Spotify Popularity by Duration, Valence, Loudness, and Instrumentalness

Joseph Zegarelli

2025-08-14

Introduction

In this analysis, I examine how four song (track) characteristics, duration, valence (happiness), loudness and instrumentalness, influence a track's popularity on the popular music streaming application Spotify. I first use summary statistics to look at different bins, and then control for any possible skew or lack of clarity in the data by calculating the percentage of tracks that are popular within each bin. A Boxplot, Barchart and table visualizations accompany each analysis, followed by my observations.

Load and Inspect Data

```
spotify_raw <- read_csv("spotify.csv")
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~
```

```r
nrow(spotify_raw)
```

```
## [1] 114000
```

Clean and Transform Data

```r
spotify <- spotify_raw %>% distinct()
colSums(is.na(spotify))
```

```
##             ...1         track_id          artists       album_name
##                0                0                1                1
##       track_name       popularity      duration_ms         explicit
##                1                0                0                0
##     danceability           energy              key         loudness
##                0                0                0                0
##             mode      speechiness     acousticness instrumentalness
##                0                0                0                0
##         liveness          valence            tempo   time_signature
##                0                0                0                0
##      track_genre
##                0
```

```r
spotify <- spotify %>% drop_na() %>%
mutate(duration_sec = round(duration_ms / 1000, 0))
```

```r
spotify <- spotify %>% filter(duration_sec >= 30 & duration_sec <= 900)
```

```r
summary(spotify$duration_sec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    30.0   174.0   213.0   226.1   261.0   900.0
```

```r
summary(spotify$popularity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   17.00   35.00   33.25   50.00  100.00
```

```r
nrow(spotify)
```

```
## [1] 113833
```

Duration Analysis

Create Duration bins based on Z-Scores

```r
spotify <- spotify %>%
mutate(duration_z = (duration_sec - mean(duration_sec) ) / sd(duration_sec))
spotify <- spotify %>% mutate(duration_bin = case_when(duration_z < -2 ~"Very Short",
                                                       duration_z >= -2 & duration_z < -1 ~ "Short",
                                                       duration_z >= -1 & duration_z <= 1 ~ "Medium",
                                                       duration_z >1 & duration_z <= 2 ~ "Long",
                                                       duration_z > 2 ~ "Very Long")) %>%
  mutate(duration_bin = factor(
  duration_bin,
  levels = c("Very Short", "Short", "Medium", "Long", "Very Long")))
```

Create and View Summary Tables

```r
duration_summary <- spotify %>%
group_by(duration_bin) %>%
summarise(
mean_popularity = mean(popularity),
median_popularity = median(popularity),
count = n()
)
duration_popularity <- spotify %>%
  mutate(is_popular = ifelse(popularity >= 70, 1, 0)) %>%
  group_by(duration_bin) %>%
  summarise(
    percent_popular = mean(is_popular) * 100,
    sd_popular = sd(is_popular * 100),
    count =n(),
    se_popular = sd_popular / sqrt(count)
  )
```

```r
duration_summary %>%
  kable(
    caption = "Mean, Median and Count of Popularity by Duration Bin",
    digits = 2
  )
```

Table 1: Mean, Median and Count of Popularity by Duration Bin

duration_bin	mean_popularity	median_popularity	count
Very Short	21.99	21	647
Short	30.37	29	10883
Medium	34.08	36	88541
Long	31.97	31	8803
Very Long	28.58	26	4959

```r
duration_popularity %>%
  kable(
    caption = "Percentage of Popular Tracks by Duration Bin",
    digits = 2
  )
```

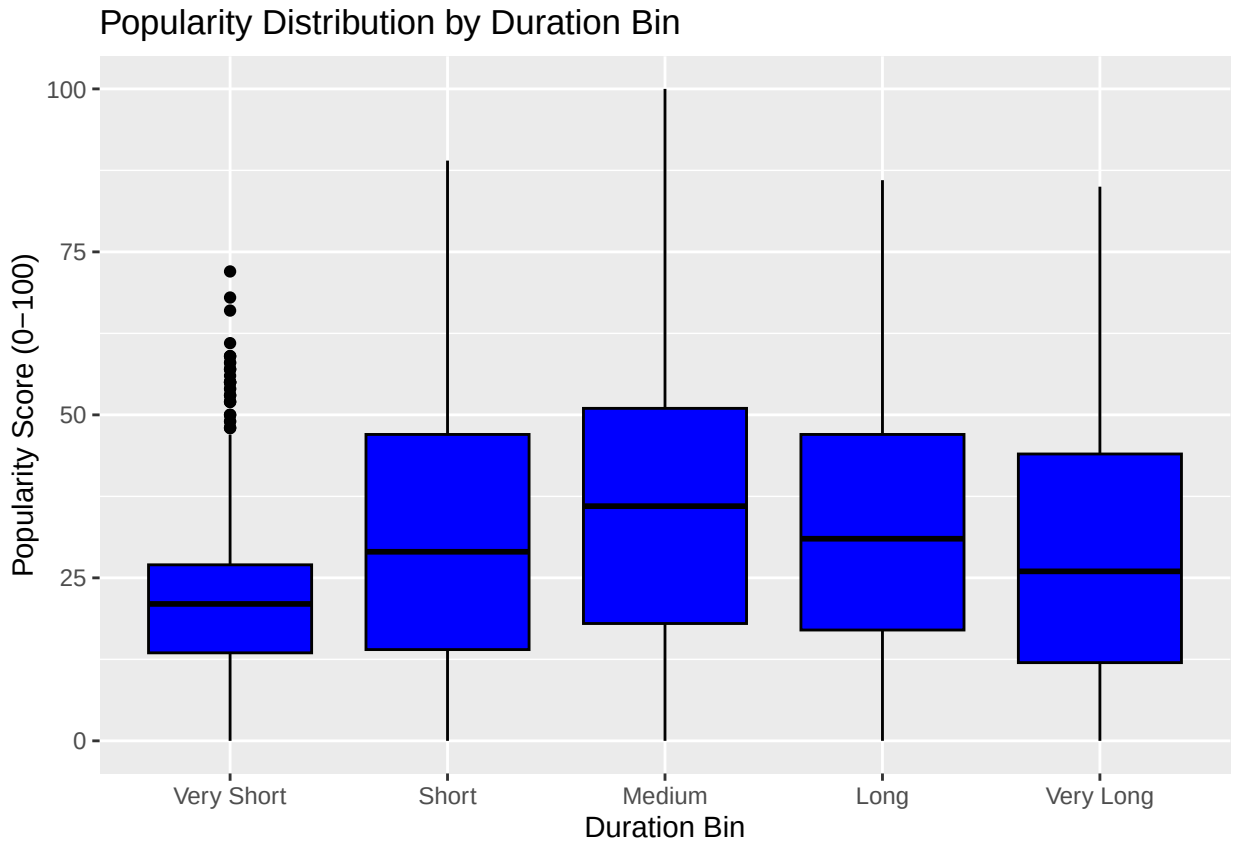
Table 2: Percentage of Popular Tracks by Duration Bin

duration_bin	percent_popular	sd_popular	count	se_popular
Very Short	0.15	3.93	647	0.15
Short	2.28	14.92	10883	0.14
Medium	5.54	22.88	88541	0.08
Long	2.86	16.68	8803	0.18
Very Long	1.27	11.20	4959	0.16

Duration Visualizations

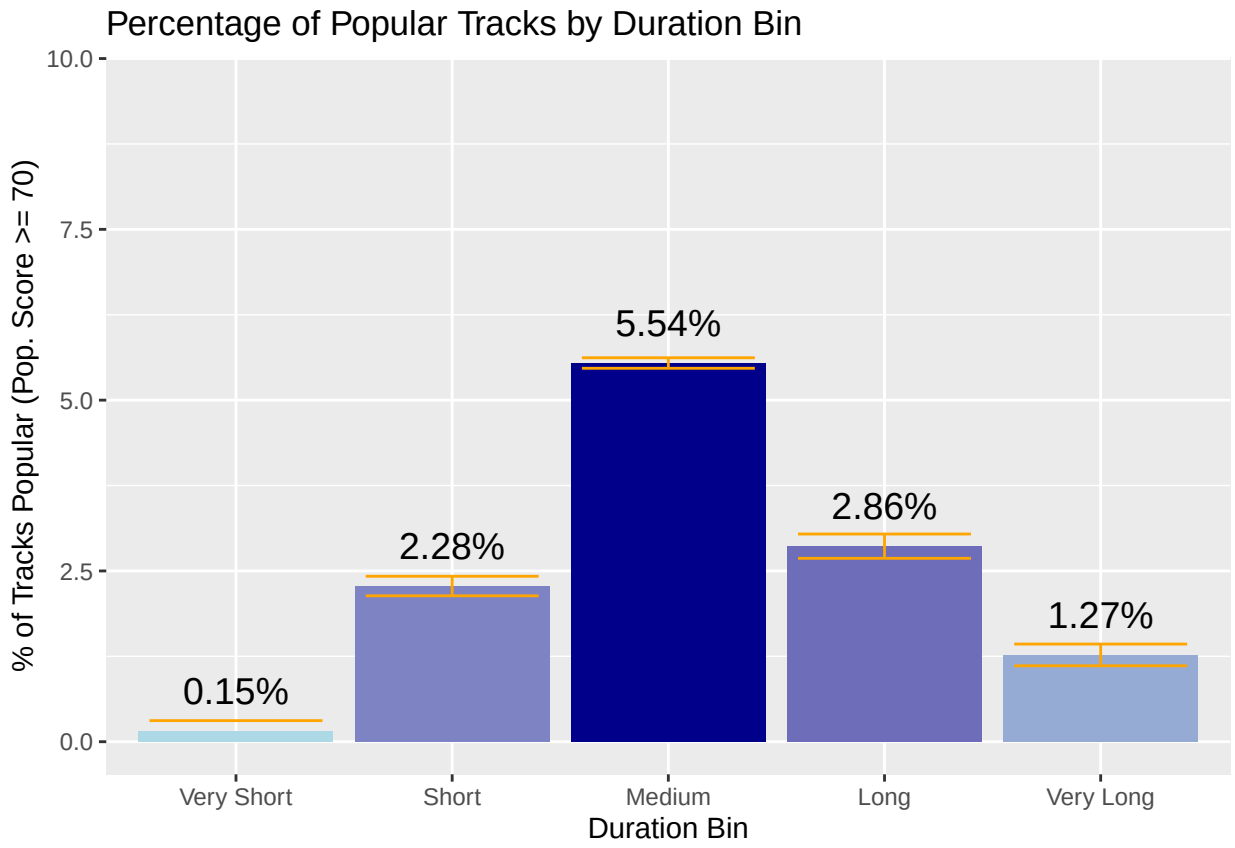
Popularity Score by Duration Bin Boxplot

```
ggplot(spotify, aes(x = duration_bin, y = popularity)) +
  geom_boxplot(fill = "blue", color = "black") +
  labs(
    title = "Popularity Distribution by Duration Bin",
    x = "Duration Bin",
    y = "Popularity Score (0-100)"
  )
```



Percentage of Popular Tracks by Duration Bin Barchart

```
ggplot(duration_popularity, aes(x = duration_bin, y = percent_popular, fill = percent_popular)) +
  geom_col(show.legend = FALSE) +
  geom_errorbar(aes(
    ymin = percent_popular - se_popular,
    ymax = percent_popular + se_popular),
    width = 0.8, color = "orange"
  ) +
  geom_text(aes(label = paste0(round(percent_popular, 2), "%")),
            vjust = -1,
            size = 5) +
  labs(
    title = "Percentage of Popular Tracks by Duration Bin",
    x = "Duration Bin",
    y = "% of Tracks Popular (Pop. Score >= 70)"
  ) +scale_fill_gradient(low = "lightblue", high = "darkblue") +
  ylim(0, max(duration_popularity$percent_popular) + 4)
```



My Observations on Duration vs. Popularity

The Medium bin shows the highest popularity score and a considerably higher percentage of popular tracks per bin. Short and Long tracks show a similar popularity, and while the popularity of Very Long tracks drops off, they're still in that approximate cluster. After controlling for count, my main findings are Very Short

tracks have almost no chance of becoming popular and Medium tracks are considerably more likely to be popular. I expected the data to show that shorter songs were more popular, however the data was slightly skewed in the opposite direction. Artists aspiring to have popular tracks should aim for Medium-length tracks and avoid Very Short ones at all costs.

Valence Analysis

(Valence = Happiness, Higher Valence Score = Happier Song)

Create Valence Bins based on Quartiles

```
spotify <- spotify %>%
  mutate(valence_bin = ntile(valence, 5)) %>%
  mutate(valence_bin = case_when(
    valence_bin == 1 ~ "Very Low",
    valence_bin == 2 ~ "Low",
    valence_bin == 3 ~ "Medium",
    valence_bin == 4 ~ "High",
    valence_bin == 5 ~ "Very High"
  )) %>%
  mutate(valence_bin = factor(valence_bin, levels = c(
    "Very Low", "Low", "Medium", "High", "Very High")))
```

Create and View Summary Tables

```
valence_summary <- spotify %>%
  group_by(valence_bin) %>%
  summarise(
    mean_popularity = mean(popularity),
    median_popularity = median(popularity),
    count = n()
  )
valence_popularity <- spotify %>%
  mutate(is_popular = ifelse(popularity >= 70, 1, 0)) %>%
  group_by(valence_bin) %>%
  summarise(
    percent_popular = mean(is_popular) * 100,
    sd_popular = sd(is_popular * 100),
    count =n(),
    se_popular = sd_popular / sqrt(count)
  )
```

```
valence_summary %>%
  kable(
    caption = "Mean, Median and Count of Popularity by Valence Bin",
    digits = 2
  )
```

Table 3: Mean, Median and Count of Popularity by Valence Bin

valence_bin	mean_popularity	median_popularity	count
Very Low	32.77	34	22767
Low	35.34	38	22767
Medium	34.83	37	22767
High	32.42	33	22766
Very High	30.90	32	22766

```
valence_popularity %>%
  kable(
    caption = "Percentage of Popular Tracks by Valence Bin",
    digits = 2
  )
```

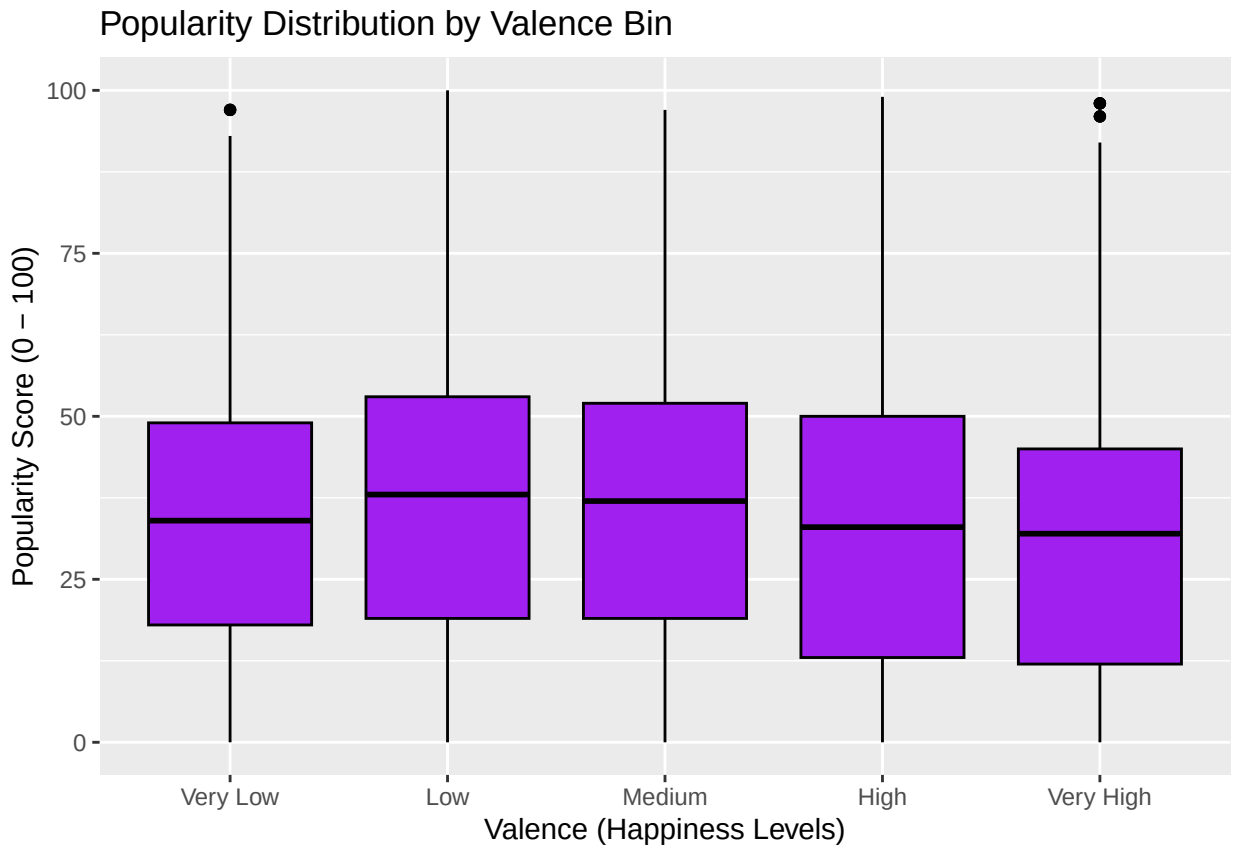
Table 4: Percentage of Popular Tracks by Valence Bin

valence_bin	percent_popular	sd_popular	count	se_popular
Very Low	2.92	16.84	22767	0.11
Low	4.81	21.40	22767	0.14
Medium	6.17	24.06	22767	0.16
High	5.56	22.92	22766	0.15
Very High	4.57	20.89	22766	0.14

Valence Visualizations

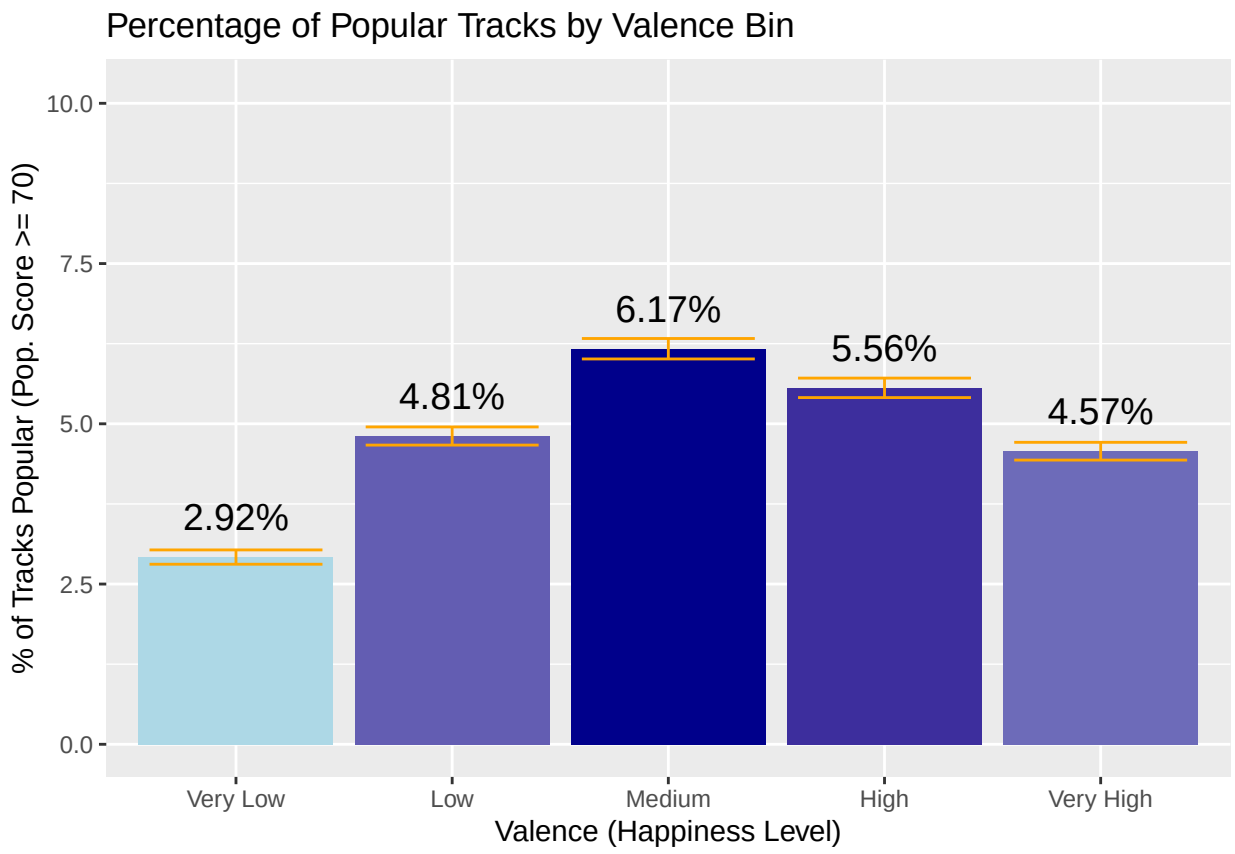
Popularity Score by Valence Bin Boxplot

```
ggplot(spotify, aes(x = valence_bin, y = popularity)) +
  geom_boxplot(fill = "purple", color = "black") +
  labs(
    title = "Popularity Distribution by Valence Bin",
    x = "Valence (Happiness Levels)",
    y = "Popularity Score (0 - 100)"
  )
```



Percentage of Popular Tracks by Valence Bin Barchart

```
ggplot(valence_popularity, aes(x = valence_bin, y = percent_popular,
                               fill = percent_popular)) +
  geom_col(show.legend = FALSE) +
  geom_errorbar(aes(
    ymin = percent_popular - se_popular,
    ymax = percent_popular + se_popular),
    width = 0.8, color = "orange"
  ) +
  geom_text(aes(label = paste0(round(percent_popular, 2), "%")),
  vjust = -1,
  size = 5) +
  labs(
  title = "Percentage of Popular Tracks by Valence Bin",
  x = "Valence (Happiness Level)",
  y = "% of Tracks Popular (Pop. Score >= 70)",
  ) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  ylim(0, max(valence_popularity$percent_popular) + 4)
```



My Observations on Valence vs. Popularity

Popularity based on Valence is relatively uniform. Medium and High Valence scores are the strongest, while Low and Very High bins follow closely. Very Low Valence is the only bin with a considerably lower chance of popularity (still about 2/3 as likely to be popular as the Very High and Low bins). I expected a larger difference in popularity based on valence, especially skewed towards the low end. Sad songs stand out as more popular to me, but perhaps those lyrics tend to be more memorable. Artists aiming for popularity on Spotify would benefit more from spending time on other aspects of their track than Valence. If Valence is considered then aiming for the Medium to High range is the most effective.

Loudness Analysis

Create Loudness Bins based on Quartiles

```
spotify_loud <- spotify %>%
  filter(loudness >= -15 & loudness <= -3) %>%
  mutate(loudness_bin = ntile(loudness, 5)) %>%
  mutate(loudness_bin = case_when(
    loudness_bin == 1 ~ "Very Low",
    loudness_bin == 2 ~ "Low",
    loudness_bin == 3 ~ "Medium",
    loudness_bin == 4 ~ "High",
```

```
    loudness_bin == 5 ~ "Very High"
  )) %>%
  mutate(loudness_bin = factor(loudness_bin,
                               levels = c("Very Low", "Low", "Medium", "High", "Very High")))
```

Create and View Summary Tables

```
loudness_summary <- spotify_loud %>%
  group_by(loudness_bin) %>%
  summarise(mean_popularity = mean(popularity),
            median_popularity = median(popularity),
            count = n())
loudness_popularity <- spotify_loud %>%
  mutate(is_popular = ifelse(popularity >= 70, 1, 0)) %>%
  group_by(loudness_bin) %>%
  summarise(
    percent_popular = mean(is_popular) * 100,
    sd_popular = sd(is_popular * 100),
    count =n(),
    se_popular = sd_popular / sqrt(count)
  )
```

```
loudness_summary %>%
  kable(
    caption = "Mean, Median and Count of Popularity by Loudness Bin",
    digits = 2
  )
```

Table 5: Mean, Median and Count of Popularity by Loudness Bin

loudness_bin	mean_popularity	median_popularity	count
Very Low	32.49	34	19674
Low	34.51	37	19674
Medium	34.40	37	19674
High	34.05	36	19674
Very High	33.04	33	19673

```
loudness_popularity %>%
  kable(
    caption = "Percentage of Popular Tracks by Loudness Bin",
    digits = 2
  )
```

Table 6: Percentage of Popular Tracks by Loudness Bin

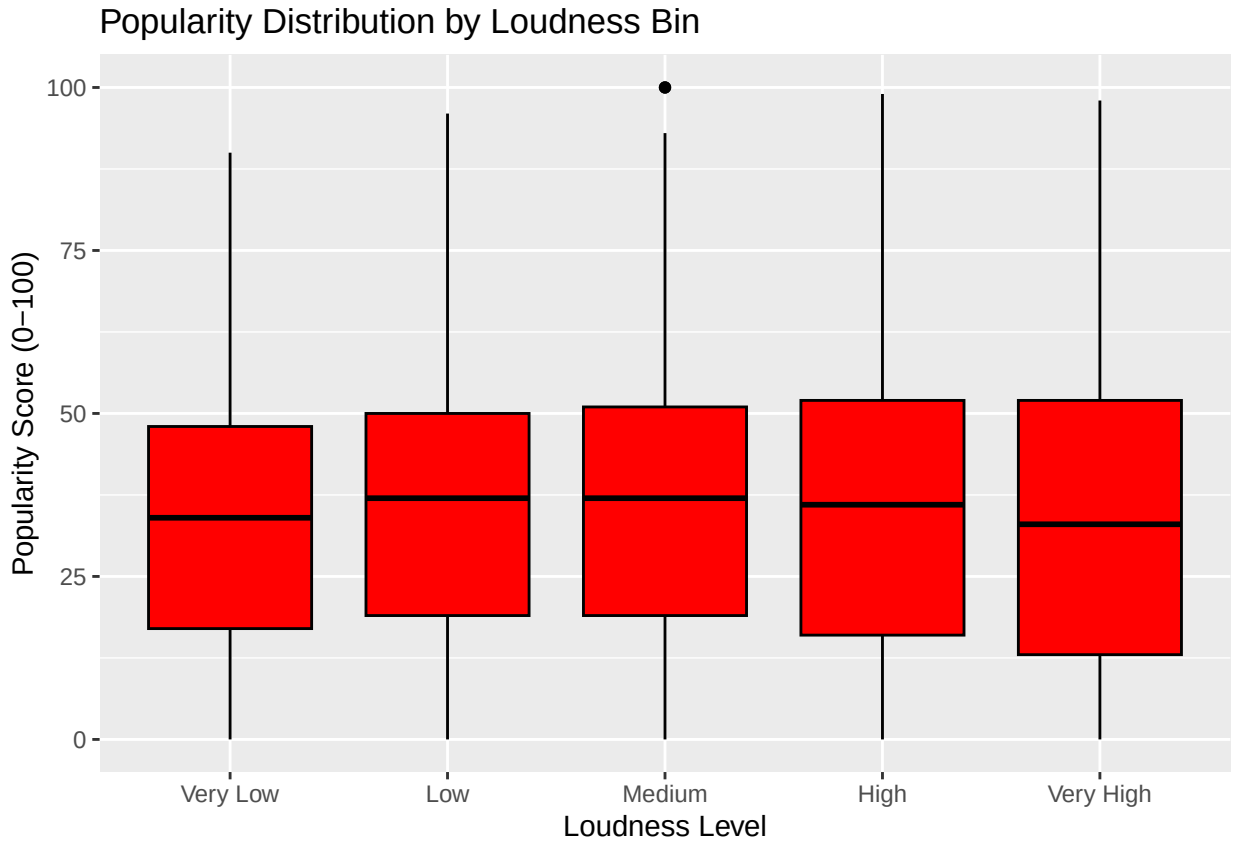
loudness_bin	percent_popular	sd_popular	count	se_popular
Very Low	3.04	17.18	19674	0.12
Low	4.30	20.29	19674	0.14

loudness_bin	percent_popular	sd_popular	count	se_popular
Medium	5.18	22.17	19674	0.16
High	6.52	24.69	19674	0.18
Very High	6.77	25.12	19673	0.18

Loudness Visualizations

Popularity Score by Loudness Bin Boxplot

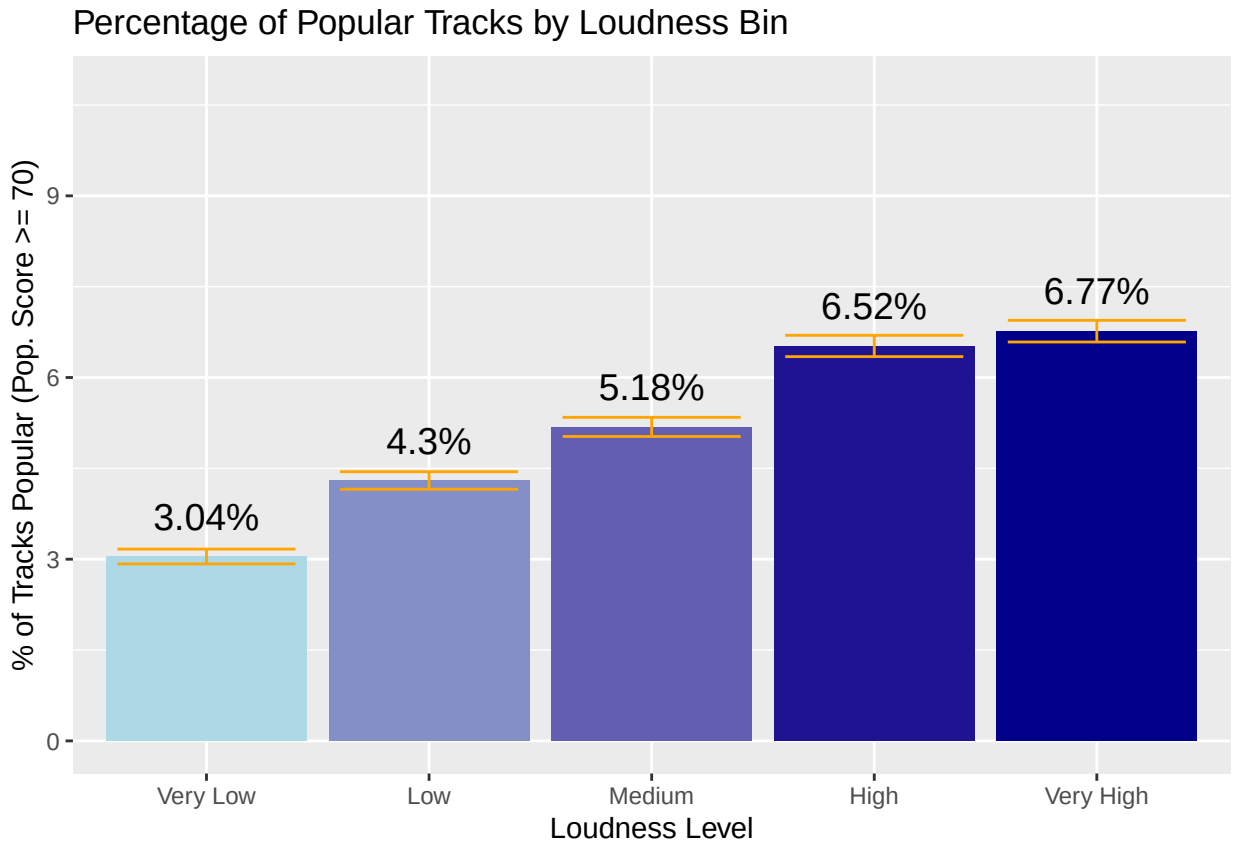
```
ggplot(spotify_loud, aes(x = loudness_bin, y = popularity)) +
  geom_boxplot(fill = "red", color = "black") +
  labs(
    title = "Popularity Distribution by Loudness Bin",
    x = "Loudness Level",
    y = "Popularity Score (0-100)"
  )
```



Percentage of Popular Tracks by Loudness Bin Barchart

```
ggplot(loudness_popularity, aes(
  x = loudness_bin, y = percent_popular, fill = percent_popular)) +
```

```r
  geom_col(show.legend = FALSE) +
  geom_errorbar(aes(
    ymin = percent_popular - se_popular,
    ymax = percent_popular + se_popular),
    width = 0.8, color = "orange"
  ) +
  geom_text(aes(label = paste0(round(percent_popular, 2), "%")),
            vjust = -1, size = 5) +
  labs(
    title = "Percentage of Popular Tracks by Loudness Bin",
    x= "Loudness Level",
    y = "% of Tracks Popular (Pop. Score >= 70)"
  ) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  ylim(0, max(loudness_popularity$percent_popular) + 4)
```



My Observations on Loudness vs. Popularity

Loudness vs. Popularity shows a clear trend: the louder a track is, the more popular it will be. It's evident when looking at the percentage of popular tracks by loudness bin that there is clear skew towards higher loudness, with Very High being the peak and trending downwards to Very Low. I fully expected this as the "Loudness Wars" is a common topic in the audio/music industry. Artists aiming for popularity on Spotify can optimize their chances by spending their time and money acquiring the best mixing-and-mastering engineer possible. These artists would also benefit from practicing clean takes in the studio so that it is possible for those engineers to push loudness to the maximum.

Instrumentalness Analysis

(Tracks with lower Instrumentalness scores have more vocals throughout)

Filter to Two Groups and Check Counts

(Two groupings contain ~90% of tracks)

```
spotify_instr_filtered <- spotify %>%
  mutate(instr_group = case_when(
    instrumentalness <= 0.05 ~ "Non-Instrumental",
    instrumentalness >= 0.8 ~ "Highly Instrumental",
    TRUE ~ "Other"
  )) %>%
  filter(instr_group != "Other") %>%
  mutate(instr_group = factor(
    instr_group, levels = c("Non-Instrumental",
                            "Highly Instrumental")))
```

```
table(spotify_instr_filtered$instr_group)
```

```
##
##    Non-Instrumental Highly Instrumental
##               85492               12841
```

Create and View Summary Tables

```
instr_summary <- spotify_instr_filtered %>%
  group_by(instr_group) %>%
  summarise(
    mean_popularity = mean(popularity),
    median_popularity = median(popularity),
    count = n()
  )
instr_group_popularity <- spotify_instr_filtered %>%
mutate(is_popular = ifelse(popularity >= 70, 1, 0)) %>%
  group_by(instr_group) %>%
  summarise(
    percent_popular = mean(is_popular) * 100,
    sd_popular = sd(is_popular * 100),
    count = n(),
    se_popular = sd_popular / sqrt(count)) %>%
  mutate(percent_total = count / sum(count) * 100)
```

```
instr_summary %>%
  kable(
    caption = "Mean, Median and Count of Popularity by Instrumentalness Group",
    digits = 2
  )
```

Table 7: Mean, Median and Count of Popularity by Instrumentalness Group

instr_group	mean_popularity	median_popularity	count
Non-Instrumental	34.38	37	85492
Highly Instrumental	28.18	24	12841

```
instr_group_popularity %>%
  kable(
    caption = "Percentage of Popular Tracks by Instrumentalness Group",
    digits = 2
  )
```

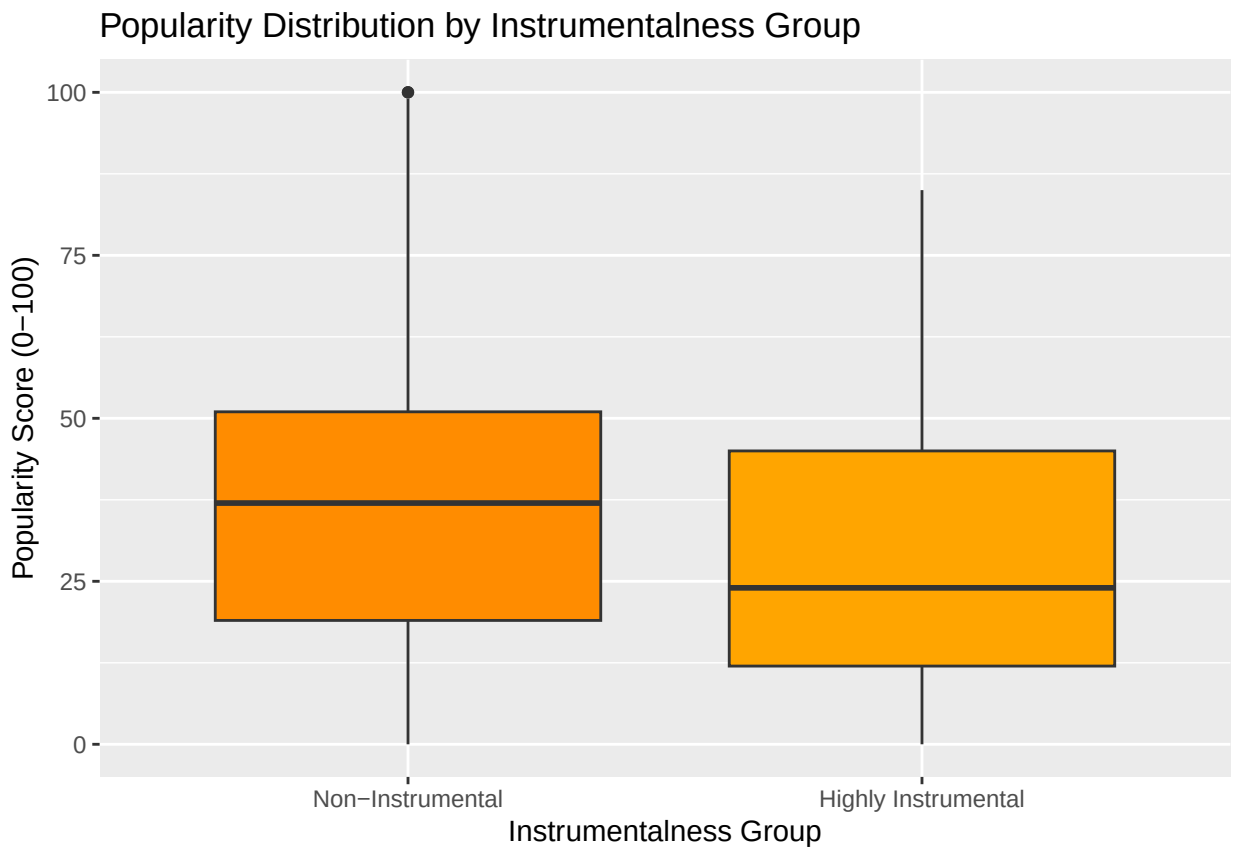
Table 8: Percentage of Popular Tracks by Instrumentalness Group

instr_group	percent_popular	sd_popular	count	se_popular	percent_total
Non-Instrumental	5.81	23.39	85492	0.08	86.94
Highly Instrumental	0.64	7.97	12841	0.07	13.06

Instrumentalness Visualizations

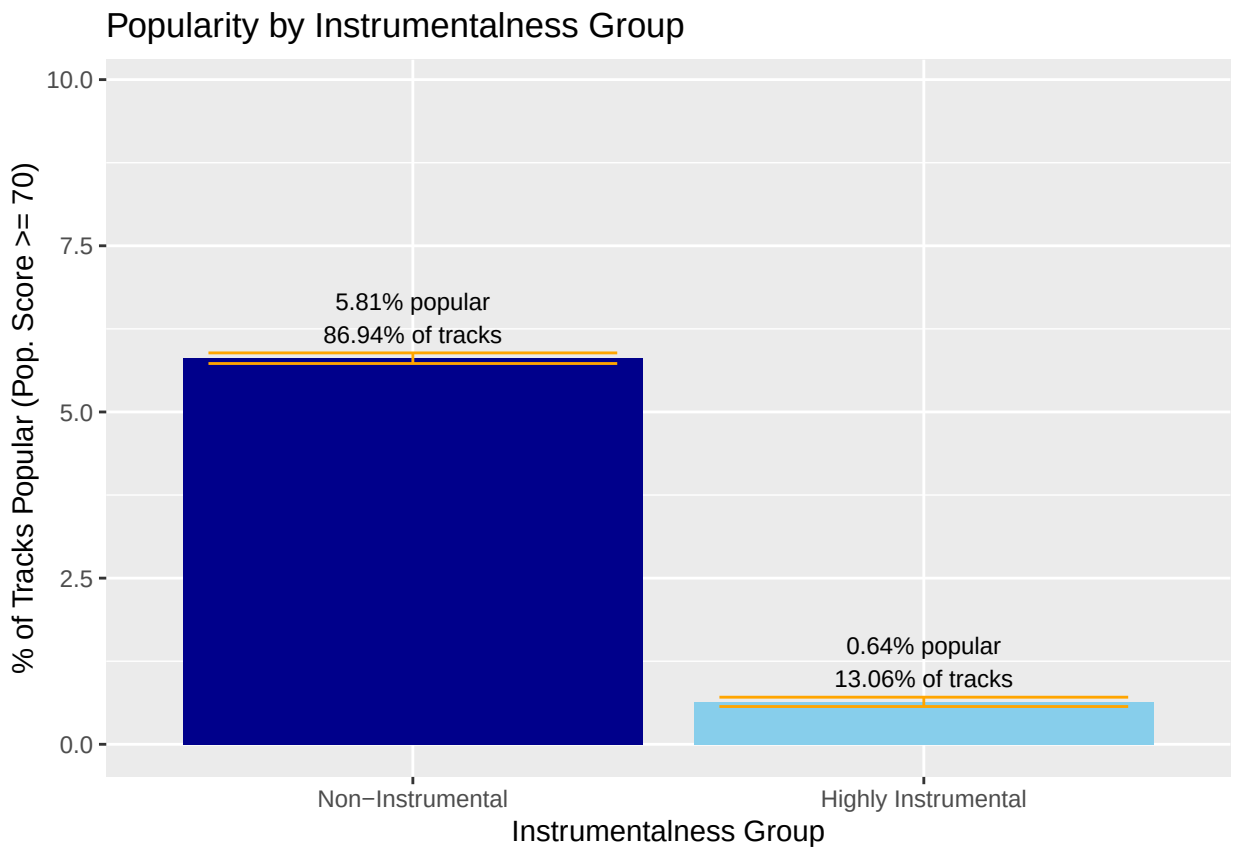
Popularity Score by Instrumentalness Group Boxplot

```
ggplot(spotify_instr_filtered, aes(x = instr_group, y = popularity,
                                   fill = instr_group)) +
  geom_boxplot() +
  labs(
    title = "Popularity Distribution by Instrumentalness Group",
    x = "Instrumentalness Group",
    y = "Popularity Score (0-100)"
  ) +
  scale_fill_manual(values = c(
    "Non-Instrumental" ="darkorange",
    "Highly Instrumental" = "orange")) +
  ylim(0, 100) +
  theme(legend.position = "none")
```



Percentage of Popular Tracks by Instrumentalness Group Barchart

```
ggplot(instr_group_popularity, aes(
  x = instr_group, y = percent_popular, fill = instr_group)) +
  geom_col(show.legend = FALSE) +
  geom_errorbar(aes(
    ymin = percent_popular - se_popular,
    ymax = percent_popular + se_popular),
    width = 0.8, color = "orange"
  ) +
    geom_text(aes(label = paste0(round(percent_popular, 2),
                                 "% popular", "\n", round(percent_total, 2),
                                 "% of tracks")),
              vjust = -0.3, size = 3) +
    labs(
      title = "Popularity by Instrumentalness Group",
      x = "Instrumentalness Group",
      y = "% of Tracks Popular (Pop. Score >= 70)"
    ) +
    ylim(0, max(instr_group_popularity$percent_popular) + 4) +
    scale_fill_manual(values = c("Non-Instrumental" = "darkblue", "Highly Instrumental" = "skyblue"))
```



My Observations on Instrumentalness vs. Popularity

In order to be popular on Spotify, constant vocals in a track are necessary. It is clear throughout the entire analysis that Non-Instrumental tracks (tracks with extremely low Instrumentalness scores) are significantly more popular than Highly Instrumental tracks. This was expected as most popular songs immediately begin with the chorus and end immediately after the final chorus. Given the rise of popular electronic music artists such as Avicii, Flume and The Chainsmokers, as well as the ever-growing Electronic Music Industry, the difference is slightly more stark than one would think. Artists aiming for popularity on Spotify would benefit from becoming better lyricists capable of filling any space not occupied by vocals with ad-libs or other fitting background vocals.

Conclusion

This analysis examined how four song (track) characteristics, duration, valence (happiness), loudness and instrumentalness, influence a track's popularity on Spotify. Duration showed a medium impact on popularity, with Medium-length tracks performing best and Very Short-length tracks being significantly unpopular. Valence displayed very little influence on popularity, though Very Sad tracks performed poorly and Medium-Valence tracks performed the best. Loudness and Instrumentalness have a huge effect with louder tracks and those filled with vocals much more likely to be popular, while Highly Instrumental tracks rarely succeed.

Artists aiming for popularity should focus on filling their track with vocals using ad-libs, background vocals, or both. Just as importantly, I recommend artists spend the extra time and money hiring the proper mixing-and-mastering engineer while also nailing their performance in the studio. This will allow the final master to be pushed to its maximum volume. Keep track lengths similar to other successful songs, and don't worry

too much about how happy or sad your song is as long as you don't go to the extreme and end up making the next version of Christmas Shoes.

Future analysis can examine how different genres influence popularity, potentially filtering out genres that are particularly unpopular. It would be interesting to test different criteria for popularity or look at the intersection of multiple variables and examine how they affect success. Additionally, calculating correlation coefficients and diving into more distinct factors of popularity by removing quiet and Highly Instrumental tracks from the analysis can provide deeper insights.

Executive Summary

This report examines how duration, valence (happiness), loudness and instrumentalness influence a track's popularity on Spotify. Medium-length tracks perform best, while Very Short tracks rarely succeed. Loud tracks with consistent vocals are significantly more likely to be popular, whereas Highly Instrumental tracks almost never achieve popularity. Artists should focus on filling tracks with vocals and working with mixing-and-mastering engineers to achieve maximum loudness. Track length should be similar to other successful tracks.