**Bộ giáo dục và đào tạo**

**Trường Đại học Ngoại ngữ - Tin học TP.HCM**



*Đề tài:*

**SỬ DỤNG NGÔN NGỮ R PHÂN TÍCH DỮ LIỆU ỨNG DỤNG NGHE NHẠC**

GVHD: PGS.TS Nguyễn Thanh Bình

Thành Viên: 21DH114518-Lê Đại Thắng

**Tháng ... Năm 2023**

# Mục lục

[**I.** **Giới thiệu.** 1](#_Toc161939574)

[**1.1** **Giới thiệu ngôn ngữ R.** 1](#_Toc161939575)

[**1.2** **Ưu và nhược điểm của R.** 1](#_Toc161939576)

[**1.3** **Các gói R được sử dụng trong phân tích dữ liệu âm nhạc.** 2](#_Toc161939577)

[**II.** **Giới thiệu về tập dữ liệu.** 3](#_Toc161939578)

[**2.1** **Giới thiệu về dữ liệu.** 3](#_Toc161939579)

[**2.2** **Mô tả tập dữ liệu.** 4](#_Toc161939580)

[**2.3 Mục đích của đề tài** 5](#_Toc161939581)

[**III.** **Xử lý dữ liệu.** 5](#_Toc161939582)

[**3.1** **Thay thế giá trị Na với giá trị trung vị (median)** 6](#_Toc161939583)

[**3.2 Xóa các hàng bị trùng lặp** 7](#_Toc161939584)

[**3.3 Chuyển đổi kiểu dữ liệu từ mili giây sang phút** 7](#_Toc161939585)

[**3.4 Xóa các cột không liên quan** 8](#_Toc161939586)

[**IV.** **Đặt câu hỏi để phân tích và trực quan tập dữ liệu.** 8](#_Toc161939587)

[**4.1** **Mức độ phổ biến của bài hát được phân bổ như thế nào? 10 bài hát hàng đầu là gì?** 8](#_Toc161939588)

[**4.2** **Top 5 thể loại nhạc phổ biến nhất?** 11](#_Toc161939589)

[**4.3** **Số lượng bài hát có nội dung không phù hợp với trẻ em trong các bài hát hàng đầu.** 12](#_Toc161939590)

[**4.4** **Mức độ phổ biến của bài hát thay đổi như thế nào với nội dung nhạy cảm? Nội dung nhạy cảm có tác động tích cực, tiêu cực hoặc trung lập đến mức độ phổ biến của bài hát không?** 13](#_Toc161939591)

[**4.5** **Tỷ lệ bài hát được đánh giá cao, trung bình hoặc thấp là bao nhiêu ?** 14](#_Toc161939592)

[**4.6** **15 nghệ sũ phát hành nhiều ca khúc nhất từ 2000 – 2020 ?** 16](#_Toc161939593)

[**4.7** **Độ dài của bài hát có thay đổi qua các năm không ?** 17](#_Toc161939594)

[**4.8** **Mức độ phổ biến và giá trị của bài hát tương quan như thế nào với thời lượng?** 19](#_Toc161939595)

[**V.** **Tổng hợp thống kê (Statistical Summary).** 20](#_Toc161939596)

[**5.1** **Sử dụng hàm summary() cho cột valance.** 20](#_Toc161939597)

[**5.2** **Sủ dụng hàm summary() cho cột duration\_min:** 21](#_Toc161939598)

[**5.3** **Sử dụng hàm summary() cho cột tempo.** 22](#_Toc161939599)

[**VI.** **Tạo các cột mới.** 23](#_Toc161939600)

[**6.1** **Tạo một cột mới với mức độ phổ biến (popularity\_rating) từ biến số đến biến phân loại.** 23](#_Toc161939601)

[**6.2** **Ta tạo cột mới là “duration\_min” và chuyển từ miliseconds (mili giây) sang minutes (phút). Sau đó, ta nhóm cột mới “duration\_min” với cột “year” thành dữ liệu mới “song\_duration”.** 24](#_Toc161939602)

[**VII.** **Kết luận.** 24](#_Toc161939603)

[**Tài liệu tham khảo.** 24](#_Toc161939604)

[**Phụ lục code demo** 25](#_Toc161939605)

**Danh Mục Hình ảnh**

[Hình 1. Một vài dòng đầu và tổng số cột của tập dữ liệu 3](#_Toc161939606)

[Hình 2. Trước khi điền dữ liệu bị null 6](#_Toc161939607)

[Hình 3. Sau khi điền dữ liệu bị null 6](#_Toc161939608)

[Hình 4. Kết quả sau khi chuyển dữ liệu từ mili giây sang phút 8](#_Toc161939609)

[Hình 5. Mức độ phổ biến của bài hát 9](#_Toc161939610)

[Hình 6. 10 bài hát phổ biến hàng đầu 10](#_Toc161939611)

[Hình 7. Top 5 thể loại nhạc phổ biến 11](#_Toc161939612)

[Hình 8. Số lượng bài hát không phù hợp với trẻ em 12](#_Toc161939613)

[Hình 9. Mức độ phổ biến của bài hát 13](#_Toc161939614)

[Hình 10. Tỷ lệ bài hát được đánh giá 14](#_Toc161939615)

[Hình 11. Biểu đồ 15 nghệ sĩ phát hành nhiều ca khúc nhất 16](#_Toc161939616)

[Hình 12. Biểu đồ độ dài của bài hát thay đổi qua các năm 17](#_Toc161939617)

[Hình 13. Biểu đồ thể hiện mức độ phổ biến và giá trị của bài hát 19](#_Toc161939618)

[Hình 14. Toàn bộ dữ liệu cho hàm tóm tắt 20](#_Toc161939619)

[Hình 15. Cột mới với popularity\_rating từ biến số đến biến phân loại 23](file:///C:\Users\Admin\Downloads\Báo%20cáo%20bigdata%20ổn%20gấp%202.docx#_Toc161939620)

[Hình 16. Cột mói của "duration\_min" 24](#_Toc161939621)

1. **Giới thiệu.**
   1. **Giới thiệu ngôn ngữ R.**

* R là một chương trình mã nguồn mở được sử dụng cho tính toán thống kê và đồ họa. Nó là 1 trong những ngôn ngữ lập trình nổi tiếng nhất ngày nay. Ngôn ngữ R được lấy cảm hứng bởi S+, nó tương tự như ngôn ngữ lập trình S
* R cung cấp một loạt các kỹ thuật thống kê (mô hình tuyến tính và phi tuyến tính, kiểm định thống kê cổ điển, phân tích chuỗi thời gian, phân loại, phân cụm,...) và kỹ thuật đồ họa đa dạng, và có khả năng mở rộng cao. Ngôn ngữ S thường được lựa chọn trong nghiên cứu về phương pháp thống kê, và R cung cấp một lối vào nguồn mở để tham gia vào hoạt động đó.
* Một trong những điểm mạnh của R là sự dễ dàng trong việc tạo ra các đồ thị chất lượng xuất bản có thiết kế tốt, bao gồm cả các ký hiệu toán học và công thức khi cần thiết. Được chú ý đặc biệt đến các giá trị mặc định cho những lựa chọn thiết kế nhỏ trong đồ họa, nhưng người dùng vẫn giữ được toàn quyền kiểm soát đầy đủ.
  1. **Ưu và nhược điểm của R.**

\* Ưu điểm:

- Miễn phí và mã nguồn mở: R là một phần mềm miễn phí và mã nguồn mở, điều này có nghĩa là bất kỳ ai cũng có thể sử dụng và thậm chí cải tiến nó.

- Hệ sinh thái gói mở rộng: R có một hệ sinh thái gói mở rộng phong phú với hàng ngàn gói phần mềm được phát triển bởi cộng đồng người dùng, cung cấp các chức năng và tính năng đa dạng từ phân tích thống kê đến học máy và visualizations.

* Trực quan hóa dữ liệu mạnh mẽ: R cung cấp các gói như ggplot2 cho việc tạo ra các biểu đồ và đồ thị một cách dễ dàng và mạnh mẽ, giúp hiểu rõ hơn về dữ liệu.
* Khả năng xử lý dữ liệu lớn: Mặc dù R ban đầu được phát triển cho việc xử lý dữ liệu nhỏ và trung bình, nhưng các gói như dplyr và data.table cung cấp các công cụ tối ưu hóa cho việc xử lý dữ liệu lớn.
* Cộng đồng lớn và tích cực: R có một cộng đồng người dùng lớn và tích cực, với nhiều tài liệu, diễn đàn trực tuyến và hỗ trợ từ cộng đồng.

\* Nhược điểm:

- Tốc độ thực thi chậm: R thường không hiệu quả trong việc xử lý dữ liệu lớn và có thể chậm so với các ngôn ngữ khác như Python, đặc biệt là khi làm việc với vòng lặp lớn.

- Cú pháp phức tạp: Một số người mới bắt đầu có thể gặp khó khăn trong việc học cú pháp của R, đặc biệt là khi làm việc với các hàm và biểu thức phức tạp.

- Quản lý bộ nhớ kém: R không quản lý bộ nhớ một cách hiệu quả, có thể dẫn đến việc sử dụng bộ nhớ quá mức khi làm việc với dữ liệu lớn.

- Phụ thuộc vào gói mở rộng: Mặc dù hệ sinh thái gói mở rộng của R rất phong phú, nhưng sự phụ thuộc vào các gói này có thể tạo ra các vấn đề về sự ổn định và tương thích.

* 1. **Các gói R được sử dụng trong phân tích dữ liệu âm nhạc.**

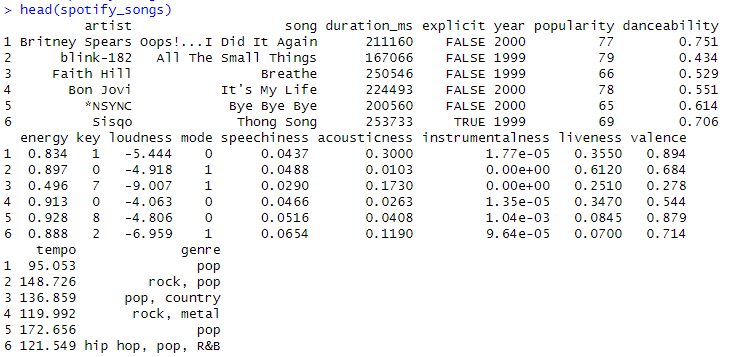
- dplyr: Gói dplyr cung cấp các công cụ mạnh mẽ cho việc xử lý và biến đổi dữ liệu

tidyr: Gói tidyr cung cấp các công cụ cho việc làm sạch và biến đổi dữ liệu trong định dạng "tidy data".

* ggplot2: Gói ggplot2 là một trong những gói mạnh mẽ nhất cho việc tạo ra các biểu đồ và đồ thị trực quan.
* tuneR: Gói tuneR cung cấp các công cụ để đọc, ghi và xử lý âm thanh trong R.
* rspectra: Gói rspectra cung cấp các công cụ cho phân tích và xử lý dữ liệu tín hiệu, bao gồm cả âm thanh.

1. **Giới thiệu về tập dữ liệu.**
   1. **Giới thiệu về dữ liệu.**

* Tập dữ liệu từ kaggle.com cung cấp một số biến để phân tích thăm dò trong đó chúng tôi có thể trực quan hóa kết quả và thực hiện phân tích thống kê trong RStudio bằng ngôn ngữ R. Thông qua quá trình này, chúng tôi hy vọng tìm thấy những hiểu biết sâu sắc và câu trả lời có giá trị cho những gì khiến một số bản nhạc trở nên phổ biến hơn những người khác. Phân tích này sẽ giúp hiểu rõ hơn những hiểu biết độc đáo và cho phép Spotify tạo ra cách phân phối nội dung được tối ưu hóa hơn sẽ hữu ích cho nhóm kỹ thuật của họ,cho phép điều tra thêm, chẳng hạn như xu hướng. Hơn nữa, nghiên cứu cần cung cấp cái nhìn sâu sắc về cách hỗ trợ người dùng tốt hơn, từ đó tăng lợi nhuận và cải thiện trải nghiệm người dùng tổng thể.
* Tổng số hàng: 2000
* Tổng số cột: 18
* Trực quan tập dữ liệu trong RStudio, hiển thị một vài dòng đầu và tổng số cột của tập dữ liệu.



Hình 1. Một vài dòng đầu và tổng số cột của tập dữ liệu

* 1. **Mô tả tập dữ liệu.**

|  |  |
| --- | --- |
| **Thuộc tính** | **Mô tả** |
| Artist | Tên của nghệ sĩ |
| Song | Tên của bài hát |
| duration\_ms | Thời lượng của bản nhạc tính bằng mili giây. |
| Explicit | Nội dung của bài hát chứa một hoặc nhiều tiêu chí có thể bị coi là xúc phạm hoặc không phù hợp với trẻ em. |
| Year | Năm phát hành bài hát |
| Popularity | Mức độ phổ biến của bài hát |
| Danceability | Khả năng khiêu vũ của bài hát. Giá trị 0.0 là ít có khả năng nhảy nhất và 1.0 là dễ nhảy nhất. |
| energy | Năng lượng của bài hát tăng từ 0,0 đến 1,0 |
| key | Các số nguyên ánh xạ tới các cao độ bằng cách sử dụng ký hiệu Lớp cao độ tiêu chuẩn. Ví dụ. 0 = C, 1 = C♯/D♭, 2 = D, v.v. Nếu không tìm thấy khóa nào, giá trị là -1. |
| loudness | Thể hiện âm lượng tổng thể của một bản nhạc tính bằng decibel (dB). Giá trị độ ồn được tính trung bình trên toàn bộ bản nhạc, với các giá trị thường nằm trong khoảng phạm vi -60 và 0 dB. |
| speechiness | Phát hiện sự hiện diện của các từ được nói trong một bản nhạc. |
| acousticness | Phát hiện xem bài hát đó có phải bài hát aucoustic hay không. |
| liveness | Phát hiện sự hiện diện của khán giả trong bản ghi. |
| valence | Thước đo từ 0,0 đến 1,0 mô tả mức độ tích cực mà bản nhạc truyền tải. |
| tempo | Nhịp độ tính bằng nhịp mỗi phút (BPM). Trong thuật ngữ âm nhạc, nhịp độ là tốc độ hoặc nhịp độ của một bản nhạc nhất định và xuất phát trực tiếp từ thời lượng nhịp trung bình. |
| genre | Thể loại của bài hát. |

## **2.3 Mục đích của đề tài**

Thông qua quá trình phân tích dữ liệu này, chúng em hy vọng tìm thấy những hiểu biết sâu sắc và câu trả lời có giá trị cho những gì khiến một số bản nhạc trở nên phổ biến hơn những bài hát khác.

Giúp cho ứng dụng nghe nhạc này nắm bắt được xu hướng, sở thích của người dùng và cho phép Spotify tạo ra cách phân phối nội dung được tối ưu hóa hơn, hỗ trợ người dùng tốt hơn từ đó tăng lợi nhuận và cải thiện trải nghiệm người dùng tổng thể.

1. **Xử lý dữ liệu.**

Dữ liệu tải xuống từ các trang web chính thức thường là dữ liệu thô. Dữ liệu này cần phải được làm sạch để giúp phân tích dữ liệu dễ dàng hơn. Chúng ta cần phải xóa dữ liệu không chính xác, trùng lặp hoặc không đầy đủ. Có thể sử dụng nhiều phương pháp làm sạch khác nhau cho dữ liệu sạch, thay đổi từ tập dữ liệu này sang tập dữ liệu khác. Dưới đây là một số phương pháp làm sạch dữ liệu được sử dụng để làm sạch Bộ dữ liệu Spotify. Dữ liệu đã được làm sạch trong RStudio bằng ngôn ngữ R.

* 1. **Thay thế giá trị Na với giá trị trung vị (median)**

Khi sử dụng hàm tóm tắt, nhóm nhận thấy 106 giá trị NA trong tập dữ liệu cho cột “popularity”. Sau đó ta sẽ thay thế giá trị NA bằng giá trị trung vị.

Mã R bên dưới thực hiện các bước sau:

1. Đầu tiên hàm read.csv này được sử dụng để tải dữ liệu có trong CSV vào khung dữ liệu có tiêu đề “songs\_normalize”.
2. Cuối cùng, bằng cách sử dụng hàm colSums(), chúng ta tổng hợp 94 giá trị NA theo cột. Chúng ta đã thay thế giá trị Na bang giá trị trung bình bằng cách sử dụng na.rm = TRUE.

Code text:

spotify\_songs<-read.csv("/content/drive/MyDrive/Colab Note books/Nam3-HK2/BigData/Project/songs\_normalize.csv")

head(spotify\_songs)

dim(spotify\_songs)

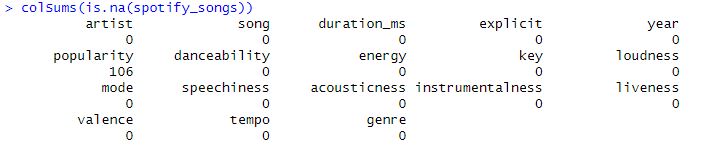
[1] 2000 18

colSums(is.na(spotify\_songs))

spotify\_songs[is.na(spotify\_songs)]<-median(spotify\_songs$popularity, na.rm=TRUE)

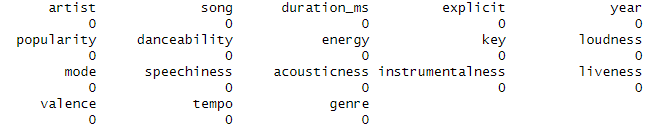
colSums(is.na(spotify\_songs))

* Trước khi điền dữ liệu bị null:



Hình 2. Trước khi điền dữ liệu bị null

* Sau khi điền dữ liệu bị null:



Hình 3. Sau khi điền dữ liệu bị null

**3.2 Xóa các hàng bị trùng lặp**

Các dòng trùng lặp có thể làm cho kết quả không chính xác. Nó cũng gây ra sự gia tăng bộ nhớ không cần thiết.; các hàng và cột trùng lặp phải được loại bỏ khỏi tập dữ liệu. Cần phải kiểm tra cho các bản sao trong bảng excel. Theo đó, chúng tôi đã thực hiện chức năng “unique” trong R studio. Ban đầu ta có 2000 hàng trong cột; sau khi loại bỏ các bản ghi trùng lặp, chúng tôi thấy tổng cộng 1941. Như hình dưới đây là kết quả của việc loại bỏ các bản ghi trùng lặp.

Code Text:

dim(spotify\_songs)

[1] 2000 18

spotify\_songs<-unique(spotify\_songs)

dim(spotify\_songs)

[1] 1941 18

**3.3 Chuyển đổi kiểu dữ liệu từ mili giây sang phút**

Chuyển đổi dữ liệu nhằm mục đích cung cấp hỗ trợ đáng kể cho việc chuyển đổi dữ liệu để phù hợp với yêu cầu. Điều này giúp loại bỏ dữ liệu không liên quan, cho phép phân tích dữ liệu chính xác hơn. Trong ví dụ này, hầu hết người dùng sẽ không thể đo bằng mili giây nhưng quen thuộc với phút và giờ. Ta sẽ chuyển đổi mili giây thành phút để thực hiện kết quả có ý nghĩa hơn đối với người dùng. Đối với trường có tên là “duration\_ms”, ta sẽ sử dụng biến đổi và áp dụng công thức: (duration\_ms/1000)/60 để chuyển đổi từ mili giây sang phút.

Code text:

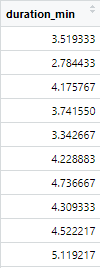
spotify\_songs <- spotify\_songs %>% mutate(duration\_min = (duration\_ms

/ 1000/

+ 60, year)

View(spotify\_songs)

Kết quả:



Hình 4. Kết quả sau khi chuyển dữ liệu từ mili giây sang phút

**3.4 Xóa các cột không liên quan**

Tập dữ liệu Spotify chứa tổng cộng 18 cột. Đối với dự án này, 15 trong số 18 cột là cần thiết. Do đó, một trong những bước để làm sạch dữ liệu là xóa các thuộc tính không cần thiết khỏi tập dữ liệu. Ta sẽ sử dụng mã bên dưới để loại bỏ dữ liệu ngoài phạm vi, chẳng hạn như duration\_ms, key, mode, và instrumentals.

Code text:

dim(spotify\_songs)

[1] 1941 19

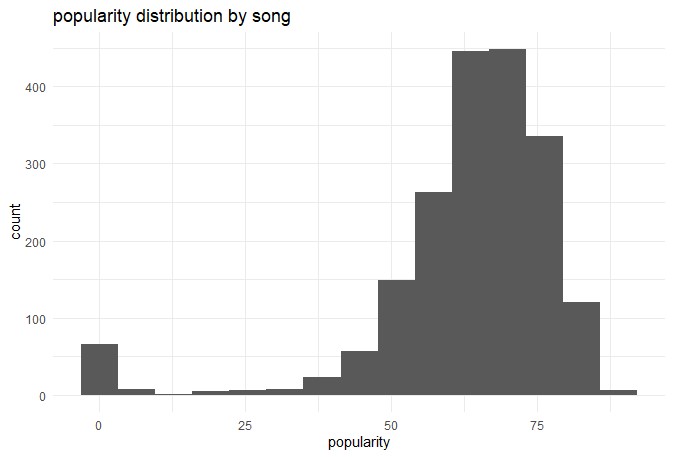
spotify\_songs<-spotify\_songs[,-c(3,9,11,14)]

dim(spotify\_songs)

[1] 1941 15

1. **Đặt câu hỏi để phân tích và trực quan tập dữ liệu.**
   1. **Mức độ phổ biến của bài hát được phân bổ như thế nào? 10 bài hát hàng đầu là gì?**

* Mức độ phổ biến của bài hát



Hình 5. Mức độ phổ biến của bài hát

Code text:

ggplot(spotify\_songs, aes(x = popularity)) +

geom\_histogram(bins = 15) +

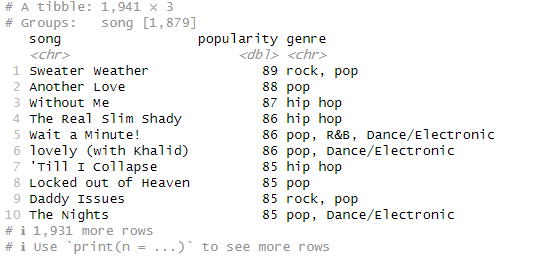
labs(title = "Popularity Distribution by Song") +

theme\_minimal()

* Plot Type: Histogram
* Functions: aes, geom\_histogram, labs, theme\_minimal
* Libraries: ggplot2, tidyverse

Nhận xét: Từ biểu đồ trên ta thấy rằng có trên 50 bài hát có mức độ phổ biến là 0 nghĩa là những bài hát đó rất không được ưa chuộng.

* 10 Bài hát phổ biến hàng đầu ?

****

Hình 6. 10 bài hát phổ biến hàng đầu

Code text:

selected\_data <- spotify\_songs %>%

select(song, popularity, genre)

grouped\_data <- selected\_data %>%

group\_by(song) %>%

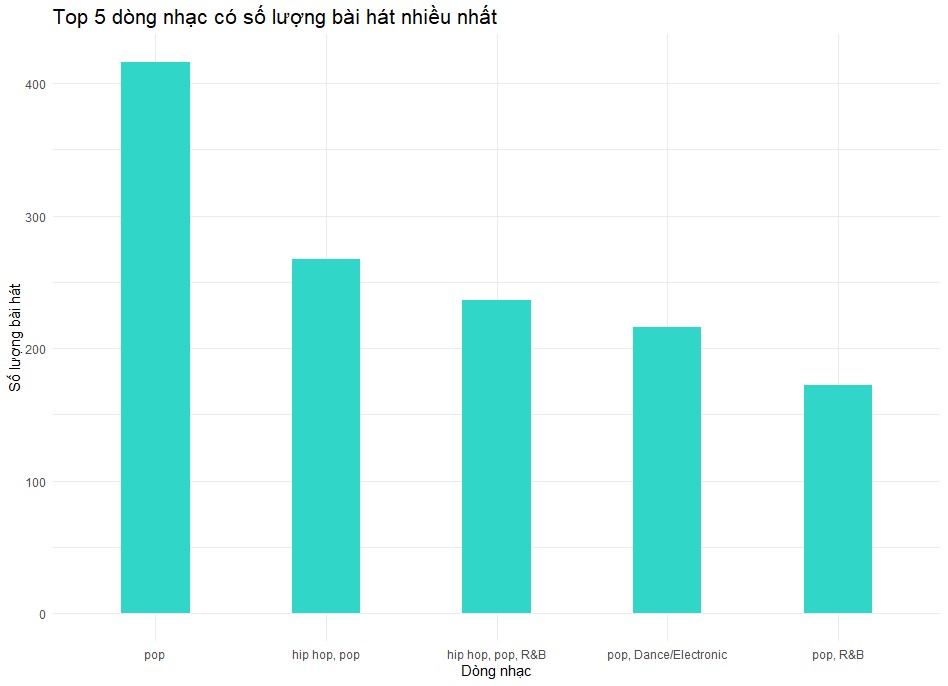
arrange(desc(popularity))

top\_songs <- head(grouped\_data, 10)

print(top\_songs)

Nhận xét: Ta có thể thấy rằng những bài hát được yêu thích nhất là của thể loại “pop”.

* 1. **Top 5 thể loại nhạc phổ biến nhất?**



Hình 7. Top 5 thể loại nhạc phổ biến

Code text:

# Đếm số lượng bài hát theo thể loại

genre <- spotify\_songs %>% count(genre, sort = TRUE, name = "Count")

# Lọc top 5 dòng nhạc có số lượng bài hát nhiều nhất

genreFil <- genre[1:5, ]

# Sắp xếp cột x theo thứ tự từ lớn đến bé

genreFil$genre <- reorder(genreFil$genre, -genreFil$Count)

# Tạo biểu đồ với cột x đã được sắp xếp

ggplot(genreFil, aes(x = genre, y = Count)) +

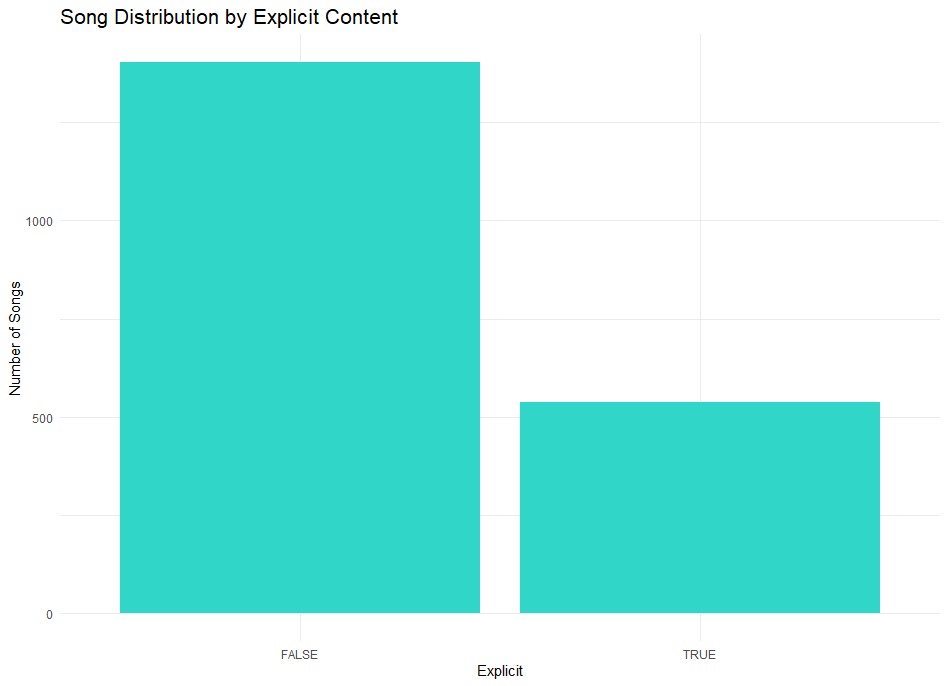
geom\_bar(stat = "identity", width = 0.4, fill = "#30d6c8") +

labs(x = "genre", y = "Number of songs", title = "Top 5 music genres with the most number of songs") +

theme\_minimal() +

theme(plot.title = element\_text(color = "black", size = 16))

* 1. **Số lượng bài hát có nội dung không phù hợp với trẻ em trong các bài hát hàng đầu.**



Hình 8. Số lượng bài hát không phù hợp với trẻ em

Code text:

explicit <- spotify\_songs %>% count(explicit, name = "Count")

# Tạo biểu đồ cột

ggplot(explicit, aes(x = explicit, y = Count)) +

geom\_col(fill = "#30d6c8") +

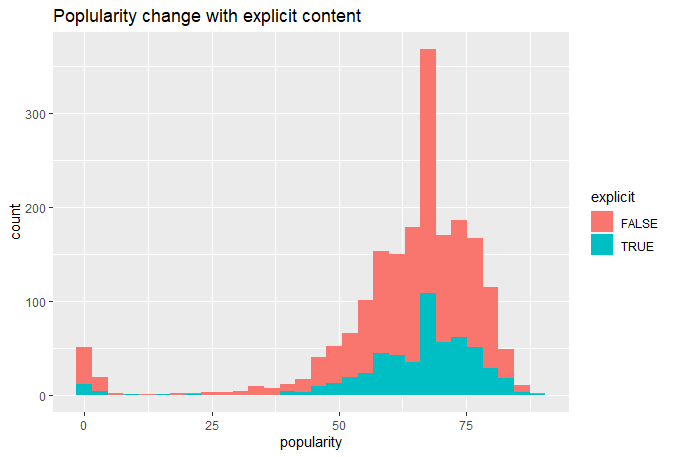
labs(x = "Explicit", y = "Number of Songs", title = "Song Dist ribution by Explicit Content") +

theme\_minimal() +

theme(plot.title = element\_text(color = "black", size = 16))

Nhận xét: Hầu hết các bản hit hàng đầu đều có nội dung bài hát lành mạnh.

* 1. **Mức độ phổ biến của bài hát thay đổi như thế nào với nội dung nhạy cảm? Nội dung nhạy cảm có tác động tích cực, tiêu cực hoặc trung lập đến mức độ phổ biến của bài hát không?**



Hình 9. Mức độ phổ biến của bài hát

Code text:

library(ggplot2)

popularity\_explicit\_content<- spotify\_songs %>%

+ ggplot(aes(x=popularity, fill=explicit))+

+ geom\_histogram()+

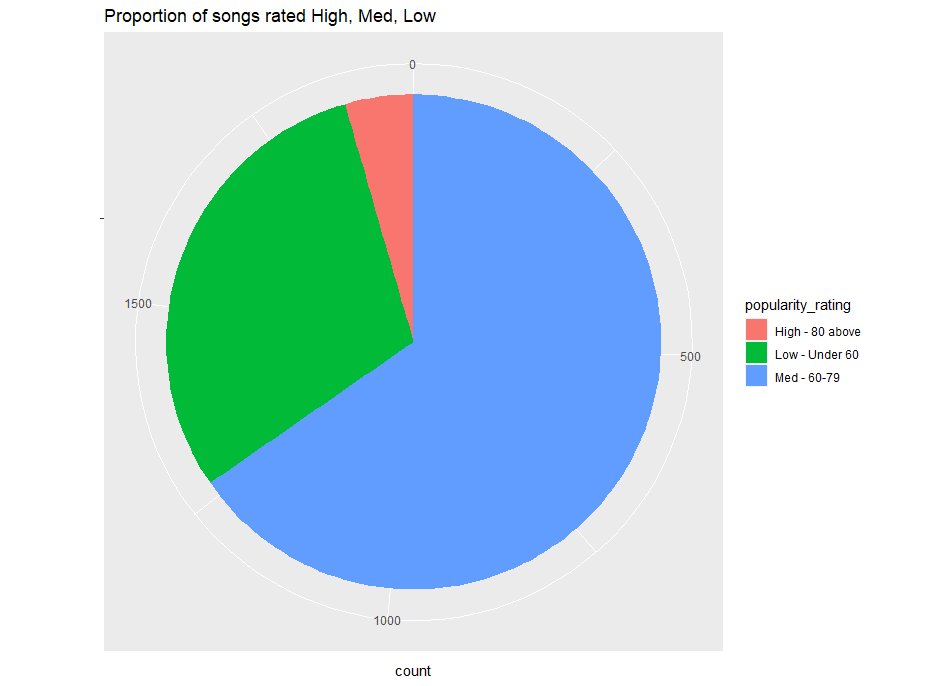
+ ggtitle("Poplularity change with explicit content")

popularity\_explicit\_content

* Plot Type: histogram
* Functions: aes, geom\_histogram, labs, theme\_minimal
* Libraries: ggplot2, tidyverse

Nhận xét: Nội dung nhạy cảm explicit = "TRUE" khi lời bài hát hoặc nội dung của bài hát chứa một hoặc nhiều tiêu chí không phù hợp với trẻ em. Trong hình ảnh trực quan trên, ta thấy rằng bài hát chứa và không chứa nội dung nhạy cảm đều giống với sự phân phối của mức độ phổ biến của bài hát 🡪 Kết luận: ta có thể kết luận rằng các bài hát “explicit = TRUE ít hoặc không ảnh hưởng đến mức độ phổ biến của các bài hát so với các bài hát “explicit = FALSE”.

* 1. **Tỷ lệ bài hát được đánh giá cao, trung bình hoặc thấp là bao nhiêu ?**
* Trong ví dụ này, ta phải tạo các biến category “high”, “medium” và “low” từ popularity. Ta tạo một cột mới có tên là 'popularity\_rated' và gán từng giá trị mức độ phổ biến vào một trong ba danh mục bằng cách sử dụng câu lệnh if lồng nhau. “as.factor” được sử dụng để chuyển đổi giá trị số thành kiểu dữ liệu phân loại cho biểu đồ và biểu đồ hình tròn bên dưới.



Hình 10. Tỷ lệ bài hát được đánh giá

Code text:

summary(spotify\_songs$popularity)

head(spotify\_songs$popularity,10)

spotify\_songs$popularity\_rating <- as.factor(

ifelse(spotify\_songs$popularity<=60, "Low - Under 60",

ifelse(spotify\_songs$popularity<=80, "Med - 60-79",

ifelse(spotify\_songs$popularity>80,'High - 80 above',"High - 80 above")

))

)

head(spotify\_songs$popularity\_rating, 10)

ggplot(spotify\_songs, aes(x = popularity\_rating))+

geom\_bar() + labs(title="Popularity of songs rated High, Med, Low") + theme\_minimal()

ggplot(spotify\_songs,aes(x=factor(""),fill=popularity\_rating))+

geom\_bar()+coord\_polar(theta = "y") +

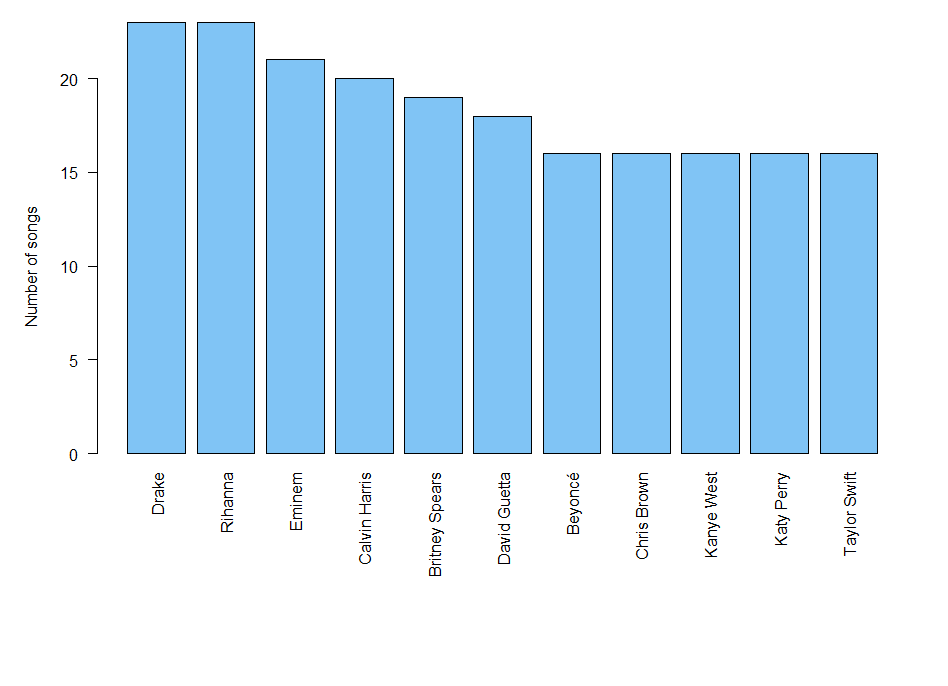
scale\_x\_discrete("") +

labs(title= "Proportion of songs rated High, Med, Low")

* Plot Type: Pie graph
* Functions: summary, as. factor, ifelse, aes, ggplot, geom\_bar, labs, theme\_minimal(), fill, coord\_polar, scale\_x\_discrete
* Libraries: ggplot2, tidyverse, scales

Nhận xét: Chúng ta có thể thấy từ biểu đồ pie ở trên có ít bài hát được đánh giá trên 80 so với các nhóm khác. Nhóm bài hát được đánh giá trung bình là đông đảo nhất.

* 1. **15 nghệ sũ phát hành nhiều ca khúc nhất từ 2000 – 2020 ?**



Hình 11. Biểu đồ 15 nghệ sĩ phát hành nhiều ca khúc nhất

Code text:

Artist\_Popular <- spotify\_songs %>% count(artist, sort = TRUE, name =

+ "Count")

Artist\_Fil <- Artist\_Popular %>% filter(Count >= 15)

par(mar = c(12, 5, 4, 2)+ 0.1)

barplot(Artist\_Fil$Count,

+ ylab = "Number of songs",

+ col = "#80C4F5",

+ names.arg= Artist\_Fil$artist,

+ width= 0.01,

+ ylim = c(0,20),

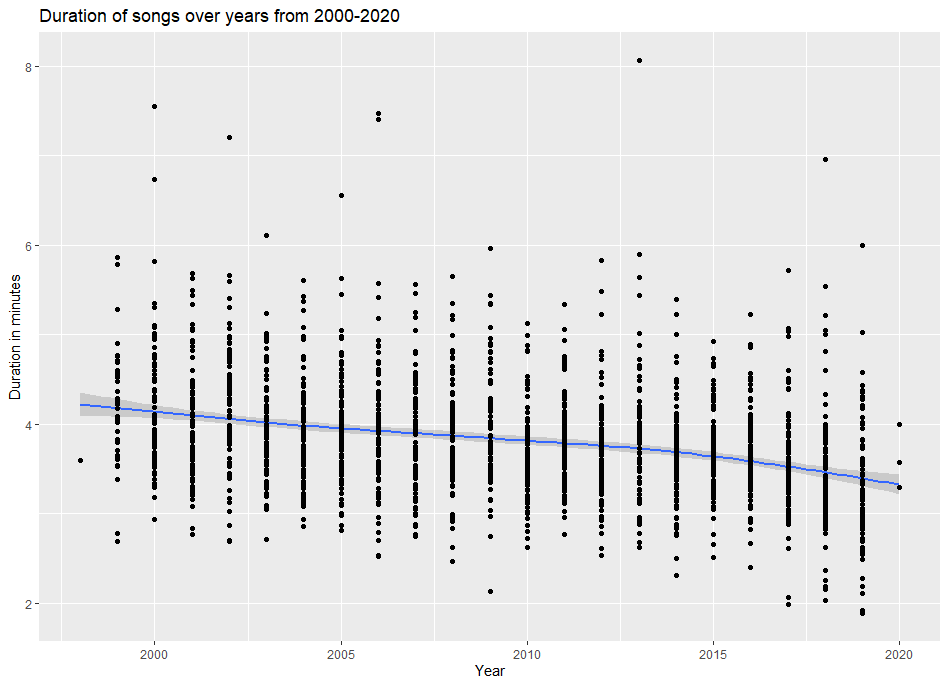
+ las = 2

+ )

* Plot Type: Bar plot(Horizontal)
* Functions: par, ylab, las, mar, ylim, filter, width, barplot
* Libraries: ggplot2

Nhận xét: Biểu đồ thanh hiển thị các nghệ sĩ nổi tiếng nhất với hơn 20 bài hát trở lên trong tập dữ liệu trong khoảng thời gian từ 2000-2020. Những nghệ sĩ có bài hát được yêu thích nhất là Drake và Rihanna, với hơn 20 bài hát. Điều này ngụ ý rằng họ là nghệ sĩ nổi tiếng nhất trên Spotify. Theo sau họ là Eminem, có số điểm là 20 với Calvin Harris. Thông tin đồ họa này cung cấp cho các nghệ sĩ rất nhiều thông tin hữu ích có thể giúp họ tăng lượng người hâm mộ và số lượt nghe bài hát của họ.

* 1. **Độ dài của bài hát có thay đổi qua các năm không ?**



Hình 12. Biểu đồ độ dài của bài hát thay đổi qua các năm

Code text:

song\_duration<- transmute(spotify\_songs, duration\_min,

+ year)

ggplot(song\_duration, aes(x=year, y=duration\_min)) +

+ labs(title = "Duration of songs over years from 2000-2020") +

+ labs(x="Year") + labs(y= "Duration in minutes") +

+

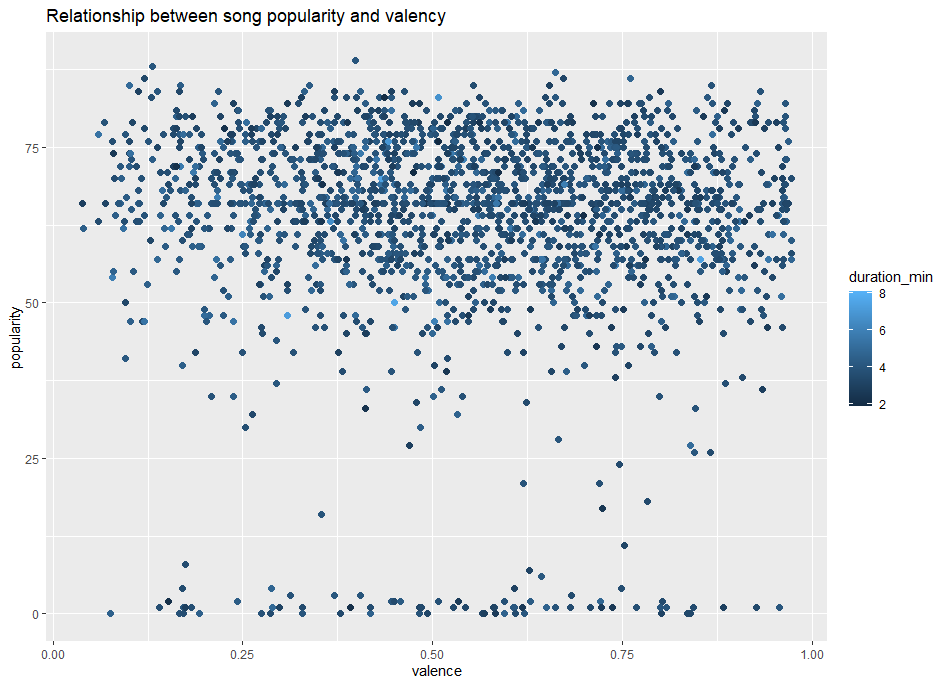
+ geom\_smooth() +

+ geom\_point()

* Plot Type: Scatter Trend Line graph
* Functions: transmute, aesthetics(aes), ggplot, labs, geom\_point, geom\_point(), title
* Libraries: ggplot, tidyverse

Nhận xét: Biểu đồ cho thấy sự thay đổi về thời lượng của các bài hát qua các năm. Theo biểu đồ đường xu hướng phân tán ở trên, thời lượng trung bình của các bài hát đã giảm theo thời gian, thể hiện rõ xu hướng đi xuống. Đường xu hướng màu xanh biểu thị sự hồi quy đi xuống của thời lượng bài hát; Trước năm 2005, thời lượng của các bài hát là hơn 4 phút, trong khi đó, sau năm 2005, thời lượng dường như giảm dần. Nó đang giảm từ mức trung bình 4 phút vào đầu những năm 2000 xuống mức trung bình gần 3,5 phút vào năm 2019. Điều này là do thế hệ trẻ có thể có xu hướng nghe các bài hát có thời lượng ngắn hơn và chuyển sang bản nhạc tiếp theo trùng với thời gian tầm quan trọng ngày càng tăng của số liệu AI và đề xuất được cá nhân hóa. Phân tích này có thể hữu ích để nghệ sĩ hoặc tác giả bài hát biết được sở thích của người dùng Spotify.

* 1. **Mức độ phổ biến và giá trị của bài hát tương quan như thế nào với thời lượng?**



Hình 13. Biểu đồ thể hiện mức độ phổ biến và giá trị của bài hát

Code text:

ggplot(spotify\_songs, aes(x=valence, y=popularity, color=duration\_min)) +

+ geom\_point(size=2) +

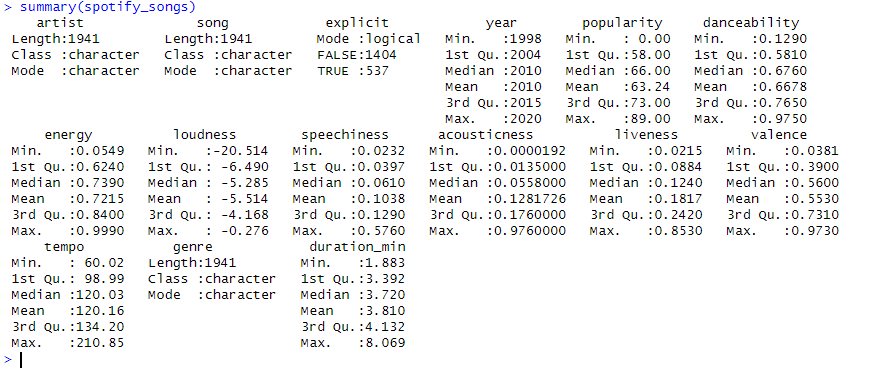
+ ggtitle("Relationship between song popularity and valency")

* Plot Type: Scatter Plot
* Functions: aesthetics(aes), ggplot, ggtitle, geom\_point, color
* Libraries: ggplot2, tidyverse

Nhận xét: Biểu đồ phân tán trực quan ở trên cho thấy mối quan hệ giữa mức độ phổ biến của bài hát và giá trị của chúng. “valency” là tính tích cực của bài hát - số “valency” càng lớn thì bài hát càng tích cực. Như đã trình bày, những bài hát tích cực nhất không nhất thiết phải là những bài hát phổ biến nhất. Phần lớn nằm trong khoảng từ 50-85 trên thang mức độ phổ biến bất kể giá trị. Tuy nhiên, thời lượng là một tiêu chí hợp lệ về mức độ phổ biến vì các bài hát dài nhất (màu xanh nhạt) thấp hơn cả về mức độ phổ biến và giá trị.

1. **Tổng hợp thống kê (Statistical Summary).**

Chúng ta có thể tính giá trị tối thiểu (min), tứ phân vị, trung vị (median), trung bình (mean), và tối đa (max) cho tất cả các biến số của tập dữ liệu cùng một lúc bằng cách sử dụng hàm **summary()**. Ảnh chụp màn hình sau đây hiển thị ảnh chụp màn hình của toàn bộ tập dữ liệu cho hàm tóm tắt.



Hình 14. Toàn bộ dữ liệu cho hàm tóm tắt

* 1. **Sử dụng hàm summary() cho cột valance.**

Code text:

summary(spotify\_songs$valence)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.0381 0.3900 0.5600 0.5530 0.7310 0.9730

min(spotify\_songs$valence)

[1] 0.0381

max(spotify\_songs$valence)

[1] 0.973

mean(spotify\_songs$valence)

[1] 0.5529662

median(spotify\_songs$valence)

[1] 0.56

sd(spotify\_songs$valence)

[1] 0.2208454

Nhận xét: Dựa trên bản tóm tắt thống kê, giá trị “valency” tối thiểu là 0,0381. Dựa trên tập dữ liệu, valency thấp hơn gây ra những cảm xúc tiêu cực, chẳng hạn như buồn bã, trầm cảm và tức giận. Mặt khác, số valency tối đa 0,973 biểu thị tính tích cực mà bài hát truyền tải. Chúng ta có thể suy ra rằng phạm vi là từ 0 đến 1,0. Giá trị valency trung bình là 0,553, có thể so sánh được tới giá trị trung vị là 0,56. Điều này có nghĩa là sự cân bằng của cả bài hát tích cực và tiêu cực từ tập dữ liệu với độ lệch trái nhẹ.

Cuối cùng, độ lệch chuẩn 0,221 cho chúng ta biết rằng các điểm dữ liệu được trải đều, trong đó 68% số điểm dữ liệu nằm trong khoảng từ 0,21 đến 0,77.6.2.

* 1. **Sử dụng hàm summary() cho cột duration\_min:**

Code text:

summary(spotify\_songs$duration\_min)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.883 3.392 3.720 3.810 4.132 8.069

min(spotify\_songs$duration\_min)

[1] 1.883333

max(spotify\_songs$duration\_min)

[1] 8.0691

median(spotify\_songs$duration\_min)

[1] 3.719767

mean(spotify\_songs$duration\_min)

[1] 3.809916

sd(spotify\_songs$duration\_min)

[1] 0.6541633

Nhận xét: “Duration\_min” là thời lượng của bất kỳ bài hát cụ thể nào tính bằng phút trong tập dữ liệu Spotify. Tóm tắt thống kê cho thấy thời lượng tối thiểu của bài hát là 1,88 phút. Độ dài bài hát tối đa là 8,069 phút. Chúng ta có thể nói rằng thời lượng của bài hát là khoảng 2 đến 8 phút. Thời lượng trung bình của các bài hát là 3,81 phút, tương đương với giá trị trung vị là 3,72. Vì không có sự khác biệt đáng kể về thời lượng của bài hát giữa giá trị trung bình và giá trị trung bình nên chúng tôi có thể kết luận rằng không có nhiều ngoại lệ. Có một chút sai lệch bên phải.

Cuối cùng, độ lệch chuẩn cho chúng ta biết rằng thời lượng của bài hát thay đổi 0,65. Điều này có nghĩa là 65% bài hát có độ dài trong vòng 0,65 phút (hoặc 39 giây) so với mức trung bình là 3,8 (hoặc 3 phút 48 giây). Điều này cho thấy thời lượng của bài hát không được dàn trải rộng rãi.

* 1. **Sử dụng hàm summary() cho cột tempo.**

Code text:

summary(spotify\_songs$tempo)

Min. 1st Qu. Median Mean 3rd Qu. Max.

60.02 98.99 120.03 120.16 134.20 210.85

min(spotify\_songs$tempo)

[1] 60.019

max(spotify\_songs$tempo)

[1] 210.851

median(spotify\_songs$tempo)

[1] 120.028

mean(spotify\_songs$tempo)

[1] 120.1584

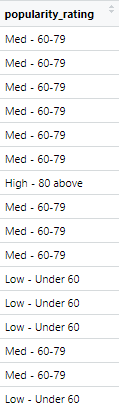
sd(spotify\_songs$tempo)

[1] 26.99047

Nhận xét: Dựa trên bản tóm tắt thống kê, nhịp tối thiểu là 60.01, có nghĩa là nhịp đập đó chính xác một lần mỗi giây. Dựa trên tập dữ liệu, nhịp độ thấp hơn sẽ kích hoạt bài hát ở tốc độ chậm và biểu thị sự êm dịu trong bài hát với giá trị tối thiểu. Mặt khác, số nhịp độ tối đa là 210.85 cho biết nhịp độ nhanh nhất dựa trên tập dữ liệu. Chúng ta có thể suy ra phạm vi đó là từ 60.01 đến 210.85. Nhịp độ trung bình của bài hát là 120.16, nhanh gấp đôi, với hai nhịp mỗi giây. Giá trị trung bình khá tương đương với giá trị trung vị là 120.02, điều này có nghĩa là có sự cân bằng giữa cả nhịp độ nhanh và nhịp độ chậm. Thống kê này được sử dụng trong ứng dụng mà thời lượng âm nhạc, chẳng hạn như trong ngành công nghiệp âm nhạc, phải hoàn toàn chính xác.

Cuối cùng, khoảng thời gian tiêu chuẩn là 26,99 cho chúng ta biết rằng các điểm dữ liệu được trải đều và không tập trung chặt chẽ xung quanh giá trị trung bình.

1. **Tạo các cột mới.**
   1. **Tạo một cột mới với mức độ phổ biến (popularity\_rating) từ biến số đến biến phân loại.**



Code text:

Hình 15. Cột mới với popularity\_rating từ biến số đến biến phân loại

f1 <- function(spotify\_songs)

+ {

+ spotify\_songs$popularity\_rating <- as.factor(

+ ifelse(spotify\_songs$popularity <= 60, "Low - Under 60",

+ ifelse(spotify\_songs$popularity <= 80, "Med - 60-79",

+ "High - 80 above"))

+ )

+ return(spotify\_songs)

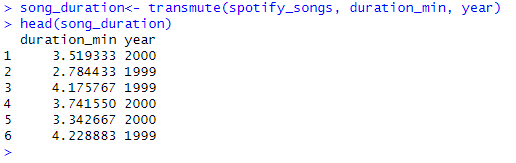
+ }

spotify\_songs <- f1(spotify\_songs)

spotify\_songs$popularity\_rating

Nhận xét: Được sử dụng để tạo hình ảnh trực quan về tỷ lệ các bài hát được xếp hạng cao (high), trung bình (med) và thấp (low).

* 1. **Ta tạo cột mới là “duration\_min” và chuyển từ miliseconds (mili giây) sang minutes (phút). Sau đó, ta nhóm cột mới “duration\_min” với cột “year” thành dữ liệu mới “song\_duration”.**



Hình 16. Cột mói của "duration\_min"

Code text:

> song\_duration<- transmute(spotify\_songs, duration\_min, year)

> head(song\_duration)

Nhận xét: Điều này sẽ được sử dụng cho mục đích trực quan hóa để làm cho các đơn vị dễ nhận biết và dễ hiểu hơn đối với người dùng.

1. **Kết luận.**

* Dòng nhạc pop là dòng nhạc phổ biến và thành công nhất trên thế giới.
* Nghệ sĩ có nhiều bài hát và nổi tiếng nhất là Drake và Rihanna.
* Hầu hết các bài hát đều được đánh giá ở mức độ trung bình
* Không có mối quan hệ quan trọng nào giữa giá trị của bài hát và mức độ phổ biến.
* Thời lượng của bài hát ngày càng ngắn hơn.

**Tài liệu tham khảo.**

1. https://www.kaggle.com/datasets/leonardopena/top-spotify-songs-from-20102019-by-year/code (Truy cập lần cuối 20/3/2024)

2. https://www.kaggle.com/code/victorzhang412/finalpro(Truy cập lần cuối 20/3/2024)

3. Tài liệu của thầy PSG.TS Nguyễn Thanh Bình HUFLIT

4. <https://viblo.asia/p/r-la-gi-nhung-ly-do-ban-nen-bat-dau-hoc-ngon-ngu-lap-trinh-r-6J3ZgBRgKmB> (Truy cập lần cuối 21/3/2024)

**Phụ lục code demo**

**1. Đọc dữ liệu**

spotify\_songs<-read.csv("/content/drive/MyDrive/Colab Notebooks/Nam3-HK2/BigData/Project/songs\_normalize.csv")

head(spotify\_songs)

dim(spotify\_songs)

**2. Làm sạch dữ liệu**

**2.1. Thay thế giá trị Na với giá trị trung vị (median).**

colSums(is.na(spotify\_songs))

spotify\_songs[is.na(spotify\_songs)]<-median(spotify\_songs$popularity,na.rm=TRUE)

colSums(is.na(spotify\_songs))

**2.2. Xóa các hàng bị trùng lặp**

dim(spotify\_songs)

spotify\_songs<-unique(spotify\_songs)

dim(spotify\_songs)

**2.3. Chuyển đổi kiểu dữ liệu từ mili giây sang phút.**

spotify\_songs <- spotify\_songs %>% mutate(duration\_min = (duration\_ms / 1000) / 60)

head(spotify\_songs)

**2.4. Xóa các cột không liên quan.**

dim(spotify\_songs)

spotify\_songs<-spotify\_songs[,-c(3,9,11,14)]

dim(spotify\_songs)

**3. Phân tích và trực quan tập dữ liệu**

**3.1. Mức độ phổ biến của bài hát được phân bổ như thế nào? 10 bài hát hàng đầu là gì?**

ggplot(spotify\_songs, aes(x = popularity)) +

geom\_histogram(bins = 15) +

labs(title = "Popularity Distribution by Song") +

theme\_minimal()

**10 bài hát phổ biến hàng đầu?**

selected\_data <- spotify\_songs %>%

select(song, popularity, genre)

grouped\_data <- selected\_data %>%

group\_by(song) %>%

arrange(desc(popularity))

top\_songs <- head(grouped\_data, 10)

print(top\_songs)

**3.2. Top 5 thể loại nhạc phổ biến nhất.**

# Đếm số lượng bài hát theo thể loại

genre <- spotify\_songs %>% count(genre, sort = TRUE, name = "Count")

# Lọc top 5 dòng nhạc có số lượng bài hát nhiều nhất

genreFil <- genre[1:5, ]

# Sắp xếp cột x theo thứ tự từ lớn đến bé

genreFil$genre <- reorder(genreFil$genre, -genreFil$Count)

# Tạo biểu đồ với cột x đã được sắp xếp

ggplot(genreFil, aes(x = genre, y = Count)) +

geom\_bar(stat = "identity", width = 0.4, fill = "#30d6c8") +

labs(x = "genre", y = "Number of songs", title = "Top 5 music genres with the most number of songs") +

theme\_minimal() +

theme(plot.title = element\_text(color = "black", size = 16))

**3.3. Số lượng bài hát có nội dung không phù hợp với trẻ em trong các bài hát hàng đầu.**

explicit <- spotify\_songs %>% count(explicit, name = "Count")

# Tạo biểu đồ cột

ggplot(explicit, aes(x = explicit, y = Count)) +

geom\_col(fill = "#30d6c8") +

labs(x = "Explicit", y = "Number of Songs", title = "Song Distribution by Explicit Content") +

theme\_minimal() +

theme(plot.title = element\_text(color = "black", size = 16))

**3.4. Mức độ phổ biến của bài hát thay đổi như thế nào với nội dung nhạy cảm? Nội dung nhạy cảm có tác động tích cực, tiêu cực hoặc trung lập đến mức độ phổ biến của bài hát không?**

library(ggplot2)

popularity\_explicit\_content<- spotify\_songs %>%

ggplot(aes(x=popularity, fill=explicit))+

geom\_histogram()+

ggtitle("Poplularity change with explicit content")

popularity\_explicit\_content

**3.5. Tỷ lệ bài hát được đánh giá cao, trung bình hoặc thấp là bao nhiêu?**

summary(spotify\_songs$popularity)

head(spotify\_songs$popularity,10)

spotify\_songs$popularity\_rating <- as.factor(

ifelse(spotify\_songs$popularity<=60, "Low - Under 60",

ifelse(spotify\_songs$popularity<=80, "Med - 60-79",

ifelse(spotify\_songs$popularity>80,'High - 80 above',"High - 80 above")

))

)

head(spotify\_songs$popularity\_rating, 10)

ggplot(spotify\_songs, aes(x = popularity\_rating))+

geom\_bar() + labs(title="Popularity of songs rated High, Med, Low") + theme\_minimal()

ggplot(spotify\_songs,aes(x=factor(""),fill=popularity\_rating))+

geom\_bar()+coord\_polar(theta = "y") +

scale\_x\_discrete("") +

labs(title= "Proportion of songs rated High, Med, Low")

**3.6. 15 nghệ sĩ phát hành nhiều ca khúc nhất từ 2000 – 2020?**

Artist\_Popular <- spotify\_songs %>% count(artist, sort = TRUE, name =

"Count")

Artist\_Fil <- Artist\_Popular %>% filter(Count >= 15)

par(mar = c(12, 5, 4, 2)+ 0.1)

barplot(Artist\_Fil$Count,

ylab = "Number of songs",

col = "#80C4F5",

names.arg= Artist\_Fil$artist,

width= 0.01,

ylim = c(0,20),

las = 2

)

**3.7. Độ dài của bài hát có thay đổi qua các năm không?**

song\_duration<- transmute(spotify\_songs, duration\_min,

year)

ggplot(song\_duration, aes(x=year, y=duration\_min)) +

labs(title = "Duration of songs over years from 2000-2020") +

labs(x="Year") + labs(y= "Duration in minutes") +

geom\_smooth() +

geom\_point()

**3.8. Mức độ phổ biến và giá trị của bài hát tương quan như thế nào với thời lượng?**

ggplot(spotify\_songs, aes(x=valence, y=popularity, color=duration\_min)) +

geom\_point(size=2) +

ggtitle("Relationship between song popularity and valency")

**4. Tổng hợp thống kê (Statistical Summary)**

summary(spotify\_songs)

**4.1. Sử dụng hàm summary() cho cột valence.**

summary(spotify\_songs$valence)

sd(spotify\_songs$valence)

**4.2. Sử dụng hàm summary() cho cột duration\_min:**

summary(spotify\_songs$duration\_min)

sd(spotify\_songs$duration\_min)

**4.3. Sử dụng hàm summary() cho cột tempo**

summary(spotify\_songs$tempo)

sd(spotify\_songs$tempo)

**5. Tạo các cột mới.**

**5.1. Câu lệnh này được sử dụng để tạo một cột mới với mức độ phổ biến (popularity\_rating) từ biến số đến biến phân loại được sử dụng để tạo hình ảnh trực quan về tỷ lệ các bài hát được xếp hạng cao (high), trung bình (med) và thấp (low).**

f1 <- function(spotify\_songs) +

{

spotify\_songs$popularity\_rating <- as.factor(

ifelse(spotify\_songs$popularity <= 60, "Low - Under 60",

ifelse(spotify\_songs$popularity <= 80, "Med - 60-79",

"High - 80 above"))

)

return(spotify\_songs)

}

spotify\_songs <- f1(spotify\_songs)

head(spotify\_songs$popularity\_rating)

head(spotify\_songs)