```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from yellowbrick.cluster import KElbowVisualizer

# Load the dataset
mall_data = pd.read_csv("/home/avcoe/Mall_Customers.csv")

# Data exploration
mall_data.info()

# Correlation heatmap
sns.heatmap(mall_data.corr(), cbar=True, square=True, fmt='.1f', annot=True, cmap='Reds')

# Gender distribution
sns.countplot(x="Gender", data=mall_data)

# Age distribution
sns.countplot(x="Age", data=mall_data)

# Relationship between annual income and spending score
sns.barplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=mall_data)

# Clustering model input
X = mall_data.iloc[:, [2, 3, 4]].values

# Use KElbowVisualizer for finding the optimal number of clusters
```

```
model = KMeans(init='k-means++', random_state=50)
visualizer = KElbowVisualizer(model, k=(1, 11))

visualizer.fit(X)  # Fit the data to the visualizer
visualizer.show()  # Finalize and render the figure

# Build the KMeans model
kmeans = KMeans(n_clusters=5, init='k-means++', random_state=0)

# Cluster assignment
y = kmeans.fit_predict(X)

# Simplified 3D scatter plot
fig = plt.figure(figsize=(10, 10))
ax = fig.add_subplot(111, projection='3d')

for cluster_num in range(5):
    ax.scatter(X[y == cluster_num, 0], X[y == cluster_num, 1], X[y == cluster_num, 2], s=40, label=f"Cluster {cluster_num + 1}")

ax.set_xlabel('Age of a customer -->')
ax.set_ylabel('Annual Income -->')
ax.set_zlabel('Spending Score -->')
ax.legend()

plt.show()
```