

1. РК №1

```
[0]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

```
[0]: data = pd.read_csv('heart.csv', sep=",")
```

2. Дата-сет сердечных заболеваний(описание)

age - age in years (возраст)
sex - (1 = male; 0 = female) (пол)
cp - chest pain type (тип боли в груди)
trestbps - resting blood pressure (in mm Hg on admission to the hospital) (кровяное давление в состоянии покоя)
chol - serum cholestoral in mg/dl (давление холестерина в сыворотке крови)
fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false) (уровень сахара в крови натощак)
restecg - resting electrocardiographic results (электрокардиографические результаты покоя)
thalach - maximum heart rate achieved (максимальная частота сердечных сокращений)
exang - exercise induced angina (1 = yes; 0 = no) (стенокардия, вызванная физической нагрузкой)
oldpeak - ST depression induced by exercise relative to rest (понижение ST-сегмента на электрокардиограмме, вызванное физ. упражнениями по сравнению с состоянием покоя)
slope - the slope of the peak exercise ST segment (элевация сегмента ST)
ca - number of major vessels (0-3) colored by fluoroscopy (количество крупных сосудов (0-3), окрашенных по цвету)
thal - 3 = normal; 6 = fixed defect; 7 = reversable defect (результат талиевого стресс-теста)
target - 1 or 0 (целевой признак: наличие или отсутствие сердечного заболевания)

```
[0]: data.head()
```

```
[0]:
```

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

[5 rows x 14 columns]

```
[0]: data.shape
```

```
[0]: (303, 14)
```

```
[0]: data.dtypes
```

```
[0]: 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
```

```
[0]: # Проверим наличие пустых значений
     # Цикл по колонкам датасета
     for col in data.columns:
         # Количество пустых значений - все значения заполнены
         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
```

```
[0]: data.describe()
```

```
[0]:
```

	age	sex	cp	...	ca	thal
target						
count	303.000000	303.000000	303.000000	...	303.000000	303.000000
mean	54.366337	0.683168	0.966997	...	0.729373	2.313531
std	9.082101	0.466011	1.032052	...	1.022606	0.612277
min	29.000000	0.000000	0.000000	...	0.000000	0.000000
25%	47.500000	0.000000	0.000000	...	0.000000	2.000000
50%	55.000000	1.000000	1.000000	...	0.000000	2.000000
75%	61.000000	1.000000	2.000000	...	1.000000	3.000000
max	77.000000	1.000000	3.000000	...	4.000000	3.000000

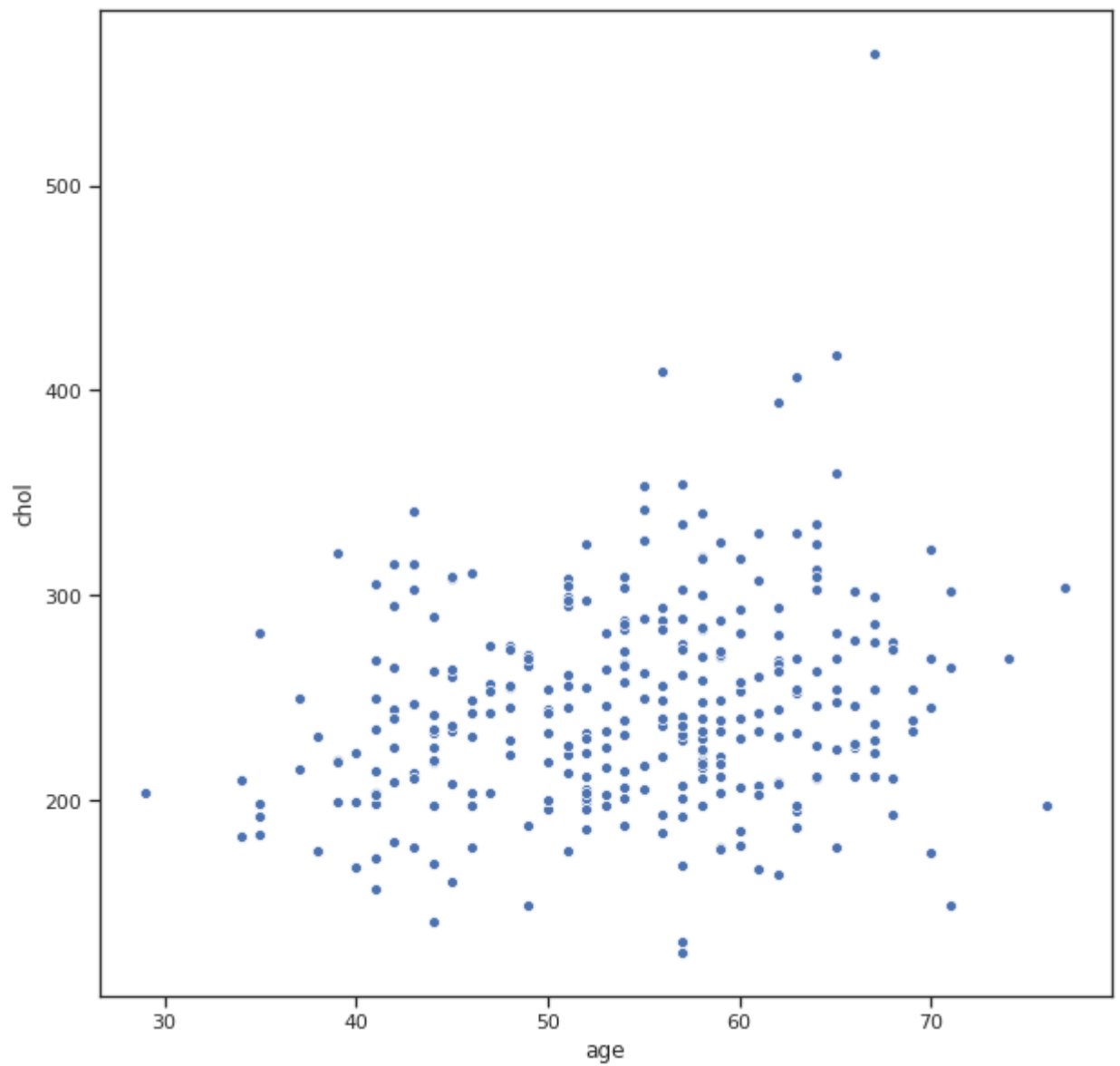
```
[8 rows x 14 columns]
```

```
[0]: # Определим уникальные значения для целевого признака
data['target'].unique()
```

```
[0]: array([1, 0])
```

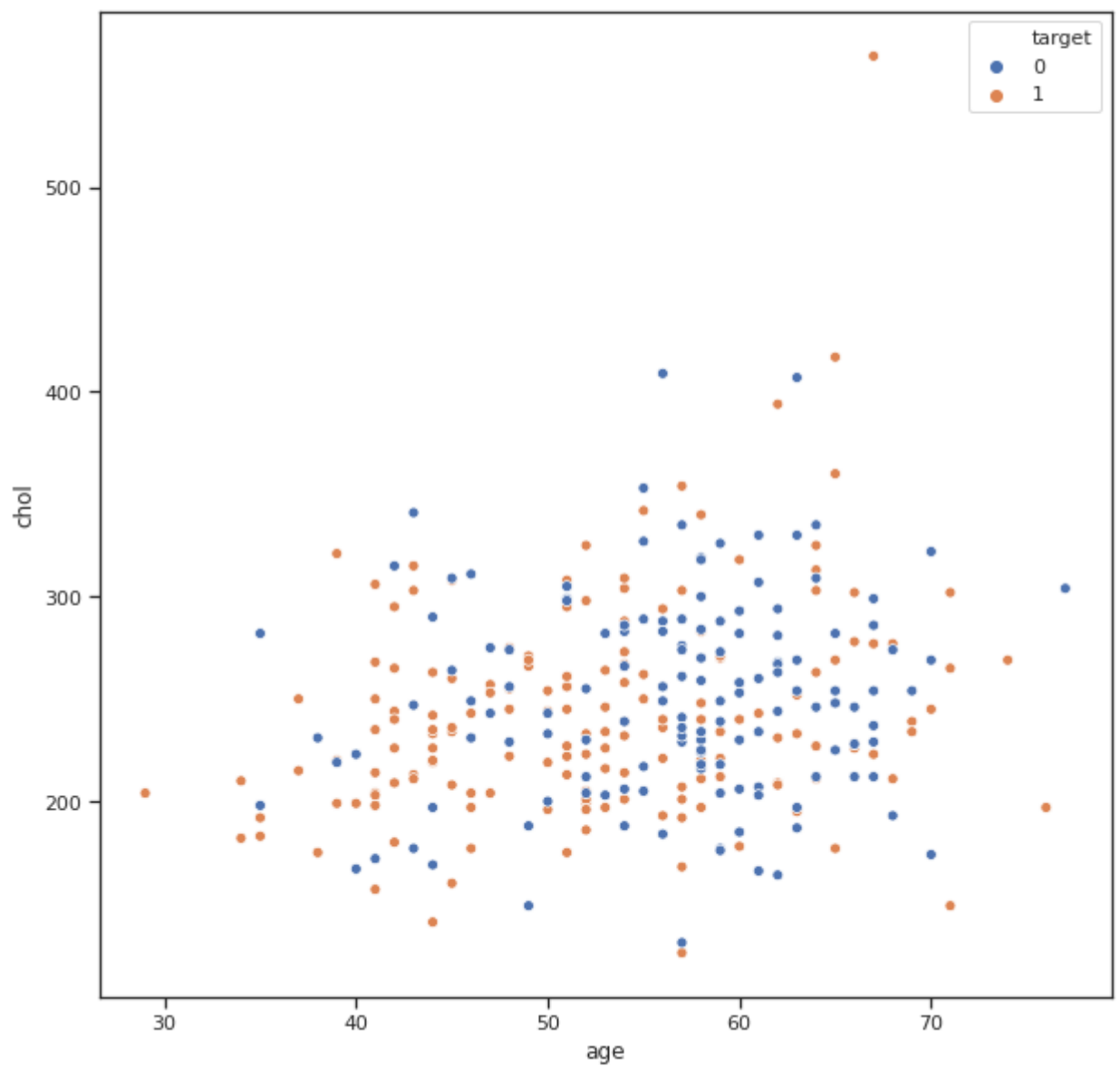
```
[0]: fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='age', y='chol', data=data)
```

```
[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d2614ff60>
```



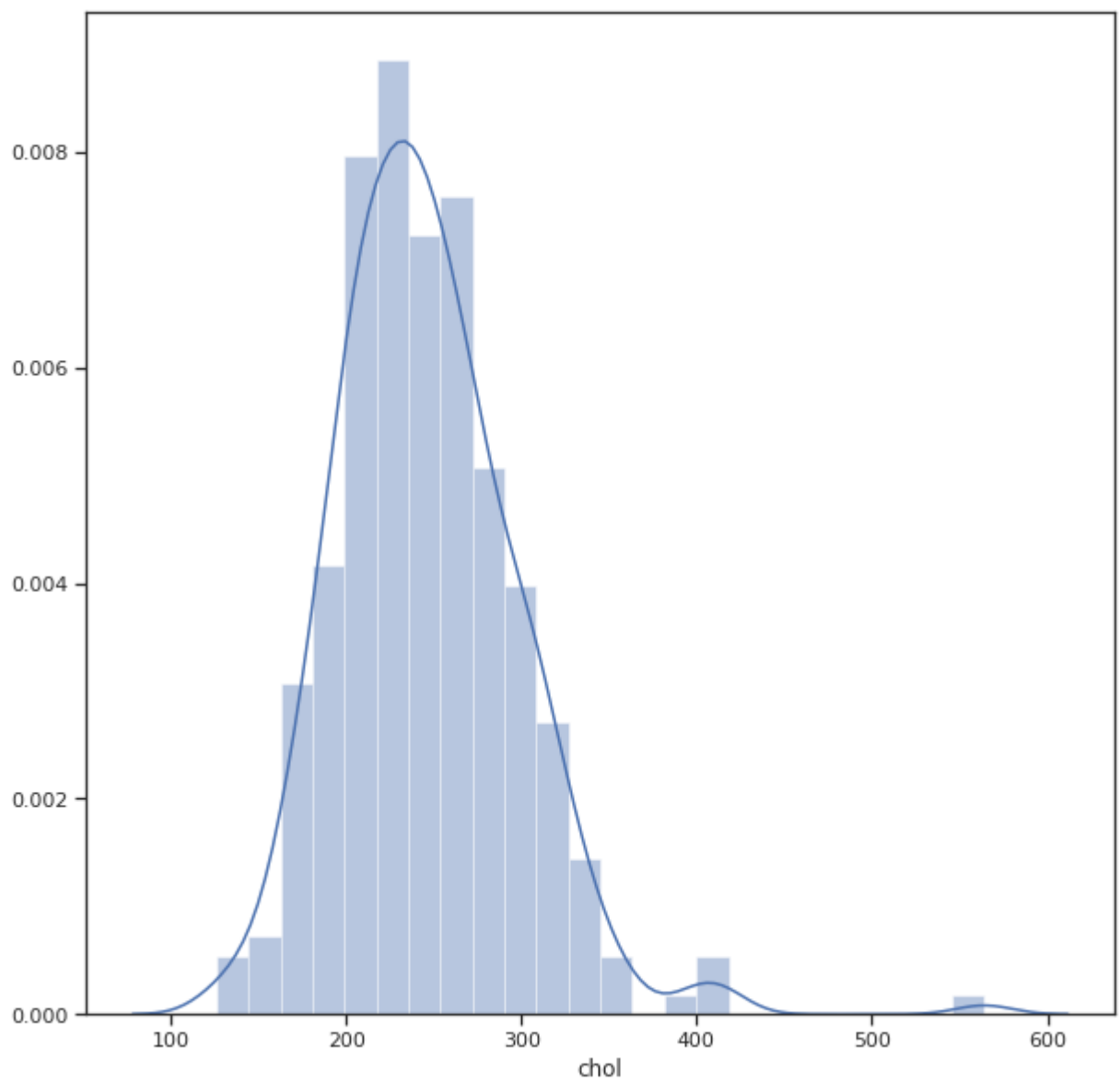
```
[0]: fig, ax = plt.subplots(figsize=(10,10))  
     sns.scatterplot(ax=ax, x='age', y='chol', data=data, hue='target')
```

```
[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d232d1898>
```



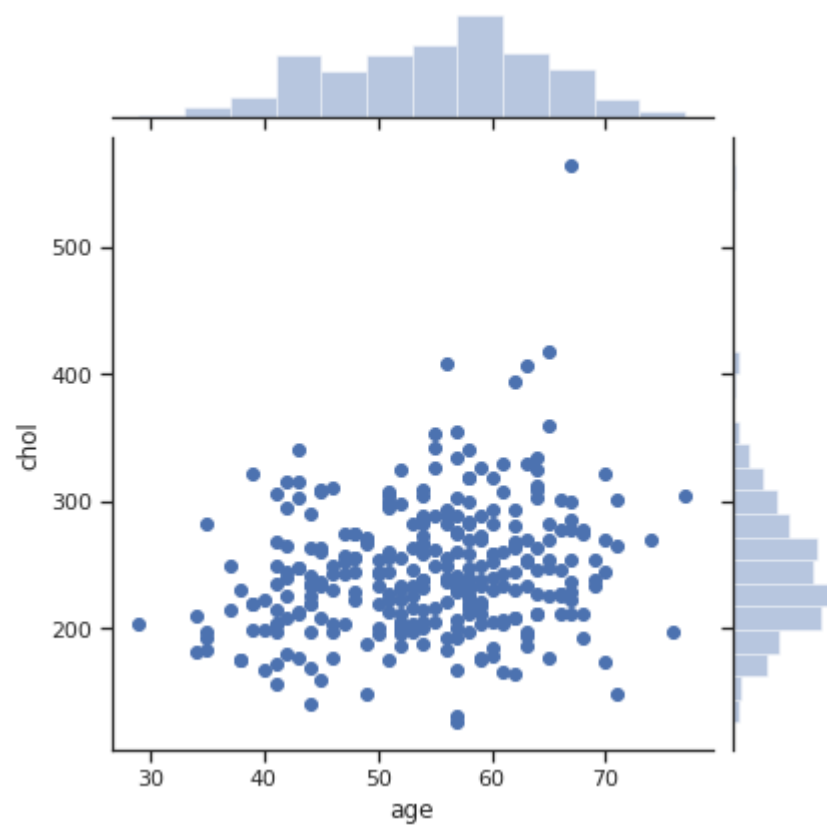
```
[0]: fig, ax = plt.subplots(figsize=(10,10))  
     sns.distplot(data['chol'])
```

```
[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d231da7b8>
```



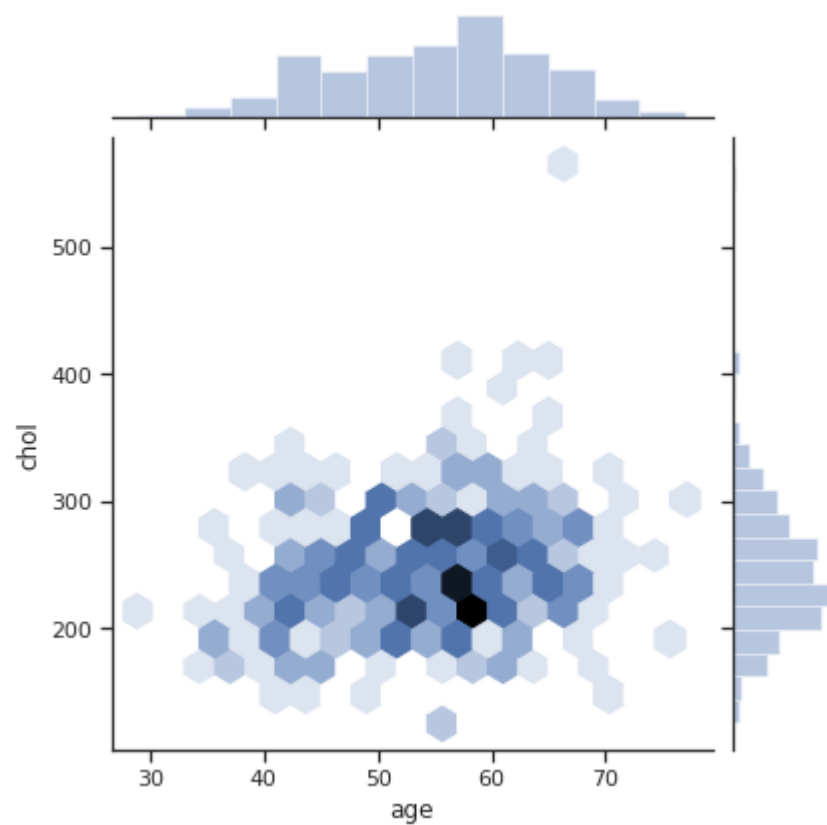
```
[0]: sns.jointplot(x='age', y='chol', data=data)
```

```
[0]: <seaborn.axisgrid.JointGrid at 0x7f7d2316ef60>
```



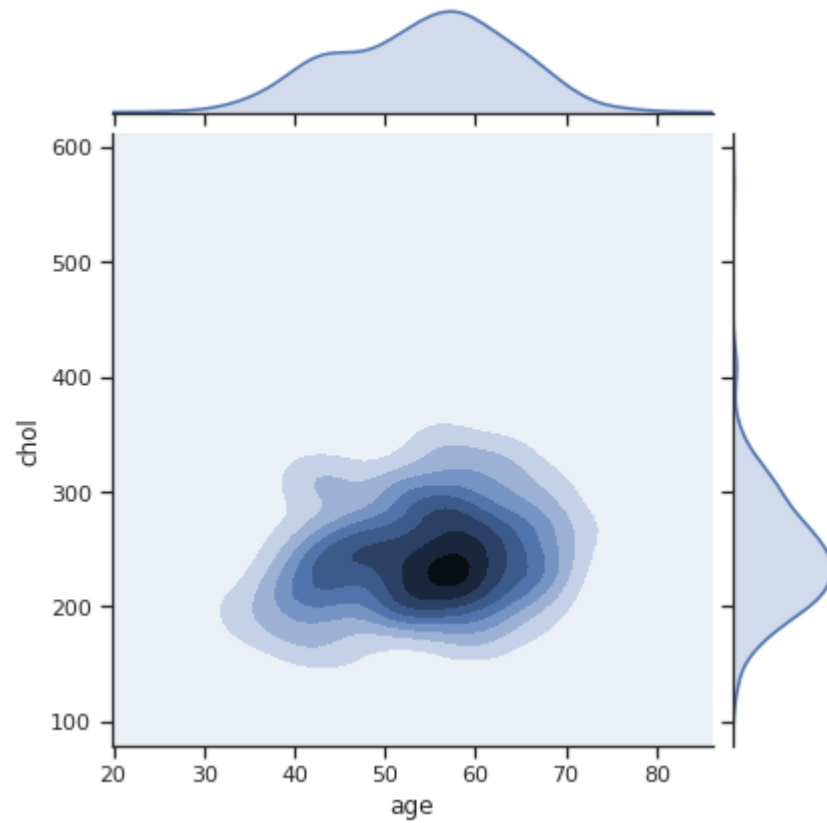
```
[0]: sns.jointplot(x='age', y='chol', data=data, kind="hex")
```

```
[0]: <seaborn.axisgrid.JointGrid at 0x7f7d23038b38>
```



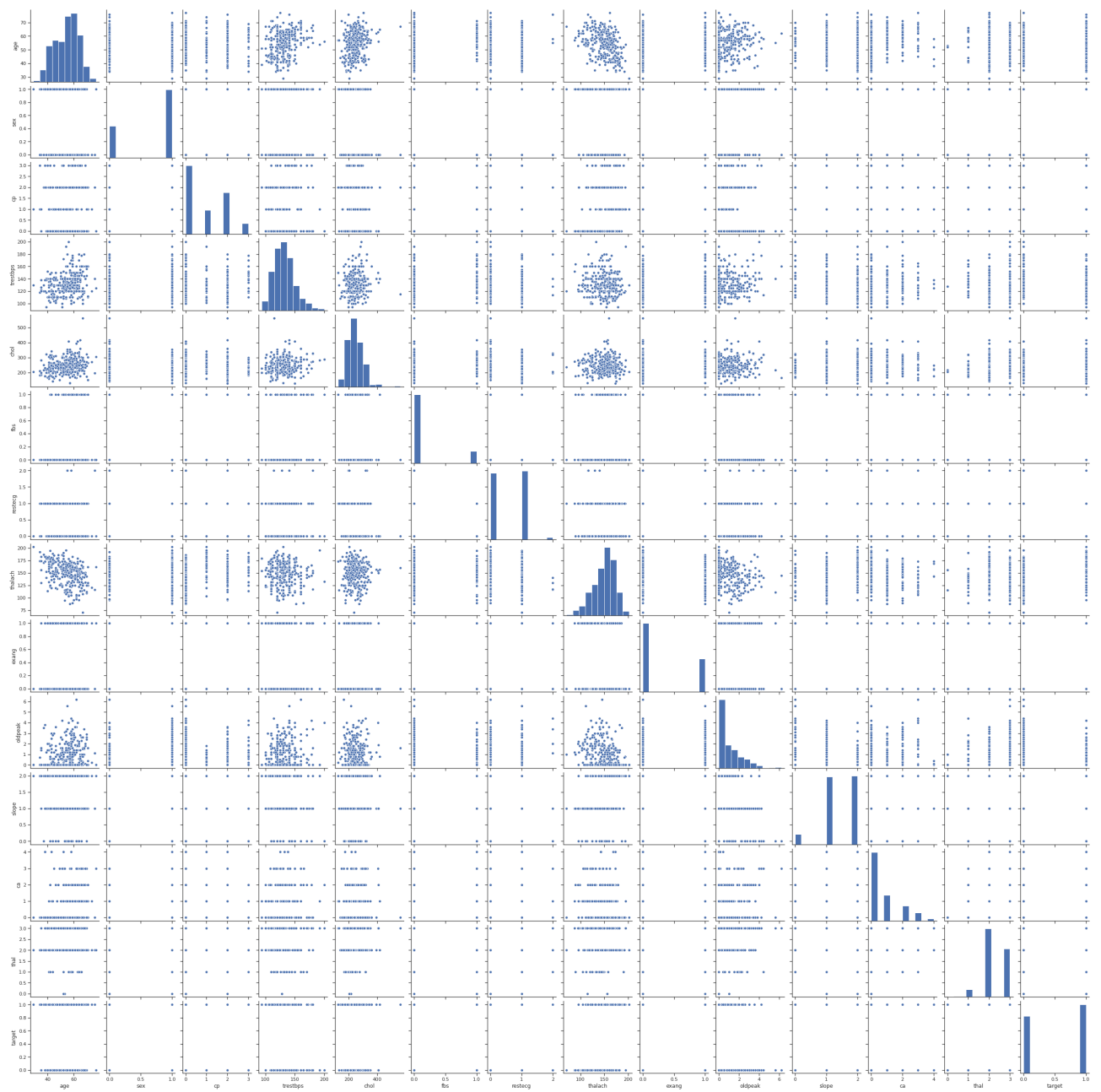
```
[0]: sns.jointplot(x='age', y='chol', data=data, kind="kde")
```

```
[0]: <seaborn.axisgrid.JointGrid at 0x7f7d22edbf60>
```



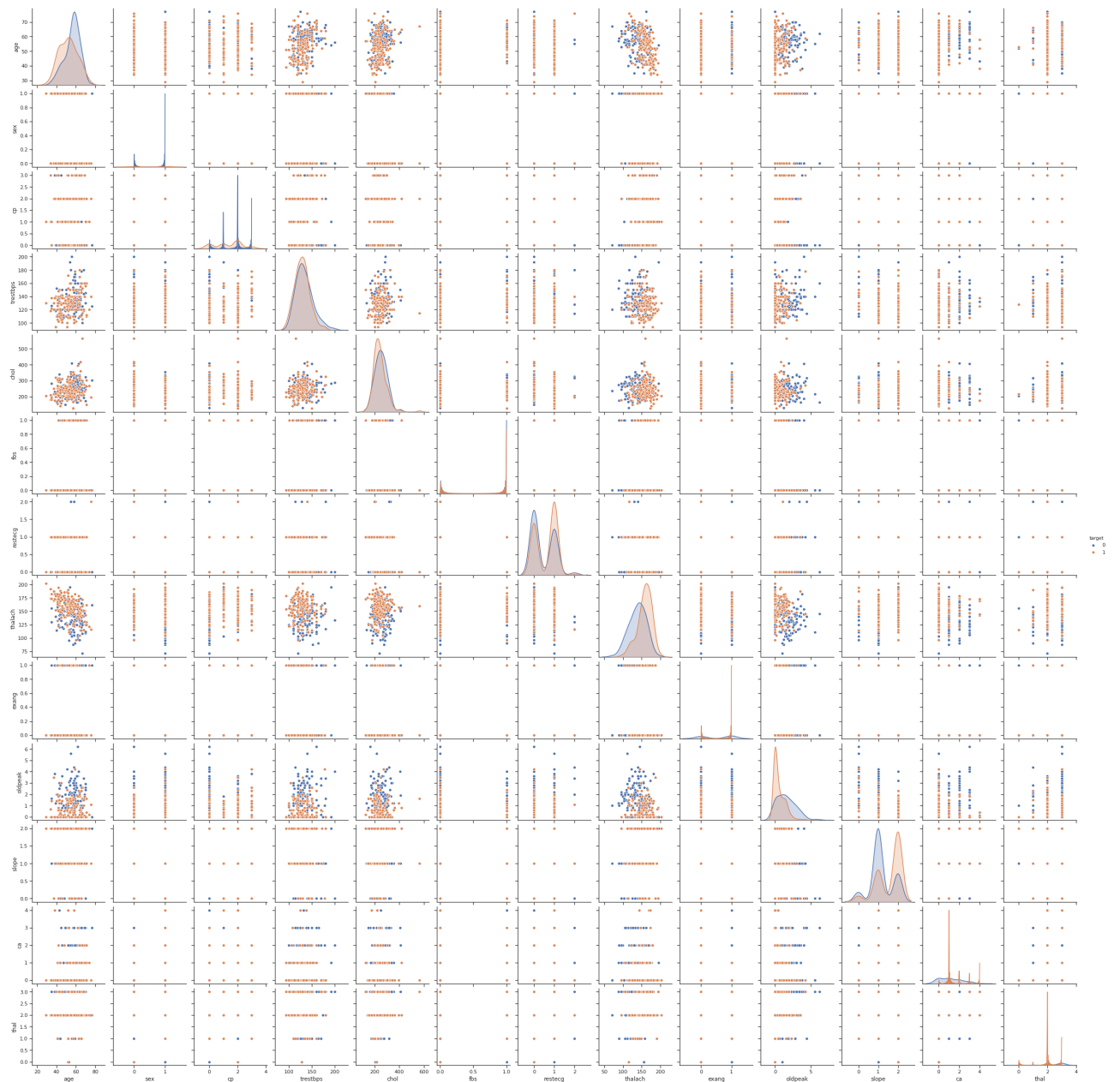
```
[0]: sns.pairplot(data)
```

```
[0]: <seaborn.axisgrid.PairGrid at 0x7f7d22edb550>
```

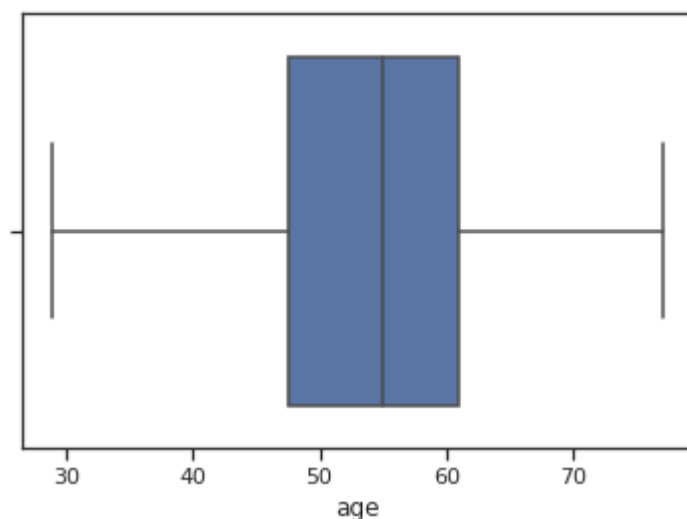
```
[0]: sns.pairplot(data, hue="target")
```

```
[0]: <seaborn.axisgrid.PairGrid at 0x7f7d1eb716a0>
```



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

```
[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d19998438>
```



```
[0]: data.corr()
```

```
[0]:
```

	age	sex	cp	...	ca	thal	target
age	1.000000	-0.098447	-0.068653	...	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	...	0.118261	0.210041	-0.280937
cp	-0.068653	-0.049353	1.000000	...	-0.181053	-0.161736	0.433798
trestbps	0.279351	-0.056769	0.047608	...	0.101389	0.062210	-0.144931
chol	0.213678	-0.197912	-0.076904	...	0.070511	0.098803	-0.085239
fbs	0.121308	0.045032	0.094444	...	0.137979	-0.032019	-0.028046
restecg	-0.116211	-0.058196	0.044421	...	-0.072042	-0.011981	0.137230
thalach	-0.398522	-0.044020	0.295762	...	-0.213177	-0.096439	0.421741
exang	0.096801	0.141664	-0.394280	...	0.115739	0.206754	-0.436757
oldpeak	0.210013	0.096093	-0.149230	...	0.222682	0.210244	-0.430696
slope	-0.168814	-0.030711	0.119717	...	-0.080155	-0.104764	0.345877
ca	0.276326	0.118261	-0.181053	...	1.000000	0.151832	-0.391724
thal	0.068001	0.210041	-0.161736	...	0.151832	1.000000	-0.344029
target	-0.225439	-0.280937	0.433798	...	-0.391724	-0.344029	1.000000

```
[14 rows x 14 columns]
```

target отчасти коррелирует с cp, thalach, exang, oldpeak, slope, ca, thal - их оставляем
target слабо коррелирует с age sex, trestbps, chol, fbs, restecg - их исключить

```
[0]: sns.heatmap(data.corr())
```

```
[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f7d196083c8>
```

