

# Acrosome integrity

Marie Bisconti, Philippe Grosjean & Elise Hennebert

## Introduction

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

## DMSO

```
ai_dms0 <- readxl::read_excel("../data/Table S2.xlsx",  
  sheet = "acrosome integrity_DMSO")  
ai_dms0$donor <- as.factor(ai_dms0$donor)  
names(ai_dms0) <- c("donor", "conc", "acrointact", "total")  
ai_dms0$acrointact_frac <- ai_dms0$acrointact / ai_dms0$total  
skimr::skim(ai_dms0)
```

Table 1: Data summary

Name	ai_dms0
Number of rows	20
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	4	6: 5, 7: 5, 8: 5, 9: 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	
acrointact	0	1	184.85	5.80	168.00	183.00	186.00	189.00	191.00	
total	0	1	200.45	2.01	200.00	200.00	200.00	200.00	209.00	
acrointact_frac	0	1	0.92	0.03	0.84	0.91	0.92	0.94	0.96	

There are four donors, no missing data.

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

```

0 0.1 0.5 1 2
6 1   1   1 1 1
7 1   1   1 1 1
8 1   1   1 1 1
9 1   1   1 1 1

```

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

```
ai_dmso_m1 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (conc | donor),
  data = ai_dmso, family = binomial(link = "logit"))
summary(ai_dmso_m1)
```

Generalized linear mixed model fit by maximum likelihood (Laplace  
Approximation) [glmerMod]

Family: binomial ( logit )

Formula: cbind(acrointact, total - acrointact) ~ conc + (conc | donor)

Data: ai\_dmso

AIC	BIC	logLik	deviance	df.resid
115.9	120.9	-53.0	105.9	15

Scaled residuals:

	Min	1Q	Median	3Q	Max
	-1.04554	-0.48370	-0.02401	0.56732	0.96683

Random effects:

Groups Name	Variance	Std.Dev.	Corr
donor (Intercept)	0.004637	0.0681	
conc	0.026459	0.1627	1.00

Number of obs: 20, groups: donor, 4

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	2.69266	0.09485	28.390	<2e-16 ***
conc	-0.25370	0.11347	-2.236	0.0254 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)
conc	-0.219

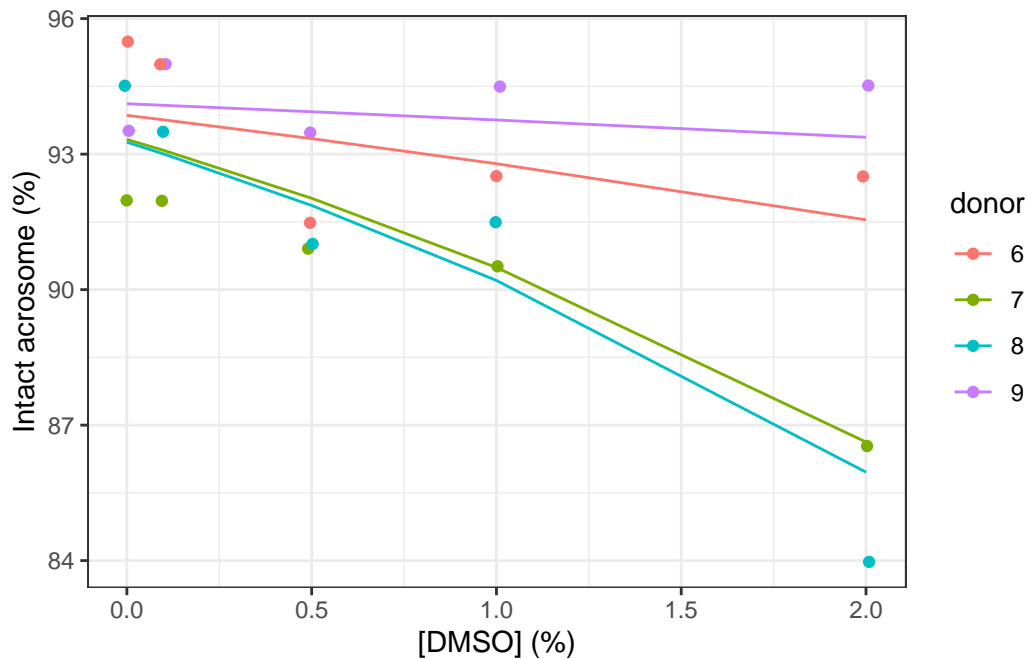
The model is:

,

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

Here is a plot of this model:

```
ggplot(data = ai_dmsso) +
  geom_jitter(aes(x = conc, y = acrointact_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(ai_dmsso_m1) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Intact acrosome (%)")
```



Generally, slopes are all negative, suggesting a negative concentration effect. Data are rather widespread. Shifts in the intercept per donor is not obvious here, but change in slope is much more marked. We may try simplifying the model so that only slopes vary between donors. Let's check it with a likelihood ratio test:

```
ai_dms0_m2 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (0 + conc:donor | donor)
data = ai_dms0, family = binomial(link = "logit"))
```

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

```
anova(ai_dms0_m1, ai_dms0_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

Data: ai\_dms0

Models:

ai\_dms0\_m1: cbind(acrointact, total - acrointact) ~ conc + (conc | donor)

ai\_dms0\_m2: cbind(acrointact, total - acrointact) ~ conc + (0 + conc:donor | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
ai_dms0_m1	5	115.92	120.89	-52.958	105.92			
ai_dms0_m2	12	129.20	141.15	-52.601	105.20	0.7123	7	0.9982

The likelihood ratio test does not detect significant differences between the full and simplified models at  $\alpha = 5\%$ . But... the model had a problem because we had a singularity. Let's try the simplification where the random effect `donor` only accounts for the intercept:

```
ai_dmso_m3 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (1 | donor),
  data = ai_dmso, family = binomial(link = "logit"))
anova(ai_dmso_m1, ai_dmso_m3, refit = FALSE) # Despite the name, it is indeed a LR test
```

Data: ai\_dmso

Models:

```
ai_dmso_m3: cbind(acrointact, total - acrointact) ~ conc + (1 | donor)
ai_dmso_m1: cbind(acrointact, total - acrointact) ~ conc + (conc | donor)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
ai_dmso_m3	3	115.50	118.49	-54.752	109.50			
ai_dmso_m1	5	115.92	120.89	-52.958	105.92	3.5899	2	0.1661

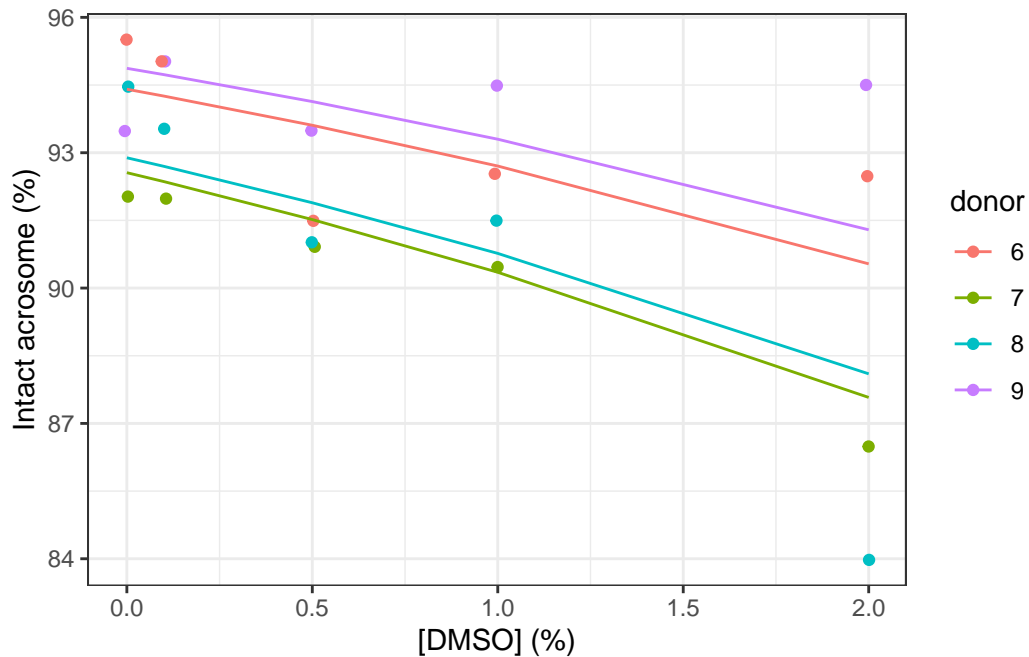
Here we draw the same conclusion, but this time our model fits without any problems. Let's continue our analysis with this simpler model `ai_dmso_m3`. This model is:

,

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

Here is a plot of this model:

```
ggplot(data = ai_dmso) +
  geom_jitter(aes(x = conc, y = acrointact_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(ai_dmso_m3) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Intact acrosome (%)")
```



Visually, it seems not too bad, but it seems we have the two last points for donor 9 suggesting a smaller slope and last point for donor 8 in favour of a larger slope. Here, we have too few data points to really decide what is the best model. However, considering all the other variables studied here, a model with intercept depending on the donor is not to be rejected (it is clearly the best model wherever more data are available).

```
summary(ai_dms0_m3)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) [glmerMod]
Family: binomial ( logit )
Formula: cbind(acrointact, total - acrointact) ~ conc + (1 | donor)
Data: ai_dms0
```

AIC	BIC	logLik	deviance	df.resid
115.5	118.5	-54.8	109.5	17

Scaled residuals:

Min	1Q	Median	3Q	Max
-1.7904	-0.4015	-0.0199	0.5077	1.6093

Random effects:

```

Groups Name      Variance Std.Dev.
donor (Intercept) 0.03849  0.1962
Number of obs: 20, groups: donor, 4

```

Fixed effects:

```

              Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.71161     0.13232  20.493 < 2e-16 ***
conc         -0.28386     0.07669  -3.701 0.000214 ***
---

```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

```

      (Intr)
conc -0.494

```

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(ai_dms0_m3, level = 0.95) # 95% CI based on profile

```

Computing profile confidence intervals ...

```

              2.5 %      97.5 %
.sig01      0.05229171  0.5531516
(Intercept)  2.40826337  3.0303645
conc        -0.43348853 -0.1322598

```

```

set.seed(96347)
# 1000x parameter bootstrap
(ai_dms0_m3_conf <- confint(ai_dms0_m3, level = 0.95, method = "boot", nsim = 1000L))

```

Computing bootstrap confidence intervals ...

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

6 warning(s): Model failed to converge with max|grad| = 0.00227339 (tol = 0.002, component 1)

	2.5 %	97.5 %
.sig01	0.0000000	0.3531096
(Intercept)	2.4379061	2.9935609
conc	-0.4339087	-0.1238000

1/5 of bootstrapped models present singularities. However, 95%CI from profiles and for parametric bootstrap are very close. So, we can trust them. 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:

```
#drop1(ai_dms0_m3, scope = "conc")
ai_dms0_m4 <- glmer(cbind(acrointact, total - acrointact) ~ 1 + (1 | donor),
  data = ai_dms0, family = binomial(link = "logit"))
anova(ai_dms0_m3, ai_dms0_m4, refit = TRUE)
```

Data: ai\_dms0

Models:

ai\_dms0\_m4: cbind(acrointact, total - acrointact) ~ 1 + (1 | donor)

ai\_dms0\_m3: cbind(acrointact, total - acrointact) ~ conc + (1 | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
ai_dms0_m4	2	126.77	128.76	-61.386	122.77			
ai_dms0_m3	3	115.50	118.49	-54.752	109.50	13.268	1	0.00027 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

The model with `conc` is significantly different at  $\alpha$  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(ai_dms0_m3)
```

```
[1] FALSE
```

... then, a report about the model convergence:



```
ai_dmso_m3_all <- allFit(ai_dmso_m3)
```

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(ai_dmso_m3_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	2.711615	-0.2838597
Nelder_Mead	2.711619	-0.2838580
nlminbwrap	2.711617	-0.2838619
nmkbw	2.711609	-0.2838352
optimx.L-BFGS-B	2.711615	-0.2838595
nloptwrap.NLOPT_LN_NELDERMEAD	2.711510	-0.2838224
nloptwrap.NLOPT_LN_BOBYQA	2.711618	-0.2838643

\$llik	bobyqa	Nelder_Mead
	-54.75245	-54.75245
	nlminbwrap	nmkbw
	-54.75245	-54.75245
	optimx.L-BFGS-B	nloptwrap.NLOPT_LN_NELDERMEAD
	-54.75245	-54.75245
	nloptwrap.NLOPT_LN_BOBYQA	
	-54.75245	

\$sdcor	donor.(Intercept)
bobyqa	0.1961798
Nelder_Mead	0.1961796
nlminbwrap	0.1961816
nmkbw	0.1961364
optimx.L-BFGS-B	0.1961824
nloptwrap.NLOPT_LN_NELDERMEAD	0.1962364
nloptwrap.NLOPT_LN_BOBYQA	0.1961850

\$theta	donor.(Intercept)
bobyqa	0.1961798

```

Nelder_Mead                0.1961796
nlminbwrap                  0.1961816
nmkbw                       0.1961364
optimx.L-BFGS-B            0.1961824
nloptwrap.NLOPT_LN_NELDERMEAD 0.1962364
nloptwrap.NLOPT_LN_BOBYQA   0.1961850

$times
               user.self sys.self elapsed user.child sys.child
bobyqa          0.042    0.000   0.041         0         0
Nelder_Mead     0.055    0.001   0.055         0         0
nlminbwrap      0.042    0.000   0.041         0         0
nmkbw           0.058    0.000   0.059         0         0
optimx.L-BFGS-B 0.343    0.001   0.345         0         0
nloptwrap.NLOPT_LN_NELDERMEAD 0.052    0.000   0.051         0         0
nloptwrap.NLOPT_LN_BOBYQA   0.041    0.000   0.042         0         0

$feval
               bobyqa               Nelder_Mead
               54               93
               nlminbwrap               nmkbw
               NA               111
               optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
               16               84
               nloptwrap.NLOPT_LN_BOBYQA
               36

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

```

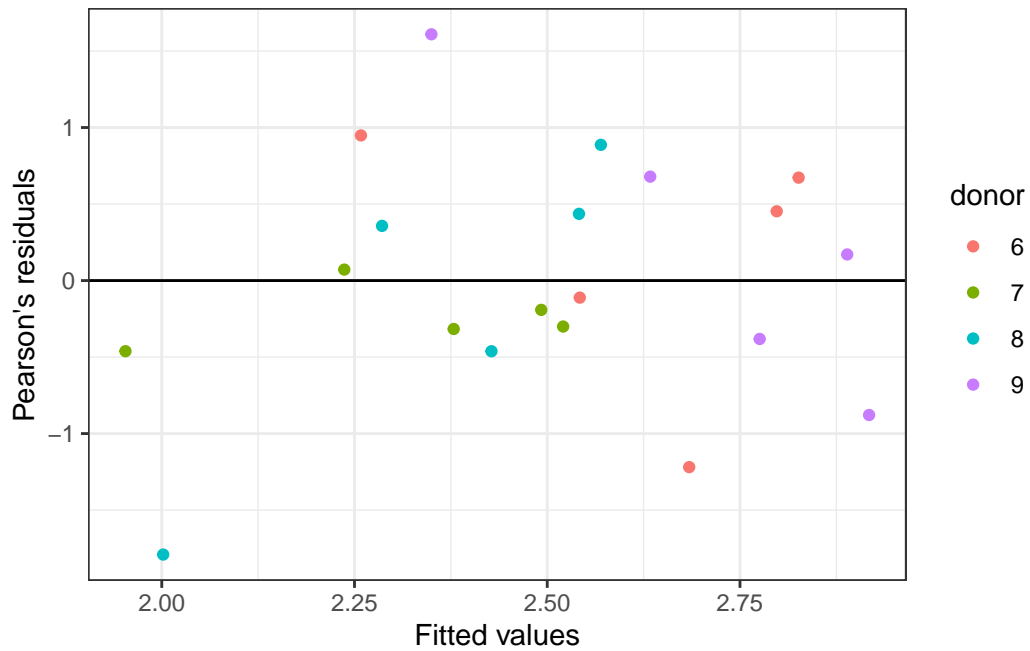
## Analysis of the residuals

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

```

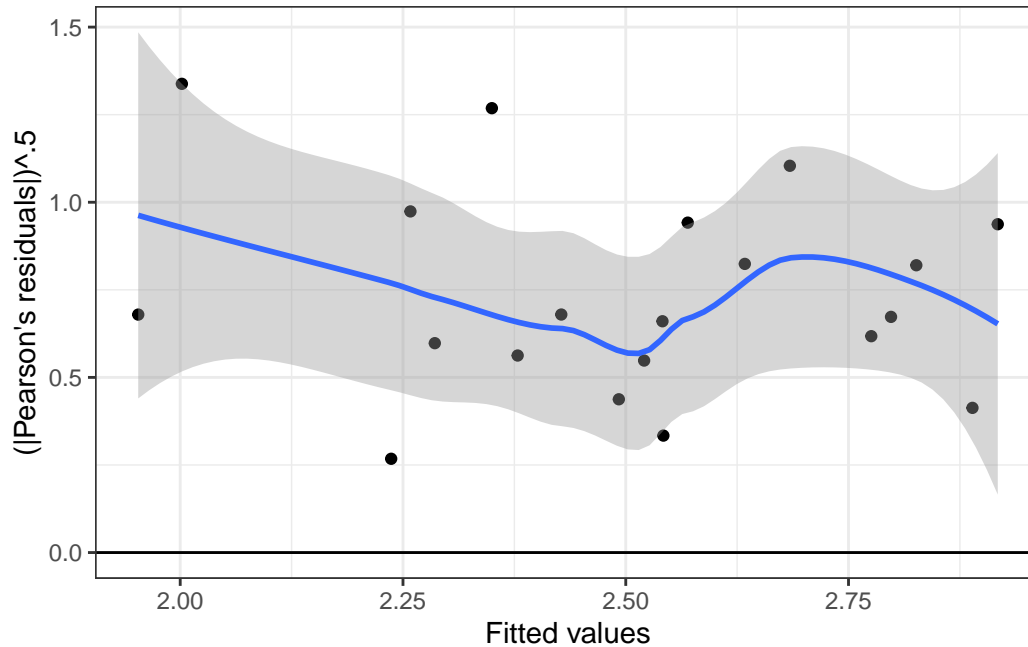
ai_dmso <- fortify.merMod(ai_dmso_m3)
ggplot(data = ai_dmso, aes(x = .fitted, y = .scresid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")

```



Residuals do not seem weird, given the scarcity of the data.

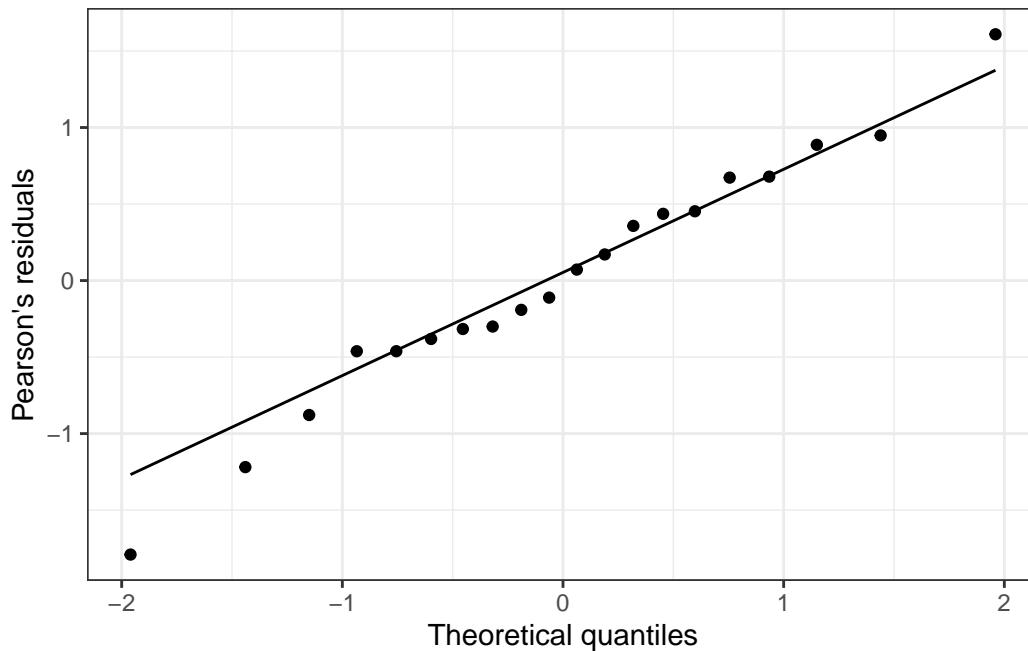
```
ggplot(data = ai_dmso, aes(x = .fitted, y = sqrt(abs(.screid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals seems 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 = ai_dms0, aes(sample = .screid)) +
  geom_qq() +
  geom_qq_line() +
  labs(x = "Theoretical quantiles", y = "Pearson's residuals")
```



It appears to be good. A Shapiro-Wilk test does not confirm Normality, but we are pretty sure it is caused by the extreme value:

```
shapiro.test(ai_dms0$.scesid)
```

Shapiro-Wilk normality test

```
data: ai_dms0$.scesid
W = 0.98424, p-value = 0.9765
```

## Predictions

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

```
ai_dms0_slope <- c(
  ci95_min = min(ai_dms0_m3_conf["conc", ]),
  estimate = fixef(ai_dms0_m3)[["conc"]],
  ci95_max = max(ai_dms0_m3_conf["conc", ]))
ai_dms0_slope
```

```

      ci95_min    estimate    ci95_max
-0.4339087 -0.2838577 -0.1238000

```

```

#saveRDS(ai_dmslo_slope, "../data/acrosome_integrity_DMSO_slope.rds")

```

Let's say we want to calculate the drop in acrosome integrity for various DMSO concentrations between 0 and 2% if the acrosome integrity of a sample without DMSO is 94%. 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))
}
dmslo_conc <- (0:20) / 10
ai_dmslo_lost <- predict_logit(dmslo_conc, 0.94, ai_dmslo_slope)
ai_dmslo_lost

```

	conc	ci95_min	estimate	ci95_max
1	0.0	1.110223e-16	1.110223e-16	1.110223e-16
2	0.1	-2.494478e-03	-1.621096e-03	-7.020474e-04
3	0.2	-5.085482e-03	-3.283042e-03	-1.411773e-03
4	0.3	-7.776143e-03	-4.986707e-03	-2.129247e-03
5	0.4	-1.056964e-02	-6.732969e-03	-2.854542e-03
6	0.5	-1.346920e-02	-8.522717e-03	-3.587731e-03
7	0.6	-1.647810e-02	-1.035685e-02	-4.328885e-03
8	0.7	-1.959964e-02	-1.223626e-02	-5.078077e-03
9	0.8	-2.283715e-02	-1.416188e-02	-5.835380e-03
10	0.9	-2.619402e-02	-1.613461e-02	-6.600868e-03
11	1.0	-2.967361e-02	-1.815539e-02	-7.374615e-03
12	1.1	-3.327934e-02	-2.022515e-02	-8.156695e-03
13	1.2	-3.701460e-02	-2.234482e-02	-8.947182e-03
14	1.3	-4.088279e-02	-2.451536e-02	-9.746151e-03
15	1.4	-4.488730e-02	-2.673770e-02	-1.055368e-02
16	1.5	-4.903149e-02	-2.901279e-02	-1.136983e-02
17	1.6	-5.331870e-02	-3.134159e-02	-1.219470e-02
18	1.7	-5.775220e-02	-3.372505e-02	-1.302835e-02
19	1.8	-6.233522e-02	-3.616412e-02	-1.387086e-02
20	1.9	-6.707093e-02	-3.865975e-02	-1.472231e-02

```
21 2.0 -7.196240e-02 -4.121290e-02 -1.558276e-02
```

```
#saveRDS(ai_dmsolost, "../data/acrosome_integrity_DMSO_lost.rds")
```

This is the lost in acrosome integrity that the model predicts. At 2% DMSO, we lose roughly 4%, and the 95%CI gives us a maximum lost of 7%.

## Ethanol

```
ai_etoh <- readxl::read_excel("../data/Table S2.xlsx",  
  sheet = "acrosome_integrity_EtOH")  
ai_etoh$donor <- as.factor(ai_etoh$donor)  
names(ai_etoh) <- c("donor", "conc", "acrointact", "total")  
ai_etoh$acrointact_frac <- ai_etoh$acrointact / ai_etoh$total  
skimr::skim(ai_etoh)
```

Table 4: Data summary

Name	ai_etoh
Number of rows	20
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	4	6: 5, 7: 5, 8: 5, 9: 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	
acrointact	0	1	187.15	7.34	174.00	183.75	187.00	189.00	213.00	



skim_variablen_missingcomplete_rate	mean	sd	p0	p25	p50	p75	p100	hist
total	0	1	202.05	6.30	200.00	200.00	200.00	200.25 228.00
acrointact_frac	0	1	0.93	0.02	0.87	0.91	0.93	0.94 0.96

There are four donors, no missing data.

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

```

  0 0.1 0.5 1 2
6 1    1    1 1 1
7 1    1    1 1 1
8 1    1    1 1 1
9 1    1    1 1 1

```

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

```
ai_etoh_m1 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (conc | donor),
  data = ai_etoh, family = binomial(link = "logit"))
```

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

```
summary(ai_etoh_m1)
```

```

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

```

```

      AIC      BIC    logLik deviance df.resid
  116.6    121.6    -53.3    106.6      15

```

Scaled residuals:

```

      Min      1Q   Median      3Q      Max
-1.04177 -0.54026  0.04142  0.43251  1.51864

```

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
donor	(Intercept)	0.00000	0.0000	
	conc	0.03567	0.1889	NaN

Number of obs: 20, groups: donor, 4

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	2.63558	0.08723	30.212	<2e-16 ***
conc	-0.11817	0.12533	-0.943	0.346

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

(Intr)  
 conc -0.469  
 optimizer (Nelder\_Mead) convergence code: 0 (OK)  
 boundary (singular) fit: see help('isSingular')

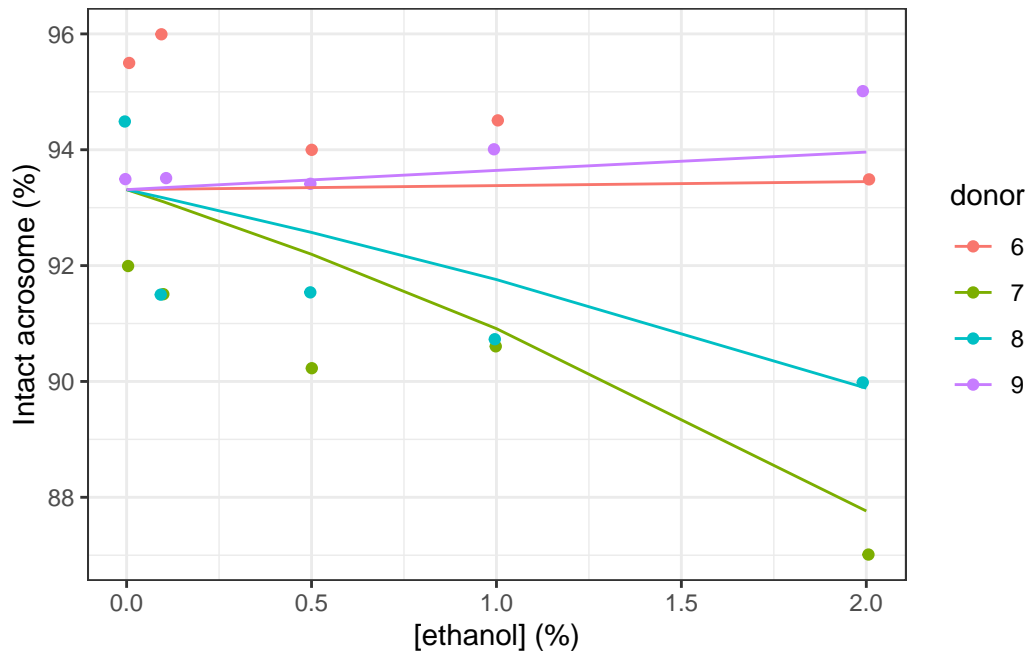
We have a singularity here. The model is:

,

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

Here is a plot of this model:

```
ggplot(data = ai_eto) +
  geom_jitter(aes(x = conc, y = acrointact_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(ai_eto_m1) * 100, col = donor)) +
  labs(x = "[ethanol] (%)", y = "Intact acrosome (%)")
```



Here the complete model was not able to estimate the variation of intercept per donor (so, it used the same one). However, data at concentration zero are more widespread. It is not clear if the model could be simplified for the intercept or the slope for the random effect `donor`. Let's look at both options...

```
ai_etoh_m2 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (1 | donor),
  data = ai_etoh, family = binomial(link = "logit"))
anova(ai_etoh_m1, ai_etoh_m2) # Despite the name, it is indeed a LR test
```

Data: ai\_etoh

Models:

```
ai_etoh_m2: cbind(acrointact, total - acrointact) ~ conc + (1 | donor)
ai_etoh_m1: cbind(acrointact, total - acrointact) ~ conc + (conc | donor)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
ai_etoh_m2	3	110.37	113.36	-52.185	104.37			
ai_etoh_m1	5	116.63	121.60	-53.313	106.63	0	2	1

Another model, with same intercept but different slopes:

```
ai_etoh_m3 <- glmer(cbind(acrointact, total - acrointact) ~ conc + (0 + conc:donor | donor),
  data = ai_etoh, family = binomial(link = "logit"))
```

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

```
anova(ai_etoh_m1, ai_etoh_m3) # Despite the name, it is indeed a LR test
```

Data: ai\_etoh

Models:

```
ai_etoh_m1: cbind(acrointact, total - acrointact) ~ conc + (conc | donor)
```

```
ai_etoh_m3: cbind(acrointact, total - acrointact) ~ conc + (0 + conc:donor | donor)
```

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
ai_etoh_m1	5	116.63	121.60	-53.313	106.63			
ai_etoh_m3	12	128.98	140.93	-52.492	104.98	1.6426	7	0.977

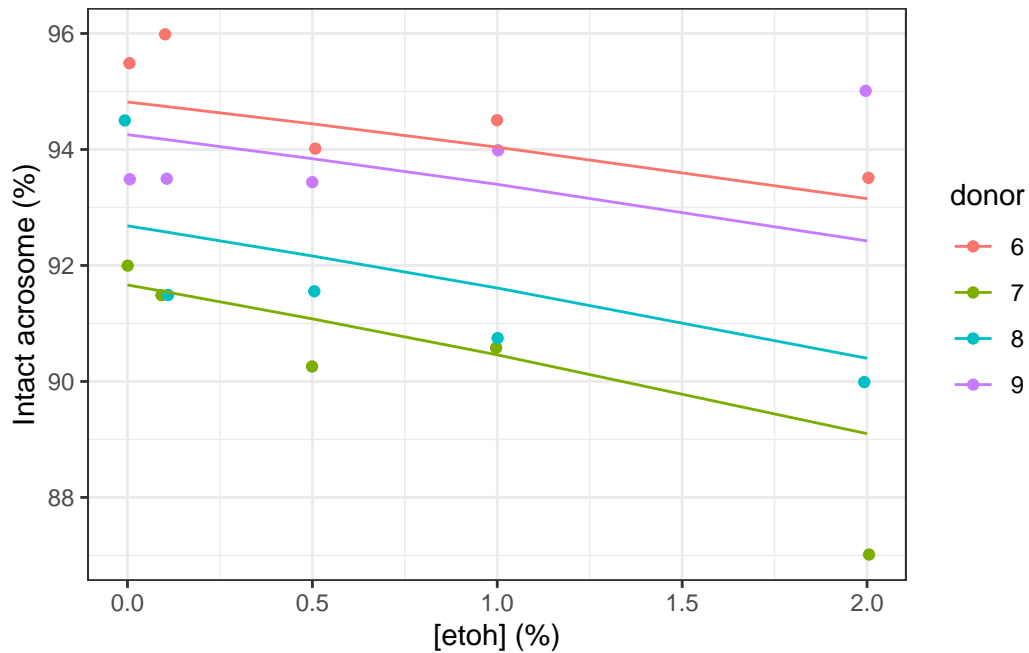
The likelihood ratio test does not detect significant differences between the full and both simplified models at  $\alpha = 5\%$ , but the second model has singularities too. Using different slopes produces singular gradient. Here is the model with only intercept depending on the donor, which is fitted without problems:

,

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

Here is a plot of this model:

```
ggplot(data = ai_etoh) +  
  geom_jitter(aes(x = conc, y = acrointact_frac * 100, col = donor),  
    width = 0.01) +  
  geom_line(aes(x = conc, y = fitted(ai_etoh_m2) * 100, col = donor)) +  
  labs(x = "[etoh] (%)", y = "Intact acrosome (%)")
```



```
summary(ai_etoh_m2)
```

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

AIC	BIC	logLik	deviance	df.resid
110.4	113.4	-52.2	104.4	17

Scaled residuals:

Min	1Q	Median	3Q	Max
-0.9529	-0.4104	-0.1084	0.2921	1.3771

Random effects:

Groups Name	Variance	Std.Dev.
donor (Intercept)	0.05266	0.2295

Number of obs: 20, groups: donor, 4

Fixed effects:

Estimate	Std. Error	z value	Pr(> z )
----------	------------	---------	----------

```
(Intercept)  2.66406    0.14470  18.411   <2e-16 ***
conc         -0.14824    0.08022  -1.848    0.0646 .
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Correlation of Fixed Effects:
```

```
  (Intr)
conc -0.436
```

The Z test indicates that `conc` is not 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(ai_etoh_m2, level = 0.95) # 95% CI based on profile
```

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

```
          2.5 %      97.5 %
.sig01      0.08352725 0.62762234
(Intercept)  2.32574073 3.01787271
conc        -0.30431936 0.01089323
```

```
set.seed(7431)
# 1000x parameter bootstrap
(ai_etoh_m2_conf <- confint(ai_etoh_m2, level = 0.95, method = "boot", nsim = 1000L))
```

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

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

```
7 warning(s): Model failed to converge with max|grad| = 0.00254552 (tol = 0.002, component 1)
```

```
          2.5 %      97.5 %
.sig01      0.0000000 0.37669024
(Intercept)  2.4006581 2.95902373
conc        -0.3153885 0.02246623
```

We had 160 bootstrapped model with singularity among the 1000. Lower bound for the bootstrapped 95%CI is rather different to the one from profiles. This is not surprising since we have rather few data here. Slope for `conc` is not significantly different from zero at  $\alpha = 5\%$  because the 95% CI contains zero. However, it could be due to the scarcity of the data. Yet, the effect appears weak with a loss of a few percents for a concentration of 2% ethanol. We conclude here that the effect is either weak, or inexistent. Using upper bound 95%CI, we would have a variation of:

```
# Let's consider a value of 0.94 at conc = 0, with a slope of -0.31
# we lose:
-0.94 + boot::inv.logit(boot::logit(0.94) - 0.32 * 2)
```

```
[1] -0.0479807
```

That is, we have less than 5% variation in acrosome integrity at worst at ethanol concentration of 2%.

### Additional verifications

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(ai_etoh_m2)
```

```
[1] FALSE
```

... then, a report about the model convergence:

```
ai_etoh_m2_all <- allFit(ai_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(ai_etoH_m2_all)
```

```
$which.OK
```

	bobyqa	Nelder_Mead
	TRUE	TRUE
	nlminbwrap	nmk bw
	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
```

```
$msgs$nmk bw
```

```
NULL
```

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

```
NULL
```

```
$msgs$nloptwrap.NLOPT_LN_NELDERMEAD
```

```
NULL
```

```
$msgs$nloptwrap.NLOPT_LN_BOBYQA
```

```
NULL
```

```
$fixef
```

	(Intercept)	conc
bobyqa	2.664067	-0.1482448
Nelder_Mead	2.664057	-0.1482393
nlminbwrap	2.664066	-0.1482438
nmk bw	2.663973	-0.1481829



optimx.L-BFGS-B	2.664067	-0.1482447
nloptwrap.NLOPT_LN_NELDERMEAD	2.664063	-0.1482756
nloptwrap.NLOPT_LN_BOBYQA	2.664068	-0.1482449

\$llik

bobyqa	Nelder_Mead
-52.18489	-52.18489
nlminbwrap	nmkbw
-52.18489	-52.18489
optimx.L-BFGS-B	nloptwrap.NLOPT_LN_NELDERMEAD
-52.18489	-52.18489
nloptwrap.NLOPT_LN_BOBYQA	
-52.18489	

\$sdcor

	donor.(Intercept)
bobyqa	0.2294803
Nelder_Mead	0.2294782
nlminbwrap	0.2294793
nmkbw	0.2294284
optimx.L-BFGS-B	0.2294825
nloptwrap.NLOPT_LN_NELDERMEAD	0.2294788
nloptwrap.NLOPT_LN_BOBYQA	0.2294829

\$theta

	donor.(Intercept)
bobyqa	0.2294803
Nelder_Mead	0.2294782
nlminbwrap	0.2294793
nmkbw	0.2294284
optimx.L-BFGS-B	0.2294825
nloptwrap.NLOPT_LN_NELDERMEAD	0.2294788
nloptwrap.NLOPT_LN_BOBYQA	0.2294829

\$times

	user.self	sys.self	elapsed	user.child	sys.child
bobyqa	0.042	0	0.043	0	0
Nelder_Mead	0.053	0	0.054	0	0
nlminbwrap	0.045	0	0.045	0	0
nmkbw	0.053	0	0.053	0	0
optimx.L-BFGS-B	0.323	0	0.323	0	0
nloptwrap.NLOPT_LN_NELDERMEAD	0.049	0	0.049	0	0
nloptwrap.NLOPT_LN_BOBYQA	0.038	0	0.039	0	0

```

$feval
      bobyqa      Nelder_Mead
      59      83
nlminbwrap      nmkbw
      NA      101
optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
      16      87
nloptwrap.NLOPT_LN_BOBYQA
      32

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

```

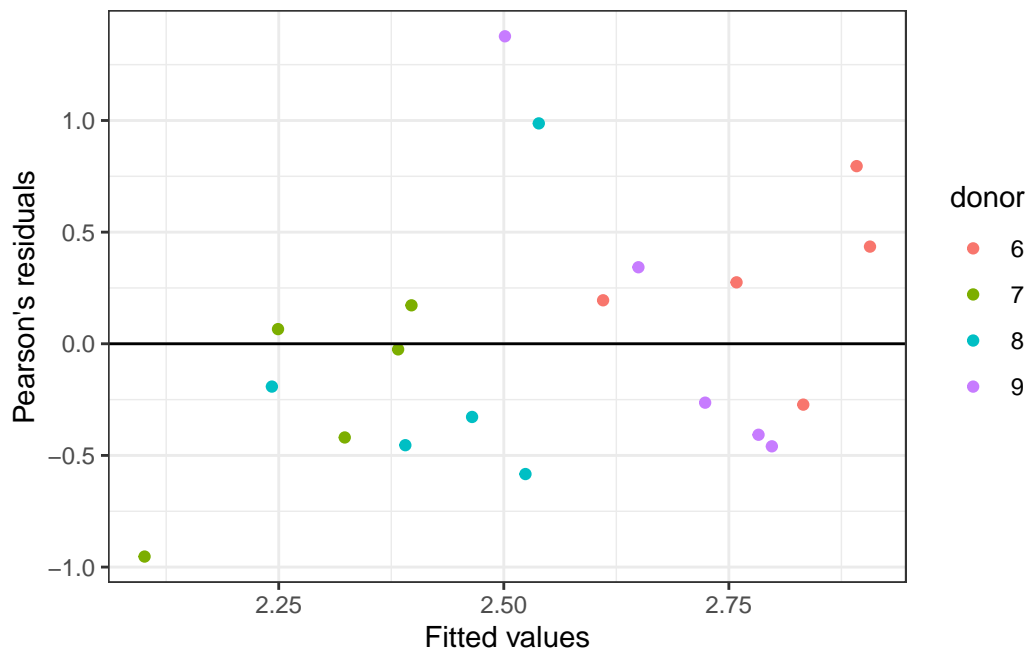
## Analysis of the residuals

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

```

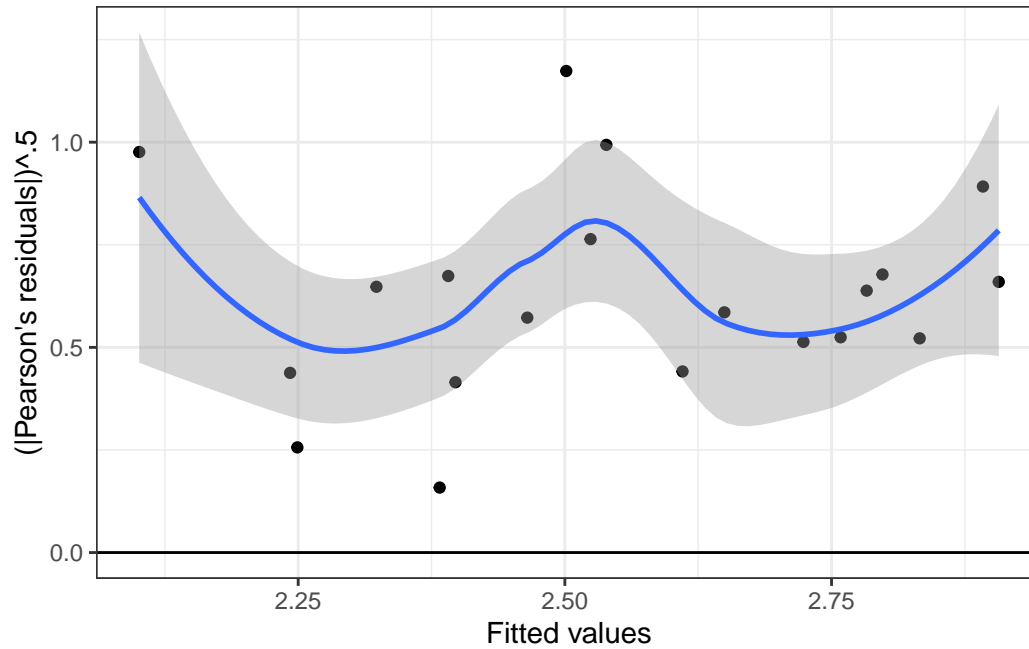
ai_etoh <- fortify.merMod(ai_etoh_m2)
ggplot(data = ai_etoh, aes(x = .fitted, y = .screid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")

```



Given the scarcity of the data, residuals do not seem abnormal.

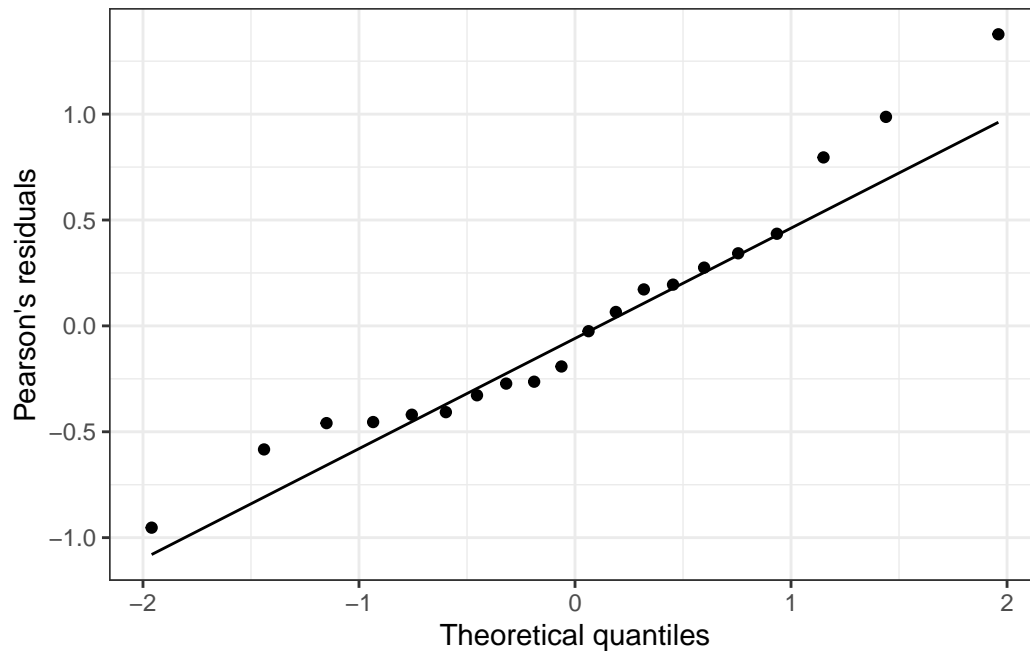
```
ggplot(data = ai_etoh, aes(x = .fitted, y = sqrt(abs(.screid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



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 = ai_eto, aes(sample = .scredid)) +
  geom_qq() +
  geom_qq_line() +
  labs(x = "Theoretical quantiles", y = "Pearson's residuals")
```



It appears not particularly bad. A Shapiro-Wilk test confirms Normality (with caution because this test tends to be conservative):

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

Shapiro-Wilk normality test

```
data:  ai_etoH$.scresid  
W = 0.94869, p-value = 0.3476
```

## 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] tidy_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		