РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ им. П. Лумумба

Факультет физико-математических и естественных наук Кафедра теории вероятностей и кибербезопасности

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 3

Дисциплина:	Компьютерный	практикум	по моделированию
	1	1	1

Студент: Королёв Иван Андреевич

Группа: НКАбд-04-22

МОСКВА

2024 г.

Цель работы: Научиться работать с библиотеками numpy и pandas.

Выполнение работы

Тема «Вычисления с помощью Numpy»

Задание 1

Создайте массив Numpy под названием а 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 теап массива Numpy. Результат запишите в массив теап_а, в нем должно быть 2 элемента. Листинг программы на языке Python:

```
а = np.array([[1, 6], [2, 8], [3, 11], [3, 10], [1, 7]])
print("Maccub 5x2:\n", a)
mean_a = a.mean(axis=0)
print("Среднее по каждому признаку: {}".format(mean_a))

✓ 0.0s

Массив 5x2:
[[ 1 6]
[ 2 8]
[ 3 11]
[ 3 10]
[ 1 7]]
Среднее по каждому признаку: [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.

Листинг программы на языке Python:

```
a_centered = a - mean_a
print("Массив a_centered: \n{}".format(a_centered))

✓ 0.0s

Массив a_centered:
[[-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 _centered_spa. Затем поделите a_centered_sp на N-1, где N - число наблюдений.

Листинг программы на языке Python:

```
пр.seterr(divide="ignore", invalid="ignore")

# Столбцы (признаки)
a_centered_column_0 = a_centered[:, 0]
a_centered_column_1 = a_centered[:, 1]

# Скалярное произведение столбцов (признаков)
a_centered_spa = a_centered_column_0.dot(a_centered_column_1)

# Ковариация двух столбцов (признаков)
N = a.shape[0] # число наблюдений
result = a_centered_spa / (N - 1)

# Вывод
print(
    "Скалярное произведение столбцов: {}\пКовариация двух признаков: {}".format(
    | a_centered_spa, result
    )
)

✓ 0.0s

Скалярное произведение столбцов: 8.0
Ковариация двух признаков: 2.0
```

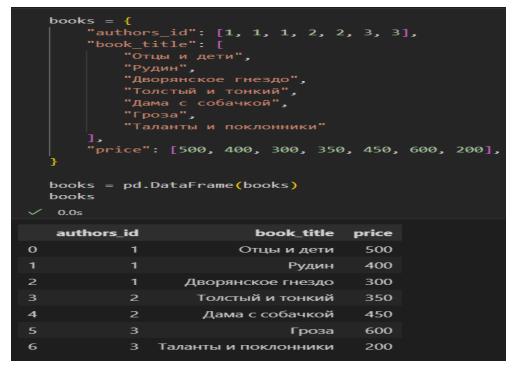
Задание 4. Число, которое мы получили в конце задания 3 является ковариацией двух признаков, содержащихся в массиве "а". В задании 4 мы делили сумму произведений центрированных признаков на N-1, а не на N, поэтому полученная нами величина является несмещенной оценкой ковариации. В этом задании проверьте получившееся число, вычислив ковариацию еще одним способом - с помощью функции пр.соv. В качестве аргумента т функция пр.соv должна принимать транспонированный массив "а". В получившейся ковариационной матрице (массив Numpy размером 2х2) искомое значение ковариации будет равно элементу в строке с индексом 0 и столбце с индексом 1.

Листинг программы на языке Python:

Тема «Работа с данными в Pandas»

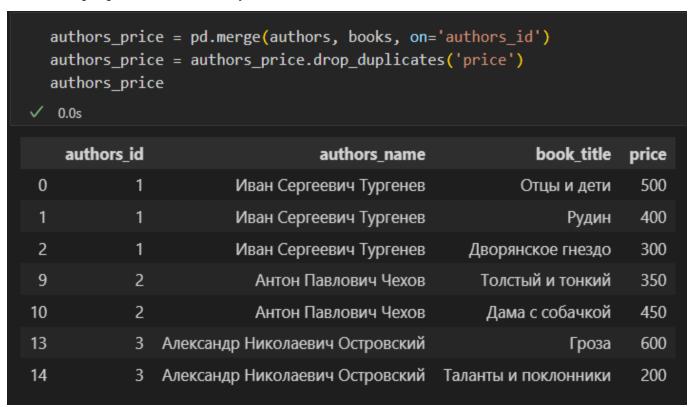
Задание 1 Импортируйте библиотеку Pandas и дайте ей псевдоним pd. Создайте датафрейм authors со столбцами author_id и author_name, в которых соответственно содержатся данные: [1, 2, 3] и ['Тургенев', 'Чехов', 'Островский']. Затем создайте датафрейм book со столбцами author_id, book_title и price Листинг программы на языке Python:

```
authors = {
        "authors_id": [1, 1, 1, 2, 2, 3, 3],
             "Иван Сергеевич Тургенев",
             "Иван Сергеевич Тургенев'
"Иван Сергеевич Тургенев'
"Антон Павлович Чехов",
"Антон Павлович Чехов",
             "Александр Николаевич Островский",
             "Александр Николаевич Островский'
   authors = pd.DataFrame(authors)
   authors
   0.0s
   authors_id
                                        authors_name
                            Иван Сергеевич Тургенев
o
              1
1
              1
                            Иван Сергеевич Тургенев
2
              1
                            Иван Сергеевич Тургенев
3
              2
                               Антон Павлович Чехов
4
              2
                               Антон Павлович Чехов
5
                 Александр Николаевич Островский
6
              3
                 Александр Николаевич Островский
```



Задание 2. Получите датафрейм authors_price, соединив дата фреймы authors и books по полю author_id.

Листинг программы на языке Python:



Задание 3. Создайте датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорогими книгами.

Листинг программы на языке Python:

t	:op5 = autho :op5 0.0s	ors_price.nlargest(5, 'price')		
	authors_id	authors_name	book_title	price
13	3	Александр Николаевич Островский	Гроза	600
0	1	Иван Сергеевич Тургенев	Отцы и дети	500
10	2	Антон Павлович Чехов	Дама с собачкой	450
1	1	Иван Сергеевич Тургенев	Рудин	400
9	2	Антон Павлович Чехов	Толстый и тонкий	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.

Листинг программы на языке Python:

```
authors_stat = (
      authors_price.groupby("authors_name")["price"]
      .agg(["min", "max", "mean"])
      .reset_index()
  authors_stat = pd.DataFrame(authors_stat)
  authors_stat
   0.0s
                     authors_name
                                    min
                                          max
                                                mean
   Александр Николаевич Островский
0
                                     200
                                           600
                                                400.0
              Антон Павлович Чехов
                                     350
                                          450
                                                400.0
1
2
            Иван Сергеевич Тургенев
                                     300
                                           500
                                                400.0
```

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']. 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 Листинг программы на языке Python:

	authors_pric	e["cover"] = ['твердая', 'мягкая	а', 'мягкая', 'твердая	', 'тв	ердая',	'мягкая', '	'мягкая']
	authors_pric	e					
✓	0.0s						
	authors_id	authors_name	book_title	price	cover		
0	1	Иван Сергеевич Тургенев	Отцы и дети	500	твердая		
1	1	Иван Сергеевич Тургенев	Рудин	400	мягкая		
2	1	Иван Сергеевич Тургенев	Дворянское гнездо	300	мягкая		
9	2	Антон Павлович Чехов	Толстый и тонкий	350	твердая		
10	2	Антон Павлович Чехов	Дама с собачкой	450	твердая		
13	3	Александр Николаевич Островский	Гроза	600	мягкая		
14	3	Александр Николаевич Островский	Таланты и поклонники	200	мягкая		

	price	
cover	мягкая	твердая
authors_name		
Александр Николаевич Островский	800	0
Антон Павлович Чехов	0	800
Иван Сергеевич Тургенев	700	500

```
# Чтение файла
book_info2 = pd.read_pickle("book_info.pkl")
book_info2
✓ 0.0s
```

	price	
cover	мягкая	твердая
authors_name		
Александр Николаевич Островский	800	0
Антон Павлович Чехов	0	800
Иван Сергеевич Тургенев	700	500

Идентичны

Заключение.

Я научился работать с библиотеками numpy и pandas.