What Effects People's Identification of Classical/Popular Music

Novak Xu*
lx2@andrew.cmu.edu

2 December 2019

Abstract

I addressed the question of what effects people's identification of classical/popular music. I examined data on Ivan Jimenez and Vincent Rossi's experiment on 70 listeners in 2012. I found that instrument is the most influential factor and some level of harmonic motion and voice leading are also influential. I also found there is difference in the way that musicians and non-musicians identify classical/popular music and most variables have opposite effect on identifying popular music compared to classical music. However, my analysis also had some weakness such as ignoring time effect which should be improved for further analysis.

1 Introduction

People listen to music everyday and obviously music has become one of the most common way of entertainment nowadays. Among all kinds of music, classical and popular music are two of the most important music types. Classical music is art music produced or rooted in the traditions of Western culture and popular music is music with wide appeal that is typically distributed to large audiences through the music industry.

In this study, I will focus on these two important kinds of music and analyze what factors effect people's judgement on how classical and popular does the music sound. I will address the following main research questions:

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

*Department of Statistics & Data Science, Carnegie Mellon University

2 Methods

The data for this study come from Ivan Jimenez, a composer and musicologist visiting the University of Pittsburgh[1] , and student Vincent Rossi. In 2012, they collected data in a designed experiment intended to measure the influence of instrument, harmonic motion, and voice leading on listeners' identification of music as "classical" or "popular". They presented 36 musical stimuli to 70 listeners, recruited from the population of undergraduates at the University of Pittsburgh, and asked the listeners to rate the music on how classical and popular does the music sound.

In all, Listeners were told that a piece could be rated as both classical and popular, neither classical nor popular, or mostly classical and not popular (or vice versa), so that the scales should have functioned more or less independently, and the following variables were measured on each:

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

[1]Now at the Sibelius Institute, University of the Arts, Helsinki Finland.

```
X1stInstr                               How proficient are you at your first
                                            musical instrument (0-5, 0=not at all)
X2ndInstr                               Same, for second musical instrument
```

The data is in the file “ratings.csv”in the hw10 area of the class website.

There are lots of missing values in the data. To deal with these missing values, firstly I counted how many data were missing for each column. The result is in Figure 1. Then for columns with more than 200 rows missing (I still kept *KnowAxis* for further inference), I deleted the whole columns because they had too many missing values. For the other columns, I deleted the rows with at least one missing value. In addition, the value of *Classical* and *Popular* shouldn't be larger than 10, but there are several 19 in the dataset. For these values, I just re-coded them to 10 because "9" is close to "0" on keyboard so it is easy type "9" mistakenly when you want to type "0". After that, I only have 1937 observations of 21 variables.

col_name	amount
ConsNotes	360
PachListen	72
ClsListen	36
KnowRob	180
KnowAxis	288
X1990s2000s	144
X1990s2000s.minus.1960s1970s	180
CollegeMusic	108
NoClass	288
APTheory	216
Composing	72
X1stInstr	1512
X2ndInstr	2196
Classical	27
Popular	27

Figure 1: Count of missing values for each column

For the analysis I relied on visual comparison of box plots and output of summary tables of models. I used R language and environment for statistical computing (R Core Team, 2017). For the modeling part, I used conventional linear models and analysis of variance models. I also used multi-level models to account for “personal biases” in ratings (random intercept and random slopes). In addition, I used backwards stepwise BIC to determine which individual covariates should be added to the model and used "LMERConvenienceFunctions" package in R automated backwards selection of fixed effects and forward selection of random effects, using "lltr" ("fitLMER.fnc()" is general-purpose function for this).

3 Results

3.1 EDA and Transformation

Before the analysis, I did EDA to help me have a better understanding of the data. I plotted the pairs of variables with absolute values of correlation bigger than 0.65 in Figure 2. The first plot

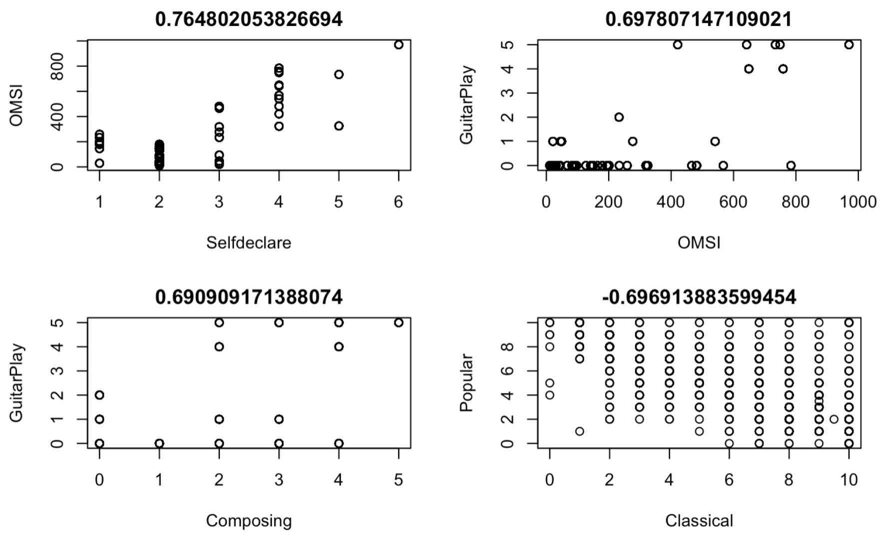


Figure 2: Correlation of Variables

has the biggest correlation which means people were "honest" enough when they self-evaluated themselves. What surprises me is the third plot, it seems like when you have played a long time of guitar, you will try to do some composing. I also used box plots to show relationship between ratings and three main effects. From Figure 3 we can find that *Instrument* has the strongest influence on ratings and for *Classical* and *Popular*, *Instrument* has totally opposite effect.

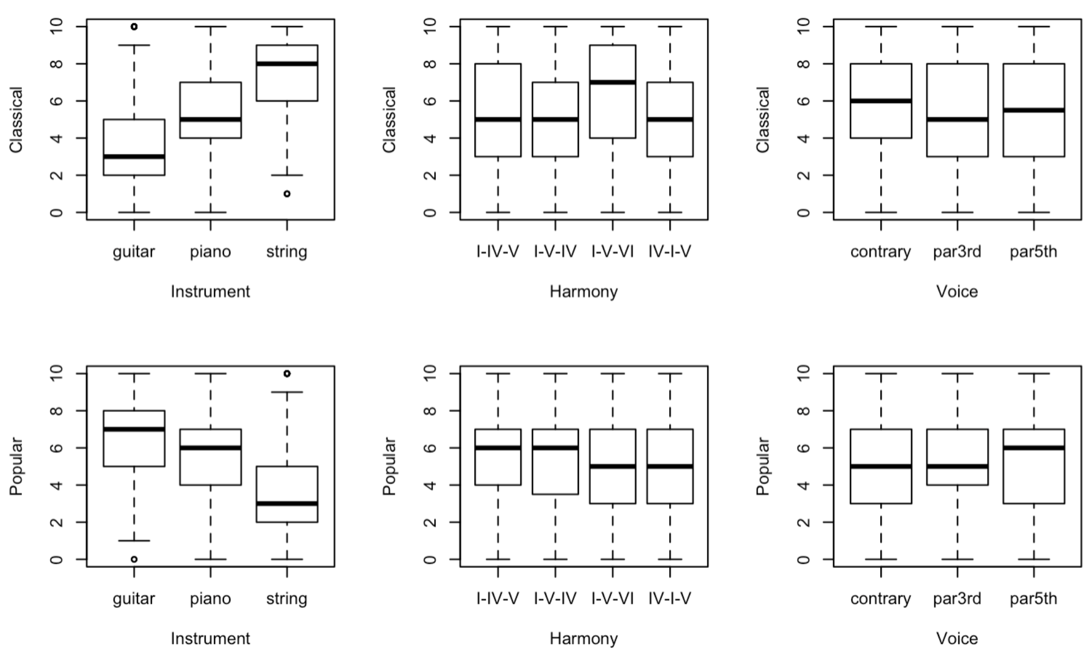


Figure 3: Box Plots Between Ratings and Three Main Effects

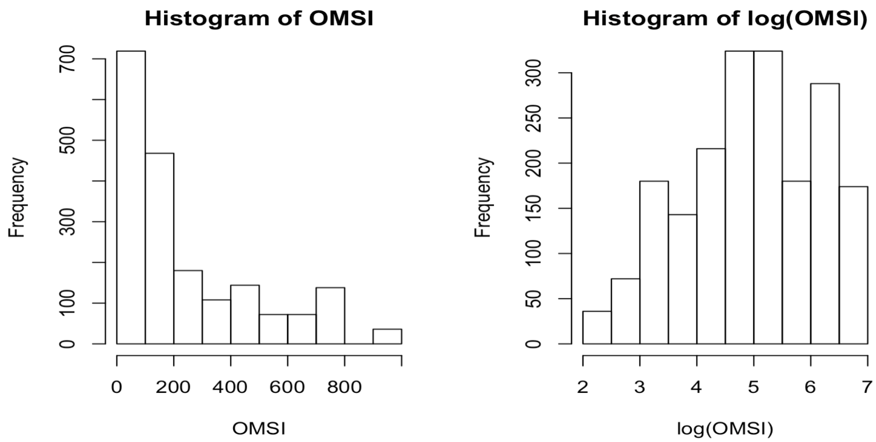


Figure 4: Transformation of $OMSI$

I also found $OMSI$ was right skewed. So for the further inference, I did a log-transformation and that improved the data a lot (Figure 4).

3.2 First Research Question

For the first research question, I fitted a linear model at first, where $Classical$ is response variable and $Instrument$, $Harmony$, $Voice$, $KnowRob$ and $KnowAxis$ are predictors. I also included the interaction between $Harmony$ and $Voice$ into the model because from the summary table of analysis of variance models for three main effects, I believed the interaction between $Harmony$ and $Voice$ was significant (see Page 4 in Appendix).

Secondly, I determined whether random intercept can account for "personal biases" in ratings, I used exact LRT test for REML fits, the result (see Page 4 in Appendix) was significant which means I should include random intercept into my model. Then I determined whether personal biases varied with the type of instrument, type of harmony, and/or type of voice leading. I tried all combinations of random slopes (see Page 5 in Appendix) and found that when I used $(1 + Harmony + Voice \mid Subject)$, the model has the smallest AIC and BIC at the same time.

Thirdly, I considered which individual covariates should be added into the model. I used backward stepwise BIC to do that(See Page 6 in Appendix). Once the fixed effects are settled, I went back and used "fitLMER.fnc()" to check whether there should be any change in the random effects(I always kept three main factors,$KnowAxis$, $KnowRob$ and interaction between $Harmony$ and $Voice$ during the whole process). The summary table of the final model is in Figure 5 (for details, see Page 7-13 in Appendix).

The residuals look good (See Page 15-19 in Appendix) so we believe this is a valid and reasonable model.

- For the first sub-question, from Figure 5, we can find that $Instrument$ is the most significant variable(with biggest t value) among three main variables which means it exerts the strongest influence among the three design factors. We can also prove that in Figure 3, $Instrument$ is the variable that separates ratings the most.

- For the second sub-question, from Figure 5, we can find that $HarmonyI - V - VI$ has the biggest coefficient and is the most significant level, so we can conclude that $HarmonyI - V - VI$ has the strongest association with classical ratings. We can also prove that in Figure 3, in the second plot, we can clearly find that $HarmonyI - V - VI$ is "outstanding" among all levels. For $KnowRob$ and $KnowAixs$, they are both not significant(t value smaller than 2) which means it doesn't seem to matter whether the respondent is familiar with one or the other (or both) of the Pachelbel rants/comedy bits.

- For the third sub-question, from Figure 5, we can find that $Voicecontrary$ has the biggest coefficient. However, no level of $Voice$ is significant (t value smaller than 2), so we can't conclude that among the levels of $Voice$, $Voicecontrary$ has a strong (the strongest?) association with $Classical$.

```
Random effects:
 Groups   Name             Variance Std.Dev. Corr
 Subject  (Intercept)      1.31441  1.1465
          Instrumentpiano  1.60450  1.2667   -0.30
          Instrumentstring 3.41539  1.8481   -0.43  0.59
          HarmonyI-V-IV    0.10139  0.3184    0.86 -0.67 -0.69
          HarmonyI-V-VI    1.82101  1.3494    0.00 -0.35 -0.60  0.10
          HarmonyIV-I-V    0.08691  0.2948   -0.26 -0.39 -0.28  0.14  0.19
 Residual                  2.38721  1.5451
Number of obs: 1937, groups:  Subject, 54

Fixed effects:
                              Estimate Std. Error t value
(Intercept)                  1.3277035  0.6556101   2.025
HarmonyI-V-IV               -0.1481481  0.1770571  -0.837
HarmonyI-V-VI                0.6305089  0.2516052   2.506
HarmonyIV-I-V                0.4112598  0.1765906   2.329
Instrumentpiano              1.5505439  0.1927371   8.045
Instrumentstring             3.4525474  0.2657492  12.992
Voicepar5th                  0.0791128  0.1719770   0.460
Voicecontrary                0.2592593  0.1716732   1.510
Selfdeclare                 -0.3142860  0.1540631  -2.040
Log.OMSI                     0.4930724  0.1459329   3.379
X16.minus.17                -0.0972952  0.0465948  -2.088
ClsListen                    0.3253091  0.0919280   3.539
KnowAxis                     0.0519033  0.0712765   0.728
KnowRob                     -0.0008695  0.0939755  -0.009
HarmonyI-V-IV:Voicepar5th    0.1676141  0.2431609   0.689
HarmonyI-V-VI:Voicepar5th    0.1761807  0.2432678   0.724
HarmonyIV-I-V:Voicepar5th   -0.3483973  0.2432557  -1.432
HarmonyI-V-IV:Voicecontrary  0.4160391  0.2429960   1.712
HarmonyI-V-VI:Voicecontrary  0.6491215  0.2433810   2.667
HarmonyIV-I-V:Voicecontrary -0.5594079  0.2429949  -2.302
```

Figure 5: Summary of the Model for the First Research Question

3.3 Second Research Question

For the second research question, I fitted analysis of variance model for each cut-off (I defined people with $Selfdeclare <$ cut-off as non-musician):

aov($Classical$ $\sim$$Musician$ * (Fixed Effects of Previous Model Except $Selfdeclasre$))

The result is in Table 1 (for more details, see Page 19-21 in Appendix) .

Cut-off	3	4	5
Variables that Significantly Interact with Musician	Harmony, Instrument, Log.OMSI, X16.minus.17	Harmony, Instrument, X16.minus.17, KnowRob	Harmony

Table 1: Variables that Significantly Interact with Musician for Different Cut-offs(Classical)

From the table we can easily find that no matter how cut-off changes, $Musician$ always has significant interaction with $Harmony$ which suggests that how "Musician" a person is will influence the impact of harmonic motion. In other words, $Musican$ reflects how professional and how much understanding people have in music. For the people who never learn music systematically or don't listen to music very often, the most obvious or the only feature they can notice or capture is which instrument is used. They may form a stereotype that electric guitar means popular music and piano means classical music. But for harmonic motion, it is hard to capture and understand unless you have some professional knowledge or a great sense of music and that is how we can distinguish musician and non-musician

Also, when we are more and more stricter(increase the cut-off), there is only one variable significantly interact with $Musician$ which suggests the result is sensitive with how we dichotomize $Musician$. For cut-off 3 and 4, $Instrument$ and $X16.minus.17$ are also significantly interact with $Musician$. We may conclude that these two variables are also very important to distinguish musician and non-musician.

3.4 Third Research Question

For the third research question, I just redid what I have done for $Classical$ on $Popular$ with the same methods and processes (See Page 21-32 in Appendix).

From Figure 6 we can easily find that for $Popular$, I dropped the interaction between $Harmony$ and $Voice$ which is the most obvious difference with $Classical$. From Figure 3 and 6 we can find $Instrument$ plays the most important role to identify classical and popular but it has totally opposite effect. For $HarmonyI-V-VI$, it has the strongest association with classical ratings but has the strongest association with popular ratings in the opposite direction. For $Voicecontrary$, it becomes significant so we can conclude that contrary motion has a strong (the strongest) association with popular ratings in the opposite direction.

We can also notice that $OMSI$ and $Selfdeclare$ don't show up in the summary table any more. Maybe that is because people have more access to popular music than classical music in daily life. For many people, the only access to classical music is in music classes or in some old movies. So it doesn't require much professional knowledge or specific training(that's what $OMSI$

and $Selfdeclare$ show) to identify popular music and that's why these two variable are not as significant as they are in the model of $Classical$.

```
Random effects:
 Groups   Name            Variance Std.Dev. Corr
 Subject  (Intercept)     1.4410   1.2004
          Instrumentpiano 1.4351   1.1980   -0.17
          Instrumentstring 2.7249  1.6507   -0.31  0.67
          HarmonyI-V-IV   0.1403   0.3746    0.53 -0.27 -0.29
          HarmonyI-V-VI   1.0733   1.0360   -0.11 -0.28 -0.35 -0.24
          HarmonyIV-I-V   0.3530   0.5941   -0.36 -0.33 -0.38 -0.52 -0.11
 Residual                 2.4863   1.5768
Number of obs: 1937, groups:  Subject, 54

Fixed effects:
                 Estimate Std. Error t value
(Intercept)       5.52711    0.60097   9.197
Instrumentpiano  -1.12733    0.18525  -6.085
Instrumentstring -2.95550    0.24113 -12.257
X16.minus.17      0.13865    0.05254   2.639
ConsInstr         0.16897    0.10017   1.687
Instr.minus.Notes -0.05843   0.09092  -0.643
X1990s2000s       0.14971    0.10213   1.466
KnowAxis          0.10300    0.07999   1.288
KnowRob          -0.00975    0.09959  -0.098
Voicepar5th       0.02261    0.08774   0.258
Voicecontrary    -0.18643    0.08778  -2.124
HarmonyI-V-IV    -0.03179    0.11342  -0.280
HarmonyI-V-VI    -0.35566    0.17366  -2.048
HarmonyIV-I-V    -0.26308    0.12958  -2.030
```

Figure 6: Summary of the Model for the *Popular*

Cut off	3	4	5
Variables that Significantly Interact with Musician	Instrument, X16.minus.17, ConsInstr, Instr.minus.Notes, X1990s2000s, KnowAxis, KnowRob, Harmony	Instrument, X16.minus.17, Instr.minus.Notes, Harmony	X16.minus.17, Harmony

Table 2: Variables that Significantly Interact with Musician for Different Cut-offs(Popular)

For $Musician$(Table 2), *Popular* is more sensitive to how we dichotomize $Musician$ and no matter how cut-off changes, $Musician$ always has significant interaction with not only $Harmony$, but also $X16.minus.17$. In all, for $Musician$, *Classical* and *Popular* show similar patterns and trend but the patterns and trend for *Popular* is more clear.

4 Discussion

The instrument, voice leading and harmonic motion appear in music and individual covariates will influence how people identify music types. In my exploratory analysis, I found that for classical music, instrument exerts the strongest influence among the three design factors (Instrument, Harmonic Motion, Voice Leading). In instrument, string helps people identify classical music the most while guitar helps people identify classical music the least. One specific harmonic motion: I-V-VI has the strongest with classical ratings. For contrary motion in voice leading and whether the respondent is familiar with one or the other (or both) of the Pachelbel rants/comedy bits, they don't have a strong association with classical ratings. For musician and non musician, harmonic motion will have different effects on them and as a result, influences how people identify classical music.

For popular music, it seems that most variables have totally opposite effect compared with classical music. Instrument still exerts the strongest influence among the three design factors but guitar helps people identify classical music the most while string helps people identify classical music the least. For I-V-VI and voice contrary, they have the strongest association with popular ratings in the opposite direction. I also found an interesting phenomenon that it doesn't require much professional knowledge or specific training to identify popular music. For musician and non-musician, the results are almost the same but popular music is more sensitive to how we dichotomize musician.

However, their experiment is limited by people's consistency on ratings. We assume that the response variables (ratings) are numeric which means we assume, for example, the difference between score 5 and 6 is equal to the difference between score 7 and score 8. But, it is nearly impossible to guarantee that people are always consistent about their ratings because there is no way to strictly quantify how classical or popular a music sounds like. In all, ratings is more subjective than objective which may cause bias to my analysis.

My analysis is also limited by ignoring one covariate: time effect. In the experiment, each participant was listened to 36 musical stimuli. It is possible that people got more and more familiar with how to identify classical/popular music or how to rate (objectively) when they were listened to more and more stimuli (practice makes perfect!). It is also possible that people got tired and exhausted after listening more than 15 stimuli and they didn't want to take the rest of ratings so seriously and carefully because that's too tired and annoying.

In summary, my analysis is a good advise to some music training institutions or college music classes. Not only just teaching knowledge about instruments, they can teach more about details in music such as voice leading and harmonic motion which can let students have a more comprehensive understanding or music. In order to improve our analysis, I should fix the above weakness for further study and they can also enlarge the scope of the study. For example, including jazz or punk, not just classical and popular music. They can also include other important aspects of music, such as major key and beats to fulfill the analysis.

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/.

Wikipedia contributors. (2019, December 6). Classical music. In $Wikipedia, The Free Encyclopedia$. Retrieved 21:22, December 6, 2019, from https://en.wikipedia.org/w/index.php?title=Classical_music&oldid=929475410

Wikipedia contributors. (2019, November 26). Popular music. In *Wikipedia, TheFreeEncyclopedia*. Retrieved 21:26, December 6, 2019, from https://en.wikipedia.org/w/index.php?title=Popular_music&oldid=927985986

Appendix

Novak Xu

12/4/2019

```r
data <- read.csv("~/Desktop/Applied Linear Model/hw10/ratings.csv")
data <- data %>% dplyr::select(- c(X, first12))
missing <- c()
for (i in 1:length(data)) {
  if(!all(!is.na(data[, colnames(data)[i]]))) {
    missing <- c(missing, colnames(data)[i])
  }
}
missing_data <- data.frame(col_name = missing, amount = rep(0, length(missing)))
for (i in 1:length(missing)) {
  missing_data[i, 2] = sum(is.na(data[, missing[i]]))
}
knitr::kable(missing_data, caption = "Misiing Data")
```

Table 1: Misiing Data

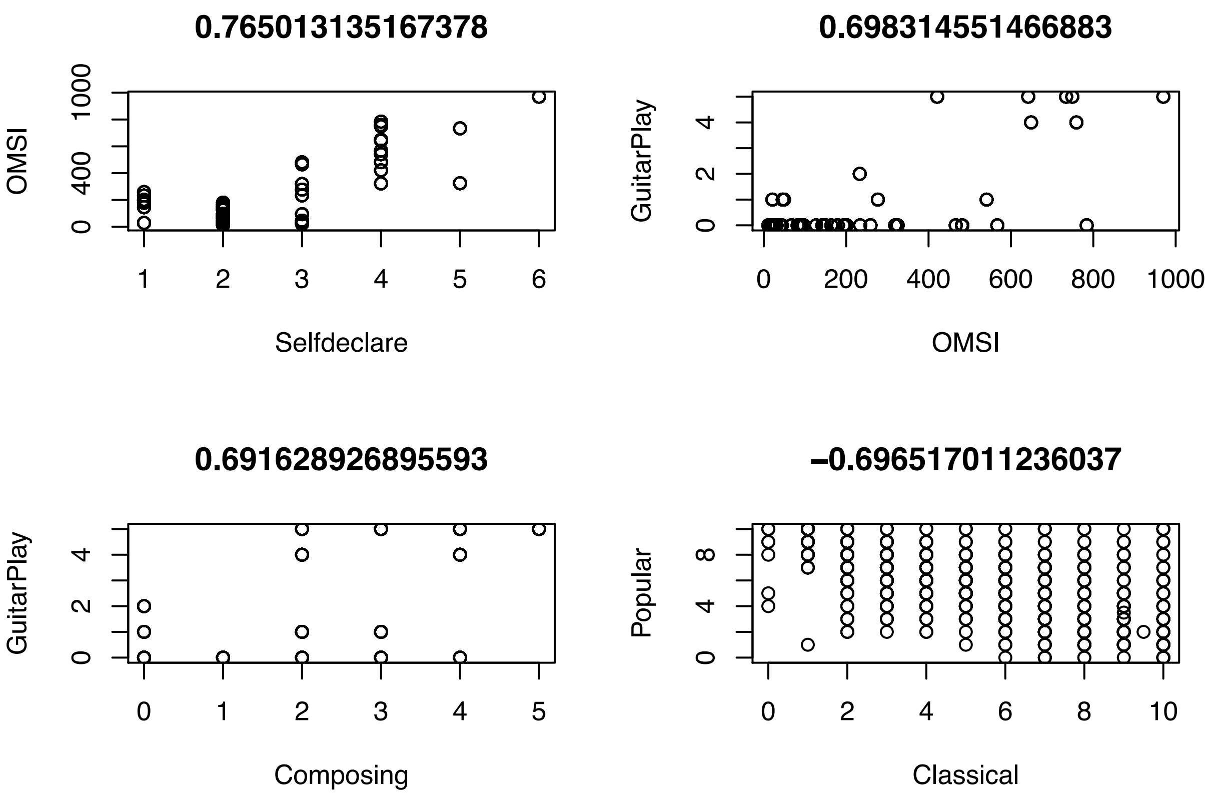
col_name	amount
ConsNotes	360
PachListen	72
ClsListen	36
KnowRob	180
KnowAxis	288
X1990s2000s	144
X1990s2000s.minus.1960s1970s	180
CollegeMusic	108
NoClass	288
APTheory	216
Composing	72
X1stInstr	1512
X2ndInstr	2196
Classical	27
Popular	27

```r
rm(i, missing, missing_data)
```

```r
data <- data %>% dplyr::select(- c(ConsNotes, NoClass, APTheory,
                                   X1stInstr, X2ndInstr)) %>%
  filter(!is.na(PachListen)) %>%
  filter(!is.na(ClsListen)) %>%
  filter(!is.na(KnowRob)) %>%
  filter(!is.na(KnowAxis)) %>%
  filter(!is.na(X1990s2000s)) %>%
  filter(!is.na(X1990s2000s.minus.1960s1970s)) %>%
  filter(!is.na(CollegeMusic)) %>%
  filter(!is.na(Composing)) %>%
  filter(!is.na(Classical))%>%
```

```
  filter(!is.na(Popular))
data$Classical[which(data$Classical == 19)] = 10
data$Popular[which(data$Popular == 19)] = 10
data$CollegeMusic <- as.factor(data$CollegeMusic)
data$Voice = factor(data$Voice, c("par3rd", "par5th", "contrary"))
```

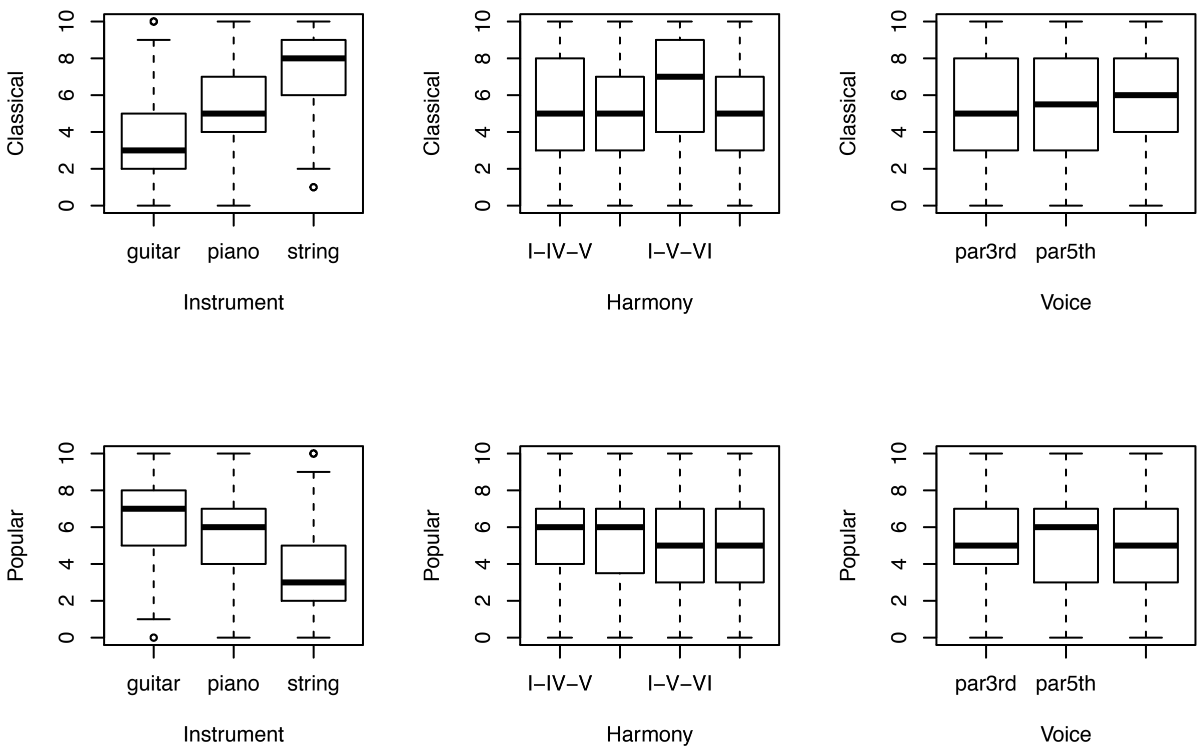
```
numeric <- c()
for (i in 1:length(data)) {
  if (is.numeric(data[, i])) {
    numeric <- c(numeric, i)
  }
}
par(mfrow = c(2, 2))
for (i in 1:(length(numeric) - 1)) {
  for (j in (1 + i):length(numeric)) {
    if (abs(cor(data[, numeric[i]], data[, numeric[j]])) > 0.65) {
      plot(data[, numeric[i]], data[, numeric[j]],
           main = cor(data[, numeric[i]], data[, numeric[j]]),
           xlab = colnames(data)[numeric[i]],
           ylab = colnames(data)[numeric[j]])
    }
  }
}
```



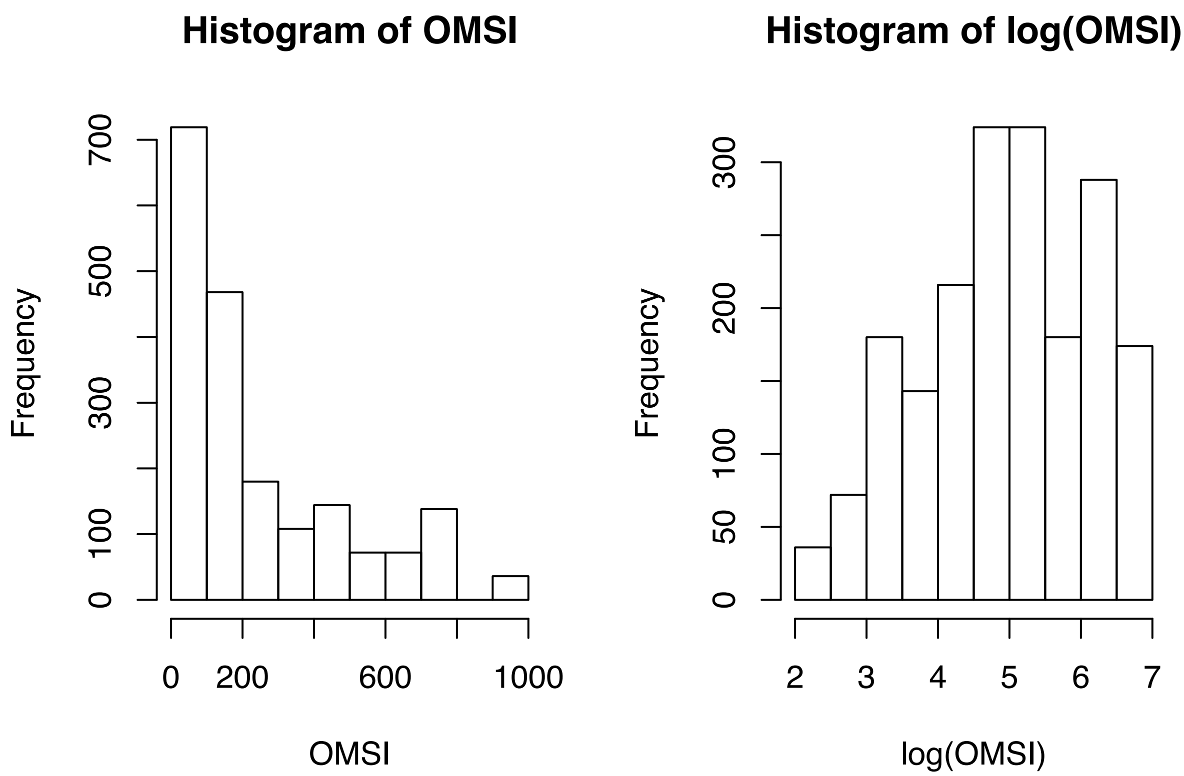

```
data$Log.OMSI = log(data$OMSI)
rm(i, j, numeric)
```

```
attach(data)
par(mfrow = c(2, 3))
```

```r
boxplot(Classical~Instrument)
boxplot(Classical~Harmony)
boxplot(Classical~Voice)
boxplot(Popular~Instrument)
boxplot(Popular~Harmony)
boxplot(Popular~Voice)
```




```r
par(mfrow = c(1, 2))
hist(OMSI, breaks = 10)
hist(log(OMSI), breaks = 15)
```




```r
model.1 <- aov(Classical ~ Instrument * Harmony * Voice)
summary(model.1)
```

```
##                           Df Sum Sq Mean Sq F value   Pr(>F)
## Instrument                 2   3873  1936.7 388.430  < 2e-16 ***
## Harmony                    3    262    87.4  17.520 3.21e-11 ***
## Voice                      2     53    26.7   5.363  0.00476 **
## Instrument:Harmony         6     19     3.1   0.622  0.71326
## Instrument:Voice           4     21     5.2   1.040  0.38496
## Harmony:Voice              6     71    11.8   2.359  0.02833 *
## Instrument:Harmony:Voice  12     65     5.4   1.086  0.36733
## Residuals               1901   9478     5.0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.2 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1|Subject), REML = F, control = lmerControl(optimizer = 'bobyqa'))
exactRLRT(model.2)
```

```
## Using restricted likelihood evaluated at ML estimators.

## Refit with method="REML" for exact results.

##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 459.66, p-value < 2.2e-16
```

```r
model.3 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Instrument|Subject), REML = F, control = lmerControl(optimizer = 'bobyqa'))
model.4 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice+ KnowAxis + KnowRob +
                  (1 + Harmony|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa'))
```

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

```r
model.5 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa'))
```

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

```r
model.6 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyq
```

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

```r
model.7 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Harmony|Subject), REML = F,  control = lmerControl(optimizer = 'bob
```

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

```r
model.8 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Harmony + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa'))
```

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

```r
model.9 <- lmer(Classical ~ 1 + Instrument + Harmony +
                  Voice + Harmony:Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Harmony + Voice|Subject), REML = F,  control = lmerControl(optimize
```

```
## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.
```

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

```r
anova(model.2, model.3, model.4, model.5, model.6, model.7, model.8, model.9)
```

```
## Data: NULL
## Models:
## model.2: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.2:     KnowAxis + KnowRob + (1 | Subject)
## model.3: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.3:     KnowAxis + KnowRob + (1 + Instrument | Subject)
## model.5: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.5:     KnowAxis + KnowRob + (1 + Voice | Subject)
## model.4: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.4:     KnowAxis + KnowRob + (1 + Harmony | Subject)
## model.6: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.6:     KnowAxis + KnowRob + (1 + Instrument + Voice | Subject)
## model.7: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.7:     KnowAxis + KnowRob + (1 + Instrument + Harmony | Subject)
## model.8: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
```

```
## model.8:     KnowAxis + KnowRob + (1 + Harmony + Voice | Subject)
## model.9: Classical ~ 1 + Instrument + Harmony + Voice + Harmony:Voice +
## model.9:     KnowAxis + KnowRob + (1 + Instrument + Harmony + Voice |
## model.9:     Subject)
##         Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## model.2 18 8146.9 8247.1 -4055.4   8110.9
## model.3 23 7859.3 7987.4 -3906.7   7813.3 297.543      5     <2e-16 ***
## model.5 23 8156.9 8285.0 -4055.4   8110.9   0.000      0          1
## model.4 27 8072.0 8222.4 -4009.0   8018.0  92.866      4     <2e-16 ***
## model.6 32 7871.2 8049.4 -3903.6   7807.2 210.772      5     <2e-16 ***
## model.7 38 7718.0 7929.6 -3821.0   7642.0 165.226      6     <2e-16 ***
## model.8 38 8084.8 8296.4 -4004.4   8008.8   0.000      0          1
## model.9 53 7727.5 8022.7 -3810.8   7621.5 387.290     15     <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.10 <- lm(Classical ~ Harmony + Instrument + Voice + Selfdeclare + Log.OMSI +
                 X16.minus.17 + ConsInstr + Instr.minus.Notes +
                 PachListen + ClsListen + KnowRob + X1990s2000s +
                X1990s2000s.minus.1960s1970s + CollegeMusic +
                Composing + PianoPlay + GuitarPlay + Harmony:Voice + KnowAxis)
stepAIC(model.10, direction = "backward", k = log(2044), trace = F)
```

```
##
## Call:
## lm(formula = Classical ~ Harmony + Instrument + Selfdeclare +
##     Log.OMSI + X16.minus.17 + Instr.minus.Notes + ClsListen +
##     X1990s2000s.minus.1960s1970s + CollegeMusic + Composing +
##     PianoPlay)
##
## Coefficients:
##                  (Intercept)                 HarmonyI-V-IV
##                      2.32382                       0.04675
##                 HarmonyI-V-VI                 HarmonyIV-I-V
##                      0.89756                       0.10722
##               Instrumentpiano              Instrumentstring
##                      1.54546                       3.45094
##                  Selfdeclare                      Log.OMSI
##                     -0.55509                       0.38253
##                 X16.minus.17             Instr.minus.Notes
##                     -0.09668                       0.12436
##                    ClsListen  X1990s2000s.minus.1960s1970s
##                      0.30647                       0.13472
##                 CollegeMusic1                     Composing
##                     -0.43605                       0.21156
##                    PianoPlay
##                      0.17384
```

```
model.12 <- lmer(Classical ~ Harmony + Instrument + Voice +
                   Harmony:Voice + Selfdeclare + Log.OMSI +
                   X16.minus.17 + Instr.minus.Notes + ClsListen +
                   X1990s2000s.minus.1960s1970s + CollegeMusic +
                   Composing + PianoPlay + KnowAxis + KnowRob +
                  (1 + Instrument|Subject) + (0 + Harmony|Subject), REML = F,
              data = data, 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

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

```
model.13 <- fitLMER.fnc(model.12, ran.effects = c("(0 + Voice|Subject)"), method = "llrt")
```

```
## ======================================================
## ===              backfitting fixed effects         ===
## ======================================================
## setting REML to FALSE

## 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

## processing model terms of interaction level 2
##   iteration 1
##     p-value for term "Harmony:Voice" = 1e-04 >= 0
##     not part of higher-order interaction

## boundary (singular) fit: see ?isSingular

##     log-likelihood ratio test p-value = 7.090844e-05 <= 0.05
##     skipping term
## length = 1
## processing model terms of interaction level 1
##   iteration 2
##     p-value for term "Selfdeclare" = 0.571 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.004988347 <= 0.05
##     skipping term
## length = 14
##   iteration 3
##     p-value for term "CollegeMusic" = 0.5408 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p.value = 0.3265016 > 0.05
##     removing term
##   iteration 4
##     p-value for term "Selfdeclare" = 0.5851 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.005676652 <= 0.05
##     skipping term
```

```
## length = 13
##   iteration 5
##     p-value for term "KnowAxis" = 0.5139 >= 0
##     not part of higher-order interaction

## boundary (singular) fit: see ?isSingular

##     log-likelihood ratio test p.value = 0.8492063 > 0.05
##     removing term
##   iteration 6
##     p-value for term "Selfdeclare" = 0.5704 >= 0
##     not part of higher-order interaction
##     log-likelihood ratio test p-value = 0.00551967 <= 0.05
##     skipping term
## length = 12
##   iteration 7
##     p-value for term "Instr.minus.Notes" = 0.4595 >= 0
##     not part of higher-order interaction

## boundary (singular) fit: see ?isSingular

##     log-likelihood ratio test p.value = 0.5572359 > 0.05
##     removing term
##   iteration 8
##     p-value for term "Selfdeclare" = 0.492 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p-value = 0.002779225 <= 0.05
##     skipping term
## length = 11
##   iteration 9
##     p-value for term "Composing" = 0.377 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p.value = 0.2143154 > 0.05
##     removing term
##   iteration 10
##     p-value for term "Selfdeclare" = 0.442 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.00650246 <= 0.05
##     skipping term
## length = 10
##   iteration 11
```

```
##     p-value for term "KnowRob" = 0.3015 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p.value = 0.3471132 > 0.05
##     removing term
##   iteration 12
##     p-value for term "Selfdeclare" = 0.5208 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p-value = 0.008290622 <= 0.05
##     skipping term
## length = 9
##   iteration 13
##     p-value for term "X1990s2000s.minus.1960s1970s" = 0.1065 >= 0
##     not part of higher-order interaction
##     log-likelihood ratio test p.value = 0.1270922 > 0.05
##     removing term
##   iteration 14
##     p-value for term "Selfdeclare" = 0.5647 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.02309739 <= 0.05
##     skipping term
## length = 8
##   iteration 15
##     p-value for term "PianoPlay" = 0.0615 >= 0
##     not part of higher-order interaction
##     log-likelihood ratio test p.value = 0.08215948 > 0.05
##     removing term
##   iteration 16
##     p-value for term "Selfdeclare" = 0.5369 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p.value = 0.05053957 > 0.05
```

```
##     removing term
##   iteration 17
##     p-value for term "X16.minus.17" = 0.021 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.02895717 <= 0.05
##     skipping term
## length = 6
##   iteration 18
##     p-value for term "Harmony" = 2e-04 >= 0
##     part of higher-order interaction
##     skipping term
##   iteration 19
##     p-value for term "Log.OMSI" = 0.0058 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p-value = 0.01126871 <= 0.05
##     skipping term
## length = 4
##   iteration 20
##     p-value for term "ClsListen" = 0.0018 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.005983524 <= 0.05
##     skipping term
## length = 3
##   iteration 21
##     p-value for term "Voice" = 0 >= 0
##     part of higher-order interaction
##     skipping term
##   iteration 22
##     p-value for term "Instrument" = 0 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p-value = 2.241259e-17 <= 0.05
##     skipping term
## length = 1
## pruning random effects structure ...
```

```
##   nothing to prune
## ========================================================
## ===            forwardfitting random effects       ===
## ========================================================
## evaluating addition of (0+Voice|Subject) to model
## boundary (singular) fit: see ?isSingular
##  log-likelihood ratio test p-value = 0.7019451
##  not adding (0+Voice|Subject) to model
## ========================================================
## ===            re-backfitting fixed effects        ===
## ========================================================
## setting REML to FALSE
## 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
## processing model terms of interaction level 2
##   iteration 1
##     p-value for term "Harmony:Voice" = 1e-04 >= 0
##     not part of higher-order interaction
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?
##     log-likelihood ratio test p-value = 7.138219e-05 <= 0.05
##     skipping term
## length = 1
## processing model terms of interaction level 1
##   iteration 2
##     p-value for term "X16.minus.17" = 0.021 >= 0
##     not part of higher-order interaction
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unide
##  - Rescale variables?
##     log-likelihood ratio test p-value = 0.02895717 <= 0.05
##     skipping term
## length = 6
##   iteration 3
##     p-value for term "Harmony" = 2e-04 >= 0
##     part of higher-order interaction
##     skipping term
##   iteration 4
##     p-value for term "Log.OMSI" = 0.0058 >= 0
##     not part of higher-order interaction
## 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
```

```
##     log-likelihood ratio test p-value = 0.01126871 <= 0.05
##     skipping term
## length = 4
##   iteration 5
##     p-value for term "ClsListen" = 0.0018 >= 0
##     not part of higher-order interaction

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?

##     log-likelihood ratio test p-value = 0.005983524 <= 0.05
##     skipping term
## length = 3
##   iteration 6
##     p-value for term "Voice" = 0 >= 0
##     part of higher-order interaction
##     skipping term
##   iteration 7
##     p-value for term "Instrument" = 0 >= 0
##     not part of higher-order interaction

## 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

##     log-likelihood ratio test p-value = 2.241259e-17 <= 0.05
##     skipping term
## length = 1
## resetting REML to TRUE

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?

## pruning random effects structure ...
##   nothing to prune
## log file is /var/folders/w4/kmmyx3455x77cwww_8985p140000gn/T//Rtmp57wfo5/fitLMER_log_Fri_Dec__6_17-2
```

```
summary(model.13)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## Classical ~ Harmony + Instrument + Voice + Log.OMSI + X16.minus.17 +
##     ClsListen + (1 + Instrument | Subject) + (0 + Harmony | Subject) +
##     Harmony:Voice
##    Data: data
## Control: lmerControl(optimizer = "bobyqa")
##
## REML criterion at convergence: 7685.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.7691 -0.5734  0.0192  0.5445  3.6248
##
## Random effects:
##  Groups    Name             Variance Std.Dev. Corr
```

```
##  Subject   (Intercept)      1.1292   1.0627
##            Instrumentpiano  1.6383   1.2800   -0.48
##            Instrumentstring 3.4816   1.8659   -0.76  0.59
##  Subject.1 HarmonyI-IV-V    0.9339   0.9664
##            HarmonyI-V-IV    1.2715   1.1276   0.99
##            HarmonyI-V-VI    1.2314   1.1097   0.15  0.26
##            HarmonyIV-I-V    0.6674   0.8170   0.96  0.97  0.06
##  Residual                   2.4079   1.5517
## Number of obs: 1937, groups:  Subject, 54
##
## Fixed effects:
##                             Estimate Std. Error t value
## (Intercept)                  1.47682    0.67251   2.196
## HarmonyI-V-IV               -0.14815    0.17521  -0.846
## HarmonyI-V-VI                0.62936    0.25269   2.491
## HarmonyIV-I-V                0.41160    0.17709   2.324
## Instrumentpiano              1.54965    0.19452   7.967
## Instrumentstring             3.45236    0.26817  12.874
## Voicepar5th                  0.07992    0.17272   0.463
## Voicecontrary                0.25926    0.17241   1.504
## Log.OMSI                     0.34128    0.12343   2.765
## X16.minus.17                -0.11307    0.04835  -2.338
## ClsListen                    0.26751    0.08888   3.010
## HarmonyI-V-IV:Voicepar5th    0.16596    0.24421   0.680
## HarmonyI-V-VI:Voicepar5th    0.17652    0.24431   0.723
## HarmonyIV-I-V:Voicepar5th   -0.34955    0.24431  -1.431
## HarmonyI-V-IV:Voicecontrary  0.41519    0.24405   1.701
## HarmonyI-V-VI:Voicecontrary  0.64986    0.24443   2.659
## HarmonyIV-I-V:Voicecontrary -0.55975    0.24405  -2.294

##
## Correlation matrix not shown by default, as p = 17 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it

## convergence code: 0
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
model.14 <- lmer(Classical ~ Harmony + Instrument + Voice +
                   Harmony:Voice + Selfdeclare +
                   Log.OMSI + X16.minus.17 + ClsListen +
                   KnowAxis + KnowRob +
                  (1 + Instrument + Harmony|Subject), REML = F,
              data = data, control = lmerControl(optimizer = 'bobyqa'))
```

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

```
summary(model.14)
```

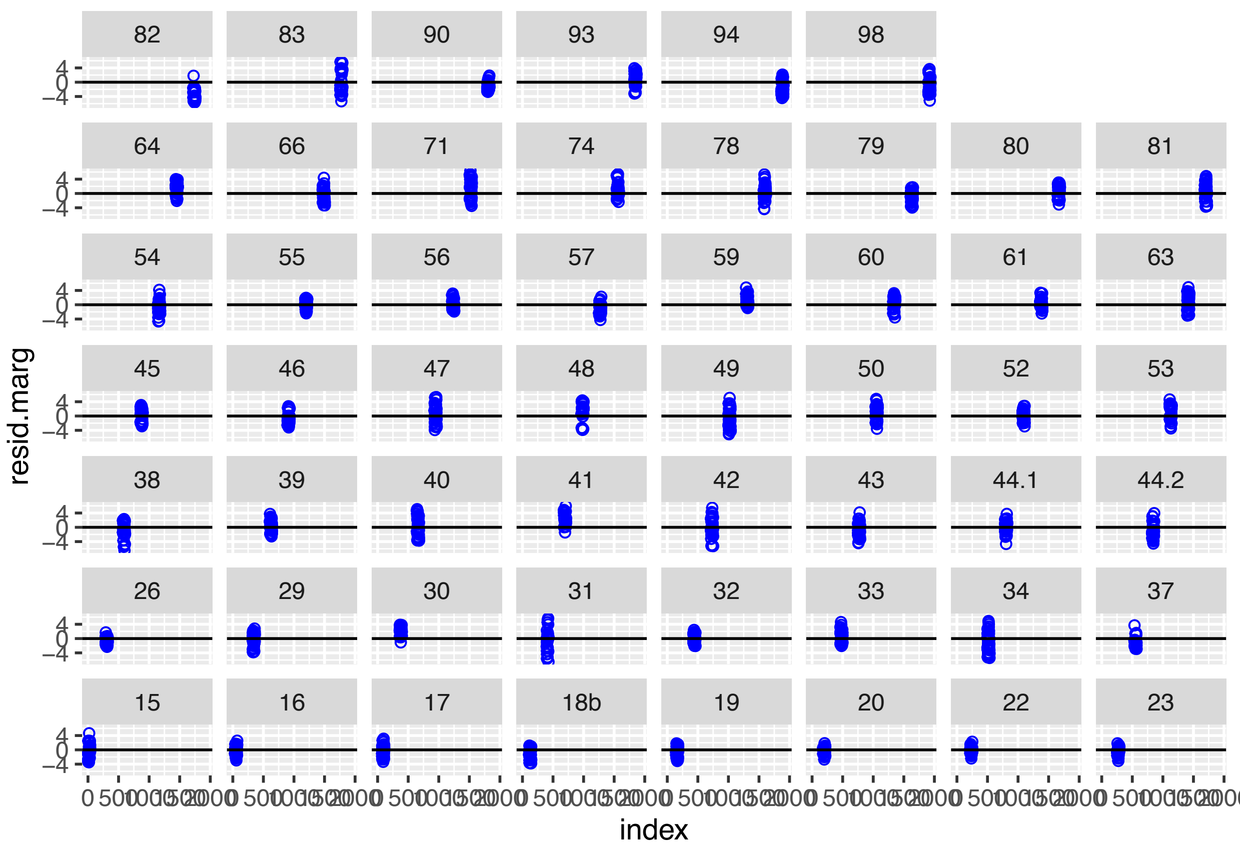
```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Classical ~ Harmony + Instrument + Voice + Harmony:Voice + Selfdeclare +
##     Log.OMSI + X16.minus.17 + ClsListen + KnowAxis + KnowRob +
##     (1 + Instrument + Harmony | Subject)
##    Data: data
```

```
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   7709.7   7943.6  -3812.8   7625.7     1895
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.6818 -0.5738  0.0115  0.5470  3.5709
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.31441  1.1465
##           Instrumentpiano  1.60450  1.2667   -0.30
##           Instrumentstring 3.41539  1.8481   -0.43  0.59
##           HarmonyI-V-IV    0.10139  0.3184    0.86 -0.67 -0.69
##           HarmonyI-V-VI    1.82101  1.3494    0.00 -0.35 -0.60  0.10
##           HarmonyIV-I-V    0.08691  0.2948   -0.26 -0.39 -0.28  0.14  0.19
##  Residual                  2.38721  1.5451
## Number of obs: 1937, groups:  Subject, 54
##
## Fixed effects:
##                               Estimate Std. Error t value
## (Intercept)                  1.3277035  0.6556101   2.025
## HarmonyI-V-IV               -0.1481481  0.1770571  -0.837
## HarmonyI-V-VI                0.6305089  0.2516052   2.506
## HarmonyIV-I-V                0.4112598  0.1765906   2.329
## Instrumentpiano              1.5505439  0.1927371   8.045
## Instrumentstring             3.4525474  0.2657492  12.992
## Voicepar5th                  0.0791128  0.1719770   0.460
## Voicecontrary                0.2592593  0.1716732   1.510
## Selfdeclare                 -0.3142860  0.1540631  -2.040
## Log.OMSI                     0.4930724  0.1459329   3.379
## X16.minus.17                -0.0972952  0.0465948  -2.088
## ClsListen                    0.3253091  0.0919280   3.539
## KnowAxis                     0.0519033  0.0712765   0.728
## KnowRob                     -0.0008695  0.0939755  -0.009
## HarmonyI-V-IV:Voicepar5th    0.1676141  0.2431609   0.689
## HarmonyI-V-VI:Voicepar5th    0.1761807  0.2432678   0.724
## HarmonyIV-I-V:Voicepar5th   -0.3483973  0.2432557  -1.432
## HarmonyI-V-IV:Voicecontrary  0.4160391  0.2429960   1.712
## HarmonyI-V-VI:Voicecontrary  0.6491215  0.2433810   2.667
## HarmonyIV-I-V:Voicecontrary -0.5594079  0.2429949  -2.302

##
## Correlation matrix not shown by default, as p = 20 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it

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

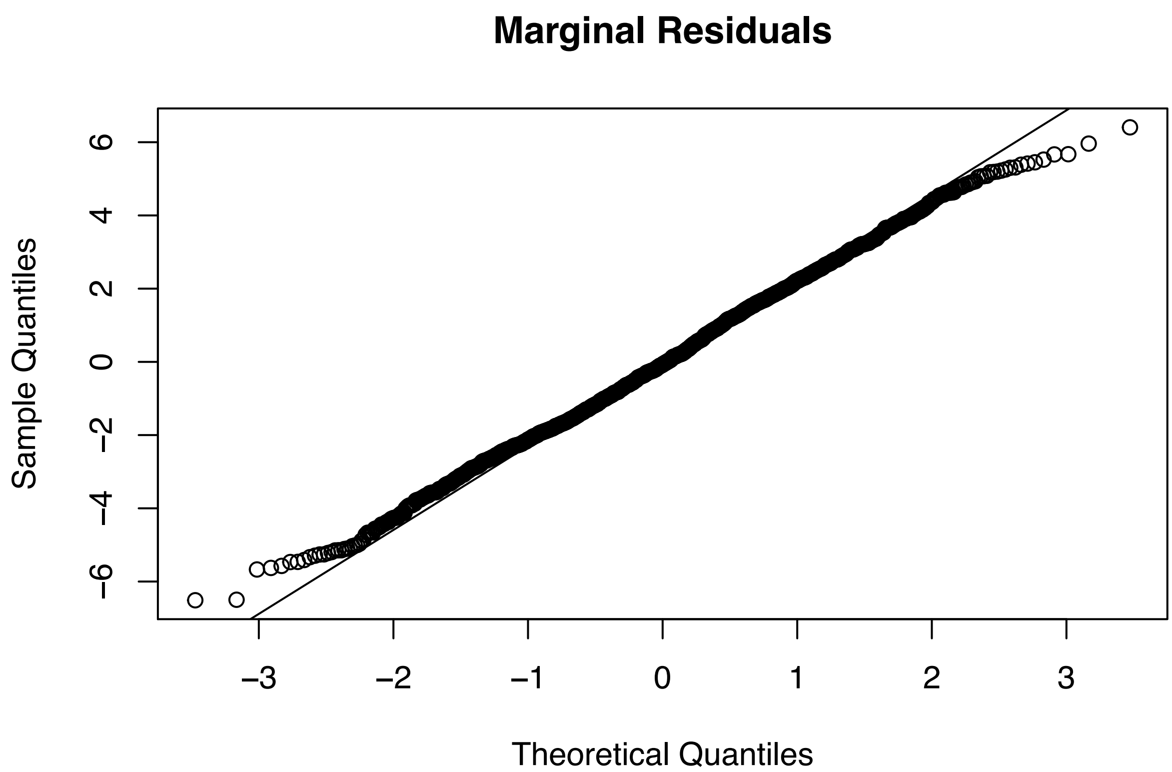
```r
resid.marg <- r.marg(model.14)
resid.cond <- r.cond(model.14)
resid.reff <- r.reff(model.14)
```

```
attach(data)

## The following objects are masked from data (pos = 3):
##
##     Classical, ClsListen, CollegeMusic, Composing, ConsInstr,
##     GuitarPlay, Harmony, Instr.minus.Notes, Instrument, KnowAxis,
##     KnowRob, Log.OMSI, OMSI, PachListen, PianoPlay, Popular,
##     Selfdeclare, Subject, Voice, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s
```

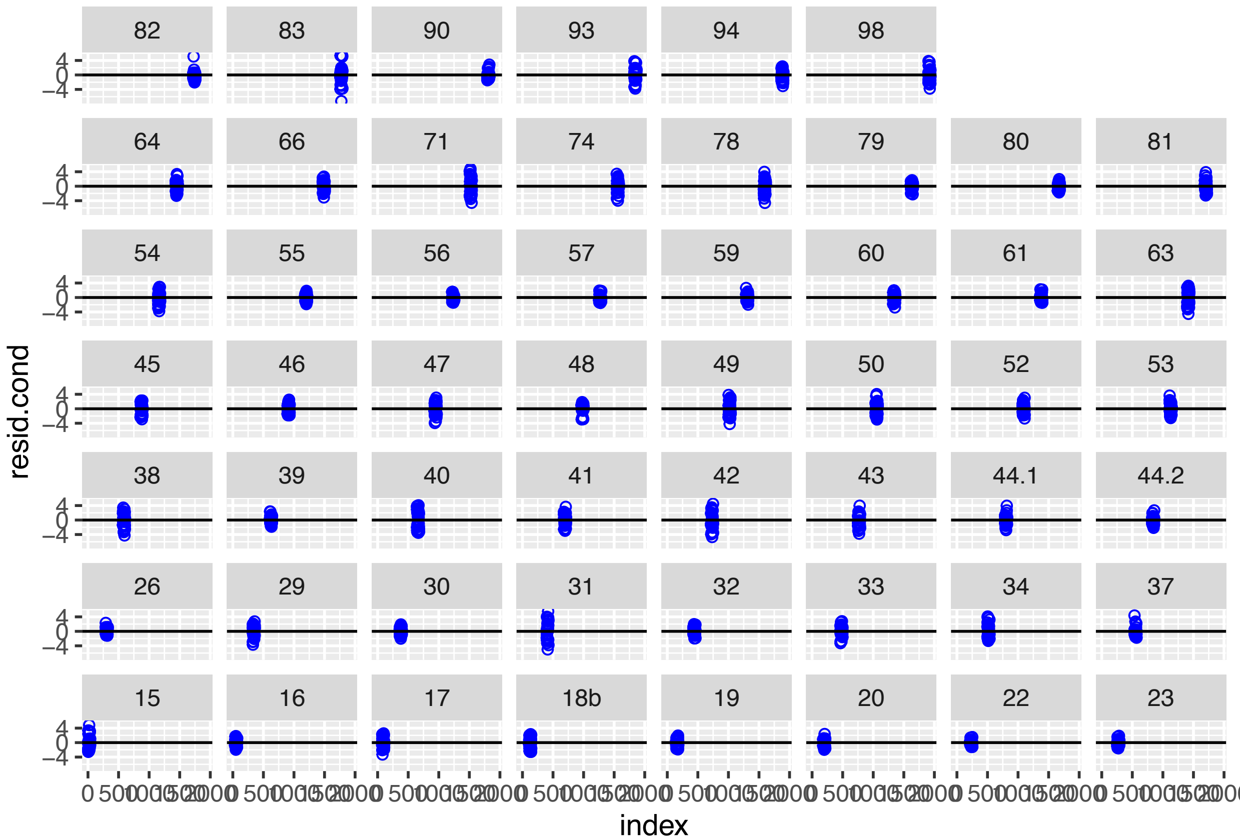
```
index <- 1:dim(data)[1]

new.data <- data.frame(index, resid.marg, Subject)
names(new.data) <- c("index","resid.marg","Subject")
ggplot(new.data,aes(x = index, y = resid.marg)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1,color = "Blue") +
    geom_hline(yintercept = 0)
```



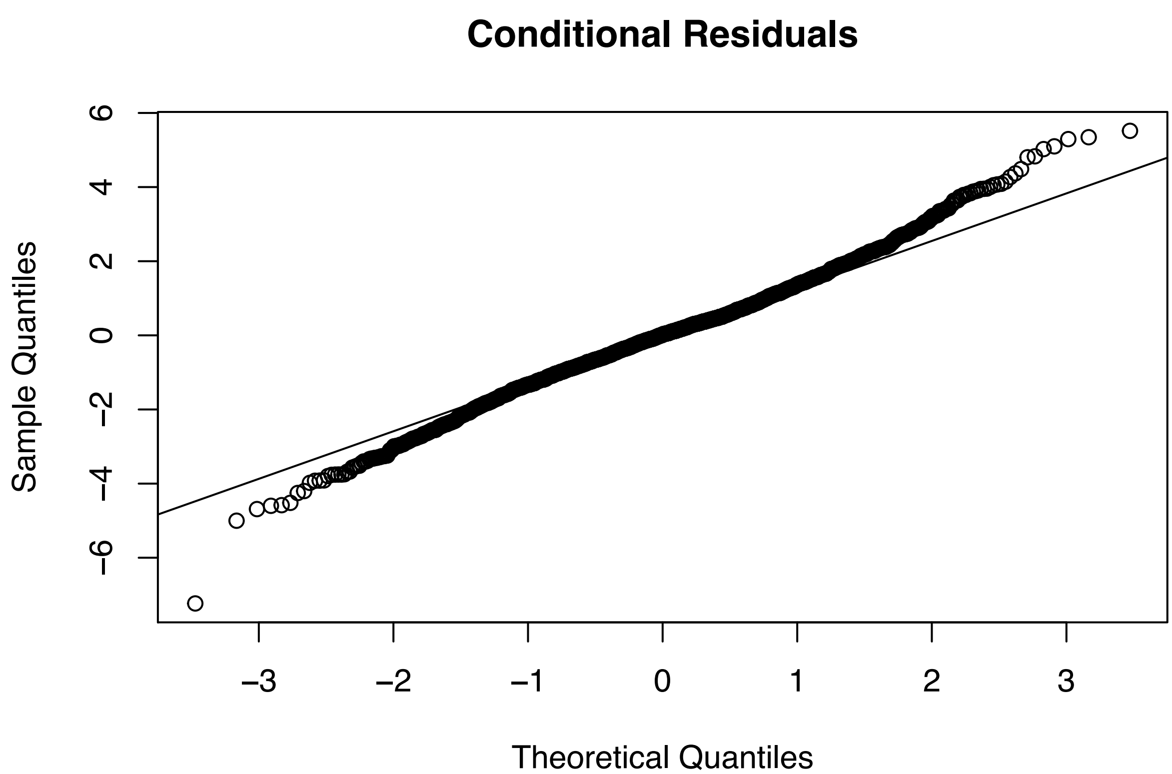

```
qqnorm(resid.marg, main = "Marginal Residuals")
qqline(resid.marg)
```



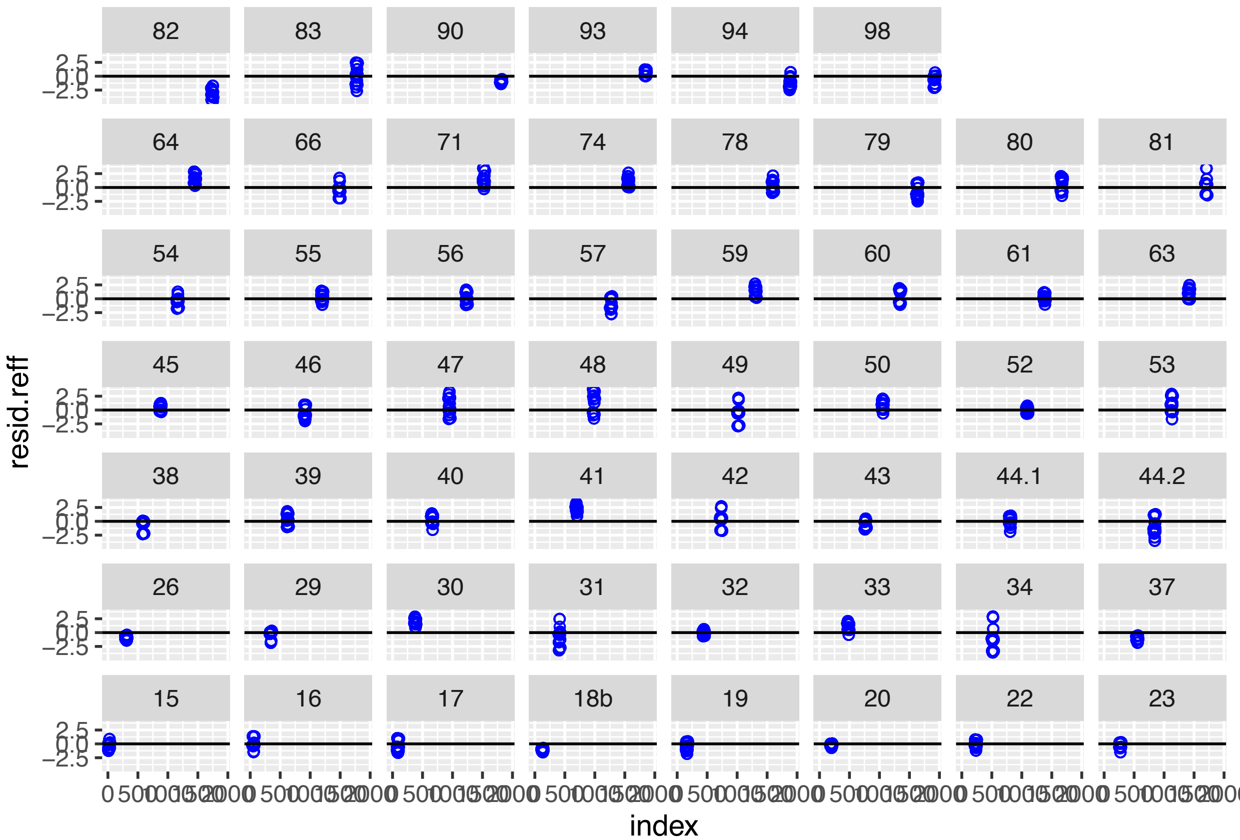

```
new.data <- data.frame(index, resid.cond, Subject)
names(new.data) <- c("index", "resid.cond", "Subject")
ggplot(new.data, aes(x = index, y = resid.cond)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1, color = "Blue") +
    geom_hline(yintercept = 0)
```



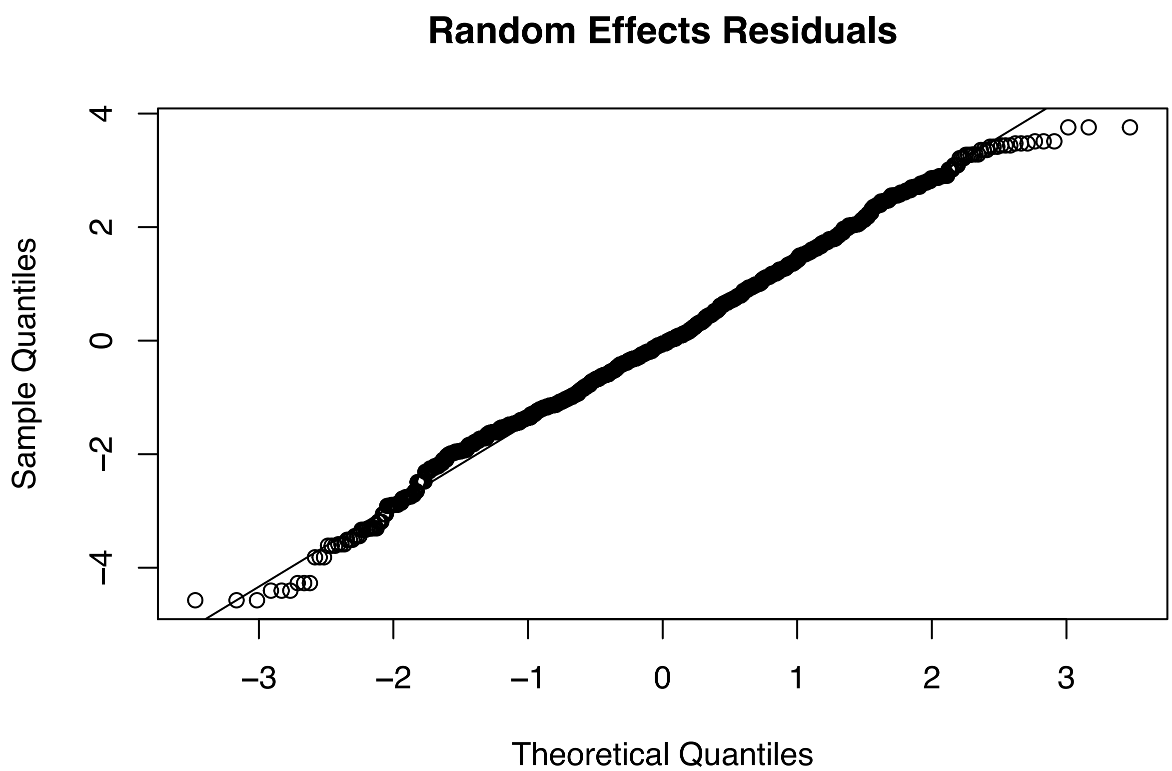

```
qqnorm(resid.cond, main = "Conditional Residuals")
qqline(resid.cond)
```



```
new.data <- data.frame(index, resid.reff, Subject)
names(new.data) <- c("index", "resid.reff", "Subject")
ggplot(new.data, aes(x = index, y = resid.reff)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1, color = "Blue") +
    geom_hline(yintercept = 0)
```




```
qqnorm(resid.reff, main = "Random Effects Residuals")
qqline(resid.reff)
```




```
rm(new.data, resid.marg, resid.reff, resid.cond, index)
```

```
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 3)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Classical ~ Musician * (Harmony + Instrument + Voice +
                                      Harmony:Voice + Log.OMSI +
                                      X16.minus.17 + ClsListen +
                                      KnowAxis + KnowRob), data))
```

```
##                        Df Sum Sq Mean Sq F value   Pr(>F)
## Musician                1      6     6.3   1.409 0.235359
## Harmony                 3    264    88.1  19.552 1.75e-12 ***
## Instrument              2   3871  1935.6 429.611  < 2e-16 ***
## Voice                   2     53    26.7   5.933 0.002701 **
## Log.OMSI                1    147   147.5  32.737 1.22e-08 ***
## X16.minus.17            1    180   180.2  39.991 3.17e-10 ***
## ClsListen               1    204   203.8  45.224 2.32e-11 ***
## KnowAxis                1     59    58.9  13.076 0.000307 ***
## KnowRob                 1      7     6.6   1.475 0.224651
## Harmony:Voice           6     71    11.8   2.626 0.015399 *
## Musician:Harmony        3     91    30.5   6.765 0.000155 ***
## Musician:Instrument     2     55    27.4   6.077 0.002340 **
## Musician:Voice          2      2     1.0   0.223 0.800463
## Musician:Log.OMSI       1     91    91.5  20.303 7.01e-06 ***
## Musician:X16.minus.17   1    145   144.8  32.139 1.66e-08 ***
## Musician:ClsListen      1      1     0.9   0.206 0.650247
## Musician:KnowAxis       1      4     4.0   0.891 0.345197
## Musician:KnowRob        1      1     1.2   0.275 0.600257
## Musician:Harmony:Voice  6     32     5.4   1.192 0.307630
```

```
## Residuals              1899   8556     4.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 4)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Classical ~ Musician * (Harmony + Instrument + Voice +
                                    Harmony:Voice + Log.OMSI +
                                    X16.minus.17 + ClsListen +
                                    KnowAxis + KnowRob), data))
```

```
##                          Df Sum Sq Mean Sq F value   Pr(>F)
## Musician                  1     30    30.4   6.789 0.009245 **
## Harmony                   3    264    88.1  19.676 1.46e-12 ***
## Instrument                2   3871  1935.5 432.146  < 2e-16 ***
## Voice                     2     53    26.7   5.967 0.002611 **
## Log.OMSI                  1    110   109.9  24.535 7.95e-07 ***
## X16.minus.17              1    199   199.0  44.431 3.44e-11 ***
## ClsListen                 1    243   242.6  54.158 2.74e-13 ***
## KnowAxis                  1     31    31.3   6.987 0.008278 **
## KnowRob                   1      0     0.4   0.088 0.767297
## Harmony:Voice             6     71    11.8   2.643 0.014802 *
## Musician:Harmony          3    101    33.7   7.535 5.19e-05 ***
## Musician:Instrument       2     76    38.2   8.524 0.000206 ***
## Musician:Voice            2      1     0.3   0.078 0.924900
## Musician:Log.OMSI         1     14    13.7   3.055 0.080666 .
## Musician:X16.minus.17     1    224   223.9  49.997 2.16e-12 ***
## Musician:ClsListen        1      5     4.7   1.051 0.305454
## Musician:KnowAxis         1      0     0.3   0.056 0.812282
## Musician:KnowRob          1     25    25.2   5.616 0.017898 *
## Musician:Harmony:Voice    6     17     2.9   0.638 0.699600
## Residuals              1899   8505     4.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 5)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Classical ~ Musician * (Harmony + Instrument + Voice +
                                    Harmony:Voice + Log.OMSI +
                                    X16.minus.17 + ClsListen +
                                    KnowAxis + KnowRob), data))
```

```
##                          Df Sum Sq Mean Sq F value   Pr(>F)
## Musician                  1     20    19.8   4.270  0.03891 *
## Harmony                   3    264    88.1  18.956 4.10e-12 ***
## Instrument                2   3871  1935.5 416.413  < 2e-16 ***
## Voice                     2     53    26.7   5.754  0.00323 **
## Log.OMSI                  1    181   181.0  38.949 5.35e-10 ***
## X16.minus.17              1    169   168.9  36.340 1.99e-09 ***
## ClsListen                 1    223   223.3  48.047 5.68e-12 ***
## KnowAxis                  1     33    32.8   7.050  0.00799 **
## KnowRob                   1     15    15.3   3.302  0.06936 .
## Harmony:Voice             6     71    11.9   2.551  0.01830 *
```

```
## Musician:Harmony          3     50    16.7   3.596  0.01309 *
## Musician:Instrument       2      9     4.3   0.935  0.39282
## Musician:Voice            2      6     3.2   0.687  0.50310
## Musician:Log.OMSI         1      6     6.4   1.388  0.23897
## Musician:X16.minus.17     1      0     0.4   0.092  0.76139
## Musician:Harmony:Voice    6     28     4.7   1.007  0.41859
## Residuals              1902   8841     4.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.1p <- aov(Popular ~ Instrument * Harmony * Voice)
summary(model.1p)
```

```
##                               Df Sum Sq Mean Sq F value Pr(>F)
## Instrument                     2   2880  1440.2 292.100 <2e-16 ***
## Harmony                        3     42    14.1   2.860 0.0357 *
## Voice                          2     17     8.5   1.715 0.1803
## Instrument:Harmony             6     17     2.9   0.582 0.7451
## Instrument:Voice               4     20     4.9   0.991 0.4112
## Harmony:Voice                  6     39     6.5   1.327 0.2414
## Instrument:Harmony:Voice      12     59     4.9   0.997 0.4490
## Residuals                   1901   9373     4.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model.2p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1|Subject), REML = F, control = lmerControl(optimizer = 'bobyqa'))
exactRLRT(model.2p)
```

```
## Using restricted likelihood evaluated at ML estimators.

## Refit with method="REML" for exact results.

##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 507.64, p-value < 2.2e-16
```

```
model.3p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Instrument|Subject), REML = F, control = lmerControl(optimizer = 'bobyqa'))
model.4p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Harmony|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa'))
```

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

```
model.5p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa'))
```

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

```
model.6p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa
```

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

```
model.7p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Harmony|Subject), REML = F,  control = lmerControl(optimizer = 'boby
```

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

```
model.8p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Harmony + Voice|Subject), REML = F,  control = lmerControl(optimizer = 'bobyqa')
```

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

```
model.9p <- lmer(Popular ~ 1 + Instrument + Harmony +
                  Voice + KnowAxis + KnowRob +
                  (1 + Instrument + Harmony + Voice|Subject), REML = F,  control = lmerControl(optimize
```

```
## 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
```

```
anova(model.2p, model.3p, model.4p, model.5p, model.6p, model.7p, model.8p, model.9p)
```

```
## Data: NULL
## Models:
## model.2p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.2p:     (1 | Subject)
## model.3p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.3p:     (1 + Instrument | Subject)
## model.5p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.5p:     (1 + Voice | Subject)
## model.4p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.4p:     (1 + Harmony | Subject)
## model.6p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.6p:     (1 + Instrument + Voice | Subject)
## model.7p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.7p:     (1 + Instrument + Harmony | Subject)
## model.8p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.8p:     (1 + Harmony + Voice | Subject)
## model.9p: Popular ~ 1 + Instrument + Harmony + Voice + KnowAxis + KnowRob +
## model.9p:     (1 + Instrument + Harmony + Voice | Subject)
##          Df    AIC    BIC  logLik deviance   Chisq Chi Df Pr(>Chisq)
## model.2p 12 8075.9 8142.7 -4025.9   8051.9
## model.3p 17 7869.3 7964.0 -3917.7   7835.3 216.544      5  < 2.2e-16 ***
## model.5p 17 8084.2 8178.9 -4025.1   8050.2   0.000      0          1
## model.4p 21 8023.1 8140.0 -3990.5   7981.1  69.121      4   3.48e-14 ***
## model.6p 26 7883.0 8027.8 -3915.5   7831.0 150.046      5  < 2.2e-16 ***
## model.7p 32 7777.5 7955.8 -3856.8   7713.5 117.485      6  < 2.2e-16 ***
```

```
## model.8p 32 8039.6 8217.8 -3987.8   7975.6   0.000      0          1
## model.9p 47 7795.5 8057.2 -3850.7   7701.5 274.133     15  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.10p <- lm(Popular ~ Harmony + Instrument + Voice + Selfdeclare + Log.OMSI +
                 X16.minus.17 + ConsInstr + Instr.minus.Notes +
                 PachListen + ClsListen + KnowRob + X1990s2000s +
                X1990s2000s.minus.1960s1970s + CollegeMusic +
                Composing + PianoPlay + GuitarPlay + KnowAxis)
stepAIC(model.10p, direction = "backward", k = log(2044), trace = F)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument + X16.minus.17 + ConsInstr +
##     Instr.minus.Notes + X1990s2000s + KnowAxis)
##
## Coefficients:
##       (Intercept)    Instrumentpiano   Instrumentstring
##            5.4161            -1.1244            -2.9550
##      X16.minus.17          ConsInstr  Instr.minus.Notes
##            0.1268             0.1537            -0.1972
##       X1990s2000s           KnowAxis
##            0.1556             0.1267
```

```r
model.12p <- lmer(Popular ~ Instrument + X16.minus.17 + ConsInstr +
    Instr.minus.Notes + X1990s2000s + KnowAxis + Voice + Harmony + KnowRob +
                  (1 + Instrument|Subject) + (0 + Harmony|Subject), REML = F,
              data = data, control = lmerControl(optimizer = 'bobyqa'))
```

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

```r
model.13p <- ffRanefLMER.fnc(model.12p, ran.effects = c("(0 + Voice|Subject)"))
```

```
## evaluating addition of (0+Voice|Subject) to model
```

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

```
##  log-likelihood ratio test p-value = 0.239012
##  not adding (0+Voice|Subject) to model
## log file is /var/folders/w4/kmmyx3455x77cwww_8985p140000gn/T//Rtmp57wfo5/ffRanefLMER_log_Fri_Dec__6_
```

```r
summary(model.13p)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
##     X1990s2000s + KnowAxis + Voice + Harmony + KnowRob + (1 +
##     Instrument | Subject) + (0 + Harmony | Subject)
##    Data: data
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   7774.7   7947.3  -3856.4   7712.7     1906
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9856 -0.5793  0.0286  0.5573  3.3267
```

```
##
## Random effects:
##  Groups    Name            Variance Std.Dev. Corr
##  Subject   (Intercept)     0.4173   0.6460
##            Instrumentpiano 1.4304   1.1960   -0.66
##            Instrumentstring 2.7207  1.6495   -1.00  0.67
##  Subject.1 HarmonyI-IV-V   1.3746   1.1724
##            HarmonyI-V-IV   1.8605   1.3640   0.98
##            HarmonyI-V-VI   1.6939   1.3015   0.66  0.61
##            HarmonyIV-I-V   0.9234   0.9609   0.88  0.79  0.42
##  Residual                  2.4935   1.5791
## Number of obs: 1937, groups:  Subject, 54
##
## Fixed effects:
##                  Estimate Std. Error t value
## (Intercept)       5.53387    0.60631   9.127
## Instrumentpiano  -1.12747    0.18507  -6.092
## Instrumentstring -2.95531    0.24101 -12.262
## X16.minus.17      0.13894    0.05253   2.645
## ConsInstr         0.16749    0.10016   1.672
## Instr.minus.Notes -0.05869   0.09091  -0.646
## X1990s2000s       0.14883    0.10212   1.457
## KnowAxis          0.10428    0.07998   1.304
## Voicepar5th       0.02250    0.08787   0.256
## Voicecontrary    -0.18662    0.08791  -2.123
## HarmonyI-V-IV    -0.03166    0.11085  -0.286
## HarmonyI-V-VI    -0.35584    0.17224  -2.066
## HarmonyIV-I-V    -0.26291    0.12712  -2.068
## KnowRob          -0.01032    0.09958  -0.104
```

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

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

```
model.14p <- lmer(Popular ~ Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
    X1990s2000s + KnowAxis + KnowRob + Voice + Harmony +
                  (1 + Instrument + Harmony|Subject), REML = F,
          data = data, control = lmerControl(optimizer = 'bobyqa'))
```

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

```
summary(model.14p)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula:
## Popular ~ Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
##     X1990s2000s + KnowAxis + KnowRob + Voice + Harmony + (1 +
##     Instrument + Harmony | Subject)
##    Data: data
## Control: lmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
```

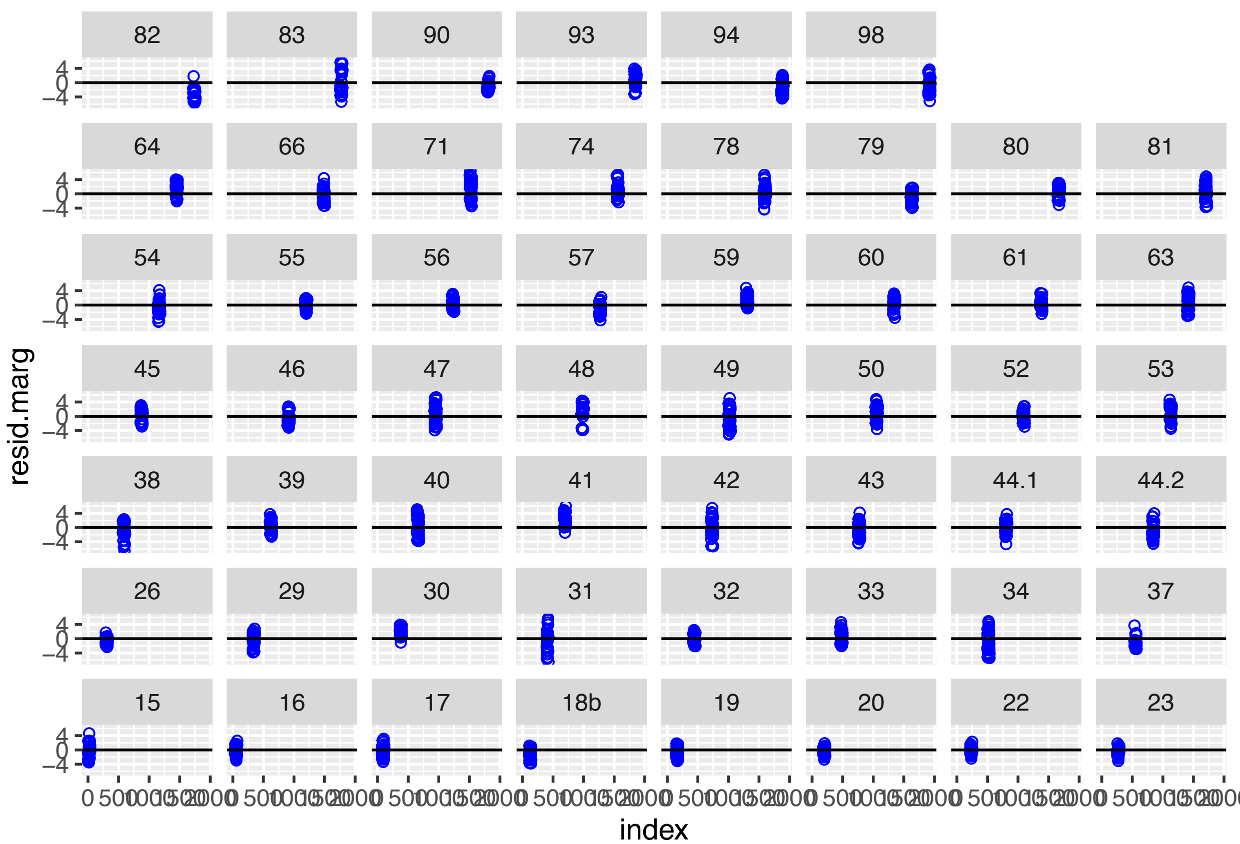
```
##    7778.1   7978.6  -3853.0   7706.1     1901
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9219 -0.5844  0.0102  0.5658  3.3406
##
## Random effects:
##  Groups   Name             Variance Std.Dev. Corr
##  Subject  (Intercept)      1.4410   1.2004
##           Instrumentpiano  1.4351   1.1980   -0.17
##           Instrumentstring 2.7249   1.6507   -0.31  0.67
##           HarmonyI-V-IV    0.1403   0.3746    0.53 -0.27 -0.29
##           HarmonyI-V-VI    1.0733   1.0360   -0.11 -0.28 -0.35 -0.24
##           HarmonyIV-I-V    0.3530   0.5941   -0.36 -0.33 -0.38 -0.52 -0.11
##  Residual                  2.4863   1.5768
## Number of obs: 1937, groups:  Subject, 54
##
## Fixed effects:
##                    Estimate Std. Error t value
## (Intercept)         5.52711    0.60097   9.197
## Instrumentpiano    -1.12733    0.18525  -6.085
## Instrumentstring   -2.95550    0.24113 -12.257
## X16.minus.17        0.13865    0.05254   2.639
## ConsInstr           0.16897    0.10017   1.687
## Instr.minus.Notes -0.05843    0.09092  -0.643
## X1990s2000s         0.14971    0.10213   1.466
## KnowAxis            0.10300    0.07999   1.288
## KnowRob            -0.00975    0.09959  -0.098
## Voicepar5th         0.02261    0.08774   0.258
## Voicecontrary      -0.18643    0.08778  -2.124
## HarmonyI-V-IV      -0.03179    0.11342  -0.280
## HarmonyI-V-VI      -0.35566    0.17366  -2.048
## HarmonyIV-I-V      -0.26308    0.12958  -2.030

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

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

```r
resid.marg <- r.marg(model.14)
resid.cond <- r.cond(model.14)
resid.reff <- r.reff(model.14)

attach(data)
```

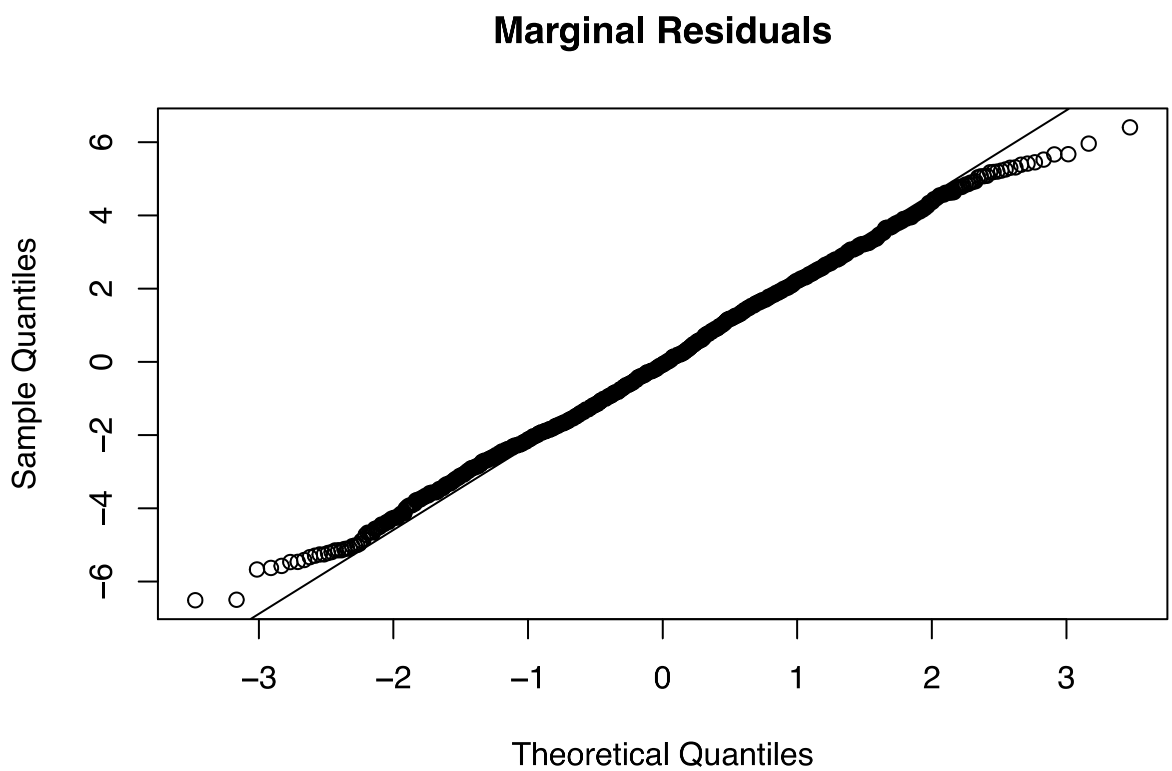
```
## The following objects are masked from data (pos = 3):
##
##     Classical, ClsListen, CollegeMusic, Composing, ConsInstr,
##     GuitarPlay, Harmony, Instr.minus.Notes, Instrument, KnowAxis,
##     KnowRob, Log.OMSI, OMSI, PachListen, PianoPlay, Popular,
##     Selfdeclare, Subject, Voice, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s
```

```
## The following objects are masked from data (pos = 4):
##
##     Classical, ClsListen, CollegeMusic, Composing, ConsInstr,
##     GuitarPlay, Harmony, Instr.minus.Notes, Instrument, KnowAxis,
##     KnowRob, Log.OMSI, OMSI, PachListen, PianoPlay, Popular,
##     Selfdeclare, Subject, Voice, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s
```

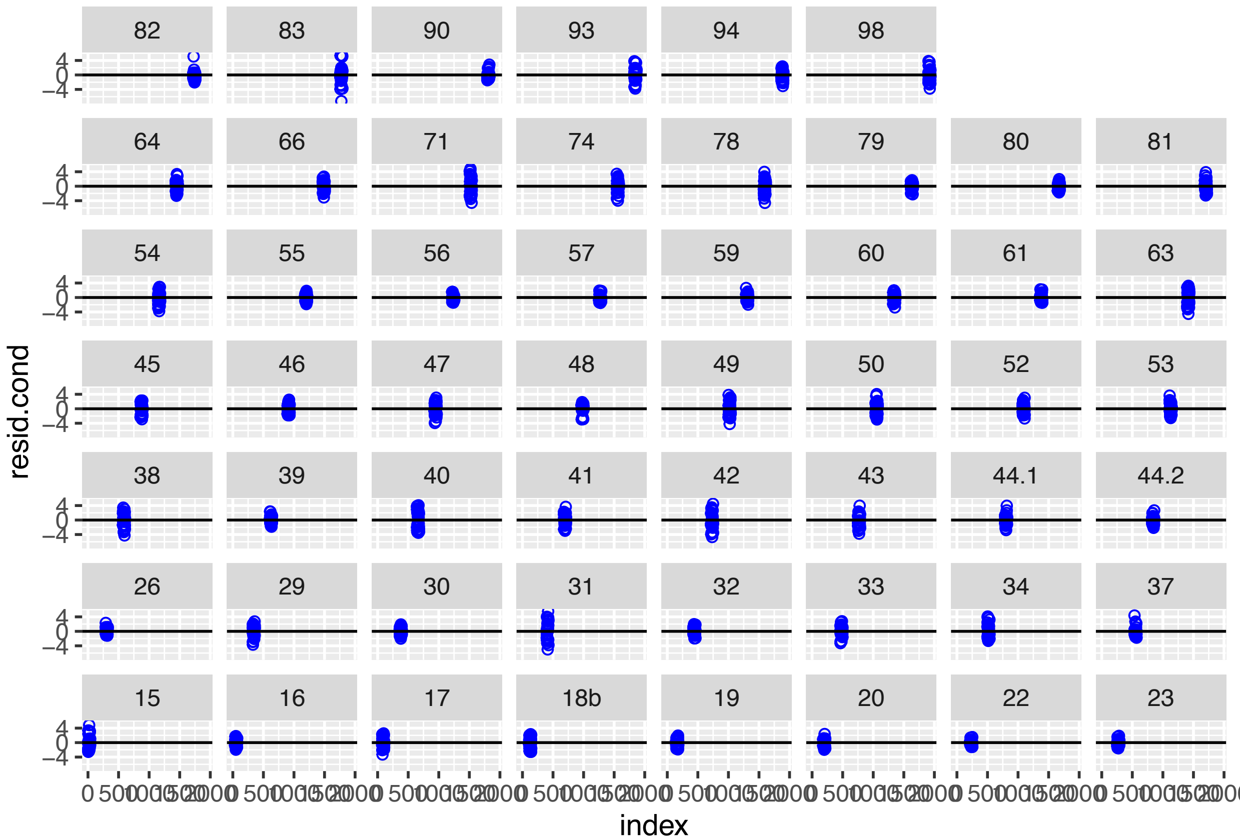
```
index <- 1:dim(data)[1]

new.data <- data.frame(index, resid.marg, Subject)
names(new.data) <- c("index","resid.marg","Subject")
ggplot(new.data,aes(x = index, y = resid.marg)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1,color = "Blue") +
    geom_hline(yintercept = 0)
```



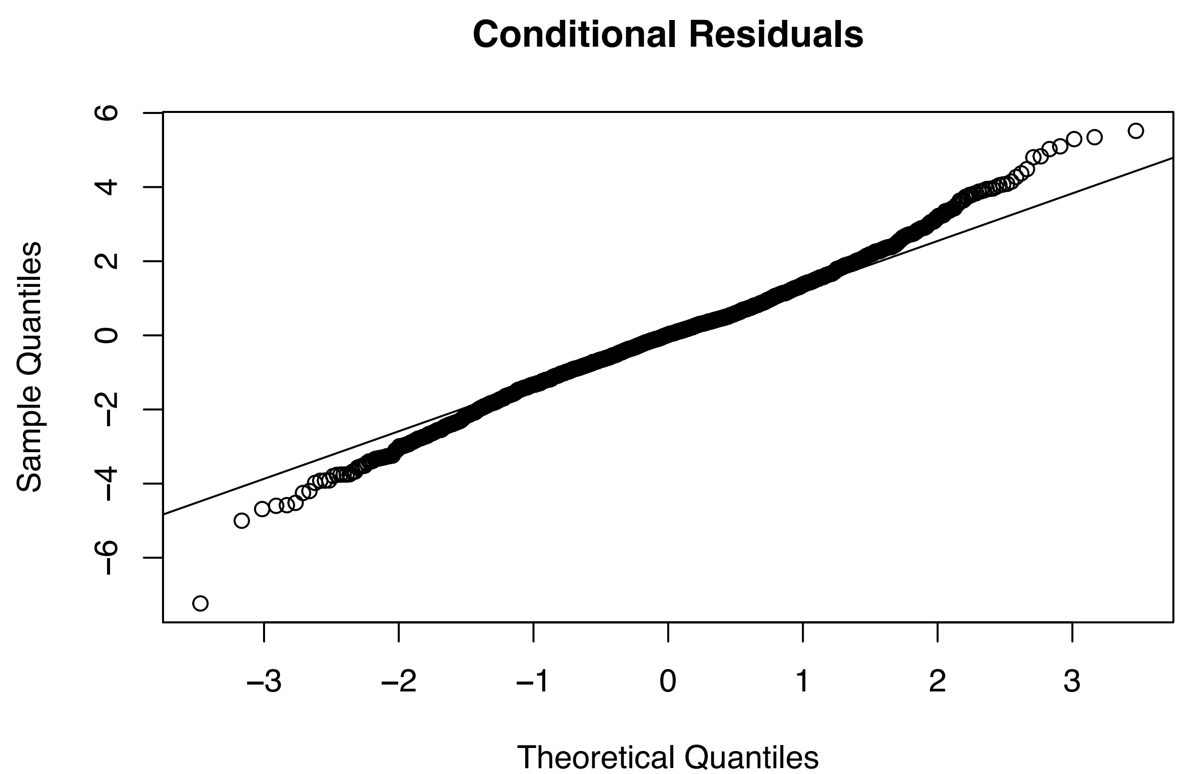

```
qqnorm(resid.marg, main = "Marginal Residuals")
qqline(resid.marg)
```



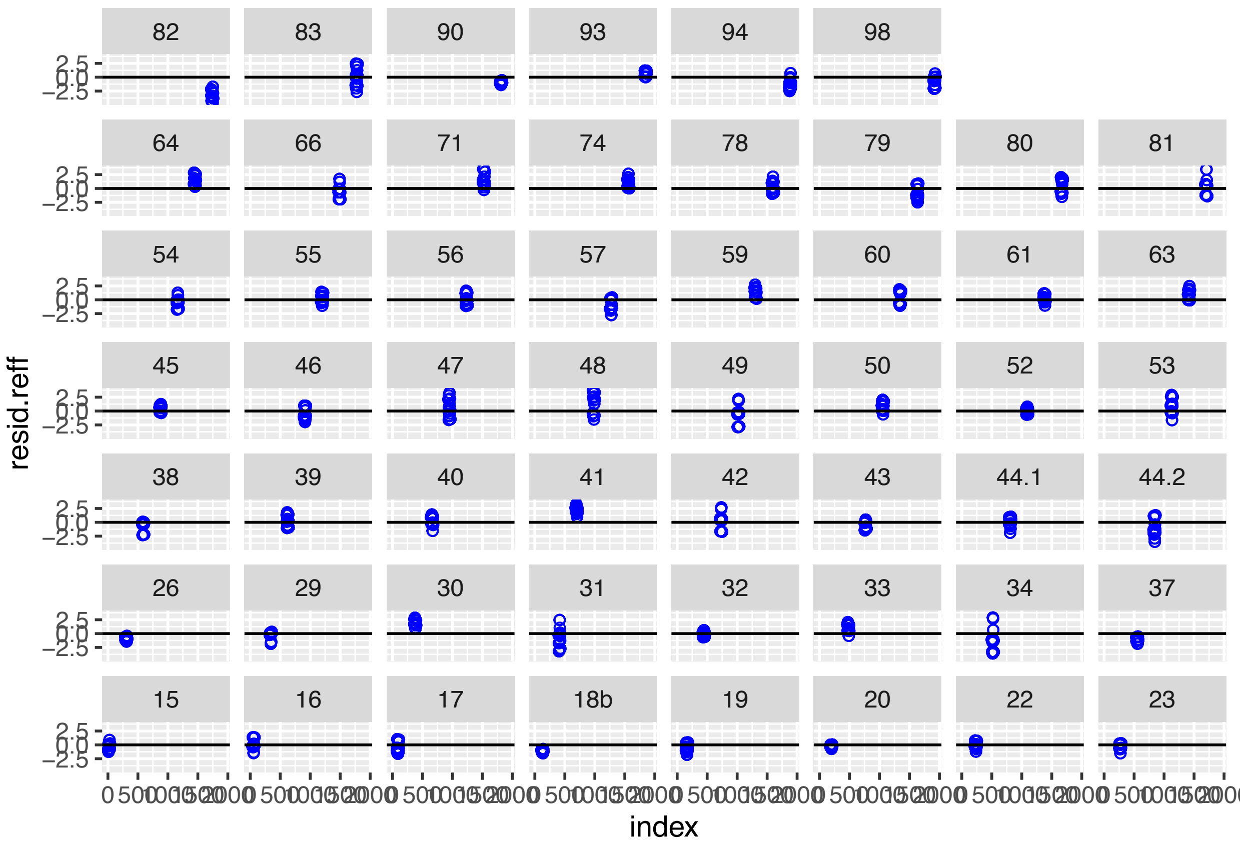

```
new.data <- data.frame(index, resid.cond, Subject)
names(new.data) <- c("index", "resid.cond", "Subject")
ggplot(new.data, aes(x = index, y = resid.cond)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1, color = "Blue") +
    geom_hline(yintercept = 0)
```



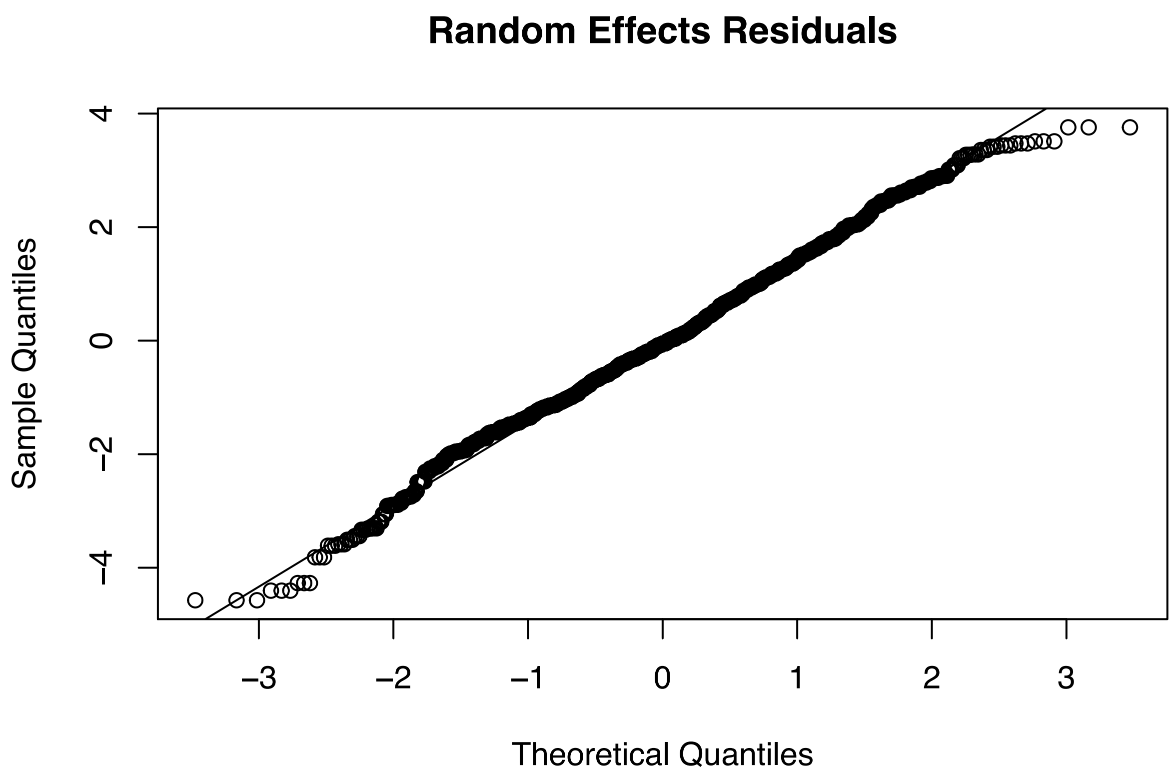

```r
qqnorm(resid.cond, main = "Conditional Residuals")
qqline(resid.cond)
```



```r
new.data <- data.frame(index, resid.reff, Subject)
names(new.data) <- c("index", "resid.reff", "Subject")
ggplot(new.data, aes(x = index, y = resid.reff)) +
  facet_wrap( ~ Subject, as.table = F) +
  geom_point(pch = 1, color = "Blue") +
    geom_hline(yintercept = 0)
```




```r
qqnorm(resid.reff, main = "Random Effects Residuals")
qqline(resid.reff)
```




```
rm(new.data, resid.marg, resid.reff, resid.cond, index)
```

```
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 3)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Popular ~ Musician * (Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
    X1990s2000s + KnowAxis + KnowRob + Voice + Harmony), data))
```

```
##                              Df Sum Sq Mean Sq F value   Pr(>F)
## Musician                      1     50    49.6  10.795  0.00104 **
## Instrument                    2   2881  1440.3 313.717  < 2e-16 ***
## X16.minus.17                  1     45    45.1   9.817  0.00176 **
## ConsInstr                     1     34    34.4   7.483  0.00629 **
## Instr.minus.Notes             1    121   121.2  26.403 3.05e-07 ***
## X1990s2000s                   1     71    70.5  15.360 9.20e-05 ***
## KnowAxis                      1    131   130.8  28.489 1.05e-07 ***
## KnowRob                       1      5     4.6   1.007  0.31569
## Voice                         2     17     8.5   1.843  0.15860
## Harmony                       3     42    14.1   3.081  0.02647 *
## Musician:Instrument           2     62    30.9   6.741  0.00121 **
## Musician:X16.minus.17         1     42    42.0   9.143  0.00253 **
## Musician:ConsInstr            1     21    21.2   4.624  0.03165 *
## Musician:Instr.minus.Notes    1     31    31.3   6.809  0.00914 **
## Musician:X1990s2000s          1     22    21.6   4.702  0.03025 *
## Musician:KnowAxis             1     37    36.6   7.974  0.00480 **
## Musician:KnowRob              1     27    26.9   5.858  0.01560 *
## Musician:Voice                2      7     3.7   0.804  0.44770
## Musician:Harmony              3     38    12.8   2.780  0.03979 *
## Residuals                  1909   8764     4.6
## ---
```

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

```
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 4)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Popular ~ Musician * (Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
    X1990s2000s + KnowAxis + KnowRob + Voice + Harmony), data))
```

```
##                              Df Sum Sq Mean Sq F value   Pr(>F)    
## Musician                      1     27    26.6   5.776 0.016343 *  
## Instrument                    2   2881  1440.4 312.401  < 2e-16 ***
## X16.minus.17                  1     66    65.9  14.297 0.000161 ***
## ConsInstr                     1     39    38.5   8.358 0.003883 ** 
## Instr.minus.Notes             1    120   119.8  25.975 3.80e-07 ***
## X1990s2000s                   1     78    78.3  16.982 3.93e-05 ***
## KnowAxis                      1    120   120.2  26.069 3.62e-07 ***
## KnowRob                       1     11    11.2   2.426 0.119502    
## Voice                         2     17     8.5   1.835 0.159888    
## Harmony                       3     42    14.1   3.067 0.026973 *  
## Musician:Instrument           2     32    16.1   3.498 0.030458 *  
## Musician:X16.minus.17         1     19    19.0   4.126 0.042365 *  
## Musician:ConsInstr            1      7     7.2   1.567 0.210820    
## Musician:Instr.minus.Notes    1      3     2.8   0.599 0.439149    
## Musician:X1990s2000s          1     96    95.8  20.783 5.47e-06 ***
## Musician:KnowAxis             1     13    13.2   2.858 0.091109 .  
## Musician:KnowRob              1      8     7.9   1.721 0.189778    
## Musician:Voice                2      2     0.8   0.177 0.838056    
## Musician:Harmony              3     65    21.7   4.703 0.002822 ** 
## Residuals                  1909   8802     4.6                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
data$Musician <- rep(0, nrow(data))
data$Musician[which(data$Selfdeclare >= 5)] <- 1
data$Musician <- as.factor(data$Musician)
summary(aov(Popular ~ Musician * (Instrument + X16.minus.17 + ConsInstr + Instr.minus.Notes +
    X1990s2000s + KnowAxis + KnowRob + Voice + Harmony), data))
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)    
## Musician                 1     22    22.4   4.787 0.028800 *  
## Instrument               2   2880  1440.1 307.892  < 2e-16 ***
## X16.minus.17             1     62    62.4  13.342 0.000266 ***
## ConsInstr                1     39    39.1   8.359 0.003881 ** 
## Instr.minus.Notes        1    127   126.7  27.093 2.15e-07 ***
## X1990s2000s              1     77    76.9  16.435 5.24e-05 ***
## KnowAxis                 1    114   114.4  24.467 8.22e-07 ***
## KnowRob                  1      2     2.5   0.526 0.468311    
## Voice                    2     17     8.5   1.810 0.163963    
## Harmony                  3     42    14.1   3.024 0.028594 *  
## Musician:Instrument      2     10     5.1   1.093 0.335526    
## Musician:X16.minus.17    1     64    63.7  13.620 0.000230 ***
## Musician:ConsInstr       1      1     0.9   0.184 0.667728    
## Musician:Voice           2      2     0.8   0.168 0.845665    
## Musician:Harmony         3     39    13.1   2.801 0.038666 *  
## Residuals             1913   8948     4.7                     
```

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