ASSIGNMENT 5

In [55]:
```
import pandas as pd
df=pd.read_csv(r"Mall_Customers.csv")
print(df)
```

```
     CustomerID   Genre  Age  Annual Income (k$)  Spending Score (1-100)
0             1    Male   19                  15                      39
1             2    Male   21                  15                      81
2             3  Female   20                  16                       6
3             4  Female   23                  16                      77
4             5  Female   31                  17                      40
..          ...     ...  ...                 ...                     ...
195         196  Female   35                 120                      79
196         197  Female   45                 126                      28
197         198    Male   32                 126                      74
198         199    Male   32                 137                      18
199         200    Male   30                 137                      83

[200 rows x 5 columns]
```

In [56]:
```
df.isnull().sum()
```

Out[56]:
```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

In [57]:
```
df.dtypes
```

Out[57]:
```
CustomerID                 int64
Genre                     object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
```

In [58]:
```
df.columns
```

Out[58]:
```
Index(['CustomerID', 'Genre', 'Age', 'Annual Income (k$)',
       'Spending Score (1-100)'],
      dtype='object')
```

Pre-processing

In [59]:
```
from sklearn.preprocessing import LabelEncoder

df=df.drop(columns=["CustomerID"],axis=1)

label_encoder=LabelEncoder()
df["Genre"]=label_encoder.fit_transform(df["Genre"])

print(df)
```

```
     Genre  Age  Annual Income (k$)  Spending Score (1-100)
0        1   19                  15                      39
1        1   21                  15                      81
2        0   20                  16                       6
3        0   23                  16                      77
4        0   31                  17                      40
..     ...  ...                 ...                     ...
195      0   35                 120                      79
196      0   45                 126                      28
197      1   32                 126                      74
198      1   32                 137                      18
199      1   30                 137                      83

[200 rows x 4 columns]
```

Splitting data

In [60]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
X = df[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']]
y = df['Genre']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


X_train,X_test,y_train,y_test=train_test_split(X_scaled,y,test_size=0.25,random_state=42)
```

Applying models

In [61]:
```
from sklearn.cluster import KMeans, AgglomerativeClustering

# K-Means Clustering
kmeans = KMeans(n_clusters=2, random_state=42,n_init=10)
kmeans_labels = kmeans.fit_predict(X_train)

# Agglomerative Hierarchical Clustering
agg_cluster = AgglomerativeClustering(n_clusters=2)
agg_labels = agg_cluster.fit_predict(X_train)
```

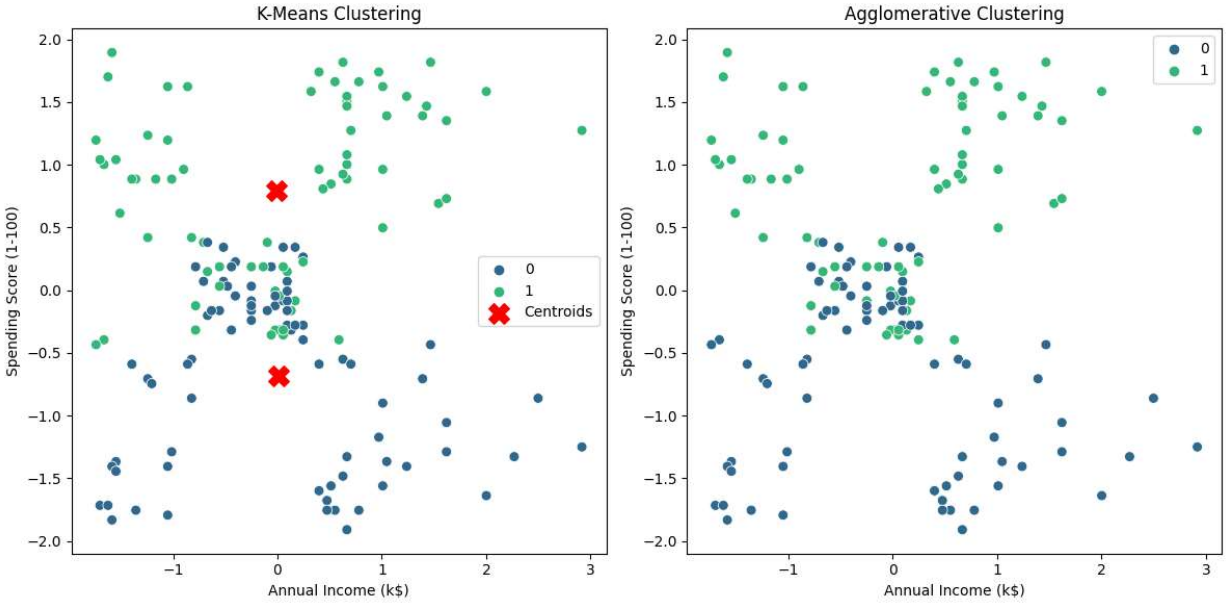
Evaluating models

In [62]:
```
from sklearn.metrics import silhouette_score

kmeans_silhouette = silhouette_score(X_train, kmeans_labels)
agg_silhouette = silhouette_score(X_train, agg_labels)

print(f"K-Means Silhouette Score: {kmeans_silhouette}")
print(f"Agglomerative Silhouette Score: {agg_silhouette}")
```

```
K-Means Silhouette Score: 0.32987964770995
Agglomerative Silhouette Score: 0.3180153806845089
```

Visualizing models

In [68]:
```
import numpy as np

# Visualize K-Means Clusters
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)

# Scatter plot for K-Means Clustering
sns.scatterplot(x=X_train[:, 1], y=X_train[:, 2], hue=kmeans_labels, palette='viridis', s=50)
plt.scatter(kmeans.cluster_centers_[:, 1], kmeans.cluster_centers_[:, 2], c='red', marker='X', s=200, label='Cen
plt.title('K-Means Clustering')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()

# Visualize Agglomerative Clusters
plt.subplot(1, 2, 2)

# Scatter plot for Agglomerative Clustering
sns.scatterplot(x=X_train[:, 1], y=X_train[:, 2], hue=agg_labels, palette='viridis', s=50)
plt.title('Agglomerative Clustering')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')

plt.tight_layout()
plt.show()

import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```


K-Means Clustering
Agglomerative Clustering
Spending Score (1-100)
Annual Income (k$)
0
1
Centroids

In [ ]: