R-Ladies Helsinki February Event

Hazel KAVILI

1/27/2020

Spotify Songs

We will work on a *TidyTuesday* dataset today.

(Try to check out the TidyTuesday concept after the event! You'll love it!)

Load libraries

```
library(tidyverse)
library(lubridate)
library(knitr)
```

Read the data set from source

```
spotify_songs <-
  readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-0
```

Start Exploring

glimpse function makes it possible to see every column and some observations in a data frame.

```
glimpse(spotify_songs)
```

```
## Observations: 32,833
## Variables: 23
## $ track_id                <chr> "6f807x0ima9a1j3VPbc7VN", "0r7CVbZTWZ...
## $ track_name              <chr> "I Don't Care (with Justin Bieber) - ...
## $ track_artist            <chr> "Ed Sheeran", "Maroon 5", "Zara Larss...
## $ track_popularity        <dbl> 66, 67, 70, 60, 69, 67, 62, 69, 68, 6...
## $ track_album_id          <chr> "2oCs0DGTsRO98Gh5ZSl2Cx", "63rPSO264u...
## $ track_album_name        <chr> "I Don't Care (with Justin Bieber) [L...
## $ track_album_release_date <chr> "2019-06-14", "2019-12-13", "2019-07-...
## $ playlist_name           <chr> "Pop Remix", "Pop Remix", "Pop Remix"...
## $ playlist_id             <chr> "37i9dQZF1DXcZDD7cfEKhW", "37i9dQZF1D...
## $ playlist_genre          <chr> "pop", "pop", "pop", "pop", "pop", "p...
## $ playlist_subgenre       <chr> "dance pop", "dance pop", "dance pop"...
## $ danceability            <dbl> 0.748, 0.726, 0.675, 0.718, 0.650, 0....
## $ energy                  <dbl> 0.916, 0.815, 0.931, 0.930, 0.833, 0....
## $ key                     <dbl> 6, 11, 1, 7, 1, 8, 5, 4, 8, 2, 6, 8, ...
## $ loudness                <dbl> -2.634, -4.969, -3.432, -3.778, -4.67...
## $ mode                    <dbl> 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1...
## $ speechiness             <dbl> 0.0583, 0.0373, 0.0742, 0.1020, 0.035...
## $ acousticness            <dbl> 0.10200, 0.07240, 0.07940, 0.02870, 0...
## $ instrumentalness        <dbl> 0.00e+00, 4.21e-03, 2.33e-05, 9.43e-0...
## $ liveness                <dbl> 0.0653, 0.3570, 0.1100, 0.2040, 0.083...
```

```
## $ valence                 <dbl> 0.518, 0.693, 0.613, 0.277, 0.725, 0....
## $ tempo                   <dbl> 122.036, 99.972, 124.008, 121.956, 12...
## $ duration_ms             <dbl> 194754, 162600, 176616, 169093, 18905...
```

Exploring data

Some songs are duplicated, because they're in different albums or in different playlist. I wonder, how many distinct tracks there are for each artist, and I'll look for top 20:

```
artists_tracks <- spotify_songs %>%
  distinct(track_id, .keep_all = TRUE) %>%
  count(track_artist, sort = TRUE) %>%
  top_n(n = 20, wt = n)
```

I want to see results in a stylish table:

```
head(artists_tracks) %>%
  kable(align = "lccrr",  caption = "Top 10 artists with most tracks")
```

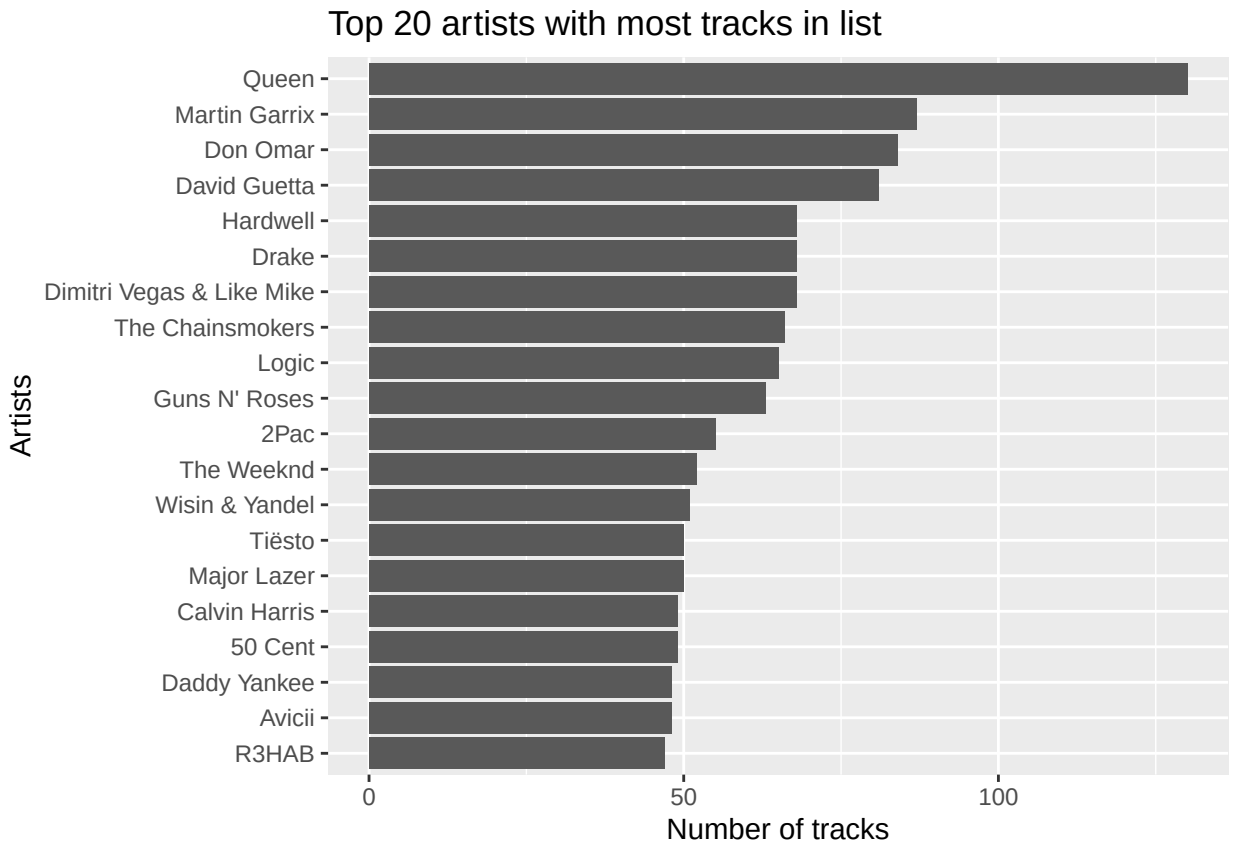
Table 1: Top 10 artists with most tracks

track_artist	n
Queen	130
Martin Garrix	87
Don Omar	84
David Guetta	81
Dimitri Vegas & Like Mike	68
Drake	68

Let's create a plot by using this data:

```
artist_plot <- ggplot(data = artists_tracks, aes(x = reorder(track_artist, n), y = n)) +
  geom_bar(stat = 'identity') +
  coord_flip() +
  labs(title = 'Top 20 artists with most tracks in list',
       x = 'Artists',
       y = 'Number of tracks')

artist_plot
```



I realise some artists released so many albums during years and I wonder is the longest time passed since they released their last album.

```
album_release_years <-
  spotify_songs %>%
  mutate(release_year = as.numeric(str_sub(track_album_release_date, 1, 4))) %>% #get only year informa
  distinct(track_id, .keep_all = TRUE) %>%
  distinct(track_name, track_artist, .keep_all = TRUE) %>%
  group_by(track_artist) %>%
  mutate(first_release_year = min(release_year),
         last_release_year = max(release_year),
         year_diff = last_release_year - first_release_year) %>%
  ungroup()  %>%
  mutate(track_artist = fct_reorder(track_artist, year_diff))
```

```
album_release_years %>%
  filter(year_diff > 50) %>%
  ggplot() +
  geom_path(aes(x = release_year, y = track_artist)) +
  geom_point(aes(release_year, track_artist, color = track_artist, alpha = 0.1), size = 2) +
  labs(title = '', x = 'Album release years', y = 'Artists') +
  theme_light()
```

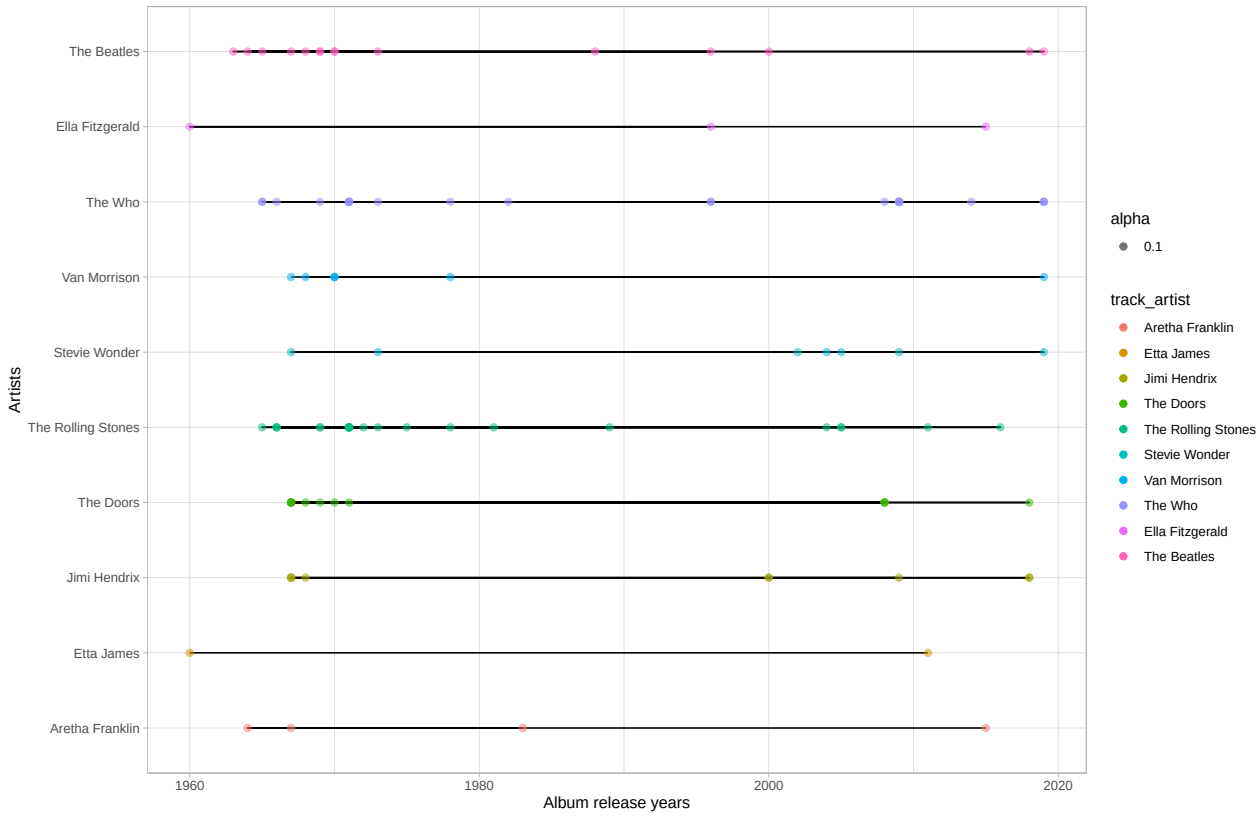


Figure 1: Years passed since first album release