Теория вероятностей и математическая статистика[¶](#gjdgxs)

## Урок 4[¶](#30j0zll)

## Непрерывные случайные величины. Функция распределения и функция плотности. Нормальное распределение. Центральная предельная теорема[¶](#1fob9te)

### Разбор домашнего задания[¶](#3znysh7)

**Задача 1**

Коробки с шоколадом упаковываются автоматически. Их средняя масса равна 1.06 кг. Известно, что 5% коробок имеют массу, меньшую 1 кг. Найдите: а) среднее квадратическое отклонение массы коробки, б) процент коробок, имеющих массу больше 1.1 кг.

*Подсказка*. 1. Можно считать, что распределение массы коробки нормальное. 2. Найдите такое значение scale, для которого значение cdf(x=1, loc=1.06, scale=scale) близко к 0.05. Точности 0.0001 будет достаточно.

**Решение**

Нам нужно найти такое значение scale, что верны неравенства:

* cdf(x=1, loc=1.06, scale=scale) < 0.05 + 0.0001
* cdf(x=1, loc=1.06, scale=scale) > 0.05 - 0.0001

Реализуем для поиска метод Ньютона. Он заключается в итеративном делении отрезка пополам, пока не будет найдено значение с нужной точностью. Он хорошо подходит для поиска конкретного значения монотонной функции.

In [1]:

**import** numpy **as** np  
**from** scipy **import** stats

In [2]:

loc **=** 1.06

Метод Ньютона:

In [3]:

start **=** 0  
end **=** 1  
  
target **=** 0.05  
err **=** 0.0001  
  
print('cdf\tscale')  
print('----------')  
  
**while** **True**:  
 scale **=** (start **+** end) **/** 2  
 cdf **=** stats**.**norm**.**cdf(1, loc**=**loc, scale**=**scale)  
   
 print(round(cdf, 4), '\t', round(scale, 4))  
   
 **if** cdf **<=** target **-** err:  
 start **=** scale  
 **elif** cdf **>=** target **+** err:  
 end **=** scale  
 **else**:  
 **break**  
   
scale

cdf scale  
----------  
0.4522 0.5  
0.4052 0.25  
0.3156 0.125  
0.1685 0.0625  
0.0274 0.0312  
0.1003 0.0469  
0.0623 0.0391  
0.0439 0.0352  
0.053 0.0371  
0.0484 0.0361  
0.0507 0.0366  
0.0495 0.0364  
0.0501 0.0365  
0.0498 0.0364  
0.05 0.0365

Out[3]:

0.036468505859375

Проверка:

In [4]:

stats**.**norm**.**cdf(1, loc**=**loc, scale**=**scale)

Out[4]:

0.049958594078015874

Тепер, когда мы знаем оба параметра нашего распределения, можно найти вероятность того, что коробка будет иметь массу более 1.1 кг. Это можно сделать с помощью функции распределения: $$p = 1 - F(1.1)$$

In [5]:

1 **-** stats**.**norm**.**cdf(1.1, loc**=**loc, scale**=**scale)

Out[5]:

0.1363563656878326

**Задача 2**

Коробка содержит 30 конфет. Известно, что масса каждой конфеты распределена равномерно в промежутке от 12 до 14 граммов. Используя центральную предельную теорему, найти вероятность, что масса всей коробки будет: а) меньше 390 граммов, б) больше 395 граммов, в) от 380 до 400 граммов.

Массой самой коробки можно пренебречь.

*Подсказка*. Для равномерного распределения в промежутке [a, b]:

1. мат. ожидание равно (a + b) / 2,
2. дисперсия равна (b - a)^2 / 12.

**Решение**

Используя центральную предельную теорему, можно найти распределение массы всей коробки, содержащей 30 конфет. Математическое ожидание и дисперсия массы каждой конфеты: $$m = \dfrac{a + b}{2} = 13, \: d = \dfrac{(b - a)^2}{12} = 1/3$$

Итак, по центральной предельной теореме масса всей коробки имеет нормальное распределение с параметрами: $$\mu = 30 \cdot m = 390, \: \sigma^2 = 30 \cdot d = 10$$

Зная эти параметры, можно найти искомые вероятности.

In [6]:

mu **=** 390  
sigma **=** np**.**sqrt(10)  
  
norm **=** stats**.**norm(loc**=**mu, scale**=**sigma)

а) вероятность, что коробка будет весить менее 390 граммов:

In [7]:

norm**.**cdf(390)

Out[7]:

0.5

б) вероятность быть больше 395 граммов:

In [8]:

1 **-** norm**.**cdf(395)

Out[8]:

0.056923149003329065

в) вероятность быть от 380 до 400 граммов. Чтобы посчитать данную вероятность, достаточно найти вероятность быть меньше 400 граммов, и вычесть из неё вероятность быть меньше 380 граммов:

In [9]:

norm**.**cdf(400) **-** norm**.**cdf(380)

Out[9]:

0.9984345977419975

**Задача 3**

Продемонстрируйте действие центральной предельной теоремы на каком-нибудь распределении на ваш выбор (кроме нормального). Что для этого нужно сделать:

1. Выберите несколько значений n (например, 2, 5, 10, 50, но можно и больше).
2. Для выбранного значения n сгенерируйте 1000 раз выборку размера n из выбранного вами распределения, посчитайте по этой выборке выборочное среднее.
3. Изобразите гистограмму из полученных 1000 значений выборочного среднего с аргументом density=True.
4. Поверх гистограммы нарисуйте функцию плотности нормального распределения с параметрами из ЦПТ (т.е. mu = M, sigma^2 = D / n, где M - мат. ожидание выбранного вами распределения,  D - его дисперсия).

При достаточно большом n гистограмма должна соответствовать построенной функции плотности.

**Решение**

Будем работать с экспоненциальным распределением. У него есть один параметр $\lambda$, его мат. ожидание и дисперсия: $$M(X) = 1 / \lambda, \:\: D(X) = 1 / \lambda^2$$

Зафиксируем параметр $\lambda = 5$.

In [10]:

lambda\_ **=** 5  
  
mean **=** 1 **/** lambda\_  
std **=** 1 **/** lambda\_

In [11]:

**from** matplotlib **import** pyplot **as** plt

In [12]:

n **=** 100  
  
n\_samples **=** 1000

In [13]:

samples **=** np**.**random**.**exponential(scale**=**std, size**=**(n, n\_samples))  
samples **=** samples**.**mean(axis**=**0)  
  
samples**.**shape

Out[13]:

(1000,)

In [14]:

mu **=** mean  
sigma **=** std **/** np**.**sqrt(n)

In [15]:

**%config** InlineBackend.figure\_formats = ['svg']

In [16]:

plt**.**hist(samples, density**=True**, bins**=**25)  
  
x0, x1 **=** plt**.**xlim()  
  
x **=** np**.**linspace(x0, x1, 500)  
y **=** stats**.**norm**.**pdf(x, loc**=**mu, scale**=**sigma)  
  
plt**.**plot(x, y)

Out[16]:

[<matplotlib.lines.Line2D at 0x1259e0130>]

2021-01-13T16:32:56.860600 image/svg+xml Matplotlib v3.3.3, https://matplotlib.org/

In [ ]: