Cointab Software Private Limited

Data Analyst - Assignment Overview Please understand the below mentioned real-life scenario and try to solve the assignment. The sample data is attached in the link provided below for your reference. Business Scenario You are a data analyst and your client has a large ecommerce company in India (let's call it X). X gets a thousand orders via their website on a daily basis and they have to deliver them as fast as they can. For delivering the goods ordered by the customers, X has tied up with multiple courier companies in India as delivery partners who charge them some amount per delivery. The charges are dependent upon two factors: ● Weight of the product ● Distance between the warehouse (pickup location) and customer's delivery address (destination location) On an average, the delivery charges are Rs. 100 per shipment. So if X ships 1,00,000 orders per month, they have to pay approximately Rs. 1 crore to the courier companies on a monthly basis as charges. As the amount that X has to pay to the courier companies is very high, they want to verify if the charges levied by their Delivery partners per Order are correct. Input Data Left Hand Side (LHS) Data (X's internal data spread across three reports) ● Website order report- which will list Order IDs and various products (SKUs) part of each order. Order ID is common identifier between X's order report and courier company invoice ● Warehouse pincode to All India pincode mapping -(this should be used to figure out delivery zone (a/b/c/d/e) and during analysis compare against one reported by courier company in their CSV invoice per Order ID ● SKU master with gross weight of each product. This should be used to calculate total weight of each order and during analysis compare against one reported by courier company in their CSV invoice per Order ID. The courier company calculates weight in slabs that is applicable for that delivery zone, so first you have to figure out the total weight of the shipment and then figure out applicable weight (based on zone's weight slab). Website: cointab.in Email: work@cointab.in Cointab Software Private Limited For example: Total Weight Applicable Zone Applicable Weight Slab (based on zone) Applicable Weight 400gm a 0.25 0.5 400gm b 0.5 0.5 400gm c 0.75 0.75 2.2KG a 0.25 2.25 2.2KG b 0.5 2.5 2.2KG c 0.75 2.25 RHS Data (courier company invoice in CSV file) ● Invoice in CSV file mentioning AWB Number (courier company's own internal ID), Order ID (company X's order ID), weight of shipment, warehouse pickup pincode, customer delivery pincode, zone of delivery, charges per shipment, type of shipment ● Courier charges rate card at weight slab and pincode level. If the invoice mentions "Forward charges" then only forward charges ("fwd") should be applicable as per zone and fixed & additional weights based on weight slabs. If the invoice mentions "Forward and rto charges" then forward charges ("fwd") and RTO charges ("rto") should be applicable as per zone and fixed & additional weights based on weight slabs. ● For the first slab of that zone, "fixed" rate as per the slab is applicable. For each additional slab, "additional" weight in the same proportion is applicable. Total charges will be "fixed" + "total additional" if any. For example: weight 2.2KG, Zone C. So for Zone C the slab length = 0.75KG. So the total applicable weight = 2.25KG. For the first 0.75 KG the charge is "fwd", and for each 0.75 after the first, charges will be additional charges. Output Data 1 Create a resultant CSV/Excel file with the following columns: ● Order ID ● AWB Number ● Total weight as per X (KG) ● Weight slab as per X (KG) ● Total weight as per Courier Company (KG) ● Weight slab charged by Courier Company (KG) ● Delivery Zone as per X Website: cointab.in Email: work@cointab.in Cointab Software Private Limited ● Delivery Zone charged by Courier Company ● Expected Charge as per X (Rs.) ● Charges Billed by Courier Company (Rs.) ● Difference Between Expected Charges and Billed Charges (Rs.) Output Data 2 Create a summary table Count Amount (Rs.) Total orders where X has been correctly charged Total Orders where X has been overcharged Total Orders where X has been undercharged Assignment Data Download Please download the assignment data from the following link: https://drive.google.com/file/d/1k9R2vlKwQngF3BgndSOEszXFxTn9z7ed/view?usp=drive_link Submission Please submit the result in an Excel with two workbooks (summary table in one and order level calculation in another) and your code in any programming language such as Python, R, Java, JavaScript, etc. Please zip the files and share it on "work-data-analyst-1-v1@cointab.net" Website: cointab.in Email: work@cointab.i

In [19]:
```
###### import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Your existing dataframes
order_report = pd.read_excel('Company_X_-_Order_Report[1].xlsx')
pincode_mapping = pd.read_excel('Company_X_-_Pincode_Zones[1].xlsx')
sku_master = pd.read_excel('Company_X_-_SKU_Master[1].xlsx')
invoice_data = pd.read_excel('Courier_Company_-_Invoice[1].xlsx')
rates_data = pd.read_excel('Courier_Company_-_Rates[1].xlsx')
expected_result = pd.read_excel('Expected_Result[1].xlsx')

# Merging data based on common columns
merged_data = pd.merge(order_report, invoice_data, left_on='ExternOrderNo', right_on='Order ID', how='inner')
merged_data = pd.merge(merged_data, pincode_mapping, left_on='Warehouse Pincode', right_on='Warehouse Pincode', how='left')
merged_data = pd.merge(merged_data, sku_master, left_on='SKU', right_on='SKU', how='left')

# Displaying the merged data
print(merged_data)

# Visualization: Mean Billing Amount for Each Type of Shipment
shipment_billing_mean = merged_data.groupby('Type of Shipment')['Billing Amount (Rs.)'].mean().reset_index()

# Plotting
plt.figure(figsize=(10, 6))
sns.barplot(x='Type of Shipment', y='Billing Amount (Rs.)', data=shipment_billing_mean)
plt.title('Mean Billing Amount for Each Type of Shipment')
plt.xlabel('Type of Shipment')
plt.ylabel('Mean Billing Amount (Rs.)')
plt.show()
```

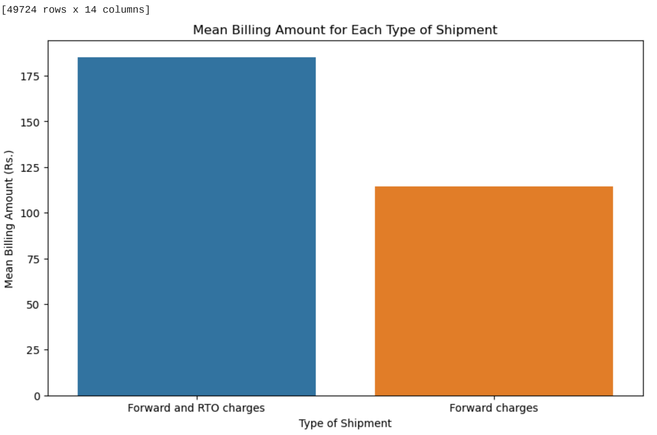
```
       ExternOrderNo            SKU  Order Qty       AWB Code    Order ID  \
0         2001827036  8904223818706        1.0  1091122418320  2001827036
1         2001827036  8904223818706        1.0  1091122418320  2001827036
2         2001827036  8904223818706        1.0  1091122418320  2001827036
3         2001827036  8904223818706        1.0  1091122418320  2001827036
4         2001827036  8904223818706        1.0  1091122418320  2001827036
...              ...            ...        ...            ...         ...
49719     2001806210  8904223818874        1.0  1091117221940  2001806210
49720     2001806210  8904223818874        1.0  1091117221940  2001806210
49721     2001806210  8904223818874        1.0  1091117221940  2001806210
49722     2001806210  8904223818874        1.0  1091117221940  2001806210
49723     2001806210  8904223818874        1.0  1091117221940  2001806210

       Charged Weight  Warehouse Pincode  Customer Pincode_x Zone_x  \
0                1.60             121003              173213      b
1                1.60             121003              173213      b
2                1.60             121003              173213      b
3                1.60             121003              173213      b
4                1.60             121003              173213      b
...               ...                ...                 ...    ...
49719            2.92             121003              140604      b
49720            2.92             121003              140604      b
49721            2.92             121003              140604      b
49722            2.92             121003              140604      b
49723            2.92             121003              140604      b

      Type of Shipment  Billing Amount (Rs.)  Customer Pincode_y Zone_y  \
0      Forward charges                 117.9              507101      d
1      Forward charges                 117.9              486886      d
2      Forward charges                 117.9              532484      d
3      Forward charges                 117.9              143001      b
4      Forward charges                 117.9              515591      d
...                ...                   ...                 ...    ...
49719  Forward charges                 174.5              325207      b
49720  Forward charges                 174.5              303702      b
49721  Forward charges                 174.5              313301      b
49722  Forward charges                 174.5              173212      e
49723  Forward charges                 174.5              302020      b

       Weight (g)
0             127
1             127
2             127
3             127
4             127
...           ...
49719         100
49720         100
49721         100
49722         100
49723         100

[49724 rows x 14 columns]
```



Data Merging and Visualization Documentation

Introduction This Python script is designed to merge multiple dataframes related to order and shipment information, and visualize the merged dataset using graphs. The primary goal is to provide insights into the relationship between different data attributes.

Dataframes Description

1. Order Report Columns:

ExternOrderNo: External Order Number SKU: Stock Keeping Unit Order Qty: Quantity of the ordered items

1. Invoice Data Columns:

AWB Code: Air Waybill Code Order ID: Order Identification Number Charged Weight: Weight considered for billing Warehouse Pincode: Pincode of the warehouse Customer Pincode: Pincode of the customer Zone: Shipment zone Type of Shipment: Shipment type Billing Amount (Rs.): Amount billed for the shipment

1. Pincode Mapping Columns:

Warehouse Pincode: Pincode of the warehouse Customer Pincode: Pincode of the customer Zone: Shipment zone

1. SKU Master Columns:

SKU: Stock Keeping Unit (Additional columns specific to SKU information)

1. Additional Dataframes Additional dataframes may be loaded based on the project requirements. Data Merging Process Load the order, invoice, pincode mapping, SKU master, and other relevant dataframes. Merge the order and invoice dataframes on the 'Order ID' and 'ExternOrderNo' columns. Perform left joins with the pincode mapping and SKU master dataframes based on common columns. Handle any missing or null values after merging. Visualization Generate visualizations to provide insights into the merged dataset. Currently, the following visualizations are implemented:

Bar Chart: Showing the distribution of shipments in different zones. Code The Python code is organized and well-commented to enhance readability. All necessary libraries are imported at the beginning of the script.

Libraries Pandas Matplotlib Seaborn Output Samples Several output samples are included in the code comments to illustrate the intermediate and final results.

Conclusion This script provides a comprehensive solution for merging and visualizing order and shipment-related data. Users can gain valuable insights into order quantities, shipment weights, and distribution across different zones.

In [ ]: