Project 1

TMA4315 GLM H2020

magnwoln@stud.ntnu.no, siljeanf@stud.ntnu.no, 10020, 10013

18 November, 2020

Problem 1

1 a)

The cluster specific model is

$$y_{ij} = \underbrace{\beta_0 + \gamma_{0i}}_{\text{intercept for cluster i}} + x_{ij} \underbrace{(\beta_1 + \gamma_{1i})}_{\text{slope for cluster i}} + \epsilon_{ij},$$

where we have n_i observations $j = 1, 2 \dots n_i$ for each of the $i = 1, 2 \dots m$, clusters/individuals.

Assume ϵ_{ij} are iid normal distributed with mean zero and variance σ^2, and γ_i are iid binomially distributed with zero mean and variance matrix

$$Q = \begin{pmatrix} \tau_0^2 & \tau_{01} \\ \tau_{01} & \tau_1^2 \end{pmatrix}$$

Now let us write out the matrices for each of the variables above clusterwise.

$$\mathbf{Y}_i = \begin{pmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{in_i} \end{pmatrix}, X_i = \begin{pmatrix} 1 & x_{i1} \\ 1 & x_{i2} \\ 1 & \vdots \\ 1 & x_{in_i} \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, U_i = X_i, \boldsymbol{\gamma}_i = \begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \end{pmatrix}, \varepsilon_i = \begin{pmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \vdots \\ \varepsilon_{in_i} \end{pmatrix}$$

$\mathbf{Y}_i$ is the response for cluster/individual i, X_i is the design matrix for the fixed part, U_i for the random part. We get the model on matrix form:

$$\mathbf{Y}_i = X_i \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} + U_i \begin{pmatrix} \gamma_{0i} \\ \gamma_{1i} \end{pmatrix} + \varepsilon_i$$

Now, write the global model including all the m clusters.

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{U}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_m \end{pmatrix}, \mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_m \end{pmatrix}, \mathbf{U} = \begin{pmatrix} \mathbf{U}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{U}_m \end{pmatrix}, \boldsymbol{\gamma} = \begin{pmatrix} \boldsymbol{\gamma}_1 \\ \boldsymbol{\gamma}_2 \\ \vdots \\ \boldsymbol{\gamma}_m \end{pmatrix}, \boldsymbol{\varepsilon} = \begin{pmatrix} \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2 \\ \vdots \\ \boldsymbol{\varepsilon}_m \end{pmatrix}$$

1 b)

Estimate $\boldsymbol{\theta} = (\tau_0^2, \tau_1^2, \tau_{01}^2, \sigma^2)$ and β_0, β_1 using ML and REML.

```
#assume ni is constant for each group
mylmm <- function(y, x, m, ni, REML=FALSE){

  n <- rep(ni, m) #vector including nr of observations per cluster in each entry
  group <- factor((rep(1:m,n))) #a way to indexing the random effects for each cluster
  # Computing design matrix for fixed param, X
  X <- cbind(1,x)
  # Computing desing matrix for random param, U
  U <- list() # set up an empty list
  for (i in 1:m) {
       U[[i]] <- cbind(1,x[group==i]) # Construct and assign i'th block to i'th list element

  }

  U <- bdiag(U) # Change the list to a block diagonal matrix

  #inital value of theta where third element is corr(tau0, tau1)
  #theta is (tau_0^2, tau_1^2, corr(tau_0^2, tau_1^2), sigma^2)
  theta = c(1,1,0,1)

  #computing V matrix
  V <- function(theta, m,n){
    #find correlation between tau_0 and tau_1
    tau12 = theta[3]*(sqrt(theta[1]*theta[2]))
    Q = matrix(c( theta[1],tau12, tau12, theta[2]), ncol = 2 )
    G <- list()
    for (i in 1:m) {
         G[[i]] <- Q}
    G = bdiag(G)
    R = theta[4]*diag(sum(n))
    return(U%*%G%*%t(U) + R)
  }

  betahat <- function(theta){
    Vinv = solve(V(theta, m, n))
    return(solve(t(X)%*%Vinv%*%X)%*%t(X)%*%Vinv%*%y )
  }

  # maximize profile likelihood/restricted likelihood
  l <- function(theta){
    V = V(theta, m, n)
    betahat = betahat(theta)
    l_p = -1/2 * ( determinant(V, logarithm = TRUE)$modulus + t(y-X%*%betahat)%*%solve(V)%*%(y-X%*%betal
    if (REML == TRUE){
      l_p = l_p - 1/2*(determinant(t(X)%*%solve(V)%*%X, logarithm = TRUE )$modulus)
    }
    return(as.vector(l_p))
  }

  l_o <- optim(c(1,1,1,1), l, control=list(fnscale=-1), lower = c(0,0,-1,0), upper = c(Inf, Inf, 1, Inf
  betas = betahat(l_o$par)
  betas = matrix(betas, ncol=1)
  thetas = matrix(l_o$par)
  results = c(betas,thetas)
  results <- cbind(c("Betahat_0", "Betahat_1", "Tau_0^2", "Tau_1^2", "Tau_01", "Sigma^2"), results)
  return(results)
  }
```

Create an artifical data set in order to test the function above.

```
ni = 3 #observations in each cluster
m <- 5 #nr of clusters
n <- rep(ni, m) #vector including nr of observations per cluster in each entry
x <- rnorm(sum(n)) #data for all clusters (length 15)
group <- factor((rep(1:m,n))) #a way to indexing the random effects for each cluster
y <- 3 + 2*x + rnorm(m)[group] + rnorm(m)[group]*x + rnorm(sum(n))  #response vector y

mylmm(y,x,m,ni)
```

```
## Warning in optim(c(1, 1, 1, 1), l, control = list(fnscale = -1), lower = c(0, :
## bounds can only be used with method L-BFGS-B (or Brent)
```

```
##                  results
## [1,] "Betahat_0" "3.90869502810978"
## [2,] "Betahat_1" "1.35527474959987"
## [3,] "Tau_0^2"   "2.61638342532948"
## [4,] "Tau_1^2"   "1.05342882617925"
## [5,] "Tau_01"    "1"
## [6,] "Sigma^2"   "0.394033492120717"
```

1c)

Now we will check the estimates of the function from b) by using the sleep data set with ML and REML.

First with ML:

```
#data set for sleep study
mod <- lmer(Reaction ~ 1 + Days + (1+ Days|Subject), data=sleepstudy, REML=FALSE)
summary(mod)
```

```
## Linear mixed model fit by maximum likelihood  ['lmerMod']
## Formula: Reaction ~ 1 + Days + (1 + Days | Subject)
##    Data: sleepstudy
##
##      AIC      BIC   logLik deviance df.resid
##   1763.9   1783.1   -876.0   1751.9      174
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9416 -0.4656  0.0289  0.4636  5.1793
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 565.48   23.780
##           Days         32.68    5.717   0.08
##  Residual             654.95   25.592
## Number of obs: 180, groups:  Subject, 18
```

```
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  251.405      6.632  37.907
## Days          10.467      1.502   6.968
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.138
```

```
#comparing the function with the summary above
mylmm(sleepstudy$Reaction,sleepstudy$Days,18,10)
```

```
## Warning in optim(c(1, 1, 1, 1), l, control = list(fnscale = -1), lower = c(0, :
## bounds can only be used with method L-BFGS-B (or Brent)
```

```
##                results
## [1,] "Betahat_0" "251.405104848485"
## [2,] "Betahat_1" "10.467285959596"
## [3,] "Tau_0^2"   "565.43035548257"
## [4,] "Tau_1^2"   "32.6820383089204"
## [5,] "Tau_01"    "0.0813427844452284"
## [6,] "Sigma^2"   "654.954280574317"
```

Then with REML:

```
#data set for sleep study
mod <- lmer(Reaction ~ 1 + Days + (1+ Days|Subject), data=sleepstudy, REML=TRUE)
summary(mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ 1 + Days + (1 + Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 612.10   24.741
##           Days         35.07    5.922   0.07
##  Residual             654.94   25.592
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  251.405      6.825  36.838
## Days          10.467      1.546   6.771
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.138
```

```r
#comparing the function with the summary above
mylmm(sleepstudy$Reaction,sleepstudy$Days,18,10,REML = TRUE)
```

```
## Warning in optim(c(1, 1, 1, 1), l, control = list(fnscale = -1), lower = c(0, :
## bounds can only be used with method L-BFGS-B (or Brent)

##                  results
## [1,] "Betahat_0" "251.405104848485"
## [2,] "Betahat_1" "10.467285959596"
## [3,] "Tau_0^2"   "612.09524647234"
## [4,] "Tau_1^2"   "35.0720656865255"
## [5,] "Tau_01"    "0.0655385020333489"
## [6,] "Sigma^2"   "654.937275952784"
```

This confirms that the estimates from the lmm function made in a) is valid.

1d)

When using MLE the variance is a nuisance variable, which means it is not the variable of immediate interest, and will be biased. This leads to the ML underestimating the true variance. If we instead use a REML estimation the variance is now the variable of immediate interest and is less biased than the MLE of the variance. We know that $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{V}^{-1}\mathbf{Y}$ which is dependent on $V = \sigma^2 I + UGU^T$ where G is blockdiagonal with the covariance matrix Q. Thus when using REML-estimation for the linear mixed effects model this will influence both the fixed effects and the random effects parameters. However, asymptotically we still have the same asymptotic distribution for the fixed effects as with MLE. Therefore we can also observe that the parameter estimates for $\boldsymbol{\beta}$ are almost identical for ML and REML estimation for the sleepdata set in 1c.

The random effect parameters in $\boldsymbol{\theta} = (\tau_0^2, \tau_1^2, \tau_{01}^2, \sigma^2)$ on the other side differ for the ML and REML estimates. These are the elements of the covariance matrix for the random effects and the variance for ϵ_{ij}. We notice that τ_0^2 and τ_1^2 are smaller for the ML estimation compared to the REML estimation. This is due to the fact that the variance is more biased for ML estimation.

Problem 2

In this problem we will use a generalized linear mixed model to analyse part of the 2018 results from the Norwegian elite football league

```r
long <- read.csv("https://www.math.ntnu.no/emner/TMA4315/2020h/eliteserie.csv", colClasses = c("factor"
```

2 a)

First, fit the model.

```r
mod <- glmmTMB(goals ~ home + (1|attack) + (1|defence), poisson, data=long, REML=TRUE)
```

Coice of indices

We will study a generalized linear mixed model with crossed random effects for the intercept We will use the following indices where i is the attack team and j the defence team and k is the game number. Let $i, j = 1, 2 \ldots 16$ and $i \neq j$, and k is either 1 or 2 as all teams play two matches against each other in order for each of the teams to play on its home field. We thus have one observation for each combination of i, j and k (and $i \neq k$), $n_{ijk} = 1$, which gives a total number of 240 matches which again gives 480 rows of data as each match contains two rows in the data set. But the given data set is incomplete and contains only 192 completed matches, or 384 rows of data.

Let y_{ijk} be number of goals of team i against team j when team i is playing home if $k = 1$ or away if $k = 2$.

Distributional assumptions

The conditional distribution of the response y_{ijk} is Poisson distributed with mean λ_{ijk} conditional on the fixed and random effects. The observations y_{ijk} are conditionally independent for all i, j, k.

Structural assumptions:

The linear predictor is given as

$$\eta_{ijk} = \beta_0 + \beta_1 h_{ijk} + a_i + d_j$$

where η_{ijk} is linked with the conditional mean λ_{ijk} through a log-link function

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

Distributional assumptions for random effects

The random effects a_i and d_j for $i, j = 1, \ldots, m$ are independent and identically distributed $a_i \sim N(0, \tau_a)$ and $d_j \sim N(0, \tau_d)$.

Parameter interpretation:

λ_{ijk} is the expected number of goals score by team i playing against team j in match number k where $k \in (1, 2)$. We assume that the advantage of playing on home field is common for all teams, denoted by the fixed slope β_1. The covariate h_{ijk} is either 1 or 0 depending if team i plays at its home field. The fixed intercept which is also common for all teams is β_0. The random effects a_i and d_j are the deviation from the population intercept β_0 where a_i measures attack strength of team i and d_j the defence strength of team j. In order to predict the number of goals for one match we need to observe both Y_{ij1} and Y_{ji2} for team i and j where team i plays at its home field.

Poisson assumptions

The poisson distribution can be used to measure the probability of independent events occurring a certain number of times within a time interval such as the number of goals scored in a football match.

Let us look at the poisson assumptions and discuss if they seem reasonable for this problem.

1. The number of goals occuring within a time interval is independent of the number of goals occuring in any other disjoint time interval.
2. The probability that a single goal occurs within a small time interval is proportional to the length of the interval.
3. The probability that more than one goal may occur within a small time interval is negligible.

For a football game we can safely assume that the two last assumptions are true. This is due to the fact that there exists only one ball and it is therefore impossible to score more than one goal at a small time intervall, and the duration of the match clearly has a significant effect on the number of goals. The first assumption on the other hand is somewhat weak because there is a chance that the number of goals in the match will have some impact on the two teams playing. More concrete, it may be wrong to assume that the goals occurring before break is independent of the goals occuring after the break because of the physiological effects of the players. Still, we will make these assumptions in order to have a Poisson process where the number of events in follows a Poisson distribution.

2b

Parameter estimates

```
summary(mod)$coefficients[1]
```

```
## $cond
##              Estimate Std. Error  z value     Pr(>|z|)
## (Intercept) 0.1242138 0.07808901 1.590670 1.116839e-01
## homeyes     0.4071583 0.08744543 4.656141 3.221913e-06
```

```
#extract fixed effects from summary of model
beta0 = summary(mod)$coef$cond[1]
beta1 = summary(mod)$coef$cond[2]

#find effect of covariate
beta1_factor =round(exp(beta1),2)
```

The output above displays the parameter estimates for the fixed effects β_0 and β_1. First, let us check what happens if we increase h_{ijk} by one unit, meaning the difference of goals achieved by a team playing at its home field and away field. Both the conditional and the marginal mean will thus increase by 1.5. This means that the fixed parameter β_1 will increase the expected number of goals with approximatly 50 %. This is a very simplified model where we assume that the effect of playing on home field is the same for all of the teams which we suspect can vary a lot from team to team.

```
cbind(ranef(mod)$cond$attack, ranef(mod)$cond$defence)
```

```
##                     (Intercept)  (Intercept)
## BodoeGlimt          -0.036781062 -0.042616090
## Brann                0.012026209 -0.123934761
## Haugesund            0.011223106 -0.061931278
## Kristiansund        -0.011367328  0.008112432
## Lillestroem         -0.049915996  0.030699257
## Molde                0.078390643 -0.036630979
## Odd                  0.003654179 -0.052013600
## Ranheim_TF           0.023375599  0.062209734
## Rosenborg            0.050622609 -0.152631173
## Sandefjord_Fotball -0.058333079  0.133164228
## Sarpsborg08          0.026946364  0.006574064
## Stabaek             -0.026801293  0.085376126
## Start               -0.060500163  0.081958112
## Stroemsgodset        0.024556017  0.040486666
## Tromsoe              0.005756700 -0.009852817
## Vaalerenga           0.007147494  0.031030079
```

The output above displays the parameter estimates for the random effects a_i (column 1) for team i (row) and d_j (column 2) for team j. Observing the random effects of each of the team we notice that a high positive a_i for team i quantifies a good attack rate while a small d_i quantifies a good defence rate. We notice that the difference between these two give a good indication about the rank of the team as we will discuss further in g).

Conditional expectation If a team of average attack strength plays at its home field against another team of average defence strength we can find the conditional expectation and variance of number of goals scored by each of the teams. We know that the average of the random effect parameters are zero from the model assumptions for a_i and d_j being independent and identically distributed $a_i \sim N(0, \tau_A)$ and $d_i \sim N(0, \tau_D)$ for all $i, j = 1, 2 \ldots, m$.

```
#calculate conditional expectation and variance when random effects are zero
goals_hometeam = exp(beta0 + beta1*1)
goals_awayteam = exp(beta0 + beta1*0)
```

From the fact that the response is Poisson distributed we know that the mean and variance of y_{ijk} is λ_{ijk}. The estimates of expected number of goals for the team playing on home field λ_{ijk} is 1.701265, and for other the team is λ_{jik} is 1.132258.

2c

As we have a log-linear Poisson random intercept model with $n_{ijk} = 1$ the marginal means and variances can be determined analytically using laws of total expectations and variance.

Recall that

$$\lambda_{ijk} = \exp(\eta_{ijk}) = e^{a_i + d_j} e^{\beta_0 + h_{ijk}^T \beta_1}$$

We have that $exp(d_i + d_i)$ is lognormal distributed with mean zero and variance $\tau_A^2 + \tau_D^2$.

$$\begin{aligned} E(y_{ijk}) &= E(E(y_{ijk} \mid a_i, d_j)) = E(\lambda_{ijk}) \\ &= E\Big(e^{a_i + d_j} e^{h_{ijk}^T \beta}\Big) \\ &= \exp(\beta_0 + h_{ijk}^T \beta_1) E(\exp(d_i + d_j)) \\ &= \exp(\beta_0 + h_{ijk}^T \beta_1 + \frac{1}{2}(\tau_a^2 + \tau_d^2)) \end{aligned}$$

$$\begin{aligned} Var(y_{ijk}) &= E(Var(y_{ijk} \mid a_i, d_j)) + Var(E(y_{ijk} \mid a_i, d_j)) \\ &= E(\exp(\beta_0 + h_{ijk}^T \beta_1 + a_i + d_j)) + Var(\exp(\beta_0 + h_{ijk}^T \beta_1 + a_i + d_j)) \\ &= \underbrace{e^{\beta_0 + h_{ijk}^T \beta_1 + \frac{1}{2}(\tau_a^2 + \tau_d^2)}}_{\text{randomness of a football game}} + \underbrace{e^{2(\beta_0 + h_{ijk}^T \beta_1)} e^{\tau_a^2 + \tau_D^2} (e^{\tau_a^2 + \tau_d^2} - 1)}_{\text{variation in team strength}} \end{aligned}$$

Let us look a bit closer on the marginal variance for y_{ijk} above. The first term is the mean of the conditional variance or Poisson variance that has to do with the randomness of the game itself. In other words it is how the number of goals would vary between the two teams, dependent on the strength of the teams playing. By taking the average across teams we get a number of how much the Poisson variance contributes to the total variance on average. The second term is the variance of the conditional expectation which gives the contribution to the total variance that has to do with the variation in team strengths. We also observe that the first term is equal to the marginal expectation from the fact that the y_{ijk} is conditional poisson distributed.

Now calculate the expectation and variance found above and compute an estimate of each of these two proportions.

```r
#extract variance of random effect parameters
tauA2 = summary(mod)$var$cond$attack[1]
tauD2 = summary(mod)$var$cond$defence[1]

#calculate marginal mean
marginal_mean = exp(beta0 + 1*beta1 + 1/2*(tauA2 + tauD2))

#calculate marginal variance and the two proportions
term2 = exp(2*(beta0 + beta1*1))*exp(tauA2 + tauD2)*(exp(tauA2 + tauD2)-1)
marginal_variance = marginal_mean + term2

proportion1 = round(marginal_mean/marginal_variance*100,2)
proportion2 = round(term2/marginal_variance*100,2)
```

The marginal mean for y_{ijk} is 1.7216834 and the marginal variance is 1.7932623. The two proportions of the marginal variance has each a contribution to the variance. The first term which quantifies the randomness of a football game contributes with 96.01% to the total variance. The second term which quantifies the variation in the individual teams strength contributes with 3.99% to the total variance.

2d

Likelihood ratio test of the two random effects

We will perform two different hypothesis test, testing each of the two random effects in order to check their significance.

Testing significance of attack parameter $H_0 : \tau_a^2 = 0$ vs. $H_1 : \tau_a^2 > 0$

Recall that $a_i \sim N(0, \tau_a^2)$ and $D_j \sim N(0, \tau_d^2)$. This gives us H_0 above equivalent to $H_0 : a_i = 0$ where $i = 1, 2 \ldots 16$.

Testing significance of defence parameter $H_0 : \tau_d^2 = 0$ vs. $H_1 : \tau_d^2 > 0$.

or equivalently, $H_0 : d_j = 0$ where $j = 1, 2 \ldots 16$.

We know that standard asymptotic theory is violated for tests of this form because the random effect variance parameters, τ_a^2 and τ_d^2, are on the boundary of the parameter space. There is a 50 % chance that the MLE of τ_a (or equivalently τ_d) under H1 falls on the boundary of the parameter space such that the LRT statistic takes a value 0. Therefore the overall asymptotic distribution of the LRT statistic is a 50-50% mixture of two chi-squares with 0 and 1 degrees of freedom.

Recall that the LRT for the the models under H_0 and H_1 is the difference in the log-likelihood of the two models multiplied by -2. Let c be the critical value for the probability of rejection set to $\alpha = 0.05$.

$$\alpha = P(LRT > c) = \underbrace{\frac{1}{2} P(\chi_0^2 > 0)}_{=0} + \frac{1}{2} P(\chi_1^2 > 0)$$

```
alpha = 0.05 #significance level
c = round ( qchisq(1-2*alpha, df = 1),2)
```

The critical value c is therefore the upper 2α quantile of the chi-square distribution with 1 degree of freedom, which equals 2.71.

```
#Testing significance of attack parameter
#mod.attack:attack as only random effect
#mod: two random effects
mod.attack <- update(mod,. ~ home +(1|attack))

#find LRT
LRT.test1 = -2*(logLik(mod.attack) - logLik(mod))[1]

#find pvalue
pvalue.test1 = round(1- (0.5*pchisq(LRT.test1, 1) + 0.5*pchisq(LRT.test1, 0) ),2)

# Testing significance of defence parameter
#mod.defence: defence as only random effect
mod.defence <- update(mod,. ~ home +(1|defence))

#find LRT
LRT.test2 = -2*(logLik(mod.defence) - logLik(mod) )[1]

#find pvalue
pvalue.test2 = round( 1- (0.5*pchisq(LRT.test2, 1) + 0.5*pchisq(LRT.test2, 0) ),2)
```

We first observe the p-values for the two tests above is 0.1 for the first test and 0.26 for the second test. Observe that both of the p-values are greater than our chosen significance level of $\alpha = 0.05$. We can also check with the critical value which is 2.71 and compare with the two test statistics from the LRT in each test which is 1.6654006 and 0.4188496. Observe that both of these are greater than the critical value which confirms that we should keep the null hypotheses which again indicates that both of the random effect parameters, attack and defence, are insignificant.

Likelihood ratio test of the fixed effects

Likelihood ratio test of the home field advantage, which is a fixed covariate.

$H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$

Lets do another LRT for this hypothesis and calculate the p-value. We will use a model based on ML because the fitted model under H_0 and H_1 leads to different mean structure using REML.

```
#model based on MLE
mod.mle <- glmmTMB(goals ~ home + (1|attack) + (1|defence), poisson, data=long, REML=FALSE)
#mod.intercept: attack and defence as random effects, only intercept as fixed effect
mod.mle.intercept = glmmTMB(goals ~ (1|attack) + (1|defence), poisson, data=long, REML=FALSE)

#find LRT
LRT.test3 = -2*(logLik(mod.mle.intercept) - logLik(mod.mle))[1]

#find p-value
pvalue.test3 = 1-pchisq(LRT.test3, 1)
```

The pvalue for the test is 2.5073284×10^{-6} which is clearly smaller than the significance level of $\alpha = 0.05$ indicating that we should reject the null hypothesis and hence β_1 is significant.

e)

In this problem we create a function to calculate the rank of each team. Based on the data frame long we calculate the points, from winning and getting a tie ,for each game, the goal difference and the total number of goals. The last two calculations are used to determine the rank if two teams have the same amount of points.

```
ranking <- function(long, n){ #Create a fuction that takes in a dataframe

liste <- data.frame(row.names = unique(long$attack), points=  c(1:16)*0, goal_diff = c(1:16)*0, total_g

for(i in 1:n){ #Loop over all the data
  if (i%%2 == 0){ #Divide by two to look at each match
      liste[as.character(long$attack[i-1]), "goal_diff"] <- liste[as.character(long$attack[i-1]), "goal_
      liste[as.character(long$attack[i]), "goal_diff"] <- liste[as.character(long$attack[i]), "goal_diff

      liste[as.character(long$attack[i-1]), "total_goal"] <- liste[as.character(long$attack[i-1]), "tota
      liste[as.character(long$attack[i]), "total_goal"] <- liste[as.character(long$attack[i]), "total_go

     if(long$goals[i-1] > long$goals[i]) { #If the home team wins add 3 points to the home team
      liste[as.character(long$attack[i-1]), "points"] <- liste[as.character(long$attack[i-1]), "points"]
      }

     else if(long$goals[i-1] < long$goals[i]){ #If the away team winns add 3 points to the away team
       liste[as.character(long$attack[i]), "points"] <- liste[as.character(long$attack[i]), "points"] +
     }

     else if(long$goals[i-1] == long$goals[i]){ #If its a tie add 1 poit to each team
       liste[as.character(long$attack[i-1]), "points"] <- liste[as.character(long$attack[i-1]), "points"
      liste[as.character(long$attack[i]), "points"] <- liste[as.character(long$attack[i]), "points"] +
    }

  }
}

order.points <- order(liste$points, liste$goal_diff, liste$total_goal, unique(long$attack)) #Give the t
liste$position[order.points] <- nrow(liste):1 #Add the position to each team
liste2 <- data.frame(row.names = unique(long$attack), liste$position) #Create new data frame with team
return(t(liste2))
}
```

The ranking of the teams based only on the 192 matches that have already been played:

```
#transpose ranking output (again) to look tidier
t(ranking(long, 384))
```

```
##                     liste.position
## Molde                            3
## Sandefjord_Fotball              16
## Stroemsgodset                   12
## Stabaek                         14
## Odd                              7
## Haugesund                        4
## BodoeGlimt                      11
## Lillestroem                     13
## Start                           15
## Tromsoe                          8
## Sarpsborg08                      9
## Rosenborg                        1
## Kristiansund                    10
## Vaalerenga                       6
## Ranheim_TF                       5
## Brann                            2
```

f)

Based on the fitted model, the expected number of goals in given matches get the expected value λ. λ is then used to simulate 1000 runs of Eliteserien from the first match to the last in the series.

```
n = 480 #2 * the amount of matches
lambda = predict(mod,  type="response", data=long) #Predict an expected number of goals
df <- matrix(0, nrow = 1000, ncol = 16) #Create empty matrix for postitionw
colnames(df) <- unique(long$attack)

for (i in 1:1000){ #1000 samples
   long$goals  <- rpois(n,lambda) #New estimate of y from a poisson distribution
   rankinglist <- ranking(long, n) #Rank each simulation
   df[i,] <- rankinglist #Add the ranking to the matrix df
 }
```

Using the data obtained by running the 1000 simulations we calculate the expected rank.

```
totalrank <- matrix(0, nrow = 1, ncol = 16, byrow=FALSE) #Empty matrix for the total rank
colnames(totalrank) <- unique(long$attack)

```

```
for (i in 1:16){ #Find the total rank for each team and add to the total rank matrix
  totalrank[1,i]= sum(df[,i])/1000
}
totalrank
```

```
##      Molde Sandefjord_Fotball Stroemsgodset Stabaek   Odd Haugesund BodoeGlimt
## [1,] 6.895             10.885         9.022   9.647 7.699     7.058      8.487
##      Lillestroem  Start Tromsoe Sarpsborg08 Rosenborg Kristiansund Vaalerenga
## [1,]       9.357 10.512   8.626       8.337     5.897        8.753       8.96
##      Ranheim_TF Brann
## [1,]      9.219 6.646
```

We see that the expected rank is quite similar as the actual result of Eliteserien in 2018. We also calculate the probability of a given team to get a given rank.

```
prob_rank <- matrix(0, nrow = 16, ncol = 16, byrow=FALSE) #Create empty matrix for position probability
colnames(prob_rank) <- unique(long$attack)
rownames(prob_rank) <- 1:16

for (i in 1:16){ #calculate the probability for getting a given position for a given team
  counter <- count(df, i) #Count number of times each team get each posititon
  prob_rank[,i] <- counter[,2]/1000 #Add the probability to the probability matrix
}

as.data.frame(prob_rank)
```

```
##    Molde Sandefjord_Fotball Stroemsgodset Stabaek   Odd Haugesund BodoeGlimt
## 1  0.113              0.019         0.038   0.036 0.084     0.085      0.058
## 2  0.077              0.026         0.051   0.037 0.079     0.093      0.061
## 3  0.086              0.026         0.052   0.041 0.067     0.086      0.059
## 4  0.091              0.033         0.052   0.052 0.070     0.080      0.075
## 5  0.079              0.037         0.061   0.050 0.075     0.080      0.050
## 6  0.083              0.043         0.070   0.056 0.053     0.071      0.064
## 7  0.062              0.043         0.072   0.056 0.078     0.077      0.074
## 8  0.066              0.057         0.058   0.058 0.062     0.076      0.059
## 9  0.047              0.055         0.068   0.067 0.063     0.050      0.060
## 10 0.053              0.068         0.059   0.067 0.062     0.063      0.064
## 11 0.053              0.071         0.070   0.084 0.065     0.044      0.075
## 12 0.049              0.072         0.071   0.067 0.057     0.053      0.060
## 13 0.037              0.097         0.066   0.086 0.055     0.047      0.060
## 14 0.043              0.101         0.091   0.082 0.046     0.031      0.069
## 15 0.032              0.123         0.067   0.079 0.048     0.033      0.057
## 16 0.029              0.129         0.054   0.082 0.036     0.031      0.055
##    Lillestroem Start Tromsoe Sarpsborg08 Rosenborg Kristiansund Vaalerenga
## 1        0.041 0.024   0.051       0.058     0.144        0.051      0.050
## 2        0.038 0.035   0.059       0.053     0.130        0.051      0.050
## 3        0.045 0.031   0.065       0.075     0.112        0.059      0.051
## 4        0.062 0.033   0.070       0.053     0.086        0.044      0.069
## 5        0.057 0.033   0.064       0.068     0.070        0.079      0.057
## 6        0.061 0.056   0.050       0.076     0.060        0.067      0.047
## 7        0.045 0.057   0.067       0.065     0.058        0.070      0.062
## 8        0.073 0.049   0.071       0.065     0.070        0.058      0.061
## 9        0.072 0.057   0.076       0.065     0.057        0.063      0.087
## 10       0.054 0.070   0.050       0.076     0.050        0.068      0.060
## 11       0.073 0.069   0.048       0.065     0.040        0.062      0.061
## 12       0.080 0.078   0.067       0.058     0.028        0.079      0.069
## 13       0.072 0.085   0.071       0.066     0.029        0.059      0.066
## 14       0.061 0.089   0.057       0.058     0.032        0.061      0.068
## 15       0.090 0.102   0.056       0.045     0.021        0.066      0.072
## 16       0.076 0.132   0.078       0.054     0.013        0.063      0.070
##    Ranheim_TF Brann
## 1       0.041 0.107
## 2       0.051 0.109
## 3       0.048 0.097
## 4       0.056 0.074
## 5       0.067 0.073
## 6       0.067 0.076
## 7       0.044 0.070
## 8       0.058 0.059
## 9       0.059 0.054
## 10      0.075 0.061
## 11      0.062 0.058
## 12      0.072 0.040
## 13      0.065 0.039
## 14      0.077 0.034
## 15      0.079 0.030
## 16      0.079 0.019
```

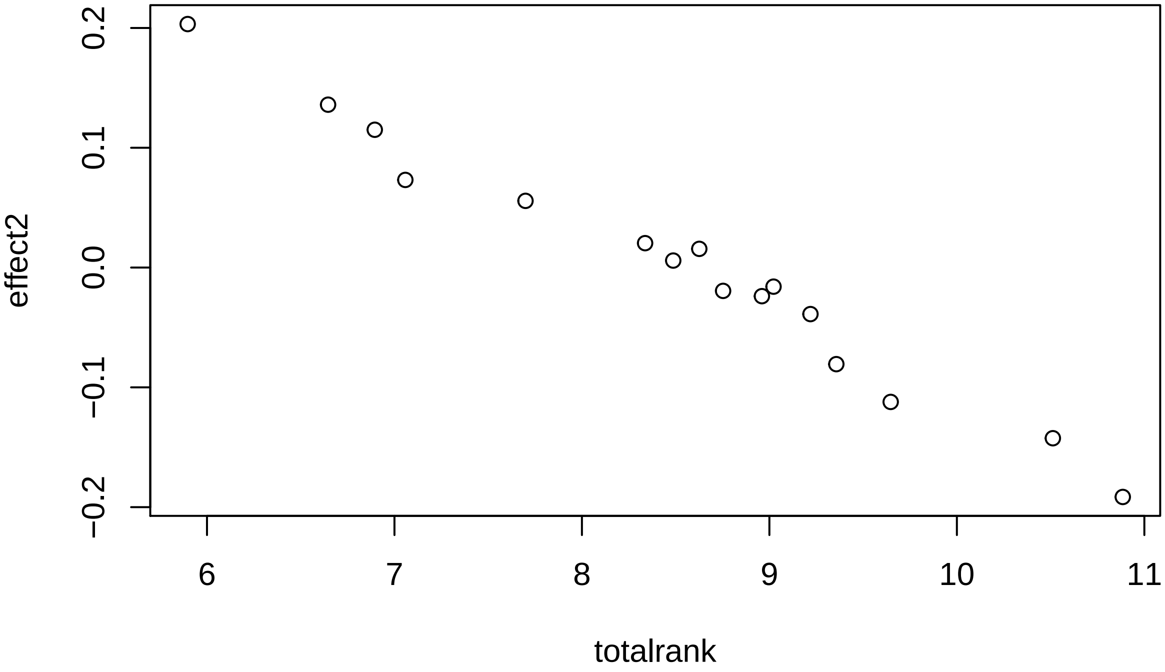
g)

We look at the difference between attack and defence and the relation to the the total rank.

```
re <- ranef(mod)

effect <- re$cond$attack - re$cond$defence #Calculate the difference between attack and defence
effect2 <- c(effect[6,1], effect[10,1], effect[14,1], effect[12,1], effect[7,1],effect[3,1],effect[1,1]

plot(totalrank, effect2) #Plot the effect towards the rank
```



Observing the plot there seems to be a somewhat linear relationship between how well the team perform and the attack and defence strentgh. This means that the random effect parameters are highly correlated to the ranking of the teams after 1000 simulations of the full Eliteserie.