Predicting Streams from Number of Playlists

Fiona Zhang, Luke Qin

This study examines the predictability of song streaming numbers on platforms like Spotify and Apple Music, focusing on the influence of playlist inclusions and intrinsic song characteristics. Using the "Spotify Music" dataset, linear regression models were compared to ascertain the best predictors of streaming success. Initial models included playlist counts and variables such as danceability and energy, while subsequent models refined the predictor selection based on statistical significance and predictive performance, indicated by BIC, Adjusted R-Squared and AIC values. The final model, which considered the number of playlist inclusions and selected song characteristics (energy, valence, acousticness, instrumentalness, and speechiness), demonstrated improved predictability. The research highlights the critical role of strategic playlist placement and specific musical qualities in enhancing a song's streaming potential, offering actionable insights for targeted marketing strategies in the music industry.

Background and Significance

This research investigates the relationship between how often songs appear on playlists on popular music streaming services like Spotify and Apple Music, and their overall streaming success. We're particularly interested in how being featured in playlists affects a song's popularity and if specific song characteristics, like how energetic or danceable a song is, can predict streaming success even better. The data for this study comes from the "Spotify Music" dataset on Kaggle, a platform where data scientists and researchers share data. This dataset includes details on how frequently songs are included in playlists and their musical attributes.

The importance of this study lies in the huge influence that digital streaming services have on the music industry today. These platforms have changed how people access music and have shaped new ways for artists and record labels to earn money. In today's highly competitive music industry, it's essential to understand what factors affect streaming numbers. This knowledge can help maximize a song's visibility and profitability.

By analyzing the effects of playlist appearances and combining this with the songs' inherent characteristics, this study aims to provide valuable insights for the music industry. These insights could help shape marketing strategies and production choices, potentially leading to more effective promotions and a deeper understanding of what makes a song popular on streaming services.Methods

Method

We downloaded our data set from Kaggle, the name of the data set is "Top Spotify Songs". The dataset is well organized and do not have any missing data. It consists the observations of individual tracks, where each track is described by the variables: track_name, artists_name, artist_cont, released_year, released_day, is_spotify_playlists, in_spotify_charts, streams, in_apple_playlists, in_apple_charts, in_deezer_playlists, in_deezer_charts, in_shazam_charts, bpm, key, mode, dancebility, valence, energy, acousticness, instrumentalness, liveness, and speechiess. These are all numeric data.

Since our research topic would be about How do number of playlists on Spotify and Apple Music platforms help predicting the streaming numbers of popular songs, we only consider in_spotify_playlists, in_apple_playlists, danceability, energy, liveness, valence, acousticness, instrumentalness, and speechiness these nine variables are relevant, and we will process our tests based on only these nine variables.

To start up, since this is a large dataset, we have to select the variables we needed from the dataset using R and change the scale of stream number so that the results are more accessible and clearer to analyze. We divided the stream number by 1000, which means the number of streams' scale is by a thousand at this point. After changing the scale, one missing data in variable "streams" is found and we deal with it by the drop_na() function in R to simply delete this missing data from the dataset.

We have three model in total, for our base model, it assesses the relationship between the number of streams a song gets and its appearances in Spotify and Apple playlists. For our mod1, which is a full model, extends the base model by including additional song features such as danceability, energy, liveness, valence, acousticness, instrumentalness, and speechiness. This model evaluates which song characteristics, alongside playlist appearances, significantly affect streaming numbers. And out final model refines mod1 by removing statistically insignificant variables, aiming to improve the model's accuracy and simplicity. We evaluate this through comparing the BIC and each variables included in mod1. Additionally, we check and compare each model through their BIC, AIC and Adjusted R-squared. Higher Adjusted R-squared and smaller AIC and BIC means the model has better prediction performance.

Since we are using multiple linear regression model, we are checking the assumptions for the multiple linear regression, which they are zero mean, linearity, normality, constant variance, independence and random. Zero mean is sure met, and for independence and random, we believe the data set from Kaggle can met these two. We will check linearity and constant variance by a residual vs. fits plot, and check normality by a QQ-plot.

Additionally, we want to make our model better through model transformation. We take the log and squared transformations of the final model we picked and compared the Residual plot to see if the transformations are making the model more linear.

Results

Table 1: Statistical Summary of all Variables included in the Final Model

Characteristic	**N = 952**
in_spotify_playlists	2,217 (875, 5,574)
in_apple_playlists	34 (13, 88)
energy	66 (53, 77)
valence	51 (32, 70)
acousticness	18 (6, 43)
instrumentalness	0 (0, 0)
speechiness	6 (4, 11)

Characteristic	N = 952

Firstly, we concludes a statistical summary of all variables that will be included in our analysis. This includes the in_spotify_playlists and in_apple_playlists, and also all music intrinsic characteristics. Later on, we summarize all the BIC values of these variables to select the variables that are best fit into the prediction of streams.

We plot the BIC comparison plot of model1, it shows the relative performance of different predictors in your model. Lower BIC values indicate a better model fit, given a trade-off between model complexity and goodness of fit. From the plot: Predictors like "in_spotify_playlists" and "in_apple_playlists" show lower BIC values, suggesting they are strong predictors for streaming numbers.

Predictors such as "danceability," and "liveness," has higher BIC values, indicating they may not significantly improve the model compared to their cost in additional complexity.

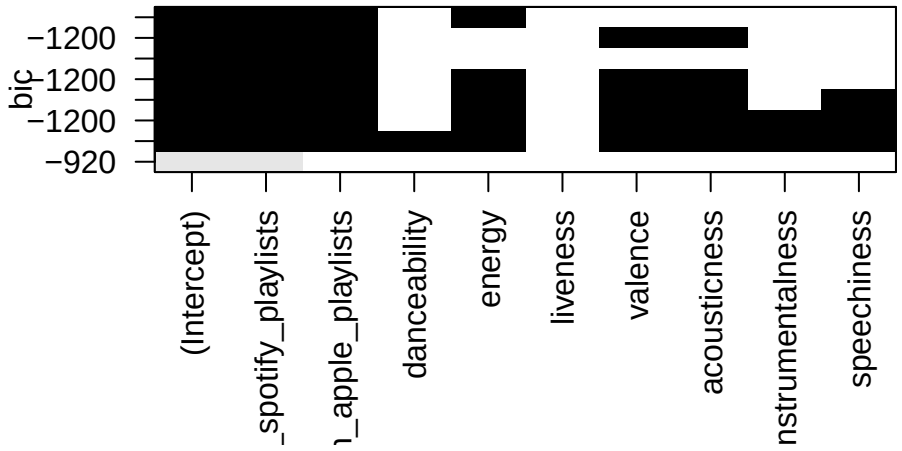

Figure 1: The BIC comparison of all variables in the model 1

The presence of several predictors with substantially varying BIC values suggests that some features are more predictive than others. Thus removing less informative predictors could simplify the model without a significant loss in performance. Then we build a

refined mod1 by removing "danceability," and "liveness," these two variables, and we call it final model.

To further confirm that the final model is indeed better, we calculate the AIC BIC and Adjusted R-squared for model 1 and final model. We can see that the AIC and BIC for the final model is smaller, and its adjusted R-squared is larger, which they tell us the same thing that our best model should be the final model.

Table 2: Comparison of Adjusted R-squared, AIC and BIC values for final model and model 1

Model	Adjusted_R2	AIC	BIC
Model 1	0.7187370	26729.96	26783.40
Final Model	0.7192285	26726.31	26770.04

Here is the equation for the final model:

$$\text{streams} = 260635.526 + 34.666 \times \text{in\_spotify\_playlists} + 2836.985 \times \text{in\_apple\_playlists} - 1112.322 \times \text{energy} - 1026.256 \times \text{valence} + 691.991 \times \text{acousticness} - 1108.651 \times \text{instrumentalness} - 1195.325 \times \text{speechiness}$$

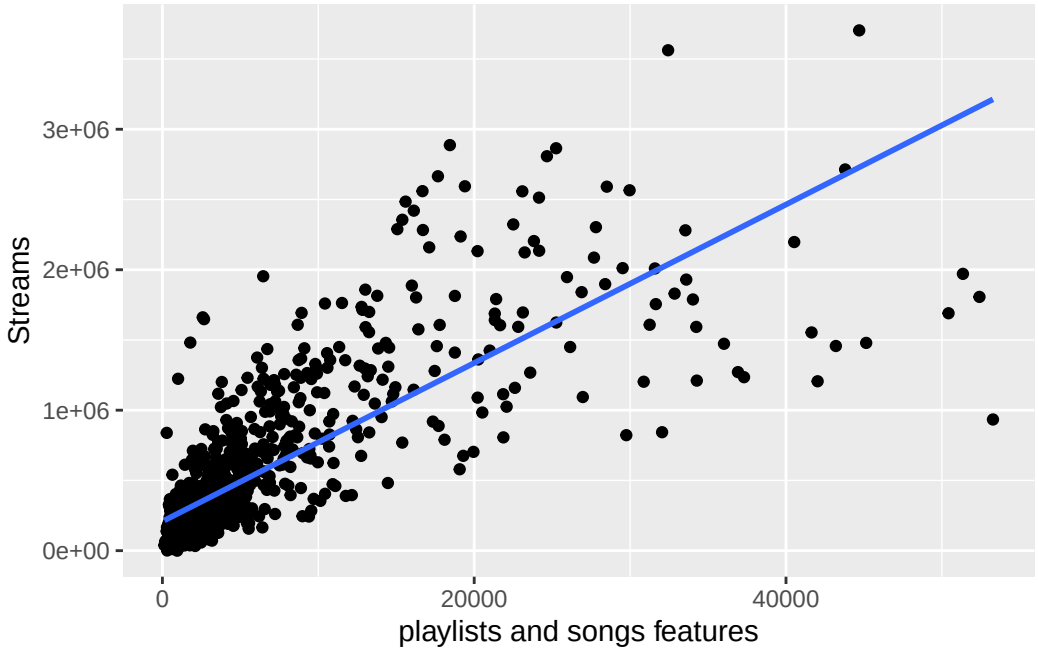

Figure 2: Visualization of the Final Model

Above is the visualization of our final model. From the plot, we can see most of the data fits into the linear model that we utilize, which means we do get a pretty good prediction of streams from the variables we chosen. We also include a table of variables included inside this final models which contains their beta, confidence interval and also the p-value.

Table 3: Prediction of Streams for Final Model

Characteristic	**Beta**	**95% CI**	**p-value**
in_spotify_playlists	35	31, 38	<0.001
in_apple_playlists	2,837	2,521, 3,153	<0.001
energy	-1,112	-2,644, 419	0.2
valence	-1,026	-1,923, -130	0.025
acousticness	692	-224, 1,608	0.14
instrumentalness	-1,109	-3,414, 1,197	0.3
speechiness	-1,195	-3,146, 755	0.2

Based on this table, the analysis of the linear regression model provides insights into how different variables influence the number of streams a song receives. The model uses variables such as the song's inclusion in Spotify and Apple Music playlists, along with its musical characteristics like energy, valence, acousticness, instrumentalness, and speechiness.

The results indicate a significant relationship between playlist inclusion and streaming numbers. Specifically, each additional placement in a Spotify playlist is associated with an increase of approximately 35 streams per song. More notably, inclusion in an Apple Music playlist has a far more substantial impact, with each addition estimated to increase streams by about 2,837. This stark difference underscores the greater efficacy of Apple Music playlists in enhancing song popularity compared to Spotify.Musical characteristics also play a crucial role but with varying effects, based on this graph.

Overall, the model highlights the importance of playlist inclusion, particularly on Apple Music, as a significant driver of song popularity. It also reveals complex relationships between musical characteristics and streams, suggesting that while certain traits like valence and instrumentalness negatively influence streams, others like acousticness may contribute positively, albeit with some uncertainty. This analysis can guide artists and producers in making strategic decisions about song promotion and the musical features that might resonate best with audiences.

To check the final model fits the assumptions, we first check the linearity and constant variance by making a residual vs. fits plot.

In our analysis to check the linearity assumption necessary for multiple linear regression, we conducted residual versus fitted value plots. These plots did not show a satisfactory linear pattern, prompting tests with logarithmic and square root transformations of the dependent variable (streams). However, these transformations did not yield improvements in the linearity of the residuals. Consequently, the decision was made to proceed with the original model specifications, as the transformed models did not offer a better fit or clearer interpretation of the data.

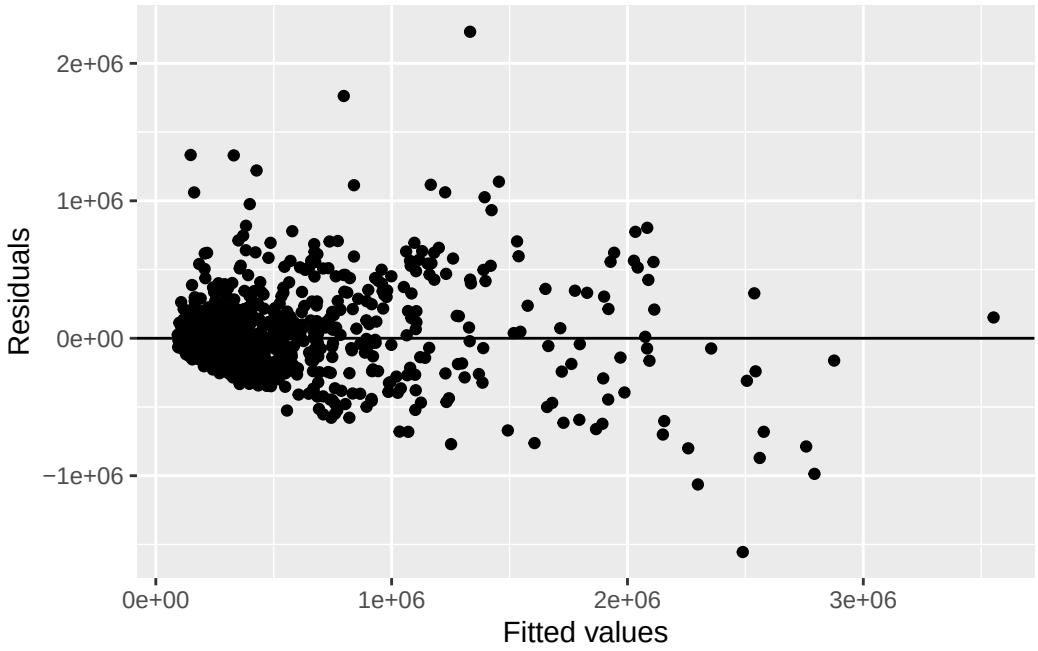

Figure 3: The Residual Plot for Final Model

The Q-Q plot is used to check the normality of residuals in your regression model. In the plot, residuals are plotted against the theoretical normal quantiles. The closer the points lie to the blue line, the more normally distributed the residuals are.

From this plot, we can observe that while the middle quantiles of the residuals align reasonably well with the line (indicating normal distribution), there are deviations at the two tails. The lower tail has some points falling well below the line, and the upper tail shows a pattern of points curving away from the line, which suggests heavy tails and potential outliers. Although pattern indicates that the residuals do not completely follow a normal distribution, we still have enough observations lies in a normal distribution.

The final model for predicting song streams integrates playlist appearances and intrinsic song characteristics, analyzed using a dataset of 952 songs. Inclusion in Spotify and Apple Music playlists significantly increases streams, with Apple playlists showing a

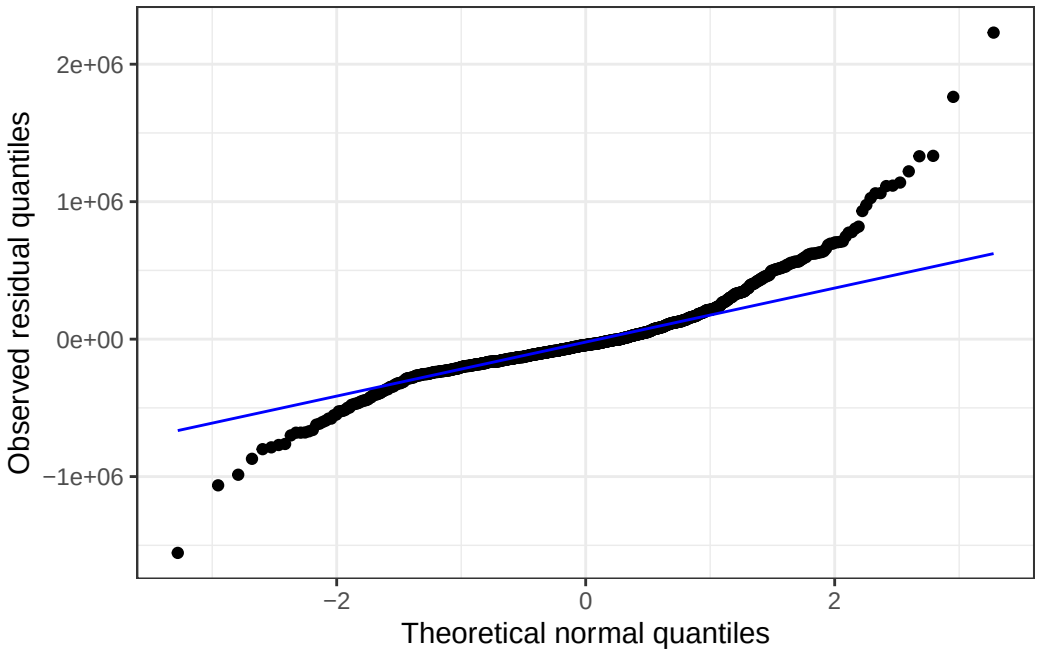

Figure 4: The QQ for Final Model

stronger impact. While intrinsic features like energy and speechiness negatively correlate with streams, their impact is statistically insignificant. Valence shows a negative but significant effect. The model, however, does not adequately conform to normality assumptions, as indicated by the Q-Q plot, which could affect the reliability of the inference drawn from this model. This analysis provides valuable insights into factors influencing streaming success, aiding in targeted marketing and production strategies within the music industry.

Discussion

The analysis conducted through multiple linear regression models has provided substantial insights into the dynamics between playlist inclusions on Spotify and Apple Music and the streaming numbers of popular songs. By incorporating intrinsic song characteristics such as energy, valence, acousticness, instrumentalness, and speechiness, the models effectively answered the research question, illustrating a quantifiable correlation between playlist appearances and streaming success.

From the models, it was discerned that both the inclusion in Spotify and Apple Music playlists and certain song characteristics are significant predictors of streaming numbers. The model refinement process highlighted that features such as energy and acousticness

are not only statistically significant but also practically relevant in enhancing the predictability of a song's streaming performance. The final selected model demonstrated a higher adjusted R-squared value and a lower AIC score compared to initial models, indicating improved explanatory power and predictive accuracy.

However, the research is not without its limitations. The reliance on a dataset solely sourced from Kaggle, while comprehensive, might not capture all nuances of global music consumption patterns. The dataset's scope could potentially limit the generalizability of the findings across different geographic and demographic contexts. Additionally, the assumption that the model's predictors have a linear relationship with the streaming numbers might oversimplify the complexities of consumer behavior in digital music consumption.

Future studies could aim to address these limitations by incorporating a broader dataset that includes a more diverse array of platforms and geographic locations. Research could also explore non-linear models or machine learning techniques to capture more complex relationships within the data. Furthermore, qualitative studies could be employed to understand the psychological and social factors influencing playlist additions and song popularity, thereby enriching the quantitative findings from this study.

In conclusion, this research has laid a foundational understanding of the predictors of song popularity on streaming platforms, which is crucial for shaping targeted marketing and production strategies in the music industry. The insights gained underscore the importance of strategic playlist placements and nuanced understanding of song characteristics, paving the way for future explorations that could further refine these predictive models.

References

Arnav (2024)

Arnav, Kumar. 2024. "Top Spotify Songs." *Kaggle*. https://www.kaggle.com/datasets/arnavvvvv/spotify-music.

Appendix

```
suppressMessages(library(tidyverse))
library(gtsummary)
library(knitr)
library(leaps)
```

```r
Popular_Spotify_Songs <- read.csv("Popular_Spotify_Songs.csv")
Popular_Spotify_Songs <- Popular_Spotify_Songs |>
  select(streams,in_spotify_playlists, in_apple_playlists, danceability, energy, liv
suppressWarnings(Popular_Spotify_Songs$streams <- as.numeric(Popular_Spotify_Songs$s
Popular_Spotify_Songs$streams <- Popular_Spotify_Songs$streams/1000
Popular_Spotify_Songs <- Popular_Spotify_Songs |>
  drop_na(streams)
basemod <- lm(streams ~ in_spotify_playlists + in_apple_playlists, data = Popular_Sp
mod1 <- lm(streams ~ in_spotify_playlists + in_apple_playlists +  + danceability + e

summary(basemod)
summary(mod1)
AIC(basemod)
AIC(mod1)
tbl_summary(Popular_Spotify_Songs, include = c(in_spotify_playlists, in_apple_playli
  modify_caption(caption = "Statistical Summary of all Variables included in the Fir
all_subsets <- regsubsets(streams ~ ., data = Popular_Spotify_Songs)
## find the combination with the smallest BIC
plot(all_subsets)
final <- lm(streams ~ in_spotify_playlists + in_apple_playlists + energy + valence +
summary(mod1)
summary(final)
AIC(mod1)
AIC(final)
comparison1 <- data.frame(
  Model = c("Model 1", "Final Model"),
  Adjusted_R2 = c(summary(mod1)$adj.r.squared, summary(final)$adj.r.squared),
  AIC = c(AIC(mod1), AIC(final)),
  BIC = c(BIC(mod1), BIC(final))
)
# Displaying the table
kable(comparison1, caption = "Comparison of Adjusted R-squared, AIC and BIC values f
ggplot(data = Popular_Spotify_Songs, aes(x = in_spotify_playlists + in_apple_playlis
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  labs(x = "playlists and songs features", y = "Streams")
mod1table <- mod1 |>
  tbl_regression() |>
  modify_caption(caption = "Prediction of Streams for Model 1")
final |>
```

```
  tbl_regression() |>
  modify_caption(caption = "Prediction of Streams for Final Model")
mod2 <- lm(log(streams) ~ in_spotify_playlists + in_apple_playlists + energy + valen
mod2graph <- Popular_Spotify_Songs |>
  mutate(y_hat = fitted(mod2),
         resid = residuals(mod2)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals")

mod3 <- lm(sqrt(streams) ~ in_spotify_playlists + in_apple_playlists + energy + vale

mod3graph <- Popular_Spotify_Songs |>
  mutate(y_hat = fitted(mod3),
         resid = residuals(mod3)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals")
final <- lm(streams ~ in_spotify_playlists + in_apple_playlists + energy + valence +

Popular_Spotify_Songs |>
  mutate(y_hat = fitted(final),
         resid = residuals(final)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals")
Popular_Spotify_Songs |>
  ggplot(aes(sample = resid(final))) +
  geom_qq() +
  geom_qq_line(color = "blue") +
  labs(x = "Theoretical normal quantiles",
       y = "Observed residual quantiles") +
  theme_bw()
# Extracting Adjusted R-squared values and AIC values
comparison2 <- data.frame(
  Model = c("Base Model", "Model 1"),
```

```r
  Adjusted_R2 = c(summary(basemod)$adj.r.squared, summary(mod1)$adj.r.squared),
  AIC = c(AIC(basemod), AIC(mod1)),
  BIC = c(BIC(basemod), BIC(mod1))
)
mod1table
kable(comparison2, caption = "Comparison of Adjusted R-squared and AIC values for ba
mod2graph
mod3graph
suppressMessages(library(tidyverse))
library(gtsummary)
library(knitr)
Popular_Spotify_Songs <- read.csv("Popular_Spotify_Songs.csv")
Popular_Spotify_Songs <- Popular_Spotify_Songs |>
  select(streams,in_spotify_playlists, in_apple_playlists, danceability, energy, liv
# handling data(changing to more identified and assessible name & changing the scale
suppressWarnings(Popular_Spotify_Songs$streams <- as.numeric(Popular_Spotify_Songs$s
Popular_Spotify_Songs$streams <- Popular_Spotify_Songs$streams/1000
Popular_Spotify_Songs <- Popular_Spotify_Songs |>
  drop_na(streams)
#basic model examing the relationship between streams and playlists numbers
basemod <- lm(streams ~ in_spotify_playlists + in_apple_playlists, data = Popular_Sp
#model that add on confounding variables(including features of the songs)
mod1 <- lm(streams ~ in_spotify_playlists + in_apple_playlists +  + danceability + e

# comparing the adjusted R-squred and AIC of two model(second one is better!)
summary(basemod)
summary(mod1)
AIC(basemod)
AIC(mod1)
#check the p-value for the mod 1 to see which variables are irrelavant to the stream
mod1 |>
  tbl_regression()
#final model that delete some of the characteristics
final <- lm(streams ~ in_spotify_playlists + in_apple_playlists + energy + valence +
# comparing the adjusted R-squred and AIC of two model(second one is better!)
summary(mod1)
summary(final)
AIC(mod1)
AIC(final)
#final table
```

```r
final |>
  tbl_regression()
# trying transformation model
mod2 <- lm(log(streams) ~ in_spotify_playlists + in_apple_playlists + energy + valer
#plotting the residual plot for mod2
Popular_Spotify_Songs |>
  mutate(y_hat = fitted(mod2),
         resid = residuals(mod2)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals", title = "Log Transformation Residual Pl

mod3 <- lm(sqrt(streams) ~ in_spotify_playlists + in_apple_playlists + energy + vale
#ploting the residual plot for mod 3
Popular_Spotify_Songs |>
  mutate(y_hat = fitted(mod3),
         resid = residuals(mod3)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals", title = "Squared Transformation Residua
#fitting the final model(try multiple transformation and failed)
final <- lm(streams ~ in_spotify_playlists + in_apple_playlists + energy + valence +
#one missing data in streams(handle it through dropping it)
#checking conditions for the final model
#residual plot(linearity, zero mean, constant variance)
Popular_Spotify_Songs |>
  mutate(y_hat = fitted(final),
         resid = residuals(final)) |>
  ggplot(aes(x = y_hat, y = resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Residuals")
#qq plot(normality)
Popular_Spotify_Songs |>
  ggplot(aes(sample = resid(final))) +
  geom_qq() +
  geom_qq_line(color = "blue") +
```

```
  labs(x = "Theoretical normal quantiles",
       y = "Observed residual quantiles") +
  theme_bw()
#plotting the final model
ggplot(data = Popular_Spotify_Songs, aes(x = in_spotify_playlists + in_apple_playlis
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  labs(x = "playlists and songs features", y = "Streams") +
  ggtitle("Linear Model of Streams")
# Extracting Adjusted R-squared values and AIC values
comparison1 <- data.frame(
  Model = c("Model 1", "Final Model"),
  Adjusted_R2 = c(summary(mod1)$adj.r.squared, summary(final)$adj.r.squared),
  AIC = c(AIC(mod1), AIC(final))
)

# Displaying the table
kable(comparison1, caption = "Comparison of Adjusted R-squared and AIC values for ba

# Extracting Adjusted R-squared values and AIC values
comparison2 <- data.frame(
  Model = c("Base Model", "Model 1"),
  Adjusted_R2 = c(summary(basemod)$adj.r.squared, summary(mod1)$adj.r.squared),
  AIC = c(AIC(basemod), AIC(mod1))
)
# Displaying the table
kable(comparison2, caption = "Comparison of Adjusted R-squared and AIC values for mo
allsubset <- regsubsets(streams ~ ., data = Popular_Spotify_Songs)
plot(allsubset)
```

Table 4: Prediction of Streams for Model 1

Characteristic	**Beta**	**95% CI**	**p-value**
in_spotify_playlists	35	31, 38	<0.001
in_apple_playlists	2,841	2,524, 3,157	<0.001
danceability	-457	-1,978, 1,064	0.6
energy	-1,150	-2,697, 397	0.14
liveness	-9.9	-1,427, 1,408	>0.9
valence	-910	-1,887, 67	0.068

Characteristic	**Beta**	**95% CI**	**p-value**
acousticness	625	-318, 1,568	0.2
instrumentalness	-1,122	-3,433, 1,189	0.3
speechiness	-1,093	-3,075, 890	0.3

Table 5: Comparison of Adjusted R-squared and AIC values for base model and model 1

Model	Adjusted_R2	AIC	BIC
Base Model	0.7139662	26739.02	26758.45
Model 1	0.7187370	26729.96	26783.40

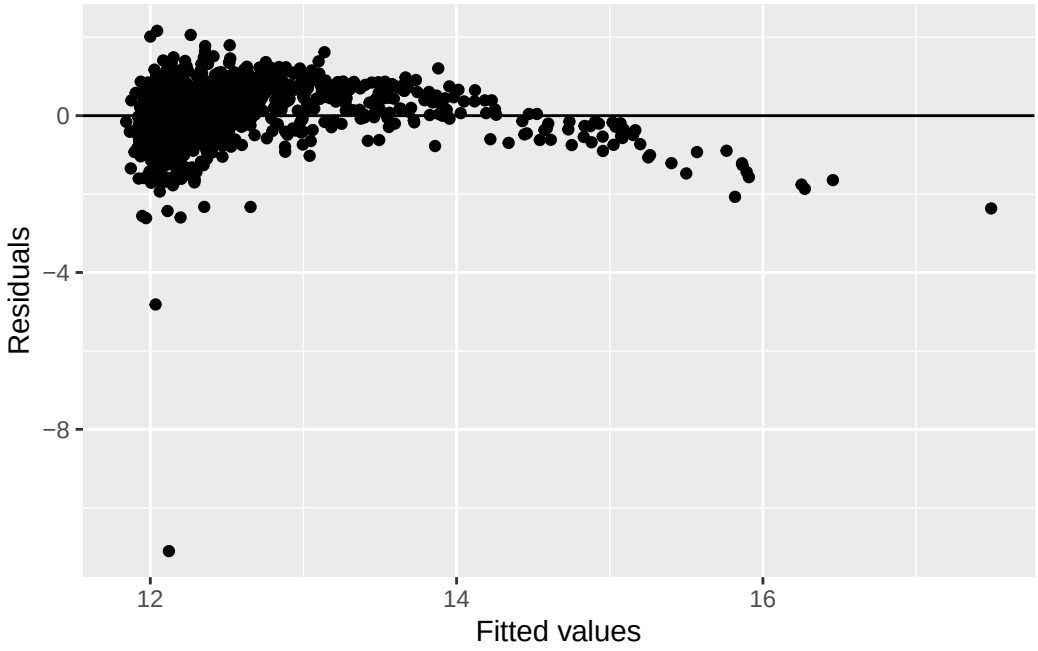

Figure 5: The Residual Plot for Log Transformation

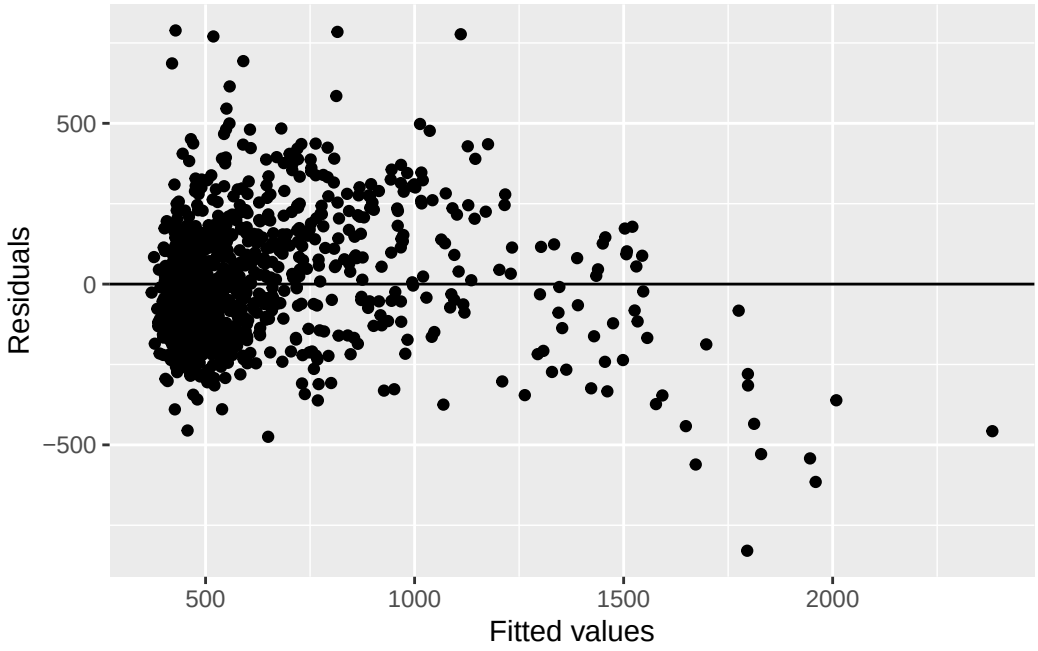

Figure 6: The Residual Plot for Squared Transformation