# РТ5-61Б, Забурунов Л. В.

# Технологии Машинного Обучения

# Рубежный Контроль №1

## "Разведочный Анализ Данных"

### 1. Загрузка и анализ структуры набора данных

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns

rk1_data = pd.read_csv("ML_Datasets/RK1/heart.csv")
```

In [3]:

```
rk1_data.shape
```

Out[3]:

```
(303, 14)
```

In [4]:

```
rk1_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Набор данных содержит следующие колонки:

1. Возраст (age);
2. Пол (sex, логическое значение);
3. Категория боли в груди (cp);
4. Кровяное давление в состоянии покоя (trestbps);
5. Холестерол (chol);
6. Уровень сахара в крови (fbs);
7. Результаты ЭКГ (restecg);
8. Максимальный достигнутый пульс (thalach);
9. Стенокардия, вызванная физ. активностью (exang, логическое значение);
10. Уровень подавления физ. активностью участка ЭКГ ST (oldpeak);
11. Наклон участка ЭКГ ST (slope);
12. Число крупных сосудов (ca);
13. ? (thal).

Итак, набор данных успешно загружен, имеет только числовые признаки и не имеет пропусков. Можно без труда переходить к корреляционному анализу.

### 2. Корреляционный анализ набора данных

В первую очередь выведем два основных элемента: таблицу корреляции и тепловую карту на её основе.

In [5]:

```
rk1_data.corr()
```

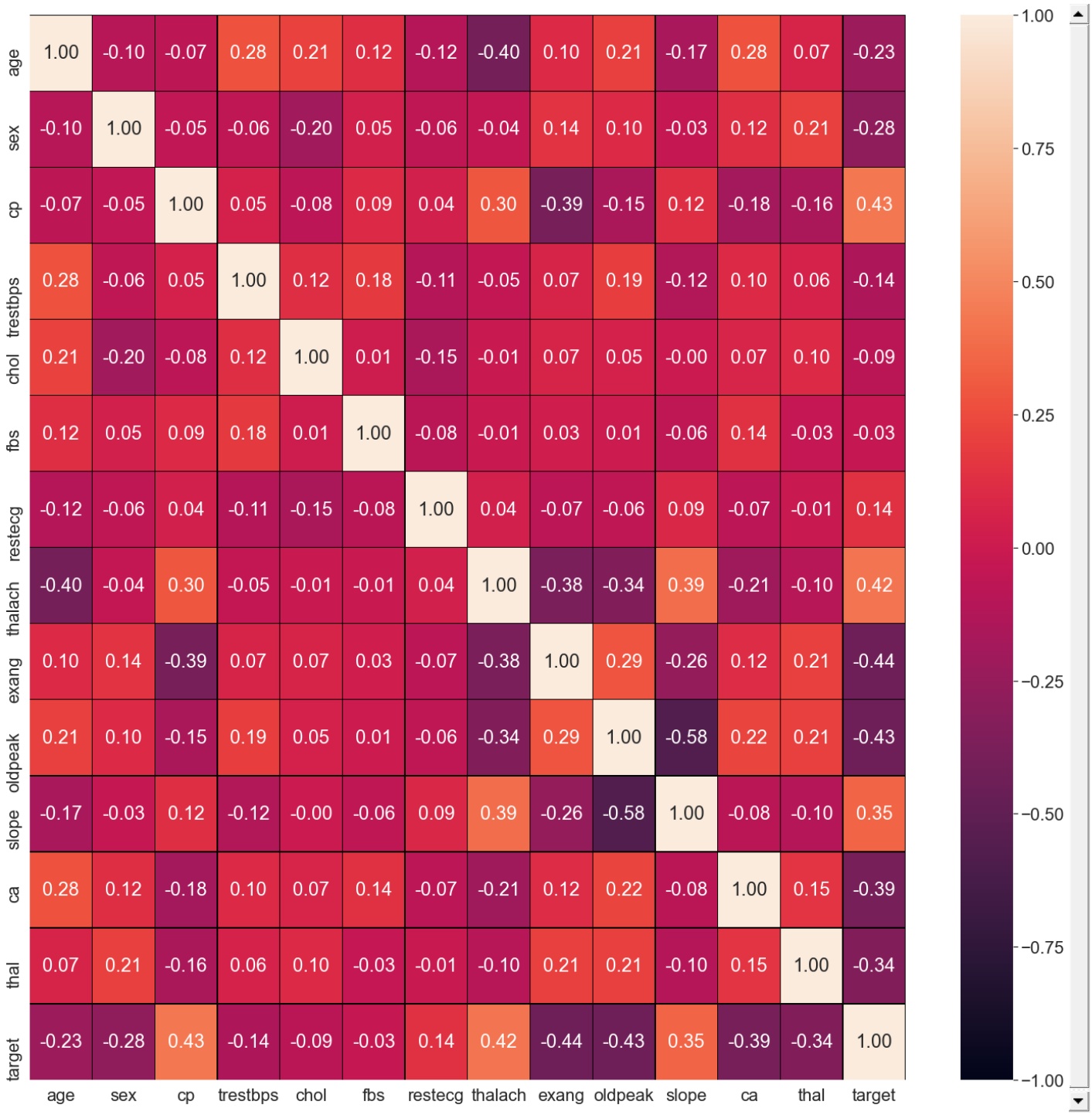
Out[5]:

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **age** | 1.000000 | -0.098447 | -0.068653 | 0.279351 | 0.213678 | 0.121308 | -0.116211 | -0.398522 | 0.096801 | 0.210013 | -0.168814 | 0.276326 | 0.068001 | -0.225439 |
| **sex** | -0.098447 | 1.000000 | -0.049353 | -0.056769 | -0.197912 | 0.045032 | -0.058196 | -0.044020 | 0.141664 | 0.096093 | -0.030711 | 0.118261 | 0.210041 | -0.280937 |
| **cp** | -0.068653 | -0.049353 | 1.000000 | 0.047608 | -0.076904 | 0.094444 | 0.044421 | 0.295762 | -0.394280 | -0.149230 | 0.119717 | -0.181053 | -0.161736 | 0.433798 |
| **trestbps** | 0.279351 | -0.056769 | 0.047608 | 1.000000 | 0.123174 | 0.177531 | -0.114103 | -0.046698 | 0.067616 | 0.193216 | -0.121475 | 0.101389 | 0.062210 | -0.144931 |
| **chol** | 0.213678 | -0.197912 | -0.076904 | 0.123174 | 1.000000 | 0.013294 | -0.151040 | -0.009940 | 0.067023 | 0.053952 | -0.004038 | 0.070511 | 0.098803 | -0.085239 |
| **fbs** | 0.121308 | 0.045032 | 0.094444 | 0.177531 | 0.013294 | 1.000000 | -0.084189 | -0.008567 | 0.025665 | 0.005747 | -0.059894 | 0.137979 | -0.032019 | -0.028046 |
| **restecg** | -0.116211 | -0.058196 | 0.044421 | -0.114103 | -0.151040 | -0.084189 | 1.000000 | 0.044123 | -0.070733 | -0.058770 | 0.093045 | -0.072042 | -0.011981 | 0.137230 |
| **thalach** | -0.398522 | -0.044020 | 0.295762 | -0.046698 | -0.009940 | -0.008567 | 0.044123 | 1.000000 | -0.378812 | -0.344187 | 0.386784 | -0.213177 | -0.096439 | 0.421741 |
| **exang** | 0.096801 | 0.141664 | -0.394280 | 0.067616 | 0.067023 | 0.025665 | -0.070733 | -0.378812 | 1.000000 | 0.288223 | -0.257748 | 0.115739 | 0.206754 | -0.436757 |
| **oldpeak** | 0.210013 | 0.096093 | -0.149230 | 0.193216 | 0.053952 | 0.005747 | -0.058770 | -0.344187 | 0.288223 | 1.000000 | -0.577537 | 0.222682 | 0.210244 | -0.430696 |
| **slope** | -0.168814 | -0.030711 | 0.119717 | -0.121475 | -0.004038 | -0.059894 | 0.093045 | 0.386784 | -0.257748 | -0.577537 | 1.000000 | -0.080155 | -0.104764 | 0.345877 |
| **ca** | 0.276326 | 0.118261 | -0.181053 | 0.101389 | 0.070511 | 0.137979 | -0.072042 | -0.213177 | 0.115739 | 0.222682 | -0.080155 | 1.000000 | 0.151832 | -0.391724 |
| **thal** | 0.068001 | 0.210041 | -0.161736 | 0.062210 | 0.098803 | -0.032019 | -0.011981 | -0.096439 | 0.206754 | 0.210244 | -0.104764 | 0.151832 | 1.000000 | -0.344029 |
| **target** | -0.225439 | -0.280937 | 0.433798 | -0.144931 | -0.085239 | -0.028046 | 0.137230 | 0.421741 | -0.436757 | -0.430696 | 0.345877 | -0.391724 | -0.344029 | 1.000000 |

In [9]:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(25, 25))
sns.set(font_scale = 2)
#mask = np.zeros_like(rk1_data.corr(), dtype=np.bool)
#mask[np.triu_indices_from(mask)] = True
hmap = sns.heatmap(rk1_data.corr(), ax=ax, annot=True, fmt=".2f", linewidths=0.3, linecolor="black", vmin = -1
```




Заметим следующее:

1. Отсутствие пар линейно зависимых признаков. Это значит, что не существует заведомо сбивающих с толку модель МО признаков;
2. Отсутствие линейно зависимых от целевого (или наоборот) признаков. Это значит, что отсуствуют сверхинформативные признаки;
3. Существенные различия в корреляции различных признаков с целевым.

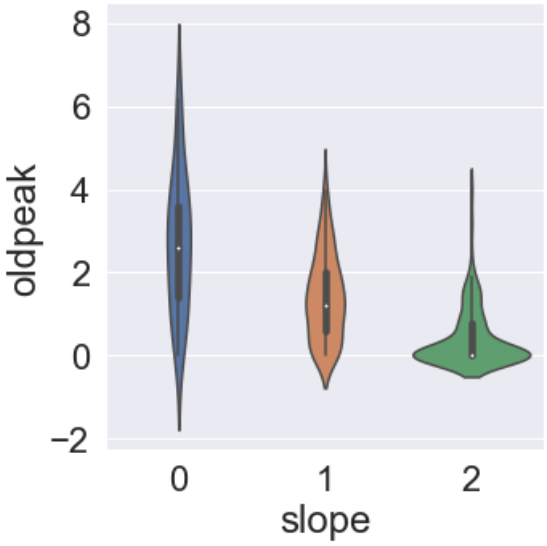
Посмотрим на некоторые пары, отличающиеся наибольшими к-тами линейной корреляции:

In [11]:

```
sns.catplot(data=rk1_data, y="oldpeak", x="slope", kind="violin")
```

Out[11]:

```
<seaborn.axisgrid.FacetGrid at 0x209ef8f0f40>
```

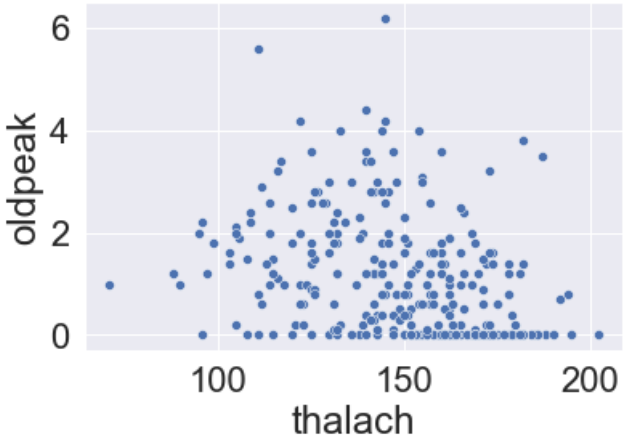



In [10]:

```
sns.scatterplot(data=rk1_data, y="oldpeak", x="thalach")
```

Out[10]:

```
<AxesSubplot:xlabel='thalach', ylabel='oldpeak'>
```

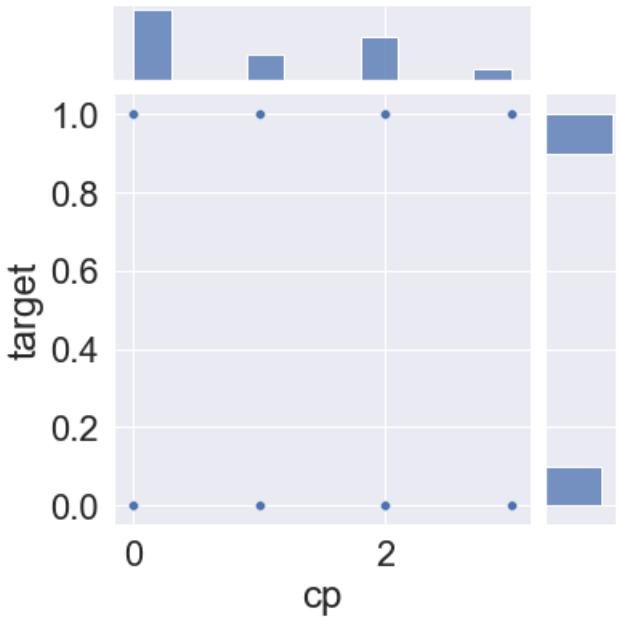



In [30]:

```
sns.jointplot(data=rk1_data, x="cp", y="target")
```

Out[30]:

```
<seaborn.axisgrid.JointGrid at 0x21e6d38cf10>
```

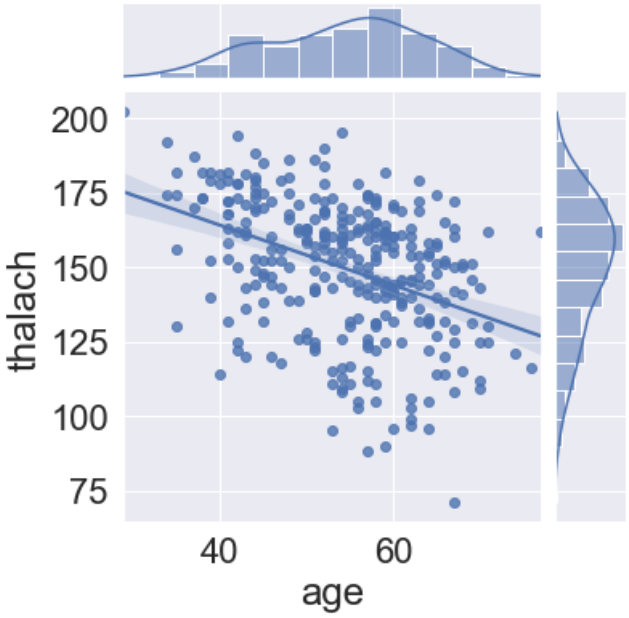



In [37]:

```
sns.jointplot(data=rk1_data, x="age", y="thalach", kind="reg")
```

Out[37]:

```
<seaborn.axisgrid.JointGrid at 0x21e6c53fd00>
```



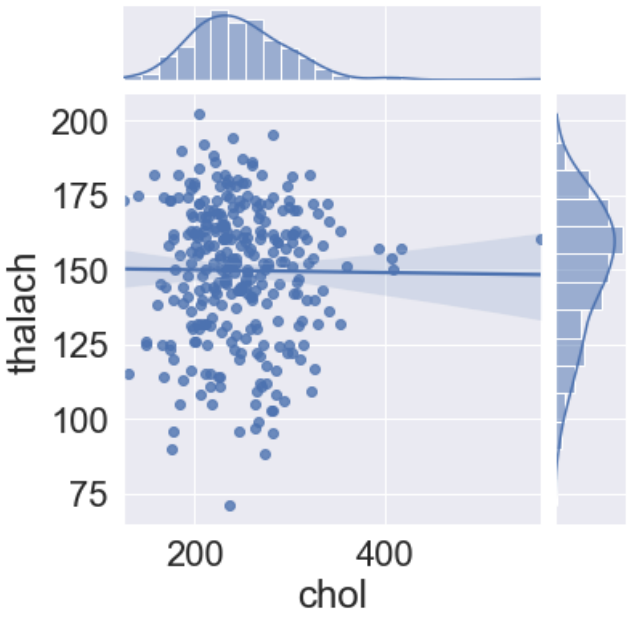

Также построим такие же графики для некоррелирующих между собой признаков:

In [40]:

```
sns.jointplot(data=rk1_data, x="chol", y="thalach", kind="reg")
```

Out[40]:

```
<seaborn.axisgrid.JointGrid at 0x21e6d54d970>
```

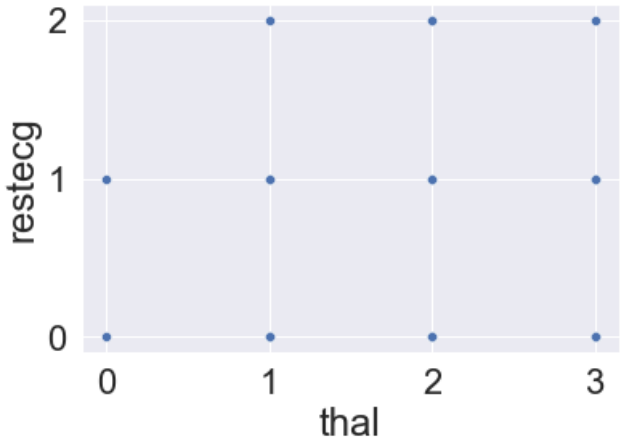



In [42]:

```
sns.scatterplot(data=rk1_data, x="thal", y="restecg")
```

Out[42]:

```
<AxesSubplot:xlabel='thal', ylabel='restecg'>
```

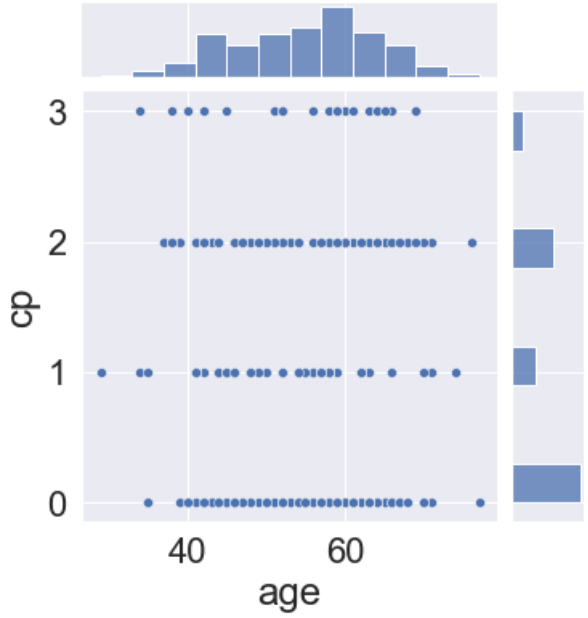



In [44]:

```
sns.jointplot(data=rk1_data, x="age", y="cp")
```

Out[44]:

```
<seaborn.axisgrid.JointGrid at 0x21e6cc18820>
```




In [ ]: