Course Project

Emily Dobar

2022-12-02

Taylor Swift Song Popularity

Taylor Swift is an American singer-songwriter whose over ten years worth of multi-genre self-narrative songs have earned her praise and awards globally and rightfully placed her as one of the top best-selling artists of all time. Her seemingly non-stop success brings about the question: what makes her music so popular? Spotify, a popular music streaming site/app, calculates popularity of songs on its platform by way of internal algorithms based on many factors, including number of *listens* each song gains over time. Spotify API data downloaded from kaggle.com provides the popularity ratings for Taylor Swift's songs, as well as many other variables such as length, danceability, acousticness, energy, instrumentalness, liveness, loudness, speechiness, valence, and tempo. Using this data, I will explore how and if these variables could contribute to predicting song popularity and compare Frequentist and Bayesian regression techniques.

```
# Loading Libraries
library(kableExtra)
library(formattable)
library(fbst)
library(tidyverse)
library(fitdistrplus)
library(devtools)
library(caret)
library(ggplot2)
library(brms)
library(betareg)
library(bayesbr)
library(bayesian)
library(bayesplot)
library(bayestestR)
library(parsnip)
library(performance)
library(rstanarm)
library(BayesFactor)
library(LearnBayes)
library(loo)
library(insight)
library(mlbench)
library(rjags)
library(R2jags)
library(MRH)
library(olsrr)
library(bamlss)
library(BayesVarSel)
library(psych)
library(corrplot)
library(dampack)
```

```
#select data
raw <- as.data.frame(read.csv(file.choose(), header=T, sep=","))

#cleaning unnecessary columns
dat <- raw[,-c(1:5)]

#convert from ms to min
dat$length <- dat$length / 1000 / 60

#rename length variable
names(dat)[1] ="duration"

#add binary variable for popularity
#If pop < 50, bin = 0
#If pop >= 50, bin = 1
dat$binary <- dat$popularity
dat$binary[dat$binary < 50] <- 0
dat$binary[dat$binary >= 50] <- 1


set.seed(1)
```

Data Exploration

First, I looked at the different contributing variables within the given data set to understand what each means within the context of the data. Table 1 summarizes each of the variables, the domains within which the values could fall, and the definition of each.

Table 1. Variables

Variable	Domain	Description
Duration	Any	Song Length in Minutes
Danceability	0 to 1	How Suitable a Track is for Dancing
Acousticness	0 to 1	Amount a Song is Acoustic
Energy	0 to 1	A Perceptual Measure of Intensity and Activity
Instrumentalness	0 to 1	The Amount of Vocals in the Song
Liveness	0 to 1	Probability that the Song was Recorded with a Live Audience
Loudness	-60 to 0	Tendency of Music to be Recorded at Steadily Higher Volumes
Speechiness	0 to 1	Presence of Spoken Words in a Track
Valence	0 to 1	A Measure of How Happy or Sad the Song Sounds
Tempo	Any	Beats per Minute
Popularity	0 to 100	Percent Popularity
Binary	0 or 1	Whether or Not A Song is Popular

Note:

Reference: Dagohoy, 2021

Speechiness: >.66, Spoken Word; .33-.66, Music & Words; <.33, No Speech

Duration was converted from milliseconds (original data) to minutes for better understanding in analysis

Binary (added to data afterwards): Popularity = 0-49, 0; Popularity = 50-100, 1

I also made sure to look at the summary of each variable in the data set, looking at the ranges and quartiles of each to get a better understanding of the domains. The summary of variables is below.

```
summary(dat)
```

```
##     duration      danceability     acousticness          energy
##  Min.   :1.786   Min.   :0.2920   Min.   :0.000191   Min.   :0.118
##  1st Qu.:3.531   1st Qu.:0.5270   1st Qu.:0.030450   1st Qu.:0.462
##  Median :3.900   Median :0.5930   Median :0.156000   Median :0.606
##  Mean   :3.944   Mean   :0.5886   Mean   :0.321634   Mean   :0.586
##  3rd Qu.:4.241   3rd Qu.:0.6555   3rd Qu.:0.674000   3rd Qu.:0.732
##  Max.   :6.731   Max.   :0.8970   Max.   :0.971000   Max.   :0.944
##  instrumentalness       liveness          loudness        speechiness
##  Min.   :0.000e+00   Min.   :0.03350   Min.   :-17.932   Min.   :0.02310
##  1st Qu.:0.000e+00   1st Qu.:0.09295   1st Qu.: -8.861   1st Qu.:0.02950
##  Median :2.010e-06   Median :0.11500   Median : -6.698   Median :0.03720
##  Mean   :2.490e-03   Mean   :0.14593   Mean   : -7.322   Mean   :0.06558
##  3rd Qu.:6.365e-05   3rd Qu.:0.16800   3rd Qu.: -5.337   3rd Qu.:0.05510
##  Max.   :1.790e-01   Max.   :0.65700   Max.   : -2.098   Max.   :0.91200
##     valence           tempo          popularity        binary
##  Min.   :0.0499   Min.   : 68.53   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:0.2775   1st Qu.: 96.05   1st Qu.:58.00   1st Qu.:1.0000
##  Median :0.4160   Median :121.96   Median :63.00   Median :1.0000
##  Mean   :0.4230   Mean   :124.14   Mean   :61.23   Mean   :0.8538
##  3rd Qu.:0.5450   3rd Qu.:146.04   3rd Qu.:67.00   3rd Qu.:1.0000
##  Max.   :0.9420   Max.   :207.48   Max.   :82.00   Max.   :1.0000
```

When looking at how the variables correlate with one another, many variables seem to have little to no correlation. Understandably, acousticness is highly negatively correlated with both energy and loudness, while energy and loudness are highly positively correlated. In investigating popularity and what variables may relate or contribute to it, most variables too have little to no correlation, though liveness has a moderately negative correlation, which is interesting to note. The more likely a song is live, which is to say that a song on an album is a live recording in front of an audience, it is more likely to be less popular. The binary variable I incorporated into the data to categorize popularity ratings as either “popular” (1) or “unpopular” (0), seems to show some mild negative correlation with liveness, loudness, and speechiness, which is also interesting to note. See Figure 1 for more detail.

Figure 1. Variable Correlation

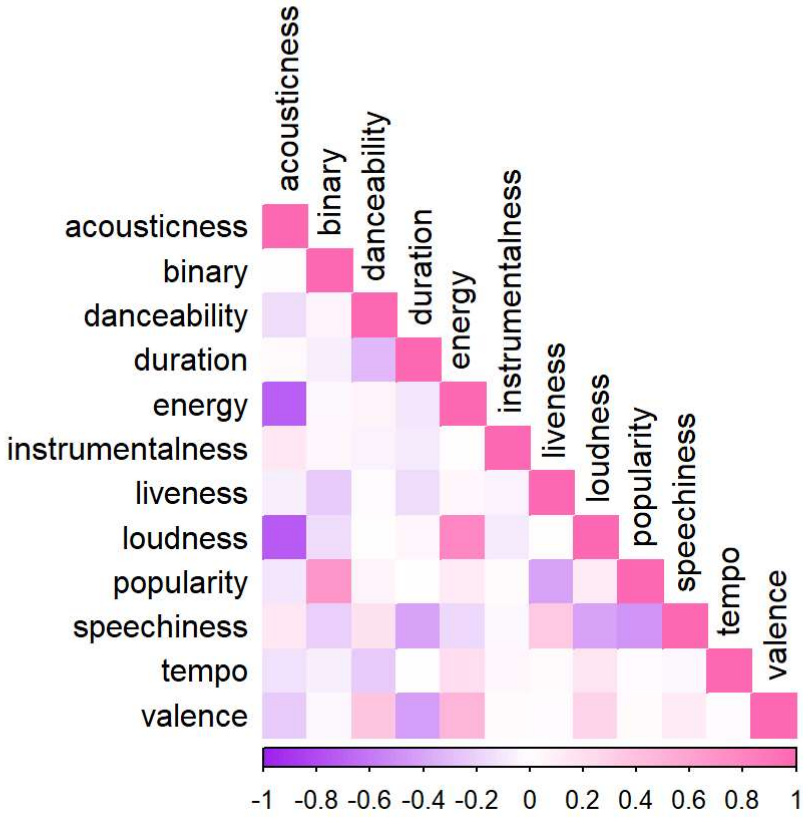


Table 2 looks more closely at the popularity variable. Popularity of Taylor Swift’s songs ranges from 0 (not popular at all) to 82 (very popular), with mean popularity being about 61, with a standard deviation of about 11.9. With this in mind, it seems as though most of Taylor Swift’s songs are rated above 50, which for the purposes of my analysis, indicates they are popular (according to my binary variable), which makes sense given Taylor Swift’s popularity streak in the music industry.

Table 2. Popularity Statistics Summary

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	171	61.22807	11.90455	63	62.17518	7.413	0	82	82	-2.334198	9.986535	0.9103635

Prior Distribution

I decided to look into the distribution of the popularity variable. Because I know that the binary variable extracted from the popularity variable could be modeled with a Bernoulli or binomial distribution, I decide to set a weakly informative beta prior for the analysis.

To fit a Beta prior to the data, I used an already-existent function (Coghlan, 2010) to find the best-fit hyperparameters for the Beta prior using three quantiles (the 10%, 50%, and 90% quantiles) from the data. The 10% quantile is 48, 50% is 63, and 90% 74, as seen in the R calculations below.

```
quantile(dat$popularity, probs= .1)
```

```
## 10%
##  48
```

```
quantile(dat$popularity, probs= .5)
```

```
## 50%
##  63
```

```
quantile(dat$popularity, probs= .9)
```

```
## 90%
##  74
```

```
q1 <- list(p=.5, x= 63/171)
q2 <- list(p=.1, x= 48/171)
q3 <- list(p=.9, x= 74/171)
```

```r
# from (Coghlan, 2010)

findBeta <- function(quantile1,quantile2,quantile3)
  {
     # find the quantiles specified by quantile1 and quantile2 and quantile3
     quantile1_p <- quantile1[[1]]; quantile1_q <- quantile1[[2]]
     quantile2_p <- quantile2[[1]]; quantile2_q <- quantile2[[2]]
     quantile3_p <- quantile3[[1]]; quantile3_q <- quantile3[[2]]

     # find the beta prior using quantile1 and quantile2
     priorA <- beta.select(quantile1,quantile2)
     priorA_a <- priorA[1]; priorA_b <- priorA[2]

     # find the beta prior using quantile1 and quantile3
     priorB <- beta.select(quantile1,quantile3)
     priorB_a <- priorB[1]; priorB_b <- priorB[2]

     # find the best possible beta prior
     diff_a <- abs(priorA_a - priorB_a); diff_b <- abs(priorB_b - priorB_b)
     step_a <- diff_a / 100; step_b <- diff_b / 100
     if (priorA_a < priorB_a) { start_a <- priorA_a; end_a <- priorB_a }
     else                     { start_a <- priorB_a; end_a <- priorA_a }
     if (priorA_b < priorB_b) { start_b <- priorA_b; end_b <- priorB_b }
     else                     { start_b <- priorB_b; end_b <- priorA_b }
     steps_a <- seq(from=start_a, to=end_a, length.out=1000)
     steps_b <- seq(from=start_b, to=end_b, length.out=1000)
     max_error <- 10000000000000000000
     best_a <- 0; best_b <- 0
     for (a in steps_a)
     {
        for (b in steps_b)
        {
           # priorC is beta(a,b)
           # find the quantile1_q, quantile2_q, quantile3_q quantiles of priorC:
           priorC_q1 <- qbeta(c(quantile1_p), a, b)
           priorC_q2 <- qbeta(c(quantile2_p), a, b)
           priorC_q3 <- qbeta(c(quantile3_p), a, b)
           priorC_error <- abs(priorC_q1-quantile1_q) +
                           abs(priorC_q2-quantile2_q) +
                           abs(priorC_q3-quantile3_q)
           if (priorC_error < max_error)
           {
             max_error <- priorC_error; best_a <- a; best_b <- b
           }
        }
     }
     print(paste("The best beta prior has a=",best_a,"b=",best_b))
  }
```

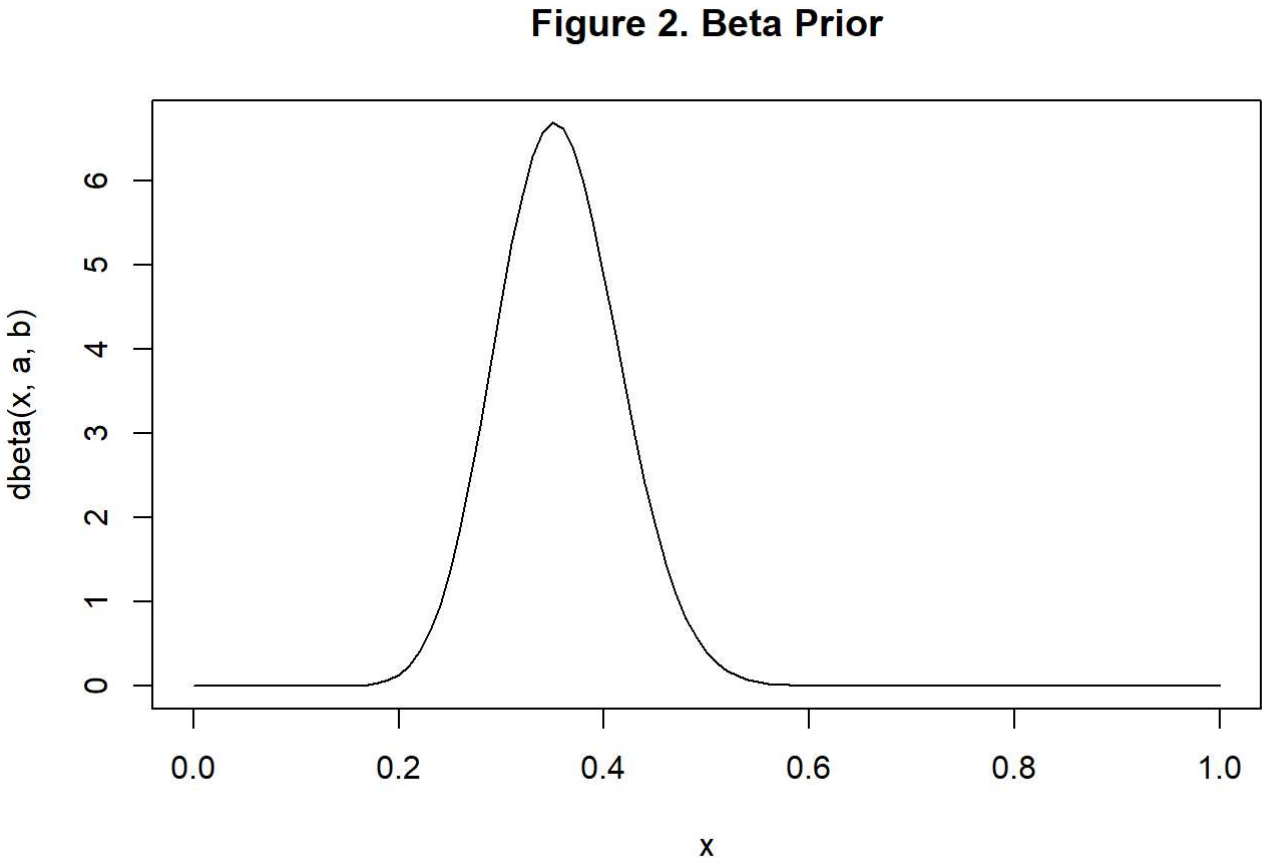
From there, it is determined that the best hyperparameters are $\alpha = 23.0097897897898$ and $\beta = 41.6689089089089$, for a distribution $\mathcal{B}(23.0097897897898, 41.6689089089089)$.

Figure 2 visualizes the Beta prior.

```
findBeta(q1, q2, q3)
```

```
## [1] "The best beta prior has a= 23.0097897897898 b= 41.6689089089089"
```

```
a <- 23.0097897897898
b <- 41.6689089089089

curve(dbeta(x, a, b))
title(main = "Figure 2. Beta Prior")
```



Likelihood

In order to further look into the posterior distribution from the Beta prior, I used another already-built function for finding the Likelihood distribution (Coghlan, 2010).

Here, I used the binary variable, setting all Binary = 1 as successes. Tabulating the binary variable, I can see that the number of successes out of the 171 observations in the data set is 146.

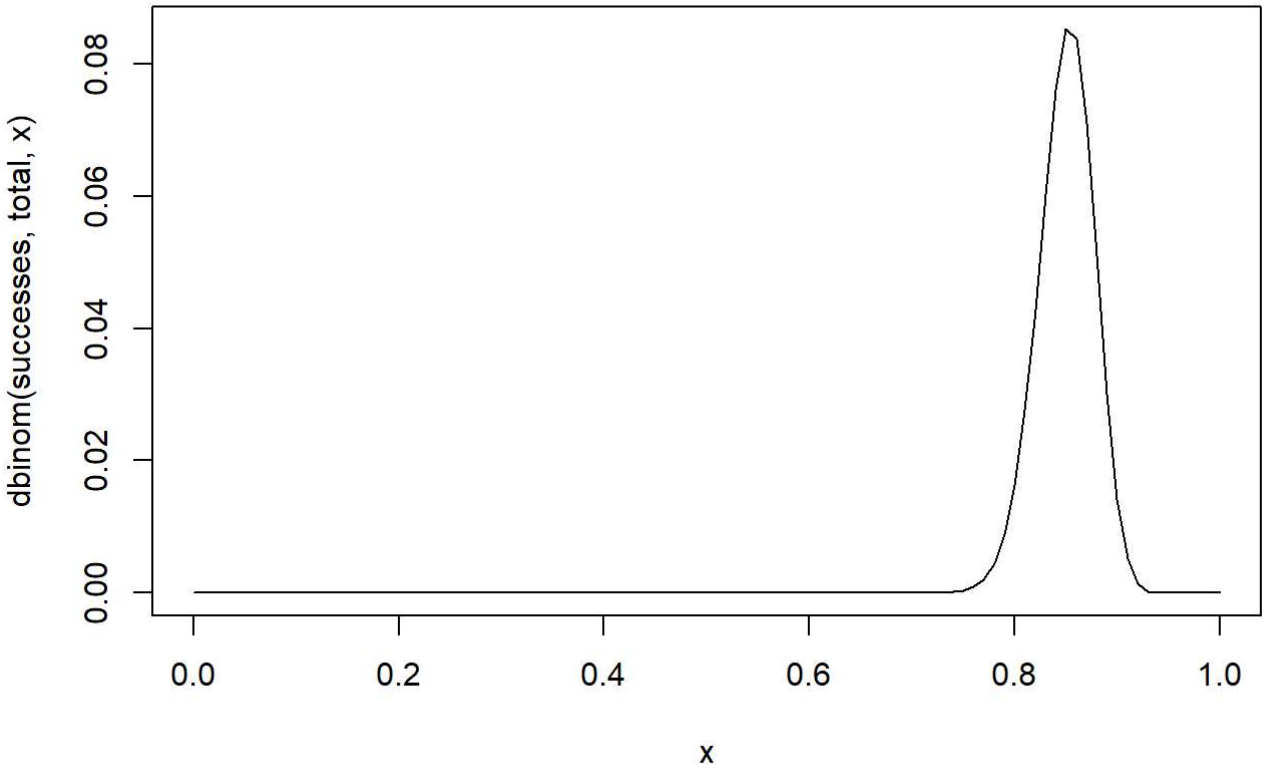
Figure 3 visualizes the binomial likelihood distribution.

```
table(dat$binary)
```

```
##
##   0   1
##  25 146
```

```
s <- 146
n <- 171

# from (Coghlan, 2010)

calcLikelihoodForProportion <- function(successes, total)
  {
     curve(dbinom(successes,total,x)) # plot the likelihood
     title(main = "Figure 3. Likelihood")
  }

calcLikelihoodForProportion(s, n)
```

Figure 3. Likelihood



Posterior Distribution

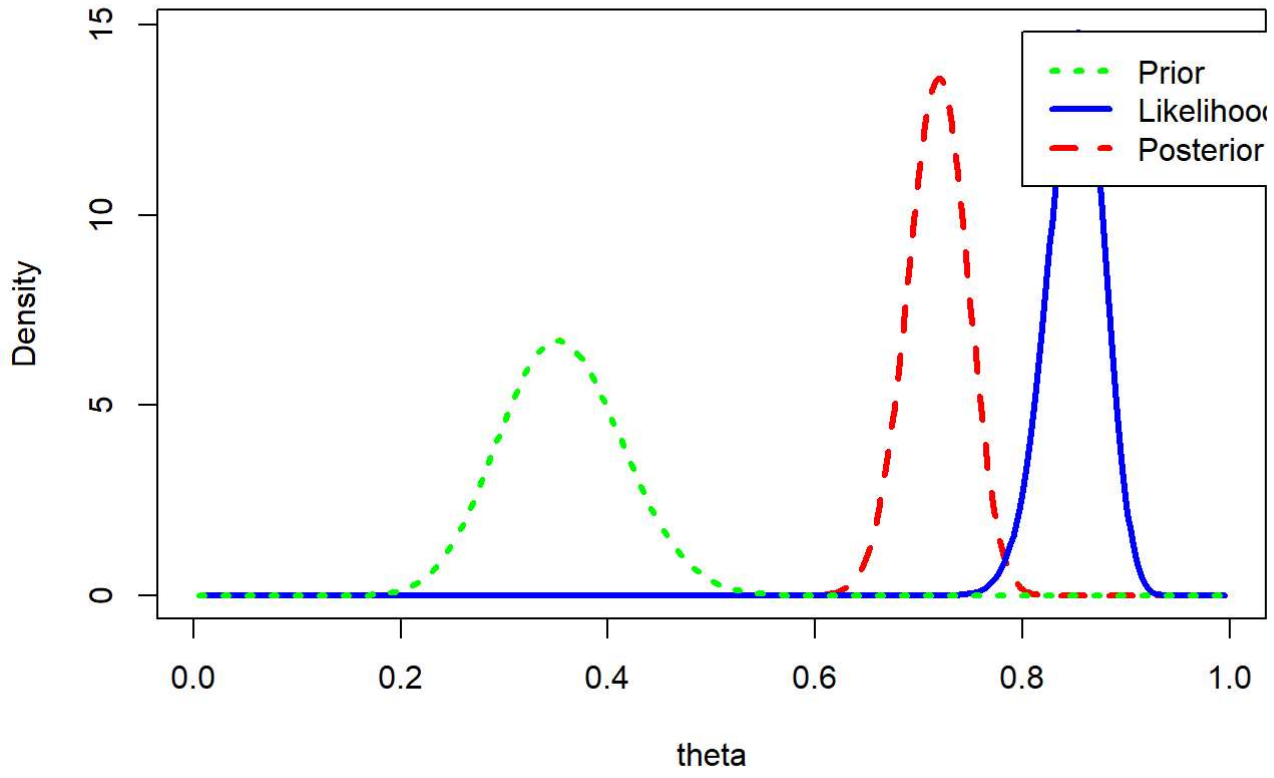
From there, I could make use of both the Beta Prior and Binomial Likelihood distributions' hyperparameters to find the posterior distribution, using another already-existent function (Coghlan, 2010).

Figure 4 visualizes the posterior distribution over the likelihood and prior distributions.

Here, it makes sense that the posterior falls between the prior and likelihood distributions. The prior, being weakly informative, has a wider distribution than the likelihood and posterior. Luckily, in adding the binary variable in for popularity, I could make use of a conjugate family of distributions, so I was able to know to use the beta-binomial family of conjugates.

```
# from (Coghlan, 2010)

calcPosteriorForProportion <- function(successes, total, a, b)
  {
     # Adapted from triplot() in the LearnBayes package
     # Plot the prior, likelihood and posterior:
     likelihood_a = successes + 1; likelihood_b = total - successes + 1
     posterior_a = a + successes;  posterior_b = b + total - successes
     theta = seq(0.005, 0.995, length = 500)
     prior = dbeta(theta, a, b)
     likelihood = dbeta(theta, likelihood_a, likelihood_b)
     posterior  = dbeta(theta, posterior_a, posterior_b)
     m = max(c(prior, likelihood, posterior))
     plot(theta, posterior, type = "l", ylab = "Density", lty = 2, lwd = 3,
          main = "Figure 4. Prior, Likelihood, and Posterior", ylim = c(0, m), col = "red")
     lines(theta, likelihood, lty = 1, lwd = 3, col = "blue")
     lines(theta, prior, lty = 3, lwd = 3, col = "green")
     legend(x=0.8,y=m, c("Prior", "Likelihood", "Posterior"), lty = c(3, 1, 2),
          lwd = c(3, 3, 3), col = c("green", "blue", "red"))
     # Print out summary statistics for the prior, likelihood and posterior:
     calcBetaMode <- function(aa, bb) { BetaMode <- (aa - 1)/(aa + bb - 2); return(BetaMode); }
     calcBetaMean <- function(aa, bb) { BetaMean <- (aa)/(aa + bb); return(BetaMean); }
     calcBetaSd   <- function(aa, bb) { BetaSd <- sqrt((aa * bb)/(((aa + bb)^2) * (aa + bb +
1))); return(BetaSd); }
     prior_mode      <- calcBetaMode(a, b)
     likelihood_mode <- calcBetaMode(likelihood_a, likelihood_b)
     posterior_mode  <- calcBetaMode(posterior_a, posterior_b)
     prior_mean      <- calcBetaMean(a, b)
     likelihood_mean <- calcBetaMean(likelihood_a, likelihood_b)
     posterior_mean  <- calcBetaMean(posterior_a, posterior_b)
     prior_sd        <- calcBetaSd(a, b)
     likelihood_sd   <- calcBetaSd(likelihood_a, likelihood_b)
     posterior_sd    <- calcBetaSd(posterior_a, posterior_b)
     print(paste("mode for prior=",prior_mode,", for likelihood=",likelihood_mode,", for posteri
or=",posterior_mode))
     print(paste("mean for prior=",prior_mean,", for likelihood=",likelihood_mean,", for posteri
or=",posterior_mean))
     print(paste("sd for prior=",prior_sd,", for likelihood=",likelihood_sd,", for posterior=",p
osterior_sd))
  }

calcPosteriorForProportion(s, n, a, b)
```

Figure 4. Prior, Likelihood, and Posterior




```
## [1] "mode for prior= 0.351152628352936 , for likelihood= 0.853801169590643 , for posterior=
0.718977770440338"
## [1] "mean for prior= 0.355755298927385 , for likelihood= 0.84971098265896 , for posterior= 0.
717119496683317"
## [1] "sd for prior= 0.0590729754796302 , for likelihood= 0.0270909736640688 , for posterior=
0.0292764009717511"
```

Frequentist Regression Models

Looking at the data through the Frequentist lense, I built a linear regression model both from the binary and the popularity variables.

First, I built a linear regression model using Binary as the response variable, with all of the other variables (not Popularity) as predictors. This yielded an adjusted $R^2 = .1927$, which is not very high at all, suggesting this model is not very fit for predicting popularity through the binary variable.

I used both-direction stepwise direction as a means of variable selection and then rebuilt the linear regression model using the yielded variables: liveness, loudness, speechiness, acousticness, duration, and energy. This new linear model yielded an adjusted $R^2 = .2082$, which, while it is higher than the first model, is still not high, suggesting that these variables are not good predictors for the binary popularity variable.

```
lm_bin <- lm(binary ~ duration+danceability+acousticness+energy+instrumentalness+liveness+loudne
ss+speechiness+valence+tempo,
             data = dat)
summary(lm_bin)
```

```
##
## Call:
## lm(formula = binary ~ duration + danceability + acousticness +
##     energy + instrumentalness + liveness + loudness + speechiness +
##     valence + tempo, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95956 -0.06764  0.08231  0.20931  0.56905
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.6345971  0.3734210   1.699  0.09118 .
## duration         -0.1163219  0.0442534  -2.629  0.00941 **
## danceability      0.1762267  0.2499690   0.705  0.48184
## acousticness     -0.2738617  0.1230119  -2.226  0.02739 *
## energy            0.4923170  0.2543816   1.935  0.05471 .
## instrumentalness -0.0433425  1.3433804  -0.032  0.97430
## liveness         -0.4124014  0.2993815  -1.378  0.17028
## loudness         -0.0867593  0.0173637  -4.997 1.52e-06 ***
## speechiness      -1.5936891  0.3172912  -5.023 1.35e-06 ***
## valence          -0.1268175  0.1768137  -0.717  0.47427
## tempo            -0.0003458  0.0008234  -0.420  0.67509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3182 on 160 degrees of freedom
## Multiple R-squared:  0.2411, Adjusted R-squared:  0.1937
## F-statistic: 5.083 on 10 and 160 DF,  p-value: 1.988e-06
```

```
ols_step_both_p(lm_bin)
```

```
##
##                                Stepwise Selection Summary
## ----------------------------------------------------------------------------------------
##                          Added/                   Adj.
## Step      Variable      Removed     R-Square    R-Square     C(p)        AIC        RMSE
## ----------------------------------------------------------------------------------------
##    1      liveness      addition       0.051       0.045    33.0770    126.4964    0.3462
##    2      loudness      addition       0.072       0.061    30.5720    124.6021    0.3433
##    3    speechiness     addition       0.127       0.111    21.1470    116.3096    0.3341
##    4    acousticness    addition       0.176       0.156    12.7140    108.3354    0.3255
##    5      duration      addition       0.221       0.198     5.1480    100.6496    0.3174
##    6       energy       addition       0.236       0.208     4.0470     99.3884    0.3153
## ----------------------------------------------------------------------------------------
```

```
lm_bin2 <- lm(binary ~ liveness+loudness+speechiness+acousticness+duration+energy, data=dat)
summary(lm_bin2)
```

```
##
## Call:
## lm(formula = binary ~ liveness + loudness + speechiness + acousticness +
##     duration + energy, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.96193 -0.05220  0.08836  0.19784  0.55835
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.68442    0.29534   2.317  0.02172 *
## liveness     -0.40348    0.29266  -1.379  0.16987
## loudness     -0.08826    0.01707  -5.169 6.77e-07 ***
## speechiness  -1.60887    0.31000  -5.190 6.15e-07 ***
## acousticness -0.30865    0.11511  -2.681  0.00808 **
## duration     -0.11314    0.04039  -2.801  0.00571 **
## energy        0.39774    0.22383   1.777  0.07742 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3153 on 164 degrees of freedom
## Multiple R-squared:  0.2361, Adjusted R-squared:  0.2082
## F-statistic:  8.45 on 6 and 164 DF,  p-value: 5.361e-08
```

Then, I built a linear regression model using Popularity as the response variable and the remaining variables (not Binary) as predictors. This yielded an adjusted $R^2 = .3415$, which is also not very high. It is higher than the linear model with the Binary response variable, but it is still not very fit for this data set.

After using both-direction stepwise regression to select variables (liveness, speechiness, and duration), I built another linear regression model, which yielded an adjusted $R^2 = .3224$. This is also not very high and is actually slightly lower than the model built with all of the predictors, which suggests that regular linear regression may not be the best fit for building a prediction model for song popularity.

```
lm_pop <- lm(popularity ~ duration+danceability+acousticness+energy+instrumentalness+liveness+lo
udness+speechiness+valence+tempo,
             data = dat)
summary(lm_pop)
```

```
##
## Call:
## lm(formula = popularity ~ duration + danceability + acousticness +
##     energy + instrumentalness + liveness + loudness + speechiness +
##     valence + tempo, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.312  -5.722  -0.175   5.645  32.429
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.424e+01  1.134e+01   5.667 6.62e-08 ***
## duration         -3.793e+00  1.344e+00  -2.823 0.005357 **
## danceability      9.749e+00  7.589e+00   1.285 0.200794
## acousticness     -6.043e+00  3.735e+00  -1.618 0.107592
## energy            1.248e+01  7.723e+00   1.616 0.108074
## instrumentalness -9.550e+00  4.079e+01  -0.234 0.815175
## liveness         -3.224e+01  9.089e+00  -3.547 0.000512 ***
## loudness         -1.529e+00  5.272e-01  -2.900 0.004257 **
## speechiness      -6.548e+01  9.633e+00  -6.798 2.00e-10 ***
## valence          -3.184e+00  5.368e+00  -0.593 0.553932
## tempo             1.164e-04  2.500e-02   0.005 0.996290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.66 on 160 degrees of freedom
## Multiple R-squared:  0.3802, Adjusted R-squared:  0.3415
## F-statistic: 9.817 on 10 and 160 DF,  p-value: 1.045e-12
```

```
ols_step_both_p(lm_pop)
```

```
##
##                                Stepwise Selection Summary
## ----------------------------------------------------------------------------------------
##                          Added/                   Adj.
## Step     Variable       Removed     R-Square    R-Square     C(p)         AIC         RMSE
## ----------------------------------------------------------------------------------------
##    1    speechiness    addition       0.229       0.224    32.1150    1292.9684    10.4857
##    2     liveness      addition       0.292       0.284    17.6860    1280.2434    10.0736
##    3     duration      addition       0.334       0.322     8.8490    1271.7858     9.7995
## ----------------------------------------------------------------------------------------
```

```
lm_pop2 <- lm(popularity ~ liveness+speechiness+duration, data=dat)
summary(lm_pop2)
```

```
##
## Call:
## lm(formula = popularity ~ liveness + speechiness + duration,
##     data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -23.2588  -5.0095  -0.0945   5.6004  29.6170
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   85.602      5.241  16.333  < 2e-16 ***
## liveness     -35.613      8.912  -3.996 9.65e-05 ***
## speechiness  -53.353      8.255  -6.463 1.08e-09 ***
## duration      -3.975      1.225  -3.245  0.00142 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.799 on 167 degrees of freedom
## Multiple R-squared:  0.3343, Adjusted R-squared:  0.3224
## F-statistic: 27.96 on 3 and 167 DF,  p-value: 1.056e-14
```

Bayesian Regression Models

Then, looking at the data through the Bayesian lense, I built a linear regression model, fitting the data to the binomial family, to the Binary response variable and using everything except Popularity as predictors. From there, I could extract a prior from the model, taking the student_t distribution student_t(3, 1, 2.5) from the extracted prior to use within the new model.

```
mod_prior1 <- stan_glm(binary ~ duration+danceability+acousticness+energy+instrumentalness+liven
ess+loudness+speechiness+valence+tempo,
               data = dat,
               family = binomial(link = "logit"),
               QR = T,
               seed = 1,
               refresh = 0)
```

Looking at the model summary, it shows negative relationships with all variables, except Tempo that seems to not have a relationship and Instrumentalness that seems to have an unusually strong positive relationship, with its mean estimated coefficient being significantly larger than any other variables' coefficients. Figure 5 visualizes the estimated coefficients for the predictors, and it is easy to see Instrumentalness is significantly larger and wider-spread than the rest of the covariates.

```
get_prior(mod_prior1, data = dat)
```

```
##                 prior     class             coef group resp dpar nlpar lb ub
##                (flat)         b
##                (flat)         b     acousticness
##                (flat)         b     danceability
##                (flat)         b         duration
##                (flat)         b           energy
##                (flat)         b instrumentalness
##                (flat)         b         liveness
##                (flat)         b         loudness
##                (flat)         b      speechiness
##                (flat)         b            tempo
##                (flat)         b          valence
##  student_t(3, 1, 2.5) Intercept
##  student_t(3, 0, 2.5)     sigma                                       0
##        source
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
##       default
```

```
t1 <- student_t(3, 1, 2.5)

y1 <- dat$binary

mod1 <- stan_glm(binary ~ duration+danceability+acousticness+energy+instrumentalness+liveness+lo
udness+speechiness+valence+tempo,
              data = dat,
              family = binomial(link = "logit"),
              prior = t1,
              prior_intercept = t1,
              QR = T,
              seed = 1,
              refresh = 0)


mod1
```

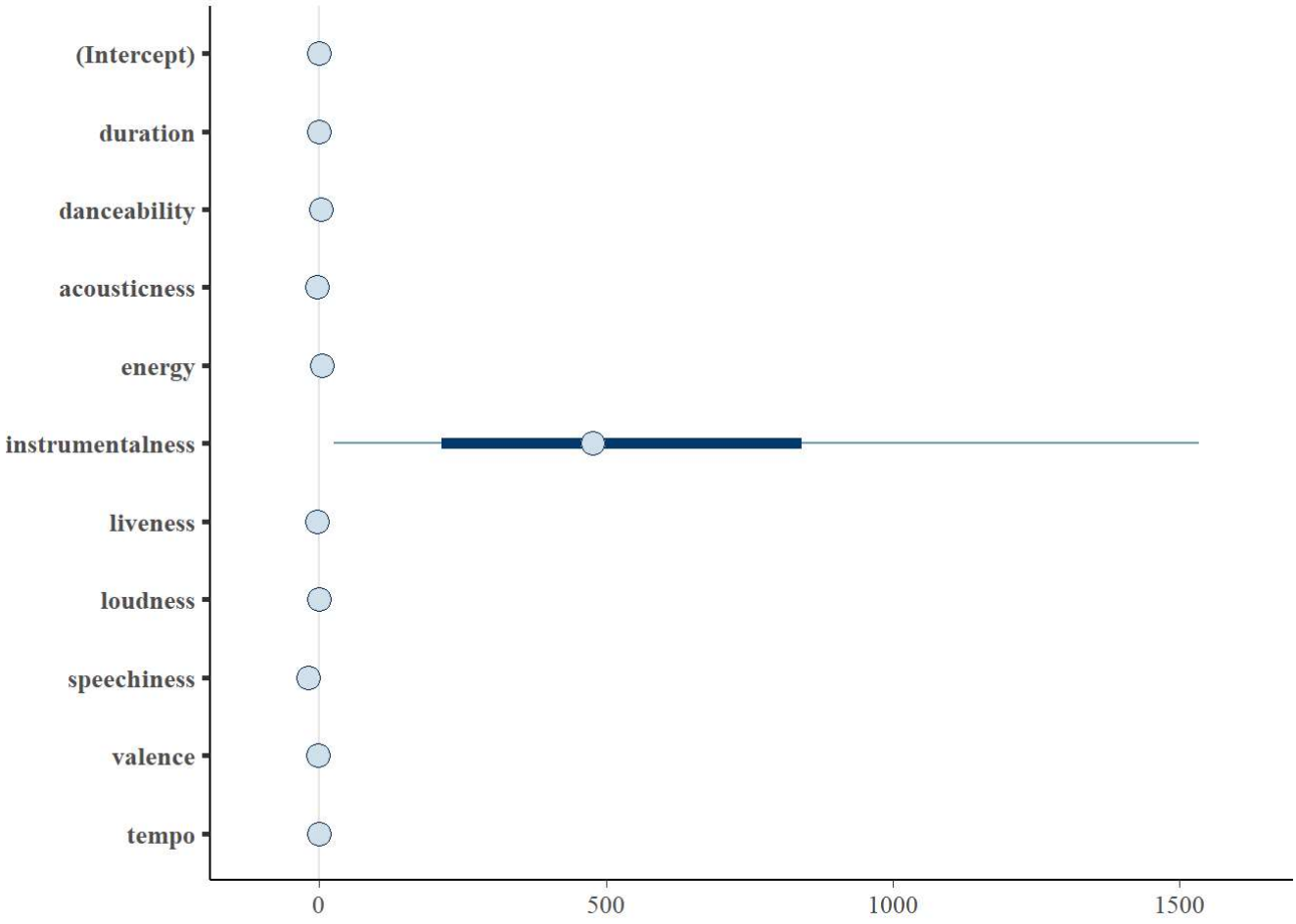
```
## stan_glm
##  family:       binomial [logit]
##  formula:      binary ~ duration + danceability + acousticness + energy + instrumentalness +
##     liveness + loudness + speechiness + valence + tempo
##  observations: 171
##  predictors:   11
## ------
##                  Median MAD_SD
## (Intercept)       -0.8    4.9
## duration          -1.3    0.5
## danceability       2.9    3.3
## acousticness      -4.2    1.6
## energy             5.0    3.2
## instrumentalness 476.3  446.5
## liveness          -3.5    3.1
## loudness          -1.2    0.3
## speechiness      -19.4    4.2
## valence           -1.9    2.0
## tempo              0.0    0.0
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
summary(mod1)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       binomial [logit]
##  formula:      binary ~ duration + danceability + acousticness + energy + instrumentalness +
##     liveness + loudness + speechiness + valence + tempo
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 171
##  predictors:   11
##
## Estimates:
##                    mean   sd     10%    50%    90%
## (Intercept)        -0.9    4.8   -7.0   -0.8    5.1
## duration           -1.3    0.5   -2.0   -1.3   -0.7
## danceability        3.0    3.3   -1.2    2.9    7.2
## acousticness       -4.3    1.6   -6.4   -4.2   -2.2
## energy              5.1    3.3    0.9    5.0    9.4
## instrumentalness  594.9  525.4   69.8  476.3 1251.1
## liveness           -3.5    3.0   -7.3   -3.5    0.3
## loudness           -1.2    0.3   -1.6   -1.2   -0.9
## speechiness       -19.6    4.4  -25.4  -19.4  -14.1
## valence            -1.9    1.9   -4.3   -1.9    0.6
## tempo               0.0    0.0    0.0    0.0    0.0
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.9    0.0  0.8   0.9   0.9
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable
(for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                  mcse Rhat n_eff
## (Intercept)       0.1  1.0 3846
## duration          0.0  1.0 3650
## danceability      0.1  1.0 4231
## acousticness      0.0  1.0 2993
## energy            0.1  1.0 3084
## instrumentalness 26.4  1.0  396
## liveness          0.1  1.0 3588
## loudness          0.0  1.0 1818
## speechiness       0.1  1.0 1833
## valence           0.0  1.0 3003
## tempo             0.0  1.0 2963
## mean_PPD          0.0  1.0 3618
## log-posterior     0.1  1.0 1021
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective
```

```
sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rh
at=1).
```

```
plot(mod1, main = "Figure 5. Bayesian Model - Binary")
```



I also looked at the posterior summary of the model, and even with Instrumentalness having a significantly higher coefficient than others, in the posterior summary, it is not the covariate with the highest probability of direction (pd) value, indicating that it is not the most statistically significant covariate in the model. Loudness and Speechiness both yielded $pd = 100$, which suggests those are the most statistically significant covariates.

For this model, the Bayesian $R^2 = .3850$, which while also not high, similar to the adjusted R^2 values seen in the Frequentist models, is higher than those, suggesting that the Bayesian model, while not very fit, may be better fit than the Frequentist models.

```
describe_posterior(mod1)
```

```
## Summary of Posterior Distribution
##
## Parameter        |    Median |            95% CI |     pd |          ROPE | % in ROPE |  Rhat
|     ESS
## ------------------------------------------------------------------------------------------
----------
## (Intercept)      |     -0.84 | [-10.09,    8.33] | 57.80% | [-0.10, 0.10] |     1.84% | 1.002
| 3846.00
## duration         |     -1.33 | [ -2.40,   -0.40] | 99.90% | [-0.10, 0.10] |        0% | 1.000
| 3650.00
## danceability     |      2.93 | [ -3.23,    9.36] | 81.62% | [-0.10, 0.10] |     2.03% | 1.001
| 4231.00
## acousticness     |     -4.24 | [ -7.60,   -1.16] | 99.75% | [-0.10, 0.10] |        0% | 1.000
| 2993.00
## energy           |      5.00 | [ -1.12,   11.95] | 94.58% | [-0.10, 0.10] |     0.95% | 1.001
| 3084.00
## instrumentalness |    476.28 | [  0.57, 1858.65] | 97.52% | [-0.10, 0.10] |        0% | 1.011
|  396.00
## liveness         |     -3.45 | [ -9.47,    2.36] | 87.40% | [-0.10, 0.10] |     1.63% | 0.999
| 3588.00
## loudness         |     -1.23 | [ -1.83,   -0.73] |   100% | [-0.10, 0.10] |        0% | 1.002
| 1818.00
## speechiness      |    -19.45 | [-28.55,  -11.49] |   100% | [-0.10, 0.10] |        0% | 1.000
| 1833.00
## valence          |     -1.89 | [ -5.66,    1.82] | 83.23% | [-0.10, 0.10] |     2.61% | 1.000
| 3003.00
## tempo            | -4.71e-03 | [ -0.02,    0.01] | 70.15% | [-0.10, 0.10] |      100% | 1.002
| 2963.00
```

```
r2_bayes(mod1, robust = TRUE, ci = 0.95, verbose = TRUE)
```

```
## # Bayesian R2 with Compatibility Interval
##
##   Conditional R2: 0.385 (95% CI [0.247, 0.507])
```

```
hdi(mod1)
```

```
## Highest Density Interval
##
## Parameter        |           95% HDI
## ------------------------------------
## (Intercept)      | [-10.09,    8.33]
## duration         | [ -2.38,   -0.39]
## danceability     | [ -3.14,    9.38]
## acousticness     | [ -7.81,   -1.41]
## energy           | [ -1.79,   11.07]
## instrumentalness | [-33.78, 1535.12]
## liveness         | [ -9.57,    2.19]
## loudness         | [ -1.81,   -0.71]
## speechiness      | [-28.03,  -11.10]
## valence          | [ -5.48,    1.95]
## tempo            | [ -0.02,    0.01]
```

Looking at accuracy of the model, I used both general prediction accuracy (finding the percent of correctly predicted values from the model), balanced classification accuracy (which looks at both successes and failures), leave-one-out cross-validation accuracy, and leave-one-out cross-validation balanced accuracy (Aki Vehtari, 2022).

The accuracies were:

- Classification Accuracy = .88
- Balanced Classification Accuracy = .81
- LOO Accuracy = .86
- LOO Balanced Accuracy = .82

Since general classification accuracy tends to be somewhat overoptimistic in calculation, I look at the leave-one-out cross-validation balanced accuracy (.82) for the more accurate accuracy estimation, which seems to indicate, even with a lower Bayesian R^2, that this model is very accurate without being overfit to the data.

Figure 6 visualizes the posterior predicted values and the LOO predictive probabilities. Looking at the plot, it looks like the data follows a trend but isn't perfectly matched, which tracks with the 81% accuracy.

```
# calculations from (Aki Vehtari, 2022)

# Predicted probabilities
linpred <- posterior_linpred(mod1)
preds <- posterior_epred(mod1)
pred <- colMeans(preds)
pr <- as.integer(pred >= 0.5)

# posterior classification accuracy
round(mean(xor(pr,as.integer(y1==0))),2)
```

```
## [1] 0.88
```

```
# posterior balanced classification accuracy
round((mean(xor(pr[y1==0]>0.5,as.integer(y1[y1==0])))+mean(xor(pr[y1==1]<0.5,as.integer(y1[y1==
1])))))/2,2)
```

```
## [1] 0.81
```

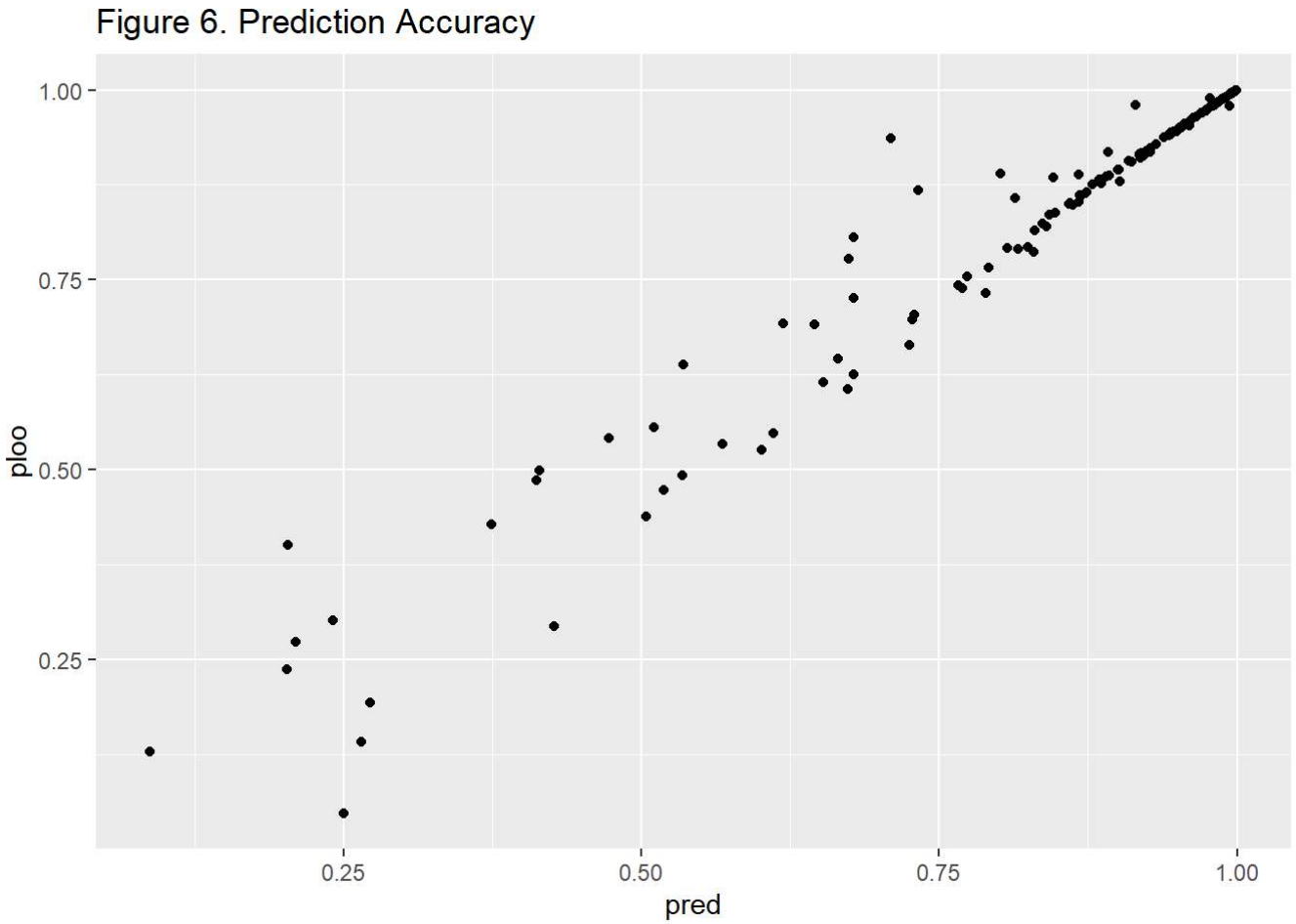
```
loo1 <- loo(mod1, save_psis = TRUE, k_threshold = 0.7)

# LOO predictive probabilities
ploo = E_loo(preds, loo1$psis_object, type="mean", log_ratios = -log_lik(mod1))$value

# LOO classification accuracy
round(mean(xor(ploo>0.5,as.integer(y1==0))),2)
```

```
## [1] 0.86
```

```
# LOO balanced classification accuracy
round((mean(xor(ploo[y1==0]>0.5,as.integer(y1[y1==0])))+mean(xor(ploo[y1==1]<0.5,as.integer(y1[y1==1])))))/2,2)
```

```
## [1] 0.82
```

```
qplot(pred, ploo, main = "Figure 6. Prediction Accuracy")
```



I followed a similar process in building a second model, using Popularity as the response variable and fitting it with a Gaussian distribution, since Popularity is not binary. I extracted the student_t distribution student_t(3, 63, 7.4) from the extracted prior to use within the new model.

```
mod_prior2 <- stan_glm(popularity ~ duration+danceability+acousticness+energy+instrumentalness+l
iveness+loudness+speechiness+valence+tempo,
              data = dat,
              family = gaussian(link="identity"),
              QR = T,
              seed = 1,
              refresh = 0)
```

This model summary shows varying positive and negative estimated coefficients for the covariates, except Tempo, which again seems to have no relationship in the model. Speechiness and the Intercept seem to be somewhat higher than the other coefficients.

Figure 7 visualizes the estimated coefficients for the predictors, and it is easy to see Instrumentalness still is significantly wider-spread than the rest of the covariates, but the other covariates are stronger in prediction, which makes sense in the context of music popularity.

```
get_prior(mod_prior2, data = dat)
```

```
##                  prior     class             coef group resp dpar nlpar lb ub
##                 (flat)         b
##                 (flat)         b     acousticness
##                 (flat)         b     danceability
##                 (flat)         b         duration
##                 (flat)         b           energy
##                 (flat)         b instrumentalness
##                 (flat)         b         liveness
##                 (flat)         b         loudness
##                 (flat)         b      speechiness
##                 (flat)         b            tempo
##                 (flat)         b          valence
##  student_t(3, 63, 7.4) Intercept
##   student_t(3, 0, 7.4)     sigma                                       0
##        source
##       default
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##  (vectorized)
##       default
##       default
```

```
t2 <-student_t(3, 63, 7.4)

y2 <- dat$popularity

mod2 <- stan_glm(popularity ~ duration+danceability+acousticness+energy+instrumentalness+livenes
s+loudness+speechiness+valence+tempo,
              data = dat,
              family = gaussian(link="identity"),
              prior = t2,
              prior_intercept = t2,
              QR = T,
              seed = 1,
              refresh = 0)


mod2
```

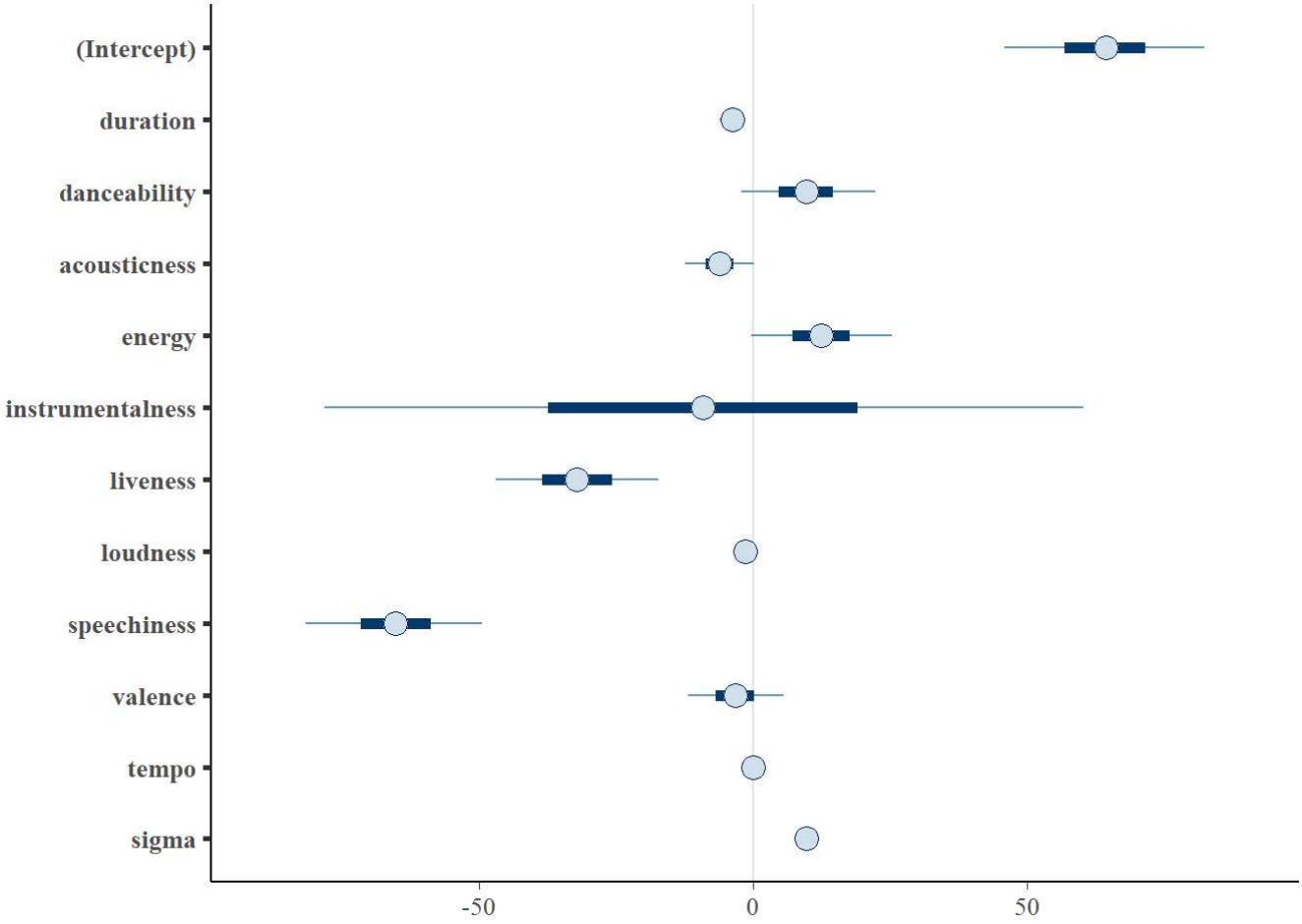
```
## stan_glm
##  family:       gaussian [identity]
##  formula:      popularity ~ duration + danceability + acousticness + energy +
##     instrumentalness + liveness + loudness + speechiness + valence +
##     tempo
##  observations: 171
##  predictors:   11
## ------
##                   Median MAD_SD
## (Intercept)        64.3   11.0
## duration           -3.8    1.4
## danceability        9.8    7.2
## acousticness       -6.1    3.8
## energy             12.4    7.8
## instrumentalness   -9.3   42.0
## liveness          -32.2    9.4
## loudness           -1.5    0.5
## speechiness       -65.3    9.5
## valence            -3.3    5.2
## tempo               0.0    0.0
##
## Auxiliary parameter(s):
##       Median MAD_SD
## sigma 9.7    0.5
##
## ------
## * For help interpreting the printed output see ?print.stanreg
## * For info on the priors used see ?prior_summary.stanreg
```

```
summary(mod2)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      popularity ~ duration + danceability + acousticness + energy +
##     instrumentalness + liveness + loudness + speechiness + valence +
##     tempo
##  algorithm:    sampling
##  sample:       4000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 171
##  predictors:   11
##
## Estimates:
##                    mean   sd    10%   50%   90%
## (Intercept)       64.2   11.1  50.2  64.3  78.3
## duration          -3.8    1.4  -5.6  -3.8  -2.0
## danceability       9.8    7.4   0.3   9.8  19.3
## acousticness      -6.1    3.8 -10.9  -6.1  -1.2
## energy            12.4    7.8   2.6  12.4  22.7
## instrumentalness  -9.0   42.3 -62.7  -9.3  44.6
## liveness         -32.2    9.0 -44.0 -32.2 -20.7
## loudness          -1.5    0.5  -2.2  -1.5  -0.9
## speechiness      -65.4    9.7 -77.9 -65.3 -53.0
## valence           -3.3    5.3 -10.1  -3.3   3.6
## tempo              0.0    0.0   0.0   0.0   0.0
## sigma              9.7    0.6   9.0   9.7  10.4
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 61.2    1.1 59.9  61.2  62.6
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable
(for details see help('summary.stanreg')).
##
## MCMC diagnostics
##                  mcse Rhat n_eff
## (Intercept)      0.1  1.0  6367
## duration         0.0  1.0  6427
## danceability     0.1  1.0  5885
## acousticness     0.0  1.0  7018
## energy           0.1  1.0  6793
## instrumentalness 0.5  1.0  5916
## liveness         0.1  1.0  5742
## loudness         0.0  1.0  6270
## speechiness      0.1  1.0  6085
## valence          0.1  1.0  5555
## tempo            0.0  1.0  5507
## sigma            0.0  1.0   578
## mean_PPD         0.0  1.0   912
## log-posterior    0.1  1.0   818
##
```

```
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence Rh
```

```
plot(mod2, main = "Figure 7. Bayesian Model - Popularity")
```



The posterior summary indicates that the Intercept, Liveness, and Speechiness all are the most statistically significant covariates in the model, each yielding $pd = 100$.

For this model, the Bayesian $R^2 = .3850$, which is the same as the previous model built around the Binary response variable, which suggests both of these Bayesian regression models are stronger fits than the Frequentist but maybe not the best since the R^2 is still relatively small.

```
describe_posterior(mod2)
```

```
## Summary of Posterior Distribution
##
## Parameter        |   Median |           95% CI |     pd |          ROPE | % in ROPE |  Rhat |
ESS
## ------------------------------------------------------------------------------------------
--------
## (Intercept)      |    64.34 | [ 42.38,  86.16] |   100% | [-0.10, 0.10] |        0% | 0.999 |
6367.00
## duration         |    -3.77 | [ -6.49,  -1.13] | 99.80% | [-0.10, 0.10] |        0% | 1.000 |
6427.00
## danceability     |     9.76 | [ -4.86,  25.12] | 90.75% | [-0.10, 0.10] |     0.34% | 1.000 |
5885.00
## acousticness     |    -6.10 | [-13.71,   1.49] | 94.33% | [-0.10, 0.10] |     0.66% | 0.999 |
7018.00
## energy           |    12.37 | [ -2.75,  27.58] | 94.65% | [-0.10, 0.10] |     0.18% | 0.999 |
6793.00
## instrumentalness |    -9.28 | [-89.59,  74.14] | 59.08% | [-0.10, 0.10] |     0.29% | 1.000 |
5916.00
## liveness         |   -32.19 | [-49.85, -14.65] |   100% | [-0.10, 0.10] |        0% | 0.999 |
5742.00
## loudness         |    -1.52 | [ -2.58,  -0.47] | 99.65% | [-0.10, 0.10] |        0% | 1.000 |
6270.00
## speechiness      |   -65.32 | [-85.19, -46.64] |   100% | [-0.10, 0.10] |        0% | 0.999 |
6085.00
## valence          |    -3.30 | [-13.59,   7.41] | 73.90% | [-0.10, 0.10] |     1.16% | 1.000 |
5555.00
## tempo            | 4.13e-04 | [ -0.05,   0.05] | 50.78% | [-0.10, 0.10] |      100% | 0.999 |
5507.00
```

```
r2_bayes(mod2, robust = TRUE, ci = 0.95, verbose = TRUE)
```

```
## # Bayesian R2 with Compatibility Interval
##
##   Conditional R2: 0.385 (95% CI [0.285, 0.486])
```

```
hdi(mod2)
```

```
## Highest Density Interval
##
## Parameter        |          95% HDI
## ----------------------------------
## (Intercept)      | [ 42.52,  86.19]
## duration         | [ -6.41,  -1.09]
## danceability     | [ -5.92,  23.87]
## acousticness     | [-13.79,   1.37]
## energy           | [ -2.50,  27.66]
## instrumentalness | [-90.58,  73.35]
## liveness         | [-50.66, -15.75]
## loudness         | [ -2.60,  -0.51]
## speechiness      | [-84.90, -46.59]
## valence          | [-13.24,   7.64]
## tempo            | [ -0.05,   0.05]
```

Using the same accuracy calculations as the previous model, the accuracies were:

- Classification Accuracy = .98
- Balanced Classification Accuracy = NA
- LOO Accuracy = .98
- LOO Balanced Accuracy = NA

The balanced accuracies were ≥ 1, which yields NA as the values, and the unbalanced accuracies were .98, which suggests that this model is much more accurate that the previous, but since it is so close to 100% accuracy, it may be overfit, which is also not the best.

Figure 8 visualizes the posterior predicted values and the LOO predictive probabilities. Looking at the plot, it is much more closely fit, with only a few potential outliers, which suggests significant overiftting.

```
#calculations from (Aki Vehtari, 2022)

# Predicted probabilities
linpred <- posterior_linpred(mod2)
preds <- posterior_epred(mod2)
pred <- colMeans(preds)
pr <- as.integer(pred >= 0.5)

# posterior classification accuracy
round(mean(xor(pr,as.integer(y2==0))),2)
```

```
## [1] 0.98
```

```
# posterior balanced classification accuracy
round((mean(xor(pr[y2==0]>0.5,as.integer(y2[y2==0])))+mean(xor(pr[y2==1]<0.5,as.integer(y2[y2==
1])))))/2,2)
```

```
## [1] NaN
```

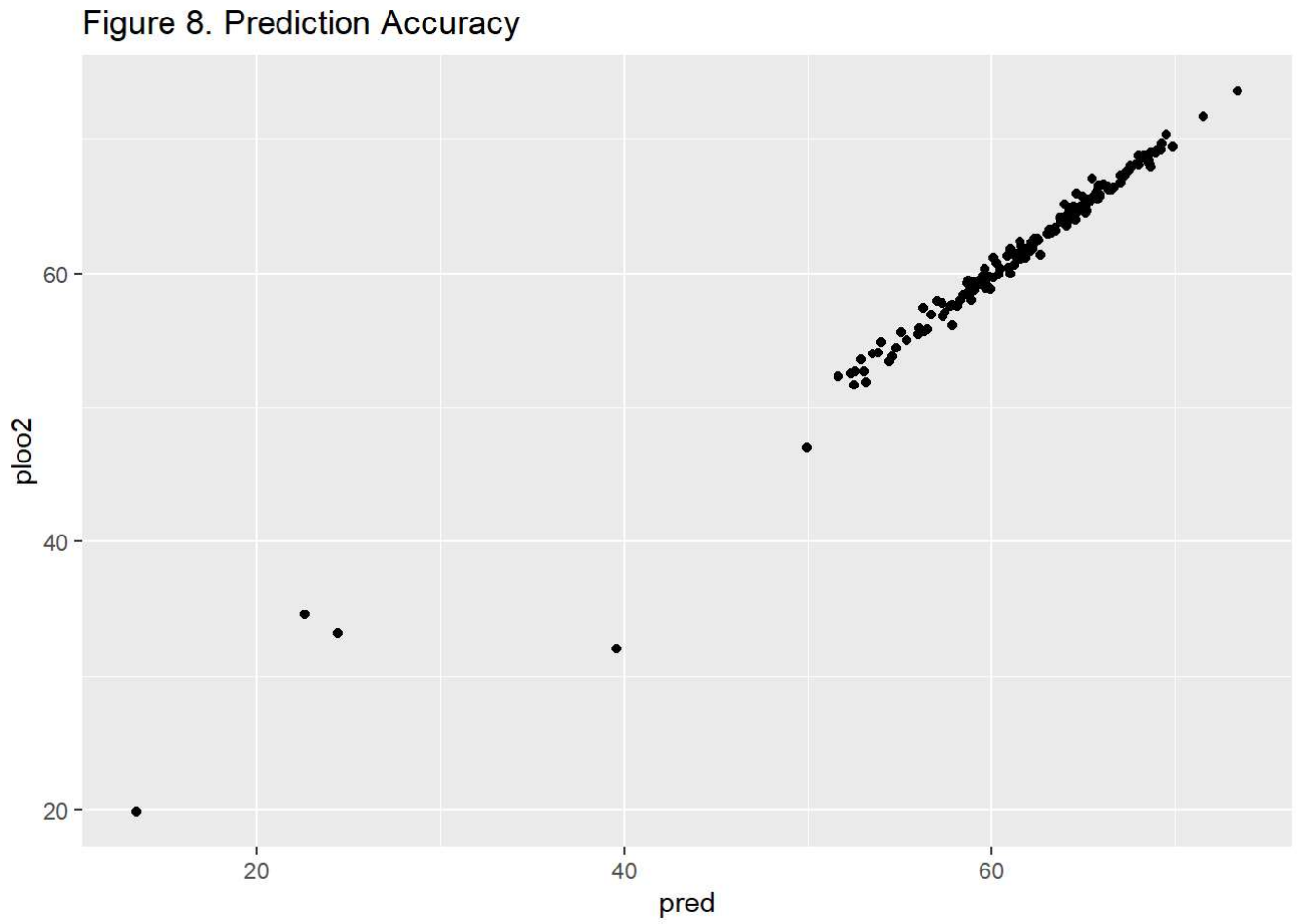
```
loo2 <- loo(mod2, save_psis = TRUE, k_threshold = 0.7)

# LOO predictive probabilities
ploo2 = E_loo(preds, loo2$psis_object, type="mean", log_ratios = -log_lik(mod2))$value

# LOO classification accuracy
round(mean(xor(ploo2>0.5,as.integer(y2==0))),2)
```

```
## [1] 0.98
```

```
# LOO balanced classification accuracy
round((mean(xor(ploo2[y2==0]>0.5,as.integer(y2[y2==0])))+mean(xor(ploo2[y2==1]<0.5,as.integer(y2
[y2==1])))))/2,2)
```

```
## [1] NaN
```

```
qplot(pred, ploo2, main = "Figure 8. Prediction Accuracy")
```



Results

Overall, in looking at the four different models, it seems as though the Bayesian regression models were better-fit for predicting song popularity.

Both the Bayesian models had higher R^2 than the Frequentist models, which is likely due to the addition of the probability distributions into the model to help in building the predictions. The Bayesian model that used Popularity as the response variable had higher prediction accuracy than that of the Bayesian model that used Binary as the response variable. However, this may be due to overfitting, so in choosing one of the models presented to be the best-fit, I would choose the Popularity Bayesian model.

Unfortunately, looking at the different models, the prior/likelihood/posterior distributions, and even the correlation plot, it looks like the predictors in the data set are not helpful in predicting song popularity. I did use a simple linear formula for my regression models, so it is very probable that these predictors could better-predict popularity with a more complex model, which could be something to investigate in future analysis with this data.

It is also possible that not all of the predictors are as helpful in contributing to popularity as others, and even though I used stepwise regression for the Frequentist models for variable selection, it's possible there are better variable selection techniques to explore with this data and model building in future analysis.

Something else to consider that may not be possible to model with the given predictors is that there is probably bias in popularity calculations due to them being calculated by Spotify based on how many times each song is listened to by Spotify users. Not everyone uses Spotify, and Spotify tends to be used by younger generations, so data collected is only being calculated from a subset of the whole population of music fans. Another possible bias is that Taylor Swift is world-famous and relevant in the media currently, especially with her many public relationship controversies and her major tours and album drops. Her status as one of the top recording artists in the world in history sets her up to be more popular simply due to her name being attached to the songs and not necessarily due to the songs meeting popularity ratings like other songs from other artists. Another potential angle to analyze these models from may be to look at data not only from Taylor Swift but from Spotify music in general or maybe from music played on other streaming platforms to compare and fine-tune the models based on larger less-biased populations.

References

Aki Vehtari, J. G. (2022, February 21). *Bayesian Logistic Regression with rstanarm*. Retrieved from Aki Vehtari: Model Selection: https://avehtari.github.io/modelselection/diabetes.html

Clark, T. (2022, November 19). *The 50 Best-Selling Music Artists of All Time*. Retrieved from Business Insider: https://www.businessinsider.com/best-selling-music-artists-of-all-time-2016-9

Coghlan, A. (2010). *Using R for Bayesian Statistics*. Retrieved from A Little Book of R for Bayesian Statistics: https://a-little-book-of-r-for-bayesian-statistics.readthedocs.io/en/latest/src/bayesianstats.html

Dagohoy, J. L. (2021, November 6). *Taylor Swift Spotify Data*. Retrieved from Kaggle: https://www.kaggle.com/datasets/thespacefreak/taylor-swift-spotify-data

Spotify. (2022). *Get Track's Audio Analysis*. Retrieved from Spotify for Developers: https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-analysis

Taylor Swift. (2022). *Taylor Swift*. Retrieved from Spotify: https://open.spotify.com/artist/06HL4z0CvFAxyc27GXpf02