haha

#Data Story #........

By Saier Gong

```
<font size=3>
Have you ever felt like that this is the first you've heard a song but have a sense of familarity and be
Did you ever fall in love with a song just because a word or sentence?
Listening to the songs, we can know the stories of the artists and sometimes the stories are just yours
```

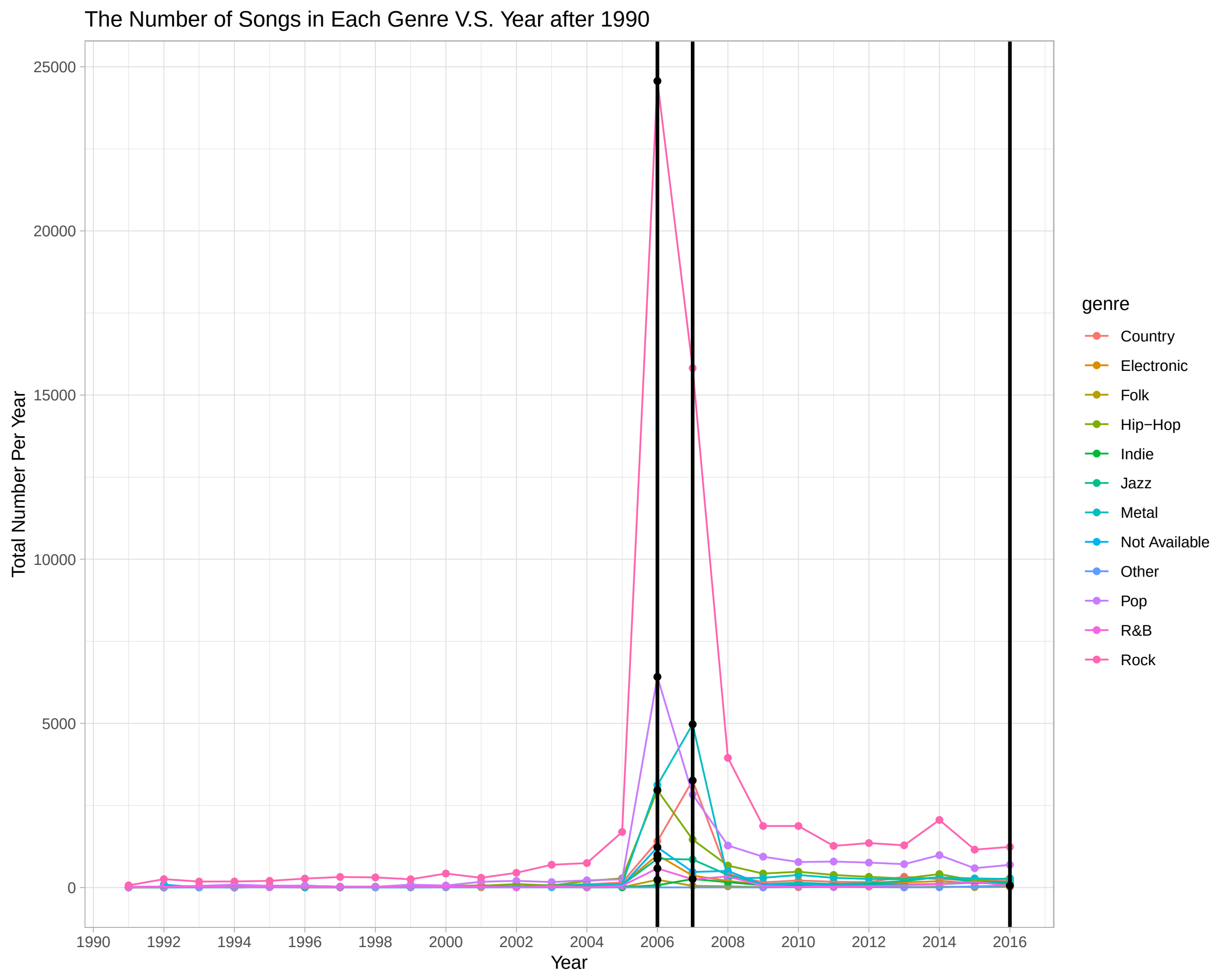
~~~
```r
#ti gang


#mei yi zhong yin yue zai 2006-2007or2008 dou da dao dian feng
#tao lun 06-08 nian ge zhong genre qing gan de bian hua
#qi zhong rock lei yin yue zai dian feng zhi hou yi zhi hen gao chan
~~~

First of all, we take a look at the number of songs in different genres in different years.

```r
lyric<-dt_lyrics


#the number of songs in each genre in each year
peak<-lyric %>%
  filter(year>1960) %>%
  filter(genre!='other') %>%
  group_by(genre,year) %>%
  count() %>%
  ungroup(year) %>%
  arrange(genre,desc(n))

#mei zhong lei xing zai zhe xie nian zhong de zui da chan chu
#the max number of songs in each genre per year through these years
peak_dot<-peak %>%
  group_by(genre) %>%
  summarise(n=max(n)) %>%
  inner_join(peak)
```

We can see that before 1990, the total number of songs in each genre per year is much smaller expecially compared with the number of songs after 1900, therefore, we only consider the songs in each genre per year after 1990.

```r
#total number of songs per gener per year after 1990

peak_plot<-ggplot(data=peak %>% filter(year>1990),mapping = aes(x=year,y=n,color=genre))+
  geom_point()+
  geom_line()+
  labs(title='The Number of Songs in Each Genre V.S. Year after 1990',
       x="Year",
       y="Total Number Per Year")+
  geom_point(data=peak_dot,mapping = aes(x=year,y=n),color='black')+
  geom_vline(data=peak_dot,aes(xintercept = year),size=1)+
  scale_x_continuous(breaks=seq(min(peak$year),max(peak$year),2))+
  theme_light()

peak_plot
```



From the “peak_plot”, we can clearly see that : 1. the total number of songs created in each genre reaches the peak after 2006; 2. in year 2006, the number of songs created in 8 genres reaches peak amount,relatively; 3. in year 2006-2008, the number of songs created in almost each genre is much more higher than other years; 4. the number of songs in genre Rock is higher than the number in any other genre in nearly each year, especially in year 2006-2008.

So, we will focus on the data between 2006 and 2008.

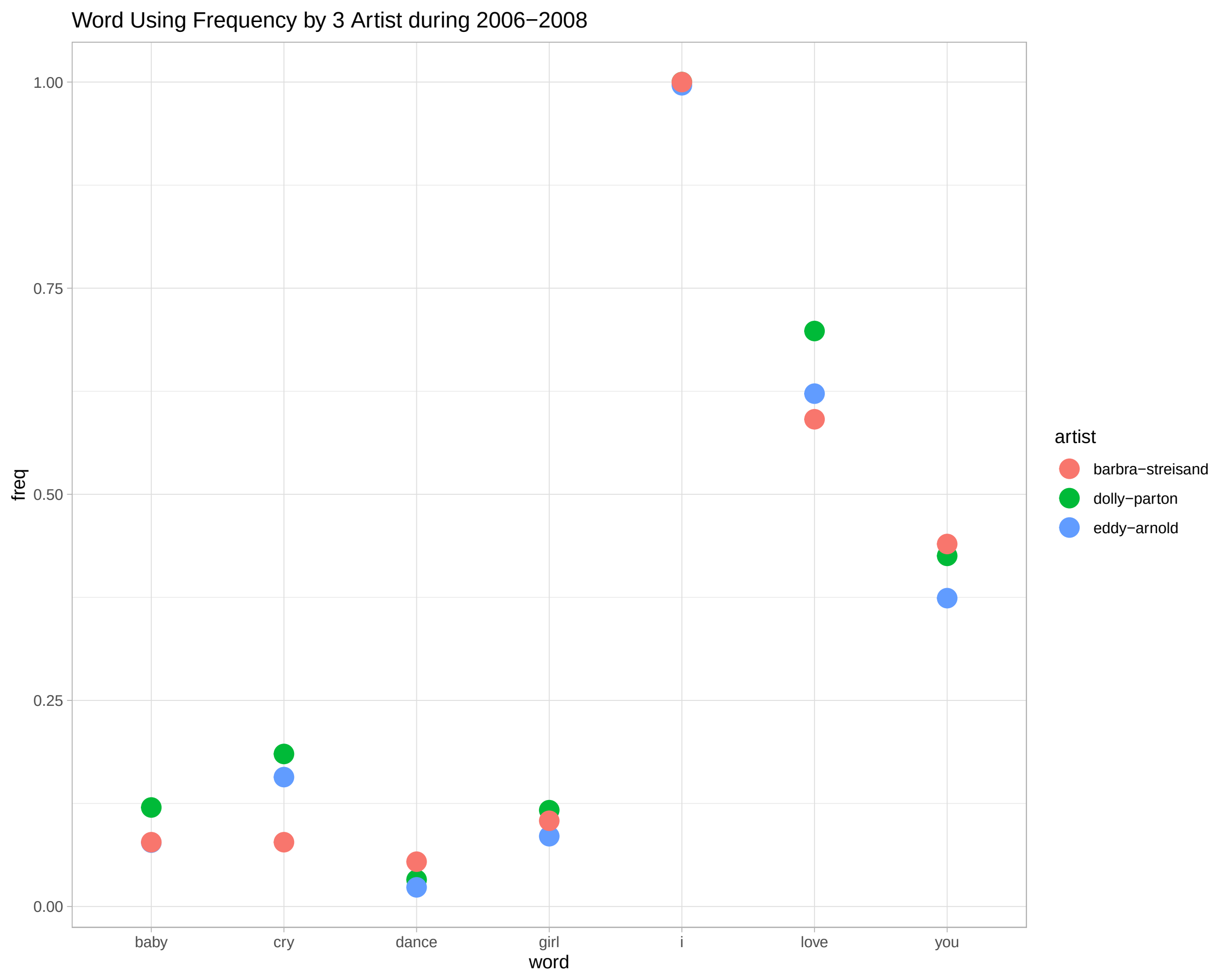
```r
#in Year 2006-2008-----new_lyric
new_lyric<-lyric %>%
  filter(year>1960) %>%
  filter(genre!='other') %>%
  filter(between(year,2006,2008))
```

Which words they would love to use in period 2006-2008?

To answer this question, we need to choose some artists that are most representative of that period (Year 2006-2008). Therefore, we make a list of the 3 artists who had the most songs in that period, and find out the frequencies they used the words in the following list to write their songs during that period.

word list{i,love,you,baby,girl,cry,dance}

```r
song_number_top_3<-new_lyric %>%
  group_by(artist) %>%
  count() %>%
  arrange(desc(n)) %>%
  head(3)


dolly_parton.lyric<-new_lyric %>%
  filter(artist=='dolly-parton')

eddy_arnold.lyric<-new_lyric %>%
  filter(artist=='eddy-arnold')

barbra_streisand.lyric<-new_lyric %>%
  filter(artist=='barbra-streisand')


#### word list{i,love,you,baby,girl,cry,dance}

word<-c("i","love","you","baby","girl","cry","dance")

count_word<-function(df,word){
  sum(map_dbl(df$stemmedwords,function(x) str_detect(x,word)))/length(df$stemmedwords)
}

dolly_freq_word<-tibble(artist="dolly-parton",word=word,freq=map_dbl(word,function(x) count_word(df=dol

eddy_freq_word<-tibble(artist="eddy-arnold",word=word,freq=map_dbl(word,function(x) count_word(df=eddy_

barbra_freq_word<-tibble(artist="barbra-streisand",word=word,freq=map_dbl(word,function(x) count_word(d


word.freq<-rbind(dolly_freq_word,eddy_freq_word,barbra_freq_word)

ggplot(data=word.freq,mapping = aes(x=word,y=freq,color=artist))+
  geom_point(size=5)+
  labs(title = "Word Using Frequency by 3 Artist during 2006-2008",
       xlab="word",ylab="frequency")+
  theme_light()
```



From this plot, we can see that every song they created had word "i", and they used "love" and "you" more than other words. So, it is interesting to imply that 10 years ago people loved to say "i love you" in their songs just like nowadays.(At least I have heard a lot of songs containg "i love you" no matter what language they are.)

##Sentence Analysis of the Top 3 Artists in Period 2006-2008

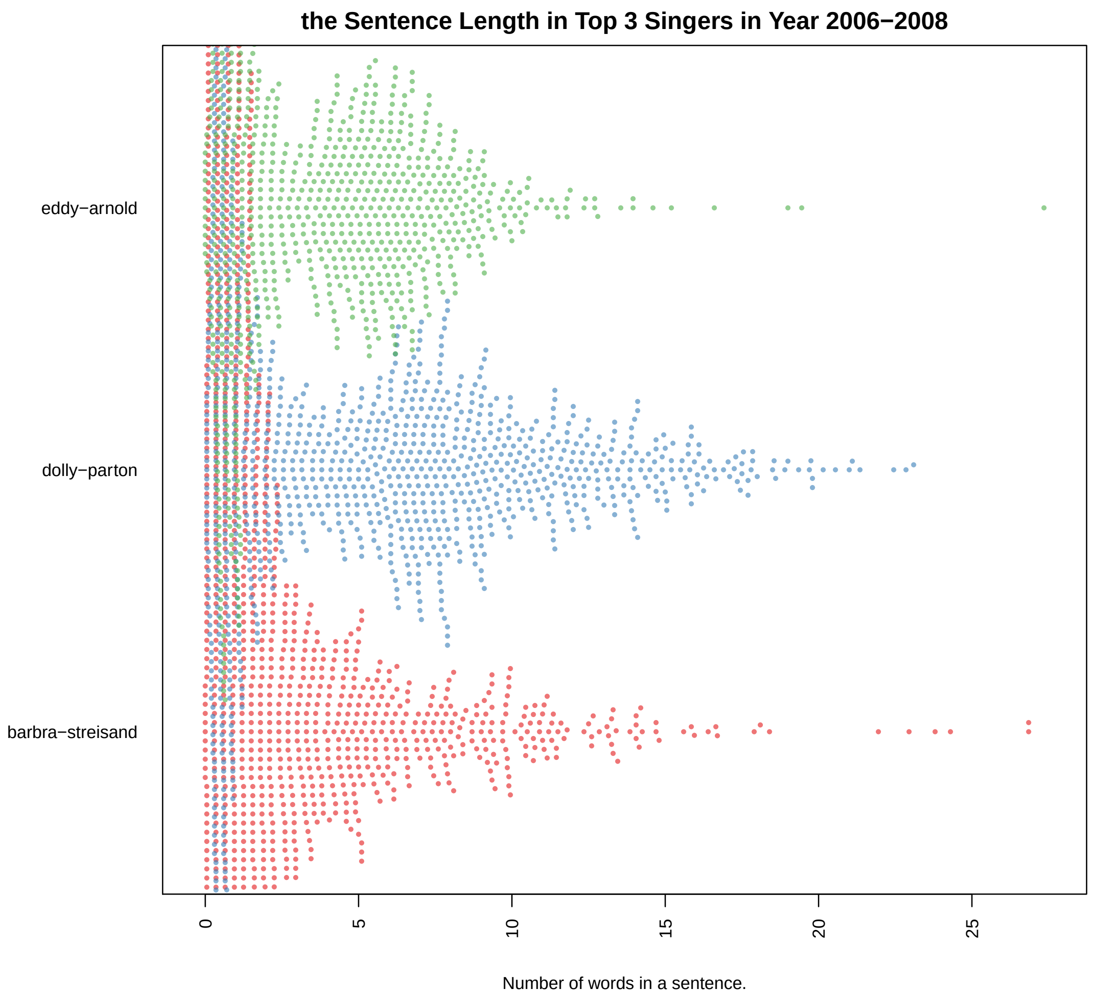
```
#2006-2008 period sentences
#sentence.list=NULL
#for(i in 1:nrow(new_lyric)){
 # sentences=syuzhet::get_sentences(new_lyric$lyrics[i])
#  if(length(sentences)>0){
 #   emotions=matrix(emotion(sentences)$emotion,
#                    nrow=length(sentences),
#                    byrow=T)
#    colnames(emotions)=emotion(sentences[1])$emotion_type
#    emotions=data.frame(emotions)
#    emotions=select(emotions,
#                   anticipation,
#                   joy,
#                   surprise,
#                   trust,
#                   anger,
#                    disgust,
#                    fear,
#                    sadness)
#    word.count=f.word_count(sentences)
    # colnames(emotions)=paste0("emo.", colnames(emotions))
    # in case the word counts are zeros?
    # emotions=diag(1/(word.count+0.01))%*%as.matrix(emotions)
#    sentence.list=rbind(sentence.list,
#                         cbind(new_lyric[i,-ncol(new_lyric)],
#                               sentences=as.character(sentences),
#                               word.count,
#                               emotions,
#                               sent.id=1:length(sentences)
#                               )
#    )
#  }
#}
#names(sentence.list)

#delete some genres that contains much less songs
#sentence.list<-sentence.list%>%
 # filter(genre!='Folk') %>%
#  filter(genre!='Indie') %>%
 # filter(genre!='R&B') %>%
#  filter(!is.na(word.count))

#the rest genres to be analyzed 1."Hip-Hop"  2. "Pop" 3."Metal" 4. "Rock"  5."Country"   6.  "Jazz"  7.
```

First, we need to load the dataset to analyze the sentences.

```
# bao cun xia lai zui jin de ge ci qing gan fen xi shu ju
#save(sentence.list,file="C:/Users/Saier/Desktop/courses/5243/sentence_list.RData")

load("C:/Users/Saier/Desktop/courses/5243/sentence_list.RData")

#write.table sentence.list,file ="sentence_list.csv" sep ="," row.names = F
```

Then, let's continue analyzing the top 3 artists discussed before, to analyze the length of sentences in their songs, respectively.

```
##the top_3 artist discussed before
par(mar=c(4, 11, 2, 2))
sel.comparison=song_number_top_3$artist
top_3=filter(sentence.list, artist%in%sel.comparison)
top_3$artist=factor(top_3$artist)
top_3$artistOrdered=reorder(top_3$artist, top_3$word.count,
                                  mean,
                                  order=T)
beeswarm(word.count/20~artist,
         data=top_3,
         horizontal = TRUE,
         pch=16, col=alpha(brewer.pal(9, "Set1"), 0.6),
         cex=0.55, cex.axis=0.8, cex.lab=0.8,
         spacing=5/nlevels(top_3$artistOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="the Sentence Length in Top 3 Singers in Year 2006-2008")
```

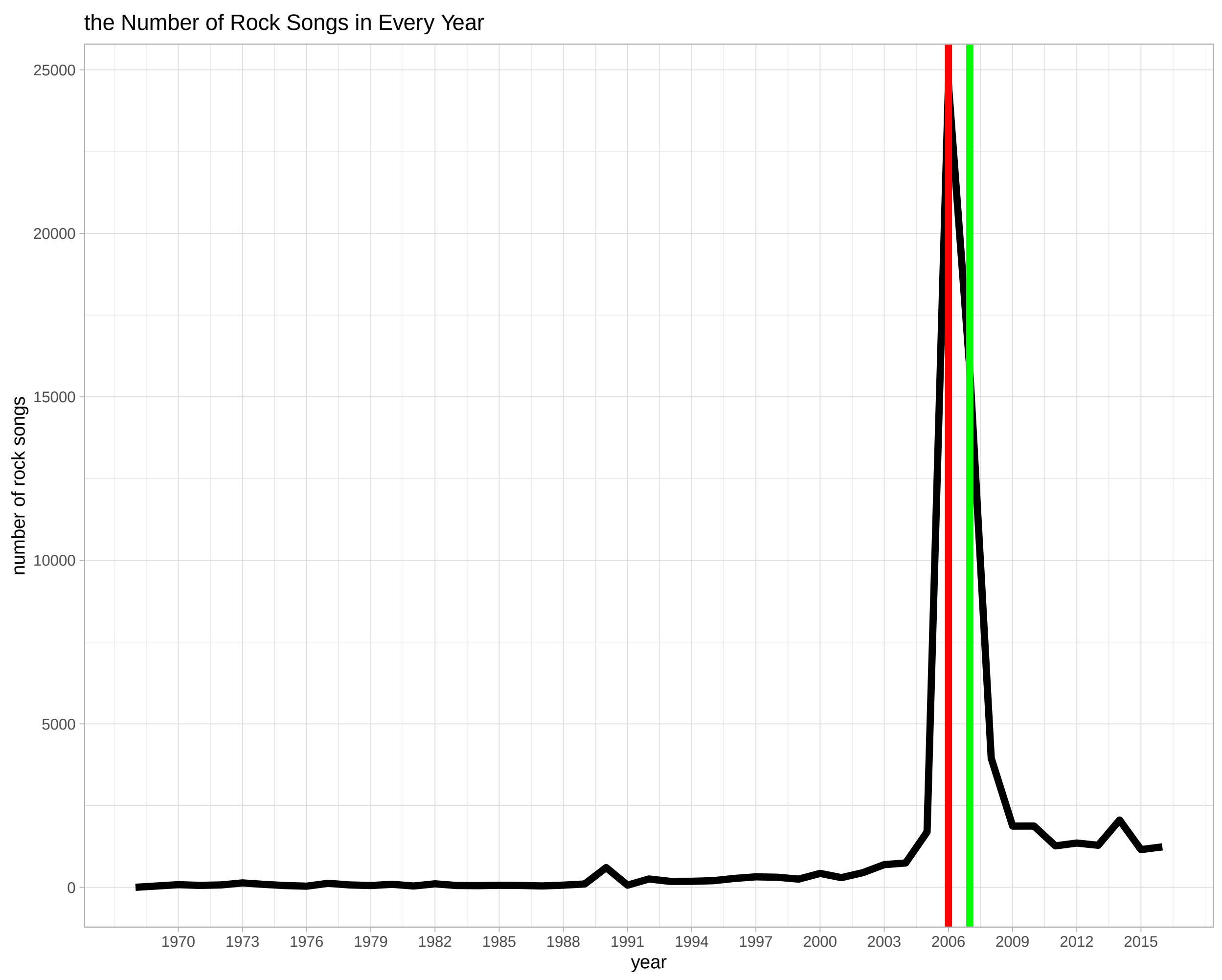


From the plot, we can see that dolly_parton prefer using shorter sentence compared with eddy and barbra. But they three all like to use shorter sentences. Maybe because shorter sentences are easy to remember and can fit the rhythm they are good at perfectly.

##Sentiment Analysis of Genre Rock As we have seen before, much more songs belong to genre Rock than other genres, therefore, it is a good idea to analysis the data and find out why many artist love Rock.

```
#######################filter genre Rock from original lyric containing all years

rock<-lyric %>%
  filter(genre=='Rock')

#the number od Rock songs in every year
rock.number<-rock %>%
  group_by(year) %>%
  count() %>%
  arrange(desc(n))
```

```
ggplot(data=rock.number,mapping = aes(x=year,y=n)) +
  geom_line(size=2)+
  geom_vline(xintercept=2006,color='red',size=2)+
  geom_vline(xintercept=2007,color='green',size=2)+
  scale_x_continuous(breaks = seq(1970,2016,3))+
  labs(title = 'the Number of Rock Songs in Every Year',
       x='year',
       y='number of rock songs')+
  theme_light()
```



From this plot, we can see that in 2006, there are more rock songs than other years.

Then, we want to know which artists contributed most to the huge number of Rock songs compared with other genres.

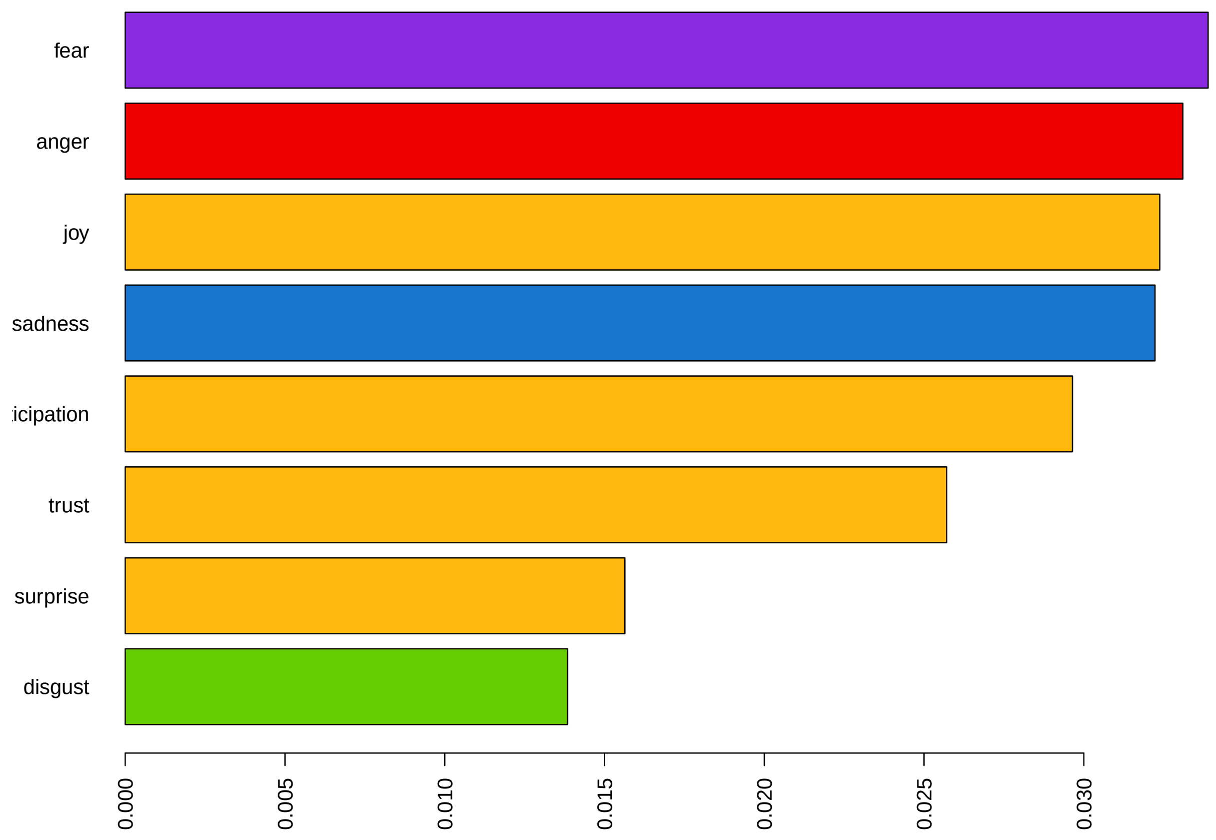
```
rock.number_artist<-rock %>%
  group_by(artist) %>%
  count() %>%
  arrange(desc(n))
```

We can see that elton-john has the most Rock songs, 619, and then it is bob-dylan, 563. So, let's have a look at their feelings in their Rock songs.

First, let's analyze elton-john's songs.

```r
rock.elton_john<-rock %>%
  filter(artist=='elton-john')

#rock.elton_john.list=NULL
#for(i in 1:nrow(rock.elton_john)){
#  sentences=syuzhet::get_sentences(rock.elton_john$lyrics[i])
#  if(length(sentences)>0){
#    emotions=matrix(emotion(sentences)$emotion,
#                    nrow=length(sentences),
#                    byrow=T)
#    colnames(emotions)=emotion(sentences[1])$emotion_type
#    emotions=data.frame(emotions)
#    emotions=select(emotions,
#                   anticipation,
#                   joy,
#                   surprise,
#                   trust,
#                   anger,
#                   disgust,
#                   fear,
#                   sadness)
#    word.count=f.word_count(sentences)
    # colnames(emotions)=paste0("emo.", colnames(emotions))
    # in case the word counts are zeros?
    # emotions=diag(1/(word.count+0.01))%*%as.matrix(emotions)
#    rock.elton_john.list=rbind(rock.elton_john.list,
#                         cbind(rock.elton_john[i,-ncol(rock.elton_john)],
#                               sentences=as.character(sentences),
#                               word.count,
#                               emotions,
#                               sent.id=1:length(sentences)
#                               )
#    )
#  }
#}
#names(rock.elton_john.list)

#save(rock.elton_john.list,file="C:/Users/Saier/Desktop/courses/5243/rock_elton_john_list.RData")
load(file="C:/Users/Saier/Desktop/courses/5243/rock_elton_john_list.RData")
```

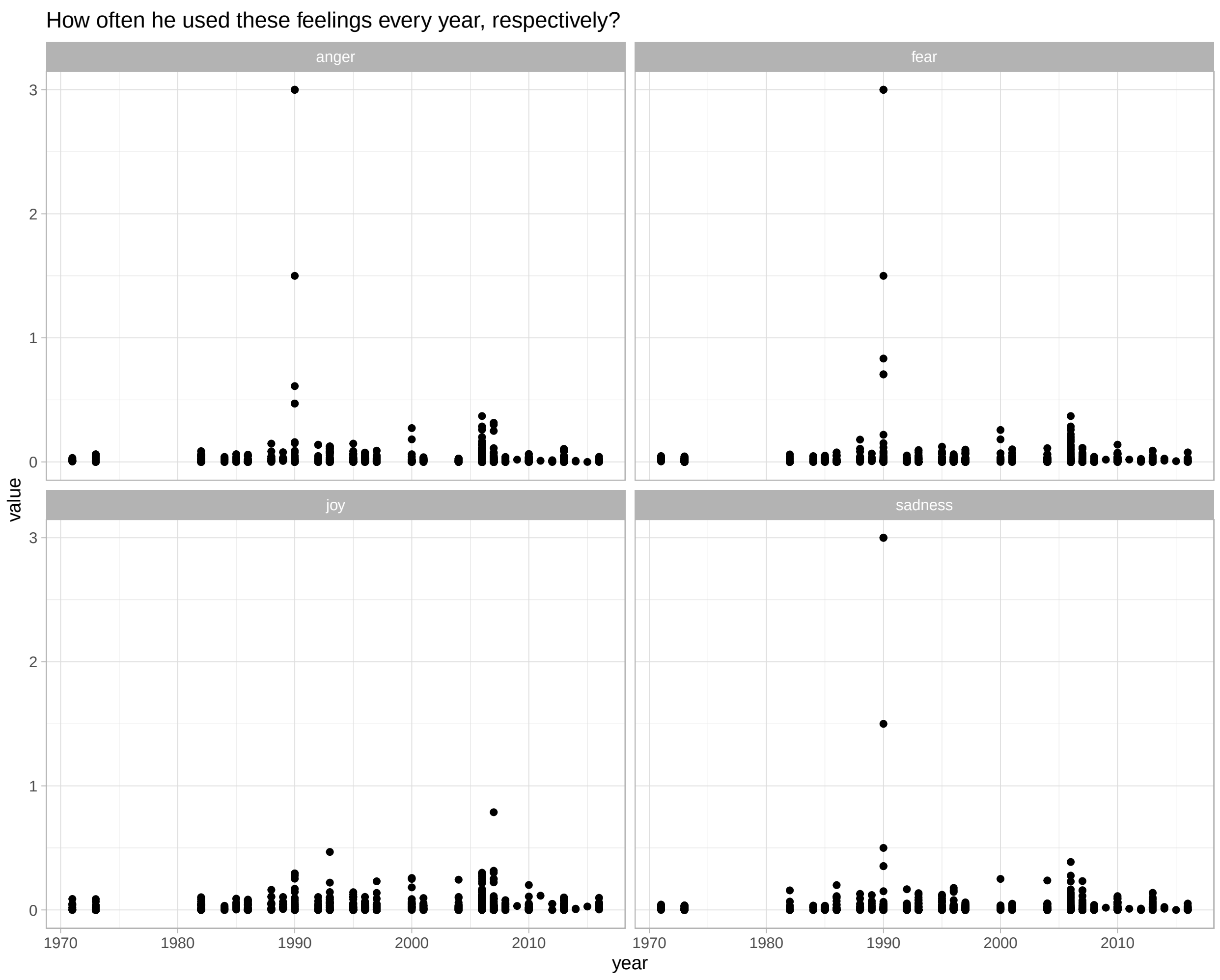
```r
emo.means=colMeans(select(rock.elton_john.list, anticipation:sadness))
col.use=c("darkgoldenrod1", "darkgoldenrod1", "darkgoldenrod1", "darkgoldenrod1",
            "red2", "chartreuse3", "blueviolet","dodgerblue3")
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, main=" ")
```



From this plot, we can dinf out that elton-john usually expressed his fear, anger,joy and sadness in his Rock songs.

So, we will focus on the four main feelings.

```
#four feelings
ggplot(data=rock.elton_john.list %>%select(year,fear,anger,joy,sadness) %>%
         pivot_longer(cols = -year,names_to = 'sentiment',values_to = 'value'))+
  geom_point(mapping = aes(x=year,y=value))+
  facet_wrap(~sentiment)+
  labs(title = "How often he used these feelings every year, respectively?",
       xlab="Year",
       ylab="value")+
  theme_light()
```



From these plots, we can see that in year 1990, he expressed more negative feelings.

Therefore, we will focus on the sentences he sang with negative feelings in year 1990.

```
#negative feelings in year 1990 sentences
df_elton_john_1990=tbl_df(rock.elton_john.list)%>%
  filter(year==1990,word.count<=50) %>%
  select(sentences,fear,anger,sadness) %>%
  filter(fear>0&anger>0&sadness>0)
df_elton_john_1990=as.data.frame(df_elton_john_1990)
(as.character(df_elton_john_1990$sentences))
```

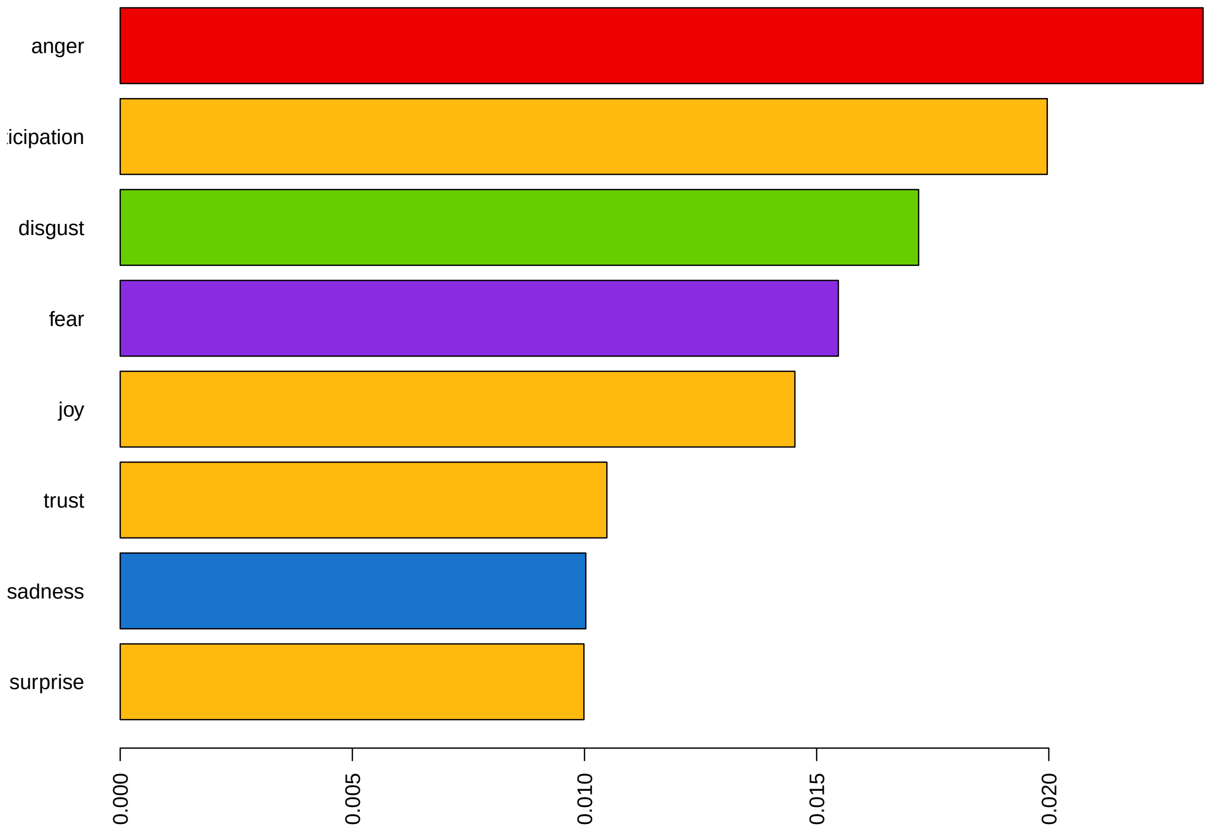
```
## [1] "Slave."
## [2] "Slave."
## [3] "To fight the violence we must be brave,\nHold on strong\nTo the love God gave, slave\nSlave!"
## [4] "Oh slave."
## [5] "To fight the violence we must be brave,\nHold on strong\nTo the love God gave,\nSlave!"
## [6] "Slave."
## [7] "To fight the violence we must be brave,\nHold on strong\nTo the love God gave, slave"
```

Second, let's go on analyzing bob-dylan, who also made great contributions to Rock songs.

```
rock.bob_dylan<-rock %>%
  filter(artist=='bob-dylan')

#rock.bob_dylan.list2=NULL
#for(i in 1:nrow(rock.bob_dylan)){
#  sentences=get_sentences(rock.bob_dylan$lyrics[i])
#  if(length(sentences)>0){
#    emotions=matrix(emotion(sentences)$emotion,
#                    nrow=length(sentences),
#                    byrow=T)
#    colnames(emotions)=emotion(sentences[1])$emotion_type
#    emotions=data.frame(emotions)
#    emotions=select(emotions,
#                   anticipation,
#                   joy,
#                   surprise,
#                   trust,
#                   anger,
#                   disgust,
#                   fear,
#                   sadness)
#    word.count=f.word_count(sentences)
    # colnames(emotions)=paste0("emo.", colnames(emotions))
    # in case the word counts are zeros?
    # emotions=diag(1/(word.count+0.01))%*%as.matrix(emotions)
#    rock.bob_dylan.list2=rbind(rock.bob_dylan.list2,
#                         cbind(rock.bob_dylan[i,-ncol(rock.bob_dylan)],
#                               sentences=as.character(sentences),
#                               word.count,
#                               emotions,
#                               sent.id=1:length(sentences)
#                               )
#    )
#  }
#}
#names(rock.bob_dylan.list)

#save(rock.bob_dylan.list,file="C:/Users/Saier/Desktop/courses/5243/rock_bob_dylan_list.RData")
load("C:/Users/Saier/Desktop/courses/5243/rock_bob_dylan_list.RData")
```

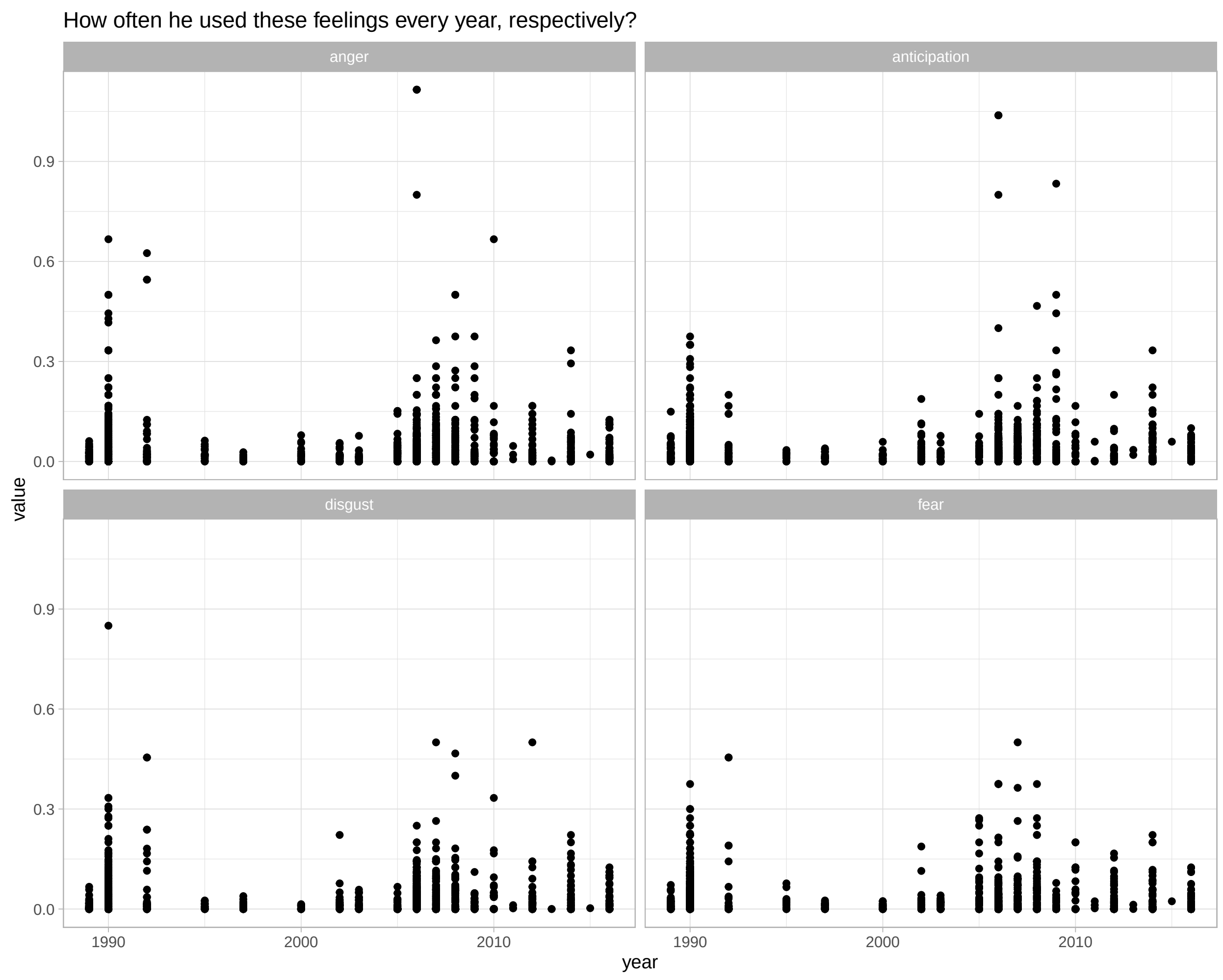
We will use the same steps as we analyze elton-john above.

```
emo.means=colMeans(select(rock.bob_dylan.list, anticipation:sadness))
col.use=c("darkgoldenrod1", "darkgoldenrod1", "darkgoldenrod1", "darkgoldenrod1",
            "red2", "chartreuse3", "blueviolet","dodgerblue3")
barplot(emo.means[order(emo.means)], las=2, col=col.use[order(emo.means)], horiz=T, main=" ")
```



From this plot, we can see that bob usually expressed anger, anticipation, disgust and fear in his Rock songs. So, we will focus on the four main feelings.

```
#four feelings
ggplot(data=rock.bob_dylan.list %>%select(year,anger,anticipation,disgust,fear) %>%
         pivot_longer(cols = -year,names_to = 'sentiment',values_to = 'value'))+
  geom_point(mapping = aes(x=year,y=value))+
  facet_wrap(~sentiment)+
  labs(title = "How often he used these feelings every year, respectively?",
       xlab="Year",
       ylab="value")+
  theme_light()
```



From these plots, we can see that bob expressed anger and anticipation more in year 2006. Let's see the sentences about anger and anticipation, respectively.

```
#2006 anger feeling
df_bob_dylan_2006_anger=tbl_df(rock.bob_dylan.list)%>%
  filter(year==2006,word.count<=10) %>%
  select(sentences,anger) %>%
  filter(anger>0)
df_bob_dylan_2006_anger=as.data.frame(df_bob_dylan_2006_anger)
(as.character(df_bob_dylan_2006_anger$sentences))
```

```
## [1] "Why'd you have to treat me so bad?"
## [2] "Good evening, ladies and gentlemen!"
## [3] "Do you lie in bed and stare at the stars?"
## [4] "Do you have any morals?"
## [5] "Stand in one place till your feet begin to hurt."
## [6] "I'm sad and lonely too."
## [7] "He said \"Young man is you name Brown?"
## [8] "Stay far from the fence\nWith the 'lectricity sting."
```

```r
df_bob_dylan_2006_anticipation=tbl_df(rock.bob_dylan.list)%>%
  filter(year==2006,word.count<=10) %>%
  select(sentences,anticipation) %>%
  filter(anticipation>0)
df_bob_dylan_2006_anticipation=as.data.frame(df_bob_dylan_2006_anticipation)
(as.character(df_bob_dylan_2006_anticipation$sentences))
```

```
## [1] "Good evening, ladies and gentlemen!"
## [2] "Do you lie in bed and stare at the stars?"
## [3] "Do you have any morals?"
## [4] "Wallflower, wallflower\nWon't you dance with me?"
## [5] "Wallflower, wallflower\nWon't you dance with me?"
## [6] "Wallflower, wallflower\nWon't you dance with me?"
## [7] "I am just a poor boy, baby,\nLookin' to connect."
## [8] "He said \"Young man is you name Brown?"
```

Also from the above plots, we can see that he expressed disgust more in year 1990.

```r
df_bob_dylan_1990_disgust=tbl_df(rock.bob_dylan.list)%>%
  filter(year==1990,word.count<=10) %>%
  select(sentences,disgust) %>%
  filter(disgust>0)
df_bob_dylan_1990_disgust=as.data.frame(df_bob_dylan_1990_disgust)
(as.character(df_bob_dylan_1990_disgust$sentences))
```

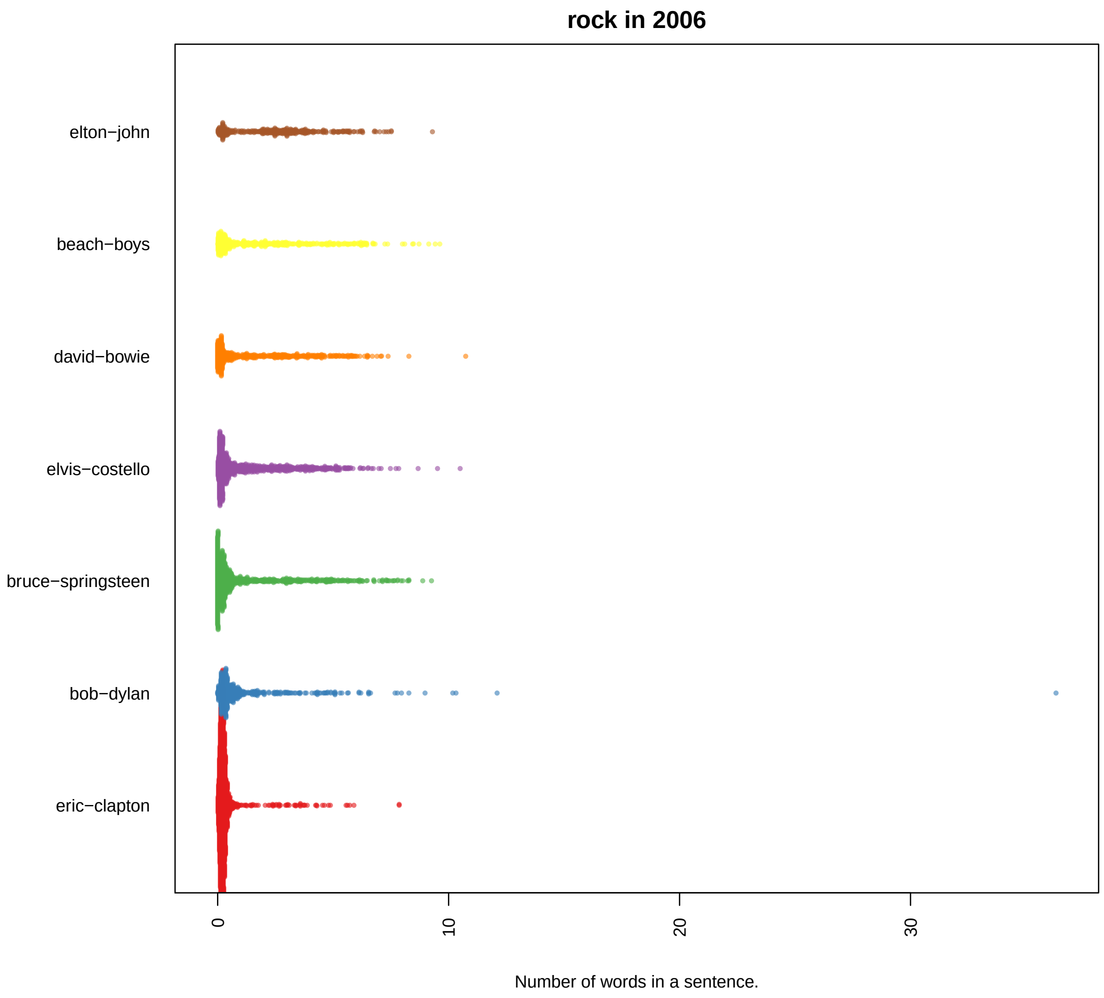
```
##  [1] "Does he change the course of rivers?"
##  [2] "When He healed the blind and crippled, did they see?"
##  [3] "When He healed the blind and crippled, did they see?"
##  [4] "When He healed the blind and crippled, did they see?"
##  [5] "When He healed the blind and crippled, did they see?"
##  [6] "Oh, where have you been, my blue-eyed son?"
##  [7] "change the sails."
##  [8] "Guess I owe You some kind of apology."
##  [9] "Saw him disappear by a tree near a lake ."
## [10] "How many more without any reward?"
## [11] "Tough mama\nCan I blow a little smoke on you?"
## [12] "Oh, child, why you want to hurt me?"
## [13] "Do you love me, or are you just extending goodwill?"
## [14] "Or is your love in vain?"
## [15] "Do you know my world, do you know my kind?"
## [16] "Or is your love in vain?"
## [17] "Or is your love in vain?"
## [18] "\"Pray for peace!\""
## [19] "Equality, liberty, humility, simplicity."
## [20] "Mercury, gravity, nobility, humility."
## [21] "Are you ready for that terrible swift sword?"
## [22] "In the stillness of the midnight,\nPrecious sacred scenes unfold."
```

##Topic Modeling about Rock Songs in Year 2006 Why we choose the year 2006? Because from early dataset, there are far more rock songs in 2006 than any other years. We will choose some represents to analyze this topic.

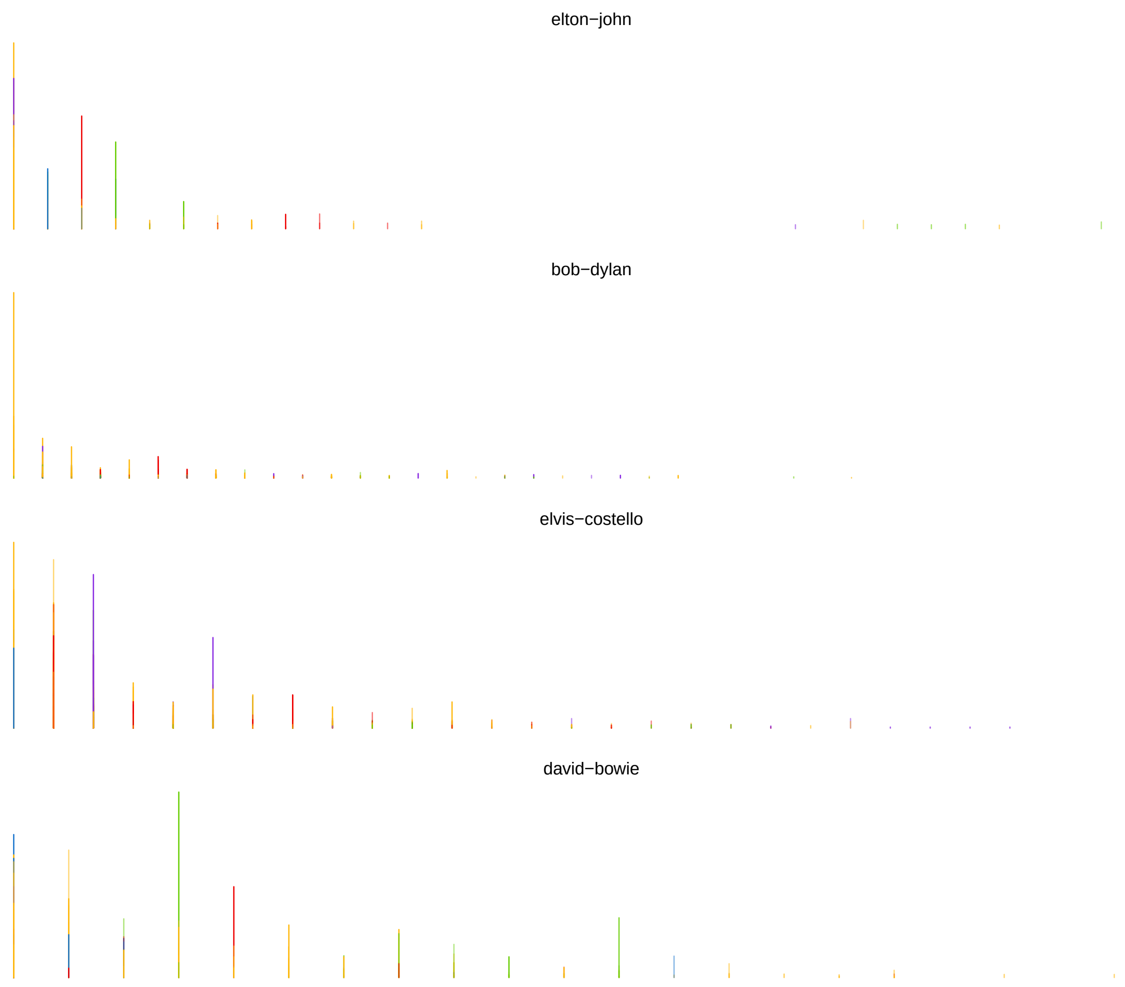
```
#2006 has the most rock songs
#choose the person which has more than 400 songs


rock_2006<-rock %>%
  filter(year==2006) %>%
  inner_join(rock.number_artist %>% filter(n>400))

#rock_2006.list=NULL
#for(i in 1:nrow(rock_2006)){
#  sentences=syuzhet::get_sentences(rock_2006$lyrics[i])
#  if(length(sentences)>0){
#    emotions=matrix(emotion(sentences)$emotion,
#                    nrow=length(sentences),
#                    byrow=T)
#    colnames(emotions)=emotion(sentences[1])$emotion_type
#    emotions=data.frame(emotions)
#    emotions=select(emotions,
#                   anticipation,
#                   joy,
#                   surprise,
#                   trust,
#                   anger,
#                   disgust,
#                   fear,
#                   sadness)
#    word.count=f.word_count(sentences)
    # colnames(emotions)=paste0("emo.", colnames(emotions))
    # in case the word counts are zeros?
    # emotions=diag(1/(word.count+0.01))%*%as.matrix(emotions)
#    rock_2006.list=rbind(rock_2006.list,
#                         cbind(rock_2006[i,-ncol(rock_2006)],
#                               sentences=as.character(sentences),
#                               word.count,
#                               emotions,
#                               sent.id=1:length(sentences)
#                               )
#    )
#  }
#}
#names(rock_2006.list)

#save(rock_2006.list,file="C:/Users/Saier/Desktop/courses/5243/rock_2006_list.RData")

load("C:/Users/Saier/Desktop/courses/5243/rock_2006_list.RData")
```

First,let's have a look at its properties of sentences.

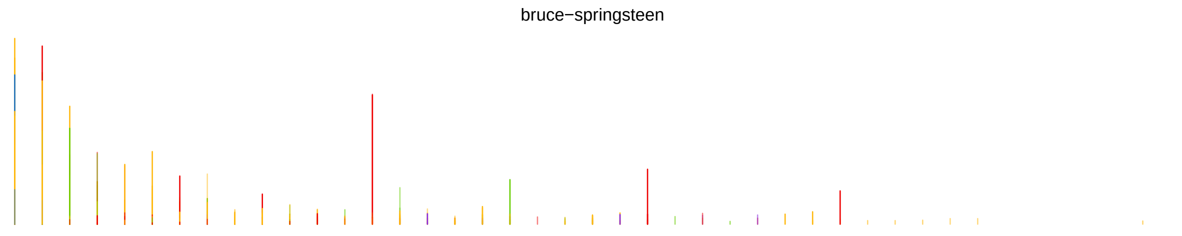
```
par(mar=c(4, 11, 2, 2))
sel.comparison=unique(rock_2006.list$artist)
rock_2006_sel=filter(rock_2006.list, artist%in%sel.comparison)
rock_2006_sel$artist=factor(rock_2006_sel$artist)
rock_2006_sel$artistOrdered=reorder(rock_2006_sel$artist,
                                  rock_2006_sel$word.count,
                                  mean,
                                  order=T)
beeswarm(word.count/50~artistOrdered,
         data=rock_2006_sel,
         horizontal = TRUE,
         pch=16, col=alpha(brewer.pal(9, "Set1"), 0.6),
         cex=0.55, cex.axis=0.8, cex.lab=0.8,
         spacing=1.2/nlevels(rock_2006_sel$artistOrdered),
         las=2, xlab="Number of words in a sentence.", ylab="",
         main="rock in 2006")
```

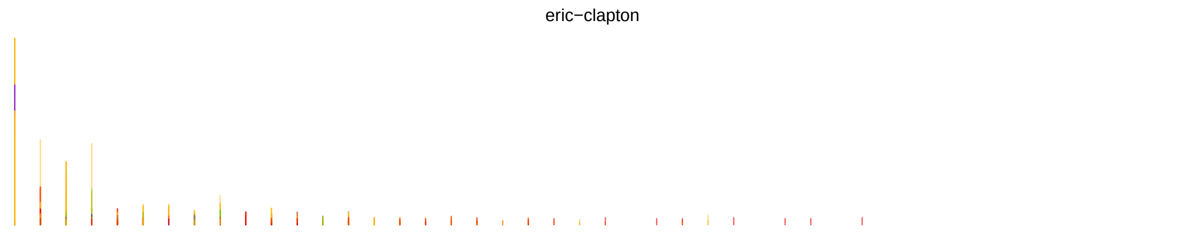


```
par(mfrow=c(4,1), mar=c(1,0,2,0), bty="n", xaxt="n", yaxt="n", font.main=1)
f.plotsent.len(In.list=rock_2006.list, InArtist="elton-john")
f.plotsent.len(In.list=rock_2006.list, InArtist="bob-dylan")
f.plotsent.len(In.list=rock_2006.list, InArtist="elvis-costello")
f.plotsent.len(In.list=rock_2006.list, InArtist="david-bowie")
```



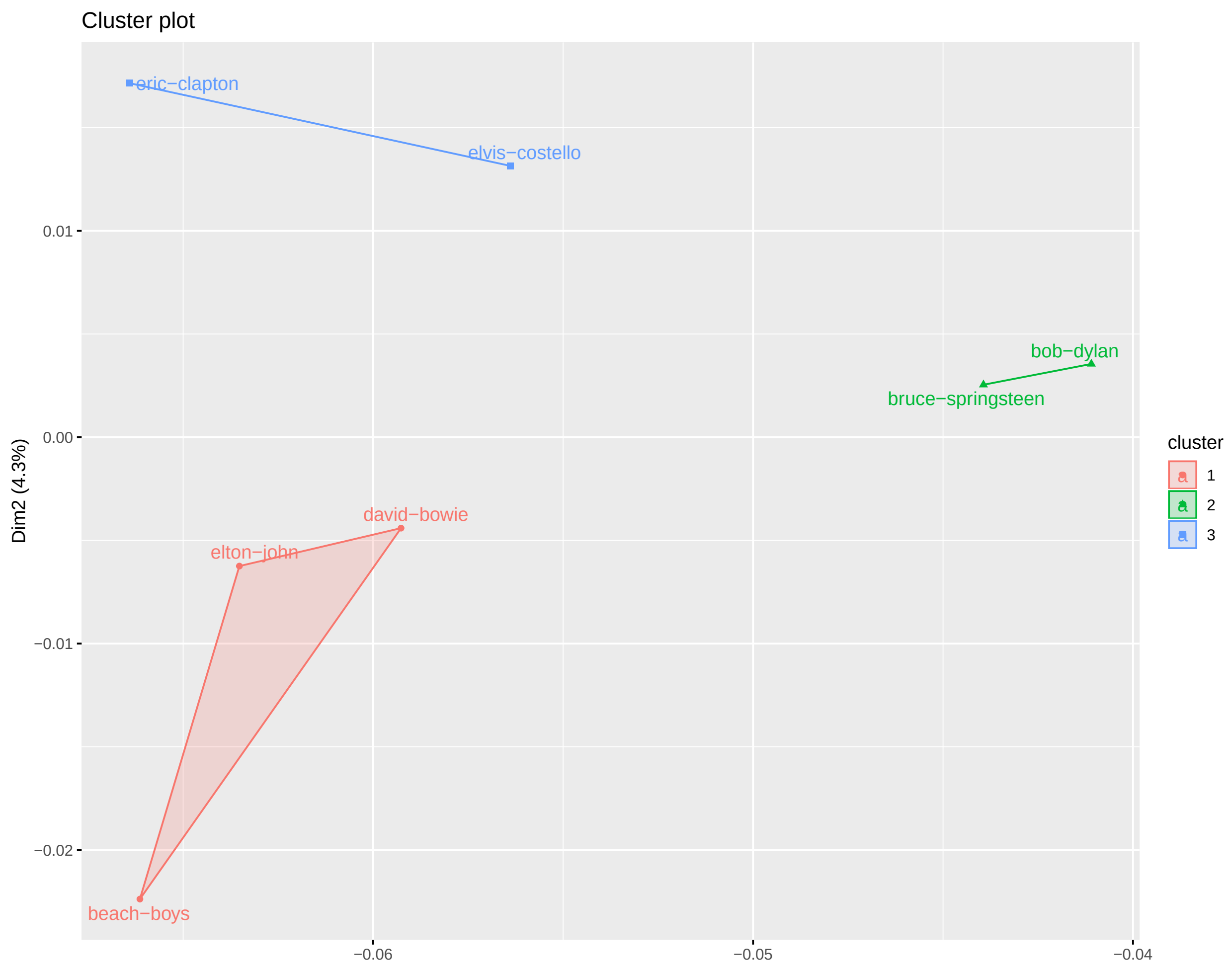
```r
f.plotsent.len(In.list=rock_2006.list, InArtist="bruce-springsteen")
f.plotsent.len(In.list=rock_2006.list, InArtist="eric-clapton")
f.plotsent.len(In.list=rock_2006.list, InArtist="beach-boys")
```





From this plots, we can see: 1.elton-john had more joy, anger and fear; 2.bob-dylan had more joy or anticipation; 3.elvis-costello has more fear; 4.david-bowie had more disgust; 5.bruce-springsteen had more joy and anger; 6.eric-clapton had more anger and joy; 7.beach-boys had more joy and sadness.

```
rock_2006.summary=tbl_df(rock_2006_sel)%>%
  #group_by(paste0(type, File))%>%
  group_by(artist)%>%
  summarise(
    anger=mean(anger),
    anticipation=mean(anticipation),
    disgust=mean(disgust),
    fear=mean(fear),
    joy=mean(joy),
    sadness=mean(sadness),
    surprise=mean(surprise),
    trust=mean(trust)
    #negative=mean(negative),
    #positive=mean(positive)
  )
rock_2006.summary=as.data.frame(rock_2006.summary)
rownames(rock_2006.summary)=as.character((rock_2006.summary[,1]))
km.res=kmeans(rock_2006.summary[,-1], iter.max=200,
                3)
fviz_cluster(km.res,
             stand=F, repel= TRUE,
             data = rock_2006.summary[,-1], xlab="", xaxt="n",
             show.clust.cent=FALSE)
```



If we want to divide these 7 artist into 3 groups based on their feeling expression, we can have elton-john, david-bowie and beach-boys in green group, bob-dylan amd bruch-springsteen in red group, and eric-clapton and elvis-costello in blue group.

After we get familiar with the data, let's start topic modeling.

```
#topic modeling


corpus.list=rock_2006.list[2:(nrow(rock_2006.list)-1), ]
sentence.pre=rock_2006.list$sentences[1:(nrow(rock_2006.list)-2)]
sentence.post=rock_2006.list$sentences[3:(nrow(rock_2006.list)-1)]
corpus.list$snipets=paste(sentence.pre, corpus.list$sentences, sentence.post, sep=" ")
rm.rows=(1:nrow(corpus.list))[corpus.list$sent.id==1]
rm.rows=c(rm.rows, rm.rows-1)
corpus.list=corpus.list[-rm.rows, ]
```

```r
docs <- Corpus(VectorSource(corpus.list$snipets))
```

```r
docs <-tm_map(docs,content_transformer(tolower))

docs <- tm_map(docs, removePunctuation)

docs <- tm_map(docs, removeNumbers)

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

docs <- tm_map(docs, stripWhitespace)

docs <- tm_map(docs,stemDocument)
```

```r
## generate document-term matrices
dtm <- DocumentTermMatrix(docs)
rowTotals <- apply(dtm , 1, sum)
dtm  <- dtm[rowTotals> 0, ]
corpus.list=corpus.list[rowTotals>0, ]
```

```r
## run LDA for all rock songs in 2006 and artists who had more than 400 songs in total
burnin <- 4000
iter <- 2000
thin <- 500
seed <-list(2003,5,63,100001,765)
nstart <- 5
best <- TRUE

# number of topics
k <- 10

# run LDA using Gibbs sampling
#ldaOut <-LDA(dtm, k, method="Gibbs", control=list(nstart=nstart,
#                                                  seed = seed, best=best,
#                                                   burnin = burnin, iter = iter,
#                                                   thin=thin))

# write out results
#save(ldaOut,file = "C:/Users/Saier/Desktop/courses/5243/ldaOut.RData")
load("C:/Users/Saier/Desktop/courses/5243/ldaOut.RData")
ldaOut.topics <- as.matrix(topics(ldaOut))
#write.csv(ldaOut.topics,file=paste("../output/LDAGibbs",k,"DocsToTopics.csv"))

# top 20 terms in each topic
ldaOut.terms <- as.matrix(terms(ldaOut,20))
ldaOut.terms
```

```
##       Topic 1  Topic 2 Topic 3  Topic 4  Topic 5  Topic 6 Topic 7  Topic 8
##  [1,] "tell"   "get"   "town"   "know"   "got"    "can"   "love"   "time"
##  [2,] "danc"   "gonna" "play"   "take"   "well"   "one"   "heart"  "day"
##  [3,] "eye"    "yeah"  "around" "will"   "back"   "just"  "way"    "chorus"
##  [4,] "name"   "feel"  "home"   "let"    "away"   "make"  "life"   "night"
##  [5,] "tri"     "now"    "stand"  "right"  "walk"   "hear"  "ive"    "turn"
##  [6,] "babi"    "man"    "littl"  "dont"   "yes"    "there" "like"   "face"
##  [7,] "cant"    "shes"   "better" "ill"    "aint"   "hold"  "dream"  "last"
##  [8,] "roll"    "help"   "door"   "wont"   "red"    "keep"  "gone"   "light"
##  [9,] "what"    "anoth"  "new"    "youll"  "now"    "aint"  "now"    "old"
## [10,] "realli"  "good"   "wall"   "believ" "theyr"  "look"  "man"    "alway"
## [11,] "cri"     "lord"   "there"  "leav"   "street" "noth"  "caus"   "pass"
## [12,] "put"     "real"   "old"    "still"  "hot"    "found" "fall"   "bad"
## [13,] "blue"    "ride"   "dark"   "woman"  "arm"    "ever"  "youv"   "like"
## [14,] "magic"   "done"   "saw"    "ever"   "fun"    "mani"  "live"   "black"
## [15,] "jump"    "work"   "said"   "now"    "pretti" "gotta" "everi"  "lost"
## [16,] "time"    "didnt"  "run"    "home"   "miss"   "son"   "broken" "place"
## [17,] "long"    "talk"   "soul"   "die"    "took"   "river" "peopl"  "line"
## [18,] "must"    "sun"    "watch"  "ground" "sale"   "step"  "chang"  "told"
## [19,] "left"    "need"   "band"   "long"   "rock"   "shot"  "someon" "everi"
## [20,] "mind"    "ladi"   "santa"  "break"  "stay"   "stop"  "hurt"   "seem"
##       Topic 9   Topic 10
##  [1,] "say"     "dont"
##  [2,] "your"    "come"
##  [3,] "just"    "want"
##  [4,] "thing"   "babi"
##  [5,] "never"   "girl"
##  [6,] "find"    "like"
##  [7,] "think"   "hey"
##  [8,] "give"    "see"
##  [9,] "see"     "littl"
## [10,] "cant"    "hand"
## [11,] "said"    "tonight"
## [12,] "look"    "somebodi"
## [13,] "someth"  "start"
## [14,] "even"    "world"
## [15,] "much"    "nobodi"
## [16,] "two"     "side"
## [17,] "though"  "song"
## [18,] "care"    "boy"
## [19,] "friend"  "right"
## [20,] "everyth" "sing"
```

```r
# probabilities associated with each topic assignment
topicProbabilities <- as.data.frame(ldaOut@gamma)
terms.beta=ldaOut@beta
terms.beta=scale(terms.beta)
topics.terms=NULL
for(i in 1:k){
  topics.terms=rbind(topics.terms, ldaOut@terms[order(terms.beta[i,], decreasing = TRUE)[1:7]])
}

topics.terms
```

```
##       [,1]    [,2]     [,3]   [,4]    [,5]     [,6]      [,7]
##  [1,] "what"  "magic"  "fame" "willi" "wake"   "ooh"     "veronica"
##  [2,] "gonna" "yeah"   "real" "ladi"  "bomp"   "brother" "hors"
##  [3,] "wall"  "santa"  "land" "claus" "crazi"  "hulli"   "across"
##  [4,] "take"  "believ" "low"  "hang"  "pay"    "almost"  "load"
##  [5,] "arm"   "fun"    "sale"  "rock"  "freehold" "daddi"   "hous"
##  [6,] "gotta" "step"   "shot"  "bell"  "secret"   "sea"     "storm"
##  [7,] "life"  "someon" "war"   "sold"  "lust"     "modern"  "careless"
##  [8,] "day"   "line"   "guess" "gold"  "luck"     "captain" "brave"
##  [9,] "think" "give"   "lover" "drop"  "clock"    "echo"    "punch"
## [10,] "youd"  "cross"  "comea" "music" "rais"     "paradis" "style"
```

```
#
```

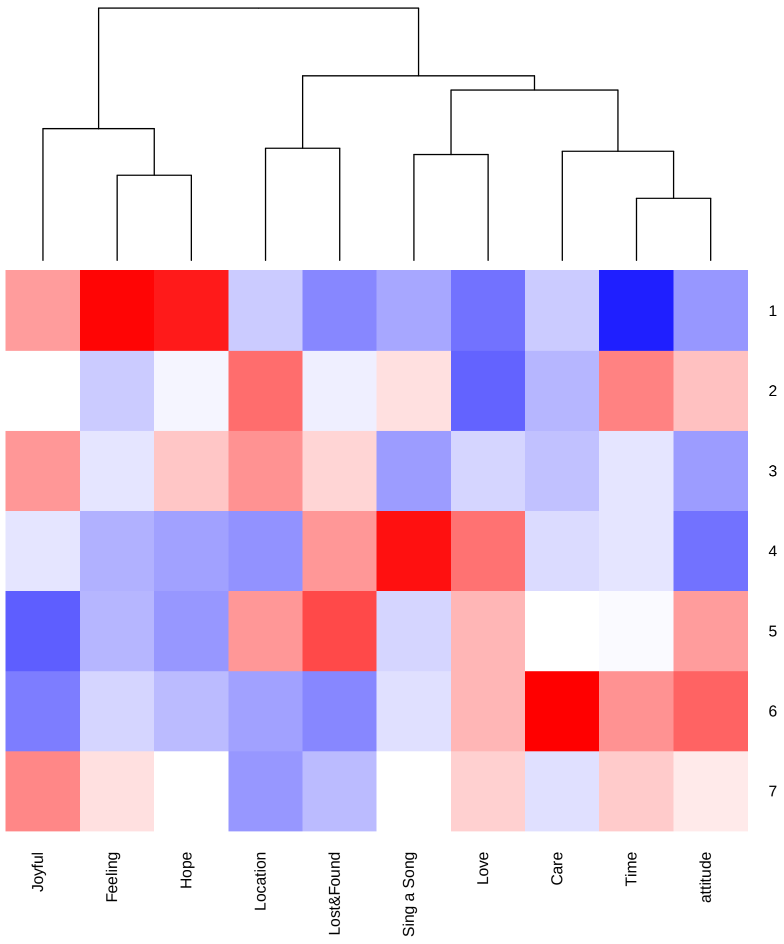
I set the topic numbers to be 10. I manually tag them as "Sing a Song", "Feeling", "Location", "attitude", "Joyful", "Lost&Found", "Love", "Time", "Care", "Hope". Because : Topic 1 contains the key words: "dance","roll"; Topic 2 contains "yeah","feel"; Topic 3 contains "town","home"; Topic 4 contains "will","right"; Topic 5 contains "well","back"; etc.

Based on the most popular terms and the most salient terms for each topic, we assign a hashtag to each topic.

```
topics.hash=c("Sing a Song", "Feeling", "Location", "attitude", "Joyful", "Lost&Found", "Love", "Time",
corpus.list$ldatopic=as.vector(ldaOut.topics)
corpus.list$ldahash=topics.hash[ldaOut.topics]
colnames(topicProbabilities)=topics.hash
corpus.list.df=cbind(corpus.list, topicProbabilities)
```

We use heatmap to see the weight allocation of topics for each artist:

```
par(mar=c(1,1,1,1))
topic.summary=tbl_df(corpus.list.df)%>%
              select(artist, `Sing a Song`:Hope)%>%
              group_by(artist)%>%
              summarise_each(funs(mean))
topic.summary=as.data.frame(topic.summary)
heatmap.2(as.matrix(topic.summary[,-1]),Rowv = FALSE,
           scale = "column", key=F,
           col = bluered(100),
           cexRow = 0.9, cexCol = 0.9, margins = c(8, 8),
           trace = "none", density.info = "none")
```


1
2
3
4
5
6
7
Joyful
Feeling
Hope
Location
Lost&Found
Sing a Song
Love
Care
Time
attitude