

Основные методы поиска оценок.

Задача 2

Ильичёв А.С., 693

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
```

1. Считаем данные.

```
df = pd.read_csv('Cauchy.csv', names=['coord'])
```

2. Найдем логарифмическую функцию правдоподобия.

В качестве параметра θ выступает x_0 .

$$p(x) = \frac{1}{\pi(1 + (x - x_0)^2)}$$

$$f_\theta(x_1, \dots, x_n) = \prod_{i=1}^n \frac{1}{\pi(1 + (x_i - x_0)^2)}$$

$$L_\theta(x_1, \dots, x_n) = -n \ln \pi - \sum_{i=1}^n \ln(1 + (x_i - x_0)^2)$$

В силу монотонности логарифма для получения оценки максимального правдоподобия достаточно найти $\underset{\theta}{\operatorname{argmax}} L_\theta(x_1, \dots, x_n)$. Первое слагаемое в L_θ не зависит от параметра,

поэтому при максимизации его можно не учитывать. Избавившись от минуса, получим задачу

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^n \ln(1 + (x_i - x_0)^2).$$

3. Оценим параметр сдвига методом максимального правдоподобия.

```
def calc_func(df, x0):
    # считаем минимизируемую функцию
    return np.sum([np.log(1 + (x - x0)**2) for x in df['coord']])
```

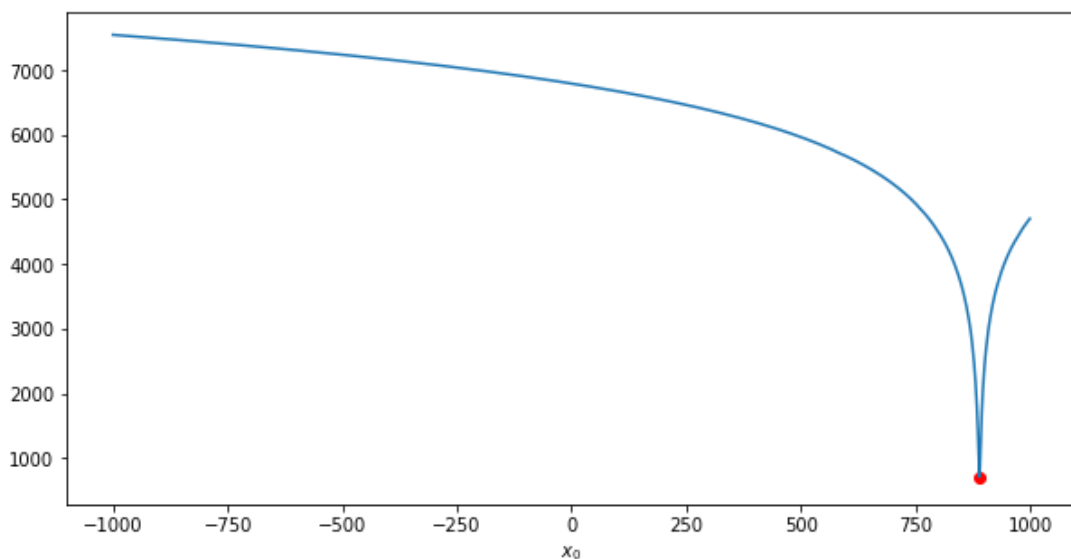
Оценку производим по сетке с шагом 0.01.

```
grid = np.linspace(-1000, 1000, 2000 * 100 + 1)
df500 = df.head(500)
```

```
def find_min(df):
    vals = np.array([calc_func(df, x0) for x0 in grid])
    plt.figure(figsize=(10,5))
    plt.plot(grid, vals)
    min_i = np.argmin(vals)
    plt.scatter([grid[min_i]], [vals[min_i]], c='red')
    plt.xlabel(r'$x_0$')
    plt.show()
    print('min = ', vals[min_i])
    print('Оценка максимального правдоподобия: theta = ', grid[min_i])
```

а) По половине выборки:

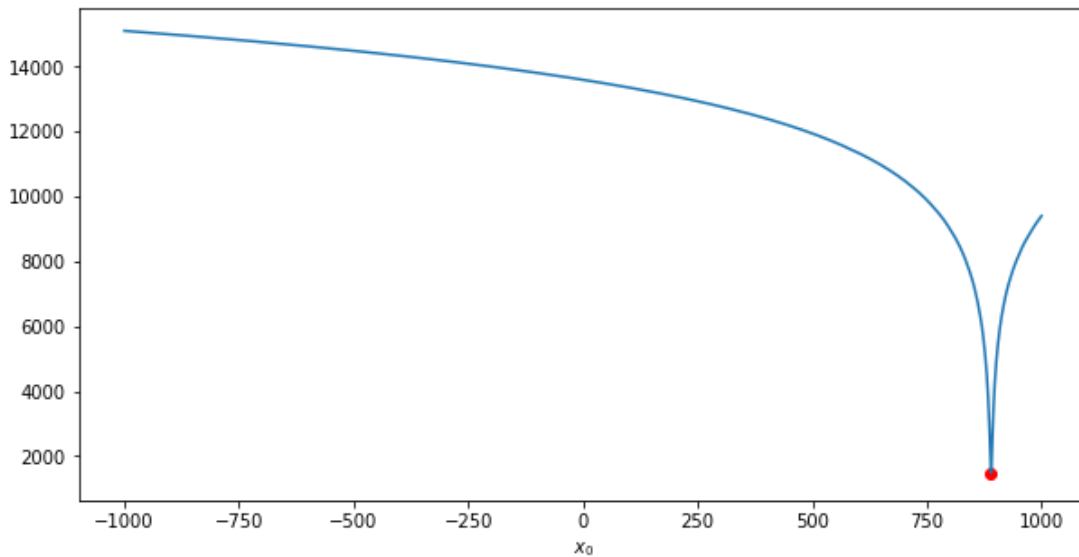
```
find_min(df500)
```



```
min = 708.2698982665836
Оценка максимального правдоподобия: theta = 890.01
```

б) По всей выборке

```
find_min(df)
```



```
min = 1486.632868733386
```

```
Оценка максимального правдоподобия: theta = 889.97
```

Вывод.

Полученные значения оценок отличаются, но не сильно. Так что в данном случае, если высокая точность оценки не требуется, можно рассматривать часть выборки, ускоряя вычисления. Кроме того, в данном случае по сути происходит минимизация среднеквадратичного отклонения, поэтому полученное значение θ близко к среднему значению по выборке:

```
df['coord'].mean()
```

```
890.62409
```