ML - Probability Practice

Sarah Dominguez

2024-08-12

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.)

```
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
```

```r
library(readr)
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
billboard <- read_csv("billboard.csv")
```

```
## New names:
## Rows: 327895 Columns: 13
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (5): url, week_id, song, performer, song_id dbl (8): ...1, week_position,
## instance, previous_week_position, peak_positio...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
# top 10 songs by total number of weeks on the chart
top_songs <- billboard %>%
  group_by(performer, song) %>%              # Group by both performer and song
  summarise(count = n(), .groups = "drop") %>% # Count the number of weeks each song appeared
  arrange(desc(count)) %>%                   # Sort the songs by the count in descending order
  slice_max(order_by = count, n = 10)        # Select the top 10 songs

# Display the result
print(top_songs)
```

```
## # 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
```

```r
# needed caption
cat("Table: Top 10 most popular songs since 1958, ranked by the total number of weeks they appeared on t
```

```
## Table: Top 10 most popular songs since 1958, ranked by the total number of weeks they appeared on the
```

```r
####Might need to come back and check for the caption ^^^^^^
#########
########
##########
########
```

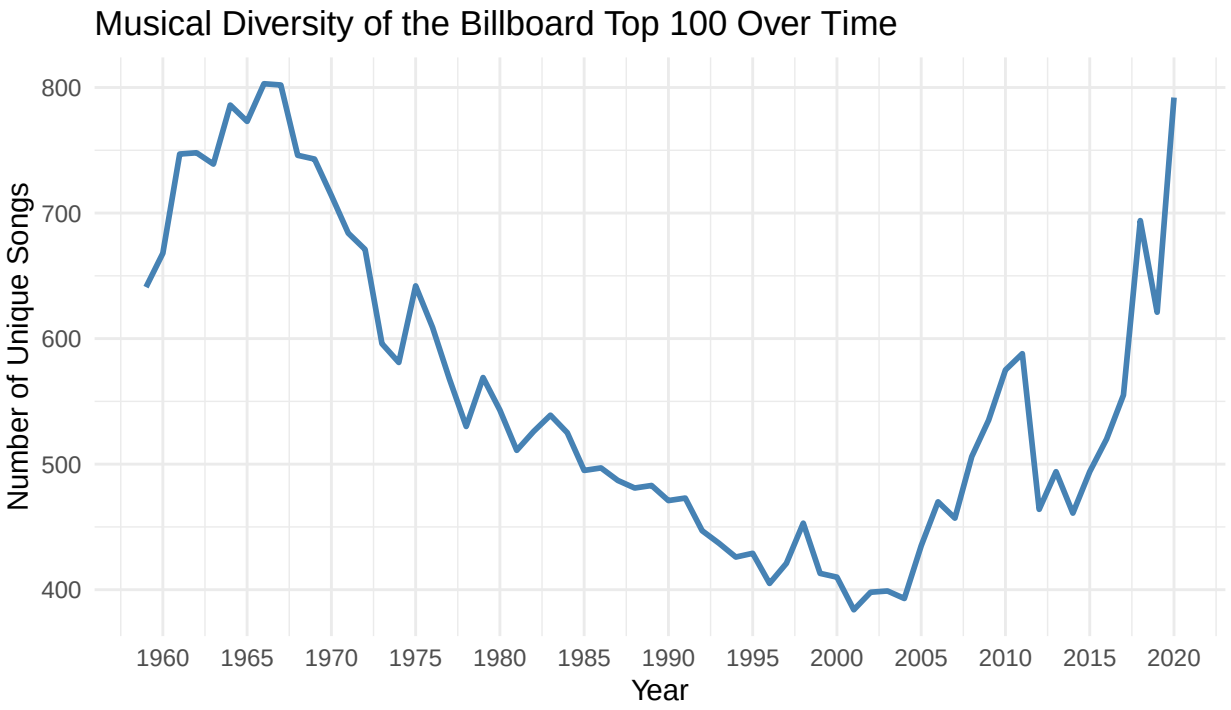
Part B: Is the "musical diversity" of the Billboard Top 100 changing over time? Let's find out. We'll measure the musical diversity of given year as *the number of unique songs that appeared in the Billboard Top 100 that year.* Make a line graph that plots this measure of musical diversity over the years. 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.

```r
# Filter the data to exclude years 1958 and 2021
filtered_data <- billboard %>%
  filter(year > 1958 & year < 2021)

# First step of data wrangling: Count the number of times each song appears in each year
song_counts <- filtered_data %>%
  group_by(year, song) %>%
  summarise(appearance_count = n(), .groups = "drop")

# Second step of data wrangling: Count the number of unique songs appearing each year
unique_songs_per_year <- song_counts %>%
  group_by(year) %>%
  summarise(unique_song_count = n_distinct(song), .groups = "drop")

# Plotting the data
ggplot(unique_songs_per_year, aes(x = year, y = unique_song_count)) +
  geom_line(color = "steelblue", size = 1) +
  scale_x_continuous(breaks = seq(1960, 2020, by = 5)) +  # Set x-axis intervals to every 2 years
  labs(title = "Musical Diversity of the Billboard Top 100 Over Time",
       x = "Year",
       y = "Number of Unique Songs",
       caption = "This line graph illustrates the changes in musical diversity on the Billboard Top 100
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
```

```
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



This line graph illustrates the changes in musical diversity on the Billboard Top 100 chart from 1959 to 2020, as measured by the number of unique songs that appeared in the chart each year. The data excludes the years 1958 and 2021 due to incomplete records. The graph shows significant fluctuations, with a peak in diversity around the late 1960s and a noticeable decline through the 1970s and 1980s. ecently, there has been a sharp increase in the number of unique songs, indicating a resurgence in musical diversity.

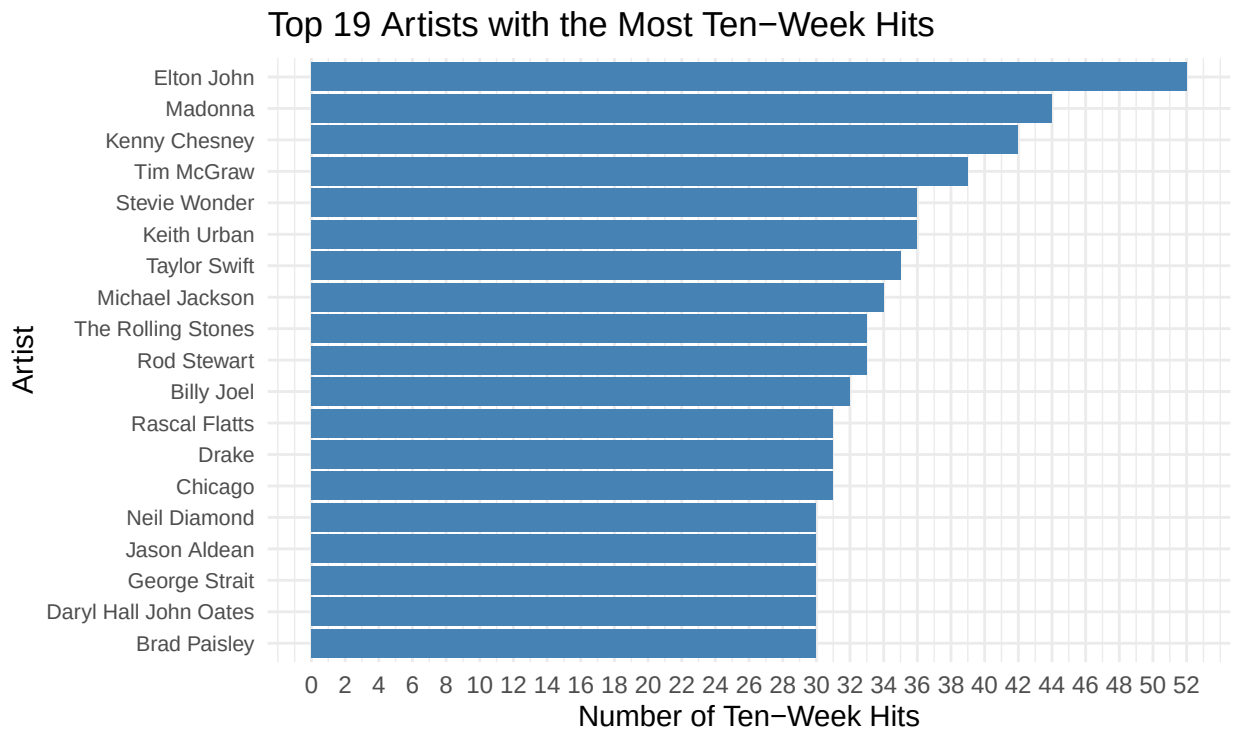
```
##### What the heck is going on with my captions, its uncentered now
######
########
#########
```

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:

1. You might find this easier to accomplish in two distinct sets of data wrangling steps.
2. 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.
3. 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. This is optional.

```r
# Identify "ten-week hits"
ten_week_hits <- billboard %>%
  group_by(performer, song) %>%
  summarise(weeks_on_chart = n(), .groups = "drop") %>%
  filter(weeks_on_chart >= 10)

# Count the number of ten-week hits per artist
artist_hits <- ten_week_hits %>%
  group_by(performer) %>%
  summarise(ten_week_hits_count = n(), .groups = "drop") %>%
  filter(ten_week_hits_count >= 30)

# Plotting
ggplot(artist_hits, aes(x = fct_reorder(performer, ten_week_hits_count), y = ten_week_hits_count)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +  # Flips the coordinates for better readability
  scale_y_continuous(breaks = seq(0, max(artist_hits$ten_week_hits_count), by = 2)) +  # Set y-axis int
  labs(title = "Top 19 Artists with the Most Ten-Week Hits",
       x = "Artist",
       y = "Number of Ten-Week Hits",
       caption = "The bar plot displays the number of ten-week hits for the top 19 artists who have had
  theme_minimal() +
  theme(axis.text.y = element_text(size = 8))  # Adjust text size for readability
```



The bar plot displays the number of ten−week hits for the top 19 artists who have had at least 30 ten−week hits on the Billboard Top 100 chart. A 'ten−week hit' is defined as a song that appeared on the Billboard Top 100 for at least ten weeks. The artists are ordered by the number of ten−week hits, with Elton John leading the list, followed by Madonna and Kenny Chesney.