## Problem Statement

An online retailer has asked us to analyse their transactional data over the past decade.

They have asked us to build out the initial data pipelines to support this. They have sent some sample data with the expectation that your code will be run against the larger dataset.

## Requirement

1. A pipeline that transforms the data into daily partitions in the Parquet format
2. A pipeline (or pipelines) that outputs the analysis suggested in the email
3. A pipeline that computes the volume-based discount suggested in the email

## Expectations

1. Find the total value of cancelled orders along with yearly segmentation.
2. Find the total value of On-Hold orders along with yearly segmentation.
3. List unique products per product line.
4. Find the sales trend of number of shipped classic cars.
5. Volume based discounts on vintage / classic cars, motorbikes and trucks and buses based on discounted scheme.

## Analysis

* Sales data is in JSON format segmented by yearly files.
* Json files has a schema structure

SalesData\_yyyy.json

----- >ORDERNUMBER

----- >PRODUCTCODE

----- >attributes

----- >QUANTITYORDERED

----- >PRICEEACH

----- >SALES

----- >ORDERDATE

----- >STATUS

----- >PRODUCTLINE

----- >MSRP

* “Attributes” in json is a list of dictionary items containing list of features of order and product.
* “Attributes” can have multiple features for an order and product.
* Order Date is in mm/dd/yyyy format
* Order date, Quantity ordered, Sales, Status and MSRP are key features for required output.

## Approach

1. Install required packages mentioned in requirements.txt
2. Read yearly json files and combine all data into a single data frame using pandas.
3. Convert “attributes” column into rows as it can have multiple features in list.
4. Extract the features from dictionary which are in “attributes” column.
5. Add Day, Month and Year attributes in data frame by extracting it from Order date.
6. Create daily files of parquet format for sales data in ‘output’ directory using gzip compression.
7. Create a text file “SaleValue.txt” in a ‘output’ directory and calculate and write following
8. Total Sale Value
9. Cancelled Ordered sales value
10. on-hold sales value
11. Create a excel file “SalesDataSummary.xlsx” in a ‘output’ directory and calculate and write following in different worksheet
12. Total Sales value segmented by year in worksheet “YearlySaleValue”
13. Total Sales value segmented by year and status in worksheet “YearlyStatusSaleValue”
14. Total Cancelled Sales value segmented by year in worksheet “YearlyCancelledOrders”
15. Total on-hold Sales value segmented by year in worksheet “YearlyOnHoldOrders”
16. List of Unique Products segmented by Product Line in worksheet “ProductPerProductLine”
17. Discount table in worksheet “Discount Table”
18. Volume based discounts on vintage / classic cars, motorbikes and trucks and buses based on discount table in worksheet “DiscountedRates”
19. Create a log file to log all events in a log’s directory for each execution.
20. Create a test script to do basic checks like
21. Read json function reading json files and returning shape as expected.
22. Attribute column data converting as expected and returning required output.
23. Function setup data is adding day, month, year columns and giving required output.

## Project structure

* pwcSales

----- >scripts

----- >\_\_init\_\_.py

----- >pwcSalesSolution.py

----- >logs

----- >pwcSalesExecutionyyyymmdd-hhmiss.log

----- >data

----- >SalesData\_2003.json

----- >SalesData\_2004.json

----- >SalesData\_2005.json

----- >output

----- >SaleValue.txt

----- >SalesDataSummary.xlsx

----- >DailySaleData\_dd-mm-yyyy.gzip

----- >tests

----- >\_\_init\_\_.py

----- >test\_pwcSalesSolution.py

----- >testSalesData.json

----- >testSalesDataLog.log

----- >requirements.txt