Project1

Rui Wang(rw2795)

2/4/2020

Step 0 - Install and load all the required libraries

```
library(stringr)
library("syuzhet")
library(tidyr)
library(dplyr)
library(purrr)
library(wordcloud)
library(memoise)
library(tidyverse)
library(tidytext)
library(plotly)
library(DT)
library(tm)
library(data.table)
library(scales)
library(wordcloud2)
library(gridExtra)
library(ngram)
library(shiny)
```

This notebook was prepared with the following environmental settings.

```
print(R.version)
```

```
##                _
## platform       x86_64-apple-darwin15.6.0
## arch           x86_64
## os             darwin15.6.0
## system         x86_64, darwin15.6.0
## status
## major          3
## minor          6.1
## year           2019
## month          07
## day            05
## svn rev        76782
## language       R
## version.string R version 3.6.1 (2019-07-05)
## nickname       Action of the Toes
```

Step 1 - Load the processed data

```
# load processed lyrics data
# processed lyrics data was generated by "Text_Processing.Rmd"
load('../output/processed_lyrics.RData')
```

Step 2 - Data analysis — Word Frequency

```
dt_rock = dt_lyrics%>%
  dplyr::filter(genre == "Rock")
lyrics_rock = dt_rock %>% pull(stemmedwords) %>% paste(collapse = " ") %>%
  strsplit(" ") %>% unlist()
words_rock = tibble(words = lyrics_rock) %>%
  group_by(words) %>% count() %>%
  arrange(desc(n))

dt_hiphop = dt_lyrics %>%
  dplyr::filter(genre == "Hip-Hop")
lyrics_hiphop = dt_hiphop %>% pull(stemmedwords) %>% paste(collapse = " ") %>%
  strsplit(" ") %>% unlist()
words_hiphop = tibble(words = lyrics_hiphop) %>%
  group_by(words) %>% count() %>%
  arrange(desc(n))
```

Step 3 - Inspect an Overall Wordcloud

```
wordcloud(words_rock$words, words_rock$n,
          scale=c(5,0.5),
          max.words=100,
          min.freq=1,
          random.order=FALSE,
          rot.per=0.3,
          use.r.layout=T,
          random.color=FALSE,
          colors=brewer.pal(9,"Blues"))
```

```
wordcloud(words_hiphop$words, words_hiphop$n,
          scale=c(5,0.5),
          max.words=100,
          min.freq=1,
          random.order=FALSE,
          rot.per=0.3,
          use.r.layout=T,
          random.color=FALSE,
          colors=brewer.pal(9,"Blues"))
```

We can see from the plots that "LOVE" is the most commonly used word in these two genres and there are many rhythmic words in hiphop lyrics, such as em, yo, ya. In general, the words used for Rock n roll is healthier and more gentle than Hip-Hop.

Step 4 - Interactive Word Cloud with Time Variation

```
### Preparations for visualization
time_list1 <- c("1970s", "1980s", "1990s", "2000s", "2010s")
time_list2 <- c("1980s", "1990s", "2000s", "2010s")
corpus <- VCorpus(VectorSource(dt_lyrics$stemmedwords))
word_tibble <- tidy(corpus) %>%
  select(text) %>%
  mutate(id = row_number()) %>%
  unnest_tokens(word, text)

### Specify the user interface for the R Shiny app
# Define UI for app that draws a histogram ----
ui <- navbarPage(strong("Lyrics Analysis with Time Variation"),
```

```
                tabPanel("Plots",
                         titlePanel("Rock vs. Hip-Hop"),
                         # Sidebar layout with input and output definitions ----
                         sidebarLayout(
                           # Sidebar panel for inputs ----
                           sidebarPanel(
                             sliderInput(inputId = "nwords1",
                                         label = "Number of terms in the first word cloud:",
                                         min = 5, max = 100, value = 50),
                             selectInput('decade1', 'Selected decade for the first plot:',
                                         time_list1, selected='1980s')

                           ),
                           # Main panel for displaying outputs ----
                           mainPanel(
                             wordcloud2Output(outputId = "WC1", height = "300")
                           )
                         ),
                         hr(),
                         sidebarLayout(
                           # Sidebar panel for inputs ----
                           sidebarPanel(
                             sliderInput(inputId = "nwords2",
                                         label = "Number of terms in the second word cloud:",
                                         min = 5, max = 100, value = 50),
                             selectInput('decade2', 'Selected decade for the second plot:',
                                         time_list2, selected='1980s')
                           ),
                           # Main panel for displaying outputs ----
                           mainPanel(
                             wordcloud2Output(outputId = "WC2", height = "300")
                           )
                         )
                ),
                tabPanel("Data",
                         DT::dataTableOutput("table"))
)

# Define server logic required for ui ----
server <- function(input, output) {
  output$WC1 <- renderWordcloud2({
    year1 = input$decade1 %>% str_sub(1, -2) %>% as.numeric()
    count(filter(word_tibble, id %in% which((dt_lyrics$genre == "Rock") & (dt_lyrics$year >= year1) & (
      slice(1:input$nwords1) %>%
      wordcloud2(size=0.6, rotateRatio=0.2)
  })
  output$WC2 <- renderWordcloud2({
    year2 = input$decade2 %>% str_sub(1, -2) %>% as.numeric()
    count(filter(word_tibble, id %in% which((dt_lyrics$genre == "Hip-Hop") & (dt_lyrics$year >= year2) &
      slice(1:input$nwords2) %>%
      wordcloud2(size=0.6, rotateRatio=0.2)
  })
  output$table <- DT::renderDataTable({
```

```
    DT::datatable(dt_lyrics)
  })
}

### Run the R Shiny app
shinyApp(ui, server)
```

Shiny applications not supported in static R Markdown documents

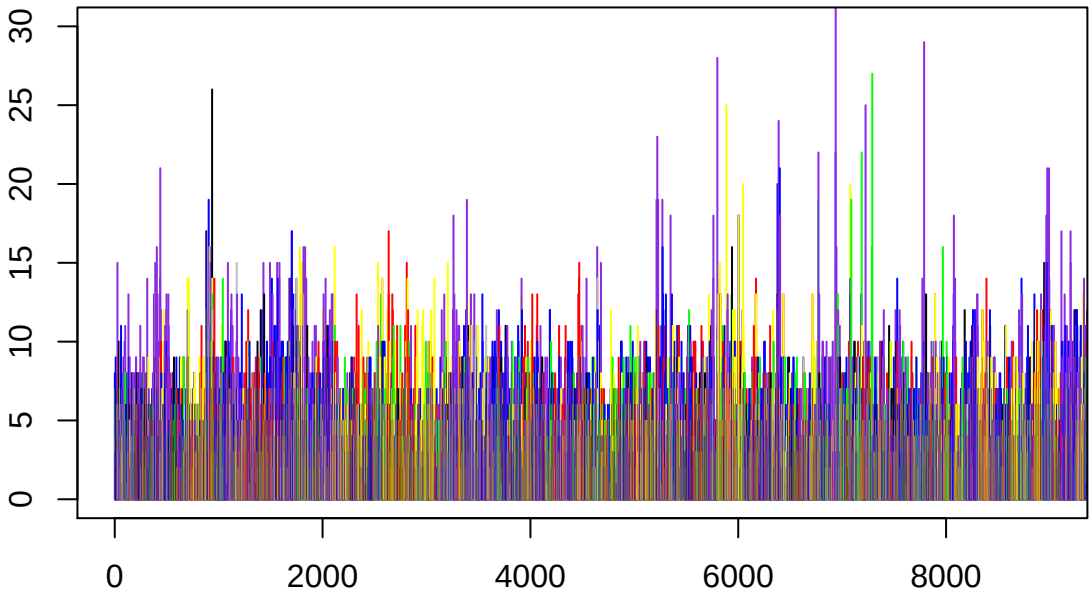
With the interactive word cloud, we found frequently used vocabulary don't change too much for Rock music. It always centres around love, time, life, etc. However, topics around Hip-Hop changed a lot from house, car, people to bitch, shit, money.

Step 5 - Data Analysis — Sentiment Analsis

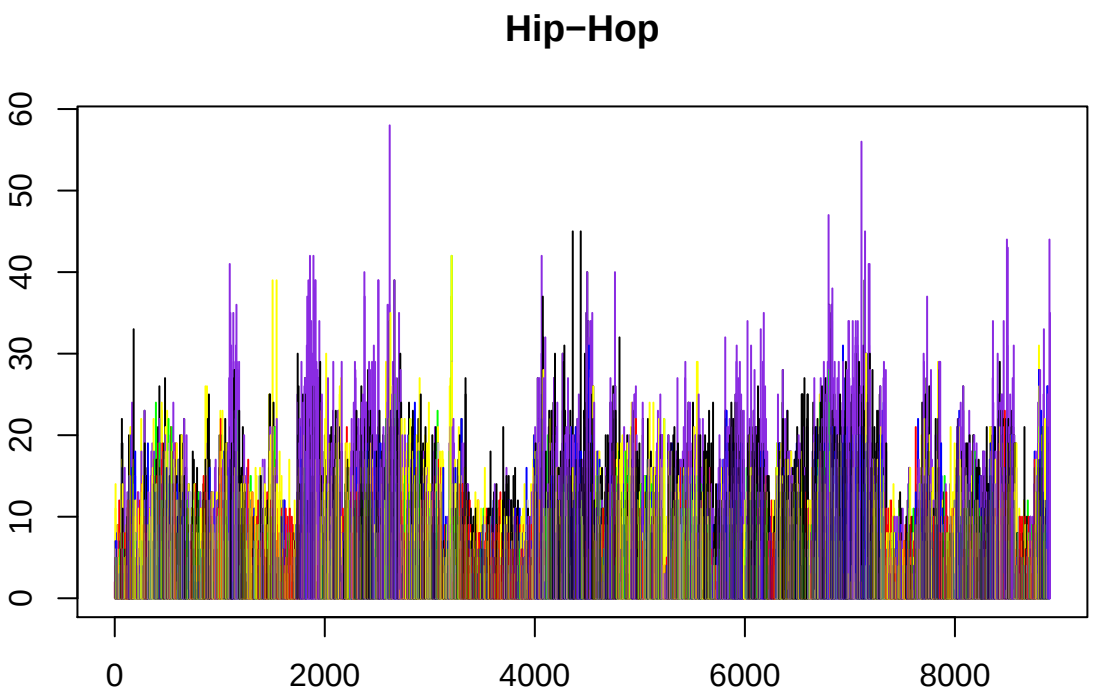
```
###using get_nrc_sentiment to get emotion data
memoise(get_nrc_sentiment)
dt_rock_emo = dt_rock %>%
  mutate(emotions=map(stemmedwords,get_nrc_sentiment)) %>%
  unnest(emotions)
save(dt_rock_emo, file="../output/dt_rock_emotions.RData")

dt_hiphop_emo = dt_hiphop %>%
  mutate(emotions=map(stemmedwords,get_nrc_sentiment)) %>%
  unnest(emotions)
save(dt_hiphop_emo, file="../output/dt_hiphop_emotions.RData")
```

```
load("../output/dt_rock_emotions.RData")
load("../output/dt_hiphop_emotions.RData")

dt_rock_emo_plot = dt_rock_emo %>%
  mutate(x = seq_along(id)) %>%
  select(x, anger:trust) %>%
  pivot_longer(anger:trust) %>%
  group_by(x) %>%
  top_n(1, value) %>%
  mutate(color = map(name, ~switch (.x,
                                    anticipation = "green",
                                    joy = "red",
                                    surprise = "orange",
                                    trust = "yellow",
                                    anger = "black",
                                    disgust = "grey",
                                    fear = "blueviolet",
                                    sadness = "blue"))
         ) %>%
  unnest(color)

plot(dt_rock_emo_plot$x, dt_rock_emo_plot$value,
     col = dt_rock_emo_plot$color, type="h", main = "Rock", xlab = "", ylab = "", xlim = c(1, 9000), yli
```

Rock




```
dt_hiphop_emo_plot = dt_hiphop_emo %>%
  mutate(x = seq_along(id)) %>%
  select(x, anger:trust) %>%
  pivot_longer(anger:trust) %>%
  group_by(x) %>%
  top_n(1, value) %>%
  mutate(color = map(name, ~switch (.x,
                                    anticipation = "green",
                                    joy = "red",
                                    surprise = "orange",
                                    trust = "yellow",
                                    anger = "black",
                                    disgust = "grey",
                                    fear = "blueviolet",
                                    sadness = "blue"))
  ) %>%
  unnest(color)

plot(dt_hiphop_emo_plot$x, dt_hiphop_emo_plot$value,
     col = dt_hiphop_emo_plot$color, type="h", main = "Hip-Hop", xlab = "", ylab = "")
```



As we can see from above, the emotion plot for Rock genre is more colorful but Hip-Hop's is darker, which means that Hip-Hop conveys more negative emotions from their lyrics.

Sammary

By analyzing the lyrics for Rock and Hip-Hop, we could get the following results.

- Lyrics for Rock music conveys healthier information which talks about life with positive trustful words.
- Lyrics for Hip-Hop music conveys kind of unhealthier information with negtive fearful words.