

Clre within analysis

Load packages

```
pacman::p_load(readr, lme4, emmeans)
```

Load data

```
long = read_csv("long.csv")
```

```
## Rows: 288 Columns: 14
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (5): Time, Deviant, Listeninghabits, Channel, Group
```

```
## dbl (9): ID, DeviantLevel, AvCor, RoQ, M_knowledge, M_enjoyment, YearsWOHear...
```

```
##
```

```
## i Use 'spec()' to retrieve the full column specification for this data.
```

```
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
long$Deviant=as.factor(long$Deviant)
```

```
long$DeviantLevel=as.factor(long$DeviantLevel)
```

```
long$Time = as.factor(long$Time)
```

```
long$ID=as.factor(long$ID)
```

```
long$M_knowledge = ordered(long$M_knowledge, levels = c("1","2","3","4","5"))
```

```
long$M_enjoyment=ordered(long$M_enjoyment, levels = c("1", "2", "3", "4", "5", "6", "7"))
```

```
long$Listeninghabits=ordered(long$Listeninghabits, levels = c("Low", "High"))
```

```
change_all = read_csv("change_all.csv")
```

```
## Rows: 144 Columns: 30
```

```
## -- Column specification -----
```

```
## Delimiter: ","
```

```
## chr (3): Deviant, Channel, Group
```

```
## dbl (25): ID, DeviantLevel, AvCor.T1, RoQ.T1, Listeninghabits.T1, M_knowledg...
```

```
## lgl (2): M_knowledge.T2, YearsWOHearing.T2
```

```
##
```

```
## i Use 'spec()' to retrieve the full column specification for this data.
```

```
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
change_all$Deviant=as.factor(change_all$Deviant)
change_all$DeviantLevel=as.factor(change_all$DeviantLevel)
change_all$ID=as.factor(change_all$ID)
```

Hierarchical mixed effects modeling - MMN Amplitude

```
M0=lmer(Amplitude~1+(1|ID), data=long, REML = FALSE, control = lmerControl(optimizer = "nloptwrap", cal
M1=lmer(Amplitude~Deviant+(1|ID), data=long, REML = FALSE, control = lmerControl(optimizer = "nloptwrap
M2=lmer(Amplitude~Deviant+DeviantLevel+(1|ID), data=long, REML = FALSE, control = lmerControl(optimizer
M3=lmer(Amplitude~Deviant+DeviantLevel+Time+(1|ID), data=long, REML = FALSE, control = lmerControl(optim
M4=lmer(Amplitude~Deviant*DeviantLevel+Time+(1|ID), data=long, REML = FALSE, control = lmerControl(optim
M5=lmer(Amplitude~Deviant*DeviantLevel+Deviant*Time+(1|ID), data=long, REML = FALSE, control = lmerCont
M6=lmer(Amplitude~Deviant*DeviantLevel+Deviant*Time+DeviantLevel*Time+(1|ID), data=long, REML = FALSE, c
anova(M0,M1,M2,M3,M4,M5,M6)
```

```
## Data: long
## Models:
## M0: Amplitude ~ 1 + (1 | ID)
## M1: Amplitude ~ Deviant + (1 | ID)
## M2: Amplitude ~ Deviant + DeviantLevel + (1 | ID)
## M3: Amplitude ~ Deviant + DeviantLevel + Time + (1 | ID)
## M4: Amplitude ~ Deviant * DeviantLevel + Time + (1 | ID)
## M5: Amplitude ~ Deviant * DeviantLevel + Deviant * Time + (1 | ID)
## M6: Amplitude ~ Deviant * DeviantLevel + Deviant * Time + DeviantLevel * Time + (1 | ID)
##      npar    AIC    BIC logLik deviance   Chisq Df Pr(>Chisq)
## M0      3 205.92 216.91 -99.959   199.92
## M1      6 201.62 223.59 -94.809   189.62 10.3009  3    0.01617 *
## M2      9 204.92 237.89 -93.462   186.92  2.6936  3    0.44131
## M3     10 202.99 239.62 -91.496   182.99  3.9316  1    0.04739 *
## M4     19 205.52 275.11 -83.759   167.52 15.4746  9    0.07870 .
## M5     22 201.39 281.98 -78.695   157.39 10.1274  3    0.01751 *
## M6     25 202.42 293.99 -76.210   152.42  4.9703  3    0.17398
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best model

```
summary(M5)
```

```
## Linear mixed model fit by maximum likelihood ['lmerMod']
## Formula: Amplitude ~ Deviant * DeviantLevel + Deviant * Time + (1 | ID)
```

```

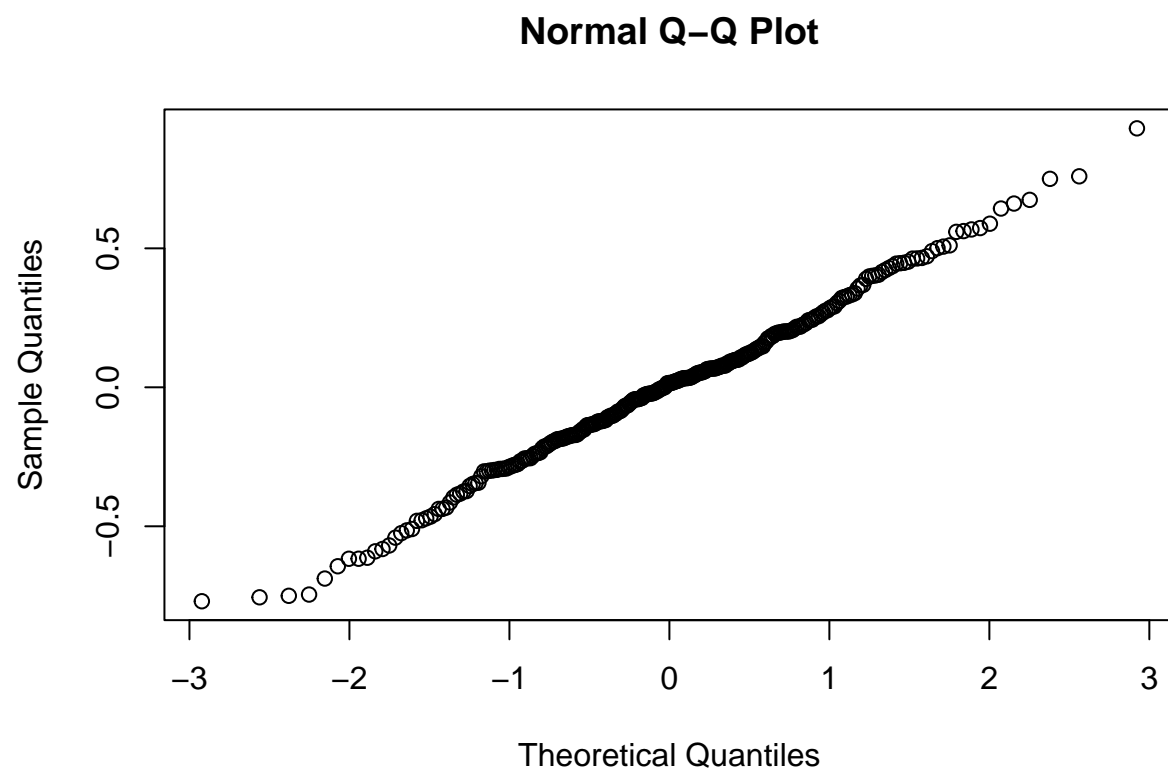
## Data: long
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC    logLik deviance df.resid
##    201.4    282.0    -78.7    157.4      266
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.54980 -0.61091  0.05113  0.64438  3.08648
##
## Random effects:
##  Groups   Name                Variance Std.Dev.
##  ID       (Intercept)  0.07706   0.2776
##  Residual                  0.09112   0.3019
## Number of obs: 288, groups: ID, 9
##
## Fixed effects:
##
##              Estimate Std. Error t value
## (Intercept)    -0.11792    0.12203  -0.966
## DeviantPitch    -0.06006    0.11250  -0.534
## DeviantRhythm     0.20109    0.11250   1.788
## DeviantTimbre    -0.11935    0.11250  -1.061
## DeviantLevel2    -0.08132    0.10062  -0.808
## DeviantLevel3    -0.06494    0.10062  -0.645
## DeviantLevel4    -0.22831    0.10062  -2.269
## TimeT2           0.11753    0.07115   1.652
## DeviantPitch:DeviantLevel2  0.16416    0.14230   1.154
## DeviantRhythm:DeviantLevel2 -0.14753    0.14230  -1.037
## DeviantTimbre:DeviantLevel2  0.19388    0.14230   1.362
## DeviantPitch:DeviantLevel3  0.10296    0.14230   0.724
## DeviantRhythm:DeviantLevel3 -0.24879    0.14230  -1.748
## DeviantTimbre:DeviantLevel3  0.10916    0.14230   0.767
## DeviantPitch:DeviantLevel4  0.28083    0.14230   1.974
## DeviantRhythm:DeviantLevel4  0.13090    0.14230   0.920
## DeviantTimbre:DeviantLevel4  0.21730    0.14230   1.527
## DeviantPitch:TimeT2    -0.27858    0.10062  -2.769
## DeviantRhythm:TimeT2    -0.20903    0.10062  -2.077
## DeviantTimbre:TimeT2    -0.27894    0.10062  -2.772
##
##
## Correlation matrix not shown by default, as p = 20 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it

```

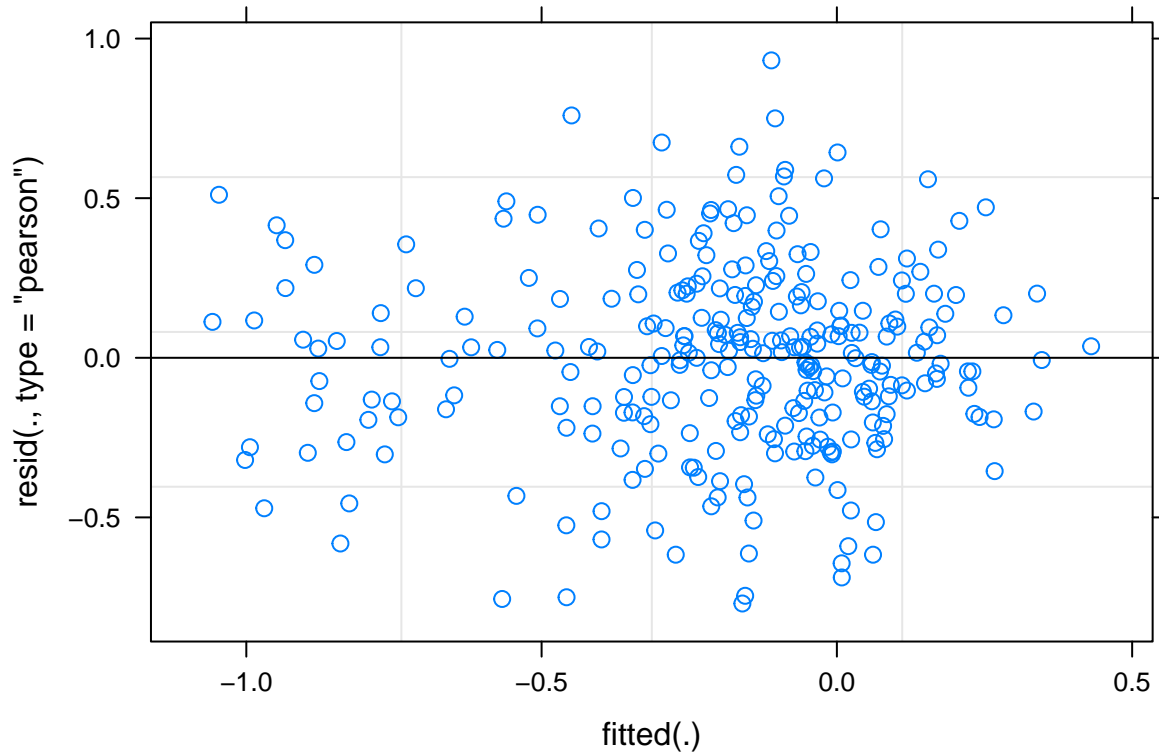
```

#Checking assumptions with diagnostic plots
qqnorm(resid(M5))

```



```
plot(M5)
```



Post-hoc analysis

```
emm_r <- emmeans(M5, pairwise ~ Time)
```

NOTE: Results may be misleading due to involvement in interactions

```
emm_rc <- emm_r[[2]]
```

```
emm_d <- emmeans(M5, pairwise ~ Deviant)
```

NOTE: Results may be misleading due to involvement in interactions

```
emm_dc <- emm_d[[2]]
```

#Time by deviant interaction

```
emm_r.d <- emmeans(M5, pairwise ~ Time | Deviant)
```

```
IC_r.d <- contrast(emm_r.d[[1]], method = "pairwise")
```

```
emm_d.r <- emmeans(M5, pairwise ~ Deviant | Time)
```

```
IC_d.r <- contrast(emm_d.r[[1]], method = "pairwise")
```

```
rbind(emm_rc, emm_dc, IC_r.d[1:4], IC_d.r[1:12], adjust = "mvt")
```

```
## contrast      Deviant    Time estimate      SE  df t.ratio p.value
## T1 - T2        .          .         0.0741 0.0369 299   2.011  0.4425
## Intensity - Pitch .          .         0.0624 0.0521 299   1.197  0.9223
## Intensity - Rhythm .          .        -0.0302 0.0521 299  -0.580  0.9989
## Intensity - Timbre .          .         0.1287 0.0521 299   2.470  0.1893
## Pitch - Rhythm .          .        -0.0926 0.0521 299  -1.777  0.6048
## Pitch - Timbre .          .         0.0664 0.0521 299   1.274  0.8947
## Rhythm - Timbre .          .         0.1590 0.0521 299   3.050  0.0434
## T1 - T2        Intensity .        -0.1175 0.0737 299  -1.595  0.7269
## T1 - T2        Pitch      .         0.1611 0.0737 299   2.185  0.3316
## T1 - T2        Rhythm     .         0.0915 0.0737 299   1.241  0.9068
## T1 - T2        Timbre     .         0.1614 0.0737 299   2.190  0.3292
## Intensity - Pitch .          T1        -0.0769 0.0737 299  -1.044  0.9617
## Intensity - Rhythm .          T1        -0.1347 0.0737 299  -1.828  0.5683
## Intensity - Timbre .          T1        -0.0107 0.0737 299  -0.146  1.0000
## Pitch - Rhythm .          T1        -0.0578 0.0737 299  -0.784  0.9926
## Pitch - Timbre .          T1         0.0662 0.0737 299   0.898  0.9835
## Rhythm - Timbre .          T1         0.1240 0.0737 299   1.683  0.6698
## Intensity - Pitch .          T2         0.2017 0.0737 299   2.736  0.1013
## Intensity - Rhythm .          T2         0.0743 0.0737 299   1.008  0.9682
## Intensity - Timbre .          T2         0.2682 0.0737 299   3.639  0.0065
## Pitch - Rhythm .          T2        -0.1274 0.0737 299  -1.728  0.6380
## Pitch - Timbre .          T2         0.0666 0.0737 299   0.903  0.9830
## Rhythm - Timbre .          T2         0.1939 0.0737 299   2.631  0.1301
##
## Results are averaged over some or all of the levels of: Deviant, DeviantLevel, Time
## Degrees-of-freedom method: kenward-roger
## P value adjustment: mvt method for 23 tests
```

```
rbind(IC_r.d[1:4], adjust = "mvt")
```

```
## contrast Deviant    estimate      SE  df t.ratio p.value
## T1 - T2 Intensity -0.1175 0.0737 299  -1.595  0.3768
## T1 - T2 Pitch      0.1611 0.0737 299   2.185  0.1132
## T1 - T2 Rhythm     0.0915 0.0737 299   1.241  0.6197
## T1 - T2 Timbre     0.1614 0.0737 299   2.190  0.1119
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: mvt method for 4 tests
```

```
rbind(IC_r.d[1:4], adjust = "none")
```

```
## contrast Deviant    estimate      SE  df t.ratio p.value
## T1 - T2 Intensity -0.1175 0.0737 299  -1.595  0.1119
## T1 - T2 Pitch      0.1611 0.0737 299   2.185  0.0297
## T1 - T2 Rhythm     0.0915 0.0737 299   1.241  0.2154
## T1 - T2 Timbre     0.1614 0.0737 299   2.190  0.0293
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
```

```
rbind(IC_r.d[c(1,2,4)], adjust = "none")
```

```
## contrast Deviant estimate SE df t.ratio p.value
## T1 - T2 Intensity -0.118 0.0737 299 -1.595 0.1119
## T1 - T2 Pitch 0.161 0.0737 299 2.185 0.0297
## T1 - T2 Timbre 0.161 0.0737 299 2.190 0.0293
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
```

```
IC_r.d[1:4]
```

```
## Note: adjust = "tukey" was changed to "sidak"
## because "tukey" is only appropriate for one set of pairwise comparisons
```

```
## contrast Deviant estimate SE df t.ratio p.value
## T1 - T2 Intensity -0.1175 0.0737 299 -1.595 0.3778
## T1 - T2 Pitch 0.1611 0.0737 299 2.185 0.1134
## T1 - T2 Rhythm 0.0915 0.0737 299 1.241 0.6210
## T1 - T2 Timbre 0.1614 0.0737 299 2.190 0.1121
##
## Results are averaged over the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 4 tests
```

```
IC_d.r[1:12]
```

```
## Note: adjust = "tukey" was changed to "sidak"
## because "tukey" is only appropriate for one set of pairwise comparisons
```

```
## contrast Time estimate SE df t.ratio p.value
## Intensity - Pitch T1 -0.0769 0.0737 299 -1.044 0.9855
## Intensity - Rhythm T1 -0.1347 0.0737 299 -1.828 0.5734
## Intensity - Timbre T1 -0.0107 0.0737 299 -0.146 1.0000
## Pitch - Rhythm T1 -0.0578 0.0737 299 -0.784 0.9989
## Pitch - Timbre T1 0.0662 0.0737 299 0.898 0.9961
## Rhythm - Timbre T1 0.1240 0.0737 299 1.683 0.6921
## Intensity - Pitch T2 0.2017 0.0737 299 2.736 0.0763
## Intensity - Rhythm T2 0.0743 0.0737 299 1.008 0.9892
## Intensity - Timbre T2 0.2682 0.0737 299 3.639 0.0039
## Pitch - Rhythm T2 -0.1274 0.0737 299 -1.728 0.6556
## Pitch - Timbre T2 0.0666 0.0737 299 0.903 0.9959
## Rhythm - Timbre T2 0.1939 0.0737 299 2.631 0.1023
##
## Results are averaged over the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 12 tests
```

```
IC_d.r[c(1,3,7,9)]
```

```
## Note: adjust = "tukey" was changed to "sidak"
## because "tukey" is only appropriate for one set of pairwise comparisons
```

```
## contrast      Time estimate      SE df t.ratio p.value
## Intensity - Pitch T1    -0.0769 0.0737 299  -1.044  0.7564
## Intensity - Timbre T1    -0.0107 0.0737 299  -0.146  0.9998
## Intensity - Pitch T2     0.2017 0.0737 299   2.736  0.0261
## Intensity - Timbre T2     0.2682 0.0737 299   3.639  0.0013
##
## Results are averaged over the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 4 tests
```

```
IC_IC_d.r <- contrast(emm_d.r[[1]], interaction = c("pairwise", "consec"), by = NULL)
rbind(IC_IC_d.r[1:6], adjust = "bonferroni")
```

```
## Deviant_pairwise Time_consec estimate      SE df t.ratio p.value
## Intensity - Pitch T2 - T1      0.278580 0.104 299   2.673  0.0476
## Intensity - Rhythm T2 - T1      0.209026 0.104 299   2.005  0.2749
## Intensity - Timbre T2 - T1      0.278942 0.104 299   2.676  0.0471
## Pitch - Rhythm T2 - T1     -0.069553 0.104 299  -0.667  1.0000
## Pitch - Timbre T2 - T1      0.000362 0.104 299   0.003  1.0000
## Rhythm - Timbre T2 - T1      0.069916 0.104 299   0.671  1.0000
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: bonferroni method for 6 tests
```

Hierarchical mixed effects modeling - 3-AFC (Behavioral)

```
M0=glmer(cbind(AvCor*6,6)~1+(1|ID), data=long, family="binomial", control = glmerControl(optimizer = "n
M1=glmer(cbind(AvCor*6,6)~Deviant+(1|ID), data=long, family="binomial", control = glmerControl(optimizer
M2=glmer(cbind(AvCor*6,6)~Deviant+DeviantLevel+(1|ID), data=long, family="binomial", control = glmerCon
M3=glmer(cbind(AvCor*6,6)~Deviant+DeviantLevel+Time+(1|ID), data=long,family="binomial", control = glme
M4=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Time+(1|ID), data=long, family="binomial", control = glm
M5=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Deviant*Time+(1|ID), data=long, family="binomial", contr
M6=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Deviant*Time+DeviantLevel*Time+(1|ID), data=long, family:
anova(M0,M1,M2,M3,M4,M5,M6)
```

```
## Data: long
## Models:
## M0: cbind(AvCor * 6, 6) ~ 1 + (1 | ID)
## M1: cbind(AvCor * 6, 6) ~ Deviant + (1 | ID)
```



```

## M2: cbind(AvCor * 6, 6) ~ Deviant + DeviantLevel + (1 | ID)
## M3: cbind(AvCor * 6, 6) ~ Deviant + DeviantLevel + Time + (1 | ID)
## M4: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Time + (1 | ID)
## M5: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Deviant * Time + (1 | ID)
## M6: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Deviant * Time + DeviantLevel * Time + (1 | ID)
##      npar      AIC      BIC logLik deviance      Chisq Df Pr(>Chisq)
## M0      2 1640.0 1647.3 -818.00   1636.0
## M1      5 1485.8 1504.1 -737.90   1475.8 160.2033   3 < 2.2e-16 ***
## M2      8 1424.8 1454.1 -704.41   1408.8  66.9716   3 1.899e-14 ***
## M3      9 1426.5 1459.4 -704.22   1408.5   0.3766   1   0.5394
## M4     18 1430.2 1496.1 -697.07   1394.2  14.3042   9   0.1119
## M5     21 1434.7 1511.6 -696.34   1392.7   1.4569   3   0.6923
## M6     24 1437.9 1525.8 -694.94   1389.9   2.8127   3   0.4214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Best model

```
summary(M2)
```

```

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: cbind(AvCor * 6, 6) ~ Deviant + DeviantLevel + (1 | ID)
## Data: long
## Control: glmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
##      AIC      BIC    logLik deviance df.resid
##  1424.8   1454.1   -704.4   1408.8     280
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -17.3877  -0.6257   0.0096   0.4718   2.6604
##
## Random effects:
## Groups Name      Variance Std.Dev.
## ID      (Intercept) 0.02301  0.1517
## Number of obs: 288, groups: ID, 9
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.20487    0.08169  39.231 < 2e-16 ***
## DeviantPitch   0.72414    0.06930  10.449 < 2e-16 ***
## DeviantRhythm  0.83091    0.06907  12.030 < 2e-16 ***
## DeviantTimbre  0.61835    0.06904   8.957 < 2e-16 ***
## DeviantLevel2  0.26418    0.06912   3.822 0.000132 ***
## DeviantLevel3  0.49143    0.06898   7.124 1.05e-12 ***
## DeviantLevel4  0.49728    0.06901   7.206 5.78e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:

```

```
##          (Intr) DvntPt DvntRh DvntTm DvntL2 DvntL3
## DeviantPtch -0.431
## DevntRhythm -0.399  0.505
## DeviantTmbr -0.419  0.504  0.504
## DeviantLvl2 -0.409 -0.008 -0.058 -0.008
## DeviantLvl3 -0.408 -0.004 -0.047 -0.031  0.504
## DeviantLvl4 -0.427  0.031 -0.028  0.006  0.504  0.505
```

Hierarchical mixed effects modeling - Latency

```
M0=lmer(Latency_peak~1+(1|ID), data=long, REML = FALSE, control = lmerControl(optimizer = "nloptwrap",
M1=lmer(Latency_peak~Deviant+(1|ID), data=long, REML = FALSE, control = lmerControl(optimizer = "nloptw
M2=lmer(Latency_peak~Deviant+DeviantLevel+(1|ID), data=long, REML = FALSE, control = lmerControl(optimi
M3=lmer(Latency_peak~Deviant+DeviantLevel+Time+(1|ID), data=long, REML = FALSE, control = lmerControl(op
M4=lmer(Latency_peak~Deviant*DeviantLevel+Time+(1|ID), data=long, REML = FALSE, control = lmerControl(op
M5=lmer(Latency_peak~Deviant*DeviantLevel+Deviant*Time+(1|ID), data=long, REML = FALSE, control = lmerC
M6=lmer(Latency_peak~Deviant*DeviantLevel+Deviant*Time+DeviantLevel*Time+(1|ID), data=long, REML = FALS

anova(M0,M1,M2,M3,M4,M5,M6)
```

```
## Data: long
## Models:
## M0: Latency_peak ~ 1 + (1 | ID)
## M1: Latency_peak ~ Deviant + (1 | ID)
## M2: Latency_peak ~ Deviant + DeviantLevel + (1 | ID)
## M3: Latency_peak ~ Deviant + DeviantLevel + Time + (1 | ID)
## M4: Latency_peak ~ Deviant * DeviantLevel + Time + (1 | ID)
## M5: Latency_peak ~ Deviant * DeviantLevel + Deviant * Time + (1 | ID)
## M6: Latency_peak ~ Deviant * DeviantLevel + Deviant * Time + DeviantLevel * Time + (1 | ID)
##      npar    AIC    BIC logLik deviance  Chisq Df Pr(>Chisq)
## M0      3 2899.4 2910.4 -1446.7  2893.4
## M1      6 2905.3 2927.3 -1446.7  2893.3  0.0828  3    0.9938
## M2      9 2909.0 2942.0 -1445.5  2891.0  2.3231  3    0.5081
## M3     10 2910.1 2946.7 -1445.0  2890.1  0.9634  1    0.3263
## M4     19 2915.2 2984.8 -1438.6  2877.2 12.8390  9    0.1700
## M5     22 2920.0 3000.6 -1438.0  2876.0  1.2282  3    0.7463
## M6     25 2921.8 3013.4 -1435.9  2871.8  4.1460  3    0.2461
```