Linear Mixed Effects Models

Why Mixed Effects Models?

Pooling

Complete:

$\hat{y}_i = \beta_o \, y_i \sim Norm(\hat{y}_i, \sigma^2)$

Partial:

$\hat{y}_i = \beta_0 + b_{g_i} \, y_i \sim Norm(\hat{y}_i, \sigma^2) \, b_g \sim Norm(0, \sigma_b^2)$

No pooling:

$\hat{y} = \beta_{g_i} \, y_i \sim Norm(\hat{y}_i, \sigma^2)$

Fitting Linear Mixed Models

```
data("cortbowl")

dat <- cortbowl
mod <- lmer(log(totCort) ~ Implant + days + Implant:days + (1|Ring),
            data = dat, REML = T)

summary(mod)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: log(totCort) ~ Implant + days + Implant:days + (1 | Ring)
   Data: dat

REML criterion at convergence: 611.9

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.4766 -0.4826  0.0172  0.5303  3.1872
```

```
Random effects:
 Groups   Name        Variance Std.Dev.
 Ring     (Intercept) 0.1145   0.3384
 Residual             0.3763   0.6134
Number of obs: 287, groups:  Ring, 151

Fixed effects:
                    Estimate Std. Error t value
(Intercept)           3.5675     0.1008  35.383
ImplantP             -1.8052     0.1470 -12.279
days20               -1.3903     0.1364 -10.190
daysbefore           -1.6531     0.1238 -13.356
ImplantP:days20       1.6248     0.1939   8.378
ImplantP:daysbefore   1.7200     0.1829   9.406

Correlation of Fixed Effects:
            (Intr) ImplnP days20 dysbfr ImP:20
ImplantP    -0.686
days20      -0.644  0.441
daysbefore  -0.703  0.482  0.524
ImplntP:d20  0.453 -0.662 -0.703 -0.369
ImplntP:dys  0.476 -0.690 -0.355 -0.677  0.526
```

```r
round(fixef(mod), 3)
```

```
        (Intercept)            ImplantP              days20          daysbefore
              3.568              -1.805              -1.390              -1.653
    ImplantP:days20 ImplantP:daysbefore
              1.625               1.720
```

```r
ranef(mod)
```

```
$Ring
         (Intercept)
898054   0.248849793
898055   0.118458626
898057  -0.107882775
898058   0.069989589
898059  -0.080864976
898061  -0.083968387
898062   0.264700928
898064   0.019519951
898070  -0.076488856
898075   0.194329598
898080   0.245225900
```

898081	-0.119566227
898084	-0.101086556
898085	0.034140643
898086	0.051104880
898087	-0.092742604
898089	0.084068954
898093	0.226925628
898094	-0.165038157
898098	-0.094085401
898100	0.151722193
898151	0.039263095
898152	0.077750735
898153	0.033041348
898154	0.224800810
898169	0.122055304
898173	0.221827234
898174	0.070278658
898179	0.007180216
898181	-0.218893634
898185	0.020458748
898188	-0.015841427
898191	0.036438083
898193	-0.035237985
898195	-0.008611546
898196	-0.032790540
898317	-0.063843404
898318	0.080706408
898322	0.135036393
898323	-0.034384455
898324	-0.001269365
898327	-0.013174155
898328	0.359955469
898331	-0.057341712
898332	0.086360054
898333	-0.074507406
898337	0.221843215
898339	0.022359312
898341	0.375355780
898342	0.062102547
898343	0.033820550
898345	-0.064915460
898347	0.006671093
898348	-0.021510348
898352	0.058847612
898356	0.232101440

```
898357   0.187995990
898361   0.185896326
898365   0.010901152
898367  -0.054216801
898368  -0.011181686
898369  -0.004997251
898373   0.308418670
898375   0.050461461
898376   0.096289589
898378  -0.220179330
898379   0.271214359
898383  -0.260396624
898384   0.064431684
898388   0.078739195
898389   0.353409606
898391   0.251941285
898392   0.057556443
M005002  0.107670948
M005003 -0.222330428
M005005  0.033919342
M005006  0.063256971
M005007 -0.004564252
M005008 -0.007313808
M005009 -0.291585401
M005010  0.216311277
M005011 -0.449035883
M005012  0.427200385
M005013  0.145577577
M005017 -0.222970414
M005019 -0.171184076
M005020 -0.362003976
M005021 -0.024818264
M005022 -0.524571781
M005023 -0.048634002
M005024 -0.075800303
M005025  0.149163917
M005026 -0.256002751
M005027  0.281525089
M005028 -0.404713542
M005029  0.043651121
M005030 -0.045148534
M005032  0.215562304
M005033  0.081783398
M005034  0.045514392
M005035 -0.274305997
```

```
M005036 -0.167997460
M005037 -0.167240354
M005038 -0.312930804
M005039 -0.127534852
M005040  0.182789987
M005041  0.119764952
M005042  0.119448816
M005043  0.349901235
M005044 -0.135631906
M005049 -0.208336383
M005233 -0.057440870
M005234 -0.736808459
M005238  0.300812564
M005239  0.041287092
M005240  0.014459139
M005241 -0.023423826
M005242 -0.084729815
M005243 -0.088532689
M005244 -0.188022515
M005245 -0.093014706
M005247  0.093083147
M005248 -0.069686254
M005249  0.132258946
M005250 -0.058380191
M005451 -0.151629636
M005452 -0.172083339
M005453 -0.117208190
M005454  0.129095509
M005456 -0.596505802
M005457 -0.496300458
M005458 -0.182030023
M005459  0.030833328
M005460 -0.094701333
M005461  0.346766362
M005462  0.025017934
M005463 -0.524776546
M005464 -0.177202179
M005465 -0.037068688
M005467 -0.084854248
M005468 -0.219016777
M005470  0.165908277
M005471 -0.039595295
M005472  0.076073158
M005474  0.049262303
M005476  0.595000659
```

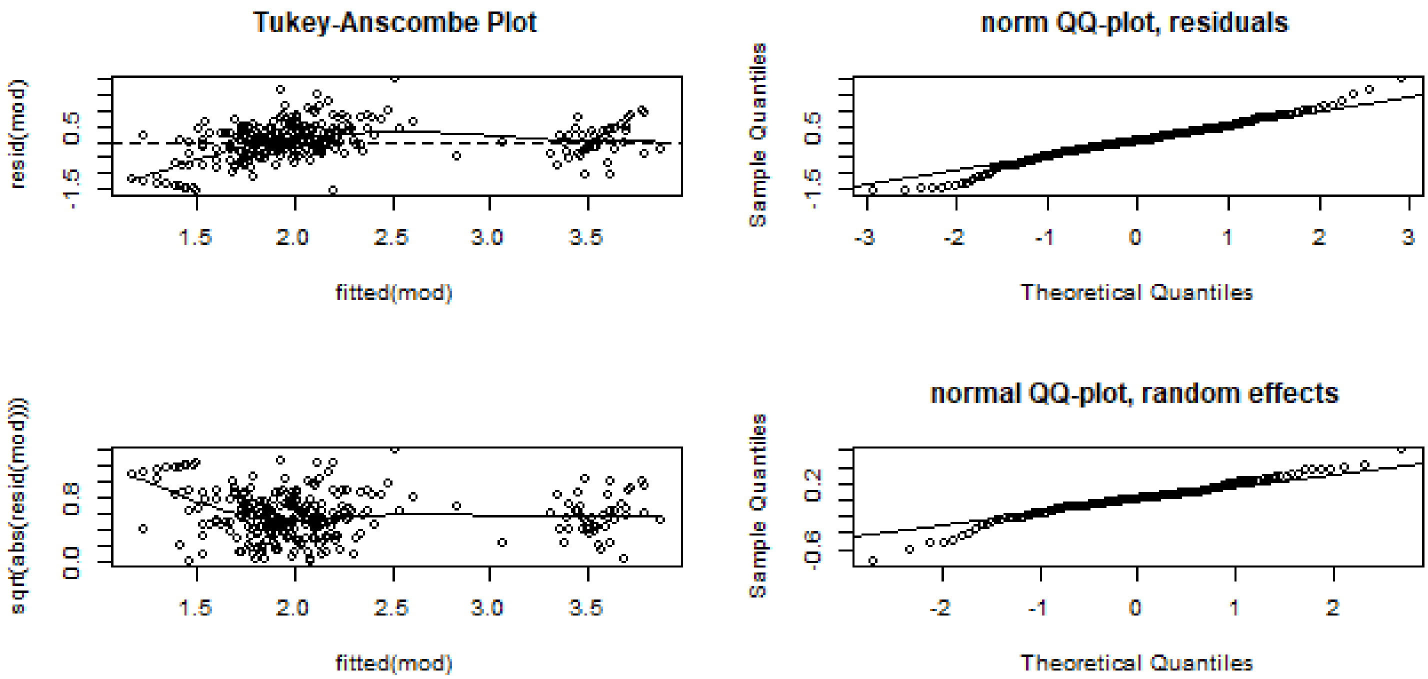
```
M005477  0.004554585
M005478 -0.007062036
M005479 -0.060078982
M005480  0.207396948
M005481  0.030426886

with conditional variances for "Ring"
```

Restricted Maximum Liklihood Estimation

```r
mod <- lmer(log(totCort) ~ Implant + days + Implant:days + (1|Ring),
            data = dat, REML = F) # use ML
```

Assessing Model Assumptions

```r
par(mfrow=c(2,2))
scatter.smooth(fitted(mod), resid(mod)); abline(h=0, lty=2)
title("Tukey-Anscombe Plot")

qqnorm(resid(mod), main = "norm QQ-plot, residuals") # qq of residuals
qqline(resid(mod))
scatter.smooth(fitted(mod), sqrt(abs(resid(mod))))

qqnorm(ranef(mod)$Ring[, 1], main = "normal QQ-plot, random effects")
qqline(ranef(mod)$Ring[, 1]) # qq random effects
```



Drawing Conclusions

```r
nsim <- 2000
bsim <- sim(mod, n.sim = nsim)
str(bsim)
```

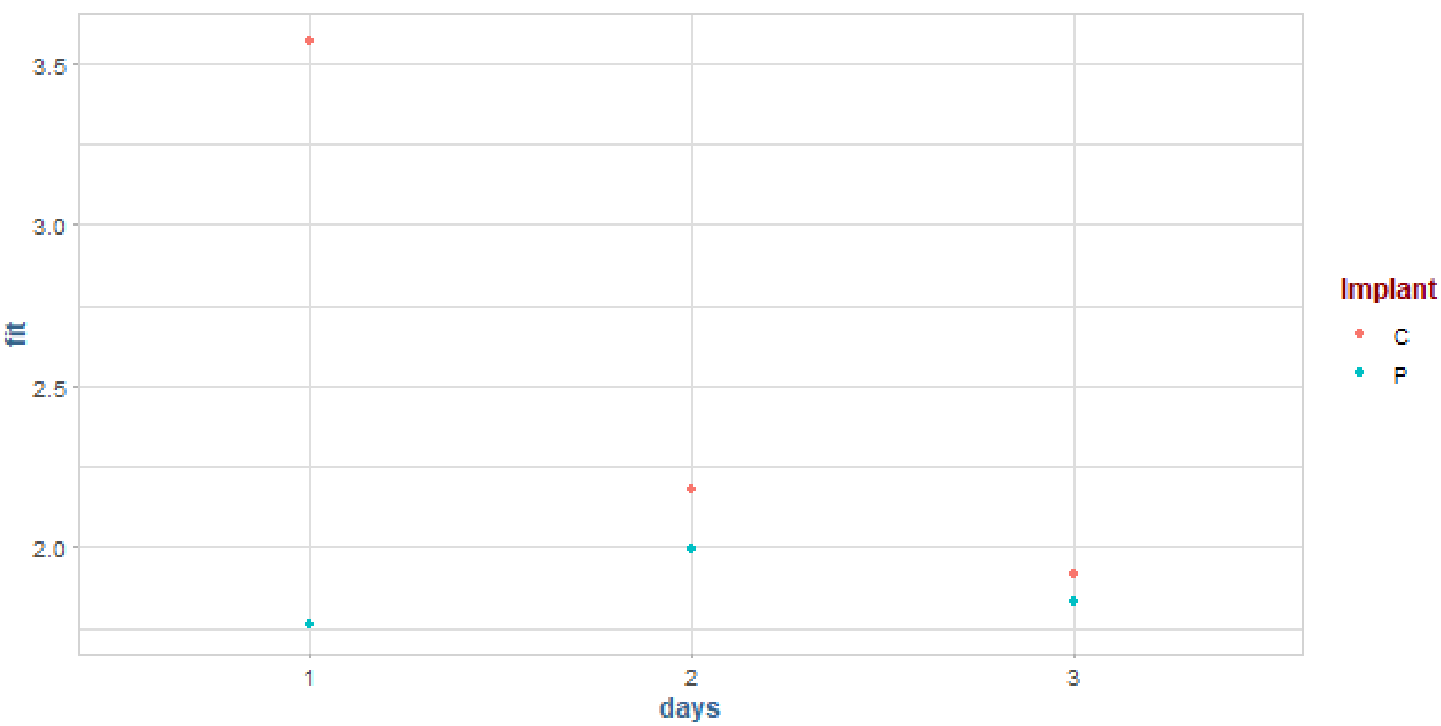
```
Formal class 'sim.merMod' [package "arm"] with 3 slots
  ..@ fixef: num [1:2000, 1:6] 3.6 3.57 3.44 3.6 3.59 ...
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : NULL
  .. .. ..$ : chr [1:6] "(Intercept)" "ImplantP" "days20" "daysbefore" ...
  ..@ ranef:List of 1
  .. ..$ Ring: num [1:2000, 1:151, 1] 0.32357 -0.00278 0.34527 0.32662 0.31222 ...
  .. .. ..- attr(*, "dimnames")=List of 3
  .. .. .. ..$ : NULL
  .. .. .. ..$ : chr [1:151] "898054" "898055" "898057" "898058" ...
  .. .. .. ..$ : chr "(Intercept)"
  ..@ sigma: num [1:2000] 0.579 0.564 0.614 0.59 0.623 ...
```

```r
round(apply(bsim@fixef, 2, quantile, prob=c(0.025, 0.5, 0.975)), 3)
```

```
      (Intercept) ImplantP days20 daysbefore ImplantP:days20
2.5%        3.373   -2.102 -1.660     -1.898           1.240
50%         3.565   -1.801 -1.387     -1.657           1.618
```

```
97.5%       3.773   -1.502 -1.122     -1.405           2.010
      ImplantP:daysbefore
2.5%                 1.352
50%                  1.722
97.5%                2.077
```

```
imp_factor <- factor(c("C", "P"), levels = levels(dat$Implant))
days_factor <- factor(c(1, 2, 3))

newdat <- expand.grid(Implant = imp_factor, days = days_factor)

Xmat <- model.matrix(~Implant + days + Implant:days, data = newdat)
fitmat <- matrix(ncol = nsim, nrow = nrow(newdat))
for(i in 1:nsim) fitmat[, i] <- Xmat %*% bsim@fixef[i, ] # fitted values
newdat$lower <- apply(fitmat, 1, quantile, prob = 0.025)
newdat$upper <- apply(fitmat, 1, quantile, prob = 0.975)
newdat$fit <- Xmat %*% fixef(mod)

ggplot(newdat, aes(days, fit, color = Implant)) +
   geom_point()
```



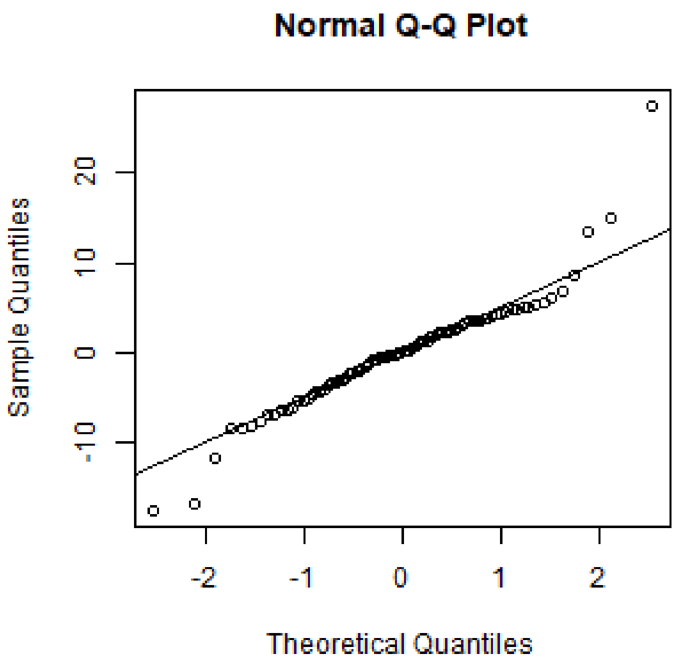
Frequentist Results

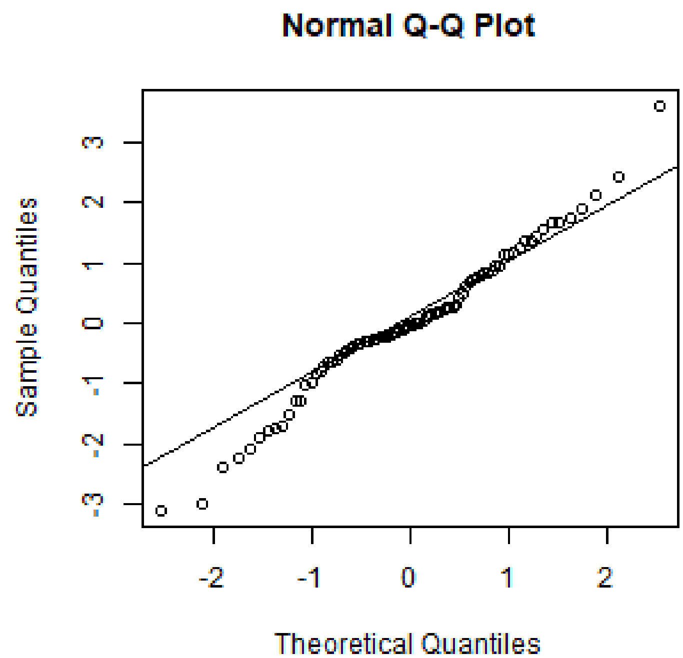
Random Intercept and Random Slope

```
data(wingbowl)
dat <- wingbowl
dat$Age.z <- scale(dat$Age)
mod <- lmer(Wing ~ Age.z + Implant + Age.z:Implant + (Age.z|Ring),
         data = dat, REML = F)
mod
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: Wing ~ Age.z + Implant + Age.z:Implant + (Age.z | Ring)
   Data: dat
      AIC       BIC    logLik  deviance  df.resid
1280.4391 1307.1778 -632.2195 1264.4391       201
Random effects:
 Groups   Name        Std.Dev. Corr
 Ring     (Intercept) 6.394
          Age.z       1.898    -0.12
 Residual             2.542
Number of obs: 209, groups:  Ring, 86
Fixed Effects:
   (Intercept)          Age.z       ImplantP  Age.z:ImplantP
       155.442         24.954          4.554           2.185
```

```
par(mfrow=c(1, 2))
qqnorm(ranef(mod)$Ring[, 1]) # Intercept
qqline(ranef(mod)$Ring[, 1])

qqnorm(ranef(mod)$Ring[, 2]) # Slope
qqline(ranef(mod)$Ring[, 2])
```






```r
nsim <- 2000
bsim <- sim(mod, n.sim = nsim)
apply(bsim@fixef, 2, quantile, prob = c(0.025, 0.975))
```

```
      (Intercept)    Age.z ImplantP Age.z:ImplantP
2.5%     153.3737 24.05478 1.650539      0.8205036
97.5%    157.5241 25.84310 7.520347      3.5108143
```

```r
quantile(bsim@fixef[, "Age.z:ImplantP"] / sd(dat$Age), prob = c(0.025, 0.975))
```

```
     2.5%     97.5%
0.1544729 0.6609667
```

```r
newdat <- expand.grid(Age = seq(23, 45, length = 100),
                      Implant = levels(dat$Implant))

newdat$Age.z <- (newdat$Age - mean(dat$Age)) / sd(dat$Age)

head(newdat)
```

```
       Age Implant     Age.z
1 23.00000       C -2.231267
2 23.22222       C -2.189430
3 23.44444       C -2.147593
4 23.66667       C -2.105756
5 23.88889       C -2.063919
6 24.11111       C -2.022082
```

```
dim(newdat)
```

```
[1] 200   3
```

```
Xmat <- model.matrix(~Age.z + Implant + Age.z:Implant, data = newdat)
fitmat <- matrix(ncol = nsim, nrow = nrow(newdat))

dim(fitmat)
```

```
[1]  200 2000
```

```
dim(bsim@fixef)
```

```
[1] 2000    4
```

```
head(bsim@fixef)
```

```
     (Intercept)    Age.z ImplantP Age.z:ImplantP
[1,]    153.7014 25.24287 4.836193       1.863615
[2,]    154.5255 24.64564 6.530480       2.082173
[3,]    156.7886 25.07462 2.045257       2.106844
[4,]    155.7526 24.63780 3.508220       2.801926
[5,]    156.0946 24.81796 2.898831       1.767796
[6,]    153.3235 25.23465 8.549733       2.263593
```

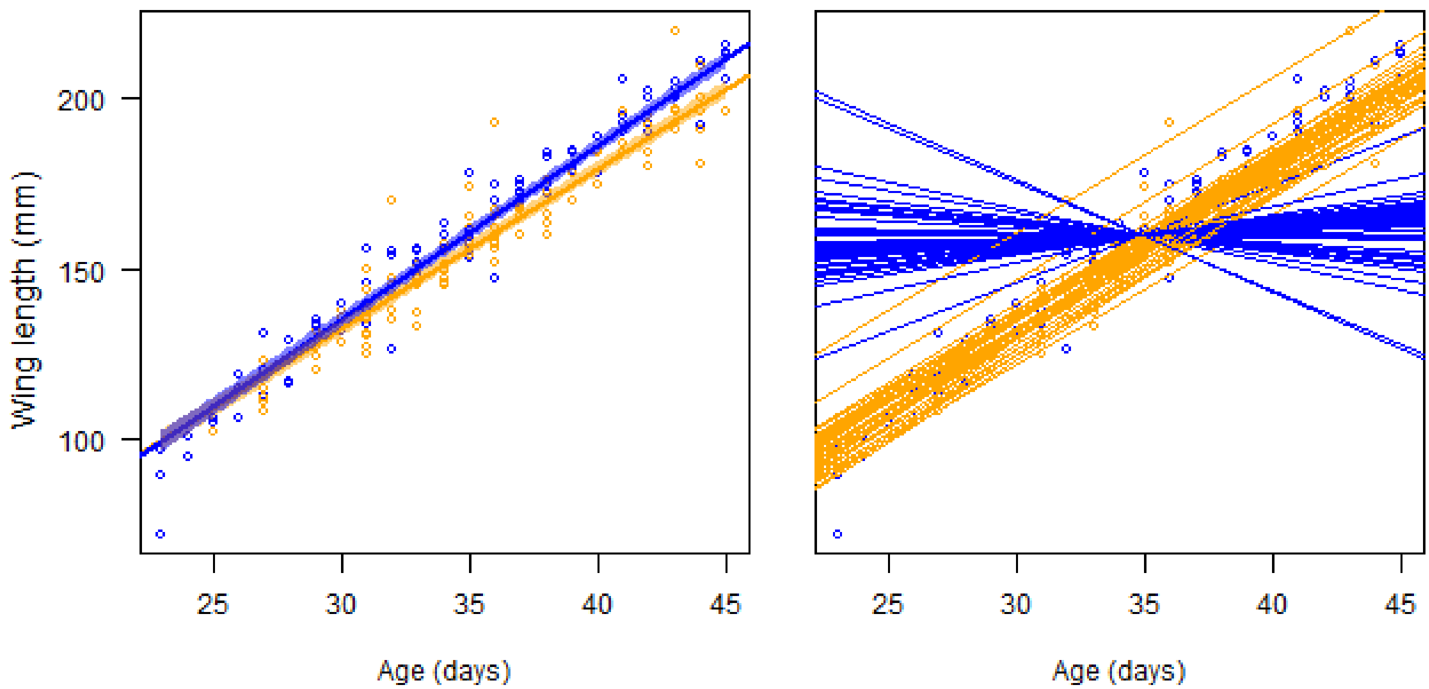
```
for(i in 1:nsim) {
   fitmat[, i] <- Xmat %*% bsim@fixef[i, ]
}

dim(bsim@fixef)
```

```
[1] 2000    4
```

```
newdat$lower <- apply(fitmat, 1, quantile, prob = 0.025)
newdat$upper <- apply(fitmat, 1, quantile, prob = 0.975)
```

```
par(mfrow = c(1, 2), mar=c(5, 1, 1, 1), oma = c(0, 4, 0, 0))
plot(dat$Age.z, dat$Wing, pch=1, cex = 0.8, las = 1,
     col = c("orange", "blue")[as.numeric(dat$Implant)],
     xlab = "Age (days)", ylab = NA, xaxt = "n")
at.x_orig <- seq(25, 45, by = 5) # values on the x-axis, original scale
at.x <- (at.x_orig - mean(dat$Age)) / sd(dat$Age) # transformed scale
axis(1, at = at.x, labels = at.x_orig) # original values at transformed
mtext("Wing length (mm)", side = 2, outer = T, line = 2, cex = 1.2, adj=0.6)
abline(fixef(mod)[1], fixef(mod)[2], col = "orange", lwd=2) # for C
abline(fixef(mod)[1] + fixef(mod)[3], fixef(mod)[2] + fixef(mod)[4],
       col = "blue", lwd = 2)

for(i in 1:2) {
```

```
   index <- newdat$Implant == levels(newdat$Implant)[i]

   x <- c(newdat$Age.z[index], rev(newdat$Age.z[index]))
   y <- c(newdat$lower[index], rev(newdat$upper[index]))

   polygon(x, y,
           border = NA, col = c(rgb(1, 0.65, 0, 0.5), rgb(0, 0, 1, 0.5))[i])
}

plot(dat$Age.z, dat$Wing, pch=1, cex=0.8, las=1,
     col = c("orange", "blue")[as.numeric(dat$Implant)],
     xlab = "Age (days)", ylab = NA, yaxt = "n", xaxt = "n")
at.x_orig <- seq(25, 45, by = 5)
ax.x <- (at.x_orig - mean(dat$Age)) / sd(dat$Age)
axis(1, at = at.x, labels = at.x_orig)
indtreat <- tapply(dat$Implant, dat$Ring, function(x) as.character(x[1]))
for(i in 1:86) {
   if(indtreat[i] == "C") abline(fixef(mod)[1] + ranef(mod)$Ring[i, 1],
                              fixef(mod)[2] + ranef(mod)$Ring[i, 2],
                              col = "Orange") else
                                 abline(fixef(mod)[1] + fixef(mod)[3],
                                        ranef(mod)$Ring[i, 1], fixef(mod)[2] +
                                           fixef(mod)[4] + ranef(mod)$Ring[i, 2],
                                        col = "blue")
}
```



Nested and Crossed Random Effects

```
data(cortbowl)
dat <- cortbowl

mod <- lmer(log(totCort) ~ Implant + days + Implant:days + (1|Brood) + (1|Ring),
            data = dat, REML = F)

mod
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: log(totCort) ~ Implant + days + Implant:days + (1 | Brood) +
    (1 | Ring)
   Data: dat
      AIC       BIC    logLik  deviance  df.resid
 604.2934  637.2287 -293.1467  586.2934       278
Random effects:
 Groups   Name        Std.Dev.
 Ring     (Intercept) 0.1917
 Brood    (Intercept) 0.2486
 Residual             0.6117
Number of obs: 287, groups:  Ring, 151; Brood, 54
Fixed Effects:
```

```
        (Intercept)             ImplantP               days20
              3.592               -1.796               -1.383
         daysbefore      ImplantP:days20  ImplantP:daysbefore
             -1.639                1.617                1.693
```

```r
mod <- lmer(log(totCort) ~ Implant + days + Implant:days + (1|Brood/Ring),
            data = dat, REML = F)
mod
```

```
Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: log(totCort) ~ Implant + days + Implant:days + (1 | Brood/Ring)
   Data: dat
      AIC       BIC    logLik  deviance  df.resid
 604.2934  637.2287 -293.1467  586.2934       278
Random effects:
 Groups     Name        Std.Dev.
 Ring:Brood (Intercept) 0.1917
 Brood      (Intercept) 0.2486
 Residual               0.6117
Number of obs: 287, groups:  Ring:Brood, 151; Brood, 54
Fixed Effects:
        (Intercept)             ImplantP               days20
              3.592               -1.796               -1.383
         daysbefore      ImplantP:days20  ImplantP:daysbefore
             -1.639                1.617                1.693
```

```r
data("ellenberg")

ellenberg$gradient <- paste(ellenberg$Year, ellenberg$Soil)
table(ellenberg$Species, ellenberg$gradient)
```

```
     1952 Loam 1952 Sand 1953 Loam 1953 Sand
  Ae        11        11        11        11
  Ap        11        11        11        11
  Be        11        11        11        11
  Dg        11        11        11        11
  Fp        11        11         0         0
  Pp        11        11         0         0
```

```r
ellenberg$Water.z <- as.numeric(scale(ellenberg$Water))
mod <- lmer(log(Yi.g) ~ Water.z + I(Water.z^2) +
               (Water.z + I(Water.z^2)|Species) + (1|gradient),
            data = ellenberg)
```

```
boundary (singular) fit: see ?isSingular
```

```
mod
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: log(Yi.g) ~ Water.z + I(Water.z^2) + (Water.z + I(Water.z^2) |
    Species) + (1 | gradient)
   Data: ellenberg
REML criterion at convergence: 540.8211
Random effects:
 Groups   Name         Std.Dev. Corr
 Species  (Intercept)  1.4376
          Water.z      0.6417    0.81
          I(Water.z^2) 0.5201   -0.80 -1.00
 gradient (Intercept)  0.5169
 Residual              0.7833
Number of obs: 208, groups:  Species, 6; gradient, 4
Fixed Effects:
 (Intercept)       Water.z  I(Water.z^2)
     3.58591      -0.08136      -0.11472
convergence code 0; 0 optimizer warnings; 1 lme4 warnings
```