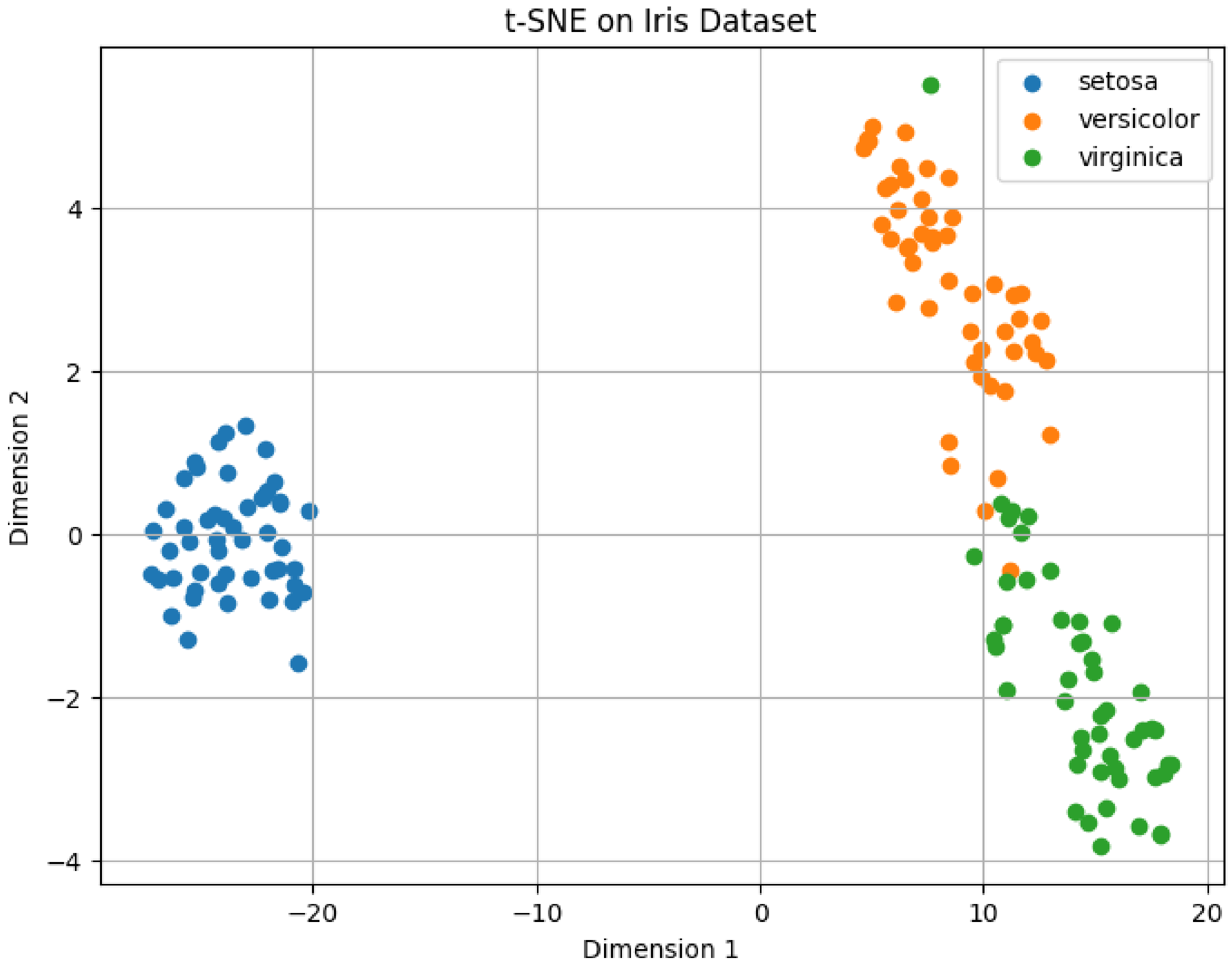
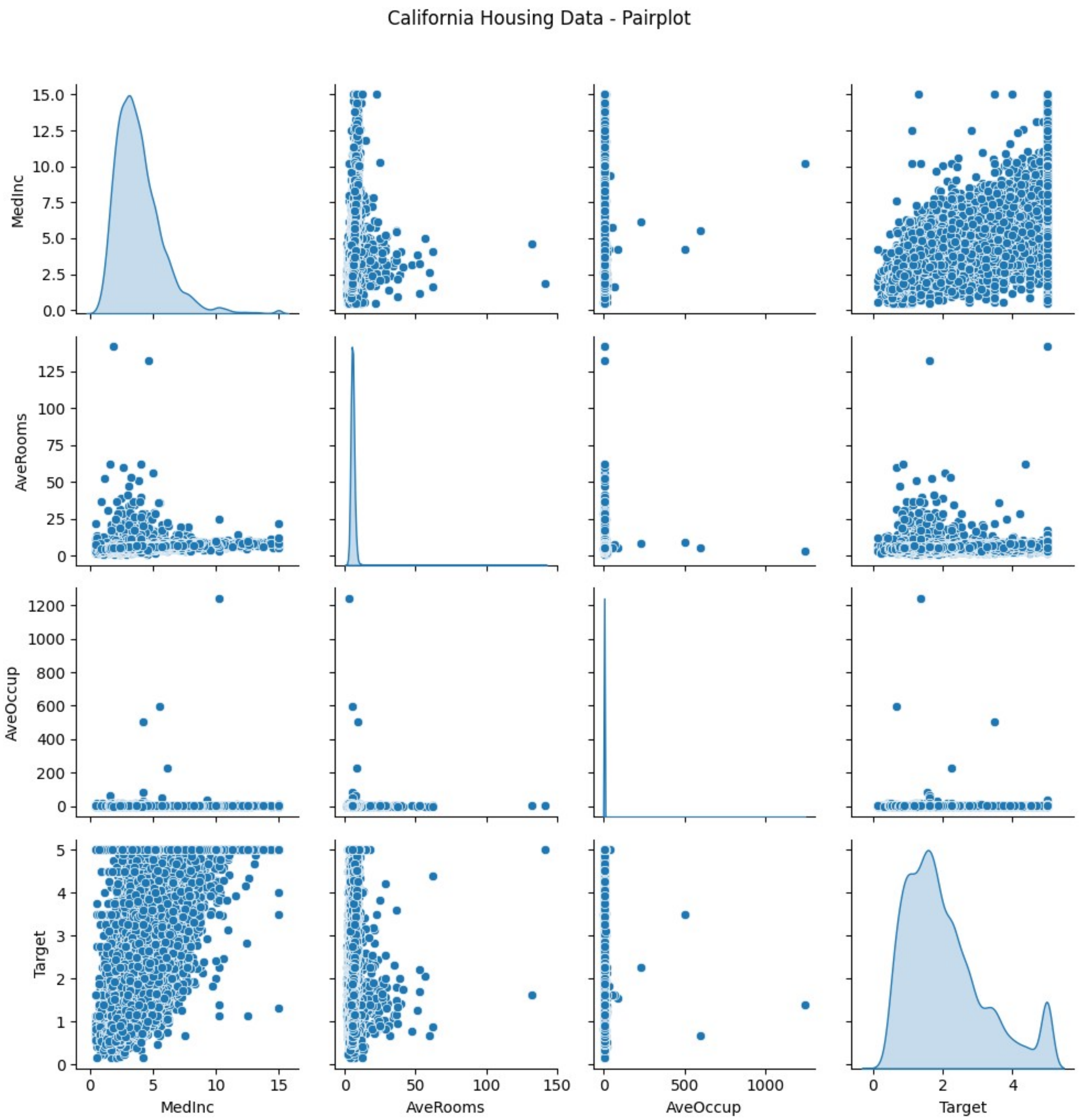
```python
#1
print('Siddharth Raj \n 2241018037')
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.manifold import TSNE
iris = load_iris()
X = iris.data
y = iris.target
labels = iris.target_names
tsne = TSNE(n_components=2, random_state=42)
X_2d = tsne.fit_transform(X)
plt.figure(figsize=(8, 6))
for i in range(len(labels)):
    plt.scatter(X_2d[y == i, 0], X_2d[y == i, 1], label=labels[i])
plt.title("t-SNE on Iris Dataset")
plt.xlabel("Dimension 1")
plt.ylabel("Dimension 2")
plt.legend()
plt.grid(True)
plt.show()
```

```
Siddharth Raj
 2241018037
```



```
#2
print('Siddharth Raj \n 2241018037')
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_california_housing
housing = fetch_california_housing()
df = pd.DataFrame(housing.data, columns=housing.feature_names)
df['Target'] = housing.target
selected_columns = ['MedInc', 'AveRooms', 'AveOccup', 'Target']
sns.pairplot(df[selected_columns], diag_kind='kde')
plt.suptitle("California Housing Data - Pairplot", y=1.02)
plt.tight_layout()
plt.show()
```

```
Siddharth Raj
 2241018037
```



```
#3
print('Siddharth Raj \n 2241018037')
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
import numpy as np
annual_df = pd.read_csv('data.csv', skiprows=4)
annual_df.columns = ['Year', 'AnnualAvgTemp']
jan_df = pd.read_csv('data.csv', skiprows=4)
jan_df.columns = ['Year', 'JanHighTemp']
df = pd.merge(annual_df, jan_df, on='Year')
```

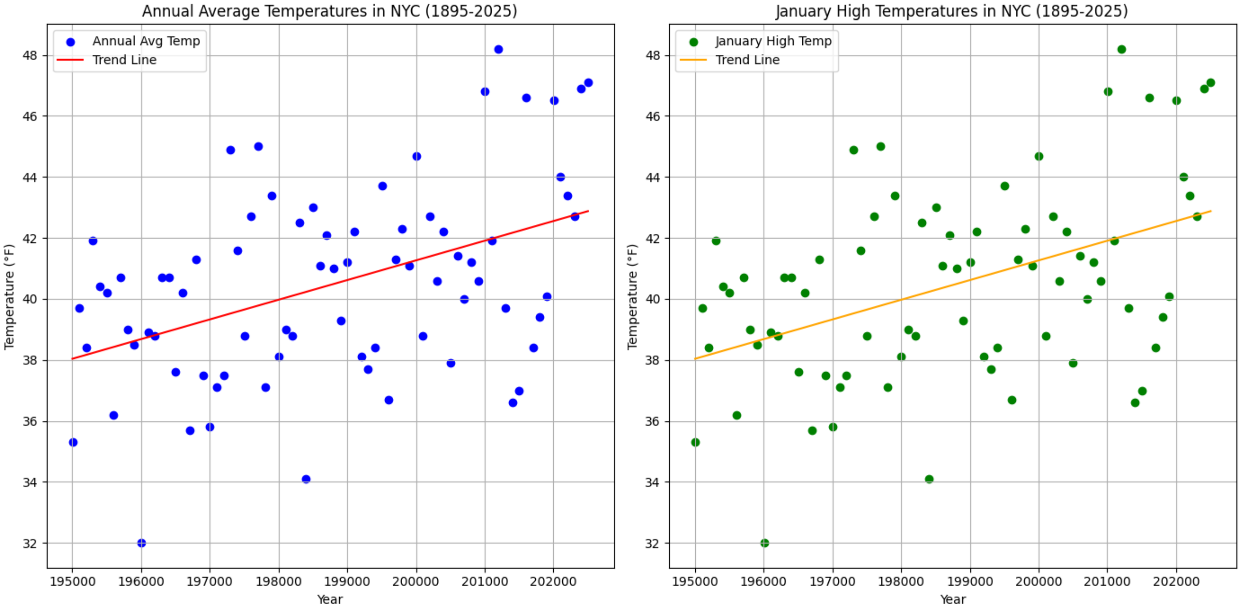
```
X = df['Year'].values.reshape(-1, 1)
y_annual = df['AnnualAvgTemp'].values
y_jan = df['JanHighTemp'].values
model_annual = LinearRegression()
model_annual.fit(X, y_annual)
trend_annual = model_annual.predict(X)
model_jan = LinearRegression()
model_jan.fit(X, y_jan)
trend_jan = model_jan.predict(X)
plt.figure(figsize=(14, 7))

plt.subplot(1, 2, 1)
plt.scatter(df['Year'], df['AnnualAvgTemp'], color='blue',
label='Annual Avg Temp')
plt.plot(df['Year'], trend_annual, color='red', label='Trend Line')
plt.title('Annual Average Temperatures in NYC (1895-2025)')
plt.xlabel('Year')
plt.ylabel('Temperature (°F)')
plt.legend()
plt.grid(True)

plt.subplot(1, 2, 2)
plt.scatter(df['Year'], df['JanHighTemp'], color='green',
label='January High Temp')
plt.plot(df['Year'], trend_jan, color='orange', label='Trend Line')
plt.title('January High Temperatures in NYC (1895-2025)')
plt.xlabel('Year')
plt.ylabel('Temperature (°F)')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()

Siddharth Raj
 2241018037
```



```
#4
print('Siddharth Raj \n 2241018037')
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Prediction Accuracy: {accuracy * 100:.2f}%")
```

```
Siddharth Raj
 2241018037
Prediction Accuracy: 100.00%
```

```
#5
'''The new point P = (3.0, 3.0) is classified as Class 1'''
```

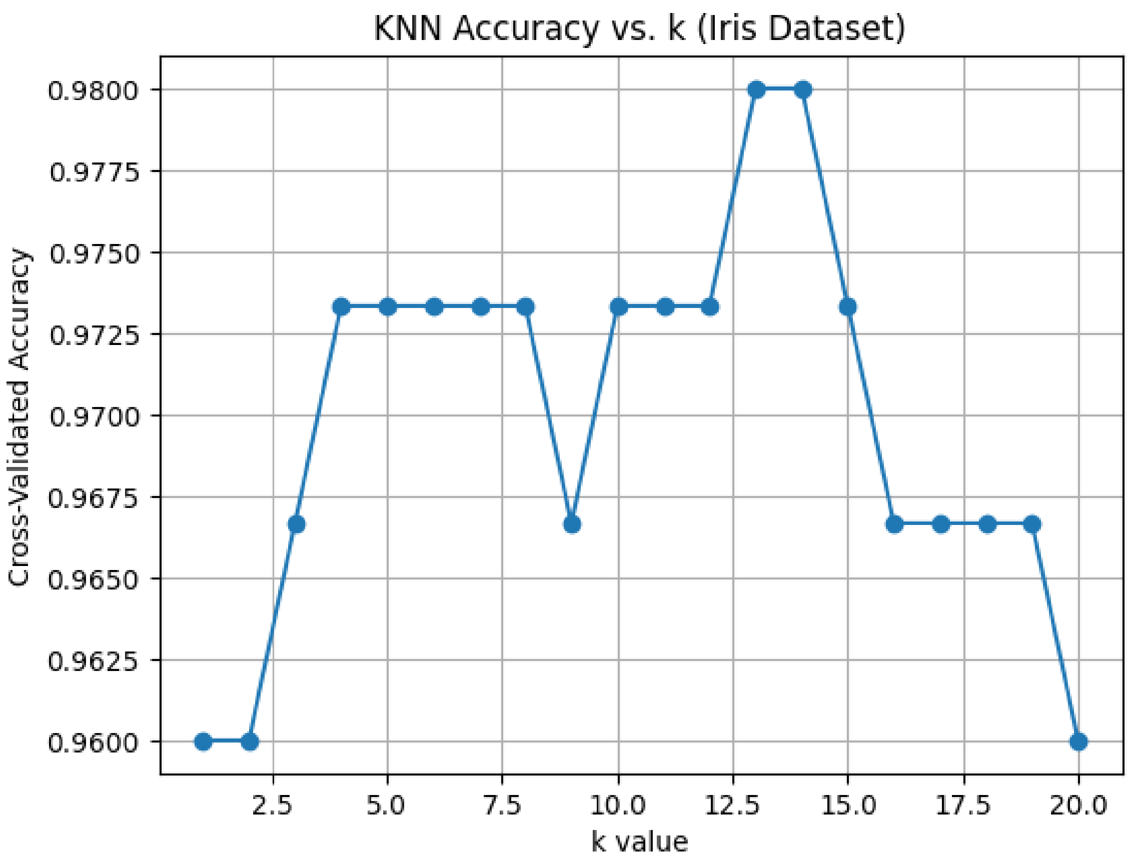
```
'The new point P = (3.0, 3.0) is classified as Class 1'
```

```
#6
'''The new student is classified as: Pass'''
```

```
'The new student is classified as: Pass'
```

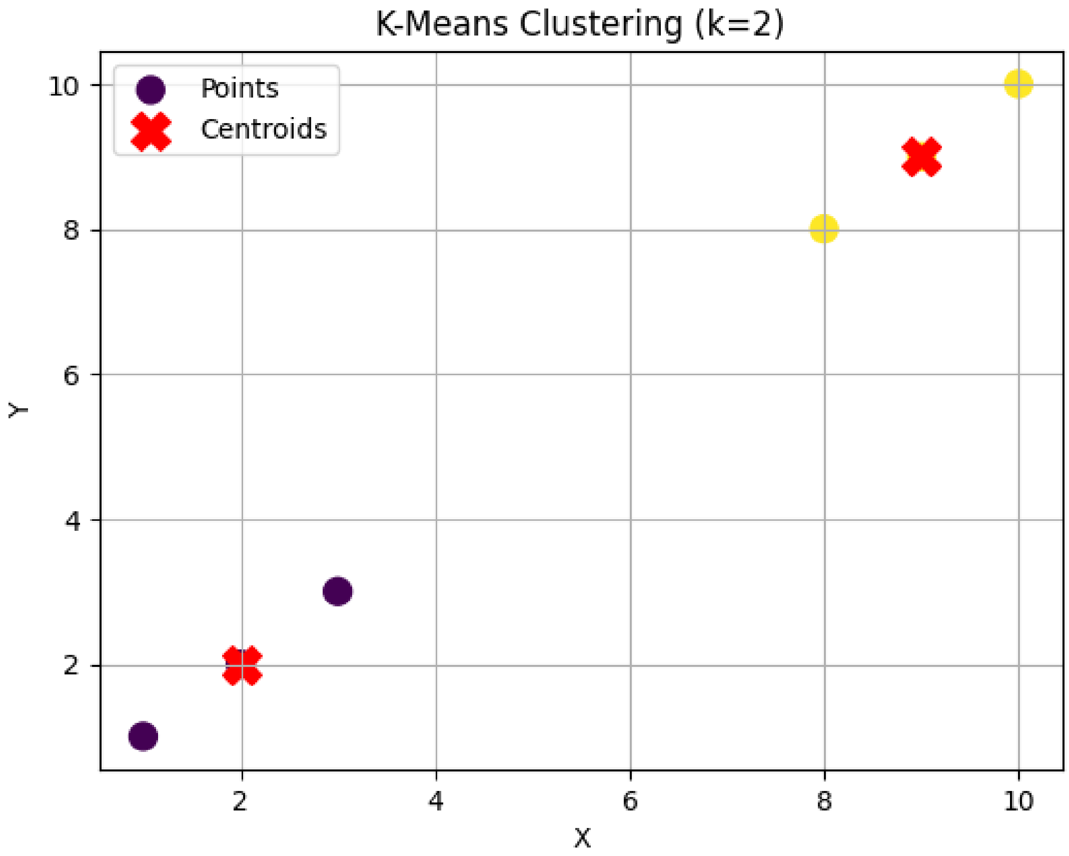
```python
#7
print('Siddharth Raj \n 2241018037')
from sklearn.datasets import load_iris
from sklearn.model_selection import KFold, cross_val_score
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
iris = load_iris()
X = iris.data
y = iris.target
kf = KFold(n_splits=5, shuffle=True, random_state=42)
k_values = list(range(1, 21))
average_scores = []
for k in k_values:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, X, y, cv=kf)
    average_scores.append(scores.mean())
best_k = k_values[average_scores.index(max(average_scores))]
print(f"Best k value: {best_k} with accuracy:
{max(average_scores)*100:.2f}%")
plt.plot(k_values, average_scores, marker='o')
plt.xlabel('k value')
plt.ylabel('Cross-Validated Accuracy')
plt.title('KNN Accuracy vs. k (Iris Dataset)')
plt.grid(True)
plt.show()
```

```
Siddharth Raj
 2241018037
Best k value: 13 with accuracy: 98.00%
```



```
#8
print('Siddharth Raj \n 2241018037')
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
data = np.array([
    [1, 1],
    [2, 2],
    [3, 3],
    [8, 8],
    [9, 9],
    [10, 10]
])
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(data)
labels = kmeans.labels_
centroids = kmeans.cluster_centers_
plt.scatter(data[:, 0], data[:, 1], c=labels, cmap='viridis', s=100,
label='Points')
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200,
marker='X', label='Centroids')
plt.title('K-Means Clustering (k=2)')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.grid(True)
plt.show()
```

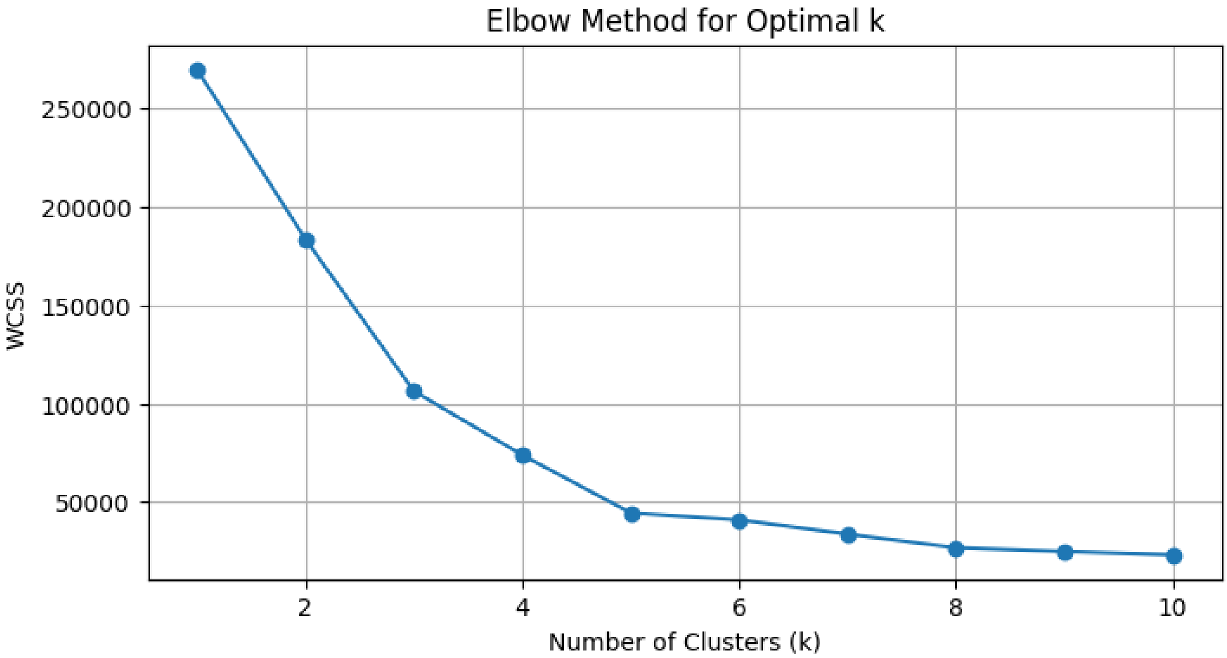
```
Siddharth Raj
 2241018037
```



```
#9
print('Siddharth Raj \n 2241018037')
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
data = pd.read_csv('Mall_Customers.csv')
X = data[['Annual Income (k$)', 'Spending Score (1-100)']]
wcss = []

for k in range(1, 11):
    kmeans = KMeans(n_clusters=k, random_state=42)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.figure(figsize=(8, 4))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal k')
plt.xlabel('Number of Clusters (k)')
plt.ylabel('WCSS')
plt.grid(True)
plt.show()
```

```
Siddharth Raj
 2241018037
```



```
#10
print('Siddharth Raj \n 2241018037')

import pandas as pd

# Part (a)
data_a = [7, 11, 13, 17]
series_a = pd.Series(data_a)
print("Series A:\n", series_a)

# Part (b)
series_b = pd.Series([100.0] * 5)
print("\nSeries B:\n", series_b)

import numpy as np

# Part (c)
random_numbers = np.random.randint(0, 101, size=20)
series_c = pd.Series(random_numbers)
```

```
print("\nSeries C (Random Numbers):\n", series_c)
print("\nDescriptive Statistics of Series C:\n", series_c.describe())

# Part (d)
temperatures = pd.Series(
    [98.6, 98.9, 100.2, 97.9],
    index=['Julie', 'Charlie', 'Sam', 'Andrea']
)

print("\nTemperatures Series:\n", temperatures)

# Part (e)
temp_dict = {
    'Julie': 98.6,
    'Charlie': 98.9,
    'Sam': 100.2,
    'Andrea': 97.9
}

series_e = pd.Series(temp_dict)
print("\nSeries from Dictionary:\n", series_e)
```