Analyzing Fixed and Random Effects of Musical Predictors on Classical and Popular Music Stimulus Ratings

Jeffrey Ho*
jmho@andrew.cmu.edu

8 December 2019

Abstract

Instruments, harmonic motion, and voice leading are often seen as three main predictive factors that differentiate major genres of music. Given a sampling of song features and test subjects' musical backgrounds from Ivan Jimenez at the University of Pittsburgh, we will use exploratory data analysis, model fitting, and various statistical criteria to determine any associations between predictors in the musical data set provided data with classical and popular song stimula, as well as whether these results change between musicians and non-musicians. Through this analysis, we will justify our models' accuracy robustness based on prior intuition of musical genre differences and significance testing between various models we create.

1 Introduction

Visiting University of Pittsburgh composer and musicologist Ivan Jimenez and a student of his designed an experiment to discover associations between predictors such as instrument, harmonic motion, and voice leading on how listeners perceive classical and popular music. Primarily, they wanted to investigate how these three main factors affect classical and popular stimuli.

Additionally, Jimenez wants to determine whether the harmonic motion I-V-vi has a strong associating with classical ratings due to its prevalence in classical pieces such as Pachebel's Canon in D, despite many popular songs in the last ∼20 years using such a progression (in songs such as Paravonian's Axis of Evil). He also hypothesizes that contrary motion is commonly a strong predictor of classical ratings, as it historically has been a common voice leading among classical music.

Secondarily, Jimenez would like to determine whether musicians and non-musicians classify classical music, and whether these results are sensitive to how we choose to dichotomize whether one is a musician.

Finally, Jimenez would like to know whether the covariates provided as measures from his data collection differ between predicting classical and popular stimuli. If stimulus ratings do differ between classical and popular songs, then simply acknowledging the differences in covariates that predict each stimulus can provide a more quantifiable metric for classifying such music.

*Department of Statistics & Data Science, Carnegie Mellon University

2 Methods

2.1 Data Collection and Processing

First, we would like to determine if we want to perform any transformations on our variables. It is important to note that all of our variables are discrete other than OMSI, X16.minus.17, and NoClass (number of music classes taken). For reference, table 1 contains the list of predictors provided in our dataset:

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

Table 1: Variable definitions for Music Rating data from Ivan Jimenez, Univeresity of Pittsburgh.

Since we know that there are NA values associated with our data, we want to determine if we can simply eliminate all rows with NA values. We see this is not possible, because not only do some ratings inherenly not have values associated with them (ie. X2ndInstr, APTheory, etc.), but if we were to discard every single row with an NA value, then we'd only be left with 180 rows out of 2520.

Because we are trying to predict classical and popular stimuli, we can eliminate any rows in which the stimulus rating for both classical and popular are NA. Additionally, based on one examination of the unique values in our data, it appears that the value 19 may be either an outlier, unclean data, or another category we were not specified of beforehand, so we eliminate these values as well. We eliminate X1stInstr and X2ndInstr due to having many NA values (1512 and 2196), and then remove all rows NA values in the table afterward.

We won't be transforming any variables because they are all discrete, and transforming any variable in our dataset would make it more difficult to interpret in any model. [1]

2.2 Modeling Methodology

To determine optimal models predicting both classical and popular ratings, we can systematically test for fixed effects by adding a multitude of fixed effects to a model predicting ratings, and then reduce them with a criterion like AIC. Once we have a series of fixed effects that we've found to be significant, we can add random effects one by one through the use of R's lmer function. To test the significance of our random effect, we can compare AIC values or make REML false in our lmer models and use R's ANOVA function.

More specifically, to determine which predictors' levels are most significant, we can do this by eliminating the intercept of our best model. This will allow us to see the individual effects of each level of a predictor and compare t-values in a standardized manner.

To determine both whether any dichotomization of musician effects the responses as given by our predictors, we can create multiple dichotomizations, find our best model, and interact each term with musicians. If we take every fixed effect and include their interaction with musicians and reduce the model with a criterion like AIC, then we are left with fixed effects that are significant, any any term with an interaction with musician is influenced by whether one is a musician. Finally, we can add random effects one by one, conditioned on musician, and keep significant random effects in our model. Any differences between models depending on our dichotomization of musician will determine how sensitive our models and predictors are to how we dichotomize.

3 Results

3.1 Influences of Instrument, Harmonic Motion, and Voice Leading on Ratings

Firstly, Jimenez proposes that song scores may be influenced by three main factors of a song: the instrument, harmonic motion, and voice leading. More specifically, he hypothesizes that instrument has the strongest influence among these three predictors.

To determine the validity of this claim, we can begin by determining adding all interactions between these three variables as fixed effects on scores, and then eliminating the predictors in which their inclusion doesn't significantly improve the variance explained by their model with a criterion like AIC[2]. If instrument ends up being the most significant predictor of both types of scores, then Jimenez's claim will be successfully verified.

When we step backwards from a full model that includes all interactions between instrument, harmonic, motion, and voice leading in predicting classical scores with AIC as or criterion, we find that only instrument, harmonic motion, voice leading, and the interaction between harmony

[1] Just to gauge distributions of our variables, we plot bar charts of the distributions on pages 1-4 of the code appendix

[2] A more detailed process of this can be found in the Code Appendix, pages 14-27 for our model predicting classical scores, and 39-47 for our model predicting popular scores.

and voice are statistically significant. The coefficients pertaining to the model we obtained when backward stepping from our full model are listed in Table 2 (for simplicity, we eliminated the intercept term to catch the direct affects of each variable):

	Estimate	Std. Error	t value	Pr(>|t|)
Instrumentguitar	3.8032723	0.2156485	17.6364457	0.0000000
Instrumentpiano	5.4585042	0.2157396	25.3013553	0.0000000
Instrumentstring	7.3893863	0.2156335	34.2682667	0.0000000
HarmonyI-V-IV	0.2223321	0.2828291	0.7861005	0.4319307
HarmonyI-V-VI	1.2619038	0.2833874	4.4529278	0.0000091
HarmonyIV-I-V	-0.3023256	0.2822775	-1.0710224	0.2843287
Voicepar3rd	-0.3100775	0.2822775	-1.0984845	0.2721663
Voicepar5th	-0.1917304	0.2828291	-0.6779022	0.4979365
HarmonyI-V-IV:Voicepar3rd	-0.4399225	0.3999797	-1.0998621	0.2715657
HarmonyI-V-VI:Voicepar3rd	-0.7138692	0.4003752	-1.7830005	0.0747849
HarmonyIV-I-V:Voicepar3rd	0.7566102	0.3995909	1.8934619	0.0584855
HarmonyI-V-IV:Voicepar5th	-0.2223321	0.4003691	-0.5553178	0.5787587
HarmonyI-V-VI:Voicepar5th	-0.5314136	0.4003762	-1.3272857	0.1846127
HarmonyIV-I-V:Voicepar5th	0.3235134	0.3995909	0.8096115	0.4182897

Table 2: Coefficients of classical score model obtained from backward-stepping all interactions between instrument, harmonic motion, and voice leading

For each categorical variable, we can interpret the coefficient associated with its linear effect on classical scores as follows:

- Holding all else constant, if the instrument in the song is a [*insert category of instrument*], then the rating of how classical the stimulus sounds would increase by [*coefficient of Estimate "Instrument" in Table 2*] in expectation.
- Holding all else constant, if the harmonic motion in the song is [*insert category of harmonic motion*], then the rating of how classical the stimulus sounds would increase by [*coefficient of Estimate "Harmony" in Table 2*] in expectation.
- Holding all else constant, if the voice leading in the song is [*insert category of voice leading*], then the rating of how classical the stimulus sounds would increase by [*coefficient of Estimate "Harmony" in Table 2*] in expectation.
- Holding all else constant, if the instrument in the song is a [*insert category of voice leading*], then the rating of how classical the stimulus sounds would increase by [*coefficient of Estimate "Voicepar" in Table 2*] in expectation.
- Holding all else constant, if the harmonic motion in the song is [*insert category of harmonic motion*], then the rating of how classical the stimulus sounds would increase by an additional [*coefficient of Estimate "harmony:voice" in Table 2*] in expectation if the voice leading in the song is [*insert category of "Voicepar"*].

From the t-values in Figure 2, when standardizing errors associated with each coefficient in our model, the t-values which are most significant are those for instrument (only values with

$Pr(>|t|) < 2 \times 10^{-16}$). As a result, it is fair to conclude that out of instrument, harmonic motion, and voice leading, instrument is the most significant predictor.

Similarly, we use the above procedure to create a model predicting popular scores against all significant predictors between instrument, harmonic motion, voice leading, and all interactions between these three variables. By backward-stepping with AIC, we obtain a model that predicts popular scores against only instrument. Our model's coefficients for predicting popular scores is show in Table 3:

	Estimate	Std. Error	t value
Instrumentguitar	6.866019	0.0990705	69.30436
Instrumentpiano	5.689587	0.0996527	57.09415
Instrumentstring	3.842054	0.0989745	38.81864

Table 3: Coefficients of popular score model obtained from backward-stepping all interactions between instrument, harmonic motion, and voice leading

The interpretation of coefficients in our model predicting popular scores from instrument is the same as that of classical scores, and is as follows:

- Holding all else constant, if the instrument in the song is a [*insert category of instrument*], then the rating of how popular the stimulus sounds would increase by [*coefficient of Estimate "Instrument" in Table 3*] in expectation.

As a result, since we can conclude that instrument is the most significant predictor of both classical and popular scores, we can affirm Jimenez's hypothesis that among the three main predictors, instrument has the strongest influence on scores.

NOTE: For the next parts, we will analyze each predictor or condition based on our best-fitting models of classical and popular music on the rest of our predictors.

3.2 Effects of Harmonic Motion and Voice Leading in Classical Music Ratings

Harmonic Motion

Additionally, we would like to determine whether out of our levels of harmonic motion, the "I-V-VI" motion has a strong (or the strongest) association with classical ratings, and whether this is influenced on whether one is familiar with Pachelbel's rant.

To do this, we create a model predicting classical scores from combinations of fixed and random effects of our predictors and check whether harmonic motion is significant, the "I-V-VI" motion has a strong association with classical ratings, and the interaction term between harmonic motion and knowledge of Pachelbel's rant is significant.

After testing for the significance of both fixed and random effects of numerous predictors, our best model predicts classical scores from fixed effects of Instrument, Harmonic Motion, Voice, Being a self-declared musician, concentration on notes, knowledge of Pachelbel Rant, whether one took AP Music Theory, how well one plays piano, and how well one plays guitar, as well as the random effects of an intercept, Instrument, Harmony, and concentration on notes conditioned on subject. A table of coefficients of this model is provided in Table 4:

Fixed Effects

	Estimate	Std. Error	t value
(Intercept)	4.2305262	0.5187990	8.1544612
Instrumentpiano	1.6549973	0.2343314	7.0626363
Instrumentstring	3.5884122	0.3090029	11.6128741
HarmonyI-V-IV	0.0027762	0.1289957	0.0215220
HarmonyI-V-VI	0.8547666	0.2336320	3.6586026
HarmonyIV-I-V	0.0590977	0.1313905	0.4497868
Voicepar3rd	-0.4017796	0.0980707	-4.0968358
Voicepar5th	-0.3018946	0.0980223	-3.0798572
Selfdeclare2	-0.6026102	0.6121877	-0.9843552
Selfdeclare3	-0.1424384	0.7403084	-0.1924042
Selfdeclare4	-1.8101630	0.8072908	-2.2422690
Selfdeclare5	-0.5871076	1.1159014	-0.5261286
Selfdeclare6	-2.3976793	1.2433737	-1.9283657
ConsNotes1	0.4157356	0.5108263	0.8138493
ConsNotes3	-0.2290781	0.4355968	-0.5258950
ConsNotes4	-1.8374389	0.9182503	-2.0010219
ConsNotes5	-1.0750487	0.4455256	-2.4129900
KnowRob1	-0.8754403	0.5468730	-1.6008110
KnowRob5	1.0557363	0.3291160	3.2077942
APTheory1	1.5495720	0.3250198	4.7676237
PianoPlay1	0.2769674	0.4000833	0.6922744
PianoPlay4	0.8215966	0.3702709	2.2189067
PianoPlay5	1.8570789	0.5958228	3.1168309
GuitarPlay1	-0.1127487	0.6257413	-0.1801843
GuitarPlay2	1.8436333	1.1371451	1.6212823
GuitarPlay4	1.7606476	0.6242269	2.8205250
GuitarPlay5	-1.0693588	0.6048936	-1.7678459

Random Effects

Random Effect	Variance
(Intercept)	0.5127
Instrumentpiano	1.9496
Instrumentstring	3.6977
HarmonyI-V-IV	0.1830
HarmonyI-V-VI	1.4175
HarmonyIV-I-V	0.1952
APTheory1	0.8914

Table 4: Coefficients of best model predicting classical scores from relevant predictors

We can interpret our fixed effects as follows:

- Holding all else constant, if the instrument in the song is a [*level within a predictor*], then the rating of how classical the stimulus sounds would increase by [*coefficient of estimate in Table 4*] in expectation.

and our random effects as below[3]:

- Holding all else constant, the random effects of our intercept conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 0.5095$)
- Holding all else constant, the random effects of instrument conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 1.9394$) if our instrument is a piano rather than a guitar.
- Holding all else constant, the random effects of instrument conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 3.6935$) if our instrument is another string instrument rather than a guitar.
- Holding all else constant, the random effects of harmony conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 0.1618$) if our harmony is I-V-IV rather than a I-IV-V
- Holding all else constant, the random effects of harmony conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 0.1585$) if our harmony is I-V-VI rather than a I-IV-V
- Holding all else constant, the random effects of harmony conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 0.1909$) if our harmony is IV-I-V rather than a I-IV-V
- Holding all else constant, the random effects of AP Music Theory conditioned by subject is drawn iid from a normal distribution (ie. $eta_j \sim N(0, \tau^2)$, where $\tau^2 = 0.9495$) if the subject took AP Music Theory

Holding all else constant, self-declaration decreased with higher ratings of whether one is a musician and higher concentration on notes when predicting classical scores. In contrast, if the song's instrument wasn't a guitar, someone taking AP Music Theory, playing the piano, knowing of Rob Paravonian's Pachelbel Rant, and playing the guitar at an intermediate level generally increased classical scores.

From this initial model, harmonic motion is a significant predictor in modeling classical scores. Based on the model, the "I-V-VI" harmonic motion is a strong predictor, with the highest standardized t-value out of the harmonic motion levels (t-score=3.66). [4].

We can see that while both harmonic motion and knowledge of Pachebel's rant are significant predictors, their interaction is not significant. After running an ANOVA test comparing models

[3]Note: the random effects added onto each fixed effect can be interpreted as an additional change η_j increase on popular stimulus rating directly conditioned on the subject, holding all else constant.

[4]To see computation of significance in adding Harmony:KnowRob as a predictor and associated t-values, see pages 22-27 in the code appendix

with and without the interaction between harmonic motion and knowledge of Pachebel's rant, the probability of obtaining a chi-square value greater than the one we see in our model is 0.285, per Table 5:

	DF	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(¿Chisq)
Reduced Model	56	6202.0	6501.0	-3045.0	6090.0			
Full Model	62	6206.6	6537.7	-3041.3	6082.6	7.3951	6 0.2858	

Table 5: ANOVA Test for significance of interaction between Harmonic Motion and Knowledge of Pachebel's Rant

As a result, we can conclude that the "I-V-VI" motion has a strong association with classical ratings, and this association is not influenced on whether one is familiar with Pachelbel' rant.

Voice Leading

Furthermore, we would like to determine whether out of our levels of Voice Leading, contrary motion has a strong (or the strongest) association with classical ratings.

Above, we created a model predicting classical scores from fixed and random effects of relevant predictors, and the coefficients pertaining to our final model can be found in Table 4. However, if we remove our intercept from our model, we expect that the level of voice leading corresponding to the most extreme t-value would have the strongest association with classical ratings[5].

When doing this, we find that contrary motion is indeed the strongest level of voice leading, with a t-value of 12.118 (compared to 11.374 and 11.129 for 5th and 3rd motions, respectively).

3.3 Differences in Classical Scores for Musicians and Non-Musicians

We want to try and distinguish between musicians and non-musicians, and determine whether how we make such a classification significantly affects any model we might produce as a result. To test this hypothesis, we will perform the same analysis as we did in problem 2(b), but try different dichotomizations of whether one is a musician.

Once we form dichotomizations, we can add significant predictors into a model that predicts classical ratings. To do this, we add interactions between musician and our three main predictors (instrument, harmonic motion, and and voice leading), reduce the model with a criterion like AIC to determine fixed effects Then, we can proceed by adding random effects that are conditioned on our musician dichotomization to determine which random effects are statistically significant.[6].

For our first model, our dichotomization yielded the following coefficients of random effects, as stated in Table 6:

[5] This computation can be found on pages 12-13

[6] To see more detailed testing of our dichotomizations, see pages 28-38 of the code appendix

Fixed Effects

	Estimate	Std. Error	t value
(Intercept)	3.6774260	0.3047609	12.0665922
Instrumentpiano	1.9611366	0.3120719	6.2842459
Instrumentstring	4.1441076	0.4040581	10.2562183
HarmonyI-V-IV	0.0096618	0.1705940	0.0566364
HarmonyI-V-VI	0.2667938	0.2877666	0.9271186
HarmonyIV-I-V	0.0048309	0.1666988	0.0289799
Voicepar3rd	-0.4016270	0.0987123	-4.0686623
Voicepar5th	-0.3011125	0.0986633	-3.0519190
musicians	0.4051169	0.4390092	0.9227982
Instrumentpiano:musicians	-0.6600728	0.4581018	-1.4408867
Instrumentstring:musicians	-1.1941076	0.5924284	-2.0156151
HarmonyI-V-IV:musicians	-0.0175020	0.2508085	-0.0697823
HarmonyI-V-VI:musicians	1.2612548	0.4221833	2.9874579
HarmonyIV-I-V:musicians	0.1151086	0.2447493	0.4703123

Random Effects

Random Effect	Variance
(Intercept)	1.6443
HarmonyI-V-IV	0.1141
HarmonyI-V-VI	1.3478
HarmonyIV-I-V	0.0839
Instrumentpiano	1.8226
Instrumentstring	3.3377

Table 6: Coefficients of best model predicting classical scores from first musician dichotomization

One first, naive split is whether one self-declared themselves as a musician or not. If we split based on whether self-declared store is at least a 3/6, then we get that 713 rows have been evaluated by musicians, and 827 rows have been evaluated by non-musicians. We find that the interaction of our dichotomization with both instrument and harmony were significant fixed effects, while we found no random effects that were significant when conditioned on musisican.

A second distinction of whether one is a musician can be determined by whether one concentrated on the notes with a rating of at least 3/5 and whether one concentrated on the instruments with a rating of at more than 4/5. This yields 864 musicians and 676 non-musicians. We find that the interaction of our dichotomization with instrument was a significant fixed effect, while we found no random effects that were significant when conditioned on musisican.

For our second model, our dichotomization yielded the following coefficients of random effects, as stated in Table 7:

Fixed Effects

	Estimate	Std. Error	t value
(Intercept)	4.0128529	0.3318303	12.0930893
Instrumentpiano	1.6323761	0.3392886	4.8111735
Instrumentstring	3.5190623	0.4295625	8.1922018
HarmonyI-V-IV	0.0018843	0.1245665	0.0151269
HarmonyI-V-VI	0.8539679	0.2314533	3.6895909
HarmonyIV-I-V	0.0586729	0.1241444	0.4726186
Voicepar3rd	-0.4016817	0.0986912	-4.0700847
Voicepar5th	-0.3015049	0.0986423	-3.0565478
musicians	-0.2632399	0.4326597	-0.6084226
Instrumentpiano:musicians	0.0387272	0.4403517	0.0879461
Instrumentstring:musicians	0.1243721	0.5341619	0.2328360

Random Effects

Random Effect	Variance
(Intercept)	1.6684
HarmonyI-V-IV	0.1091
HarmonyI-V-VI	1.7451
HarmonyIV-I-V	0.1062
Instrumentpiano	1.9164
Instrumentstring	3.6956

Table 7: Coefficients of best model predicting classical scores from second musician dichotomization

A third distinction of whether one is a musician can be determined by whether one rated their piano playing or guitar playing at at least 2/5. We get that 784 rows of our dataset were evaluated by musicians, and 756 were evaluated by non-musicians. We find that the interaction of our dichotomization with both instrument and harmony were significant fixed effects, while we found that our intercept had a significant random effect when conditioned on musician.

For our third model, our dichotomization yielded the following coefficients of random effects, as stated in Table 8:

Fixed Effects

	Estimate	Std. Error	t value
(Intercept)	3.4249980	0.3086516	11.0966488
Instrumentpiano	1.7579365	0.3334023	5.2727179
Instrumentstring	3.8293651	0.4389209	8.7244984
HarmonyI-V-IV	0.1640212	0.1809828	0.9062802
HarmonyI-V-VI	0.4814815	0.3217158	1.4966053
HarmonyIV-I-V	0.1481481	0.1797729	0.8240849
Voicepar3rd	-0.4019058	0.0985217	-4.0793653
Voicepar5th	-0.3016595	0.0984730	-3.0633729
musicians	0.8626698	0.4242199	2.0335439
Instrumentpiano:musicians	-0.2040301	0.4666131	-0.4372576
Instrumentstring:musicians	-0.4711443	0.6136676	-0.7677516
HarmonyI-V-IV:musicians	-0.3182393	0.2535661	-1.2550546
HarmonyI-V-VI:musicians	0.7286247	0.4501110	1.6187665
HarmonyIV-I-V:musicians	-0.1752610	0.2515895	-0.6966148

Random Effects

Random Effect	Variance
Intercept)	1.518
HarmonyI-V-IV	0.1348
HarmonyI-V-VI	1.620
HarmonyIV-I-V	0.1256
Instrumentpiano	1.919
Instrumentstring	3.631

Table 8: Coefficients of best model predicting classical scores from third musician dichotomization

As a result, from this small series of tests, we conclude that while in people who self-identify as musicians may be influenced by things that do not influence non-musicians in some cases, our results are sensitive to where we choose to dichotomize musicians.

3.4 Predicting Popular Scores from Fixed and Random Effects of Dataset Predictors

In order to analyze the differences in predictors for classical and popular music, we must first begin by developing an optimal model that predicts popular music ratings from both fixed and random effects of other predictors in our dataset similar to how we created a model predicting classical ratings from the same predictors in Section 3.2.

We add all of predictors as fixed effects, and eliminate those that aren't statistically significant with a criterion like AIC. Then, by incrementally adding random effects, we can finalize a single model we believe is sufficient in effectively predicting popular music ratings.

The model we finalized on for predicting popular music ratings depends on the fixed effects of Instrument, Voice, Self-declared musician rating, concentration on instrument, concentration on notes, familiarity with Pachebel's Canon in D, classical music listening rating, knowledge of Axis of Evil's Comedy bit on the 4 Pachelbel chords,pop listening ratings from 1990s and 2000s, and

Composing, as well as the random effects of instrument and voice conditioned on subject[7]. The equation of our final model can be illustrated as

$Popular_i = \alpha_{0j[i]} + \alpha_{1j[i]} Instrument_i + \alpha_{2j[i]} Voice_i + \beta_{3i} Selfdeclare_i + \beta_{4i} ConsInstr_i + \beta_{5i} ConsNotes_i + \beta_{6i} PachListen_i + \beta_{7i} ClsListen_i + \beta_{8i} KnowAxis_i + \beta_{9i} X1990s2000s_i + \beta_{10i} Composing_i + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$, where $\sigma^2 = 2.93849$.

Coefficients of this model can be found in Table 9:

[7] To see the full process of selecting a final model that predicts popular ratings from our predictors, see pages 39-47 of the code appendix

Fixed Effects

	Estimate	Std. Error	t value
(Intercept)	7.5673004	1.2126096	6.2405084
Instrumentpiano	-1.1662356	0.2280376	-5.1142255
Instrumentstring	-3.0239784	0.2680375	-11.2819248
Voicepar3rd	0.1764828	0.1118371	1.5780351
Voicepar5th	0.2309608	0.1108334	2.0838565
Selfdeclare2	-0.3780865	0.3575288	-1.0574996
Selfdeclare3	-1.7418698	0.5307413	-3.2819563
Selfdeclare4	-0.1748599	0.3757979	-0.4653030
Selfdeclare5	0.1057470	0.6700287	0.1578246
Selfdeclare6	-1.9978262	1.0263844	-1.9464697
ConsInstr0.67	-4.9618777	1.3644303	-3.6365930
ConsInstr1	0.4006768	1.0178512	0.3936497
ConsInstr1.67	2.6656820	1.5750573	1.6924350
ConsInstr2.33	1.9001651	1.0550492	1.8010203
ConsInstr2.67	0.1436741	1.0662734	0.1347442
ConsInstr3	-0.1357771	1.3152307	-0.1032344
ConsInstr3.33	4.6355578	1.3943630	3.3244985
ConsInstr3.67	0.3362448	1.3057110	0.2575185
ConsInstr4	-0.3188719	1.2671629	-0.2516423
ConsInstr4.33	1.6828681	1.3726352	1.2260127
ConsInstr5	1.9355953	1.4849871	1.3034425
ConsNotes1	1.1199518	0.6245881	1.7931044
ConsNotes3	-0.3681176	0.4228662	-0.8705298
ConsNotes4	1.9553423	0.7967944	2.4540112
ConsNotes5	0.4180971	0.5702421	0.7331923
PachListen3	-1.7314820	0.7872750	-2.1993358
PachListen5	-2.7273810	0.4537431	-6.0108484
ClsListen1	0.9382561	0.4409526	2.1277934
ClsListen3	0.5776399	0.4529840	1.2751882
ClsListen4	0.7597136	0.8793866	0.8639131
ClsListen5	-1.3294615	0.5665096	-2.3467592
KnowAxis1	3.4702066	0.7814128	4.4409391
KnowAxis5	0.0832981	0.3287620	0.2533690
X1990s2000s2	1.6436366	0.6004118	2.7375157
X1990s2000s3	1.1601454	0.4874164	2.3801935
X1990s2000s4	1.5063101	1.0697865	1.4080473
X1990s2000s5	0.4708066	0.4736876	0.9939179
Composing1	-0.5520028	0.2870140	-1.9232610
Composing2	0.8179189	0.3988927	2.0504733
Composing3	-0.3188115	0.3610982	-0.8828941
Composing4	-1.5005116	0.5130034	-2.9249546

Random Effects

Random Effect	Variance
(Intercept)	1.283222
Instrumentpiano	1.617602
Instrumentstring	2.556281
Voicepar3rd	0.039568
Voicepar5th	0.004443

Table 9: Coefficients of best model predicting popular scores from significant predictors

In short, to interpret:

- fixed effects (β_{ij}'s): Holding all else constant, if predictor i's fixed effect lies in group j rather than the base level assumed by our intercept term, then in expectation our popular stimulus score increases by β_{ij}
- random effects (η_{ij}'s): Holding all else constant, if predictor i's random effect lies in group j rather than the base level assumed by our intercept term, then in expectation our popular stimulus score increases by an iid draw from $N(0, \tau_j^2)$, where τ_j^2 is the variance associated with a random effect within the predictor's group j, conditioned on subject.

Holding all else constant, if the song's instrument wasn't a guitar, low or high self-declaration of being a musician, higher familiarity of Pachelbel's Canon in D, and more composing skills generally result in decreased popular music scores. Concentration on instrument seemed to have mixed effects on popular music scores, while greater exposure to 1990s and 2000's popular songs generally increased one's popular song score.

4 Discussion

4.1 Differences in Predictors for Classical and Popular Scores

Influence on Main Experimental Factors (Instrument, Harmony, and Voice)

Our models developed in predicting classical scores from instrument, harmony, voice showed a statistically significant dependence on all three variables on predictors. We found that in our final model, classical scores depended on both the fixed and random effects of instrument and harmony, while we only found a statistically significant fixed effect dependence on voice.

On the other hand, the models we created in predicting popular scores showed a statistically significant dependence on only instrument and voice. We found that out of these three predictors, our model only found the fixed effects of instrument to be statistically significant, while it also incorporated the random effects of voice to be significant as well, conditioned on subject.

Our secondary hypothesis of whether musicians and non-musicians might be influenced by different predictors was found to be plausible, as our results in Sections 3.3 and 3.4 for both classical and popular song scores indicate that there exists predictors in which random effects are statistically significant when conditioned on a dichotomized "musician" variable. However, we found that while determining which random effects are significant was sensitive to the methods of dichotomization when predicting classical scores, predicting popular scores were not nearly as sensitive to methods of dichotomization. For predicting popular scores, we found that only none of our variables were significant depending on dichotomization, but the fixed effects of instrument and harmony were generally dependent on musician dichotomization.

Variance Components

The models that we created for predicting both classical and popular song scores included multiple variance components.

As found in Section 3.3, we predicted classical scores using variance components stemming from random effects that included our intercept, instrument, harmony, and taking AP Music Theory, conditioned on the subject associated with a data point.

As found in Section 3.4, we predicted popular scores using variance components stemming from random effects included our intercept, instrument, and voice, conditioned on the subject associated with a data point.

In both models, we found the intercept and instrument to be statistically significant in displaying random effects that varied depending on subject. The indication that we had multiple random effects indicates that our model incorporates effects statistically significantly different than what's defined for a standard repeated measures model.

Other Individual Covariates

As found in Section 3.3, we predicted classical song scores from fixed effects of instrument, harmony, self-declaring as a musician, concentration on notes, knowledge of Pachebel rant, whether one took AP Music Theory, one's piano playing rating, and one's guitar playing rating. As found in Section 3.4, we predicted popular song scores from fixed effects of instrument, self-declaring as a musician, concentration on instrument, concentration on notes, knowledge of Axis of Evil's Comedy bit, familiarity with Pachebel's Canon in D, how much one listened to classical music, how much one listened to popular music, and whether one has conducted before.

While most inferences we can extract from our models show consistent associations with what we'd expect, one would intuitively think that higher ratings of being a musician would associate with higher classical or popular scores. Additionally, familiarity with note structure should would intuitively signal a higher classical stimulus score due to classical music's general score complexity. However, for the most part, directional associations between predictors and our popular stimulus scores seem fairly reasonable.

4.2 Future Scope

The results we found in predicting classical and popular scores indicated that they do indeed depend on different predictors, and our dichotomization of musician is significant in mapping dependencies on various predictors. In the future, this work may be generalized to other genres of music that may seem closer in nature (unlike the perceived distance between classical and popular music characterizations).

References

R Core Team (2017), *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Jimenez et al. (2012). University of Pittsburgh, Pittsburgh, Pennsylvania.

36-617 Final Project - Code Appendix

Jeffrey Ho

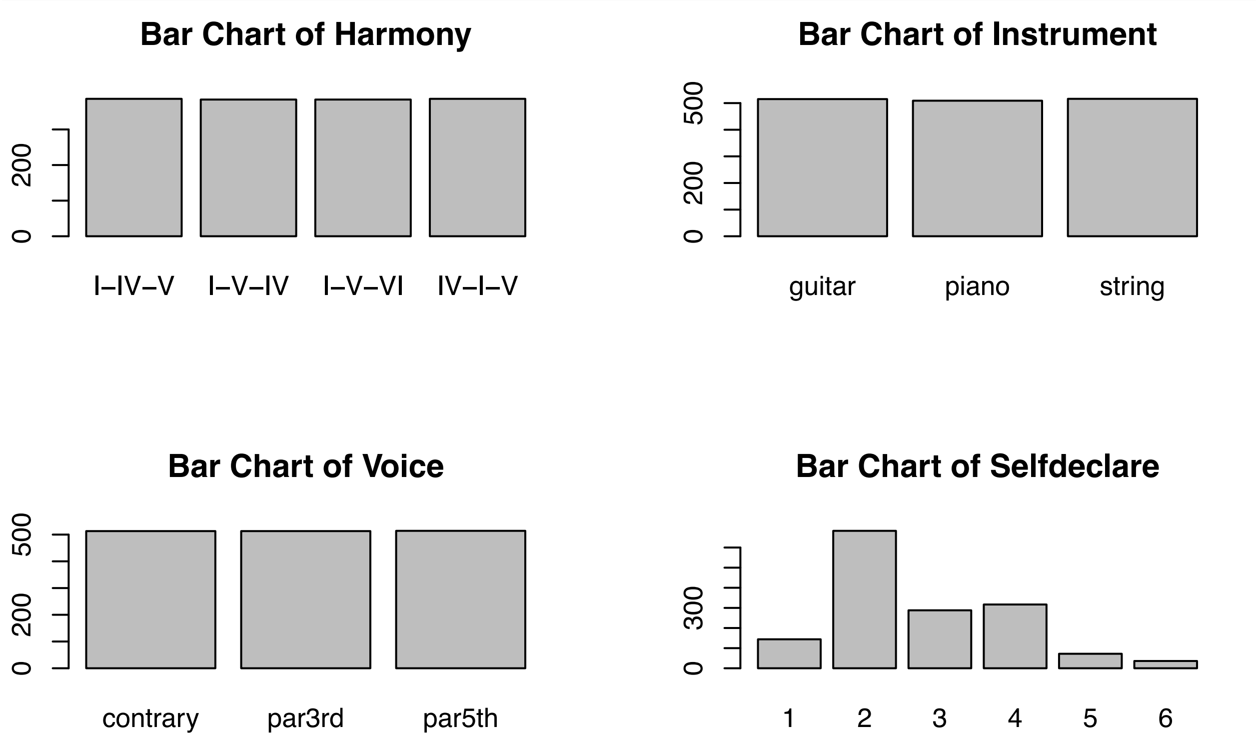
11/22/2019

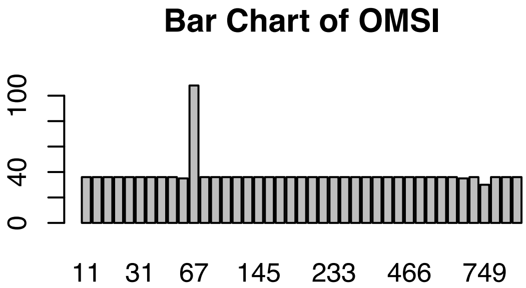
Code Appendix

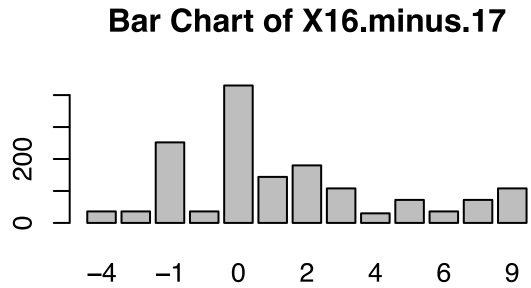
```r
# reading in data
ratings = read_csv("ratings.csv")%>%
  dplyr::select(-c(X1, first12))
```

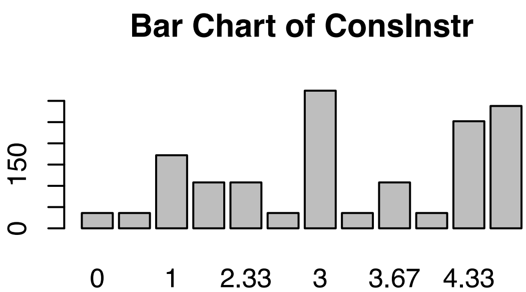
1. Data Cleaning, Transformations, EDA

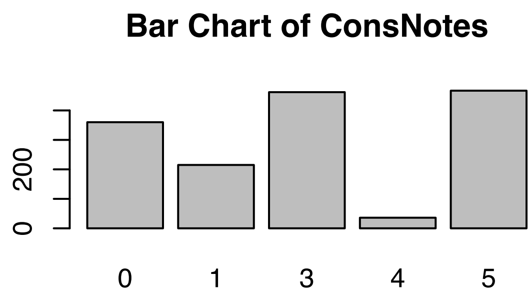
```r
# filter data by NA values
ratings = ratings %>%
  dplyr::filter(!is.na(Classical)) %>%
  dplyr::filter(!is.na(Popular)) %>%
  dplyr::filter(!is.na(Subject)) %>%
  dplyr::filter(Classical != 19.0) %>%
  dplyr::filter(Popular != 19.0) %>%
  dplyr::select(-c(X1stInstr, X2ndInstr))
ratings = na.omit(ratings)
```

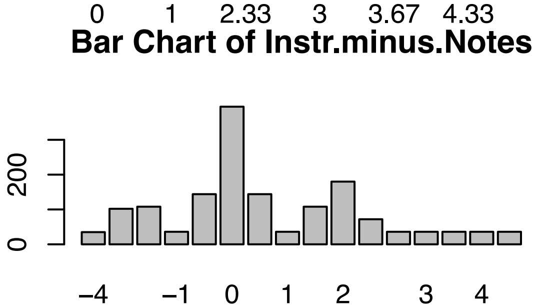


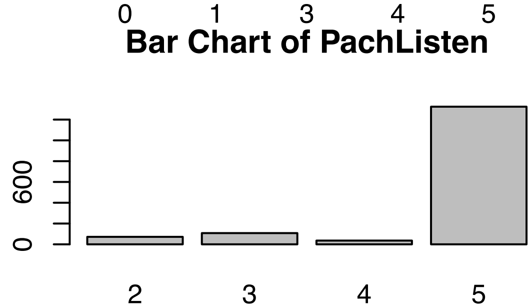

Bar Chart of OMSI
0
40
100
11
31
67
145
233
466
749

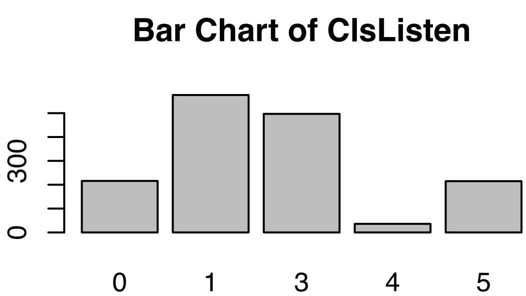

Bar Chart of X16.minus.17
0
200
−4
−1
0
2
4
6
9

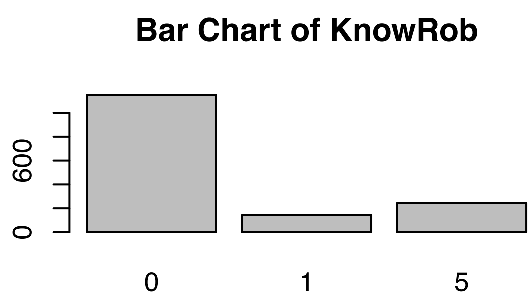

Bar Chart of ConsInstr
0
150
0
1
2.33
3
3.67
4.33

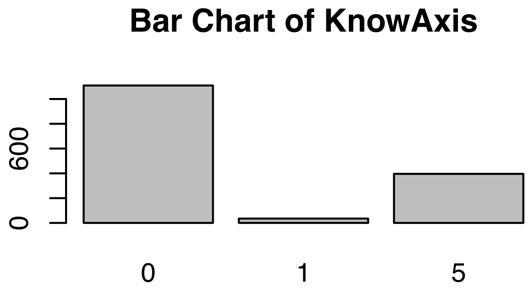

Bar Chart of ConsNotes
0
200
0
1
3
4
5

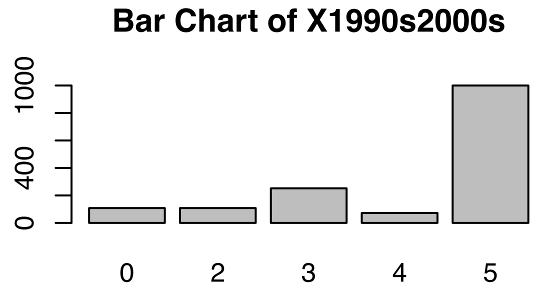

Bar Chart of Instr.minus.Notes
0
200
−4
−1
0
1
2
3
4

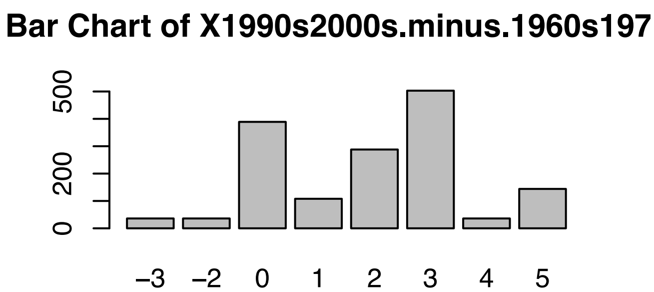

Bar Chart of PachListen
0
600
2
3
4
5

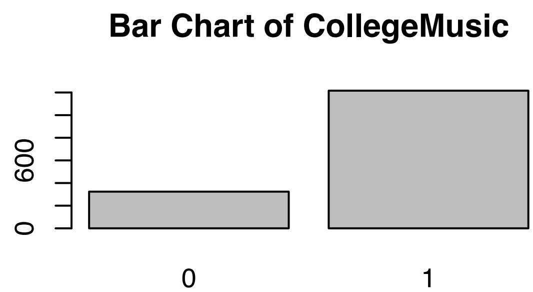

Bar Chart of ClsListen
0
300
0
1
3
4
5

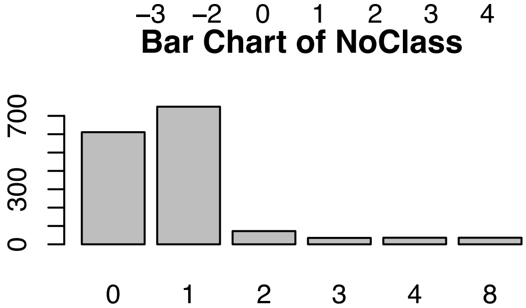

Bar Chart of KnowRob
0
600
0
1
5

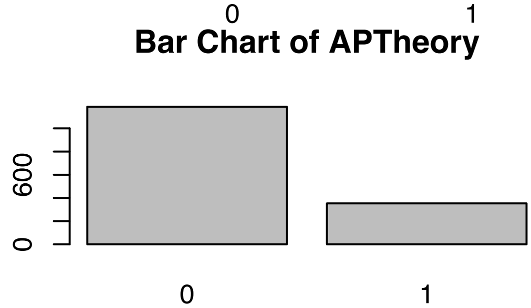

Bar Chart of KnowAxis
0
600
0 1 5

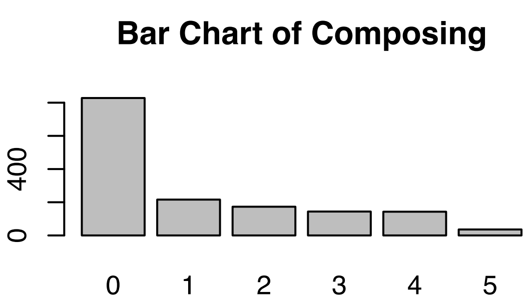

Bar Chart of X1990s2000s
0
400
1000
0 2 3 4 5

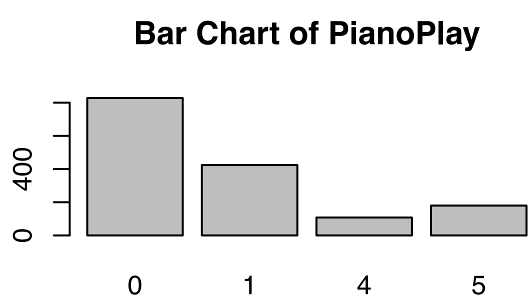

Bar Chart of X1990s2000s.minus.1960s197
0
200
500
−3 −2 0 1 2 3 4 5

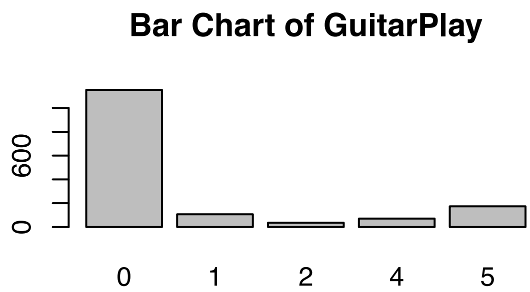

Bar Chart of CollegeMusic
0
600
0 1

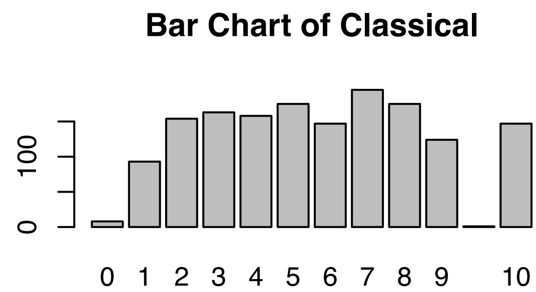

Bar Chart of NoClass
0
300
700
0 1 2 3 4 8

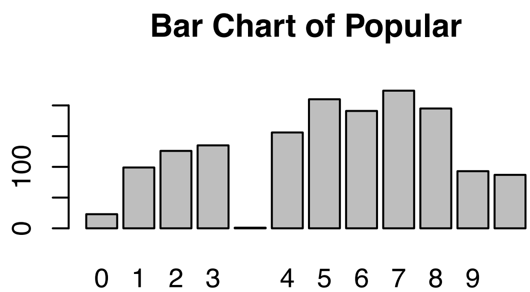

Bar Chart of APTheory
0
600
0 1


Bar Chart of Composing
0
400
0 1 2 3 4 5


Bar Chart of PianoPlay
0
400
0 1 4 5


Bar Chart of GuitarPlay
0
600
0
1
2
4
5


Bar Chart of Classical
0
100
0
1
2
3
4
5
6
7
8
9
10


Bar Chart of Popular
0
100
0
1
2
3
4
5
6
7
8
9

2. Main Experimental Factors

```r
# separate out classical and popular ratings
classical = ratings %>% dplyr::select(-Popular)
popular = ratings %>% dplyr::select(-Classical)
```

a.

We want to begin by examining the influence of the three main experimental factors (Instrument, Harmony & Voice) on Classical ratings, using conventional linear models and/or analysis of variance models.

We begin by fitting a multiple least-squares regression model on all interactions between our predictors in order to predict Classical ratings. Then, we can elminiate predictors that aren't statistically significant by stepping backward with AIC. We do this below, while printing the summary of our reduced model, the VIF values associated with each predictor, and the diagnostic plots associated with our reduced model:

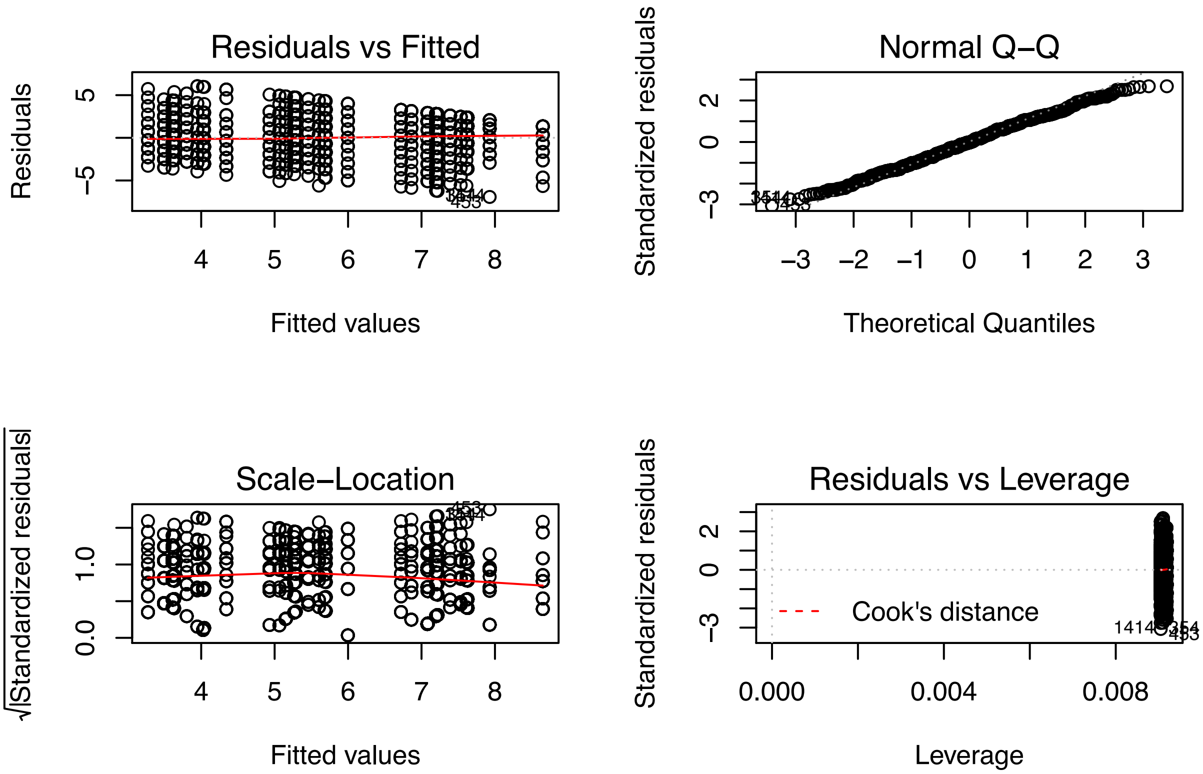
```r
classical_lm1 = lm(Classical ~ Instrument * Harmony * Voice - 1, data=classical)

classical_lm_step1 = stepAIC(classical_lm1, trace=FALSE)
x = summary(classical_lm_step1)
knitr::kable(x$coefficients)
```

	Estimate	Std. Error	t value	Pr(>|t|)
Instrumentguitar	3.8032723	0.2156485	17.6364457	0.0000000
Instrumentpiano	5.4585042	0.2157396	25.3013553	0.0000000
Instrumentstring	7.3893863	0.2156335	34.2682667	0.0000000
HarmonyI-V-IV	0.2223321	0.2828291	0.7861005	0.4319307
HarmonyI-V-VI	1.2619038	0.2833874	4.4529278	0.0000091
HarmonyIV-I-V	-0.3023256	0.2822775	-1.0710224	0.2843287
Voicepar3rd	-0.3100775	0.2822775	-1.0984845	0.2721663
Voicepar5th	-0.1917304	0.2828291	-0.6779022	0.4979365
HarmonyI-V-IV:Voicepar3rd	-0.4399225	0.3999797	-1.0998621	0.2715657
HarmonyI-V-VI:Voicepar3rd	-0.7138692	0.4003752	-1.7830005	0.0747849
HarmonyIV-I-V:Voicepar3rd	0.7566102	0.3995909	1.8934619	0.0584855
HarmonyI-V-IV:Voicepar5th	-0.2223321	0.4003691	-0.5553178	0.5787587
HarmonyI-V-VI:Voicepar5th	-0.5314136	0.4003762	-1.3272857	0.1846127
HarmonyIV-I-V:Voicepar5th	0.3235134	0.3995909	0.8096115	0.4182897

```r
vif(classical_lm_step1)
```

```
##                    GVIF Df GVIF^(1/(2*Df))
## Instrument     11.93840  3        1.511788
## Harmony       107.93929  3        2.182043
## Voice          47.78387  2        2.629180
## Harmony:Voice 432.03900  6        1.658162
```

```r
# plot diagnostic plots of our data
par(mfrow=c(2,2))
plot(classical_lm_step1)
```


Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

b.

i.

We fit a random-intercept model for each participant below:

```r
library(lme4)
lmer.intercept.only = lmer(Classical ~ 1 + (1 | Subject), data=classical,
                           control=lmerControl(optimizer = 'bobyqa'))
summary(lmer.intercept.only)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Classical ~ 1 + (1 | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
## REML criterion at convergence: 7251.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.67457 -0.78816 -0.03926  0.78468  2.60867
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1.394    1.181
##  Residual             6.103    2.470
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   5.6104     0.1908   29.41
```

ii.

We want to test whether our random intercept is needed in our model. We do this by creating a null model, one only containing a fixed intercept, before running an ANOVA test to determine whether adding the random intercept makes for a statistically significantly better model.

```r
lmer.fixed.only = lm(Classical ~ 1, data=classical)

summary(lmer.fixed.only)
```

```
##
## Call:
## lm(formula = Classical ~ 1, data = classical)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6101 -2.6101  0.3899  2.3899  4.3899
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.61006    0.06963   80.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.733 on 1539 degrees of freedom
```

```r
# check difference in AIC of two models (one including random intercept, one not)
# to determine significance of random intercept
AIC(lmer.fixed.only) - AIC(lmer.intercept.only)
```

```
## [1] 211.8694
```

When running an ANOVA test where our reduced model is a fixed intercept-only model and our full model is our random-intercept model, we can see that the difference in AIC of our linear model with only a fixed term and a linear model with the random intercept is 211.8694. We therefore have reasonable evidence that

iii.

Now that we know our random intercept improves our model compared to our fixed-intercept model, we re-fit our model found in (a), where our model predicted Classical from Instrument, Harmony, Voice, and the interaction between Harmony and Voice. To determine whether they are all significant when we incorporate our random intercept, we incrementally add each predictor to our model and run an ANOVA model where the full model has only one more predictor than its repsective reduced model. Below, we do this by adding Instrument, Harmony, Voice, and Harmony:Voice in order to determine if individiaul predictors statistically significantly improve our model. We find that adding all four predictors (similar to the model we found in (a), but with a random intercept) still improved our model because the p-values associated with each ANOVA test are statistically significant at any reasonable alpha level.

```r
lmer.random2 = lmer(Classical ~ Instrument + (1 | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.intercept.only, lmer.random2)
```

```
## Data: classical
## Models:
## lmer.intercept.only: Classical ~ 1 + (1 | Subject)
## lmer.random2: Classical ~ Instrument + (1 | Subject)
##                     Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.intercept.only  3 7256.2 7272.2 -3625.1   7250.2
## lmer.random2         5 6584.5 6611.2 -3287.2   6574.5 675.71      2
##                     Pr(>Chisq)
## lmer.intercept.only
## lmer.random2         < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lmer.random3 = lmer(Classical ~ Instrument + Harmony + (1 | Subject),
                    data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random2, lmer.random3)
```

```
## Data: classical
## Models:
## lmer.random2: Classical ~ Instrument + (1 | Subject)
## lmer.random3: Classical ~ Instrument + Harmony + (1 | Subject)
##              Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.random2  5 6584.5 6611.2 -3287.2   6574.5
## lmer.random3  8 6538.5 6581.2 -3261.2   6522.5 52.014      3  2.975e-11
##
## lmer.random2
## lmer.random3 ***
## ---
```

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

```r
lmer.random4 = lmer(Classical ~ Instrument + Harmony + Voice + (1 | Subject),
                    data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random3, lmer.random4)
```

```
## Data: classical
## Models:
## lmer.random3: Classical ~ Instrument + Harmony + (1 | Subject)
## lmer.random4: Classical ~ Instrument + Harmony + Voice + (1 | Subject)
##              Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.random3  8 6538.5 6581.2 -3261.2   6522.5
## lmer.random4 10 6530.1 6583.5 -3255.1   6510.1 12.346      2   0.002085 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lmer.random5 = lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice
                    + (1 | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random4, lmer.random5)
```

```
## Data: classical
## Models:
## lmer.random4: Classical ~ Instrument + Harmony + Voice + (1 | Subject)
## lmer.random5: Classical ~ Instrument + Harmony + Voice + Harmony:Voice + (1 |
## lmer.random5:     Subject)
##              Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.random4 10 6530.1 6583.5 -3255.1   6510.1
## lmer.random5 16 6520.2 6605.6 -3244.1   6488.2 21.951      6   0.001236 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

c.

i.

We first want to determine whether we can find any random effects beyond an intercept that might improve our model. We can do this by incrementally by adding random effects and running ANOVA tests on the addition of signle effects to determine their significance. We do this below, when adding combinations of instrument, harmony, and voice as random effects:

```
lmer.random6 = lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice
                    + (1 + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# check difference in AIC of two models (one including random intercept, one not)
# to determine significance of random intercept
AIC(lmer.random5) - AIC(lmer.random6)
```

```
## [1] 251.7203
```

```
lmer.random7 = lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice
                    + (1 + Instrument + Harmony | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# check difference in AIC of two models (one including random intercept, one not)
# to determine significance of random intercept
AIC(lmer.random6) - AIC(lmer.random7)
```

```
## [1] 82.49552
```

```
lmer.random8 = lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice
                    + (1 + Instrument + Harmony + Voice | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# check difference in AIC of two models (one including random intercept, one not)
# to determine significance of random intercept
AIC(lmer.random7) - AIC(lmer.random8)
```

```
## [1] -8.953879
```

ii.

Given we want our random effects to include instrument and harmonies that depend on subject, we want to re-investigate the influences of our three main experimental factors as fixed effects: instrument, harmony, and voice.

To do this, we take our random effects (intercept, instrument, and harmony), and begin including our three predictors in order. After adding each predictor, we run an ANOVA test on models that differ by a single predictor in order to find the model that most statistically significantly improves our prediction of classical stimulus rating.

```
# most reduced model - only contains random effects of intercept, instrument, harmony
lmer.random.bii.1 = lmer(Classical ~ 1 +
                           (1 + Instrument + Harmony | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

lmer.random.bii.2 = lmer(Classical ~ Instrument +
                           (1 + Instrument + Harmony | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random.bii.1, lmer.random.bii.2)
```

```
## Data: classical
## Models:
## lmer.random.bii.1: Classical ~ 1 + (1 + Instrument + Harmony | Subject)
## lmer.random.bii.2: Classical ~ Instrument + (1 + Instrument + Harmony | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.bii.1 23 6293.7 6416.5 -3123.9   6247.7
## lmer.random.bii.2 25 6227.5 6361.0 -3088.7   6177.5 70.262      2
##                    Pr(>Chisq)
## lmer.random.bii.1
## lmer.random.bii.2  5.531e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmer.random.bii.3 = lmer(Classical ~ Instrument + Harmony +
                           (1 + Instrument + Harmony | Subject),
                    data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random.bii.2, lmer.random.bii.3)
```

```
## Data: classical
## Models:
## lmer.random.bii.2: Classical ~ Instrument + (1 + Instrument + Harmony | Subject)
## lmer.random.bii.3: Classical ~ Instrument + Harmony + (1 + Instrument + Harmony |
## lmer.random.bii.3:     Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.bii.2 25 6227.5 6361.0 -3088.7   6177.5
## lmer.random.bii.3 28 6219.7 6369.2 -3081.8   6163.7 13.783      3
##                    Pr(>Chisq)
## lmer.random.bii.2
## lmer.random.bii.3   0.003217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmer.random.bii.4 = lmer(Classical ~ Instrument + Harmony + Voice +
                           (1 + Instrument + Harmony | Subject),
                    data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random.bii.3, lmer.random.bii.4)
```

```
## Data: classical
## Models:
## lmer.random.bii.3: Classical ~ Instrument + Harmony + (1 + Instrument + Harmony |
## lmer.random.bii.3:     Subject)
## lmer.random.bii.4: Classical ~ Instrument + Harmony + Voice + (1 + Instrument +
## lmer.random.bii.4:     Harmony | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.bii.3 28 6219.7 6369.2 -3081.8   6163.7
## lmer.random.bii.4 30 6205.8 6366.0 -3072.9   6145.8 17.841      2
##                    Pr(>Chisq)
## lmer.random.bii.3
## lmer.random.bii.4  0.0001336 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmer.random.bii.5 = lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice
                    + (1 + Instrument + Harmony | Subject), data=classical,
```

```r
                     REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(lmer.random.bii.4, lmer.random.bii.5)
```

```
## Data: classical
## Models:
## lmer.random.bii.4: Classical ~ Instrument + Harmony + Voice + (1 + Instrument +
## lmer.random.bii.4:     Harmony | Subject)
## lmer.random.bii.5: Classical ~ Instrument + Harmony + Voice + Harmony:Voice + (1 +
## lmer.random.bii.5:     Instrument + Harmony | Subject)
##                   Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.bii.4 30 6205.8 6366.0 -3072.9   6145.8
## lmer.random.bii.5 36 6185.9 6378.2 -3057.0   6113.9 31.902      6
##                   Pr(>Chisq)
## lmer.random.bii.4
## lmer.random.bii.5  1.704e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From our series of ANOVA tests above, we find that our model containing Instrument, Harmony, Voice, and Harmony:Voice as fixed effects, as well as our intercept, Instrument, and Harmony as random effects that depend on subject turn out to be our best model.

Below is the final model we have produced:

```r
summary(lmer.random.bii.5)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Instrument + Harmony + Voice + Harmony:Voice + (1 +
##     Instrument + Harmony | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6185.9   6378.2  -3057.0   6113.9     1504
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6169 -0.5611  0.0264  0.5316  3.3929
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.6820   1.2969
##           Instrumentpiano  1.9104   1.3822   -0.26
##           Instrumentstring 3.6865   1.9200   -0.54  0.62
##           HarmonyI-V-IV    0.1105   0.3324    0.72 -0.66 -0.80
##           HarmonyI-V-VI    1.7542   1.3245    0.21 -0.40 -0.59  0.46
##           HarmonyIV-I-V    0.1221   0.3494    0.18 -0.26 -0.29 -0.10  0.41
##  Residual                  2.4374   1.5612
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                   3.8027     0.2473  15.374
## Instrumentpiano               1.6554     0.2324   7.124
## Instrumentstring              3.5877     0.3085  11.628
```

```
## HarmonyI-V-IV                0.2115     0.2013   1.050
## HarmonyI-V-VI                1.2659     0.2810   4.506
## HarmonyIV-I-V               -0.3023     0.2016  -1.500
## Voicepar3rd                 -0.3101     0.1944  -1.595
## Voicepar5th                 -0.2034     0.1948  -1.044
## HarmonyI-V-IV:Voicepar3rd  -0.4172     0.2755  -1.514
## HarmonyI-V-VI:Voicepar3rd  -0.7074     0.2758  -2.565
## HarmonyIV-I-V:Voicepar3rd   0.7516     0.2752   2.731
## HarmonyI-V-IV:Voicepar5th  -0.2107     0.2758  -0.764
## HarmonyI-V-VI:Voicepar5th  -0.5238     0.2759  -1.898
## HarmonyIV-I-V:Voicepar5th   0.3352     0.2752   1.218
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

3. Individual Covariates

Now that we have a baseline model that predicts how classical a stimulus sounds, we want to determine any fixed effects that should be added to our model. To do this, we can fit a base model that

```
library(LMERConvenienceFunctions)

# make all discrete variables factors
cols = c("Selfdeclare", "ConsInstr", "ConsNotes", "PachListen", "ClsListen", "KnowRob", "KnowAxis", "X19
classical[cols] <- lapply(classical[cols], as.factor)
popular[cols] <- lapply(popular[cols], as.factor)
```

a.

We want to start by determining which variables ought to be added to our model as fixed effects. To do this, we will add all our predictors and backward step wtih AIC. However, due to many NA values, we will eliminate X2ndInstr, as this is not defined for the vast majority of rows inour dataset. We do this by reducing our model below:

```
lmer.full.1 = lmer(Classical ~ Subject + Harmony + Instrument + Voice + Selfdeclare + OMSI + X16.minus.1
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
```

```r
# going to delete all extraneous pages associated with trace output I can't hide
lmer.reduced.1 = fitLMER.fnc(lmer.full.1, method="AIC",
          ran.effects=c("(0 + Instrument | Subject)", "(0 + Harmony | Subject)"),
          set.REML.FALSE = TRUE)
```

By reducing our model, we simply want to gain a model with significant, but fewer fixed predictors than we'd get by fitting every single predictor. We go back to fit the random effects we found in Section 2, as below:

Our final model leaves the following predictors: Instrument, harmony, voice, selfdeclare, OMSI, ConsNotes, KnowRob, NoClass, APTheory, PianoPlay, GuitarPlay, and the random effect of our intercept, instrument, and harmony all conditioned on subject.

However, we do want to eliminate predictors that are collinear with other predictors in our dataset. To do this, we call vif() to determine whether we should eliminate predictors.

```r
# summary(lmer.reduced.1)
vif(lmer.reduced.1)
```

```
##                   GVIF Df GVIF^(1/(2*Df))
## Instrument    1.370700  2        1.082021
## Harmony       1.370718  3        1.053961
## Voice         1.000029  2        1.000007
## Selfdeclare 329.867893  5        1.785805
## OMSI          8.544558  1        2.923108
## ConsNotes     8.449957  4        1.305740
## KnowRob       4.649962  2        1.468461
## NoClass       6.188159  1        2.487601
## APTheory      2.625147  1        1.620230
## PianoPlay    17.786799  3        1.615659
## GuitarPlay   51.969196  4        1.638582
```

From our table of Generalized VIF values, we want to see if eliminating NoClass sufficiently reduces multicollinearity. We update our model below:

```r
lmer.reduced.1 = update(lmer.reduced.1, . ~ . - NoClass - OMSI, data=classical)
summary(lmer.reduced.1)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
##     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
##     Harmony | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6204.8   6466.5  -3053.4   6106.8     1491
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6060 -0.5658 -0.0030  0.5518  3.4598
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      0.2909   0.5394
##           Instrumentpiano  1.9203   1.3858    0.08
##           Instrumentstring 3.6846   1.9195   -0.25  0.62
```

```
##           HarmonyI-V-IV    0.1207   0.3475    0.53 -0.60 -0.75
##           HarmonyI-V-VI    1.7560   1.3252    0.22 -0.40 -0.59  0.47
##           HarmonyIV-I-V    0.1326   0.3641    0.51 -0.23 -0.25  0.06  0.42
##  Residual                  2.4931   1.5790
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)       4.207408   0.448363   9.384
## Instrumentpiano   1.653873   0.233329   7.088
## Instrumentstring  3.588421   0.308807  11.620
## HarmonyI-V-IV     0.002132   0.125554   0.017
## HarmonyI-V-VI     0.853828   0.231951   3.681
## HarmonyIV-I-V     0.058625   0.126495   0.463
## Voicepar3rd      -0.401961   0.098605  -4.076
## Voicepar5th      -0.301487   0.098556  -3.059
## Selfdeclare2     -0.651313   0.526993  -1.236
## Selfdeclare3      0.050942   0.653726   0.078
## Selfdeclare4     -1.692256   0.759992  -2.227
## Selfdeclare5      0.083614   1.077007   0.078
## Selfdeclare6     -1.795251   1.412940  -1.271
## ConsNotes1        0.420950   0.537470   0.783
## ConsNotes3       -0.089227   0.416576  -0.214
## ConsNotes4       -1.664757   0.787044  -2.115
## ConsNotes5       -1.055028   0.436830  -2.415
## KnowRob1         -1.098519   0.513497  -2.139
## KnowRob5          0.939866   0.441813   2.127
## APTheory1         1.608133   0.427625   3.761
## PianoPlay1        0.149390   0.364482   0.410
## PianoPlay4       -0.092718   0.657903  -0.141
## PianoPlay5        1.351829   0.661774   2.043
## GuitarPlay1      -0.384715   0.600056  -0.641
## GuitarPlay2       2.629477   1.132828   2.321
## GuitarPlay4       2.589625   0.712311   3.636
## GuitarPlay5      -1.006492   0.728812  -1.381
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

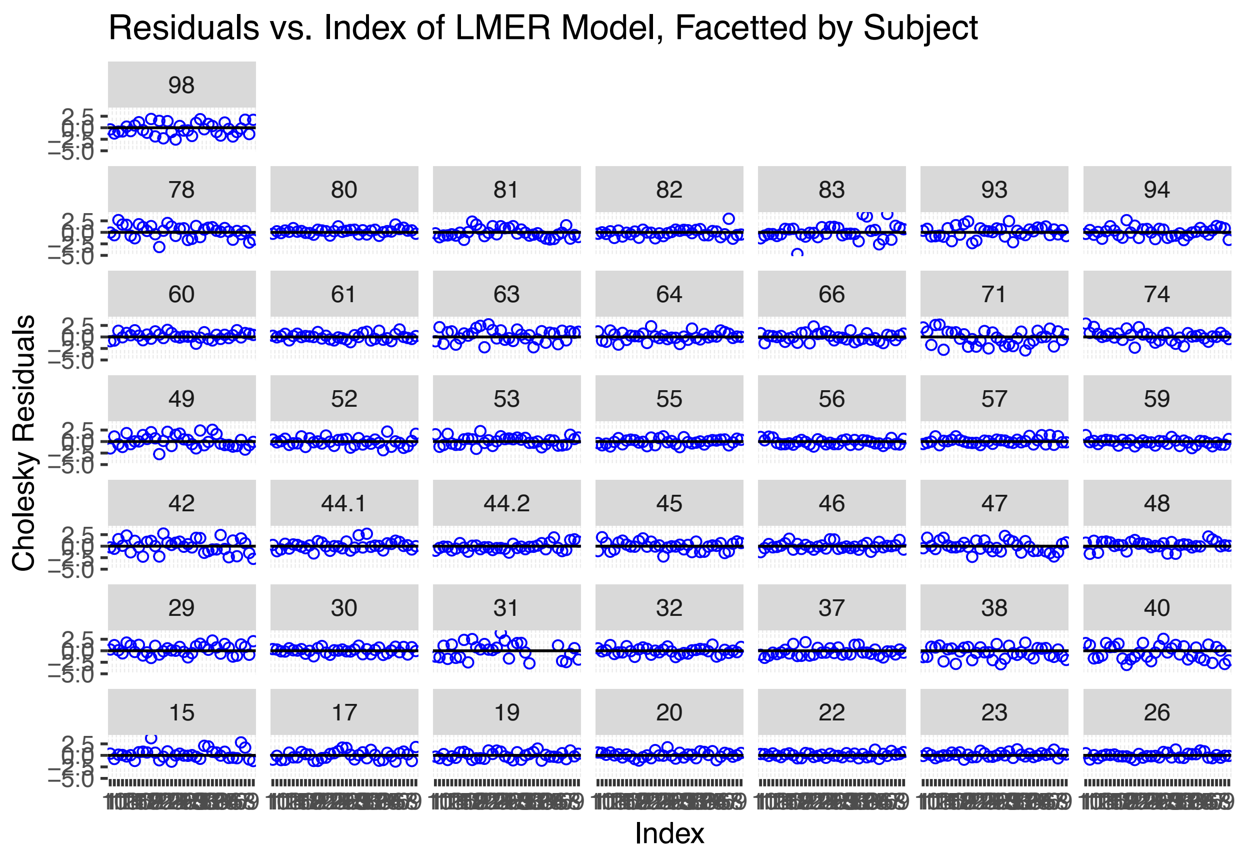
```r
vif(lmer.reduced.1)
```

```
##                  GVIF Df GVIF^(1/(2*Df))
## Instrument   1.371310  2        1.082141
## Harmony      1.371328  3        1.054040
## Voice        1.000030  2        1.000008
## Selfdeclare 49.771155  5        1.478079
## ConsNotes    6.541822  4        1.264627
## KnowRob      3.703573  2        1.387252
## APTheory     2.609583  1        1.615420
## PianoPlay   13.604876  3        1.545073
## GuitarPlay  28.217106  4        1.518148
```

From the VIF values, we now know that our predictors don't suffer as much from concerning collinearity. Additionally, APTheory's and Selfdeclare's relation in determining how classical a stimulus sounds seem more reasonable now compared to before (they're both positive).

We want to ensure that our residuals are relatively homoscedastic and centered around 0, so we plot our residuals below (conditioned on subject).

```
rchol = r.chol(lmer.reduced.1)

sub = (classical$Subject)
index <- sub
for (j in unique(sub)) {
  len <- sum(sub==j)
  index[sub==j] <- 1:len
}

new.data <- data.frame(index,rchol,sub)
names(new.data) <- c("index","rchol","sub")
ggplot(new.data,aes(x=index,y=rchol)) +
  facet_wrap( ~ sub, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0) +
  labs(title="Residuals vs. Index of LMER Model, Facetted by Subject",
       x = "Index", y = "Cholesky Residuals")
```



From the residual plots above, we can see that our marginal residuals are fairly reasonably spread, indicating homoscedasticity, and for the most part, by group, the residuals are roughly centered around 0. As a result, we think that our fixed effects are roughly reasonable.

b.

Now that we have a series of fixed effects that we find reasonable in predicting classical stimuli, we want to re-test the random effects we've obtained. Since we found with backward stepping that both our intercept, instrument, and harmony are all significant random intercepts conditioned on subject, we can begin by adding other random effects and compare their significance with ANOVA tests. We do this with our other variables below:

```
# testing for significance of random intercept

lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + Voice | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# voice as a random effect isn't signifciant at alpha=0.05 level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + Voice | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 64 6211.1 6552.8 -3041.5   6083.1 23.759     15
##                    Pr(>Chisq)
## lmer.reduced.1
## lmer.random.final1     0.0693 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + Selfdeclare | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# Selfdeclare as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + Selfdeclare | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 94 6246.9 6748.8 -3029.5   6058.9 47.903     45
##                    Pr(>Chisq)
```

```
## lmer.reduced.1
## lmer.random.final1     0.3558
```

```
lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# OMSI as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony | Subject)
##                    Df    AIC    BIC  logLik deviance Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 49 6204.8 6466.5 -3053.4   6106.8     0      0
##                    Pr(>Chisq)
## lmer.reduced.1
## lmer.random.final1          1
```

```
lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + ConsNotes | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# ConsNotes as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + ConsNotes | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 83 6240.2 6683.4 -3037.1   6074.2 32.601     34
##                    Pr(>Chisq)
## lmer.reduced.1
## lmer.random.final1     0.5362
```

```
lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + KnowRob | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
```

```r
# KnowRob as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + KnowRob | Subject)
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 64 6211.8 6553.5 -3041.9   6083.8 23.015     15
##                    Pr(>Chisq)
## lmer.reduced.1
## lmer.random.final1    0.08383 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lmer.random.final1 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + APTheory | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# APTheory as a random effect IS signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.reduced.1, lmer.random.final1)
```

```
## Data: classical
## Models:
## lmer.reduced.1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.reduced.1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.reduced.1:     Harmony | Subject)
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + APTheory | Subject)
##                    Df    AIC    BIC  logLik deviance Chisq Chi Df
## lmer.reduced.1     49 6204.8 6466.5 -3053.4   6106.8
## lmer.random.final1 56 6202.0 6501.0 -3045.0   6090.0 16.79      7
##                    Pr(>Chisq)
## lmer.reduced.1
## lmer.random.final1    0.01881 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lmer.random.final2 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + APTheory + PianoPlay | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# PianoPlay as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.random.final1, lmer.random.final2)
```

```
## Data: classical
## Models:
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + APTheory | Subject)
## lmer.random.final2: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final2:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final2:     Harmony + APTheory + PianoPlay | Subject)
##                     Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.final1 56 6202.0 6501.0 -3045.0   6090.0
## lmer.random.final2 83 6242.6 6685.8 -3038.3   6076.6 13.432     27
##                     Pr(>Chisq)
## lmer.random.final1
## lmer.random.final2     0.9862
```

```
lmer.random.final2 = lmer(Classical ~ Instrument + Harmony + Voice + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + APTheory + GuitarPlay | Subject) ,
                        data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# GuitarPlay as a random effect isn't signifciant at alpha=0.05
# level based on AIC returned from ANOVA call
anova(lmer.random.final1, lmer.random.final2)
```

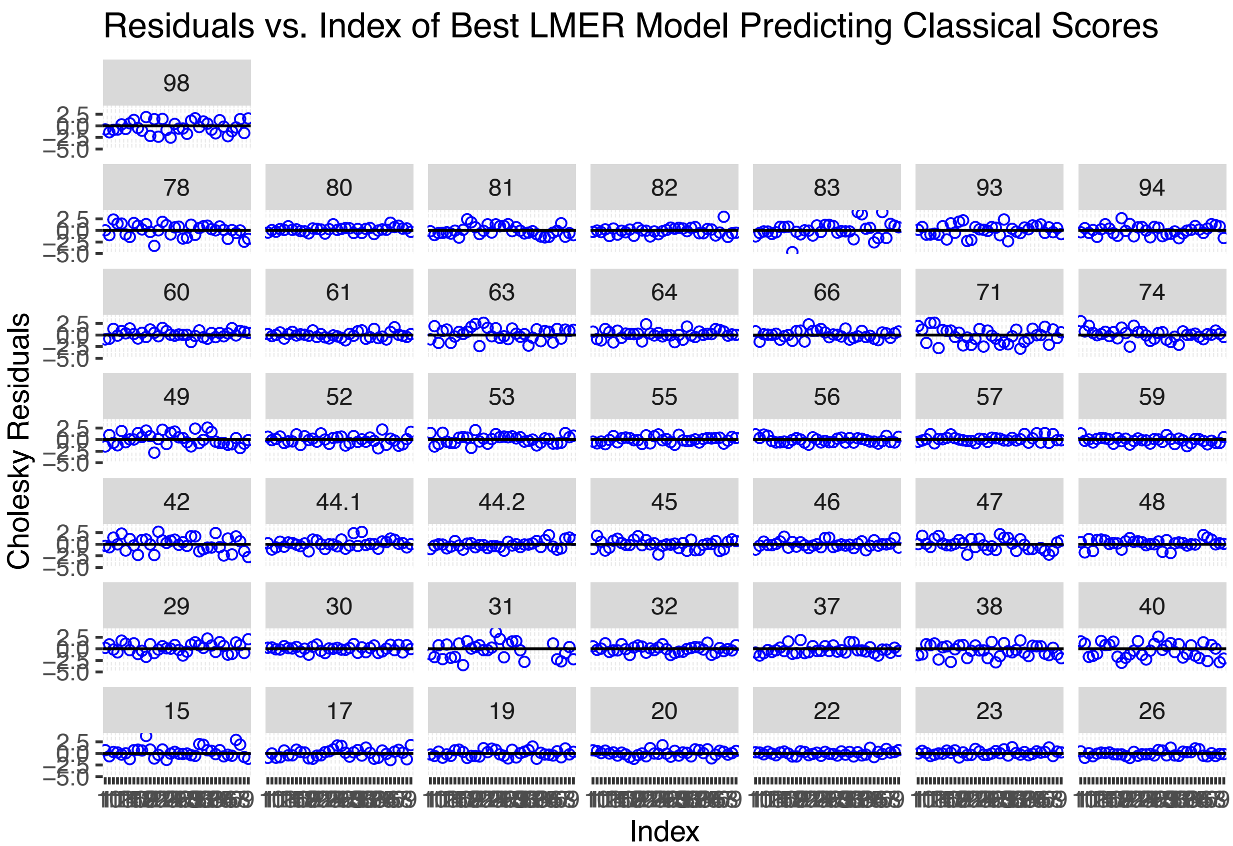
```
## Data: classical
## Models:
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + APTheory | Subject)
## lmer.random.final2: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final2:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final2:     Harmony + APTheory + GuitarPlay | Subject)
##                     Df  AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.final1 56 6202 6501.0 -3045.0     6090
## lmer.random.final2 94 6267 6768.9 -3039.5     6079 11.021     38
##                     Pr(>Chisq)
## lmer.random.final1
## lmer.random.final2          1
```

By testing for random effects of every single predictor that had fixed effects in our model, we concluded that adding the random effect of having taken AP Music theory would significantly improve our model. Thus our final model ends up predicting classical stimulus from the fixed effects of Instrument, Harmony, Voice, Selfdeclare, ConsNotes, KnowRob, APTheory, PianoPlay, GuitarPlay, as well as a random intercept, instrument, harmony, and APTheory.

To ensure that our final model is actually reasonable, we want to ensure that our residuals are homoskedastic and centered arond 0. We plot our residuals below:

```
rchol = r.chol(lmer.random.final1)

sub = (classical$Subject)
index <- sub
for (j in unique(sub)) {
  len <- sum(sub==j)
  index[sub==j] <- 1:len
}
```

```r
new.data <- data.frame(index,rchol,sub)
names(new.data) <- c("index","rchol","sub")
ggplot(new.data,aes(x=index,y=rchol)) +
  facet_wrap( ~ sub, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0) +
  labs(title="Residuals vs. Index of Best LMER Model Predicting Classical Scores",
       x = "Index", y = "Cholesky Residuals")
```



Because our residuals look fairly homoscedastic and centered around 0, we conclude that this model seems like a reasonable linear model to predict classical scores.

We also want to determine if the interaction between harmonic motion and knowledge of the Pachelbel rants are significant. To do this, we fit an additional term Harmony:KnowRob to check if our coefficients change and our model is significantly improved:

```r
summary(lmer.random.final1)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
##     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
##     Harmony + APTheory | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##     6202     6501    -3045     6090     1484
```

```
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5865 -0.5581 -0.0021  0.5478  3.4443
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      0.5002   0.7073
##           Instrumentpiano  1.9450   1.3946   -0.19
##           Instrumentstring 3.6943   1.9220   -0.40  0.62
##           HarmonyI-V-IV    0.1644   0.4055    0.76 -0.53 -0.64
##           HarmonyI-V-VI    1.7957   1.3400    0.39 -0.39 -0.58  0.47
##           HarmonyIV-I-V    0.1928   0.4391    0.79 -0.19 -0.21  0.32  0.42
##           APTheory1        0.9236   0.9610   -0.80 -0.15 -0.14 -0.41 -0.35
##  Residual                  2.4662   1.5704
##
##
##
##
##
##
##
##  -0.76
##
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)       4.230526   0.518799   8.154
## Instrumentpiano   1.654997   0.234331   7.063
## Instrumentstring  3.588412   0.309003  11.613
## HarmonyI-V-IV     0.002776   0.128996   0.022
## HarmonyI-V-VI     0.854767   0.233632   3.659
## HarmonyIV-I-V     0.059098   0.131390   0.450
## Voicepar3rd      -0.401780   0.098071  -4.097
## Voicepar5th      -0.301895   0.098022  -3.080
## Selfdeclare2     -0.602610   0.612188  -0.984
## Selfdeclare3     -0.142438   0.740308  -0.192
## Selfdeclare4     -1.810163   0.807291  -2.242
## Selfdeclare5     -0.587108   1.115901  -0.526
## Selfdeclare6     -2.397679   1.243374  -1.928
## ConsNotes1        0.415736   0.510826   0.814
## ConsNotes3       -0.229078   0.435597  -0.526
## ConsNotes4       -1.837439   0.918250  -2.001
## ConsNotes5       -1.075049   0.445526  -2.413
## KnowRob1         -0.875440   0.546873  -1.601
## KnowRob5          1.055736   0.329116   3.208
## APTheory1         1.549572   0.325020   4.768
## PianoPlay1        0.276967   0.400083   0.692
## PianoPlay4        0.821597   0.370271   2.219
## PianoPlay5        1.857079   0.595823   3.117
## GuitarPlay1      -0.112749   0.625741  -0.180
## GuitarPlay2       1.843633   1.137145   1.621
## GuitarPlay4       1.760648   0.624227   2.821
```

```
## GuitarPlay5      -1.069359   0.604894  -1.768
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
lmer.random.pach = update(lmer.random.final1, . ~ . + Harmony:KnowRob)
anova(lmer.random.final1, lmer.random.pach)
```

```
## Data: classical
## Models:
## lmer.random.final1: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.final1:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.final1:     Harmony + APTheory | Subject)
## lmer.random.pach: Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
## lmer.random.pach:     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
## lmer.random.pach:     Harmony + APTheory | Subject) + Harmony:KnowRob
##                    Df    AIC    BIC  logLik deviance  Chisq Chi Df
## lmer.random.final1 56 6202.0 6501.0 -3045.0   6090.0
## lmer.random.pach   62 6206.6 6537.7 -3041.3   6082.6 7.3951      6
##                    Pr(>Chisq)
## lmer.random.final1
## lmer.random.pach       0.2858
```

```r
summary(lmer.random.pach)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + Selfdeclare + ConsNotes +
##     KnowRob + APTheory + PianoPlay + GuitarPlay + (1 + Instrument +
##     Harmony + APTheory | Subject) + Harmony:KnowRob
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6206.6   6537.7  -3041.3   6082.6     1478
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5503 -0.5682  0.0036  0.5474  3.4682
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      0.5127   0.7160
##           Instrumentpiano  1.9496   1.3963   -0.19
##           Instrumentstring 3.6977   1.9229   -0.39  0.62
##           HarmonyI-V-IV    0.1830   0.4278    0.75 -0.55 -0.70
##           HarmonyI-V-VI    1.4175   1.1906    0.35 -0.40 -0.54  0.55
##           HarmonyIV-I-V    0.1952   0.4418    0.77 -0.24 -0.24  0.36  0.45
##           APTheory1        0.8914   0.9441   -0.81 -0.14 -0.13 -0.42 -0.35
##  Residual                  2.4650   1.5700
##
##
##
##
##
##
```

```
##
##  -0.75
##
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)              4.212540   0.521838   8.072
## Instrumentpiano          1.654520   0.234550   7.054
## Instrumentstring         3.588499   0.309124  11.609
## HarmonyI-V-IV            0.050663   0.148607   0.341
## HarmonyI-V-VI            0.609217   0.240791   2.530
## HarmonyIV-I-V            0.051726   0.151303   0.342
## Voicepar3rd             -0.401893   0.098047  -4.099
## Voicepar5th             -0.301955   0.097998  -3.081
## Selfdeclare2            -0.574167   0.615262  -0.933
## Selfdeclare3            -0.105745   0.744490  -0.142
## Selfdeclare4            -1.781638   0.811223  -2.196
## Selfdeclare5            -0.474227   1.122541  -0.422
## Selfdeclare6            -2.291766   1.249158  -1.835
## ConsNotes1               0.417565   0.513275   0.814
## ConsNotes3              -0.193326   0.439711  -0.440
## ConsNotes4              -1.844614   0.922877  -1.999
## ConsNotes5              -1.055480   0.449909  -2.346
## KnowRob1                -0.957032   0.561219  -1.705
## KnowRob5                 0.943722   0.382209   2.469
## APTheory1                1.579179   0.326975   4.830
## PianoPlay1               0.284364   0.402120   0.707
## PianoPlay4               0.811965   0.371188   2.187
## PianoPlay5               1.864246   0.599035   3.112
## GuitarPlay1             -0.141073   0.628756  -0.224
## GuitarPlay2              1.828259   1.143957   1.598
## GuitarPlay4              1.705528   0.627431   2.718
## GuitarPlay5             -1.164038   0.607500  -1.916
## HarmonyI-V-IV:KnowRob1 -0.008145   0.426774  -0.019
## HarmonyI-V-VI:KnowRob1  0.575636   0.668374   0.861
## HarmonyIV-I-V:KnowRob1  0.163369   0.453226   0.360
## HarmonyI-V-IV:KnowRob5 -0.296418   0.336075  -0.882
## HarmonyI-V-VI:KnowRob5  1.181331   0.514701   2.295
## HarmonyIV-I-V:KnowRob5 -0.049177   0.348302  -0.141
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
sum.f1 = summary(lmer.random.final1)
sum.f2 = summary(lmer.random.pach)
knitr::kable(sum.f1$coefficients)
knitr::kable(sum.f2$coefficients)
```

To determine the actual effect of voice leading in our model, we need to remove the intercept and re-add voice as a predictor. We do this below:

```
lmer.random.final.no.intercept = lmer(Classical ~ Voice - 1 + Instrument + Harmony  + Selfdeclare +
    ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
      (1 + Instrument + Harmony + APTheory | Subject) ,
                        data=classical,
```

```r
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

summary(lmer.random.final.no.intercept)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Voice - 1 + Instrument + Harmony + Selfdeclare +
##     ConsNotes + KnowRob + APTheory + PianoPlay + GuitarPlay +
##     (1 + Instrument + Harmony + APTheory | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##     6202     6501    -3045     6090     1484
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5865 -0.5581 -0.0021  0.5478  3.4443
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      0.5002   0.7073
##           Instrumentpiano  1.9450   1.3946   -0.19
##           Instrumentstring 3.6943   1.9220   -0.40  0.62
##           HarmonyI-V-IV    0.1644   0.4055    0.76 -0.53 -0.64
##           HarmonyI-V-VI    1.7957   1.3400    0.39 -0.39 -0.58  0.47
##           HarmonyIV-I-V    0.1928   0.4391    0.79 -0.19 -0.21  0.32  0.42
##           APTheory1        0.9236   0.9610   -0.80 -0.15 -0.14 -0.41 -0.35
##  Residual                  2.4662   1.5704
##
##
##
##
##
##
##
##  -0.76
##
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                   Estimate Std. Error t value
## Voicecontrary     4.230523   0.518799   8.154
## Voicepar3rd       3.828744   0.518811   7.380
## Voicepar5th       3.928628   0.518814   7.572
## Instrumentpiano   1.654997   0.234331   7.063
## Instrumentstring  3.588412   0.309003  11.613
## HarmonyI-V-IV     0.002776   0.128996   0.022
## HarmonyI-V-VI     0.854767   0.233632   3.659
## HarmonyIV-I-V     0.059098   0.131390   0.450
## Selfdeclare2     -0.602608   0.612188  -0.984
## Selfdeclare3     -0.142434   0.740309  -0.192
## Selfdeclare4     -1.810160   0.807291  -2.242
## Selfdeclare5     -0.587097   1.115902  -0.526
## Selfdeclare6     -2.397674   1.243374  -1.928
```

```
## ConsNotes1        0.415735   0.510826   0.814
## ConsNotes3       -0.229078   0.435597  -0.526
## ConsNotes4       -1.837439   0.918251  -2.001
## ConsNotes5       -1.075049   0.445526  -2.413
## KnowRob1         -0.875441   0.546873  -1.601
## KnowRob5          1.055741   0.329116   3.208
## APTheory1         1.549574   0.325020   4.768
## PianoPlay1        0.276968   0.400083   0.692
## PianoPlay4        0.821597   0.370271   2.219
## PianoPlay5        1.857078   0.595823   3.117
## GuitarPlay1      -0.112749   0.625741  -0.180
## GuitarPlay2       1.843635   1.137145   1.621
## GuitarPlay4       1.760644   0.624227   2.821
## GuitarPlay5      -1.069363   0.604894  -1.768
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

4. Musicians vs. Non-Musicians

To determine whether musicians actually identify classical music differently, we can fit our best classical model, which already takes into account one's self-declaration of being a musician, and add an interaction between these variables and Selfdeclare. Then by backward stepping with a criterion like AIC, we can determine which fixed and random effects are influenced by musicians vs. non-musicians.

One naive split is whether one self-declared themselves as a musician or not. If we split based on whether self-declared store is at least a 3/6, then we get that 713 rows have been evaluated by musicians, and 827 rows have been evaluated by non-musicians.

```
classical = classical %>% mutate(musicians = ifelse(as.numeric(Selfdeclare) >= 3,1,0))
```

Now, let's test adding instrument, harmony, and voice, along with their interaction with musician dichotomization, as fixed and random effects in our model to determine our best model:

Finding signigficant fixed effects with backward stepping of AIC:

```
lm.full.4.1 = lm(Classical ~ (Instrument + Harmony + Voice) * musicians, data=classical)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Classical ~ Instrument + Harmony + Voice + musicians +
##     Instrument:musicians + Harmony:musicians, data = classical)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2531 -1.6399 -0.0787  1.6103  6.3514
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.678340   0.208063  17.679  < 2e-16 ***
## Instrumentpiano            1.960340   0.191634  10.230  < 2e-16 ***
## Instrumentstring           4.143311   0.191634  21.621  < 2e-16 ***
## HarmonyI-V-IV              0.009662   0.221078   0.044 0.965147
## HarmonyI-V-VI              0.267856   0.221347   1.210 0.226421
## HarmonyIV-I-V              0.004831   0.221078   0.022 0.982569
## Voicepar3rd               -0.406302   0.140435  -2.893 0.003868 **
## Voicepar5th               -0.297586   0.140367  -2.120 0.034162 *
## musicians                  0.409510   0.281051   1.457 0.145303
## Instrumentpiano:musicians  -0.651688   0.281981  -2.311 0.020960 *
## Instrumentstring:musicians -1.193311   0.280853  -4.249 2.28e-05 ***
## HarmonyI-V-IV:musicians    -0.018777   0.325142  -0.058 0.953956
## HarmonyI-V-VI:musicians     1.244997   0.325077   3.830 0.000133 ***
## HarmonyIV-I-V:musicians     0.111880   0.324649   0.345 0.730428
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.249 on 1526 degrees of freedom
## Multiple R-squared:  0.3283, Adjusted R-squared:  0.3226
## F-statistic: 57.37 on 13 and 1526 DF,  p-value: < 2.2e-16
```

```
lmer.full.4.1 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
```

```
      (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that the intercept is not significant when it comes to
# adding random effect based on differences in AIC values
anova(lmer.full.4.1, lmer.full.4.2, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 36 6207.5 6399.8 -3067.8   6135.5
## lmer.full.4.2 37 6208.6 6406.2 -3067.3   6134.6 0.9505      1     0.3296
```

```
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (0 + Instrument | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is not significant when it comes to
# adding random effect based on differences in AIC values
anova(lmer.full.4.2, lmer.full.4.1, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Instrument | musicians) + (1 + Harmony +
## lmer.full.4.2:     Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 36 6207.5 6399.8 -3067.8   6135.5
## lmer.full.4.2 42 6218.6 6442.9 -3067.3   6134.6 0.9505      6     0.9874
```

```
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (0 + Harmony | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is significant when it comes to adding
# random effect based on differences in AIC values

anova(lmer.full.4.2, lmer.full.4.1, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Harmony | musicians) + (1 + Harmony +
```

```
## lmer.full.4.2:     Instrument | Subject)
##                Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 36 6207.5 6399.8 -3067.8   6135.5
## lmer.full.4.2 46 6226.6 6472.2 -3067.3   6134.6 0.9505     10     0.9999
```

```
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (0 + Voice | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# we find that voice is not significant when it comes to adding random
# effect based on differences in AIC values

anova(lmer.full.4.2, lmer.full.4.1, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Voice | musicians) + (1 + Harmony +
## lmer.full.4.2:     Instrument | Subject)
##                Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 36 6207.5 6399.8 -3067.8   6135.5
## lmer.full.4.2 42 6219.5 6443.8 -3067.8   6135.5     0      6          1
```

```
music1 = lmer.full.4.1
```

With this first distinction of whether one is a musician, we can see that the interaction terms between musicians/instrument and musicians/harmony were significant random effects, and there were no significant random effects on classical stimulus rating.

Let's decide a second distinction of whether one is a musician by whether one concentrated on the notes with a rating of at least 3/5 and whether one concentrated on the instruments with a rating of at more than 4/5. This yields 864 musicians and 676 non-musicians. We perform the same analysis as above with this new classification of musician:

```
classical = classical %>% mutate(musicians =
                                    ifelse(as.numeric(classical$ConsNotes) >= 3 &
                                    as.numeric(classical$ConsInstr) > 4 ,1,0))

lm.full.4.1 = lm(Classical ~ (Instrument + Harmony + Voice) * musicians, data=classical)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Classical ~ Instrument + Harmony + Voice + musicians,
##     data = classical)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1262 -1.5344 -0.0643  1.6488  6.3597
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)       3.991243   0.176100  22.665  < 2e-16 ***
## Instrumentpiano   1.656264   0.142032  11.661  < 2e-16 ***
## Instrumentstring  3.586555   0.141544  25.339  < 2e-16 ***
## HarmonyI-V-IV     0.002311   0.163787   0.014  0.98875
## HarmonyI-V-VI     0.846430   0.163788   5.168 2.68e-07 ***
## HarmonyIV-I-V     0.056711   0.163574   0.347  0.72887
## Voicepar3rd      -0.407668   0.141889  -2.873  0.00412 **
## Voicepar5th      -0.298024   0.141821  -2.101  0.03577 *
## musicians        -0.215514   0.116690  -1.847  0.06495 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.272 on 1531 degrees of freedom
## Multiple R-squared:  0.312,  Adjusted R-squared:  0.3085
## F-statistic: 86.81 on 8 and 1531 DF,  p-value: < 2.2e-16
```

```
lmer.full.4.1 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians +
      (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians +
      (1 | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# we find that the rahdom intercept is not significant when it comes to adding random effect
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     (1 | musicians) + (1 + Harmony + Instrument | Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 33 6211.6 6387.8 -3072.8   6145.6
## lmer.full.4.2 34 6214.3 6395.9 -3073.2   6146.3     0      1          1
```

```
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians +
      (0 + Instrument | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is not significant when it comes to adding random effect
anova(lmer.full.4.2, lmer.full.4.1)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     (0 + Instrument | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 33 6211.6 6387.8 -3072.8   6145.6
## lmer.full.4.2 39 6224.3 6432.6 -3073.2   6146.3     0      6          1
```

```r
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians +
      (0 + Harmony| musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                   REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is not significant when it comes to adding random effect
anova(lmer.full.4.2, lmer.full.4.1)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     (0 + Harmony | musicians) + (1 + Harmony + Instrument | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 33 6211.6 6387.8 -3072.8   6145.6
## lmer.full.4.2 43 6231.5 6461.1 -3072.7   6145.5 0.1062     10          1
```

```r
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians +
      (0 + Voice| musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                   REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is not significant when it comes to adding random effect
anova(lmer.full.4.2, lmer.full.4.1)
```

```
## Data: classical
## Models:
## lmer.full.4.1: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.1:     (1 + Harmony + Instrument | Subject)
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     (0 + Voice | musicians) + (1 + Harmony + Instrument | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 33 6211.6 6387.8 -3072.8   6145.6
## lmer.full.4.2 39 6223.5 6431.7 -3072.7   6145.5 0.1062      6          1
```

```r
music2 = lmer.full.4.1
```

We find that with this classification of musician, instrument was the only fixed effect that depended on musicians, and no random effects were statistically significant.

If we try a third dichotomization, where someone rated their piano playing or guitar playing at at least 2/5, we get that 784 rows of our dataset were evaluated by musicians, and 756 were evaluated by non-musicians. We can continue to test random effects of our model to see if any variables have random effects that significantly impact classical stimulus rating.

```r
classical = classical %>% mutate(musicians =
                                   ifelse(as.numeric(classical$GuitarPlay) >= 2 |
                                          as.numeric(classical$PianoPlay) >= 2 ,
                                          1,0))

lm.full.4.1 = lm(Classical ~ (Instrument + Harmony + Voice) * musicians, data=classical)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Classical ~ Instrument + Harmony + Voice + musicians +
```

```
##     Harmony:musicians, data = classical)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6758 -1.6270  0.0244  1.5946  6.2772
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               3.5411     0.1991  17.790  < 2e-16 ***
## Instrumentpiano           1.6582     0.1401  11.840  < 2e-16 ***
## Instrumentstring          3.5855     0.1396  25.689  < 2e-16 ***
## HarmonyI-V-IV             0.1640     0.2305   0.712  0.47684
## HarmonyI-V-VI             0.4815     0.2305   2.089  0.03689 *
## HarmonyIV-I-V             0.1481     0.2305   0.643  0.52051
## Voicepar3rd              -0.4085     0.1399  -2.920  0.00355 **
## Voicepar5th              -0.2998     0.1398  -2.144  0.03221 *
## musicians                 0.6468     0.2282   2.835  0.00464 **
## HarmonyI-V-IV:musicians  -0.3164     0.3231  -0.979  0.32753
## HarmonyI-V-VI:musicians   0.7207     0.3231   2.231  0.02584 *
## HarmonyIV-I-V:musicians  -0.1792     0.3227  -0.555  0.57880
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.241 on 1528 degrees of freedom
## Multiple R-squared:  0.3324, Adjusted R-squared:  0.3276
## F-statistic: 69.16 on 11 and 1528 DF,  p-value: < 2.2e-16
```

```r
lmer.full.4.1 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
lmer.full.4.2 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that intercept is significant when it comes to adding
# random effect based on differences in AIC values
anova(lmer.full.4.2, lm.full.4.1, REML=FALSE)
```

```
## Data: classical
## Models:
## lm.full.4.1: Classical ~ (Instrument + Harmony + Voice) * musicians
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lm.full.4.1   17 6874.1 6964.9 -3420.1   6840.1
## lmer.full.4.2 37 6207.0 6404.6 -3066.5   6133.0 707.12     20  < 2.2e-16
##
## lm.full.4.1
## lmer.full.4.2 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
lmer.full.4.3 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 + Instrument | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is not significant when it comes to adding
# random effect based on differences in AIC values
anova(lmer.full.4.2, lmer.full.4.3, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
## lmer.full.4.3: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.3:     Harmony:musicians + (1 + Instrument | musicians) + (1 + Harmony +
## lmer.full.4.3:     Instrument | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.2 37 6207.0 6404.6 -3066.5   6133.0
## lmer.full.4.3 42 6216.6 6440.8 -3066.3   6132.6 0.4236      5     0.9947
```

```r
lmer.full.4.3 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 + Harmony | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is not significant when it comes to adding random
# effect based on differences in AIC values
anova(lmer.full.4.2, lmer.full.4.3, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
## lmer.full.4.3: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.3:     Harmony:musicians + (1 + Harmony | musicians) + (1 + Harmony +
## lmer.full.4.3:     Instrument | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.2 37 6207.0 6404.6 -3066.5   6133.0
## lmer.full.4.3 46 6224.6 6470.2 -3066.3   6132.6 0.4228      9          1
```

```r
lmer.full.4.3 = lmer(Classical ~ Instrument + Harmony + Voice + musicians +
    Instrument:musicians + Harmony:musicians +
      (1 + Voice | musicians) + (1 + Harmony + Instrument | Subject), data=classical,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that voice is not significant when it comes to adding random
# effect based on differences in AIC values
anova(lmer.full.4.2, lmer.full.4.3, REML=FALSE)
```

```
## Data: classical
## Models:
## lmer.full.4.2: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
## lmer.full.4.2:     Subject)
## lmer.full.4.3: Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
## lmer.full.4.3:     Harmony:musicians + (1 + Voice | musicians) + (1 + Harmony +
```

```
## lmer.full.4.3:     Instrument | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.2 37 6207.0 6404.6 -3066.5   6133.0
## lmer.full.4.3 42 6216.6 6440.8 -3066.3   6132.6 0.4232      5     0.9947
```

```
music3 = lmer.full.4.2
```

Here, we can see that this dichotomization yields instrument and harmony that have significant interactions with musician, and the intercept is the only statistically significant random effect.

Below are summaries of our final models based on our three dichotomizations of musician:

```
m1 = summary(music1)
m2 = summary(music2)
m3 = summary(music3)
# full summary of dichotomization 1
m1
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
##     Harmony:musicians + (1 + Harmony + Instrument | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6207.5   6399.8  -3067.8   6135.5     1504
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.4578 -0.5707  0.0046  0.5544  3.5492
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.6443   1.2823
##           HarmonyI-V-IV    0.1141   0.3378    0.70
##           HarmonyI-V-VI    1.3478   1.1610    0.16  0.53
##           HarmonyIV-I-V    0.0839   0.2896    0.20 -0.20  0.42
##           Instrumentpiano  1.8226   1.3501   -0.27 -0.40 -0.30  0.00
##           Instrumentstring 3.3377   1.8269   -0.55 -0.67 -0.51 -0.12  0.59
##  Residual                  2.4986   1.5807
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                 3.677426   0.304761  12.067
## Instrumentpiano             1.961137   0.312072   6.284
## Instrumentstring            4.144108   0.404058  10.256
## HarmonyI-V-IV               0.009662   0.170594   0.057
## HarmonyI-V-VI               0.266794   0.287767   0.927
## HarmonyIV-I-V               0.004831   0.166699   0.029
## Voicepar3rd                -0.401627   0.098712  -4.069
## Voicepar5th                -0.301112   0.098663  -3.052
## musicians                   0.405117   0.439009   0.923
## Instrumentpiano:musicians  -0.660073   0.458102  -1.441
## Instrumentstring:musicians -1.194108   0.592428  -2.016
```

```
## HarmonyI-V-IV:musicians    -0.017502   0.250809  -0.070
## HarmonyI-V-VI:musicians     1.261255   0.422183   2.987
## HarmonyIV-I-V:musicians     0.115109   0.244749   0.470
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
# full summary of dichotomization 2
m2
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
##     (1 + Harmony + Instrument | Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6211.6   6387.8  -3072.8   6145.6     1507
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5459 -0.5707 -0.0010  0.5340  3.5074
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.6684   1.2917
##           HarmonyI-V-IV    0.1091   0.3304    0.71
##           HarmonyI-V-VI    1.7451   1.3210    0.23  0.47
##           HarmonyIV-I-V    0.1062   0.3260    0.18 -0.12  0.43
##           Instrumentpiano  1.9164   1.3843   -0.26 -0.66 -0.40 -0.29
##           Instrumentstring 3.6956   1.9224   -0.56 -0.73 -0.59 -0.24  0.62
##  Residual                  2.4975   1.5803
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                 4.012853   0.331830  12.093
## Instrumentpiano             1.632376   0.339289   4.811
## Instrumentstring            3.519062   0.429562   8.192
## HarmonyI-V-IV               0.001884   0.124566   0.015
## HarmonyI-V-VI               0.853968   0.231453   3.690
## HarmonyIV-I-V               0.058673   0.124144   0.473
## Voicepar3rd                -0.401682   0.098691  -4.070
## Voicepar5th                -0.301505   0.098642  -3.057
## musicians                  -0.263240   0.432660  -0.608
## Instrumentpiano:musicians   0.038727   0.440352   0.088
## Instrumentstring:musicians  0.124372   0.534162   0.233
##
## Correlation of Fixed Effects:
##             (Intr) Instrmntp Instrmnts HI-V-I HI-V-V HIV-I- Vcpr3r Vcpr5t
## Instrumntpn -0.297
## Instrmntstr -0.557  0.564
## HrmnyI-V-IV  0.013 -0.164    -0.202
## HrmnyI-V-VI  0.033 -0.218    -0.350     0.389
## HrmnyIV-I-V -0.114 -0.071    -0.066     0.398  0.375
```

```
## Voicepar3rd -0.149  0.001     0.000     0.000 -0.001  0.001
## Voicepar5th -0.148  0.000     0.000    -0.001 -0.002 -0.001  0.500
## musicians   -0.728  0.218     0.408     0.001  0.000  0.000  0.000  0.000
## Instrmntpn:  0.219 -0.726    -0.367    -0.002 -0.001  0.000 -0.001  0.000
## Instrmntst:  0.427 -0.384    -0.694    -0.001 -0.001  0.000  0.000  0.001
##             muscns Instrmntp:
## Instrumntpn
## Instrmntstr
## HrmnyI-V-IV
## HrmnyI-V-VI
## HrmnyIV-I-V
## Voicepar3rd
## Voicepar5th
## musicians
## Instrmntpn: -0.301
## Instrmntst: -0.587  0.529
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
# full summary of dichotomization 3
m3
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Instrument + Harmony + Voice + musicians + Instrument:musicians +
##     Harmony:musicians + (1 | musicians) + (1 + Harmony + Instrument |
##     Subject)
##    Data: classical
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6207.0   6404.6  -3066.5   6133.0     1503
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6116 -0.5662 -0.0001  0.5474  3.5292
##
## Random effects:
##  Groups    Name             Variance  Std.Dev.  Corr
##  Subject   (Intercept)      1.518e+00 1.232e+00
##            HarmonyI-V-IV    1.348e-01 3.671e-01  0.75
##            HarmonyI-V-VI    1.620e+00 1.273e+00  0.12  0.60
##            HarmonyIV-I-V    1.256e-01 3.544e-01  0.22  0.06  0.51
##            Instrumentpiano  1.919e+00 1.385e+00 -0.28 -0.43 -0.37 -0.12
##            Instrumentstring 3.631e+00 1.905e+00 -0.54 -0.72 -0.58 -0.26
##  musicians (Intercept)      1.215e-14 1.102e-07
##  Residual                   2.489e+00 1.578e+00
##
##
##
##
##
##
##   0.62
##
```

```
##
## Number of obs: 1540, groups:  Subject, 43; musicians, 2
##
## Fixed effects:
##                            Estimate Std. Error t value
## (Intercept)                 3.42500    0.30865  11.097
## Instrumentpiano             1.75794    0.33340   5.273
## Instrumentstring            3.82937    0.43892   8.724
## HarmonyI-V-IV               0.16402    0.18098   0.906
## HarmonyI-V-VI               0.48148    0.32172   1.497
## HarmonyIV-I-V               0.14815    0.17977   0.824
## Voicepar3rd                -0.40191    0.09852  -4.079
## Voicepar5th                -0.30166    0.09847  -3.063
## musicians                   0.86267    0.42422   2.034
## Instrumentpiano:musicians  -0.20403    0.46661  -0.437
## Instrumentstring:musicians -0.47114    0.61367  -0.768
## HarmonyI-V-IV:musicians    -0.31824    0.25357  -1.255
## HarmonyI-V-VI:musicians     0.72862    0.45011   1.619
## HarmonyIV-I-V:musicians    -0.17526    0.25159  -0.697
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
knitr::kable(m1$coefficients)
knitr::kable(m2$coefficients)
knitr::kable(m3$coefficients)
```

As a result, from this small series of tests, we conclude that while in people who self-identify as musicians may be influenced by things that do not influence non-musicians in some cases, our results are sensitive to where we choose to dichotomize.

5. Classical vs. Popular

We want to replicate our findings, but instead create a model that will predict popular music stimulus ratings.

a.

First, we want to test the influence of Instrument, Harmony & Voice on Popular music reatings. To do this, we can begin by testing fixed effects of each variable, and then we can begin to add in possible random effects afterward.

We begin by finding which fixed effects should be added to our model. We do this by fitting popular ratings against all interactions of instrument, harmony, and voice, before reducing the number of predictors with a step function and AIC as our criterion.

```
pop.lm.full.1 = lm(Popular ~ Instrument * Harmony * Voice - 1, data=popular)
pop.lm.reduced.1 = stepAIC(pop.lm.full.1, trace=FALSE)
```

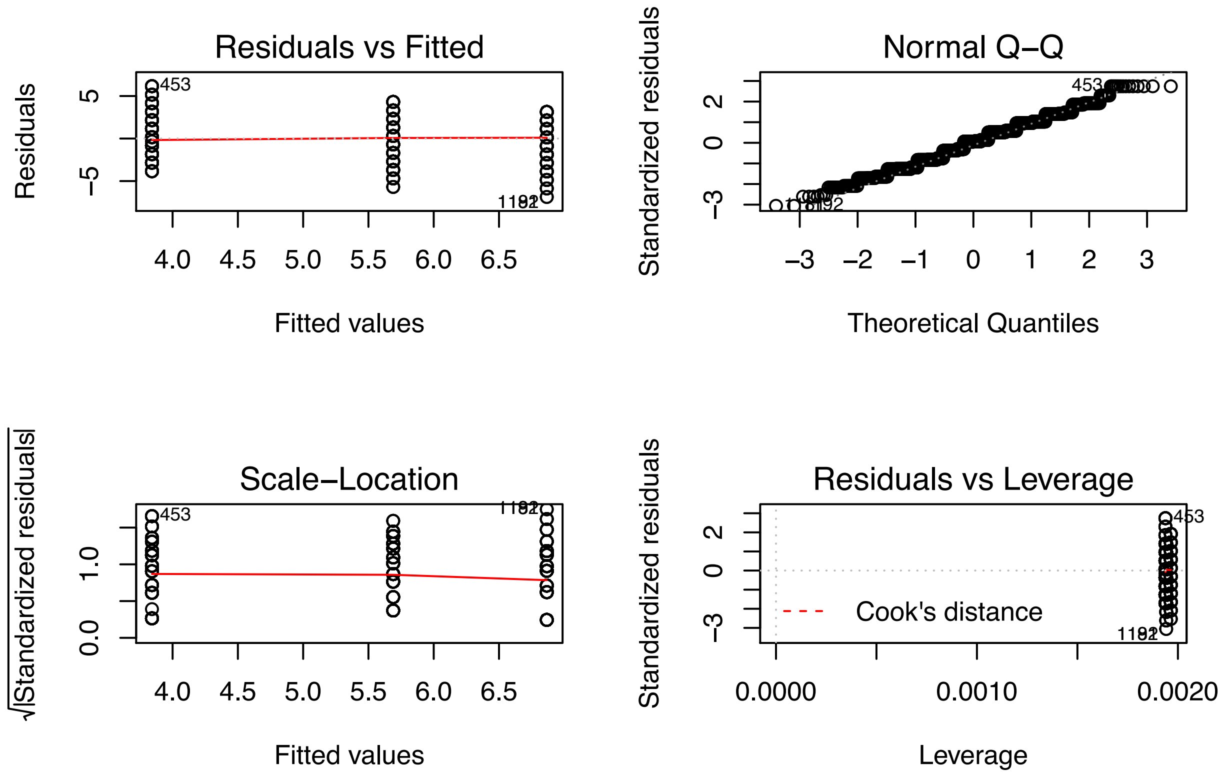
```
x2 = summary(pop.lm.reduced.1)

knitr::kable(x2$coefficients)
```

	Estimate	Std. Error	t value	Pr(>|t|)
Instrumentguitar	6.866019	0.0990705	69.30436	0
Instrumentpiano	5.689587	0.0996527	57.09415	0
Instrumentstring	3.842054	0.0989745	38.81864	0

From our initial addition of fixed effects to a linear model, we find that only instrument is significant in predicting popular scores from the summary above.

To analyze the fit of our basic model predicting popular song scores from instrument, we plot diagnostic plots of this basic model below:

```
par(mfrow=c(2,2))
plot(pop.lm.reduced.1)
```



While our data is categorical, we can't infer too much from our diagnostic plots. However, the residuals vs. fitted lines show a regression curve that is roughly horizontal and around 0 with few outliers. Additionally, our normal-QQ plot seems to roughly follow a linear trend and our scale-location plots show a regression line that is roughly horizontal. We don't see any points with high Cook's distance in our model, either.

Next, we would like to determine any random effects that might influence popular scores, as conditioned on subject, to determine whether any of these effects may vary across subject. We do this by incrementally adding random effects, and determining via an ANOVA test as to whether they are significant.

Starting with testing the intercept, we will incrementally test random effects below:

```
# test significance of random intercept: we find that this is significant via ANOVA
pop.lmer.1 = lmer(Popular ~ Instrument + (1 | Subject), data=popular,
                  REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(pop.lmer.1, pop.lm.reduced.1)
```

```
## Data: popular
## Models:
## pop.lm.reduced.1: Popular ~ Instrument - 1
## pop.lmer.1: Popular ~ Instrument + (1 | Subject)
##                  Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## pop.lm.reduced.1  4 6870.6 6892.0 -3431.3   6862.6
## pop.lmer.1        5 6479.4 6506.1 -3234.7   6469.4 393.2      1  < 2.2e-16
##
## pop.lm.reduced.1
## pop.lmer.1       ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# test significance of instrument as random effect: we find that this
# is significant via difference in AIC
```

```r
pop.lmer.2 = lmer(Popular ~ Instrument + (1 + Instrument| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(pop.lmer.1, pop.lmer.2)
```

```
## Data: popular
## Models:
## pop.lmer.1: Popular ~ Instrument + (1 | Subject)
## pop.lmer.2: Popular ~ Instrument + (1 + Instrument | Subject)
##            Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## pop.lmer.1  5 6479.4 6506.1 -3234.7   6469.4
## pop.lmer.2 10 6327.4 6380.8 -3153.7   6307.4 162.05      5  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# test significance of voice as random effect: we find that this is not
# significant via difference in AIC
pop.lmer.3 = lmer(Popular ~ Instrument + Voice +
                    (1 + Instrument + Voice | Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
anova(pop.lmer.2, pop.lmer.3)
```

```
## Data: popular
## Models:
## pop.lmer.2: Popular ~ Instrument + (1 + Instrument | Subject)
## pop.lmer.3: Popular ~ Instrument + Voice + (1 + Instrument + Voice | Subject)
##            Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## pop.lmer.2 10 6327.4 6380.8 -3153.7   6307.4
## pop.lmer.3 21 6340.5 6452.6 -3149.2   6298.5 8.8674     11     0.6341
```

In summary, we find that our best model predicts popular scores from the fixed effect of instrument and voice, as well as the random effects of an intercept, instrument, and voice. When only considering instrument, harmony and voice, we get the following model, seen in the summary below:

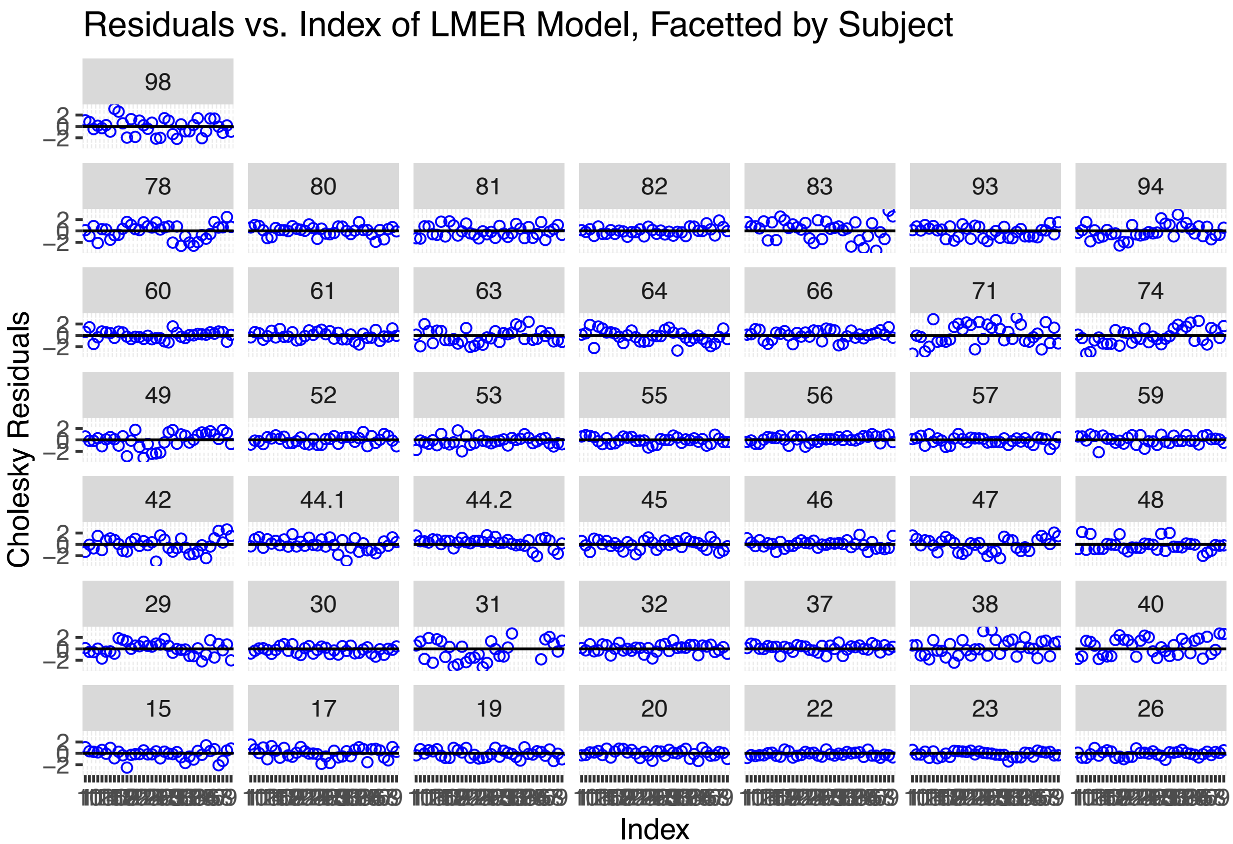
```r
summary(pop.lmer.3)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Popular ~ Instrument + Voice + (1 + Instrument + Voice | Subject)
##    Data: popular
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6340.5   6452.6  -3149.3   6298.5     1519
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4594 -0.5782  0.0254  0.6070  2.9113
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.415490 1.18974
##           Instrumentpiano  1.617753 1.27191  -0.21
##           Instrumentstring 2.556242 1.59883  -0.35  0.72
##           Voicepar3rd      0.036140 0.19010  -0.91  0.22  0.11
##           Voicepar5th      0.005832 0.07637  -0.07 -0.26  0.38 -0.13
##  Residual                  2.984337 1.72752
## Number of obs: 1540, groups:  Subject, 43
```

```
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)        6.7294     0.2064  32.601
## Instrumentpiano   -1.1667     0.2221  -5.253
## Instrumentstring  -3.0234     0.2665 -11.344
## Voicepar3rd        0.1768     0.1117   1.583
## Voicepar5th        0.2314     0.1084   2.134
##
## Correlation of Fixed Effects:
##             (Intr) Instrmntp Instrmnts Vcpr3r
## Instrumntpn -0.284
## Instrmntstr -0.389  0.673
## Voicepar3rd -0.461  0.051     0.026
## Voicepar5th -0.267 -0.024     0.038     0.477
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

To test the residuals of this model, we plot Cholesky residuals against fitted values, facetted by subject below:

```r
rchol = r.chol(pop.lmer.3)

sub = (popular$Subject)
index <- sub
for (j in unique(sub)) {
  len <- sum(sub==j)
  index[sub==j] <- 1:len
}

new.data <- data.frame(index,rchol,sub)
names(new.data) <- c("index","rchol","sub")
ggplot(new.data,aes(x=index,y=rchol)) +
  facet_wrap( ~ sub, as.table=F) +
  geom_point(pch=1,color="Blue") +
  geom_hline(yintercept=0) +
  labs(title="Residuals vs. Index of LMER Model, Facetted by Subject",
       x = "Index", y = "Cholesky Residuals")
```



We can see that our residuals look fairly homoscedastic and centered around 0. This indicates that our model seems reasonable.

b.

Next, we'd like to add ohter fixed effects on top of instrument. We can do this in an automated manner by first adding all the other predictors, and then stepping backward with a criterion like AIC. Afterward, we can reduce the number of predictors if necessary depending on whether there exists multicollinearity in our model. We do this below:

```
pop.full.1 = lmer(Popular ~ Subject + Harmony + Instrument + Voice + Selfdeclare +
                    OMSI + X16.minus.17 + ConsInstr + ConsNotes + Instr.minus.Notes +
                    PachListen + ClsListen + KnowRob + KnowAxis + X1990s2000s +
                    X1990s2000s.minus.1960s1970s + CollegeMusic + NoClass +
                    APTheory + Composing + PianoPlay + GuitarPlay + (1 | Subject),
                        data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
```

```r
# going to delete all extraneous pages associated with trace output I can't hide
pop.reduced.1 = fitLMER.fnc(pop.full.1, method="AIC",
                            ran.effects=c("(0 + Instrument | Subject)",
                                          "(0 + Voice | Subject)"),
                            set.REML.FALSE = TRUE)
```

```r
# refit model without multicollinear terms (commented out long summary and print
# final model summary below)
pop.reduced.1 = update(pop.reduced.1, . ~ . - Instr.minus.Notes - OMSI - APTheory - X16.minus.17)
vif(pop.reduced.1)
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## Instrument  1.013545e+00  2        1.003369
## Voice       1.013572e+00  2        1.003376
## Selfdeclare 7.287245e+02  5        1.933109
## ConsInstr   5.734550e+05 11        1.827048
## ConsNotes   4.370827e+02  4        2.138309
## PachListen  1.613246e+01  2        2.004127
## ClsListen   6.094326e+02  4        2.229029
## KnowAxis    2.628111e+01  2        2.264180
## X1990s2000s 1.714004e+03  4        2.536597
## Composing   2.547661e+02  4        1.998793
```

```r
# summary(pop.reduced.1)
```

From this process, we've reduced our fixed effects on popular scores to Instrument, Selfdeclare, ConsInstr, ConsNotes, PachListen, ClsListen, KnowAxis, X1990s2000s, and Composing

We'd like to re-examine some random effects to determine whether any may be significant given our new list of fixed effects.

After trying every single variable in our dataset that we haven't already excluded in Parts 1 and 2, we found that there weren't any random effects conditioned on subject that were worthwhile to add (given the most staitstically significant addition of random effect was X1990s2000s, with a p-value of 0.6824 when running an ANOVA test for the addition of a single random effect.)

Thus, our final model predicts popular song scores from fixed effects of Instrument, Selfdeclare, ConsInstr, ConsNotes, PachListen, ClsListen, KnowAxis, X1990s2000s, Composing, as well as random effects of instrument and voice. The equation of

$Popular_i = \alpha_{0j[i]} + \alpha_{1j[i]} Instrument_i + \alpha_{2j[i]} Voice_i + \beta_{3i} Selfdeclare_i + \beta_{4i} ConsInstr_i + \beta_{5i} ConsNotes_i + \beta_{6i} PachListen_i + \beta_{7i} ClsListen_i + \beta_{8i} KnowAxis_i + \beta_{9i} X1990s2000s_i + \beta_{10i} Composing_i + \epsilon_i; \epsilon_i \sim N(0, \sigma^2)$,

where $\sigma^2 = 2.93849$

In short, to interpret: - fixed effects (β_{ij}'s): Holding all else constant, if predictor i's fixed effect lies in group j rather than the base level assumed by our intercept term, then in expectation our popular stimulus score increases by β_{ij} - random effects (η_{ij}'s): Holding all else constant, if predictor i's random effect lies in group j rather than the base level assumed by our intercept term, then in expectation our popular stimulus score increases by an iid draw from $N(0, \tau_j^2)$, where τ_j^2 is the variance associated with a random effect within the predictor's group j, conditioned on subject.

The summary of our model can be seen below, where values of random effects variances (ie. τ^2 terms associated with random iid draws), as well as fixed effect estimates (β terms) can be found below:

```r
pop_final = summary(pop.reduced.1)
pop_final
```

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

```
## Formula:
## Popular ~ Instrument + Voice + Selfdeclare + ConsInstr + ConsNotes +
##     PachListen + ClsListen + KnowAxis + X1990s2000s + Composing +
##     (1 + Instrument + Voice | Subject)
##    Data: popular
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   6299.5   6603.8  -3092.7   6185.5     1483
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4614 -0.5786  0.0270  0.6214  2.9639
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      0.26405  0.5139
##           Instrumentpiano  1.74078  1.3194   -0.72
##           Instrumentstring 2.59926  1.6122   -0.99  0.72
##           Voicepar3rd      0.04534  0.2129   -0.31  0.32  0.15
##           Voicepar5th      0.03625  0.1904   -0.39  0.16  0.27  0.74
##  Residual                  2.93715  1.7138
## Number of obs: 1540, groups:  Subject, 43
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)        7.5673     1.2126   6.241
## Instrumentpiano   -1.1662     0.2280  -5.114
## Instrumentstring  -3.0240     0.2680 -11.282
## Voicepar3rd        0.1765     0.1118   1.578
## Voicepar5th        0.2310     0.1108   2.084
## Selfdeclare2      -0.3781     0.3575  -1.057
## Selfdeclare3      -1.7419     0.5307  -3.282
## Selfdeclare4      -0.1749     0.3758  -0.465
## Selfdeclare5       0.1057     0.6700   0.158
## Selfdeclare6      -1.9978     1.0264  -1.946
## ConsInstr0.67     -4.9619     1.3644  -3.637
## ConsInstr1         0.4007     1.0178   0.394
## ConsInstr1.67      2.6657     1.5751   1.692
## ConsInstr2.33      1.9002     1.0551   1.801
## ConsInstr2.67      0.1437     1.0663   0.135
## ConsInstr3        -0.1358     1.3152  -0.103
## ConsInstr3.33      4.6356     1.3944   3.324
## ConsInstr3.67      0.3362     1.3057   0.258
## ConsInstr4        -0.3189     1.2672  -0.252
## ConsInstr4.33      1.6829     1.3726   1.226
## ConsInstr5         1.9356     1.4850   1.303
## ConsNotes1         1.1200     0.6246   1.793
## ConsNotes3        -0.3681     0.4229  -0.871
## ConsNotes4         1.9553     0.7968   2.454
## ConsNotes5         0.4181     0.5702   0.733
## PachListen3       -1.7315     0.7873  -2.199
## PachListen5       -2.7274     0.4537  -6.011
## ClsListen1         0.9383     0.4410   2.128
```

```
## ClsListen3         0.5776     0.4530   1.275
## ClsListen4         0.7597     0.8794   0.864
## ClsListen5        -1.3295     0.5665  -2.347
## KnowAxis1          3.4702     0.7814   4.441
## KnowAxis5          0.0833     0.3288   0.253
## X1990s2000s2       1.6436     0.6004   2.738
## X1990s2000s3       1.1602     0.4874   2.380
## X1990s2000s4       1.5063     1.0698   1.408
## X1990s2000s5       0.4708     0.4737   0.994
## Composing1        -0.5520     0.2870  -1.923
## Composing2         0.8179     0.3989   2.050
## Composing3        -0.3188     0.3611  -0.883
## Composing4        -1.5005     0.5130  -2.925
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```
# knitr::kable(pop_final$coefficients)
```

c.

Finally, we would like to determine if our dichotomization of subjects into musicians vs. non-musicians show that certain predictors affect musicians and don't affect non-musicians in predicting popular stimuli scores. Additionally, we would like to find out if these predictors that affect musicians are sensitive to how we create our dichotomization, similar to Section 4.

To answer these questions, we can formulate multiple different dichotomizations of musicians.

One first naive split is whether one self-declared themselves as a musician or not. If we split based on whether self-declared store is at least a 3/6, then we get that 713 rows have been evaluated by musicians, and 827 rows have been evaluated by non-musicians.

```
popular = popular %>% mutate(musicians = ifelse(as.numeric(Selfdeclare) >= 3,1,0))
```

Now, let's test adding instrument, harmony, and voice as fixed and random effects in our model to determine our best model:

```
lm.full.4.1 = lm(Popular ~ (Instrument + Harmony + Voice) * musicians, data=popular)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
##     Harmony:musicians, data = popular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.0397 -1.6577  0.0798  1.5173  6.4954
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 7.040e+00  1.900e-01  37.048  < 2e-16 ***
## Instrumentpiano            -1.396e+00  1.901e-01  -7.341 3.44e-13 ***
## Instrumentstring           -3.535e+00  1.901e-01 -18.594  < 2e-16 ***
## HarmonyI-V-IV               4.460e-15  2.193e-01   0.000   1.0000
## HarmonyI-V-VI               1.337e-01  2.196e-01   0.609   0.5427
## HarmonyIV-I-V              -2.174e-01  2.193e-01  -0.991   0.3218
## musicians                  -9.705e-02  2.788e-01  -0.348   0.7278
## Instrumentpiano:musicians   4.722e-01  2.797e-01   1.688   0.0916 .
## Instrumentstring:musicians  1.098e+00  2.786e-01   3.939 8.55e-05 ***
## HarmonyI-V-IV:musicians    -2.238e-02  3.226e-01  -0.069   0.9447
## HarmonyI-V-VI:musicians    -8.302e-01  3.225e-01  -2.574   0.0101 *
## HarmonyIV-I-V:musicians    -6.752e-02  3.221e-01  -0.210   0.8340
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.231 on 1528 degrees of freedom
## Multiple R-squared:  0.2516, Adjusted R-squared:  0.2462
## F-statistic:  46.7 on 11 and 1528 DF,  p-value: < 2.2e-16
```

```
lmer.full.4.1 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
    Harmony:musicians +
                    + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
    Harmony:musicians +
                    (1 | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# we find that the intercept is not significant when it comes to adding random effect
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (1 | musicians) + (1 + Voice | Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6464.5 6565.9 -3213.2   6426.5
## lmer.full.4.2 20 6466.5 6573.3 -3213.2   6426.5     0      1          1
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
    Harmony:musicians +
                    (0 + Instrument | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is somewhat significant when it comes to adding random effect
# p=0.051
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Instrument | musicians) + (1 + Voice |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6464.5 6565.9 -3213.2   6426.5
## lmer.full.4.2 25 6476.5 6610.0 -3213.2   6426.5     0      6          1
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
    Harmony:musicians +
                    (0 + Voice | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is not significant when it comes to adding random effect

anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Voice | musicians) + (1 + Voice |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6464.5 6565.9 -3213.2   6426.5
## lmer.full.4.2 25 6475.8 6609.3 -3212.9   6425.8 0.656      6     0.9954
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
    Harmony:musicians +
                    (0 + Harmony | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that voice is not significant when it comes to adding random effect

anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + (0 + Harmony | musicians) + (1 + Voice |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6464.5 6565.9 -3213.2   6426.5
## lmer.full.4.2 29 6484.5 6639.3 -3213.2   6426.5     0     10          1
```

From this initial dichotomization, we find that our intercept and harmony have significant random effects when conditioned on whether one is a musician.

Let's decide a second distinction of whether one is a musician by whether one concentrated on the notes with a rating of at least 3/5 and whether one concentrated on the instruments with a rating of at more than 4/5. This yields 864 musicians and 676 non-musicians. We perform the same analysis as above with this new classification of musician (in this case, 864 data points are classified by musicians, while 676 rows are classified by non-musicians):

```
popular = popular %>% mutate(musicians =
                                  ifelse(as.numeric(ConsNotes) >= 3 &
                                  as.numeric(ConsInstr) > 4 ,1,0))

lm.full.4.1 = lm(Popular ~ (Instrument + Harmony + Voice) * musicians, data=popular)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
##     Harmony:musicians, data = popular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8847 -1.6765  0.1383  1.4717  6.1713
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   6.85771    0.21054  32.572  < 2e-16 ***
## Instrumentpiano              -1.09504    0.21194  -5.167  2.7e-07 ***
## Instrumentstring             -2.71034    0.21029 -12.889  < 2e-16 ***
## HarmonyI-V-IV                 0.02694    0.24399   0.110   0.9121
## HarmonyI-V-VI                 0.09893    0.24398   0.405   0.6852
## HarmonyIV-I-V                -0.32941    0.24326  -1.354   0.1759
## musicians                     0.24356    0.28153   0.865   0.3871
## Instrumentpiano:musicians    -0.14455    0.28258  -0.512   0.6091
## Instrumentstring:musicians   -0.56223    0.28134  -1.998   0.0458 *
## HarmonyI-V-IV:musicians      -0.06860    0.32574  -0.211   0.8332
## HarmonyI-V-VI:musicians      -0.62439    0.32573  -1.917   0.0554 .
## HarmonyIV-I-V:musicians       0.14423    0.32519   0.444   0.6575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.243 on 1528 degrees of freedom
## Multiple R-squared:  0.2439, Adjusted R-squared:  0.2384
## F-statistic:  44.8 on 11 and 1528 DF,  p-value: < 2.2e-16
```

```
lmer.full.4.1 = lmer(Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
                    + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
                    + (1 | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

# we find that the intercept is not significant when it comes to adding random effect
```

```r
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.1:     +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.2:     +(1 | musicians) + (1 + Voice | Subject)
##               Df  AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 17 6486 6576.7  -3226     6452
## lmer.full.4.2 18 6488 6584.1  -3226     6452     0      1          1
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
                    + (0 + Instrument | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is somewhat significant when it comes to adding random effect
# p=0.051
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.1:     +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.2:     +(0 + Instrument | musicians) + (1 + Voice | Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 17 6486.0 6576.7 -3226.0   6452.0
## lmer.full.4.2 23 6496.1 6618.9 -3225.1   6450.1 1.8338      6     0.9343
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
                    + (0 + Voice | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that voice is not significant when it comes to adding random effect

anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.1:     +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.2:     +(0 + Voice | musicians) + (1 + Voice | Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 17 6486.0 6576.7 -3226.0   6452.0
## lmer.full.4.2 23 6497.2 6620.0 -3225.6   6451.2 0.775      6     0.9927
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
                    + (0 + Harmony | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is not significant when it comes to adding random effect

anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
```

```
## lmer.full.4.1:     +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Harmony:musicians +
## lmer.full.4.2:     +(0 + Harmony | musicians) + (1 + Voice | Subject)
##               Df  AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 17 6486 6576.7  -3226     6452
## lmer.full.4.2 27 6506 6650.1  -3226     6452     0     10          1
```

We find that with this classification of musician, none of the random effects were statistically significant.

If we try a third dichotomization, where someone rated their piano playing or guitar playing at at least 2/5, we get that 784 rows of our dataset were evaluated by musicians, and 756 were evaluated by non-musicians. We can continue to test random effects of our model to see if any variables have random effects that significantly impact popular stimulus rating.

```
popular = popular %>% mutate(musicians =
                                   ifelse(as.numeric(popular$GuitarPlay) >= 2 |
                                          as.numeric(popular$PianoPlay) >= 2 ,
                                          1,0))

lm.full.4.1 = lm(Popular ~ (Instrument + Harmony + Voice) * musicians, data=popular)

lm.reduced.4.1 = stepAIC(lm.full.4.1, trace=FALSE)
summary(lm.reduced.4.1)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
##     Harmony:musicians, data = popular)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9577 -1.7235  0.0423  1.4463  6.7637
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                6.95767    0.19912  34.942  < 2e-16 ***
## Instrumentpiano           -0.86905    0.19912  -4.364 1.36e-05 ***
## Instrumentstring          -2.81349    0.19912 -14.129  < 2e-16 ***
## HarmonyI-V-IV             -0.20635    0.22993  -0.897   0.3696
## HarmonyI-V-VI              0.07407    0.22993   0.322   0.7474
## HarmonyIV-I-V             -0.36508    0.22993  -1.588   0.1125
## musicians                  0.07075    0.27866   0.254   0.7996
## Instrumentpiano:musicians  -0.60570    0.27944  -2.168   0.0303 *
## Instrumentstring:musicians -0.40984    0.27851  -1.472   0.1414
## HarmonyI-V-IV:musicians     0.38126    0.32225   1.183   0.2370
## HarmonyI-V-VI:musicians    -0.64288    0.32225  -1.995   0.0462 *
## HarmonyIV-I-V:musicians     0.22802    0.32185   0.708   0.4788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.235 on 1528 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.2436
## F-statistic: 46.06 on 11 and 1528 DF,  p-value: < 2.2e-16
```

```
lmer.full.4.1 = lmer(Popular ~ Instrument + Harmony + musicians +
                       Instrument:musicians + Harmony:musicians +
                    + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))

lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians + Instrument:musicians + Harmony:musicia
                    + (1 | musicians) + (1 + Voice| Subject), data=popular,
                    REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
```

```r
# we find that the intercept is significant when it comes to adding random effect
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + +(1 | musicians) + (1 + Voice | Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6474.7 6576.1 -3218.3   6436.7
## lmer.full.4.2 20 6476.7 6583.5 -3218.3   6436.7     0      1          1
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians +
                       Instrument:musicians + Harmony:musicians +
                     + (0 + Instrument | musicians) + (1 + Voice| Subject), data=popular,
                     REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that instrument is somewhat significant when it comes to adding random effect
# p=0.051
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + +(0 + Instrument | musicians) + (1 +
## lmer.full.4.2:     Voice | Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6474.7 6576.1 -3218.3   6436.7
## lmer.full.4.2 25 6486.7 6620.2 -3218.3   6436.7     0      6          1
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians +
                       Instrument:musicians + Harmony:musicians +
                     + (0 + Voice | musicians) + (1 + Voice| Subject), data=popular,
                     REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that voice is not significant when it comes to adding random effect

anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + +(0 + Voice | musicians) + (1 + Voice |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6474.7 6576.1 -3218.3   6436.7
## lmer.full.4.2 25 6485.8 6619.3 -3217.9   6435.8 0.8475      6     0.9907
```

```r
lmer.full.4.2 = lmer(Popular ~ Instrument + Harmony + musicians +
                       Instrument:musicians + Harmony:musicians +
                     + (0 + Harmony | musicians) + (1 + Voice| Subject), data=popular,
                     REML=FALSE, control=lmerControl(optimizer = 'bobyqa'))
# we find that harmony is not significant when it comes to adding random effect
```

```r
anova(lmer.full.4.1, lmer.full.4.2)
```

```
## Data: popular
## Models:
## lmer.full.4.1: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.1:     Harmony:musicians + +(1 + Voice | Subject)
## lmer.full.4.2: Popular ~ Instrument + Harmony + musicians + Instrument:musicians +
## lmer.full.4.2:     Harmony:musicians + +(0 + Harmony | musicians) + (1 + Voice |
## lmer.full.4.2:     Subject)
##               Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## lmer.full.4.1 19 6474.7 6576.1 -3218.3   6436.7
## lmer.full.4.2 29 6494.7 6649.5 -3218.3   6436.7     0     10          1
```

Here, we can see that this dichotomization of whether one is a musician creates random effects in which the intercept and harmony are statistically significant in contributing to a prediction of popular stimulus rating.

As a result, from this small series of tests, we are inclined to believe that there are in fact random effects conditioned on musician's status in which musicians are influenced by predictors that non-musicians aren't influenced by (ie. harmony, intercept). However, compared to our classical scores model, our dichotomizations for popular scores seem less sensitive to boundaries in which I dichotomized musicians.