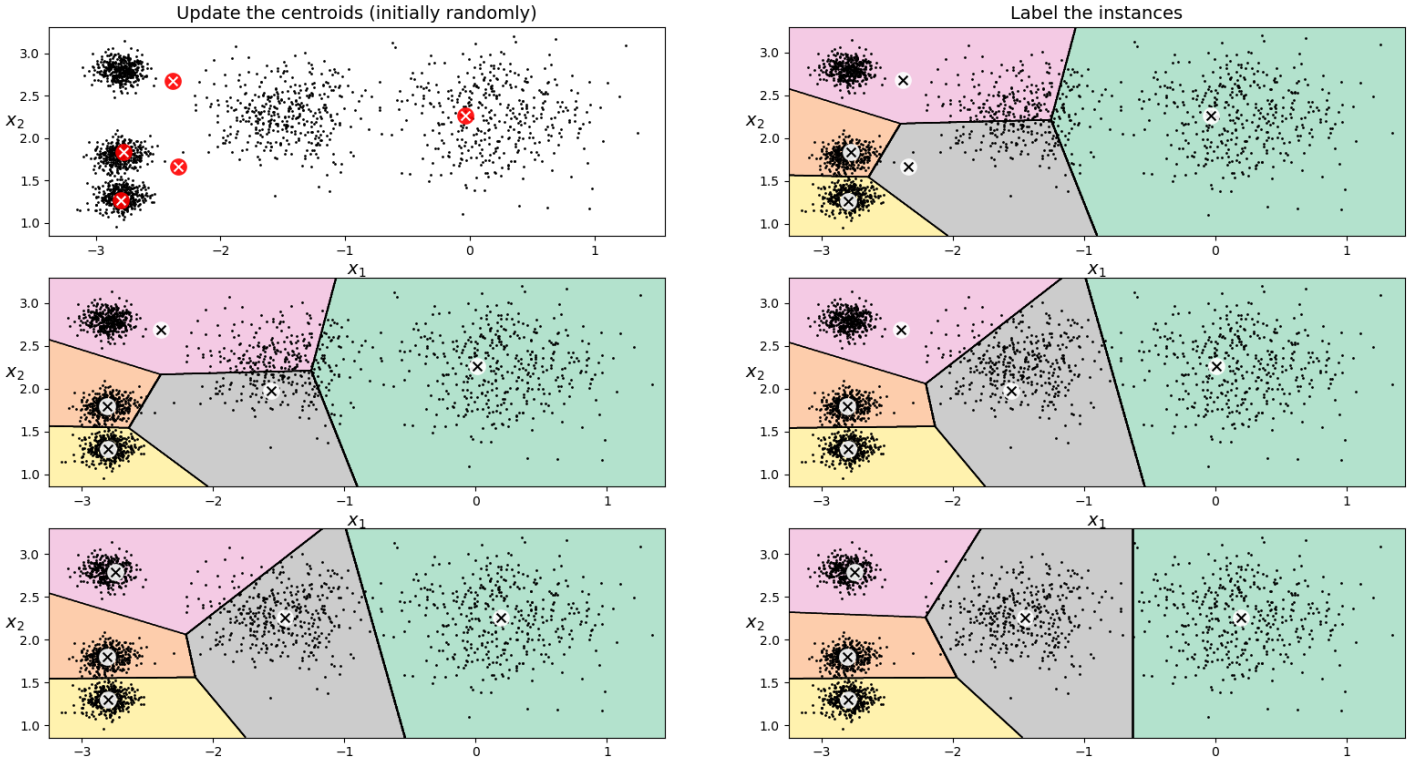
Q1-2. The Effect of Iteration on K-Means Clustering Results

```
import numpy as np
import os

import matplotlib as mpl
import matplotlib.pyplot as plt

import sklearn
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans

##########################################################
### The Effect of Iteration on K-Means Clustering Results ###
##########################################################

### Prepare blob dataset ###########################################################################################################################################

# Blob centers
blob_centers = np.array([[ 0.2,  2.3],
                         [-1.5 ,  2.3],
                         [-2.8,  1.8],
                         [-2.8,  2.8],
                         [-2.8,  1.3]])

# Blob standard deviation to define the distribution of dataset
blob_std = np.array([0.4, 0.3, 0.1, 0.1, 0.1])

# Produce the blob with centers and standard deviation
X, y = make_blobs(n_samples=2000, centers=blob_centers,
                  cluster_std=blob_std, random_state=7)

### K-Means Clustering Fitting ###
# K-Means with 1 iteration
kmeans_iter1 = KMeans(n_clusters=5, init='random', n_init=1, algorithm='full', max_iter=1, random_state=9)

# K-Means with 2 iterations
kmeans_iter2 = KMeans(n_clusters=5, init='random', n_init=1, algorithm='full', max_iter=2, random_state=9)

# K-Means with 3 iterations
kmeans_iter3 = KMeans(n_clusters=5, init='random', n_init=1, algorithm='full', max_iter=3, random_state=9)

kmeans_iter1.fit(X)     # Fit K-Means with 1 iteration
kmeans_iter2.fit(X)     # Fit K-Means with 2 iterations
kmeans_iter3.fit(X)     # Fit K-Means with 3 iterations

### K-Means Clustering Result Plotting ###############################################################################################################################
plt.figure(figsize=(8, 4))

# Segment K-Means clusters by clustering all the points within X range and Y range
# Acquire the value range of x1 and x2
mins = X.min(axis=0) - 0.1
maxs = X.max(axis=0) + 0.1

# Acquire all the (x1, x2) points within the range
xx, yy = np.meshgrid(np.linspace(mins[0], maxs[0], 1000),
                     np.linspace(mins[1], maxs[1], 1000))

# Subplot 1 ##########################################################################################################################################################
plt.subplot(321)
plt.title("Update the centroids (initially randomly)", fontsize=14)

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter1.cluster_centers_[:, 0], kmeans_iter1.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='r', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter1.cluster_centers_[:, 0], kmeans_iter1.cluster_centers_[:, 1], marker='x', s=50, color='w', zorder=11, alpha=1)

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)

# Subplot 2 ##########################################################################################################################################################
plt.subplot(322)
plt.title("Label the instances", fontsize=14)

# Cluster all the (x1, x2) points within the range / Cluster labels are used as height of contour
Z = kmeans_iter1.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)                             # Reshape for plotting

# Color all the (x1, x2) points with height according to cluster label
plt.contourf(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), cmap="Pastel2")

# Connect and draw the contour lines
plt.contour(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), linewidths=1, colors='k')

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)     # Plot the data on the contour

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter1.cluster_centers_[:, 0], kmeans_iter1.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='w', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter1.cluster_centers_[:, 0], kmeans_iter1.cluster_centers_[:, 1], marker='x', s=50, color='k', zorder=11, alpha=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)


# Subplot 3 ##########################################################################################################################################################
plt.subplot(323)

# Cluster all the (x1, x2) points within the range / Cluster labels are used as height of contour
Z = kmeans_iter1.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)                             # Reshape for plotting

# Color all the (x1, x2) points with height according to cluster label
plt.contourf(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), cmap="Pastel2")

# Connect and draw the contour lines
plt.contour(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), linewidths=1, colors='k')

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)     # Plot the data on the contour

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter2.cluster_centers_[:, 0], kmeans_iter2.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='w', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter2.cluster_centers_[:, 0], kmeans_iter2.cluster_centers_[:, 1], marker='x', s=50, color='k', zorder=11, alpha=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)


# Subplot 4 ##########################################################################################################################################################
plt.subplot(324)

# Cluster all the (x1, x2) points within the range / Cluster labels are used as height of contour
Z = kmeans_iter2.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)                             # Reshape for plotting

# Color all the (x1, x2) points with height according to cluster label
plt.contourf(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), cmap="Pastel2")

# Connect and draw the contour lines
plt.contour(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), linewidths=1, colors='k')

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)     # Plot the data on the contour

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter2.cluster_centers_[:, 0], kmeans_iter2.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='w', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter2.cluster_centers_[:, 0], kmeans_iter2.cluster_centers_[:, 1], marker='x', s=50, color='k', zorder=11, alpha=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)
```

```
# Subplot 5 ##################################################################################################################################################
plt.subplot(325)

# Cluster all the (x1, x2) points within the range / Cluster labels are used as height of contour
Z = kmeans_iter2.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)                          # Reshape for plotting

# Color all the (x1, x2) points with height according to cluster label
plt.contourf(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), cmap="Pastel2")

# Connect and draw the contour lines
plt.contour(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), linewidths=1, colors='k')

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)     # Plot the data on the contour

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter3.cluster_centers_[:, 0], kmeans_iter3.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='w', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter3.cluster_centers_[:, 0], kmeans_iter3.cluster_centers_[:, 1], marker='x', s=50, color='k', zorder=11, alpha=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)


# Subplot 6 ##################################################################################################################################################
plt.subplot(326)

# Cluster all the (x1, x2) points within the range / Cluster labels are used as height of contour
Z = kmeans_iter3.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)                          # Reshape for plotting

# Color all the (x1, x2) points with height according to cluster label
plt.contourf(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), cmap="Pastel2")

# Connect and draw the contour lines
plt.contour(Z, extent=(mins[0], maxs[0], mins[1], maxs[1]), linewidths=1, colors='k')

plt.scatter(X[:, 0], X[:, 1], c='k', s=1)     # Plot the data on the contour

# Draw the centroids of K-Means clustering results
plt.scatter(kmeans_iter3.cluster_centers_[:, 0], kmeans_iter3.cluster_centers_[:, 1], marker='o', s=30, linewidths=8, color='w', zorder=10, alpha=0.9)
plt.scatter(kmeans_iter3.cluster_centers_[:, 0], kmeans_iter3.cluster_centers_[:, 1], marker='x', s=50, color='k', zorder=11, alpha=1)

plt.xlabel("$x_1$", fontsize=14)
plt.ylabel("$x_2$", fontsize=14, rotation=0)
plt.show()
```



- **최초 Cluster 의 중심점으로 사용될 좌표 (Codebook, Centroid) 는 랜덤하게 생성됨**
- **랜덤하게 생성된 Cluster 중심을 기준으로 모든 데이터는 최초 Cluster Label 을 가짐 (1,2 Subplot 결과)**
- **각 Cluster 중심점은 데이터와의 거리를 최소화 시키기 위해 자신에게 배정된 데이터의 Mean 의 위치로 배치됨 (2, 1 Subplot 결과)**
- **각 Cluster 중심점이 새로운 위치에 도착한 후 모든 데이터를 자신과 제일 가까운 Cluster 중심점으로 재배치됨 (2, 2 Subplot 결과)**
- **‘최단 Cluster 중심점에 대한 데이터 Label 배치 ⟶ 중심점 Mean 지점 이동’ 반복하면서 Cluster 의 중심점이 데이터가 밀집된 곳으로 점차적으로 이동하게 되고 , 그에 따라 데이터가 Cluster 중심을 기준으로 Labeling 되는 것을 볼 수 있**