Identifying Music as Classical or Popular

Alexander Messam

December 2019

Abstract

There are a variety of different factors that contribute to how people identify music with specific genres. Researchers were specifically interested in what factors make a song sound more classical or more popular. 70 undergraduate students from University of Pittsburgh were made to listen to a variety of songs and data on 27 characteristics of the students and the songs they listened to were recorded, including ratings for how classical and popular they thought the music sounded. Multilevel models were then trained to investigate which factors contributed to how people identified songs as classical or popular. Almost the same subset of the features were significantly correlated with classical and popular ratings. However, the signs for many of the variables between the two models were flipped, indicating that in general songs that are identified as classical on average won't be as popular, an vice versa.

Introduction

Throughout history, music has gone through many different phases and new genres of music emerge. Each of these different genres of music sound different in one way or another, but not all music is interpreted in the same way by different people. Researchers were interested in investigating what factors contribute to how people identify music of different genres, particularly the genres classical and popular. The data we used to investigate these song genre identifications were from a designed experiment conducted at University of Pittsburgh in 2012 intended to measure the influence of instrument, harmonic motion, and voice leading on listeners' identification of music as classical or popular. Said data were collected by composer/musicologist Ivan Jimenez and student Vincent Rossi.(Data can be found). The data contains info on 27 variables that were recorded for 70 listeners (undergraduate students at University of Pittsburgh). Using this data, we will primarily focus on determining which music stimuli variables have a significant effect on how classical or popular a song sounds to someone. However, while conducting this study researchers made specific hypotheses about what factors would lead to music sounding classical or popular.

So in addition to our main topic of interest, we will pay special attention to the following sub-questions:

- Does Instrument exert the strongest influence among the three design factors (Instrument, Harmonic Motion, Voice Leading)?
- Among the levels of Harmonic Motion, does *I-V-VI* have the strongest association with classical ratings? Does it seem to matter whether the respondent is familiar with one or the other (or both) of the Pachelbel rants/comedy bits?
- Among the levels of Voice Leading, does contrary motion have a strong association with classical ratings?
- Are there differences in the way that musicians and non-musicians identify classical music?
- Are there differences in the things that drive classical vs./ popular ratings?

Methods

The 70 University of Pittsburgh students were presented with 36 music stimuli that were chosen by completely crossing the following design factors:
Instrument: String Quartet, Piano, Electric Guitar
Harmonic Motion: I-V-VI, I-VI-V, I-V-IV, IV-I-V
Voice Leading: Contrary Motion, Parallel 3rds, Parallel 5ths

After being exposed to those stimuli, they were then asked to rate the music that they listened to on two different scales:

1. How classical does the music sound (1 to 10, 1 = not at all, 10 = very classical sounding)
2. How popular does the music sound (1 to 10, 1 = not at all, 10 = very popular sounding)

The students were told that the music could be rated as popular, classical, both, or neither. Data were collected on the students' ratings of the songs, the design factors exemplified in the song, and various other personal characteristics of the students themselves. Our analysis/data was processed in R. The description of all of the variables is as follows:

Classical	How classical does the stimulus sound?
Popular	How popular does the stimulus sound?
Subject	Unique subject ID
Harmony	Harmonic Motion (4 levels)
Instrument	Instrument (3 levels)
Voice	Voice Leading (3 levels)
Selfdeclare	Are you a musician? (1-6, 1=not at all)
OMSI	Score on a test of musical knowledge
X16.minus.17	Auxiliary measure of listener's ability to distinguish classical vs popular music
ConsInstr	How much did you concentrate on the instrument while listening (0-5, 0=not at all)
ConsNotes	How much did you concentrate on the notes while listening? (0-5, 0=not at all)
Instr.minus.Notes	Difference between prev. two variables
PachListen	How familiar are you with Pachelbel's Canon in D (0-5, 0=not at all)
ClsListen	How much do you listen to classical music? (0-5, 0=not at all)
KnowRob	Have you heard Rob Paravonian's Pachelbel Rant (0-5, 0=not at all)
KnowAxis	Have you heard Axis of Evil's Comedy bit on the 4 Pachelbel chords in popular music? (0-5, 0=not at all)
X1990s2000s	How much do you listen to pop and rock from the 90's and 2000's? (0-5, 0=not at all)
X1990s2000s.minus.1960s1970s	Difference between prev variable and a similar variable referring to 60's and 70's pop and rock.
CollegeMusic	Have you taken music classes in college (0=no, 1=yes)
NoClass	How many music classes have you taken?
APTheory	Did you take AP Music Theory class in High School (0=no, 1=yes)
Composing	Have you done any music composing (0-5, 0=not at all)
PianoPlay	Do you play piano (0-5, 0=not at all)
GuitarPlay	Do you play guitar (0-5, 0=not at all)
X1stInstr	How proficient are you at your first musical instrument (0-5, 0=not at all)
X2ndInstr	Same, for second musical instrument
first12	In the experiment, which instrument was presented to the subject in the first 12 stimuli? (IGNORE FOR THIS ASSIGNMENT.)

Figure 1

Once all of the data was cleaned, we trained multilevel-linear models to predict scores (from 1 to 10) of how classical and popular different music sounded using our data and included both fixed and random effects into the models. The models were then fit again with slight alterations in order to observe the effects of the design variables on the ratings without the random effect components. All of these models were used to answer our research questions.

Results

Before any analysis was completed we needed to extensively clean our data. Most of our data looked appropriate with reasonable values, but we had to manipulate these variables in the following ways to prepare for the modeling phase of our analysis:

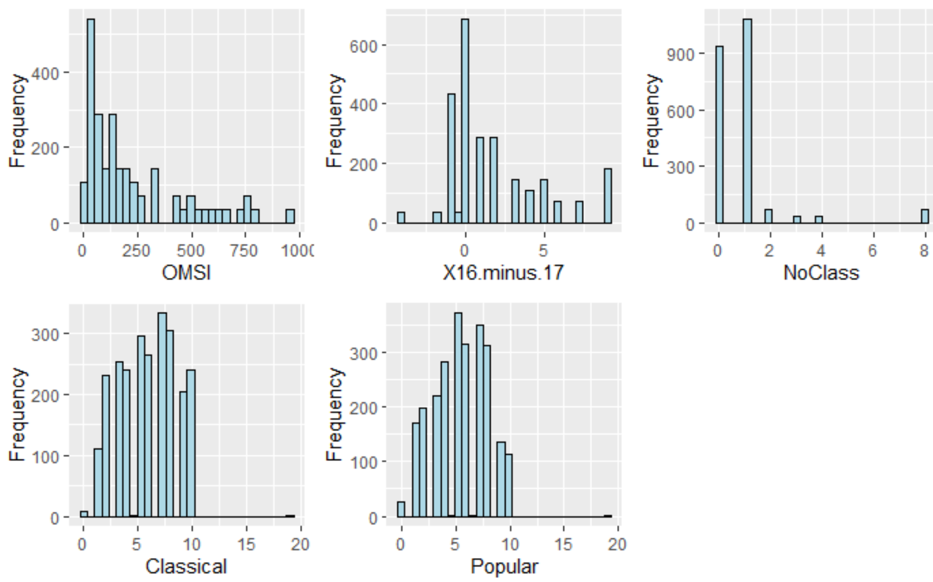


Figure 2

Classical) This variable was supposed to be a rating of how classical a song sounded to the listener, on a scale of 1-10. However, ratings of 0, 9.5, 4.6, 3.5, and 4.2 were present in the data along with some null values. Considering that this only made up 1.54% of the data, these data points were removed. There was also a rating of 19 present in the data, but we'll assume that this was supposed to be a 10 (considering how close 0 and 9 are on the keyboard) and re-code it.

Popular) This variable had the same problem as Classical. 2.22% of the data is missing or has values of 0, 3.5, 4.6, 6.8, and 4.2. Again, these data points were removed from our dataset since it is such a small proportion of the data. There was also a rating of 19 present in the data, and again re-code it to 10 for the same reason as above.

OMSI) This Variable is very right skewed (Figure 2), which would have introduced leverage points into our model and messed up our analysis. Thus, this variable was transformed by taking its square root.

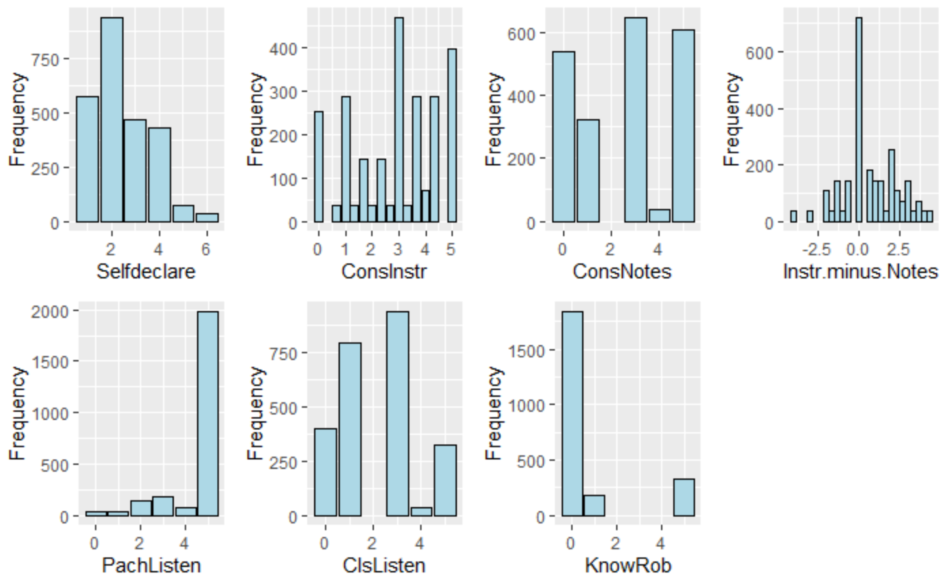


Figure 3

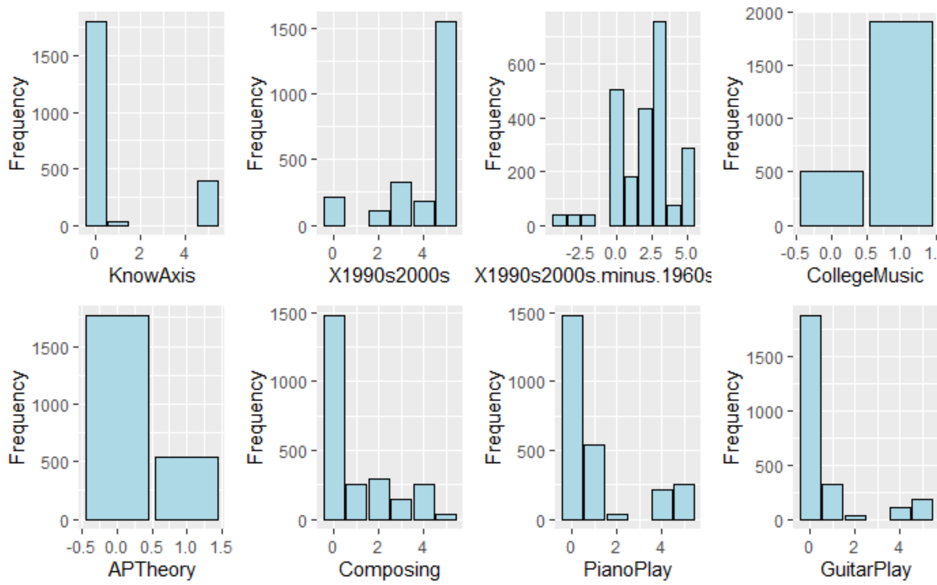


Figure 4

ConsInstr) This variable is supposed to be a categorical variable whose values range from 0-5. However, there are about 40% of the data that are in between these values (e.g., 3.33 or 1.67, see Figure 3). This was remedied by rounding all of those values to the closest integer so only integers [0-5] present.

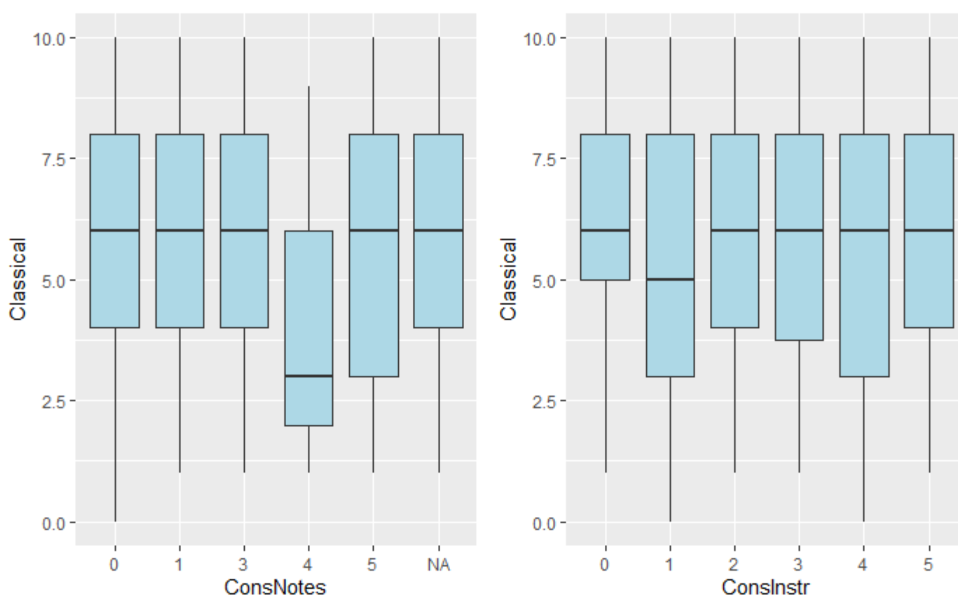


Figure 5

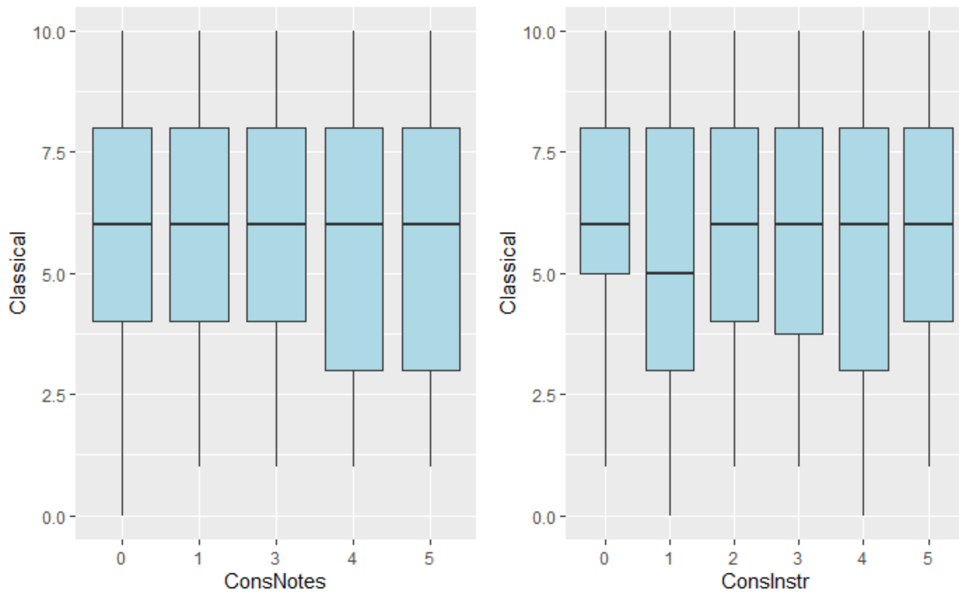


Figure 6

ConsNotes) Upon inspection it was peculiar that there were so few students rating how much they concentrated on the notes of the song a 4. The other ratings with positive amounts of votes had hundreds of records, but there were only 36 votes for concentration ratings 4 (Figure 3), along with 360 null values. When the average person is asked whether they concentrated on instruments or notes when listening to a song, they are likely to interpret both of those questions very similarly. Thus, we'd expect those variables to be distributed fairly similarly. If all the null values present in ConsNotes are put into a rating of 4, the distributions look very similar (Figure 5 pre-clean, Figure 6 post-clean). So we put all of the null values within category 4.

Instr.minus.Note) This variable is supposed to be a categorical variable whose values are whole numbers. However, this variable had inappropriate decimal values (Figure 3) since it was coded as (ConsInstr - ConsNotes) and ConsInstr had inappropriate values. Thus, we just re-coded this variable the same way but after cleaning ConsInstr and ConsNotes.

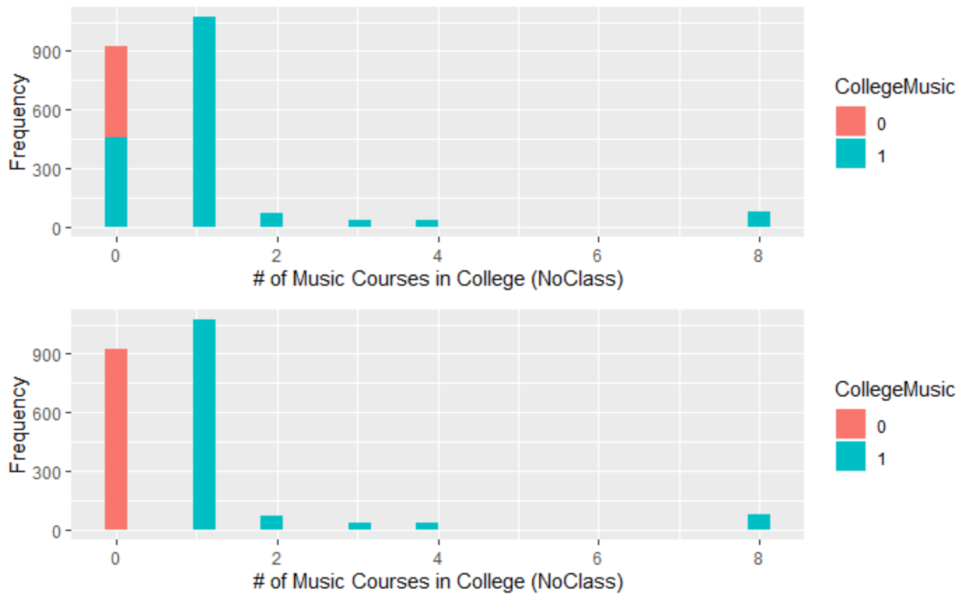


Figure 7

CollegeMusic) Anybody who has this variable coded as 1 should have a positive number of music courses taken in college (NoClass positive). However, this wasn't the case and there were some students whose value of CollegeMusic were coded incorrectly (Figure 7). Thus, we set CollegeMusic to 0 for each student whose number of music classes taken in college (NoClass) was 0.

X1stInstr and X2ndInstr) Will not be considered because there are way too many null values to extrapolate useful information from them (See appendix page 4). We cannot necessarily interpret NA as doesn't play an instrument. Not playing an instrument is likely captured by other variables (such as selfDeclare).

Missing Values) The 3 design factors (Instrument, Harmony, and Voice) have no null values. However, Popular and Classical have 27 null values (25 of them in common). Since this was such a small proportion of our data (and these variables are important) we will remove these records. Many of the other variables in our dataset have missing values as well (See appendix pages 3/4). Median imputation was used for missing continuous variables with missing values (NA's replaced with median) and Mode imputation used to for categorical data (NA's replaced with most common category).

After dirty data was dealt with, we trained our multilevel models to predict classical and popular ratings from the other variables in our dataset. However, it is important to note that because Instrument, Voice, and Harmony are design variables in the experiment, the main effects for the three experimental factors were included in all model considerations.

By hand, forward stepwise regression was used to select which variables in our dataset we included in our model as fixed effects. Each variable was considered, and the ones whose individual contribution reduced the BIC of the model were included as fixed effects. For classical ratings, the variables that were found to have significant fixed effects were X16.minus.17, NoClass, and PianoPlay (See appendix pages 6/7). Random effects were also selected in a similar manner. All linear combinations of design variables were considered for random effects in our model to control for personal biases in song identification among individuals (e.g., some individuals may be more prone to identify all music as classical than others based on which instrument they hear). And since 36 ratings from each participant were present, a random intercept for each participant was also considered. Classical ratings were found to be appropriately adjusted by including random effects for each individual subject, Instrument, and Harmony as a result of this combination of variables reducing BIC of our model the most (See appendix pages 9/10).

The same process was applied to finding variables used to predict popular ratings. X16.minus.17, NoClass, and Selfdeclare were found to have significant fixed effects (See appendix pages 23/24), while random effects for individual subjects, Instrument, and Harmony were found to be significant (See appendix pages 25-27). The final models can be presented as follows:

$$\begin{aligned} Classical = {} & \beta_1 Instrument + \beta_2 Voice + \beta_3 Harmony \\ & + \beta_4 X16.Minus.17 + \beta_5 NoClass + \beta_6 PianoPlay \\ & + (1 + Harmony + Instrument | Subject) \end{aligned} \tag{1}$$

Below are the interpretations for the fixed effects of our model.

If the song has the Instrument guitar in it, uses Contrary Voice, and has a I-IV-V Harmony, the person listening to the song hasn't taken any college music courses, doesn't know how to play the piano, and isn't able to distinguish classical vs popular music, we'd expect that the song would receive a Classical score of 4.07 on average.

On average, we'd expect that the classical rating of a song with a piano in it would be 1.49 higher than a song that has a guitar in it, all else held constant.

On average, we'd expect that the classical rating of a song with a string instrument in it would be 3.28 higher than a song that has a guitar in it, all else held constant.

On average, we'd expect that the classical rating of a song with Parallel thirds in it would be -.37 lower than a song that has contrary motion in it, all else held constant.

On average, we'd expect that the classical rating of a song with Parallel fifths in it would be -.33 lower than a song that has contrary motion in it, all else held constant.

The coefficient for Harmony I-V-IV wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that the classical rating of a song with the I-V-IV harmony to be -.06 lower than a song that has a I-VI-V harmony, all else held constant.

On average, we'd expect that the classical rating of a song with the I-V-VI harmony to be .76 higher than a song that has a I-VI-V harmony, all else held constant.

The coefficient for Harmony IV-I-V wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that the classical rating of a song with the IV-I-V harmony to be -.006 lower than a song that has a I-VI-V harmony, all else held constant.

On average, we'd expect that a person with a piano playing proficiency of 1/5 would have classical ratings .72 higher than a person with a piano playing proficiency of 0/5, all else held constant.

On average, we'd expect that a person with a piano playing proficiency of 2/5 would have classical ratings 2.36 higher than a person with a piano playing proficiency of 0/5, all else held constant.

The coefficient for PianoPlay4 wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that a person with a piano playing proficiency of 4/5 would have classical ratings .69 higher than a person with a piano playing proficiency of 0/5, all else held constant.

The coefficient for PianoPlay5 wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that a person with a piano playing proficiency of 5/5 would have classical ratings .7 higher than a person with a piano playing proficiency of 0/5, all else held constant.

The coefficient for X16.minus.17 wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that if your auxiliary score of being able to distinguish classical vs popular music was one unit higher than someone else's, your classical ratings of songs will be -.08 lower, all else held constant.

The coefficient for NoClass wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that if you've taken one more music class in college than someone, your classical ratings of songs will be -.03 lower, all else held constant.

$$\begin{aligned} Popular = {} & \beta_1 Instrument + \beta_2 Voice + \beta_3 Harmony \\ & + \beta_4 X16.Minus.17 + \beta_5 NoClass + \beta_6 Selfdeclare \\ & + (1 + Harmony + Instrument|Subject) \end{aligned} \tag{2}$$

Below are the interpretations for the fixed effects of our model. (See appendix pages 28/29 for model diagnostics).

If the song has the Instrument guitar in it, uses Contrary Voice, and has a I-IV-V Harmony, the person listening to the song hasn't taken any college music courses, doesn't identify as a musician, and isn't able to distinguish popular vs popular music, we'd expect that the song would receive a popular score of 5.55 on average.

On average, we'd expect that the popular rating of a song with a piano in it would be -1.0 lower than a song that has a guitar in it, all else held constant.

On average, we'd expect that the popular rating of a song with a string instrument in it would be -2.72 lower than a song that has a guitar in it, all else held constant.

On average, we'd expect that the popular rating of a song with Parallel thirds in it would be .17 higher than a song that has contrary motion in it, all else held constant.

On average, we'd expect that the popular rating of a song with Parallel fifths in it would be .18 higher than a song that has contrary motion in it, all else held constant.

The coefficient for Harmony I-V-IV wasn't deemed statistically significantly different from. However, if the sample size increased, decreasing the variance of the variable and making it statistically significant, we'd expect that the popular rating of a song with the I-V-IV harmony to be -.04 lower than a song that has a I-VI-V harmony, all else held constant.

The coefficient for Harmony I-V-VI wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that the popular rating of a song with the I-V-VI harmony to be -.27 lower than a song that has a I-VI-V harmony, all else held constant.

The coefficient for Harmony IV-I-V wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that the popular rating of a song with the IV-I-V harmony to be -.21 lower than a song that has a I-VI-V harmony, all else held constant.

On average, we'd expect that a person who rates themselves as a 2/6 musician would give popular ratings 1.23 higher than a person who rates themselves as a 1/6 musician, all else held constant.

On average, we'd expect that a person who rates themselves as a 3/6 musician would give popular ratings 1.04 higher than a person who rates themselves as a 1/6 musician, all else held constant.

The coefficient for Selfdeclare4 wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that a person who rates themselves as a 4/6 musician would give popular ratings .86 higher than a person who rates themselves as a 1/6 musician, all else held constant.

The coefficient for Selfdeclare5 wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that a person who rates themselves as a 5/6 musician would give popular ratings 1.38 higher than a person who rates themselves as a 1/6 musician, all else held constant.

The coefficient for Selfdeclare6 wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that a person who rates themselves as a 6/6 musician would give popular ratings -.60 lower than a person who rates themselves as a 1/6 musician, all else held constant.

The coefficient for X16.minus.17 wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that if your auxiliary score of being able to distinguish classical vs popular music was one unit higher than someone else's, your classical ratings of songs will be .1 higher, all else held constant.

The coefficient for NoClass wasn't deemed statistically significantly different from. However, if the sample size slightly increased, decreasing the variance of the variable and making it statistically significant, we'd expect that if you've taken one more music class in college than someone, your classical ratings of songs will be .1 higher, all else held constant.

The random effects can be interpreted as the additional change on would expect on classical or popular ratings conditioned on the subject, all else held constant. Important to note that the variance of the residuals is larger than some of the variances of our random effects, which is not ideal. This means that our model isn't capturing all of the individual biases that people may have that make them more likely to think a song is classical/popular.

Which design factor has the greatest influence over how Classical/Popular a song sounds?

Upon inspection of our model summary output for both classical and popular (See appendix pages 11 or 20 and 28), we can see that the coefficients for the different levels of Instrument are statistically significantly different from 0 with t-values ≥ 3. However, what's of more interest here is the fact that the coefficients for Instrument are very large relative to the other design factors' coefficients. This is strong evidence that on average, Instrument has a larger influence on how classical a song sounds than Voice Leading and Harmonic Motion.

Which of the four Harmonic Motions has the most influence over how Classical a songs sounds?

In order to observe the effects of different Harmonic Motions on Classical ratings without the effects of the variable being included in the noise, we refit the model slightly differently (See appendix page 13). This allowed us to be able to take out this variables contribution to the random effects and only be able to concentrate on the fixed effects alone to see which ones were most prevalent. Upon inspection of the model, we can see that the I-V-VI Harmony has the largest coefficient and the most statistically significantly different from 0 coefficient (largest t-value).

This is strong evidence that on average, the I-V-VI Harmony has a stronger association with classical ratings than the other 2 harmonies.

When we attempted to see whether the rants/comedy bits (KnowRob and KnowAxis) were significant predictors, neither of them improved the prediction accuracy of the model evident by the increase in AIC after adding them (See appendix page 8). So it doesn't seem like being familiar with either the rant or the comedy bit have any significant effect when it comes to how you identify classical music.

Does contrary motion make a song sound more classical than parallel 3rds and parallel 5ths?

In order to observe the effects of different Voice Leadings on Classical ratings without the effects of the variable being included in the noise, we refit the model slightly differently (See appendix page 14). This allowed us to be able to take out this variables contribution to the random effects and only be able to concentrate on the fixed effects alone to see which ones were most prevalent. Upon inspection of the model, we can see that the Contrary Voice has the largest coefficient and the largest t-value. This is strong evidence that on average, the Contrary Voice has a stronger association with classical ratings than Contraty 3rds and 5ths.

Do musicians/non-musicians identify classical music differently?

Whether someone was classified as a musician or not was self defined. The Selfdeclare variable was dichotomized so that students with Selfdeclare ≤ 2 were said to not be musicians, and all others were said to be musicians. This split was chosen because it is the split that most closely splits up the subjects into a 50/50 ratio. The model fit was of the following form:

$$\begin{aligned} Classical = (1 + Harmony + Instrument|Subject) + (\beta_1 Instrument + \beta_2 Voice \\ + \beta_3 Harmony + \beta_4 X16.Minus.17 + \beta_5 NoClass + \beta_6 Selfdeclare) \\ * Musician \end{aligned} \tag{3}$$

This was done in order to see whether the coefficients that were statistically significant were different among musicians and non-musicians. From the model summary (See appendix pages 15/16), we can see that the coefficients that were deemed statistically significantly different from 0 were different between the musicians and the non-musicians. While normally Instrument, Voice, Harmony, and PianoPlay have significance when it comes to predicting classical ratings, only Harmony had a statistically significant coefficient. So the factors that effect how musicians/non-musicians identify classical music are different. Other dichotomizations of Selfdeclare to define when someone is classified as a Musician or not were attempted and different dichotomizaitons have different results as to which variables were deemed significant predictors (See appendix pages 16-19).

Are there different factors that drive classical and popular ratings?

From our model summaries from predicting classical and popular ratings (See appendix pages 10/11 and 28), we can see that the subset of variables from our dataset that were deemed significant predictors for the two different ratings were slightly different. The random effects that were needed in both our models were same, a random intercept for each subject, and random effects for both Instrument and Harmony. However, the fixed effects that were found to be significant were different between the two ratings. The factors that drive whether a song sounds classical were Instrument, Voice, Harmony, whether you play the piano (PianoPlay), your ability to distinguish classical vs popular music (X16.minus.17), and the number of music classes you've taken in college (NoClass). Whereas the factors that drive whether a song sounds popular were Instrument, Voice, Harmony, X16.minus.17, NoClass, and how much you identify yourself as a musician (Selfdeclare). The only difference between the factors that are important for the different ratings were that PianoPlay is significant for predicting classical ratings and Selfdeclare is significant for predicting popular ratings. Important to note that even though a lot of the same covariates were chosen, the signs of those covariates are flipped for the two models. So while there are a lot of common factors driving these ratings, they drive the ratings in different ways.

Discussion

After investigating our models for predicting classical and popular ratings, we learned that nearly the same set of features were significantly correlated with the two different ratings. Instrument, Voice, Harmony, NoClass, and X16.minus.17 were significant in both models. The only differences between the two models were that PianoPlay was significantly correlated with classical ratings and Selfdelcare was significantly correlated with popular ratings. However, it is important to note that even though a lot of the same features are important in both models, the signs of the coefficients are flipped. This means that in general, The factors that contribute to a song sounding more classical are also affiliated with it sounding less popular. And vice versa. It makes sense that music sounding classical or popular are negatively correlated because the subjects of the study were undergraduate students, and in modern times classical music isn't very popular among our youth.

The Harmonic Motion I-V-VI was found to be more significant than the other harmonies. This is likely because it is the beginning of a famous classical piece of music Pachelbel's *Canon in D*. People may have heard this piece but not know what it was called so cannot identify it if asked only when the song is played. Contrary motion was found to have a large effect on how classical a song sounds (more so than other voice leadings), and the presence of pianos and string instruments did as well.

One things that remains unanswered to us is which features would have been deemed significantly correlated with popular and classical ratings if the 3 design variables weren't forcefully included in the model. Yes they were deemed significant predictors. However, if some of them were excluded, we could have discovered a different subset of predictor that predicted ratings even better than our current models.

A limitation with the approach that we used to conduct this study was the fact that the subjects that were surveyed were all undergraduate students, who are usually in the age range 18-22. It is expected that young people wouldn't identify classical music as popular, and we may get dramatically different results as the age of the subjects we used to collect our data increases.

However, it is good that we now know what factors are attributed with what makes a song sound "popular". In the future, this information can be used by artists to produce music that fits there criterion and will be more likely to be enjoyed by the average listener.

References

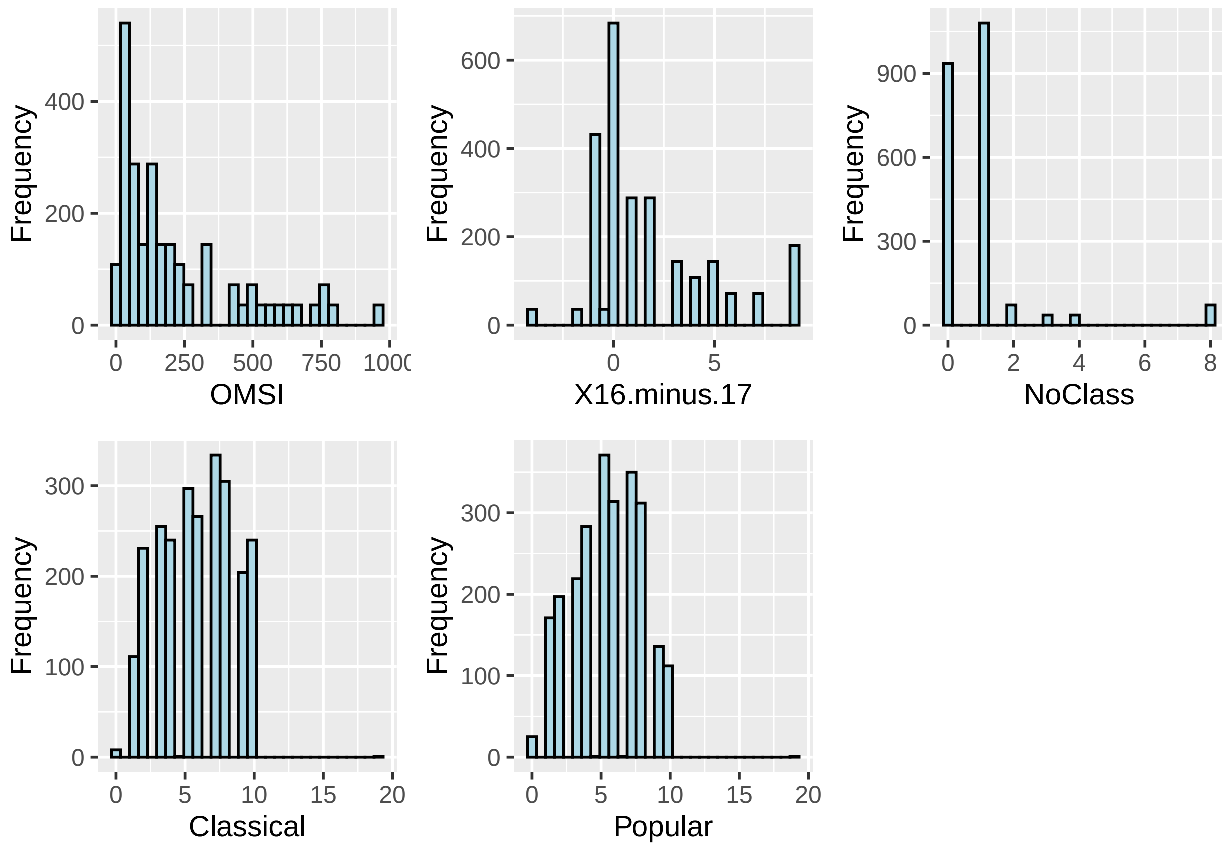
Axis of Awesome. (2011, July 20 post date). *The Axis of Awesome*, http://www.youtube.com/watch?v=o)lDewpCfZQ

Jimenex, I. Rossi, V. (2012). Ratings.csv. University of Pittsburgh. https://canvas.cmu.edu/courses/11853/files/folder/hw10

Jimenez, I. Rossi, V. (2013). *The Influence of Timbra, Harmony, and Voice Leading on Listener's Distinction between Popular and Classical Music.* University of Pittsburgh. https://canvas.cmu.edu/courses/11853/files/folder/hw10

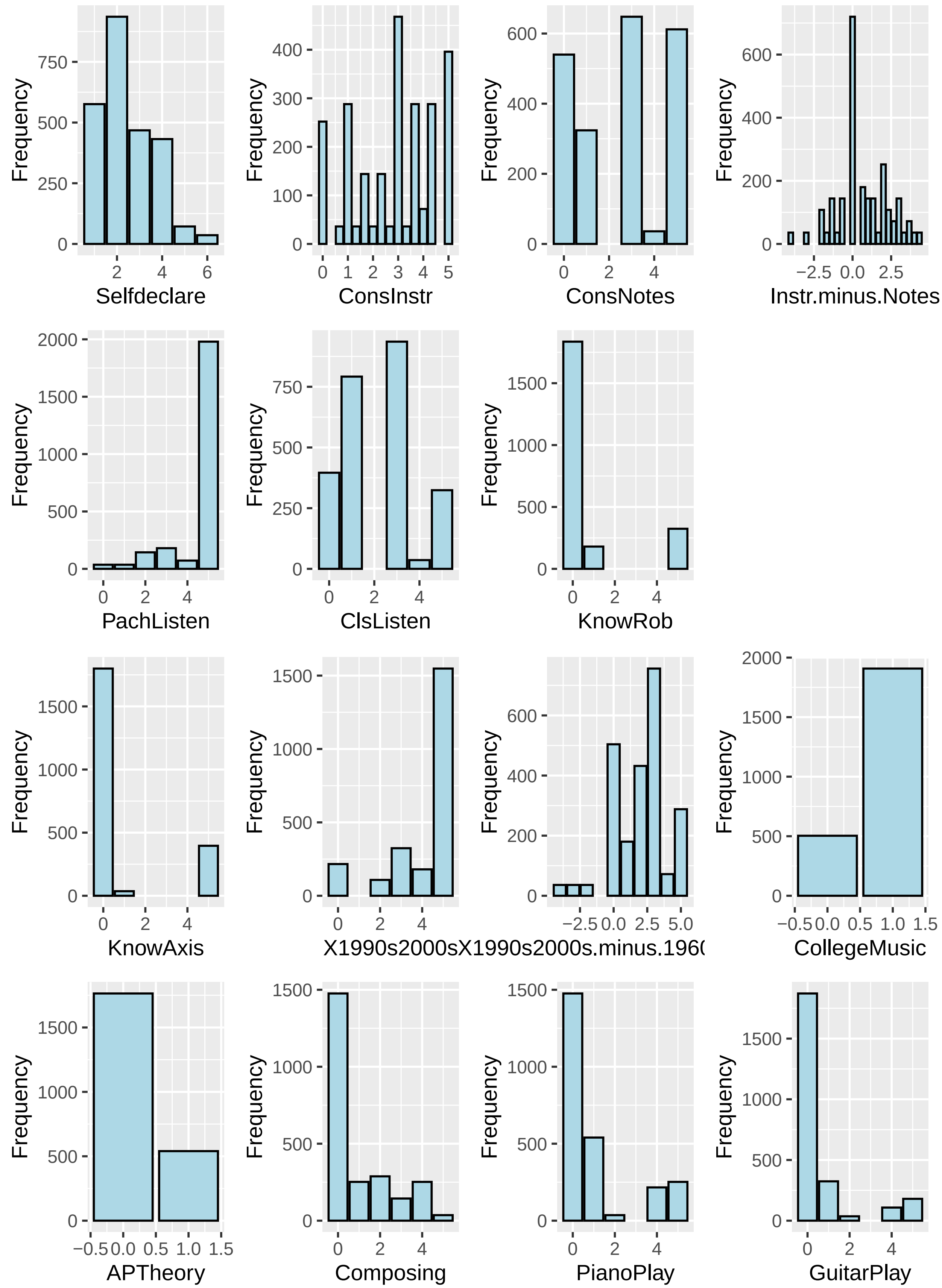
Final Project Technial Appendix

Plots of Continuous Variables




```
## Unique values of 'Classical': 3 1 2 8 10 6 5 4 9 7 NA 0 19 9.5 4.6 3.5 4.2
## Unique values of 'Popular': 9 7 8 3 1 4 5 6 2 10 0 NA 19 3.5 4.6 6.8 4.2
```

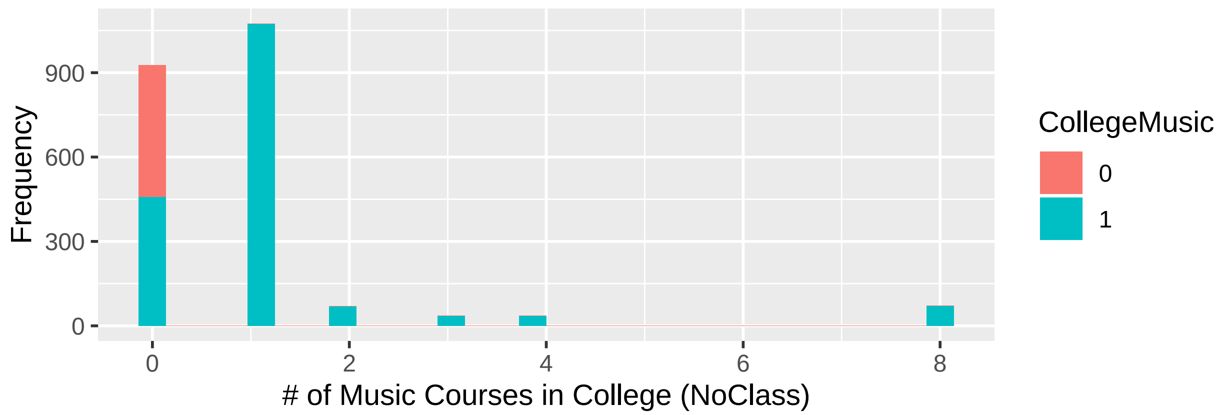
Plots of Categorical Variables

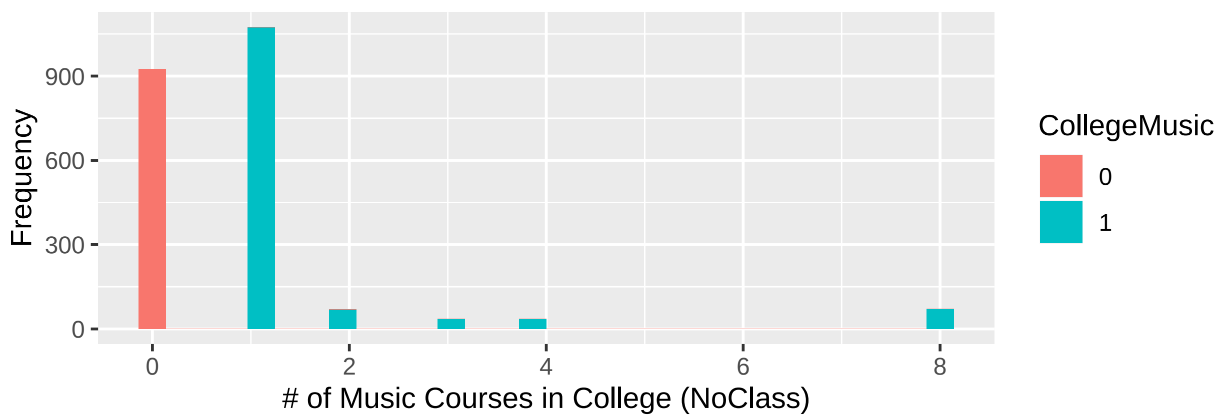


```
## Unique values of 'ConsInstr': 4.33 2.33 1 3.67 3 5 0 1.67 0.67 2.67 4 3.33 2 1.33
```

Data Cleaning

```
## % of data with inappropriate values for 'Popular':  1.27 %
## % of data with inappropriate values for 'Classical':  1.27 %
## % of data with inappropriate values for 'InstrError':  40 %
## ConsNotes Summary:  528 323 637 36 602 360
## 18.42317 % of data where record indicated they took a music course in college (CollegeMusic) and
##     had NA for the number of music classes they've taken in college (NoClass)
```






```
## After deleting records, we are left with 98.65079 % of our original data
```

Imputing missing data

```
## 0 % of Selfdeclare missing
## 0 % of ConsInstr missing
## 2.413516 % of PachListen missing
## 0.9654063 % of ClsListen missing
## 5.309735 % of X1990s2000s missing
```

```
## 3.740949 % of CollegeMusic missing
```

```
## 2.775543 % of Composing missing
```

```
## 0 % of PianoPlay missing
```

```
## 0 % of GuitarPlay missing
```

```
## 0 % of X16.minus.17 missing
```

```
## 0 % of Instr.minus.Notes missing
```

```
## 6.757844 % of X1990s2000s.minus.1960s1970s missing
```

```
## 8.085278 % of APTheory missing
```

```
## 10.9815 % of NoClass missing
```

```
## 0 % of ConsNotes missing
```

```
## 6.87852 % of KnowRob missing
```

```
## 11.58488 % of KnowAxis missing
```

X1stInstr and X2ndInstr will not be considered because there are way too many null values to extrapolate useful information from them. We cannot necessarily interpret NA as doesn't play an instrument. Not playing an instrument is likely captured by other variables (such as selfDeclare).

```
## 59.89541 % of X1stInstr missing
```

```
## 87.32904 % of X2ndInstr missing
```

We ran an anova test to test the significance of the random intercept for classical model.

```
## Data: cleanData
## Models:
## linModel: Classical ~ Instrument + Voice + Harmony
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## linModel        9 11173 11226 -5577.7    11155
## randomIntModel 10 10410 10468 -5195.1    10390 765.17      1  < 2.2e-16
##
## linModel
## randomIntModel ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Initial test for random effects for classical ratings

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer1: Classical ~ Instrument + Voice + Harmony + (1 + Instrument |
## lmer1:     Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410 10468 -5195.1    10390
## lmer1          15 10040 10128 -5005.1    10010 380.05      5  < 2.2e-16
##
## randomIntModel
## lmer1          ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer2: Classical ~ Instrument + Voice + Harmony + (1 + Harmony | Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410 10468 -5195.1    10390
## lmer2          19 10326 10437 -5144.1    10288 102.09      9  < 2.2e-16
##
## randomIntModel
## lmer2          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer3: Classical ~ Instrument + Voice + Harmony + (1 + Voice | Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410 10468 -5195.1    10390
## lmer3          15 10420 10508 -5195.1    10390 0.0328      5          1
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer4: Classical ~ Instrument + Voice + Harmony + (1 + Voice + Instrument |
## lmer4:     Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410 10468 -5195.1    10390
## lmer4          24 10053 10192 -5002.4    10005 385.51     14  < 2.2e-16
##
## randomIntModel
## lmer4          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer5: Classical ~ Instrument + Voice + Harmony + (1 + Instrument +
## lmer5:     Harmony | Subject)
##                Df     AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410.2 10468 -5195.1  10390.2
## lmer5          30  9890.9 10065 -4915.4   9830.9 559.37     20  < 2.2e-16
##
## randomIntModel
## lmer5          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer6: Classical ~ Instrument + Voice + Harmony + (1 + Voice + Harmony |
```

```
## lmer6:     Subject)
##                Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410 10468 -5195.1    10390
## lmer6          30 10342 10516 -5140.8    10282 108.71     20  3.374e-14
##
## randomIntModel
## lmer6          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + (1 | Subject)
## lmer7: Classical ~ Instrument + Voice + Harmony + (1 + Instrument +
## lmer7:     Voice + Harmony | Subject)
##                Df    AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 10 10410.2 10468 -5195.1  10390.2
## lmer7          45  9904.1 10166 -4907.0   9814.1 576.19     35  < 2.2e-16
##
## randomIntModel
## lmer7          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next we will go through and perform forward stepwise regression on the fixed effects by hand to predict classical ratings, one variable at a time to see whether they individually decrease AIC by a significant amount. If so, then add them to the model as a fixed effect.

```
## AIC with no fixed effects: 9890.879
```

```
## AIC with fixed effects: 8801.531
```

After testing all variables, we found that the following reduced AIC by a significant amount on their own: PianoPlay, X16.minus.17, and NoClass. I attempted to use fitlmer.fnc to find which subset of fixed effects to include. However, when done with AIC almost all of the fixed effects are included (which is a bit overkill...). And when done with BIC, none of the fixed effects were deemed significant enough (which also isn't super useful). Thus, we will take our by-hand results without an automated methods input. Below, I'll show that none of the other covariates reduced AIC, and that the 3 that I chose did reduce AIC.

```
# showing correct fixed effects did decrease AIC
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          PianoPlay, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9889.814
```

```
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X16.minus.17, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9890.018
```

```
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          NoClass, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 8800.54
```

```r
# showing incorrect fixed effects didnt decrease AIC
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Selfdeclare, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9897.12
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ConsInstr, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9897.384
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          PachListen, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9895.174
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ClsListen, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9894.365
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X1990s2000s, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9894.97
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          CollegeMusic, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9892.869
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Composing, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9893.227
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          GuitarPlay, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9893.17
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X16.minus.17, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9890.018
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Instr.minus.Notes, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9900.016
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X1990s2000s.minus.1960s1970s, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9897.953
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          APTheory, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9890.373
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ConsNotes, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9898.492
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          KnowRob, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9893.177
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          KnowAxis, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9894.127
```

```r
wrongFixedEffect = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          OMSI, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9892.404
```

Testing Random Effects again for Classical ratings.

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer1: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer1:     NoClass + (1 + Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer1          21 8955.7 9075.5 -4456.9   8913.7 358.11      5  < 2.2e-16
##
## randomIntModel
## lmer1          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer2: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer2:     NoClass + (1 + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer2          25 9218.1 9360.7 -4584.1   9168.1 103.73      9  < 2.2e-16
##
## randomIntModel
## lmer2          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer3: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer3:     NoClass + (1 + Voice | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer3          21 9313.5 9433.2 -4635.7   9271.5 0.3851      5     0.9957
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer4: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer4:     NoClass + (1 + Voice + Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer4          30 8968.4 9139.5 -4454.2   8908.4 363.45     14  < 2.2e-16
##
## randomIntModel
## lmer4          ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer5: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer5:     NoClass + (1 + Instrument + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer5          36 8801.5 9006.8 -4364.8   8729.5 542.32     20  < 2.2e-16
##
## randomIntModel
## lmer5          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer6: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer6:     NoClass + (1 + Voice + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer6          36 9238.8 9444.1 -4583.4   9166.8 105.02     20  1.577e-13
##
## randomIntModel
## lmer6          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Classical ~ 1 + Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer7: Classical ~ Instrument + Voice + Harmony + PianoPlay + X16.minus.17 +
## lmer7:     NoClass + (1 + Instrument + Voice + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 16 9303.9 9395.1 -4635.9   9271.9
## lmer7          51 8812.4 9103.2 -4355.2   8710.4 561.49     35  < 2.2e-16
##
## randomIntModel
## lmer7          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at the BIC's for all the different combinations of random effects for classical, the same one was deemed the best since its BIC was the lowest. The model is of the form:

$Classical \sim Instrument + Voice + Harmony + PianoPlay + X16.minus.17 + NoClass + (1 + Instrument + Harmony|Subject)$

c) Classical Final Model

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
```

```
## Formula:
## Classical ~ Instrument + Voice + Harmony + (1 + Harmony + Instrument |
##     Subject) + PianoPlay + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8801.5   9006.8  -4364.8   8729.5     2177
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6959 -0.5906  0.0108  0.5396  3.8490
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.18157  1.4770
##           HarmonyI-V-IV    0.09571  0.3094    0.70
##           HarmonyI-V-VI    1.71656  1.3102   -0.07  0.15
##           HarmonyIV-I-V    0.02247  0.1499    0.14 -0.16 -0.15
##           Instrumentpiano  1.63819  1.2799   -0.34 -0.47 -0.30 -0.06
##           Instrumentstring 3.63069  1.9054   -0.58 -0.30 -0.45  0.18  0.63
##  Residual                  2.38077  1.5430
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)       4.065115   0.257519  15.786
## Instrumentpiano   1.489874   0.181539   8.207
## Instrumentstring  3.281655   0.254913  12.874
## Voicepar3rd      -0.374066   0.080401  -4.653
## Voicepar5th      -0.325293   0.080377  -4.047
## HarmonyI-V-IV    -0.055420   0.100701  -0.550
## HarmonyI-V-VI     0.763262   0.190557   4.005
## HarmonyIV-I-V    -0.006025   0.094610  -0.064
## PianoPlay1        0.724709   0.365838   1.981
## PianoPlay2        2.364401   1.138505   2.077
## PianoPlay4        0.687001   0.502568   1.367
## PianoPlay5        0.709937   0.505739   1.404
## X16.minus.17     -0.080311   0.049809  -1.612
## NoClass          -0.031779   0.098970  -0.321
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Seeing which categories of design variables were most significant ignoring random effects for classical ratings.

Instrument

```
classicalInstr = lmer(Classical ~ (Instrument - 1) + Voice + Harmony + (Harmony+Instrument|Subject) +
                                PianoPlay + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(classicalInstr)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ (Instrument - 1) + Voice + Harmony + (Harmony + Instrument |
##     Subject) + PianoPlay + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8801.5   9006.8  -4364.8   8729.5     2177
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6959 -0.5906  0.0108  0.5396  3.8490
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.18156  1.4770
##           HarmonyI-V-IV    0.09571  0.3094    0.70
##           HarmonyI-V-VI    1.71655  1.3102   -0.07  0.15
##           HarmonyIV-I-V    0.02247  0.1499    0.14 -0.16 -0.15
##           Instrumentpiano  1.63819  1.2799   -0.34 -0.47 -0.30 -0.06
##           Instrumentstring 3.63069  1.9054   -0.58 -0.30 -0.45  0.18  0.63
##  Residual                  2.38077  1.5430
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                   Estimate Std. Error t value
## Instrumentguitar  4.065116   0.257519  15.786
## Instrumentpiano   5.554990   0.268888  20.659
## Instrumentstring  7.346770   0.268038  27.409
## Voicepar3rd      -0.374066   0.080401  -4.653
## Voicepar5th      -0.325293   0.080377  -4.047
## HarmonyI-V-IV    -0.055420   0.100701  -0.550
## HarmonyI-V-VI     0.763262   0.190557   4.005
## HarmonyIV-I-V    -0.006025   0.094610  -0.064
## PianoPlay1        0.724708   0.365838   1.981
## PianoPlay2        2.364400   1.138506   2.077
## PianoPlay4        0.687001   0.502568   1.367
## PianoPlay5        0.709936   0.505739   1.404
## X16.minus.17     -0.080311   0.049809  -1.612
## NoClass          -0.031779   0.098970  -0.321

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it

## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Harmony

```r
classicalHarm = lmer(Classical ~ Harmony + Voice + Instrument - 1 + (Harmony+Instrument|Subject) +
                                PianoPlay + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(classicalHarm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Harmony + Voice + Instrument - 1 + (Harmony + Instrument |
##     Subject) + PianoPlay + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8801.5   9006.8  -4364.8   8729.5     2177
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6959 -0.5906  0.0108  0.5396  3.8490
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.18157  1.4770
##           HarmonyI-V-IV    0.09571  0.3094    0.70
##           HarmonyI-V-VI    1.71655  1.3102   -0.07  0.15
##           HarmonyIV-I-V    0.02247  0.1499    0.14 -0.16 -0.15
##           Instrumentpiano  1.63819  1.2799   -0.34 -0.47 -0.30 -0.06
##           Instrumentstring 3.63069  1.9054   -0.58 -0.30 -0.45  0.18  0.63
##  Residual                  2.38077  1.5430
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                  Estimate Std. Error t value
## HarmonyI-IV-V     4.06512    0.25752  15.786
## HarmonyI-V-IV     4.00970    0.27954  14.344
## HarmonyI-V-VI     4.82838    0.29981  16.105
## HarmonyIV-I-V     4.05909    0.26020  15.600
## Voicepar3rd      -0.37407    0.08040  -4.653
## Voicepar5th      -0.32529    0.08038  -4.047
## Instrumentpiano   1.48987    0.18154   8.207
## Instrumentstring  3.28165    0.25491  12.874
## PianoPlay1        0.72471    0.36584   1.981
## PianoPlay2        2.36440    1.13851   2.077
## PianoPlay4        0.68700    0.50257   1.367
## PianoPlay5        0.70994    0.50574   1.404
## X16.minus.17     -0.08031    0.04981  -1.612
## NoClass          -0.03178    0.09897  -0.321

##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(x, correlation=TRUE)  or
```

```
##     vcov(x)        if you need it
```

```
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Voice

```r
classicalVoice = lmer(Classical ~ Voice + Harmony + Instrument - 1 + (Harmony+Instrument|Subject) +
                                PianoPlay + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(classicalVoice)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Voice + Harmony + Instrument - 1 + (Harmony + Instrument |
##     Subject) + PianoPlay + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8801.5   9006.8  -4364.8   8729.5     2177
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6959 -0.5906  0.0108  0.5396  3.8490
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.18156  1.4770
##           HarmonyI-V-IV    0.09571  0.3094    0.70
##           HarmonyI-V-VI    1.71655  1.3102   -0.07  0.15
##           HarmonyIV-I-V    0.02247  0.1499    0.14 -0.16 -0.15
##           Instrumentpiano  1.63818  1.2799   -0.34 -0.47 -0.30 -0.06
##           Instrumentstring 3.63068  1.9054   -0.58 -0.30 -0.45  0.18  0.63
##  Residual                  2.38077  1.5430
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                  Estimate Std. Error t value
## Voicecontrary     4.065115   0.257519  15.786
## Voicepar3rd       3.691050   0.257492  14.335
## Voicepar5th       3.739823   0.257542  14.521
## HarmonyI-V-IV    -0.055420   0.100701  -0.550
## HarmonyI-V-VI     0.763262   0.190557   4.005
## HarmonyIV-I-V    -0.006025   0.094610  -0.064
## Instrumentpiano   1.489874   0.181539   8.207
## Instrumentstring  3.281655   0.254913  12.874
## PianoPlay1        0.724710   0.365838   1.981
## PianoPlay2        2.364401   1.138505   2.077
## PianoPlay4        0.687002   0.502568   1.367
## PianoPlay5        0.709937   0.505738   1.404
```

```
## X16.minus.17     -0.080311   0.049809  -1.612
## NoClass          -0.031779   0.098970  -0.321
##
## Correlation matrix not shown by default, as p = 14 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

We'll start with the closest to 50/50 dichotomization, dichotomizing by Selfdeclare <= 2.

```
## Percentage of subjects that aren't musicians with a dichotomization at Selfdelcare <= 2: 60.09654 %
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ (1 + Harmony + Instrument | Subject) + (Instrument +
##     Voice + Harmony + PianoPlay + X16.minus.17 + NoClass) * Musician
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8806.4   9085.8  -4354.2   8708.4     2164
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6556 -0.5821  0.0150  0.5557  3.8114
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.90798  1.381
##           HarmonyI-V-IV    0.09241  0.304     0.68
##           HarmonyI-V-VI    1.32771  1.152    -0.14  0.08
##           HarmonyIV-I-V    0.02625  0.162     0.28 -0.15 -0.21
##           Instrumentpiano  1.56596  1.251    -0.39 -0.45 -0.23 -0.03
##           Instrumentstring 3.51210  1.874    -0.56 -0.28 -0.42  0.20  0.62
##  Residual                  2.37920  1.542
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                           Estimate Std. Error t value
## (Intercept)               4.275759   0.456286   9.371
## Instrumentpiano           1.156485   0.281599   4.107
## Instrumentstring          2.862276   0.395443   7.238
## Voicepar3rd              -0.346774   0.127271  -2.725
## Voicepar5th              -0.379531   0.127088  -2.986
## HarmonyI-V-IV             0.008973   0.158882   0.056
## HarmonyI-V-VI             1.527384   0.273205   5.591
## HarmonyIV-I-V             0.021816   0.150145   0.145
## PianoPlay1                0.939717   0.527892   1.780
## PianoPlay2                2.468296   1.081950   2.281
## PianoPlay4               -0.217839   0.720513  -0.302
## PianoPlay5                0.853758   0.618926   1.379
## X16.minus.17             -0.143304   0.080664  -1.777
## NoClass                  -0.118052   0.136501  -0.865
## Musician1                -0.288627   0.546119  -0.529
```

```
## Instrumentpiano:Musician1   0.557973   0.363803   1.534
## Instrumentstring:Musician1  0.702713   0.511918   1.373
## Voicepar3rd:Musician1      -0.044538   0.164144  -0.271
## Voicepar5th:Musician1       0.091087   0.164033   0.555
## HarmonyI-V-IV:Musician1    -0.108072   0.205014  -0.527
## HarmonyI-V-VI:Musician1    -1.279888   0.353408  -3.622
## HarmonyIV-I-V:Musician1    -0.045840   0.193759  -0.237
## PianoPlay1:Musician1       -0.471594   0.742681  -0.635
## PianoPlay4:Musician1        1.992666   0.969589   2.055
## PianoPlay5:Musician1       -1.025841   1.241382  -0.826
## X16.minus.17:Musician1      0.105428   0.101810   1.036
## NoClass:Musician1           0.151669   0.189726   0.799
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

From the above summary, we can see that the fixed effects that had coefficients that were statistically significantly different from 0 did change after dichotomizing by Selfdeclare with a split <= 2. We can tell since only PianoPlay and Harmony have abs(t-values) > 2 (which signifies significance, insignificance otherwise).

Now let's try a split of <= 1

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ (1 + Harmony + Instrument | Subject) + (Instrument +
##     Voice + Harmony + PianoPlay + X16.minus.17 + NoClass) * Musician
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8801.8   9075.5  -4352.9   8705.8     2165
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7026 -0.5776  0.0213  0.5501  3.8656
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.84359  1.3578
##           HarmonyI-V-IV    0.11307  0.3363    0.78
##           HarmonyI-V-VI    1.45412  1.2059    0.09  0.12
##           HarmonyIV-I-V    0.02461  0.1569    0.23  0.04 -0.35
##           Instrumentpiano  1.58413  1.2586   -0.25 -0.46 -0.41 -0.15
##           Instrumentstring 3.49731  1.8701   -0.58 -0.31 -0.58  0.07  0.62
##  Residual                  2.37741  1.5419
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 3.72489    0.27563  13.514
## Instrumentpiano             1.61031    0.20157   7.989
## Instrumentstring            3.47032    0.28195  12.308
## Voicepar3rd                -0.36762    0.09049  -4.062
## Voicepar5th                -0.30596    0.09046  -3.382
## HarmonyI-V-IV              -0.02636    0.11485  -0.230
```

```
## HarmonyI-V-VI               1.03176    0.20150   5.120
## HarmonyIV-I-V               0.03994    0.10664   0.375
## PianoPlay1                  0.68873    0.38375   1.795
## PianoPlay2                  2.41340    1.13674   2.123
## PianoPlay4                  0.42460    0.53268   0.797
## PianoPlay5                  0.83431    0.51008   1.636
## X16.minus.17               -0.03076    0.05198  -0.592
## NoClass                    -0.08499    0.12752  -0.666
## Musician1                   1.63437    0.58492   2.794
## Instrumentpiano:Musician1  -0.57185    0.43928  -1.302
## Instrumentstring:Musician1 -0.89981    0.61563  -1.462
## Voicepar3rd:Musician1      -0.02982    0.19664  -0.152
## Voicepar5th:Musician1      -0.09148    0.19663  -0.465
## HarmonyI-V-IV:Musician1    -0.13603    0.25006  -0.544
## HarmonyI-V-VI:Musician1    -1.27962    0.43943  -2.912
## HarmonyIV-I-V:Musician1    -0.21943    0.23217  -0.945
## PianoPlay1:Musician1        1.16777    1.19546   0.977
## PianoPlay4:Musician1        0.99479    1.23734   0.804
## X16.minus.17:Musician1     -0.28192    0.13448  -2.096
## NoClass:Musician1           0.10911    0.19492   0.560
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

From the above summary, we can see that the fixed effects that had coefficients that were statistically significantly different from 0 did change after dichotomizing by Selfdeclare with a split <= 1. We can tell since now X16.minus.17 and Harmony have abs(t-values) > 2 (which signifies significance, insignificance otherwise).

Now let's try a split of <= 3

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ (1 + Harmony + Instrument | Subject) + (Instrument +
##     Voice + Harmony + PianoPlay + X16.minus.17 + NoClass) * Musician
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8808.4   9087.8  -4355.2   8710.4     2164
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7074 -0.5792  0.0103  0.5555  3.8536
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.06773  1.438
##           HarmonyI-V-IV    0.10692  0.327     0.75
##           HarmonyI-V-VI    1.36671  1.169    -0.04  0.09
##           HarmonyIV-I-V    0.02404  0.155     0.22  0.04 -0.58
##           Instrumentpiano  1.60143  1.265    -0.40 -0.41 -0.26  0.10
##           Instrumentstring 3.48741  1.867    -0.62 -0.25 -0.41  0.38  0.62
##  Residual                  2.37611  1.541
## Number of obs: 2213, groups:  Subject, 62
```

```
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 3.38820    1.18433   2.861
## Instrumentpiano             1.10621    0.39344   2.812
## Instrumentstring            2.54487    0.54656   4.656
## Voicepar3rd                -0.43650    0.17599  -2.480
## Voicepar5th                -0.27504    0.17571  -1.565
## HarmonyI-V-IV               0.07268    0.22270   0.326
## HarmonyI-V-VI               1.93041    0.38259   5.046
## HarmonyIV-I-V               0.29132    0.20693   1.408
## PianoPlay1                  1.42864    1.16356   1.228
## PianoPlay2                  2.44110    1.09050   2.239
## PianoPlay4                 -0.63185    1.67804  -0.377
## PianoPlay5                  1.62414    1.18635   1.369
## X16.minus.17               -0.11346    0.21290  -0.533
## NoClass                    -0.13834    0.15625  -0.885
## Musician1                   0.67141    1.21523   0.553
## Instrumentpiano:Musician1   0.48525    0.44237   1.097
## Instrumentstring:Musician1  0.93219    0.61483   1.516
## Voicepar3rd:Musician1       0.07877    0.19779   0.398
## Voicepar5th:Musician1      -0.06359    0.19754  -0.322
## HarmonyI-V-IV:Musician1    -0.16211    0.25020  -0.648
## HarmonyI-V-VI:Musician1    -1.47735    0.43028  -3.433
## HarmonyIV-I-V:Musician1    -0.37654    0.23270  -1.618
## PianoPlay1:Musician1       -0.67019    1.24570  -0.538
## PianoPlay4:Musician1        1.71571    1.75630   0.977
## PianoPlay5:Musician1       -1.62145    1.60864  -1.008
## X16.minus.17:Musician1      0.04586    0.21860   0.210
## NoClass:Musician1           0.17000    0.20340   0.836
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

From the above summary, we can see that the fixed effects that had coefficients that were statistically significantly different from 0 did change after dichotomizing by Selfdeclare with a split <= 3. We can tell since now only Harmony has abs(t-values) > 2 (which signifies significance, insignificance otherwise).

Now let's try a split of <= 4

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ (1 + Harmony + Instrument | Subject) + (Instrument +
##     Voice + Harmony + PianoPlay + X16.minus.17 + NoClass) * Musician
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8807.4   9069.7  -4357.7   8715.4     2167
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7077 -0.5813  0.0119  0.5514  3.8870
##
## Random effects:
```

```
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      2.09336  1.4468
##           HarmonyI-V-IV    0.11051  0.3324    0.75
##           HarmonyI-V-VI    1.54371  1.2425    0.04  0.08
##           HarmonyIV-I-V    0.02114  0.1454    0.17  0.00 -0.45
##           Instrumentpiano  1.63459  1.2785   -0.35 -0.41 -0.30  0.01
##           Instrumentstring 3.61887  1.9023   -0.64 -0.26 -0.45  0.26  0.63
##  Residual                  2.37241  1.5403
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                -0.244942   2.183026  -0.112
## Instrumentpiano             1.166667   0.822596   1.418
## Instrumentstring            2.777778   1.156759   2.401
## Voicepar3rd                -0.944444   0.363043  -2.601
## Voicepar5th                -0.444444   0.363043  -1.224
## HarmonyI-V-IV               0.407407   0.461053   0.884
## HarmonyI-V-VI               2.666667   0.830846   3.210
## HarmonyIV-I-V               0.703704   0.427528   1.646
## PianoPlay1                  1.892090   1.906208   0.993
## PianoPlay2                  2.378784   1.103175   2.156
## PianoPlay4                  0.608896   0.486970   1.250
## PianoPlay5                  1.219927   0.572505   2.131
## X16.minus.17                0.557860   0.581752   0.959
## NoClass                    -0.007691   0.119138  -0.065
## Musician1                   4.339425   2.198203   1.974
## Instrumentpiano:Musician1   0.339857   0.843338   0.403
## Instrumentstring:Musician1  0.529402   1.185814   0.446
## Voicepar3rd:Musician1       0.599595   0.372254   1.611
## Voicepar5th:Musician1       0.125628   0.372248   0.337
## HarmonyI-V-IV:Musician1    -0.486391   0.472704  -1.029
## HarmonyI-V-VI:Musician1    -2.000823   0.851767  -2.349
## HarmonyIV-I-V:Musician1    -0.746305   0.438332  -1.703
## PianoPlay1:Musician1       -1.264543   1.955745  -0.647
## X16.minus.17:Musician1     -0.643178   0.582131  -1.105
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 4 columns / coefficients
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Again, we can see that the fixed effects that had coefficients that were statistically significantly different from 0 did change after dichotomizing by Selfdeclare with a split <= 4.Again, only Harmony has abs(t-values) > 2 (which signifies significance, insignificance otherwise).

In conclusion, dichotomizing by Musician is sensative to where in Selfdeclare that you choose to make the split.

Next we will go through and perform forward stepwise regression on the fixed effects by hand to predict popular ratings, one variable at a time to see whether they individually decrease AIC by a significant amount. If so, then add them to the model as a fixed effect.

```
## AIC with no fixed effects: 8801.531
```

```
## AIC with fixed effects: 8855.563
```

Seeing which categories of design variables were most significant ignoring random effects for popular ratings.

Instrument

```r
popularInstr = lmer(Popular ~ (Instrument - 1) + Voice + Harmony + (Harmony+Instrument|Subject) +
                              Selfdeclare + X16.minus.17 + NoClass, # fixed effects
                              data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(popularInstr)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ (Instrument - 1) + Voice + Harmony + (Harmony + Instrument |
##     Subject) + Selfdeclare + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8855.6   9066.5  -4390.8   8781.6     2176
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9233 -0.5844  0.0175  0.5796  3.3787
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.2838   1.1331
##           HarmonyI-V-IV    0.1414   0.3760    0.32
##           HarmonyI-V-VI    0.8524   0.9233   -0.02 -0.41
##           HarmonyIV-I-V    0.3071   0.5542   -0.37 -0.49 -0.30
##           Instrumentpiano  1.4957   1.2230   -0.11 -0.32 -0.26 -0.17
##           Instrumentstring 3.3622   1.8336   -0.29 -0.22 -0.24 -0.05  0.72
##  Residual                  2.4483   1.5647
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                   Estimate Std. Error t value
## Instrumentguitar  5.55783    0.34171  16.265
## Instrumentpiano   4.55646    0.36882  12.354
## Instrumentstring  2.83227    0.38972   7.268
## Voicepar3rd       0.17366    0.08153   2.130
## Voicepar5th       0.18014    0.08151   2.210
## HarmonyI-V-IV    -0.04127    0.10546  -0.391
## HarmonyI-V-VI    -0.27142    0.15039  -1.805
## HarmonyIV-I-V    -0.21165    0.11743  -1.802
## Selfdeclare2      1.23119    0.37176   3.312
## Selfdeclare3      1.03711    0.43254   2.398
## Selfdeclare4      0.85668    0.44877   1.909
## Selfdeclare5      1.38842    0.81806   1.697
## Selfdeclare6     -0.59566    1.37258  -0.434
## X16.minus.17      0.10005    0.04839   2.068
```

```
## NoClass           0.09807    0.11616   0.844
##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Harmony

```
popularHarm = lmer(Popular ~ Harmony + Voice + Instrument - 1 + (Harmony+Instrument|Subject) +
                                Selfdeclare + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(popularHarm)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Harmony + Voice + Instrument - 1 + (Harmony + Instrument |
##     Subject) + Selfdeclare + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8855.3   9066.3  -4390.7   8781.3     2176
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9512 -0.5801  0.0213  0.5830  3.3595
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.2877   1.1348
##           HarmonyI-V-IV    0.1430   0.3781    0.31
##           HarmonyI-V-VI    0.8523   0.9232   -0.03 -0.41
##           HarmonyIV-I-V    0.3069   0.5540   -0.38 -0.49 -0.30
##           Instrumentpiano  1.4964   1.2233   -0.11 -0.30 -0.25 -0.16
##           Instrumentstring 3.3581   1.8325   -0.27 -0.29 -0.27 -0.09  0.72
##  Residual                  2.4479   1.5646
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                  Estimate Std. Error t value
## HarmonyI-IV-V     5.55269    0.34161  16.255
## HarmonyI-V-IV     5.51124    0.35115  15.695
## HarmonyI-V-VI     5.28112    0.35999  14.670
## HarmonyIV-I-V     5.34089    0.33772  15.815
## Voicepar3rd       0.17363    0.08153   2.130
## Voicepar5th       0.18018    0.08150   2.211
## Instrumentpiano  -1.00154    0.17564  -5.702
## Instrumentstring -2.72558    0.24651 -11.057
```

```
## Selfdeclare2      1.23557    0.37149   3.326
## Selfdeclare3      1.04238    0.43223   2.412
## Selfdeclare4      0.85809    0.44844   1.913
## Selfdeclare5      1.37570    0.81746   1.683
## Selfdeclare6     -0.59187    1.37157  -0.432
## X16.minus.17      0.10106    0.04835   2.090
## NoClass           0.09920    0.11607   0.855
##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Voice

```r
popularVoice = lmer(Popular ~ Voice + Harmony + Instrument - 1 + (Harmony+Instrument|Subject) +
                                Selfdeclare + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```r
summary(popularVoice)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Voice + Harmony + Instrument - 1 + (Harmony + Instrument |
##     Subject) + Selfdeclare + X16.minus.17 + NoClass
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8855.3   9066.3  -4390.7   8781.3     2176
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9512 -0.5801  0.0213  0.5830  3.3595
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.2877   1.1348
##           HarmonyI-V-IV    0.1430   0.3781    0.31
##           HarmonyI-V-VI    0.8523   0.9232   -0.03 -0.41
##           HarmonyIV-I-V    0.3069   0.5540   -0.38 -0.49 -0.30
##           Instrumentpiano  1.4965   1.2233   -0.11 -0.30 -0.25 -0.16
##           Instrumentstring 3.3581   1.8325   -0.27 -0.29 -0.27 -0.09  0.72
##  Residual                  2.4479   1.5646
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                  Estimate Std. Error t value
## Voicecontrary     5.55269    0.34161  16.255
## Voicepar3rd       5.72633    0.34159  16.764
```

```
## Voicepar5th       5.73288    0.34162  16.781
## HarmonyI-V-IV    -0.04145    0.10558  -0.393
## HarmonyI-V-VI    -0.27158    0.15038  -1.806
## HarmonyIV-I-V    -0.21180    0.11741  -1.804
## Instrumentpiano  -1.00154    0.17564  -5.702
## Instrumentstring -2.72558    0.24651 -11.057
## Selfdeclare2      1.23557    0.37149   3.326
## Selfdeclare3      1.04238    0.43223   2.412
## Selfdeclare4      0.85809    0.44844   1.914
## Selfdeclare5      1.37570    0.81746   1.683
## Selfdeclare6     -0.59187    1.37157  -0.432
## X16.minus.17      0.10106    0.04835   2.090
## NoClass           0.09920    0.11607   0.855
```

```
##
## Correlation matrix not shown by default, as p = 15 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

After testing all variables, we found that the following reduced AIC by a significant amount on their own: Selfdeclare, X16.minus.17, and NoClass. I attempted to use fitlmer.fnc to find which subset of fixed effects to include. However, when done with AIC almost all of the fixed effects are included (which is a bit overkill…). And when done with BIC, none of the fixed effects were deemed significant enough (which also isn't super useful). Thus, we will take our by-hand results without an automated methods input. Below, I'll show that none of the other covariates reduced AIC.

```
# showing correct fixed effects did decrease AIC
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X16.minus.17, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9946.254
```

```
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          NoClass, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 8857.73
```

```
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Selfdeclare, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9945.001
```

```
# showing incorrect fixed effects didnt decrease AIC
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Instr.minus.Notes, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9946.98
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          PianoPlay, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9952.545
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ConsInstr, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9955.365
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          PachListen, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9952.934
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ClsListen, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9954.591
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X1990s2000s, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9953.295
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          CollegeMusic, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9949.586
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Composing, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9951.637
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          GuitarPlay, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9951.2
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          Instr.minus.Notes, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
```

```r
AIC(wrongFixedEffect)
```

```
## [1] 9946.98
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          X1990s2000s.minus.1960s1970s, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9954.013
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          APTheory, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9949.591
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          ConsNotes, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9950.312
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          KnowRob, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9947.315
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          KnowAxis, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9948.85
```

```r
wrongFixedEffect = lmer(Popular ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subject) +
                          OMSI, # fixed effects
                        data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FALSE)
AIC(wrongFixedEffect)
```

```
## [1] 9949.135
```

Investigating which random effect to include to the popular ratings model

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer1: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer1:     NoClass + (1 + Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248.0 9345.0 -4607.0   9214.0
## lmer1          22 8942.8 9068.2 -4449.4   8898.8 315.25      5  < 2.2e-16
```

```
##
## randomIntModel
## lmer1          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer2: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer2:     NoClass + (1 + Harmony | Subject)
##                Df  AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248 9345.0  -4607     9214
## lmer2          26 9206 9354.2  -4577     9154 60.079      9  1.294e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer3: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer3:     NoClass + (1 + Voice | Subject)
##                Df  AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248 9345.0 -4607.0     9214
## lmer3          22 9257 9382.5 -4606.5     9213 1.0077      5     0.9619
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer4: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer4:     NoClass + (1 + Voice + Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248.0 9345.0 -4607.0   9214.0
## lmer4          31 8953.7 9130.5 -4445.9   8891.7 322.31     14  < 2.2e-16
##
## randomIntModel
## lmer4          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer5: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer5:     NoClass + (1 + Instrument + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248.0 9345.0 -4607.0   9214.0
## lmer5          37 8855.3 9066.3 -4390.7   8781.3 432.7     20  < 2.2e-16
##
## randomIntModel
## lmer5          ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer6: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer6:     NoClass + (1 + Voice + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248.0 9345.0 -4607.0   9214.0
## lmer6          37 9220.1 9431.1 -4573.1   9146.1 67.933     20  3.946e-07
##
## randomIntModel
## lmer6          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Data: cleanData
## Models:
## randomIntModel: Popular ~ 1 + Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## randomIntModel:     NoClass + (1 | Subject)
## lmer7: Popular ~ Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 +
## lmer7:     NoClass + (1 + Instrument + Voice + Harmony | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## randomIntModel 17 9248.0 9345.0 -4607.0   9214.0
## lmer7          52 8870.9 9167.5 -4383.5   8766.9 447.11     35  < 2.2e-16
##
## randomIntModel
## lmer7          ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looking at the BIC's for all the different combinations of random effects, the same one was deemed the best since its BIC was the lowest. The model is of the form:

$Popular \sim Instrument + Voice + Harmony + Selfdeclare + X16.minus.17 + NoClass + (1 + Instrument + Harmony|Subject)$

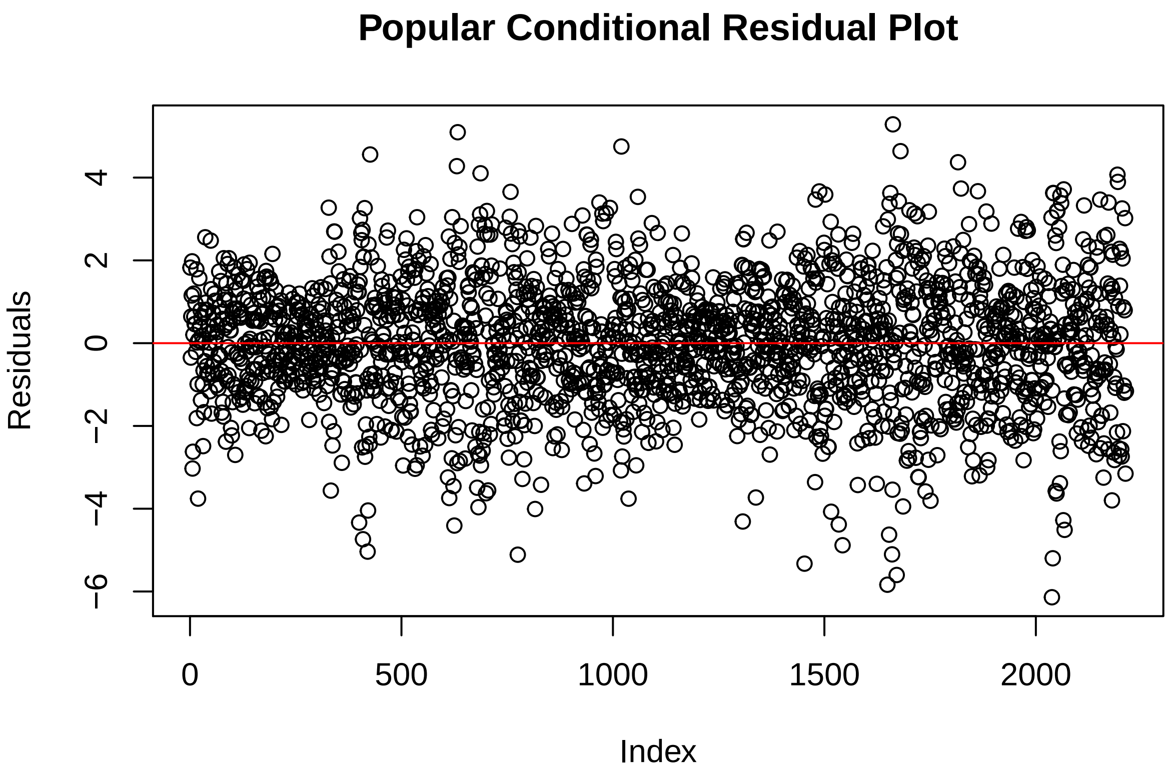
Final Model for Popular

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Instrument + Voice + Harmony + (1 + Harmony + Instrument |
##     Subject) + X16.minus.17 + NoClass + Selfdeclare
##    Data: cleanData
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   8855.6   9066.5  -4390.8   8781.6     2176
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9233 -0.5844  0.0175  0.5796  3.3787
##
## Random effects:
```

```
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.2838   1.1331
##           HarmonyI-V-IV    0.1414   0.3760    0.32
##           HarmonyI-V-VI    0.8524   0.9233   -0.02 -0.41
##           HarmonyIV-I-V    0.3071   0.5542   -0.37 -0.49 -0.30
##           Instrumentpiano  1.4957   1.2230   -0.11 -0.32 -0.26 -0.17
##           Instrumentstring 3.3622   1.8336   -0.29 -0.22 -0.24 -0.05  0.72
##  Residual                  2.4483   1.5647
## Number of obs: 2213, groups:  Subject, 62
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)       5.55783    0.34171  16.265
## Instrumentpiano  -1.00136    0.17560  -5.702
## Instrumentstring -2.72555    0.24664 -11.051
## Voicepar3rd       0.17366    0.08153   2.130
## Voicepar5th       0.18014    0.08151   2.210
## HarmonyI-V-IV    -0.04127    0.10546  -0.391
## HarmonyI-V-VI    -0.27142    0.15039  -1.805
## HarmonyIV-I-V    -0.21165    0.11743  -1.802
## X16.minus.17      0.10005    0.04839   2.068
## NoClass           0.09807    0.11616   0.844
## Selfdeclare2      1.23120    0.37176   3.312
## Selfdeclare3      1.03711    0.43254   2.398
## Selfdeclare4      0.85668    0.44877   1.909
## Selfdeclare5      1.38842    0.81806   1.697
## Selfdeclare6     -0.59566    1.37258  -0.434
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Conditional residual plot for popular model

```r
plot(residuals(fixedAndRandomEffectLmer), ylab = "Residuals", main = "Popular Conditional Residual Plot
  abline(h = 0, col = "red")
```



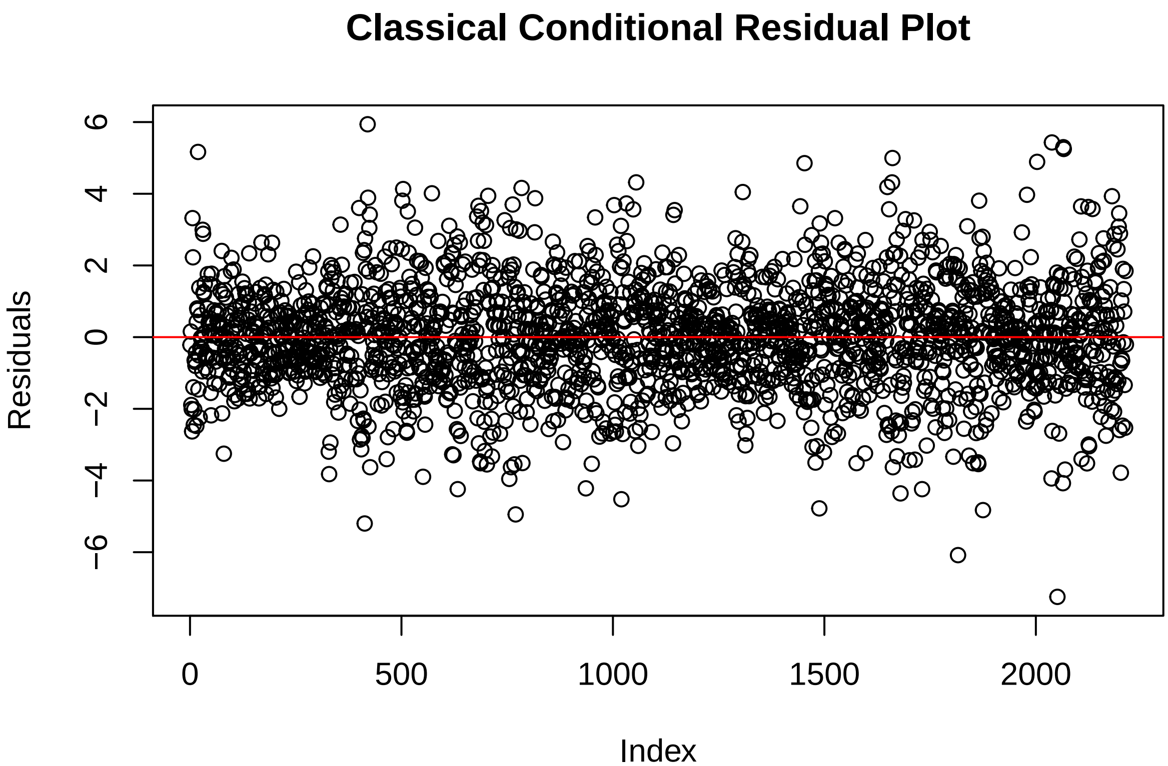

```
## integer(0)
```

The residuals look good for the popular ratings model. They have mean zero with no grouping structure and relatively constant variance.

```
fixedAndRandomEffectLmer = lmer(Classical ~ Instrument + Voice + Harmony + (1+Harmony+Instrument|Subjec
                                PianoPlay + X16.minus.17 + NoClass, # fixed effects
                                data = cleanData, control = lmerControl(optimizer = 'bobyqa'),REML = FA
```

```
## boundary (singular) fit: see ?isSingular
```

```
plot(residuals(fixedAndRandomEffectLmer), ylab = "Residuals", main = "Classical Conditional Residual Pl
  abline(h = 0, col = "red")
```




```
## integer(0)
```

The residuals look good for the classical ratings model. They have mean zero with no grouping structure and relatively constant variance.