Data Science 2025

Leonie Wegeler

Student ID No.: 400881943

Jill Safarli

Student ID No.: 400860593

Isabel viela Wetz

Student ID No.: 400920787

Lecturer: Mr. Groß & Prof. Dr. Huber

Team

Isabel Viela Wetz
IBM 3

Leonie Wegeler
IBM 3

Jill Safarli
IBM 3

Research Question

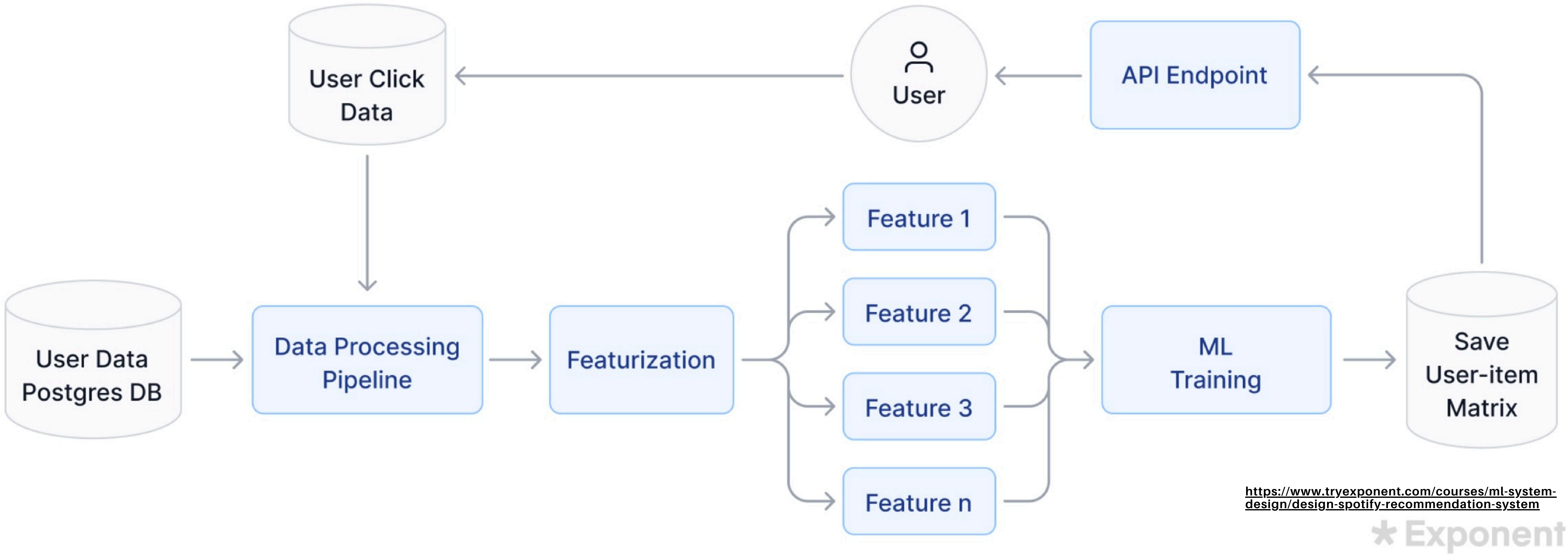
To what extent do the Spotify audio features 'danceability' and 'energy' predict a song's commercial success?

AGENDA

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES

- Spotify in a Business context
- Data Sources + Challenges
- Exploratory Data Analysis
- First conclusion
- Limitations
- Next Steps (final Report)

Spotify relevance in a Business context



https://www.tryexponent.com/courses/ml-system-design/design-spotify-recommendation-system

Data sources + used variables

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES

Spotify API

- ✗ Problem: A track ID is required for every single song → extremely time-consuming
- ✗ Tokens expire every 60 minutes → API repeatedly stops working

Conclusion: API unsuitable for large datasets

Kaggle Dataset

- ✗ Python Code
- ✓ Originally available only in Python → had to be converted to R & Quarto for our project

Listen Notes API

- ✓ Used only for general overview, not suitable for analysis

MusicBrainz API

- ✓ Useful additional data (genre, release year)
- ✗ Inconsistent data quality

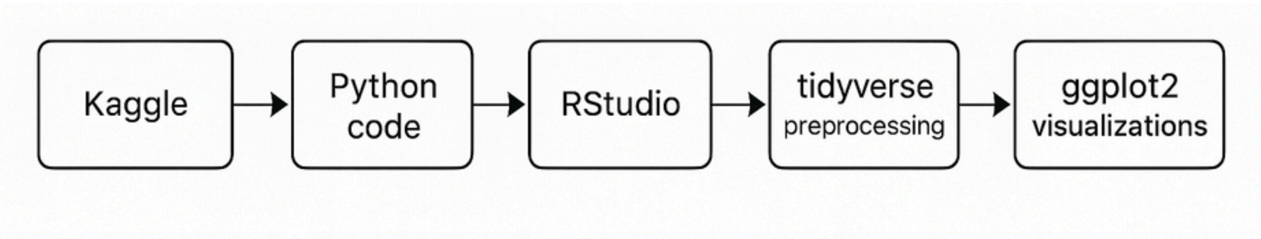
Charts Website

- ✓ Good overview of real chart data
- ✗ no audio features → not usable for analysis

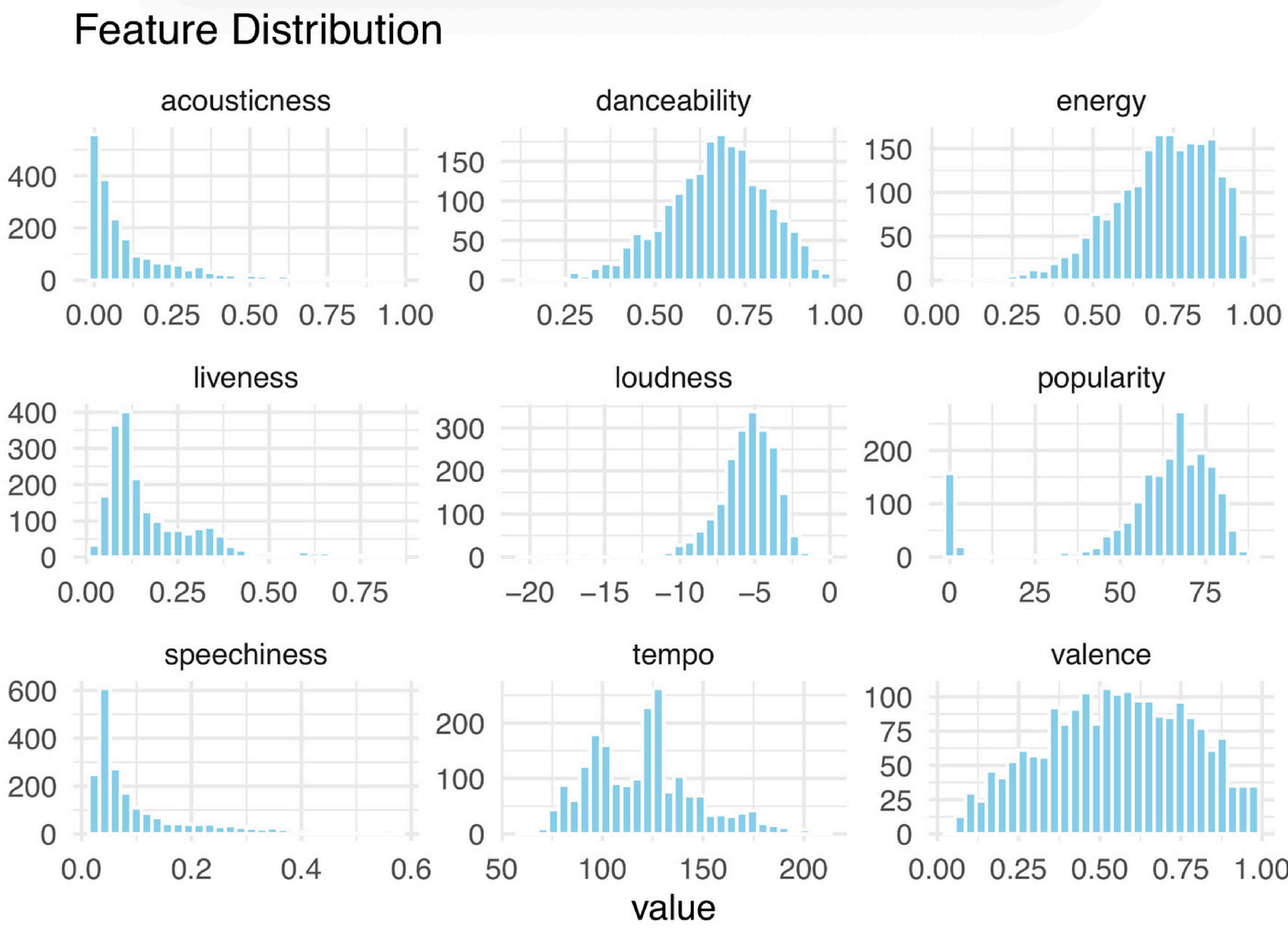
Exploratory Data Analysis

Distribution of Spotify Audio Features

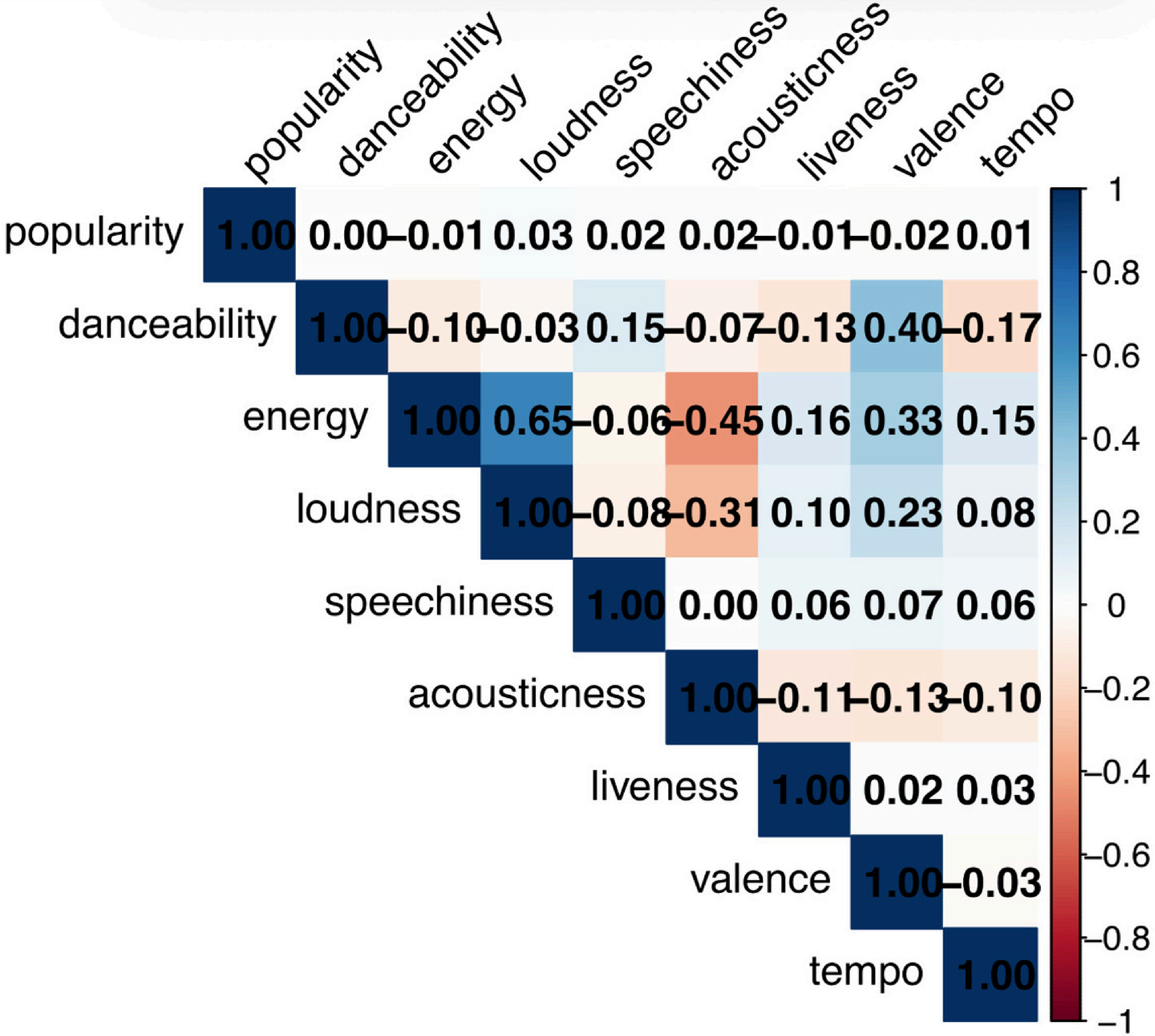
- Many features are skewed → strong clustering
- Energy & danceability show wide variation
- Popularity is widely spread → good for prediction analysis




```
> features_long <- spotify %>%
+     select(popularity, danceability, energy, loudness, speechiness,
             acousticness, liveness, valence, tempo) %>%
+     pivot_longer(cols = everything(),
+                  names_to = "feature",
+                  values_to = "value")
> > ggplot(features_long, aes(x = value)) +
+     geom_histogram(bins = 30, fill = "skyblue", color = "white") +
+     facet_wrap(~ feature, scales = "free") +
+     theme_minimal() +
+     labs(title = "Feature Distribution")
```



Correlation between Spotify Audio Features and Popularity

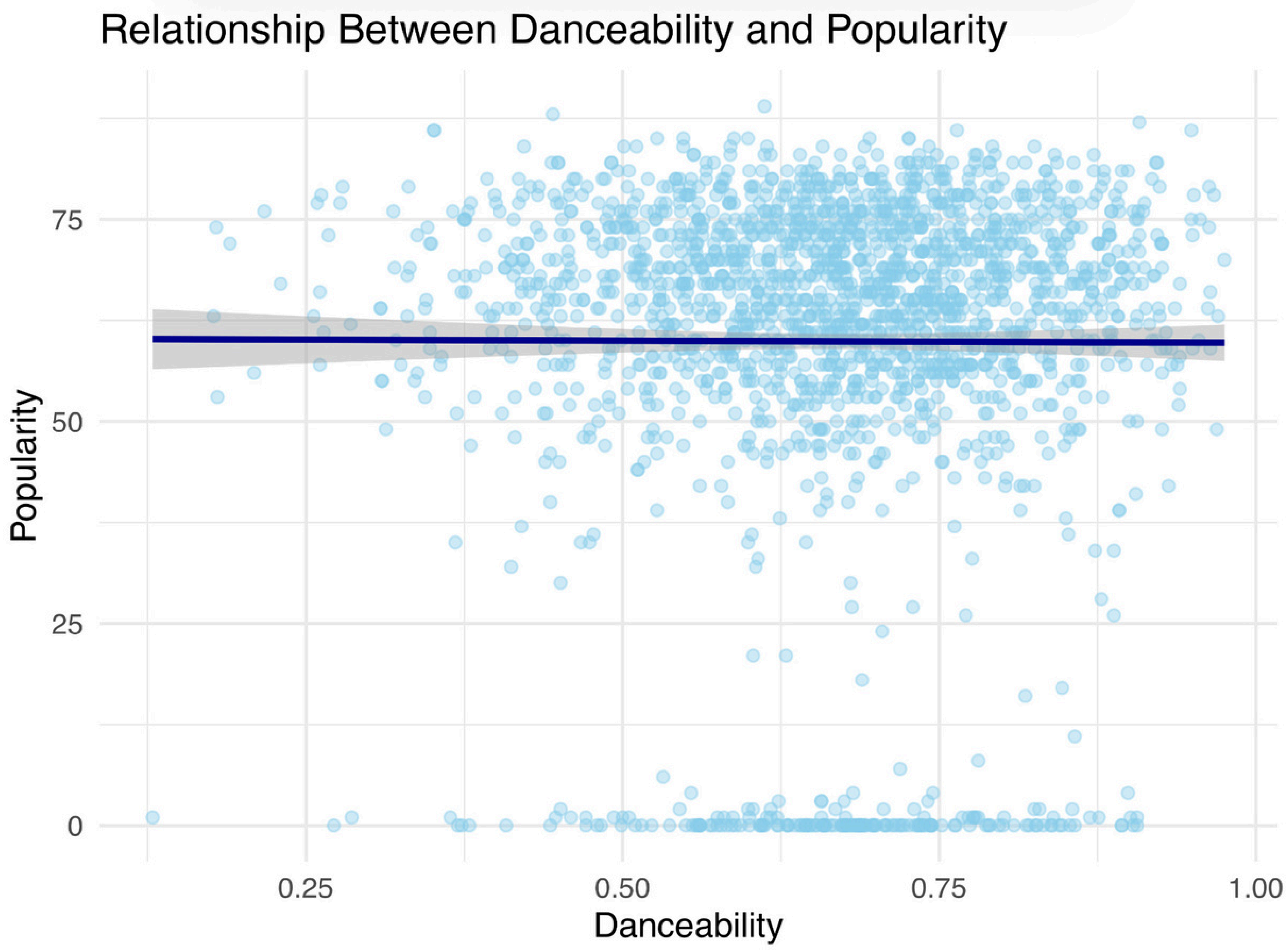


- Energy shows the strongest positive correlation with popularity
- Danceability also has a slight positive correlation with popularity
- Acousticness is negatively correlated with both energy and popularity
- Valence (positiveness) shows almost no correlation with popularity → a "happy vibe" alone does not predict a hit

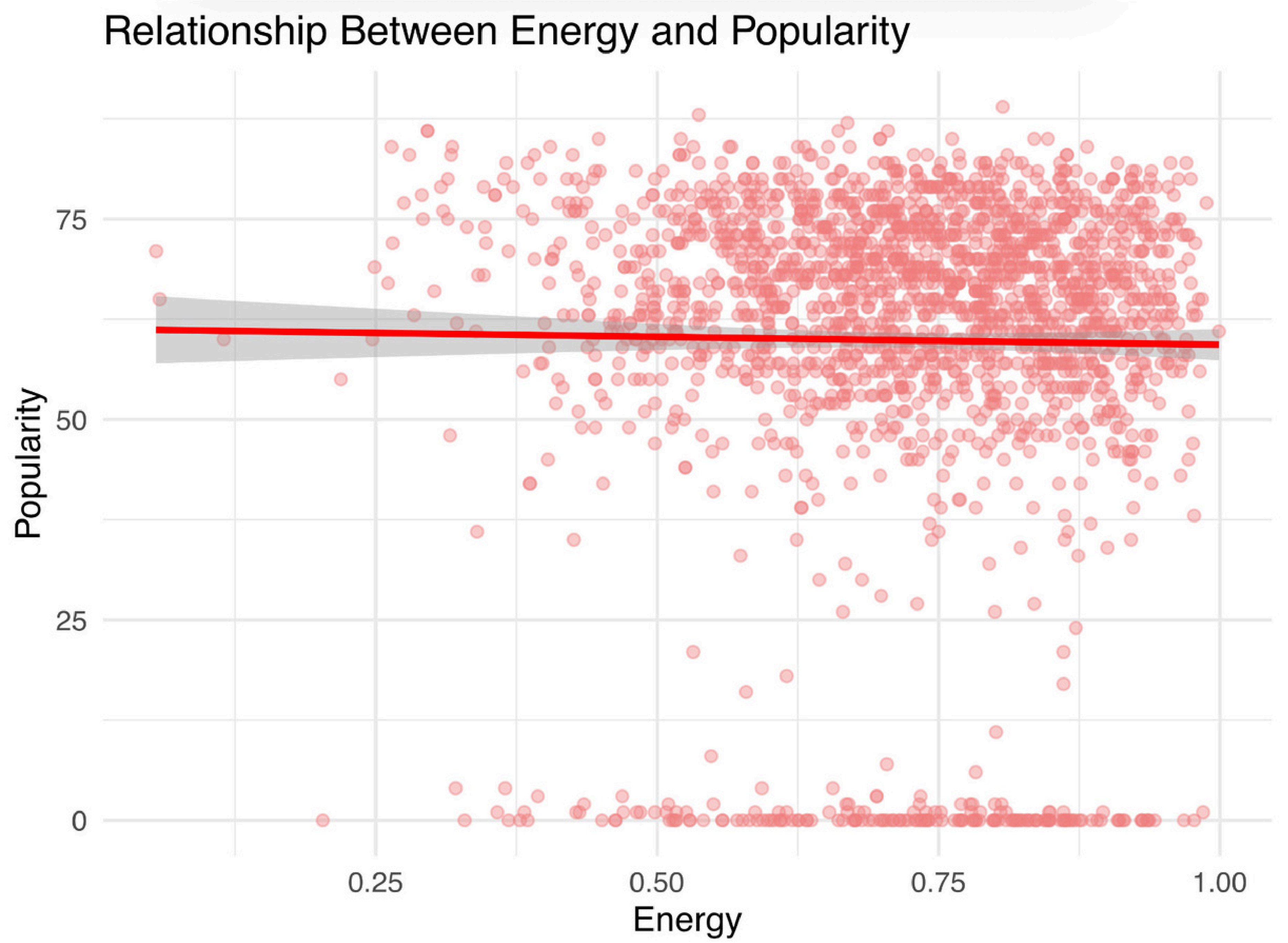

```
spotify_numeric <- spotify %>%
+     select(popularity, danceability, energy, loudness, speechiness,
+            acousticness, liveness, valence, tempo)
> corr_matrix <- cor(spotify_numeric, use = "complete.obs")
> corrplot(corr_matrix,
+          method = "color",
+          type = "upper",
+          addCoef.col = "black",
+          tl.col = "black",
+          tl.srt = 45)
```

Scatterplot Insights

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES



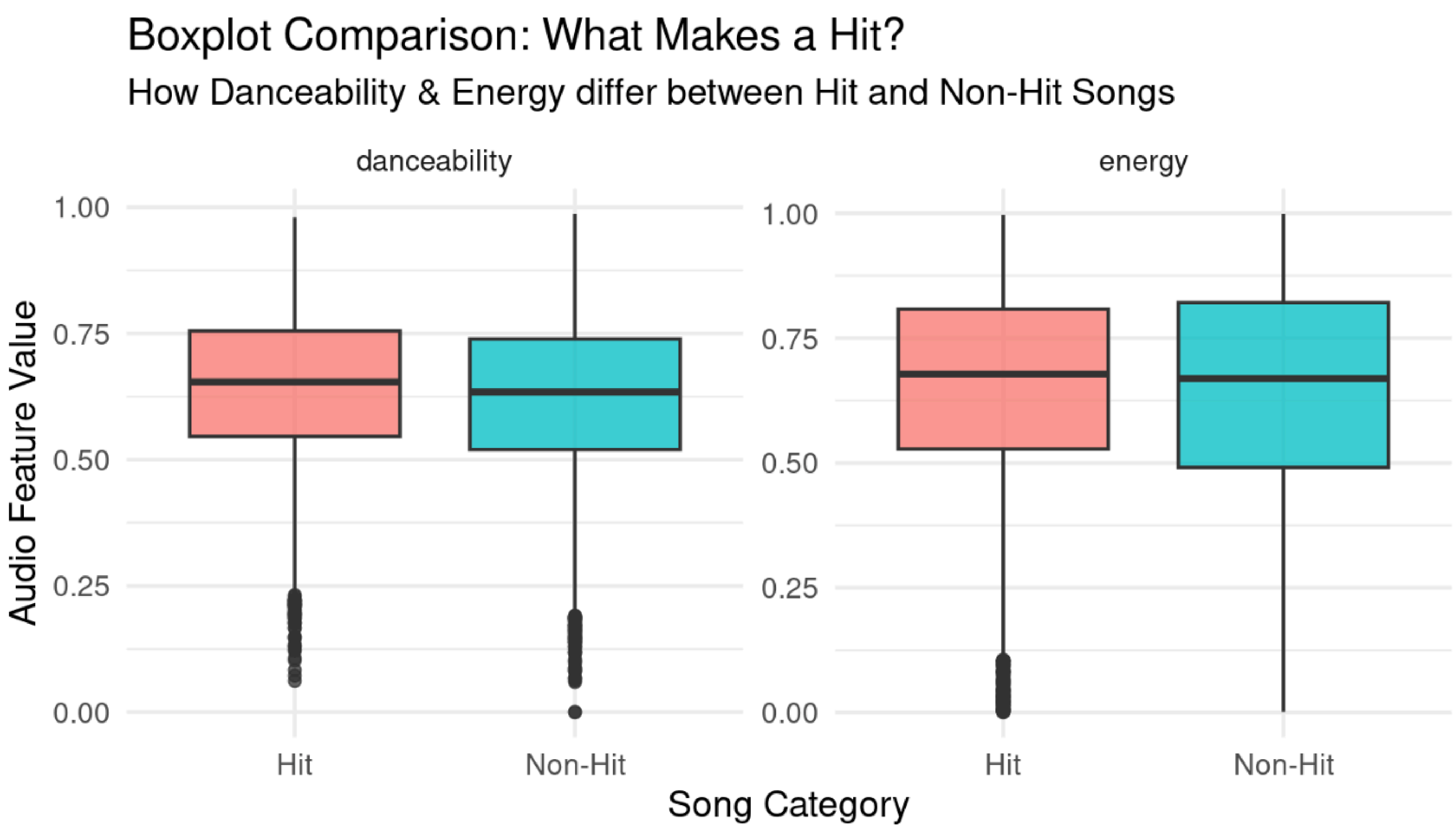

```
ggplot(spotify, aes(x = danceability, y = popularity)) +
geom_point(alpha = 0.4, color = "skyblue") +
geom_smooth(method = "lm", color = "darkblue") +
theme_minimal() +
labs(
title = "Relationship Between Danceability and Popularity",
x = "Danceability",
y = "Popularity"
)
```




```
ggplot(spotify, aes(x = energy, y = popularity)) +
+     geom_point(alpha = 0.4, color = "lightcoral") +
+     geom_smooth(method = "lm", color = "red") +
+     theme_minimal() +
+     labs(
+         title = "Relationship Between Energy and Popularity",
+         x = "Energy",
+         y = "Popularity"
+     )
```

Boxplot comparison

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES




```r
library(tidyverse)

# 1. derive Hit vs. Non-Hit from Popularity (Median-Split)
threshold <- median(song_data$song_popularity, na.rm = TRUE)

song_data <- song_data %>%
  mutate(
    hit = if_else(song_popularity >= threshold, "Hit", "Non-Hit")
  )

# 2. Data in Long-Format for two Boxplots (Danceability & Energy)
song_data_long <- song_data %>%
  pivot_longer(
    cols = c(danceability, energy),
    names_to = "feature",
    values_to = "value"
  )

# 3. draw Boxplot
ggplot(song_data_long, aes(x = hit, y = value, fill = hit)) +
  geom_boxplot(alpha = 0.75) +
  facet_wrap(~ feature, scales = "free_y") +
  labs(
    title = "Boxplot Comparison: What Makes a Hit?",
    subtitle = "How Danceability & Energy differ between Hit and Non-Hit Songs",
    x = "Song Category",
    y = "Audio Feature Value"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none")
```

- **Hits tend to be more predictable in their energy and danceability**
- **Danceability does not clearly separate Hits from Non-Hits**
- **Energy seems more important → Hits are relaibly energetic, while Non-Hits vary more widely**

Regression Model

```
Call:
lm(formula = song_popularity ~ danceability * energy, data = song_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-60.562 -12.484   2.716  15.846  46.214 

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
(Intercept)           53.666      1.693  31.708  < 2e-16 ***
danceability          -2.655      2.840  -0.935     0.35    
energy               -16.422      2.582  -6.359 2.08e-10 ***
danceability:energy   28.291      4.350   6.503 8.06e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 21.76 on 18831 degrees of freedom
Multiple R-squared:  0.0131,	Adjusted R-squared:  0.01295 
F-statistic: 83.34 on 3 and 18831 DF,  p-value: < 2.2e-16
```

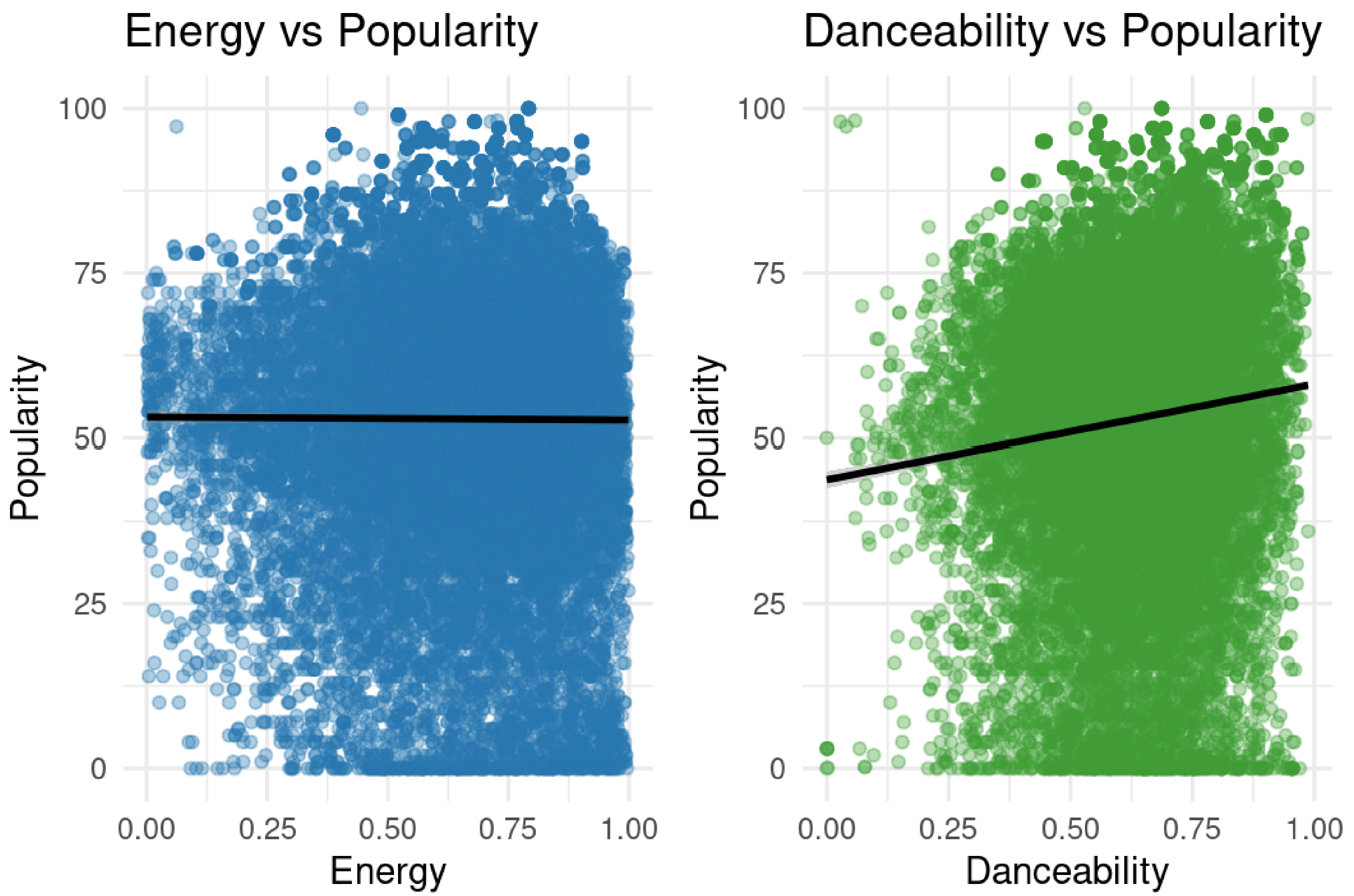
```
# Load the song data dataset
song_data <- read.csv("song_data.csv")

# overview
> str(song_data)

# Fit a linear regression model
m3 <- lm(song_popularity ~ danceability * energy, data = song_data)

# Summary of the model
summary(m3)
```

```
# create table
m3 <- lm(song_popularity ~ danceability * energy, data = song_data)
apa_lm <- apa_print(m3)
tt(apa_lm$table)
View(m3)
```

term	estimate	conf.int	statistic	df	p.value
Intercept	53.67	[50.35, 56.98]	31.71	18831	< .001
Danceability	-2.65	[-8.22, 2.91]	-0.93	18831	.350
Energy	-16.42	[-21.48, -11.36]	-6.36	18831	< .001
Danceability $\times$ Energy	28.29	[19.76, 36.82]	6.50	18831	< .001

- **Danceability alone does not predict popularity**
- **Energy alone predicts lower popularity**
- **High energy and high popularity in combination significantly increase popularity**

Popularity Trend Analysis
(commercial success)

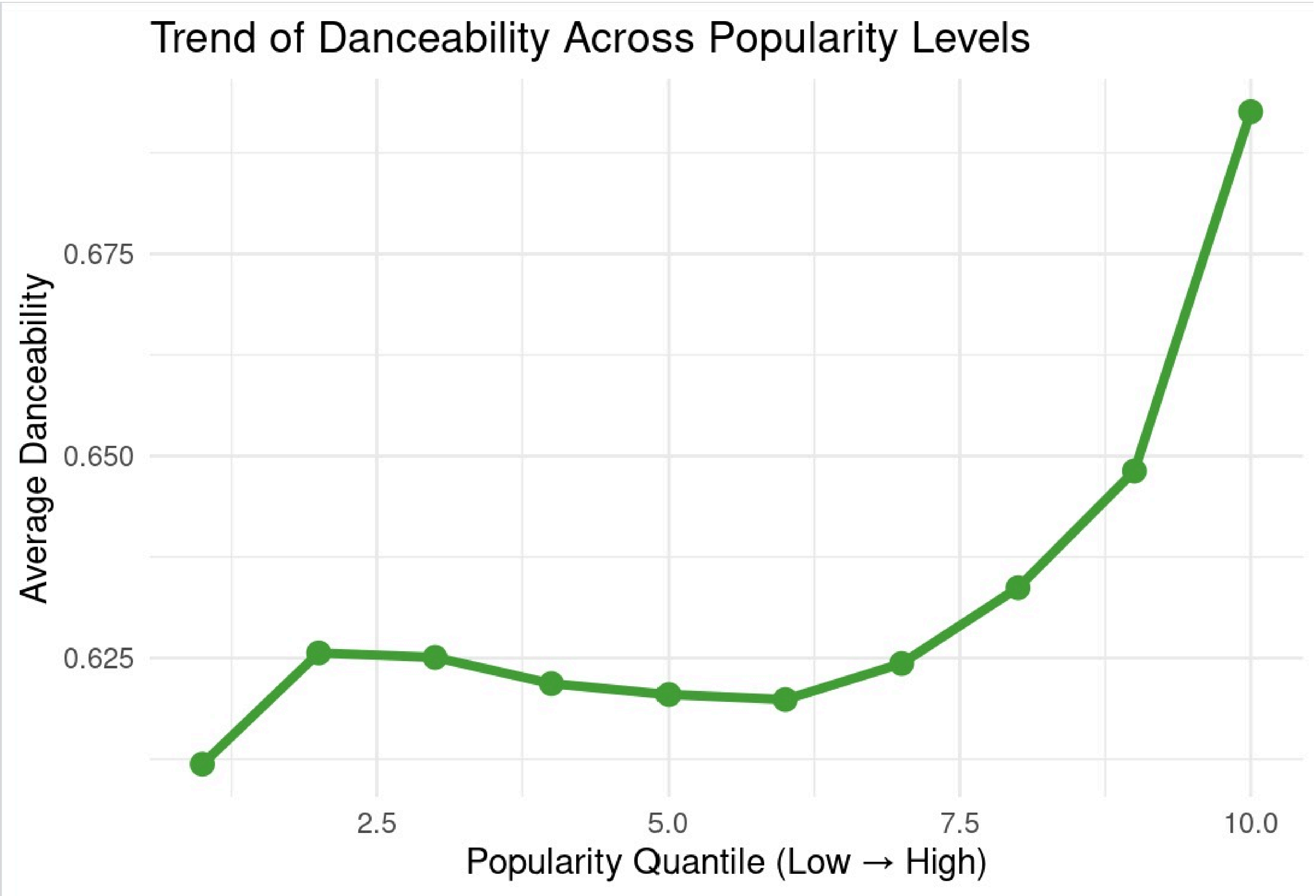


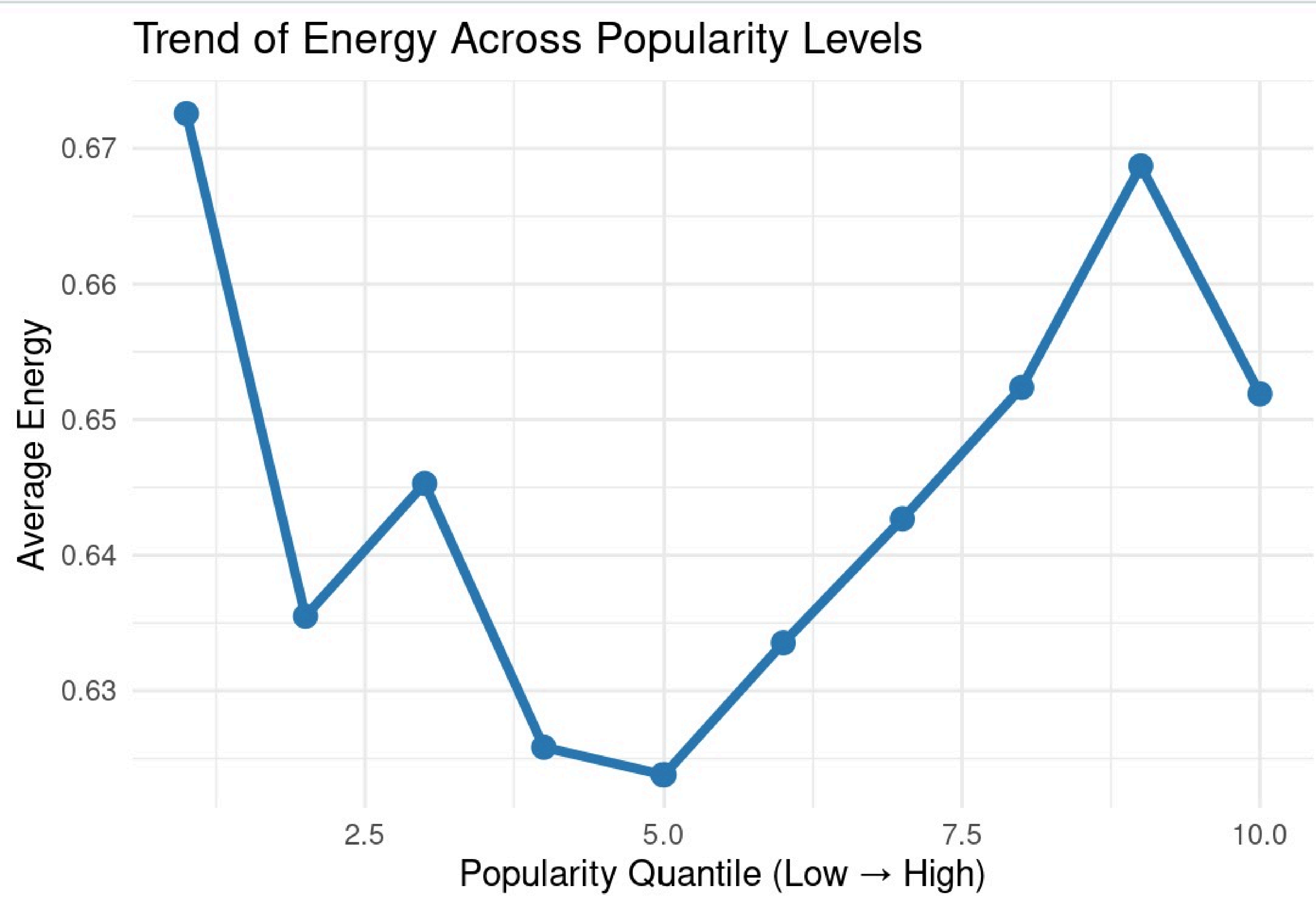
- **Danceability increases more clearly across popularity levels**
- **Energy shows a weaker, more stable pattern**
- **Popularity is sufficiently varied for prediction analysis**

Code:

```
library(tidyverse)
library(patchwork)

# Load data
spotify <- read_csv("song_data.csv")

# Clean and standardise values
spotify <- spotify %>%
  mutate(
    song_popularity = as.numeric(gsub("[^0-9.]", "", song_popularity)),
    danceability   = as.numeric(gsub("[^0-9.]", "", danceability)),
    danceability   = ifelse(danceability > 1, danceability / 100000, danceability),
    song_popularity = ifelse(song_popularity > 100,
                             song_popularity / 20,
                             song_popularity)
  )

# Remove invalid rows
spotify_clean <- spotify %>%
  filter(
    !is.na(song_popularity),
    !is.na(energy),
    !is.na(danceability),
    song_popularity >= 0 & song_popularity <= 100,
    danceability >= 0 & danceability <= 1
  )

# Energy scatterplot
p_energy <- ggplot(spotify_clean, aes(energy, song_popularity)) +
  geom_point(color = "#1f78b4", alpha = 0.35, size = 1.4) +
  geom_smooth(method = "lm", color = "black", se = TRUE) +
  theme_minimal() +
  labs(title = "Energy vs Popularity",
       x = "Energy", y = "Popularity")

# Danceability scatterplot
p_dance <- ggplot(spotify_clean, aes(danceability, song_popularity)) +
  geom_point(color = "#33a02c", alpha = 0.35, size = 1.4) +
  geom_smooth(method = "lm", color = "black", se = TRUE) +
  theme_minimal() +
  labs(title = "Danceability vs Popularity",
       x = "Danceability", y = "Popularity")

# Combine the two plots
p_energy + p_dance
```

How Audio Features Change With Increasing Popularity





Code:

```r
# Create popularity quantiles and group means
spotify_quantile <- spotify_clean %>%
  mutate(pop_quantile = ntile(song_popularity, 10)) %>%
  group_by(pop_quantile) %>%
  summarise(
    avg_dance  = mean(danceability, na.rm = TRUE),
    avg_energy = mean(energy,       na.rm = TRUE)
  )

# Danceability trend
ggplot(spotify_quantile, aes(pop_quantile, avg_dance)) +
  geom_line(color = "#33a02c", linewidth = 1.4) +
  geom_point(color = "#33a02c", size = 3) +
  theme_minimal() +
  labs(
    title = "Trend of Danceability Across Popularity Levels",
    x = "Popularity Quantile (Low → High)",
    y = "Average Danceability"
  )

# Energy trend
ggplot(spotify_quantile, aes(pop_quantile, avg_energy)) +
  geom_line(color = "#1f78b4", linewidth = 1.4) +
  geom_point(color = "#1f78b4", size = 3) +
  theme_minimal() +
  labs(
    title = "Trend of Energy Across Popularity Levels",
    x = "Popularity Quantile (Low → High)",
    y = "Average Energy"
  )
```

First conclusion

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES

- songs have enough variation in features for analysis
- Energy & danceability show positive connection with popularity
- Scatterplots confirm - trend exists, though not very strong
- Boxplots show: more popular songs - usually higher danceability & energy
- Regression: features matter; only explain small part → why a song becomes successful
- Trend analysis supports this: danceability increases as popularity increases, while energy stays stable

Limitations

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES

API Constraints

- Spotify API expires every 60 minutes → unstable for large data
- Required track-by-track calls → unrealistic for thousands of songs

Data Quality & Consistency Issues

- Kaggle CSV exports contained inconsistent fields
- Missing values had to be cleaned manually

No Genre or Year Controls Yet

- Genre strongly influences danceability + energy
- Release year affects popularity (e.g., old songs rank lower)

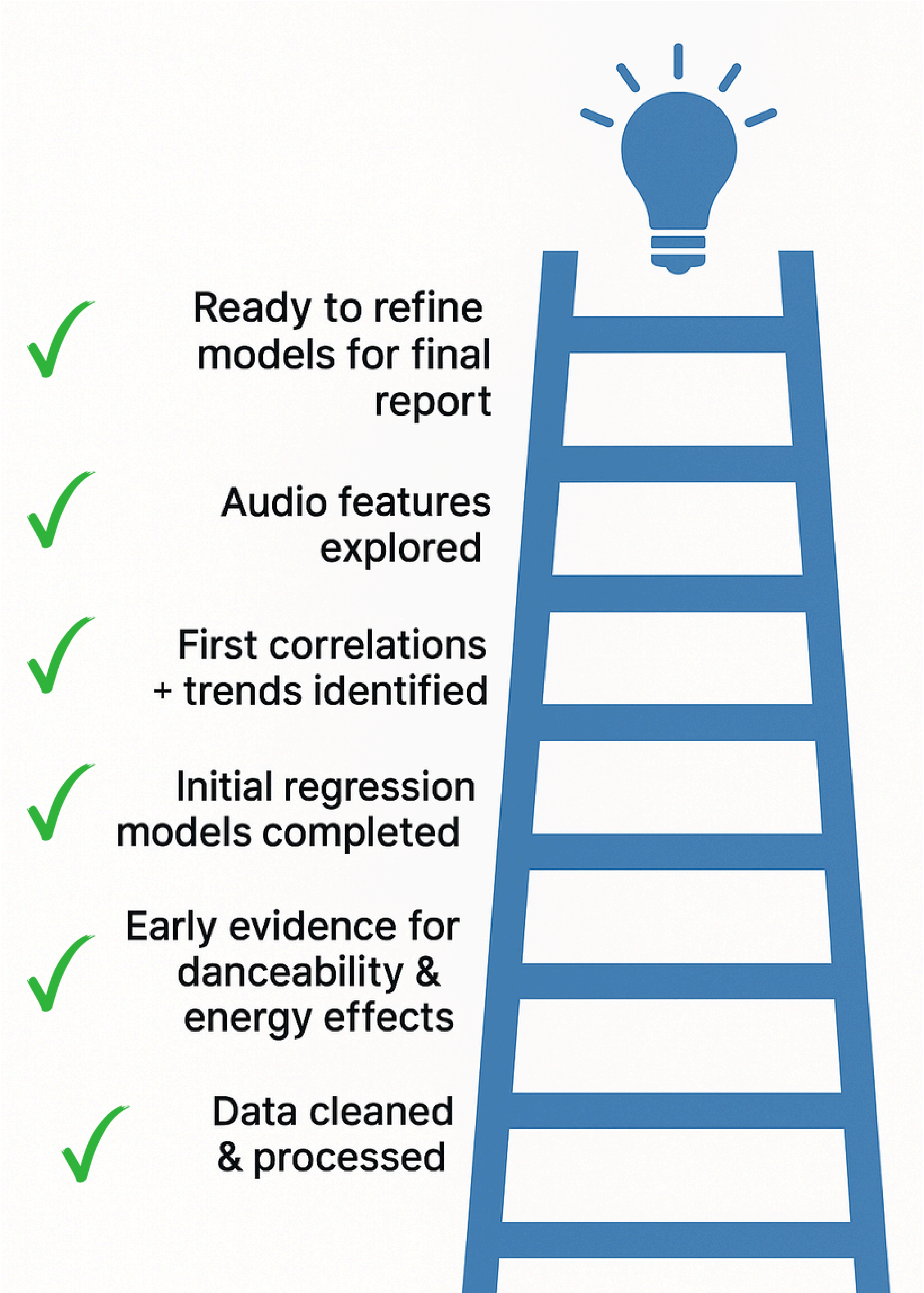
Limited Predictive Power of Current Model

- R^2 is very low → audio features explain only a small part of popularity
- Success is influenced by external factors (marketing, playlists, virality)

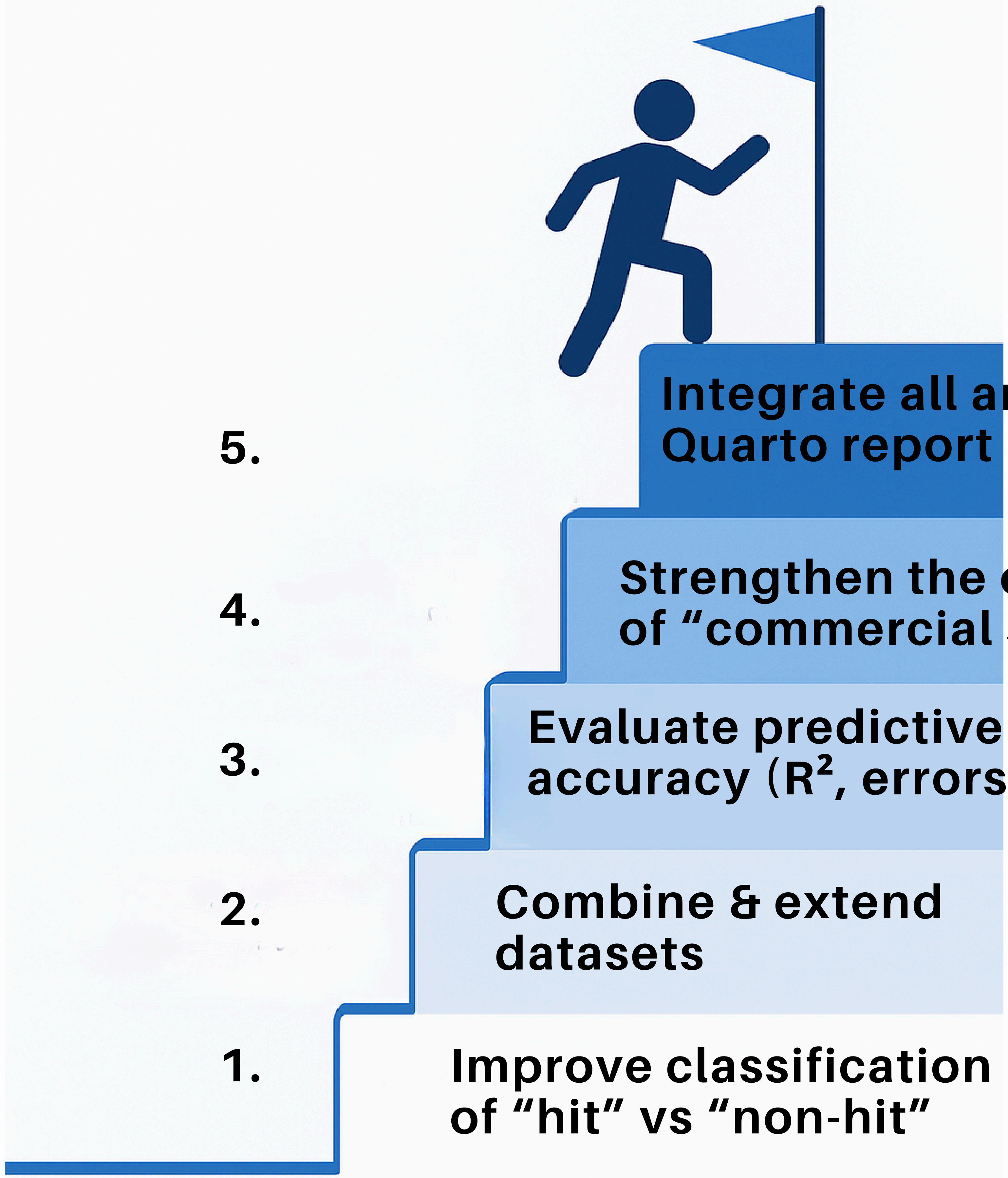
Potential Sampling Bias

- The dataset may overrepresent popular, which could distort the relationship between audio features and commercial success.
- Missing or uneven genre distribution (e.g., Pop dominating the dataset) may bias the predictive patterns for danceability and energy

Where are we now?



Next steps until final report



1. Improve classification of “hit” vs “non-hit”
2. Combine & extend datasets
3. Evaluate predictive accuracy (R^2, errors)
4. Strengthen the operational definition of “commercial success”
5. Integrate all analyses into the final Quarto report

Thanks for your attention!!

HOCHSCHULE FRESENIUS
UNIVERSITY OF APPLIED SCIENCES

Q & A

Sources/data utilized

- https://www.kaggle.com/datasets/edalrami/19000-spotify-songs?resource=download
- https://tidyverse.org
- https://otexts.com
- https://www.kaggle.com/code/varunsaikanuri/spotif-data-visualization