Capacitation

Marie Bisconti, Philippe Grosjean & Elise Hennebert

Introduction

The effect of DMSO and ethanol is evaluated at concentrations from 0 up to 2% on spermatozoa capacitation (i.e., phosphotyrosines).

DMSO

```
cap_dmso <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "capacitation_DMSO")
cap_dmso$donor <- as.factor(cap_dmso$donor)
names(cap_dmso) <- c("donor", "conc", "capa")
skimr::skim(cap_dmso)
```

Table 1: Data summary

Name	cap_dmso
Number of rows	25
Number of columns	3
Column type frequency:	
factor	1
numeric	2
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
donor	0	1	FALSE	5	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	0.75	0.00	0.10	0.50	1.00	2.00	
capa	0	1	0.93	0.29	0.46	0.78	0.94	1.06	1.55	

There are five donors, no missing data.

```
table(cap_dmso$donor, as.factor(cap_dmso$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
```

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

```
cap_dmso_m1 <- lmer(capa ~ conc + (conc | donor), data = cap_dmso)
```

```
boundary (singular) fit: see help('isSingular')
```

```
summary(cap_dmso_m1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: capa ~ conc + (conc | donor)
   Data: cap_dmso

REML criterion at convergence: 9.7
```

```
Scaled residuals:
     Min       1Q   Median       3Q      Max 
-1.76685 -0.49863 -0.02128  0.45054  1.77530 

Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 donor    (Intercept) 0.0120077 0.10958      
          conc        0.0002396 0.01548  1.00
 Residual             0.0604968 0.24596      
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)    
(Intercept)  1.04475    0.08469  5.57516  12.336 2.94e-05 ***
conc        -0.15598    0.06770 17.57874  -2.304   0.0337 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.510
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

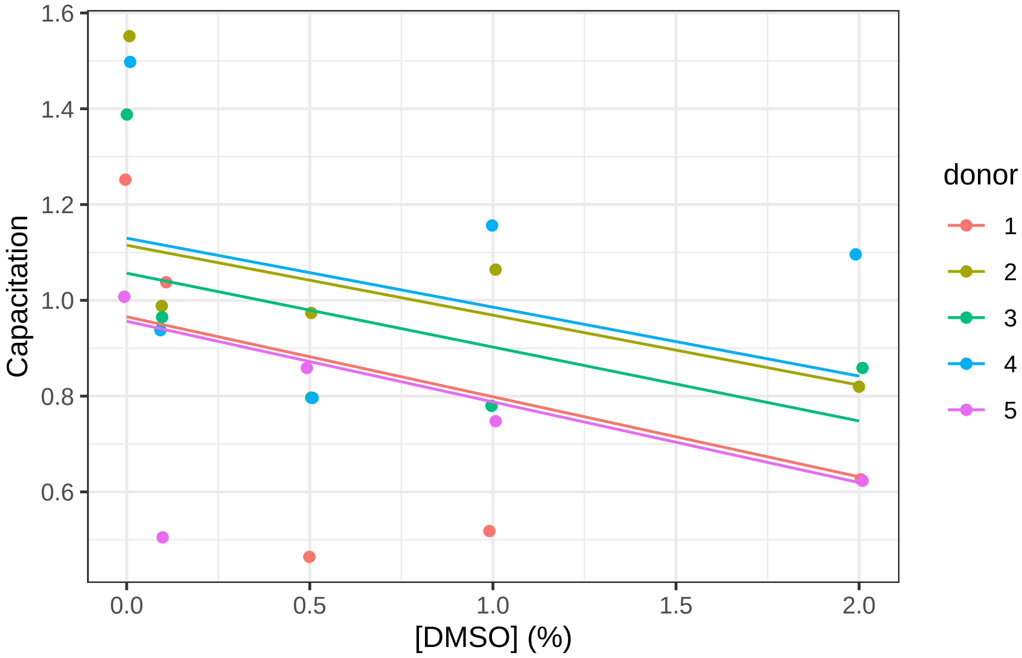
The fit is singular, due to a parameter evaluated at the boundary (correlation parameter between `conc` and `donor` is one in the random effect). The model is:

,

$$\begin{aligned} &\text{capa}_i \sim N\left(\alpha_{j[i]} + \beta_{1j[i]}(\text{conc}), \sigma^2\right) \\ ,&\begin{pmatrix}\alpha_j \\ \beta_{1j}\end{pmatrix} \sim N\left(\begin{pmatrix}\mu_{\alpha_j} \\ \mu_{\beta_{1j}}\end{pmatrix}, \begin{pmatrix}\sigma^2_{\alpha_j} & \rho_{\alpha_j\beta_{1j}} \\ \rho_{\beta_{1j}\alpha_j} & \sigma^2_{\beta_{1j}}\end{pmatrix}\right), \text{for donor j} = 1, \ldots, \text{J}' \end{aligned} \tag{1}$$

Here is a plot of this model:

```
ggplot(data = cap_dmso) +
  geom_jitter(aes(x = conc, y = capa, col = donor), width = 0.01) +
  geom_line(aes(x = conc, y = fitted(cap_dmso_m1), col = donor)) +
  labs(x = "[DMSO] (%)", y = "Capacitation")
```



Slopes are all negative, suggesting a negative concentration effect. Data are rather widespread. Lines are rather parallel, and a simplification of the model should also deal with the singularity (only different intercepts for `donor`). Let's check it with a likelihood ratio test:

```
ranova(cap_dmso_m1, reduce.terms = TRUE)
```

```
ANOVA-like table for random-effects: Single term deletions

Model:
capa ~ conc + (conc | donor)
                       npar  logLik    AIC      LRT Df Pr(>Chisq)
<none>                    6 -4.8716 21.743
conc in (conc | donor)    4 -4.8892 17.778 0.035031  2     0.9826
```

or (this is the same):

```
cap_dmso_m2 <- lmer(capa ~ conc + (1 | donor), data = cap_dmso)
anova(cap_dmso_m1, cap_dmso_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

```
Data: cap_dmso
```

```
Models:
cap_dmso_m2: capa ~ conc + (1 | donor)
cap_dmso_m1: capa ~ conc + (conc | donor)
            npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
cap_dmso_m2    4 17.778 22.654 -4.8892   9.7783
cap_dmso_m1    6 21.743 29.057 -4.8716   9.7433 0.035  2     0.9826
```

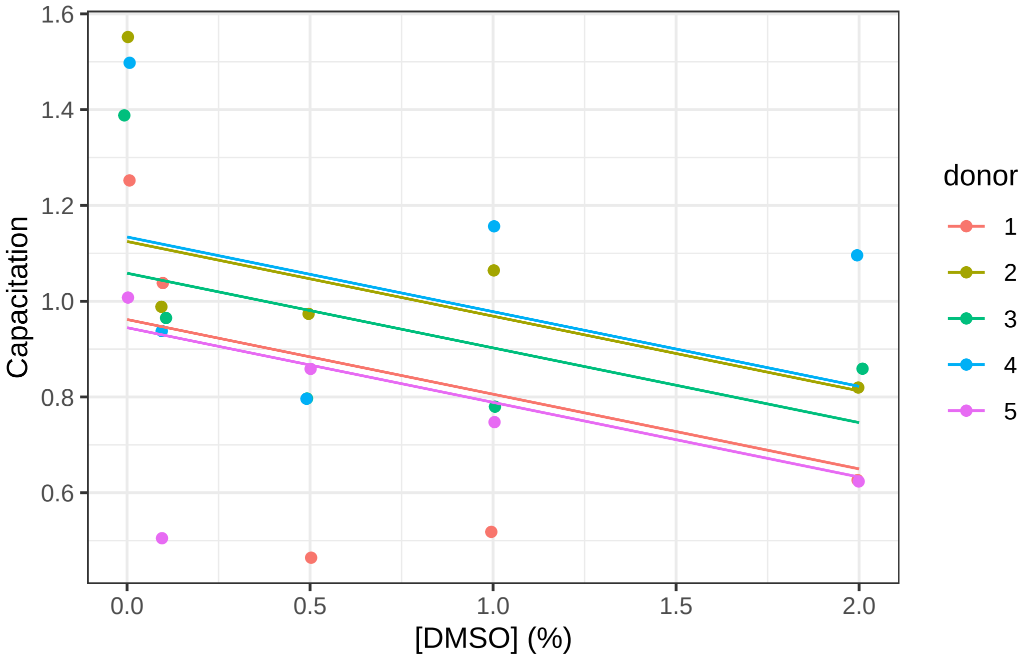
The likelihood ratio test (models *not* refitted using ML, not necessary because fixed effects are the same between the two models) does not detect significant differences between the full and simplified random effect term at $\alpha = 5\%$. We could thus use the simplest `cap_dmso_m2` model with only a shift in the slope per donor. This model is:

,

$$,\begin{aligned} \text{capa}_i &\sim N\left(\alpha_{j[i]} + \beta_1(\text{conc}), \sigma^2\right) \\ \alpha_j &\sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{ for donor j} = 1, \ldots, \text{J}' \end{aligned} \qquad (2)$$

Here is a plot of this model:

```r
ggplot(data = cap_dmso) +
  geom_jitter(aes(x = conc, y = capa, col = donor), width = 0.01) +
  geom_line(aes(x = conc, y = fitted(cap_dmso_m2), col = donor)) +
  labs(x = "[DMSO] (%)", y = "Capacitation")
```



Here is a summary of the final model:

```r
summary(cap_dmso_m2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: capa ~ conc + (1 | donor)
   Data: cap_dmso

REML criterion at convergence: 9.8

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-1.7214 -0.4981 -0.0374  0.4564  1.7331 

Random effects:
 Groups   Name        Variance Std.Dev.
 donor    (Intercept) 0.01447  0.1203  
 Residual             0.06070  0.2464  
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)    
(Intercept)  1.04475    0.08764  8.00119  11.921 2.25e-06 ***
conc        -0.15598    0.06746 19.00000  -2.312   0.0321 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.554
```

The t test indicates that `conc` is significantly different from zero at $\alpha = 5\%$, but not at $\alpha = 1\%$. Yet, t test is not the best one 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):

```r
confint(cap_dmso_m2, level = 0.95) # 95% CI based on profile
```

```
Computing profile confidence intervals ...
```

```
                 2.5 %      97.5 %
.sig01       0.0000000  0.29941678
.sigma       0.1813240  0.33894333
(Intercept)  0.8683698  1.22113067
conc        -0.2912820 -0.02066896
```

```
set.seed(52)
# 1000x parameter bootstrap
(cap_dmso_m2_conf <- confint(cap_dmso_m2, level = 0.95, method = "boot", nsim  = 1000L))
```

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

```
213 message(s): boundary (singular) fit: see help('isSingular')
```

```
                 2.5 %      97.5 %
.sig01       0.0000000  0.25281969
.sigma       0.1656595  0.31286299
(Intercept)  0.8774568  1.22378981
conc        -0.2885403 -0.02125504
```

About 1/5 of the bootstrapped models were singular. However, the 95%ICs calculated from profiles and using bootstrap do not differ much. So, we could trust these results. The slope `conc` significantly different to zero at $\alpha = 5\%$ because the 95% CIs do not contain zero (but it is very close to it at its minimum boundary).

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:

```
drop.scope(terms(cap_dmso_m2))
```

```
[1] "conc"
```

```
drop1(cap_dmso_m2, scope = "conc")
```

```
Single term deletions using Satterthwaite's method:

Model:
capa ~ conc + (1 | donor)
      Sum Sq Mean Sq NumDF DenDF F value  Pr(>F)
conc 0.32454 0.32454     1    19  5.3465 0.03213 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

or:

```r
cap_dmso_m3 <- lmer(capa ~ 1 + (1 | donor), data = cap_dmso)
anova(cap_dmso_m2, cap_dmso_m3, refit = TRUE)
```

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


Data: cap_dmso
Models:
cap_dmso_m3: capa ~ 1 + (1 | donor)
cap_dmso_m2: capa ~ conc + (1 | donor)
            npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
cap_dmso_m3    3 13.643 17.300 -3.8215   7.6429
cap_dmso_m2    4 10.684 15.559 -1.3420   2.6839 4.959  1    0.02596 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with a non zero slope is significantly different at α level 5% from a reference model using horizontal lines. 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?

```r
isSingular(cap_dmso_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```r
cap_dmso_m2_all <- allFit(cap_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]
```

```r
summary(cap_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
```

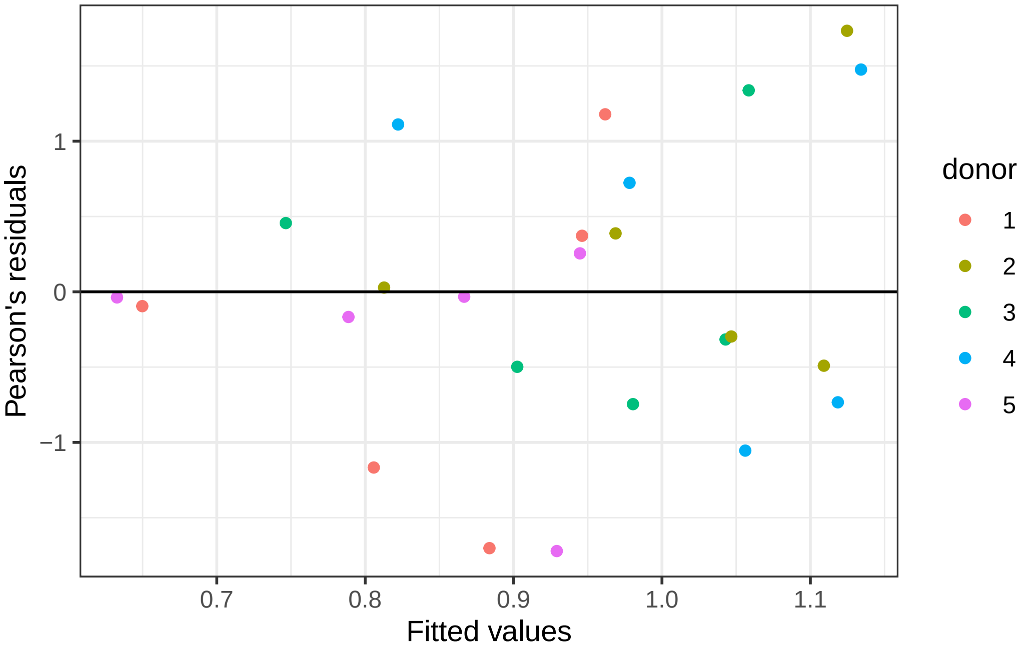
```
$msgs$nloptwrap.NLOPT_LN_NELDERMEAD
NULL

$msgs$nloptwrap.NLOPT_LN_BOBYQA
NULL


$fixef
                              (Intercept)       conc
bobyqa                            1.04475 -0.1559755
Nelder_Mead                       1.04475 -0.1559755
nlminbwrap                        1.04475 -0.1559755
nmkbw                             1.04475 -0.1559755
optimx.L-BFGS-B                   1.04475 -0.1559755
nloptwrap.NLOPT_LN_NELDERMEAD     1.04475 -0.1559755
nloptwrap.NLOPT_LN_BOBYQA         1.04475 -0.1559755

$llik
                       bobyqa                   Nelder_Mead
                    -4.889164                     -4.889164
                   nlminbwrap                         nmkbw
                    -4.889164                     -4.889164
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                    -4.889164                     -4.889164
    nloptwrap.NLOPT_LN_BOBYQA
                    -4.889164

$sdcor
                              donor.(Intercept)     sigma
bobyqa                                0.1202798 0.2463765
Nelder_Mead                           0.1202790 0.2463767
nlminbwrap                            0.1202798 0.2463765
nmkbw                                 0.1202798 0.2463765
optimx.L-BFGS-B                       0.1202798 0.2463765
nloptwrap.NLOPT_LN_NELDERMEAD         0.1202839 0.2463756
nloptwrap.NLOPT_LN_BOBYQA             0.1202798 0.2463765

$theta
                              donor.(Intercept)
bobyqa                                0.4881950
Nelder_Mead                           0.4881915
nlminbwrap                            0.4881950
nmkbw                                 0.4881950
```

```
optimx.L-BFGS-B                       0.4881953
nloptwrap.NLOPT_LN_NELDERMEAD         0.4882133
nloptwrap.NLOPT_LN_BOBYQA             0.4881950

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.035    0.000   0.035          0         0
Nelder_Mead                       0.039    0.001   0.039          0         0
nlminbwrap                        0.035    0.000   0.035          0         0
nmkbw                             0.034    0.000   0.035          0         0
optimx.L-BFGS-B                   0.189    0.000   0.189          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.037    0.000   0.037          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.038    0.000   0.038          0         0

$feval
                       bobyqa                   Nelder_Mead
                           16                            32
                   nlminbwrap                         nmkbw
                           NA                            NA
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                           14                            28
    nloptwrap.NLOPT_LN_BOBYQA
                           12

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

Analysis of the residuals

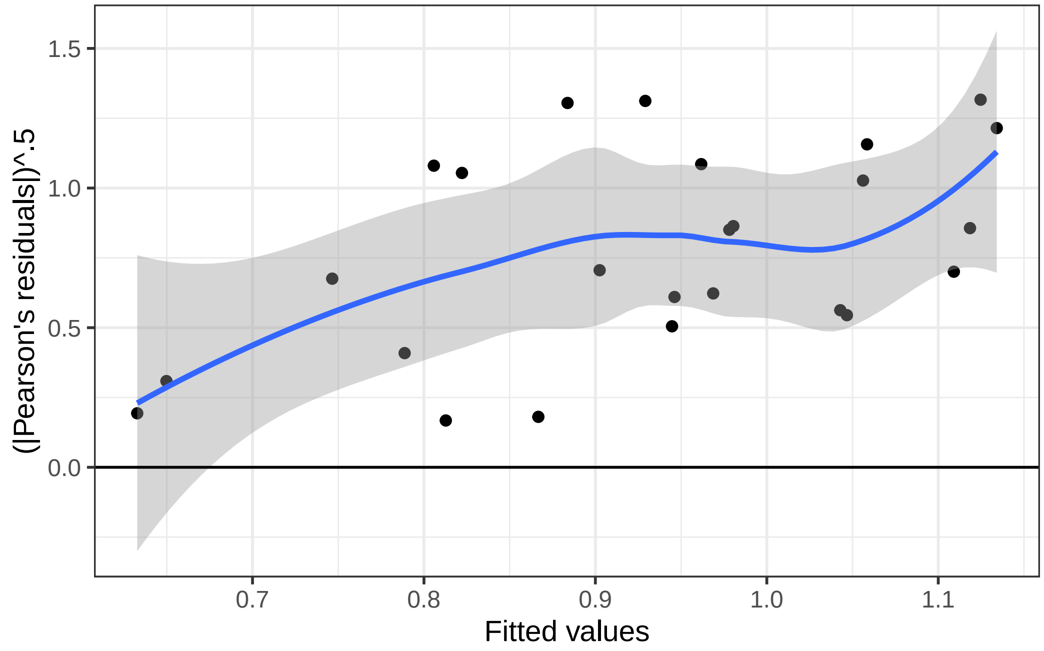
Let's check how the residuals distribute and if there is homoscedasticity.

```r
cap_dmso <- fortify.merMod(cap_dmso_m2)
ggplot(data = cap_dmso, aes(x = .fitted, y = .scresid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")
```



Residuals seem rather correctly distributed. Linearity is good here.

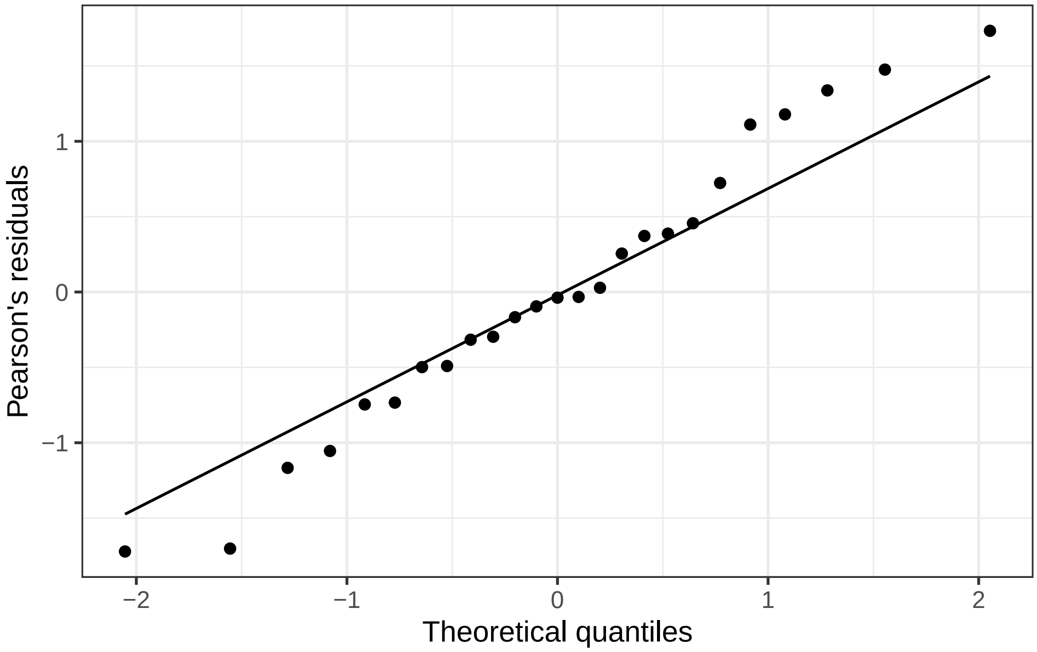
```
ggplot(data = cap_dmso, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals does not seem to be matched here, but on a closer look, the lower variance for fitted values < 0.7 is due to having only two points in this area. So, we cannot conclude against homoscedasticity, since without these two points, the rest is OK. The amount of data is a little scarce here.

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:

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



It appears not that bad. A Shapiro-Wilk test is not against Normality either:

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

```

	Shapiro-Wilk normality test

data:  cap_dmso$.scresid
W = 0.97655, p-value = 0.8094
```

Predictions

The model allows to calculate the drop in capacitation according to DMSO concentration from 0 to 2%.

```
cap_dmso_slope <- c(
  ci95_min   = min(cap_dmso_m2_conf["conc", ]),
  estimate = fixef(cap_dmso_m2)[["conc"]],
  ci95_max  = max(cap_dmso_m2_conf["conc", ]))
cap_dmso_slope
```

```
   ci95_min    estimate    ci95_max 
-0.28854026 -0.15597549 -0.02125504 
```

```
#saveRDS(cap_dmso_slope, "../data/capacitation_DMSO_slope.rds")
```

Let's say we want to calculate the drop in capacitation for various DMSO concentrations between 0 and 2% if the capacitation of a sample without DMSO is 1.1. The calculation is:

```
predict_with_ci <- function(conc, intercept = 1, slopes) {
  slopes_mat <- matrix(slopes, nrow = 1,
    dimnames = list(NULL, names(slopes)))
  data.frame(conc = conc, -intercept +
      (intercept + conc %*% slopes_mat))
}
dmso_conc <- (0:20) / 10
cap_dmso_lost <- predict_with_ci(dmso_conc, 1.1, cap_dmso_slope)
cap_dmso_lost
```

```
   conc    ci95_min    estimate     ci95_max
1   0.0  0.00000000  0.00000000  0.000000000
2   0.1 -0.02885403 -0.01559755 -0.002125504
3   0.2 -0.05770805 -0.03119510 -0.004251007
4   0.3 -0.08656208 -0.04679265 -0.006376511
5   0.4 -0.11541610 -0.06239020 -0.008502014
6   0.5 -0.14427013 -0.07798775 -0.010627518
7   0.6 -0.17312416 -0.09358529 -0.012753021
8   0.7 -0.20197818 -0.10918284 -0.014878525
9   0.8 -0.23083221 -0.12478039 -0.017004028
10  0.9 -0.25968623 -0.14037794 -0.019129532
11  1.0 -0.28854026 -0.15597549 -0.021255035
12  1.1 -0.31739428 -0.17157304 -0.023380539
13  1.2 -0.34624831 -0.18717059 -0.025506042
14  1.3 -0.37510234 -0.20276814 -0.027631546
15  1.4 -0.40395636 -0.21836569 -0.029757049
16  1.5 -0.43281039 -0.23396324 -0.031882553
17  1.6 -0.46166441 -0.24956079 -0.034008057
18  1.7 -0.49051844 -0.26515834 -0.036133560
19  1.8 -0.51937247 -0.28075588 -0.038259064
20  1.9 -0.54822649 -0.29635343 -0.040384567
21  2.0 -0.57708052 -0.31195098 -0.042510071
```

```
#saveRDS(cap_dmso_lost, "../data/capacitation_DMSO_lost.rds")
```

This is the lost in capacitation that the model predicts with 95%CIs.

Ethanol

```
cap_etoh <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "capacitation_EtOH")
cap_etoh$donor <- as.factor(cap_etoh$donor)
names(cap_etoh) <- c("donor", "conc", "capa")
skimr::skim(cap_etoh)
```

Table 4: Data summary

Name	cap_etoh
Number of rows	25
Number of columns	3
Column type frequency:	
factor	1
numeric	2
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
donor	0	1	FALSE	5	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	0.75	0.00	0.10	0.50	1.00	2.00	
capa	0	1	1.04	0.29	0.24	0.88	1.02	1.24	1.57	

There are five donors, no missing data.

```
table(cap_etoh$donor, as.factor(cap_etoh$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
```

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

```
cap_etoh_m1 <- lmer(capa ~ conc + (conc | donor), data = cap_etoh)
```

```
boundary (singular) fit: see help('isSingular')
```

```
Warning: Model failed to converge with 1 negative eigenvalue: -4.6e+00
```

```
summary(cap_etoh_m1)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: capa ~ conc + (conc | donor)
   Data: cap_etoh

REML criterion at convergence: 11.2

Scaled residuals:
    Min      1Q  Median      3Q     Max 
-1.6871 -0.6063  0.0791  0.6660  1.6183 

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 donor    (Intercept) 0.000000 0.00000      
          conc        0.008346 0.09136   NaN
 Residual             0.067898 0.26057      
```

```
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)    
(Intercept)  1.14388    0.07317 18.86241  15.632    3e-12 ***
conc        -0.15010    0.08221  9.01039  -1.826    0.101    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.609
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

The fit is singular and the model failed to converge. We now try with a simplified random term where only the intercept depends on donor, like for DMSO.

```r
cap_etoh_m2 <- lmer(capa ~ conc + (1 | donor), data = cap_etoh)
summary(cap_etoh_m2)
```

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: capa ~ conc + (1 | donor)
   Data: cap_etoh

REML criterion at convergence: 11.6

Scaled residuals: 
     Min       1Q   Median       3Q      Max 
-2.07340 -0.63780  0.01602  0.73795  1.56525 

Random effects:
 Groups   Name        Variance Std.Dev.
 donor    (Intercept) 0.004107 0.06409 
 Residual             0.071961 0.26826 
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error       df t value Pr(>|t|)    
(Intercept)  1.14388    0.08060 11.00773  14.192 2.02e-08 ***
```

```
conc        -0.15010    0.07345 19.00000  -2.044   0.0551 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.656
```

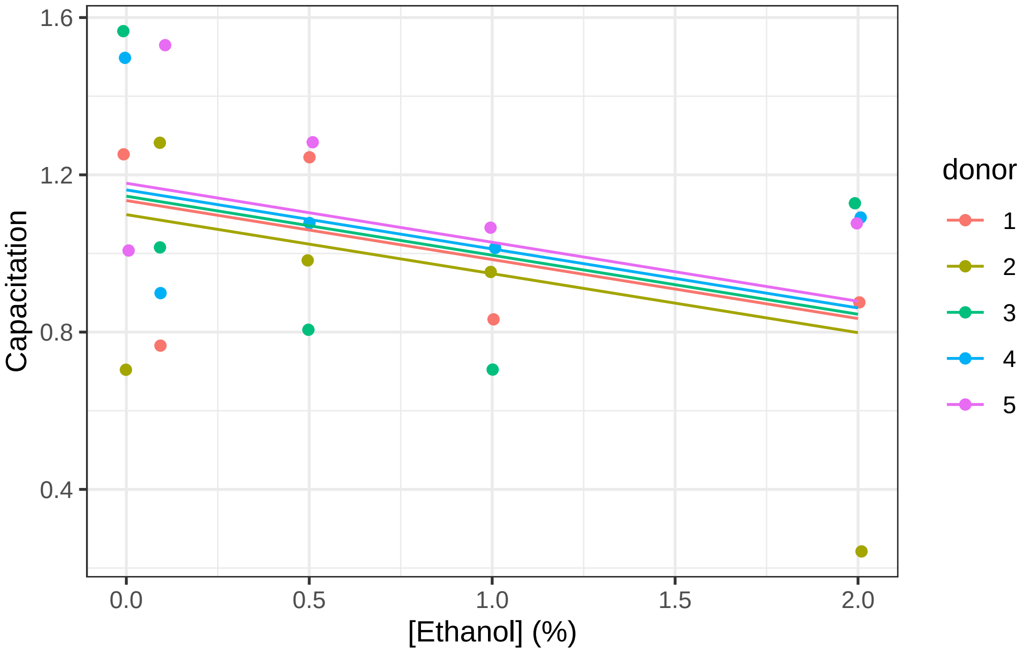
The simplified model fits well. According to t test, the slope is (just) not significantly different to zero at $\alpha = 5\%$. This model is:

,

$$
,\begin{aligned}
\text{capa}_i &\sim N\left(\alpha_{j[i]} + \beta_1(\text{conc}), \sigma^2\right) \\
\alpha_j &\sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{ for donor j} = 1, \ldots, \text{J}
\end{aligned}, \tag{3}
$$

Here is a plot of this model:

```
ggplot(data = cap_etoh) +
  geom_jitter(aes(x = conc, y = capa, col = donor), width = 0.01) +
  geom_line(aes(x = conc, y = fitted(cap_etoh_m2), col = donor)) +
  labs(x = "[Ethanol] (%)", y = "Capacitation")
```



Data are very widespread. Here, there seems to be less differences from one donor to the other. However, the random term `donor` in the model accounts for the repeated measures (same donor for different concentrations). Hence, this term *cannot* be dropped, even if it appears to be non siginificant. Otherwise, we will end up with a model that does not take correlation of observations for a the same donor into account and it would be a pseudo-replication error!

To check if the slope for `conc`‘is different from zero, we prefer to rely on 95% confidence intervals, especially those calculated using parametric bootstrap:

```
confint(cap_etoh_m2, level = 0.95) # 95% CI based on profile
```

```
Computing profile confidence intervals ...
```

```
                2.5 %       97.5 %
.sig01       0.0000000  0.238410794
.sigma       0.1974261  0.357537791
(Intercept)  0.9850698  1.302680613
conc        -0.2969104 -0.003292061
```

```
set.seed(874356)
# 1000x parameter bootstrap
(cap_etoh_m2_conf <- confint(cap_etoh_m2, level = 0.95,
  method = "boot", nsim  = 1000L))
```

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

```
429 message(s): boundary (singular) fit: see help('isSingular')
```

```
                2.5 %       97.5 %
.sig01       0.0000000  0.204865243
.sigma       0.1739216  0.345822810
(Intercept)  0.9945774  1.294731632
conc        -0.2827724 -0.004787316
```

Here almost 1/2 of the bootstrapped samples led to singularity (probably because the slope `conc`‘was very close to zero). However, the 95CIs are still similar to those calculated from the profile of our original model that was correctly fitted. The slope `conc` appears to be significantly different from zero at $\alpha = 5\%$ because the 95% CIs do not contain zero (but it is very, very close to it at its minimum boundary). In this case, we should redo the analysis with a larger set of data to confirm or inform the slope is different from zero. Anyway, one could analyze the upper boundary of the 95%CI to determine if the effect might be problematic here or not. If not, it is not necessary to further investigate.

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
drop.scope(terms(cap_etoh_m2))
```

```
[1] "conc"
```

```r
drop1(cap_etoh_m2, scope = "conc")
```

```
Single term deletions using Satterthwaite's method:

Model:
capa ~ conc + (1 | donor)
      Sum Sq Mean Sq NumDF DenDF F value Pr(>F)
conc 0.30056 0.30056     1    19  4.1766 0.0551 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

or:

```r
cap_etoh_m3 <- lmer(capa ~ 1 + (1 | donor), data = cap_etoh)
anova(cap_etoh_m2, cap_etoh_m3, refit = TRUE)
```

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

Data: cap_etoh
Models:
cap_etoh_m3: capa ~ 1 + (1 | donor)
cap_etoh_m2: capa ~ conc + (1 | donor)
            npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
cap_etoh_m3    3 14.272 17.928 -4.1359   8.2718
```

```
cap_etoh_m2    4 12.270 17.145 -2.1349   4.2698 4.002  1    0.04545 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with a non zero slope is just significantly different at α level 5% from a reference model using horizontal lines. 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(cap_etoh_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```
cap_etoh_m2_all <- allFit(cap_etoh_m2)
```

```
bobyqa : [OK]
Nelder_Mead : [OK]
nlminbwrap :

boundary (singular) fit: see help('isSingular')

Warning: Model failed to converge with 1 negative eigenvalue: -9.0e+00

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

```
summary(cap_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
[1] "boundary (singular) fit: see help('isSingular')"

$msgs$nmkbw
NULL

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

$msgs$nloptwrap.NLOPT_LN_NELDERMEAD
NULL

$msgs$nloptwrap.NLOPT_LN_BOBYQA
NULL


$fixef
                              (Intercept)       conc
bobyqa                           1.143875 -0.1501013
Nelder_Mead                      1.143875 -0.1501013
nlminbwrap                       1.143875 -0.1501013
nmkbw                            1.143875 -0.1501013
optimx.L-BFGS-B                  1.143875 -0.1501013
nloptwrap.NLOPT_LN_NELDERMEAD    1.143875 -0.1501013
nloptwrap.NLOPT_LN_BOBYQA        1.143875 -0.1501013
```

```
$llik
                       bobyqa                   Nelder_Mead
                     -5.77876                      -5.77876
                   nlminbwrap                         nmkbw
                     -5.83370                      -5.77876
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                     -5.77876                      -5.77876
    nloptwrap.NLOPT_LN_BOBYQA
                     -5.77876

$sdcor
                              donor.(Intercept)     sigma
bobyqa                             6.408808e-02 0.2682554
Nelder_Mead                        6.408892e-02 0.2682553
nlminbwrap                         6.052857e-07 0.2748318
nmkbw                              6.408808e-02 0.2682554
optimx.L-BFGS-B                    6.408784e-02 0.2682555
nloptwrap.NLOPT_LN_NELDERMEAD      6.409170e-02 0.2682548
nloptwrap.NLOPT_LN_BOBYQA          6.408808e-02 0.2682554

$theta
                              donor.(Intercept)
bobyqa                             2.389069e-01
Nelder_Mead                        2.389102e-01
nlminbwrap                         2.202386e-06
nmkbw                              2.389069e-01
optimx.L-BFGS-B                    2.389060e-01
nloptwrap.NLOPT_LN_NELDERMEAD      2.389210e-01
nloptwrap.NLOPT_LN_BOBYQA          2.389069e-01

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.035        0   0.036          0         0
Nelder_Mead                       0.040        0   0.040          0         0
nlminbwrap                        0.035        0   0.035          0         0
nmkbw                             0.035        0   0.035          0         0
optimx.L-BFGS-B                   0.186        0   0.187          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.037        0   0.037          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.037        0   0.037          0         0

$feval
                       bobyqa                   Nelder_Mead
                           16                            36
```

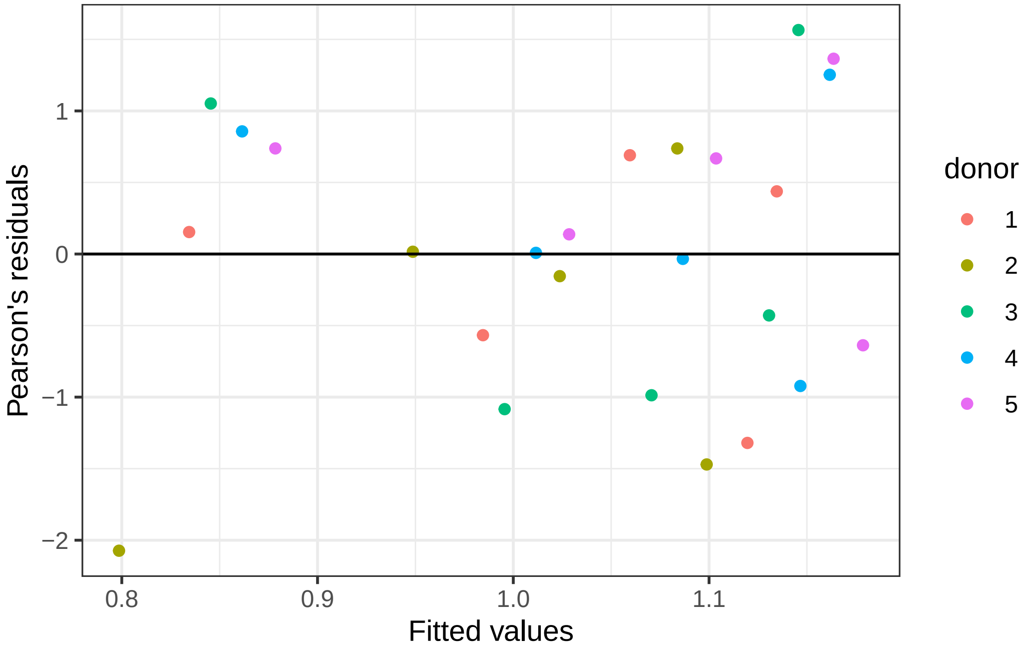
```
                  nlminbwrap                          nmkbw
                          NA                             NA
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                          24                             31
    nloptwrap.NLOPT_LN_BOBYQA
                          13

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

Only the nlminbwrap algorithm with default parameters was not able to fit the mode. For the other algorithms, the convergence towards the same solution suggests we got probably a global optimum.

Analysis of the residuals

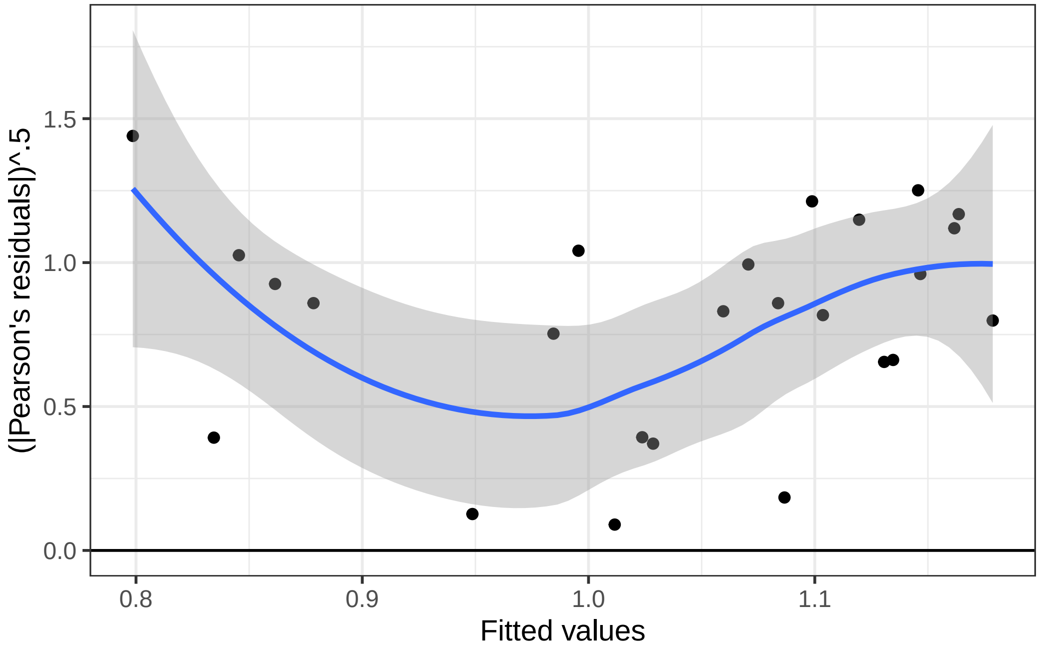
Let's check how the residuals distribute and if there is homoscedasticity.

```
cap_etoh <- fortify.merMod(cap_etoh_m2)
ggplot(data = cap_etoh, aes(x = .fitted, y = .scresid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")
```



Residuals seem rather correctly distributed. Linearity is good here.

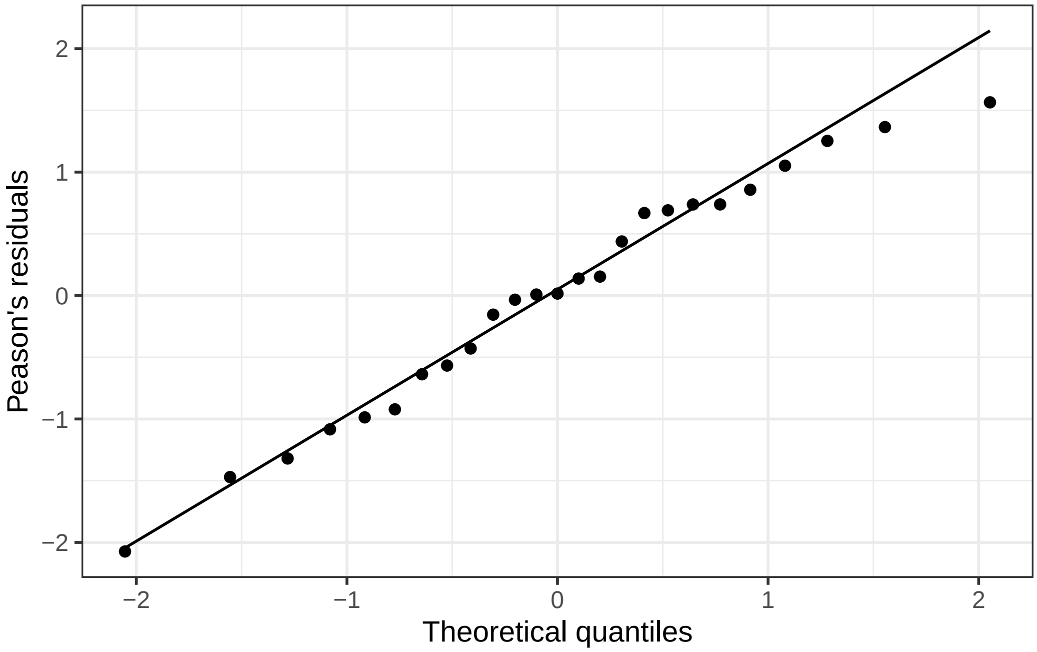
```
ggplot(data = cap_etoh, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals does not seem too bad. The amount of data is a little scarce here for low fitted values < 0.95.

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 = cap_etoh, aes(sample = .scresid)) +
  geom_qq() +
  geom_qq_line() +
  labs(x = "Theoretical quantiles", y = "Peason's residuals")
```



It appears rather good. A Shapiro-Wilk test is not against Normality either:

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

```


	Shapiro-Wilk normality test

data:  cap_etoh$.scresid
W = 0.97416, p-value = 0.7508
```

Predictions

The model allows to calculate the drop in capacitation according to ethanol concentration from 0 to 2%.

```
cap_etoh_slope <- c(
  ci95_min   = min(cap_etoh_m2_conf["conc", ]),
  estimate = fixef(cap_etoh_m2)[["conc"]],
  ci95_max  = max(cap_etoh_m2_conf["conc", ]))
cap_etoh_slope
```

```
    ci95_min     estimate     ci95_max 
-0.282772367 -0.150101260 -0.004787316 
```

```r
#saveRDS(cap_etoh_slope, "../data/capacitation_ETOH_slope.rds")
```

Let's say we want to calculate the drop in capacitation for various ethanol concentrations between 0 and 2% if the capacitation of a sample without ethanol is 1.1. The calculation is:

```r
predict_with_ci <- function(conc, intercept = 1, slopes) {
  slopes_mat <- matrix(slopes, nrow = 1,
    dimnames = list(NULL, names(slopes)))
  data.frame(conc = conc, -intercept +
      (intercept + conc %*% slopes_mat))
}
etoh_conc <- (0:20) / 10
cap_etoh_lost <- predict_with_ci(etoh_conc, 1.1, cap_etoh_slope)
cap_etoh_lost
```

```
   conc    ci95_min    estimate      ci95_max
1   0.0  0.00000000  0.00000000  0.0000000000
2   0.1 -0.02827724 -0.01501013 -0.0004787316
3   0.2 -0.05655447 -0.03002025 -0.0009574632
4   0.3 -0.08483171 -0.04503038 -0.0014361948
5   0.4 -0.11310895 -0.06004050 -0.0019149264
6   0.5 -0.14138618 -0.07505063 -0.0023936579
7   0.6 -0.16966342 -0.09006076 -0.0028723895
8   0.7 -0.19794066 -0.10507088 -0.0033511211
9   0.8 -0.22621789 -0.12008101 -0.0038298527
10  0.9 -0.25449513 -0.13509113 -0.0043085843
11  1.0 -0.28277237 -0.15010126 -0.0047873159
12  1.1 -0.31104960 -0.16511139 -0.0052660475
13  1.2 -0.33932684 -0.18012151 -0.0057447791
14  1.3 -0.36760408 -0.19513164 -0.0062235106
15  1.4 -0.39588131 -0.21014176 -0.0067022422
16  1.5 -0.42415855 -0.22515189 -0.0071809738
17  1.6 -0.45243579 -0.24016202 -0.0076597054
18  1.7 -0.48071302 -0.25517214 -0.0081384370
19  1.8 -0.50899026 -0.27018227 -0.0086171686
20  1.9 -0.53726750 -0.28519239 -0.0090959002
21  2.0 -0.56554473 -0.30020252 -0.0095746318
```

```
#saveRDS(cap_etoh_lost, "../data/capacitation_ETOH_lost.rds")
```

This is the lost in capacitation that the model predicts with 95%CIs.

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  lmerTest_3.1-3 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
```