README

Задание 3

Составила: студентка гр. 312 Бурамбекова А.Н.

Содержание

1 Постановка задачи

Описание

Applepen – это большая торговая сеть, которая занимается продажей всего двух продуктов: яблок и карандашей. Ее магазины расположены в различных уголках Соединенных Штатов и более 10 лет обслуживают покупателей.

Недавно топ-менеджмент компании решил более активно использовать имеющиеся у них данные в принятии решений. Каждый магазин собирает информацию о:

1. закупках (поставки яблок и карандашей два раза в месяц),

2. продажах (лог транзакций, по записи на каждую проданную позицию),

3. запасы на складе (месячные данные общего количества яблок и карандашей на складе).

Данные доступны в формате CSV. Внутри файла данные отсортированы по дате.

К сожалению, данные никогда не консолидировались и не проверялись. Нам необходимо получить следующие данные в CSV-файлах.

1. Состояние склада на каждый день

Данные о состоянии склада в конце каждого дня после того как все поставки и продажи были совершены. Подобная информация будет очень ценна менеджерам магазинов. Состояние склада должно строиться на основе месячных данных об инвентаре. Известно, что люди воруют из магазинов, но сейчас нет никакой возможности узнать объем сворованного товара в каждый день.

также требуется определить три лучших месяца с точки зрения самых эффективных продаж (по каждому из товаров). Должны быть указаны месяцы и количество проданного товара за каждый из них.

Данные должны быть в следующем виде:

date	apple	pen
2006-01-01	14105	770
...		

2. Месячные данные о количестве сворованного товара

Данные должны быть в следующем виде:

date	apple	pen
2006-01-31	4	4
2006-02-28	5	0

Также требуется определить три худших года с точки зрения самых больших краж (по каждому из товаров). Должны быть указаны года и количество украденного товара за каждый из них.

2 Теория

Названия входных файлов можно условно поделить на 2 части: префикс, который имеет следующий вид: «Штат>-<Название магазина>-"и суффикс - "supply.csv "sell.csv "inventory.csv". Мы отдельно работаем с каждым суффиксом, с помощью которого получаем доступ к 3 файлам: закупки, продажи и инвентарь.

Сначала подсчитаем число яблок и ручек, проданных за каждый день. Для этого в файле sell.csv просто подсчитываем число строк, содержащих "ре"/"ар"для каждого дня.

Состояние склада на каждый день

Мы знаем, что поступления на склад происходят только 2 раза в месяц. Для дальнейшей работы с наборами данных мы расширим inventory.csv, заполнив недостающие значения нулями. Отсюда вычтем то, что было продано - 1 и 15 числа будут положительные значения в таблице, в остальные дни - отрицательные (ничего на склад не поступало, были только продажи). Далее просуммируем за каждый месяц - получим состояние склада на каждый день.

Месячные данные о количестве сворованного товара

Чтобы получить значения сворованного товара за месяц, достаточно вычесть из вычисленного реального состояния склада в конце месяца количество товара, указанное в инвентаре.

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

Добавляем индексы штата и года в список statistics. Суммируем по ним все продажи и количество сворованного товара и конкатенируем в единый DataFrame.

3 Этапы решения

1) DataFrame.set_index(keys, inplace=False) - устанавливает индексирование в наборе данных; keys - столбец, отвечающий за индексирование; inplace - False - создавать/ True - не создавать новый объект в результате применения операции - работать со старым

2) DataFrame.groupby(by=None) - группирует данные по заданному аргументу; by - столбец, относительно которого будет осуществляться перегруппировка

3) DataFrame.agg(func) - агрегирует данные по заданному аргументу, т.е. имеющуюся в наборе данных информацию распределяет по новым стобцам по некоторому правилу; func - функция, используемая для агрегирования данных

4) DataFrame.reindex_like(other) - возвращает набор данных с индексированием, подобным объекту-аргументу; other - объект для копирования его индексирования

5) DataFrame.fillna(value=None, method=None, axis=None) - заполняет ячейки N/A некоторыми значениями; value - значение, которым заполняются пустые ячейки; method - метод заполнения - ('ffill' - ячейки заполняются значениями из вышестоящих ячеек);

6) DataFrame.shift(periods=1) - сдвигает набор данных на указанное в periods значение; peridos - на сколько сдвинуть

7) DataFrame.join(other, lsuffix='', rsuffix='') - присоединяет к набору данных столбцы, добавляет соответствующие суффиксы к названиям столбцов; other - другой набор данных, при помощи которого осуществляется присоединение lsuffix - суффикс для левых столбцов; rsuffix - суффикс для правых столбцов;

8) Series.map(arg) - при помощи arg редактирует соответствующий столбец; arg - аргумент

9) DataFrame.to_csv(path=None) - создает на основе входного набора данных csv файл с именем path; path - имя файла

10) os.path.join(path, *paths) - конкатенация path и элементов paths - '/<paths[i]>' for all i; path - список элементов

11) os.path.isfile(path) - возвращает True, если файл по указанному пути дей-

ствительно существует; path - путь

12) os.listdir(path) - возвращает список файлов, находящихся в указанной директории; path - путь

13) os.path.exists(path) - возвращает True, если указанный путь действительно существует; path - путь

14) os.makedirs(path) - создает директорию с заданным именем. Если такая директория уже существует, выбрасывается исключение; path - имя создаваемой директории

15) shutil.rmtree(path) - рекурсивно удаляет содержимое указанной директории. Если её нет - выбрасывается исключение path - путь к подлежащей удалению директории

Запуск

1.Поместить в папку 'input' файлы для обработки

2.Cell -> Run All

3.Результат будет находиться в папке 'output'