Analysis of Artist Streaming Data Using Regression and Visualization

---

Objective

This study's main goal was to look at how different types of artist streaming numbers—solo, featured, and lead—relate to their total Spotify streams. This research is to figure out how various team-ups and individual work affect an artist's overall streams, which shows how popular they are on the platform. By using a dataset with detailed streaming stats, this report aims to shed light on what makes an artist successful in the music world. It tries to answer the key question: *How do working alone or with others, including solo and featured streams, shape an artist's overall popularity as shown by their total Spotify streams?*

Getting a grip on what makes Spotify streams successful is key for artists, managers, and industry pros who want to fine-tune their growth plans. This analysis looks at how solo work, team-ups, and taking the lead affect overall popularity. It gives useful tips for artists who want to reach more listeners and boost their streaming numbers. Let's say featured streams (team-ups) have a stronger link to total streams. In that case, artists might want to team up with popular performers to get noticed more. On the flip side, if solo streams matter more, the findings would push artists to make more of their own content.

Code Overview

The program starts by processing a CSV file with Spotify streaming data. Each record in the dataset has details like total streams, solo streams featured streams, and lead streams for different artists. The software deals with missing or wrong data well making sure clean and correct entries go through. The processed data goes into a structured format using the ArtistData struct. This struct organizes metrics such as total streams, solo streams featured streams, and lead streams for each artist. This structured setup makes the next steps of analysis and visualization easier.

Linear Regression Analysis: The software uses the parsed data to calculate regression equations. These equations help uncover links between different streaming metrics. Linear regression is a statistical method that measures how two variables connect and what kind of relationship they have. In this case, solo streams featured streams, and lead streams act as independent variables. Total streams serve as the dependent variable. The program figures out two key parts of the regression model:

- Slope: Indicates how much the dependent variable (total streams) changes for every additional increase in the independent variable (such as streams on solo songs). A steeper slope suggests a stronger impact.

- Intercept: The intercept is the total streams for an artist when the independent variable is set to zero. This helps understand what is the least level of popularity that an artist would be able to get even without the presence of a specific metric of his streaming activity. By doing regression for each pair, the program is also able to determine such questions as whether the solo work or the collaboration or lead performances were more effective for streaming than the rest of the activities.

Testing: It consists of several unit tests to ensure accuracy and reliability of its components. These tests verify the appropriate execution of a few critical functions, such as the parsing of data and regression calculation. Some examples are:

Data parsing for artists will be executed by the parse_artist_data. This file, that is artists.csv, harbors within it several metrics. These include total streams, solo streams, lead streams, and featured streams for an artist. It has also ensured that all numerical data is valid and will substitute commas in numbers (i.e. "10,000" -> "10000") with default values where records are not found or invalid values exist. The parsed records will be kept in a Vec<ArtistData> format above, where each entry stands for metrics for a specific artist.

Regression tests to demonstrate that specific slope and intercept relate to the particular dataset. Such tests instill good programming principles to ensure the software, in which they are included, is robust and accurate.

Model of Regression to the Regression Model: The calculate_regression function calculates slope and intercept of a linear regression model based on two metrics such as those of solo streams and total streams. This correlates changes with one of its metrics with the other.

Relevance to Objective: By making correlations among the collaboration types and total streams, the regression models address more directly the research questions.

Steps Performed

1. Data Parsing

The dataset includes columns such as:

- **Total Streams**: The total amount of streams an artist has received.
- **Solo Streams:** Streams from individual projects.
- **Featured Streams**: Streams that showcase the artist as a partner.
- **Lead Streams**: Streams in which the artist takes the major lead.

The program:

- Read the dataset.

- Handles invalid or missing values gracefully by skipping problematic records.
- Converts numeric values (e.g., "85,041.3") into proper floating-point numbers.
- Extracts relevant metrics for analysis.

Output:
Successfully parsed **3000 valid records**, ensuring the dataset is clean and ready for analysis.

2. Linear Regression

To examine the relationships between the metrics, the program calculates regression equations for three comparisons:

- **Total Streams vs Solo Streams**
- **Total Streams vs Featured Streams**
- **Total Streams vs Lead Streams**

For each comparison, the program calculates:

- **Slope**: Represents the rate of change (e.g., how an increase in solo streams impacts total streams).
- **Intercept**: Represents the baseline value when the independent variable is zero.

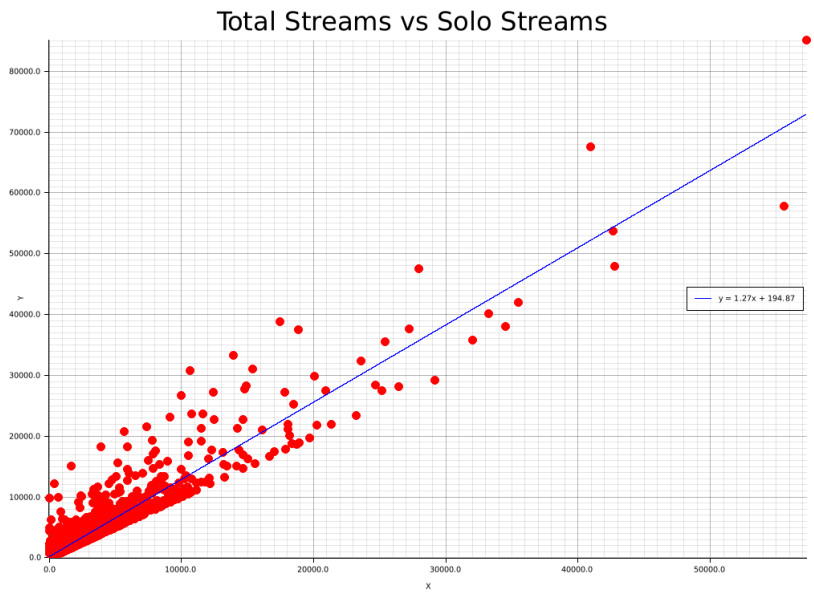
3. Scatter Plots and Regression Lines

The program creates scatter plots for each association and overlays a regression line to show trends. These visualisations give an intuitive grasp of how total streams correspond with each independent variable.

Findings

1. Total Streams vs Solo Streams

- **Regression Equation**:
  $y=1.27x+194.87$
- **Interpretation**:
  Solo streams are strongly correlated with total streams. The slope of 1.27 suggests that for every additional solo stream, total streams increase by 1.27 on average.
- **Visualization**:

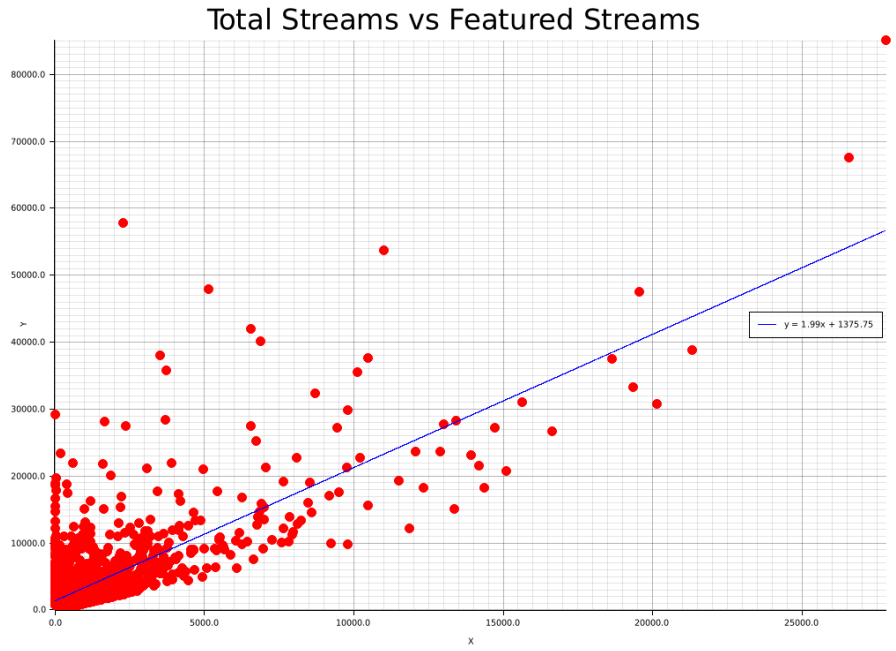


The scatter plot displays individual artists represented as red dots, with their solo streams shown on the x-axis and total streams on the y-axis. A blue regression line runs through the graph, illustrating the overall trend.

Insights: The regression equation y=1.27x+194.87 indicates a strong positive correlation between solo streams and total streams. The slope of 1.27 suggests that for each additional solo stream, an artist's total streams increase by an average of 1.27. The intercept of 194.87 means that even without any solo streams, artists still have a baseline level of total streams, likely influenced by factors such as collaborations or prior popularity. The close grouping of data points around the regression line highlights this strong relationship

2. Total Streams vs Featured Streams

- **Regression Equation**:
  y=1.99x+1375.75
- **Interpretation**:
  Featured streams also contribute significantly to total streams. However, the higher intercept value suggests that even artists with fewer featured streams tend to have a baseline level of total streams.
- **Visualization**:

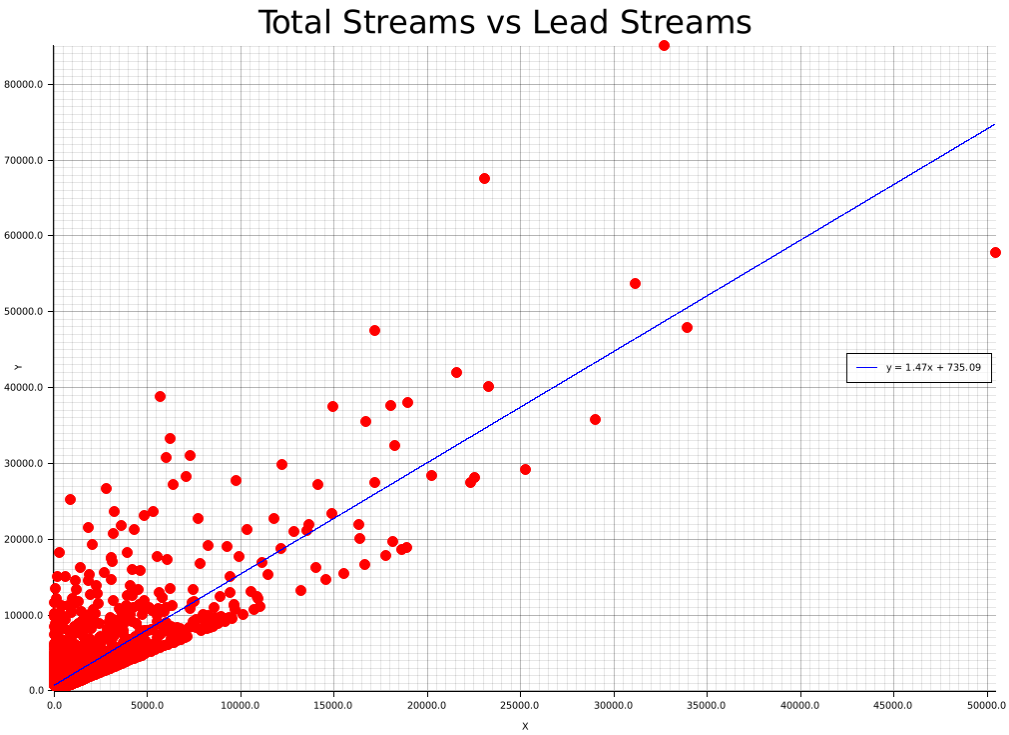


This scatter plot illustrates the connection between featured streams and total streams, with featured streams plotted on the x-axis and total streams on the y-axis. Artists are marked by red dots, while the regression line is depicted in blue.

Insights: The regression equation y=1.99x+1375.75 reveals that featured streams significantly affect total streams. The slope of 1.99 indicates that for each additional featured stream, an artist's total streams rise by almost 2, emphasizing the effectiveness of collaborations. The higher intercept of 1375.75 implies that even in the absence of featured streams, artists generally maintain a solid baseline of total streams, likely due to their individual work or overall brand appeal. In contrast to the solo streams visualization, this plot displays a slightly wider distribution, suggesting variability in how featured streams impact overall popularity.

3. Total Streams vs Lead Streams

- **Regression Equation**:
  y=1.47x+735.09
- **Interpretation**:
  Lead streams have a substantial impact on total streams, with a slope of 1.47 indicating a strong correlation. This suggests that being the lead artist is a key driver of streaming success.
- **Visualization**:



This scatter plot illustrates the connection between lead streams (where the artist takes the lead role) and total streams. Lead streams are represented on the x-axis, while total streams are shown on the y-axis. Artists are indicated by red dots, and the regression line is depicted in blue.

Insights: The regression equation y=1.47x+735.09 reveals a strong positive correlation between lead streams and total streams. The slope of 1.47 suggests that for each additional lead stream, an artist's total streams increase by an average of 1.47. The intercept of 735.09 indicates that even in the absence of lead streams, artists still reach a baseline number of total streams. This visualization highlights the significance of being a lead artist in achieving streaming success, although the variation in data points implies that the impact of lead streams can differ among various artists.

Overall Analysis of Visualizations

The scatter plots together illustrate how various types of streams—solo, featured, and lead—impact an artist's overall streaming performance. While all three metrics show positive correlations with total streams, the slopes and intercepts of the regression equations offer more detailed insights:

- Solo Streams: There is a steady and linear relationship, indicating that an artist's individual work consistently contributes to their success.

- Featured Streams: A slightly steeper slope and higher baseline value suggest that collaborations can significantly enhance an artist's reach and visibility.
- Lead Streams: A notable slope highlights the importance of being the main artist, but the lower intercept compared to featured streams implies that lead performances should be complemented by collaborations or solo work for optimal success.

Conclusion

Insights

1. **Solo Work Matters**:
   Solo streams have the most direct association with total streams. Artists that focus on solo projects find a consistent increase in overall streams.
2. **Collaboration Benefits**:
   Both featured streams and lead streams have a substantial impact on an artist's popularity, but with slightly different dynamics:
   - The greater intercept value indicates that featured streams provide a solid baseline.
   - Lead streams have a more gradual influence but are nonetheless important for overall success.

Key Takeaways

- **Solo Streams are Key**: Artists with considerable solo work had greater overall streams, highlighting the value of personal branding.
- **Balanced Strategy**: While solo work is predominant, collaborations (both as a feature and as a lead) give additional benefits, allowing musicians to reach a larger audience.
- **Visual Trends**: The scatter plots clearly demonstrate these associations, with regression lines emphasizing the patterns.

This Rust program was designed to analyze the Spotify stream data on artists and estimate correlations among various stream types--solo, featured, and lead--and total counts accumulated by them. The program first parses the CSV file through a data cleaning process of removing items that fail to have or contain incorrect values and converting structured numeric values into one consistent form (for instance, 10,000). The software goes through each record and extracts the following metrics: total streams, solo streams (streams due to tracks performed solely by the artist), featured streams (guest or collaborative streams), and lead streams. These values are saved into a structured manner (ArtistData struct) which enables easy manipulation and also analysis of the data. In the next stage of this workflow, a regression line would be calculated to determine the association between two variables-the independent variable can be considered as solo streams, while the dependent variable is total streams. The application uses the ordinary

least squares linear regression formulas to find the slope and intercept of the best-fit line for the trend most appropriate to the two variables. For example, the regression line for solo streams against total streams could be y=1.27x+194.87, which would mean that for every other 'solo' stream an artist gets, their total streams increase by around 1.27, starting from a constant value of 194.87 streams when solo streams are at zero. Similarly, regression equations would be derived by treating featured and lead streams as variables to show how they correlate with the overall performance of the artist on Spotify.

Besides quantitative analysis, the application makes it possible to use the plotters library to generate these scatter plots in which each correlation is represented. Individual scatter plots show every artist as a red dot with the total streams divided into three categories: solo, featured, or lead streams. A blue regression line is overlaid on the scatter plot to show the overall trend of the data. The graph is clearly labeled on both axes and includes a caption with the regression equation. For example, one could explore the relationship between Featured Streams and Total Streams to examine whether collaborations are more impactful and efficient than solo performances at garnering success for an artist.

These scatter plots are saved as PNGs by the application, allowing after-the-fact visualization analysis. The filenames are determined according to the relationship: total_stream_vs_solo_streams.png, total_stream_vs_featured_streams.png, total_stream_vs_lead_streams.png, and so on. This is being done via the main function, which first parses the dataset, then carries out regression analysis, and finally produces the visuals. It also includes error handling to ensure proper grace with issues such as missing files or wrong data as well as useful messages about errors for probe purposes.

This tool takes a comprehensive approach to analyzing artist streaming data, using statistical modeling and visual representation to identify music business trends. For example, it can reveal if solo streams or collaborations generate more overall streams, offering valuable information into the methods that will propel an artist's success on Spotify. The software examines 3,000 records from the collection and creates regression models for important associations, providing both a data-driven conclusion and accessible visuals for additional investigation and decision-making. It analyzes regression data and generates scatter plots, as well as running updated unit tests on the primary functions within the software to indicate correctness and reliability. The tests, including parsing the dataset and obtaining the correct regression equations, tests whether the dataset has been parsed correctly.

Test_parse_artist_data: This test simulates parsing a simple CSV dataset so the function parse_artist_data can extract and structure the artist data into the ArtistData struct. It verifies that the total number of records parsed is what is expected and verifies each of the parsed data points for correct extraction.

Test_calculate_regression: Regression correctness is tested through this test against a trivial dataset that has an inbuilt relation to ensure that the calculated slope and intercept are very high in their accuracy as ascertained by the known values.

It makes the code reliable and more forgiving of minor idiocy because it's tested on larger datasets. The tests were done through Rust's built-in testing framework and run using the cargo test command. This way, a quick answer pertaining to how critical parts of the program would behave against their expectations is guaranteed.