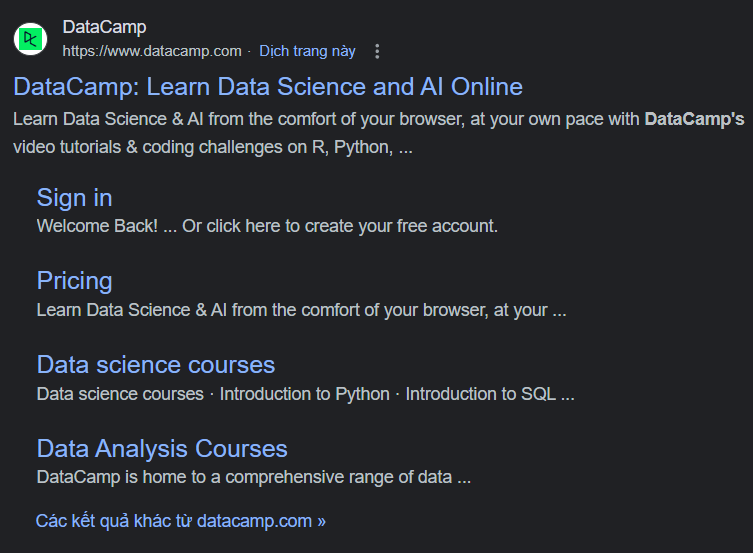
**Lab 9: Clustering in R**

**Họ và Tên**: Trần Nguyễn Gia Long

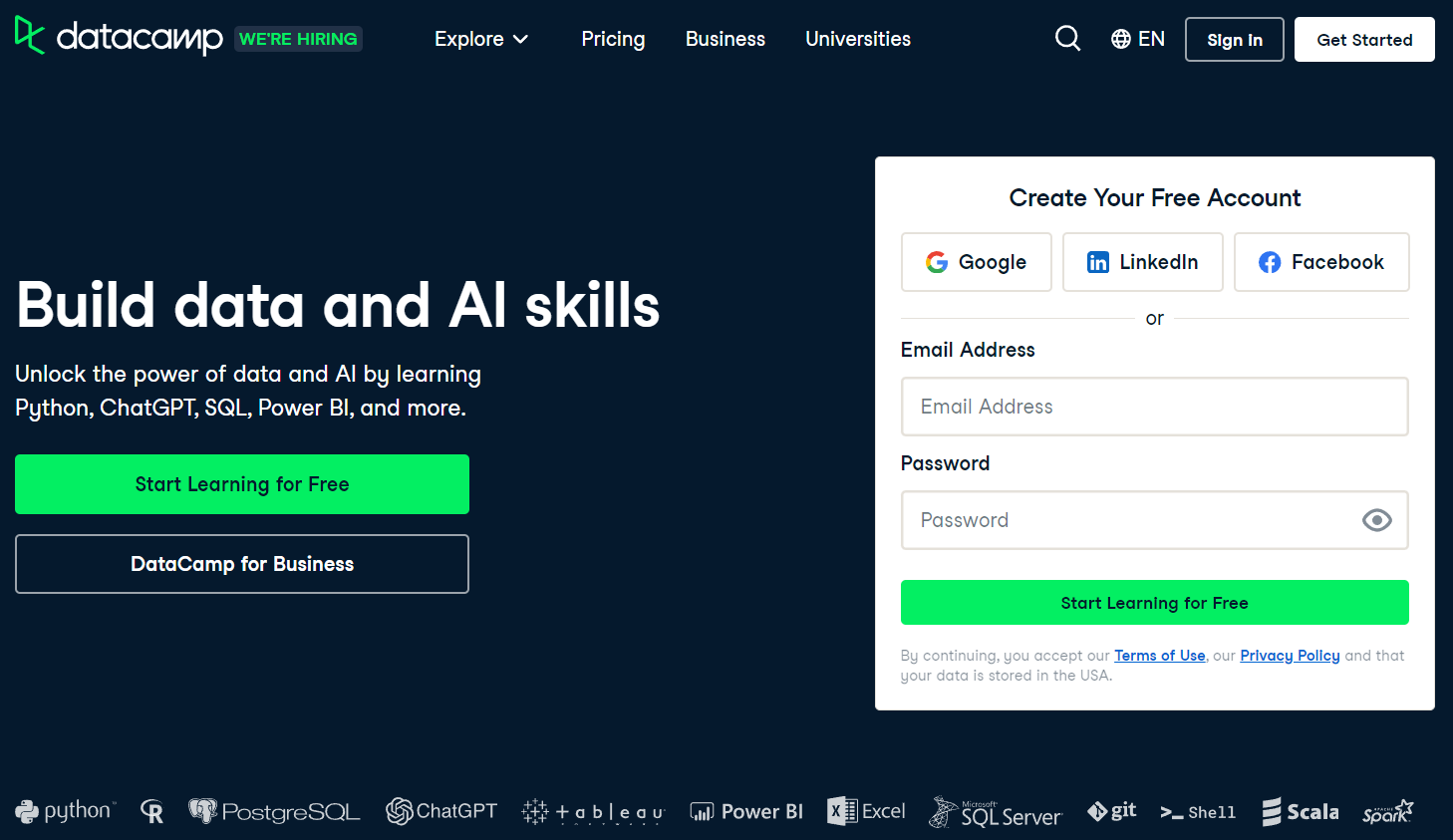
**MSSV**: 1050080059

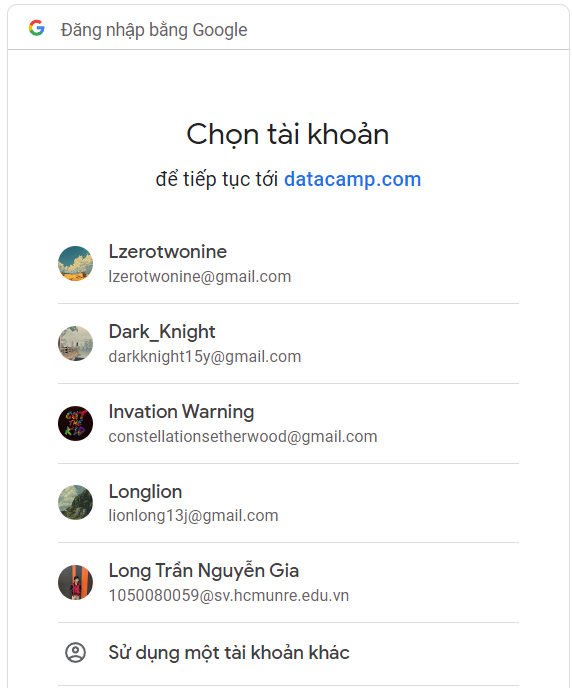
# 1. Cài đặt Clustering

- Sử dụng trang DataCamp để chạy Clustering.R.

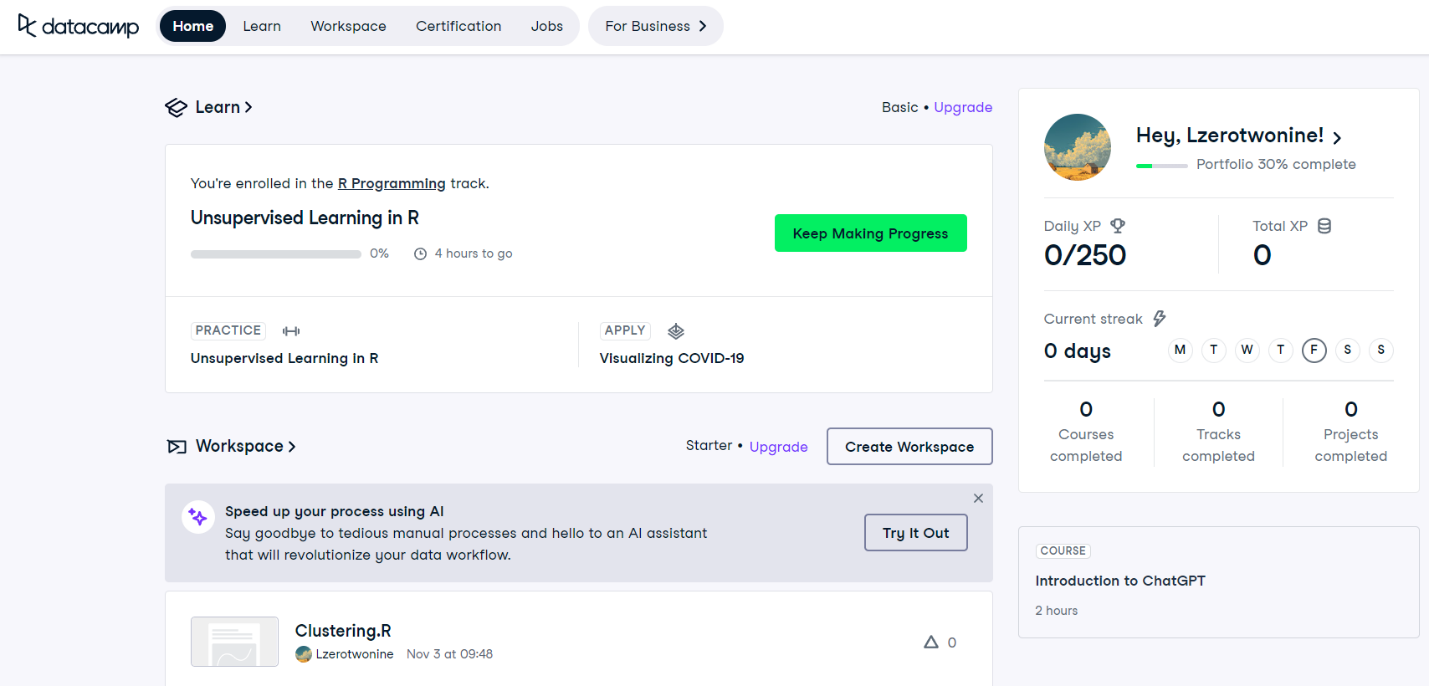


- Nhấp vào trang, tiến hành việc đăng nhập.

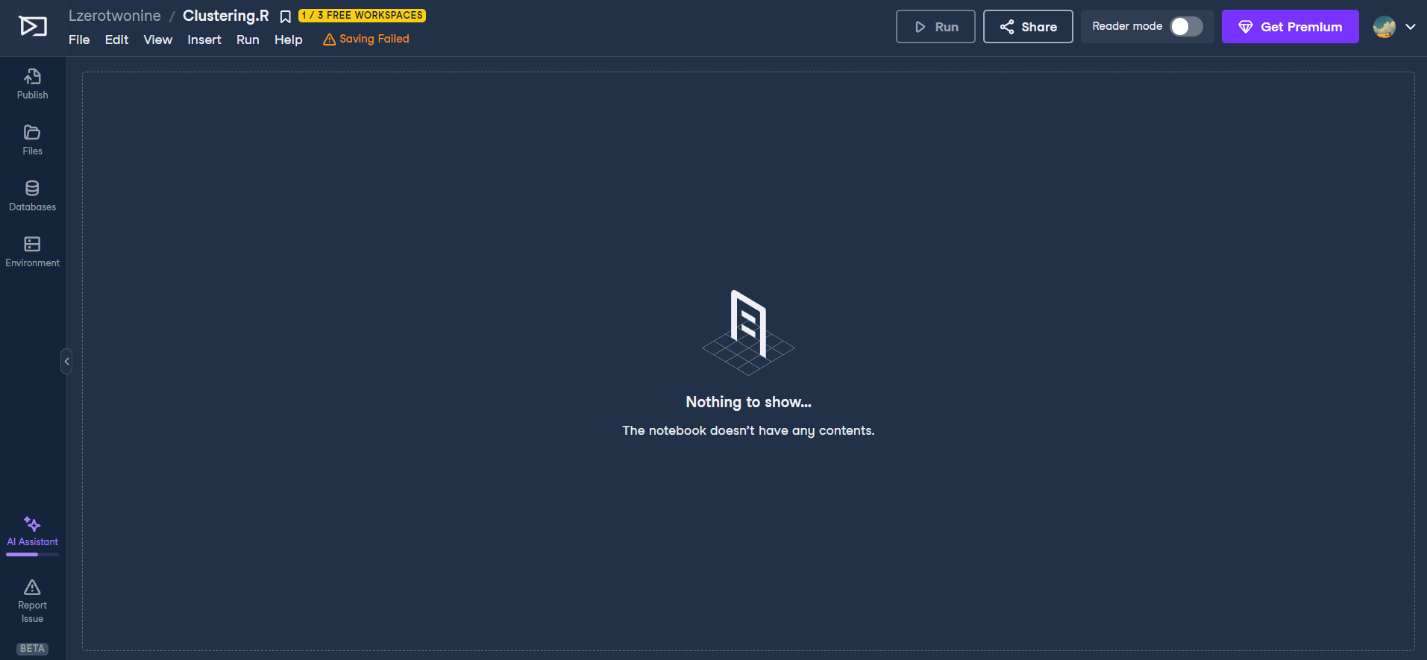




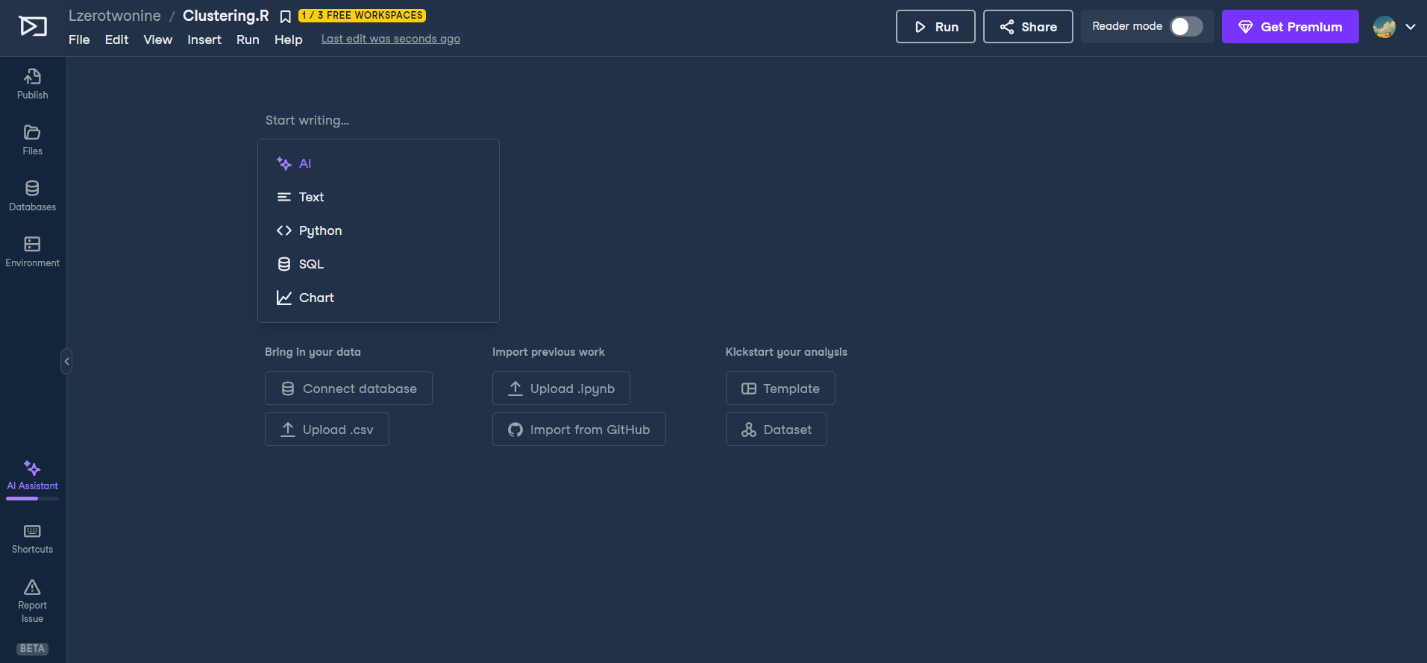
- Ở đây em chọn đăng nhập bằng tài khoản Google để vào nhanh chóng.

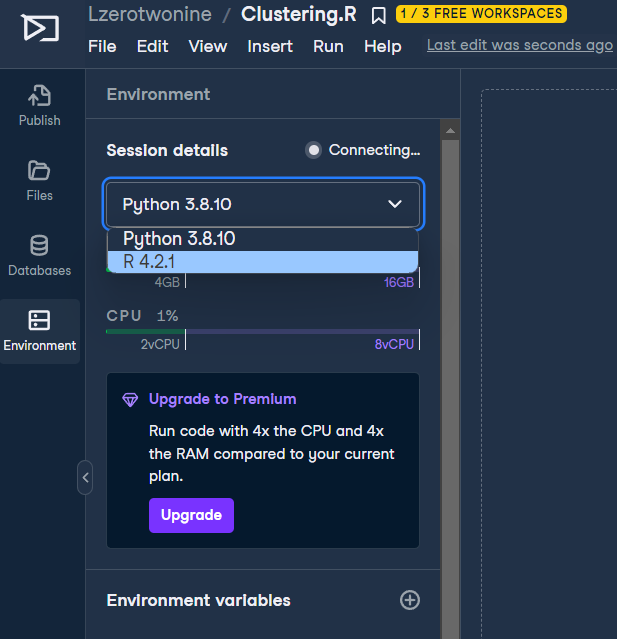


- Sau khi tạo tài khoản, ta bắt đầu tạo một workspace để dán và chạy code.



- Đây là giao diện của workspace.

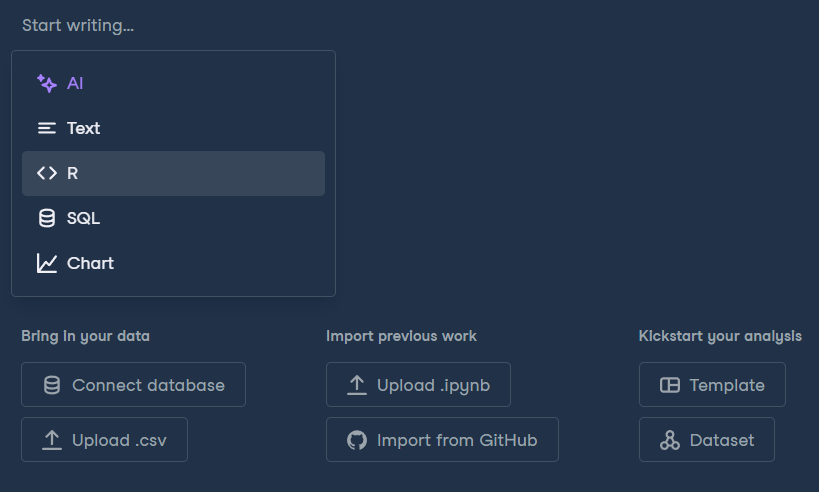
  
- Sau khi load lại, ở giữa có thêm các tuỳ chọn.



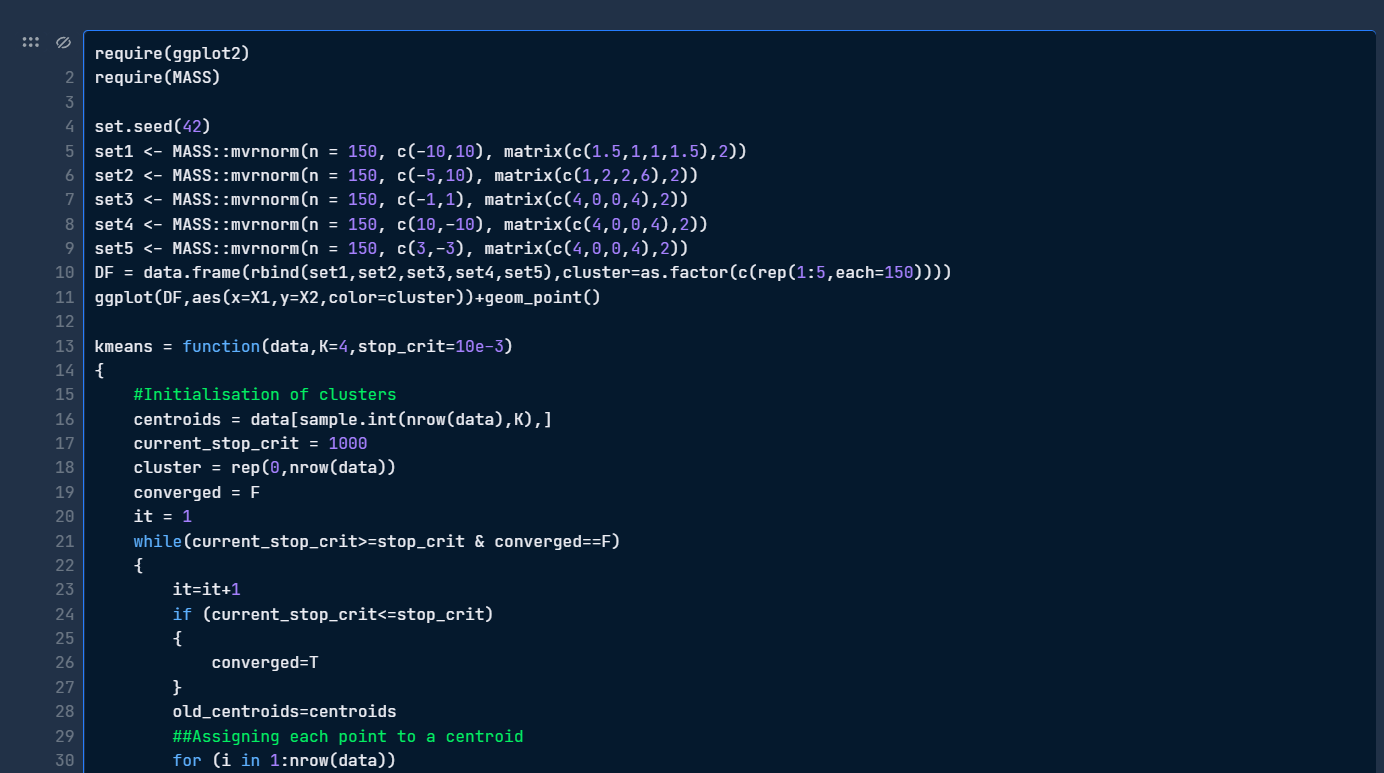
- Thay đổi ngôn ngữ lập trình thành R.

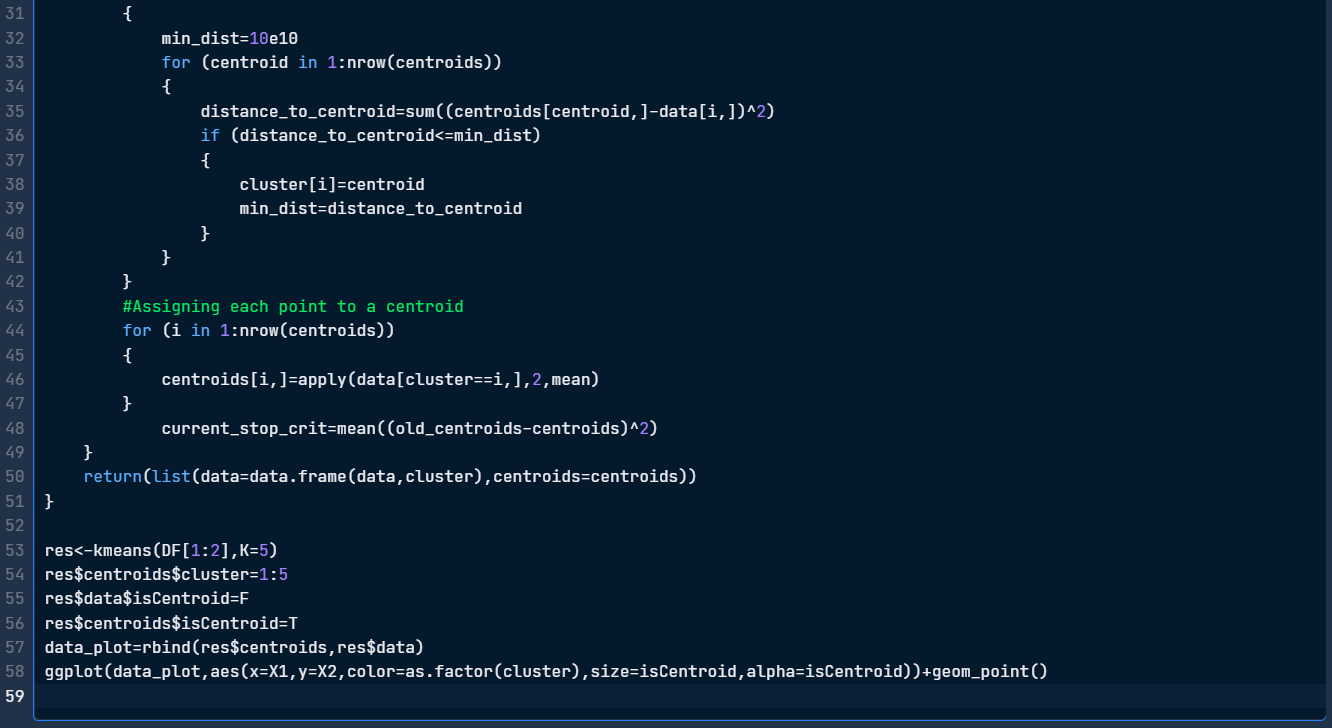
# 2. Chạy Clustering.

- Chọn viết code bằng R.



- Dán đoạn code từ bài Lab để chạy.





require(ggplot2)

require(MASS)

**1. Khởi tạo dữ liệu:** Đầu tiên, mã tạo ra 5 tập dữ liệu ngẫu nhiên với các phân phối chuẩn đa biến khác nhau. Mỗi tập dữ liệu đại diện cho một cụm. Sau đó, nó kết hợp tất cả các tập dữ liệu này thành một khung dữ liệu duy nhất.

set.seed(42)

set1 <- MASS::mvrnorm(n = 150, c(-10,10), matrix(c(1.5,1,1,1.5),2))

set2 <- MASS::mvrnorm(n = 150, c(-5,10), matrix(c(1,2,2,6),2))

set3 <- MASS::mvrnorm(n = 150, c(-1,1), matrix(c(4,0,0,4),2))

set4 <- MASS::mvrnorm(n = 150, c(10,-10), matrix(c(4,0,0,4),2))

set5 <- MASS::mvrnorm(n = 150, c(3,-3), matrix(c(4,0,0,4),2))

**2. Vẽ biểu đồ:** Mã sau đó vẽ biểu đồ phân tán của dữ liệu, với mỗi cụm được hiển thị bằng một màu khác nhau.

DF = data.frame(rbind(set1,set2,set3,set4,set5),cluster=as.factor(c(rep(1:5,each=150))))

ggplot(DF,aes(x=X1,y=X2,color=cluster))+geom\_point()

**3. Định nghĩa hàm K-means:** Mã tiếp tục bằng cách định nghĩa một hàm để thực hiện thuật toán K-means. Hàm này nhận vào dữ liệu và số lượng cụm mong muốn, sau đó thực hiện thuật toán K-means để phân chia dữ liệu thành các cụm.

kmeans = function(data,K=4,stop\_crit=10e-3)

{

#Initialisation of clusters

centroids = data[sample.int(nrow(data),K),]

current\_stop\_crit = 1000

cluster = rep(0,nrow(data))

converged = F

it = 1

while(current\_stop\_crit>=stop\_crit & converged==F)

{

it=it+1

if (current\_stop\_crit<=stop\_crit)

{

converged=T

}

old\_centroids=centroids

##Assigning each point to a centroid

for (i in 1:nrow(data))

{

min\_dist=10e10

for (centroid in 1:nrow(centroids))

{

distance\_to\_centroid=sum((centroids[centroid,]-data[i,])^2)

if (distance\_to\_centroid<=min\_dist)

{

cluster[i]=centroid

min\_dist=distance\_to\_centroid

}

}

}

#Assigning each point to a centroid

for (i in 1:nrow(centroids))

{

centroids[i,]=apply(data[cluster==i,],2,mean)

}

current\_stop\_crit=mean((old\_centroids-centroids)^2)

}

return(list(data=data.frame(data,cluster),centroids=centroids))

}

**4. Thực hiện K-means:** Mã sau đó gọi hàm K-means với dữ liệu và số lượng cụm là 5.

res<-kmeans(DF[1:2],K=5)

res$centroids$cluster=1:5

res$data$isCentroid=F

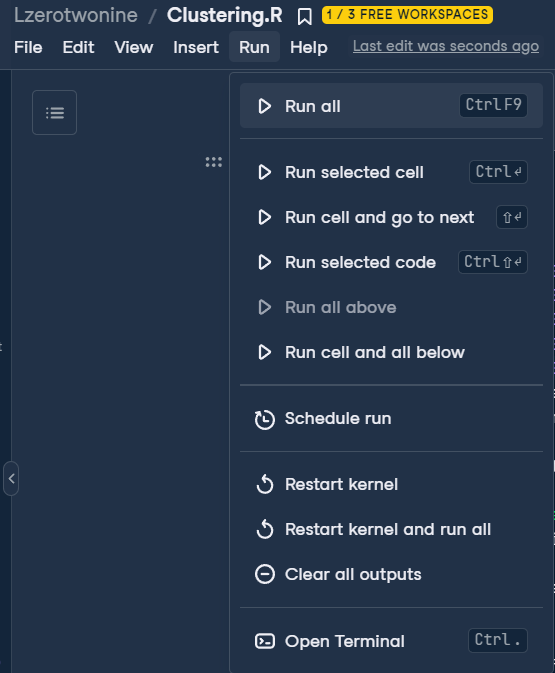
res$centroids$isCentroid=T

data\_plot=rbind(res$centroids,res$data)

ggplot(data\_plot,aes(x=X1,y=X2,color=as.factor(cluster),size=isCentroid,alpha=isCentroid))+geom\_point()

**5. Vẽ kết quả:** Cuối cùng, mã vẽ biểu đồ phân tán của kết quả phân cụm, với mỗi cụm được hiển thị bằng một màu khác nhau và trung tâm của mỗi cụm được đánh dấu.

- Sau đó là bấm chạy



3. Kết quả

