Correction Activité 6 : CAH

```
from google.colab import drive
drive.mount('/content/gdrive')
#importation des données
import pandas
from matplotlib import pyplot as plt
from scipy.cluster.hierarchy import dendrogram, linkage
from sklearn.cluster import AgglomerativeClustering
from sklearn.preprocessing import MinMaxScaler
#lecture du fichier
fromage = pandas.read_table("/content/gdrive/MyDrive/ML/pratique/fromage.txt",sep="\t",
                    header=0,index_col=0)
# affichage des dimension des données
print(fromage.shape)
#affichage des nom des variables
print(fromage.columns)
#pour déterminer l'intervalle de variation de chacune des variables, il suffit d'afficher
#les statistiques sur les observations
print(fromage.describe())
#Réccupérer les valeurs des variables pour toutes les observations
X = fromage.values
#Changement de l'échelle  ==> nécessaire car les varaibles n'ont pas le même intervalle de variation
scaler = MinMaxScaler(feature_range=(0, 1))
X_normalise = scaler.fit_transform(X)
#---------- creation et affichage du dendrogramme
Z = linkage(X_normalise,method='single',metric='euclidean')
#tester différent types d'indices d'aggrégation ==>
                                    #changer 'ward' par 'single' , 'complete' , 'average'
                                    #commenter à chauqe fois les résultats obtenus
plt.title("CAH222")
dendrogram(Z,labels=fromage.index,orientation='left')
plt.show()
Z = linkage(X_normalise,method='ward',metric='euclidean')
#tester différent types d'indices d'aggrégation ==>
        #changer 'ward' par 'single' , 'complete' , 'average'
        #commenter à chauqe fois les résultats obtenus
plt.title("CAH")
dendrogram(Z,labels=fromage.index,orientation='left')
plt.show()
"""
==> k à choisir (selon le dendrogramme) :
   average  ==>  k=3 ou 4
   single ==> aglomération en échelle ==> non optimal k=2
   complete ==>  k=3 ou 4
   ward ==> k=5
```

"""

#------- creation des classes par la CHA
#------- en utilisant les deux premiers attributs
#selection des attributs
new_data = X_normalise[:,1:3]

#appliquer la CHA

k=3 # faire varier le k entre 2 et 5 ==> commenter à chaque fois les résultats obtenus

##tester différent types d'indices d'aggrégation
  #changer 'ward' par complete' , 'average' ==> commenter à chaque fois les résultats obtenus
y_hc = AgglomerativeClustering(n_clusters=3, metric = 'euclidean', linkage = 'ward').fit_predict(new_data)

# affichage du résultat
plt.scatter(new_data[y_hc ==0,0], new_data[y_hc == 0,1], s=20, c='r')
plt.scatter(new_data[y_hc ==1,0], new_data[y_hc == 1,1], s=20, c='m')
plt.scatter(new_data[y_hc ==2,0], new_data[y_hc == 2,1], s=20, c='y')
plt.scatter(new_data[y_hc ==3,0], new_data[y_hc == 3,1], s=20, c='b')
#plt.scatter(new_data[y_hc ==4,0], new_data[y_hc == 4,1], s=20, c='g')
plt.show()