## Pyspark code

## Sentiment Analysis Code

setwd("C:/Ankit/Y2S1/DATA7201 Data Analytics at Scale/Project/My Project")

df=read.csv("Data/4.ID\_Tweet by Date/July/15\_j.csv", header=TRUE, sep=",")

library("stringr")

clean\_tweets=data.frame()

for (i in 1:length(df$ID)){

clean\_tweet=gsub('#\\S\*', "", df$Tweet[i]) #Get Rif of Hashtag

clean\_tweet=gsub("&amp", "", clean\_tweet)

clean\_tweet=gsub("(RT|via)((?:\\b\\W\*@\\w+)+)", "", clean\_tweet)

clean\_tweet=gsub("@\\w+", "", clean\_tweet)

clean\_tweet=gsub("[[:punct:]]", "", clean\_tweet)

clean\_tweet=gsub("[[:digit:]]", "", clean\_tweet)

clean\_tweet=gsub("http\\w+", "", clean\_tweet)

clean\_tweet=gsub("[ \t]{3,}", "", clean\_tweet)

clean\_tweet=gsub("^\\s|+\\s+$", "", clean\_tweet)

clean\_tweet=gsub("rt", "", clean\_tweet)

clean\_tweet=gsub('€\\S\*', "", clean\_tweet)

clean\_tweet=gsub('â\\S\*', "", clean\_tweet)

clean\_tweet=gsub("<.\*>", "", clean\_tweet)

clean\_tweet=gsub("ðÿ", "", clean\_tweet)

clean\_tweet=str\_replace\_all(clean\_tweet," "," ")

clean\_tweets[i,1]=clean\_tweet

}

df['cleaned\_tweet']=clean\_tweets

#df2=df%>%select(ID, cleaned\_tweet)

#detach("package:tidytext", unload=TRUE)

#install.packages("syuzhet")

library("syuzhet")

df['nrc\_sentiment']=get\_sentiment(df$cleaned\_tweet, method="nrc")

df['bing\_sentiment']=get\_sentiment(df$cleaned\_tweet, method="bing")

df['afinn\_sentiment']=get\_sentiment(df$cleaned\_tweet, method="afinn")

emotions=get\_nrc\_sentiment(df$cleaned\_tweet)

df=cbind.data.frame(df, emotions)

write.csv(df, file = "Data/5.Sentiment + Hashtag/July/15\_j.csv",row.names=FALSE)

## Tweet Trend Code

require(vcd)

require(MASS)

# data generation

ex <- rexp(10000, rate = 1.85) # generate some exponential distribution

control <- abs(rnorm(10000)) # generate some other distribution

data <- c(29235,14756,2720,1756,1550,1530,1075,1010,1239,1255,1267,1151,991,416,607,325,378,290,366,383,240,210,236,366,555,455,282,323,514)

# estimate the parameters

fit1 <- fitdistr(ex, "exponential")

fit2 <- fitdistr(control, "exponential")

fit3 <- fitdistr(log(vector2), "exponential")

# goodness of fit test

ks.test(ex, "pexp", fit1$estimate) # p-value > 0.05 -> distribution not refused

ks.test(control, "pexp", fit2$estimate) # significant p-value -> distribution refused

ks.test(log(vector2), "pexp", fit3$estimate) # significant p-value -> distribution refused

# plot a graph

hist(ex, freq = FALSE, breaks = 100, xlim = c(0, quantile(ex, 0.99)))

curve(dexp(x, rate = fit1$estimate), from = 0, col = "red", add = TRUE)

plot(vector2)