

CI_group_analysis

Load packages

```
pacman::p_load(readr, lme4, emmeans, yarr, ggplot2, dplyr)
```

Load data

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

```
## Rows: 320 Columns: 7
```

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

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

```
## chr (4): Time, Channel, Deviant, Group
```

```
## dbl (3): ID, Amplitude, DeviantLevel
```

```
##
```

```
## 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.
```

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

```
## Rows: 576 Columns: 7
```

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

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

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

```
## dbl (4): ID, DeviantLevel, Amplitude.T1, Amplitude.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.
```

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

```
## Rows: 320 Columns: 7
```

```
## -- Column specification -----
## Delimiter: ","
## chr (4): Time, Channel, Deviant, Group
## dbl (3): ID, Latency_peak, DeviantLevel

##
## 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.
```

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

```
## Rows: 576 Columns: 7
```

```
## -- Column specification -----
## Delimiter: ","
## chr (3): Channel, Deviant, Group
## dbl (4): ID, DeviantLevel, Latency_peak.T1, Latency_peak.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.
```

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

```
## Rows: 752 Columns: 6
```

```
## -- Column specification -----
## Delimiter: ","
## chr (3): Round, Deviant, Group
## dbl (3): ID, DeviantLevel, AvCor

##
## 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.
```

```
amp$ID=as.factor(amp$ID)
amp$Deviant=as.factor(amp$Deviant)
amp$DeviantLevel=as.factor(amp$DeviantLevel)
amp$Group=as.factor(amp$Group)

amp_all$ID=as.factor(amp_all$ID)
amp_all$Deviant=as.factor(amp_all$Deviant)
amp_all$DeviantLevel=as.factor(amp_all$DeviantLevel)
amp_all$Group=as.factor(amp_all$Group)

lat$ID=as.factor(lat$ID)
lat$Deviant=as.factor(lat$Deviant)
lat$DeviantLevel=as.factor(lat$DeviantLevel)
lat$Group=as.factor(lat$Group)

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

```

lat_all$Deviant=as.factor(lat_all$Deviant)
lat_all$DeviantLevel=as.factor(lat_all$DeviantLevel)
lat_all$Group=as.factor(lat_all$Group)

All_behav$ID=as.factor(All_behav$ID)
All_behav$Deviant=as.factor(All_behav$Deviant)
All_behav$DeviantLevel=as.factor(All_behav$DeviantLevel)
All_behav$Group=as.factor(All_behav$Group)

```

Hierarchical mixed effects modeling - MMN Amplitude (T2)

```

M0=lmer(Amplitude.T2~1+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M1=lmer(Amplitude.T2~Deviant+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M2=lmer(Amplitude.T2~Deviant+DeviantLevel+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M3=lmer(Amplitude.T2~Deviant+DeviantLevel+Group+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M4=lmer(Amplitude.T2~Deviant*DeviantLevel+Group+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M5=lmer(Amplitude.T2~Deviant*DeviantLevel+Deviant*Group+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M6=lmer(Amplitude.T2~Deviant*DeviantLevel+Deviant*Group+DeviantLevel*Group+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))
M7=lmer(Amplitude.T2~Deviant*Group*DeviantLevel+(1|ID), data=amp_all, REML = FALSE, control = lmerControl(optimizer = "nloptwrap"))

#comparing models
anova(M0,M1,M2,M3,M4,M5,M6,M7)

```

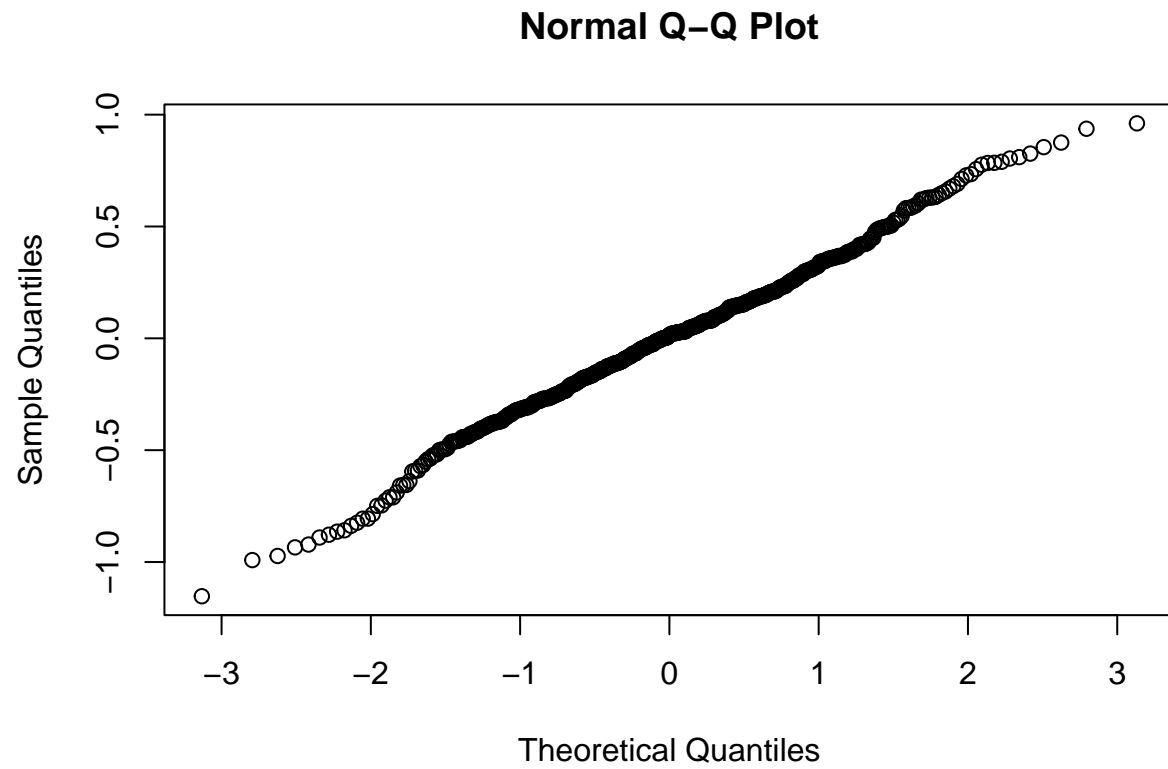
```

## Data: amp_all
## Models:
## M0: Amplitude.T2 ~ 1 + (1 | ID)
## M1: Amplitude.T2 ~ Deviant + (1 | ID)
## M2: Amplitude.T2 ~ Deviant + DeviantLevel + (1 | ID)
## M3: Amplitude.T2 ~ Deviant + DeviantLevel + Group + (1 | ID)
## M4: Amplitude.T2 ~ Deviant * DeviantLevel + Group + (1 | ID)
## M5: Amplitude.T2 ~ Deviant * DeviantLevel + Deviant * Group + (1 | ID)
## M6: Amplitude.T2 ~ Deviant * DeviantLevel + Deviant * Group + DeviantLevel * Group + (1 | ID)
## M7: Amplitude.T2 ~ Deviant * Group * DeviantLevel + (1 | ID)
##      npar    AIC    BIC logLik deviance   Chisq Df Pr(>Chisq)
## M0      3 630.01 643.08 -312.00   624.01
## M1      6 595.29 621.43 -291.64   583.29 40.7199  3 7.498e-09 ***
## M2      9 588.03 627.23 -285.01   570.03 13.2616  3 0.004104 **
## M3     11 580.13 628.05 -279.07   558.13 11.8963  2 0.002611 **
## M4     20 560.60 647.72 -260.30   520.60 37.5337  9 2.113e-05 ***
## M5     26 564.79 678.05 -256.40   512.79  7.8036  6 0.252851
## M6     32 566.77 706.17 -251.39   502.77 10.0219  6 0.123732
## M7     50 584.80 802.61 -242.40   484.80 17.9704 18 0.457605
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

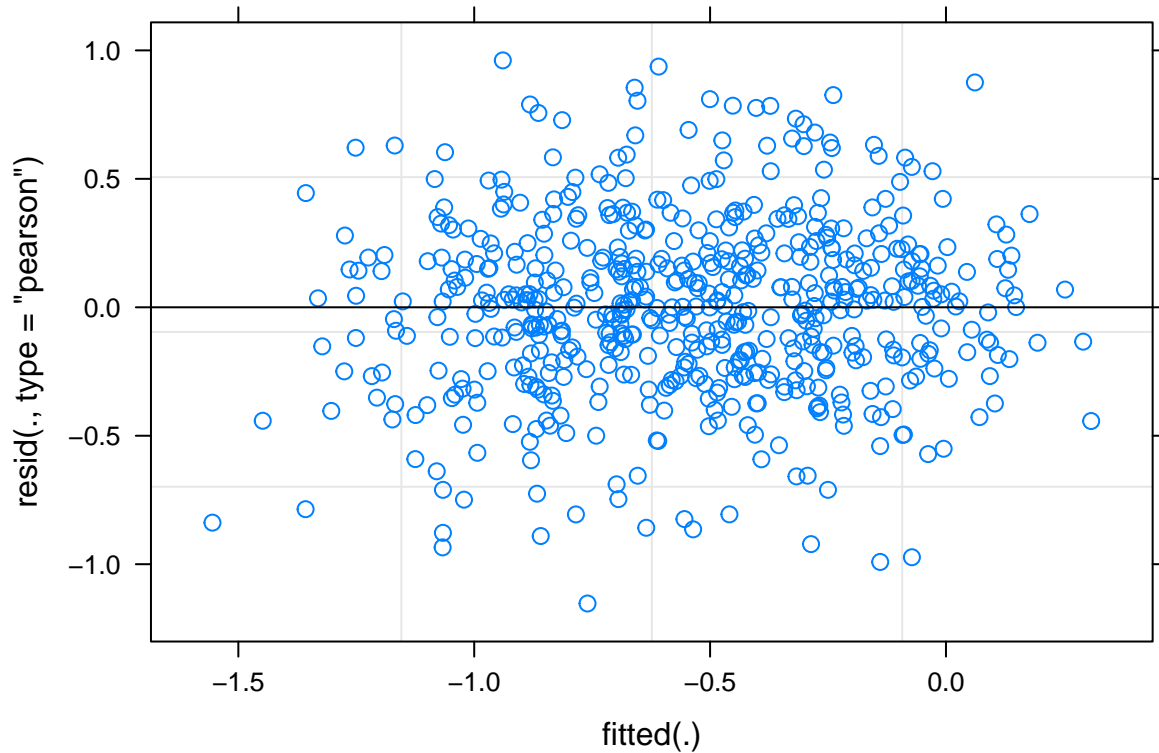
```

Best model

```
qqnorm(resid(M4))
```



```
plot(M4)
```



Post-hoc analysis

```
emm_g <- emmeans(M4, pairwise ~ Group)
emm_gc <- emm_g[[2]]

emm_d <- emmeans(M4, pairwise ~ Deviant)
```

NOTE: Results may be misleading due to involvement in interactions

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

#Group by deviant interaction

emm_g.d <- emmeans(M4, pairwise ~ Group | Deviant)
IC_g.d <- contrast(emm_g.d[[1]], method = "pairwise")

emm_d.g <- emmeans(M4, pairwise ~ Deviant | Group)
IC_d.g <- contrast(emm_d.g[[1]], method = "pairwise")

rbind(emm_gc, emm_dc, IC_g.d[1:4], IC_d.g[1:12], adjust = "mvt")
```

##	contrast	Deviant	Group	estimate	SE	df	t.ratio	p.value
##	CI_ex - CI_re	.	.	-0.4127	0.1320	39.3	-3.126	0.0256

```
## CI_ex - NH_old . . 0.0160 0.1173 39.3 0.137 1.0000
## CI_re - NH_old . . 0.4287 0.1301 39.3 3.296 0.0164
## Intensity - Pitch . . 0.0818 0.0422 555.4 1.940 0.3130
## Intensity - Rhythm . . -0.1078 0.0422 555.4 -2.557 0.0800
## Intensity - Timbre . . 0.1629 0.0422 555.4 3.862 0.0011
## Pitch - Rhythm . . -0.1897 0.0422 555.4 -4.497 0.0001
## Pitch - Timbre . . 0.0811 0.0422 555.4 1.922 0.3231
## Rhythm - Timbre . . 0.2707 0.0422 555.4 6.420 <.0001
## CI_ex - CI_re Intensity . -0.4127 0.1320 39.3 -3.126 0.0256
## CI_ex - NH_old Intensity . 0.0160 0.1173 39.3 0.137 1.0000
## CI_re - NH_old Intensity . 0.4287 0.1301 39.3 3.296 0.0165
## CI_ex - CI_re Pitch . -0.4127 0.1320 39.3 -3.126 0.0256
## Intensity - Pitch . CI_ex 0.0818 0.0422 555.4 1.940 0.3130
## Intensity - Rhythm . CI_ex -0.1078 0.0422 555.4 -2.557 0.0800
## Intensity - Timbre . CI_ex 0.1629 0.0422 555.4 3.862 0.0011
## Pitch - Rhythm . CI_ex -0.1897 0.0422 555.4 -4.497 0.0001
## Pitch - Timbre . CI_ex 0.0811 0.0422 555.4 1.922 0.3231
## Rhythm - Timbre . CI_ex 0.2707 0.0422 555.4 6.420 <.0001
## Intensity - Pitch . CI_re 0.0818 0.0422 555.4 1.940 0.3130
## Intensity - Rhythm . CI_re -0.1078 0.0422 555.4 -2.557 0.0801
## Intensity - Timbre . CI_re 0.1629 0.0422 555.4 3.862 0.0011
## Pitch - Rhythm . CI_re -0.1897 0.0422 555.4 -4.497 0.0001
## Pitch - Timbre . CI_re 0.0811 0.0422 555.4 1.922 0.3231
## Rhythm - Timbre . CI_re 0.2707 0.0422 555.4 6.420 <.0001
##
## Results are averaged over some or all of the levels of: Deviant, DeviantLevel, Group
## Degrees-of-freedom method: kenward-roger
## P value adjustment: mvt method for 25 tests
```

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

```
## contrast      Deviant estimate SE df t.ratio p.value
## CI_ex - CI_re Intensity -0.413 0.132 39.3 -3.126 0.0091
## CI_ex - NH_old Intensity 0.016 0.117 39.3 0.137 0.9897
## CI_re - NH_old Intensity 0.429 0.130 39.3 3.296 0.0058
## CI_ex - CI_re Pitch -0.413 0.132 39.3 -3.126 0.0092
##
## 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_g.d[1:4], adjust = "none")
```

```
## contrast      Deviant estimate SE df t.ratio p.value
## CI_ex - CI_re Intensity -0.413 0.132 39.3 -3.126 0.0033
## CI_ex - NH_old Intensity 0.016 0.117 39.3 0.137 0.8920
## CI_re - NH_old Intensity 0.429 0.130 39.3 3.296 0.0021
## CI_ex - CI_re Pitch -0.413 0.132 39.3 -3.126 0.0033
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
```

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

```
## contrast      Deviant estimate SE df t.ratio p.value
## CI_ex - CI_re Intensity -0.413 0.132 39.3 -3.126 0.0033
## CI_ex - NH_old Intensity 0.016 0.117 39.3 0.137 0.8920
## CI_ex - CI_re Pitch -0.413 0.132 39.3 -3.126 0.0033
##
## Results are averaged over some or all of the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
```

```
IC_g.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
## CI_ex - CI_re Intensity -0.413 0.132 39.3 -3.126 0.0133
## CI_ex - NH_old Intensity 0.016 0.117 39.3 0.137 0.9999
## CI_re - NH_old Intensity 0.429 0.130 39.3 3.296 0.0083
## CI_ex - CI_re Pitch -0.413 0.132 39.3 -3.126 0.0133
##
## Results are averaged over the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 4 tests
```

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

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

```
## contrast      Group estimate SE df t.ratio p.value
## Intensity - Pitch CI_ex 0.0818 0.0422 555 1.940 0.4789
## Intensity - Rhythm CI_ex -0.1078 0.0422 555 -2.557 0.1224
## Intensity - Timbre CI_ex 0.1629 0.0422 555 3.862 0.0015
## Pitch - Rhythm CI_ex -0.1897 0.0422 555 -4.497 0.0001
## Pitch - Timbre CI_ex 0.0811 0.0422 555 1.922 0.4932
## Rhythm - Timbre CI_ex 0.2707 0.0422 555 6.420 <.0001
## Intensity - Pitch CI_re 0.0818 0.0422 555 1.940 0.4789
## Intensity - Rhythm CI_re -0.1078 0.0422 555 -2.557 0.1224
## Intensity - Timbre CI_re 0.1629 0.0422 555 3.862 0.0015
## Pitch - Rhythm CI_re -0.1897 0.0422 555 -4.497 0.0001
## Pitch - Timbre CI_re 0.0811 0.0422 555 1.922 0.4932
## Rhythm - Timbre CI_re 0.2707 0.0422 555 6.420 <.0001
##
## Results are averaged over the levels of: DeviantLevel
## Degrees-of-freedom method: kenward-roger
## P value adjustment: sidak method for 12 tests
```

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

Plot

```
ID_plot_data = read_csv("ID_plot_data_amp.csv")
```

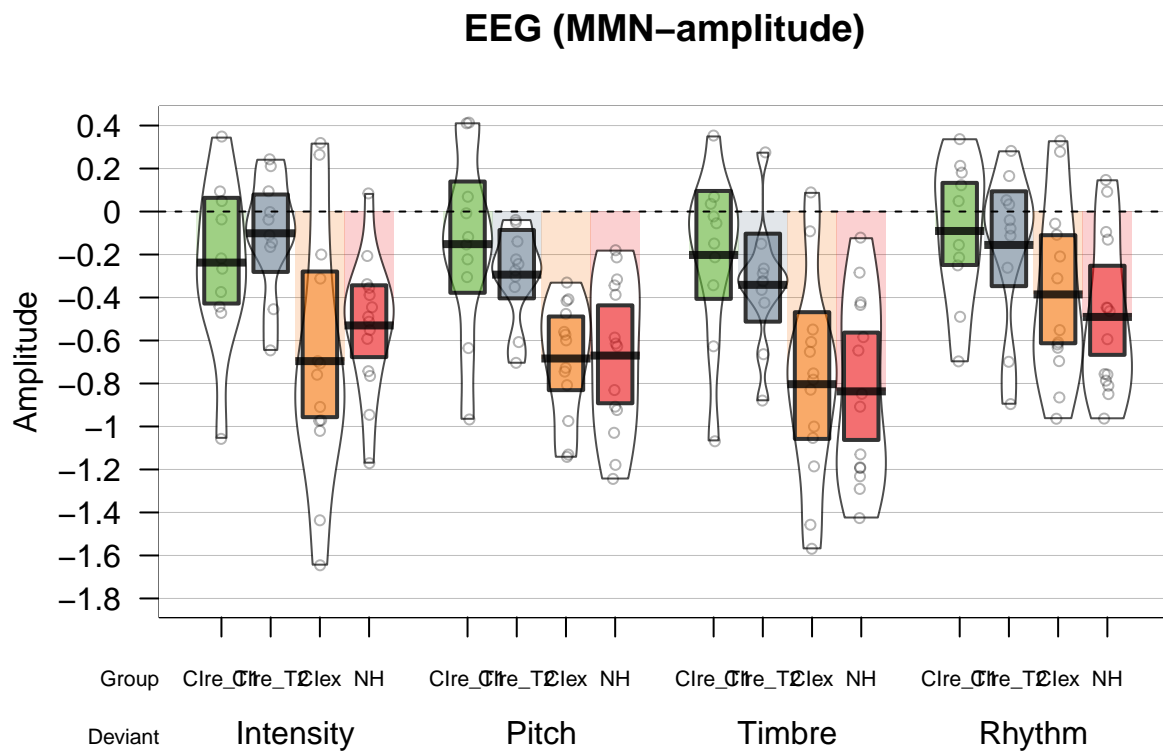
```
## Rows: 188 Columns: 5
```

```
## -- Column specification -----
## Delimiter: ","
## chr (3): Group, Time, Deviant
## dbl (2): ID, Amp
```

```
##
## 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.
```

```
pirat = ID_plot_data
pirat$Group[pirat$Time == "CIre_T1"] = gsub("CI_re", "CIre_T1", pirat$Group[pirat$Time == "CIre_T1"])
pirat$Group[pirat$Time == "CIre_T2"] = gsub("CI_re", "CIre_T2", pirat$Group[pirat$Time == "CIre_T2"])
pirat$Group = ordered(pirat$Group, levels = c("CIre_T1", "CIre_T2", "CIex", "NH"))
pirat$Deviant = ordered(pirat$Deviant, levels = c("Intensity", "Pitch", "Timbre", "Rhythm"))

pirateplot(formula = Amp ~ Group + Deviant, data = pirat, main = "EEG (MMN-amplitude)", xlab = "group",
  abline(a=0, b=0, lwd=1, lty="dashed")
```



```
## integer(0)
```


Hierarchical mixed effects modeling - Latency (T2)

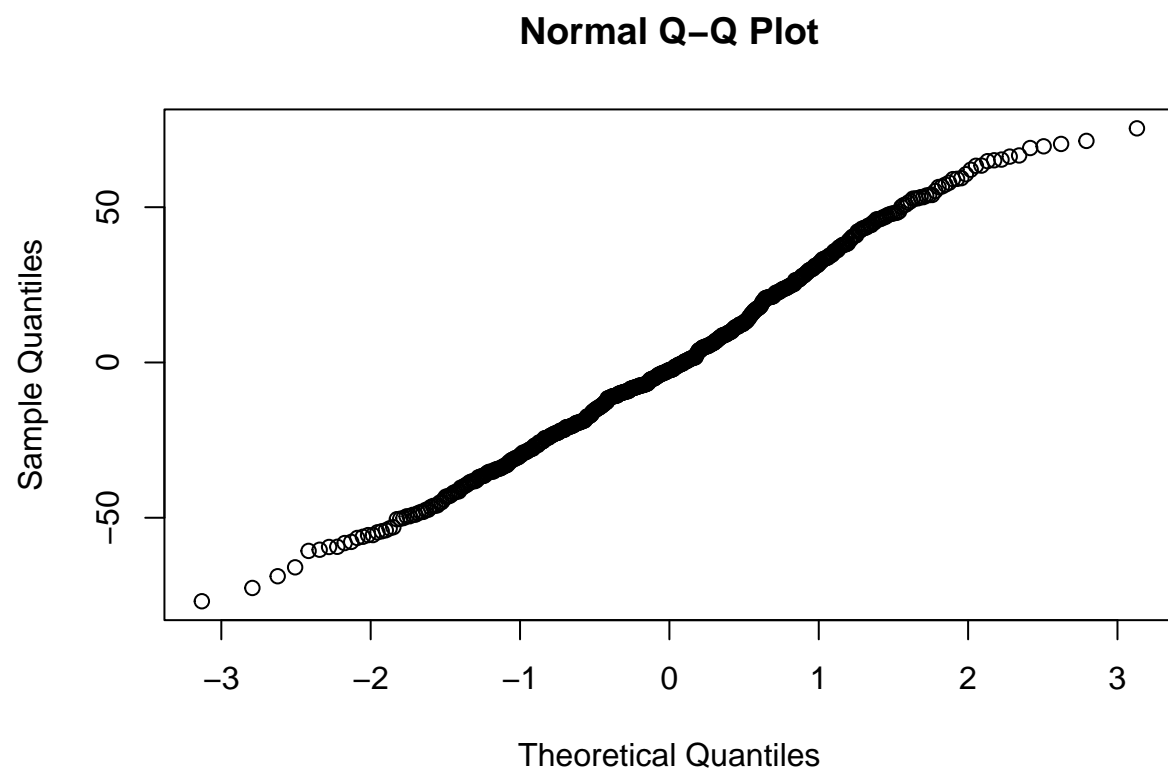
```
M0=lmer(Latency_peak.T2~1+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M1=lmer(Latency_peak.T2~Deviant+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M2=lmer(Latency_peak.T2~Deviant+DeviantLevel+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M3=lmer(Latency_peak.T2~Deviant+DeviantLevel+Group+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M4=lmer(Latency_peak.T2~Deviant*DeviantLevel+Group+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M5=lmer(Latency_peak.T2~Deviant*DeviantLevel+Deviant*Group+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M6=lmer(Latency_peak.T2~Deviant*DeviantLevel+Deviant*Group+DeviantLevel*Group+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw
M7=lmer(Latency_peak.T2~Deviant*DeviantLevel*Group+(1|ID), data=lat_all, REML = FALSE, control = lmerControl(optimizer = "nloptw

#comparing models
anova(M0,M1,M2,M3,M4,M5,M6, M7)
```

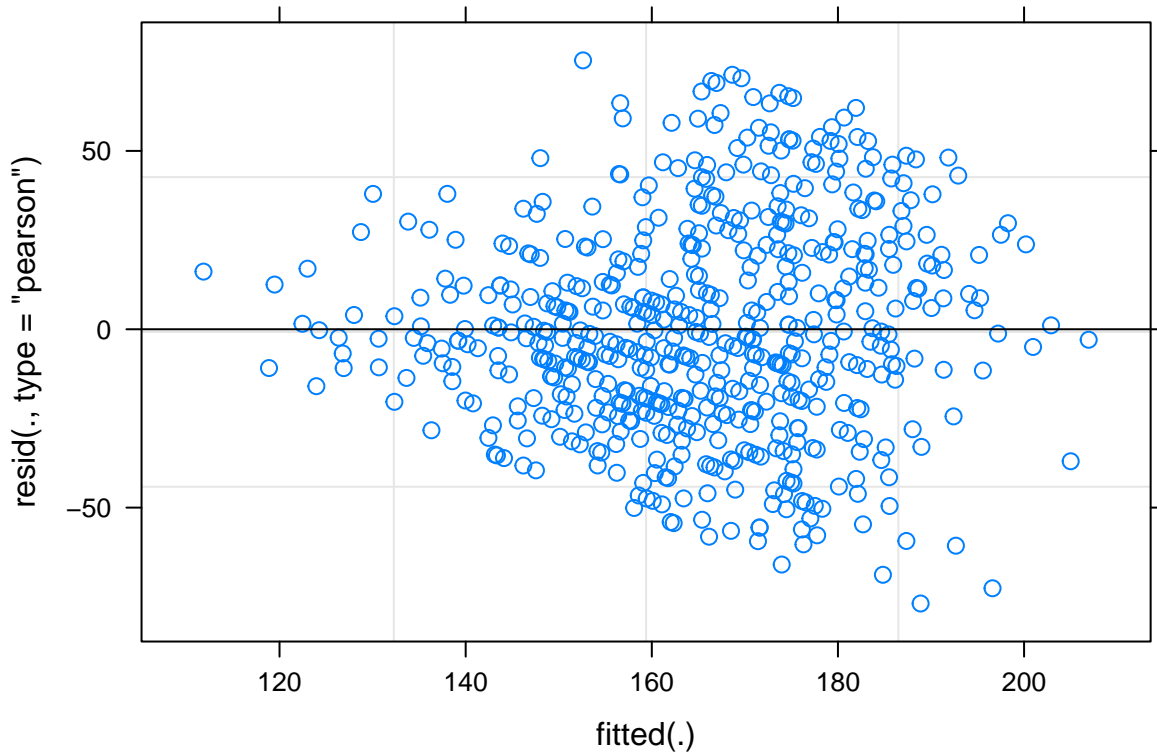
```
## Data: lat_all
## Models:
## M0: Latency_peak.T2 ~ 1 + (1 | ID)
## M1: Latency_peak.T2 ~ Deviant + (1 | ID)
## M2: Latency_peak.T2 ~ Deviant + DeviantLevel + (1 | ID)
## M3: Latency_peak.T2 ~ Deviant + DeviantLevel + Group + (1 | ID)
## M4: Latency_peak.T2 ~ Deviant * DeviantLevel + Group + (1 | ID)
## M5: Latency_peak.T2 ~ Deviant * DeviantLevel + Deviant * Group + (1 | ID)
## M6: Latency_peak.T2 ~ Deviant * DeviantLevel + Deviant * Group + DeviantLevel * Group + (1 | ID)
## M7: Latency_peak.T2 ~ Deviant * DeviantLevel * Group + (1 | ID)
##      npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## M0      3 5647.3 5660.4 -2820.7   5641.3
## M1      6 5645.7 5671.8 -2816.9   5633.7  7.5894  3 0.0553056 .
## M2      9 5648.2 5687.3 -2815.1   5630.2  3.5513  3 0.3141733
## M3     11 5632.1 5680.0 -2805.1   5610.1 20.0574  2 4.411e-05 ***
## M4     20 5628.1 5715.2 -2794.1   5588.1 21.9737  9 0.0089626 **
## M5     26 5616.1 5729.3 -2782.1   5564.1 23.9943  6 0.0005235 ***
## M6     32 5624.6 5763.8 -2780.3   5560.6  3.5498  6 0.7373360
## M7     50 5646.3 5863.8 -2773.1   5546.3 14.3246 18 0.7076967
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Best model

```
qqnorm(resid(M5))
```



```
plot(M5)
```



```
## Plot
```

```
ID_plot_data = read_csv("ID_plot_data_lat.csv")
```

```
## Rows: 188 Columns: 5
```

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

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

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

```
## dbl (2): ID, Lat
```

```
##
```

```
## 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.
```

```
pirat = ID_plot_data
```

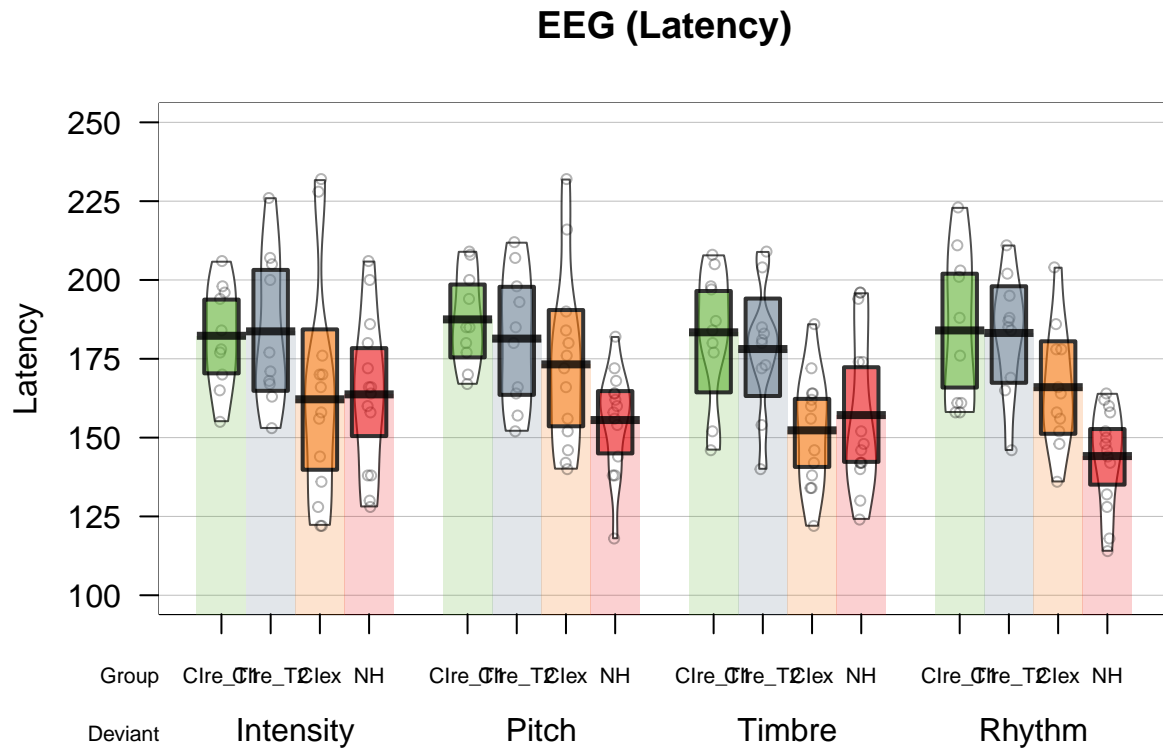
```
pirat$Group[pirat$Time == "CIre_T1"] = gsub("CI_re", "CIre_T1", pirat$Group[pirat$Time == "CIre_T1"])
```

```
pirat$Group[pirat$Time == "CIre_T2"] = gsub("CI_re", "CIre_T2", pirat$Group[pirat$Time == "CIre_T2"])
```

```
pirat$Group = ordered(pirat$Group, levels = c("CIre_T1", "CIre_T2", "CIex", "NH"))
```

```
pirat$Deviant = ordered(pirat$Deviant, levels = c("Intensity", "Pitch", "Timbre", "Rhythm"))
```

```
pirateplot(formula = Lat ~ Group + Deviant, data = pirat, main = "EEG (Latency)", xlab = "group", ylab = "Latency")
```



Hierarchical mixed effects modeling - Behavioral (T2)

```

M0=glmer(cbind(AvCor*6,6)~1+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",], family="binomial", con
M1=glmer(cbind(AvCor*6,6)~Deviant+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",], family="binomial
M2=glmer(cbind(AvCor*6,6)~Deviant+DeviantLevel+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",], fam
M3=glmer(cbind(AvCor*6,6)~Deviant+DeviantLevel+Group+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",]
M4=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Group+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",]
M5=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Deviant*Group+(1|ID), data=All_behav[All_behav$Group!="C
M6=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel+Deviant*Group+DeviantLevel*Group+(1|ID), data=All_behav[
M7=glmer(cbind(AvCor*6,6)~Deviant*DeviantLevel*Group+(1|ID), data=All_behav[All_behav$Group!="C_Ire_T1",]

#Comparing models
anova(M0, M1, M2, M3, M4, M5, M6, M7)

```

```
## Data: All_behav[All_behav$Group != "CIre_T1", ]
## 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 + Group + (1 | ID)
## M4: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Group + (1 | ID)
## M5: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Deviant * Group + (1 | ID)
## M6: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel + Deviant * Group + DeviantLevel * Group + (1 | ID)
## M7: cbind(AvCor * 6, 6) ~ Deviant * DeviantLevel * Group + (1 | ID)
##      npar      AIC      BIC logLik deviance      Chisq Df Pr(>Chisq)
## M0      2 3285.4 3294.2 -1640.7   3281.4
## M1      5 3148.9 3170.9 -1569.4   3138.9 142.4992   3 < 2.2e-16 ***
## M2      8 3023.4 3058.6 -1503.7   3007.4 131.5106   3 < 2.2e-16 ***
## M3     10 3013.1 3057.2 -1496.5   2993.1  14.2960   2 0.0007865 ***
## M4     19 3000.5 3084.3 -1481.2   2962.5  30.5745   9 0.0003501 ***
## M5     25 2989.8 3100.0 -1469.9   2939.8  22.7074   6 0.0009006 ***
## M6     31 2999.6 3136.4 -1468.8   2937.6   2.1517   6 0.9052355
## M7     49 3028.3 3244.4 -1465.2   2930.3   7.3054  18 0.9872881
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Plots behavioral

```
ID_plot_data = read_csv("ID_plot_data_behav.csv")
```

```
## Rows: 188 Columns: 5
```

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

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

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

```
## dbl (2): ID, AvCor
```

```
##
```

```
## 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.
```

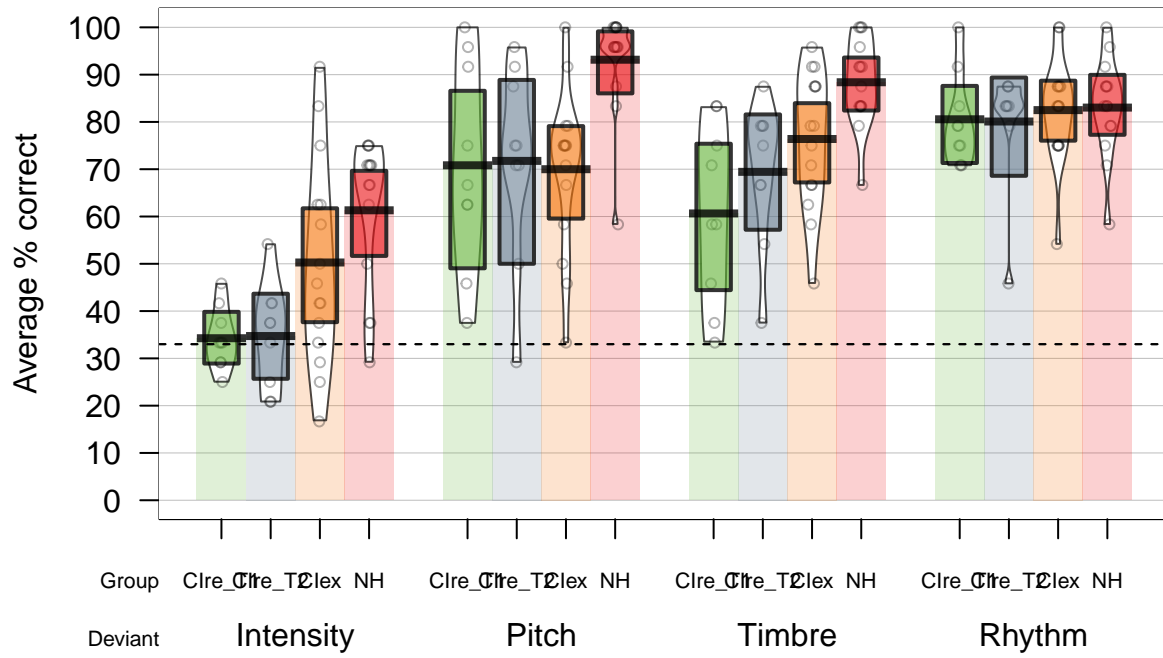
```
ID_plot_data$Group=ordered(ID_plot_data$Group, levels=c("CIre_T1", "CIre_T2", "CIex", "NH"))
```

```
ID_plot_data$Deviant=ordered(ID_plot_data$Deviant, levels=c("Intensity", "Pitch", "Timbre", "Rhythm"))
```

```
## Creating pirateplot
```

```
pirateplot(formula = AvCor ~ Group + Deviant, data = ID_plot_data, main = "Behavioral (average correct)"
  abline(a=33, b=0, lwd=1, lty="dashed")
```

Behavioral (average correct)



integer(0)