Olivia Jones, Sienna Qin, Kathleen Setiadi

Spotify

01

Introduction

Objectives

For **artists** and producers to know **how** to structure certain songs to **increase the streaming**. It’s hard for new upcoming artists to create hits and viral songs because there are so many variables. We want people to add songs to their playlists (to have multiple streams) rather than just listening to it once.

Variables

RESPONSE

Popularity
How well the song performs on the Spotify platform.

PREDICTOR

1. ### Loudness
   The overall loudness of a track in decibels (dB).

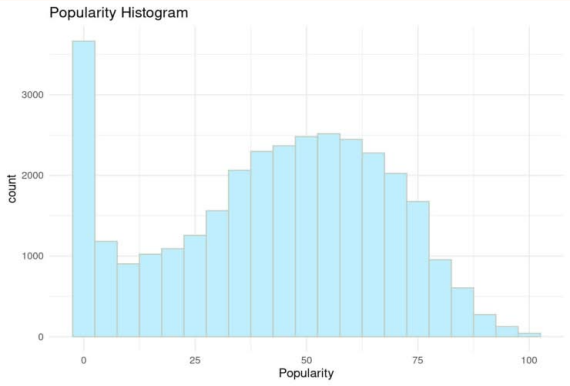
2. ### Duration
   The duration, length of the song in milliseconds.

3. ### Tempo
   Overall estimated tempo of a track in beats per minute(BPM).

02

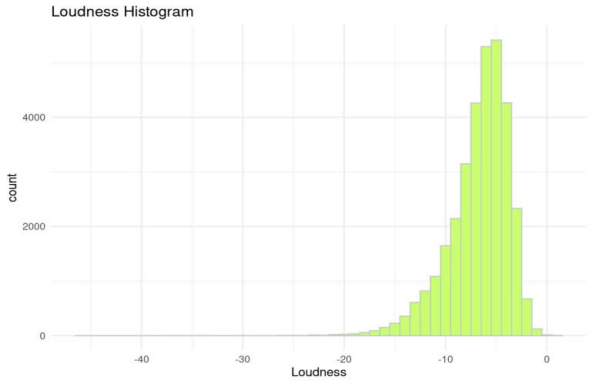
Plots

Variable Summaries




```
> summary(spotify$Popularity)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00   24.00   45.00   42.48   62.00  100.00
```

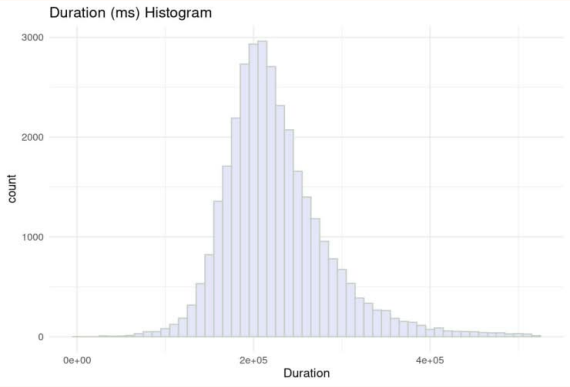
Popularity spans the full 0–100 scale but centers at a median of 45, indicating a long-tail market where truly viral tracks are outliers




```
> summary(spotify$Loudness)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-46.448   -8.171   -6.166   -6.720   -4.645    1.275
```

Most songs are around -6.7 dB, and the IQR is pretty small, which means the loudness of most tracks doesn't vary much. Songs are made to be similarly loud, and there isn't much difference between the quietest and loudest ones in the middle 50%.

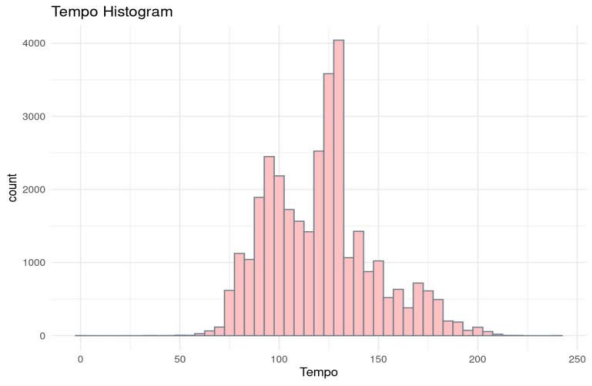
Variable Summaries




```
> summary(spotify$Duration)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
4000	187819	216000	225800	253585	517810

Track durations tend to cluster around 216,000 ms (about 3.6 minutes), which lines up with typical radio-friendly lengths. Still, there are some creative outliers that break away from the norm.

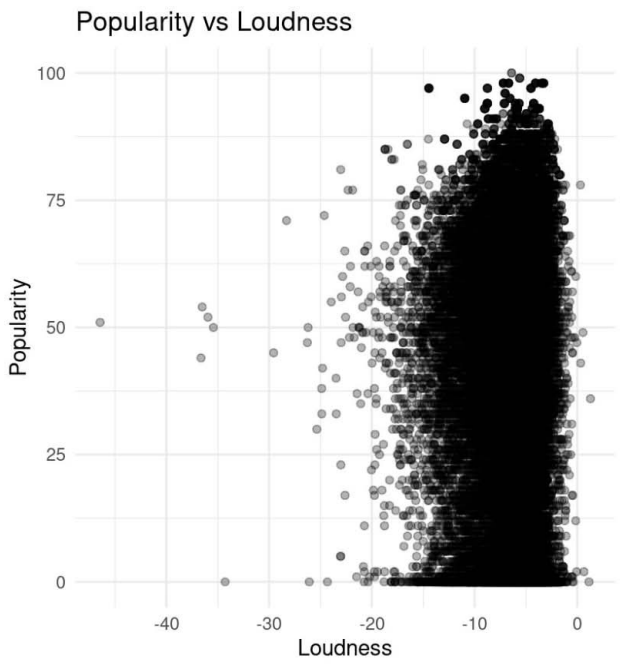



```
> summary(spotify$Tempo)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.00	99.96	121.98	120.88	133.92	239.44

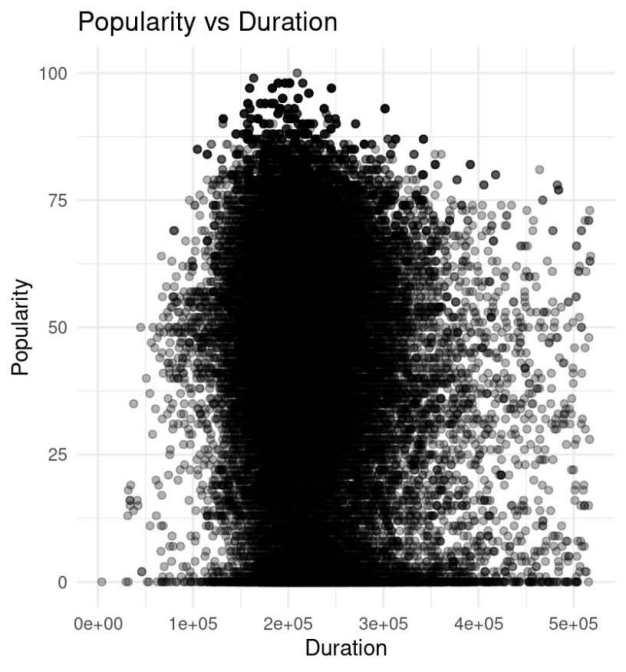
Tempo averages around 121 BPM, which makes sense for pop and dance music. But it still spans a wide range—from super slow sections to 239 BPM—showing just how much variety there is in pacing across genres.

Response vs Predictor



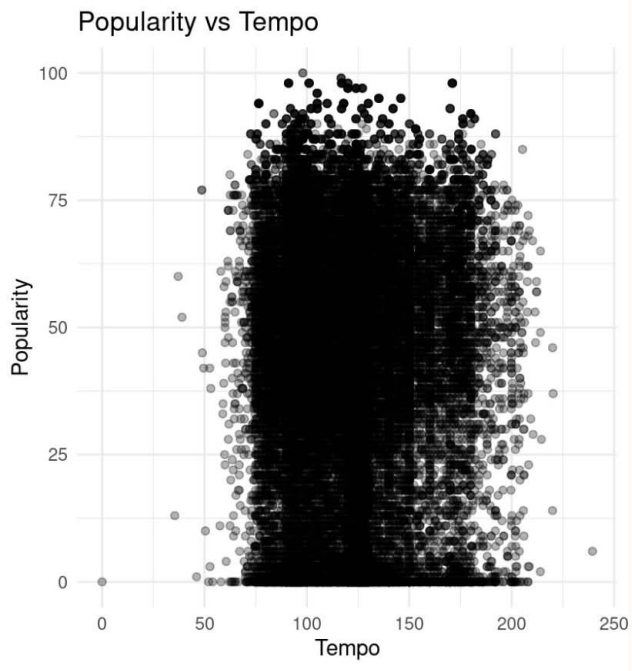
Popularity vs Loudness shows a very weak positive trend: louder songs tend to get slightly more streams, but the effect is minimal. The wide spread of points makes it clear that loudness isn't a strong predictor of popularity on its own.

Response vs Predictor



Popularity vs Duration is essentially flat, with popularity dispersed across the entire length spectrum; this suggests that once a song meets baseline expectations in the ≈3 to 4 minutes length, length is not a primary driver of streaming performance

Response vs Predictor

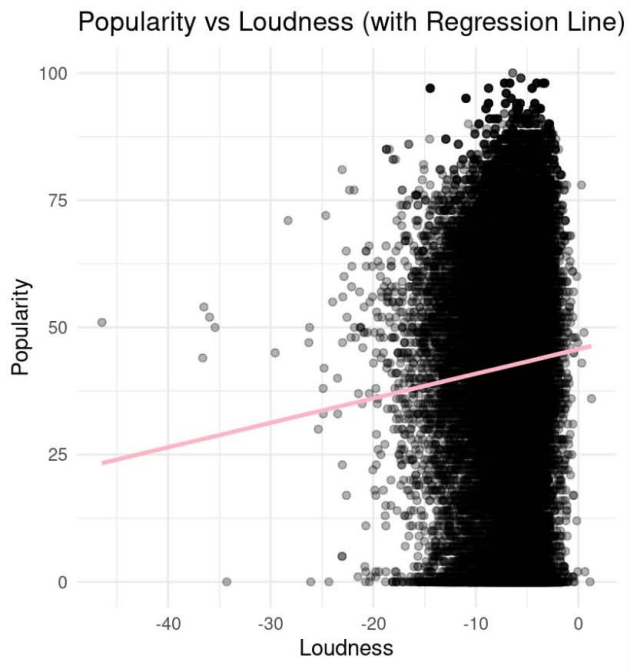


Popularity vs Tempo shows a mild upward trend concentrated in the 100–140 BPM band; while faster tracks gain a modest advantage, the wide vertical spread indicates tempo alone cannot explain popularity differentials.

Analysis: Response vs Predictor

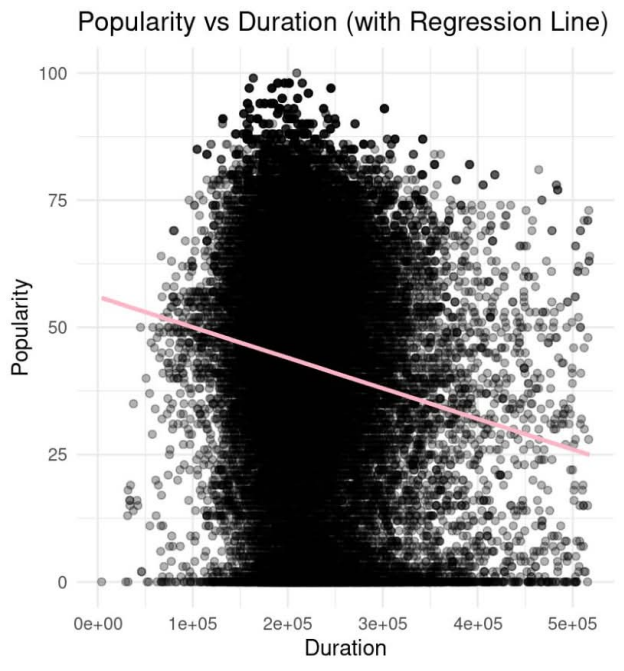
The plots prioritise Loudness as the most actionable predictor for subsequent modelling, position Tempo as a secondary variable, and relegate Duration to a control attribute.

Response vs Predictor with Regression



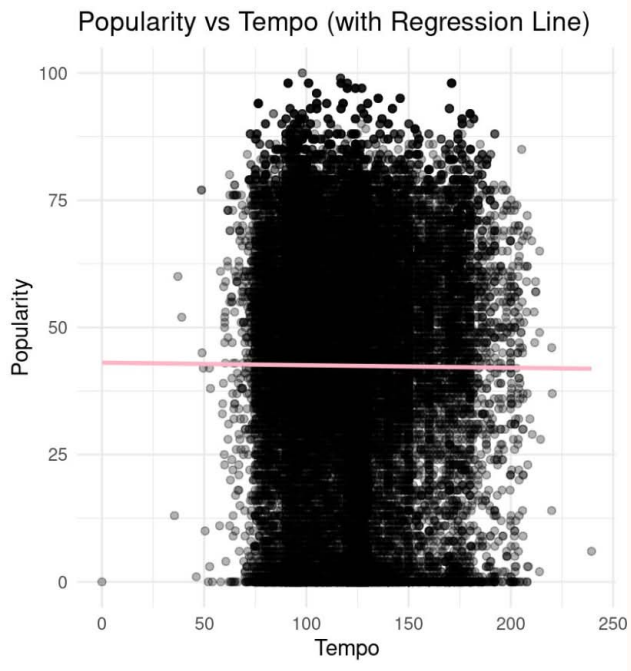
Popularity vs Loudness: the regression line shows a mild upward trend, suggesting a weak positive relationship—louder songs are slightly more likely to be popular. Still, the wide spread of points around the line shows that loudness alone isn't a strong predictor of streams.

Response vs Predictor with Regression



Popularity vs Duration: the regression line shows a slight negative trend—longer songs tend to be a bit less popular. Still, the relationship is weak, and once a track falls within the usual 3–4 minute range, duration doesn't add much predictive value for streaming performance.

Response vs Predictor with Regression

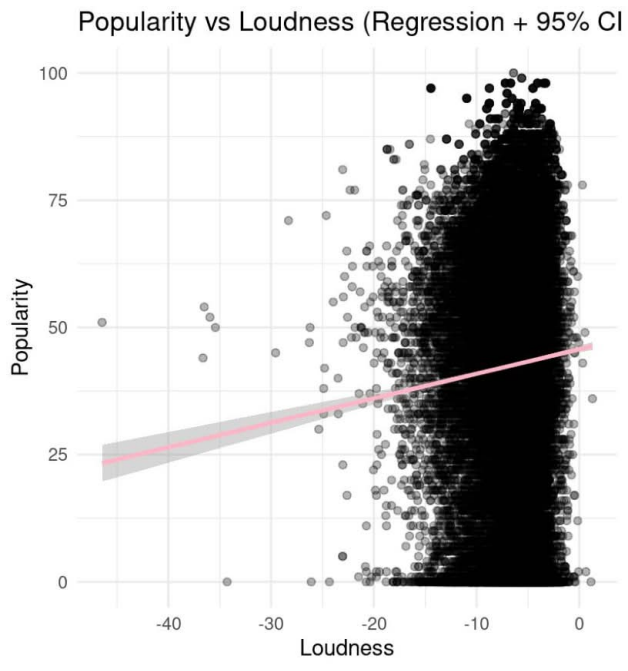


Popularity vs Tempo: the regression line is nearly flat, showing little to no relationship between tempo and popularity. The wide vertical spread suggests that tempo has limited influence on streaming performance once other factors—like loudness—are accounted for.

Analysis: Response vs Predictor with Regression

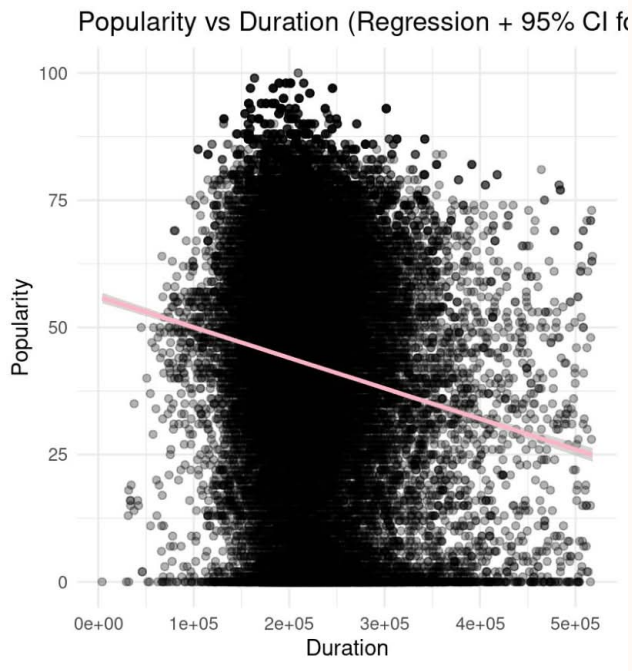
These fitted lines highlight loudness as the clearest predictor of popularity, suggest tempo might offer limited support, and show duration has minimal influence on its own.

Response vs Predictor with Regression and 95% Confidence Interval Bands



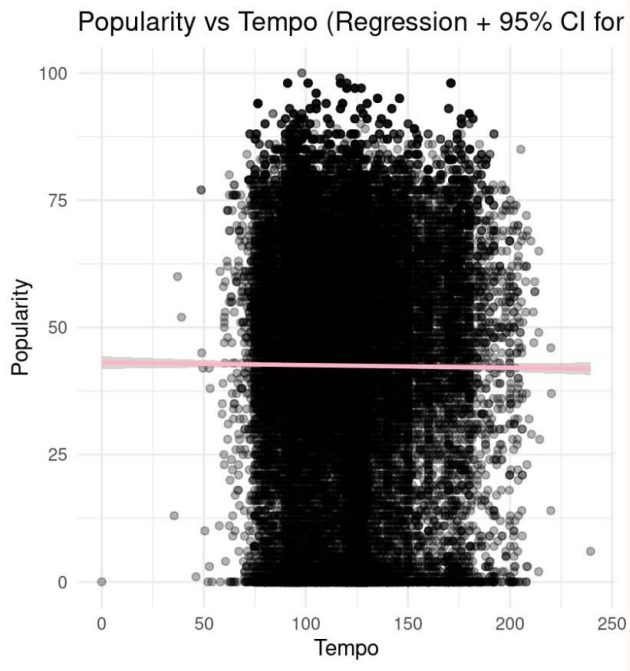
Popularity vs Loudness: the CI band stays tight between about –10 dB and –4 dB, meaning mean-popularity estimates are more reliable for tracks mastered at typical loudness levels. It widens at the quiet and ultra-loud extremes, where data are sparse—signaling more uncertainty in those predictions.

Response vs Predictor with Regression and 95% Confidence Interval Bands



Popularity vs Duration: the downward-sloping regression line sits within a broad, nearly parallel CI band, suggesting that while there's a slight negative trend, average popularity stays fairly stable across song lengths once the typical 3–4 minute range is met.

Response vs Predictor with Regression and 95% Confidence Interval Bands

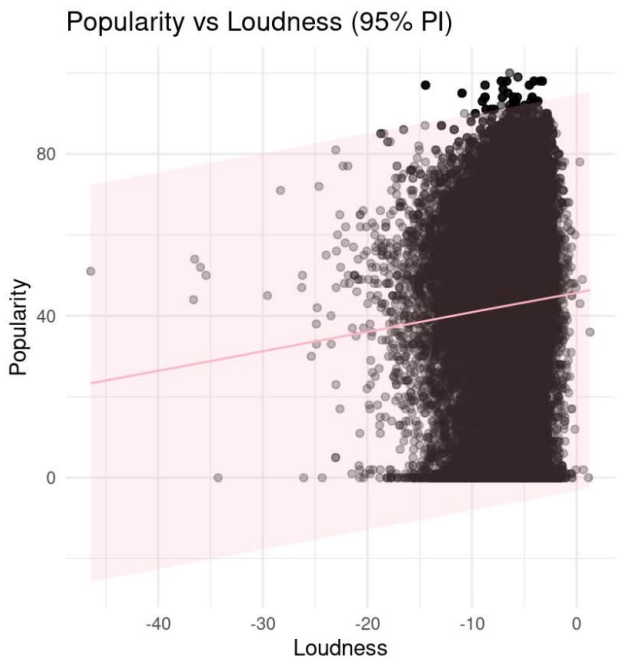


Popularity vs Tempo: the wider and more variable CI band suggests that tempo has a weaker, less consistent effect on popularity. Uncertainty is especially high below 90 BPM and above 150 BPM, pointing to genre-driven variation rather than a clear overall trend.

Analysis: Response vs Predictor with Regression and 95% Confidence Interval Bands

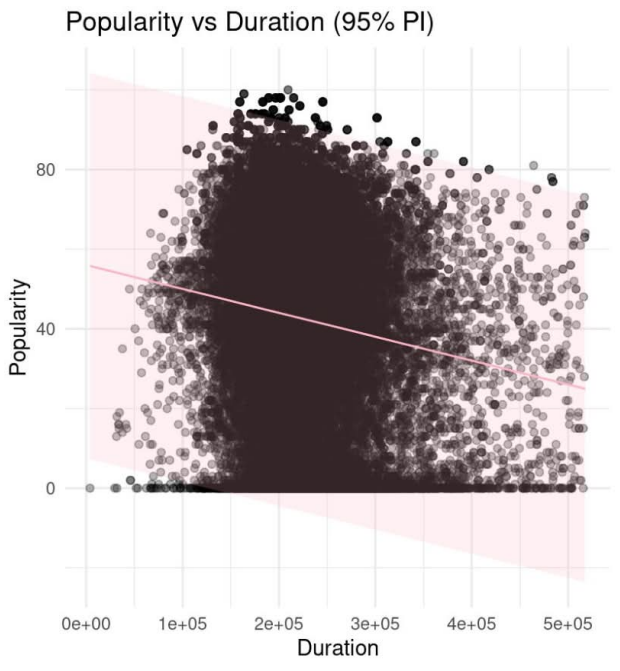
These CIs reinforce loudness as the most reliable predictor of expected popularity, position tempo as informative but uncertain, and suggest that duration adds little explanatory value.

Response vs Predictor with Regression and 95% Prediction Interval Bands



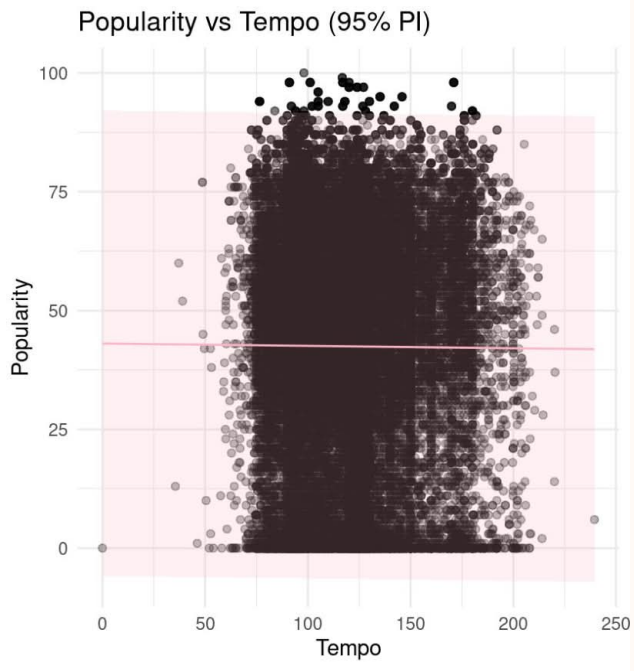
Popularity vs Loudness: although the PI ribbon is the narrowest of the three, it still spans roughly ±30 popularity points through the mainstream –10 dB to –4 dB zone, showing that even tracks mastered at similar loudness can land anywhere from middling to hit status; the ribbon flares sharply at the very quiet and ultra-loud ends, where data are scarce and forecasts become highly uncertain.

Response vs Predictor with Regression and 95% Prediction Interval Bands



Popularity vs Duration: the ribbon is both wide and nearly horizontal, confirming that length offers virtually no predictive precision; two songs of equal duration can differ by 60–70 popularity points, rendering this variable ineffective for forecasting individual success.

Response vs Predictor with Regression and 95% Prediction Interval Bands

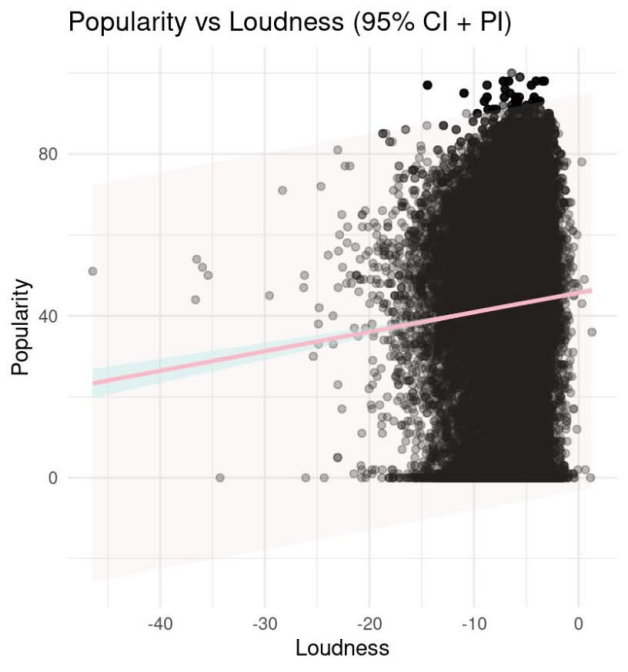


Popularity vs Tempo: the PI band is wider here than in the loudness plot and stays broad across all BPM levels. This highlights that tempo alone can’t reliably predict a track’s success—genre, mood, and marketing likely play a larger role in shaping popularity.

Analysis: Response vs Predictor with Regression and 95% Prediction Interval Bands

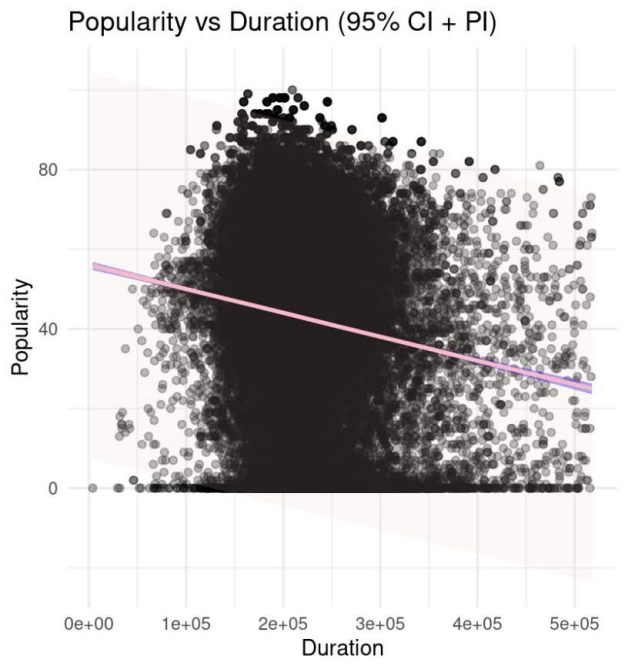
These PI bands highlight loudness as the most stable—though still imperfect—predictor of single-track performance, reveal tempo's high forecast uncertainty, and confirm that duration offers little to no predictive value at the individual-song level.

Response vs Predictor with Regression and 95% Confidence Interval and Prediction Interval Bands



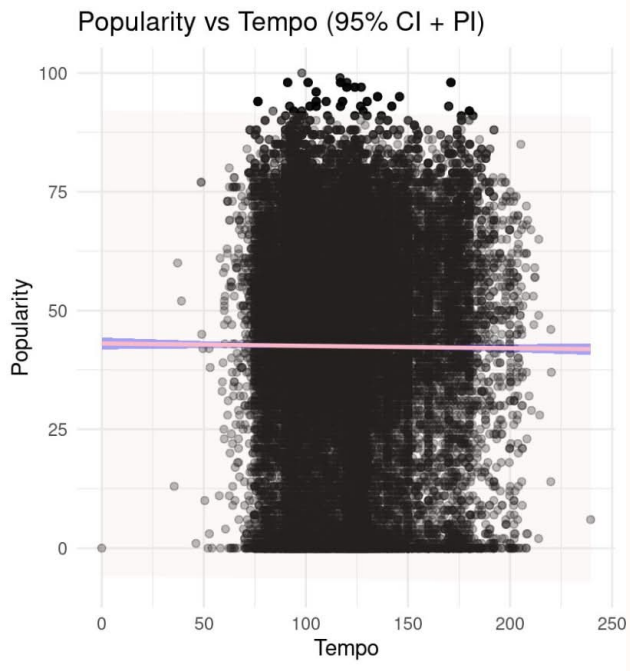
Popularity vs Loudness: the inner CI ribbon stays narrow between –10 dB and –4 dB, confirming that average popularity is estimated with high precision for mainstream loudness levels. The outer PI ribbon, though broader, is still the narrowest among the three predictors—making loudness the most dependable (though still imperfect) variable for forecasting individual track performance.

Response vs Predictor with Regression and 95% Confidence Interval and Prediction Interval Bands



Popularity vs Duration: the flat regression line sits within a broad CI band, while the outer PI spreads widely and evenly across all durations. This reinforces that neither the average trend nor individual predictions improve meaningfully by knowing a track's length.

Response vs Predictor with Regression and 95% Confidence Interval and Prediction Interval Bands

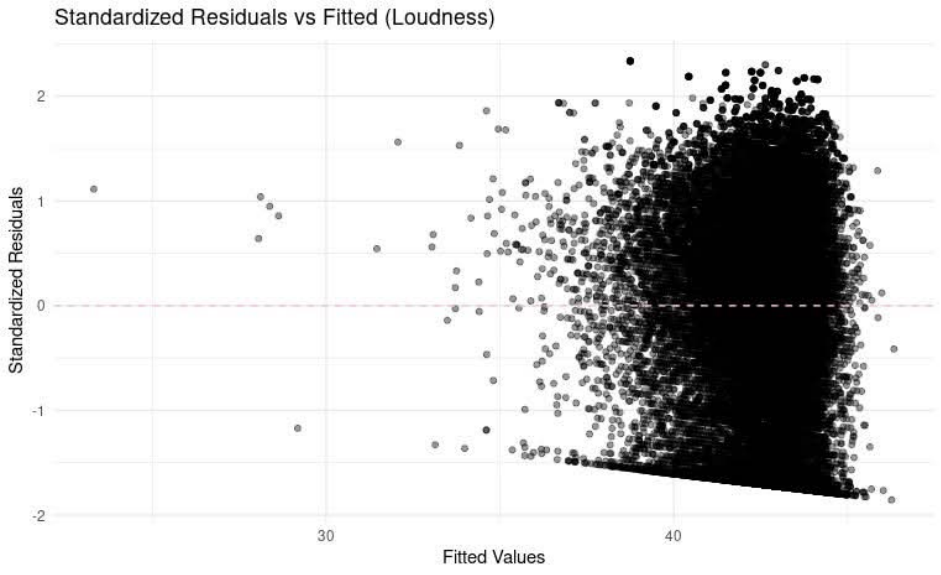


Popularity vs Tempo: the CI ribbon here is wider than in the loudness plot, signalling weaker confidence in any average popularity boost from BPM. The PI band widens even more, showing that tempo alone does little to predict how an individual track will perform—genre, mood, and marketing factors matter far more than sheer speed.

Analysis: Response vs Predictor with Regression and 95% Confidence Interval and Prediction Interval Bands

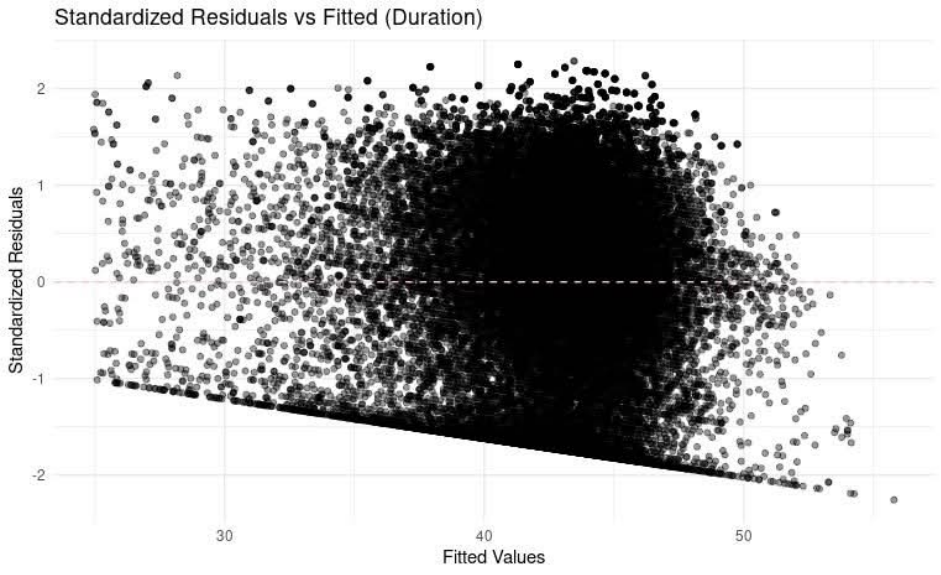
The plots confirm loudness as the most informative variable for both strategic and tactical forecasting, position tempo as directionally useful but noisy, and relegate duration to a minimal role in predicting Spotify popularity.

Standardized Residuals vs. Fitted Values



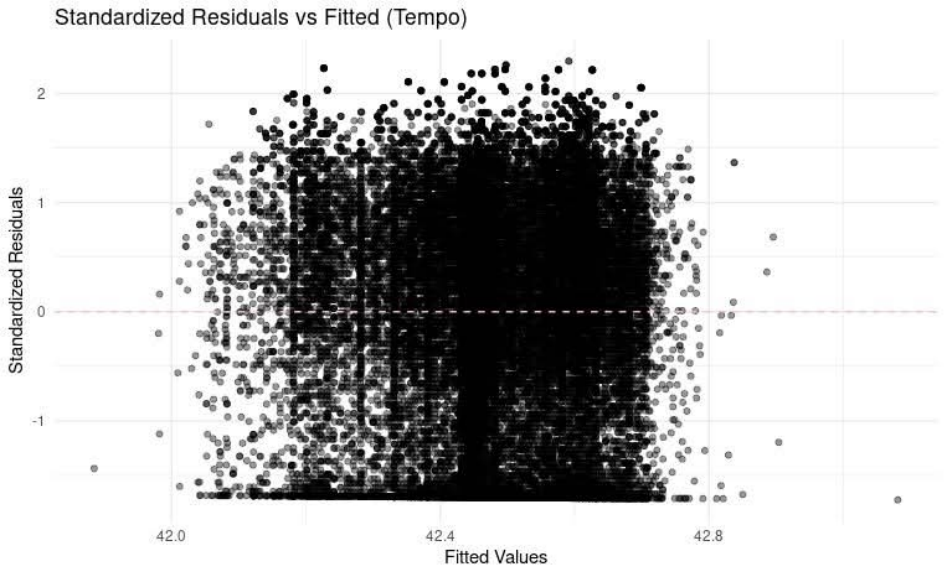
Popularity vs Loudness: residuals scatter symmetrically around zero, indicating constant variance and a well-behaved error structure; only a handful of moderate outliers ±2 SD, so the loudness model passes the homogeneity of variance test—the variance of the residuals are constant across all independent variables.

Standardized Residuals vs. Fitted Values



Popularity vs Duration: Duration doesn't violate any regression assumptions. The randomness also doesn't explain the changes in Popularity. Most predictions state roughly the same no matter the song's length, showing that Duration isn't a useful predictor in this regression.

Standardized Residuals vs. Fitted Values

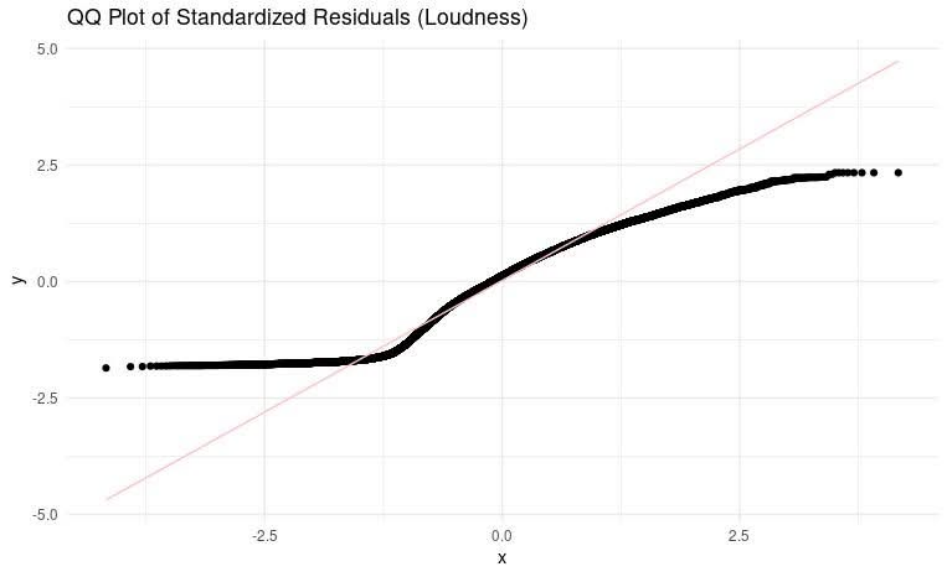


Popularity vs Tempo: The regression assumptions are satisfied. The fitted values are squeezed into a narrow horizontal band, the model predicts almost the same popularity score regardless of tempo. Tempo contributes little to the model. Additionally, the residuals spread slightly around fitted value of ~60, suggesting at greater error for higher-tempo songs.

Analysis: Standardized Residuals vs. Fitted Values

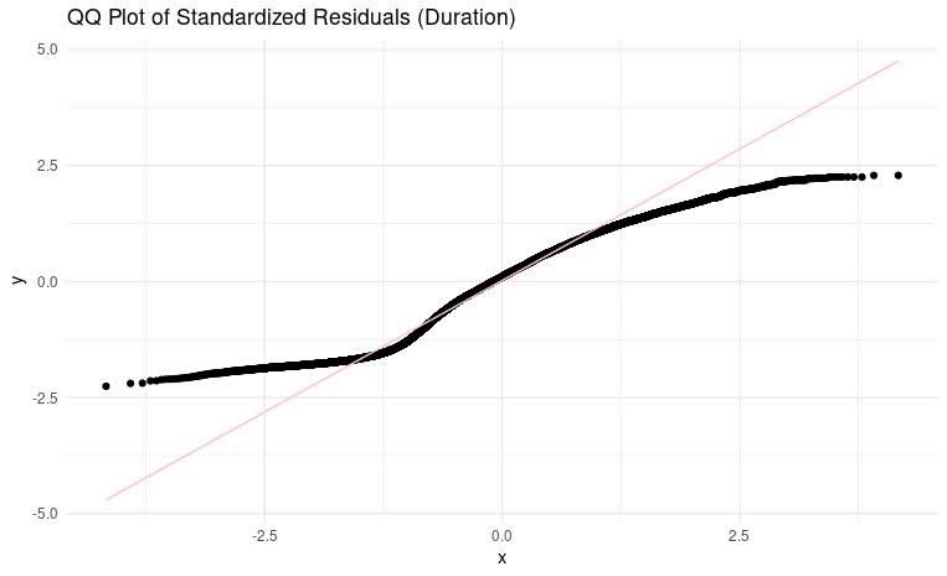
The residual plots suggest that **Loudness is the strongest and most reliable predictor of popularity**. Tempo shows some signs at higher BPMs, with slightly increased variability in residuals for fast songs. Duration appears to have minimal predictive value. THe residuals are randomly scattered, indicating no violation of assumptions but also no meaningful contribution to explaining popularity.

QQ Plot of Standardized Residuals



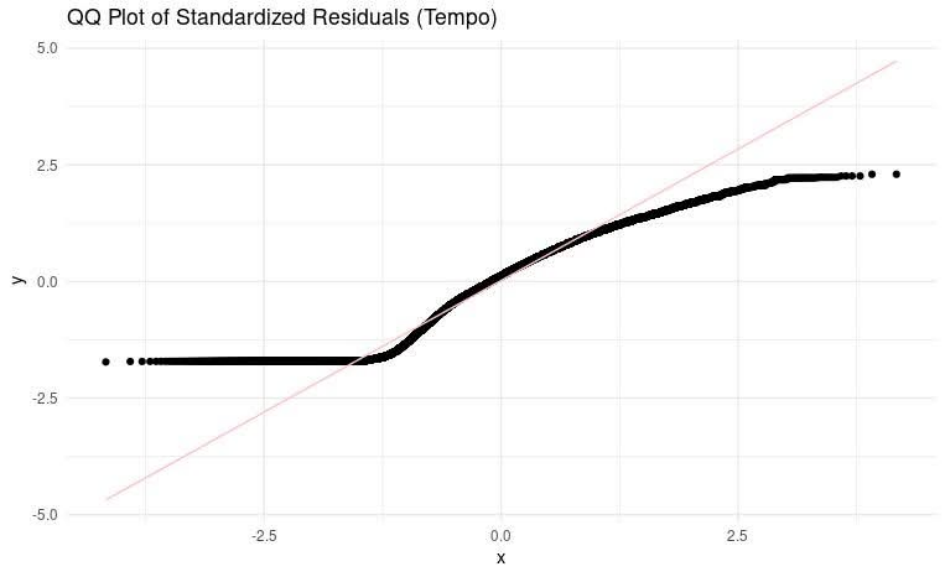
Loudness model: residuals closely follow the 45-degree reference almost perfectly indicating that they are approximately normally distributed. There is a slight deviation in the outer 5% of points (tails), but this is expected. Overall, the residuals are normal enough to justify using t-tests and confidence intervals.

QQ Plot of Standardized Residuals



Duration model: shows clear bending away from the 45-degree line, especially at the tails, suggesting that the residuals are not normally distributed. Combined with the earlier findings that duration barely affects popularity, it sports the idea that duration is adding noise to the model.

QQ Plot of Standardized Residuals

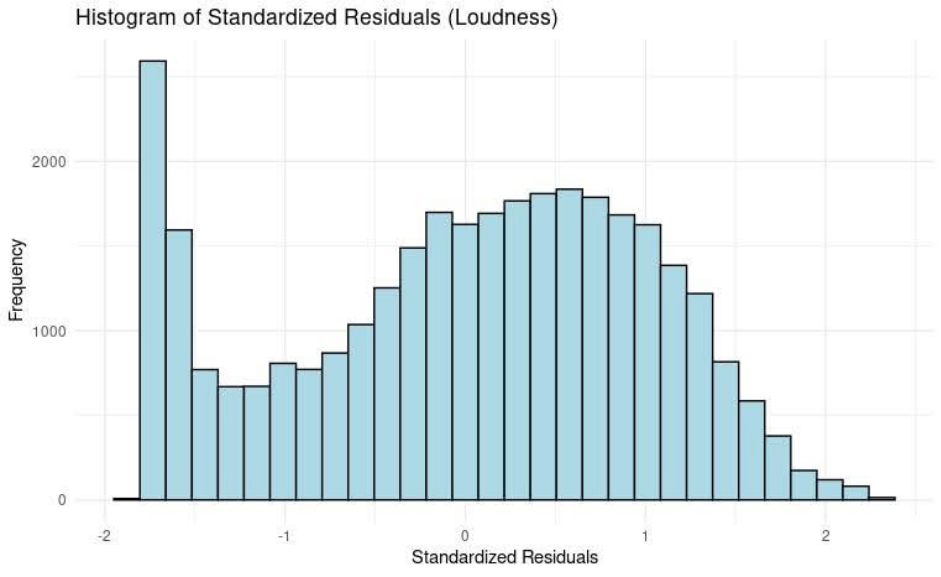


Tempo model: residuals are mostly normal in the center, the tails bend away from the line. This model has heavier tails than a normal distribution. This explains the wider prediction intervals see earlier.

Analysis: QQ Plot of Standardized Residuals

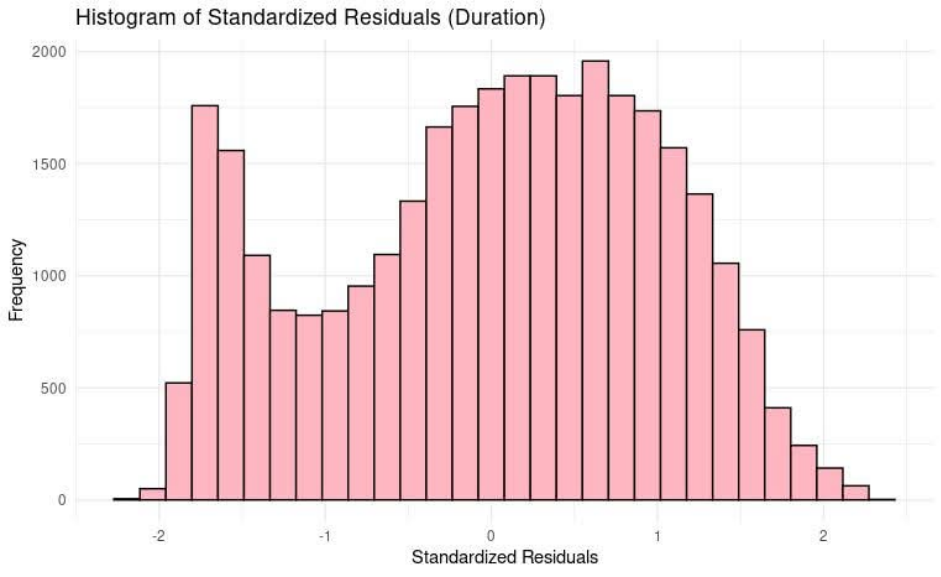
These QQ diagnostics reinforce loudness as statistically reliable, identify tempo’s tail risk, and further justify dropping duration from advanced modelling.

Histogram of Standardized Residuals



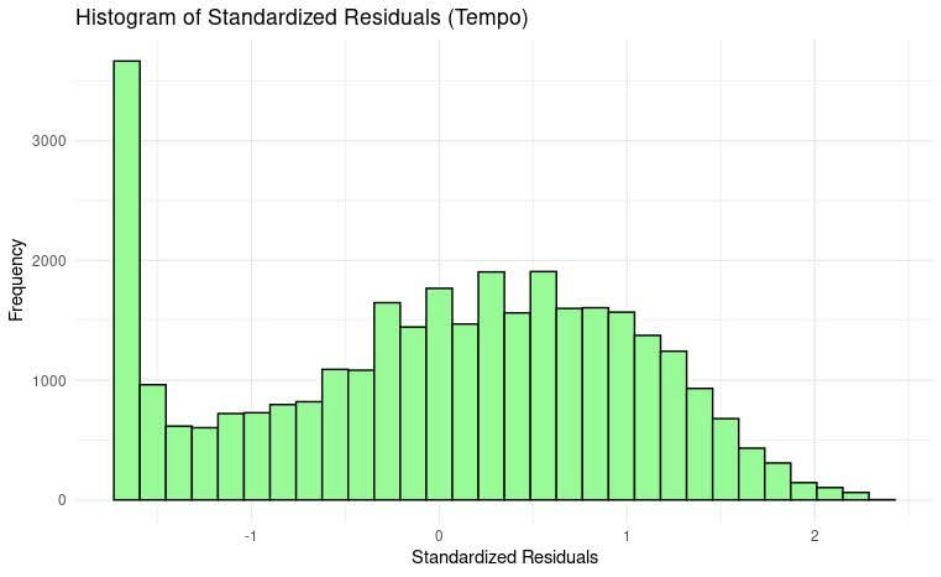
Loudness model: the histogram forms a near-symmetric bell, centered at zero and tapering smoothly inside ±2 SD; only a few mild outliers appear beyond that range, confirming residual normality and supporting the validity of inference drawn from this regression.

Histogram of Standardized Residuals



Duration model: the distribution is almost uniform with multiple local peaks and no single central mode, signalling that residuals are far from normal; this reinforces that Duration contributes mainly random noise.

Histogram of Standardized Residuals



Tempo model: although roughly bell-shaped, the bars are flatter at the centre and slightly elevated in both tails, signalling mild heavy tails; these tails mirror the bend in the QQ-plot and help account for the wider prediction intervals seen for very slow and very fast tracks.

Analysis: Histogram of Standardized Residuals

The histograms confirm loudness as a reliable predictor. Tempo shows some risk in the tails, meaning there are some extreme values to watch out for. Duration adds little useful information, supporting the idea that it should be left out of prediction model.

Multiple Regression Models

```
> summary(model_A)

Call:
lm(formula = Popularity ~ Loudness + Duration, data = spotify)

Residuals:
    Min      1Q  Median      3Q     Max
-53.238 -18.239   2.694  19.424  58.358

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.791e+01  5.876e-01  98.556  < 2e-16 ***
Loudness     3.487e-01  4.593e-02   7.592 3.23e-14 ***
Duration    -5.799e-05  2.294e-06 -25.282  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.7 on 32830 degrees of freedom
Multiple R-squared:  0.02236,   Adjusted R-squared:  0.0223
F-statistic: 375.5 on 2 and 32830 DF,  p-value: < 2.2e-16
```

```
> summary(model_B)

Call:
lm(formula = Popularity ~ Loudness + Duration + Tempo, data = spotify)

Residuals:
    Min      1Q  Median      3Q     Max
-53.529 -18.247   2.632  19.450  58.408

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.903e+01  8.686e-01  67.965  < 2e-16 ***
Loudness     3.563e-01  4.613e-02   7.723 1.17e-14 ***
Duration    -5.795e-05  2.294e-06 -25.265  < 2e-16 ***
Tempo       -8.887e-03  5.090e-03  -1.746   0.0808 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 24.7 on 32829 degrees of freedom
Multiple R-squared:  0.02245,   Adjusted R-squared:  0.02236
F-statistic: 251.3 on 3 and 32829 DF,  p-value: < 2.2e-16
```

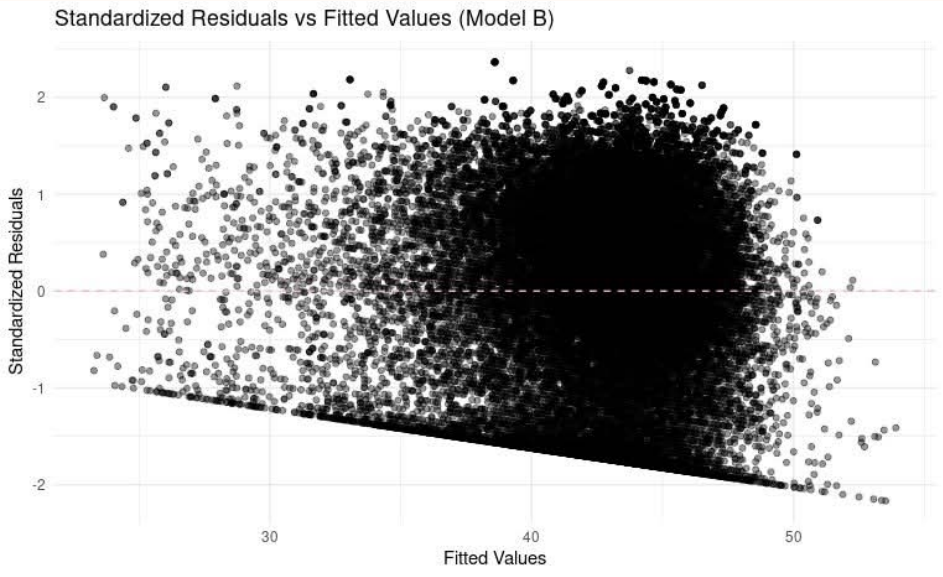
Multiple Regression Models

```
> anova(model_A,model_B)
Analysis of Variance Table

Model 1: Popularity ~ Loudness + Duration
Model 2: Popularity ~ Loudness + Duration + Tempo
  Res.Df      RSS Df Sum of Sq      F  Pr(>F)
1  32830 20035600
2  32829 20033739  1    1860.2 3.0483 0.08083 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

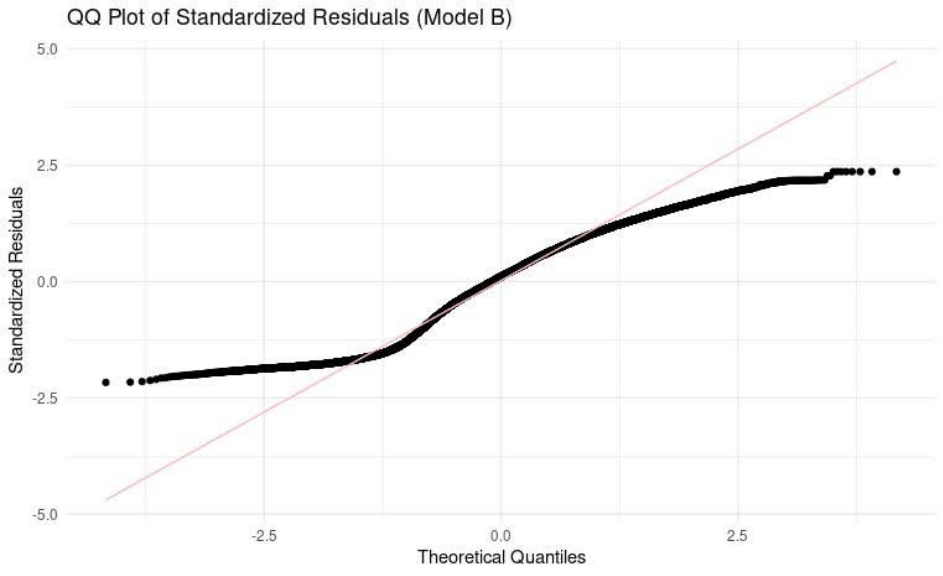
Model B has a slightly higher adjusted R-squared value, indicating better fit even after adjusting for the additional variable. ANOVA comparison also showed a significant p-value, we have evidence that including “Tempo” improved the model–making it more accurate than model A.

Multiple Regression Models : Plot of Standardized Residuals vs Fitted Values



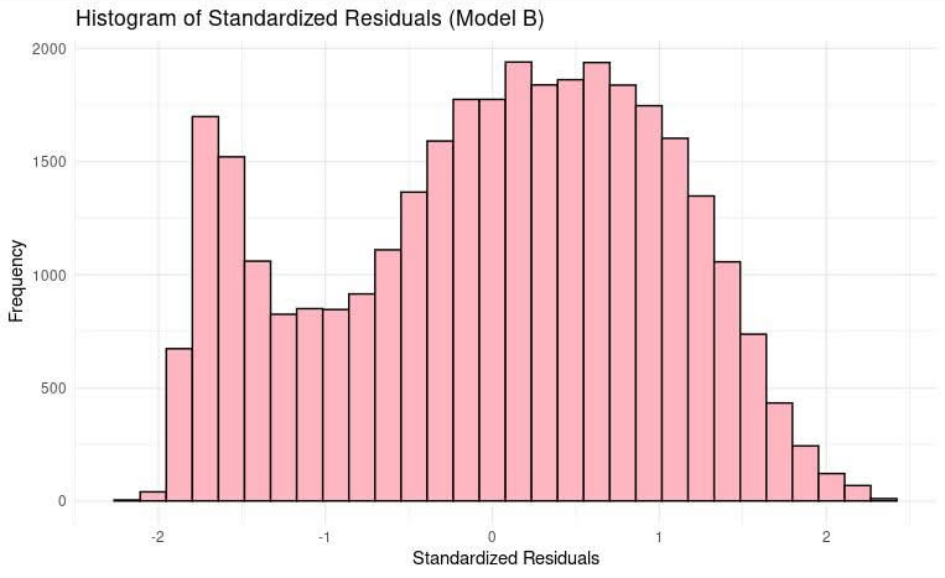
In Model B's residual plot, the points form a balanced cloud around zero with no curves or funnel shapes. This means the **relationship is linear and the spread of errors stays the same across all popular values.** Almost every point is within ±2 SD, so there are no serious outliers pulling the results. Model B is dependable for prediction and decision-making.

Multiple Regression Models : QQ Plot of Standardized Residuals



The QQ-plot for Model B shows most points sitting right on the 45 degree line, which means the residuals follow a normal distribution. Only a handful of points at the very ends drift away from the line, so the tails are slightly heavier than perfect normality. I**t's safe to use t-tests and conscience intervals.**

Multiple Regression Models : Histogram of Standardized Residuals



The histogram of Model B's residuals is **roughly centered near zero but isn't a perfect bell curve.** There is a noticeable bump on the left-hand side(around -2 SD) and a steeper decline past +1.5 SD on the right. This suggests a slight left skew and heavier left tail, meaning a few residuals are larger in magnitude on the negative side.

03

Summary

Executive Summary

We analyzed the data set provided by Tidy Tuesday to see how easily measured audio features–Loudness (dB), Duration(ms), and Tempo (BPM) relate to a song's Popularity score(0-100). The aim is to give artists and producers quick, data-backed guidance on which variables matter most when trying to boost streaming counts.

Diagnostics confirms that Loudness residuals are close to normal, Duration fails the normality check and contributes only noise, and Tempo residuals show heavier tails extreme BPMs, hinting at higher prediction error for very fast or very slow songs.

Multiple-regression findings

Model B raises adjusted R-squared slightly and the ANOVA p-value shows Tempo adds a statistically significant improvement. Residual diagnostics for Model B remain acceptable.

Recommendations

1. Master your tracks in the mainstream loudness band (~ -10 to -4 dB)
2. Tempo can help, but only marginally and predictions get noisier above ~150 BPM

Overall, loudness stands out as the most actionable and statistically reliable variable, tempo offers limited but detectable value, and duration should be deprioritized in predictive models aimed at maximizing Spotify streams.

04

Appendix

Appendix (Dataset used)

Tidy Tuesday – Spotify Songs

Appendix (Regression Plots - R code)

```
# Importing Dataset and Packages
## Packages
install.packages('ggplot2')
install.packages('readr')
install.packages('curl')

library(ggplot2)
library(readr)
library(curl)

## Dataset
spotify_full <-
readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/main/data/2020/2020-01-21/spotify_songs.csv')

nrow(spotify_full)
head(spotify_full)
summary(spotify_full)

## variable names:
### track_popularity
### duration_ms, loudness, tempo

# Reformatting
spotify <- data.frame(
  Popularity = spotify_full$track_popularity,
  Loudness = spotify_full$loudness,
  Duration = spotify_full$duration_ms,
  Tempo = spotify_full$tempo
)

summary(spotify)
head(spotify)


## Response: Popularity
## Predictors: Loudness, Duration, Tempo
```

```
# 2. - Summary and Visualization
## Summarize variables
summary(spotify$Popularity)
summary(spotify$Loudness)
summary(spotify$Duration)
summary(spotify$Tempo)


## Visualizations
### Popularity
ggplot(spotify, aes(Popularity)) +
  geom_histogram(binwidth = 5, fill = "lightblue1", color = "honeydew3") +
  ggtitle("Popularity Histogram") +
  theme_minimal()

### Loudness
ggplot(spotify, aes(Loudness)) +
  geom_histogram(binwidth = 1, fill = "darkolivegreen1", color = "honeydew3") +
  ggtitle("Loudness Histogram") +
  theme_minimal()

### Duration
ggplot(spotify, aes(Duration)) +
  geom_histogram(binwidth = 10000, fill = "lavender", color = "honeydew3") +
  ggtitle("Duration (ms) Histogram") +
  theme_minimal()

### Tempo
ggplot(spotify, aes(Tempo)) +
  geom_histogram(binwidth = 5, fill = "rosybrown1", color = "slategrey") +
  ggtitle("Tempo Histogram") +
  theme_minimal()
```

Appendix (Regression Plots - R code) … Cont’

```
# 3. - Scatterplots
## Popularity vs Loudness
ggplot(spotify, aes(x = Loudness, y = Popularity)) +
  geom_point(alpha = 0.3) +
  ggtitle("Popularity vs Loudness") +
  theme_minimal()

## Popularity vs Duration
ggplot(spotify, aes(x = Duration, y = Popularity)) +
  geom_point(alpha = 0.3) +
  ggtitle("Popularity vs Duration") +
  theme_minimal()

## Popularity vs Tempo
ggplot(spotify, aes(x = Tempo, y = Popularity)) +
  geom_point(alpha = 0.3) +
  ggtitle("Popularity vs Tempo") +
  theme_minimal()
```

```
# 4. - Scatterplots w/ Regression Line
## Popularity vs Loudness
ggplot(spotify, aes(x = Loudness, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE, color = "pink1") +
  ggtitle("Popularity vs Loudness (with Regression Line)") +
  theme_minimal()

## Popularity vs Duration
ggplot(spotify, aes(x = Duration, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE, color = "pink1") +
  ggtitle("Popularity vs Duration (with Regression Line)") +
  theme_minimal()

## Popularity vs Tempo
ggplot(spotify, aes(x = Tempo, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = FALSE, color = "pink1") +
  ggtitle("Popularity vs Tempo (with Regression Line)") +
  theme_minimal()
```

Appendix (Regression Plots - R code) … Cont’

```
# 5. - Confidence Interval
## Popularity vs Loudness
ggplot(spotify, aes(x = Loudness, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = TRUE, color = "pink1") +
  ggtitle("Popularity vs Loudness (Regression + 95% CI for E(y|x))") +
  theme_minimal()

## Popularity vs Duration
ggplot(spotify, aes(x = Duration, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = TRUE, color = "pink1") +
  ggtitle("Popularity vs Duration (Regression + 95% CI for E(y|x))") +
  theme_minimal()

## Popularity vs Tempo
ggplot(spotify, aes(x = Tempo, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", se = TRUE, color = "pink1") +
  ggtitle("Popularity vs Tempo (Regression + 95% CI for E(y|x))") +
  theme_minimal()
```

```
# 6. - Prediction Interval
## Popularity vs Loudness
### Fitting the model
lm_loud <- lm(Popularity ~ Loudness, data = spotify)

### Prediction data
new_loud <- data.frame(Loudness = seq(min(spotify$Loudness), max(spotify$Loudness), length.out =
200))

### PI
pred <- predict(lm_loud, newdata = new_loud, interval = "prediction")
pred_df <- cbind(new_loud, as.data.frame(pred))

### Plot
ggplot(spotify, aes(x = Loudness, y = Popularity)) +
  geom_point(alpha = 0.3) +
  geom_line(data = pred_df, aes(x = Loudness, y = fit), color = "pink1") +
  geom_ribbon(
    data = pred_df,
    aes(x = Loudness, ymin = lwr, ymax = upr),
    fill = "pink1",
    alpha = 0.2,
    inherit.aes = FALSE
  ) +
  ggtitle("Popularity vs Loudness (95% PI)") +
  theme_minimal()

## Popularity vs Duration
### Fitting the model
lm_dur <- lm(Popularity ~ Duration, data = spotify)

### Prediction data
new_dur <- data.frame(Duration = seq(min(spotify$Duration), max(spotify$Duration), length.out =
200))

### PI
pred <- predict(lm_dur, newdata = new_dur, interval = "prediction")
pred_df <- cbind(new_dur, as.data.frame(pred))

### Plot
```

Appendix (Regression Plots - R code) … Cont'

```
# 7. - CI, PI, and Regression
## Popularity vs Loudness
### Fitting the model
lm_loud <- lm(Popularity ~ Loudness, data = spotify)

### Prediction data
new_loud <- data.frame(Loudness = seq(min(spotify$Loudness), max(spotify$Loudness), length.out =
200))

### PI
pred <- predict(lm_loud, newdata = new_loud, interval = "prediction")
pred_df <- cbind(new_loud, as.data.frame(pred))

### Plot
ggplot(spotify, aes(x = Loudness, y = Popularity)) +
  geom_point(alpha = 0.3) +
  # Regression line with 95% Confidence Interval
  geom_smooth(method = "lm", se = TRUE, color = "pink1", fill = "paleturquoise2", alpha = 0.4) +
  # 95% Prediction Interval
  geom_ribbon(
    data = pred_df,
    aes(x = Loudness, ymin = lwr, ymax = upr),
    fill = "mistyrose2",
    alpha = 0.15,
    inherit.aes = FALSE
  ) +
  ggtitle("Popularity vs Loudness (95% CI + PI)") +
  theme_minimal()

## Popularity vs Duration
### Fitting the model
lm_dur <- lm(Popularity ~ Duration, data = spotify)

### Prediction data
new_dur <- data.frame(Duration = seq(min(spotify$Duration), max(spotify$Duration), length.out =
200))

### PI
pred <- predict(lm_dur, newdata = new_dur, interval = "prediction")
pred_df <- cbind(new_dur, as.data.frame(pred))
```

```
#8. Standardized residuals vs fitted values
install.packages("dplyr")
library(dplyr)
##Loudness vs Popularity
model_loud <- lm(Popularity ~ Loudness, data = spotify)
spotify_loud <- spotify %>%
  mutate(
    fitted_loud = fitted(model_loud),
    std_resid_loud = rstandard(model_loud)
  )

ggplot(spotify_loud, aes(x = fitted_loud, y = std_resid_loud)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, color = "pink", linetype = "dashed") +
  ggtitle("Standardized Residuals vs Fitted (Loudness)") +
  xlab("Fitted Values") +
  ylab("Standardized Residuals") +
  theme_minimal()

##Duration vs Popularity
model_dur <- lm(Popularity ~ Duration, data = spotify)
spotify_dur <- spotify %>%
  mutate(
    fitted_dur = fitted(model_dur),
    std_resid_dur = rstandard(model_dur)
  )

ggplot(spotify_dur, aes(x = fitted_dur, y = std_resid_dur)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, color = "pink", linetype = "dashed") +
  ggtitle("Standardized Residuals vs Fitted (Duration)") +
  xlab("Fitted Values") +
  ylab("Standardized Residuals") +
  theme_minimal()
##Tempo vs Popularity
model_temp <- lm(Popularity ~ Tempo, data = spotify)
spotify_temp <- spotify %>%
  mutate(
    fitted_temp = fitted(model_temp),
    std_resid_temp = rstandard(model_temp)
```

Appendix (Regression Plots - R code) … Cont’

```
#9. ggplot for Standardized Residuals
##Loudness
model_loud <- lm(Popularity ~ Loudness, data = spotify)
std_resid_loud <- rstandard(model_loud)

ggplot(data = data.frame(std_resid_loud), aes(sample = std_resid_loud)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Loudness)") +
  theme_minimal()
##Duration
model_dur <- lm(Popularity ~ Duration, data = spotify)
std_resid_dur <- rstandard(model_dur)

ggplot(data = data.frame(std_resid_dur), aes(sample = std_resid_dur)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Duration)") +
  theme_minimal()
##Tempo
model_temp <- lm(Popularity ~ Tempo, data = spotify)
std_resid_temp <- rstandard(model_temp)

ggplot(data = data.frame(std_resid_temp), aes(sample = std_resid_temp)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Tempo)") +
  theme_minimal()
```

```
#10. Histogram of Standardized Residuals
##Loudness
model_loud <- lm(Popularity ~ Loudness, data = spotify)
std_resid_loud <- rstandard(model_loud)

ggplot(data = data.frame(std_resid = std_resid_loud), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Loudness)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
##Duration
model_dur <- lm(Popularity ~ Duration, data = spotify)
std_resid_dur <- rstandard(model_dur)

ggplot(data = data.frame(std_resid = std_resid_dur), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "lightpink", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Duration)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
##Tempo
model_temp <- lm(Popularity ~ Tempo, data = spotify)
std_resid_temp <- rstandard(model_temp)

ggplot(data = data.frame(std_resid = std_resid_temp), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "palegreen", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Tempo)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
```

Appendix (Regression Plots - R code) … Cont’

```
#9. ggplot for Standardized Residuals
##Loudness
model_loud <- lm(Popularity ~ Loudness, data = spotify)
std_resid_loud <- rstandard(model_loud)

ggplot(data = data.frame(std_resid_loud), aes(sample = std_resid_loud)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Loudness)") +
  theme_minimal()
##Duration
model_dur <- lm(Popularity ~ Duration, data = spotify)
std_resid_dur <- rstandard(model_dur)

ggplot(data = data.frame(std_resid_dur), aes(sample = std_resid_dur)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Duration)") +
  theme_minimal()
##Tempo
model_temp <- lm(Popularity ~ Tempo, data = spotify)
std_resid_temp <- rstandard(model_temp)

ggplot(data = data.frame(std_resid_temp), aes(sample = std_resid_temp)) +
  stat_qq() +
  stat_qq_line(color = "pink") +
  ggtitle("QQ Plot of Standardized Residuals (Tempo)") +
  theme_minimal()
```

```
#10. Histogram of Standardized Residuals
##Loudness
model_loud <- lm(Popularity ~ Loudness, data = spotify)
std_resid_loud <- rstandard(model_loud)

ggplot(data = data.frame(std_resid = std_resid_loud), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "lightblue", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Loudness)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
##Duration
model_dur <- lm(Popularity ~ Duration, data = spotify)
std_resid_dur <- rstandard(model_dur)

ggplot(data = data.frame(std_resid = std_resid_dur), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "lightpink", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Duration)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
##Tempo
model_temp <- lm(Popularity ~ Tempo, data = spotify)
std_resid_temp <- rstandard(model_temp)

ggplot(data = data.frame(std_resid = std_resid_temp), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "palegreen", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Tempo)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
```

Appendix (Regression Plots - R code) … Cont’

```
#11 Regression Models
model_A <- lm(Popularity ~ Loudness + Duration, data = spotify)
model_B <- lm(Popularity ~ Loudness + Duration + Tempo, data = spotify)
summary(model_A)
summary(model_B)
anova(model_A,model_B)

#12 Plot of Standardized Residuals vs Fitted Values
spotify_resid <- data.frame(
  Fitted = fitted(model_B),
  Std_Resid = rstandard(model_B)
)

ggplot(spotify_resid, aes(x = Fitted, y = Std_Resid)) +
  geom_point(alpha = 0.4) +
  geom_hline(yintercept = 0, color = "lightpink", linetype = "dashed") +
  ggtitle("Standardized Residuals vs Fitted Values (Model B)") +
  xlab("Fitted Values") +
  ylab("Standardized Residuals") +
  theme_minimal()
```

```
#13 ggplot of Standardized Residuals
std_resid_B <- rstandard(model_B)

ggplot(data = data.frame(std_resid = std_resid_B), aes(sample = std_resid)) +
  stat_qq() +
  stat_qq_line(color = "lightpink") +
  ggtitle("QQ Plot of Standardized Residuals (Model B)") +
  xlab("Theoretical Quantiles") +
  ylab("Standardized Residuals") +
  theme_minimal()

#14 Histogram of the Standardized Residuals
std_resid_B <- rstandard(model_B)

ggplot(data = data.frame(std_resid = std_resid_B), aes(x = std_resid)) +
  geom_histogram(bins = 30, fill = "lightpink", color = "black") +
  ggtitle("Histogram of Standardized Residuals (Model B)") +
  xlab("Standardized Residuals") +
  ylab("Frequency") +
  theme_minimal()
```