Spotify Popularity Data using Bayesian Hierarchical Models

MSDA Spring 2022 Capstone

By Israh Imam

# **INTRODUCTION**

Spotify is one of the largest audio streaming platforms, providing access to over 50 million songs to its users. Naturally, in such a large and diverse pool, popularity differs greatly amongst songs and artists themselves. It’s easy to get lost in this sea of options and forecasting success for artists and creators can be a daunting task. What if there was a way to predict the popularity of an artist’s next release? Doing this requires a deeper understanding of the typical popularity of a song, the variability in popularity from artist to artist, and popularity within a single artist's catalog of songs.

To explore these aspects of popularity, Bayesian hierarchical models will be created. First, an exploration of this data set will be performed, then the **complete-pooled** and **no-pooled** model approaches will be utilized to understand their effects on the prior and posterior prediction distributions of popularity. Finally, a **partial-pooled model** will be constructed to observe an artist's popularity, predict the popularity rating of an artist's next song, and predict the popularity of a song by artist not in the data set.

Using **artist** as a grouping-variable during the construction of these hierarchical models will allow to not only learn about the artists in this specific data set but help learn about the broader artist population on Spotify.

## **SPOTIFY DATA**

The data for this project was mined using the Spotify API. About 800 songs from Spotify's "All Out" playlists with release dates from 1950 to 2021 were collected. The "All Out" playlists contain the biggest songs from their respective decades. The variables obtained through the API for the songs and artists are as follows:

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Acousticness | Confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 = high confidence that track is acoustic. |
| Danceability | Describes how suitable a track is for dancing based on combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. 0.0 = least danceable and 1.0 = most danceable. |
| Energy | Measure from 0.0 to 1.0 that represents a perceptual measure of intensity and activity. Energetic tracks feel fast, loud, and noisy. |
| Instrumentalness | Predicts whether a track contains no vocals. "Ooh" & "aah" sounds are treated as instrumental. 1.0 measurement = greater likelihood that track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence increases as values approach 1.0. |
| Key | Key the track is in. Integers map to standard pitch notation (e.g. 0 = C, 1 = C#/D-flat, 2 = D, 3 = D#/E-flat, 4 = E, 5 = F, 6 = F#/G-flat, 7 = G, 8 = G#/A-flat, 9 = A, 10 = A#/B-flat, 11 = B). If no key detected, values are -1. |
| Length | Length of a track in milliseconds. |
| Liveness | Detects presence of audience in recording. Higher values represent probability that track was performed live. Values above 0.8 provide strong likelihood that track is live. |
| Loudness | Overall loudness of a track in decibels (dB). Loudness values are averaged across entire track and are useful for comparing relative loudness of tracks. Values range between -60 and 0 db. |
| Mode | Indicates modality (major or minor) of a track, type of scale from which melodic content is derived. Major = 1 and Minor = 0. |
| Speechiness | Detects the presence of spoken words in a track. The closer to 1.0, the more exclusively speech-like the recording (e.g. audio books, podcast, poetry). Values above 0.66 describe tracks that are most likely entirely made of spoken words. Values between 0.33 and 0.66 most likely are a mix between containing music and speech. Lastly, values below 0.33 most likely contain mainly music. |
| Tempo | Estimated tempo of a track in beats per minute (BPM). |
| Time Signature | Time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). This value ranges from 3 to 7 indicating time signatures of "3/4", to "7/4". |
| Valence | Measure from 0.0 to 1.0 describing musical positiveness conveyed by a track. Tracks with high valence sounds more positive, while tracks low valence sound more negative. |

# **RESULTS & ANALYSIS**

## **DATA EXPLORATION**

To begin the data exploration, characteristics of the various genres are shown through density plots. Some of the most popular songs are in the "Pop", "Adult Standards", and "Rock" genres. The "Blues", "Alternative", and "Reggae" genres have the least number of tracks.

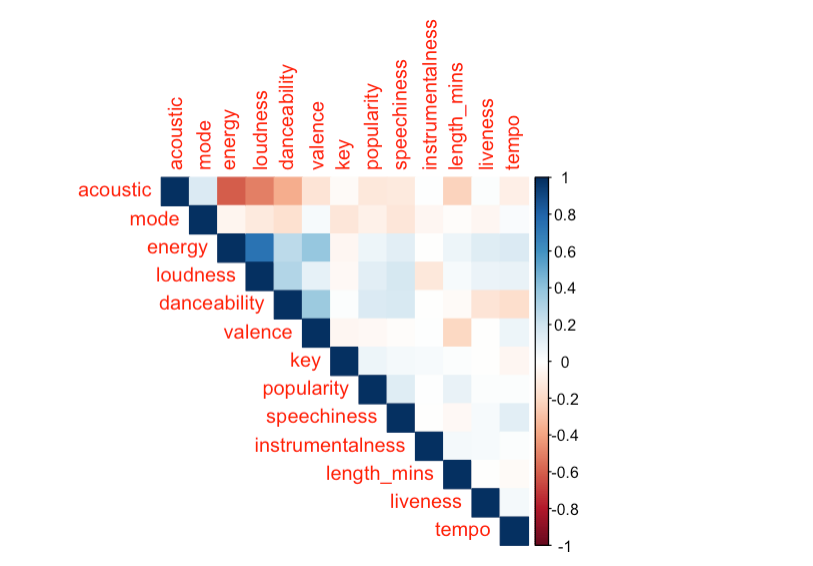
Chart, histogram

Description automatically generated

Table

Description automatically generated

**In terms of characteristics that make up a popular genre, the density plots show the following:**

* Most popular songs have a danceability rating between 0.50 and 0.75.
* The Country, Dance, Hip-Hop/Rap, and Pop genres have songs with high danceability.
* Most genres have low instrumentalness indicating that songs mostly contain spoken words.
* Most Alternative music has a valence score of about 0.50 indicating a mix of positive and negative feeling songs. It appears that Alternative music doesn't have much speechiness indicating that there is more music than words in the songs. It does not have much acousticness or liveness.
* Reggae music has high energy, loudness, and tempo.
* Apart from Reggae music being mostly written in Key "C", the other genres have songs written in a variety of keys.
* Most songs have energy between 0.50 and 1.00 indicating that these songs are more loud, fast, and noisy that others.
* This data set contains songs with a variety of valence ratings indicating a mix of positive and negative feeling songs.

**From the correlation plot pictured above we can see the following positive correlations:**

* Energy and loudness
* Danceability and energy
* Energy and valence
* Loudness and danceability
* Danceability and speechiness
* Danceability and valence

**From the correlation plot above we can see the following negative correlations:**

* Acousticness and energy
* Loudness and acousticness
* Danceability and tempo
* Acousticness and danceability
* Acousticness and length
* Valence and length

When observing the correlation plot to determine popularity’s relationship with other song characteristics, popularity appears to have a positive correlation with energy, loudness, danceability, speechiness, length, and key. These characteristics could be assumed to contribute to a song’s popularity rating.

**The jitter plots show that most popular songs have high values of the following characteristics. These variables are explored further through the plots below:**

* Most popular songs are low in acousticness
* Popular songs have a high danceability
* High energy is apparent in popular songs
* Most songs are low in instrumentalness
* Songs are written in a variety of keys
* Most songs are low in liveness
* Popular songs are loud
* Popular songs are low in speechiness
* Popular songs appear somewhat low in tempo
* Popular songs have a variety of valence ratings

Chart, bar chart

Description automatically generatedChart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart, waterfall chart

Description automatically generatedChart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated

Further confirming what was seen in the jitter plots, most popular songs have high danceability, energy, and are loud. Majority of the top 15 songs have a high tempo as well indicating that they are more upbeat and faster. Most popular songs have low values of liveness indicating that the available recording was not performed live. Speechiness of these songs are low as well. This means that most popular songs contain more music than they do words (or lyrics). Overall, these features (danceability, energy, loudness, speechiness) could indicate a song's popularity, but an artist's overall popularity could also contribute. To explore this, hierarchical models will be created.

## **Hierarchical Models**

In the previous section, an exploration and light analyses were performed to reveal what characteristics contribute to a popular song. To better understand popularity within our data set and the broader Spotify population, hierarchical models without predictors are created. The **Bayesian Hierarchical Modeling** method is a very popular form of statistical modeling that uses "levels" to create a complete model. These models are known to be a middle ground between fitting a model to different groups of data versus fitting a model to a whole set all at once. Hierarchical modeling is helpful because it describes variability in the parameters in a model.

Partitioning uncertainty and borrowing strength (information) are other strengths of these models as well. Partitioning uncertainty allows a process to be split into multiple terms, for example, presenting the residual error and any unexplained variability within a data set instead of having just one overall residual error. Borrowing information (or strength) allows for the generation of an informative prior when trying to compensate for not having the same level of detail for every point in a data set.

Three approaches that can be used to create hierarchical models are: **complete-pooling**, **no- pooling**, and **partial-pooling**. The two extremes are complete- and no-pooling while partial-pooling is typically considered a good balance of the others.

* **Complete-Pooling** pools all data points together and considers a "universal model" for all groups. This model usually doesn't allow for individuality between data points.
* **No-Pooling** creates a model for each group and assumes that one model doesn't hold valuable information about another. This model doesn't allow for generalization outside of the current data set being used.
* **Partial-Pooling**, being the middle ground, acknowledges a grouping structure and borrows information from other models while allowing each group to have its own model.

While the three approaches are very different and have their strengths and weaknesses, it's important to note that **complete- and no-pooled** models may **oversimplify** an analysis.

### **The Complete-Pooled Model**

The assumption for this section is that there is a response variable **Y,** but no predictors **X**.

After cleaning and removing songs with a popularity rating below or equal to 10, there are 643 songs and 373 artists in the data set.

The Complete-Pooled Model is created with the following parameters:

* ***j*** indicates an artist
* ***i*** indicates a specific song
* Chart, line chart

  Description automatically generated***nj*** denotes the number of songs for artist ***j***

The density plot above appears to be left skewed and Normality is assumed for this model. Using a **prior** (probability of an event before new data is collected) for reflects a weak understanding that average popularity rating for songs could be around **60.** Independently, a weakly informative prior for is understood as well. Since and are shared by every song ,they can be thought of as global parameters which do not vary by artist ***j.***

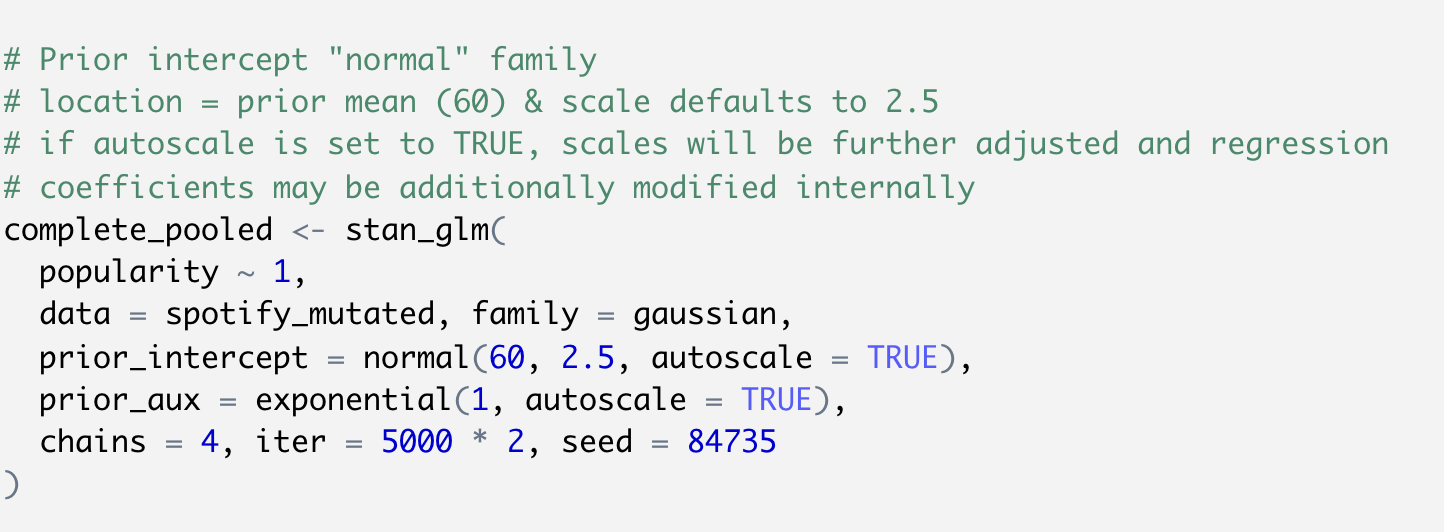
Therefore,

= global mean popularity; and

= global standard deviation in popularity from song to song

#### **Intercept Only Model**

By substituting **0**for global mean , this model becomes an intercept-only regression model with no predictors. The **stan\_glm()** function with the formula where **1** means “intercept-only” is used.



Table

Description automatically generated with low confidence

Using the **prior summary**, the completed-pooled model is the following:

|  |  |
| --- | --- |
| **Layer** | **Description** |
|  | Data layer describes Normal distribution |
|  | Prior layer describes 60 prior knowledge of popularity |
|  | Hyper prior layer with Exponential prior of 0.09 |

Table

Description automatically generatedThe summary below shows a popularity rating of about **72.0** with a standard deviation between songs **11.0**.

Chart, line chart

Description automatically generatedThe following plots show the predictive popularity (light dots) along with the observed mean popularity (dark dots) for all artists in this data set and illustrates the consequence of oversimplification.

Since all songs were lumped together and assigned the same mean all artists have been treated the same though they are not. The mean\_PPD (mean prior prediction distribution) is very close to its mu since this model ignored all artist-specific information.

Chart, histogram

Description automatically generated

### **The No-Pooled Model**

A no-pooled model considers each artist and separately analyzes their popularity. Though there are no predictors, in this section, **artist** will be treated as a predictor. This will limit the understanding to only the artists in this data set but will help understand this model. A density plot is shown to begin exploration.

Based on the density plot above, it appears that most artists have popularity ratings between 50 and 90. Since a no-pooled model ignores the shared global mean popularity level, it incorporates **group-specific parameters** that vary by artist. For this model, it will be assumed that each artist **j** and the popularity of songs **i** are normally distributed around some **j** with standard deviation **.**

Therefore,

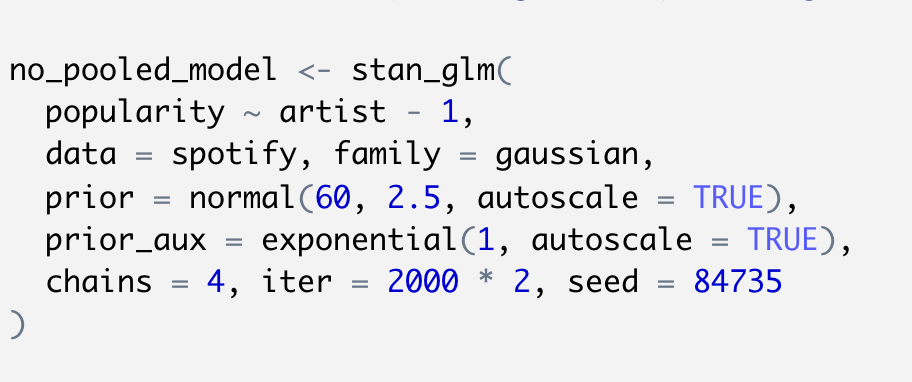
= mean song popularity for artist **j**; and

= standard deviation in popularity from song to song within each artist

For the **no-pooled** model it’s assumed that mean popularity is different between artists ***j***, but variability in popularity from song to song, , is the same for each artist. Because this model does not borrow information from other artists, all songs by the same artist share the same mean popularity. In addition, songs by different artists have different means, so each one gets a unique mean, and since information is not shared about popularity, by assuming artists share the same standard deviation parameter , information is pooled from artists to learn about .

The finalized no-pooled model utilizes weakly informative Normal priors for (average popularity being around 60) and an Exponential prior for the standard deviation .

To create the no-pooled model, a call to **stan\_glm()** with formula is used to treat **artist** as a predictor in a regressional model of popularity. The posterior predictive model is shown below.



Chart

Description automatically generated

Using this model, each artist has received a unique mean popularity, thus getting a unique posterior predictive model. The plot appears to confirm that the posterior predictive model correctly predicted each artist's next song popularity based on their observed sample songs. While this could be cause for celebration, there a few issues that arise from a no-pooled model. The no-pooled model does not borrow strength which results in ignoring data on one artist when learning about typical popularities of another. This creates the risk of ignoring valuable information about other artists. In addition to this weakness, this model also cannot be generalized outside this sample.

### **The Partial-Pooled Model**

The partial-pooled model will treat **artist** as a grouping variable vs. a predictor with the hopes that insight into the artists in the data and the broader Spotify population will be understood.

The main difference between a grouping variable and predictor is that if **X** covers all categories of interest, it's a potential predictor. If **X** is a random sample from many variables of interest, then it's a potential grouping variable. Since there is a small and random sample of artists from the broader population of Spotify artists, **artist** is treated as a grouping variable for the partial-pooled model.

This model is comprised of multiple layers that are ultimately hierarchically structured. Two of the layers are linked to the hierarchically structured song data and the third layer is linked to the prior models on the model parameters.

#### **Partial-Pooled Model Layers**

##### Layer 1: how song popularity varies within an artist

Like the no-pooled model, Layer 1 is artist-specific and recognizes that popularity does partially depend on an artist. Within an artist it’s assumed that the popularity of songs is Normally distributed around a mean with standard deviation .

The Layer 1 parameters are the same as the no-pooled model:

= mean song popularity for artist *j*; and

= standard deviation in song popularity within an artist

##### Layer 2: how popularity varies between artists

The second layer of this model acknowledges that artists in this data set are sampled from the general Spotify population. In this population, popularity varies by artist, thus the mean is modeled **between** artists by assuming mean popularity levels are Normally distributed around with standard deviation

The Layer 2 parameters are:

= global average mean song popularity across all artists *j*; and

= between-group variability (standard deviation in mean popularity from artist to artist)

##### Layer 3: the prior understanding of the entire Spotify population (the global parameters)

Layer 3 specifies the prior models for the global parameters. These parameters are shared across all of Spotify artists.

The final model for this partial-pooled model is the following:

individual songs within an artist

variability between artists

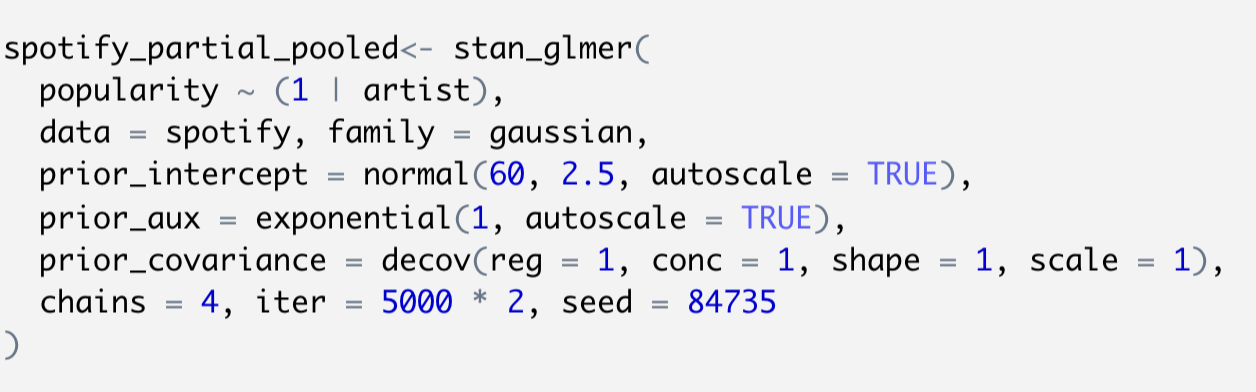
prior models on global parameters

Adjusting for any artist with a popularity above or below the average mean, Layers 1 and 2 can also be written as follows:

)

can be thought of as adjustments to the global mean song popularity and the adjustments can be thought of normal deviations from 0 from standard deviation

Using the **stan\_glmer()** function with formula is used to indicate that **artist** is a grouping variable instead of being a predictor of popularity. For the posterior simulation, the Markov chain Monte Carlo (MCMC) method is used. The MCMC method is utilized to **approximate the posterior distribution** of a parameter of interest by random sampling in a probabilistic space.



Text

Description automatically generated

The following plot shows the posterior simulated song popularity (light blue) and the observed popularity (dark blue). The predictions are almost aligned with the observed popularities.

Chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

After creating a data frame using the **partial-pooled model,** there are 376 parameters (373 artist-specific and 3 global parameters). The simulation contains Markov chains of length 20,000 for each of the 376 parameters.

A snapshot of the data stored is found below:

Graphical user interface, text, application

Description automatically generated

#### **Global Parameters Analysis**

The parameters , **y**, and **µ** are shared by all artists within and beyond this sample. The parameters and their meaning within the results of **stan\_glmer()** output are listed below:

= intercept

**y** = sigma

**µ** 2 = Sigma[artist:(Intercept), (Intercept)]

A summary of the partial pooled models is shown below. The "fixed" effect term signifies "non-varying" or "global".

Text

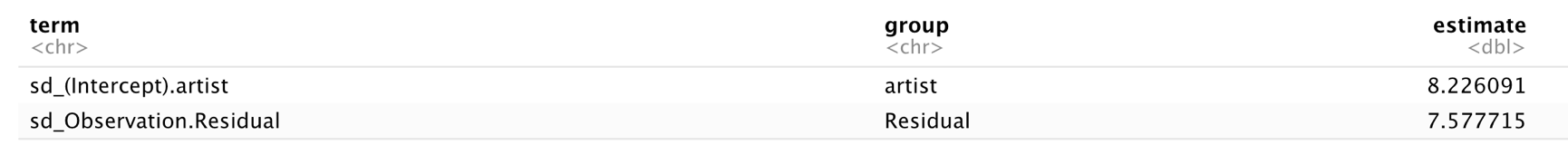
Description automatically generated with medium confidence

A picture containing text

Description automatically generated

The summary shows an 80% confidence that the average artist during the years 1950 - 2021 has a mean popularity rating between 70.6 and 72. The following table shows the parameter randomness or variability in the posterior medians.

A picture containing logo

Description automatically generated

The posterior median **(sd\_Observation.Residual)**  shows that song popularity ratings can vary by about 7.5 points **within** an artist indicating that some songs by a specific artist are more popular than others. The standard deviation **(sd\_(Intercept).artist)**  variable is about 8.2 which tells us that the mean popularity from artist to artist can vary by 8.2 points. Using the following formulas:

The variability in song popularity between artists and differences in popularity of songs within an artist can be calculated.

Text

Description automatically generated

From the above calculations, it can be concluded that about **54%** of the variability in song popularity is explained between artists and about **45%** is explained by differences in song popularity within an artist. Next, a posterior analysis of the group-specific parameters is explored. The artist-specific mean popularity as adjustments to the global popularity rating can be written as:

represents the difference between an artist's j popularity and the global mean popularity. A summary of all the posterior terms is shown. The "ran\_vals" effect is used to show random artist-specific values:

Text

Description automatically generated

Background pattern

Description automatically generated

The summary table above shows, based on an 80% **credible interval** (the range containing a particular percentage of probable values) for a given artist's adjustment, there's an 80% chance that Olivia Rodrigo's average popularity rating is between 12.7 and 21.7 **above** an average artist while Trevor Daniel's rating is between 23.6 and 39.2 points **below** an average artist. By combining the MCMC simulations for the global mean and artist adjustments to simulate posterior models for the artist-specific means we get the following result:

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generatedGraphical user interface, text, application

Description automatically generated

The summary shows that NSYNC's mean popularity rating is about 64.0, 6.2 points lower than the average 70.2 while a-ha's mean popularity rating is about 73.4, 3.1 points higher than the average. Next, using the mean\_qi() function, the posterior summaries for each artist's mean popularity, the posterior mean, and 80% credible interval are observed:

Table

Description automatically generated

Graphical user interface, application

Description automatically generated

With 80% posterior certainty, it can be assumed that SZA’s mean popularity rating is between 72.0 and 86.3 and Shawn Mendes’s rating is between 76.1 and 83.5. A subset of the 80% posterior credible interval is shown below and provides a picture of the variability in the posterior understanding of the artist's mean popularity levels:

The plot above appears to show what is expected; some artists are more popular than others, which is tied to their location on the plot. It also shows that artist popularity varies in scale, their 80% posterior credible intervals being wider than others (SZA & Shawn Mendes highlighted). This could be due to the number of songs that are being analyzed per artist. Some artists have as little as one song analyzed while another has as many as 9 songs analyzedA picture containing text, antenna

Description automatically generated.

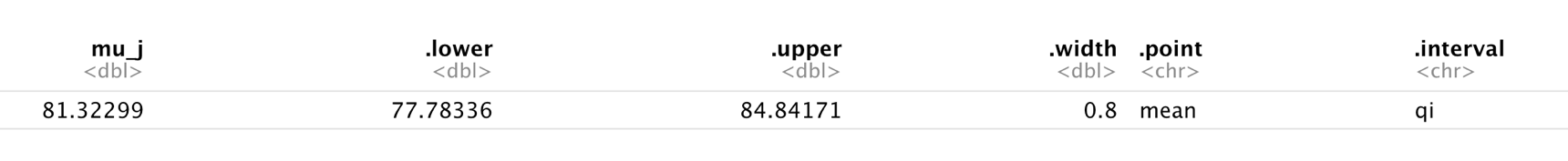
### **Posterior Prediction**

This section will focus on predicting the popularity of the next new song released by an artist on Spotify. The posterior prediction for an observed group or artist will be considered and layer 1 of the hierarchical model is used for this portion. Layer 1 assumes that the popularity of an artist's songs are Normally distributed around their own mean popularity level with standard deviation . A prediction is simulated from the Layer 1 model evaluated at each of the 20,000 MCMC parameter sets.

The first Markov chain for is obtained by summing the global mean and the artist adjustment. A simulation is then created to predict the popularity from the Layer 1 model at each MCMC parameter. This prediction will be for artist, Dua Lipa:

Graphical user interface, text, application

Description automatically generated



Posterior predictive model summary

Posterior summary of

Dua Lipa's posterior predictive model shows an 80% posterior chance that her next song will have a popularity rating between 71 and 91.6 points. How great for Dua! Comparing this prediction with the posterior summary of the range in popularity for her next song is much wider than the range in the summary of her underlying mean song popularity of 77.7 and 84.8. This approach for predicting song popularity is focused on two things: within-group variability and posterior variability. Within-group variability is understanding that not all Dua Lipa songs are equally popular and posterior variability is the understanding that the underlying mean and variability for Dua Lipa’s popularity across her songs are unknown and can also vary. With these variabilities, a prediction for Dua Lipa’s underlying mean popularity can be more confident than a prediction for a single song.

Next, a prediction for an artist whose information we don't have is performed. The prediction is for artist, Jay Sean:

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated with medium confidence

With 80% posterior certainty, Jay Sean's next song may have a rating between 56.8 and 85.4 points. The prediction was completed by simulating a mean through the Layer 2 model evaluated at each MCMC parameter set. Popularity was then predicted using the Layer 1 model evaluated at each MCMC parameter set. The prediction approach for Jay Sean also addresses between-group variability or the understanding that not all artists on Spotify are equally popular.

Text

Description automatically generated

Chart, line chart, histogram

Description automatically generated

Using the shortcut method, **posterior\_predict()**, to create an 80% posterior predictive model, it can be observed that because there is information on Dua Lipa, her next song popularity range is much narrower than Jay Sean's next song popularity rating; whom there is no information for.

# **CONCLUSION**

Throughout this analysis a few approaches have been taken to understand the popularity ratings of the artist:

* Data exploration to explore characteristics that make up a popular song on Spotify
* Complete-pooled model creation
* No-pooled model creation
* Partial-pooled model creation
* Predicting the popularity of an artist’s new song

The first two (complete- & no-pooled) models oversimplified the analysis and understanding of popularity ratings. Complete pooling completely ignored artists, lumped all songs together, and treated all artists the same though they are not. The no-pooled model raised two issues:

* + It ignored data on one artist when learning about typical popularities of another; and
  + The model could not be generalized outside this sample

The partial-pooled model was the **balance** between these models as the predictions **shrunk** towards the global trend of the complete-pooled model and away from the local trends of the no-pooled model. This phenomenon is called, **shrinkage.** Shrinkage occurs when group-specific local trends in a partial-pooled model are pulled or shrunk toward the global trends. Partial-pooled models allow for **generalization** of observations from a sampled group to the broader populations while **borrowing strength** from all sampled groups when learning about an individual sampled group.

Partial-pooled models also allow a balance in bias-variance trade-offs. The complete-pooled model was oversimplified and ignored an artist's mean popularity and therefore had higher bias and lower variance. The no-pooled model had the opposite problem with lower bias and higher variance. Because the no-pooled model looked at the artists in this specific data set, and detected group-specific trends, it had less bias. However, if the sample that was analyzed changed by adding or taking away an artist, the model could have changed, and produced a different conclusion which contributes to its higher variance.

Since partial-pooled models take group-specific trends into account it's less biased, and since it takes global trends into account it's less variable. This type of model is known to be a good balance between complete- and no-pooled models.

# **REFERENCES**

B. “A Zero-Math Introduction to Markov Chain Monte Carlo Methods.” *Medium*, Towards Data Science, 21 June 2018, towardsdatascience.com/a-zero-math-introduction-to-markov-chain-monte-carlo-methods-dcba889e0c50.

“Bayesian Hierarchical Models.” *YouTube*, uploaded by NEON Science, 27 Mar. 2020, www.youtube.com/watch?v=SMWleVKO9ZM&t=268s.

“Credible Intervals (CI).” *bayestestR*, easystats.github.io/bayestestR/articles/credible\_interval.html. Accessed 2 Apr. 2022.

Johnson, Alicia, et al. *Bayes Rules! (Chapman and Hall/CRC Texts in Statistical Science)*. 1st ed., CRC Press, 2022, www.bayesrulesbook.com.

Krishnamurthy, Surya. “Introduction to Hierarchical Modeling - Towards Data Science.” *Medium*, Towards Data Science, 15 Dec. 2021, towardsdatascience.com/introduction-to-hierarchical-modeling-a5c7b2ebb1ca.