

Customer Segmentation Report (Mall-style synthetic dataset)

Generated analysis — 2025-11-13 12:48:00
Best K (by silhouette): 6 (score 0.273)

Executive Summary: - Generated a synthetic dataset with 200 customers resembling a mall-customer profile. - Performed cleaning, exploratory analysis, and K-Means clustering. - Selected 6 clusters based on silhouette score. - Included cluster profiling and actionable recommendations.

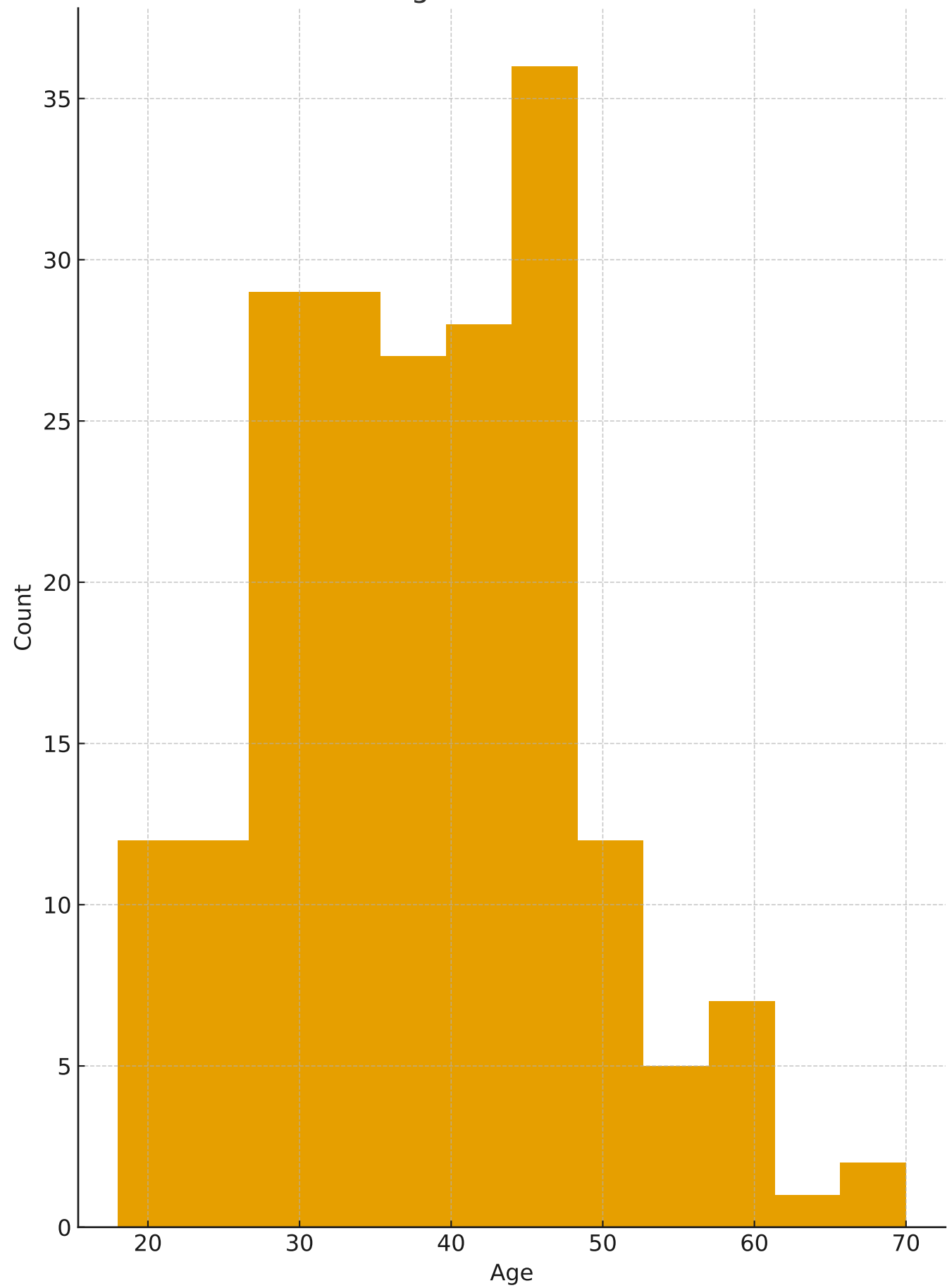
Top Recommendations:

- Target high-spending clusters with premium offers.
- Create retention campaigns for younger, low-spending clusters.
- Use gender-tailored promotions where relevant.

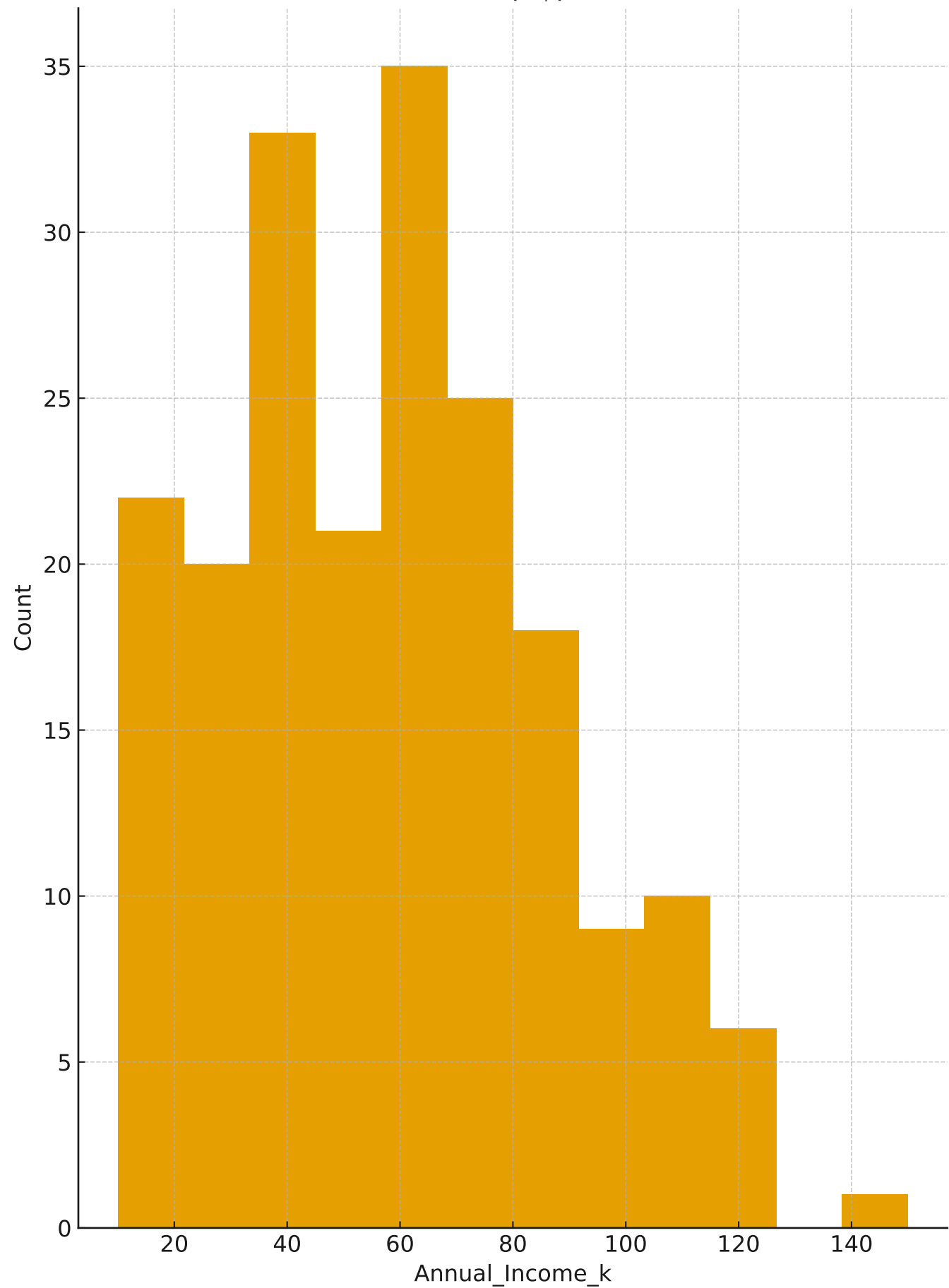
Dataset Snapshot (first 10 rows)

CustomerID	Gender	Age	Annual_Income_k	Spending_Score	Cluster
1	Male	30.0	85.0	79	2
2	Female	40.0	55.0	56	2
3	Female	41.0	59.0	58	2
4	Female	30.0	29.0	39	4
5	Male	58.0	59.0	37	6
6	Male	43.0	51.0	39	3
7	Male	24.0	69.0	59	5
8	Female	45.0	35.0	39	1
9	Female	27.0	75.0	57	5
10	Female	46.0	105.0	100	2

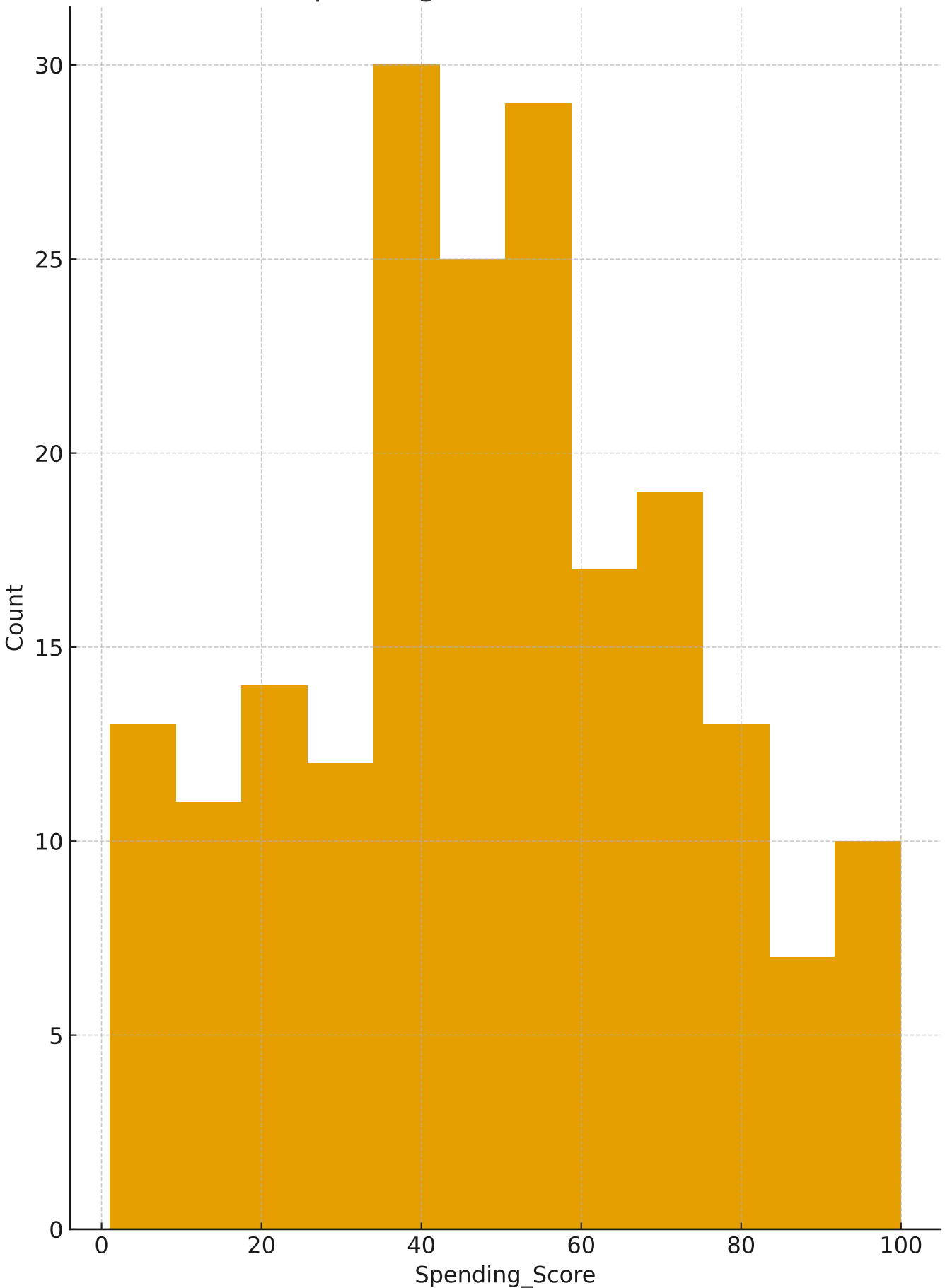
Age Distribution



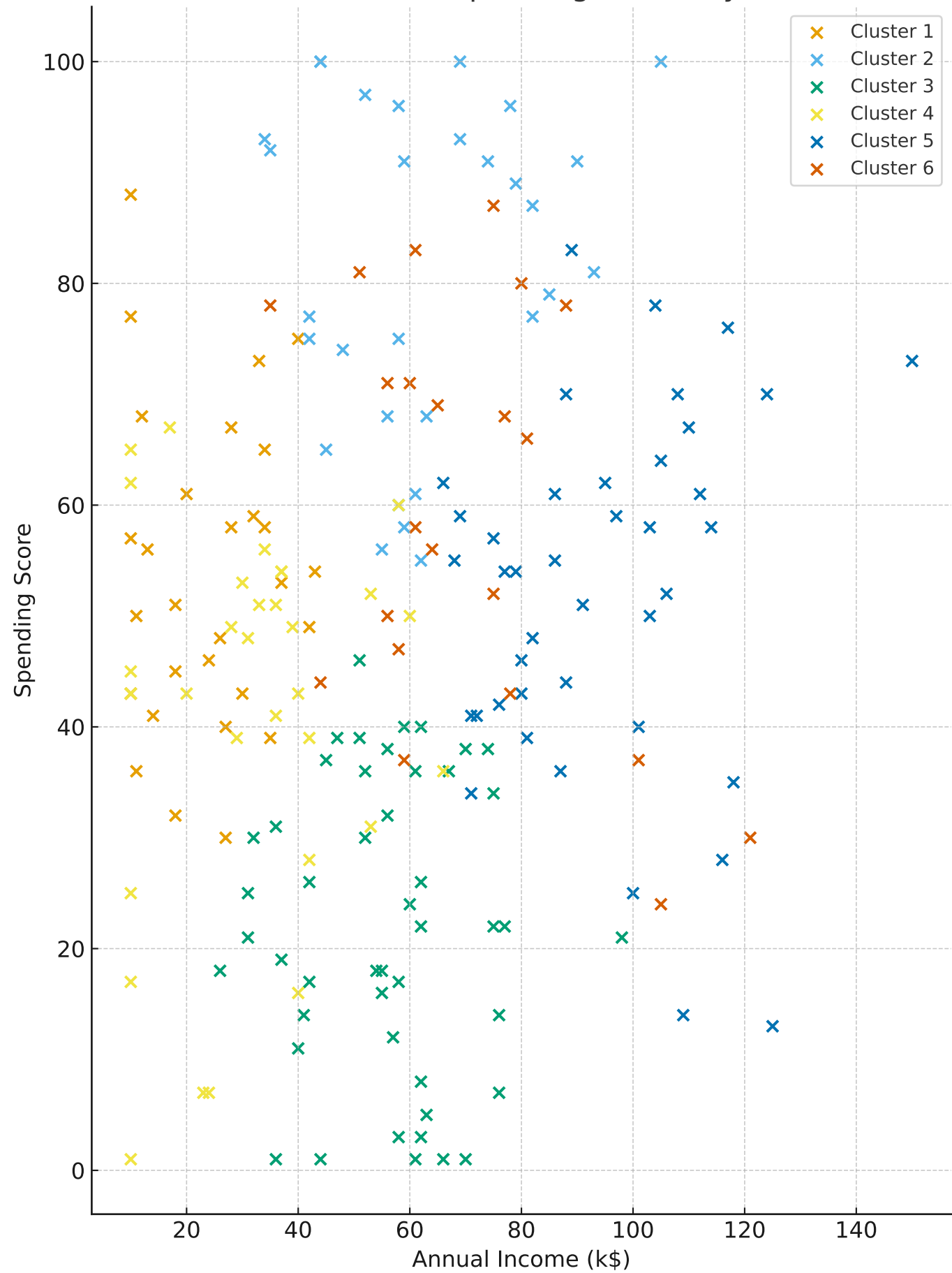
Annual Income (k\$) Distribution



Spending Score Distribution



Annual Income vs Spending Score (by Cluster)



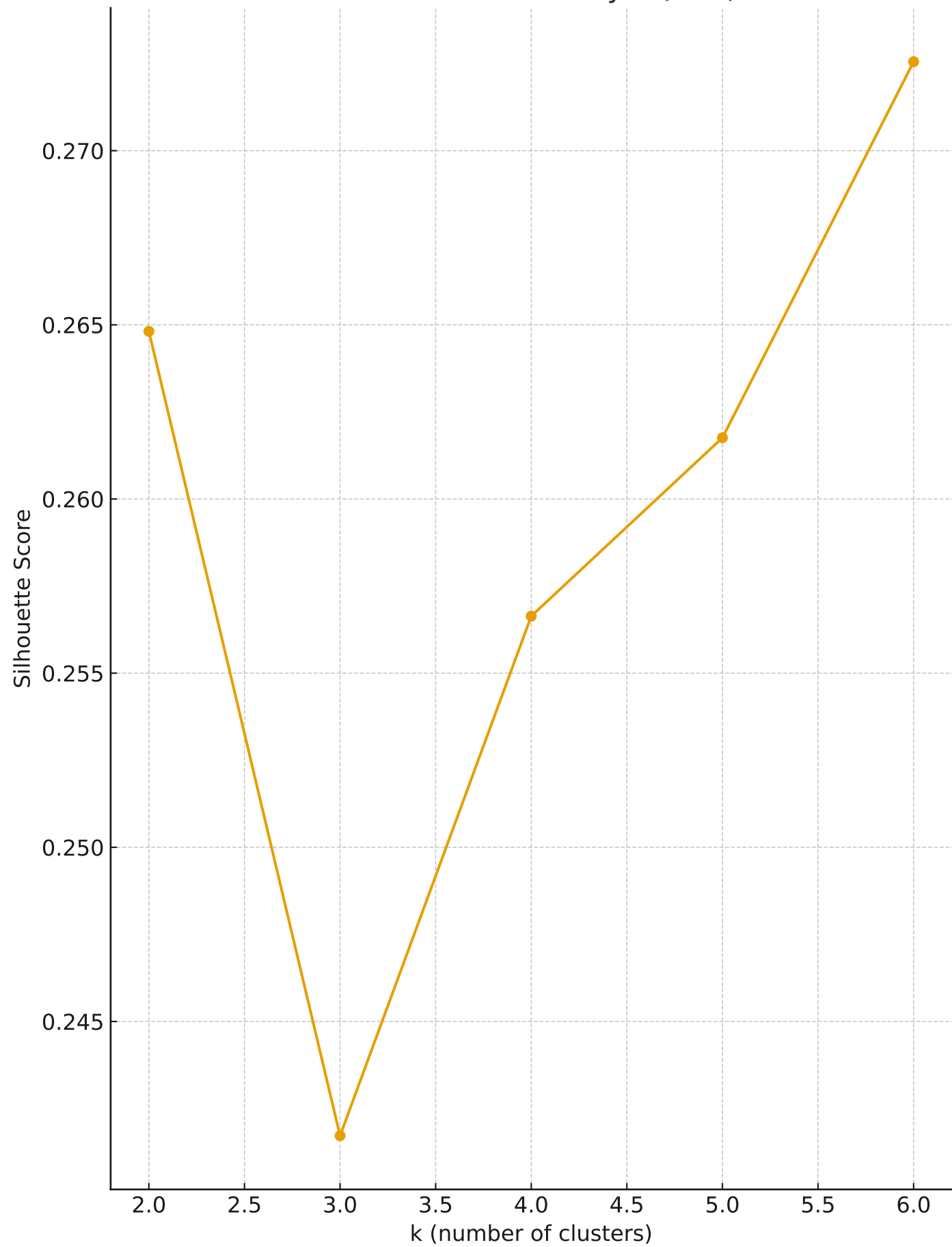
Cluster Centers (profile)

Cluster	Age	Annual_Income_k	Spending_Score
1.0	45.7	24.0	53.9
2.0	34.7	62.7	81.5
3.0	42.3	55.8	22.0
4.0	27.5	31.5	41.4
5.0	30.9	94.6	51.9
6.0	53.3	70.5	59.5

Cluster Profile Summary

Cluster	Count	Avg_Age	Avg_Income_k	Avg_Spending_Score	Male_Pct
1.0	29.0	45.7	24.0	53.9	55.2
2.0	30.0	34.7	62.7	81.5	43.3
3.0	47.0	42.3	55.8	22.0	53.2
4.0	31.0	27.5	31.5	41.4	48.4
5.0	41.0	30.9	94.6	51.9	46.3
6.0	22.0	53.3	70.5	59.5	45.5

Silhouette Scores by k (2..6)



Key Code Snippet (clustering steps)

```
# Features and scaling      features =
df_clean[['Age','Annual_Income_k','Spending_Score']]      scaler =
StandardScaler()      X_scaled = scaler.fit_transform(features)      # Find
best k with silhouette      for k in range(2,7):      labels =
KMeans(n_clusters=k, random_state=42, n_init=10).fit_predict(X_scaled)
sil = silhouette_score(X_scaled, labels)      # Fit final KMeans, assign
clusters      kmeans = KMeans(n_clusters=best_k, random_state=42, n_init=10)
df_clean['Cluster'] = kmeans.fit_predict(X_scaled) + 1
```