Question 2 Wrangling the Billboard Top 100

Wrangling the Billboard Top 100

** PART A **

Part A: Make a table of the top 10 most popular songs since 1958, as measured by the total number of weeks that a song spent on the Billboard Top 100. Note that these data end in week 22 of 2021, so the most popular songs of 2021 will not have up-to-the-minute data; please send our apologies to The Weeknd.

Your table should have 10 rows and 3 columns: performer, song, and count, where count represents the number of weeks that song appeared in the Billboard Top 100. Make sure the entries are sorted in descending order of the count variable, so that the more popular songs appear at the top of the table. Give your table a short caption describing what is shown in the table.

(Note: you'll want to use both performer and song in any group_by operations, to account for the fact that multiple unique songs can share the same title.)

```
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)

billboard <- read.csv("/Users/jagrutaadvani/Downloads/billboard.csv")

# Preview the dataframe
head(billboard)
```

	X	url	week_id	week_position
	<int>	<chr>	<chr>	<int>
1	1	http://www.billboard.com/charts/hot-100/1965-07-17	7/17/1965	34
2	2	http://www.billboard.com/charts/hot-100/1965-07-24	7/24/1965	22
3	3	http://www.billboard.com/charts/hot-100/1965-07-31	7/31/1965	14

	X	url	week_id	week_position
	<int>	<chr>	<chr>	<int>
4	4	http://www.billboard.com/charts/hot-100/1965-08-07	8/7/1965	10
5	5	http://www.billboard.com/charts/hot-100/1965-08-14	8/14/1965	8
6	6	http://www.billboard.com/charts/hot-100/1965-08-21	8/21/1965	8

6 rows | 1-5 of 14 columns

```
summary(billboard)
```

```
##        X              url              week_id          week_position
##  Min.   :     1   Length:327895      Length:327895      Min.   :  1.0
##  1st Qu.: 81974   Class :character   Class :character   1st Qu.: 25.5
##  Median :163948   Mode  :character   Mode  :character   Median : 50.0
##  Mean   :163948                                        Mean   : 50.5
##  3rd Qu.:245922                                        3rd Qu.: 75.0
##  Max.   :327895                                        Max.   :100.0
##
##      song            performer           song_id            instance
##  Length:327895      Length:327895      Length:327895      Min.   : 1.000
##  Class :character   Class :character   Class :character   1st Qu.: 1.000
##  Mode  :character   Mode  :character   Mode  :character   Median : 1.000
##                                                           Mean   : 1.073
##                                                           3rd Qu.: 1.000
##                                                           Max.   :10.000
##
##  previous_week_position peak_position    weeks_on_chart        year
##  Min.   :  1.0          Min.   :  1.00   Min.   : 1.000   Min.   :1958
##  1st Qu.: 23.0          1st Qu.: 14.00   1st Qu.: 4.000   1st Qu.:1974
##  Median : 47.0          Median : 39.00   Median : 7.000   Median :1989
##  Mean   : 47.6          Mean   : 41.36   Mean   : 9.154   Mean   :1989
##  3rd Qu.: 72.0          3rd Qu.: 66.00   3rd Qu.:13.000   3rd Qu.:2005
##  Max.   :100.0          Max.   :100.00   Max.   :87.000   Max.   :2021
##  NA's   :31954
##       week
##  Min.   : 1.00
##  1st Qu.:14.00
##  Median :27.00
##  Mean   :26.59
##  3rd Qu.:40.00
##  Max.   :53.00
##
```

```
# Grouped billboard by song and performer, then counting number of weeks
top_songs <- billboard %>%
  group_by(performer, song) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
```

```
## `summarise()` has grouped output by 'performer'. You can override using the
## `.groups` argument.
```

```
top_10_songs <- head(top_songs, 10)

cat("Top 10 Most Popular Songs Since 1958, Measured by Total Weeks on the Billboard Top
100")
```

```
## Top 10 Most Popular Songs Since 1958, Measured by Total Weeks on the Billboard Top 10
0
```

```
print(top_10_songs)
```

```
## # A tibble: 10 × 3
## # Groups:   performer [10]
##    performer                                song                           count
##    <chr>                                    <chr>                          <int>
##  1 Imagine Dragons                          Radioactive                       87
##  2 AWOLNATION                               Sail                              79
##  3 Jason Mraz                               I'm Yours                         76
##  4 The Weeknd                               Blinding Lights                   76
##  5 LeAnn Rimes                              How Do I Live                     69
##  6 LMFAO Featuring Lauren Bennett & GoonRock Party Rock Anthem                68
##  7 OneRepublic                              Counting Stars                    68
##  8 Adele                                    Rolling In The Deep               65
##  9 Jewel                                    Foolish Games/You Were Meant…     65
## 10 Carrie Underwood                         Before He Cheats                  64
```

*** PART B ***

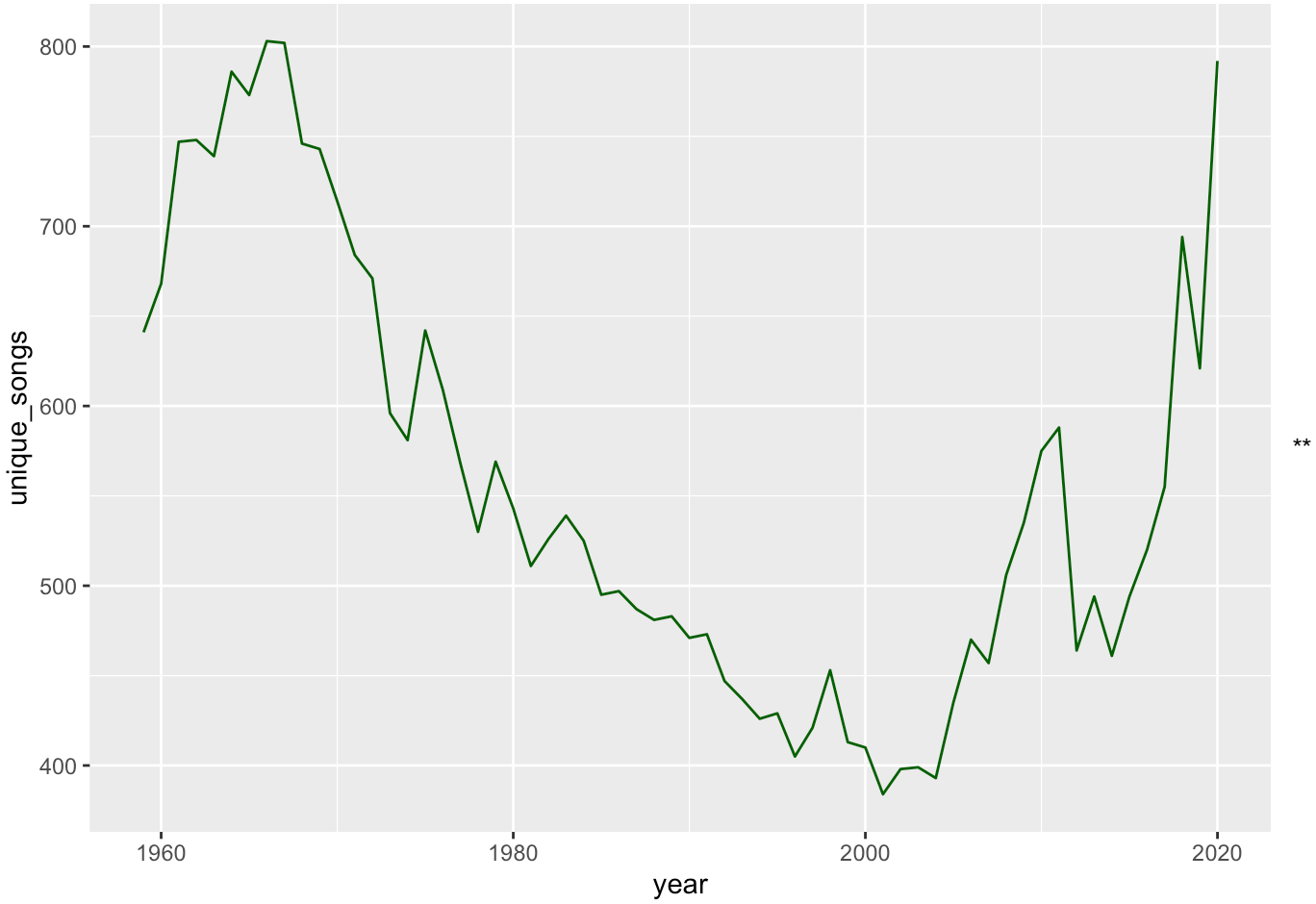
Part B: Is the “musical diversity” of the Billboard Top 100 changing over time? Let’s find out. We’ll measure the musical diversity of given year as the number of unique songs that appeared in the Billboard Top 100 that year. Make a line graph that plots this measure of musical diversity over the years. The x axis should show the year, while the y axis should show the number of unique songs appearing at any position on the Billboard Top 100 chart in any week that year. For this part, please filter the data set so that it excludes the years 1958 and 2021, since we do not have complete data on either of those years. Give the figure an informative caption in which you explain what is shown in the figure and comment on any interesting trends you see.

There are number of ways to accomplish the data wrangling here. For example, you could use two distinct sets of data-wrangling steps. The first set of steps would get you a table that counts the number of times that a given song appears on the Top 100 in a given year. The second set of steps operate on the result of the first set of steps; it would count the number of unique songs that appeared on the Top 100 in each year, irrespective of how many times it had appeared.

```
# Filter Data excluding 1958 and 2021
billboard_filter <- billboard %>%
  filter(year > 1958 & year < 2021)

# Group by year and count the number of unique songs
unique_songs_yearly <- billboard_filter %>%
  group_by(year) %>%
  summarize(unique_songs = n_distinct(song))

ggplot(unique_songs_yearly, aes(x = year, y = unique_songs)) +
  geom_line(color = "darkgreen")
```



**

INTERESTING TRENDS OBSERVED

*1980s to Early 2000s Decline: The number of unique songs on the Billboard Top 100 gradually decreased, hitting a low around 2001-2002. This period saw dominant artists like Michael Jackson and Madonna with fewer new entries each year.

*Post-2000s Increase: Musical diversity sharply increased after 2002, with the number of unique songs rising by over 50% by the late 2000s. This aligns with the digital music revolution and the rise of platforms like YouTube.

*2012-2013 Decline: A sharp decline in 2012-2013 saw the number of unique songs drop by approximately 20%, likely due to the influence of streaming algorithms favoring fewer, highly popular tracks.

*** PART C ***

Part C: Let's define a "ten-week hit" as a single song that appeared on the Billboard Top 100 for at least ten weeks. There are 19 artists in U.S. musical history since 1958 who have had at least 30 songs that were "ten-week hits." Make a bar plot for these 19 artists, showing how many ten-week hits each one had in their musical career. Give the plot an informative caption in which you explain what is shown.

Notes:

You might find this easier to accomplish in two distinct sets of data wrangling steps. Make sure that the individuals names of the artists are readable in your plot, and that they're not all jumbled together. If you find that your plot isn't readable with vertical bars, you can add a coord_flip() layer to your plot to make the bars (and labels) run horizontally instead. By default a bar plot will order the artists in alphabetical order. This is acceptable to turn in. But if you'd like to order them according to some other variable, you can use the fct_reorder function, described in this blog post.

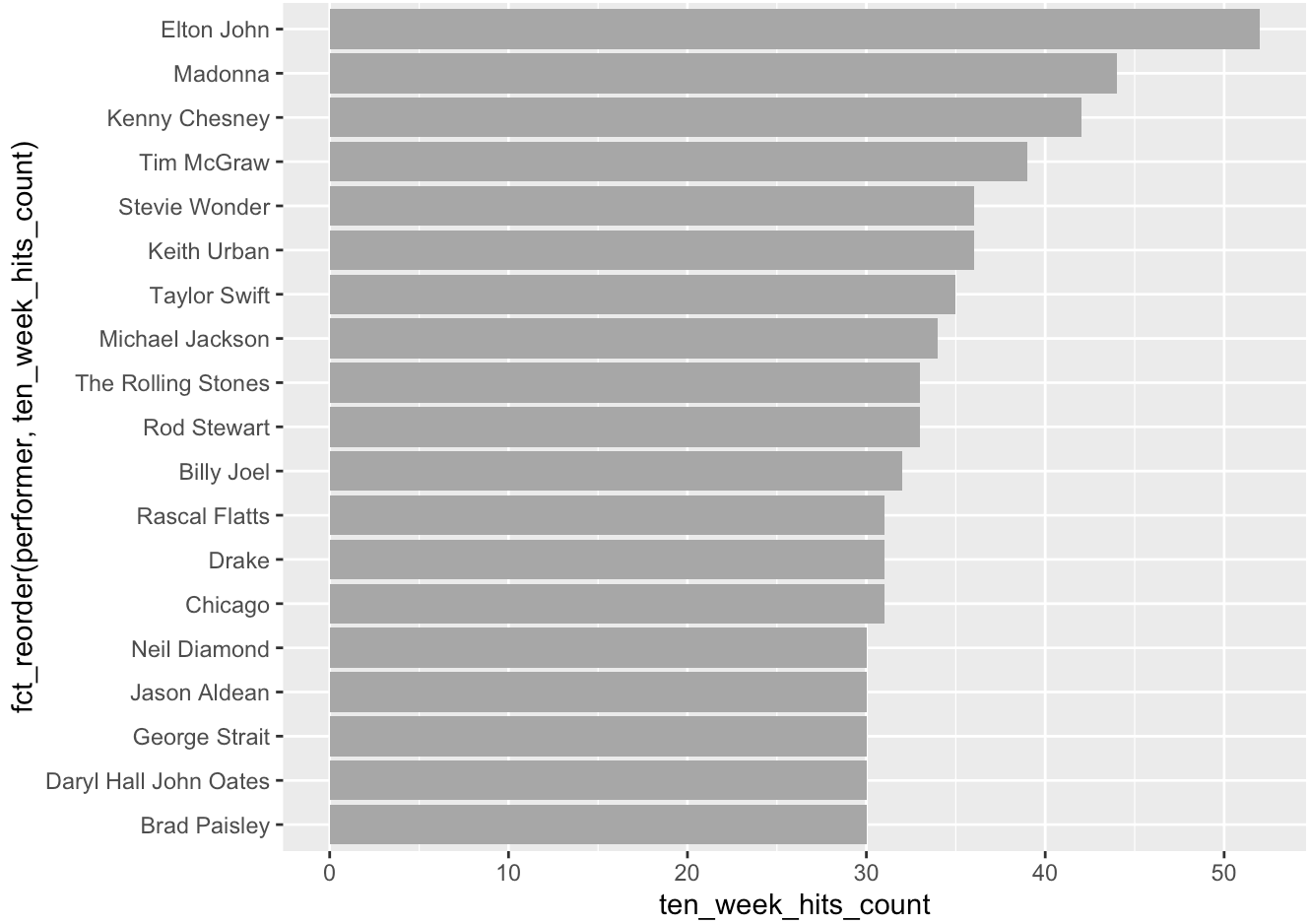
```
library(forcats)

ten_week_hits <- billboard %>%
  group_by(performer, song) %>%
  summarize(week_count = n()) %>%
  filter(week_count >= 10)
```

```
## `summarise()` has grouped output by 'performer'. You can override using the
## `.groups` argument.
```

```
artist_count <- ten_week_hits %>%
  group_by(performer) %>%
  summarize(ten_week_hits_count = n()) %>%
  filter(ten_week_hits_count >= 30)

ggplot(artist_count, aes(x = fct_reorder(performer, ten_week_hits_count), y = ten_week_h
its_count)) +
  geom_bar(stat = "identity", fill = "darkgrey") +
  coord_flip()
```


Elton John
Madonna
Kenny Chesney
Tim McGraw
Stevie Wonder
Keith Urban
Taylor Swift
Michael Jackson
The Rolling Stones
Rod Stewart
Billy Joel
Rascal Flatts
Drake
Chicago
Neil Diamond
Jason Aldean
George Strait
Daryl Hall John Oates
Brad Paisley
fct_reorder(performer, ten_week_hits_count)
0
10
20
30
40
50
ten_week_hits_count