Drafting Visualizations: The Anatomy of a Spotify Hit

Vedika Shirtekar

2/23/26

The purpose of this document is to draft proposed visuals for this infographic based on the Spotify hits dataset obtained from Kaggle and organized by Solomon Ameh. Final versions of summarized data and visualizations are subject to change.

Import & Wrangle Data

Here, the necessary packages are loaded, and the Spotify dataset is read from the local directory.

```r
library(tidyverse)
library(ggthemes)
```

```r
# Import most popular songs
high_popular <- read.csv(
  here::here("data", "high_popularity_spotify_data.csv"))

low_popular <- read.csv(
  here::here("data", "low_popularity_spotify_data.csv"))
```

The dataset is filtered to retain only variables relevant to answer each question, ensuring a focused and efficient visualization workflow. Additionally, popularity groups were assigned based on each original popularity dataset prior to merging and creating `spotify_clean`.

```r
# Combine high and low popularity songs
low_popular$popularity_group <- "low"
high_popular$popularity_group <- "high"
```

```
spotify_clean <- bind_rows(low_popular, high_popular)

# Select variables relevant to the analysis
spotify_clean <- spotify_clean %>%
  select(track_id, track_name,
         playlist_genre,
         track_popularity, popularity_group,
         valence, energy, danceability) %>%
  distinct() %>%        # Remove duplicate rows
  drop_na()             # Remove rows with missing values

spotify_clean <- spotify_clean %>% mutate(popularity_group = factor(popularity_group, leve
```

Pre-planning: Restating Infographic Objectives

1. Restate the questions you hope to answer with your inforgraphic. This should include *one overarching question* (think of this as driving the overall theme of your infographic) and *at least three subquestions* (each of which will be addressed by your infographic's component visualizations). Have these questions changed at all since FPM #1? If yes, how so?

- This infographic aims to explore the audio and genre characteristics that distinguish high popularity songs from low popularity songs on Spotify. Essentially, the goal is to identify patterns that explictly define a "hit". The overarching question in mind is: What makes a song a Spotify hit? This will be addressed with three subquestions:

  - Are hits emotionally different from non-hits?
  - Do hit songs have a signature sound based on certain energy-danceability combinations?
  - Which genres are most likely to produce a hit?

- Both the first and second subquestions have changed. Originally, my first subquestion was framed as "are hit songs happier?", which implied a yes/no answer. I reframed this approach to focus on emotion difference between the two popularity groups (high vs low popularity) more broadly to better reflect what the valence (emotional) distribution actually shows. The second question was also initially one-sided which asked only whether popular songs cluster in the energy-danceability space. I refined this question to explicitly compare high and low popularity groups since I wanted to specifically determine WHERE hits cluster compared to non-hits.

2. Explain which variables from your data set(s) you will use to answer the above questions, and how.

The analysis uses `track_popularity` as the primary variable to define two groups: high popularity (scored 68-100) and low popularity (scored 11-68). These thresholds were established by Ameh in the original dataset, where songs were pre-classified into separate high and low popularity subsets before being combined for this analysis. These two groups serve as the basis for all three subquestion comparisons.

- Are hit songs emotionally different from non-hits?
    - `valence` (emotional enjoyment) is the primary variable used for this subquestion. Valence is defined as a measure of musical positiveness on a scale of 0-1. Higher valence indicates a happier and more enjoyable sound while lower valence indicates a sadder and more negative sound. A violin plot overlaid with a boxplot compares the full distribution of valence scores between high and low popularity groups to showcase both the shape of the distribution and the median difference.
- Do hit songs have a signature sound based on energy and dancability scoring?
    - Both `energy` and `danceability` are used together as a 2D space to observe potential clustering or close relationships between the two popularity groups. Energy measures how intense and active a song is on a 0-1 scale, while danceability measures how suitable a track is for dancing based on tempo, rhythm stability, and beat strength. A 2D density contour plot was chosen to represent both popularity groups in this space, revealing where each group concentrates and highlighting a distinct zone of high energy and high danceability for Spotify hits.
- Which genres are most likely to produce a hit?
    - `playlist_genre` and `popularity_group` are used together for this subquestion, where playlist_genre details the specific genre a song is grouped under by Spotify. For each genre, the percentage of songs that fall into the high popularity group is calculated. A lollipop plot was chosen to display the hit rate of the top 10 genres by total song count, ranked from highest to lowest. Hit rate was calculated as the percentage of each genre's songs that fall into the high popularity group.

3. In FPM #2, you created some exploratory data viz to better understand your data. You may already have some ideas of how you plan to formally visualize your data, but it's *incredibly* helpful to look at visualizations by other creators for inspiration. Find *at least two* data visualizations that you could (potentially) borrow / adapt pieces from. Download and embed them into your `drafting-viz.qmd` file, and explain which elements you might borrow (e.g. the graphic form, legend design, layout, etc.).

For my third subquestion, I am considering breaking the highest and lowest popularity genres into faceted groups to better compare the most and least popular categories. Although I may decide not to pursue this idea, the author's use of faceted lollipop charts by group tells a compelling story that I could potentially adapt for my own visualization.

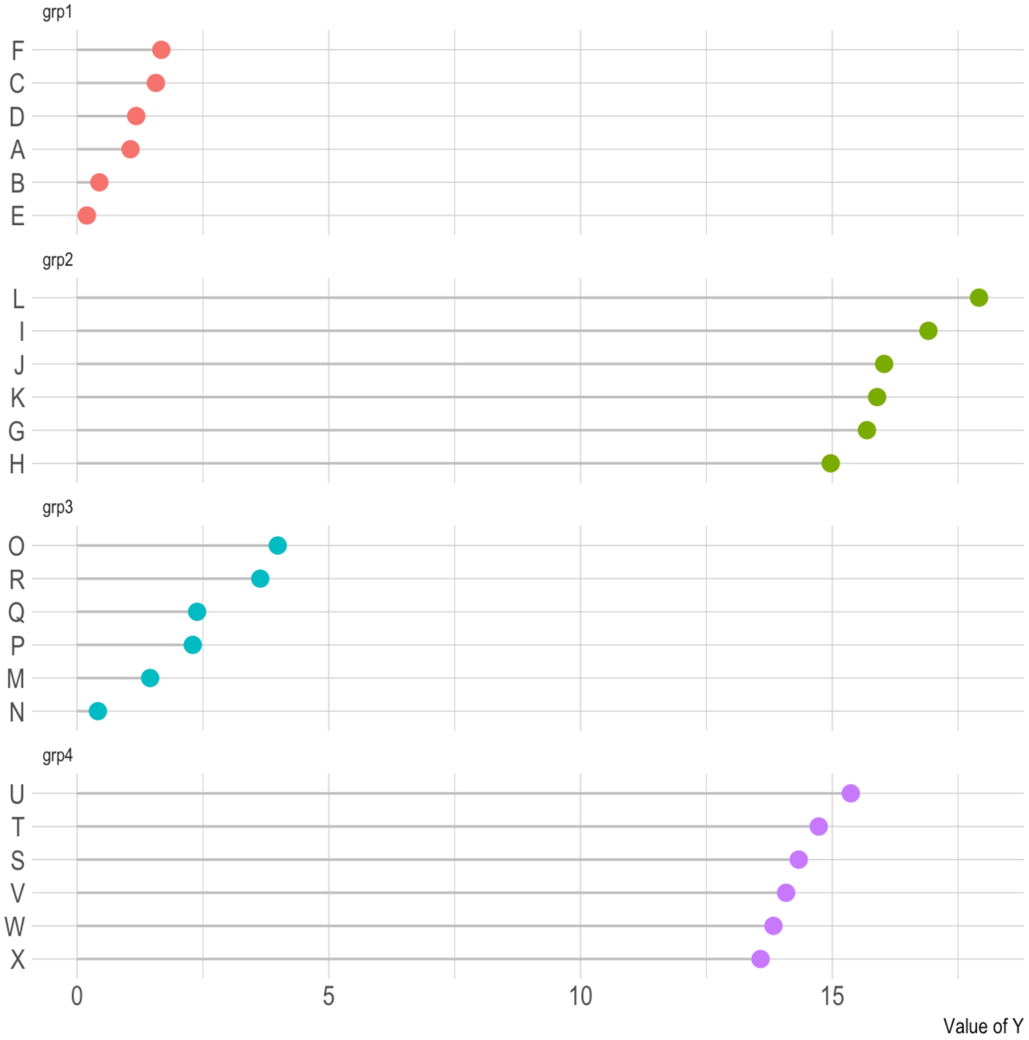


Figure 1: Figure 1. Facetted lollipop plot for multiple groups. Image courtesy of Data to Viz.

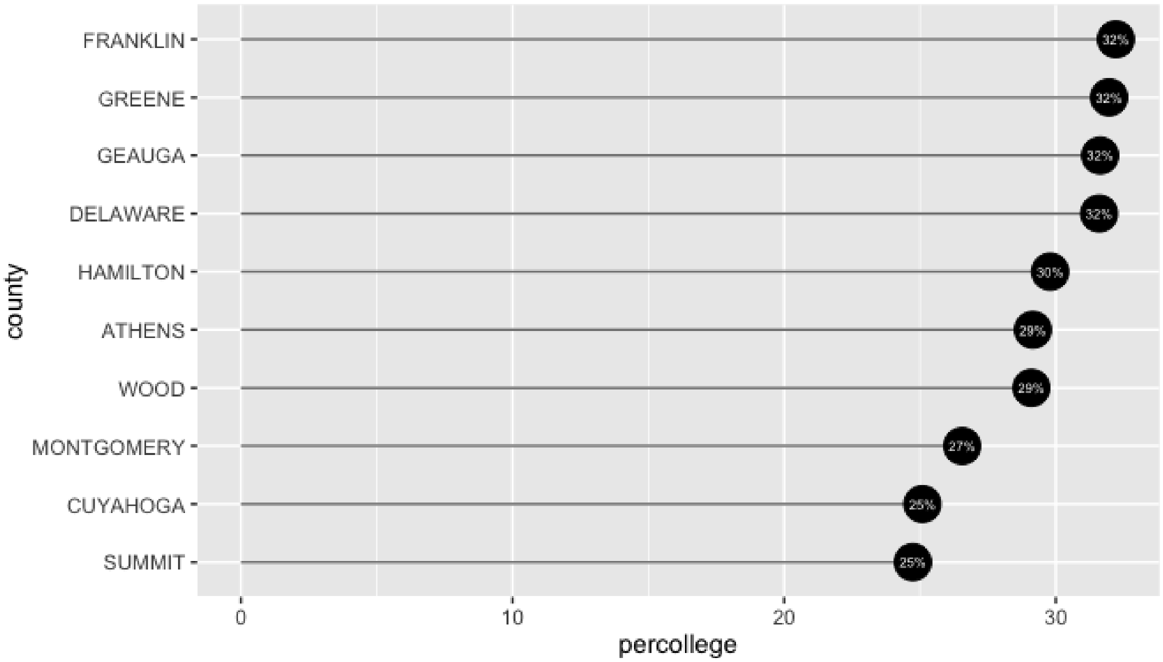


Figure 2: Figure 2. Lollipop chart with labelled points for percent reference. Image courtesy of UC Business Analytics R Programming Guide.

Based on the lollipop chart above, I was interested in applying a similar labeling style by displaying the percentage value at each point. While my lollipop plot of the top 10 most popular genres is expected to show considerable variation, labeling each point helps clarify small differences in popularity between groups and prevents low-ranking genres from appearing as though they have zero popularity.

The main element I was interested in borrowing from this scatterplot with confidence ellipses (Figure 3) is the use of `stat_ellipse` to showcase group clustering in 2D space. I recognize this as an efficient approach that could be adapted for my energy-danceability plot as an alternative or complement to `geom_density_2d`; however, I recognize the need to test for the appropriate amount of scatter to determine whether a contour or scatterplot is most appropriate for this question. I also found the fill and color pairing to be effective in creating a clean group distinction without being too visually overwhelming.

Hand drawn drafted visualizations

Recreating hand drawn visualizations

The following visualizations were created to observe trends in several components that may contribute to the popularity of Spotify hits such as: emotional enjoyment, certain combinations of energetic and danceability of tracks, and popular genres.

Are hit songs emotionally different from non-hits?

```
# Create a summary table for distribution of valence score
# based on popularity
valence_dist <- spotify_clean %>%
  select(popularity_group, valence) %>%
  distinct()
```

```
# Violin and boxplot overlaid to visualize distribution
valence_dist %>%
  ggplot(aes(x = popularity_group, y = valence, fill = popularity_group)) +
  geom_violin(trim = T, alpha = 0.3) +
  geom_boxplot(width = 0.1, fill = "white", outlier.shape = NA) +
  labs(
    title = "Are Hit Songs Emotionally Different from Non-Hits? ",
    x = "Popularity Group",
    y = "Valence (Happiness Score)"
  ) +
```

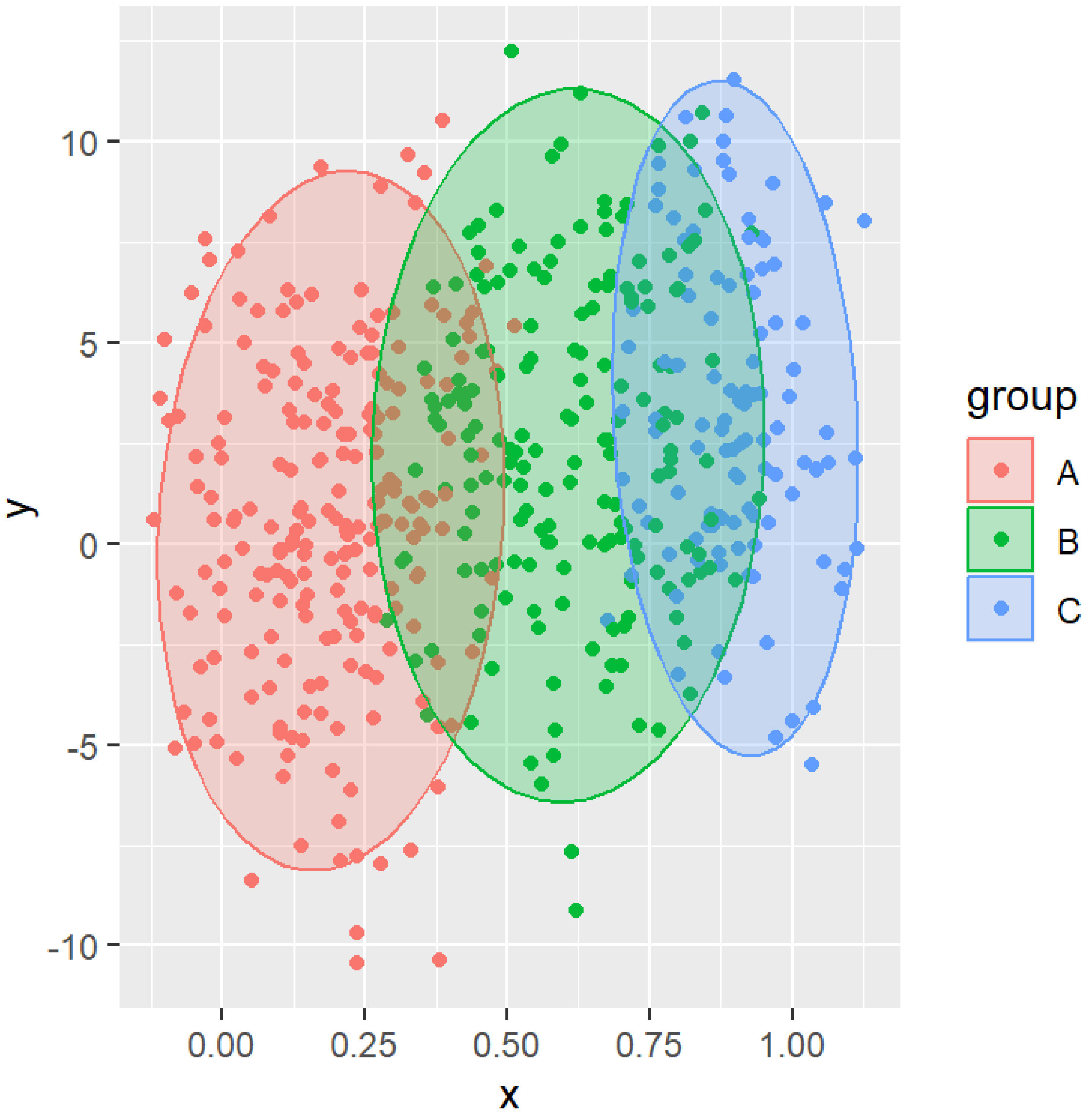
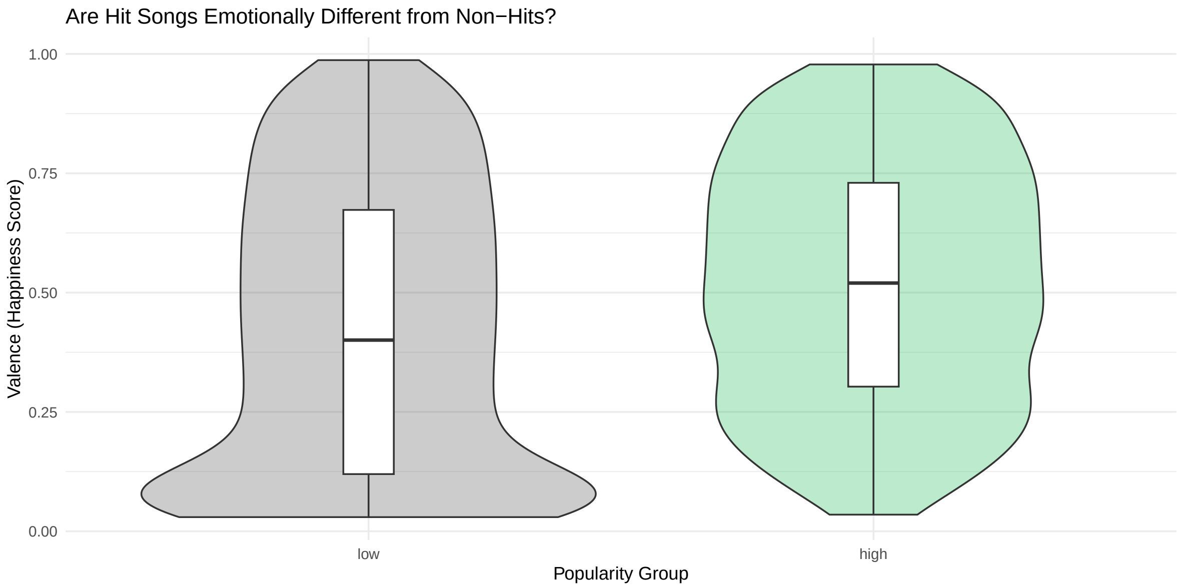


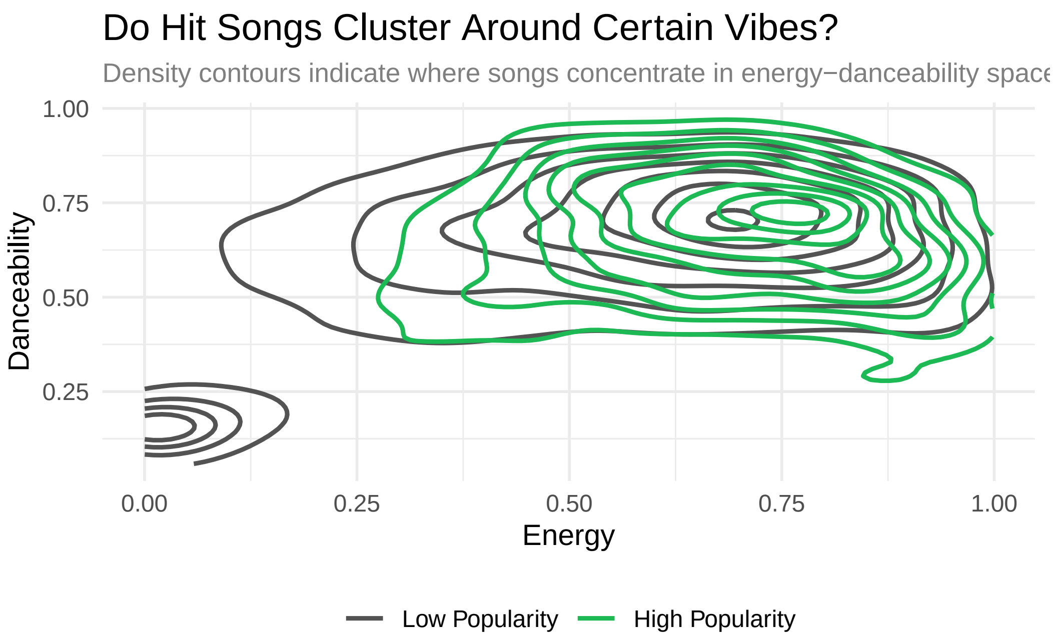
Figure 3: Figure 3. Scatterplot with ellipses. Image courtesy of R Charts.

```
  guides(fill = "none") +
  theme_minimal() +
  # Set group fill based on Spotify theme colors
scale_fill_manual(values = c("low" = "#535353", "high" = "#1DB954"))
```



Do hit songs have a signature sound based on energy and dancability scoring (is there a certain "vibe")?

```
# Create subset for variables of interest
energy_dance_df <- spotify_clean %>%
  select(energy, danceability, popularity_group)

energy_dance_df %>%
  ggplot(aes(x = energy, y = danceability, color = popularity_group)) +

  # Use contour plot with numerous bins
  geom_density_2d(linewidth = 0.8, bins = 10) +

  # Set popularity bin fill to Spotify theme
  scale_color_manual(values = c("low" = "#535353", "high" = "#1DB954"),
                     labels = c("low" = "Low Popularity", "high" = "High Popularity")) +
  labs(
    title = "Do Hit Songs Cluster Around Certain Vibes?",
    subtitle = "Density contours indicate where songs concentrate in energy-danceability s
    x = "Energy", y = "Danceability", color = NULL
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 14),
    plot.subtitle = element_text(color = "grey50", size = 10),
    legend.position = "bottom"
  )
```

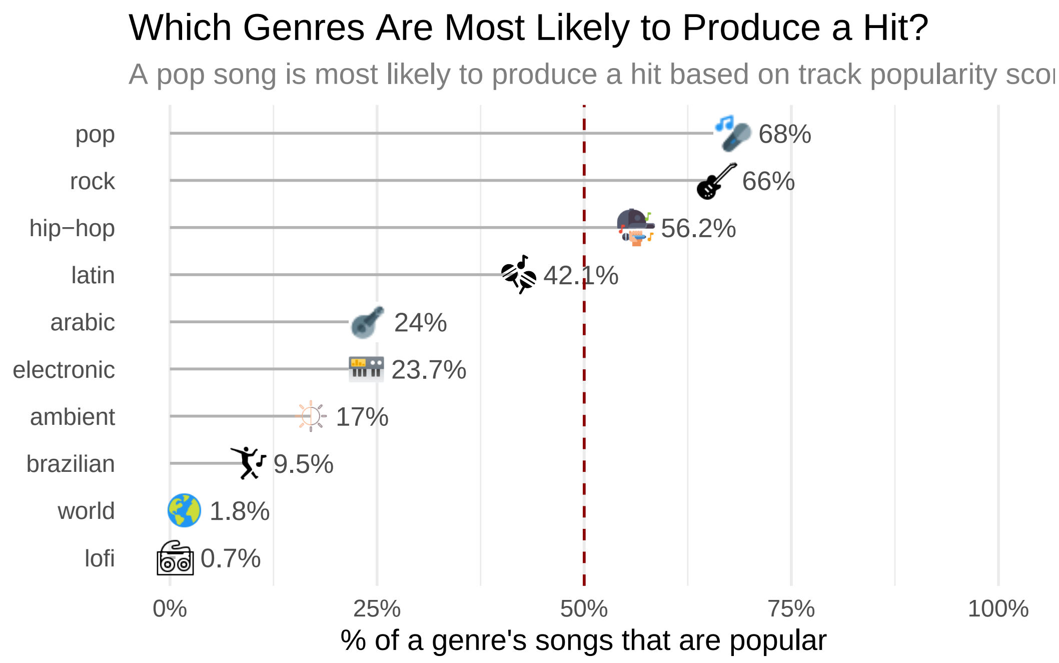


Which genres are most likely to produce a hit?

```
# For now, only observe differences for most popular genres
# Lowest popularity comparison possible for later analyses
###________________________________________________________

# Use ggimage for icons
library(ggimage)
```

```r
# Create subset to calculate % of highly popular songs within each genre
genre_pct <- spotify_clean %>%
  # Count # of songs in each genre by pop group
  count(playlist_genre, popularity_group) %>%
  # Group by genre
  group_by(playlist_genre) %>%
  # Calculate % of songs in each popularity group within genre
  mutate(pct = n / sum(n) * 100) %>%
  ungroup() %>%
  # Keep only high popularity group
  filter(popularity_group == "high")

# Filter to top 10 genres
top_genres <- spotify_clean %>%
  count(playlist_genre) %>%
  # Cut top 10 in count "n"
  slice_max(n, n = 10) %>%
  # Pull genre column only and return as vector
  pull(playlist_genre)

# Identify shared genres in top 10
genre_pct <- genre_pct %>%
  filter(playlist_genre %in% top_genres)

# Add icon column
genre_pct <- genre_pct %>%
  mutate(icon_path = case_when(
    playlist_genre == "pop"       ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-pop-mu
    playlist_genre == "rock"      ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-rock-m
    playlist_genre == "hip-hop"   ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-hip-ho
    playlist_genre == "latin"     ~ "~/MEDS/eds-240/eds240-infographic/icons/latin.png",
    playlist_genre == "electronic"~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-electr
    playlist_genre == "arabic"    ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-lute-2
    playlist_genre == "ambient"   ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-contra
    playlist_genre == "brazilian" ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-dancin
    playlist_genre == "world"     ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-world-
    playlist_genre == "lofi"      ~ "~/MEDS/eds-240/eds240-infographic/icons/icons8-casset
  ))


# Create lollipop chart with associated icons
```

```r
genre_pct %>%
  # Order in descending order based on pct
  ggplot(aes(x = pct, y = reorder(playlist_genre, pct))) +
  # Draw line segment from 0 to pct of each genre
  geom_segment(aes(x = 0,
                   xend = pct,
                   yend = playlist_genre),
               color = "grey70",
               linewidth = 0.5) +
  # Call icons as an image from file path
  geom_image(aes(image = icon_path), size = 0.05, asp = 10/6) +
  # Add pct as % amount adjusted to 3 units to right of icon
  # Round pct to nearest hundreth
  geom_text(aes(label = paste0(round(pct, 1), "%"), x = pct + 3),
          hjust = 0, size = 3.5, color = "grey30") +
  # Add threshold line
  geom_vline(xintercept = 50, linetype = "dashed",
             color = "darkred", linewidth = 0.5) +
  # Add % sign to each x label
  scale_x_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100)) +
  labs(
    title = "Which Genres Are Most Likely to Produce a Hit?",
    subtitle = "A pop song is most likely to produce a hit based on track popularity scori
    x = "% of a genre's songs that are popular",
    y = NULL # No y axis label
  ) +
  theme_minimal() +
  # Adjust axis elements (size, remove y grid)
  theme(
    plot.title = element_text(size = 14),
    plot.subtitle = element_text(color = "grey50", size = 11),
    panel.grid.major.y = element_blank()
  )
```



Draft Visualization Questions

After completing the above steps, answer the following questions:

1. What are the key insights you want your infographic to communicate, and how will your design choices help highlight and support those messages?

2. What challenges did you encounter or anticipate encountering as you continue to build / iterate on your visualizations in R? If you struggled with mocking up any of your three visualizations, describe those challenges here.

3. What ggplot extension tools / packages do you need to use to build your visualizations? Are there any that we haven't covered in class that you'll be learning how to use for your visualizations?

4. What feedback do you need from the instructional team and / or your peers to ensure that your intended message and key insights are clear?