

# PRACTICAL TASK #1

## PART #1

```
B [1]: import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

```
B [2]: X = [69, 74, 68, 70, 72, 67, 66, 70, 76, 68, 72, 79, 74, 67,
Y = [53, 175, 155, 135, 172, 150, 115, 137, 200, 130, 140, 26
```

```
B [3]: X = np.array(X)
Y = np.array(Y)
```

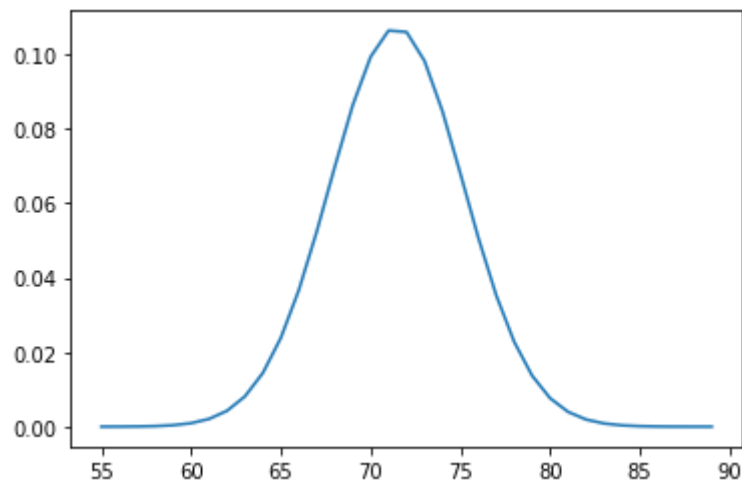
```
B [4]: # Найти среднее, медиану и моду величины X
np.mean(X) np.median(X) *stats.mode(X)[0]
```

Out[4]: (71.45, 71.5, 74)

```
B [5]: # Найти дисперсию Y
np.var(Y)
```

Out[5]: 1961.2100000000003

```
B [6]: # Построить график нормального распределения для X
x_axis = np.arange(55, 90)
plt.plot(x_axis, stats.norm.pdf(x_axis, np.mean(X), np.std(X))
plt.show()
```



```
B [7]: # Найти вероятность того, что возраст больше 80
1 - stats.norm(np.mean(X), np.std(X)).cdf(80)
```

Out[7]: 0.010791377919371459

```
B [8]: # Найти двумерное мат. ожидания и ковариационную матрицу для
nn.mean([X, Y], axis=1)
```

```
Out[8]: array([ 71.45, 159.7 ])
```

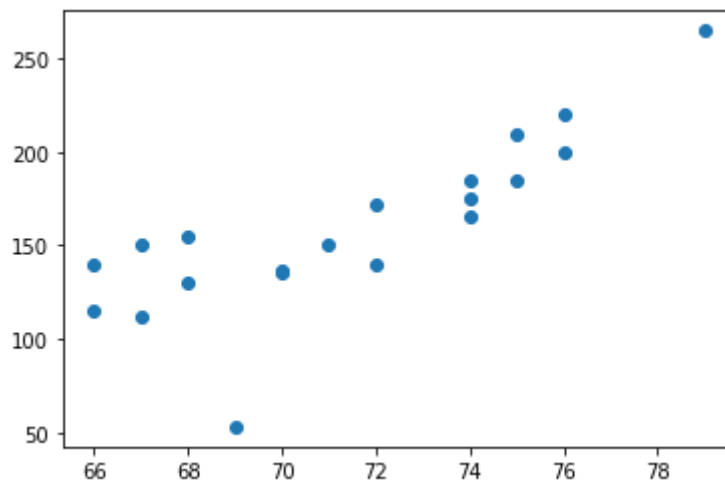
```
B [9]: # Найти ковариационную матрицу для этих двух величин
nn.cov(X, Y)
```

```
Out[9]: array([[ 14.57631579, 141.77368421],
               [ 141.77368421, 2064.43157895]])
```

```
B [10]: # Определять корреляцию между X и Y
nn.corrcoef(X, Y)[0, 1]
```

```
Out[10]: 0.8172811723193554
```

```
B [11]: # Построить диаграмму рассеяния, отображающая зависимость меж
plt.scatter(X, Y)
plt.show()
```



## PART #2

```
B [12]: M = [[17, 17, 12],
              [11, 9, 13],
              [11, 8, 19]]
```

```
B [13]: # Рассчитайте ковариационную матрицу
nn.cov(M)
```

```
Out[13]: array([[ 8.33333333, -5.          , -15.83333333],
                [-5.          ,  4.          ,  11.          ],
                [-15.83333333,  11.          ,  32.33333333]])
```

```
B [14]: # и обобщенную дисперсию
nn.linalg.det(nn.cov(M)) # определитель ковариационной матрицы
```

```
Out[14]: 2.2204460492503156e-14
```

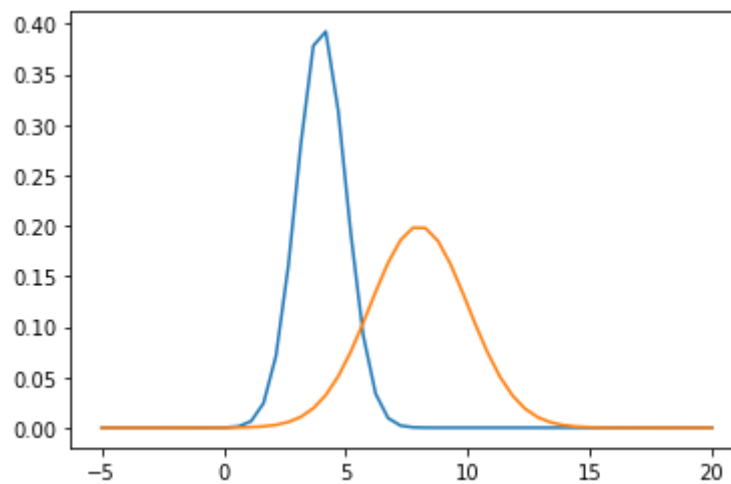
## PART #3

```
B [15]: Na = stats.norm(4, 1)
        Nb = stats.norm(8, 2)
```

```
B [16]: # Для каждого из значения {5,6,7} определите какое из распределений
        ["Na" if Na.pdf(n) > Nb.pdf(n) else "Nb" for n in [5, 6, 7]]
```

```
Out[16]: ['Na', 'Nb', 'Nb']
```

```
B [17]: # Найди значение, которое могло быть сгенерировано обеими распределениями
        ls = np.linspace(-5, 20)
        plt.plot(ls, Na.pdf(ls))
        plt.plot(ls, Nb.pdf(ls))
        plt.show()
```



```
B [18]: ls = np.linspace(5, 6, 100)
        [e1 for e1 in ls if (abs(Na.pdf(e1) - Nb.pdf(e1)) < 0.001)]
```

```
Out[18]: [5.656565656565657]
```