

R Code for visualizations -

Average Tempo Over Time by Playlist Genre

```
library(tidyverse)

# Load the Spotify dataset
spotify <- read_csv(
  "https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2020/2020-01-21/spotify_songs.csv"
)

# Create a year variable and keep the 6 main genres
tempo_year <- spotify %>%
  mutate(year = as.integer(substr(track_album_release_date, 1, 4))) %>%
  filter(!is.na(year),
    playlist_genre %in% c("edm", "latin", "pop", "r&b", "rap", "rock"),
    year >= 1990) %>%
  group_by(year, playlist_genre) %>%
  summarise(avg_tempo = mean(tempo, na.rm = TRUE), .groups = "drop")

# Line plot
ggplot(tempo_year, aes(x = year, y = avg_tempo, color = playlist_genre)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Average Tempo Over Time by Playlist Genre",
    x = "Year",
    y = "Average Tempo (BPM)",
    color = "Genre"
  ) +
  theme_minimal()
```

Correlation Between Audio Features and Popularity

```
library(tidyverse)

# Select features to compare with popularity
features <- c("danceability", "energy", "loudness", "speechiness",
  "acousticness", "instrumentalness", "liveness",
  "valence", "tempo")

# Compute correlations with popularity
corr_data <- spotify %>%
```

```
select(track_popularity, all_of(features)) %>%
summarise(across(all_of(features),
~ cor(.x, track_popularity, use = "complete.obs"))) %>%
pivot_longer(cols = everything(),
names_to = "feature",
values_to = "correlation")

# Bar chart of correlations
ggplot(corr_data, aes(x = reorder(feature, correlation),
y = correlation,
fill = correlation)) +
geom_col() +
coord_flip() +
labs(
  title = "Correlation of Audio Features with Track Popularity",
  x = "Audio Feature",
  y = "Correlation"
) +
scale_fill_gradient2(low = "steelblue", high = "darkred", midpoint = 0) +
theme_minimal() +
theme(legend.position = "none")
```