

Final 210 Project

Sean Villoresi and Ellie Kang

Introduction

Music is an essential part of culture, creativity, and history. Specific songs and types of music can have great significance to groups of people and individuals alike. In America today, the music industry is both highly regarded and hotly debated. One successful song can launch an artist to the top of the charts, etching them into modern history. The importance of music and potential significance of a single song motivates the question - what makes a song successful?

Data

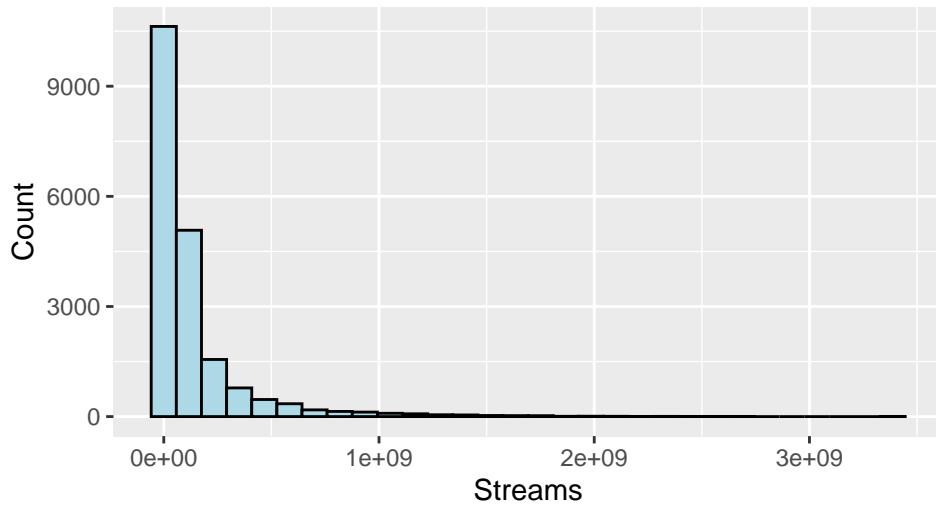
Wanting to explore this question in our project, we found a dataset¹ on Kaggle with data on Spotify streams, Youtube views, and various song characteristics. There are 28 columns, and 20,719 observations. The data was collected on February 7th, 2023 by extracting the data from YouTube and Spotify. Our goal is to determine and develop the best model for predicting the success of a song based on the number of streams. We chose to use streams (as opposed to YouTube views) as our outcome variable because of inconsistencies within the data when it comes to music videos. Some music videos were not from the artist's channel (unofficial), and we wanted to test this variable as a predictor.

Variables: We will be using streams as the model's outcome variable. We chose the following variables as potential predictors based on their relevance to the listening experience of a song (as opposed to a more descriptive variable such as Description) Stream: number of streams of the song on Spotify Energy: a measure from 0.0 to 1.0 representing a perceptual measure (dynamic range, loudness, timbre, onset rate, general entropy) of intensity and activity. Key: the key the track is in measured in integers representing pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C /D , 2 = D. If no key was detected, the value is -1. Loudness: the overall loudness of a track in decibels (dB) Speechiness: a measure from 0.0 to 1.0 representing the presence of spoken words in a track. Acousticness: a from 0.0 to 1.0 of whether the track is acoustic. Instrumentalness: a measure from 0.0 to 1.0 that predicts whether a track contains no vocals. Liveness: a measure from 0.0 to 1.0 that detects the presence of an audience in the recording. Valence: a measure from 0.0 to 1.0 describing the musical positiveness conveyed

by a track. Tempo: the overall estimated tempo of a track in beats per minute (BPM). Duration_ms: the duration of the track in milliseconds. Official_video: boolean value that indicates if the video found is the official video of the song. We felt that the dataset had a sufficient number of both quantitative and categorical variables to test predictability. Thus, we chose not to create additional predictors. However, in our data cleaning process, we removed any observations with missing values for streams, danceability, and licensed. After removing missing values for these variables, there were no remaining observations with missing data for relevant variables as listed above.

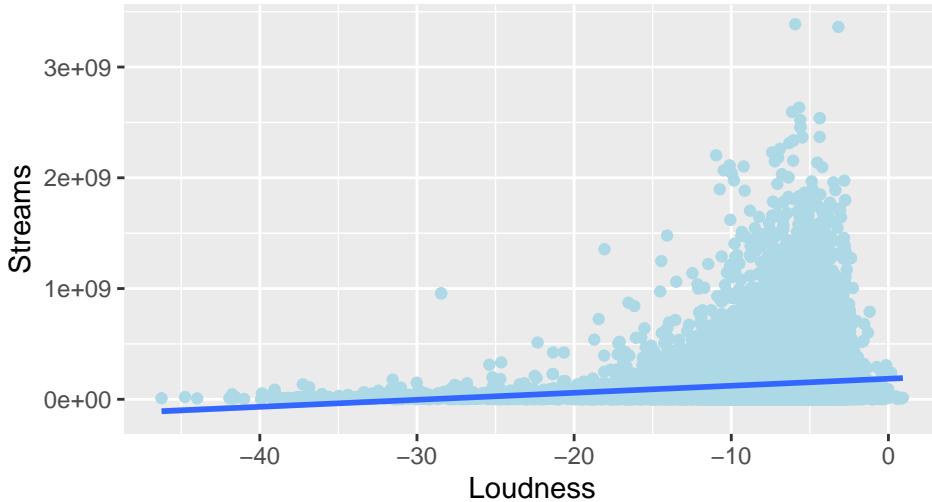
Exploratory Data Analysis

Distribution of Streams

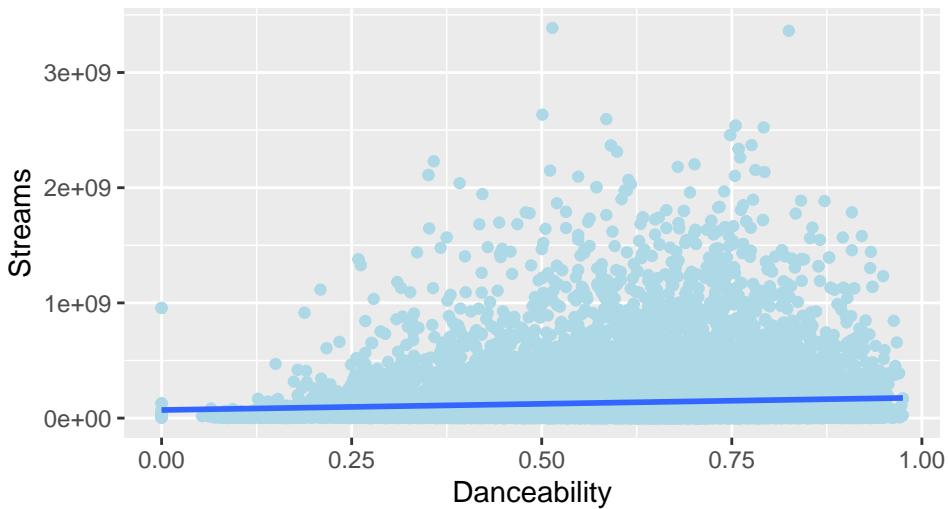


The distribution of the response variable Stream is skewed to the right with most of the data having lower values for Streams. This is understandable because a handful of songs are extremely popular while most songs on Spotify tend to be listened to by specific niches of people.

Streams vs. Loudness



Streams vs. Danceability



We visualized the relationship between Streams and all the predictors in the dataset. Danceability and Loudness showed slight positive relationships with the response variable Stream. The remaining visualizations are in the Appendix.

Danceability	Energy	Loudness	Speechiness
1.614911	3.447032	3.240748	1.090299
Acousticness	Instrumentalness	Liveness	Valence
1.915335	1.569425	1.066099	1.529906
Tempo	Duration_ms	official_video	

1.063285 1.014723 1.030153

The results of looking at Variance Inflation Factor are values in a range between 1 and 4. Therefore, we chose to keep all the initial predictors when performing variable selection.

Min. 1st Qu. Median Mean 3rd Qu. Max.
6.574e+03 1.777e+07 4.974e+07 1.369e+08 1.391e+08 3.387e+09

Min. 1st Qu. Median Mean 3rd Qu. Max.
0.0000 0.5190 0.6390 0.6209 0.7420 0.9750

Min. 1st Qu. Median Mean 3rd Qu. Max.
-46.251 -8.786 -6.518 -7.641 -4.931 0.920

interpretation

Methods

Variable Selection

Step-wise Selection

Step: AIC=760164.9
Stream ~ Danceability + Energy + factor(Key) + Loudness + Speechiness +
Acousticness + Instrumentalness + Liveness + Valence + Duration_ms +
official_video

	Df	Sum of Sq	RSS	AIC
<none>		1.1458e+21	760165	
+ Tempo	1	4.1097e+16	1.1457e+21	760166
- Duration_ms	1	1.9879e+17	1.1460e+21	760166
- factor(Key)	11	1.4476e+18	1.1472e+21	760168
- Instrumentalness	1	6.8194e+17	1.1465e+21	760175
- Liveness	1	8.9435e+17	1.1467e+21	760178
- Speechiness	1	1.0322e+18	1.1468e+21	760181
- Danceability	1	1.1555e+18	1.1469e+21	760183
- Valence	1	2.1406e+18	1.1479e+21	760200
- Energy	1	4.6795e+18	1.1505e+21	760243
- Acousticness	1	5.7319e+18	1.1515e+21	760261
- Loudness	1	6.0882e+18	1.1519e+21	760267

```
- official_video     1 8.2293e+18 1.1540e+21 760304
```

Call:

```
lm(formula = Stream ~ Danceability + Energy + factor(Key) + Loudness +
    Speechiness + Acousticness + Instrumentalness + Liveness +
    Valence + Duration_ms + official_video, data = music)
```

Coefficients:

	(Intercept)	Danceability	Energy	factor(Key)1
	2.763e+08	5.804e+07	-1.341e+08	1.218e+07
factor(Key)2		factor(Key)3	factor(Key)4	factor(Key)5
	-1.055e+07	-5.933e+06	-2.509e+06	-2.213e+05
factor(Key)6		factor(Key)7	factor(Key)8	factor(Key)9
	2.366e+06	-1.237e+07	3.048e+06	-1.790e+07
factor(Key)10		factor(Key)11	Loudness	Speechiness
	-4.902e+06	6.495e+06	6.844e+06	-7.161e+07
Acousticness		Instrumentalness	Liveness	Valence
	-8.247e+07	-3.828e+07	-4.217e+07	-5.239e+07
Duration_ms	official_videoTRUE			
	-2.530e+01	4.996e+07		

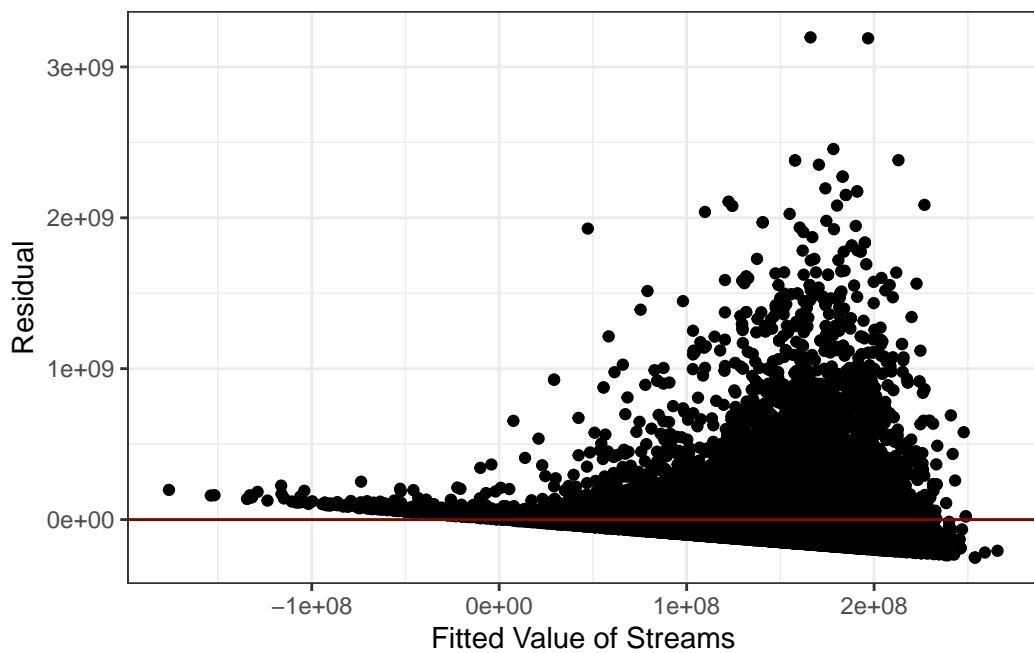
Lasso Model

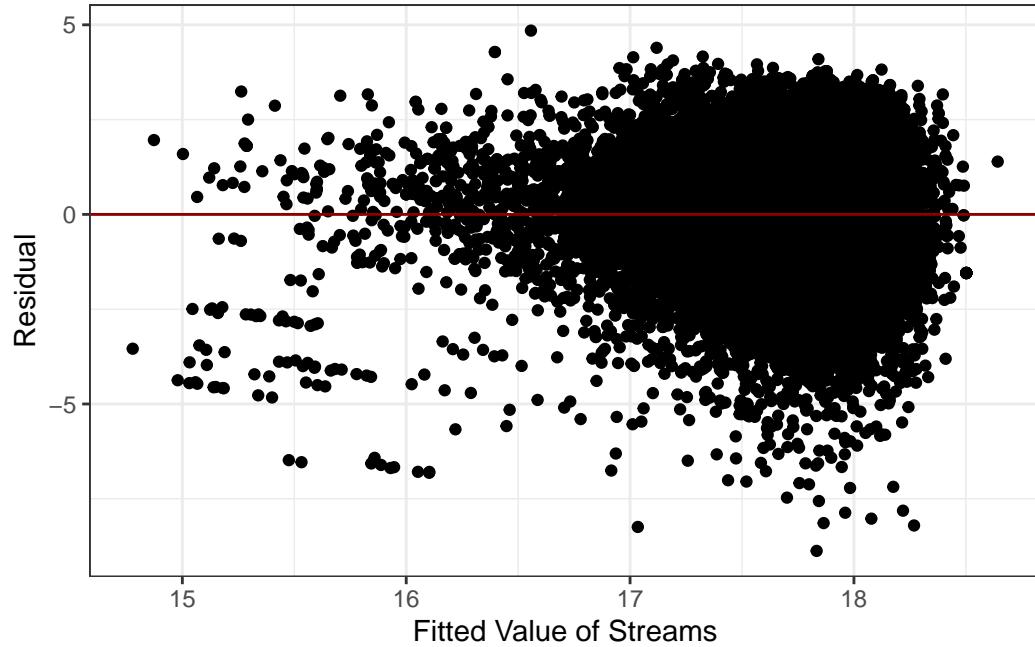
```
23 x 1 sparse Matrix of class "dgCMatrix"
      s0
(Intercept) .
Danceability   5.425383e+07
Energy        -1.225667e+08
factor(Key)1   1.186910e+07
factor(Key)2   -7.844484e+06
factor(Key)3   -1.761978e+06
factor(Key)4   .
factor(Key)5   .
factor(Key)6   1.613351e+06
factor(Key)7   -9.784489e+06
factor(Key)8   2.302353e+06
factor(Key)9   -1.506024e+07
factor(Key)10  -1.676901e+06
factor(Key)11  5.852992e+06
Loudness       6.532253e+06
Speechiness    -6.401535e+07
Acousticness   -7.858671e+07
```

Instrumentalness	-3.777677e+07
Liveness	-4.111619e+07
Valence	-4.954162e+07
Tempo	-3.435953e+04
Duration_ms	-2.149974e+01
official_videoTRUE	4.901070e+07

Linearity Assumptions and Checks for Transformations

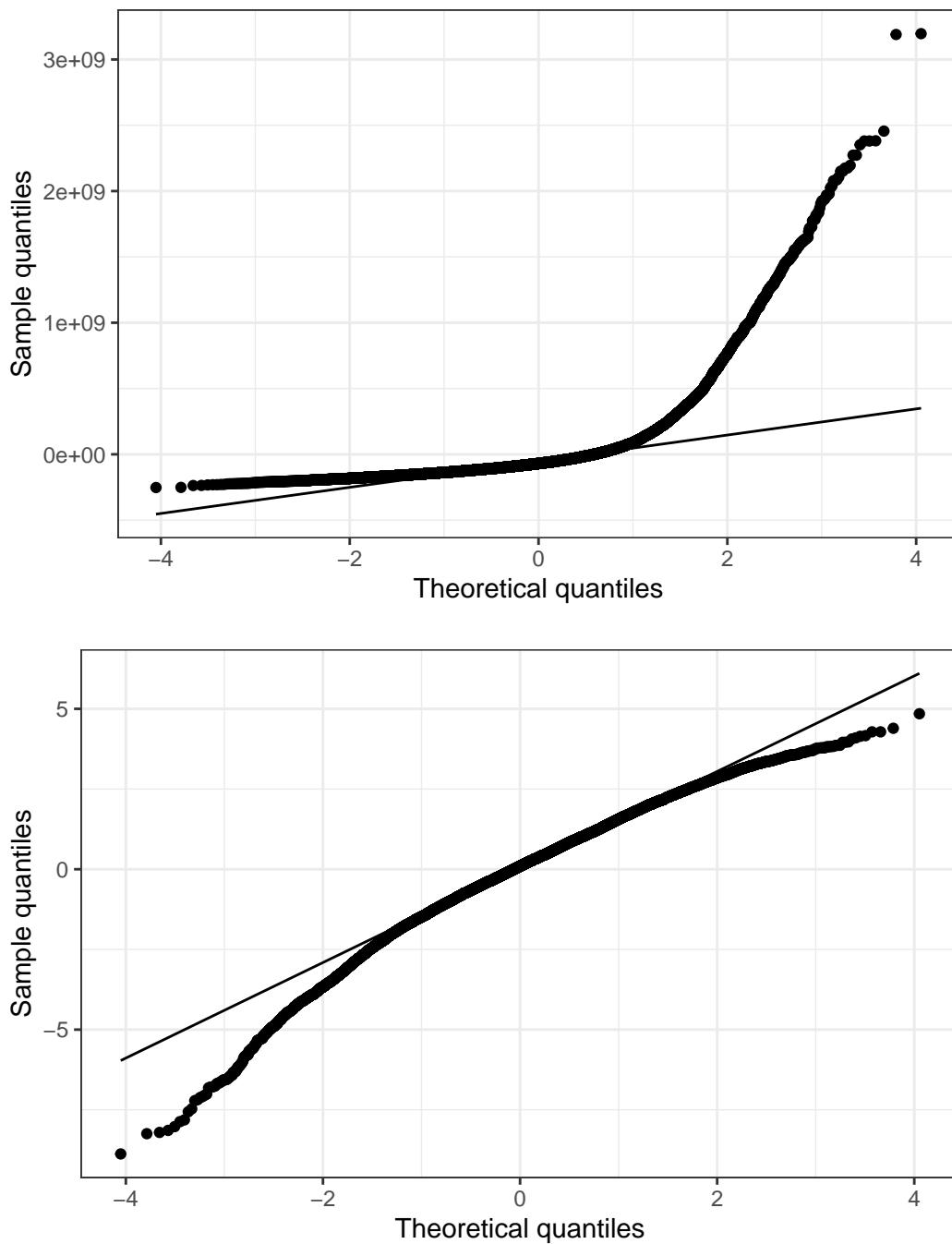
Residual Models





Looking at the visualizations above, we can see that the transformed model gives us a much better spread on the residual split around our red line then our untransformed model. As such, the residuals appear roughly symmetrical along the horizontal axis for our transformed plot, so we feel it safe to assume approximate linearity.

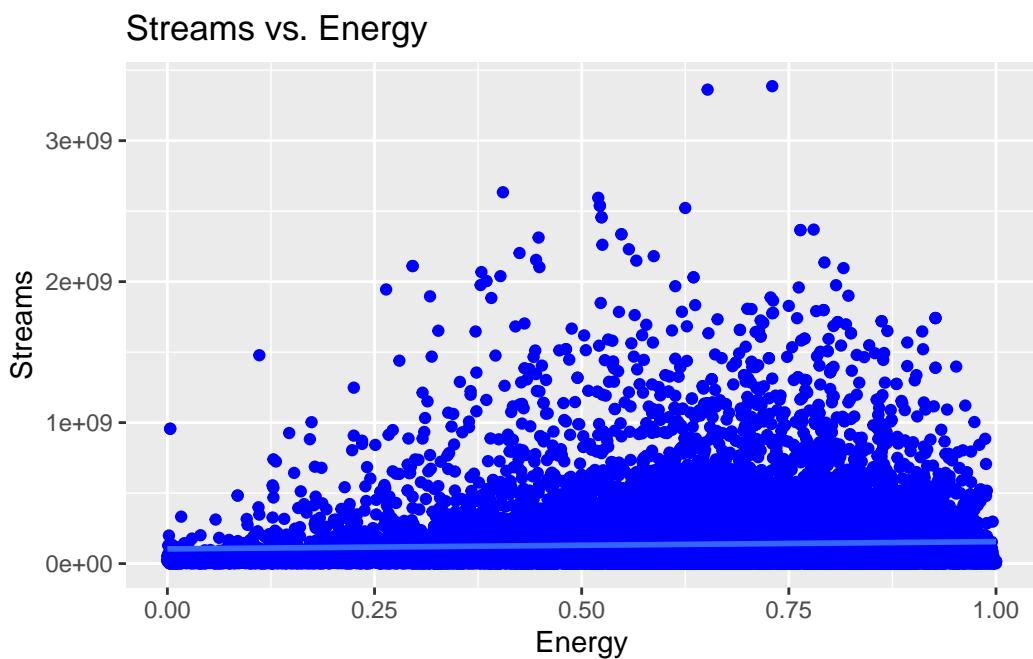
QQ Plots



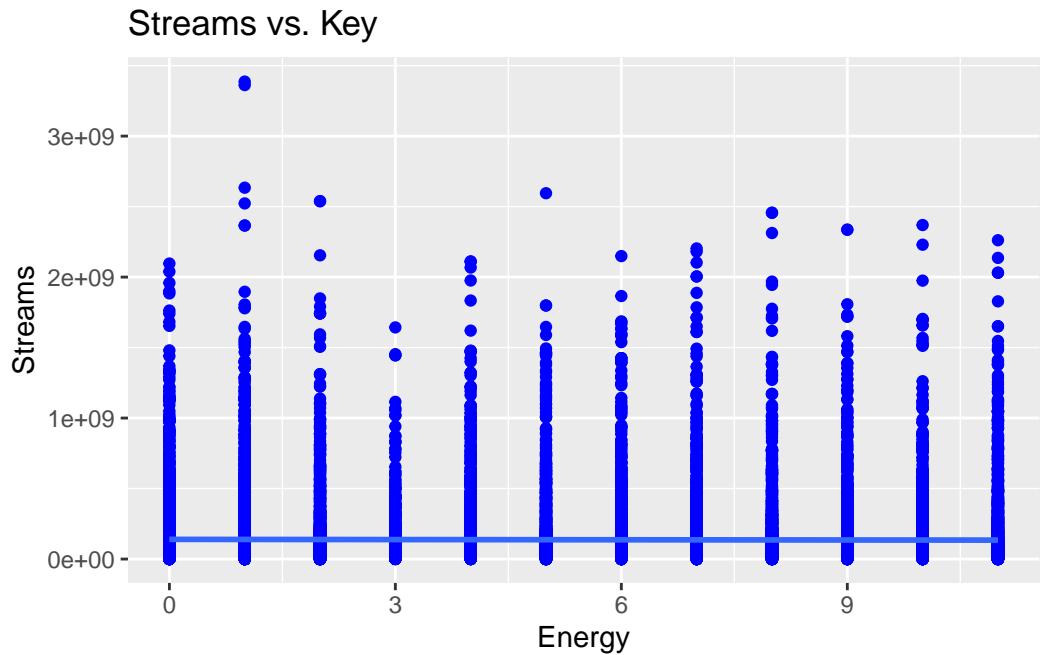
Results

Appendix

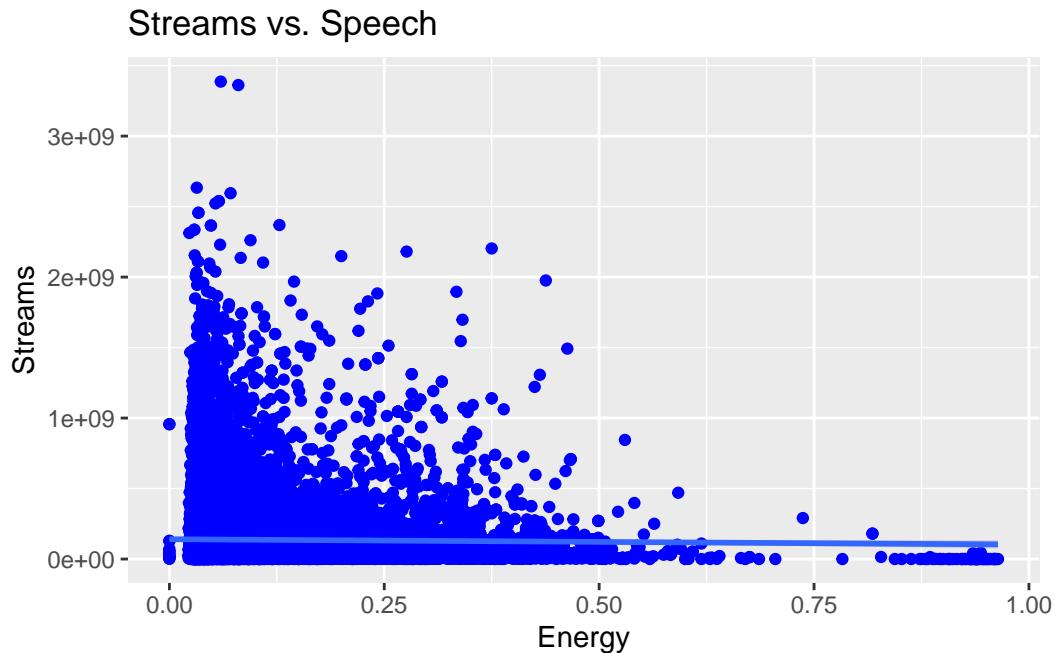
```
ggplot(music, aes(x = Energy, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Energy",  
       x = "Energy", y = "Streams")  
  
'geom_smooth()' using formula = 'y ~ x'
```



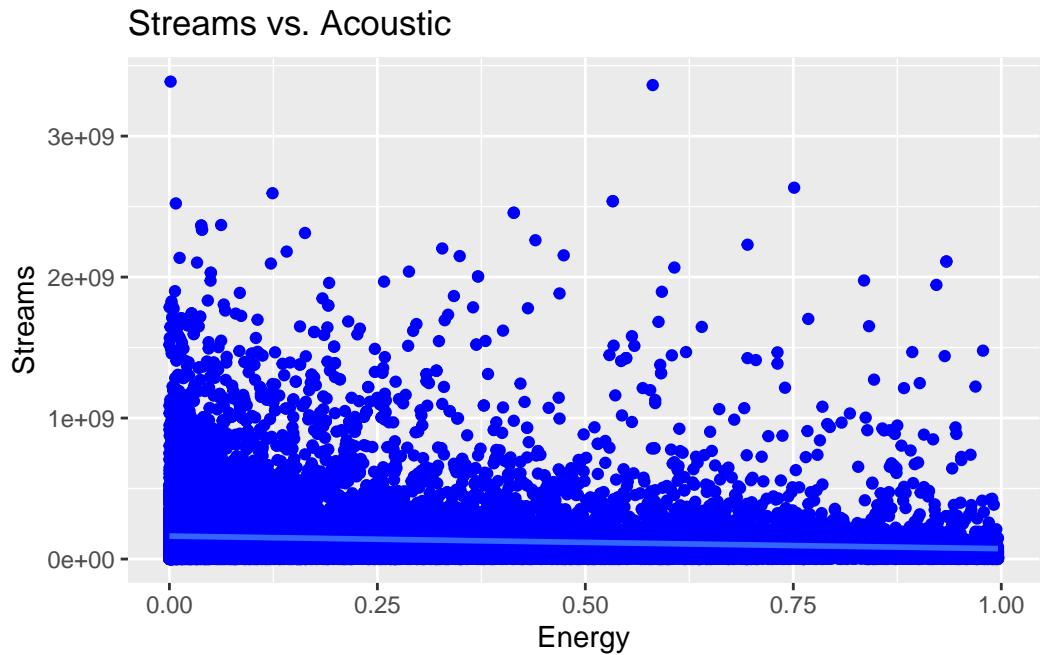
```
ggplot(music, aes(x = Key, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Key",  
       x = "Energy", y = "Streams")  
  
'geom_smooth()' using formula = 'y ~ x'
```



```
ggplot(music, aes(x = Speechiness, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Speech",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```



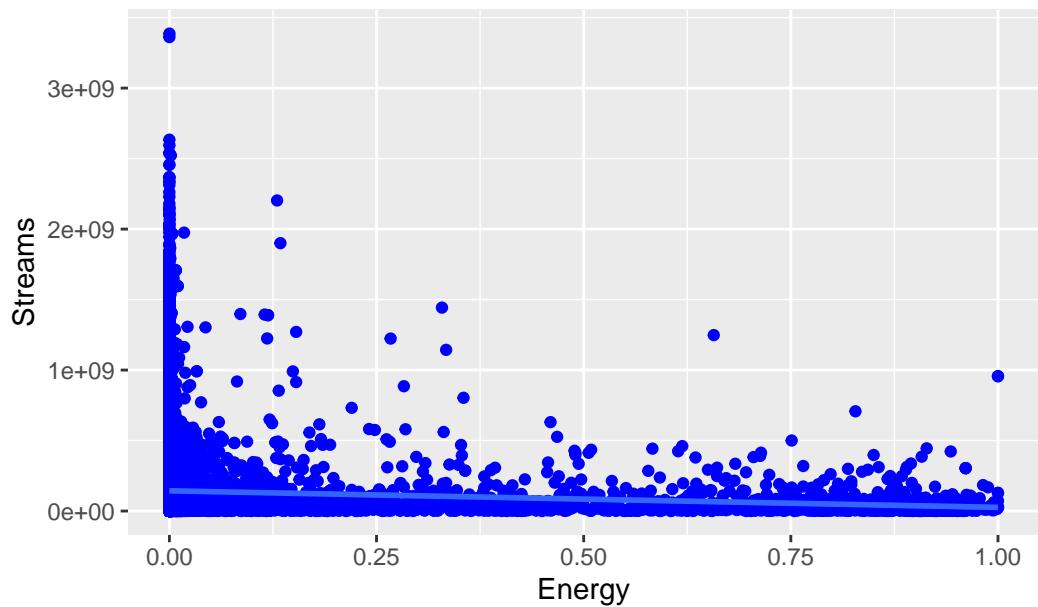
```
ggplot(music, aes(x = Acousticness, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Acoustic",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(music, aes(x = Instrumentalness, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Instrumental",  
       x = "Energy", y = "Streams")
```

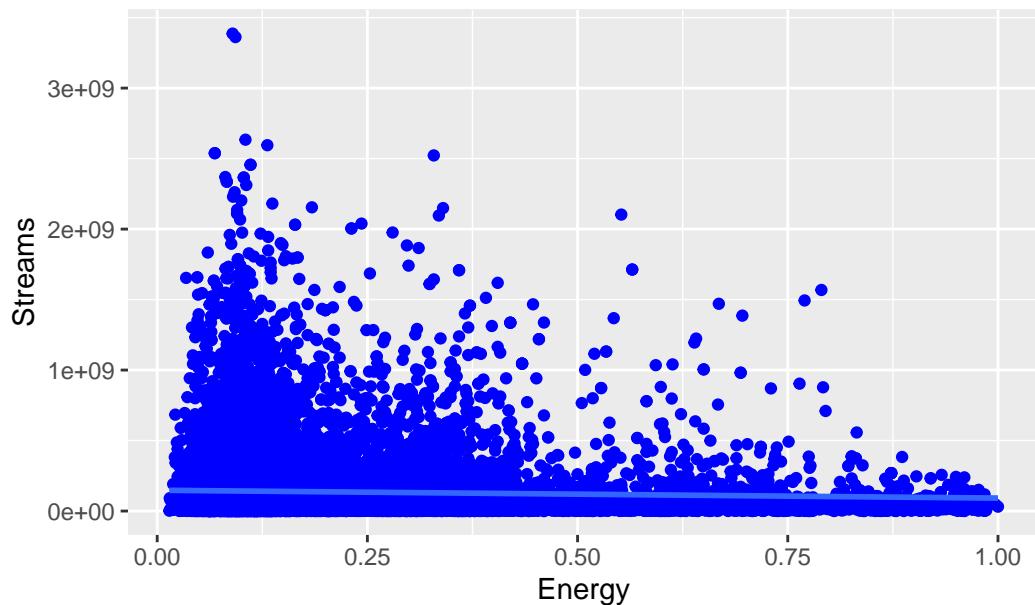
```
`geom_smooth()` using formula = 'y ~ x'
```

Streams vs. Instrumental

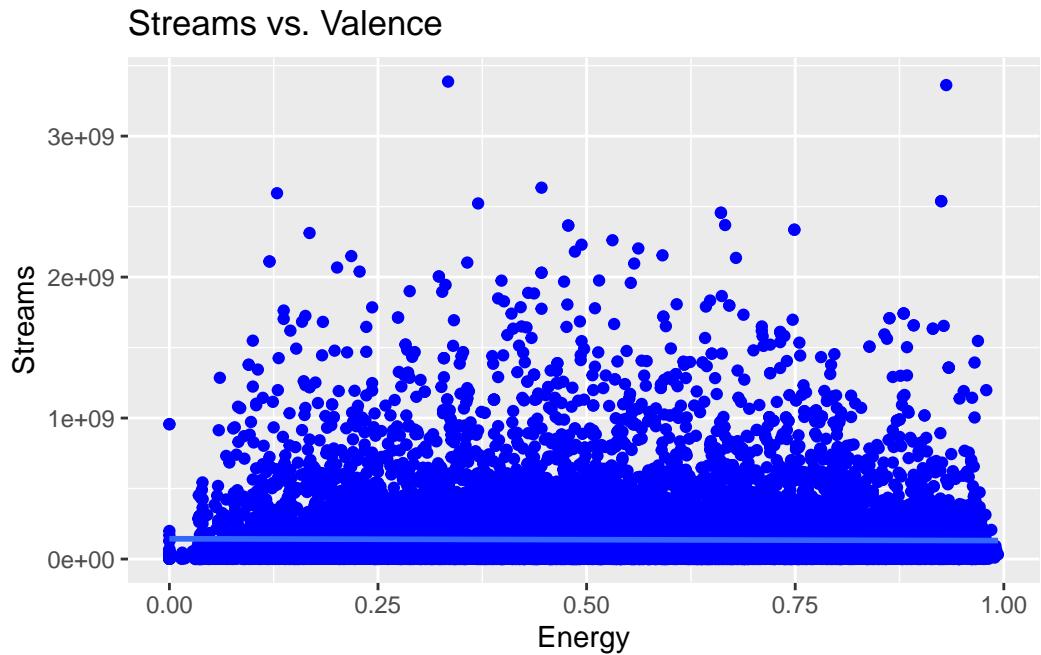


```
ggplot(music, aes(x = Liveness, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Liveness",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```

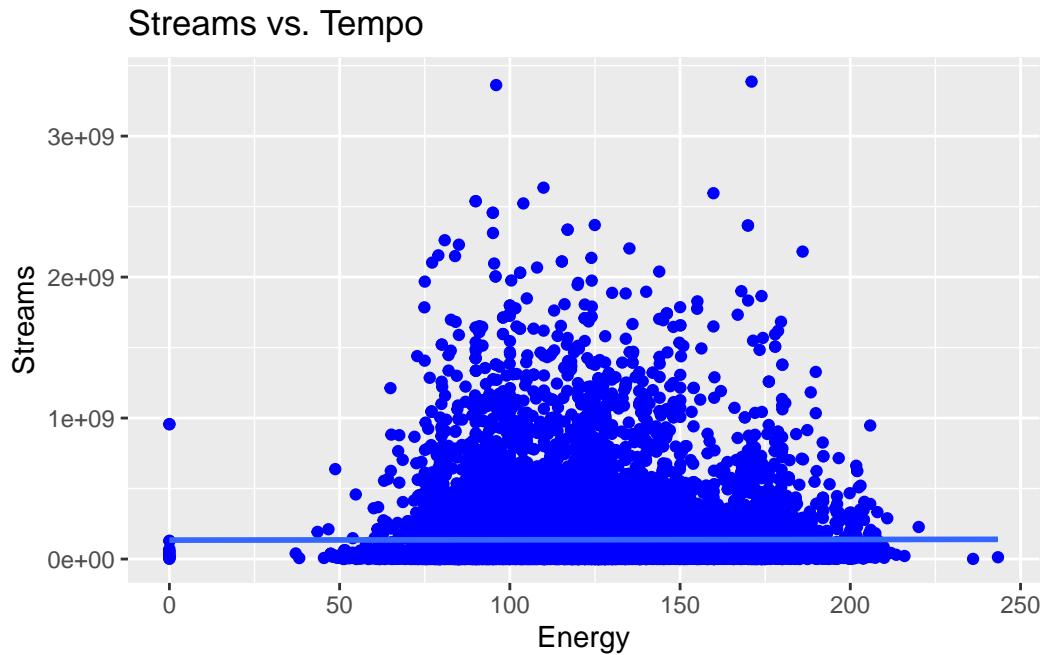
Streams vs. Liveness



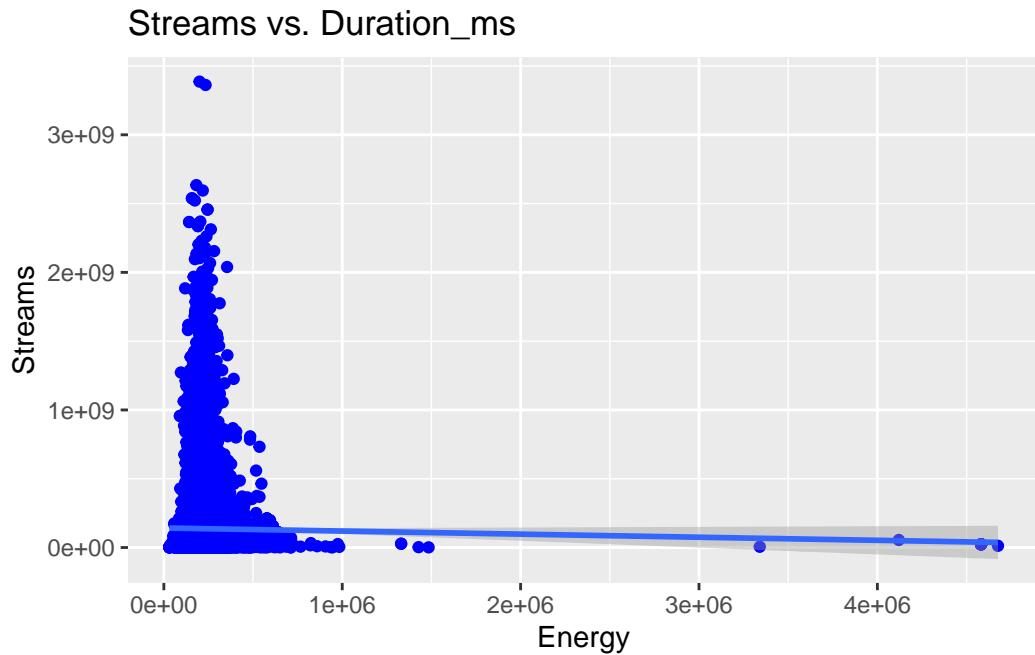
```
ggplot(music, aes(x = Valence, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Valence",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(music, aes(x = Tempo, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Tempo",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```



```
ggplot(music, aes(x = Duration_ms, y = Stream)) +  
  geom_point(color = "blue") +  
  geom_smooth(method = "lm") +  
  labs(title = "Streams vs. Duration_ms",  
       x = "Energy", y = "Streams")  
  
`geom_smooth()` using formula = 'y ~ x'
```



```
summary(music[8:18])
```

Key	Loudness	Speechiness	Acousticness
Min. : 0.000	Min. :-46.251	Min. : 0.0000	Min. : 0.0000011
1st Qu.: 2.000	1st Qu.: -8.786	1st Qu.: 0.0357	1st Qu.: 0.0444000
Median : 5.000	Median : -6.518	Median : 0.0505	Median : 0.1910000
Mean : 5.294	Mean : -7.641	Mean : 0.0952	Mean : 0.2895899
3rd Qu.: 8.000	3rd Qu.: -4.931	3rd Qu.: 0.1040	3rd Qu.: 0.4720000
Max. :11.000	Max. : 0.920	Max. : 0.9640	Max. : 0.9960000
Instrumentalness	Liveness	Valence	Tempo
Min. : 0.0000000	Min. : 0.0145	Min. : 0.000	Min. : 0.00
1st Qu.: 0.0000000	1st Qu.: 0.0940	1st Qu.: 0.338	1st Qu.: 96.99
Median : 0.0000024	Median : 0.1250	Median : 0.536	Median : 119.96
Mean : 0.0554328	Mean : 0.1912	Mean : 0.529	Mean : 120.57
3rd Qu.: 0.0004420	3rd Qu.: 0.2340	3rd Qu.: 0.725	3rd Qu.: 139.94
Max. : 1.0000000	Max. : 1.0000	Max. : 0.993	Max. : 243.37
Duration_ms	Title	Channel	
Min. : 30985	Length:19691	Length:19691	
1st Qu.: 180160	Class :character	Class :character	
Median : 213267	Mode :character	Mode :character	
Mean : 224577			
3rd Qu.: 251876			

Max. : 4676058