midterm project

Danya Zhang

2022-12-08

R Markdown

Reading in data

```
setwd("/Users/dz/Documents/MSSP/GitHub/MA678 midterm project/Midterm-project")
top100 <- read.csv("Spotify 2010 - 2019 Top 100.csv")
top100 <- top100[-c(1001:1003), ]  #last 3 rows NA
```

Cleaning data

```
subgenre_df <- as.data.frame(table(top100$subgenre))
# rename top.genre column to subgenre
names(top100)[names(top100) == "top.genre"] <- "subgenre"

# divide into 10 general categories
pop_rows <- grep(paste(c("pop", "neo mellow", "talent show", "indietronica",
    "adult standards", "boy band", "bubblegum", "idol"), collapse = "|"),
    top100$subgenre, ignore.case = TRUE)
hiphop_rows <- grep(paste(c("hip hop", "rap", "trap", "g funk", "uk drill"),
    collapse = "|"), top100$subgenre, ignore.case = TRUE)
rock_rows <- grep(paste(c("rock", "permanent wave", "icelandic indie",
    "emo"), collapse = "|"), top100$subgenre, ignore.case = TRUE)
country_rows <- grep("country", top100$subgenre, ignore.case = TRUE)
latin_rows <- grep(paste(c("latin", "reggae"), collapse = "|"), top100$subgenre,
    ignore.case = TRUE)
randb_rows <- grep(paste(c("soul", "r&b"), collapse = "|"), top100$subgenre,
    ignore.case = TRUE)
edm_rows <- grep(paste(c("house", "grime", "edm", "australian dance", "tronica",
    "dancefloor dnb", "french shoegaze", "big room", "techno", "electro",
    "brostep", "complextro", "alternative dance"), collapse = "|"), top100$subgenre,
    ignore.case = TRUE)
metal_rows <- grep("metal", top100$subgenre, ignore.case = TRUE)

# make new column for parent genre 10 genres
top100$genre <- ""
top100 <- top100[, c(1, 2, 18, 3:17)]
top100[pop_rows, 3] <- "pop"
top100[hiphop_rows, 3] <- "hip hop"
top100[rock_rows, 3] <- "rock"
top100[country_rows, 3] <- "country"
```

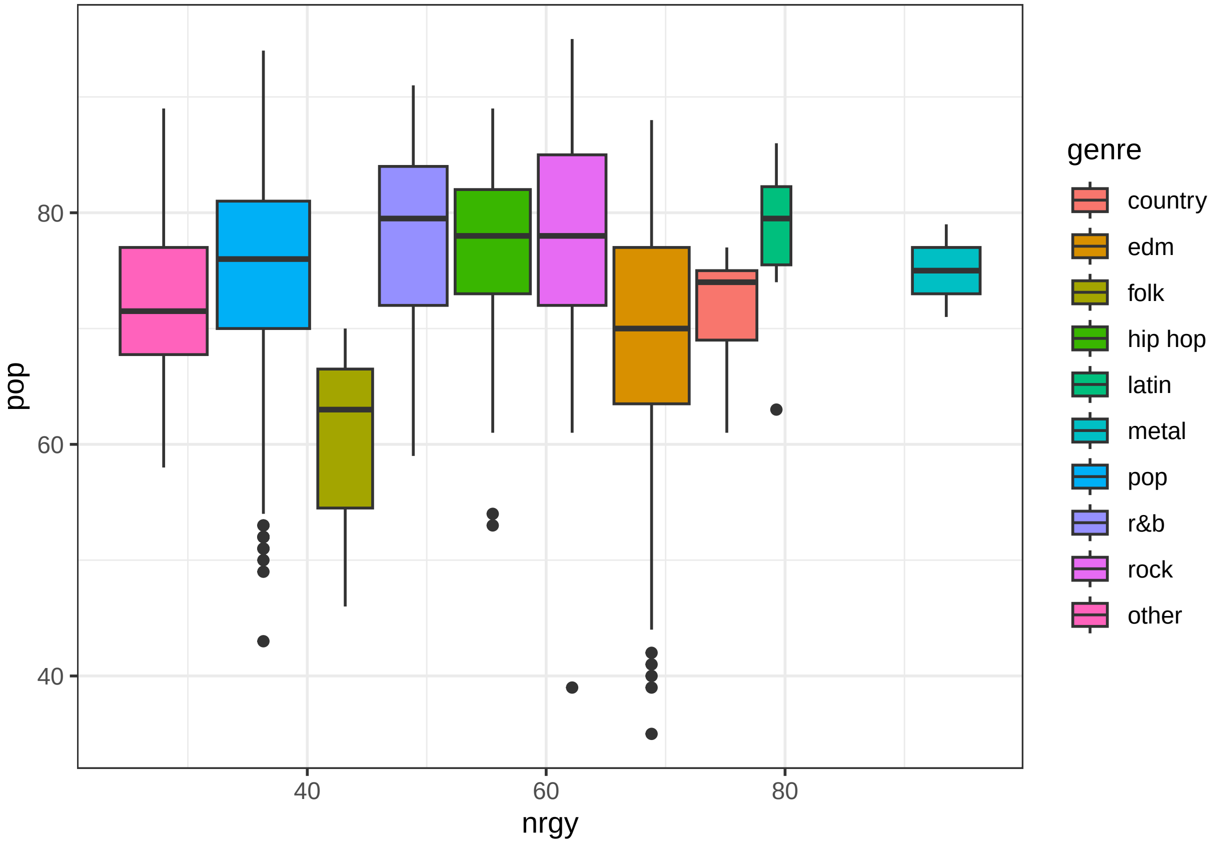
```
top100[latin_rows, 3] <- "latin"
top100[c(21, 177, 111), 3] <- "folk"
top100[randb_rows, 3] <- "r&b"
top100[edm_rows, 3] <- "edm"
top100[metal_rows, 3] <- "metal"
top100$genre <- sub("^$", "other", top100$genre)
```

##Visualizations

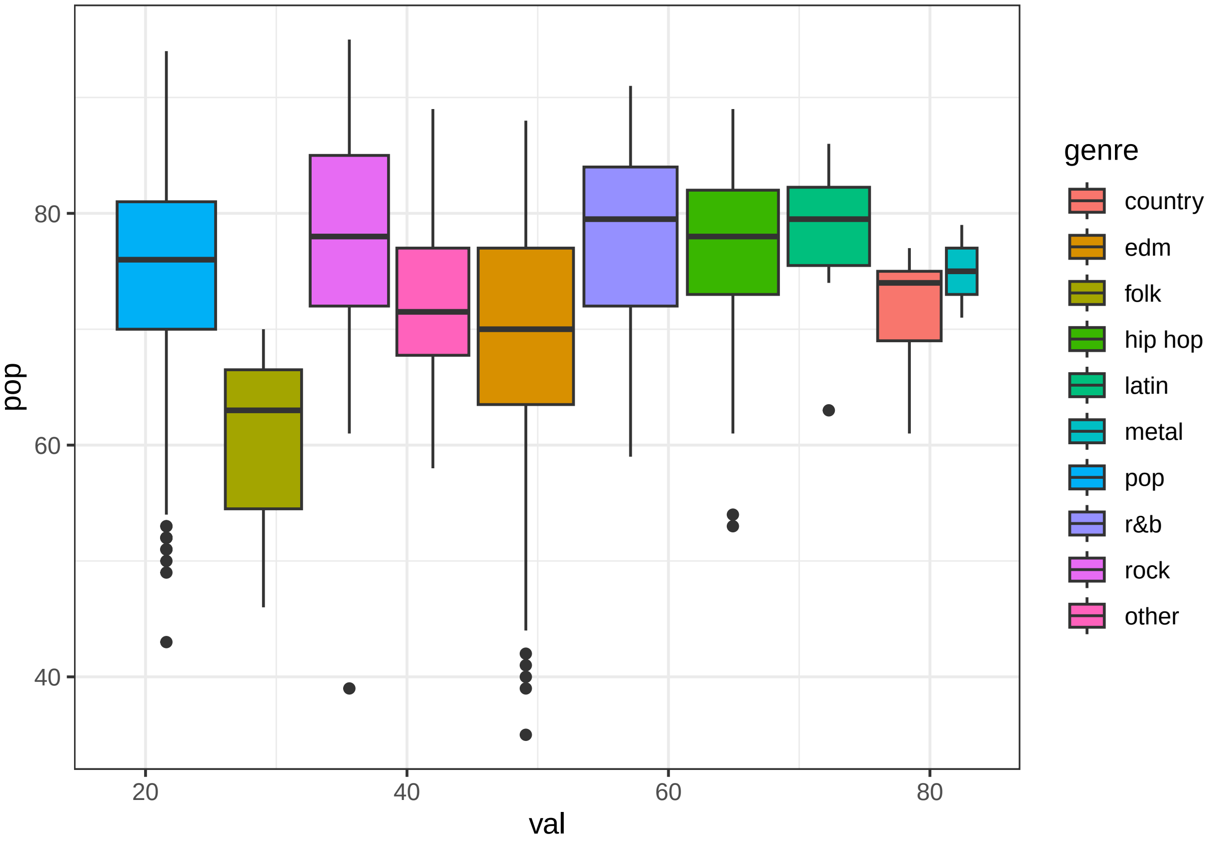
```
# boxplot grouped by genre for popularity vs energy
library(ggplot2)
library(dplyr)
```

```
##
## 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
```

```
genres <- sort(unique(top100$genre))
genres <- c(genres[1:6], genres[8:10], genres[7])
top100$genre <- factor(top100$genre, levels = genres)
top100 %>%
    ggplot(mapping = aes(x = nrgy, y = pop, fill = genre)) + geom_boxplot() +
    scale_fill_discrete(breaks = genres) + theme_bw()
```



```
# boxplot grouped by genre for positivity vs energy
top100 %>%
    ggplot(mapping = aes(x = val, y = pop, fill = genre)) + geom_boxplot() +
    scale_fill_discrete(breaks = genres) + theme_bw()
```



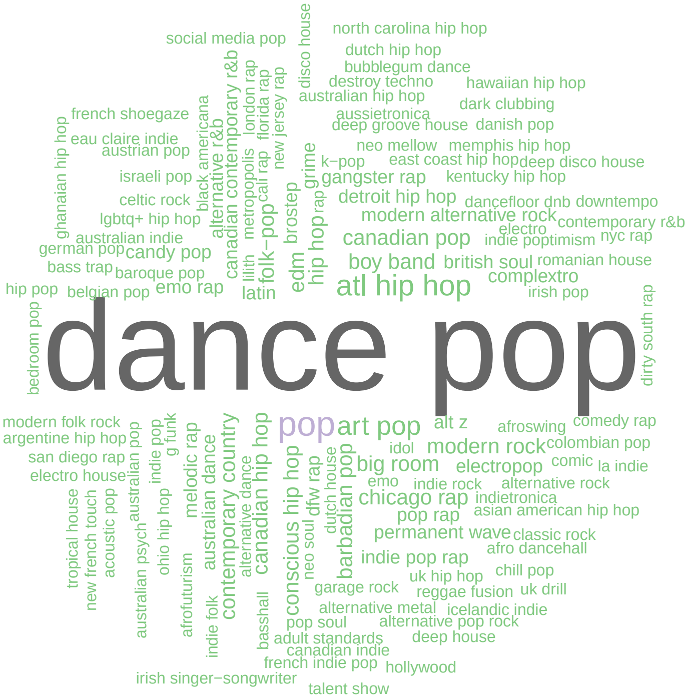
```
# boxplots demonstrate that certain characteristics could vary by
# genre and influnece popularity
```

#Wordcloud

```
# subgenres
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

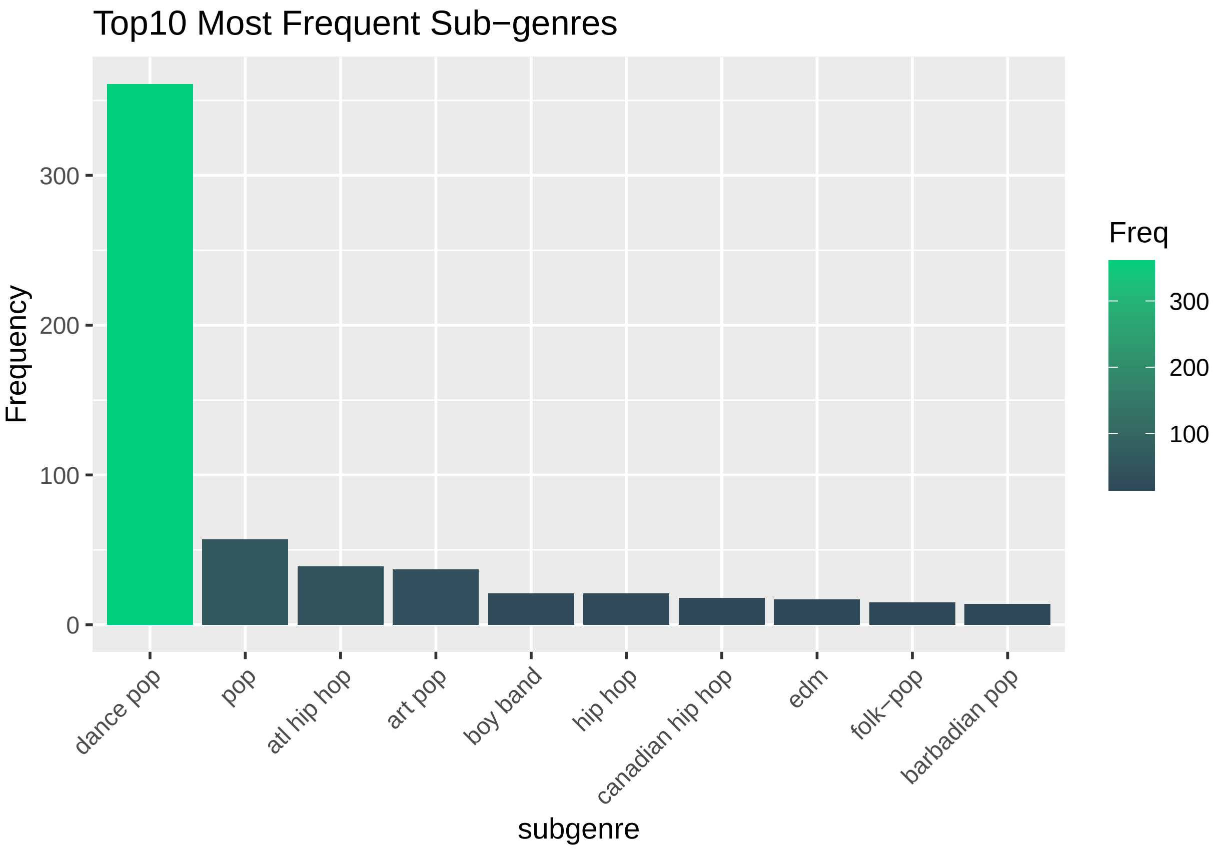
```
subgenre_freq <- as.data.frame(table(top100$subgenre))
names(subgenre_freq)[names(subgenre_freq) == "Var1"] <- "subgenre"

set.seed(7)
wordcloud(words = subgenre_freq$subgenre, freq = subgenre_freq$Freq, min.freq = 1,
    max.words = 200, random.order = FALSE, rot.per = 0.35, colors = brewer.pal(n = 8,
        name = "Accent"))
```



```
# most common sub-genre is overwhelmingly dance pop, followed by pop
```

#barplot

```
# barplot for the most frequent genres in the top100 over all years
subset <- subgenre_freq[order(-subgenre_freq$Freq), ]
top10_genre <- subset[1:10, ]
top10_genre %>%
    ggplot(aes(reorder(subgenre, -Freq), Freq, fill = Freq)) + labs(title = "Top10 Most Frequent Sub-ge
    ylab("Frequency") + xlab("subgenre") + geom_bar(stat = "identity") +
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
    scale_fill_gradient(low = "#2F4858", high = "#01CD7C")
```



#Popular genres for each year

```
# count genres for each year
genre_freq_year <- top100 %>%
    dplyr::select(top.year, genre) %>%
    count(top.year, genre) %>%
    arrange(top.year, desc(n))

# top3 genres per year
top3_per_year <- genre_freq_year %>%
    arrange(desc(n)) %>%
    group_by(top.year) %>%
    slice(1:3) %>%
    rename(Freq = n)

last_per_year <- genre_freq_year %>%
    arrange(desc(n)) %>%
    group_by(top.year) %>%
    slice(4:10) %>%
    group_by(top.year) %>%
    summarise(Freq = sum(n)) %>%
    mutate(genre = "others")

# new data frame that sums up frequencies of all genres not in the
# top3 for each year as others
genre_freq_per_year_others <- rbind(top3_per_year, last_per_year) %>%
    rename(Year = top.year)
```

```
# piedonut chart visualization library(webr)
# genre_freq_per_year_others %>% PieDonut(aes(Year, genre,
# count=Freq), #title = 'Top Genres: 2010-2019', showRatioThreshold =
# 0.015, donutLabelSize = 2.6, showRatioPie = FALSE, color='azure')
```

The PieDonut chart above, which unfortunately does not knit to pdf, shows that pop and hip hop music dominated the charts in almost all years. The minimum threshold for displaying percentages was set to a relative frequency of 0.15. The interesting thing is that hip hop fell in the chart from 2011-2014 but made a resurgence 2015 and then again in 2017 and onward.

##Fitting Multilevel Models

```
attach(top100)
top100$artist.type_ind <- ifelse(artist.type == "Duo", 2, ifelse(artist.type ==
    "Solo", 1, ifelse(artist.type == "Band/Group", 4, 3)))  #make new indicator for artist.type
subset_big <- top100[, c(3, 5, 7:17, 19)]
subset_big$year.released <- as.factor(subset_big$year.released)
```

#Varying intercepts without varying slopes

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(arm)
```

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## arm (Version 1.13-1, built: 2022-8-25)

## Working directory is /Users/dz/Documents/MSSP/GitHub/MA678 midterm project/Midterm-project
```

```
# varying intercepts with popularity as response, group by genre
M1_p_genre <- lmer(pop ~ bpm + nrgy + dnce + val + year.released + (1 |
    genre), data = subset_big)
coef(M1_p_genre)
library(performance)
```

```
##
## Attaching package: 'performance'

## The following object is masked from 'package:arm':
##
##     display
```

```
model_performance(M1_p_genre)

library(cAIC4)
```

```
## Loading required package: stats4

## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##     lmList

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
# stepwise
full.model <- lm(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + artist.type_ind + year.released, data = subset_big)
gc <- c("genre", "artist.type_ind", "year.released")
stepwise_M1 <- stepcAIC(full.model, groupCandidates = gc, data = subset_big,
    trace = TRUE, direction = "forward", returnResult = TRUE)

# compare models to delete excess predictors
M1_mod1 <- lmer(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + artist.type_ind + year.released + (1 | artist.type_ind) + (1 |
    genre), data = subset_big)
M1_mod2 <- lmer(pop ~ bpm + nrgy + dnce + val + acous + spch + artist.type_ind +
    year.released + (1 | artist.type_ind) + (1 | genre), data = subset_big)

model_performance(M1_mod1)
model_performance(M1_mod2)
# difference in R2 (cond) is minimal so select second model
M1_final <- lmer(pop ~ dnce + acous + bpm + nrgy + nrgy:dnce + artist.type_ind +
    year.released + (1 | artist.type_ind) + (1 | genre), data = subset_big)
model_performance(M1_final)
```

#Varying intercepts and slopes

```r
attach(subset_big)
```

```
## The following objects are masked from top100:
##
##     acous, bpm, dB, dnce, dur, genre, live, nrgy, pop, spch, top.year,
##     val, year.released
```

```r
sc <- c("bpm", "nrgy", "dnce", "acous")
stepwise_M2 <- stepcAIC(M1_final, slopeCandidates = sc, data = subset_big,
    trace = TRUE, steps = 100, direction = "forward", returnResult = TRUE)
```

```
## boundary (singular) fit: see help('isSingular')

## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
## boundary (singular) fit: see help('isSingular')
```

```r
M2_final <- lmer(pop ~ dnce + acous + bpm + nrgy + artist.type_ind + year.released +
    (1 + dnce | artist.type_ind) + (1 + nrgy | genre) + dnce:nrgy, data = subset_big)
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
model_performance(M2_final)
```