```
import pandas as pd

from sklearn.preprocessing import StandardScaler

df = pd.read_csv('Cust_Segmentation.csv')
df.head()
```

	Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Defaulted	Address	DebtIncomeRatio
0	1	41	2	6	19	0.124	1.073	0.0	NBA001	6.3
1	2	47	1	26	100	4.582	8.218	0.0	NBA021	12.8
2	3	33	2	10	57	6.111	5.802	1.0	NBA013	20.9
3	4	29	2	4	19	0.681	0.516	0.0	NBA009	6.3
4	5	47	1	31	253	9.308	8.908	0.0	NBA008	7.2

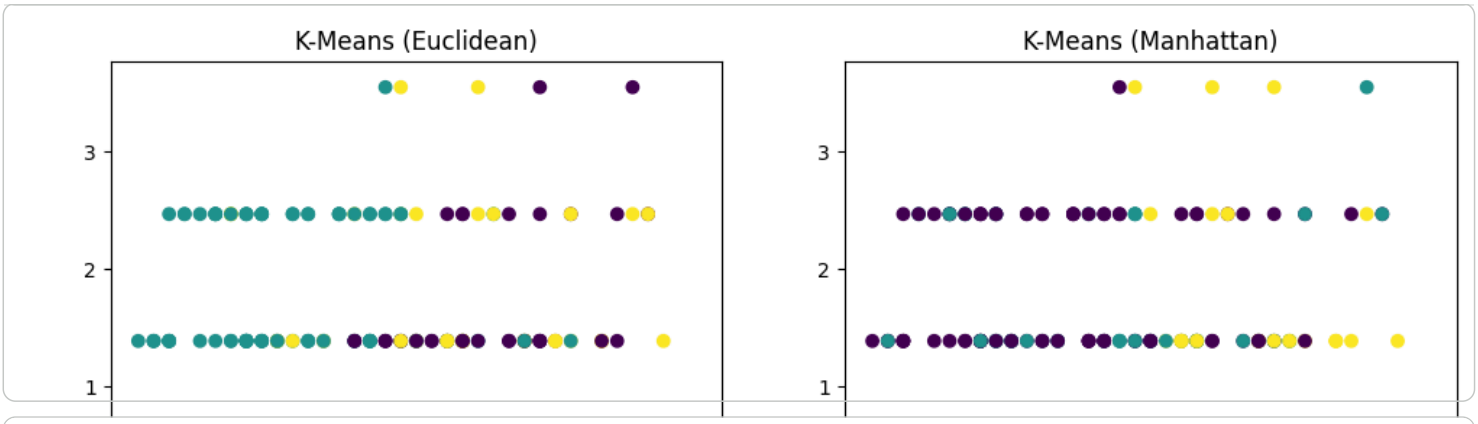
Next steps: Generate code with df | New interactive sheet

```
from sklearn.preprocessing import StandardScaler
X = df[['Age', 'Edu', 'Years Employed', 'Income', 'Card Debt', 'Other Debt', 'DebtIncomeRatio']]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
from sklearn.cluster import KMeans

kmeans_euclidean = KMeans(n_clusters=3, random_state=42)
labels_euclidean = kmeans_euclidean.fit_predict(X_scaled)
df['Cluster_Euclidean'] = labels_euclidean
```

```
import numpy as np
from sklearn.metrics import pairwise_distances

def kmeans_manhattan(X, k, max_iter=100):
    np.random.seed(42)
    centroids = X[np.random.choice(X.shape[0], k, replace=False)]
    for _ in range(max_iter):
        distances = pairwise_distances(X, centroids, metric='manhattan')
        labels = np.argmin(distances, axis=1)
        new_centroids = np.array([X[labels == i].mean(axis=0) for i in range(k)])
        if np.allclose(centroids, new_centroids): break
        centroids = new_centroids
    return labels, centroids

labels_manhattan, centroids_manhattan = kmeans_manhattan(X_scaled, 3)
df['Cluster_Manhattan'] = labels_manhattan
```

```
import matplotlib.pyplot as plt

plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
plt.scatter(X_scaled[:,0], X_scaled[:,1], c=labels_euclidean, cmap='viridis')
plt.title('K-Means (Euclidean)')

plt.subplot(1,2,2)
plt.scatter(X_scaled[:,0], X_scaled[:,1], c=labels_manhattan, cmap='viridis')
plt.title('K-Means (Manhattan)')
plt.show()
```




```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(12,5))
ax = fig.add_subplot(121, projection='3d')
ax.scatter(X_scaled[:,0], X_scaled[:,1], X_scaled[:,2], c=labels_euclidean, cmap='viridis')
ax.set_title('3D Clusters (Euclidean)')

ax2 = fig.add_subplot(122, projection='3d')
ax2.scatter(X_scaled[:,0], X_scaled[:,1], X_scaled[:,2], c=labels_manhattan, cmap='viridis')
ax2.set_title('3D Clusters (Manhattan)')
plt.show()
```