

Billboard Top 100

2024-08-13

Wrangling the Billboard Top 100

Consider the data in `billboard.csv` containing every song to appear on the weekly Billboard Top 100 chart since 1958, up through the middle of 2021. Each row of this data corresponds to a single song in a single week. For our purposes, the relevant columns here are:

- *performer*: who performed the song
 - *song*: the title of the song
 - *year*: year (1958 to 2021)
 - *week*: chart week of that year (1, 2, etc)
 - *week_position*: what position that song occupied that week on the Billboard top 100 chart
- Use your skills in data wrangling and plotting to answer the following three questions

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.)

```
billboardTop <- billboardRelevant %>%
  filter(week_position <= 100, year >= 1958) %>%
  group_by(performer, song) %>%
  summarise(count = n(), .groups = 'drop') %>%
  arrange(desc(count))

tenthValue = unique(billboardTop$count)[10]

billboardTop10 <- billboardTop %>%
  filter(count >= tenthValue) %>%
  arrange(desc(count)) %>%
  head(10)

print(billboardTop10)
```

```
## # A tibble: 10 x 3
```

##	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

Table Caption:

The greatest count in the table is 87 and the lowest count is 60.

Other than Imagine Dragons, no other performer really repeats in the top 10 table.

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.

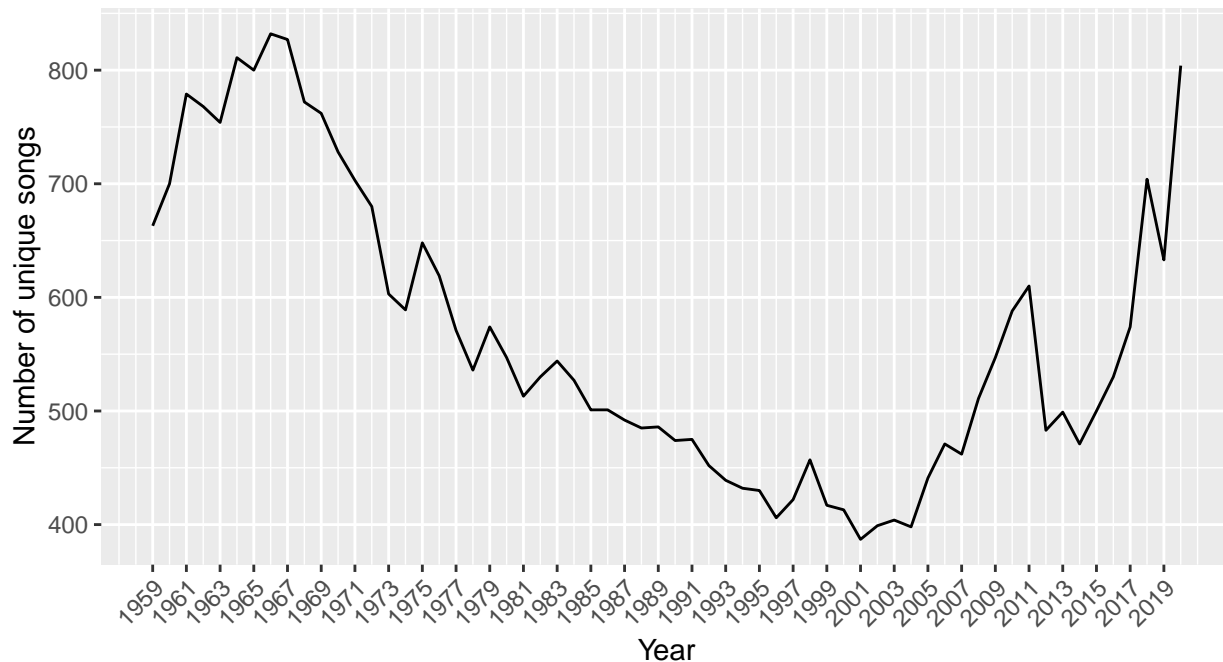
```
any_time_per_year = billboardRelevant %>%
  filter(week_position <= 100)

any_time_per_year = any_time_per_year %>%
  group_by(year) %>%
  summarise(countOfSongs = n_distinct(performer, song))

ggplot(data = any_time_per_year[any_time_per_year$year > 1958
                                & any_time_per_year$year < 2021, ]) +
  geom_line(aes(x = year, y = countOfSongs)) +
  scale_x_continuous(breaks = seq(1959, 2020, 2), guide = guide_axis(angle = 45)) +
  xlab(label = "Year") +
  ylab(label = "Number of unique songs") +
  labs(title = "Musical Diversity",
        subtitle = "Unique songs across the years on the Billboard Top 100 Music list",
        caption = "The diversity frequently changes starting with peaks in the 1950s into the 1960s
        before then heading to a steep and continuous decline during the 1990s before picking back
        up and reaching a 1960s level peak in the late 2010s.") +
  theme(plot.caption = element_text(hjust = 0.5))
```

Musical Diverity

Unique songs across the years on the Billboard Top 100 Music list



The diversity frequently changes starting with peaks in the 1950s into the 1960s before then heading to a steep and continuous decline during the 1990s before picking back up and reaching a 1960s level peak in the late 2010s.

Comments:

The analysis begins in 1958, where the graph shows a little over 650 unique songs. From there, there is a steady overall increase until reaching a significant peak between 1965 and 1967, with just under 850 unique songs. After this peak, the graph experiences a sharp and continuous decline, reaching its lowest point in 2001 with fewer than 400 unique songs. Following this low, the trend reverses, showing an increase (with some fluctuations) up to the final year on the graph, 2019, which also marks another peak with around 800 unique songs.

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.

```
tenWeekHit = billboardRelevant %>%
  filter(week_position <= 100) %>%
  group_by(performer, song) %>%
  summarise(songsWeeklyCount = n()) %>%
  filter(songsWeeklyCount >= 10) %>%
  group_by(performer) %>%
  summarise(songsCount = n()) %>%
  filter(songsCount >= 30)
tenWeekHit
```

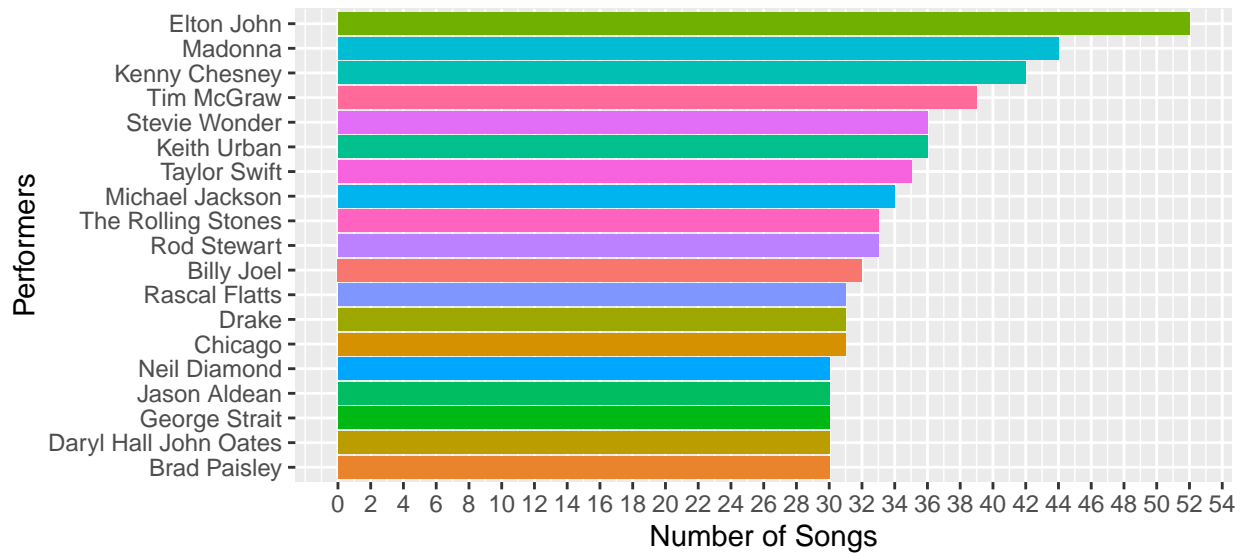
```
## # A tibble: 19 x 2
##   performer      songsCount
##   <chr>          <int>
## 1 Billy Joel      32
```

```
## 2 Brad Paisley          30
## 3 Chicago               31
## 4 Daryl Hall John Oates 30
## 5 Drake                 31
## 6 Elton John            52
## 7 George Strait         30
## 8 Jason Aldean          30
## 9 Keith Urban           36
## 10 Kenny Chesney        42
## 11 Madonna              44
## 12 Michael Jackson      34
## 13 Neil Diamond         30
## 14 Rascal Flatts        31
## 15 Rod Stewart          33
## 16 Stevie Wonder        36
## 17 Taylor Swift         35
## 18 The Rolling Stones   33
## 19 Tim McGraw           39
```

```
ggplot(data = tenWeekHit) +
  geom_col(aes(x = reorder(performer, songsCount), y = songsCount, fill=performer)) +
  coord_flip() +
  scale_y_continuous(breaks = seq(0, 60, 2)) +
  xlab(label = "Performers") +
  ylab(label = "Number of Songs") +
  labs(title = "Billboard Ten week hits",
        subtitle = " \n\n", # adds some space below the title so the entire legend can fit
        caption = "Artists with at least 30 songs that were 10-week hits on the Billboard Music list.
        Elton John takes the lead followed by Maddona, Kenny Chesney, and
        Tim McGraw.
        Greater variation at the top of the list but as you go down,
        the numbers are packed together.") +
  theme(plot.caption = element_text(hjust = 0.5)) +
  guides(fill = FALSE)
```

```
## Warning: The `scale` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Billboard Ten week hits



Artists with at least 30 songs that were 10-week hits on the Billboard Music list.
 Elton John takes the lead followed by Maddona, Kenny Chesney, and
 Tim McGraw.
 Greater variation at the top of the list but as you go down,
 the numbers are packed together.

Comments: See Graph Caption

There is greater variation at the top of the list but as you move down these is more standarization in that artists start to have the same number of songs that meet this criteria. We also see a wide variety of musical genres represented, with country being particularly prevalent.