Vitality

Marie Bisconti, Philippe Grosjean & Elise Hennebert

Introduction

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

DMSO

```r
vit_dmso <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "vitality_DMSO")
vit_dmso$donor <- as.factor(vit_dmso$donor)
names(vit_dmso) <- c("donor", "conc", "live", "total")
vit_dmso$live_frac <- vit_dmso$live / vit_dmso$total
skimr::skim(vit_dmso)
```

Table 1: Data summary

Name	vit_dmso
Number of rows	25
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	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	
live	0	1	280.96	7.29	265.00	277.00	282.00	286.00	295.00	
total	0	1	300.00	0.00	300.00	300.00	300.00	300.00	300.00	
live_frac	0	1	0.94	0.02	0.88	0.92	0.94	0.95	0.98	

There are five donors, no missing data.

```r
table(vit_dmso$donor, as.factor(vit_dmso$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.

```r
vit_dmso_m1 <- glmer(cbind(live, total - live) ~ conc + (conc | donor),
  data = vit_dmso, family = binomial(link = "logit"))
```

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

```r
summary(vit_dmso_m1)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
```

```
Formula: cbind(live, total - live) ~ conc + (conc | donor)
   Data: vit_dmso

     AIC      BIC   logLik deviance df.resid
   159.2    165.3    -74.6    149.2       20

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.57723 -0.59356 -0.01316  0.60050  2.13289

Random effects:
 Groups Name        Variance Std.Dev. Corr
 donor  (Intercept) 0.098243 0.31344
        conc        0.001485 0.03854  -1.00
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.92013    0.15795  18.487  < 2e-16 ***
conc        -0.24812    0.06587  -3.767 0.000165 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.561
optimizer (Nelder_Mead) convergence code: 0 (OK)
boundary (singular) fit: see help('isSingular')
```

We observe a singularity because correlation between slope and intercept for the random term `donor` is close to the boundary. Let's see if a simplified model where only the intercept depends on the donor fits better…

```r
vit_dmso_m2 <- glmer(cbind(live, total - live) ~ conc  + (1 | donor),
  data = vit_dmso, family = binomial(link = "logit"))
summary(vit_dmso_m2)
```

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

```
   Data: vit_dmso

     AIC      BIC   logLik deviance df.resid
   155.5    159.2    -74.8    149.5       22

Scaled residuals:
     Min       1Q   Median       3Q      Max
-1.59763 -0.62843  0.07171  0.45241  2.18968

Random effects:
 Groups Name        Variance Std.Dev.
 donor  (Intercept) 0.07905  0.2812
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.91196    0.14451  20.150  < 2e-16 ***
conc        -0.23906    0.06188  -3.863 0.000112 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.358
```

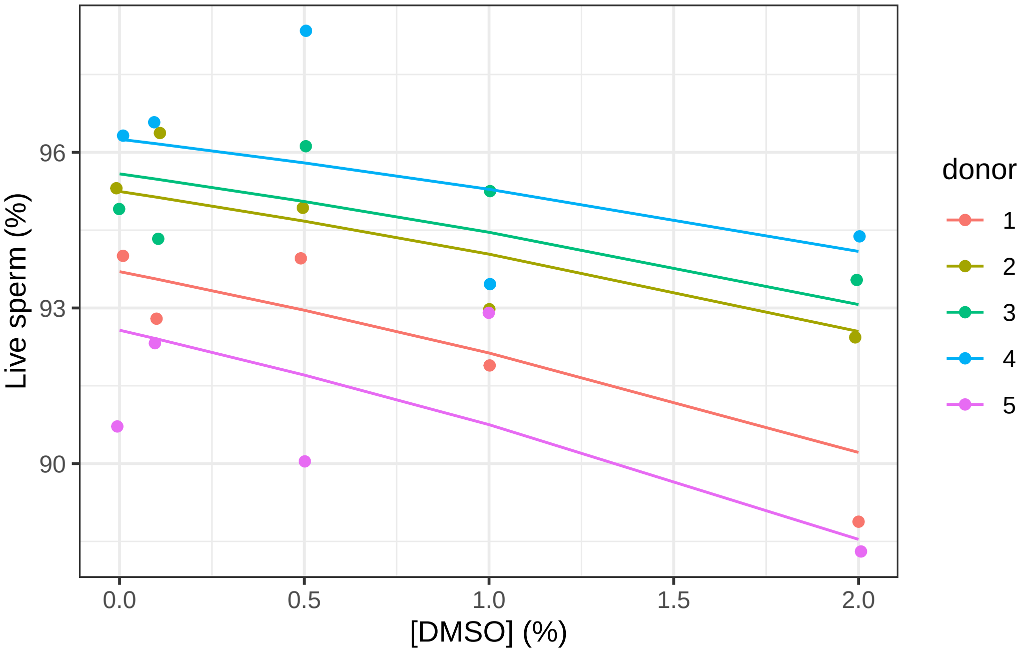
No singularities any more. We stick with this second model. This model is:

,

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

Here is a plot of this model:

```r
ggplot(data = vit_dmso) +
  geom_jitter(aes(x = conc, y = live_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(vit_dmso_m2) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Live sperm (%)")
```



The Z test indicates that `conc` is significantly different from zero at $\alpha = 5\%$ (see summary of the model above). 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(vit_dmso_m2, level = 0.95) # 95% CI based on profile
```

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

```
                 2.5 %     97.5 %
.sig01       0.1446416  0.6475042
(Intercept)  2.5820574  3.2526924
conc        -0.3598545 -0.1167625
```

```
set.seed(7400)
# 1000x parameter bootstrap
(vit_dmso_m2_conf <- confint(vit_dmso_m2, level = 0.95,
  method = "boot", nsim  = 1000L))
```

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

```
45 message(s): boundary (singular) fit: see help('isSingular')
5 warning(s): Model failed to converge with max|grad| = 0.00265077 (tol = 0.002, component 1
```

```
                    2.5 %     97.5 %
.sig01       4.975943e-07  0.4312967
(Intercept)  2.645017e+00  3.2162631
conc        -3.691651e-01 -0.1103052
```

We have 1/20 of the bootstrapped models that exhibit singularities. However, 95%CI from parametric bootstraps and from profiles are not very different. 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(vit_dmso_m2, scope = "conc")
vit_dmso_m3 <- glmer(cbind(live, total - live) ~ 1 + (1 | donor),
  data = vit_dmso, family = binomial(link = "logit"))
anova(vit_dmso_m2, vit_dmso_m3, refit = TRUE)
```

```
Data: vit_dmso
Models:
vit_dmso_m3: cbind(live, total - live) ~ 1 + (1 | donor)
vit_dmso_m2: cbind(live, total - live) ~ conc + (1 | donor)
            npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
vit_dmso_m3    2 167.99 170.43 -81.996   163.99
vit_dmso_m2    3 155.52 159.18 -74.761   149.52 14.468  1  0.0001425 ***
---
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 $\texttt{conc} = 0$. 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?

```r
isSingular(vit_dmso_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```r
vit_dmso_m2_all <- allFit(vit_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(vit_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
```

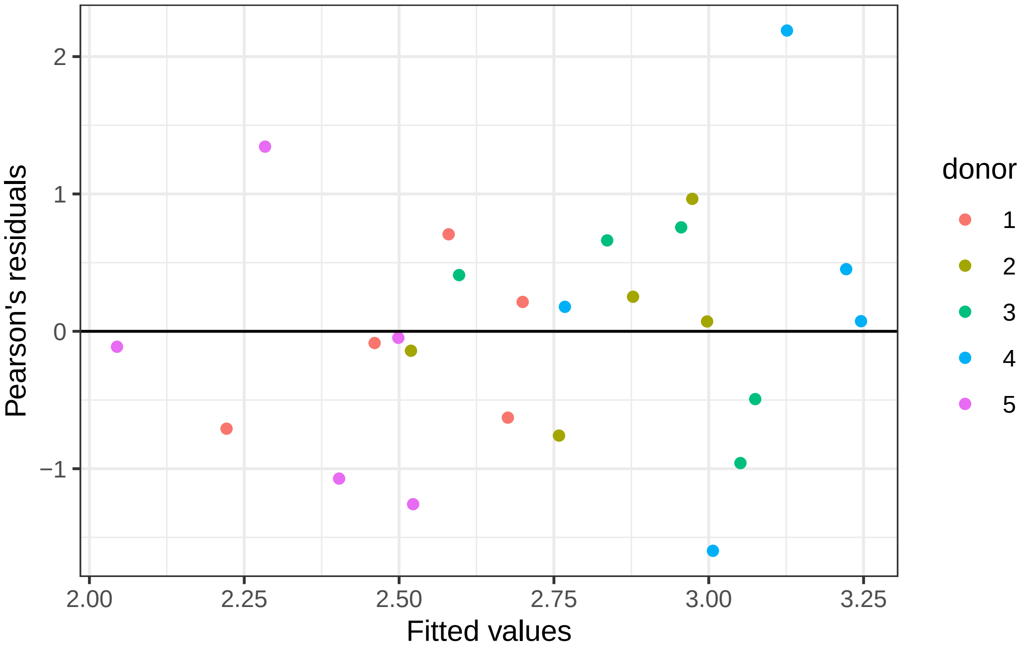
```
NULL

$msgs$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                           2.911968 -0.2390588
Nelder_Mead                      2.911962 -0.2390595
nlminbwrap                       2.911969 -0.2390596
nmkbw                            2.911916 -0.2390368
optimx.L-BFGS-B                  2.911968 -0.2390588
nloptwrap.NLOPT_LN_NELDERMEAD    2.912075 -0.2390898
nloptwrap.NLOPT_LN_BOBYQA        2.911967 -0.2390585

$llik
                       bobyqa                   Nelder_Mead
                    -74.76146                     -74.76146
                   nlminbwrap                         nmkbw
                    -74.76146                     -74.76146
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                    -74.76146                     -74.76146
    nloptwrap.NLOPT_LN_BOBYQA
                    -74.76146

$sdcor
                              donor.(Intercept)
bobyqa                                0.2811539
Nelder_Mead                           0.2811531
nlminbwrap                            0.2811546
```

```
nmkbw                              0.2811194
optimx.L-BFGS-B                    0.2811505
nloptwrap.NLOPT_LN_NELDERMEAD      0.2811708
nloptwrap.NLOPT_LN_BOBYQA          0.2811587

$theta
                              donor.(Intercept)
bobyqa                             0.2811539
Nelder_Mead                        0.2811531
nlminbwrap                         0.2811546
nmkbw                              0.2811194
optimx.L-BFGS-B                    0.2811505
nloptwrap.NLOPT_LN_NELDERMEAD      0.2811708
nloptwrap.NLOPT_LN_BOBYQA          0.2811587

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.044    0.001   0.045          0         0
Nelder_Mead                       0.060    0.000   0.060          0         0
nlminbwrap                        0.049    0.000   0.050          0         0
nmkbw                             0.060    0.001   0.063          0         0
optimx.L-BFGS-B                   0.339    0.000   0.339          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.053    0.000   0.054          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.040    0.000   0.040          0         0

$feval
                       bobyqa                   Nelder_Mead
                           60                            88
                   nlminbwrap                         nmkbw
                           NA                           106
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                           12                            88
    nloptwrap.NLOPT_LN_BOBYQA
                           40

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

Analysis of the residuals

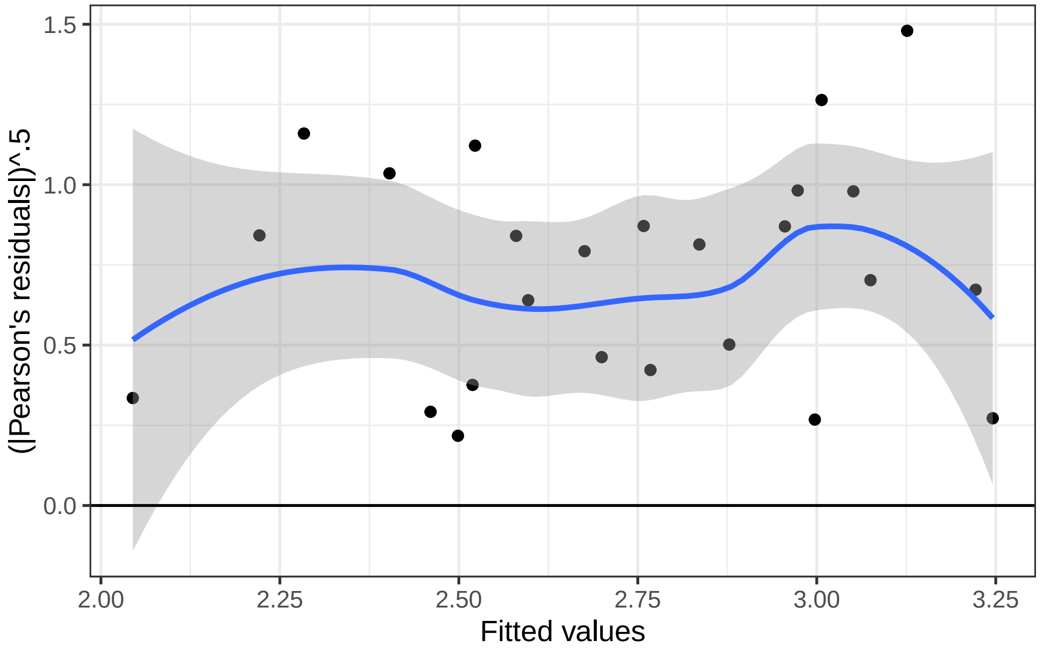
Let’s check how the residuals distribute and if there is homoscedasticity.

```
vit_dmso <- fortify.merMod(vit_dmso_m2)
ggplot(data = vit_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 or two extreme values, but otherwise, residuals seem rather correctly distributed. Linearity is good here.

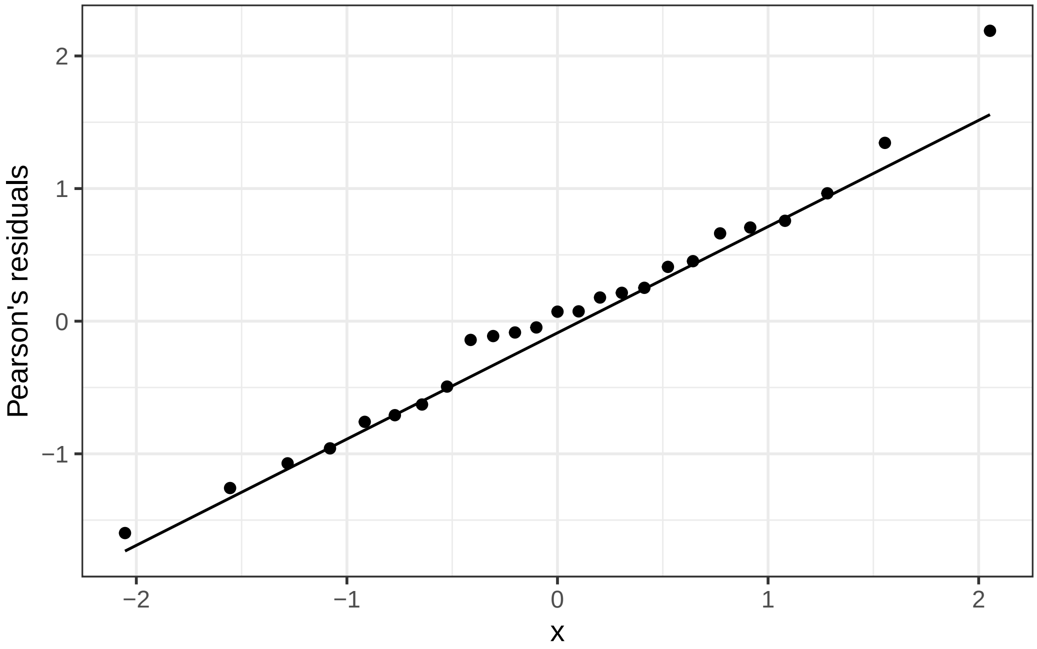
```
ggplot(data = vit_dmso, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



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:

```r
ggplot(data = vit_dmso, aes(sample = .scresid)) +
  geom_qq() +
  geom_qq_line() +
  labs(c = "Theaeretical quantile", y = "Pearson's residuals")
```



It appears not too bad, except for one extreme value that is clearly visible here at the top. A Shapiro-Wilk test confirms Normality:

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

```

	Shapiro-Wilk normality test

data:  vit_dmso$.scresid
W = 0.98161, p-value = 0.915
```

Predictions

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

```
vit_dmso_slope <- c(
  ci95_min   = min(vit_dmso_m2_conf["conc", ]),
  estimate = fixef(vit_dmso_m2)[["conc"]],
  ci95_max  = max(vit_dmso_m2_conf["conc", ]))
vit_dmso_slope
```

```
  ci95_min   estimate   ci95_max 
-0.3691651 -0.2390572 -0.1103052 
```

```
#saveRDS(vit_dmso_slope, "../data/vitality_DMSO_slope.rds")
```

Let's say we want to calculate the drop in vitality for various DMSO concentrations between 0 and 2% if the vitality 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))
}
dmso_conc <- (0:20) / 10
vit_dmso_lost <- predict_logit(dmso_conc, 0.94, vit_dmso_slope)
vit_dmso_lost
```

```
   conc      ci95_min      estimate      ci95_max
1   0.0  1.110223e-16  1.110223e-16  1.110223e-16
2   0.1 -2.116225e-03 -1.362550e-03 -6.251494e-04
3   0.2 -4.301985e-03 -2.753976e-03 -1.256388e-03
4   0.3 -6.559199e-03 -4.174797e-03 -1.893766e-03
5   0.4 -8.889815e-03 -5.625534e-03 -2.537334e-03
6   0.5 -1.129581e-02 -7.106715e-03 -3.187143e-03
7   0.6 -1.377917e-02 -8.618871e-03 -3.843244e-03
8   0.7 -1.634192e-02 -1.016254e-02 -4.505689e-03
9   0.8 -1.898610e-02 -1.173826e-02 -5.174529e-03
10  0.9 -2.171376e-02 -1.334658e-02 -5.849816e-03
11  1.0 -2.452697e-02 -1.498804e-02 -6.531601e-03
12  1.1 -2.742781e-02 -1.666319e-02 -7.219938e-03
13  1.2 -3.041838e-02 -1.837260e-02 -7.914878e-03
14  1.3 -3.350075e-02 -2.011682e-02 -8.616474e-03
15  1.4 -3.667704e-02 -2.189640e-02 -9.324779e-03
16  1.5 -3.994933e-02 -2.371192e-02 -1.003985e-02
17  1.6 -4.331971e-02 -2.556393e-02 -1.076173e-02
18  1.7 -4.679026e-02 -2.745300e-02 -1.149048e-02
19  1.8 -5.036304e-02 -2.937971e-02 -1.222615e-02
20  1.9 -5.404009e-02 -3.134461e-02 -1.296879e-02
21  2.0 -5.782342e-02 -3.334829e-02 -1.371847e-02
```

```
#saveRDS(vit_dmso_lost, "../data/vitality_DMSO_lost.rds")
```

This is the lost in vitality that the model predicts. Despite being significant at $\alpha = 5\%$, its effect is rather small because for 2% DMSO, we loose just a little bit more than 3%, with an upper bound of the 95%CI (most pessimistic value) of less than 6%.

Ethanol

```
vit_etoh <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "vitality_EtOH")
vit_etoh$donor <- as.factor(vit_etoh$donor)
names(vit_etoh) <- c("donor", "conc", "live", "total")
vit_etoh$live_frac <- vit_etoh$live / vit_etoh$total
skimr::skim(vit_etoh)
```

Table 4: Data summary

Name	vit_etoh
Number of rows	25
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	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	
live	0	1	281.28	7.61	260.00	277.00	282.00	286.00	295.00	
total	0	1	300.00	0.00	300.00	300.00	300.00	300.00	300.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
live_frac	0	1	0.94	0.03	0.87	0.92	0.94	0.95	0.98	

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

```r
table(vit_etoh$donor, as.factor(vit_etoh$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.

```r
vit_etoh_m1 <- glmer(cbind(live, total - live) ~ conc + (conc | donor),
  data = vit_etoh, family = binomial(link = "logit"))
summary(vit_etoh_m1)
```

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

     AIC      BIC   logLik deviance df.resid 
   181.2    187.3    -85.6    171.2       20 

Scaled residuals: 
    Min      1Q  Median      3Q     Max 
-4.1253 -0.4670  0.1203  0.7488  2.2050 

Random effects:
 Groups Name        Variance Std.Dev. Corr 
 donor  (Intercept) 0.100880 0.31762       
        conc        0.003301 0.05745  -0.49
```

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

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.77477    0.15846   17.51   <2e-16 ***
conc        -0.03681    0.07220   -0.51     0.61
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.448
```

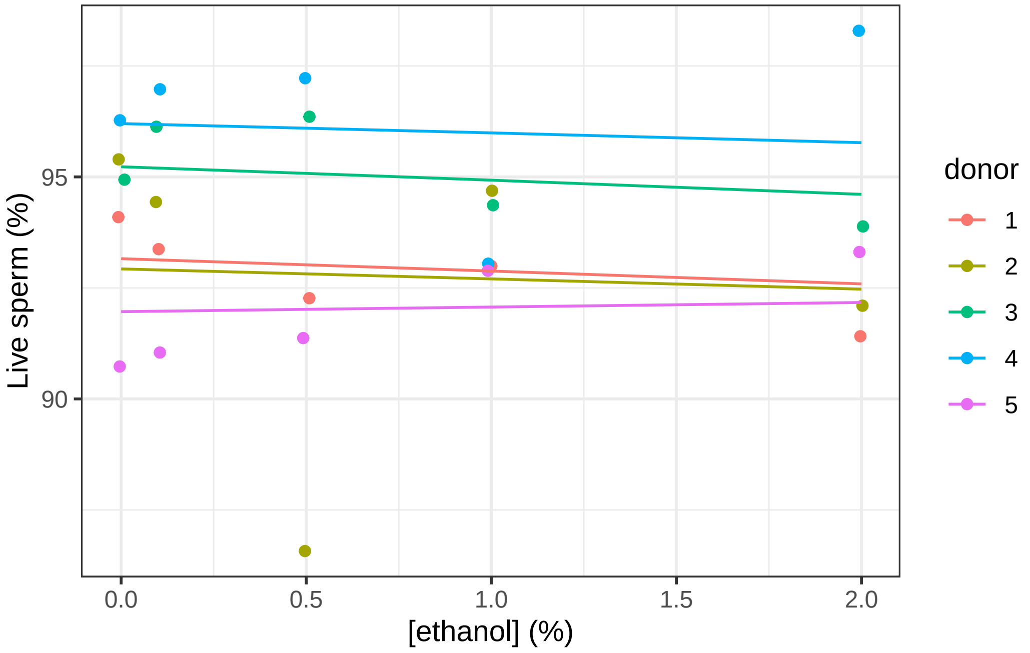
The model is:

,

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

Here is a plot of this model:

```
ggplot(data = vit_etoh) +
  geom_jitter(aes(x = conc, y = live_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(vit_etoh_m1) * 100, col = donor)) +
  labs(x = "[ethanol] (%)", y = "Live sperm (%)")
```



Generally, slopes seem rather close to zero, suggesting no or negligible effect of the concentration in ethanol up to 2%. Data are rather widespread. Let's check if the model can be simplified is a similar way as for DMSO using a likelihood ratio test:

```
vit_etoh_m2 <- glmer(cbind(live, total - live) ~ conc  + (1 | donor),
  data = vit_etoh, family = binomial(link = "logit"))
anova(vit_etoh_m1, vit_etoh_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

```
Data: vit_etoh
Models:
vit_etoh_m2: cbind(live, total - live) ~ conc + (1 | donor)
vit_etoh_m1: cbind(live, total - live) ~ conc + (conc | donor)
            npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
vit_etoh_m2    3 177.31 180.96 -85.654   171.31                    
vit_etoh_m1    5 181.16 187.25 -85.579   171.16  0.15  2     0.9277
```

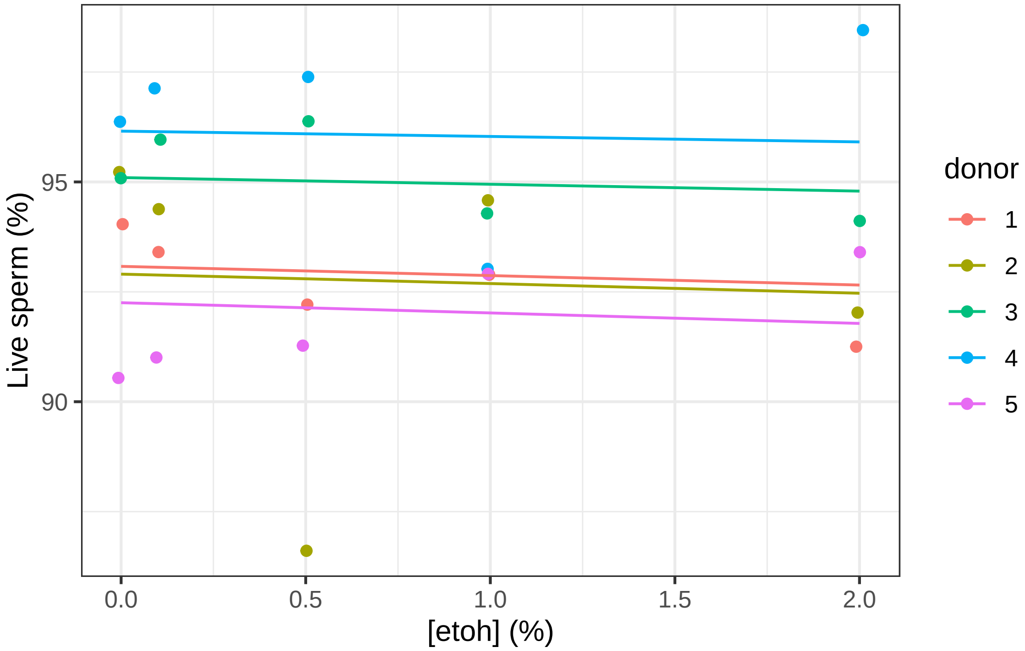
The likelihood ratio test does not detects significant differences between the full and simplified models at $\alpha = 5\%$. We could thus use the simplest `vit_etoh_m2` model with only a shift in the slope per donor. This model is:

,

$$\begin{array}{c} \text{live}_i \sim \text{Binomial}(n = 1, \text{prob}_{\text{live=1}} = \widehat{P}) \\ , \log\left[\frac{\hat{P}}{1-\hat{P}}\right] = \alpha_{j[i]} + \beta_1(\text{conc}) \qquad\qquad\qquad , \\ \alpha_j \sim N\left(\mu_{\alpha_j}, \sigma^2_{\alpha_j}\right), \text{ for donor j} = 1, \ldots, \text{J} \end{array} \tag{3}$$

Here is a plot of this model:

```
ggplot(data = vit_etoh) +
  geom_jitter(aes(x = conc, y = live_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(vit_etoh_m2) * 100, col = donor)) +
  labs(x = "[etoh] (%)", y = "Live sperm (%)")
```



```
summary(vit_etoh_m2)
```

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

```
   Data: vit_etoh

     AIC      BIC   logLik deviance df.resid
   177.3    181.0    -85.7    171.3       22

Scaled residuals:
    Min      1Q  Median      3Q     Max
-4.1062 -0.5172  0.1606  0.9779  2.1189

Random effects:
 Groups Name        Variance Std.Dev.
 donor  (Intercept) 0.09001  0.3
Number of obs: 25, groups:  donor, 5

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  2.77062    0.15079  18.374   <2e-16 ***
conc        -0.03210    0.06489  -0.495    0.621
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
     (Intr)
conc -0.316
```

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(vit_etoh_m2, level = 0.95) # 95% CI based on profile
```

```
Computing profile confidence intervals ...


                 2.5 %     97.5 %
.sig01       0.1522654 0.69476915
(Intercept)  2.4218361 3.13168088
conc        -0.1583188 0.09662959
```

```
set.seed(74588)
# 1000x parameter bootstrap
(vit_etoh_m2_conf <- confint(vit_etoh_m2, level = 0.95, method = "boot", nsim  = 1000L))
```

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

```
30 message(s): boundary (singular) fit: see help('isSingular')
1 warning(s): Model failed to converge with max|grad| = 0.00317722 (tol = 0.002, component 1
```

```
                    2.5 %    97.5 %
.sig01       1.566769e-05 0.4918353
(Intercept)  2.486681e+00 3.0805498
conc        -1.555630e-01 0.1018761
```

30 fits on bootstrapped data had singularities and one failed to converge. However, 95%CI from profiles and from parametric bootstraps are close. So, we can trust them. They indicate that the slope for `conc` is not significantly different from zero at $\alpha = 5\%$. He, we detect no significant effect of ethanol up to 2% on the spermatozoa vitality.

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(vit_etoh_m2, scope = "conc")
vit_etoh_m3 <- glmer(cbind(live, total - live) ~ 1 + (1 | donor),
  data = vit_etoh, family = binomial(link = "logit"))
anova(vit_etoh_m2, vit_etoh_m3, refit = TRUE)
```

```
Data: vit_etoh
Models:
vit_etoh_m3: cbind(live, total - live) ~ 1 + (1 | donor)
vit_etoh_m2: cbind(live, total - live) ~ conc + (1 | donor)
            npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
vit_etoh_m3    2 175.55 177.99 -85.775   171.55                     
vit_etoh_m2    3 177.31 180.96 -85.654   171.31 0.2426  1     0.6223
```

The model with `conc` is not significantly different at α level 5% from a reference model that sets the slope `conc` $= 0$. This is in accordance with the results we observed using 95%CIs.

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(vit_etoh_m2)
```

```
[1] FALSE
```

… then, a report about the model convergence:

```
vit_etoh_m2_all <- allFit(vit_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(vit_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
```

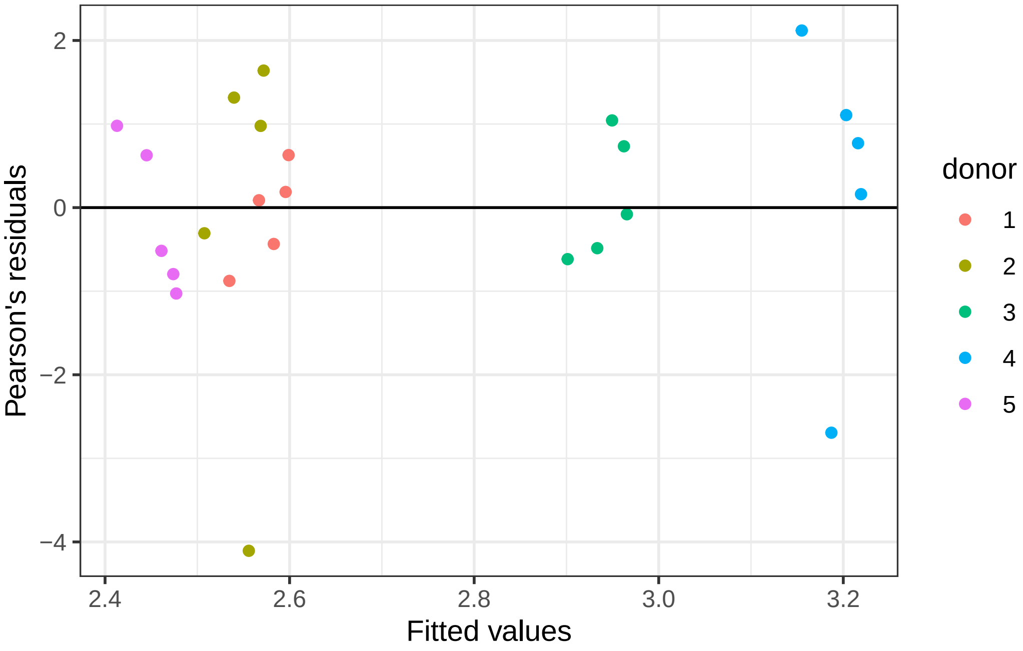
```
NULL

$msgs$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                           2.770624 -0.03209916
Nelder_Mead                      2.770621 -0.03209735
nlminbwrap                       2.770624 -0.03209918
nmkbw                            2.770518 -0.03209494
optimx.L-BFGS-B                  2.770625 -0.03209909
nloptwrap.NLOPT_LN_NELDERMEAD    2.770698 -0.03210482
nloptwrap.NLOPT_LN_BOBYQA        2.770623 -0.03209608

$llik
                       bobyqa                   Nelder_Mead
                    -85.65359                     -85.65359
                   nlminbwrap                         nmkbw
                    -85.65359                     -85.65359
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                    -85.65359                     -85.65359
    nloptwrap.NLOPT_LN_BOBYQA
                    -85.65359

$sdcor
                              donor.(Intercept)
bobyqa                                0.3000132
Nelder_Mead                           0.3000161
nlminbwrap                            0.3000133
```

```
nmkbw                                0.3000197
optimx.L-BFGS-B                      0.3000141
nloptwrap.NLOPT_LN_NELDERMEAD        0.3000486
nloptwrap.NLOPT_LN_BOBYQA            0.3000174

$theta
                              donor.(Intercept)
bobyqa                               0.3000132
Nelder_Mead                          0.3000161
nlminbwrap                           0.3000133
nmkbw                                0.3000197
optimx.L-BFGS-B                      0.3000141
nloptwrap.NLOPT_LN_NELDERMEAD        0.3000486
nloptwrap.NLOPT_LN_BOBYQA            0.3000174

$times
                              user.self sys.self elapsed user.child sys.child
bobyqa                            0.043    0.000   0.042          0         0
Nelder_Mead                       0.055    0.000   0.055          0         0
nlminbwrap                        0.046    0.000   0.046          0         0
nmkbw                             0.060    0.000   0.060          0         0
optimx.L-BFGS-B                   0.349    0.002   0.350          0         0
nloptwrap.NLOPT_LN_NELDERMEAD     0.051    0.001   0.051          0         0
nloptwrap.NLOPT_LN_BOBYQA         0.040    0.000   0.041          0         0

$feval
                       bobyqa                   Nelder_Mead
                           59                            89
                   nlminbwrap                         nmkbw
                           NA                           103
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
                           12                            84
    nloptwrap.NLOPT_LN_BOBYQA
                           29

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

Analysis of the residuals

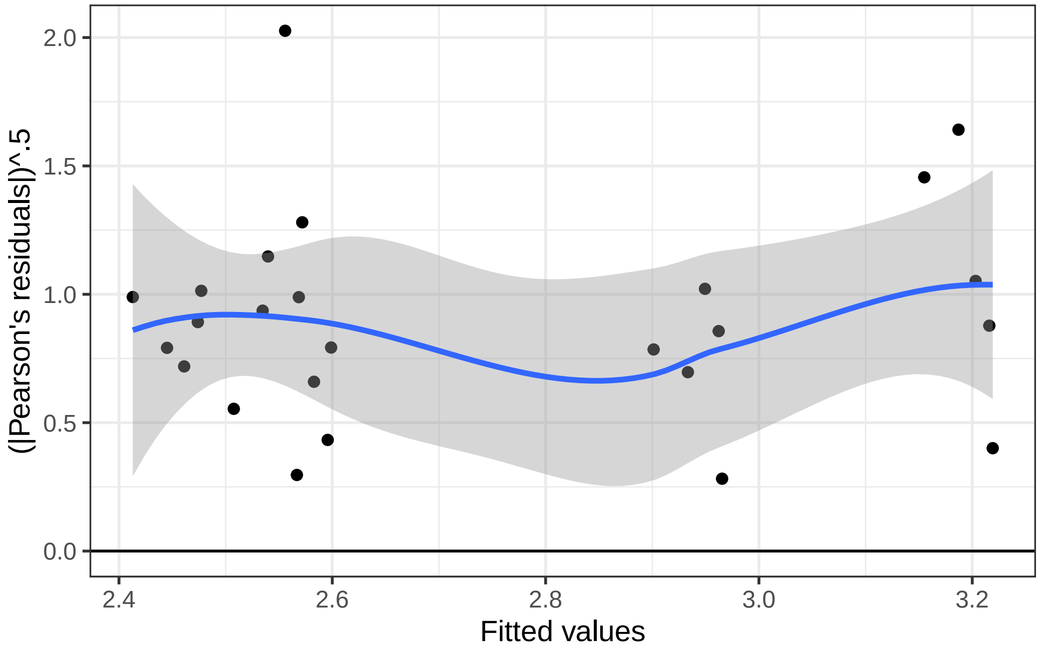
Let’s check how the residuals distribute and if there is homoscedasticity.

```
vit_etoh <- fortify.merMod(vit_etoh_m2)
ggplot(data = vit_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, but otherwise, residuals seem rather correctly distributed. Linearity is good here.

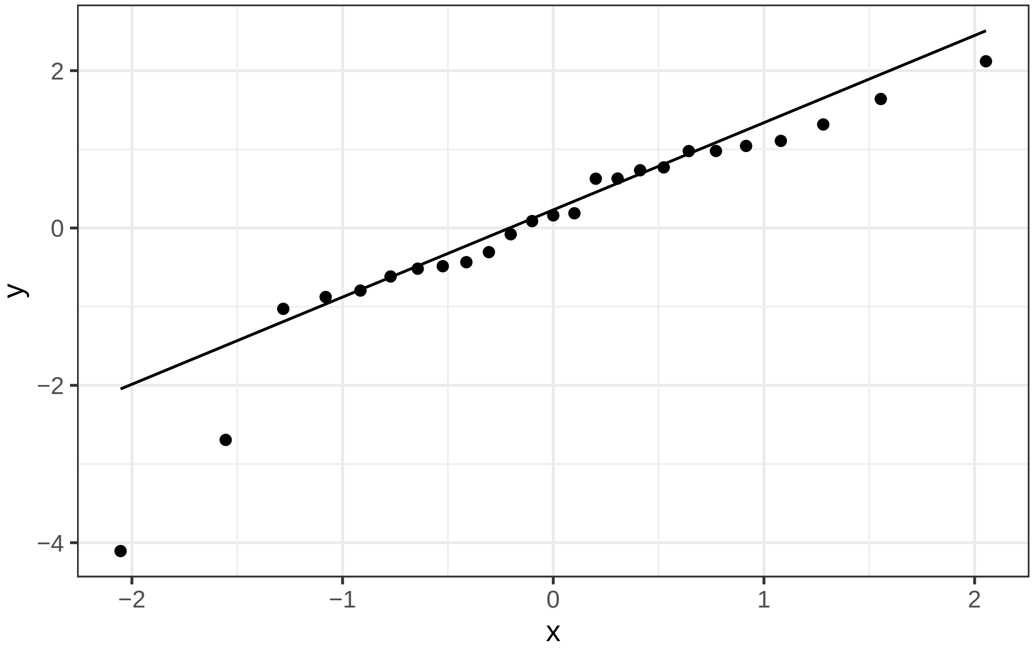
```
ggplot(data = vit_etoh, aes(x = .fitted, y = sqrt(abs(.scresid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



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:

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



It appears not too bad, except for two lower points. A Shapiro-Wilk test indicates non Normality (probably dues to the two extreme values):

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

```


	Shapiro-Wilk normality test

data:  vit_etoh$.scresid
W = 0.89958, p-value = 0.01797
```

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
```