In [1]:

```
import numpy as np
import pandas as pd
import datetime
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px

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

In [2]:

```
#Loading the data from csv files.
df=pd.read_csv('Daily_Order_Data_July_2022_Consolidated',parse_dates=['Date'])
df.head()
```

Out[2]:

Date	RiderCustomerId	Rider Name	LLEmpId	Order Type	Order Allocated	Delivered Orders	Revenue	Custom
2022-07-01	12339	Chandrashekhar	12339	Trip Order	11.0	3.0	415	Porter 2
2022-07-01	13617	Deepak D	13617	Trip Order	25.0	2.0	294	Porter 2
2022-07-01	11874	Girish S	11874	Trip Order	18.0	9.0	1263	Porter 2
2022-07-01	12008	Golam Ahmed	12008	Trip Order	16.0	6.0	776	Porter 2
2022-07-01	14160	Harish R	14160	Trip Order	16.0	3.0	500	Porter 2

In [3]:

```
df.shape
```

Out[3]:

```
(46420, 14)
```

In [4]:

```
missing_values_count=df.isnull().sum()
missing_values_count[0:10]
```

Out[4]:

```
Date                   0
RiderCustomerId     6634
Rider Name            42
LLEmpId             5170
Order Type             2
Order Allocated        3
Delivered Orders       3
Revenue                2
Customer               0
City                   0
dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

```
(46420, 14)
```

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46420 entries, 0 to 46419
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              46420 non-null  datetime64[ns]
 1   RiderCustomerId   39786 non-null  object
 2   Rider Name        46378 non-null  object
 3   LLEmpId           41250 non-null  object
 4   Order Type        46418 non-null  object
 5   Order Allocated   46417 non-null  float64
 6   Delivered Orders  46417 non-null  float64
 7   Revenue           46418 non-null  object
 8   Customer          46420 non-null  object
 9   City              46420 non-null  object
 10  Vehicle type      46363 non-null  object
 11  Hub Name          46136 non-null  object
 12  Area              24138 non-null  object
 13  Pincode           8203 non-null   object
dtypes: datetime64[ns](1), float64(2), object(11)
memory usage: 5.0+ MB
```

In [7]:

```
df['Revenue'] = pd.to_numeric(df['Revenue'],errors='coerce')
```

In [8]:

```
df.dtypes
```

Out[8]:

```
Date                    datetime64[ns]
RiderCustomerId                 object
Rider Name                      object
LLEmpId                         object
Order Type                      object
Order Allocated                float64
Delivered Orders               float64
Revenue                        float64
Customer                        object
City                            object
Vehicle type                    object
Hub Name                        object
Area                            object
Pincode                         object
dtype: object
```

In [9]:

```
df1=df.copy()
```

In [10]:

```
df1.shape
```

Out[10]:

```
(46420, 14)
```

In [11]:

```
df1['Vehicle type'].unique()
```

Out[11]:

```
array(['EV', 'ICE', 'VAN', 'Van', nan], dtype=object)
```

In [12]:

```
df1=df1.drop(['LLEmpId','RiderCustomerId','Pincode','Area'],axis='columns')
```

In [13]:

```
df1.shape
```

Out[13]:

```
(46420, 10)
```

In [14]:

```
# bool_series = pd.isnull(df1["Revenue"])
# df1[bool_series]
```

In [15]:

```
df1.isnull().sum()
```

Out[15]:

```
Date                  0
Rider Name           42
Order Type            2
Order Allocated       3
Delivered Orders      3
Revenue            1105
Customer              0
City                  0
Vehicle type         57
Hub Name            284
dtype: int64
```

In [16]:

```
df1.sample(10)
```

Out[16]:

	Date	Rider Name	Order Type	Order Allocated	Delivered Orders	Revenue	Customer	City
11908	2022-07-16	Dhanush RoopR	Normal	47.0	42.0	714.0	Flipkart	Bangalore
36167	2022-07-26	Muneeb Muneeb	Normal	57.0	48.0	792.0	Flipkart	Kerala
12042	2022-07-16	Chakali .Chennakeshavulu	Normal	28.0	24.0	1128.0	BigBasket	Bangalore
44374	2022-07-28	VAMSHI KRISHNA	Normal	20.0	20.0	1040.0	BigBasket	Hyderabad
31605	2022-07-23	radha Krishna	Normal	19.0	19.0	912.0	BigBasket	Chennai
3075	2022-07-04	NITISH NAMASUTRA	Normal	18.0	18.0	756.0	SFX Food	Bangalore
8415	2022-07-12	Shivu M	Trip Order	3.0	3.0	971.0	Porter	Bangalore
9145	2022-07-13	Girish S	Trip Order	6.0	6.0	582.0	Porter 2W	Bangalore
12682	2022-07-17	SameerAhmed	Normal	117.0	53.0	1166.0	Flipkart	Bangalore
25812	2022-07-07	Karthi	Trip Order	5.0	5.0	602.0	Porter 2W	Chennai

In [17]:

```
# bool_series = pd.isnull(df1["Vehicle type"])
# df1[bool_series]
```

In [18]:

```
# df1[df1['Revenue']==0]
```

- Some of Riders are getting Order Allocated but they have had not delivery orders.
- Some of Riders are getting orders but they are keeping themselves.

In [19]:

```
df1.loc[2385, 'Revenue'] = 900
```

In [20]:

```
df1.loc[12024, 'Revenue'] = 900
```

In [21]:

```
#Number of 0 values
for column_name in df1.columns:
    column = df1[column_name]
    # Get the count of Zeros in column
    count = (column == 0).sum()
    print('Count of zeros in column ', column_name, ' is : ', count)
```

```
Count of zeros in column  Date  is :  0
Count of zeros in column  Rider Name  is :  0
Count of zeros in column  Order Type  is :  0
Count of zeros in column  Order Allocated  is :  310
Count of zeros in column  Delivered Orders  is :  520
Count of zeros in column  Revenue  is :  470
Count of zeros in column  Customer   is :  0
Count of zeros in column  City  is :  0
Count of zeros in column  Vehicle type  is :  0
Count of zeros in column  Hub Name  is :  0
```

In [22]:

```
df1['Customer '].unique()
```

Out[22]:

```
array(['Porter 2W', 'Flipkart', 'Flipkart_ROK', 'TOW', 'Amazon_Fresh',
       'Delhivery', 'Masthysa Dharshini', 'Lilac', 'Mapple Tree',
       'Flipkart Grocery', 'SFX E-com', 'BigBasket', 'SFX Food', 'BB NOW',
       'Grofers', 'Porter', 'Flipkart ROTN', 'Xpressbees', 'Bluedart',
       'Amazon'], dtype=object)
```

In [23]:

```
df1['Vehicle type'].unique()
```

Out[23]:

```
array(['EV', 'ICE', 'VAN', 'Van', nan], dtype=object)
```

In [24]:

```
df1['Vehicle type']=df1['Vehicle type'].str.replace("Van","VAN")
```

In [25]:

```
df1.sample(10)
```

Out[25]:

	Date	Rider Name	Order Type	Order Allocated	Delivered Orders	Revenue	Customer	City	Ve
2461	2022-07-04	Arun kumar R	Normal	87.0	71.0	1101.0	Flipkart_ROK	Bangalore	
25750	2022-07-06	janagiraman	Normal	15.0	15.0	720.0	BigBasket	Chennai	
10393	2022-07-14	Arun NijaguniBilagi	Normal	77.0	71.0	1562.0	Flipkart	Bangalore	
12588	2022-07-17	GajendraG	Normal	18.0	15.0	255.0	Flipkart	Bangalore	
40234	2022-07-05	PANDARI MUDHIRAJ	Normal	30.0	21.0	672.0	Amazon	Hyderabad	
7368	2022-07-10	Keerthi S	Normal	33.0	27.0	459.0	Flipkart	Bangalore	
5148	2022-07-07	RanjithA	Normal	57.0	51.0	867.0	Flipkart	Bangalore	
5567	2022-07-08	Rudra Gowda	Normal	41.0	32.0	544.0	Flipkart	Bangalore	
10180	2022-07-14	R K Imtiyaz Ahmed	Normal	35.0	16.0	272.0	Flipkart	Bangalore	
20805	2022-07-28	Shrinivas Naik	Normal	85.0	77.0	1309.0	Flipkart	Bangalore	

In [26]:

```
# t=pd.isnull(df1['Rider Name'])
# df1[t]
```

In [27]:

```
# t=pd.isnull(df1['Order Type'])
# # df1[t]
```

In [28]:

```
df1.loc[45881, 'Order Type'] = 'Normal'
```

In [29]:

```
df1.loc[45907, 'Order Type'] = 'Normal'
```

In [30]:

```
df1=df1.dropna(subset=['Rider Name','Order Allocated','Revenue','Vehicle type'],axis=0)
```

In [31]:

```
df1.isnull().sum()
```

Out[31]:

```
Date                  0
Rider Name            0
Order Type            0
Order Allocated       0
Delivered Orders      0
Revenue               0
Customer              0
City                  0
Vehicle type          0
Hub Name            245
dtype: int64
```

In [32]:

```
# df1[df1['Rider Name']=='Do Not Delete'].head()
```

In [33]:

```
# df1[df1['Rider Name']=='donot delete'].head()
```

In [34]:

```
# df1[df1['Rider Name']=='DON NOT DELETE']
```

In [35]:

```
df1=df1[~((df1['Rider Name']=='Do Not Delete')|(df1['Rider Name']=='donot delete')|(df1['Ri
```

In [36]:

```
df1.shape
```

Out[36]:

```
(45187, 10)
```

In [37]:

```
df1["Order Allocated"].isnull().sum()
```

Out[37]:

```
0
```

In [38]:

```
# df1[(df1['Order Allocated']==0)]
```

In [39]:

```
df1=df1[~(df1['Order Allocated']==0)]
```

In [40]:

```
df1.shape
```

Out[40]:

```
(44963, 10)
```

In [41]:

```
df1.isnull().sum()
```

Out[41]:

```
Date                 0
Rider Name           0
Order Type           0
Order Allocated      0
Delivered Orders     0
Revenue              0
Customer             0
City                 0
Vehicle type         0
Hub Name           241
dtype: int64
```

In [42]:

```
df1=df1[~(df1['Revenue']==0)]
```

In [43]:

```
# df1[(df1['Revenue']==800)]
```

In [44]:

```
# df1.hist(figsize=(15,10))
```

In [45]:

```
df1.shape
```

Out[45]:

```
(44750, 10)
```

In [46]:

```
df1.describe()
```

Out[46]:

	Order Allocated	Delivered Orders	Revenue
count	44750.000000	44750.000000	44750.000000
mean	39.509698	33.279642	823.509497
std	29.701039	24.811964	436.343591
min	1.000000	0.000000	15.000000
25%	17.000000	16.000000	520.000000
50%	30.000000	26.000000	768.000000
75%	58.000000	47.000000	1040.000000
max	270.000000	364.000000	6188.000000

In [47]:

```
list(df1)
```

Out[47]:

```
['Date',
 'Rider Name',
 'Order Type',
 'Order Allocated',
 'Delivered Orders',
 'Revenue',
 'Customer ',
 'City',
 'Vehicle type',
 'Hub Name']
```

In [48]:

```
df1.rename(    columns=({ 'Customer ': 'Client'}),
    inplace=True)
df1.head()
```

Out[48]:

	Date	Rider Name	Order Type	Order Allocated	Delivered Orders	Revenue	Client	City	Vehicle type	
0	2022-07-01	Chandrashekhar	Trip Order	11.0	3.0	415.0	Porter 2W	Bangalore	EV	
1	2022-07-01	Deepak D	Trip Order	25.0	2.0	294.0	Porter 2W	Bangalore	EV	
2	2022-07-01	Girish S	Trip Order	18.0	9.0	1263.0	Porter 2W	Bangalore	EV	BAI
3	2022-07-01	Golam Ahmed	Trip Order	16.0	6.0	776.0	Porter 2W	Bangalore	EV	IN
4	2022-07-01	Harish R	Trip Order	16.0	3.0	500.0	Porter 2W	Bangalore	EV	

In [49]:

```
df1.sample(10)
```

Out[49]:

	Date	Rider Name	Order Type	Order Allocated	Delivered Orders	Revenue	Client	City	Vehicle type
9154	2022-07-13	N Heera Sunar	Trip Order	2.0	2.0	720.0	Porter 2W	Bangalore	EV
44112	2022-07-26	DINESH	Normal	70.0	70.0	1805.0	Amazon	Hyderabad	ICE
22961	2022-07-30	Naveen Kumar S	Normal	15.0	15.0	705.0	BigBasket	Bangalore	EV
1121	2022-07-02	Nandish N	Normal	61.0	48.0	816.0	Flipkart	Bangalore	EV
40643	2022-07-07	ATIF	Normal	12.0	11.0	652.0	Amazon	Hyderabad	EV
35441	2022-07-01	Basil Eldhose	Normal	45.0	36.0	594.0	Flipkart	Kerala	ICE
44465	2022-07-28	THARUN	Normal	18.0	18.0	810.0	BB NOW	Hyderabad	EV
40577	2022-07-07	B AMAR	Normal	21.0	21.0	1092.0	BigBasket	Hyderabad	EV
15114	2022-07-20	ABDUL REHMAN	Normal	2.0	2.0	84.0	SFX Food	Bangalore	EV
26219	2022-07-08	Thufail Ahmed	Normal	4.0	4.0	550.0	BB NOW	Chennai	EV

In [54]:

```
print('Mean is',np.round(df1['Revenue'].mean()),
'\n Median is ',df1['Revenue'].median(),
'\n Mode is',df1['Revenue'].mode()[0])
```

```
Mean is 824.0
 Median is  768.0
 Mode is 550.0
```

In [55]:

```
print('Mean is',np.round(df1['Order Allocated'].mean()),
      '\n Median is ',df1['Order Allocated'].median(),
      '\n Mode is ',df1['Order Allocated'].mode()[0])
```

```
Mean is 40.0
 Median is  30.0
 Mode is  20.0
```

In [56]:

```
print('Mean is',np.round(df1['Delivered Orders'].mean()),
'\n Median is ',df1['Delivered Orders'].median(),
'\n Mode is ',df1['Delivered Orders'].mode()[0])
```

```
Mean is 33.0
 Median is  26.0
 Mode is  20.0
```

In [57]:

```
df1['Delivered Orders'].std()
```

Out[57]:

```
24.811964219778044
```

In [58]:

```
list(df1)
```

Out[58]:

```
['Date',
 'Rider Name',
 'Order Type',
 'Order Allocated',
 'Delivered Orders',
 'Revenue',
 'Client',
 'City',
 'Vehicle type',
 'Hub Name']
```

In [59]:

```
df1.Client.unique()
```

Out[59]:

```
array(['Porter 2W', 'Flipkart', 'Flipkart_ROK', 'TOW', 'Amazon_Fresh',
       'Delhivery', 'Lilac', 'Mapple Tree', 'Flipkart Grocery',
       'SFX E-com', 'BigBasket', 'SFX Food', 'BB NOW', 'Grofers',
       'Porter', 'Flipkart ROTN', 'Xpressbees', 'Bluedart', 'Amazon'],
      dtype=object)
```

In [60]:

```
df_city=np.round(df1.groupby('City')['Revenue'].agg(['count','mean','median','std']))
df_city
```

Out[60]:

	count	mean	median	std
City				
Bangalore	23499	842.0	799.0	463.0
Chennai	11498	817.0	776.0	381.0
Delhi	1891	574.0	525.0	204.0
Gujarat	26	353.0	342.0	141.0
Hyderabad	5643	923.0	832.0	469.0
Kerala	874	637.0	611.0	243.0
Madurai	581	680.0	624.0	336.0
Mysore	711	575.0	527.0	363.0
maharashtra	27	537.0	517.0	69.0

In [61]:

```
df_client=np.round(df1.groupby('Client')['Revenue'].agg(['count','mean','median','std']))
df_client
```

Out[61]:

	count	mean	median	std
Client				
Amazon	1841	1003.0	850.0	520.0
Amazon_Fresh	243	1025.0	957.0	537.0
BB NOW	1812	690.0	600.0	256.0
BigBasket	10972	927.0	893.0	396.0
Bluedart	5	814.0	770.0	211.0
Delhivery	48	878.0	908.0	229.0
Flipkart	21234	808.0	731.0	461.0
Flipkart Grocery	374	822.0	810.0	271.0
Flipkart ROTN	1450	602.0	561.0	287.0
Flipkart_ROK	1119	784.0	744.0	369.0
Grofers	35	1070.0	1100.0	458.0
Lilac	22	900.0	900.0	0.0
Mapple Tree	87	1800.0	1800.0	0.0
Porter	1166	948.0	880.0	427.0
Porter 2W	721	575.0	585.0	265.0
SFX E-com	1246	857.0	817.0	388.0
SFX Food	1945	484.0	462.0	288.0
TOW	369	830.0	825.0	28.0
Xpressbees	61	334.0	238.0	216.0

In [62]:

```
df_vehicle=np.round(df1.groupby('Vehicle type')['Revenue'].agg(['count','mean','median','st
df_vehicle
```

Out[62]:

	count	mean	median	std
Vehicle type				
EV	33029	802.0	765.0	402.0
ICE	9890	834.0	731.0	492.0
VAN	1831	1156.0	1045.0	558.0

In [63]:

```
df_hubname=np.round(df1.groupby('Hub Name')['Revenue'].agg(['count','mean','median','std'])
df_hubname
```

Out[63]:

```
                count   mean   median    std
      Hub Name
    Pallavaram     14   550.0   550.0    0.0
       3w west     28   825.0   825.0    0.0
          AJAY      1   825.0   825.0    NaN
       ATTAPUR    115   590.0   594.0  229.0
 AdimalyHub_ADL      8   386.0   289.0  251.0
           ...    ...     ...     ...    ...
   soukya road      1   564.0   564.0    NaN
      tambaram      1  1305.0  1305.0    NaN
     vanagaram   1024   864.0   840.0  317.0
     vedapatti      9   416.0   500.0  277.0
        vepery     14   276.0   238.0   85.0
```

483 rows × 4 columns

In [64]:

```
df_date=np.round(df1.groupby('Date')['Revenue'].agg(['count','mean','median','std']))
df_date
```

Out[64]:

```
            count   mean  median    std
      Date
2022-07-01   1435  762.0   697.0  417.0
2022-07-02   1452  772.0   710.0  421.0
2022-07-03   1308  809.0   748.0  436.0
2022-07-04   1438  859.0   798.0  457.0
2022-07-05   1424  838.0   782.0  457.0
2022-07-06   1474  819.0   752.0  447.0
2022-07-07   1448  798.0   752.0  436.0
2022-07-08   1439  843.0   792.0  456.0
2022-07-09   1426  802.0   738.0  438.0
```

In [ ]:

Pie Chart

In [65]:

```
data = {'Revenue':'sum_revenue'}
agg_data = {'Revenue':'sum'}
data_bar = df1.groupby(['Vehicle type'], as_index=False).agg(agg_data).rename(columns=data)
```

In [66]:

```
data_bar
```

Out[66]:

	Vehicle type	sum_revenue
0	EV	26485266.0
1	ICE	8249242.0
2	VAN	2117542.0

In [67]:

```
df1.Revenue.sum()
```

Out[67]:

```
36852050.0
```

Vehicle type | Values | Percentage

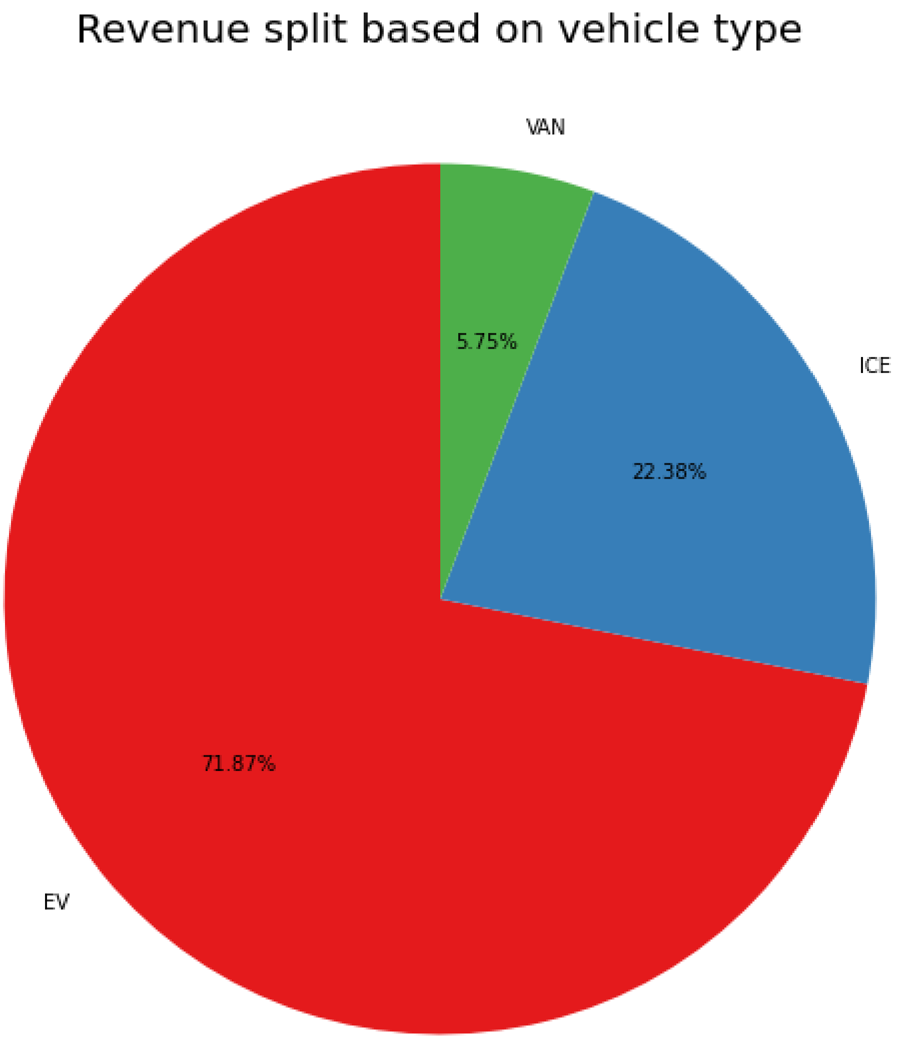
EV | 26485266 | 71.87%

ICE | 8249242 | 22.38%

VAN | 2117542 | 5.75%

In [68]:

```
plt.figure(figsize=(10,10))
palette_color = sns.color_palette('Set1')
plt.pie(data_bar.sum_revenue, colors=palette_color,
        labels=data_bar['Vehicle type'], startangle=90,autopct='%.2f%%')
plt.title('Revenue split based on vehicle type',fontsize=20)
plt.show()
```



In [69]:

```
data = {'Revenue':'sum_revenue'}
agg_data = {'Revenue':'sum'}
data_bar = df1.groupby(['City'], as_index=False).agg(agg_data).rename(columns=data).sort_in
```

In [70]:

```
data_bar
```

Out[70]:

	City	sum_revenue
0	Bangalore	19786304.0
1	Chennai	9388193.0
2	Delhi	1084743.0
3	Gujarat	9177.0
4	Hyderabad	5208231.0
5	Kerala	556962.0
6	Madurai	394909.5
7	Mysore	409039.5
8	maharashtra	14491.0

In [71]:

```
# plt.figure(figsize=(10,10))
# palette_color = sns.color_palette('Greens')
# plt.pie(data_bar.sum_revenue, colors=palette_color,labels=data_bar['City'], rotatelabels
#         autopct='%.1f%%',counterclock=False)
# plt.title('Pie chart for City with revenue in SUM',fontsize=20)
# plt.show()
```

In [72]:

```
data = {'Revenue':'sum_revenue'}
agg_data = {'Revenue':'sum'}
data_bar = df1.groupby(['Client'], as_index=False).agg(agg_data).rename(columns=data)
```

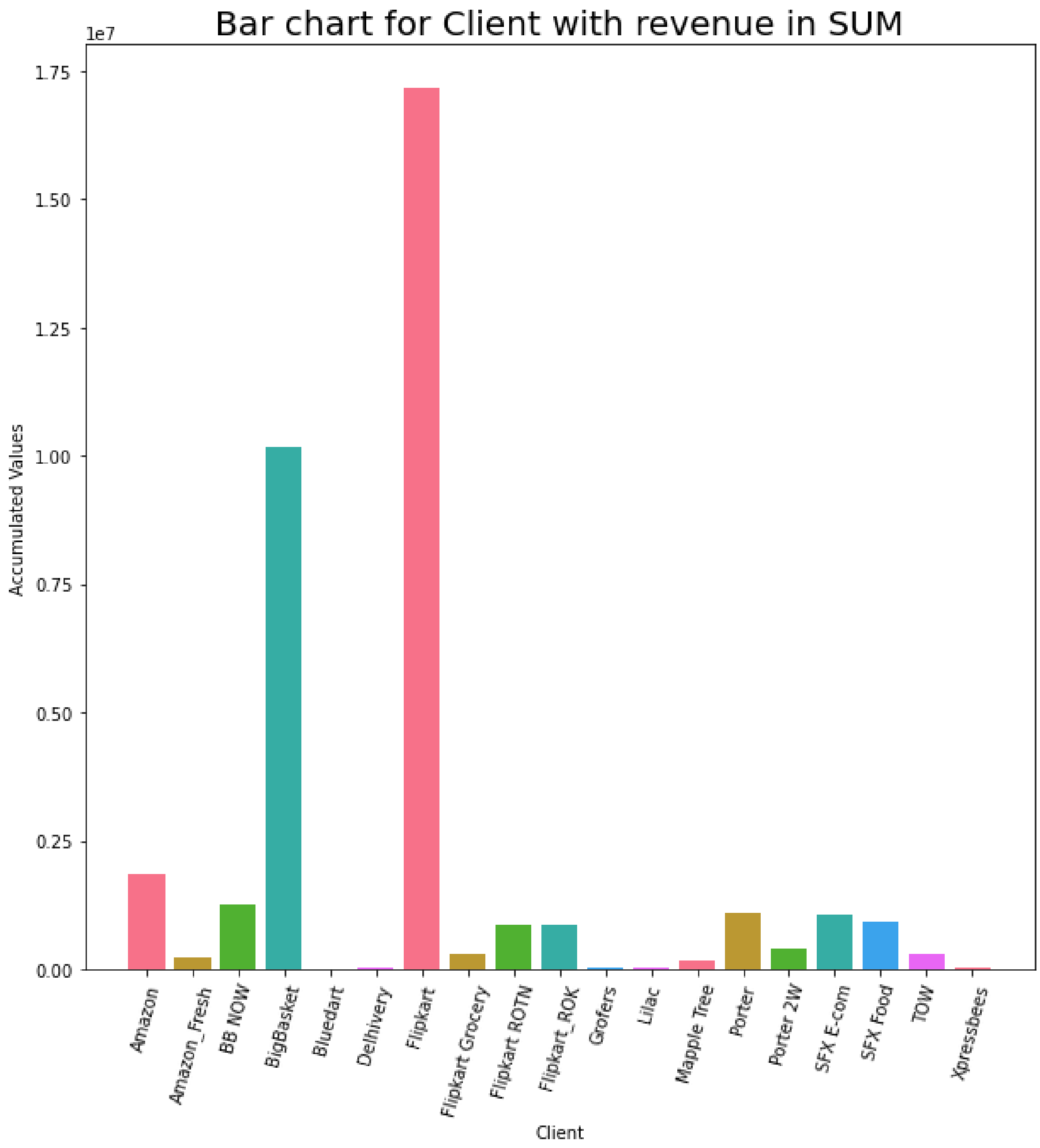
In [73]:

```
data_bar
```

```
 5        Delhivery       42150.0
 6         Flipkart    17156793.5
 7  Flipkart Grocery     307260.0
 8     Flipkart ROTN     872473.5
 9      Flipkart_ROK     877354.0
10          Grofers       37455.0
11            Lilac       19800.0
12      Mapple Tree      156600.0
13           Porter     1105873.0
14        Porter 2W      414335.0
15        SFX E-com     1067476.0
16         SFX Food      941807.0
17              TOW      306185.0
```

In [74]:

```
plt.figure(figsize=(10,10))
index = np.arange(len(data_bar['Client']))
palette_color = sns.color_palette('husl')

plt.bar(index, data_bar.sum_revenue,color=palette_color)
plt.xlabel('Client', fontsize=10)
plt.ylabel('Accumulated Values', fontsize=10)
plt.xticks(index, data_bar['Client'], fontsize=10, rotation=75)
plt.title('Bar chart for Client with revenue in SUM',fontsize=20)
plt.show()
```



In [75]:

```
np.round(df1.groupby(['Vehicle type','City'])['Revenue'].agg(['count','mean','median','std'
```

Out[75]:

		count	mean	median	std
Vehicle type	**City**				
EV	**Bangalore**	17660	801.0	765.0	433.0
	Chennai	9173	849.0	816.0	380.0
	Delhi	940	540.0	500.0	201.0
	Hyderabad	4148	811.0	780.0	309.0
	Madurai	399	708.0	636.0	338.0
	Mysore	709	576.0	527.0	364.0
ICE	**Bangalore**	4871	909.0	850.0	510.0
	Chennai	2087	674.0	611.0	352.0
	Delhi	582	542.0	460.0	177.0

In [76]:

```
np.round(df1.groupby(['City','Client'])['Revenue'].agg(['count','mean','median','std']))
```

Out[76]:

		count	mean	median	std
City	**Client**				
Bangalore	**Amazon_Fresh**	243	1025.0	957.0	537.0
	BB NOW	197	754.0	760.0	228.0
	BigBasket	3765	914.0	893.0	385.0
	Delhivery	48	878.0	908.0	229.0
	Flipkart	14384	861.0	799.0	494.0
	Flipkart Grocery	148	811.0	855.0	140.0
	Flipkart_ROK	1119	784.0	744.0	369.0
	Grofers	35	1070.0	1100.0	458.0
	Lilac	22	900.0	900.0	0.0

In [77]:

```
a = np.round(df1.groupby(['City','Client'])['Revenue'].agg(['count','mean','median','std'])
```

In [78]:

```
a.head()
```

Out[78]:

	City	Client	count	mean	median	std
0	Bangalore	Amazon_Fresh	243	1025.0	957.0	537.0
1	Bangalore	BB NOW	197	754.0	760.0	228.0
2	Bangalore	BigBasket	3765	914.0	893.0	385.0
3	Bangalore	Delhivery	48	878.0	908.0	229.0
4	Bangalore	Flipkart	14384	861.0	799.0	494.0

In [79]:

```
city_df = a[a['City'] == 'Bangalore']
city_df.sort_values('count', ascending=False)
```

Out[79]:

	City	Client	count	mean	median	std
4	Bangalore	Flipkart	14384	861.0	799.0	494.0
2	Bangalore	BigBasket	3765	914.0	893.0	385.0
13	Bangalore	SFX Food	1757	509.0	504.0	286.0
6	Bangalore	Flipkart_ROK	1119	784.0	744.0	369.0
11	Bangalore	Porter 2W	661	594.0	601.0	261.0
10	Bangalore	Porter	564	1157.0	1179.0	452.0
14	Bangalore	TOW	369	830.0	825.0	28.0
0	Bangalore	Amazon_Fresh	243	1025.0	957.0	537.0
1	Bangalore	BB NOW	197	754.0	760.0	228.0
5	Bangalore	Flipkart Grocery	148	811.0	855.0	140.0
12	Bangalore	SFX E-com	100	671.0	656.0	263.0
9	Bangalore	Mapple Tree	87	1800.0	1800.0	0.0
3	Bangalore	Delhivery	48	878.0	908.0	229.0
7	Bangalore	Grofers	35	1070.0	1100.0	458.0
8	Bangalore	Lilac	22	900.0	900.0	0.0

In [80]:

```
city_df['client%'] = city_df['count']*100/city_df['count'].sum()
```

In [81]:

```
city_df['Client_new'] = np.where(city_df['client%'] < 2, 'Others', city_df['Client'])
```

In [82]:

```
b = city_df.groupby('Client_new')['count'].mean().reset_index()
```

In [83]:

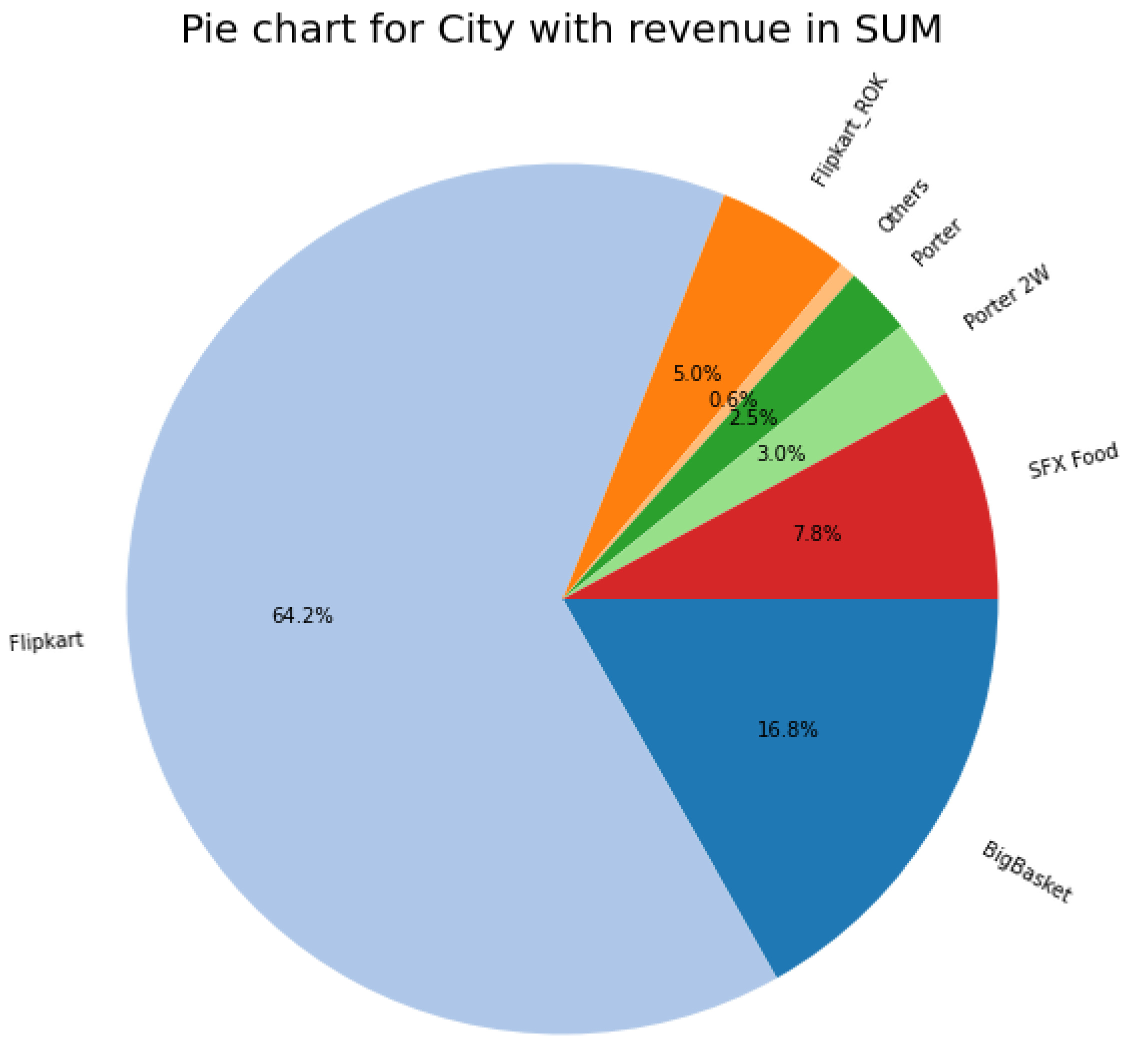
```
b
```

Out[83]:

	Client_new	count
0	BigBasket	3765.000000
1	Flipkart	14384.000000
2	Flipkart_ROK	1119.000000
3	Others	138.777778
4	Porter	564.000000
5	Porter 2W	661.000000
6	SFX Food	1757.000000

In [84]:

```
plt.figure(figsize=(10,10))
palette_color = sns.color_palette('tab20')
plt.pie(b['count'], colors=palette_color,labels=b['Client_new'], rotatelabels =True,
        autopct='%.1f%%',counterclock=False)
plt.title('Pie chart for City with revenue in SUM',fontsize=20)
plt.show()
```



In [89]:

```
a_1=df1.groupby(['Vehicle type','City'])['Revenue'].agg(['sum']).reset_index()
vehicle_df = a_1[a_1['Vehicle type'] == 'EV']
vehicle_df.sort_values('sum', ascending=False)
vehicle_df['City%'] = vehicle_df['sum']*100/vehicle_df['sum'].sum()
vehicle_df.head()
```

Out[89]:

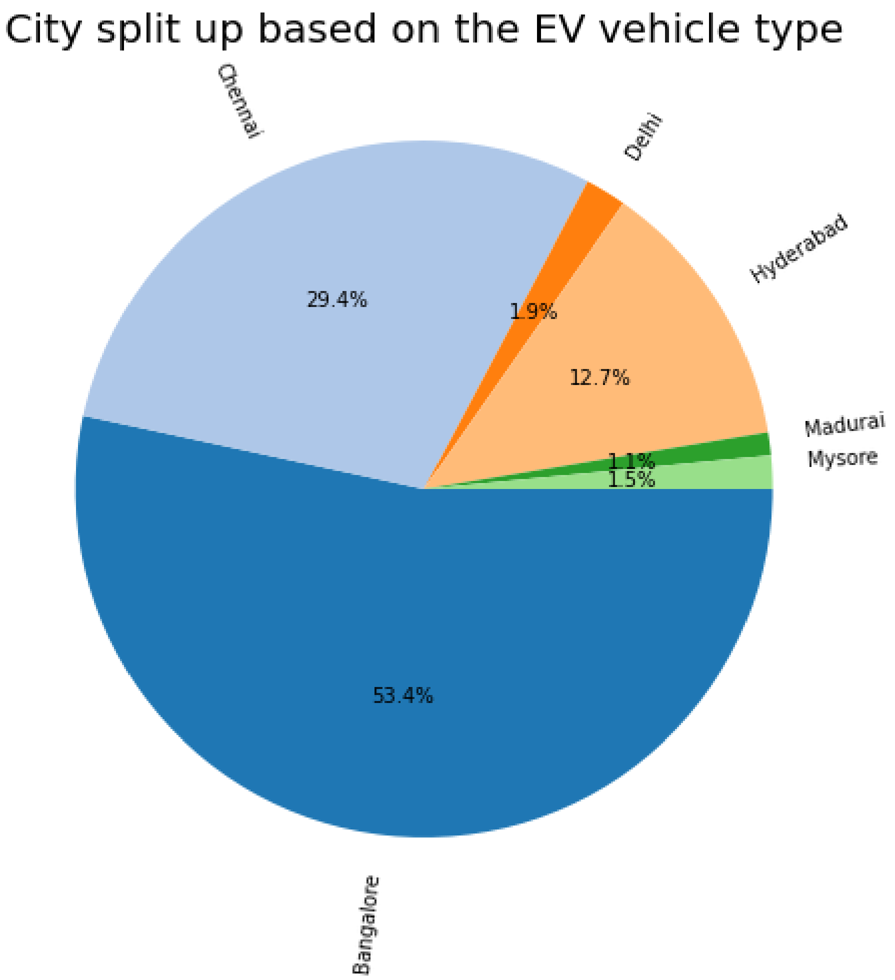
	Vehicle type	City	sum	City%
0	EV	Bangalore	14138076.0	53.380910
1	EV	Chennai	7786871.0	29.400766
2	EV	Delhi	507409.0	1.915816
3	EV	Hyderabad	3362216.0	12.694666
4	EV	Madurai	282383.5	1.066191

In [90]:

```
m=vehicle_df.groupby('City')['sum'].sum().reset_index()
```

In [91]:

```
plt.figure(figsize=(10,8))
palette_color = sns.color_palette('tab20')
plt.pie(m['sum'], colors=palette_color,labels=m['City'], rotatelabels =True,
        autopct='%.1f%%',counterclock=False)
plt.title(' City split up based on the EV vehicle type ',fontsize=20)
plt.show()
```



In [92]:

```
# df1.groupby(['Vehicle type','City'])['Revenue'].mean().sort_values(ascending=False).unsta
#
# plt.title('Bar chart for Vehicle type with revenue in MEAN',fontsize=20)
# plt.show()
```

In [93]:

```
# df1.groupby(['Vehicle type'])['Revenue'].sum().plot.bar(
# plt.title('Bar chart for Vehicle type with revenue in MEAN',fontsize=20)
# plt.show()
```

In [146]:

```
df_city_client_vehicle=df1.groupby(['Vehicle type','City','Client'])['Revenue'].agg(['mean'
```

In [147]:

```
df_city_client_vehicle.head()
```

Out[147]:

	Vehicle type	City	Client	mean
0	EV	Bangalore	Amazon_Fresh	1045.081545
1	EV	Bangalore	BB NOW	753.705584
2	EV	Bangalore	BigBasket	885.119457
3	EV	Bangalore	Delhivery	878.125000
4	EV	Bangalore	Flipkart	835.644442

In [148]:

```
y=df_city_client_vehicle[df_city_client_vehicle['Vehicle type']=='EV']
```

In [149]:

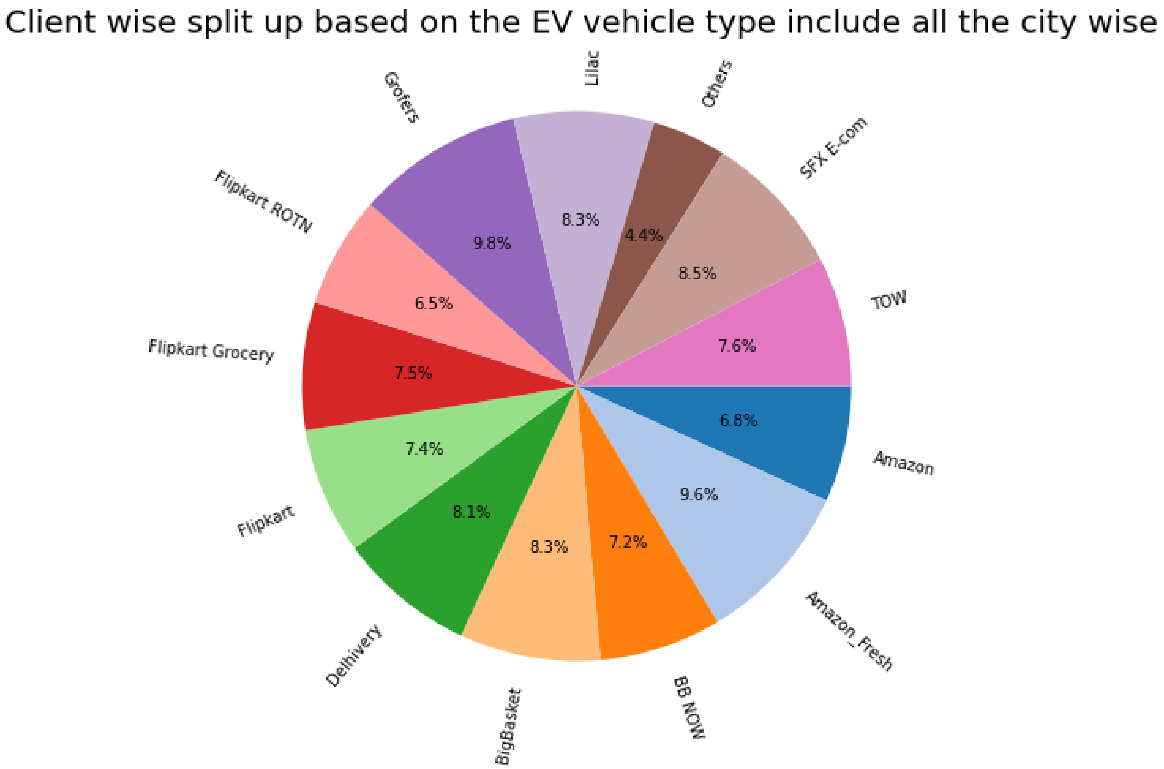
```
y.sort_values('mean', ascending=False)
y['client%'] = y['mean']*100/y['mean'].sum()
y['Client_new'] = np.where(y['client%'] <3,'Others', y['Client'])
y
```

Out[149]:

	Vehicle type	City	Client	mean	client%	Client_new
0	EV	Bangalore	Amazon_Fresh	1045.081545	4.417509	Amazon_Fresh
1	EV	Bangalore	BB NOW	753.705584	3.185877	BB NOW
2	EV	Bangalore	BigBasket	885.119457	3.741357	BigBasket
3	EV	Bangalore	Delhivery	878.125000	3.711792	Delhivery
4	EV	Bangalore	Flipkart	835.644442	3.532228	Flipkart
5	EV	Bangalore	Flipkart Grocery	810.912162	3.427686	Flipkart Grocery
6	EV	Bangalore	Flipkart_ROK	506.262357	2.139946	Others
7	EV	Bangalore	Grofers	1070.142857	4.523442	Grofers
8	EV	Bangalore	Lilac	900.000000	3.804256	Lilac
9	EV	Bangalore	Porter 2W	593.804841	2.509984	Others

In [98]:

```
y=y.groupby('Client_new')['mean'].mean().reset_index()

plt.figure(figsize=(10,8))
palette_color = sns.color_palette('tab20')
plt.pie(y['mean'], colors=palette_color,labels=y['Client_new'], rotatelabels =True,
        autopct='%.1f%%',counterclock=False)
plt.title(' Client wise split up based on the EV vehicle type include all the city wise',fo
plt.show()
```



In [99]:

```
# df1.groupby(['Vehicle type','City','Client'])['Revenue'].mean().sort_values(ascending=Tru
#                                                                                 colormap=
# plt.title('Bar chart for Vehicle type, City and Client with revenue in MEAN',fontsize=20)
# plt.show()
```

In [100]:

```
df1.groupby(['Date','Vehicle type','City','Client'])['Revenue'].agg(['count','mean','median
```

Out[100]:

				count	mean	median	std
Date	**Vehicle type**	**City**	**Client**				
2022-07-01	**EV**	**Bangalore**	**Amazon_Fresh**	7	1076.428571	1000.0	758.011402
			BB NOW	10	776.000000	840.0	343.679437
			BigBasket	99	850.272727	893.0	278.791738
			Delhivery	1	645.000000	645.0	NaN
			Flipkart	336	725.663690	680.0	388.320221
...	**...**	**...**	**...**	...	...	...	...
2022-07-31	**VAN**	**Bangalore**	**Porter**	7	1178.142857	1213.0	428.268774
			TOW	1	825.000000	825.0	NaN
		Chennai	**Porter**	3	616.333333	478.0	258.886719
		Delhi	**Porter**	5	729.800000	742.0	196.278374
		Hyderabad	**BigBasket**	6	2076.666667	1925.0	758.674282

1707 rows × 4 columns

In [112]:

```
order_type=df1.groupby(['Order Type','Vehicle type','City'])['Order Allocated'].agg(['sum']
order_type
```

Out[112]:

	Order Type	Vehicle type	City	sum
0	NJIT	EV	Chennai	446.0
1	Normal	EV	Bangalore	707448.0
2	Normal	EV	Chennai	315892.0
3	Normal	EV	Delhi	25855.0
4	Normal	EV	Hyderabad	88543.0
5	Normal	EV	Madurai	21493.0
6	Normal	EV	Mysore	29978.0
7	Normal	ICE	Bangalore	291770.0
8	Normal	ICE	Chennai	102144.0
9	Normal	ICE	Delhi	19651.0

In [113]:

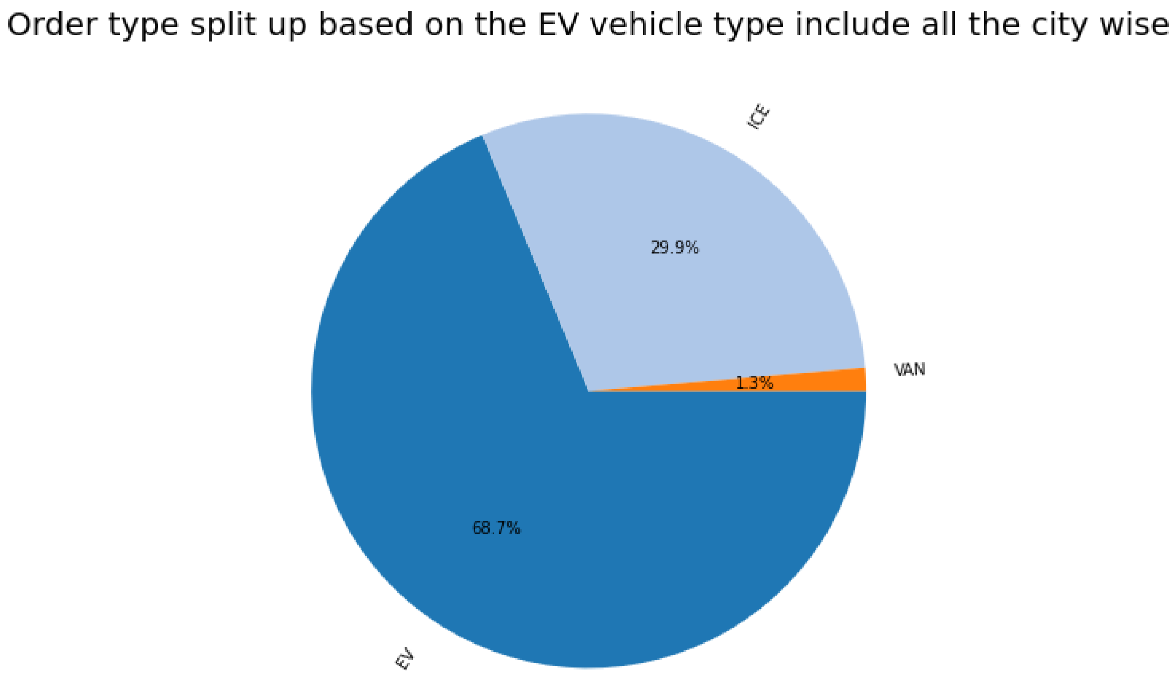
```
order_type.sort_values('sum', ascending=False)
order_type['Vehicle type%'] = order_type['sum']*100/order_type['sum'].sum()
#order_type['Vehicle type'] = np.where(order_type['Vehicle type%'] <3,'Others', order_type[
order_type
```

Out[113]:

	Order Type	Vehicle type	City	sum	Vehicle type%
0	NJIT	EV	Chennai	446.0	0.025225
1	Normal	EV	Bangalore	707448.0	40.012692
2	Normal	EV	Chennai	315892.0	17.866598
3	Normal	EV	Delhi	25855.0	1.462338
4	Normal	EV	Hyderabad	88543.0	5.007921
5	Normal	EV	Madurai	21493.0	1.215627
6	Normal	EV	Mysore	29978.0	1.695532
7	Normal	ICE	Bangalore	291770.0	16.502277
8	Normal	ICE	Chennai	102144.0	5.777183
9	Normal	ICE	Delhi	19651.0	1.111445

In [150]:

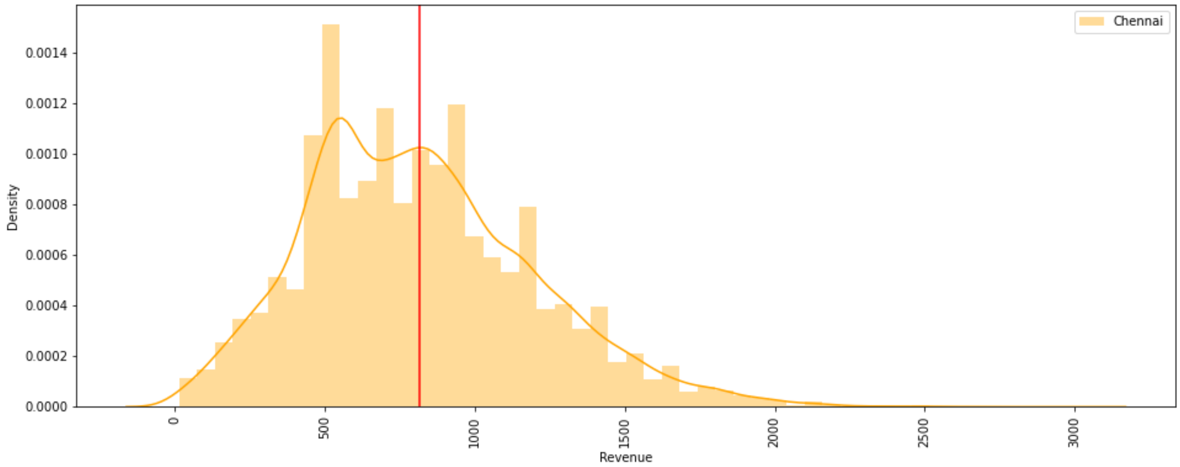
```
order_type=order_type.groupby('Vehicle type')['sum'].sum().reset_index()

plt.figure(figsize=(10,8))
palette_color = sns.color_palette('tab20')
plt.pie(order_type['sum'], colors=palette_color,labels=order_type['Vehicle type'], rotatela
        autopct='%.1f%%',counterclock=False)
plt.title('Order type split up based on the EV vehicle type include all the city wise',font
plt.show()
```



Outlier on the Revenue and City wise

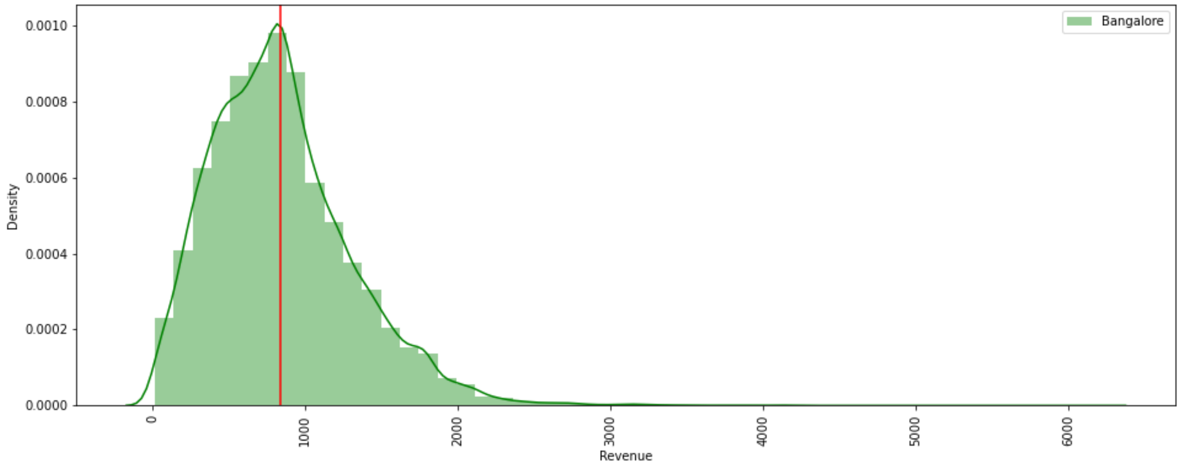
In [115]:

```
# type 3 : add avg line
fig, ax = plt.subplots(1,1,figsize=(16, 6))
sns.distplot(df1[df1['City']=='Chennai']['Revenue'],color='orange',  label='Chennai')

 # avg line
plt.axvline(df1[df1['City']=='Chennai']['Revenue'].mean(), color='red')

plt.legend()
plt.xticks(rotation='vertical')
plt.show()
```



In [116]:

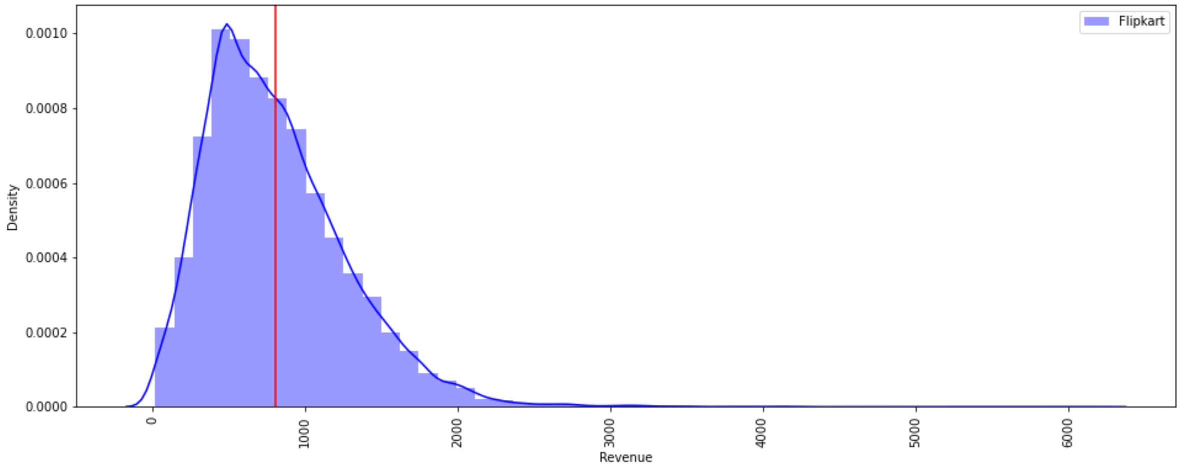
```
# type 3 : add avg line
fig, ax = plt.subplots(1,1,figsize=(16, 6))
sns.distplot(df1[df1['City']=='Bangalore']['Revenue'], color='green', label='Bangalore')

 # avg line
plt.axvline(df1[df1['City']=='Bangalore']['Revenue'].mean(), color='red')

plt.legend()
plt.xticks(rotation='vertical')
plt.show()
```



Outlier on the Revenue and Client wise

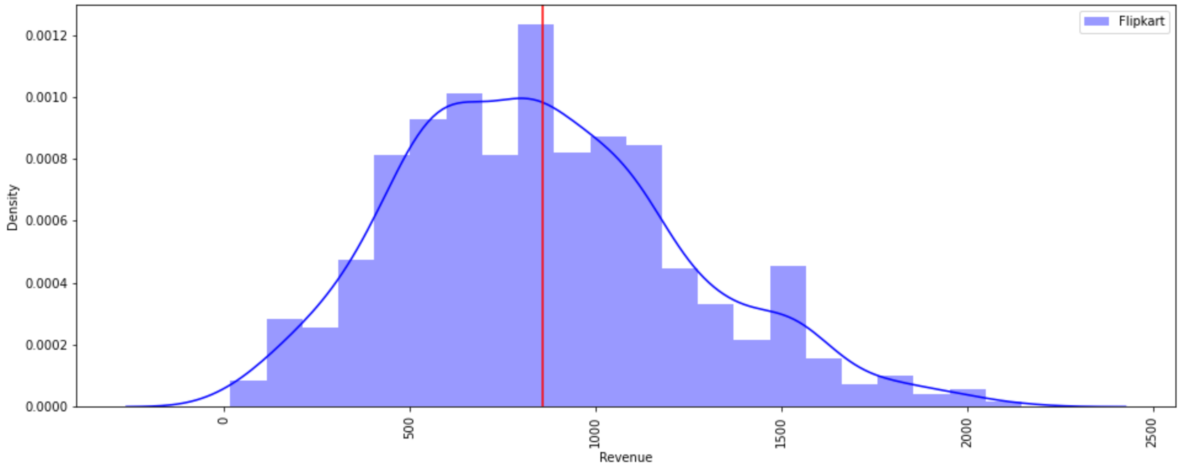
In [117]:

```
# type 3 : add avg line
fig, ax = plt.subplots(1,1,figsize=(16, 6))
sns.distplot(df1[df1['Client']=='Flipkart']['Revenue'],  norm_hist = True,color='blue', ax=

 # avg line
plt.axvline(df1[df1['Client']=='Flipkart']['Revenue'].mean(), color='red')

plt.legend()
plt.xticks(rotation='vertical')
plt.show()
```



In [118]:

```
# type 3 : add avg line
fig, ax = plt.subplots(1,1,figsize=(16, 6))
sns.distplot(df1[df1['Client']=='SFX E-com']['Revenue'],  norm_hist = True,color='blue', ax

 # avg line
plt.axvline(df1[df1['Client']=='SFX E-com']['Revenue'].mean(), color='red')

plt.legend()
plt.xticks(rotation='vertical')
plt.show()
```

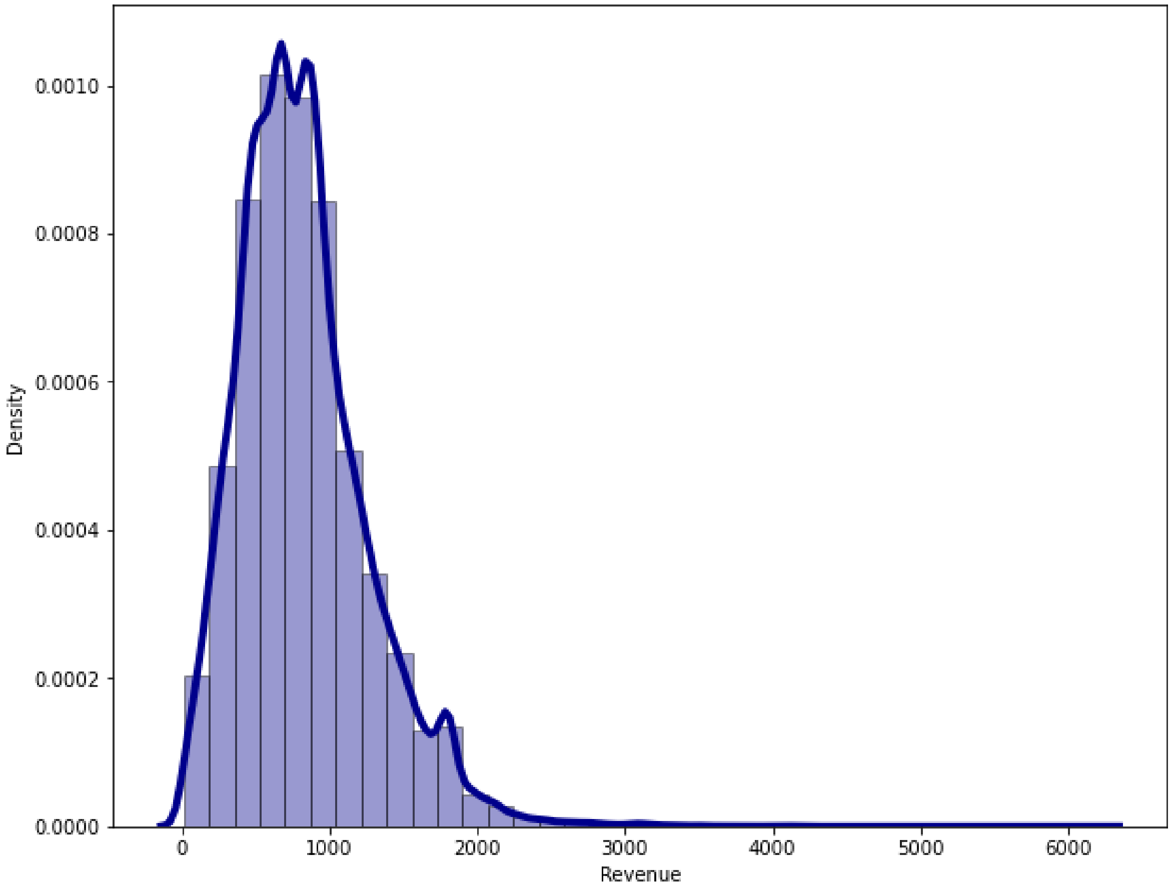


In [121]:

```
# Density Plot and Histogram of all revenue
plt.figure(figsize=(10,8))
sns.distplot(df1['Revenue'], hist=True, kde=True,
             bins=int(180/5), color = 'darkblue',
             hist_kws={'edgecolor':'black'},
             kde_kws={'linewidth': 4})
```

Out[121]:

```
<AxesSubplot:xlabel='Revenue', ylabel='Density'>
```

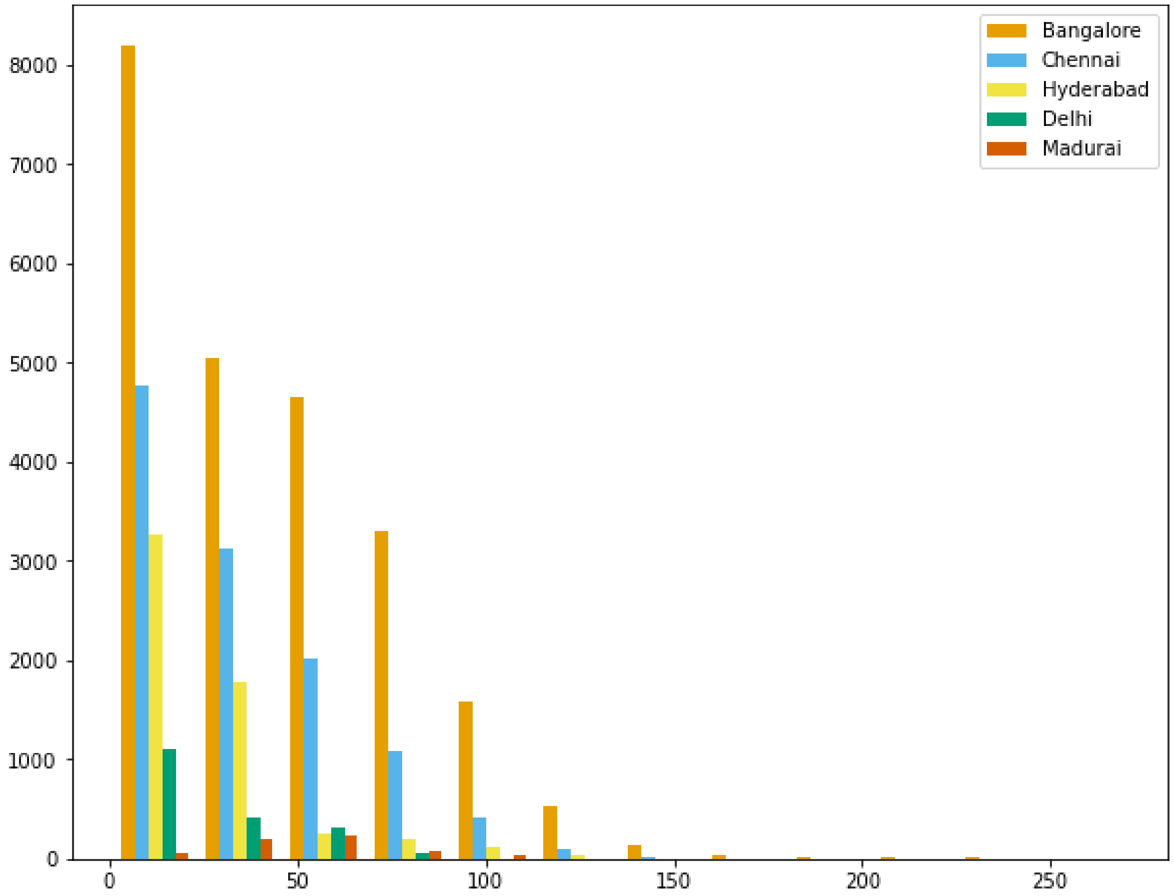


In [129]:

```
x1 = list(df1[df1['City']=='Bangalore']['Order Allocated'])
x2 = list(df1[df1['City']=='Chennai']['Order Allocated'])
x3 = list(df1[df1['City']=='Hyderabad']['Order Allocated'])
x4 = list(df1[df1['City']=='Delhi']['Order Allocated'])
x5 = list(df1[df1['City']=='Madurai']['Order Allocated'])

plt.figure(figsize=(10,8))

# Assign colors for each airline and the names
colors = ['#E69F00', '#56B4E9', '#F0E442', '#009E73', '#D55E00']
names = ['Bangalore', 'Chennai', 'Hyderabad',
         'Delhi', 'Madurai']

# Make the histogram using a list of lists
# Normalize the flights and assign colors and names
plt.hist([x1, x2, x3, x4, x5], bins = int(180/15),
         color = colors, label=names)

# Plot formatting
plt.legend()
plt.xlabel('Delay (min)')
plt.ylabel('Normalized order allocated')
plt.title('Side-by-Side Histogram with City')
```

Out[129]:

```
<matplotlib.legend.Legend at 0x1c24737e550>
```



In [134]:

```
m=df1.groupby('City')['Revenue'].agg(['mean','std']).reset_index()
```

In [135]:

```
from stat import *m
```

Out[135]:

	City	mean	std
0	Bangalore	842.006213	463.225740
1	Chennai	816.506610	381.041954
2	Delhi	573.634585	203.580361
3	Gujarat	352.961538	140.976730
4	Hyderabad	922.954280	469.231493
5	Kerala	637.256293	242.943633
6	Madurai	679.706540	336.128923
7	Mysore	575.301688	363.346453
8	maharashtra	536.703704	68.805084

In [ ]:

```

```