

Домашняя работа по Математической Статистике

Митин Арсений (3 курс, СКБ171)

26.10.2019

Содержание

1	Задание №1	3
1.1	Выбор распределений	3
1.2	Описание основных характеристик распределений	3
1.3	Примеры событий и интерпретации	13
1.4	Способы моделирования случайных величин	14

1. Задание №1

1.1. Выбор распределений

Выбранные распределения:

- Дискретное: *гипергеометрическое*
- Непрерывное: *нормальное*

1.2. Описание основных характеристик распределений

Гипергеометрическое распределение

Гипергеометрическое распределение - дискретное распределение, описывающее вероятность события, при котором ровно k из n случайно выбранных элементов окажутся *помеченными*, при этом выборка осуществляется из множества мощности N , в котором присутствует m помеченных элементов. Считается, что каждый из элементов может быть выбран с одинаковой вероятностью $\frac{1}{N}$. Запишем это формально:

$$N \in \mathbb{N}, m \in \overline{0, N}, n \in \overline{0, N}, \\ k \in \overline{0, n}$$

Тогда $HG(N, m, n)$ описывает вероятность события, при котором ровно k из n элементов выборки окажутся *помеченными*:

$$\{ \xi \sim HG(N, m, n) \} \iff \left\{ \mathbb{P}(\xi = k) = \frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}} \right\} \quad (1)$$

Математическое ожидание

По определению, математическое ожидание случайной величины -- это ее 1^й начальный момент. Для начала, найдем $k^{\text{й}}$ начальный момент для ξ (это понадобится для дальнейших выводов):

$$\mathbb{E}[\xi^r] = \sum_{k=0}^n k^r \cdot \mathbb{P}(\xi = k) = \sum_{k=0}^n k^r \frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}} \quad (2)$$

Можем считать, что сумма берется при $k = \overline{1, n}$, так как слагаемое при $k = 0$ будет равно 0. Заметим, что

$$\begin{aligned} k \binom{m}{k} &= k \frac{m!}{k!(m-k)!} = \\ &= k \frac{m \cdot (m-1)!}{k \cdot (k-1)! \cdot (m-k)!} = \\ &= m \frac{(m-1)!}{(k-1)! \cdot (m-1-(k-1))!} = \\ &= m \binom{m-1}{k-1} \end{aligned} \quad (3)$$

и, как следствие,

$$\binom{N}{n} = \frac{1}{n} \cdot n \cdot \binom{N}{n} = \frac{1}{n} N \binom{N-1}{n-1} \quad (4)$$

Подставим 3 и 4 в 2:

$$\mathbb{E}[\xi^r] = \frac{n \cdot m}{N} \sum_{k=1}^{r-1} \frac{\binom{m-1}{k-1} \binom{N-m}{n-k}}{\binom{N-1}{n-1}}$$

Положим $j := k-1$ и изменим индекс суммирования на $j = \overline{0, n-1}$. Заметим, что $n-k = n-(j+1) = (n-1)-j$ и $N-m = (N-1) - (m-1)$:

$$\mathbb{E}[\xi^r] = \frac{n \cdot m}{N} \sum_{j=0}^{n-1} (j+1)^{r-1} \frac{\binom{m-1}{j} \binom{(N-1)-(m-1)}{(n-1)-j}}{\binom{N-1}{n-1}}$$

Заметим, что выделенная часть выражения может быть записана, как $\mathbb{E}[(\theta+1)^{r-1}]$, где $\theta \sim HG(N-1, m-1, n-1)$. Следовательно,

$$\mathbb{E}[\xi^r] = \frac{n \cdot m}{N} \mathbb{E}[(\theta+1)^{r-1}] \quad (5)$$

Таким образом,

$$\boxed{\mathbb{E}[\xi] = \frac{n \cdot m}{N}} \quad (6)$$

Дисперсия

По определению дисперсии,

$$\text{Var}[\xi] = \mathbb{E}[(\xi - \mathbb{E}[\xi])^2] = \mathbb{E}[\xi^2] - (\mathbb{E}[\xi])^2 \quad (7)$$

Выведем 2^й начальный момент из 5:

$$\mathbb{E}[\xi^2] = \frac{n \cdot m}{N} \mathbb{E}[\theta+1] = \frac{n \cdot m}{N} \left(\frac{(n-1)(m-1)}{N-1} + 1 \right) \quad (8)$$

Подставим 6 и 8 в 7:

$$\begin{aligned} \text{Var}[\xi] &= \mathbb{E}[\xi^2] - (\mathbb{E}[\xi])^2 = \\ &= \frac{n \cdot m}{N} \left(\frac{(n-1)(m-1)}{N-1} + 1 \right) - \left(\frac{n \cdot m}{N} \right)^2 = \\ &= \frac{n \cdot m}{N} \left(\frac{(n-1)(m-1)}{N-1} + 1 - \frac{n \cdot m}{N} \right) \end{aligned}$$

Таким образом,

$$\boxed{\text{Var}[\xi] = \frac{n \cdot m}{N} \left(\frac{(n-1)(m-1)}{N-1} + 1 - \frac{n \cdot m}{N} \right)}$$

Производящая функция моментов

По определению, производящая функция моментов $M_\xi(t)$ для случайной величины ξ -- это математическое ожидание новой случайной величины $e^{t\xi}$. То есть:

$$M_\xi(t) = \mathbb{E}[e^{t\xi}] \quad (9)$$

Для $\xi \sim HG(N, m, n)$ производящая функция **выглядит** так:

$$M_{\xi}(t) = \frac{\binom{N-D}{n}}{\binom{N}{n}} {}_2F_1(-n, -D; N - D - n + 1; e^t) \quad (10)$$

Здесь ${}_2F_1$ - это **гипергеометрическая функция**, определенная следующим образом:

$${}_2F_1(a, b; c; z) = \sum_{n=0}^{\infty} \frac{a^{(n)} b^{(n)}}{c^{(n)}} \frac{z^n}{n!} \quad (11)$$

, а $x^{(n)}$ - **возрастающий факториал**, определенный как:

$$x^{(n)} = \prod_{k=0}^{n-1} (x + k)$$

Характеристическая функция

По определению, характеристическая функция случайной величины ξ задается следующим образом:

$$\varphi_{\xi}(t) = \mathbb{E} [e^{it\xi}]$$

Для $\xi \sim HG(N, m, n)$ характеристическая функция **выглядит** так:

$$M_{\xi}(t) = \frac{\binom{N-D}{n}}{\binom{N}{n}} {}_2F_1(-n, -D; N - D - n + 1; e^{it})$$

Здесь ${}_2F_1$ - это гипергеометрическая функция (11).

Гистограмма вероятностей

Гистограмма -- это графическое представление функции, приближающей плотность вероятности распределения на основе выборки из него.

Чтобы построить гистограмму, сначала нужно разбить множество значений выборки на несколько отрезков. Чаще всего, берут отрезки одинаковой длины, чтобы облегчить восприятие получившегося результата, однако это необязательно. Далее подсчитывается количество вхождений элементов выборки в каждый из отрезков и рисуются прямоугольники, по площади пропорциональные количеству попавших элементов выборки в соответствующий отрезок.

Вообще говоря, гистограмму можно использовать не только для приближения плотности на основе выборки, но и для визуализации самой плотности распределения, зная его плотность.

Мы будем строить гистограмму вероятностей, писать будем на языке **Python3** и использовать следующие библиотеки:

- **NumPy** для работы с массивами
- **SciPy** для комбинаторных и статистических функций
- **Plotly** для визуализации

Итак, для начала, определим класс гипергеометрического распределения `hg`, который будет содержать в себе

информацию о параметрах N , m и n и предоставлять метод $p(k)$, возвращающий вероятность принятия случайной величиной значения k при данных параметрах:

```

1 import scipy as sp
2
3 class HG(object):
4     def __init__(self, N: int, m: int, n: int):
5         self.N = N
6         self.m = m
7         self.n = n
8
9     def p(self, k: int) → float:
10         return sp.special.comb(self.m, k) * sp.special.comb(self.N-self.m, self.n-k) / sp.special.comb(self.N, self.n)
11
12     def __str__(self) → str:
13         return f'HG({self.N}, {self.m}, {self.n})'
```

Далее создадим объект случайной величины $\xi \sim HG(30, 15, 20)$:

```

14 xi = HG(30, 15, 20)
```

Следующим шагом, определим интервал $\overline{0, n}$, на котором мы будем рисовать нашу гистограмму:

```

15 import numpy as np
16
17 hist_data_x = np.arange(xi.n+1)
```

И, наконец, построим гистограмму и выведем ее:

```

18 import plotly
19 import plotly.graph_objs as go
20 hg_hist_fig = go.Figure(
21     data=(go.Scatter(
22         x=list(hist_data_x),
23         y=list(map(xi.p, hist_data_x)),
24         mode='markers',
25     )),
26     layout=go.Layout(
27         title=go.layout.Title(
28             text=r'$\xi \sim ' + str(xi) + '$',
29             x=.5,
30         ),
31         yaxis=go.layout.YAxis(title=go.layout.yaxis.Title(
32             text=r'$\mathbb{P}(\xi=k)$',
33         )),
34         xaxis=go.layout.XAxis(title=go.layout.xaxis.Title(
35             text=r'$k$',
36         )),
37     ),
38 )
39 plotly.offline.iplot(hg_hist_fig)
```

Получившаяся гистограмма имеет следующий вид:

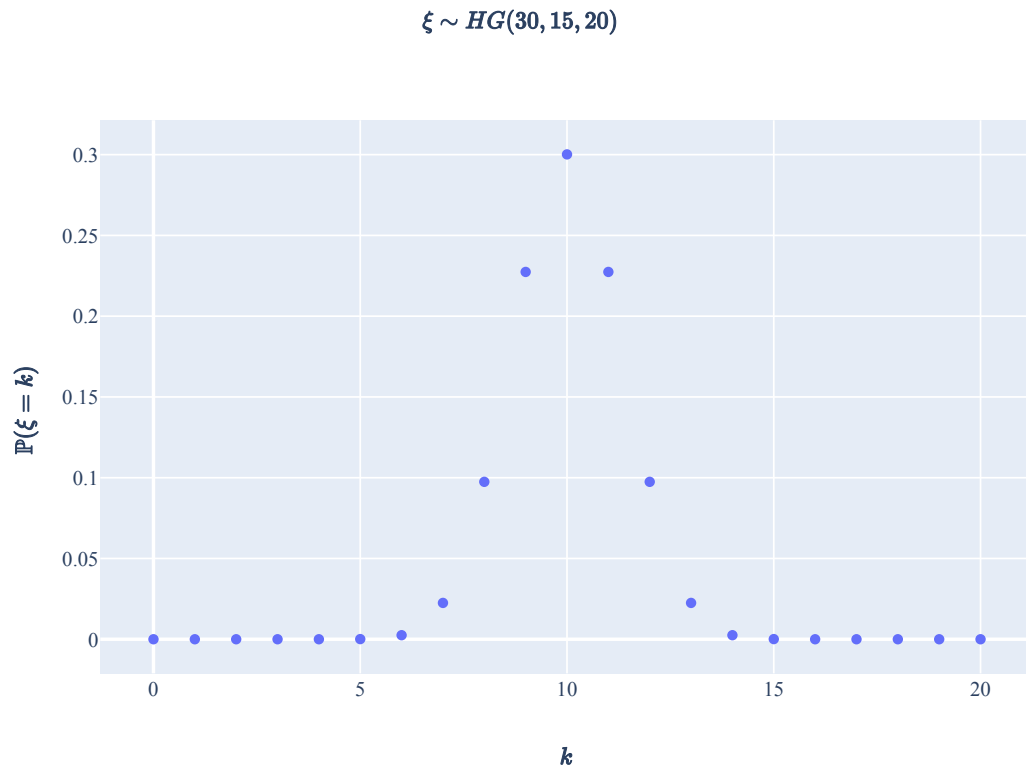


Рис. 1: Гистограмма вероятностей гипергеометрического распределения

Функция распределения

По определению, функция распределения $F_{\xi}(k) = \mathbb{P}(\xi < k)$. Для дискретной случайной величины событие $\{\xi < k\} = \bigcup_{i=0}^{k-1} \{\xi = i\}$. Каждое из событий $\{\xi = i\} \forall i \in \overline{0, k-1}$ являются попарно несовместными. То есть $\forall i, j \in \overline{0, k-1} : i \neq j$ выполняется $\{\xi = i\} \cap \{\xi = j\} = \emptyset$. Из этого следует, что

$$\mathbb{P}(\xi < k) = \sum_{i=0}^{k-1} \mathbb{P}(\xi = i)$$

Подставим 1 в это выражение и получим:

$$F_{\xi}(k) = \sum_{i=0}^{k-1} \mathbb{P}(\xi = i) = \sum_{i=0}^{k-1} \frac{\binom{m}{i} \binom{N-m}{n-i}}{\binom{N}{n}}$$

Построим график этой функции, учитывая, что аргументом k должно быть натуральное число, не превосходящее n :

```

40 n_dist_fig = go.Figure(
41     data=(go.Scatter(
42         x=list(n_data_x),
43         y=list(map(eta.p, n_data_x)),

```

```

44 ),),
45 layout=go.Layout(
46     title=go.layout.Title(
47         text=r'\eta \sim ' + str(eta) + '$',
48         x=.5,
49     ),
50     yaxis=go.layout.YAxis(
51         title=go.layout.YAxis.Title(
52             text=r'\mathbb{P}(\eta < x)$',
53         ),
54     ),
55     xaxis=go.layout.XAxis(
56         title=go.layout.XAxis.Title(
57             text=r'$x$',
58         ),
59     ),
60     # paper_bgcolor='rgba(0,0,0,0)',
61 ),
62 )
63 plotly.offline.iplot(n_dist_fig)

```

График будет следующим:

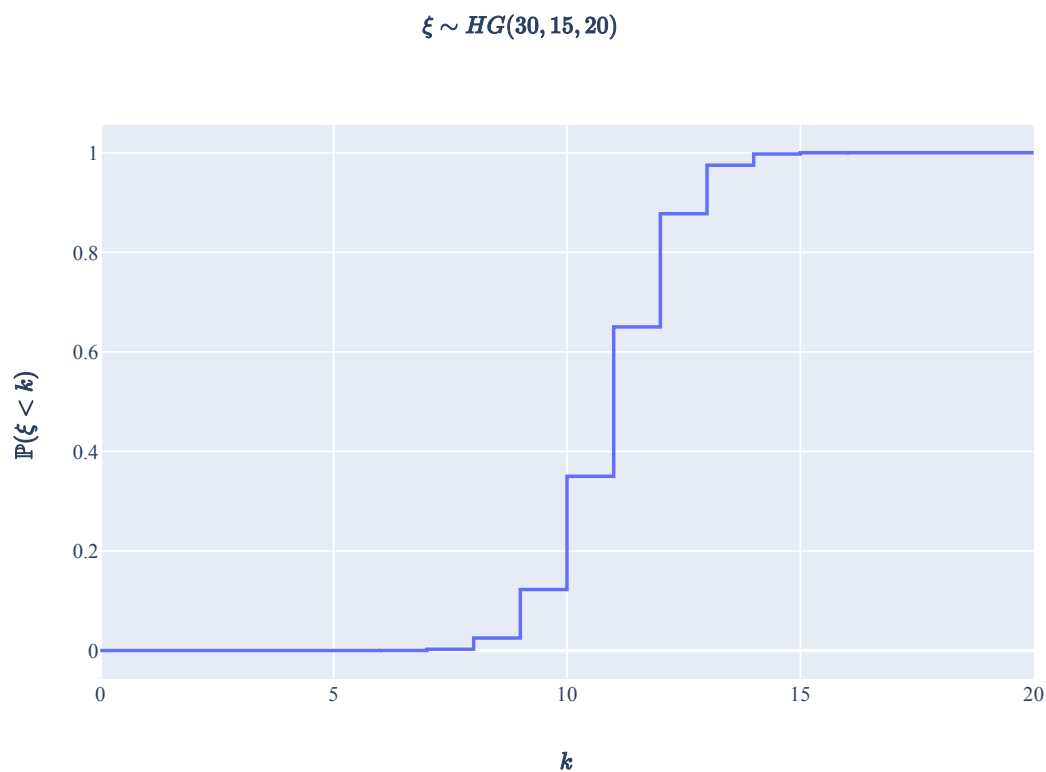


Рис. 2: Функция распределения гипергеометрического распределения

Нормальное распределение

Нормальное распределение - непрерывное распределение, описывающее поведение величины отклонения измеряемого значения x от истинного значения μ (которое является математическим ожиданием) и в рамках некоторого разброса σ (среднеквадратичного отклонения). Запишем это формально:

$$\{\eta \sim N(\mu, \sigma^2)\} \Leftrightarrow \left\{ \begin{array}{l} F_\eta(x) = \mathbb{P}(\eta < x) = \int_{-\infty}^x f_\eta(x) dx, \\ \text{где } f_\eta(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} - \text{плотность вероятности} \end{array} \right\} \quad (12)$$

Математическое ожидание

Найдем математическое ожидание $\eta \sim N(\mu, \sigma^2)$:

$$\begin{aligned} \mathbb{E}[\eta] &= \int_{-\infty}^{+\infty} x \cdot f_\eta(x) dx = \\ &= \int_{-\infty}^{+\infty} x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} x e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \end{aligned}$$

Сделаем замену $t = \frac{x-\mu}{\sqrt{2}\sigma}$:

$$\begin{aligned} \mathbb{E}[\eta] &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} (\sigma\sqrt{2}t + \mu) e^{-t^2} d\left(\frac{x-\mu}{\sqrt{2}\sigma}\right) = \\ &= \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \int_{-\infty}^{+\infty} t e^{-t^2} dt + \frac{\mu}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-t^2} dt = \\ &= \frac{\sigma\sqrt{2}}{\sqrt{\pi}} \left(\int_{-\infty}^0 t e^{-t^2} dt - \int_0^{+\infty} t e^{-t^2} dt \right) + \frac{\mu}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-t^2} dt = \\ &= \frac{\mu}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-t^2} dt \end{aligned}$$

Заметим, что получившееся выражение содержит интеграл, который может быть сведен к интегралу [Эйлера-Пуассона](#):

$$\int_{-\infty}^{+\infty} e^{-t^2} dt = 2 \int_0^{+\infty} e^{-t^2} dt = \sqrt{\pi}$$

Таким образом,

$$\boxed{\mathbb{E}[\eta] = \mu} \quad (13)$$

Дисперсия

Подставим 13 в определение дисперсии 7:

$$\begin{aligned}\text{Var}[\eta] &= \mathbb{E}[(\eta - \mu)^2] = \\ &= \int_{-\infty}^{+\infty} (x - \mu)^2 \cdot f_{\eta}(x) dx = \\ &= \int_{-\infty}^{+\infty} (x - \mu)^2 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} (x - \mu)^2 e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx\end{aligned}$$

Сделаем ту же замену переменной $t = \frac{x-\mu}{\sqrt{2}\sigma}$, тогда $x = t\sqrt{2}\sigma + \mu$ и:

$$\begin{aligned}\text{Var}[\eta] &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} (\sqrt{2}\sigma)^2 t^2 e^{-t^2} d(t\sqrt{2}\sigma + \mu) = \\ &= \frac{2\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{+\infty} t^2 e^{-t^2} dt\end{aligned}$$

Проинтегрируем по частям:

$$\begin{aligned}\text{Var}[\eta] &= \frac{\sigma^2}{\sqrt{\pi}} \int_{-\infty}^{+\infty} t 2t e^{-t^2} dt = \\ &= \frac{\sigma^2}{\sqrt{\pi}} \left(-te^{-t^2} \Big|_{-\infty}^{+\infty} + \int_{-\infty}^{+\infty} e^{-t^2} dt \right)\end{aligned}$$

Здесь снова появляется интеграл [Эйлера-Пуассона](#) и, в итоге, получаем:

$$\boxed{\text{Var}[\eta] = \sigma^2}$$

То есть, σ является среднеквадратичным отклонением.

Производящая функция моментов

Производящая функция моментов для $\eta \sim N(\mu, \sigma^2)$ имеет вид:

$$M_{\eta}(t) = \exp\left(\mu t + \frac{\sigma^2 t^2}{2}\right)$$

Характеристическая функция

Характеристическая функция для $\eta \sim N(\mu, \sigma^2)$ имеет вид:

$$\varphi_{\eta}(t) = \exp\left(\mu it + \frac{\sigma^2 t^2}{2}\right)$$

Плотность вероятности

Определим класс нормального распределения `н`, который будет содержать в себе информацию о параметрах μ и σ и предоставлять следующие методы:

- `f(x)` - возвращает значение плотности в точке `x`

- $p(k)$ - возвращает $\mathbb{P}(\eta < x) = \int_{-\infty}^x f_{\eta}(x)dx$

```

1 class N(object):
2     def __init__(self, mu: float, sigma: float):
3         self.mu = mu
4         self.sigma = sigma
5     def f(self, x: float) → float:
6         return np.exp(-(x-self.mu)**2)/(2*self.sigma**2)/(self.sigma*(2*np.pi)**.5)
7     def p(self, x: float) → float:
8         return sp.integrate.quad(self.f, -np.inf, x)[0]
9     def __str__(self):
10        return f'N({self.mu}, {self.sigma}^2)'

```

Далее создадим объект случайной величины $\xi \sim HG(30, 15, 20)$:

```
11 eta = N(0, 1)
```

Следующим шагом, руководствуясь [правилом трех сигм](#) определим интервал $(-3\sigma, 3\sigma)$:

```
12 n_data_x = np.linspace(-3*eta.sigma, 3*eta.sigma, 100)
```

И, наконец, построим плотность, используя метод `n.f` нашего класса:

```

13 n_dens_fig = go.Figure(
14     data=(go.Scatter(
15         x=list(n_data_x),
16         y=list(map(eta.f, n_data_x)),
17     )),
18     layout=go.Layout(
19         title=go.layout.Title(
20             text=r'$\eta \sim ' + str(eta) + '$',
21             x=.5,
22         ),
23         yaxis=go.layout.YAxis(
24             title=go.layout.YAxis.Title(
25                 text=r'$f_{\eta}(x)$',
26             ),
27         ),
28         xaxis=go.layout.XAxis(
29             title=go.layout.XAxis.Title(
30                 text=r'$x$',
31             ),
32         ),
33         # paper_bgcolor='rgba(0,0,0,0)',
34     ),
35 )
36 plotly.offline.iplot(n_dens_fig)

```

Получившийся график имеет следующий вид:

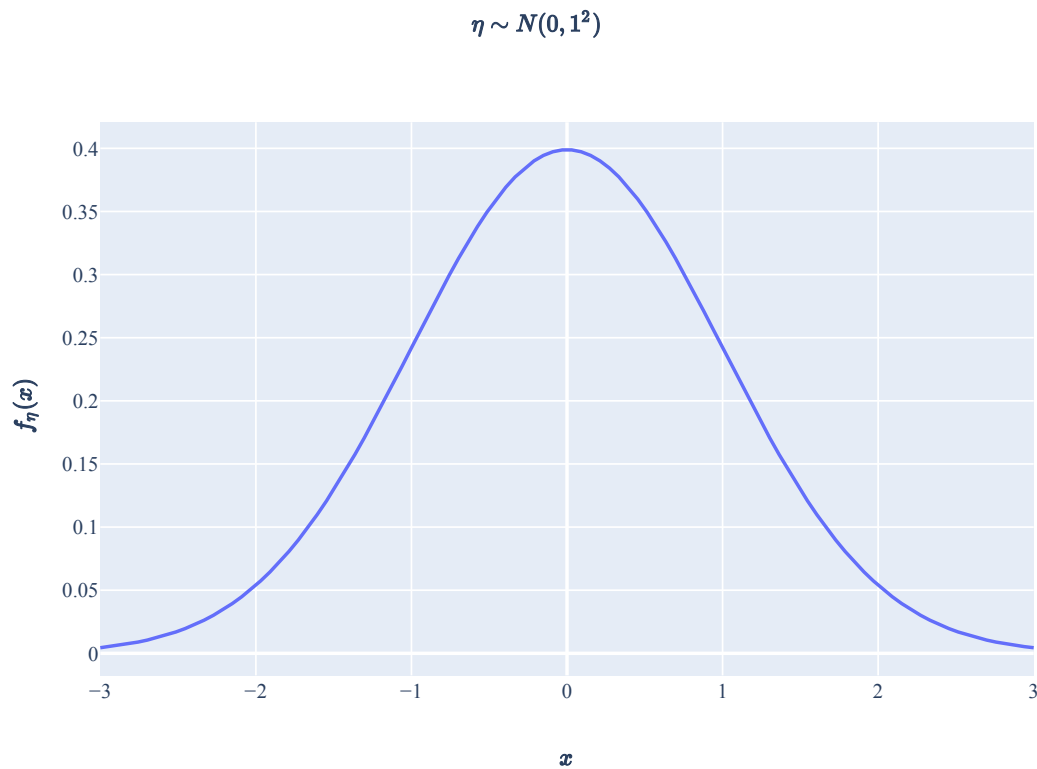


Рис. 3: Плотность нормального распределения

Функция распределения

Для построения функции распределения, будем использовать метод n.p.:

```

37 n_dist_fig = go.Figure(
38     data=(
39         go.Scatter(
40             x=list(n_data_x),
41             y=list(map(eta.p, n_data_x)),
42         ),
43     ),
44     layout=go.Layout(
45         title=go.layout.Title(
46             text=r'$\eta \sim ' + str(eta) + '$',
47             x=.5,
48         ),
49         yaxis=go.layout.YAxis(
50             title=go.layout.YAxis.Title(
51                 text=r'$\mathbb{P}(\eta < x)$',
52             ),
53         ),
54         xaxis=go.layout.XAxis(
55             title=go.layout.XAxis.Title(
56                 text=r'$x$',
57             ),
58         ),
59         # paper_bgcolor='rgba(0,0,0,0)',
60     ),

```

```
61 | )
62 | plotly.offline.iplot(n_dist_fig)
```

Функция распределения имеет вид:

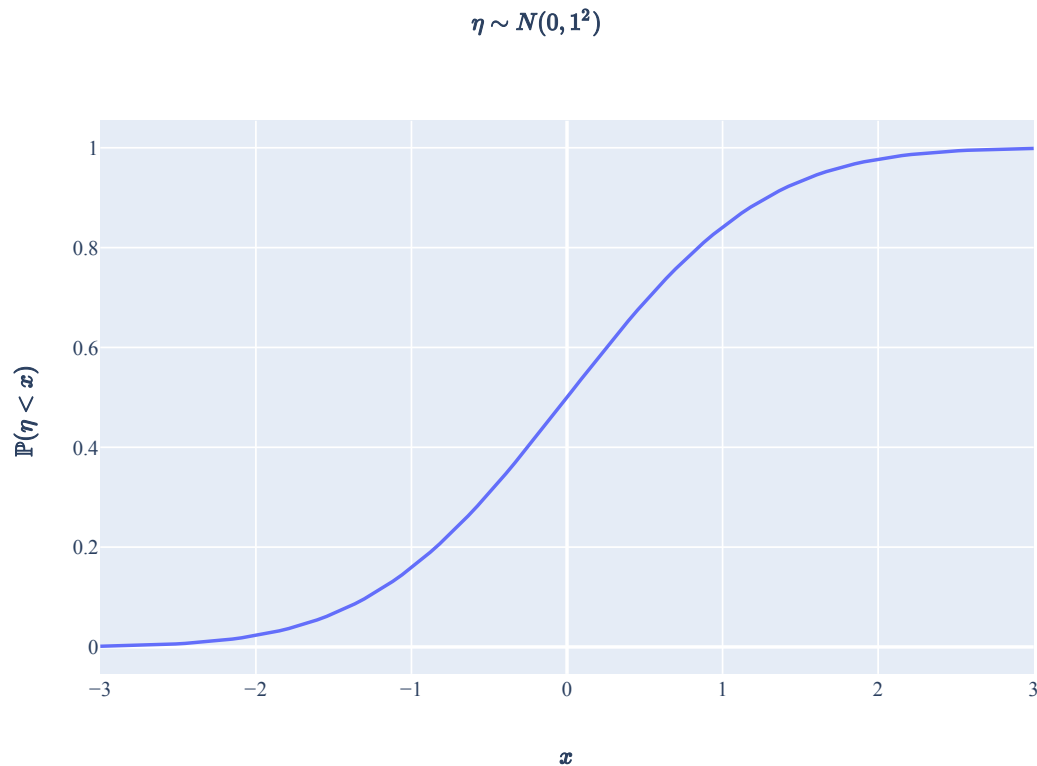


Рис. 4: Функция распределения нормального распределения

1.3. Примеры событий и интерпретации

Гипергеометрическое распределение

Типичная интерпретация

Типичной интерпретацией гипергеометрического распределения является выборка без возвращения из множества элементов, некоторые из которых являются помеченными. Представим, что в нашем распоряжении имеется корзина, наполненная шарами двух цветов: черные и белые. Причём всего в корзине находится N шаров, m из которых -- белые. Шары в корзине тщательно перемешиваются, чтобы каждый из них мог быть вытащен с одинаковой вероятностью $\frac{1}{N}$. Далее случайно вытаскиваются n шаров без возвращения. Гипергеометрическое распределение описывает вероятность того, что среди вытаскиваемых шаров ровно k окажутся белыми.

Действительно, всего существует $\binom{N}{n}$ выборок размера n , $\binom{m}{k}$ способов выбрать k помеченных объектов (белых шаров), и $\binom{N-m}{n-k}$ способов заполнить оставшиеся $n - k$ слотов непомеченными объектами (черными шарами). Таким образом, вероятность того, что среди n вытаскиваемых объектов окажется ровно k помеченных, будет равна

$$\frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}}.$$

Известные соотношения между распределениями

- Если зафиксировать размер выборки и количество помеченных элементов, а мощность множества, из которого ведется выборка, устремить к бесконечности, то гипергеометрическое распределение будет сходиться к биномиальному:

$$HG(N, m, n) \xrightarrow{N \rightarrow \infty} Bi\left(n, \frac{m}{N}\right)$$

Нормальное распределение

Типичная интерпретация

Нормальное распределение описывает нормированную случайную величину, которая является суммой многих случайных слабо взаимосвязанных величин, каждая из которых вносит малый вклад относительно общей суммы. Это вытекает из [центральной предельной теоремы](#).

Известные соотношения между распределениями

- Сумма двух независимых случайных величин, имеющих нормальное распределение, имеет [распределение Коши](#):

$$\begin{aligned}\xi &\sim N(\mu_1, \sigma_1^2) \\ \eta &\sim N(\mu_2, \sigma_2^2) \\ \xi + \eta &\sim C(\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2})\end{aligned}$$

- Сумма квадратов k независимых стандартных нормальных случайных величин имеет [распределение \$\chi^2\$](#) с k степенями свободы:

$$\begin{aligned}\forall i \in \overline{1, k} \quad \xi_i &\sim N(0, 1) \\ \sum_{i=1}^k \xi_i^2 &\sim \chi^2(k)\end{aligned}$$

- Натуральный логарифм [логнормального распределения](#) имеет нормальное распределение:

$$\begin{aligned}\xi &\sim \text{Log}N(\mu, \sigma^2) \\ \ln \xi &\sim N(\mu, \sigma^2)\end{aligned}$$

1.4. Способы моделирования случайных величин

Для моделирования выбранных случайных величин, будем использовать библиотеку [SciPy](#).

Гипергеометрическое распределение

Для моделирования гипергеометрического распределения будем использовать класс `scipy.stats.hypergeom`:

1 | `hg = sp.stats.hypergeom(xi.N, xi.m, xi.n)`

Проверим, что `hg.pmf` возвращает правильные значения для вероятности. Для этого построим с помощью этой функции гистограмму и наложим ее на ту, которую мы получали ранее:

```

1 hg_model_fig = go.Figure(
2     data=(
3         go.Scatter(
4             name=f'model',
5             x=list(hg_data_x),
6             y=hg.pmf(hg_data_x),
7             mode='markers',
8             marker_size=10,
9         ),
10        hg_hist_plot,
11    ),
12    layout=go.Layout(
13        title=go.layout.Title(
14            text=r'$\xi \sim $ ' + str(xi) + '$',
15            x=.5,
16        ),
17        yaxis=go.layout.YAxis(title=go.layout.yaxis.Title(
18            text=r'$\mathbb{P}(\xi=k)$',
19        )),
20        xaxis=go.layout.XAxis(title=go.layout.xaxis.Title(
21            text=r'$k$',
22        )),
23    ),
24 )
25 plotly.offline.iplot(hg_model_fig)

```

График получится следующий:

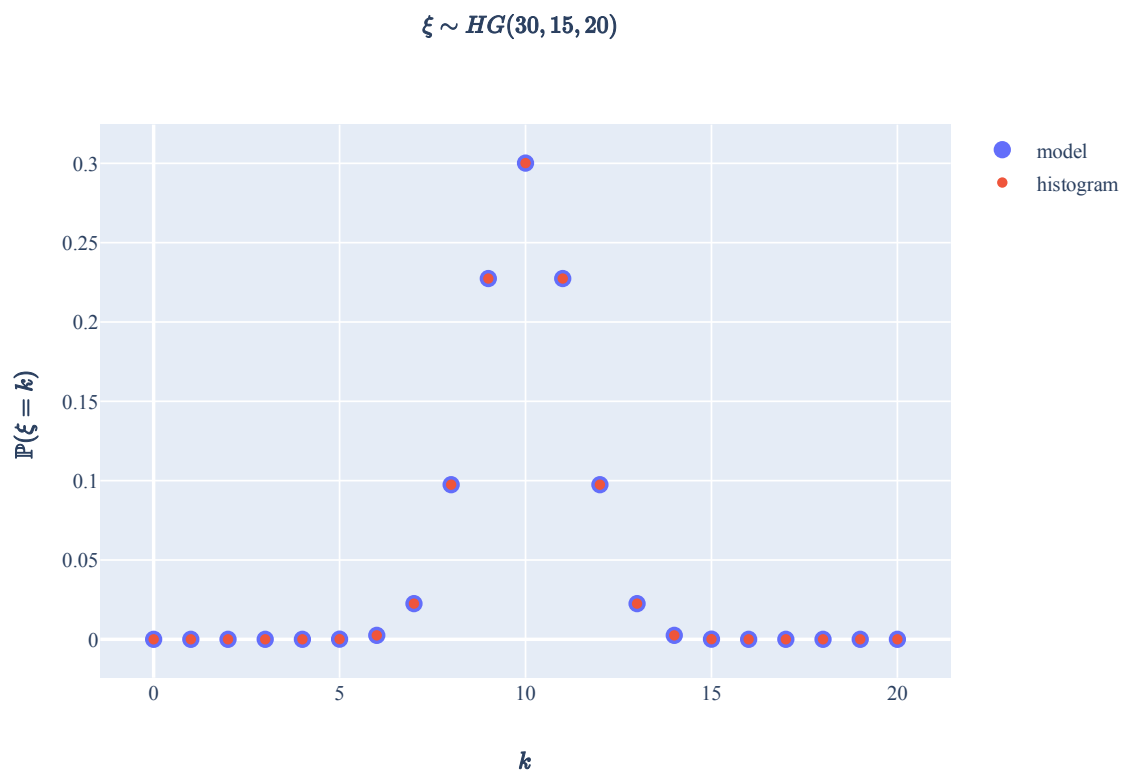


Рис. 5: Проверка функции генерации выборки из гипергеометрического распределения с помощью SciPy

Нормальное распределение

Для моделирования нормального распределения будем использовать класс `scipy.stats.norm`:

```
1 | n = sp.stats.norm(eta.mu,eta.sigma)
```

Проверим, что `n.pdf` возвращает правильные значения для вероятности. Для этого построим с помощью этой функции плотность и наложим ее на ту, которую мы получали ранее:

```
1 | n_model_fig = go.Figure(  
2 |     data=(  
3 |         go.Scatter(  
4 |             name=f'model',  
5 |             x=list(n_data_x),  
6 |             y=n.pdf(n_data_x),  
7 |             line_width=8,  
8 |         ),  
9 |         n_dens_plot,  
10 |     ),  
11 |     layout=go.Layout(  
12 |         title=go.layout.Title(  
13 |             text=r'$\eta$ \sim ' + str(eta) + '$',  
14 |             x=.5,  
15 |         ),  
16 |         yaxis=go.layout.YAxis(title=go.layout.yaxis.Title(  
17 |             text=r'$f_\eta(x)$',  
18 |         )),  
19 |         xaxis=go.layout.XAxis(title=go.layout.xaxis.Title(  
20 |             text=r'$x$',  
21 |         )),  
22 |     ),  
23 | )  
24 | plotly.offline.iplot(n_model_fig)
```

График получится следующий:

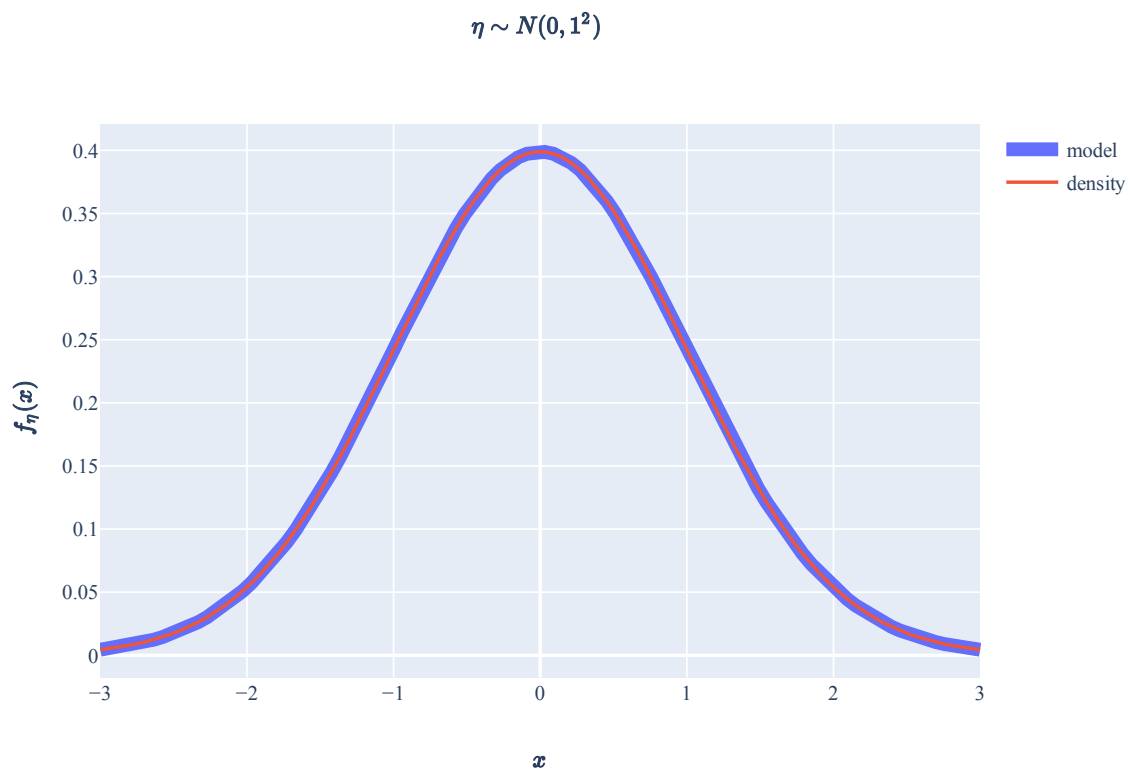


Рис. 6: Проверка функции генерации выборки из нормального распределения с помощью SciPy