In [4]:
```
import pandas as pd
import numpy as np
df=pd.read_csv(r"C:\Users\evang\Downloads\Range-Queries-Aggregates.csv")
df
```

Out[4]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg
0	0	1.159191e+06	1.894756e+06	5225.375665	2981.728431	96046.0	34927.0	1111.618901
1	1	1.159293e+06	1.898922e+06	3499.176007	6879.352245	152668.0	54847.0	1192.855949
2	3	1.160321e+06	1.903776e+06	6495.796780	854.898277	22297.0	5082.0	1260.094676
3	5	1.159843e+06	1.904821e+06	1376.380800	1049.534031	9957.0	28239.0	1311.296003
4	6	1.161389e+06	1.899015e+06	4047.408899	7855.346749	161713.0	48617.0	1218.767774
...	...	...	...	...	...	...	...	...
199995	249994	1.160293e+06	1.904088e+06	7429.771662	3333.061508	140909.0	36974.0	1247.330965
199996	249995	1.158267e+06	1.908710e+06	3008.240474	11278.972817	218960.0	63718.0	1331.949740
199997	249996	1.157245e+06	1.915337e+06	5036.593779	6021.532949	184049.0	42101.0	1448.809339
199998	249997	1.159126e+06	1.911090e+06	1702.060546	10547.069447	104823.0	23446.0	1399.619501
199999	249999	1.157132e+06	1.907102e+06	3663.660879	12449.702027	337362.0	107242.0	1285.571789

200000 rows × 8 columns

In [5]:
```
df.shape
```

Out[5]:
```
(200000, 8)
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
Unnamed: 0      0
x               0
y               0
x_range         0
y_range         0
count           0
sum_            0
avg           157
dtype: int64
```

In [8]:
```
df.info()
```

Out[8]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 8 columns):
 #   Column      Non-Null Count   Dtype  
---  ------      --------------   -----  
 0   Unnamed: 0  200000 non-null  int64  
 1   x           200000 non-null  float64
 2   y           200000 non-null  float64
 3   x_range     200000 non-null  float64
 4   y_range     200000 non-null  float64
 5   count       200000 non-null  float64
 6   sum_        200000 non-null  float64
 7   avg         199843 non-null  float64
dtypes: float64(7), int64(1)
memory usage: 12.2 MB
```

In [9]:
```
mean=df["avg"].mean()
mean
```

Out[9]:
```
1042.6866300470676
```

In [10]:
```
df["avg"]=df["avg"].fillna(mean)
df["avg"].isnull().sum()
```

Out[10]:
```
0
```

In [14]:
```
df.isnull().sum()
```

Out[14]:
```
Unnamed: 0    0
x             0
y             0
x_range       0
y_range       0
count         0
sum_          0
avg           0
dtype: int64
```

In [12]:
```
x=df.columns
x
```

Out[12]:
```
Index(['Unnamed: 0', 'x', 'y', 'x_range', 'y_range', 'count', 'sum_', 'avg'], dtype='object')
```

In [13]:
```
df.dtypes
```

Out[13]:
```
Unnamed: 0      int64
x             float64
y             float64
x_range       float64
y_range       float64
count         float64
sum_          float64
avg           float64
dtype: object
```

In [15]:
```
df1=df.copy()
df1
```

Out[15]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg
0	0	1.159191e+06	1.894756e+06	5225.375665	2981.728431	96046.0	34927.0	1111.618901
1	1	1.159293e+06	1.898922e+06	3499.176007	6879.352245	152668.0	54847.0	1192.855949
2	3	1.160321e+06	1.903776e+06	6495.796780	854.898277	22297.0	5082.0	1260.094676
3	5	1.159843e+06	1.904821e+06	1376.380800	1049.534031	9957.0	28239.0	1311.296003
4	6	1.161389e+06	1.899015e+06	4047.408899	7855.346749	161713.0	48617.0	1218.767774
...	...	...	...	...	...	...	...	...
199995	249994	1.160293e+06	1.904088e+06	7429.771662	3333.061508	140909.0	36974.0	1247.330965
199996	249995	1.158267e+06	1.908710e+06	3008.240474	11278.972817	218960.0	63718.0	1331.949740
199997	249996	1.157245e+06	1.915337e+06	5036.593779	6021.532949	184049.0	42101.0	1448.809339
199998	249997	1.159126e+06	1.911090e+06	1702.060546	10547.069447	104823.0	23446.0	1399.619501
199999	249999	1.157132e+06	1.907102e+06	3663.660879	12449.702027	337362.0	107242.0	1285.571789

200000 rows × 8 columns

In [16]:
```
df1.info()
```

Out[16]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 8 columns):
 #   Column      Non-Null Count   Dtype  
---  ------      --------------   -----  
 0   Unnamed: 0  200000 non-null  int64  
 1   x           200000 non-null  float64
 2   y           200000 non-null  float64
 3   x_range     200000 non-null  float64
 4   y_range     200000 non-null  float64
 5   count       200000 non-null  float64
 6   sum_        200000 non-null  float64
 7   avg         200000 non-null  float64
dtypes: float64(7), int64(1)
memory usage: 12.2 MB
```

In [17]:
```
# NORMALIZATION

# SIMPLE FEATURE SCALING (METHOD 1)

for i in x:
    df1[i]=df1[i]/df1[i].max()
df1
```

Out[17]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg
0	0.000000	0.977902	0.985217	0.645872	0.189867	0.097481	0.100526	0.494115
1	0.000004	0.977988	0.987383	0.432509	0.438056	0.154948	0.157860	0.530224
2	0.000012	0.978855	0.989907	0.802900	0.054437	0.022630	0.014627	0.560112
3	0.000020	0.978451	0.990450	0.170125	0.066923	0.010057	0.081277	0.582871
4	0.000024	0.979756	0.987431	0.500272	0.500204	0.164128	0.139929	0.541742
...	...	...	...	...	...	...	...	...
199995	0.999980	0.978831	0.990069	0.918342	0.212239	0.143014	0.106418	0.554439
199996	0.999984	0.977122	0.992473	0.371827	0.718210	0.222230	0.183392	0.592052
199997	0.999988	0.976260	0.995918	0.622538	0.383433	0.186798	0.121175	0.643996
199998	0.999992	0.977847	0.993710	0.210380	0.671605	0.106389	0.067482	0.622331
199999	1.000000	0.976165	0.991636	0.452839	0.792759	0.342401	0.308662	0.571437

200000 rows × 8 columns

In [20]:
```
df2=df.copy()
```

In [21]:
```
# MIN - MAX
for i in x:
    df2[i]=(df2[i]-df2[i].min())/(df2[i].max()-df2[i].min())
df2
```

Out[21]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg
0	0.000000	0.263018	0.535204	0.645871	0.189866	0.097481	0.100526	0.462691
1	0.000004	0.265877	0.603308	0.432508	0.438055	0.154948	0.157860	0.501044
2	0.000012	0.294793	0.682671	0.802899	0.054436	0.022630	0.014627	0.532788
3	0.000020	0.281344	0.699750	0.170124	0.066923	0.010057	0.081277	0.556961
4	0.000024	0.324840	0.604830	0.500271	0.500204	0.164128	0.139929	0.513277
...	...	...	...	...	...	...	...	...
199995	0.999980	0.294014	0.687774	0.918342	0.212239	0.143014	0.106418	0.526762
199996	0.999984	0.237001	0.763333	0.371827	0.718210	0.222230	0.183392	0.566711
199997	0.999988	0.208264	0.871665	0.622537	0.383432	0.186798	0.121175	0.621882
199998	0.999992	0.261190	0.802236	0.210379	0.671604	0.106389	0.067482	0.598659
199999	1.000000	0.205095	0.737041	0.452839	0.792758	0.342401	0.308662	0.544816

200000 rows × 8 columns

In [24]:
```
df3=df.copy()
```

In [25]:
```
# Z - SCORE
for i in x:
    df3[i]=(df3[i]-df3[i].mean())/df[i].std()
df3
```

Out[25]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg
0	-1.731929	-0.423695	0.403357	0.506984	1.073065	0.411112	0.271236	0.256611
1	-1.731915	-0.410412	0.712307	-0.232367	-0.213913	-0.044569	0.144217	0.559028
2	-1.731887	-0.286195	1.073326	1.050946	-1.541882	-0.889525	-0.893687	0.809334
3	-1.731859	-0.343968	1.148602	-1.141461	0.484890	-0.388299	-0.410722	0.999938
4	-1.731845	-0.157115	0.719208	0.002415	0.001225	-0.013984	0.014284	0.655488
...	...	...	...	...	...	...	...	...
199995	1.734544	-0.289538	1.095476	1.450924	-0.995621	-0.120691	-0.228544	0.761819
199996	1.734558	-0.534458	1.438238	-0.442612	0.755894	0.384572	0.329232	1.076825
199997	1.734572	-0.657907	1.929676	-0.426037	-0.403002	0.158576	-0.121415	1.511852
199998	1.734586	-0.430545	1.614719	-1.001808	0.594561	-0.354294	-0.519085	1.329736
199999	1.734614	-0.671523	1.318969	-0.161926	1.013958	1.151047	1.236973	0.904176

200000 rows × 8 columns

In [26]:
```
# BINNING
bins=np.linspace(min(df["avg"]),max(df["avg"]),4)
bins
```

Out[26]:
```
array([ 131.57157895,  837.62067931, 1543.66977967, 2249.71888003])
```

In [27]:
```
group_name=["low","medium","high"]
```

In [28]:
```
df["avg-binned"]=pd.cut(df["avg"],bins,labels=group_name,include_lowest=True)
```

In [30]:
```
df[["avg","avg-binned"]]
```

Out[30]:

	avg	avg-binned
0	1111.618901	medium
1	1192.855949	medium
2	1260.094676	medium
3	1311.296003	medium
4	1218.767774	medium
...	...	...
199995	1247.330965	medium
199996	1331.949740	medium
199997	1448.809339	medium
199998	1399.619501	medium
199999	1285.571789	medium

200000 rows × 2 columns

In [31]:
```
df["avg-binned"].value_counts()
```

Out[31]:
```
medium    166257
low         29748
high         3995
Name: avg-binned, dtype: int64
```

In [32]:
```
df
```

Out[32]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg	avg-binned
0	0	1.159191e+06	1.894756e+06	5225.375665	2981.728431	96046.0	34927.0	1111.618901	medium
1	1	1.159293e+06	1.898922e+06	3499.176007	6879.352245	152668.0	54847.0	1192.855949	medium
2	3	1.160321e+06	1.903776e+06	6495.796780	854.898277	22297.0	5082.0	1260.094676	medium
3	5	1.159843e+06	1.904821e+06	1376.380800	1049.534031	9957.0	28239.0	1311.296003	medium
4	6	1.161389e+06	1.899015e+06	4047.408899	7855.346749	161713.0	48617.0	1218.767774	medium
...	...	...	...	...	...	...	...	...	...
199995	249994	1.160293e+06	1.904088e+06	7429.771662	3333.061508	140909.0	36974.0	1247.330965	medium
199996	249995	1.158267e+06	1.908710e+06	3008.240474	11278.972817	218960.0	63718.0	1331.949740	medium
199997	249996	1.157245e+06	1.915337e+06	5036.593779	6021.532949	184049.0	42101.0	1448.809339	medium
199998	249997	1.159126e+06	1.911090e+06	1702.060546	10547.069447	104823.0	23446.0	1399.619501	medium
199999	249999	1.157132e+06	1.907102e+06	3663.660879	12449.702027	337362.0	107242.0	1285.571789	medium

200000 rows × 9 columns

In [33]:
```
df["avg-binned"].head(10)
```

Out[33]:
```
0    medium
1    medium
2    medium
3    medium
4    medium
5    medium
6    medium
7    medium
8    medium
9    medium
Name: avg-binned, dtype: category
Categories (3, object): ['low' < 'medium' < 'high']
```

In [35]:
```
bins=np.linspace(min(df["count"]),max(df["count"]),4)
group_name=["low","medium","high"]
df["count-binned"]=pd.cut(df["count"],bins,labels=group_name,include_lowest=True)
```

In [36]:
```
df
```

Out[36]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg	avg-binned	count-binned
0	0	1.159191e+06	1.894756e+06	5225.375665	2981.728431	96046.0	34927.0	1111.618901	medium	low
1	1	1.159293e+06	1.898922e+06	3499.176007	6879.352245	152668.0	54847.0	1192.855949	medium	low
2	3	1.160321e+06	1.903776e+06	6495.796780	854.898277	22297.0	5082.0	1260.094676	medium	low
3	5	1.159843e+06	1.904821e+06	1376.380800	1049.534031	9957.0	28239.0	1311.296003	medium	low
4	6	1.161389e+06	1.899015e+06	4047.408899	7855.346749	161713.0	48617.0	1218.767774	medium	low
...	...	...	...	...	...	...	...	...	...	...
199995	249994	1.160293e+06	1.904088e+06	7429.771662	3333.061508	140909.0	36974.0	1247.330965	medium	low
199996	249995	1.158267e+06	1.908710e+06	3008.240474	11278.972817	218960.0	63718.0	1331.949740	medium	low
199997	249996	1.157245e+06	1.915337e+06	5036.593779	6021.532949	184049.0	42101.0	1448.809339	medium	low
199998	249997	1.159126e+06	1.911090e+06	1702.060546	10547.069447	104823.0	23446.0	1399.619501	medium	low
199999	249999	1.157132e+06	1.907102e+06	3663.660879	12449.702027	337362.0	107242.0	1285.571789	medium	medium

200000 rows × 10 columns

In [44]:
```
bins=np.linspace(min(df["sum_"]),max(df["sum_"]),4)
group_name=["low","medium","high"]
df["sum_-binned"]=pd.cut(df["sum_"],bins,labels=group_name,include_lowest=True)
```

In [45]:
```
df
```

Out[45]:

	Unnamed: 0	x	y	x_range	y_range	count	sum_	avg	avg-binned	count-binned	sum_-binned
0	0	1.159191e+06	1.894756e+06	5225.375665	2981.728431	96046.0	34927.0	1111.618901	medium	low	low
1	1	1.159293e+06	1.898922e+06	3499.176007	6879.352245	152668.0	54847.0	1192.855949	medium	low	low
2	3	1.160321e+06	1.903776e+06	6495.796780	854.898277	22297.0	5082.0	1260.094676	medium	low	low
3	5	1.159843e+06	1.904821e+06	1376.380800	1049.534031	9957.0	28239.0	1311.296003	medium	low	low
4	6	1.161389e+06	1.899015e+06	4047.408899	7855.346749	161713.0	48617.0	1218.767774	medium	low	low
...	...	...	...	...	...	...	...	...	...	...	...
199995	249994	1.160293e+06	1.904088e+06	7429.771662	3333.061508	140909.0	36974.0	1247.330965	medium	low	low
199996	249995	1.158267e+06	1.908710e+06	3008.240474	11278.972817	218960.0	63718.0	1331.949740	medium	low	low
199997	249996	1.157245e+06	1.915337e+06	5036.593779	6021.532949	184049.0	42101.0	1448.809339	medium	low	low
199998	249997	1.159126e+06	1.911090e+06	1702.060546	10547.069447	104823.0	23446.0	1399.619501	medium	low	low
199999	249999	1.157132e+06	1.907102e+06	3663.660879	12449.702027	337362.0	107242.0	1285.571789	medium	medium	low

200000 rows × 11 columns

In [ ]:
```

```