

Распределения

Распределение	Формула	$E[X]$	$Var(X)$	Оценки
Бернулли	$p^k(1-p)^{1-k}$	p	$p(1-p)$	$p = \bar{X}$ $\theta_1 = \theta_2 = \bar{X}$ $\theta = \bar{X}$
Экспоненциальное	$\begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$	$\theta_k = \sqrt{\frac{k!}{X^k}}$ $\theta = \frac{1}{\bar{X}}$
Равномерное	$\begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a \text{ или } x > b \end{cases}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$	$a = 2\bar{X} - b$
Пуассона	$\frac{\lambda^k e^{-\lambda}}{k!}$	λ	λ	$\lambda = \bar{X}$
Биномиальное	$\binom{n}{k} p^k (1-p)^{n-k}$	np	$np(1-p)$	$p = \frac{\bar{X}}{n}$ $\theta = 1 - \frac{S^2}{\bar{x}}$ $\theta = \frac{\bar{X}}{m}$
Нормальное		a	σ^2	$a = \bar{X}$
Геометрическое	$(1-p)^{k-1} p$	$\frac{1}{p}$	$\frac{1-p}{p^2}$	$p = \frac{1}{\bar{X}}$ $\theta_1 = \frac{1}{\bar{X}}$ $\theta_2 = \frac{-1 + \sqrt{1 + 8\bar{X}^2}}{2\bar{X}^2}$ $\theta = \frac{1}{\bar{X}}$

Табл. 1. Виды распределений и их параметры.

Построение гистограммы

```
import numpy as np
import matplotlib.pyplot as plt

def main() -> None:
    data = np.array([v for v in open("data.csv", 'r')])

    plt.figure()

    plt.hist(data, bins = 10, edgecolor='black', justify=True)

    plt.show() # показать гистограмму

a, b = np.histogram(data, bins=10)
print(a) # количество элементов в каждом промежутке
```

```
if __name__ == "__main__":  
    main()
```

Нахождение параметров выборки

```
import numpy as np  
  
def main() -> None:  
  
    data = np.array([v for v in open("data.csv", 'r')])  
  
    print(data.mean())  
    print(data.var(ddof=1)) # поделенное на (n - 1) (несмещенное)  
    print(data.var(ddof=0)) # смещенное  
  
if __name__ == "__main__":  
    main()
```

Нормальное/Гауссово распределение

```
np.random.normal(loc, scale, size)  
  
# loc - (mean) where the peak of the bell exists.  
# scale - (standard deviation) how flat the graph distribution should be.  
# size - the shape of the returned array.
```

Биномиальное распределение

```
np.random.binomial(n, p, size)  
  
# n - number of trials.  
# p - probability of occurrence of each trial (e.g. for toss of a coin 0.5 each).  
# size - the shape of the returned array.
```

Распределение Пуассона

```
np.random.poisson(lam, size)  
  
# lam - rate or known number of occurrences e.g. 2 for above problem.  
# size - the shape of the returned array.
```

Равномерное распределение

```
np.random.uniform(low, high, size)  
  
# low - lower bound - default 0.0  
# high - upper bound - default 1.0  
# size - The shape of the returned array
```

Экспоненциальное/показательное распределение

```
np.random.exponential(scale, size)
```

```
# scale - inverse of rate ( see lam in poisson distribution ) defaults to 1.0  
# size - The shape of the returned array
```

Геометрическое распределение

```
np.random.geometric(p, size)
```

```
# p (float): Probability of success ( $0 < p \leq 1$ )  
# size (int or tuple, optional): output shape. if None, returns a single value.
```