K-means clustering in R

Program :

```
setwd("C:/RStudio/test")
data<-read.csv("movies_metadata.csv")
library(dplyr)
data[, 3] <- as.numeric(as.character( data[, 3] ))
data[, 11] <- as.numeric(as.character( data[, 11] ))
data1 <- filter(data, budget > 26999, revenue > 0, runtime > 0, popularity > 0)
data1<-select(data1, budget, revenue, runtime,popularity)
library(VIM)
aggr(data1)
set.seed(20)
clusters <- kmeans(data1[,2:3], 5)
data1$rtime_rev <- as.factor(clusters$cluster)
clusters <- kmeans(data1[,1:2], 5)
data1$budg_rev <- as.factor(clusters$cluster)
str(clusters)
head(data1, n=10)
clusters <- kmeans(data1[,2:4], 5)
data1$runt_rev_pop <- as.factor(clusters$cluster)
head(data1, n=10)
```

Output :

budget	revenue	runtime	popularity	rtime_rev	budg_rev	runt_rev_pop
3.0e+07	373554033	81	21.946943	1	3	5
6.5e+07	262797249	104	17.015539	1	3	5
1.6e+07	81452156	127	3.859495	2	5	1
6.0e+07	187436818	170	17.924927	2	4	1
3.5e+07	64350171	106	5.231580	5	5	4
5.8e+07	352194034	130	14.686036	1	3	5
6.2e+07	107879496	106	6.318445	2	4	1
4.4e+07	13681765	192	5.092000	5	5	4
9.8e+07	10017322	119	7.284477	5	5	4
5.2e+07	116112375	178	10.137389	2	4	1

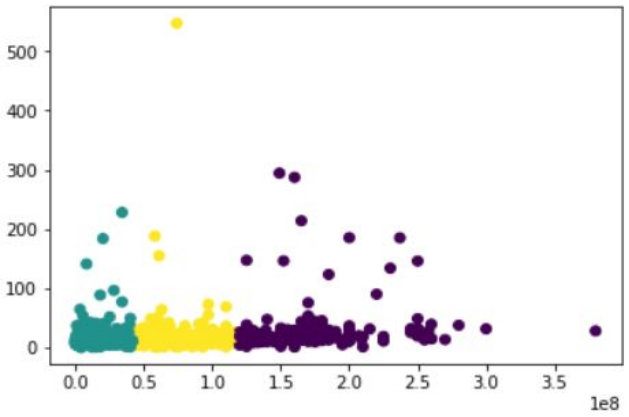
K-means clustering in Python

Program :

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('movies_metadata.csv')
df=df[["budget","popularity","vote_average","vote_count"]]
df=df.dropna()
df['budget'] = df['budget'].astype(float)
df=df[df.budget>29000]
df=df[df.vote_count>200]
df=df.drop(columns=["vote_count"])
from sklearn.cluster import KMeans
kmeans=KMeans(n_clusters=3,max_iter=600,algorithm='auto',random_state=20)
kmeans.fit(df)
x=kmeans.fit_predict(df)
df['cluster']=x
df2=df.iloc[:,0]
df3=df.iloc[:,1]
plt.scatter(df2,df3,c=x)
```

Output :

	budget	popularity	vote_average	cluster
0	30000000.0	21.9469	7.7	1
1	65000000.0	17.0155	6.9	2
5	60000000.0	17.9249	7.7	2
9	58000000.0	14.686	6.6	2
15	52000000.0	10.1374	7.8	2
16	16500000.0	10.6732	7.2	1
17	4000000.0	9.02659	6.5	1
18	30000000.0	8.20545	6.1	1

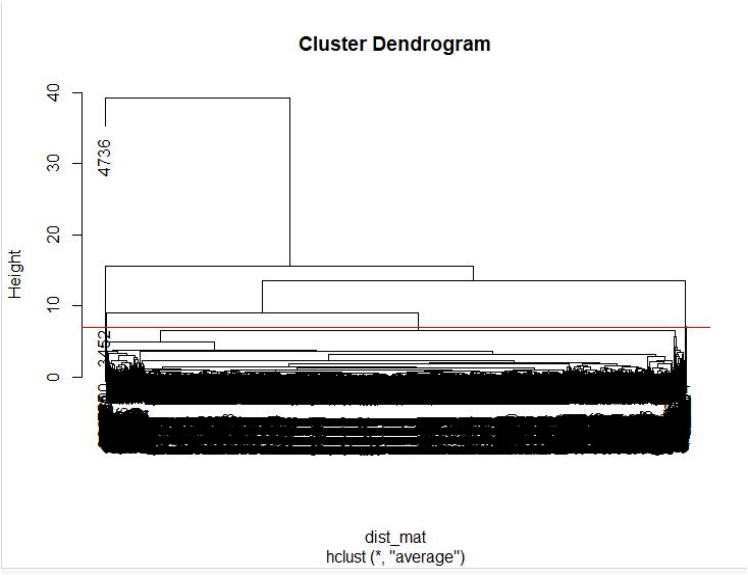


Hierarchical-clustering in R

Program :

```
setwd("C:/RStudio/test")
data<-read.csv("movies_metadata.csv")
summary(data)
library(dplyr)
data[, 3] <- as.numeric(as.character( data[, 3] ))
data[, 11] <- as.numeric(as.character( data[, 11] ))
data2 <- filter(data, budget > 26999, revenue > 0, runtime > 0, popularity > 0)
data2<-select(data2, budget, revenue, runtime,popularity)
summary(data2)
data2_sc<- as.data.frame(scale(data2))
summary(data2_sc)
dist_mat <- dist(data2_sc, method = 'euclidean')
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)
cut_avg <- cutree(hclust_avg, k = 20)
plot(hclust_avg)
abline(h = 7,col='red')
data2_cl <- mutate(data2, cluster = cut_avg)
```

Output :



	budget	revenue	runtime	popularity	cluster
207	25000000	56505065	130	11.338194	1
208	12500000	12281551	120	17.189328	1
209	27000000	13670688	80	7.436001	1
210	18000000	476684675	103	0.702543	2
211	22000000	505000000	127	11.945397	2
212	28000000	504050219	90	16.357419	2
213	100000000	520000000	137	22.661695	2
214	22000000	424208848	181	11.654349	1

Hierarchical-clustering in Python

Program :

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('movies_metadata.csv')
df=df[["budget","popularity","vote_average","vote_count"]]
df=df.dropna()
#df.apply(lambda x : pd.factorize(x)[0]).corr(method='pearson',
min_periods=1)
df['budget'] = df['budget'].astype(float)
df['popularity'] = df['popularity'].astype(float)
df=df[df.budget>29000]
df=df[df.vote_count>200]
df=df.drop(columns=["vote_count"])
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
df = df.apply(le.fit_transform)
import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
dend = shc.dendrogram(shc.linkage(df, method='ward'))
```

Output :