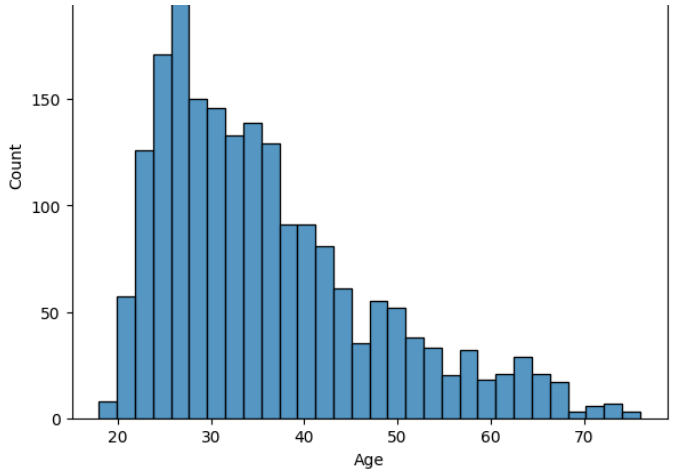
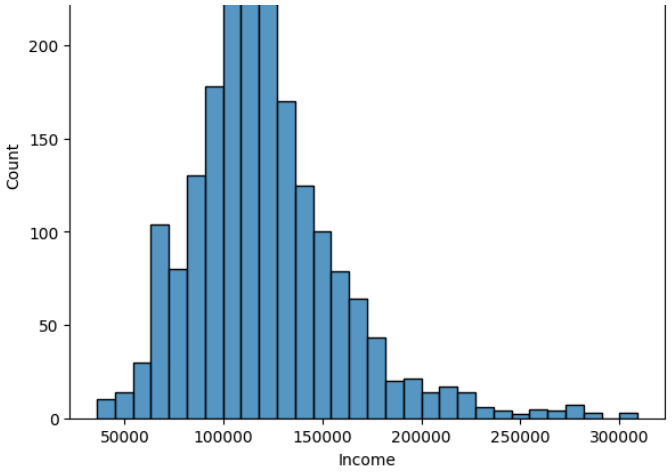
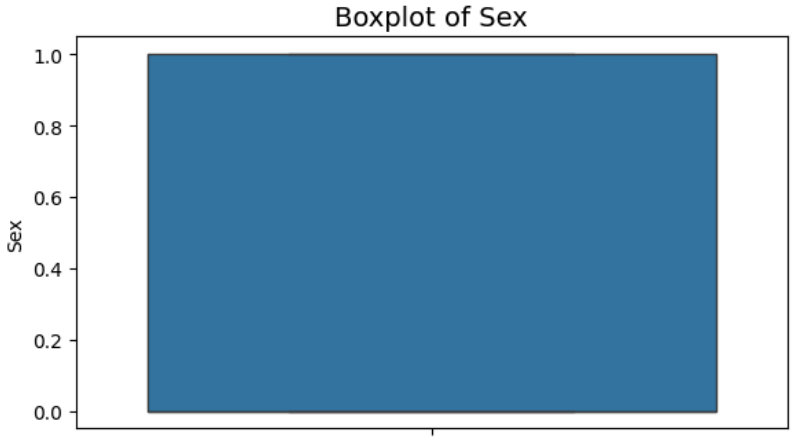
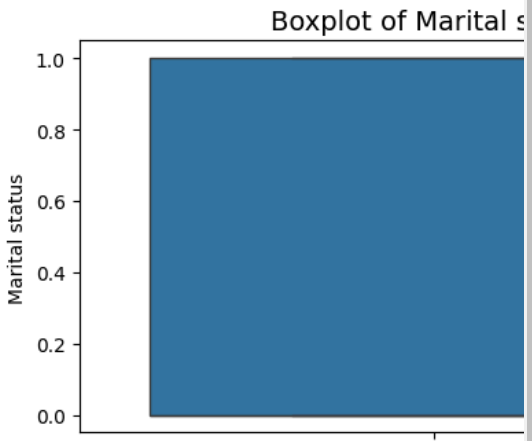
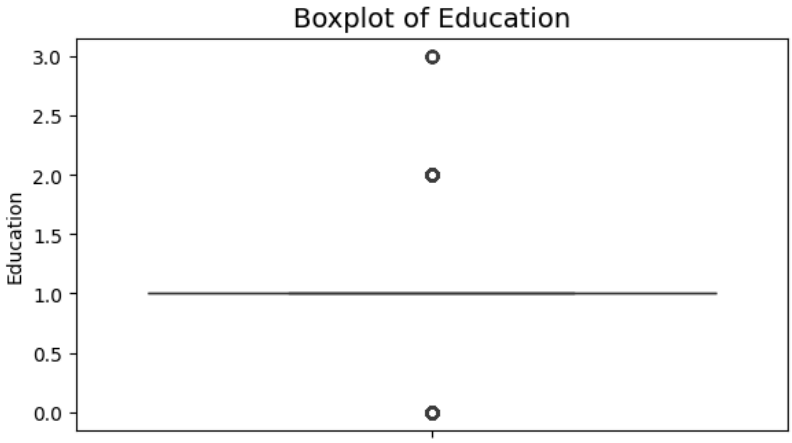
```
import pandas as pd

# Load the data excluding the 'ID' column
cust_data = pd.read_csv('segmentation data.csv', usecols=['Sex', 'Marital status', 'Age', 'Education', 'Income', 'Occupation', 'Settlem
cust_data.head()

df = cust_data



from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('segmentation data.csv')

df_cluster = df.drop('ID', axis=1)

# Checking for missing values
missing_values = df_cluster.isnull().sum()

# Checking data types
data_types = df_cluster.dtypes
descriptive_stats = df_cluster.describe()

# Checking for any missing values in the dataset
missing_data = df_cluster.isnull().sum()

# Plotting histograms for the numerical variables
plt.figure(figsize=(12,10))

# Iterate over each numerical feature to create a histogram
for index, feature in enumerate(['Age', 'Income']):
    plt.subplot(2, 2, index+1)
    sns.histplot(df_cluster[feature], kde=False, bins=30)
    plt.title(f'Distribution of {feature}', size=14)
    plt.tight_layout()


plt.figure(figsize=(12,10))

for index, feature in enumerate(['Sex', 'Marital status', 'Education', 'Occupation', 'Settlement size']):
    plt.subplot(3, 2, index+1)
    sns.boxplot(y=df_cluster[feature])
    plt.title(f'Boxplot of {feature}', size=14)
    plt.tight_layout()

plt.show()
descriptive_stats, missing_data
```

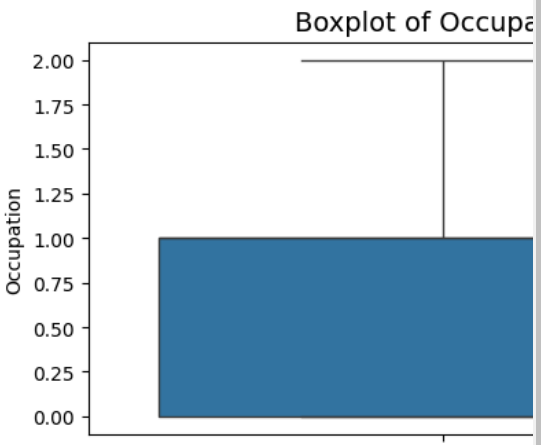


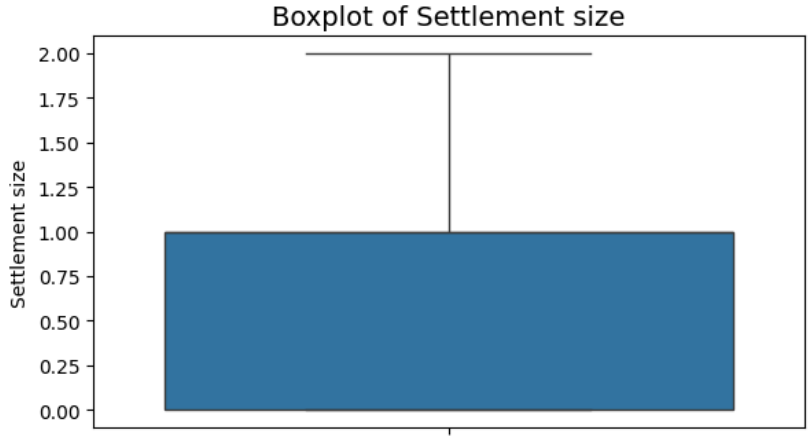










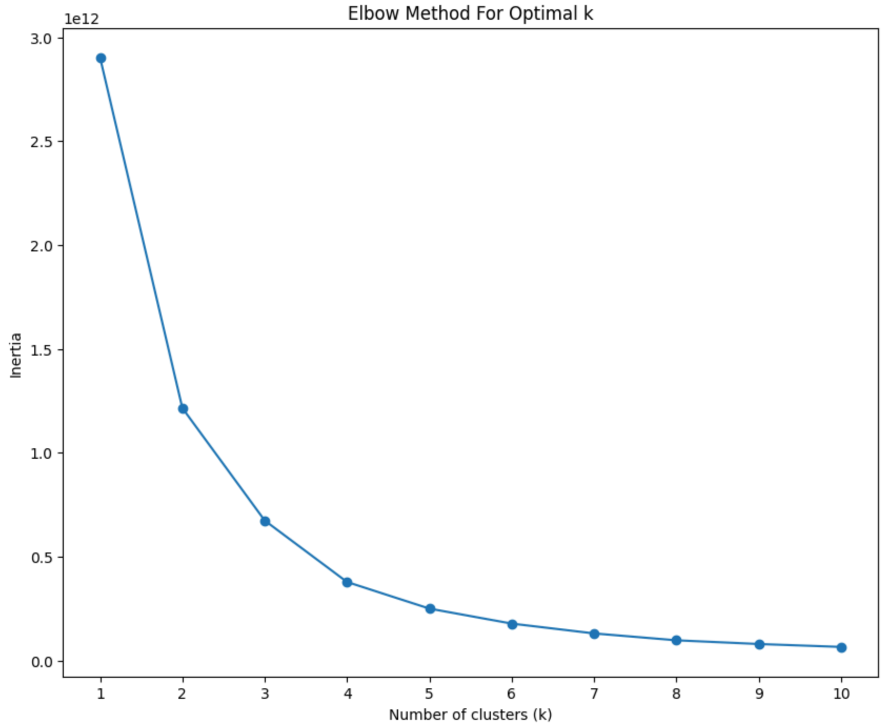



```
(              Sex  Marital status          Age   Education         Income  \
 count  2000.000000     2000.000000  2000.000000  2000.00000    2000.000000
 mean      0.457000        0.496500    35.909000     1.03800  120954.419000
 std       0.498272        0.500113    11.719402     0.59978   38108.824679
 min       0.000000        0.000000    18.000000     0.00000   35832.000000
```

```
25%       0.000000        0.000000    27.000000     1.00000   97663.250000
50%       0.000000        0.000000    33.000000     1.00000  115548.500000
75%       1.000000        1.000000    42.000000     1.00000  138072.250000
max       1.000000        1.000000    76.000000     3.00000  309364.000000

        Occupation  Settlement size
count  2000.000000      2000.000000
mean      0.810500         0.739000
std       0.638587         0.812533
min       0.000000         0.000000
25%       0.000000         0.000000
50%       1.000000         1.000000
75%       1.000000         1.000000
max       2.000000         2.000000  ,
Sex               0
Marital status    0
Age               0
Education         0
Income            0
Occupation        0
Settlement size   0
dtype: int64)
```

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

inertia = []

k_values = range(1, 11)

# Loop over each k value, fit the KMeans model, and add the inertia to the list
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df)
    inertia.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(10, 8))
plt.plot(k_values, inertia, marker='o')
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
```

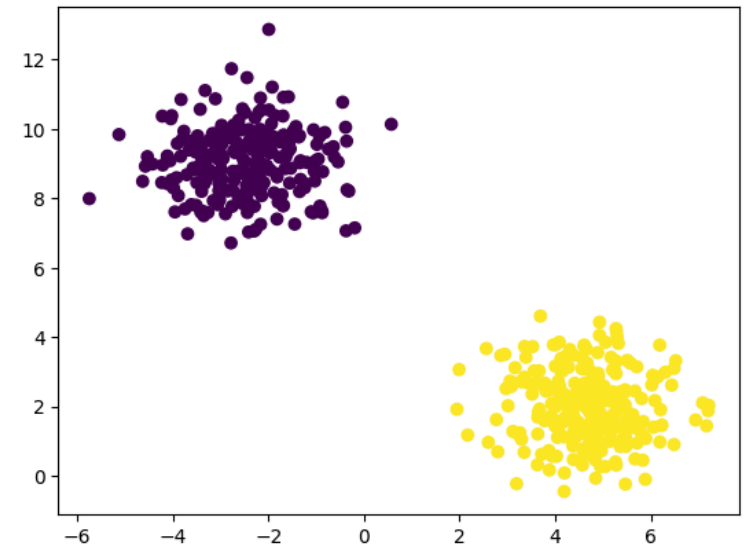



```
from sklearn.metrics import silhouette_score

# Fit K-Means with the chosen number of clusters
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(df.drop('Cluster', axis=1))

# Assign the clusters to the dataframe
df['Cluster'] = kmeans.labels_

# Calculate silhouette score
silhouette_avg = silhouette_score(df.drop('Cluster', axis=1), kmeans.labels_)
print(f'Silhouette Score: {silhouette_avg:.2f}')

cluster_centers = pd.DataFrame(kmeans.cluster_centers_, columns=df.columns[:-1])
print(cluster_centers)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change fr
  warnings.warn(
Silhouette Score: 0.51
             ID       Sex  Marital status        Age  Education  \
0  1.000009e+08  0.383929        0.447545  37.934152   1.079241
1  1.000012e+08  0.569024        0.554433  31.953984   0.920314
2  1.000007e+08  0.295775        0.460094  43.934272   1.356808

          Income  Occupation  Settlement size
0  132208.322545    0.998884         0.954241
1   90883.315376    0.434343         0.370370
2  199404.446009    1.591549         1.375587
```

```
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Generate synthetic data with 2 centers
features, _ = make_blobs(n_samples=500, centers=2, random_state=42)

# Run K-Means clustering
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(features)

# Plot the clustered data
plt.scatter(features[:, 0], features[:, 1], c=kmeans.labels_, cmap='viridis')
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarnin
  warnings.warn(
```




```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt

# Assuming df is your DataFrame after loading 'segmentation data.csv'

# Fit K-Means and calculate silhouette scores for k=2 and k=3
for k in [2, 3]:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df)
    silhouette_avg = silhouette_score(df, kmeans.labels_)
    print(f'Silhouette Score for k={k}: {silhouette_avg:.2f}')

# Calculate inertia for a range of k values to plot the elbow graph
inertia = []
k_values = range(1, 11)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(df)
```

```
    inertia.append(kmeans.inertia_)

# Plot the elbow curve
plt.figure(figsize=(10, 8))
plt.plot(k_values, inertia, marker='o')
plt.title('Elbow Method For Optimal k')
plt.xlabel('Number of clusters (k)')
plt.ylabel('Inertia')
plt.xticks(k_values)
plt.show()
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
Silhouette Score for k=2: 0.58
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
Silhouette Score for k=3: 0.51
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: FutureWarning: The default value of `n_init` will change
  warnings.warn(
```