## 

## 

## 

ISYE 6404 Nonparametric Data Analysis

Project Final Report

*Spotlight on Spotify: Predicting Music Trends and Hits*

**Group 9 Members: Vaishnavi Uppaluru, Estella Tu, Sarah Friedrichs, and Eyas Ayesh**

## 

## 

## **Summary**

The music industry has undergone a seismic shift in the digital era, with streaming platforms like Spotify serving as key indicators of a song’s success. Understanding how musical attributes influence popularity can provide critical insights for artists, producers, and streaming platforms. This study explores a dataset comprising the most-streamed Spotify tracks to examine relationships between musical characteristics (e.g., danceability, valence, energy) and popularity, measured by the number of streams. Given the non-normality of streaming data, nonparametric methods were employed to ensure robust analysis.

The analysis began with data cleaning and preliminary exploration of the statistical behavior of the explanatory variable to identify any abnormalities and validate underlying assumptions. When necessary, transformations were used to address skewness in variables like acousticness and speechiness. Normality and the fitting of distributions were assessed using goodness-of-fit tests, including the Kolmogorov-Smirnov test, Anderson-Darling test, and QQ-plots. Additionally, the similarity of distributions after segmenting the data by year was tested using the Smirnov test and Kruskal-Wallis test. Next, Friedman test was applied to evaluate differences in distributions across musical attributes, followed by pairwise Wilcoxon rank-sum tests to identify specific attribute discrepancies. Theil-Sen regression further quantified the effects of attributes like danceability and valence on streams, revealing significant negative relationships. Spearman correlations provided complementary insights, albeit showing weak overall monotonic trends.

The results underscore how higher levels of speechiness or danceability may detract from mainstream appeal, while tonal key also influences performance. These findings highlight the nuanced relationship between song features and audience preferences, offering valuable guidance for optimizing music production and marketing strategies. Future research could expand by analyzing genre or listener demographics to build a more holistic model of musical success.

## **Table of Contents**

[Summary 1](#_7uj4wavgmj9v)

[Table of Contents 2](#_zbovpt2ue0vs)

[Introduction 3](#_cw9pldvb8re1)

[1.1 Description of Reason for Our Study 3](#_x665rdg482u4)

[1.2 Description of Raw Data 3](#_tw45u96j0kcv)

[1.3 Data Cleaning 3](#_pw6u8uaxzx2e)

[Statistical Analysis 5](#_4d8jad9k1kek)

[2.1 Goodness of Fit 5](#_r1zo8pvb61n)

[2.1.1 Beats Per Minute (BPM) 6](#_2fypj67lcabm)

[2.1.2 Danceability 7](#_m4sfuwmnie76)

[2.1.3 Normality of Streams 8](#_j1u27v9s2bkv)

[2.2 Comparing Distributions of Years and Streams 9](#_s4wiyg7vc59r)

[3.0 Subject Matter Implications of Study 13](#_y3dawzjw5p2q)

[4.0 Further questions 14](#_qjshl5msnqsm)

[5.0 Conclusion 14](#_isjsk0iykod7)

[Appendix 15](#_ekh0oz307wne)

## 

## 

## 

## 

## **Introduction**

## **1.1 Description of Reason for Our Study**

The goal of this project is to conduct comprehensive data analysis to understand the trends in popular music over time and build a model for estimating the performance (rank in playlists) of a song from its known features. To that end, we explore the Spotify dataset to enable a deeper understanding of the various attributes of songs on the platform, and how these attributes can affect performance.In our approach, we make an effort to understand the actual statistical attributes of each feature without projecting assumptions of normality that may not be correct.

## **1.2 Description of Raw Data**

This dataset contains comprehensive information on Spotify's most streamed songs, providing insights into trends across various artists and years Additionally, the dataset contains information about the rankings of these same songs on other streaming platforms enabling a deeper understanding of how the specific platform may affect the trend. The information includes basic track details such as song name, year, artists, etc as well as additional streaming metrics and musical attributes. Musical attributes are those concerned with the actual production quality of the song and include variables such as bpm, key, mode,danceability, valence, energy, acousticness, instrumentalness, liveliness, and speechiness. The data set contains 943 unique observations of tracks or songs.

The data is publicly available on kaggle as “Spotify Most Streamed Songs.” More information can be found [here](https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs).

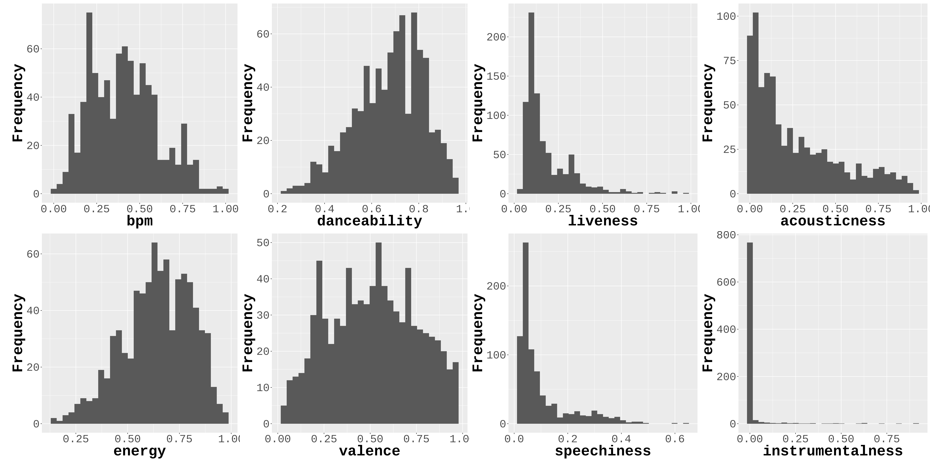
## **1.3 Data Cleaning**

The data cleaning process involved several steps. First we explored the contents of the data set and separated it into three categories: explanatory variables, predicted variables and supplementary information. Initially, it appeared that none of the data was missing, but upon further examination we determined that the numerical columns that had missing values were formatted as strings. After converting all numerical columns back to floats and omitting rows with missing data, we had 826 observations left.

Next, all the numerical columns were scaled via a min-max linear scaling (minimum value set to zero, and maximum value set to one) in order to allow for fair comparisons and equal training in the regression model. The numerical columns at this stage were bpm, ket, danceability, energy, acousticness, instrumentalness, liveness, and speechines.

After basic preprocessing, the next step was to visualize the distributions of these variables and determine whether it would be appropriate to transform them. Ideally, the distributions of these parameters would appear normal. However, failing to meet that condition, we prefer transformations that reduce the skew of the data and allow for more spread out distributions.

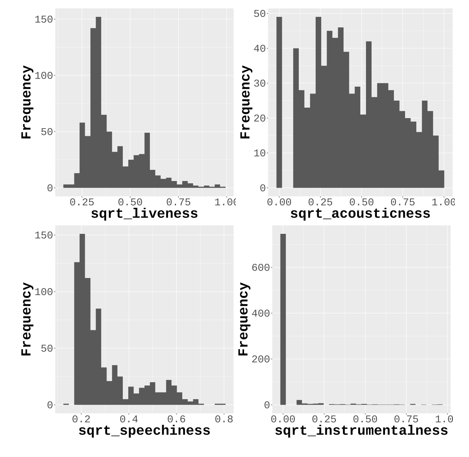
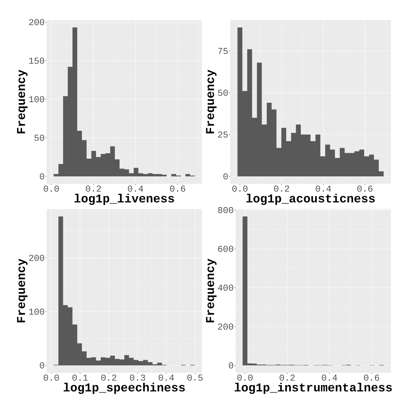
A rough idea of the underlying distribution for each of the numerical variables can be obtained by generating histogram plots. In Figure 1 below, while not explicitly normal, we can see that 4 categories stand out for having healthy distributions: bpm, valence, danceability, and energy. The other 4 appear to be heavily right skewed with the instrumentalness category being of particular concern.



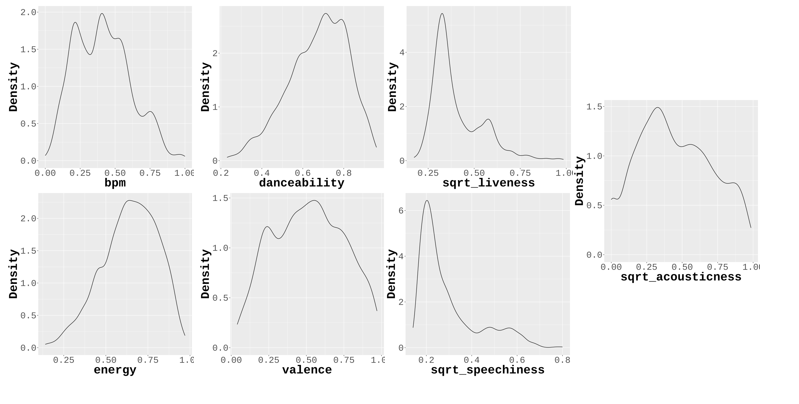
*Figure 1.* The histogram distributions of the unprocessed numerical attributes

When it came to transforming the data, we explored two options - a log(x+1) transform and a square root transform. In figure a we can observe that the log transformation did not change much about the distribution of the variable observations. On the other hand, the square root transformation shown in Figure2 was able to effectively spread out the distribution of the acousticness and move the liveness and speechiness slightly less skewed. The instrumentalness remained unaffected. The reason for this was because for most songs (over 92%) there are no instrumentals. For this reason, we simply decided to transform it into a binary categorical variable with values of True corresponding to the existence of instruments in the song.

Once the transformations were completed, we estimated the probability density of the distributions non-parametrically using the kernel method. The best estimates were obtained with a gaussian kernel and a bandwidth of 0.75. These can be seen in Figure 4

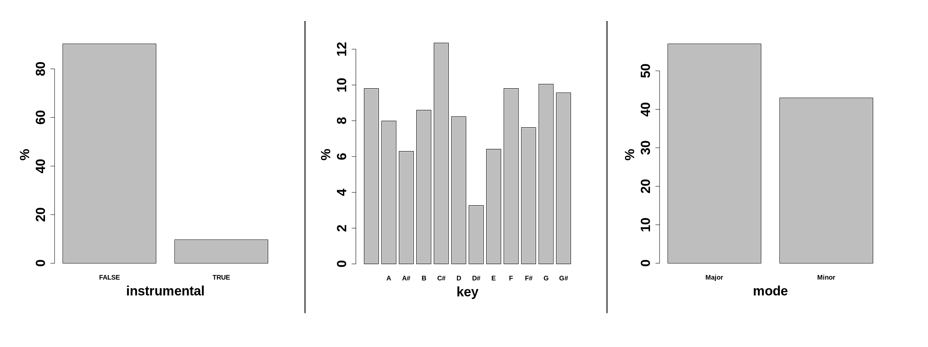


*Figure 2.* Log transformations (left) andsquare root transformations (right)



*Figure 4.* Probability density estimations with a gaussian kernel and 0.75 bandwidth

For non-categorical data we generated bar plots to observe their relative proportions. For both the keys and the mode, the data seems to be roughly evenly distributed, but with instrumental the data is still extremely skewed.



*Figure 5*. Bar plots of categorical data

## **Statistical Analysis**

## **2.1 Goodness of Fit**

The first goal was to determine if any of the common characteristics of the top songs could be fit using a parametric distribution. Each of the characteristics was normalized within its own set of values. The characteristics that were fit are the beats per minute and danceability. From there, the *fitdistrplus* library in R was utilized to fit each of five common distributions to the dataset. The distributions that the data were fit to are beta, exponential, gamma, weibull, and normal. These distributions were each fit using the maximum likelihood estimator method. The fit distributions were compared to the dataset itself for goodness of fit using the Kolmogorov Test to determine if the distribution was an appropriate fit. Furthermore, the Anderson-Darling test was conducted to further determine if the data were normally distributed. A QQ plot was also produced of the best reference probability distribution to further validate the findings of the fit distribution in comparison to the data.

For each Kolmogorov Test that was completed, the null hypothesis was that the distribution of the characteristic was the same as the fitted distribution and the alternate hypothesis being that the distribution of characteristic was not the same as the fitted distribution for each of the five distributions that were fit for it.

An Anderson Darling test was also completed to further validate the results of the Kolmogorov Test for the normal distribution that was fit to the data of each of the characteristics. The null hypothesis of this test was that the distribution of the characteristic data are normally distributed and the alternate hypothesis was that the distribution of characteristic data are not normally distributed.

### **2.1.1 Beats Per Minute (BPM)**

The first characteristic that was fit to a distribution was the beats per minute of the songs. The parameters that were found using the *fitdistrplus* library in R for the BPM distribution can be found in table 1.

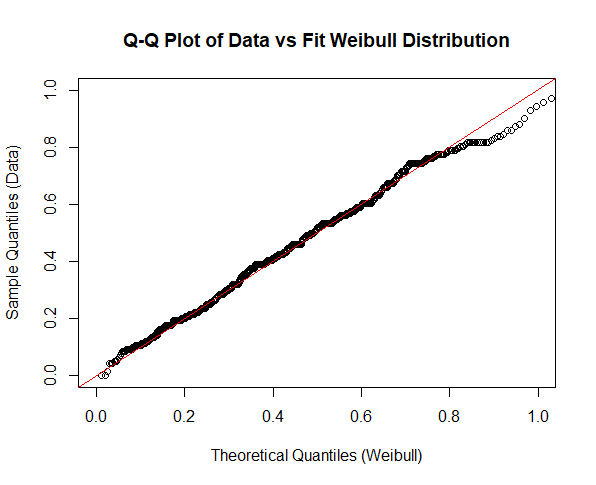
*Table 1:* Parameters of Distributions fit to BPM Data

| Distribution | Fit Parameters | p-values |
| --- | --- | --- |
| Beta | shape1 = 1.580601, shape2 = 2.199773 | 5.573\*10-6 |
| Exponential | rate = 2.450452 | < 2.2\*10-16 |
| Gamma | shape = 2.84181, rate = 6.96323 | 2.942\*10-7 |
| Weibull | shape = 2.0590890, scale = 0.4564433 | 0.004624 |
| Normal | mean = 0.4080879, sd = 0.1988871 | 0.002762 |

The p-values for the Kolmogorov Test for each fit distribution with distribution of BPMs is found in table X. The alpha-level of our test is 0.05. Therefore, for each of the tests, we reject the null hypothesis that the fit dataset matches that of the BPM dataset itself and have evidence to support that they are significantly different.

The p-value of the respective Anderson-Darling test was found to be 7.927\*10-13 which means that on the 0.05 alpha-level we reject the null hypothesis that the BPM dataset is normally distributed.

We then plotted a QQ plot of the data against the best fit distribution, the weibull distribution. Figure 6 emphasizes how the data do not follow the line y=x and provides further support that these distributions are not appropriate.



*Figure 6.* Q-Q plot vs Weibeull Distributions

### **2.1.2 Danceability**

The second characteristic that was fit to a distribution was the danceability of the songs. The parameters that were found using the *fitdistrplus* library in R for the danceability distribution can be found in table 2.

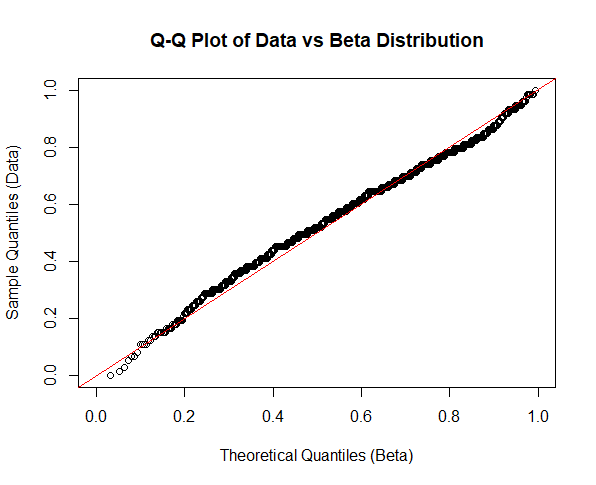
*Table 2:* Parameters of Distributions fit to Danceability Data

| Distribution | Fit Parameters | p-values |
| --- | --- | --- |
| Beta | shape1=2.46951, shape2=1.67329 | 0.003179 |
| Exponential | rate=1.660239 | < 2.2\*10-16 |
| Gamma | shape=5.126017,rate=8.507536 | 5.334\*10-12 |
| Weibull | shape = 3.258829, scale = 0.665768 | 1.383\*10-6 |
| Normal | mean=0.6023229, sd=0.2003141 | 3.019\*10-5 |

The p-values for the Kolmogorov Test for each fit distribution with distribution of songs’ danceability is found in table 2. The alpha-level of our test is 0.05. Therefore, for each of the tests, we reject the null hypothesis that the fit dataset matches that of the danceability dataset itself and have evidence to support that they are significantly different.

The p-value of the respective Anderson-Darling test was found to be 2.398\*10-12 which means that on the 0.05 alpha-level we reject the null hypothesis that the song danceability dataset is normally distributed.

We then plotted a QQ plot of the data against the best fit distribution, the weibull distribution. Figure 7 emphasizes how the data do not follow the line y=x and provides further support that these distributions are not appropriate.

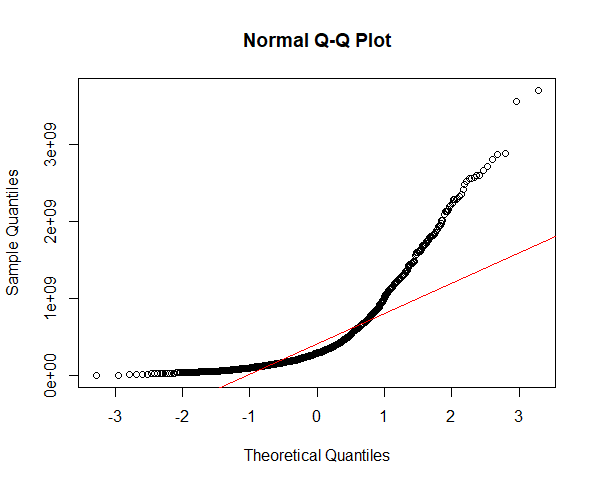


*Figure 7: Q-Q Plot of Danceability Data against Fit Beta Distribution*

### **2.1.3 Normality of Streams**

Our goal is to identify how the different characteristics impact the popularity of songs which is indicated by the number of streams. Given that this value is an integer it cannot be fit to a distribution as the other characteristics can be. We still want to know if the number of streams of songs is normally distributed or not. Therefore, we performed an Anderson Darling test on the dataset of the number of streams of songs. The null hypothesis is that the number of streams is normally distributed and the alternate hypothesis is that it is not normally distributed. The test returned that the p-value is less than 2.2\*10-16 which means that on the 0.05 alpha-level we reject the null hypothesis meaning that the number of streams of a song is not normally distributed.

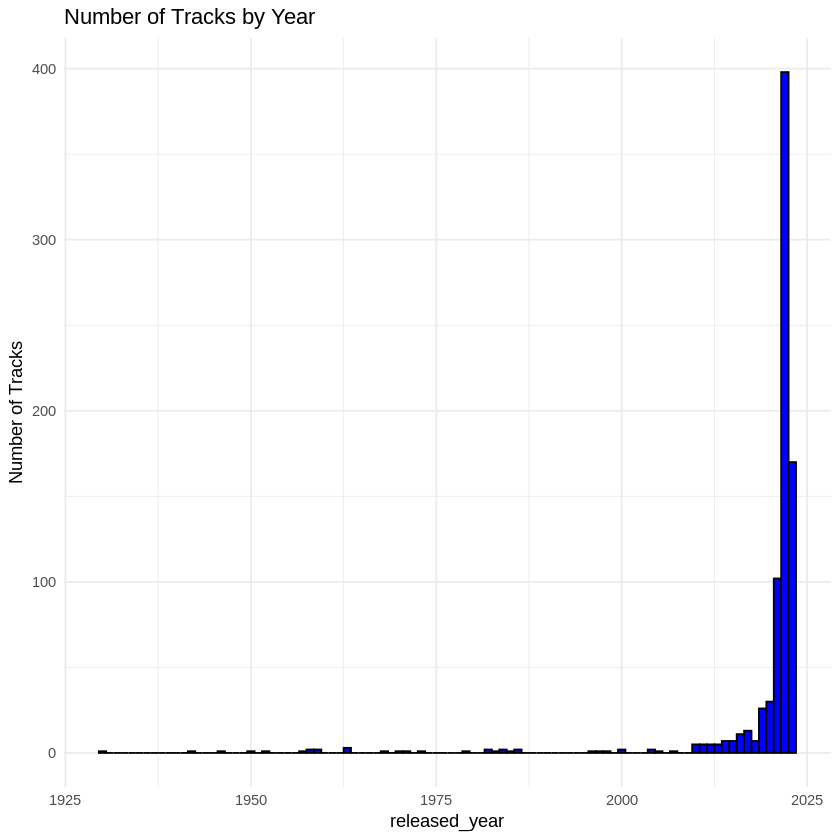
We then plot this data in a Q-Q plot to confirm our findings. Figure 8 shows the Q-Q plot and further supports our findings that the number of streams is not normally distributed due to how the quantile points stray far away from the y=x line that the points would follow if they were the same distribution.



*Figure 8:* Q-Q Plot of Streams against Fit Normal Distribution

## **2.2 Comparing Distributions of Years and Streams**

Based on the data we decided to try to categorize tracks by its associated year of release. We want to compare the distributions of the number of streams by the different years. First we analyzed the frequency of tracks by year through plotting them on a histogram. From the histogram below we can see that there are a lot more tracks in the later years so that is how we decided to choose which years to compare.



*Figure 9:* Histogram of tracks by year

We decided to compare 2021,2022, and 2023 as these were the years we had the most data for.

A Smirnovs test can be done to test the distributions of two different populations. We performed 3 different Smirnovs tests to compare the different distributions of the different years using an alpha of 0.05 the results are shown in the Table 3 below.

*Table 3:* Smirnov’s Test Results

| Years | Null Hypothesis | Alternative Hypothesis | P-value |
| --- | --- | --- | --- |
| 2021-2022 | The year 2021 has the same distribution of streams compared to the year 2022 | At least one group has a significantly different distribution of streams. | 1.438e-14 |
| 2021-2023 | The year 2021 has the same distribution of streams compared to the year 2023 | At least one group has a significantly different distribution of streams. | 2e-16 |
| 2022-2023 | The year 2022 has the same distribution of streams compared to the year 2023 | At least one group has a significantly different distribution of streams. | 2e-16 |

Based on the Smirnovs test for comparing 2021 streams to 2022 streams we found a p-value of 1.438e-14 which is less than our alpha of 0.05 so we would reject H0 meaning that these two years do not have the same distribution of streams. We did the same thing to compare the distribution for 2021 streams to 2023 streams. We found a p-value of 2.2e-16 which is less than alpha of 0.05 so we reject H0 meaning that these two years do not have the same distribution of streams. We did the same thing to compare the distribution for 2022 streams to 2023 streams. We found a p-value of 2.2e-16 which is less than alpha of 0.05 so we reject H0 meaning that these two years do not have the same distribution of streams.

**Comparing Median for Different years**

To compare the median number of streams for different years we did a kruskal-wallis test with an alpha of 0.05.

The hypothesis would be:

**Null Hypothesis (H0​):**The medians of streams are the same across the years 2021,2022, and 2023.

**Alternative Hypothesis (H1​):**At least one group has a significantly different median streams.

From our test we found a p-value of 2.2e-16 which is less than alpha = 0.05 thus we would reject the H0 meaning that there are different medians for different years.

After performing the Kruskal-Wallis Test we decided to use a Conover’s test to discuss streaming differences among the years. The results from Conover's test are shown in the table below. The null hypothesis for the concovers test would be that the media streams for the year are equal to the comparison year. The alternative hypothesis is that the years that are being compared do not have an equal median.

*Table 4:* Conover's Test Results

| Years | mean.rank.difference | P-value |
| --- | --- | --- |
| 2021-2022 | -162.0695 | 2e-16 \*\*\* |
| 2021-2023 | -321.2461 | 2e-16 \*\*\* |
| 2022-2023 | -159.1766 | 2e-16 \*\*\* |

Signif. Codes:0 ‘\*\*\*’

From the pvalues we would reject H0 using an alpha of 0.05. This implies that the different groups do not have equal variances meaning that parametric ANOVA will not work and that our data needs to be evaluated using nonparametric methods.

**2.3 ANOVA**

Parametric Analysis:

Based on the ANOVA for comparison between danceability and streams, it was deduced that at the 0.05 significance level at least one category of **danceability\_cat** differs significantly in its mean **streams.** Based on this, we moved forward in determining which group differs significantly by conducting the Tukey’s HSD test. Tukey's Honest Significant Difference (HSD) test, is a post-hoc pairwise comparison test used after an ANOVA to determine which specific groups differ significantly.

The Tukey’s HSD test showed us that there is a difference in high vs low danceability (high being over 67% danceability vs low of below 33% danceability):

* + High **danceability** leads to fewer **streams** compared to Low danceability based on the pvalue of 0.044 which is less than the significance level of 0.05
  + **Medium vs Low**: Not significant (p=0.982).
  + **High vs Medium**: Borderline significant (p=0.072).

It is important to note that this method is NOT accurate for the scope of this project due to the fact that the Shapiro Wilk normality test rejected the null hypothesis H0 at the 0.05 level which suggests that the **streams** variable does not follow a normal distribution. The parametric method simply assumes normality and will help in understanding the differences between parametric and nonparametric methods. Therefore, we move to nonparametric methods for analyzing differences in metrics for streams.

Nonparametric Analysis:

For the nonparametric approach, we assumed our hypothesis as:

**Null Hypothesis (H0​):**The medians of streams are the same across the danceability\_cat groups.

**Alternative Hypothesis (H1​):**At least one group has a significantly different median streams.

The results of the Kruskal Wallis Test for comparing the **danceability** variable with streams gave a pvalue=0.2341, which at the 0.05 significance level, is deemed as not statistically significant- rejecting the null hypothesis H0. This test shows that there is no difference in medians of **streams** across **danceability**.

So far we compared danceability vs streams due to the fact that danceability had the most significant impact on streams. However, from our analysis we found that there could be other variables that also confound the effect on streams. Therefore, we can conduct the Friedman Test.

Friedman Test:

We use the Friedman Test now to compare if streams differ significantly across various musical attributes for the same songs. First we extracted the musical attributes we are most interested in analyzing:**"bpm","key","mode","danceability","valence","energy","acousticness","instrumentalness","liveness","speechiness"**

Our null hypothesis H0 is that there is no significant difference in the distributions (ranks) of the variables. The alternative H1 hypothesis is that at least one variable differs significantly in its distribution compared to others. The Friedman statistic gave us a value of p=2.2e-16. This indicates that at least one of the variables has a significantly different rank distribution compared to the others.

To determine which variable is significantly different we do the pairwise Wilcoxon Sum Rank test. The analysis highlights significant differences in rank distributions across musical attributes. This suggests that certain attributes, such as **danceability**, **valence**, and **speechiness**, may contribute uniquely to variability in the data due to their low p values of p<2×10−16, especially when considering their potential relationship to streams. Factors like bpm and sqrt\_liveness, however, appear to align more closely with other variables and may contribute less to overall variability. **Danceability** and **valence** are emotionally and energetically descriptive variables, and their distinctiveness suggests they may be stronger predictors of engagement (streams) than others. Furthermore, **key** also has a low p value indicating a significant difference from the others. The significant differences shown by **key** suggests that the tonal structure of a song plays a noticeable role in its rank distribution.

**Theil Sen Regression**

We can now take the top musical attribute contributors and further analyze how they affect streams. Specifically, we use Theil-Sen regression to examine how **danceability**, **valence**, **key**, and **speechiness** relate to streams. Theil-Sen regression is a nonparametric robust regression technique that estimates the median slope between each predictor and the response variable, making it resistant to outliers.

From the Theil-Sen regression results, we can interpret the following about each of the predictors:

* **Danceability**:

For every unit increase in danceability, the expected median decrease in streams is ~50 million and with a p value of 0.00116, it has a statistically significant relationship with streams. Despite being a common metric for "dance-friendliness," higher danceability may not favorably impact streams. This could reflect audience preferences for more balanced or complex compositions.

* **Valence**:

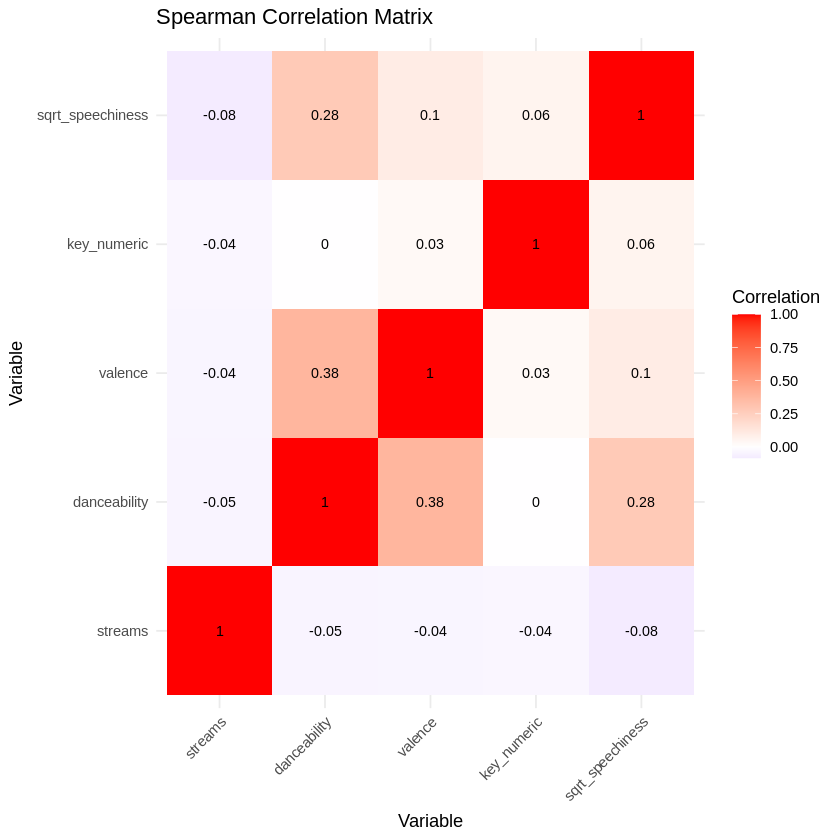
For every unit increase in valence (positivity of a song), the expected median decrease in streams is ~33 million and with a p value of 0.00108, it also has a significant relationship with streams. Songs with higher emotional positivity may not resonate as strongly with listeners on Spotify, who might prefer more emotionally low or darker songs.

* **Key**:

For a one-unit increase in key\_numeric (tonal key), the expected median decrease in streams is ~3.5 million, with a p value of 0.00341 which is significant. While the effect is smaller than danceability or valence, certain keys may not be as popular among people. This could be due to listener preferences for certain tonalities.

* **Speechiness**:

For every unit increase in speechiness (proportion of spoken words), the expected median decrease in streams is ~116 million, and with a very low p value of 0.0000447 it is highly significant. Songs with a higher proportion of spoken words (like rap or dialogue-heavy tracks) significantly reduce streams, suggesting audience preference for more melodic or sung compositions.



*Figure 10*: Spearman Correlation Matrix

To further complement the Theil-Sen regression analysis, we can look at Figure 10, the Spearman COrrelation Matrix, which visually highlights how variables relate to one another and to streams. Strong correlations (positive or negative) highlight which attributes most influence streams and also reveal potential collinearity. While Theil-Sen provides regression coefficients for the relationship between predictors and streams, the The Spearman correlation matrix reveals weak monotonic relationships between streams and musical attributes such as danceability and speechiness. This suggests that these attributes do not have a consistent directional relationship with streams across their entire range. However, significant effects observed in Theil-Sen regression and nonparametric tests indicate the presence of localized or nonlinear interactions that Spearman correlation may miss.

## **3.0 Subject Matter Implications of Study**

The findings of this study have significant implications for the music industry, particularly in understanding the factors that contribute to the popularity of a song on Spotify. By leveraging nonparametric methods, we addressed the non-normality and heterogeneity in the data, allowing for robust insights into the relationships between musical attributes and song performance. The results challenge common assumptions about what makes a song popular and provide actionable insights for music producers, artists, and streaming platforms. Insights into how musical attributes influence streams can guide production decisions, marketing strategies, and the tailoring of music to target audiences. Understanding these relationships could help optimize playlist curation, track promotion, and recommendation systems.

## **4.0 Further questions**

After completing our study we thought of further questions that might be investigated in the future.

* How would genre impact the popularity of a track? We would need a different data set or to join datasets for genre in order to answer this question as our original dataset does not contain genre.
* Do all streaming services follow similar trends?
* How would the age of the listener effect the popularity trends?
* Investigating the impact of external factors like marketing, collaborations, and social media presence on song performance.

## **5.0 Conclusion**

This study aimed to explore the relationships between various musical attributes and the popularity of songs, as measured by streams on Spotify. By leveraging a combination of nonparametric and parametric methods, we gained insights into how these attributes influence song performance, while also addressing the challenges posed by non-normal and heterogeneous data. To make sure our dataset was not normal or fit to another distribution we assessed using goodness-of-fit tests, including the Kolmogorov-Smirnov test, Anderson-Darling test, and QQ-plots. Additionally, the similarity of distributions after segmenting the data by year was analyzed using the Smirnov test and Kruskal-Wallis test. Theil-Sen regression revealed significant negative relationships between streams and attributes like danceability, valence, speechiness, and key. Notably, speechiness had the strongest negative impact, indicating that songs with higher proportions of spoken words are less popular. Attributes like danceability and valence, traditionally associated with audience engagement, showed inverse relationships, suggesting a preference for more emotionally complex tracks. While parametric methods like ANOVA and Tukey’s HSD test provided some initial insights, their reliance on assumptions of normality and homogeneity of variance made them unsuitable for this dataset. Nonparametric methods such as the Friedman test and Theil-Sen regression provided robust and reliable results.

Contrary to the intuitive expectation that higher danceability and positivity (valence) would correlate with increased streams, the analysis shows significant negative relationships. This suggests that audiences may prefer tracks with more complexity or emotional depth over high-energy or overly positive songs. Artists and producers may need to reconsider how much emphasis they place on these attributes when crafting music intended for broad appeal. Further, the strong negative impact of speechiness on streams highlights audience preferences for melodic or sung compositions over spoken-word content. This insight could be vital for marketing strategies in genres like rap or spoken-word music, where blending melodic elements may help reach broader audiences. Streaming platforms like Spotify can incorporate these findings into their recommendation algorithms, giving more weight to attributes like danceability, valence, and speechiness when personalizing playlists or promoting new tracks.

A limitation we encountered was that the dataset lacks information on genre, listener demographics, and external factors like marketing efforts, which likely play significant roles in song popularity.

## **Appendix**

Link to Collab Code Notebook: <https://colab.research.google.com/drive/1iFFr3vmCNk4u_Pz3zvs-SyGiSfx_MNzZ?usp=sharing>

Note: Estella’s analysis wouldn’t upload to this so Goodness of fit tests code are shown separately in File ‘Project\_ETU.R’.

Connecting to Kaggle Data:

# IMPORTANT: RUN THIS CELL IN ORDER TO IMPORT YOUR KAGGLE DATA SOURCES

# TO THE CORRECT LOCATION (/kaggle/input) IN YOUR NOTEBOOK,

# THEN FEEL FREE TO DELETE THIS CELL.

# NOTE: THIS NOTEBOOK ENVIRONMENT DIFFERS FROM KAGGLE'S R

# ENVIRONMENT SO THERE MAY BE MISSING LIBRARIES USED BY YOUR

# NOTEBOOK.

DATA\_SOURCE\_MAPPING = 'spotify-most-streamed-songs:https%3A%2F%2Fstorage.googleapis.com%2Fkaggle-data-sets%2F5660867%2F9341051%2Fbundle%2Farchive.zip%3FX-Goog-Algorithm%3DGOOG4-RSA-SHA256%26X-Goog-Credential%3Dgcp-kaggle-com%2540kaggle-161607.iam.gserviceaccount.com%252F20241204%252Fauto%252Fstorage%252Fgoog4\_request%26X-Goog-Date%3D20241204T190336Z%26X-Goog-Expires%3D259200%26X-Goog-SignedHeaders%3Dhost%26X-Goog-Signature%'

KAGGLE\_INPUT\_PATH = '/kaggle/input'

KAGGLE\_WORKING\_PATH = '/kaggle/working'

system(paste0('sudo umount ', '/kaggle/input'))

system(paste0('sudo rmdir ', '/kaggle/input'))

system(paste0('sudo mkdir -p -- ', KAGGLE\_INPUT\_PATH), intern=TRUE)

system(paste0('sudo chmod 777 ', KAGGLE\_INPUT\_PATH), intern=TRUE)

system(

paste0('sudo ln -sfn ', KAGGLE\_INPUT\_PATH,' ',file.path('..', 'input')),

intern=TRUE)

system(paste0('sudo mkdir -p -- ', KAGGLE\_WORKING\_PATH), intern=TRUE)

system(paste0('sudo chmod 777 ', KAGGLE\_WORKING\_PATH), intern=TRUE)

system(

paste0('sudo ln -sfn ', KAGGLE\_WORKING\_PATH, ' ', file.path('..', 'working')),

intern=TRUE)

data\_source\_mappings = strsplit(DATA\_SOURCE\_MAPPING, ',')[[1]]

for (data\_source\_mapping in data\_source\_mappings) {

path\_and\_url = strsplit(data\_source\_mapping, ':')

directory = path\_and\_url[[1]][1]

download\_url = URLdecode(path\_and\_url[[1]][2])

filename = sub("\\?.+", "", download\_url)

destination\_path = file.path(KAGGLE\_INPUT\_PATH, directory)

print(paste0('Downloading and uncompressing: ', directory))

if (endsWith(filename, '.zip')){

temp = tempfile(fileext = '.zip')

download.file(download\_url, temp)

unzip(temp, overwrite = TRUE, exdir = destination\_path)

unlink(temp)

}

else{

temp = tempfile(fileext = '.tar')

download.file(download\_url, temp)

untar(temp, exdir = destination\_path)

unlink(temp)

}

print(paste0('Downloaded and uncompressed: ', directory))

}

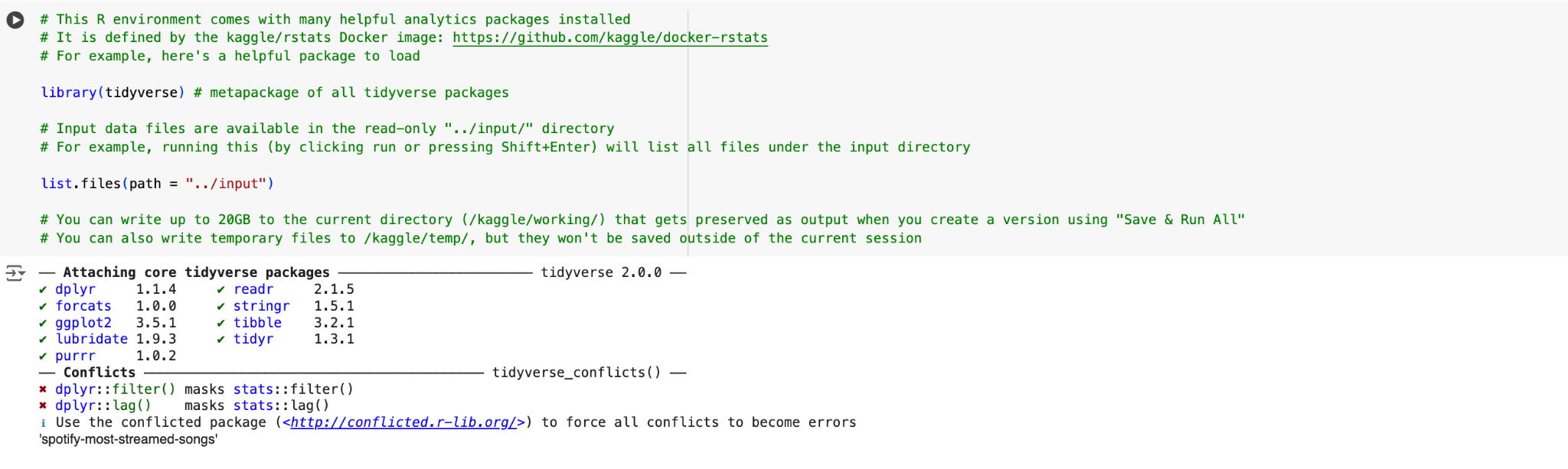
print(paste0('Data source import complete'))

[1] "Downloading and uncompressing: spotify-most-streamed-songs"

[1] "Downloaded and uncompressed: spotify-most-streamed-songs"

[1] "Data source import complete"

Setting Up R environment:



Reading in file

df <- read.csv('../input/spotify-most-streamed-songs/Spotify Most Streamed Songs.csv')

print(str(df))

'data.frame': 953 obs. of 25 variables:

$ track\_name : chr "Seven (feat. Latto) (Explicit Ver.)" "LALA" "vampire" "Cruel Summer" ...

$ artist.s.\_name : chr "Latto, Jung Kook" "Myke Towers" "Olivia Rodrigo" "Taylor Swift" ...

$ artist\_count : int 2 1 1 1 1 2 2 1 1 2 ...

$ released\_year : int 2023 2023 2023 2019 2023 2023 2023 2023 2023 2023 ...

$ released\_month : int 7 3 6 8 5 6 3 7 5 3 ...

$ released\_day : int 14 23 30 23 18 1 16 7 15 17 ...

$ in\_spotify\_playlists: int 553 1474 1397 7858 3133 2186 3090 714 1096 2953 ...

$ in\_spotify\_charts : int 147 48 113 100 50 91 50 43 83 44 ...

$ streams : chr "141381703" "133716286" "140003974" "800840817" ...

$ in\_apple\_playlists : int 43 48 94 116 84 67 34 25 60 49 ...

$ in\_apple\_charts : int 263 126 207 207 133 213 222 89 210 110 ...

$ in\_deezer\_playlists : chr "45" "58" "91" "125" ...

$ in\_deezer\_charts : int 10 14 14 12 15 17 13 13 11 13 ...

$ in\_shazam\_charts : chr "826" "382" "949" "548" ...

$ bpm : int 125 92 138 170 144 141 148 100 130 170 ...

$ key : chr "B" "C#" "F" "A" ...

$ mode : chr "Major" "Major" "Major" "Major" ...

$ danceability\_. : int 80 71 51 55 65 92 67 67 85 81 ...

$ valence\_. : int 89 61 32 58 23 66 83 26 22 56 ...

$ energy\_. : int 83 74 53 72 80 58 76 71 62 48 ...

$ acousticness\_. : int 31 7 17 11 14 19 48 37 12 21 ...

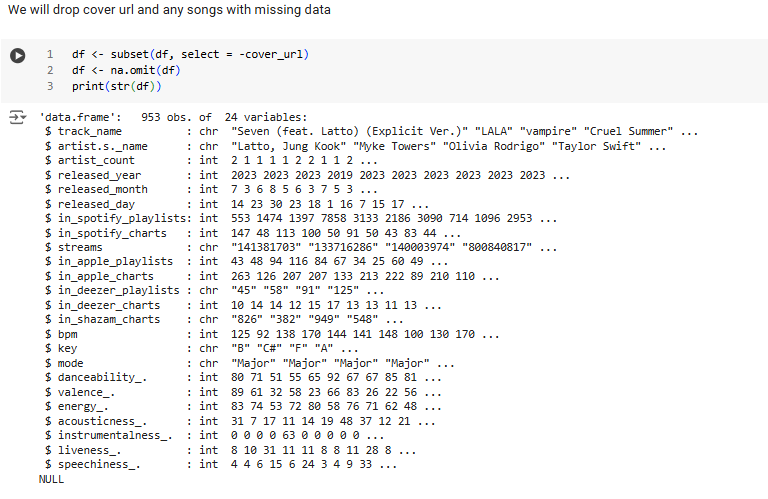
$ instrumentalness\_. : int 0 0 0 0 63 0 0 0 0 0 ...

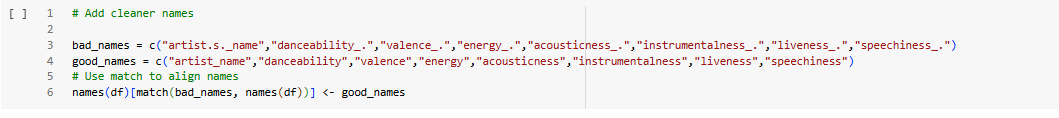
$ liveness\_. : int 8 10 31 11 11 8 8 11 28 8 ...

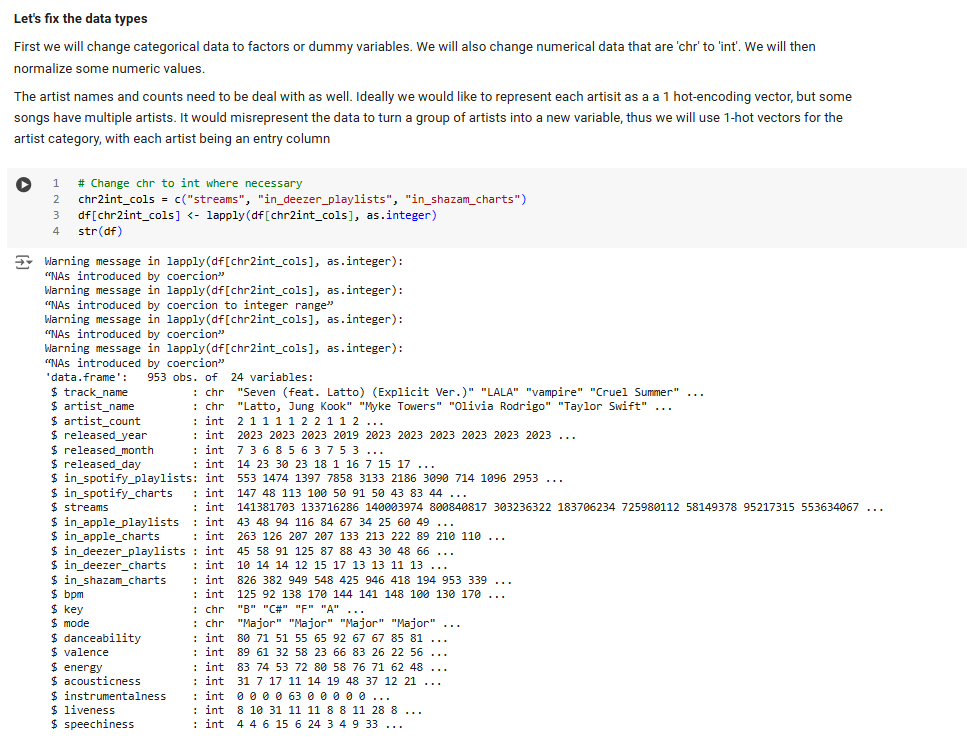
$ speechiness\_. : int 4 4 6 15 6 24 3 4 9 33 ...

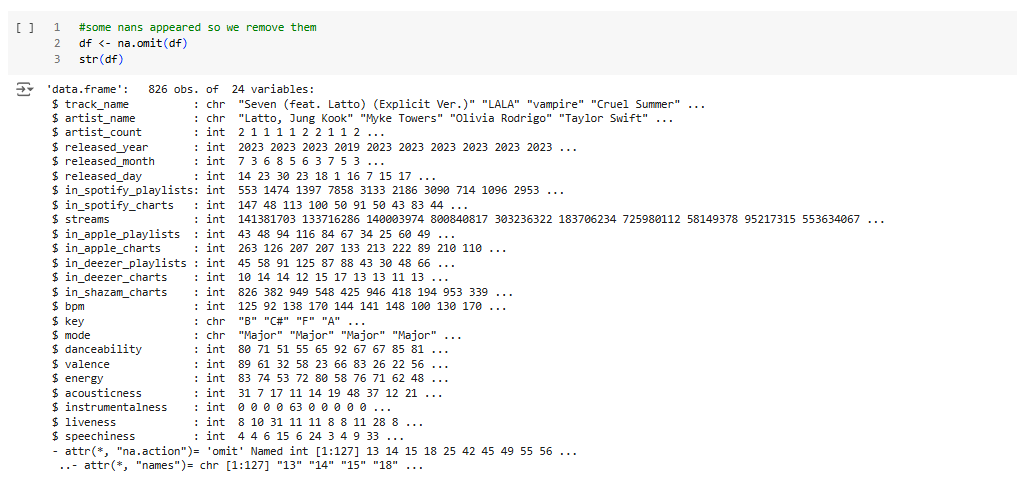
$ cover\_url : chr "Not Found" "<https://i.scdn.co/image/ab67616d0000b2730656d5ce813ca3cc4b677e05>" "<https://i.scdn.co/image/ab67616d0000b273e85259a1cae29a8d91f2093d>" "<https://i.scdn.co/image/ab67616d0000b273e787cffec20aa2a396a61647>" ...

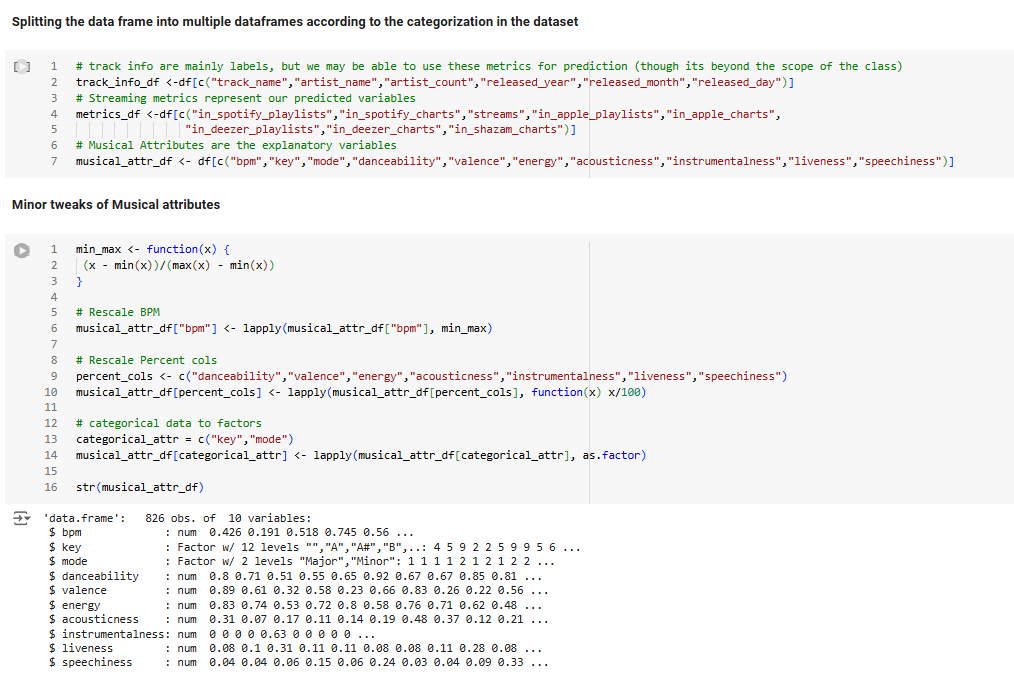
NULL

CODE for Data Cleaning 

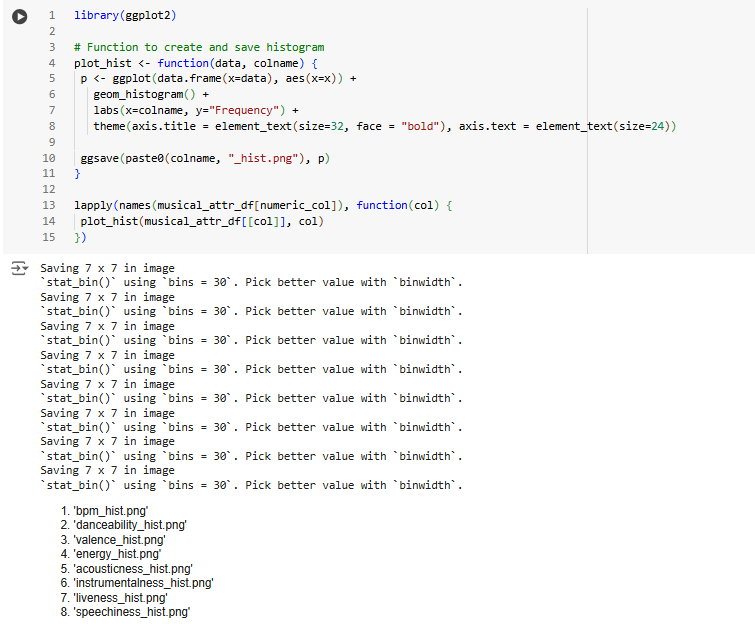
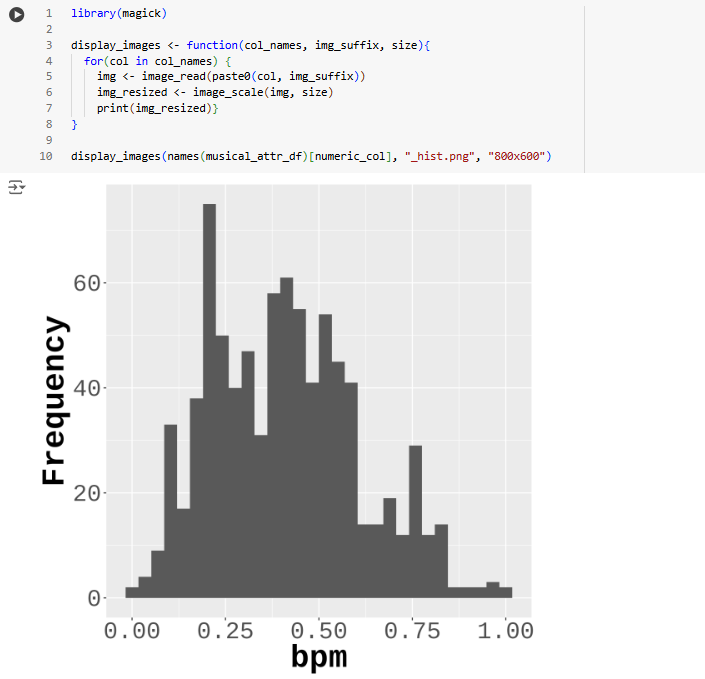




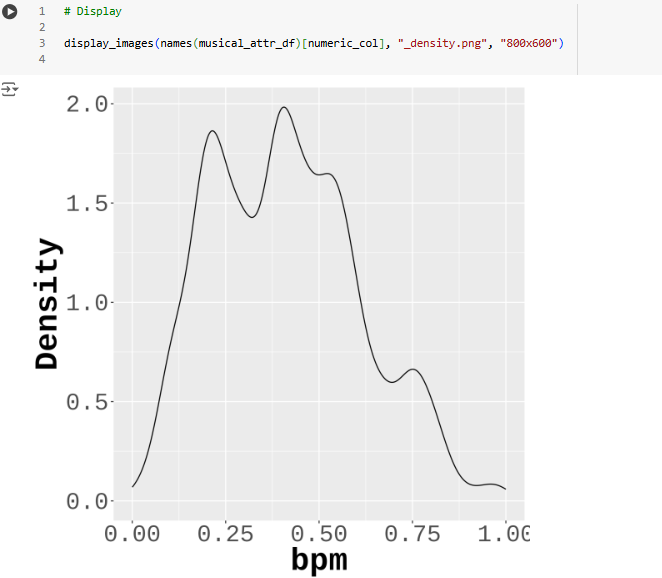


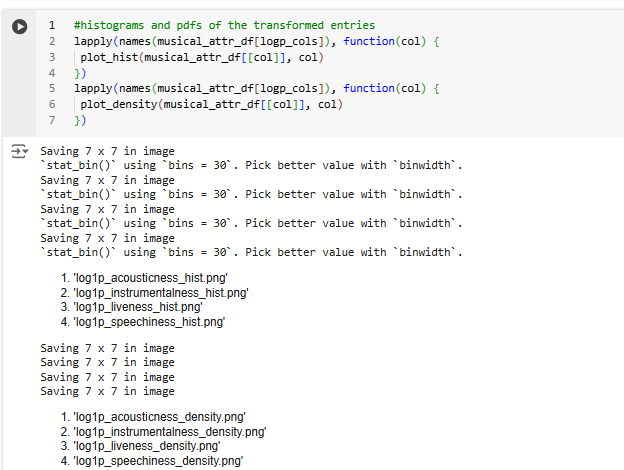


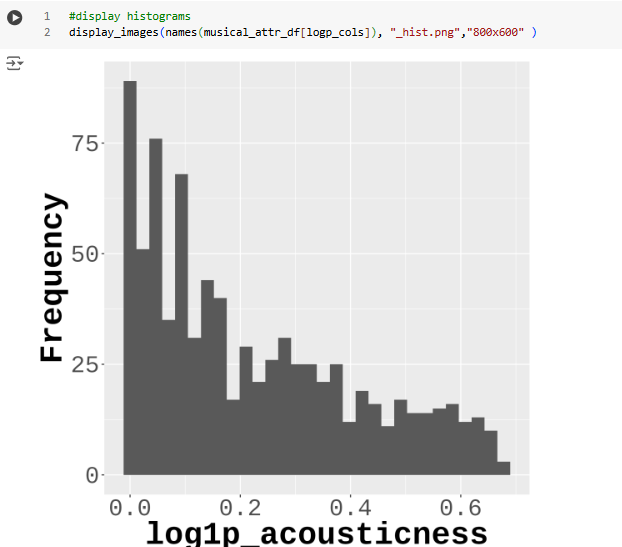


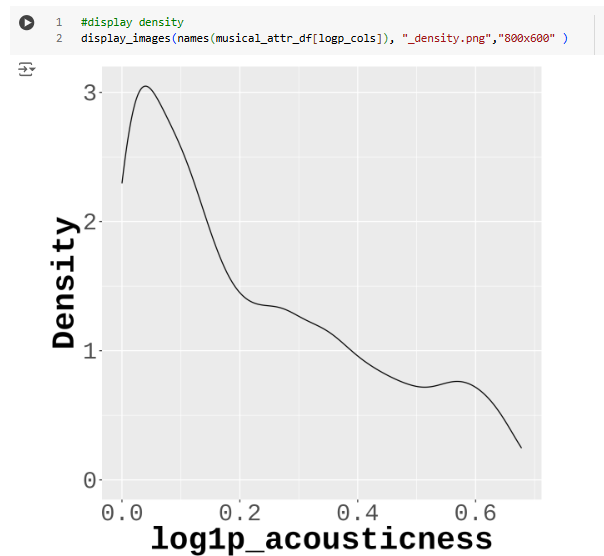


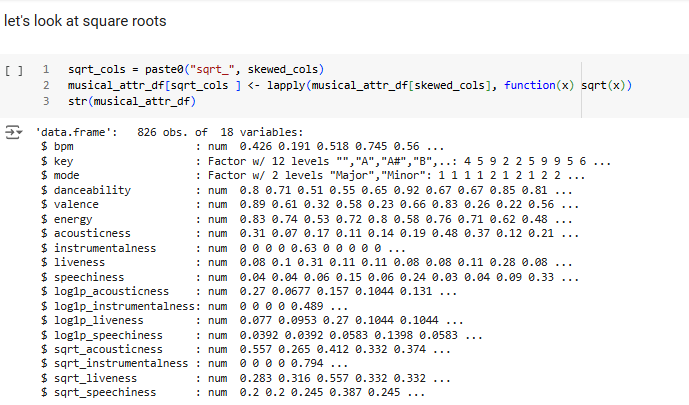


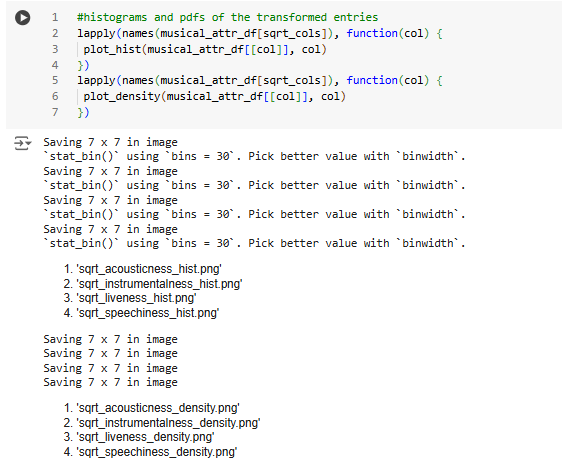


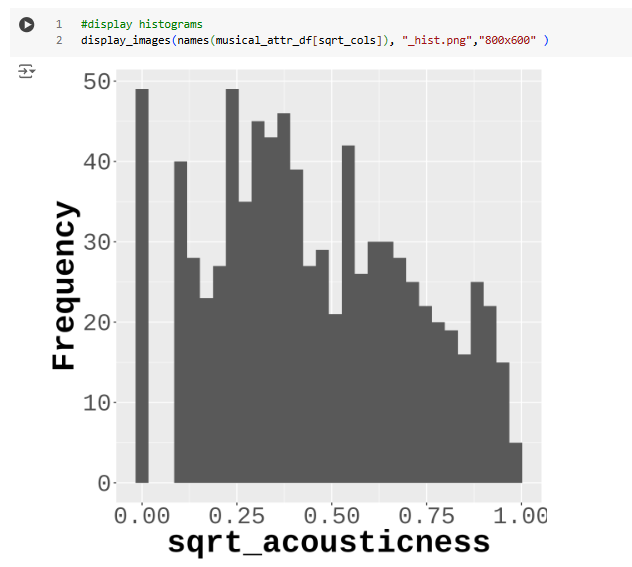


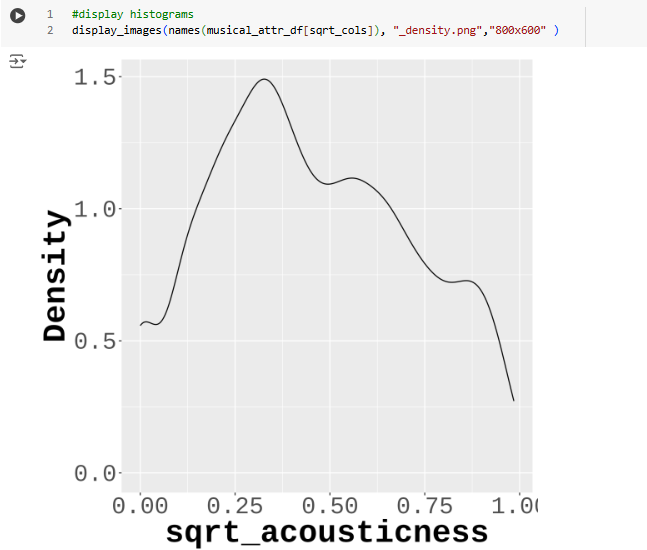


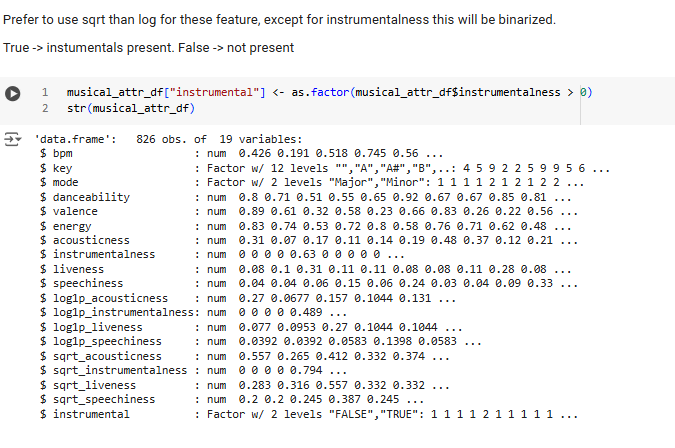


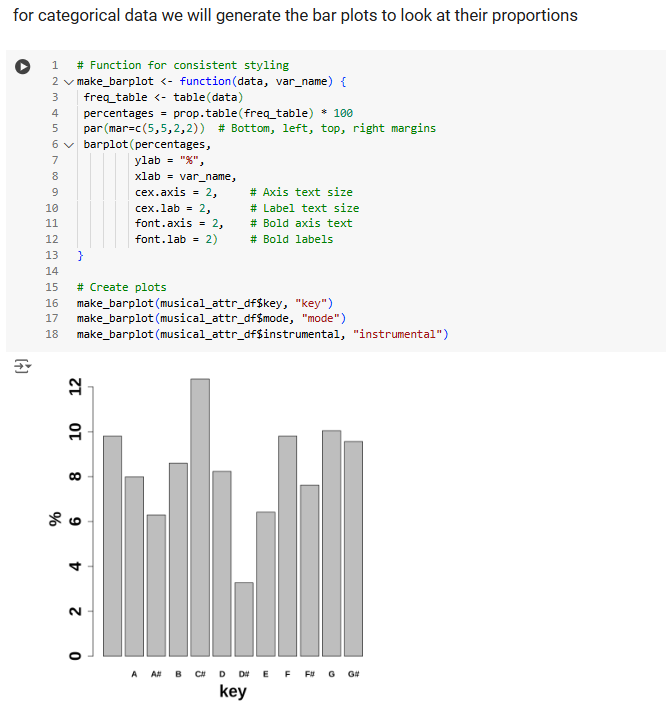












Code for Goodness of Fit Tests

install.packages("fitdistrplus")

library(fitdistrplus)

library(nortest)

######################################################################

#Danceability

# Example data (replace this with your actual dataset)

dance\_data <- spotify$danceability

# Normalize data if not in [0, 1]

norm\_dance\_data <- (dance\_data - min(dance\_data)) / (max(dance\_data) - min(dance\_data))

norm\_dance\_data <- ifelse(norm\_dance\_data == 0, 1e-10, norm\_dance\_data)

norm\_dance\_data <- ifelse(norm\_dance\_data == 1, 1 - 1e-10, norm\_dance\_data)

# Fit beta distribution using fitdistrplus

fitdist(norm\_dance\_data, "beta", method = "mle")

#shape1=2.46951, shape2=1.67329

fitdist(norm\_dance\_data, "exp", method = "mle")

#rate=1.660239

fitdist(norm\_dance\_data, "gamma", method = "mle")

#shape=5.126017

fitdist(norm\_dance\_data, "weibull", method = "mle")

#shape = 3.258829, scale = 0.665768

fitdist(norm\_dance\_data, "norm", method = "mle")

#mean=0.6023229, sd=0.2003141

# Display results

summary(fit)

ks.test(norm\_dance\_data,"pbeta",shape1=2.46951, shape2=1.67329)

#p-value = 0.003179

ks.test(norm\_dance\_data,"pexp",rate=1.660239)

#p-value < 2.2e-16

ks.test(norm\_dance\_data,"pgamma",shape=5.126017,rate=8.507536)

#p-value = 5.334e-12

ks.test(norm\_dance\_data,"pweibull",shape = 3.258829, scale = 0.665768)

#p-value = 1.383e-06

ks.test(norm\_dance\_data,"pnorm", mean=0.6023229, sd=0.2003141)

#p-value = 3.019e-05

ad.test(spotify$danceability)

#p-value = 2.398e-12

# Generate theoretical quantiles from a beta distribution

theoretical\_quantiles <- qbeta(ppoints(length(norm\_dance\_data)), shape1 = 2.46951, shape2 = 1.67329)

# Sort the data

sorted\_data <- sort(norm\_dance\_data)

# Create the Q-Q plot with x-axis limits set to [0, 1]

qqplot(theoretical\_quantiles, sorted\_data,

main = "Q-Q Plot of Data vs Beta Distribution",

xlab = "Theoretical Quantiles (Beta)",

ylab = "Sample Quantiles (Data)",

xlim = c(0, 1))

abline(0, 1, col = "red") # Add a reference line

######################################################################

#bpm

# Example data (replace this with your actual dataset)

bpm\_data <- spotify$bpm

# Normalize data if not in [0, 1]

norm\_bpm\_data <- (bpm\_data - min(bpm\_data)) / (max(bpm\_data) - min(bpm\_data))

norm\_bpm\_data <- ifelse(norm\_bpm\_data == 0, 1e-10, norm\_bpm\_data)

norm\_bpm\_data <- ifelse(norm\_bpm\_data == 1, 1 - 1e-10, norm\_bpm\_data)

# Fit beta distribution using fitdistrplus

fitdist(norm\_bpm\_data, "beta", method = "mle")

fitdist(norm\_bpm\_data, "exp", method = "mle")

fitdist(norm\_bpm\_data, "gamma", method = "mle")

fitdist(norm\_bpm\_data, "weibull", method = "mle")

fitdist(norm\_bpm\_data, "norm", method = "mle")

ks.test(norm\_bpm\_data,"pbeta",shape1=1.580601, shape2=2.199773)

#p-value = 5.573e-06

ks.test(norm\_bpm\_data,"pexp",rate=2.450452)

#p-value < 2.2e-16

ks.test(norm\_bpm\_data,"pgamma",shape=2.84181,rate=6.96323)

#p-value = 2.942e-07

ks.test(norm\_bpm\_data,"pweibull",shape = 2.0590890, scale = 0.4564433)

#p-value = 0.004624

ks.test(norm\_bpm\_data,"pnorm", mean = 0.4080879, sd = 0.1988871)

#p-value = 0.002762

ad.test(spotify$bpm)

#p-value = 7.927e-13

# Generate theoretical quantiles from a beta distribution

bpm\_quantiles <- qweibull(ppoints(length(norm\_bpm\_data)), shape = 2.0590890, scale = 0.4564433)

# Sort the data

bpm\_data\_sorted <- sort(norm\_bpm\_data)

# Create the Q-Q plot with x-axis limits set to [0, 1]

qqplot(bpm\_quantiles, bpm\_data\_sorted,

main = "Q-Q Plot of Data vs Fit Weibull Distribution",

xlab = "Theoretical Quantiles (Weibull)",

ylab = "Sample Quantiles (Data)",

xlim = c(0, 1))

abline(0, 1, col = "red") # Add a reference line

######################################################################

#Streams

spotify$streams <- as.numeric(spotify$streams)

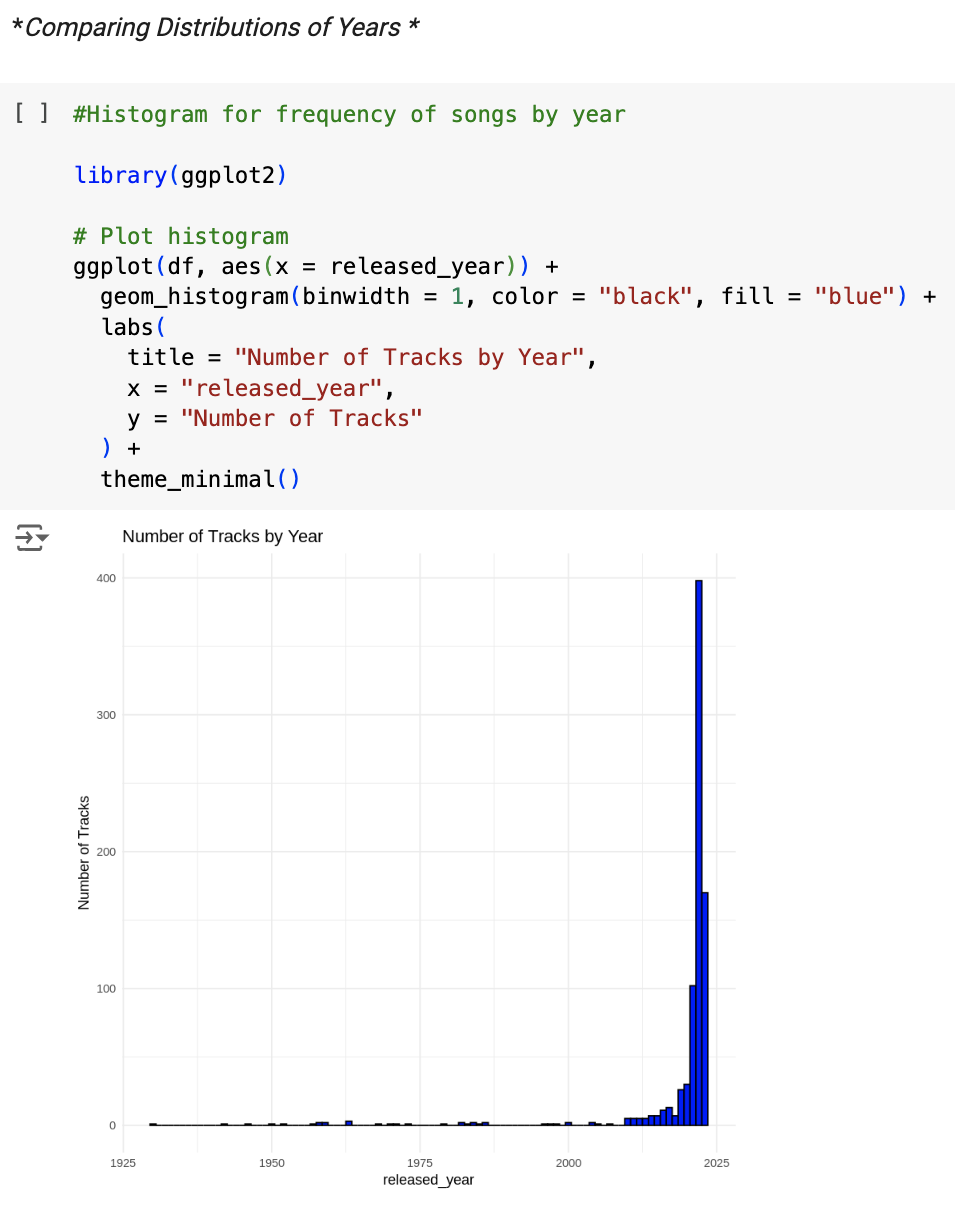
ad.test(spotify$streams)

#p-value < 2.2e-16

qqnorm(spotify$streams)

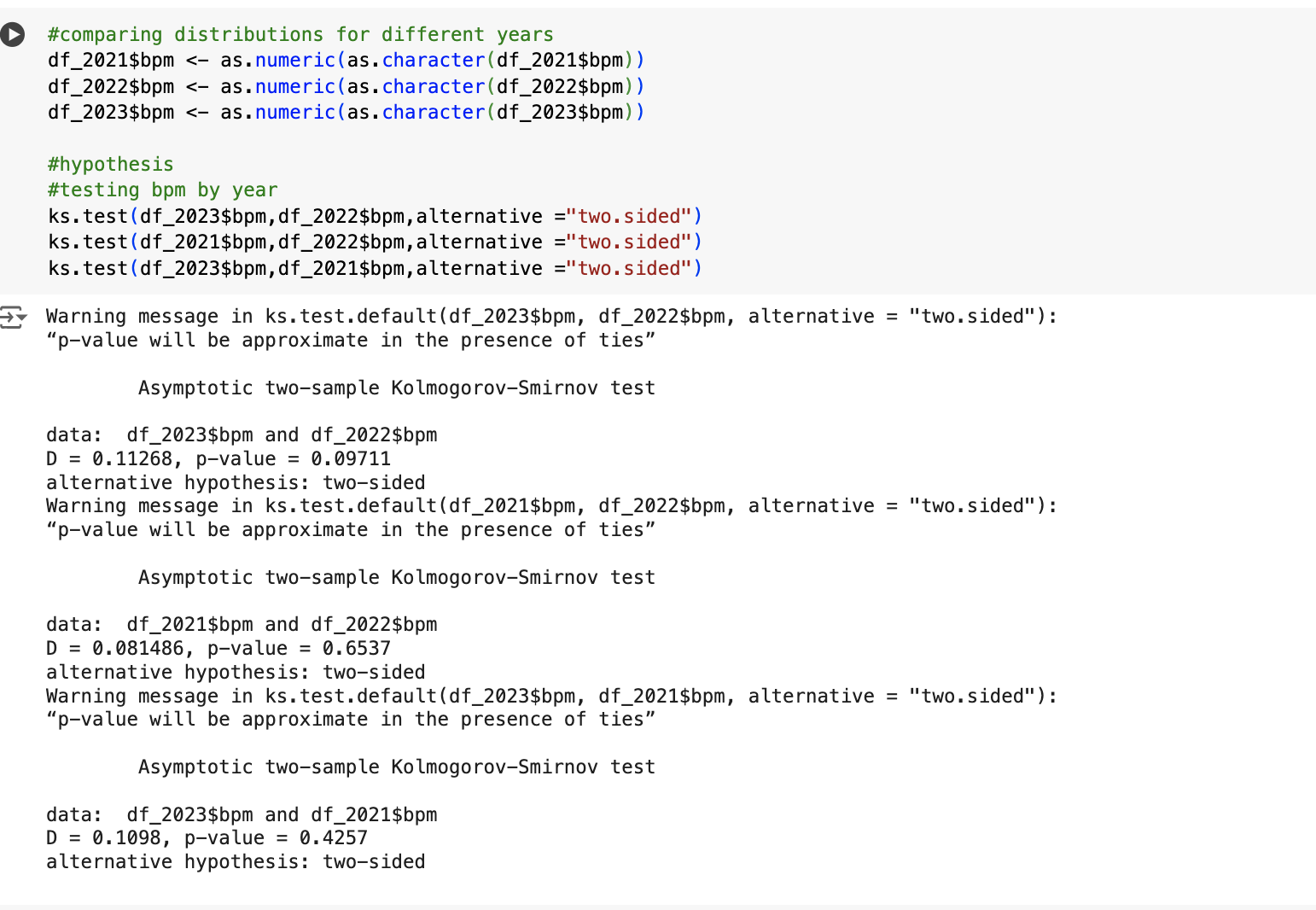
qqline(spotify$streams, col = "red")

**Distribution Testing (Section 2.2 Code)**

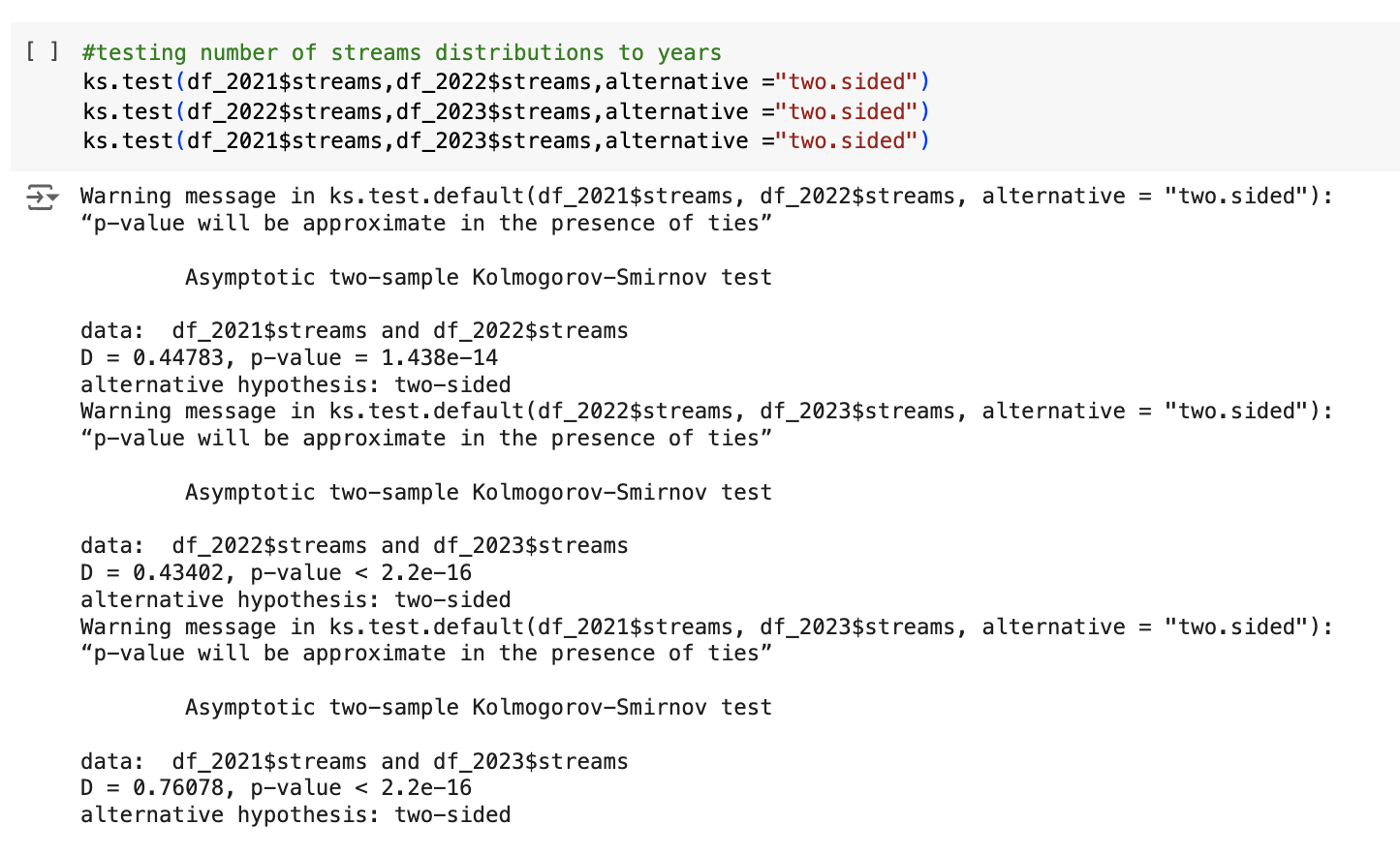




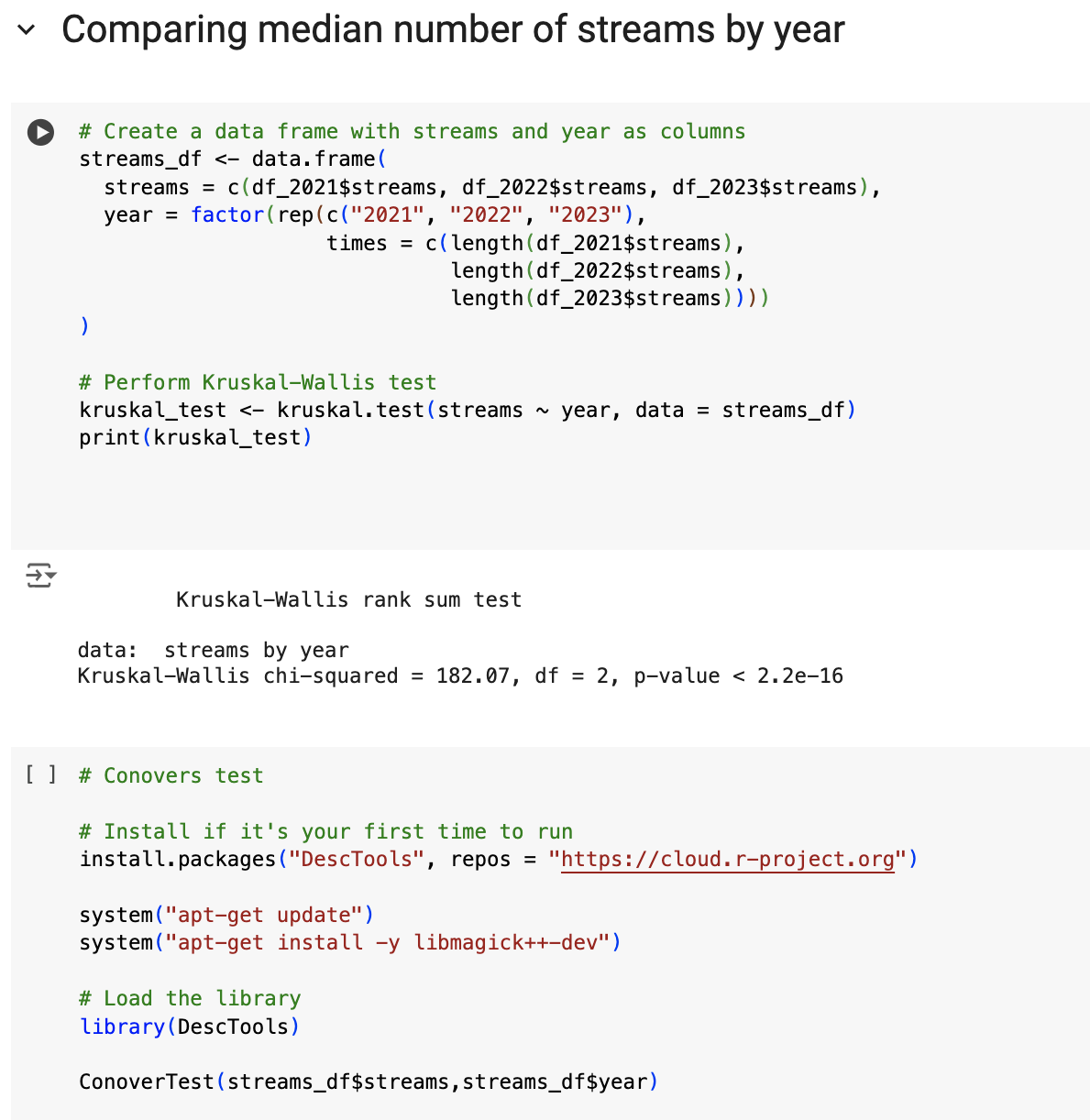
**Testing Year and Bpm Analysis**

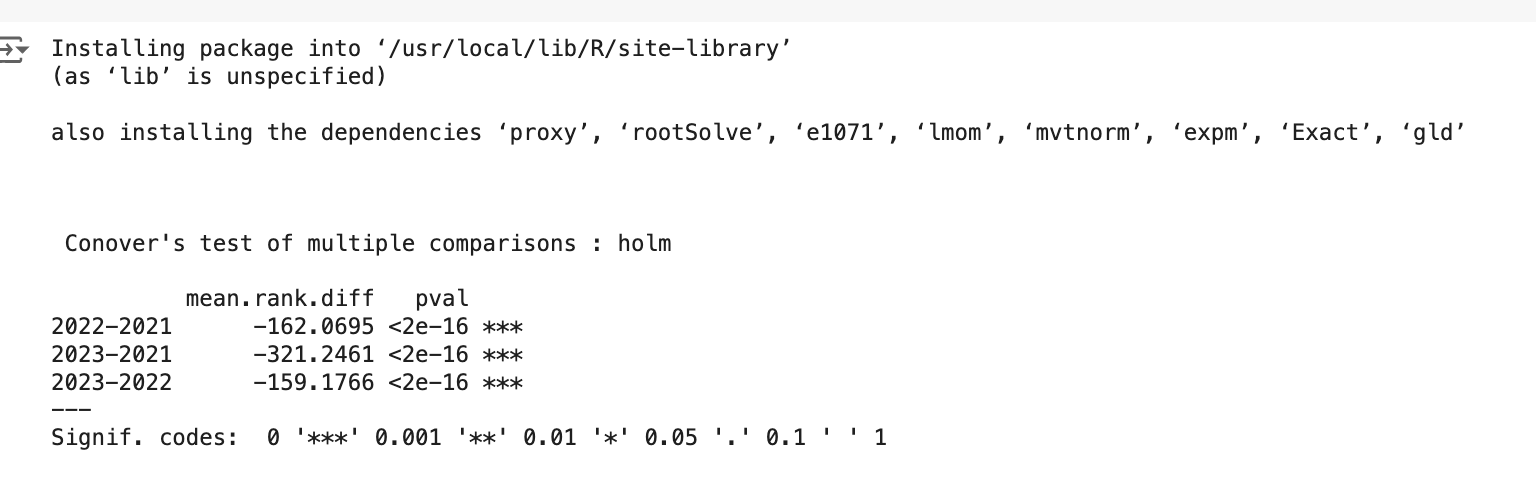


**Testing Number of Streams by year**

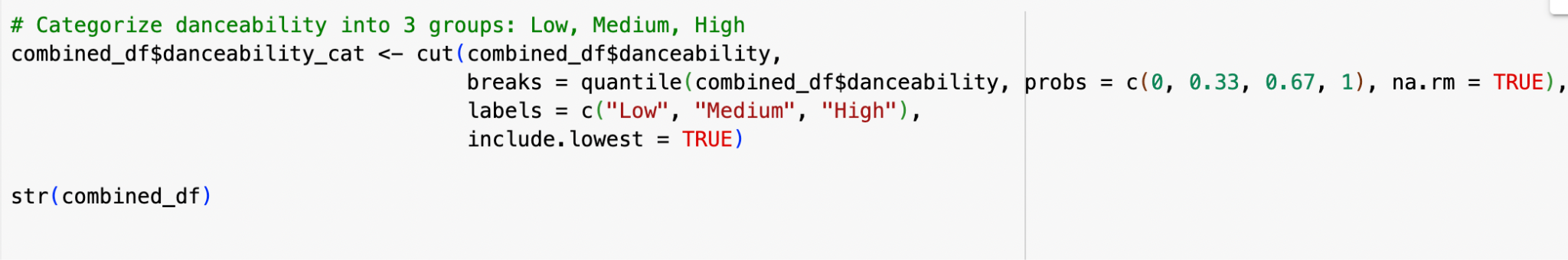


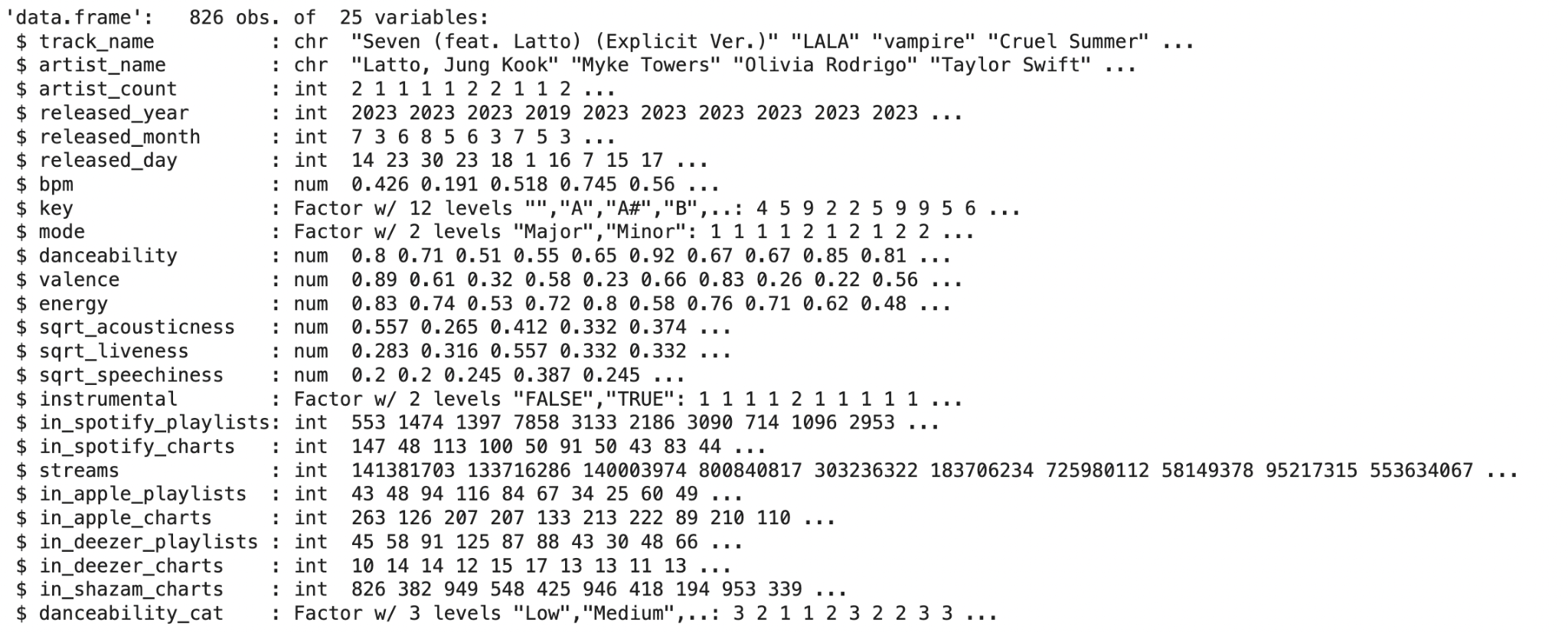
**Comparing Median number of streams by year**



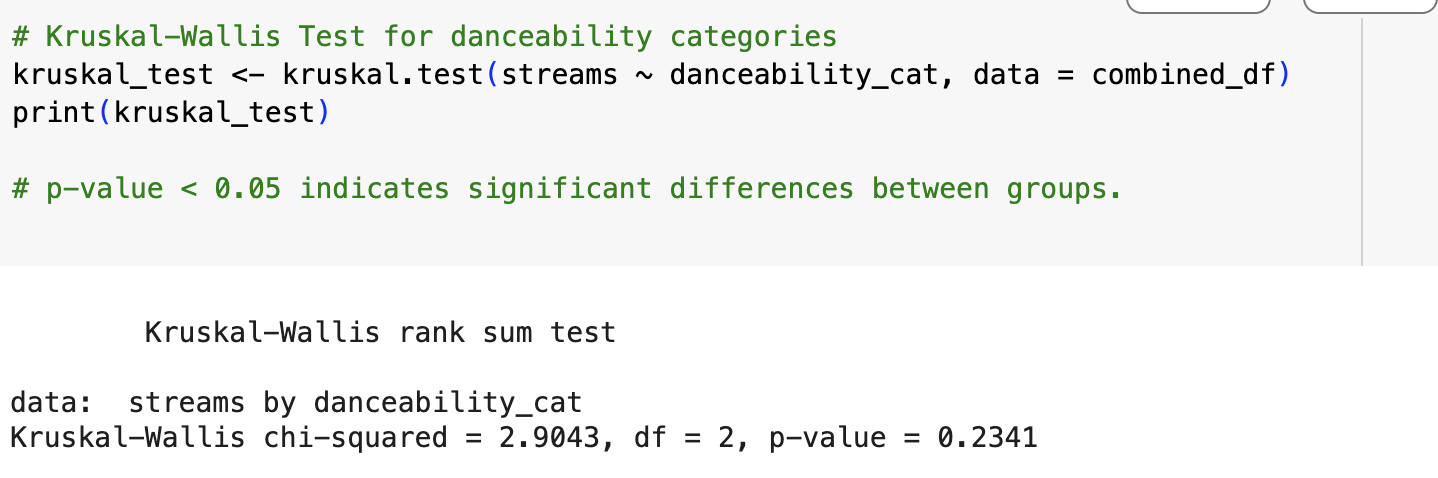


**Danceability Analysis**

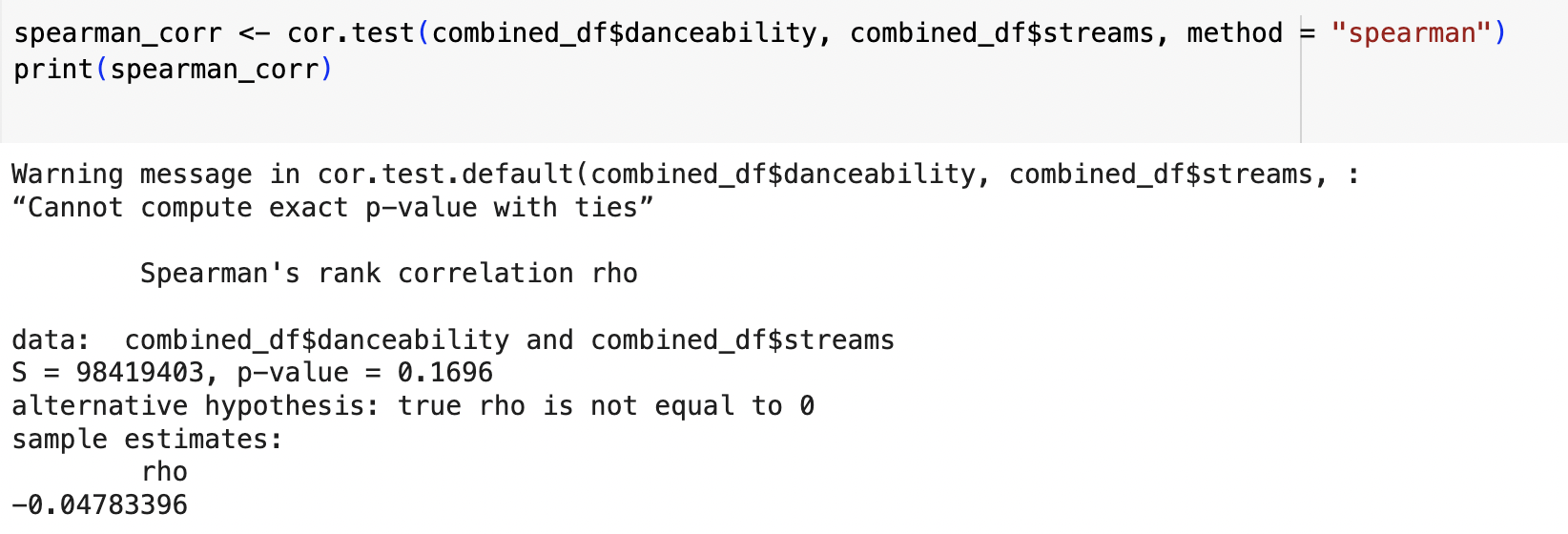




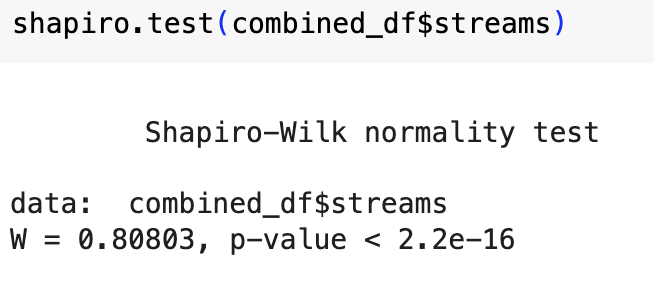
**Kruskal Wallis Test for Danceability**



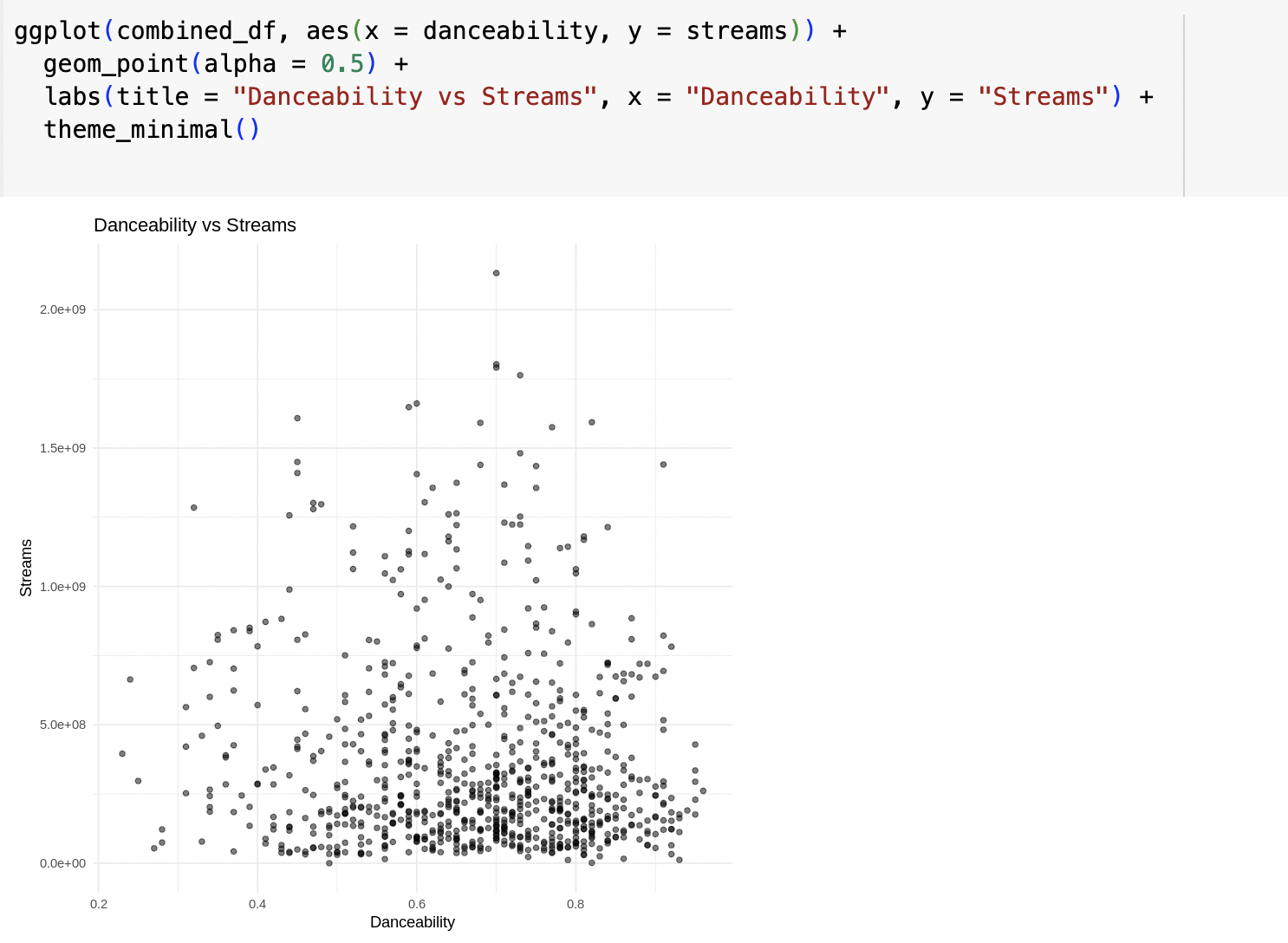
**Spearman Correlation for Danceability vs Streams**



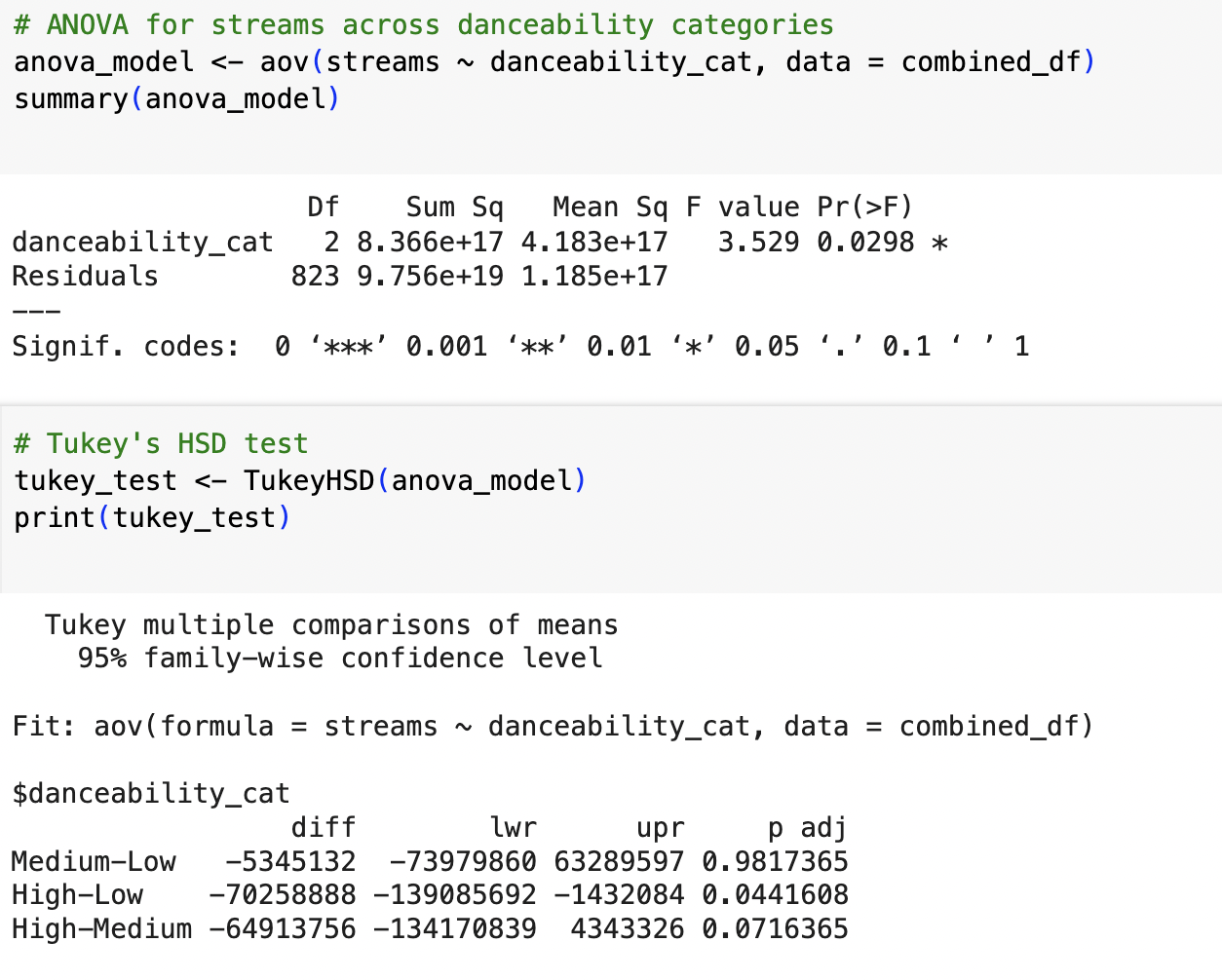
**Shapiro Wilk Normality test for Streams**



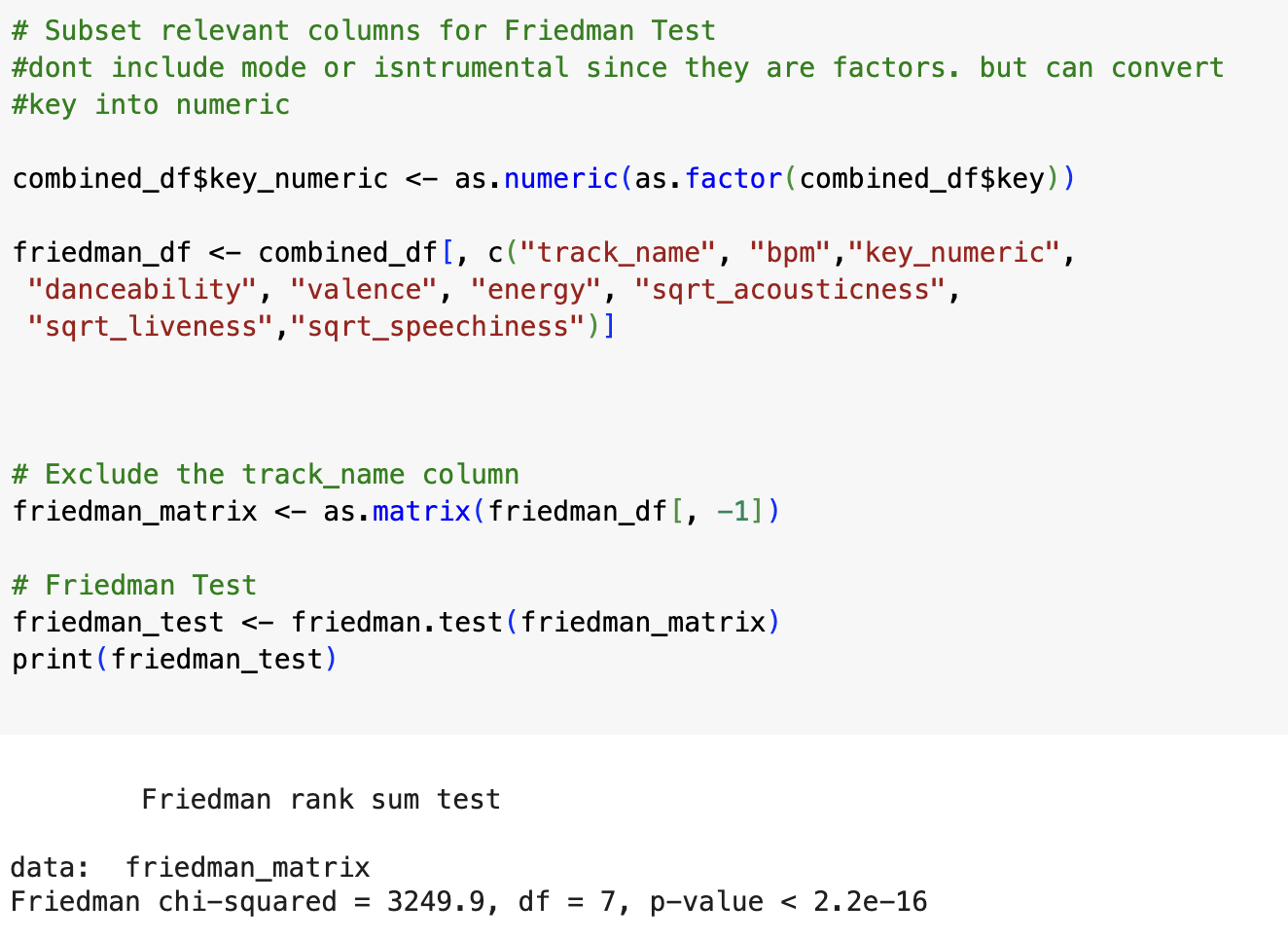
**Plot of danceability vs Streams**



**Parametric ANOVA followed by Tukeys HSD test**



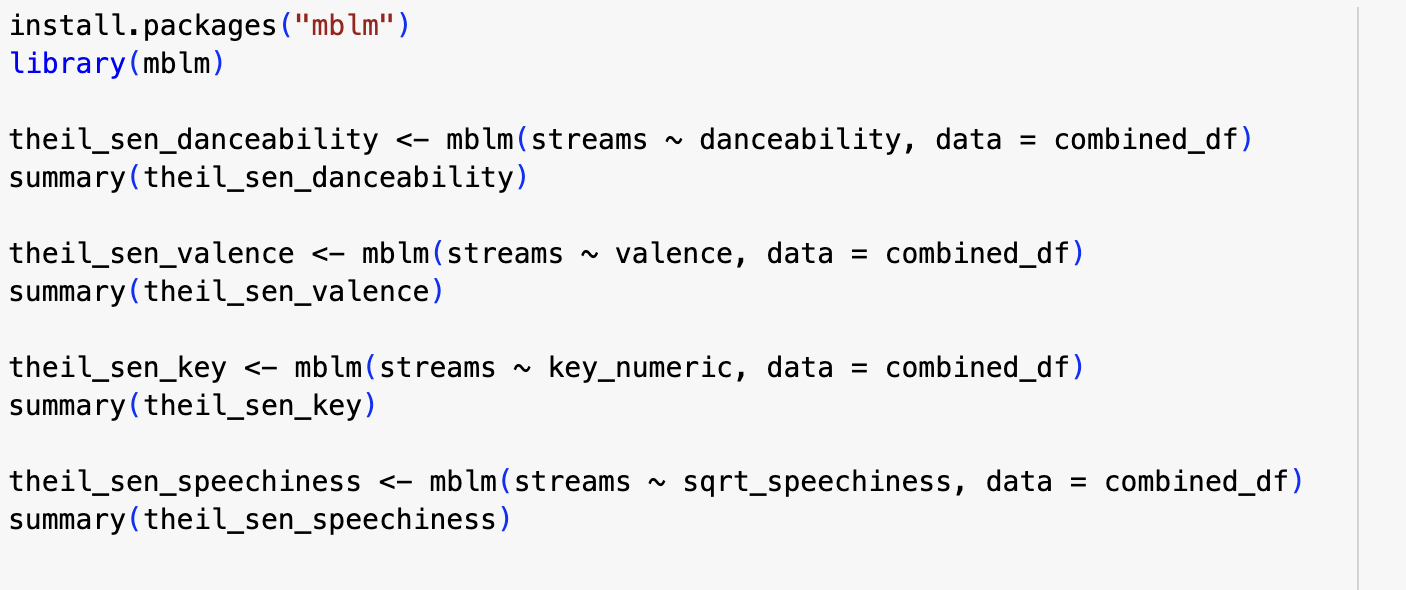
**Friedman Test**

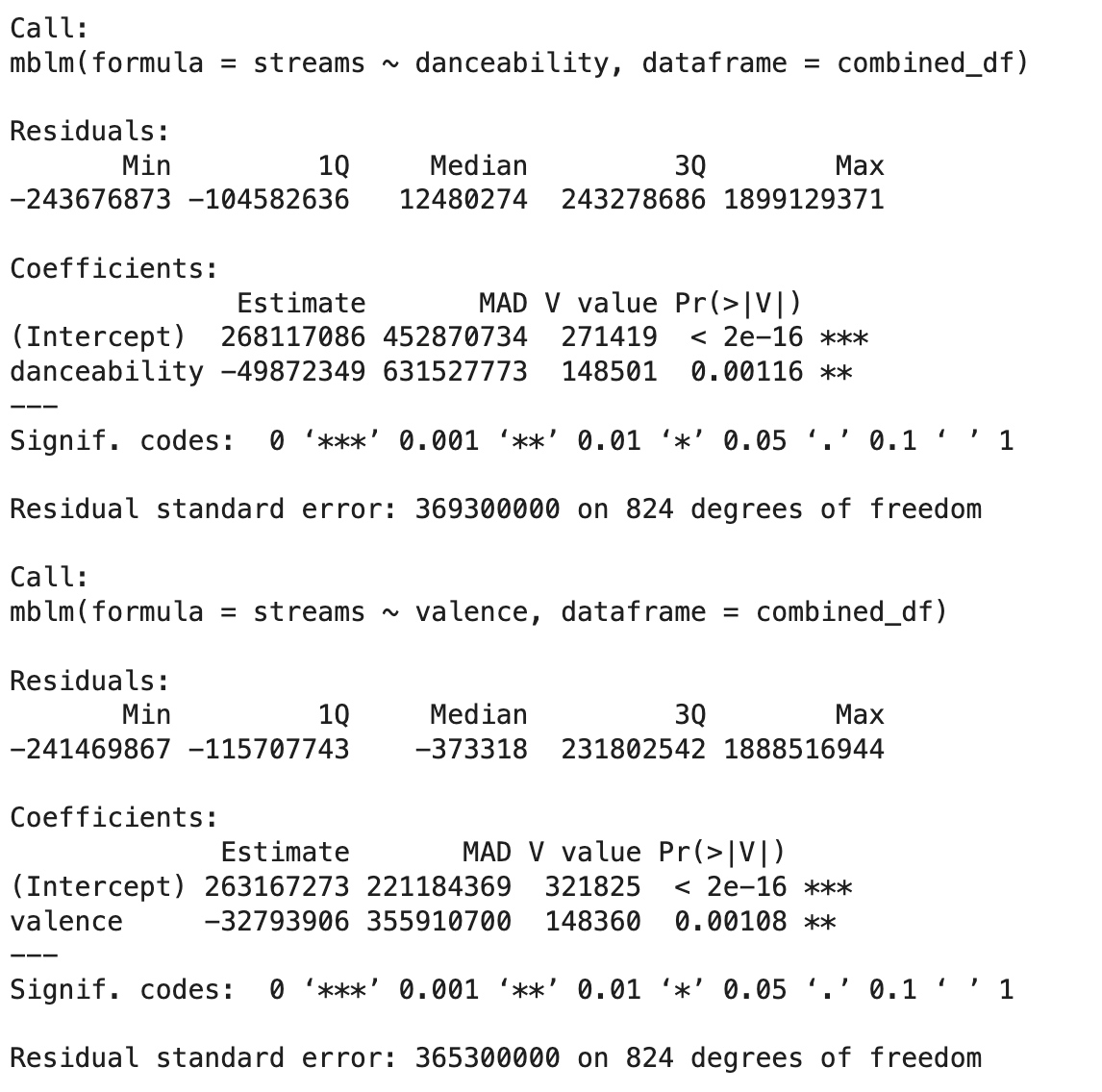


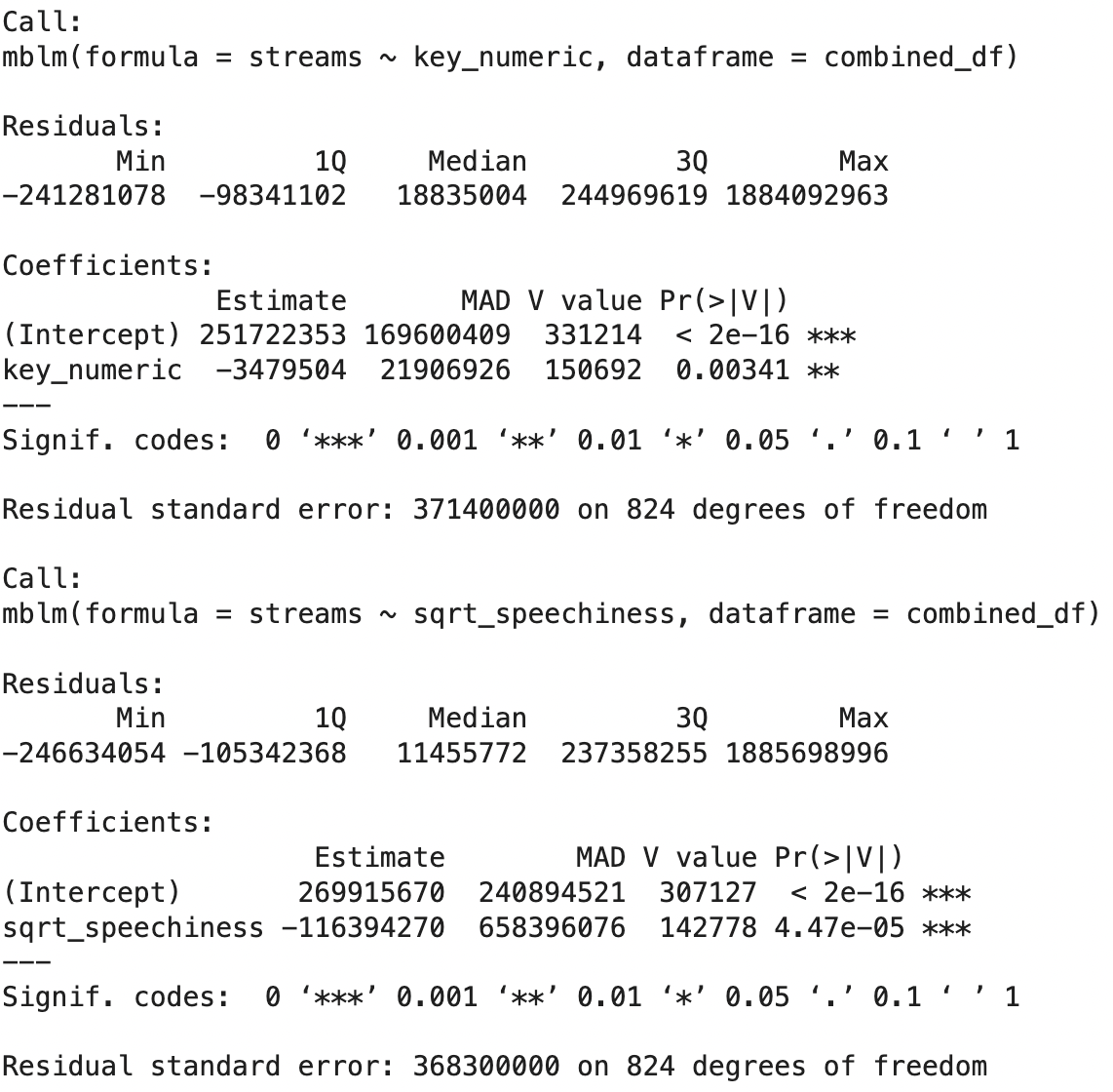
**Pairwise Wilcoxon Sum Rank Test**



**Theil Sen Regression Analysis**









**Spearman Correlation matrix**

