Josh Jelin

Hierarchical Linear Models
Homework 5

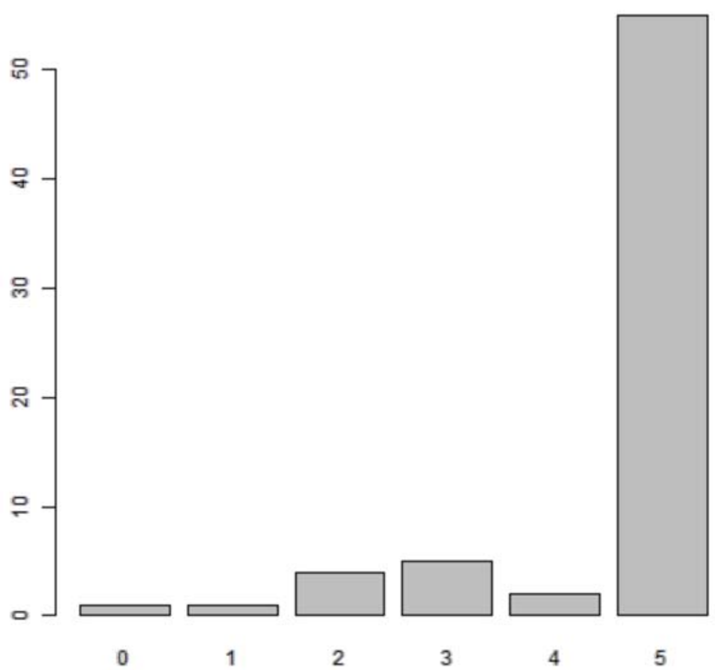
EDA.

While none of the questions specifically ask for EDA, I thought it would be important, so I've included it here. You're welcome to skip over it. Before you do, realize that I found a significant amount of missing data in this data set. Obvious corrections were made, but subjects with observations that we not easy to fix were excluded from the analysis. If this were a professional data analysis project, I'd make sure to investigate how these exclusions impacted our results. However, that seemed out-of-scope for this homework assignment.

Additionally, because of all the missing data, I avoided adding variables to the model during model selection unless explicitly told to do so, in order to avoid running into more missing values.

```
###### Basic EDA There are a bunch of NAs around here

# table(Subject)

par(mfrow = c(2, 2))
plot(Harmony)
plot(Instrument)
plot(Voice)
par(mfrow = c(1, 1))
title("Check to ensure the design is balanced")
```

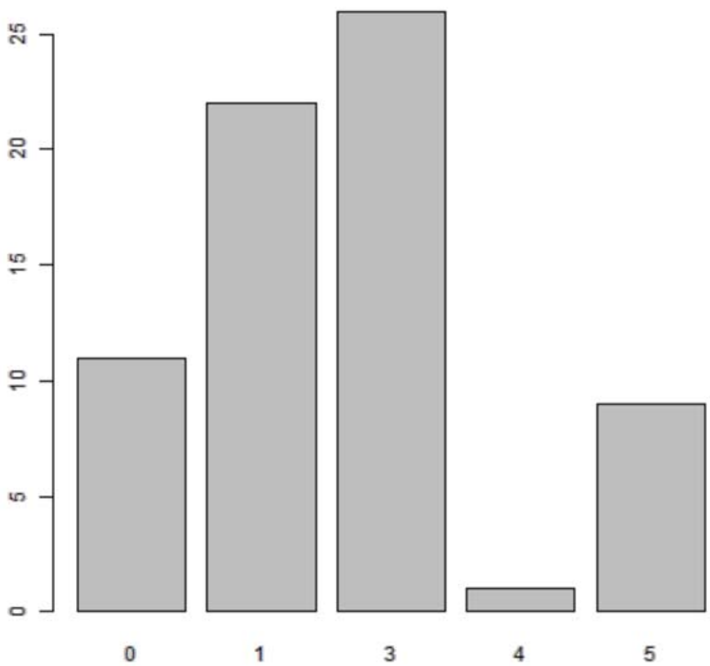
```
## all of our data is evenly distributed, so so a repeated measures model is
## valid

# par(mfrow = c(2,2)) hist(Selfdeclare) hist(OMSI) hist(X16.minus.17)

# par(mfrow = c(2,2)) hist(ConsInstr) hist(ConsNotes)
# hist(Instr.minus.Notes)

barplot(table(PachListen)/36)
```




```
table(table(PachListen)/36)
```

```
##
##  1  2  4  5 55
##  2  1  1  1  1
```

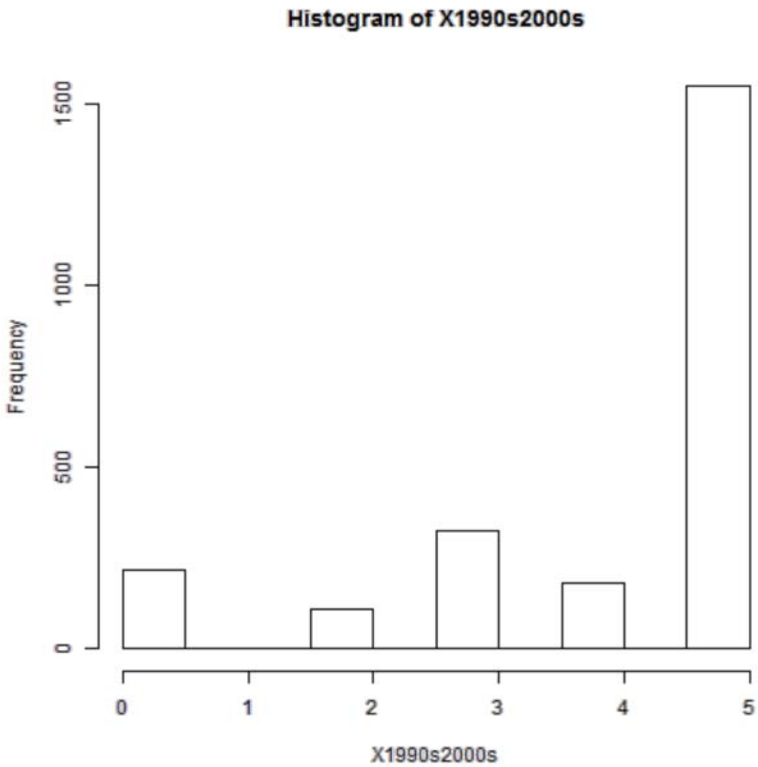
```
## Nearly everyone (55/67) answered '5' to this one. One person answered '0.'
## I'm not sure how to interpret any answers between these two values. Maybe
## they've seen part of it?  Considering the issues with this variable, I'm
## inclined to exclude it.

barplot(table(ClsListen)/36)
```



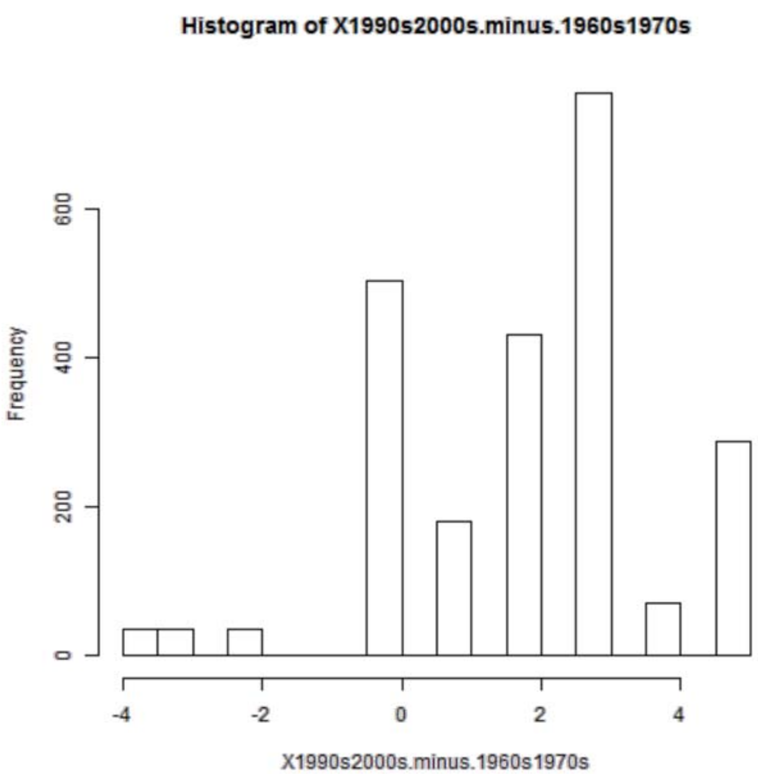

```
table(table(ClsListen)/36)
```

```
##
##  1  9 11 22 26
##  1  1  1  1  1
```

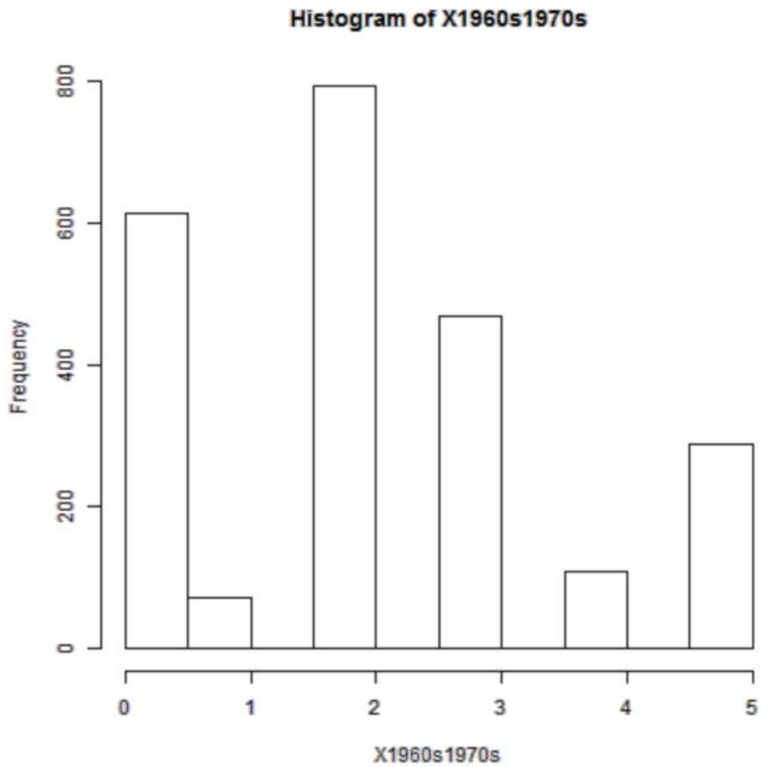
```
## Similar to last time, I don't know what levels 1, 3, and 4 mean.  The
## question is actually worded incorrectly. The comedy bit in question was
## performed by Axis of AWESOME.  I'm inclined to exclude this one as well.

hist(X1990s2000s)  ## a lot of 5s
```




```
hist(X1990s2000s.minus.1960s1970s)
```




```
X1960s1970s = X1990s2000s - X1990s2000s.minus.1960s1970s
hist(X1960s1970s)  ## more even spread
```




```
summary(as.factor(CollegeMusic))  ## 108 NAs!
```

```
##    0    1 NA's
##  504 1908  108
```

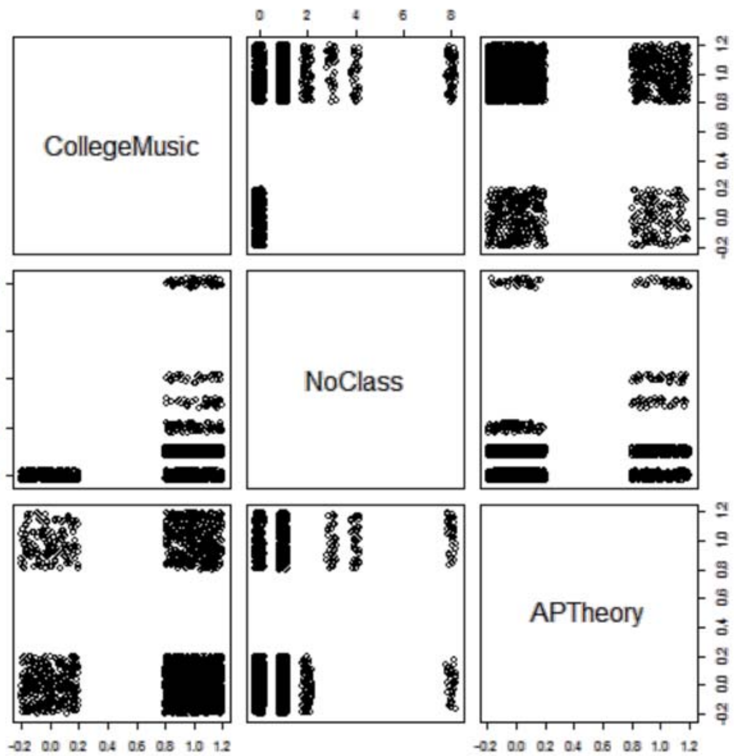
```
table(NoClass)  ## more people took classes in college, than have ever taken
classes
```

```
## NoClass
##    0    1    2    3    4    8
##  936 1080   72   36   36   72
```

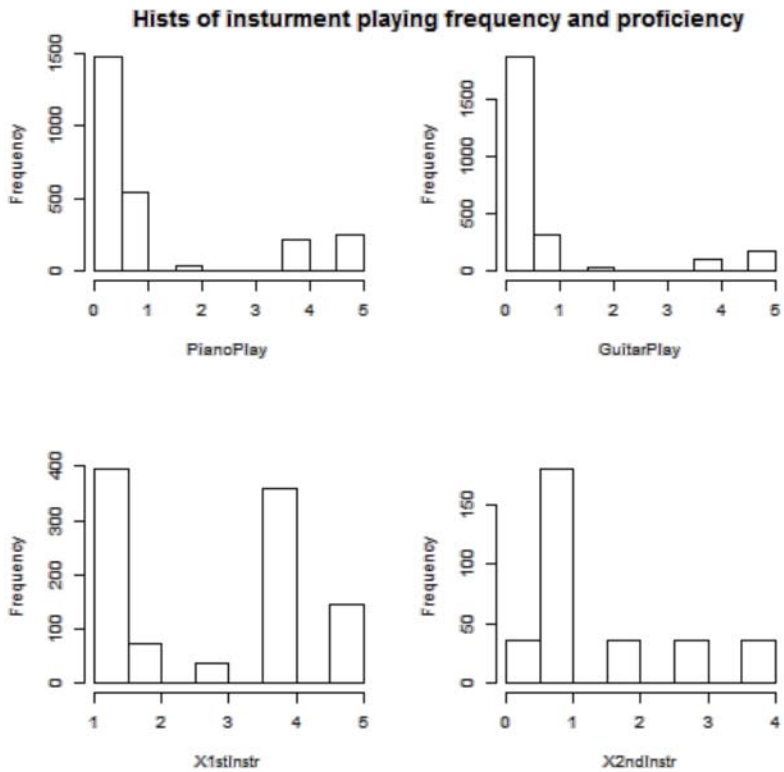
```
summary(as.factor(APTheory))  ## 216 NAs!
```

```
##    0    1 NA's
## 1764  540  216
```

```
pairs(jitter(cbind(CollegeMusic, NoClass, APTheory)))
```



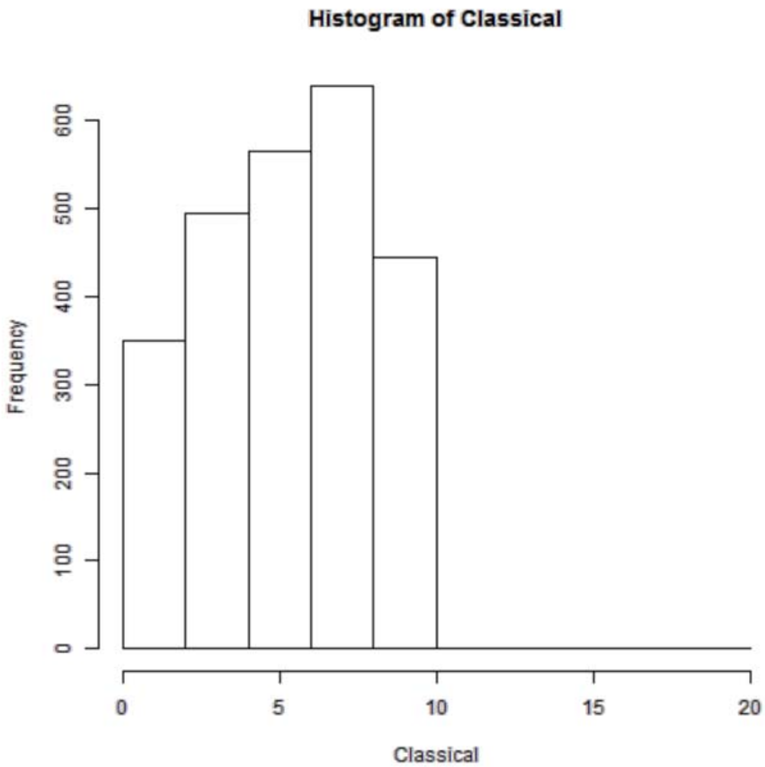

```
## Yeah there are clearly people who have taken 0 classes, who took AP Theory
## or courses in college!

par(mfrow = c(2, 2))
hist(PianoPlay, main = "")
hist(GuitarPlay, main = "")
hist(X1stInstr, main = "")
hist(X2ndInstr, main = "")
par(mfrow = c(1, 1))
title("Hists of insturment playing frequency and proficiency")
```



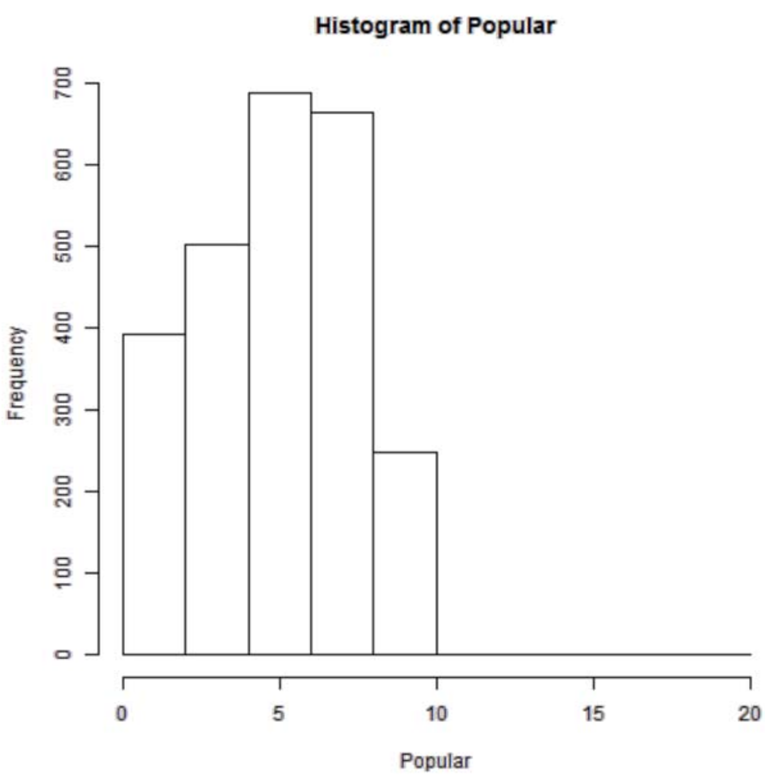

```
table(X1stInstr)  ## no 0s. Everyone plays an insturment.
```

```
## X1stInstr
##   1   2   3   4   5 
## 396  72  36 360 144
```

```
hist(Classical)
```




```
hist(Popular)
```




```
summary(Classical)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    4.00    6.00    5.78    8.00   19.00      27
```

```
summary(Popular)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    4.00    5.00    5.38    7.00   19.00      27
```

```
## some music sounds 19! That doesn't make sense.  27 NAs. We might need to
## exclude those observations

###### Error correction
apply(data, 2, function(x) sum(is.na(x)))
```

```
##                             X                       Subject 
##                             0                             0 
##                       Harmony                    Instrument 
##                             0                             0 
##                         Voice                    Selfdeclare 
##                             0                             0 
##                          OMSI                  X16.minus.17 
##                             0                             0 
##                     ConsInstr                     ConsNotes 
##                             0                           360 
##             Instr.minus.Notes                    PachListen 
##                             0                            72 
##                     ClsListen                       KnowRob 
##                            36                           180 
##                     KnowAxis                    X1990s2000s 
##                           288                           144 
## X1990s2000s.minus.1960s1970s                   CollegeMusic 
##                           180                           108 
##                       NoClass                      APTheory 
##                           288                           216 
##                     Composing                     PianoPlay 
##                            72                             0 
##                    GuitarPlay                     X1stInstr 
##                             0                          1512 
##                     X2ndInstr                       first12 
##                          2196                             0 
##                     Classical                       Popular 
##                            27                            27
```

```
## Classical and Popular have 27 NAs each, which we'll have to deal with The
## rest of the NAs are all 'by subject.'

str(data)
```

```
## 'data.frame':    2520 obs. of  28 variables:
##  $ X                           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Subject                     : Factor w/ 70 levels "15","16","17",..: 1 1
1 1 1 1 1 1 1 1 ...
##  $ Harmony                     : Factor w/ 4 levels "I-IV-V","I-V-IV",..: 1
1 1 1 1 1 1 1 1 2 ...
##  $ Instrument                  : Factor w/ 3 levels "guitar","piano",..: 1
1 1 2 2 2 3 3 3 1 ...
##  $ Voice                       : Factor w/ 3 levels "contrary","par3rd",..:
1 2 3 1 2 3 1 2 3 1 ...
##  $ Selfdeclare                 : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ OMSI                        : int  734 734 734 734 734 734 734 734 734
734 ...
##  $ X16.minus.17                : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ ConsInstr                   : num  4.33 4.33 4.33 4.33 4.33 4.33 4.33
4.33 4.33 4.33 ...
##  $ ConsNotes                   : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Instr.minus.Notes           : num  -0.67 -0.67 -0.67 -0.67 -0.67 -0.67
-0.67 -0.67 -0.67 -0.67 ...
##  $ PachListen                  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ ClsListen                   : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ KnowRob                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ KnowAxis                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1990s2000s                 : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1990s2000s.minus.1960s1970s: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ CollegeMusic                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NoClass                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APTheory                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Composing                   : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ PianoPlay                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ GuitarPlay                  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1stInstr                   : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ X2ndInstr                   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ first12                     : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 3 3 ...
##  $ Classical                   : num  3 3 1 3 2 8 10 6 5 1 ...
##  $ Popular                     : num  9 7 8 7 8 3 1 4 5 8 ...
```

```
str(na.omit(data))
```

```
## 'data.frame':    180 obs. of  28 variables:
##  $ X                           : int  469 470 471 472 473 474 475 476 477
478 ...
##  $ Subject                     : Factor w/ 70 levels "15","16","17",..: 14
14 14 14 14 14 14 14 14 14 ...
##  $ Harmony                     : Factor w/ 4 levels "I-IV-V","I-V-IV",..: 1
1 1 1 1 1 1 1 1 2 ...
##  $ Instrument                  : Factor w/ 3 levels "guitar","piano",..: 1
1 1 2 2 2 3 3 3 1 ...
##  $ Voice                       : Factor w/ 3 levels "contrary","par3rd",..:
1 2 3 1 2 3 1 2 3 1 ...
##  $ Selfdeclare                 : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ OMSI                        : int  88 88 88 88 88 88 88 88 88 88 ...
##  $ X16.minus.17                : num  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##  $ ConsInstr                   : num  4.33 4.33 4.33 4.33 4.33 4.33 4.33
4.33 4.33 4.33 ...
##  $ ConsNotes                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Instr.minus.Notes           : num  4.33 4.33 4.33 4.33 4.33 4.33 4.33
4.33 4.33 4.33 ...
##  $ PachListen                  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ ClsListen                   : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ KnowRob                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ KnowAxis                    : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1990s2000s                 : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1990s2000s.minus.1960s1970s: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ CollegeMusic                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NoClass                     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ APTheory                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Composing                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PianoPlay                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GuitarPlay                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stInstr                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ X2ndInstr                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ first12                     : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 3 3 ...
##  $ Classical                   : num  4 5 6 3 2 6 8 5 10 6 ...
##  $ Popular                     : num  7 7 7 9 4 8 4 6 1 6 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:2340] 1 2 3 4 5 6 7 8 9
10 ...
##   .. ..- attr(*, "names")= chr [1:2340] "1" "2" "3" "4" ...
```

```
## excluding all missing data reduces our data set from 2520 observations to
## 180 observations, so that's not a good option.

sum(na.omit(Classical) > 10)
```

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

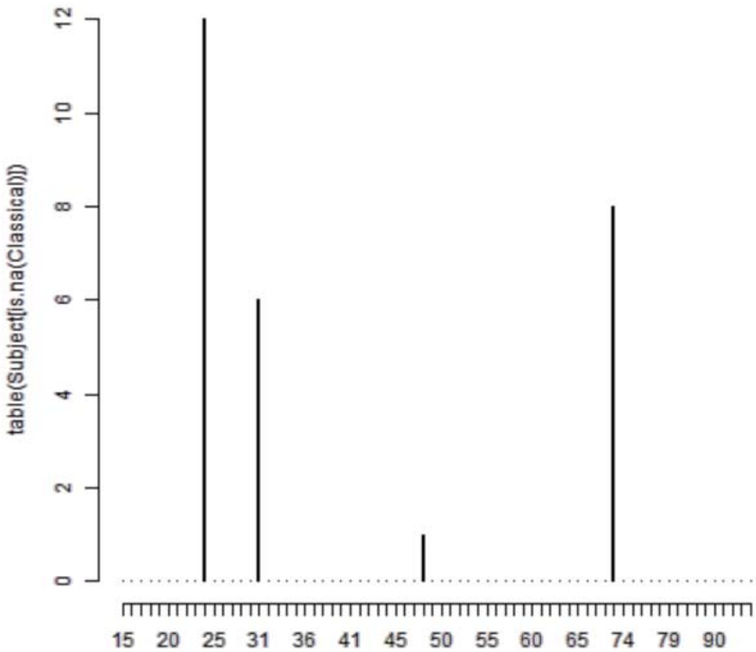
```
sum(na.omit(Popular) > 10)
```

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

```
## This only happens once for each variable.

Popular[is.na(Classical)]
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [24] NA NA NA NA
```

```
## 100% overlap
plot(table(Subject[is.na(Classical)]))  ## four subjects have missing entries
```




```
Subject[na.omit(Classical) > 10]  ## 73, our man with a bad answer, also has
missing entries
```

```
## [1] 73
## 70 Levels: 15 16 17 18b 19 20 21 22 23 24 25 26 28 29 30 31 32 33 ... 98
```

```

subjects = unique(data[, c(2, 6:26)])
str(na.omit(subjects))
```

```
## 'data.frame':    5 obs. of  22 variables:
##  $ Subject                    : Factor w/ 70 levels "15","16","17",..: 14
27 52 63 69
##  $ Selfdeclare                : int  2 2 4 3 4
##  $ OMSI                       : int  88 164 421 30 323
##  $ X16.minus.17               : num  -1 -2 2 7 2
##  $ ConsInstr                  : num  4.33 4.33 1.67 3.67 4.33
##  $ ConsNotes                  : int  0 5 1 5 5
##  $ Instr.minus.Notes          : num  4.33 -0.67 0.67 -1.33 -0.67
##  $ PachListen                 : int  5 5 5 5 5
##  $ ClsListen                  : int  3 0 5 1 5
##  $ KnowRob                    : int  0 5 5 0 5
##  $ KnowAxis                   : int  5 5 5 0 0
##  $ X1990s2000s                : int  5 5 3 2 0
##  $ X1990s2000s.minus.1960s1970s: int  2 5 3 0 -3
##  $ CollegeMusic               : int  1 0 1 1 0
##  $ NoClass                    : int  1 0 1 2 0
##  $ APTheory                   : int  0 0 1 0 0
##  $ Composing                  : int  0 1 4 0 0
##  $ PianoPlay                  : int  0 0 1 1 1
##  $ GuitarPlay                 : int  0 0 5 0 0
##  $ X1stInstr                  : int  1 4 5 4 5
##  $ X2ndInstr                  : int  1 0 4 2 3
##  $ first12                    : Factor w/ 3 levels "guitar","piano",..: 3
3 1 2 1
##  - attr(*, "na.action")=Class 'omit'  Named int [1:65] 1 2 3 4 5 6 7 8 9 10
...
##   .. ..- attr(*, "names")= chr [1:65] "1" "37" "73" "109" ...
```

```
## The problem appears to be that most of the subjects (65/70) are missing
## some information.  The big problem appears to be in X1stInstr and
## X2ndInstr, which are missing for obvious reasons.

## Assume anyone who does not play guitar or piano, and answered NA for both
## Instr questions, has 0 skill
doesNotPlay = (PianoPlay == 0) & (GuitarPlay == 0) & (is.na(X1stInstr)) &
(is.na(X2ndInstr))
# mean(doesNotPlay)

data2 = data

data2$X1stInstr[doesNotPlay] = 0
data2$X2ndInstr[doesNotPlay] = 0

attach(data2)
```

```
## The following objects are masked from data:
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
```

```
subjects = unique(data2[, c(2, 6:26)])
str(na.omit(subjects))  ## Now our study is up to 16 subjects
```

```
## 'data.frame':    16 obs. of  22 variables:
##  $ Subject                     : Factor w/ 70 levels "15","16","17",..: 3 9
12 14 22 25 27 29 43 49 ...
##  $ Selfdeclare                 : int  2 3 1 2 2 1 2 2 2 2 ...
##  $ OMSI                        : int  67 319 194 88 11 29 164 142 150 23
...
##  $ X16.minus.17                : num  3 0 0 -1 -1 9 -2 0 0 0 ...
##  $ ConsInstr                   : num  2.33 5 3 4.33 3 1 4.33 0 3 2.67 ...
##  $ ConsNotes                   : int  0 5 0 0 3 0 5 0 4 0 ...
##  $ Instr.minus.Notes           : num  2.33 0 3 4.33 0 1 -0.67 0 -1 2.67 ...
##  $ PachListen                  : int  5 5 5 5 3 5 5 5 5 5 ...
##  $ ClsListen                   : int  1 1 1 3 1 1 0 3 0 3 ...
##  $ KnowRob                     : int  0 0 0 0 0 0 5 0 0 0 ...
##  $ KnowAxis                    : int  0 0 5 5 0 0 5 0 0 0 ...
##  $ X1990s2000s                 : int  5 5 5 5 2 5 5 4 5 5 ...
##  $ X1990s2000s.minus.1960s1970s: int  5 3 3 2 0 3 5 0 3 3 ...
##  $ CollegeMusic                : int  1 1 1 1 1 0 0 1 1 1 ...
##  $ NoClass                     : int  0 1 1 1 0 0 0 1 1 1 ...
##  $ APTheory                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Composing                   : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ PianoPlay                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GuitarPlay                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stInstr                   : num  0 0 0 1 0 0 4 0 0 0 ...
##  $ X2ndInstr                   : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ first12                     : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 1 1 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:54] 1 2 4 5 6 7 8 10 11
13 ...
##   .. ..- attr(*, "names")= chr [1:54] "1" "37" "109" "145" ...
```

```
# apply(data, 2, function(x) sum(is.na(x)))
apply(data2, 2, function(x) sum(is.na(x)))
```

```
##                            X                      Subject
##                            0                            0
##                      Harmony                   Instrument
##                            0                            0
##                        Voice                  Selfdeclare
##                            0                            0
##                         OMSI                 X16.minus.17
##                            0                            0
##                    ConsInstr                    ConsNotes
##                            0                          360
##            Instr.minus.Notes                   PachListen
##                            0                           72
##                    ClsListen                      KnowRob
##                           36                          180
##                     KnowAxis                  X1990s2000s
##                          288                          144
## X1990s2000s.minus.1960s1970s                 CollegeMusic
##                          180                          108
##                      NoClass                     APTheory
##                          288                          216
##                    Composing                    PianoPlay
##                           72                            0
##                   GuitarPlay                    X1stInstr
##                            0                          756
##                    X2ndInstr                      first12
##                         1440                            0
##                    Classical                      Popular
##                           27                           27
```

```
## Assume anyone who plays only one insturment, and answered a value for one
## insturment but NA for second, has 0 skill with the second.
data2$X2ndInstr[((PianoPlay == 0) | (GuitarPlay == 0)) & (is.na(X2ndInstr)) &
    (!is.na(X1stInstr))] = 0

attach(data2)
```

```
## The following objects are masked from data2 (position 3):
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
## The following objects are masked from data:
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
```

```
subjects = unique(data2[, c(2, 6:26)])
str(na.omit(subjects))  ## Now our study is up to 30 subjects
```

```
## 'data.frame':    30 obs. of  22 variables:
##  $ Subject                     : Factor w/ 70 levels "15","16","17",..: 3 5
6 8 9 12 14 15 17 22 ...
##  $ Selfdeclare                 : int  2 2 4 2 3 1 2 4 2 2 ...
##  $ OMSI                        : int  67 67 784 67 319 194 88 567 147 11
...
##  $ X16.minus.17                : num  3 -4 2 -1 0 0 -1 0 0 -1 ...
##  $ ConsInstr                   : num  2.33 3.67 3 3 5 3 4.33 3 5 3 ...
##  $ ConsNotes                   : int  0 3 1 5 5 0 0 3 5 3 ...
##  $ Instr.minus.Notes           : num  2.33 0.67 2 -2 0 3 4.33 0 0 0 ...
##  $ PachListen                  : int  5 5 3 5 5 5 5 5 5 3 ...
##  $ ClsListen                   : int  1 1 0 0 1 1 3 1 1 1 ...
##  $ KnowRob                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ KnowAxis                    : int  0 0 0 0 0 5 5 0 0 0 ...
##  $ X1990s2000s                 : int  5 5 5 5 5 5 5 5 5 2 ...
##  $ X1990s2000s.minus.1960s1970s: int  5 2 5 4 3 3 2 3 2 0 ...
##  $ CollegeMusic                : int  1 0 1 1 1 1 1 0 0 1 ...
##  $ NoClass                     : int  0 0 1 0 1 1 1 0 0 0 ...
##  $ APTheory                    : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ Composing                   : int  0 0 1 0 0 0 0 0 1 0 ...
##  $ PianoPlay                   : int  0 0 0 0 0 0 0 5 0 0 ...
##  $ GuitarPlay                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stInstr                   : num  0 1 5 1 0 0 1 4 1 0 ...
##  $ X2ndInstr                   : num  0 0 0 0 0 0 1 0 0 0 ...
##  $ first12                     : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 3 3 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:40] 1 2 4 7 10 11 13 16
18 19 ...
##   .. ..- attr(*, "names")= chr [1:40] "1" "37" "109" "217" ...
```

```
temp = data2[is.na(Classical), ]
temp = temp[, 1:26]
str(na.omit(temp))
```

```
## 'data.frame':    0 obs. of  26 variables:
##  $ X                            : int 
##  $ Subject                      : Factor w/ 70 levels "15","16","17",..: 
##  $ Harmony                      : Factor w/ 4 levels "I-IV-V","I-V-IV",..: 
##  $ Instrument                   : Factor w/ 3 levels "guitar","piano",..: 
##  $ Voice                        : Factor w/ 3 levels "contrary","par3rd",..: 
##  $ Selfdeclare                  : int 
##  $ OMSI                         : int 
##  $ X16.minus.17                 : num 
##  $ ConsInstr                    : num 
##  $ ConsNotes                    : int 
##  $ Instr.minus.Notes            : num 
##  $ PachListen                   : int 
##  $ ClsListen                    : int 
##  $ KnowRob                      : int 
##  $ KnowAxis                     : int 
##  $ X1990s2000s                  : int 
##  $ X1990s2000s.minus.1960s1970s: int 
##  $ CollegeMusic                 : int 
##  $ NoClass                      : int 
##  $ APTheory                     : int 
##  $ Composing                    : int 
##  $ PianoPlay                    : int 
##  $ GuitarPlay                   : int 
##  $ X1stInstr                    : num 
##  $ X2ndInstr                    : num 
##  $ first12                      : Factor w/ 3 levels "guitar","piano",..: 
##  - attr(*, "na.action")=Class 'omit'  Named int [1:27] 1 2 3 4 5 6 7 8 9 10
...
##   .. ..- attr(*, "names")= chr [1:27] "326" "330" "334" "336" ...
```

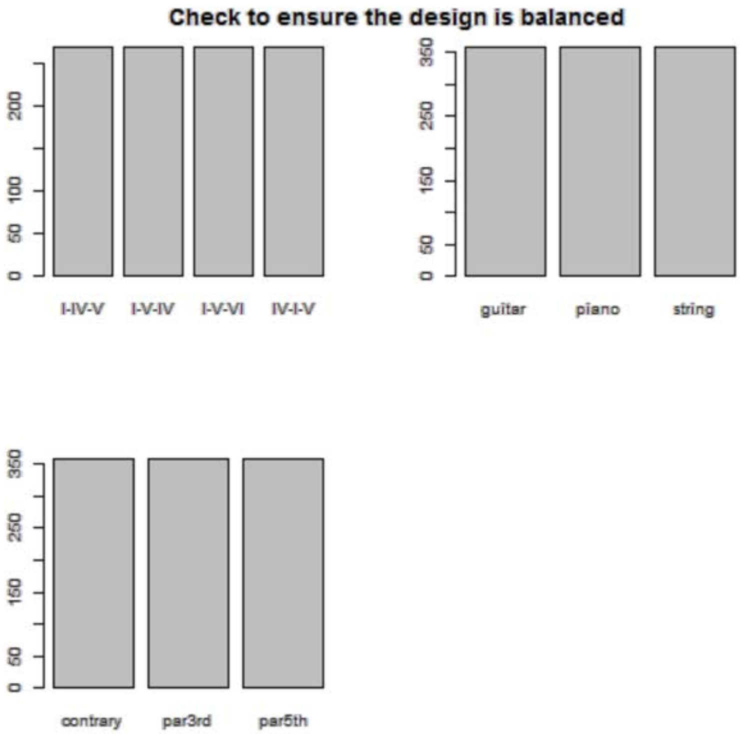
```
## all 27 of our bad observations are missing at least one other piece of
## information.  We're just going to stick with our 30 good subjects.  I know
## this is bad practice, but fixing these problems 'for real' seems out of
## scope for this assignment.


data3 = na.omit(data2)
str(data3)
```

```
## 'data.frame':    1080 obs. of  28 variables:
##  $ X                           : int  73 74 75 76 77 78 79 80 81 82 ...
##  $ Subject                     : Factor w/ 70 levels "15","16","17",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ Harmony                     : Factor w/ 4 levels "I-IV-V","I-V-IV",..: 1
1 1 1 1 1 1 1 1 2 ...
##  $ Instrument                  : Factor w/ 3 levels "guitar","piano",..: 1
1 1 2 2 2 3 3 3 1 ...
##  $ Voice                       : Factor w/ 3 levels "contrary","par3rd",..:
1 2 3 1 2 3 1 2 3 1 ...
##  $ Selfdeclare                 : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ OMSI                        : int  67 67 67 67 67 67 67 67 67 67 ...
##  $ X16.minus.17                : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ ConsInstr                   : num  2.33 2.33 2.33 2.33 2.33 2.33 2.33
2.33 2.33 2.33 ...
##  $ ConsNotes                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Instr.minus.Notes           : num  2.33 2.33 2.33 2.33 2.33 2.33 2.33
2.33 2.33 2.33 ...
##  $ PachListen                  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ ClsListen                   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ KnowRob                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ KnowAxis                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1990s2000s                 : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1990s2000s.minus.1960s1970s: int  5 5 5 5 5 5 5 5 5 5 ...
##  $ CollegeMusic                : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NoClass                     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APTheory                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Composing                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PianoPlay                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GuitarPlay                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stInstr                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ X2ndInstr                   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ first12                     : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 3 3 ...
##  $ Classical                   : num  2 1 4 2 4 5 7 4 9 3 ...
##  $ Popular                     : num  10 8 7 8 8 6 2 6 5 9 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:1440] 1 2 3 4 5 6 7 8 9
10 ...
##   .. ..- attr(*, "names")= chr [1:1440] "1" "2" "3" "4" ...
```

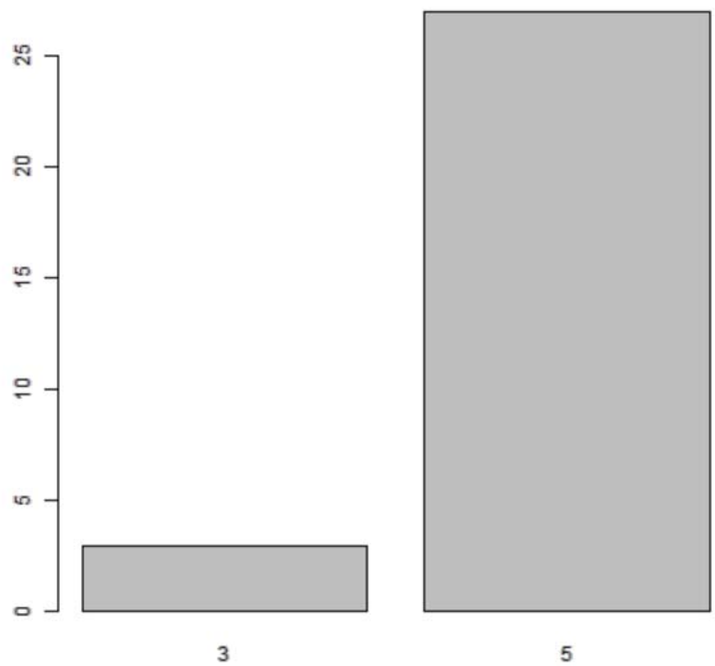
```
attach(data3)
```

```
## The following objects are masked from data2 (position 3):
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
## The following objects are masked from data2 (position 4):
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
## The following objects are masked from data:
##
##     APTheory, Classical, ClsListen, CollegeMusic, Composing,
##     ConsInstr, ConsNotes, first12, GuitarPlay, Harmony,
##     Instr.minus.Notes, Instrument, KnowAxis, KnowRob, NoClass,
##     OMSI, PachListen, PianoPlay, Popular, Selfdeclare, Subject,
##     Voice, X, X16.minus.17, X1990s2000s,
##     X1990s2000s.minus.1960s1970s, X1stInstr, X2ndInstr
```

```
###### Second EDA (on complete data)

par(mfrow = c(2, 2))
plot(Harmony)
plot(Instrument)
plot(Voice)
par(mfrow = c(1, 1))
title("Check to ensure the design is balanced")
```



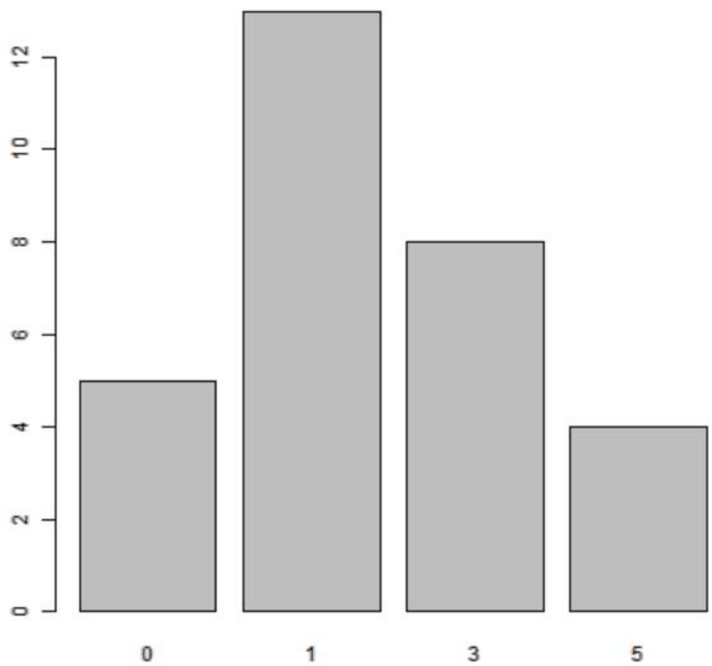

```
## all of our data is evenly distributed, so so a repeated measures model is
## valid

# hist(Selfdeclare) hist(OMSI) hist(X16.minus.17) hist(ConsInstr)
# hist(ConsNotes) hist(Instr.minus.Notes)

barplot(table(PachListen)/36)
```




```
table(table(PachListen)/36)
```

```
##
##  3 27
##  1  1
```

```
## Now just 1 sibject didn't answer '5' We should exclude this variable

barplot(table(ClsListen)/36)
```




```
table(table(ClsListen)/36)
```

```
##
##  4  5  8 13
##  1  1  1  1
```

```
## I still don't know how to interpret this one.  Since no questions
## specifically ask about this, I'm excluding it as well.

# hist(X1990s2000s) ## a lot of 5s hist(X1990s2000s.minus.1960s1970s)
# X1960s1970s = X1990s2000s - X1990s2000s.minus.1960s1970s hist(X1960s1970s)
# ## more even spread

summary(as.factor(CollegeMusic))  ## 0 NAs!
```

```
##   0   1
## 252 828
```

```
table(NoClass)  ## more people took classes in college, than have ever taken
classes
```

```
## NoClass
##   0   1   2
## 396 612  72
```

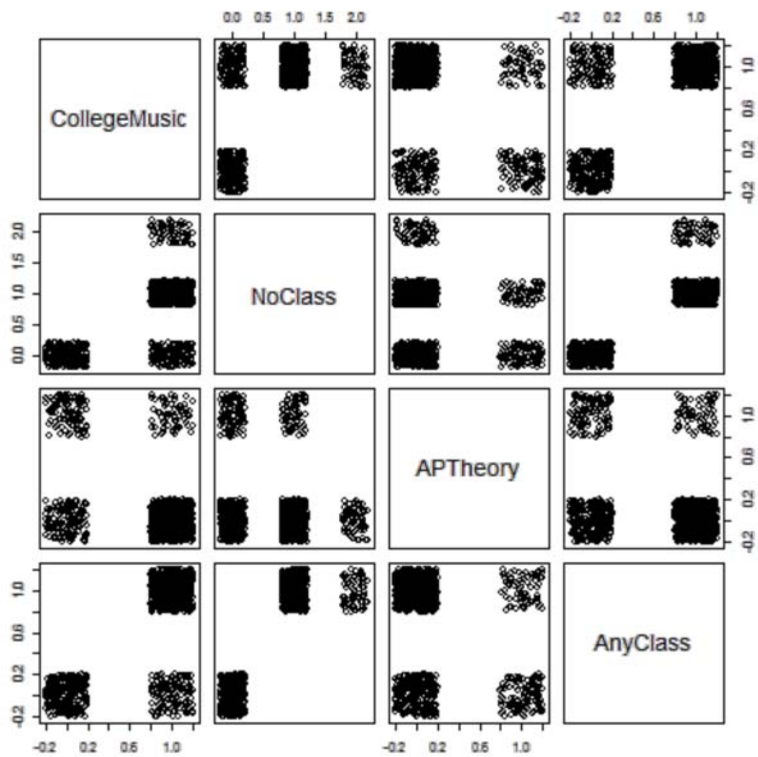
```
## Subjects have taken only 0, 1, or 2 classes Maybe we should use AnyClass
## instead?

AnyClass = NoClass
AnyClass[NoClass > 0] = 1
summary(AnyClass)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   0.633   1.000   1.000
```

```
summary(as.factor(APTheory))  ## 0 NAs!
```

```
##   0   1
## 900 180
```

```
pairs(jitter(cbind(CollegeMusic, NoClass, APTheory, AnyClass)))
```




```
## Yeah there are clearly people who have taken 0 classes, who took AP Theory
## or courses in college!

par(mfrow = c(2, 2))
hist(PianoPlay, main = "")  ## We eliminated 20/28 of our piano players
sum(data$PianoPlay > 0)/36
```

```
## [1] 29
```

```
sum(PianoPlay > 0)/36
```

```
## [1] 8
```

```
hist(GuitarPlay, main = "")  ## And 16/18 of our guitar players
sum(data$GuitarPlay > 0)/36
```

```
## [1] 18
```

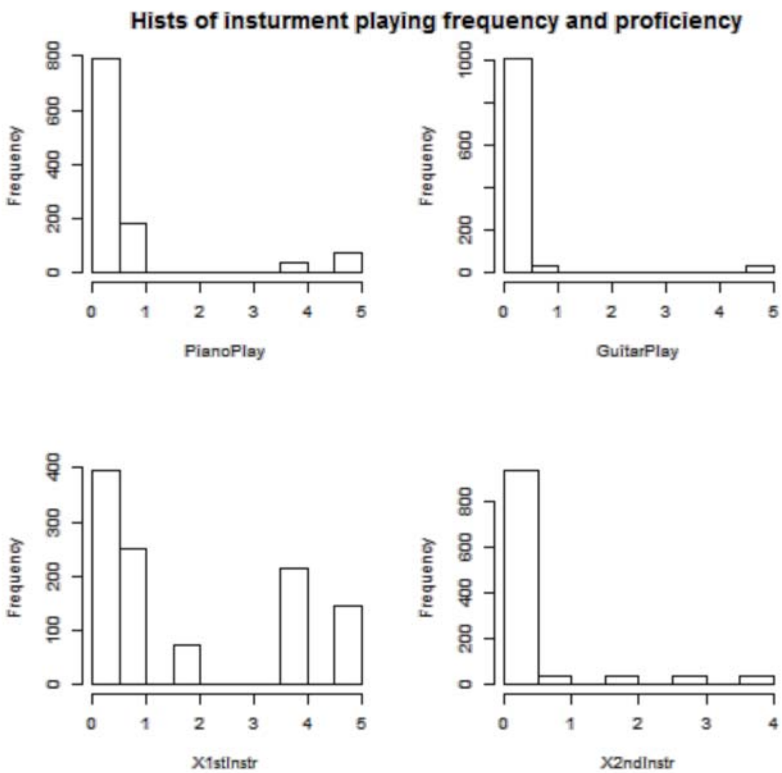
```
sum(GuitarPlay > 0)/36
```

```
## [1] 2
```

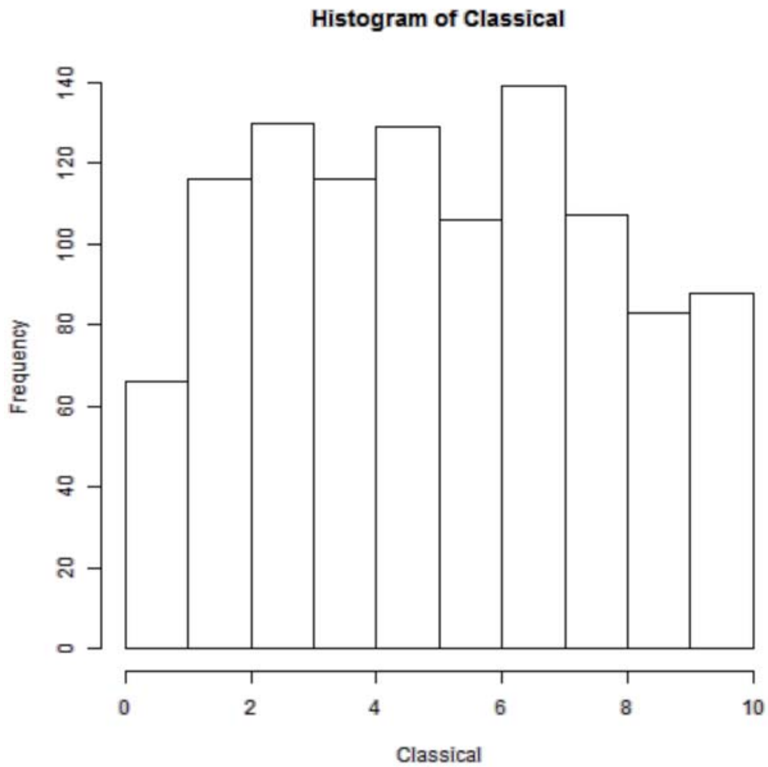
```
hist(X1stInstr, main = "")
table(X1stInstr)  ## no 0s. Everyone plays an insturment.
```

```
## X1stInstr
##   0   1   2   4   5
## 396 252  72 216 144
```

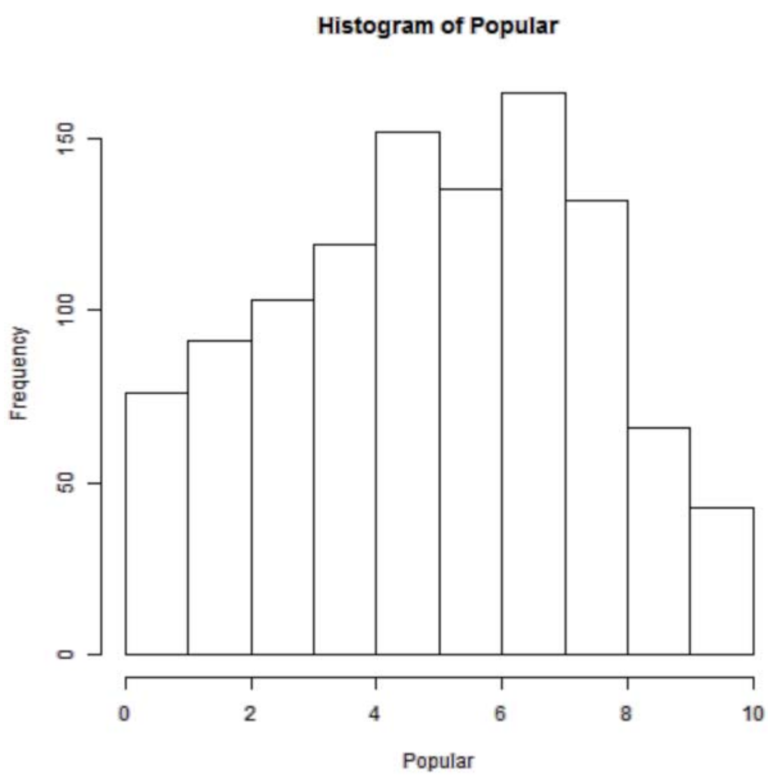
```
hist(X2ndInstr, main = "")
par(mfrow = c(1, 1))
title("Hists of insturment playing frequency and proficiency")
```




```
hist(Classical)
```




```
hist(Popular)
```




```
summary(Classical)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00    5.00    5.44    8.00   10.00
```

```
summary(Popular)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.38    5.00    5.39    7.00   10.00
```

```
str(data3)
```

```
## 'data.frame':    1080 obs. of  28 variables:
##  $ X                          : int  73 74 75 76 77 78 79 80 81 82 ...
##  $ Subject                    : Factor w/ 70 levels "15","16","17",..: 3 3
3 3 3 3 3 3 3 3 ...
##  $ Harmony                    : Factor w/ 4 levels "I-IV-V","I-V-IV",..: 1
1 1 1 1 1 1 1 1 2 ...
##  $ Instrument                 : Factor w/ 3 levels "guitar","piano",..: 1
1 1 2 2 2 3 3 3 1 ...
##  $ Voice                      : Factor w/ 3 levels "contrary","par3rd",..:
1 2 3 1 2 3 1 2 3 1 ...
##  $ Selfdeclare                : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ OMSI                       : int  67 67 67 67 67 67 67 67 67 67 ...
##  $ X16.minus.17               : num  3 3 3 3 3 3 3 3 3 3 ...
##  $ ConsInstr                  : num  2.33 2.33 2.33 2.33 2.33 2.33 2.33
2.33 2.33 2.33 ...
##  $ ConsNotes                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Instr.minus.Notes          : num  2.33 2.33 2.33 2.33 2.33 2.33 2.33
2.33 2.33 2.33 ...
##  $ PachListen                 : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ ClsListen                  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ KnowRob                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ KnowAxis                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1990s2000s                : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ X1990s2000s.minus.1960s1970s: int  5 5 5 5 5 5 5 5 5 5 ...
##  $ CollegeMusic               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NoClass                    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APTheory                   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Composing                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ PianoPlay                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ GuitarPlay                 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X1stInstr                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ X2ndInstr                  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ first12                    : Factor w/ 3 levels "guitar","piano",..: 3
3 3 3 3 3 3 3 3 3 ...
##  $ Classical                  : num  2 1 4 2 4 5 7 4 9 3 ...
##  $ Popular                    : num  10 8 7 8 8 6 2 6 5 9 ...
##  - attr(*, "na.action")=Class 'omit'  Named int [1:1440] 1 2 3 4 5 6 7 8 9
10 ...
##   .. ..- attr(*, "names")= chr [1:1440] "1" "2" "3" "4" ...
```

So, our final data set is 30 subjects with either complete data or easily explainable missing entries. Since a number of our variables have noticeably different spreads in our final data set than they did in our original, I'm going to avoid including variables in our model unless they are specifically called for.

1.

a)

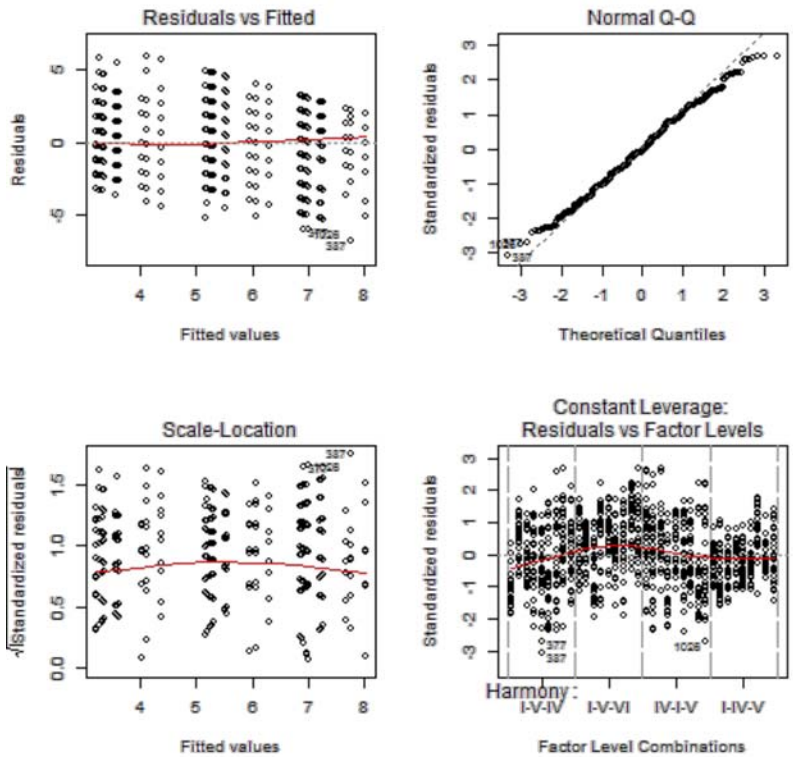
```
model1a_C = lm(Classical ~ Harmony + Instrument + Voice)
summary(model1a_C)
```

```
##
## Call:
## lm(formula = Classical ~ Harmony + Instrument + Voice)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.767 -1.509 -0.154  1.736  5.879
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.5815     0.1895   18.90  < 2e-16 ***
## HarmonyI-V-IV      0.0370     0.1895    0.20    0.845
## HarmonyI-V-VI      0.7907     0.1895    4.17  3.3e-05 ***
## HarmonyIV-I-V     -0.0111     0.1895   -0.06    0.953
## Instrumentpiano    1.9389     0.1641   11.81  < 2e-16 ***
## Instrumentstring   3.6458     0.1641   22.21  < 2e-16 ***
## Voicepar3rd       -0.3556     0.1641   -2.17    0.031 *
## Voicepar5th       -0.2514     0.1641   -1.53    0.126
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.2 on 1072 degrees of freedom
## Multiple R-squared:  0.329,  Adjusted R-squared:  0.324
## F-statistic: 74.9 on 7 and 1072 DF,  p-value: <2e-16
```

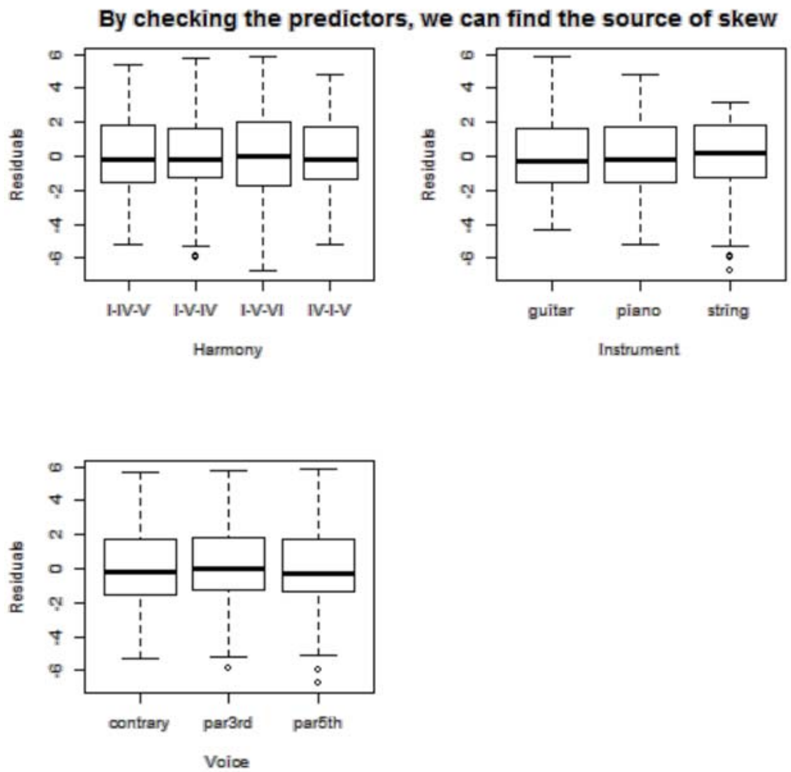
```
anova(model1a_C)  ## All three effects are significant
```

```
## Analysis of Variance Table
##
## Response: Classical
##              Df Sum Sq Mean Sq F value  Pr(>F)
## Harmony       3    124      41    8.54 1.3e-05 ***
## Instrument    2   2396    1198  247.03 < 2e-16 ***
## Voice         2     24      12    2.48   0.084 .
## Residuals  1072   5198       5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

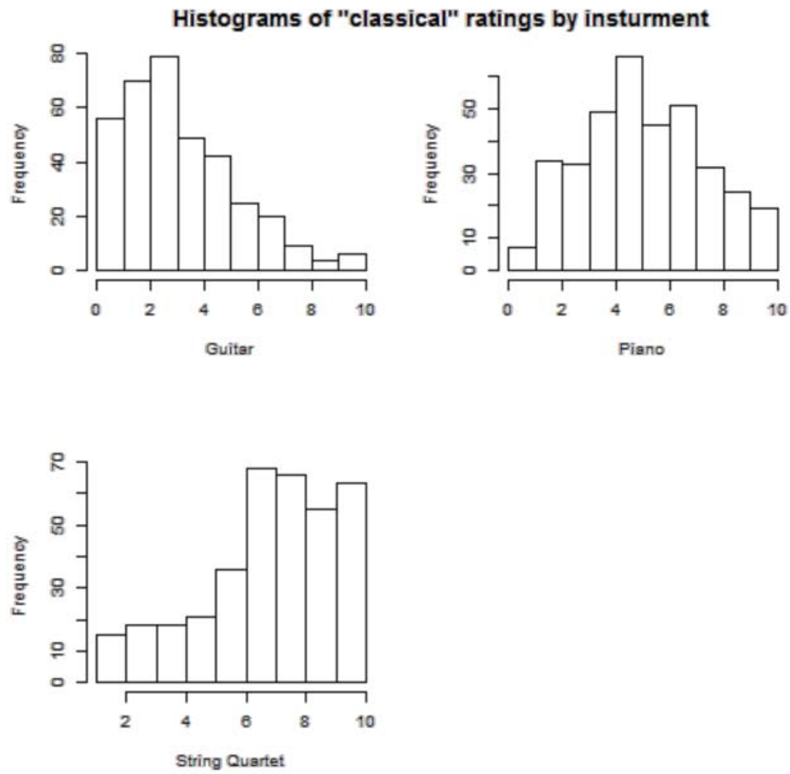
```
par(mfrow = c(2, 2))
plot(model1a_C)
```




```
par(mfrow = c(1, 1))
## Well, those residuals are terrible.


###### Diagnostics sidebar
par(mfrow = c(2, 2))
plot(model1a_C$resid ~ Harmony, ylab = "Residuals")
plot(model1a_C$resid ~ Instrument, ylab = "Residuals")  ## There we go
plot(model1a_C$resid ~ Voice, ylab = "Residuals")
par(mfrow = c(1, 1))
title("By checking the predictors, we can find the source of skew")
```




```
## Okay so the skewness of our distribution varies based on the insturment?
par(mfrow = c(2, 2))
hist(Classical[Instrument == "guitar"], main = "", xlab = "Guitar")
hist(Classical[Instrument == "piano"], main = "", xlab = "Piano")
hist(Classical[Instrument == "string"], main = "", xlab = "String Quartet")
par(mfrow = c(1, 1))
title("Histograms of \"classical\" ratings by insturment")
```




```
## Oh this actually probably isn't an issue. The mean of each response is
## changing, but since our response variable is bounded at 0 and 10, the
## distribution just looks more skewed than it really is.
```

In the determination of classical music, all three predictor variables (Harmony, Insturment, and Voice) are significant. The most important predictor is Insturment (SSE = 23.958, $p < 2.2*10^{-16}$). Next most important is Harmony (SSE = 124.2, p = $1.321 * 10^{-5}$). Finally Voice makes a small contribution in our model (SSE = 24.1, p = 0.08419). Combining these gives us a model $R^2 = 0.3286$ and $p < 2.2*10^{-16}$. However, the combination of these three doesn't explain that much, since the residuals squares error is still 5198.4, making our $R^2 = 0.324$.

While not all levels of each factor were statistically significant, there are still some interesting results. Our "baseline" song is one with harmonic motion "I-VI-V", played on the electric guitar, and with a voice leading in contrary motion. This song had an expected "Classical" rating of 3.58148. As the researchers predicted, a song in harmonic motion "I-V-VI" was was rated significantly higher (β = 0.79074, p = $3.26*10^{-5}$). Piano music was rated significantly higher than electric guitar music (β = 1.93889, $p < 2.2*10^{-16}$) as was the string quartet (β = 3.64583, p = $p < 2.2*10^{-16}$). Consistent with the researchers' hypothesis, contrary motion was rated as more classical than either parallel 3rds (β = -0.35556, p = 0.0305) or parallel 5ths (β = -0.25139, p = 0.1259).

b)

i.

$$Classical_{ij} = Subject_j + \beta_{Harmony} Harmony_i + \beta_{Instrument} Instrument_i + \beta_{Voice} Voice_i + \epsilon_{ij}$$

$$Subject_j = \beta_0 + \eta_j$$

$$\epsilon_{ij} \sim N(0, \sigma^2)$$

$$\eta_j \sim N(0, \tau^2)$$

Where “i” is the song number and “j” is the subject index. Note that since our fixed effects are actually categorical variables, each variable actually represents a vector of indicator variables, and each β is the vector of weights for each indicator.

ii.

```
mem1_C = lmer(Classical ~ Harmony + Instrument + Voice + (1 | Subject))
display(mem1_C)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + Voice + (1 |
##     Subject))
##                  coef.est coef.se
## (Intercept)       3.58     0.27
## HarmonyI-V-IV     0.04     0.16
## HarmonyI-V-VI     0.79     0.16
## HarmonyIV-I-V    -0.01     0.16
## Instrumentpiano   1.94     0.14
## Instrumentstring  3.65     0.14
## Voicepar3rd      -0.36     0.14
## Voicepar5th      -0.25     0.14
##
## Error terms:
##  Groups   Name        Std.Dev.
##  Subject  (Intercept) 1.16
##  Residual             1.88
## ---
## number of obs: 1080, groups: Subject, 30
## AIC = 4538.6, DIC = 4486
## deviance = 4502.4
```

```
# summary(mem1_C)

AIC(mem1_C, model1a_C)
```

```
##            df  AIC
## mem1_C     10 4539
## model1a_C   9 4780
```

```
BIC(mem1_C, model1a_C)
```

```
##            df  BIC
## mem1_C     10 4588
## model1a_C   9 4825
```

The AIC and BIC are both meaningfully lower for our mixed-effects model than they are for our fixed-effects model.

iii.

The point estimates of the fixed effects of the random-effects model are identical to their counterparts in the model with only fixed effects. While R does not provide clear p-values, we can see that the coefficent standard error for “Harmony” has gone from $\approx$ 0.19 to $\approx$ 0.16, indicating greater significance in our mixed-effects model. Likewise, Insturment's s.e. dropped from $\approx$ 0.16 to 0.14 and Voice's s.e. dropped from $\approx$ 0.16 to 0.14. All three effects were significant before, and they are moreso now.

c)

i.

```
mem2_C = lmer(Classical ~ Harmony + Instrument + Voice + (1 | Subject:Harmony)
+
    (1 | Subject:Instrument) + (1 | Subject:Voice))
display(mem2_C)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + Voice + (1 |
##     Subject:Harmony) + (1 | Subject:Instrument) + (1 | Subject:Voice))
##                  coef.est coef.se
## (Intercept)       3.58     0.31
## HarmonyI-V-IV     0.04     0.23
## HarmonyI-V-VI     0.79     0.23
## HarmonyIV-I-V    -0.01     0.23
## Instrumentpiano   1.94     0.35
## Instrumentstring  3.65     0.35
## Voicepar3rd      -0.36     0.15
## Voicepar5th      -0.25     0.15
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.72
##  Subject:Voice      (Intercept) 0.34
##  Subject:Instrument (Intercept) 1.28
##  Residual                       1.56
## ---
## number of obs: 1080, groups: Subject:Harmony, 120; Subject:Voice, 90;
Subject:Instrument, 90
## AIC = 4382.9, DIC = 4337
## deviance = 4347.8
```

```
# summary(mem2_C)
anova(mem2_C)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Harmony     3   42.6    14.2    5.83
## Instrument  2  265.4   132.7   54.45
## Voice       2   15.3     7.6    3.13
```

```


AIC(mem2_C, mem1_C, model1a_C)
```

```
##           df  AIC
## mem2_C    12 4383
## mem1_C    10 4539
## model1a_C  9 4780
```

```
BIC(mem2_C, mem1_C, model1a_C)
```

```
##           df  BIC
## mem2_C    12 4443
## mem1_C    10 4588
## model1a_C  9 4825
```

The version of the model with random draws for each insturment-person combination has the lowest AIC and BIC yet, and is therefore our best.

ii.

Once again, our point estimates are unchanged. However in this model, the s.e.s have increased. Harmony's s.e. is now $\approx$ 0.23, though this is still small compared to $\beta_{Harmony_{I-V-VI}} = 0.79$. The s.e. of insturment has risen to $\approx$ 0.35, but again this is small compared to $\beta_{Harmony} = [1.94, 3.65]^T$. Voice's s.e. is up to $\approx$ 0.15, which is close to $\beta_{Voice} = [-0.36, -.025]^T$. Voice may no longer be statistically significant in this model.

iii.

$$Classical_{ij} = \beta_0 + \beta_{Harmony} Harmony_i + \beta_{Instrument} Instrument_i + \beta_{Voice} Voice_i + \epsilon_{ij} \\ + SubjectHarmony_{jk[i]} + SubjectInsturment_{jl[i]} + SubjectVoice_{jm[i]}$$

$$SubjectHarmony_{jk[i]} = \eta_{j1}$$

$$SubjectInsturment_{jl[i]} = \eta_{j2}$$

$$SubjectVoice_{jm[i]} = \eta_{j3}$$

$$\epsilon_{ij} \sim N(0, \sigma^2)$$

$$\eta_{j1} \sim N(0, \tau_1^2)$$

$$\eta_{j2} \sim N(0, \tau_2^2)$$

$$\eta_{j3} \sim N(0, \tau_3^2)$$

Where “i” is the song number, “j” is the subject index, k[i] is the harmonic structure designated for song i, l[i] is the primary insturment designated for song i, and m[i] is the vocal style for song i.

2

a)

Everything besides Harmony, Instrument, and Voice is basically just a description of the subject, which is clearly a random effect. Therefore, all of those effects will likely want to be treated as random as well. Additionally, as we saw in EDA, many of those variables are rife with issues. Instead, let's analyze if Harmony, Insturment, and Voice are all still needed in the model.

```
mem2_C_HI = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject:Voice))
mem2_C_HV = lmer(Classical ~ Harmony + Voice + (1 | Subject:Harmony) + (1 |
    Subject:Instrument) + (1 | Subject:Voice))
mem2_C_VI = lmer(Classical ~ Instrument + Voice + (1 | Subject:Harmony) + (1 |
    Subject:Instrument) + (1 | Subject:Voice))

AIC(mem2_C, mem2_C_HI, mem2_C_HV, mem2_C_VI)
```

```
##           df  AIC
## mem2_C    12 4383
## mem2_C_HI 10 4381
## mem2_C_HV 10 4446
## mem2_C_VI  9 4389
```

```
BIC(mem2_C, mem2_C_HI, mem2_C_HV, mem2_C_VI)
```

```
##            df  BIC
## mem2_C    12 4443
## mem2_C_HI 10 4431
## mem2_C_HV 10 4496
## mem2_C_VI  9 4434
```

It appears that excluding Voice improves BIC meaningfully and AIC marginally.

```
mem2_C_H = lmer(Classical ~ Harmony + (1 | Subject:Harmony) + (1 |
Subject:Instrument) +
    (1 | Subject:Voice))
mem2_C_I = lmer(Classical ~ Instrument + (1 | Subject:Harmony) + (1 |
Subject:Instrument) +
    (1 | Subject:Voice))

AIC(mem2_C_HI, mem2_C_H, mem2_C_I)
```

```
##            df  AIC
## mem2_C_HI 10 4381
## mem2_C_H   8 4444
## mem2_C_I   7 4387
```

```
BIC(mem2_C_HI, mem2_C_H, mem2_C_I)
```

```
##            df  BIC
## mem2_C_HI 10 4431
## mem2_C_H   8 4484
## mem2_C_I   7 4422
```

Our tests disagree here. AIC tells us to keep both fixed effects, but BIC says to drop Harmony. I'm a little concerened that we're overfitting our random effects, so for now I'm going to choose the model with both fixed effects, Harmony and Insturment.

b)

```
## First let's try to take out the random effect for voice
mem_best = mem2_C_HI
mem_novoice = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject))  ## try eliminating voice
entirely, adding 'Subject' back in
mem_novoice2 = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Instrument))  ## try eliminating voice entirely
mem_ranslopes = lmer(Classical ~ Harmony + Instrument + (1 + Harmony |
Subject) +
    (1 + Instrument | Subject))  ## Random slopes instead of random draws
mem_simple = lmer(Classical ~ Harmony + Instrument + (1 | Subject))  ##
reminder of our original MEM

AIC(mem_best, mem_novoice, mem_novoice2, mem_ranslopes, mem_simple)
```

```
##                df  AIC
## mem_best      10 4381
## mem_novoice   10 4376
## mem_novoice2   9 4387
## mem_ranslopes 23 4360
## mem_simple     8 4537
```

```
BIC(mem_best, mem_novoice, mem_novoice2, mem_ranslopes, mem_simple)
```

```
##               df  BIC
## mem_best      10 4431
## mem_novoice   10 4426
## mem_novoice2   9 4432
## mem_ranslopes 23 4475
## mem_simple     8 4577
```

Inconclusive, though mem_novoice is strictly better than our previous best, and mem_ranslopes shows promise. Maybe we should experiment more with random slopes?

```
mem_ranslope_harm = lmer(Classical ~ Harmony + Instrument + (1 + Harmony |
Subject))
mem_ranslope_inst = lmer(Classical ~ Harmony + Instrument + (1 + Instrument |
    Subject))

AIC(mem_best, mem_ranslope_harm, mem_ranslope_inst)
```

```
##                   df  AIC
## mem_best          10 4381
## mem_ranslope_harm 17 4510
## mem_ranslope_inst 13 4416
```

```
BIC(mem_best, mem_ranslope_harm, mem_ranslope_inst)
```

```
##                   df  BIC
## mem_best          10 4431
## mem_ranslope_harm 17 4594
## mem_ranslope_inst 13 4480
```

Nope. What about fewer random draws?

```
mem_noharm = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Instrument)
+
    (1 | Subject:Voice) + (1 | Subject))
mem_noinst = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Voice) + (1 | Subject))
mem_noharm2 = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Instrument)
+
    (1 | Subject:Voice))
mem_noinst2 = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Voice))

AIC(mem_best, mem_novoice, mem_noharm, mem_noinst, mem_noharm2, mem_noinst2)
```

```
##             df  AIC
## mem_best    10 4381
## mem_novoice 10 4376
## mem_noharm  10 4420
## mem_noinst  10 4516
## mem_noharm2  9 4434
## mem_noinst2  9 4567
```

```
BIC(mem_best, mem_novoice, mem_noharm, mem_noinst, mem_noharm2, mem_noinst2)
```

```
##             df  BIC
## mem_best    10 4431
## mem_novoice 10 4426
## mem_noharm  10 4470
## mem_noinst  10 4566
## mem_noharm2  9 4478
## mem_noinst2  9 4611
```

Nope.

```
mem_randrand = lmer(Classical ~ Harmony + Instrument + (1 + Harmony | 
Subject:Harmony) + 
    (1 + Instrument | Subject:Instrument))  ## This is probably a bad idea
```

```
## Warning: failure to converge in 10000 evaluations
```

```


AIC(mem_best, mem_randrand)
```

```
##              df  AIC
## mem_best     10 4381
## mem_randrand 23 4386
```

```
BIC(mem_best, mem_randrand)
```

```
##              df  BIC
## mem_best     10 4431
## mem_randrand 23 4501
```

```
## it is!
```

```
mem_ranslopes_voice = lmer(Classical ~ Harmony + Instrument + (1 + Harmony | 
    Subject) + (1 + Instrument | Subject) + (1 + Voice | Subject))  ## Random 
slopes instead of random draws
```

```
## Warning: failure to converge in 10000 evaluations
```

```


AIC(mem_best, mem_ranslopes_voice, mem_ranslopes)
```

```
##                     df  AIC
## mem_best            10 4381
## mem_ranslopes_voice 29 4367
## mem_ranslopes       23 4360
```

```
BIC(mem_best, mem_ranslopes_voice, mem_ranslopes)
```

```
##                     df  BIC
## mem_best            10 4431
## mem_ranslopes_voice 29 4512
## mem_ranslopes       23 4475
```

The random slopes model is better off without voice.

```
AIC(mem_novoice, mem_ranslopes)
```

```
##               df  AIC
## mem_novoice   10 4376
## mem_ranslopes 23 4360
```

```
BIC(mem_novoice, mem_ranslopes)
```

```
##                df  BIC
## mem_novoice   10 4426
## mem_ranslopes 23 4475
```

It doesn't appear that we'll be able to determine which of these two models is superior withour some more involved selection critera, such as cross-validation.

```
display(mem_novoice)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
##     (1 | Subject:Instrument) + (1 | Subject))
##                   coef.est coef.se
## (Intercept)        3.38     0.31
## HarmonyI-V-IV      0.04     0.22
## HarmonyI-V-VI      0.79     0.22
## HarmonyIV-I-V     -0.01     0.22
## Instrumentpiano    1.94     0.29
## Instrumentstring   3.65     0.29
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.67
##  Subject:Instrument (Intercept) 1.00
##  Subject            (Intercept) 0.96
##  Residual                       1.60
## ---
## number of obs: 1080, groups: Subject:Harmony, 120; Subject:Instrument, 90;
Subject, 30
## AIC = 4376.5, DIC = 4342
## deviance = 4349.3
```

```
display(mem_ranslopes)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + (1 + Harmony |
##     Subject) + (1 + Instrument | Subject))
##                   coef.est coef.se
## (Intercept)        3.38     0.21
## HarmonyI-V-IV      0.04     0.16
## HarmonyI-V-VI      0.79     0.34
## HarmonyIV-I-V     -0.01     0.16
## Instrumentpiano    1.94     0.23
## Instrumentstring   3.65     0.32
##
## Error terms:
##  Groups    Name             Std.Dev. Corr
##  Subject   (Intercept)      0.53
##            HarmonyI-V-IV    0.42     0.95
##            HarmonyI-V-VI    1.72     0.47  0.33
##            HarmonyIV-I-V    0.44     0.79  0.93  0.00
##  Subject.1 (Intercept)      0.81
##            Instrumentpiano  1.07      0.15
##            Instrumentstring 1.61     -0.33  0.62
##  Residual                   1.58
## ---
## number of obs: 1080, groups: Subject, 30
## AIC = 4359.9, DIC = 4285
## deviance = 4299.6
```

```
anova(mem_novoice)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Harmony     3     48      16    6.28
## Instrument  2    416     208   81.75
```

```
anova(mem_ranslopes)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Harmony     3     14     4.7    1.89
## Instrument  2    342   171.0   68.25
```

I'm unable to choose between these two models, and really, I shouldn't have to. The correct choice here is to choose both models as our class of "best models". The bright side is that the fixed effects are the same for both of these models, so our analysis conclusions will not care which we choose.

c)

While not all levels of each factor were statistically significant, here are the results which are. The expected "Classical" score for a song with harmonic motion "I-VI-V", played on the electric guitar, and with a voice leading in contrary motion (our baseline) was 3.37916667. As the researchers predicted, a song in harmonic motion "I-V-VI" was was rated as much more classical (β = 0.79074). A song in harmonic motion "I-V-IV" was rated as only marginally more classical (β = 0.037037) and Harmonic motion of "IV-I-V" was rated as marginally less classical (β = -0.01111111). Piano music was rated as much more classical than electric guitar music (β = 1.93889) as was the string quartet (β = 3.64583). Note that these results are extremely similar to the results we saw in the model without any random effects.

3.

a)

```
table(Selfdeclare)/36
```

```
## Selfdeclare
##  1  2  3  4  5
##  2 17  6  4  1
```

Let's designate the five subjects who responded with a 4 or 5 as "musicians" and five random subjects who responded with a 1 or 2 "non-musicians."

```
mus = unique(Subject[Selfdeclare > 3])
set.seed(1)
nmus = sample(Subject[Selfdeclare < 3], 5)

unlist(list(mus, nmus))
```

```
##  [1] 20 30 49 66 94 32 40 55 80 26
## 70 Levels: 15 16 17 18b 19 20 21 22 23 24 25 26 28 29 30 31 32 33 ... 98
```

```
data4 = data3[Subject %in% unlist(list(mus, nmus)), ]

musician = data4$Selfdeclare > 3
# mean(musician) ## Everything is in order

mem_ranslopes_music = lmer(Classical ~ Harmony + Instrument + musician + (1 +
    Harmony | Subject) + (1 + Instrument | Subject), data = data4)
```

```
## Warning: failure to converge in 10000 evaluations
```

```
display(mem_ranslopes_music)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + musician +
##     (1 + Harmony | Subject) + (1 + Instrument | Subject), data = data4)
##                  coef.est coef.se
## (Intercept)       3.72     0.59
## HarmonyI-V-IV    -0.03     0.23
## HarmonyI-V-VI     1.06     0.92
## HarmonyIV-I-V    -0.21     0.24
## Instrumentpiano   1.97     0.42
## Instrumentstring  3.34     0.50
## musicianTRUE     -0.19     0.74
##
## Error terms:
##  Groups    Name             Std.Dev. Corr
##  Subject   (Intercept)      1.01
##            HarmonyI-V-IV    0.28     -0.93
##            HarmonyI-V-VI    2.83      0.10 -0.10
##            HarmonyIV-I-V    0.34      0.24  0.00  0.74
##  Subject.1 (Intercept)      0.85
##            Instrumentpiano  1.21     -0.15
##            Instrumentstring 1.47     -0.48  0.72
##  Residual                   1.41
## ---
## number of obs: 360, groups: Subject, 10
## AIC = 1412.4, DIC = 1340
## deviance = 1352.4
```

```
anova(mem_ranslopes_music)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Harmony     3    7.8     2.6    1.31
## Instrument  2   89.0    44.5   22.35
## musician    1    0.1     0.1    0.06
```

```
mem_novoice_music = lmer(Classical ~ Harmony + Instrument + musician + (1 |
    Subject:Harmony) + (1 | Subject:Instrument) + (1 | Subject), data = data4)
display(mem_novoice_music)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + musician +
##     (1 | Subject:Harmony) + (1 | Subject:Instrument) + (1 | Subject),
##     data = data4)
##                  coef.est coef.se
## (Intercept)       3.53     0.57
## HarmonyI-V-IV    -0.03     0.40
## HarmonyI-V-VI     1.06     0.40
## HarmonyIV-I-V    -0.21     0.40
## Instrumentpiano   1.97     0.44
## Instrumentstring  3.34     0.44
## musicianTRUE      0.20     0.63
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.74
##  Subject:Instrument (Intercept) 0.89
##  Subject            (Intercept) 0.73
##  Residual                       1.44
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1400.5, DIC = 1377
## deviance = 1377.9
```

```
anova(mem_novoice_music)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Harmony     3   26.7     8.9    4.28
## Instrument  2  121.2    60.6   29.07
## musician    1    0.2     0.2    0.10
```

The effect is very small, if it exists at all, as an ordinary fixed effect. The model with random draws for each subject/harmony and subject/insturment pair seems to be doing a better job on this data subset (random slopes model doesn't converge), so I'll be using that one exclusively going forward.

```
mem_music1 = mem_novoice_music
mem_best = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) + (1
|
    Subject:Instrument) + (1 | Subject), data = data4)

mem_music2 = lmer(Classical ~ Harmony * musician + Instrument * musician + (1
|
    Subject:Harmony) + (1 | Subject:Instrument) + (1 | Subject), data = data4)

mem_music3 = lmer(Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
    (1 | Subject:Instrument) + (1 + musician | Subject), data = data4)

AIC(mem_best, mem_music1, mem_music2, mem_music3)
```

```
##            df  AIC
## mem_best   10 1400
## mem_music1 11 1401
## mem_music2 16 1394
## mem_music3 12 1403
```

```
BIC(mem_best, mem_music1, mem_music2, mem_music3)  ## we can throw out 3
```

```
##            df  BIC
## mem_best   10 1438
## mem_music1 11 1443
## mem_music2 16 1456
## mem_music3 12 1449
```

```

display(mem_music2)
```

```
## lmer(formula = Classical ~ Harmony * musician + Instrument *
##     musician + (1 | Subject:Harmony) + (1 | Subject:Instrument) +
##     (1 | Subject), data = data4)
##                                coef.est coef.se
## (Intercept)                     3.59     0.65
## HarmonyI-V-IV                  -0.09     0.49
## HarmonyI-V-VI                   0.08     0.49
## HarmonyIV-I-V                  -0.29     0.49
## musicianTRUE                    0.07     0.92
## Instrumentpiano                 2.13     0.64
## Instrumentstring                3.82     0.64
## HarmonyI-V-IV:musicianTRUE      0.11     0.70
## HarmonyI-V-VI:musicianTRUE      1.97     0.70
## HarmonyIV-I-V:musicianTRUE      0.16     0.70
## musicianTRUE:Instrumentpiano   -0.32     0.90
## musicianTRUE:Instrumentstring  -0.97     0.90
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.61
##  Subject:Instrument (Intercept) 0.92
##  Subject            (Intercept) 0.75
##  Residual                       1.44
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1393.7, DIC = 1368
## deviance = 1365.1
```

```
anova(mem_music2)
```

```
## Analysis of Variance Table
##                     Df Sum Sq Mean Sq F value
## Harmony              3   34.7    11.6    5.54
## musician             1    0.2     0.2    0.10
## Instrument           2  115.9    58.0   27.82
## Harmony:musician     3   22.9     7.6    3.67
## musician:Instrument  2    2.5     1.3    0.61
```

Harmony may have a significant interaction with whether or not the subject is a musician. This could make sense: musicians would likely be more sensitive to changes in the harmony patterns than non-musicians.

```
mem_music4 = lmer(Classical ~ Harmony * musician + Instrument + (1 |
Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject), data = data4)

BIC(mem_best, mem_music1, mem_music2, mem_music4)
```

```
##            df  BIC
## mem_best   10 1438
## mem_music1 11 1443
## mem_music2 16 1456
## mem_music4 14 1448
```

```
display(mem_best)
```

```
## lmer(formula = Classical ~ Harmony + Instrument + (1 | Subject:Harmony) +
##     (1 | Subject:Instrument) + (1 | Subject), data = data4)
##                  coef.est coef.se
## (Intercept)       3.63     0.46
## HarmonyI-V-IV    -0.03     0.40
## HarmonyI-V-VI     1.06     0.40
## HarmonyIV-I-V    -0.21     0.40
## Instrumentpiano   1.97     0.44
## Instrumentstring  3.34     0.44
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.74
##  Subject:Instrument (Intercept) 0.89
##  Subject            (Intercept) 0.66
##  Residual                       1.44
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1399.5, DIC = 1376
## deviance = 1378.0
```

```
display(mem_music4)
```

```
## lmer(formula = Classical ~ Harmony * musician + Instrument +
##     (1 | Subject:Harmony) + (1 | Subject:Instrument) + (1 | Subject),
##     data = data4)
##                              coef.est coef.se
## (Intercept)                   3.81     0.60
## HarmonyI-V-IV                -0.09     0.49
## HarmonyI-V-VI                 0.08     0.49
## HarmonyIV-I-V                -0.29     0.49
## musicianTRUE                 -0.36     0.76
## Instrumentpiano               1.98     0.44
## Instrumentstring              3.34     0.44
## HarmonyI-V-IV:musicianTRUE  0.11     0.70
## HarmonyI-V-VI:musicianTRUE  1.97     0.70
## HarmonyIV-I-V:musicianTRUE  0.16     0.70
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.61
##  Subject:Instrument (Intercept) 0.89
##  Subject            (Intercept) 0.76
##  Residual                       1.44
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1393.9, DIC = 1367
## deviance = 1366.5
```

```
anova(mem_music4)
```

```
## Analysis of Variance Table
##                  Df Sum Sq Mean Sq F value
## Harmony           3   34.7    11.6    5.54
## musician          1    0.2     0.2    0.10
## Instrument        2  121.2    60.6   29.07
## Harmony:musician  3   22.9     7.6    3.67
```

Adding an interaction between whether the subject is a self-identifed musician or not appears to improve the model slightly, though the results are not conclusive. AIC and DIC are both lower for the model which includes “musician” as a fixed effect (AIC goes from 13.995 to 1393.9, DIC goes from 1376.5 to 1367.2) though BIC increases (from 1438.4 to 1448.3). We only have 10 observations here; it's possible that with more data we'd have more conclusive results.

If this interaction is significant, the results are very notable. Recall that the β s for Harmony were previously [0, -0.03, +1.06, -0.21]. In plain language, the only harmony type rated as more or less classical was "I-V-VI", which increased the rating of classical by a point. In the model with an interaction, the β s for Harmony for non-musicians is [0, -0.09, +0.08, -0.29]. In other words, non-musicians don't really take the change in harmonic structure into account when rating a song's classical-ness. For musicians, however, the β s for Harmony go to [-0.36, -0.34, +1.69, -0.49]. Musicians rate songs with an "I-V-VI" harmoic structure as 2.03 more "classical" than the next highest harmoic structure! Since the β s for insturment are [0, 1.98, 3.34], that means that harmonic structure is about as important to musicians as the insturment.

4.

a)

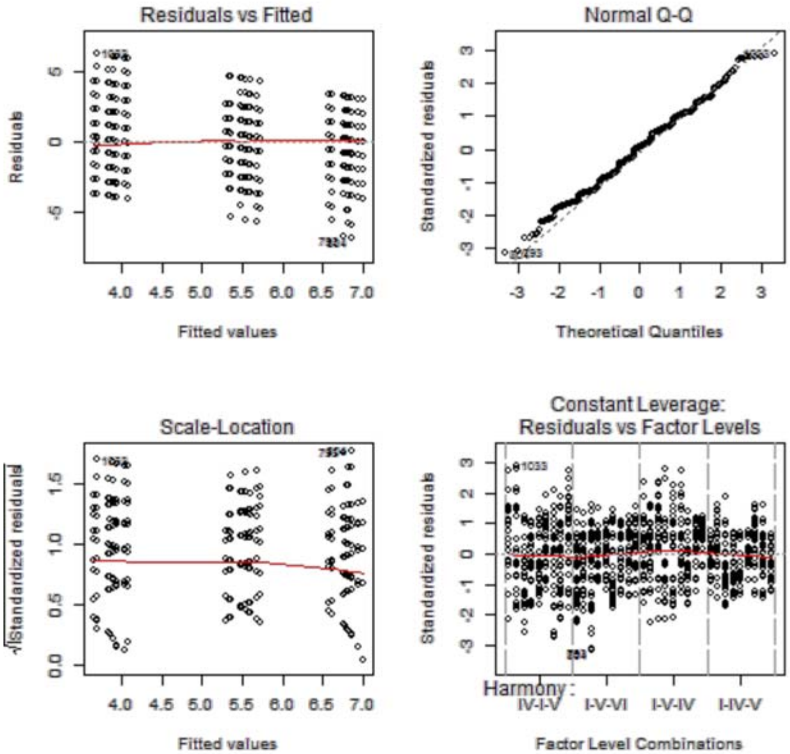
```
###### Popular

model1a_P = lm(Popular ~ Harmony + Instrument + Voice)
summary(model1a_P)
```

```
##
## Call:
## lm(formula = Popular ~ Harmony + Instrument + Voice)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.861 -1.655  0.148  1.430  6.306
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.751      0.187   36.07  < 2e-16 ***
## HarmonyI-V-IV      -0.133      0.187   -0.71     0.48
## HarmonyI-V-VI      -0.143      0.187   -0.76     0.45
## HarmonyIV-I-V      -0.181      0.187   -0.97     0.33
## Instrumentpiano    -1.267      0.162   -7.81  1.3e-14 ***
## Instrumentstring   -2.915      0.162  -17.98  < 2e-16 ***
## Voicepar3rd         0.196      0.162    1.21     0.23
## Voicepar5th         0.243      0.162    1.50     0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.17 on 1072 degrees of freedom
## Multiple R-squared:  0.235,  Adjusted R-squared:  0.23
## F-statistic:   47 on 7 and 1072 DF,  p-value: <2e-16
```

```
anova(model1a_P)
```

```
## Analysis of Variance Table
##
## Response: Popular
##              Df Sum Sq Mean Sq F value Pr(>F)
## Harmony       3      5       2    0.36   0.78
## Instrument    2   1539     769  162.62 <2e-16 ***
## Voice         2     12       6    1.26   0.28
## Residuals  1072   5071       5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow = c(2, 2))
plot(model1a_P)
```




```
par(mfrow = c(1, 1))
## We see the same trend in our residuals, but we've already explained this

###### Model Selection
model1a_P2 = lm(Popular ~ Instrument + Voice)
summary(model1a_P2)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument + Voice)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6.88  -1.64   0.12   1.39   6.28
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.637      0.148   44.89  < 2e-16 ***
## Instrumentpiano      -1.267      0.162   -7.82  1.2e-14 ***
## Instrumentstring     -2.915      0.162  -18.00  < 2e-16 ***
## Voicepar3rd           0.196      0.162    1.21     0.23
## Voicepar5th           0.243      0.162    1.50     0.13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.17 on 1075 degrees of freedom
## Multiple R-squared:  0.234,  Adjusted R-squared:  0.231
## F-statistic: 82.1 on 4 and 1075 DF,  p-value: <2e-16
```

```
anova(model1a_P2)
```

```
## Analysis of Variance Table
##
## Response: Popular
##              Df Sum Sq Mean Sq F value Pr(>F)
## Instrument    2   1539     769  162.92 <2e-16 ***
## Voice         2     12       6    1.27   0.28
## Residuals  1075   5076       5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# par(mfrow = c(2,2)) plot(model1a_P2) par(mfrow = c(1,1))

model1a_P3 = lm(Popular ~ Instrument)
summary(model1a_P3)
```

```
##
## Call:
## lm(formula = Popular ~ Instrument)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.783 -1.783  0.217  1.483  6.132
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.783      0.115   59.21  < 2e-16 ***
## Instrumentpiano    -1.267      0.162   -7.82  1.3e-14 ***
## Instrumentstring   -2.915      0.162  -17.99  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.17 on 1077 degrees of freedom
## Multiple R-squared:  0.232,  Adjusted R-squared:  0.231
## F-statistic:  163 on 2 and 1077 DF,  p-value: <2e-16
```

```
# par(mfrow = c(2,2)) plot(model1a_P3) par(mfrow = c(1,1))

AIC(model1a_P, model1a_P2, model1a_P3)
```

```
##            df  AIC
## model1a_P   9 4753
## model1a_P2  6 4748
## model1a_P3  4 4747
```

```
BIC(model1a_P, model1a_P2, model1a_P3)
```

```
##            df  BIC
## model1a_P   9 4798
## model1a_P2  6 4778
## model1a_P3  4 4767
```

Our best model for determining popular music was simply Popular ~ Insturment. This is consistent with the researchers expectations. In this case, both other predictor variables were not statistically significant, and the models which included them had higher AICs and BICs. For our best model, we found an $R^2 = 0.2308$, $p < 2.2*10^{-16}$, and both levels of our factor to be significant. In that model, we find the expected "Popular" rating for electric guitar music to be be 6.7833, the expected rating for piano music to be 5.5167, and the expected rating for string quartet music to be 3.8681.

Just for fun let's put everything back in when we add random effects.

plus, we must include the experimental factors in all models

```
mem1_P = lmer(Popular ~ Harmony + Instrument + Voice + (1 | Subject))
mem2_P = lmer(Popular ~ Harmony + Instrument + Voice + (1 | Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject:Voice))

AIC(mem2_P, mem1_P, model1a_P, model1a_P2, model1a_P3)
```

```
##             df  AIC
## mem2_P      12 4459
## mem1_P      10 4578
## model1a_P    9 4753
## model1a_P2   6 4748
## model1a_P3   4 4747
```

```
BIC(mem2_P, mem1_P, model1a_P, model1a_P2, model1a_P3)
```

```
##             df  BIC
## mem2_P      12 4519
## mem1_P      10 4627
## model1a_P    9 4798
## model1a_P2   6 4778
## model1a_P3   4 4767
```

The best model by far is mem2_P.

```
mem_best = mem2_P
mem3_P = lmer(Popular ~ Harmony + Instrument + (1 | Subject:Harmony) + (1 |
    Subject:Instrument))
mem4_P = lmer(Popular ~ Instrument + Voice + (1 | Subject:Instrument) + (1 |
    Subject:Voice))
mem5_P = lmer(Popular ~ Instrument + (1 | Subject:Instrument))

AIC(mem_best, mem3_P, mem4_P, mem5_P)
```

```
##          df  AIC
## mem_best 12 4459
## mem3_P    9 4457
## mem4_P    8 4490
## mem5_P    5 4486
```

```
BIC(mem_best, mem3_P, mem4_P, mem5_P)
```

```
##          df  BIC
## mem_best 12 4519
## mem3_P    9 4502
## mem4_P    8 4530
## mem5_P    5 4511
```

mem3_P is our front-runner.

```
mem_best = mem3_P
mem6_P = lmer(Popular ~ Harmony + Instrument + (1 | Subject:Harmony) + (1 |
    Subject:Instrument) + (1 | Subject))
mem7_P = lmer(Popular ~ Instrument + (1 | Subject:Harmony) + (1 |
Subject:Instrument) +
    (1 | Subject))
mem8_P = lmer(Popular ~ Instrument + (1 | Subject:Harmony) + (1 |
Subject:Instrument))
mem9_P = lmer(Popular ~ Instrument + (1 | Subject:Instrument) + (1 | Subject))

AIC(mem_best, mem6_P, mem7_P, mem8_P, mem9_P)
```

```
##           df  AIC
## mem_best  9 4457
## mem6_P   10 4450
## mem7_P    7 4440
## mem8_P    6 4448
## mem9_P    6 4474
```

```
BIC(mem_best, mem6_P, mem7_P, mem8_P, mem9_P)
```

```
##           df  BIC
## mem_best  9 4502
## mem6_P   10 4500
## mem7_P    7 4475
## mem8_P    6 4478
## mem9_P    6 4504
```

mem7_P is our winner, the simiar model that we landed on to measure classical, except it doesn't include Harmony as a fixed effect. The harmony/subject pairing remains intact however.

should include all experimental factors in the model. Also, usally incude fixed effects corresponding to each random effect

b)

```
mem_best = mem7_P
display(mem_best)
```

```
## lmer(formula = Popular ~ Instrument + (1 | Subject:Harmony) +
##     (1 | Subject:Instrument) + (1 | Subject))
##                  coef.est coef.se
## (Intercept)       6.78     0.25
## Instrumentpiano  -1.27     0.27
## Instrumentstring -2.92     0.27
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.64
##  Subject:Instrument (Intercept) 0.94
##  Subject            (Intercept) 0.83
##  Residual                       1.67
## ---
## number of obs: 1080, groups: Subject:Harmony, 120; Subject:Instrument, 90;
Subject, 30
## AIC = 4440.2, DIC = 4420
## deviance = 4422.9
```

```
anova(mem_best)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Instrument  2    323     161    57.7
```

Only one fixed effect remains in our model: the insturment which the song was played on. The estimated rating of “Popular” for a guitar song is 6.78. For piano songs, the estimated rating is 5.51. For songs played by a string quartet, the estimate is 3.86. Insturment is a highly significant predictor, since $\beta_{Insturment} = [-1.27, -2.92]$ while the the s.e. is only $\approx$ 0.27.

c)

some good ideas for exploring interactions with "musician" but the exploration is incomplete

```
mem_music0 = lmer(Popular ~ Instrument + (1 | Subject:Harmony) + (1 |
Subject:Instrument) +
    (1 | Subject), data = data4)
mem_music1 = lmer(Popular ~ Instrument + musician + (1 | Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject), data = data4)
mem_music2 = lmer(Popular ~ Harmony * musician + Instrument + (1 |
Subject:Harmony) +
    (1 | Subject:Instrument) + (1 | Subject), data = data4)

AIC(mem_music0, mem_music1, mem_music2)
```

```
##            df  AIC
## mem_music0  7 1416
## mem_music1  8 1416
## mem_music2 14 1420
```

```
BIC(mem_music0, mem_music1, mem_music2)
```

```
##            df  BIC
## mem_music0  7 1443
## mem_music1  8 1447
## mem_music2 14 1474
```

```


display(mem_music0)
```

```
## lmer(formula = Popular ~ Instrument + (1 | Subject:Harmony) +
##     (1 | Subject:Instrument) + (1 | Subject), data = data4)
##                  coef.est coef.se
## (Intercept)       6.85     0.43
## Instrumentpiano  -1.81     0.49
## Instrumentstring -3.15     0.49
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.61
##  Subject:Instrument (Intercept) 1.00
##  Subject            (Intercept) 0.75
##  Residual                       1.50
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1415.9, DIC = 1402
## deviance = 1401.8
```

```
display(mem_music1)
```

```
## lmer(formula = Popular ~ Instrument + musician + (1 | Subject:Harmony) +
##     (1 | Subject:Instrument) + (1 | Subject), data = data4)
##                   coef.est coef.se
## (Intercept)        7.11     0.55
## Instrumentpiano   -1.81     0.49
## Instrumentstring  -3.15     0.49
## musicianTRUE      -0.52     0.66
##
## Error terms:
##  Groups             Name        Std.Dev.
##  Subject:Harmony    (Intercept) 0.61
##  Subject:Instrument (Intercept) 1.00
##  Subject            (Intercept) 0.78
##  Residual                       1.50
## ---
## number of obs: 360, groups: Subject:Harmony, 40; Subject:Instrument, 30;
Subject, 10
## AIC = 1416.3, DIC = 1402
## deviance = 1401.1
```

```
anova(mem_music1)
```

```
## Analysis of Variance Table
##            Df Sum Sq Mean Sq F value
## Instrument  2   94.8    47.4   21.12
## musician    1    1.4     1.4    0.62
```

“Musician” is not a significant predictor of popular songs. The model without musician has the lowest AIC, BIC, and DIC. In the model which contains “musician,” the predictor's standard error (0.66) has a larger magnitude than the beta (-0.52).

5.

Rating Music as Classical or Popular

Josh Jelin

4: 16
5: 20

36

Executive Summary

Dr. Ivan Jimenez and a student, Vincent Rossi, 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". Our data analysis found that instrument played in a song was the main driver in a person's rating of that song as either "classical" ($p = 9.53 * 10^{-8}$) or "popular" ($p < 2*10^{-16}$). Additionally, music listeners who described themselves as "musicians" took the harmonic motion of a song into account when rating a song as classical or not classical ($p = 0.0199$).

Introduction

The experimenters recruited 70 listeners, undergraduates from the University of Pittsburgh, to rate the music on two scales:

- How classical does the music sound (1 to 10, 1 = not at all, 10 = very classical sounding).
- How popular does the music sound (1 to 10, 1 = not at all, 10 = very popular sounding).

The 36 stimuli were chosen by completely crossing these factors:

Inurement	Harmonic Motion	Voice Leading
String Quartet, Piano, Electric Guitar	I-V-VI, I-VI-V, I-V-IV, IV-I-V	Contrary Motion, Parallel 3rds, Parallel 5ths

Methods

Unfortunately, 65 of the 70 listeners failed to fully complete the study. By correcting some obvious unintentional blanks, we were able to construct a complete data set of 30 music listeners to perform analysis upon.

We conducted our analysis in R, using the lme4 package to produce a linear mixed-effects model. For both classical and popular music, each listener had a noticeable difference in how a song's harmonic motion and instrument influenced their ratings. To account for listener variation, we included harmonic motion by listener and instrument by listener as random effects in both models.

Results

For the ratings of popular music, the instrument being played was the only significant predictor variable ($p < 2*10^{-16}$). The expected rating of "Popular" for a guitar song was 6.78. For piano songs, the expected rating was 5.51. For songs played by a string quartet, the expected rating was 3.86.

The relationship is more complicated in the ratings of classical music. Again, the instrument being played was the most significant predictor variable ($p = 9.53 * 10^{-8}$). For a non-musician, the expected rating of "Classical" for a guitar song was approximately 3.7. For piano songs, the expected rating was approximately 5.7. For songs played by a string quartet, the expected rating was approximately 7.1. These estimated varied slightly for non-musicians depending on the harmonic motion of the song. However, listeners who described themselves as musicians rated songs significantly differently depending on the song's harmonic motion ($p = 0.0199$). Specifically, musicians rated songs with harmonic motion "`I-V-VI`" as 2.12 points more classical versus other types of harmonic motion ($p = 0.0016$).

Conclusion

As predicted by our researchers, instrument has the largest influence on rating for both classical ($p = 9.53*10^{-8}$) and popular ($p < 2*10^{-16}$) music. Our researchers also correctly predicted that the harmonic progression "`I-V-VI`" was frequently rated as classical. However, that change in rating was only significant for listeners who identified themselves as musicians ($p = 0.0016$). It's noteworthy that the voice leading in the song had no significant relationship with the rating of "classical" or "popular" once the other factors had been taken into account by our model.