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

In [4]:
```
df = pd.read_csv('sales_data_sample.csv',encoding='ISO-8859-1')
```

In [5]:
```
df.head()
```

Out[5]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORDER
0	10107	30	95.70	2	2871.00	2/2
1	10121	34	81.35	5	2765.90	5/7/200
2	10134	41	94.74	2	3884.34	7/1/200
3	10145	45	83.26	6	3746.70	8/2
4	10159	49	100.00	14	5205.27	10/1

5 rows × 25 columns

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

Out[6]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORI
2818	10350	20	100.00	15	2244.40	
2819	10373	29	100.00	1	3978.51	
2820	10386	43	100.00	4	5417.57	3/1/
2821	10397	34	62.24	1	2116.16	
2822	10414	47	65.52	9	3079.44	5/6/

5 rows × 25 columns

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   ORDERNUMBER       2823 non-null   int64  
 1   QUANTITYORDERED   2823 non-null   int64  
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64  
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object 
 6   STATUS            2823 non-null   object 
 7   QTR_ID            2823 non-null   int64  
 8   MONTH_ID          2823 non-null   int64  
 9   YEAR_ID           2823 non-null   int64  
 10  PRODUCTLINE       2823 non-null   object 
 11  MSRP              2823 non-null   int64  
 12  PRODUCTCODE       2823 non-null   object 
 13  CUSTOMERNAME      2823 non-null   object 
 14  PHONE             2823 non-null   object 
 15  ADDRESSLINE1      2823 non-null   object 
 16  ADDRESSLINE2      302 non-null    object 
 17  CITY              2823 non-null   object 
 18  STATE             1337 non-null   object 
 19  POSTALCODE        2747 non-null   object 
 20  COUNTRY           2823 non-null   object 
 21  TERRITORY         1749 non-null   object 
 22  CONTACTLASTNAME   2823 non-null   object 
 23  CONTACTFIRSTNAME  2823 non-null   object 
 24  DEALSIZE          2823 non-null   object 
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

In [8]:
```
df.isnull()
```

Out[8]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORD
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
...	...	...	...	...	...	
2818	False	False	False	False	False	
2819	False	False	False	False	False	
2820	False	False	False	False	False	
2821	False	False	False	False	False	
2822	False	False	False	False	False	

2823 rows × 25 columns

In [9]:
```
df.dropna()
```

Out[9]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALES	ORI
10	10223	37	100.00	1	3965.66	
21	10361	20	72.55	13	1451.00	1
40	10270	21	100.00	9	4905.39	
47	10347	30	100.00	1	3944.70	1
51	10391	24	100.00	4	2416.56	3/9/
...	...	...	...	...	...	
2667	10120	43	76.00	14	3268.00	
2673	10223	26	67.20	15	1747.20	
2685	10361	44	100.00	10	5001.92	1
2764	10361	35	100.00	11	4277.35	1
2791	10361	23	95.20	12	2189.60	1

147 rows × 25 columns

In [10]:
```
df.describe()
```

Out[10]:

	ORDERNUMBER	QUANTITYORDERED	PRICEEACH	ORDERLINENUMBER	SALE
count	2823.000000	2823.000000	2823.000000	2823.000000	2823.00000
mean	10258.725115	35.092809	83.658544	6.466171	3553.88907
std	92.085478	9.741443	20.174277	4.225841	1841.86510
min	10100.000000	6.000000	26.880000	1.000000	482.13000
25%	10180.000000	27.000000	68.860000	3.000000	2203.43000
50%	10262.000000	35.000000	95.700000	6.000000	3184.80000
75%	10333.500000	43.000000	100.000000	9.000000	4508.00000
max	10425.000000	97.000000	100.000000	18.000000	14082.80000

In [11]:
```
df.shape
```

Out[11]:
```
(2823, 25)
```

In [12]:
```
df.corr()
```

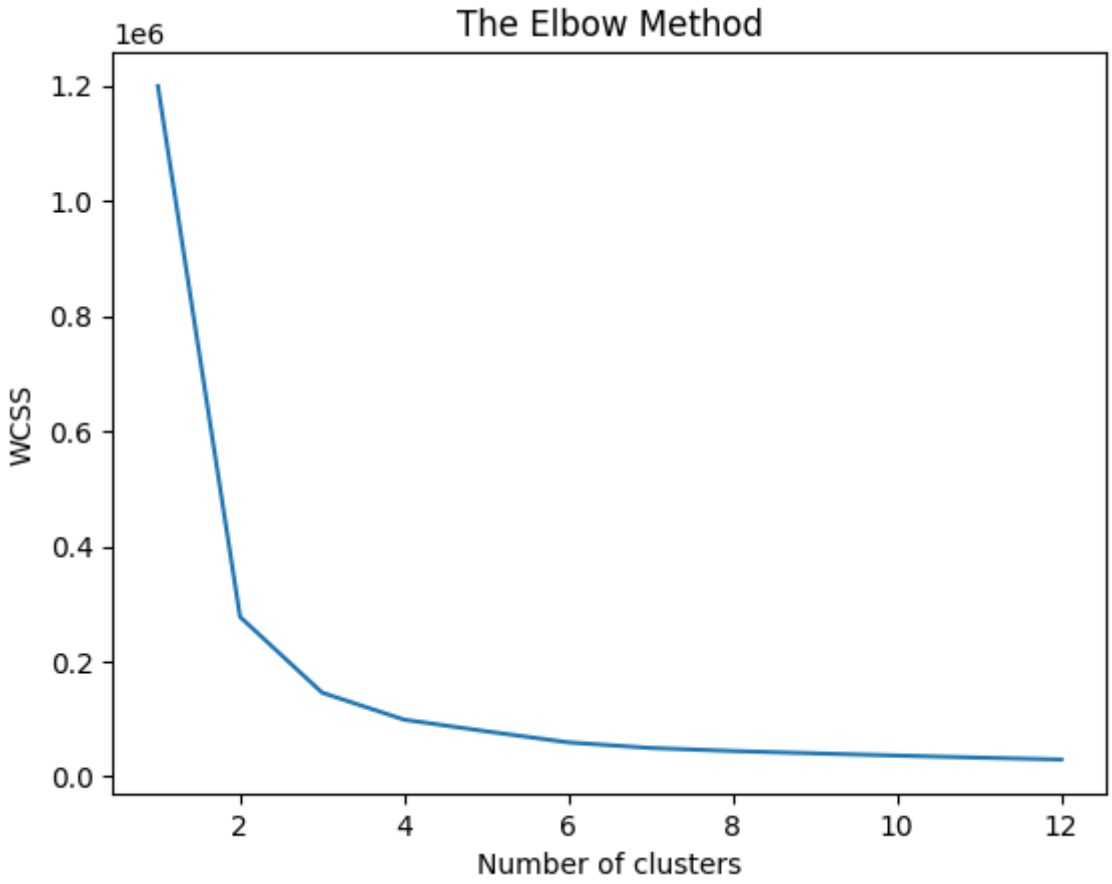
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[12], line 1
----> 1 df.corr()

File ~\Desktop\New folder (3)\lib\site-packages\pandas\core\frame.py:10059, in DataFrame.corr(self, method, min_periods, numeric_only)
  10057 cols = data.columns
  10058 idx = cols.copy()
> 10059 mat = data.to_numpy(dtype=float, na_value=np.nan, copy=False)
  10061 if method == "pearson":
  10062     correl = libalgos.nancorr(mat, minp=min_periods)

File ~\Desktop\New folder (3)\lib\site-packages\pandas\core\frame.py:1838, in DataFrame.to_numpy(self, dtype, copy, na_value)
   1836 if dtype is not None:
   1837     dtype = np.dtype(dtype)
-> 1838 result = self._mgr.as_array(dtype=dtype, copy=copy, na_value=na_value)
   1839 if result.dtype is not dtype:
   1840     result = np.array(result, dtype=dtype, copy=False)

File ~\Desktop\New folder (3)\lib\site-packages\pandas\core\internals\managers.py:1732, in BlockManager.as_array(self, dtype, copy, na_value)
   1730         arr.flags.writeable = False
   1731 else:
-> 1732     arr = self._interleave(dtype=dtype, na_value=na_value)
   1733     # The underlying data was copied within _interleave, so no need
   1734     # to further copy if copy=True or setting na_value
   1736 if na_value is not lib.no_default:

File ~\Desktop\New folder (3)\lib\site-packages\pandas\core\internals\managers.py:1794, in BlockManager._interleave(self, dtype, na_value)
   1792     else:
   1793         arr = blk.get_values(dtype)
-> 1794     result[rl.indexer] = arr
   1795     itemmask[rl.indexer] = 1
   1797 if not itemmask.all():

ValueError: could not convert string to float: '2/24/2003 0:00'
```

In [13]:
```
x=df.iloc[:,2:4].values
```

In [17]:

```
from sklearn.cluster import KMeans
wcss=[]
for i in range(1,13):
 kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
 kmeans.fit(x)
 wcss.append(kmeans.inertia_)
plt.plot(range(1, 13), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

```
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
```

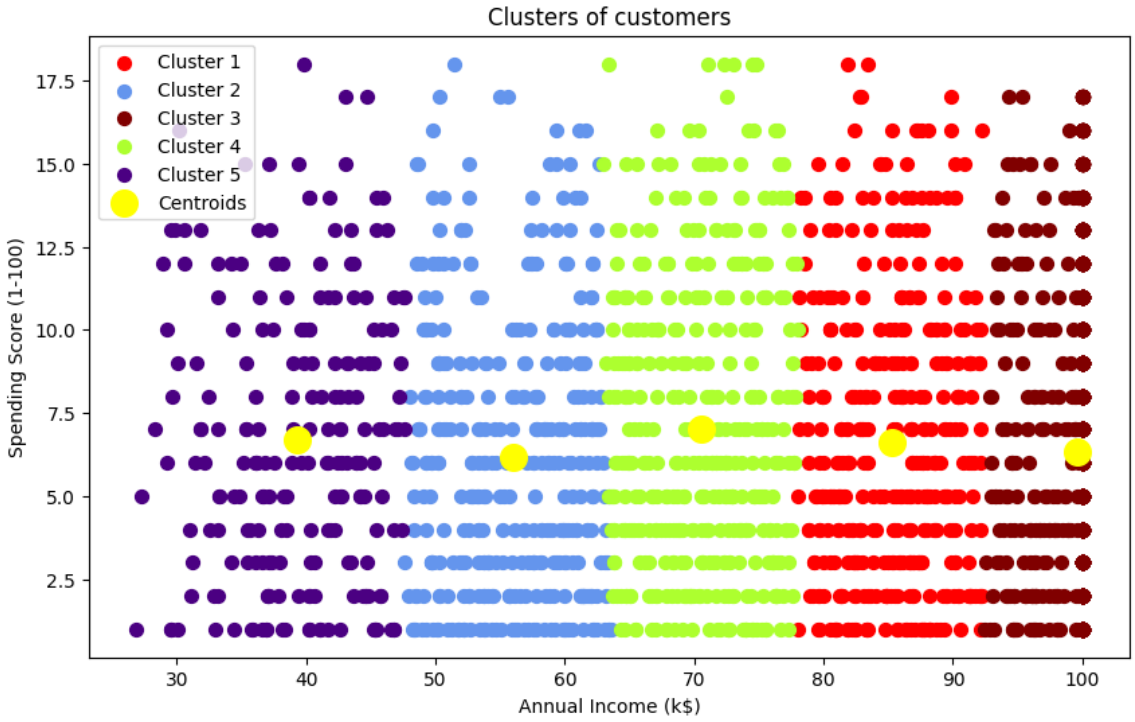


In [18]:
```
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = kmeans.fit_predict(x)
y_kmeans
```

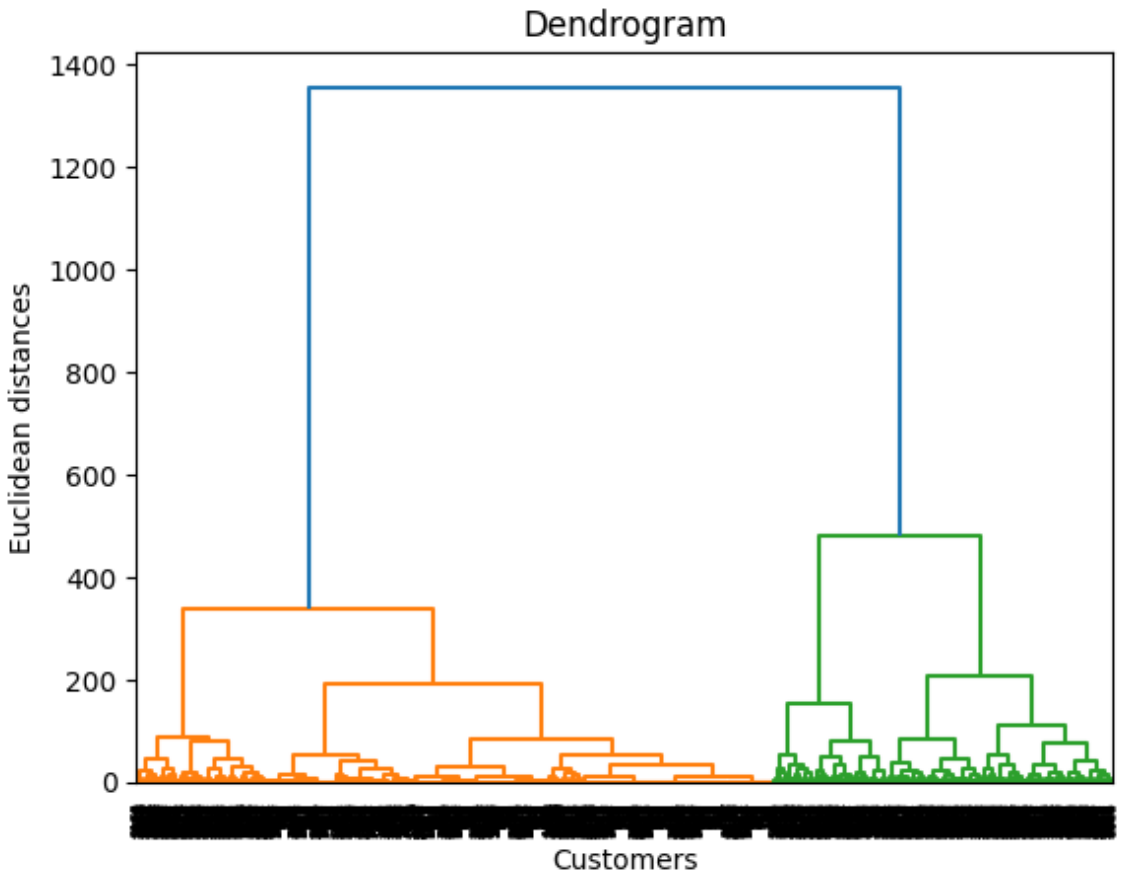
```
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
om 10 to 'auto' in 1.4. Set the value of `n_init` explicitly to suppress t
he warning
  warnings.warn(
```

Out[18]:
```
array([2, 0, 2, ..., 2, 1, 3])
```

In [20]:
```
fig, (ax1) = plt.subplots(1, figsize=(10, 6))
ax1.scatter(x[y_kmeans == 0, 0], x[y_kmeans == 0, 1], s=50, c='red', label=
ax1.scatter(x[y_kmeans == 1, 0], x[y_kmeans == 1, 1], s=50, c='cornflowerblu
ax1.scatter(x[y_kmeans == 2, 0], x[y_kmeans == 2, 1], s=50, c='maroon', labe
ax1.scatter(x[y_kmeans == 3, 0], x[y_kmeans == 3, 1], s=50, c='greenyellow'
ax1.scatter(x[y_kmeans == 4, 0], x[y_kmeans == 4, 1], s=50, c='indigo', labe
ax1.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



In [21]:
```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(x, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```



In [23]:
```
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkag
y_hc = hc.fit_predict(x)
```

```
C:\Users\satyam\Desktop\New folder (3)\lib\site-packages\sklearn\cluster\_
agglomerative.py:983: FutureWarning: Attribute `affinity` was deprecated i
n version 1.2 and will be removed in 1.4. Use `metric` instead
  warnings.warn(
```

In [25]:
```
fig, ax1 = plt.subplots(1, figsize=(10, 6))
ax1.scatter(x[y_hc == 0, 0], x[y_hc == 0, 1], s=80, c='gold', label='Cluste
ax1.scatter(x[y_hc == 1, 0], x[y_hc == 1, 1], s=80, c='crimson', label='Clu
ax1.scatter(x[y_hc == 2, 0], x[y_hc == 2, 1], s=80, c='green', label='Clust
ax1.scatter(x[y_hc == 3, 0], x[y_hc == 3, 1], s=80, c='orangered', label='C
ax1.scatter(x[y_hc == 4, 0], x[y_hc == 4, 1], s=80, c='navy', label='Cluste
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```

In [ ]: