Задание 1.
Исходные данные:
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:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([ [1, 6], [2, 8], [3, 11], [3, 10], [1, 7] ])
>>> mean_a = a.mean()
>>> print(mean_a)
5.2
>>>
```

Задание 2.
Исходные данные:
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.
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([ [1, 6], [2, 8], [3, 11], [3, 10], [1, 7] ])
>>> mean_a = a.mean()
>>> b = a - mean_a
>>> print(b)
[[-4.2  0.8]
 [-3.2  2.8]
 [-2.2  5.8]
 [-2.2  4.8]
 [-4.2  1.8]]
>>>
```

Задание 3.
Исходные данные:
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:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> n = 5
>>> a = np.array([ [1, 6], [2, 8], [3, 11], [3, 10], [1, 7] ])
```

```
>>> mean_a = a.mean()
>>> b = a - mean_a
>>> c = np.hsplit(b, 2)
>>> d = b[0]
>>> f = b[1]
>>> k = (d * f).sum()
>>> l = k / n-1
>>> print(l)
2.136
>>>
```

Задание 4.
Исходные данные:
Число, которое мы получили в конце задания 3 является ковариацией двух признаков, содержащихся в массиве “b”. В задании 3 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:
Выборочная ковариация и выборочная дисперсия — Студопедия
В этом задании проверьте получившееся число, вычислив ковариацию ещё 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:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([ [1, 6], [2, 8], [3, 11], [3, 10], [1, 7] ])
>>> a_t = np.transpose(a)
>>> print(np.cov(a_t))
[[1.  2. ]
 [2.  4.3]]
>>>
```

Задание 5.
Исходные данные:
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].
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
```

```
>>> import pandas as pd
>>> a = [{"author_id": 1, "author_name": "Тургенев"}, {"author_id": 2, "author_name": "Чехов"},
{"author_id": 3, "author_name": "Островский"}]
>>> df = pd.DataFrame(a)
>>> print(df)
   author_id author_name
0          1    Тургенев
1          2       Чехов
2          3  Островский
>>> b = [{"author_id": 1, "book_title": "Отцы и дети", "price": 450}, {"author_id": 1,
"book_title": "Рудин", "price": 300}, {"author_id": 1, "book_title": "Дворянское гнездо", "price":
350}, {"author_id": 2, "book_title": "Толстый и тонкий", "price": 500}, {"author_id": 2,
"book_title": "Дама с собачкой", "price": 450}, {"author_id": 3, "book_title": "Гроза", "price":
370}, {"author_id": 3, "book_title": "Таланты и поклонники", "price": 290}]
>>> df1 = pd.DataFrame(b)
>>> print(df1)
   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 6.
Исходные данные:
Получите датафрейм authors_price, соединив датафреймы authors и books по полю author_id.
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> a = [{"author_id": 1, "author_name": "Тургенев"}, {"author_id": 2, "author_name": "Чехов"},
{"author_id": 3, "author_name": "Островский"}]
>>> df = pd.DataFrame(a)
>>> b = [{"author_id": 1, "book_title": "Отцы и дети", "price": 450}, {"author_id": 1,
"book_title": "Рудин", "price": 300}, {"author_id": 1, "book_title": "Дворянское гнездо", "price":
350}, {"author_id": 2, "book_title": "Толстый и тонкий", "price": 500}, {"author_id": 2,
"book_title": "Дама с собачкой", "price": 450}, {"author_id": 3, "book_title": "Гроза", "price":
370}, {"author_id": 3, "book_title": "Таланты и поклонники", "price": 290}]
>>> df1 = pd.DataFrame(b)
>>> res = df.merge(df1, on=["author_id"])
>>> print(res)
   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 7.
Исходные данные:
Создайте датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорогими книгами.
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> a = [{"author_id": 1, "author_name": "Тургенев"}, {"author_id": 2, "author_name": "Чехов"},
{"author_id": 3, "author_name": "Островский"}]
>>> df = pd.DataFrame(a)
>>> b = [{"author_id": 1, "book_title": "Отцы и дети", "price": 450}, {"author_id": 1,
"book_title": "Рудин", "price": 300}, {"author_id": 1, "book_title": "Дворянское гнездо", "price":
350}, {"author_id": 2, "book_title": "Толстый и тонкий", "price": 500}, {"author_id": 2,
"book_title": "Дама с собачкой", "price": 450}, {"author_id": 3, "book_title": "Гроза", "price":
370}, {"author_id": 3, "book_title": "Таланты и поклонники", "price": 290}]
>>> df1 = pd.DataFrame(b)
>>> res = df.merge(df1, on=["author_id"])
>>> top5 = res.nlargest(5, "price")
>>> print(top5)
   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 8.
Исходные данные:
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:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> a = [{"author_id": 1, "author_name": "Тургенев"}, {"author_id": 2, "author_name": "Чехов"},
{"author_id": 3, "author_name": "Островский"}]
>>> df = pd.DataFrame(a)
>>> b = [{"author_id": 1, "book_title": "Отцы и дети", "price": 450}, {"author_id": 1,
"book_title": "Рудин", "price": 300}, {"author_id": 1, "book_title": "Дворянское гнездо", "price":
```

```
350}, {"author_id": 2, "book_title": "Толстый и тонкий", "price": 500}, {"author_id": 2,
"book_title": "Дама с собачкой", "price": 450}, {"author_id": 3, "book_title": "Гроза", "price":
370}, {"author_id": 3, "book_title": "Таланты и поклонники", "price": 290}]
>>> df1 = pd.DataFrame(b)
>>> res = df.merge(df1, on=["author_id"])
>>> df2 = res.groupby("author_name").agg({"price": "min"}).rename(columns={"price":
"min_price"})
>>> df3 = res.groupby("author_name").agg({"price": "max"}).rename(columns={"price":
"max_price"})
>>> df4 = res.groupby("author_name").agg({"price":"mean"}).rename(columns={"price":
"mean_price"})
>>> df5 = pd.concat([df2, df3, df4], axis = 1)
>>> print(df5)
             min_price  max_price  mean_price
author_name
Островский         290        370  330.000000
Тургенев           300        450  366.666667
Чехов              450        500  475.000000
>>>
```

Задание 9.
Исходные данные:

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:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)

[GCC 9.4.0] on linux

Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import numpy as np

>>> import pandas as pd

>>> a = [{"author_id": 1, "author_name": "Тургенев"}, {"author_id": 2, "author_name": "Чехов"},
{"author_id": 3, "author_name": "Островский"}]

>>> df = pd.DataFrame(a)

>>> b = [{"author_id": 1, "book_title": "Отцы и дети", "price": 450}, {"author_id": 1,
"book_title": "Рудин", "price": 300}, {"author_id": 1, "book_title": "Дворянское гнездо", "price":
350}, {"author_id": 2, "book_title": "Толстый и тонкий", "price": 500}, {"author_id": 2,
"book_title": "Дама с собачкой", "price": 450}, {"author_id": 3, "book_title": "Гроза", "price":
370}, {"author_id": 3, "book_title": "Таланты и поклонники", "price": 290}]

>>> df1 = pd.DataFrame(b)

>>> res = df.merge(df1, on=["author_id"])

>>> c = [{"cover": "твердая"}, {"cover": "мягкая"}, {"cover": "мягкая"}, {"cover": "твердая"},
{"cover": "твердая"}, {"cover": "мягкая"}, {"cover": "мягкая"}]

>>> df2 = pd.DataFrame(c, columns=["cover"])

>>> df3 = pd.concat([res, df2], axis = 1)

>>> print(df3)

   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

>>> book_info = pd.pivot_table(res, values="price", index=["author_name"],

...                     aggfunc=np.sum, fill_value=0)

>>> print(book_info)

             price

author_name

Островский     660

Тургенев      1100

Чехов          950

>>> book_info.to_pickle('book_info.pkl')

>>> book_info2 = pd.read_pickle('book_info.pkl')

>>> book_info==book_info2

             price

author_name

Островский    True

Тургенев      True

Чехов         True

>>>
```