Computational Musicology: Russian Classical Music

Anthony Thieme

2024-02-15

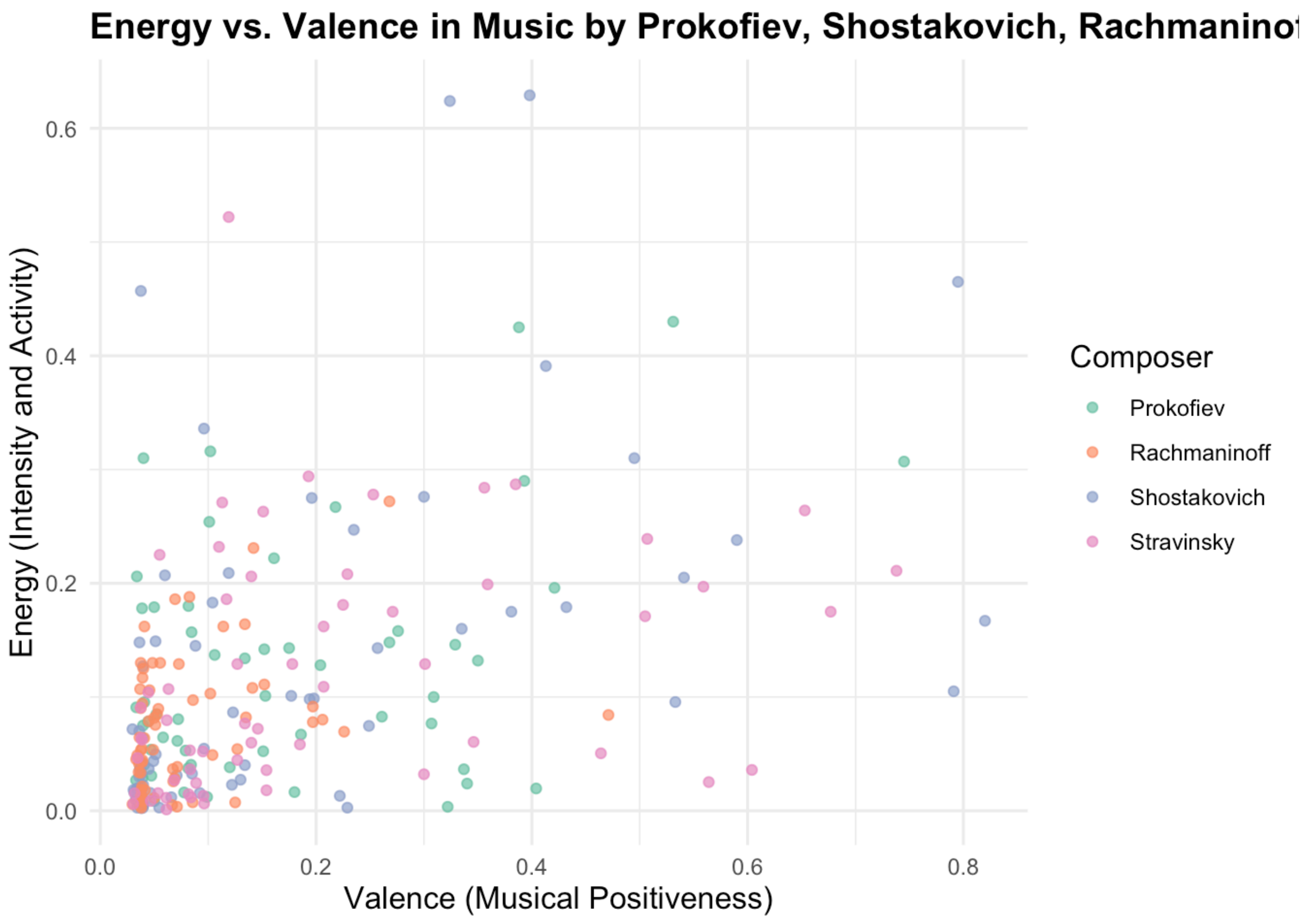
introduction

The corpus is centered on the distinct, robust, and often intense style of Russian composers like Prokofiev, Shostakovich, and Stravinsky. This selection is driven by a fascination with how these composers' works reflect the tumultuous history of Russia, embodying a range of emotions from despair to triumph, often with a raw, powerful intensity. Their music, marked by innovative orchestration and harmonies, serves as a window into the Russian soul, culture, and the political environment that shaped their creative output.

```r
prokofiev <- get_playlist_audio_features("","37i9dQZF1DX1tT8vLysk8r")
shostakovic <- get_playlist_audio_features("","37i9dQZF1DXbJgoX0EGLWU")
rachmaninoff <- get_playlist_audio_features("","37i9dQZF1DX5Flpl98I3He")
stravinsky <- get_playlist_audio_features("","37i9dQZF1DWUXrxTKKqFir")

# Combine all tracks into one dataframe and add a composer column
tracks_combined <- bind_rows(
  prokofiev %>% mutate(composer = "Prokofiev"),
  shostakovic %>% mutate(composer = "Shostakovich"),
  rachmaninoff %>% mutate(composer = "Rachmaninoff"),
  stravinsky %>% mutate(composer = "Stravinsky")
)

# Prepare the data by selecting relevant features and renaming appropriately
tracks_prepared <- tracks_combined %>%
  select(track.id, track.name, track.artists,composer, energy, valence, tempo, ins
trumentalness, danceability, acousticness, liveness, loudness, speechiness, time_s
ignature, mode, key)
```

```
# Create the plot
plot1 <- ggplot(tracks_prepared, aes(x = valence, y = energy, color = composer)) +
  geom_point(alpha = 0.7) +  # Use semi-transparent points to handle overplotting
  theme_minimal() +  # Clean and minimal theme
  labs(
    title = "Energy vs. Valence in Music by Prokofiev, Shostakovich, Rachmaninoff,
and Stravinsky",
    x = "Valence (Musical Positiveness)",
    y = "Energy (Intensity and Activity)",
    color = "Composer"
  ) +
  scale_color_brewer(palette = "Set2") +  # Color palette for better distinction

  theme(
    plot.title = element_text(face = "bold", size = 14),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.position = "right"
  )
```

```
plot(plot1)
```

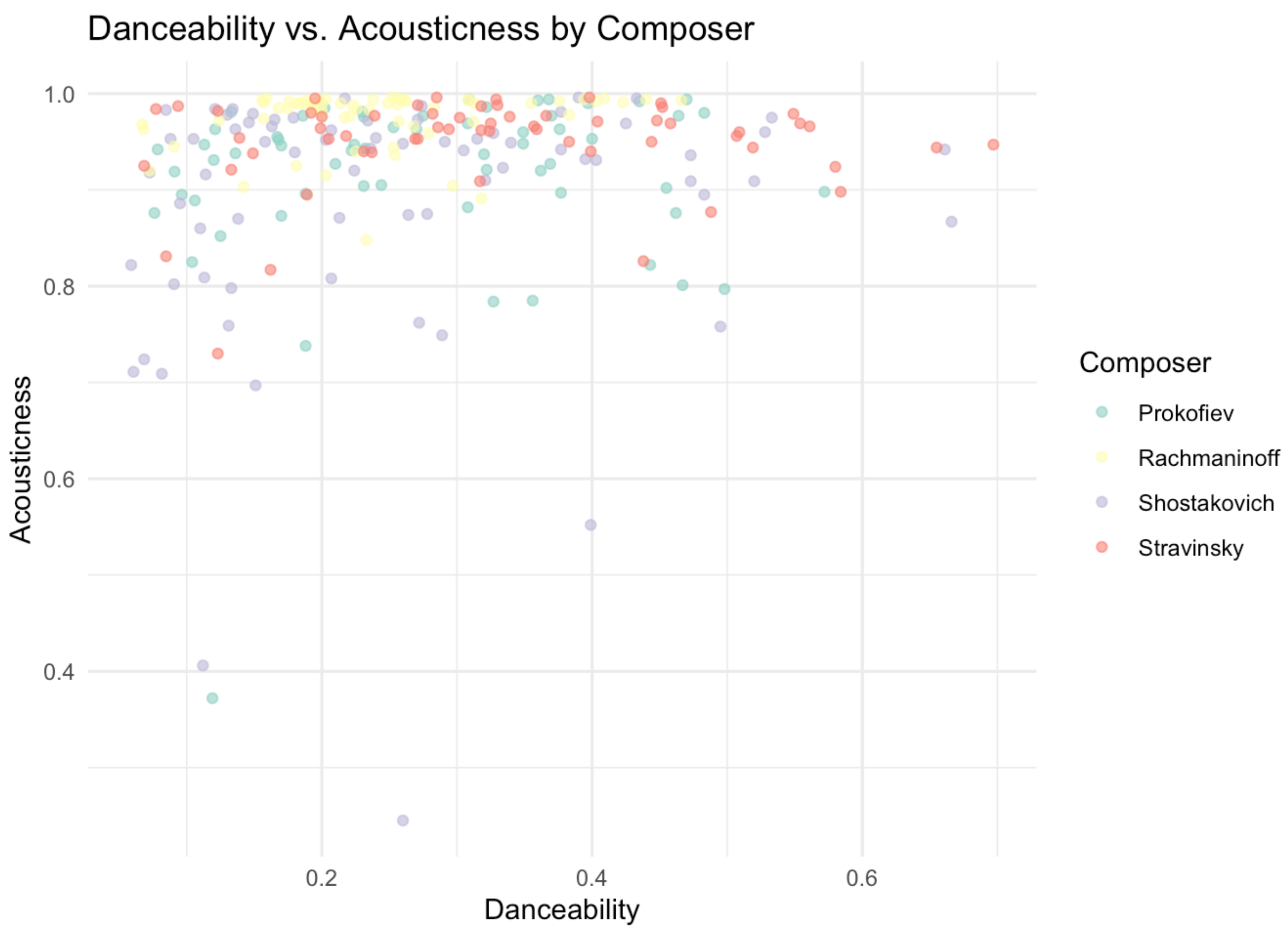


##**Energy and Valence**: This comparison delves into the interplay between the intensity and the emotional positivity of compositions, offering a multifaceted view of a composer's musical language. **Energy** reflects the physical power and dynamism of the music—its ability to excite, stir, and energize the listener. Highenergy pieces often employ faster tempos, louder volumes, and a more pronounced rhythmic drive, embodying the vigor and force of the musical expression. **Valence**, on the other hand, captures the emotional spectrum that music conveys, ranging from cheerful and euphoric to somber and melancholic. By examining energy and valence together, we uncover the emotional depth and the kinetic force of the music, revealing how composers orchestrate these elements to craft experiences that can be uplifting, contemplative, or anything in between. This juxtaposition not only highlights the emotional intent behind compositions but also showcases how composers modulate energy to align with the emotional landscapes they wish to evoke, thus providing insights into their creative nuances and the affective power of their work.

Danceability and Acousticness:

To explore how danceable and acoustic their compositions are, which might give insights into their style's robustness and intensity.

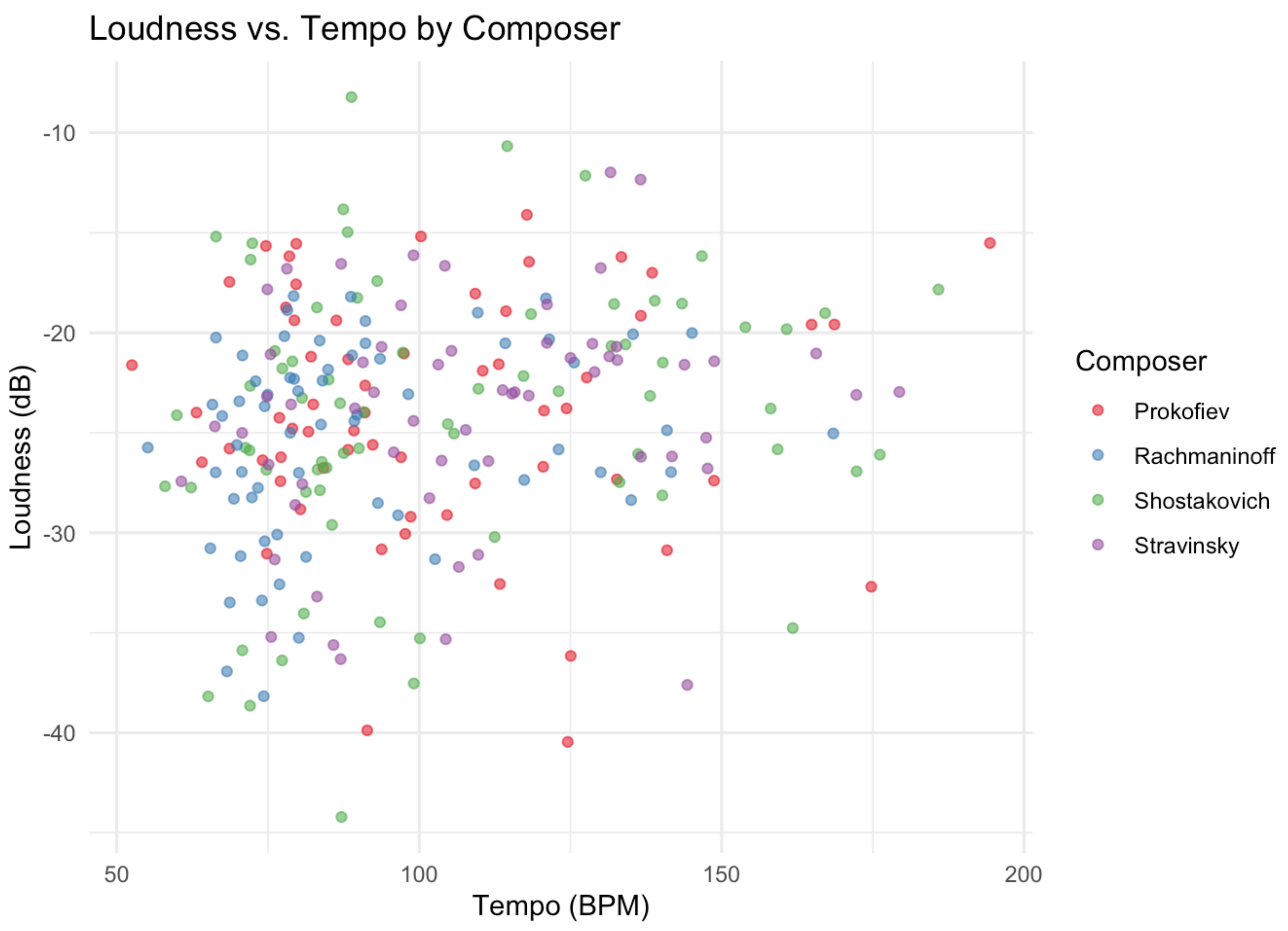
```
ggplot(tracks_prepared, aes(x = danceability, y = acousticness, color = composer))
+
  geom_point(alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Danceability vs. Acousticness by Composer",
    x = "Danceability",
    y = "Acousticness",
    color = "Composer"
  ) +
  scale_color_brewer(palette = "Set3") +
  theme(legend.position = "right")
```



Loudness and Tempo:

To understand the dynamic range and pace, which can contribute to the perceived intensity of their music.

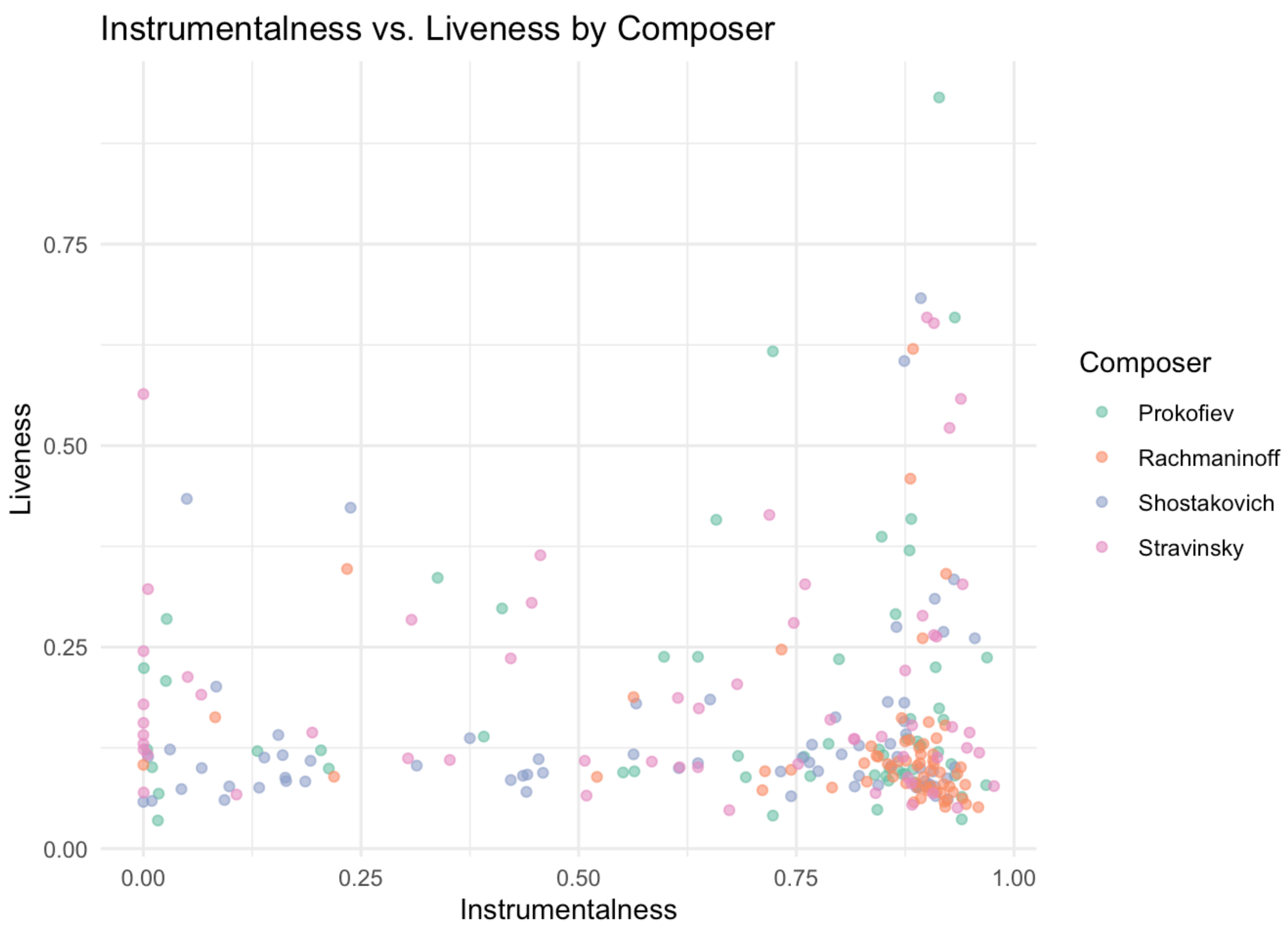
```
ggplot(tracks_prepared, aes(x = tempo, y = loudness, color = composer)) +
  geom_point(alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Loudness vs. Tempo by Composer",
    x = "Tempo (BPM)",
    y = "Loudness (dB)",
    color = "Composer"
  ) +
  scale_color_brewer(palette = "Set1") +
  theme(legend.position = "right")
```



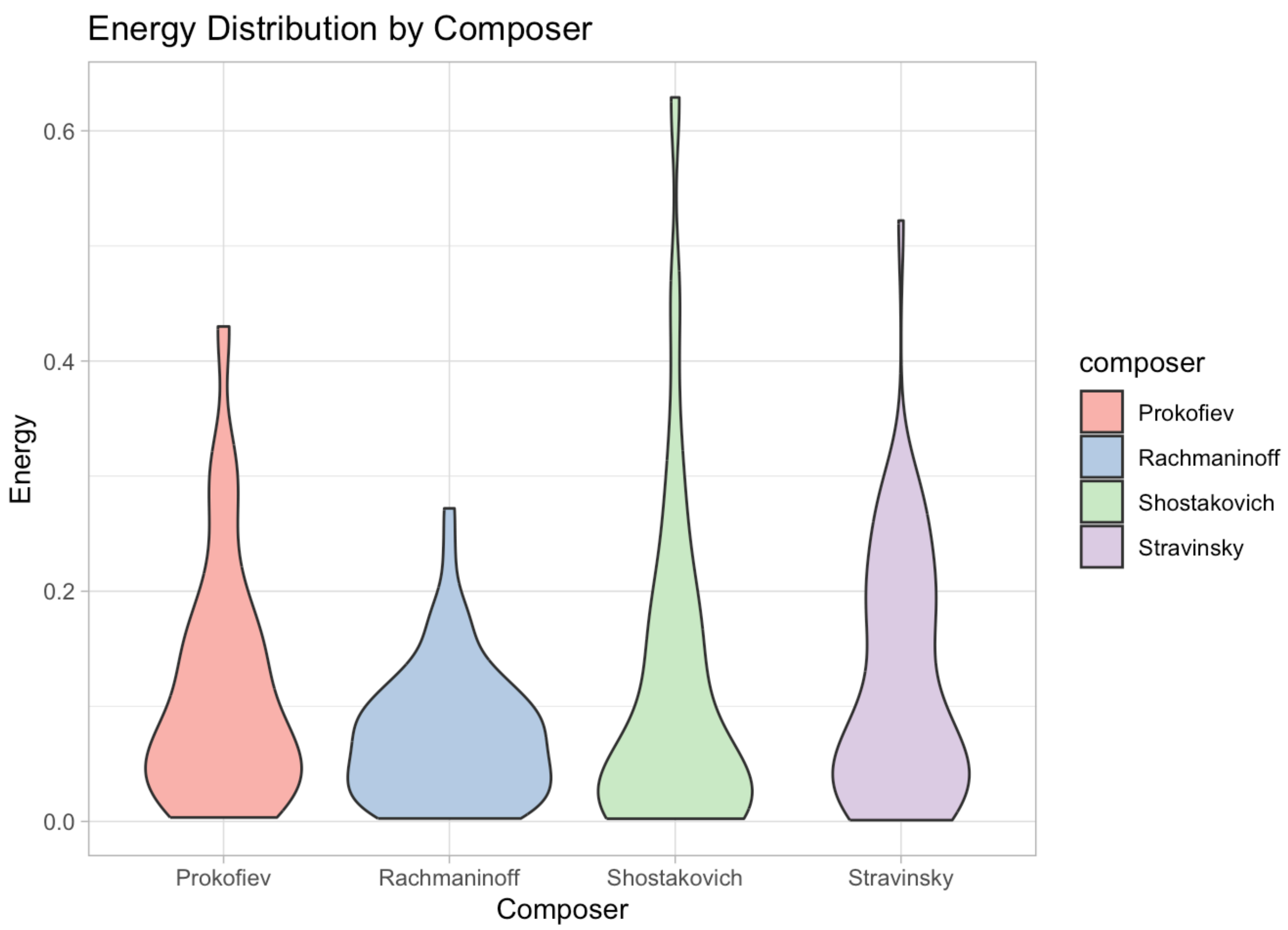
strumentalness and Liveness:

To gauge the focus on instrumental music versus live recordings, reflecting on the composers’ preferences for studio precision or live performance energy.

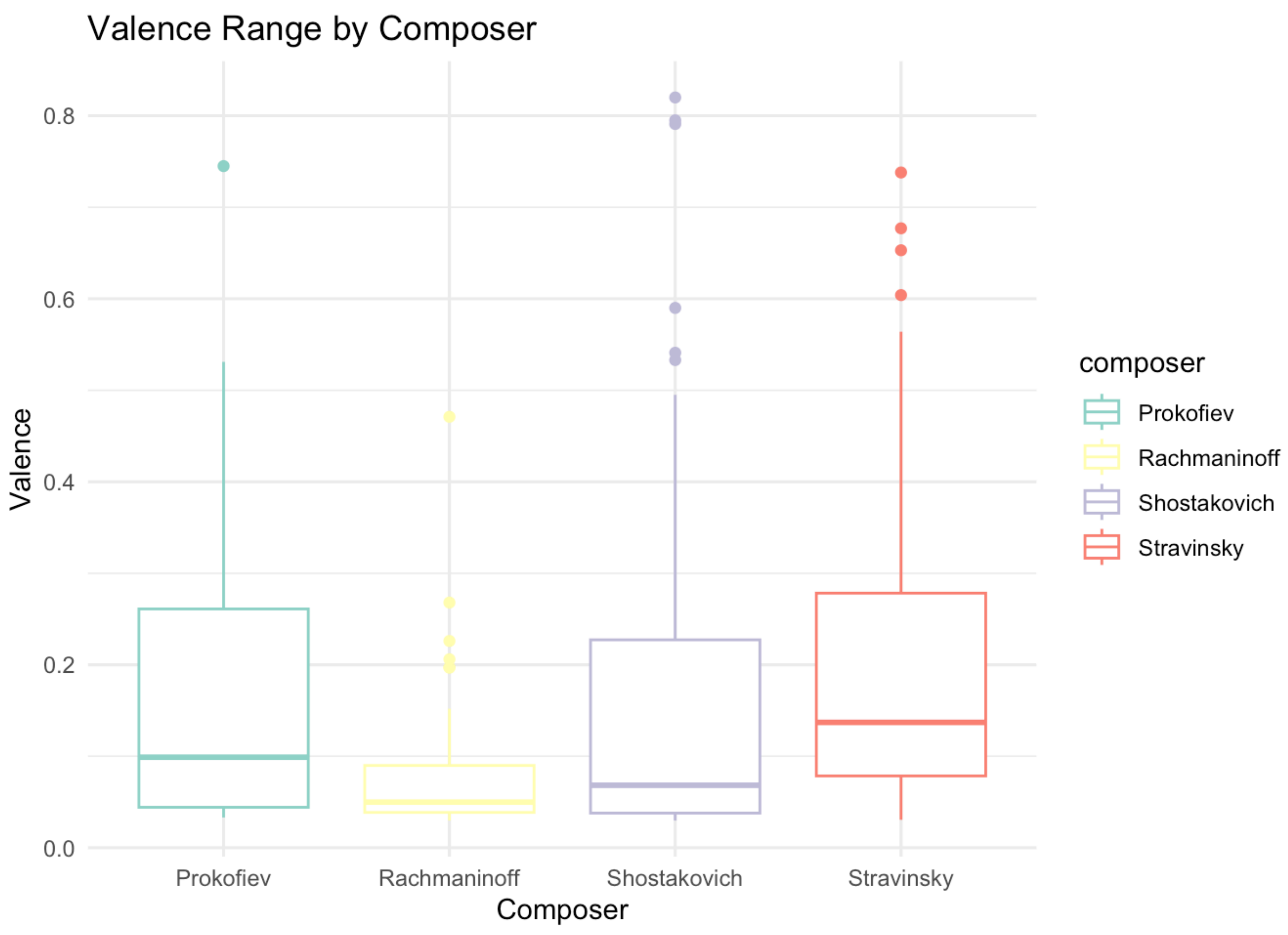
```
ggplot(tracks_prepared, aes(x = instrumentalness, y = liveness, color = composer))
+
  geom_point(alpha = 0.6) +
  theme_minimal() +
  labs(
    title = "Instrumentalness vs. Liveness by Composer",
    x = "Instrumentalness",
    y = "Liveness",
    color = "Composer"
  ) +
  scale_color_brewer(palette = "Set2") +
  theme(legend.position = "right")
```



```
ggplot(tracks_prepared, aes(x = composer, y = energy, fill = composer)) +
  geom_violin() +
  labs(title = "Energy Distribution by Composer", x = "Composer", y = "Energy") +
  theme_light() +
  scale_fill_brewer(palette = "Pastel1")
```

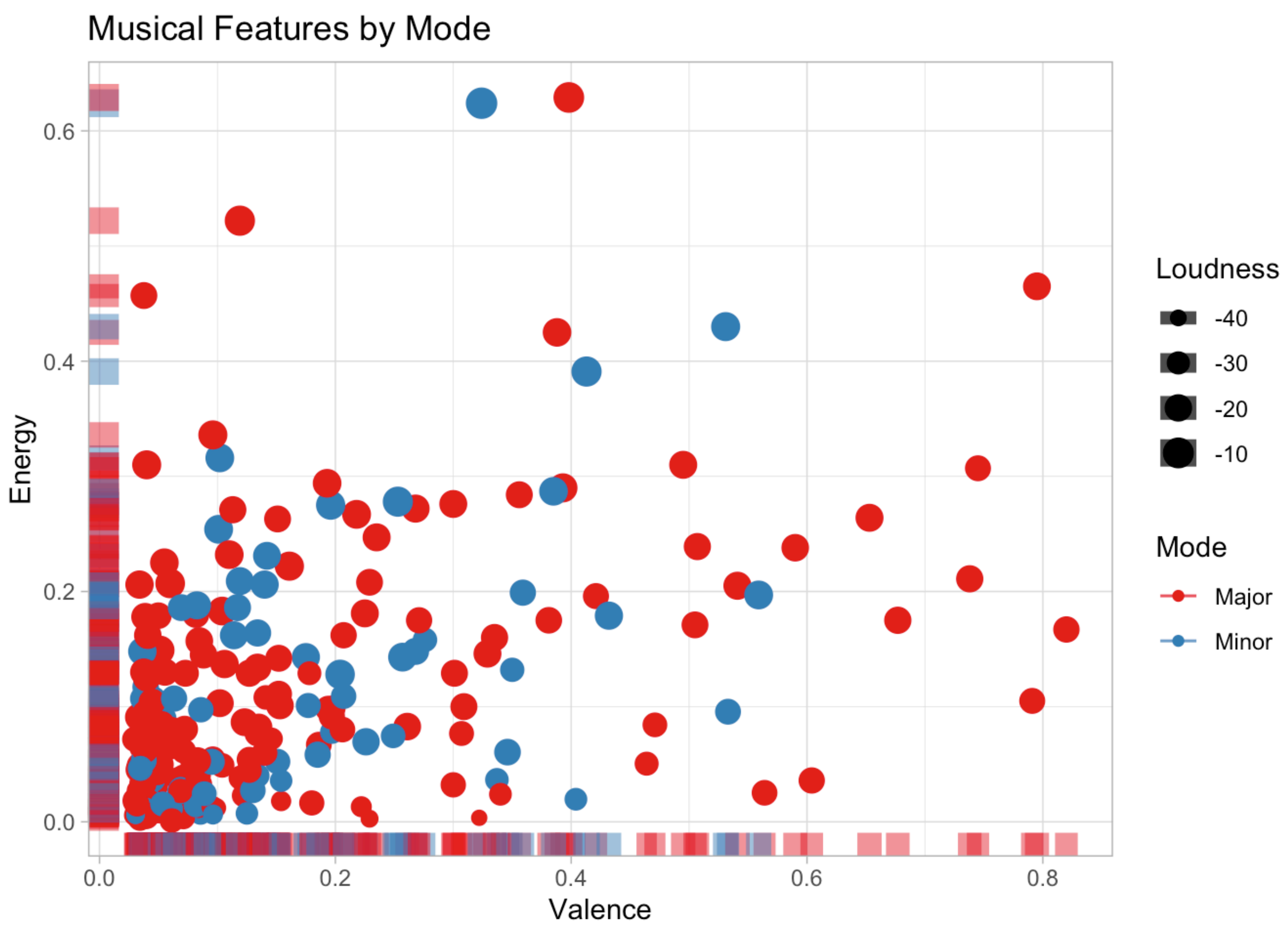


```
ggplot(tracks_prepared, aes(x = composer, y = valence, color = composer)) +
  geom_boxplot() +
  labs(title = "Valence Range by Composer", x = "Composer", y = "Valence") +
  theme_minimal() +
  scale_color_brewer(palette = "Set3")
```

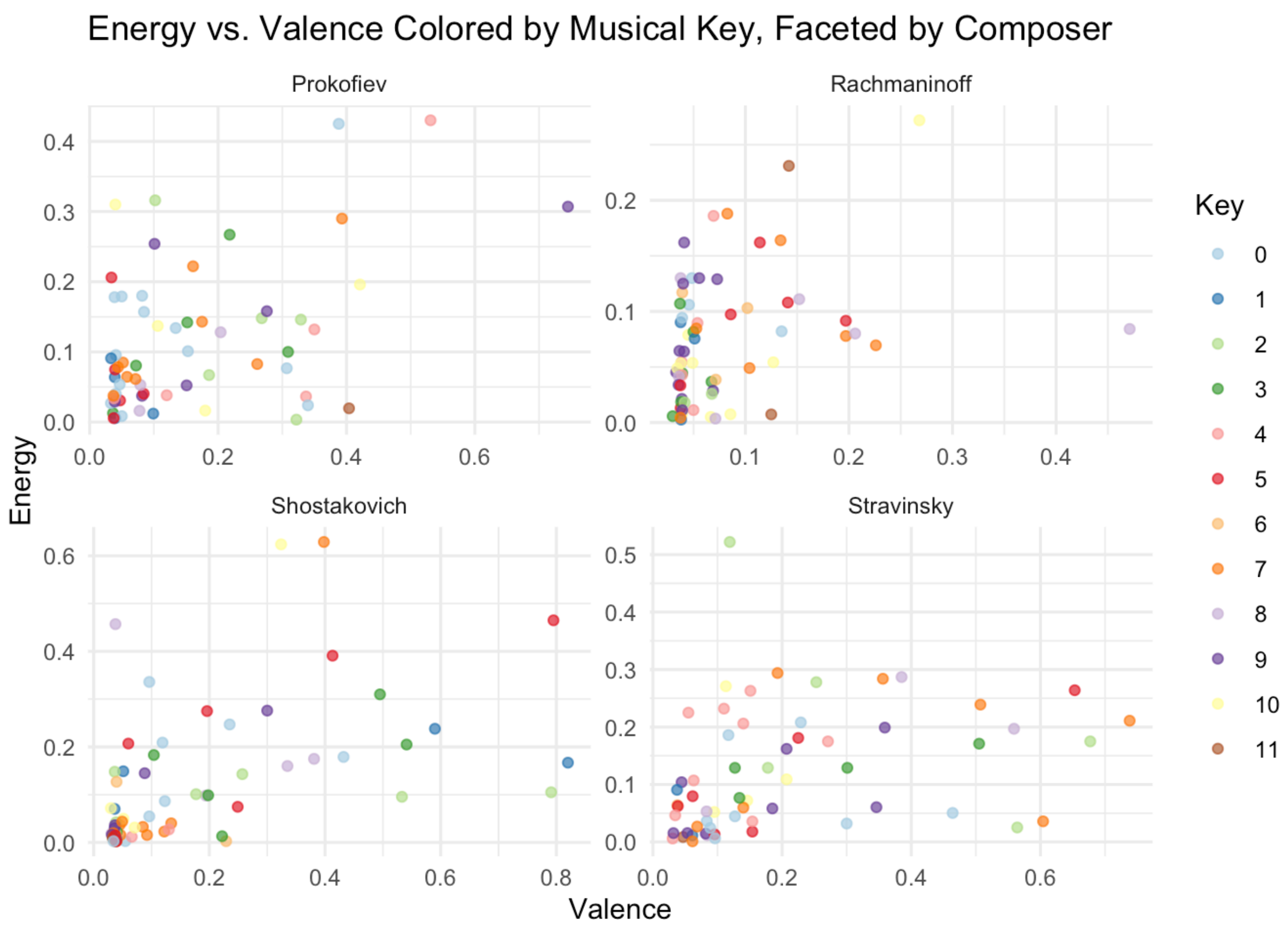


```r
tracks_prepared <- tracks_prepared %>%
  mutate(mode = ifelse(mode == 0, "Minor", "Major")) # Example mutation

ggplot(tracks_prepared, aes(x = valence, y = energy, size = loudness, color = mode
)) +
  geom_point() +
  geom_rug(sides = "b", alpha = 0.5) + # Bottom side rugs for valence
  geom_rug(sides = "l", alpha = 0.5) + # Left side rugs for energy
  theme_light() +
  labs(title = "Musical Features by Mode", x = "Valence", y = "Energy", color = "M
ode", size = "Loudness") +
  scale_color_brewer(palette = "Set1") +
  scale_size(range = c(1, 5)) # Adjusting point size scale for visibility
```

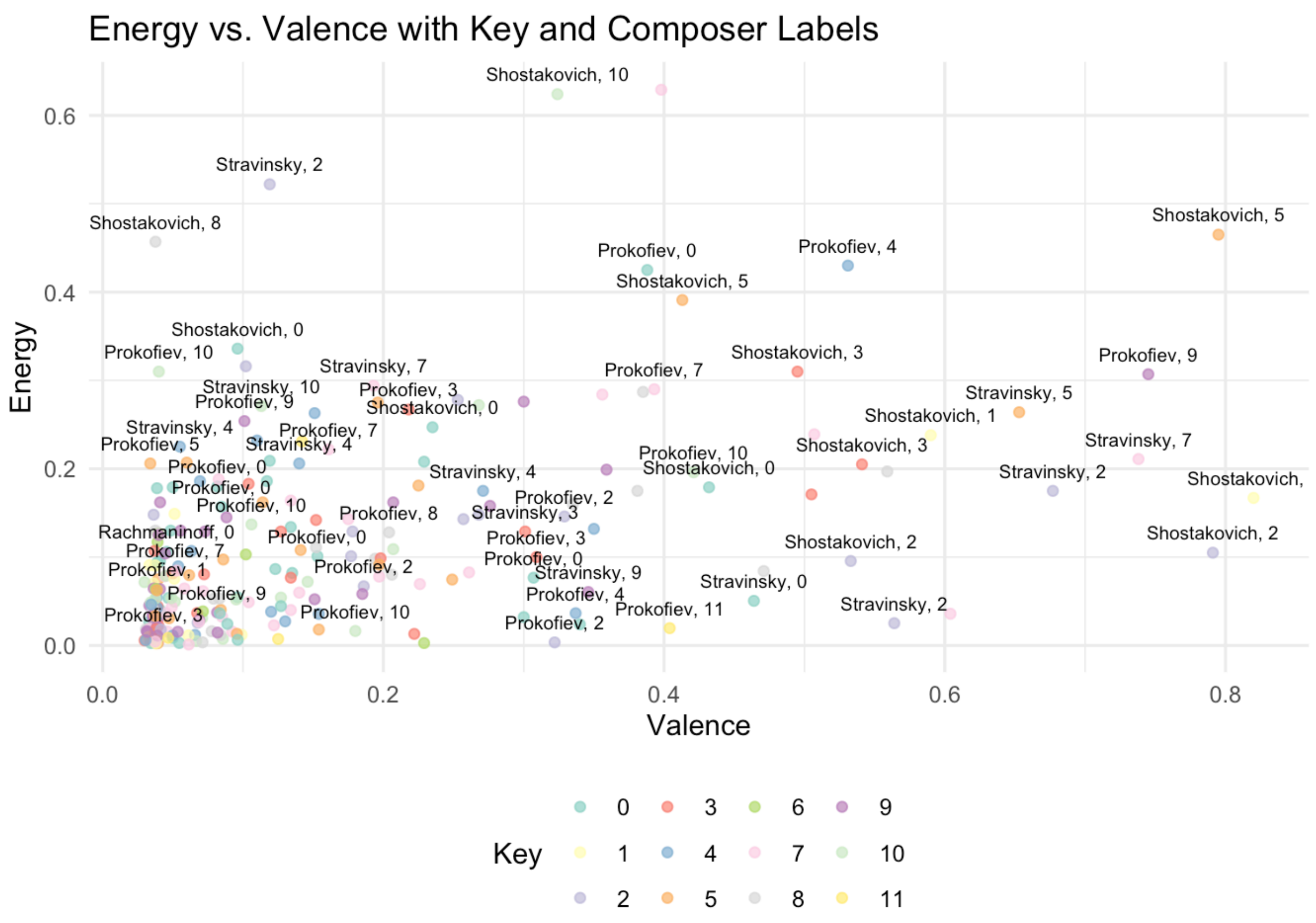
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
ggplot(tracks_prepared, aes(x = valence, y = energy, color = as.factor(key))) +
  geom_point(alpha = 0.7) +
  facet_wrap(~composer, scales = "free", ncol = 2) +  # Adjust `ncol` for layout p
references
  theme_minimal() +
  labs(
    title = "Energy vs. Valence Colored by Musical Key, Faceted by Composer",
    x = "Valence",
    y = "Energy",
    color = "Key"
  ) +
  scale_color_brewer(palette = "Paired", type = "qual", direction = 1) +
  theme(legend.position = "right")
```

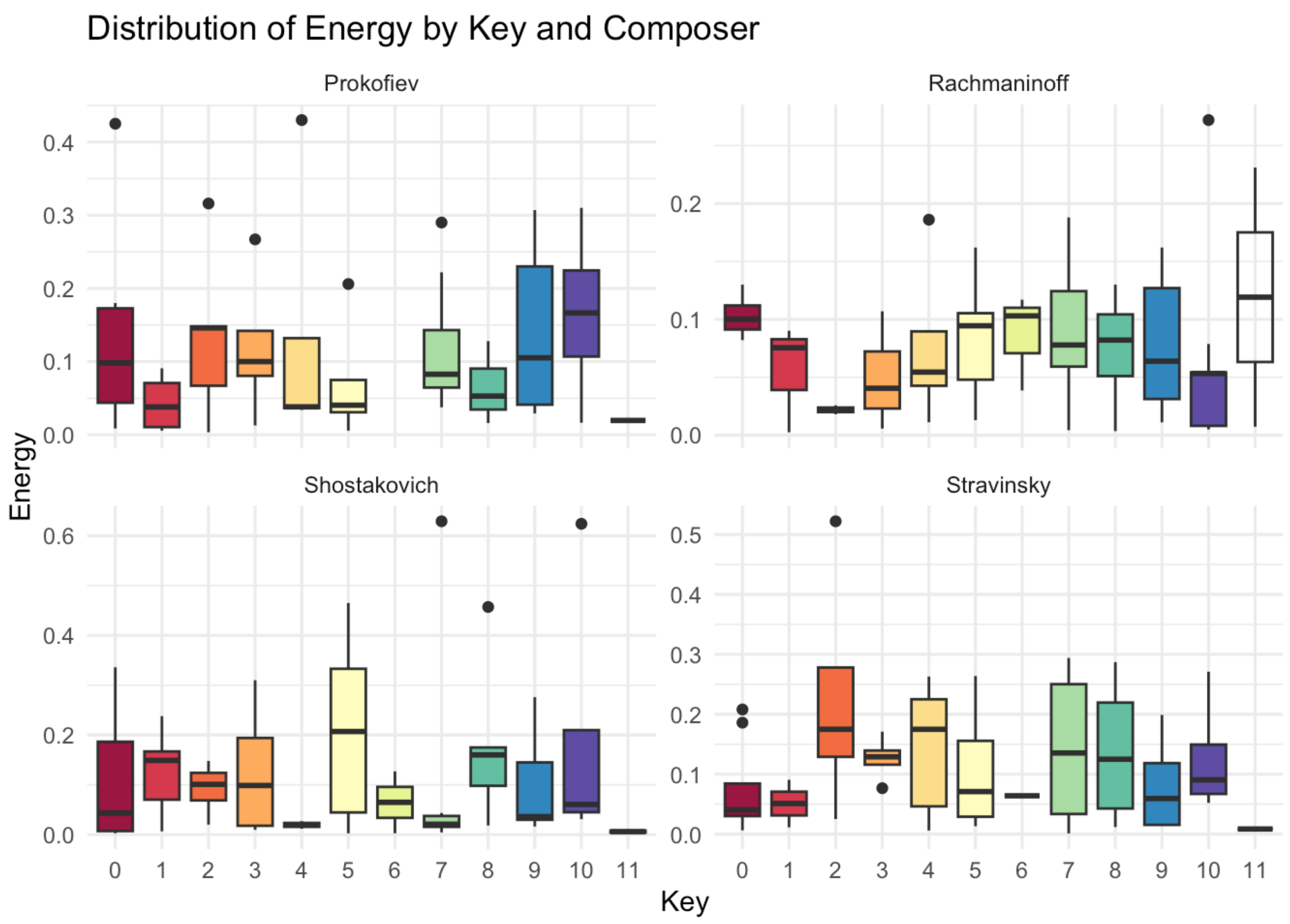


```
ggplot(tracks_prepared, aes(x = valence, y = energy)) +
  geom_point(aes(color = as.factor(key)), alpha = 0.7) +
  geom_text(aes(label = paste(composer, key, sep = ", ")), check_overlap = TRUE, v
just = -1, size = 2.5) +
  theme_minimal() +
  labs(
    title = "Energy vs. Valence with Key and Composer Labels",
    x = "Valence",
    y = "Energy",
    color = "Key"
  ) +
  scale_color_brewer(palette = "Set3") +
  theme(legend.position = "bottom")
```



```
ggplot(tracks_prepared, aes(x = as.factor(key), y = energy, fill = as.factor(key))
) +
  geom_boxplot() +
  facet_wrap(~composer, scales = "free_y") +
  theme_minimal() +
  labs(
    title = "Distribution of Energy by Key and Composer",
    x = "Key",
    y = "Energy",
    fill = "Key"
  ) +
  scale_fill_brewer(palette = "Spectral") +
  theme(legend.position = "none")  # Hide legend if redundant
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for p
alette Spectral is 11
## Returning the palette you asked for with that many colors
```


Distribution of Energy by Key and Composer
Prokofiev
Rachmaninoff
Shostakovich
Stravinsky
Energy
Key
0.0
0.1
0.2
0.3
0.4
0.6
0.5
0 1 2 3 4 5 6 7 8 9 10 11