Date-23/10/2023

Team ID-689

Project Title-Market Basket Analysis for Fresh Product Location Improvement

```
import pandas as pd
import numpy as np

#for viz
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

#to avoid warning
import warnings
warnings.filterwarnings('ignore')

#to display all feature if the number increase
pd.set_option('display.max_columns', None)
```

```
data=pd.read_excel('/content/Assignment-1_Data.xlsx')
```

```
data.head()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom

```
data.tail()
```

	BillNo	Itemname	Quantity	Date	Price	CustomerID	Country
522059	581587	PACK OF 20 SPACEBOY NAPKINS	12	2011-12-09 12:50:00	0.85	12680.0	France
522060	581587	CHILDREN'S APRON DOLLY GIRL	6	2011-12-09 12:50:00	2.10	12680.0	France
522061	581587	CHILDRENS CUTLERY DOLLY GIRL	4	2011-12-09 12:50:00	4.15	12680.0	France
522062	581587	CHILDRENS CUTLERY CIRCUS PARADE	4	2011-12-09 12:50:00	4.15	12680.0	France
522063	581587	BAKING SET 9 PIECE RETROSPOT	3	2011-12-09 12:50:00	4.95	12680.0	France

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 522064 entries, 0 to 522063
Data columns (total 7 columns):
 #   Column      Non-Null Count   Dtype         
---  ------      --------------   -----         
 0   BillNo      522064 non-null  object        
 1   Itemname    520609 non-null  object        
 2   Quantity    522064 non-null  int64         
 3   Date        522064 non-null  datetime64[ns]
 4   Price       522064 non-null  float64       
 5   CustomerID  388023 non-null  float64       
 6   Country     522064 non-null  object        
dtypes: datetime64[ns](1), float64(2), int64(1), object(3)
memory usage: 27.9+ MB
```

```
data.shape
```

```
(522064, 7)
```

```
data.describe()
```

	Quantity	Price	CustomerID
count	522064.000000	522064.000000	388023.000000
mean	10.090435	3.826801	15316.931710
std	161.110525	41.900599	1721.846964
min	-9600.000000	-11062.060000	12346.000000
25%	1.000000	1.250000	13950.000000
50%	3.000000	2.080000	15265.000000
75%	10.000000	4.130000	16837.000000

```
# check for duplicate entries
data.duplicated().sum()
```

```
5286
```

```
# there are 5286 duplicates transcations are present in the dataset Lets remove them
data.drop_duplicates(inplace=True)
```

```
#Let remove the space in that word
data['Itemname'] = data['Itemname'].str.strip()
```

```
#Lets Check for null Values
data.isnull().sum()
```

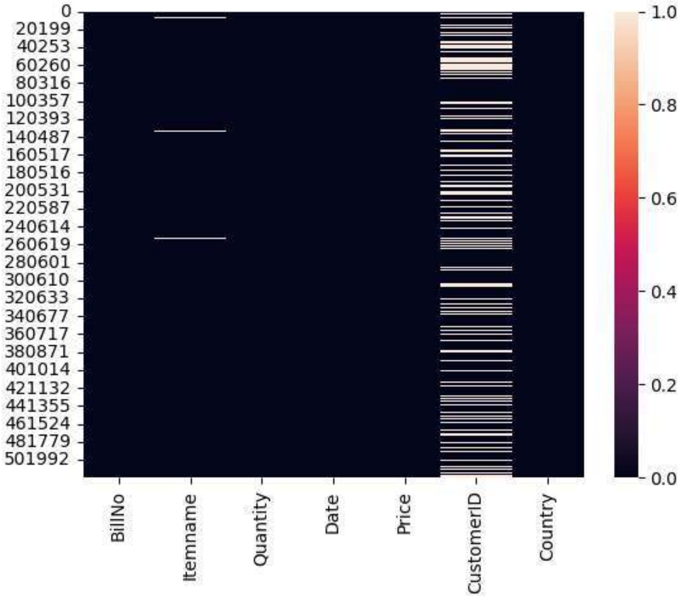
```
BillNo             0
Itemname        1455
Quantity           0
Date               0
Price              0
CustomerID    133967
Country            0
dtype: int64
```

```
data.isnull().mean()*100
```

```
BillNo         0.000000
Itemname       0.281552
Quantity       0.000000
Date           0.000000
Price          0.000000
CustomerID    25.923511
Country        0.000000
dtype: float64
```

```
sns.heatmap(data.isnull())
```

```
<Axes: >
```



```
#we can spearate the Data and time to different columns
```

```
import datetime as datetime
from datetime import datetime

#datetime.strptime('2013-01-01 09:10:12', '%Y-%m-%d %H:%M:%S')
data['date'] = data['Date'].dt.date
data['hour'] = data['Date'].dt.hour

### Converting invoice date to data time
data['date']= pd.to_datetime(data['date'], infer_datetime_format= True)
data.drop('Date',inplace=True,axis=1)

data.head(3)
```

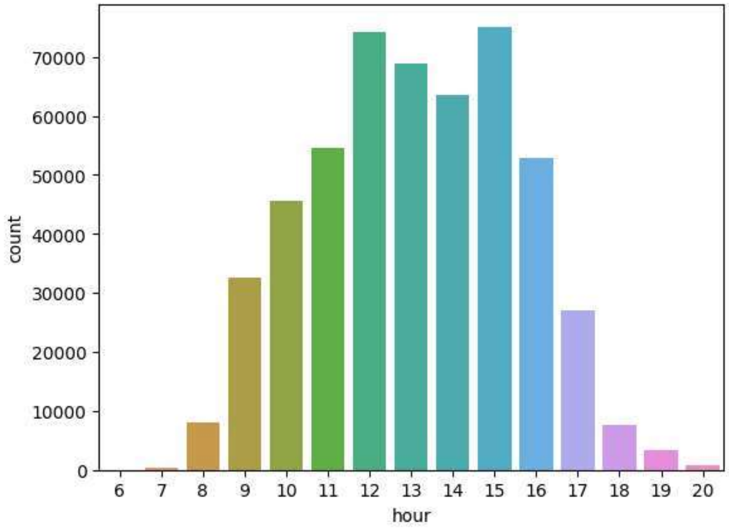
	BillNo	Itemname	Quantity	Price	CustomerID	Country	date	hour
0	536365	WHITE HANGING HEART T-LIGHT HOLDER	6	2.55	17850.0	United Kingdom	2010-12-01	8
1	536365	WHITE METAL LANTERN	6	3.39	17850.0	United Kingdom	2010-12-01	8
2	536365	CREAM CUPID HEARTS COAT HANGER	8	2.75	17850.0	United Kingdom	2010-12-01	8

```
#remove the rows which has the buyed quality is small or equal to zero
data=data[data['Quantity']>0]

#remove the rows which price is small or equal to zero
data=data[data['Price']>0]
data.shape
```

```
    (514270, 8)
```

```
#lets do some viz

sns.countplot(data=data,x='hour')
```

```
    <Axes: xlabel='hour', ylabel='count'>
```

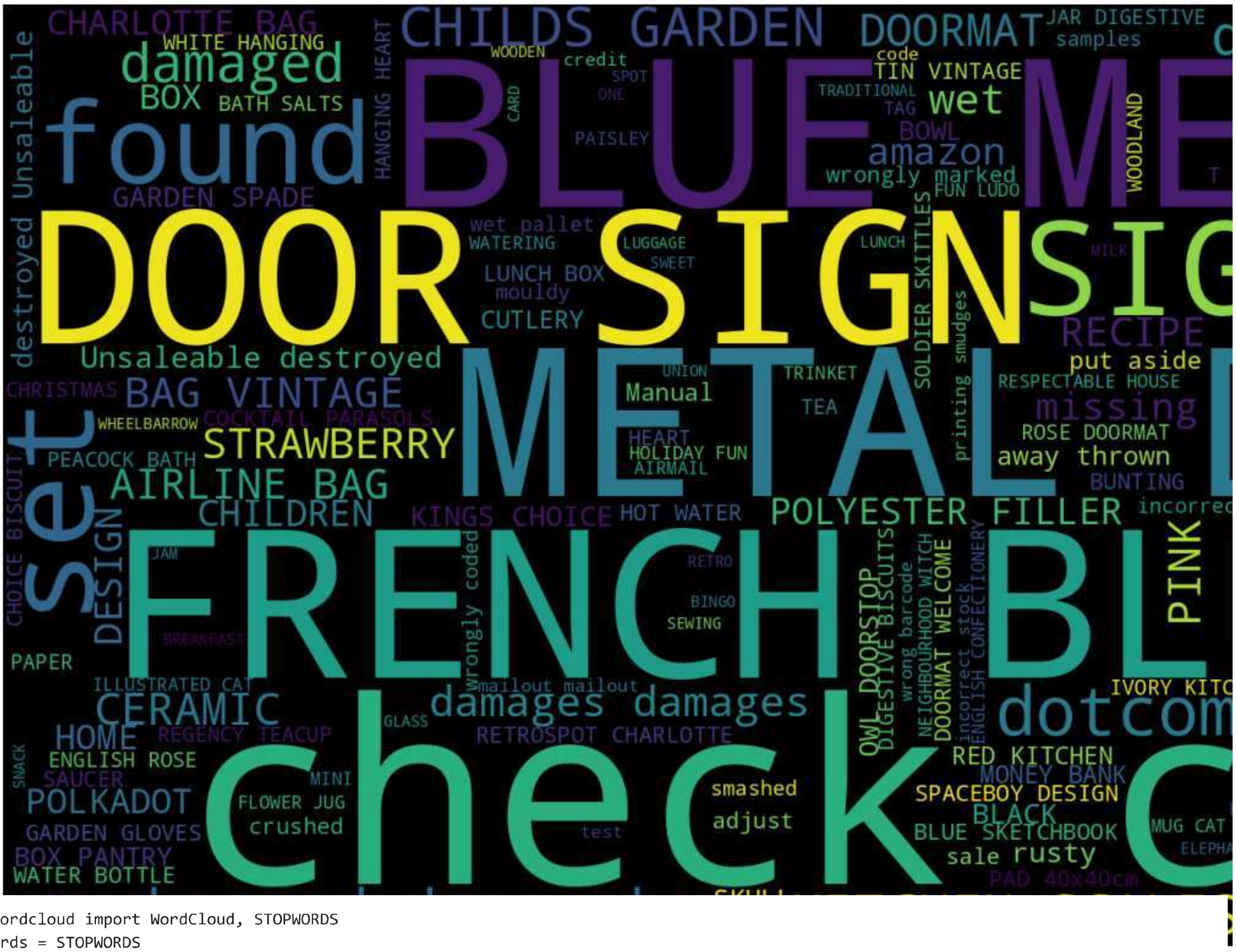


```
data[data['BillNo']==536527]
```

	BillNo	Itemname	Quantity	Price	CustomerID	Country	date	hour
1099	536527	SET OF 6 T-LIGHTS SANTA	6	2.95	12662.0	Germany	2010-12-01	13
1100	536527	ROTATING SILVER ANGELS T-LIGHT HLDR	6	2.55	12662.0	Germany	2010-12-01	13
1101	536527	MULTI COLOUR SILVER T-LIGHT HOLDER	12	0.85	12662.0	Germany	2010-12-01	13

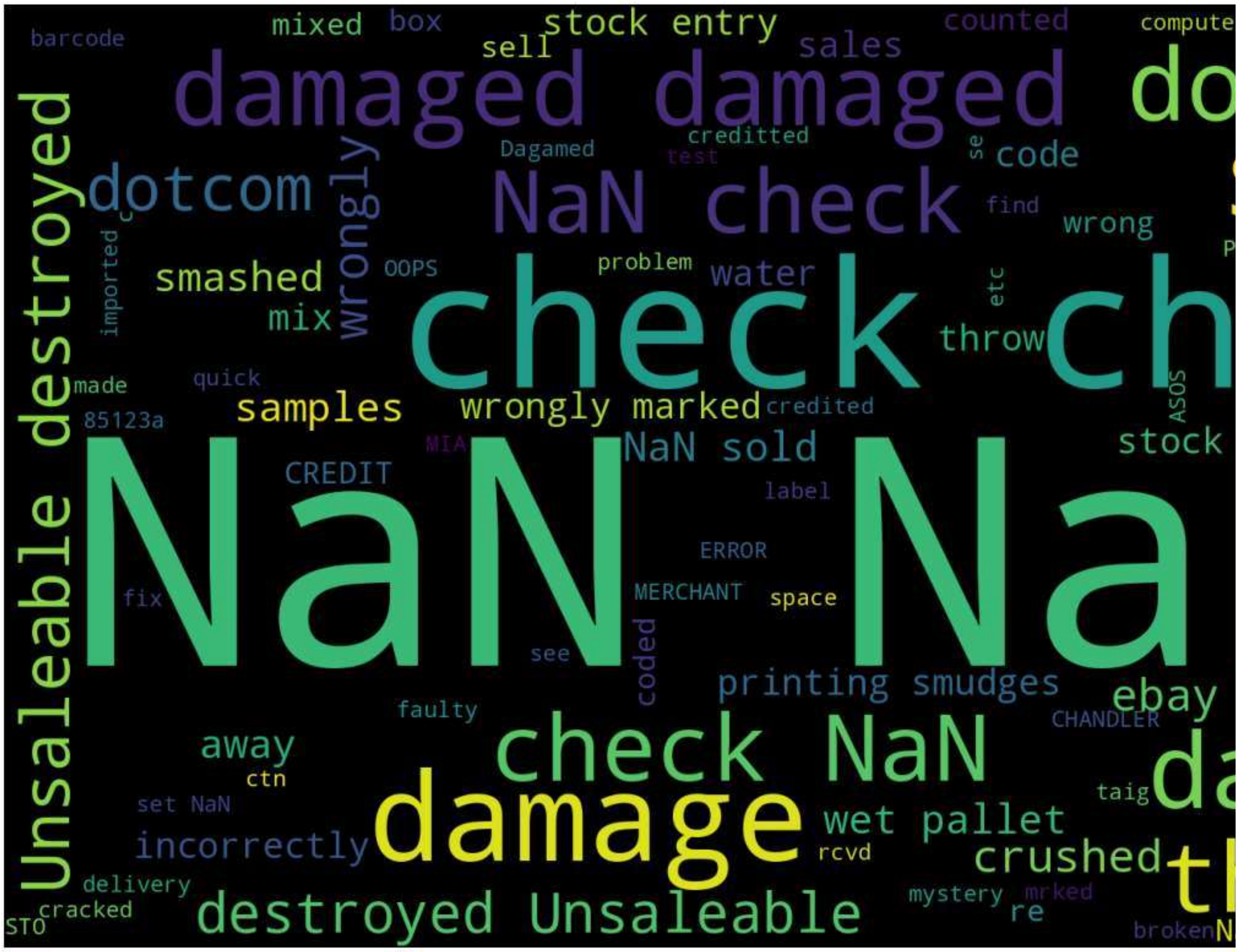
```
#next we can see for price small or equal to 0
temp=data[data['Price']<=0]
body =temp['Itemname'].dropna().to_string(index=False)
### Generate word cloud
worldcloud.generate(body)
## Visualize
plt.figure(figsize=(22,10))
plt.imshow(worldcloud)
plt.axis("off")
```

```
(-0.5, 1999.5, 999.5, -0.5)
```



```
from wordcloud import WordCloud, STOPWORDS
stopwords = STOPWORDS
worldcloud= WordCloud(background_color='Black',stopwords=stopwords, height=1000, width =2000)

temp=data[data['Quantity']<0]
body =temp['Itemname'].to_string(index=False)
### Generate word cloud
worldcloud.generate(body)
## Visualize
plt.figure(figsize=(22,10))
plt.imshow(worldcloud)
plt.axis("off")
```

```
(-0.5, 1999.5, 999.5, -0.5)
```


barcode
mixed
box
stock entry
counted
compute
sell
sales
damaged damaged
do
Unsaleable destroyed
creditted
Dagamed
test
se
code
dotcom
wrongly
NaN check
find
wrong
imported
c
smashed
OOPS
problem
water
P
check
ch
mix
etc
throw
made
quick
samples
wrongly marked
credited
ASOS
85123a
MIA
NaN sold
stock
NaN
Na
CREDIT
label
ERROR
fix
MERCHANT
space
see
coded
printing smudges
ebay
faulty
CHANDLER
check NaN
away
ctn
damage
da
set NaN
taig
wet pallet
incorrectly
rcvd
crushed
th
delivery
mystery
mrked
cracked
destroyed Unsaleable
re
STO
broken
N