Visualizations Appendix

Jonathan Neimann

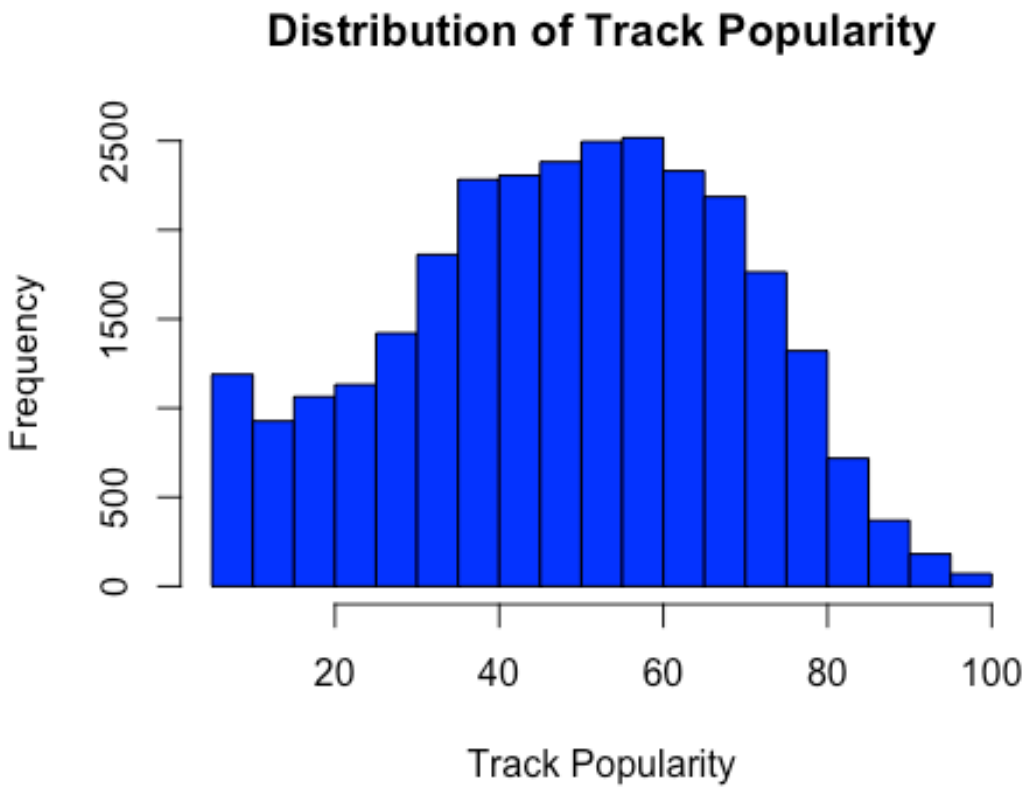
2024-12-10

All Visualizations and Code

Visualizations From Report

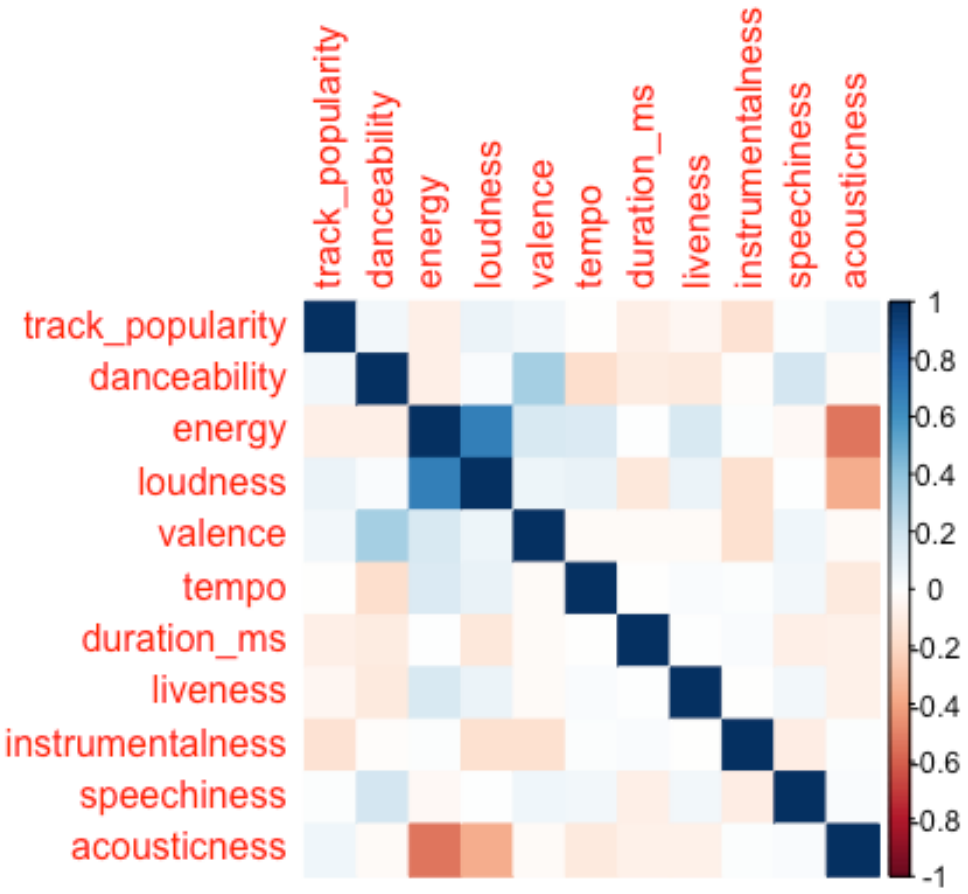
Popularity distributions

```
# Create a new dataframe where track_popularity is 5 or higher
spotify_popularity <- spotify30k[spotify30k$track_popularity >= 5, ]

hist(spotify_popularity$track_popularity,
     main = "Distribution of Track Popularity",
     xlab = "Track Popularity",
     col = "blue")
```



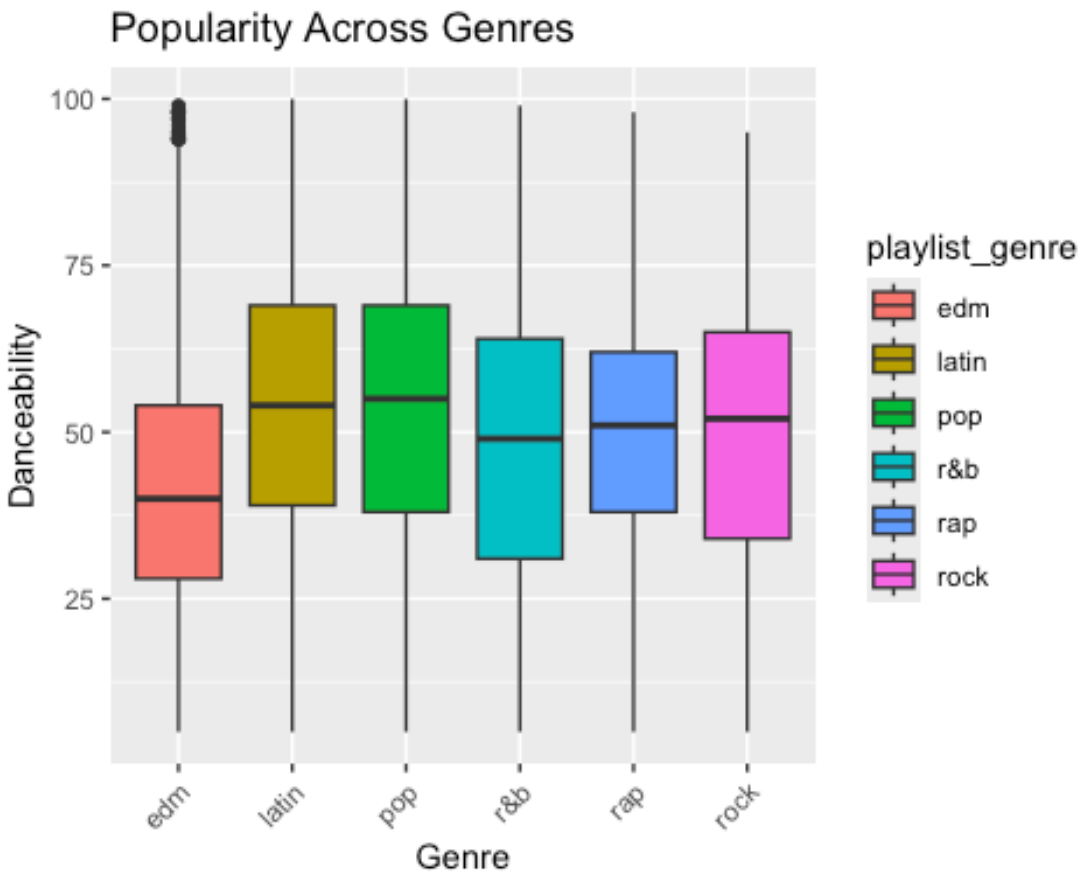
Coorelation Matrix

```
library(corrplot)

## corrplot 0.95 loaded

corr_data <- spotify_popularity %>%
  select(track_popularity, danceability, energy, loudness, valence, tempo,
duration_ms, liveness, instrumentalness, tempo, speechiness, acousticness)
%>%
  cor(use = "complete.obs")
corrplot(corr_data, method = "color")
```

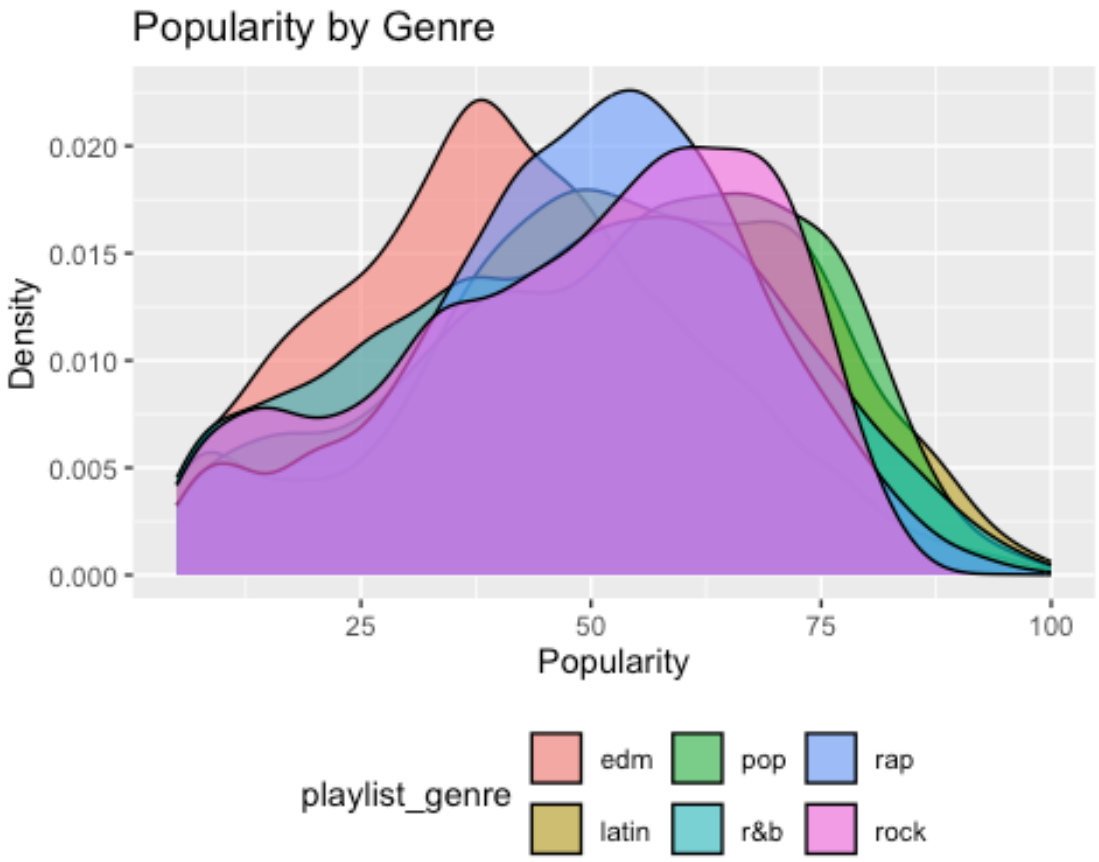


Popularity bar and density by Genre

```
ggplot(spotify_popularity, aes(x = playlist_genre, y = track_popularity, fill
= playlist_genre)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Popularity Across Genres", x = "Genre", y = "Danceability")
```

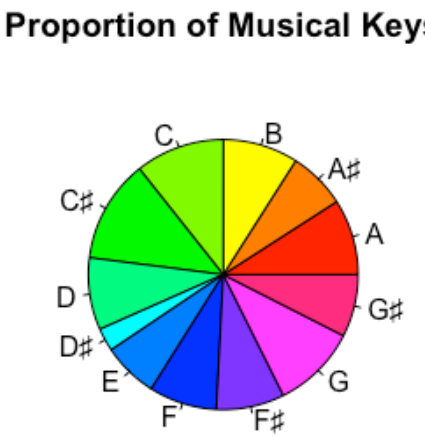



```
ggplot(spotify_popularity, aes(x = track_popularity, fill = playlist_genre))
+
  geom_density(alpha = 0.6) +
  labs(title = "Popularity by Genre", x = "Popularity", y = "Density") +
  theme(legend.position = "bottom")
```

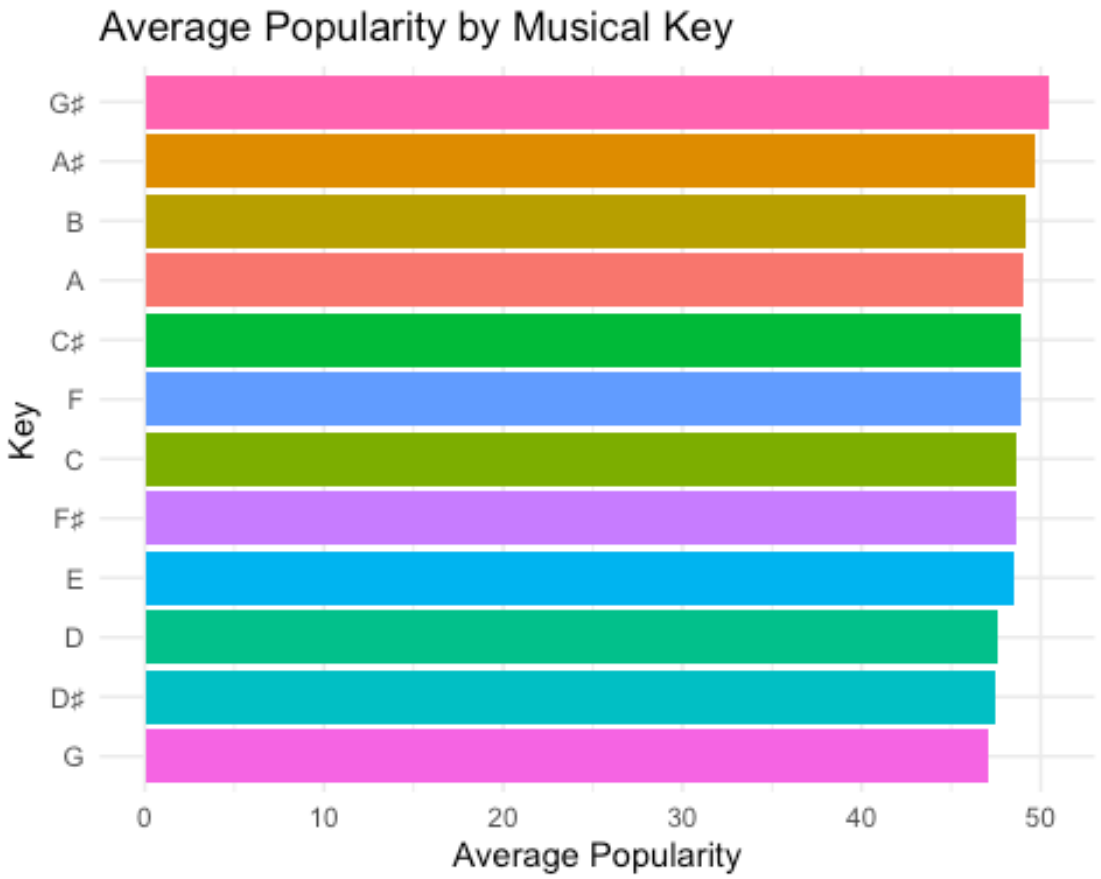

Popularity by Genre
0.020
0.015
0.010
0.005
0.000
Density
25
50
75
100
Popularity
playlist_genre
edm
pop
rap
latin
r&b
rock

Musical Keys

```
key_mapping <- c("C", "C♯", "D", "D♯", "E", "F", "F♯", "G", "G♯", "A", "A♯",
"B")
spotify_popularity$key_label <- key_mapping[spotify_popularity$key + 1]

key_counts <- table(spotify_popularity$key_label)

pie(key_counts,
    main = "Proportion of Musical Keys",
    col = rainbow(length(key_counts)))
```

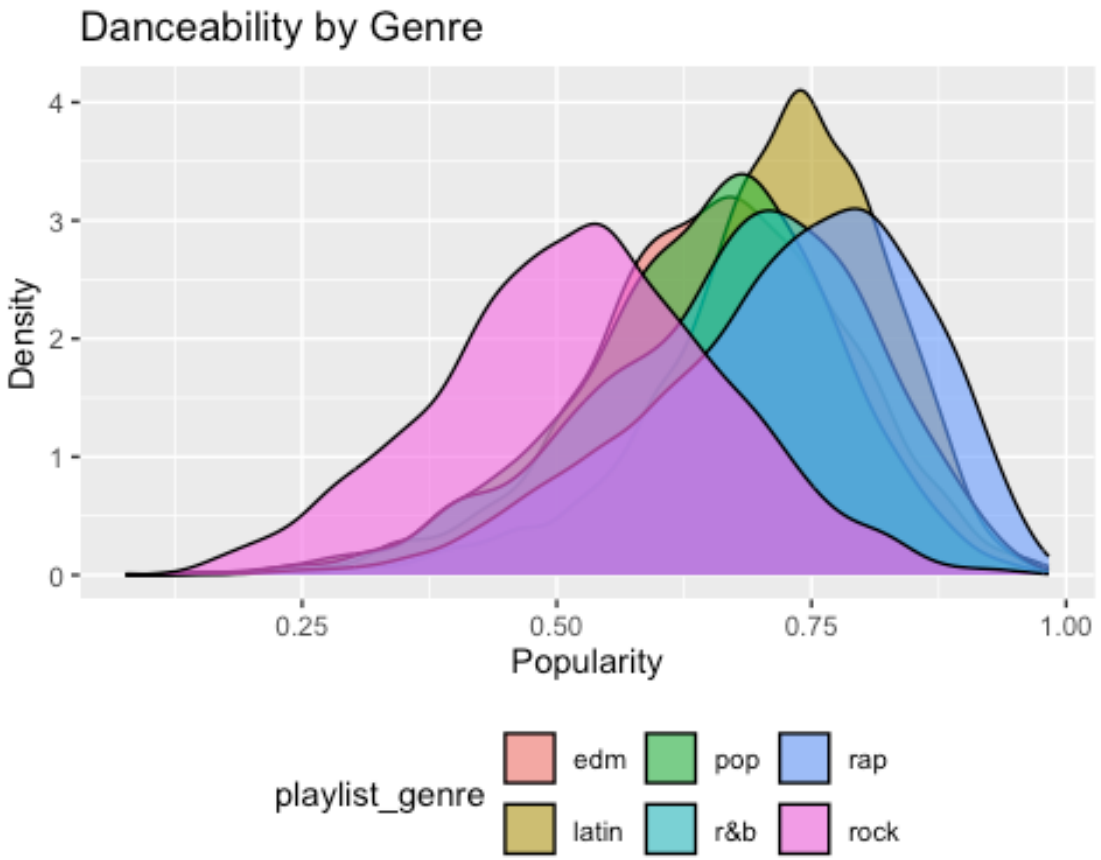



```
avg_popularity_by_key <- spotify_popularity %>%
  group_by(key_label) %>%
  summarize(avg_popularity = mean(track_popularity, na.rm = TRUE))

ggplot(avg_popularity_by_key, aes(x = reorder(key_label, avg_popularity), y =
avg_popularity, fill = key_label)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Average Popularity by Musical Key", x = "Key", y = "Average
Popularity") +
  theme_minimal() +
  theme(legend.position = "none")
```



Danceability by genre

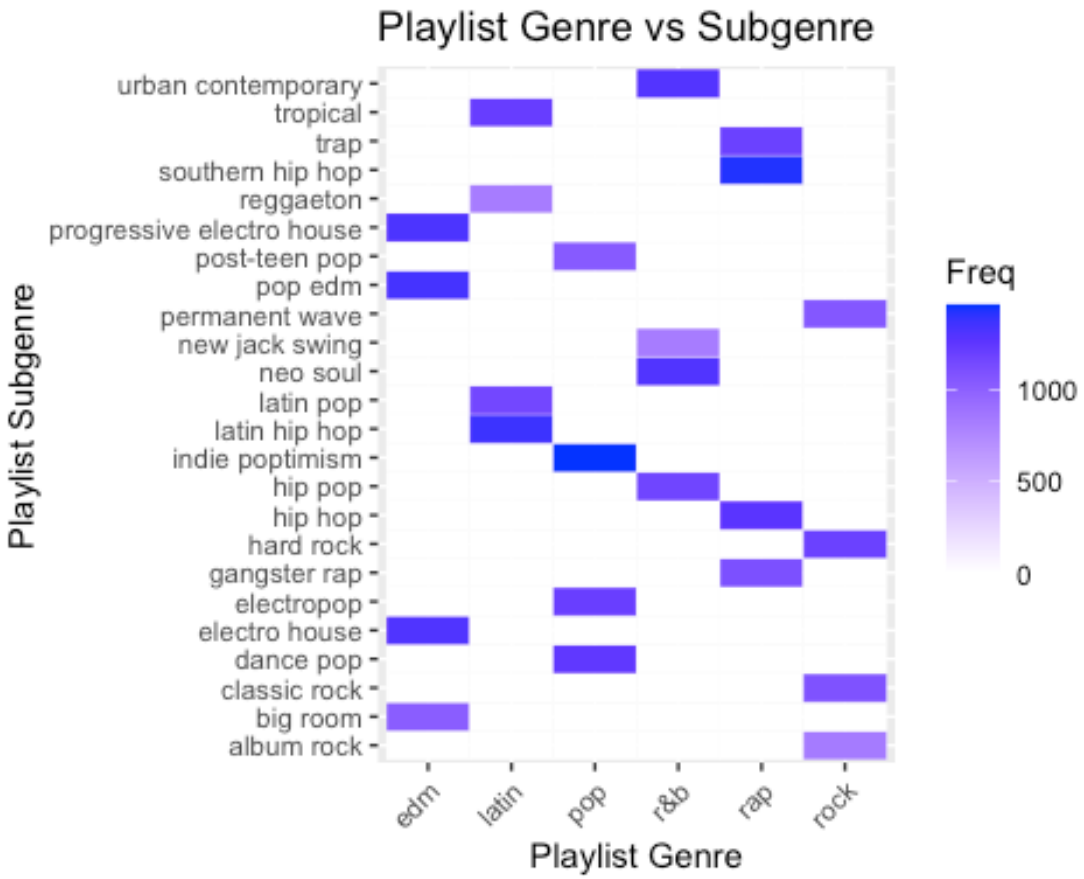
```r
ggplot(spotify_popularity, aes(x = danceability, fill = playlist_genre)) +
  geom_density(alpha = 0.6) +
  labs(title = "Danceability by Genre", x = "Popularity", y = "Density") +
  theme(legend.position = "bottom")
```


Danceability by Genre
Density
4
3
2
1
0
0.25
0.50
0.75
1.00
Popularity
playlist_genre
edm
pop
rap
latin
r&b
rock

Additional Visualizations

Playlist Genre vs Subgenre

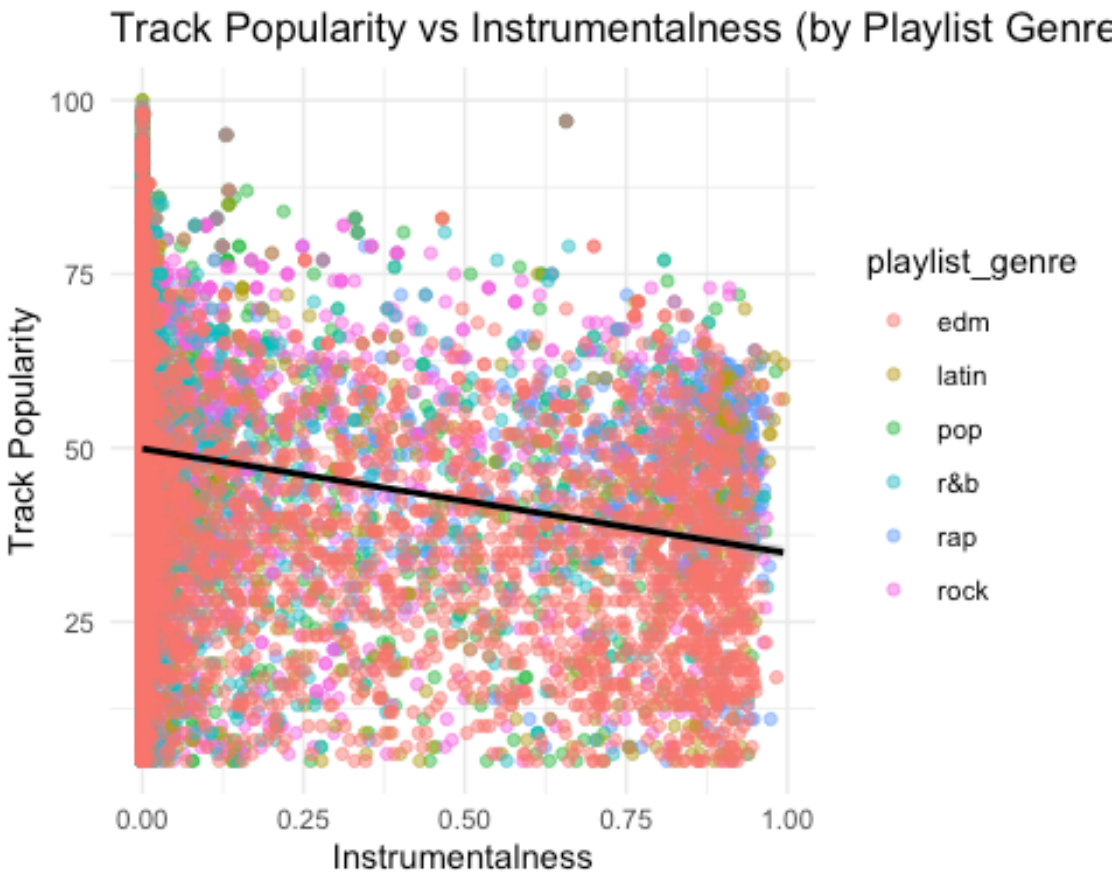
```
# Create contingency table
genre_subgenre_table <- table(spotify_popularity$playlist_genre,
spotify_popularity$playlist_subgenre)

# Convert to a data frame for ggplot
heatmap_data <- as.data.frame(as.table(genre_subgenre_table))

# Plot heatmap
ggplot(heatmap_data, aes(x = Var1, y = Var2, fill = Freq)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(title = "Playlist Genre vs Subgenre", x = "Playlist Genre", y =
"Playlist Subgenre") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Track Popularity vs different variables scatterplots

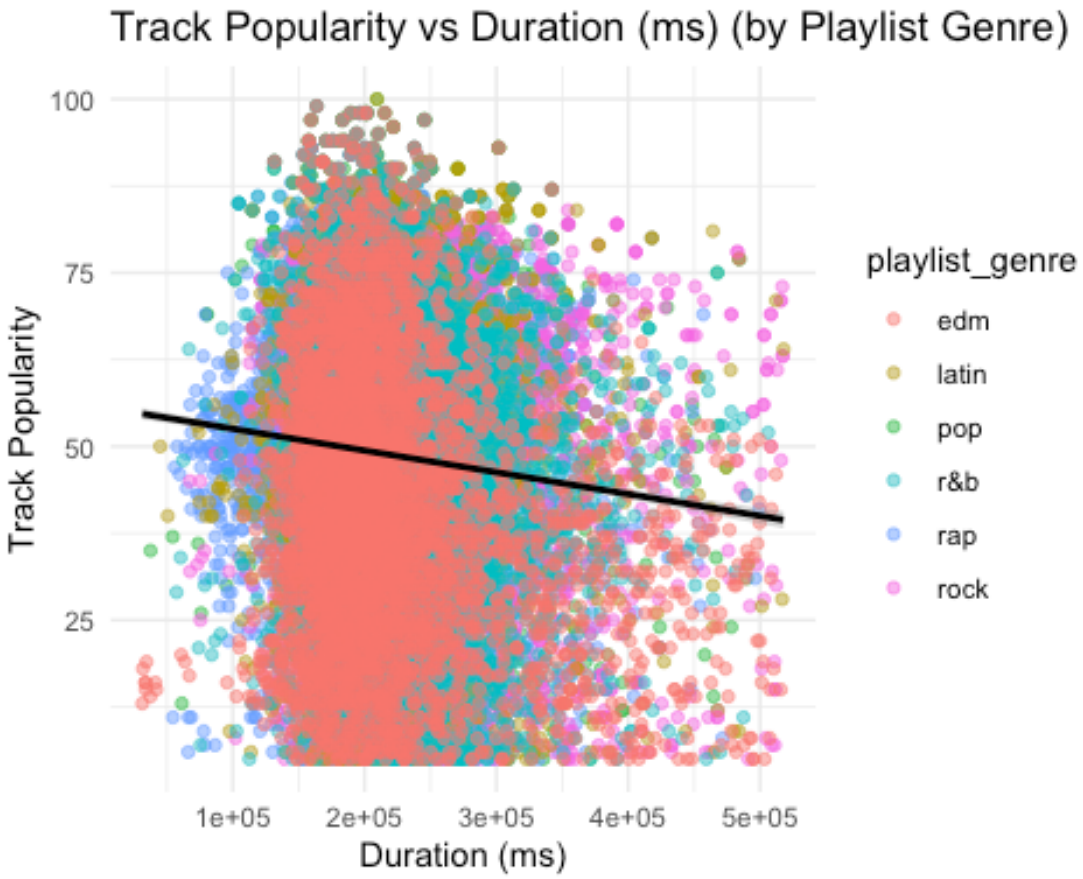
```
ggplot(spotify_popularity, aes(x = instrumentalness, y = track_popularity,
color = playlist_genre)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "black") +
  labs(title = "Track Popularity vs Instrumentalness (by Playlist Genre)",
       x = "Instrumentalness", y = "Track Popularity") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



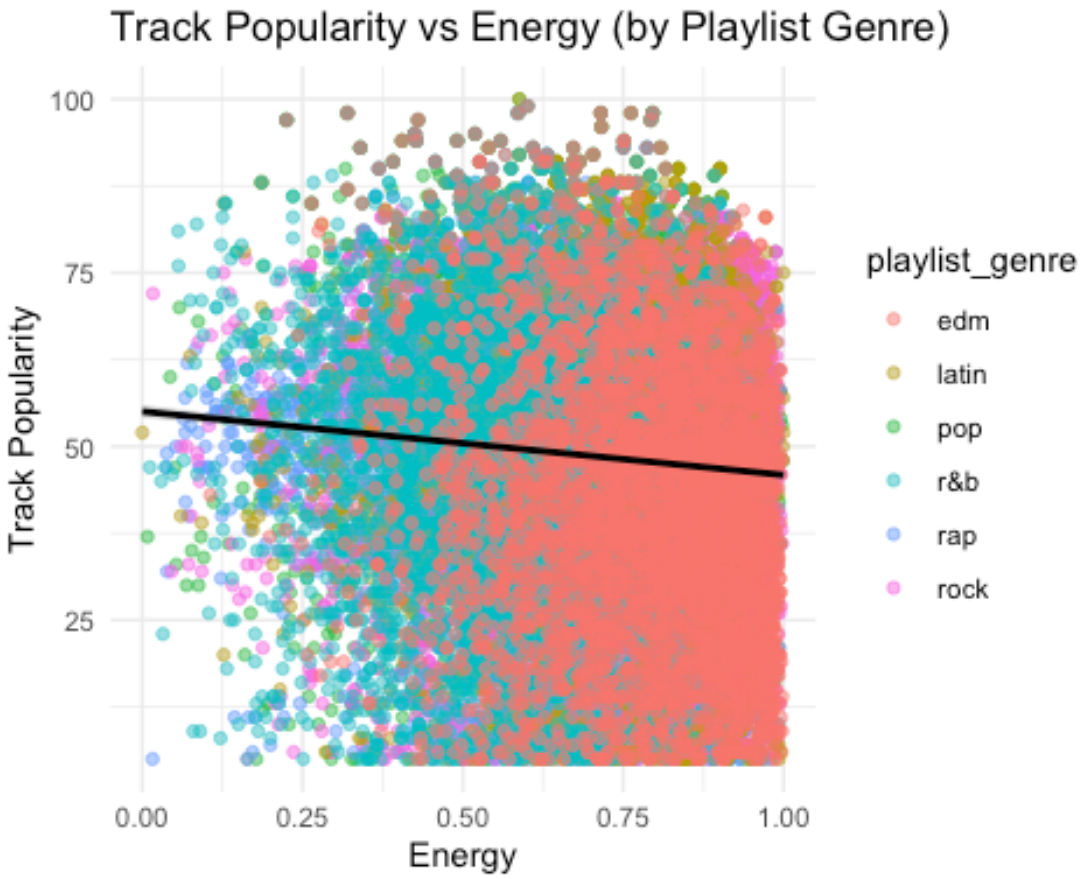

```
# Scatterplot: track_popularity vs duration_ms
ggplot(spotify_popularity, aes(x = duration_ms, y = track_popularity, color =
playlist_genre)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "black") +
  labs(title = "Track Popularity vs Duration (ms) (by Playlist Genre)",
       x = "Duration (ms)", y = "Track Popularity") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



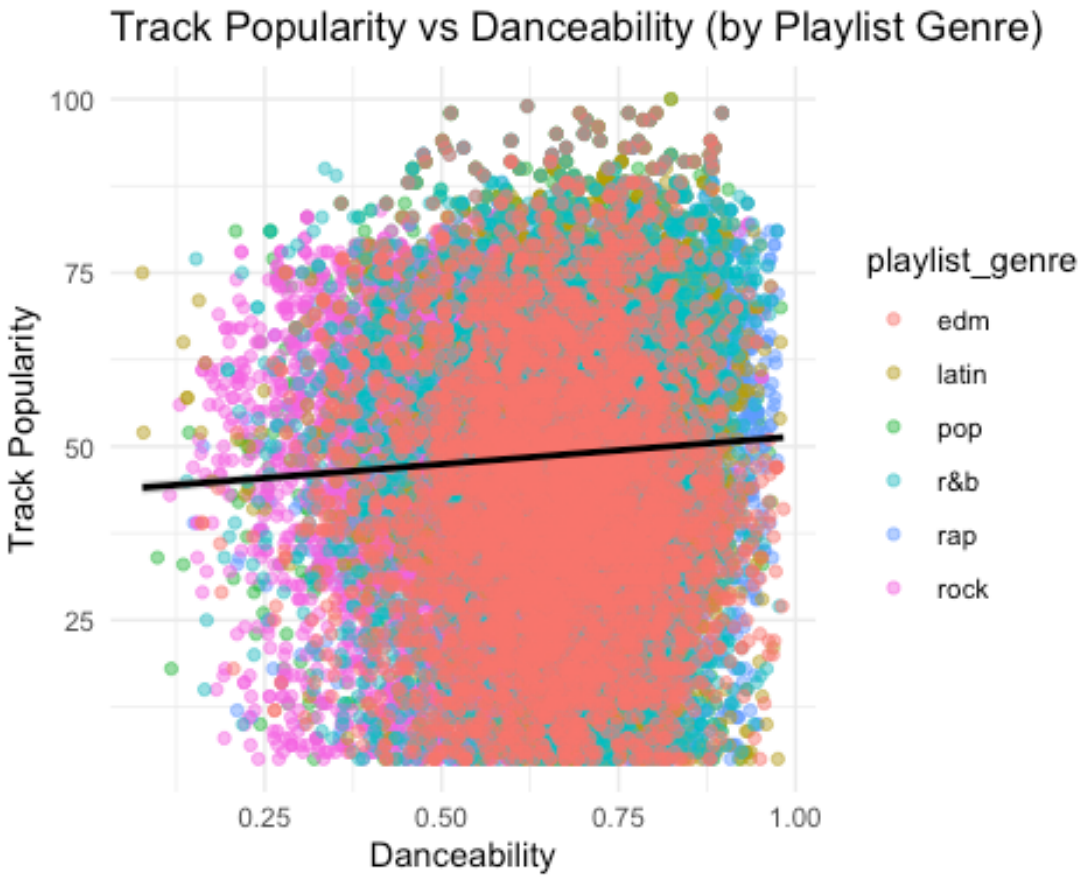

```
# Scatterplot: track_popularity vs energy
ggplot(spotify_popularity, aes(x = energy, y = track_popularity, color =
playlist_genre)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "black") +
  labs(title = "Track Popularity vs Energy (by Playlist Genre)",
       x = "Energy", y = "Track Popularity") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



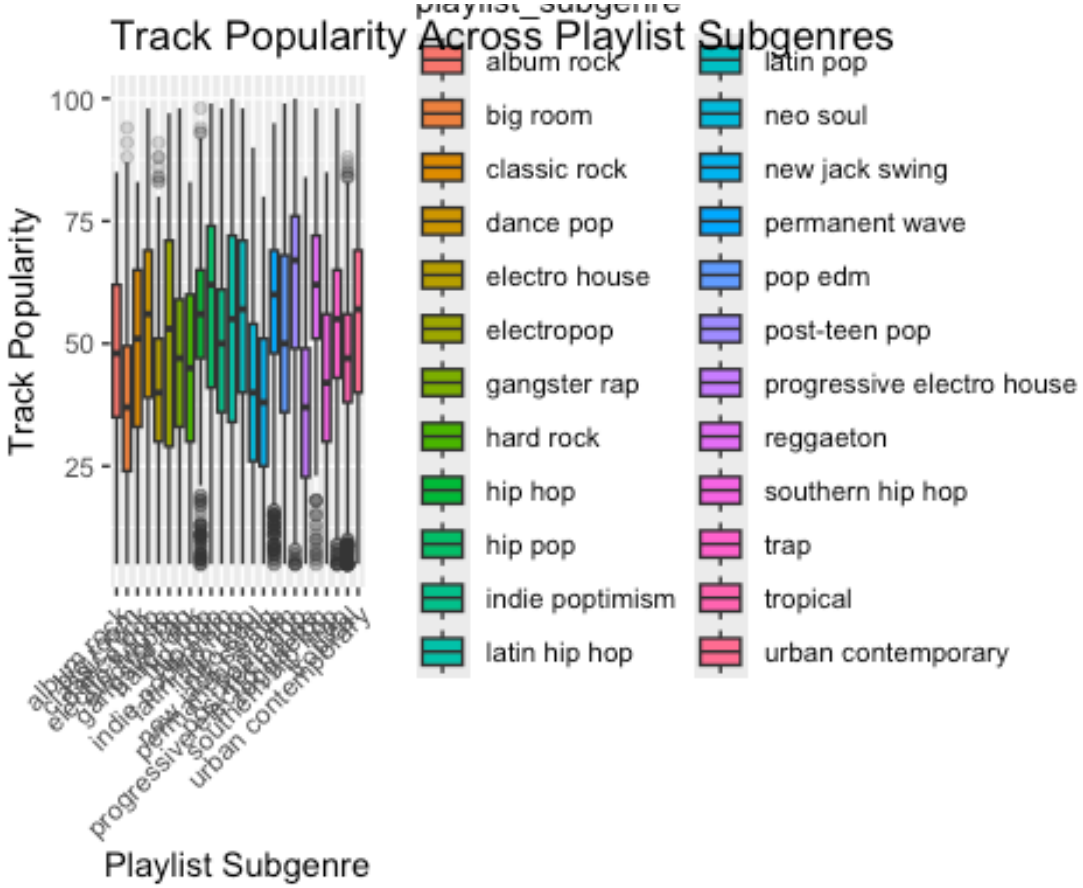

```
ggplot(spotify_popularity, aes(x = danceability, y = track_popularity, color
= playlist_genre)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "black") +
  labs(title = "Track Popularity vs Danceability (by Playlist Genre)",
       x = "Danceability", y = "Track Popularity") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


Track Popularity vs Danceability (by Playlist Genre)
Track Popularity
100
75
50
25
0.25
0.50
0.75
1.00
Danceability
playlist_genre
edm
latin
pop
r&b
rap
rock

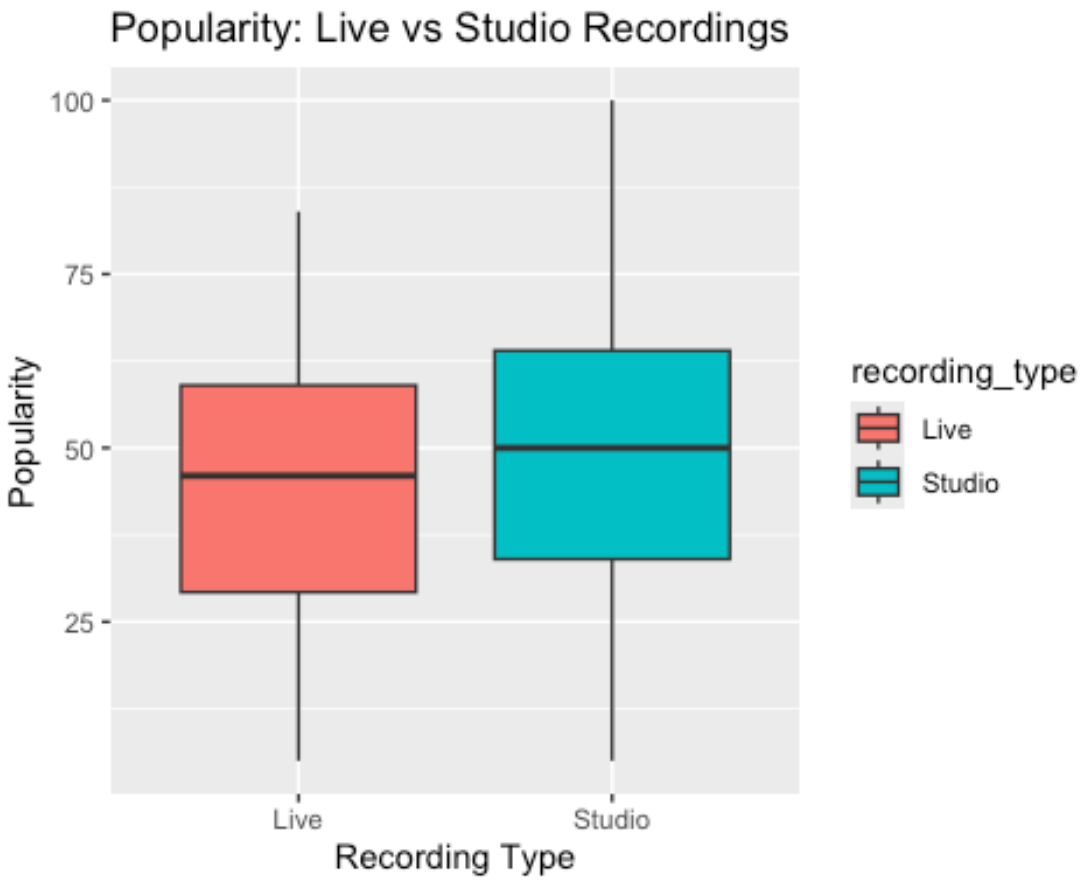
Popularity by subgenre

```
ggplot(spotify_popularity, aes(x = playlist_subgenre, y = track_popularity,
fill = playlist_subgenre)) +
  geom_boxplot(outlier.alpha = 0.2) +
  labs(title = "Track Popularity Across Playlist Subgenres",
       x = "Playlist Subgenre", y = "Track Popularity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



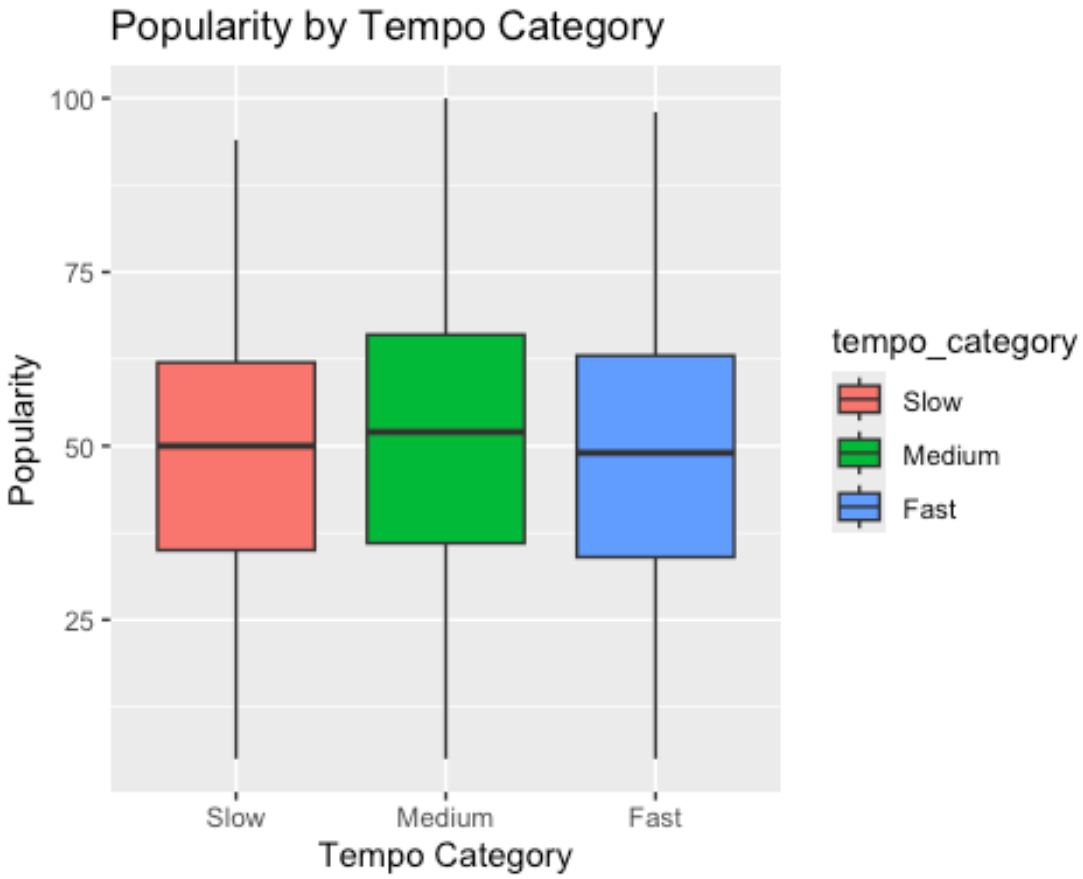
Live vs Studio

```
spotify_popularity <- spotify_popularity %>%
  mutate(recording_type = ifelse(liveness > 0.8, "Live", "Studio"))

ggplot(spotify_popularity, aes(x = recording_type, y = track_popularity, fill
= recording_type)) +
  geom_boxplot() +
  labs(title = "Popularity: Live vs Studio Recordings", x = "Recording Type",
y = "Popularity")
```



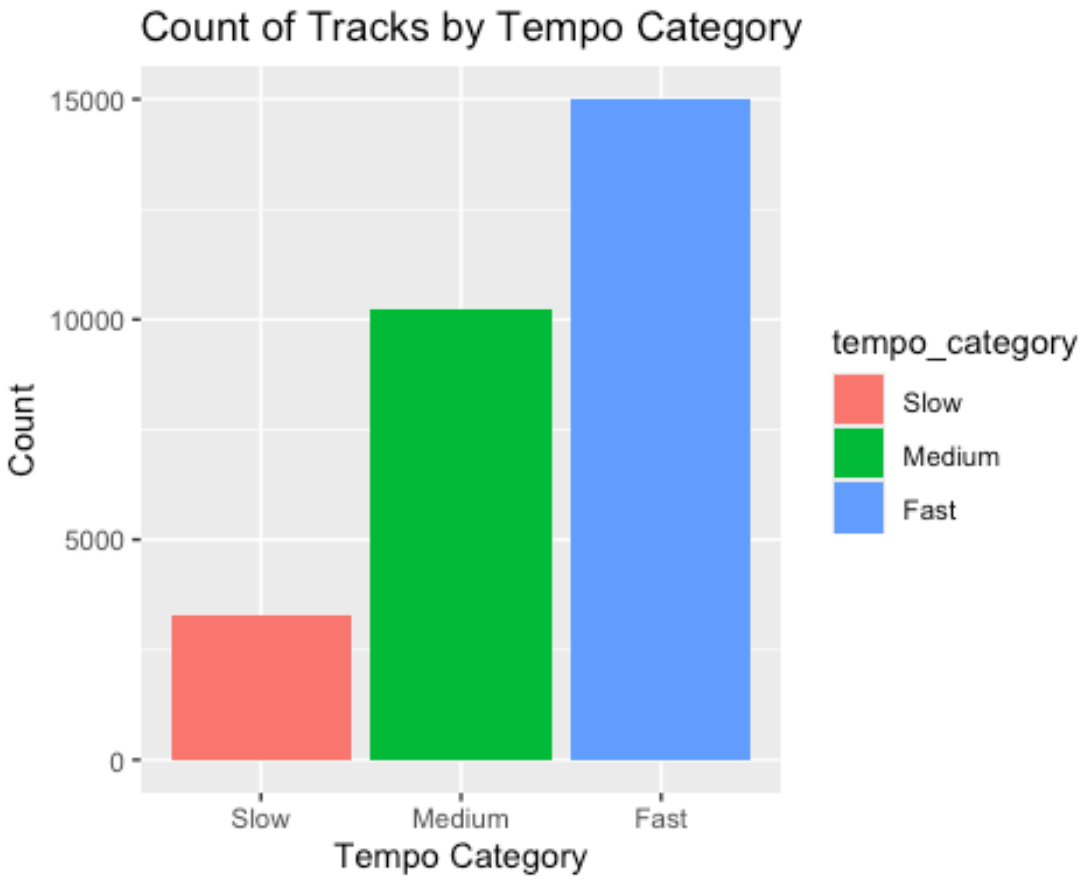
popularity by tempo

```
spotify_popularity <- spotify_popularity %>%
  mutate(tempo_category = cut(tempo, breaks = c(0, 90, 120, Inf),
                              labels = c("Slow", "Medium", "Fast")))

ggplot(spotify_popularity, aes(x = tempo_category, y = track_popularity, fill
= tempo_category)) +
  geom_boxplot() +
  labs(title = "Popularity by Tempo Category", x = "Tempo Category", y =
"Popularity")
```



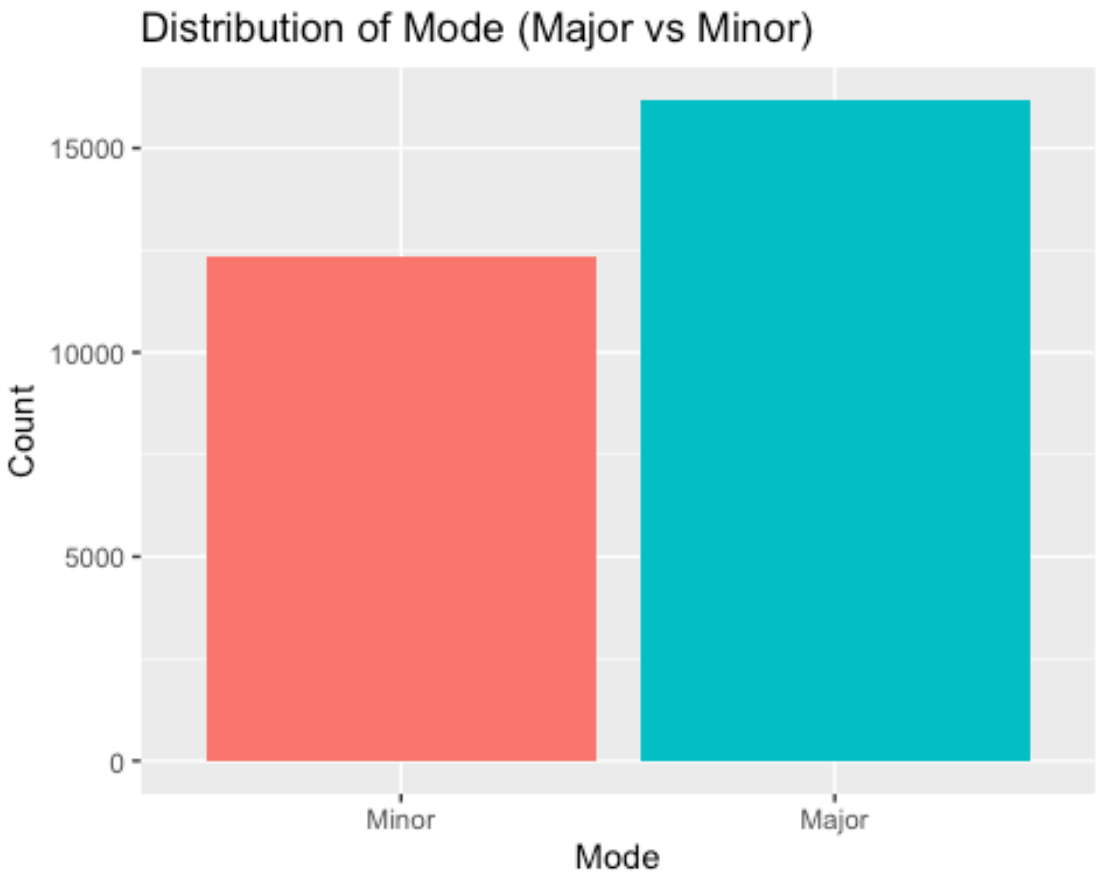
Track Count by Tempo

```
spotify_popularity <- spotify_popularity %>%
  mutate(tempo_category = cut(tempo, breaks = c(0, 90, 120, Inf),
                              labels = c("Slow", "Medium", "Fast")))

ggplot(spotify_popularity, aes(x = tempo_category, fill = tempo_category)) +
  geom_bar() +
  labs(title = "Count of Tracks by Tempo Category", x = "Tempo Category", y =
"Count")
```



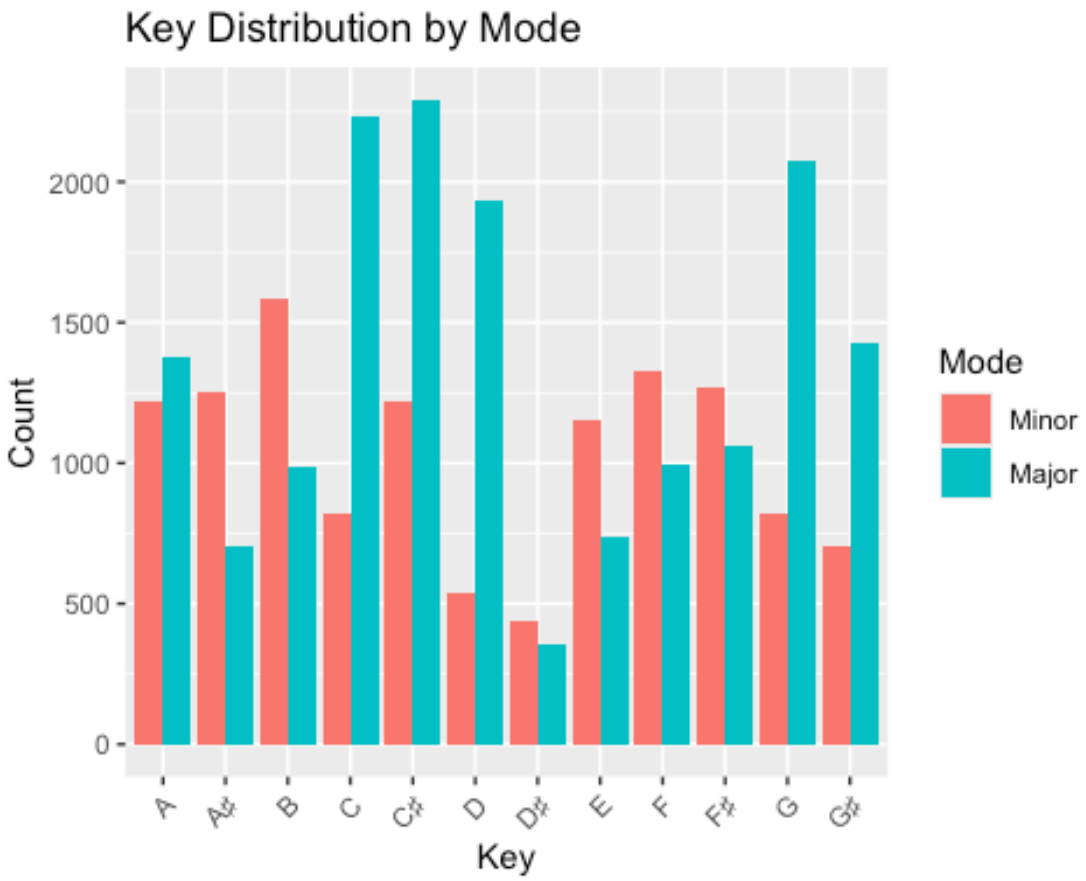
Mode Distribution

```
ggplot(spotify_popularity, aes(x = factor(mode, labels = c("Minor",
"Major")), fill = factor(mode))) +
  geom_bar() +
  labs(title = "Distribution of Mode (Major vs Minor)", x = "Mode", y =
"Count") +
  theme(legend.position = "none")
```



Key DIstribution by Mode

```
ggplot(spotify_popularity, aes(x = key_label, fill = factor(mode, labels = c("Minor", "Major")))) +
  geom_bar(position = "dodge") +
  labs(title = "Key Distribution by Mode", x = "Key", y = "Count", fill = "Mode") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Popularity by Key and Mode

```
avg_popularity_key_mode <- spotify_popularity %>%
  group_by(key_label, mode = factor(mode, labels = c("Minor", "Major"))) %>%
  summarize(avg_popularity = mean(track_popularity, na.rm = TRUE)) %>%
  ungroup()

## `summarise()` has grouped output by 'key_label'. You can override using
the
## `.groups` argument.

ggplot(avg_popularity_key_mode, aes(x = key_label, y = mode, fill =
avg_popularity)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Average Popularity by Key and Mode", x = "Key", y = "Mode",
fill = "Popularity") +
  theme_minimal()
```