Lab 7 - Random Effects and Least Significant Differences

Lab Goals

1. Derive the distribution of conditional normal random vectors; and simulate these.
2. Provide optimal predictions of random effects in a Randomized Block design via a conditional argument.
3. Implement random effects models in R using the `lme4` package.
4. Express a split plot design in terms of effect matrices.

Conditional Normal Random Vectors

Here we will examine the distribution of a normal random vector, when you know part of the vector. That is, we we have

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \right)$$

and we see y_1, what does that tell us about y_2?

This can be done directly by looking at the joint density of y_1 and y_2 and treating y_1 as fixed, but there is a simpler argument looking at $z = y_2 - Ay_1$ for a specific choice

$$A = \Sigma_{21}\Sigma_{11}^{-1}$$

1. Show that $cov(z, y_1) = 0$ and hence that z is independent of y_1. What is $E(z)$?

$$\begin{aligned} cov(z, y_1) &= cov(y_2 - Ay_1, y1) \\ &= cov(y_2, y_1) - Acov(y_1, y_1) \\ &= \Sigma_{21} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{11} \\ &= 0 \end{aligned}$$

Hence z is independent of y_1. We note that

$$Ez = E(y_2) - AEy_1 = \mu_2 - \Sigma_{21}\Sigma_{11}^{-1}\mu_1$$

2. By expressing $y_2 = z + Ay_1$ and using the independence above, find $E(y_2|y_1)$.

$$\begin{aligned} E(z - Ay_1|y_1) &= E(z|y_1) - AE(y_1|y_1) \\ &= E(z) - Ay_1 \text{ (independence)} \\ &= \mu_2 - \Sigma_{21}\Sigma_{11}^{-1}\mu_1 + \Sigma_{21}\Sigma_{11}^{-1}y_1 \\ &= \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y_1 - \mu_1) \end{aligned}$$

3. Taking the same approach show that $var(y_2|y_1) = var(z)$.

$$var(z + Ay_1|y_1) = var(z|y_1) + Avar(y_1|y_1)A^T = var(z)$$

*because y__1 conditioned on itself has no variance, and z is independent of y_1.

4. Find $var(z)$

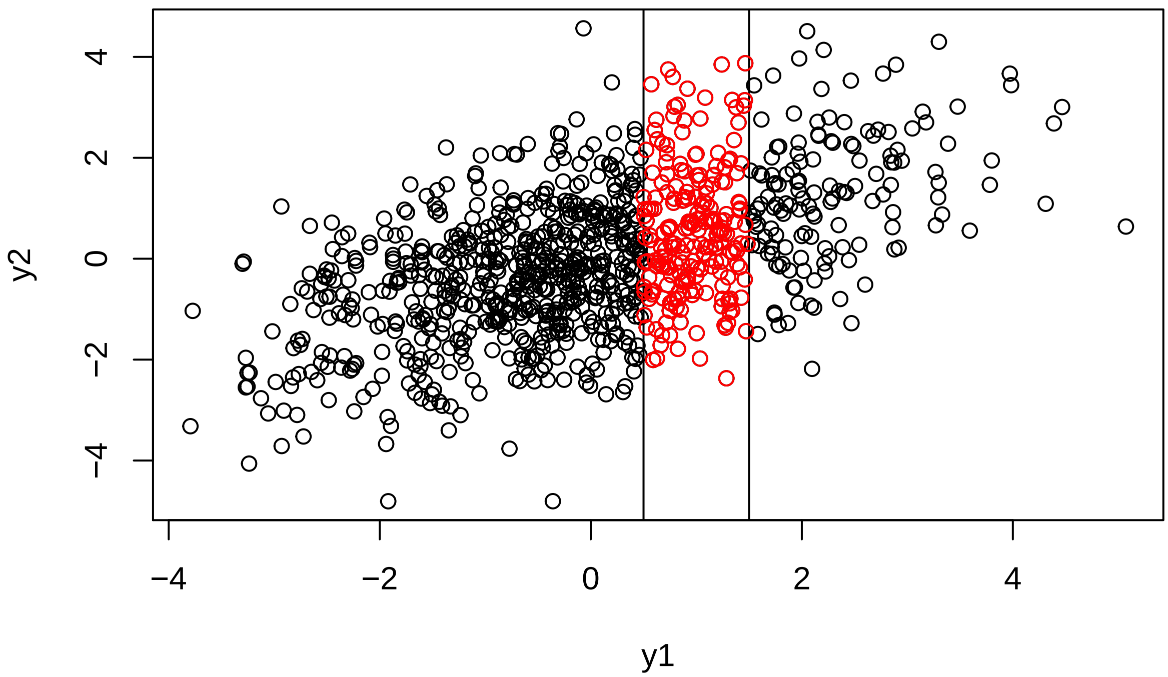
$$\begin{aligned}
var(y_2 - Ay_1) &= var(y_2) - Acov(y_1, y_2) - cov(y_2, y_1)A^T + Avar(y_1)A^T \\
&= \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} + \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{11}\Sigma_{11}^{-1}\Sigma_{12} \\
&= \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}
\end{aligned}$$

5. The code below conducts a simulation study. First, we generate 1000 bivariate normal random variables, with variance 2 and covariance 1:

```
eta = rnorm(1000)    # This is a common effect to both y's
y1 = rnorm(1000) + eta
y2 = rnorm(1000)+eta
```

Now we'll look at taking the y_2 that correspond to y_1 close to the value 1. We can illustrate this with the following plot

```
plot(y1,y2)
abline(v = c(0.5,1.5))

#Take only those y2 near y1=1

cy2 = y2[ (y1>0.5) & (y1<1.5) ]

# Plot these in red
points(  y1[ (y1>0.5) & (y1<1.5) ], cy2, col='red')
```



According to our conditional variance formula, these should be Normal with mean 1/2 and variance (2 - 1/2) = 3/2

```
mean(cy2)
```

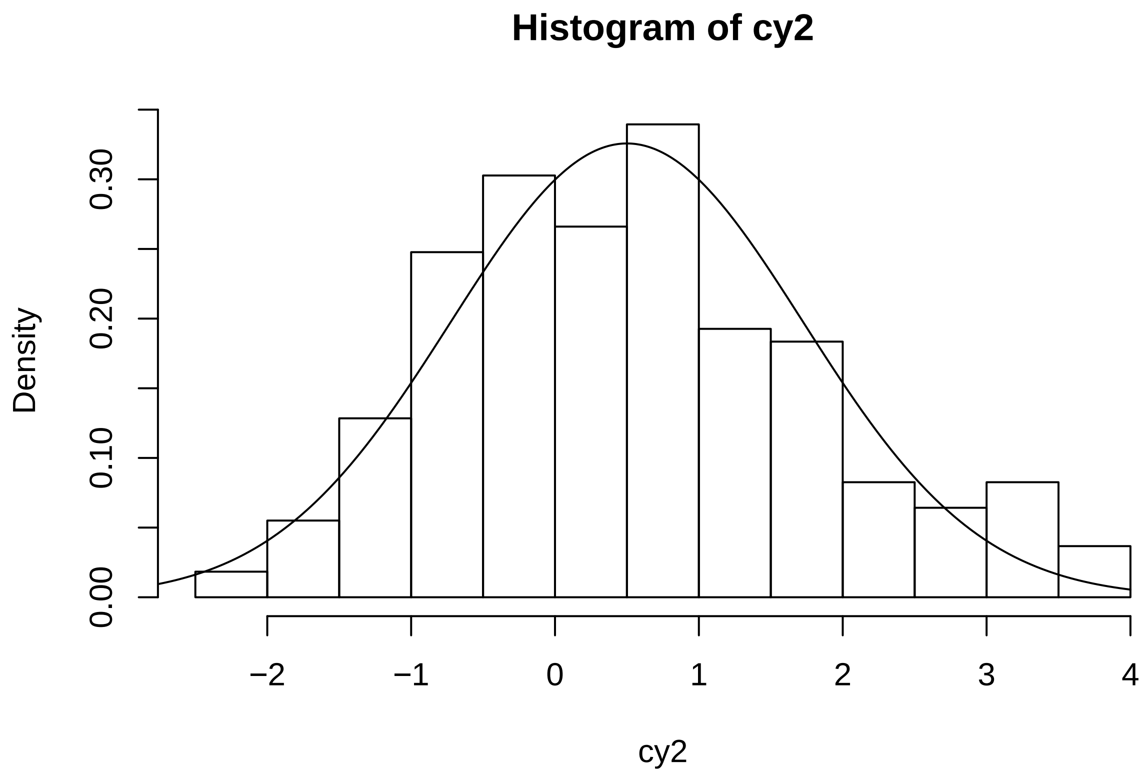
```
## [1] 0.54079
```

```
var(cy2)
```

```
## [1] 1.656855
```

Of course, these will be a bit distorted because we used `y1`'s that were only close to 1. If we look at the histogram, in comparison to the expected density.

```
hist(cy2,20,prob=TRUE)

xpts = seq(-4,4,len=1001)
lines(xpts, dnorm(xpts,mean=0.5,sd=sqrt(3/2)))
```



Best Linear Unbiassed Predictors in a Randomized Block Design

Here we will use the result in a Randomized Blocks Design in which we have

$$y_{ij} = \mu + \alpha_i + \beta_j + e_{ij}$$

with $e_{ij} \sim N(0, \sigma_e^2)$ and $\alpha_i \sim N(0, \sigma_a^2)$ and asume that the β_j sum to zero.

1. We consider estimating α_i by $\hat{\alpha}_i = \bar{y}_{i\cdot} - \mu$. Give an expression for the error, $\hat{\alpha}_i - \alpha_i$ in terms of μ, α, β and e.

$$\hat{\alpha}_i - \alpha_i = \mu + \alpha_i + \bar{\beta} + \bar{\epsilon}_{i\cdot} - \mu - \alpha_i = \bar{\epsilon}_{i\cdot}.$$

2. Hence find the expected squared error $E(\hat{\alpha}_i - \alpha_i)^2$.

$$E(\hat{\alpha}_i - \alpha_i)^2 = E(\bar{\epsilon}_{i\cdot}) = \sigma_e^2/r$$

3. The BLUP estimates for α_i is

$$\tilde{\alpha}_i = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2/r}(\bar{y}_{i\cdot} - \mu)$$

   Derive the espected squared error $E(\tilde{\alpha}_i - \alpha_i)^2$

We'll start by writing

$$c = \frac{\sigma_a^2}{\sigma_a^2 + \sigma_e^2/r}$$

then,

$$
\begin{aligned}
E(\tilde{\alpha}_i - \alpha_i)^2 &= E(c\alpha_i + c\bar{\epsilon}_{i\cdot} - \alpha_i)^2 \\
&= (1-c)^2 E\alpha_i^2 + c^2 E\bar{\epsilon}_{i\cdot}^2 \\
&= \frac{\sigma_e^4/r^2}{(\sigma_a^2+\sigma_e^2/r)^2}\sigma_a^2 + \frac{\sigma_a^4}{(\sigma_a^2+\sigma_e^2/r)^2}\sigma_e^2/r \\
&= \frac{\sigma_a^2\sigma_e^2/r}{(\sigma_a^2+\sigma_e^2/r)^2}(\sigma_a^2+\sigma_e^2/r) \\
&= \frac{\sigma_a^2}{(\sigma_a^2+\sigma_e^2/r)^2}\sigma_e^2/r
\end{aligned}
$$

4. Show that the error for $\tilde{\alpha}_i$ is smaller than for $\hat{\alpha}_i$.

We see that the difference is a factor of $\sigma_a^2/(\sigma_a^2+\sigma_e^2/r) < 1$.

5. *bonus* We have assumed we know μ in the calculations above. What if we replace it by $\bar{y}_{\cdot\cdot}$?

A Randomized Blocks Analysis

In Lab 6, we analyzed the distance that 4 different brands of golfballs travelled when hit. In fact, there were 10 different golfers in this study, given by the first column. We'll first load in these data

```
golf = read.table('golfballs.dat')
names(golf) = c('golfer','brand','dist')
golf$golfer = as.factor(golf$golfer)
```

1. We can express the mean-model framework for this design by kronecker products:

```
# Golfer design matrix
Xg = diag(10)%x%matrix(1,4,1)

# Check this
Xg[1:12,1:4]
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1    0    0    0
##  [2,]    1    0    0    0
##  [3,]    1    0    0    0
##  [4,]    1    0    0    0
##  [5,]    0    1    0    0
##  [6,]    0    1    0    0
##  [7,]    0    1    0    0
##  [8,]    0    1    0    0
##  [9,]    0    0    1    0
## [10,]    0    0    1    0
## [11,]    0    0    1    0
## [12,]    0    0    1    0
```

```
# Design for golf ball brands
Xb = matrix(1,10,1)%x%diag(4)

# Check
Xb[1:12,]
```

```
##       [,1] [,2] [,3] [,4]
##  [1,]    1    0    0    0
##  [2,]    0    1    0    0
##  [3,]    0    0    1    0
##  [4,]    0    0    0    1
##  [5,]    1    0    0    0
##  [6,]    0    1    0    0
##  [7,]    0    0    1    0
##  [8,]    0    0    0    1
##  [9,]    1    0    0    0
## [10,]    0    1    0    0
## [11,]    0    0    1    0
## [12,]    0    0    0    1
```

To be able to estiamate a model, we have to use a reference category for each factor, and then add an intercept.

```
X = cbind( rep(1,40), Xg[,-1], Xb[,-1])
```

2. To get parameters for each golfer and each brand, we can simply do linear regression

```
betahat = solve( t(X)%*%X)%*%t(X)%*%golf$dist
```

We can compare this to a simple linear regression model

```
mod1 = lm(dist~golfer+brand,data=golf)
mod1$coefficients
```

```
## (Intercept)     golfer2     golfer3     golfer4     golfer5     golfer6
##    203.7025     39.5750     15.5500     28.7000     -2.7750     44.9750
##     golfer7     golfer8     golfer9    golfer10      brandB      brandC
##      6.6000     35.4500     19.1250     46.2750      6.1300     18.2800
##      brandD
##     -6.3200
```

```
t(betahat)
```

```
##          [,1]   [,2]  [,3] [,4]   [,5]   [,6] [,7]  [,8]   [,9]  [,10]
## [1,] 203.7025 39.575 15.55 28.7 -2.775 44.975  6.6 35.45 19.125 46.275
##      [,11] [,12] [,13]
## [1,]  6.13 18.28 -6.32
```

3. We can also express this in terms of averages for each group. To do this, we need to

a. Extract the over-all mean

```
ydd = mean(golf$dist)
ydd
```

```
## [1] 231.5725
```

b. Average for each golfer

```
yid = t(Xg)%*%golf$dist/4
t(yid)
```

```
##         [,1]  [,2]    [,3]    [,4]   [,5]  [,6]    [,7]    [,8]   [,9]
## [1,] 208.225 247.8 223.775 236.925 205.45 253.2 214.825 243.675 227.35
##      [,10]
## [1,] 254.5
```

c. Average each brand

```
ydj = t(Xb)%*%golf$dist/10
t(ydj)
```

```
##         [,1]   [,2]   [,3]   [,4]
## [1,] 227.05 233.18 245.33 220.73
```

Notice that if we predict the first element of the data set by

```
yid[1] + ydj[1] - ydd
```

```
## [1] 203.7025
```

We get the same thing as

```
mod1$fit[1]
```

```
##        1
## 203.7025
```

bonus How do you map the coefficients of the linear model onto these averages? (Note that this isn't completely straightforward).

d. We can now estimate variances

```
# For errors
sig2e = sum( (golf$dist - Xg%*%yid - Xb%*%ydj + ydd)^2)/(3*9)

# For golfers
sig2g = sum( (yid - ydd)^2 )/9 - sig2e/4
```

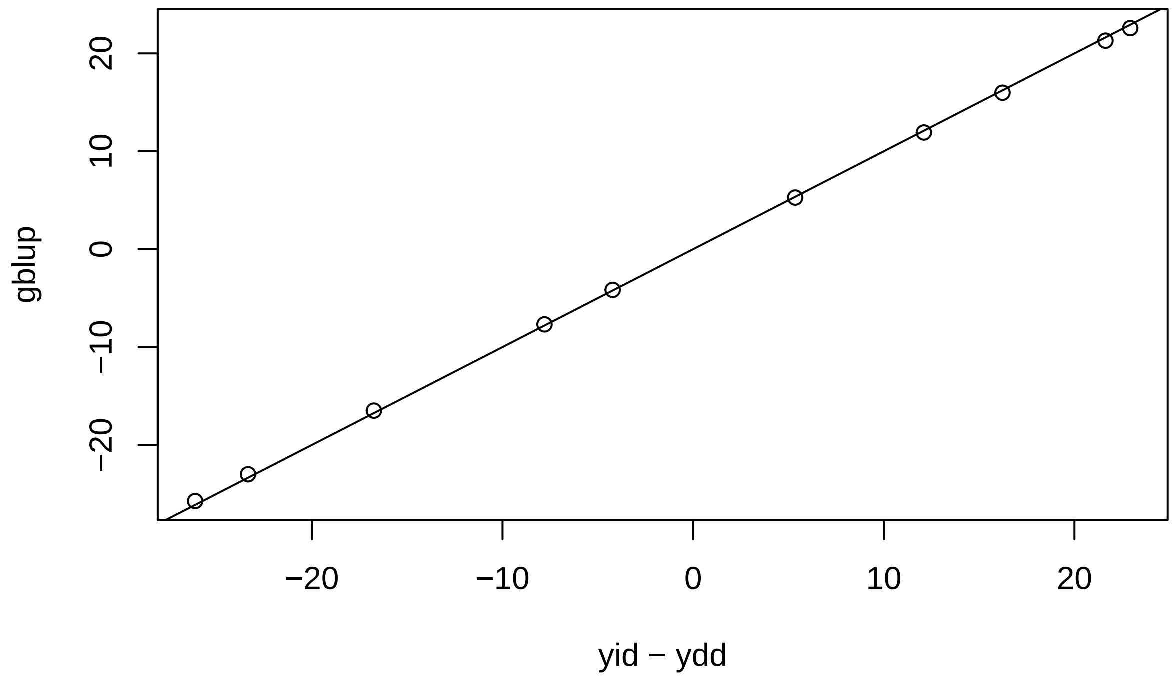
e. And look at BLUPS for golfers

```
gblup = sig2g/(sig2g+sig2e/4)*(yid-ydd)
t(gblup)
```

```
##           [,1]     [,2]      [,3]     [,4]      [,5]     [,6]      [,7]
## [1,] -22.99516 15.98261 -7.679828 5.271725 -25.72829 21.30112 -16.49476
##          [,8]      [,9]   [,10]
## [1,] 11.91986 -4.158778 22.5815
```

We can plot these, too

```
plot(yid-ydd,gblup)
abline(c(0,1))
```



Here the shrinkage is very slight because between-golfer variance is much larger than the error variance.

4. To perform a mixed effects analysis in R, we use the package `lme4`

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.5.1
```

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

```
mod2 = lmer(dist~brand + (1|golfer),data=golf)
summary(mod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: dist ~ brand + (1 | golfer)
##    Data: golf
##
## REML criterion at convergence: 257.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.16064 -0.38059  0.00973  0.52941  1.42909
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  golfer   (Intercept) 330.32   18.175
##  Residual              20.25    4.499
## Number of obs: 40, groups:  golfer, 10
##
```

```
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  227.050      5.921  38.347
## brandB         6.130      2.012   3.046
## brandC        18.280      2.012   9.084
## brandD        -6.320      2.012  -3.141
##
## Correlation of Fixed Effects:
##        (Intr) brandB brandC
## brandB -0.170
## brandC -0.170  0.500
## brandD -0.170  0.500  0.500
```

Notice that we estimate variances between golfers and hits within golfers as we obtained manually.

We can also get the BLUPs for each golfer

```
ranef(mod2)
```

```
## $golfer
##    (Intercept)
## 1   -22.995163
## 2    15.982611
## 3    -7.679828
## 4     5.271725
## 5   -25.728285
## 6    21.301119
## 7   -16.494764
## 8    11.919861
## 9    -4.158778
## 10   22.581501
```

These are not the least squares estimates, but they are the same as the blups we calculated.

A Simulation Exercize

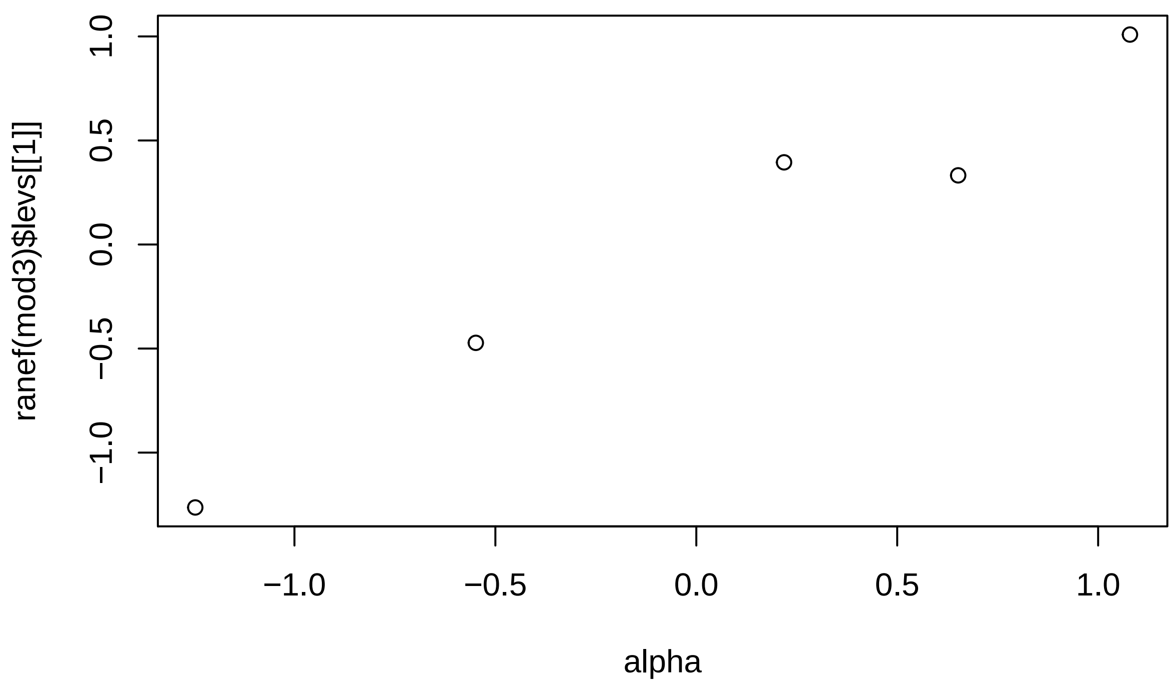
If you get time. Simulate a one-way random effects model with 5 levels and 2 observations per level and set $\mu = 0$, $\sigma_a^2 = \sigma_e^2 = 1$.

For example

```
Xa = diag(5)%x%matrix(1,2,1)
levs = as.factor( (1:5)%x%rep(1,2) )

# Random effects
alpha = rnorm(5)

# Create data
y = Xa%*%alpha + rnorm(10)

# Let's run a model
mod3 = lmer(y~1+(1|levs))

# See what this looks like
summary(mod3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ 1 + (1 | levs)
##
## REML criterion at convergence: 32.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.1852 -0.4853 -0.2466  0.3738  1.4787
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  levs     (Intercept) 1.1130   1.0550
##  Residual             0.9664   0.9831
## Number of obs: 10, groups:  levs, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)   0.1694     0.5650     0.3
```

```r
# And check estimation of random effects
plot(alpha,ranef(mod3)$levs[[1]])
```



Simulate this 1000 times. Show that the estimate of μ that we would find is more variable than we would have if we modeled this with fixed effects. Show that the blups for this model have smaller squared error (over different αs) than using the un-shrunken estimates.

```r
mu = rep(NA,1000)           # Over-all mean
raneff.err = rep(NA,1000)   # Squared error of random effects
```

```r
alphahat.err = rep(NA,1000) # Squared error of averages for each observation

for(i in 1:1000){
  # Random effects
  alpha = rnorm(5)

  # Create data
  y = Xa%*%alpha + rnorm(10)

  # Let's run a model
  mod3 = lmer(y~1+(1|levs))

  # We'll record an estimate of mu
  mu[i] = mean(y)

  # Error in random effects
  raneff.err[i] = mean( (ranef(mod3)$levs[[1]]-alpha)^2 )

  # Error in averages

  alphahat = t(Xa)%*%y/2 - mu[i]

  alphahat.err[i] = mean( (alphahat - alpha)^2 )
}


# And we note that

var(mu)
```

```
## [1] 0.2928274
```

```r
# is much larger than the 1/10 we would expect (with sigma = 1) in
# a fixed effects model

# and we can compare

mean(raneff.err)
```

```
## [1] 0.5652571
```

```r
mean(alphahat.err)
```

```
## [1] 0.6126689
```

A Longitudinal Data Analysis

First, let's look at an interraction between categorical and continuous covariates.

1. Suppose that X^S is a matrix coding the indicator for different levels of a (fixed) factor S. So that X_1^S is an indicator that $S = 1$ and so forth. We will also take t to be a continuous covariate – say, the time at which a measurement is taken.

Writing

$$y = X^S\beta_x + tX^S\beta_t + e$$

Show that this gives each level of S its own slope an intercept.

Here we observe $\beta_x = (\beta_{x_1}, \ldots, \beta_{x_s})^T$ and similarly for β_t.

Then given that the rth row X^S has a 1 in column k if the rth observation corresponds to group k, we have for group k that

$$y = \beta_{x_k} + \beta_{t_k} t + \epsilon$$

which is different for each group k.

2. In a fixed effects setting in which X^S is given with reference coding, write down a model that extracts a reference linear regression and interpret the interraction effects.

In this case, we replace the first column of X^S with a vector of all 1's and therefore have that for the kth group

$$y = \beta_{x_1} + \beta_{x_k} + (\beta_{t_1} + \beta_{t_k})t + \epsilon$$

and we interpret β_{t_k} as the difference in slopes between group 1 and group k.

3. For random effects, we don't use reference level coding since each level of S is random. Specify X and Z in the model

$$y = X\beta + Zb + e$$

   with $e \sim N(0, \sigma_e^2)$ and $b \sim N(0, G)$ so that each level of S has its own random slope and intercept.

In this case, we need to encode that

$$E\mathbf{y} = \beta_0 + \beta_1$$

in X which is therefore $[\mathbf{1}, \mathbf{t}]$ if $\mathbf{t}$ conveys the observation times.

For Z we need $[X^S, tX^S]$ where X^S is the subject indicator.

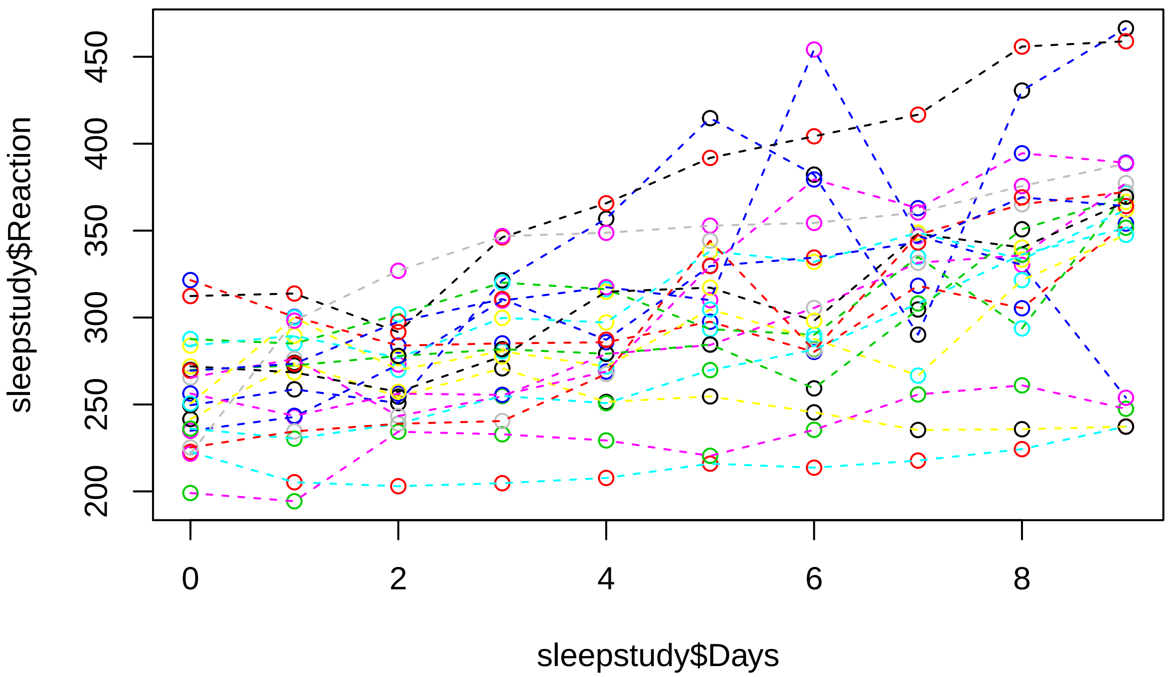
4. This model applies to data on reaction times in the data `sleepstudy` from the package `lme4`.

```
library(lme4)
data(sleepstudy)
```

Here we have 18 subjects coded in `Subject`, they were each restricted to 3 hours sleep per night for 10 nights and their reaction time (`Reaction`) measured on each of 10 days (`Day`). Here we assume that each suject's reaction time increases with longer sleep deprivation (ie, each has their own linear regression) but that this relationship is random between subjects.

a. We can see this more clearly by plotting the data. Connecting the values for each sbuject with lines

```
plot(sleepstudy$Days,sleepstudy$Reaction,col=sleepstudy$Subject)
for(j in levels(sleepstudy$Subject)){
  which.obs = (sleepstudy$Subject == j)   # These rows correspond to subject j
  lines(sleepstudy[which.obs,c(2,1)],col=j,lty=2)       # Plot day against time for these rows
}
```



b. This model can be fit in `lme4` as follows

```
longmod = lmer(Reaction~Days+(Days|Subject),data=sleepstudy)
summary(longmod)
```

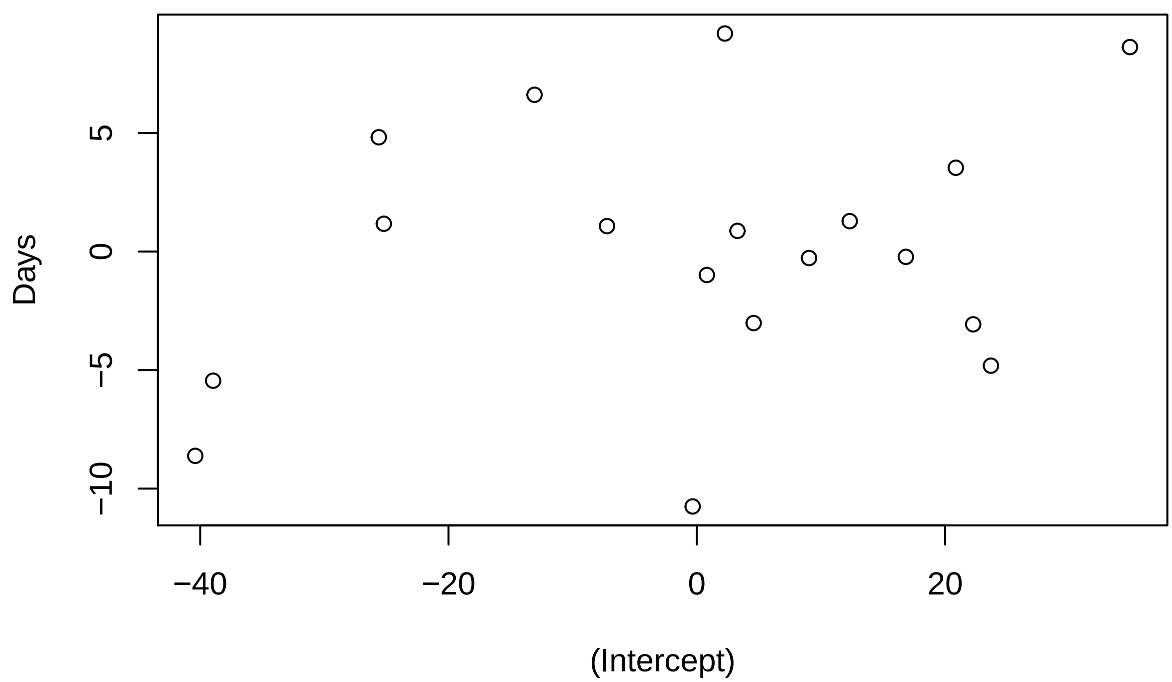
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9536 -0.4634  0.0231  0.4634  5.1793
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 612.09   24.740
##           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
```

Notice that specifying `(Days|Subject)` provides both an intercept and a slope for each subject.

c. We can extract and plot the random effects with

```
rand.effects = ranef(longmod)
plot(rand.effects$Subject)
```



d. By the default, the model allows there to be a correlation between each subjects slope and intercept. The (very small) positive correlation estimated here says that a subject with a longer starting reaction time is also affected more by sleep deprivation.

That correlation is pretty small, though. We can tell `lme4` not to allow it with the following

```
longmod2 = lmer(Reaction~Days + (1|Subject)+(0+Days|Subject),data=sleepstudy)
summary(longmod2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days + (1 | Subject) + (0 + Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1743.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.9626 -0.4625  0.0204  0.4653  5.1860
##
```

```
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  Subject   (Intercept) 627.57   25.051
##  Subject.1 Days         35.86    5.988
##  Residual              653.58   25.565
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  251.405      6.885  36.513
## Days          10.467      1.560   6.712
##
## Correlation of Fixed Effects:
##      (Intr)
## Days -0.184
```

Here we have to specify `(0+Days|Subject)` so that the second random term doesn't include an intercept. You can see that we no longer have an estimated correlation.

e. (Beyond course material). Are these models different? We can compare their likelihoods in a formal test (see BTRY/STSCI 4090) using

```
anova(longmod,longmod2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: sleepstudy
## Models:
## longmod2: Reaction ~ Days + (1 | Subject) + (0 + Days | Subject)
## longmod: Reaction ~ Days + (Days | Subject)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## longmod2  5 1762.0 1778.0 -876.00   1752.0
## longmod   6 1763.9 1783.1 -875.97   1751.9 0.0639      1     0.8004
```

5. In the fully study, there was a group that got 6 hours sleep instead of 3. How would you set up a model to test whether the *average* slope in the second group was lower than that in the first?

We can do this artificially by making the first 9 subjects group 1 and the second 9 group 2

```
sleepstudy$Group = rep(c(0,1),1,each=90)
longmod3 = lmer(Reaction~Days*Group+(Days|Subject),data=sleepstudy)
summary(longmod3)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Reaction ~ Days * Group + (Days | Subject)
##    Data: sleepstudy
##
## REML criterion at convergence: 1728.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8401 -0.4660  0.0439  0.4836  5.2190
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  Subject  (Intercept) 663.6    25.760
##           Days         27.1     5.205   0.08
##  Residual             654.9    25.592
```

```
## Number of obs: 180, groups:  Subject, 18
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  252.292      9.943  25.373
## Days           7.388      1.973   3.745
## Group         -1.773     14.062  -0.126
## Days:Group     6.158      2.790   2.207
##
## Correlation of Fixed Effects:
##            (Intr) Days   Group
## Days       -0.140
## Group      -0.707  0.099
## Days:Group  0.099 -0.707 -0.140
```

The t-statistic then gives us a test, or we can check

```r
anova(longmod3,longmod)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: sleepstudy
## Models:
## longmod: Reaction ~ Days + (Days | Subject)
## longmod3: Reaction ~ Days * Group + (Days | Subject)
##          Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## longmod   6 1763.9 1783.1 -875.97   1751.9
## longmod3  8 1763.1 1788.7 -873.56   1747.1 4.8124      2    0.09016 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```