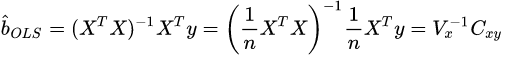
Задача 1



Выше представлена формула (между **первым** и **вторым** знаками =) для получения оценок коэффициентов линейной регрессии (не пугайтесь: T означает транспонирование матрицы, возведение в степень -1 - взятие обратной матрицы. Все эти операции в numpy уже реализованы). Матрица X и вектор y - фичи (предикторы) и таргет (целевая переменная) в вашей модели соответственно. Найдите вектор b: напишите функцию, которая будет принимать на вход матрицу X и вектор y и выдавать вектор b. Опробуйте ее примерах ниже.

**Ободрение.** Несмотря на то, что задание выглядит сложно, сама формула переносится в одну строку кода.

X = np.array([

    [34.98, 95.49, 72.92, 46.52, 83.85, 96.29, 26.44, 71.55, 4.82, 76.71],

    [20.15, 1.89, -26.61, -36.19, 27.71, -20.6, -20.73, 23.95, -28.55, -48.78],

    [0.381, 3.597, -1.634, 16.979, 10.622, 19.857, -8.045, -0.557, 9.775, 10.587],

    [0.4, -0.2, 0, 1.2, 0.1, -0.5, -1, -2.4, 0.5, 0.9],

    [-971.2, -951.6, -951.1, -978.2, -979.7, -932.7, -986.9, -963.3, -900.6, -923.6],

    [-616, -823, -942, -631, -946, -674, -698, -728, -810, -731],

    [11.33, 14.109, 13.702, -0.5, 14.7, 14.845, 1.063, -9.091, 17.277, 0.701],

    [55.73, 45.37, 14.75, 15.33, 63.15, 71.65, 71, 47.08, 36.42, 23.73],

    [80, 56, 53, 35, 39, 12, 61, 71, 25, 3],

    [19, -4, 9, 9, 5, 14, 2, 4, 19, 8]

])

y = np.array([2.084, -3.273, 4.624, 1.467, 0.105, 5.994, -0.417, 4.177, 5.806, 6.222])

### Задача 2

Вам необходимо реализовать функцию, которая будет считать т. н. [Евклидово расстояние](https://ru.wikipedia.org/wiki/%D0%95%D0%B2%D0%BA%D0%BB%D0%B8%D0%B4%D0%BE%D0%B2%D0%B0_%D0%BC%D0%B5%D1%82%D1%80%D0%B8%D0%BA%D0%B0) между двумя векторами. Функция должна принимать на вход два массива и возвращать значение Евклидова расстояния.

Использование циклов и любых подобных встроенных инструментов Python (генераторы списков, функция map()) запрещено.

Для проверки данных сгенерируйте два вектора случайных значений длиной 100 (параметр size=). Один должен содержать значения из нормального распределения (np.random.normal()) с параметрами [0, 1], другой - из распределения пуассона (np.random.poisson()) с параметром 10.