

Практические задания № 1

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
from statistics import mean, median, mode, variance, pvariance
from scipy.stats import norm
from scipy.optimize import root
import seaborn as sns
from math import sqrt
```

```
In [3]: X = [69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71, 74, 75, 75, 76]
Y = [153, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185, 112, 140, 150, 165, 185, 210, 220]
]
```

Задание 1

A. Найти среднее, медиану и моду величины X

```
In [4]: print('Среднее X:', mean(X))
print('Медиана X:', median(X))
print('Мода X:', mode(X))
```

Среднее X: 71.45
Медиана X: 71.5
Мода X: 74

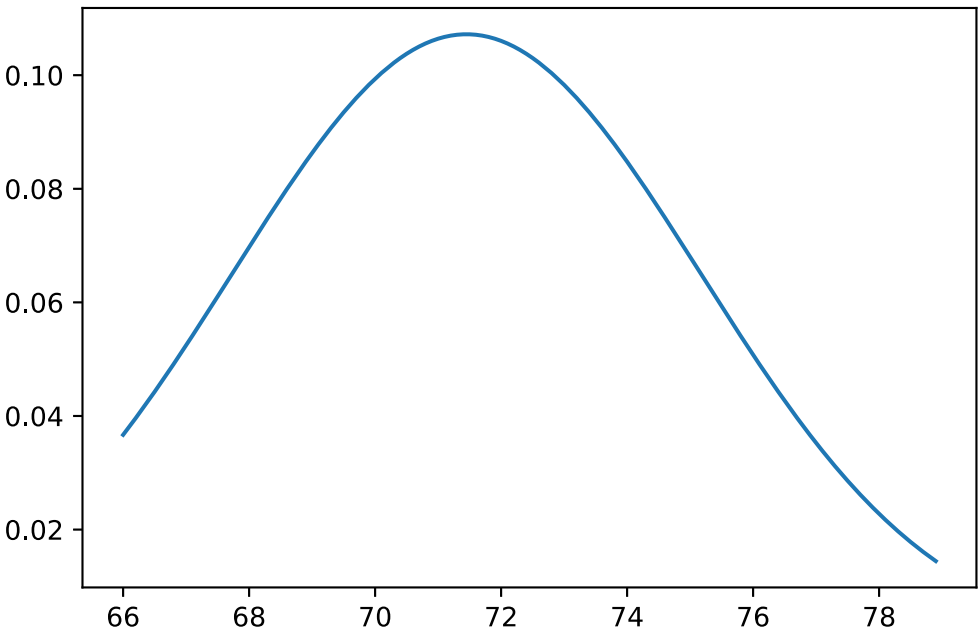
B. Найти дисперсию Y

```
In [5]: print('Дисперсия Y')
print(pvariance(Y))
print(sum([(mean(Y) - y) ** 2 for y in Y]) / len(Y))
```

Дисперсия Y
1369.21
1369.2099999999998

C. Построить график нормального распределения для X

```
In [20]: mean_x = mean(X)
var_x = pvariance(X)
x_axis = np.arange(min(X), max(X), 0.1)
f = [np.exp(-(x - mean_x)**2 / (2 * var_x)) / sqrt(2 * np.pi * var_x) for x in x_axis]
plt.plot(x_axis, f)
plt.show()
```



D. Найти вероятность того, что возраст больше 80

```
In [8]: x = np.array(X)
print('Вероятность, что возраст больше 80:', (x > 80).sum() / len(X))
```

Вероятность, что возраст больше 80: 0.0

E. Найти двумерное мат. ожидания и ковариационную матрицу для этих двух величин

```
In [9]: print('Мат ожидание', np.mean([X, Y], axis=1))
print('Ковариационная матрица')
np.cov([X, Y])
```

Мат ожидание [71.45 164.7]
Ковариационная матрица

```
Out[9]: array([[ 14.57631579, 128.87894737],
               [128.87894737, 1441.27368421]])
```

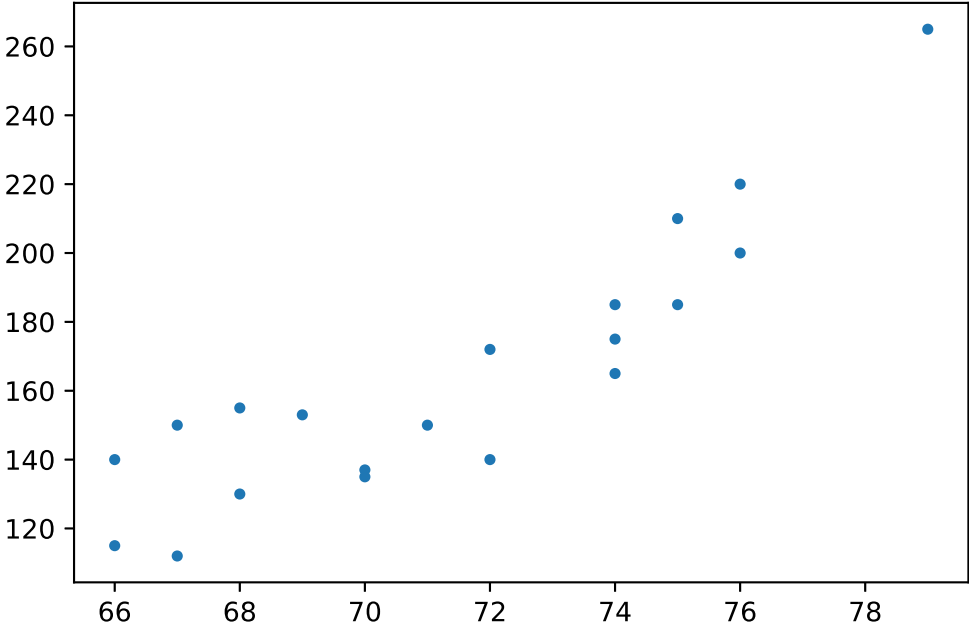
F. Определять корреляцию между X и Y

```
In [10]: mean_xy = mean([X[i] * Y[i] for i in range(len(Y))])
mean_x = mean(X)
mean_y = mean(Y)
print(np.corrcoef([X, Y]))
print('Коэффициент корреляции: ', (mean_xy - mean_x * mean_y) / (sqrt(pvariance(X) * pvariance(Y))))
```

[[1. 0.88917014]
[0.88917014 1.]]
Коэффициент корреляции: 0.8891701351748014

G. Построить диаграмму рассеяния, отображающая зависимость между возрастом и весом

```
In [21]: plt.scatter(X, Y, s=10)
plt.show()
```



Задание 2

```
In [13]: X1 = [17, 11, 11]
X2 = [17, 9, 8]
X3 = [12, 13, 19]
```

```
In [14]: cov = np.cov([X1, X2, X3])
cov
```

```
Out[14]: array([[12.  , 17.  , -8.  ],
               [17.  , 24.33333333, -12.83333333],
               [-8.  , -12.83333333, 14.33333333]])
```

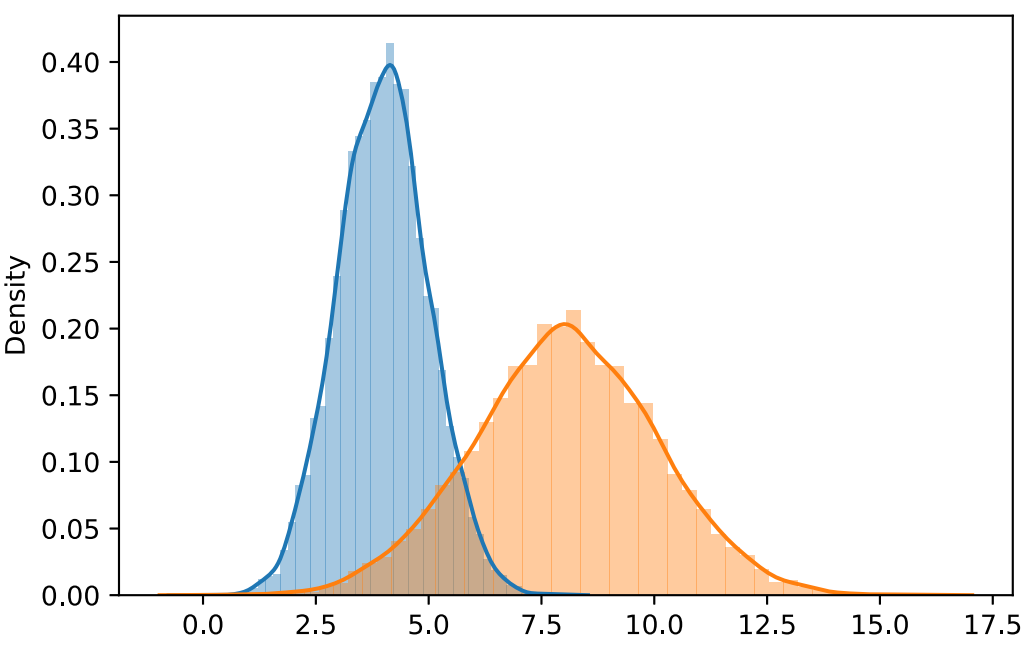
```
In [15]: print('Многомерная дисперсия:', cov[0][0] * cov[1][1] * cov[2][2] + cov[1][0] * cov[2][1] * cov[0][2] +
cov[0][1] * cov[1][2] * cov[2][0] - cov[2][0] * cov[1][1] * cov[0][2] - cov[0][0] * cov[2][1] * cov[1]
[2] - cov[0][1] * cov[1][0] * cov[2][2])
np.linalg.det(cov)
np.trace(cov)
```

Многомерная дисперсия: -9.094947017729282e-13

```
Out[15]: 50.66666666666667
```

Задание 3

```
In [22]: N = 10000
mean_Na, mean_Nb = 4, 8
std_Na, std_Nb = 1, 2
Na = np.random.normal(mean_Na, std_Na, N)
Nb = np.random.normal(mean_Nb, std_Nb, N)
sns.distplot(Na)
sns.distplot(Nb)
plt.show()
```



A. Для каждого из значения {5,6,7} определите какое из распределений сгенерировало значение с большей вероятностью.

```
In [17]: def getP(n):
return (np.abs(Na - n) < 1).sum() / N, (np.abs(Nb - n) < 1).sum() / N
Pa5, Pb5 = getP(5)
Pa6, Pb6 = getP(6)
Pa7, Pb7 = getP(7)
print('Pa(5) =', Pa5, 'Pb(5) =', Pb5)
print('Pa(6) =', Pa6, 'Pb(6) =', Pb6)
print('Pa(7) =', Pa7, 'Pb(7) =', Pb7)
```

Pa(5) = 0.4772 Pb(5) = 0.1367
Pa(6) = 0.1559 Pb(6) = 0.2409
Pa(7) = 0.0214 Pb(7) = 0.3328

5 - с большей вероятностью сгенерировало Na

6 - с большей вероятностью сгенерировало Nb

7 - с большей вероятностью сгенерировало Nb

B. Найди значение, которой могло быть сгенерировано обеими распределениями с равной вероятностью

```
In [18]: result = root(lambda x : norm.pdf(x, mean_Nb, std_Nb) - norm.pdf(x, mean_Na, std_Na), 5)
print('Ответ: ', result.x[0])
```

Ответ: 5.659909655901637