R Notebook

Load all the required libraries

```
library(tidytext)
library(textdata)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(DT)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(scales)
library(wordcloud2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(ngram)
library(tidyr)
library(ggplot2)
```

Load the processed lyrics data and split it by genre into different datasets

```
load('../output/processed_lyrics.RData')
Country<-dt_lyrics[which(dt_lyrics$genre=='Country'),]
Electronic<-dt_lyrics[which(dt_lyrics$genre=='Electronic'),]
HipHop<-dt_lyrics[which(dt_lyrics$genre=='Hip-Hop'),]
Folk<-dt_lyrics[which(dt_lyrics$genre=='Folk'),]
Jazz<-dt_lyrics[which(dt_lyrics$genre=='Jazz'),]
Indie<-dt_lyrics[which(dt_lyrics$genre=='Indie'),]
RandB<-dt_lyrics[which(dt_lyrics$genre=='R&B'),]
Rock<-dt_lyrics[which(dt_lyrics$genre=='Rock'),]
Metal<-dt_lyrics[which(dt_lyrics$genre=='Metal'),]
Pop<-dt_lyrics[which(dt_lyrics$genre=='Pop'),]
Other<-dt_lyrics[which(dt_lyrics$genre=='Not Available'),]
```

```
country_corpus <- VCorpus(VectorSource(Country$stemmedwords))
country_word_tibble <- tidy(country_corpus) %>%
  select(text) %>%
  mutate(id = row_number()) %>%
  unnest_tokens(word, text)

nrc_joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

nrc_sadness<-get_sentiments("nrc") %>%
  filter(sentiment == "sadness")

country_word_tibble %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 340 x 2
##    word        n
##    <chr>   <int>
##  1 love    13308
##  2 baby     3425
##  3 sweet    1465
```

```
##  4 true     1331
##  5 sing     1245
##  6 kiss     1221
##  7 friend   1214
##  8 god      1203
##  9 feeling  1138
## 10 sun      1122
## # ... with 330 more rows
```

```
country_word_tibble %>%
  inner_join(nrc_sadness) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 440 x 2
##    word        n
##    <chr>   <int>
##  1 ill      4770
##  2 leave    1897
##  3 blue     1878
##  4 cry      1747
##  5 fall     1367
##  6 sing     1245
##  7 die      1179
##  8 feeling  1138
##  9 lonely   1016
## 10 lie      1011
## # ... with 430 more rows
```

```
corpus <- VCorpus(VectorSource(dt_lyrics$stemmedwords))
dt_lyrics$genre1<-factor(dt_lyrics$genre,levels=c("Folk", "R&B", "Electronic", "Jazz", "Indie", "Country
word_tibble <-  data.frame(genre=dt_lyrics$genre1,tidy(corpus))%>%
  group_by(genre) %>%
  mutate(id = row_number()) %>%
  ungroup()%>%
  unnest_tokens(word, text)
lyrics_sentiment <- word_tibble %>%
  inner_join(get_sentiments("bing")) %>%
  count(genre, index = id %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

```
bing_pos<-get_sentiments("bing") %>%
  filter(sentiment=="positive")
bing_neg<-get_sentiments("bing") %>%
  filter(sentiment=="negative")
metal_corpus <- VCorpus(VectorSource(Metal$stemmedwords))
metal_word_tibble <- tidy(metal_corpus) %>%
  select(text) %>%
  mutate(id = row_number()) %>%
  unnest_tokens(word, text)
metal_word_tibble %>%
  inner_join(bing_pos) %>%
```

```
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

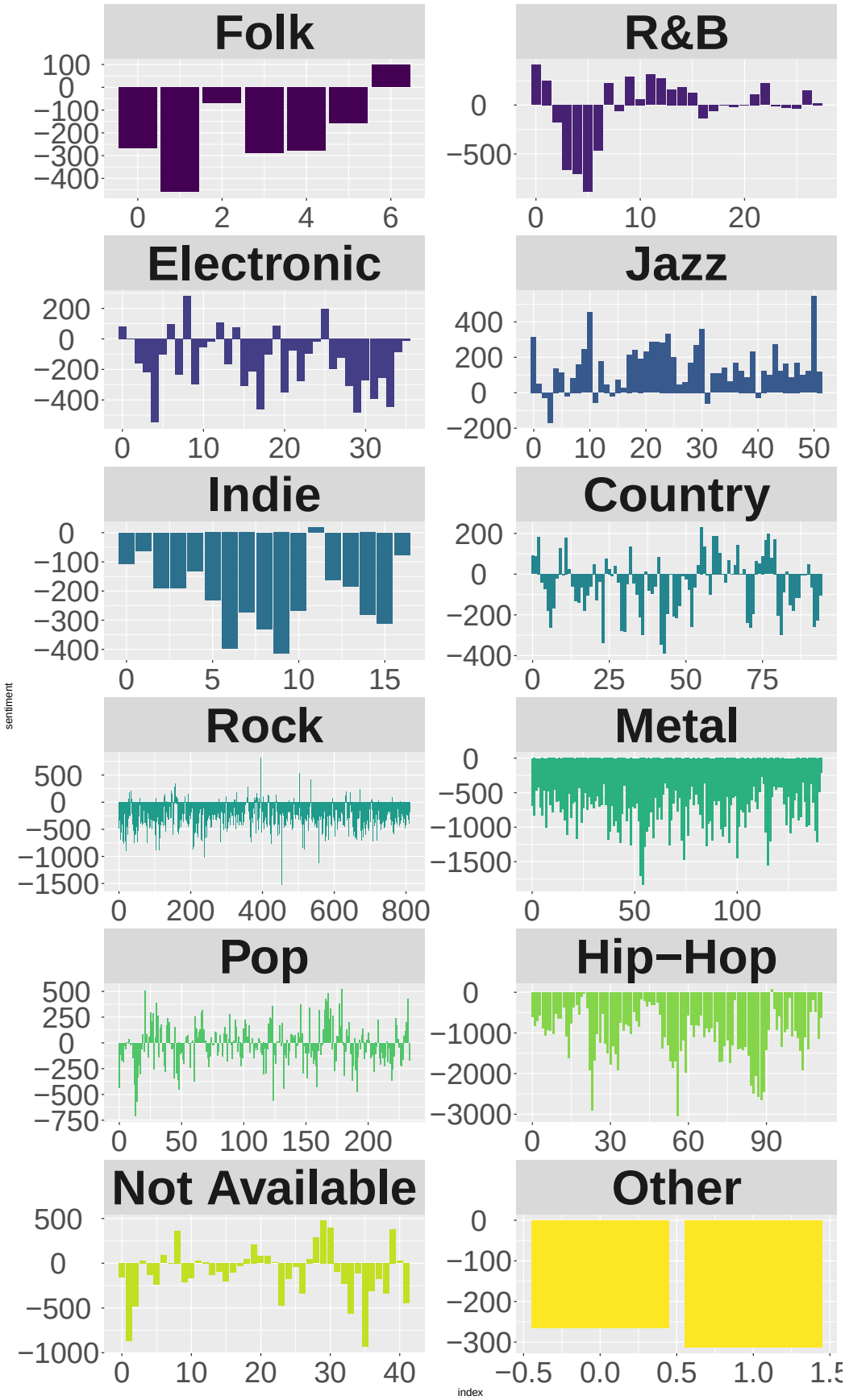
```
## # A tibble: 765 x 2
##    word          n
##    <chr>     <int>
##  1 love       4870
##  2 free       1999
##  3 lead       1131
##  4 angel      1114
##  5 heaven     1083
##  6 shine       883
##  7 strong      860
##  8 faith       854
##  9 smile       809
## 10 beautiful   798
## # ... with 755 more rows
```

```
metal_word_tibble %>%
  inner_join(bing_neg) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 1,918 x 2
##    word      n
##    <chr> <int>
##  1 die    5253
##  2 death  3834
##  3 pain   3754
##  4 burn   3731
##  5 dark   3726
##  6 lie    3598
##  7 dead   3574
##  8 fall   3392
##  9 fear   2826
## 10 kill   2644
## # ... with 1,908 more rows
```

```
ggplot(lyrics_sentiment, aes(index, sentiment, fill = genre)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~genre, ncol = 2, scales = "free")+
  theme(axis.text.x = element_text(size = 30, vjust = 0.5, hjust = 0.5),axis.text.y = element_text(size
```


Folk
R&B
Electronic
Jazz
Indie
Country
Rock
Metal
Pop
Hip−Hop
Not Available
Other
sentiment
index