

Code Appendix

Gunho Jung

12/08/2019

Loading the Data

```
ratings <- read.csv("ratings.csv", header = TRUE)
```

Loading all the Packages

```
library(arm)
library(MASS)
library(car)
library(marginalmodelplots)
library(lme4)
library(RLRsim)
library(LMERConvenienceFunctions)
source("residual-functions.r")
```

Data Cleaning / Imputation & EDA

Data Cleaning / Imputation

```
na_response <- which(is.na(ratings$Classical) == TRUE & is.na(ratings$Popular) == TRUE)
ratings_df <- ratings[-c(na_response), ]
extreme_rating_ind_cla <- which(ratings_df$Classical > 10)
extreme_rating_ind_pop <- which(ratings_df$Popular > 10)
ratings_df$Classical[extreme_rating_ind_cla] <- 10
ratings_df$Popular[extreme_rating_ind_pop] <- 10
ratings_df <- ratings_df[, -c(1, 11, 26)]
ratings_df <- ratings_df[, -c(22, 23)]
for (i in 1:ncol(ratings_df)) {
  if (length(which(is.na(ratings_df[, i]) == TRUE)) > 0) {
    na_ind <- which(is.na(ratings_df[, i]) == TRUE)
    median_val <- median(as.numeric(ratings_df[, i]), na.rm = TRUE)
    ratings_df[na_ind, i] = median_val
  }
}
ratings_df_cont <- ratings_df[, c("Classical", "Popular", "OMSI", "NoClass")]
ratings_df_cat <- ratings_df[, -c(6, 17, 22, 23)]
```

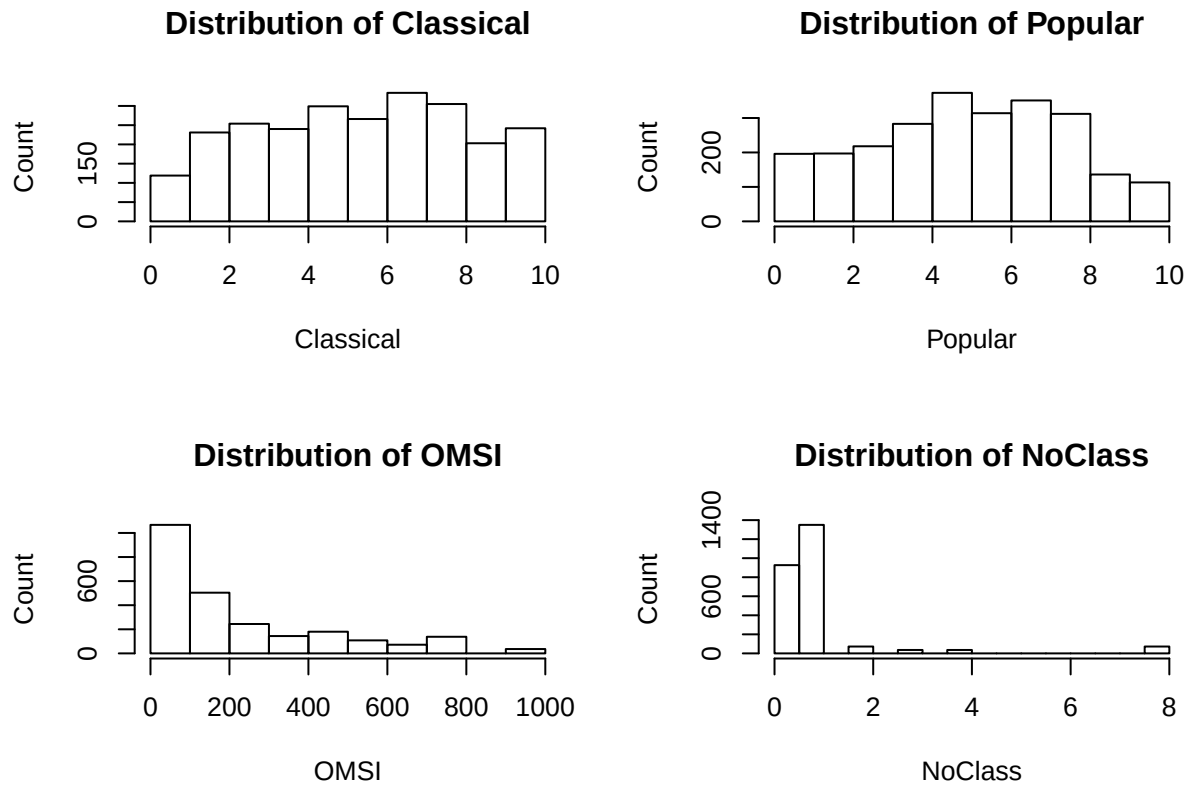
Univariate EDA

```
par(mfrow = c(2, 2))
for(i in 1:ncol(ratings_df_cont)) {
  hist(ratings_df_cont[, i], xlab = colnames(ratings_df_cont)[i],
```

```

    ylab = "Count", main = paste("Distribution of", colnames(ratings_df_cont)[i]))
  }

```



```

ratings_df$OMSI <- log(ratings_df$OMSI)
colnames(ratings_df)[6] <- "LogOMSI"

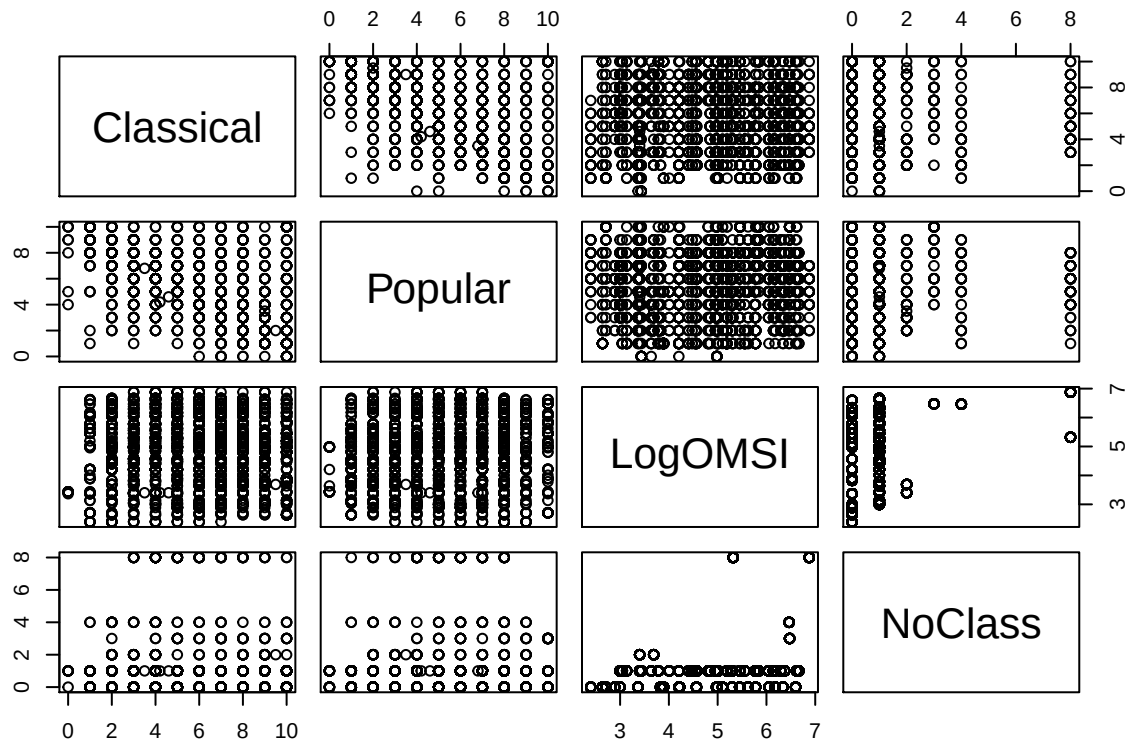
```

Multivariate EDA

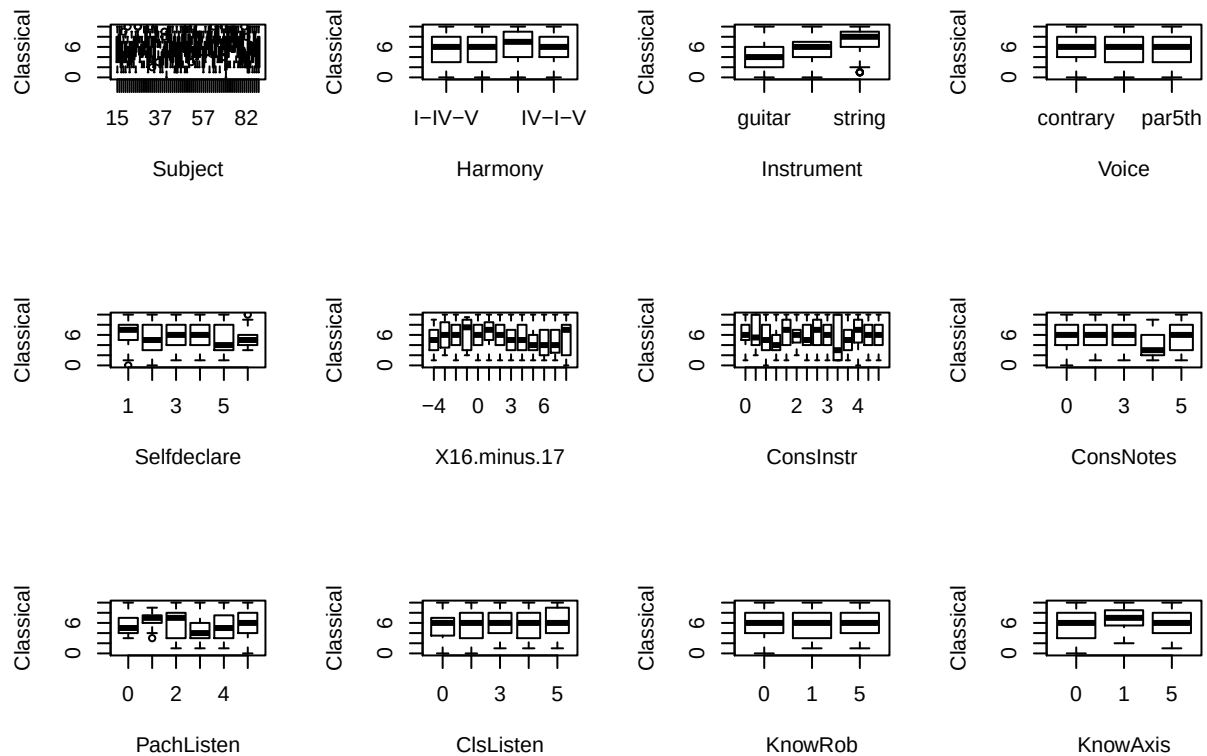
```

ratings_df_cont <- ratings_df[, c("Classical", "Popular", "LogOMSI", "NoClass")]
# Multivariate EDA on Continuous Variables
pairs(ratings_df_cont)

```

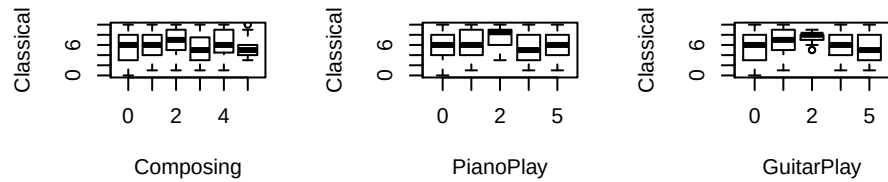
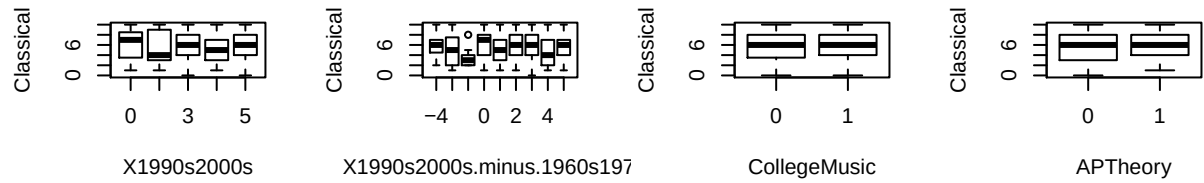


```
# Classical vs Categorical Variables
par(mfrow = c(3, 4))
for (i in 1:ncol(ratings_df_cat)) {
  plot(x = as.factor(ratings_df_cat[, i]), y = ratings_df$Classical,
       xlab = colnames(ratings_df_cat)[i], ylab = "Classical")
}
```

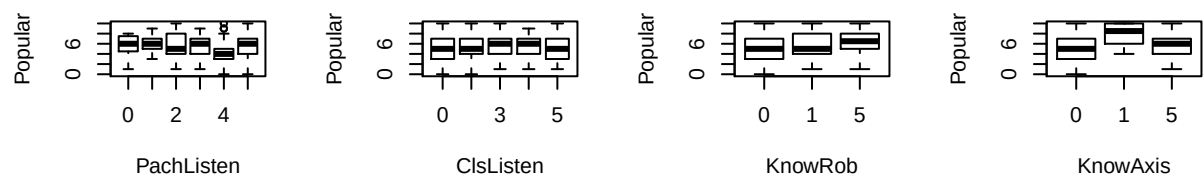
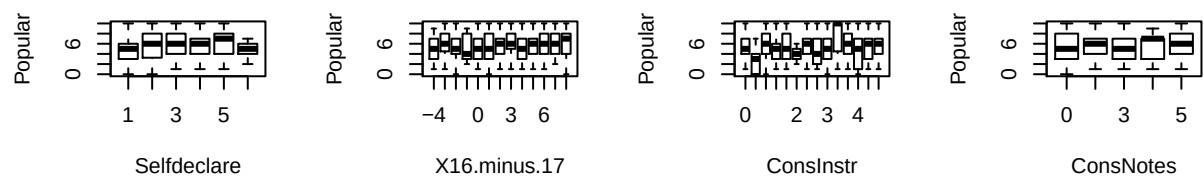
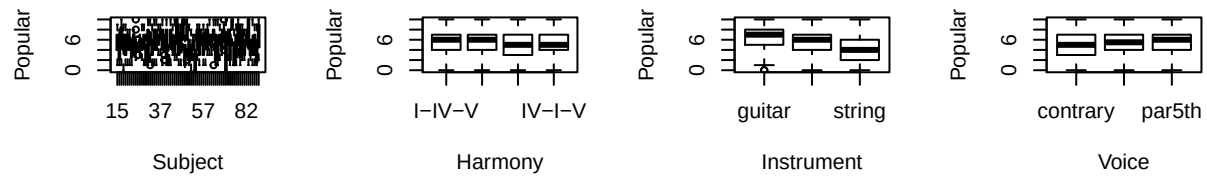


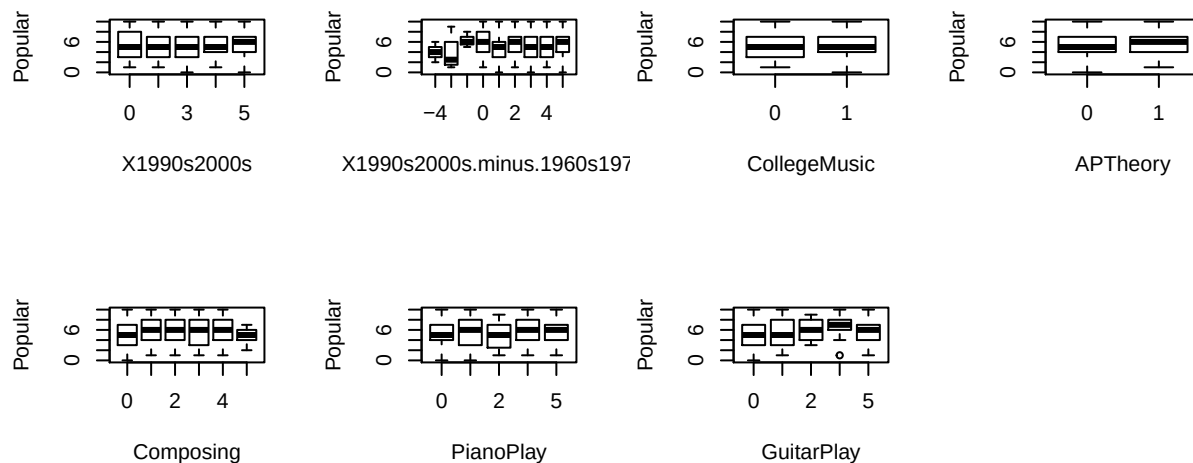
```
# Popular vs Categorical Variables
```

```
par(mfrow = c(3, 4))
```



```
for (i in 1:ncol(ratings_df_cat)) {  
  plot(x = as.factor(ratings_df_cat[, i]), y = ratings_df$Popular,  
       xlab = colnames(ratings_df_cat)[i], ylab = "Popular")  
}
```





Classical Ratings

Experimental Factors with Interaction

```
mod_a <- lm(Classical ~ Instrument * Harmony * Voice, data = ratings_df)
summary(mod_a)
```

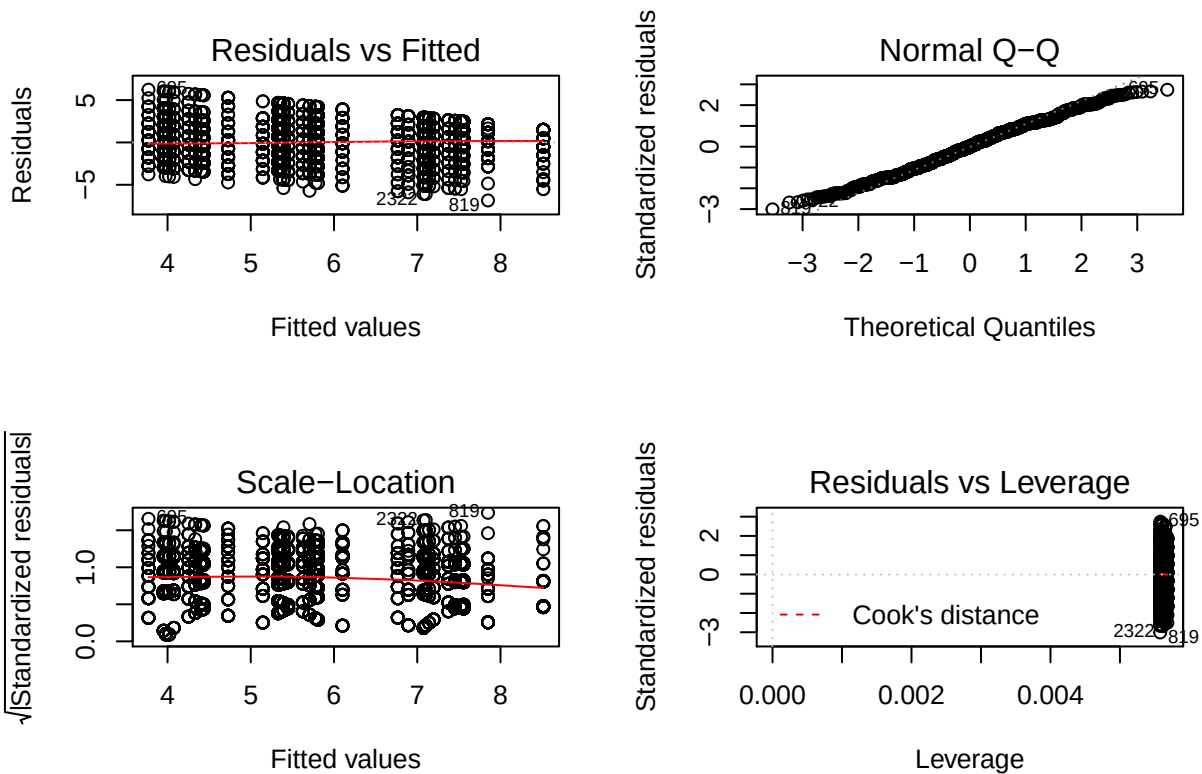
Experimental Factors Variable Selection

```
stepAIC(mod_a, direction = "both", k = 2)
stepAIC(mod_a, direction = "both", k = log(nrow(ratings_df)))
```

```
mod_a_full <- lm(Classical ~ Instrument + Harmony + Voice + Harmony:Voice,
  data = ratings_df)
mod_a_red <- lm(Classical ~ Instrument + Harmony + Voice, data = ratings_df)
anova(mod_a_red, mod_a_full)
summary(mod_a_full)
```

Model Diagnostics of the selected model

```
par(mfrow = c(2, 2))
plot(mod_a_full)
```



```
dev.off()
marginalModelPlots(mod_a_full)
vif(mod_a_full)
```

Deciding Random Effects

```
mod_b <- lmer(Classical ~ Instrument + Harmony + Voice + Harmony:Voice +
              (1|Subject), data = ratings_df, REML = FALSE,
              control = lmerControl(optimizer = "bobyqa"))
exactRLRT(mod_b)
AIC(mod_a_full)
AIC(mod_b)
BIC(mod_a_full)
BIC(mod_b)
display(mod_b)
```

Improving Random Effects

```
mod_c_i <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
                (1|Subject), data = ratings_df, REML = FALSE,
                control = lmerControl(optimizer = "bobyqa"))

mod_c_ii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
                  (Instrument|Subject), data = ratings_df, REML = FALSE,
                  control = lmerControl(optimizer = "bobyqa"))

mod_c_iii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
                   (Harmony|Subject), data = ratings_df, REML = FALSE,
                   control = lmerControl(optimizer = "bobyqa"))

mod_c_iv <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
```

```

      (Voice|Subject), data = ratings_df, REML = FALSE,
      control = lmerControl(optimizer = "bobyqa"))

mod_c_v <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument + Harmony|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_vi <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument + Voice|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_vii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Harmony + Voice|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_viii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument + Harmony + Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_c_ix <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument|Subject) + (Harmony|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_c_x <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument|Subject) + (Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_c_xi <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Harmony|Subject) + (Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_c_xii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument|Subject) + (Harmony|Subject) + (Voice|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_xiii <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument + Harmony | Subject) + (Voice | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_xiv <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Harmony + Voice | Subject) + (Instrument | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_c_xv <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
  (Instrument + Voice | Subject) + (Harmony | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

anova(mod_c_i, mod_c_ii, mod_c_iii, mod_c_iv, mod_c_v, mod_c_vi, mod_c_vii,
  mod_c_viii, mod_c_ix, mod_c_x, mod_c_xi, mod_c_xii, mod_c_xiii, mod_c_xiv,

```

```
mod_c_xv)

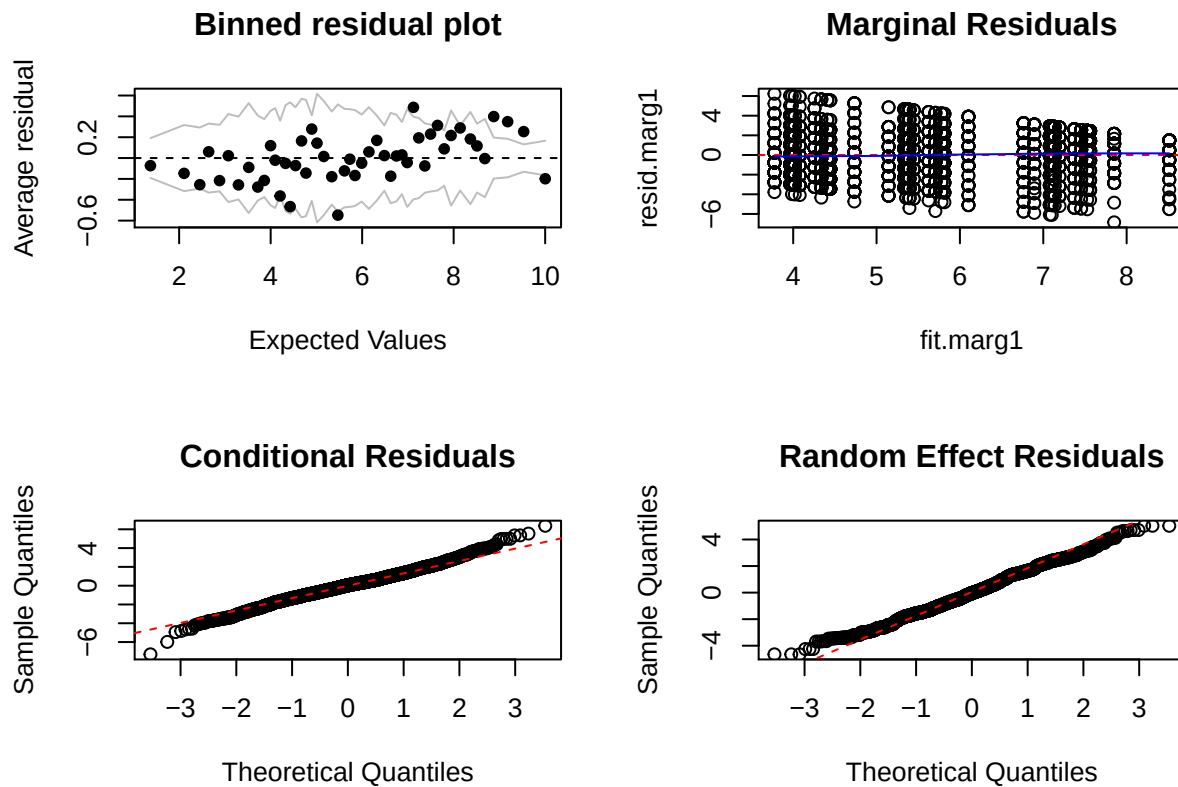
anova(mod_c_ii, mod_c_v)
```

Model Summary of the improved model

```
summary(mod_c_v)
```

Model Diagnostics of Improved Model

```
resid.marg1 <- r.marg(mod_c_v)
resid.cond1 <- r.cond(mod_c_v)
resid.reff1 <- r.reff(mod_c_v)
fit.marg1 <- yhat.marg(mod_c_v)
par(mfrow = c(2, 2))
binnedplot(x = predict(mod_c_v), y = resid(mod_c_v))
plot(fit.marg1, resid.marg1, main = "Marginal Residuals")
abline(h = 0, col = "red", lty = "dashed")
lines(loess.smooth(fit.marg1, resid.marg1), col = "blue")
qqnorm(resid.cond1, main = "Conditional Residuals")
qqline(resid.cond1, col = "red", lty = "dashed")
qqnorm(resid.reff1, main = "Random Effect Residuals")
qqline(resid.reff1, col = "red", lty = "dashed")
```



Pachelbel Rants / Comedy Bits vs Harmony

```
mod_v_pachelbel <- lmer(Classical ~ 1 + Instrument + Harmony + Voice + KnowRob +
  KnowAxis + Harmony:Voice + Harmony:KnowRob +
  Harmony:KnowAxis + (Instrument + Harmony|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))
```



```
summary(mod_v_pachelbel)
```

Additional Driving Factors

```
# Factoring Variables to be categorical from numeric
ratings_df$Selfdeclare <- as.factor(ratings_df$Selfdeclare)
ratings_df$X16.minus.17 <- as.factor(ratings_df$X16.minus.17)
ratings_df$ConsInstr <- as.factor(ratings_df$ConsInstr)
ratings_df$ConsNotes <- as.factor(ratings_df$ConsNotes)
ratings_df$PachListen <- as.factor(ratings_df$PachListen)
ratings_df$ClsListen <- as.factor(ratings_df$ClsListen)
ratings_df$KnowRob <- as.factor(ratings_df$KnowRob)
ratings_df$KnowAxis <- as.factor(ratings_df$KnowAxis)
ratings_df$X1990s2000s <- as.factor(ratings_df$X1990s2000s)
ratings_df$X1990s2000s.minus.1960s1970s <- as.factor(ratings_df$X1990s2000s.minus.1960s1970s)
ratings_df$CollegeMusic <- as.factor(ratings_df$CollegeMusic)
ratings_df$APTheory <- as.factor(ratings_df$APTheory)
ratings_df$Composing <- as.factor(ratings_df$Composing)
ratings_df$PianoPlay <- as.factor(ratings_df$PianoPlay)
ratings_df$GuitarPlay <- as.factor(ratings_df$GuitarPlay)

# Individual Covariates Selection
mod_full_lm <- lm(Classical ~ . + Harmony:Voice - Popular - Subject, data = ratings_df)
stepAIC(mod_full_lm, k = 2, direction = "both")
stepAIC(mod_full_lm, k = log(nrow(ratings_df)), direction = "both")

mod_full_aic_bic <- lmer(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
  X16.minus.17 + ConsInstr + PachListen + ClsListen +
  KnowRob + KnowAxis + X1990s2000s.minus.1960s1970s +
  APTheory + Composing + PianoPlay + Harmony:Voice +
  (1|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))
```

Random Effects Selection with additional Driving Factors

```
lmer_full_aic <- fitLMER.fnc(mod_full_aic_bic,
  ran.effects = c("(Instrument|Subject)",
    "(Harmony|Subject)",
    "(Voice|Subject)"), method = "AIC",
  keep.single.factors = TRUE)

lmer_full_bic <- fitLMER.fnc(mod_full_aic_bic,
  ran.effects = c("(Instrument|Subject)",
    "(Harmony|Subject)",
    "(Voice|Subject)"), method = "BIC",
  keep.single.factors = TRUE)

summary(lmer_full_aic)

summary(lmer_full_bic)

lmer_full_aic2 <- lmer(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
  X16.minus.17 + ConsInstr + PachListen + ClsListen +
  KnowRob + KnowAxis + X1990s2000s.minus.1960s1970s +
  APTheory + Composing + PianoPlay +
```

```

      (Instrument + Harmony | Subject) + Harmony:Voice,
data = ratings_df, REML = FALSE,
control = lmerControl(optimizer = "bobyqa"))

anova(lmer_full_aic2, lmer_full_aic)

mod_3 <- lmer_full_aic2
summary(mod_3)

```

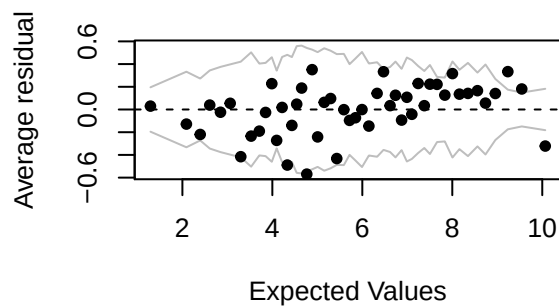
Model Diagnostics on the new model

```

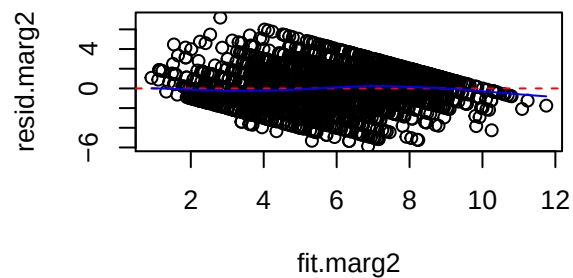
resid.marg2 <- r.marg(mod_3)
resid.cond2 <- r.cond(mod_3)
resid.reff2 <- r.reff(mod_3)
fit.marg2 <- yhat.marg(mod_3)
par(mfrow = c(2, 2))
binnedplot(x = predict(mod_3), y = resid(mod_3))
plot(fit.marg2, resid.marg2, main = "Marginal Residuals")
abline(h = 0, col = "red", lty = "dashed")
lines(loess.smooth(fit.marg2, resid.marg2), col = "blue")
qqnorm(resid.cond2, main = "Conditional Residuals")
qqline(resid.cond2, col = "red", lty = "dashed")
qqnorm(resid.reff2, main = "Random Effect Residuals")
qqline(resid.reff2, col = "red", lty = "dashed")

```

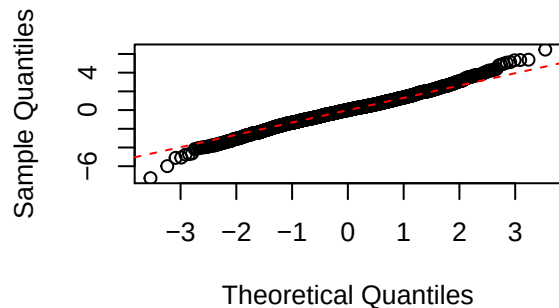
Binned residual plot



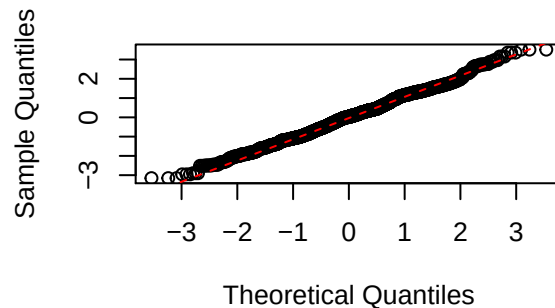
Marginal Residuals



Conditional Residuals



Random Effect Residuals



Popular Ratings

Experimental Factors and Variable Selection

```
mod_5_lm <- lm(Popular ~ Instrument * Harmony * Voice, data = ratings_df)
stepAIC(mod_5_lm, direction = "both", k = 2)
stepAIC(mod_5_lm, direction = "both", k = log(nrow(ratings_df)))
mod_5_aic <- lm(Popular ~ Instrument + Harmony, data = ratings_df)
mod_5_bic <- lm(Popular ~ Instrument, data = ratings_df)
anova(mod_5_bic, mod_5_aic)
```

Deciding Random Effects

```
mod_5_full_lm <- lm(Popular ~ Instrument + Harmony + Voice, data = ratings_df)
mod_5_ran_eff <- lmer(Popular ~ Instrument + Harmony + Voice + (1|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))
exactRLRT(mod_5_ran_eff)
AIC(mod_5_full_lm)
AIC(mod_5_ran_eff)
BIC(mod_5_full_lm)
BIC(mod_5_ran_eff)
```

Improving Random Effects

```
mod_5_ran_eff1 <- lmer(Popular ~ Instrument + Harmony + Voice + (1|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff2 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff3 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Harmony|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff4 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice + (Voice|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff5 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument + Harmony|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff6 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument + Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff7 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Harmony + Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff8 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument + Harmony + Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))
```

```

mod_5_ran_eff9 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument|Subject) + (Harmony|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff10 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument|Subject) + (Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff11 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Harmony|Subject) + (Voice|Subject), data = ratings_df,
  REML = FALSE, control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff12 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument|Subject) + (Harmony|Subject) +
  (Voice|Subject), data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff13 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument + Harmony | Subject) + (Voice | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff14 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Harmony + Voice | Subject) + (Instrument | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

mod_5_ran_eff15 <- lmer(Popular ~ 1 + Instrument + Harmony + Voice +
  (Instrument + Voice | Subject) + (Harmony | Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))

anova(mod_5_ran_eff1, mod_5_ran_eff2, mod_5_ran_eff3, mod_5_ran_eff4, mod_5_ran_eff5,
  mod_5_ran_eff6, mod_5_ran_eff7, mod_5_ran_eff8, mod_5_ran_eff9, mod_5_ran_eff10,
  mod_5_ran_eff11, mod_5_ran_eff12, mod_5_ran_eff13, mod_5_ran_eff14,
  mod_5_ran_eff15)

summary(mod_5_ran_eff5)

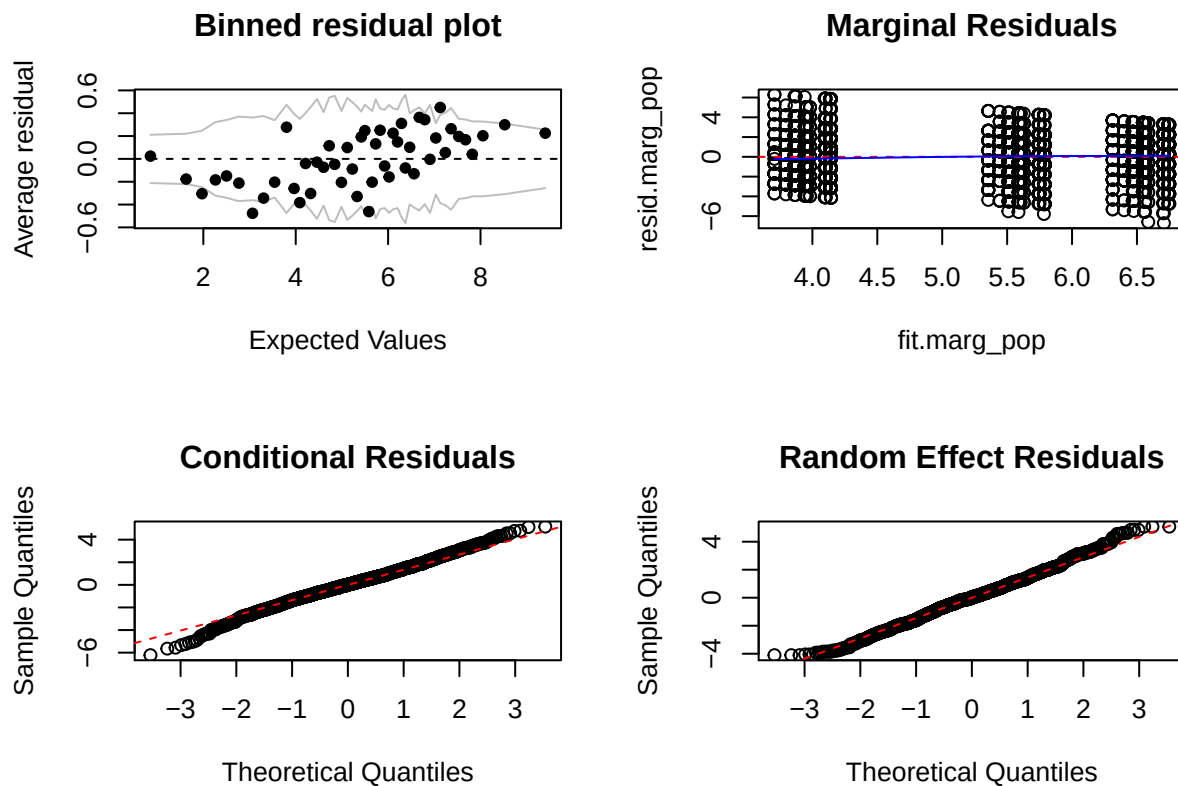
```

Model Diagnostics on the selected model

```

resid.marg_pop <- r.marg(mod_5_ran_eff5)
resid.cond_pop <- r.cond(mod_5_ran_eff5)
resid.reff_pop <- r.reff(mod_5_ran_eff5)
fit.marg_pop <- yhat.marg(mod_5_ran_eff5)
par(mfrow = c(2, 2))
binnedplot(x = predict(mod_5_ran_eff5), y = resid(mod_5_ran_eff5))
plot(fit.marg_pop, resid.marg_pop, main = "Marginal Residuals")
abline(h = 0, col = "red", lty = "dashed")
lines(loess.smooth(fit.marg_pop, resid.marg_pop), col = "blue")
qqnorm(resid.cond_pop, main = "Conditional Residuals")
qqline(resid.cond_pop, col = "red", lty = "dashed")
qqnorm(resid.reff_pop, main = "Random Effect Residuals")
qqline(resid.reff_pop, col = "red", lty = "dashed")

```



Additional Driving Factors Selection

```
# Individual Covariates Selection for Popular
mod_full_lm_pop <- lm(Popular ~ . - Classical - Subject, data = ratings_df)
stepAIC(mod_full_lm_pop, k = 2, direction = "both")
stepAIC(mod_full_lm_pop, k = log(nrow(ratings_df)), direction = "both")

mod_full_aic_bic_pop <- lmer(Popular ~ Instrument + Harmony + Voice + Selfdeclare +
  LogOMSI + X16.minus.17 + ConsInstr + ConsNotes +
  PachListen + KnowAxis + X1990s2000s.minus.1960s1970s +
  CollegeMusic + PianoPlay +(1|Subject),
  data = ratings_df, REML = FALSE,
  control = lmerControl(optimizer = "bobyqa"))
```

Improving Random Effects with new driving factors

```
lmer_full_aic_pop <- fitLMER.fnc(mod_full_aic_bic_pop,
  ran.effects = c("(Instrument|Subject)",
    "(Harmony|Subject)",
    "(Voice|Subject)"), method = "AIC",
  keep.single.factors = TRUE)

lmer_full_bic_pop <- fitLMER.fnc(mod_full_aic_bic_pop,
  ran.effects = c("(Instrument|Subject)",
    "(Harmony|Subject)",
    "(Voice|Subject)"), method = "BIC",
  keep.single.factors = TRUE)

summary(lmer_full_aic_pop)
```

```
summary(lmer_full_bic_pop)

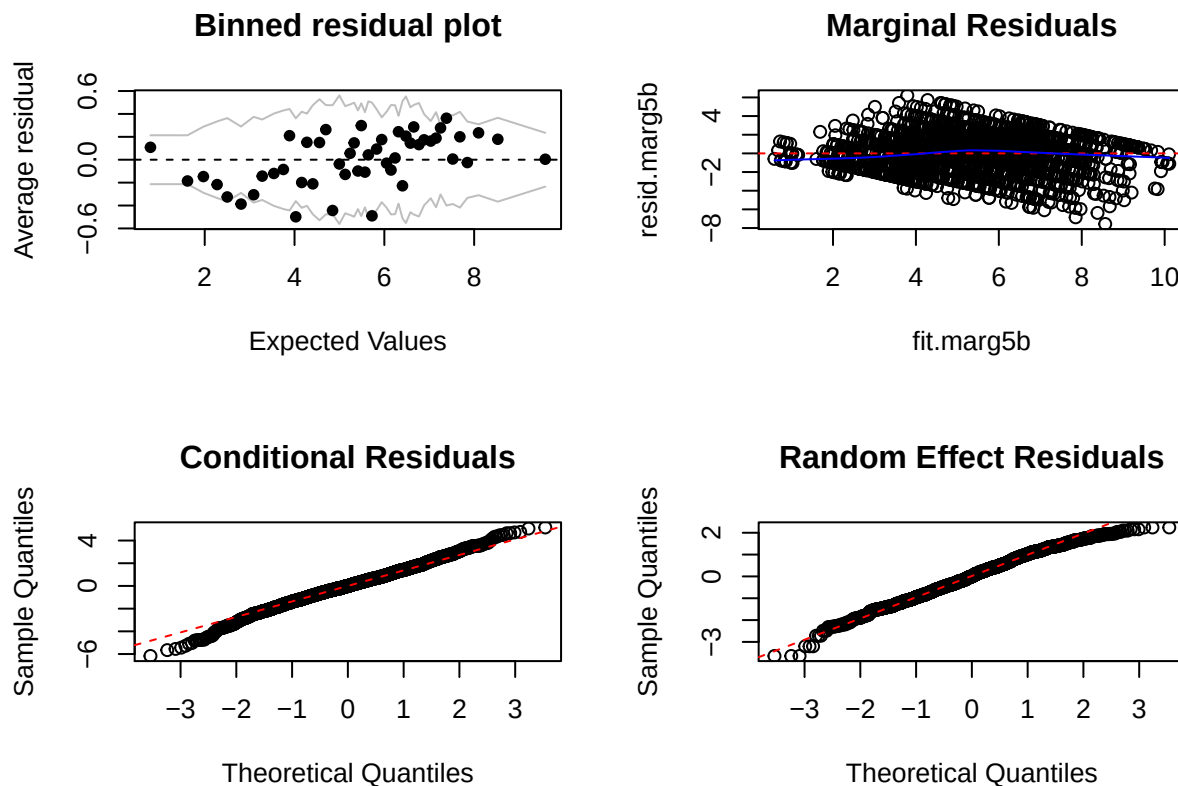
lmer_full_aic_bic_pop2 <- lmer(Popular ~ Instrument + Harmony + Voice + Selfdeclare +
                               LogOMSI + X16.minus.17 + ConsInstr + ConsNotes +
                               PachListen + KnowAxis + X1990s2000s.minus.1960s1970s +
                               CollegeMusic + PianoPlay +
                               (Instrument + Harmony|Subject), data = ratings_df,
                               REML = FALSE,
                               control = lmerControl(optimizer = "bobyqa"))

anova(lmer_full_aic_bic_pop2, lmer_full_aic_pop)

mod_5b <- lmer_full_aic_bic_pop2
summary(mod_5b)
```

Model Diagnostics of the newly selected model

```
resid.marg5b <- r.marg(mod_5b)
resid.cond5b <- r.cond(mod_5b)
resid.reff5b <- r.reff(mod_5b)
fit.marg5b <- yhat.marg(mod_5b)
# Plotting Fitted Values vs Marginal Residuals
par(mfrow = c(2, 2))
binnedplot(x = predict(mod_5b), y = resid(mod_5b))
plot(fit.marg5b, resid.marg5b, main = "Marginal Residuals")
abline(h = 0, col = "red", lty = "dashed")
lines(loess.smooth(fit.marg5b, resid.marg5b), col = "blue")
qqnorm(resid.cond5b, main = "Conditional Residuals")
qqline(resid.cond5b, col = "red", lty = "dashed")
qqnorm(resid.reff5b, main = "Random Effect Residuals")
qqline(resid.reff5b, col = "red", lty = "dashed")
```



Differences between Musicians and Non-musicians for Classical Music

Two Different Dichotomizations on Selfdeclare

```
ratings_df_dich <- ratings_df
ratings_df_dich$Selfdeclare <- as.numeric(ratings_df_dich$Selfdeclare)
ratings_df_dich$Selfdeclare <- ifelse(ratings_df_dich$Selfdeclare > 2, 1, 0)
ratings_df_dich$Selfdeclare <- as.factor(ratings_df_dich$Selfdeclare)

ratings_df_dich2 <- ratings_df
ratings_df_dich2$Selfdeclare <- as.numeric(ratings_df_dich2$Selfdeclare)
ratings_df_dich2$Selfdeclare <- ifelse(ratings_df_dich2$Selfdeclare > 3, 1, 0)
ratings_df_dich2$Selfdeclare <- as.factor(ratings_df_dich2$Selfdeclare)
```

Musicians vs Non-Musicians Model Selection

```
# Variable Selection for the Interaction Term
mod_4_dich_lm <- lm(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
  X16.minus.17 + ConsInstr + PachListen + ClsListen + KnowRob +
  KnowAxis + X1990s2000s.minus.1960s1970s + APTheory +
  Composing + PianoPlay + Harmony:Selfdeclare +
  Instrument:Selfdeclare + Voice:Selfdeclare +
  X16.minus.17:Selfdeclare + ConsInstr:Selfdeclare +
  PachListen:Selfdeclare + ClsListen:Selfdeclare +
  KnowRob:Selfdeclare + KnowAxis:Selfdeclare +
  X1990s2000s.minus.1960s1970s:Selfdeclare +
  APTheory:Selfdeclare + Composing:Selfdeclare +
```

```

PianoPlay:Selfdeclare + Harmony:Voice, data = ratings_df_dich)

mod_4_dich2_lm <- lm(Classical ~ Harmony + Instrument + Voice + Selfdeclare +
  X16.minus.17 + ConsInstr + PachListen + ClsListen + KnowRob +
  KnowAxis + X1990s2000s.minus.1960s1970s + APTheory +
  Composing + PianoPlay + Harmony:Selfdeclare +
  Instrument:Selfdeclare + Voice:Selfdeclare +
  X16.minus.17:Selfdeclare + ConsInstr:Selfdeclare +
  PachListen:Selfdeclare + ClsListen:Selfdeclare +
  KnowRob:Selfdeclare + KnowAxis:Selfdeclare +
  X1990s2000s.minus.1960s1970s:Selfdeclare +
  APTheory:Selfdeclare + Composing:Selfdeclare +
  PianoPlay:Selfdeclare + Harmony:Voice,
  data = ratings_df_dich2)

stepAIC(mod_4_dich_lm, k = 2, direction = "both")

stepAIC(mod_4_dich2_lm, k = 2, direction = "both")

mod_4_dich_lm_final <- lm(Classical ~ Instrument + Harmony + Voice + Harmony:Voice +
  Selfdeclare + X16.minus.17 + ConsInstr + PachListen +
  ClsListen + KnowRob + KnowAxis + X1990s2000s.minus.1960s1970s +
  APTheory + Composing + PianoPlay + Instrument:Selfdeclare +
  Harmony:Selfdeclare + Selfdeclare:LogOMSI +
  Selfdeclare:X16.minus.17 + Selfdeclare:ConsInstr +
  Selfdeclare:ConsNotes + Selfdeclare:KnowAxis +
  Selfdeclare:PachListen, data = ratings_df_dich)

mod_4_dich2_lm_final <- lm(Classical ~ Instrument + Harmony + Voice + Harmony:Voice +
  Selfdeclare + X16.minus.17 + ConsInstr + PachListen +
  ClsListen + KnowRob + KnowAxis + X1990s2000s.minus.1960s1970s +
  APTheory + Composing + PianoPlay + Instrument:Selfdeclare +
  Harmony:Selfdeclare + Selfdeclare:LogOMSI +
  Selfdeclare:X16.minus.17 + Selfdeclare:ConsInstr +
  Selfdeclare:PachListen, data = ratings_df_dich2)

summary(mod_4_dich_lm_final)
summary(mod_4_dich2_lm_final)

```