Diagonal random-effects covariance matrix using the lme4::modular framework.

Introduction

This document demonstrates how to implement a diagonal random effects covariance matrix using the **lme4** package in R. The goal is to fit a mixed-effects model with a diagonal covariance structure for the random effects and compare the results with the **glmmTMB** package.

Step 1: Load Required Libraries and Data

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
library(nloptr)
library(glmmTMB)

data(cbpp)
```

Step 2: Prepare the Model Formula

```
lf <- lFormula(incidence / size ~ period + (period | herd), data = cbpp,
               control = lmerControl(check.nobs.vs.nRE = "ignore"))

lower_indices <- lf$reTrms$lower
```

Step 3: Create the Deviance Function

```
devfun <- mkLmerDevfun(lf$fr, lf$X, lf$reTrms)
```

Step 4: Write a Wrapper Function for Diagonal Covariance

```
diagonal_wrapper <- function(theta_diag) {
  theta <- numeric(length(lf$reTrms$lower))

  theta[lower_indices == 0] <- theta_diag

  devfun(theta)
}
```

Step 5: Fit the Model Using nloptwrap

```
theta_diag_init <- rep(0, sum(lower_indices == 0))
lower_bounds <- rep(0, length(theta_diag_init))
```

```r
upper_bounds <- rep(Inf, length(theta_diag_init))

opt <- nloptwrap(
  par = theta_diag_init,
  fn = diagonal_wrapper,
  lower = lower_bounds,
  upper = upper_bounds,
  control = list(algorithm = "NLOPT_LN_BOBYQA")
)
```

Step 6: Create the merMod Object

```r
lmer_fit <- mkMerMod(
  rho = environment(devfun),
  opt = opt,
  reTrms = lf$reTrms,
  fr = lf$fr,
  mc = match.call()
)

print(lmer_fit)
```

```
## Linear mixed model fit by REML ['lmerMod']
## REML criterion at convergence: -58.918
## Random effects:
##  Groups   Name        Std.Dev.  Corr
##  herd     (Intercept) 4.161e-07
##           period2     0.000e+00  NaN
##           period3     8.361e-02 0.00  NaN
##           period4     2.129e-07 0.00  NaN 0.00
##  Residual             1.179e-01
## Number of obs: 56, groups:  herd, 15
## Fixed Effects:
## (Intercept)      period2      period3      period4
##      0.2198      -0.1458      -0.1305      -0.1785
```

Step 7: Compare with glmmTMB

```r
glmmTMB_fit <- glmmTMB(incidence / size ~ period + diag(period | herd), data = cbpp, REML = TRUE)
print(summary(glmmTMB_fit))
```

```
##  Family: gaussian  ( identity )
## Formula:          incidence/size ~ period + diag(period | herd)
## Data: cbpp
##
##      AIC      BIC   logLik deviance df.resid
##    -40.9    -22.7     29.5    -58.9       47
##
## Random effects:
##
## Conditional model:
##  Groups   Name        Variance  Std.Dev.  Corr
##  herd     (Intercept) 4.371e-11 6.611e-06
```

```
##           period2     2.128e-13 4.613e-07 0.00
##           period3     6.990e-03 8.361e-02 0.00 0.00
##           period4     2.729e-15 5.224e-08 0.00 0.00 0.00
##  Residual             1.390e-02 1.179e-01
## Number of obs: 56, groups:  herd, 15
##
## Dispersion estimate for gaussian family (sigma^2): 0.0139
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2198         NA      NA       NA
## period2      -0.1458         NA      NA       NA
## period3      -0.1305         NA      NA       NA
## period4      -0.1785         NA      NA       NA
```

Comparison of Results

Fixed Effects

We compare the fixed effects estimates from **lme4** and **glmmTMB**.

```
# Compare fixed effects
cat("lme4 fixed effects:\n")
```

```
## lme4 fixed effects:
```

```
print(fixef(lmer_fit))
```

```
## (Intercept)     period2     period3     period4
##   0.2197948  -0.1457808  -0.1304627  -0.1785360
```

```
cat("glmmTMB fixed effects:\n")
```

```
## glmmTMB fixed effects:
```

```
print(fixef(glmmTMB_fit)$cond)
```

```
## (Intercept)     period2     period3     period4
##   0.2197948  -0.1457808  -0.1304627  -0.1785360
```

Random Effects Standard Deviations

We compare the random effects standard deviations from **lme4** and **glmmTMB**.

```
# Compare random effects standard deviations
cat("lme4 random effects standard deviations:\n")
```

```
## lme4 random effects standard deviations:
```

```
print(VarCorr(lmer_fit)$herd)
```

```
##              (Intercept) period2     period3      period4
## (Intercept) 1.731753e-13       0 0.000000000 0.000000e+00
## period2     0.000000e+00       0 0.000000000 0.000000e+00
## period3     0.000000e+00       0 0.006989957 0.000000e+00
## period4     0.000000e+00       0 0.000000000 4.532982e-14
## attr(,"stddev")
##  (Intercept)      period2      period3      period4
## 4.161434e-07 0.000000e+00 8.360596e-02 2.129080e-07
## attr(,"correlation")
```

```
##             (Intercept) period2 period3 period4
## (Intercept)           1     NaN       0       0
## period2             NaN       1     NaN     NaN
## period3               0     NaN       1       0
## period4               0     NaN       0       1
```

```
cat("glmmTMB random effects standard deviations:\n")
```

```
## glmmTMB random effects standard deviations:
```

```
print(VarCorr(glmmTMB_fit)$cond$herd)
```

```
##              (Intercept)      period2     period3      period4
## (Intercept) 4.370748e-11 0.000000e+00 0.000000000 0.000000e+00
## period2     0.000000e+00 2.127808e-13 0.000000000 0.000000e+00
## period3     0.000000e+00 0.000000e+00 0.006989948 0.000000e+00
## period4     0.000000e+00 0.000000e+00 0.000000000 2.728778e-15
## attr(,"stddev")
##  (Intercept)      period2      period3      period4
## 6.611164e-06 4.612817e-07 8.360591e-02 5.223770e-08
## attr(,"correlation")
##             (Intercept) period2 period3 period4
## (Intercept)           1       0       0       0
## period2               0       1       0       0
## period3               0       0       1       0
## period4               0       0       0       1
## attr(,"blockCode")
## diag
##    0
```

Conclusion

The results show that the fixed effects estimates are identical between **lme4** and **glmmTMB**. The random effects standard deviations are also consistent, with minor differences due to numerical precision and optimization algorithms. The diagonal covariance structure was successfully implemented in both models.