

A Bayesian Analysis of Spotify Data

Nathaniel Maxwell, Jessie Bierschenk

30 April, 2021

Introduction

For many musicians, the art of composing/performing/marketing a new song is an arduous process. Even after all the work has been completed and a song is ready to be played to the public, the biggest uncertainty still awaits: How will the song be received? Will it become a hit? Will it be a song that everyone skips over, or never becomes popular? The purpose of this analysis is to investigate which characteristics of a song (such as tempo, duration, mode, acousticness, etc.) would make it more “likeable,” less likely to be skipped, or more popular. Of course, music taste is a very subjective matter, and thus, there will be quite a bit of uncertainty around any variables that are deemed important/unimportant. What one person likes; another person may dislike. Therefore, looking at such musical characteristics through a Bayesian lens will help to quantify the uncertainty surrounding any of our findings. Through this analysis we hope to provide some conclusions that an aspiring musician (or even a well-established musician) can have at their disposal when creating new music.

Pre-Analysis

Data

Two datasets were utilized during this analysis.

1. The first dataset consists of 83,939 observations on Spotify of whether or not a track was skipped by users. In total, 65,417 different tracks were included in the dataset. Each track has the following characteristics:
 - (a) Release Year (Year the song was released)
 - (b) Duration (length of song in seconds)
 - (c) US Popularity Estimate (A popularity rating of song, on a scale 1-100)
 - (d) Acousticness (A confidence measure from 0-1 on whether the track is acoustic, where values near 1 represent high confidence that the track is acoustic)
 - (e) Beat Strength (The strength of the beat from 0-1, where 1 represents a very strong sense of beat)
 - (f) Bounciness (A rating of the bounciness from 0-1, where 1 represents a strong sense of bounciness)
 - (g) Danceability (A rating from 0-1 of how suitable the track is for dancing, where values near 1 represent high suitability)
 - (h) Energy (A rating from 0-1 representing a perceptual measure of intensity and activity, where values near 1 represent high energy)
 - (i) Instrumentalness (A rating from 0-1 that predicts whether a track has no vocals, where values close to 1 represent high confidence that there are no vocals)

- (j) Mode (Predicts whether or not a song is major or minor)
- (k) Speechiness (A rating from 0-1 that detects the presence of spoken words in a track, with values near 1 representing an exclusively speech-like track)
- (l) Tempo (The estimated tempo of the track in Beats Per Minute (BPM))
- (m) Valence (A rating from 0-1 that represents the positivity of the song, with 1 representing high positivity)
- (n) Skipped (Denotes whether or not that particular track was skipped or played the entire way through)

Note: in order to try to obtain tracks most representative of new music, only the following tracks were kept:

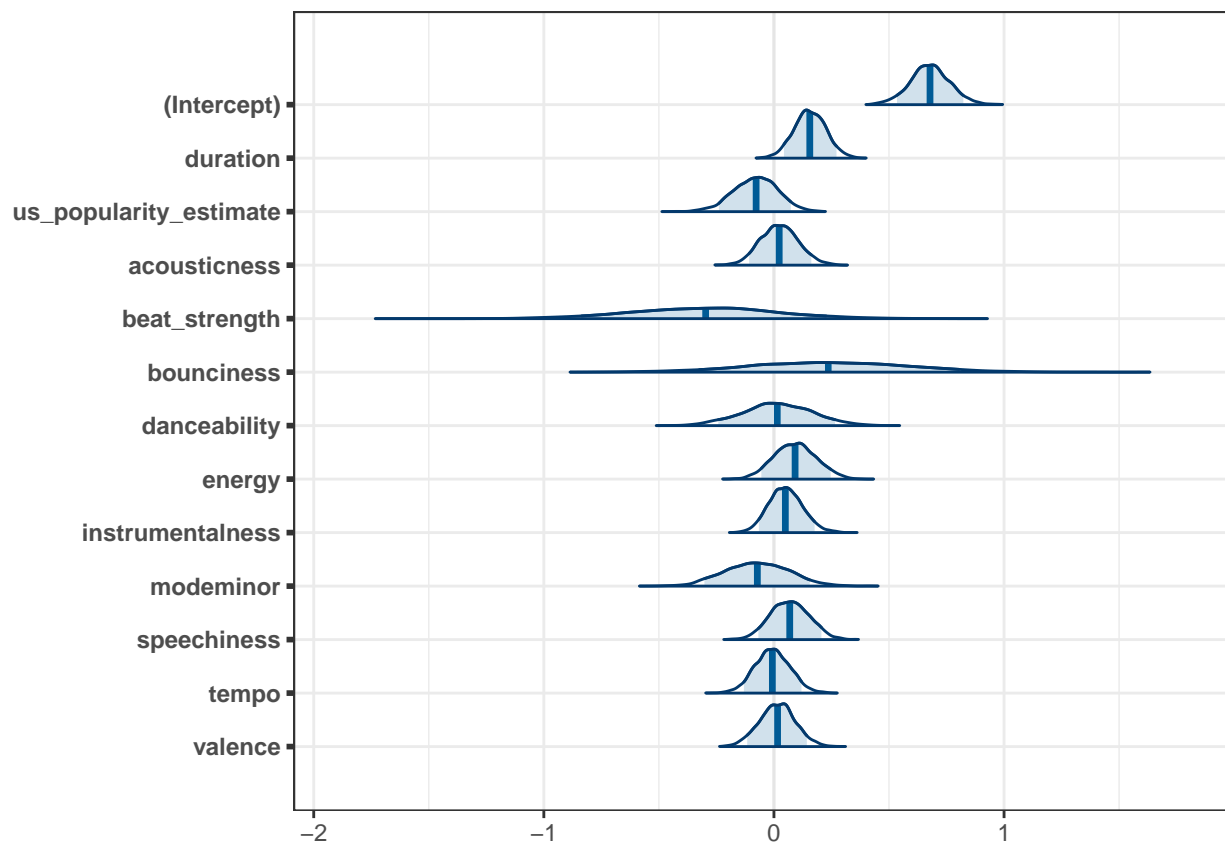
- (a) Tracks from 2010-present
 - (b) Tracks with a speechiness value ≤ 0.4 (filters out tracks that are mostly spoken, such as podcasts and ebooks)
 - (c) Tracks with an instrumentalness value ≤ 0.6 (filters out tracks that contain no vocals)
 - (d) Tracks with a duration ≤ 360 seconds (given that the average new song is 3-5 minutes, a cutoff of 6 minutes seemed appropriate)
2. The second dataset consisted of 2017 songs compiled by a single person, where a portion of the songs are songs that he likes, and the other portion are songs that he dislikes. This dataset includes similar variables as the first dataset, including:
- (a) Acousticness
 - (b) Danceability
 - (c) Duration
 - (d) Energy
 - (e) Instrumentalness
 - (f) Key (The particular grouping of chords and notes in a song)
 - (g) Liveness (rating from 0-1 of whether the track was performed live, with 1 representing high confidence the track was performed live)
 - (h) Loudness (Overall loudness of the track in decibels (dB))
 - (i) Mode
 - (j) Speechiness
 - (k) Tempo
 - (l) Time Signature (The way in which beats of the song are organized)
 - (m) Valence

Model Selection

For the first dataset, we wanted to estimate the values of the coefficients β for each of the variables to find out how they impact whether or not a track is skipped. We are assuming little knowledge about each variable's effect, so we propose a weakly informative prior for $[\beta]$: Using recommendations from Gelman, Jakulin, Pittau, and Su, we use a $\text{cauchy}(0, 2.5)$ prior for each scaled variable (we scaled the variables). Our response variable, \mathbf{y} , will follow a logistic regression model, where 1 means the track was skipped. This is equivalent to the Bernoulli distribution $\mathbf{y}|\theta \sim \text{Bern}(\theta)$. We will use the logit link, where $\text{logit}(\theta) = \eta$, and $\eta = \mathbf{x}^T \beta$, where \mathbf{x} is the covariate space for $\text{textbf{Y}}$. Using the `rstanarm` package, Rstudio will compute the posterior and draw MCMC samples from the posterior distribution $[\beta|\mathbf{Y}, \mathbf{X}]$.

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

Posterior Estimates



```
##              5%   95%
## (Intercept)    0.535 0.823
## duration        0.043 0.272
## us_popularity_estimate -0.239 0.073
## acousticness    -0.108 0.162
## beat_strength   -0.836 0.227
## bounciness      -0.324 0.804
## danceability    -0.242 0.266
## energy          -0.055 0.246
## instrumentalness -0.065 0.177
```

```

## modeminor          -0.302 0.156
## speechiness        -0.068 0.206
## tempo              -0.130 0.120
## valence             -0.116 0.143

##
## Computed from 4000 by 1000 log-likelihood matrix
##
##           Estimate   SE
## elpd_loo   -654.9 10.2
## p_loo       13.2  0.8
## looic      1309.8 20.3
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.

##
## Computed from 4000 by 1000 log-likelihood matrix
##
##           Estimate   SE
## elpd_loo   -648.5  9.3
## p_loo        1.0  0.0
## looic      1296.9 18.6
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.

##           elpd_diff se_diff
## posterior0  0.0        0.0
## posterior1 -6.5        3.7

```

After running the `rstanarm` function and including all of the variables, we see that there is only variable whose 90% confidence interval does not include 0. That variable is duration, and furthermore, when calculating the 'leave-one-out' cross-validation information criterion (looic), we see that this model actually has a *higher* value than the looic of a baseline model with no predictors. In other words, our model is worse at predicting whether or not a song is skipped than if someone randomly guessed! Therefore, we will drop all variables that were not deemed significant at a 90% confidence interval (included 0 in their posterior interval), and rerun the model. In this case, 'duration' is the only variable remaining.

```

posterior2 <- stan_glm(skipped ~ duration, data = Track_features_a,
                      family = binomial(link = "logit"),
                      prior = cauchy(0,2.5), prior_intercept = cauchy(0,2.5),
                      seed = seed,
                      refresh = 0)
(loo2 <- loo(posterior2, save_psis = TRUE))

```

```

##
## Computed from 4000 by 1000 log-likelihood matrix

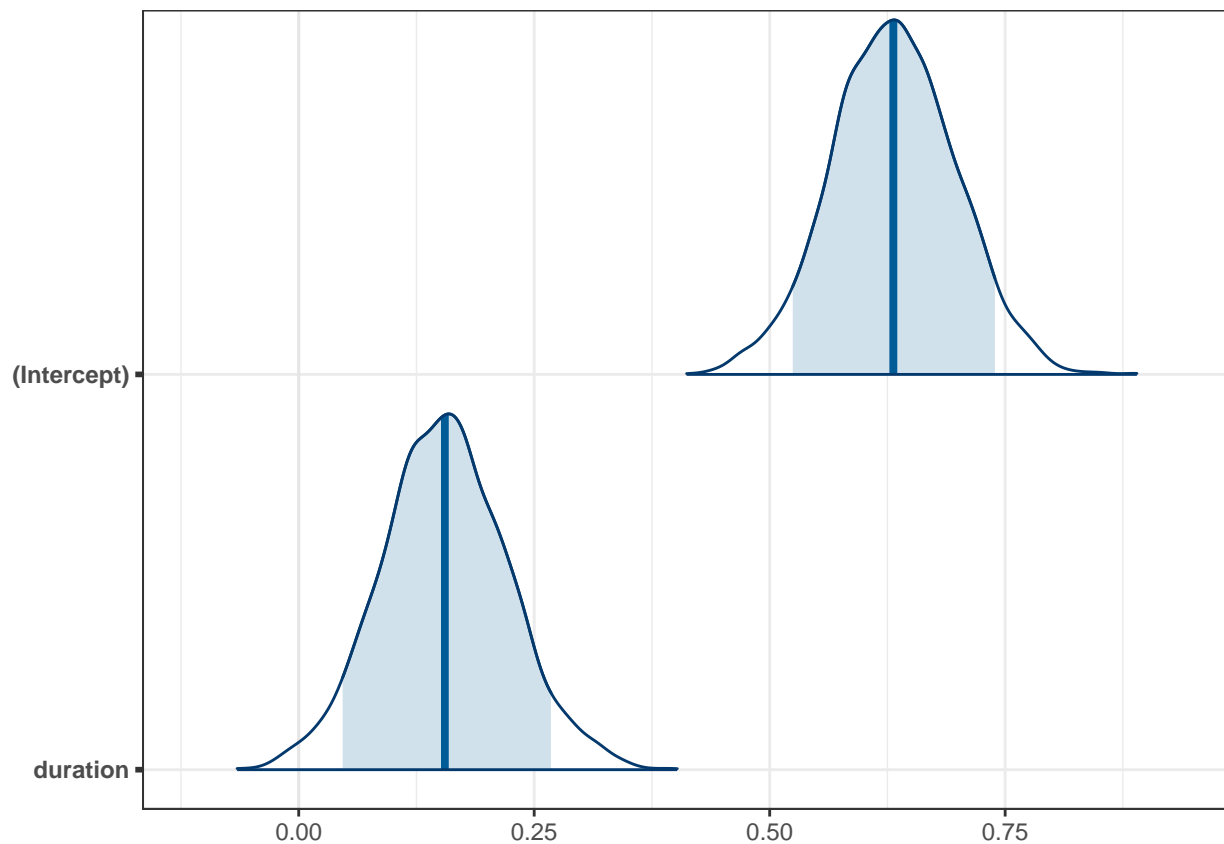
```

```
##
##           Estimate   SE
## elpd_loo   -646.7  9.6
## p_loo       2.0  0.1
## looic      1293.4 19.3
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
rstanarm::loo_compare(loo0, loo2)
```

```
##           elpd_diff se_diff
## posterior2  0.0       0.0
## posterior0 -1.8       2.4
```

```
mcmc_areas(as.matrix(posterior2), prob = 0.90, prob_outer = 1)
```



```
round(posterior_interval(posterior2, prob = 0.90), 3)
```

```
##           5%  95%
## (Intercept) 0.524 0.739
## duration    0.047 0.268
```

This model proved to be better, but not by much. Furthermore, the p

```
## Instead of posterior_linpred(..., transform=TRUE) please call posterior_epred(), which provides equi
```

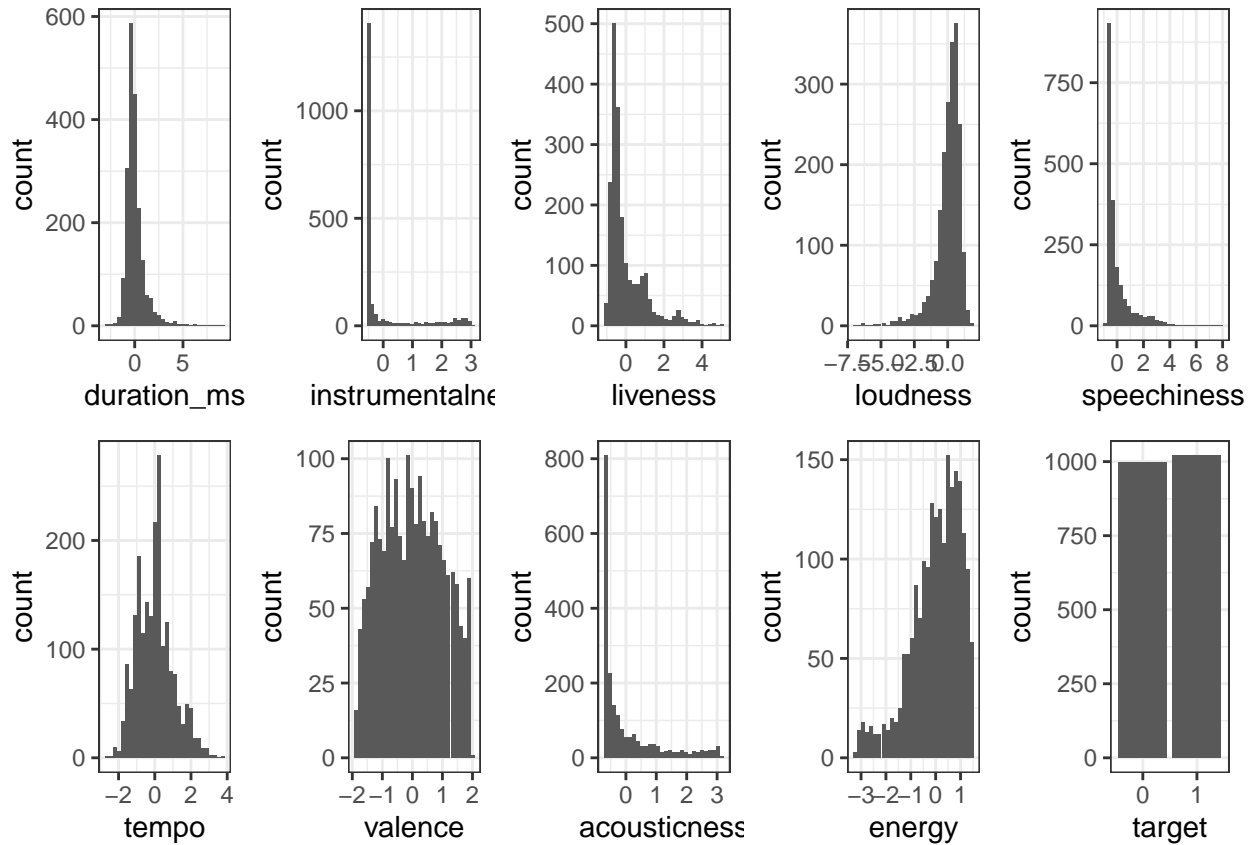
```
## [1] 0.65
```

```
## [1] 0.65
```

```
###New Data
```

```
#Drop un-needed variables
spotify1 <- spotify[-c(1,16,17)]
#View(spotify1)
spotify1$target <- factor(spotify1$target)
spotify1$mode <- factor(spotify1$mode)
spotify1$key <- factor(spotify1$key)
spotify1 <- spotify1 %>%
  mutate(duration_ms = duration_ms / 1000)
# scale the covariates for easier comparison of coefficient posteriors
spotify1$acousticness <- scale(spotify1$acousticness)
spotify1$danceability <- scale(spotify1$danceability)
spotify1$duration_ms <- scale(spotify1$duration_ms)
spotify1$energy <- scale(spotify1$energy)
spotify1$instrumentalness <- scale(spotify1$instrumentalness)
spotify1$liveness <- scale(spotify1$liveness)
spotify1$loudness <- scale(spotify1$loudness)
spotify1$speechiness <- scale(spotify1$speechiness)
spotify1$tempo <- scale(spotify1$tempo)
spotify1$valence <- scale(spotify1$valence)
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

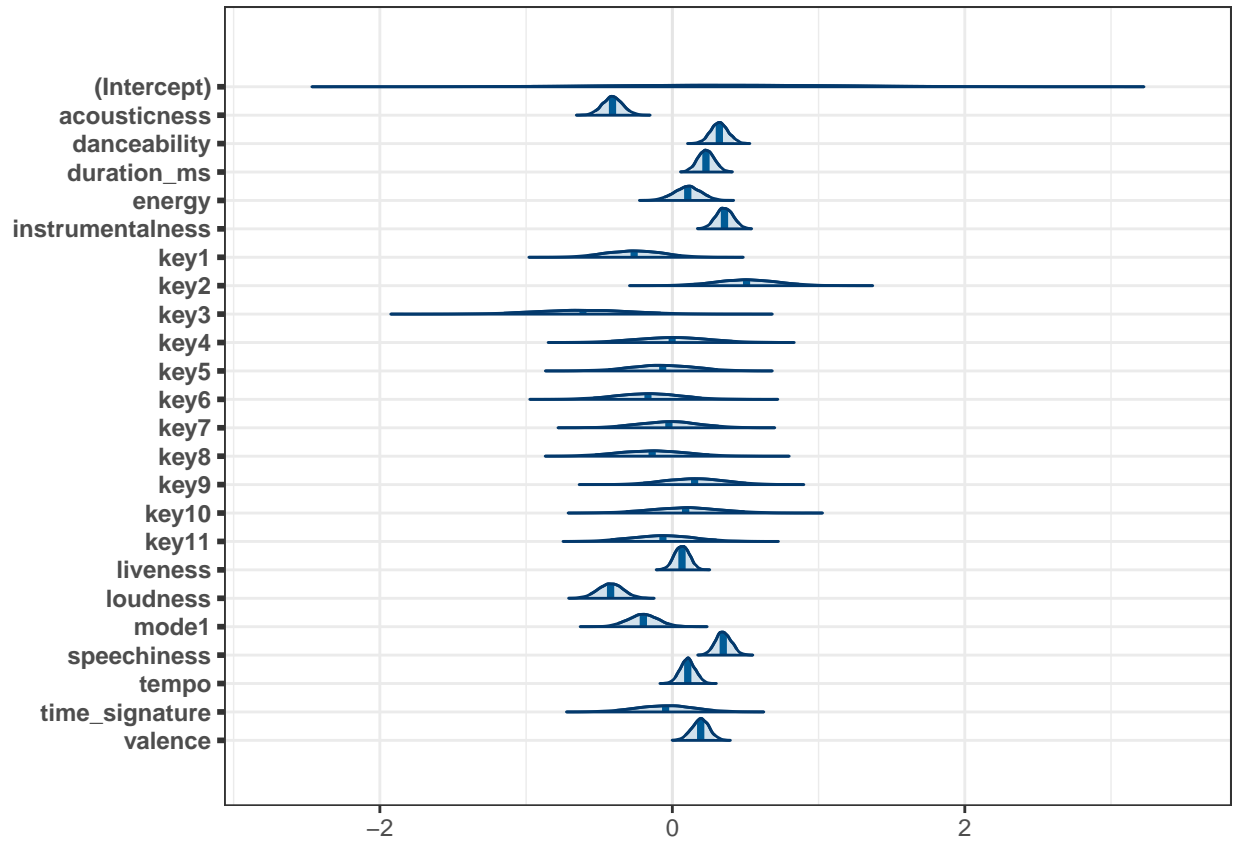


```
##
## Model Info:
## function:      stan_glm
## family:        binomial [logit]
## formula:       target ~ .
## algorithm:     sampling
## sample:        4000 (posterior sample size)
## priors:        see help('prior_summary')
## observations:  2017
## predictors:    24
##
## Estimates:
##              mean    sd  10%  50%  90%
## (Intercept)   0.4    0.8 -0.7   0.4   1.4
## acousticness -0.4    0.1 -0.5  -0.4  -0.3
## danceability  0.3    0.1  0.2   0.3   0.4
## duration_ms   0.2    0.1  0.2   0.2   0.3
## energy        0.1    0.1  0.0   0.1   0.2
## instrumentalness 0.4    0.1  0.3   0.4   0.4
## key1         -0.3    0.2 -0.5  -0.3   0.0
## key2          0.5    0.2  0.2   0.5   0.8
## key3         -0.6    0.3 -1.0  -0.6  -0.2
## key4          0.0    0.3 -0.3   0.0   0.3
## key5         -0.1    0.2 -0.3  -0.1   0.2
## key6         -0.2    0.2 -0.5  -0.2   0.1
## key7          0.0    0.2 -0.3   0.0   0.2
```

```

## key8          -0.1    0.2 -0.4  -0.1    0.2
## key9          0.2     0.2 -0.1   0.2     0.4
## key10         0.1     0.2 -0.2   0.1     0.4
## key11        -0.1     0.2 -0.3  -0.1     0.2
## liveness      0.1     0.1  0.0   0.1     0.1
## loudness     -0.4     0.1 -0.5  -0.4    -0.3
## model1       -0.2     0.1 -0.3  -0.2    -0.1
## speechiness   0.3     0.1  0.3   0.3     0.4
## tempo         0.1     0.1  0.0   0.1     0.2
## time_signature 0.0     0.2 -0.3   0.0     0.2
## valence       0.2     0.1  0.1   0.2     0.3
##
## Fit Diagnostics:
##           mean    sd   10%   50%   90%
## mean_PPD 0.5     0.0  0.5   0.5   0.5
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##           mcse Rhat n_eff
## (Intercept) 0.0   1.0  6427
## acousticness 0.0   1.0  5178
## danceability 0.0   1.0  4508
## duration_ms  0.0   1.0  7388
## energy        0.0   1.0  3166
## instrumentalness 0.0 1.0  6279
## key1          0.0   1.0  1729
## key2          0.0   1.0  2056
## key3          0.0   1.0  3784
## key4          0.0   1.0  2661
## key5          0.0   1.0  2200
## key6          0.0   1.0  2169
## key7          0.0   1.0  2015
## key8          0.0   1.0  2316
## key9          0.0   1.0  1983
## key10         0.0   1.0  2432
## key11         0.0   1.0  1986
## liveness      0.0   1.0  6678
## loudness      0.0   1.0  3862
## model1        0.0   1.0  6291
## speechiness   0.0   1.0  6375
## tempo         0.0   1.0  8025
## time_signature 0.0   1.0  7416
## valence       0.0   1.0  4655
## mean_PPD      0.0   1.0  5700
## log-posterior 0.1   1.0  1719
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample

```

##	(Intercept)	acousticness	danceability	duration_ms
##	0.371	-0.409	0.321	0.230
##	energy	instrumentalness	key1	key2
##	0.105	0.356	-0.261	0.507
##	key3	key4	key5	key6
##	-0.611	-0.001	-0.066	-0.167
##	key7	key8	key9	key10
##	-0.024	-0.138	0.152	0.091
##	key11	liveness	loudness	mode1
##	-0.064	0.066	-0.422	-0.198
##	speechiness	tempo	time_signature	valence
##	0.348	0.105	-0.046	0.194

##		5%	95%
##	(Intercept)	-0.964	1.736
##	acousticness	-0.522	-0.295
##	danceability	0.224	0.423
##	duration_ms	0.137	0.324
##	energy	-0.050	0.250
##	instrumentalness	0.262	0.454
##	key1	-0.576	0.054
##	key2	0.147	0.865
##	key3	-1.151	-0.083
##	key4	-0.418	0.410
##	key5	-0.431	0.303

```
## key6          -0.536  0.196
## key7          -0.374  0.314
## key8          -0.523  0.248
## key9          -0.195  0.503
## key10         -0.311  0.489
## key11         -0.417  0.299
## liveness      -0.017  0.149
## loudness      -0.561 -0.282
## model         -0.369 -0.028
## speechiness    0.263  0.438
## tempo         0.022  0.191
## time_signature -0.380  0.280
## valence        0.093  0.289
```

```
(loo3 <- loo(posterior3, save_psis = TRUE))
```

```
##
## Computed from 4000 by 2017 log-likelihood matrix
##
##           Estimate   SE
## elpd_loo -1268.1 17.0
## p_loo      25.0  0.6
## looic      2536.3 34.1
## -----
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
##
## Computed from 4000 by 2017 log-likelihood matrix
##
##           Estimate   SE
## elpd_loo -1398.9 0.5
## p_loo      1.0  0.0
## looic      2797.9 1.0
## -----
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
##           elpd_diff se_diff
## posterior3      0.0      0.0
## posterior4 -130.8     17.1
```

```
## Instead of posterior_linpred(..., transform=TRUE) please call posterior_epred(), which provides equi
```

```
## [1] 0.676
```

```
## [1] 0.667
```