Data Analysis for Perception and Action exam paper on Human Detection of Audio Deepfakes by Hannah Cohen and Aurelija Spunde

Aurelija Spunde

2026-01-05

Importing data and loading libraries

```
library(tidyverse)
library(lme4)
library(lmerTest)
library(dplyr)
library(emmeans)
library(psycho)
library(DHARMa)
library(psych)
library(mgcv)
library(patchwork)
```

```
files <- list.files(path = "logfiles" , full.names = TRUE)

data <- purrr::map_dfr(files, readr::read_csv)

head(data)
```

ParticipantID	A..	Gen…	EnglishNativity	Familiarity	Trial	Filename	Respo
<dbl>	<dbl>	<chr>	<chr>	<chr>	<dbl>	<chr>	
1	20	Female	No	Unfamiliar	1	LA_D_2773764.flac	0.
1	20	Female	No	Unfamiliar	2	LA_D_7339762.flac	0.
1	20	Female	No	Unfamiliar	3	LA_D_8884919.flac	0.
1	20	Female	No	Unfamiliar	4	LA_D_1026446.flac	0.
1	20	Female	No	Unfamiliar	5	LA_D_1000265.flac	2.
1	20	Female	No	Unfamiliar	6	LA_D_1556595.flac	2.

6 rows | 1-8 of 29 columns

```
data <- data %>%
  mutate(
    ParticipantID = factor(ParticipantID),
    EnglishNativity = factor(EnglishNativity),
    Familiarity = factor(Familiarity),
    Filename = factor(Filename),
    ActualAuthenticity = factor(ActualAuthenticity),
    Difficulty = factor(Difficulty),
    Condition = factor(Condition),
    Correct = as.integer(Correct),
    ResponseTime = as.numeric(ResponseTime),
    Response = factor(Response),
    Confidence = as.numeric(Confidence),
    Naturalness = as.numeric(Naturalness),
  )
```

Data Preprocessing

Data filtering

```
data <- data %>% filter(ResponseTime >= 0.2, ResponseTime <= 10)

colSums(is.na(data))
```

```
##      ParticipantID                Age             Gender    EnglishNativity
##                  0                  0                  0                  0
##        Familiarity              Trial           Filename       ResponseTime
##                  0                  0                  0                  0
## ActualAuthenticity         Difficulty          Condition           Response
##                  0                  0                  0                  0
##            Correct         Confidence        Naturalness            F0_mean
##                  0                  0                  0                  0
##             F0_std     Intensity_mean      Intensity_std                 F1
##                  0                  0                  0                  0
##                 F2                 F3             Jitter            Shimmer
##                  0                  0               1689               1689
##                HNR   SpectralCentroid  SpectralBandwidth    SpectralRolloff
##                  0                  0                  0                  0
##   ZeroCrossingRate
##                  0
```

Standardizing data

```
data$Confidence_z <- scale(data$Confidence)

audio_data <- data %>%
  distinct(Filename, F0_mean, F0_std, Intensity_mean, Intensity_std,
           F1, F2, F3, HNR, SpectralCentroid, SpectralBandwidth, ZeroCrossingRate)

acoustics_scaled <- scale(audio_data %>% select(-Filename))
```

Participants’ Demographics

```
participants_age <-  data %>%
  distinct(ParticipantID, .keep_all = TRUE) %>%
  summarise(Range = range(Age),
            Mean = mean(Age),
            SD = sd(Age))

participants_age
```

Range	Mean	SD
<dbl>	<dbl>	<dbl>
19	22.27273	2.05129
27	22.27273	2.05129

2 rows

```
participants_gender <-  data %>%
  distinct(ParticipantID, .keep_all = TRUE) %>%
  count(Gender)

participants_gender
```

Gender	n
<chr>	<int>
Female	12
Male	10

2 rows

Data Analysis Plan

0. Overall Accuracy

1. Signal Detection Theory (SDT)

2. Behavioral Performance and Subjective Judgments Analyses

   2.1. Response Time by Condition

   2.2. Accuracy

   2.2.1. Accuracy by Condition

   2.2.2. Accuracy by Confidence

   2.3. Naturalness by Condition

   2.4. Confidence by Condition

3. Analysis of Acoustic Features

   3.1. Principal Component Analysis (PCA)

0. Overall Accuracy

```
# Accuracy per participant for each of the conditions

data_acc_participants <- data %>%
  group_by(Condition, ParticipantID) %>%
  summarise(
    Accuracy = mean(Correct),
    sd = sd(Correct)
  )

head(data_acc_participants)
```

Condition <fct>	ParticipantID <fct>	Accuracy <dbl>	sd <dbl>
fake_easy	1	1.0000000	0.0000000
fake_easy	2	1.0000000	0.0000000
fake_easy	3	1.0000000	0.0000000
fake_easy	4	0.8947368	0.3153018
fake_easy	5	1.0000000	0.0000000
fake_easy	6	1.0000000	0.0000000

6 rows

```
# Overall accuracy

data_acc_pop <- data %>%
  group_by(Condition) %>%
  summarise(
    pop_Accuracy = mean(Correct),
    sd = sd(Correct)
  )

data_acc_pop
```

Condition <fct>	pop_Accuracy <dbl>	sd <dbl>
fake_easy	0.9686747	0.1744056
fake_hard	0.7274882	0.4457803
real_easy	0.7872340	0.4097481
real_hard	0.7156177	0.4516464

4 rows

```
data_accuracy <- data_acc_participants %>%
  summarise(
    n_participants = n(),
    overall_mean = mean(Accuracy),
    overall_sd = sd(Accuracy),
    overall_se = overall_sd / sqrt(n_participants)
  )

print(paste("Overall mean accuracy is ", round(mean(data_accuracy$overall_mean),3), "(SE = ",
round(mean(data_accuracy$overall_se),3),")"))
```

```
## [1] "Overall mean accuracy is  0.8 (SE =  0.027 )"
```

1. Signal Detection Theory (SDT)

Defining responses

```
sdt <- data %>%
  mutate(
    Type = case_when(
      ActualAuthenticity == "fake" & Response == "fake" ~ "hit", # Hit
      ActualAuthenticity == "real" & Response == "real" ~ "cr",  # Correct rejection
      ActualAuthenticity == "fake" & Response == "real" ~ "miss", # Miss
      ActualAuthenticity == "real" & Response == "fake" ~ "fa", # False alarm
    )
  )

sdt %>%
  group_by(ActualAuthenticity) %>%
  count(ActualAuthenticity, Response, Correct, Type)
```

ActualAuthenticity	Response	Correct	Type	n
<fct>	<fct>	<int>	<chr>	<int>
fake	fake	1	hit	709
fake	real	0	miss	128
real	fake	0	fa	212
real	real	1	cr	640

4 rows

Aggregating per participant x difficulty

```
 sdt <- sdt %>%
  count(ParticipantID, Difficulty, Type) %>%
  pivot_wider(
    names_from = Type,
    values_from = n,
    values_fill = 0
  )
```

Calculating hit rate and false alarm rate

```
sdt <- sdt %>%
  mutate(
    hit_rate = (hit + 0.5) / (hit + miss + 1),
    fa_rate = (fa + 0.5) / (fa + cr + 1),
    zhit_rate = qnorm(hit_rate),
    zfa_rate = qnorm(fa_rate),
    dprime = zhit_rate - zfa_rate,
    criterion = -0.5 * (zhit_rate + zfa_rate)
  )
```

Population level Sensitivity and Criterion

```
sdt_sum <- sdt %>%
  group_by(Difficulty) %>%
  summarise(
    pop_dprime = mean(dprime),
    pop_criterion = mean(criterion),
    pop_hit_rate = mean(hit_rate),
    pop_fa_rate = mean(fa_rate),
  )

sdt_sum
```

Difficulty	pop_dprime	pop_criterion	pop_hit_rate	pop_fa_rate
<fct>	<dbl>	<dbl>	<dbl>	<dbl>
easy	2.614337	-0.4395175	0.9443860	0.2267303
hard	1.205645	-0.0190136	0.7168366	0.2951735

2 rows

Report: Overall participants were more sensitive in detection of fake audio in the easy condition (*d’* = 2.6) compared to hard condition (*d’* = 1.2). Moreover, in the easy condition they showed a more liberal bias towards categorizing an audio as fake (*c* = -.44) compared to the hard condition where they showed almost no bias at all (*c* = -.02). The hit rate is by almost 23 percent points lower in the hard condition (71.68%) than in the easy condition (94.43%). The false alarm rate is by almost 7 percent points higher in the hard condition (29.52%) as compared to the easy condition (22.67%).

Modeling Sensitivity by Difficulty

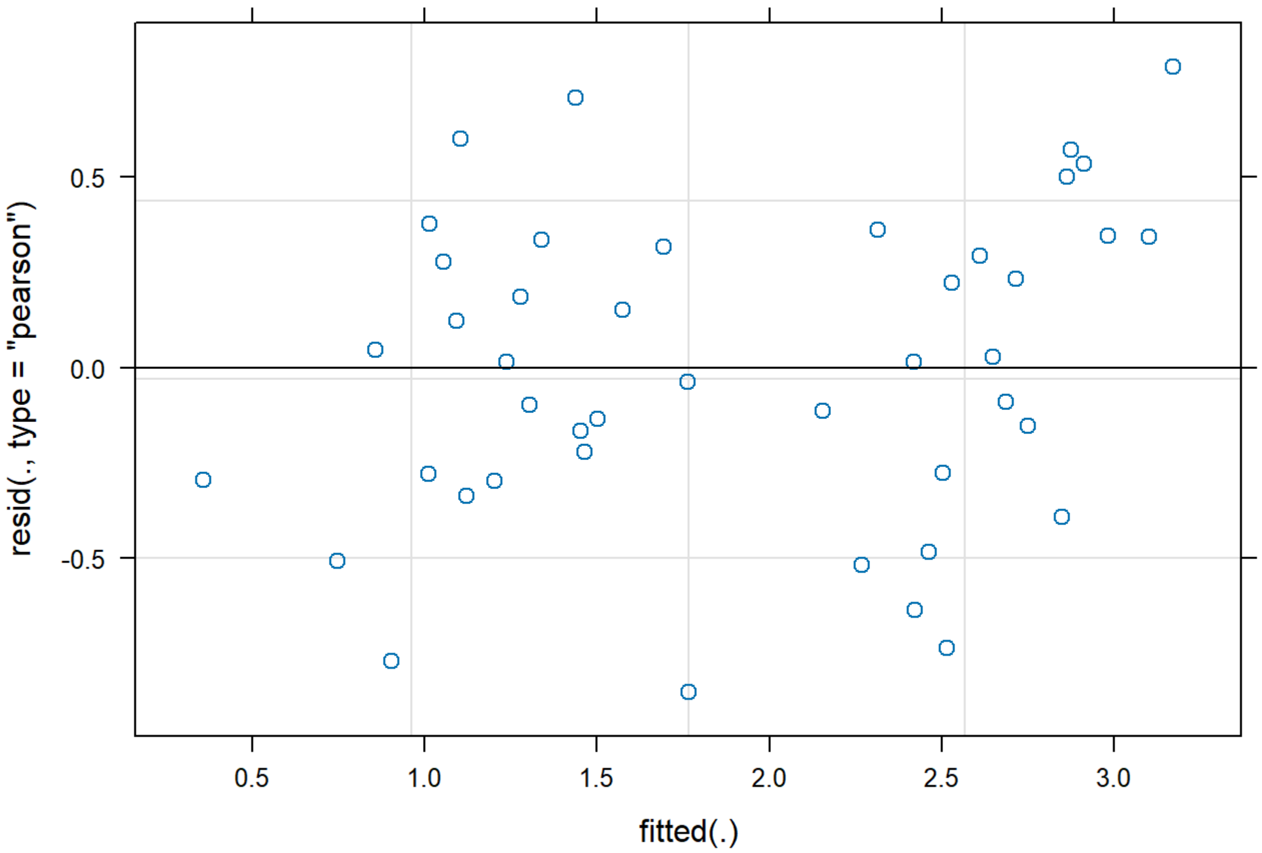
```
m_dprime <- lmer(dprime ~ Difficulty + (1| ParticipantID), data = sdt)

summary(m_dprime)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: dprime ~ Difficulty + (1 | ParticipantID)
##    Data: sdt
## 
## REML criterion at convergence: 85.1
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max 
## -1.72432 -0.57287  0.03022  0.65072  1.60037 
## 
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  ParticipantID (Intercept) 0.1801   0.4244  
##  Residual                  0.2438   0.4938  
## Number of obs: 44, groups:  ParticipantID, 22
## 
## Fixed effects:
##                Estimate Std. Error      df t value Pr(>|t|)    
## (Intercept)      2.6143     0.1388 35.5773  18.833  < 2e-16 ***
## Difficultyhard  -1.4087     0.1489 21.0000  -9.462 5.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr)
## Diffcltyhrd -0.536
```

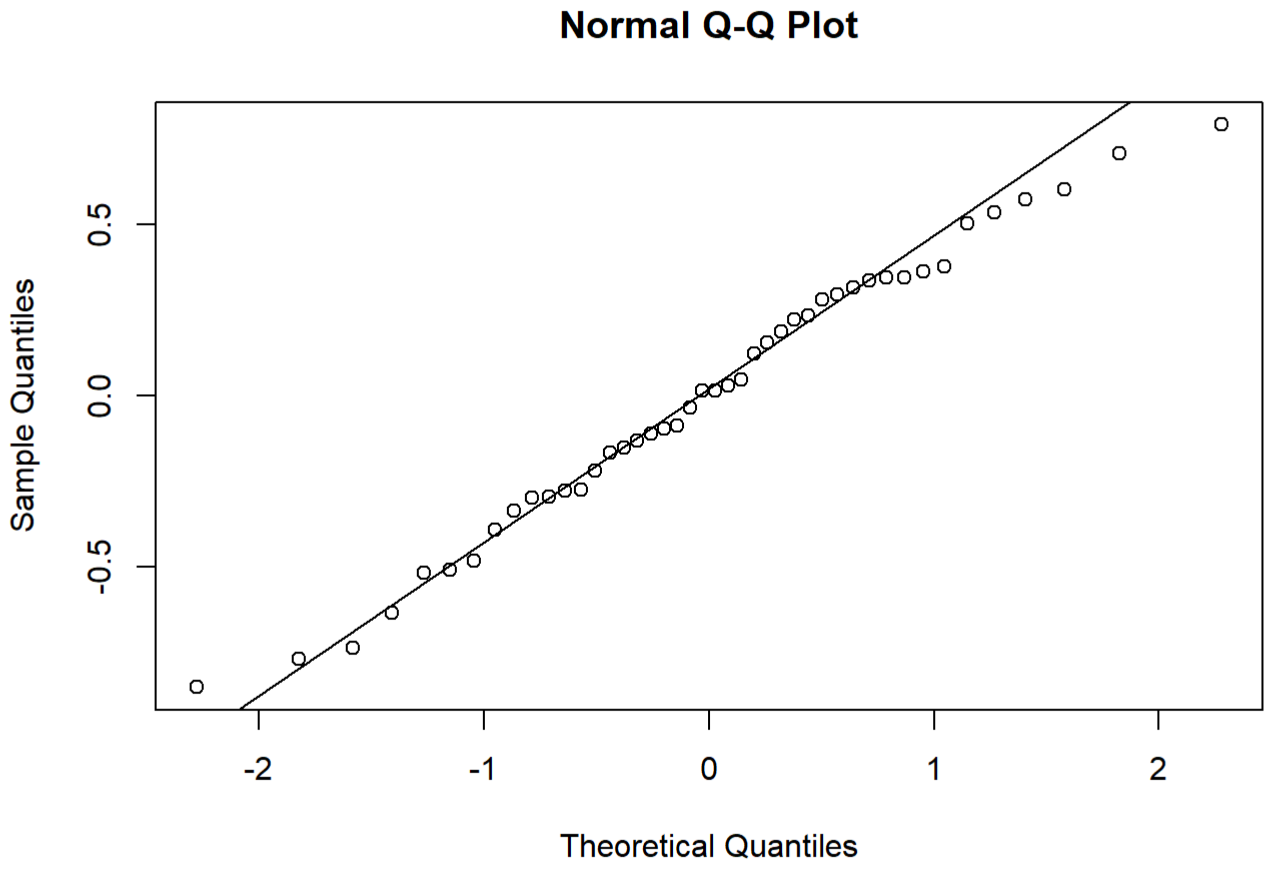
Report: A linear mixed-effects model was fitted to predict detection sensitivity (d’) by Difficulty as fixed effect and random intercept for participants. Results showed a significant effect of Difficulty on the sensitivity, with d’ dropping by more than 50% in the hard condition (*M* = 1.2, *SE* = 0.14) compared to the the easy condition (*M* = 2.6, *SE* = 0.15), representing a significant decrease (*b* = -1.4, *SE* = 0.15, *t*(21) = -9.46, *p* < .0001).

Checking assumptions

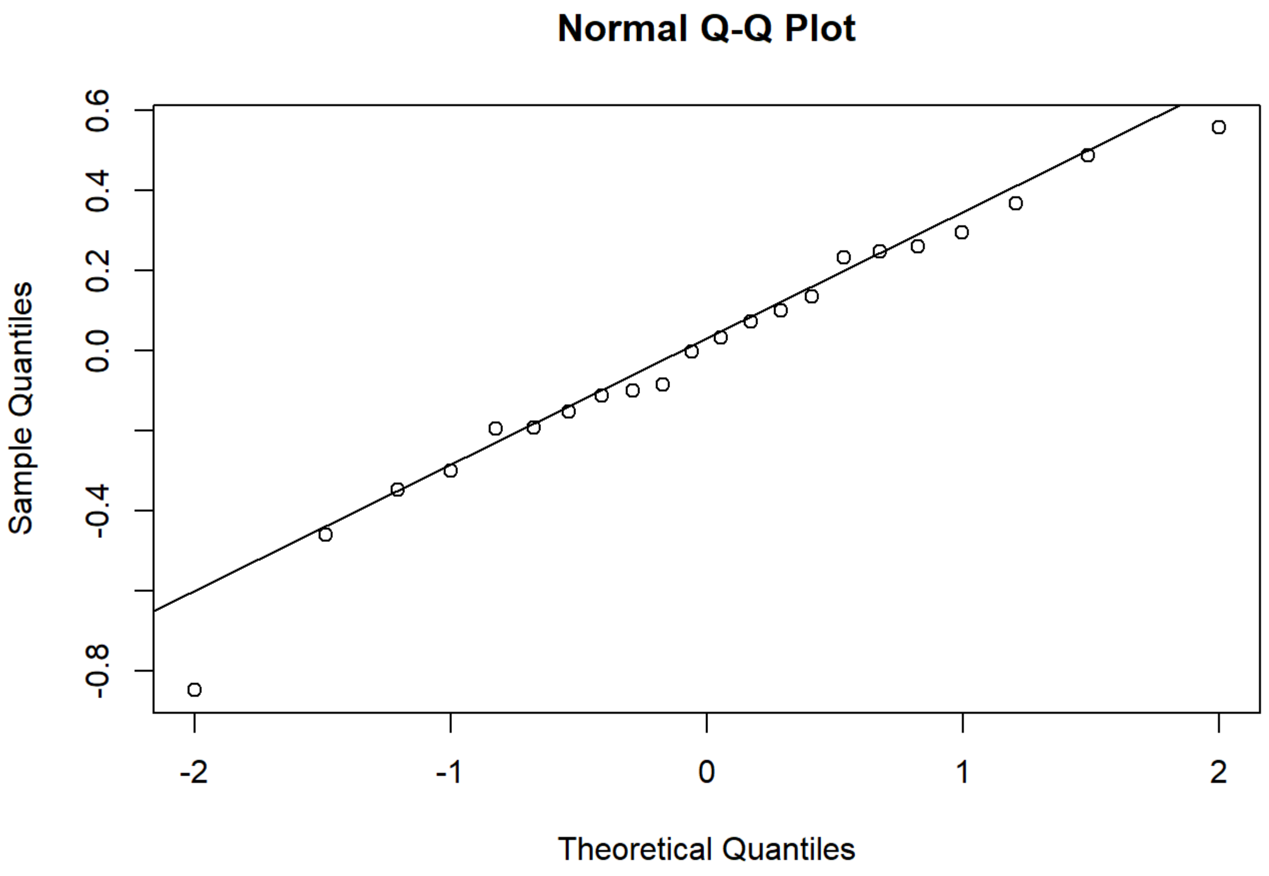
```
plot(m_dprime)
```



```
qqnorm(residuals(m_dprime))
qqline(residuals(m_dprime))
```



```
qqnorm(ranef(m_dprime)$ParticipantID[[1]])
qqline(ranef(m_dprime)$ParticipantID[[1]])
```



Report: The visual inspection of QQ-plots for residuals and random effects showed no significant deviation from normality.

Modeling bias by Difficulty

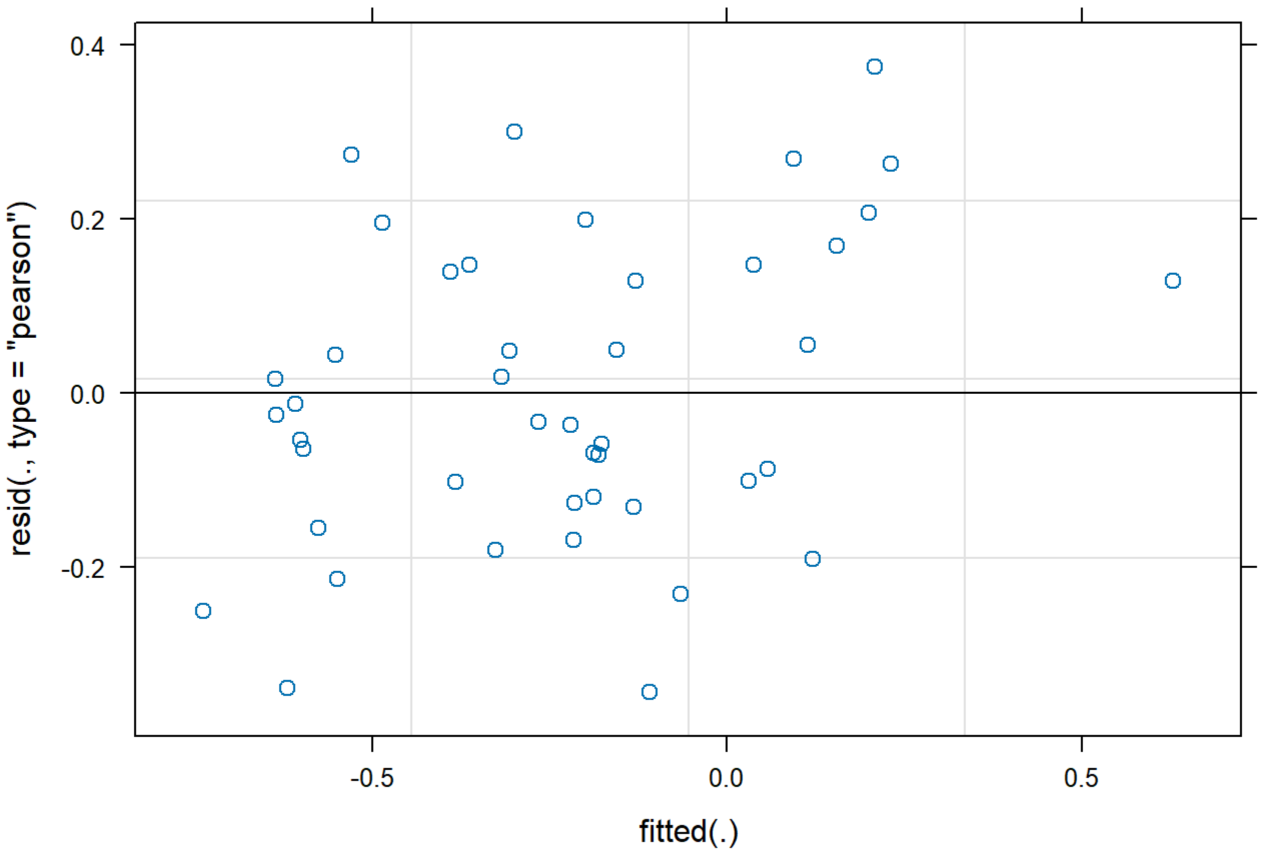
```
m_criterion <- lmer(criterion ~ Difficulty + (1| ParticipantID), data = sdt)

summary(m_criterion)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: criterion ~ Difficulty + (1 | ParticipantID)
##    Data: sdt
## 
## REML criterion at convergence: 25.6
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -1.5542 -0.5461 -0.1341  0.6372  1.6924 
## 
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  ParticipantID (Intercept) 0.06324  0.2515  
##  Residual                  0.04920  0.2218  
## Number of obs: 44, groups:  ParticipantID, 22
## 
## Fixed effects:
##                Estimate Std. Error       df t value Pr(>|t|)    
## (Intercept)    -0.43952    0.07149 31.90631  -6.148 7.16e-07 ***
## Difficultyhard  0.42050    0.06688 21.00000   6.288 3.10e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr)
## Diffcltyhrd -0.468
```

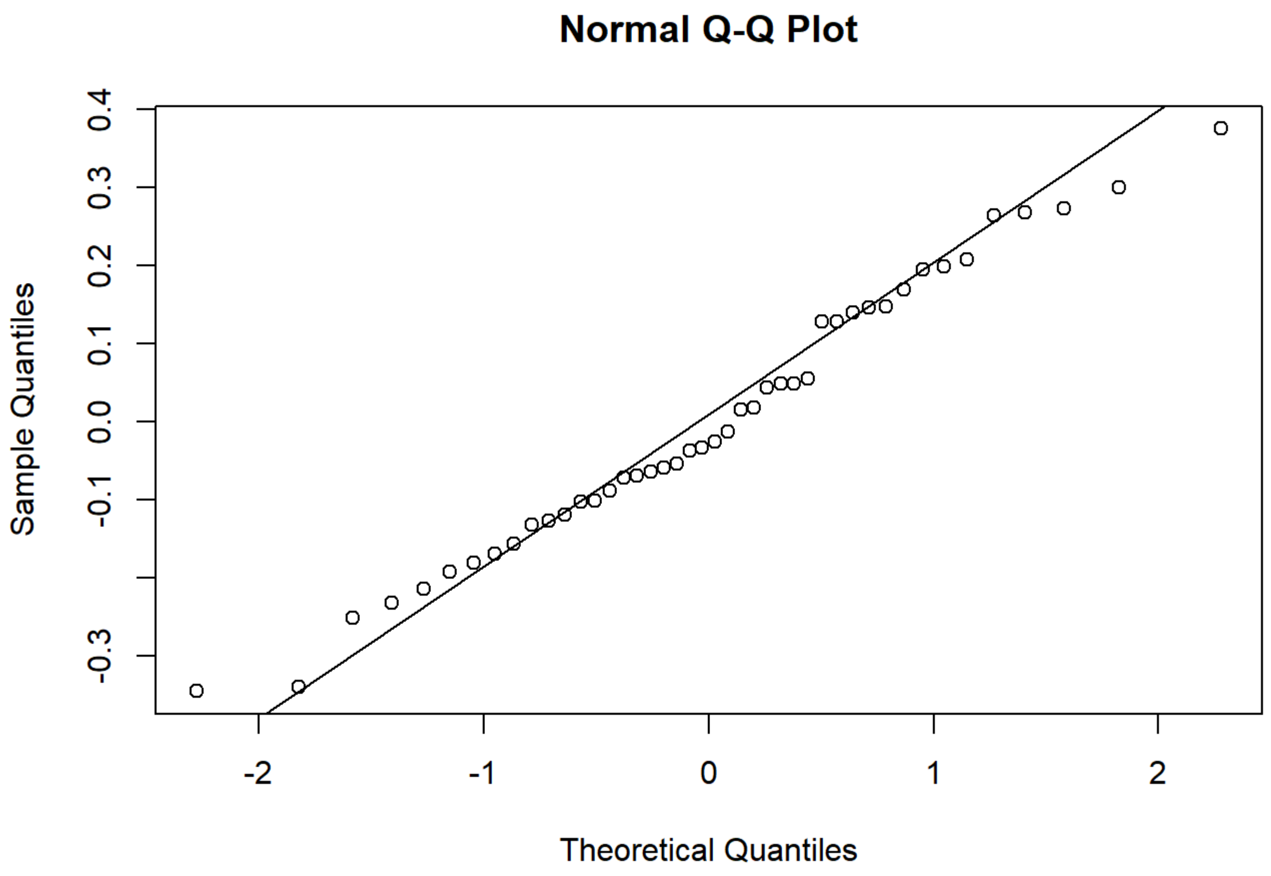
Report: A linear mixed-effects model was fitted to predict detection bias (c) by Difficulty as fixed effect and random intercept for participants. It showed that Difficulty has a significant effect on the criterion, with participants showing by 95% less bias in the hard condition (*M* = -0.02, *SE* = 0.07) as compared to the easy condition (*M* = -0.44, *SE* = 0.07), resulting in a shift towards neutrality (*b* = 0.42, *SE* = 0.07, *t*(21) = 6.29, *p* < .0001).

Checking assumptions

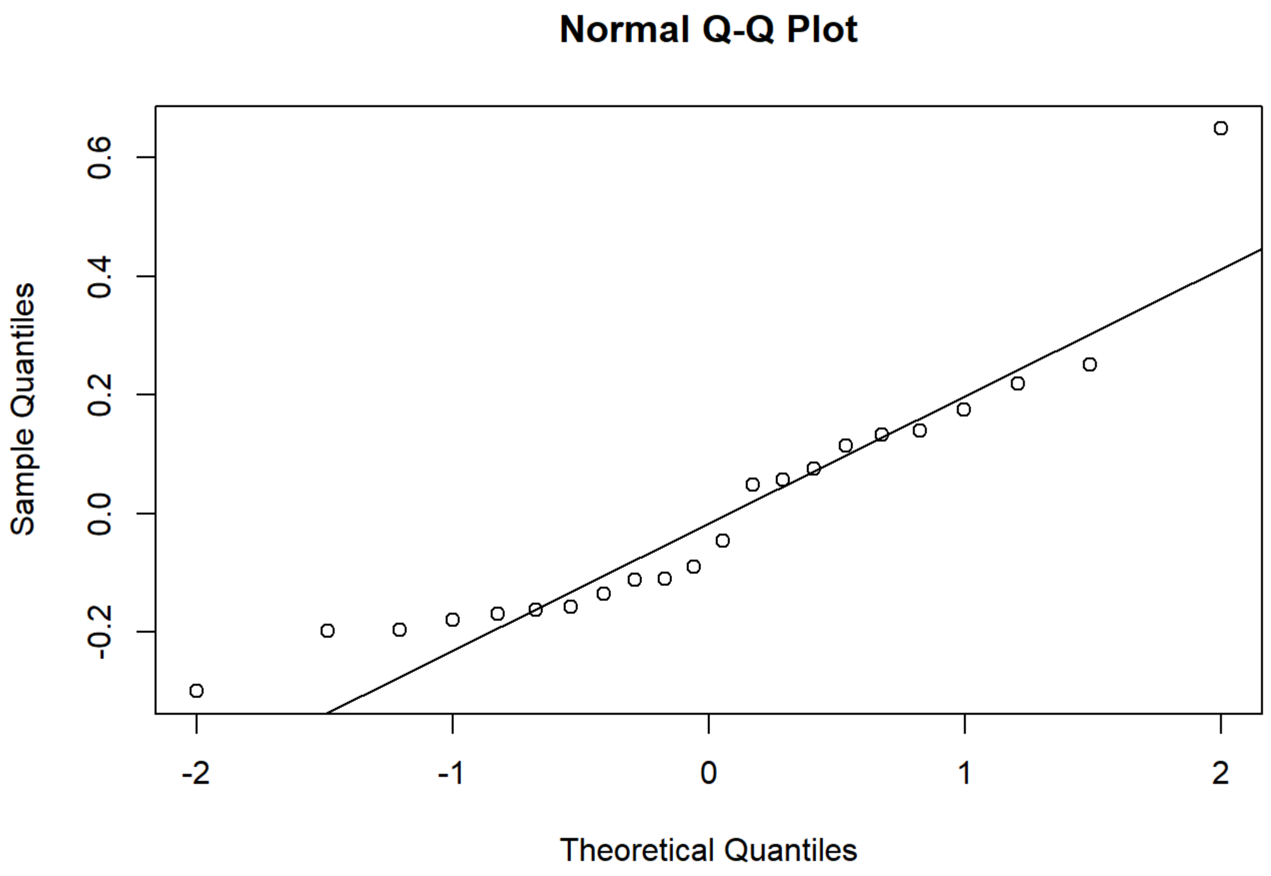
```
plot(m_criterion)
```



```
## Residuals are homogenic and the relationship is linear

qqnorm(residuals(m_criterion))
qqline(residuals(m_criterion))
```



```
## Residuals are approximately normal with symmetric deviations at the tails

qqnorm(ranef(m_criterion)$ParticipantID[[1]])
qqline(ranef(m_criterion)$ParticipantID[[1]])
```

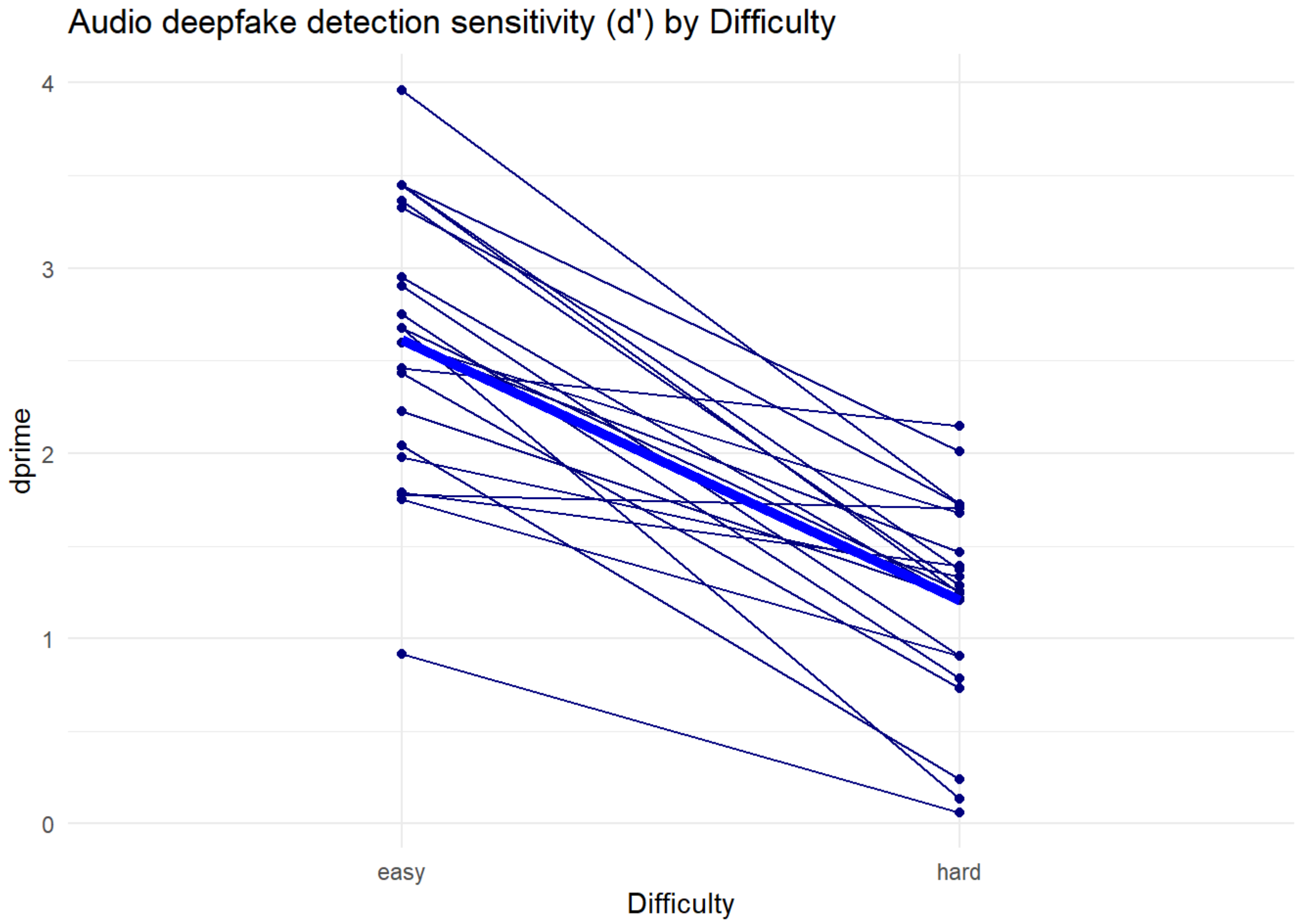


```
## Random effect per participant slightly deviate from normality, but the center is approxima
tely at the line
```

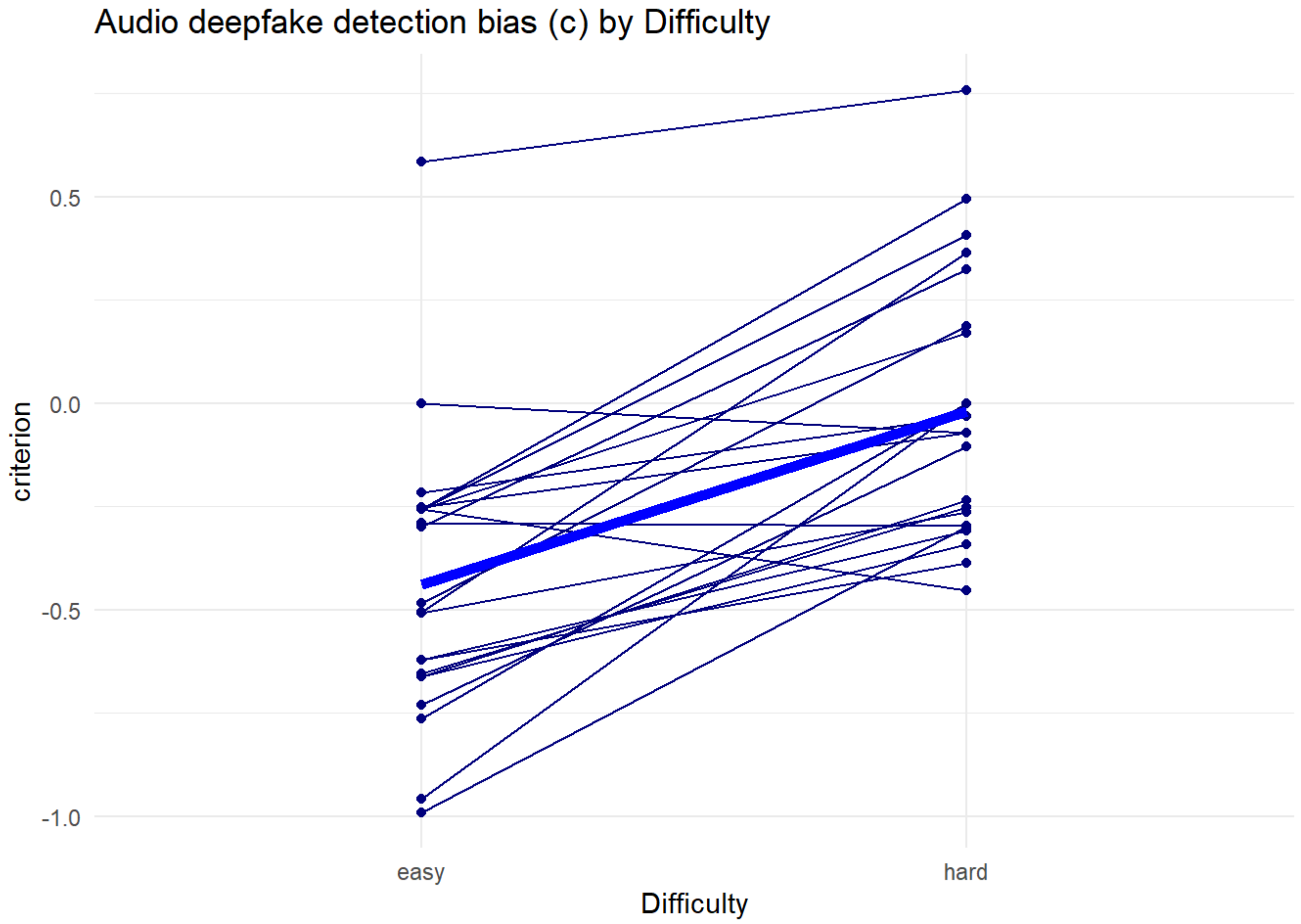
Report: The visual inspection of QQ-plots for residuals and random effects showed no significant deviation from normality.

Making plots

```
# Plot 1: d' by Difficulty

ggplot(data = sdt) +
  geom_point(aes(x = Difficulty , y = dprime), colour = "navy", show.legend = FALSE) +
  geom_line(aes(x = Difficulty , y = dprime, group = ParticipantID), colour = "navy", show.le
gend = FALSE) +
  geom_line(data = sdt_sum,aes(x = Difficulty , y = pop_dprime, group = NA), colour = "blue",
size = 2, show.legend = FALSE) +
  ggtitle("Audio deepfake detection sensitivity (d') by Difficulty") +
  theme_minimal()
```



```
# Plot 2: c by Difficulty

ggplot(data = sdt) +
  geom_point(aes(x = Difficulty , y = criterion), colour = "navy", show.legend = FALSE) +
  geom_line(aes(x = Difficulty , y = criterion, group = ParticipantID), colour = "navy", sho
w.legend = FALSE) +
  geom_line(data = sdt_sum,aes(x = Difficulty , y = pop_criterion, group = NA), colour = "blu
e", size = 2, show.legend = FALSE) +
  ggtitle("Audio deepfake detection bias (c) by Difficulty") +
  theme_minimal()
```

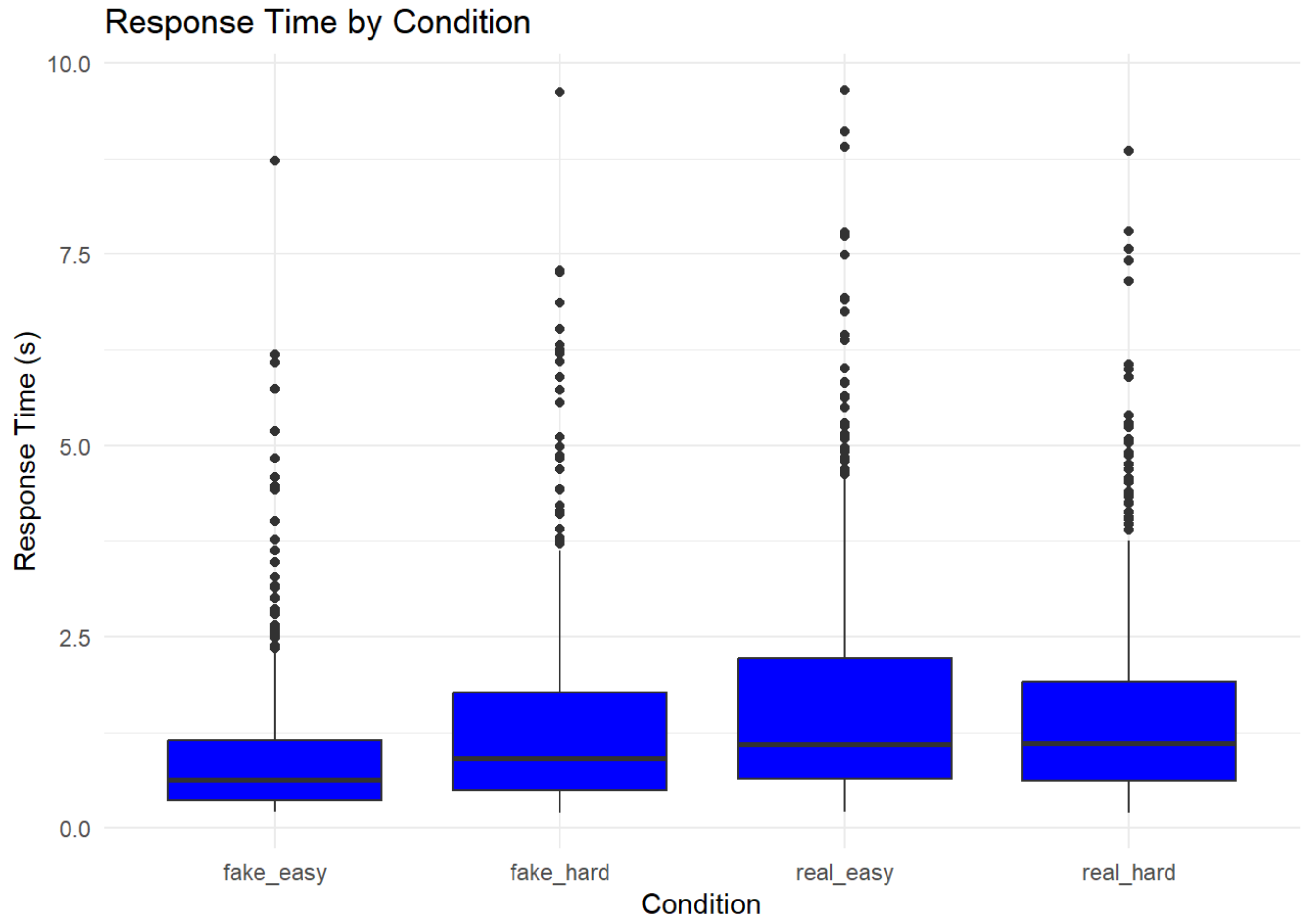


2. Behavioral Performance and Subjective Judgments Analyses

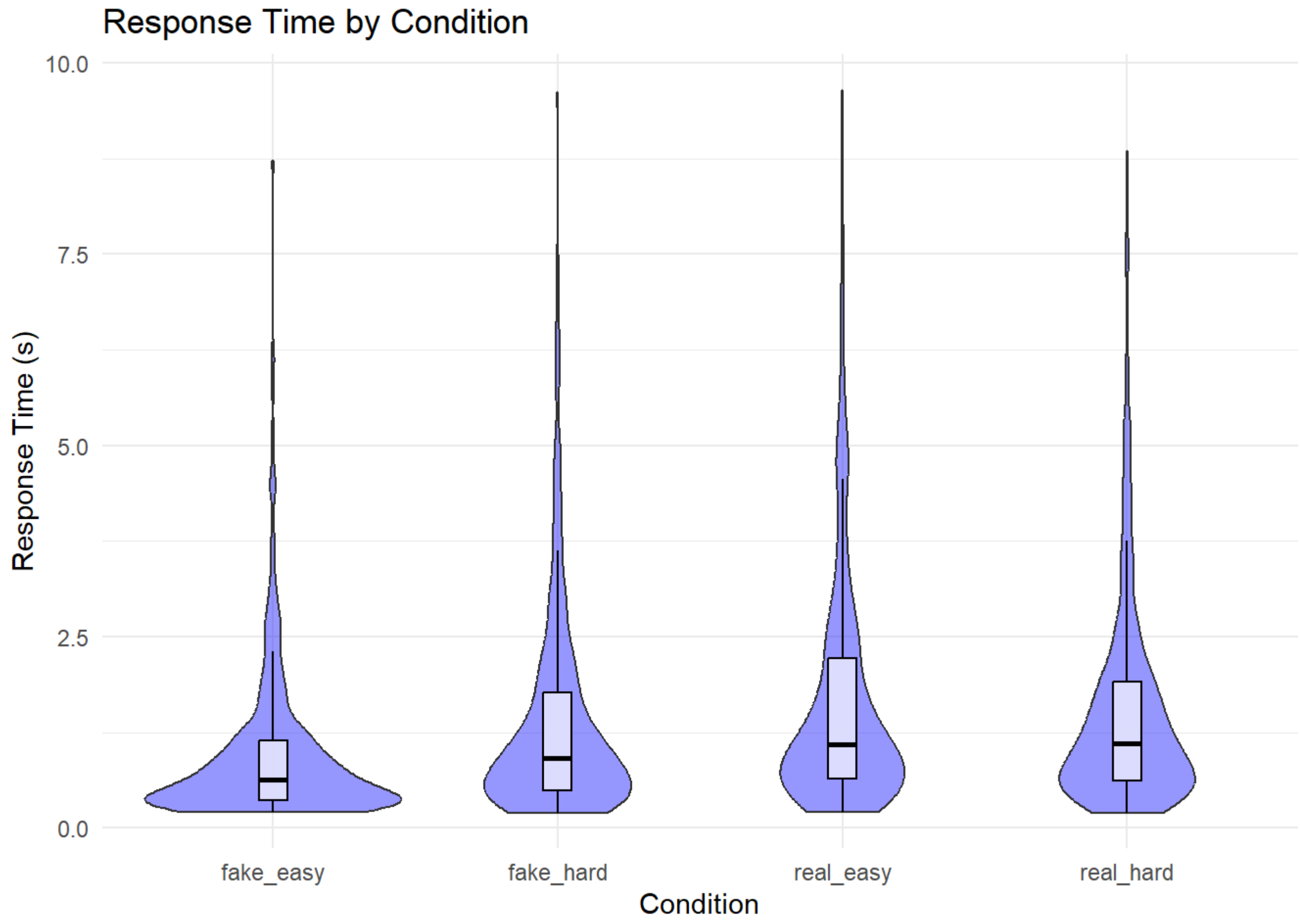
2.1. Response Time by Condition

Plot: ResponseTime by Condition

```
data %>%
  ggplot(aes(x = Condition, y = ResponseTime)) +
  geom_boxplot(fill = "blue") +
  labs(y = "Response Time (s)", title = "Response Time by Condition") +
  theme_minimal()
```



```
data %>%
  ggplot(aes(x = Condition, y = ResponseTime)) +
  geom_violin(trim = TRUE, alpha = 0.4, fill = "blue") +
  geom_boxplot(width = 0.1, color = "black", outlier.shape = NA, alpha = 0.7) +
    labs(x = "Condition", y = "Response Time (s)",
       title = "Response Time by Condition") +
  theme_minimal() +
  theme(legend.position = "none")
```



Modeling Response Time by Condition

```
m_rt_by_condition <- glmer(ResponseTime ~ Condition + (1|ParticipantID) + (1|Filename), data
= data, family = Gamma(link = "log"))
summary(m_rt_by_condition)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Gamma  ( log )
## Formula: ResponseTime ~ Condition + (1 | ParticipantID) + (1 | Filename)
##    Data: data
## 
##       AIC       BIC    logLik -2*log(L)  df.resid 
##    4106.9    4144.9   -2046.4    4092.9      1682 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -1.0207 -0.6572 -0.3285  0.2871  7.2658 
## 
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Filename      (Intercept) 0.03808  0.1951  
##  ParticipantID (Intercept) 0.05567  0.2359  
##  Residual                  0.78451  0.8857  
## Number of obs: 1689, groups:  Filename, 80; ParticipantID, 22
## 
## Fixed effects:
##                    Estimate Std. Error t value Pr(>|z|)    
## (Intercept)        -0.09521    0.08125  -1.172    0.241    
## Conditionfake_hard  0.37209    0.08300   4.483 7.36e-06 ***
## Conditionreal_easy  0.54957    0.08323   6.603 4.04e-11 ***
## Conditionreal_hard  0.46160    0.08312   5.553 2.80e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr) Cndtnf_ Cndtnrl_s
## Cndtnfk_hrd -0.513                  
## Condtnrl_sy -0.514  0.501           
## Cndtnrl_hrd -0.516  0.503   0.506   
```

```
m0 <- glmer(ResponseTime ~ (1|ParticipantID) + (1|Filename), data = data, family = Gamma(link
= "log"))

anova(m0, m_rt_by_condition)
```

	n... <dbl>	AIC <dbl>	BIC <dbl>	logLik <dbl>	-2*log(L) <dbl>	Chisq <dbl>	Df <dbl>	Pr(>Chis <d
m0	4	4139.059	4160.787	-2065.530	4131.059	NA	NA	
m_rt_by_condition	7	4106.891	4144.914	-2046.445	4092.891	38.16859	3	2.603456e-

2 rows

```
emmeans(m_rt_by_condition, pairwise ~ Condition, type = "response")
```

```
## $emmeans
##  Condition response     SE  df asymp.LCL asymp.UCL
##  fake_easy    0.909 0.0739 Inf     0.775      1.07
##  fake_hard    1.319 0.1070 Inf     1.125      1.55
##  real_easy    1.575 0.1280 Inf     1.344      1.85
##  real_hard    1.443 0.1170 Inf     1.231      1.69
## 
## Confidence level used: 0.95 
## Intervals are back-transformed from the log scale 
## 
## $contrasts
##  contrast              ratio     SE  df null z.ratio p.value
##  fake_easy / fake_hard 0.689 0.0572 Inf    1  -4.483 <0.0001
##  fake_easy / real_easy 0.577 0.0480 Inf    1  -6.603 <0.0001
##  fake_easy / real_hard 0.630 0.0524 Inf    1  -5.553 <0.0001
##  fake_hard / real_easy 0.837 0.0695 Inf    1  -2.138  0.1410
##  fake_hard / real_hard 0.914 0.0757 Inf    1  -1.081  0.7014
##  real_easy / real_hard 1.092 0.0903 Inf    1   1.064  0.7117
## 
## P value adjustment: tukey method for comparing a family of 4 estimates 
## Tests are performed on the log scale
```

```
print(paste("Difference between fake_easy and fake_hard:", round((1 - 0.689) * 100, 1),"% (p
< .0001)"))
```

```
## [1] "Difference between fake_easy and fake_hard: 31.1 % (p < .0001)"
```

```
print(paste("Difference between fake_easy and real_easy:", round((1 - 0.577) * 100, 1),"% (p
< .0001)"))
```

```
## [1] "Difference between fake_easy and real_easy: 42.3 % (p < .0001)"
```

```
print(paste("Difference between fake_easy and real_hard:", round((1 -  0.630) * 100, 1),"% (p
< .0001)"))
```

```
## [1] "Difference between fake_easy and real_hard: 37 % (p < .0001)"
```

```
print(paste("Difference between fake_hard and real_easy:", round((1 - 0.837) * 100, 1),"% (p
= .1410)"))
```

```
## [1] "Difference between fake_hard and real_easy: 16.3 % (p = .1410)"
```

```
print(paste("Difference between fake_hard and real_hard:", round((1 - 0.914) * 100, 1),"% (p
= .7014)"))
```

```
## [1] "Difference between fake_hard and real_hard: 8.6 % (p = .7014)"
```

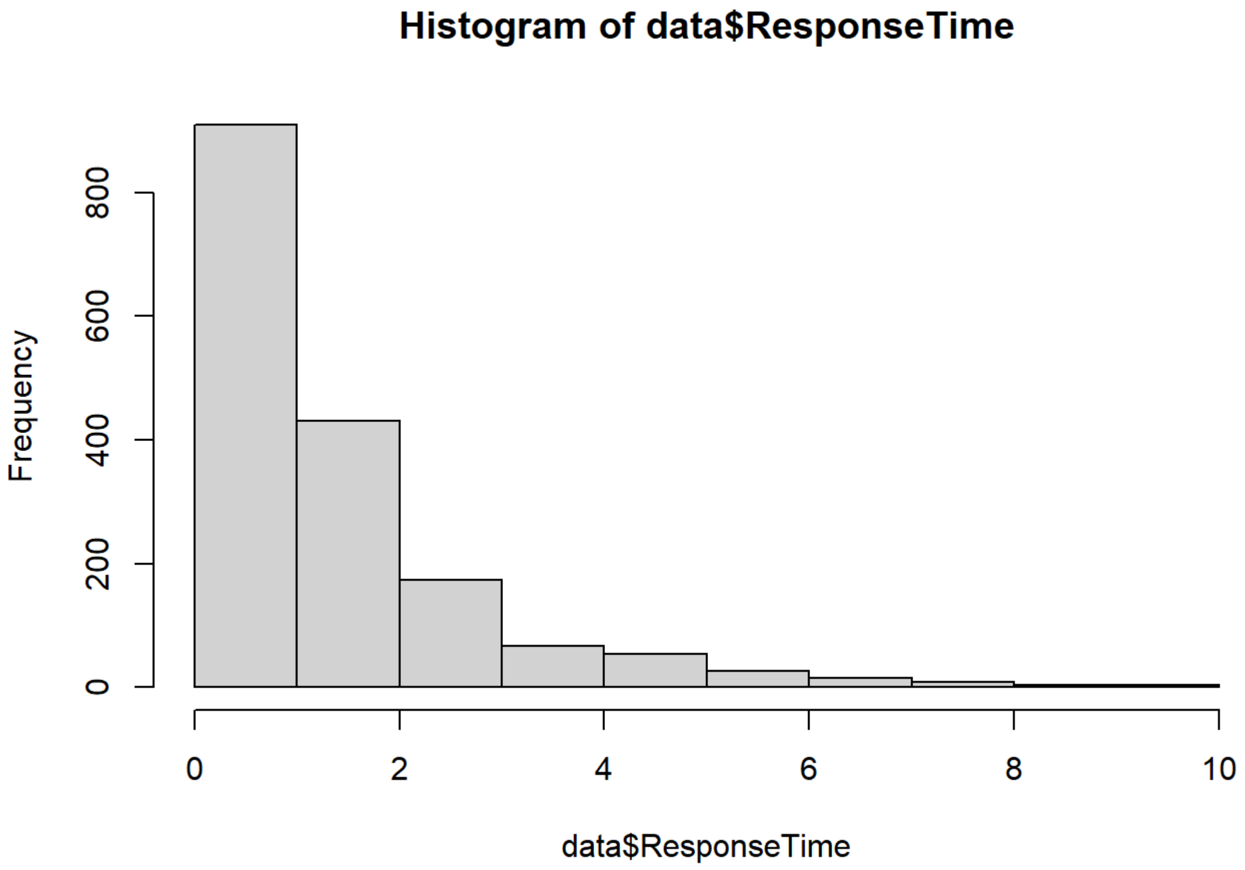
```
print(paste("Difference between real_easy and real_hard:", round((1 - 1.092) * 100, 1),"% (p
= .7117)"))
```

```
## [1] "Difference between real_easy and real_hard: -9.2 % (p = .7117)"
```

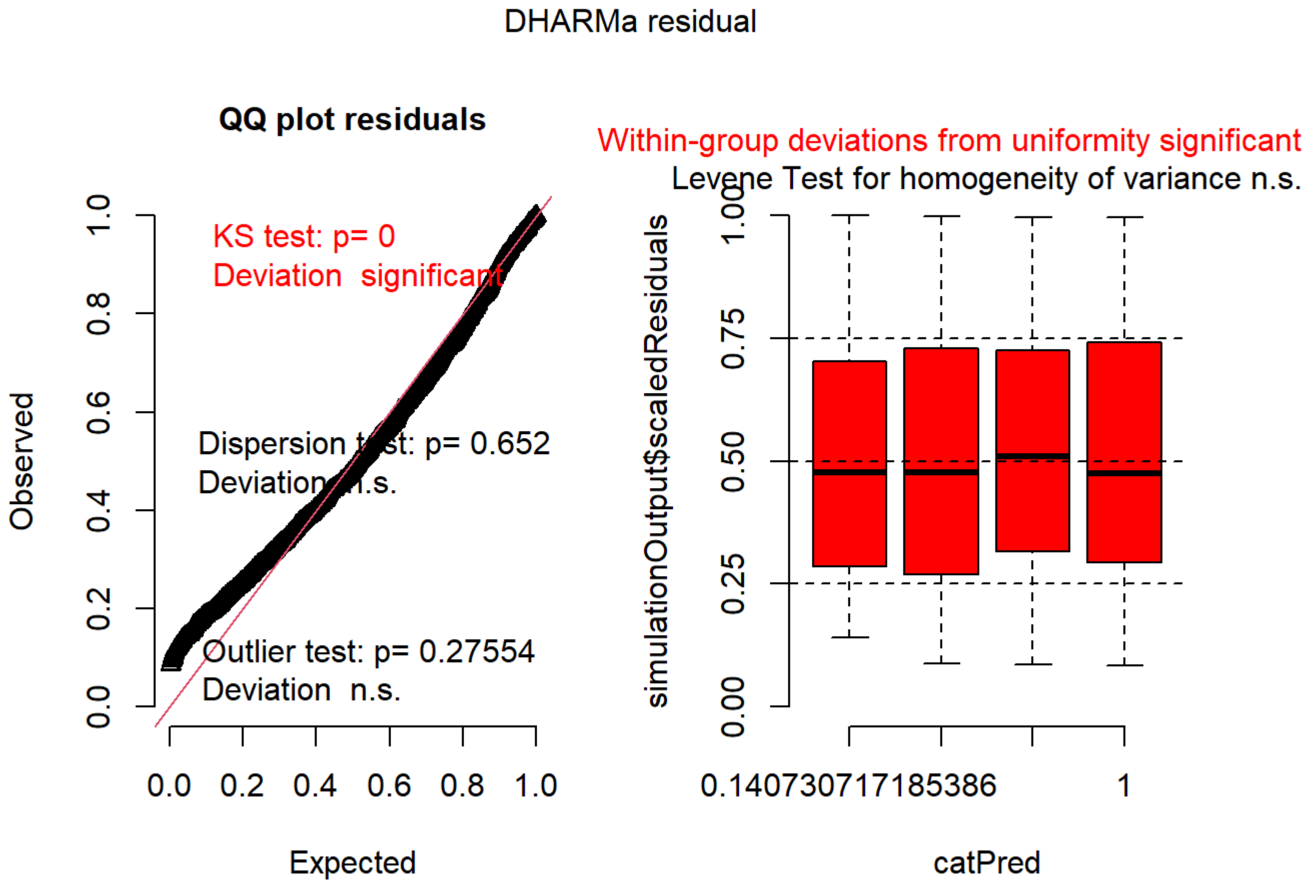
Report: To analyze the relationship between response times and Condition, a generalized linear mixed model was fit with a gamma distribution and a log link function, accounting for random intercepts for participants and stimuli. A likelihood-ratio test comparing the full model (random-effects and Condition) with a null model (random-effects only) showed that Condition significantly improved model fit ($\chi^2(3) = 38.17$, $p < .0001$). A post-hoc pairwise comparison of different conditions showed that response time for “fake_easy” is significantly different from all other conditions (all *p*s < .0001). Specifically, “fake_easy” (*M* = 0.909s) has by 31.1% shorter response time than “fake_hard” (*M* = 1.319s, *Ratio* = 0.689, *SE* = 0.057, *p* < .0001), 42.3% shorter response time than “real_easy” (*M* = 1.575s, *Ratio* = 0.577, *SE* = 0.048, *p* < .0001), and 37.0% shorter response time than “real_hard” (*M* = 1.443s, *Ratio* = 0.630, *SE* = 0.0524, *p* < .0001). All other comparisons were non-significant (all *p*s > .14). While the non-significant difference between “fake_hard” and both “real” conditions suggests that the “fake_hard” sounded realistic (as was intended), the non-significant difference between “real_hard” and “real_easy” suggests that the stimuli selection process failed to select the right stimuli. Furthermore, although non-significant, response time for “real_easy” (*M* = 1.575s) was actually by 9.2% higher than for “real_hard” (*M* = 1.443s, *Ratio* = 1.092, *SE* = 0.090, *p* = .7117), suggesting that our selection for “real_hard” and “real_easy” might have given opposite results than intended.

Checking assumptions

```
hist(data$ResponseTime)
```



```
# Checking for overdispersion

sim_resid <- simulateResiduals(fittedModel = m_rt_by_condition, n = 1000)

plot(sim_resid)
```



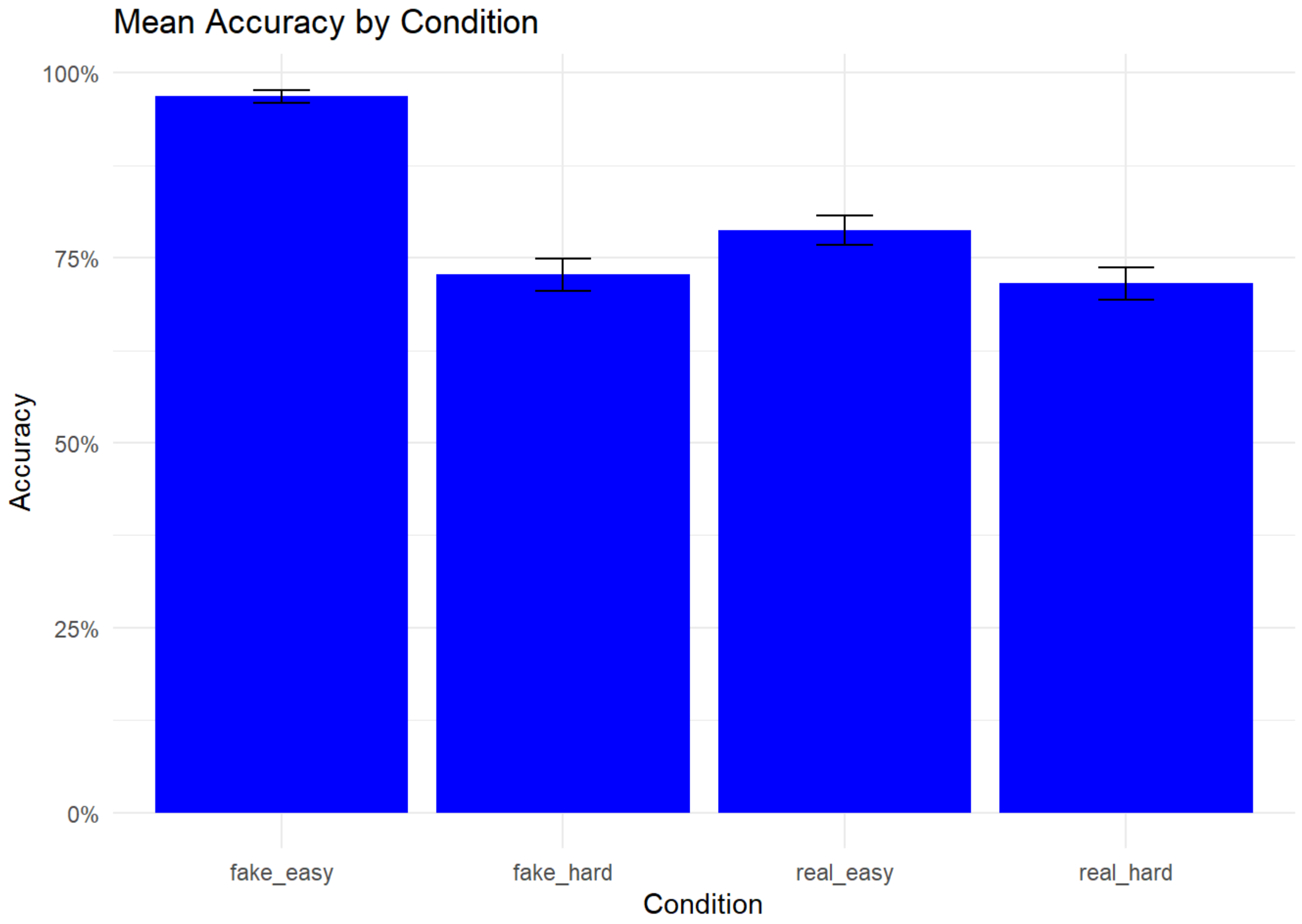
Report: Histogram of the raw response times shows that they are leftly skewed with a right tail, suggesting the use of Gamma distribution. Dispersion test for the model showed no overdispersion (*p* = 0.546).

2.2. Accuracy

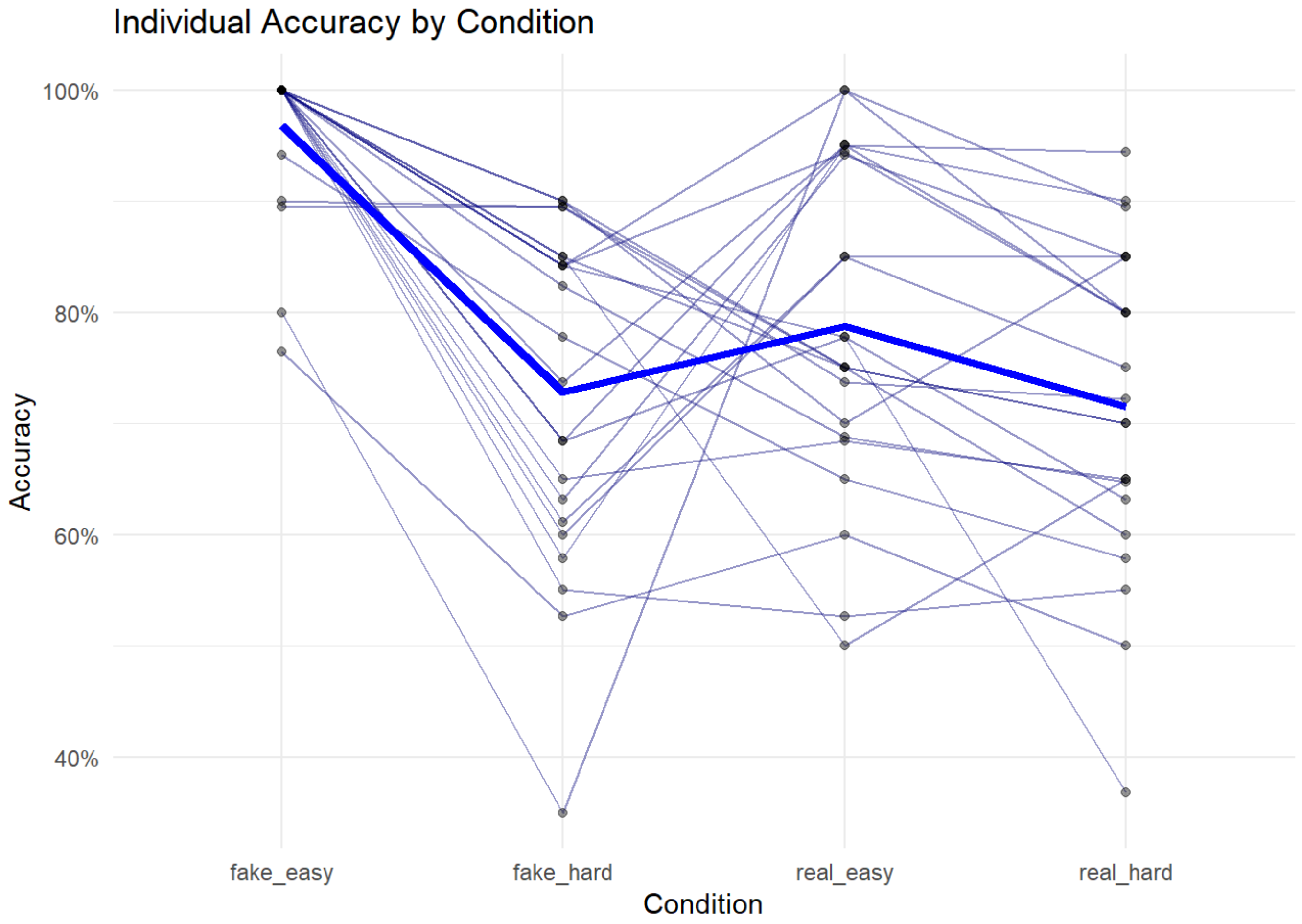
2.2.1. Accuracy by Condition

Plots: Accuracy by Condition

```
data %>%
  group_by(Condition) %>%
  summarise(accuracy = mean(Correct), se = sd(Correct)/sqrt(n())) %>%
  ggplot(aes(x = Condition, y = accuracy)) +
  geom_col(fill = "blue") + geom_errorbar(aes(ymin = accuracy - se, ymax = accuracy + se), wi
dth = .2) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(y = "Accuracy", title = "Mean Accuracy by Condition") +
  theme_minimal()
```



```
ggplot(data_acc_participants, aes(x = Condition, y = Accuracy, group = ParticipantID)) +
  geom_line(alpha = 0.4, color = "navy") +
  geom_point(alpha = 0.4) +
  stat_summary(aes(group = 1), fun = mean, geom = "line", size = 1.5, color = "blue") +
  scale_y_continuous(labels = scales::percent_format(Accuracy = 1)) +
  theme_minimal() +
  labs(title = "Individual Accuracy by Condition",y = "Accuracy", x = "Condition")
```



Modeling Accuracy by Condition

```
m_accuracy_by_condition <- glmer(Correct ~ Condition + (1|ParticipantID) + (1|Filename), fami
ly = binomial, glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 2e5)), data = data)

summary(m_accuracy_by_condition)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ Condition + (1 | ParticipantID) + (1 | Filename)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
## 
##       AIC       BIC    logLik -2*log(L)  df.resid 
##    1520.0    1552.6    -754.0    1508.0      1683 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -5.8856  0.1357  0.3640  0.5371  1.3162 
## 
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Filename      (Intercept) 0.3832   0.6191  
##  ParticipantID (Intercept) 0.2068   0.4548  
## Number of obs: 1689, groups:  Filename, 80; ParticipantID, 22
## 
## Fixed effects:
##                    Estimate Std. Error z value Pr(>|z|)    
## (Intercept)          3.6718     0.3324  11.047  < 2e-16 ***
## Conditionfake_hard  -2.5508     0.3632  -7.023 2.17e-12 ***
## Conditionreal_easy  -2.2169     0.3657  -6.062 1.35e-09 ***
## Conditionreal_hard  -2.6437     0.3627  -7.290 3.10e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr) Cndtnf_ Cndtnrl_s
## Cndtnfk_hrd -0.826                  
## Condtnrl_sy -0.820  0.745           
## Cndtnrl_hrd -0.831  0.753   0.748   
```

```
m0_accuracy <- glmer(Correct ~ (1|ParticipantID) + (1|Filename), data = data, family = binomial, glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 2e5)))

anova(m0_accuracy, m_accuracy_by_condition)
```

	n... <dbl>	AIC <dbl>	BIC <dbl>	logLik <dbl>	-2*log(L) <dbl>	Chisq <dbl>	Df <dbl>	
m0_accuracy	3	1574.907	1591.203	-784.4536	1568.907	NA	NA	
m_accuracy_by_condition	6	1520.023	1552.615	-754.0117	1508.023	60.88386	3	3.8

2 rows

```
acc_means <- emmeans(m_accuracy_by_condition, "Condition")

pairs(acc_means, adjust = "bonferroni")
```

```
##  contrast              estimate    SE  df z.ratio p.value
##  fake_easy - fake_hard    2.551 0.363 Inf   7.023 <0.0001
##  fake_easy - real_easy    2.217 0.366 Inf   6.062 <0.0001
##  fake_easy - real_hard    2.644 0.363 Inf   7.290 <0.0001
##  fake_hard - real_easy   -0.334 0.260 Inf  -1.284  1.0000
##  fake_hard - real_hard    0.093 0.255 Inf   0.365  1.0000
##  real_easy - real_hard    0.427 0.258 Inf   1.651  0.5920
## 
## Results are given on the log odds ratio (not the response) scale. 
## P value adjustment: bonferroni method for 6 tests
```

```
print(paste("Probability of being correct for fake_easy:", round(plogis(3.6718) * 100,2),
"%"))
```

```
## [1] "Probability of being correct for fake_easy: 97.52 %"
```

```
print(paste("Probability of being correct for fake_hard:", round(plogis(1.121) * 100,2),
"%"))
```

```
## [1] "Probability of being correct for fake_hard: 75.42 %"
```

```
print(paste("Probability of being correct for real_easy:", round(plogis(1.4549)* 100,2),
"%"))
```

```
## [1] "Probability of being correct for real_easy: 81.08 %"
```

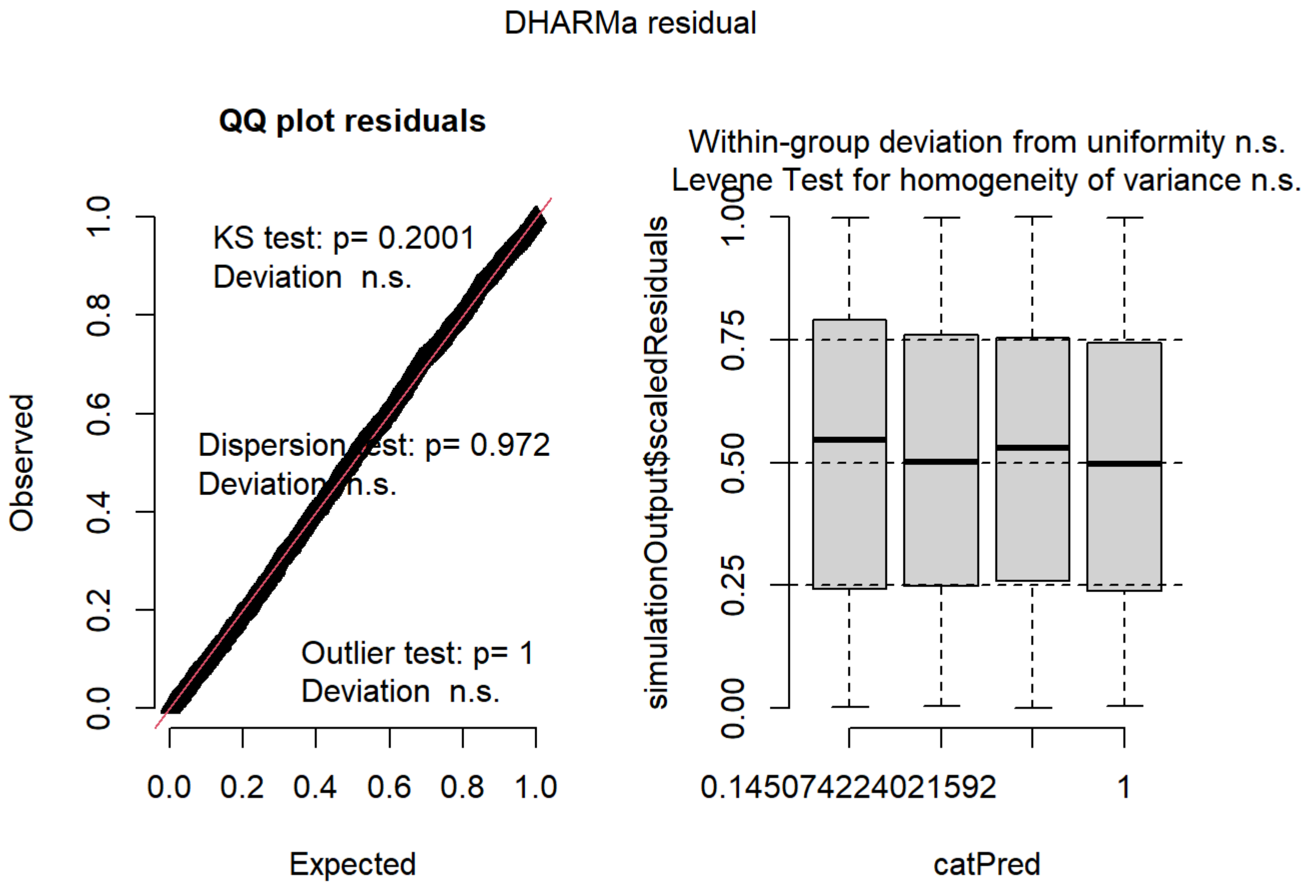
```
print(paste("Probability of being correct for real_hard:", round(plogis(1.0281) * 100,2),
"%"))
```

```
## [1] "Probability of being correct for real_hard: 73.65 %"
```

Report: A generalized linear mixed-effects model was fit to predict accuracy by condition as fixed effect and random intercepts for participants and stimuli. A likelihood-ratio test comparing the full model (random-effects and Condition) with a null model (random-effects only) showed that Condition significantly improved model fit ($\chi^2(3) = 60.88$, $p < .0001$). Then, post-hoc pairwise comparison showed that within the fake stimuli, the probability of being correct was reduced from 98% in the easy condition to 75% in the hard condition ($b = -2.55$, $SE = 0.36$, $z = -7.02$, $p < .0001$). The difference between fake_easy and real_easy is 17 percent points (81%, $b = -2.22$, $SE = 0.37$, $z = -6.06$, $p < .0001$), and between fake_easy and real_hard is 24 percent points (74%, $b = -2.64$, $SE = 0.36$, $z = -7.29$, $p < .0001$). The difference between fake_hard - real_easy ($b = -0.33$, $SE = 0.26$, $z = -1.28$, $p = 1$) and fake_hard - real_hard ($b = 0.09$, $SE = 0.26$, $z = 0.365$, $p = 1$) was small and non-significant. The difference between the difficulty of real stimuli was also not significant with the difference in 7 percent points ($b = 0.43$, $SE = 0.26$, $z = 1.65$, $p = .59$).

Checking for overdispersion

```
sim_resid <- simulateResiduals(fittedModel = m_accuracy_by_condition, n = 1000)

plot(sim_resid)
```

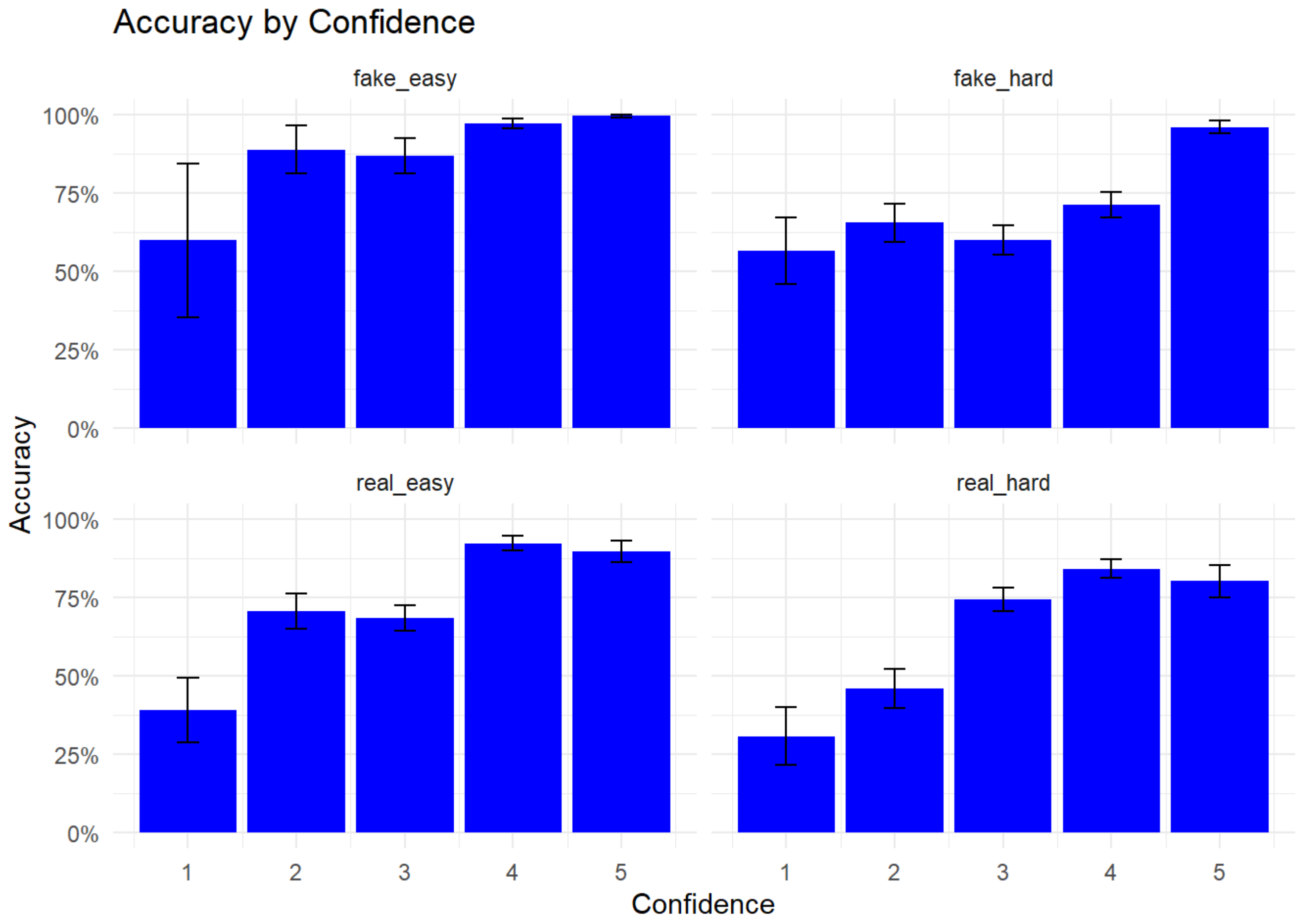


Report: Dispersion test showed no overdispersion (*p* = 0.968).

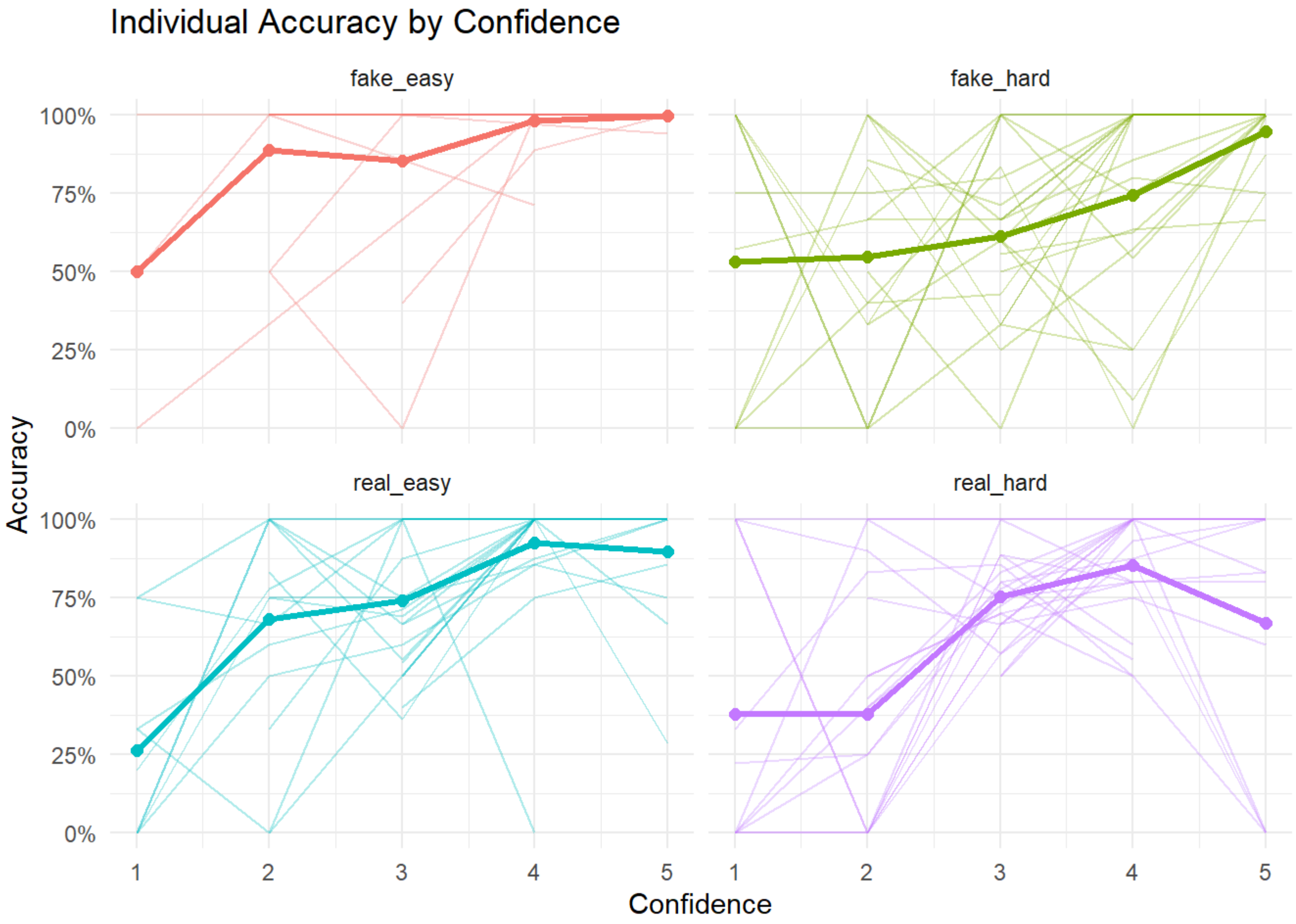
2.2.2. Accuracy by Confidence

Plot: Accuracy by Confidence

```
data %>%
  group_by(Confidence, Condition) %>%
  summarise(accuracy = mean(Correct), se = sd(Correct)/sqrt(n())) %>%
  ggplot(aes(x = Confidence, y = accuracy)) +
  geom_col(fill = "blue") + geom_errorbar(aes(ymin = accuracy - se, ymax = accuracy + se), wi
dth = .2) +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(y = "Accuracy", title = "Accuracy by Confidence") +
  theme_minimal() +
  facet_wrap(~Condition)
```



```
data_acc_conf_participants <- data %>%
  group_by(Condition, ParticipantID, Confidence) %>%
  summarise(Accuracy = mean(Correct), .groups = "drop")

ggplot(data_acc_conf_participants, aes(x = Confidence, y = Accuracy, color = Condition)) +
  geom_line(aes(group = ParticipantID), alpha = 0.3, show.legend = FALSE) +
  stat_summary(fun = mean, geom = "line", size = 1.2, show.legend = FALSE) +
  stat_summary(fun = mean, geom = "point", size = 2, show.legend = FALSE) +
  facet_wrap(~Condition) +
  scale_y_continuous(labels = scales::percent) +
  theme_minimal() +
  labs(title = "Individual Accuracy by Confidence")
```



Modelling Accuracy by Confidence

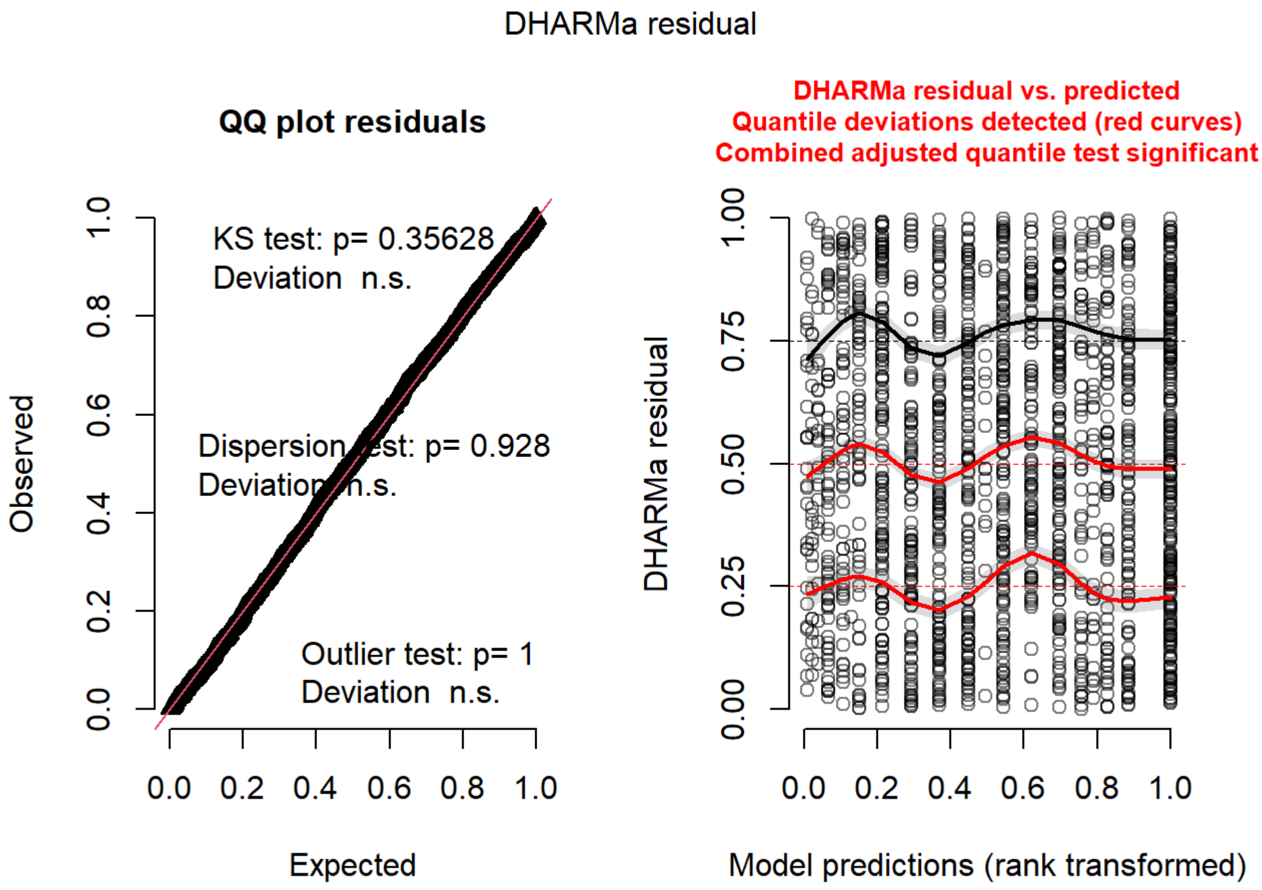
```
m_accuracy_by_confidence <- glmer(Correct ~ Confidence_z * Condition + (1|ParticipantID) + (1
|Filename), family = binomial, glmerControl(optimizer="bobyqa", optCtrl = list(maxfun = 2e
5)), data = data)

summary(m_accuracy_by_confidence)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: Correct ~ Confidence_z * Condition + (1 | ParticipantID) + (1 |
##     Filename)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 2e+05))
##
##       AIC       BIC    logLik -2*log(L)  df.resid
##    1407.9    1462.2    -694.0    1387.9      1679
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -12.1142   0.0741   0.2823   0.4902   2.5912
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Filename      (Intercept) 0.3086   0.5555
##  ParticipantID (Intercept) 0.1983   0.4453
## Number of obs: 1689, groups:  Filename, 80; ParticipantID, 22
##
## Fixed effects:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       3.5184     0.3669   9.591  < 2e-16 ***
## Confidence_z                      1.3364     0.2869   4.658 3.19e-06 ***
## Conditionfake_hard               -2.2537     0.3938  -5.723 1.05e-08 ***
## Conditionreal_easy               -1.7218     0.4026  -4.276 1.90e-05 ***
## Conditionreal_hard               -2.1449     0.3973  -5.399 6.71e-08 ***
## Confidence_z:Conditionfake_hard  -0.7244     0.3113  -2.327   0.0199 *
## Confidence_z:Conditionreal_easy  -0.4507     0.3202  -1.407   0.1593
## Confidence_z:Conditionreal_hard  -0.4658     0.3172  -1.469   0.1419
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                    (Intr) Cnfdn_ Cndtnf_ Cndtnrl_s Cndtnrl_h Cnfdnc_z:Cndtnf_
## Confidenc_z         0.322
## Cndtnfk_hrd        -0.858 -0.296
## Condtnrl_sy        -0.838 -0.289  0.778
## Cndtnrl_hrd        -0.853 -0.293  0.790   0.772
## Cnfdnc_z:Cndtnf_   -0.296 -0.915  0.317   0.270     0.274
## Cnfdnc_z:Cndtnrl_s -0.282 -0.889  0.264   0.346     0.263     0.825
## Cnfdnc_z:Cndtnrl_h -0.285 -0.898  0.267   0.262     0.338     0.832
##                    Cnfdnc_z:Cndtnrl_s
## Confidenc_z
## Cndtnfk_hrd
## Condtnrl_sy
## Cndtnrl_hrd
## Cnfdnc_z:Cndtnf_
## Cnfdnc_z:Cndtnrl_s
## Cnfdnc_z:Cndtnrl_h  0.813
```

Report: To see how Confidence and its interaction with Condition predicts Accuracy, a generalized linear mixed-effects model was fit, with random intercepts for participants and stimuli. With “fake_easy” as intercept, the results show a significant main effect of Confidence (*b* = 1.33, *SE* = 0.29, *z* = 4.66, *p* < .0001), implying that generally higher Confidence resulted in higher probability of being Correct. For all other conditions, the effect of Condition on Accuracy was significantly lower than the baseline (all *p*s < .0001). Furthermore, only one of the interaction effects was significant, namely, the negative interaction between Confidence and “fake_hard” condition (*b* = -0.72, *SE* = 0.31, *z* = -2.33, *p* = .02), suggesting that while the other conditions did not change the relationship between Confidence and Accuracy, “fake_hard” condition weakened it. Finally, the results also showed that a random intercept for stimuli explains more variance than a random intercept for participants, suggesting that the different acoustic features have an effect on accuracy.

Checking for overdispersion

```
sim_resid <- simulateResiduals(fittedModel = m_accuracy_by_confidence, n = 1000)

plot(sim_resid)
```

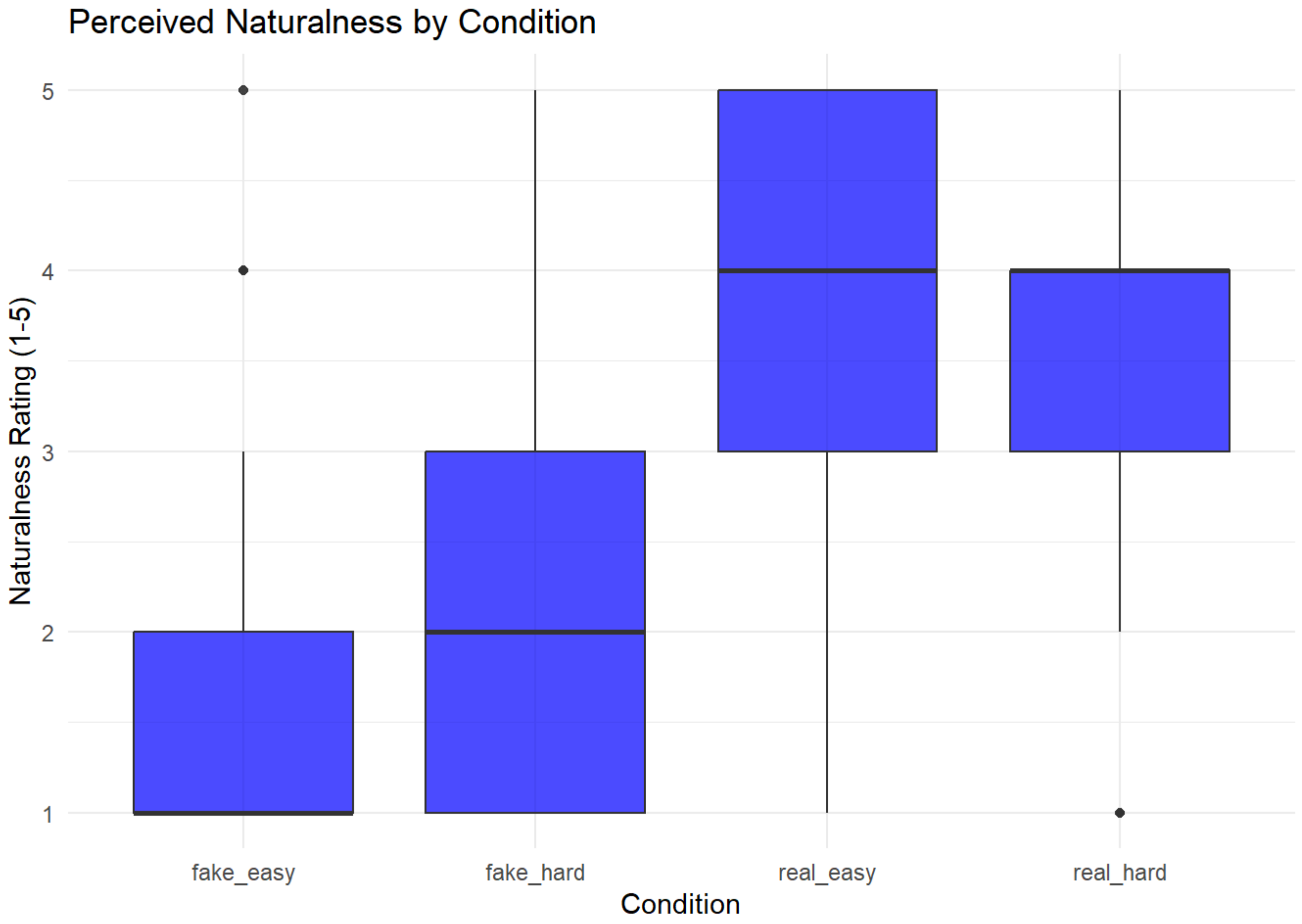


Report: Dispersion test showed no overdispersion (*p* = 0.918).

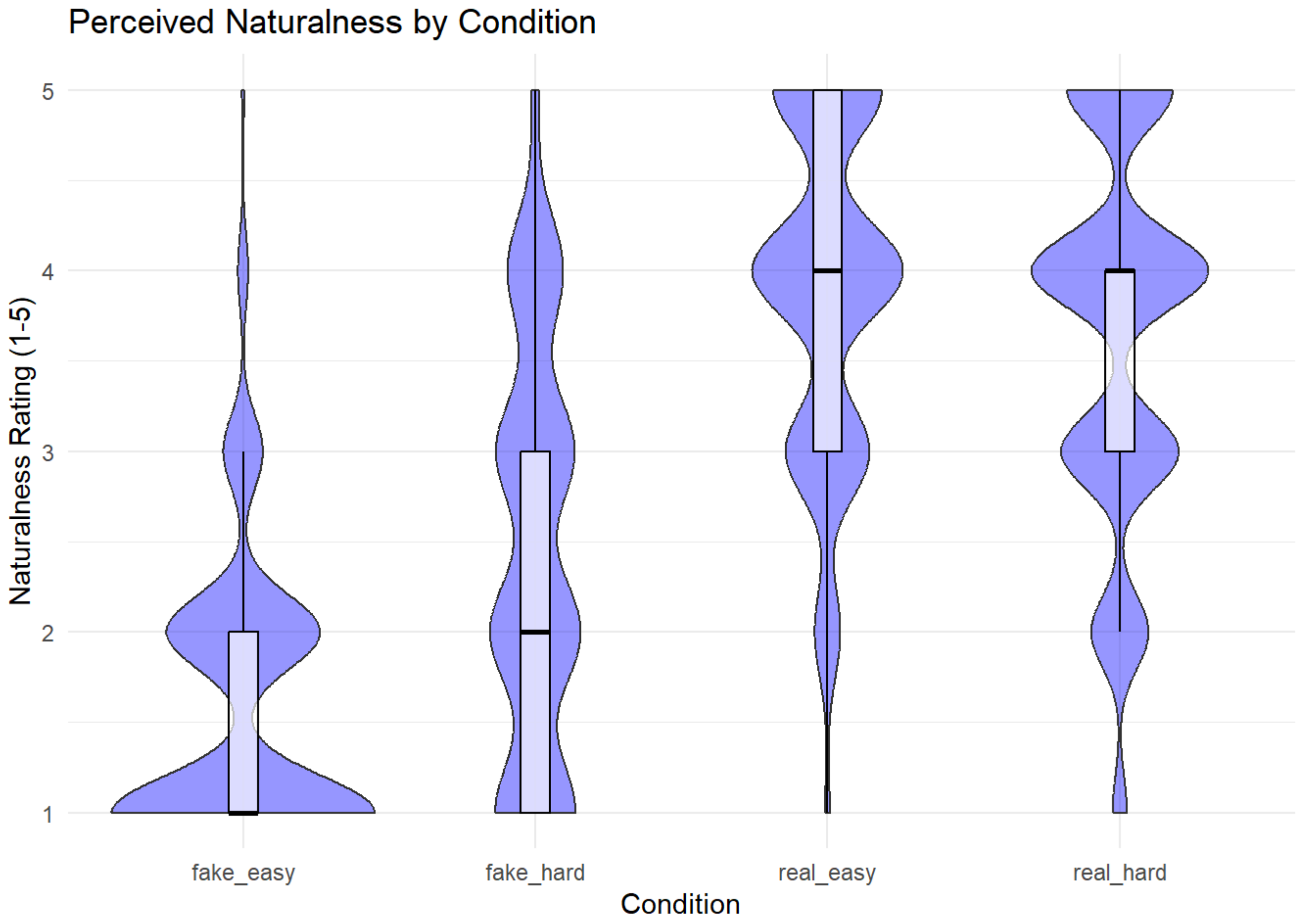
2.3. Naturalness by Condition

Plotting

```
data %>%
  ggplot(aes(x = Condition, y = Naturalness)) +
  geom_boxplot(alpha = 0.7, fill = "blue") +
  coord_cartesian(ylim = c(1, 5)) +
  labs(x = "Condition", y = "Naturalness Rating (1-5)",
       title = "Perceived Naturalness by Condition") +
  theme_minimal() +
  theme(legend.position = "none")
```



```
data %>%
  ggplot(aes(x = Condition, y = Naturalness)) +
  geom_violin(trim = TRUE, alpha = 0.4, fill = "blue") +
  geom_boxplot(width = 0.1, color = "black", outlier.shape = NA, alpha = 0.7) +
    labs(x = "Condition", y = "Naturalness Rating (1-5)",
       title = "Perceived Naturalness by Condition") +
  theme_minimal() +
  theme(legend.position = "none")
```



Modelling

```
m_naturalness_by_condition <- lmer(Naturalness ~ Condition + (1 | ParticipantID) + (1 | Filename), data = data)

summary(m_naturalness_by_condition)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Naturalness ~ Condition + (1 | ParticipantID) + (1 | Filename)
##    Data: data
##
## REML criterion at convergence: 4558
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.3817 -0.6466 -0.0076  0.6727  3.7986
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Filename      (Intercept) 0.08683  0.2947
##  ParticipantID (Intercept) 0.09464  0.3076
##  Residual                  0.79459  0.8914
## Number of obs: 1689, groups:  Filename, 80; ParticipantID, 22
##
## Fixed effects:
##                     Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)           1.5725     0.1028 69.9931  15.298  < 2e-16 ***
## Conditionfake_hard    0.8572     0.1117 76.6037   7.671 4.48e-11 ***
## Conditionreal_easy    2.3230     0.1117 76.5748  20.789  < 2e-16 ***
## Conditionreal_hard    2.0686     0.1116 76.2392  18.534  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Cndtnf_ Cndtnrl_s
## Cndtnfk_hrd -0.545
## Condtnrl_sy -0.545  0.501
## Cndtnrl_hrd -0.546  0.502   0.502
```

```
m0_naturalness <- lmer(Naturalness ~ (1|ParticipantID) + (1|Filename), data = data)

anova(m0_naturalness, m_naturalness_by_condition)
```

```
## refitting model(s) with ML (instead of REML)
```

	n…	AIC	BIC	logLik	-2*log(L)	Chisq	Df
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
m0_naturalness	4	4724.013	4745.740	-2358.006	4716.013	NA	NA
m_naturalness_by_condition	7	4560.324	4598.347	-2273.162	4546.324	169.689	3

2 rows

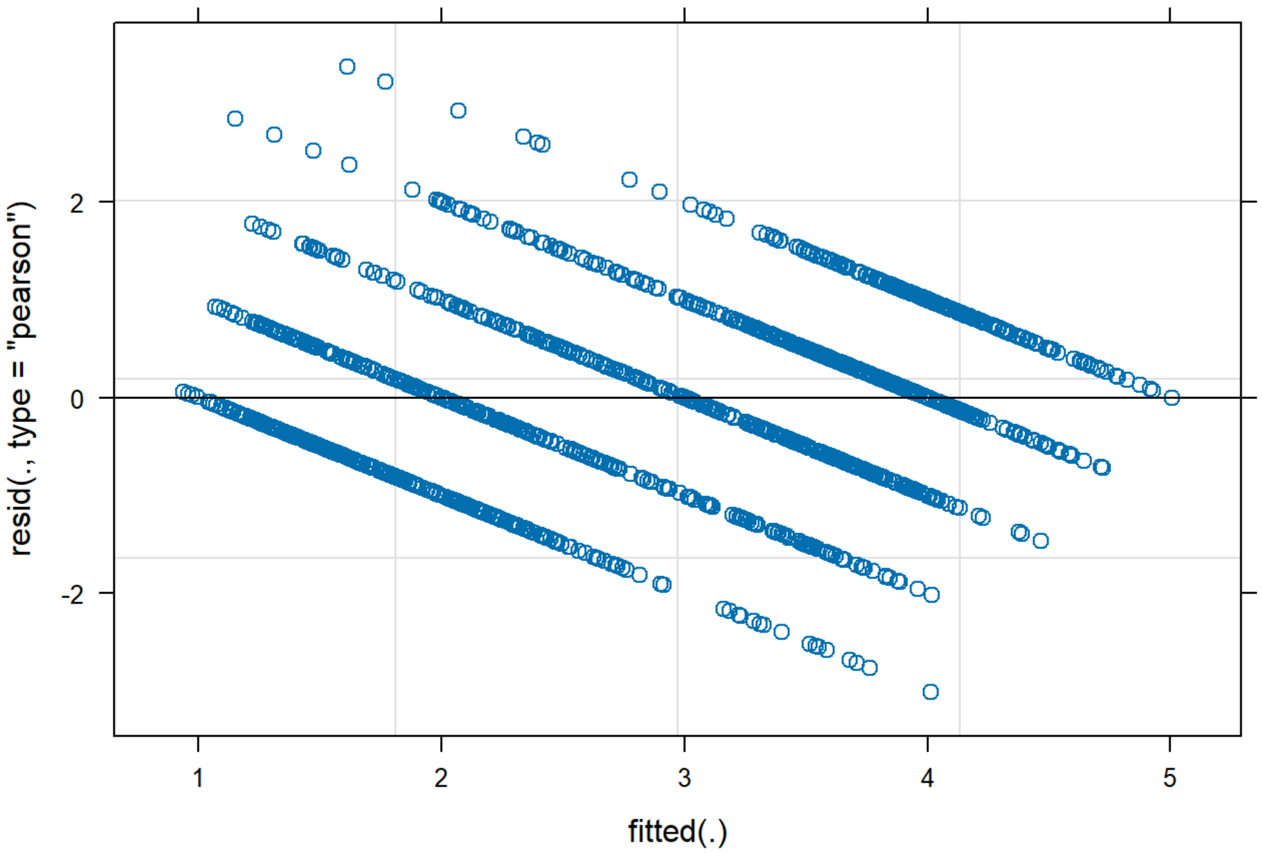
```
nat_means <- emmeans(m_naturalness_by_condition, "Condition")

pairs(nat_means, adjust = "bonferroni")
```

```
##  contrast              estimate    SE   df t.ratio p.value
##  fake_easy - fake_hard   -0.857 0.112 76.4  -7.671 <0.0001
##  fake_easy - real_easy   -2.323 0.112 76.4 -20.789 <0.0001
##  fake_easy - real_hard   -2.069 0.112 76.0 -18.534 <0.0001
##  fake_hard - real_easy   -1.466 0.112 76.0 -13.136 <0.0001
##  fake_hard - real_hard   -1.211 0.111 75.6 -10.870 <0.0001
##  real_easy - real_hard    0.254 0.111 75.6   2.282  0.1518
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: bonferroni method for 6 tests
```

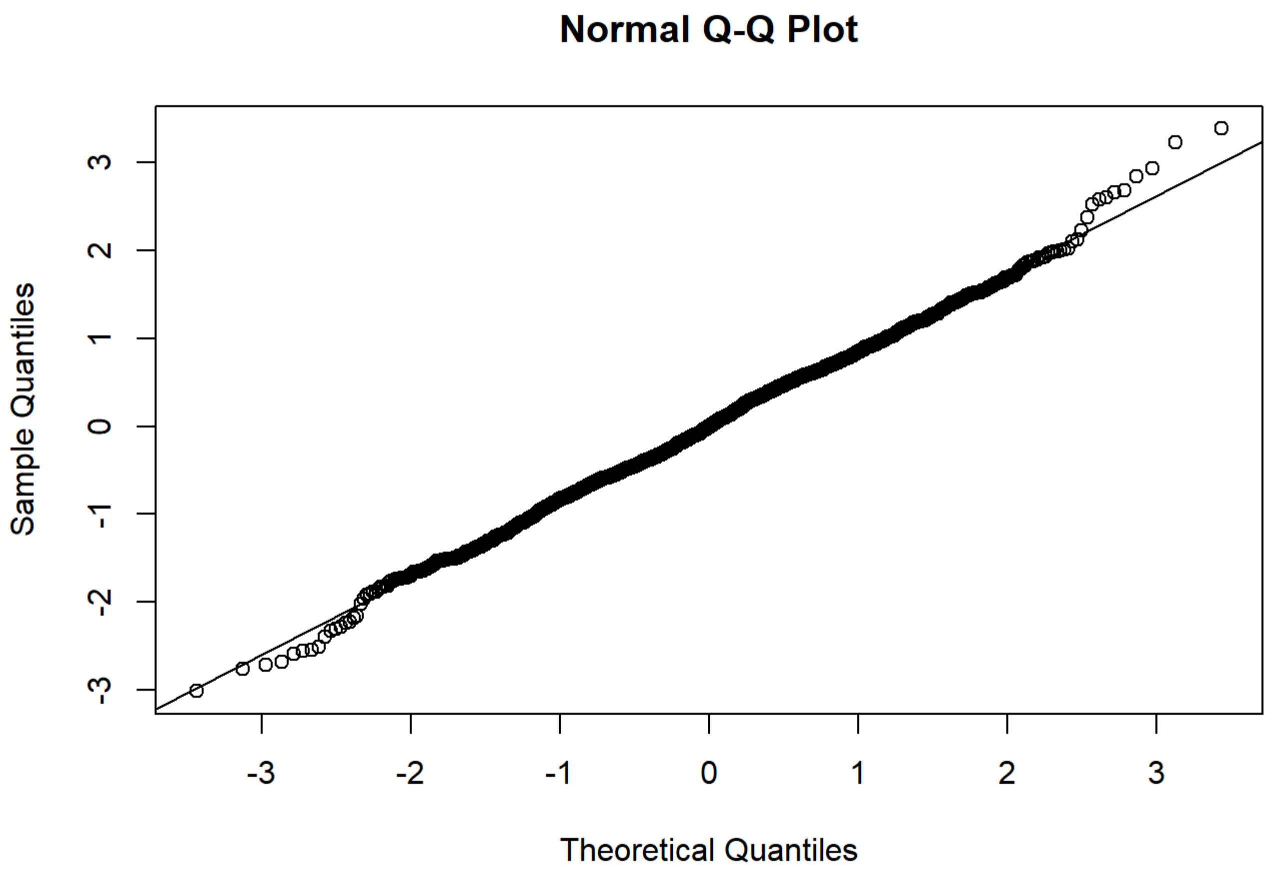
Report: To analyze whether our construction of Condition can predict the subjective rating of Naturalness, a linear mixed-effects model was fit, with Condition as fixed effect and random intercepts for participants and stimuli. A likelihood-ratio test comparing the full model (random-effects and Condition) with a null model (random-effects only) showed that Condition significantly improved model fit ($\chi^2(3) = 169.69$, *p* < .0001). The post-hoc pairwise comparisons showed that, all conditions were significantly different in mean Naturalness scores, except for the “real” conditions. Specifically, the mean Naturalness score for “fake_easy” (*M* = 1.57) was significantly lower compared to all other conditions (all *p*s < .0001). The highest mean Naturalness score for “real_easy” (*M* = 3.89) was significantly higher than the score for “fake_hard” (*b* = 1.466, *SE* = 0.112, *p* < .0001), but non-significantly higher than “real_hard” (*b* = 0.254, *SE* = 0.111, *p* = 0.1518), indicating that “fake_hard” still sounded less natural than human voices, while the human voices in themselves did not differ significantly in perceived Naturalness. Notably, even for “real_easy” the mean Naturalness score was only 3.89 out of 5 (where 5 is “very natural”), indicating that even theoretically the most natural sounding stimuli were perceived as somewhat ambiguously natural. This might might be explained by the bias towards “fake” in the “easy” conditions (which was shown earlier in SDT analysis).

Checking assumptions

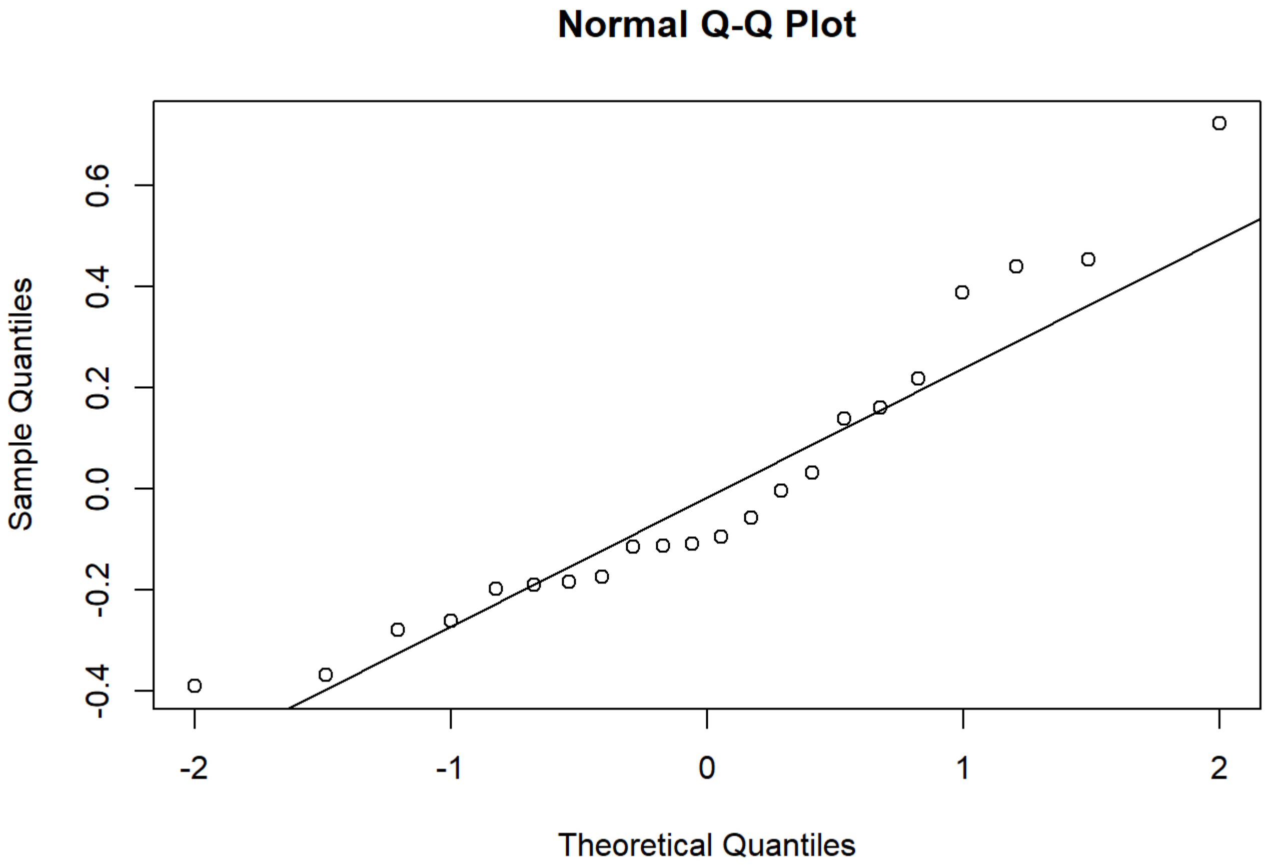
```
plot(m_naturalness_by_condition)
```



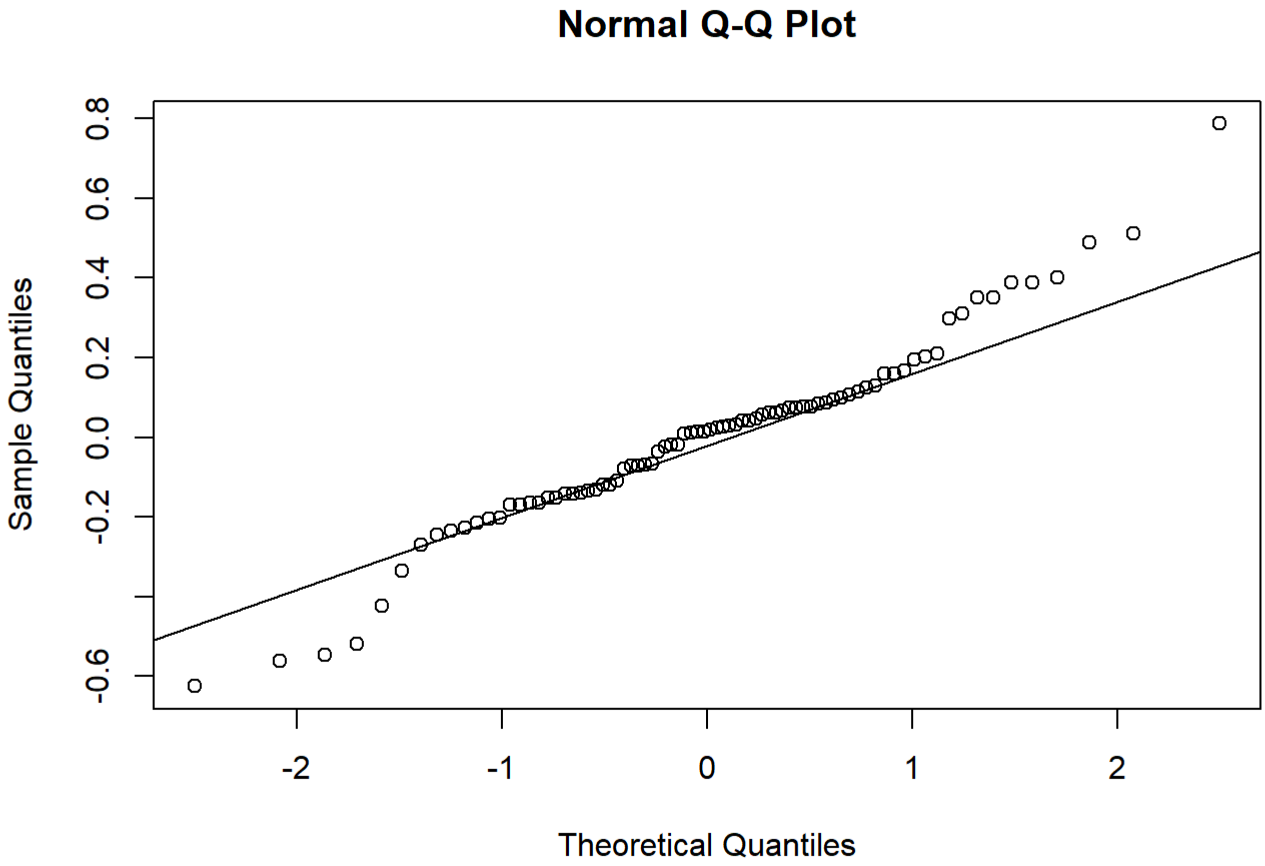
```
    ## Residuals are homogenic and the relationship is linear (for ordinal data)

qqnorm(residuals(m_naturalness_by_condition))
qqline(residuals(m_naturalness_by_condition))
```



```
    ## Residuals are approximately normal with symmetric deviations at the tails

qqnorm(ranef(m_naturalness_by_condition)$ParticipantID[[1]])
qqline(ranef(m_naturalness_by_condition)$ParticipantID[[1]])
```



```
    ## Random effect per participant somewhat deviates from normality, but given the robustne
ss of the model, it should still be acceptable

qqnorm(ranef(m_naturalness_by_condition)$Filename[[1]])
qqline(ranef(m_naturalness_by_condition)$Filename[[1]])
```



```
    ## Random effect per filename slightly deviates from normality, but given the robustness
of the model, it should still be acceptable
```

Report: The visual inspection of QQ-plots for residuals and random effects, showed some deviations from normality, which were nevertheless deemed acceptable, given the robustness of the model.

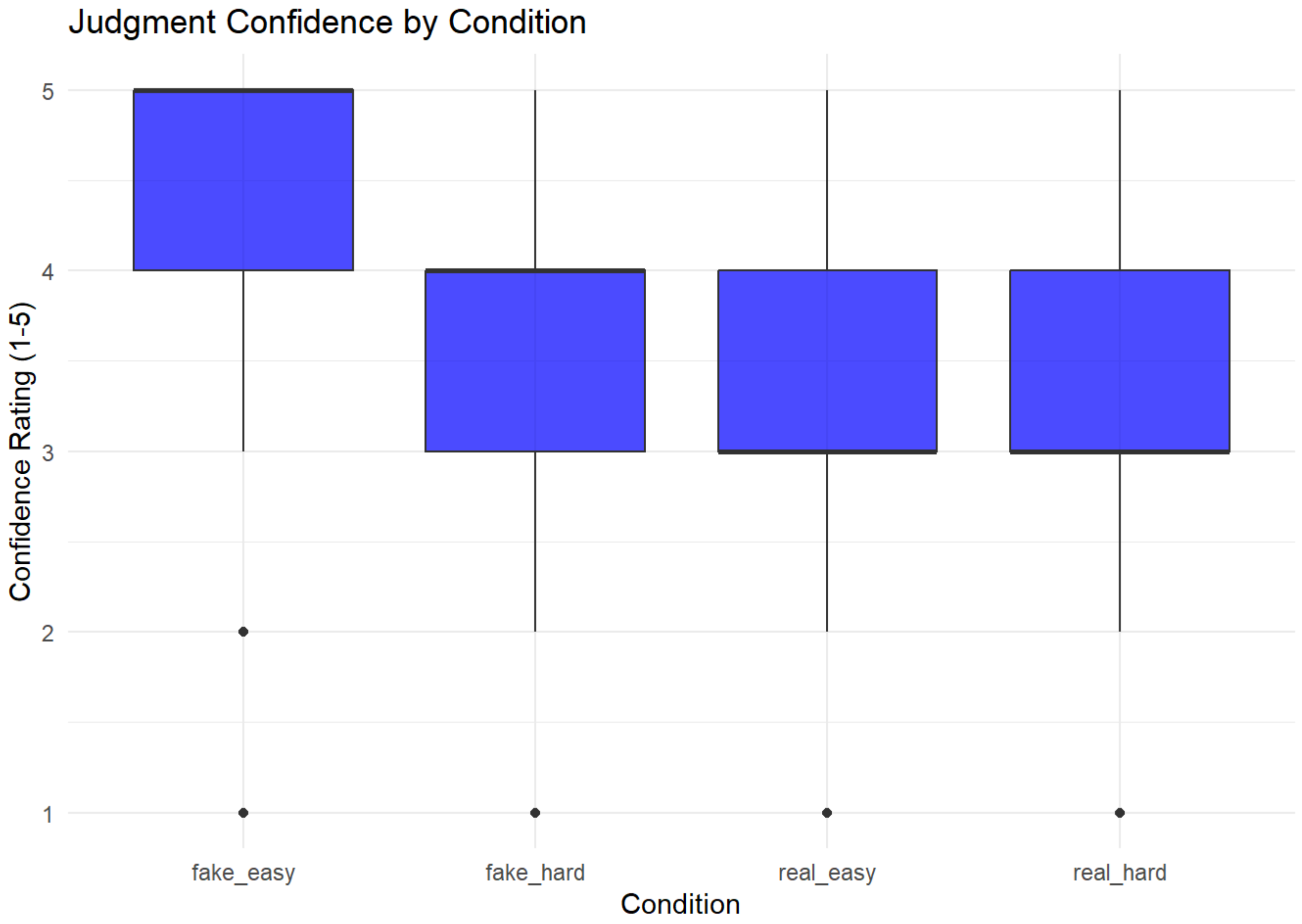
2.4. Confidence by Condition

Plotting

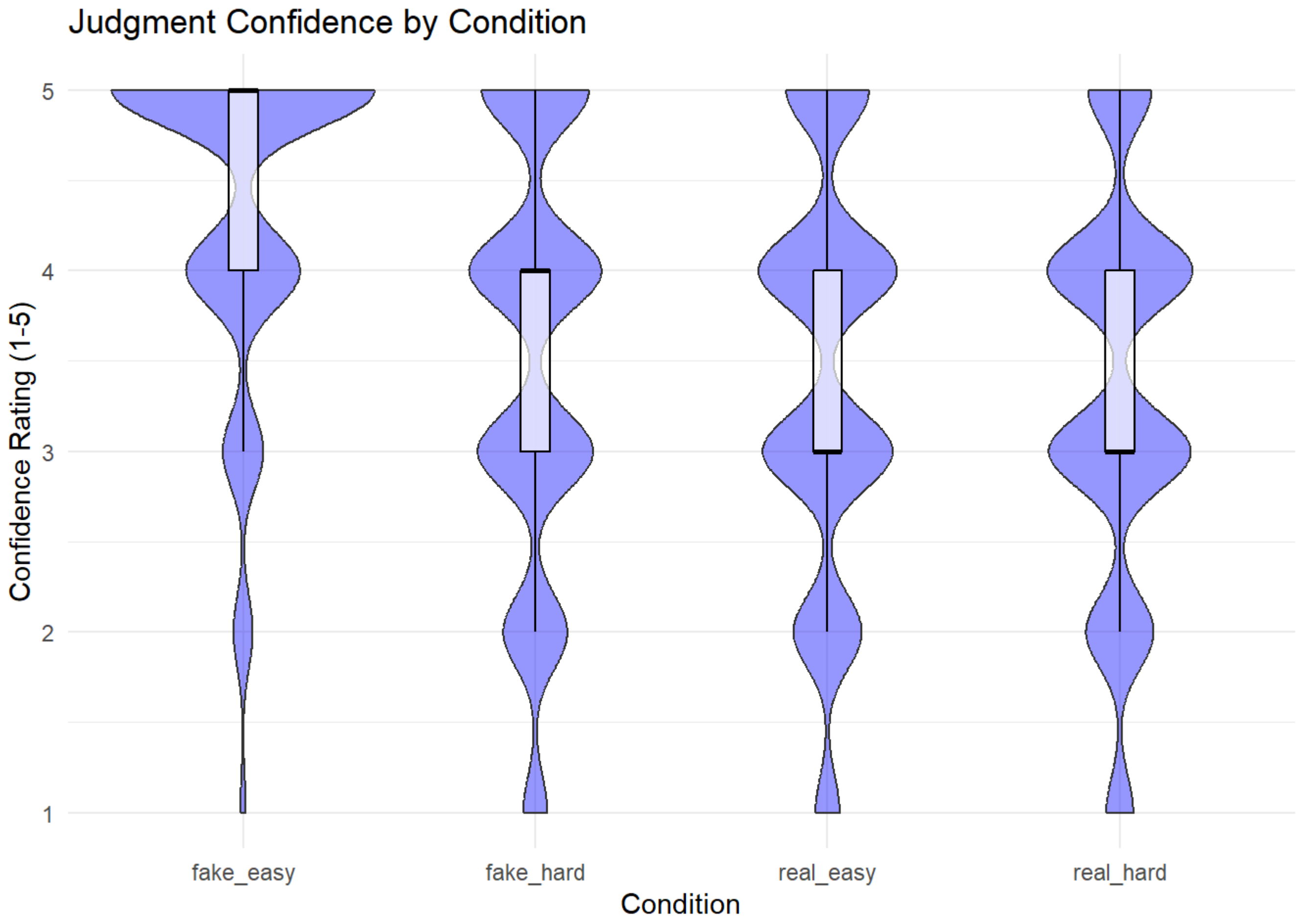
```
mean(data$Confidence)
```

```
## [1] 3.663114
```

```
data %>%
  ggplot(aes(x = Condition, y = Confidence)) +
  geom_boxplot(alpha = 0.7, fill = "blue") +
  coord_cartesian(ylim = c(1, 5)) +
  labs(x = "Condition", y = "Confidence Rating (1-5)",
       title = "Judgment Confidence by Condition") +
  theme_minimal() +
  theme(legend.position = "none")
```



```
data %>%
  ggplot(aes(x = Condition, y = Confidence)) +
  geom_violin(trim = TRUE, alpha = 0.4, fill = "blue") +
  geom_boxplot(width = 0.1, color = "black", outlier.shape = NA, alpha = 0.7) +
    labs(x = "Condition", y = "Confidence Rating (1-5)",
       title = "Judgment Confidence by Condition") +
  theme_minimal() +
  theme(legend.position = "none")
```



Modelling

```
m_confidence_by_condition <- lmer(Confidence ~ Condition + (1 | ParticipantID) + (1 | Filenam
e), data = data)

summary(m_confidence_by_condition)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Confidence ~ Condition + (1 | ParticipantID) + (1 | Filename)
##    Data: data
##
## REML criterion at convergence: 4791
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.7060 -0.6356  0.1775  0.6106  2.7884
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Filename      (Intercept) 0.02225  0.1492
##  ParticipantID (Intercept) 0.20816  0.4562
##  Residual                  0.93872  0.9689
## Number of obs: 1689, groups:  Filename, 80; ParticipantID, 22
##
## Fixed effects:
##                    Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)         4.35979    0.11333 33.36367   38.47  < 2e-16 ***
## Conditionfake_hard -0.83759    0.08195 78.54099  -10.22 4.41e-16 ***
## Conditionreal_easy -0.94849    0.08193 78.40107  -11.58  < 2e-16 ***
## Conditionreal_hard -1.02838    0.08174 77.71182  -12.58  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Cndtnf_ Cndtnrl_s
## Cndtnfk_hrd -0.364
## Condtnrl_sy -0.364  0.503
## Cndtnrl_hrd -0.365  0.504   0.505
```

```
m0_confidence <- lmer(Confidence ~ (1|ParticipantID) + (1|Filename), data = data)

anova(m0_confidence, m_confidence_by_condition)
```

```
## refitting model(s) with ML (instead of REML)
```

	n…	AIC	BIC	logLik	-2*log(L)	Chisq	Df
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
m0_confidence	4	4889.164	4910.891	-2440.582	4881.164	NA	NA
m_confidence_by_condition	7	4791.933	4829.956	-2388.966	4777.933	103.2308	3

2 rows

```
conf_means <- emmeans(m_confidence_by_condition, "Condition")

pairs(conf_means, adjust = "bonferroni")
```

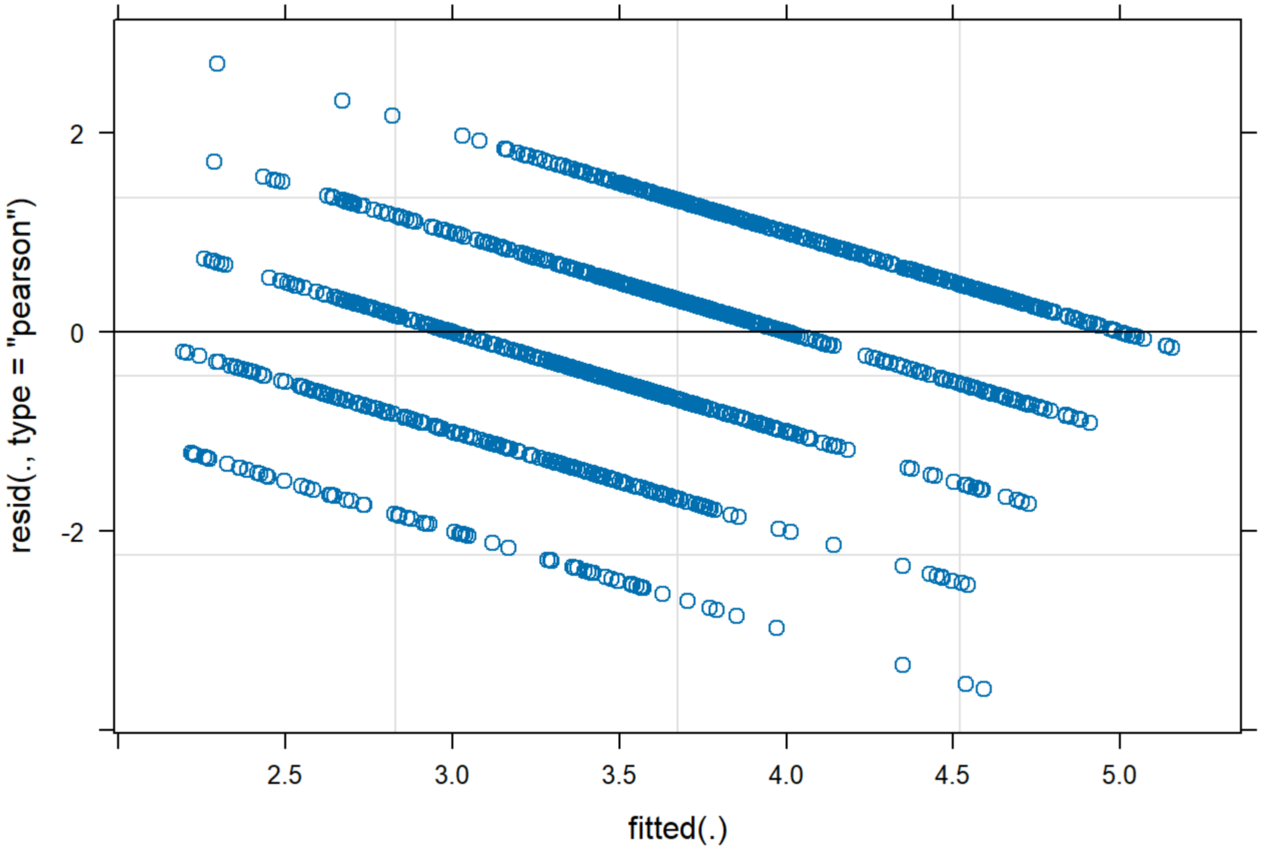
```
##  contrast              estimate     SE   df t.ratio p.value
##  fake_easy - fake_hard   0.8376 0.0820 76.9  10.220 <0.0001
##  fake_easy - real_easy   0.9485 0.0819 76.8  11.576 <0.0001
##  fake_easy - real_hard   1.0284 0.0817 76.1  12.581 <0.0001
##  fake_hard - real_easy   0.1109 0.0817 75.8   1.358  1.0000
##  fake_hard - real_hard   0.1908 0.0815 75.2   2.341  0.1312
##  real_easy - real_hard   0.0799 0.0815 75.0   0.981  1.0000
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: bonferroni method for 6 tests
```

Report: To analyse the effect of Condition on the subjective Confidence score, a linear mixed-effect model was fit, with Condition as fixed effect and random intercepts for participants and stimuli.The results of the likelihood-ratio test comparing the full model (random-effects and Condition) with a null model (random-effects only) showed that Condition significantly improved model fit ($\chi^2(3) = 103.23$, $p < .0001$). The post-hoc pairwise comparisons showed that the mean confidence score for “fake_easy” (*M* = 4.36) was significantly higher than all other conditions (all *p*s < .0001). However, other conditions did not significantly differ among each other in their confidence scores (all *p*s > .13), with “real_hard” having the lowest confidence score (*M* = 3.33) and the overall mean Confidence being 3.66. The results suggest that “fake_easy” was notably easier to classify compared to all other conditions, that sounded more realistic.

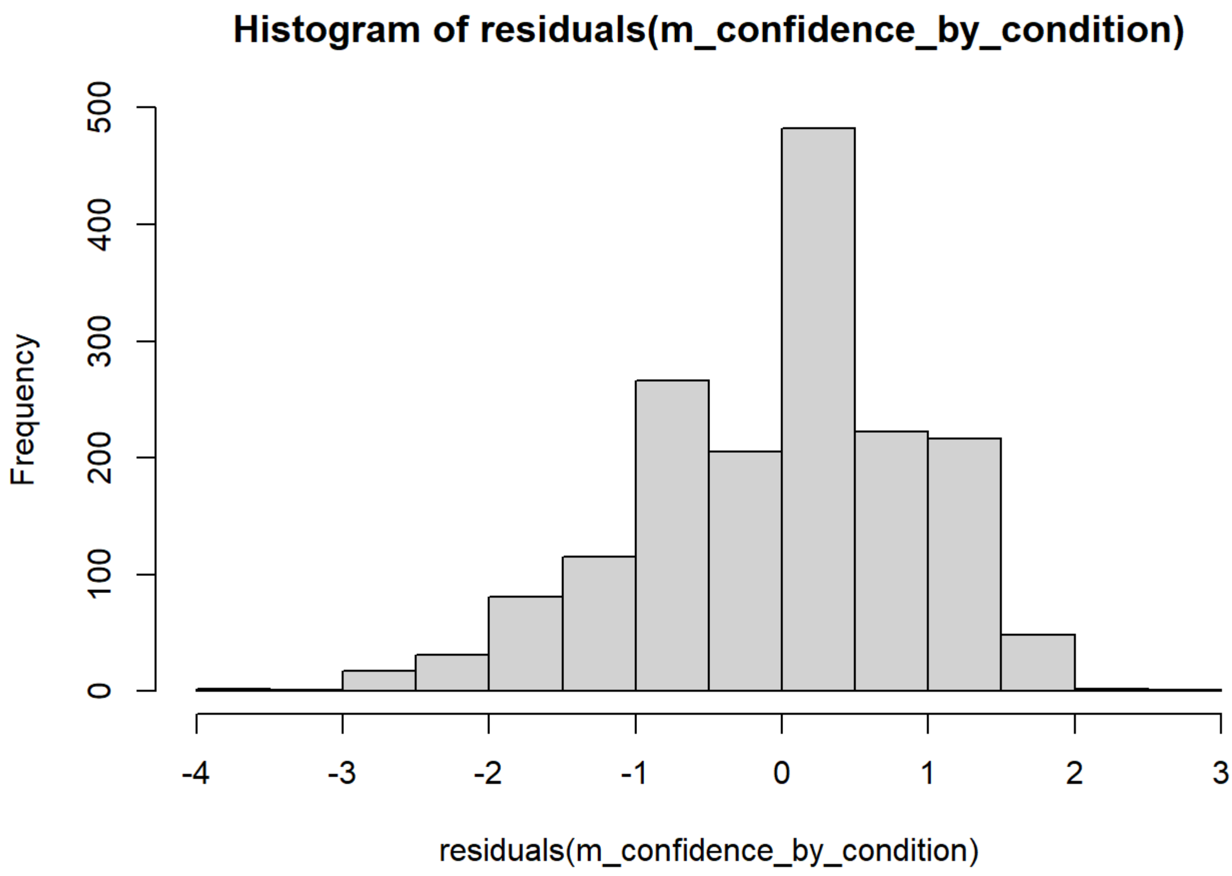
Furthermore, the mean confidence score of 3.66 suggests that participants were overall more confident than not in their answers. However, interestingly, there seems to be an interesting inverse relationship between perceived Naturalness and Confidence. Comparing with the analysis of the effect of Condition on Naturalness, “real” voices received higher Naturalness scores, but also lower Confidence scores. It suggests that participants were more Confident with unnatural voices than with natural ones. It might be the case that unnatural voices seem less likely to be human, while there is the possibility of an AI producing natural voices.

Checking assumptions

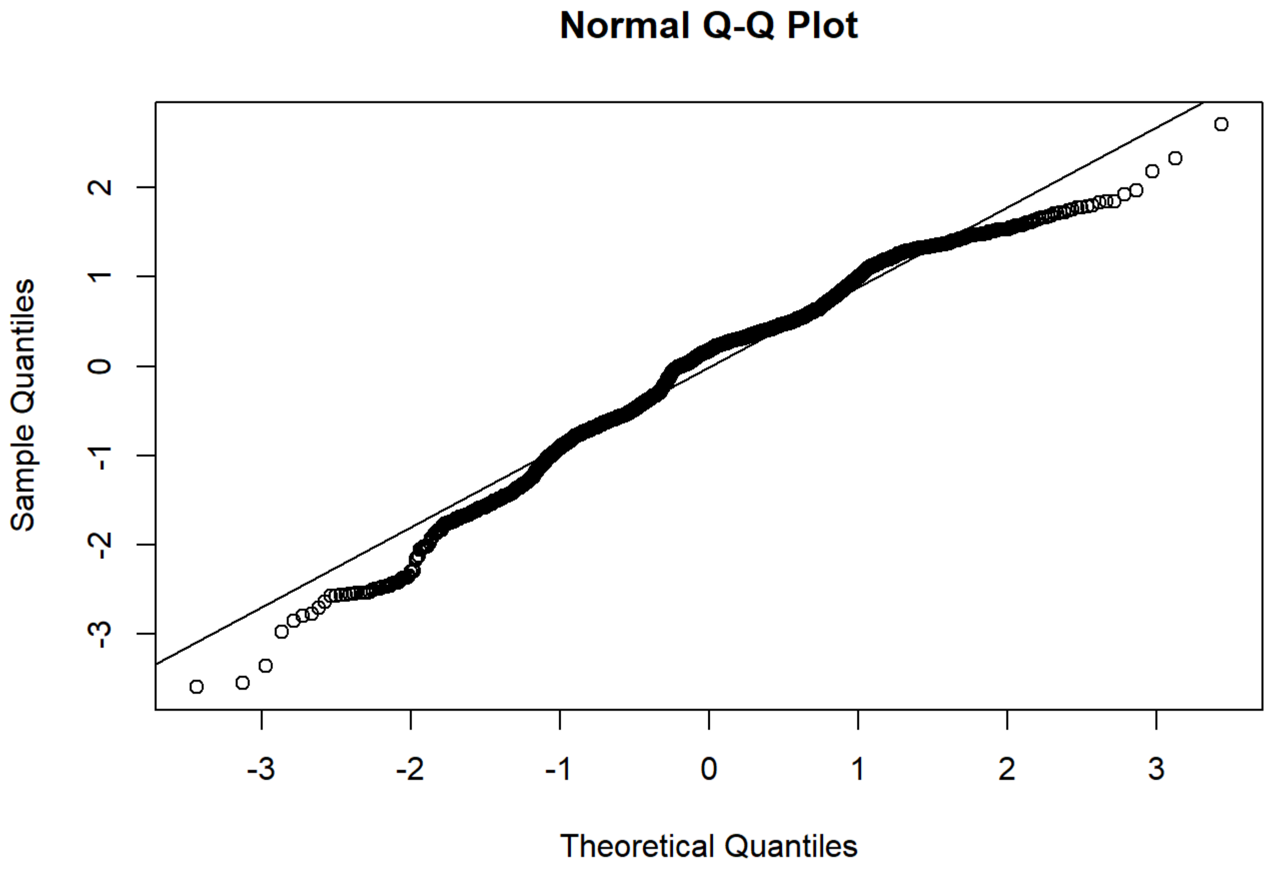
```
plot(m_confidence_by_condition)
```



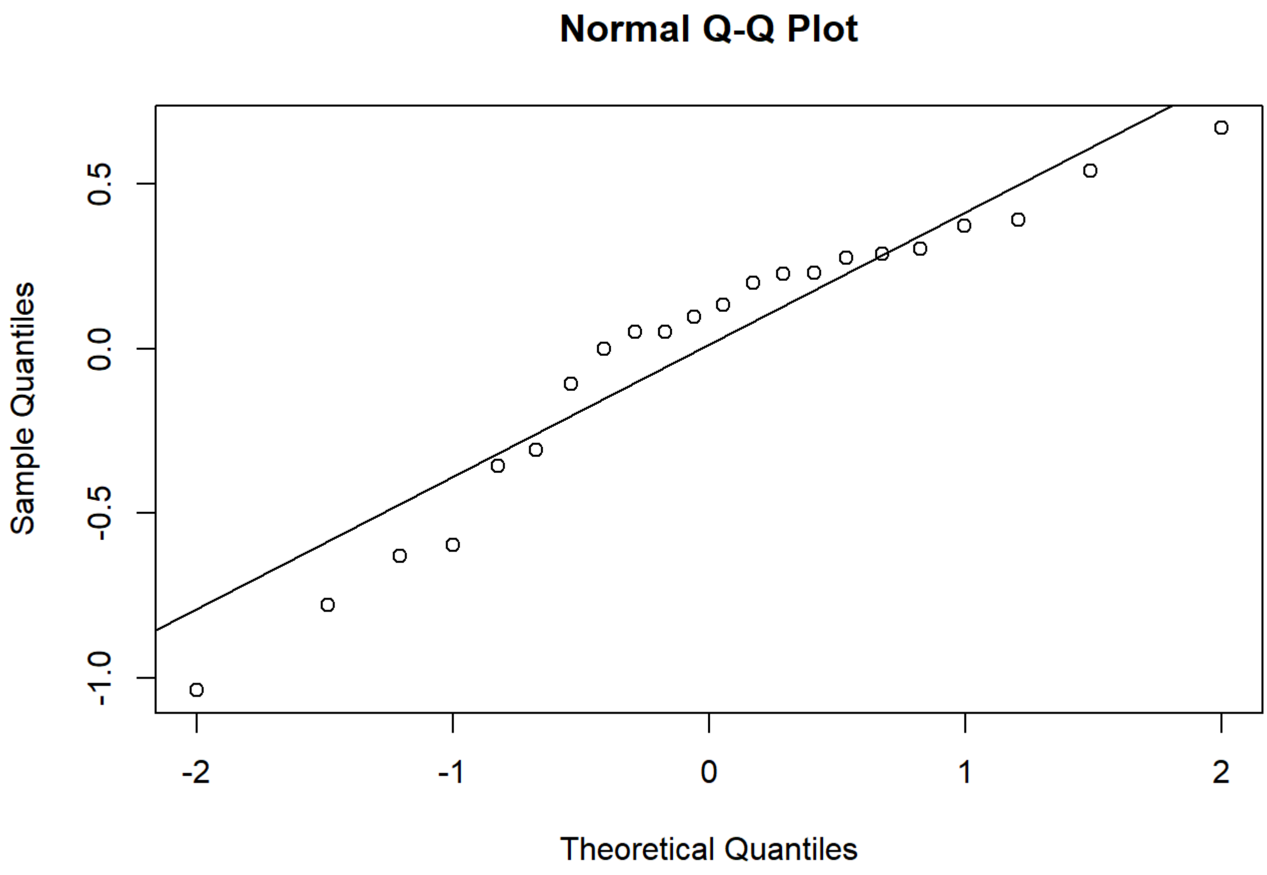
```
    ## Residuals are homogenic and the relationship is linear (for ordinal data)

hist(residuals(m_confidence_by_condition))
```



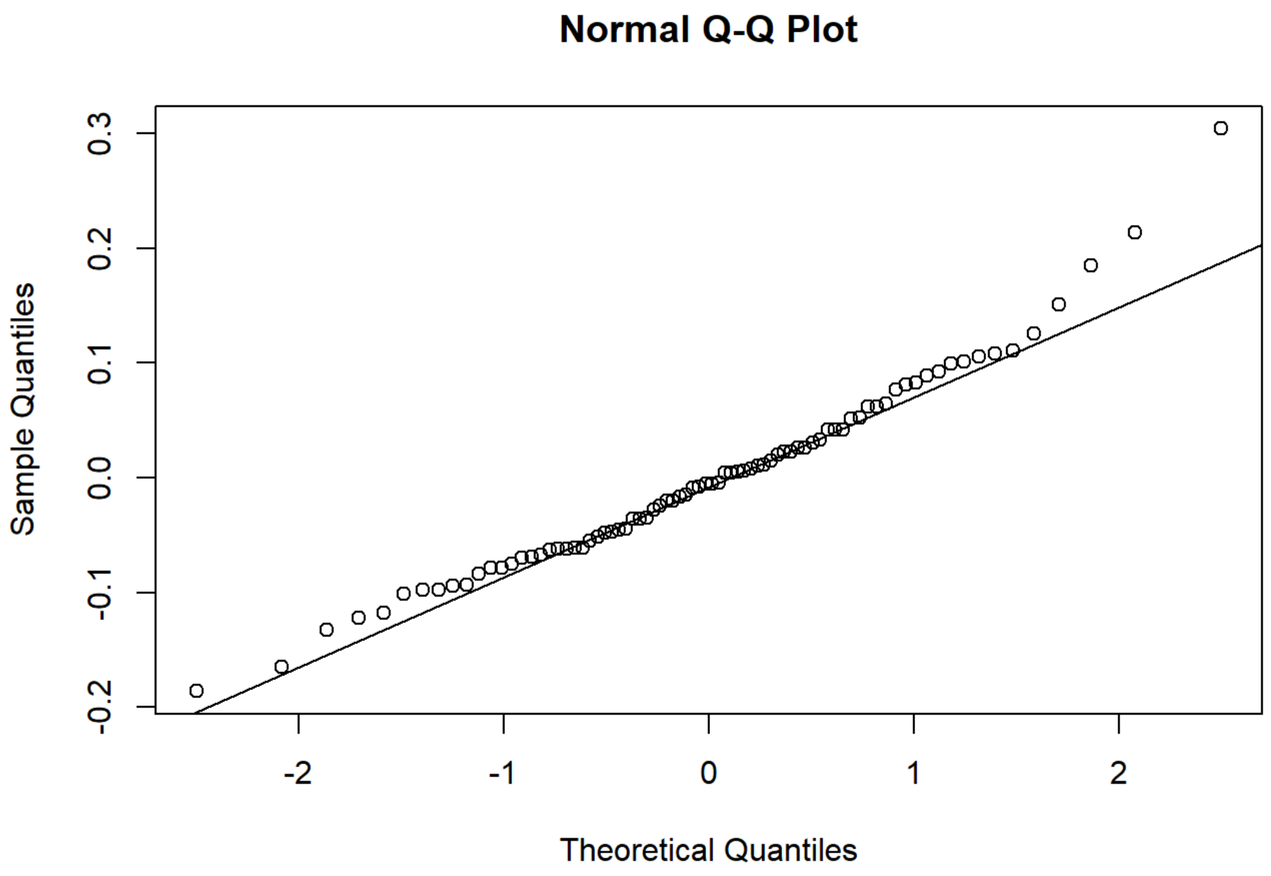
```
qqnorm(residuals(m_confidence_by_condition))
qqline(residuals(m_confidence_by_condition))
```



```
    ## Residuals are approximately normal with symmetric deviations at the tails

qqnorm(ranef(m_confidence_by_condition)$ParticipantID[[1]])
qqline(ranef(m_confidence_by_condition)$ParticipantID[[1]])
```



```
    ## Random effect per participant somewhat deviates from normality, but given the robustness of the model, it should still be acceptable

qqnorm(ranef(m_confidence_by_condition)$Filename[[1]])
qqline(ranef(m_confidence_by_condition)$Filename[[1]])
```



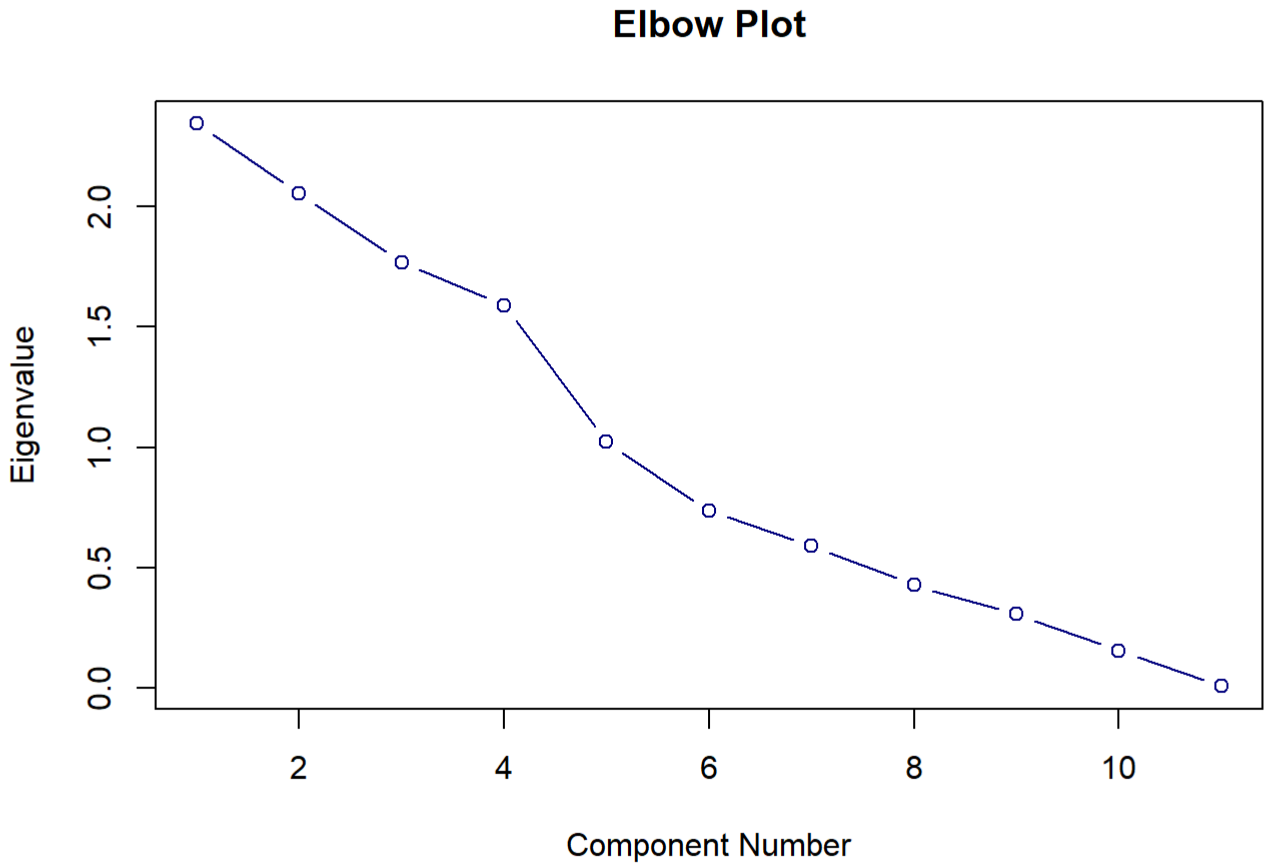
```
    ## Random effect per filename slightly deviates from normality, but given the robustness
of the model, it should still be acceptable
```

Report: The visual inspection of QQ-plots for residuals and random effects, showed some deviations from normality, which were nevertheless deemed acceptable, given the robustness of the model.

3. Analysis of Acoustic Features

3.1. Principal Component Analysis (PCA)

```
eigvals <- eigen(cor(acoustics_scaled))

eigenvalues <- eigvals$values

plot(eigenvalues, type="b", main="Elbow Plot",
     xlab="Component Number", ylab="Eigenvalue", col="navy")
```



```
## The elbow plot shows that the number of components that explain most of the meaningful variance is 4

pca <- principal(acoustics_scaled, nfactors = 4, rotate = "varimax")
pca$loadings
```

```
##
## Loadings:
##                   RC1    RC4    RC2    RC3
## F0_mean            0.135  0.892              
## F0_std                          -0.246 -0.202
## Intensity_mean                  -0.104  0.829
## Intensity_std             0.252        -0.830
## F1                               0.679 -0.185
## F2                               0.824       
## F3                -0.160         0.846  0.132
## HNR                       0.885              
## SpectralCentroid   0.992                     
## SpectralBandwidth  0.987                     
## ZeroCrossingRate  -0.152  0.538         0.562
##
##                  RC1   RC4   RC2   RC3
## SS loadings    2.044 1.960 1.946 1.803
## Proportion Var 0.186 0.178 0.177 0.164
## Cumulative Var 0.186 0.364 0.541 0.705
```

```
audio_data$PC1 <- pca$scores[,1]
audio_data$PC2 <- pca$scores[,2]
audio_data$PC3 <- pca$scores[,3]
audio_data$PC4 <- pca$scores[,4]



data <- data %>%
  left_join(audio_data %>% select(Filename, PC1, PC2, PC3, PC4), by = "Filename")
```

Report: To reduce the number of acoustic features to the most relevant and to reduce multicollinearity of these features, a Principal Component Analysis (PCA) with Varimax rotation for interpretation was performed on 11 extracted acoustic properties of the 80 audio files. After visually inspecting a scree plot, a number of 4 principal components was chosen, capturing 70.5% of the variance.

Based on the features that had the highest correlation with the components, they were interpreted as:

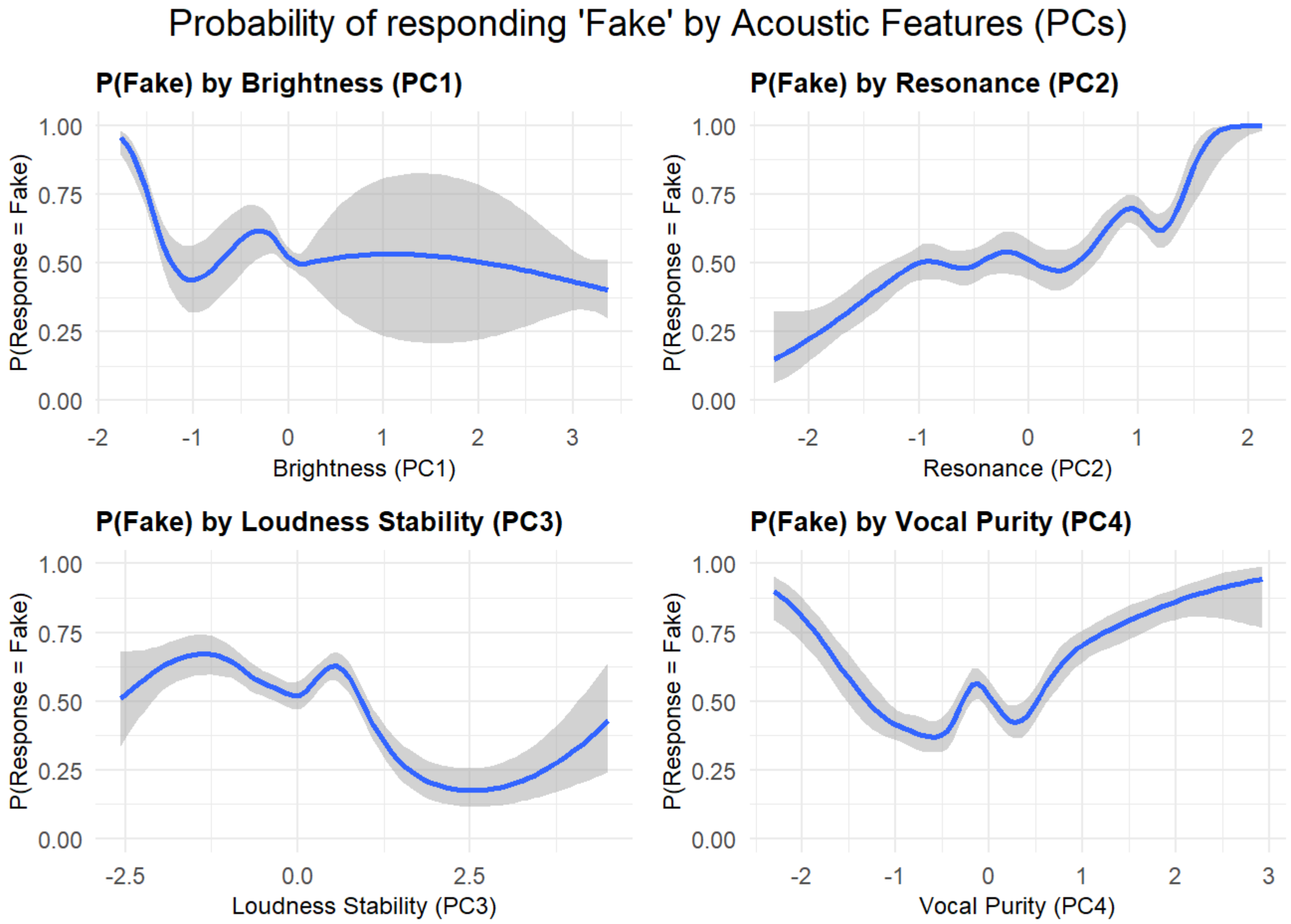
1. Brightness (sharpness) (RC1 (18.6% of variance): main corresponding features are SpectralCentroid (cor = .992) and SpectralBandwidth (cor = .987)),
2. Vocal purity (cleanness) (RC4 (17.8% of variance): F0_mean (cor = .892), HNR (cor = .885)),
3. Resonance (RC2 (17.7% of variance): F3 (cor = .846), F2 (cor = .824), F1 (cor = .679)),
4. Loudness stability (RC3 (16.4% of variance): Intensity_mean (cor = .829), Intensity_std (cor = -.830)).

All of the components explain roughly the same amount of variance, with brightness explaining slightly more than others and loudness explaining the least of it.

3.2 Response by PCs

Plotting Response by Acoustic features

```
plot_pc1 <- ggplot(data, aes(x = PC1, y = as.numeric(Response == "fake"))) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), method.args = list(family = binomial)) +
  labs(x = "Brightness (PC1)", y = "P(Response = Fake)", title = "P(Fake) by Brightness (PC1)") +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 9))

plot_pc2 <- ggplot(data, aes(x = PC2, y = as.numeric(Response == "fake"))) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), method.args = list(family = binomial)) +
  labs(x = "Resonance (PC2)", y = "P(Response = Fake)", title = "P(Fake) by Resonance (PC2)") +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 9))

plot_pc3 <- ggplot(data, aes(x = PC3, y = as.numeric(Response == "fake"))) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), method.args = list(family = binomial)) +
  labs(x = "Loudness Stability (PC3)", y = "P(Response = Fake)", title = "P(Fake) by Loudness Stability (PC3)") +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 9))

plot_pc4 <- ggplot(data, aes(x = PC4, y = as.numeric(Response == "fake"))) +
  geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"), method.args = list(family = binomial)) +
  labs(x = "Vocal Purity (PC4)", y = "P(Response = Fake)", title = "P(Fake) by Vocal Purity (PC4)") +
  coord_cartesian(ylim = c(0, 1)) +
  theme_minimal() +
  theme(plot.title = element_text(size = 10, face = "bold"),
        axis.title = element_text(size = 9))

(plot_pc1 + plot_pc2) / (plot_pc3 + plot_pc4) +
  plot_annotation(title = "Probability of responding 'Fake' by Acoustic Features (PCs)",
                  theme = theme(plot.title = element_text(size = 14, hjust = 0.5)))
```



Modelling Response by PCs, using generalized additive model (GAM)

```
data$Response <- relevel(data$Response, ref = "real")

m_response_by_pcs <- gam(Response ~ s(PC1) + s(PC2) + s(PC3) + s(PC4) +s(ParticipantID, bs =
"re")  + s(Filename, bs = "re"), family = binomial(link = "logit"), data = data, method = "RE
ML")

summary(m_response_by_pcs)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## Response ~ s(PC1) + s(PC2) + s(PC3) + s(PC4) + s(ParticipantID,
##     bs = "re") + s(Filename, bs = "re")
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4891     0.2613   1.872   0.0612 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df  Chi.sq p-value
## s(PC1)            1.000  1.000   2.500 0.11386
## s(PC2)            1.000  1.001  10.223 0.00139 **
## s(PC3)            1.000  1.000   1.082 0.29828
## s(PC4)            2.544  2.653   8.641 0.02633 *
## s(ParticipantID) 16.463 21.000  91.352 < 2e-16 ***
## s(Filename)      64.397 75.000 468.717 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.47   Deviance explained = 43.1%
## -REML = 809.74  Scale est. = 1         n = 1689
```

```
print(paste("Baseline probability of responding 'fake' (all PCs have average values): ", round(plogis(0.4891) * 100, 1),"%"))
```

```
## [1] "Baseline probability of responding 'fake' (all PCs have average values):  62 %"
```

Report: To analyze how acoustic features predict the Response of participants, a possibility of non-linear relationship was allowed by fitting a Generalized Additive Model with a logit link function, with 4 principal acoustic components as fixed effects and random intercepts for participants and stimuli. The model explained 43.1% of deviance.

The results showed that baseline probability of responding “fake” was 62%, but it did not significantly differ from chance level (*b* = 0.489, *SE* = 0.261, *z* = 1.87, *p* = 0.0612). 2 out of 4 principal components had a significant effect on Response. Resonance had a positive linear relationship with probability of responding “fake” (PC2; *edf* = 1.000, χ^2 = 10.223, *p* = .0014), while Vocal Purity had a non-linear U-shaped relationship with low and high values resulting in high probability and middle values in a lower probability of responding “fake”. (PC4; *edf* = 2.544, χ^2 = 8.641, *p* = .0263). Brightness (PC1) and Loudness Stability (PC3) had a non-significant linear effects on response (both *edf*s = 1, both *p*s > .11). Moreover, the results show significant variation across participants and stimuli (both *p*s < .0001).