

# Hearing Threshold Data

Hao Lu

July 18, 2018

## 1. Load and format data from xls file

```
# load data into R
data = read.csv("H:/AnooR/all_data_forR.csv",header = TRUE)

# Rename variables
names(data)[1] <- "Phase"

# set variable as factor
data$Phase %<>% as.factor()
data$F0 %<>% as.factor()
data$HarmonicNum %<>% as.factor()
data$Subject %<>% as.factor()

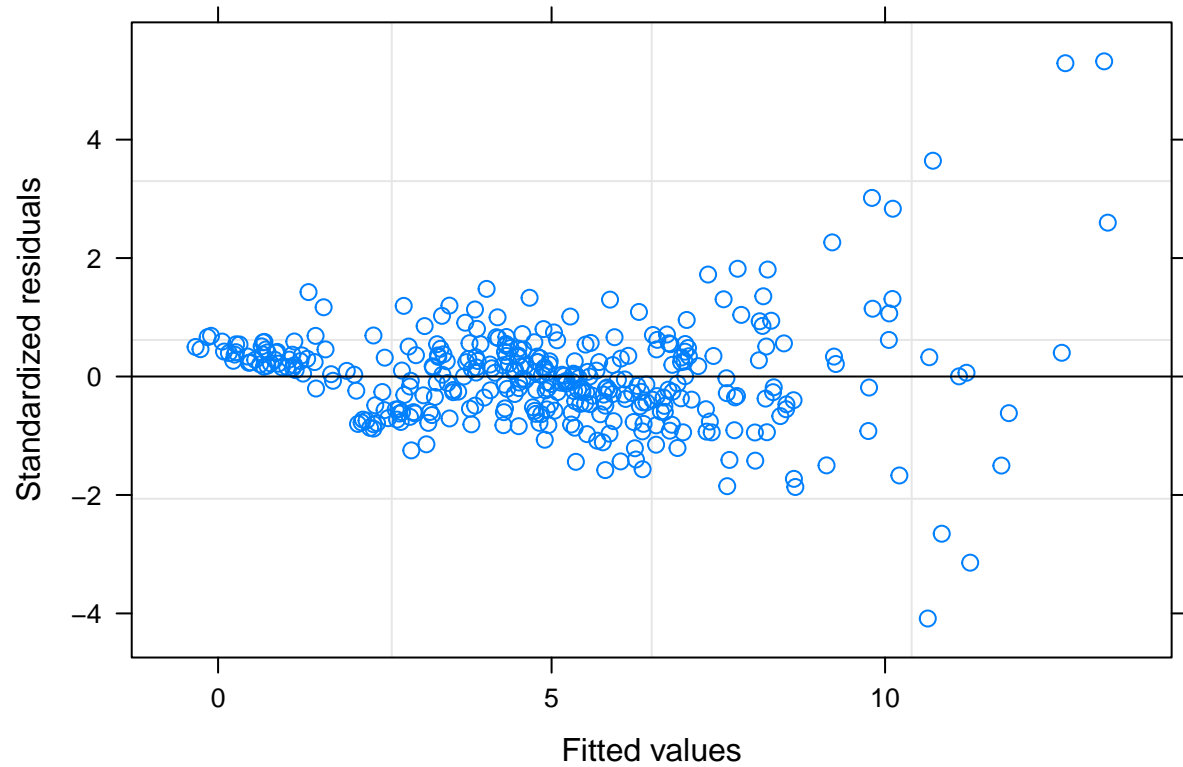
# Show the loaded data
summary(data)
```

```
## Phase      F0      HarmonicNum Subject  Thresholds
## 0:180    30 :60    3 :72         1:60    Min.     : 0.5158
## 1:180    40 :60    6 :72         2:60    1st Qu.: 3.0652
##          50 :60    9 :72         3:60    Median  : 4.8697
##          70 :60   12:72         4:60    Mean     : 4.9712
##          100:60   15:72         5:60    3rd Qu.: 6.0867
##          200:60           6:60    Max.     :22.3872
```

Four predictors (Phase, F0, Harmonic number, Subject code) were set as categorical. Response (Thresholds) remained numeric

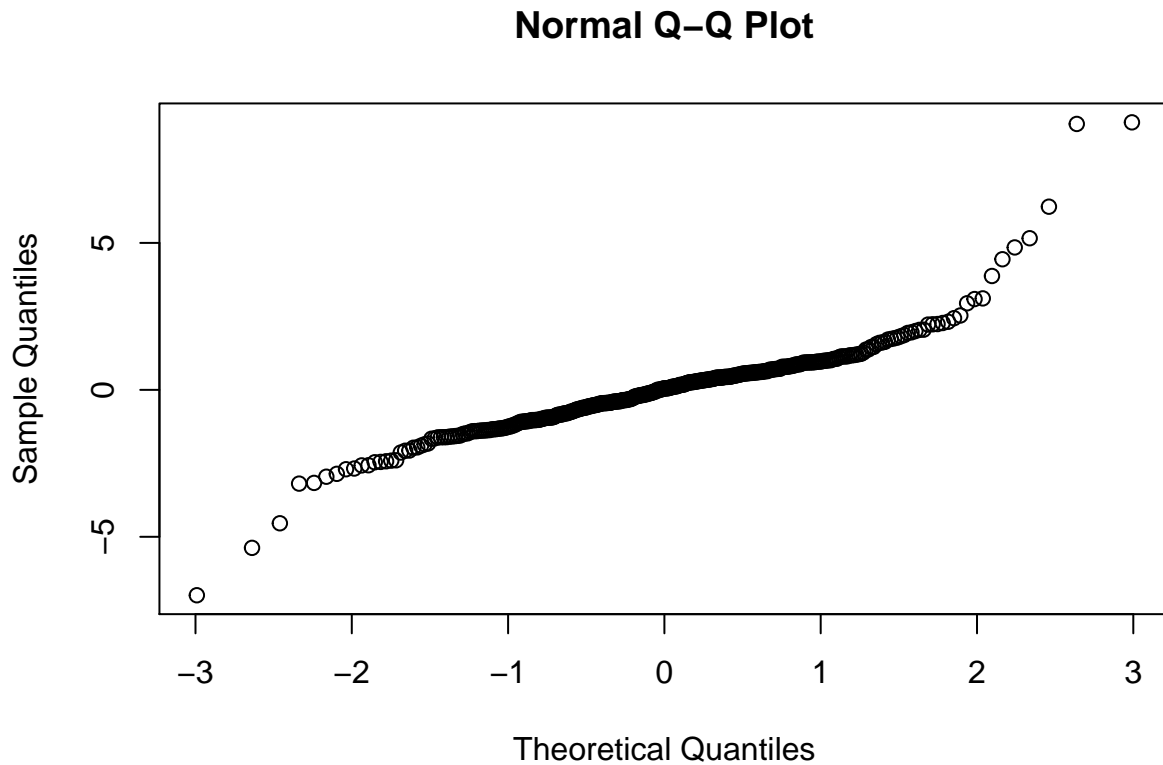
## 2. Check model assumptions

```
# starting from a full model with all possible effects. The error correlation structure was set to be A
m1 <- lme(Thresholds ~ Phase*F0*HarmonicNum, data = data, random = ~1|Subject,correlation = corAR1())
plot(m1)
```



The plot showed that variance residual increases with fitted values, so the error and fitted values are not independent.

```
qqnorm(residuals(m1))
```

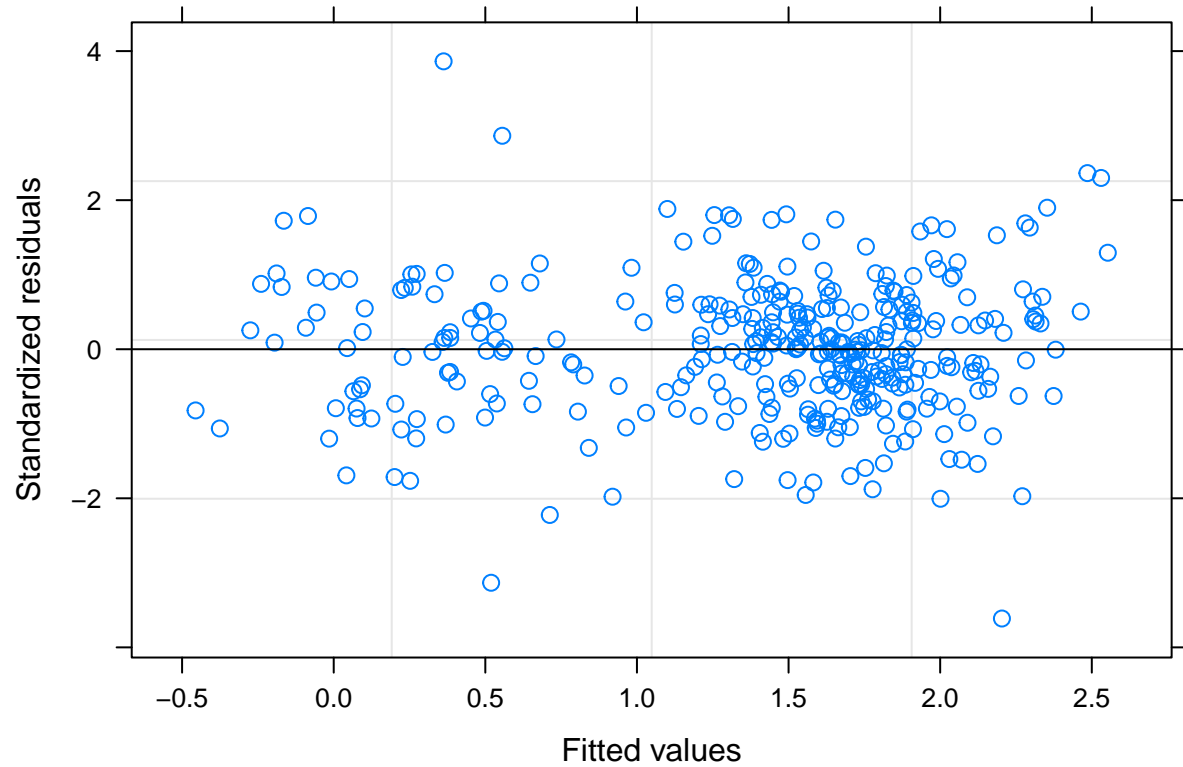


The QQ plot showed that the residuals had heavy tails, which also violate the assumption of normal distributed error.

It has been shown that the original threshold could not be fitted with a repeated measurement linear model, so we consider doing a log transform on the thresholds.

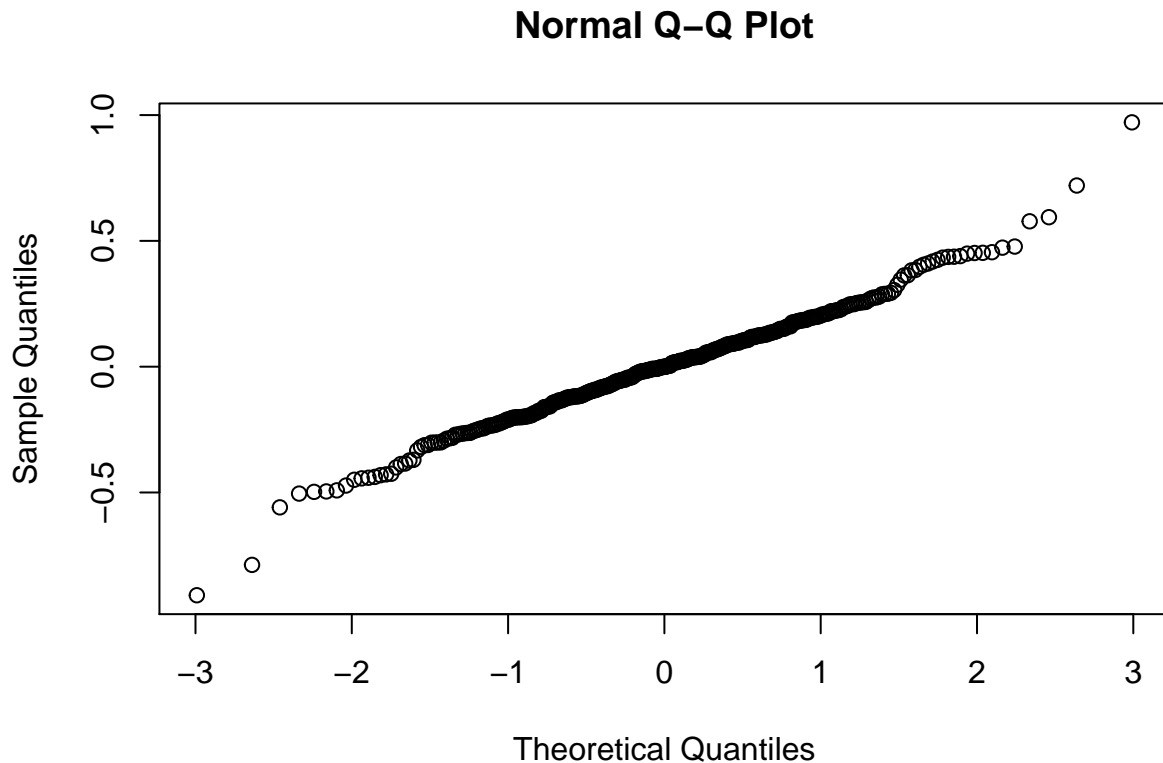
```
# log transform
data$logThre <- log(data$Threshold)

# fit a new model with the transformed data
m2 <- lme(logThre ~ Phase*F0*HarmonicNum, data = data, random = ~1|Subject, correlation = corAR1())
plot(m2)
```



This time it looks much better, the variance of residual doesn't change with fitted values.

```
qqnorm(residuals(m2))
```



There are still heavy tails, but at least better than the original data.

All following analyzation will use the transformed threshold data as response.

### 3. Model selection

Now we need to find a best model to fit our data.

For this dataset, the only random effect is subject.

First, let's see if we can drop all effects include phase.

```
# Fit the nested model to check if Phase could be dropped
rm1 <- lme(logThre ~ Phase*F0*HarmonicNum, data = data, method = "ML", random = ~1|Subject, correlation = cor)
rm2 <- lme(logThre ~ F0*HarmonicNum, data = data, method = "ML", random = ~1|Subject, correlation = cor)

# Because we are testing a fixed effect between nested models, so the default model fitting method "REML" is not appropriate
anova(rm1,rm2)
```

	##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	rm1		1 63	98.73716	343.5617	13.63142			
##	rm2		2 33	94.10806	222.3495	-14.05403	1 vs 2	55.37091	0.0032

Although dropping these effects related with phase improved both AIC and BIC, but our model was significantly worse at fitting data. Perhaps not all effects should be dropped.

I usually tend to be conservative with this so I am not going to drop all effects.

```
# test all fixed effects
anova(rm1)
```

```
##               numDF denDF  F-value p-value
## (Intercept)         1   295 364.0981 <.0001
## Phase               1   295   1.6420 0.2011
## F0                  5   295 147.3480 <.0001
## HarmonicNum         4   295 208.1708 <.0001
## Phase:F0            5   295   3.8539 0.0021
## Phase:HarmonicNum   4   295   4.2168 0.0025
## F0:HarmonicNum     20   295  26.0090 <.0001
## Phase:F0:HarmonicNum 20   295   0.9700 0.4989
```

It can be seen that the two-way interaction of phase and F0 is still significant, so it should not be dropped. Therefore, we modify the model to keep the interaction of phase and F0 but drop the main effect of phase and the three-way interaction.

```
rm3 <- lme(logThre ~ F0*HarmonicNum + Phase:F0 + Phase:HarmonicNum,data = data, method = "ML", random = 
anova(rm1,rm3)
```

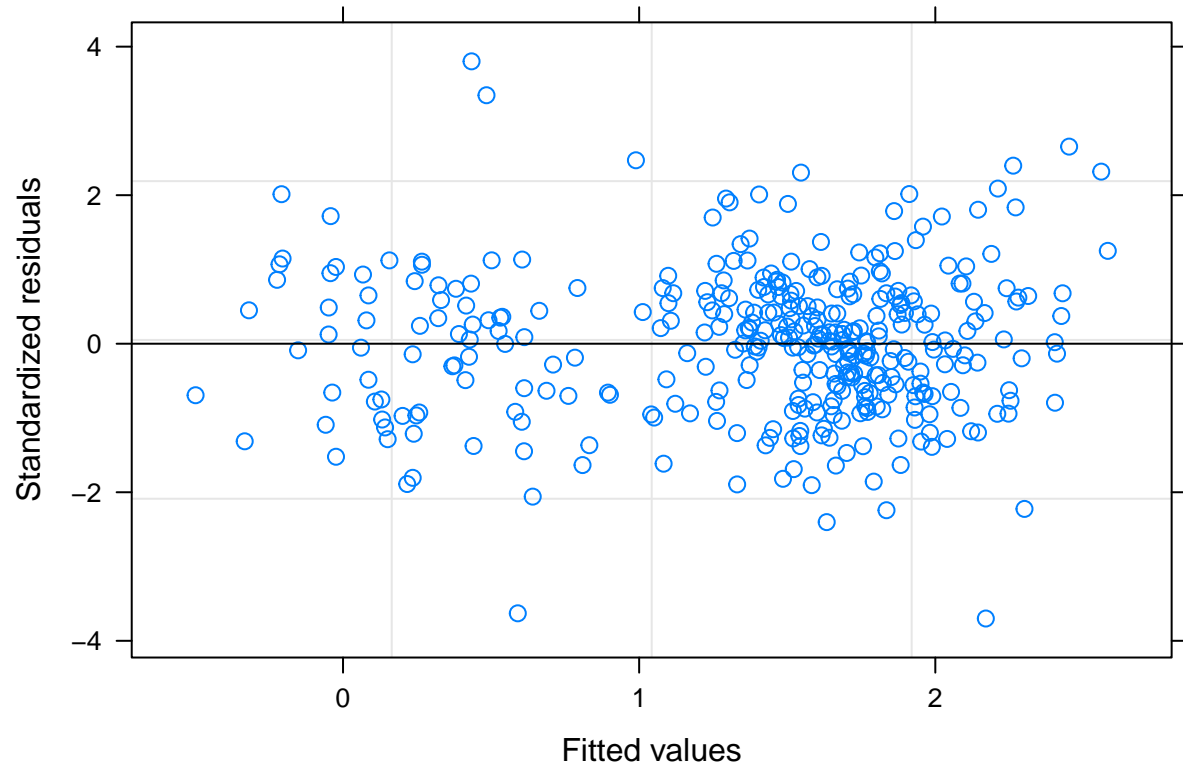
```
##      Model df      AIC      BIC    logLik    Test  L.Ratio p-value
## rm1      1 63 98.73716 343.5617 13.631422
## rm3      2 43 81.25534 248.3578  2.372331 1 vs 2 22.51818  0.3131
```

This result showed that this simplified model is as good as the full model in fitting the data. The new model also improved AIC and BIC because it has less predictors.

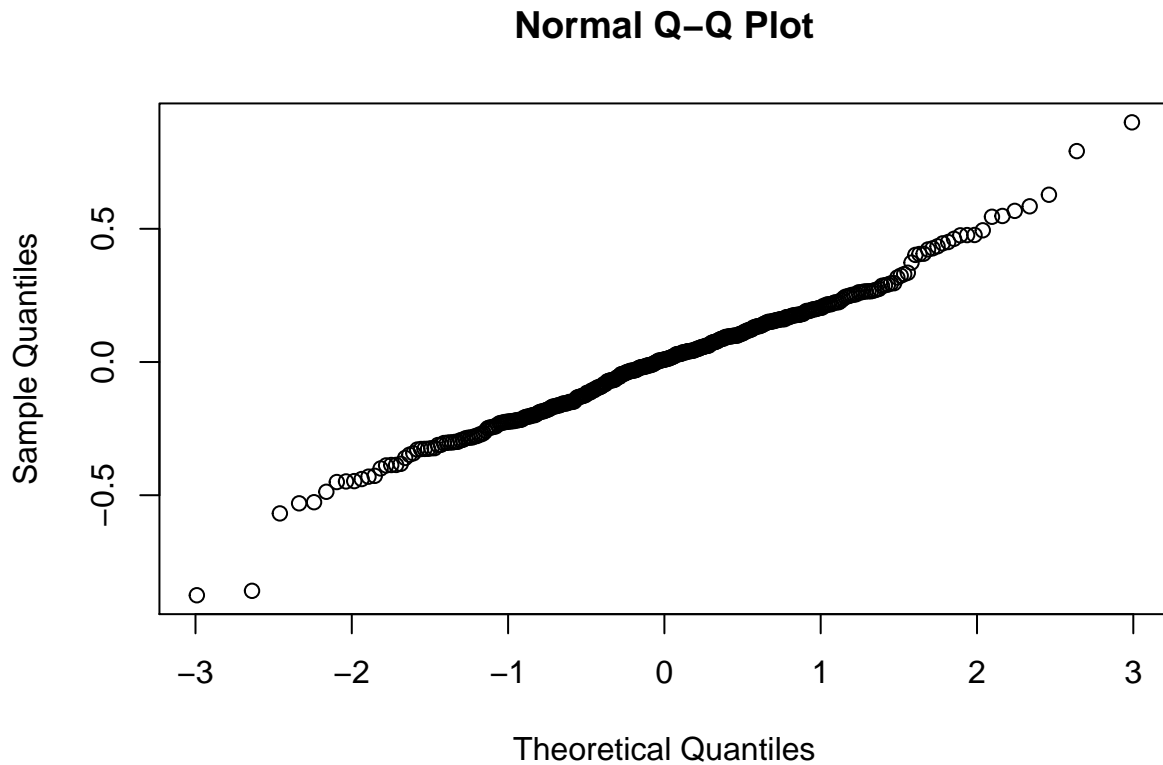
This should be the model we are looking for.

Check the error just in case.

```
plot(rm3)
```



```
qqnorm(residuals(rm3))
```



Everything about model assumption looks good. I would choose the simpler model.

Similarly, you can test different error correlation structures, I am just too lazy to do it here.

## 4. Post hot tests

Now we are going to show the pairwise test result with the model we chose.

```
emmeans(rm3, pairwise ~ HarmonicNum|F0)$contrasts
```

```
## NOTE: A nesting structure was detected in the fitted model:
##   Phase %in% F0
## If this is incorrect, re-run or update with `nesting` specified

## F0 = 30:
##   contrast      estimate      SE   df t.ratio p.value
## 3 - 6      0.172920603 0.09491609 315   1.822  0.3628
## 3 - 9      0.224695898 0.10135420 315   2.217  0.1761
## 3 - 12     0.295825094 0.10222593 315   2.894  0.0329
## 3 - 15     0.583644427 0.10234778 315   5.703 <.0001
## 6 - 9      0.051775295 0.09497821 315   0.545  0.9825
## 6 - 12     0.122904491 0.10142987 315   1.212  0.7446
## 6 - 15     0.410723823 0.10230357 315   4.015  0.0007
## 9 - 12     0.071129196 0.09497944 315   0.749  0.9447
## 9 - 15     0.358948529 0.10143136 315   3.539  0.0042
## 12 - 15    0.287819333 0.09497946 315   3.030  0.0221
```



```

##
## F0 = 40:
## contrast      estimate      SE  df t.ratio p.value
## 3 - 6      -0.586039962 0.09497946 315 -6.170 <.0001
## 3 - 9      -0.750738767 0.10143139 315 -7.401 <.0001
## 3 - 12     -0.617220813 0.10230513 315 -6.033 <.0001
## 3 - 15     -0.491697946 0.10242727 315 -4.800 <.0001
## 6 - 9      -0.164698805 0.09497946 315 -1.734 0.4146
## 6 - 12     -0.031180852 0.10143139 315 -0.307 0.9981
## 6 - 15      0.094342016 0.10230513 315  0.922 0.8882
## 9 - 12      0.133517954 0.09497946 315  1.406 0.6244
## 9 - 15      0.259040821 0.10143139 315  2.554 0.0817
## 12 - 15     0.125522868 0.09497946 315  1.322 0.6780
##
## F0 = 50:
## contrast      estimate      SE  df t.ratio p.value
## 3 - 6      -0.992989821 0.09497946 315 -10.455 <.0001
## 3 - 9      -1.138500407 0.10143139 315 -11.224 <.0001
## 3 - 12     -1.112916128 0.10230513 315 -10.878 <.0001
## 3 - 15     -1.109718093 0.10242727 315 -10.834 <.0001
## 6 - 9      -0.145510586 0.09497946 315 -1.532 0.5424
## 6 - 12     -0.119926307 0.10143139 315 -1.182 0.7615
## 6 - 15     -0.116728272 0.10230513 315 -1.141 0.7846
## 9 - 12      0.025584279 0.09497946 315  0.269 0.9988
## 9 - 15      0.028782314 0.10143139 315  0.284 0.9986
## 12 - 15     0.003198035 0.09497946 315  0.034 1.0000
##
## F0 = 70:
## contrast      estimate      SE  df t.ratio p.value
## 3 - 6      -0.650800092 0.09497946 315 -6.852 <.0001
## 3 - 9      -1.167282721 0.10143139 315 -11.508 <.0001
## 3 - 12     -1.306397237 0.10230513 315 -12.770 <.0001
## 3 - 15     -1.216852261 0.10242727 315 -11.880 <.0001
## 6 - 9      -0.516482628 0.09497946 315 -5.438 <.0001
## 6 - 12     -0.655597145 0.10143139 315 -6.463 <.0001
## 6 - 15     -0.566052169 0.10230513 315 -5.533 <.0001
## 9 - 12     -0.139114516 0.09497946 315 -1.465 0.5862
## 9 - 15     -0.049569540 0.10143139 315 -0.489 0.9884
## 12 - 15     0.089544976 0.09497946 315  0.943 0.8799
##
## F0 = 100:
## contrast      estimate      SE  df t.ratio p.value
## 3 - 6      -0.389360743 0.09497946 315 -4.099 0.0005
## 3 - 9      -1.397541228 0.10143139 315 -13.778 <.0001
## 3 - 12     -1.451108298 0.10230513 315 -14.184 <.0001
## 3 - 15     -1.395942097 0.10242727 315 -13.629 <.0001
## 6 - 9      -1.008180485 0.09497946 315 -10.615 <.0001
## 6 - 12     -1.061747555 0.10143139 315 -10.468 <.0001
## 6 - 15     -1.006581354 0.10230513 315 -9.839 <.0001
## 9 - 12     -0.053567070 0.09497946 315 -0.564 0.9801
## 9 - 15      0.001599131 0.10143139 315  0.016 1.0000
## 12 - 15     0.055166201 0.09497946 315  0.581 0.9778
##
## F0 = 200:

```

```
## contrast      estimate      SE  df t.ratio p.value
## 3 - 6      -0.433808392 0.09497946 315  -4.567  0.0001
## 3 - 9      -1.666934781 0.10143136 315 -16.434  <.0001
## 3 - 12     -1.735437329 0.10230357 315 -16.964  <.0001
## 3 - 15     -1.708036335 0.10234778 315 -16.689  <.0001
## 6 - 9      -1.233126390 0.09497944 315 -12.983  <.0001
## 6 - 12     -1.301628938 0.10142987 315 -12.833  <.0001
## 6 - 15     -1.274227943 0.10222593 315 -12.465  <.0001
## 9 - 12     -0.068502548 0.09497821 315  -0.721  0.9515
## 9 - 15     -0.041101554 0.10135420 315  -0.406  0.9943
## 12 - 15     0.027400994 0.09491609 315   0.289  0.9985
##
## Results are averaged over the levels of: Phase
## P value adjustment: tukey method for comparing a family of 5 estimates
```

The output above showed all pairs of harmonic numbers within a level of F0. One can easily see if there is significant difference between harmonic numbers through p value. The result was adjusted with tukey method to correct multiple comparison error.