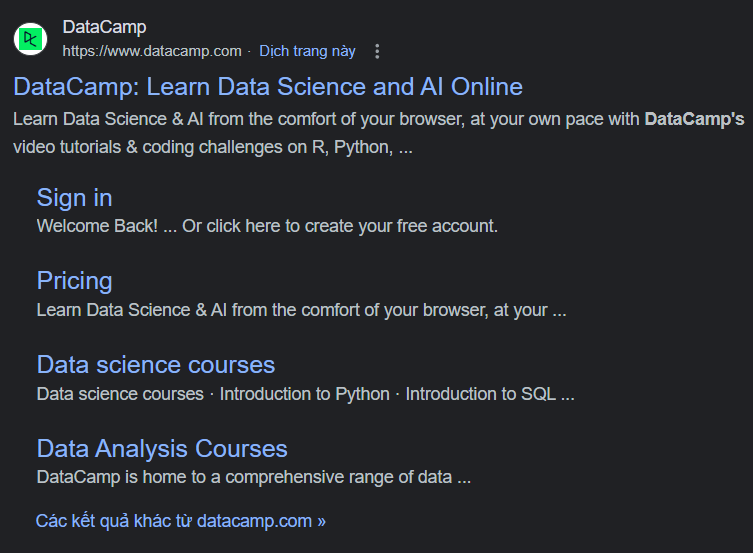
**Lab 9: Clustering in R**

**Họ và Tên**: Trần Nguyễn Gia Long

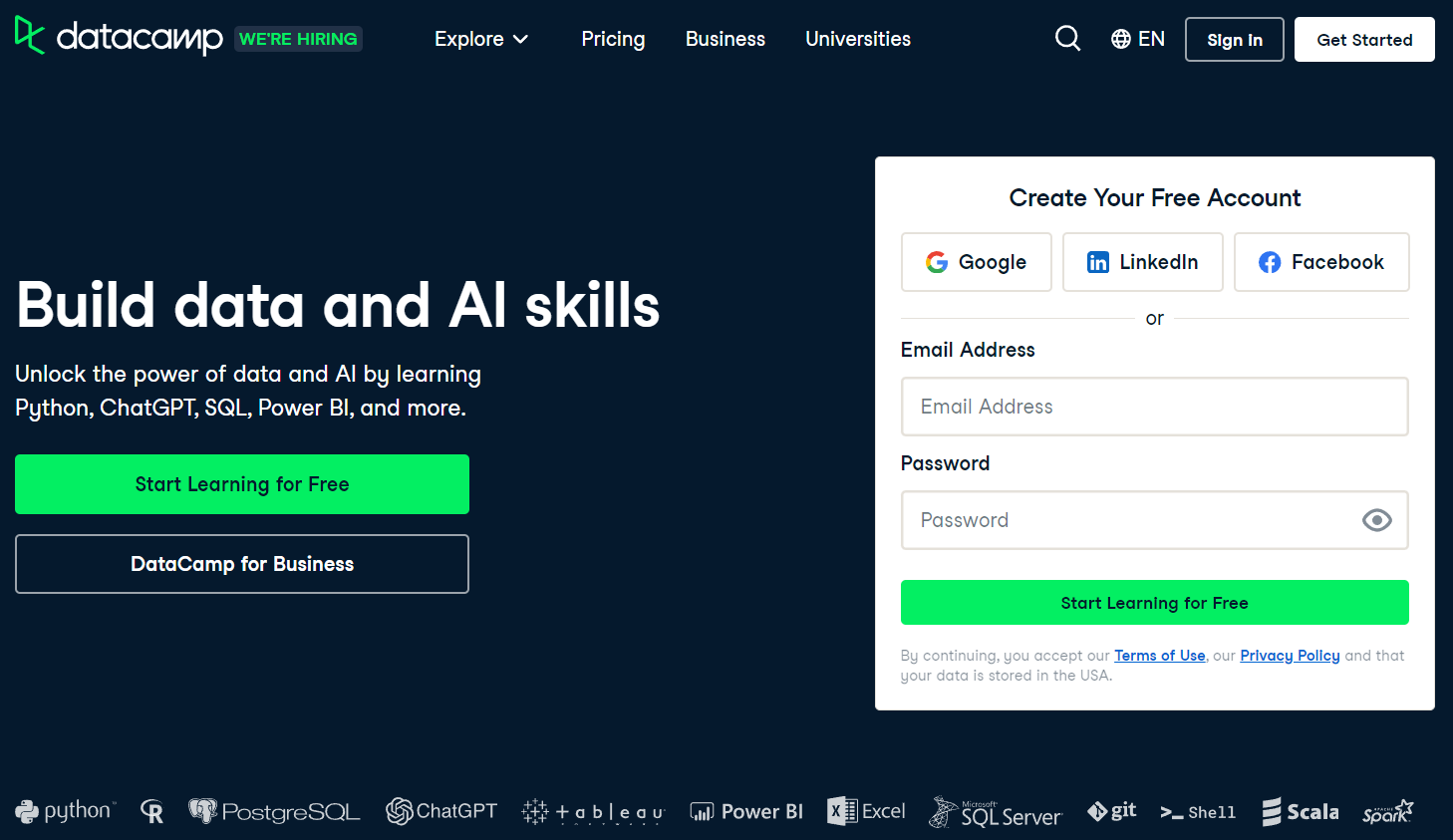
**MSSV**: 1050080059

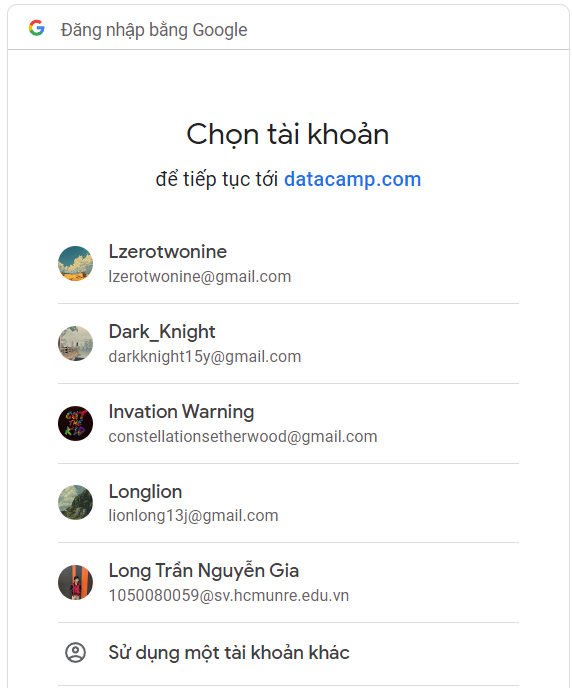
# 1. Cài đặt Clustering

- Sử dụng trang DataCamp để chạy Cosine.R.

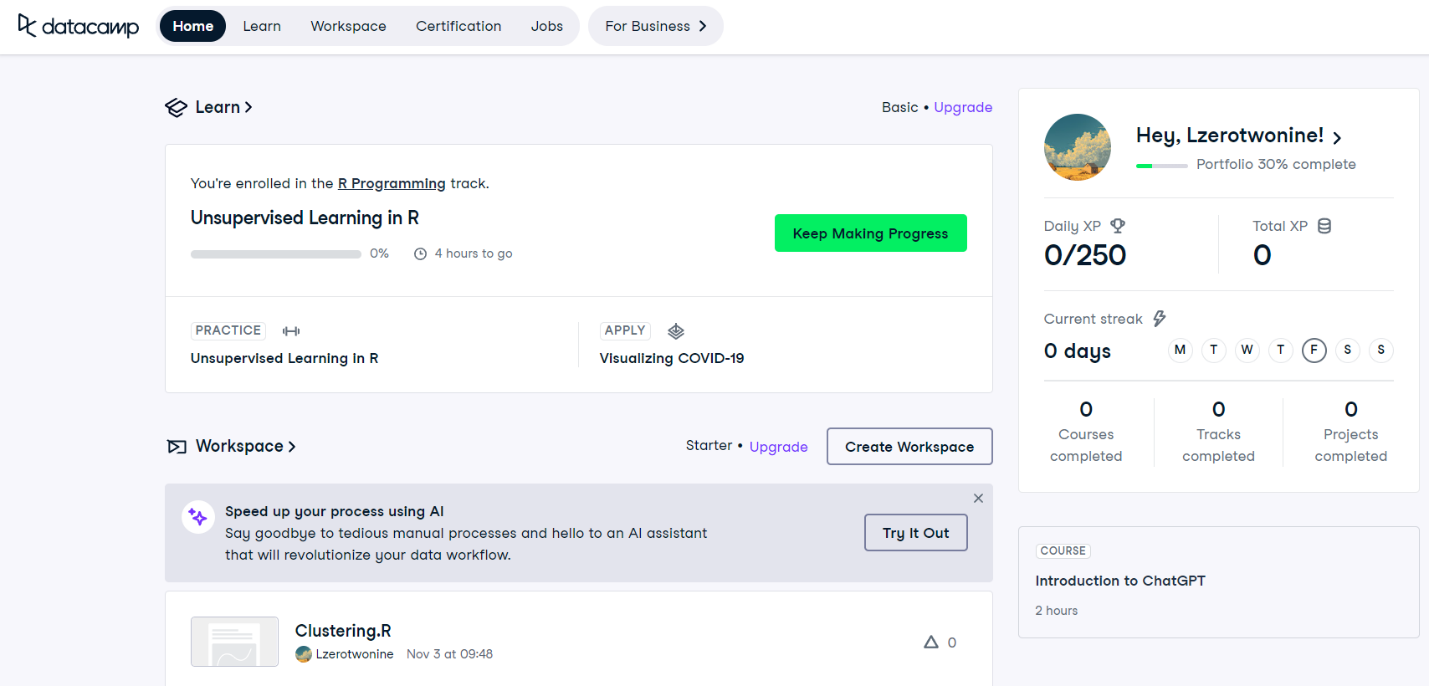


- Nhấp vào trang, tiến hành việc đăng nhập.

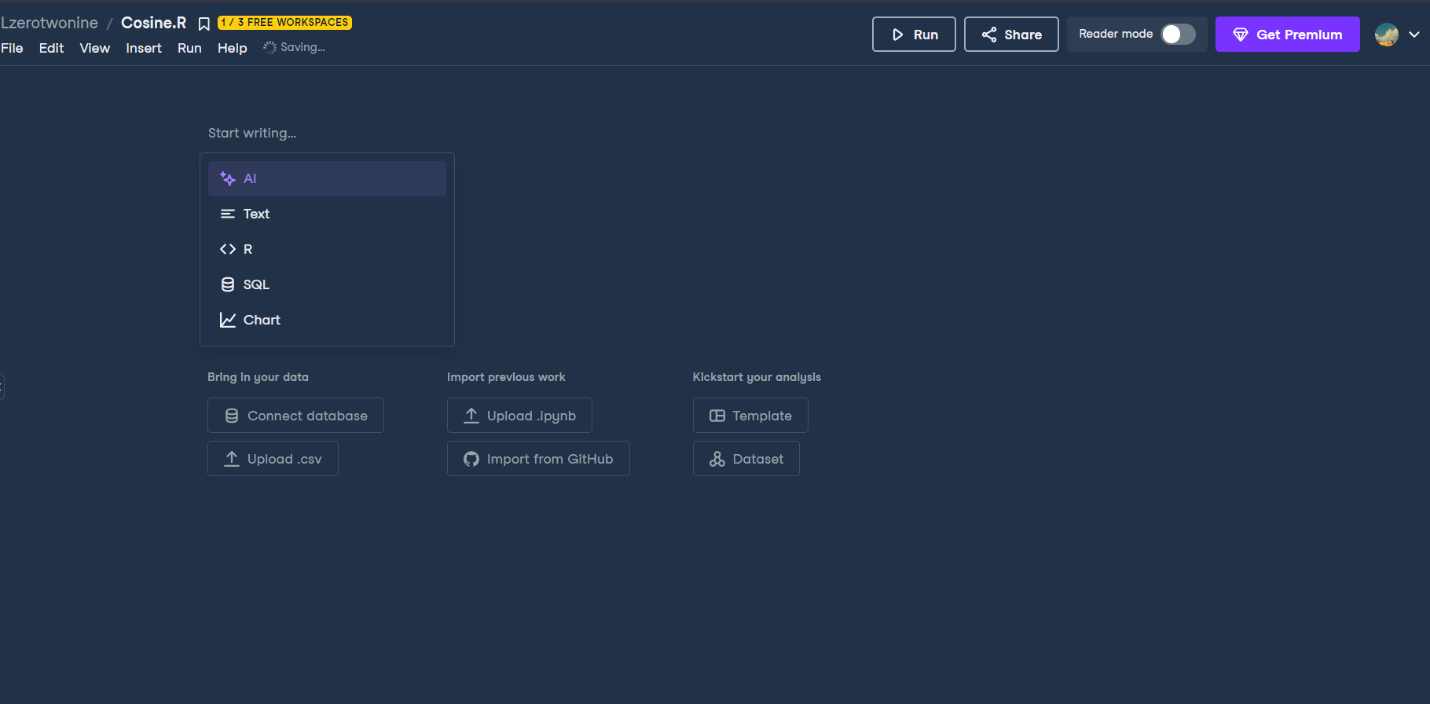


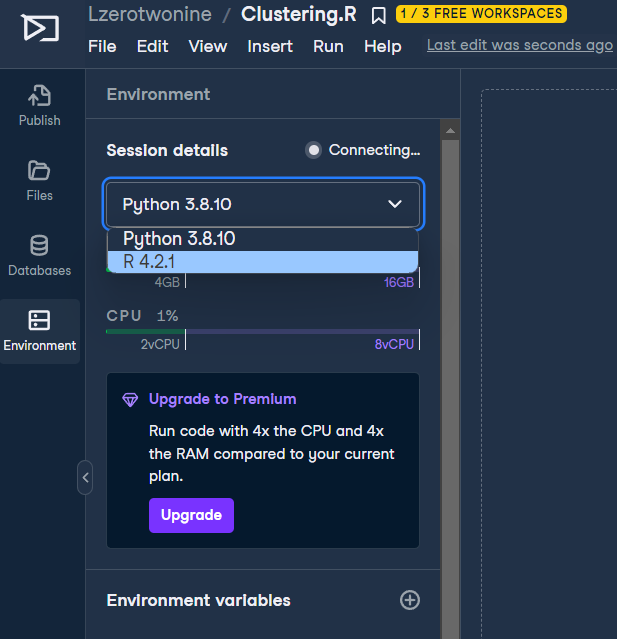


- Ở đây em chọn đăng nhập bằng tài khoản Google để vào nhanh chóng.



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

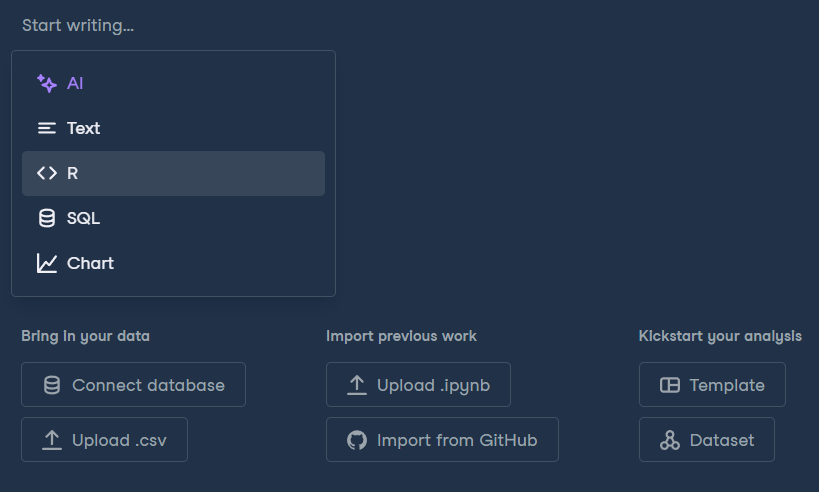
- Đây là giao diện của workspace.

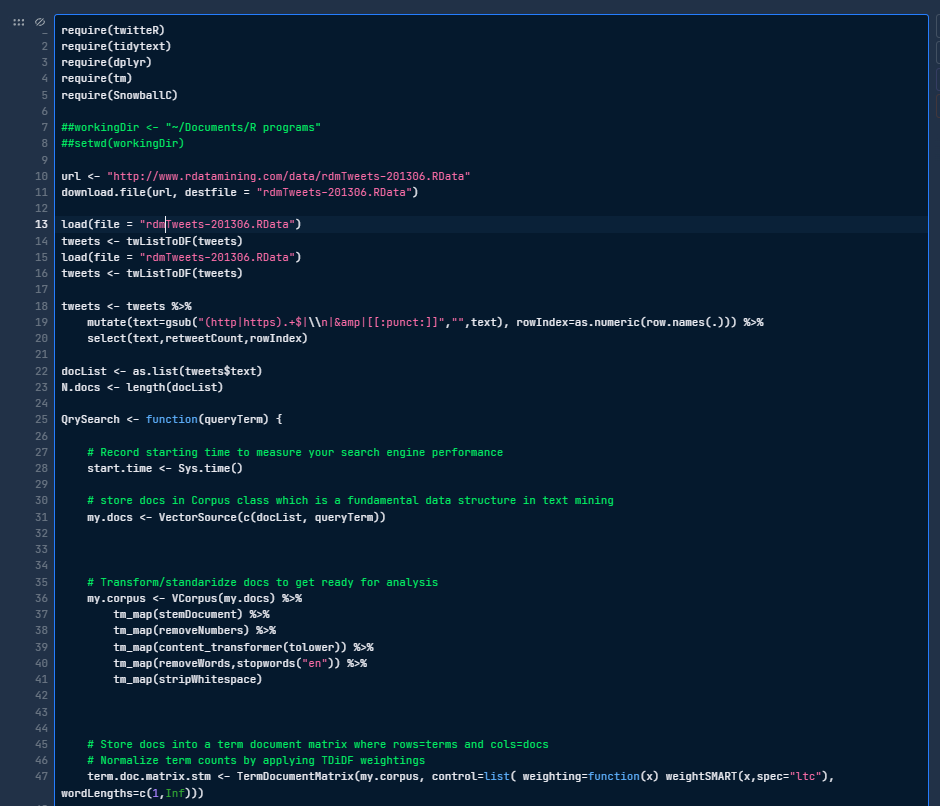


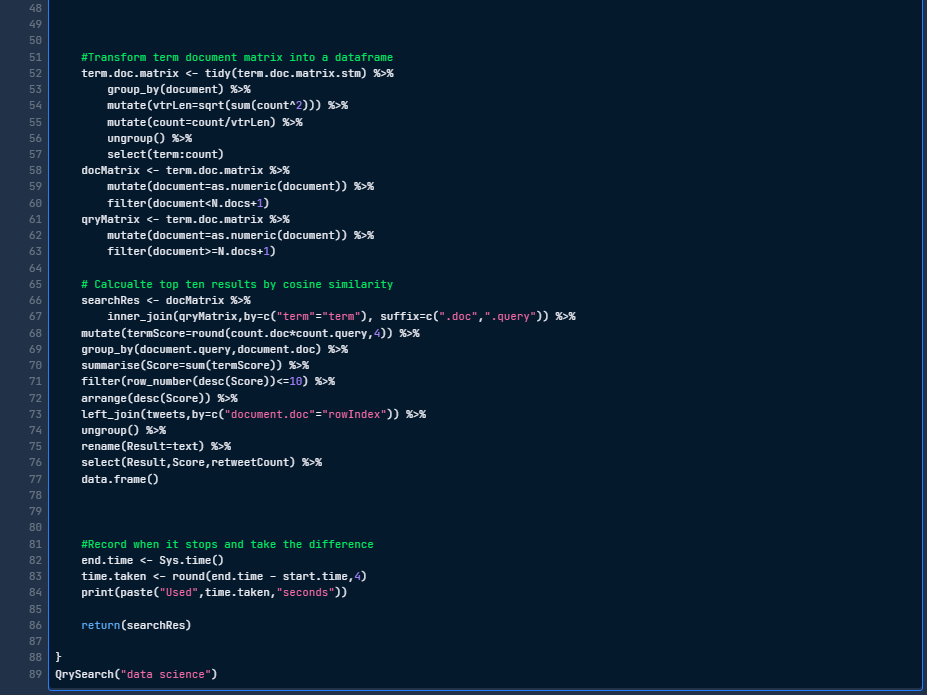
- Thay đổi ngôn ngữ lập trình thành R.

# 2. Chạy Cosine.

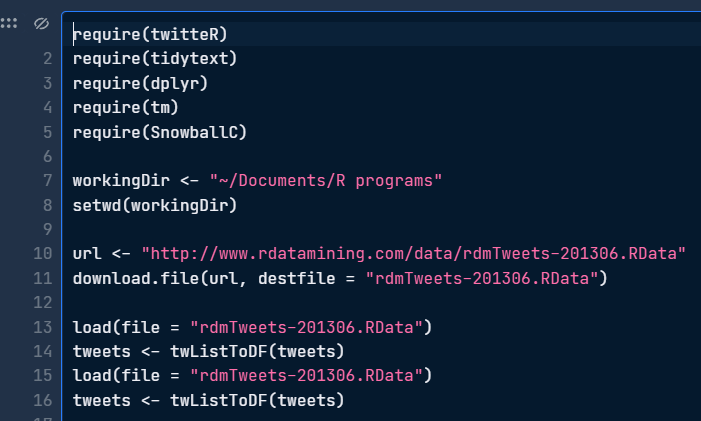
- Chọn viết code bằng R, sau đó dán đoạn code từ Lab 10 vào.



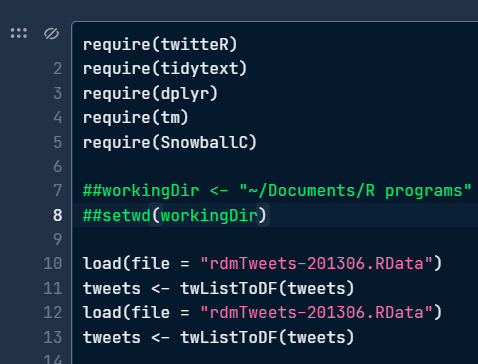




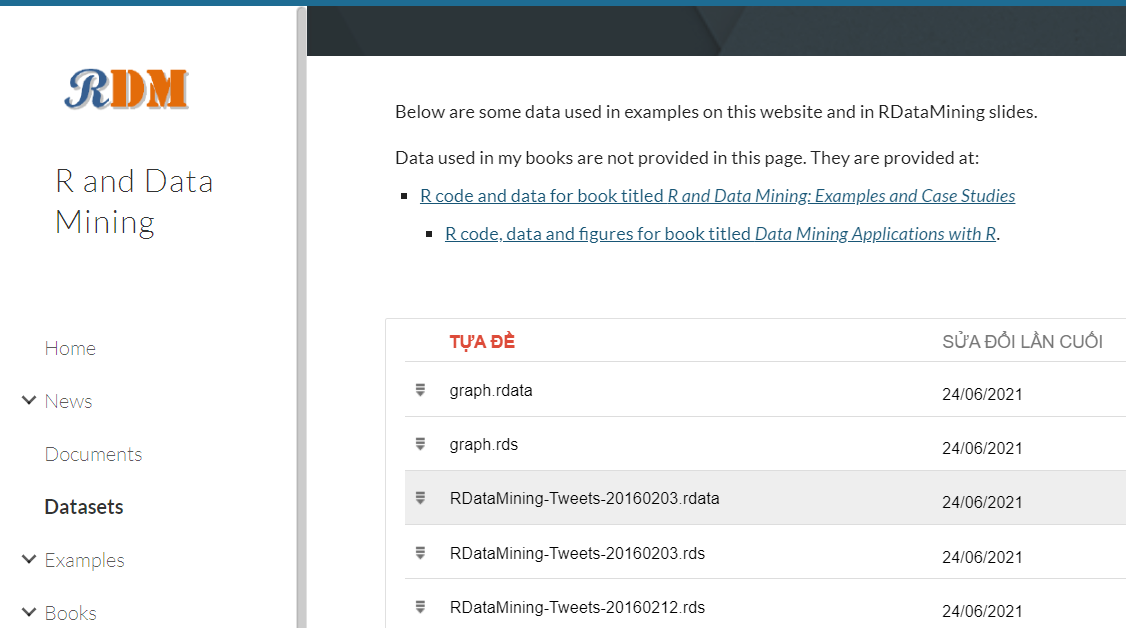
- Chỉnh sử đoạn code từ bài Lab.



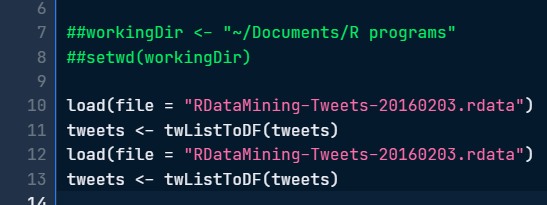
- Ở dòng 7, 8 ta sẽ chuyển thành chú thích, vì không cần phải thiết lập thư mục thực thi của R.



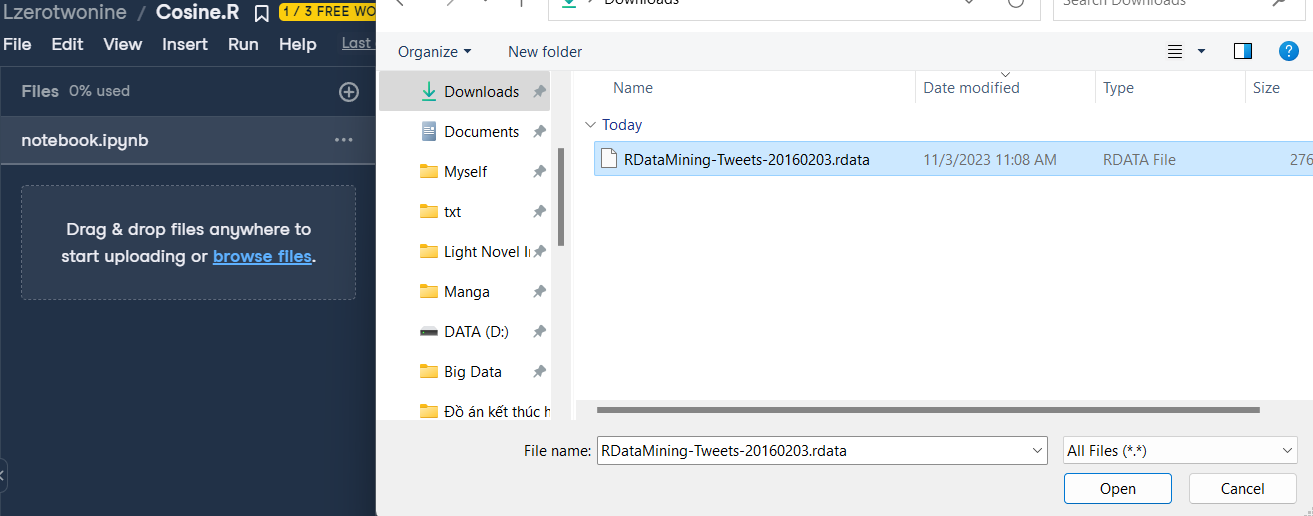
- Xoá dòng code 10, 11 vì đường link không thể mở được mà phải tải thủ công từ trang <http://www.rdatamining.com/datasets>



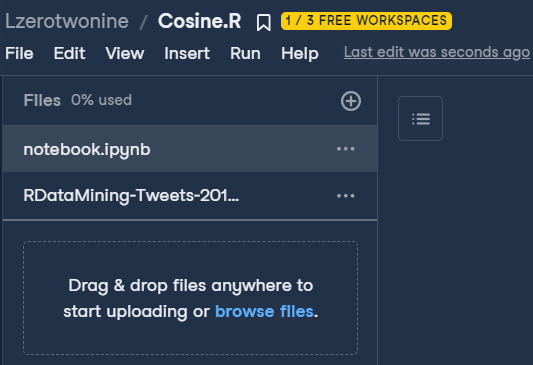
- Ở đây ta sẽ tải datasets tweets đầu tiên để lấy làm dữ liệu.



- Đổi tên file load thành tên file ta đã chọn.



- Ở góc trái của workspace, chọn Files để upload file.



require(twitteR)

require(tidytext)

require(dplyr)

require(tm)

require(SnowballC)

##workingDir <- "~/Documents/R programs"

##setwd(workingDir)

load(file = "RDataMining-Tweets-20160203.rdata")

tweets <- twListToDF(tweets)

load(file = "RDataMining-Tweets-20160203.rdata")

tweets <- twListToDF(tweets)

tweets <- tweets %>%

mutate(text=gsub("(http|https).+$|\\n|&amp|[[:punct:]]","",text), rowIndex=as.numeric(row.names(.))) %>%

select(text,retweetCount,rowIndex)

docList <- as.list(tweets$text)

N.docs <- length(docList)

QrySearch <- function(queryTerm) {

# Record starting time to measure your search engine performance

start.time <- Sys.time()

# store docs in Corpus class which is a fundamental data structure in text mining

my.docs <- VectorSource(c(docList, queryTerm))

# Transform/standaridze docs to get ready for analysis

my.corpus <- VCorpus(my.docs) %>%

tm\_map(stemDocument) %>%

tm\_map(removeNumbers) %>%

tm\_map(content\_transformer(tolower)) %>%

tm\_map(removeWords,stopwords("en")) %>%

tm\_map(stripWhitespace)

# Store docs into a term document matrix where rows=terms and cols=docs

# Normalize term counts by applying TDiDF weightings

term.doc.matrix.stm <- TermDocumentMatrix(my.corpus, control=list( weighting=function(x) weightSMART(x,spec="ltc"), wordLengths=c(1,Inf)))

#Transform term document matrix into a dataframe

term.doc.matrix <- tidy(term.doc.matrix.stm) %>%

group\_by(document) %>%

mutate(vtrLen=sqrt(sum(count^2))) %>%

mutate(count=count/vtrLen) %>%

ungroup() %>%

select(term:count)

docMatrix <- term.doc.matrix %>%

mutate(document=as.numeric(document)) %>%

filter(document<N.docs+1)

qryMatrix <- term.doc.matrix %>%

mutate(document=as.numeric(document)) %>%

filter(document>=N.docs+1)

# Calcualte top ten results by cosine similarity

searchRes <- docMatrix %>%

inner\_join(qryMatrix,by=c("term"="term"), suffix=c(".doc",".query")) %>%

mutate(termScore=round(count.doc\*count.query,4)) %>%

group\_by(document.query,document.doc) %>%

summarise(Score=sum(termScore)) %>%

filter(row\_number(desc(Score))<=10) %>%

arrange(desc(Score)) %>%

left\_join(tweets,by=c("document.doc"="rowIndex")) %>%

ungroup() %>%

rename(Result=text) %>%

select(Result,Score,retweetCount) %>%

data.frame()

#Record when it stops and take the difference

end.time <- Sys.time()

time.taken <- round(end.time - start.time,4)

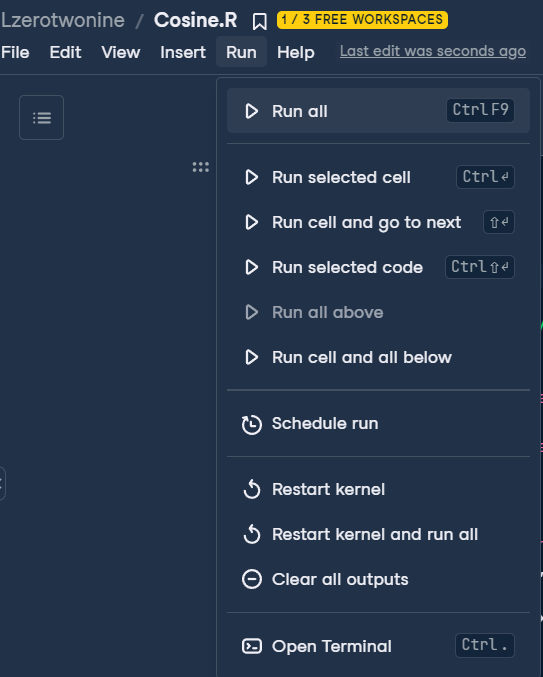
print(paste("Used",time.taken,"seconds"))

return(searchRes)

}

QrySearch("data science")

- Sau đó là bấm chạy



# 3. Kết quả

