Importing numpy and pandas

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

Reading basket data.csv file

In [2]:
```
bask=pd.read_csv("/home/placement/Downloads/basket_details.csv")
bask
```

Out[2]:

	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

Reading customer details.csv file

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

Out[3]:

	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 [4]:
```
head1=bask.head(10)
head1
```

Out[4]:

	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
5	32037116	33739394	2019-06-19	2
6	17565651	46000191	2019-06-19	2
7	42079380	46881033	2019-06-19	2
8	25533477	44752779	2019-06-19	2
9	10385144	41882886	2019-06-19	2

In [5]:
```
head=cust.head(10)
head
```

Out[5]:

	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
5	11248447	Male	37.0	68
6	819721	Male	46.0	129
7	4713723	Male	35.0	115
8	11141669	Male	36.0	69
9	10844015	Male	37.0	73

Converting strings into integers

In [6]:
```
cust['sex']=cust['sex'].map({'Male':1,'Female':2})
cust
```

Out[6]:

	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

Checking values

In [7]:
```
a=cust.loc[cust.sex==2]
a
```

Out[7]:

	customer_id	sex	customer_age	tenure
16	831271	2.0	38.0	129
18	11350661	2.0	24.0	66
23	11328737	2.0	41.0	66
28	12417929	2.0	35.0	54
32	10189011	2.0	39.0	86
...	...	...	...	...
19973	12623079	2.0	49.0	52
19977	12606531	2.0	36.0	52
19986	12560981	2.0	46.0	52
19987	12525219	2.0	40.0	52
19990	12595849	2.0	27.0	52

4669 rows × 4 columns

Grouping data

In [8]:
```
bask.groupby(['customer_id']).count()
```

Out[8]:

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 [9]:
```
cust.groupby(['customer_id']).count()
```

Out[9]:

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

Importing seaborn and matplotlib

In [10]:
```
import seaborn as sb
import matplotlib.pyplot as mp
```

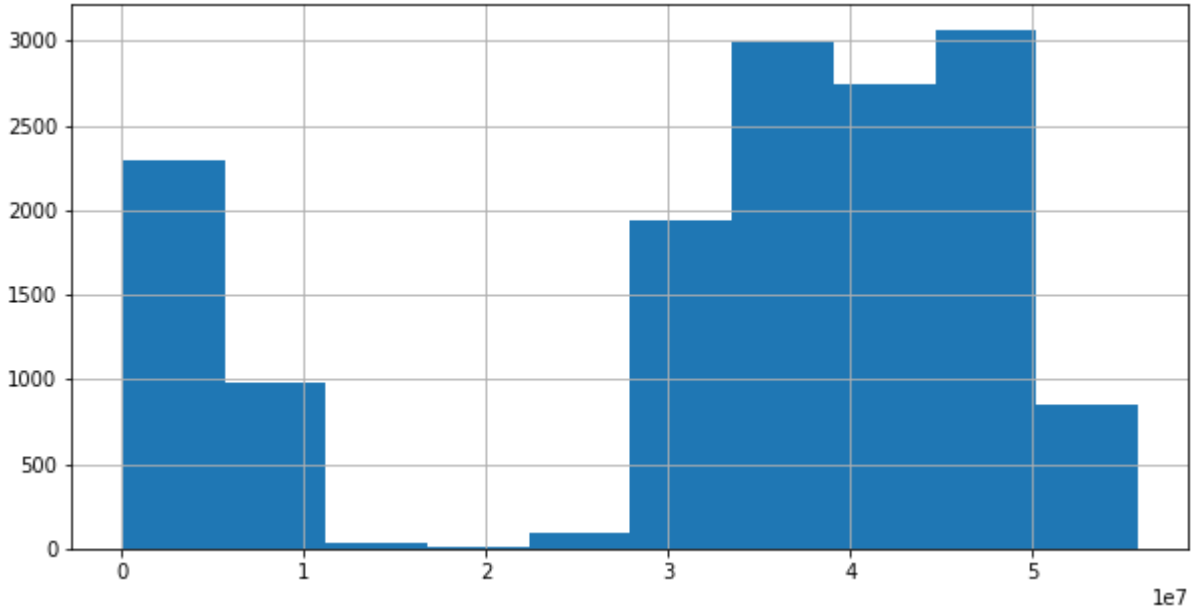
Correlating graph to the data

In [11]:
```
cor=cust.corr()
sb.heatmap(cor,vmax=1,vmin=-1,annot=True,linewidth=5,cmap='bwr')
```

Out[11]: <Axes: >

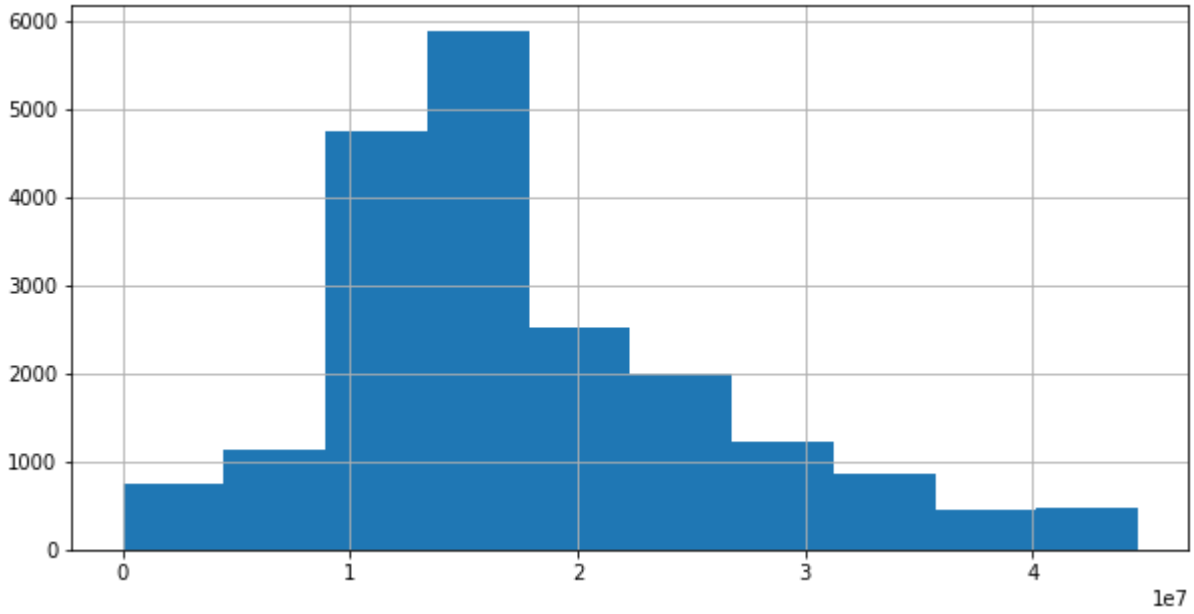
Histograph to the data

In [12]:
```
bask['product_id'].hist(figsize=(10,5))
mp.show()
```



In [13]:
```
cust['customer_id'].hist(figsize=(10,5))
mp.plot()
```

Out[13]: []



Merging the two data files

In [14]:
```
test=pd.merge(bask,cust,on='customer_id')
test
```

Out[14]:

	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	2.0	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

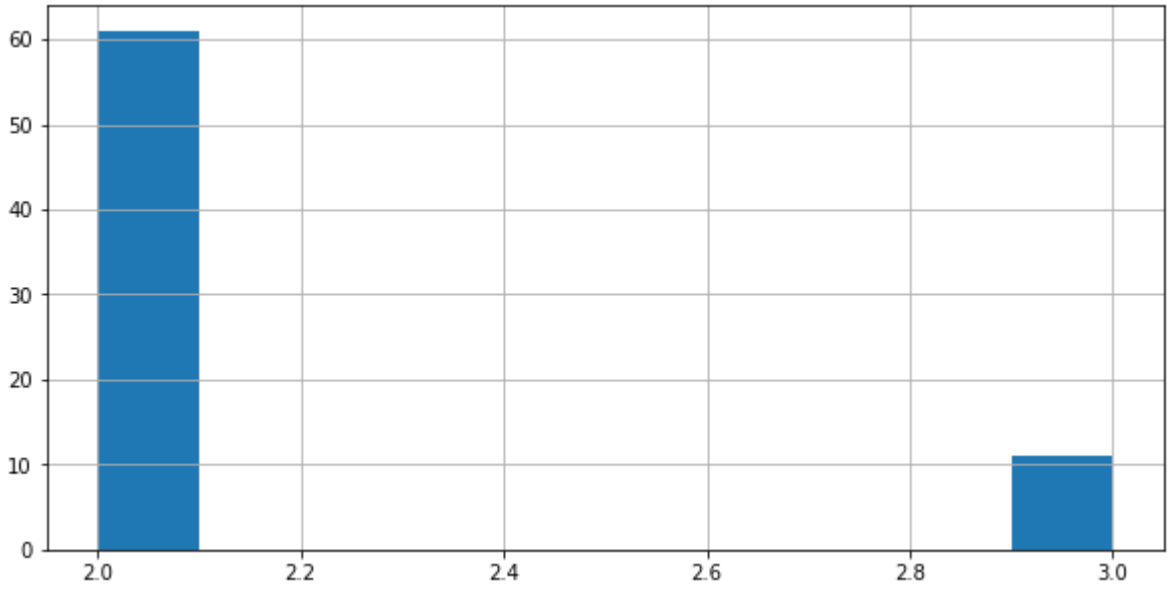
In [15]:
```
test.describe()
```

Out[15]:

	customer_id	product_id	basket_count	sex	customer_age	tenure
count	7.200000e+01	7.200000e+01	72.000000	72.000000	72.000000	72.000000
mean	1.554364e+07	3.140376e+07	2.152778	1.194444	68.458333	56.180556
std	9.961282e+06	1.616160e+07	0.362298	0.398550	234.574289	38.948621
min	3.809750e+05	8.287500e+04	2.000000	1.000000	5.000000	4.000000
25%	1.026443e+07	2.980404e+07	2.000000	1.000000	29.000000	24.750000
50%	1.352736e+07	3.498005e+07	2.000000	1.000000	35.500000	45.500000
75%	2.037478e+07	4.359420e+07	2.000000	1.000000	43.000000	83.750000
max	4.328080e+07	5.130767e+07	3.000000	2.000000	2022.000000	130.000000

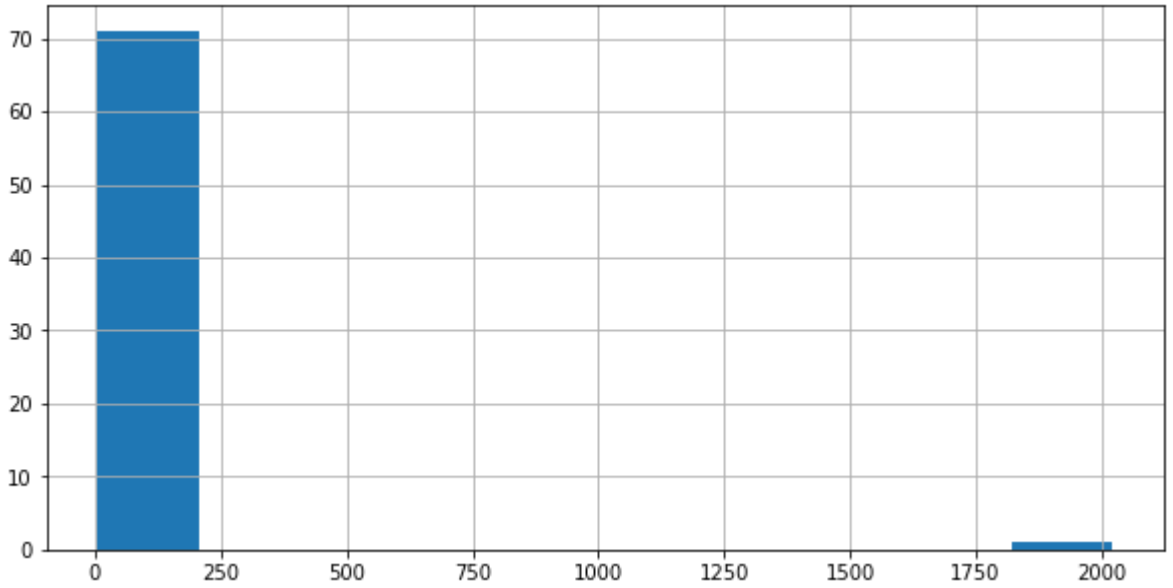
In [16]:
```
test['basket_count'].hist(figsize=(10,5))
mp.plot()
```

Out[16]: []



In [17]:
```
test['customer_age'].hist(figsize=(10,5))
mp.plot()
```

Out[17]: []



unique values

In [18]:
```
test.customer_id.unique()
```

Out[18]:
```
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])
```

Grouping data and arranging in descending order

In [19]:
```
bask.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=False)
```

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]:
```
bask.groupby(['product_id'])['basket_count'].sum().sort_values(ascending=True)
```

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 [ ]:

In [ ]: