Sales Data

In [5]:
```
import pandas as pd
import numpy as np
k=pd.read_csv("/home/placement/Downloads/basket_details.csv")
```

In [6]:
```
kk=pd.read_csv("/home/placement/Downloads/customer_details.csv")
```

In [7]:
```
k
```

Out[7]:

	customer_id	product_id	basket_date	basket_count
0	42366585	41475073	2019-06-19	2
1	35956841	43279538	2019-06-19	2
2	26139578	31715598	2019-06-19	3
3	3262253	47880260	2019-06-19	2
4	20056678	44747002	2019-06-19	2
...	...	...	...	...
14995	8336862	50977318	2019-05-26	2
14996	9500785	43862061	2019-05-26	2
14997	22787344	6041664	2019-05-26	2
14998	8221263	3597369	2019-05-26	2
14999	4912577	46646893	2019-05-26	2

15000 rows × 4 columns

In [8]:
```
list(k)
```

Out[8]:
```
['customer_id', 'product_id', 'basket_date', 'basket_count']
```

In [9]:
```
#k['model']=k['model'].map({'lounge':1,"pop":2,'sport':3})
k
```

Out[9]:

	customer_id	product_id	basket_date	basket_count
0	42366585	41475073	2019-06-19	2
1	35956841	43279538	2019-06-19	2
2	26139578	31715598	2019-06-19	3
3	3262253	47880260	2019-06-19	2
4	20056678	44747002	2019-06-19	2
...	...	...	...	...
14995	8336862	50977318	2019-05-26	2
14996	9500785	43862061	2019-05-26	2
14997	22787344	6041664	2019-05-26	2
14998	8221263	3597369	2019-05-26	2
14999	4912577	46646893	2019-05-26	2

15000 rows × 4 columns

In [10]:
```
kk
```

Out[10]:

	customer_id	sex	customer_age	tenure
0	9798859	Male	44.0	93
1	11413563	Male	36.0	65
2	818195	Male	35.0	129
3	12049009	Male	33.0	58
4	10083045	Male	42.0	88
...	...	...	...	...
19995	12557307	Male	41.0	52
19996	12595961	Male	29.0	52
19997	12520991	Male	35.0	52
19998	12612719	Male	39.0	52
19999	12572063	Male	28.0	52

20000 rows × 4 columns

In [11]:
```
list(kk)
```

Out[11]:
```
['customer_id', 'sex', 'customer_age', 'tenure']
```

Converting strings to numbers

In [12]:
```
kk['sex']=kk['sex'].map({"Male":1,"Feamale":0})
kk
```

Out[12]:

	customer_id	sex	customer_age	tenure
0	9798859	1.0	44.0	93
1	11413563	1.0	36.0	65
2	818195	1.0	35.0	129
3	12049009	1.0	33.0	58
4	10083045	1.0	42.0	88
...	...	...	...	...
19995	12557307	1.0	41.0	52
19996	12595961	1.0	29.0	52
19997	12520991	1.0	35.0	52
19998	12612719	1.0	39.0	52
19999	12572063	1.0	28.0	52

20000 rows × 4 columns

In [13]:
```
k.describe()
```

Out[13]:

	customer_id	product_id	basket_count
count	1.500000e+04	1.500000e+04	15000.000000
mean	1.808567e+07	3.269771e+07	2.153733
std	1.233000e+07	1.629455e+07	0.517929
min	4.784000e+03	4.939000e+04	2.000000
25%	8.659327e+06	3.137412e+07	2.000000
50%	1.520775e+07	3.694759e+07	2.000000
75%	2.663904e+07	4.502408e+07	2.000000
max	4.460824e+07	5.579097e+07	10.000000

In [14]:
```
kk.describe()
```

Out[14]:

	customer_id	sex	customer_age	tenure
count	2.000000e+04	15322.0	20000.000000	20000.000000
mean	1.760040e+07	1.0	262.222550	44.396800
std	8.679505e+06	0.0	604.321589	31.998376
min	2.093000e+03	1.0	-34.000000	4.000000
25%	1.188115e+07	1.0	29.000000	21.000000
50%	1.560912e+07	1.0	38.000000	35.000000
75%	2.228484e+07	1.0	123.000000	60.000000
max	4.462566e+07	1.0	2022.000000	133.000000

In [15]:
```
kk.groupby(["customer_id"]).count()
```

Out[15]:

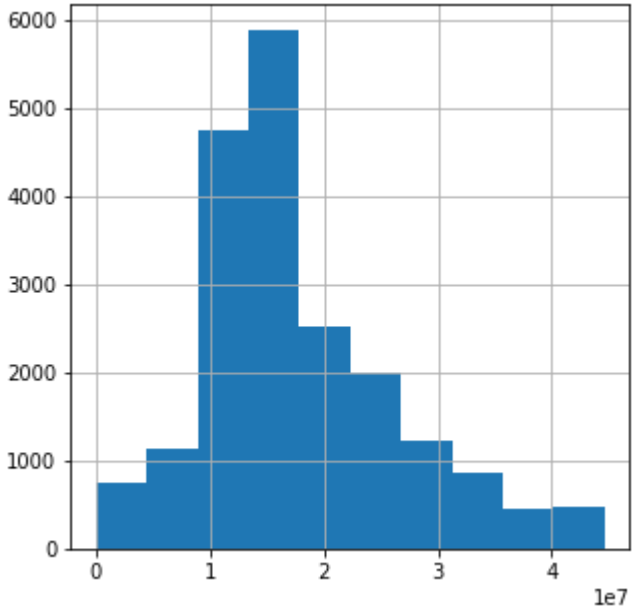
customer_id	sex	customer_age	tenure
2093	1	1	1
12817	1	1	1
14309	0	1	1
15155	1	1	1
23205	1	1	1
...	...	...	...
44392831	1	1	1
44401175	0	1	1
44431821	0	1	1
44621778	1	1	1
44625658	0	1	1

20000 rows × 3 columns

Histograph

In [16]:
```
kk['customer_id'].hist(figsize=(5,5))
```

Out[16]: <Axes: >



Merging Data

In [17]:
```
test=pd.merge(k,kk,on="customer_id")
test
```

Out[17]:

	customer_id	product_id	basket_date	basket_count	sex	customer_age	tenure
0	4897641	34525548	2019-06-15	2	1.0	40.0	114
1	11623549	50394038	2019-06-18	2	1.0	30.0	63
2	11665521	41476812	2019-06-15	2	NaN	51.0	62
3	4193819	6455162	2019-06-15	2	1.0	42.0	117
4	1030589	38578121	2019-05-26	2	1.0	45.0	127
...	...	...	...	...	...	...	...
67	12574807	32056122	2019-05-25	2	1.0	33.0	52
68	15192667	31272089	2019-05-24	2	1.0	46.0	37
69	14248059	48790153	2019-05-21	2	1.0	29.0	41
70	10629563	47864502	2019-06-01	2	1.0	29.0	76
71	11737579	46626448	2019-05-27	2	1.0	35.0	61

72 rows × 7 columns

In [18]:
```
test.describe()
```

Out[18]:

	customer_id	product_id	basket_count	sex	customer_age	tenure
count	7.200000e+01	7.200000e+01	72.000000	58.0	72.000000	72.000000
mean	1.554364e+07	3.140376e+07	2.152778	1.0	68.458333	56.180556
std	9.961282e+06	1.616160e+07	0.362298	0.0	234.574289	38.948621
min	3.809750e+05	8.287500e+04	2.000000	1.0	5.000000	4.000000
25%	1.026443e+07	2.980404e+07	2.000000	1.0	29.000000	24.750000
50%	1.352736e+07	3.498005e+07	2.000000	1.0	35.500000	45.500000
75%	2.037478e+07	4.359420e+07	2.000000	1.0	43.000000	83.750000
max	4.328080e+07	5.130767e+07	3.000000	1.0	2022.000000	130.000000

In [19]:
```
test.customer_id.unique()
```

Out[19]:
```
array([ 4897641, 11623549, 11665521,  4193819,  1030589, 20236456,
       15436141, 10394153, 10619833, 21765975, 16029475, 12737235,
       21142247, 15067633,  4238087, 17909829, 11346069, 25567283,
         380975,  4257099, 11440499, 20174063,   537173, 25055107,
       39814593,  9654043, 16398473, 11724853,  4643359,  9700145,
       29144255, 14053193, 36623391, 22524187,  8508353, 12901520,
       20789769, 16944627, 23179191, 15141119, 41790413, 27081691,
        9804585, 18256077,  4912369, 43280797,  9500953, 12410433,
        9875271,   851739, 10439331, 13776147, 11072047, 15570891,
       14966315, 10814041, 34677755, 17830393, 13278573, 12574807,
       15192667, 14248059, 10629563, 11737579])
```

In [20]:
```
k.head()
```

Out[20]:

	customer_id	product_id	basket_date	basket_count
0	42366585	41475073	2019-06-19	2
1	35956841	43279538	2019-06-19	2
2	26139578	31715598	2019-06-19	3
3	3262253	47880260	2019-06-19	2
4	20056678	44747002	2019-06-19	2

In [21]:
```
k.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 4 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   customer_id   15000 non-null  int64 
 1   product_id    15000 non-null  int64 
 2   basket_date   15000 non-null  object
 3   basket_count  15000 non-null  int64 
dtypes: int64(3), object(1)
memory usage: 468.9+ KB
```

In [22]:
```
k.loc[k.customer_id>42366585]
```

Out[22]:

	customer_id	product_id	basket_date	basket_count
16	44025439	34631629	2019-06-18	2
23	44417560	43864701	2019-06-18	2
35	43213385	4131698	2019-06-17	2
36	43352582	43357885	2019-06-17	2
40	44416693	40276068	2019-06-17	2
...	...	...	...	...
14819	42470642	41688490	2019-05-27	2
14820	42555676	32699421	2019-05-27	5
14849	42556595	5938721	2019-05-27	2
14881	42514496	5904669	2019-05-26	2
14958	42451678	46411638	2019-05-26	2

439 rows × 4 columns

Group by

In [23]:
```
group= test.groupby(["customer_id"])['basket_count'].sum().sort_values(ascending=False)
group
```

Out[23]:
```
customer_id
39814593    5
20236456    5
12737235    5
380975      4
27081691    4
           ..
14053193    2
14248059    2
14966315    2
15067633    2
13278573    2
Name: basket_count, Length: 64, dtype: int64
```

In [24]:
```
group.describe()
```

Out[24]:
```
count    64.000000
mean      2.421875
std       0.831993
min       2.000000
25%       2.000000
50%       2.000000
75%       2.250000
max       5.000000
Name: basket_count, dtype: float64
```

Correlation

In [25]:
```
cor=kk.corr()
cor
```

Out[25]:

	customer_id	sex	customer_age	tenure
customer_id	1.000000	NaN	-0.075467	-0.855410
sex	NaN	NaN	NaN	NaN
customer_age	-0.075467	NaN	1.000000	-0.095013
tenure	-0.855410	NaN	-0.095013	1.000000

Heat map

In [26]:
```
import seaborn as s
s.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr')
```

Out[26]: <Axes: >

In [ ]:
```
4
```