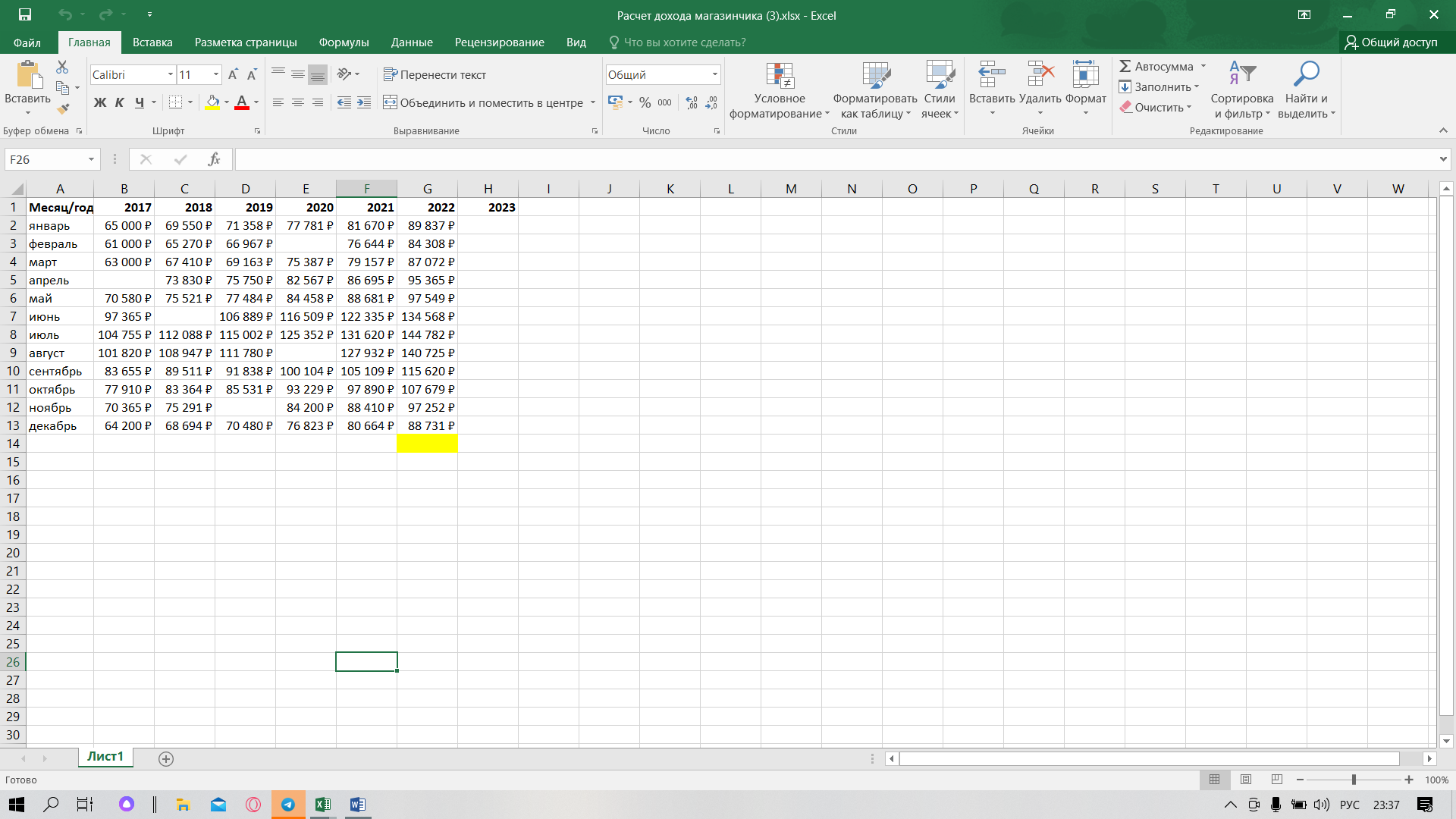
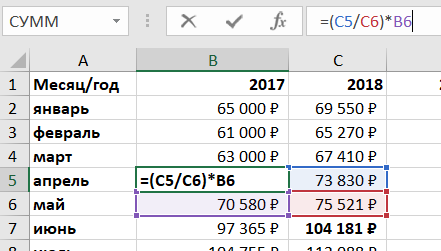
**Регламент выполненной работы**  
**Цели и задачи:** заполнить пропущенные значения, рассчитать нейтральный, оптимистичный и пессимистичные прогнозы на два следующих года

**Выполнила: Шаманаева Ангелина**

***Заполнение пропусков в таблице доходов магазинчика***



Для того, чтобы найти недостающие значения(сумму дохода) я использую коэффициент сезонности



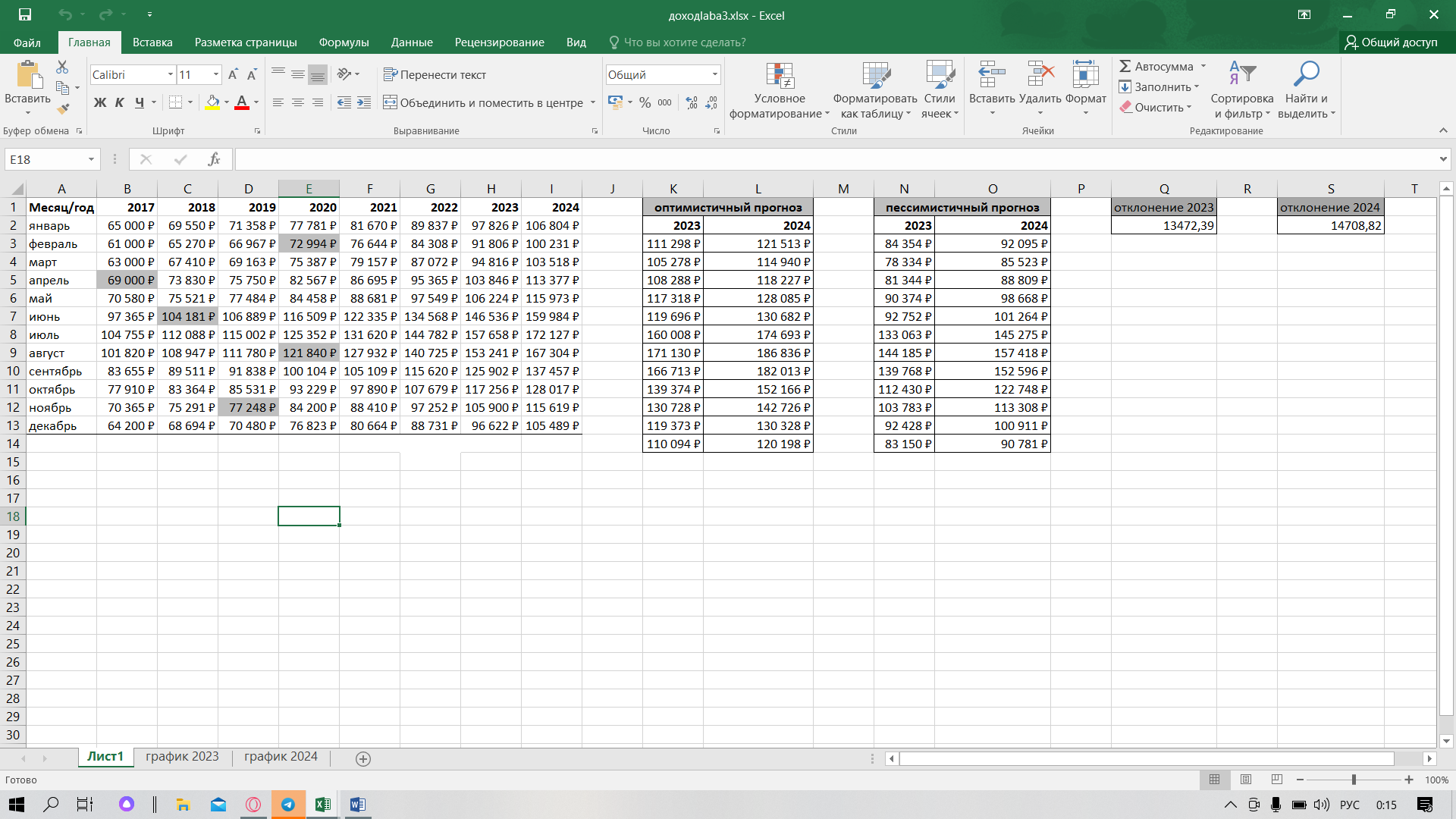
Апрель 2017 = (апрель 2018 : март 2018) \* март 2017 = (73 830 : 67 410) \* 63 000 = 69 000

Июнь 2018 = (106 889 : 77 484) \* 75 521 = 104 181

Ноябрь 2019 = (84 200 : 93 229) \* 85 531 = 77 248

Август 2020 = (127 932 : 131 620) \* 125 352 = 121 840

Февраль 2020 = (76 644 : 81 670) \* 77 781 = 72 994



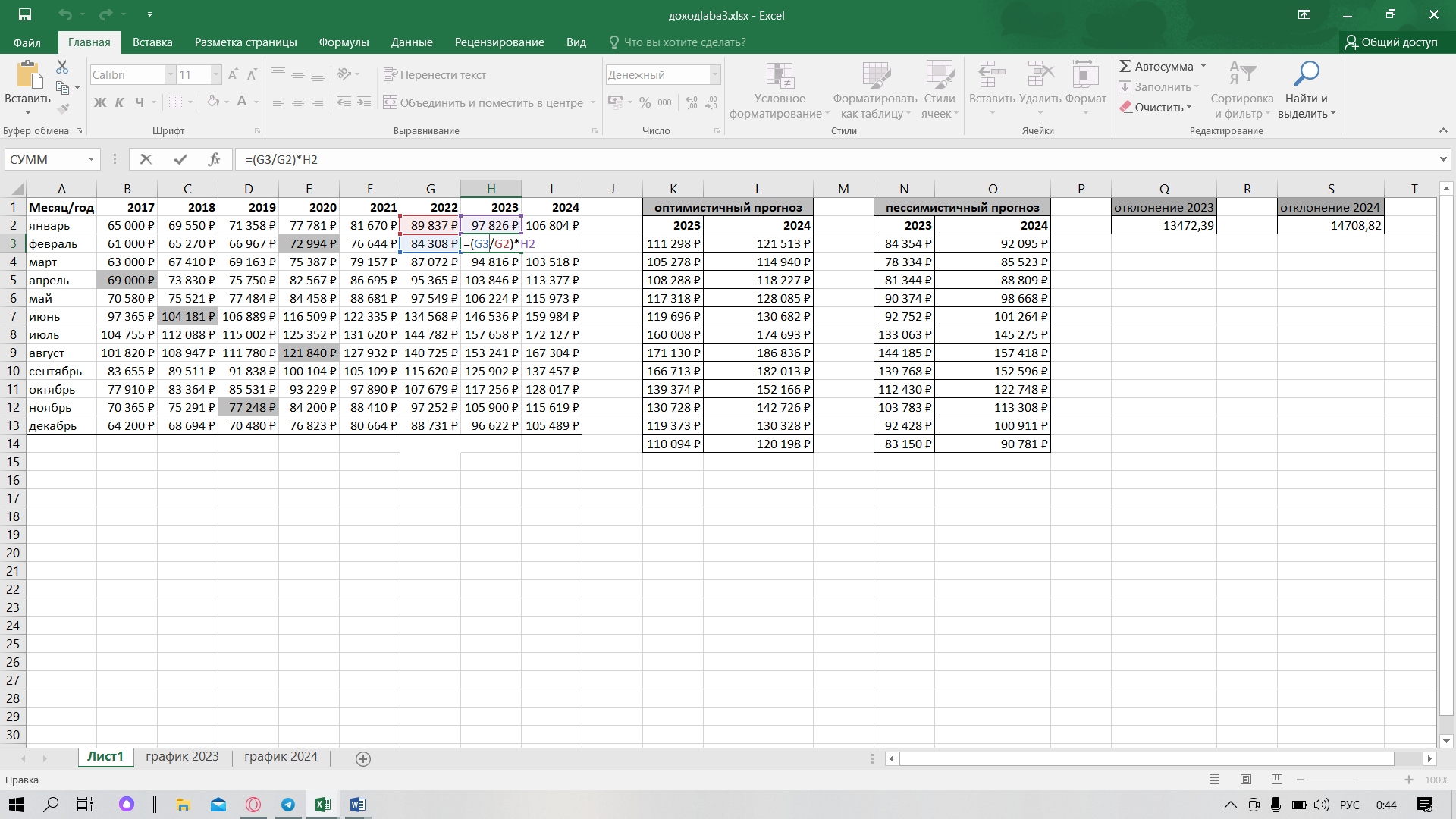
***Нахождение нейтральных прогнозов на 2023 и 2024 год***

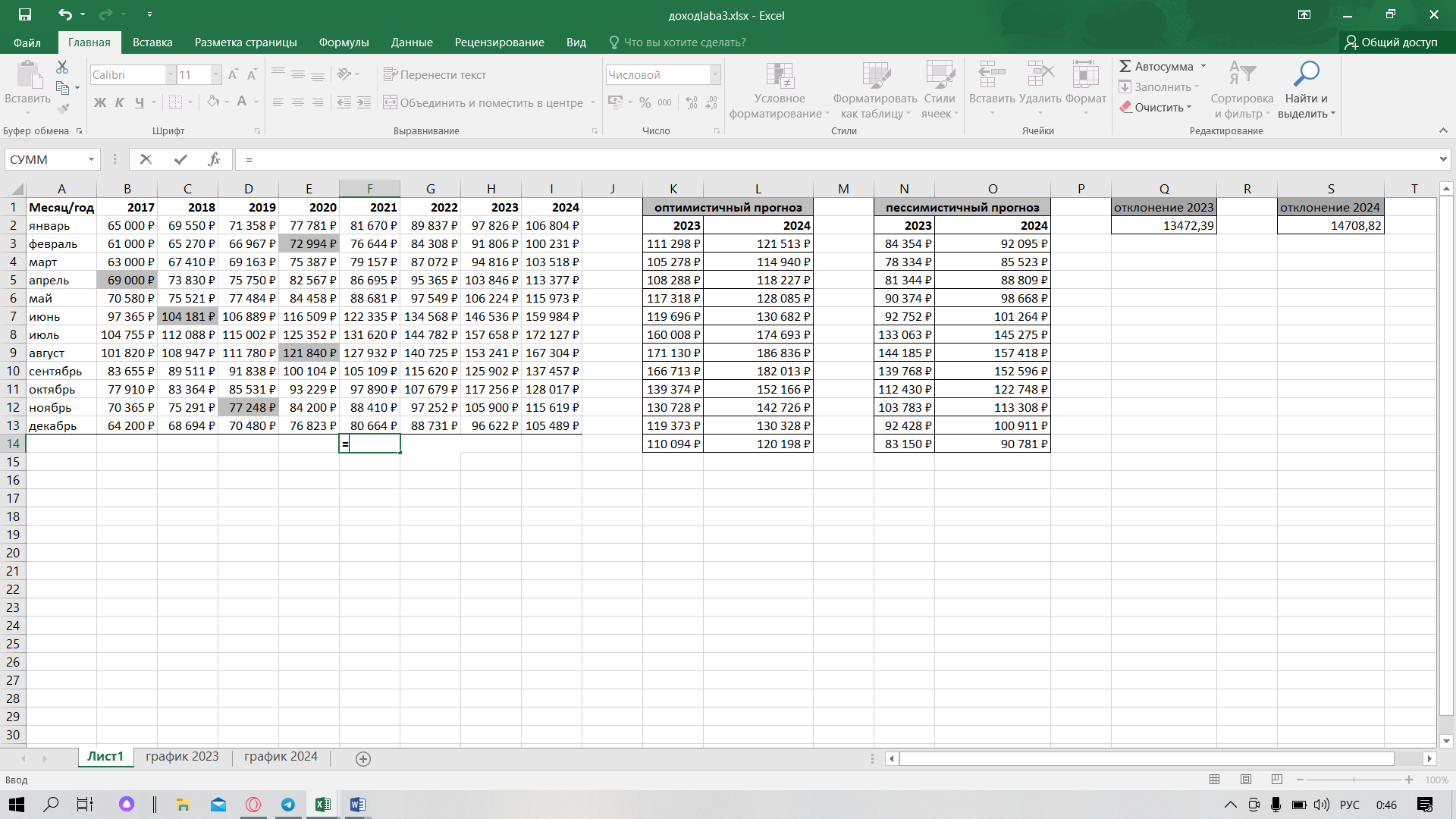
1. Для того, чтобы найти доход на январь 2021 и январь 2022 нахожу коэффициент математического ожидания

Январь 2023 = (2018:2017) + (2019: 2018) + ( 2020: 2019) + (2021: 2020) + (2022: 2021) \ 6 = 0, 889334

Январь 2024 = (2018:2017) + (2019: 2018) + ( 2020: 2019) + (2021: 2020) + (2022: 2021) + (2023: 2022) \ 6 = 0,917848

1. Далее все оставшиеся месяца нахожу через коэффициент сезонности:





***Нахождение оптимистичного и пессимистичного прогноза на 2023 и 2024 год***

Для начала нахожу отклонение через функцию ДОВЕРИТ в Exel

Функция возвращает доверительный интервал для среднего генеральной совокупности с нормальным распределением. Доверительный интервал — это диапазон значений. Выборка "x" находится в центре этого диапазона, а диапазон — x ± ДОВЕРИТ.

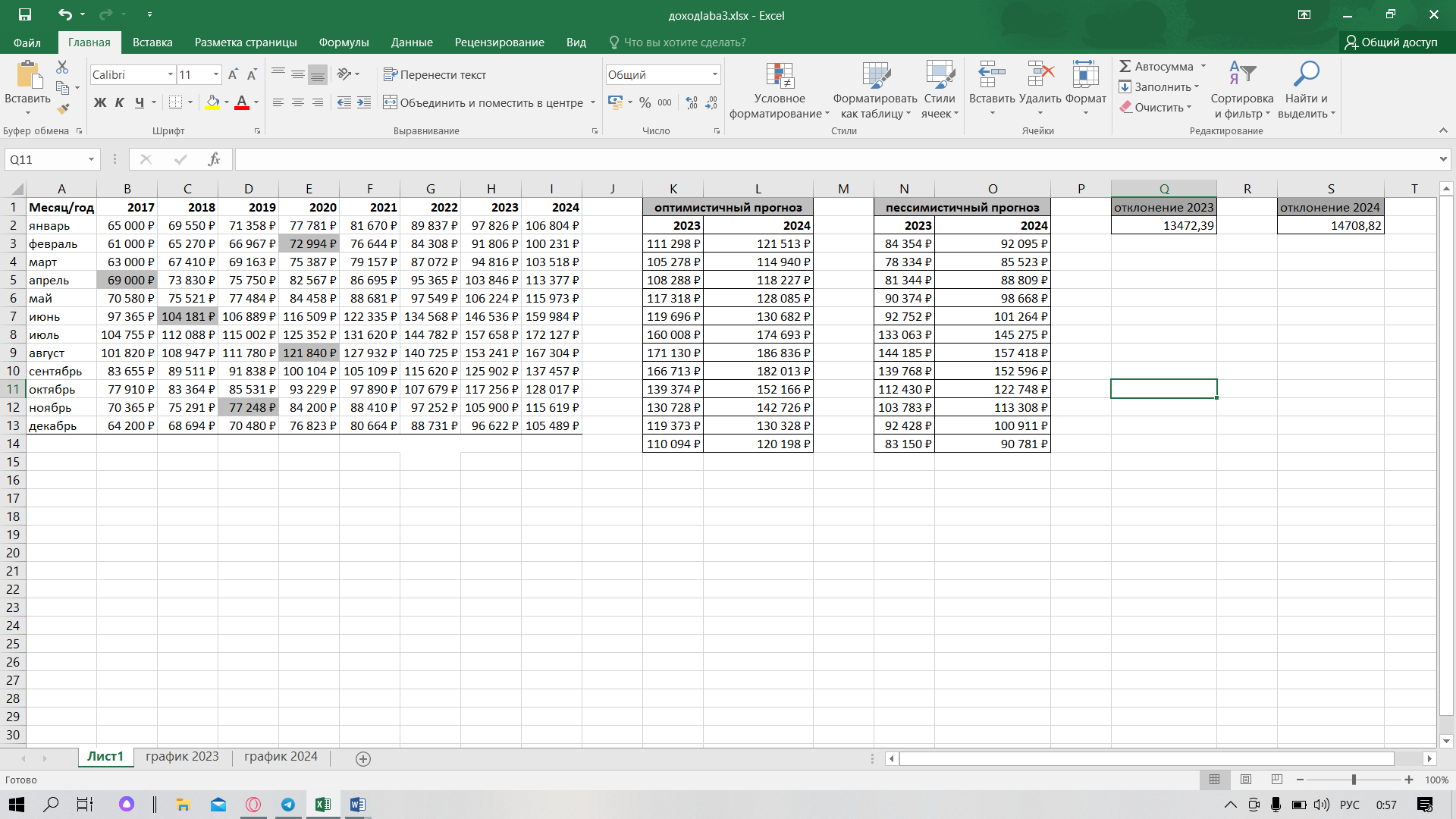
ДОВЕРИТ(альфа;стандартное\_откл;размер)

Аргументы функции:

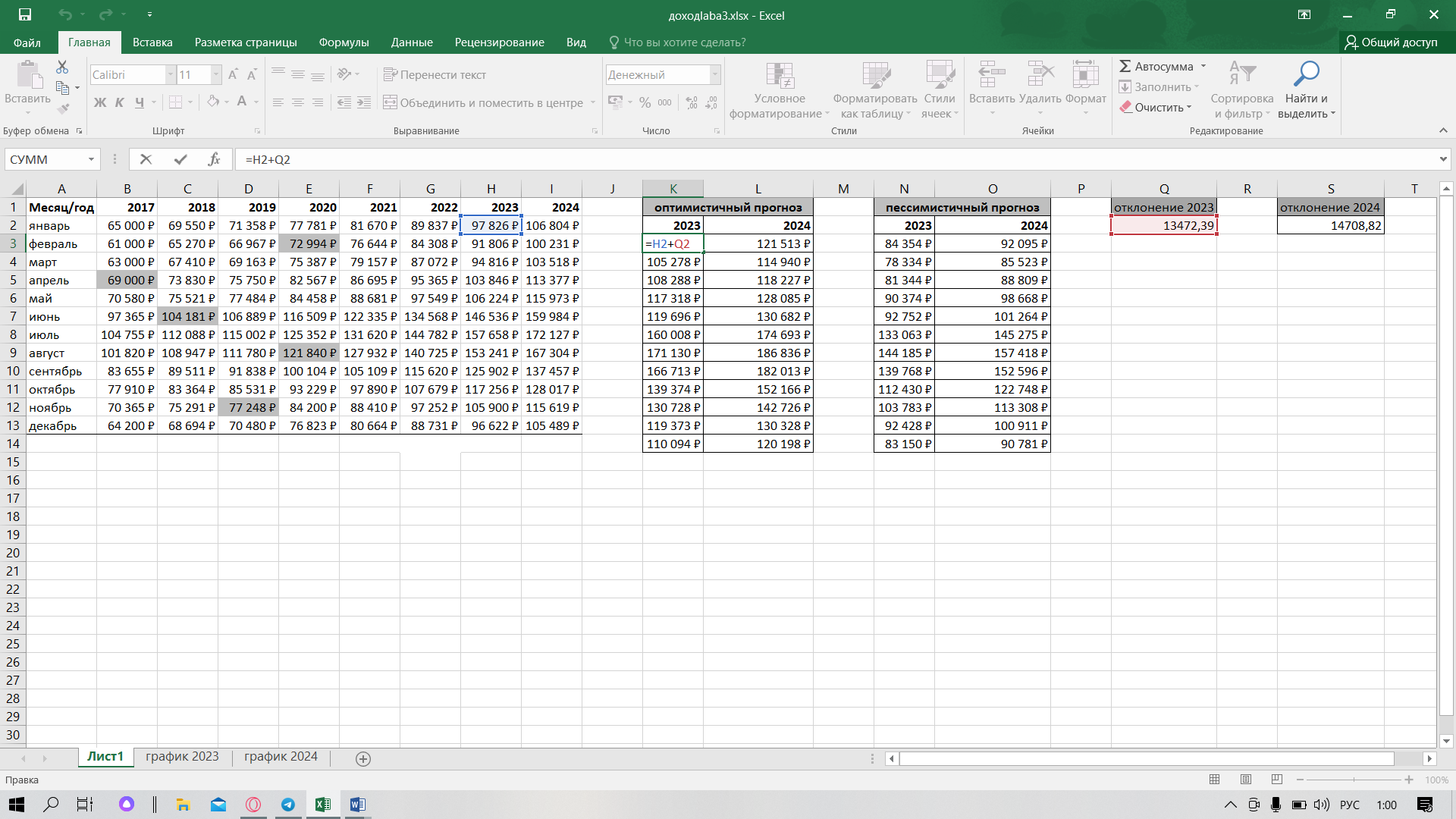
* **Альфа**     — обязательный аргумент. Уровень значимости, используемый для вычисления доверительного уровня. Доверительный уровень равен 100\*(1 - альфа) процентам или, иными словами, значение аргумента "альфа", равное 0,05, означает 95-процентный доверительный уровень.
* **Стандартное\_откл**     — обязательный аргумент. Стандартное отклонение генеральной совокупности для диапазона данных, предполагается известным.
* **Размер**     — обязательный аргумент. Размер выборки.

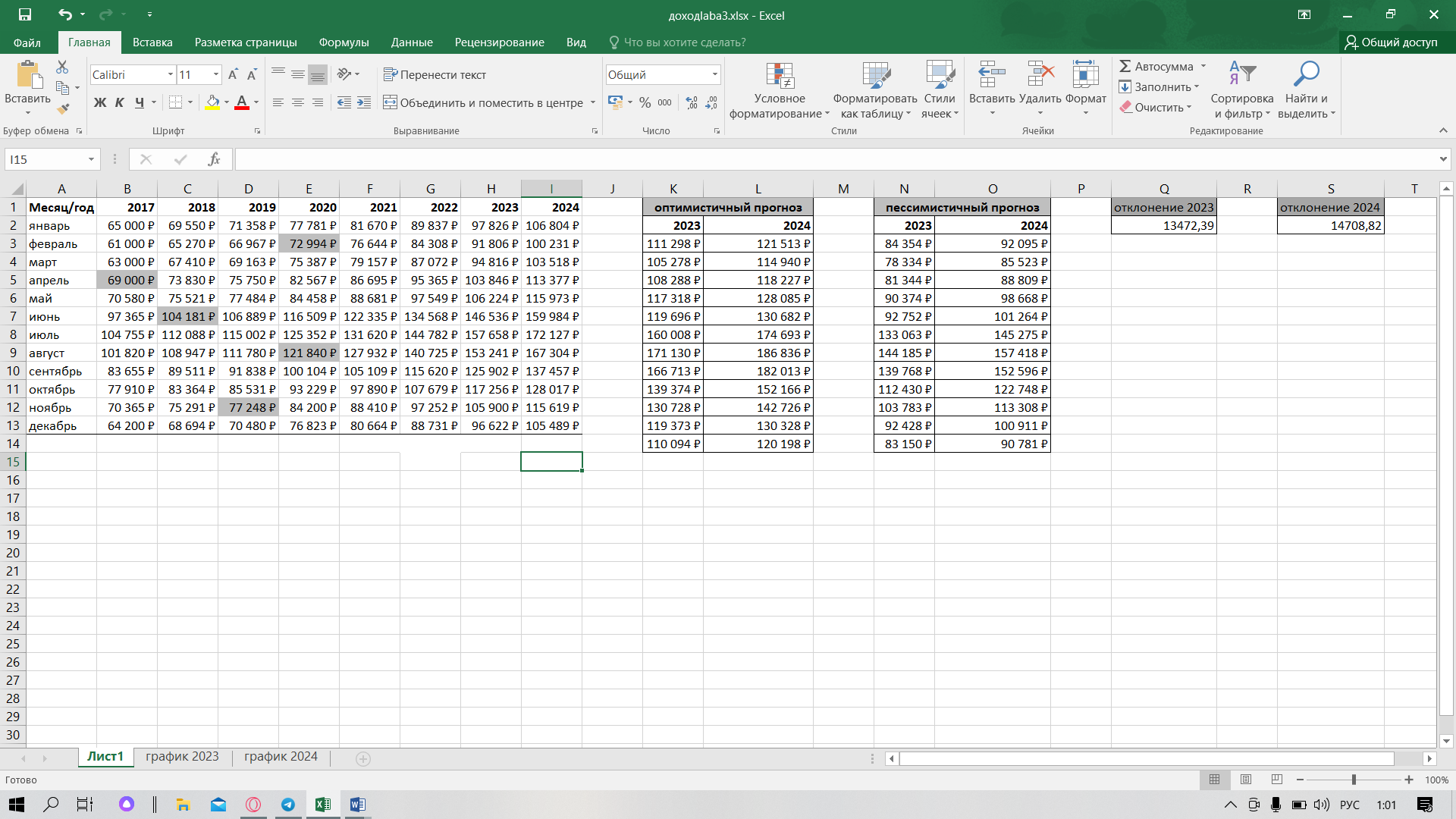
 ***Нахождение отклонения на 2023 год:***

***Аналогично находится отклонение на 2024 год***

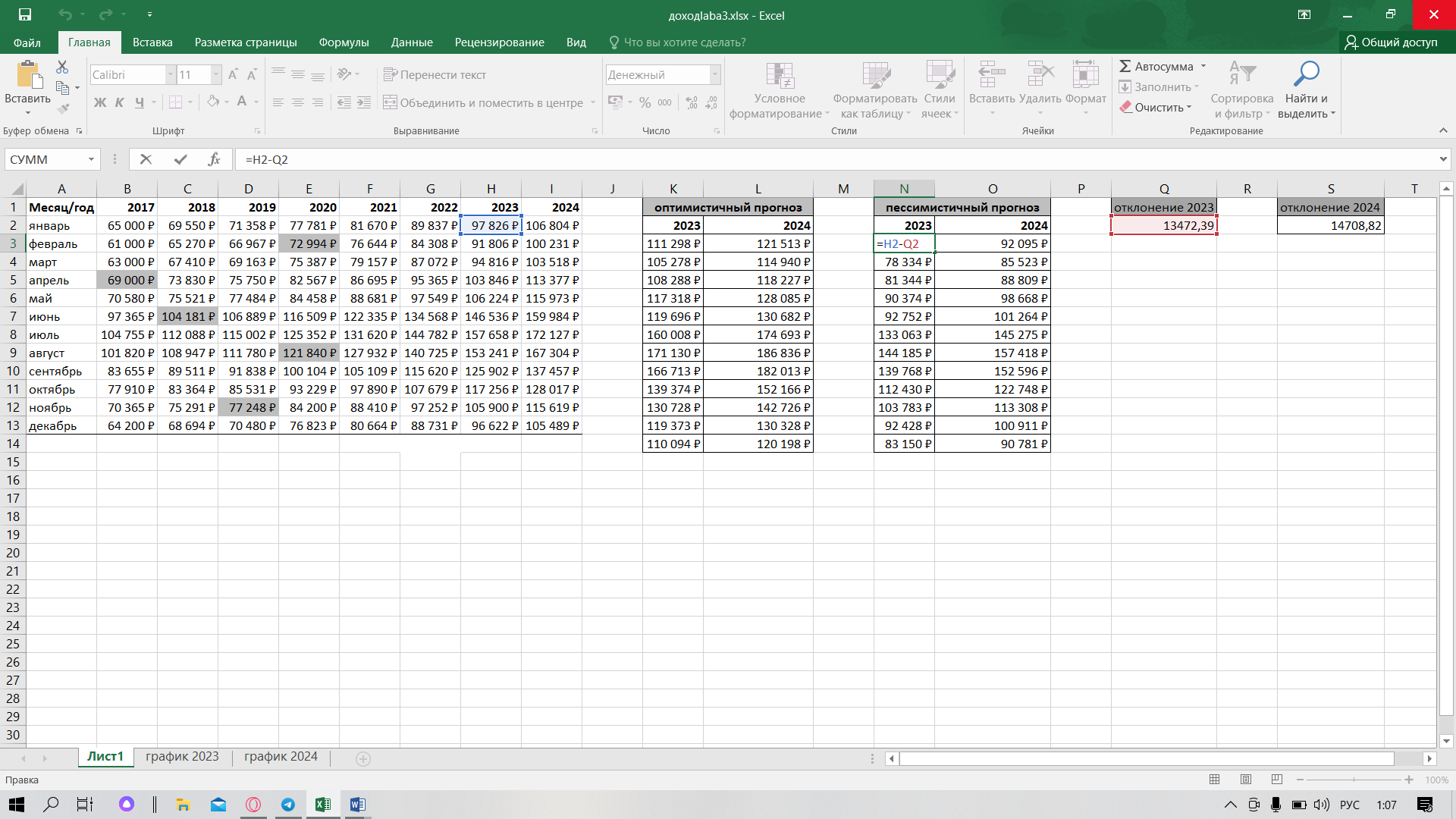


***Для расчета оптимистичного прогноза отклонение суммируется со значением искомого месяца***

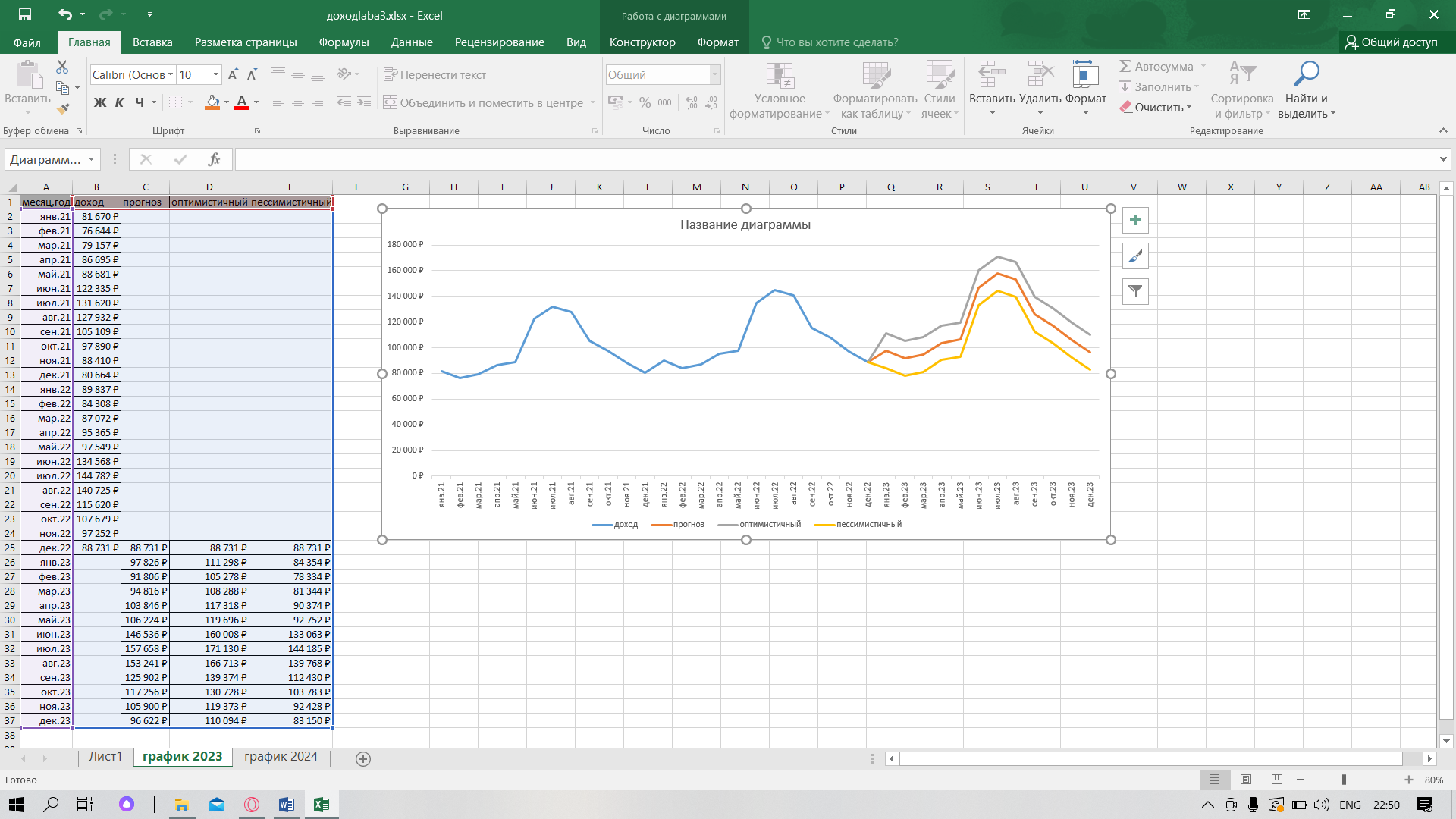


Результат: 

***Для расчета пессимистичного прогноза в 2023 году нужно из нейтрального прогноза в искомом месяце вычесть отклонение***

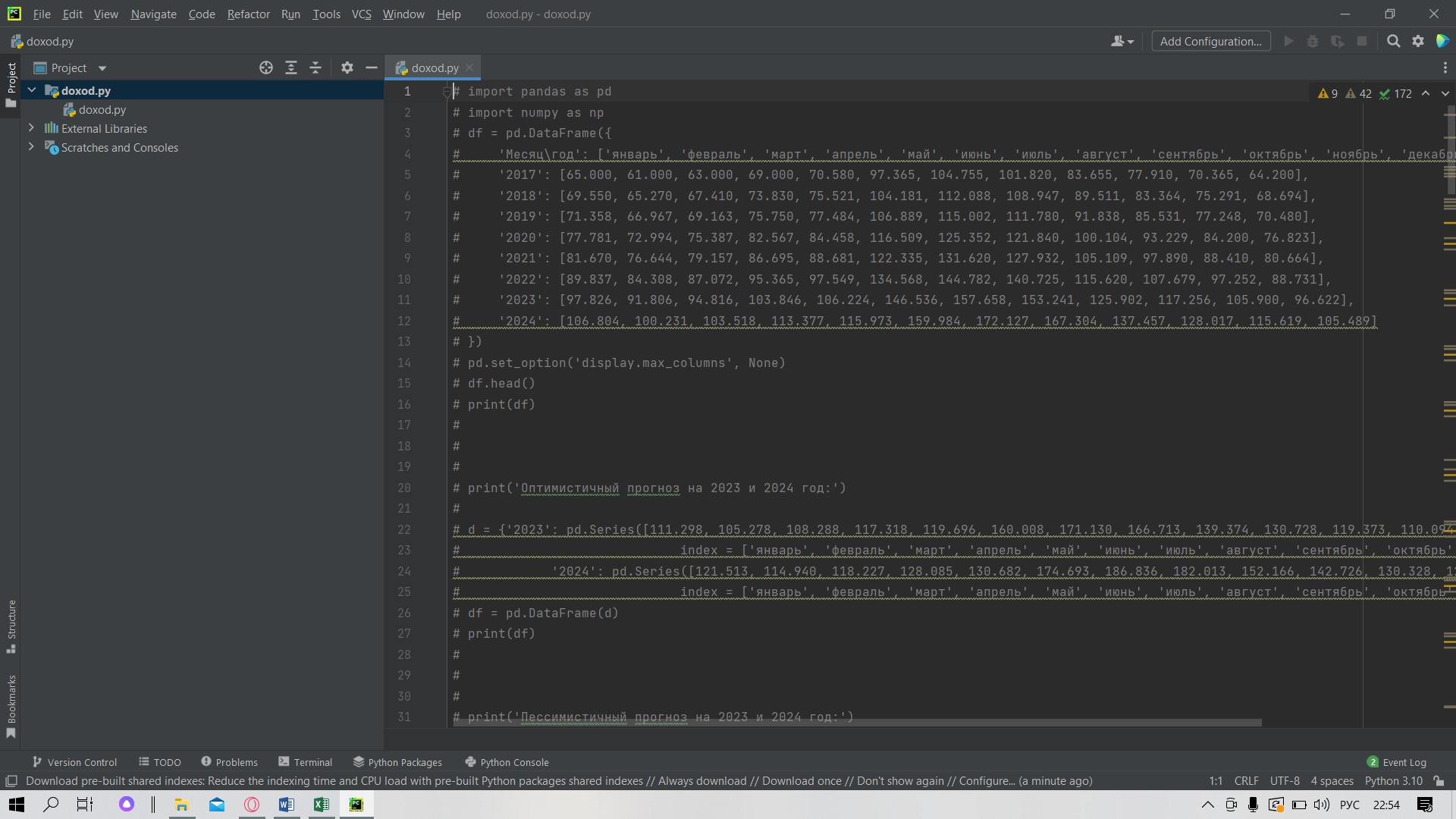


***Построение диаграммы***

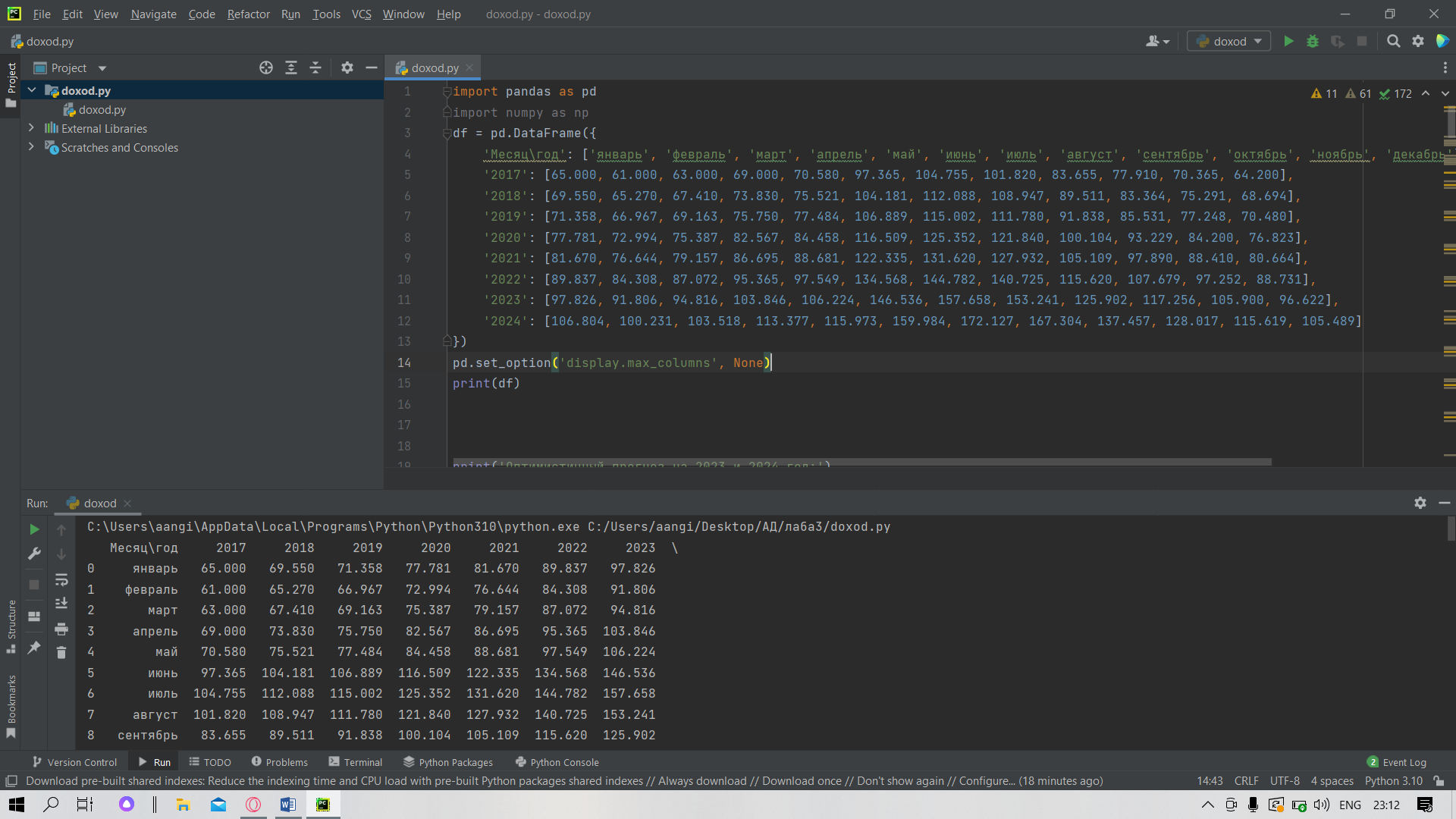


***Python***

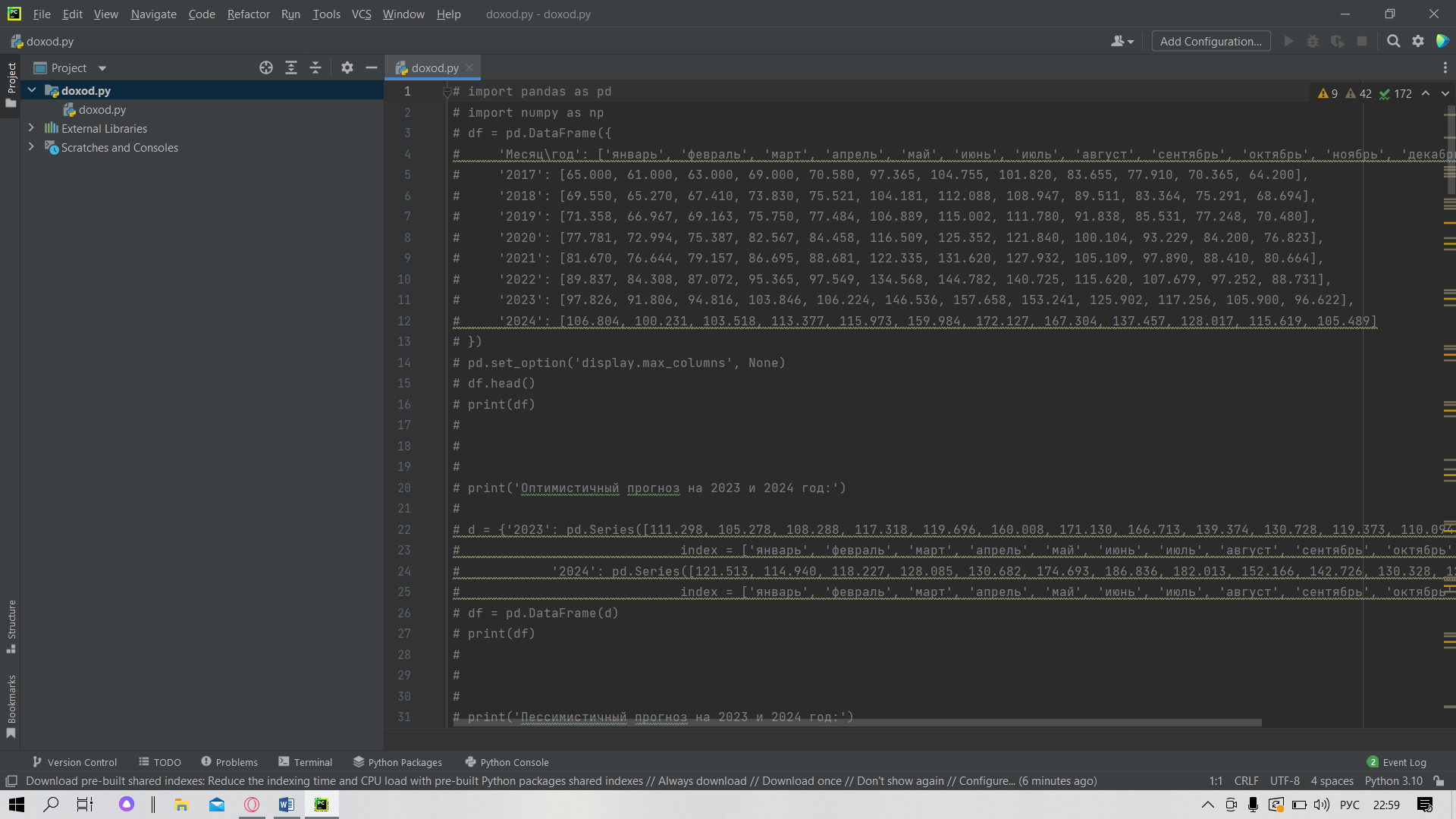
Импортирую Pandas и NumPy



Создаю словарь и передаю его в качестве аргумента в метод DataFrame(). Распечатываю DataFrame.



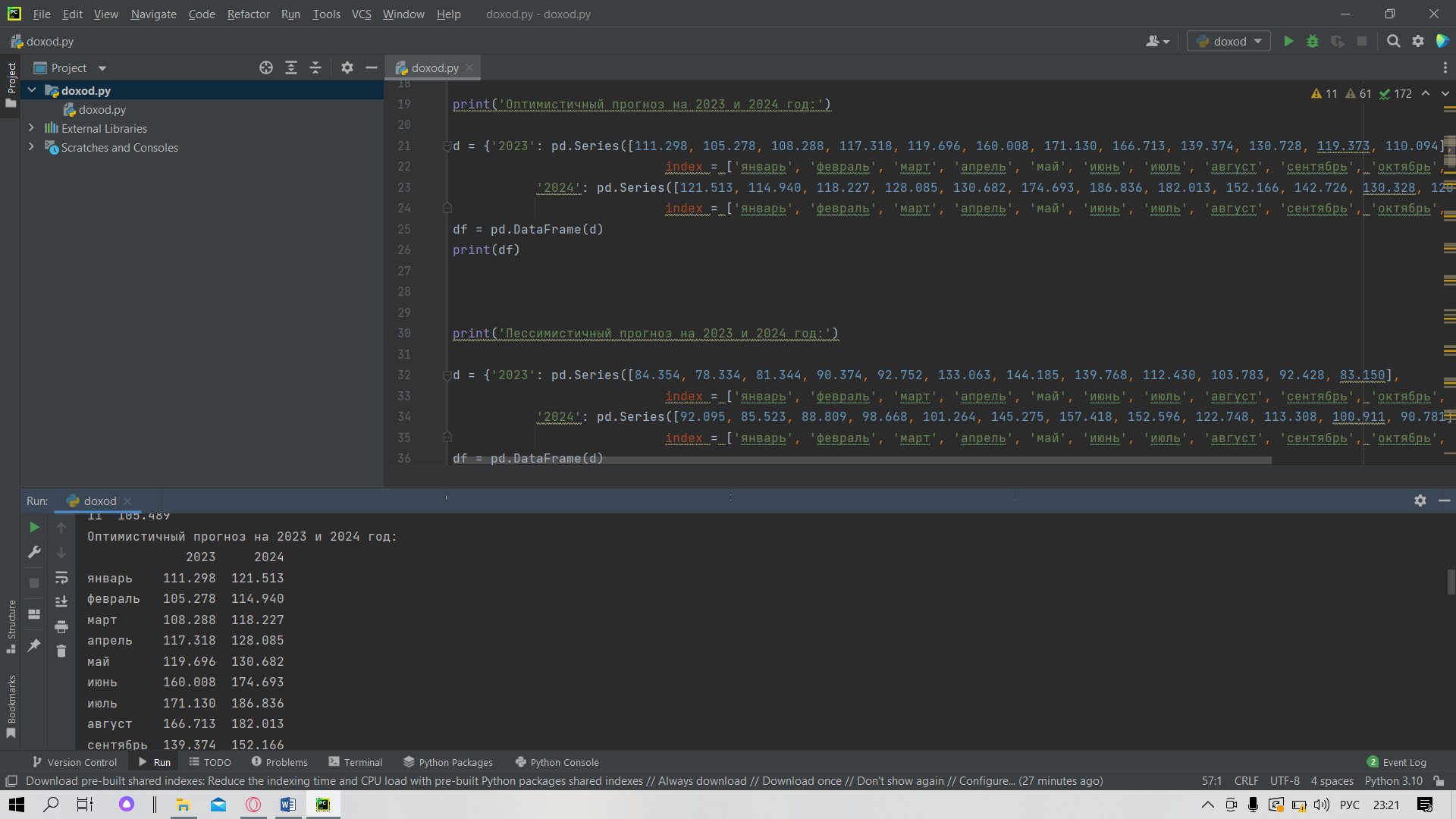
**Чтобы выводились все стобцы без многоточий использую опцию “display.max\_columns” и устанавливаю значение None.**



Результат:



Для оптимистичного и пессимистичного прогнозов создаю DataFrame, передав словарь Series. Series — это структура, используемая для работы с последовательностью одномерных данных. Индексам присваиваю месяца:

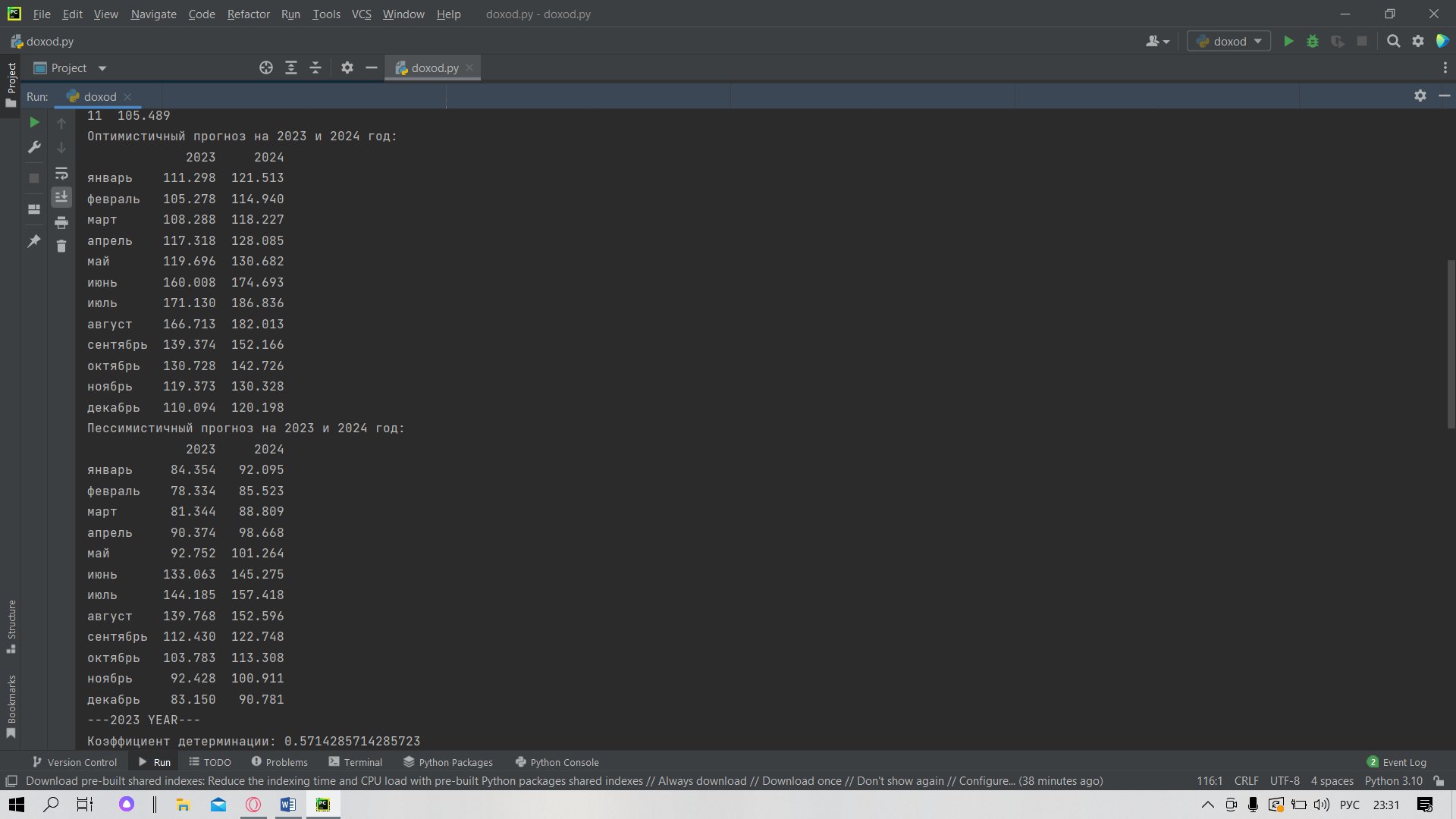


Создаю словарь и передаю его в качестве аргумента в метод DataFrame(). Распечатываю DataFrame:



Аналогично для пессимистичного прогноза

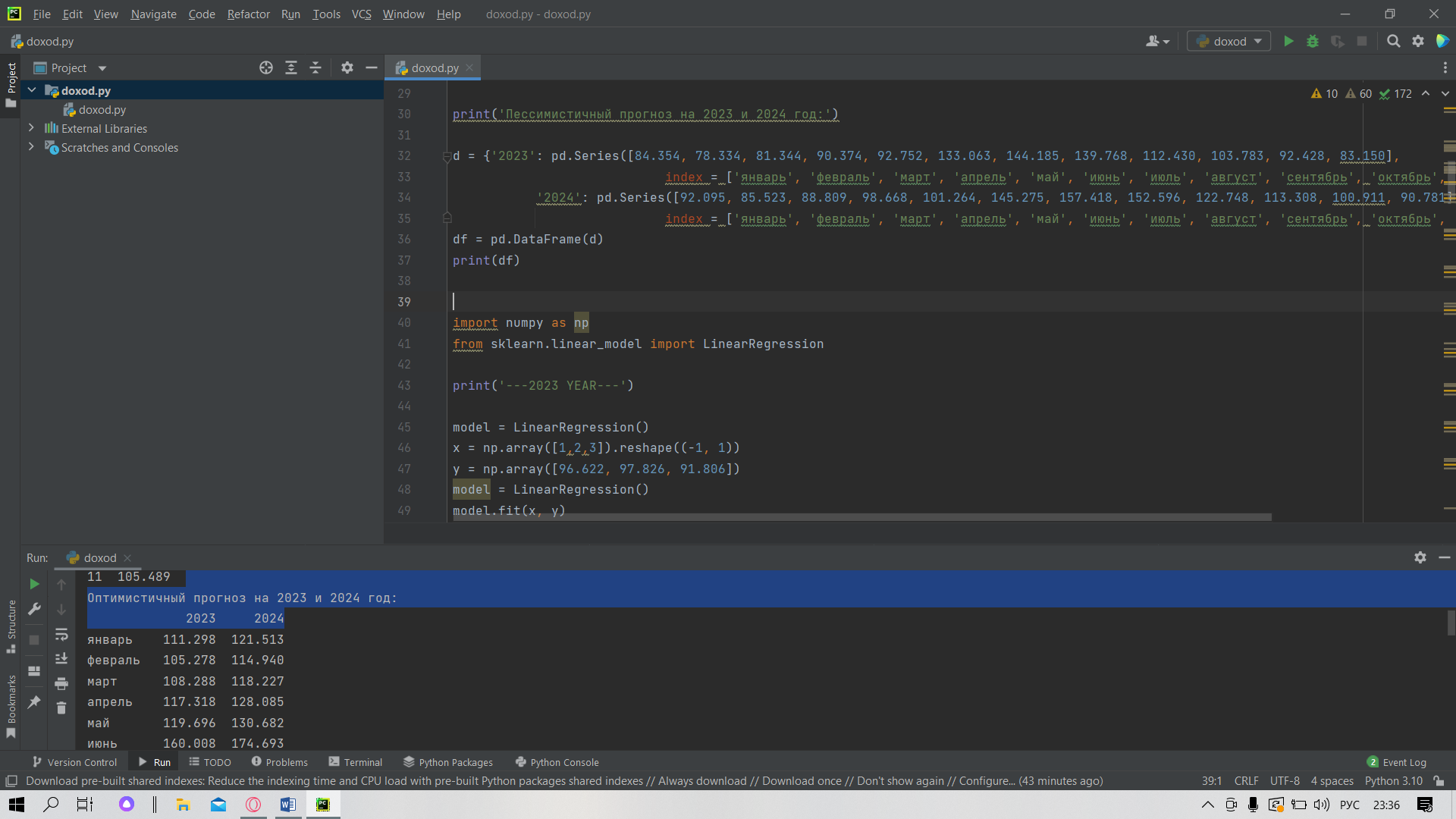
Результат:



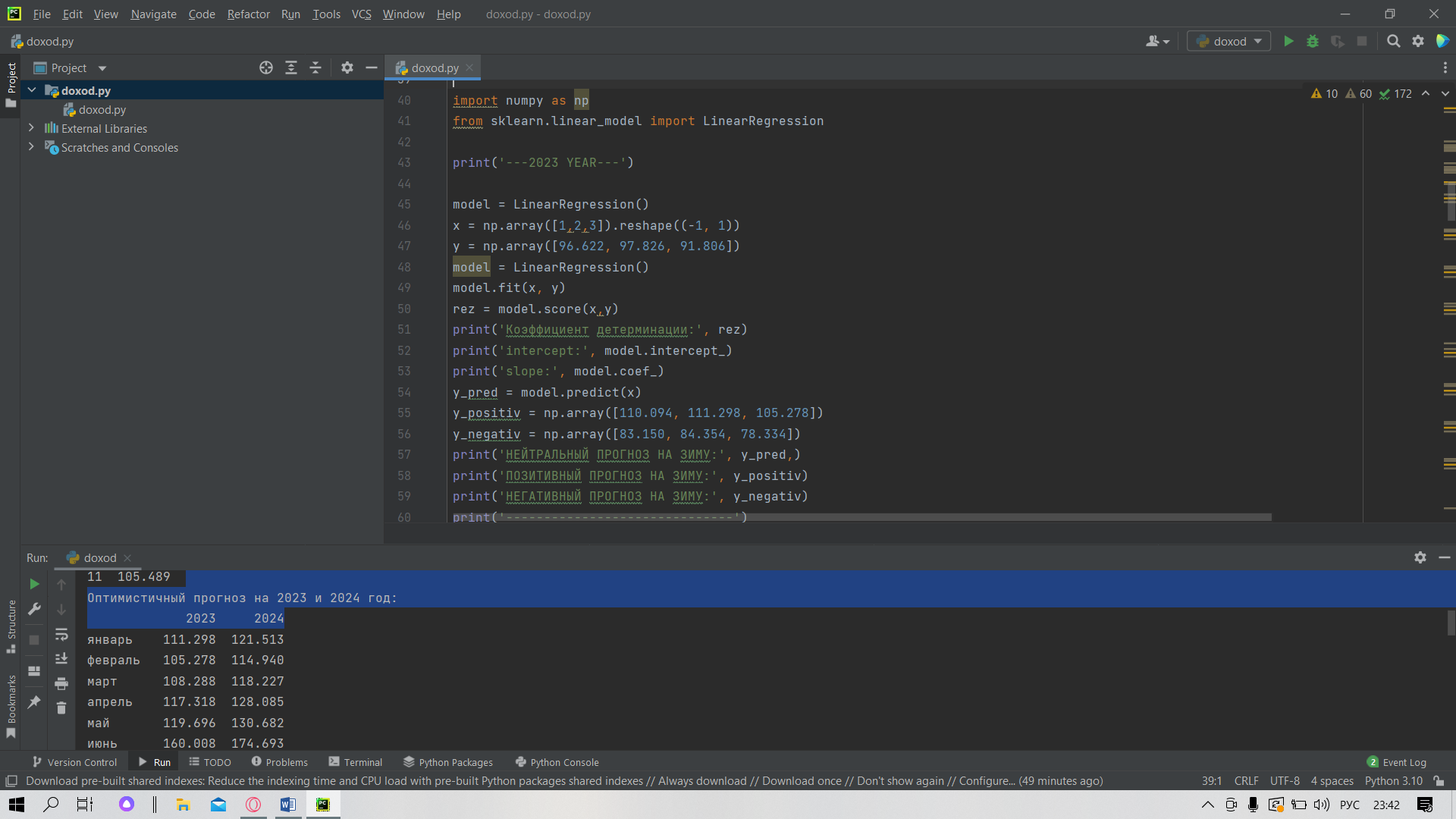
Линейная регрессия

Регрессия ищет отношения между переменными. Она полезна **для** **прогнозирования ответа** на новые условия.

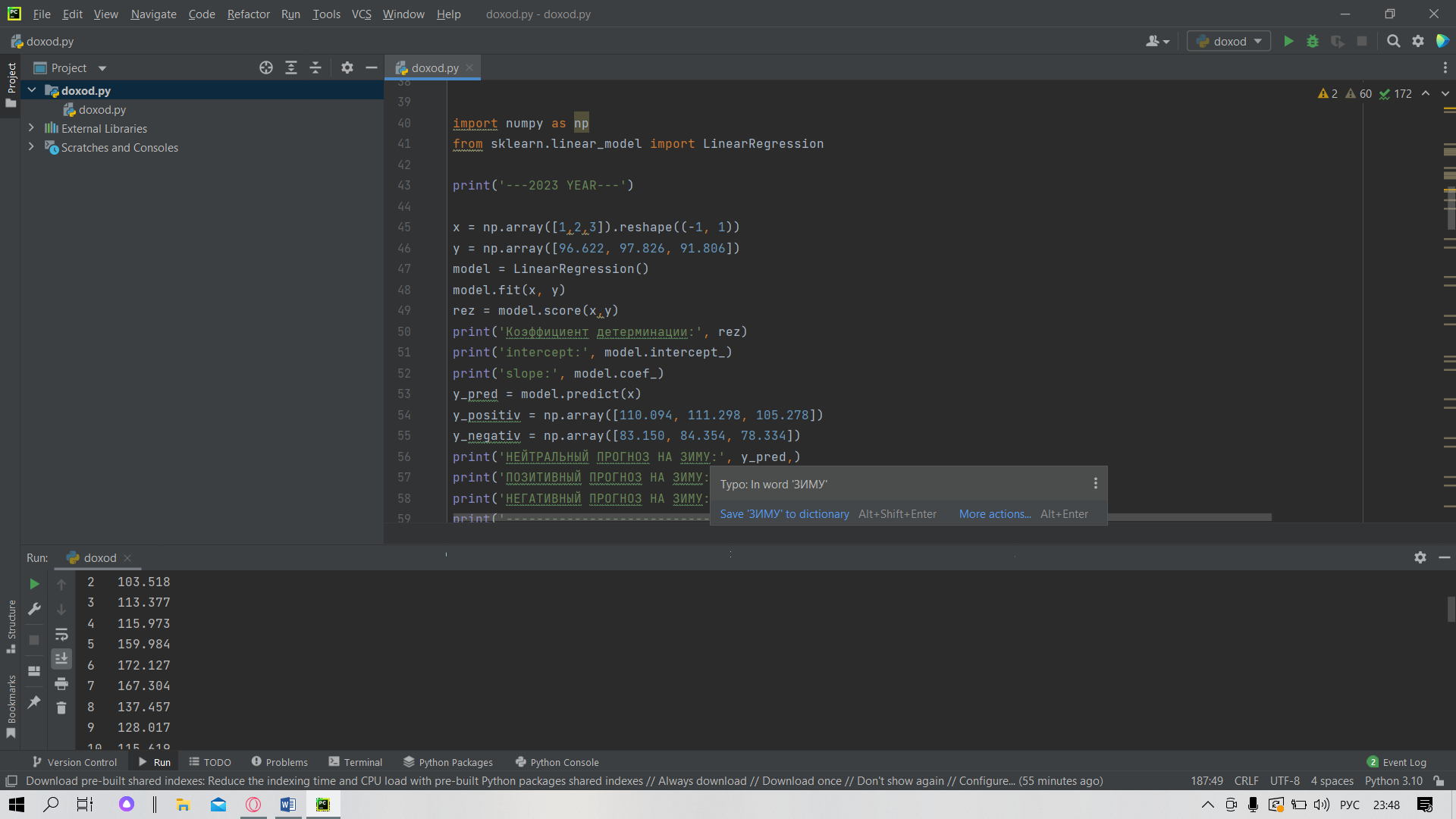
Импортирую пакет NumPy и класс LinearRegression из sklearn.linear\_model:



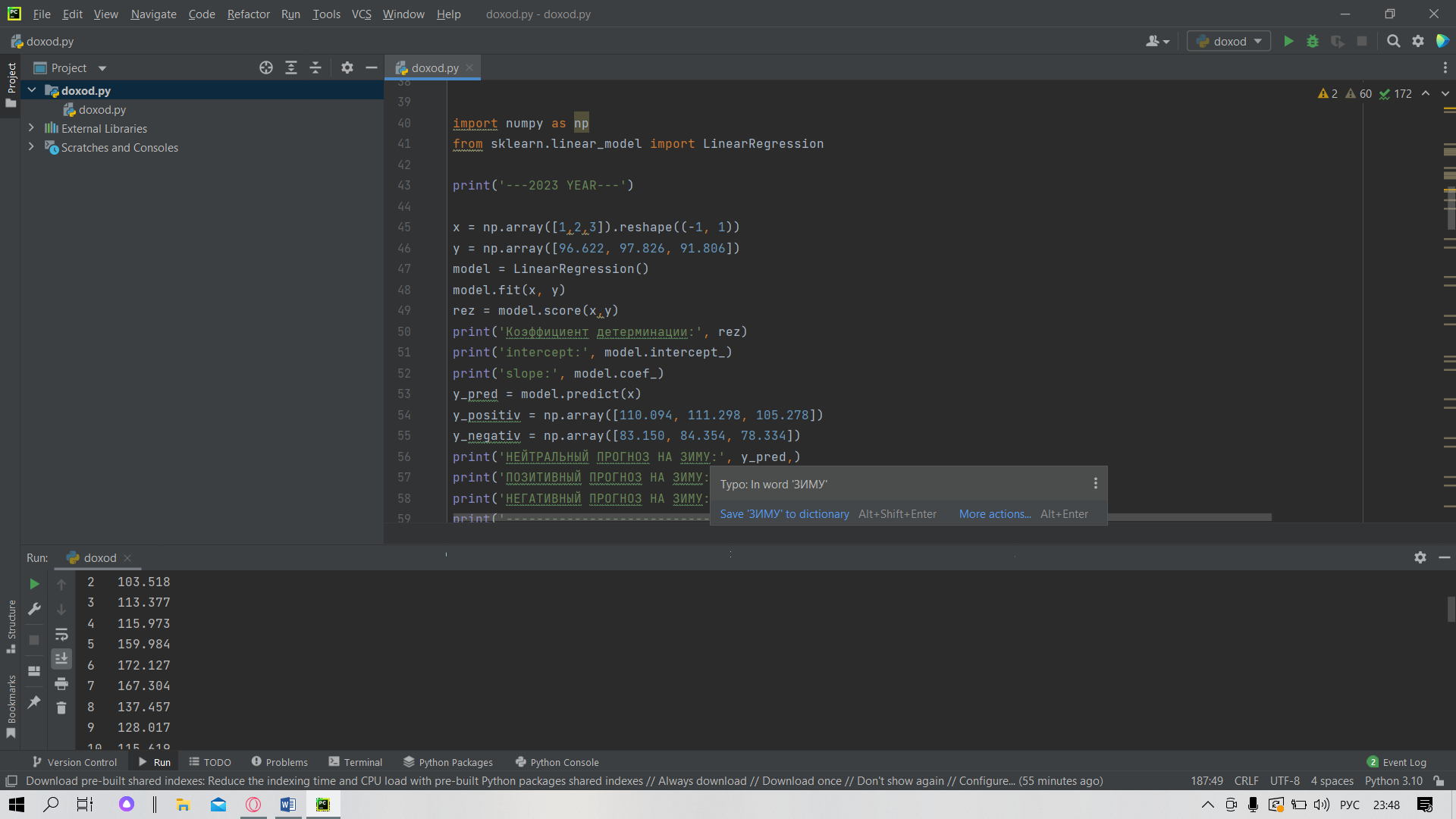
### **Предоставляю данные.** Входы (регрессоры, x) и выход (предиктор, y). Вызываю .reshape() на **x**, потому что этот массив должен быть **двумерным**или более точным – иметь **одну колонку и необходимое количество рядов**. Это определяет аргумент **(-1, 1)**:



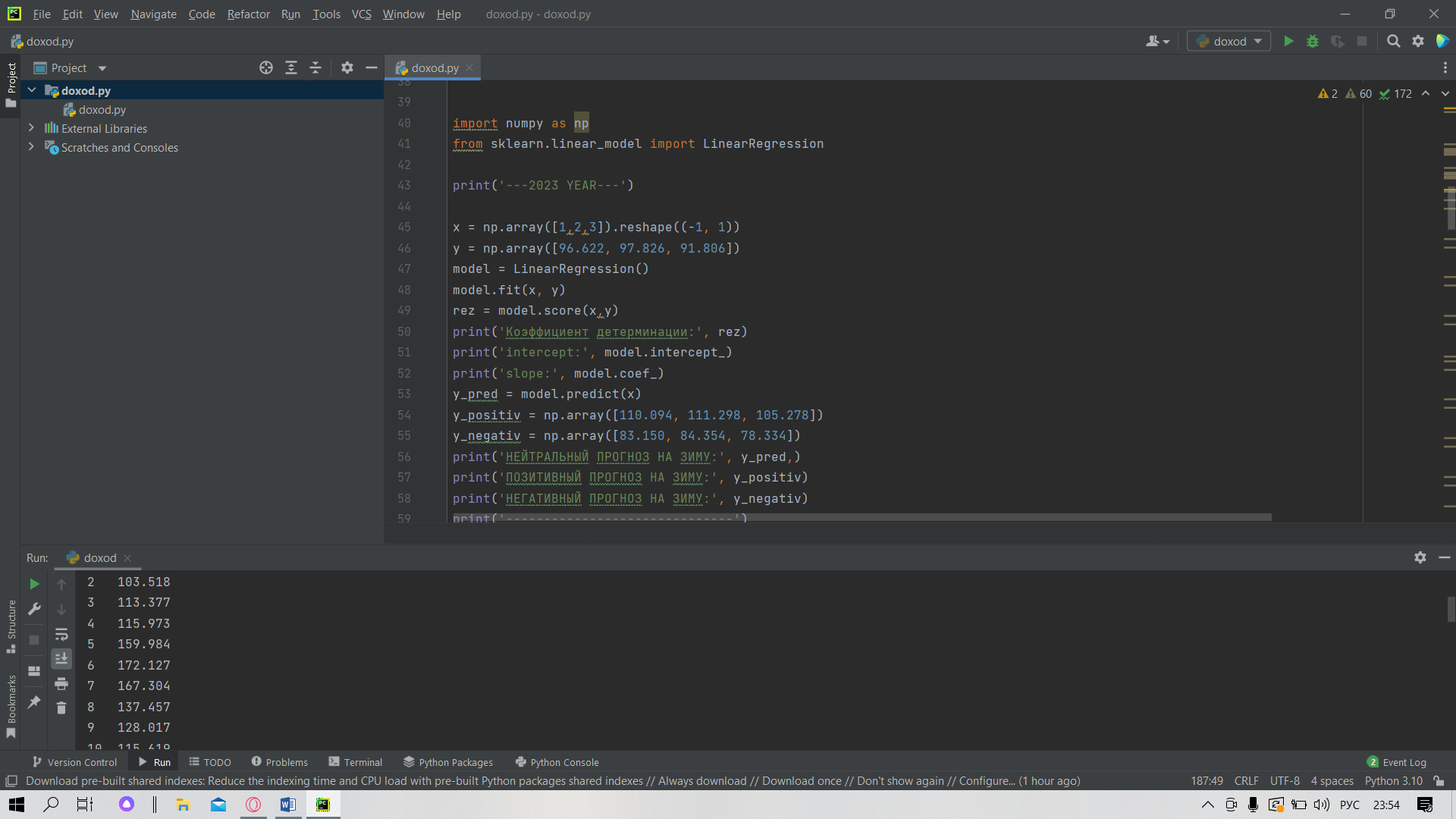
Создаю переменную model в качестве экземпляра LinearRegression:



С помощью .fit() вычисляю оптимальные значение весов b₀ и b₁, используя существующие вход и выход (**x** и **y**) в качестве аргументов. Другими словами, .fit() **совмещает модель**.

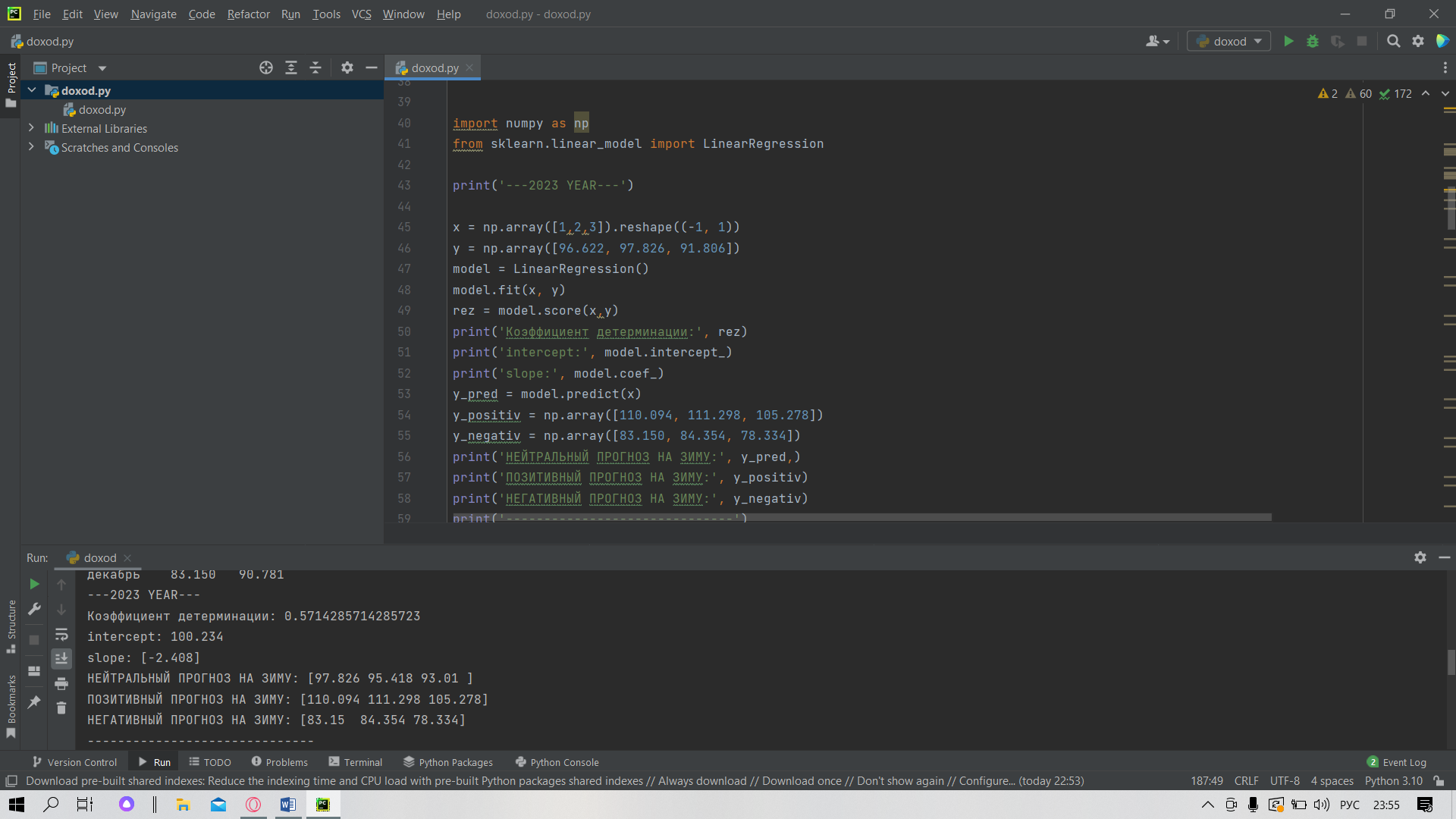


 получаю коэффициент детерминации с помощью .score(), вызванной на model:

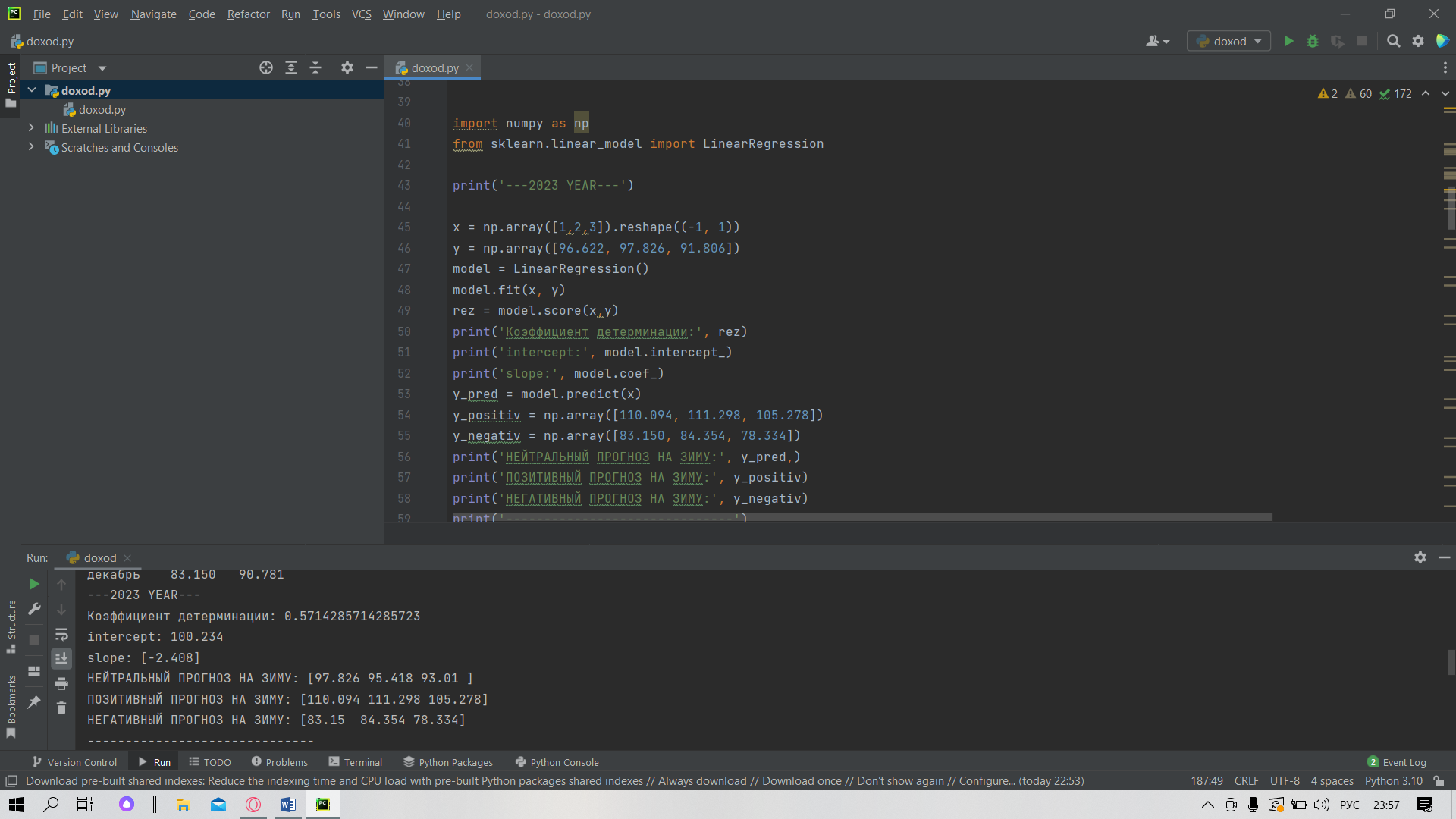


.score() принимает в качестве аргументов предсказатель **x** и регрессор **y**, и возвращает значение R².

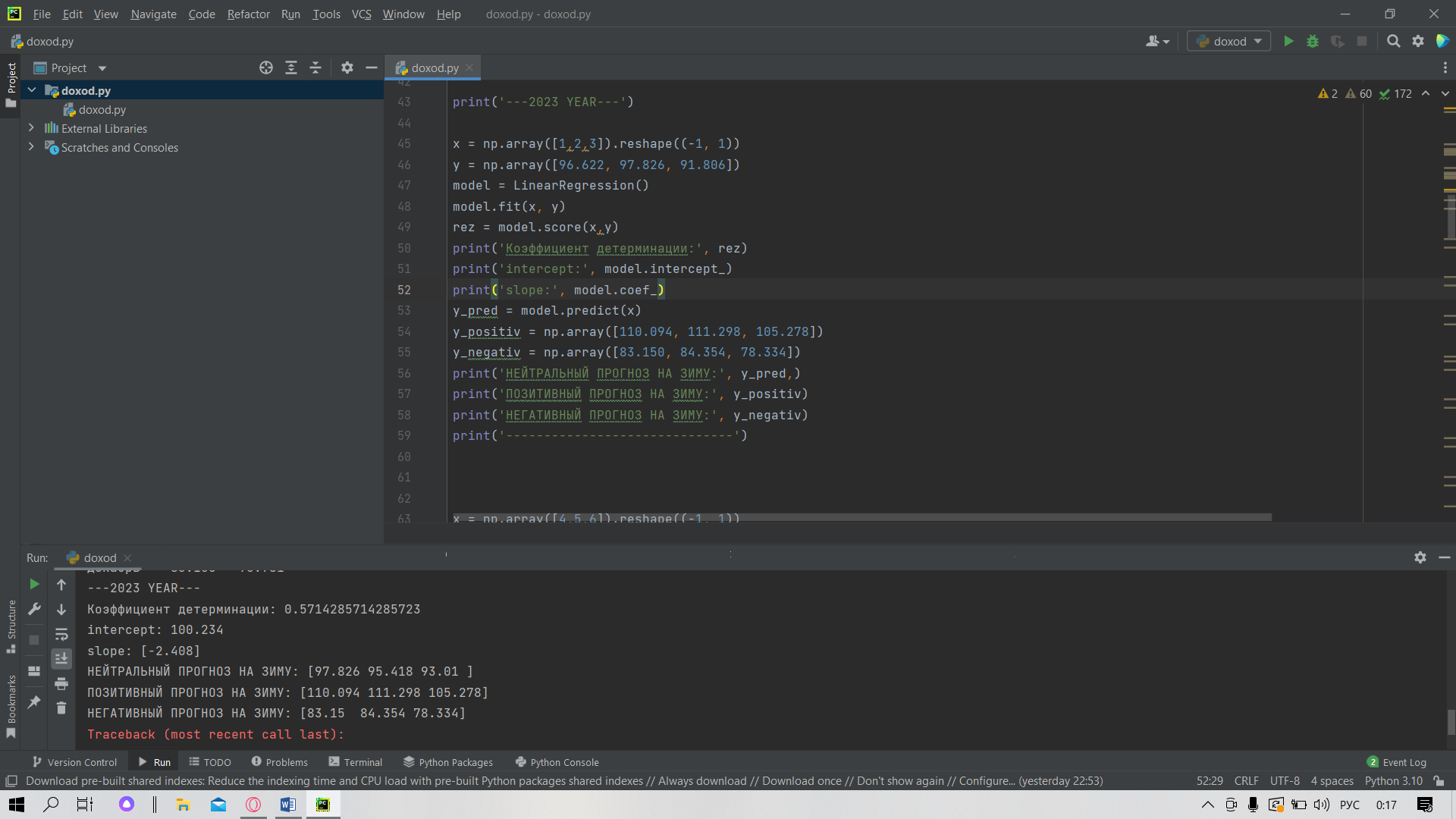
Вывод:



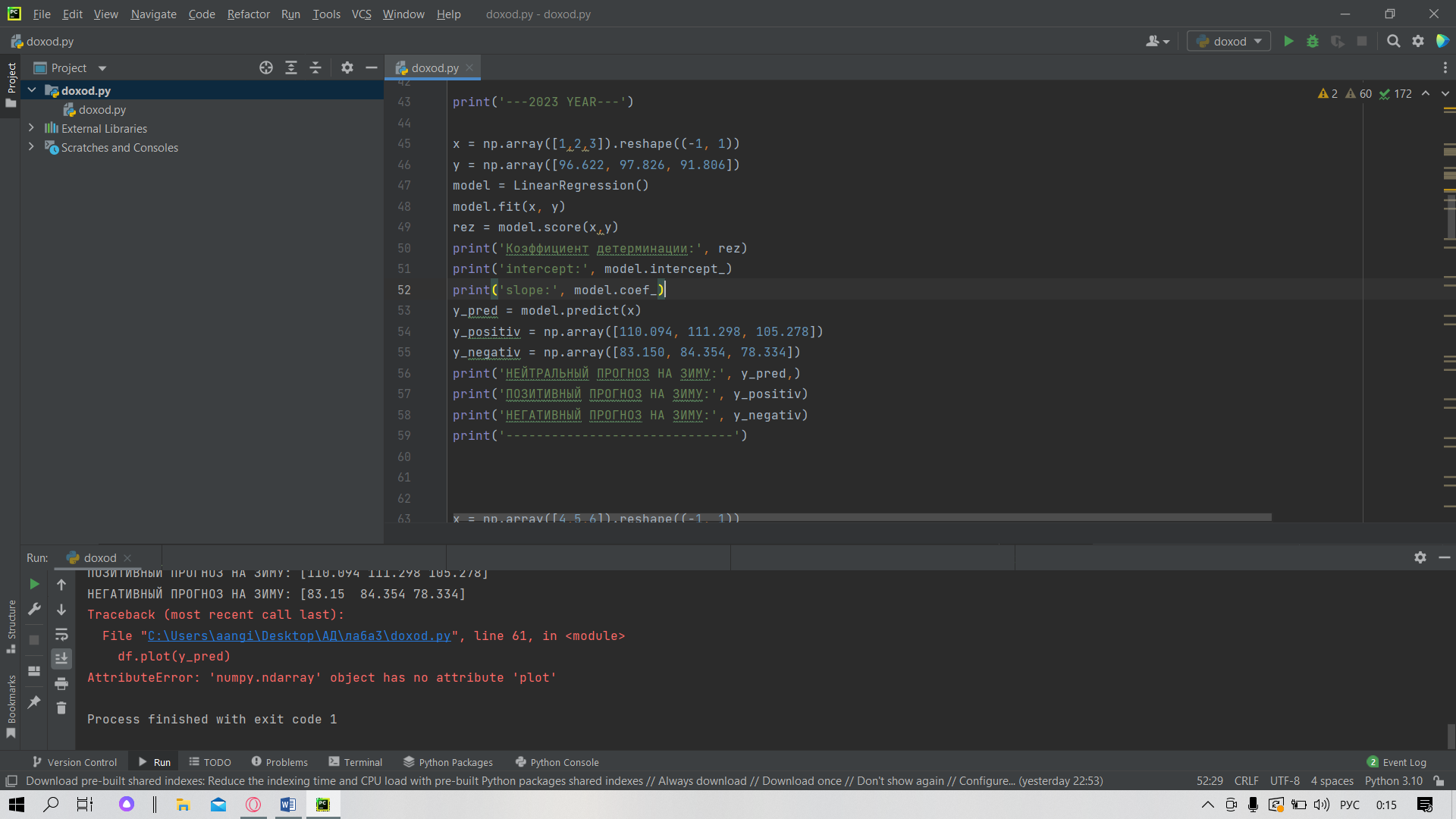
model содержит атрибуты .intercept\_, который представляет собой коэффициент, и b₀ с .coef\_, которые представляют b₁.



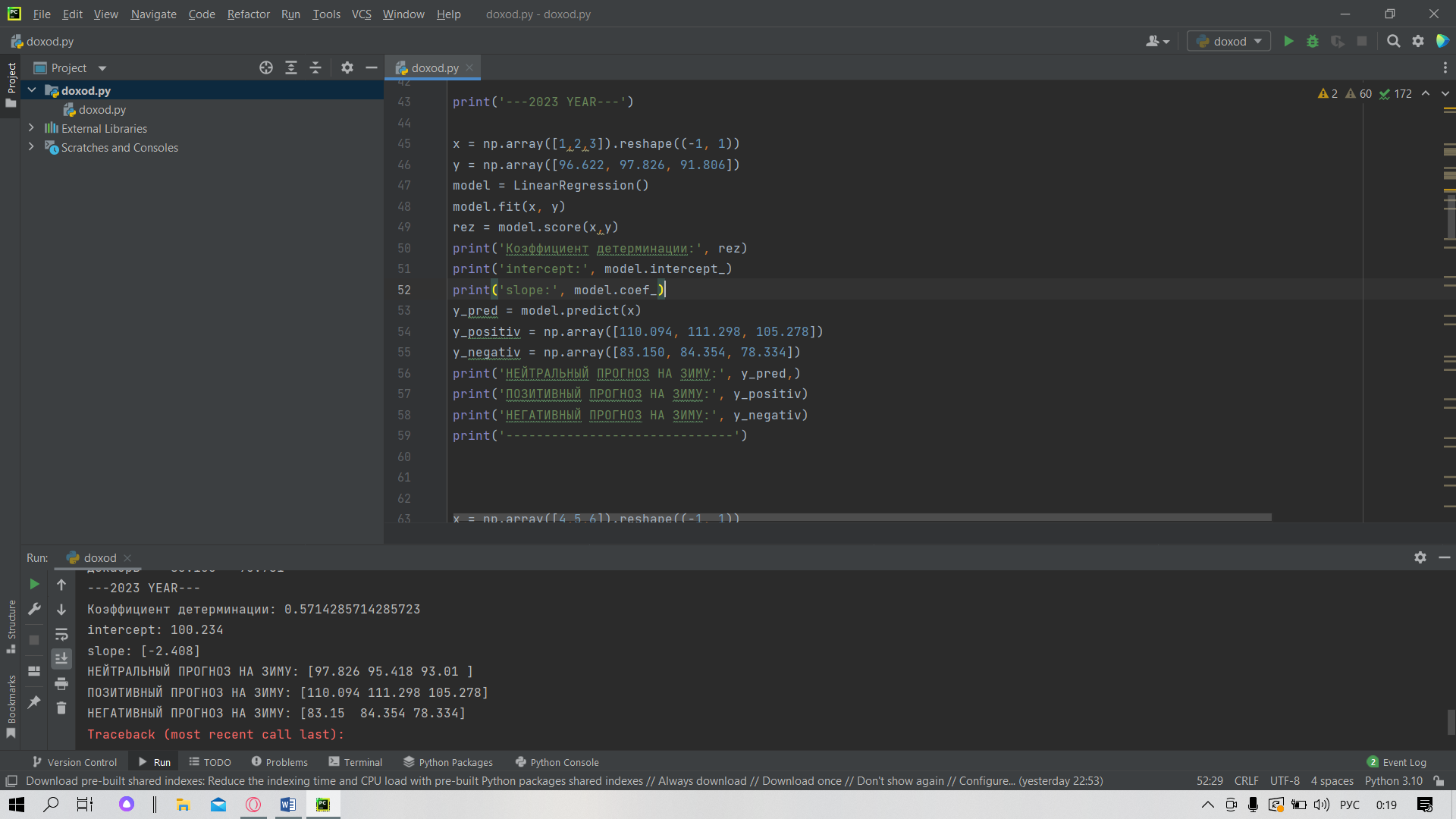
**Вывод: получаю** b₀(насколько прямая опущена или поднята на графике, величина пересечения) и b₁ (наклон прямой).



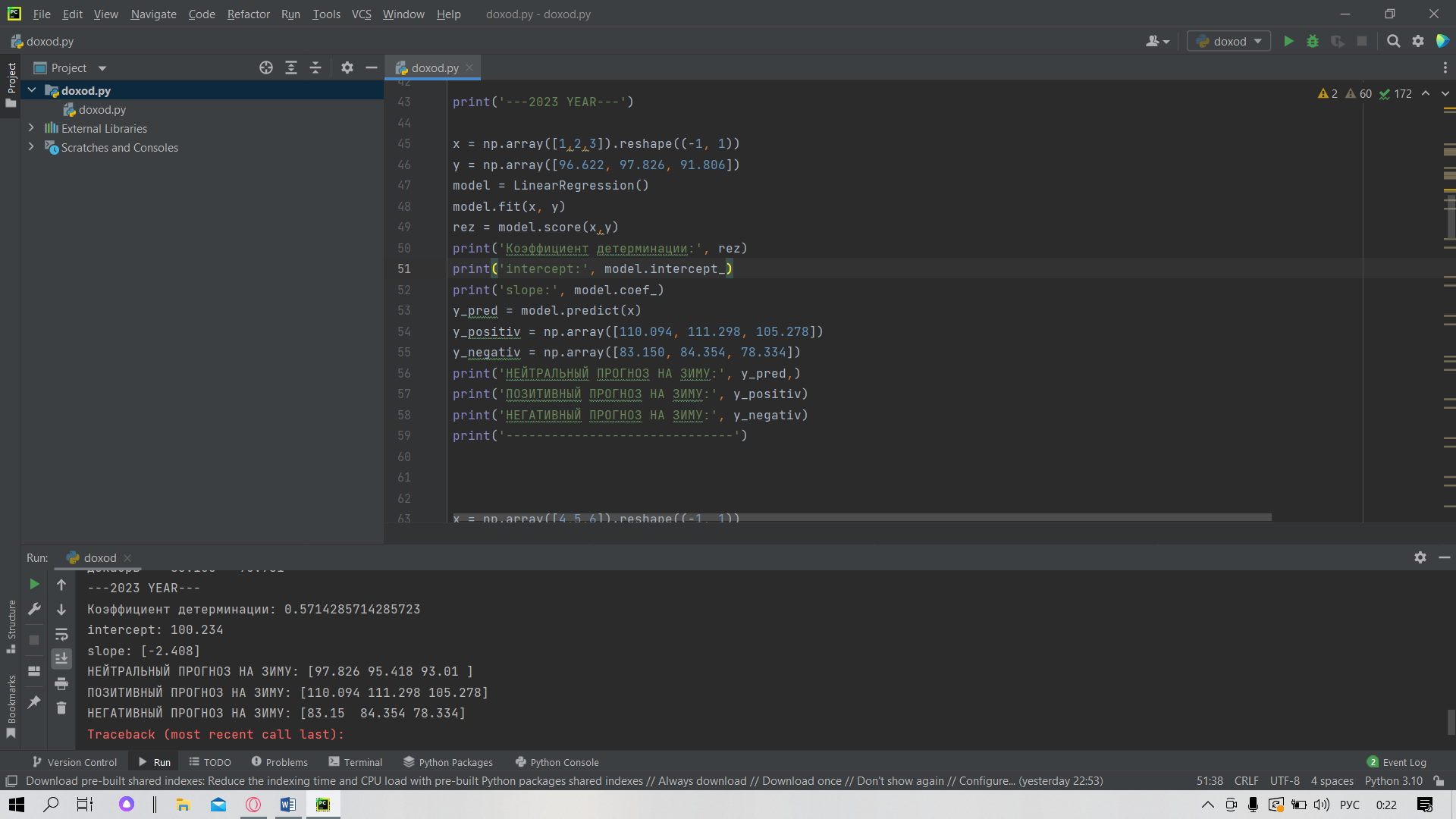
Применяя .predict(), передаю регрессор в качестве аргумента:



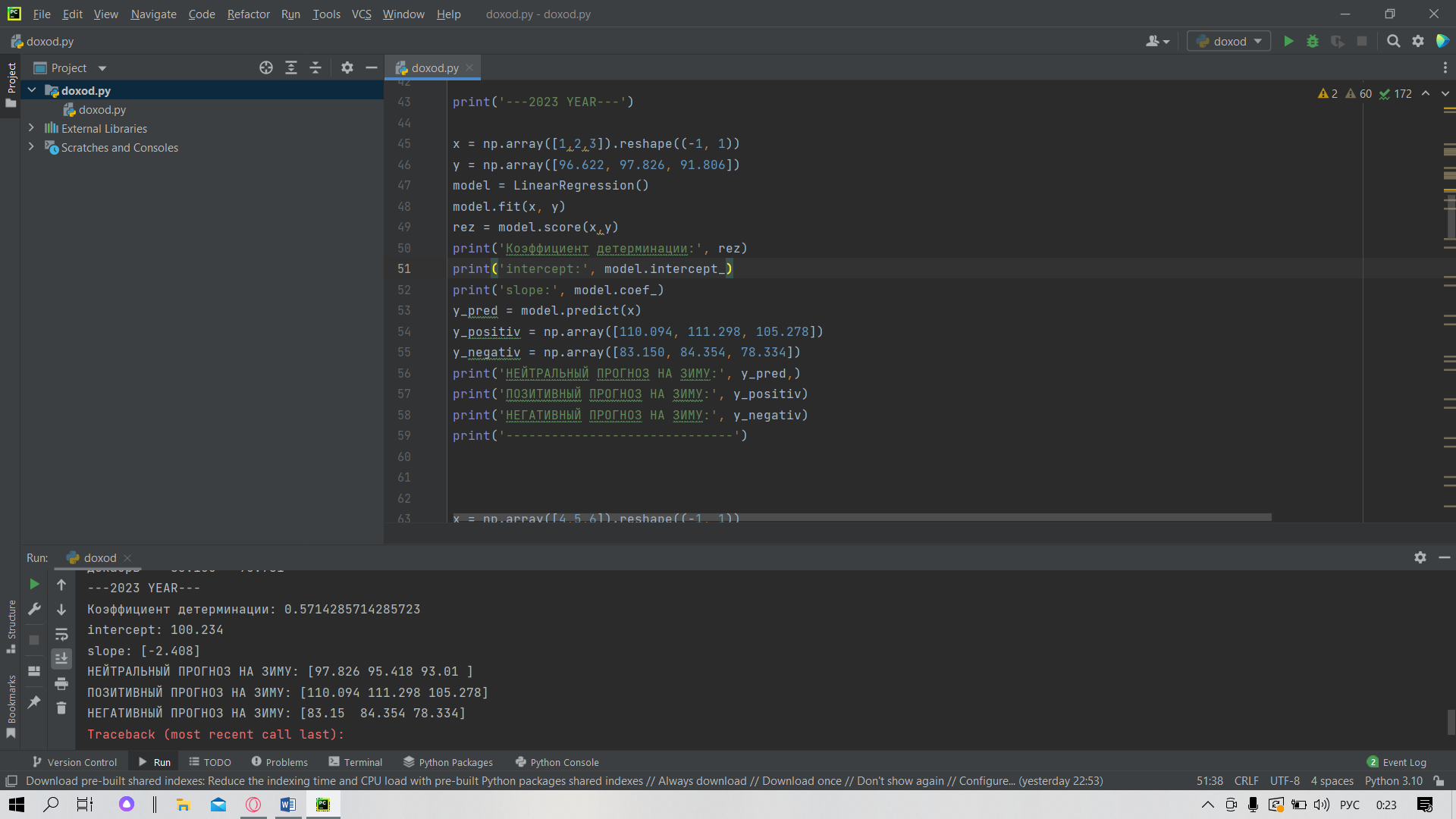
получаю соответствующий предсказанный ответ:



Присваиваю y новые значения пессимистичного и оптимистичного прогнозов:



Получаю прогноз:



Вывод:

