Total motility

Marie Bisconti, Philippe Grosjean & Elise Hennebert

Introduction

The effect of DMSO and ethanol is evaluated at concentrations from 0 up to 2% on spermatozoa total motility.

DMSO

```
mt_dmso <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "total motility_DMSO")
mt_dmso$donor <- as.factor(mt_dmso$donor)
names(mt_dmso) <- c("donor", "conc", "motile", "total")
mt_dmso$motile_frac <- mt_dmso$motile / mt_dmso$total
skimr::skim(mt_dmso)
```

Table 1: Data summary

Name	mt_dmso
Number of rows	40
Number of columns	5
_____	
Column type frequency:	
factor	1
numeric	4
_____	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
donor	0	1	FALSE	8	1: 5, 2: 5, 3: 5, 4: 5

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
conc	0	1	0.72	0.74	0.00	0.10	0.5	1.00	2.00	
motile	0	1	100.65	44.93	41.00	72.00	91.0	119.00	260.00	
total	0	1	131.20	56.45	54.00	92.00	124.5	150.50	301.00	
motile_frac	0	1	0.77	0.09	0.53	0.73	0.8	0.83	0.89	

There are eight donors, no missing data.

```
table(mt_dmso$donor, as.factor(mt_dmso$conc))
```

```
  0 0.1 0.5 1 2
1 1   1   1 1 1
2 1   1   1 1 1
3 1   1   1 1 1
4 1   1   1 1 1
5 1   1   1 1 1
6 1   1   1 1 1
7 1   1   1 1 1
8 1   1   1 1 1
```

The data are balanced with one observation for each concentration and each donor and no missing data.

```
mt_dmso_m1 <- glmer(cbind(motile, total - motile) ~ conc + (conc | donor),
  data = mt_dmso, family = binomial(link = "logit"))
summary(mt_dmso_m1)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(motile, total - motile) ~ conc + (conc | donor)
```

```
   Data: mt_dmso

     AIC      BIC   logLik deviance df.resid
   314.1    322.6   -152.1    304.1       35

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.8305 -0.8161  0.0966  0.7369  5.0822

Random effects:
 Groups Name        Variance Std.Dev. Corr
 donor  (Intercept) 0.111404 0.3338
        conc        0.007157 0.0846   0.02
Number of obs: 40, groups:  donor, 8

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.39733    0.12833   10.89  < 2e-16 ***
conc        -0.18586    0.05615   -3.31 0.000933 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.220
```

The model is:

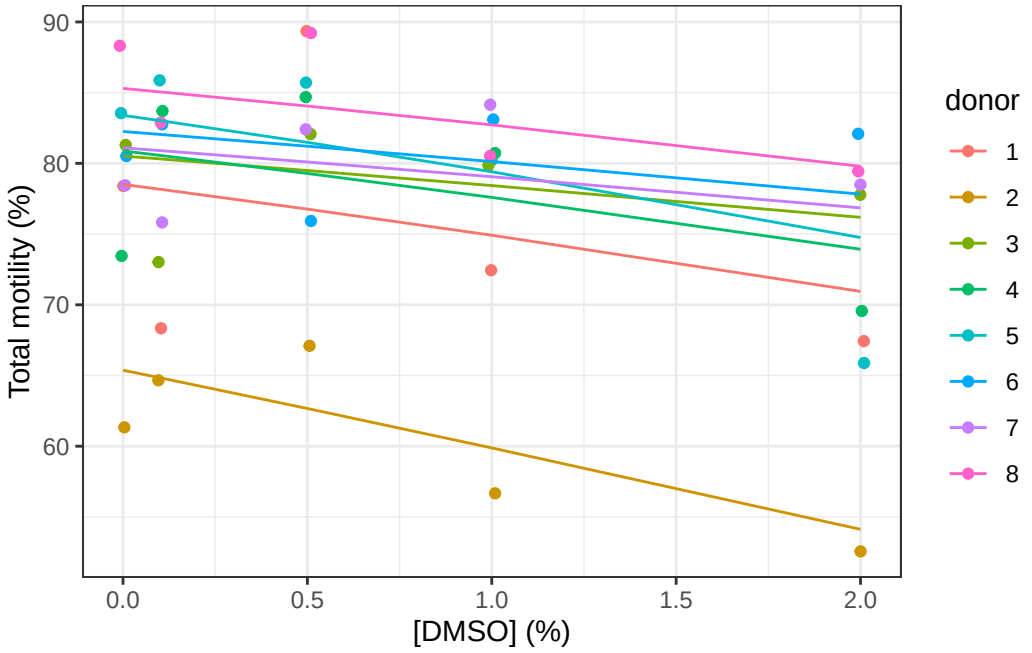
,

$$
\begin{aligned}
&\text{motile}_i \sim \text{Binomial}(n = 1, \text{prob}_{\text{motile}=1} = \widehat{P}) \\
&, \log\left[\frac{\hat{P}}{1-\hat{P}}\right] = \alpha_{j[i]} + \beta_{1j[i]}(\text{conc}) \\
&\begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}} \end{pmatrix}\right), \text{for donor j} = 1, \ldots, \text{J}
\end{aligned}
\quad , \qquad (1)
$$

Here is a plot of this model:

```
ggplot(data = mt_dmso) +
  geom_jitter(aes(x = conc, y = motile_frac * 100, col = donor),
    width = 0.01) +
```

```
  geom_line(aes(x = conc, y = fitted(mt_dmso_m1) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Total motility (%)")
```



Generally, slopes are all negative, suggesting a negative concentration effect. Data are rather widespread. There are definitely shifts in the intercept per donor (different motility at [DMSO] = 0). Slopes seems not too different between donors (can the model be simplified?). Let's check it with a likelihood ratio test:

```
mt_dmso_m2 <- glmer(cbind(motile, total - motile) ~ conc  + (1 | donor),
  data = mt_dmso, family = binomial(link = "logit"))
anova(mt_dmso_m1, mt_dmso_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

```
Data: mt_dmso
Models:
mt_dmso_m2: cbind(motile, total - motile) ~ conc + (1 | donor)
mt_dmso_m1: cbind(motile, total - motile) ~ conc + (conc | donor)
           npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
mt_dmso_m2    3 310.59 315.66 -152.30   304.59
mt_dmso_m1    5 314.11 322.56 -152.06   304.11  0.48  2     0.7866
```

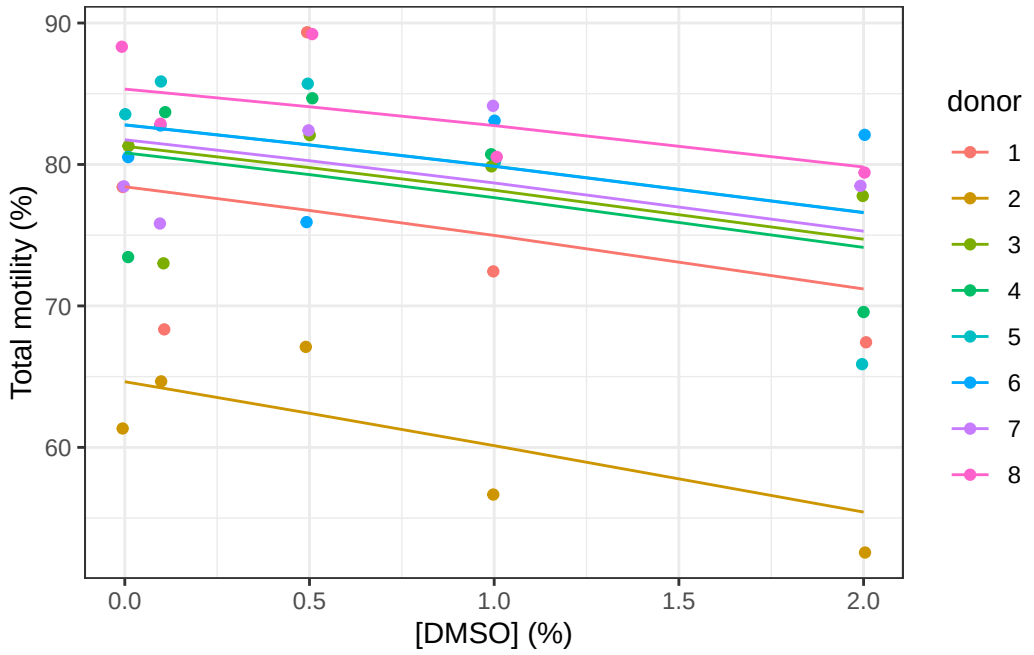
The likelihood ratio test does not detects significant differences between the full and simplified models at $\alpha = 5\%$. We could thus use the simplest `mt_dmso_m2` model with only a shift in the slope per donor. This model is:

,

$$\begin{aligned} \text{motile}_i &\sim \text{Binomial}(n = 1, \text{prob}_{\text{motile}=1} = \widehat{P}) \\ \text{, log}\left[\frac{\hat{P}}{1-\hat{P}}\right] &= \alpha_{j[i]} + \beta_1(\text{conc}) \qquad \text{,} \\ \alpha_j &\sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{ for donor j} = 1, \ldots, \text{J} \end{aligned} \tag{2}$$

Here is a plot of this model:

```
ggplot(data = mt_dmso) +
  geom_jitter(aes(x = conc, y = motile_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mt_dmso_m2) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Total motility (%)")
```




```
summary(mt_dmso_m2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(motile, total - motile) ~ conc + (1 | donor)
   Data: mt_dmso

     AIC      BIC   logLik deviance df.resid
   310.6    315.7   -152.3    304.6       37

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.7957 -0.8682  0.0641  0.8505  5.0878

Random effects:
 Groups Name        Variance Std.Dev.
 donor  (Intercept) 0.1166   0.3415
Number of obs: 40, groups:  donor, 8

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.40292    0.13047  10.753   <2e-16 ***
conc        -0.19271    0.04519  -4.265    2e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.256
```

The Z test indicates that `conc` is significantly different from zero at $\alpha = 5\%$. However, it is not the best test in the case of a mixed model like here. We prefer to rely on the 95% confidence interval calculated either on the profile, or via parametric bootstrap (and especially the later one):

```
confint(mt_dmso_m2, level = 0.95) # 95% CI based on profile
```

```
Computing profile confidence intervals ...

                2.5 %     97.5 %
.sig01       0.2143841  0.6264441
(Intercept)  1.1208466  1.6907146
conc        -0.2810892 -0.1038180
```

```r
set.seed(8434)
# 1000x parameter bootstrap
(mt_dmso_m2_conf <- confint(mt_dmso_m2, level = 0.95,
  method = "boot", nsim  = 1000L))
```

```
Computing bootstrap confidence intervals ...
```

```
2 warning(s): Model failed to converge: degenerate  Hessian with 1 negative eigenvalues (and
```

```
                 2.5 %      97.5 %
.sig01       0.1220672  0.49749800
(Intercept)  1.1470026  1.66783797
conc        -0.2800799 -0.09954095
```

Slope for `conc` is significantly different from zero at $\alpha = 5\%$ because the 95% CI does not contain zero.

Additional verifications

Check if there is not a overdispersion (in this case, a binomial generalized model would not be adequate), $Var(Y) = \varphi Np(1-p)$ with φ, the overdispersion coefficient that has to be close to zero. However, "overdispersion is not estimable (and hence practically irrelevant) for Bernoulli models (= binary data = binomial with N=1).", see glmmFAQ. Thus, it cannot be estimated here.

We could double-check the significance of the slope `conc` by looking at a likelihood ratio test when dropping `conc` from the model:

```r
#drop1(mt_dmso_m2, scope = "conc")
mt_dmso_m3 <- glmer(cbind(motile, total - motile) ~ 1 + (1 | donor),
  data = mt_dmso, family = binomial(link = "logit"))
anova(mt_dmso_m2, mt_dmso_m3, refit = TRUE)
```

```
Data: mt_dmso
Models:
mt_dmso_m3: cbind(motile, total - motile) ~ 1 + (1 | donor)
mt_dmso_m2: cbind(motile, total - motile) ~ conc + (1 | donor)
           npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
```

```
mt_dmso_m3    2 326.50 329.88 -161.25   322.50
mt_dmso_m2    3 310.59 315.66 -152.30   304.59 17.906  1   2.32e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with `conc` is significantly different at α level 5% from a reference model that sets the slope `conc` $= 0$. There is thus a significant effect of DMSO concentration (confirmation of results obtained from 95% CI).

We also double-check convergence of the model by trying different optimisation engines (just to make sure). First, is there a singularity in the model?

```
isSingular(mt_dmso_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```
mt_dmso_m2_all <- allFit(mt_dmso_m2)
```

```
Loading required namespace: dfoptim
```

```
Loading required namespace: optimx
```

```
bobyqa : [OK]
Nelder_Mead : [OK]
nlminbwrap : [OK]
nmkbw : [OK]
optimx.L-BFGS-B : [OK]
nloptwrap.NLOPT_LN_NELDERMEAD : [OK]
nloptwrap.NLOPT_LN_BOBYQA : [OK]
```

```
summary(mt_dmso_m2_all)
```

```
$which.OK
                        bobyqa                     Nelder_Mead
                          TRUE                            TRUE
                     nlminbwrap                           nmkbw
                          TRUE                            TRUE
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                          TRUE                            TRUE
    nloptwrap.NLOPT_LN_BOBYQA
                          TRUE

$msgs
$msgs$bobyqa
NULL

$msgs$Nelder_Mead
NULL

$msgs$nlminbwrap
NULL

$msgs$nmkbw
NULL

$msgs$`optimx.L-BFGS-B`
NULL

$msgs$nloptwrap.NLOPT_LN_NELDERMEAD
NULL

$msgs$nloptwrap.NLOPT_LN_BOBYQA
NULL


$fixef
                               (Intercept)       conc
bobyqa                           1.402918 -0.1927083
Nelder_Mead                      1.402924 -0.1927091
nlminbwrap                       1.402918 -0.1927084
nmkbw                            1.402896 -0.1927420
optimx.L-BFGS-B                  1.402907 -0.1927069
nloptwrap.NLOPT_LN_NELDERMEAD    1.402879 -0.1927109
nloptwrap.NLOPT_LN_BOBYQA        1.402918 -0.1927070
```

```
$llik
                      bobyqa                   Nelder_Mead
                    -152.2962                     -152.2962
                   nlminbwrap                         nmkbw
                    -152.2962                     -152.2962
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                    -152.2962                     -152.2962
    nloptwrap.NLOPT_LN_BOBYQA
                    -152.2962

$sdcor
                              donor.(Intercept)
bobyqa                                0.3415152
Nelder_Mead                           0.3415163
nlminbwrap                            0.3415153
nmkbw                                 0.3415363
optimx.L-BFGS-B                       0.3415439
nloptwrap.NLOPT_LN_NELDERMEAD         0.3415166
nloptwrap.NLOPT_LN_BOBYQA             0.3415171

$theta
                              donor.(Intercept)
bobyqa                                0.3415152
Nelder_Mead                           0.3415163
nlminbwrap                            0.3415153
nmkbw                                 0.3415363
optimx.L-BFGS-B                       0.3415439
nloptwrap.NLOPT_LN_NELDERMEAD         0.3415166
nloptwrap.NLOPT_LN_BOBYQA             0.3415171

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.049    0.000   0.049          0         0
Nelder_Mead                       0.058    0.000   0.058          0         0
nlminbwrap                        0.046    0.000   0.046          0         0
nmkbw                             0.062    0.001   0.063          0         0
optimx.L-BFGS-B                   0.345    0.002   0.346          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.057    0.000   0.057          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.042    0.000   0.042          0         0

$feval
                      bobyqa                   Nelder_Mead
                          74                            95
```

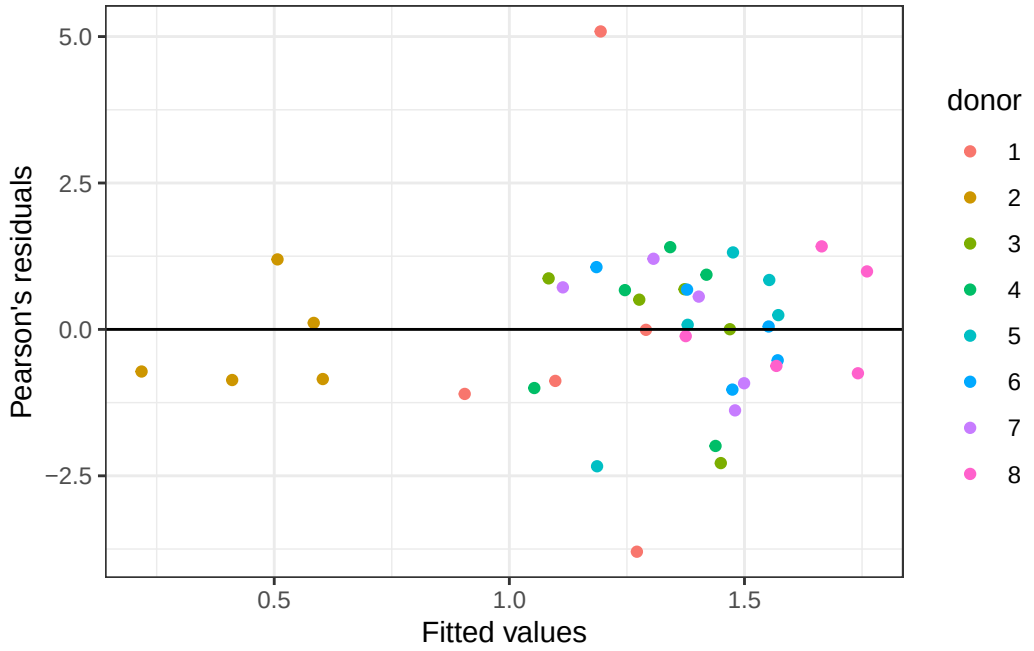
```
                    nlminbwrap                              nmkbw
                            NA                                104
                optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                            14                                88
    nloptwrap.NLOPT_LN_BOBYQA
                            36
```

```
attr(,"class")
[1] "summary.allFit"
```

Analysis of the residuals

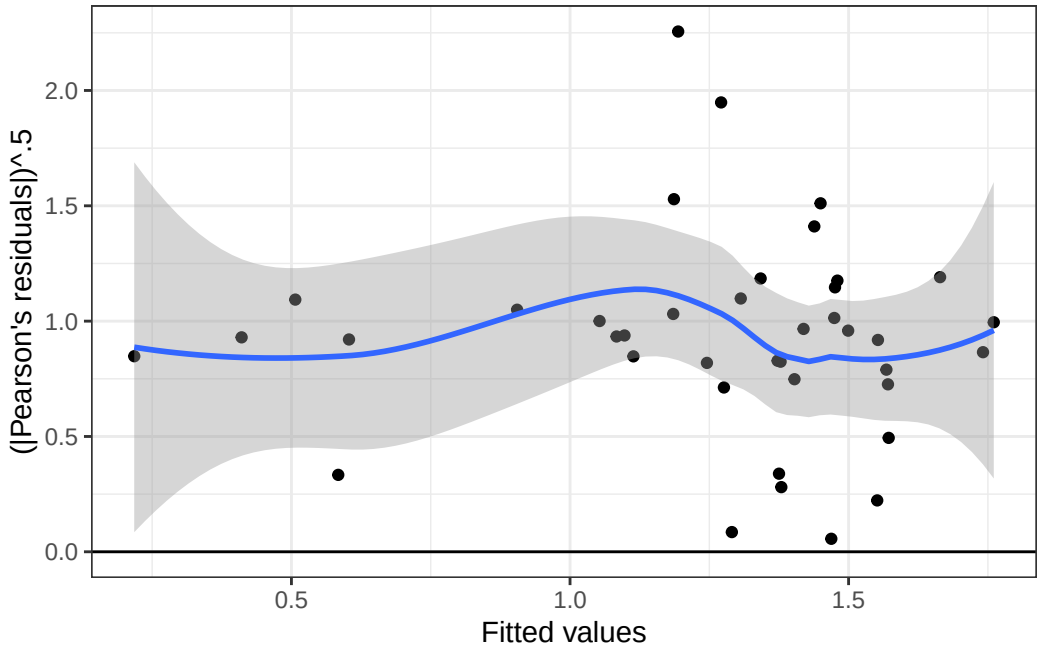
Let's check how the residuals distribute and if there is homoscedasticity.

```
mt_dmso <- fortify.merMod(mt_dmso_m2)
ggplot(data = mt_dmso, aes(x = .fitted, y = .scresid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")
```



There is one extreme value, but otherwise, residuals seem rather correctly distributed. Linearity is good here.

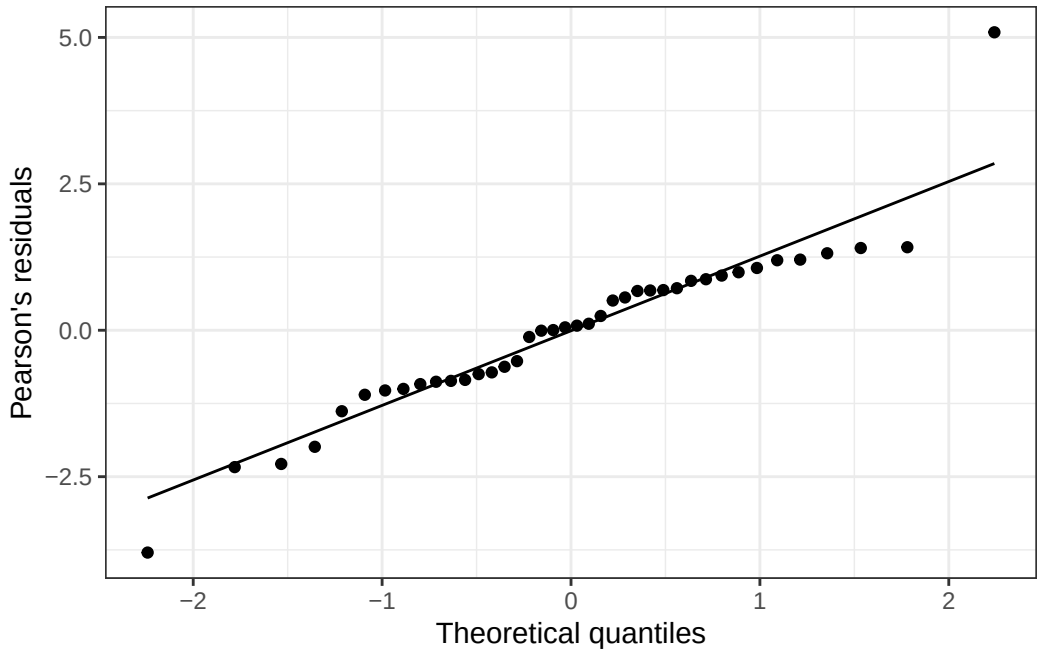
```
ggplot(data = mt_dmso, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals seems here acceptable (the blue curve that is a loess smoothing in the data is relatively horizontal).

Note: according to Gauss-Markov theorem that indicates that linearity, random sample, non-collinearity between predictors, non-correlation between predictors and error term and homoscedasticity are the only requirements for our GLMM, we do not have to check it. Also the model is robust to departure of Normality, and none of the tests we made depend on a Normal distribution of the residuals (we said we don't trust z/t tests and replace them by likelihood ratio tests and parameterized bootstrapped confidence intervals). However, for completeness, here is the quantile-quantile plot of the residuals:

```
ggplot(data = mt_dmso, aes(sample = .scresid)) +
  geom_qq() +
  geom_qq_line() +
  labs(x = "Theoretical quantiles", y = "Pearson's residuals")
```



It appears not too bad, except for our extreme value that is clearly visible here at the top. A Shapiro-Wilk test does not confirm Normality, but we are pretty sure it is caused by the extreme value:

```
shapiro.test(mt_dmso$.scresid)
```

```
    Shapiro-Wilk normality test

data:  mt_dmso$.scresid
W = 0.91868, p-value = 0.007
```

Predictions

The model allows to calculate the drop in total mobility according to DMSO concentration from 0 to 2%. Note that an inverse logit transformation is required. Here is an example:

```
mt_dmso_slope <- c(
  ci95_min   = min(mt_dmso_m2_conf["conc", ]),
  estimate = fixef(mt_dmso_m2)[["conc"]],
  ci95_max  = max(mt_dmso_m2_conf["conc", ]))
```

```
mt_dmso_slope
```

```
   ci95_min    estimate    ci95_max
-0.28007992 -0.19270912 -0.09954095
```

```
#saveRDS(mt_dmso_slope, "../data/motility_total_DMSO_slope.rds")
```

Let's say we want to calculate the drop in total mobility for various DMSO concentrations between 0 and 2% if the total mobility of a sample without DMSO is 80%. The calculation is:

```
predict_logit <- function(conc, intercept = 1, slopes) {
  slopes_mat <- matrix(slopes, nrow = 1,
    dimnames = list(NULL, names(slopes)))
  data.frame(conc = conc, -intercept +
      boot::inv.logit(boot::logit(intercept) +
      conc %*% slopes_mat))
}
dmso_conc <- (0:20) / 10
mt_dmso_lost <- predict_logit(dmso_conc, 0.8, mt_dmso_slope)
mt_dmso_lost
```

```
   conc     ci95_min     estimate     ci95_max
1   0.0  0.000000000  0.000000000  0.000000000
2   0.1 -0.004518953 -0.003101179 -0.001597412
3   0.2 -0.009113322 -0.006238047 -0.003204342
4   0.3 -0.013783163 -0.009410633 -0.004820795
5   0.4 -0.018528474 -0.012618950 -0.006446775
6   0.5 -0.023349194 -0.015863000 -0.008082285
7   0.6 -0.028245198 -0.019142771 -0.009727328
8   0.7 -0.033216298 -0.022458236 -0.011381905
9   0.8 -0.038262242 -0.025809357 -0.013046015
10  0.9 -0.043382709 -0.029196078 -0.014719660
11  1.0 -0.048577308 -0.032618332 -0.016402837
12  1.1 -0.053845580 -0.036076035 -0.018095543
13  1.2 -0.059186994 -0.039569089 -0.019797777
14  1.3 -0.064600945 -0.043097380 -0.021509532
15  1.4 -0.070086754 -0.046660779 -0.023230804
16  1.5 -0.075643667 -0.050259141 -0.024961586
```

```
17  1.6 -0.081270853 -0.053892306 -0.026701871
18  1.7 -0.086967406 -0.057560096 -0.028451651
19  1.8 -0.092732341 -0.061262318 -0.030210915
20  1.9 -0.098564595 -0.064998762 -0.031979654
21  2.0 -0.104463027 -0.068769201 -0.033757856
```

```
#saveRDS(mt_dmso_lost, "../data/motility_total_DMSO_lost.rds")
```

This is the lost in total mobility that the model predicts.

Ethanol

```
mt_etoh <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "total motility_EtOH")
mt_etoh$donor <- as.factor(mt_etoh$donor)
names(mt_etoh) <- c("donor", "conc", "motile", "total")
mt_etoh$motile_frac <- mt_etoh$motile / mt_etoh$total
skimr::skim(mt_etoh)
```

Table 4: Data summary

Name	mt_etoh
Number of rows	40
Number of columns	5
_____	
Column type frequency:	
factor	1
numeric	4
_____	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
donor	0	1	FALSE	8	1: 5, 2: 5, 3: 5, 4: 5

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
conc	0	1	0.72	0.74	0.0	0.10	0.50	1.00	2.00	
motile	0	1	92.67	45.35	39.0	61.50	82.50	104.25	238.00	
total	0	1	127.20	56.93	66.0	87.00	117.00	140.25	301.00	
motile_frac	0	1	0.73	0.12	0.4	0.67	0.78	0.81	0.88	

There are also the same eight donors, no missing data.

```
table(mt_etoh$donor, as.factor(mt_etoh$conc))
```

```
  0 0.1 0.5 1 2
1 1   1   1 1 1
2 1   1   1 1 1
3 1   1   1 1 1
4 1   1   1 1 1
5 1   1   1 1 1
6 1   1   1 1 1
7 1   1   1 1 1
8 1   1   1 1 1
```

The data are balanced with one observation for each concentration and each donor and no missing data.

```
mt_etoh_m1 <- glmer(cbind(motile, total - motile) ~ conc + (conc | donor),
  data = mt_etoh, family = binomial(link = "logit"))
summary(mt_etoh_m1)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(motile, total - motile) ~ conc + (conc | donor)
   Data: mt_etoh

     AIC      BIC   logLik deviance df.resid
   320.3    328.7   -155.1    310.3       35

Scaled residuals:
    Min      1Q  Median      3Q     Max
```

```
-4.9916 -0.7375  0.0028  0.9353  1.8933

Random effects:
 Groups Name        Variance Std.Dev. Corr
 donor  (Intercept) 0.16977  0.4120
        conc        0.02106  0.1451   -0.02
Number of obs: 40, groups:  donor, 8

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.26479    0.15390   8.219  < 2e-16 ***
conc        -0.29582    0.06886  -4.296 1.74e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.163
```

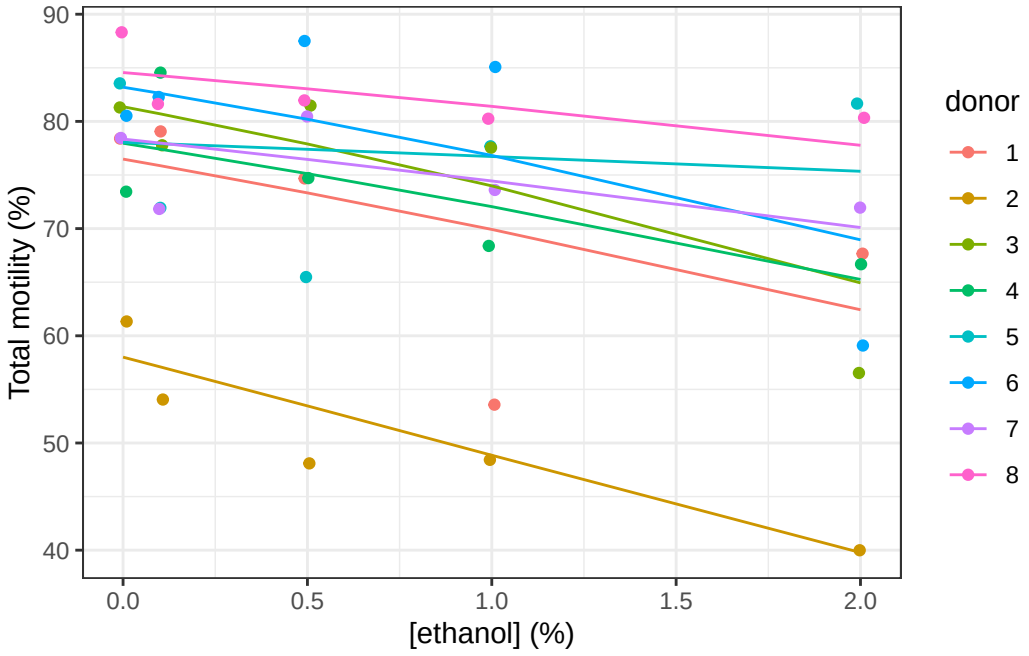
The model is:

,

$$
\begin{aligned}
&\text{motile}_i \sim \text{Binomial}(n = 1, \text{prob}_{\text{motile}=1} = \widehat{P}) \\
&\log\left[\frac{\hat{P}}{1-\hat{P}}\right] = \alpha_{j[i]} + \beta_{1j[i]}(\text{conc}) \\
&\begin{pmatrix} \alpha_j \\ \beta_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_{\alpha_j} \\ \mu_{\beta_{1j}} \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}} \end{pmatrix}\right), \text{for donor j} = 1, \ldots, \text{J}
\end{aligned}
\qquad (3)
$$

Here is a plot of this model:

```
ggplot(data = mt_etoh) +
  geom_jitter(aes(x = conc, y = motile_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mt_etoh_m1) * 100, col = donor)) +
  labs(x = "[ethanol] (%)", y = "Total motility (%)")
```



Generally, slopes are all negative, suggesting a negative concentration effect. Data are rather widespread. There are definitely shifts in the intercept per donor (different motility at [ethanol] = 0). Slopes seems more different between donors than for DMSO. However, we will also check if the model can be simplified using a likelihood ratio test:

```
mt_etoh_m2 <- glmer(cbind(motile, total - motile) ~ conc  + (1 | donor),
  data = mt_etoh, family = binomial(link = "logit"))
anova(mt_etoh_m1, mt_etoh_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

```
Data: mt_etoh
Models:
mt_etoh_m2: cbind(motile, total - motile) ~ conc + (1 | donor)
mt_etoh_m1: cbind(motile, total - motile) ~ conc + (conc | donor)
           npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
mt_etoh_m2    3 319.58 324.65 -156.79   313.58
mt_etoh_m1    5 320.28 328.72 -155.14   310.28 3.302  2     0.1919
```

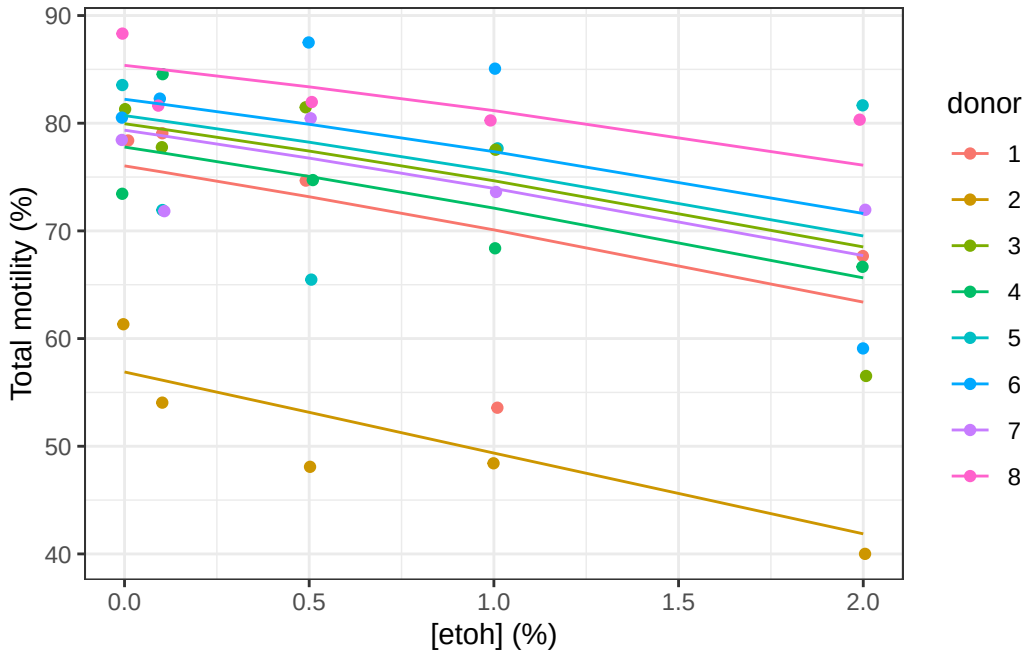
The likelihood ratio test does not detects significant differences between the full and simplified models at $\alpha = 5\%$. We could thus use the simplest `mt_etoh_m2` model with only a shift in the slope per donor. This model is:

,

$$
\begin{aligned}
\text{motile}_i &\sim \text{Binomial}(n = 1, \text{prob}_{\text{motile}=1} = \widehat{P}) \\
, \log\left[\frac{\hat{P}}{1-\hat{P}}\right] &= \alpha_{j[i]} + \beta_1(\text{conc}) \qquad , \qquad (4) \\
\alpha_j &\sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{for donor j} = 1, \ldots, \text{J}
\end{aligned}
$$

Here is a plot of this model:

```
ggplot(data = mt_etoh) +
  geom_jitter(aes(x = conc, y = motile_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mt_etoh_m2) * 100, col = donor)) +
  labs(x = "[etoh] (%)", y = "Total motility (%)")
```




```
summary(mt_etoh_m2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(motile, total - motile) ~ conc + (1 | donor)
```

```
   Data: mt_etoh

     AIC      BIC   logLik deviance df.resid
   319.6    324.6   -156.8    313.6       37

Scaled residuals:
    Min      1Q  Median      3Q     Max
-5.0521 -0.5711  0.0252  0.9661  2.8884

Random effects:
 Groups Name        Variance Std.Dev.
 donor  (Intercept) 0.1794   0.4236
Number of obs: 40, groups:  donor, 8

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.26983    0.15723   8.076 6.69e-16 ***
conc        -0.30297    0.04266  -7.103 1.22e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.209
```

The Z test indicates that `conc` is significantly different from zero at $\alpha = 5\%$. However, it is not the best test in the case of a mixed model like here. We prefer to rely on the 95% confidence interval calculated either on the profile, or via parametric bootstrap (and especially the later one):

```
confint(mt_etoh_m2, level = 0.95) # 95% CI based on profile
```

```
Computing profile confidence intervals ...

                2.5 %     97.5 %
.sig01       0.2695168  0.7719890
(Intercept)  0.9261419  1.6176274
conc        -0.3865530 -0.2192498
```

```r
set.seed(535)
# 1000x parameter bootstrap
(mt_etoh_m2_conf <- confint(mt_etoh_m2, level = 0.95,
  method = "boot", nsim  = 1000L))
```

```
Computing bootstrap confidence intervals ...
```

```
4 warning(s): Model failed to converge with max|grad| = 0.00611004 (tol = 0.002, component 1)
```

```
                 2.5 %     97.5 %
.sig01       0.1811891  0.6041279
(Intercept)  0.9728800  1.5769631
conc        -0.3882783 -0.2237793
```

We had one model with singularity among the 1000, not a big problem (we may ignore this warning). Slope for `conc` is significantly different from zero at $\alpha = 5\%$ because the 95% CI does not contain zero.

Additional verifications

We could double-check the significance of the slope `conc` by looking at a likelihood ratio test when dropping `conc` from the model:

```r
#drop1(mt_etoh_m2, scope = "conc")
mt_etoh_m3 <- glmer(cbind(motile, total - motile) ~ 1 + (1 | donor),
  data = mt_etoh, family = binomial(link = "logit"))
anova(mt_etoh_m2, mt_etoh_m3, refit = FALSE)
```

```
Data: mt_etoh
Models:
mt_etoh_m3: cbind(motile, total - motile) ~ 1 + (1 | donor)
mt_etoh_m2: cbind(motile, total - motile) ~ conc + (1 | donor)
           npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
mt_etoh_m3    2 367.46 370.84 -181.73   363.46
mt_etoh_m2    3 319.58 324.65 -156.79   313.58 49.88  1  1.635e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with `conc` is significantly different at α level 5% from a reference model that sets the slope `conc` = 0. There is thus a significant effect of ethanol concentration (confirmation of results obtained from 95% CI).

We also double-check convergence of the model by trying different optimisation engines (just to make sure). First, is there a singularity in the model?

```
isSingular(mt_etoh_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```
mt_etoh_m2_all <- allFit(mt_etoh_m2)
```

```
bobyqa : [OK]
Nelder_Mead : [OK]
nlminbwrap : [OK]
nmkbw : [OK]
optimx.L-BFGS-B : [OK]
nloptwrap.NLOPT_LN_NELDERMEAD : [OK]
nloptwrap.NLOPT_LN_BOBYQA : [OK]
```

```
summary(mt_etoh_m2_all)
```

```
$which.OK
                         bobyqa                     Nelder_Mead
                           TRUE                            TRUE
                     nlminbwrap                          nmkbw
                           TRUE                            TRUE
                optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                           TRUE                            TRUE
    nloptwrap.NLOPT_LN_BOBYQA
                           TRUE

$msgs
$msgs$bobyqa
NULL
```

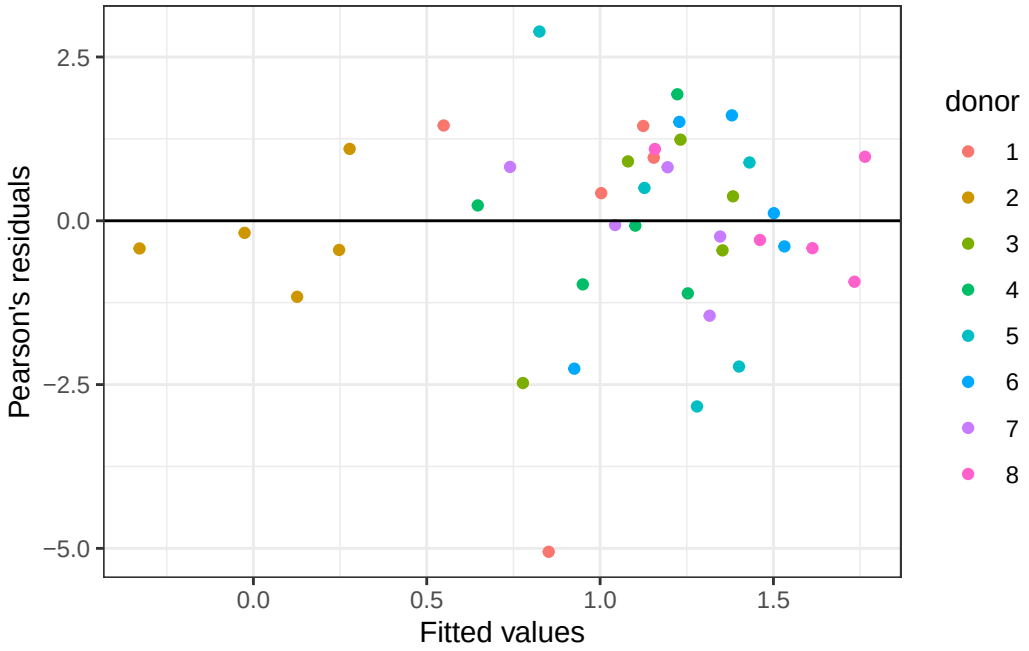
```
$msgs$Nelder_Mead
NULL

$msgs$nlminbwrap
NULL

$msgs$nmkbw
NULL

$msgs$`optimx.L-BFGS-B`
NULL

$msgs$nloptwrap.NLOPT_LN_NELDERMEAD
NULL

$msgs$nloptwrap.NLOPT_LN_BOBYQA
NULL


$fixef
                             (Intercept)       conc
bobyqa                         1.269828 -0.3029685
Nelder_Mead                    1.269826 -0.3029668
nlminbwrap                     1.269829 -0.3029702
nmkbw                          1.269732 -0.3029680
optimx.L-BFGS-B                1.269829 -0.3029684
nloptwrap.NLOPT_LN_NELDERMEAD  1.269756 -0.3029708
nloptwrap.NLOPT_LN_BOBYQA      1.269828 -0.3029680

$llik
                      bobyqa                    Nelder_Mead
                    -156.7893                      -156.7893
                   nlminbwrap                          nmkbw
                    -156.7893                      -156.7893
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                    -156.7893                      -156.7893
    nloptwrap.NLOPT_LN_BOBYQA
                    -156.7893

$sdcor
                             donor.(Intercept)
bobyqa                               0.4235709
Nelder_Mead                          0.4235679
```

```
nlminbwrap                             0.4235741
nmkbw                                  0.4235690
optimx.L-BFGS-B                        0.4235601
nloptwrap.NLOPT_LN_NELDERMEAD          0.4235924
nloptwrap.NLOPT_LN_BOBYQA              0.4235698

$theta
                              donor.(Intercept)
bobyqa                                0.4235709
Nelder_Mead                           0.4235679
nlminbwrap                            0.4235741
nmkbw                                 0.4235690
optimx.L-BFGS-B                       0.4235601
nloptwrap.NLOPT_LN_NELDERMEAD         0.4235924
nloptwrap.NLOPT_LN_BOBYQA             0.4235698

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.045    0.000   0.046          0         0
Nelder_Mead                       0.057    0.001   0.058          0         0
nlminbwrap                        0.049    0.000   0.050          0         0
nmkbw                             0.055    0.000   0.056          0         0
optimx.L-BFGS-B                   0.363    0.003   0.368          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.054    0.000   0.055          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.042    0.001   0.043          0         0

$feval
                         bobyqa                     Nelder_Mead
                             65                              88
                      nlminbwrap                           nmkbw
                             NA                             100
                optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                             14                              92
    nloptwrap.NLOPT_LN_BOBYQA
                             31

attr(,"class")
[1] "summary.allFit"
```

Analysis of the residuals

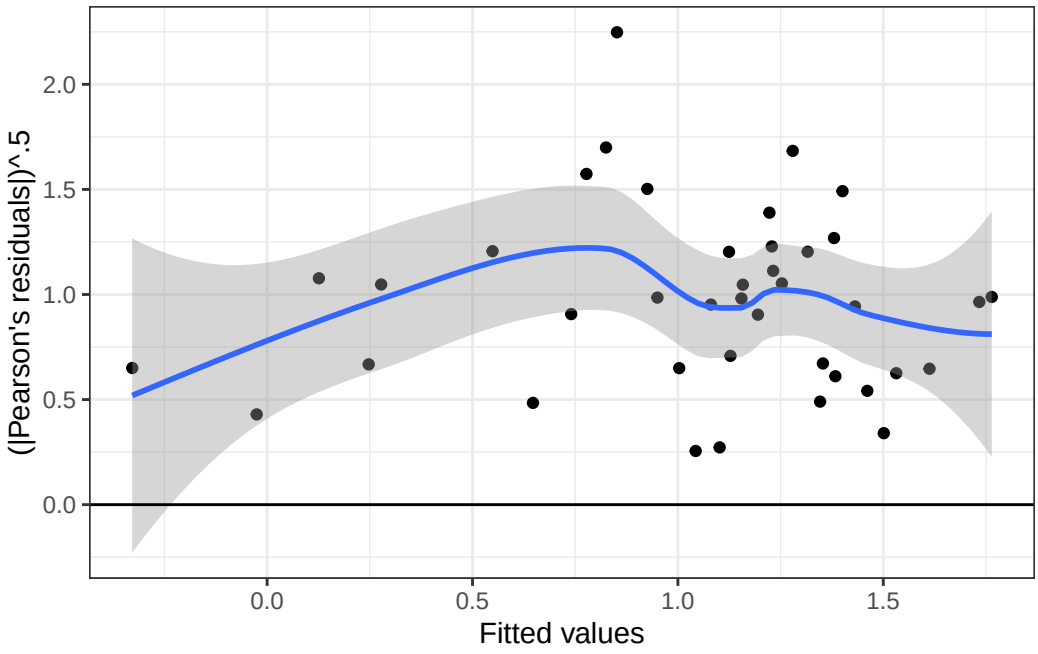
Let's check how the residuals distribute and if there is homoscedasticity.

```
mt_etoh <- fortify.merMod(mt_etoh_m2)
ggplot(data = mt_etoh, aes(x = .fitted, y = .scresid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")
```



There is one extreme value (less extreme than for DMSO), but otherwise, residuals seem rather correctly distributed. Linearity is good here.

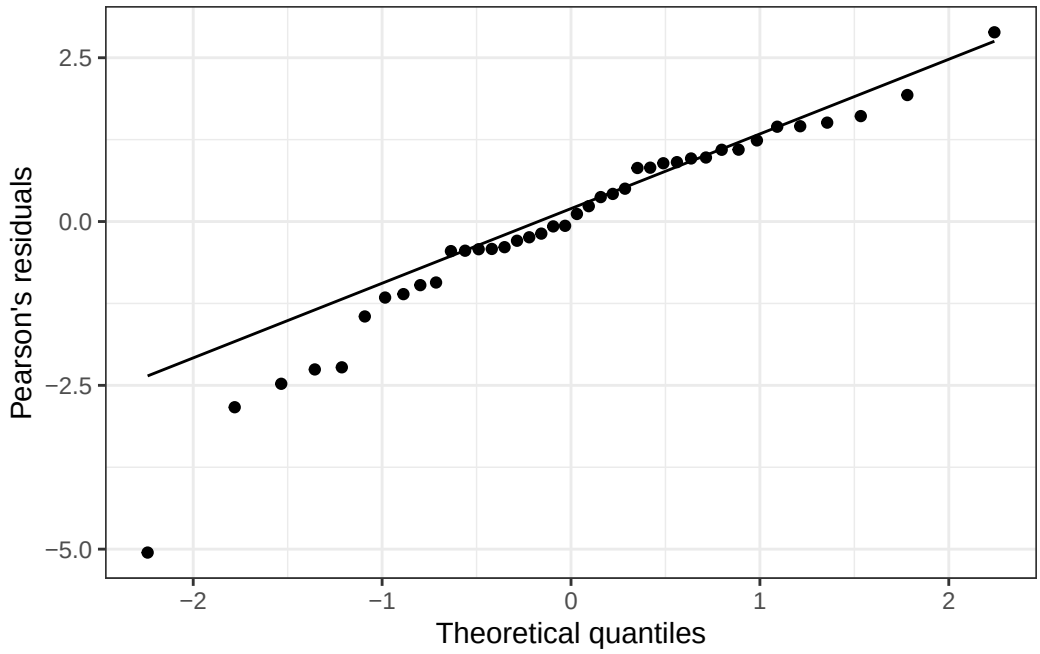
```
ggplot(data = mt_etoh, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals seems here acceptable (the blue curve that is a loess smoothing in the data is relatively horizontal).

With the same remark as for DMSO, here is the quantile-quantile plot of the residuals:

```
ggplot(data = mt_etoh, aes(sample = .scresid)) +
  geom_qq() +
  geom_qq_line() +
  labs(x = "Theoretical quantiles", y = "Pearson's residuals")
```



It appears not too good. A Shapiro-Wilk test indicates mild non Normality (with caution because this test tends to be conservative):

```
shapiro.test(mt_etoh$.scresid)
```

```
    Shapiro-Wilk normality test

data:  mt_etoh$.scresid
W = 0.94028, p-value = 0.03534
```

Predictions

The model allows to calculate the drop in total mobility according to ethanol concentration from 0 to 2%. Note that an inverse logit transformation is required. Here is an example:

```
mt_etoh_slope <- c(
  ci95_min   = min(mt_etoh_m2_conf["conc", ]),
  estimate = fixef(mt_etoh_m2)[["conc"]],
  ci95_max  = max(mt_etoh_m2_conf["conc", ]))
mt_etoh_slope
```

```
  ci95_min   estimate   ci95_max
-0.3882783 -0.3029708 -0.2237793
```

```
#saveRDS(mt_etoh_slope, "../data/motility_total_ETOH_slope.rds")
```

Let's say we want to calculate the drop in total mobility for various ethanol concentrations between 0 and 2% if the total mobility in a sample without ethanol is 80%. The calculation is:

```
predict_logit <- function(conc, intercept = 1, slopes) {
  slopes_mat <- matrix(mt_etoh_slope, nrow = 1,
    dimnames = list(NULL, names(mt_etoh_slope)))
  data.frame(conc = conc, -intercept +
    boot::inv.logit(boot::logit(intercept) +
    conc %*% slopes_mat))
}
etoh_conc <- (0:20) / 10
mt_etoh_lost <- predict_logit(etoh_conc, 0.8, mt_etoh_slope)
mt_etoh_lost
```

```
   conc     ci95_min     estimate     ci95_max
1   0.0  0.000000000  0.000000000  0.000000000
2   0.1 -0.006284872 -0.004891619 -0.003604517
3   0.2 -0.012714728 -0.009871491 -0.007257167
4   0.3 -0.019289630 -0.014939680 -0.010957988
5   0.4 -0.026009417 -0.020096167 -0.014706993
6   0.5 -0.032873700 -0.025340853 -0.018504173
7   0.6 -0.039881848 -0.030673549 -0.022349493
8   0.7 -0.047032985 -0.036093983 -0.026242894
9   0.8 -0.054325977 -0.041601787 -0.030184290
10  0.9 -0.061759427 -0.047196504 -0.034173569
11  1.0 -0.069331667 -0.052877577 -0.038210591
12  1.1 -0.077040754 -0.058644355 -0.042295190
13  1.2 -0.084884466 -0.064496085 -0.046427171
14  1.3 -0.092860294 -0.070431914 -0.050606312
15  1.4 -0.100965441 -0.076450885 -0.054832359
16  1.5 -0.109196822 -0.082551937 -0.059105030
17  1.6 -0.117551062 -0.088733904 -0.063424016
18  1.7 -0.126024496 -0.094995513 -0.067788973
19  1.8 -0.134613171 -0.101335385 -0.072199529
20  1.9 -0.143312852 -0.107752031 -0.076655281
```

```
21  2.0 -0.152119022 -0.114243858 -0.081155794
```

```
#saveRDS(mt_etoh_lost, "../data/motility_total_ETOH_lost.rds")
```

This is the lost in total mobility that the model predicts.

General informations

```
sessionInfo()
```

```
R version 4.1.3 (2022-03-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur/Monterey 10.16

Matrix products: default
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] ggplot2_3.3.5 lme4_1.1-29   Matrix_1.4-1

loaded via a namespace (and not attached):
 [1] tidyr_1.2.0         jsonlite_1.8.0      splines_4.1.3
 [4] equatiomatic_0.3.1  shiny_1.7.1         assertthat_0.2.1
 [7] highr_0.9           broom.mixed_0.2.9.4 cellranger_1.1.0
[10] yaml_2.3.5          globals_0.14.0      numDeriv_2016.8-1.1
[13] pillar_1.7.0        backports_1.4.1     lattice_0.20-45
[16] glue_1.6.2          digest_0.6.29       promises_1.2.0.1
[19] minqa_1.2.4         colorspace_2.0-3    dfoptim_2020.10-1
[22] htmltools_0.5.2     httpuv_1.6.5        pkgconfig_2.0.3
[25] broom_0.8.0         listenv_0.8.0       purrr_0.3.4
[28] xtable_1.8-4        scales_1.2.0        later_1.3.0
[31] tibble_3.1.6        mgcv_1.8-40         generics_0.1.2
[34] farver_2.1.0        ellipsis_0.3.2      withr_2.5.0
[37] furrr_0.2.3         repr_1.1.4          skimr_2.1.4
```

```
[40] cli_3.2.0           magrittr_2.0.3      crayon_1.5.1
[43] readxl_1.4.0        mime_0.12           evaluate_0.15
[46] fansi_1.0.3         future_1.24.0       parallelly_1.31.0
[49] nlme_3.1-157        MASS_7.3-56         forcats_0.5.1
[52] tools_4.1.3         lifecycle_1.0.1     stringr_1.4.0
[55] munsell_0.5.0       compiler_4.1.3      rlang_1.0.2
[58] grid_4.1.3          nloptr_2.0.0        rstudioapi_0.13
[61] base64enc_0.1-3     labeling_0.4.2      rmarkdown_2.13
[64] boot_1.3-28         gtable_0.3.0        codetools_0.2-18
[67] DBI_1.1.2           R6_2.5.1            knitr_1.38
[70] dplyr_1.0.8         optimx_2021-10.12   fastmap_1.1.0
[73] utf8_1.2.2          stringi_1.7.6       parallel_4.1.3
[76] Rcpp_1.0.8.3        vctrs_0.4.1         tidyselect_1.1.2
[79] xfun_0.30
```