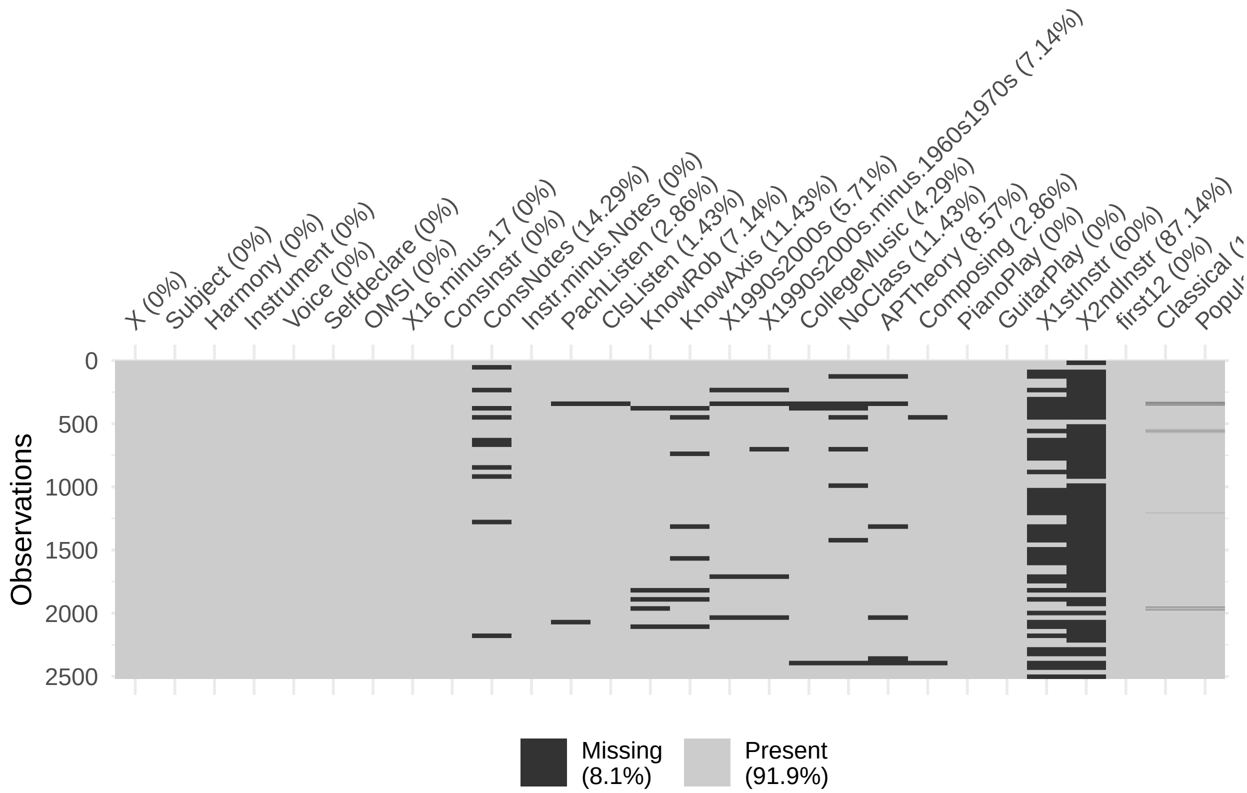
Appendix

Mitchell Pudil

11/30/2019

EDA and Variable Transformations

Let's begin by looking at the missing data



We notice there are several columns that have missing data.

A couple of the variables we will be using throughout the study are KnowRob and PachListen. However, both of these variables have many missing data points. However, nobody has both missing. We can exploit these observations and fill in the missing data by the average of the levels in the other.

First, determine the relationship between the two variables

```r
summary(lm(ratings$KnowRob ~ ratings$PachListen))
```

```
##
## Call:
## lm(formula = ratings$KnowRob ~ ratings$PachListen)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.9308 -0.9308 -0.9308  0.0692  4.0692 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)         -0.50943    0.15245  -3.342 0.000846 ***
## ratings$PachListen  0.28805    0.03275   8.795  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.71 on 2266 degrees of freedom
##   (252 observations deleted due to missingness)
## Multiple R-squared:  0.03301,    Adjusted R-squared:  0.03258 
## F-statistic: 77.35 on 1 and 2266 DF,  p-value: < 2.2e-16
```

```
summary(lm(ratings$PachListen ~ ratings$KnowRob))
```

```
## 
## Call:
## lm(formula = ratings$PachListen ~ ratings$KnowRob)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.4329 -0.0058  0.5671  0.5671  0.5671 
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      4.43286    0.02490 178.057   <2e-16 ***
## ratings$KnowRob  0.11460    0.01303   8.795   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.079 on 2266 degrees of freedom
##   (252 observations deleted due to missingness)
## Multiple R-squared:  0.03301,    Adjusted R-squared:  0.03258 
## F-statistic: 77.35 on 1 and 2266 DF,  p-value: < 2.2e-16
```

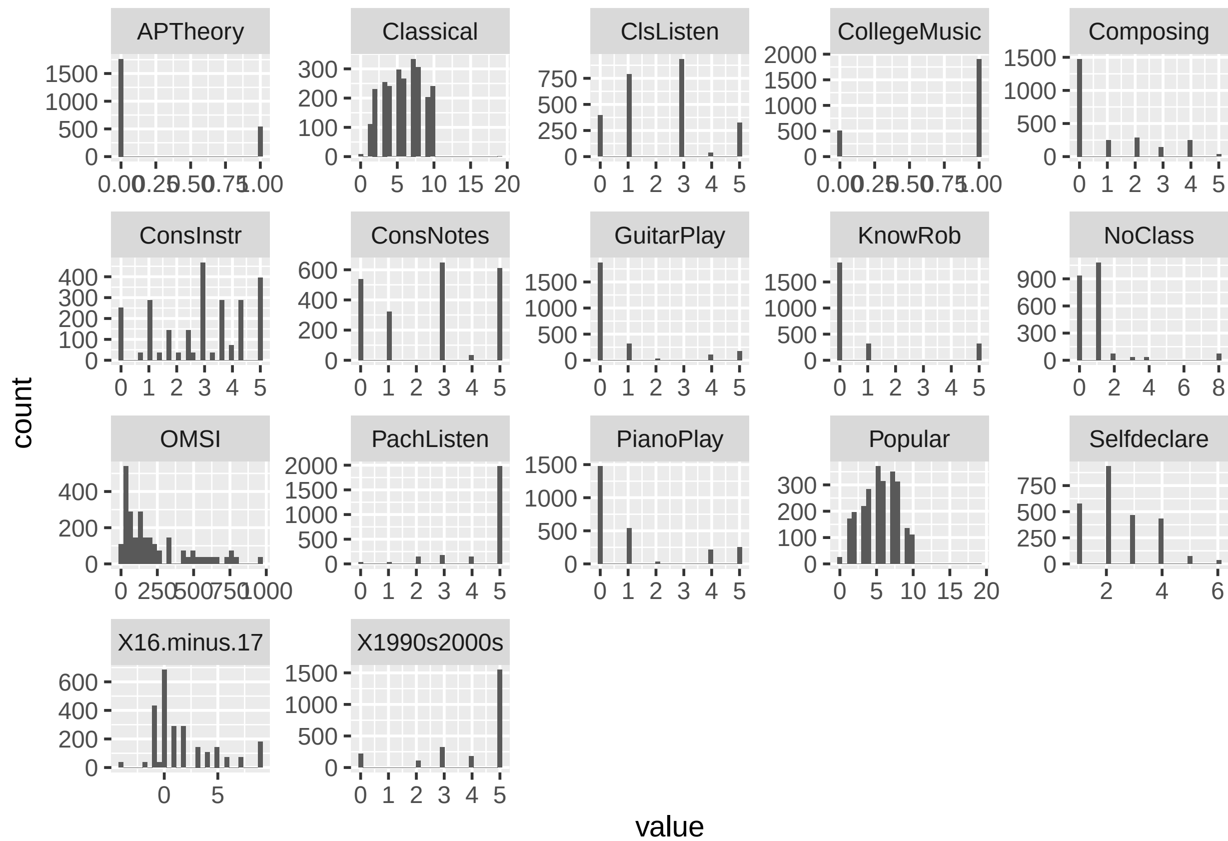
Then fill in missing data

```
for(i in 1:nrow(ratings)){
  if(is.na(ratings$PachListen[i])) {
    ratings$PachListen[i] <- round(4.43 + 0.114*ratings$KnowRob[i])
  }

  if(is.na(ratings$KnowRob[i])) {
    ratings$KnowRob[i] <- max(round(-0.5 + 0.28*ratings$PachListen[i]), 0)
  }
}
```

Plot Numeric Variables

```r
ratings[,-c(1:5, 11, 15, 17, 24, 25, 26)] %>%
  keep(is.numeric) %>%
  gather() %>%
  ggplot(aes(value)) +
    facet_wrap(~ key, scales = "free") +
    geom_histogram()
```

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

## Warning: Removed 1278 rows containing non-finite values (stat_bin).
```

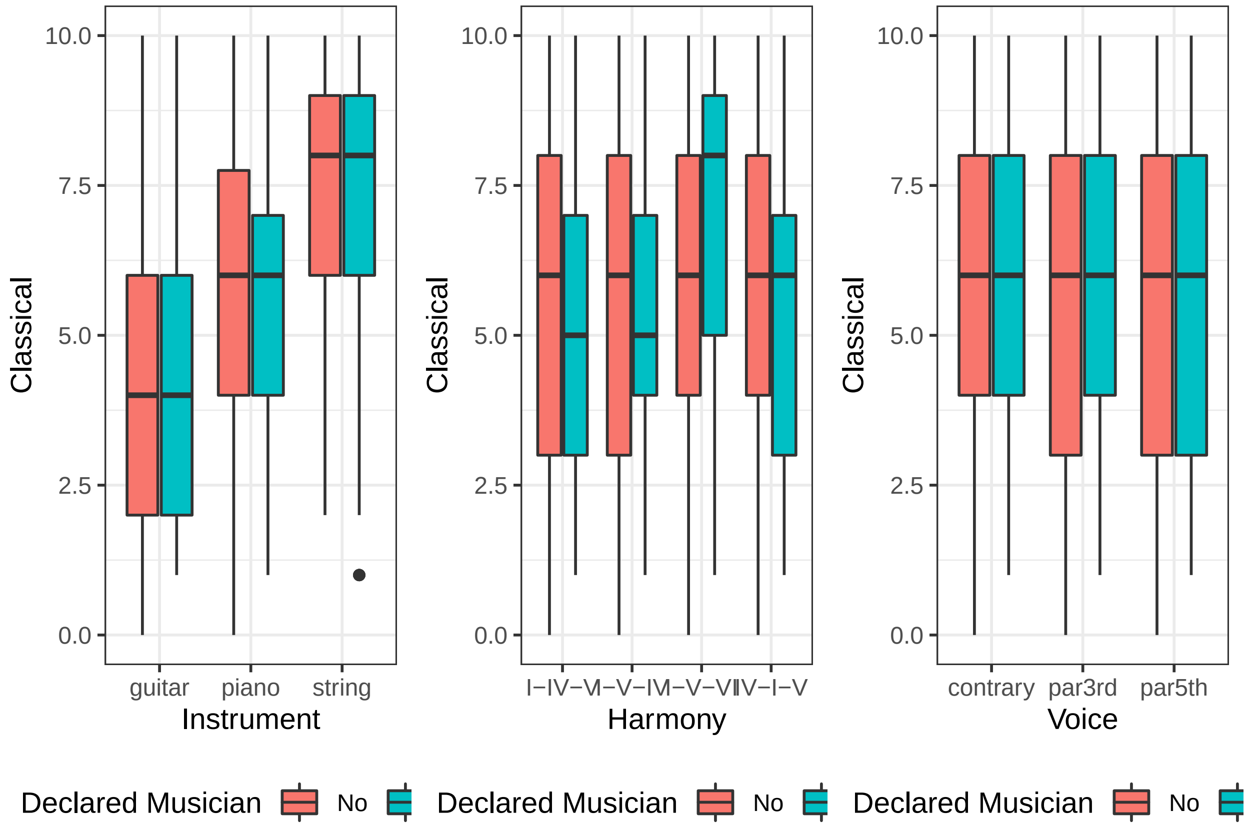


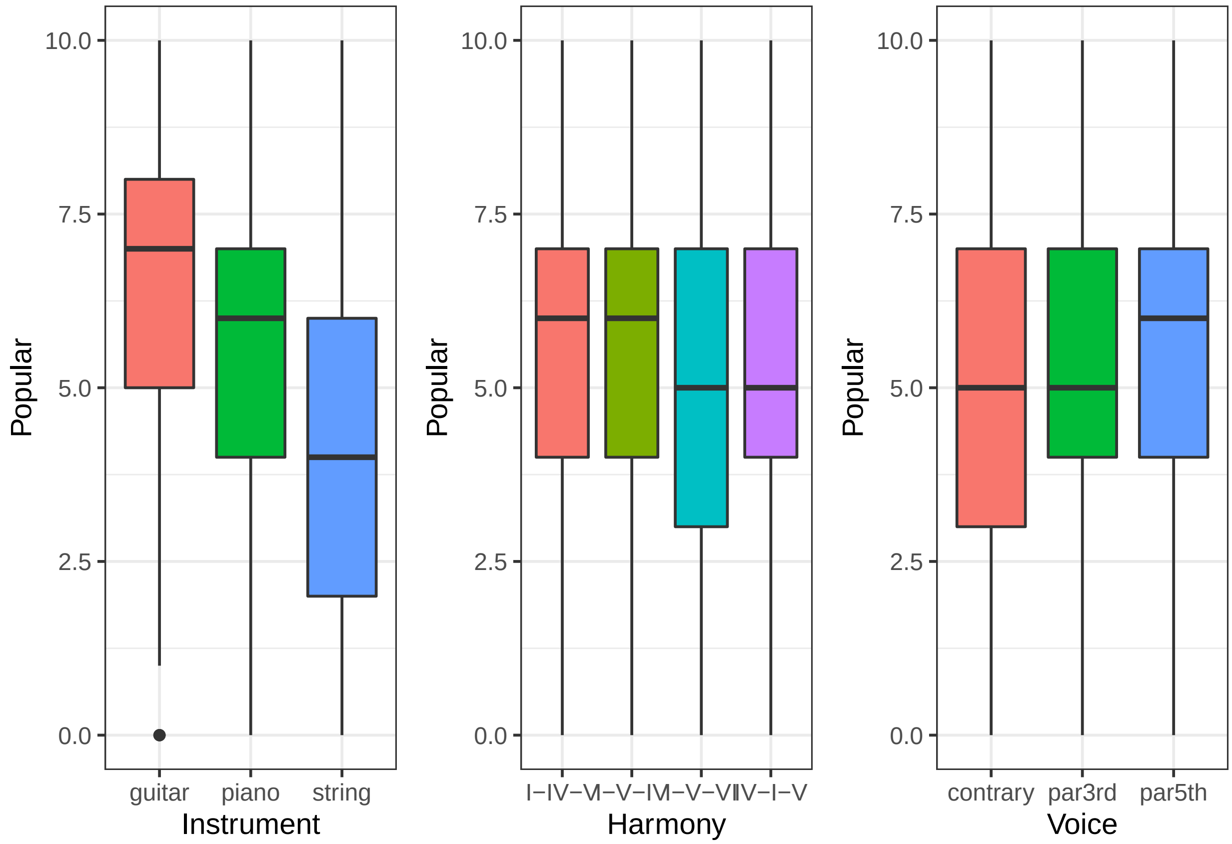
OMSI Score and classical/popular outliers

```r
powerTransform(OMSI ~ 1, data=ratings) # Should use log in model
```

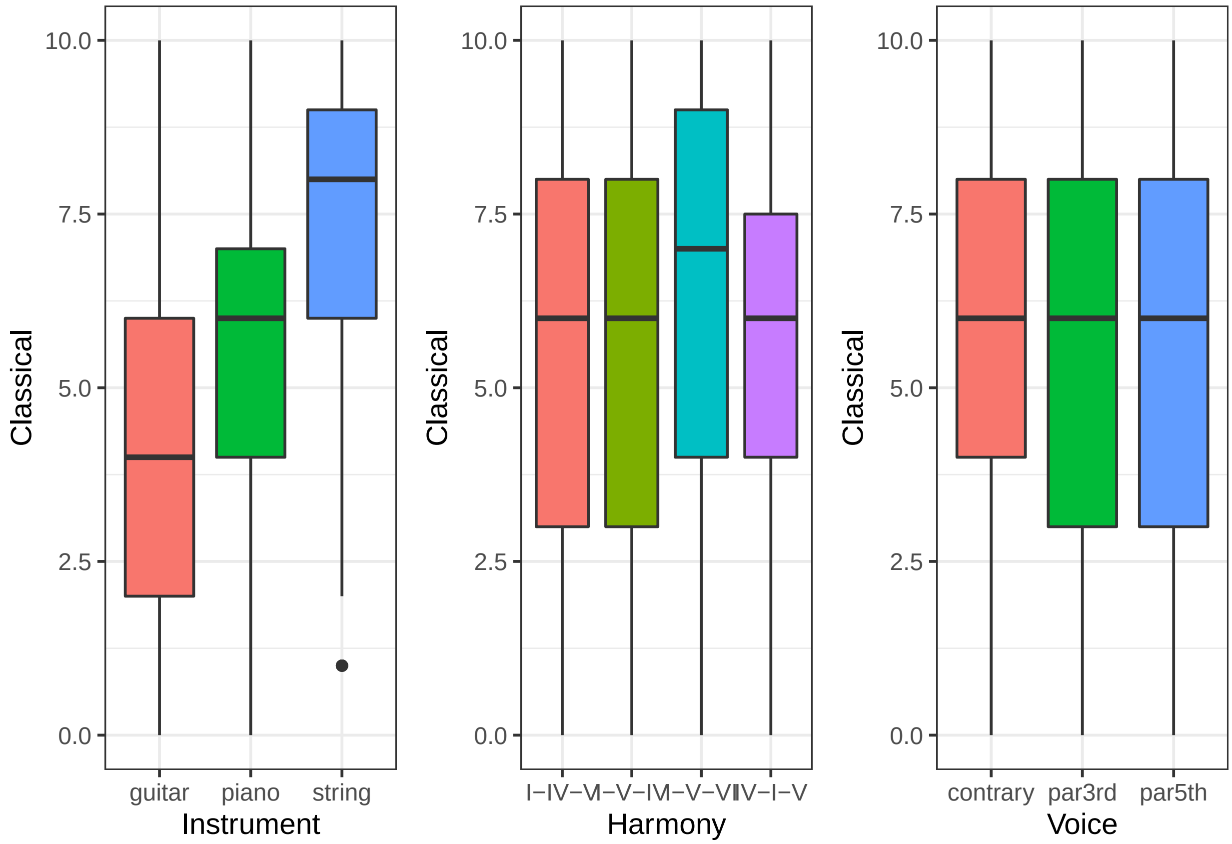
```
## Estimated transformation parameter
##         Y1
## 0.08565721
```

```r
ratings$binaryrob <- ifelse(ratings$KnowRob==5, 1, 0)
ratings <- subset(ratings, Popular <= 10 & Classical <= 10)
```

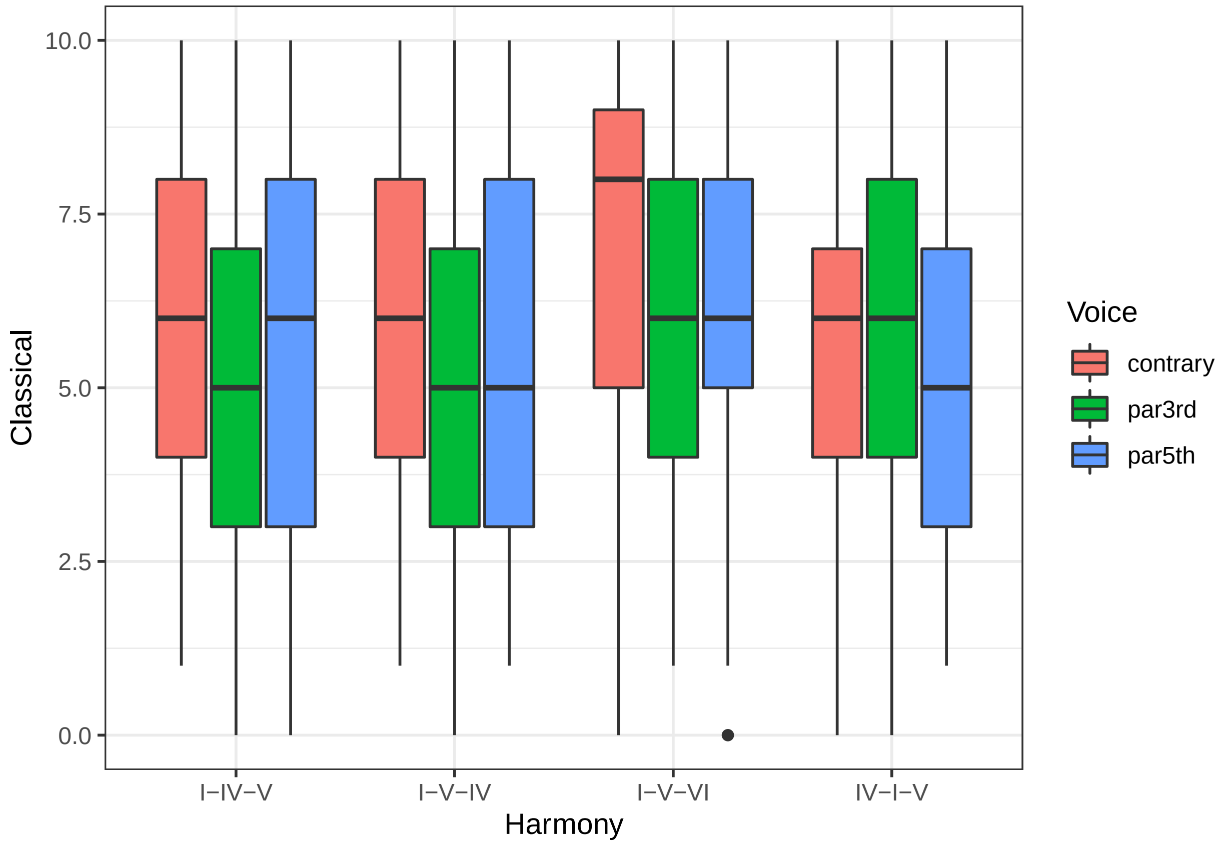

10.0
7.5
5.0
2.5
0.0
Classical
guitar
piano
string
Instrument
I−IV−V
I−V−IV
I−V−VI
IV−I−V
Harmony
contrary
par3rd
par5th
Voice
Declared Musician
No



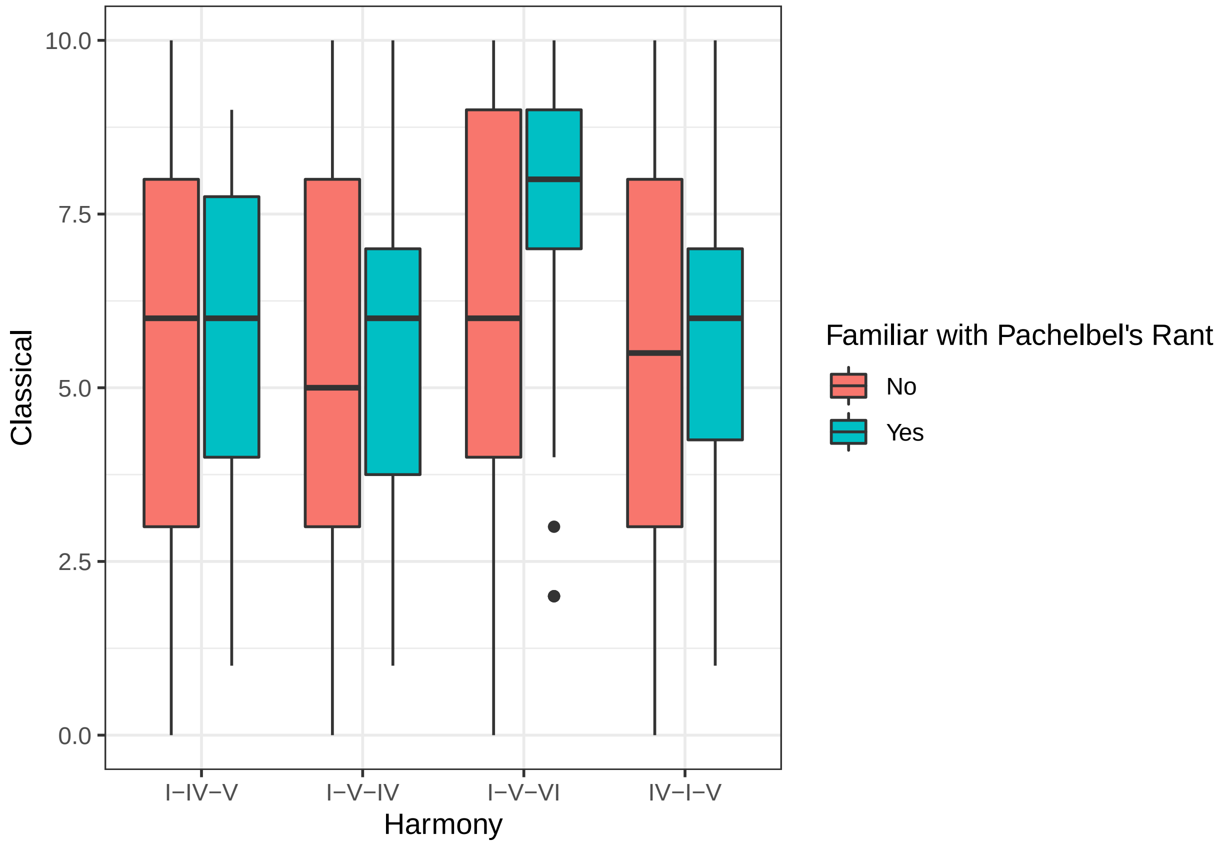
10.0
7.5
5.0
2.5
0.0
Popular
guitar
piano
string
Instrument
I−IV−V
I−V−IV
I−V−VI
IV−I−V
Harmony
contrary
par3rd
par5th
Voice



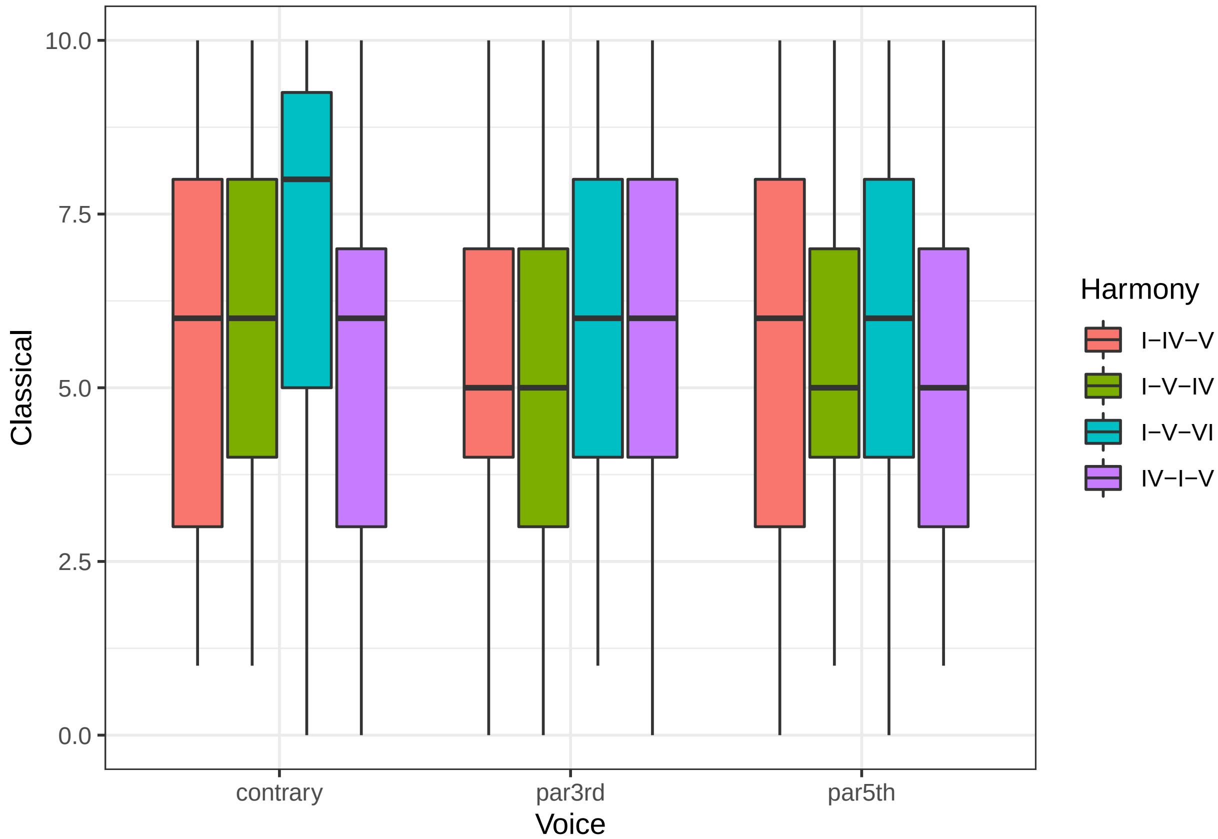
More boxplots to back up classical model



10.0
7.5
5.0
2.5
0.0
Classical
I−IV−V
I−V−IV
I−V−VI
IV−I−V
Harmony
Voice
contrary
par3rd
par5th




```
ggplot(ratings2, aes(x=Voice, y=Classical, fill=Harmony)) + geom_boxplot() +
  theme_bw() + labs(fill= "Harmony")
```



Classical Model

Note: The following is a simplified version of the steps that were taken to produce the final model.

First, determine best linear model.

```
lm1 <- lm(Classical ~ Instrument*Voice*Harmony, data=ratings2)

step.model <- stepAIC(lm1, direction = "both",
                      trace = FALSE)
step.model[["call"]][["formula"]]
```

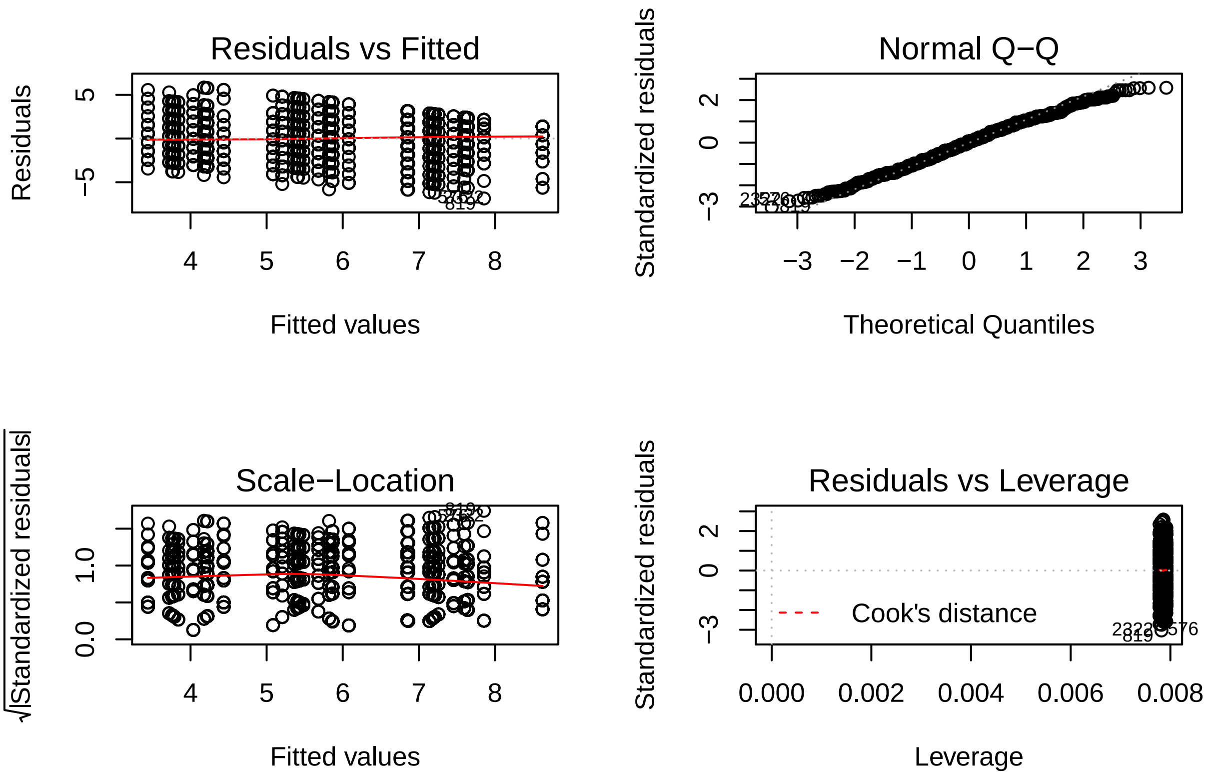
```
## Classical ~ Instrument + Voice + Harmony + Voice:Harmony
```

```
AIC(step.model)
```

```
## [1] 7996.031
```

Plot residuals

```
par(mfrow=c(2,2))
plot(step.model)
```



Determine if random intercept is important

```
lmer.intercept.only <- lmer(Classical ~ Harmony*Voice + Instrument +
                              (1|Subject), data=ratings2, REML=FALSE,
                            control = lmerControl(optimizer = "bobyqa"))

anova(lmer.intercept.only, step.model)
```

```
## Data: ratings2
## Models:
## step.model: Classical ~ Instrument + Voice + Harmony + Voice:Harmony
## lmer.intercept.only: Classical ~ Harmony * Voice + Instrument + (1 | Subject)
##                     Df    AIC    BIC  logLik deviance  Chisq Chi Df
## step.model          15 7996.0 8078.3 -3983.0   7966.0
## lmer.intercept.only 16 7535.4 7623.2 -3751.7   7503.4 462.64      1
##                     Pr(>Chisq)
## step.model
## lmer.intercept.only  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pr(>Chisq) << 0.05, and AIC much smaller with intercept model, so we will update model.

Now compare with more random effects (note: no other covariates yet).

```r
lmer.voice <- lmer(Classical ~Instrument + Harmony*Voice + (1 | Subject) +
                     (0 + Voice|Subject), data=ratings2, REML=FALSE,
                   control = lmerControl(optimizer= "bobyqa"))
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -1.2e+00
```

```r
lmer.instrument <- lmer(Classical ~Instrument + Harmony*Voice + (1 | Subject) +
                          (0 + Instrument|Subject),
                        data=ratings2, REML=FALSE, control = lmerControl(optimizer= "bobyqa"))
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues
```

```r
lmer.voice.instrument <- lmer(Classical ~ Instrument + Harmony*Voice + (1 | Subject) +
                                (0 + Voice | Subject) +
                                (0 + Instrument | Subject), data=ratings2, REML=FALSE,
                              control = lmerControl(optimizer = "bobyqa"))
```

```
## Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp), :
## convergence code 1 from bobyqa: bobyqa -- maximum number of function
## evaluations exceeded
```

```r
lmer.voice.harmony <- lmer(Classical ~ Instrument + Harmony*Voice + (1 | Subject) +
                             (0 + Harmony | Subject) +
                             (0 + Voice | Subject), data=ratings2, REML=FALSE,control = lmerControl(opt



lmer.voice.instrument.harmony <- lmer(Classical ~ Instrument + Harmony*Voice + (1 | Subject) +
                                        (0 + Voice | Subject) +
                                      (0 + Instrument | Subject) + (0 + Harmony | Subject), data=rating
                                      control = lmerControl(optimizer = "bobyqa"))
```

```
## Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp), :
## convergence code 1 from bobyqa: bobyqa -- maximum number of function
## evaluations exceeded

## Warning: Model failed to converge with 4 negative eigenvalues: -3.3e-02
## -1.6e-01 -7.7e+00 -6.6e+01
```

```r
anova(lmer.voice, lmer.instrument, lmer.voice.instrument, lmer.voice.harmony, lmer.voice.instrument.har
```

```
## Data: ratings2
## Models:
## lmer.voice: Classical ~ Instrument + Harmony * Voice + (1 | Subject) + (0 +
```

```
## lmer.voice:     Voice | Subject)
## lmer.instrument: Classical ~ Instrument + Harmony * Voice + (1 | Subject) + (0 +
## lmer.instrument:     Instrument | Subject)
## lmer.voice.instrument: Classical ~ Instrument + Harmony * Voice + (1 | Subject) + (0 +
## lmer.voice.instrument:     Voice | Subject) + (0 + Instrument | Subject)
## lmer.voice.harmony: Classical ~ Instrument + Harmony * Voice + (1 | Subject) + (0 +
## lmer.voice.harmony:     Harmony | Subject) + (0 + Voice | Subject)
## lmer.voice.instrument.harmony: Classical ~ Instrument + Harmony * Voice + (1 | Subject) + (0 +
## lmer.voice.instrument.harmony:     Voice | Subject) + (0 + Instrument | Subject) + (0 + Harmony |
## lmer.voice.instrument.harmony:     Subject)
##                               Df    AIC    BIC  logLik deviance    Chisq
## lmer.voice                    22 7547.2 7667.9 -3751.6   7503.2
## lmer.instrument               22 7239.2 7359.9 -3597.6   7195.2 308.0413
## lmer.voice.instrument         28 7250.1 7403.7 -3597.1   7194.1   1.0547
## lmer.voice.harmony            32 7512.9 7688.5 -3724.5   7448.9   0.0000
## lmer.voice.instrument.harmony 38 7168.7 7377.1 -3546.3   7092.7 356.2545
##                               Chi Df Pr(>Chisq)
## lmer.voice
## lmer.instrument                    0     <2e-16 ***
## lmer.voice.instrument              6     0.9835
## lmer.voice.harmony                 4     1.0000
## lmer.voice.instrument.harmony      6     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The lmer with the instrument random effect works best (note that some models were not shown because of non-convergence).

Add covariates

```
class(ratings2$Instrument) <- class(ratings2$Harmony) <- class(ratings2$Voice) <- "factor"
ratings2$musician <- ifelse(ratings2$Selfdeclare <= 2, 0, 1)
#ratings2$binaryrob <- ifelse(ratings2$KnowRob == 5, 1, 0)

lmer.harmony.full <- lmer(Classical ~ Harmony*Voice + Instrument + musician*Harmony +
                            musician*Voice + musician*Instrument + Harmony*KnowRob*PachListen + log(OMSI
                       PianoPlay + GuitarPlay + X16.minus.17 + ConsInstr + ConsNotes + ClsListen + X1990
                         CollegeMusic + NoClass + APTheory + Composing
                         + (1 | Subject) + (0 + Harmony | Subject),
                     data=ratings2, REML=FALSE, control = lmerControl(optimizer = "bobyqa"))
```

Stepwise regression to determine best model with covariates

```
# summary(final_fm)
```

Determine if we should change random effects

```
lmer.fe.int <- lmer(Classical ~ Harmony*KnowRob + Voice + Instrument + musician +
                            PianoPlay + X16.minus.17 + ConsNotes + (0 + Harmony | Subject) +
                            Harmony:Voice + Harmony:musician + Instrument:musician +
                         (1 | Subject), data=ratings2, REML=FALSE, control = lmerControl(optimizer = "bo
```

```r
lmer.fe.voice <- lmer(Classical ~ Harmony*KnowRob + Voice + Instrument + musician +
                           PianoPlay + X16.minus.17 + ConsNotes +
                           Harmony:Voice + Harmony:musician + Instrument:musician +
                         (1 | Subject) + (0 + Voice | Subject), data=ratings2, REML=FALSE,
                         control = lmerControl(optimizer = "bobyqa"))


lmer.fe.harm <- lmer(Classical ~ Harmony*KnowRob + Voice + Instrument + musician +
                           PianoPlay + X16.minus.17 + ConsNotes +
                           Harmony:Voice + Harmony:musician + Instrument:musician +
                         (1 | Subject) + (0 + Harmony | Subject), data=ratings2, REML=FALSE,
                         control = lmerControl(optimizer = "bobyqa"))
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -1.6e+02
```

```r
lmer.fe.voice.harm.inst <- lmer(Classical ~ Harmony*KnowRob + Voice + Instrument + musician +
                           PianoPlay + X16.minus.17 + ConsNotes +
                           Harmony:Voice + Harmony:musician + Instrument:musician +
                         (1 | Subject) + (0 + Harmony | Subject) + (0 + Voice | Subject) +
                           (0 + Instrument | Subject), data=ratings2, REML=FALSE,
                         control = lmerControl(optimizer = "bobyqa"))
```

```r
anova(lmer.fe.int, lmer.fe.voice, lmer.fe.harm, lmer.fe.voice.harm.inst)
```

```
## Data: ratings2
## Models:
## lmer.fe.voice: Classical ~ Harmony * KnowRob + Voice + Instrument + musician +
## lmer.fe.voice:     PianoPlay + X16.minus.17 + ConsNotes + Harmony:Voice + Harmony:musician +
## lmer.fe.voice:     Instrument:musician + (1 | Subject) + (0 + Voice | Subject)
## lmer.fe.int: Classical ~ Harmony * KnowRob + Voice + Instrument + musician +
## lmer.fe.int:     PianoPlay + X16.minus.17 + ConsNotes + (0 + Harmony | Subject) +
## lmer.fe.int:     Harmony:Voice + Harmony:musician + Instrument:musician +
## lmer.fe.int:     (1 | Subject)
## lmer.fe.harm: Classical ~ Harmony * KnowRob + Voice + Instrument + musician +
## lmer.fe.harm:     PianoPlay + X16.minus.17 + ConsNotes + Harmony:Voice + Harmony:musician +
## lmer.fe.harm:     Instrument:musician + (1 | Subject) + (0 + Harmony | Subject)
## lmer.fe.voice.harm.inst: Classical ~ Harmony * KnowRob + Voice + Instrument + musician +
## lmer.fe.voice.harm.inst:     PianoPlay + X16.minus.17 + ConsNotes + Harmony:Voice + Harmony:musician
## lmer.fe.voice.harm.inst:     Instrument:musician + (1 | Subject) + (0 + Harmony | Subject) +
## lmer.fe.voice.harm.inst:     (0 + Voice | Subject) + (0 + Instrument | Subject)
##                         Df    AIC    BIC  logLik deviance   Chisq Chi Df
## lmer.fe.voice           35 7494.1 7686.1 -3712.1   7424.1
## lmer.fe.int             39 7480.5 7694.5 -3701.3   7402.5  21.594      4
## lmer.fe.harm            39 7480.5 7694.5 -3701.3   7402.5   0.000      0
## lmer.fe.voice.harm.inst 51 7158.5 7438.3 -3528.3   7056.5 345.985     12
##                         Pr(>Chisq)
## lmer.fe.voice
## lmer.fe.int              0.0002414 ***
## lmer.fe.harm             1.0000000
## lmer.fe.voice.harm.inst  < 2.2e-16 ***
## ---
```

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

It now appears that we should use voice, harmony, and instrument random effects.

```
lmer.fe.voice.harm.inst
```

```
## Linear mixed model fit by maximum likelihood  ['lmerModLmerTest']
## Formula: Classical ~ Harmony * KnowRob + Voice + Instrument + musician +
##     PianoPlay + X16.minus.17 + ConsNotes + Harmony:Voice + Harmony:musician +
##     Instrument:musician + (1 | Subject) + (0 + Harmony | Subject) +
##     (0 + Voice | Subject) + (0 + Instrument | Subject)
##    Data: ratings2
##       AIC       BIC    logLik  deviance  df.resid
##  7158.531  7438.320 -3528.265  7056.531      1732
## Random effects:
##  Groups    Name             Std.Dev.  Corr
##  Subject   (Intercept)      1.991e-05
##  Subject.1 HarmonyI-IV-V    8.449e-01
##            HarmonyI-V-IV    1.055e+00 0.95
##            HarmonyI-V-VI    1.109e+00 0.54  0.72
##            HarmonyIV-I-V    9.126e-01 0.96  0.89  0.63
##  Subject.2 Voicecontrary    5.597e-01
##            Voicepar3rd      5.946e-01 0.78
##            Voicepar5th      4.535e-01 0.87  0.99
##  Subject.3 Instrumentguitar 9.266e-01
##            Instrumentpiano  1.118e+00  0.17
##            Instrumentstring 9.902e-01 -0.99 -0.02
##  Residual                   1.555e+00
## Number of obs: 1783, groups:  Subject, 50
## Fixed Effects:
##              (Intercept)              HarmonyI-V-IV
##                 4.251337                   0.163046
##            HarmonyI-V-VI              HarmonyIV-I-V
##                 0.456219                  -0.304244
##                  KnowRob                Voicepar3rd
##                 0.062385                  -0.276201
##              Voicepar5th            Instrumentpiano
##                -0.208396                   1.886213
##         Instrumentstring                   musician
##                 3.765677                  -0.281390
##                PianoPlay               X16.minus.17
##                 0.281173                  -0.067127
##                ConsNotes      HarmonyI-V-IV:KnowRob
##                -0.125301                   0.002246
##    HarmonyI-V-VI:KnowRob      HarmonyIV-I-V:KnowRob
##                 0.296622                   0.012698
## HarmonyI-V-IV:Voicepar3rd  HarmonyI-V-VI:Voicepar3rd
##                -0.489231                  -0.702249
## HarmonyIV-I-V:Voicepar3rd  HarmonyI-V-IV:Voicepar5th
##                 0.670296                  -0.229224
## HarmonyI-V-VI:Voicepar5th  HarmonyIV-I-V:Voicepar5th
##                -0.558935                   0.153321
##   HarmonyI-V-IV:musician     HarmonyI-V-VI:musician
##                 0.032438                   1.048662
```

```
##    HarmonyIV-I-V:musician   Instrumentpiano:musician
##                  0.073178                  -0.552774
## Instrumentstring:musician
##                 -0.776579
## convergence code 0; 1 optimizer warnings; 0 lme4 warnings
```

Relevel Voice so we can compare contrary

```
#ratings3 <- within(ratings2, Voice <- relevel(Voice, ref = 2))

final.classical.releveled <- lmer(Classical ~ Voice + Harmony*binaryrob+ Instrument + musician +
                            PianoPlay + X16.minus.17 + ConsNotes +
                            Harmony:Voice + Harmony:musician + Instrument:musician -1 +
                         (1 | Subject) + (0 + Harmony | Subject) + (0 + Voice | Subject) +
                           (0 + Instrument | Subject), data=ratings2, REML=FALSE,
                         control = lmerControl(optimizer = "bobyqa"))

summary(final.classical.releveled)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use
##   Satterthwaite's method [lmerModLmerTest]
## Formula:
## Classical ~ Voice + Harmony * binaryrob + Instrument + musician +
##     PianoPlay + X16.minus.17 + ConsNotes + Harmony:Voice + Harmony:musician +
##     Instrument:musician - 1 + (1 | Subject) + (0 + Harmony |
##     Subject) + (0 + Voice | Subject) + (0 + Instrument | Subject)
##    Data: ratings2
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   7158.9   7438.7  -3528.5   7056.9     1732
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6199 -0.5791  0.0274  0.5502  3.4722
##
## Random effects:
##  Groups    Name             Variance Std.Dev. Corr
##  Subject   (Intercept)      0.0000   0.0000
##  Subject.1 HarmonyI-IV-V    0.7012   0.8374
##            HarmonyI-V-IV    1.0994   1.0485   0.95
##            HarmonyI-V-VI    1.2252   1.1069   0.55  0.72
##            HarmonyIV-I-V    0.8187   0.9048   0.96  0.89  0.63
##  Subject.2 Voicecontrary    0.3270   0.5718
##            Voicepar3rd      0.3710   0.6091   0.79
##            Voicepar5th      0.2198   0.4688   0.88  0.99
##  Subject.3 Instrumentguitar 0.8665   0.9309
##            Instrumentpiano  1.2835   1.1329    0.19
##            Instrumentstring 0.9769   0.9884   -0.98 -0.01
##  Residual                   2.4174   1.5548
## Number of obs: 1783, groups:  Subject, 50
##
## Fixed effects:
```

```
##                            Estimate Std. Error         df t value
## Voicecontrary               4.25630    0.37298   75.78880  11.412
## Voicepar3rd                 3.97996    0.37411   73.28222  10.638
## Voicepar5th                 4.04774    0.37009   73.74761  10.937
## HarmonyI-V-IV               0.15959    0.21461  274.60966   0.744
## HarmonyI-V-VI               0.48746    0.27394   99.71232   1.779
## HarmonyIV-I-V              -0.30602    0.20957  328.02620  -1.460
## binaryrob                   0.21105    0.48389   50.95506   0.436
## Instrumentpiano             1.88616    0.27702   49.57512   6.809
## Instrumentstring            3.76562    0.38061   49.99122   9.894
## musician                   -0.26233    0.46140   54.25792  -0.569
## PianoPlay                   0.28063    0.09539   50.39704   2.942
## X16.minus.17               -0.06652    0.05152   50.37173  -1.291
## ConsNotes                  -0.12151    0.08407   50.36919  -1.445
## HarmonyI-V-IV:binaryrob     0.06262    0.34466   88.01617   0.182
## HarmonyI-V-VI:binaryrob     1.49544    0.50601   51.24660   2.955
## HarmonyIV-I-V:binaryrob     0.10677    0.32757   97.13036   0.326
## Voicepar3rd:HarmonyI-V-IV  -0.48903    0.25507 1474.27460  -1.917
## Voicepar5th:HarmonyI-V-IV  -0.22936    0.25526 1473.61199  -0.899
## Voicepar3rd:HarmonyI-V-VI  -0.70261    0.25516 1475.16277  -2.754
## Voicepar5th:HarmonyI-V-VI  -0.55938    0.25539 1474.78989  -2.190
## Voicepar3rd:HarmonyIV-I-V   0.67045    0.25507 1474.64461   2.628
## Voicepar5th:HarmonyIV-I-V   0.15331    0.25508 1474.55014   0.601
## HarmonyI-V-IV:musician      0.02522    0.23910   85.63812   0.105
## HarmonyI-V-VI:musician      1.08171    0.35287   50.79003   3.065
## HarmonyIV-I-V:musician      0.06815    0.22853   96.26374   0.298
## Instrumentpiano:musician   -0.55238    0.41888   50.10386  -1.319
## Instrumentstring:musician  -0.77650    0.57379   49.99350  -1.353
##                           Pr(>|t|)
## Voicecontrary              < 2e-16 ***
## Voicepar3rd                < 2e-16 ***
## Voicepar5th                < 2e-16 ***
## HarmonyI-V-IV              0.45773
## HarmonyI-V-VI              0.07821 .
## HarmonyIV-I-V              0.14518
## binaryrob                  0.66456
## Instrumentpiano           1.24e-08 ***
## Instrumentstring          2.30e-13 ***
## musician                   0.57201
## PianoPlay                  0.00492 **
## X16.minus.17               0.20254
## ConsNotes                  0.15455
## HarmonyI-V-IV:binaryrob    0.85625
## HarmonyI-V-VI:binaryrob    0.00471 **
## HarmonyIV-I-V:binaryrob    0.74517
## Voicepar3rd:HarmonyI-V-IV  0.05540 .
## Voicepar5th:HarmonyI-V-IV  0.36904
## Voicepar3rd:HarmonyI-V-VI  0.00597 **
## Voicepar5th:HarmonyI-V-VI  0.02866 *
## Voicepar3rd:HarmonyIV-I-V  0.00867 **
## Voicepar5th:HarmonyIV-I-V  0.54793
## HarmonyI-V-IV:musician     0.91626
## HarmonyI-V-VI:musician     0.00348 **
## HarmonyIV-I-V:musician     0.76619
```

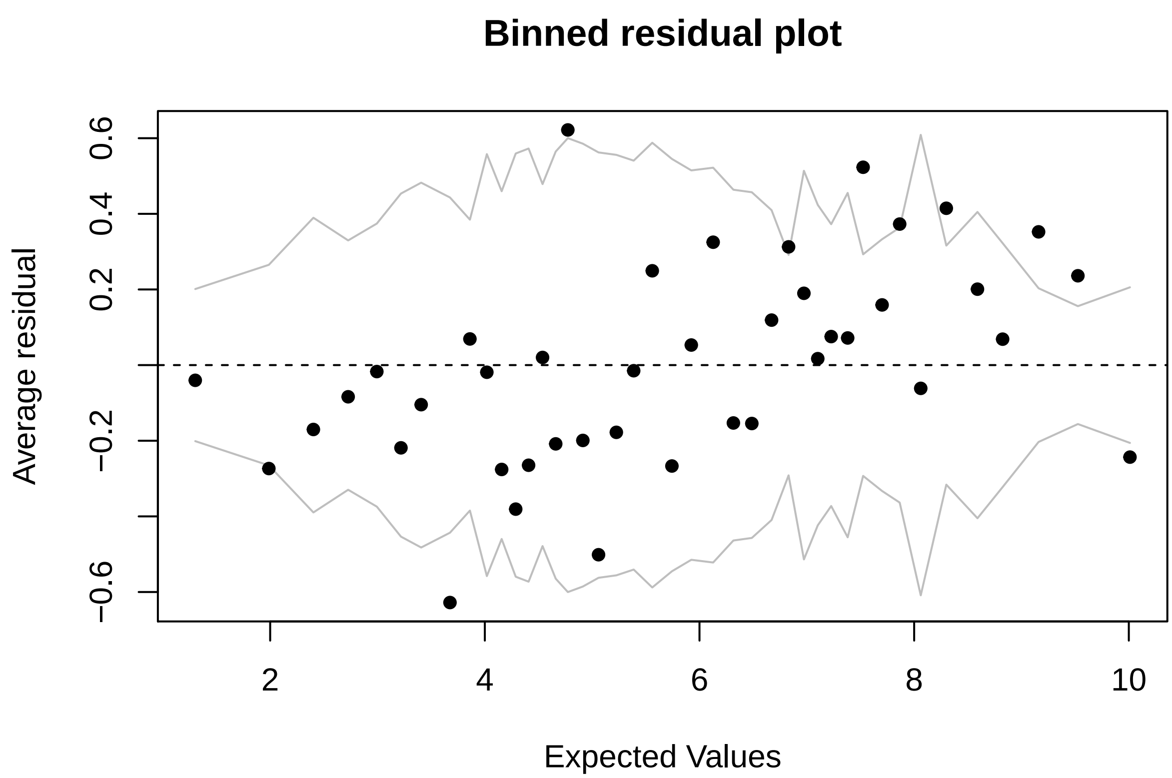
```
## Instrumentpiano:musician   0.19326
## Instrumentstring:musician  0.18205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

See paper for regression output.

Check errors for Classical Model

We'll start by looking at the binned residuals

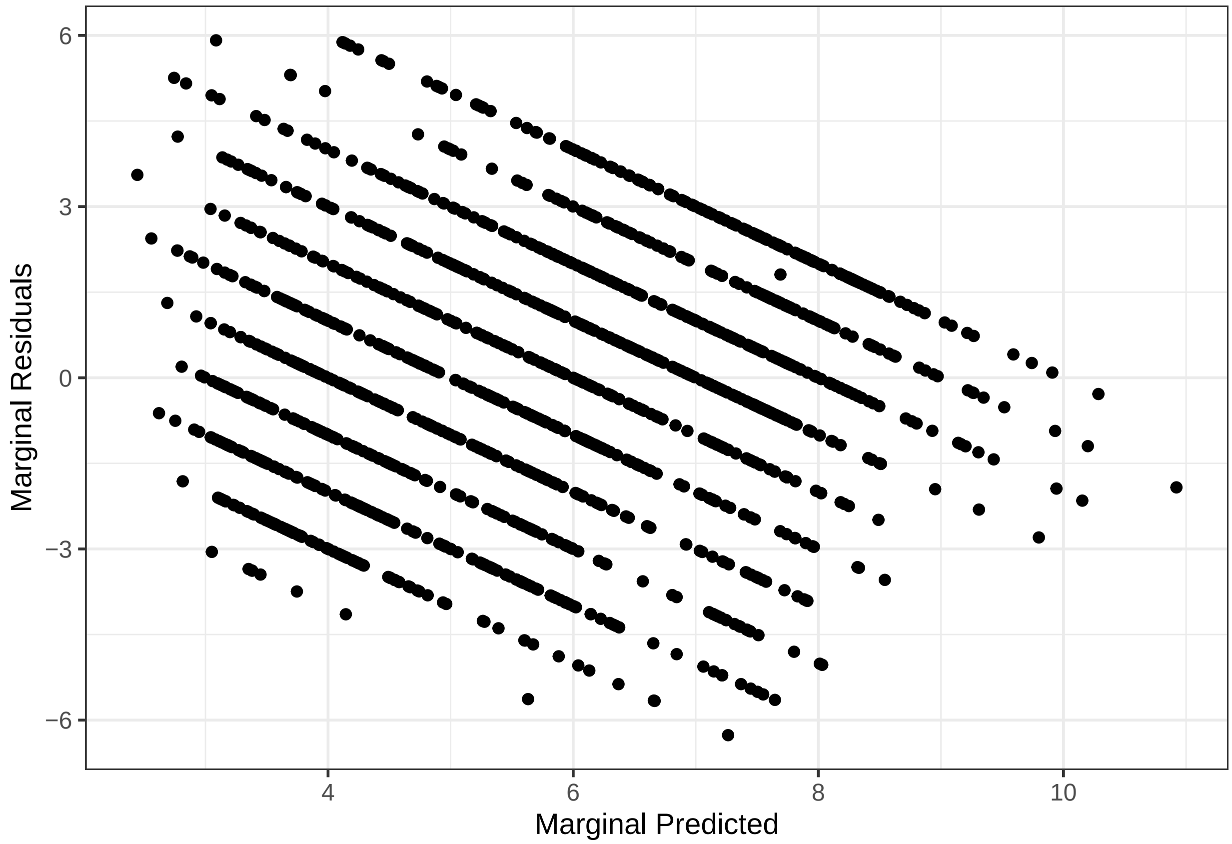
```r
modelc <- final.classical.releveled
binnedplot(fitted(modelc),resid(modelc))
```



It appears that the majority of the residuals are within the bin.

Next, we look at marginal fitted values vs. residuals

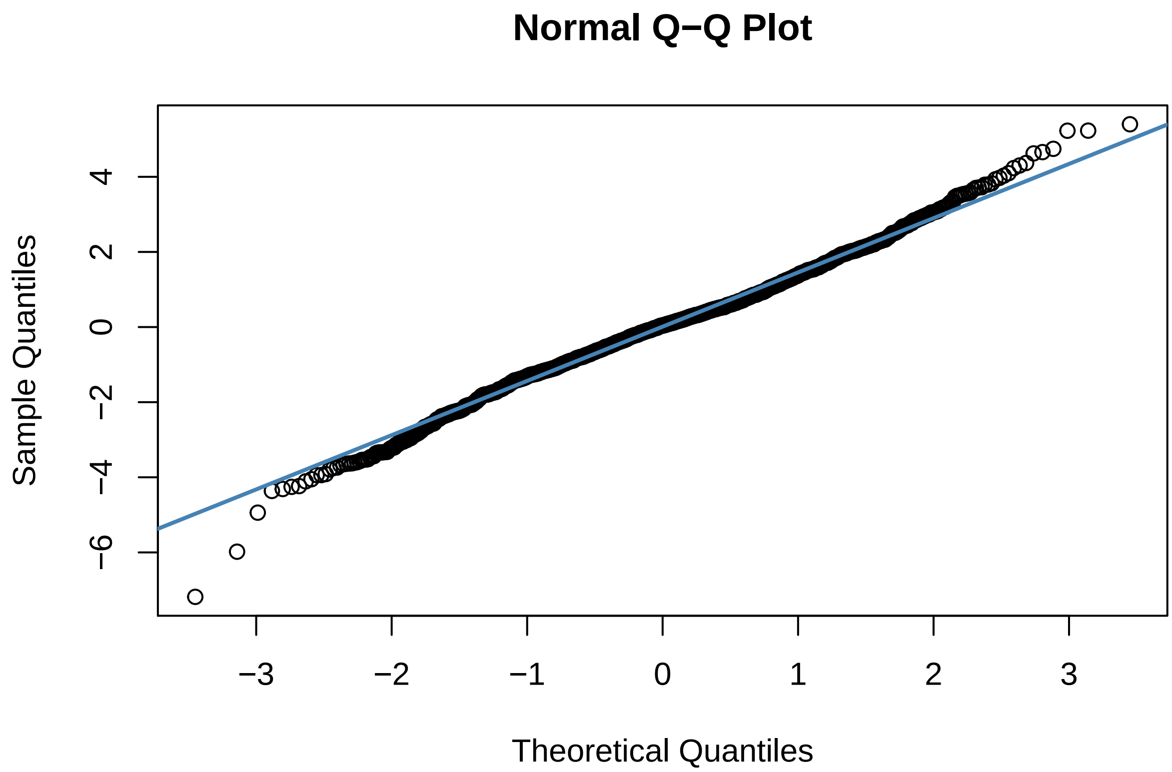
```r
ggplot(mapping=aes(yhat.marg(modelc), r.marg(modelc))) + geom_point() +
  labs(x="Marginal Predicted", y="Marginal Residuals") + theme_bw()
```



In the marginal models residuals plot above, we don't care about trends, but are more focused on the spread of the points. Therefore, the marginal residuals plot above looks good.

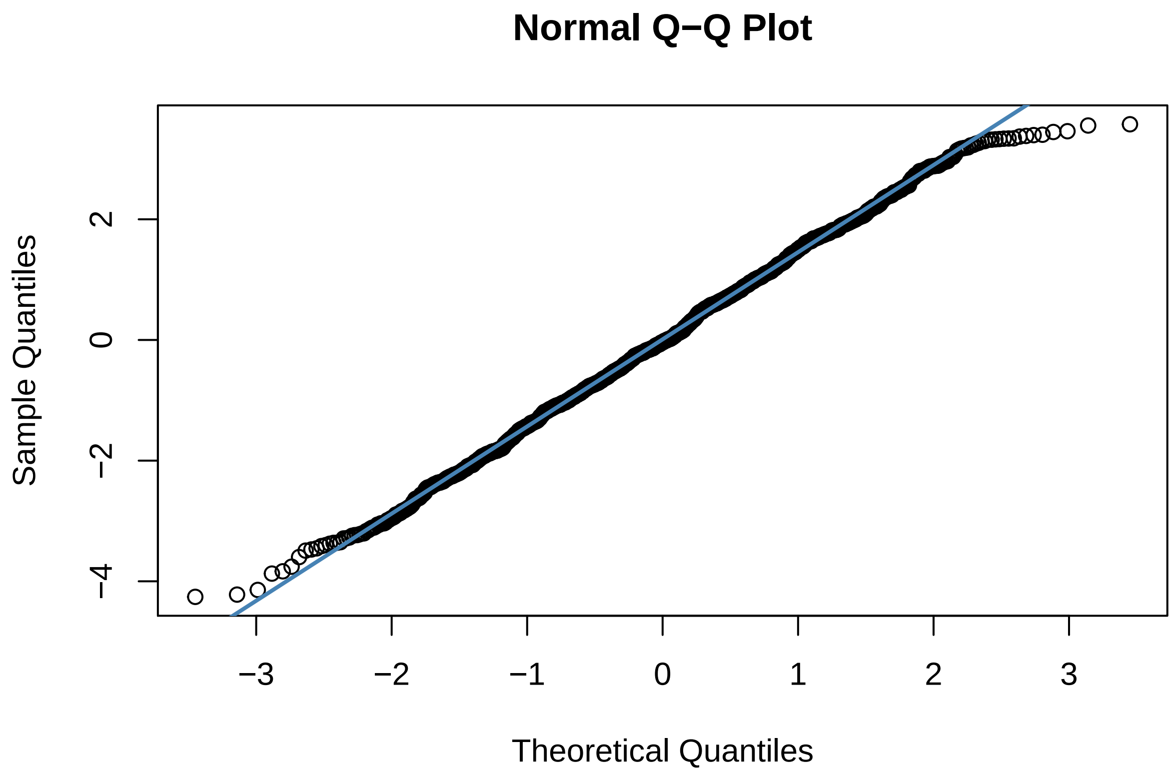
Now we will look at the QQ plot for conditional residuals.

```
qqnorm(r.cond(modelc))
qqline(r.reff(modelc), col = "steelblue", lwd = 2)
```



And the QQ plot for the random effects

```
qqnorm(r.reff(modelc))
qqline(r.reff(modelc), col = "steelblue", lwd = 2)
```



Both QQ plots look linear, suggesting normality of residuals.

We now move to determining the popular ratings model.

Popular Ratings Model

We'll start by running a stepwise regression to determine the optimal fixed effects.

```
lm.pop <- lm(Popular ~ Harmony*Voice + Instrument + musician*Harmony + musician*Voice + musician*Instru
               Harmony*KnowRob*PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 + ConsIns
               ConsNotes + ClsListen + X1990s2000s + CollegeMusic + NoClass + APTheory + Composing, data

step.model <- stepAIC(lm.pop, direction = "both",
                      trace = FALSE)
# summary(step.model)
```

```
lmer.pop.int <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen + (1 | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)

lmer.pop.harm <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
```

```r
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen + (1 | Subject) + (0 + Harmony | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)
```

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

## Warning: Model failed to converge with 1 negative eigenvalue: -2.9e+00
```

```r
lmer.pop.voice <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen +  (1 | Subject) + (0 + Voice | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)
```

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

```r
lmer.pop.harm.inst <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen + (1 | Subject) + (0 + Harmony | Subject) +
                       (0 + Instrument | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)
```

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

## Warning: Model failed to converge with 2 negative eigenvalues: -1.8e-05
## -8.4e+00
```

```r
lmer.pop.harm.voice <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen +
                       (0 + Voice | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)
```

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

```r
lmer.pop.voice.inst <- lmer(Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
    PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
    ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
    APTheory + Composing + Harmony:musician + Instrument:musician +
    KnowRob:PachListen +
                       (0 + Instrument | Subject), data=ratings2,
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)
```

There were a few models that failed to converge, but we can still compare the models that did converge.

```r
anova(lmer.pop.int, lmer.pop.harm, lmer.pop.voice, lmer.pop.harm.inst, lmer.pop.harm.voice, lmer.pop.vo
```

```
## Data: ratings2
## Models:
## lmer.pop.int: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.int:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.int:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.int:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.int:     KnowRob:PachListen + (1 | Subject)
## lmer.pop.harm.voice: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.harm.voice:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.harm.voice:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.harm.voice:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.harm.voice:     KnowRob:PachListen + (0 + Voice | Subject)
## lmer.pop.voice.inst: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.voice.inst:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.voice.inst:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.voice.inst:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.voice.inst:     KnowRob:PachListen + (0 + Instrument | Subject)
## lmer.pop.voice: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.voice:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.voice:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.voice:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.voice:     KnowRob:PachListen + (1 | Subject) + (0 + Voice | Subject)
## lmer.pop.harm: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.harm:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.harm:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.harm:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.harm:     KnowRob:PachListen + (1 | Subject) + (0 + Harmony | Subject)
## lmer.pop.harm.inst: Popular ~ Harmony + Voice + Instrument + musician + KnowRob +
## lmer.pop.harm.inst:     PachListen + log(OMSI) + PianoPlay + GuitarPlay + X16.minus.17 +
## lmer.pop.harm.inst:     ConsInstr + ConsNotes + X1990s2000s + CollegeMusic + NoClass +
## lmer.pop.harm.inst:     APTheory + Composing + Harmony:musician + Instrument:musician +
## lmer.pop.harm.inst:     KnowRob:PachListen + (1 | Subject) + (0 + Harmony | Subject) +
## lmer.pop.harm.inst:     (0 + Instrument | Subject)
##                     Df    AIC    BIC  logLik deviance   Chisq Chi Df
## lmer.pop.int        30 7525.8 7690.3 -3732.9   7465.8
## lmer.pop.harm.voice 35 7532.6 7724.6 -3731.3   7462.6   3.166      5
## lmer.pop.voice.inst 35 7306.7 7498.7 -3618.4   7236.7 225.883      0
## lmer.pop.voice      36 7534.6 7732.1 -3731.3   7462.6   0.000      1
## lmer.pop.harm       40 7510.9 7730.3 -3715.4   7430.9  31.736      4
## lmer.pop.harm.inst  46 7262.3 7514.6 -3585.1   7170.3 260.592      6
##                     Pr(>Chisq)
## lmer.pop.int
## lmer.pop.harm.voice     0.6744
## lmer.pop.voice.inst  < 2.2e-16 ***
## lmer.pop.voice          1.0000
## lmer.pop.harm        2.166e-06 ***
## lmer.pop.harm.inst   < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The smallest BIC overall appears to be the one where we used voice and instrument random effects. Let's do one more check with fixed effects to see if there are any more variables we should drop.

```
step_fm <- step(lmer.pop.voice.inst)
final_fm <- get_model(step_fm)
# summary(final_fm)
final_fm
```

```
## Linear mixed model fit by maximum likelihood  ['lmerModLmerTest']
## Formula: Popular ~ Harmony + Instrument + musician + (0 + Instrument |
##     Subject) + Harmony:musician
##    Data: ratings2
##       AIC       BIC    logLik  deviance  df.resid
##  7285.833  7379.096 -3625.917  7251.833      1766
## Random effects:
##  Groups   Name             Std.Dev. Corr
##  Subject  Instrumentguitar 1.102
##           Instrumentpiano  1.506    0.54
##           Instrumentstring 1.710    0.22 0.73
##  Residual                  1.705
## Number of obs: 1783, groups:  Subject, 50
## Fixed Effects:
##           (Intercept)           HarmonyI-V-IV           HarmonyI-V-VI
##               6.84581                -0.13095                 0.08099
##          HarmonyIV-I-V         Instrumentpiano        Instrumentstring
##              -0.25397                -1.09293                -2.76959
##              musician  HarmonyI-V-IV:musician  HarmonyI-V-VI:musician
##               0.13023                 0.18816                -0.75599
## HarmonyIV-I-V:musician
##               0.04709
```

The stepwise regression model suggests to only use the variables Harmony, Voice, musician, Instrument, and the interaction Harmony:musician.

Since adding back in Voice and Harmony random effects would cause negative eigenvalues, the final model then is:

```
lmer.pop.final <- lmer(Popular ~  Voice + Harmony*musician + Instrument -1 +
                       (1 | Subject) + (0 + Voice | Subject) + (0 + Instrument | Subject), data=ratings2
                     control= lmerControl(optimizer = "bobyqa"), REML = FALSE)

summary(lmer.pop.final)
```

```
## Linear mixed model fit by maximum likelihood . t-tests use
##   Satterthwaite's method [lmerModLmerTest]
## Formula: Popular ~ Voice + Harmony * musician + Instrument - 1 + (1 |
##     Subject) + (0 + Voice | Subject) + (0 + Instrument | Subject)
##    Data: ratings2
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   7296.3   7438.9  -3622.2   7244.3     1757
##
## Scaled residuals:
```

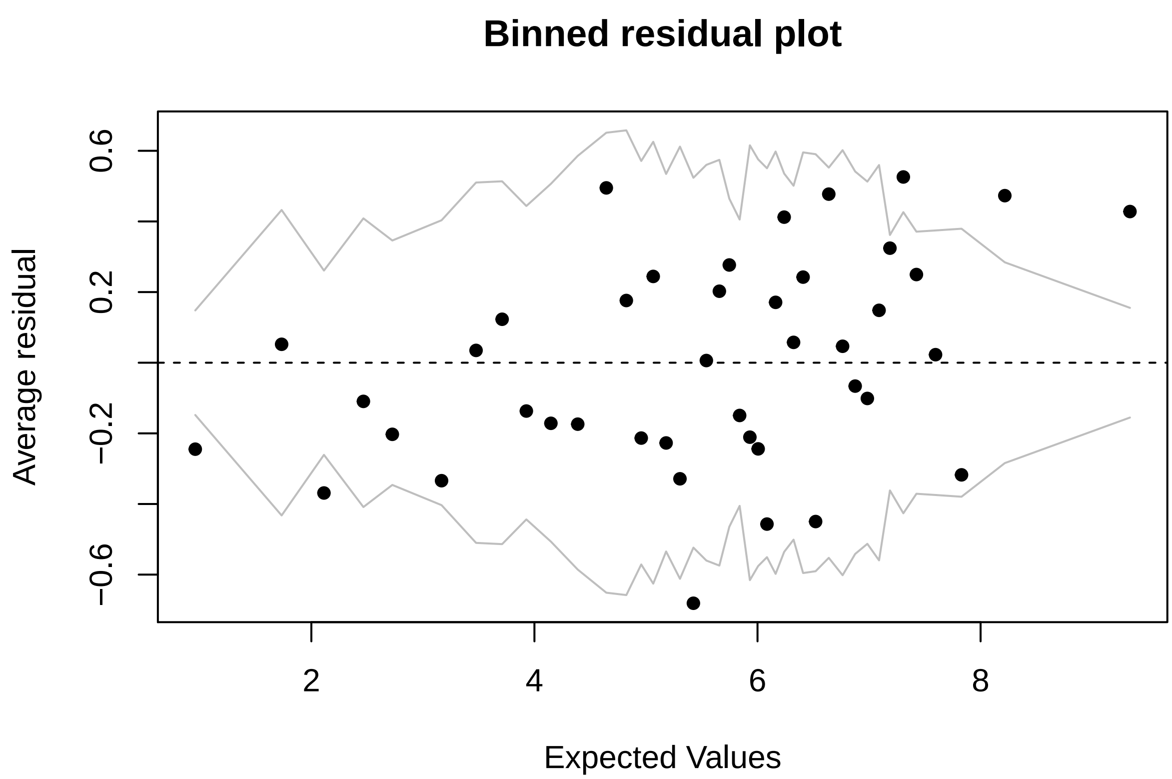
```
##     Min      1Q  Median      3Q     Max
## -3.5708 -0.6051  0.0168  0.6162  3.2564
##
## Random effects:
##  Groups    Name             Variance  Std.Dev.  Corr
##  Subject   (Intercept)      1.863e-11 4.317e-06
##  Subject.1 Voicecontrary    1.105e+00 1.051e+00
##            Voicepar3rd      8.106e-01 9.003e-01 1.00
##            Voicepar5th      1.099e+00 1.049e+00 0.99  1.00
##  Subject.2 Instrumentguitar 1.956e-01 4.423e-01
##            Instrumentpiano  1.359e+00 1.166e+00 -0.13
##            Instrumentstring 2.008e+00 1.417e+00 -0.87  0.59
##  Residual                   2.889e+00 1.700e+00
## Number of obs: 1783, groups:  Subject, 50
##
## Fixed effects:
##                         Estimate Std. Error         df t value Pr(>|t|)
## Voicecontrary           6.776e+00  2.456e-01  8.454e+01  27.591  < 2e-16
## Voicepar3rd             6.941e+00  2.331e-01  7.804e+01  29.773  < 2e-16
## Voicepar5th             6.990e+00  2.453e-01  8.194e+01  28.494  < 2e-16
## HarmonyI-V-IV          -1.310e-01  1.514e-01  1.584e+03  -0.865  0.38727
## HarmonyI-V-VI           8.033e-02  1.516e-01  1.584e+03   0.530  0.59623
## HarmonyIV-I-V          -2.540e-01  1.514e-01  1.584e+03  -1.677  0.09370
## musician                1.601e-03  3.326e-01  7.419e+01   0.005  0.99617
## Instrumentpiano        -1.093e+00  2.088e-01  4.983e+01  -5.236 3.29e-06
## Instrumentstring       -2.770e+00  2.750e-01  4.996e+01 -10.070 1.28e-13
## HarmonyI-V-IV:musician  1.869e-01  2.296e-01  1.585e+03   0.814  0.41579
## HarmonyI-V-VI:musician -7.550e-01  2.296e-01  1.585e+03  -3.289  0.00103
## HarmonyIV-I-V:musician  4.636e-02  2.294e-01  1.585e+03   0.202  0.83989
##
## Voicecontrary          ***
## Voicepar3rd            ***
## Voicepar5th            ***
## HarmonyI-V-IV
## HarmonyI-V-VI
## HarmonyIV-I-V          .
## musician
## Instrumentpiano        ***
## Instrumentstring       ***
## HarmonyI-V-IV:musician
## HarmonyI-V-VI:musician **
## HarmonyIV-I-V:musician
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             Vccntr Vcpr3r Vcpr5t HrI-V-IV HrI-V-VI HrIV-I-V musicn
## Voicepar3rd  0.912
## Voicepar5th  0.916  0.912
## HrmnyI-V-IV -0.308 -0.325 -0.309
## HrmnyI-V-VI -0.307 -0.324 -0.308  0.499
## HrmnyIV-I-V -0.308 -0.325 -0.309  0.500    0.499
## musician    -0.595 -0.627 -0.596  0.228    0.227    0.228
## Instrumntpn -0.196 -0.206 -0.197  0.000   -0.001    0.000    0.000
```

```
## Instrmntstr -0.292 -0.307 -0.292  0.000    0.000    0.000    0.000
## HrmnI-V-IV:  0.203  0.214  0.202 -0.659   -0.329   -0.330   -0.344
## HrmnI-V-VI:  0.202  0.214  0.202 -0.330   -0.660   -0.330   -0.344
## HrmnIV-I-V:  0.203  0.215  0.202 -0.330   -0.330   -0.660   -0.344
##             Instrmntp Instrmnts HI-V-IV: HI-V-VI:
## Voicepar3rd
## Voicepar5th
## HrmnyI-V-IV
## HrmnyI-V-VI
## HrmnyIV-I-V
## musician
## Instrumntpn
## Instrmntstr  0.707
## HrmnI-V-IV:  0.001     0.000
## HrmnI-V-VI:  0.001     0.000     0.499
## HrmnIV-I-V:  0.000     0.000     0.499    0.499
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

Check errors for Popular Model

We'll start by looking at the binned residuals

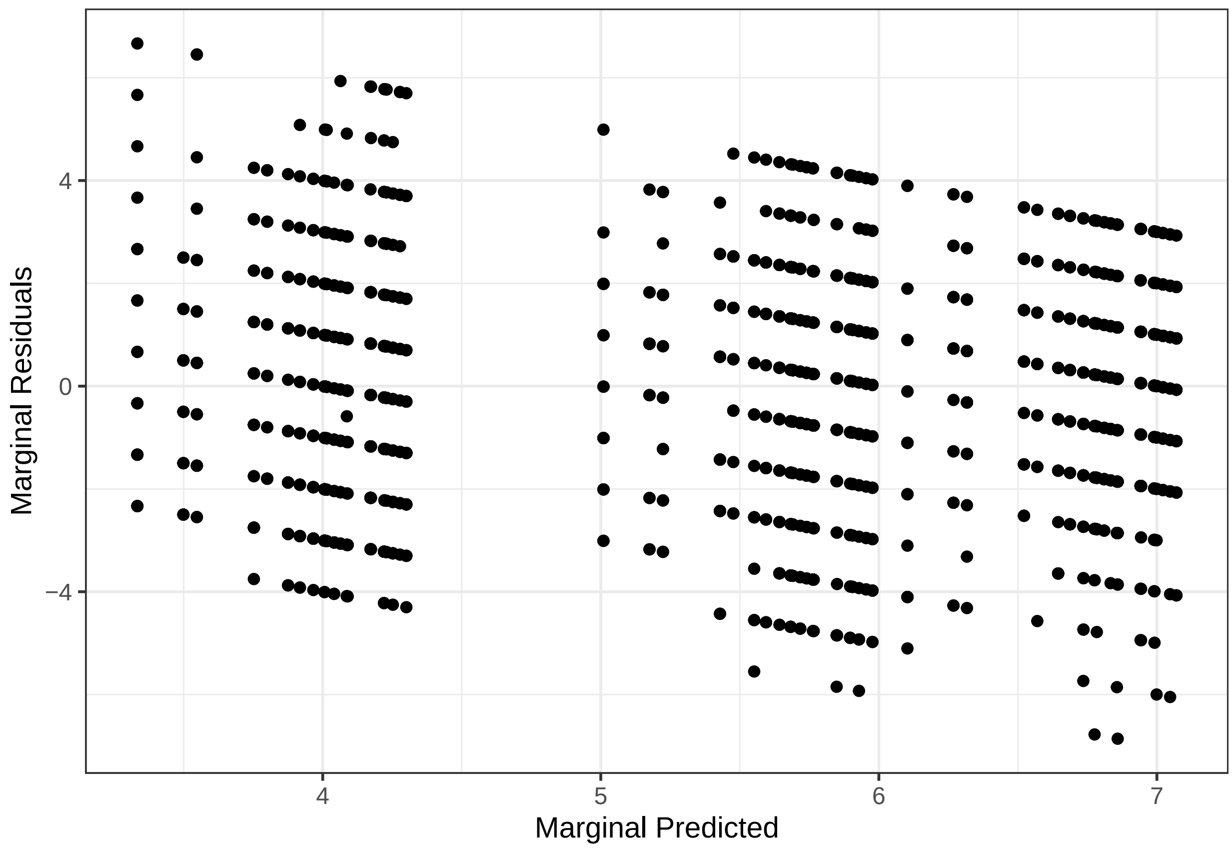
```
modelp <- lmer.pop.final
binnedplot(fitted(modelp),resid(modelp))
```



It appears that the majority of the residuals are within the bin.

Next, we look at marginal fitted values vs. residuals

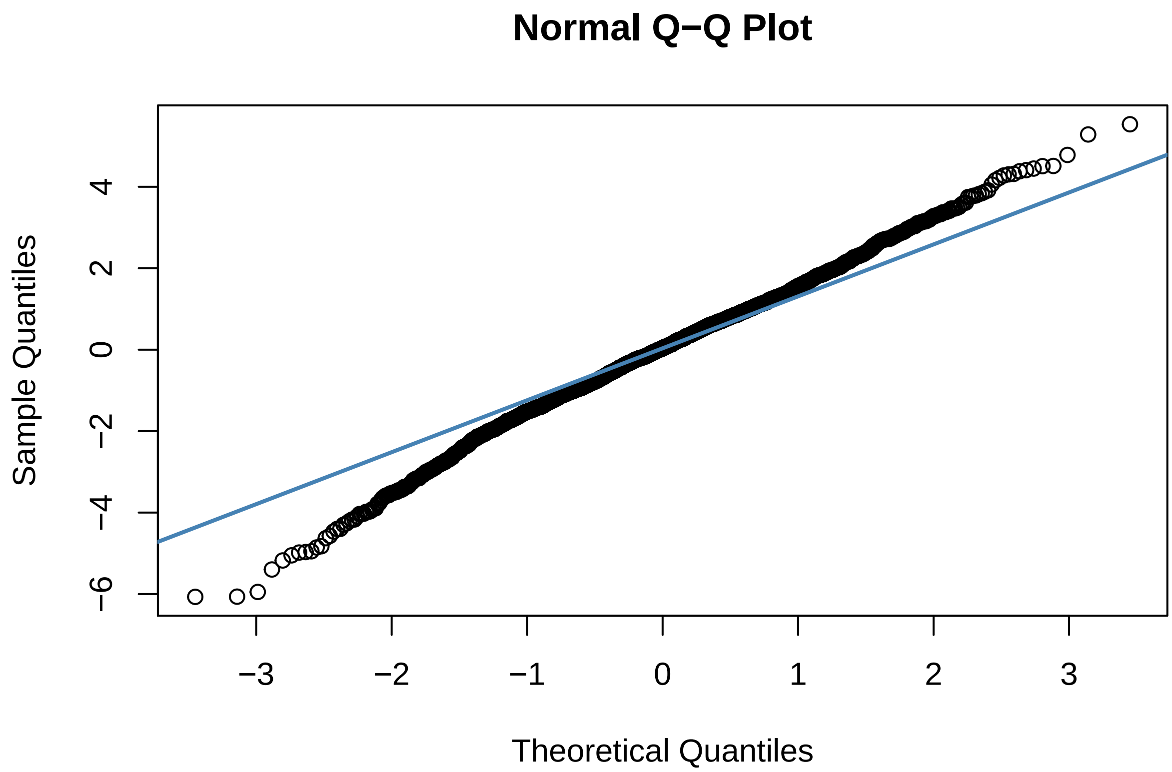
```
ggplot(mapping=aes(yhat.marg(modelp), r.marg(modelp))) + geom_point() +
  labs(x="Marginal Predicted", y="Marginal Residuals") + theme_bw()
```



In the marginal models residuals plot above, we don't care about trends, but are more focused on the spread of the points. Therefore, the marginal residuals plot above looks good.

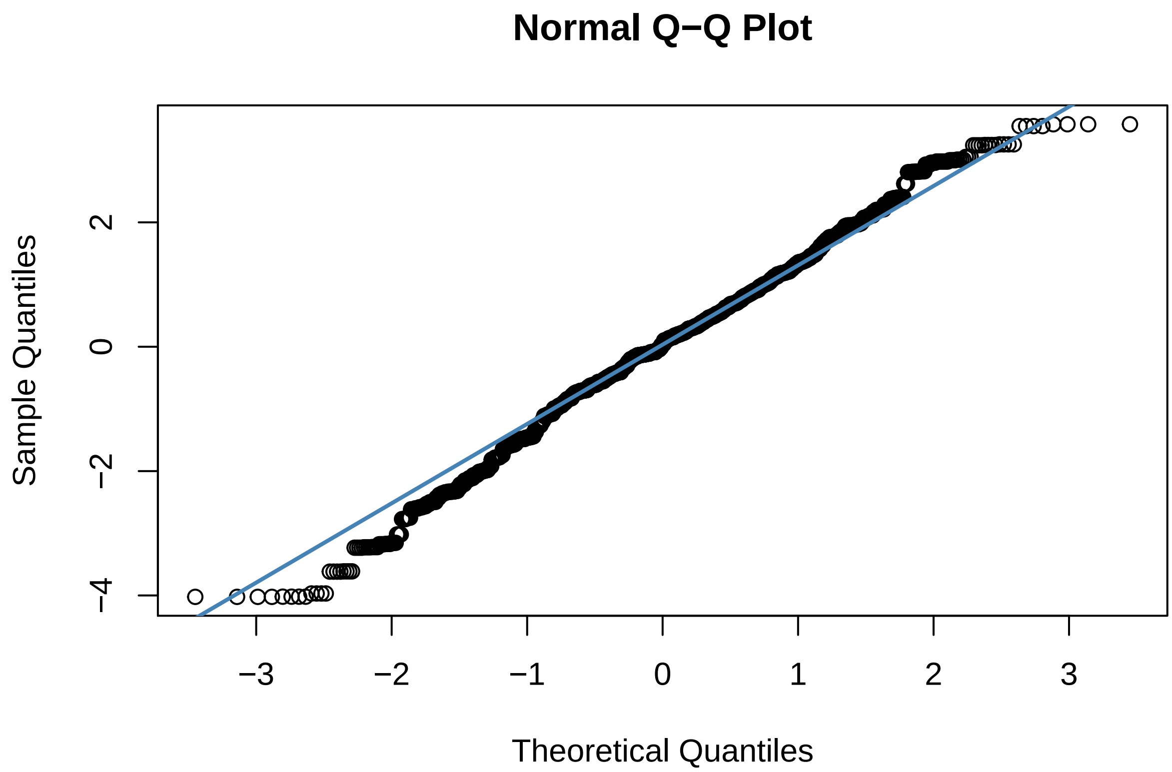
Now we will look at the QQ plot for conditional residuals.

```
qqnorm(r.cond(modelp))
qqline(r.reff(modelp), col = "steelblue", lwd = 2)
```



And the QQ plot for the random effects

```
qqnorm(r.reff(modelp))
qqline(r.reff(modelp), col = "steelblue", lwd = 2)
```



Both QQ plots look linear, suggesting normality of residuals.