Introduction

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.

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
```

Inserting excel file

```
Company_X_Order_Report = pd.read_excel("D:\\Interview Questions\\Final
Assignment Data -\\New folder\\Company X - Order Report.xlsx")
Company_X_Pincode_Zones = pd.read_excel("D:\\Interview Questions\\
Final Assignment Data -\\New folder\\Company X - Pincode Zones.xlsx")
Company_X_SKU_Master = pd.read_excel("D:\\Interview Questions\\Final
Assignment Data -\\New folder\Company X - SKU Master.xlsx")
Courier_Company_Invoice = pd.read_excel("D:\\Interview Questions\\
Final Assignment Data -\\New folder\\Courier Company - Invoice.xlsx")
Courier_Company_Rates = pd.read_excel("D:\\Interview Questions\\Final
Assignment Data -\\New folder\\Courier Company - Rates.xlsx")
```

Delete Duplicates

```
Company_X_Order_Report = Company_X_Order_Report.drop_duplicates()

Company_X_Pincode_Zones = Company_X_Pincode_Zones.drop_duplicates()
```

```
Company_X_SKU_Master = Company_X_SKU_Master.drop_duplicates()

Courier_Company_Invoice = Courier_Company_Invoice.drop_duplicates()

Courier_Company_Rates = Courier_Company_Rates.drop_duplicates()
```

Checking the Information

```
Company_X_Order_Report.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 398 entries, 0 to 399
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   ExternOrderNo  398 non-null    int64
 1   SKU            398 non-null    object
 2   Order Qty      398 non-null    float64
dtypes: float64(1), int64(1), object(1)
memory usage: 12.4+ KB
```

```
Company_X_SKU_Master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 65 entries, 0 to 65
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   SKU         65 non-null     object
 1   Weight (g)  65 non-null     int64
dtypes: int64(1), object(1)
memory usage: 1.5+ KB
```

```
Company_X_Pincode_Zones.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 108 entries, 0 to 121
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Warehouse Pincode  108 non-null    int64
 1   Customer Pincode   108 non-null    int64
 2   Zone               108 non-null    object
dtypes: int64(2), object(1)
memory usage: 3.4+ KB
```

```
Courier_Company_Invoice.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 124 entries, 0 to 123
```

```
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   AWB Code              124 non-null    int64
 1   Order ID              124 non-null    int64
 2   Charged Weight        124 non-null    float64
 3   Warehouse Pincode     124 non-null    int64
 4   Customer Pincode      124 non-null    int64
 5   Zone                  124 non-null    object
 6   Type of Shipment      124 non-null    object
 7   Billing Amount (Rs.)  124 non-null    float64
dtypes: float64(2), int64(4), object(2)
memory usage: 7.9+ KB
```

```
Courier_Company_Rates.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 6 columns):
 #   Column                                  Non-Null Count  Dtype
---  ------                                  --------------  -----
 0   Zone                                    5 non-null      object
 1   Weight Slabs                            5 non-null      float64
 2   Forward Fixed Charge                    5 non-null      float64
 3   Forward Additional Weight Slab Charge   5 non-null      float64
 4   RTO Fixed Charge                        5 non-null      float64
 5   RTO Additional Weight Slab Charge       5 non-null      float64
dtypes: float64(5), object(1)
memory usage: 368.0+ bytes
```

Checking the number of NULL value

```
Company_X_Order_Report.isnull().sum()
```

```
ExternOrderNo    0
SKU              0
Order Qty        0
dtype: int64
```

```
Courier_Company_Rates.isnull().sum()
```

```
Zone                                    0
Weight Slabs                            0
Forward Fixed Charge                    0
Forward Additional Weight Slab Charge   0
RTO Fixed Charge                        0
RTO Additional Weight Slab Charge       0
dtype: int64
```

```
Courier_Company_Invoice.isnull().sum()
```

```
AWB Code                0
Order ID                0
Charged Weight          0
Warehouse Pincode       0
Customer Pincode        0
Zone                    0
Type of Shipment        0
Billing Amount (Rs.)    0
dtype: int64
```

```
Company_X_Pincode_Zones.isnull().sum()
```

```
Warehouse Pincode    0
Customer Pincode     0
Zone                 0
dtype: int64
```

```
Company_X_SKU_Master.isnull().sum()
```

```
SKU           0
Weight (g)    0
dtype: int64
```

```
Company_X_Order_Report =
Company_X_Order_Report.merge(Company_X_SKU_Master,on = 'SKU') ##
merging two tables

Company_X_Order_Report['Weight (KG)'] = Company_X_Order_Report['Weight
(g)']/1000 ##convert gm to KG

Company_X_Order_Report['Total Weight (KG)'] =
Company_X_Order_Report['Weight (KG)']*Company_X_Order_Report['Order
Qty'] ## Adding one columns with Total weight

Company_X_Order_Report.rename(columns = {'ExternOrderNo':'Order ID'},
inplace = True) ## Change column name

Final_Company_X_Order_Report = Company_X_Order_Report.groupby('Order
ID', as_index = False).agg({'Total Weight (KG)':'sum'}) ## Summerize
using Group by to check the total weight per order

Final_Company_X_Order_Report.head() ## Checking the first 5 values for
New created table
```

	Order ID	Total Weight (KG)
0	2001806210	0.220
1	2001806226	0.480
2	2001806229	0.500
3	2001806232	1.302
4	2001806233	0.245

```
Courier_Company_Invoice =
Courier_Company_Invoice.merge(Final_Company_X_Order_Report, on  =
'Order ID') ## Merging two table

Courier_Company_Invoice.head()
```

```
        AWB Code    Order ID  Charged Weight  Warehouse Pincode  \
0  1091117222124  2001806232            1.30             121003
1  1091117222194  2001806273            1.00             121003
2  1091117222931  2001806408            2.50             121003
3  1091117223244  2001806458            1.00             121003
4  1091117229345  2001807012            0.15             121003

   Customer Pincode Zone Type of Shipment  Billing Amount (Rs.)  \
0            507101    d  Forward charges                 135.0
1            486886    d  Forward charges                  90.2
2            532484    d  Forward charges                 224.6
3            143001    b  Forward charges                  61.3
4            515591    d  Forward charges                  45.4

   Total Weight (KG)
0              1.302
1              0.615
2              2.265
3              0.700
4              0.240
```

```
Courier_Company_Invoice =
Courier_Company_Invoice.merge(Company_X_Pincode_Zones,on = 'Customer
Pincode') ## Merging the invoice table with pincode table
```

Rename some of the columns

```
Courier_Company_Invoice.rename(columns = {'Warehouse
Pincode_x':'Warehouse Pincode','Zone_x':'Delivery Zone charged by
Courier Company','Zone_y':'Delivery Zone as per X'}, inplace = True)

Courier_Company_Invoice.rename(columns = {'Total Weight (KG)':'Total
weight as per X (KG)'},inplace = True)

Courier_Company_Rates.rename(columns = {'Zone':'Delivery Zone charged
by Courier Company'}, inplace = True)
```

As Python is case sensitive thats why changing the value of Zone column from Rates Table to do the marge with invoice table

```
Courier_Company_Rates['Delivery Zone charged by Courier Company'] =
Courier_Company_Rates['Delivery Zone charged by Courier
Company'].replace('A','a')

Courier_Company_Rates['Delivery Zone charged by Courier Company'] =
Courier_Company_Rates['Delivery Zone charged by Courier
Company'].replace('B','b')

Courier_Company_Rates['Delivery Zone charged by Courier Company'] =
Courier_Company_Rates['Delivery Zone charged by Courier
Company'].replace('C','c')

Courier_Company_Rates['Delivery Zone charged by Courier Company'] =
Courier_Company_Rates['Delivery Zone charged by Courier
Company'].replace('D','d')

Courier_Company_Rates['Delivery Zone charged by Courier Company'] =
Courier_Company_Rates['Delivery Zone charged by Courier
Company'].replace('E','e')
```

Merge two tables

```
Courier_Company_Invoice =
Courier_Company_Invoice.merge(Courier_Company_Rates, on  = 'Delivery
Zone charged by Courier Company')

Courier_Company_Invoice.rename(columns = {'Weight Slabs':'Weight slab
charged by Courier Company (KG)'},inplace = True)

Courier_Company_Rates.rename(columns = {'Delivery Zone charged by
Courier Company':'Delivery Zone as per X'}, inplace = True)

Courier_Company_Invoice =
Courier_Company_Invoice.merge(Courier_Company_Rates,on  = 'Delivery
Zone as per X')

Courier_Company_Invoice['Total Amount (Rs.)'] = np.nan ## Create one
null column
```

Creating column **Expected Charge as per X (Rs.)** below

```
Courier_Company_Invoice['Total Amount (Rs.)']
=np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='d') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
```

```
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='e') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='c') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='b') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='b') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge'],

np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='b') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
Courier_Company_Invoice['Forward Additional Weight Slab Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='d') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
Courier_Company_Invoice['Forward Additional Weight Slab Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='c') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
```

```
Courier_Company_Invoice['Forward Additional Weight Slab Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='e') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
Courier_Company_Invoice['Forward Additional Weight Slab Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='a') &
(Courier_Company_Invoice['Type of Shipment']=='Forward charges') &
(Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
Courier_Company_Invoice['Forward Additional Weight Slab Charge'],

np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='d') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='b') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='c') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='e') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge'],
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='a') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']<=Courier_Company_Invoice['Weight
Slabs'])),Courier_Company_Invoice['Forward Fixed Charge']+
```

```
Courier_Company_Invoice['RTO Fixed Charge'],



np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='a') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
(Courier_Company_Invoice['Forward Additional Weight Slab Charge']+
Courier_Company_Invoice['RTO Additional Weight Slab Charge']),
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='b') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
(Courier_Company_Invoice['Forward Additional Weight Slab Charge']+
Courier_Company_Invoice['RTO Additional Weight Slab Charge']),
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='c') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
(Courier_Company_Invoice['Forward Additional Weight Slab Charge']+
Courier_Company_Invoice['RTO Additional Weight Slab Charge']),
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='d') &
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
(Courier_Company_Invoice['Forward Additional Weight Slab Charge']+
Courier_Company_Invoice['RTO Additional Weight Slab Charge']),
np.where(((Courier_Company_Invoice['Delivery Zone as per X']=='e') &
```

```
(Courier_Company_Invoice['Type of Shipment']=='Forward and RTO
charges') & (Courier_Company_Invoice['Total weight as per X
(KG)']>Courier_Company_Invoice['Weight Slabs'])),
Courier_Company_Invoice['Forward Fixed Charge']+
Courier_Company_Invoice['RTO Fixed Charge']+
round((Courier_Company_Invoice['Total weight as per X (KG)']-
Courier_Company_Invoice['Weight Slabs'])/
Courier_Company_Invoice['Weight Slabs'],0)*
(Courier_Company_Invoice['Forward Additional Weight Slab Charge']+
Courier_Company_Invoice['RTO Additional Weight Slab
Charge']),0))))))))))))))))))
```

Rename columns as per requirements

```
Courier_Company_Invoice.rename(columns = {'Weight Slabs':'Weight slab
as per X (KG)'}, inplace = True)

Courier_Company_Invoice.rename(columns = {'Charged Weight':'Total
weight as per Courier Company (KG)'}, inplace = True)

Courier_Company_Invoice.rename(columns = {'Total Amount
(Rs.)':'Expected Charge as per X (Rs.)'}, inplace = True)

Courier_Company_Invoice.rename(columns = {'Billing Amount
(Rs.)':'Charges Billed by Courier Company (Rs.)'}, inplace = True)
```

Creating columns where we can see the different between expected bill amount and charged bill amount

```
Courier_Company_Invoice['Difference Between Expected Charges and
Billed Charges (Rs.)'] = round(Courier_Company_Invoice['Charges Billed
by Courier Company (Rs.)']-Courier_Company_Invoice['Expected Charge as
per X (Rs.)'],2)



Courier_Company_Invoice.rename(columns = {'AWB Code':'AWB Number'},
inplace = True) ## Rename column name
```

Create the table with required columns

```
Courier_Company_Invoice_New = Courier_Company_Invoice[['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','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.)']]

Courier_Company_Invoice_New.head()
```

```
     Order ID     AWB Number  Total weight as per X (KG)  \
0  2001806232  1091117222124                       1.302
1  2001806273  1091117222194                       0.615
2  2001806408  1091117222931                       2.265
3  2001807012  1091117229345                       0.240
4  2001806686  1091117229555                       0.240

   Weight slab as per X (KG)  Total weight as per Courier Company (KG)
\
0                        1.5                                      1.30

1                        1.5                                      1.00

2                        1.5                                      2.50

3                        1.5                                      0.15

4                        1.5                                      0.15


   Weight slab charged by Courier Company (KG) Delivery Zone as per X
\
0                                         1.5                       d

1                                         1.5                       d

2                                         1.5                       d

3                                         1.5                       d

4                                         1.5                       d


  Delivery Zone charged by Courier Company  Expected Charge as per X
(Rs.)  \
0                                        d
45.4
1                                        d
45.4
2                                        d
90.2
3                                        d
45.4
```

```
4                                              d
45.4

   Charges Billed by Courier Company (Rs.)  \
0                                    135.0
1                                     90.2
2                                    224.6
3                                     45.4
4                                     45.4

   Difference Between Expected Charges and Billed Charges (Rs.)
0                                                  89.6
1                                                  44.8
2                                                 134.4
3                                                   0.0
4                                                   0.0
```

```
Courier_Company_Invoice_New.to_csv('Output1.csv') ### Create CSV file
with required column
```

Summary Table

```
correctly_charged =
Courier_Company_Invoice_New[Courier_Company_Invoice_New['Difference
Between Expected Charges and Billed Charges (Rs.)']==0.0]
correctly=["Total orders where X has been correctly
charged",len(correctly_charged),sum(correctly_charged['Charges Billed
by Courier Company (Rs.)'])]

over_charged =
Courier_Company_Invoice_New[Courier_Company_Invoice_New['Difference
Between Expected Charges and Billed Charges (Rs.)']>0.0]
over=["Total Orders where X has been
overcharged",len(over_charged),sum(over_charged['Difference Between
Expected Charges and Billed Charges (Rs.)'])]

under_charged =
Courier_Company_Invoice_New[Courier_Company_Invoice_New['Difference
Between Expected Charges and Billed Charges (Rs.)']<0.0]
under=["Total Orders where X has been
undercharged",len(under_charged),np.abs(sum(under_charged['Difference
Between Expected Charges and Billed Charges (Rs.)']))]



Result_Output2 =
pd.DataFrame([correctly,over,under],columns=['Description','Count','Am
ount (Rs.)'])
```

```
Result_Output2
```

	Description	Count	Amount (Rs.)
0	Total orders where X has been correctly charged	7	367.3
1	Total Orders where X has been overcharged	115	8201.6
2	Total Orders where X has been undercharged	2	47.2

```
Result_Output2.to_csv("Output2.csv")  ### Create CSV file with Summary
```