Analysis of Song Lyrics and Newspaper Headlines

Word Cloud of Song titles (All Years)

```
Titles <- read.csv("ArtistSong.csv")

TitleCorpus <- Corpus(VectorSource(Titles$Song))
TitleCorpus <- tm_map(TitleCorpus, removePunctuation)
TitleCorpus <- tm_map(TitleCorpus, removeWords, c('the', 'this', 'The', 'Version', 'and', stopwords('eng
TitleCorpus <- tm_map(TitleCorpus, stemDocument)

wordcloud(TitleCorpus, max.words = 100, random.order = FALSE)
```


```
dtm <- TermDocumentMatrix(TitleCorpus)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
```

```
##          word freq
## you       you  481
## album   album  466
## love     love  364
## live     live  227
```

```
## your     your  173
## remast remast  167
## blue     blue  147
## for       for  145
## digit   digit  137
## and       and  130
```

Word Cloud of NYT Headlines (1996-present)

```
data(NYTimes)

TitleCorpus2 <- Corpus(VectorSource(NYTimes$Title))
TitleCorpus2 <- tm_map(TitleCorpus2, removePunctuation)
TitleCorpus2 <- tm_map(TitleCorpus2, removeWords, c('the', 'this', 'The', 'and','For','THE', stopwords(
TitleCorpus2 <- tm_map(TitleCorpus2, stemDocument)

wordcloud(TitleCorpus2, max.words = 100, random.order = FALSE)
```


```
dtm2 <- TermDocumentMatrix(TitleCorpus2)
m2 <- as.matrix(dtm2)
v2 <- sort(rowSums(m2),decreasing=TRUE)
d2 <- data.frame(word = names(v2),freq=v2)
head(d2, 10)
```

```
##              word freq
```

```
## new           new  210
## bush         bush  126
## with         with  113
## and           and  104
## iraq         iraq  104
## war           war   86
## overview overview   80
## for           for   79
## nation     nation   76
## plan         plan   75
```

Artist Year and Album Title(1996-present)

```
Year <- read.csv("ArtistYear.csv")
Year <- Year[(Year$Year=='1996' | Year$Year=='1997' | Year$Year=='1998' |Year$Year=='1999' |Year$Year==

TitleCorpus3 <- Corpus(VectorSource(Year$Album))
TitleCorpus3 <- tm_map(TitleCorpus3, removePunctuation)
TitleCorpus3 <- tm_map(TitleCorpus3, removeWords, c('the', 'this', 'The', 'and','For','album','Album','
TitleCorpus3 <- tm_map(TitleCorpus3, stemDocument)

wordcloud(TitleCorpus3, max.words = 100, random.order = FALSE)
```

```
dtm3 <- TermDocumentMatrix(TitleCorpus3)
m3 <- as.matrix(dtm3)
v3 <- sort(rowSums(m3),decreasing=TRUE)
d3 <- data.frame(word = names(v3),freq=v3)
head(d3, 10)
```

```
##      word freq
## you   you  159
## love love   88
## live live   55
## dont dont   46
## all   all   46
## and   and   42
## your your   41
## blue blue   36
## what what   35
## from from   32
```

Create Data Frames for each Year

```
###Million Songs

NYTDate <- mutate(NYTimes, yr = str_sub(NYTimes$Date, start= -2))

Year96 <- Year[(Year$Year=='1996'),]
TitleCorpus96 <- Corpus(VectorSource(Year96$Album))
TitleCorpus96 <- tm_map(TitleCorpus96, removePunctuation)
TitleCorpus96 <- tm_map(TitleCorpus96, removeWords, c('the', 'this', 'The', 'and','For','album','Album'
TitleCorpus96 <- tm_map(TitleCorpus96, stemDocument)
dtm96 <- TermDocumentMatrix(TitleCorpus96)
m96 <- as.matrix(dtm96)
v96 <- sort(rowSums(m96),decreasing=TRUE)
d96 <- data.frame(word = names(v96),freq=v96)
NineSix <- data.frame(head(d96, 5))
NineSix <- mutate(NineSix, Date = '96')

Year97 <- Year[(Year$Year=='1997'),]
TitleCorpus97 <- Corpus(VectorSource(Year97$Album))
TitleCorpus97 <- tm_map(TitleCorpus97, removePunctuation)
TitleCorpus97 <- tm_map(TitleCorpus97, removeWords, c('the', 'this', 'The', 'and','For','album','Album'
TitleCorpus97 <- tm_map(TitleCorpus97, stemDocument)
dtm97 <- TermDocumentMatrix(TitleCorpus97)
m97 <- as.matrix(dtm97)
v97 <- sort(rowSums(m97),decreasing=TRUE)
d97 <- data.frame(word = names(v97),freq=v97)
NineSeven <- data.frame(head(d97, 5))
NineSeven <- mutate(NineSeven, Date = '97')

Year98 <- Year[(Year$Year=='1998'),]
TitleCorpus98 <- Corpus(VectorSource(Year98$Album))
TitleCorpus98 <- tm_map(TitleCorpus98, removePunctuation)
TitleCorpus98 <- tm_map(TitleCorpus98, removeWords, c('the', 'this', 'The', 'and','For','album','Album'
```

```
TitleCorpus98 <- tm_map(TitleCorpus98, stemDocument)
dtm98 <- TermDocumentMatrix(TitleCorpus98)
m98 <- as.matrix(dtm98)
v98 <- sort(rowSums(m98),decreasing=TRUE)
d98 <- data.frame(word = names(v98),freq=v98)
NineEight <- data.frame(head(d98, 5))
NineEight <- mutate(NineEight, Date = '98')

Year99 <- Year[(Year$Year=='1999'),]
TitleCorpus99 <- Corpus(VectorSource(Year99$Album))
TitleCorpus99 <- tm_map(TitleCorpus99, removePunctuation)
TitleCorpus99 <- tm_map(TitleCorpus99, removeWords, c('the', 'this', 'The', 'and','For','album','Album'
TitleCorpus99 <- tm_map(TitleCorpus99, stemDocument)
dtm99 <- TermDocumentMatrix(TitleCorpus99)
m99 <- as.matrix(dtm99)
v99 <- sort(rowSums(m99),decreasing=TRUE)
d99 <- data.frame(word = names(v99),freq=v99)
NineNine <- data.frame(head(d99, 5))
NineNine <- mutate(NineNine, Date = '99')

Common <- rbind(NineSix,NineSeven,NineEight,NineNine)

###NYT

NYT96 <- NYTDate[(NYTDate$yr=='96'),]
NYT96 <- Corpus(VectorSource(NYT96$Title))
NYT96 <- tm_map(NYT96, removePunctuation)
NYT96 <- tm_map(NYT96, removeWords, c('the', 'this', 'The', 'and','For','album','Album','Version','versi
NYT96 <- tm_map(NYT96, stemDocument)
dtmNYT96 <- TermDocumentMatrix(NYT96)
mNYT96 <- as.matrix(dtmNYT96)
vNYT96 <- sort(rowSums(mNYT96),decreasing=TRUE)
dNYT96 <- data.frame(word = names(vNYT96),freq=vNYT96)
NineSixNYT <- data.frame(head(dNYT96, 5))
NineSixNYT <- mutate(NineSixNYT, Date = '96')

NYT97 <- NYTDate[(NYTDate$yr=='97'),]
NYT97 <- Corpus(VectorSource(NYT97$Title))
NYT97 <- tm_map(NYT97, removePunctuation)
NYT97 <- tm_map(NYT97, removeWords, c('the', 'this', 'The', 'and','For','album','Album','Version','versi
NYT97 <- tm_map(NYT97, stemDocument)
dtmNYT97 <- TermDocumentMatrix(NYT97)
mNYT97 <- as.matrix(dtmNYT97)
vNYT97 <- sort(rowSums(mNYT97),decreasing=TRUE)
dNYT97 <- data.frame(word = names(vNYT97),freq=vNYT97)
NineSevenNYT <- data.frame(head(dNYT97, 5))
NineSevenNYT <- mutate(NineSevenNYT, Date = '97')

NYT98 <- NYTDate[(NYTDate$yr=='98'),]
NYT98 <- Corpus(VectorSource(NYT98$Title))
NYT98 <- tm_map(NYT98, removePunctuation)
NYT98 <- tm_map(NYT98, removeWords, c('the', 'this', 'The', 'and','For','album','Album','Version','versi
NYT98 <- tm_map(NYT98, stemDocument)
```

```
dtmNYT98 <- TermDocumentMatrix(NYT98)
mNYT98 <- as.matrix(dtmNYT98)
vNYT98 <- sort(rowSums(mNYT98),decreasing=TRUE)
dNYT98 <- data.frame(word = names(vNYT98),freq=vNYT98)
NineEightNYT <- data.frame(head(dNYT98, 5))
NineEightNYT <- mutate(NineEightNYT, Date = '98')

NYT99 <- NYTDate[(NYTDate$yr=='99'),]
NYT99 <- Corpus(VectorSource(NYT99$Title))
NYT99 <- tm_map(NYT99, removePunctuation)
NYT99 <- tm_map(NYT99, removeWords, c('the', 'this', 'The', 'and','For','album','Album','Version','versi
NYT99 <- tm_map(NYT99, stemDocument)
dtmNYT99 <- TermDocumentMatrix(NYT99)
mNYT99 <- as.matrix(dtmNYT99)
vNYT99 <- sort(rowSums(mNYT99),decreasing=TRUE)
dNYT99 <- data.frame(word = names(vNYT99),freq=vNYT99)
NineNineNYT <- data.frame(head(dNYT99, 5))
NineNineNYT <- mutate(NineNineNYT, Date = '99')

CommonNYT <- rbind(NineSixNYT,NineSevenNYT,NineEightNYT,NineNineNYT)

Common <- mutate(Common, Source = 'MS')
CommonNYT <- mutate(CommonNYT, Source = 'NYT')

CommonAll <- rbind(Common,CommonNYT)
```

Year by Year Comparison of Album Titles and NYT Headlines

```
server <- function(input, output) {

  # Filter data based on selections
  output$table <- DT::renderDataTable(DT::datatable({
    data <- CommonAll
    if (input$Date != "All") {
      data <- data[data$Date == input$Date,]
    }
    if (input$Source != "All") {
      data <- data[data$Source == input$Source,]
    }
    data
  }))

}

ui <- fluidPage(
  titlePanel("Basic DataTable"),

  # Create a new Row in the UI for selectInputs
  fluidRow(
    column(4,
        selectInput("Date",
```

```
                    "Year:",
                    c("All",
                      unique(as.character(CommonAll$Date))))
    ),
    column(4,
        selectInput("Source",
                    "Source:",
                    c("All",
                      unique(as.character(CommonAll$Source))))
    ),
    column(4,
        selectInput("word",
                    "Word:",
                    c("All",
                      unique(as.character(CommonAll$word))))
    )
  ),
  # Create a new row for the table.
  fluidRow(
    DT::dataTableOutput("table")
  )
)

shinyApp(ui = ui, server = server)
```

Shiny applications not supported in static R Markdown documents