Importing Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
```

About Data ¶

Customers are asked to rate the store out of 10 i.e. **"satisfaction" point** given by the customers to the store. Using an algorithm, store gives **"Loyalty points"** to customers based on the no. of items purchased by them in the last year + the amount of money spent by them in the store.

Loading Data

In [2]:
```
data = pd.read_csv(r'C:\Users\vamsi\Desktop\ML\15.DB Scan Clustering\market_data.
```

Data Exploration

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

Out[3]:

	Satisfaction	Loyalty
0	4	-1.33
1	6	-0.28
2	5	-0.99
3	7	-0.29
4	4	1.06

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Satisfaction  30 non-null     int64  
 1   Loyalty       30 non-null     float64
dtypes: float64(1), int64(1)
memory usage: 608.0 bytes
```

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

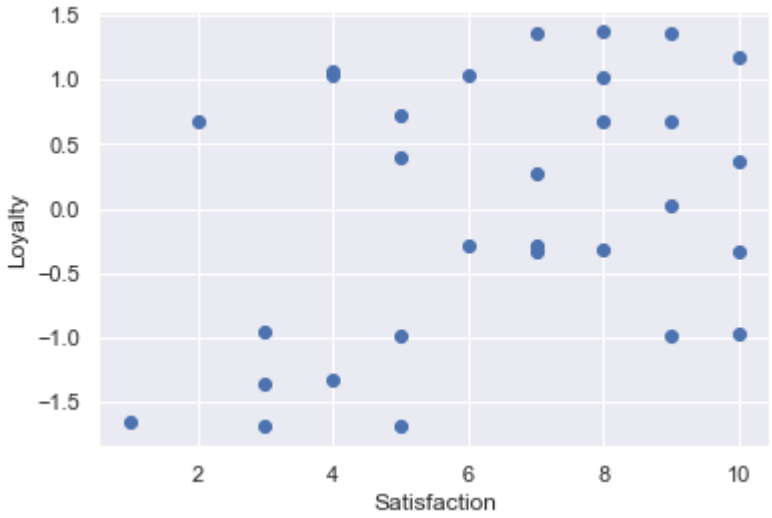
Out[5]:

	Satisfaction	Loyalty
count	30.000000	30.000000
mean	6.400000	0.001000
std	2.620871	1.016476
min	1.000000	-1.690000
25%	4.250000	-0.967500
50%	7.000000	0.150000
75%	8.750000	0.947500
max	10.000000	1.380000

Let's plot

In [6]:
```
plt.scatter(data['Satisfaction'], data['Loyalty'])
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[6]:
```
Text(0, 0.5, 'Loyalty')
```



Preparing input data

In [7]:
```
from sklearn import preprocessing
data_scaled = data.copy()
data_scaled = preprocessing.scale(data)
data_scaled
```

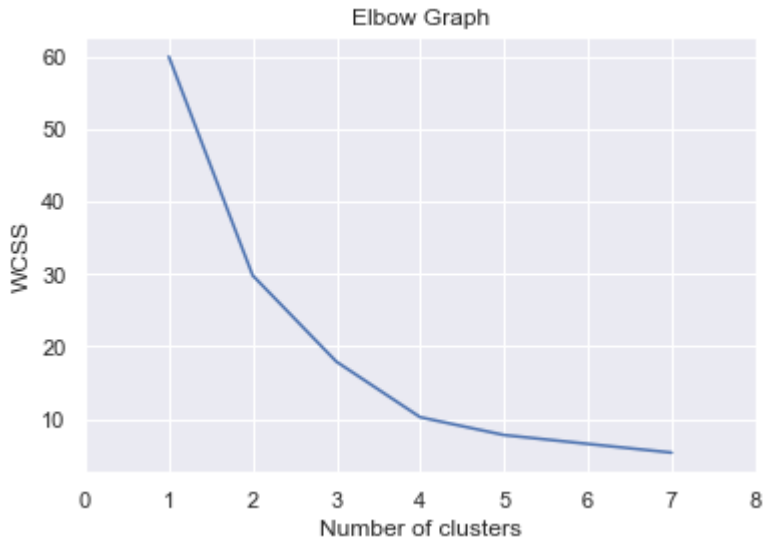
Out[7]:
```
array([[-0.93138063, -1.3318111 ],
       [-0.15523011, -0.28117124],
       [-0.54330537, -0.99160391],
       [ 0.23284516, -0.29117733],
       [-0.93138063,  1.05964534],
       [-2.09560642, -1.6620122 ],
       [ 1.39707095, -0.97159172],
       [ 0.62092042, -0.32119561],
       [ 0.62092042,  1.01962097],
       [ 0.62092042,  0.67941378],
       [ 1.39707095, -0.3412078 ],
       [-0.54330537,  0.38923705],
       [-0.54330537, -1.69203048],
       [-1.70753116,  0.66940768],
       [ 0.23284516,  0.26916393],
       [ 1.00899568,  1.35982816],
       [ 0.62092042,  1.37984035],
       [ 0.23284516,  1.35982816],
       [ 0.23284516, -0.3412078 ],
       [ 1.00899568,  0.66940768],
       [ 1.39707095,  1.17971847],
       [-1.31945589, -1.69203048],
       [-0.93138063,  1.03963316],
       [-1.31945589, -0.96158562],
       [-0.15523011,  1.02962706],
       [ 1.00899568, -0.99160391],
       [ 1.39707095,  0.36922486],
       [ 1.00899568,  0.02901767],
       [-1.31945589, -1.36182938],
       [-0.54330537,  0.72944425]])
```

Elbow Method

In [8]:
```
from sklearn.cluster import KMeans

wcss = []
for i in range (1, 8):
  kmeans = KMeans(n_clusters = i, random_state = 0)
  kmeans.fit(data_scaled)
  wcss.append(kmeans.inertia_)
```

In [9]:
```
plt.plot(range(1, 8), wcss)
plt.xlim( 0, 8)
plt.title('Elbow Graph')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



K-Means

Training Model

In [10]:
```
from sklearn.cluster import DBSCAN

dbscan_cluster = DBSCAN(min_samples = 2)
dbscan_cluster.fit(data_scaled)
```

Out[10]:
```
DBSCAN(min_samples=2)
```

In [11]:
```
cluster_data= data.copy()
cluster_data['cluster_pred'] = dbscan_cluster.fit_predict(data_scaled)
```

Clustering Labels

In [12]:
```
dbscan_cluster.labels_
```

Out[12]:
```
array([ 0,  1, -1,  1,  2, -1,  3,  1,  4,  4, -1,  2, -1, -1, -1,  4,  4,
        4,  1,  4,  4,  0,  2,  0,  2,  3,  4, -1,  0,  2], dtype=int64)
```

In [13]:
```
n_clusters_ = len(np.unique(dbscan_cluster.labels_))
n_clusters_
```

Out[13]:
```
6
```

In [14]:
```
cluster_data
```

Out[14]:

	Satisfaction	Loyalty	cluster_pred
0	4	-1.33	0
1	6	-0.28	1
2	5	-0.99	-1
3	7	-0.29	1
4	4	1.06	2
5	1	-1.66	-1
6	10	-0.97	3
7	8	-0.32	1
8	8	1.02	4
9	8	0.68	4
10	10	-0.34	-1
11	5	0.39	2
12	5	-1.69	-1
13	2	0.67	-1
14	7	0.27	-1
15	9	1.36	4
16	8	1.38	4
17	7	1.36	4
18	7	-0.34	1
19	9	0.67	4
20	10	1.18	4
21	3	-1.69	0
22	4	1.04	2
23	3	-0.96	0
24	6	1.03	2
25	9	-0.99	3
26	10	0.37	4
27	9	0.03	-1
28	3	-1.36	0
29	5	0.73	2

Visualisation

In [15]:
```
plt.scatter(data['Satisfaction'], data['Loyalty'], c = cluster_data['cluster_pred
plt.xlabel('Satisfaction')
plt.ylabel('Loyalty')
```

Out[15]:
```
Text(0, 0.5, 'Loyalty')
```