**Домашняя работа №1**

**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.

**Решение:**

import numpy as np

a=np.array([[1,2,3,3,1],[6,8,11,10,7]])

array([[ 1, 2, 3, 3, 1],

[ 6, 8, 11, 10, 7]])

At=a.transpose()

At

array([[ 1, 6],

[ 2, 8],

[ 3, 11],

[ 3, 10],

[ 1, 7]])

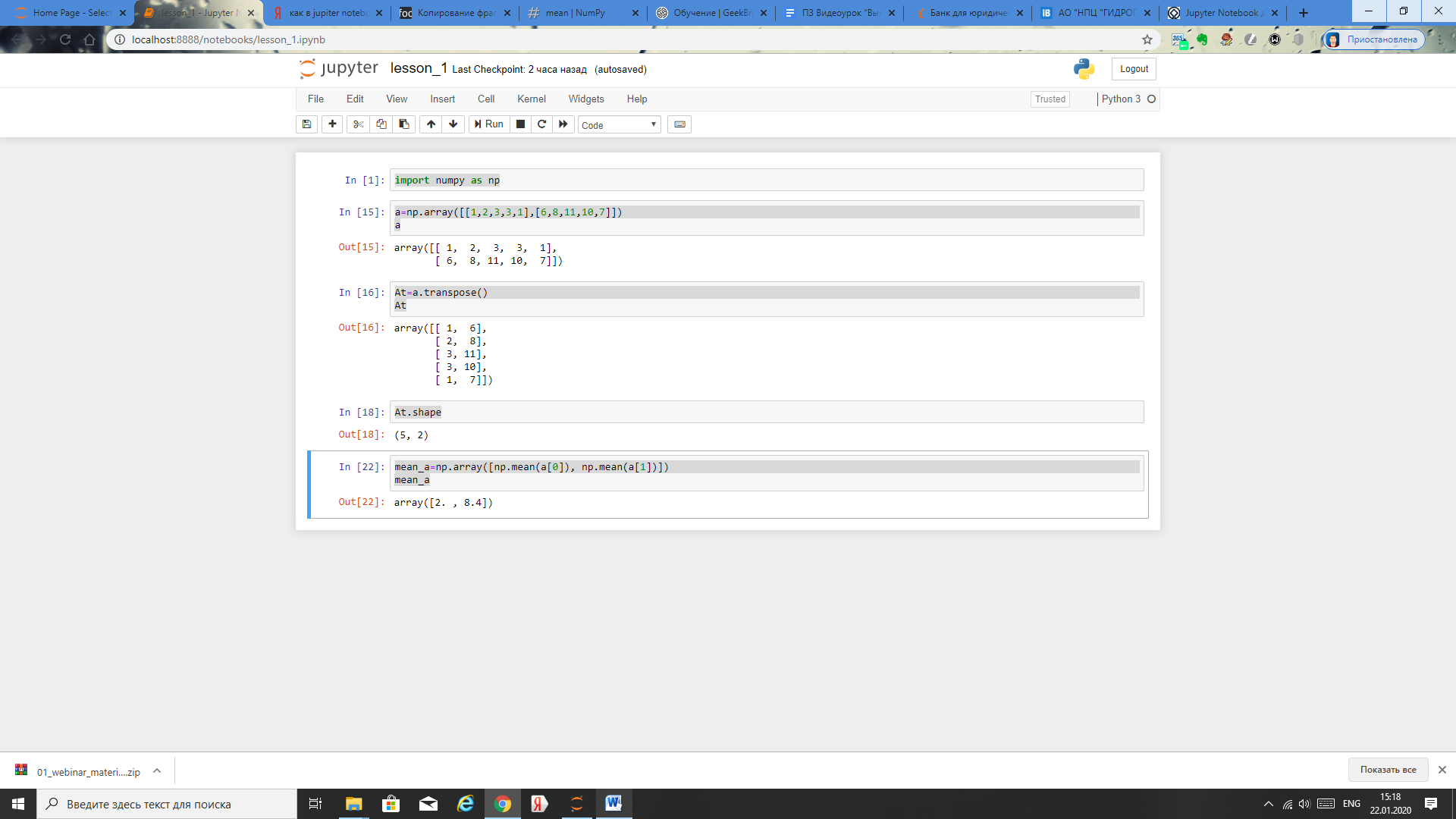
At.shape

(5, 2)

mean\_a=np.array([np.mean(a[0]), np.mean(a[1])])

mean\_a

array([2. , 8.4])



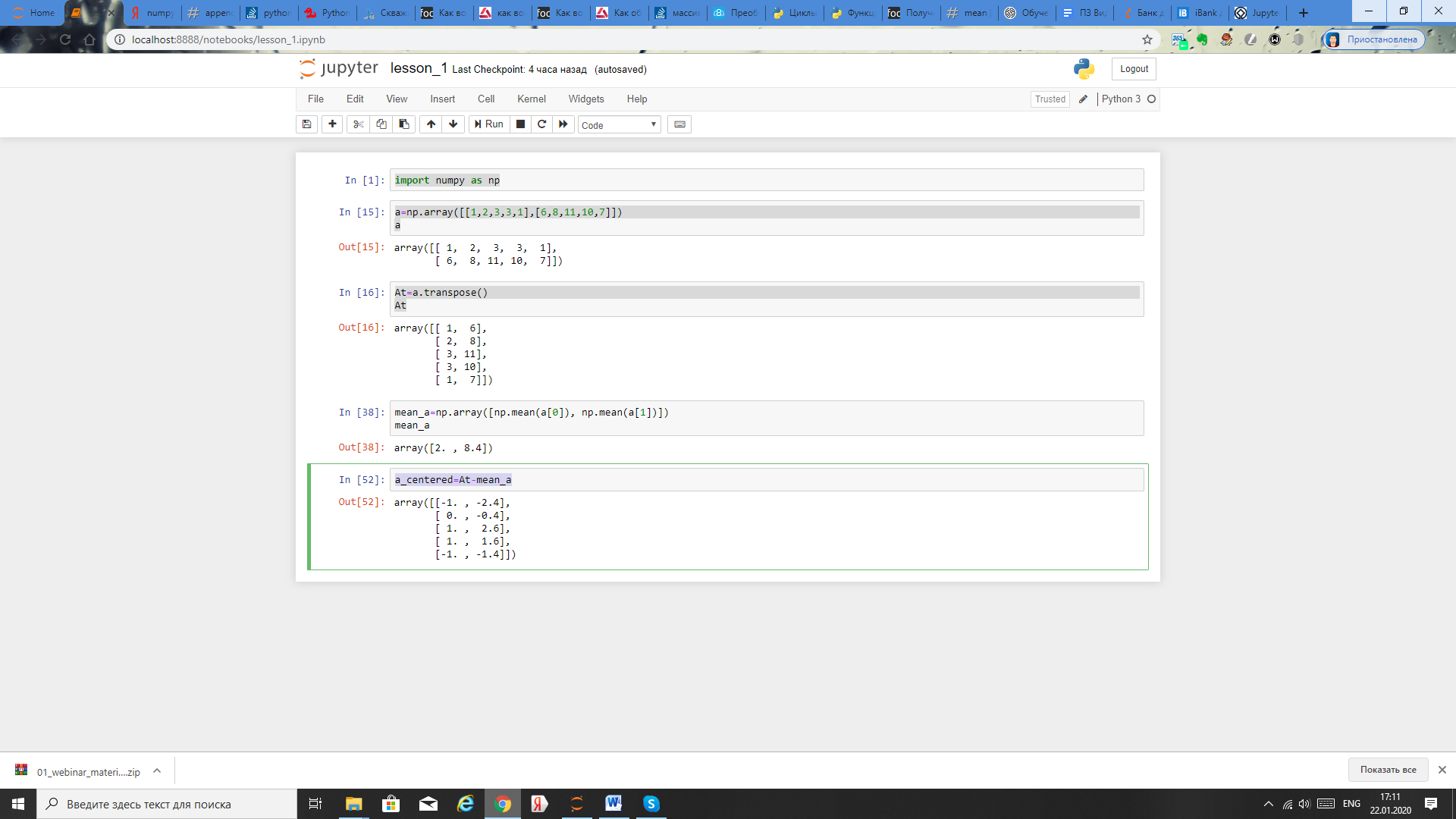
**Задание 2**

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

**Решение:**

Согласно концепции Broadcasting

a\_centered=At-mean\_a



**Задание 3**

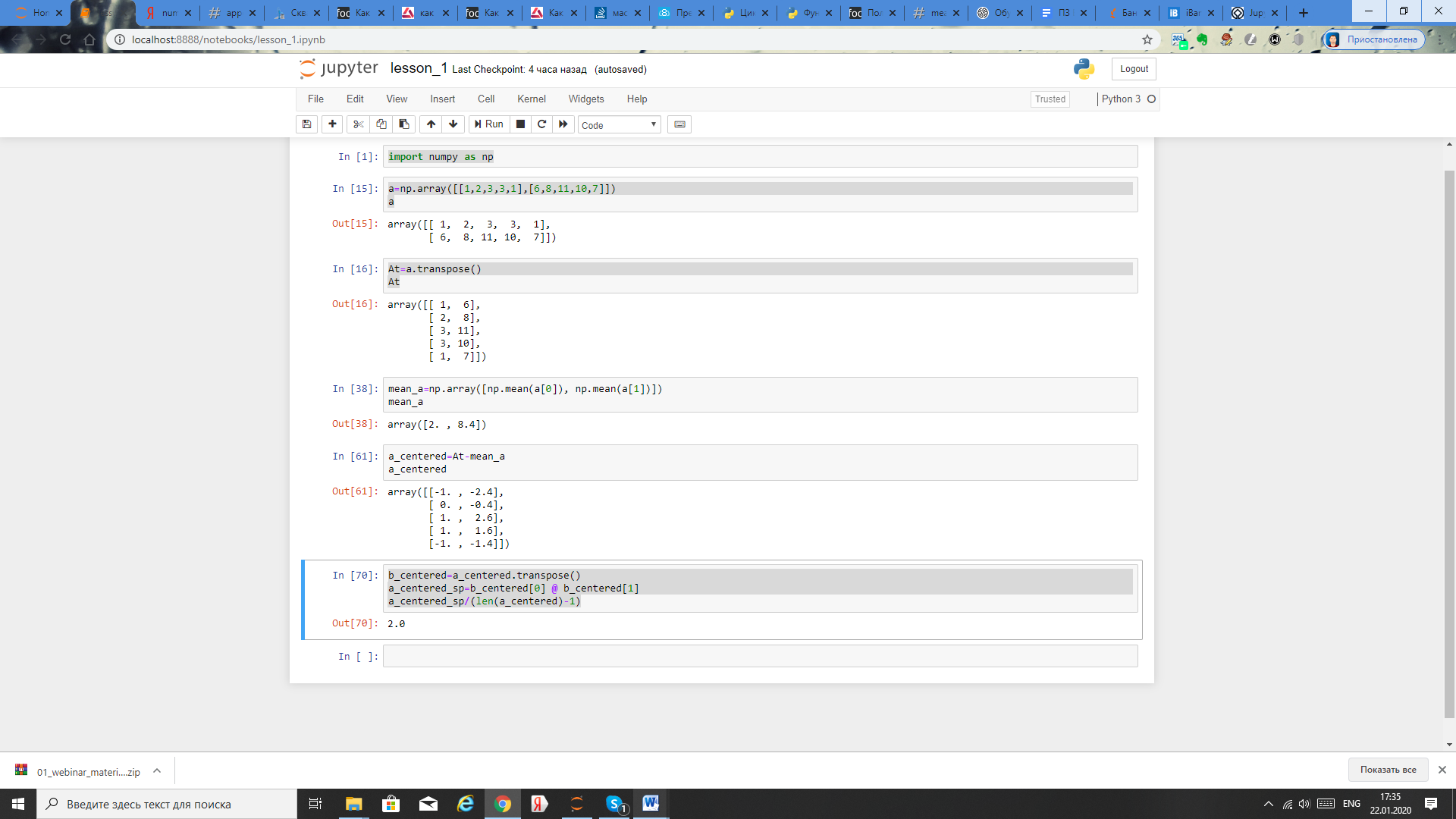
Найдите скалярное произведение столбцов массива a\_centered. В результате должна получиться величина a\_centered\_sp. Затем поделите a\_centered\_sp на N-1, где N - число наблюдений.

**Решение:**

b\_centered=a\_centered.transpose()

a\_centered\_sp=b\_centered[0] @ b\_centered[1]

a\_centered\_sp/(len(a\_centered)-1)



**Задание 4\*\***

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

Подробнее узнать о ковариации можно здесь:

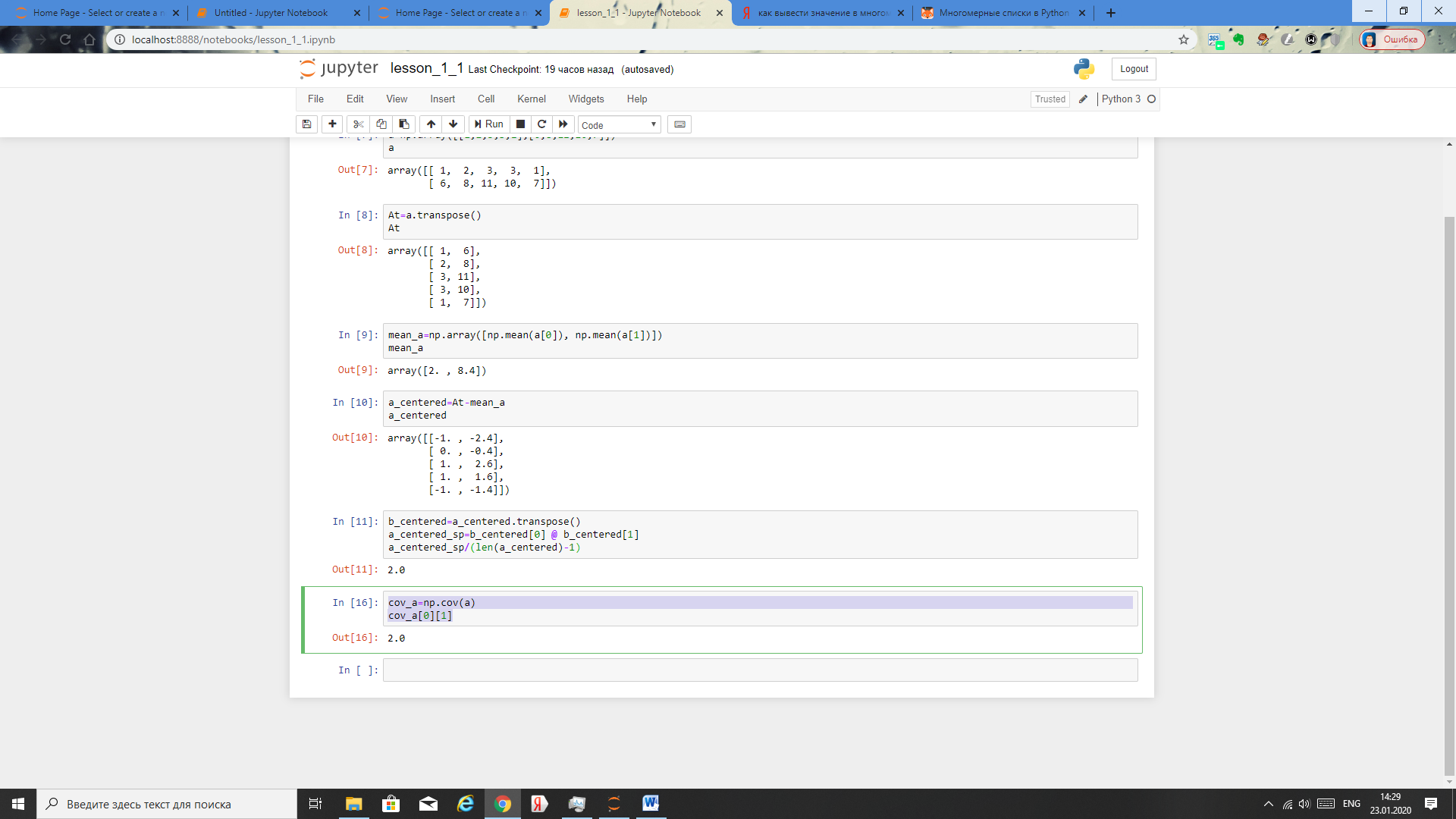
<https://studopedia.ru/9_153900_viborochnaya-kovariatsiya-i-viborochnaya-dispersiya.html>

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 2x2) искомое значение ковариации будет равно элементу в строке с индексом 0 и столбце с индексом 1.

**Решение:**

cov\_a=np.cov(a)

cov\_a[0][1]



## *Тема “Работа с данными в Pandas”*

**Задание 1**

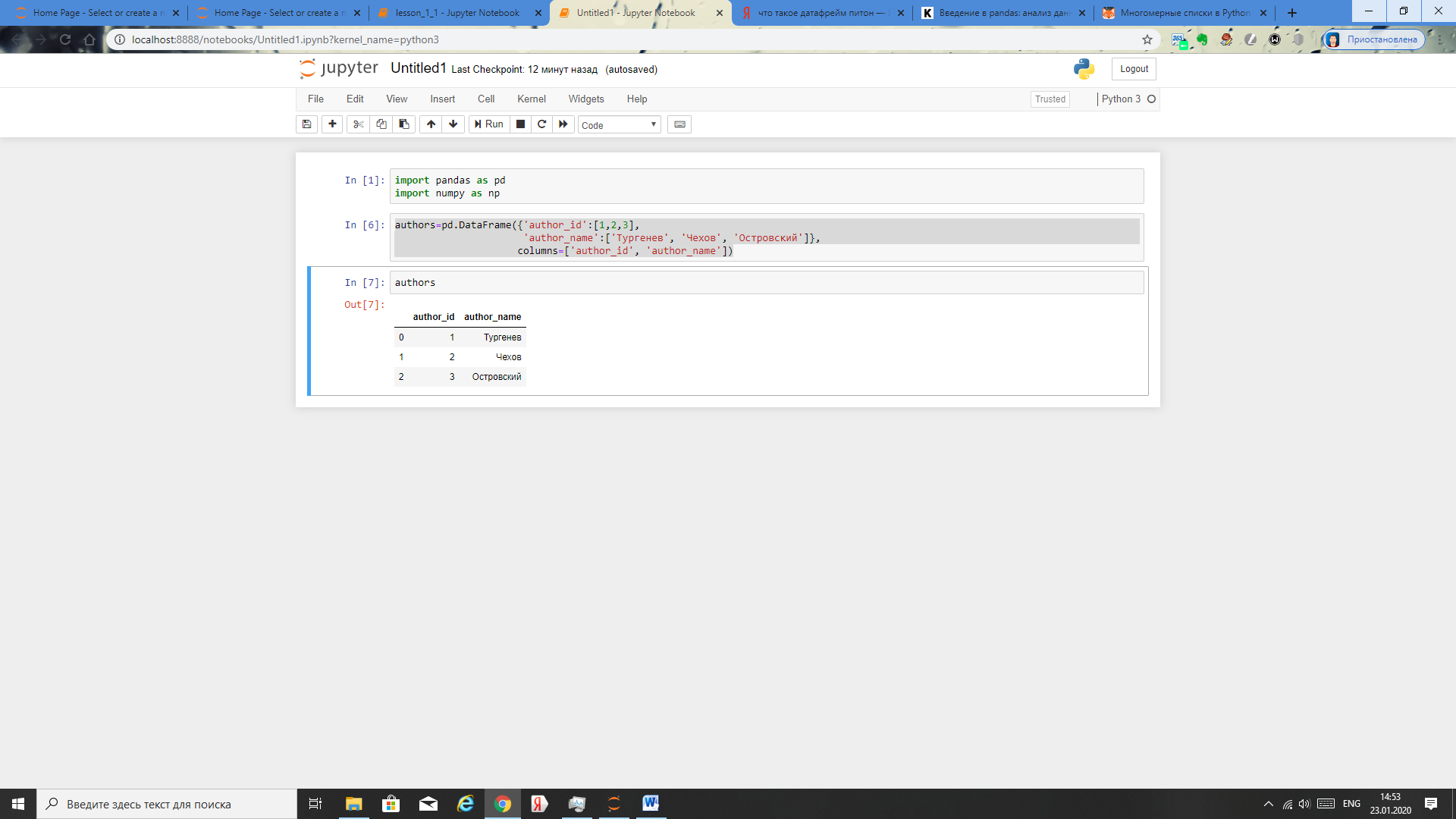
Импортируйте библиотеку Pandas и дайте ей псевдоним pd. Создайте датафрейм authors со столбцами author\_id и author\_name, в которых соответственно содержатся данные: [1, 2, 3] и ['Тургенев', 'Чехов', 'Островский'].

Решение:

authors=pd.DataFrame({'author\_id':[1,2,3],

'author\_name':['Тургенев', 'Чехов', 'Островский']},

columns=['author\_id', 'author\_name'])



Затем создайте датафрейм book cо столбцами author\_id, book\_title и price, в которых соответственно содержатся данные:

[1, 1, 1, 2, 2, 3, 3],

['Отцы и дети', 'Рудин', 'Дворянское гнездо', 'Толстый и тонкий', 'Дама с собачкой', 'Гроза', 'Таланты и поклонники'],

[450, 300, 350, 500, 450, 370, 290].

Решение:

book=pd.DataFrame({'author\_id':[1,1,1,2,2,3,3],

'book\_title':['Отцы и дети',

'Рудин',

'Дворянское гнездо',

'Толстый и тонкий',

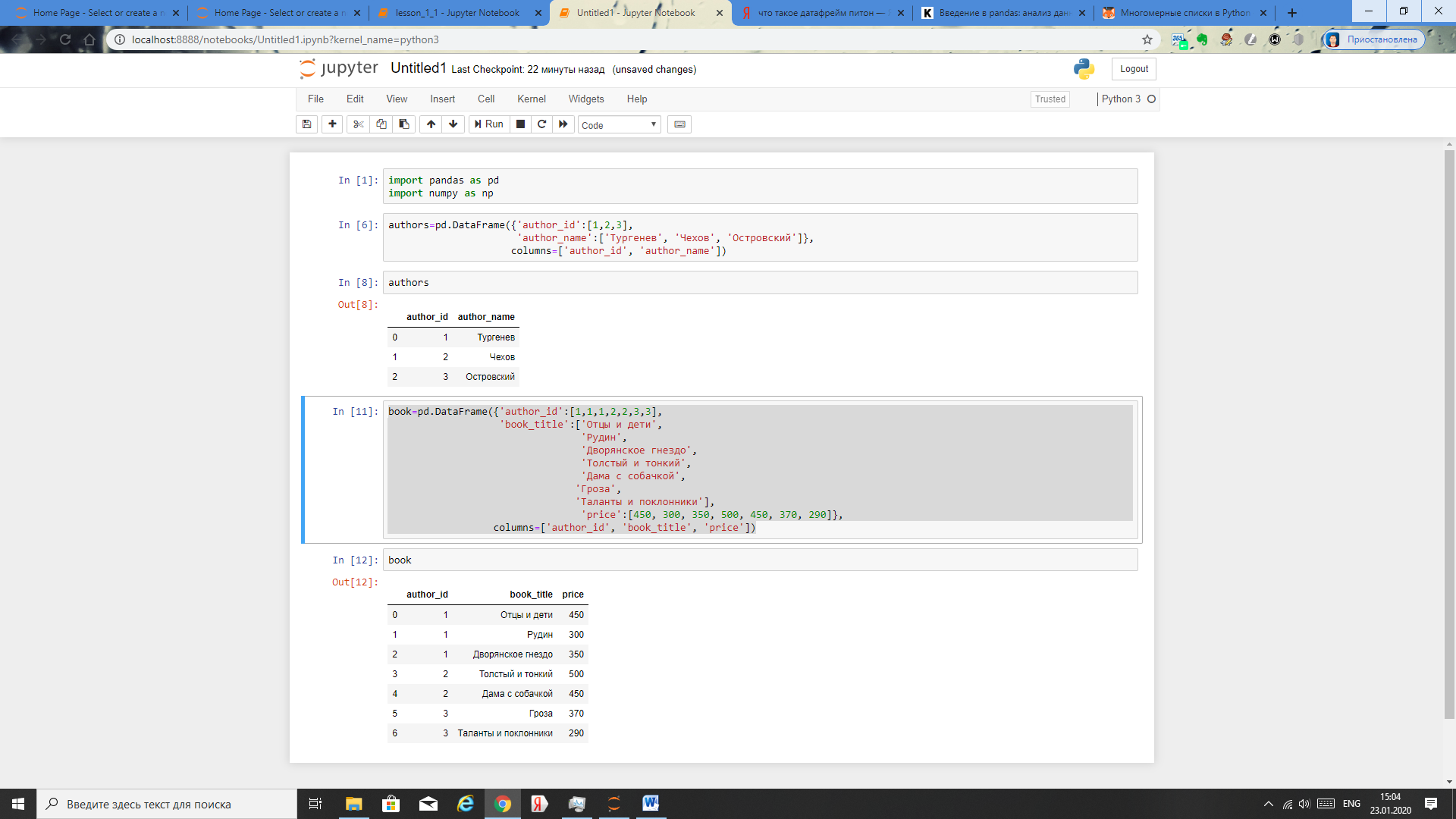
'Дама с собачкой',

'Гроза',

'Таланты и поклонники'],

'price':[450, 300, 350, 500, 450, 370, 290]},

columns=['author\_id', 'book\_title', 'price'])



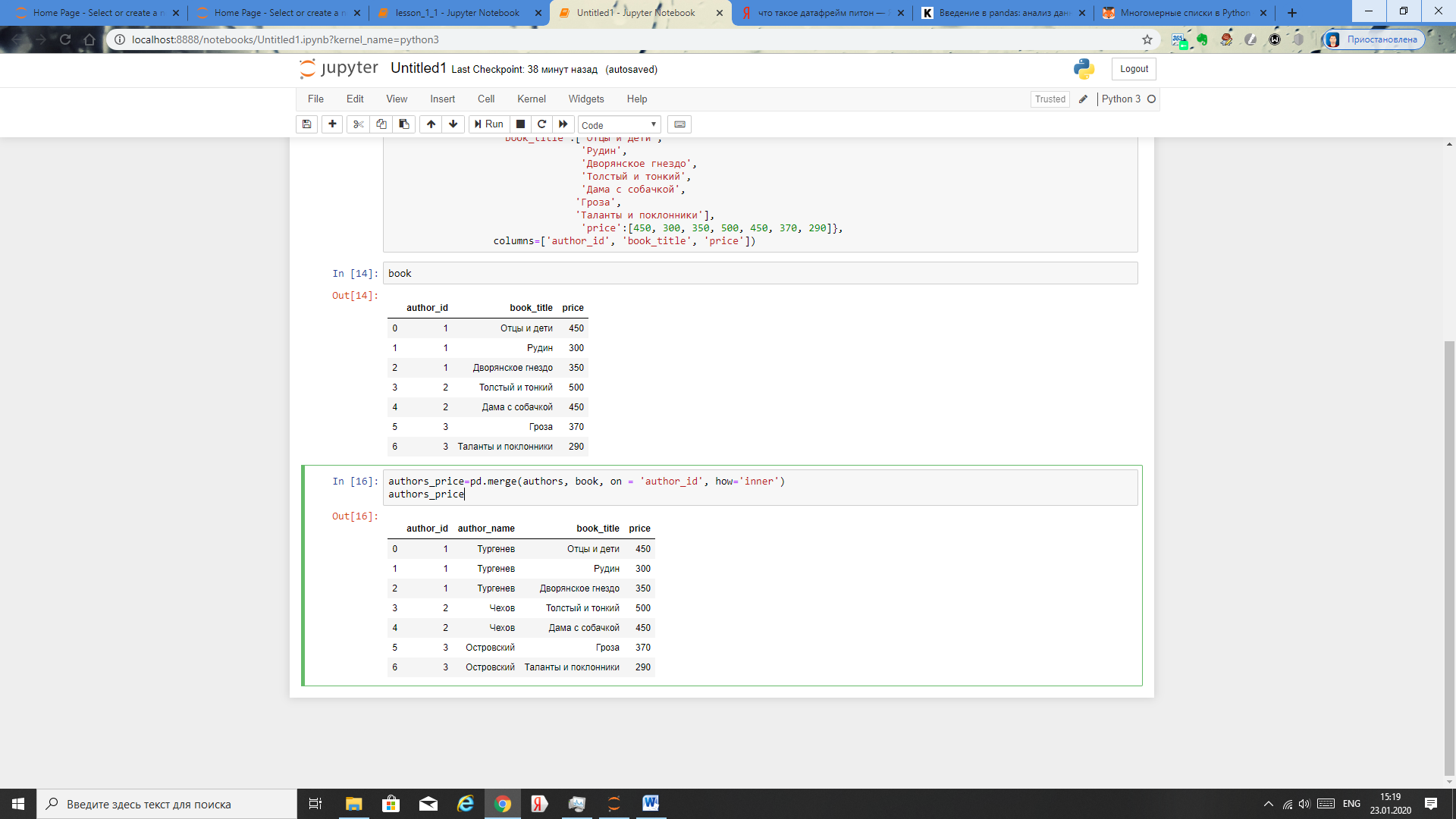
**Задание 2**

Получите датафрейм authors\_price, соединив датафреймы authors и books по полю author\_id.

Решение:

authors\_price=pd.merge(authors, book, on = 'author\_id', how='inner')

authors\_price

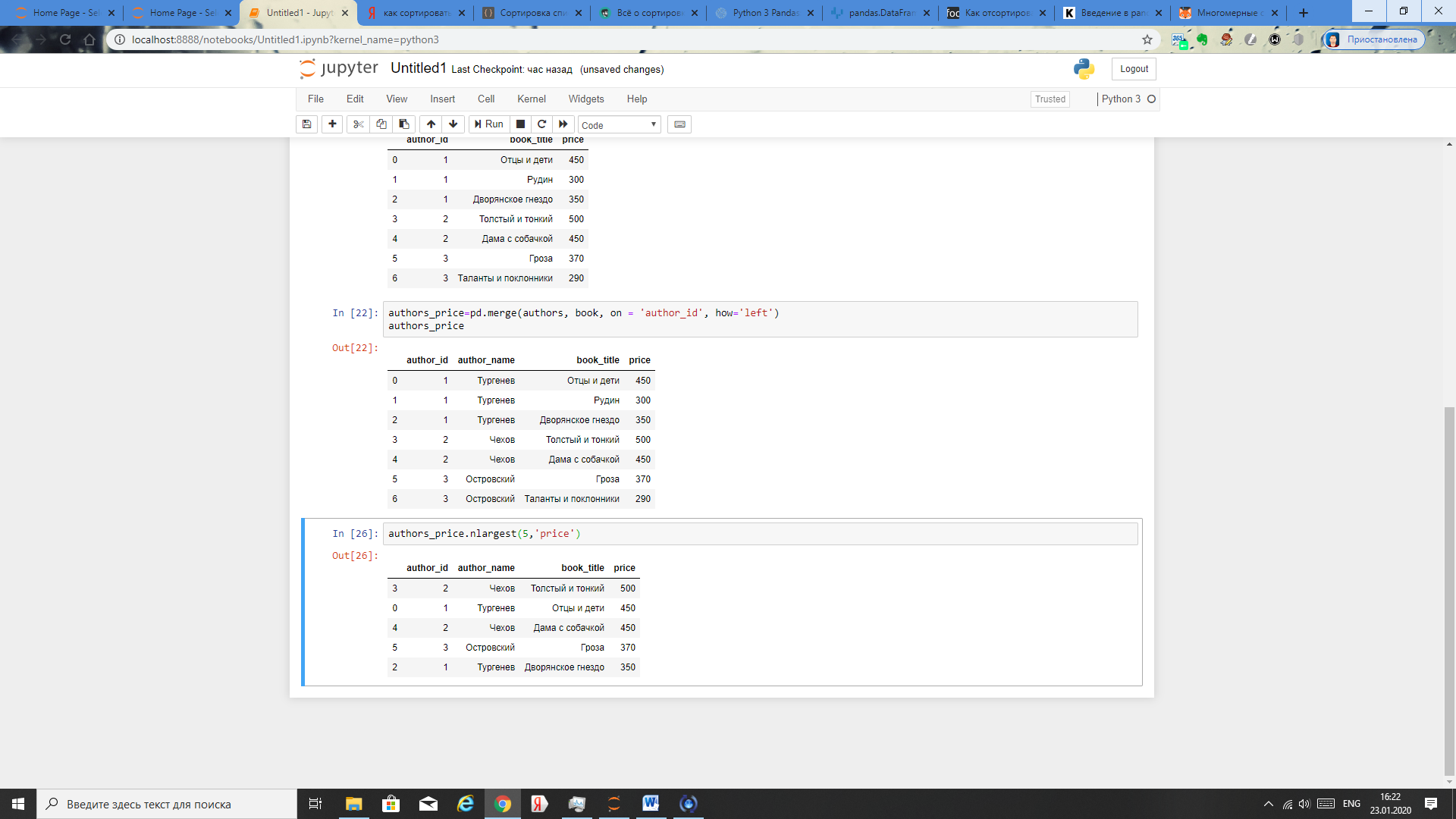


**Задание 3**

Создайте датафрейм top5, в котором содержатся строки из authors\_price с пятью самыми дорогими книгами.

Решение:

authors\_price.nlargest(5,'price')



**Задание 4**

Создайте датафрейм authors\_stat на основе информации из authors\_price. В датафрейме authors\_stat должны быть четыре столбца:

author\_name, min\_price, max\_price и mean\_price,

в которых должны содержаться соответственно имя автора,минимальная, максимальная и средняя цена на книги этого автора.

Решение:

min\_price=authors\_price.groupby('author\_name').agg({'price': 'min'})

min\_price

max\_price=authors\_price.groupby('author\_name').agg({'price': 'max'})

min\_price

mean\_price=authors\_price.groupby('author\_name').agg({'price': 'mean'})

mean\_price

authors\_stat1=pd.merge(min\_price, max\_price, on = 'author\_name', how='inner')

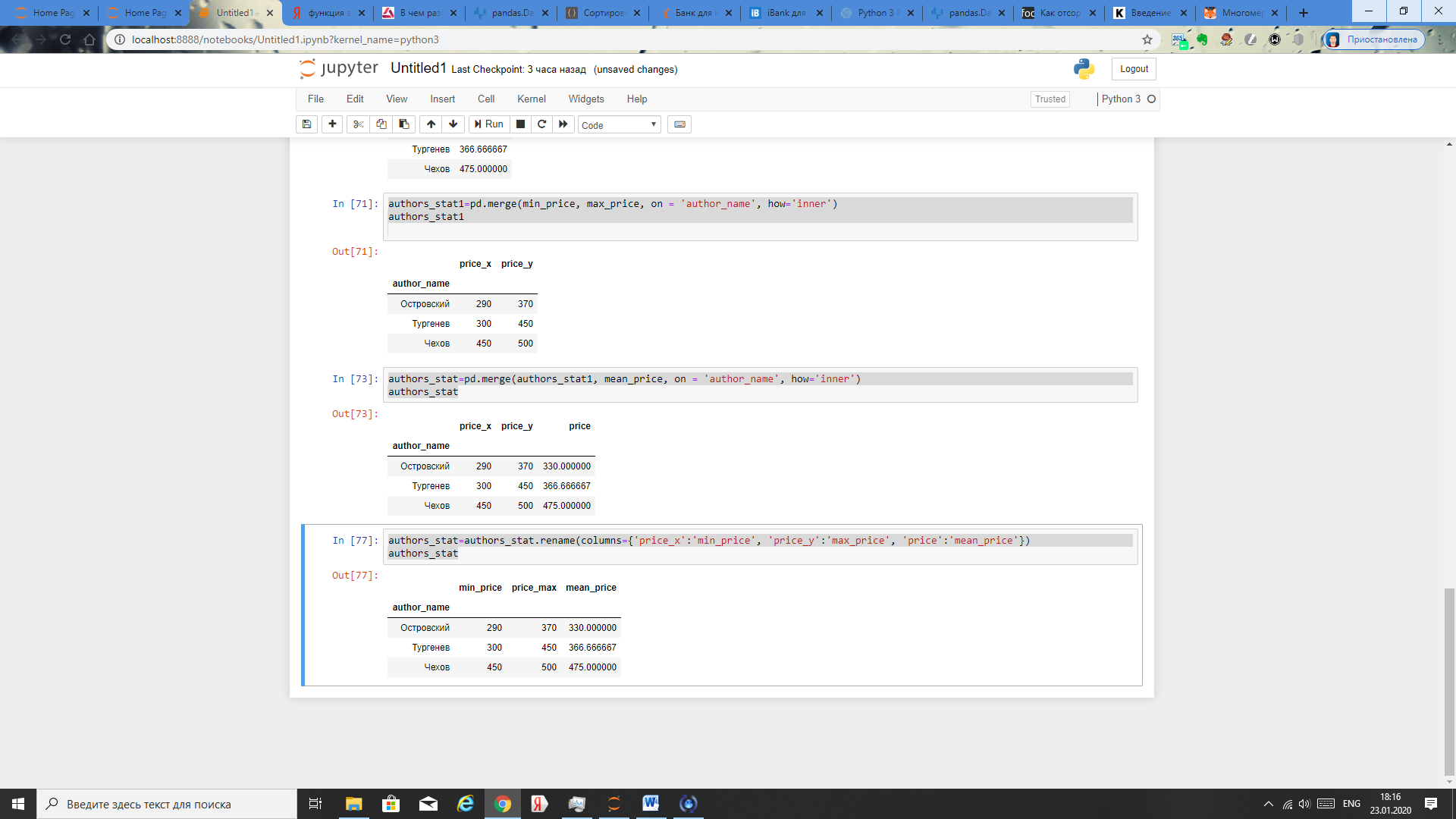
authors\_stat1

authors\_stat=pd.merge(authors\_stat1, mean\_price, on = 'author\_name', how='inner')

authors\_stat

authors\_stat=authors\_stat.rename(columns={'price\_x':'min\_price', 'price\_y':'max\_price', 'price':'mean\_price'})

authors\_stat



**Задание 5\*\***

Создайте новый столбец в датафрейме authors\_price под названием cover, в нем будут располагаться данные о том, какая обложка у данной книги - твердая или мягкая. В этот столбец поместите данные из следующего списка:

['твердая', 'мягкая', 'мягкая', 'твердая', 'твердая', 'мягкая', 'мягкая'].

Просмотрите документацию по функции pd.pivot\_table с помощью вопросительного знака.Для каждого автора посчитайте суммарную стоимость книг в твердой и мягкой обложке. Используйте для этого функцию pd.pivot\_table. При этом столбцы должны называться "твердая" и "мягкая", а индексами должны быть фамилии авторов. Пропущенные значения стоимостей заполните нулями, при необходимости загрузите библиотеку Numpy.

Назовите полученный датасет book\_info и сохраните его в формат pickle под названием "book\_info.pkl". Затем загрузите из этого файла датафрейм и назовите его book\_info2. Удостоверьтесь, что датафреймы book\_info и book\_info2 идентичны.

**Решение: не выполнено**