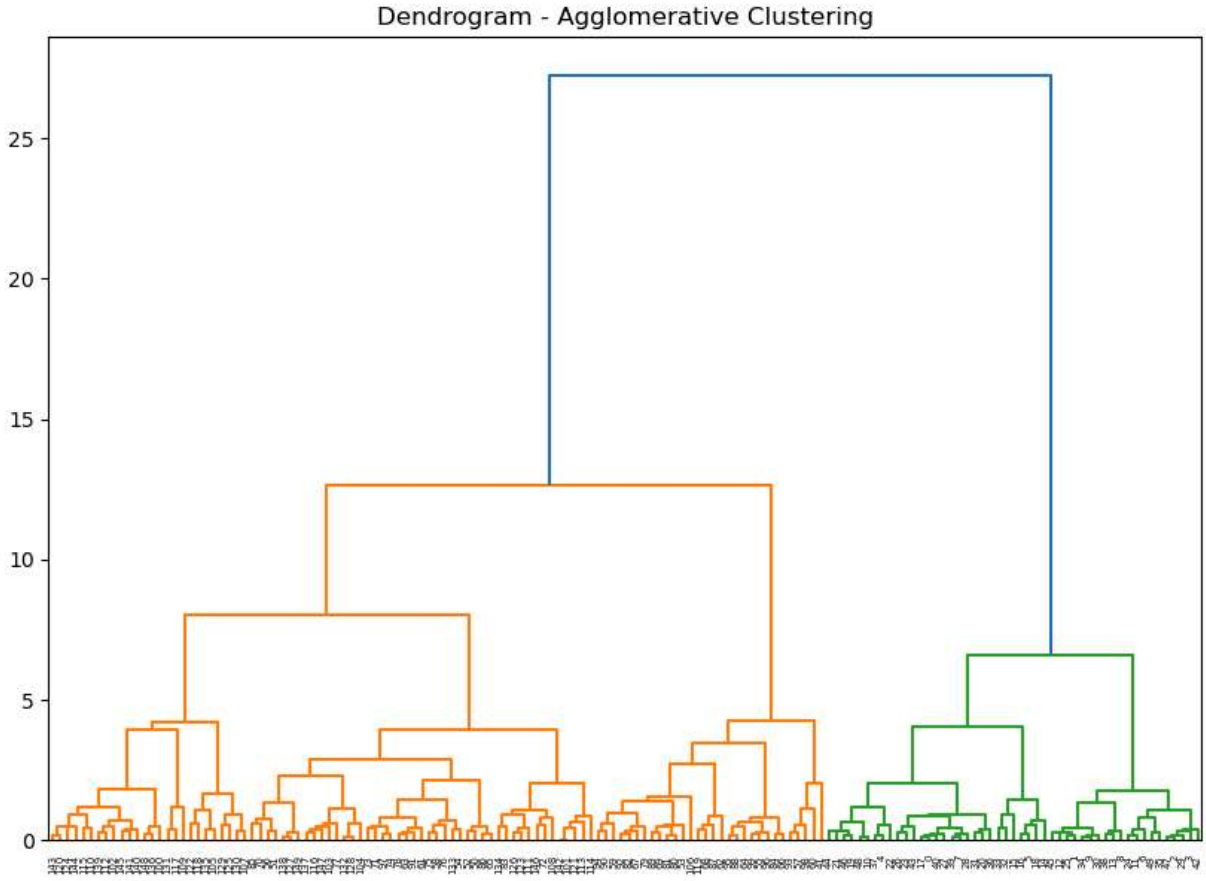
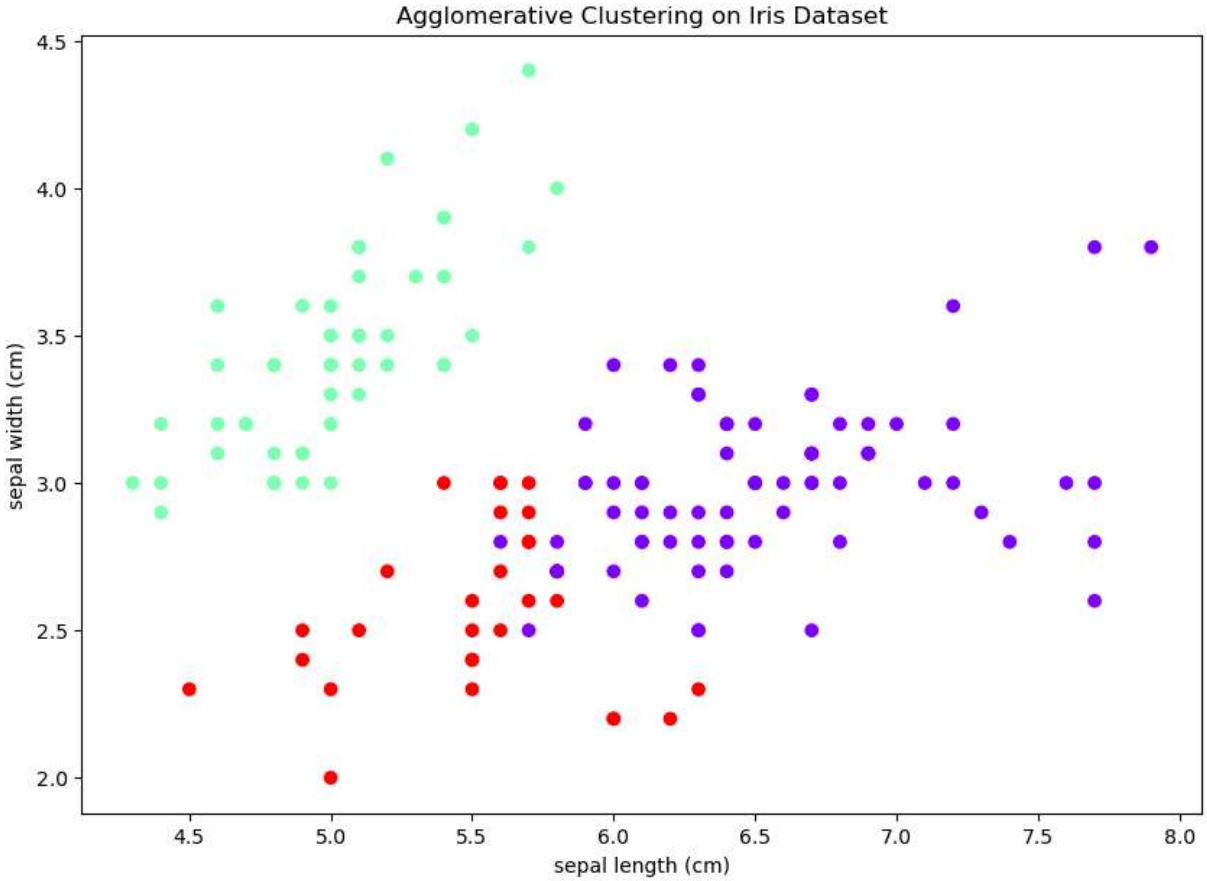
Implement Agglomerative hierarchical clustering algorithm using appropriate dataset

In [4]:

```
# Step 1: Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage

# Step 2: Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
df = pd.DataFrame(X, columns=iris.feature_names)

# Step 3: Preprocess the data by standardizing it
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Step 4: Visualize the dendrogram for hierarchical clustering
plt.figure(figsize=(10, 7))
plt.title("Dendrogram - Agglomerative Clustering")
linked = linkage(X_scaled, method='ward')
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=
plt.show()

# Step 5: Apply Agglomerative Clustering
agg_clustering = AgglomerativeClustering(n_clusters=3, linkage='ward')  # Removed a
cluster_labels = agg_clustering.fit_predict(X_scaled)

# Step 6: Add the cluster labels to the dataset
df['Cluster'] = cluster_labels

# Step 7: Visualize the clusters (for simplicity, we'll plot only two features)
plt.figure(figsize=(10, 7))
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c=df['Cluster'], cmap='rainbow', marker='
plt.title("Agglomerative Clustering on Iris Dataset")
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.show()

# Step 8: Display the cluster assignments
print(f"Cluster assignments: \n{df[['Cluster']].head()}")
```


Dendrogram - Agglomerative Clustering
25
20
15
10
5
0


Agglomerative Clustering on Iris Dataset
sepal width (cm)
sepal length (cm)
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

```
Cluster assignments:
   Cluster
0        1
1        1
2        1
3        1
4        1
```

In [ ]: