

```
In [8]: import numpy as np
```

Тема “Вычисления с помощью Numpy”

Задание 1 Импортируйте библиотеку Numpy и дайте ей псевдоним np. Создайте массив Numpy под названием a размером 5x2, то есть состоящий из 5 строк и 2 столбцов. Первый столбец должен содержать числа 1, 2, 3, 3, 1, а второй - числа 6, 8, 11, 10, 7. Будем считать, что каждый столбец - это признак, а строка - наблюдение. Затем найдите среднее значение по каждому признаку, используя метод mean массива Numpy. Результат запишите в массив mean_a, в нем должно быть 2 элемента.

```
In [9]: a = np.array([[1, 2, 3, 3, 1],[6, 8, 11, 10, 7]]).T
```

```
In [10]: a
```

```
Out[10]: array([[ 1,  6],
                [ 2,  8],
                [ 3, 11],
                [ 3, 10],
                [ 1,  7]])
```

```
In [11]: mean_a = np.mean(a, axis=0)
```

```
In [12]: mean_a
```

```
Out[12]: array([2. , 8.4])
```

Задание 2

Вычислите массив a_centered, отняв от значений массива “a” средние значения соответствующих признаков, содержащиеся в массиве mean_a. Вычисление должно производиться в одно действие. Получившийся массив должен иметь размер 5x2.

```
In [13]: a_centered = a - mean_a
```

```
In [14]: a_centered
```

```
Out[14]: array([[ -1. , -2.4],
                [  0. , -0.4],
                [  1. ,  2.6],
                [  1. ,  1.6],
                [ -1. , -1.4]])
```

Задание 3

Найдите скалярное произведение столбцов массива `a_centered`. В результате должна получиться величина `a_centered_sp`. Затем поделите `a_centered_sp` на $N-1$, где N - число наблюдений.

```
In [15]: a_centered_sp = np.inner(a_centered[:, 0], a_centered[:, 1])  
a_centered_sp
```

```
Out[15]: 8.0
```

```
In [16]: a_segmentation = a_centered_sp / (len(a) - 1)  
a_segmentation
```

```
Out[16]: 2.0
```

Задание 4**

Число, которое мы получили в конце задания 3 является ковариацией двух признаков, содержащихся в массиве "a". В задании 4 мы делили сумму произведений центрированных признаков на $N-1$, а не на N , поэтому полученная нами величина является несмещенной оценкой ковариации.

Выборочная ковариация и выборочная дисперсия — Студопедия В этом задании проверьте получившееся число, вычислив ковариацию еще одним способом - с помощью функции `np.cov`. В качестве аргумента `m` функция `np.cov` должна принимать транспонированный массив "a". В получившейся ковариационной матрице (массив Numpy размером 2×2) искомое значение ковариации будет равно элементу в строке с индексом 0 и столбце с индексом 1.

```
In [18]: a_np_cov = np.cov(a.T)  
a_np_cov
```

```
Out[18]: array([[1. , 2. ],  
                [2. , 4.3]])
```

```
In [21]: a_np_cov[0, 1] == a_segmentation  
a_segmentation
```

```
Out[21]: 2.0
```

```
In [22]: a_np_cov[0, 1] == a_segmentation
```

```
Out[22]: True
```

```
In [ ]:
```

