

Моделирование беспроводных сетей.

Лабораторная работа №1

Логинов Сергей

НФИМд-01-22

```
In [58]: import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import random
import pandas as pd
```

1. Построить матрицу с количеством строк 10 и столбцов 2, переменные заполняются случайными числами с равномерным распределением. Полученные сгенерированные случайные числа представить на графике в виде точек.

Генерируем данные

```
In [70]: a = []
for i in range(10):
    a.append([stats.uniform.rvs(1, 10, 1)[0] for i in range(2)])
a
```

```
Out[70]: [[6.871493098612426, 6.582424668700861],
[3.3027786047104626, 4.526143330784749],
[1.4919322473973153, 4.976064900765487],
[4.710858654721726, 6.276730314547376],
[1.841709600734749, 6.732411321373563],
[1.509282926148722, 1.7323649133820276],
[5.859038819529221, 2.654924627862272],
[3.993282689524107, 9.575047727898509],
[6.561481154013555, 4.8780012511342585],
[1.7160161561914002, 7.1238784679199485]]
```

На основе данных создаем фрейм

```
In [71]: head = ['x', 'y']
df = pd.DataFrame(a, columns=head)
df
```

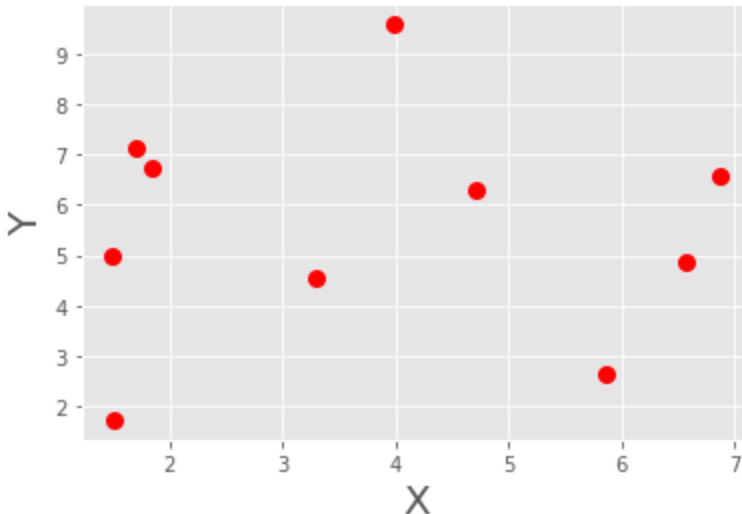
```
Out[71]:
```

	x	y
0	6.871493	6.582425
1	3.302779	4.526143
2	1.491932	4.976065
3	4.710859	6.276730
4	1.841710	6.732411
5	1.509283	1.732365
6	5.859039	2.654925

7	3.993283	9.575048
8	6.561481	4.878001
9	1.716016	7.123878

Используем точечную диаграмму

```
In [79]: plt.style.use('ggplot')
plt.scatter(df.x, df.y, c='red', linewidth=3)
plt.xlabel('X', fontsize=20)
plt.ylabel('Y', fontsize=20)
plt.show()
```



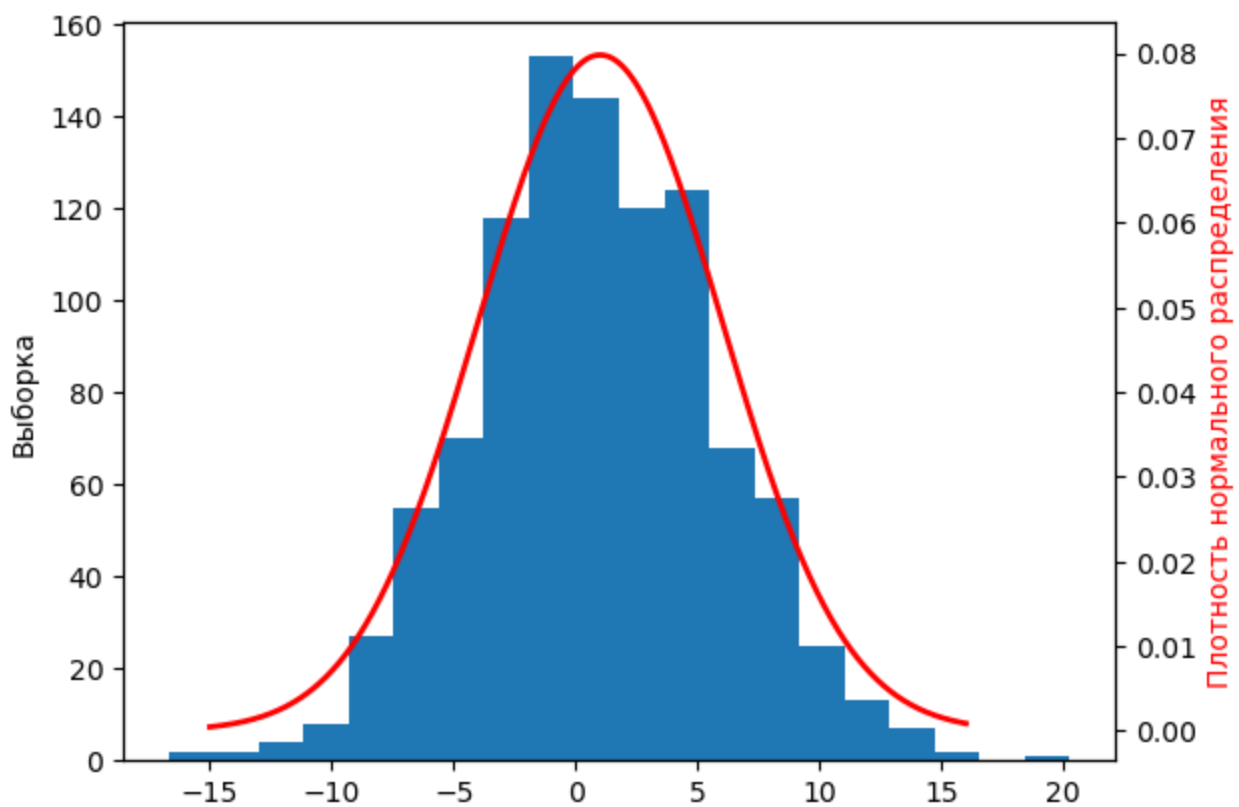
1. Сгенерировать 1000 случайных чисел с любым известным распределением и построить их гистограмму, математическое ожидание, дисперсию.

Генерируем выборку размера 1000 из нормального распределения с МО = 1 и СКО = 5 А также генерируем вспомогательный пр-массив для отрисовки кривой нормального распределения

```
In [119... sample = stats.norm.rvs(1, 5, 1000)
x = np.arange(-15, 16, 0.001)
```

Для коррекции значений по оси Y на ум первым пришло решение через объектный подход и получаем две разных оси Y на одной оси X.

```
In [120... fig, ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.hist(sample, bins=20)
ax2.plot(x, stats.norm.pdf(x, 1, 5), c='red', linewidth=2)
ax1.set_ylabel('Выборка')
ax2.set_ylabel('Плотность нормального распределения', c='red')
plt.show()
```



```
In [121... print('Математическое ожидание: ', sample.mean(),
        '\nДисперсия: ', sample.var())
```

```
Математическое ожидание: 0.88491807827625
Дисперсия: 25.35155501641047
```

1. Сгенерировать случайную точку, равномерно распределенную в квадрате со стороной a .

Будем работать с квадратом со стороной $a = 5$, генерируем точку в нем

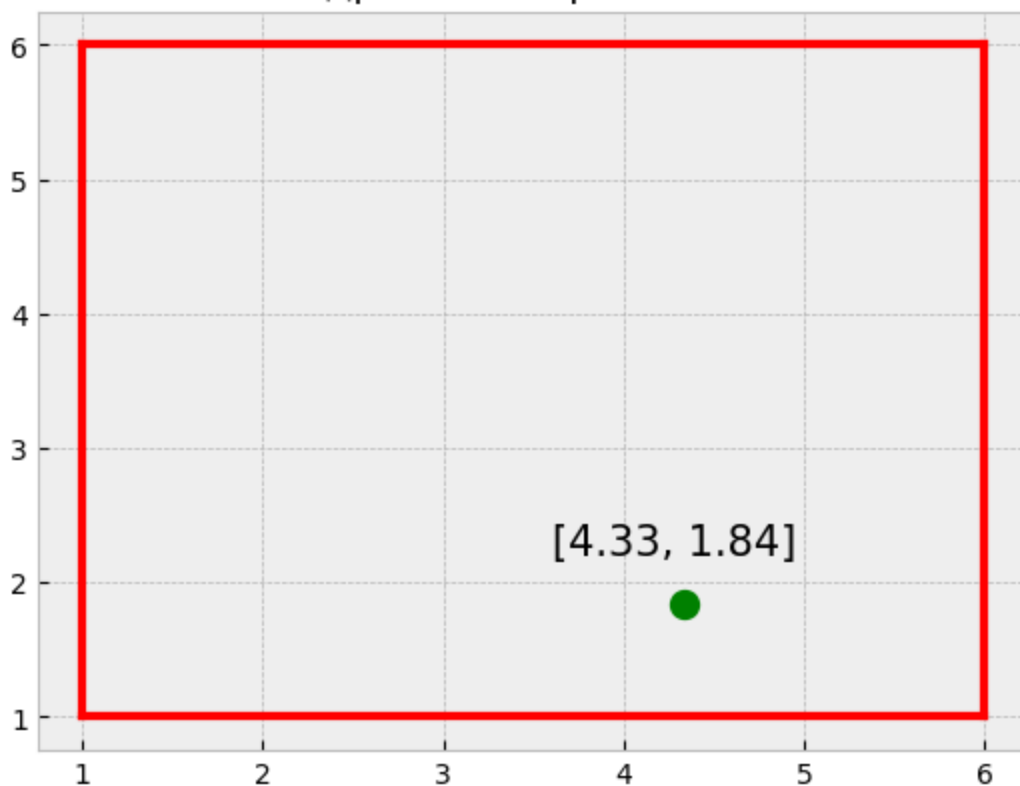
```
In [177... x = np.random.uniform(1, 6, 1)[0]
y = np.random.uniform(1, 6, 1)[0]
print(x, y)
```

```
4.333251002297283 1.8438159539503298
```

Отрисовываем квадрат и точку

```
In [196... plt.style.use('bmh')
plt.plot([1, 6], [1, 1],
         [1, 1], [1, 6],
         [1, 6], [6, 6],
         [6, 6], [6, 1],
         color='r', linewidth=3)
plt.scatter(x, y, color='g', linewidth=5)
plt.text(3.6, 2.2, '[4.33, 1.84]', fontsize=15)
plt.title('Квадрат со стороной a = 5')
plt.show()
```

Квадрат со стороной $a = 5$



И функция для любого значения a

```
In [216... def square_point(a):
    if a > 0:
        a += 1
        x, y = np.random.uniform(1, a, 1)[0], np.random.uniform(1, a, 1)[0]
        print('Координаты точки: ', round(x, 4), round(y, 4))
        plt.plot([1, a], [1, 1],
                  [1, 1], [1, a],
                  [1, a], [a, a],
                  [a, a], [a, 1],
                  color='r', linewidth=3)
        plt.scatter(x, y, color='g', linewidth=5)
        plt.title('Квадрат со стороной a = ' + str(a - 1))
        plt.show()
    else:
        print('Неверная длина стороны квадрата')
```

```
In [217... square_point(1)
```

Координаты точки: 1.5342 1.6637

Квадрат со стороной $a = 1$

