

```
import scipy.stats as sts import numpy as np import matplotlib.pyplot as plt %matplotlib inline
```

Придумайте распределение, у которого конечны первые четыре момента, а пятое — нет. Это распределение с плотностью  $5(x^{-6})I([1, \infty))$ .

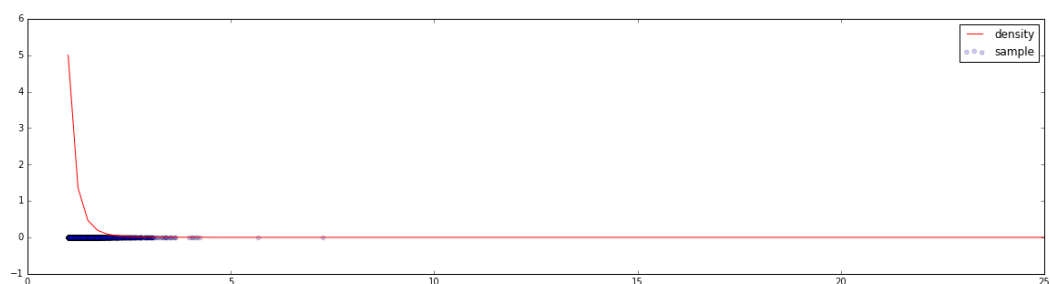
```
In [3]: class mystat(sts.rv_continuous):
        def _pdf(self, x):
            return 5 * (x ** (-6))
        stat = mystat(a=1, name='stat')
```

Сгенерируйте выборку  $X_1, \dots, X_N$  из этого распределения для  $N = 10^4$ .

```
In [6]: samples = stat.rvs(size=10000)
        N = np.size(samples)
```

Постройте график плотности, а также нанесите точки выборки на график (с нулевой у-координатой).

```
In [7]: grid = np.linspace(0, 25, 100)
        plt.figure(figsize=(20, 5))
        plt.xlim([0, 25])
        plt.scatter(samples, np.zeros(N), alpha=0.2, label='sample')
        plt.plot(grid, stat.pdf(grid), color='red', label='density')
        plt.legend()
        plt.show()
```

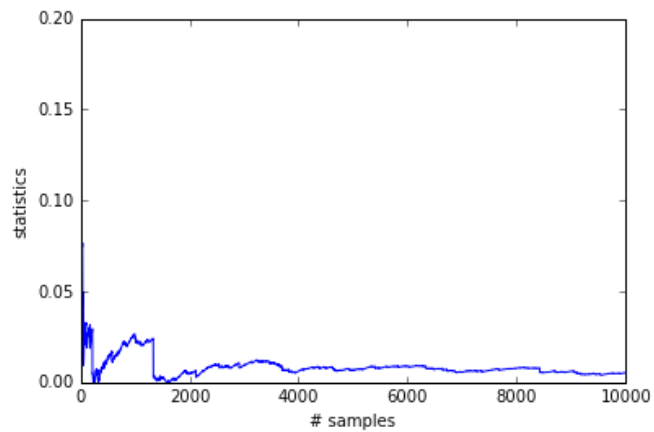


Для всех  $n \leq N$  посчитайте оценку  $S^2 = S^2(X_1, \dots, X_n)$  для дисперсии.

```
In [8]: res = []
        var = stat.var()
        for n in range(1, N + 1):
            res.append(np.abs(np.var(samples[:n]) - var))
```

Постройте график зависимости модуля разности оценки дисперсии и ее истинного значения от  $n$ .

```
In [9]: x = np.arange(N) + 1.  
plt.xlabel("# samples")  
plt.ylabel("statistics")  
plt.ylim([0., 0.2])  
plt.xlim([1., N])  
plt.plot(x, res)  
plt.show()
```

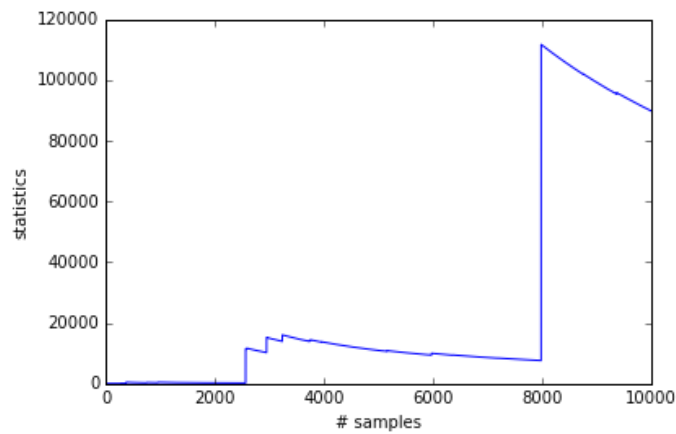
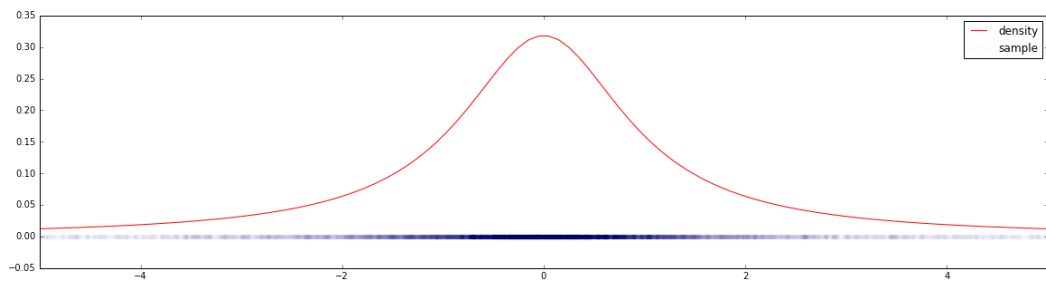


Проведите аналогичное исследование для выборки из распределения Коши, где вместо графика модуля разности оценки дисперсии и ее истинного значения (которого не существует) постройте график оценки дисперсии.

```

In [10]: samples = sts.cauchy.rvs(size=10000)
N = np.size(samples)
grid = np.linspace(-5, 5, 100)
plt.figure(figsize=(20, 5))
plt.xlim([-5, 5])
plt.scatter(samples, np.zeros(N), alpha=0.01, label='sample')
plt.plot(grid, sts.cauchy.pdf(grid), color='red', label='density')
plt.legend()
plt.show()
res = []
var = stat.var()
for n in range(1, N + 1):
    res.append(np.var(samples[:n]))
x = np.arange(N) + 1.
plt.xlabel("# samples")
plt.ylabel("statistics")
plt.plot(x, res)
plt.show()

```



In [ ]: