README
Applepen.

Весенний семестр 2020 г.

1 Постановка задачи.

Applepen - большая торговая сеть, которая занимается продажей всего двух продуктов: яблок и карандашей. Её магазины расположены в различных уголках Соединённых Штатов и более 10 лет обслуживает покупателей.
В CSV-файлах дана информация, отсортированная по дате:

1. о закупках (поставки яблок и карандашей два раза в месяц),
2. о продажах (лог транзакций, по записи на каждую проданную позицию),
3. об инвентаре (месячные данные общего количества яблок и карандашей на складе).

Необходимо получить следующие данные в CSV-файлах:

1. состояние склада на каждый день,
2. месячные данные о количестве своровaнного товара,
3. агрегированные данные об объёмах продаж и количестве своровaнной продукции по штату и году.

Наборы входных и выходных данных для тестирования решения скачаны по ссылке: https://console.cloud.google.com/storage/browser/artem-pyanykh-cmc-prac-task3-seed17. В скачанной папке с исходными данными содержится 3 типа файлов для каждого магазина в штате:

1. **(имя штата)-(название магазина)-supply.csv** - информация по поставкам (1го и 15го числа месяца),
2. **(имя штата)-(название магазина)-sell.csv** - транзакции на каждый проданный продукт,
3. **(имя штата)-(название магазина)-inventory.csv** - месячные данные общего количества яблок и карандашей на складе.

2 Математическое решение задачи.

Топ-менеджеры требуют предоставить информацию о состоянии склада на каждый день. Мы знаем, что поступления на склад происходят только два раза в месяц – 1 и 15 числа, так же нам известно о всех транзакциях за день и о количестве товара на складе в конце каждого месяца. Рассмотрим модель относительно одного товара, так как яблоки и карандаши независимы, вычисления будут одинаковыми.
Представим, что у нас есть данные, что сегодня на складе находится **СКЛАД** единиц товара, просуммировав все транзакции за день, мы точно знаем, что сегодня всего было распродано **ПРОДАЛИ** единиц товара. Тогда завтра на складе будет:

СКЛАД_ЗАВТРА = СКЛАД – ПРОДАЛИ

Так мы вычисляем данные до дня поставок. В день, когда завозят новый товар, на складе становится на ПОСТАВКА единиц товара больше, поэтому состояние склада будет определяться, как

СКЛАД_ЗАВТРА = СКЛАД + ПОСТАВКА – ПРОДАЛИ

При этом мы знаем, что сотрудники воруют: количество реального товара и количество товара “на бумаге” отличается, поэтому в конце каждого месяца в файл **daily**, в котором мы храним информацию о состоянии склада на каждый день, записываем данные из **inventory** – действительное количество товара. На этом шаге, мы параллельно заполняем и файл **steal**, в котором хранится месячная информация о ворованном товаре, записывая туда разницу между значениями

УКРАЛИ = СКЛАД_РЕЗУЛЬТАТ - СКЛАД_РЕАЛЬНОСТЬ

Первое из них – это то, что мы получили по информации о поставках и продажах, а второе - после инвентаризации **СКЛАД_РЕАЛЬНОСТЬ**.
А что касательно последнего пункта? Агрегация, или агрегирование (лат. aggregatio "присоединение") — процесс объединения элементов в одну систему. То есть теперь, нам надо объединить полученные нами данные по году и штату, в котором расположены торговые точки.

3 Программный подход к решению задачи.

Своё решение мы представим в виде программы на языке Python. Нам понадобятся:

- модуль **pandas** – библиотека для обработки и анализа данных, работы с объектами DataFrame, метод pandas.testing.assert_frame_equal – для сравнения двух DateFrame;
- класс **datetime.timedelta** – для вычисления разница между двумя моментами времени, с точностью до микросекунд;

- класс **datetime.datetime** - содержит информацию о времени и дате, основываясь на данных из Григорианского календаря;

- модуль **os** - библиотека функций для работы с операционной системой. Методы, включенные в неё позволяют определять тип операционной системы, получать доступ к переменным окружения, управлять директориями и файлами. Например, os.listdir - список файлов и директорий в папке, os.mkdir - создаёт директорию, os.path.isfile - является ли путь файлом, os.path.join - соединяет пути с учётом особенностей операционной системы;

- модуль **shutil** содержит набор функций высокого уровня для обработки файлов, групп файлов, и папок. В частности, shutil.rmtree - удаляет текущую директорию и все поддиректории.

Отметим, что в реализуемой нами программе важно, чтобы входные данные хранились в папке **input**, выходные – в **answer**, при этом они находились в одной директории вместе с **Task_3.ipynb**.
Результатом выполнения программы будет папка **our_answer**, созданная в той же директории, в которой находится сама программа и входные данные. В эту папку записываются результаты в следующем формате:

1. **(имя штата)-(название магазина)-daily.csv** - состояние склада на каждый день,

2. **(имя штата)-(название магазина)-steal.csv** - месячные данные о количестве сворованного товара,

3. **aggregation** - агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году.

По ходу программы мы рассматриваем магазины, которые есть в папке **input**, каждый раз вызывая функцию **shop_processing(inventory, sell, supply, state)**, параметрами которой являются считанные файлы и название штата, а результатом - **daily, steal, aggregation**.
Первые два файла строятся по описанному выше алгоритму, для последнего в процессе работы мы считаем **apple_sold_year, pen_sold_year, apple_stolen, pen_stolen** простым суммированием. Чтобы получить исходные агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году, мы сначала конкатенируем все полученные таблицы aggregation с помощью метода contact, затем группируем их по штату и году с помощью метода groupby и для завершения суммируем значения по полям.
В итоге получается такая формула:

aggregation_sum = pd.concat([aggregation_sum,aggregation]).
groupby(['year','state'])
['apple_sold', 'apple_stolen', 'pen_sold', 'pen_stolen'].sum()

Для завершения работы, чтобы проверить корректность полученных нами

данных, мы сверяем файлы из двух папок answer и our_answer, если встречаем различия, то выбрасываем исключение и выводим соответствующую информацию.