Hierachical clustering(Dendogram) Agglouramative clustering

In [11]:

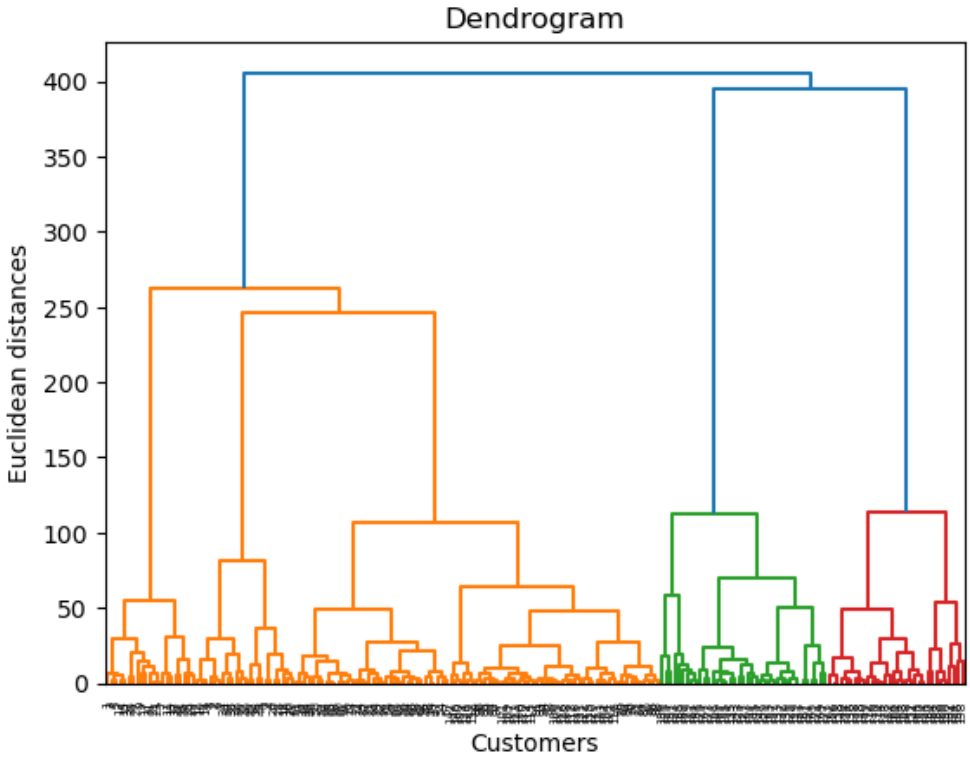
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [12]:

```
dataset = pd.read_csv('/Users/myyntiimac/Desktop/Mall_Customers.csv')
X = dataset.iloc[:, [3, 4]].values
```

In [13]:

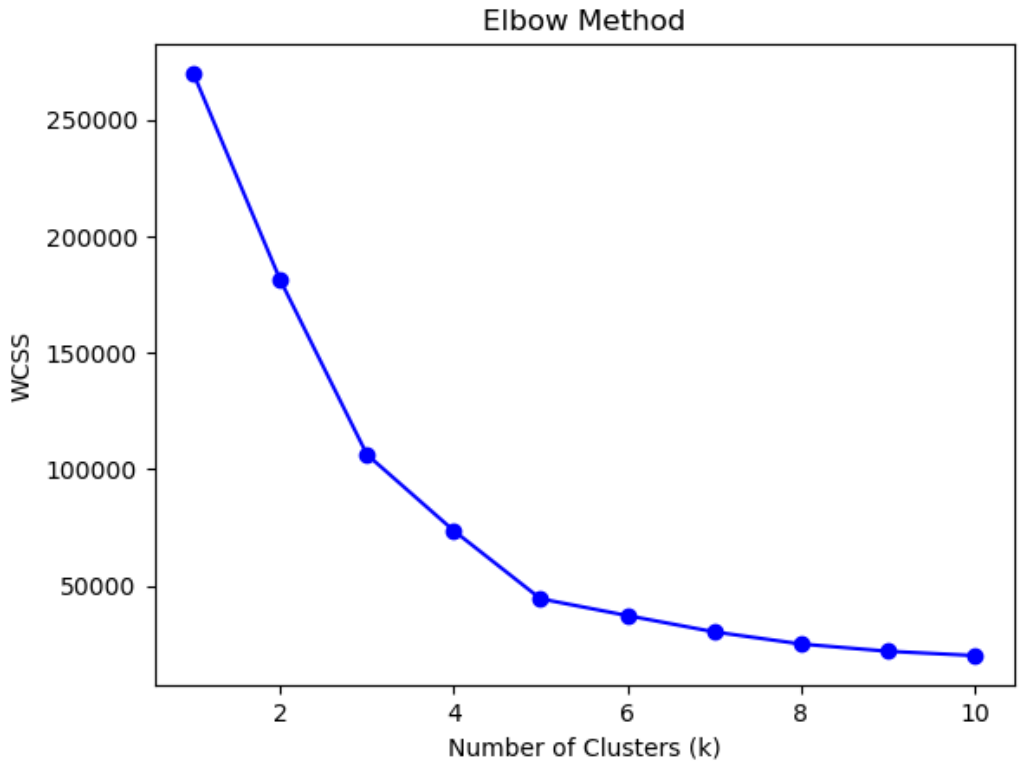
```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.xlabel('Customers')
plt.ylabel('Euclidean distances')
plt.show()
```




```
#to determine cut level or trushold , implement elbow method
Elbow method: Calculate a measure of dissimilarity (e.g., within-cluster sum of squares,
WCSS) for different cut levels. Plot the dissimilarity measure against the number of
clusters. Look for a point where the rate of decrease in dissimilarity slows down
significantly (forming an "elbow" shape). This point can be considered as a potential cut
level.
```

In [14]:

```
import warnings

# Ignore all warnings
warnings.filterwarnings("ignore")
from sklearn.cluster import KMeans

# Assuming you have your data stored in 'X'
# X should be a 2D array or matrix with shape (n_samples, n_features)

# Initialize an empty list to store the WCSS values for different numbers of clusters
wcss = []

# Define the range of cluster numbers to try
k_values = range(1, 11)  # Try cluster numbers from 1 to 10

# Calculate WCSS for each cluster number
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)  # Inertia is the WCSS value

# Plot the WCSS values against the number of clusters
plt.plot(k_values, wcss, 'bo-')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('WCSS')
plt.title('Elbow Method')
plt.show()
```



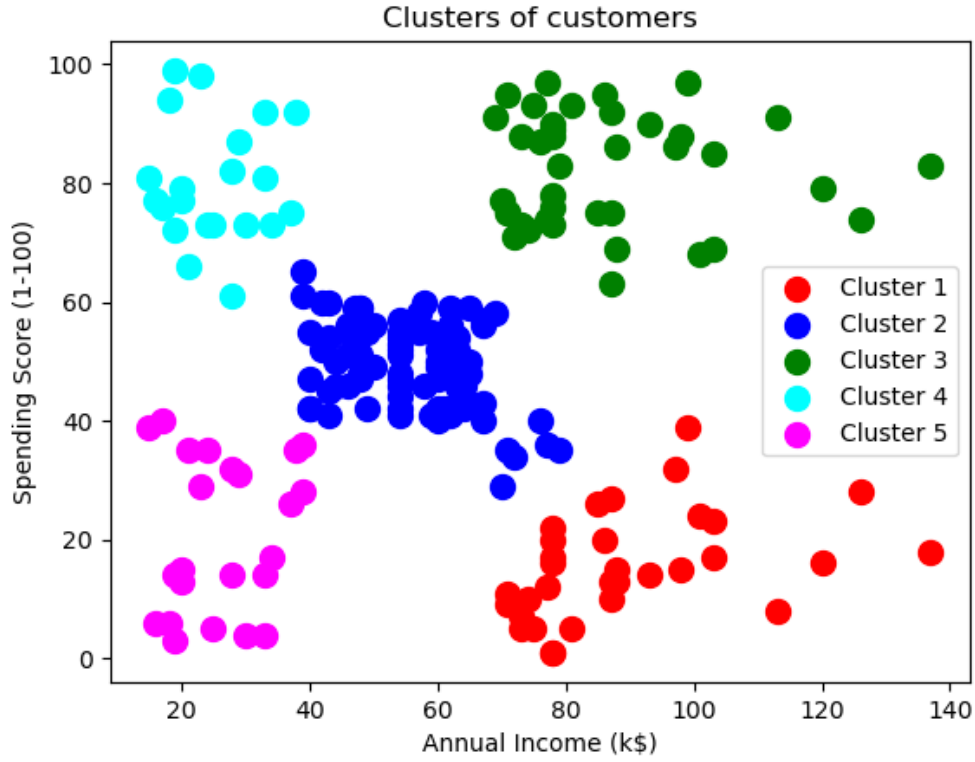

```
insight: from this figure , we assome cluster no is 5
```

In [15]:

```
#training  the data with aggluramative clustering
from sklearn.cluster import AgglomerativeClustering
hc = AgglomerativeClustering(n_clusters = 5, affinity = 'euclidean', linkage = 'ward')
y_hc = hc.fit_predict(X)
```

In [17]:

```
# Visualizing the clusters
plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s = 100, c = 'red', label = 'Cluster 1')
plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s = 100, c = 'blue', label = 'Cluster 2')
plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s = 100, c = 'green', label = 'Cluster 3')
plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



In [18]:

```
df1=pd.read_csv("/Users/myyntiimac/Desktop/modified_dataset.csv")
df1.head()
```

Out[18]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)	Cluster
0	1	Male	19	15	39	2
1	2	Male	21	15	81	3
2	3	Female	20	16	6	2
3	4	Female	23	16	77	3
4	5	Female	31	17	40	2

In [19]:

```
df1["agguluramative"] = y_hc
```

In [20]:

```
df1.head()
```

Out[20]:

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)	Cluster	agguluramative
0	1	Male	19	15	39	2	4
1	2	Male	21	15	81	3	3
2	3	Female	20	16	6	2	4
3	4	Female	23	16	77	3	3
4	5	Female	31	17	40	2	4

In [21]:

```
df1.to_csv('new_dataframe.csv', index=False)
```

In [22]:

```
import os

# Get the current working directory
current_dir = os.getcwd()

# Specify the filename
filename = 'new_dataframe.csv'

# Combine the directory and filename to get the full file path
file_path = os.path.join(current_dir, filename)

# Print the file path
print("File saved at:", file_path)
```

```
File saved at: /Users/myyntiimac/new_dataframe.csv
```

In [ ]:

```

```