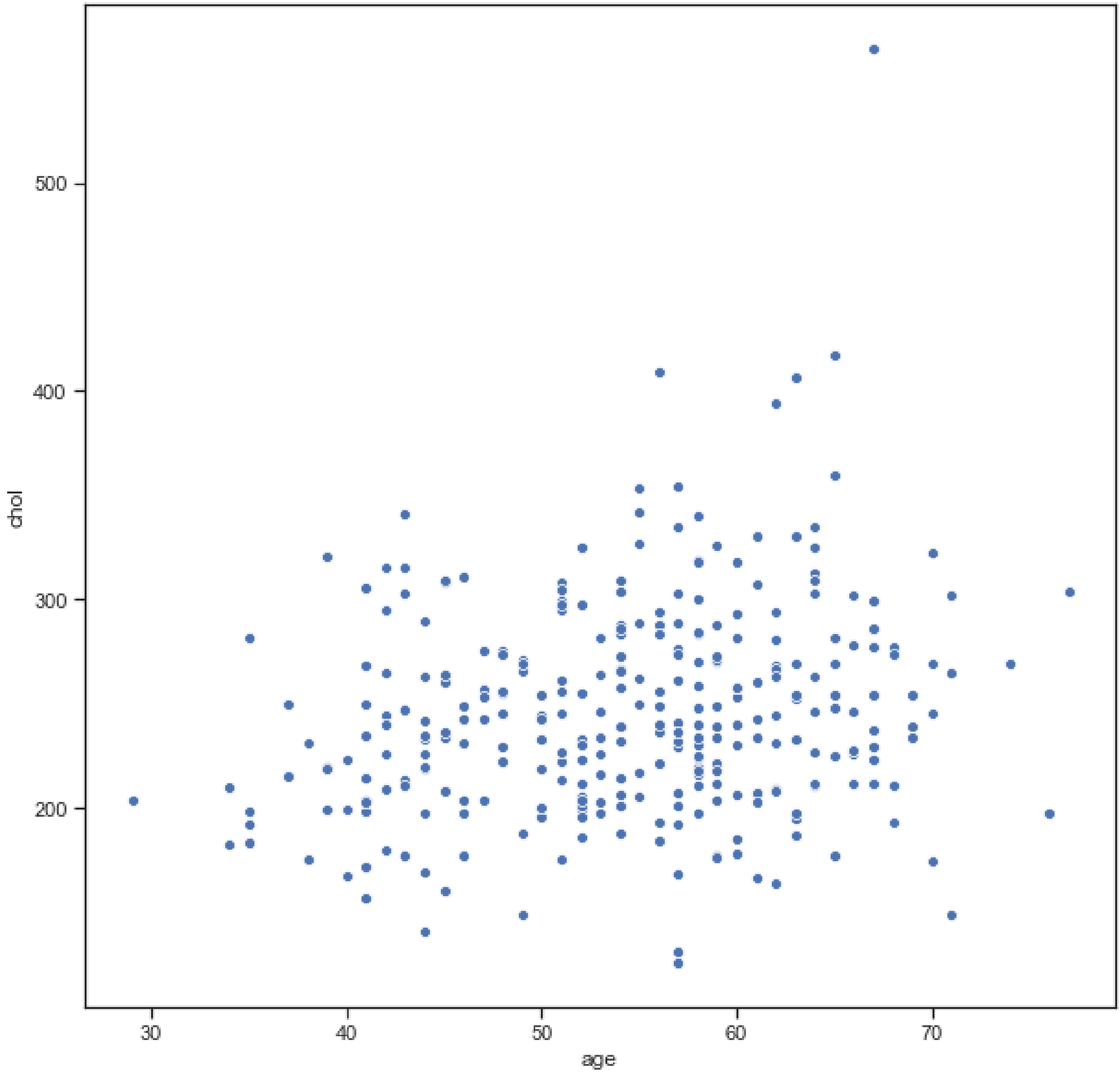
Диаграмма рассеяния

In [68]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='chol', data=data)
```

Out[68]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d676490>
```

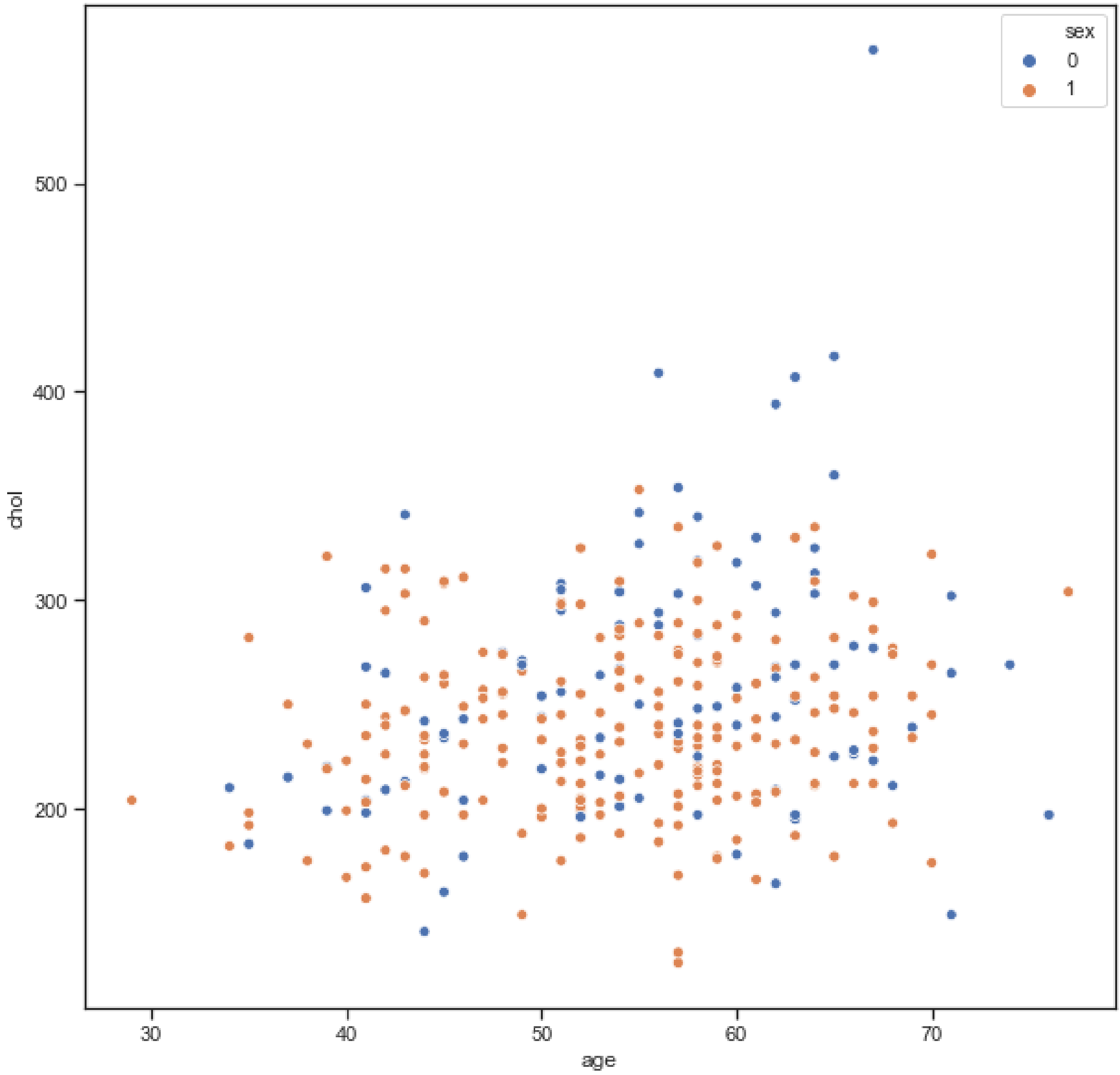


In [69]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='chol', data=data, hue='sex')
```

Out[69]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d99df30>
```



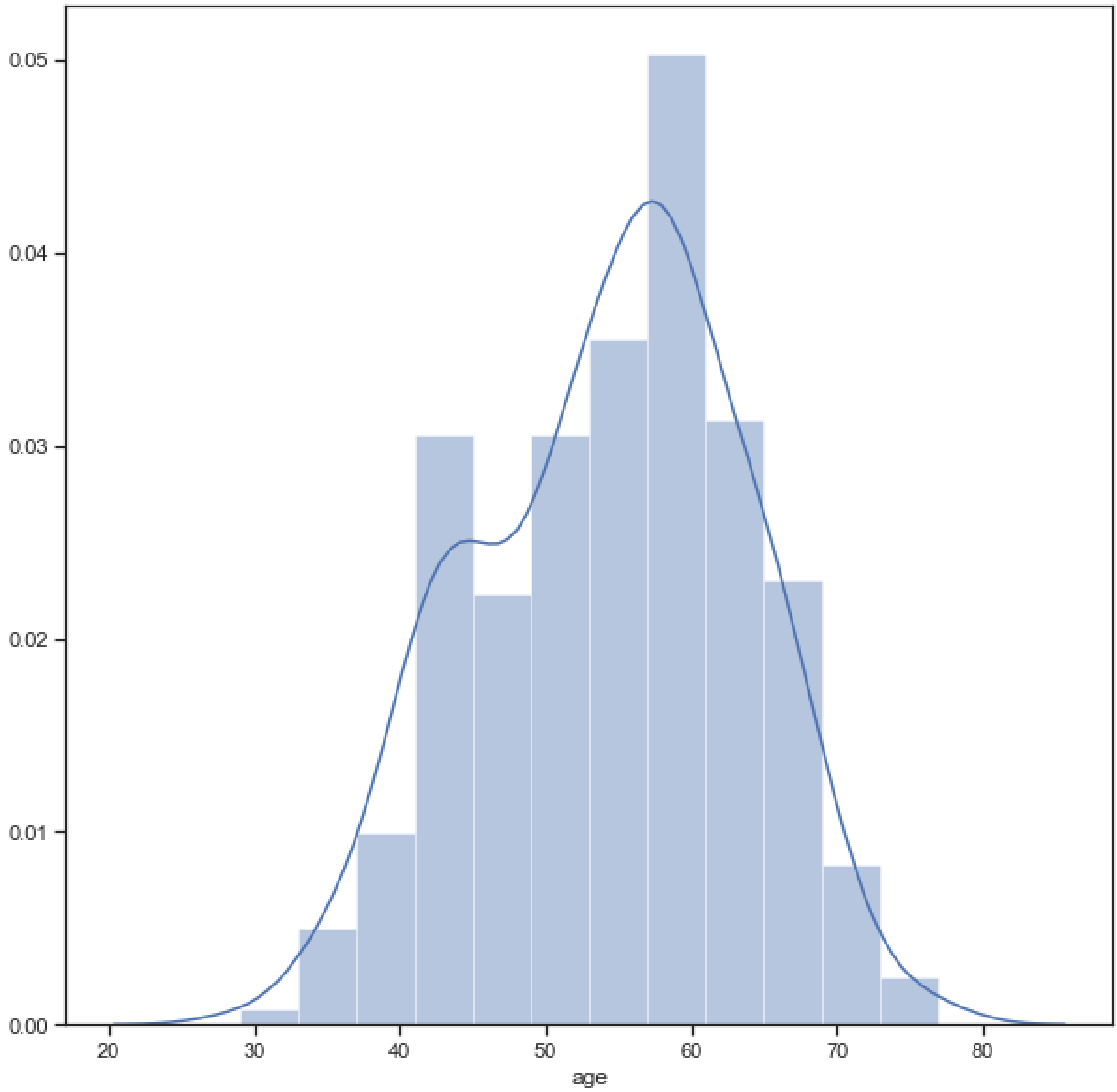
Гистограмма

In [70]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['age'])
```

Out[70]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d9e8490>
```



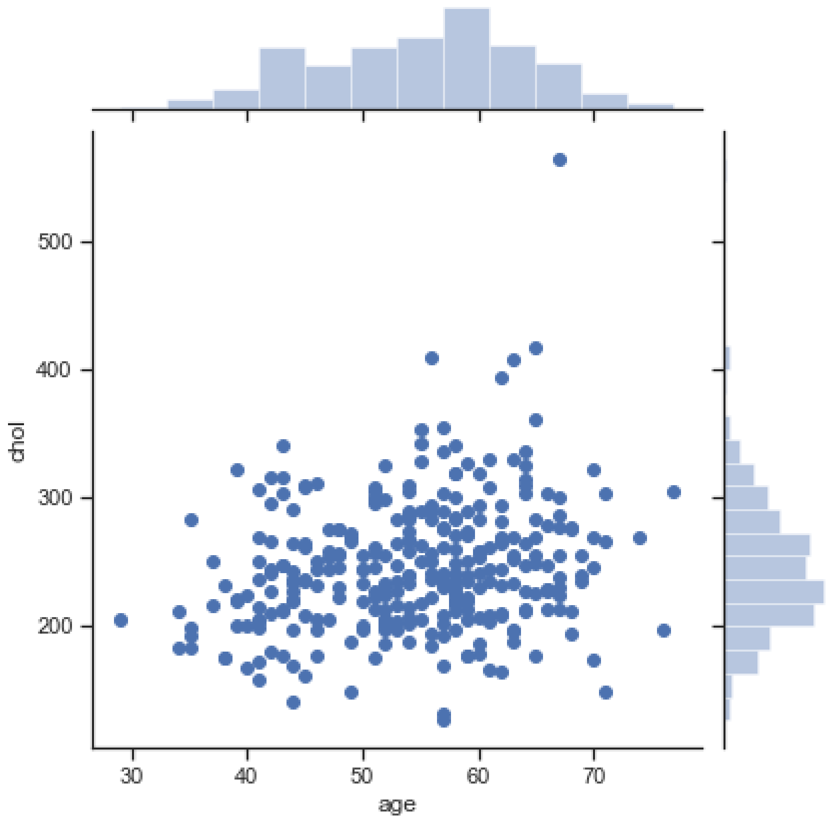
Jointplot Комбинация гистограмм и диаграмм рассеивания.

In [71]:

```
sns.jointplot(x='age', y='chol', data=data)
```

Out[71]:

```
<seaborn.axisgrid.JointGrid at 0x1dbaded0>
```

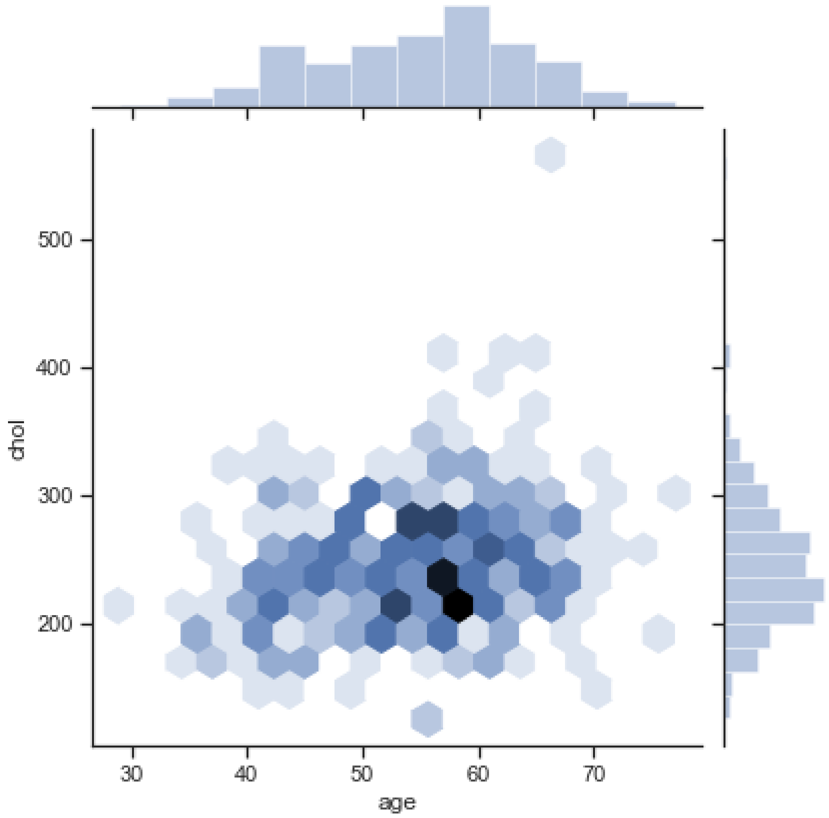


In [72]:

```
sns.jointplot(x='age', y='chol', data=data, kind="hex")
```

Out[72]:

<seaborn.axisgrid.JointGrid at 0x1dea4890>

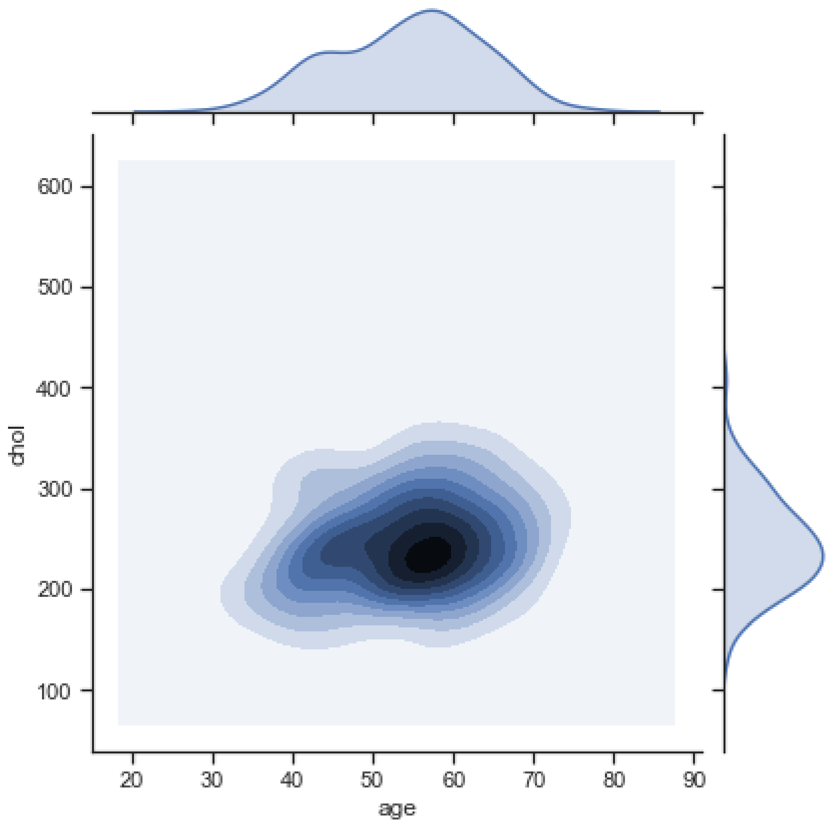


In [73]:

```
sns.jointplot(x='age', y='chol', data=data, kind="kde")
```

Out[73]:

```
<seaborn.axisgrid.JointGrid at 0x1e29bf30>
```



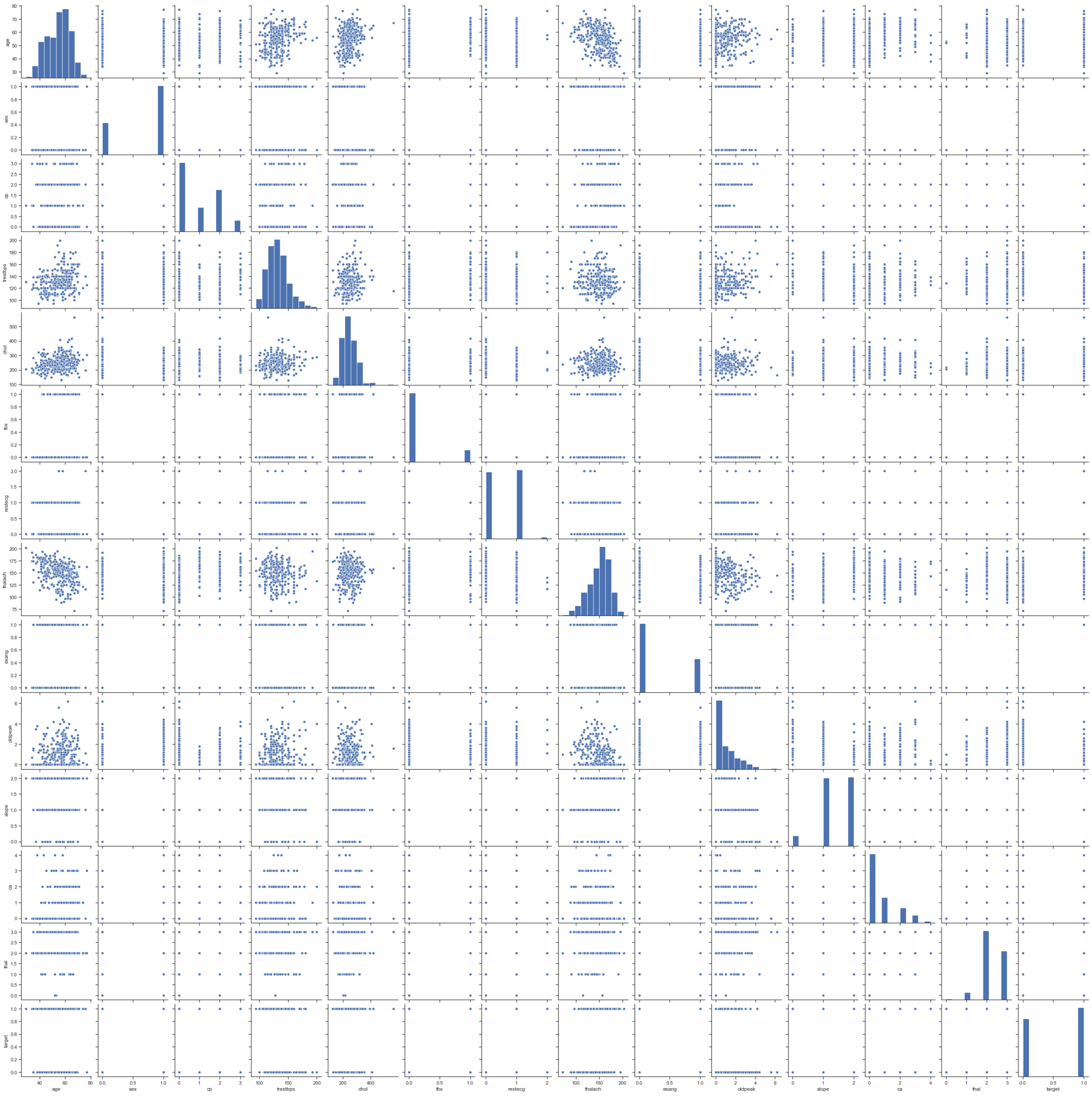
Парные диаграммы

In [74]:

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

Out[74]:

```
<seaborn.axisgrid.PairGrid at 0x1e2ad9b0>
```

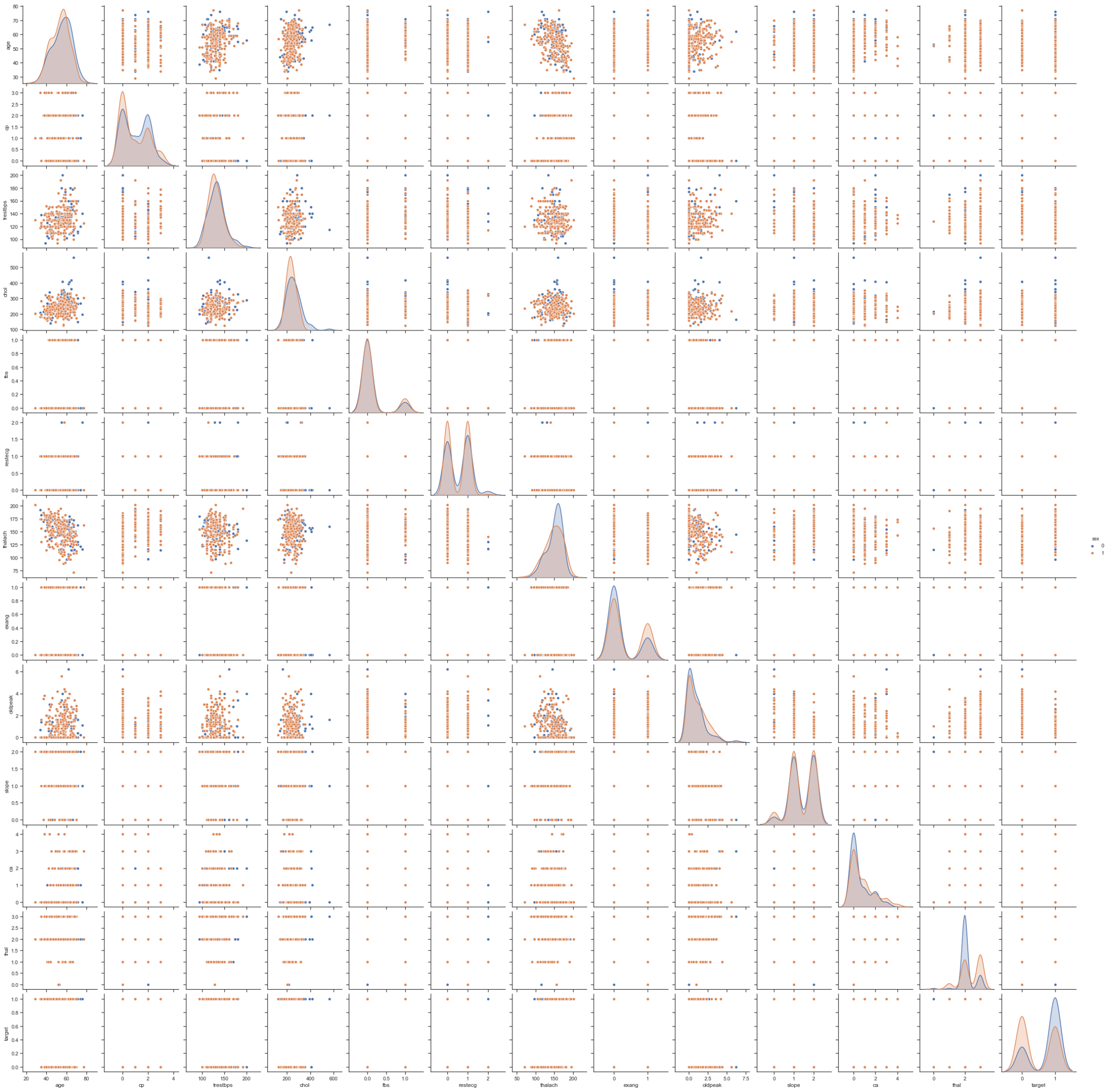


In [75]:

```
sns.pairplot(data, hue="sex")
```

Out[75]:

```
<seaborn.axisgrid.PairGrid at 0x25017dd0>
```



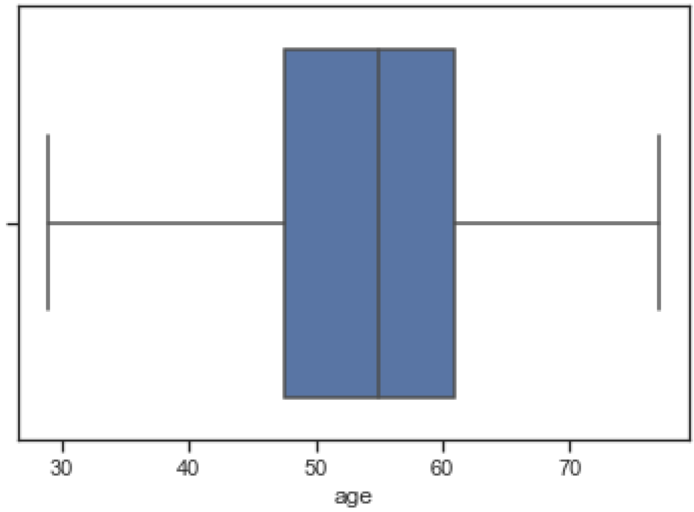
Ящик с усами

In [80]:

```
sns.boxplot(x=data['age'])
```

Out[80]:

<matplotlib.axes._subplots.AxesSubplot at 0x29ee5e50>

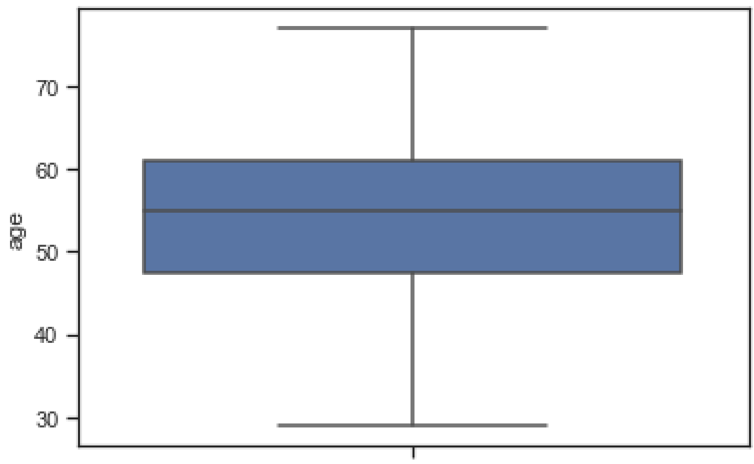


In [79]:

```
sns.boxplot(y=data['age'])
```

Out[79]:

<matplotlib.axes._subplots.AxesSubplot at 0x29ebc0f0>

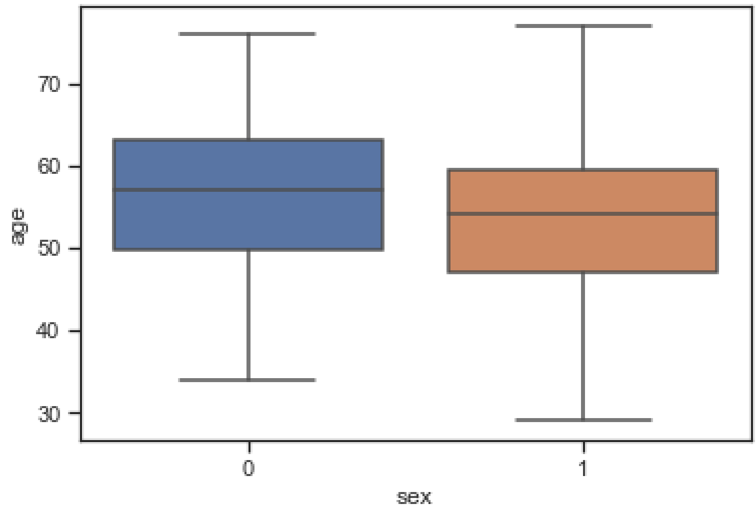


In [81]:

```
sns.boxplot(x='sex', y='age', data=data)
```

Out[81]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29f19130>
```



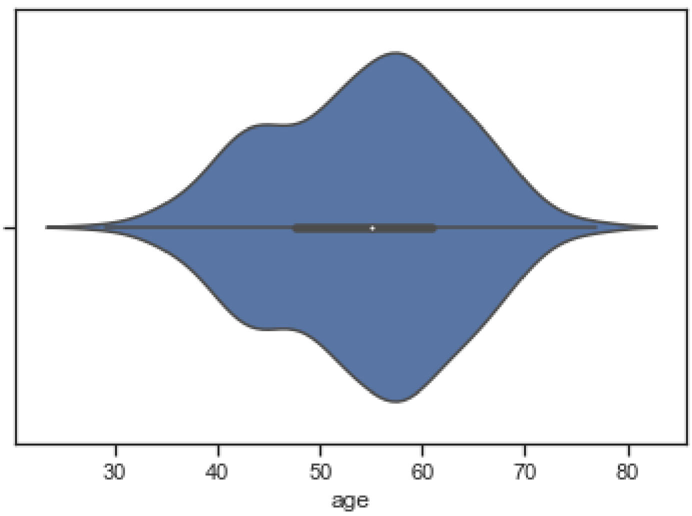
Violin plot

In [83]:

```
sns.violinplot(x=data['age'])
```

Out[83]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x3653bed0>
```

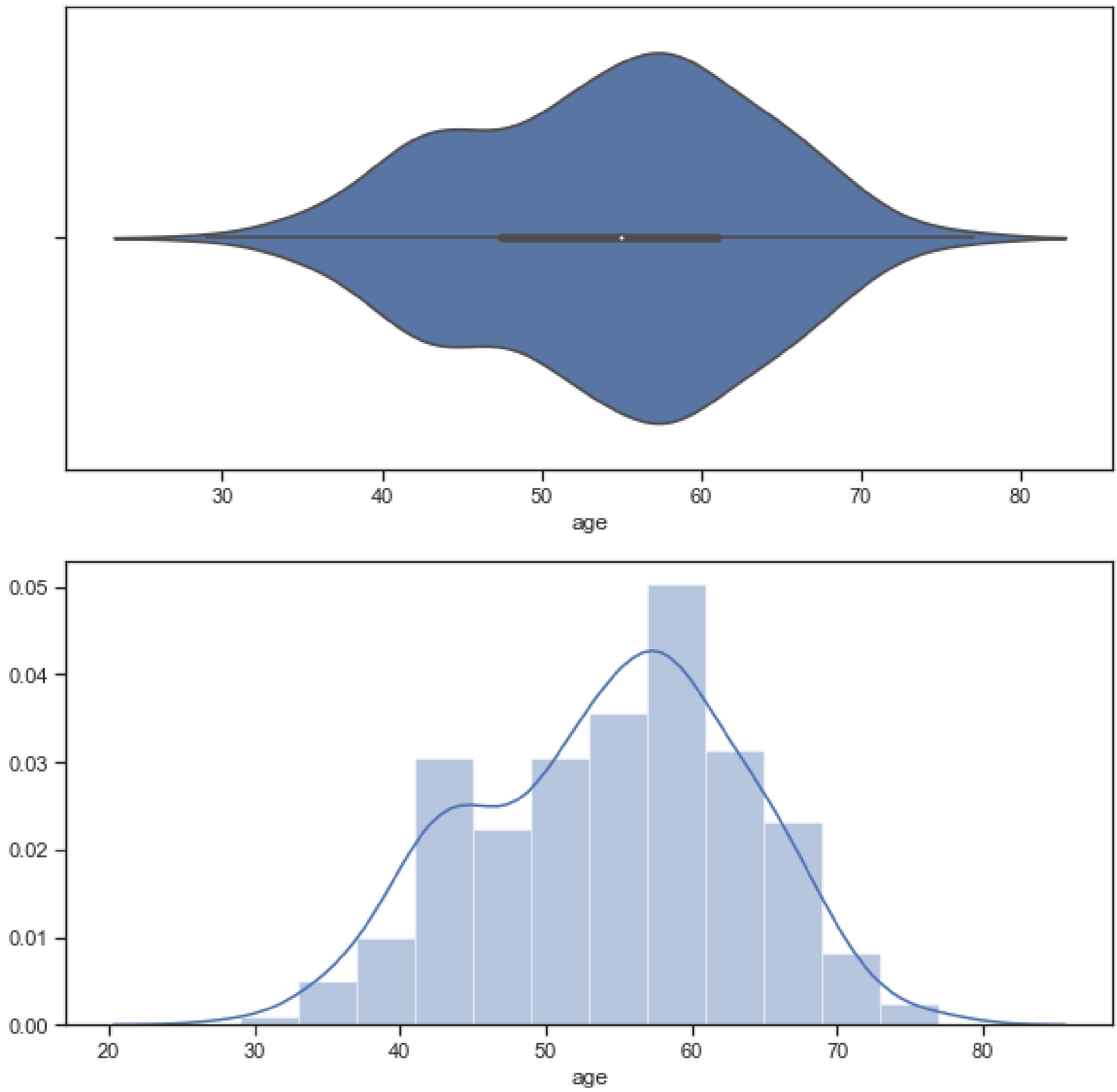


In [84]:

```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['age'])
sns.distplot(data['age'], ax=ax[1])
```

Out[84]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x36594070>
```

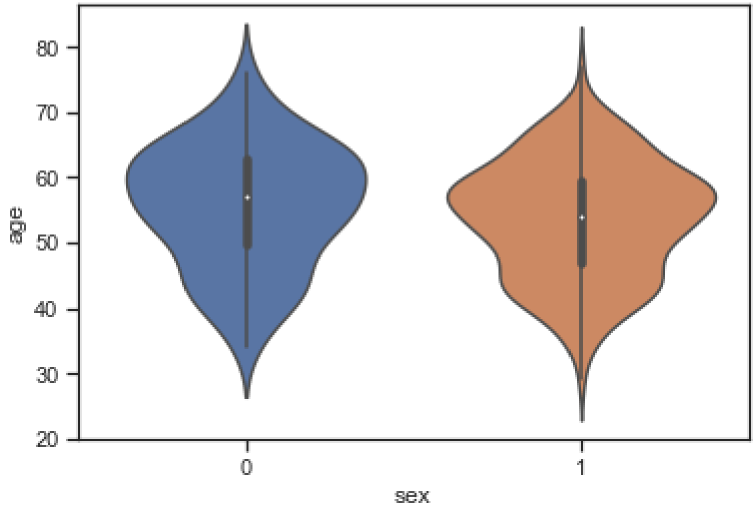


In [85]:

```
sns.violinplot(x='sex', y='age', data=data)
```

Out[85]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x36600490>
```

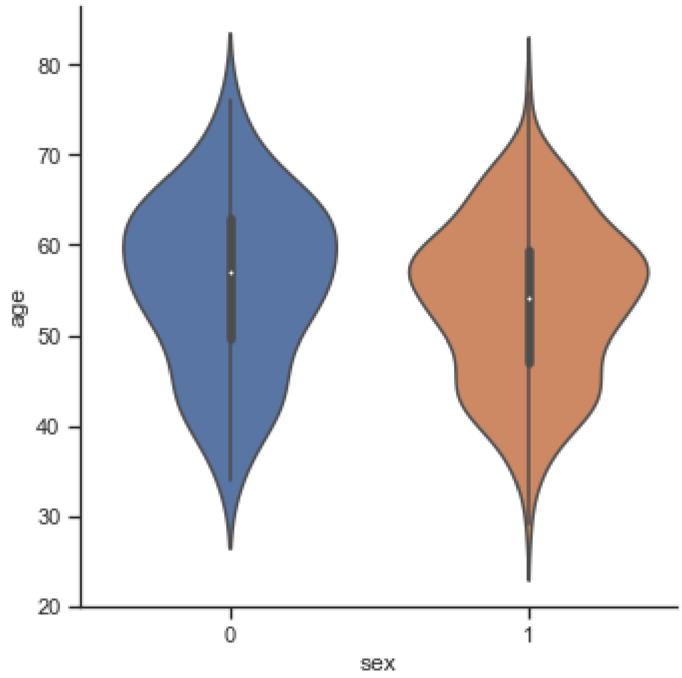


In [86]:

```
sns.catplot(y='age', x='sex', data=data, kind="violin", split=True)
```

Out[86]:

<seaborn.axisgrid.FacetGrid at 0x36634a70>



Информация о корреляции признаков

In [4]:

```
data.corr()
```

Out[4]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [5]:

```
data.corr(method='pearson')
```

Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020
cp	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762
trestbps	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698
chol	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940
fbs	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123
thalach	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784
ca	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741

In [7]:

```
data.corr(method='kendall')
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.082272	-0.071577	0.201071	0.135062	0.094595	-0.109349	-0.280009
sex	-0.082272	1.000000	-0.057955	-0.044438	-0.124104	0.045032	-0.048085	-0.032817
cp	-0.071577	-0.057955	1.000000	0.027548	-0.069899	0.083862	0.060839	0.246160
trestbps	0.201071	-0.044438	0.027548	1.000000	0.086474	0.127574	-0.105147	-0.027760
chol	0.135062	-0.124104	-0.069899	0.086474	1.000000	0.015140	-0.132664	-0.031437
fbs	0.094595	0.045032	0.083862	0.127574	0.015140	1.000000	-0.080996	-0.011749
restecg	-0.109349	-0.048085	0.060839	-0.105147	-0.132664	-0.080996	1.000000	0.072481
thalach	-0.280009	-0.032817	0.246160	-0.027760	-0.031437	-0.011749	0.072481	1.000000
exang	0.074427	0.141664	-0.390708	0.044419	0.075044	0.025665	-0.076913	-0.329965
oldpeak	0.193269	0.086437	-0.125081	0.109103	0.035176	0.024342	-0.066262	-0.306843
slope	-0.147713	-0.024333	0.145796	-0.070360	-0.010039	-0.044546	0.110042	0.349702
ca	0.273255	0.112199	-0.189400	0.070387	0.088549	0.126434	-0.091541	-0.198407
thal	0.070722	0.244164	-0.188999	0.049028	0.066255	-0.006559	-0.010692	-0.130239
target	-0.197857	-0.280937	0.430506	-0.102064	-0.099131	-0.028046	0.147678	0.352609

In [9]:

```
data.corr(method='spearman')
```

Out[9]:

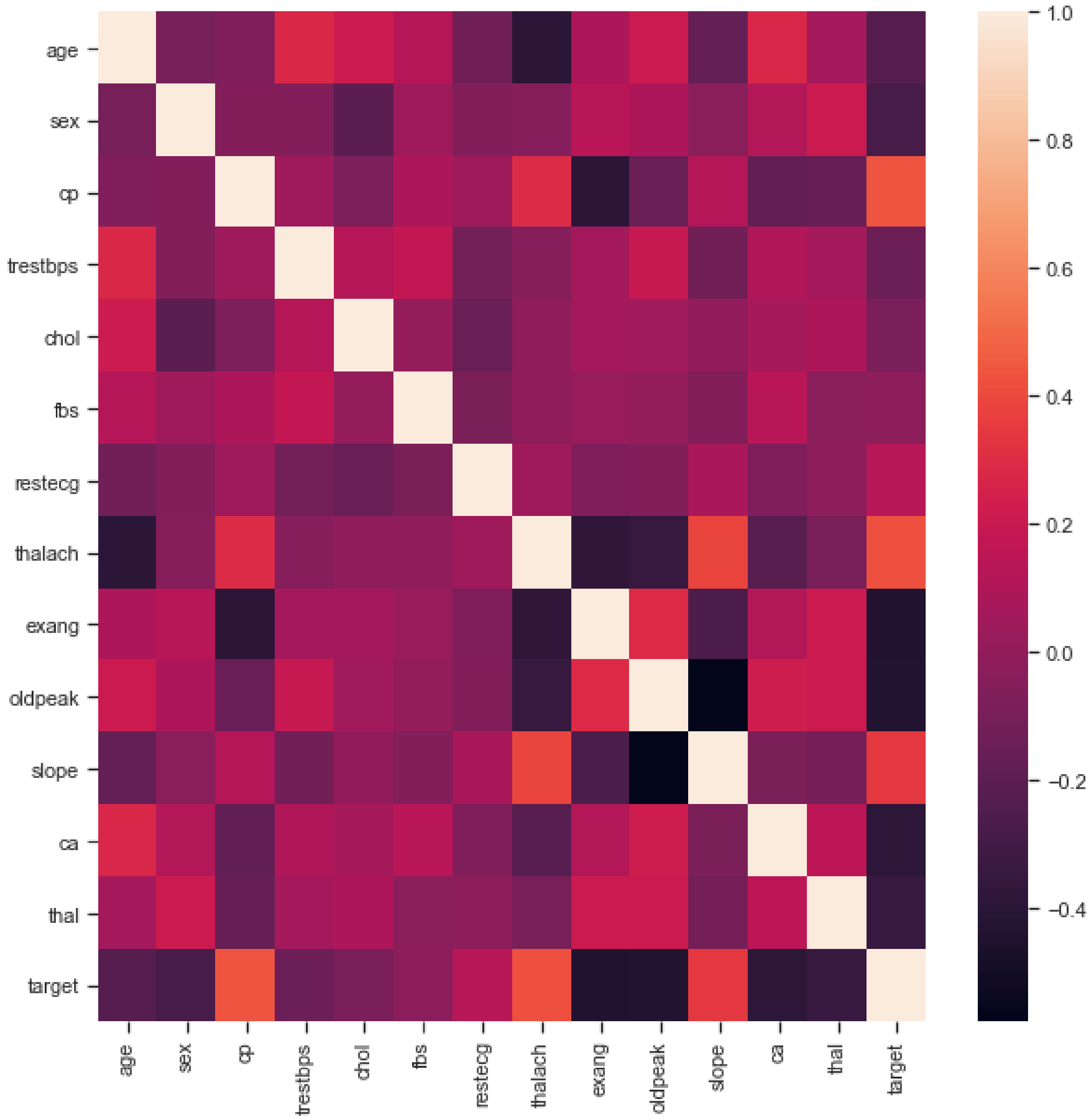
	age	sex	cp	trestbps	chol	fbs	restecg	thalach
age	1.000000	-0.099131	-0.087494	0.285617	0.195786	0.113978	-0.132769	-0.398052
sex	-0.099131	1.000000	-0.062041	-0.052941	-0.151342	0.045032	-0.048389	-0.039868
cp	-0.087494	-0.062041	1.000000	0.035413	-0.091721	0.089775	0.065640	0.324013
trestbps	0.285617	-0.052941	0.035413	1.000000	0.126562	0.151984	-0.125841	-0.040407
chol	0.195786	-0.151342	-0.091721	0.126562	1.000000	0.018463	-0.161933	-0.046766
fbs	0.113978	0.045032	0.089775	0.151984	0.018463	1.000000	-0.081508	-0.014273
restecg	-0.132769	-0.048389	0.065640	-0.125841	-0.161933	-0.081508	1.000000	0.087863
thalach	-0.398052	-0.039868	0.324013	-0.040407	-0.046766	-0.014273	0.087863	1.000000
exang	0.089679	0.141664	-0.418256	0.052918	0.091514	0.025665	-0.077399	-0.400860
oldpeak	0.268291	0.100715	-0.161449	0.154267	0.045260	0.028363	-0.077372	-0.433241
slope	-0.184048	-0.025010	0.159478	-0.086570	-0.012551	-0.045786	0.113661	0.436968
ca	0.340955	0.119368	-0.216006	0.090140	0.111981	0.134513	-0.097862	-0.257347
thal	0.087254	0.250821	-0.207840	0.059673	0.083628	-0.006737	-0.010982	-0.160581
target	-0.238400	-0.280937	0.460860	-0.121593	-0.120888	-0.028046	0.148612	0.428370

In [17]:

```
sns.heatmap(data.corr())
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xa921c90>
```

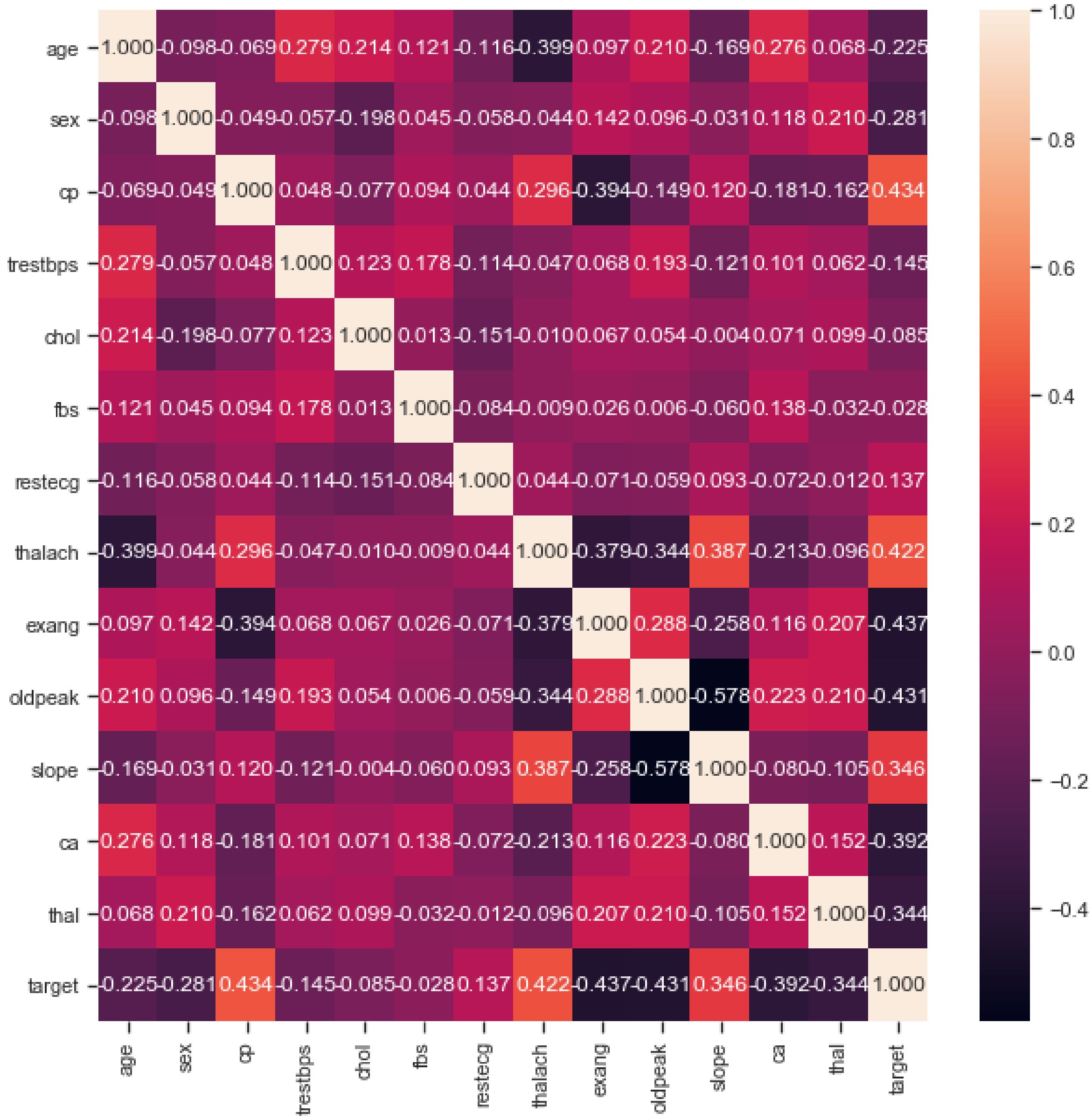


In [19]:

```
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe0c4110>
```

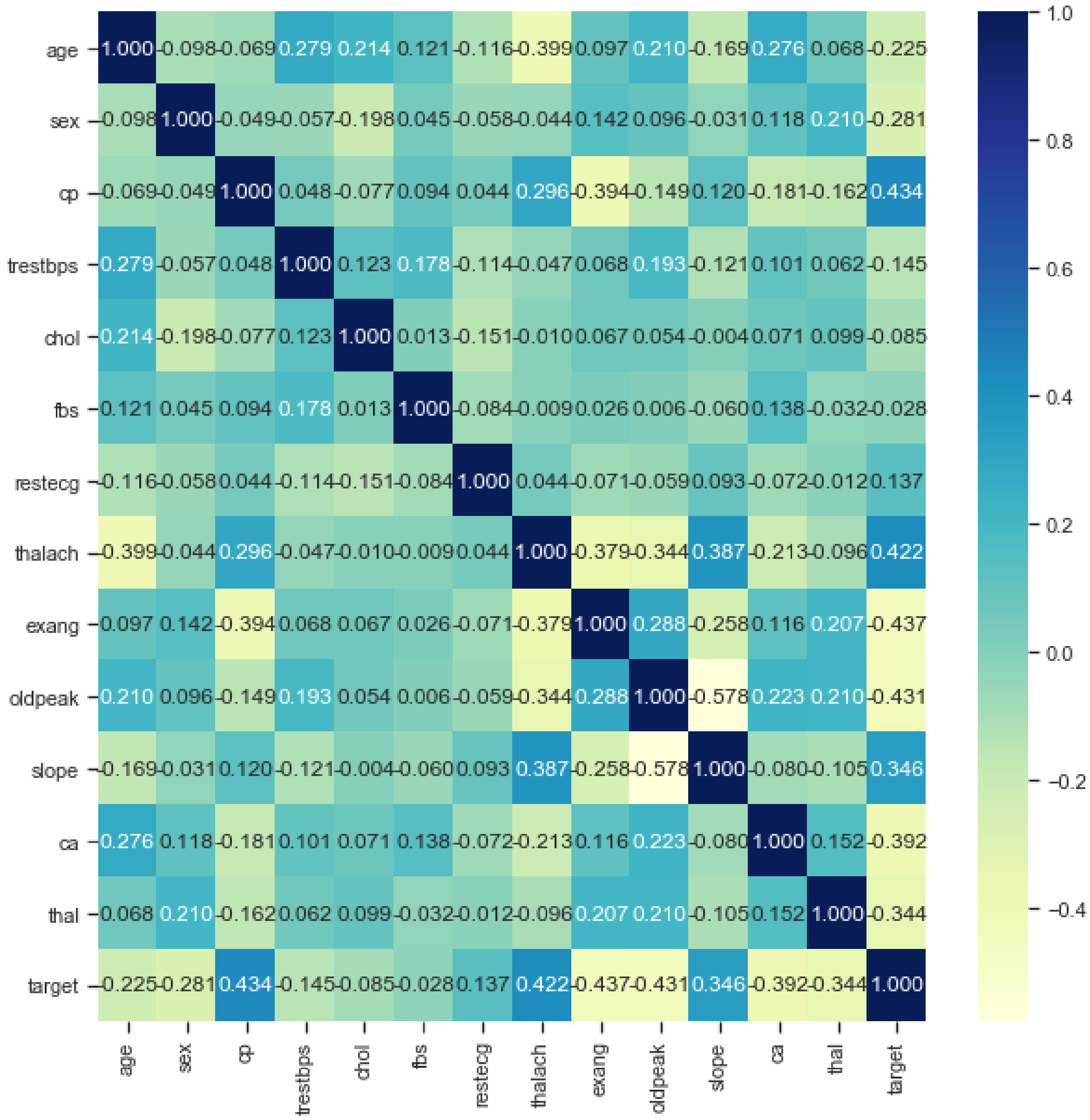


In [22]:

```
sns.heatmap(data.corr(), annot=True, fmt='.3f', cmap='YlGnBu')
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x4959870>
```

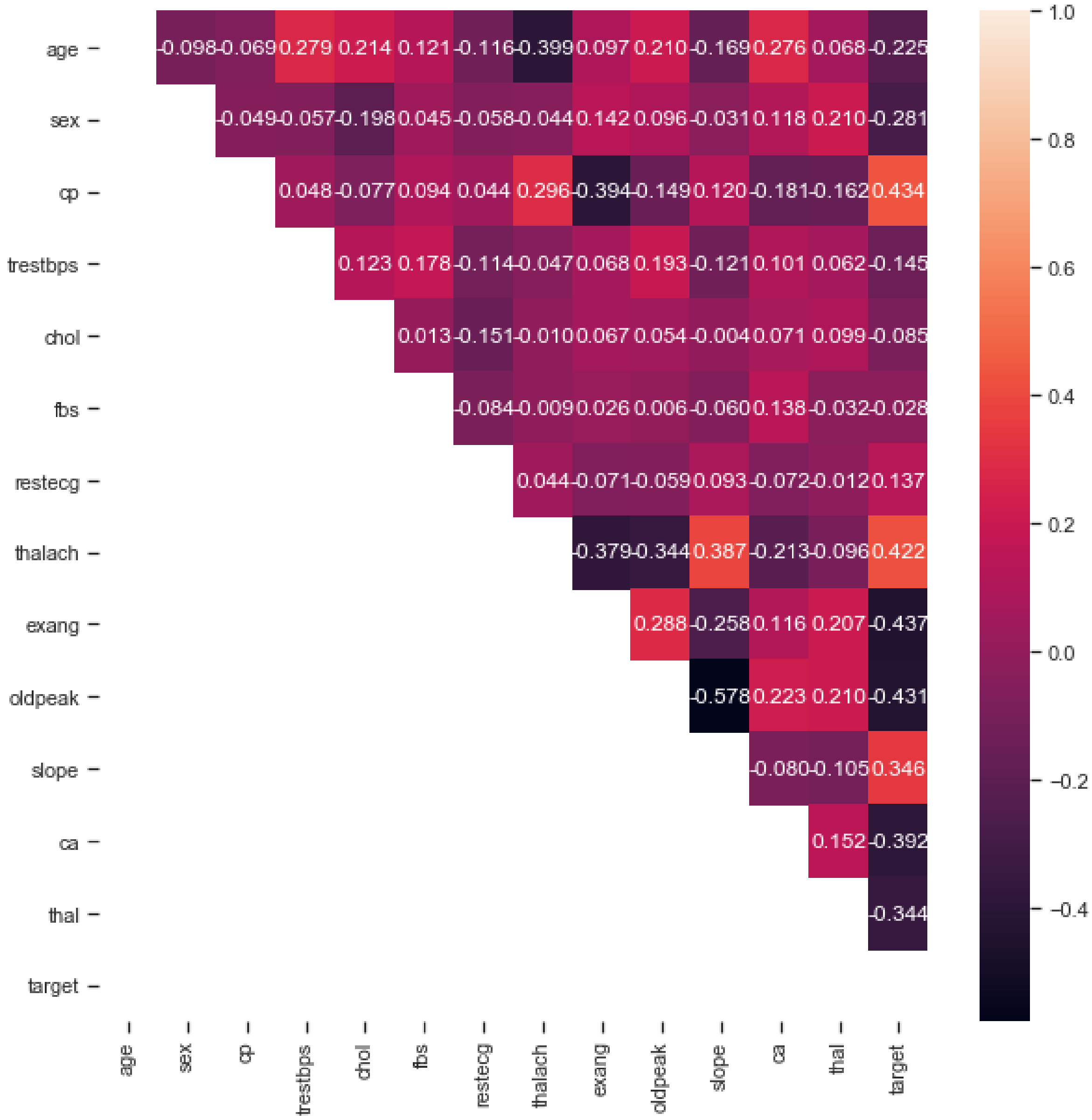


In [23]:

```
# Треугольный вариант матрицы
mask = np.zeros_like(data.corr(), dtype=np.bool)
# чтобы оставить нижнюю часть матрицы
# mask[np.triu_indices_from(mask)] = True
# чтобы оставить верхнюю часть матрицы
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

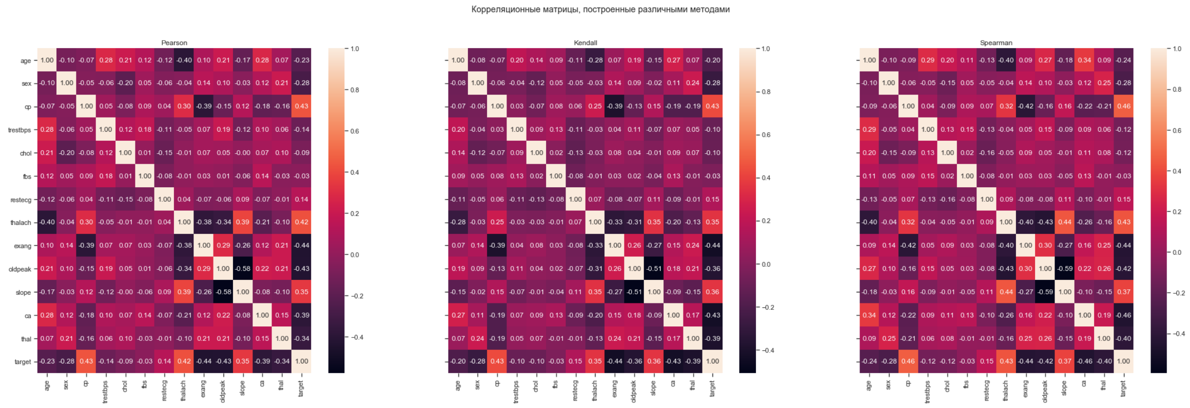
Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe4bdef0>
```



In [33]:

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(35,10))
sns.heatmap(data.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```



In [ ]:

```

```