

# Lab Assignment 2: Billboard Hot 100 Songs

## Instructions

- Please answer these questions using code.
- The **only** R printouts should be the answers to the questions. Make sure your code does not display any extra information.
- You do not need to write any text for this assignment; only code.
- Submit your .html file to the assignment on PolyLearn.

## Introduction

Today, we will study song popularity. In the US, the Billboard Hot 100 is a list that comes out every week, showing the 100 most played songs that week. The following code will load a dataset of Billboard Hot 100 songs, as well as the tidyverse and lubridate packages. More information about the creation of this dataset, as well as some analyses by the author, can be found here: <https://mikekling.com/analyzing-the-billboard-hot-100/>. The dataset you are provided is a limited version of the full data, containing: - The title - The artist - The highest rank the song ever reached (1 is the best) - The number of weeks the song was on the chart - The latest date the song appeared on the Billboard Hot 100

```
library(tidyverse)
library(lubridate)
songs <- read.table("https://www.dropbox.com/s/jrwjthqo9b5o07g/billboard_songs.txt?dl=1", header = TRUE)
```

## Advice

This is a very large dataset! Consider testing out your steps on a smaller subset of songs - say, the first 100 on the list - before you run your code and try to figure out errors on the full 34,605 songs. `slice()` will help you make a smaller dataset.

## Questions

- 0) [Not graded, but useful!] Make your variables easier to work and/or adjust variable types.

*Hint: use `str_sub` to find the year, month, and day of the chart leaving date, then create a datetime object from these.*

```
songs <- songs %>%
  mutate(
    date = as.character(chart.date),
    year = str_sub(date, 0, 4),
    month = str_sub(date, 5, 6),
    day = str_sub(date, 7, 8),
    date = make_date(year = year, month = month, day = day)
  )
```

- 1) What 10 songs spent the longest on the charts? Give only the title, artist, and weeks.

```
songs %>%
  top_n(10, weeks.on.chart) %>%
```

```
arrange(desc(weeks.on.chart)) %>%
select(title, artist, weeks.on.chart)
```

```
##              title
## 1      Radioactive
## 2      I'm Yours
## 3      How Do I Live
## 4      Counting Stars
## 5      Party Rock Anthem
## 6 Foolish Games / You Were Meant For Me
## 7      Rolling In The Deep
## 8      Before He Cheats
## 9      Ho Hey
## 10     Need You Now
##              artist weeks.on.chart
## 1      IMAGINE DRAGONS           85
## 2      JASON MRAZ              76
## 3      LeANN RIMES             69
## 4      ONEREPUBLIC             68
## 5 LMFAO featuring LAUREN BENNETT & GOONROCK 67
## 6      JEWEL                  65
## 7      ADELE                  64
## 8      CARRIE UNDERWOOD       64
## 9      THE LUMINEERS          62
## 10     LADY ANTEBELLUM        60
```

2) What date did the oldest song(s) in this dataset leave the charts? Give only the date.

```
songs %>%
  top_n(1, desc(date)) %>%
  distinct(date)
```

```
##      date
## 1 1940-07-20
```

3) What songs could have been played at your 16th birthday party? That is, which songs overall peaked at #1 and **entered** the charts within a couple months (before or after) your 16th birthday? Give only the song title, artist, and date of chart entry.

```
my_bday <- mdy("April 2, 2005")

songs %>%
  mutate(
    entry_date = date - weeks(weeks.on.chart)
  ) %>%
  filter(overall.peak == 1 & entry_date %within% interval(my_bday - months(2), my_bday + months(2))) %>%
  select(title, artist, entry_date)
```

```
##      title      artist entry_date
## 1 We Belong Together MARIAH CAREY 2005-04-09
## 2      Hollaback GWEN STEFANI 2005-03-26
```

4) What is the most common word, at least 4 letters long, used in the title of any song? Give only the word itself, and its count.

```
titles <- songs %>%
  pull(title)
```

```
four_char_words <- titles %>%
  str_split(" ") %>%
  unlist() %>%
  str_extract("[:alpha:]{4,}")
```

```
four_char_words %>%
  table() %>%
  sort() %>%
  rev() %>%
  head(1)
```

```
## Love
## 2863
```

5) Which five artists were the most successful in terms of:

- Number of songs that were #1 hits
- Total weeks any of their songs spent on the chart.

*For this question, you may ignore collaborations; i.e., “Ariana Grande” is considered a different artist from “Jessie J, Ariana Grande & Nicki Minaj”*

*Hint: the function `n()` might help you summarize.*

```
songs %>%
  filter(overall_peak == 1) %>%
  group_by(artist) %>%
  summarize(num_songs = n()) %>%
  top_n(5, num_songs)
```

```
## # A tibble: 5 x 2
##   artist      num_songs
##   <chr>         <int>
## 1 ELVIS PRESLEY      13
## 2 MADONNA           12
## 3 MARIAH CAREY      16
## 4 MICHAEL JACKSON   11
## 5 THE BEATLES       21
```

```
songs %>%
  group_by(artist) %>%
  summarize(tot_weeks = sum(weeks.on.chart)) %>%
  top_n(5, tot_weeks)
```

```
## # A tibble: 5 x 2
##   artist      tot_weeks
##   <chr>         <int>
## 1 ELTON JOHN      930
## 2 ELVIS PRESLEY  1284
## 3 KENNY CHESNEY   673
## 4 MADONNA         870
## 5 PERRY COMO      718
```

6) Which artist has **featured** on the most Billboard charting songs? Which artist has **collaborated** on the most Billboard charting songs?

Definitions:

RAE SREMMURD featuring NICKI MINAJ & YOUNG THUG

In this string, Nicki Minaj and Young Thug are considered to be **featured**.

JESSIE J, ARIANA GRANDE & NICKI MINAJ

In this string, Jessie J and Ariana Grande and Nicki Minaj are all considered to have **collaborated** on the song.

```
artist_creds <- songs %>% pull(artist)

feat_artists <- artist_creds %>%
  str_subset("featuring") %>%
  str_replace_all(".* featuring", "") %>%
  str_extract_all("[^,&]+") %>%
  unlist() %>%
  str_trim(side = "both")

feat_artists %>%
  table() %>%
  sort() %>%
  rev() %>%
  head(1)
```

```
## LIL WAYNE
##          68
```

```
collab_artists <- artist_creds %>%
  str_replace_all("featuring .*", "") %>%
  str_extract_all("[^,&]+") %>%
  unlist() %>%
  str_trim(side = "both")

collab_artists %>%
  table() %>%
  sort() %>%
  rev() %>%
  head(1)
```

```
## GLEE CAST
##          204
```