In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df= pd.read_csv(r"C:\Users\dhruv\Desktop\ASI 2019\DataTEST\HST.csv")
```

```
C:\Users\dhruv\Anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y:2785: DtypeWarning: Columns (2,23) have mixed types. Specify dtype option
on import or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

In [3]:
```
df.head()
```

Out[3]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	PACK	WHO	...	LVL2QTY	LVL2PRIC
0	230	2008-01-01	2008-01-31	3	0.0	0.0	NaN	1	1	asi	...	0	0.
1	230	2008-02-01	2008-02-29	2	0.0	0.0	NaN	1	1	asi	...	0	0.
2	230	2008-03-01	2008-03-31	8	0.0	0.0	NaN	1	1	asi	...	0	0.
3	230	2008-04-01	2008-04-30	5	0.0	0.0	NaN	1	1	asi	...	0	0.
4	230	2008-05-01	2008-05-31	2	0.0	0.0	NaN	1	1	asi	...	0	0.

5 rows × 24 columns

In [4]:
```
df = df[df.PRICE != 0]
```

In [5]:
```
pd.set_option('display.max_columns', None)
df.tail()
```

Out[5]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	PACK	WHO	TSTAMP	LVL1C
574744	34073	2019-01-29	NaN	3	42.97	21.00	CLUB	4	1	pos	2019-01-29 14:11:08	
574745	33631	2019-01-29	NaN	3	44.97	32.01	NaN	4	1	pos	2019-01-29 14:11:08	
574746	32945	2019-01-29	NaN	4	38.96	23.96	CLUB	4	1	pos	2019-01-29 14:11:08	
574747	32684	2019-01-29	NaN	3	36.97	26.07	CLUB	4	1	pos	2019-01-29 14:11:08	
574748	33447	2019-01-29	NaN	2	19.98	13.34	CLUB	4	1	pos	2019-01-29 14:11:08	

In [6]:
```
df.tail()
```

Out[6]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	PACK	WHO	TSTAMP	LVL1C
574744	34073	2019-01-29	NaN	3	42.97	21.00	CLUB	4	1	pos	2019-01-29 14:11:08	
574745	33631	2019-01-29	NaN	3	44.97	32.01	NaN	4	1	pos	2019-01-29 14:11:08	
574746	32945	2019-01-29	NaN	4	38.96	23.96	CLUB	4	1	pos	2019-01-29 14:11:08	
574747	32684	2019-01-29	NaN	3	36.97	26.07	CLUB	4	1	pos	2019-01-29 14:11:08	
574748	33447	2019-01-29	NaN	2	19.98	13.34	CLUB	4	1	pos	2019-01-29 14:11:08	

In [7]:
```
df.drop('PACK',axis=1, inplace=True)
```

In [8]:
```
df.drop('WHO',axis=1, inplace=True)
df.head()
```

Out[8]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	TSTAMP	LVL1QTY	LVL1PRICE
3317	1288	2008-05-07	NaN	12	9.49	6.80	NaN	1	2008-05-07 14:07:10	0	0.0
3318	146	2008-05-07	NaN	60	27.98	23.10	NaN	1	2008-05-07 14:07:11	0	0.0
3319	1062	2008-05-07	NaN	36	25.98	21.00	NaN	1	2008-05-07 14:07:10	0	0.0
3320	996	2008-05-07	NaN	6	8.49	5.89	NaN	1	2008-05-07 14:07:10	0	0.0
3321	100	2008-05-07	NaN	24	15.49	14.00	NaN	1	2008-05-07 14:07:10	0	0.0

In [9]:
```
tf=df[df.SENT == True]
tf
```

Out[9]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	TSTAMP	LVL1QTY	LVL1PRICE	LVL1C

In [10]:
```
df.drop('SENT',axis=1, inplace=True)
df.head()
```

Out[10]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	TSTAMP	LVL1QTY	LVL1PRICE
3317	1288	2008-05-07	NaN	12	9.49	6.80	NaN	1	2008-05-07 14:07:10	0	0.0
3318	146	2008-05-07	NaN	60	27.98	23.10	NaN	1	2008-05-07 14:07:11	0	0.0
3319	1062	2008-05-07	NaN	36	25.98	21.00	NaN	1	2008-05-07 14:07:10	0	0.0
3320	996	2008-05-07	NaN	6	8.49	5.89	NaN	1	2008-05-07 14:07:10	0	0.0
3321	100	2008-05-07	NaN	24	15.49	14.00	NaN	1	2008-05-07 14:07:10	0	0.0

Above, we dropped a couple unnecessary columns.

In [11]:
```
tf= df[df.PRICE < 0]
tf.count()
```

Out[11]:
```
SKU          1355
DATE         1355
EDATE           0
QTY          1355
PRICE        1355
COST         1355
PROMO          50
STORE        1355
TSTAMP       1355
LVL1QTY      1355
LVL1PRICE    1355
LVL1COST     1355
LVL2QTY      1355
LVL2PRICE    1355
LVL2COST     1355
LVL3QTY      1355
LVL3PRICE    1355
LVL3COST     1355
LVL4QTY      1355
LVL4PRICE    1355
LVL4COST     1355
dtype: int64
```

Here we can see that about 1322 rows of data have prices and costs in the negative range. We shall remove these rows from our dataset.

In [12]:
```
df = df[df.PRICE > 0]
df = df[df.COST > 0]
```

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

Out[13]:

	SKU	QTY	PRICE	COST	STORE	LVL1
count	569429.000000	569429.000000	569429.000000	569429.000000	569429.000000	569429.000
mean	11524.053622	20.165016	24.666444	19.697673	1.000037	5.388
std	10991.314509	41.659781	35.094940	45.318014	0.010518	10.473
min	3.000000	0.000000	0.010000	0.010000	1.000000	-108.000
25%	815.000000	6.000000	8.990000	6.650000	1.000000	0.000
50%	11990.000000	10.000000	14.290000	10.780000	1.000000	3.000
75%	20117.000000	18.000000	26.970000	20.340000	1.000000	6.000
max	34407.000000	1590.000000	1531.670000	24475.350000	4.000000	1062.000

Here we can see that the Mean or Average Price of an item is 24.66 and the average cost of the item is 19.69 and we have no prices in the negative range.

Now we can begin our analysis.

Figure out the profit margins of each product and add the column to the datatable.

In [14]:
```
df['PROFIT'] = df['PRICE']-df['COST']
df.head()
```

Out[14]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	TSTAMP	LVL1QTY	LVL1PRICE
3317	1288	2008-05-07	NaN	12	9.49	6.80	NaN	1	2008-05-07 14:07:10	0	0.0
3318	146	2008-05-07	NaN	60	27.98	23.10	NaN	1	2008-05-07 14:07:11	0	0.0
3319	1062	2008-05-07	NaN	36	25.98	21.00	NaN	1	2008-05-07 14:07:10	0	0.0
3320	996	2008-05-07	NaN	6	8.49	5.89	NaN	1	2008-05-07 14:07:10	0	0.0
3321	100	2008-05-07	NaN	24	15.49	14.00	NaN	1	2008-05-07 14:07:10	0	0.0

In [15]:
```
df.head()
```

Out[15]:

	SKU	DATE	EDATE	QTY	PRICE	COST	PROMO	STORE	TSTAMP	LVL1QTY	LVL1PRICE
3317	1288	2008-05-07	NaN	12	9.49	6.80	NaN	1	2008-05-07 14:07:10	0	0.0
3318	146	2008-05-07	NaN	60	27.98	23.10	NaN	1	2008-05-07 14:07:11	0	0.0
3319	1062	2008-05-07	NaN	36	25.98	21.00	NaN	1	2008-05-07 14:07:10	0	0.0
3320	996	2008-05-07	NaN	6	8.49	5.89	NaN	1	2008-05-07 14:07:10	0	0.0
3321	100	2008-05-07	NaN	24	15.49	14.00	NaN	1	2008-05-07 14:07:10	0	0.0

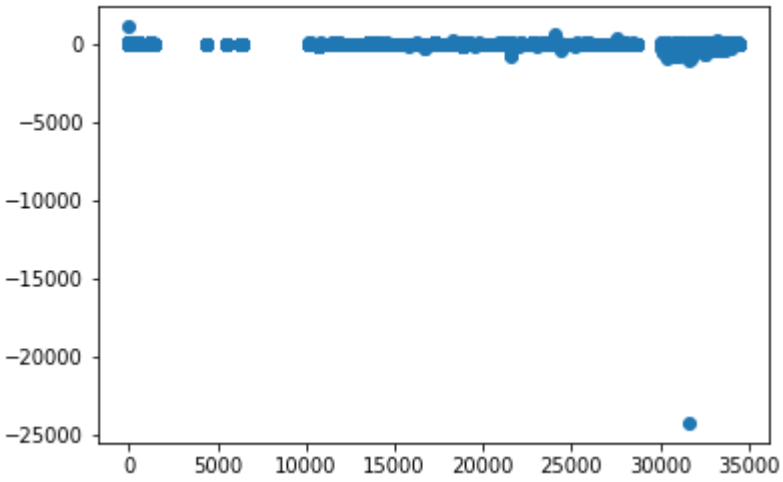
In [16]:
```
losses = df[df.PROFIT < 0]
print(losses.PROFIT.sum())
print(losses.PROFIT.count())
```

```
-216517.62999999998
4913
```

Here we can see that all the losses added up to $216,517 over the course of 4913 items sold within the history of this dataset.

In [17]:
```
plt.scatter(df.SKU,df.PROFIT)
```

Out[17]: <matplotlib.collections.PathCollection at 0x1e2a3d45278>

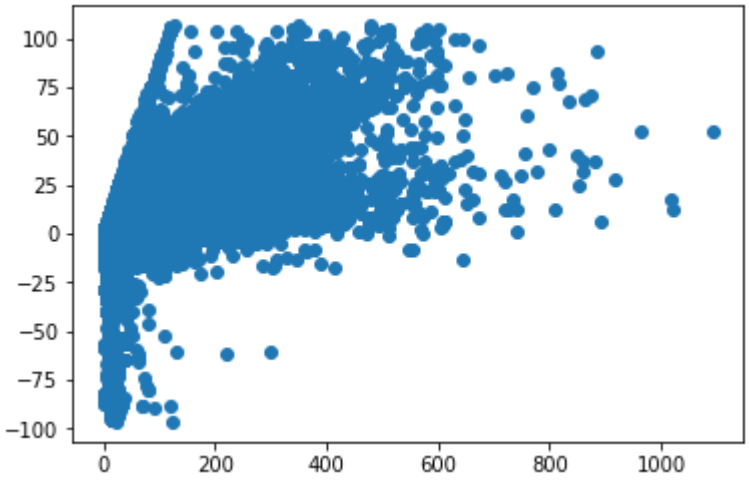


In [18]:
```
df= df[np.abs(df.PROFIT-df.PROFIT.mean()) <= (3*df.PROFIT.std())]
```

Removed the single outlier

In [19]:
```
plt.scatter(df.PRICE,df.PROFIT)
```

Out[19]: <matplotlib.collections.PathCollection at 0x1e2a3d9f668>



In [1]:
```
#pd.to_datetime(df.DATE)
```