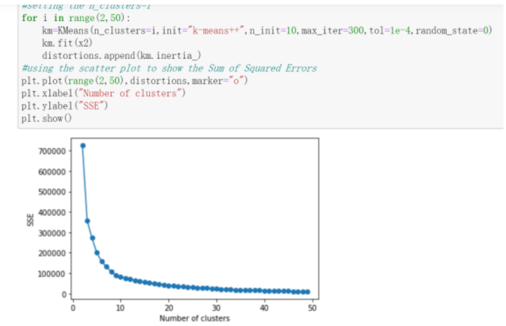
Assignment02

Part 1

1. Using the Sum of Squares Error to calculate the number of clusters.
As shown in the figure 1 below:

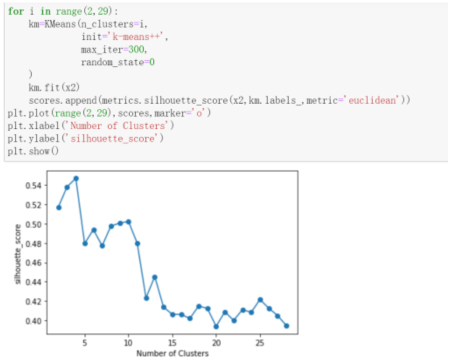


```
#setting the n_clusters=i
for i in range(2,50):
    km=KMeans(n_clusters=i,init="k-means++",n_init=10,max_iter=300,tol=1e-4,random_state=0)
    km.fit(x2)
    distortions.append(km.inertia_)
#using the scatter plot to show the Sum of Squared Errors
plt.plot(range(2,50),distortions,marker="o")
plt.xlabel("Number of clusters")
plt.ylabel("SSE")
plt.show()
```

Figure1

As we can see that the value of clusters equals 8, the curve shows an inflection point. After that, the curve tends to smooth.

2. Using the Silhouette score to calculate the number of clusters. As shown in the figure 2 below:

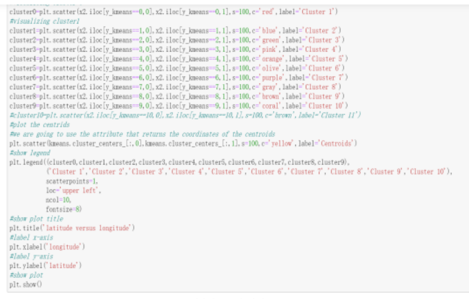


```
for i in range(2,29):
    km=KMeans(n_clusters=i,
              init='k-means++',
              max_iter=300,
              random_state=0
    )
    km.fit(x2)
    scores.append(metrics.silhouette_score(x2,km.labels_,metric='euclidean'))
plt.plot(range(2,29),scores,marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('silhouette_score')
plt.show()
```

Figure2

We set the initial value of clusters is 2, and the highest point is reached when the cluster value equals 4 from figure2. But the number of clusters equals 4 ,which not our best parameters. As observed from figure1, the SSE curve is not region smoothed at the cluster value of 4. When the cluster value equals 10 from figure2, the second high point is reached. At the same time, the SSE curve tends to smooth in the figure1.

3. The visualization of clusters on a scatter plot. As shown in the figure 4 below:



```
cluster0=plt.scatter(x2.iloc[y_kmeans==0,0],x2.iloc[y_kmeans==0,1],s=100,c='red',label='Cluster 1')
#visualizing cluster1
cluster1=plt.scatter(x2.iloc[y_kmeans==1,0],x2.iloc[y_kmeans==1,1],s=100,c='blue',label='Cluster 2')
cluster2=plt.scatter(x2.iloc[y_kmeans==2,0],x2.iloc[y_kmeans==2,1],s=100,c='green',label='Cluster 3')
cluster3=plt.scatter(x2.iloc[y_kmeans==3,0],x2.iloc[y_kmeans==3,1],s=100,c='pink',label='Cluster 4')
cluster4=plt.scatter(x2.iloc[y_kmeans==4,0],x2.iloc[y_kmeans==4,1],s=100,c='orange',label='Cluster 5')
cluster5=plt.scatter(x2.iloc[y_kmeans==5,0],x2.iloc[y_kmeans==5,1],s=100,c='olive',label='Cluster 6')
cluster6=plt.scatter(x2.iloc[y_kmeans==6,0],x2.iloc[y_kmeans==6,1],s=100,c='purple',label='Cluster 7')
cluster7=plt.scatter(x2.iloc[y_kmeans==7,0],x2.iloc[y_kmeans==7,1],s=100,c='gray',label='Cluster 8')
cluster8=plt.scatter(x2.iloc[y_kmeans==8,0],x2.iloc[y_kmeans==8,1],s=100,c='brown',label='Cluster 9')
cluster9=plt.scatter(x2.iloc[y_kmeans==9,0],x2.iloc[y_kmeans==9,1],s=100,c='coral',label='Cluster 10')
#cluster10=plt.scatter(x2.iloc[y_kmeans==10,0],x2.iloc[y_kmeans==10,1],s=100,c='brown',label='Cluster 11')
#plot the centrids
#we are going to use the attribute that returns the coordinates of the centroids
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=100,c='yellow',label='Centroids')
#show legend
plt.legend((cluster0,cluster1,cluster2,cluster3,cluster4,cluster5,cluster6,cluster7,cluster8,cluster9),
          ('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Cluster 7','Cluster 8','Cluster 9','Cluster 10'),
          scatterpoints=1,
          loc='upper left',
          ncol=10,
          fontsize=8)
#show plot title
plt.title('latitude versus longitude')
#label x-axis
plt.xlabel('longitude')
#label y-axis
plt.ylabel('latitude')
#show plot
plt.show()
```

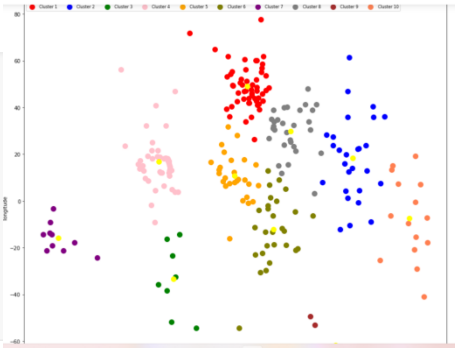
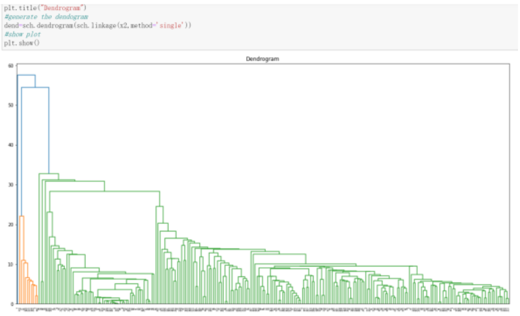


Figure3 Figure4

Cluster	Centroid	Continent
0	11.04527028E ,49.13415273N	Europe
1	108.62935637E ,18.33127919N	Asia
2	-57.62434537, -33.3509395	South America
3	-71.06985896,16.78186493	North America
4	-1.10167986 ,10.75386428	Western Africa
5	35.16229227 , -12.06182649	Southern Africa
6	-164.35100116 ,-15.84210807	Oceania
7	51.06670653 , 29.9210421	Western Asia
8	92.6175717 , -61.74164267	Antarctica
9	160.9867064 ,-7.36144245	Oceania

Part 2

1. Using the hierarchical clustering with setting the linkage of single.
As shown in the figure 5 below:



```
plt.title("Dendrogram")
#generate the dendogram
dend=sch.dendrogram(sch.linkage(x2,method='single'))
#show plot
plt.show()
```

Figure5

As we can observe that the dendrogram is unbalanced from the Figure5. This indicates that the linkage of single is not well-for cluster.

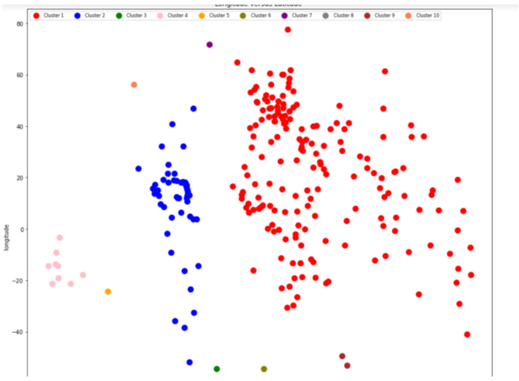
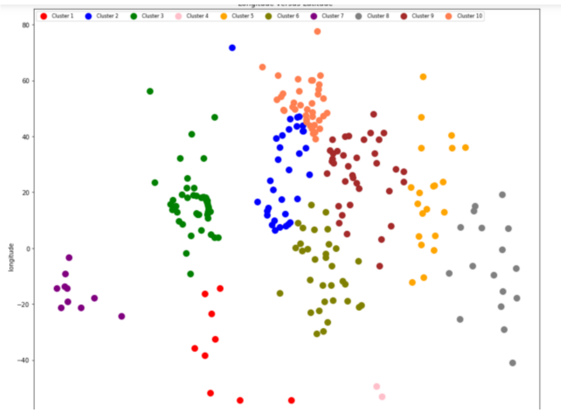
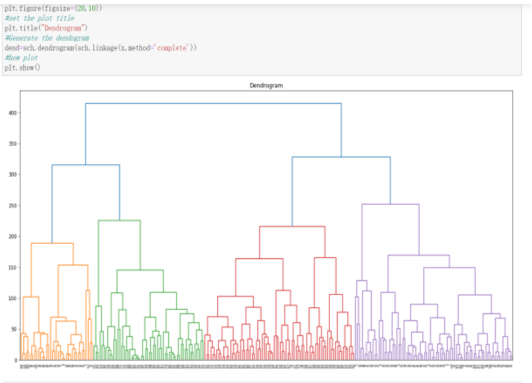


Figure6

From the scatter plot above, we observe that using the linkage “single” produces bad clusters as most of the data points are grouped into one cluster.(cluster1 includes some points that do not belong to it)

Using the hierarchical clustering with setting the linkage “complete”. As shown in the figure 7 and figure8 below:





```
plt.figure(figsize=(20,10))
#set the plot title
plt.title("Dendrogram")
#Generate the dendogram
dend=sch.dendrogram(sch.linkage(x,method='complete'))
#how plot
plt.show()
```

Figure7 Figure8

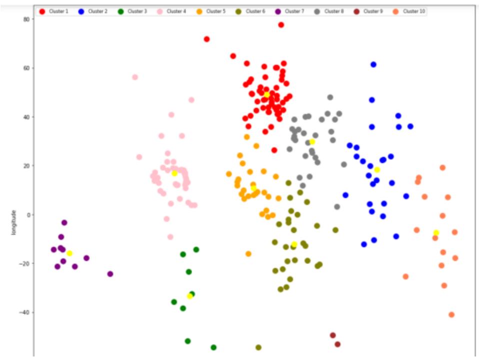
From the scatter plot above, we observe that using the linkage “complete” produces good clusters groups.

As we can see that the dendrogram is balanced from the Figure8. This indicates that the complete of linkage is good for cluster. There are 4 groups in the dendrogram.

2.Using the Hierarchical clustering algorithm:

Cluster	Continent
0	North America
1	Southern Africa
2	North America
3	Antarctica
4	Asia
5	Southern Africa
6	Oceania
7	Western Africa
8	Western Asia
9	Europe

Part 3



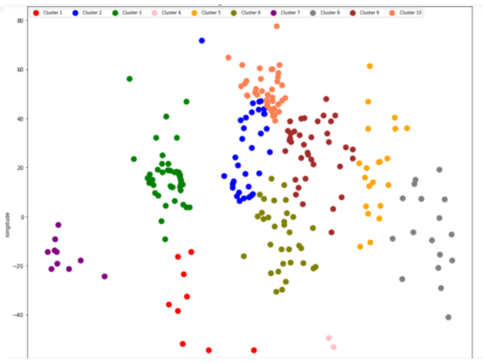


Figure 9(K-means clustering)

Figure10(Hierarchical clustering)

Figure9 :Using the K-means cluster algorithm and setting the number of n_clusters is 10

Figure10:Using the Hierarchical clustering algorithm and the complete of linkage.

As we can observe that using the k-means clustering could obtain better cluster groups than hierarchical clustering.

Calculate all samples and getting coordinates of cluster centers in the following table:

Cluster	Coordinates of cluster centers	Continent
0	39.54339501N, 5.17023238E	Europe-Spain
1	25.47397978N , 31.60266473E	Asia
3	22.8199807 , 18.16315455	North America
1	-4.80879576 ,110.58307336	Asia
2	7.31948212 , -78.96365931	South America
1	5.60873546 ,142.49530159	Asia
6	-15.03337581, -166.3133258	Oceania
2	9.65361715 , -65.60752127	South America
4	20.37021315 ,21.52163949	Africa
5	14.07084439 , 64.669801	Africa

Selecting the k-means clustering algorithm to provide the cluster number to the continent name.

Cluster	Continent	Countries name
0	Europe	Albania
1	Asia	Bangladesh
2	South America	Brazil
3	North America	Belize
4	Western Africa	Benin
5	Southern Africa	Angola
6	Oceania	American Samoa
7	Western Asia	Afghanistan
8	Antarctica	Heard & McDonald Islands
9	Oceania	Australia