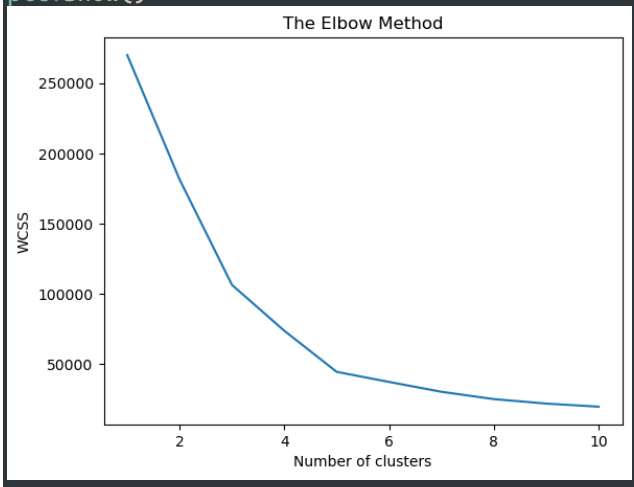
Viikko 44 -tehtävät

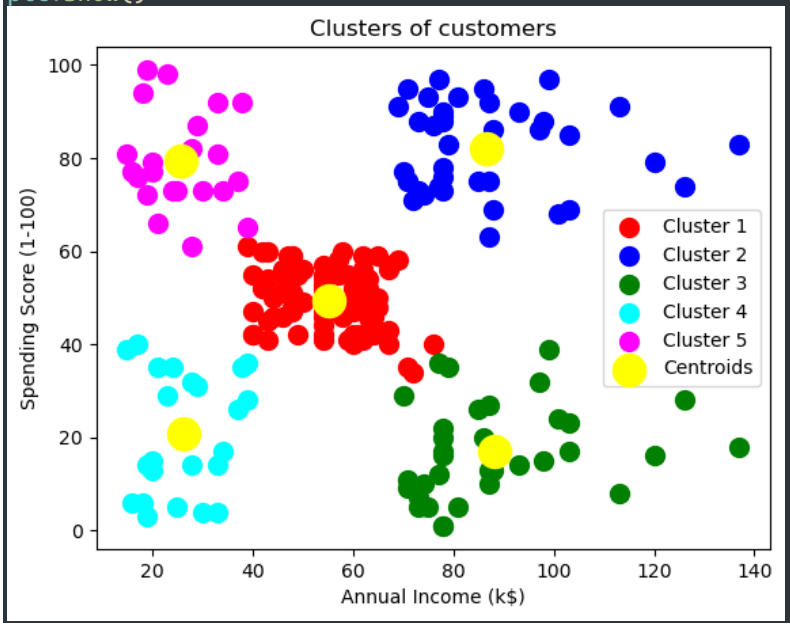
Tehtävä 1

```
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
df = pd.read_csv('./work/viikko9/datasets/customers.csv')
X = df.iloc[:, [3, 4]].values

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    model = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    model.fit(X)
    wcss.append(model.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



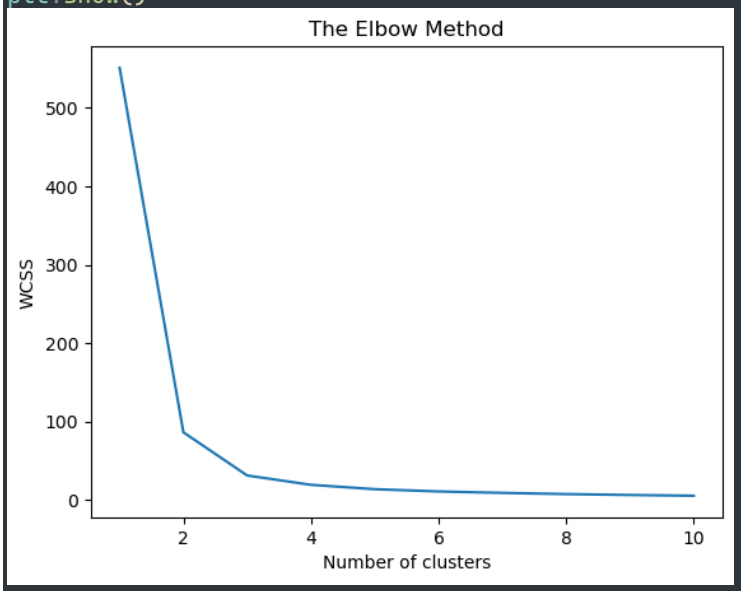

```
# Training the K-Means model on the dataset
model = KMeans(n_clusters = 5, init = 'k-means++', random_state = 0)
y_kmeans = model.fit_predict(X)

# Visualising the clusters
```

```
plt.scatter(X[y_kmeans == 0, 0], X[y_kmeans == 0, 1], s = 100, c = 'red', label =
'Cluster 1')
plt.scatter(X[y_kmeans == 1, 0], X[y_kmeans == 1, 1], s = 100, c = 'blue', label =
'Cluster 2')
plt.scatter(X[y_kmeans == 2, 0], X[y_kmeans == 2, 1], s = 100, c = 'green', label =
'Cluster 3')
plt.scatter(X[y_kmeans == 3, 0], X[y_kmeans == 3, 1], s = 100, c = 'cyan', label =
'Cluster 4')
plt.scatter(X[y_kmeans == 4, 0], X[y_kmeans == 4, 1], s = 100, c = 'magenta', label =
'Cluster 5')
plt.scatter(model.cluster_centers_[:, 0], model.cluster_centers_[:, 1], s = 300, c =
'yellow', label = 'Centroids')
plt.title('Clusters of customers')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.legend()
plt.show()
```



Tehtävä 2

```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

# Importing the dataset
df = pd.read_csv('./work/viikko9/datasets/iris.csv')
X = df[['petal_length', 'petal_width']]

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    model = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    model.fit(X)
    wcss.append(model.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```




```
iris_map = {0: 'versicolor', 1: 'setosa', 2: 'virginica'}
df_pred = pd.DataFrame(y_kmeans)
df_pred.columns = ['predict']
df_pred['predict'] = df_pred['predict'].map(iris_map)
```

```
df_pred['real'] = df['species']
pd.crosstab(df_pred['predict'], df_pred['real'])
```

predict \ real	setosa	versicolor	virginica
setosa	50	0	0
versicolor	0	48	4
virginica	0	2	46

```
# Visualising the clusters
plt.figure(figsize=(10, 7))
sns.scatterplot(x = 'petal_length', y = 'petal_width', hue=df_pred['predict'],
data=df, palette=['red', 'green', 'blue'])
sns.scatterplot(x = model.cluster_centers_[:, 0], y = model.cluster_centers_[:, 1],
color = 'black', marker='x', s = 150, label = 'Centroids')
plt.title('Clusters of Iris Flowers')
plt.xlabel('Petal length')
plt.ylabel('Petal width')
plt.legend()
plt.show()
```