Bob Dylan Project

Code

Hide

```
NA
NA
```

Hide

```
library(dplyr)
```

```
There were 38 warnings (use warnings() to see them)
```

Hide

```
setwd("C:/Users/lavin/Desktop/Bob Dylan Project/BobDylan")
#read dataset
bob <- read.csv("BobDylanDataset.csv", header = TRUE, sep= ";", stringsAsFactors = FALSE)
## DATA CLEANING
# function to expand contractions in an English-language source
fix.contractions <- function(doc) {
  doc <- gsub("won't", "will not", doc)
  doc <- gsub("can't", "can not", doc)
  doc <- gsub("n't", " not", doc)
  doc <- gsub("'ll", " will", doc)
  doc <- gsub("'re", " are", doc)
  doc <- gsub("'ve", " have", doc)
  doc <- gsub("'m", " am", doc)
  doc <- gsub("'d", " would", doc)
  # 's could be 'is' or could be possessive: it has no expansion
  doc <- gsub("'s", "", doc)
  return(doc)
}
# fix (expand) contractions
bob$lyric <- sapply(bob$lyric, fix.contractions)

# function to remove special characters, if any
removeSpecialChars <- function(x) gsub("[^a-zA-Z0-9 ]", " ", x)
# remove special characters
bob$lyric <- sapply(bob$lyric, removeSpecialChars)
# convert everything to lower case
bob$lyric <- sapply(bob$lyric, tolower)

#create buckets and group the years into decades
bob <- bob %>%
  mutate(decade =
           ifelse(bob$year %in% 1962:1969, "1960s",
                  ifelse(bob$year %in% 1970:1979, "1970s",
                         ifelse(bob$year %in% 1980:1989, "1980s",
                                ifelse(bob$year %in% 1990:1999, "1990s",
                                       ifelse(bob$year %in% 2000:2009, "2000s",
                                              ifelse(bob$year %in% 2010:2017, "2010s",
                                                     "NA")))))))
#REMOVE INSTRUMENTAL SONGS
#bob$lyric=="instrumental"
bob <- bob[!(bob$lyric=="instrumental"),]
# 8 songs were instrumental, now the dataset contains 440 "lyrical" songs
```

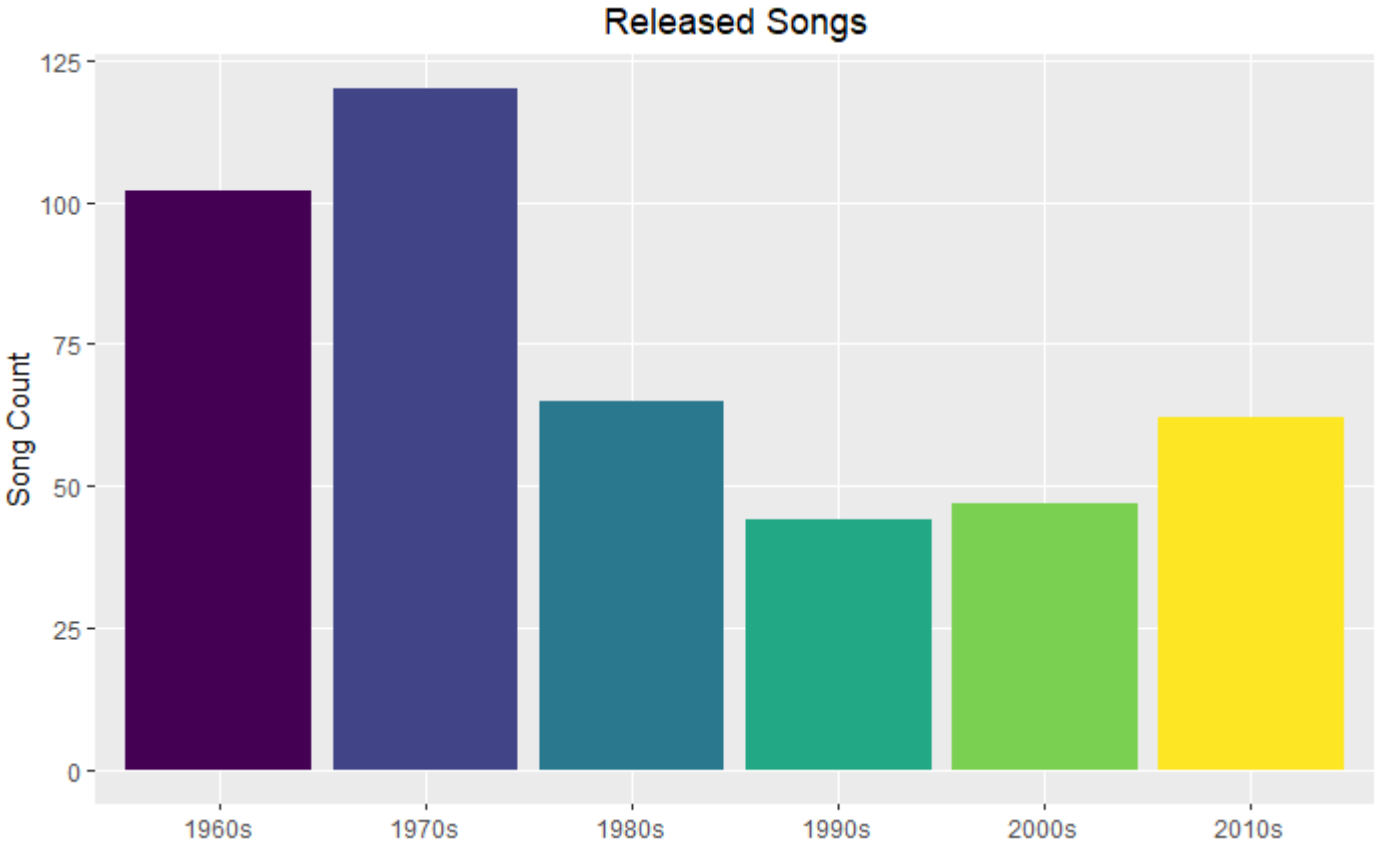
Hide

```
library(ggplot2)
```

```
There were 38 warnings (use warnings() to see them)
```

Hide

```
#Released Songs per decade
bob %>%
  filter(decade != "NA") %>%
  group_by(decade) %>%
  summarise(number_of_songs = n()) %>%
  ggplot() +
  geom_bar(aes(x = decade, y = number_of_songs), stat="identity", fill=viridis(6))  +
  theme(plot.title = element_text(hjust = 0.5),
        legend.title = element_blank(),
        panel.grid.minor = element_blank()) +
  ggtitle("Released Songs") +
  labs(x = NULL, y = "Song Count")
```



Hide

```
undesirable_words <- c("huh", "chorus", "lyrics",
                       "theres", "bridge", "fe0f", "yeah", "baby",
                       "alright", "wanna", "gonna", "chorus", "verse",
                       "whoa", "gotta", "make", "pum", "2",
                       "4", "ooh", "uurh", "pheromone", "poompoom",
                       "matic", " ai ", " ca ", " la ", "hey", " na ",
                       " da ", " uh ", " tin ", "  ll")

# Unnest and remove stop words
bob_words_filtered <- bob %>%
  unnest_tokens(word, lyric) %>%
  anti_join(stop_words) %>%
  distinct() %>%
  filter(!word %in% undesirable_words) %>%
  filter(nchar(word) > 3)
```

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

Hide

```
# TOP 10 SONGS WITH HIGHEST WORD COUNT
full_word_count <- bob %>%
  unnest_tokens(word, lyric) %>%
  group_by(track_title, album) %>%
  summarise(num_words = n()) %>%
  arrange(desc(num_words))

full_word_count[1:10,] %>%
  ungroup(num_words, track_title) %>%
  mutate(num_words = color_bar("lightblue")(num_words)) %>%
  mutate(song = color_tile("lightpink","lightpink")(track_title)) %>%
  kable("html", escape = FALSE, align = "c", caption = "Songs With Highest Word Count") %>%
  kable_styling(bootstrap_options = c("striped", "condensed", "bordered"), full_width = FALSE)
```

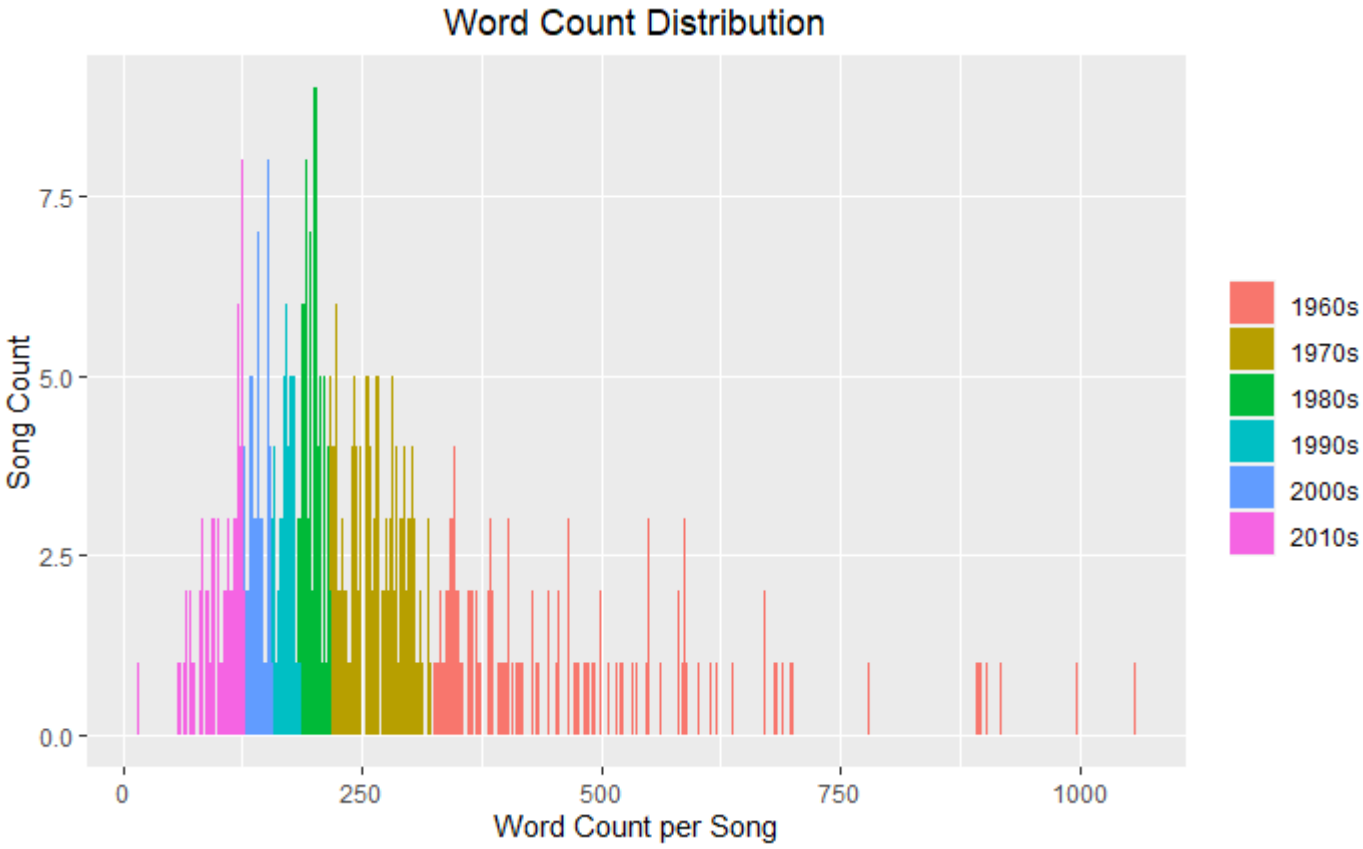
```
si 玫蛎 prodotto un NA per coercizione
```

Songs With Highest Word Count

track_title	album	num_words	song
Brownsville Girl	Knocked out loaded	1058	Brownsville Girl
Tin Angel	Tempest	998	Tin Angel
Tempest	Tempest	918	Tempest
Lily, Rosemary and the Jack of Hearts	Blood on the Tracks	904	Lily, Rosemary and the Jack of Hearts
Hurricane	Desire	896	Hurricane
Highlands	Time out of mind	895	Highlands
Bob Dylan's 115th Dream	Bringing It All Back Home	781	Bob Dylan's 115th Dream
Narrow Way	Tempest	700	Narrow Way
Froggie Went A-Courtin'	Good as i been to you	690	Froggie Went A-Courtin'
It's Alright, Ma (I'm Only Bleeding)	Bringing It All Back Home	683	It's Alright, Ma (I'm Only Bleeding)

Hide

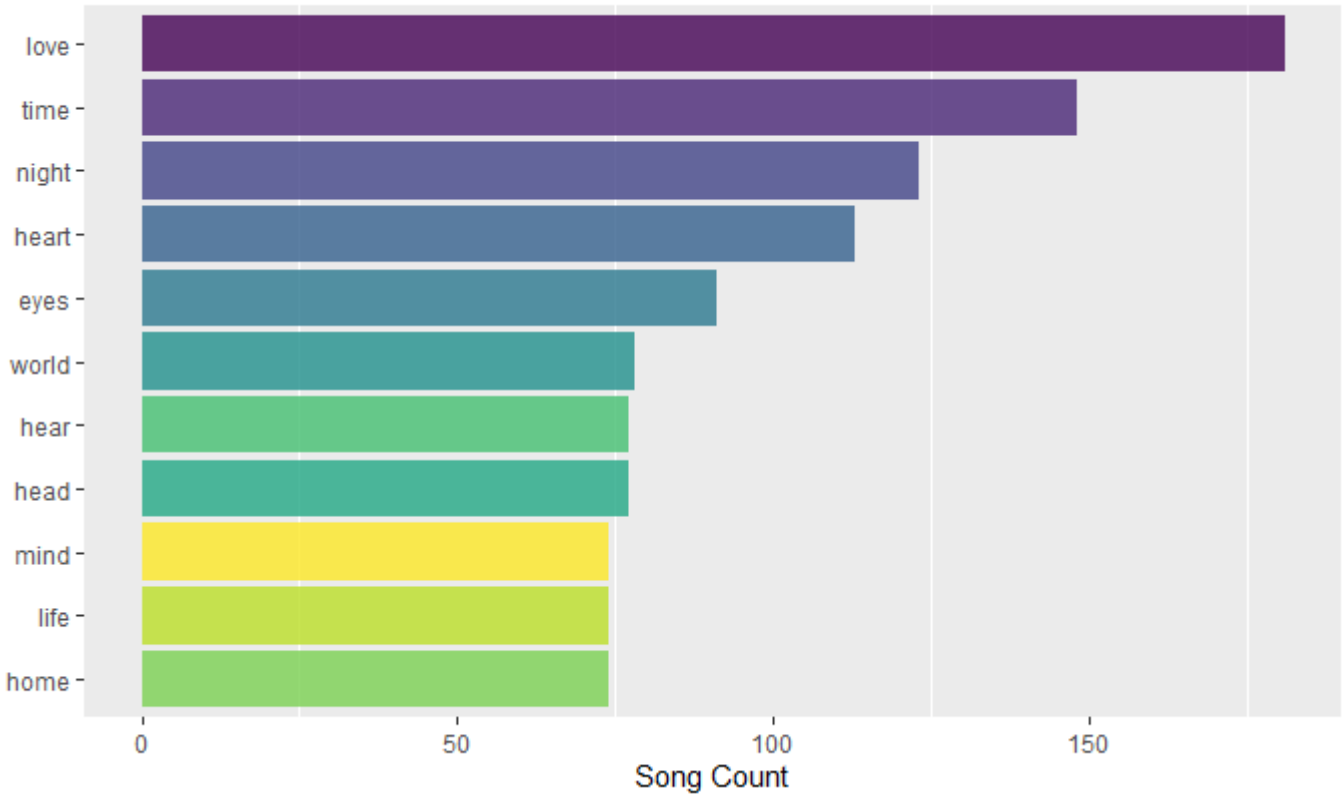
```
# WORD COUNT DISTRIBUTION
full_word_count %>%
  ggplot() +
  geom_histogram(aes(x = num_words, fill = bob$decade), bins = nrow(bob)) +
  ylab("Song Count") +
  xlab("Word Count per Song") +
  ggtitle("Word Count Distribution") +
  theme(plot.title = element_text(hjust = 0.5),legend.title = element_blank(), panel.grid.minor.y = ele
ment_blank())
```



Hide

```
bob_words_filtered %>%
  count(word, sort = TRUE) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot() +
  geom_col(aes(word, n), fill = viridis(11, alpha = 0.8)) +
  theme(legend.position = "none",
        plot.title = element_text(hjust = 0.5),
        panel.grid.major = element_blank()) +
  xlab("") +
  ylab("Song Count") +
  ggtitle("Most Frequently Used Words in Bob Dylan Lyrics") +
  coord_flip()
```

```
Selecting by n
```



Hide

```
# WORD CLOUD
bob_words_counts <- bob_words_filtered %>%
  count(word, sort = TRUE)

wordcloud2(bob_words_counts[1:300, ], size = .5)
```

Hide

```
library(tidyr)
```

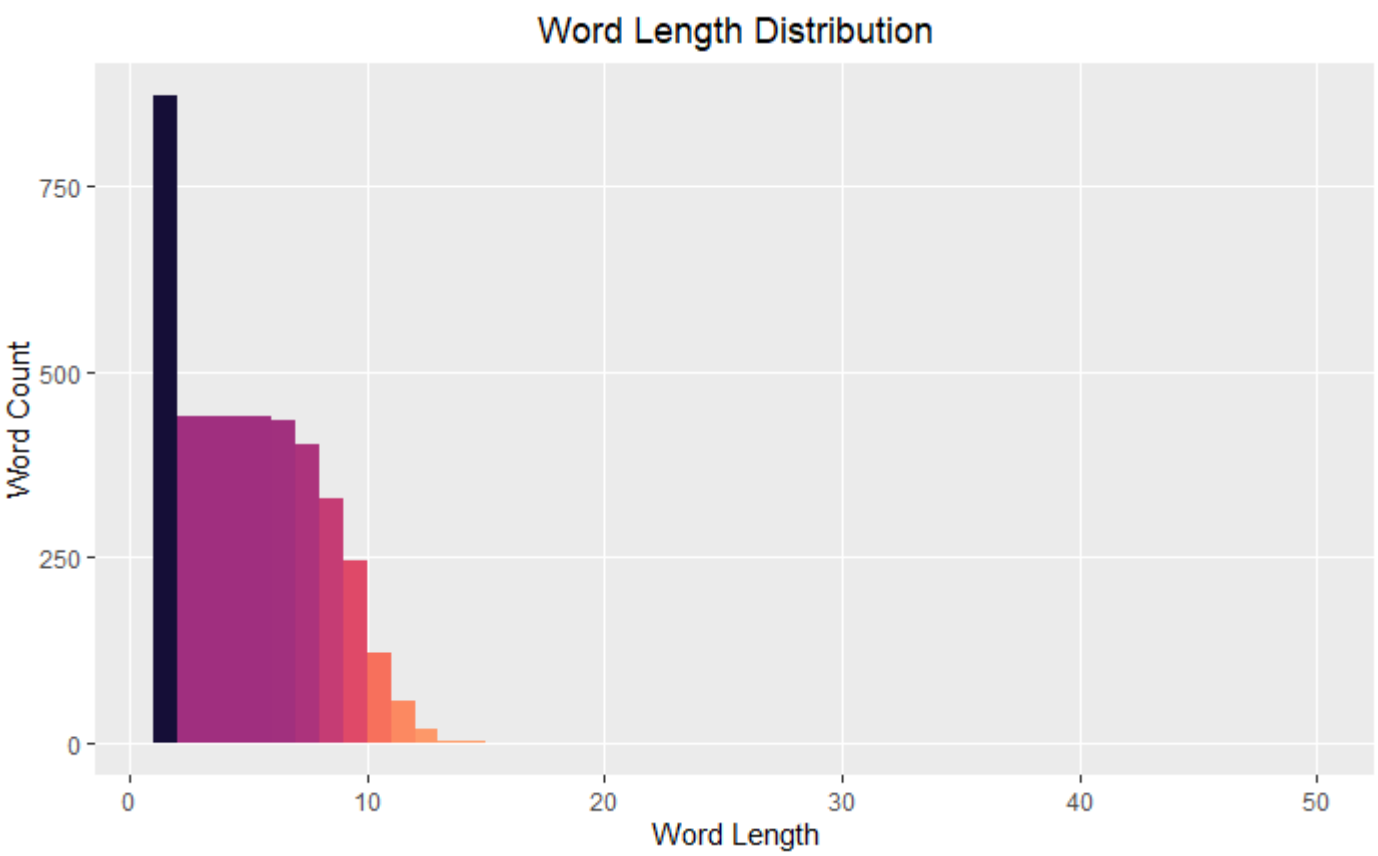
```
There were 50 or more warnings (use warnings() to see the first 50)
```

Hide

```
theme_lyrics <- function()
{
  theme(plot.title = element_text(hjust = 0.5),
        axis.text.x = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.position = "none")
}
#TTIMELESS WORDS: words over decades

timeless_words <- bob_words_filtered %>%
  filter(decade != 'NA') %>%
  group_by(decade) %>%
  count(word, decade, sort = TRUE) %>%
  slice(seq_len(8)) %>%
  ungroup() %>%
  arrange(decade,n) %>%
  mutate(row = row_number())

timeless_words %>%
  ggplot(aes(row, n, fill = decade)) +
  geom_col(show.legend = NULL) +
  labs(x = NULL, y = "Song Count") +
  ggtitle("Timeless Words") +
  theme_lyrics() +
  facet_wrap(~decade, scales = "free", ncol = 6) +
  scale_x_continuous(
    breaks = timeless_words$row,
    labels = timeless_words$word) +
  coord_flip()
```

Hide

```
# WORD LENGTH
bob_word_lengths <- bob %>%
  unnest_tokens(word, lyric) %>%
  group_by(track_title,decade) %>%
  distinct() %>%
  filter(!word %in% undesirable_words) %>%
  mutate(word_length = nchar(word))

bob_word_lengths %>%
  count(word_length, sort = TRUE) %>%
  ggplot(aes(word_length), binwidth = 0.5) +
  geom_histogram(aes(fill = ..count..), breaks = seq(1,50, by = 1), show.legend = FALSE, na.rm = TRUE)
 +
  xlab("Word Length") +
  ylab("Word Count") +
  ggtitle("Word Length Distribution") +
  theme(plot.title = element_text(hjust = 0.5),
        panel.grid.minor = element_blank())+
  scale_fill_viridis(option="magma" , begin = 0.1, end = 0.8, direction = -1)
```



Hide

```
NA
NA
NA
```

Hide

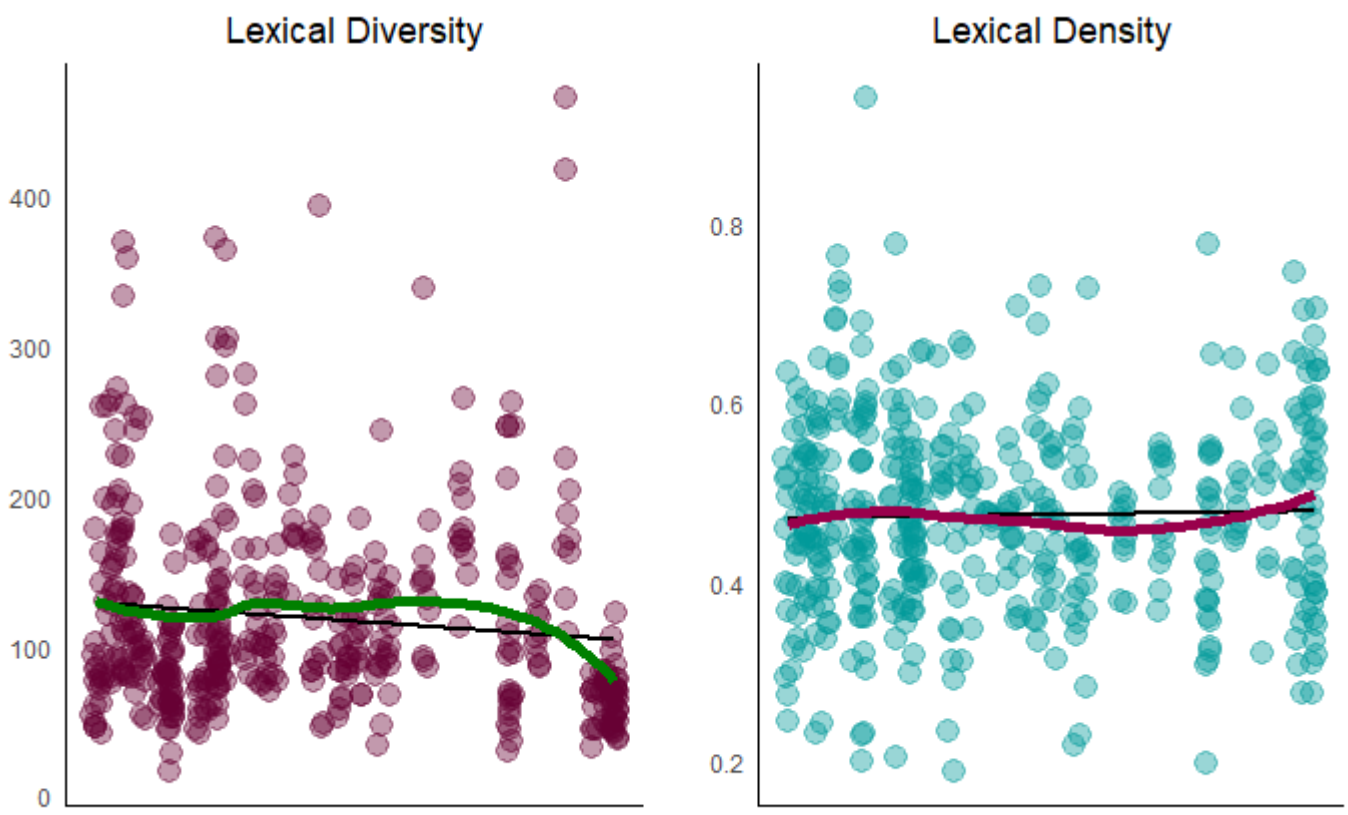
```
wc <- bob_word_lengths %>%
  ungroup() %>%
  select(word, word_length) %>%
  distinct() %>%
  arrange(desc(word_length))
```

Hide

```
wordcloud2(wc[1:300, ],
           size = .15,
           #minSize = .0005,
           #ellipticity = .3,
           #rotateRatio = 1,
           fontWeight = "bold",
           )
```

Hide

```
#LEXICAL DENSITY AND DIVERSITY
lex_diversity_per_year <- bob %>%
  filter(decade != "NA") %>%
  unnest_tokens(word, lyric) %>%
  group_by(track_title,year) %>%
  summarise(lex_diversity = n_distinct(word)) %>%
  arrange(desc(lex_diversity))

diversity_plot <- lex_diversity_per_year %>%
  ggplot(aes(year, lex_diversity)) +
  geom_point(color = "#660033",
             alpha = .4,
             size = 4,
             position = "jitter") +
  stat_smooth(color = "black", se = FALSE, method = "lm") +
  geom_smooth(aes(x = year, y = lex_diversity), se = FALSE,
              color = "#008000", lwd = 2) +
  ggtitle("Lexical Diversity") +
  xlab("") +
  ylab("") +
  scale_color_manual(values = my_colors) +
  theme_classic() +
  theme_lyrics()

# Over the last decades there was a clear decrease in Dylan's lyric diversity

# Lexical density defined as the number of unique words divided by the total number of words which indi
cates the evtual term repetition acros songs in this case stop words are included
lex_density_per_year <- bob %>%
  filter(decade != "NA") %>%
  unnest_tokens(word, lyric) %>%
  group_by(track_title,year) %>%
  summarise(lex_density = n_distinct(word)/n()) %>%
  arrange(desc(lex_density))

density_plot <- lex_density_per_year %>%
  ggplot(aes(year, lex_density)) +
  geom_point(color = "#009999",
             alpha = .4,
             size = 4,
             position = "jitter") +
  stat_smooth(color = "black",
              se = FALSE,
              method = "lm") +
  geom_smooth(aes(x = year, y = lex_density),
              se = FALSE,
              color = "#99004C",
              lwd = 2) +
  ggtitle("Lexical Density") +
  xlab("") +
  ylab("") +
  scale_color_manual(values = my_colors) +
  theme_classic() +
  theme_lyrics()


grid.arrange(diversity_plot, density_plot, ncol=2)
```



Hide

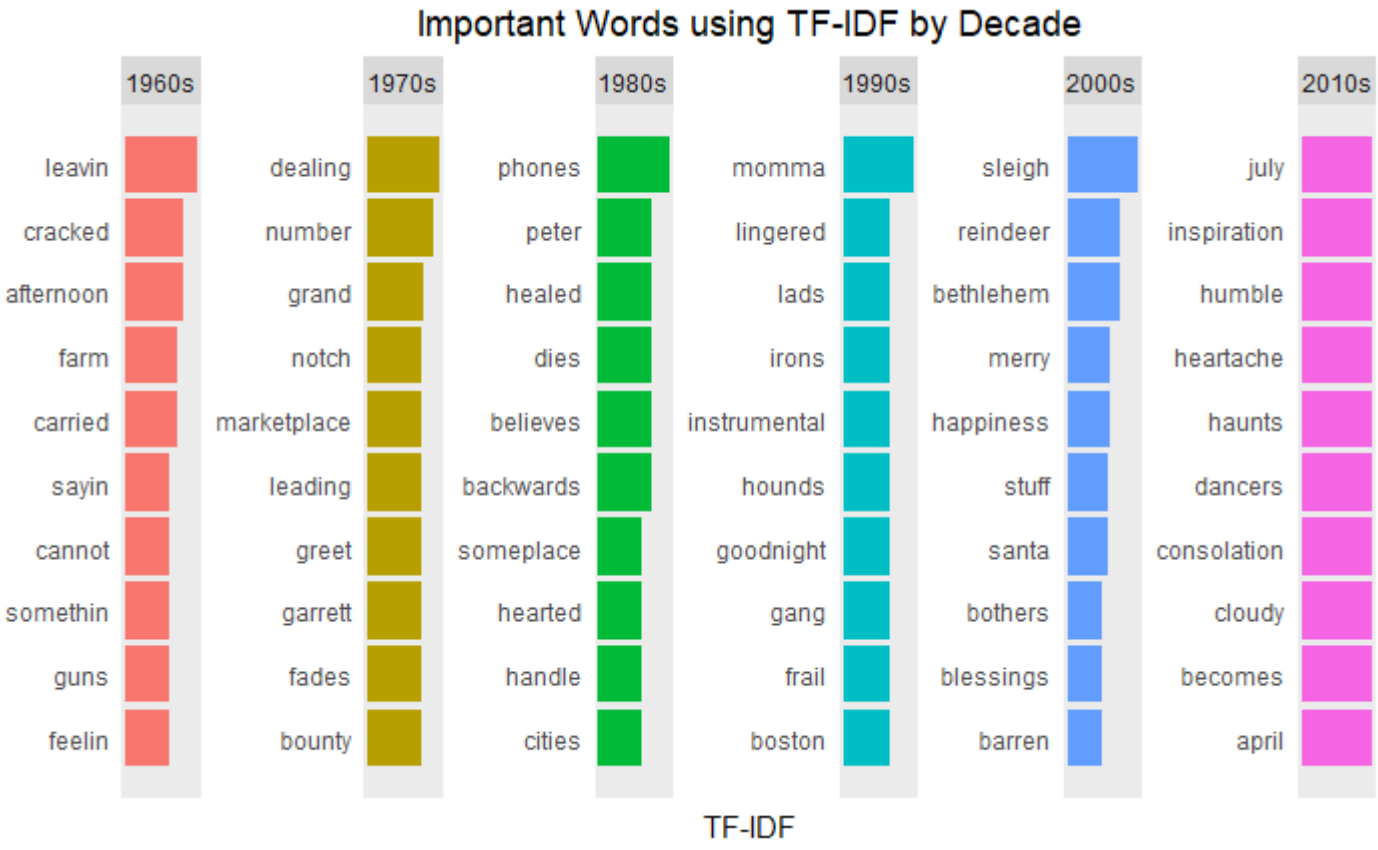
```
popular_tfidf_words <- bob %>%
  unnest_tokens(word, lyric) %>%
  distinct() %>%
  filter(!word %in% undesirable_words) %>%
  filter(nchar(word) > 3) %>%
  count(decade, word, sort = TRUE) %>%
  ungroup() %>%
  bind_tf_idf(word, decade, n)

head(popular_tfidf_words)
```

decade	word	n	tf	idf	tf_idf
<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>
1970s	that	86	0.008856849	0	0
1960s	that	83	0.008770076	0	0
1970s	with	72	0.007415036	0	0
1970s	have	71	0.007312049	0	0
1970s	will	71	0.007312049	0	0
1970s	your	70	0.007209063	0	0

6 rows

Hide

```
top_popular_tfidf_words <- popular_tfidf_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(decade) %>%
  slice(seq_len(10)) %>%
  ungroup() %>%
  arrange(decade, tf_idf) %>%
  mutate(row = row_number())

top_popular_tfidf_words %>%
  ggplot(aes(x = row, tf_idf, fill = decade)) +
  geom_col(show.legend = NULL) +
  labs(x = NULL, y = "TF-IDF") +
  ggtitle("Important Words using TF-IDF by Decade") +
  theme_lyrics() +
  facet_wrap(~decade, ncol = 6, scales = "free") +
  scale_x_continuous(  # This handles replacement of row
    breaks = top_popular_tfidf_words$row, # notice need to reuse data frame
    labels = top_popular_tfidf_words$word) +
  coord_flip()
```



Hide

```
tfidf_words_decade <- bob %>%
  unnest_tokens(word, lyric) %>%
  distinct() %>%
  filter(nchar(word) > 3) %>%
  count(decade, word, sort = TRUE) %>%
  ungroup() %>%
  bind_tf_idf(word, decade, n) %>%
  arrange(desc(tf_idf))
head(tfidf_words_decade)
```

decade	word	n	tf	idf	tf_idf
<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>

decade	word	n	tf	idf	tf_idf
<chr>	<chr>	<int>	<dbl>	<dbl>	<dbl>
2000s	sleigh	4	0.0008944544	1.791759	0.0016026471
1990s	momma	3	0.0008097166	1.791759	0.0014508174
1980s	phones	4	0.0007171029	1.791759	0.0012848759
2000s	bethlehem	3	0.0006708408	1.791759	0.0012019853
2000s	reindeer	3	0.0006708408	1.791759	0.0012019853
2000s	happiness	4	0.0008944544	1.098612	0.0009826586

6 rows

Hide

```
wc <- tfidf_words_decade %>%
  arrange(desc(tf_idf)) %>%
  select(word, tf_idf)
```

Hide

```
wordcloud2(wc[1:300, ],
           size = .15,
           minSize = .0005,
           #ellipticity = .3,
           #rotateRatio = 1,
           fontWeight = "bold",
)
```

Hide

```
### ZIPFS LAW
# By decade
# Word frequencies
song_words <- bob %>%
  unnest_tokens(words, lyric) %>%
  count(decade, words, sort = TRUE)
total_words <- song_words %>%
  group_by(decade) %>%
  summarize(total = sum(n))
song_words <- left_join(song_words, total_words)
```

```
Joining, by = "decade"
```

Hide

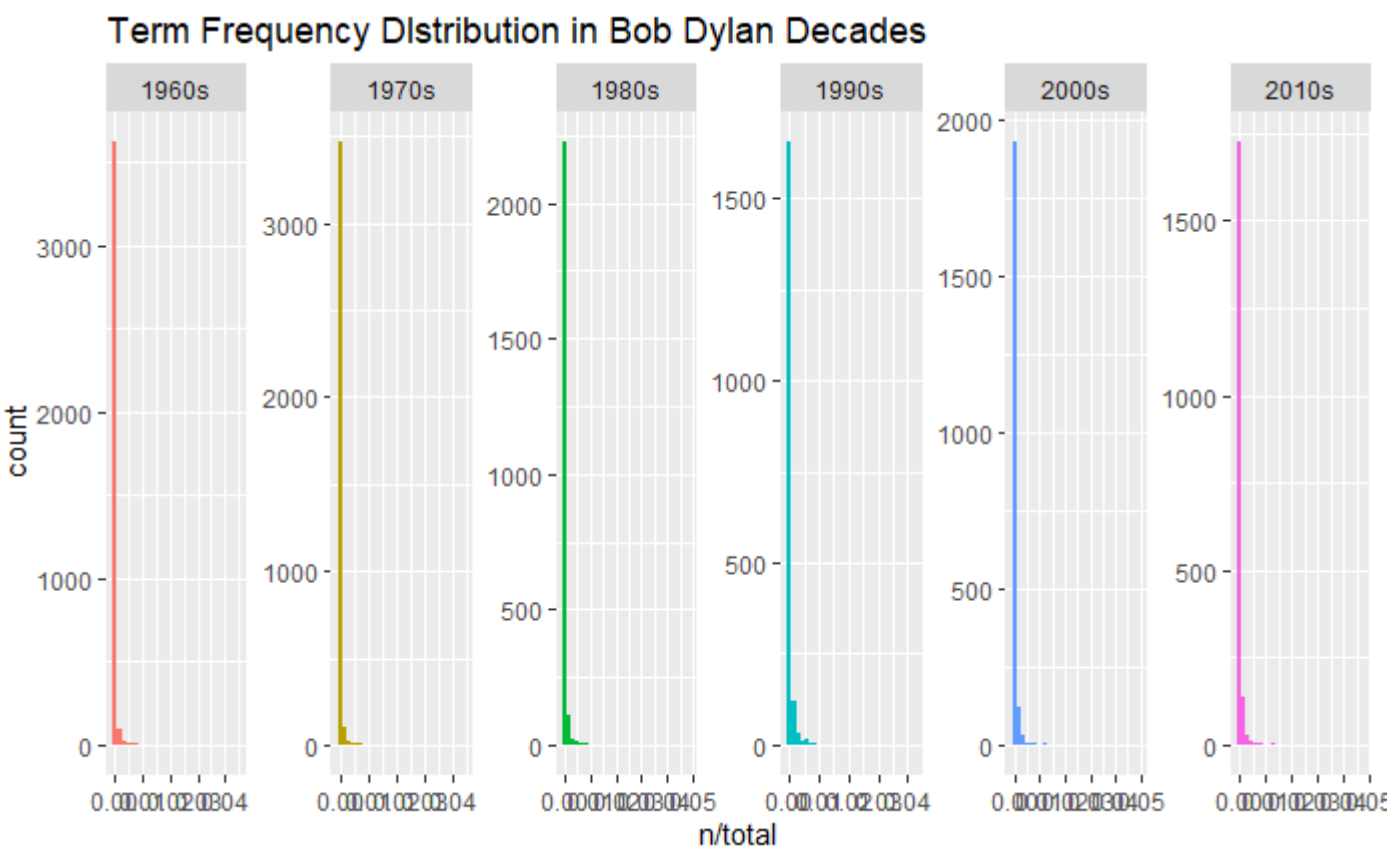
```
song_words
```

decade <chr>	words <chr>	n <int>	total <int>
1970s	the	1359	31105
1960s	the	1289	28610
1960s	i	1126	28610
1970s	you	1012	31105
1970s	i	978	31105
1970s	to	889	31105
1960s	you	855	28610
1980s	the	809	16707
1970s	and	799	31105
1960s	and	785	28610

1-10 of 15,736 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
library(ggplot2)

ggplot(song_words, aes(n/total, fill = decade)) +
  geom_histogram(show.legend = FALSE, bins = 30) +
  #xlim(NA, 0.0009) +
  facet_wrap(~decade, ncol = 6, scales = "free")+
  ggtitle("Term Frequency DIstribution in Bob Dylan Decades")
```



Hide

```
#Zipf's law states that the frequency that a word appears is inversely proportional to its rank.
freq_by_rank <- song_words %>%
  group_by(decade) %>%
  mutate(rank=row_number(), "term frequency"= n/total)
freq_by_rank
```

decade	words	n	total	rank	term frequency
<chr>	<chr>	<int>	<int>	<int>	<dbl>
1970s	the	1359	31105	1	0.0436907250
1960s	the	1289	28610	1	0.0450541769
1960s	i	1126	28610	2	0.0393568682
1970s	you	1012	31105	2	0.0325349622
1970s	i	978	31105	3	0.0314418904
1970s	to	889	31105	4	0.0285806140
1960s	you	855	28610	3	0.0298846557
1980s	the	809	16707	1	0.0484228168
1970s	and	799	31105	5	0.0256871886
1960s	and	785	28610	4	0.0274379588

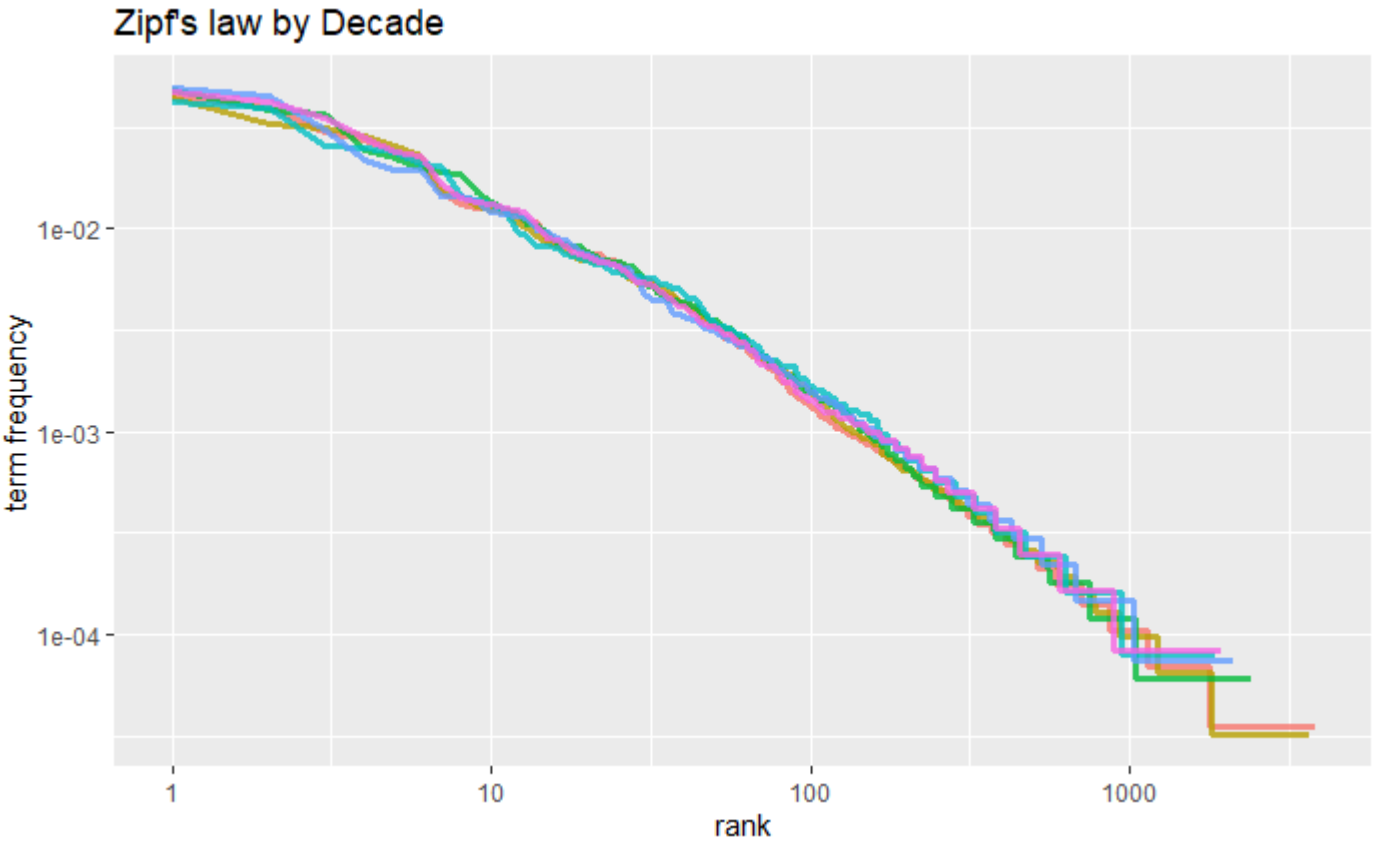
1-10 of 15,736 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
# rank column: rank of each word within the frequency table
```

Hide

```
freq_by_rank %>%
  ggplot(aes(rank, `term frequency`, color = decade)) +
  geom_line(size = 1.1, alpha = 0.8, show.legend = FALSE) +
  scale_x_log10() +
  scale_y_log10() +
  ggtitle("Zipf's law by Decade")
```



Hide

```
# decades similar to each other
```

Hide

```
#By album
song_words <- bob %>%
  unnest_tokens(words, lyric) %>%
  count(album, words, sort = TRUE)
total_words <- song_words %>%
  group_by(album) %>%
  summarize(total = sum(n))
song_words <- left_join(song_words, total_words)
```

```
Joining, by = "album"
```

Hide

```
song_words
```

album <chr>	words <chr>	n <int>	total <int>
Time out of mind	i	292	3744
Tempest	the	273	4785

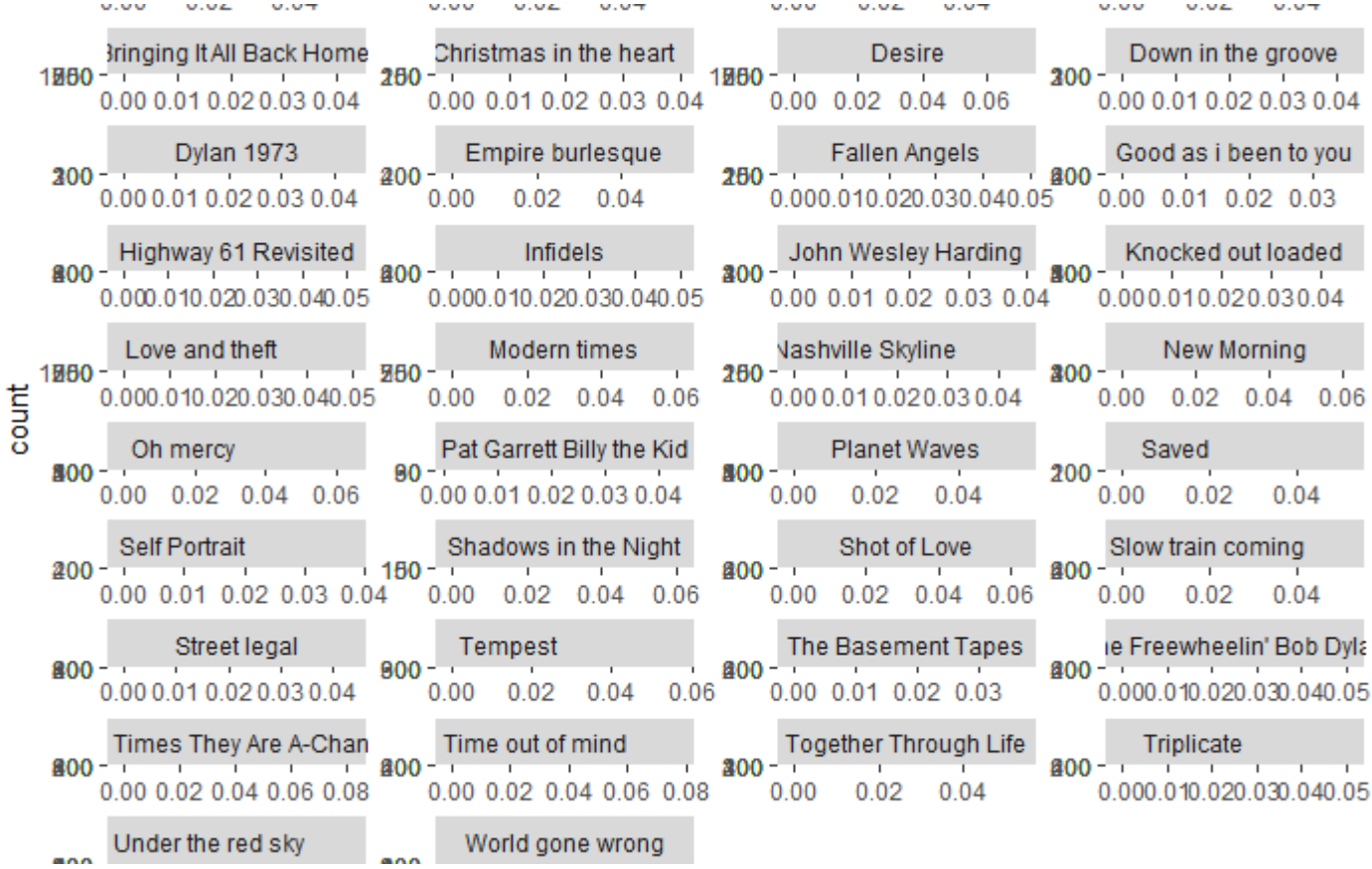
album <chr>	words <chr>	n <int>	total <int>
Desire\t	the	269	3801
Modern times	i	267	4384
The Times They Are A-Changin'	the	263	3215
Modern times	the	243	4384
Blonde on Blonde	you	241	4493
Love and theft	the	231	4616
Triplicate	i	213	4192
Blood on the Tracks	the	211	4006

1-10 of 29,207 rows Previous **1** 2 3 4 5 6 … 100 Next

Hide

```
ggplot(song_words, aes(n/total, fill = album)) +
  geom_histogram(show.legend = FALSE, bin = 30, alpha = 0.6) +
  #xlim(NA, 0.0009) +
  facet_wrap(~album, ncol = 4, scales = "free")
```

```
Ignoring unknown parameters: bin
```

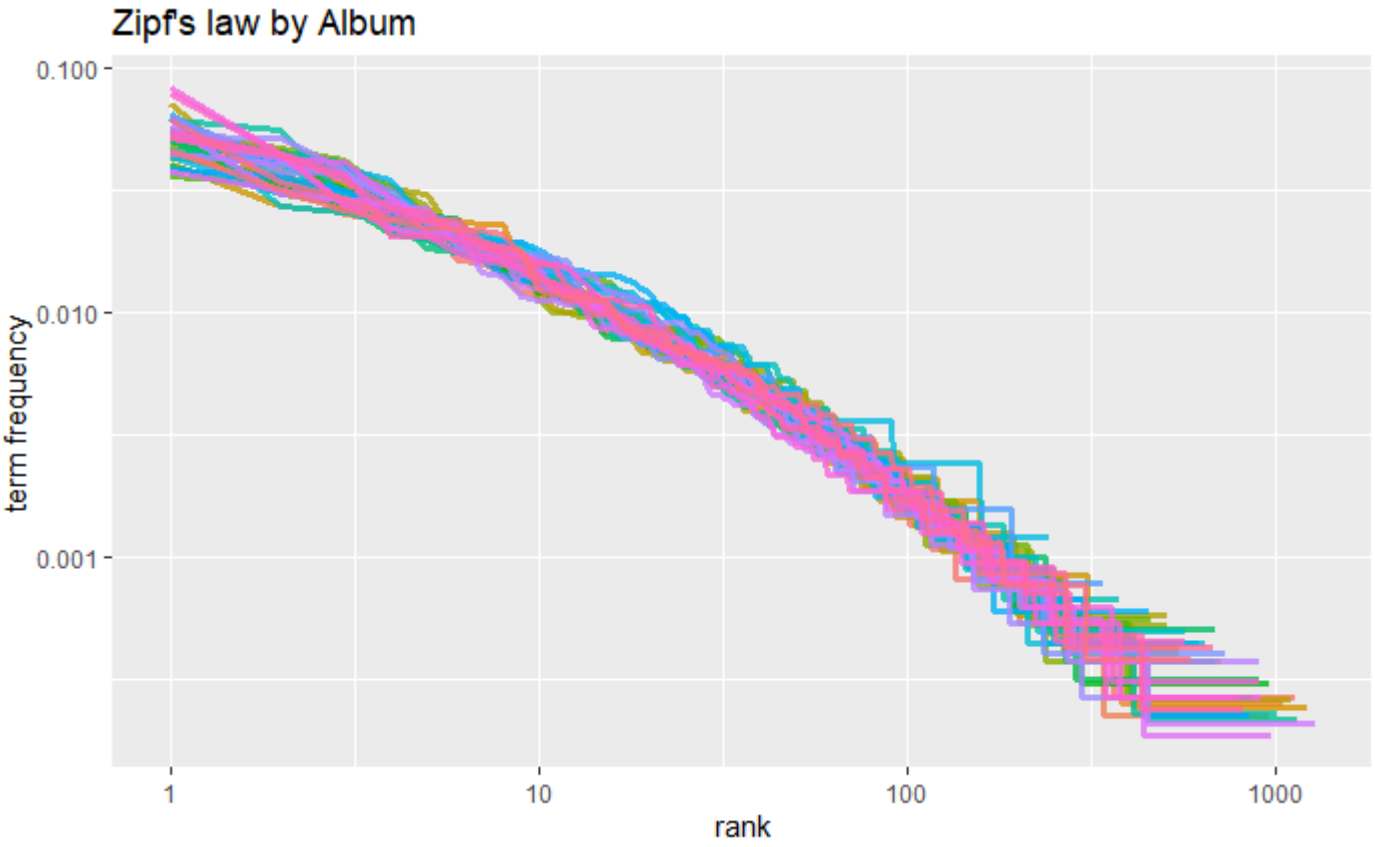


Hide

```
# zipf's law

freq_by_rank <- song_words %>%
  group_by(album) %>%
  mutate(rank=row_number(), "term frequency"= n/total)
freq_by_rank
```

album <chr>	words <chr>	n <int>	total <int>	rank <int>	term frequency <dbl>
Time out of mind	i	292	3744	1	0.077991453
Tempest	the	273	4785	1	0.057053292
Desire\t	the	269	3801	1	0.070770850
Modern times	i	267	4384	1	0.060903285
The Times They Are A-Changin'	the	263	3215	1	0.081804044
Modern times	the	243	4384	2	0.055428832
Blonde on Blonde	you	241	4493	1	0.053638994
Love and theft	the	231	4616	1	0.050043328
Triplicate	i	213	4192	1	0.050811069
Blood on the Tracks	the	211	4006	1	0.052670994

1-10 of 29,207 rows Previous **1** 2 3 4 5 6 … 100 Next



SENTIMENT ANALYSIS

Hide

```
#Customize the text tables for consistency using HTML formatting
my_kable_styling <- function(dat, caption) {
  kable(dat, "html", escape = FALSE, caption = caption) %>%
    kable_styling(bootstrap_options = c("striped", "condensed", "bordered"),
                  full_width = FALSE)
}

undesirable_words <- c("huh", "lyrics",  "bridge", "fe0f", "yeah", "baby",
                       "alright", "wanna", "gonna",
                       "whoa", "gotta", "make", "pum",
                       "ooh", "uurh", "pheromone", "poompoom",
                       "matic", " ai ", " ca ", " la ", "hey", " na ",
                       " da ", " uh ", " tin ", "  ll", "ooh", "uurh",
                       "repeats", "la", "da", "uh", "ah")

#Create tidy text format: Unnested, Unsummarized, -Undesirables, Stop and Short words
bob_tidy <- bob %>%
  unnest_tokens(word, lyric) %>% #Break the lyrics into individual words
  filter(!word %in% undesirable_words) %>% #Remove undesirables
  filter(!nchar(word) < 3) %>% #Words like "ah" or "oo" used in music
  anti_join(stop_words) #Data provided by the tidytext package
```

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

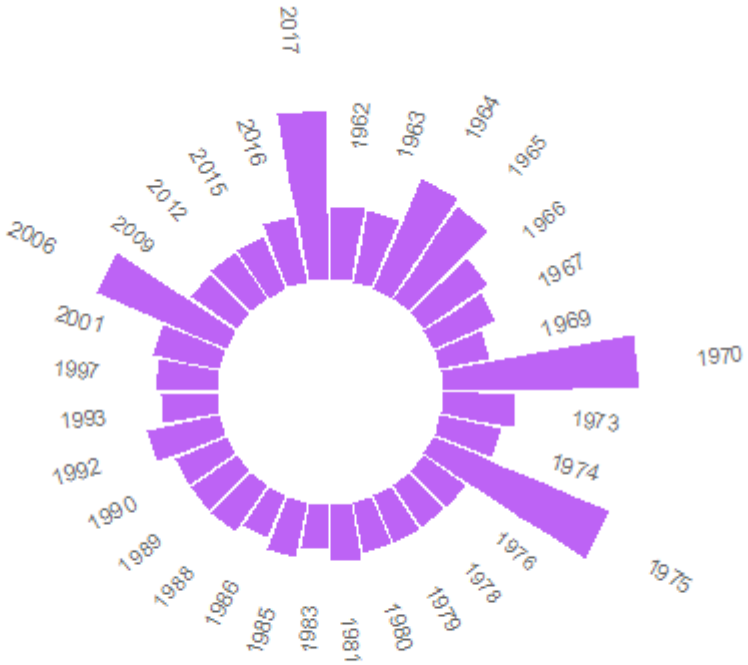
Hide

```
glimpse(bob_tidy)
```

```
Rows: 34,762
Columns: 6
$ track_title □[3m□[38;5;246m<chr>□[39m□[23m "You're No Good", "You're No Good", "You're No Good", "Yo
u're No Good", "You're No Good", "You're N...
$ track_n     □[3m□[38;5;246m<int>□[39m□[23m 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
$ album       □[3m□[38;5;246m<chr>□[39m□[23m "Bob Dylan                          ", "Bob Dylan
", "Bob Dylan             ...
$ year        □[3m□[38;5;246m<int>□[39m□[23m 1962, 1962, 1962, 1962, 1962, 1962, 1962, 1962, 1962, 196
2, 1962, 1962, 1962, 1962, 1962, 1962, 196...
$ decade      □[3m□[38;5;246m<chr>□[39m□[23m "1960s", "1960s", "1960s", "1960s", "1960s", "1960s", "19
60s", "1960s", "1960s", "1960s", "1960s", ...
$ word        □[3m□[38;5;246m<chr>□[39m□[23m "love", "world", "devil", "sleeping", "lion", "den", "hom
e", "night", "sweet", "crazy", "notion", "...
```

Hide

```
## SONG COUNT PER YEAR
songs_year <- bob %>%
  select(track_title, year) %>%
  group_by(year) %>%
  summarise(song_count = n())

id <- seq_len(nrow(songs_year))
songs_year <- cbind(songs_year, id)
label_data = songs_year
number_of_bar = nrow(label_data) #Calculate the ANGLE of the labels
angle = 90 - 360 * (label_data$id - 0.5) / number_of_bar #Center things
label_data$hjust <- ifelse(angle < -90, 1, 0) #Align label
label_data$angle <- ifelse(angle < -90, angle + 180, angle) #Flip angle
ggplot(songs_year, aes(x = as.factor(id), y = song_count)) +
  geom_bar(stat = "identity", fill = alpha("purple", 0.7)) +
  geom_text(data = label_data, aes(x = id, y = song_count + 10, label = year, hjust = hjust), color =
"black", alpha = 0.6,
            size = 3, angle =  label_data$angle, inherit.aes = FALSE ) +
  coord_polar(start = 0) +
  ylim(-20, 150) + #Size of the circle
  theme_minimal() +
  theme(axis.text = element_blank(),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        plot.margin = unit(rep(-4,4), "in"),
        plot.title = element_text(margin = margin(t = 10, b = -10)))
```



Hide

```
NA
NA
```