Project 3

Wenxuan Zhou wz4388

This is the dataset used in this project:

```r
spotify_songs <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/d
spotify_songs <- spotify_songs %>% na.omit()
```

Link to the dataset: https://github.com/rfordatascience/tidytuesday/tree/master/data/2020/2020-01-21

Part 1

Question: Within the rap genre, do the four subgenres differ in danceability, energy, instrumentalness, tempo, and valence?

Introduction: We are working with the `spotify_songs` dataset, which contains information about songs from 6 main categories (EDM, Latin, Pop, R&B, Rap, & Rock). Each row of the dataset represents information about a single song. The dataset contains 23 columns that provide information like track name, track popularity, danceability, energy, etc.

To determine whether the subgenres of rap differ in danceability, energy, instrumentalness, tempo, and valence, we will be working with the following columns:

1. `playlist_genre`: the genre of the song
2. `playlist_subgenre`: the subgenre of the song
3. `danceability`: how suitable a track is for dancing based on a combination of musical elements including rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
4. `energy`: a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
5. `instrumentalness`: predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
6. `tempo`: the overall estimated tempo of a track in beats per minute (BPM).
7. `valence`: a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

Approach: Our approach is to conduct a Principle Components Analysis (PCA). PCA aligns the major axes with directions of maximum variation in the data, which helps us to visualize the data of the four subgenres. A rotation plot of components and a plot of the eigenvalues (shows the amount of variance explained by the various components) will be generated to help with the interpretation of the hexagonal bin plot of PC2 versus PC1.

To generate the rotation plot of components, we use the following functions:

1. `tidy(matrix = 'rotation')` to extract rotation matrix
2. `pivot_wider()` to extract the the number of the principle components put them in new columns

3. `geom_segment()` to draw a straight line between points (x, y) and (xend, yend)
4. `geom_text()` to add text to the plot

To generate the plot of the eigenvalues, we use the following functions:

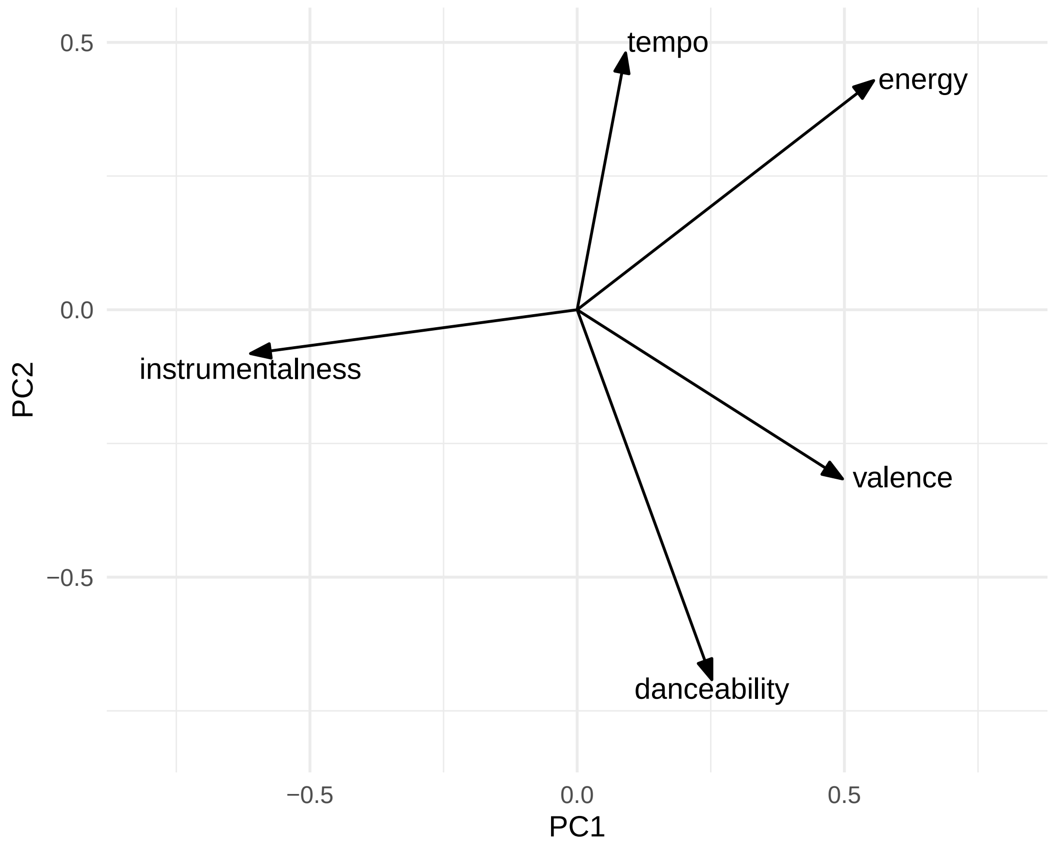
1. `tidy(matrix = "eigenvalues")` to extract eigenvalue matrix
2. `geom_col()` to create a bar plot of the variance explained by the various components

To generate the hexagonal bin plot of PC2 versus PC1, we use the following functions:

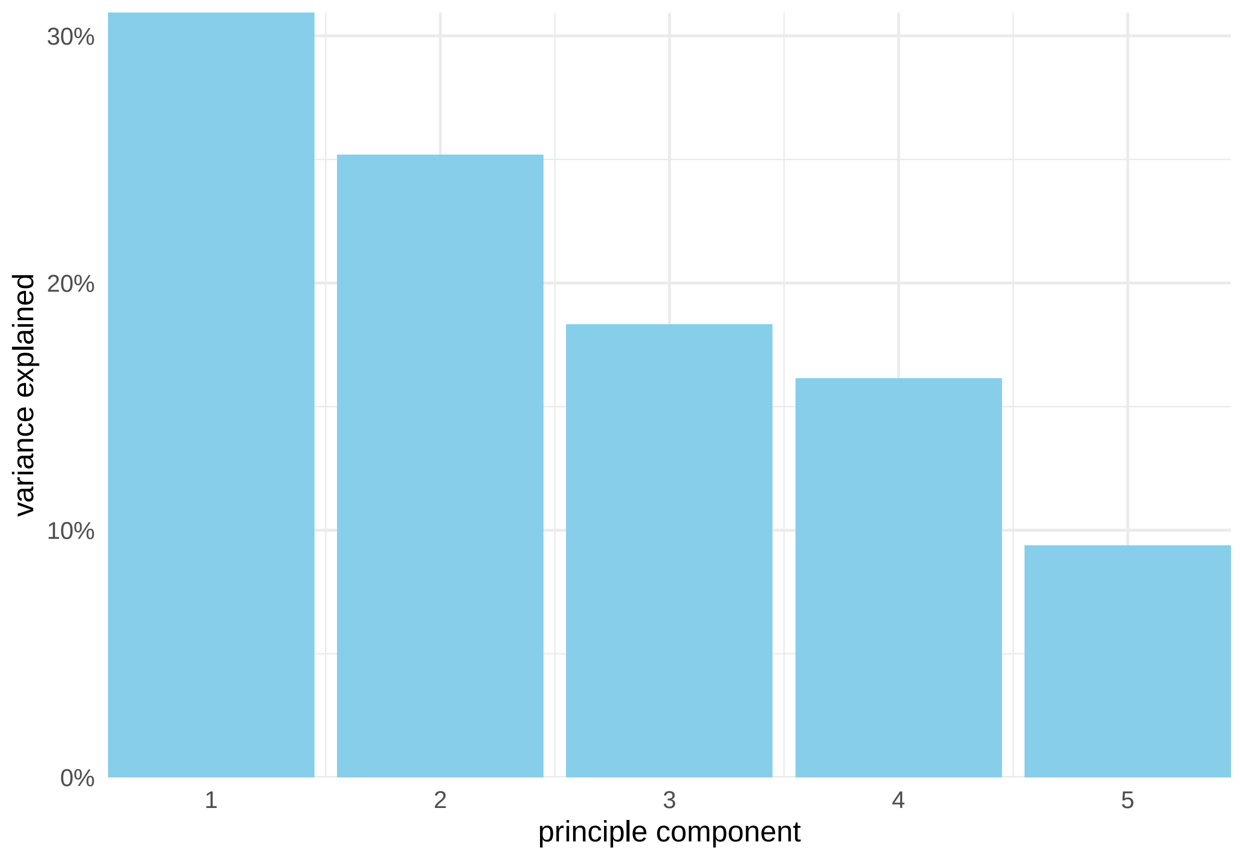
1. `augment()` to augment data according to a tidied model
2. `geom_hex()` to divide the plane into regular hexagons, count the number of cases in each hexagon, and then map the number of cases to the hexagon fill
3. `facet_wrap()` to create hexagonal bin plot facets for each subgenre

Analysis:

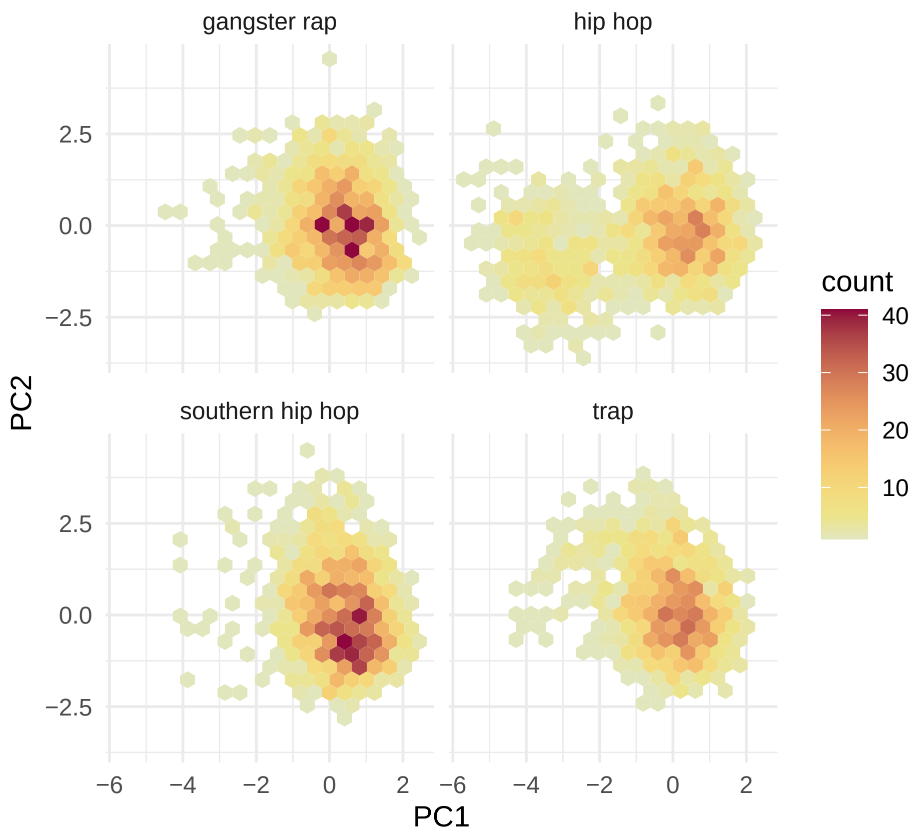
```r
spotify_songs_1 <- spotify_songs %>%
  filter(playlist_genre %in% c('rap')) # only look at rap

pca_fit <- spotify_songs_1 %>%
  select('danceability', 'energy', 'instrumentalness', 'tempo', 'valence') %>%
  scale() %>%
  prcomp() # pca

# create arrow
arrow_style <- arrow(
  angle = 20, length = grid::unit(8, "pt"),
  ends = "first", type = "closed"
)

# rotation plot
pca_fit %>%
  tidy(matrix = "rotation") %>%  # extract rotation matrix
  pivot_wider(
    names_from = "PC", values_from = "value",
    names_prefix = "PC"
  ) %>%
  ggplot(aes(PC1, PC2)) +
  geom_segment(
    xend = 0, yend = 0,
    arrow = arrow_style
  ) +
  geom_text(aes(label = column), hjust = c(0.5, -0.05, 0.5, -0.02, -0.1),
            vjust = c(0.9, 0.4, 1.2, -0.05, 0.4)) + # put text
  xlim(-0.8, 0.8) + ylim(-0.8, 0.5) +
  coord_fixed() + theme_minimal()
```



```
# plot of the eigenvalues
pca_fit %>%
  tidy(matrix = "eigenvalues") %>%
  ggplot(aes(PC, percent)) +
  geom_col(fill = 'skyblue') + # change color
  scale_x_continuous(name = 'principle component', breaks = 1:5,
                     expand = c(0, 0)) +
  scale_y_continuous(name = 'variance explained',
                     labels = scales::label_percent(), expand = c(0, 0)) +
  theme_minimal()
```



```r
# pca plot
pca_fit %>%
  augment(spotify_songs_1) %>%
  ggplot(aes(.fittedPC1, .fittedPC2)) +
  coord_fixed() +
  # hex bins to deal with overlapping points
  labs(x = 'PC1', y = 'PC2') + geom_hex(bins = 20) +
  facet_wrap(~playlist_subgenre) + # facet by subgenre
  scale_fill_continuous_sequential(palette = 'Heat') +
  theme_minimal()
```



Discussion: According to the plot of the eigenvalues, it can be observed that PC1 and PC2 capture around 55% of the total variance in the dataset. Looking at the hexagonal bin plot, along with the rotation plot, we can see that there is no difference in danceability, energy, instrumentalness, tempo, and valence for all the subgenres except for hip hop. In the hip hop subgenre, one group is high on instrumentalness and low on others (like other subgenres) while another group is low on instrumentalness and high on others. In conclusion, gangster rap, southern hip hop, and trap do not differ in danceability, energy, instrumentalness, tempo, and valence, and hip hop is different from the other subgenres.

Part 2

Question: Is popularity related to genre?

Introduction: To determine whether education level is related to card category or not, we will be working with the following columns:

1. `playlist_genre`: the genre of the song
2. `track_popularity`: song popularity (0-100) where higher is better

Approach: Our approach is to first create a summary table containing the number of songs with different popularity for each genre. I will categorize the popularity into four groups for comparison: 0 - 25, 25 - 50, 50 - 75, and 75 - 100. Next, we will visualize the relative proportion of songs with different popularity groups in each genre by using faceted pie charts. There are not many genres and popularity groups, so pie charts are appropriate for comparing the proportions. The alternative visualization is stacked bar plot. However, pie chart is more visually appealing for small datasets, so faceted pie charts are more appropriate.

To create the summary table containing the number of songs with different popularity groups for each genre, these functions will be applied:

1. `mutate() and case_when()` to create a new `popularity` column based on track_popularity
2. `group_by()` and `summarize()` to count the number of songs based on `playlist_genre` and `popularity`
3. `pivot_wider()` to extract the the number of songs based on `popularity` and put them in new columns

To plot the relative proportion of songs with different popularity in each genre, we use the following functions:

1. `mutate() and case_when()` to create a new `popularity` column based on track_popularity
2. `group_by()` and `summarize()` to count the number of songs based on `playlist_genre` and `popularity`
3. `geom_arc_bar()` to create pie charts of the relative proportions
4. `facet_wrap()` to create pie charts facet for each genre
5. `coord_fixed()`: to fix the coordinate system

Analysis:

```r
spotify_songs %>%
  # create popularity column based on track_popularity
  mutate(popularity = case_when(track_popularity >= 75 ~ '75 - 100',
                               ((track_popularity >= 50) &
                                  (track_popularity < 75)) ~ '50 - 75',
                               ((track_popularity >= 25) &
                                  (track_popularity < 50)) ~ '25 - 50',
                               ((track_popularity >= 0) &
                                  (track_popularity < 25)) ~ '0 - 25')) %>%
  group_by(playlist_genre, popularity) %>%
  summarize(n = n()) %>% # count the total number of songs
  pivot_wider(names_from = "popularity", values_from = "n")
```

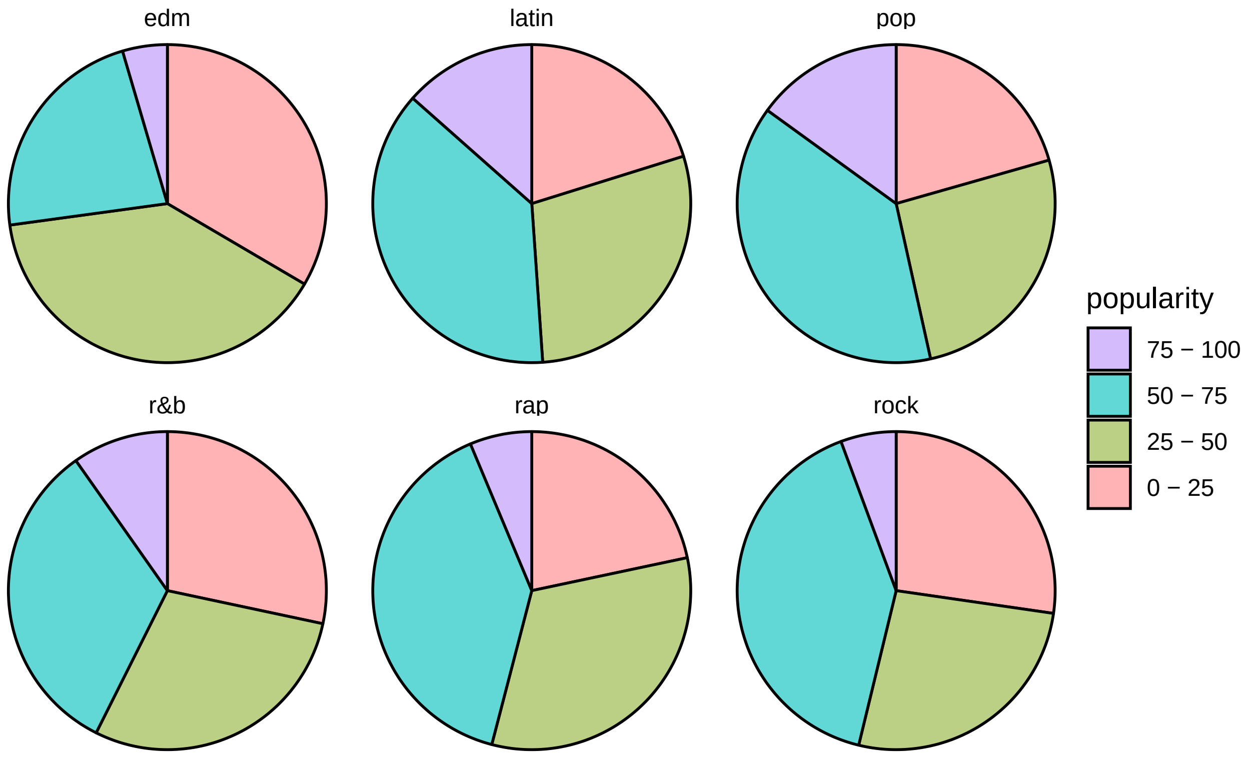
```
## `summarise()` regrouping output by 'playlist_genre' (override with `.groups` argument)
```

```
## # A tibble: 6 x 5
## # Groups:   playlist_genre [6]
##   playlist_genre `0 - 25` `25 - 50` `50 - 75` `75 - 100`
##   <chr>             <int>     <int>     <int>      <int>
## 1 edm                2021      2381      1366        275
## 2 latin              1040      1480      1939        694
## 3 pop                1134      1428      2118        827
## 4 r&b                1539      1578      1784        530
## 5 rap                1245      1859      2277        362
## 6 rock               1351      1312      2009        279
```

```r
spotify_songs_2 <- spotify_songs %>%
  mutate(popularity = case_when(track_popularity >= 75 ~ '75 - 100',
                               ((track_popularity >= 50) &
                                  (track_popularity < 75)) ~ '50 - 75',
                               ((track_popularity >= 25) &
                                  (track_popularity < 50)) ~ '25 - 50',
                               ((track_popularity >= 0) &
                                  (track_popularity < 25)) ~ '0 - 25')) %>%
  group_by(playlist_genre, popularity) %>%
  summarize(n = n())
```

```
## `summarise()` regrouping output by 'playlist_genre' (override with `.groups` argument)
```

```r
ggplot(spotify_songs_2) +
  aes(
    x0 = 0, y0 = 0, # position of pie center
    r0 = 0, r = 1,  # inner and outer radius
    amount = n, # size of pie slices
    fill = popularity
  ) +
  geom_arc_bar(stat = "pie") +
  facet_wrap(~playlist_genre) + # facet by genre
  coord_fixed() +
  scale_fill_discrete_qualitative(palette = "Set 3", name = 'popularity') +
  guides(fill = guide_legend(reverse = TRUE)) +
  theme_void()
```



Discussion: Looking at the faceted pie charts, we can see that the relative proportions of songs with different popularity are similar for r&b, rap, and rock. The popularity of more than half of the songs are less than 50, and the proportion of songs with 75-100 popularity is very small. Latin and pop also have similar relative proportions of songs with different popularity. More than half of the songs have a popularity greater than 50, and the proportion of songs with 75-100 popularity is larger compared to other genres. The relative proportion of edm is different from all the other genres. About 70% of the songs in edm genre have a popularity less than 50. In conclusion, popularity is related to genre.