Курс - Библиотеки Python для Data Science: Numpy, Matplotlib, Scikit-learn

Урок 2. Практическое задание 1 (2)

Видеоурок. Вычисления с помощью Numpy. Работа с данными в Pandas.

Тема “Работа с данными в Pandas”

Задание 1

Импортируйте библиотеку Pandas и дайте ей псевдоним pd.

В [10]:

```
import pandas as pd
```

Создайте датафрейм authors со столбцами author_id и author_name, в которых соответственно содержатся данные: [1, 2, 3] и ['Тургенев', 'Чехов', 'Островский'].

В [11]:

```
# Создайте датафрейм authors со столбцами author_id и author_name
auth = {
    "author_id": [1, 2, 3],
    "author_name": ['Тургенев', 'Чехов', 'Островский']
}

authors = pd.DataFrame(auth)
authors
```

Out[11]:

	author_id	author_name
0	1	Тургенев
1	2	Чехов
2	3	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].

Type *Markdown* and LaTeX: α^2

B [12]:

```
# Затем создайте датафрейм book со столбцами author_id, book_title и price
bk = {
    "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]
}
books = pd.DataFrame(bk)
books
```

Out[12]:

	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.

B [13]:

```
authors_price = pd.merge(authors, books, on='author_id', how='inner')
authors_price
```

Out[13]:

	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.

В [14]:

```
# authors_price.sort_values(by="price", inplace=True)
# top5 = authors_price.head(5)
# top5 = authors_price.tail(5)
# top5.reset_index(drop=True, inplace=True)

# Создаём датафрейм top5, в котором содержатся строки из authors_price с пятью самыми дорог
top5 = authors_price.nlargest(5, "price")
top5
```

Out[14]:

	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

В [15]:

```
type(top5)
```

Out[15]:

```
pandas.core.frame.DataFrame
```

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 .

В [26]:

```
groupby = authors_price.groupby("author_name")
groupby.agg({"price":["min", "max", "mean"]})
#groupby.agg({"price":'min', "price":'max', "price":'mean', suffixes=['min', 'max', 'mean']
```

Out[26]:

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

В [27]:

```
authors_price.groupby("author_name")['price'].agg(["min", "max", "mean"]).reset_index(). \
        rename(columns={'min':'min_price','max':'max_price','mean':'mean_price'})
```

Out[27]:

	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'].

In [17]:

```
# Создаём новый столбец в датафрейме authors_price под названием cover
vals = ['твердая', 'мягкая', 'мягкая', 'твердая', 'твердая', 'мягкая', 'мягкая']
authors_price['cover'] = vals
authors_price
```

Out[17]:

	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

Просмотрите документацию по функции pd.pivot_table с помощью вопросительного знака.

In [18]:

```
?pd.pivot_table
```

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.

In [24]:

```
# Для каждого автора посчитайте суммарную стоимость книг в твердой и мягкой обложке.

# 1 вариант с groupby
# groupby = authors_price.groupby(["author_name", "cover"])
# groupby.agg({"price":"sum"}).unstack()
groupby = authors_price.groupby(["author_name", "cover"])["price"].agg('sum').unstack()
groupby.fillna("0")
```

Out[24]:

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

В [25]:

```
# 2 вариант с pivot_table
book_info = authors_price.pivot_table("price", index="author_name", columns="cover").fillna
book_info
```

Out[25]:

cover	мягкая	твердая
author_name		
Островский	330	0
Тургенев	325	450
Чехов	0	475

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.

В [22]:

```
# Сохраняем book_info в формат pickle под названием "book_info.pkl"
book_info.to_pickle("book_info.pkl")

# Загружаем датафрейм из файла "book_info.pkl" в book_info2
book_info2 = pd.read_pickle("book_info.pkl")
book_info2
```

Out[22]:

cover	мягкая	твердая
author_name		
Островский	330	0
Тургенев	325	450
Чехов	0	475

Удостоверьтесь, что датафреймы book_info и book_info2 идентичны.

В [23]:

```
book_info == book_info2
```

Out[23]:

cover	мягкая	твердая
author_name		
Островский	True	True
Тургенев	True	True
Чехов	True	True

B [ ]:

```

```