#16/june/23 #basket and customer csv files both files we can read and analize

In [1]:
```
import pandas as pd
```

In [2]:
```
!pip3 install seaborn
```

```
Requirement already satisfied: python-dateutil>=2.7 in /home/placement/anaconda3/lib/python3.10/site-pack
ages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: packaging>=20.0 in /home/placement/anaconda3/lib/python3.10/site-packages
(from matplotlib!=3.6.1,>=3.1->seaborn) (22.0)
Requirement already satisfied: pyparsing>=2.3.1 in /home/placement/anaconda3/lib/python3.10/site-packages
(from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /home/placement/anaconda3/lib/python3.10/site-package
s (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in /home/placement/anaconda3/lib/python3.10/site-packages
(from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.5)
Requirement already satisfied: pillow>=6.2.0 in /home/placement/anaconda3/lib/python3.10/site-packages (f
rom matplotlib!=3.6.1,>=3.1->seaborn) (9.4.0)
Requirement already satisfied: fonttools>=4.22.0 in /home/placement/anaconda3/lib/python3.10/site-package
s (from matplotlib!=3.6.1,>=3.1->seaborn) (4.25.0)
Requirement already satisfied: cycler>=0.10 in /home/placement/anaconda3/lib/python3.10/site-packages (fr
om matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: pytz>=2020.1 in /home/placement/anaconda3/lib/python3.10/site-packages (fr
om pandas>=0.25->seaborn) (2022.7)
Requirement already satisfied: six>=1.5 in /home/placement/anaconda3/lib/python3.10/site-packages (from p
ython-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
```

In [ ]:
```
```

In [3]:
```
data=pd.read_csv("/home/placement/Downloads/customer_details.csv")
data1=pd.read_csv("/home/placement/Downloads/basket_details.csv")
```

In [4]:
```
data.describe()
```

Out[4]:

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

In [5]:
```
data1.describe()
```

Out[5]:

	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 [6]:
```
data.shape
```

Out[6]:
```
(20000, 4)
```

In [7]:
```
data1.shape
```

Out[7]:
```
(15000, 4)
```

In [8]:
```
data.tail
```

Out[8]:
```
<bound method NDFrame.tail of        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 x 4 columns]>
```

In [9]:
```
data1.tail
```

Out[9]:
```
<bound method NDFrame.tail of        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 x 4 columns]>
```

In [10]:
```
data1.groupby(['customer_id']).count()
```

Out[10]:

customer_id	product_id	basket_date	basket_count
4784	1	1	1
8314	2	2	2
8857	1	1	1
9273	1	1	1
11172	1	1	1
...	...	...	...
44460516	1	1	1
44461180	1	1	1
44473609	1	1	1
44486815	1	1	1
44608245	1	1	1

13871 rows × 3 columns

In [11]:
```
data.groupby(['customer_id']).count()
```

Out[11]:

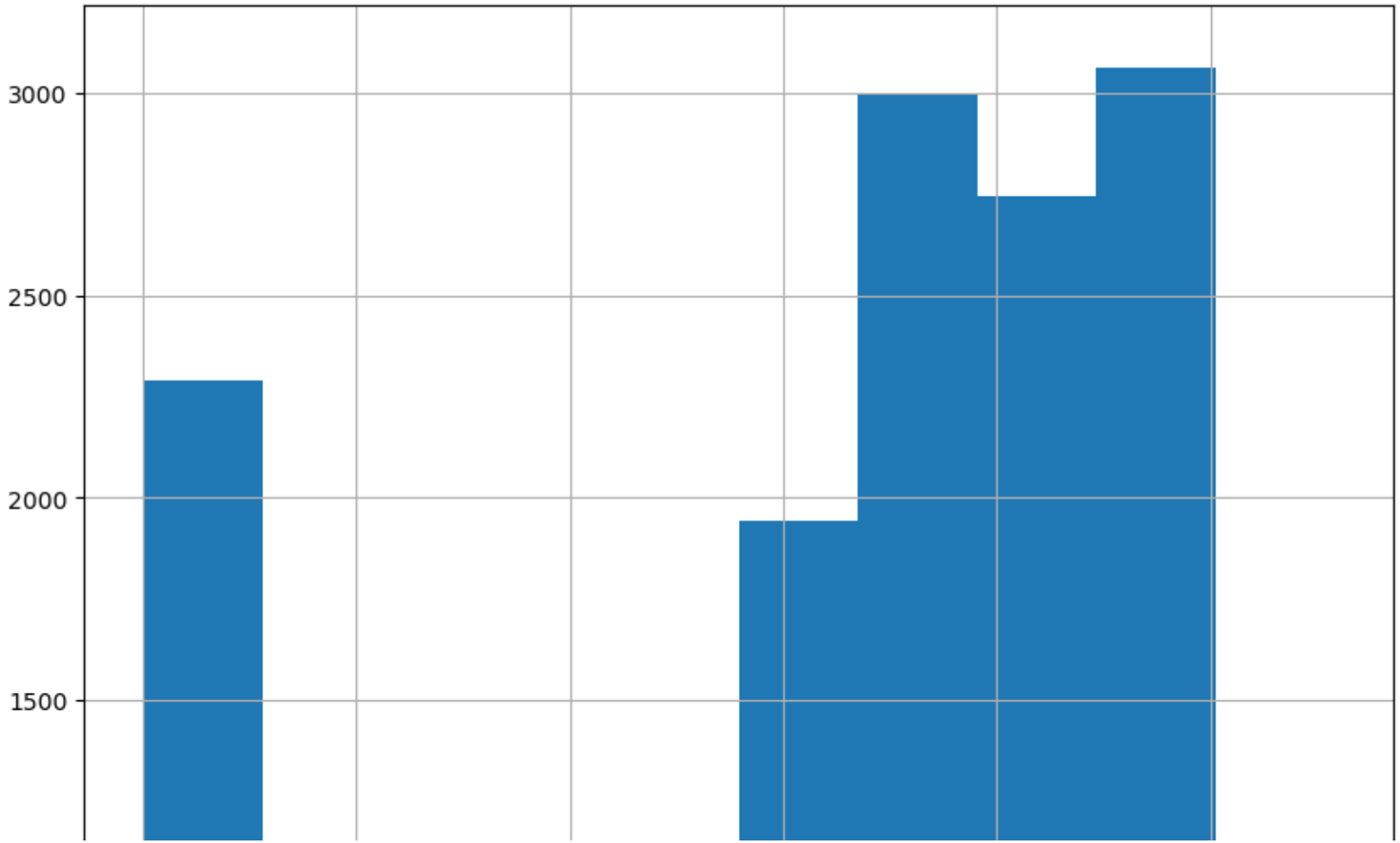
customer_id	sex	customer_age	tenure
2093	1	1	1
12817	1	1	1
14309	1	1	1
15155	1	1	1
23205	1	1	1
...	...	...	...
44392831	1	1	1
44401175	1	1	1
44431821	1	1	1
44621778	1	1	1
44625658	1	1	1

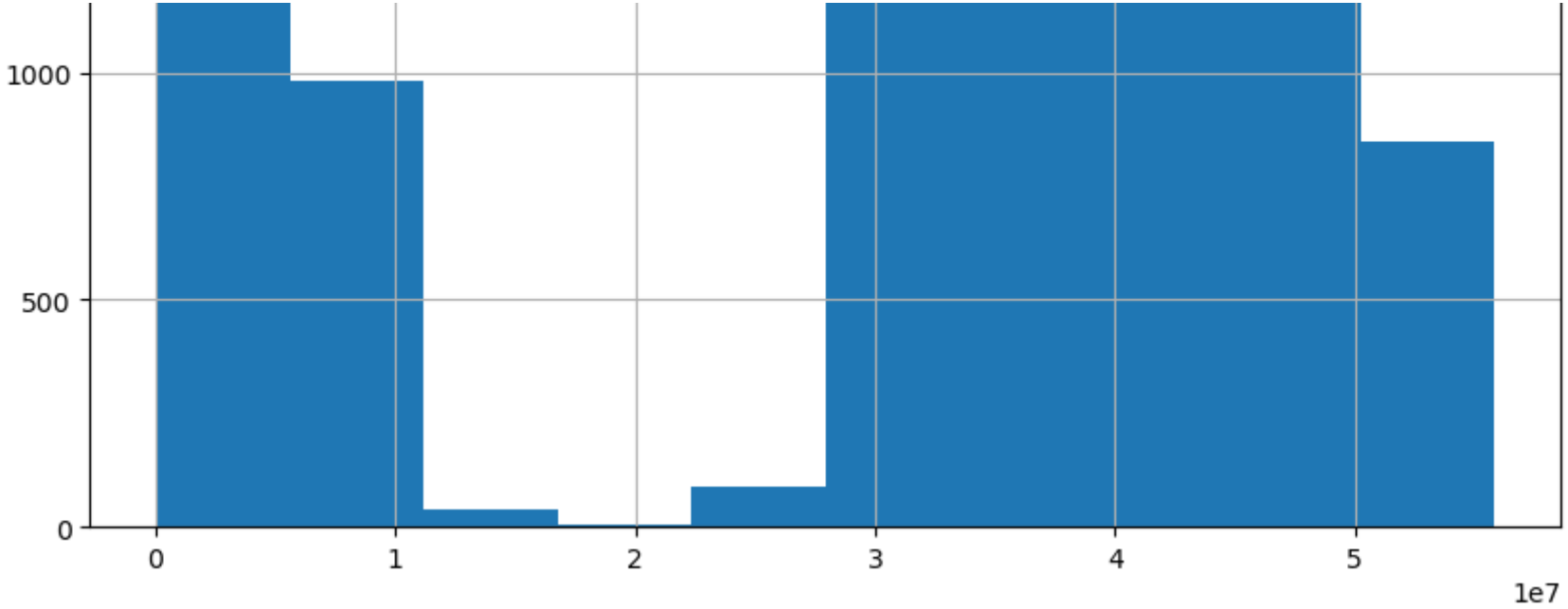
20000 rows × 3 columns

In [12]:
```
data1['product_id'].hist(figsize=(10,10))
plt.show()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[12], line 2
      1 data1['product_id'].hist(figsize=(10,10))
----> 2 plt.show()

NameError: name 'plt' is not defined
```





In [13]:
```
test=pd.merge(data, data1, on= "customer_id")
```

In [14]:
```
test
```

Out[14]:

	customer_id	sex	customer_age	tenure	product_id	basket_date	basket_count
0	9500953	Male	55.0	96	3446783	2019-06-10	3
1	851739	Male	40.0	129	32920704	2019-06-19	2
2	9654043	Male	37.0	95	51307669	2019-06-08	2
3	4912369	Male	36.0	114	33923115	2019-05-20	2
4	9875271	Male	34.0	92	31586037	2019-06-06	2
...	...	...	...	...	...	...	...
67	13278573	Male	28.0	47	4488682	2019-05-26	2
68	12901520	Female	40.0	50	38610580	2019-05-28	3
69	12737235	Male	39.0	51	32933848	2019-05-21	2
70	12737235	Male	39.0	51	46373374	2019-05-21	3
71	12574807	Male	33.0	52	32056122	2019-05-25	2

72 rows × 7 columns

In [15]:
```
test.head
```

Out[15]:
```
<bound method NDFrame.head of     customer_id     sex  customer_age  tenure  product_id basket_date  \
0       9500953    Male          55.0      96     3446783  2019-06-10
1        851739    Male          40.0     129    32920704  2019-06-19
2       9654043    Male          37.0      95    51307669  2019-06-08
3       4912369    Male          36.0     114    33923115  2019-05-20
4       9875271    Male          34.0      92    31586037  2019-06-06
..          ...     ...           ...     ...         ...         ...
67     13278573    Male          28.0      47     4488682  2019-05-26
68     12901520  Female          40.0      50    38610580  2019-05-28
69     12737235    Male          39.0      51    32933848  2019-05-21
70     12737235    Male          39.0      51    46373374  2019-05-21
71     12574807    Male          33.0      52    32056122  2019-05-25

    basket_count
0              3
1              2
2              2
3              2
4              2
..           ...
67             2
68             3
69             2
70             3
71             2

[72 rows x 7 columns]>
```

In [16]:
```
test.describe()
```

Out[16]:

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

In [17]:
```
test.customer_id.unique()
```

Out[17]:
```
array([ 9500953,   851739,  9654043,  4912369,  9875271, 11737579,
       10619833,  4193819,  4897641,  4643359,   380975, 11623549,
       11724853, 12410433, 10394153,   537173, 11440499, 10439331,
       10629563,  4257099, 11346069,  8508353,  9700145, 10814041,
        9804585,  4238087, 11665521,  1030589, 11072047, 43280797,
       41790413, 39814593, 36623391, 34677755, 29144255, 27081691,
       25055107, 25567283, 23179191, 22524187, 21765975, 21142247,
       20789769, 20236456, 20174063, 17909829, 18256077, 17830393,
       16944627, 16398473, 16029475, 15436141, 15570891, 15192667,
       15067633, 14966315, 15141119, 14248059, 14053193, 13776147,
       13278573, 12901520, 12737235, 12574807])
```

In [18]:
```
data1.head
```

Out[18]:
```
<bound method NDFrame.head of        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 x 4 columns]>
```

In [19]:
```
data1.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=False)
#true means ascending order
#false means decending order
```

Out[19]:
```
product_id
43524799    69
31516269    59
39833031    50
46130148    36
34913531    28
            ..
34003520     2
34003697     2
34004660     2
34013459     2
55790974     2
Name: basket_count, Length: 13161, dtype: int64
```

In [20]:
```
data1.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=True)
#true means ascending order
#false means decending order
```

Out[20]:
```
product_id
49390        2
42094163     2
42102274     2
42110403     2
42110580     2
            ..
34913531    28
46130148    36
39833031    50
31516269    59
43524799    69
Name: basket_count, Length: 13161, dtype: int64
```

In [21]:
```
#true means ascending order
#false means decending order
```

In [22]:
```
test.groupby(['customer_age']).count()
```

Out[22]:

customer_age	customer_id	sex	tenure	product_id	basket_date	basket_count
5.0	1	1	1	1	1	1
22.0	2	2	2	2	2	2
23.0	1	1	1	1	1	1
24.0	2	2	2	2	2	2
25.0	2	2	2	2	2	2
26.0	1	1	1	1	1	1
27.0	4	4	4	4	4	4
28.0	3	3	3	3	3	3
29.0	6	6	6	6	6	6
30.0	3	3	3	3	3	3
32.0	4	4	4	4	4	4
33.0	2	2	2	2	2	2
34.0	3	3	3	3	3	3
35.0	2	2	2	2	2	2
36.0	4	4	4	4	4	4
37.0	2	2	2	2	2	2
39.0	3	3	3	3	3	3
40.0	5	5	5	5	5	5
41.0	1	1	1	1	1	1
42.0	2	2	2	2	2	2
43.0	3	3	3	3	3	3
45.0	1	1	1	1	1	1
46.0	1	1	1	1	1	1

customer_age	customer_id	sex	tenure	product_id	basket_date	basket_count
51.0	3	3	3	3	3	3
55.0	1	1	1	1	1	1
57.0	2	2	2	2	2	2
61.0	1	1	1	1	1	1
67.0	2	2	2	2	2	2
123.0	4	4	4	4	4	4
2022.0	1	1	1	1	1	1

In [23]:
```
#-------------------------------------------------------------
import seaborn as sns
```

In [ ]:
```
```

In [24]:
```
cor=data1.corr()
cor
```

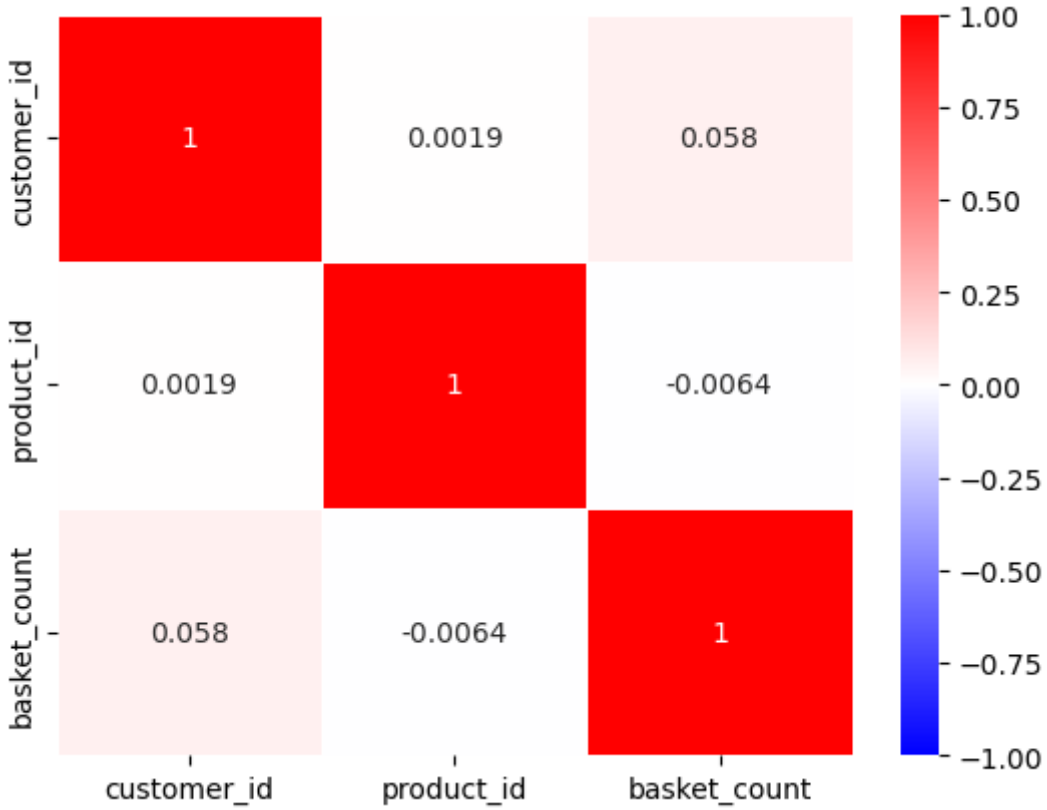
```
/tmp/ipykernel_5213/870474124.py:1: FutureWarning: The default value of numeric_only in DataFrame.corr is d
eprecated. In a future version, it will default to False. Select only valid columns or specify the value of
numeric_only to silence this warning.
  cor=data1.corr()
```

Out[24]:

	customer_id	product_id	basket_count
customer_id	1.000000	0.001937	0.058235
product_id	0.001937	1.000000	-0.006407
basket_count	0.058235	-0.006407	1.000000

In [25]:
```
#-------------------------------------------------------------
import seaborn as sns
sns.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidths=.5,cmap='bwr')
```

Out[25]: <Axes: >



In [ ]: