

# R Notebook

```
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(ggplot2)
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##   annotate
```

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
music <- read.csv("~/Software/musics/output.csv")
count(music)
```

```
##           n
## 1 21428
```

```
head(music)
```

```
##   id          title          album artist_name
## 1  1      Good Ol' 39 Days #3 (Deluxe Version) The Script
## 2  2  A Great Big Sled (feat. Toni Halliday) (RED) Christmas EP The Killers
## 3  3      Brutal Love          ;TRÉ!  Green Day
## 4  4      Love Me Do          1 The Beatles
## 5  5 Low Low (ft Tee Cee & Nipsey Hussle & RJ)      10 Summers DJ Mustard
## 6  6      Chandelier  1000 Forms Of Fear          Sia
##           genre year length bitrate lyrics album_id artist_id
## 1 Alternative Rock 2012 264045 320000    NA      54      26
## 2      Holiday 2011 259030 320000    NA     375     286
## 3      Rock 2012 296356 320000    NA    1000     591
```

```
## 4          Pop 2000 141349 192000    NA      617      458
## 5      Hip-Hop 2014 156290 320000    NA      204      183
## 6          Pop 2014 216120 320000    NA      610       74
```

```
cbind(lapply(lapply(music, is.na), sum))
```

```
##           [,1]
## id          0
## title       0
## album       0
## artist_name 0
## genre       0
## year        0
## length      0
## bitrate     0
## lyrics     21428
## album_id    0
## artist_id   0
```

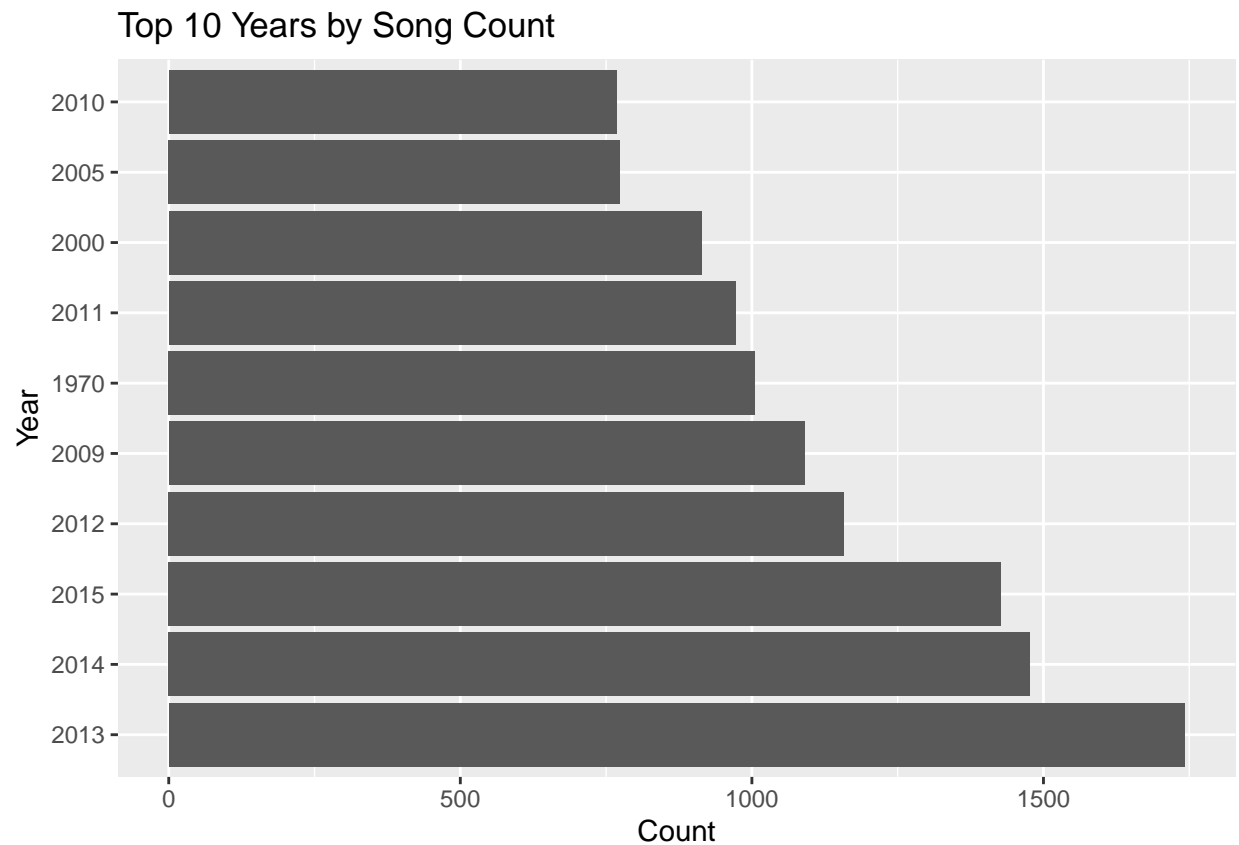
```
music <- subset(music, select=-c(lyrics, id, album_id, artist_id))
head(music)
```

```
##           title          album artist_name
## 1      Good Ol' 39 Days #3 (Deluxe Version) The Script
## 2  A Great Big Sled (feat. Toni Halliday) (RED) Christmas EP The Killers
## 3      Brutal Love          ;TRÉ!   Green Day
## 4      Love Me Do          1 The Beatles
## 5 Low Low (ft Tee Cee & Nipsey Hussle & RJ)      10 Summers DJ Mustard
## 6      Chandelier  1000 Forms Of Fear          Sia
##           genre year length bitrate
## 1 Alternative Rock 2012 264045 320000
## 2      Holiday 2011 259030 320000
## 3      Rock 2012 296356 320000
## 4      Pop 2000 141349 192000
## 5      Hip-Hop 2014 156290 320000
## 6      Pop 2014 216120 320000
```

```
top_10 <- music %>% count(year) %>% arrange(desc(n)) %>% top_n(10)
```

```
## Selecting by n
```

```
filtered <- music %>% filter(year %in% top_10$year) %>% mutate(Years = factor(year, levels=top_10$year))
ggplot(filtered, aes(x=Years)) +
  geom_bar() + coord_flip() +
  labs(x = "Year", y = "Count", title = "Top 10 Years by Song Count")
```

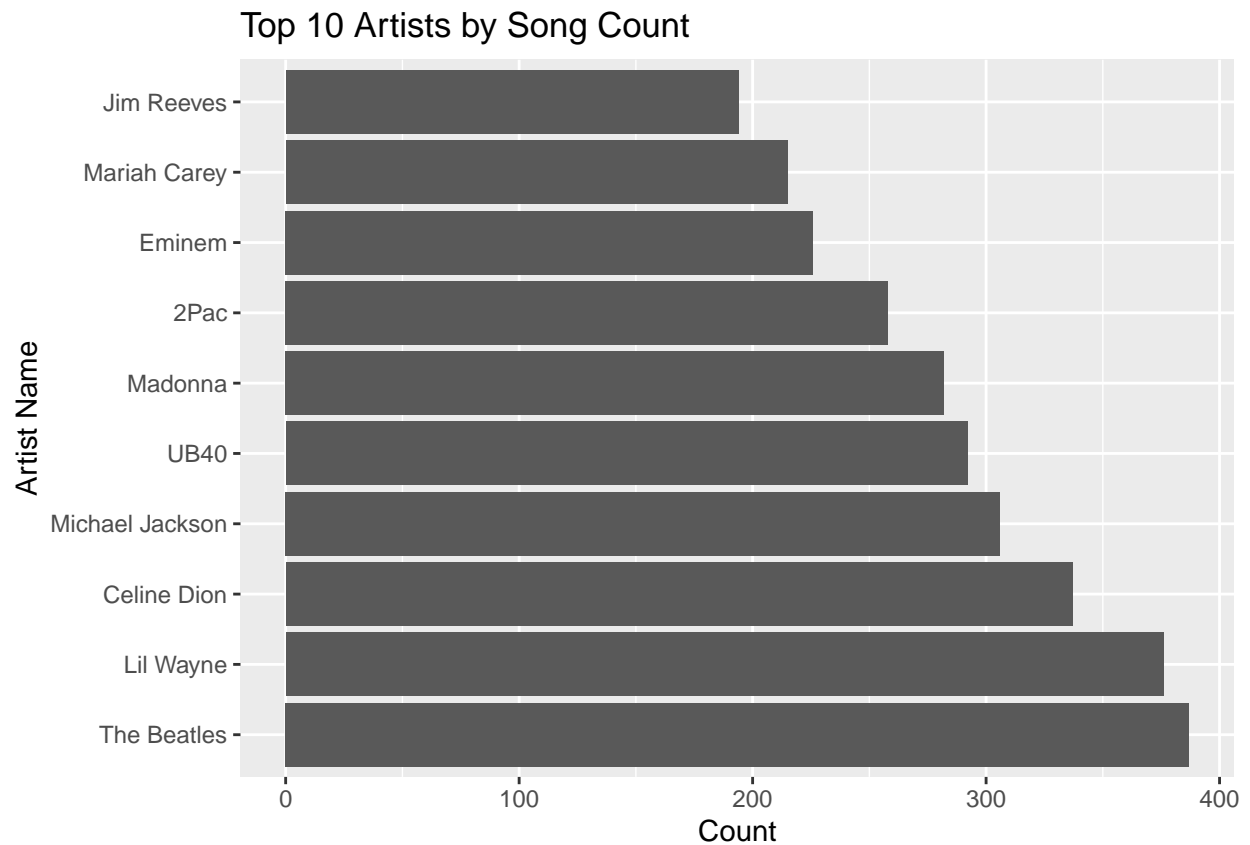


```
top_10_art <- music %>% count(artist_name) %>% arrange(desc(n)) %>% top_n(10)
```

```
## Selecting by n
```

```
filtered_art <- music %>% filter(artist_name %in% top_10_art$artist_name) %>% mutate(Artists = factor(a
```

```
ggplot(filtered_art, aes(x=Artists), fill=Artists) +  
  geom_bar() + coord_flip() +  
  labs(x = "Artist Name", y = "Count", title = "Top 10 Artists by Song Count")
```

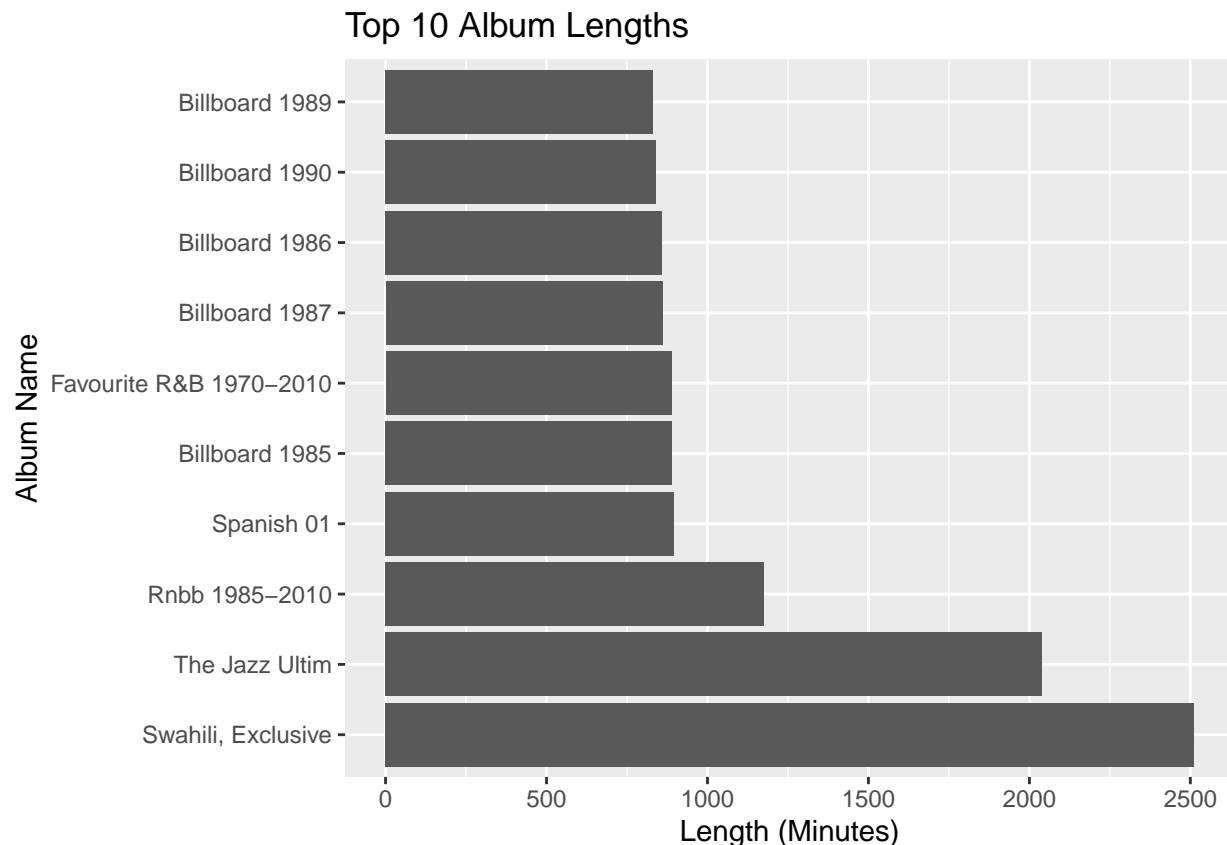


```
album_lengths <- music %>% group_by(album) %>% summarize(length_minutes = sum(length)/1000/60) %>% arrange(desc(length_minutes))
```

```
## Selecting by length_minutes
```

```
album_lengths$album <- factor(album_lengths$album, levels = album_lengths$album)
```

```
ggplot(album_lengths, aes(x = album, y = length_minutes)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(x = "Album Name", y = "Length (Minutes)", title = "Top 10 Album Lengths")
```



```
corpus <- Corpus(VectorSource(music$album))
corpus <- tm_map(corpus, content_transformer(tolower))

## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents

corpus <- tm_map(corpus, removePunctuation)

## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents

corpus <- tm_map(corpus, removeNumbers)

## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents

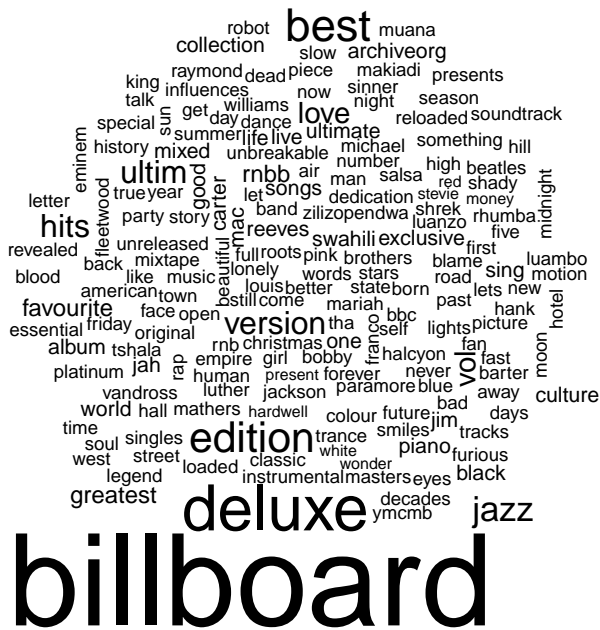
corpus <- tm_map(corpus, removeWords, stopwords("english"))

## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")):
## transformation drops documents

corpus <- tm_map(corpus, stripWhitespace)

## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents

term_doc_matrix <- TermDocumentMatrix(corpus)
word_frequency <- rowSums(as.matrix(term_doc_matrix))
wordcloud(words = names(word_frequency), freq = word_frequency, min.freq = 40)
```



```
corpus <- Corpus(VectorSource(music$title))
corpus <- tm_map(corpus, content_transformer(tolower))

## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents

corpus <- tm_map(corpus, removePunctuation)

## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents

corpus <- tm_map(corpus, removeNumbers)

## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents

corpus <- tm_map(corpus, removeWords, stopwords("english"))

## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")):
## transformation drops documents

corpus <- tm_map(corpus, stripWhitespace)

## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents

term_doc_matrix <- TermDocumentMatrix(corpus)
word_frequency <- rowSums(as.matrix(term_doc_matrix))
wordcloud(words = names(word_frequency), freq = word_frequency, min.freq = 50)
```

[illegible]