# W8_WebScraping

Nicole Dwenger

2020/11/01

## Week 8: Webscraping

I decided to scrape some data from Spotify Charts, and make some plots with the data. To do this, I followed this Tutorial.

```
library(pacman)
pacman::p_load(tidyverse, rvest, magrittr, scales, knitr, lubridate, tibble, gdata)
```

```
## Installing package into '/Users/nicoledwenger/Library/R/4.0/library'
## (as 'lib' is unspecified)

##
## The downloaded binary packages are in
##  /var/folders/2v/vhv0py1534jdjsfn2bhgv5w40000gn/T//RtmpEMktPF/downloaded_packages

##
## gdata installed
```

### 1. Defining urls to scrape from

```
# defining the base url
dk <- "https://spotifycharts.com/regional/dk/weekly/" # denmark
de <- "https://spotifycharts.com/regional/de/weekly/" # germany
us <- "https://spotifycharts.com/regional/us/weekly/" # us
uk <- "https://spotifycharts.com/regional/gb/weekly/" # uk

# defining the variation in the ulr (wich is the week)
timevalues1 <- seq(as.Date("2019/12/27"), as.Date("2020/10/23"), by = "week")
timevalues2 <- seq(as.Date("2020-01-03"), as.Date("2020/10/30"), by = "week")

# to make the final urls from which I will be scaping, I add together the base + variation
unitedata <- function(country, timevalue1, timevalue2){
 full_url <- paste0(country, timevalue1, "--", timevalue2)
 full_url
}

# for each country I apply the function to get all the final urls I want to scrape from
fullurl_dk <- unitedata(dk, timevalues1, timevalues2)
```

```
fullurl_de <- unitedata(de, timevalues1, timevalues2)
fullurl_us <- unitedata(us, timevalues1, timevalues2)
fullurl_uk <- unitedata(uk, timevalues1, timevalues2)
```

## 2. Defining function to use for scraping and applying it

```
spotify_scrape <- function(x){
 page <- x
 rank <- page %>% read_html() %>% html_nodes('.chart-table-position') %>% html_text() %>% as.data.frame
 track <- page %>% read_html() %>% html_nodes('strong') %>% html_text() %>% as.data.frame()
 artist <- page %>% read_html() %>% html_nodes('.chart-table-track span') %>% html_text() %>% as.data.fr
 streams <- page %>% read_html() %>% html_nodes('td.chart-table-streams') %>% html_text() %>% as.data.fr
 date <- page %>% read_html() %>% html_nodes('.responsive-select-value') %>% html_text() %>% .[3]
 url <- page %>% read_html() %>% html_nodes('td.chart-table-image a') %>% html_attr('href')
 # combine, name, and make it a tibble
 chart <- cbind(rank, track, artist, streams, url)
 chart$date <- date
 names(chart) <- c("rank", "track", "artist", "streams", "url", "date")
 chart <- as_tibble(chart)
 return(chart)
}

# apply the function to scape the data
data_dk <- map_df(fullurl_dk, spotify_scrape)
data_de <- map_df(fullurl_de, spotify_scrape)
data_us <- map_df(fullurl_us, spotify_scrape)
data_uk <- map_df(fullurl_uk, spotify_scrape)
```

## 3. Cleaning data

```
# combine data from all countries,
data <- combine(data_dk, data_de, data_us, data_uk) %>%
  # clean the columns and make some new ones from the existing ones
  mutate(country = str_extract(source, ".{2}$"),
         streams = as.numeric(gsub(",", "", streams)),
         artist = str_remove(artist, "by "),
         date = as.Date(date, "%m/%d/%Y"),
         month = str_extract(date, "\\b\\d{2}\\b"),
         track_id = str_remove(url, "https://open.spotify.com/track/")) %>%
  # and put it into a nice order
  select(rank,
         track,
         artist,
         streams,
         date,
         month,
         country,
         url,
         track_id)
```

```r
# calculate the total streams for each country and add to dataframe
total_country_streams <- data %>%
  group_by(country, date) %>%
  summarise(total_country = sum(streams), .groups = "drop_last")
data <- merge(data, total_country_streams, by = c("country", "date"))

# save data
write.csv(data, "W8_spotify_data.csv")
```

## 4. Top artists in each country

I am aware that here it would make sense to also take into account how many songs each artist had, but I ignored that here and just went for the streams of each artist.

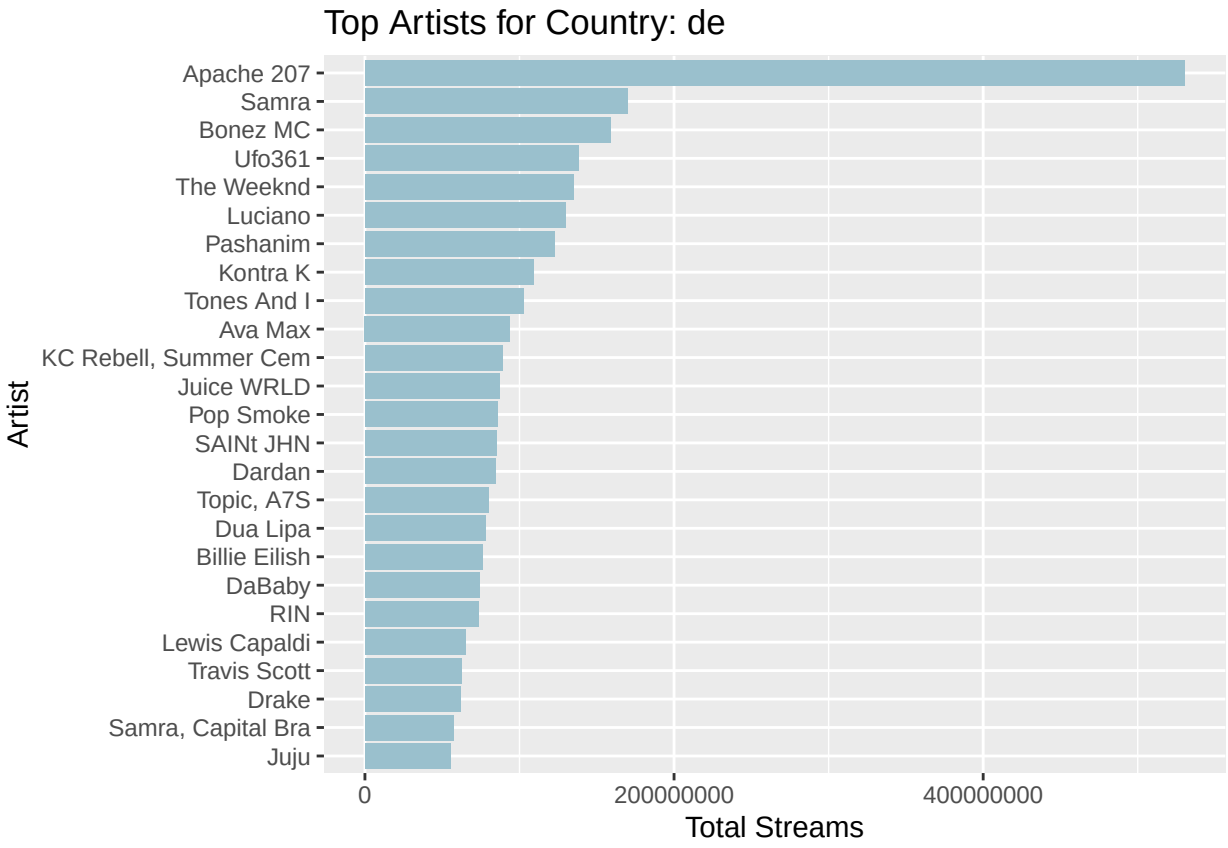
```r
# load data
data <- read_csv("W8_spotify_data.csv")
```

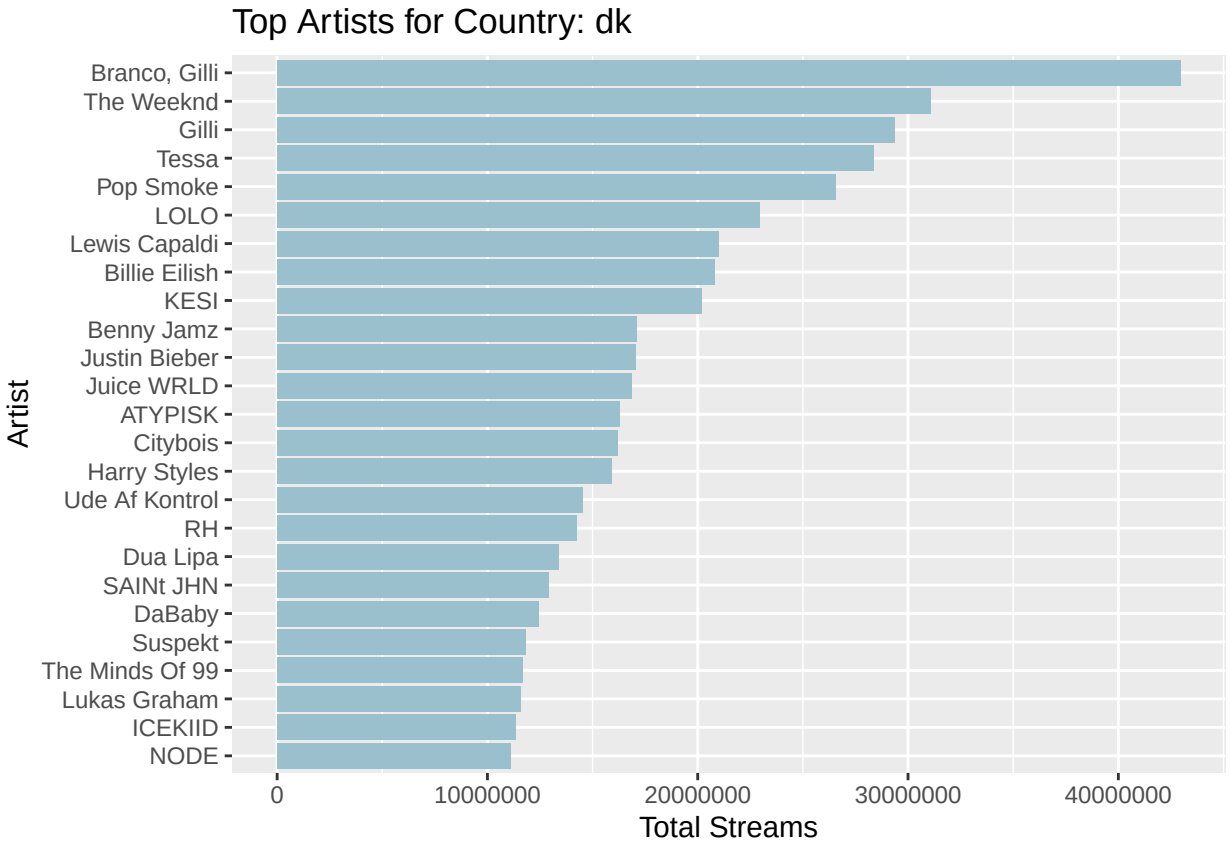
```
## Warning: Missing column names filled in: 'X1' [1]
```

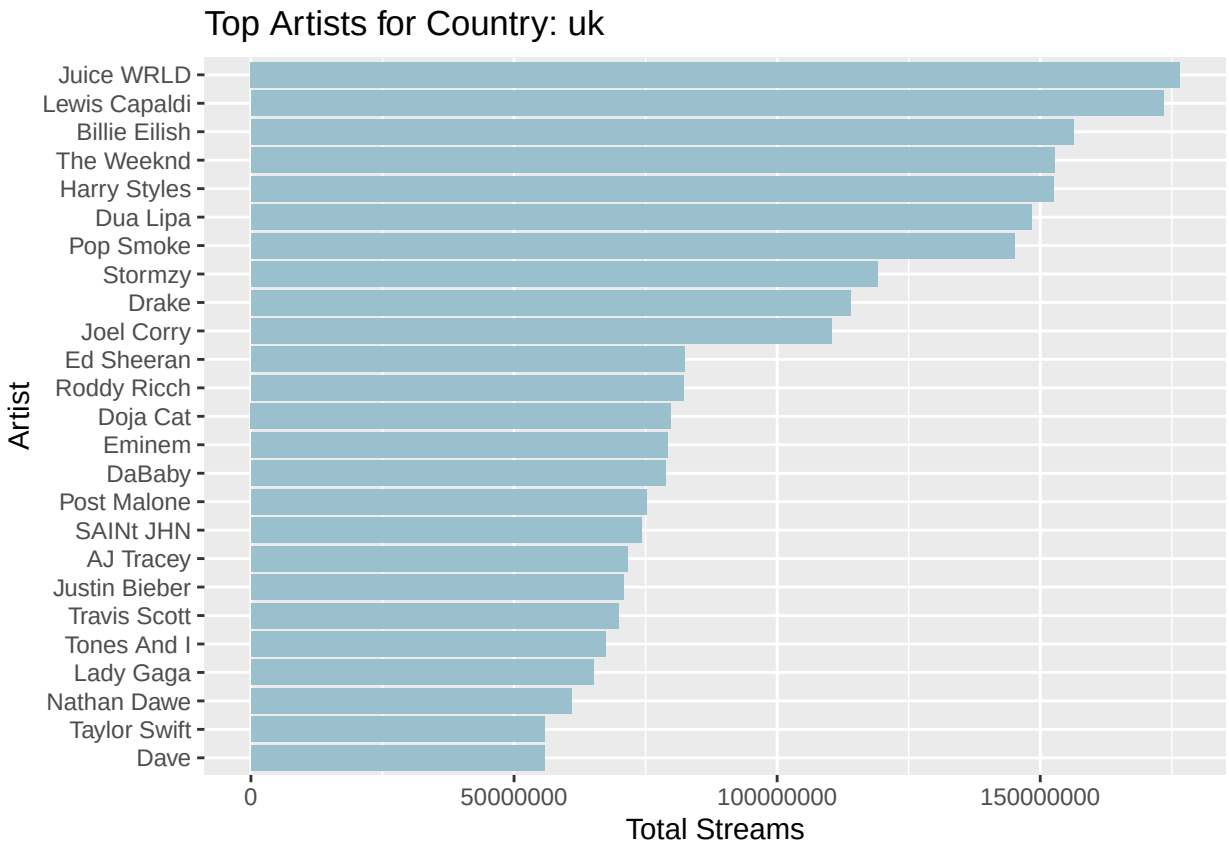
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   country = col_character(),
##   date = col_date(format = ""),
##   rank = col_double(),
##   track = col_character(),
##   artist = col_character(),
##   streams = col_double(),
##   month = col_character(),
##   url = col_character(),
##   track_id = col_character(),
##   total_country = col_double()
## )
```

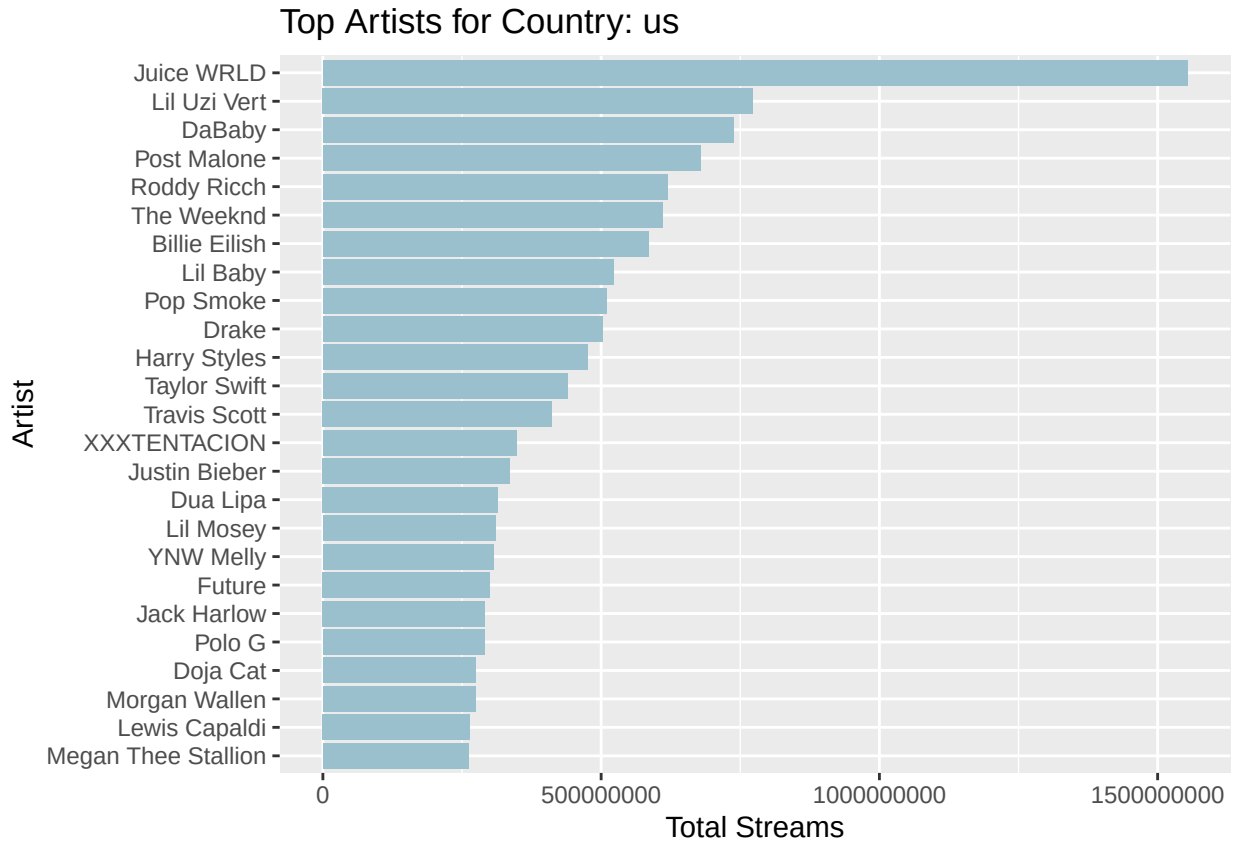
```r
# disable scientific notation
options(scipen=999)

# make function for plot
top_artist_plot <- function(short_country) {
  data %>%
     filter(country == short_country) %>%
     group_by(artist) %>%
     summarise(total_artist = sum(streams), .groups = "drop_last") %>%
     arrange(desc(total_artist)) %>%
     top_n(25, total_artist) %>%
 ggplot() +
     geom_col(aes(reorder(artist, total_artist), y = total_artist), fill = "lightblue3") +
     coord_flip() +
     labs(x = "Artist", y = "Total Streams", title = paste("Top Artists for Country:", short_country))
}
```

```r
# apply for each country
for (country in unique(data$country)) {
  print(top_artist_plot(country))
}
```




Top Artists for Country: dk
Artist
Branco, Gilli
The Weeknd
Gilli
Tessa
Pop Smoke
LOLO
Lewis Capaldi
Billie Eilish
KESI
Benny Jamz
Justin Bieber
Juice WRLD
ATYPISK
Citybois
Harry Styles
Ude Af Kontrol
RH
Dua Lipa
SAINt JHN
DaBaby
Suspekt
The Minds Of 99
Lukas Graham
ICEKIID
NODE
0
10000000
20000000
30000000
40000000
Total Streams


Top Artists for Country: uk
Juice WRLD
Lewis Capaldi
Billie Eilish
The Weeknd
Harry Styles
Dua Lipa
Pop Smoke
Stormzy
Drake
Joel Corry
Ed Sheeran
Roddy Ricch
Doja Cat
Eminem
DaBaby
Post Malone
SAINt JHN
AJ Tracey
Justin Bieber
Travis Scott
Tones And I
Lady Gaga
Nathan Dawe
Taylor Swift
Dave
Artist
0
50000000
100000000
150000000
Total Streams

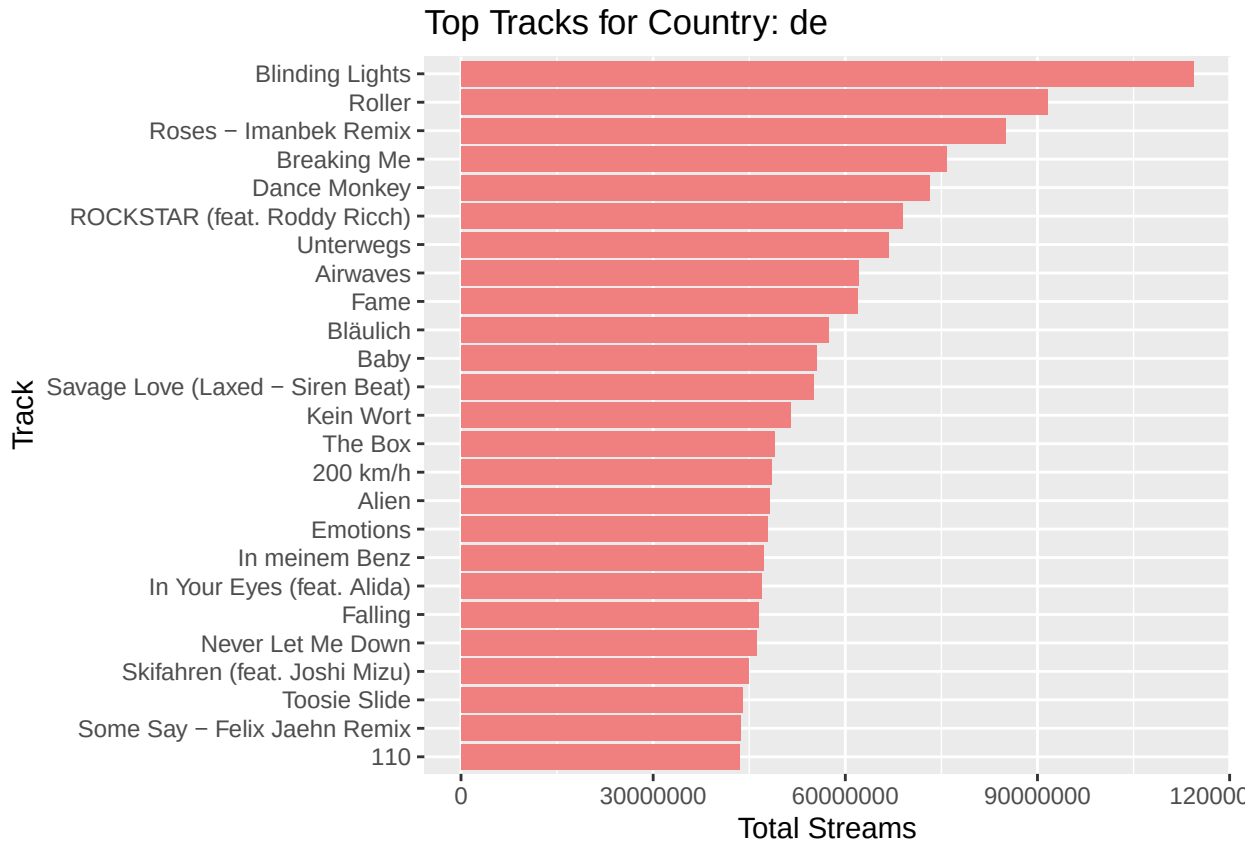


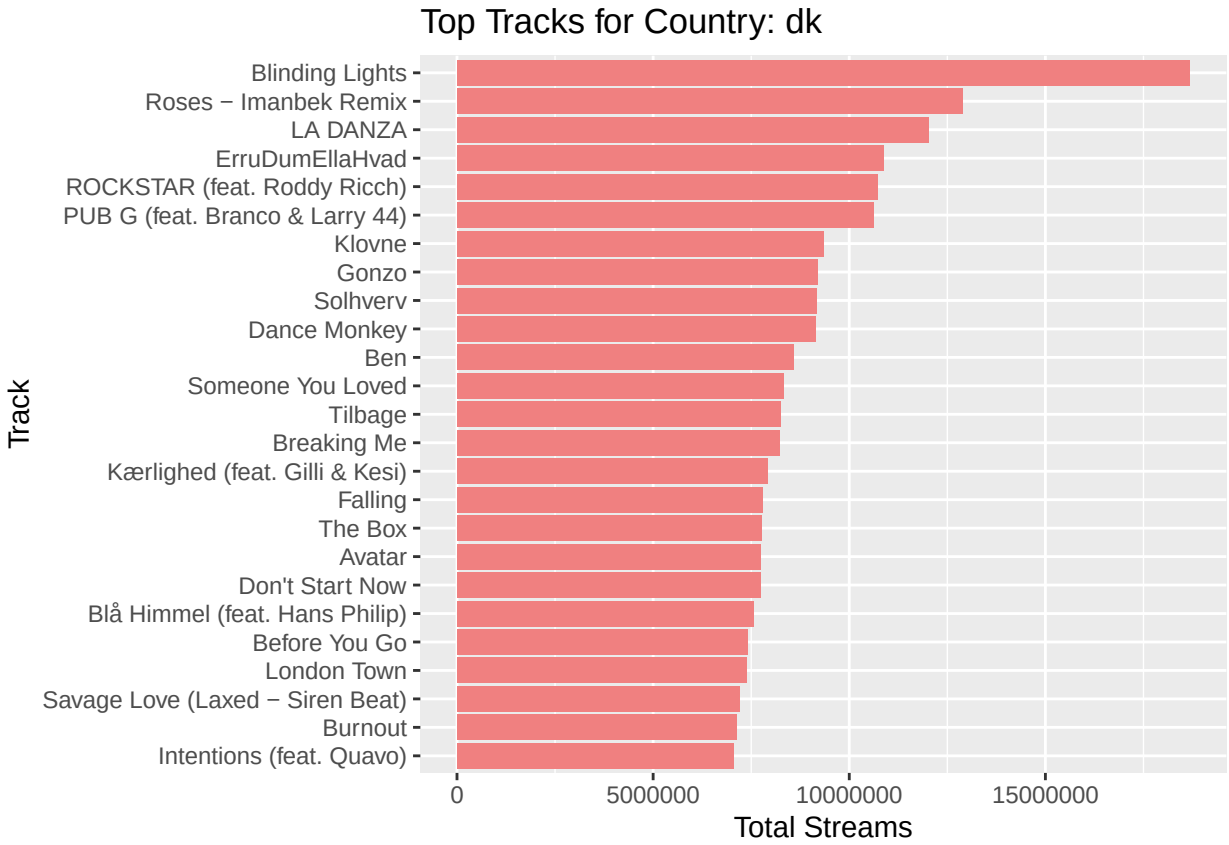

## 5. Top songs in each country

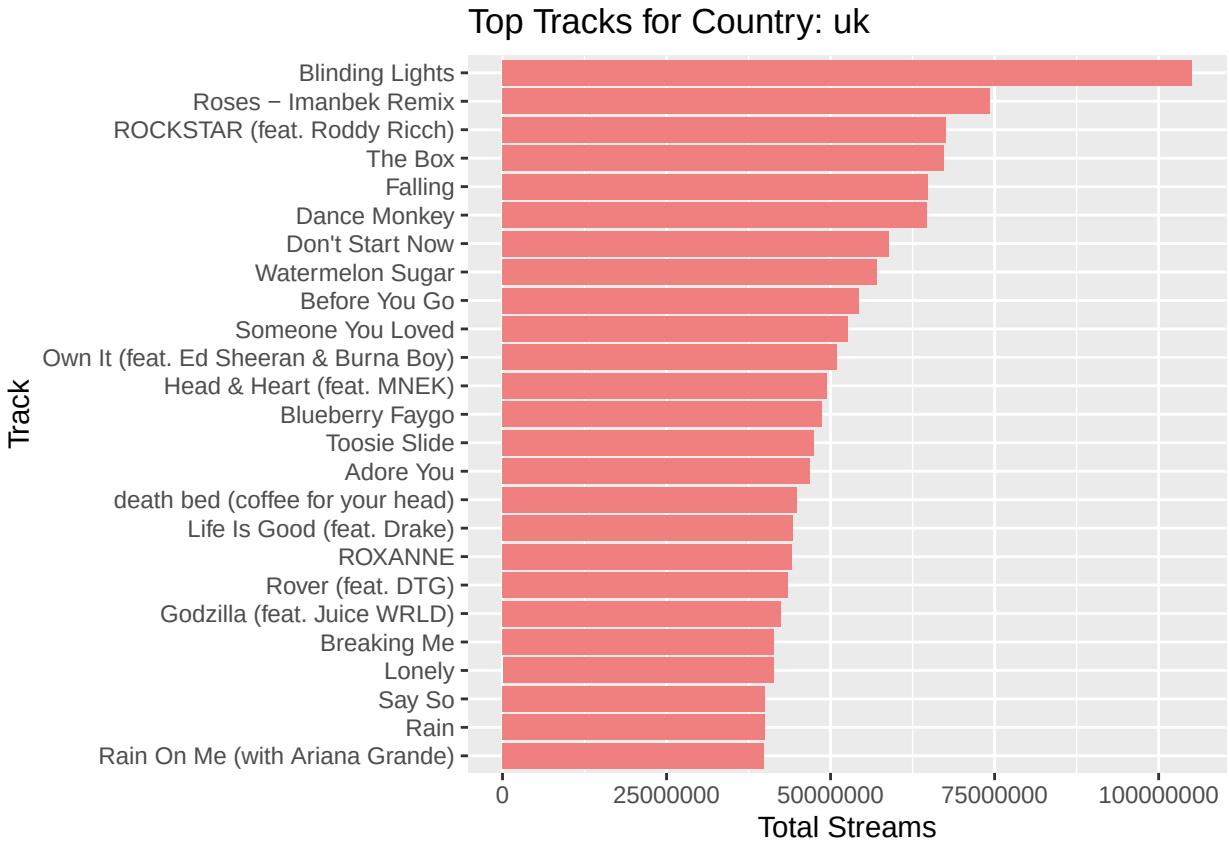
I am aware that here it would make sense to also take into account how many songs each artist had, but I ignored that here and just went for the streams of each artist.

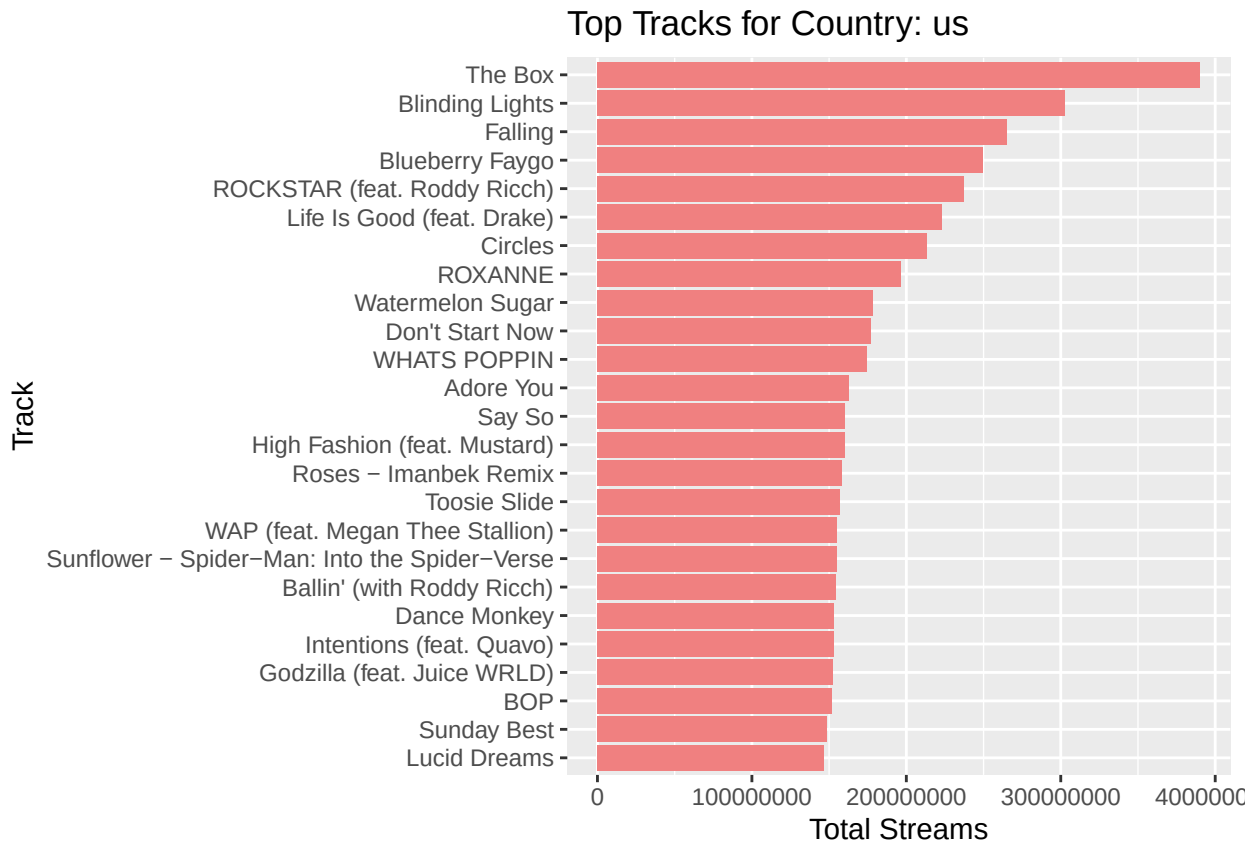
```
# disable scientific notation
options(scipen=999)

# make function for plot
top_track_plot <- function(short_country) {
  data %>%
     filter(country == short_country) %>%
     group_by(track) %>%
     summarise(total_track = sum(streams), .groups = "drop_last") %>%
     arrange(desc(total_track)) %>%
     top_n(25, total_track) %>%
 ggplot() +
     geom_col(aes(reorder(track, total_track), y = total_track), fill = "lightcoral") +
     coord_flip() +
     labs(x = "Track", y = "Total Streams", title = paste("Top Tracks for Country:", short_country))
}

# apply for each country
for (country in unique(data$country)) {
```

```
  print(top_track_plot(country))
}
```




Top Tracks for Country: dk
Track
Blinding Lights
Roses − Imanbek Remix
LA DANZA
ErruDumEllaHvad
ROCKSTAR (feat. Roddy Ricch)
PUB G (feat. Branco & Larry 44)
Klovne
Gonzo
Solhverv
Dance Monkey
Ben
Someone You Loved
Tilbage
Breaking Me
Kærlighed (feat. Gilli & Kesi)
Falling
The Box
Avatar
Don't Start Now
Blå Himmel (feat. Hans Philip)
Before You Go
London Town
Savage Love (Laxed − Siren Beat)
Burnout
Intentions (feat. Quavo)
0
5000000
10000000
15000000
Total Streams

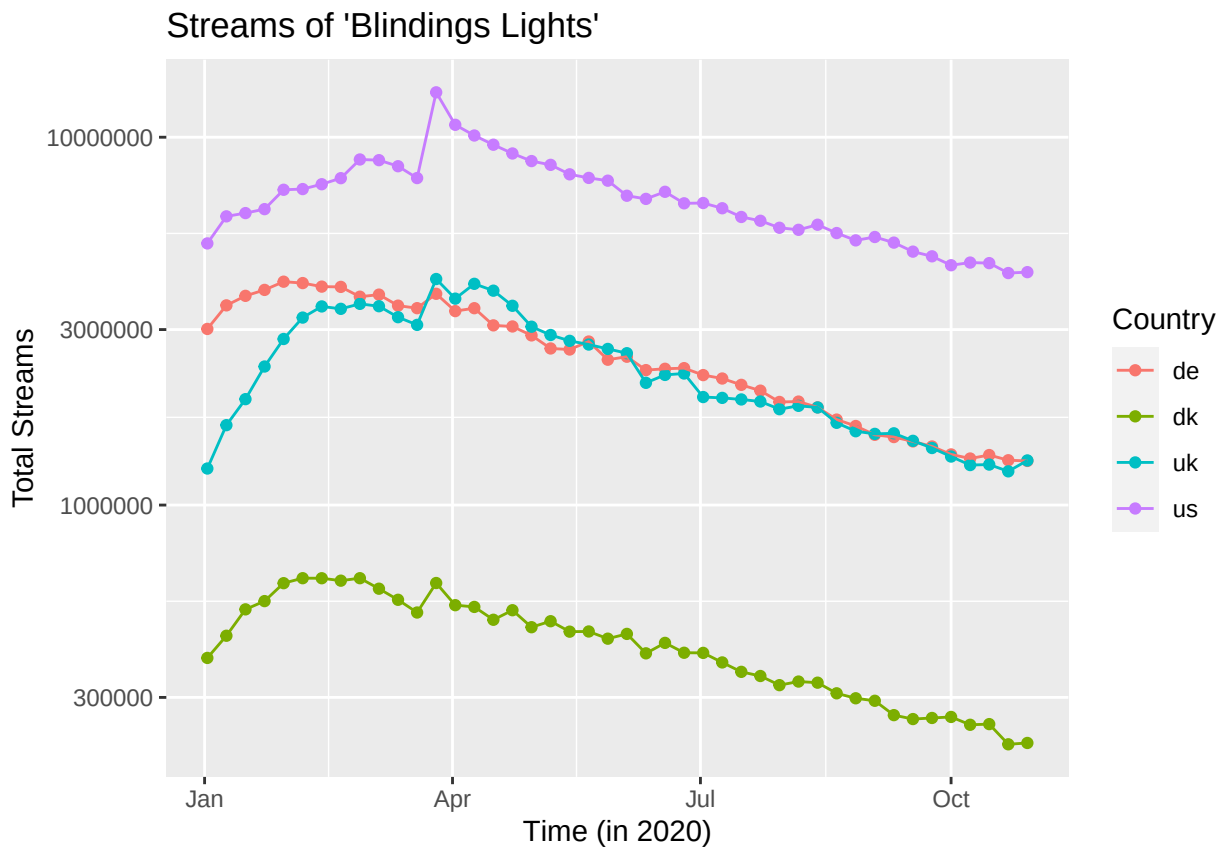

Top Tracks for Country: uk
Track
Blinding Lights
Roses − Imanbek Remix
ROCKSTAR (feat. Roddy Ricch)
The Box
Falling
Dance Monkey
Don't Start Now
Watermelon Sugar
Before You Go
Someone You Loved
Own It (feat. Ed Sheeran & Burna Boy)
Head & Heart (feat. MNEK)
Blueberry Faygo
Toosie Slide
Adore You
death bed (coffee for your head)
Life Is Good (feat. Drake)
ROXANNE
Rover (feat. DTG)
Godzilla (feat. Juice WRLD)
Breaking Me
Lonely
Say So
Rain
Rain On Me (with Ariana Grande)
0
25000000
50000000
75000000
100000000
Total Streams



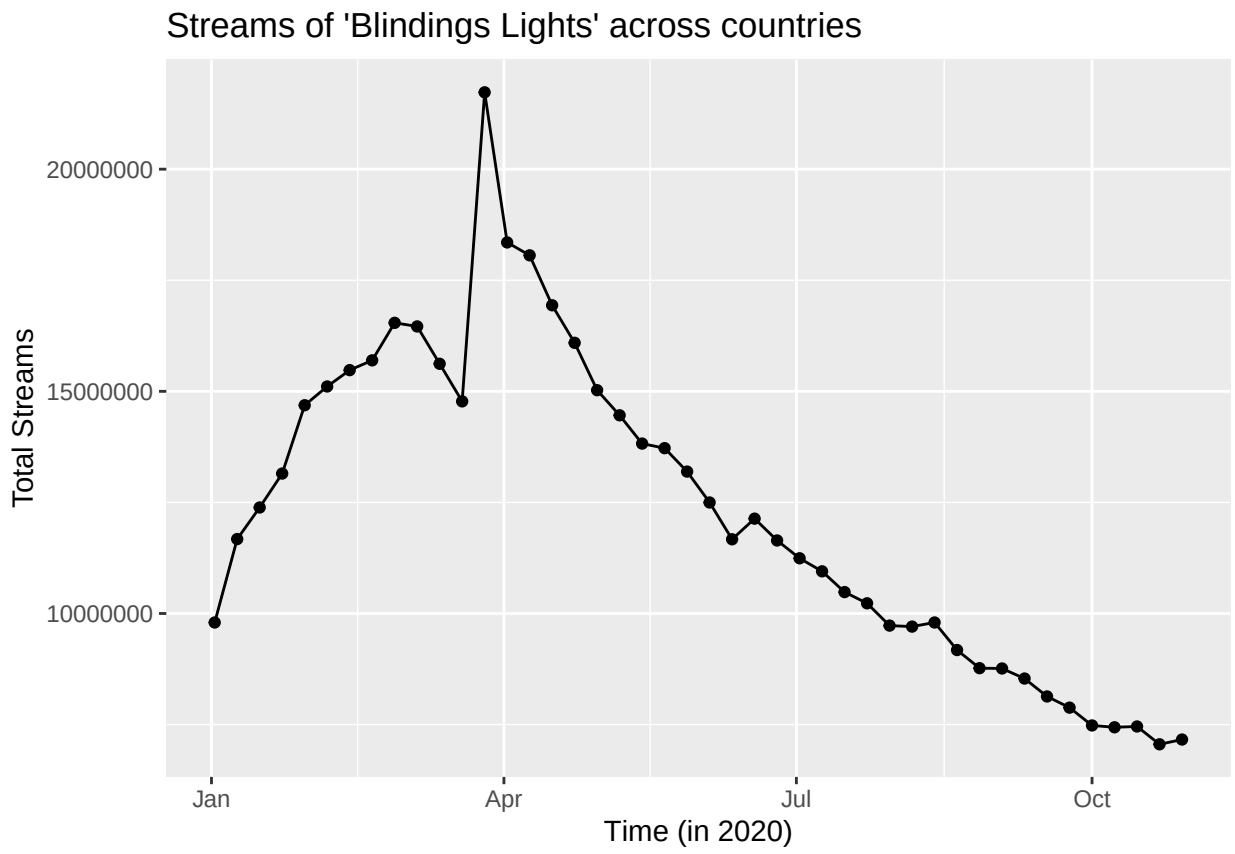

## 6. Development of song "Blinding Lights"

I saw that the song was the Nr. 1 for 3 of the countries, and Nr. 2 for the us, so I decided to plot how the number of streams developed in the countries.

```
data %>%
  filter(track == "Blinding Lights") %>%
  ggplot() +
    geom_point(aes(y = streams, x = date, color = country)) +
    geom_line(aes(y = streams, x = date, color = country)) +
    scale_y_log10() +
    labs(x = "Time (in 2020)", y = "Total Streams", color = "Country", title = "Streams of 'Blindings Li
```

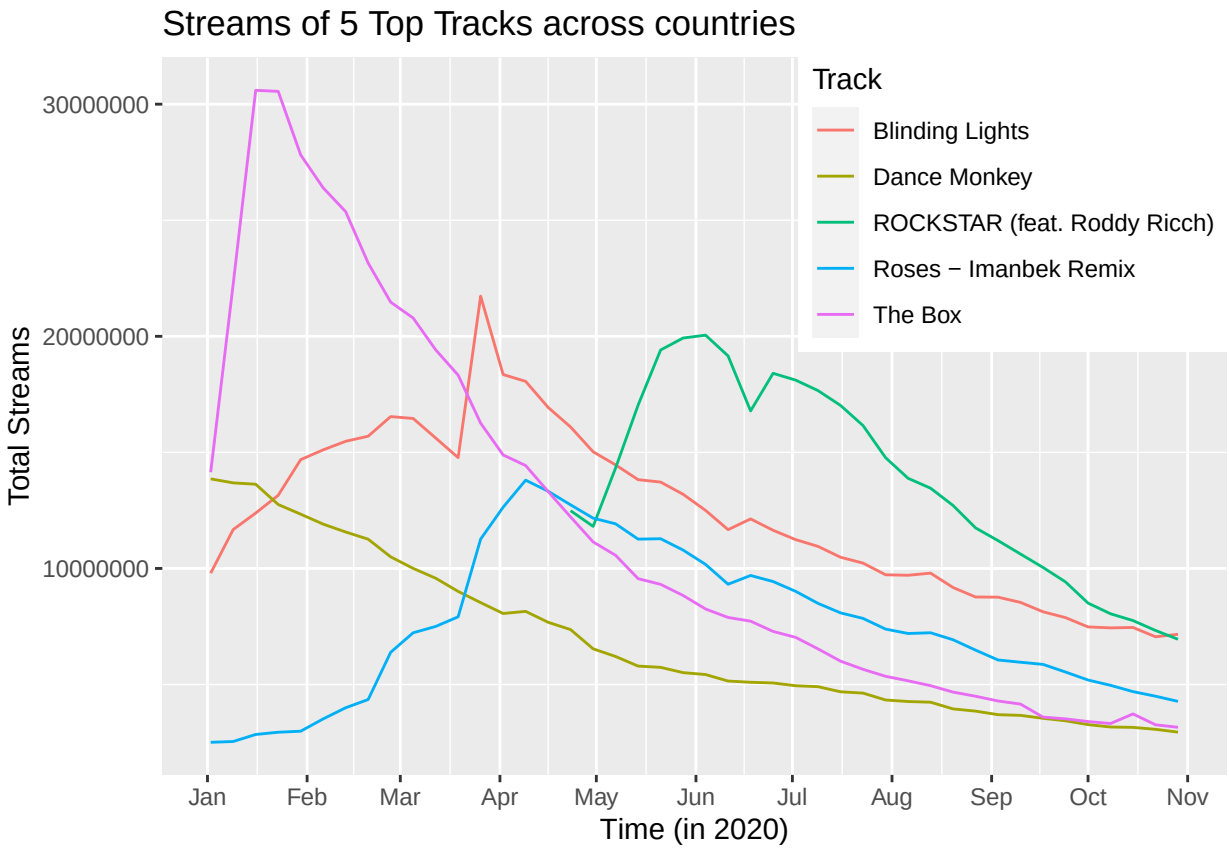



```
data %>%
  filter(track == "Blinding Lights") %>%
  group_by(date) %>%
  summarise(total = sum(streams), .groups = "drop_last") %>%
  ggplot(aes(y = total, x = date)) +
    geom_point() +
    geom_line() +
    labs(x = "Time (in 2020)", y = "Total Streams", title = "Streams of 'Blindings Lights' across counti
```




## 7. Development of 5 songs across all countries

```
data %>%
  filter(track == "Roses - Imanbek Remix" | track == "The Box" | track == "Blinding Lights" | track ==
  group_by(track, date) %>%
  summarise(total = sum(streams), .groups = "drop_last") %>%
  ggplot(aes(y = total, x = date, color = track, group = track)) +
    #geom_point() +
    geom_line() +
    scale_x_date(date_breaks = "1 month", date_labels = "%b") +
    theme(legend.position = c(0.8, 0.8)) +
    labs(x = "Time (in 2020)", y = "Total Streams", color = "Track", title = "Streams of 5 Top Tracks a
```


Streams of 5 Top Tracks across countries
Track
Blinding Lights
Dance Monkey
ROCKSTAR (feat. Roddy Ricch)
Roses − Imanbek Remix
The Box
Total Streams
30000000
20000000
10000000
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Time (in 2020)