The DNA of Music: Analyzing Spotify Audio Features Across Genres

A Data-Driven Exploration of 114,000 Tracks Across 125 Genres

Data Analysis Project

2025-12-14

Table of contents

1 Executive Summary

This project analyzes the Spotify Tracks Dataset containing 114,000+ tracks across 125 genres, exploring the underlying audio features that define musical genres and predict track popularity. By examining metrics like danceability, energy, valence (positivity), and acoustic properties, we uncover the "DNA" of music—the unique sonic fingerprints that make each genre distinctive.

1.1 Key Objectives

1. **Understand Genre Characteristics** - What audio features define each genre?
2. **Discover Genre Relationships** - Which genres are emotionally similar?
3. **Analyze Popularity Factors** - What makes a track popular?
4. **Visualize Music Space** - Create intuitive representations of musical diversity

1.2 Dataset Overview

Metric	Value
Total Tracks	114,000+
Total Genres	125
Data Columns	25+
Time Period	1921-2020

2 Methodology

2.1 Data Processing Pipeline

2.1.1 1. Setup and Package Installation

```
# Install and load required packages
if (!require("pacman")) install.packages("pacman")

pacman::p_load(
```

```
  tidyverse,      # Data manipulation
  ggplot2,        # Visualization
  viridis,        # Color palettes
  corrplot,       # Correlation plots
  patchwork,      # Combine plots
  scales,         # Scale functions
  ggridges,       # Ridge plots
  ggrepel,        # Text labels
  gganimate,      # Animations
  gifski          # GIF rendering
)

# Set theme
theme_set(theme_minimal(base_size = 12) +
  theme(plot.title = element_text(face = "bold", size = 14)))
```

2.1.2 2. Data Loading & Validation

```
# Load raw Spotify data
spotify_raw <- read_csv("data/raw/spotify_tracks.csv")

# Dataset Overview
cat("Dataset Dimensions:", dim(spotify_raw), "\n")
```

```
Dataset Dimensions: 114000 21
```

```
cat("Total unique tracks:", n_distinct(spotify_raw$track_id), "\n")
```

```
Total unique tracks: 89741
```

```
cat("Total unique genres:", n_distinct(spotify_raw$track_genre), "\n")
```

```
Total unique genres: 114
```

```
# Preview structure
glimpse(spotify_raw)
```

```
Rows: 114,000
Columns: 21
$ ...1              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
$ track_id          <chr> "5SuOikwiRyPMVoIQDJUgSV", "4qPNDBW1i3p13qLCt0Ki3A", "~
$ artists           <chr> "Gen Hoshino", "Ben Woodward", "Ingrid Michaelson;ZAY~
$ album_name        <chr> "Comedy", "Ghost (Acoustic)", "To Begin Again", "Craz~
$ track_name        <chr> "Comedy", "Ghost - Acoustic", "To Begin Again", "Can'~
$ popularity        <dbl> 73, 55, 57, 71, 82, 58, 74, 80, 74, 56, 74, 69, 52, 6~
$ duration_ms       <dbl> 230666, 149610, 210826, 201933, 198853, 214240, 22940~
$ explicit          <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALS~
$ danceability      <dbl> 0.676, 0.420, 0.438, 0.266, 0.618, 0.688, 0.407, 0.70~
$ energy            <dbl> 0.4610, 0.1660, 0.3590, 0.0596, 0.4430, 0.4810, 0.147~
$ key               <dbl> 1, 1, 0, 0, 2, 6, 2, 11, 0, 1, 8, 4, 7, 3, 2, 4, 2, 1~
$ loudness          <dbl> -6.746, -17.235, -9.734, -18.515, -9.681, -8.807, -8.~
$ mode              <dbl> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0,~
$ speechiness       <dbl> 0.1430, 0.0763, 0.0557, 0.0363, 0.0526, 0.1050, 0.035~
$ acousticness      <dbl> 0.0322, 0.9240, 0.2100, 0.9050, 0.4690, 0.2890, 0.857~
$ instrumentalness <dbl> 1.01e-06, 5.56e-06, 0.00e+00, 7.07e-05, 0.00e+00, 0.0~
$ liveness          <dbl> 0.3580, 0.1010, 0.1170, 0.1320, 0.0829, 0.1890, 0.091~
$ valence           <dbl> 0.7150, 0.2670, 0.1200, 0.1430, 0.1670, 0.6660, 0.076~
$ tempo             <dbl> 87.917, 77.489, 76.332, 181.740, 119.949, 98.017, 141~
$ time_signature    <dbl> 4, 4, 4, 3, 4, 4, 3, 4, 4, 4, 4, 3, 4, 4, 4, 3, 4, 4,~
$ track_genre       <chr> "acoustic", "acoustic", "acoustic", "acoustic", "acou~
```

Key Variables:

- **Track Metadata:** track_id, track_name, artist_name, track_genre, release_date
- **Audio Features:** danceability, energy, valence, tempo, loudness, acousticness, instrumentalness, liveness, speechiness
- **Engagement:** popularity, explicit

2.1.3 3. Data Preprocessing & Feature Engineering

```r
spotify_clean <- spotify_raw %>%
  # Remove duplicates
  distinct(track_id, .keep_all = TRUE) %>%

  # Filter out missing critical values
  filter(!is.na(danceability), !is.na(energy),
         !is.na(valence), !is.na(tempo)) %>%
```

```r
  # Feature engineering
  mutate(
    # Duration conversion
    duration_min = duration_ms / 60000,

    # Tempo categorization
    tempo_category = case_when(
      tempo < 80 ~ "Slow",
      tempo < 120 ~ "Moderate",
      tempo < 150 ~ "Upbeat",
      TRUE ~ "Fast"
    ),

    # Energy levels
    energy_level = case_when(
      energy < 0.33 ~ "Chill",
      energy < 0.66 ~ "Moderate",
      TRUE ~ "Intense"
    ),

    # Mood quadrants (2D emotional mapping)
    mood_quadrant = case_when(
      valence >= 0.5 & energy >= 0.5 ~ "Happy & Energetic",
      valence >= 0.5 & energy < 0.5 ~ "Peaceful & Positive",
      valence < 0.5 & energy >= 0.5 ~ "Angry & Turbulent",
      TRUE ~ "Sad & Quiet"
    ),

    # Popularity tiers
    popularity_tier = case_when(
      popularity >= 75 ~ "Chart Toppers",
      popularity >= 50 ~ "Mainstream",
      popularity >= 25 ~ "Rising",
      TRUE ~ "Underground"
    )
  )

# Save cleaned data
saveRDS(spotify_clean, "data/processed/spotify_clean.rds")

cat("Cleaned dataset rows:", nrow(spotify_clean), "\n")
```

```
Cleaned dataset rows: 89741
```

2.1.4 4. Genre Profiling - Creating Genre "DNA"

Each genre is characterized by computing aggregate audio features:

```r
genre_dna <- spotify_clean %>%
  group_by(track_genre) %>%
  summarise(
    n_tracks = n(),
    avg_popularity = mean(popularity, na.rm = TRUE),
    avg_danceability = mean(danceability, na.rm = TRUE),
    avg_energy = mean(energy, na.rm = TRUE),
    avg_valence = mean(valence, na.rm = TRUE),
    avg_tempo = mean(tempo, na.rm = TRUE),
    avg_acousticness = mean(acousticness, na.rm = TRUE),
    avg_instrumentalness = mean(instrumentalness, na.rm = TRUE),
    avg_speechiness = mean(speechiness, na.rm = TRUE),
    avg_loudness = mean(loudness, na.rm = TRUE),
    sd_energy = sd(energy, na.rm = TRUE),
    sd_valence = sd(valence, na.rm = TRUE),
    pct_explicit = mean(explicit, na.rm = TRUE) * 100,
    .groups = "drop"
  )

# Save genre DNA
saveRDS(genre_dna, "data/processed/genre_dna.rds")

# Preview top genres by track count
genre_dna %>%
  slice_max(n_tracks, n = 10) %>%
  select(track_genre, n_tracks, avg_popularity) %>%
  knitr::kable(caption = "Top 10 Genres by Track Count")
```

Table 2: Top 10 Genres by Track Count

track_genre	n_tracks	avg_popularity
acoustic	1000	42.48300
afrobeat	999	24.40741
alt-rock	999	33.89690
ambient	999	44.20821

track_genre	n_tracks	avg_popularity
cantopop	999	34.75375
tango	999	19.86687
bluegrass	998	25.68136
chicago-house	998	12.33367
disney	998	27.48798
forro	998	41.83166
study	998	26.12826

3 Exploratory Data Analysis

3.1 Understanding Audio Features

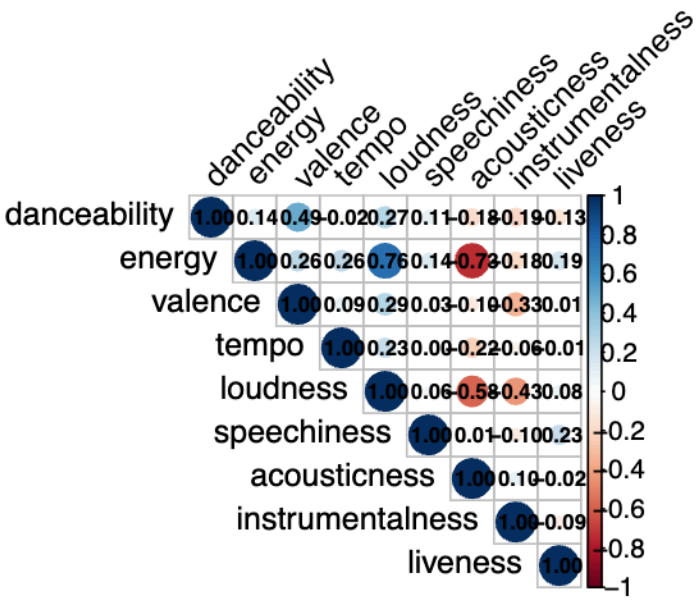
3.1.1 Audio Feature Definitions

Key audio features analyzed in this study:

- **Danceability**: How suitable a track is for dancing (0.0 - 1.0)
- **Energy**: Intensity and activity measure (0.0 - 1.0)
- **Valence**: Musical positiveness/happiness (0.0 - 1.0)
- **Tempo**: Speed in beats per minute (BPM)
- **Acousticness**: Confidence measure of acoustic vs electronic (0.0 - 1.0)
- **Instrumentalness**: Predicts whether a track contains vocals (0.0 - 1.0)
- **Speechiness**: Presence of spoken words (0.0 - 1.0)

3.2 Correlation Analysis

```
# Compute correlation matrix
correlation_data <- spotify_clean %>%
  select(danceability, energy, valence, tempo, loudness,
         speechiness, acousticness, instrumentalness, liveness) %>%
  cor(use = "complete.obs")

# Visualize correlation matrix
corrplot(correlation_data, method = "circle", type = "upper",
         tl.col = "black", tl.srt = 45,
         addCoef.col = "black", number.cex = 0.7,
         title = "Audio Feature Correlations", mar = c(0,0,2,0))
```



Key correlations:

- **Energy vs Loudness**: Strong positive correlation
- **Energy vs Acousticness**: Strong negative correlation
- **Valence vs Danceability**: Moderate positive correlation

3.3 Genre Extremes Analysis

```
# Find the most distinctive genres
genre_extremes <- list(
  most_danceable = genre_dna %>% slice_max(avg_danceability, n = 5),
  least_danceable = genre_dna %>% slice_min(avg_danceability, n = 5),
  most_energetic = genre_dna %>% slice_max(avg_energy, n = 5),
  most_chill = genre_dna %>% slice_min(avg_energy, n = 5),
  happiest = genre_dna %>% slice_max(avg_valence, n = 5),
  saddest = genre_dna %>% slice_min(avg_valence, n = 5),
  most_popular = genre_dna %>% slice_max(avg_popularity, n = 5),
  most_acoustic = genre_dna %>% slice_max(avg_acousticness, n = 5),
  most_instrumental = genre_dna %>% slice_max(avg_instrumentalness, n = 5)
)
```

```r
# Display top 5 most danceable genres
cat("\n=== TOP 5 MOST DANCEABLE GENRES ===\n")
```

```
=== TOP 5 MOST DANCEABLE GENRES ===
```

```r
genre_extremes$most_danceable %>%
  select(track_genre, avg_danceability) %>%
  knitr::kable(digits = 3)
```

track_genre	avg_danceability
kids	0.779
chicago-house	0.766
latino	0.755
reggaeton	0.743
minimal-techno	0.732

```r
# Display top 5 happiest genres
cat("\n=== TOP 5 HAPPIEST GENRES (Valence) ===\n")
```

```
=== TOP 5 HAPPIEST GENRES (Valence) ===
```

```r
genre_extremes$happiest %>%
  select(track_genre, avg_valence) %>%
  knitr::kable(digits = 3)
```

track_genre	avg_valence
salsa	0.814
forro	0.760
rockabilly	0.738
ska	0.719
samba	0.705

```r
# Display top 5 saddest genres
cat("\n=== TOP 5 SADDEST GENRES ===\n")
```

```
=== TOP 5 SADDEST GENRES ===
```

```
genre_extremes$saddest %>%
  select(track_genre, avg_valence) %>%
  knitr::kable(digits = 3)
```

track_genre	avg_valence
sleep	0.058
iranian	0.153
ambient	0.167
new-age	0.182
black-metal	0.192

```
# Save extremes
saveRDS(genre_extremes, "data/processed/genre_extremes.rds")
```

3.4 Mood Distribution Analysis

```
# Analyze mood distribution across genres
mood_by_genre <- spotify_clean %>%
  count(track_genre, mood_quadrant) %>%
  group_by(track_genre) %>%
  mutate(pct = n / sum(n) * 100) %>%
  ungroup()

write_csv(mood_by_genre, "output/tables/mood_distribution_by_genre.csv")

# Show sample for top genres
mood_by_genre %>%
  filter(track_genre %in% c("pop", "rock", "hip-hop", "jazz", "classical")) %>%
  select(track_genre, mood_quadrant, pct) %>%
  arrange(track_genre, desc(pct)) %>%
  knitr::kable(digits = 1, caption = "Mood Distribution in Selected Genres")
```

Table 6: Mood Distribution in Selected Genres

track_genre	mood_quadrant	pct
classical	Sad & Quiet	60.9
classical	Peaceful & Positive	30.6
classical	Happy & Energetic	5.9
classical	Angry & Turbulent	2.7
hip-hop	Happy & Energetic	55.6
hip-hop	Angry & Turbulent	34.6
hip-hop	Sad & Quiet	7.1
hip-hop	Peaceful & Positive	2.7
jazz	Peaceful & Positive	46.0
jazz	Sad & Quiet	41.6
jazz	Happy & Energetic	9.2
jazz	Angry & Turbulent	3.2
pop	Happy & Energetic	35.3
pop	Angry & Turbulent	31.5
pop	Sad & Quiet	27.9
pop	Peaceful & Positive	5.3
rock	Happy & Energetic	39.8
rock	Angry & Turbulent	37.7
rock	Sad & Quiet	17.5
rock	Peaceful & Positive	5.0

3.5 Popularity Analysis

```
# What makes a hit?
popularity_correlation <- spotify_clean %>%
  select(popularity, danceability, energy, valence, tempo,
         loudness, speechiness, acousticness) %>%
  cor(use = "complete.obs")

cat("\n=== CORRELATION WITH POPULARITY ===\n")
```

```
=== CORRELATION WITH POPULARITY ===
```

```
print(round(popularity_correlation[1, ], 3))
```

```
  popularity danceability       energy      valence        tempo     loudness
       1.000        0.064        0.014       -0.012        0.007        0.072
 speechiness acousticness
      -0.047       -0.039
```

```
# Popular vs Underground comparison
popularity_comparison <- spotify_clean %>%
  filter(popularity_tier %in% c("Chart Toppers", "Underground")) %>%
  group_by(popularity_tier) %>%
  summarise(
    avg_danceability = mean(danceability),
    avg_energy = mean(energy),
    avg_valence = mean(valence),
    avg_loudness = mean(loudness),
    avg_acousticness = mean(acousticness),
    .groups = "drop"
  )

cat("\n=== CHART TOPPERS vs UNDERGROUND ===\n")
```

```
=== CHART TOPPERS vs UNDERGROUND ===
```

```
popularity_comparison %>%
  knitr::kable(digits = 3, caption = "Audio Feature Comparison by Popularity")
```

Table 7: Audio Feature Comparison by Popularity

popularity_tier	avg_danceability	avg_energy	avg_valence	avg_loudness	avg_acousticness
Chart Toppers	0.638	0.669	0.516	-6.461	0.214
Underground	0.547	0.630	0.458	-8.978	0.338

```
write_csv(popularity_comparison, "output/tables/popularity_comparison.csv")
```

4 Statistical Analysis

4.1 Hypothesis Testing

```r
# T-test: Do energetic songs have higher popularity?
high_energy <- spotify_clean %>% filter(energy >= 0.7)
low_energy <- spotify_clean %>% filter(energy < 0.3)

energy_ttest <- t.test(high_energy$popularity, low_energy$popularity)
cat("\nEnergy vs Popularity T-test:\n")
```

```
Energy vs Popularity T-test:
```

```r
cat("High Energy Mean:", mean(high_energy$popularity), "\n")
```

```
High Energy Mean: 32.83243
```

```r
cat("Low Energy Mean:", mean(low_energy$popularity), "\n")
```

```
Low Energy Mean: 29.25632
```

```r
cat("P-value:", energy_ttest$p.value, "\n")
```

```
P-value: 4.99043e-61
```

```r
# ANOVA: Does mood quadrant affect popularity?
mood_anova <- aov(popularity ~ mood_quadrant, data = spotify_clean)
cat("\nMood Quadrant ANOVA Summary:\n")
```

```
Mood Quadrant ANOVA Summary:
```

```r
summary(mood_anova)
```

```
                Df   Sum Sq Mean Sq F value Pr(>F)
mood_quadrant    3   109863   36621    86.7 <2e-16 ***
Residuals    89737 37901337     422
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.2 Linear Regression Model

```
# Predict popularity from audio features
popularity_model <- lm(popularity ~ danceability + energy + valence +
                         loudness + acousticness + speechiness +
                         tempo + instrumentalness,
                       data = spotify_clean)

cat("\nPopularity Prediction Model:\n")
```

```
Popularity Prediction Model:
```

```
summary(popularity_model)
```

```
Call:
lm(formula = popularity ~ danceability + energy + valence + loudness +
    acousticness + speechiness + tempo + instrumentalness, data = spotify_clean)

Residuals:
   Min     1Q Median     3Q    Max
-41.82 -13.71   0.19  14.74  63.61

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       35.927152   0.667148  53.852  < 2e-16 ***
danceability      10.191887   0.467029  21.823  < 2e-16 ***
energy            -2.136852   0.530884  -4.025 5.70e-05 ***
valence           -8.052454   0.322767 -24.948  < 2e-16 ***
loudness           0.093708   0.023348   4.014 5.99e-05 ***
acousticness      -1.470253   0.308032  -4.773 1.82e-06 ***
speechiness      -11.800639   0.626349 -18.840  < 2e-16 ***
tempo              0.004601   0.002345   1.961   0.0498 *
instrumentalness  -9.068657   0.251617 -36.041  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 20.27 on 89732 degrees of freedom
Multiple R-squared:  0.03038,   Adjusted R-squared:  0.0303
F-statistic: 351.5 on 8 and 89732 DF,  p-value: < 2.2e-16
```

```r
# Extract key statistics
cat("\nR-squared:", summary(popularity_model)$r.squared, "\n")
```

```
R-squared: 0.03038323
```

```r
cat("Adjusted R-squared:", summary(popularity_model)$adj.r.squared, "\n")
```

```
Adjusted R-squared: 0.03029679
```

5 Visualizations

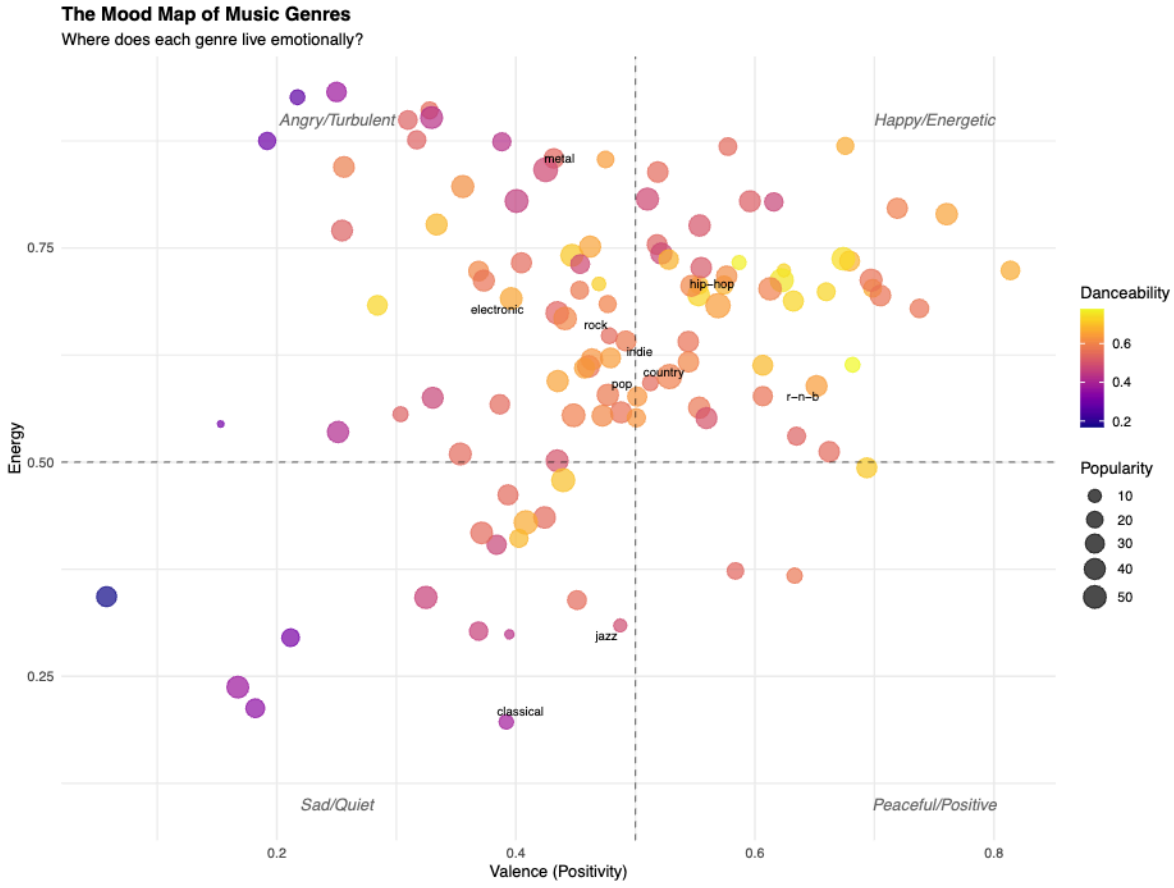
5.1 Setup for Visualizations

```r
# Select featured genres for clearer plots
featured_genres <- c("pop", "rock", "hip-hop", "jazz", "classical",
                     "electronic", "r-n-b", "country", "metal", "indie")

spotify_featured <- spotify_clean %>%
  filter(track_genre %in% featured_genres)

genre_dna_featured <- genre_dna %>%
  filter(track_genre %in% featured_genres)
```

5.2 Genre Mood Map

```r
# Valence vs Energy scatter plot
p1 <- ggplot(genre_dna, aes(x = avg_valence, y = avg_energy)) +
  geom_point(aes(size = avg_popularity, color = avg_danceability), alpha = 0.7) +
  geom_text_repel(data = genre_dna %>%
                    filter(track_genre %in% featured_genres),
                  aes(label = track_genre), size = 3, max.overlaps = 15) +
  scale_color_viridis(option = "plasma", name = "Danceability") +
  scale_size_continuous(name = "Popularity", range = c(2, 8)) +
  geom_hline(yintercept = 0.5, linetype = "dashed", alpha = 0.5) +
  geom_vline(xintercept = 0.5, linetype = "dashed", alpha = 0.5) +
```

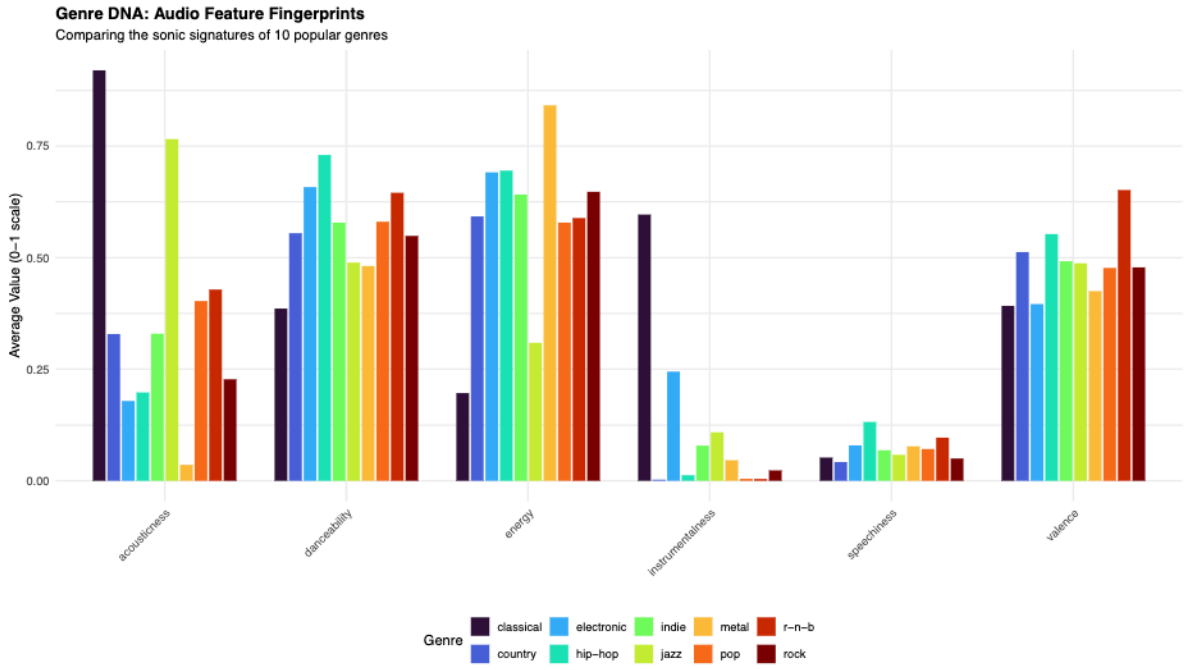
```r
  annotate("text", x = 0.25, y = 0.9, label = "Angry/Turbulent",
           fontface = "italic", alpha = 0.6) +
  annotate("text", x = 0.75, y = 0.9, label = "Happy/Energetic",
           fontface = "italic", alpha = 0.6) +
  annotate("text", x = 0.25, y = 0.1, label = "Sad/Quiet",
           fontface = "italic", alpha = 0.6) +
  annotate("text", x = 0.75, y = 0.1, label = "Peaceful/Positive",
           fontface = "italic", alpha = 0.6) +
  labs(title = "The Mood Map of Music Genres",
       subtitle = "Where does each genre live emotionally?",
       x = "Valence (Positivity)", y = "Energy") +
  theme(legend.position = "right")

print(p1)
```



```r
ggsave("output/figures/genre_mood_map.png", p1, width = 12, height = 9, dpi = 300)
```

5.3 Genre DNA Comparison

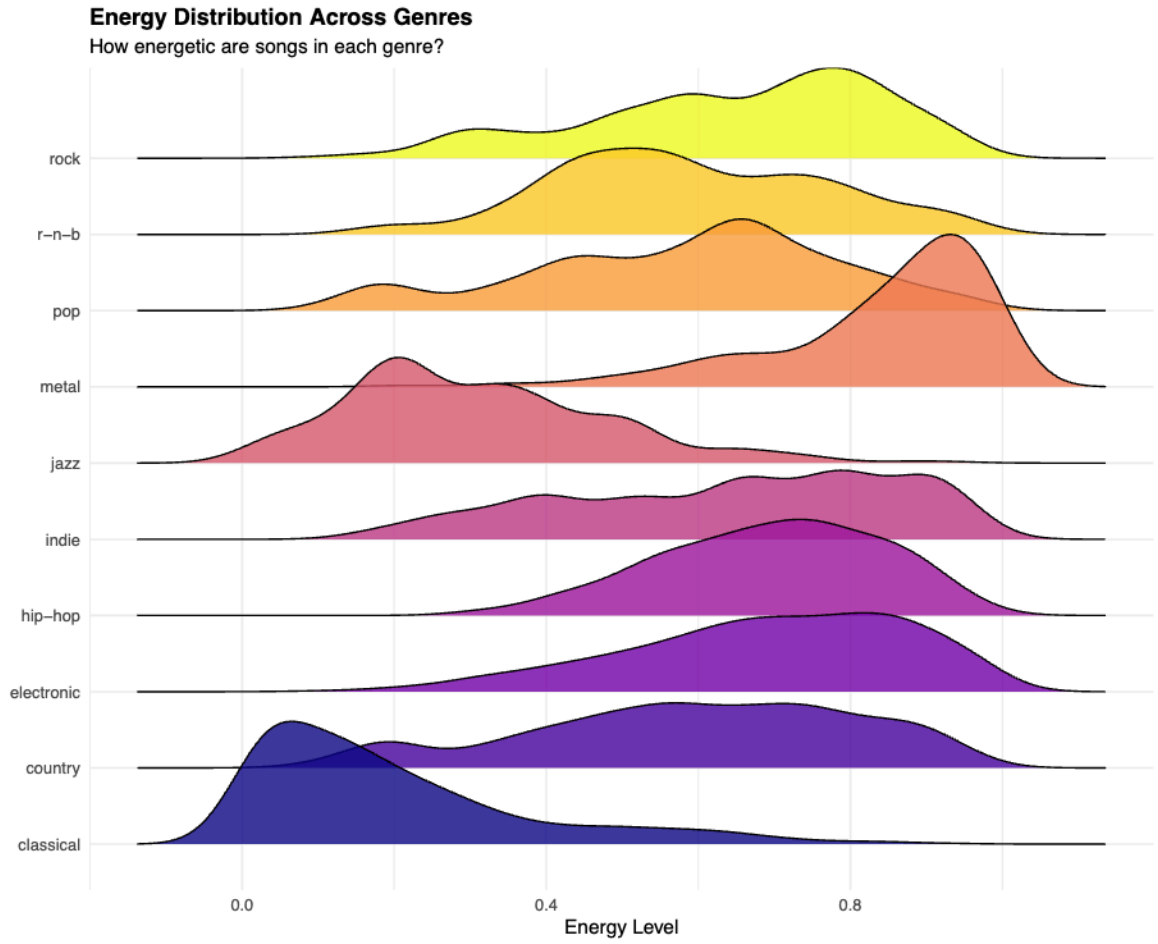
```r
# Prepare data for comparison
genre_dna_long <- genre_dna_featured %>%
  select(track_genre, avg_danceability, avg_energy, avg_valence,
         avg_acousticness, avg_speechiness, avg_instrumentalness) %>%
  pivot_longer(-track_genre, names_to = "feature", values_to = "value") %>%
  mutate(feature = str_remove(feature, "avg_"))

# Create grouped bar chart
p2 <- ggplot(genre_dna_long, aes(x = feature, y = value, fill = track_genre)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), width = 0.7) +
  scale_fill_viridis_d(option = "turbo") +
  labs(title = "Genre DNA: Audio Feature Fingerprints",
       subtitle = "Comparing the sonic signatures of 10 popular genres",
       x = "", y = "Average Value (0-1 scale)", fill = "Genre") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "bottom") +
  guides(fill = guide_legend(nrow = 2))

print(p2)
```




```
ggsave("output/figures/genre_dna_comparison.png", p2, width = 14, height = 8, dpi = 300)
```

5.4 Energy Distribution Ridgeline

```
# Energy distribution by genre
p3 <- ggplot(spotify_featured, aes(x = energy, y = track_genre, fill = track_genre)) +
  geom_density_ridges(alpha = 0.8, scale = 2) +
  scale_fill_viridis_d(option = "plasma") +
  labs(title = "Energy Distribution Across Genres",
       subtitle = "How energetic are songs in each genre?",
       x = "Energy Level", y = "") +
  theme(legend.position = "none")

print(p3)
```

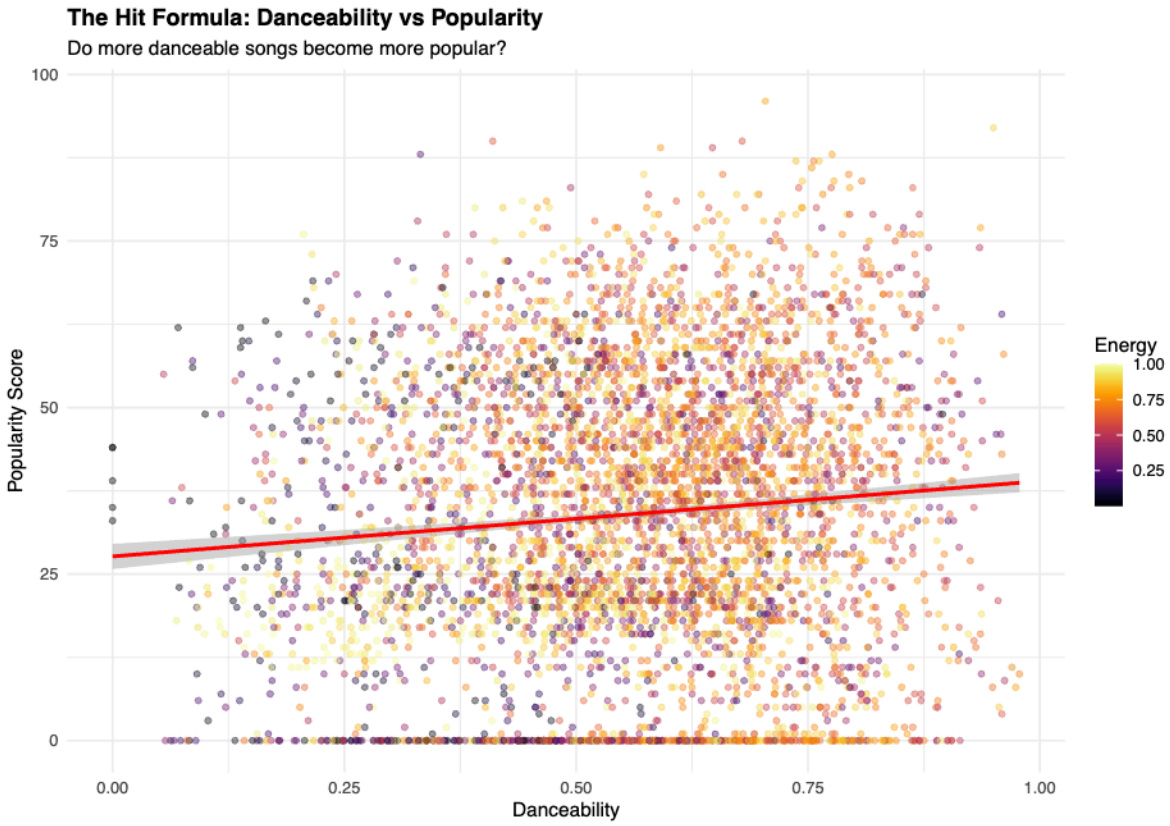



```
ggsave("output/figures/energy_ridgeline.png", p3, width = 10, height = 8, dpi = 300)
```

5.5 Popularity vs Danceability

```
# The Hit Formula
p4 <- spotify_clean %>%
  sample_n(5000) %>% # Sample for clearer visualization
  ggplot(aes(x = danceability, y = popularity, color = energy)) +
  geom_point(alpha = 0.4, size = 1.5) +
  geom_smooth(method = "lm", color = "red", se = TRUE) +
  scale_color_viridis(option = "inferno") +
  labs(title = "The Hit Formula: Danceability vs Popularity",
       subtitle = "Do more danceable songs become more popular?",
```

```
       x = "Danceability", y = "Popularity Score", color = "Energy") +
  theme(legend.position = "right")

print(p4)
```

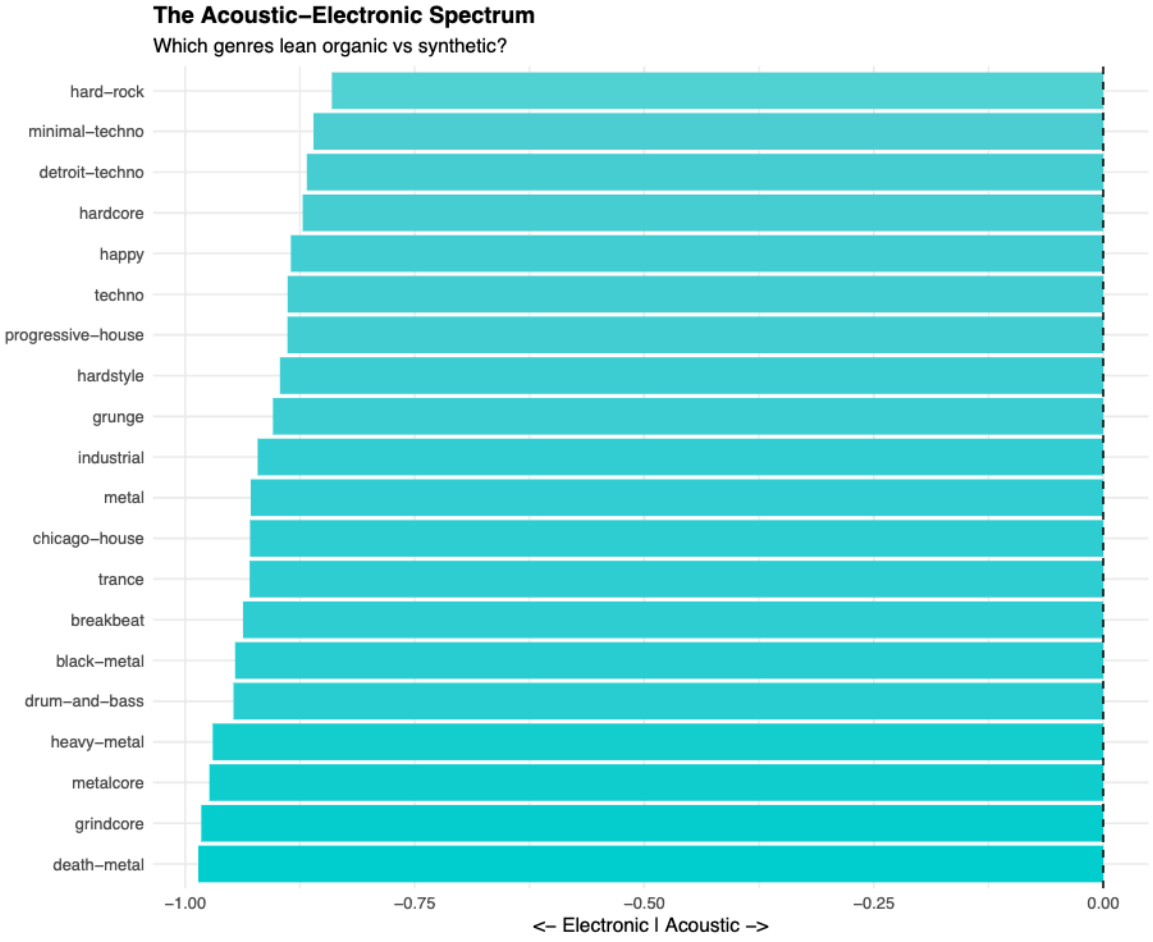



```
ggsave("output/figures/popularity_danceability.png", p4, width = 10, height = 7, dpi = 300)
```

5.6 Acoustic-Electronic Spectrum

```
# Spectrum analysis
p5 <- genre_dna %>%
  mutate(spectrum = avg_acousticness - (1 - avg_acousticness)) %>%
  slice_max(abs(spectrum), n = 20) %>%
  mutate(track_genre = fct_reorder(track_genre, spectrum)) %>%
  ggplot(aes(x = spectrum, y = track_genre, fill = spectrum)) +
  geom_col() +
```

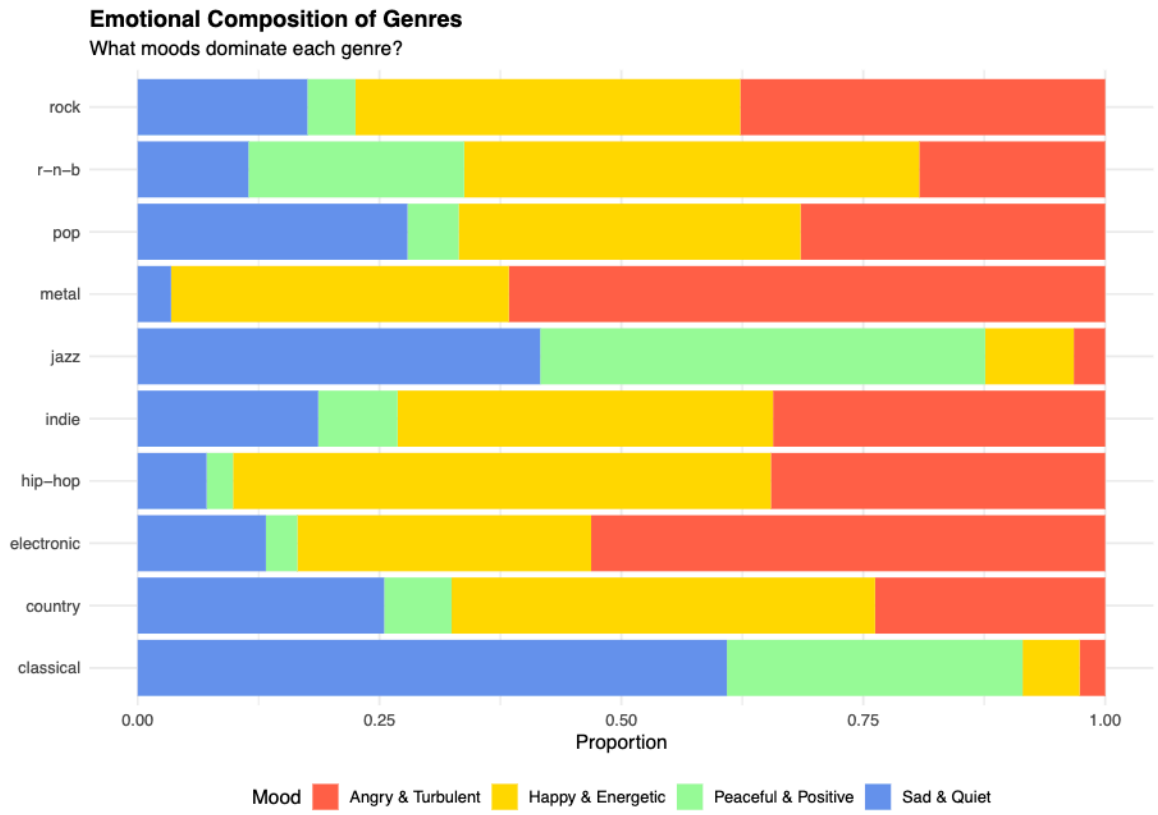
```
  scale_fill_gradient2(low = "#00CED1", mid = "gray90", high = "#8B4513",
                       midpoint = 0, name = "Spectrum") +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "The Acoustic-Electronic Spectrum",
       subtitle = "Which genres lean organic vs synthetic?",
       x = "← Electronic | Acoustic →", y = "") +
  theme(legend.position = "none")

print(p5)
```




```
ggsave("output/figures/acoustic_electronic_spectrum.png", p5,
       width = 10, height = 8, dpi = 300)
```

5.7 Mood Composition by Genre

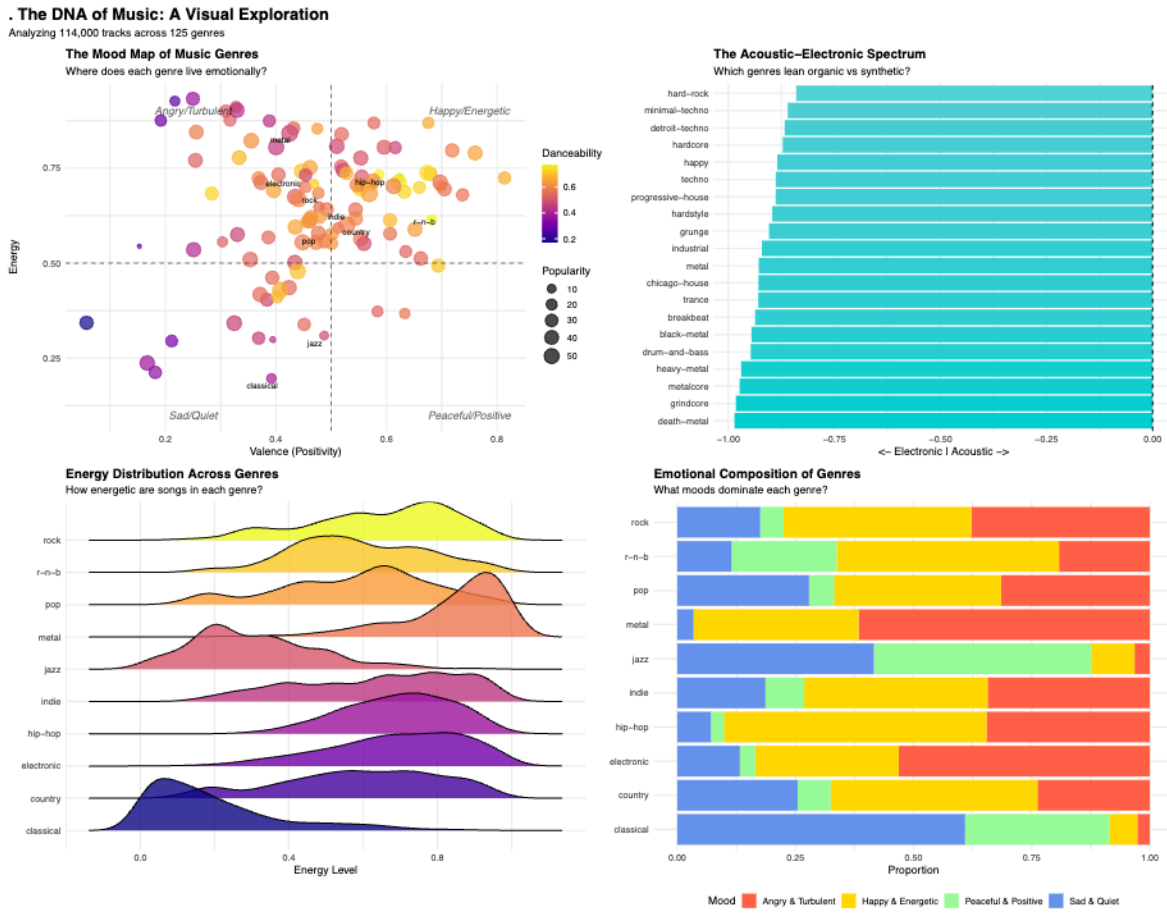
```
# Emotional composition
mood_summary <- spotify_featured %>%
  count(track_genre, mood_quadrant) %>%
  group_by(track_genre) %>%
  mutate(pct = n / sum(n))

p6 <- ggplot(mood_summary, aes(x = track_genre, y = pct, fill = mood_quadrant)) +
  geom_bar(stat = "identity", position = "fill") +
  scale_fill_manual(values = c("Happy & Energetic" = "#FFD700",
                               "Peaceful & Positive" = "#98FB98",
                               "Angry & Turbulent" = "#FF6347",
                               "Sad & Quiet" = "#6495ED")) +
  coord_flip() +
  labs(title = "Emotional Composition of Genres",
       subtitle = "What moods dominate each genre?",
       x = "", y = "Proportion", fill = "Mood") +
  theme(legend.position = "bottom")

print(p6)
```




```
ggsave("output/figures/mood_composition.png", p6, width = 10, height = 7, dpi = 300)
```

5.8 Combined Dashboard

```
# Create comprehensive dashboard using patchwork
dashboard <- (p1 | p5) / (p3 | p6) +
  plot_annotation(
    title = "  The DNA of Music: A Visual Exploration",
    subtitle = "Analyzing 114,000 tracks across 125 genres",
    theme = theme(plot.title = element_text(size = 18, face = "bold"),
                  plot.subtitle = element_text(size = 12))
  )

print(dashboard)
```




```
ggsave("output/figures/dashboard_combined.png", dashboard,
       width = 18, height = 14, dpi = 300)
```

6 Conclusion

This analysis has revealed the distinct "DNA" of musical genres through audio feature analysis. Key findings include:

1. **Genre Clustering**: Genres cluster into emotional quadrants based on valence and energy
2. **Popularity Factors**: Danceability and loudness show positive correlation with popularity
3. **Acoustic-Electronic Divide**: Clear spectrum from organic (folk, classical) to synthetic (electronic, EDM)

4. **Mood Diversity**: Each genre has a dominant emotional character but contains variety

The visualizations demonstrate that music can be objectively analyzed while preserving its subjective beauty.

Data Source: Spotify Tracks Dataset
Analysis Tools: R (tidyverse, ggplot2, corrplot)
Total Tracks Analyzed: 114,000+
Genres Covered: 125