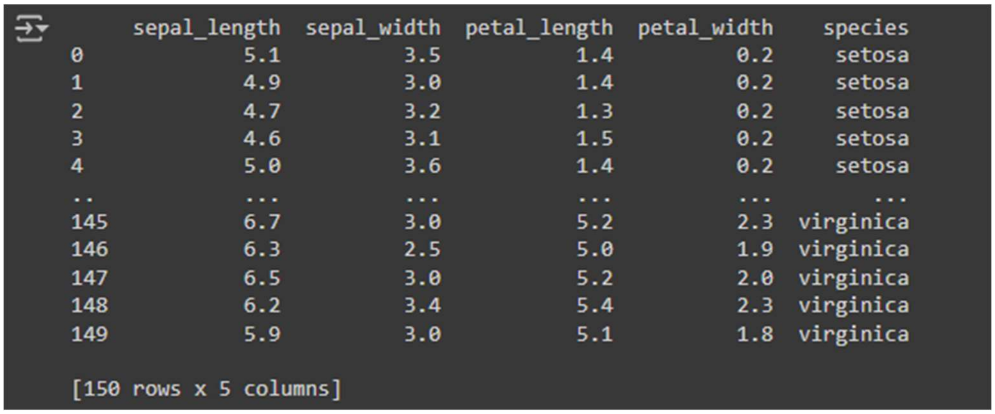
```
from sklearn.cluster import KMeans
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
#importing the required libraries for K-means and EDA
```

```
data = pd.read_csv('iris.csv')
#reading the dataset using pandas
```

```
print(data)
```

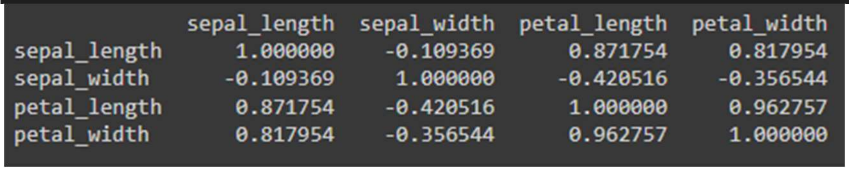


```
     sepal_length  sepal_width  petal_length  petal_width    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]
```

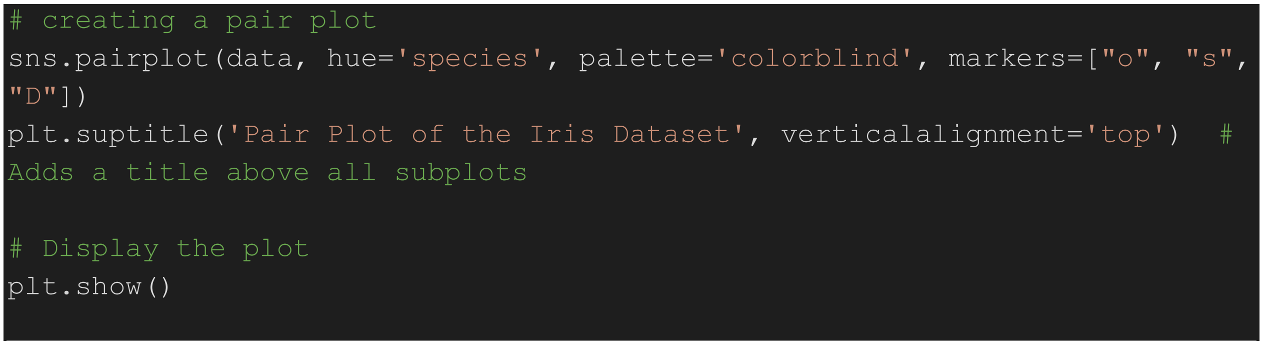
```
data.drop('species', inplace=True, axis=1)
#removing the species column to have only numerical data
```

```
corr_matrix = data.corr()
print(corr_matrix)
```



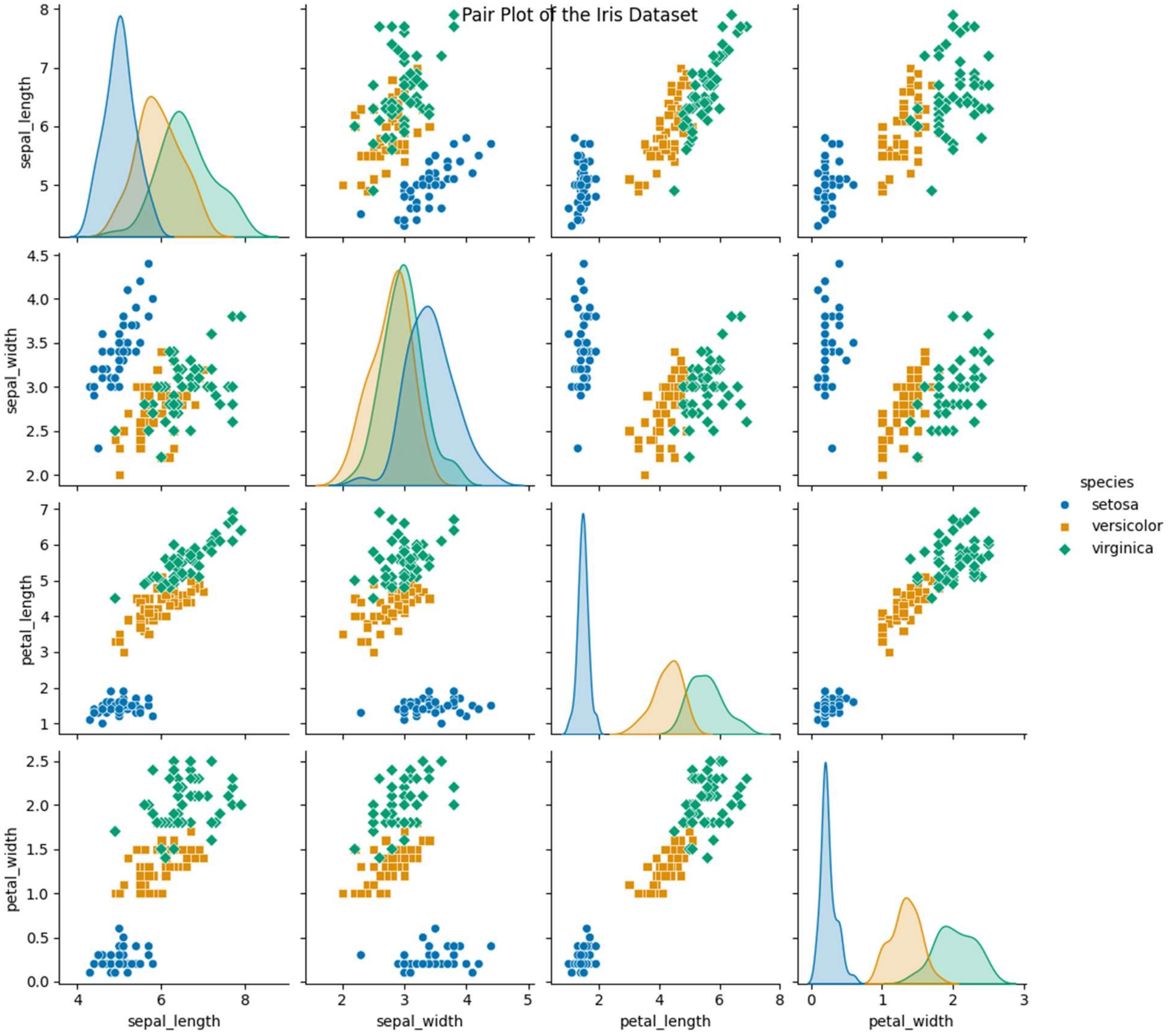
```
              sepal_length  sepal_width  petal_length  petal_width
sepal_length      1.000000    -0.109369      0.871754     0.817954
sepal_width      -0.109369     1.000000     -0.420516    -0.356544
petal_length      0.871754    -0.420516      1.000000     0.962757
petal_width       0.817954    -0.356544      0.962757     1.000000
```

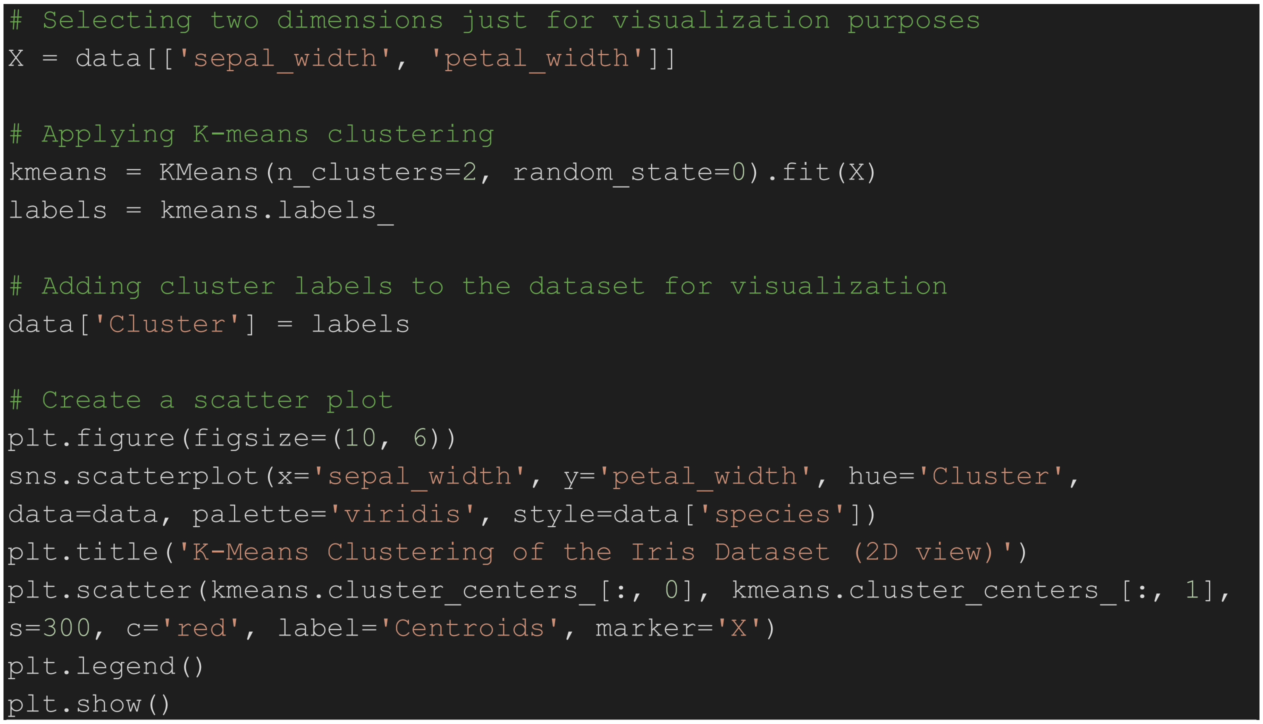
```
data = pd.read_csv('iris.csv')
#reading the dataset using pandas
```



```
# creating a pair plot
sns.pairplot(data, hue='species', palette='colorblind', markers=["o", "s",
"D"])
plt.suptitle('Pair Plot of the Iris Dataset', verticalalignment='top')  #
Adds a title above all subplots

# Display the plot
plt.show()
```



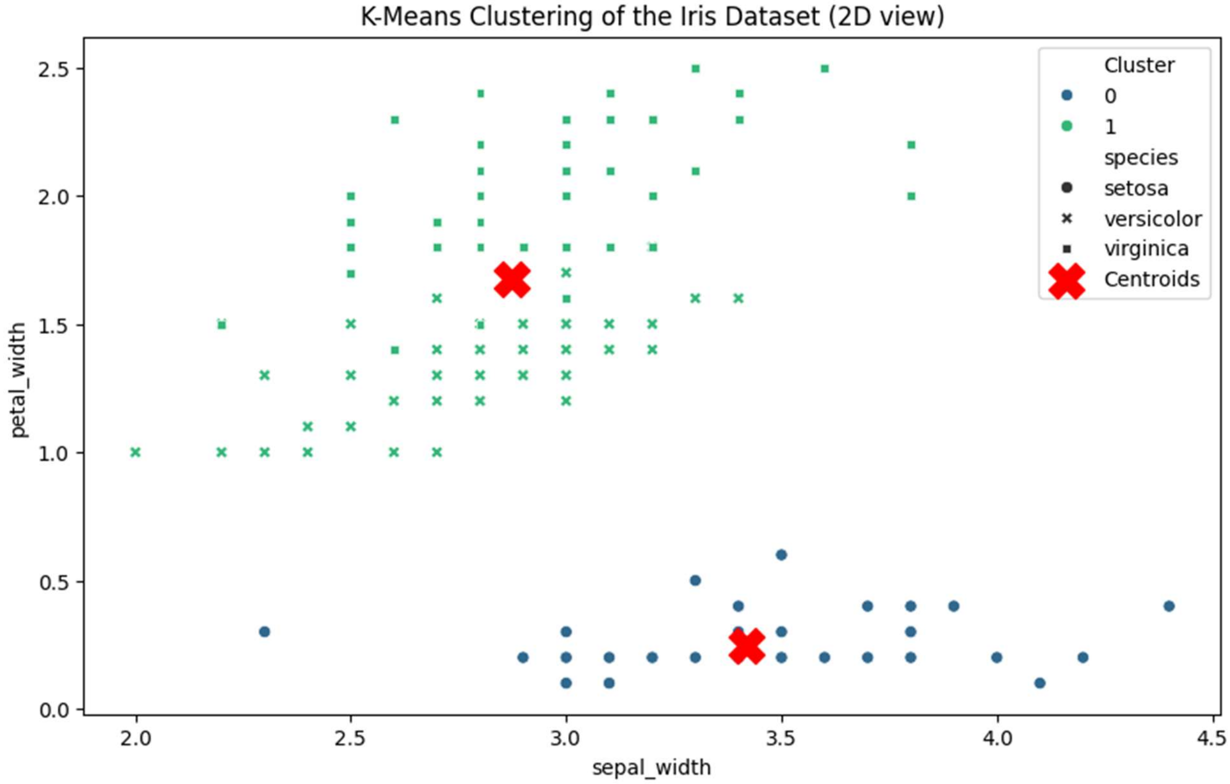


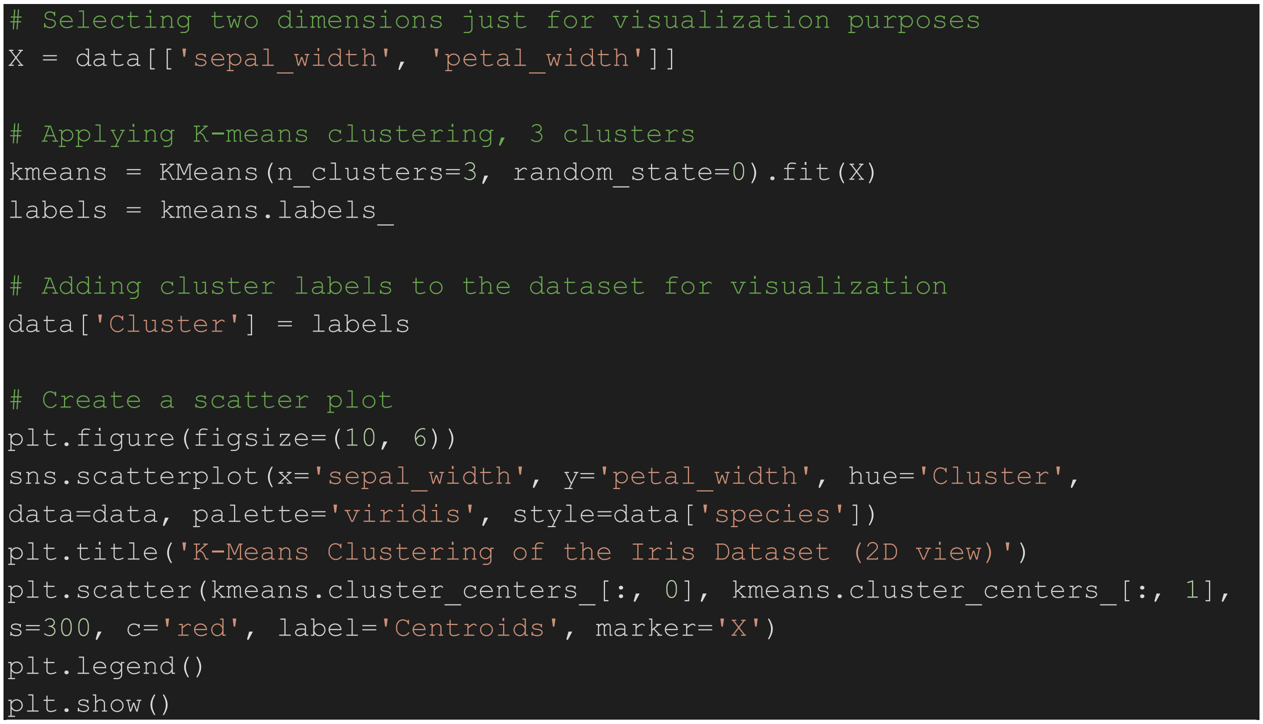
```
# Selecting two dimensions just for visualization purposes
X = data[['sepal_width', 'petal_width']]

# Applying K-means clustering
kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
labels = kmeans.labels_

# Adding cluster labels to the dataset for visualization
data['Cluster'] = labels

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_width', y='petal_width', hue='Cluster',
data=data, palette='viridis', style=data['species'])
plt.title('K-Means Clustering of the Iris Dataset (2D view)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red', label='Centroids', marker='X')
plt.legend()
plt.show()
```



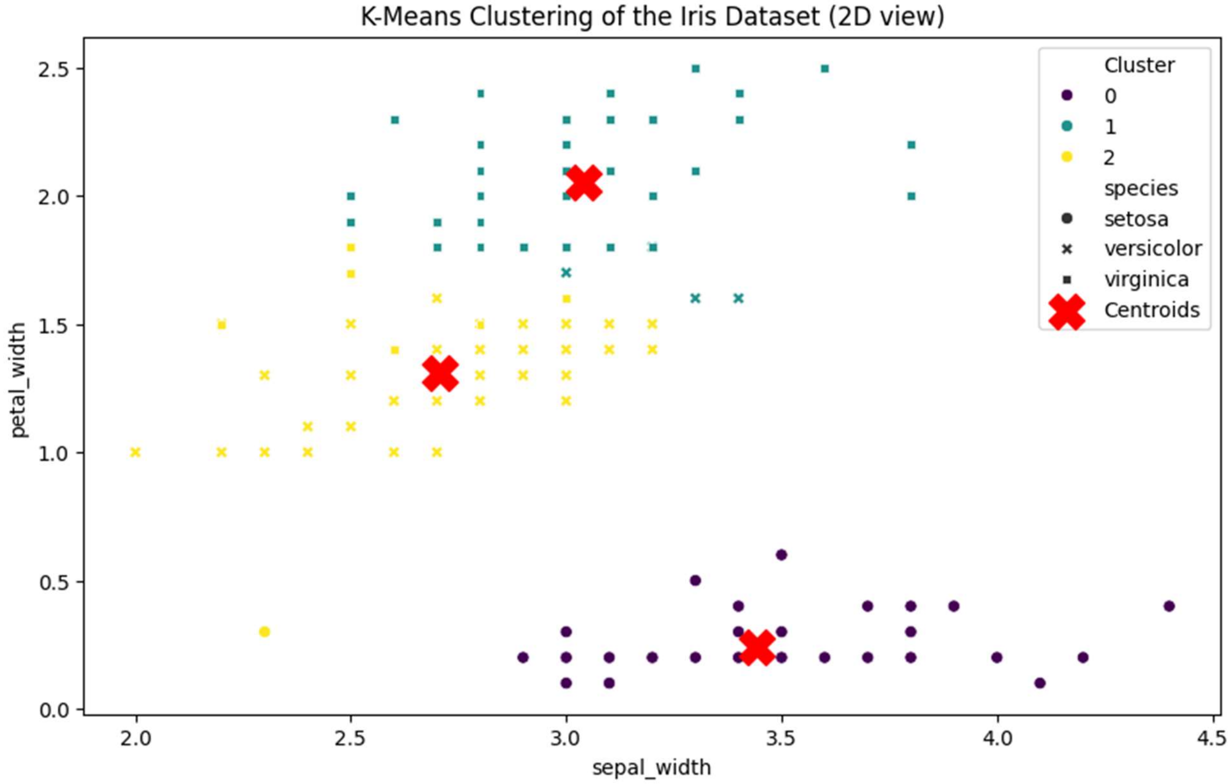


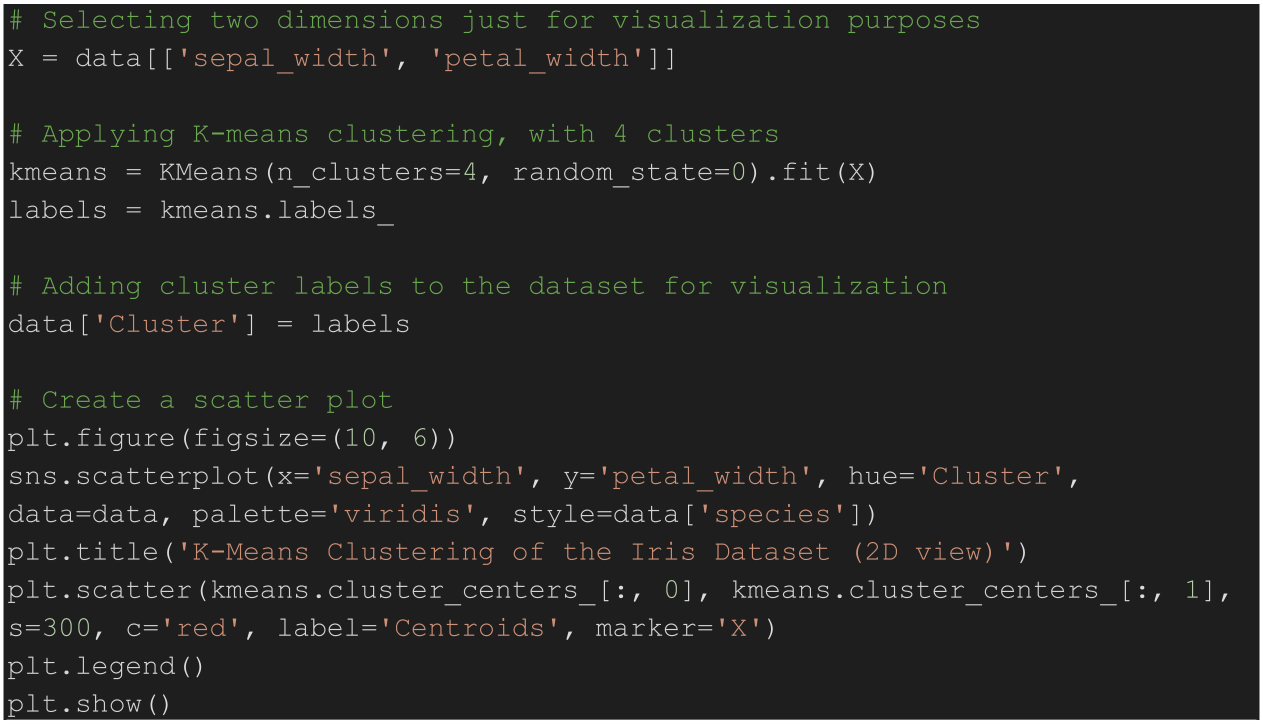
```
# Selecting two dimensions just for visualization purposes
X = data[['sepal_width', 'petal_width']]

# Applying K-means clustering, 3 clusters
kmeans = KMeans(n_clusters=3, random_state=0).fit(X)
labels = kmeans.labels_

# Adding cluster labels to the dataset for visualization
data['Cluster'] = labels

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_width', y='petal_width', hue='Cluster',
data=data, palette='viridis', style=data['species'])
plt.title('K-Means Clustering of the Iris Dataset (2D view)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red', label='Centroids', marker='X')
plt.legend()
plt.show()
```



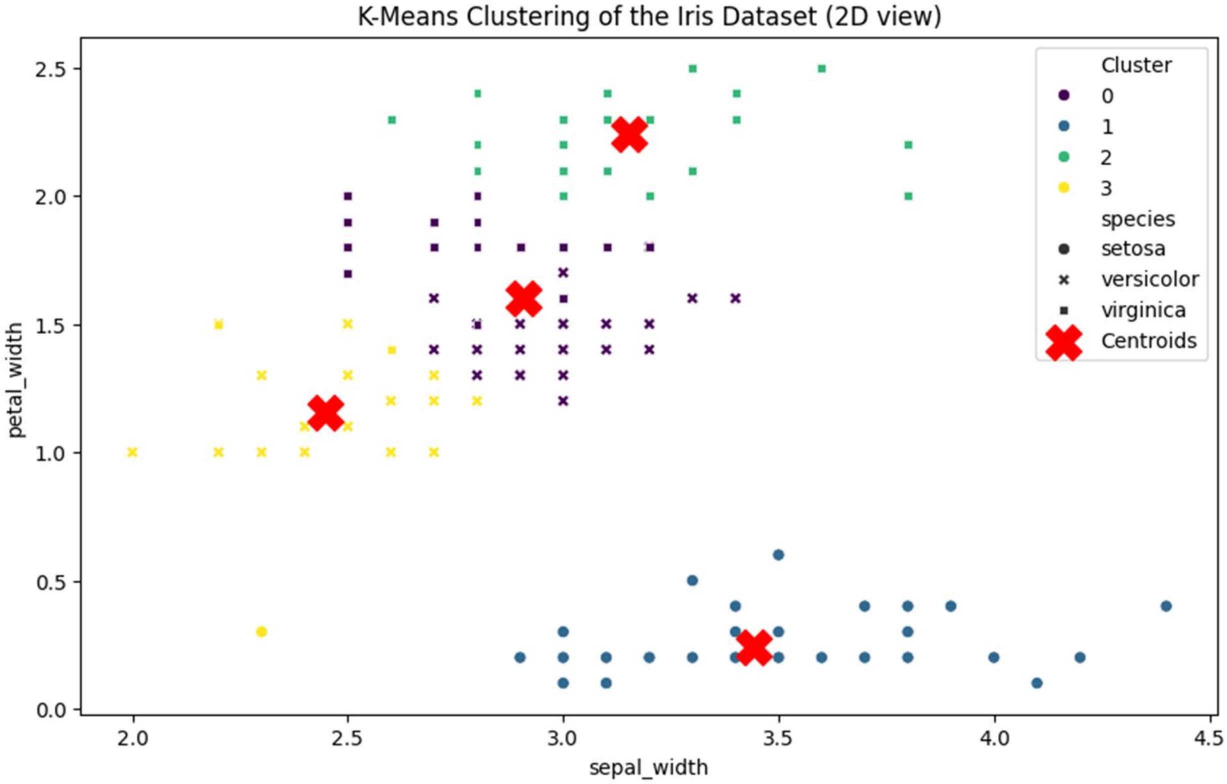


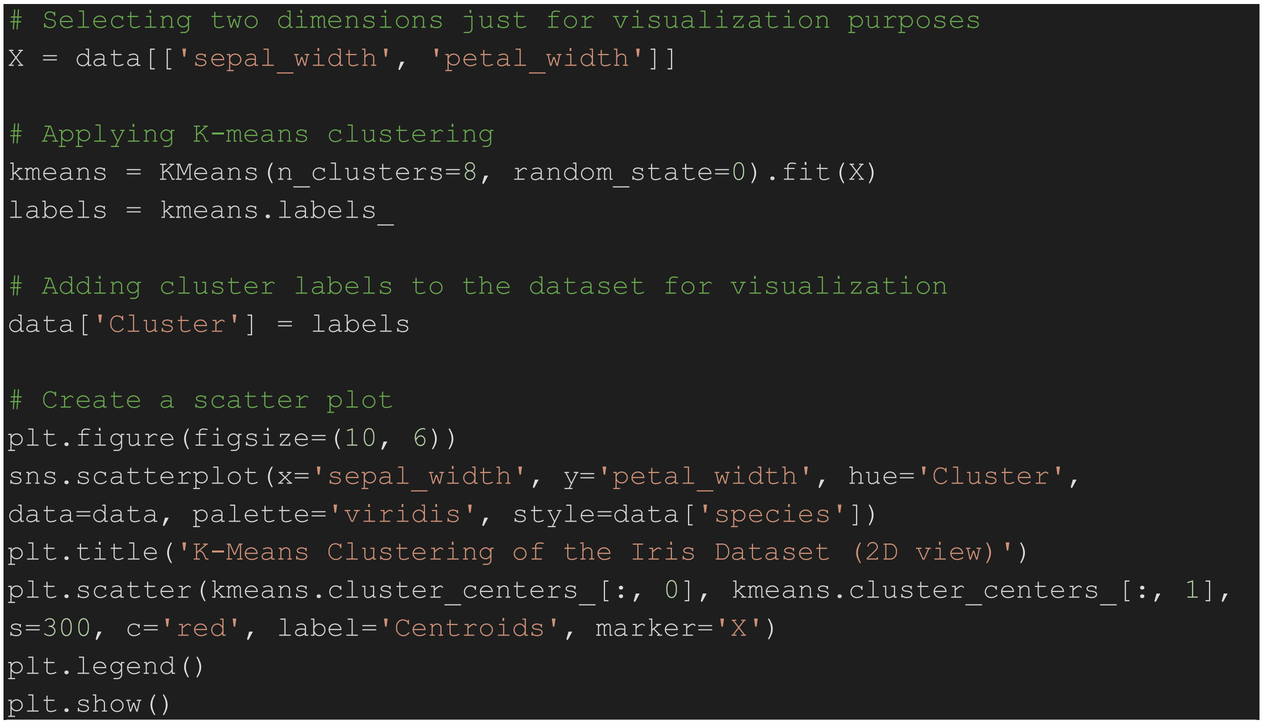
```
# Selecting two dimensions just for visualization purposes
X = data[['sepal_width', 'petal_width']]

# Applying K-means clustering, with 4 clusters
kmeans = KMeans(n_clusters=4, random_state=0).fit(X)
labels = kmeans.labels_

# Adding cluster labels to the dataset for visualization
data['Cluster'] = labels

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_width', y='petal_width', hue='Cluster',
data=data, palette='viridis', style=data['species'])
plt.title('K-Means Clustering of the Iris Dataset (2D view)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red', label='Centroids', marker='X')
plt.legend()
plt.show()
```



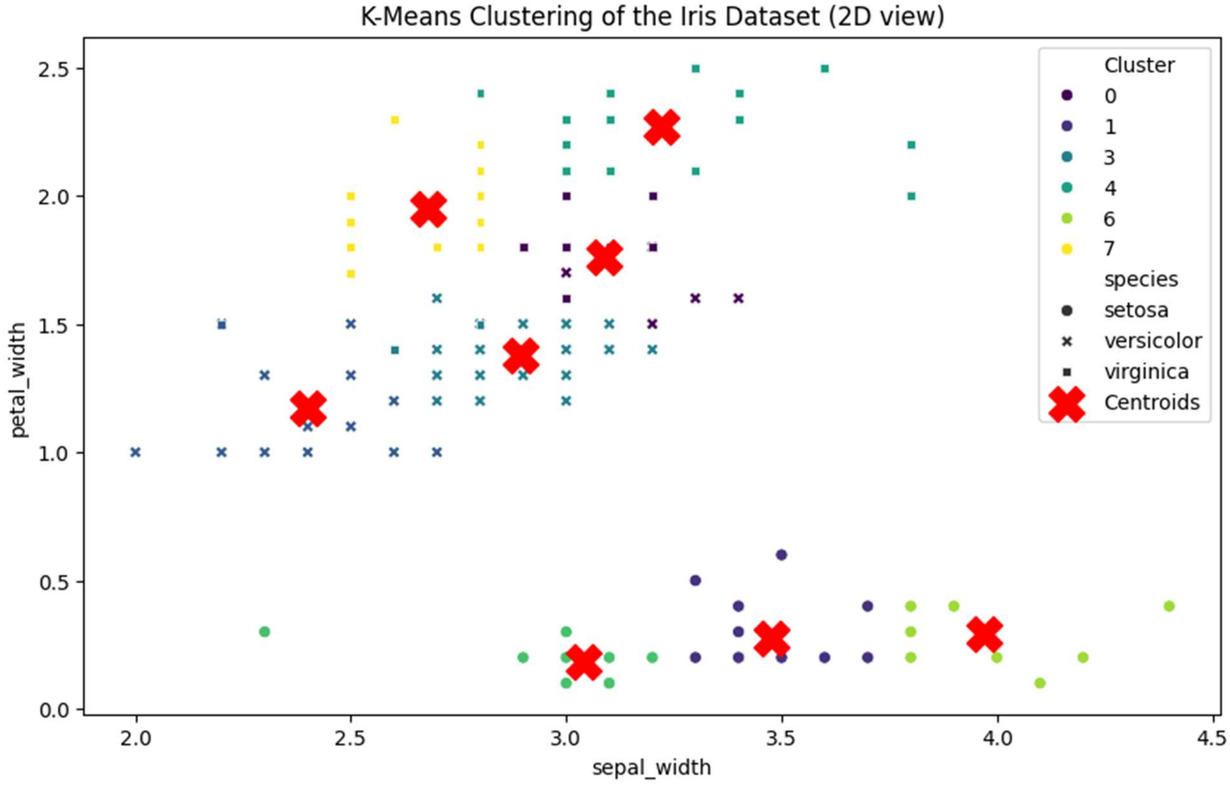


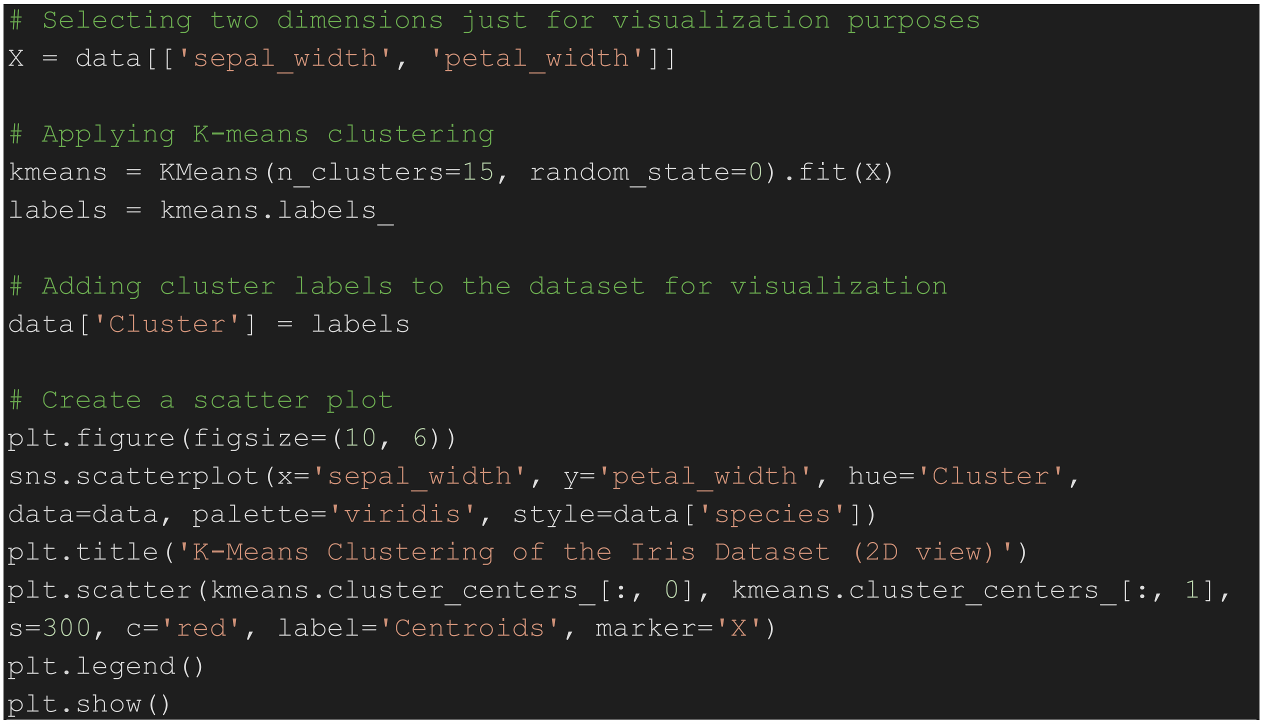
```
# Selecting two dimensions just for visualization purposes
X = data[['sepal_width', 'petal_width']]

# Applying K-means clustering
kmeans = KMeans(n_clusters=8, random_state=0).fit(X)
labels = kmeans.labels_

# Adding cluster labels to the dataset for visualization
data['Cluster'] = labels

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_width', y='petal_width', hue='Cluster',
data=data, palette='viridis', style=data['species'])
plt.title('K-Means Clustering of the Iris Dataset (2D view)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red', label='Centroids', marker='X')
plt.legend()
plt.show()
```



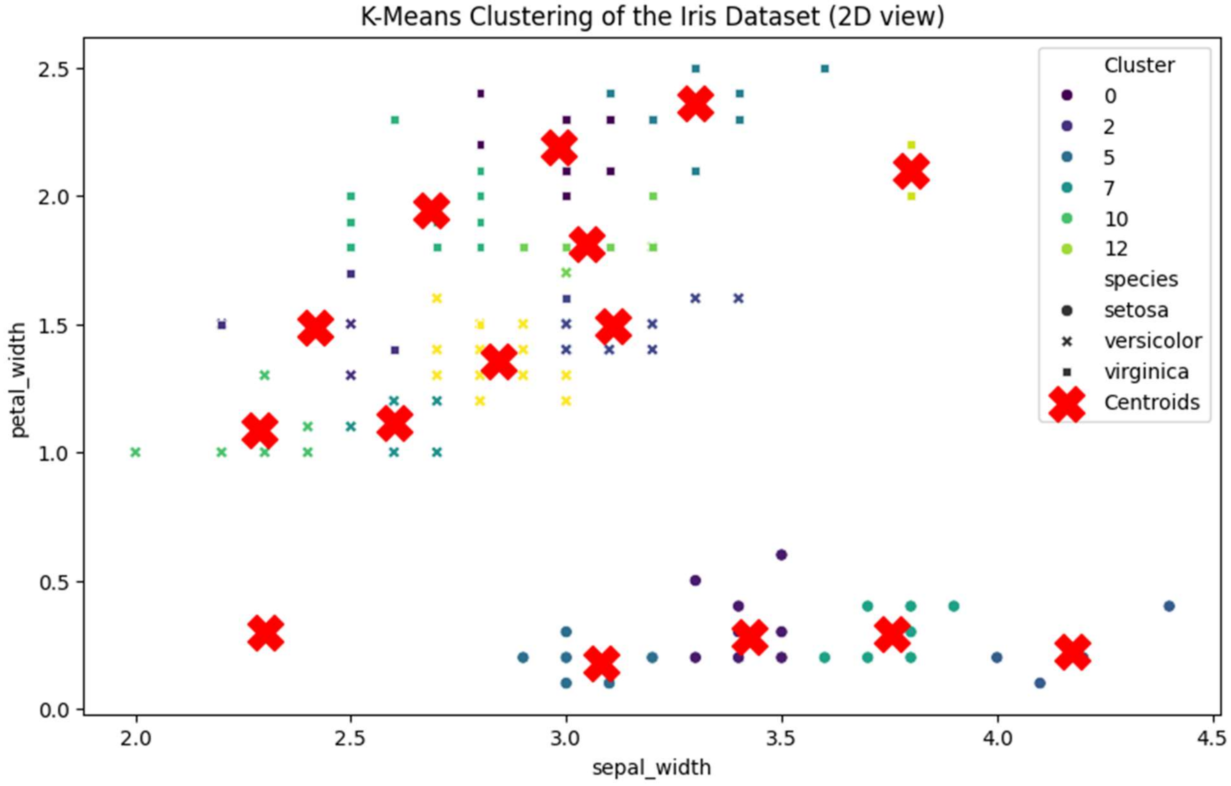


```
# Selecting two dimensions just for visualization purposes
X = data[['sepal_width', 'petal_width']]

# Applying K-means clustering
kmeans = KMeans(n_clusters=15, random_state=0).fit(X)
labels = kmeans.labels_

# Adding cluster labels to the dataset for visualization
data['Cluster'] = labels

# Create a scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x='sepal_width', y='petal_width', hue='Cluster',
data=data, palette='viridis', style=data['species'])
plt.title('K-Means Clustering of the Iris Dataset (2D view)')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
s=300, c='red', label='Centroids', marker='X')
plt.legend()
plt.show()
```



4 clusters appear to be the ideal number of clusters to interpret the data set and group it to compare the species listed.