midterm project

Danya Zhang

2022-12-01

R Markdown

Reading in data

```
top100 <- read.csv("/Users/dz/Downloads/archive (1)/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

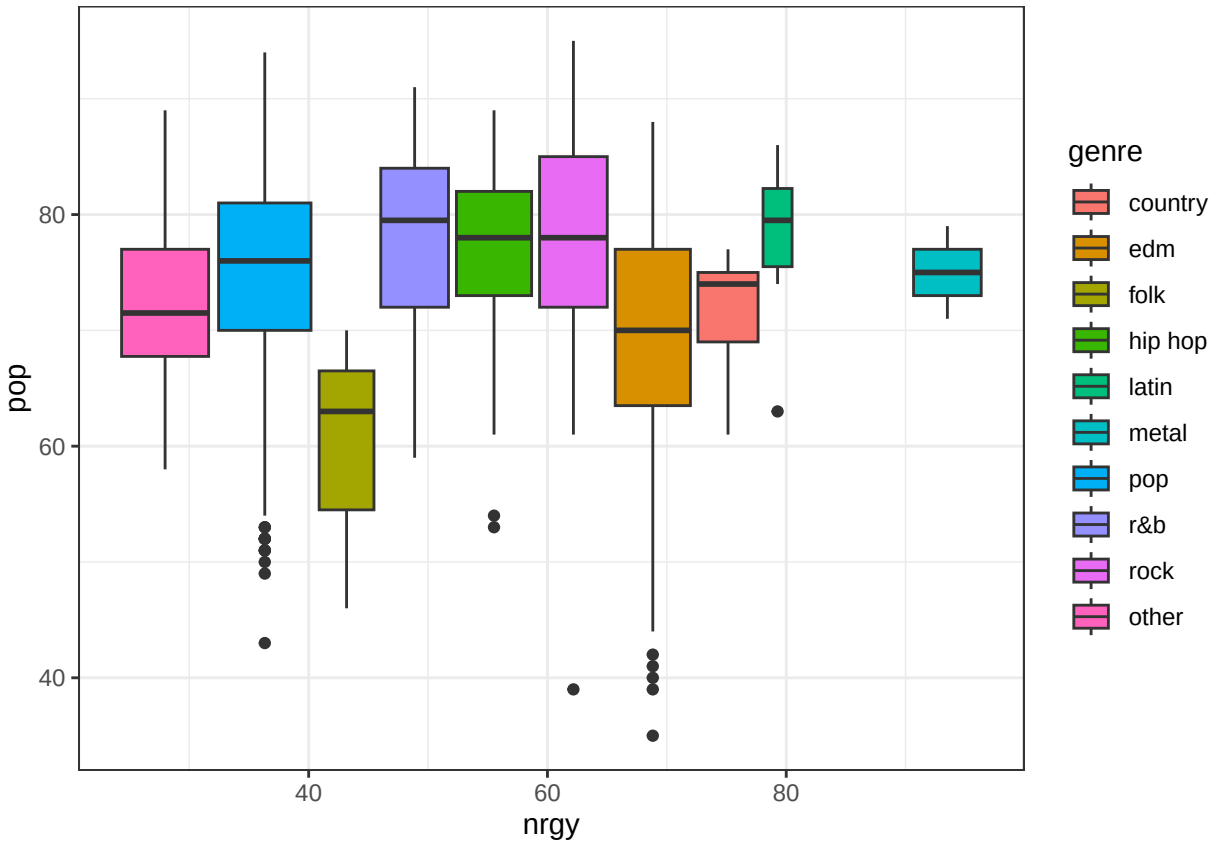
```
# boxplot grouped by genre for popularity vs energy
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

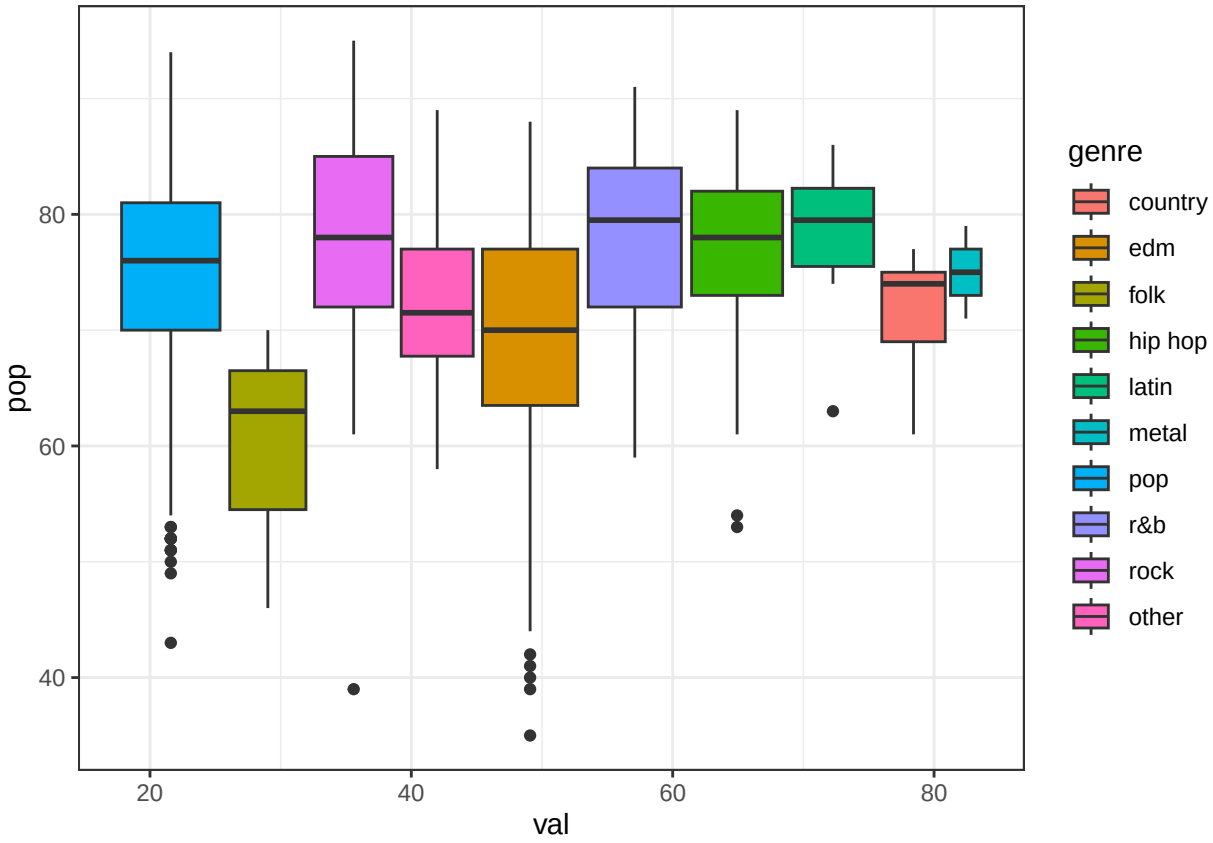
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## 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()
```



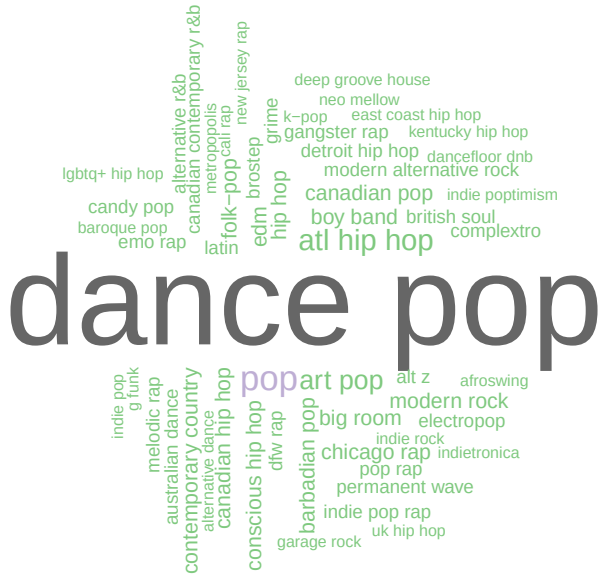
#Wordcloud

```
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, 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
```

```
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 miderm project/Midterm-project
```

```
attach(top100)
# varying intercepts with popularity as response, group by genre
M1_p_genre <- lmer(pop ~ nrgy + dnce + bpm + val + year.released + (1 |
    genre))
coef(M1_p_genre)
M2_p_genre <- lmer(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + year.released + top.year + (1 | genre))
M3_p_genre <- lmer(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + top.year + year.released + artist.type + nrgy:dnce + (1 | genre))
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
M4_p_genre <- lmer(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + top.year + year.released + artist.type + nrgy:dnce + val:nrgy +
    spch:dur + bpm:dnce + dB:nrgy + (1 | genre))
```

```
## Warning: Some predictor variables are on very different scales: consider
```

```
## rescaling
```

```
# check fit
library(performance)
```

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

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

```
model_performance(M1_p_genre)
model_performance(M2_p_genre)
model_performance(M3_p_genre)
summary(M3_p_genre)
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
# varying intercepts with popularity as response, group by year
M1_p_year <- lmer(pop ~ bpm + nrgy + dnce + dB + live + val + dur + acous +
    spch + genre + year.released + (1 | top.year))
```

```
# varying intercepts with genre as response tmp <- lmer(pop ~ bpm +
# nrgy + dnce + dB + live + val + dur + acous + spch + year.released
# + artist.type + (1 | top.year))
```