Задание prac-io-3

Краткое описание задания

На основании данных о закупках, продажах и инвентаре, доступных в формате CSV получить следующие сведения:

1. Состояние склада на каждый день.
2. Месячные данные о количестве сворованного товара.
3. Агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году.
   Данные предоставить в формате CSV.

Файлы проекта:

- **zadanie3.ipynb** — программма, реализующая задание;
- **README. md** — описание проекта;
- Исходные данные, на которых выполняется программа, доступны по ссылке.

Подход к решению

Каждый файл соответствует шаблону **MS-state-what.csv** , где

- **state** — штат, в котором расположен магазин;
- **what** — содержимое файла(3 варианта):
    1. *inventory* — данные о товаре (apple, pen) на складе за

каждый месяц, запись осуществляется в последний день месяца.

Запись состоит из:

- date - дата проверки состояния склада;
- apple - количество яблок на складе;
- pen - количество ручек на складе.

2. *sell* — лог транзакций – данные о продаже товара, по записи на каждую проданную позицию.

   Запись состоит из:

   - date – дата продажи;
   - sku_num – информация о продаже, имеет вид: **MS-state-product-id**, где
     - **state** – штат
     - **product** - ap/pe – какой товар был продан(яблоко/ручка)
     - **id** – идентификационный номер продажи.

3. *supply* — данные о закупках на склад, производящихся 2 раза в месяц: 1 и 15 числа каждого месяца.

   Запись состоит из:

   - date - дата поставки;
   - apple - количество поставленных яблок;
   - pen - количество поставленных ручек.

Исходя из шаблонности названия файлов, реализована функция **work(statename)**, в которую передаётся только название штата. Для каждого штата производится аналогичная обработка данных.

Реализация решения

Считаем, что непосредственно перед первой поставкой товаров склады были пустые.

1. Состояние склада на каждый день

Создаём объект `df` типа `DataFrame`, который будет нести информацию о складе на каждый день. Для того, чтобы заполнить данные об одном дне:

- прибавляем количество завезённых на склад товаров 1 и 15 числа каждого месяца;
- вычитаем количество проданных за каждый день товаров;
- в последний день месяца копируем значение из inventory.

Заполнив все даты, получаем файл **MS-state-daily.csv**, соответствующий штату **state**.

2. Месячные данные о количестве своровaнного товара

Создаём объект `df_steal` типа `DataFrame`, который будет нести информацию о своровaнном товаре. Данные должны быть предоставлены за последний день каждого месяца. Необходимая информация вычисляется следующим образом:

‘количесвто своровaнного товара’ = 'количество товара согласно `df` *’ -*

*'количество товара согласно **inventory**'.*

Результат выводится в файл **MS-state-steal.csv**, соответствующий штату **state**.

3. Агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году

Создаём объект `d3` типа `DataFrame`, который будет нести информацию об объемах продаж и количестве сворованной продукции по штату и году. Изначально все данные обнулены.

- пока идёт текущий год
    - увеличиваем на 1 число продаж ручек/яблок за каждую запись в логе транзакций в текущем году;
    - каждый раз когда вычисляем количество сворованного товара в конце месяца, прибавляем это же число и к количеству сворованного товара в `d3`.

Заполнив всю необходимую информацию о каждом штате, выводим её в файл **agregate.csv**.

Используемая версия Python

- Python 3.6.1

Используемые библиотеки

- **pandas**
    - pandas.read_csv(‘testing.csv’, sep=’, ', parse_dates=[‘Date’], index_col=‘Date’) – считывает данные из csv-файла.
        - ‘testing.csv’ – название файла;
        - sep=’, ’ – разделитель полей;
        - parse_dates=[‘Date’] – указывает название колонки, в которой находятся даты;
        - index_col=‘Date’ – изменят индекс на значения в колонке ‘Date’;
    - pandas.DataFrame(index, columns) – создаёт объект класса DataFrame
    - pandas.DataFrame.iloc[i] – выбор “строки” i в DataFrame
    - pandas.date_range(start, end, freq=‘D’) – создаёт список, состоящий из дат, начиная с start, по end; freq - шаг (D = day)
    - pandas.DataFrame.to_csv(path) – выводит данные из DataFrame в csv-файл с именем path

Необходимое ПО

- *Jupyter Notebook*
- *Python*
- *gcloud (реком.)*

Запуск программы

В директории проекта должна находиться папка `out/`, содержащая необходимые для работы программы данные *(см. ‘Файлы проекта’)*.
В *Jupyter Notebook:*

- открываем файл **zadanie3.ipynb**;
- `Cell -> Run All`.

Задание выполнили:

Павлов Антон

Сметанин Даниил

Кононов Сергей

Осипа Андрей