Multilevel Analysis on How to Distinguish Classical and Popular Music

Long Jiang
longj1@andrew.cmu.edu

8 December 2019

Abstract

We build multilevel models to decide which variables are influential on listeners' identification of music as classical or popular. We examine data collected by a student from the University of Pittsburgh from an experiment desinged by Ivan Jimenez in 2012. We have two independent mutlilevel models for classical and popular ratings, and we believe they enable us to find out influential predictor. In the future research, by includig more observations and predictors may help improve the current models.

1 Introduction

In 2012, Ivan Jimenez, a composer and musicologist conducted an experiment on measuring the influence of instrument, harmonic motion, and voice leading on listeners' identification of music as "classical" or "popular". In this paper, we are going to expand this experiment by including more predictors and to see which variabales are influential on distinguishing classical music and popular music.

We achieve the above goal using statistical methods, especially multilevel analysis. Specifically, we will answer the following question:

- What experimental factor, or combination of factors, has the strongest influence on ratings?
- Are there differences in the way that musicians and non-musicians identify classical music?
- Are there differences in the things that drive classical, vs. popular, ratings?

2 Methods

By summarizing the data set, we find many variables contain missing values, and we split missing values into three cases. For dependent variables, classical and Popular, we deleted rows with missing values in either classical or popular since we thought imputing dependent variables decreases our model validity; for variables X1stInstr and X2ndInstr, which contains over 50% of missing value among total observations, we deleted thses two variable; for other varibles containing missing values, we impute the data with their modes since mode imputation works well with both numeric and factor variables.

To build our final model, we first use backward stepwise AIC to select predictor that are potentially influential on classical and popular ratings. Than, we fit a standard repeated measure model with random intercept and predictors we select before. After checking the validty of the standard repeated measure model with exactLRt function, we manully remove predictors that are not statistically significant at the significance level of 5%. Finally, we fit multilevel models on remaining predictors and random effect on intercept, harmony, instrument, and voice.

3 Results

The data set was collected by Vincent Rossi, a student of the University of Pittsburgh, from an experiment where 70 participant identitfied 36 music stimuli as either classical or popular. We have two dependent variables, classical and popular. Classical represents how classical does the music sound (1 to 10, 1 = not at all, 10 = very classical sounding), and popular represents how popular does the music sound (1 to 10, 1 = not at all, 10 = very popular sounding). The data set includes 26 varibles and both numeric and categorical, which are listed below. We converted all variables other than classical, popular, and OMSI to factors, and encoded Selfdclare to dummy variable, where selfdeclare with values greater than 2 were converted to 1 and considered musicians, and selfdeclare with values less than or equal to 2 were converted to 0 and considered non-musicians.

Classical = How classical does the music sound (1 to 10, 1 = not at all, 10 = very classical sounding)
Popular = How popular does the music sound (1 to 10, 1 = not at all, 10 = very popular sounding)
Subject = Unique subject ID
Harmony = Harmonic Motion: I-V-vi, I-VI-V, I-V-IV, IV-I-V
Instrument = Instrument: String Quartet, Piano, Electric Guitar
Voice = Voice Leading: Contrary Motion, Parallel 3rds, Parallel 5ths
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 variables ConsInstr and ConsNotes
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 = How proficient are you at your second musical instrument (0-5, 0 = not at all)

By looking into pair plot (Figure 1) among three numeric variables, classical, popular, and popular, we find that classical and popular are negatively related, which makes sense.

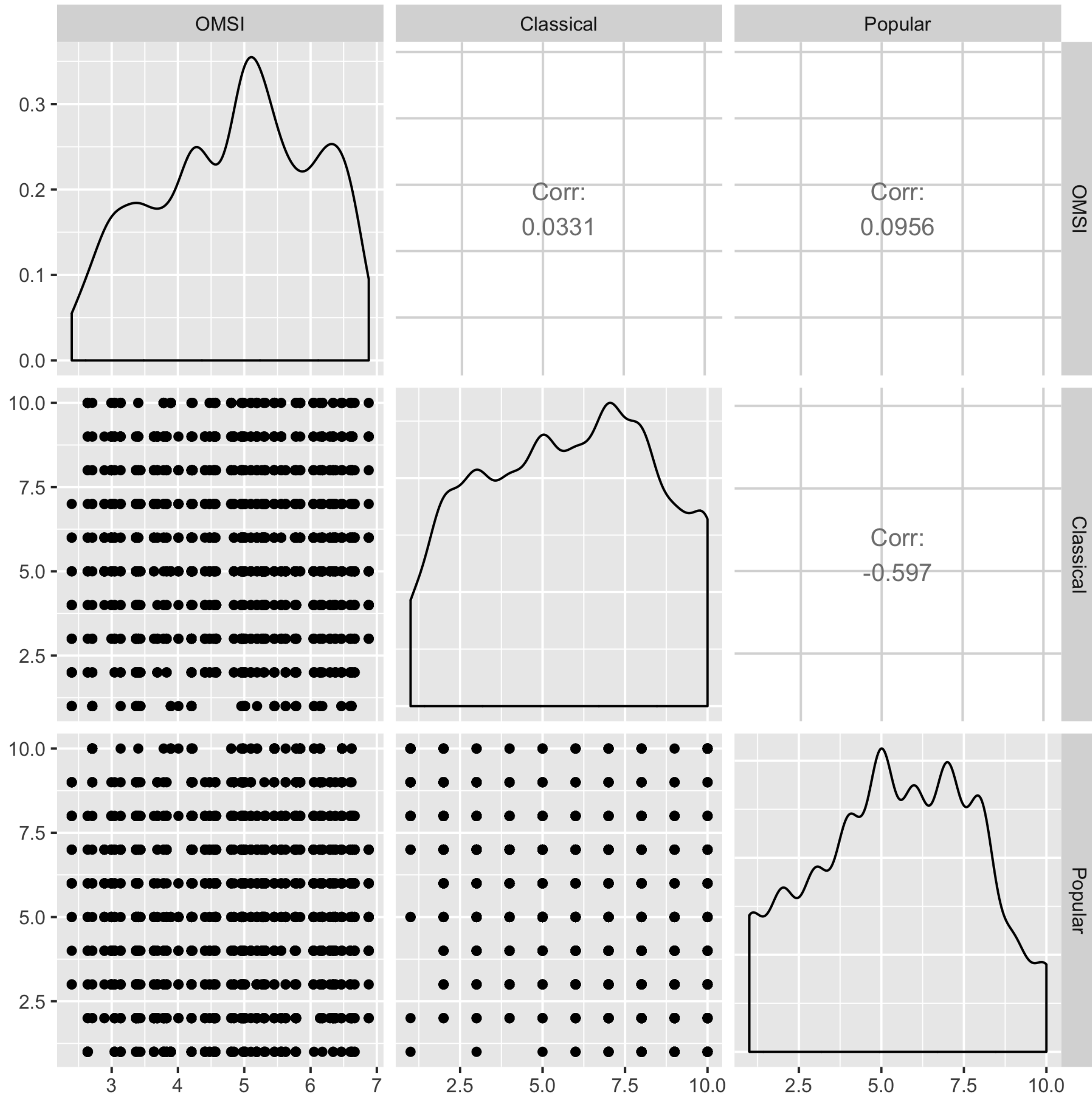

Figure 1: Pair Plot of Numeric Variables.

By looking into distributions categorical variables (Figure 2), taking APThoery and CollegeMusic as example, we find that their categories are imbalenced. However, since they are factors and we are not going to building any classification models, we do not transform them.

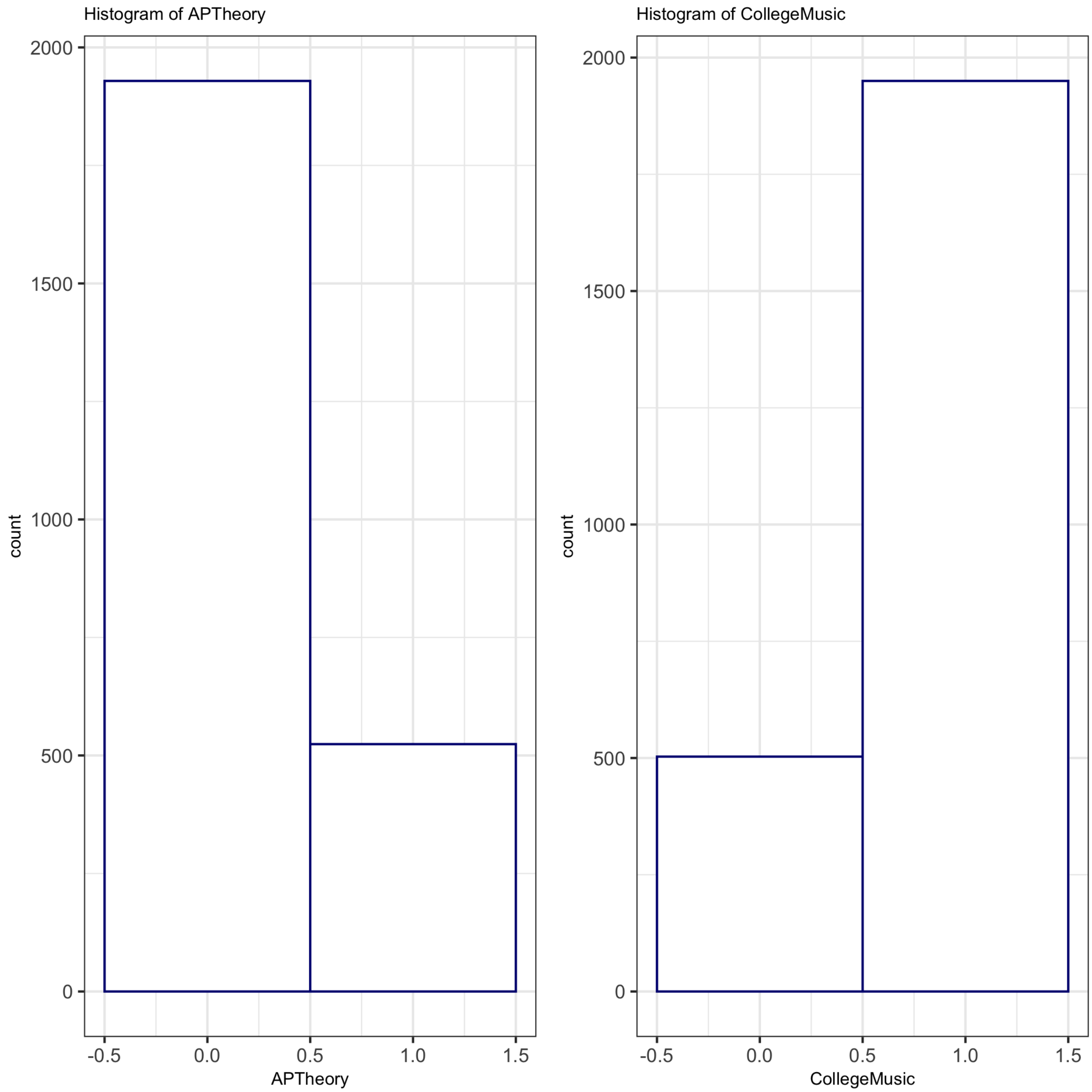

Figure 2: Histogram.

By going through all the steps mentioned in the Method section, we developed two independent multilevel models with random intercepts and slopes for Classical and Popular ratings as following:

[1]$Classical \sim Harmony + Voice + Instrument + Selfdeclare + ConsInstr + Instr.minus.Notes + ClsListen + CollegeMusic + ApTheory + PianoPlay + GuitarPlay + (1 + Harmony + Instrument +$

[1]See Appendix 5(a) i. for model coefficient estimates and statistics

$Voice|Subject)+Harmony:Selfdeclare+X16.minus.17:Selfdeclare+CollgeMusic:Selfdeclare+GuitarPlay:Selfdeclare$

[2] $Popular \sim Harmony+Voice+Instrument+Selfdeclare+ConsInstr+ConsNotes+X16.minus.17+Instr.minus.Notes+PachListen+KnowRob+X1990s2000s+ApTheory+Composing+(1+Harmony+Instrument+Voice|Subject)+PachListen:Selfdeclare+X1990s2000s.minus.1960s1970s:Selfdeclare+Composing:Selfdeclare$

3.1 Which of Harmony, Instrument, and Voice has the strongest influence on ratings?

For classical ratings, based on the summary table (Appendix 5(a) i) of our final model for classical ratings, Harmony, Voice, and Instrument are all statistically significant at a significance level of 5% since at least one of their levels has a t-value greater than or equal to 2. By further looking into the coefficient estimates of three main factors, we can tell that two levels of Instrument have the highest t-values among all levels of three main factors and much larger coefficient estimates than other two do. Thus, we can confirm that Instrument exerts the strongest influence among Instrument, Harmony, and Voice, on rating Classical music. By referring to the summary table in Appendix 5(a) iii., if a music stimulus includes instrument piano, its rating of classical music will increase by 5.44; if a music stimulus includes instrument string, its rating of classical music will increase by 7.33; if a music stimulus includes instrument guitar, its rating of classical music will increase by 4.09. Since classical and popular ratings were given by 70 participants to 36 music stimuli, personal bias exists in ratings due to different definitions of classical and popular music. To deal with such personal bias, we include random effect on intercept, harmony, instrument, and voice by subject (participant) in our final model. From the summary table (Appendix 5(a) iii.), we can see that the standard deviation of instrument guitar is 1.62, and thus, the 95% confidence interval of guitar is from 0.85 to 7.33. Since its 95% confidence interval is positive, we conclude that guitar is positively associated with classical ratings. By going through the same procedure, the 95% confidence interval of piano is [1.26, 9.64], and it's positively associated with classical ratings; the 95% confidence interval of piano is [3.93, 10.73], and it's positively associated with classical ratings. Moreover, the instrument guitar also has the largest random effect variance among all three level, which means that instrument guitar's influence on classical ratings vary greatly by participants. As a result, the instrument has the strongest influence on classical ratings, and all levels of it are positively associated with classical ratings.

Although harmony and voice has weaker influence on classical ratings than instrument does, harmony and voice still somehow influence classical ratings. From the summary table (Appendix 5(a) ii.), we can tell that all levels of harmony are statistically significant, and among all four levels, Harmony I-V-VI has the strongest influence on classical ratings. If a music stimulus includes Harmony I-V-VI, its rating of classical music will increase by 4.09. We can also confirm that all levels of harmony are positively related with classical ratings. Moreover, whether the respondent is familiar with one or the other (or both) of the Pachelbel rants/comedy bits does not matter to classical ratings since neither ‘KnowRob‘ nor ‘KnowAxis‘ were selected as a influencial predictor in our final model for classical ratings.

[2] See Appendix 5(b) i. for model coefficient estimates and statistics

From the summary table (Appendix 5(a) vi.), we can tell that all levels of voice are statistically significant, and among all three levels, voice contrary has the strongest influence on classical ratings. If a music stimulus includes voice contrary, its rating of classical music will increase by 3.77. Also, we conclude that voice is negatively related with classical ratings.

By going through the same procedure, we conclude that Instrument has the strongest influence among the three design factors on rating Popular music by referring to summary tables (Appendix 5(b) i - vi.). Almost all levels of three main factors are statistically significant. Harmony and instrument are negatively associated with popular ratings, but voice is positively related with popular ratings. For example, if the music stimulus includes instrument string, its popular ratings will decrease by 2.54.

3.2 Do musicians and non-musicians identify classical music differently?

From the summary table (Appendix 5(a) i.), we can tell that the variable Selfdeclare is not statistically significant at the significance level of 5%. However, its interactions with CollegeMusic and GuitarPlay are statistically significant at the significance level of 5%. If the partipant declares both musician and having taken music course(s) in college (interaction term of Selfdeclare and CollegeMusic), the classical rating will decrease by 1.55; if the partipant declares both musician and playing guitar (interaction term of Selfdeclare and GuitarPlay), the classical rating will decrease by 1.2. We think one possible interpretation would be that musicians who have taken college music course(s) are able to distinguish music stimuli that are confusing in their categories to most people. Thus, musicians are more dependent on their music background (CollegeMusic) and experiences of playing guitar in daily life (GuitarPlay) when rating music stimuli as classical music than non-musicians do. In general, we conclude that musicians and non-musicians identify classical music differently.

3.3 Do predictors drive classical and popular music differ?

There are differences in the things that drive classical and popular ratings. By refering to summary tables of classical and popular ratings (Appendix 5(a) i. and 5(b) i.), we can first conclude that all three main factors are influencial on both classical and popular ratings. While harmony and instrument are positively associated with classical ratings, and voice is negatively associated with classical ratings, harmony and instrument are negatively associated with popular ratings, and voice is positively associated with popular ratings. Among levels of harmony, Harmony I-V-VI has the strongest influence on both classical and popular ratings; among levels of instrument, instrument string has the strongest influence on classical ratings, and instrument guitar has the strongest influence on popular ratings; among levels of voice, voice contrary has the strongest influence on classical ratings, and voice 5th has the strongest influence on popular ratings.

There are also some differences in predictors other than three main factors driving classical and popular ratings. By looking to summary tables of classical and popular ratings (Appendix 5(a) i. and 5(b) i.) again, we can tell that influential predictors differ greatly. We define a predictor with a t-value greater than or equal to 2 as stististically significant at the significance level of 5%. APTheory, GuitarPlay, Selfdeclare:Harmony I-V-VI, Selfdeclare:CollegeMusic, and Selfdeclare:GuitarPlay are predictors driving classical ratings; X16.minus.17, Selfdeclare, ConsInstr, PachListen, KnowRob, Selfdeclare:PachListen, and Selfdeclare:X1990s2000s.minus.1960s1970s are predictors driving popular ratings. Popular ratings depend more predictors than classical ratings do. Moreover, classical ratings depends more on interaction terms of Selfdeclare and other predictotrs.

4 Discussion

In this paper, we use the data collected Vincent Rossi, a student of the University of Pittsburgh, from an experiment where 70 participant identitfied 36 music stimuli as either classical or popular to develop models. By going through steps such as preliminary data cleaning and selecting fixed effect and random effect, we built two multilevel models for classical ratings and popular ratings. With our final model, we identify instrument as the most influential predictors among three design factors. Furthermore, we identify predictors such as Selfdclare and APTheory are alo influential when distinguish popular music and classical music. Moreover, we find that musicians and non-musician give classical ratings depending on different fatcors. Our final models include fixed effect for interpretability and random effect to address personal bias across 70 participants. Last but not least, the results driven by our final models are consistent with experiment results.

However, our models have weakness. In both models for classical and popular ratings, we can see that some predictors are not statistically significant, which decreases the validity and interpretability of models. To address this issue, we can try picking fixed effect manually instead of automatic method in the future. Furthermore, the interpretability of our model could be further improved. For example, we have interaction terms of Selfdeclare and other predictors. They are actually hard to interpret. In the future resrearch, we should develop a valid way to interpret these interaction terms.

This paper tests Ivan Jimenez's experiment on measuring the influence of instrument, harmonic motion, and voice leading on listeners' identification of music as "classical" or "popular" and expands the scale by including more predictors. Finally, we drive results that are consistent with experiment results and valid the experiment.

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/.

Appendix

1.

```r
library(ggplot2)
library(dplyr)
library(gridExtra)
library(knitr)
library(kableExtra)
library(GGally)
library(lme4)
library(LMERConvenienceFunctions)
library(RLRsim)
library(knitr)
library(papeR)
source("mlm-facet-plots.r")
source("residual-functions.r")

ratings <- read.csv("ratings.csv")
# Reomve the first column as it's same as the rownames
# As instruction in HW10 doc, we're ignoring this variable for this assignment
ratings <- ratings[ , -c(1, which(colnames(ratings) == "first12"))]

# Remove NA's rows of Classical and Popular.
# Remove variables `X1stInstr` and `X2ndInstr`.
ratings <- ratings[which(is.na(ratings$Classical) == F |
                           is.na(ratings$Popular) == F),
                   -c(which(colnames(ratings) == "X1stInstr"),
                      which(colnames(ratings) == "X2ndInstr"))]
# summary(ratings)
# str(ratings)

# Remove rows with abnormal Classical or Popular values
ratings <- ratings[!(ratings$Classical < 1 | ratings$Classical > 10 |
                       ratings$Popular < 1 | ratings$Popular > 10), ]

# Remove decimal numbers from dataset
ratings <- ratings[ratings$Classical == as.integer(ratings$Classical) &
                     ratings$Popular == as.integer(ratings$Popular), ]

# Interpolation with Mode
na.vars <- c("ConsNotes", "PachListen", "ClsListen", "KnowRob", "KnowAxis",
             "X1990s2000s", "X1990s2000s.minus.1960s1970s", "CollegeMusic",
             "NoClass", "APTheory", "Composing")
get.mode <- function(v) {
   uniq.val <- unique(v)
   uniq.val[which.max(tabulate(match(v, uniq.val)))]
}
```

```r
for (var in na.vars) {
  ratings[which(is.na(ratings[, var])), var] <- get.mode(na.omit(ratings[, var]))
}

# ggplot(data = ratings, aes(x = Instr.minus.Notes, y = ConsInstr)) +
  # geom_point(pch = 16) +
  # geom_smooth(method = "loess")
# ggpairs(ratings[ , c("OMSI", "Classical", "Popular")])


h1 <- ggplot(data = ratings, aes(x = OMSI)) +
  geom_histogram(binwidth = 100, col = "navy", fill = "white") +
  labs(title = "Histogram of OMSI") +
  theme_bw() +
  theme(axis.title = element_text(size = 8),
        plot.title = element_text(size = 8))
h2 <- ggplot(data = ratings, aes(x = Classical)) +
  geom_histogram(binwidth = 1, col = "navy", fill = "white") +
  labs(title = "Histogram of Claasical") +
  theme_bw() +
  theme(axis.title = element_text(size = 8),
        plot.title = element_text(size = 8))
h3 <- ggplot(data = ratings, aes(x = Popular)) +
  geom_histogram(binwidth = 1, col = "navy", fill = "white") +
  labs(title = "Histogram of Popular") +
  theme_bw()+
  theme(axis.title = element_text(size = 8),
        plot.title = element_text(size = 8))
# grid.arrange(h1, h2, h3, ncol = 3)


# Log transform
ratings$OMSI = log(ratings$OMSI)

h4 <- ggplot(data = ratings, aes(x = APTheory)) +
  geom_histogram(binwidth = 1, col = "navy", fill = "white") +
  labs(title = "Histogram of APTheory") +
  theme_bw() +
  theme(axis.title = element_text(size = 8),
        plot.title = element_text(size = 8))
h5 <- ggplot(data = ratings, aes(x = CollegeMusic)) +
  geom_histogram(binwidth = 1, col = "navy", fill = "white") +
  labs(title = "Histogram of CollegeMusic") +
  theme_bw()+
  theme(axis.title = element_text(size = 8),
        plot.title = element_text(size = 8))
# grid.arrange(h4, h5, ncol = 2)
```

2(a).

```r
fit.1 <- lm(Classical ~ Instrument + Harmony + Voice, data = ratings)
par(mfrow = c(2, 2))
# plot(fit.1)
# summary(fit.1)

fit.2 <- step(lm(Classical ~ Instrument * Harmony * Voice, data = ratings),
              direction = "backward", trace = 0)
par(mfrow = c(2, 2))
# plot(fit.2)
# summary(fit.2)

# Model Selection
# anova(fit.1, fit.2)
measure <- data.frame(AIC = c(AIC(fit.1), AIC(fit.2)),
                      BIC = c(BIC(fit.1), BIC(fit.2)))
rownames(measure) <- c("Fit.1", "Fit.2")
# kable(measure, "latex", booktabs = T) %>%
#   kable_styling(position = "center")
```

2(b).

```r
# (ii)
mlm.1 <- lmer(Classical ~ Instrument + Harmony + Voice + (1 | Subject),
              REML = F, lmerControl(optimizer = "bobyqa"), data = ratings)
mlm1.sum <- summary(mlm.1)
measures <- data.frame(Fit.1 = c(AIC(fit.1), BIC(fit.1)),
                       MLM.1 = c(mlm1.sum$AICtab[1], mlm1.sum$AICtab[2]),
                       Diff =  c(mlm1.sum$AICtab[1], mlm1.sum$AICtab[2]) -
                         c(AIC(fit.1), BIC(fit.1)))
# kable(measures, "latex", booktabs = T) %>%
#   kable_styling(position = "center")
```

2(c).

```r
# (i)
mlm.2.fixed <- fitLMER.fnc(mlm.1, method="AIC", set.REML.FALSE = T)
```

```r
vars <- attr(terms(formula(mlm.2.fixed)), "term.labels")
vars <- vars[-length(vars)]
```

```r
mlm.2 <- fitLMER.fnc(mlm.2.fixed, ran.effects = list(slopes = vars,
                                                     by.vars = "Subject",
                                                     corr = rep(1, length(vars))))
```

(i).

```r
mlm2.sum <- summary(mlm.2)
# anova(mlm.1, mlm.2)
measures <- data.frame(MLM.1 = c(mlm1.sum$AICtab[1]),
                       MLM.2 = c(mlm2.sum$AICtab[1]),
                       Diff =  c(mlm2.sum$AICtab[1]) - c(mlm1.sum$AICtab[1]))
# kable(measures, "latex", booktabs = T) %>%
#   kable_styling(position = "center")
```

(ii).

```r
res <- r.cond(mlm.2)
robust.sd <- diff(quantile(res,c(.025,.975)))/(2*1.96)
res <- res/robust.sd
fit <- yhat.cond(mlm.2)
resdata <- data.frame(subject = ratings$Subject, resid = res, fitted = fit)
resparams <- data.frame(subject = unique(ratings$Subject),
                        int1 = 0, slo1 = 0,
                        int2 = 2, slo2 = 0,
                        int3 = -2, slo3 = 0)
# mlm_facets(resdata,"subject", x = "fitted", y = "resid", params = resparams,
#            lty=c(1, 2, 2),size=c(0.5, 0.5, 0.5))
```

3(a).

```r
# Convert some numeric variables to factors
to.factor <- c("CollegeMusic", "APTheory")
for (var in to.factor) {
  ratings[ , var] <- as.factor(ratings[ , var])
}

# Backward-fitting on fixed effect
mlm.3.fixed <- step(lm(Classical ~ . + Harmony:Voice - Subject - Popular, data = ratings),
                    direction = "backward", trace = 0)
# summary(mlm.3.fixed)
mlm.3.fixed <- lm(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
                    OMSI + X16.minus.17 + Instr.minus.Notes + ClsListen +
                    X1990s2000s + X1990s2000s.minus.1960s1970s + NoClass +
                    APTheory + Composing + PianoPlay + Harmony:Voice,
                  data = ratings)
# summary(mlm.3.fixed)
```

3(b).

```r
mlm.3 <- lmer(as.formula(paste("Classical ~",
                               paste(as.character(formula(mlm.3.fixed))[3],
                                     "(1 + Harmony + Instrument | Subject)",
```

```r
                                      sep = "+"))),
              lmerControl(optimizer = "bobyqa"),  REML = F, data = ratings)
# summary(mlm.3)
```

4.

```r
is.musician <- as.numeric(ratings$Selfdeclare)
is.musician[which(is.musician <= 2)] <- 0
is.musician[which(is.musician > 2)] <- 1
is.musician  <- as.factor(is.musician)
musician.ratings <- ratings
musician.ratings$Selfdeclare <- is.musician
```

5(a) i.

```r
# Step BIC on Fixed effect
mlm.final.fix <- step(lm(Classical ~ Selfdeclare*(. - Popular - Subject),
                         data = musician.ratings),
                  direction = "backward", trace = 0,
                  k = log(nrow(musician.ratings)))
summary(mlm.final.fix)
```

```
##
## Call:
## lm(formula = Classical ~ Selfdeclare + Harmony + Instrument +
##     Voice + X16.minus.17 + ConsInstr + ConsNotes + Instr.minus.Notes +
##     ClsListen + KnowRob + KnowAxis + X1990s2000s + CollegeMusic +
##     APTheory + Composing + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
##     Selfdeclare:X16.minus.17 + Selfdeclare:Instr.minus.Notes +
##     Selfdeclare:ClsListen + Selfdeclare:KnowRob + Selfdeclare:CollegeMusic +
##     Selfdeclare:APTheory + Selfdeclare:PianoPlay + Selfdeclare:GuitarPlay,
##     data = musician.ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7997 -1.4969  0.0304  1.4014  7.2923
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.274505   0.270420  15.807  < 2e-16 ***
## Selfdeclare1                    1.487254   0.338824   4.389 1.19e-05 ***
## HarmonyI-V-IV                  -0.074421   0.154223  -0.483 0.629456
## HarmonyI-V-VI                   0.297395   0.154237   1.928 0.053951 .
## HarmonyIV-I-V                   0.057598   0.153803   0.374 0.708072
## Instrumentpiano                 1.339457   0.102903  13.017  < 2e-16 ***
## Instrumentstring                3.036443   0.102908  29.506  < 2e-16 ***
## Voicepar3rd                    -0.380936   0.103013  -3.698 0.000222 ***
## Voicepar5th                    -0.359658   0.102952  -3.493 0.000485 ***
## X16.minus.17                    0.014929   0.020447   0.730 0.465368
```

```
## ConsInstr                      -0.346743   0.049645  -6.984 3.68e-12 ***
## ConsNotes                       0.191911   0.053957   3.557 0.000383 ***
## Instr.minus.Notes               0.352026   0.055895   6.298 3.57e-10 ***
## ClsListen                      -0.027528   0.039108  -0.704 0.481568
## KnowRob                         0.249412   0.064009   3.897 0.000100 ***
## KnowAxis                        0.136515   0.029657   4.603 4.38e-06 ***
## X1990s2000s                    -0.193493   0.033954  -5.699 1.35e-08 ***
## CollegeMusic1                   0.985775   0.166660   5.915 3.79e-09 ***
## APTheory1                       1.937835   0.260055   7.452 1.28e-13 ***
## Composing                       0.224405   0.046337   4.843 1.36e-06 ***
## PianoPlay                      -0.500381   0.076931  -6.504 9.45e-11 ***
## GuitarPlay                      2.210948   0.197350  11.203  < 2e-16 ***
## Selfdeclare1:HarmonyI-V-IV      0.076793   0.242369   0.317 0.751390
## Selfdeclare1:HarmonyI-V-VI      1.194450   0.242229   4.931 8.73e-07 ***
## Selfdeclare1:HarmonyIV-I-V     -0.009465   0.241951  -0.039 0.968798
## Selfdeclare1:X16.minus.17      -0.383053   0.038792  -9.874  < 2e-16 ***
## Selfdeclare1:Instr.minus.Notes -0.210059   0.062548  -3.358 0.000796 ***
## Selfdeclare1:ClsListen          0.250711   0.074402   3.370 0.000764 ***
## Selfdeclare1:KnowRob           -0.374644   0.076195  -4.917 9.38e-07 ***
## Selfdeclare1:CollegeMusic1     -1.487293   0.245393  -6.061 1.57e-09 ***
## Selfdeclare1:APTheory1         -1.775444   0.325110  -5.461 5.22e-08 ***
## Selfdeclare1:PianoPlay          0.473797   0.085188   5.562 2.96e-08 ***
## Selfdeclare1:GuitarPlay        -2.281522   0.196080 -11.636  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.082 on 2420 degrees of freedom
## Multiple R-squared:  0.3752, Adjusted R-squared:  0.367
## F-statistic: 45.42 on 32 and 2420 DF,  p-value: < 2.2e-16
```

```r
# random intercept
mlm.semi <- lmer(as.formula(paste("Classical ~",
                                   paste(as.character(formula(mlm.final.fix))[3],
                                         "(1 | Subject)",
                                         sep = "+"))),
                  lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)

exactLRT(mlm.semi, mlm.final.fix)
```

```
##
##  simulated finite sample distribution of LRT. (p-value based on 10000
##  simulated values)
##
## data:
## LRT = 410.3, p-value < 2.2e-16
```

```r
summary(mlm.semi)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Selfdeclare + Harmony + Instrument + Voice + X16.minus.17 +
##     ConsInstr + ConsNotes + Instr.minus.Notes + ClsListen + KnowRob +
##     KnowAxis + X1990s2000s + CollegeMusic + APTheory + Composing +
```

```
##     PianoPlay + GuitarPlay + Selfdeclare:Harmony + Selfdeclare:X16.minus.17 +
##     Selfdeclare:Instr.minus.Notes + Selfdeclare:ClsListen + Selfdeclare:KnowRob +
##     Selfdeclare:CollegeMusic + Selfdeclare:APTheory + Selfdeclare:PianoPlay +
##     Selfdeclare:GuitarPlay + (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10185.3  10388.5  -5057.7  10115.3     2418
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2308 -0.6216 -0.0154  0.6412  3.7578
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.8822   0.9393
##  Residual             3.3866   1.8403
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                Estimate Std. Error t value
## (Intercept)                    4.214966   0.677893   6.218
## Selfdeclare1                   1.495439   0.875074   1.709
## HarmonyI-V-IV                 -0.076385   0.136340  -0.560
## HarmonyI-V-VI                  0.311125   0.136393   2.281
## HarmonyIV-I-V                  0.061116   0.135965   0.449
## Instrumentpiano                1.348845   0.091033  14.817
## Instrumentstring               3.039591   0.091089  33.369
## Voicepar3rd                   -0.384239   0.091064  -4.219
## Voicepar5th                   -0.357336   0.091014  -3.926
## X16.minus.17                  -0.003133   0.054330  -0.058
## ConsInstr                     -0.354193   0.140482  -2.521
## ConsNotes                      0.194722   0.152434   1.277
## Instr.minus.Notes              0.336549   0.157157   2.141
## ClsListen                     -0.019069   0.109282  -0.174
## KnowRob                        0.244597   0.181828   1.345
## KnowAxis                       0.137128   0.084072   1.631
## X1990s2000s                   -0.178893   0.094812  -1.887
## CollegeMusic1                  0.985621   0.473023   2.084
## APTheory1                      1.946655   0.736875   2.642
## Composing                      0.223801   0.130234   1.718
## PianoPlay                     -0.489821   0.217257  -2.255
## GuitarPlay                     2.194759   0.558415   3.930
## Selfdeclare1:HarmonyI-V-IV     0.081159   0.214251   0.379
## Selfdeclare1:HarmonyI-V-VI     1.182255   0.214152   5.521
## Selfdeclare1:HarmonyIV-I-V    -0.012886   0.213880  -0.060
## Selfdeclare1:X16.minus.17     -0.359163   0.107309  -3.347
## Selfdeclare1:Instr.minus.Notes -0.190110   0.175599  -1.083
## Selfdeclare1:ClsListen         0.246706   0.209987   1.175
## Selfdeclare1:KnowRob          -0.368477   0.216710  -1.700
## Selfdeclare1:CollegeMusic1    -1.500499   0.696278  -2.155
## Selfdeclare1:APTheory1        -1.768150   0.917616  -1.927
## Selfdeclare1:PianoPlay         0.459623   0.240378   1.912
```

```
## Selfdeclare1:GuitarPlay        -2.271040   0.555482  -4.088
```

```r
# General
mlm.semi.2 <- lme4::lmer(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
                     ConsInstr  + Instr.minus.Notes + ClsListen  + CollegeMusic +
                     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
                     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic +
                     Selfdeclare:GuitarPlay + (1 | Subject),
                  lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 589.43, p-value < 2.2e-16
```

```r
# Random slope
mlm.final <- ffRanefLMER.fnc(mlm.semi.2,
                             ran.effects = c("(Harmony + Voice + Instrument | Subject)"),
                             log.file = F)
```

```
## evaluating addition of (Harmony+Voice+Instrument|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##       maxfun < 10 * length(par)^2 is not recommended.
##     Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp),  :
##       convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations exceeded
##     boundary (singular) fit: see ?isSingular
##  not adding (Harmony+Voice+Instrument|Subject) to model
```

```r
summary(mlm.final)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Harmony + Instrument + Voice + Selfdeclare + ConsInstr +
##     Instr.minus.Notes + ClsListen + CollegeMusic + APTheory +
##     PianoPlay + GuitarPlay + Selfdeclare:Harmony + Selfdeclare:X16.minus.17 +
##     Selfdeclare:CollegeMusic + Selfdeclare:GuitarPlay + (1 |      Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10183.7  10328.8  -5066.9  10133.7     2428
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2004 -0.6187 -0.0180  0.6349  3.7075
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1.178    1.085
```

```
##  Residual             3.387    1.840
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                              Estimate Std. Error t value
## (Intercept)                   4.09308    0.51370   7.968
## HarmonyI-V-IV                -0.07663    0.13634  -0.562
## HarmonyI-V-VI                 0.31238    0.13639   2.290
## HarmonyIV-I-V                 0.06114    0.13596   0.450
## Instrumentpiano               1.35037    0.09103  14.834
## Instrumentstring              3.04031    0.09109  33.377
## Voicepar3rd                  -0.38466    0.09106  -4.224
## Voicepar5th                  -0.35712    0.09101  -3.924
## Selfdeclare1                  1.25115    0.72387   1.728
## ConsInstr                    -0.13741    0.09392  -1.463
## Instr.minus.Notes             0.11910    0.09007   1.322
## ClsListen                     0.11745    0.09450   1.243
## CollegeMusic1                 0.35870    0.47111   0.761
## APTheory1                     0.51619    0.42130   1.225
## PianoPlay                    -0.07871    0.10205  -0.771
## GuitarPlay                    1.44503    0.50587   2.857
## HarmonyI-V-IV:Selfdeclare1  0.08168    0.21425   0.381
## HarmonyI-V-VI:Selfdeclare1  1.18138    0.21415   5.517
## HarmonyIV-I-V:Selfdeclare1 -0.01280    0.21388  -0.060
## Selfdeclare0:X16.minus.17   -0.03965    0.05858  -0.677
## Selfdeclare1:X16.minus.17   -0.26118    0.08838  -2.955
## Selfdeclare1:CollegeMusic1 -0.89845    0.72694  -1.236
## Selfdeclare1:GuitarPlay    -1.47579    0.52481  -2.812
```

5(a) ii.

```
# All levels of Harmony
mlm.semi.2 <- lme4::lmer(Classical ~ Harmony - 1 + Instrument + Voice + Selfdeclare +
                     ConsInstr  + Instr.minus.Notes + ClsListen  + CollegeMusic +
                     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
                     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic +
                     Selfdeclare:GuitarPlay + (1 | Subject),
                  lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 589.43, p-value < 2.2e-16
```

```
# Random slope
mlm.final.harmony <- ffRanefLMER.fnc(mlm.semi.2,
                            ran.effects = c("(Harmony - 1 + Voice + Instrument | Subject)"),
                            log.file = F)
```

```
## evaluating addition of (Harmony-1+Voice+Instrument|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp),  :
##        convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations exceeded
##      Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
##        Model failed to converge with max|grad| = 0.059141 (tol = 0.002, component 1)
##  not adding (Harmony-1+Voice+Instrument|Subject) to model
```

```r
summary(mlm.final.harmony)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Harmony - 1 + Instrument + Voice + Selfdeclare +
##     ConsInstr + Instr.minus.Notes + ClsListen + CollegeMusic +
##     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
##     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic + Selfdeclare:GuitarPlay +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10183.7  10328.8  -5066.9  10133.7     2428
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2004 -0.6187 -0.0180  0.6349  3.7075
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1.178    1.085
##  Residual             3.387    1.840
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                            Estimate Std. Error t value
## HarmonyI-IV-V               4.09308    0.51370   7.968
## HarmonyI-V-IV               4.01645    0.51376   7.818
## HarmonyI-V-VI               4.40546    0.51388   8.573
## HarmonyIV-I-V               4.15423    0.51369   8.087
## Instrumentpiano             1.35037    0.09103  14.834
## Instrumentstring            3.04031    0.09109  33.377
## Voicepar3rd                -0.38466    0.09106  -4.224
## Voicepar5th                -0.35712    0.09101  -3.924
## Selfdeclare1                1.25115    0.72387   1.728
## ConsInstr                  -0.13741    0.09392  -1.463
## Instr.minus.Notes           0.11910    0.09007   1.322
## ClsListen                   0.11745    0.09450   1.243
## CollegeMusic1               0.35870    0.47111   0.761
## APTheory1                   0.51619    0.42130   1.225
## PianoPlay                  -0.07871    0.10205  -0.771
## GuitarPlay                  1.44503    0.50587   2.857
## HarmonyI-V-IV:Selfdeclare1  0.08168    0.21425   0.381
## HarmonyI-V-VI:Selfdeclare1  1.18138    0.21415   5.517
## HarmonyIV-I-V:Selfdeclare1 -0.01280    0.21388  -0.060
```

```
## Selfdeclare0:X16.minus.17  -0.03965    0.05858  -0.677
## Selfdeclare1:X16.minus.17  -0.26118    0.08838  -2.955
## Selfdeclare1:CollegeMusic1 -0.89845    0.72694  -1.236
## Selfdeclare1:GuitarPlay    -1.47579    0.52481  -2.812
```

5(a) iii.

```
# All levels of Instrument
mlm.semi.2 <- lme4::lmer(Classical ~ Instrument - 1 + Harmony + Voice + Selfdeclare +
                     ConsInstr  + Instr.minus.Notes + ClsListen  + CollegeMusic +
                     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
                     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic +
                     Selfdeclare:GuitarPlay + (1 | Subject),
                  lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 589.43, p-value < 2.2e-16
```

```
# Random slope
mlm.final.instr <- ffRanefLMER.fnc(mlm.semi.2,
                              ran.effects = c("(Instrument - 1 + Voice + Harmony | Subject)"),
                              log.file = F)
```

```
## evaluating addition of (Instrument-1+Voice+Harmony|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      boundary (singular) fit: see ?isSingular
##  not adding (Instrument-1+Voice+Harmony|Subject) to model
```

```
summary(mlm.final.instr)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Instrument - 1 + Harmony + Voice + Selfdeclare +
##     ConsInstr + Instr.minus.Notes + ClsListen + CollegeMusic +
##     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
##     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic + Selfdeclare:GuitarPlay +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10183.7  10328.8  -5066.9  10133.7     2428
##
## Scaled residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3.2004 -0.6187 -0.0180  0.6349  3.7075
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1.178    1.085
##  Residual             3.387    1.840
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                            Estimate Std. Error t value
## Instrumentguitar            4.09308    0.51370   7.968
## Instrumentpiano             5.44346    0.51359  10.599
## Instrumentstring            7.13340    0.51385  13.882
## HarmonyI-V-IV              -0.07663    0.13634  -0.562
## HarmonyI-V-VI               0.31238    0.13639   2.290
## HarmonyIV-I-V               0.06114    0.13596   0.450
## Voicepar3rd                -0.38466    0.09106  -4.224
## Voicepar5th                -0.35712    0.09101  -3.924
## Selfdeclare1                1.25115    0.72387   1.728
## ConsInstr                  -0.13741    0.09392  -1.463
## Instr.minus.Notes           0.11910    0.09007   1.322
## ClsListen                   0.11745    0.09450   1.243
## CollegeMusic1               0.35870    0.47111   0.761
## APTheory1                   0.51619    0.42130   1.225
## PianoPlay                  -0.07871    0.10205  -0.771
## GuitarPlay                  1.44503    0.50587   2.857
## HarmonyI-V-IV:Selfdeclare1  0.08168    0.21425   0.381
## HarmonyI-V-VI:Selfdeclare1  1.18138    0.21415   5.517
## HarmonyIV-I-V:Selfdeclare1 -0.01280    0.21388  -0.060
## Selfdeclare0:X16.minus.17  -0.03965    0.05858  -0.677
## Selfdeclare1:X16.minus.17  -0.26118    0.08838  -2.955
## Selfdeclare1:CollegeMusic1 -0.89845    0.72694  -1.236
## Selfdeclare1:GuitarPlay    -1.47579    0.52481  -2.812
```

5(a) vi.

```
# All levels of Voice
mlm.semi.2 <- lme4::lmer(Classical ~ Voice - 1 + Instrument + Harmony + Selfdeclare +
                      ConsInstr  + Instr.minus.Notes + ClsListen  + CollegeMusic +
                      APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
                      Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic +
                      Selfdeclare:GuitarPlay + (1 | Subject),
                   lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
```

```
## RLRT = 589.43, p-value < 2.2e-16
```

```r
# Random slope
mlm.final.voice <- ffRanefLMER.fnc(mlm.semi.2,
                              ran.effects = c("(Voice - 1 + Harmony + Instrument | Subject)"),
                              log.file = F)
```

```
## evaluating addition of (Voice-1+Harmony+Instrument|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp),  :
##        convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations exceeded
##      boundary (singular) fit: see ?isSingular
##  not adding (Voice-1+Harmony+Instrument|Subject) to model
```

```r
summary(mlm.final.voice)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Classical ~ Voice - 1 + Instrument + Harmony + Selfdeclare +
##     ConsInstr + Instr.minus.Notes + ClsListen + CollegeMusic +
##     APTheory + PianoPlay + GuitarPlay + Selfdeclare:Harmony +
##     Selfdeclare:X16.minus.17 + Selfdeclare:CollegeMusic + Selfdeclare:GuitarPlay +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10183.7  10328.8  -5066.9  10133.7     2428
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2004 -0.6187 -0.0180  0.6349  3.7075
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 1.178    1.085
##  Residual             3.387    1.840
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                              Estimate Std. Error t value
## Voicecontrary                 4.09308    0.51370   7.968
## Voicepar3rd                   3.70843    0.51370   7.219
## Voicepar5th                   3.73596    0.51383   7.271
## Instrumentpiano               1.35037    0.09103  14.834
## Instrumentstring              3.04031    0.09109  33.377
## HarmonyI-V-IV                -0.07663    0.13634  -0.562
## HarmonyI-V-VI                 0.31238    0.13639   2.290
## HarmonyIV-I-V                 0.06114    0.13596   0.450
## Selfdeclare1                  1.25115    0.72387   1.728
## ConsInstr                    -0.13741    0.09392  -1.463
## Instr.minus.Notes             0.11910    0.09007   1.322
## ClsListen                     0.11745    0.09450   1.243
```

```
## CollegeMusic1               0.35870    0.47111   0.761
## APTheory1                   0.51619    0.42130   1.225
## PianoPlay                  -0.07871    0.10205  -0.771
## GuitarPlay                  1.44503    0.50587   2.857
## HarmonyI-V-IV:Selfdeclare1  0.08168    0.21425   0.381
## HarmonyI-V-VI:Selfdeclare1  1.18138    0.21415   5.517
## HarmonyIV-I-V:Selfdeclare1 -0.01280    0.21388  -0.060
## Selfdeclare0:X16.minus.17  -0.03965    0.05858  -0.677
## Selfdeclare1:X16.minus.17  -0.26118    0.08838  -2.955
## Selfdeclare1:CollegeMusic1 -0.89845    0.72694  -1.236
## Selfdeclare1:GuitarPlay    -1.47579    0.52481  -2.812
```

5(b) i.

```
# Step BIC on Fixed effect
mlm.final.fix <- step(lm(Popular ~ Selfdeclare*(. - Classical - Subject),
                         data = musician.ratings),
                  direction = "backward", trace = 0,
                  k = log(nrow(musician.ratings)))
summary(mlm.final.fix)
```

```
##
## Call:
## lm(formula = Popular ~ Selfdeclare + Instrument + X16.minus.17 +
##     ConsInstr + ConsNotes + Instr.minus.Notes + PachListen +
##     ClsListen + KnowRob + X1990s2000s + X1990s2000s.minus.1960s1970s +
##     CollegeMusic + NoClass + APTheory + Composing + GuitarPlay +
##     Selfdeclare:X16.minus.17 + Selfdeclare:PachListen + Selfdeclare:ClsListen +
##     Selfdeclare:X1990s2000s + Selfdeclare:X1990s2000s.minus.1960s1970s +
##     Selfdeclare:CollegeMusic + Selfdeclare:NoClass + Selfdeclare:APTheory +
##     Selfdeclare:Composing, data = musician.ratings)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2339 -1.4662 -0.0109  1.4523  5.6765
##
## Coefficients:
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                               7.44727    0.32634  22.821  < 2e-16
## Selfdeclare1                             -6.84112    0.78461  -8.719  < 2e-16
## Instrumentpiano                          -0.95353    0.09995  -9.540  < 2e-16
## Instrumentstring                         -2.52069    0.09995 -25.219  < 2e-16
## X16.minus.17                              0.10653    0.01996   5.337 1.03e-07
## ConsInstr                                 0.33506    0.04535   7.388 2.04e-13
## ConsNotes                                -0.26279    0.05153  -5.100 3.66e-07
## Instr.minus.Notes                        -0.29080    0.04919  -5.911 3.87e-09
## PachListen                               -0.50975    0.05570  -9.152  < 2e-16
## ClsListen                                -0.06293    0.04911  -1.281 0.200186
## KnowRob                                   0.24911    0.03460   7.201 7.95e-13
## X1990s2000s                               0.43833    0.05613   7.809 8.51e-15
## X1990s2000s.minus.1960s1970s             -0.22181    0.04237  -5.235 1.79e-07
## CollegeMusic1                            -0.23055    0.14803  -1.557 0.119484
```

```
## NoClass                                  -0.18484    0.05334  -3.465 0.000539
## APTheory1                                 1.06189    0.17740   5.986 2.47e-09
## Composing                                -0.50119    0.07129  -7.030 2.67e-12
## GuitarPlay                               -0.21247    0.05968  -3.560 0.000377
## Selfdeclare1:X16.minus.17                 0.19089    0.04363   4.375 1.27e-05
## Selfdeclare1:PachListen                   1.03729    0.11905   8.713  < 2e-16
## Selfdeclare1:ClsListen                    0.33992    0.08347   4.072 4.80e-05
## Selfdeclare1:X1990s2000s                 -0.36754    0.07736  -4.751 2.14e-06
## Selfdeclare1:X1990s2000s.minus.1960s1970s  0.72510    0.07781   9.319  < 2e-16
## Selfdeclare1:CollegeMusic1                1.03536    0.24024   4.310 1.70e-05
## Selfdeclare1:NoClass                      0.26444    0.08170   3.237 0.001225
## Selfdeclare1:APTheory1                   -1.34422    0.26540  -5.065 4.39e-07
## Selfdeclare1:Composing                    0.58623    0.09830   5.964 2.82e-09
##
## (Intercept)                              ***
## Selfdeclare1                             ***
## Instrumentpiano                          ***
## Instrumentstring                         ***
## X16.minus.17                             ***
## ConsInstr                                ***
## ConsNotes                                ***
## Instr.minus.Notes                        ***
## PachListen                               ***
## ClsListen
## KnowRob                                  ***
## X1990s2000s                              ***
## X1990s2000s.minus.1960s1970s             ***
## CollegeMusic1
## NoClass                                  ***
## APTheory1                                ***
## Composing                                ***
## GuitarPlay                               ***
## Selfdeclare1:X16.minus.17                ***
## Selfdeclare1:PachListen                  ***
## Selfdeclare1:ClsListen                   ***
## Selfdeclare1:X1990s2000s                 ***
## Selfdeclare1:X1990s2000s.minus.1960s1970s ***
## Selfdeclare1:CollegeMusic1               ***
## Selfdeclare1:NoClass                     **
## Selfdeclare1:APTheory1                   ***
## Selfdeclare1:Composing                   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.022 on 2426 degrees of freedom
## Multiple R-squared:  0.3203, Adjusted R-squared:  0.313
## F-statistic: 43.97 on 26 and 2426 DF,  p-value: < 2.2e-16
```

```r
# random intercept
mlm.semi <- lmer(as.formula(paste("Popular ~",
                                   paste(as.character(formula(mlm.final.fix))[3],
                                         "(1 | Subject)",
                                         sep = "+"))),
                  lmerControl(optimizer = "bobyqa"),  REML = F, data = musician.ratings)
```

```r
exactLRT(mlm.semi, mlm.final.fix)
```

```
##
##  simulated finite sample distribution of LRT. (p-value based on 10000
##  simulated values)
##
## data:
## LRT = 284.28, p-value < 2.2e-16
```

```r
summary(mlm.semi)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Popular ~ Selfdeclare + Instrument + X16.minus.17 + ConsInstr +
##     ConsNotes + Instr.minus.Notes + PachListen + ClsListen +
##     KnowRob + X1990s2000s + X1990s2000s.minus.1960s1970s + CollegeMusic +
##     NoClass + APTheory + Composing + GuitarPlay + Selfdeclare:X16.minus.17 +
##     Selfdeclare:PachListen + Selfdeclare:ClsListen + Selfdeclare:X1990s2000s +
##     Selfdeclare:X1990s2000s.minus.1960s1970s + Selfdeclare:CollegeMusic +
##     Selfdeclare:NoClass + Selfdeclare:APTheory + Selfdeclare:Composing +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10162.5  10330.9  -5052.3  10104.5     2424
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8394 -0.6592  0.0485  0.6651  3.1279
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.650    0.8062
##  Residual             3.398    1.8434
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                           Estimate Std. Error t value
## (Intercept)                                7.35247    0.81497   9.022
## Selfdeclare1                              -6.80125    1.99902  -3.402
## Instrumentpiano                           -0.95043    0.09118 -10.423
## Instrumentstring                          -2.54678    0.09124 -27.914
## X16.minus.17                               0.11438    0.04791   2.387
## ConsInstr                                  0.33495    0.11551   2.900
## ConsNotes                                 -0.24680    0.13085  -1.886
## Instr.minus.Notes                         -0.27726    0.12516  -2.215
## PachListen                                -0.49798    0.14132  -3.524
## ClsListen                                 -0.05138    0.12314  -0.417
## KnowRob                                    0.25257    0.08814   2.865
## X1990s2000s                                0.42621    0.13960   3.053
## X1990s2000s.minus.1960s1970s              -0.20659    0.10734  -1.925
## CollegeMusic1                             -0.25329    0.37759  -0.671
```

```
## NoClass                                        -0.17757    0.13629  -1.303
## APTheory1                                       1.02341    0.44953   2.277
## Composing                                      -0.49683    0.17841  -2.785
## GuitarPlay                                     -0.21433    0.15158  -1.414
## Selfdeclare1:X16.minus.17                       0.18509    0.11040   1.677
## Selfdeclare1:PachListen                         1.02495    0.30429   3.368
## Selfdeclare1:ClsListen                          0.32485    0.21124   1.538
## Selfdeclare1:X1990s2000s                       -0.35905    0.19445  -1.846
## Selfdeclare1:X1990s2000s.minus.1960s1970s       0.71537    0.19796   3.614
## Selfdeclare1:CollegeMusic1                      1.06228    0.61100   1.739
## Selfdeclare1:NoClass                            0.25251    0.20745   1.217
## Selfdeclare1:APTheory1                         -1.28252    0.66871  -1.918
## Selfdeclare1:Composing                          0.59084    0.24768   2.386
```

```r
# General
mlm.semi.2 <- lme4::lmer(Popular ~ Instrument + Harmony + Voice + X16.minus.17 + Selfdeclare +
                           ConsInstr + ConsNotes + Instr.minus.Notes +
                           PachListen + KnowRob + X1990s2000s + APTheory +
                           Composing + Selfdeclare:PachListen +
                           Selfdeclare:X1990s2000s.minus.1960s1970s +
                           Selfdeclare:Composing + (1 | Subject),
                         lmerControl(optimizer = "bobyqa"),  REML = F,
                         data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 453.06, p-value < 2.2e-16
```

```r
# Random slope
mlm.final <- ffRanefLMER.fnc(mlm.semi.2,
                             ran.effects = c("(Instrument + Voice + Harmony | Subject)"),
                             log.file = F)
```

```
## evaluating addition of (Instrument+Voice+Harmony|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      boundary (singular) fit: see ?isSingular
##  not adding (Instrument+Voice+Harmony|Subject) to model
```

```r
summary(mlm.final)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Popular ~ Instrument + Harmony + Voice + X16.minus.17 + Selfdeclare +
##     ConsInstr + ConsNotes + Instr.minus.Notes + PachListen +
##     KnowRob + X1990s2000s + APTheory + Composing + Selfdeclare:PachListen +
##     Selfdeclare:X1990s2000s.minus.1960s1970s + Selfdeclare:Composing +
##     (1 | Subject)
```

```
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10157.3  10296.6  -5054.7  10109.3     2429
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8163 -0.6403  0.0382  0.6699  3.2034
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.913    0.9555
##  Residual             3.375    1.8371
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                           Estimate Std. Error t value
## (Intercept)                               7.099008   0.740008   9.593
## Instrumentpiano                          -0.951026   0.090876 -10.465
## Instrumentstring                         -2.548543   0.090932 -28.027
## HarmonyI-V-IV                            -0.044918   0.104993  -0.428
## HarmonyI-V-VI                            -0.309657   0.104973  -2.950
## HarmonyIV-I-V                            -0.236449   0.104778  -2.257
## Voicepar3rd                               0.148537   0.090907   1.634
## Voicepar5th                               0.173435   0.090857   1.909
## X16.minus.17                              0.150862   0.048823   3.090
## Selfdeclare1                             -3.622146   1.510221  -2.398
## ConsInstr                                 0.247484   0.129651   1.909
## ConsNotes                                -0.124425   0.138134  -0.901
## Instr.minus.Notes                        -0.154414   0.137245  -1.125
## PachListen                               -0.283738   0.135447  -2.095
## KnowRob                                   0.218424   0.088887   2.457
## X1990s2000s                               0.091513   0.102086   0.896
## APTheory1                                 0.008937   0.319672   0.028
## Composing                                -0.449455   0.185290  -2.426
## Selfdeclare1:PachListen                   0.538012   0.300753   1.789
## Selfdeclare0:X1990s2000s.minus.1960s1970s -0.072373   0.094284  -0.768
## Selfdeclare1:X1990s2000s.minus.1960s1970s  0.285299   0.151773   1.880
## Selfdeclare1:Composing                    0.603921   0.232349   2.599
```

5(b) ii.

```r
# All levels of Harmony
mlm.semi.2 <- lme4::lmer(Popular ~ Harmony - 1 + Instrument + Voice + X16.minus.17 + Selfdeclare +
                           ConsInstr + ConsNotes + Instr.minus.Notes +
                           PachListen + KnowRob + X1990s2000s + APTheory +
                           Composing + Selfdeclare:PachListen +
                           Selfdeclare:X1990s2000s.minus.1960s1970s +
                           Selfdeclare:Composing + (1 | Subject),
                         lmerControl(optimizer = "bobyqa"),  REML = F,
                         data = musician.ratings)
```

```r
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 453.06, p-value < 2.2e-16
```

```r
# Random slope
mlm.final.harmony <- ffRanefLMER.fnc(mlm.semi.2,
                              ran.effects = c("(Harmony - 1 + Voice + Instrument | Subject)"),
                              log.file = F)
```

```
## evaluating addition of (Harmony-1+Voice+Instrument|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp),  :
##        convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations exceeded
##      boundary (singular) fit: see ?isSingular
##  not adding (Harmony-1+Voice+Instrument|Subject) to model
```

```r
summary(mlm.final.harmony)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Harmony - 1 + Instrument + Voice + X16.minus.17 + Selfdeclare +
##     ConsInstr + ConsNotes + Instr.minus.Notes + PachListen +
##     KnowRob + X1990s2000s + APTheory + Composing + Selfdeclare:PachListen +
##     Selfdeclare:X1990s2000s.minus.1960s1970s + Selfdeclare:Composing +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10157.3  10296.6  -5054.7  10109.3     2429
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8163 -0.6403  0.0382  0.6699  3.2034
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.913    0.9555
##  Residual             3.375    1.8371
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                           Estimate Std. Error t value
## HarmonyI-IV-V                             7.099008   0.740008   9.593
## HarmonyI-V-IV                             7.054090   0.740079   9.532
```

```
## HarmonyI-V-VI                              6.789351   0.740165   9.173
## HarmonyIV-I-V                              6.862559   0.740012   9.274
## Instrumentpiano                           -0.951026   0.090876 -10.465
## Instrumentstring                          -2.548543   0.090932 -28.027
## Voicepar3rd                                0.148537   0.090907   1.634
## Voicepar5th                                0.173435   0.090857   1.909
## X16.minus.17                               0.150862   0.048823   3.090
## Selfdeclare1                              -3.622146   1.510221  -2.398
## ConsInstr                                  0.247484   0.129651   1.909
## ConsNotes                                 -0.124425   0.138134  -0.901
## Instr.minus.Notes                         -0.154414   0.137245  -1.125
## PachListen                                -0.283738   0.135447  -2.095
## KnowRob                                    0.218424   0.088887   2.457
## X1990s2000s                                0.091513   0.102086   0.896
## APTheory1                                  0.008937   0.319672   0.028
## Composing                                 -0.449455   0.185290  -2.426
## Selfdeclare1:PachListen                    0.538012   0.300753   1.789
## Selfdeclare0:X1990s2000s.minus.1960s1970s -0.072373   0.094284  -0.768
## Selfdeclare1:X1990s2000s.minus.1960s1970s  0.285299   0.151773   1.880
## Selfdeclare1:Composing                     0.603921   0.232349   2.599
```

5(b) iii.

```
# All levels of Instrument
mlm.semi.2 <- lme4::lmer(Popular ~ Instrument - 1 + Harmony + Voice + X16.minus.17 + Selfdeclare +
                           ConsInstr + ConsNotes + Instr.minus.Notes +
                           PachListen + KnowRob + X1990s2000s + APTheory +
                           Composing + Selfdeclare:PachListen +
                           Selfdeclare:X1990s2000s.minus.1960s1970s +
                           Selfdeclare:Composing + (1 | Subject),
                         lmerControl(optimizer = "bobyqa"),  REML = F,
                         data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 453.06, p-value < 2.2e-16
```

```
# Random slope
mlm.final.instr <- ffRanefLMER.fnc(mlm.semi.2,
                              ran.effects = c("(Instrument - 1 + Voice + Harmony | Subject)"),
                              log.file = F)
```

```
## evaluating addition of (Instrument-1+Voice+Harmony|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##       maxfun < 10 * length(par)^2 is not recommended.
##     boundary (singular) fit: see ?isSingular
##  not adding (Instrument-1+Voice+Harmony|Subject) to model
```

```r
summary(mlm.final.instr)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Instrument - 1 + Harmony + Voice + X16.minus.17 + Selfdeclare +
##     ConsInstr + ConsNotes + Instr.minus.Notes + PachListen +
##     KnowRob + X1990s2000s + APTheory + Composing + Selfdeclare:PachListen +
##     Selfdeclare:X1990s2000s.minus.1960s1970s + Selfdeclare:Composing +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10157.3  10296.6  -5054.7  10109.3     2429
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8163 -0.6403  0.0382  0.6699  3.2034
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.913    0.9555
##  Residual             3.375    1.8371
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                           Estimate Std. Error t value
## Instrumentguitar                          7.099008   0.740008   9.593
## Instrumentpiano                           6.147982   0.739998   8.308
## Instrumentstring                          4.550465   0.740412   6.146
## HarmonyI-V-IV                            -0.044918   0.104993  -0.428
## HarmonyI-V-VI                            -0.309657   0.104973  -2.950
## HarmonyIV-I-V                            -0.236449   0.104778  -2.257
## Voicepar3rd                               0.148537   0.090907   1.634
## Voicepar5th                               0.173435   0.090857   1.909
## X16.minus.17                              0.150862   0.048823   3.090
## Selfdeclare1                             -3.622146   1.510221  -2.398
## ConsInstr                                 0.247484   0.129651   1.909
## ConsNotes                                -0.124425   0.138134  -0.901
## Instr.minus.Notes                        -0.154414   0.137245  -1.125
## PachListen                               -0.283738   0.135447  -2.095
## KnowRob                                   0.218424   0.088887   2.457
## X1990s2000s                               0.091513   0.102086   0.896
## APTheory1                                 0.008937   0.319672   0.028
## Composing                                -0.449455   0.185290  -2.426
## Selfdeclare1:PachListen                   0.538012   0.300753   1.789
## Selfdeclare0:X1990s2000s.minus.1960s1970s -0.072373   0.094284  -0.768
## Selfdeclare1:X1990s2000s.minus.1960s1970s  0.285299   0.151773   1.880
## Selfdeclare1:Composing                    0.603921   0.232349   2.599
```

5(b) vi.

```r
# All levels of Voice
mlm.semi.2 <- lme4::lmer(Popular ~ Voice - 1 + Instrument + Harmony + X16.minus.17 + Selfdeclare +
                           ConsInstr + ConsNotes + Instr.minus.Notes +
                           PachListen + KnowRob + X1990s2000s + APTheory +
                           Composing + Selfdeclare:PachListen +
                           Selfdeclare:X1990s2000s.minus.1960s1970s +
                           Selfdeclare:Composing + (1 | Subject),
                         lmerControl(optimizer = "bobyqa"),  REML = F,
                         data = musician.ratings)
exactRLRT(mlm.semi.2)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 453.06, p-value < 2.2e-16
```

```r
# Random slope
mlm.final.voice <- ffRanefLMER.fnc(mlm.semi.2,
                                   ran.effects = c("(Voice - 1 + Harmony + Instrument | Subject)"),
                                   log.file = F)
```

```
## evaluating addition of (Voice-1+Harmony+Instrument|Subject) to model
##  Warning in commonArgs(par, fn, control, environment()) :
##        maxfun < 10 * length(par)^2 is not recommended.
##      Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp),  :
##        convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations exceeded
##      boundary (singular) fit: see ?isSingular
##  not adding (Voice-1+Harmony+Instrument|Subject) to model
```

```r
summary(mlm.final.voice)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Voice - 1 + Instrument + Harmony + X16.minus.17 + Selfdeclare +
##     ConsInstr + ConsNotes + Instr.minus.Notes + PachListen +
##     KnowRob + X1990s2000s + APTheory + Composing + Selfdeclare:PachListen +
##     Selfdeclare:X1990s2000s.minus.1960s1970s + Selfdeclare:Composing +
##     (1 | Subject)
##    Data: musician.ratings
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10157.3  10296.6  -5054.7  10109.3     2429
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.8163 -0.6403  0.0382  0.6699  3.2034
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Subject  (Intercept) 0.913    0.9555
##  Residual             3.375    1.8371
## Number of obs: 2453, groups:  Subject, 70
##
## Fixed effects:
##                                           Estimate Std. Error t value
## Voicecontrary                             7.099008   0.740008   9.593
## Voicepar3rd                               7.247545   0.739981   9.794
## Voicepar5th                               7.272443   0.740098   9.826
## Instrumentpiano                          -0.951026   0.090876 -10.465
## Instrumentstring                         -2.548543   0.090932 -28.027
## HarmonyI-V-IV                            -0.044918   0.104993  -0.428
## HarmonyI-V-VI                            -0.309657   0.104973  -2.950
## HarmonyIV-I-V                            -0.236449   0.104778  -2.257
## X16.minus.17                              0.150862   0.048823   3.090
## Selfdeclare1                             -3.622146   1.510221  -2.398
## ConsInstr                                 0.247484   0.129651   1.909
## ConsNotes                                -0.124425   0.138134  -0.901
## Instr.minus.Notes                        -0.154414   0.137245  -1.125
## PachListen                               -0.283738   0.135447  -2.095
## KnowRob                                   0.218424   0.088887   2.457
## X1990s2000s                               0.091513   0.102086   0.896
## APTheory1                                 0.008937   0.319672   0.028
## Composing                                -0.449455   0.185290  -2.426
## Selfdeclare1:PachListen                   0.538012   0.300753   1.789
## Selfdeclare0:X1990s2000s.minus.1960s1970s -0.072373   0.094284  -0.768
## Selfdeclare1:X1990s2000s.minus.1960s1970s  0.285299   0.151773   1.880
## Selfdeclare1:Composing                    0.603921   0.232349   2.599
```