

Практическая работа №1

Задание 1

```
In [12]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
from scipy import stats
import scipy
import numpy as np

# Data
X = np.array([69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67, 66, 71,
Y = np.array([153, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 265, 185
```

A. Найти среднее, медиану и моду величины X

```
In [3]: print(f"Среднее: {np.mean(X)}, Медиана: {np.median(X)}, Мода: {stats.mode(X)[0][1]

Среднее: 71.45, Медиана: 71.5, Мода: 74
```

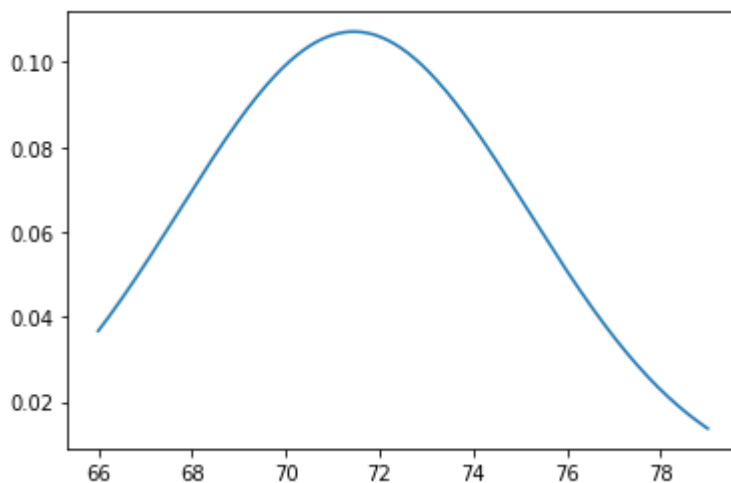
B. Найти дисперсию Y

```
In [4]: print(f"Дисперсия Y: {round(np.var(Y), 2)}")

Дисперсия Y: 1369.21
```

C. Построить график нормального распределения для X

```
In [5]: x = np.linspace(X.min(), X.max(), 1000000)
plt.plot(x, stats.norm.pdf(x, np.mean(X), np.sqrt(np.var(X))))
plt.show()
```



D. Найти вероятность того, что возраст больше 80

```
In [6]: print(np.count_nonzero(X > 80) / len(X))

0.0
```

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

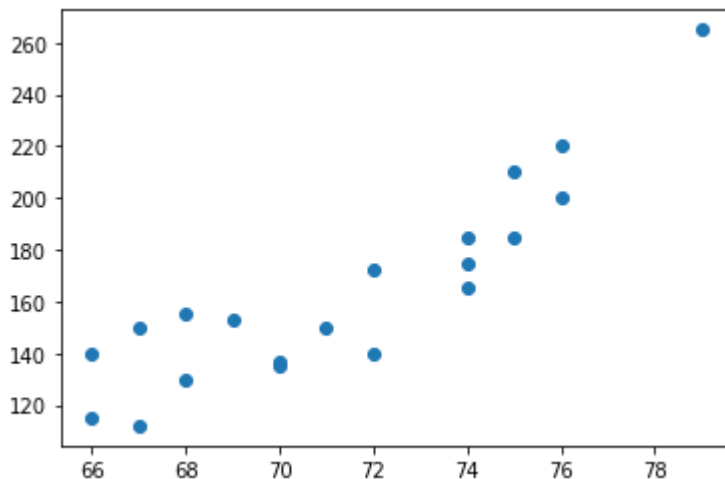
```
In [7]: print(f"Мат. ожидание: {np.mean([X, Y], axis=1)} \nКовариационная матрица: \n {np.
Мат. ожидание: [ 71.45 164.7 ]
Ковариационная матрица:
[[ 14.57631579 128.87894737]
 [ 128.87894737 1441.27368421]]
```

F. Определять корреляцию между X и Y

```
In [8]: print(np.corrcoef(X, Y)[0, 1])
0.8891701351748048
```

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

```
In [9]: plt.scatter(X, Y)
plt.show()
```



Задание 2

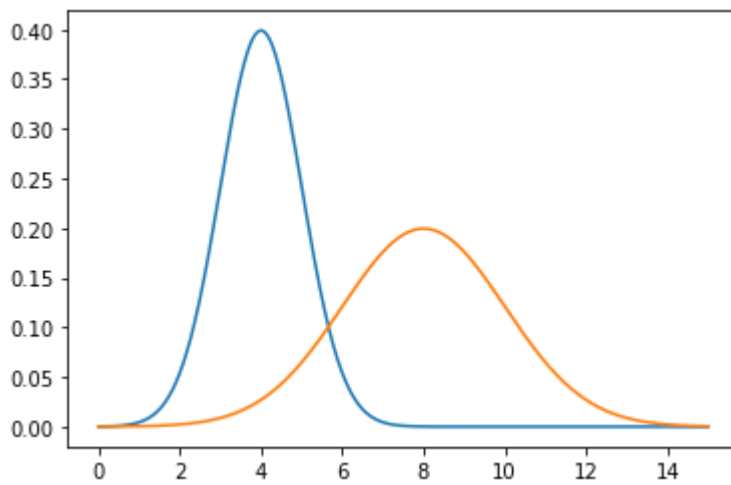
```
In [10]: X1 = [17, 11, 11]
X2 = [17, 9, 8]
X3 = [12, 13, 19]
m = np.cov([X1, X2, X3])
print(f'Ковариационная матрица: \n{m}')
print(f'Обобщенная дисперсия: {np.linalg.det(m)}')
```

```
Ковариационная матрица:
[[ 12.          17.          -8.          ]
 [ 17.          24.33333333 -12.83333333]
 [ -8.          -12.83333333  14.33333333]]
Обобщенная дисперсия: 0.0
```

Задание 3

```
In [14]: x = np.linspace(0, 15, 1000000)
a = stats.norm.pdf(x, 4, 1)
b = stats.norm.pdf(x, 8, 2)
plt.plot(x, a)
plt.plot(x, b)
plt.show()

result = scipy.optimize.root(lambda x: stats.norm.pdf(x, 8, 2) - stats.norm.pdf(x, 4, 1))
print(result)
```



```
fjac: array([[ -1.]])
fun: array([4.16333634e-17])
message: 'The solution converged.'
nfev: 8
qtf: array([4.49894288e-12])
r: array([-0.22584348])
status: 1
success: True
x: array([5.65990966])
```

С большей вероятностью значение 5 сгенерировало первое распределение

С большей вероятностью значение 6 сгенерировало второе распределение

С большей вероятностью значение 7 сгенерировало второе распределение

Значение с однокровой вероятностью: 5.65990966

In []: