

# Progressive motility

Marie Bisconti, Philippe Grosjean & Elise Hennebert

## Introduction

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

## DMSO

```
mp_dms0 <- readxl::read_excel("../data/Table S2.xlsx",  
  sheet = "progressive motility_DMSO")  
mp_dms0$donor <- as.factor(mp_dms0$donor)  
names(mp_dms0) <- c("donor", "conc", "prog", "total")  
mp_dms0$prog_frac <- mp_dms0$prog / mp_dms0$total  
skimr::skim(mp_dms0)
```

Table 1: Data summary

Name	mp_dms0
Number of rows	40
Number of columns	5
Column type frequency:	
factor	1
numeric	4
Group variables	None

**Variable type: factor**

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

### Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
conc	0	1	0.72	0.74	0.0	0.10	0.50	1.00	2.00	
prog	0	1	88.30	42.36	33.0	62.75	77.50	102.25	249.00	
total	0	1	131.20	56.45	54.0	92.00	124.50	150.50	301.00	
prog_frac	0	1	0.67	0.10	0.4	0.62	0.69	0.74	0.86	

There are eight donors, no missing data.

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

```

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

```

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

```
mp_dmso_m1 <- glmer(cbind(prog, total - prog) ~ conc + (conc | donor),
  data = mp_dmso, family = binomial(link = "logit"))
summary(mp_dmso_m1)
```

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

Data: mp\_dms0

AIC	BIC	logLik	deviance	df.resid
339.9	348.3	-164.9	329.9	35

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.8987	-0.9856	-0.2334	0.7372	6.0252

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
donor	(Intercept)	0.10858	0.3295	
	conc	0.01235	0.1111	-0.16

Number of obs: 40, groups: donor, 8

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.91013	0.12473	7.297	2.95e-13 ***
conc	-0.21091	0.05805	-3.633	0.00028 ***

---

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

Correlation of Fixed Effects:

	(Intr)
conc	-0.285

The model is:

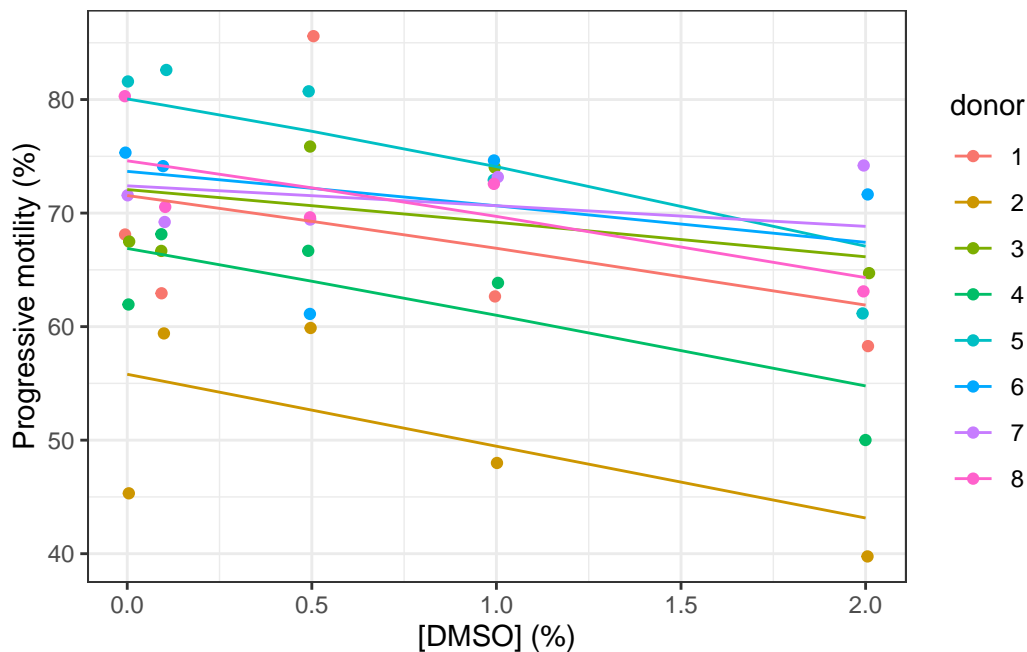
,

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

Here is a plot of this model:

```
ggplot(data = mp_dms0) +  
  geom_jitter(aes(x = conc, y = prog_frac * 100, col = donor),  
    width = 0.01) +
```

```
geom_line(aes(x = conc, y = fitted(mp_dmso_m1) * 100, col = donor)) +
labs(x = "[DMSO] (%)", y = "Progressive motility (%)")
```



Here, we have all negative slopes. Data are rather widespread. There are definitely shifts in the intercept per donor (different motility at [DMSO] = 0). Slopes seems different between donors. There is an extreme point for donor 1, DMSO 0.5%. Let's check if the model can be simplified with a likelihood ratio test:

```
mp_dmso_m2 <- glmer(cbind(prog, total - prog) ~ conc + (1 | donor),
  data = mp_dmso, family = binomial(link = "logit"))
anova(mp_dmso_m1, mp_dmso_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

Data: mp\_dmso

Models:

mp\_dmso\_m2: cbind(prog, total - prog) ~ conc + (1 | donor)

mp\_dmso\_m1: cbind(prog, total - prog) ~ conc + (conc | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mp_dmso_m2	3	337.14	342.21	-165.57	331.14			
mp_dmso_m1	5	339.87	348.32	-164.94	329.87	1.2721	2	0.5294

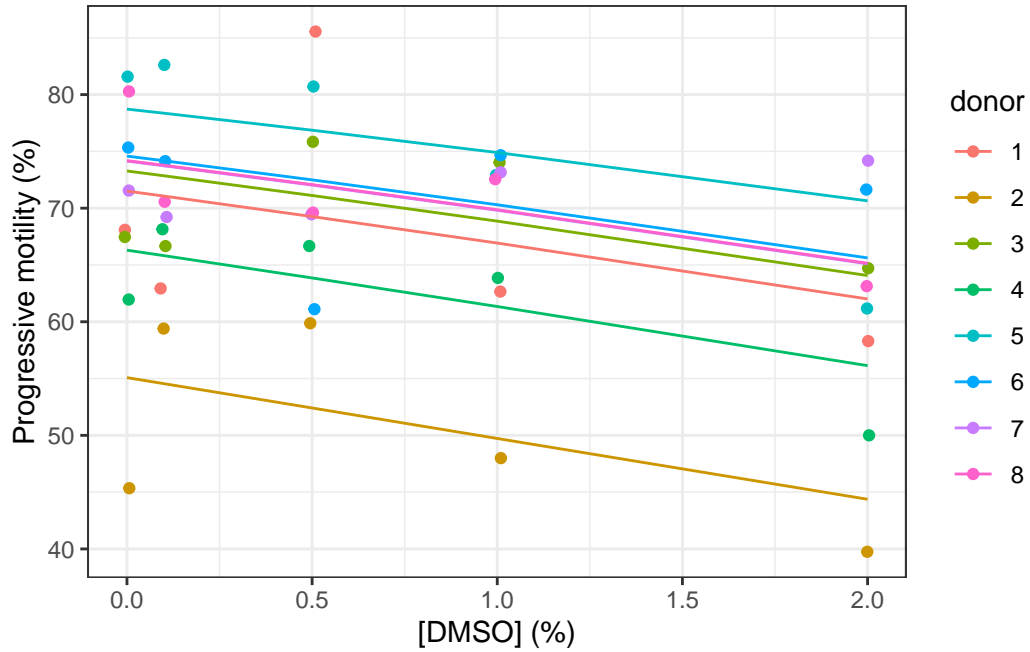
The likelihood ratio does not detect significant differences between the full and simplified models at  $\alpha = 5\%$ . We could use the simplest `mp_dmsom2` model with only a shift in the slope per donor.

This model is:

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

Here is a plot of this model:

```
ggplot(data = mp_dmsom) +
  geom_jitter(aes(x = conc, y = prog_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mp_dmsom2) * 100, col = donor)) +
  labs(x = "[DMSO] (%)", y = "Progressive motility (%)")
```



```
summary(mp_dms0_m2)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
Approximation) [glmerMod]
Family: binomial ( logit )
Formula: cbind(prog, total - prog) ~ conc + (1 | donor)
Data: mp_dms0
```

AIC	BIC	logLik	deviance	df.resid
337.1	342.2	-165.6	331.1	37

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.8853	-1.0519	-0.2145	0.8927	6.0275

Random effects:

Groups	Name	Variance	Std.Dev.
donor	(Intercept)	0.1069	0.3269

Number of obs: 40, groups: donor, 8

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.91430	0.12360	7.397	1.39e-13 ***
conc	-0.21504	0.04105	-5.239	1.62e-07 ***

---

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

Correlation of Fixed Effects:

	(Intr)
conc	-0.240

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(mp_dms0_m2, level = 0.95) # 95% CI based on profile
```

Computing profile confidence intervals ...

	2.5 %	97.5 %
.sig01	0.2049673	0.6001027
(Intercept)	0.6450188	1.1870224
conc	-0.2954498	-0.1344784

```
set.seed(964)
# 1000x parameter bootstrap
(mp_dmso_m2_conf <- confint(mp_dmso_m2, level = 0.95, method = "boot", nsim = 1000L))
```

Computing bootstrap confidence intervals ...

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

	2.5 %	97.5 %
.sig01	0.1276551	0.4774600
(Intercept)	0.6812327	1.1625425
conc	-0.3043773	-0.1325507

Among the 1000 bootstrapped models, two are singular. We can ignore this warning, since this impact is probably negligible on the overall calculations. 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(mp_dmso_m2, scope = "conc")
mp_dmso_m3 <- glmer(cbind(prog, total - prog) ~ 1 + (1 | donor),
  data = mp_dmso, family = binomial(link = "logit"))
anova(mp_dmso_m2, mp_dmso_m3, refit = TRUE)
```

Data: mp\_dmso

Models:

mp\_dmso\_m3: cbind(prog, total - prog) ~ 1 + (1 | donor)

mp\_dmso\_m2: cbind(prog, total - prog) ~ conc + (1 | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mp_dmso_m3	1	100.00	100.00	-50.00	100.00	1.00	1	1.00
mp_dmso_m2	2	95.00	95.00	-47.50	95.00	5.00	1	0.02

```
mp_dms0_m3      2 362.36 365.74 -179.18   358.36
mp_dms0_m2      3 337.14 342.21 -165.57   331.14 27.221   1  1.814e-07 ***
---
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(mp_dms0_m2)
```

```
[1] FALSE
```

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

```
mp_dms0_m2_all <- allFit(mp_dms0_m2)
```

```
Loading required namespace: dfoptim
```

```
Loading required namespace: optimx
```

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

```
summary(mp_dms0_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          0.9142878 -0.2150449
Nelder_Mead          0.9142827 -0.2150454
nlminbwrap          0.9142878 -0.2150458
nmkbw          0.9144157 -0.2150565
optimx.L-BFGS-B          0.9142877 -0.2150448
nloptwrap.NLOPT_LN_NELDERMEAD          0.9143021 -0.2150591
nloptwrap.NLOPT_LN_BOBYQA          0.9142875 -0.2150433

```

\$llik		
	bobyqa	Nelder_Mead
	-165.5716	-165.5716
	nlminbwrap	nmkbw
	-165.5716	-165.5716
	optimx.L-BFGS-B	nloptwrap.NLOPT_LN_NELDERMEAD
	-165.5716	-165.5716
	nloptwrap.NLOPT_LN_BOBYQA	
	-165.5716	

\$sdcor		donor.(Intercept)
	bobyqa	0.3269477
	Nelder_Mead	0.3269484
	nlminbwrap	0.3269484
	nmkbw	0.3268936
	optimx.L-BFGS-B	0.3269497
	nloptwrap.NLOPT_LN_NELDERMEAD	0.3269479
	nloptwrap.NLOPT_LN_BOBYQA	0.3269549

\$theta		donor.(Intercept)
	bobyqa	0.3269477
	Nelder_Mead	0.3269484
	nlminbwrap	0.3269484
	nmkbw	0.3268936
	optimx.L-BFGS-B	0.3269497
	nloptwrap.NLOPT_LN_NELDERMEAD	0.3269479
	nloptwrap.NLOPT_LN_BOBYQA	0.3269549

\$times		user.self	sys.self	elapsed	user.child	sys.child
	bobyqa	0.046	0.000	0.046	0	0
	Nelder_Mead	0.058	0.000	0.058	0	0
	nlminbwrap	0.048	0.000	0.048	0	0
	nmkbw	0.063	0.000	0.063	0	0
	optimx.L-BFGS-B	0.341	0.002	0.343	0	0
	nloptwrap.NLOPT_LN_NELDERMEAD	0.055	0.000	0.055	0	0
	nloptwrap.NLOPT_LN_BOBYQA	0.040	0.000	0.040	0	0

\$feval		
	bobyqa	Nelder_Mead
	70	87

```

      nlminbwrap      nmkbw
      NA             96
optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
      12             97
nloptwrap.NLOPT_LN_BOBYQA
      30

```

```

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

```

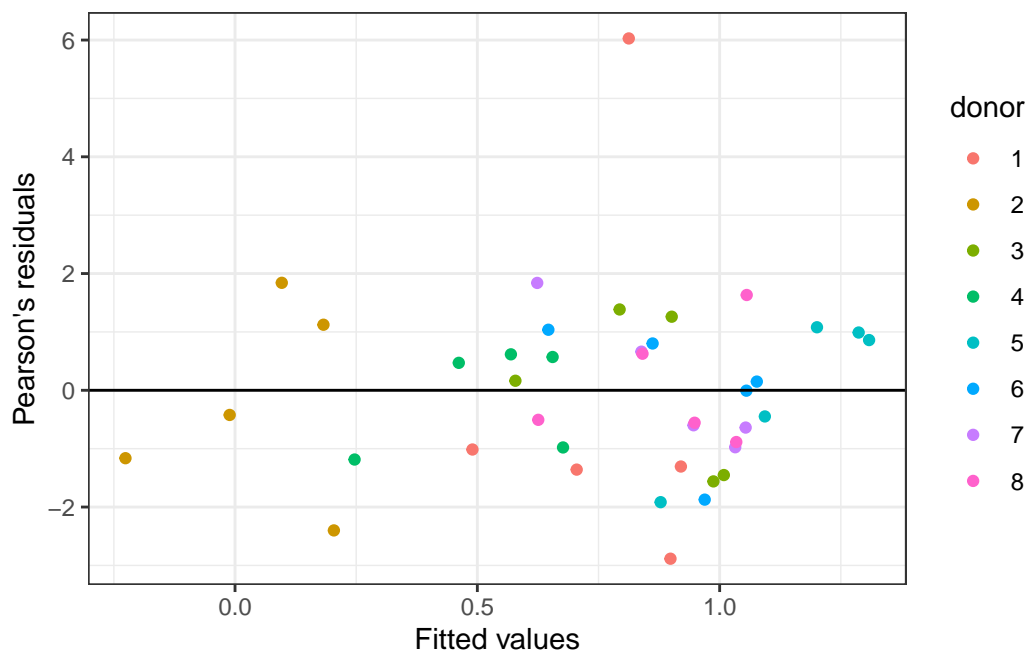
## Analysis of the residuals

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

```

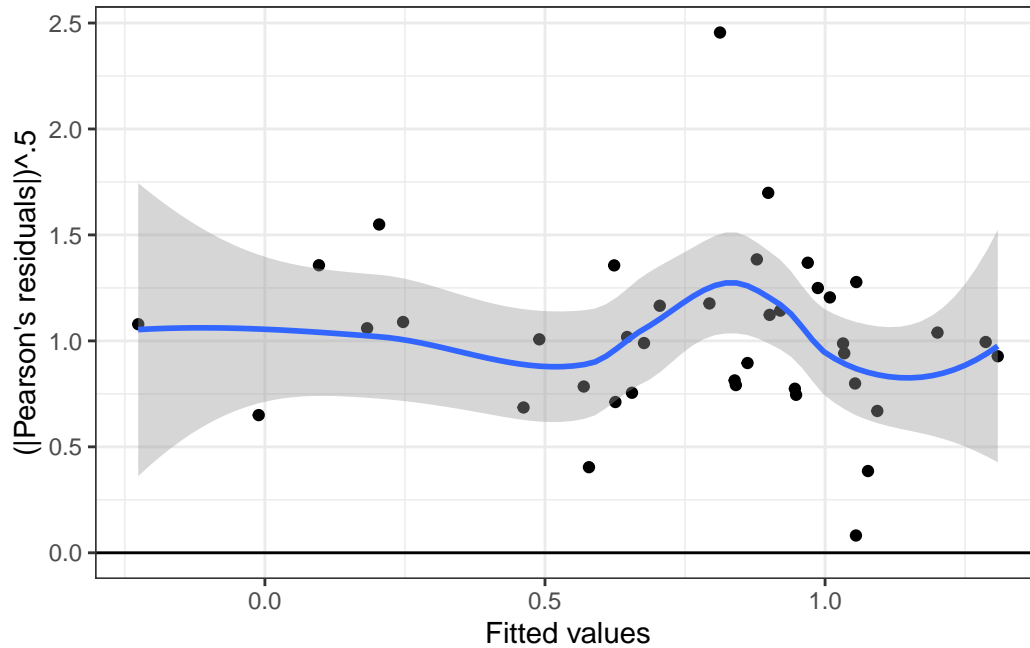
mp_dmso <- fortify.merMod(mp_dmso_m2)
ggplot(data = mp_dmso, aes(x = .fitted, y = .scre resid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")

```



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

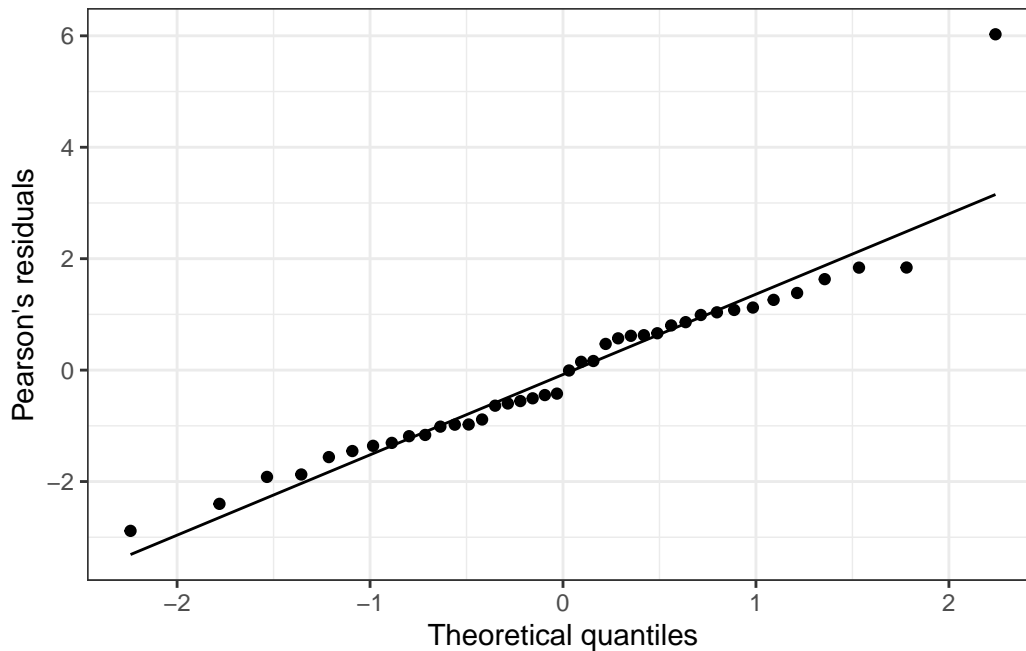
```
ggplot(data = mp_dmsso, aes(x = .fitted, y = sqrt(abs(.screid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



Homoscedasticity of the residuals seems here acceptable (the blue curve that is a loess smoothing in the data is relatively horizontal). Of course, the extreme value impacts the loess curve locally, but otherwise the variance in the residuals is homogeneous along fitted values.

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



It appears to be excellent, except for our extreme value, of course. A Shapiro-Wilk test does not confirm Normality, but we are pretty sure it is caused by the extreme value when looking at the quantile-quantile plot:

```
shapiro.test(mp_dms0$.sresid)
```

Shapiro-Wilk normality test

```
data: mp_dms0$.sresid
W = 0.90587, p-value = 0.002846
```

## Predictions

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

```
mp_dms0_slope <- c(
  ci95_min = min(mp_dms0_m2_conf["conc", ]),
  estimate = fixef(mp_dms0_m2)[["conc"]],
  ci95_max = max(mp_dms0_m2_conf["conc", ]))
```

```
mp_dms0_slope
```

```
      ci95_min    estimate    ci95_max  
-0.3043773 -0.2150449 -0.1325507
```

```
#saveRDS(mp_dms0_slope, "../data/motility_prog_DMS0_slope.rds")
```

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

```
predict_logit <- function(conc, intercept = 1, slopes) {  
  slopes_mat <- matrix(slopes, nrow = 1,  
    dimnames = list(NULL, names(slopes)))  
  data.frame(conc = conc, -intercept +  
    boot::inv.logit(boot::logit(intercept) +  
    conc %*% slopes_mat))  
}  
dms0_conc <- (0:20) / 10  
mp_dms0_lost <- predict_logit(dms0_conc, 0.8, mp_dms0_slope)  
mp_dms0_lost
```

	conc	ci95_min	estimate	ci95_max
1	0.0	0.000000000	0.000000000	0.000000000
2	0.1	-0.004914533	-0.003462926	-0.002129247
3	0.2	-0.009918142	-0.006970298	-0.004275373
4	0.3	-0.015010889	-0.010522152	-0.006438391
5	0.4	-0.020192755	-0.014118503	-0.008618308
6	0.5	-0.025463637	-0.017759344	-0.010815128
7	0.6	-0.030823344	-0.021444650	-0.013028855
8	0.7	-0.036271596	-0.025174371	-0.015259487
9	0.8	-0.041808018	-0.028948437	-0.017507021
10	0.9	-0.047432141	-0.032766757	-0.019771449
11	1.0	-0.053143399	-0.036629214	-0.022052762
12	1.1	-0.058941127	-0.040535669	-0.024350946
13	1.2	-0.064824556	-0.044485961	-0.026665986
14	1.3	-0.070792815	-0.048479903	-0.028997861
15	1.4	-0.076844928	-0.052517282	-0.031346550
16	1.5	-0.082979813	-0.056597864	-0.033712025

```

17  1.6 -0.089196280 -0.060721386 -0.036094258
18  1.7 -0.095493030 -0.064887562 -0.038493216
19  1.8 -0.101868656 -0.069096079 -0.040908863
20  1.9 -0.108321642 -0.073346597 -0.043341158
21  2.0 -0.114850360 -0.077638750 -0.045790059

```

```
#saveRDS(mp_dmso_lost, "../data/motility_prog_DMSO_lost.rds")
```

This is the lost in progressive motility that the model predicts.

## Ethanol

```

mp_etoh <- readxl::read_excel("../data/Table S2.xlsx",
  sheet = "progressive motility_EtOH")
mp_etoh$donor <- as.factor(mp_etoh$donor)
names(mp_etoh) <- c("donor", "conc", "prog", "total")
mp_etoh$prog_frac <- mp_etoh$prog / mp_etoh$total
skimr::skim(mp_etoh)

```

Table 4: Data summary

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

### Variable type: factor

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

### Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
conc	0	1	0.72	0.74	0.00	0.10	0.50	1.00	2.00	
prog	0	1	79.80	41.47	30.00	49.75	70.50	90.25	211.00	
total	0	1	127.20	56.93	66.00	87.00	117.00	140.25	301.00	
prog_frac	0	1	0.63	0.14	0.27	0.53	0.68	0.72	0.82	

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

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

```

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

```

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

```
mp_etoh_m1 <- glmer(cbind(prog, total - prog) ~ conc + (conc | donor),
  data = mp_etoh, family = binomial(link = "logit"))
summary(mp_etoh_m1)
```

Generalized linear mixed model fit by maximum likelihood (Laplace

Approximation) [glmerMod]

Family: binomial ( logit )

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

Data: mp\_etoh

AIC	BIC	logLik	deviance	df.resid
344.5	352.9	-167.2	334.5	35

Scaled residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----



-4.2577 -0.8537 0.1724 0.9715 2.1316

Random effects:

Groups	Name	Variance	Std.Dev.	Corr
donor	(Intercept)	0.192212	0.43842	
	conc	0.006819	0.08258	0.00

Number of obs: 40, groups: donor, 8

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.80299	0.16168	4.967	6.81e-07 ***
conc	-0.34557	0.05287	-6.536	6.31e-11 ***

---

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

Correlation of Fixed Effects:

	(Intr)
conc	-0.172

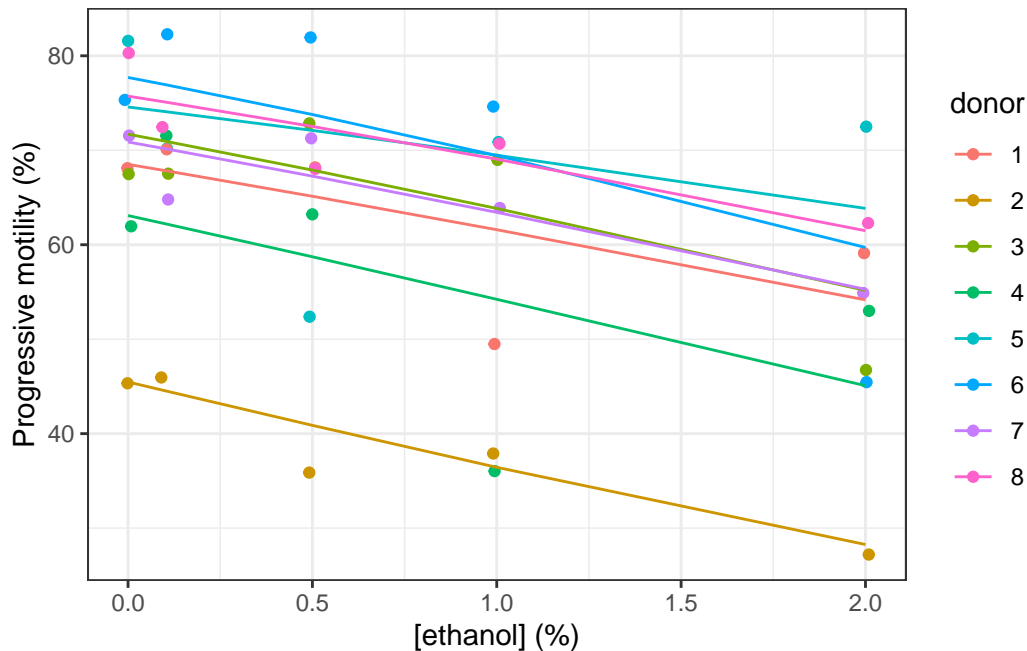
The model is:

,

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

Here is a plot of this model:

```
ggplot(data = mp_eto) +
  geom_jitter(aes(x = conc, y = prog_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mp_eto_m1) * 100, col = donor)) +
  labs(x = "[ethanol] (%)", y = "Progressive motility (%)")
```



Slopes are all negative, suggesting a negative concentration effect. Data are rather widespread. There are definitely shifts in the intercept per donor (different motility at [etoh] = 0). Slopes seems very similar between donors. Let's check if the model can be simplified using a likelihood ratio test:

```
mp_etoh_m2 <- glmer(cbind(prog, total - prog) ~ conc + (1 | donor),
  data = mp_etoh, family = binomial(link = "logit"))
anova(mp_etoh_m1, mp_etoh_m2, refit = FALSE) # Despite the name, it is indeed a LR test
```

Data: mp\_etoh

Models:

mp\_etoh\_m2: cbind(prog, total - prog) ~ conc + (1 | donor)

mp\_etoh\_m1: cbind(prog, total - prog) ~ conc + (conc | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mp_etoh_m2	3	341.03	346.09	-167.51	335.03			
mp_etoh_m1	5	344.46	352.91	-167.23	334.46	0.5634	2	0.7545

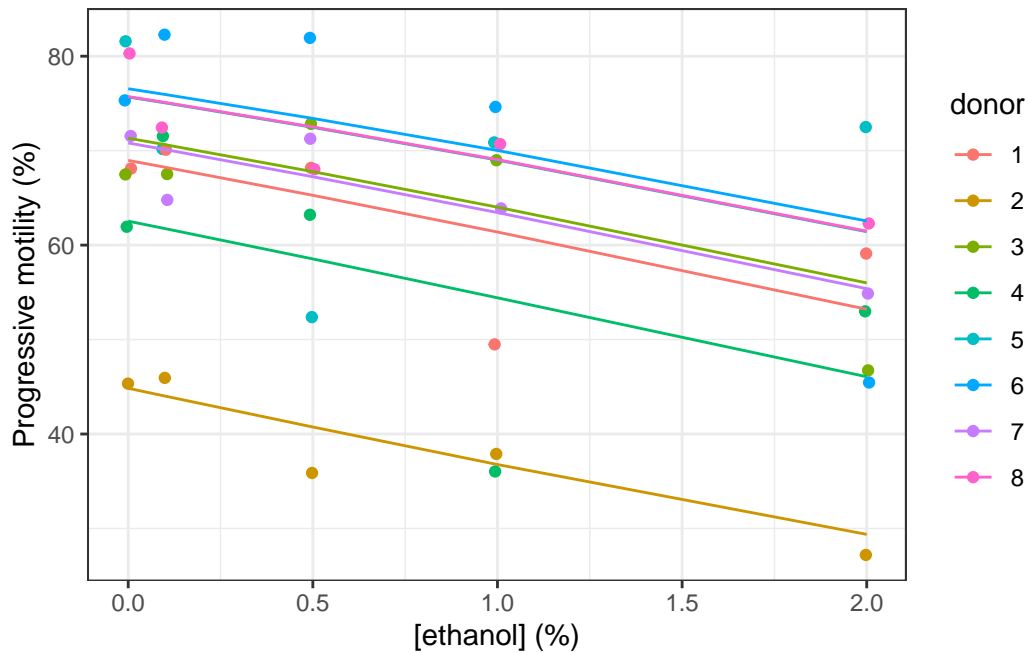
The likelihood ratio test does not detect significant differences between the full and simplified models at  $\alpha = 5\%$ . We could thus use the simplest mp\_etoh\_m2 model with only a shift in the slope per donor. This model is:

,

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

Here is a plot of this model:

```
ggplot(data = mp_etoh) +
  geom_jitter(aes(x = conc, y = prog_frac * 100, col = donor),
    width = 0.01) +
  geom_line(aes(x = conc, y = fitted(mp_etoh_m2) * 100, col = donor)) +
  labs(x = "[ethanol] (%)", y = "Progressive motility (%)")
```



```
summary(mp_etoh_m2)
```

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

Data: mp\_eto

AIC	BIC	logLik	deviance	df.resid
341.0	346.1	-167.5	335.0	37

Scaled residuals:

Min	1Q	Median	3Q	Max
-4.3068	-0.7902	0.1791	0.9763	2.4924

Random effects:

Groups	Name	Variance	Std.Dev.
donor	(Intercept)	0.1947	0.4413

Number of obs: 40, groups: donor, 8

Fixed effects:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.79579	0.16194	4.914	8.93e-07 ***
conc	-0.33502	0.04006	-8.362	< 2e-16 ***

---

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

Correlation of Fixed Effects:

	(Intr)
conc	-0.182

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(mp_eto_m2, level = 0.95) # 95% CI based on profile
```

Computing profile confidence intervals ...

	2.5 %	97.5 %
.sig01	0.2816489	0.8030597
(Intercept)	0.4397141	1.1543188
conc	-0.4136293	-0.2565405

```

set.seed(2784)
# 1000x parameter bootstrap
(mp_etoh_m2_conf <- confint(mp_etoh_m2, level = 0.95,
  method = "boot", nsim = 1000L))

```

Computing bootstrap confidence intervals ...

1 warning(s): Model failed to converge with max|grad| = 0.00694994 (tol = 0.002, component 1)

	2.5 %	97.5 %
.sig01	0.1895877	0.6227712
(Intercept)	0.4836082	1.1217895
conc	-0.4130732	-0.2636878

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

## Additional verifications

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

```

#drop1(mp_etoh_m2, scope = "conc")
mp_etoh_m3 <- glmer(cbind(prog, total - prog) ~ 1 + (1 | donor),
  data = mp_etoh, family = binomial(link = "logit"))
anova(mp_etoh_m2, mp_etoh_m3, refit = FALSE)

```

Data: mp\_etoh

Models:

mp\_etoh\_m3: cbind(prog, total - prog) ~ 1 + (1 | donor)

mp\_etoh\_m2: cbind(prog, total - prog) ~ conc + (1 | donor)

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mp_etoh_m3	2	409.02	412.40	-202.51	405.02			
mp_etoh_m2	3	341.03	346.09	-167.51	335.03	69.995	1	< 2.2e-16 ***

---

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 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(mp_etoh_m2)
```

```
[1] FALSE
```

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

```
mp_etoh_m2_all <- allFit(mp_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(mp_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\$nlptwrap.NLOPT\_LN\_NELDERMEAD  
NULL

\$msgs\$nlptwrap.NLOPT\_LN\_BOBYQA  
NULL

\$fixef

	(Intercept)	conc
bobyqa	0.7957873	-0.3350232
Nelder_Mead	0.7957923	-0.3350248
nlminbwrap	0.7957905	-0.3350226
nmkbw	0.7957880	-0.3349973
optimx.L-BFGS-B	0.7957876	-0.3350233
nlptwrap.NLOPT_LN_NELDERMEAD	0.7957571	-0.3350295
nlptwrap.NLOPT_LN_BOBYQA	0.7957865	-0.3350226

\$llik

	bobyqa	Nelder_Mead
	-167.5134	-167.5134
	nlminbwrap	nmkbw
	-167.5134	-167.5134
	optimx.L-BFGS-B	nlptwrap.NLOPT_LN_NELDERMEAD
	-167.5134	-167.5134
	nlptwrap.NLOPT_LN_BOBYQA	
	-167.5134	

\$sdcor

	donor.(Intercept)
bobyqa	0.4412697
Nelder_Mead	0.4412728

```

nlminbwrap                0.4412718
nmkbw                     0.4413067
optimx.L-BFGS-B           0.4412720
nloptwrap.NLOPT_LN_NELDERMEAD 0.4412887
nloptwrap.NLOPT_LN_BOBYQA  0.4412697

$theta
              donor.(Intercept)
bobyqa                0.4412697
Nelder_Mead           0.4412728
nlminbwrap            0.4412718
nmkbw                 0.4413067
optimx.L-BFGS-B       0.4412720
nloptwrap.NLOPT_LN_NELDERMEAD 0.4412887
nloptwrap.NLOPT_LN_BOBYQA  0.4412697

$times
              user.self sys.self elapsed user.child sys.child
bobyqa                0.042   0.000   0.042         0         0
Nelder_Mead           0.061   0.000   0.062         0         0
nlminbwrap            0.052   0.000   0.054         0         0
nmkbw                 0.063   0.001   0.065         0         0
optimx.L-BFGS-B       0.350   0.002   0.353         0         0
nloptwrap.NLOPT_LN_NELDERMEAD 0.054   0.000   0.055         0         0
nloptwrap.NLOPT_LN_BOBYQA  0.041   0.000   0.042         0         0

$feval
              bobyqa                Nelder_Mead
              48                95
              nlminbwrap                nmkbw
              NA                104
              optimx.L-BFGS-B nloptwrap.NLOPT_LN_NELDERMEAD
              14                94
              nloptwrap.NLOPT_LN_BOBYQA
              30

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

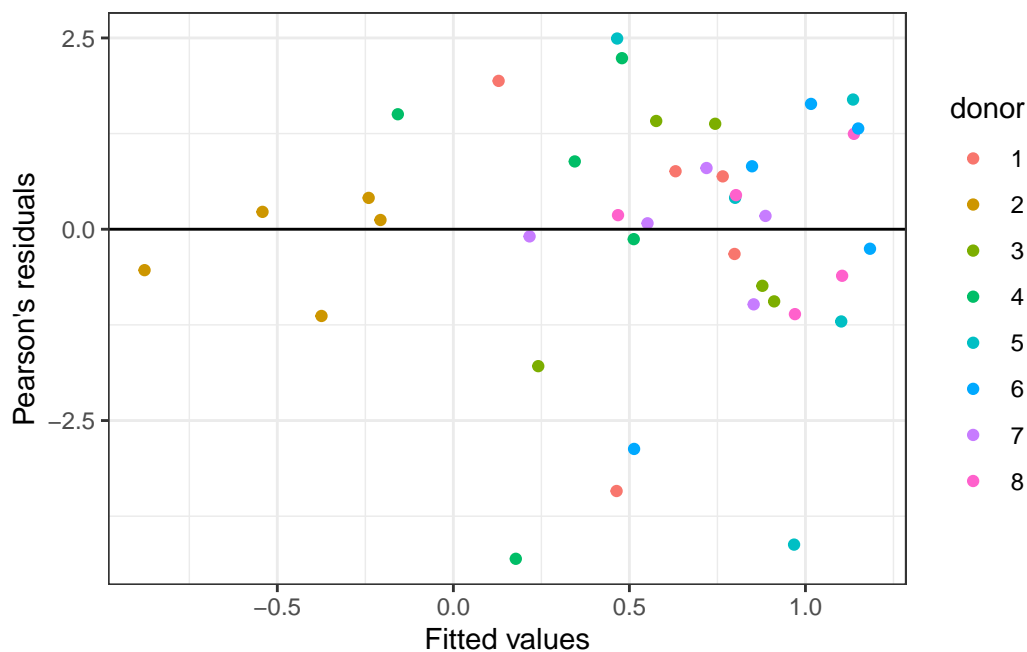
```

### Analysis of the residuals

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

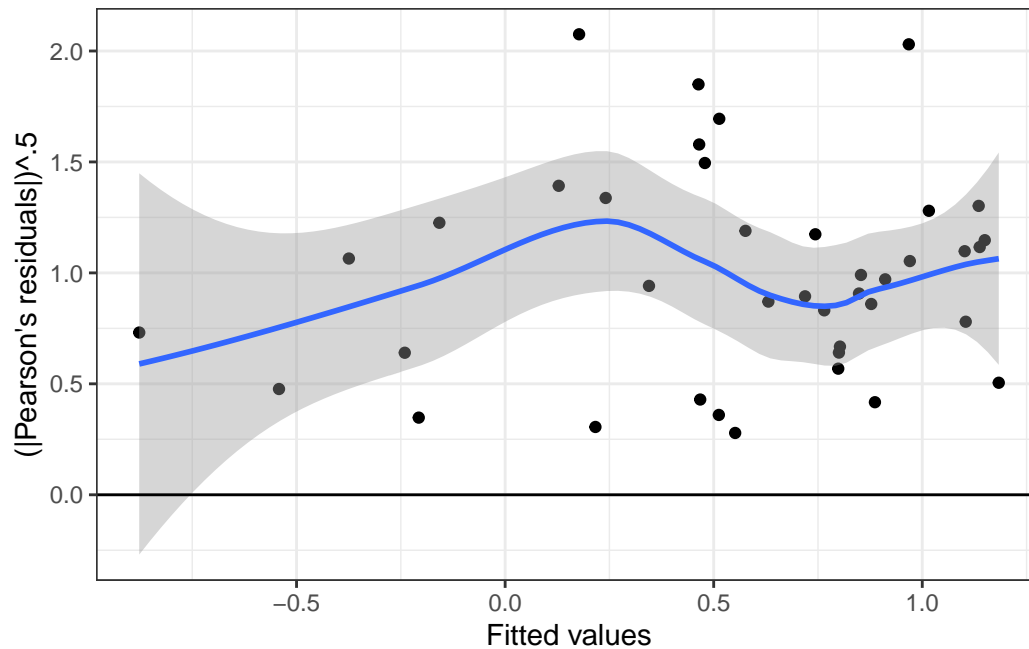


```
mp_etoh <- fortify.merMod(mp_etoh_m2)
ggplot(data = mp_etoh, aes(x = .fitted, y = .screid, col = donor)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "Pearson's residuals")
```



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

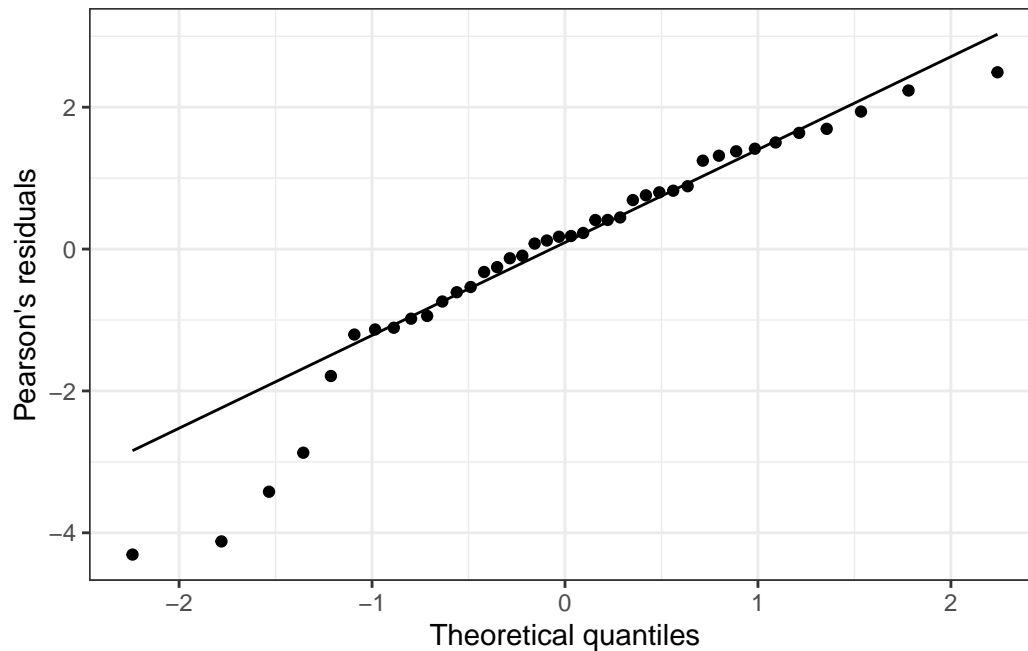
```
ggplot(data = mp_etoh, aes(x = .fitted, y = sqrt(abs(.screid)))) +
  geom_point() +
  geom_smooth(method = "loess", formula = y ~ x) +
  geom_hline(yintercept = 0) +
  labs(x = "Fitted values", y = "(|Pearson's residuals|)^.5")
```



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



It appears not Normal for. A Shapiro-Wilk test confirms non Normality:

```
shapiro.test(mp_etoh$.scre resid)
```

Shapiro-Wilk normality test

```
data: mp_etoh$.scre resid
W = 0.92514, p-value = 0.01121
```

## Predictions

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

```
mp_etoh_slope <- c(
  ci95_min = min(mp_etoh_m2_conf["conc", ]),
  estimate = fixef(mp_etoh_m2)[["conc"]],
  ci95_max = max(mp_etoh_m2_conf["conc", ])
)
mp_etoh_slope
```

```

ci95_min    estimate    ci95_max
-0.4130732 -0.3350244 -0.2636878

```

```

#saveRDS(mp_etoh_slope, "../data/motility_prog_ETOH_slope.rds")

```

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

```

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

```

	conc	ci95_min	estimate	ci95_max
1	0.0	0.000000000	0.000000000	0.000000000
2	0.1	-0.006691138	-0.005414301	-0.004252397
3	0.2	-0.013546378	-0.010936529	-0.008571636
4	0.3	-0.020565768	-0.016566750	-0.012957769
5	0.4	-0.027749071	-0.022304912	-0.017410801
6	0.5	-0.035095756	-0.028150837	-0.021930692
7	0.6	-0.042604977	-0.034104214	-0.026517350
8	0.7	-0.050275572	-0.040164602	-0.031170638
9	0.8	-0.058106045	-0.046331419	-0.035890365
10	0.9	-0.066094558	-0.052603943	-0.040676287
11	1.0	-0.074238925	-0.058981303	-0.045528109
12	1.1	-0.082536602	-0.065462483	-0.050445480
13	1.2	-0.090984682	-0.072046313	-0.055427992
14	1.3	-0.099579891	-0.078731468	-0.060475183
15	1.4	-0.108318583	-0.085516469	-0.065586531
16	1.5	-0.117196743	-0.092399677	-0.070761457
17	1.6	-0.126209984	-0.099379293	-0.075999323
18	1.7	-0.135353550	-0.106453359	-0.081299428
19	1.8	-0.144622323	-0.113619756	-0.086661014
20	1.9	-0.154010826	-0.120876203	-0.092083261

```
21 2.0 -0.163513232 -0.128220261 -0.097565288
```

```
#saveRDS(mp_etoh_lost, "../data/motility_prog_ETOH_lost.rds")
```

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

## General informations

```
sessionInfo()
```

```
R version 4.1.3 (2022-03-10)
```

```
Platform: x86_64-apple-darwin17.0 (64-bit)
```

```
Running under: macOS Big Sur/Monterey 10.16
```

```
Matrix products: default
```

```
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
```

```
locale:
```

```
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
attached base packages:
```

```
[1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
other attached packages:
```

```
[1] ggplot2_3.3.5 lme4_1.1-29  Matrix_1.4-1
```

```
loaded via a namespace (and not attached):
```

```
[1] tidyr_1.2.0      jsonlite_1.8.0    splines_4.1.3
[4] equationomatic_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		