In [1]:

```python
import numpy as np
import pandas as pd
```

In [2]:

```python
df_OR = pd.read_excel('Company X - Order Report.xlsx')
```

In [3]:

```python
df_OR
```

Out[3]:

	ExternOrderNo	SKU	Order Qty
0	2001827036	8904223818706	1.0
1	2001827036	8904223819093	1.0
2	2001827036	8904223819109	1.0
3	2001827036	8904223818430	1.0
4	2001827036	8904223819277	1.0
...	...	...	...
395	2001806229	8904223818942	1.0
396	2001806229	8904223818850	1.0
397	2001806226	8904223818850	2.0
398	2001806210	8904223816214	1.0
399	2001806210	8904223818874	1.0

400 rows × 3 columns

In [4]:

```python
df_OR.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 3 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   ExternOrderNo  400 non-null    int64  
 1   SKU            400 non-null    object 
 2   Order Qty      400 non-null    float64
dtypes: float64(1), int64(1), object(1)
memory usage: 9.5+ KB
```

In [5]:

```python
df_SM = pd.read_excel('Company X - SKU Master.xlsx')
```

In [6]:

```python
df_SM
```

Out[6]:

	SKU	Weight (g)
0	8904223815682	210
1	8904223815859	165
2	8904223815866	113
3	8904223815873	65
4	8904223816214	120
...	...	...
61	8904223819505	210
62	8904223819499	210
63	8904223819512	210
64	8904223819543	300
65	SACHETS001	10

66 rows × 2 columns

In [7]:

```python
df_SM.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66 entries, 0 to 65
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   SKU         66 non-null     object
 1   Weight (g)  66 non-null     int64 
dtypes: int64(1), object(1)
memory usage: 1.2+ KB
```

In [8]:
```
df_PZ = pd.read_excel('Company X - Pincode Zones.xlsx')
```

In [9]:
```
df_PZ
```

Out[9]:

	Warehouse Pincode	Customer Pincode	Zone
0	121003	507101	d
1	121003	486886	d
2	121003	532484	d
3	121003	143001	b
4	121003	515591	d
...	...	...	...
119	121003	325207	b
120	121003	303702	b
121	121003	313301	b
122	121003	173212	e
123	121003	302020	b

124 rows × 3 columns

In [10]:
```
df_PZ.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 124 entries, 0 to 123
Data columns (total 3 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Warehouse Pincode  124 non-null    int64 
 1   Customer Pincode   124 non-null    int64 
 2   Zone               124 non-null    object
dtypes: int64(2), object(1)
memory usage: 3.0+ KB
```

In [11]:
```
df_CI = pd.read_excel('Courier Company - Invoice.xlsx')
```

In [12]:
```
df_CI
```

Out[12]:

	AWB Code	Order ID	Charged Weight	Warehouse Pincode	Customer Pincode	Zone	Type of Shipment	Billing Amount (Rs.)
0	1091117222124	2001806232	1.30	121003	507101	d	Forward charges	135.0
1	1091117222194	2001806273	1.00	121003	486886	d	Forward charges	90.2
2	1091117222931	2001806408	2.50	121003	532484	d	Forward charges	224.6
3	1091117223244	2001806458	1.00	121003	143001	b	Forward charges	61.3
4	1091117229345	2001807012	0.15	121003	515591	d	Forward charges	45.4
...	...	...	...	...	...	...	...	...
119	1091118551656	2001812941	0.73	121003	325207	d	Forward charges	90.2
120	1091117614452	2001809383	0.50	121003	303702	d	Forward and RTO charges	86.7
121	1091120922803	2001820978	0.50	121003	313301	d	Forward charges	45.4
122	1091121844806	2001811475	0.50	121003	173212	b	Forward charges	33.0
123	1091121846136	2001811305	0.50	121003	302020	d	Forward charges	45.4

124 rows × 8 columns

In [13]:
```
df_CI['Order ID']
```

```
Out[13]: 0      2001806232
         1      2001806273
         2      2001806408
         3      2001806458
         4      2001807012
                   ...
         119    2001812941
         120    2001809383
         121    2001820978
         122    2001811475
         123    2001811305
         Name: Order ID, Length: 124, dtype: int64
```

```
In [14]: df_CI.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
```

```
In [15]: df_CR = pd.read_excel('Courier Company - Rates.xlsx')
```

```
In [16]: df_CR
```

Out[16]:

	fwd_a_fixed	fwd_a_additional	fwd_b_fixed	fwd_b_additional	fwd_c_fixed	fwd_c_additional	fwd_d_fixed	fwd_d_additional	fwd_e_fixed	fwd_e_additic
0	29.5	23.6	33	28.3	40.1	38.9	45.4	44.8	56.6	5

```
In [17]: df_CR.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   fwd_a_fixed       1 non-null      float64
 1   fwd_a_additional  1 non-null      float64
 2   fwd_b_fixed       1 non-null      int64  
 3   fwd_b_additional  1 non-null      float64
 4   fwd_c_fixed       1 non-null      float64
 5   fwd_c_additional  1 non-null      float64
 6   fwd_d_fixed       1 non-null      float64
 7   fwd_d_additional  1 non-null      float64
 8   fwd_e_fixed       1 non-null      float64
 9   fwd_e_additional  1 non-null      float64
 10  rto_a_fixed       1 non-null      float64
 11  rto_a_additional  1 non-null      float64
 12  rto_b_fixed       1 non-null      float64
 13  rto_b_additional  1 non-null      float64
 14  rto_c_fixed       1 non-null      float64
 15  rto_c_additional  1 non-null      float64
 16  rto_d_fixed       1 non-null      float64
 17  rto_d_additional  1 non-null      float64
 18  rto_e_fixed       1 non-null      float64
 19  rto_e_additional  1 non-null      float64
dtypes: float64(19), int64(1)
memory usage: 292.0 bytes
```

```
In [18]: column_name =  ['Order ID']
```

```
In [19]: df = pd.DataFrame(columns=column_name)
```

```
In [20]: df
```

Out[20]:

Order ID

```
In [21]: df['Order ID'] = df_OR['ExternOrderNo']
```

```
In [22]: df
```

Out[22]:

	Order ID
0	2001827036
1	2001827036
2	2001827036
3	2001827036
4	2001827036
...	...
395	2001806229
396	2001806229
397	2001806226
398	2001806210
399	2001806210

400 rows × 1 columns

In [23]:

```
merge_data = pd.merge(df_OR,df_SM,on = 'SKU',how = 'left')
```

In [24]:

```
merge_data
```

Out[24]:

	ExternOrderNo	SKU	Order Qty	Weight (g)
0	2001827036	8904223818706	1.0	127
1	2001827036	8904223819093	1.0	150
2	2001827036	8904223819109	1.0	100
3	2001827036	8904223818430	1.0	165
4	2001827036	8904223819277	1.0	350
...	...	...	...	...
396	2001806229	8904223818942	1.0	133
397	2001806229	8904223818850	1.0	240
398	2001806226	8904223818850	2.0	240
399	2001806210	8904223816214	1.0	120
400	2001806210	8904223818874	1.0	100

401 rows × 4 columns

In [25]:

```
merge_data['Total weight as per X (KG)'] = merge_data['Order Qty']*merge_data['Weight (g)']/1000
```

In [26]:

```
merge_data
```

Out[26]:

	ExternOrderNo	SKU	Order Qty	Weight (g)	Total weight as per X (KG)
0	2001827036	8904223818706	1.0	127	0.127
1	2001827036	8904223819093	1.0	150	0.150
2	2001827036	8904223819109	1.0	100	0.100
3	2001827036	8904223818430	1.0	165	0.165
4	2001827036	8904223819277	1.0	350	0.350
...	...	...	...	...	...
396	2001806229	8904223818942	1.0	133	0.133
397	2001806229	8904223818850	1.0	240	0.240
398	2001806226	8904223818850	2.0	240	0.480
399	2001806210	8904223816214	1.0	120	0.120
400	2001806210	8904223818874	1.0	100	0.100

401 rows × 5 columns

```
In [27]: columns_drop = ['SKU','Order Qty','Weight (g)']
```

```
In [28]: c_drop = merge_data.drop(columns_drop,axis = 1)
```

```
In [29]: c_drop
```

Out[29]:

	ExternOrderNo	Total weight as per X (KG)
0	2001827036	0.127
1	2001827036	0.150
2	2001827036	0.100
3	2001827036	0.165
4	2001827036	0.350
...	...	...
396	2001806229	0.133
397	2001806229	0.240
398	2001806226	0.480
399	2001806210	0.120
400	2001806210	0.100

401 rows × 2 columns

```
In [30]: group_by = pd.DataFrame(c_drop.groupby(['ExternOrderNo'],as_index = False).sum().round(2))
```

```
In [31]: group_by
```

Out[31]:

	ExternOrderNo	Total weight as per X (KG)
0	2001806210	0.22
1	2001806226	0.48
2	2001806229	0.50
3	2001806232	1.30
4	2001806233	0.24
...	...	...
119	2001821995	0.48
120	2001822466	1.38
121	2001823564	0.67
122	2001825261	1.56
123	2001827036	2.18

124 rows × 2 columns

In [32]:
```
group_by.rename(columns={'ExternOrderNo':'Order ID'},inplace = True)
```

In [33]:
```
value = lambda x : 0.5 if(x <= 0.5) else (1.0 if (x > 0.5 and x<= 1.0) else (1.5 if(x>1.0 and x <=1.5) else (2.0 if(x >1.5 and x
```

In [34]:
```
group_by['Weight slab as per X (KG)'] = group_by['Total weight as per X (KG)'].apply(value)
```

In [35]:
```
group_by
```

Out[35]:

	Order ID	Total weight as per X (KG)	Weight slab as per X (KG)
0	2001806210	0.22	0.5
1	2001806226	0.48	0.5
2	2001806229	0.50	0.5
3	2001806232	1.30	1.5
4	2001806233	0.24	0.5
...	...	...	...
119	2001821995	0.48	0.5
120	2001822466	1.38	1.5
121	2001823564	0.67	1.0
122	2001825261	1.56	2.0
123	2001827036	2.18	2.5

124 rows × 3 columns

In [36]:
```
merg_data = pd.merge(group_by,df_CI,on = 'Order ID',how = 'left')
```

In [37]:
```
merg_data
```

Out[37]:

	Order ID	Total weight as per X (KG)	Weight slab as per X (KG)	AWB Code	Charged Weight	Warehouse Pincode	Customer Pincode	Zone	Type of Shipment	Billing Amount (Rs.)
0	2001806210	0.22	0.5	1091117221940	2.92	121003	140604	b	Forward charges	174.5
1	2001806226	0.48	0.5	1091117222065	0.68	121003	723146	d	Forward charges	90.2
2	2001806229	0.50	0.5	1091117222080	0.71	121003	421204	d	Forward charges	90.2
3	2001806232	1.30	1.5	1091117222124	1.30	121003	507101	d	Forward charges	135.0
4	2001806233	0.24	0.5	1091117222135	0.78	121003	263139	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	0.48	0.5	1091121183730	0.50	121003	342008	d	Forward charges	45.4
120	2001822466	1.38	1.5	1091121305541	1.10	121003	342301	d	Forward charges	135.0
121	2001823564	0.67	1.0	1091121666133	0.70	121003	492001	d	Forward and RTO charges	172.8
122	2001825261	1.56	2.0	1091121981575	1.60	121003	517128	d	Forward and RTO charges	345.0
123	2001827036	2.18	2.5	1091122418320	1.60	121003	173213	b	Forward charges	117.9

124 rows × 10 columns

In [38]:
```
columns = merg_data.columns.tolist()
```

In [39]:
```
columns.insert(1,columns.pop(3))
```

In [40]:
```
arrange = merg_data[columns]
```

In [41]:
```
arrange
```

Out[41]:

	Order ID	AWB Code	Total weight as per X (KG)	Weight slab as per X (KG)	Charged Weight	Warehouse Pincode	Customer Pincode	Zone	Type of Shipment	Billing Amount (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	121003	140604	b	Forward charges	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	121003	723146	d	Forward charges	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	121003	421204	d	Forward charges	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	121003	507101	d	Forward charges	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	121003	263139	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	121003	342008	d	Forward charges	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	121003	342301	d	Forward charges	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	121003	492001	d	Forward and RTO charges	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	121003	517128	d	Forward and RTO charges	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	121003	173213	b	Forward charges	117.9

124 rows × 10 columns

In [42]:
```
arrange.rename(columns={'Charged Weight':'Total weight as per Courier Company (KG)','Zone':'Delivery Zone charged by Courier Comp
```

In [43]:
```
arrange
```

Out[43]:

	Order ID	AWB Code	Total weight as per X (KG)	Weight slab as per X (KG)	Total weight as per Courier Company (KG)	Warehouse Pincode	Customer Pincode	Delivery Zone charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	121003	140604	b	Forward charges	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	121003	723146	d	Forward charges	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	121003	421204	d	Forward charges	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	121003	507101	d	Forward charges	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	121003	263139	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	121003	342008	d	Forward charges	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	121003	342301	d	Forward charges	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	121003	492001	d	Forward and RTO charges	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	121003	517128	d	Forward and RTO charges	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	121003	173213	b	Forward charges	117.9

124 rows × 10 columns

In [44]:
```
arrange.insert(5,'Weight slab charged by Courier Company (KG)',arrange['Total weight as per Courier Company (KG)'].apply(value))
```

In [45]:
```
arrange
```

Out[45]:

	Order ID	AWB Code	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)	Warehouse Pincode	Customer Pincode	Delivery Zone charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	121003	140604	b	Forward charges	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	121003	723146	d	Forward charges	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	121003	421204	d	Forward charges	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	121003	507101	d	Forward charges	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	121003	263139	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	121003	342008	d	Forward charges	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	121003	342301	d	Forward charges	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	121003	492001	d	Forward and RTO charges	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	121003	517128	d	Forward and RTO charges	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	121003	173213	b	Forward charges	117.9

124 rows × 11 columns

In [46]:
```
com = pd.merge(arrange,df_PZ,on = 'Customer Pincode',how = 'left')
```

```
In [47]: com
```

Out[47]:

	Order ID	AWB Code	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)	Warehouse Pincode_x	Customer Pincode	Delivery Zone charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)	Warehouse Pincode_y	Zone
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	121003	140604	b	Forward charges	174.5	121003	b
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	121003	723146	d	Forward charges	90.2	121003	d
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	121003	421204	d	Forward charges	90.2	121003	d
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	121003	507101	d	Forward charges	135.0	121003	d
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	121003	263139	b	Forward charges	61.3	121003	b
...	...	...	...	...	...	...	...	...	...	...	...	...	...
169	2001821995	1091121183730	0.48	0.5	0.50	0.5	121003	342008	d	Forward charges	45.4	121003	b
170	2001822466	1091121305541	1.38	1.5	1.10	1.5	121003	342301	d	Forward charges	135.0	121003	b
171	2001823564	1091121666133	0.67	1.0	0.70	1.0	121003	492001	d	Forward and RTO charges	172.8	121003	d
172	2001825261	1091121981575	1.56	2.0	1.60	2.0	121003	517128	d	Forward and RTO charges	345.0	121003	d
173	2001827036	1091122418320	2.18	2.5	1.60	2.0	121003	173213	b	Forward charges	117.9	121003	e

174 rows × 13 columns

```
In [48]: dup = com.drop_duplicates(subset='Order ID')
```

```
In [49]: d = dup.reset_index(drop=True)
```

```
In [50]: d
```

Out[50]:

	Order ID	AWB Code	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)	Warehouse Pincode_x	Customer Pincode	Delivery Zone charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)	Warehouse Pincode_y	Zone
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	121003	140604	b	Forward charges	174.5	121003	b
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	121003	723146	d	Forward charges	90.2	121003	d
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	121003	421204	d	Forward charges	90.2	121003	d
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	121003	507101	d	Forward charges	135.0	121003	d
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	121003	263139	b	Forward charges	61.3	121003	b
...	...	...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	121003	342008	d	Forward charges	45.4	121003	b
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	121003	342301	d	Forward charges	135.0	121003	b
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	121003	492001	d	Forward and RTO charges	172.8	121003	d
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	121003	517128	d	Forward and RTO charges	345.0	121003	d
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	121003	173213	b	Forward charges	117.9	121003	e

124 rows × 13 columns

In [51]:
```
columns_d = ['Warehouse Pincode_x','Customer Pincode','Warehouse Pincode_y']
```

In [52]:
```
_drop = d.drop(columns_d,axis = 1)
```

In [53]:
```
_drop
```

Out[53]:

	Order ID	AWB Code	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 charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)	Zone
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	Forward charges	174.5	b
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	Forward charges	90.2	d
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	Forward charges	90.2	d
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	Forward charges	135.0	d
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	Forward charges	61.3	b
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	d	Forward charges	45.4	b
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	d	Forward charges	135.0	b
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	Forward and RTO charges	172.8	d
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	Forward and RTO charges	345.0	d
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	b	Forward charges	117.9	e

124 rows × 10 columns

In [54]:
```
columns = _drop.columns.tolist()
```

In [55]:

```
columns.insert(6,columns.pop(9))
```

In [56]:

```
_drop = _drop[columns]
```

In [57]:

```
_drop
```

Out[57]:

	Order ID	AWB Code	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)	Zone	Delivery Zone charged by Courier Company	Type of Shipment	Charges Billed by Courier Company (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	Forward charges	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	Forward charges	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	Forward charges	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	Forward charges	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	Forward charges	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	Forward charges	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	Forward and RTO charges	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	Forward and RTO charges	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	Forward charges	117.9

124 rows × 10 columns

In [58]:

```
_drop.rename(columns={'Zone':'Delivery Zone as per X'},inplace = True)
```

In [59]:

```
_drop
```

Out[59]:

	Order ID	AWB Code	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	Type of Shipment	Charges Billed by Courier Company (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	Forward charges	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	Forward charges	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	Forward charges	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	Forward charges	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	Forward charges	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	Forward charges	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	Forward charges	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	Forward and RTO charges	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	Forward and RTO charges	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	Forward charges	117.9

124 rows × 10 columns

In [60]:
```
def calculate_value (row):

    if row['Type of Shipment'] == 'Forward charges' and row['Delivery Zone as per X'] == 'a':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*23.6+29.5
       return result
    elif row['Type of Shipment'] == 'Forward charges' and row['Delivery Zone as per X'] == 'b':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*28.3+33
       return result
    elif row['Type of Shipment'] == 'Forward charges' and row['Delivery Zone as per X'] == 'c':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*38.9+40.1
       return result
    elif row['Type of Shipment'] == 'Forward charges' and row['Delivery Zone as per X'] == 'd':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*44.8+45.4
       return result
    elif row['Type of Shipment'] == 'Forward charges' and row['Delivery Zone as per X'] == 'e':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*55.5+56.6
       return result
    elif row['Type of Shipment'] == 'Forward and RTO charges' and row['Delivery Zone as per X'] == 'a':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*(23.6*2)+29.5+13.6
       return result
    elif row['Type of Shipment'] == 'Forward and RTO charges' and row['Delivery Zone as per X'] == 'b':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*(28.3*2)+33+20.5
       return result
    elif row['Type of Shipment'] == 'Forward and RTO charges' and row['Delivery Zone as per X'] == 'c':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*(38.9*2)+40.1+31.9
       return result
    elif row['Type of Shipment'] == 'Forward and RTO charges' and row['Delivery Zone as per X'] == 'd':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*(44.8*2)+45.4+41.3
       return result
    elif row['Type of Shipment'] == 'Forward and RTO charges' and row['Delivery Zone as per X'] == 'e':
       result = (row['Weight slab as per X (KG)']-0.5)/0.5*(55.5*2)+56.6+50.7
       return result
    else:
       return None
```

In [61]:
```
_drop.insert(9,'Expected Charge as per X (Rs.)', _drop.apply(lambda row : calculate_value (row),axis=1))
```

In [62]:
```
_drop
```

Out[62]:

	Order ID	AWB Code	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	Type of Shipment	Expected Charge as per X (Rs.)	Charges Billed by Courier Company (Rs.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	Forward charges	33.0	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	Forward charges	45.4	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	Forward charges	45.4	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	Forward charges	135.0	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	Forward charges	33.0	61.3
...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	Forward charges	33.0	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	Forward charges	89.6	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	Forward and RTO charges	176.3	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	Forward and RTO charges	355.5	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	Forward charges	278.6	117.9

124 rows × 11 columns

In [63]:
```
df = _drop.drop('Type of Shipment',axis = 1)
```

In [64]:
```
df
```

Out[64]:

	Order ID	AWB Code	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.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	33.0	174.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	45.4	90.2
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	45.4	90.2
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	135.0	135.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	33.0	61.3
...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	33.0	45.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	89.6	135.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	176.3	172.8
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	355.5	345.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	278.6	117.9

124 rows × 10 columns

In [65]:
```
df['Difference Between Expected Charges and Billed Charges (Rs.)'] = df['Expected Charge as per X (Rs.)'] - df['Charges Billed by
```

In [66]:
```
df
```

Out[66]:

	Order ID	AWB Code	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.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	33.0	174.5	-141.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	45.4	90.2	-44.8
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	45.4	90.2	-44.8
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	135.0	135.0	0.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	33.0	61.3	-28.3
...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	33.0	45.4	-12.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	89.6	135.0	-45.4
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	176.3	172.8	3.5
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	355.5	345.0	10.5
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	278.6	117.9	160.7

124 rows × 11 columns

In [67]:
```
writer = pd.ExcelWriter("Logistics Assignment calculation.xlsx")
```

In [68]:
```
df.to_excel(writer,sheet_name="Logistic Assignment Analysis")
```

In [69]:
```
writer._save()
```

In [70]:
```
d =  {" " : ['Total orders where X has been correctly charged']}
```

In [71]:
```
df1 = pd.DataFrame(data=d)
```

In [72]:
```
df1
```

Out[72]:

0	Total orders where X has been correctly charged

In [73]:
```
df1['Count'] = (df['Difference Between Expected Charges and Billed Charges (Rs.)']==0).sum()
```

In [74]:
```
df1
```

Out[74]:

		Count
0	Total orders where X has been correctly charged	17

In [75]:
```
df1['Amount (Rs.)'] = df['Expected Charge as per X (Rs.)'].sum()
```

In [76]:
```
df1
```

Out[76]:

		Count	Amount (Rs.)
0	Total orders where X has been correctly charged	17	9795.6

In [77]:
```
df['over'] = np.where(df['Difference Between Expected Charges and Billed Charges (Rs.)'] <0,df['Difference Between Expected Charg
```

In [78]:
```
df
```

Out[78]:

	Order ID	AWB Code	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.)	over
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	33.0	174.5	-141.5	-141.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	45.4	90.2	-44.8	-44.8
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	45.4	90.2	-44.8	-44.8
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	135.0	135.0	0.0	0.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	33.0	61.3	-28.3	-28.3
...	...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	33.0	45.4	-12.4	-12.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	89.6	135.0	-45.4	-45.4
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	176.3	172.8	3.5	0.0
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	355.5	345.0	10.5	0.0
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	278.6	117.9	160.7	0.0

124 rows × 12 columns

In [79]:
```
df['under'] = np.where(df['Difference Between Expected Charges and Billed Charges (Rs.)'] >0,df['Difference Between Expected Char
```

In [80]:
```
df
```

Out[80]:

	Order ID	AWB Code	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.)	over	under
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	33.0	174.5	-141.5	-141.5	0.0
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	45.4	90.2	-44.8	-44.8	0.0
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	45.4	90.2	-44.8	-44.8	0.0
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	135.0	135.0	0.0	0.0	0.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	33.0	61.3	-28.3	-28.3	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	33.0	45.4	-12.4	-12.4	0.0
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	89.6	135.0	-45.4	-45.4	0.0
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	176.3	172.8	3.5	0.0	3.5
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	355.5	345.0	10.5	0.0	10.5
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	278.6	117.9	160.7	0.0	160.7

124 rows × 13 columns

```
In [81]: a =  {" " : ['Total Orders where X has been overcharged']}
```

```
In [82]: df2 = pd.DataFrame(data=a)
```

```
In [83]: df2
```

Out[83]:

0	Total Orders where X has been overcharged

```
In [84]: df2['Count'] = (df['Difference Between Expected Charges and Billed Charges (Rs.)']<0).sum()
```

```
In [85]: df2
```

Out[85]:

		Count
0	Total Orders where X has been overcharged	84

```
In [86]: df2['Amount (Rs.)'] = df['over'].sum()
```

```
In [87]: df2
```

Out[87]:

		Count	Amount (Rs.)
0	Total Orders where X has been overcharged	84	-4454.9

```
In [88]: b =  {" " : ['Total Orders where X has been undercharged']}
```

```
In [89]: df3 = pd.DataFrame(data=b)
```

```
In [90]: df3
```

Out[90]:

0	Total Orders where X has been undercharged

```
In [91]: df3['Count'] = (df['Difference Between Expected Charges and Billed Charges (Rs.)']>0).sum()
```

```
In [92]: df3
```

Out[92]:

		Count
0	Total Orders where X has been undercharged	23

```
In [93]: df3['Amount (Rs.)'] = df['under'].sum()
```

```
In [94]: df3
```

Out[94]:

		Count	Amount (Rs.)
0	Total Orders where X has been undercharged	23	602.3

```
In [95]: df4 = pd.concat([df1,df2,df3],ignore_index = True)
```

```
In [96]: df4
```

Out[96]:

		Count	Amount (Rs.)
0	Total orders where X has been correctly charged	17	9795.6
1	Total Orders where X has been overcharged	84	-4454.9
2	Total Orders where X has been undercharged	23	602.3

```
In [97]: dff = df.drop(['over','under'],axis = 1)
```

```
In [98]: dff
```

Out[98]:

	Order ID	AWB Code	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.)
0	2001806210	1091117221940	0.22	0.5	2.92	3.0	b	b	33.0	174.5	-141.5
1	2001806226	1091117222065	0.48	0.5	0.68	1.0	d	d	45.4	90.2	-44.8
2	2001806229	1091117222080	0.50	0.5	0.71	1.0	d	d	45.4	90.2	-44.8
3	2001806232	1091117222124	1.30	1.5	1.30	1.5	d	d	135.0	135.0	0.0
4	2001806233	1091117222135	0.24	0.5	0.78	1.0	b	b	33.0	61.3	-28.3
...	...	...	...	...	...	...	...	...	...	...	...
119	2001821995	1091121183730	0.48	0.5	0.50	0.5	b	d	33.0	45.4	-12.4
120	2001822466	1091121305541	1.38	1.5	1.10	1.5	b	d	89.6	135.0	-45.4
121	2001823564	1091121666133	0.67	1.0	0.70	1.0	d	d	176.3	172.8	3.5
122	2001825261	1091121981575	1.56	2.0	1.60	2.0	d	d	355.5	345.0	10.5
123	2001827036	1091122418320	2.18	2.5	1.60	2.0	e	b	278.6	117.9	160.7

124 rows × 11 columns

```
In [99]: writer = pd.ExcelWriter("summary table.xlsx")
```

```
In [100… df4.to_excel(writer,sheet_name="summary")
```

```
In [101… writer._save()
```

```
In [ ]:
```