TMA4315 Generalized linear models H2018

Module 8: Generalized linear mixed effects models (GLMM)

Mette Langaas and Ingeborg Hem, Department of Mathematical Sciences, NTNU

15.11.2018 [PL], 16.11.2018 [IL]

Contents

(Latest changes: 16.11: IL added links, 13.11.2018 - first version).

Overview

Aim: Present methods for analysing correlated responses in a generalized linear models setting - LMM meets GLM to become *GLMM*.

Also here (as in module LMM) we will only consider *two-level models* and in particular focus on *random intercept models* for binomial and Poisson responses. Emphasis will be on understanding.

Learning material

- Textbook: Fahrmeir et al (2013): Chapter 7.5: 389-394, 7.7.
- Classnotes (15.11.2018)

Topics

- beaches example - revisited
- notation
- the generalized linear mixed effect model (three ingredients)
- the GLMM with random intercept
- the marginal model
- parameter estimation and Laplace approximation
- summing up: what do we need to know about the GLMM?
- additional info on different software (not on reading list)

Jump to interactive.

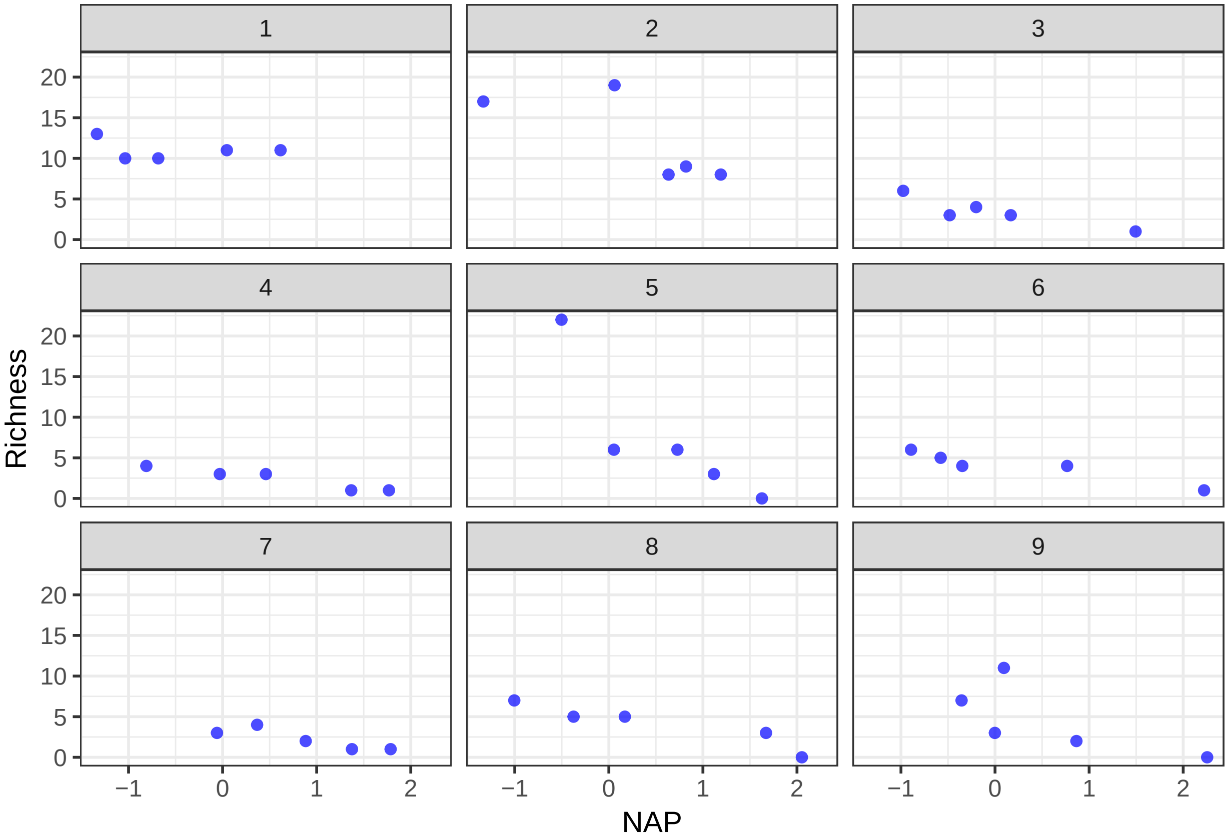
Beaches example - revisited

This example is taken from Zuur et al. (2009, chapter 5, pages 101-142), and data are referred to as RIKZ. Data were collected on nine different beaches in the Netherlands with the aim to investigate if there is a relationship between the

- richness of species (number of species observed) and
- NAP: the height of the sampling station compared to mean tidal level.

Data: 45 observations, taken at 9 beaches:

- beach: the beach that the samples were taken, for each beach 5 different samples were taken.



We want to use the richness of species as the response in a regression model. In Module LMM we assumed that the response could be viewed as a normal variable, but now we rather assume that the response comes from a *Poisson distribution*.

Q: Why do we want to assume a Poisson distribution?

A:

We observe events that may occur within a time interval or a region.

1. The number of events occuring within a time interval or a region, is independent of the number of events that occurs in any other disjoint (non-overlapping) time interval or region.
2. The probability that a single event occurs within a small time interval or region, is proportional to the length of the interval or the size of the region.
3. The probability that more than one event may occur within a small time interval or region is negligable.

When all of these three properties are funfilled we have a *Poisson process*.

- The number of events in a Poisson process follows a Poisson distribution.

We go ahead and fit a Poission GLM with log-link and NAP as covariate. In Module LMM we talked about possible solutions to how to handle multiple observations for each beach and came up with:

Possible solutions:

1. Use all 45 observations in a Poisson regression - with a common intercept and linear effect in NAP. **Problem:** violation of assumption of independent observations lead to wrong estimates for variances

of parameter estimates.

2. Add beach as covariate to Poisson regression - then we estimate one regression coefficient for each beach (intercepts) in addition to the linear effect in NAP. **Problem:** why do we want to add 8 extra parameters to estimate (why 8?) values for the 9 beaches? Loss of power and what would we use the beach estimates for?
3. Add beach as a *random* covariate to the Poisson regression: this is called *random intercept models*. **Problem:** new stuff - slightly complicated. We do this because beaches not of interest in themselves, only random sample from population of beaches, and therefore we only need to account for beaches, not estimate separate parameters.

But - first: what do we remember about GLMs (and Poission regression with log-link in particular)?

Assumptions for GLM model (Poisson):

1. Random component from exponential family: We have for our beaches: $Y_i \sim \text{Poisson}(\lambda_i)$, with $\text{E}(Y_i) = \lambda_i$, and $\text{Var}(Y_i) = \lambda_i$.
2. Linear predictor: $\eta_i = \mathbf{x}_i^T \boldsymbol{\beta}$.
3. Log link

   $$\eta_i = \ln(\lambda_i) = g(\lambda_i)$$

   and (inverse thereof) response function

   $$\lambda_i = \exp(\eta_i)$$

   Canonical link for Poisson is log.

Parameter estimation is performed using the method of maximum likelihood, numerically using Fisher scoring.

Solution 1: all observations in one GLM together - standard errors not correct since assumption of independent obsevations is violated.

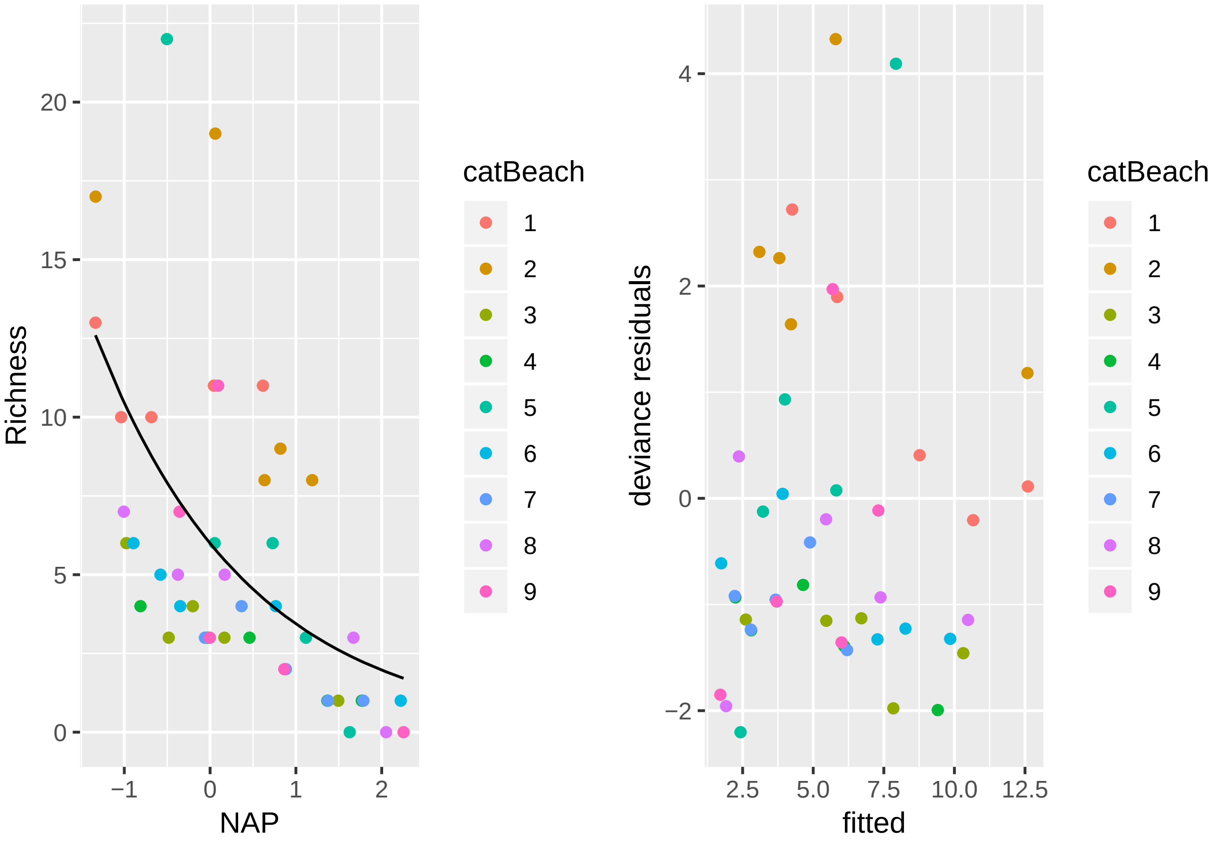
Remember: For interpretation. If x_{i1} increases by one unit to $x_{i1} + 1$ then the mean $\text{E}(Y_i)$ will in our model change by a *factor* $\exp(\beta_1)$. Not so easy to compare to the normal case.

```r
fitall = glm(Richness ~ NAP, data = RIKZ, family = poisson(link = log))
summary(fitall)
fitall$coefficients[2]
exp(fitall$coefficients[2])
```

```
##
## Call:
## glm(formula = Richness ~ NAP, family = poisson(link = log), data = RIKZ)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2029  -1.2432  -0.9199   0.3943   4.3256
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.79100    0.06329  28.297  < 2e-16 ***
## NAP         -0.55597    0.07163  -7.762 8.39e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 179.75  on 44  degrees of freedom
## Residual deviance: 113.18  on 43  degrees of freedom
## AIC: 259.18
```

```
##
## Number of Fisher Scoring iterations: 5
##
##        NAP
## -0.5559735
##       NAP
## 0.5735136
```

```
library(ggplot2)
library(ggpubr)
df=data.frame("Y"=RIKZ$Richness,"x"=RIKZ$NAP,"fitted"=fitall$fitted.values,"dres"=residuals(fitall,type="deviance")
gg1=ggplot(df)+geom_point(aes(x=x,y=Y,colour=catBeach))+geom_line(aes(x=x,y=fitted))+xlab("NAP")+ylab("Richness")
gg2=ggplot(df)+geom_point(aes(x=fitted,y=dres,color=catBeach))+xlab("fitted")+ylab("deviance residuals")
ggarrange(gg1,gg2)
```



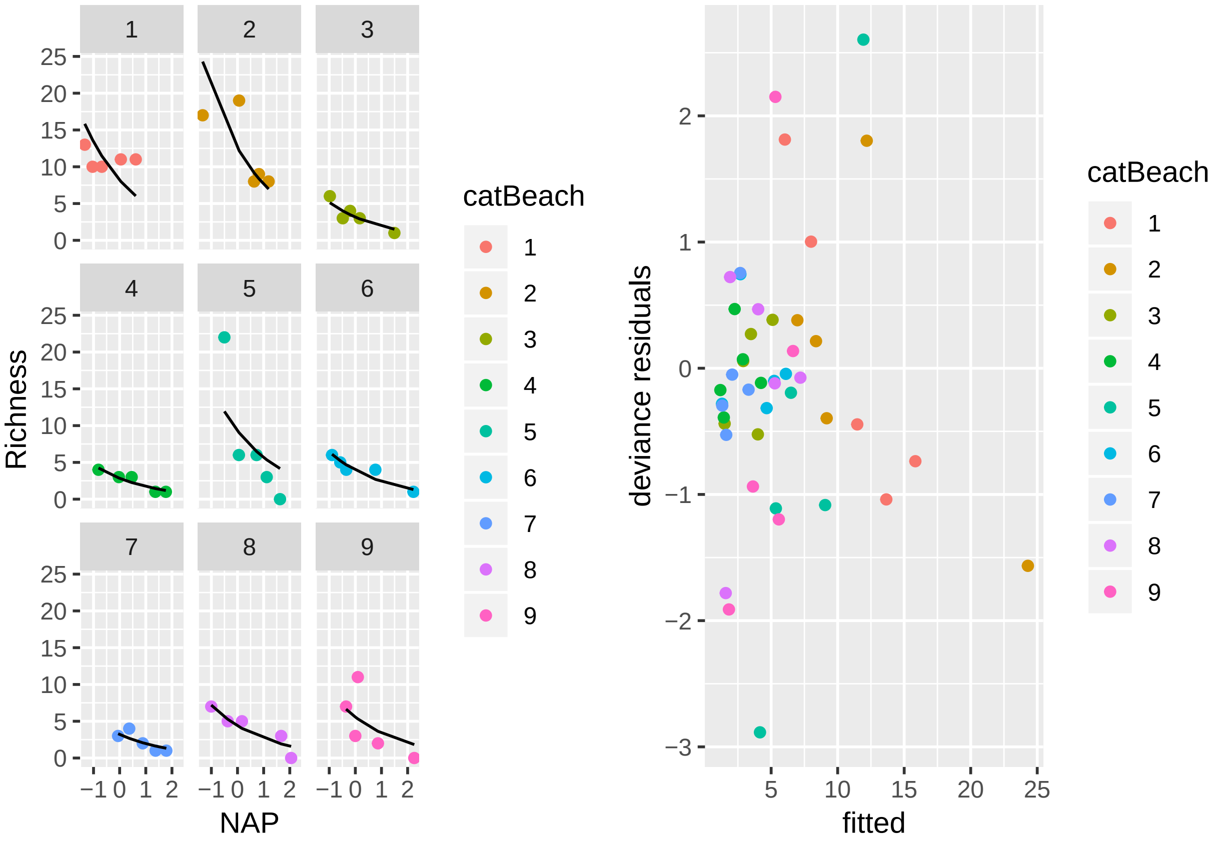
Q: **What is the formula for the black line on the left panel? A**: see classnotes.

Solution 2: fixed effects for each beach - many estimates not so much of interest when population is in focus.

```
RIKZ$beachfactor = as.factor(RIKZ$Beach)
fitbeach = glm(Richness ~ NAP + beachfactor, data = RIKZ, contrasts = list(beachfactor = "contr.sum"),
    family = poisson(link = "log"))
summary(fitbeach)
```

```
##
## Call:
## glm(formula = Richness ~ NAP + beachfactor, family = poisson(link = "log"),
##     data = RIKZ, contrasts = list(beachfactor = "contr.sum"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8849  -0.5240  -0.1194   0.3805   2.6037
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.64239    0.07356  22.326  < 2e-16 ***
## NAP           -0.49467    0.07642  -6.473 9.61e-11 ***
## beachfactor1   0.45928    0.14943   3.074 0.002115 **
## beachfactor2   0.88808    0.13527   6.565 5.20e-11 ***
## beachfactor3  -0.49422    0.22710  -2.176 0.029540 *
## beachfactor4  -0.59982    0.26525  -2.261 0.023740 *
## beachfactor5   0.58824    0.16469   3.572 0.000355 ***
## beachfactor6  -0.27433    0.21124  -1.299 0.194068
## beachfactor7  -0.47708    0.28070  -1.700 0.089202 .
## beachfactor8  -0.16553    0.21057  -0.786 0.431786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 179.753  on 44  degrees of freedom
## Residual deviance:  47.073  on 35  degrees of freedom
## AIC: 209.07
##
## Number of Fisher Scoring iterations: 5
```

```r
library(ggplot2)
library(ggpubr)
df=data.frame("Y"=RIKZ$Richness,"x"=RIKZ$NAP,"fitted"=fitbeach$fitted.values,"dres"=residuals(fitbeach,type="devian
gg1=ggplot(df)+geom_point(aes(x=x,y=Y,colour=catBeach))+geom_line(aes(x=x,y=fitted))+xlab("NAP")+ylab("Richness")+f
gg2=ggplot(df)+geom_point(aes(x=fitted,y=dres,color=catBeach))+xlab("fitted")+ylab("deviance residuals")
ggarrange(gg1,gg2)
```



Generalized linear mixed effects models

Notation

combining GLM (Module 5) and LMM (Module 7)

We have n_i repeated observations ($j = 1, \ldots, n_i$) from each of $i = 1, \ldots, m$ clusters (or individuals).

- Responses: $Y_{i1}, Y_{i2}, \ldots, Y_{in_i}$ (e.g. species richness at beach i sample j)
- Covariates: $\mathbf{x}_{i1}, \mathbf{x}_{i1}, \ldots, \mathbf{x}_{in_i}$ (e.g. NAP for beach i sample j)

The covariates $\mathbf{x}_{ij}$ are $p \times 1$ vectors (as before, k covariates and one intercept so $p = k + 1$).

Repetition GLM

1. **Random component** - exponential family (now y_{ij})

$$f(y_{ij} \mid \theta_{ij}) = \exp\left(\frac{y_{ij}\theta_{ij} - b(\theta_{ij})}{\phi} \cdot w_{ij} + c(y_{ij}, \phi, w_{ij})\right)$$

- θ_{ij} is called the canonical parameter and is a parameter of interest
- ϕ is called a nuisance parameter (and is not of interest to us=therefore a nuisance (plage))
- w_{ij} is a weight function, in most cases $w_{ij} = 1$
- b and c are known functions.

We looked in detail into the binomial, Poisson and gamma distribution (in addition to the normal).

2. **Linear predictor** (previously “systematic component”):

$$\eta_{ij} = \mathbf{x}_{ij}^T \boldsymbol{\beta}$$

3. **Link function g- and response function h**

$$\eta_{ij} = g(\mu_{ij})$$

$$\mu_{ij} = h(\eta_{ij})$$

New element

we add a random component (effect) to the *linear predictor*!

2. $$\eta_{ij} = \mathbf{x}_{ij}^T \boldsymbol{\beta} + \mathbf{u}_{ij}^T \gamma_i$$

   and we make the same assumptions as for the LMM

$$\gamma_i \sim N(\mathbf{0}, \mathbf{Q})$$

and independent from covariates. Observe: we still assume normality for the random effects!

In addition, the mean $\mu_{ij} = \text{E}(Y_{ij} \mid \gamma_i)$ is now to be seen to be conditional on the random effect γ_i. This means that the specified response distribution is conditional on the random effect: $f(y_{ij} \mid \gamma_i)$.

GLMM model assumptions

Distributional assumptions

(previously "1. Random component from exponential family")

- Given the random effects γ_i and the covariates $\mathbf{x}_{ij}$ and $\mathbf{u}_{ij}$
- the responses Y_{ij} are *conditionally independent* and
- the *conditional distribution* $f(y_{ij} \mid \gamma_i)$ belongs to and exponential family.

Structural assumptions

(previously "2. Systematic component" and "3. Link function")

- The *conditional mean* $\mu_{ij} = \text{E}(Y_{ij} \mid \gamma_i)$ is linked to
- the *linear predictor* $\eta_{ij} = \mathbf{x}_{ij}^T \boldsymbol{\beta} + \mathbf{u}_{ij}\gamma_i$
- through the link function

$$\eta_{ij} = g(\mu_{ij})$$

- or equivalently through the response function

$$\mu_{ij} = h(\eta_{ij})$$

- where $h^{-1} = g$ (inverse functions).

Distributional assumptions for random effects

- The random effects γ_i, $i = 1, \ldots, m$ are independent and identically distributed

$$\gamma_i \sim N(\mathbf{0}, \mathbf{Q})$$

- where the covariance matrix $\mathbf{Q}$ is a $(q+1) \times (q+1)$ positive definite.
- An important special case is $\mathbf{Q} = diag(\tau_0^2, \tau_1^2, \ldots, \tau_q^2)$.

Q: Any questions? Notice that we still have normal distribution for the random effects!

Remark: for LMMs we used that $\varepsilon \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, but said that it was possible to let the error terms be correlated $\varepsilon \sim N(\mathbf{0}, \mathbf{R})$ where $\mathbf{R}$ could har non-zero off-diagonal elements. However, for GLMMs this is much more complicated.

Alternative two-step formulation

Consider a random variable Y_{ij} and a random effect γ_i for cluster i.

Alternative two step formulation:

1. $\gamma_i \sim N(\mathbf{0}, \mathbf{Q})$
2. Given the realization of γ_i, then the Y_{ij}s are independent and each have the distribution $f(y_{ij} \mid \gamma_i)$ with mean $\mu_{ij} = h(\eta_{ij})$ and linear predictor $\eta_{ij} = \mathbf{x}_{ij}^T\boldsymbol{\beta} + \mathbf{u}_{ij}^T\gamma_i$.

Remark: this conditional formulation is "easy" to write down, but to do inference we need the marginal distribution of all Y_{ij}s together to construct the likelihood. We will soon see that this is hard.

Q: How would you proceed to generate a GLMM data set - e.g. based on the beaches example?

A: see class notes

GLMM random intercept model

Distributional assumptions: we will focus on binomial and Poisson responses.

Structural assumptions: The random effect is added to the linear predictor

$$\eta_{ij} = \mathbf{x}_{ij}^T\boldsymbol{\beta} + \gamma_{0i}$$

and we only consider canonical links, so logit for binomial and log for Poisson.

Distributional assumptions for random effects

The γ_{0i}, $i = 1, \ldots, m$ are independent and identically distributed

$$\gamma_{0i} \sim N(0, \tau_0^2)$$

Poisson random intercept model

Distributional assumptions: The conditional distribution of the response Y_{ij} is Poisson with mean λ_{ij}

$$f(y_{ij} \mid \gamma_i) = \frac{\lambda_{ij}^{y_{ij}}}{y_{ij}!} \exp(-\lambda_{ij}) \text{ for } y = 0, 1, 2, \ldots$$

This is conditional on λ_{ij}, which means that it is really conditional on the linear predictor - since the mean is a function of the linear predictor, and in the linear predictor we have fixed and random effects. So, conditional on the values for the fixed and random effects.

Then the observations Y_{ij} are *conditionally* independent for all i and j - but, not *marginally* independent.

Structural assumptions:

$$\eta_{ij} = \mathbf{x}_{ij}^T\boldsymbol{\beta} + \gamma_{0i}$$

$$\ln(\lambda_{ij}) = \eta_{ij} \text{ or } \lambda_{ij} = \exp(\eta_{ij})$$

Distributional assumptions for random effects

The γ_{0i}, $i = 1, \ldots, m$ are independent and identically distributed

$$\gamma_{0i} \sim N(0, \tau_0^2)$$

Remark: It is not necessary that $n_i > 1$ to use this model, and this model can be used to take care of overdispersion (when all $n_i = 1$).

Beach-example: Poisson GLMM with beach as random intercept

(we will talk about parameter estimation soon)

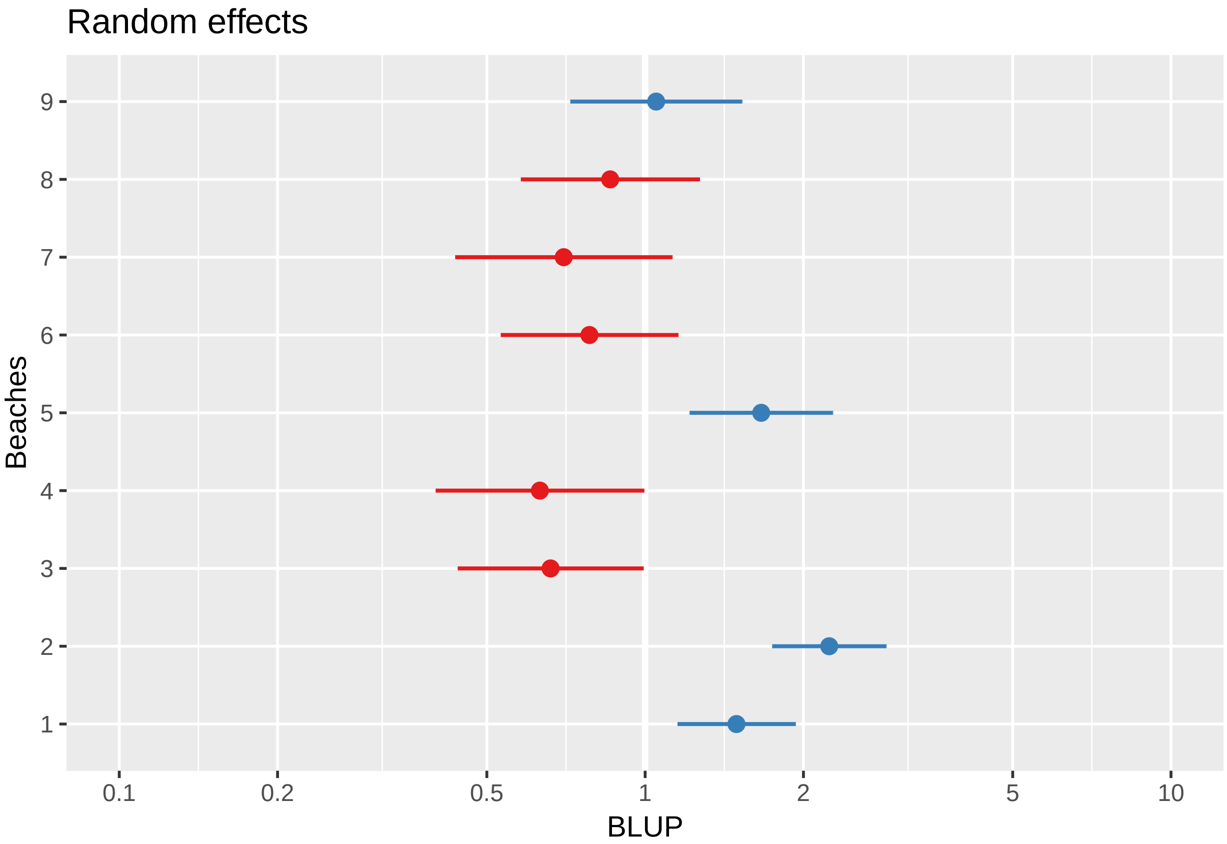
```r
library("AED")
data(RIKZ)
library(lme4)
fitRI = glmer(Richness ~ NAP + (1 | Beach), data = RIKZ, family = poisson(link = log))
summary(fitRI)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Richness ~ NAP + (1 | Beach)
##    Data: RIKZ
##
##      AIC      BIC   logLik deviance df.resid
##    220.8    226.2   -107.4    214.8       42
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.9648 -0.6155 -0.2243  0.2236  3.1869
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  Beach  (Intercept) 0.2249   0.4743
## Number of obs: 45, groups:  Beach, 9
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.66233    0.17373   9.569  < 2e-16 ***
## NAP         -0.50389    0.07535  -6.687 2.28e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## NAP 0.013
```

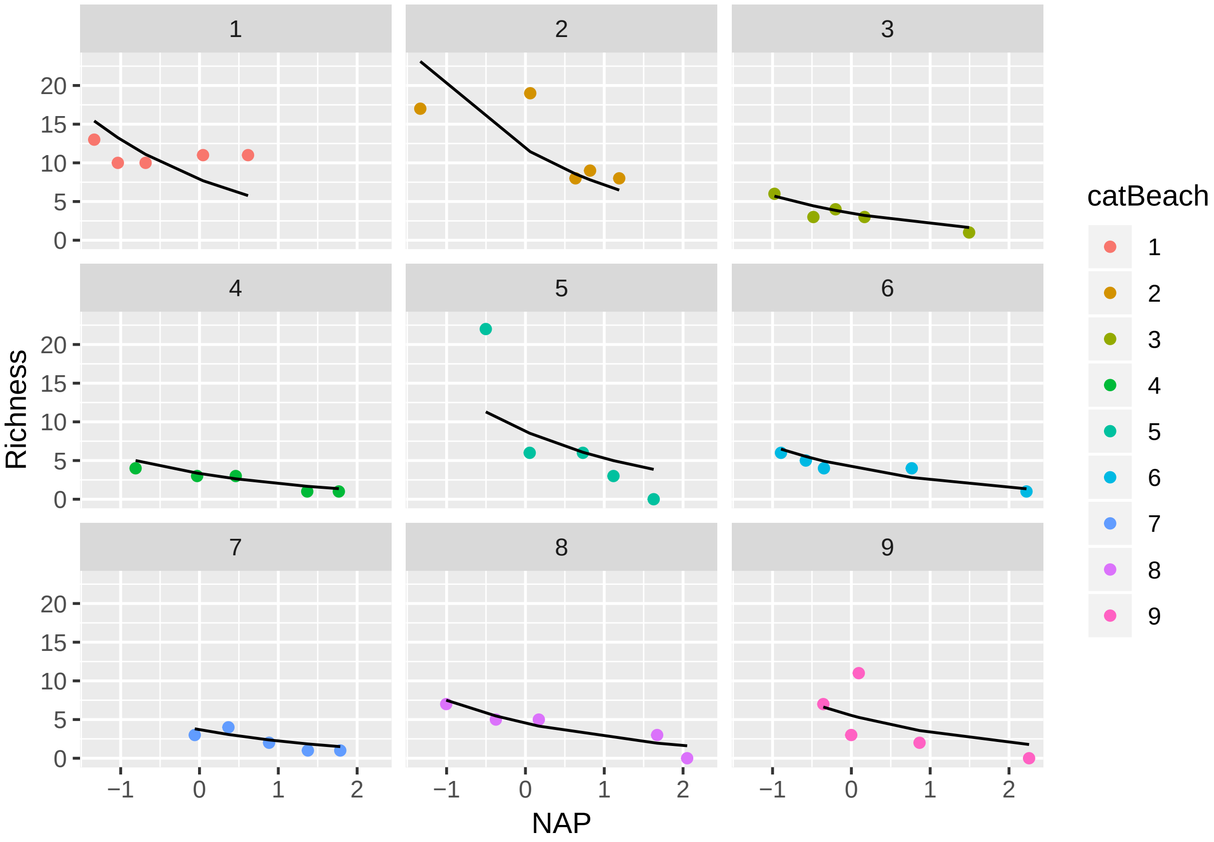
Q: Comment on what you see.

A: `glmer`, “Laplace approximation”, we now have p-values, and for the fixed effects we read z and not t. No estimate for σ^2 (or `residual` term in random effects part) - since the model does not contain any dispersion parameter.

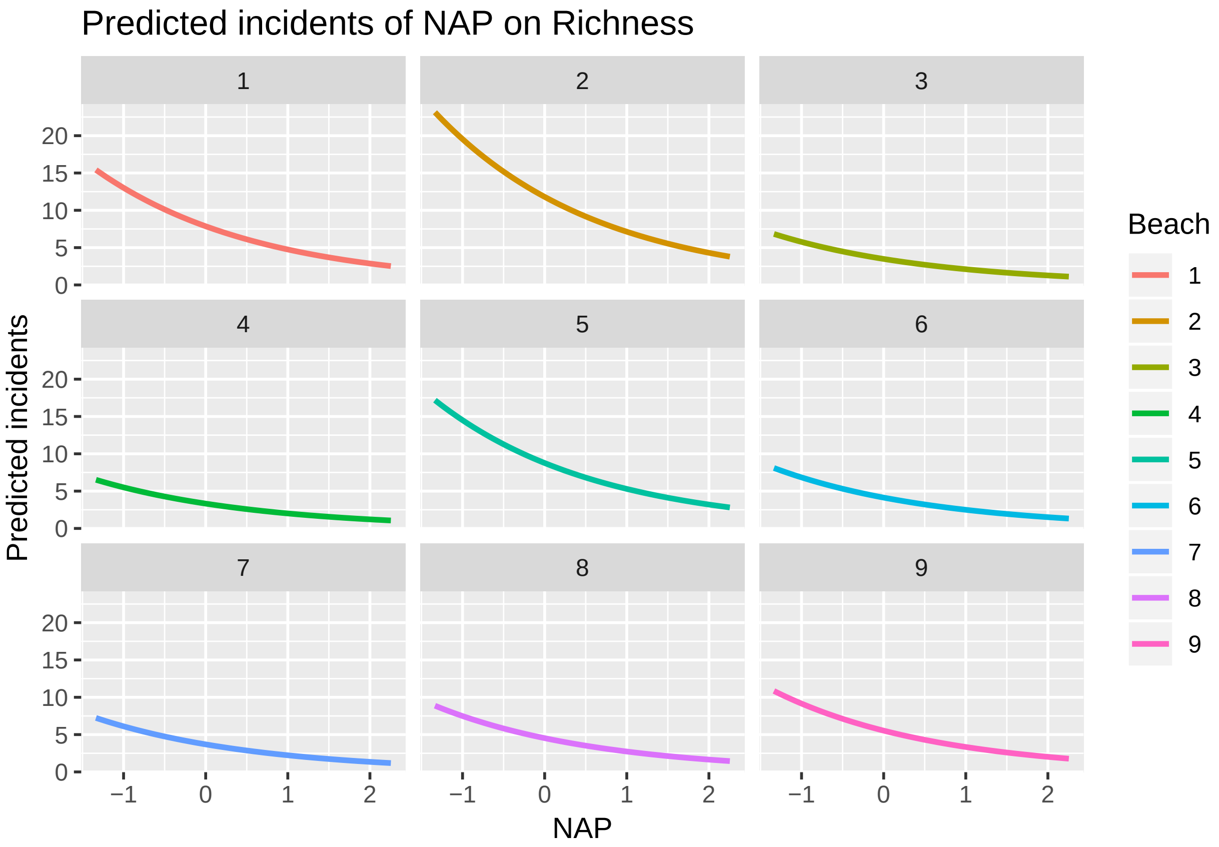
```r
library(sjPlot)
library(ggplot2)

plot_model(fitRI, type = "re") + ylab("BLUP") + xlab("Beaches")
```



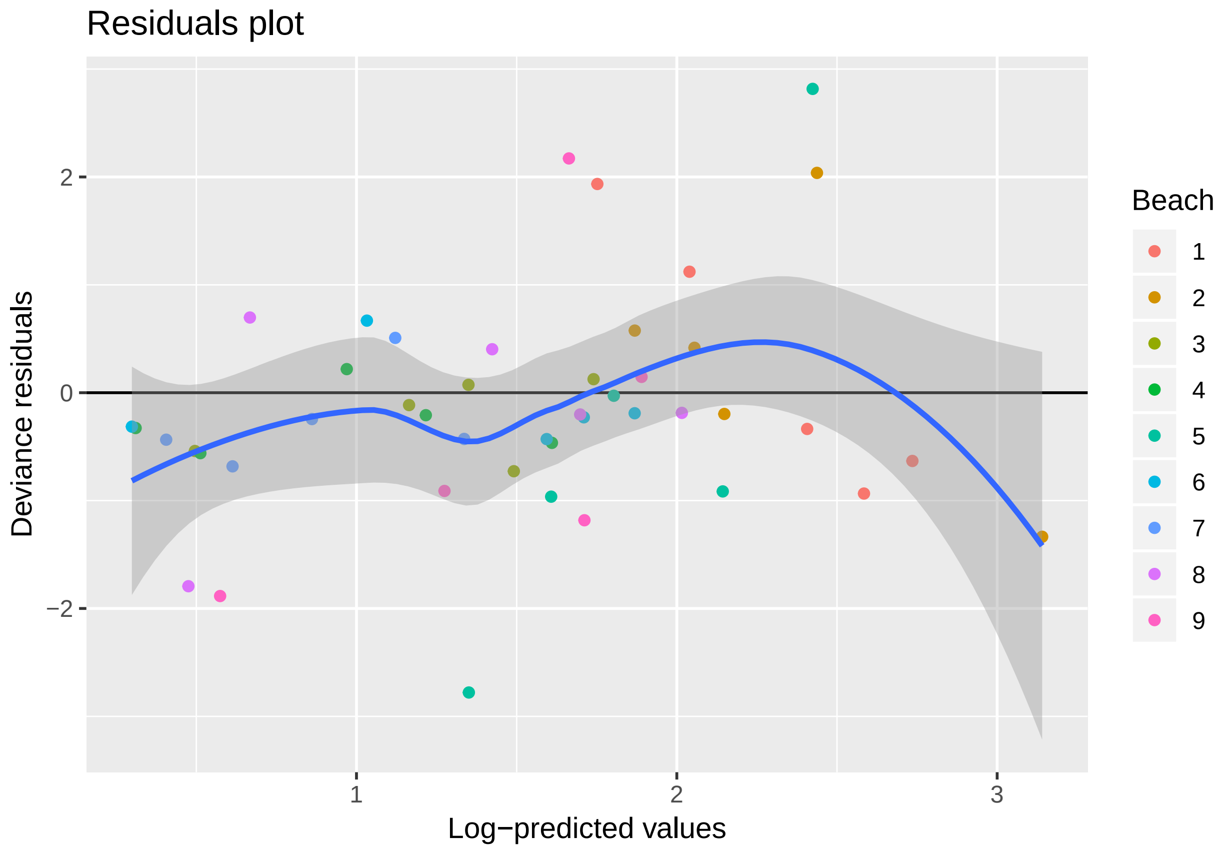
```r
df <- data.frame(Y = RIKZ$Richness, x = RIKZ$NAP, fitted = fitted(fitRI), dres = residuals(fitRI,
    scale = TRUE, type = "deviance"), catBeach = as.factor(RIKZ$Beach))
ggplot(df) + geom_point(aes(x = x, y = Y, col = catBeach)) + geom_line(aes(x = x,
    y = fitted)) + xlab("NAP") + ylab("Richness") + facet_wrap(~catBeach)
```



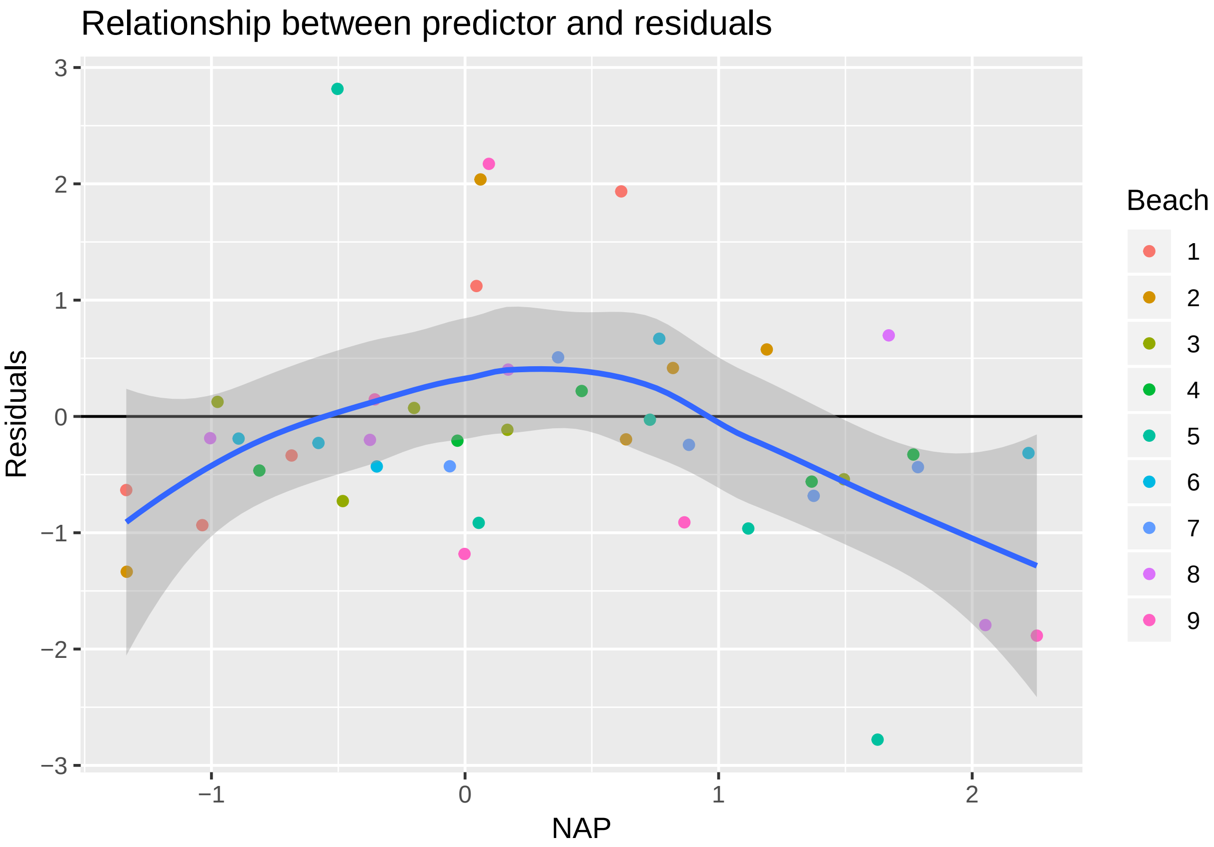
```r
xx <- rep(seq(min(RIKZ$NAP), max(RIKZ$NAP), length.out = 100), each = 9)
df <- data.frame(x = xx, eta = coef(fitRI)$Beach[, 1] + coef(fitRI)$Beach[,
    2] * xx, Beach = factor(rep(1:9, times = 100)))
ggplot(df, aes(x = x, y = exp(eta), col = Beach)) + geom_line(size = 1) + labs(x = "NAP",
    y = "Predicted incidents", title = "Predicted incidents of NAP on Richness") +
    facet_wrap(~Beach)
```



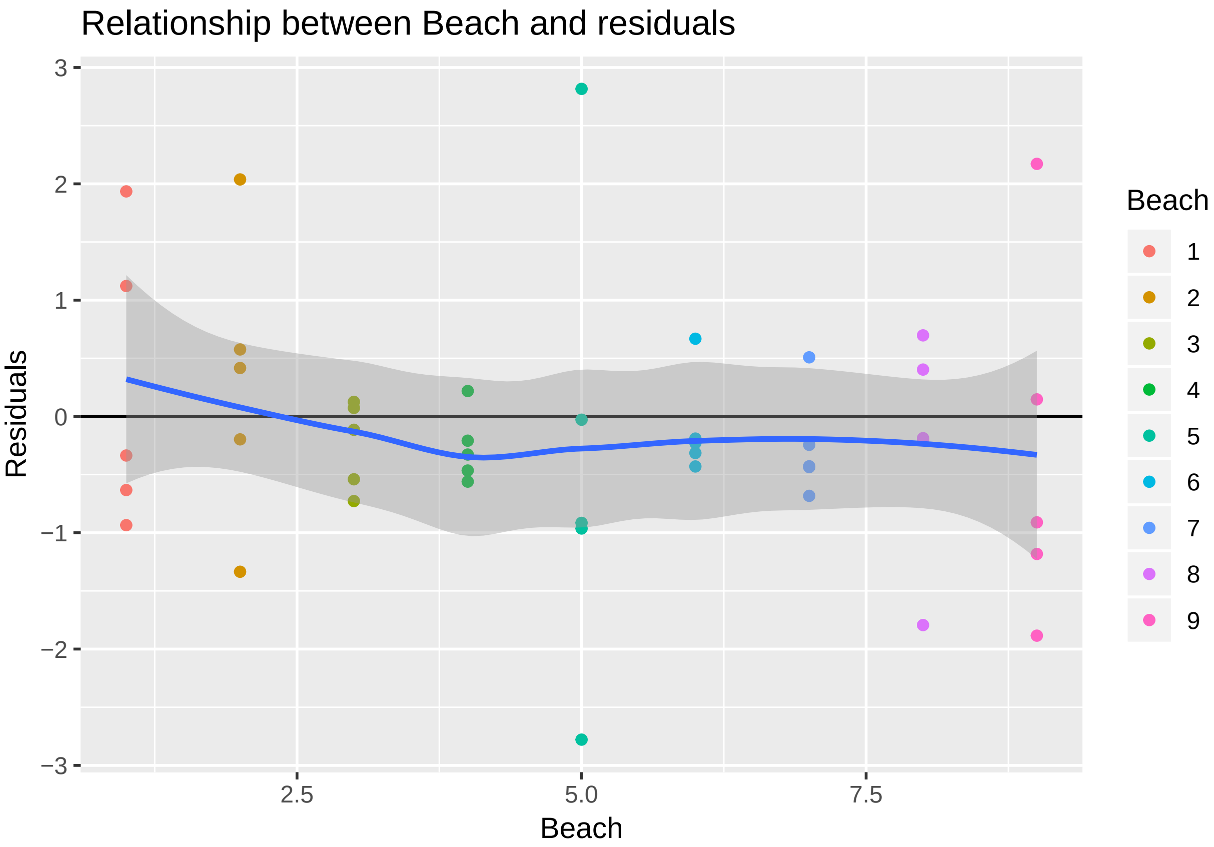
```
df <- data.frame(fitted = fitted(fitRI), resid = residuals(fitRI, scaled = TRUE,
    type = "deviance"), Beach = factor(RIKZ$Beach))
ggplot(df, aes(x = log(fitted), y = resid)) + geom_hline(yintercept = 0) + geom_point(aes(col = Beach)) +
    geom_smooth(method = "loess") + labs(x = "Log-predicted values", y = "Deviance residuals",
    title = "Residuals plot")
```



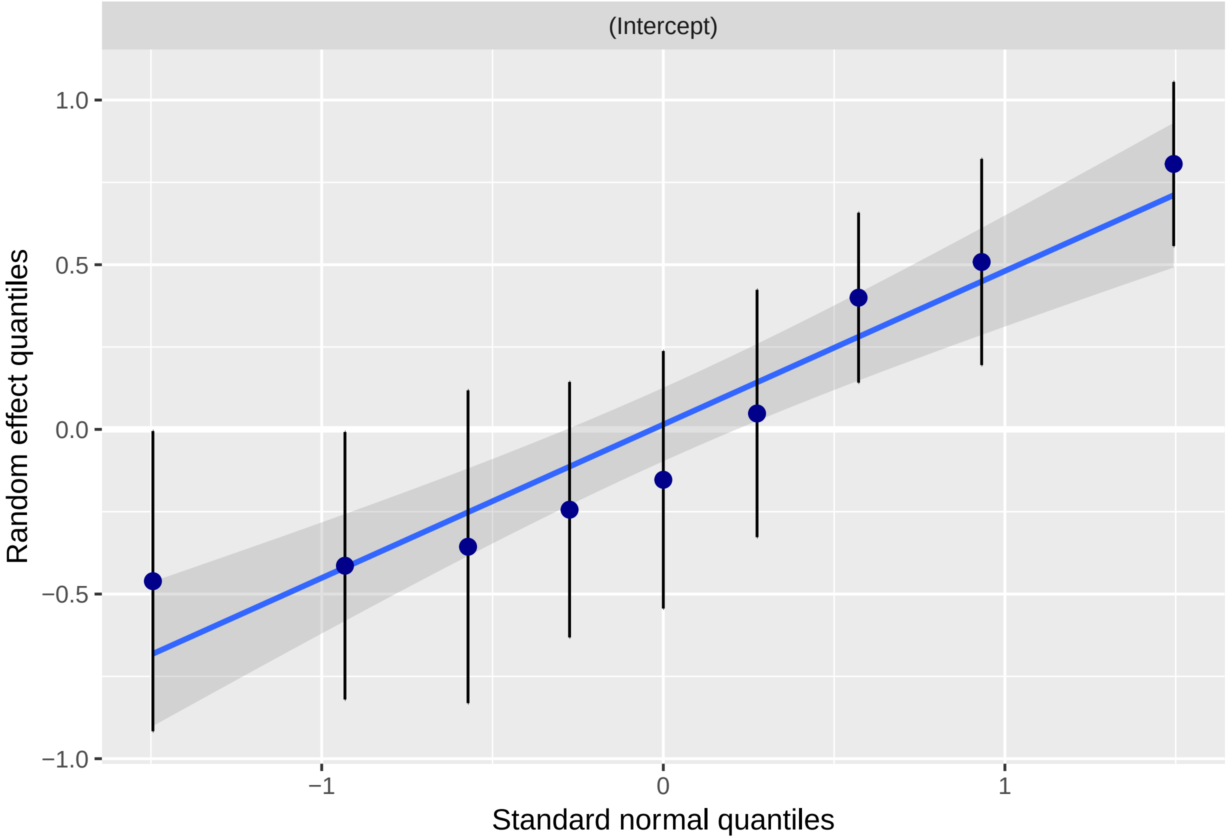
```r
df <- data.frame(NAP = RIKZ$NAP, resid = residuals(fitRI, scaled = TRUE, type = "deviance"),
    Beach = factor(RIKZ$Beach))
ggplot(df, aes(x = NAP, y = resid)) + geom_hline(yintercept = 0) + geom_point(aes(col = Beach)) +
    geom_smooth(method = "loess") + labs(x = "NAP", y = "Residuals", title = "Relationship between predictor and residuals")
```



```
df <- data.frame(Beach2 = RIKZ$Beach, resid = residuals(fitRI, scaled = TRUE,
    type = "deviance"), Beach = factor(RIKZ$Beach))
ggplot(df, aes(x = Beach2, y = resid)) + geom_hline(yintercept = 0) + geom_point(aes(col = Beach)) +
    geom_smooth(method = "loess") + labs(x = "Beach", y = "Residuals", title = "Relationship between Beach and residuals")
```



```
plot_model(fitRI, type = "diag")$Beach
```



Simulation example

Aim: to show the difference between random intercept and random slope log Poisson models, and to compare the cluster curves, mean of cluster curves and the marginal mean

This example a slightly modified version (added line and changed to ggplot) of an example in the Lecture notes by Magne Aldrin, Norwegian Computing Centre (which is way the notation differs slightly from what we have used).

The model we use is

$$\begin{pmatrix}\gamma_{0i}\\ \gamma_{1i}\end{pmatrix} \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.2^2 & 0\\ 0 & 0.07^2\end{pmatrix}\right)$$

and then

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0i}$$

for the random intercept model and

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0i} + \gamma_{1i} x_{ij}$$

for the random slope model.

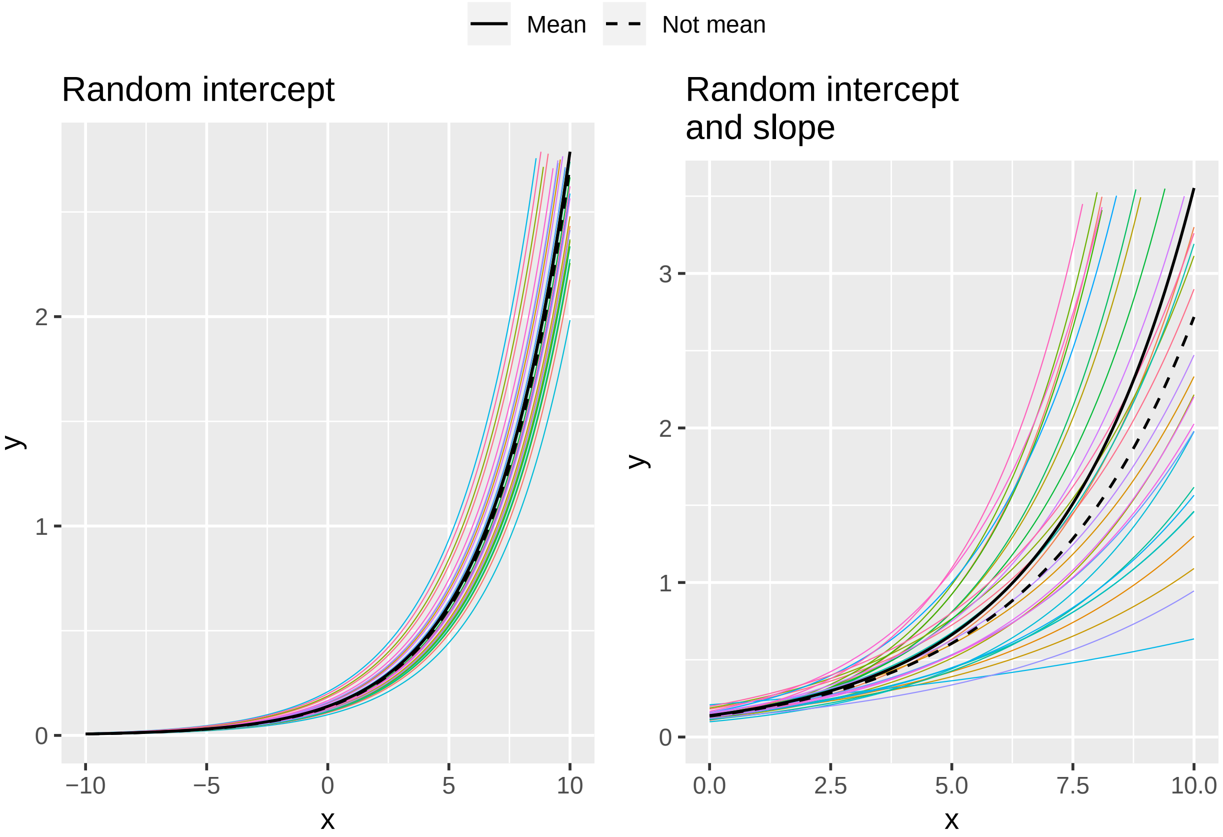
Values chosen are $\beta_0 = -2$ and $\beta_1 = 0.3$.

Log-link is used so $\eta_{ij} = \ln(\mu_{ij})$.

Only data for the random intercept and slope are simulated, while parameter estimates are considered known. The number of clusters is $m = 1000$ and $n_i = 201$ value of x from -10 to 10 are studied.

Below 30 random lines are plotted together (in different colours). The black dashed line is $\exp(\beta_0 + \beta_i x)$ (denoted “not mean” in the plot) and the black solid line is the population mean (really, the average of the m curves).

(Q: what would happen if we simulate data and estimate parameters instead of only plotting means?)



Q: comment on what you see. Would you expect that the solid and dashed curve were identical?

Binomial random intercept model

We will study this model in the interactive session.

Simulation example

Aim: to show the difference between random intercept and random slope logit models, and to compare the mean cluster curve with the population curve (as for the Poisson model).

This example a slightly modified version (added line and changed to ggplot) of an example in the Lecture notes by Magne Aldrin, Norwegian Computing Centre (which is way the notation differs slightly from what we have used).

The model we use is

$$\begin{pmatrix}\gamma_{0i}\\ \gamma_{1i}\end{pmatrix} \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}4 & 0\\ 0 & 1\end{pmatrix}\right)$$

and then

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0i}$$

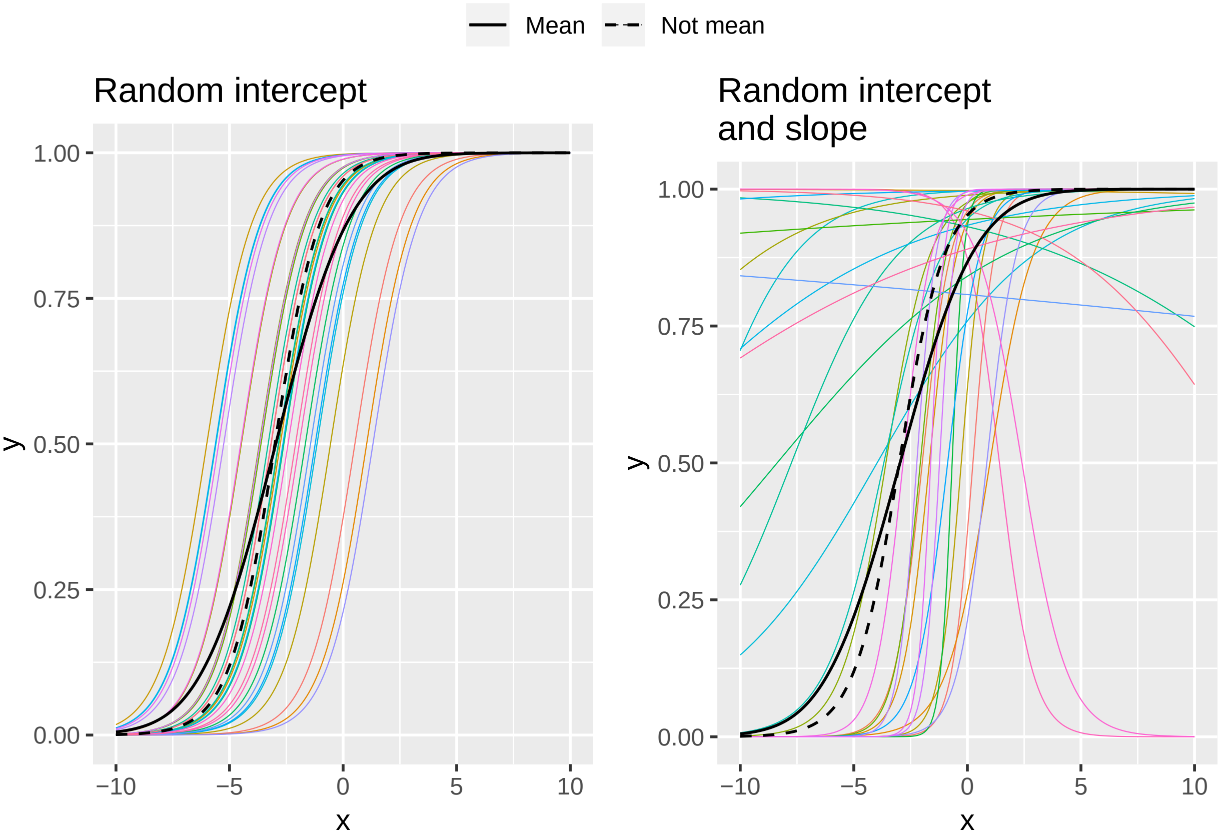
for the random intercept model and

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0i} + \gamma_{1i} x_{ij}$$

for the random slope model where $\beta_0 = 3$ and $\beta_1 = 1$. Logit-link is used so $\eta_{ij} = \ln(\frac{\pi_{ij}}{1-\pi_{ij}})$.

Only data for the random intercept and slope are simulated, while parameter estimates are considered known. The number of clusters is $m = 1000$ and $n_i = 201$ value of x from -10 to 10 are studied.

Below 30 random lines are plotted together (in different colours). The black dashed line is $\exp(\beta_0 + \beta_i x)/(1 + \exp(\beta_0 + \beta_i x))$ (denoted "not mean" in the plot) and the black solid line is the population mean (really, the average of the m curves).



Q: comment on what you see.

Conditional and marginal model

To do parameter estimation in the LMM we started with the marginal model for Y_{ij}.

$$f(y_{ij}) = \int_{\gamma_i} f(y_{ij} \mid \gamma_i) f(\gamma_i) d\gamma_i$$

where

$$f(\gamma_i) \text{ is } N(\mathbf{0}, \mathbf{Q})$$

and $f(y_{ij} \mid \gamma_i)$ might be the binomial, Poisson, normal, gamma, … distribution - chosen as an exponential family.

It turns out that this integral can only be written out analytically in special cases.

In the special case that $f(y_{ij} \mid \gamma_i)$ is normal we saw in Module LMM that the marginal distribution was also multivariate normal.

Another special case is the log-linear Poisson random intercept where all $n_i = 1$. See pages 393-394 in our textbook to see the solution (that can be used to handle overdispersion in Poisson regression).

Remark: for the random intercept Poisson GLMM with log-link the expected marginal mean is (Agresti, 2015, page 309)

$$\text{E}(Y_{ij}) = \text{E}[\text{E}(Y_{ij} \mid \gamma_{0i})] = \exp(\mathbf{x}_{ij}^T \boldsymbol{\beta} + \tau_0^2/2)$$

The solid line in the plot to the left above gives this curve, while the dashed is $\exp(\mathbf{x}_{ij}^T \boldsymbol{\beta})$. In Agresti (2015, page 310) the variance of the marginal distribution is given as

$$\text{Var}(Y_{ij}) = \cdots = \text{E}(Y_{ij}) + [\text{E}(Y_{ij})^2 (\exp(\tau_0^2) - 1)]$$

so we see that the variance is larger than that of a Poisson distribution.

Remark: the conditional mean is $\exp(\mathbf{x}_{ij}^T \boldsymbol{\beta} + \gamma_{0i})$, while the marginal mean is $\exp(\mathbf{x}_{ij}^T \boldsymbol{\beta} + \tau_0^2/2)$.

(If the link function chosen had been the identity link then the marginal mean would have been $\mathbf{x}_{ij}^T \boldsymbol{\beta}$.)

Parameter estimation

The most popular method for parameter estimation for regression is maximum likelihood. The likelihood of the data is given by the marginal distribution of all Y_{ij}s jointly. However, this can not be found analytically (as we saw above), so we must resort to numerial methods to evaluate the likelihood of a general GLMM.

The parameters we want to estimate are the fixed effects $\boldsymbol{\beta}$s, and the parameters in $\mathbf{Q}$ for the random effects - denoted by ϑ.

The contribution from observations in the ith cluster is

$$f(\mathbf{y}_i \mid \boldsymbol{\beta}, \vartheta) = \int_{\boldsymbol{\gamma}_i} f(\mathbf{y}_i \mid \boldsymbol{\gamma}_i, \boldsymbol{\beta}) f(\boldsymbol{\gamma}_i \mid \mathbf{Q}) d\boldsymbol{\gamma}_i = \int_{\boldsymbol{\gamma}_i} \prod_{j=1}^{n_i} f(y_{ij} \mid \boldsymbol{\gamma}_i, \boldsymbol{\beta}) f(\boldsymbol{\gamma}_i \mid \mathbf{Q}) d\boldsymbol{\gamma}_i$$

And the likelihood is then (since the clusters are independent)

$$L(\boldsymbol{\beta}, \vartheta) = \prod_{i=1}^{m} f(\mathbf{y_i} \mid \boldsymbol{\beta}, \vartheta)$$

To arrive at parameter estimates we need to maximize the likelihood with respect to $\boldsymbol{\beta}$ and ϑ, which is a complicated numerical problem.

There are several solution to this, and the `glmer` function in `lme4` uses the Laplace approximation. Other solutions are adaptive Gaussian quadratures, penalized quasilikelihood and various Bayesian methods (study the posterior distribution of the parameters). This is an active area of research.

Gauss-Hermite Quadrature

If the dimension of $\boldsymbol{\gamma}_i$ is small, for example our random intercept model, then the Gauss-Hermite quadrature may be used to approximate the integral above. This will be of the form

$$\int_{-\infty}^{\infty} h(\gamma)\exp(-\gamma^2)d\gamma$$

which can be written as a sum of weights and quadrature points that are roots of Hermite polynomials. This can then be maximized using Newton-Raphson methods.

When the dimension of $\boldsymbol{\gamma}$ gets more than two, other methods are used.

Laplace approximation (for the interested student)

The notation in this section is taken from Kristensen et al (2016): "TMB: Automatic Differentiation and Laplace Approximation", *Journal of Statistical Software*, 70, 5. doi:10.18637/jss.v070.i05, which is the background for the `glmmTMB` package. The notation is different from what is used in this module, but this reflects that often different notation is used for the same models.

Let $f(u,\theta)$ be the *joint log-likelihood of the data and the random effects*. This will in our case be related to $\ln[f(\mathbf{y}_i \mid \gamma_i, \boldsymbol{\beta})f(\gamma_i \mid \mathbf{Q})]$.

Observe that $f(u,\theta)$ depends on the unknown random effects u and the parameters in θ (here we have the fixed effects and the parameters in $\mathbf{Q}$ for the random effects).

The maximum likelihood estimate for θ maximized

$$L(\theta) = \int_u \exp(f(u,\theta))du$$

with respect to θ. Observe that the random effects u are now integrated out and the marginal likelihood $L(\theta)$ is the likelihood of the data as a function of the parameters only.

Let $\hat{u}(\theta)$ be the minimizer of $f(u,\theta)$, and $H(\theta)$ is the Hessian (matrix of partial second derivatives) of $f(u,\theta)$ evaluated at $\hat{u}(\theta)$.

The Laplace approximation for the marginal likelihood $L(\theta)$ is

$$L^*(\theta) = (2\pi)^{n/2}\det(H(\theta))^{-1/2}\exp(f(\hat{u},\theta))$$

Some more details: Let $g(u) = \ln(f(y \mid u)f(u))$, and look at the second order Taylor expansion around $\hat{u}(\theta)$:

$$g(u) \approx \tilde{g}(u) = g(\hat{u}) + (u-\hat{u})g`(\hat{u}) + \frac{1}{2}(u-\hat{u})^2 g''(\hat{u}) = g(\hat{u}) - \frac{1}{2}(u-\hat{u})^2(-g''(\hat{u}))$$

This means that $\exp(\tilde{g}(u))$ is proportional to the normal density with mean $\mu = \hat{u}$ and variance $\sigma^2 = -1/g''(\hat{u})$. Putting this back into our integral

$$L(\theta) = \int_u \exp(g(u))du \approx \int_u \exp(\tilde{g}(u))du = \exp(g(\hat{u})) \int_u \exp(-\frac{1}{2\sigma^2}(u-\mu)^2)du = \exp(g(\hat{u})\sqrt{2\pi\sigma^2}$$

This last part was taken from http://people.math.aau.dk/~rw/Undervisning/Topics/Handouts/6.hand.pdf.

What have we not covered?

- We have skipped a lot of technicalities about the parameter estimation.
- How to predict the random effects γ_i: "the conditional modes" of the random effects.
- What is fitted values, and how are residuals calculated.
- How to test hypotheses?
- AIC to compare models (very similar to LMM).
- And, surely, much more.

Summing up - what have we learned about the GLMM?

- The GLMM can be formulated using three ingredients:
  - distributional assumption: $f(y_{ij} \mid \gamma_i)$ from exponential family
  - structural assumptions: linear predictor $\eta_{ij} = \mathbf{x}_{ij}^T \boldsymbol{\beta} + \mathbf{u}_{ij}\gamma_i$ and link function $\eta_{ij} = g(\mu_{ij})$ where $\mu_{ij} = \text{E}(Y_{ij} \mid \gamma_i)$.
  - distributional assumptions for the random effects: $\gamma_i \sim N(\mathbf{0}, \mathbf{Q})$.
- The GLMM likelihood function is expressed as an integral with respect to the random effects and does (in general) not have a closed form solution.
- Numerical approximation methods need to be used to find parameter estimates, and one possibility is to use the Laplace approximation, but many competing method exists.

- Three `R` packages that can be investigated is `lme4`(function `glmer`) and `glmmTMB` (template model builder), and the NTNU-flagship `inla`. How to use these three packages on a simulated data set (binary data, logit link, random intercept and slope) is shown in the end of the module page (NOT on our reading list but for the interested student).

Interactive session week

This is the last interactive session of the course.

- First hour: Problem 1
- 15 min before end first hour (or if you finish earlier): answer the following questions in your group (togehter) and fill in this Google form
- What was the top 5 (or so) concepts in this course that you feel that you need to work more to understand? (input to summing-up last lecture)
- You are here today - you are the students that come to the IL, even the last one. Why do you attend the IL? What is good about the IL? What can be challenging?
- If you want: here is the draft for the summing-up (module 6 on categorical missing and some QC needed) - any comments?

- Second hour: continue the work above, or work on (and get help) with Compulsory 3!

Problem 1: Taken from UiO, STK3100, 2011, problem 2

a) Show that the binomial distribution is a member of the exponential family of distributions. What do we mean with canonical link, and which advantages do we get when using canonical link? What is the canonical link for the binomial distibution?

Remark: this has been done several times before. Instead of doing the maths on the board, focus on the concepts. Also discuss this relationship: $\theta_i \leftrightarrow \mu_i \leftrightarrow \eta_i \leftrightarrow \beta$ and fill in what is over the arrows, and the missing arrow for the canonical link.

b) *This excercise is modified!*

Assume the following model:

$$y_{ij} \sim \text{Bin}(1, \pi_{ij})$$

$$g(\pi_{ij}) = \beta_0 + \beta_1 x_{1,ij} + \beta_2 x_{2,i}$$

where $g(\cdot)$ is a suitable link function and all y_{ij} are independent. What kind of model is this?
The table below shows AIC-values for three different link functions often used with binary response data:

Link function	AIC
probit	57.51571
cloglog	58.30338
logit	57.69237

Explain what AIC is, and argue why it is reasonable to use such criteria in this situation (instead of other tests from this course). Based on these three values, which link-function do you prefer? Why?

Remark: what is the take home message here? Hint: nested models?

We will, as we did a few weeks back, look at a fish dataset consisting of observations of cod from the year 2000, but we are now interested in the age of each fish, rather than the weight. The dataset is from Havforskningsinstituttet in Bergen, and we use only a subset of a large dataset with information about fish in Barentshavet. The following variables are available on a random sample within each catch (a draft of trawl (trål)):

- `length`: the length of fish (in cm)
- `weight`: the weight of fish (in grams)
- `age`: the age of fish (in years)
- `haulsize`: the size of the total catch (in ton (1000 kg))

Let i be the index for catch, and j be the index for an individual fish in a given catch.

Age is a categorical variable between 2 and 12, but we create a new variable A_{ij}:

$$A_{ij} = \begin{cases} 1 \text{ if age}_{ij} > 6 \\ 0 \text{ else} \end{cases}$$

which is a binary variable, and we use this for the response.

Remark: before we used weight as response (normal), but now we use dichotomized age as a binary response.

We look at the following model:

$$A_{ij}|\gamma_{0i} \sim \text{Bin}(1, \pi_{ij})$$
$$g(\pi_{ij}) = \beta_0 + \beta_1 \log(\texttt{length}_{ij}) + \beta_2 \log(\texttt{haulsize}_i) + \gamma_{0i}$$

$$\gamma_{0i} \sim N(0, \tau_0^2)$$

where $g(\cdot)$ is the logit link function, and all A_{ij} are conditional independent (given γ_i).

c1) *New:* What kind of model is this? What is γ_{0i}? Compare to the linear mixed effects model- similarities and differences.

Below you find an excerpt of this model:

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: A ~ log(length) + log(haulsize) + (1 | haul)
##    Data: fish
##
##      AIC      BIC   logLik deviance df.resid
##   2524.7   2551.2  -1258.4   2516.7     5483
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -20.037  -0.176  -0.040   0.138  40.513
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  haul   (Intercept) 1.921    1.386
## Number of obs: 5487, groups:  haul, 63
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -89.3202     3.0256 -29.522   <2e-16 ***
## log(length)    20.7783     0.7080  29.348   <2e-16 ***
## log(haulsize)  -0.2396     0.1400  -1.711    0.087 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) lg(ln)
## log(length) -0.997
## log(haulsz)  0.021  0.017
```

c2) *New:* Explain what it means that the model is fitted by maximum likelihood. What does Laplace approximation has to do with the model fit?

Skip d)- not in our reading list to test random effects - so too advanced. Instead discuss why this is difficult.

d) The log-likelihood value for the corresponding model without any random effects (which corresponds to a model with $\tau_0^2 = 0$) is -1498.04. Use this to perform a likelihood ratio test with $H_0 : \tau_0^2 = 0$. Calculate the corresponding p-value and draw a conclusion.

e) The Havforskningsinstituttet believes that the haulsize is important in the modelling of age and length of the fish (read: as a fixed effect). Based on the excerpts above (on dichotomized age) and below (on weight), what do you think about this?

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log(weight) ~ log(length) + log(haulsize) + (1 | haul)
##    Data: fish
##
## REML criterion at convergence: -10188.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1204 -0.6107 -0.0314  0.5813  5.1458
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  haul     (Intercept) 0.003030 0.05505
##  Residual             0.008594 0.09270
## Number of obs: 5433, groups:  haul, 63
##
## Fixed effects:
##                Estimate Std. Error  t value
## (Intercept)   -4.286029   0.041554 -103.143
## log(length)    2.901037   0.009703  298.984
## log(haulsize)  0.006169   0.005328    1.158
##
## Correlation of Fixed Effects:
##             (Intr) lg(ln)
## log(length) -0.980
## log(haulsz)  0.047  0.056
```

Remark: It is in general a very bad idea to do a dichotomization of a continuous variable - like what is done here for transforming the age into A_{ij}, because then information is lost. In a hypothesis testing set-up this will in general give substantial loss of power to detect the effect of a covariate.

Software for GLMM: demonstration of `glmer`, `glmmTMB` and `inla` for analysing GLMMs

This is **NOT** on the reading list, but an extra service to the interested students.

```
install.packages("arm")
install.packages("reshape2")
install.packages("sp")
install.packages("glmmTBM")
install.packages("INLA", repos="https://inla.r-inla-download.org/R/stable", dep=TRUE)
}
```

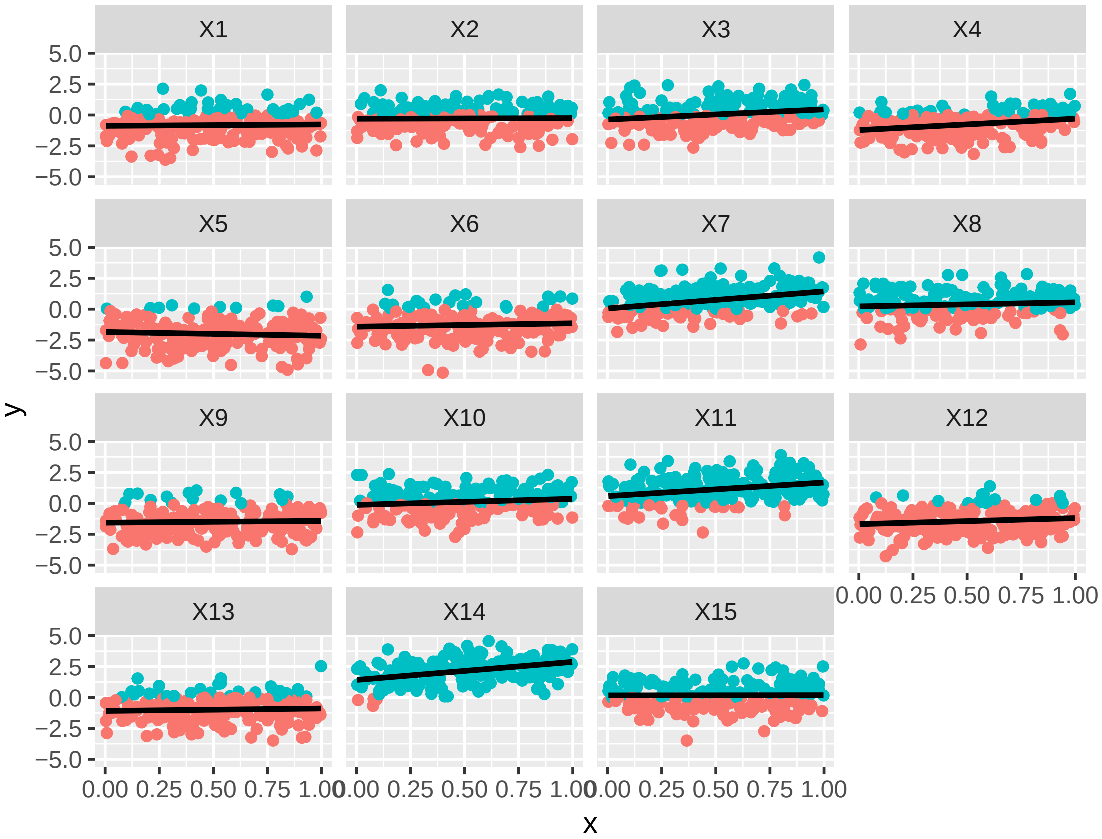
```
library(ggplot2)
library(ggpubr)
library(lme4)
```

```r
library(glmmTMB)
library(INLA)
library(reshape2)
library(arm)
```

For this demonstration, we will simulate a dataset with one fixed effect (x) and random intercept and random slope (always nice to know the true values for parameters to compare with estimates).

Q: What is the model that we simulate from?

```r
set.seed(90) # to get reproduciability

no.gr <- 15
x1 <- runif(200, 0, 1)

y <- matrix(NA, nrow = 200, ncol = 15)

Q <- matrix(c(0.5, 0.3, 0.3, 0.5), nrow = 2)
random_both <- mvrnorm(no.gr, c(0,0), Q)
random_int <- random_both[,1]; random_slope <- random_both[,2]

# to test out
#random_int <- runif(15, -2, 2) # when we want to simulate from a model that does not fit exactly
#random_slope <- runif(15, -2, 2)

beta_0 <- -0.45
beta_1 <- 0.76

for (i in 1:15){
  y[,i] <- beta_0 + beta_1*x1 + random_int[i] + random_slope[i]*x1 + rnorm(200, 0, 1)
}

y_binom <- as.numeric(c(y) > 0)

ggplot(cbind(melt(data.frame(y)), x = rep(x1, 15), y = as.factor(y_binom)),
       mapping = aes(x = x, y = value)) +
  geom_point(aes(col = y)) + geom_smooth(method = "lm", col = "black", se = FALSE) +
  facet_wrap(~ variable) + labs(x = "x", y = "y")
```



```r
mydata <- data.frame(y = y_binom, x = x1, group = rep(1:15, each = 200))

head(mydata)
```

```
##   y         x group
## 1 0 0.5307603     1
## 2 0 0.8850698     1
## 3 0 0.4399639     1
## 4 1 0.9422908     1
## 5 1 0.1883275     1
## 6 0 0.6555310     1
```

Use a binary regression with logit link to model the probability. That is, we have one covariate and make the following assumptions.

1. Distributional assumptions: $Y_{ij} \mid \gamma_i \sim \text{Bin}(n_i, \pi_{ij})$ for $i = 1, \ldots, 15,\ j = 1, \ldots, n_i = 200,\ N = \sum_i n_i = 3000$.

2. Structural assumptions: The linear predictor (with random effects) with one fixed covariate:

$$\eta_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_{0i} + \gamma_{1i} x_{ij}$$

   where we will choose $x_{ij} = u_{ij}$. The link function is:

$$\eta_{ij} = \ln\left(\frac{\pi_i}{1 - \pi_i}\right)$$

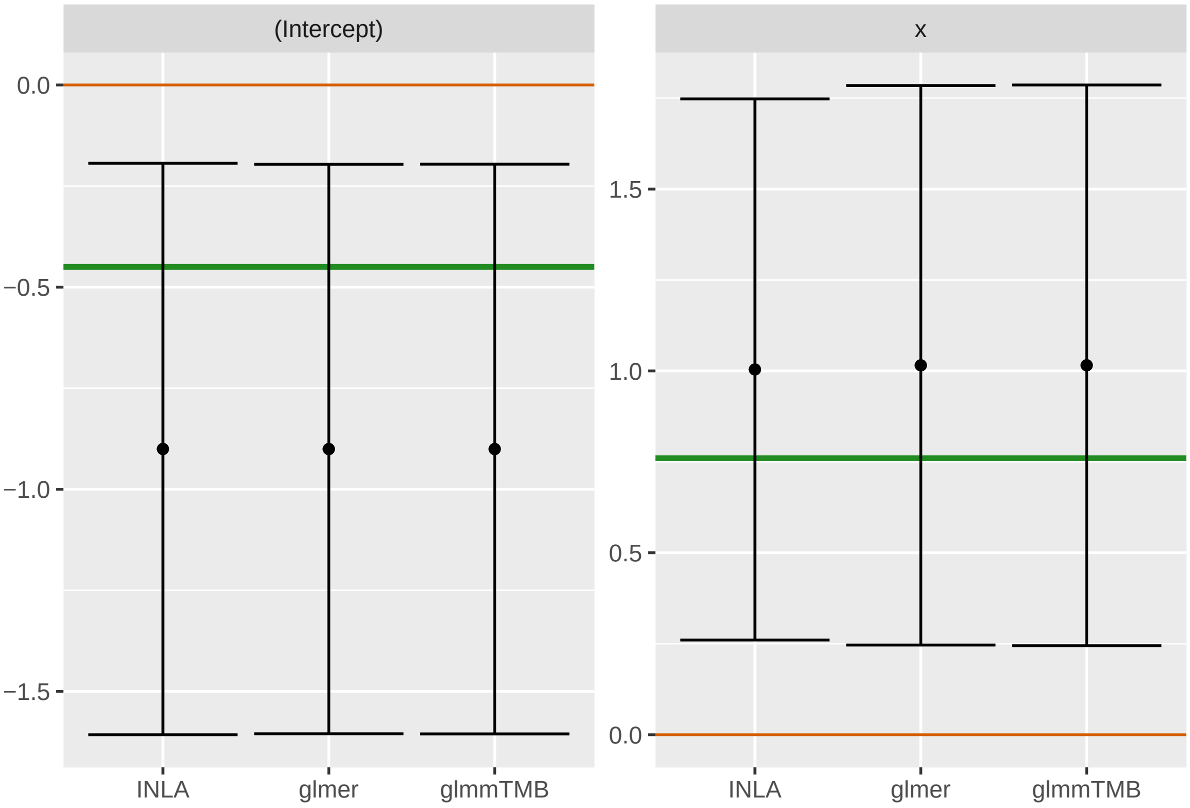
3. Distributional assumptions for random effects

$$\begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_0^2 & \tau_{01} \\ \tau_{01} & \tau_1^2 \end{pmatrix}\right)$$

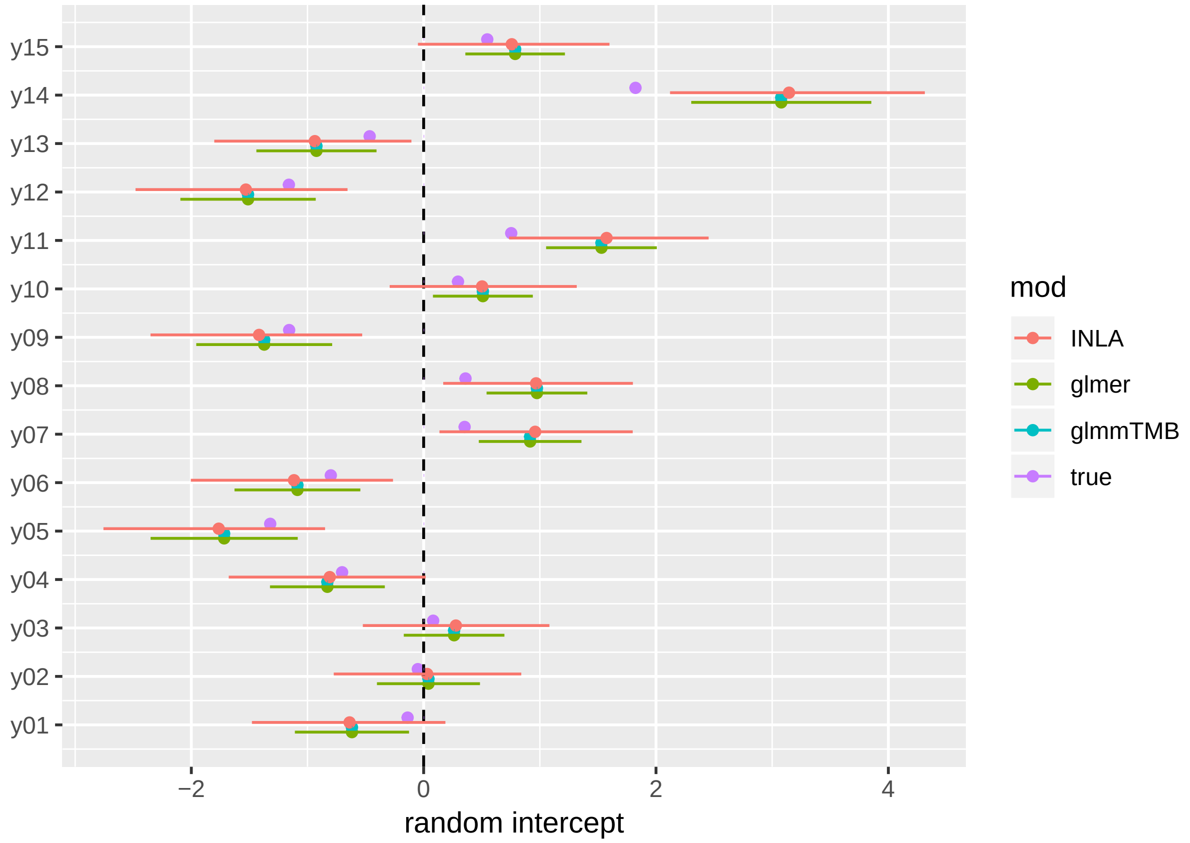
Now we will look at three common ways of analysing such models. For each type we will fit the model and look at results.

```
fit_glmer <- glmer(y ~ x + (1 + x | group), data = mydata, family = binomial(link = "logit"))

fit_glmmTMB <- glmmTMB(y ~ x + (1 + x | group), data = mydata, family = binomial(link = "logit"))

n.gr <- max(mydata$group)
mydata$group2 <- mydata$group + n.gr
fit_inla <- inla(y ~ x + f(group, model = "iid2d", n = 2*n.gr) + f(group2, x, copy = "group"),
                 data = mydata, family = "binomial",
                 control.compute = list(dic = TRUE),
                 control.family = list(link = "logit"))
```

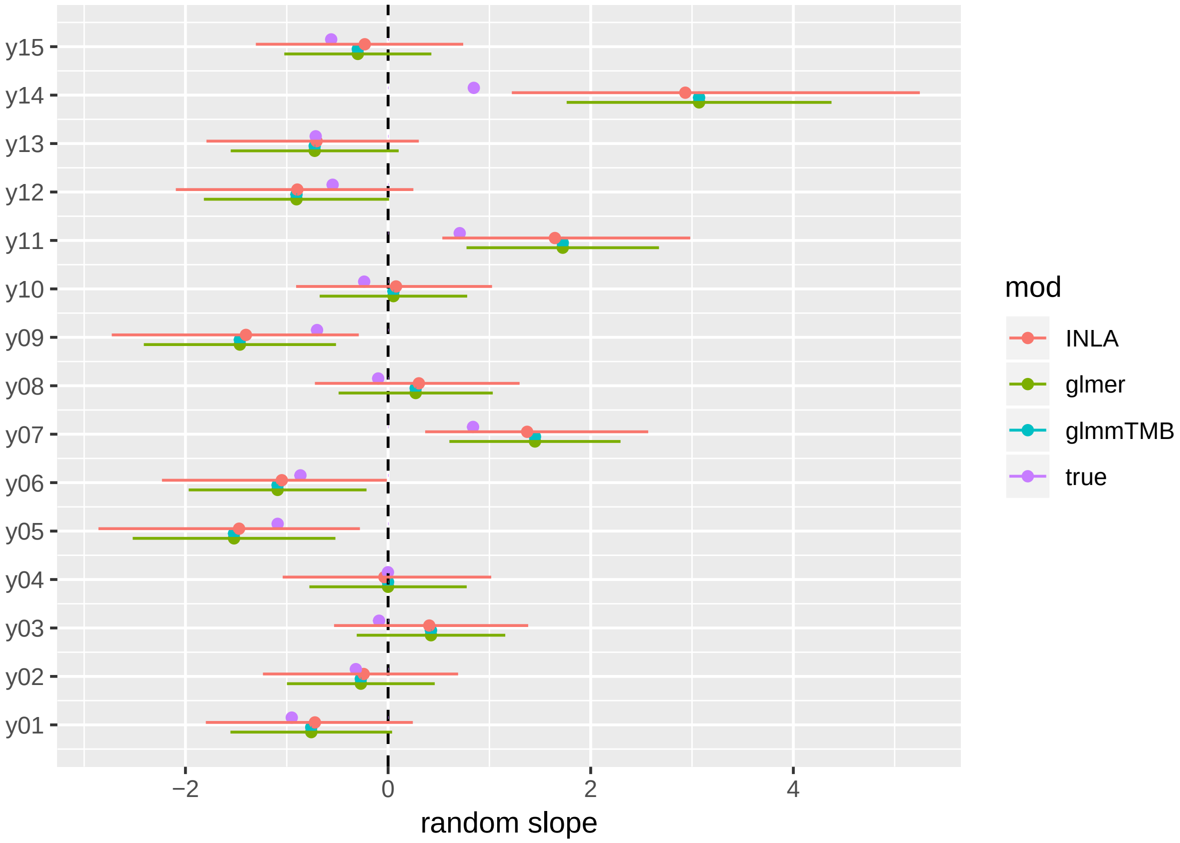
```
coefdf <- data.frame(mean =
                       c(summary(fit_glmer)$coefficients[,1],
                         summary(fit_glmmTMB)$coefficients$cond[,1],
                         fit_inla$summary.fixed[,1]),
                     sd =
                       c(summary(fit_glmer)$coefficients[,2],
                         summary(fit_glmmTMB)$coefficients$cond[,2],
                         fit_inla$summary.fixed[,2]),
                     mod = rep(c("glmer", "glmmTMB", "INLA"), each = length(fit_inla$names.fixed)),
                     par = rep(fit_inla$names.fixed, 3))

true_frame <- data.frame(par = fit_inla$names.fixed, beta = c(beta_0, beta_1))

critval <- qnorm(0.025, lower.tail = FALSE)
ggplot(coefdf) + geom_point(aes(x = mod, y = mean)) +
  geom_hline(aes(yintercept = 0), col = "#D55E00") +
  geom_hline(data = true_frame, aes(yintercept = beta), col = "forestgreen", size = 1) +
  geom_errorbar(aes(x = mod, ymin = mean-critval*sd, ymax = mean+critval*sd)) +
  facet_wrap(~ par, scales = "free_y") + labs (x = "", y = "")
```



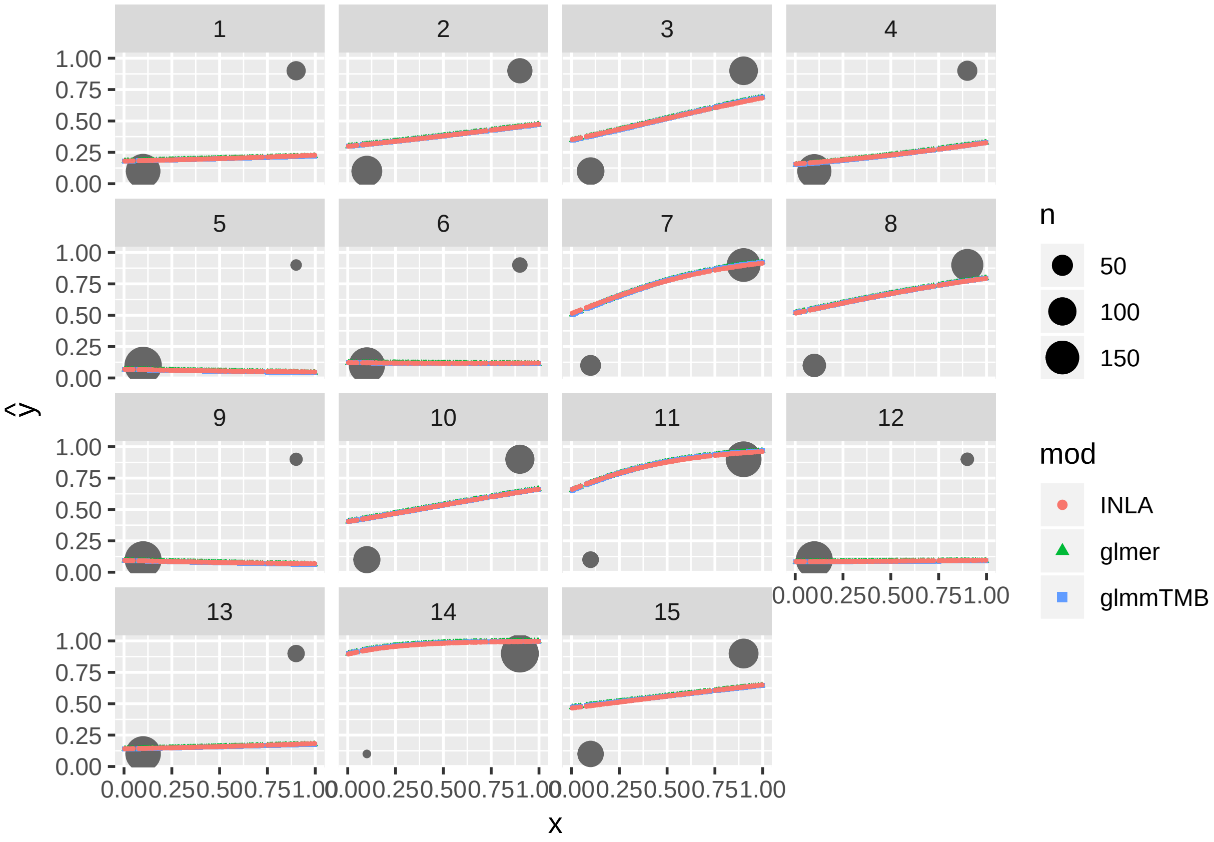
```
randdf_intercept <- data.frame(mean =
                      c(ranef(fit_glmer)$group[,1],
                        ranef(fit_glmmTMB)$cond$group[,1],
                        fit_inla$summary.random$group$mean[unique(mydata$group)],
                        random_int),
                    mod = rep(c("glmer", "glmmTMB", "INLA", "true"), each = length(unique(mydata$group)
                    x = paste0("y", sprintf("%02d", rep(1:length(unique(mydata$group)), 4))))

critval <- qnorm(0.025, lower.tail = FALSE)
randdf_intercept$low <- c(se.ranef(fit_glmer)$group[,1]*(-critval) + ranef(fit_glmer)$group[,1],
                          ranef(fit_glmmTMB)$cond$group[,1], # do not know how to get these values for
                          fit_inla$summary.random$group$`0.025quant`[unique(mydata$group)],
                          rep(0, 15))
randdf_intercept$high <- c(se.ranef(fit_glmer)$group[,1]*(critval) + ranef(fit_glmer)$group[,1],
                          ranef(fit_glmmTMB)$cond$group[,1], # do not know how to get these values for
                          fit_inla$summary.random$group$`0.975quant`[unique(mydata$group)],
                          rep(0, 15))
randdf_intercept$x2 <- rep(1:15, 4) + rep(c(-0.15,-0.05,0.05,0.15), each = 15)

ggplot(randdf_intercept) + geom_point(aes(x = mean, y = x2, col = mod)) + geom_vline(xintercept = 0, lty
  geom_segment(aes(x = low, xend = high, y = x2, yend = x2, col = mod)) +
  labs(x = "random intercept", y = "") +
  scale_y_continuous(breaks = 1:15, labels = paste0("y", sprintf("%02d", rep(1:length(unique(mydata$grou
```



```
randdf_slope <- data.frame(mean =
                       c(ranef(fit_glmer)$group[,2],
                         ranef(fit_glmmTMB)$cond$group[,2],
                         fit_inla$summary.random$group$mean[unique(mydata$group2)],
                         random_slope),
                     mod = rep(c("glmer", "glmmTMB", "INLA", "true"), each = length(unique(mydata$group)
                     x = paste0("y", sprintf("%02d", rep(1:length(unique(mydata$group)), 4))))


critval <- qnorm(0.025, lower.tail = FALSE)
randdf_slope$low <- c(se.ranef(fit_glmer)$group[,2]*(-critval) + ranef(fit_glmer)$group[,2],
                           ranef(fit_glmmTMB)$cond$group[,2], # do not know how to get these values for
                           fit_inla$summary.random$group$`0.025quant`[unique(mydata$group2)],
                           rep(0, 15))
randdf_slope$high <- c(se.ranef(fit_glmer)$group[,2]*(critval) + ranef(fit_glmer)$group[,2],
                           ranef(fit_glmmTMB)$cond$group[,2], # do not know how to get these values for
                           fit_inla$summary.random$group$`0.975quant`[unique(mydata$group2)],
                           rep(0, 15))
randdf_slope$x2 <- rep(1:15, 4) + rep(c(-0.15,-0.05,0.05,0.15), each = 15)

ggplot(randdf_slope) + geom_point(aes(x = mean, y = x2, col = mod)) + geom_vline(xintercept = 0, lty =
  geom_segment(aes(x = low, xend = high, y = x2, yend = x2, col = mod)) +
  labs(x = "random slope", y = "") +
  scale_y_continuous(breaks = 1:15, labels = paste0("y", sprintf("%02d", rep(1:length(unique(mydata$grou
```



```
fittedvalues <- data.frame(mean =
                             c(fitted.values(fit_glmer),
                               fitted.values(fit_glmmTMB),
                               fit_inla$summary.fitted.values$mean),
                           mod = rep(c("glmer", "glmmTMB", "INLA"), each = nrow(mydata)),
                           x = rep(mydata$x, 3),
                           group = rep(mydata$group,3))

ggplot(fittedvalues) + geom_count(data = mydata, aes(x = y*0.8+0.1, y = y*0.8+0.1), col = grey(0.4)) + s
  geom_point(aes(x = x, y = mean, col = mod, pch = mod, size = 1.5)) +
  facet_wrap(~ group) + labs(x = "x", y = expression(hat(y)))
```



The results are quite similar (as approximation is necessary in the computations, some differences are expected). We can make more complicated models, and (especially from INLA) we also get much more information about the model, but this is way out of the scope of this course. Models like this require computer intenisve methods, and if you are interested in learning more about this, you can take the course TMA4300 - Computer Intensive Statistical Methods in the spring semester.

R packages

```
install.packages("arm")
install.packages("reshape2")
install.packages("sp")
install.packages("glmmTBM")
install.packages("INLA", repos = "https://inla.r-inla-download.org/R/stable",
    dep = TRUE)
install.packages("lme4")
install.packages("devtools")
library(devtools)
install_github("romunov/AED")
install.packages("sjPlot")
install.packages("sjmisc")
```

Further reading

- Bolker et al. (2008):*Generalized linear mixed models: a practical guide for ecology and evolution*
- Zuur et al. (2009): "Mixed Effects Models and Extensions in Ecology with R", chapter 13 (pages 323-341). Available as free ebook from Springer for NTNU students. More explanations and less mathematics than Fahrmeir et al (2013), more focus on understanding. Link to ebook Chapter 13
- Agresti (2015): Foundations of linear and generalized linear models, Chapter 9.