Описание

На этот раз данные имеют следующую структуру:

* записываются для каждого пользователя, совершившего покупки, каждый день
* для каждой даты есть своя папка, внутри неё – папки для каждого пользователя
* внутри каждой папки есть файл data.csv, где и хранятся данные

Схематично выглядит так:

└── data

├── 2020-12-30

│  ├── FirstName\_LastName1

│ │   └── data.csv

│  ├── FirstName\_LastName2

│  │   └── data.csv

│  └── FirstName\_LastName3

│      └── data.csv

└── 2020-12-31

├── FirstName\_LastName1

│   └── data.csv

└── FirstName\_LastName5

└── data.csv

Например, 30 декабря три покупателя сделали покупки, 31 – два   
(папки 2020-12-30 и 2020-12-31 соответственно). Поскольку клиент FirstName\_LastName1 купил товары в оба дня, для него имеется папка в папке для каждой из дат. Для других клиентов – по одной.

*Note*: данные в задании покрывают другой временной период, имена тоже другие. Подробности, примеры и возможные подсказки можно найти в текстах следующих шагов.

Задачи

1. Соберите все данные из папки data в один датафрэйм, имеющий следующие столбцы: колонки из самих файлов (product\_id, quantity), а также имя пользователя (name), и дата этих покупок (date), соответствует названию папки, где лежит папка с пользователем)
2. Выясните, какой пользователь купил больше всего товаров. Если их несколько, то перечислите имена через запятую с пробелом и в алфавитном порядке.
3. Найдите топ-10 товаров по числу проданных единиц за всё время и постройте барплот. Сколько было продано единиц товара с product\_id==56?
4. Визуализируйте продажи по дням.
5. Сколько пользователей приобрели какой-либо товар повторно (более 1 раза)? Повтором будем считать покупку товара с одинаковым product\_id, совершенную в разные дни.

Найти данные можно либо на JupyterHub, либо скачать архив [отсюда](https://yadi.sk/d/fkzS9UYSr59EQQ).

💀 Сложное задание! 💀

Соберите все данные из папки data в один датафрэйм, имеющий следующие столбцы: колонки из самих файлов (product\_id, quantity), имя пользователя (name), и дату этих покупок (date), соответствует названию папки, где лежит папка с пользователем).

Пример итоговой таблицы:

   product\_id  quantity   name           date

0    56          2         Anatoly\_Karpov  2020-12-30

1    7           2         Anatoly\_Karpov  2020-12-30

2    9           3         Anatoly\_Karpov   2020-12-30

Для объединения датафреймов, читаемых из файлов, можно использовать метод pd.concat(). Например:

1. Имеются следующие данные:

> df\_1

   product\_id  quantity   name           date

0    56          2         Anatoly\_Karpov   2020-12-30

1    7           2         Anatoly\_Karpov  2020-12-30

2    9           3         Anatoly\_Karpov    2020-12-30

> df\_2

   product\_id  quantity   name           date

0    4           2         Кatya\_Skriptsova  2020-12-30

1    71          1         Кatya\_Skriptsova  2020-12-31

2. Соединяем, передав pd.concat на вход **список** датафреймов, которые нужно объединить. Обратите внимание: индексы повторяются, поэтому после соединения всех данных нужно их сбросить.

> df = pd.concat([df\_1, df\_2])

> df

   product\_id  quantity   name           date

0    56          2         Anatoly\_Karpov  2020-12-30

1    7           2         Anatoly\_Karpov  2020-12-30

2    9           3         Anatoly\_Karpov    2020-12-30

0    4           2         Кatya\_Skriptsova 2020-12-30

1    71          1         Кatya\_Skriptsova   2020-12-31

**В качестве ответа** укажите сумму по колонке quantity.

Также могут пригодиться:

* [Просмотр содержимого папок](https://lab.karpov.courses/learning/212/module/2218/lesson/20492/59043/277336/)
* [Циклы](https://lab.karpov.courses/learning/212/module/2218/lesson/20492/59043/277337/)
* [Сбросить индексы](https://lab.karpov.courses/learning/200/module/2111/lesson/19612/56595/266594/)
* [Присоединить датафрейм](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.concat.html)

[Подсказки](https://www.notion.so/4-1-python-1995e58783344a23b5c9ea5cec01668e)

Выясните, какой пользователь купил больше всего товаров. Если их несколько, то перечислите имена через запятую с пробелом и в алфавитном порядке. Например:

Anatoly\_Karpov, Nekto\_Ktotovich, Lena\_Uhanova, Ignat, Sasha\_Tokarev

[Подсказки](https://www.notion.so/4-2-python-b3c23913583748a4b2835d78759076a7)

Найдите топ-10 товаров по числу проданных единиц за всё время и постройте барплот (столбчатую диаграмму, sns.barplot), где:

* по оси x – идентификаторы товаров (product\_id)
* по оси y – суммарное число их покупок (сколько товаров с таким product\_id было продано)

Для практики попробуйте изменить параметры графика: цвет (color), прозрачность (alpha), сортировку значений по оси х (order).

В качестве ответа укажите, сколько было продано единиц товара с product\_id **равным 56**.

[Подсказки](https://www.notion.so/4-3-python-6eccd80104584ef2adfd4b055195568e)

Посмотрим на продажи по дням! Для визуализации снова используйте барплот, только теперь по оси x будут дни, по которым у нас есть данные (date). Далее выберите верные утверждения:

[Подсказки](https://www.notion.so/4-4-python-e1a349a289c1411d81cfaa181e7d3c69)

4 декабря было куплено почти в два раза больше товаров, чем в предыдущий день

Меньше всего продуктов было приобретено 4 декабря

Максимальное число товаров было продано 4 декабря

Меньше всего товаров было приобретено 6 декабря

8 и 9 декабря было продано примерно одинаковое число товаров

5 декабря было куплено почти в два раза больше товаров, чем в предыдущий день

7 декабря было куплено почти в два раза больше товаров, чем в предыдущий день

5 декабря было куплено почти в три раза меньше товаров, чем в предыдущий день

💀 Сложное задание! 💀

Сколько пользователей приобрели какой-либо товар повторно (более 1 раза)? Повтором будем считать покупку товара с одинаковым product\_id, совершенную в **разные** дни.

Например, пользователь Sasha Tsarev дважды приобрел товар 6. Поскольку покупка была совершена в один и тот же день, повторной в данном случае она считаться не будет. Katya Skriptsova также сделал две покупки товара 7, но в разные дни, поэтому этот случай засчитывается.

product\_id quantity name date

25 2 Katya Skriptsova 2020-12-05

54 1 Olya Silyutina 2020-12-05

7 4 Katya Skriptsova 2020-12-05

6 4 Sasha Tsarev 2020-12-06

6 1 Sasha Tsarev 2020-12-06

7 5 Katya Skriptsova 2020-12-06

Может пригодиться: [.drop\_duplicates()](https://lab.karpov.courses/learning/212/module/2218/lesson/20492/59043/277331/)

[Подсказки](https://www.notion.so/4-5-python-60702b54830449b2b8d85979bb169df3)

## Дополнительный проект

И небольшой минипроект на фильтрацию данных!

Имеется таблица с записями о компаниях, включая их ИНН, и отдельный текстовый файл с набором ИНН (колонка head\_inn), которые хранятся в папке 4\_inn.

* [inn.xls](https://yadi.sk/i/o2dds5AuREg9Uw) — таблица в экселевском формате
* [necessary\_inn.txt](https://yadi.sk/d/UPU6456vZSGIqQ) — перечень необходимых ИНН в текстовом формате

### **Задачи:**

* извлечь из таблицы записи с ИНН, указанными в текстовом файле
* записать результат в файл selected\_inn.csv
* В качестве проверки укажите сумму колонки income,RUB в отобранных данных.
* **NB!**У набора данных necessary\_inn.txt нет заголовка, поэтому вам пригодится аргумент header=None.

[Подсказки](https://www.notion.so/4-1-python-3ef0ebae5db94eaaa907024d3736ed18)