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 [2]: import numpy as np
```

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

```
In [4]: a
```

```
Out[4]: array([[ 1,  6],
               [ 2,  8],
               [ 3, 11],
               [ 3, 10],
               [ 1,  7]])
```

```
In [5]: a.shape
```

```
Out[5]: (5, 2)
```

```
In [6]: mean_a = a.mean(axis=0)
```

```
In [7]: mean_a
```

```
Out[7]: 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 [8]: a_centered = a - mean_a
```

```
In [9]: a_centered
```

```
Out[9]: 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 [10]: a_centered_sp = a_centered[:,0] @ a_centered[:,1]
```

```
In [13]: a_centered_sp
```

```
Out[13]: 8.0
```

```
In [18]: a_centered_sp / (a.shape[0] - 1)
```

```
Out[18]: 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. 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.

```
In [19]: covariance = np.cov(a.T)[0,1]
```

```
In [20]: covariance
```

```
Out[20]: 2.0
```

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']. Затем создайте датафрейм book со 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].

In [21]:
```
import pandas as pd
```

In [22]:
```
authors = pd.DataFrame({"author_id":[1,2,3], "author_name":['Тургенев','Чехов'
,'Островский']})
```

In [23]:
```
authors
```

Out[23]:

	author_id	author_name
0	1	Тургенев
1	2	Чехов
2	3	Островский

In [27]:
```
books = 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 покл
онники'],
                      "price":[450,300,350,500,450,370,290]})
```

In [28]:
```
books
```

Out[28]:

	author_id	book_title	price
0	1	Отцы и дети	450
1	1	Рудин	300
2	1	Дворянское гнездо	350
3	2	Толстый и тонкий	500
4	2	Дама с собачкой	450
5	3	Гроза	370
6	3	Таланты и поклонники	290

Задание 2

Получите датафрейм authors_price, соединив датафреймы authors и books по полю author_id.

```
In [29]: authors_price = pd.merge(authors, books, on='author_id', how='inner')
```

```
In [30]: authors_price
```

Out[30]:

	author_id	author_name	book_title	price
0	1	Тургенев	Отцы и дети	450
1	1	Тургенев	Рудин	300
2	1	Тургенев	Дворянское гнездо	350
3	2	Чехов	Толстый и тонкий	500
4	2	Чехов	Дама с собачкой	450
5	3	Островский	Гроза	370
6	3	Островский	Таланты и поклонники	290

Задание 3

Создайте датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорогими книгами.

```
In [31]: top5 = authors_price.nlargest(5, 'price')
```

```
In [32]: top5
```

Out[32]:

	author_id	author_name	book_title	price
3	2	Чехов	Толстый и тонкий	500
0	1	Тургенев	Отцы и дети	450
4	2	Чехов	Дама с собачкой	450
5	3	Островский	Гроза	370
2	1	Тургенев	Дворянское гнездо	350

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.

In [71]:
```
authors_stat = authors_price.groupby("author_name").agg({"price":("min","max",
"mean")})
```

In [72]:
```
authors_stat
```

Out[72]:

	price		
	min	max	mean
author_name			
Островский	290	370	330.000000
Тургенев	300	450	366.666667
Чехов	450	500	475.000000

In [73]:
```
authors_stat.columns = authors_stat.columns.get_level_values(1) + '_' + author
s_stat.columns.get_level_values(0)
authors_stat.reset_index(inplace=True)
authors_stat
```

Out[73]:

	author_name	min_price	max_price	mean_price
0	Островский	290	370	330.000000
1	Тургенев	300	450	366.666667
2	Чехов	450	500	475.000000

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.

In [74]:
```
authors_price["cover"] = ['твердая', 'мягкая', 'мягкая', 'твердая', 'твердая',
'мягкая', 'мягкая']
authors_price
```

Out[74]:

	author_id	author_name	book_title	price	cover
0	1	Тургенев	Отцы и дети	450	твердая
1	1	Тургенев	Рудин	300	мягкая
2	1	Тургенев	Дворянское гнездо	350	мягкая
3	2	Чехов	Толстый и тонкий	500	твердая
4	2	Чехов	Дама с собачкой	450	твердая
5	3	Островский	Гроза	370	мягкая
6	3	Островский	Таланты и поклонники	290	мягкая

In [76]:
```
book_info = pd.pivot_table(authors_price, values="price", index=["author_name"
],
                           columns="cover", aggfunc=np.sum, fill_value=0)
book_info
```

Out[76]:

cover	мягкая	твердая
author_name		
Островский	660	0
Тургенев	650	450
Чехов	0	950

In [77]:
```
book_info.to_pickle("book_info.pkl")
```

In [78]:
```
book_info2 = pd.read_pickle("book_info.pkl")
```

In [80]:
```
book_info
```

Out[80]:

cover	мягкая	твердая
author_name		
Островский	660	0
Тургенев	650	450
Чехов	0	950

In [81]:
```
book_info2
```

Out[81]:

cover	мягкая	твердая
author_name		
Островский	660	0
Тургенев	650	450
Чехов	0	950

In [ ]: