In [2]:
```
#Author:Nichole Etienne
#Date: Wednesday September 23, 2021
#This code is meant to apply the unsupervised K-means Algorithm to the
```

In [28]:
```
#import the required Libraries
## matplotlib.pyplot: a collection of command style functions that mak
#pandas: data analysis tool kit
#Seaborn: library for making statistical graphics in Python
#warning :ignore warnings
#StandardScaler: for Standardize features
# KMeans: for kmeans clustering
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
```

In [29]:
```
# Import the Facebook Live Sellers in the old Faithful Geyser Dataset
data= pd.read_csv('./Desktop/OFData.csv')
```

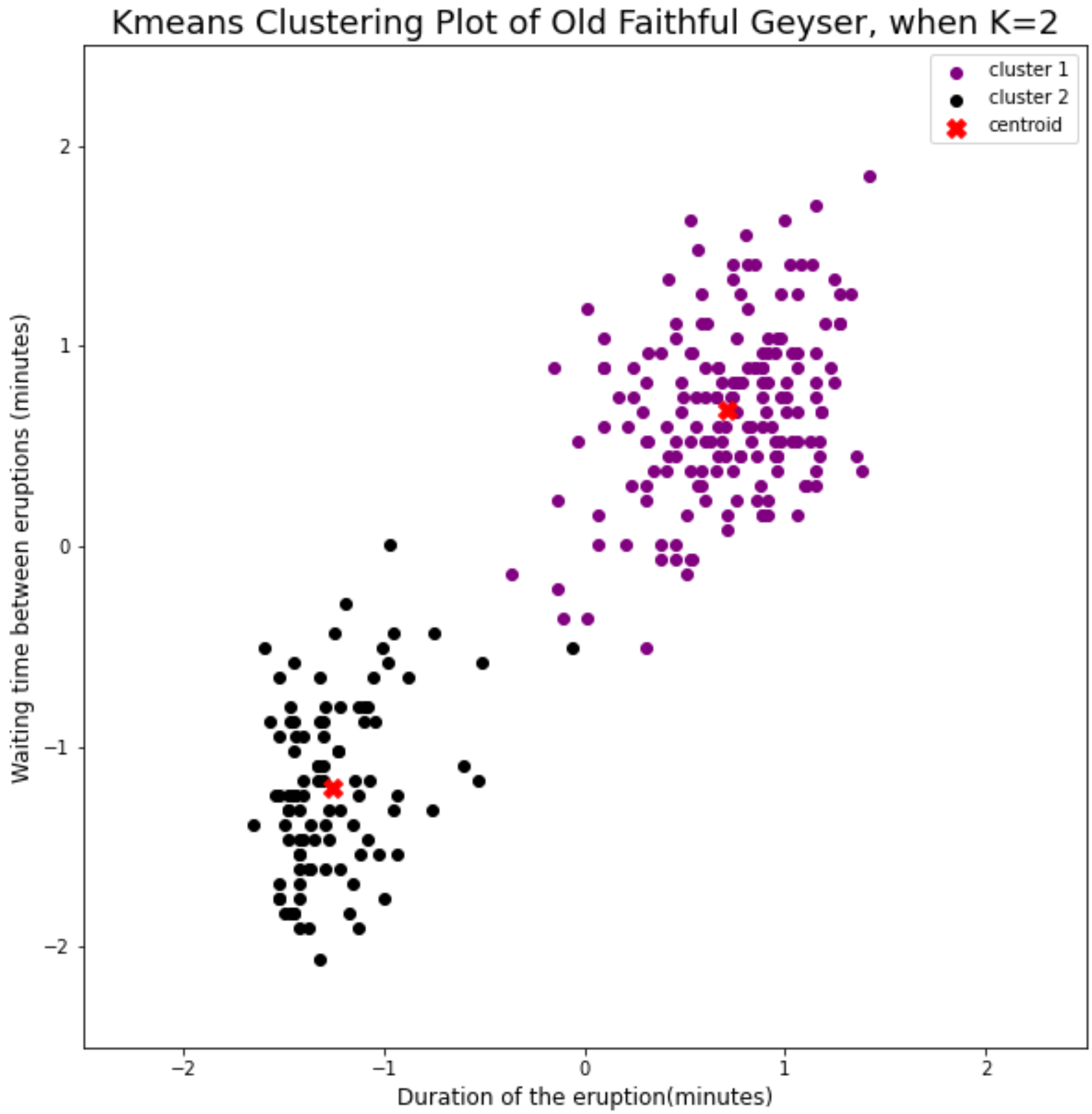
In [30]:
```
# Standardize features by removing the mean and scaling to unit varian
StandardizedData = StandardScaler().fit_transform(data)
```

In [31]:
```
#apply kmeans
kmeansalgorithm = KMeans(n_clusters=2, max_iter=100)
kmeansalgorithm.fit(StandardizedData)
# cluster_centers_ is called the code book and each value returned by
centroids = kmeansalgorithm.cluster_centers_
```

In [32]:
```
print (centroids)
```

```
[[ 0.70970327  0.67674488]
 [-1.26008539 -1.20156744]]
```

In [33]:
```
# Plot Kmeans Cluster of the Dataset k=2
fig, ax = plt.subplots(figsize=(10, 10))
plt.scatter(StandardizedData[kmeansalgorithm.labels_ == 0, 0], Standar
            c='purple', label='cluster 1')
plt.scatter(StandardizedData[kmeansalgorithm.labels_ == 1, 0], Standar
            c='black', label='cluster 2')
plt.scatter(centroids[:, 0], centroids[:, 1], marker='X', s=100,
            c='r', label='centroid')
plt.legend()
plt.xlim([-2.5, 2.5])
plt.ylim([-2.5, 2.5])
plt.xlabel('Duration of the eruption(minutes)', fontsize=12)
plt.ylabel('Waiting time between eruptions (minutes)', fontsize=12)
plt.title('Kmeans Clustering Plot of Old Faithful Geyser, when K=2',
ax.set_aspect('equal')
```



In [57]:

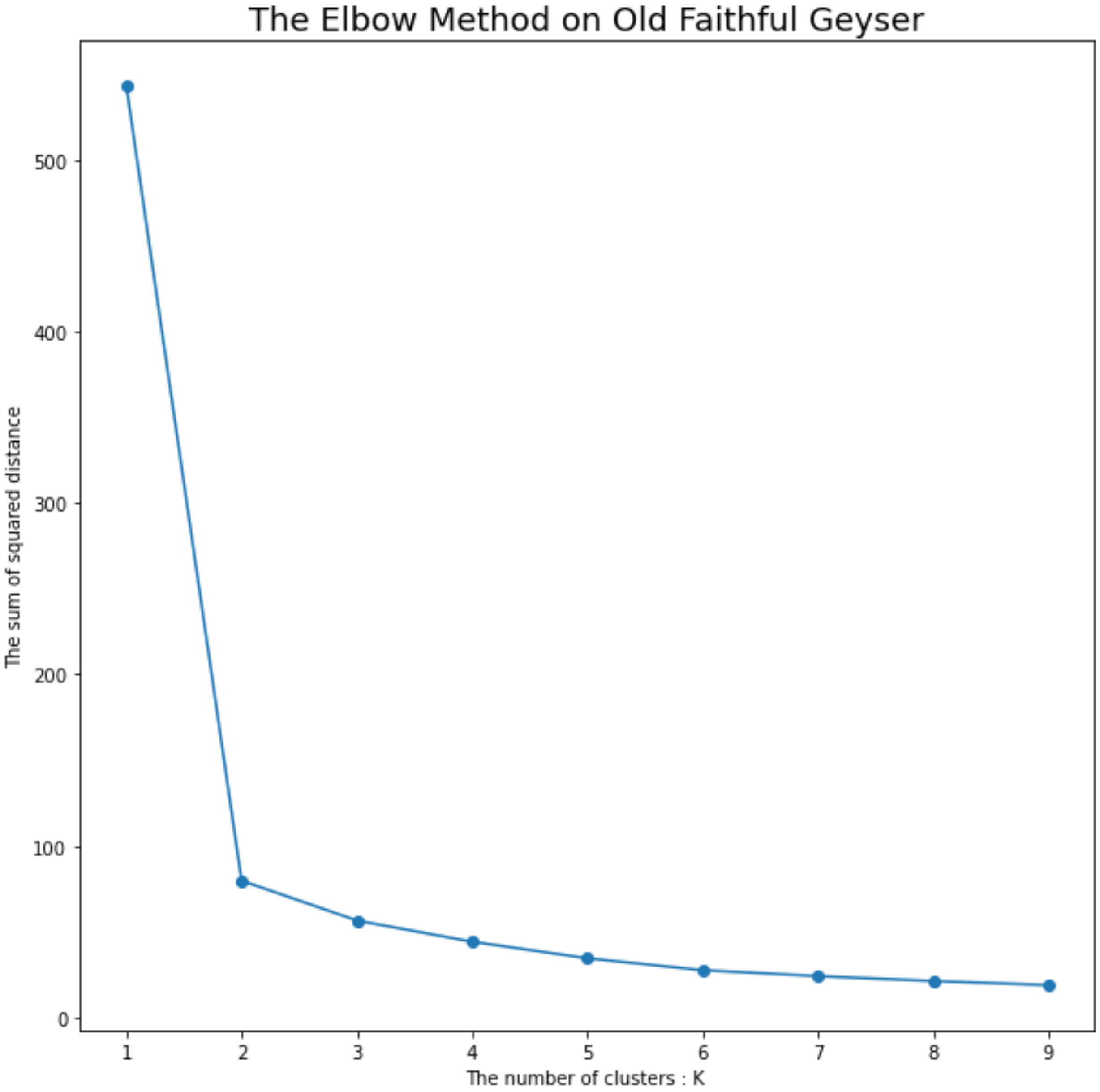
```
# Run the Elblow Method

WSS = []
Kvalues = list(range(1, 10))

for k in list_k:
    kmalgorithm = KMeans(n_clusters=k)
    kmalgorithm.fit(StandardizedData)
    WSS.append(kmalgorithm.inertia_)

# Plot The Sum of square erros and the Number of clusters : K
plt.figure(figsize=(10, 10))
plt.plot(Kvalues, WSS, '-o')
plt.xlabel(r'The number of clusters : K ')
plt.ylabel('The sum of squared distance')
plt.title('The Elbow Method on Old Faithful Geyser',  fontsize=18)
```

```
Out[57]: Text(0.5, 1.0, 'The Elbow Method on Old Faithful Geyser')
```



In [ ]:

In [ ]: