lme4

Luke Chang

Last Revised July 16, 2010

1 Using lme4

1.1 Fitting Linear Mixed Models with a Varying Intercept

We will now work through the same Ultimatum Game example from the regression section and the introduction using the lme4 package. The lme4 package is unique in that it allows for correlated random variance structures and also allows for crossed random factors, which makes it particularly suited for analyzing psychology datasets. First, it is necessary to load the package and a data file.

```
> library(lme4)
> data <- read.table(paste(website, "UG_Data.txt",
        sep = ""), header = TRUE, na.strings = 999999)
> data$Condition <- relevel(data$Condition, ref = "Computer")
```

Next we can build the first model. Remember that we are interested in examining the effect of the amount of money Offered on Reaction Time. We can specify that `Subject` is a random effect with a varying intercept with the `(1|Subject)` term

```
> m1 <- lmer(RT ~ Offer + (1 | Subject), data = data)
> summary(m1)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10901 10919  -5446    10913   10893
```

```
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept)  241013   490.93
 Residual             1139459  1067.45
Number of obs: 648, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  2260.71     145.67  15.519
Offer        -125.17      28.35  -4.415

Correlation of Fixed Effects:
      (Intr)
Offer -0.535
```

Consistent with both previous examples, we see that offer amount is negatively related to reaction time. We see that the random effect has some variability, which represents variability in the participant's intercepts. We can easily examine the random effects by using the `ranef()` or `coef()` commands. These values are referred to as Best Linear Unbiased Predictors (BLUPs) by the developers of the `lme4` package.

```
> coef(m1)

$Subject
    (Intercept)     Offer
212    1792.293 -125.1725
213    1948.706 -125.1725
214    1338.696 -125.1725
215    2936.108 -125.1725
216    2209.936 -125.1725
217    2936.795 -125.1725
218    1781.708 -125.1725
301    2041.808 -125.1725
302    2599.778 -125.1725
303    2358.933 -125.1725
304    2240.708 -125.1725
309    2416.817 -125.1725
310    1667.462 -125.1725
311    2423.620 -125.1725
```

```
405     2143.308 -125.1725
406     2484.893 -125.1725
407     2291.863 -125.1725
408     3079.382 -125.1725
```

We can see that these intercept values are very similar to the ones we fit in the introductory example, with a correlation of 0.94.

```
> cor(ranef(m1)$Subject, dat$Intercept)
```

```
                 [,1]
(Intercept) 0.9409049
```

1.2 Fitting Linear Mixed Models with a Varying Slope

The previous example examined a varying intercept model. This section will fit a varying slope model on the same dataset. To do this we tell `lmer` that we do not want a varying intercept, but we do want varying slope of Offer amount for each subject `(0+Offer|Subject)`.

```
> m2 <- lmer(RT ~ Offer + (0 + Offer | Subject),
        data = data)
> summary(m1)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10901 10919  -5446    10913   10893
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept)  241013   490.93
 Residual             1139459  1067.45
Number of obs: 648, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  2260.71     145.67  15.519
Offer        -125.17      28.35  -4.415
```

```
Correlation of Fixed Effects:
      (Intr)
Offer -0.535
```

We can examine the fitted coefficients the same way using the `coeff()` command.

```
> coef(m2)
```

```
$Subject
    (Intercept)      Offer
212    2260.712 -219.53839
213    2260.712 -209.93628
214    2260.712 -346.81506
215    2260.712   20.35713
216    2260.712 -144.58262
217    2260.712   43.64018
218    2260.712 -233.25636
301    2260.712 -173.84963
302    2260.712  -63.40676
303    2260.712 -113.61648
304    2260.712  -96.31508
309    2260.712  -82.46747
310    2260.712 -270.37776
311    2260.712  -58.52931
405    2260.712 -163.82389
406    2260.712  -88.63200
407    2260.712 -138.42735
408    2260.712   86.47238
```

Mixed models can be compared the same way as the linear models using the `anova()` function. However, now rather than F tests, `anova()` calculates a χ^2.

```
> anova(m1, m2)
```

```
Data: data
Models:
m1: RT ~ Offer + (1 | Subject)
m2: RT ~ Offer + (0 + Offer | Subject)
   Df   AIC   BIC  logLik Chisq Chi Df Pr(>Chisq)
m1  4 10921 10939 -5456.4
m2  4 10960 10978 -5476.3     0      0  < 2.2e-16 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The χ^2 test indicates that the first model fits the data significantly better than the second model.

1.3 Fitting Linear Mixed Models with a Varying Intercept and Slope

We can also fit a varying intercept and slope model to the data. We simply need to tell `lmer` that we want both a varying intercept and varying slopes of Offer amount for each subject `(1+Offer|Subject)`.

```
> m3 <- lmer(RT ~ Offer + (1 + Offer | Subject),
        data = data)
> summary(m3)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer + (1 + Offer | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10901 10928  -5445    10910   10889
Random effects:
 Groups   Name        Variance  Std.Dev. Corr
 Subject  (Intercept)  410745.0  640.894
          Offer          2950.9   54.323 -1.000
 Residual             1132799.6 1064.331
Number of obs: 648, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  2260.71     174.96  12.921
Offer        -125.17      31.04  -4.033

Correlation of Fixed Effects:
      (Intr)
Offer -0.761
```

Again we can check the BLUPs using the `coef()` command. We can also compare this model with the varying intercept model.

```
> coef(m3)

$Subject
    (Intercept)      Offer
212    1623.062  -71.12478
213    1868.758  -91.95015
214    1059.288  -23.33889
215    3165.954 -201.90148
216    2205.325 -120.47787
217    3132.268 -199.04620
218    1625.984  -71.37248
301    1969.570 -100.49504
302    2732.062 -165.12447
303    2406.731 -137.54917
304    2184.283 -118.69434
309    2456.376 -141.75710
310    1491.564  -59.97891
311    2431.877 -139.68054
405    2123.331 -113.52798
406    2578.784 -152.13253
407    2332.334 -131.24319
408    3305.266 -213.70962

> anova(m1, m3)

Data: data
Models:
m1: RT ~ Offer + (1 | Subject)
m3: RT ~ Offer + (1 + Offer | Subject)
   Df   AIC   BIC  logLik Chisq Chi Df Pr(>Chisq)
m1  4 10921 10939 -5456.4
m3  6 10922 10948 -5454.8 3.247      2     0.1972
```

We see that the varying intercept and slope model does not fit the data any better than the simplest varying intercept model, so we will proceed with model 1. We can now continue with the example using identical models to the regression section except this time in the context of mixed models.

```
> m4 <- lmer(RT ~ Offer + I(Offer^2) + (1 | Subject),
        data = data)
> summary(m4)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer + I(Offer^2) + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10870 10892  -5430    10888   10860
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept)  242175   492.11
 Residual             1097608  1047.67
Number of obs: 648, groups: Subject, 18

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1431.87     220.21   6.502
Offer         587.64     145.30   4.044
I(Offer^2)   -116.04      23.22  -4.998

Correlation of Fixed Effects:
           (Intr) Offer
Offer      -0.806
I(Offer^2)  0.753 -0.981
```

```
> m5 <- lmer(RT ~ Offer * Condition + I(Offer^2) +
        (1 | Subject), data = data)
> summary(m5)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer * Condition + I(Offer^2) + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10852 10883  -5419    10887   10838
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept)  242149   492.09
 Residual             1098577  1048.13
Number of obs: 648, groups: Subject, 18
```

```
Fixed effects:
                     Estimate Std. Error t value
(Intercept)           1577.55     252.26   6.254
Offer                  542.73     150.60   3.604
ConditionHuman        -218.53     184.40  -1.185
I(Offer^2)            -116.04      23.23  -4.996
Offer:ConditionHuman    67.36      59.06   1.141

Correlation of Fixed Effects:
            (Intr) Offer  CndtnH I(O^2)
Offer       -0.791
ConditinHmn -0.487  0.230
I(Offer^2)   0.658 -0.947  0.000
Offr:CndtnH  0.429 -0.261 -0.881  0.000
```

```
> m6 <- lmer(RT ~ Offer * Condition + I(Offer^2) *
        Condition + (1 | Subject), data = data)
> summary(m6)
```

```
Linear mixed model fit by REML
Formula: RT ~ Offer * Condition + I(Offer^2) * Condition + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance REMLdev
 10839 10874  -5411    10881   10823
Random effects:
 Groups   Name        Variance Std.Dev.
 Subject  (Intercept)  242371   492.31
 Residual             1090571  1044.30
Number of obs: 648, groups: Subject, 18

Fixed effects:
                          Estimate Std. Error t value
(Intercept)                2130.50     343.40   6.204
Offer                        67.19     250.85   0.268
ConditionHuman            -1047.95     395.85  -2.647
I(Offer^2)                  -38.63      40.08  -0.964
Offer:ConditionHuman        780.66     307.23   2.541
ConditionHuman:I(Offer^2)  -116.12      49.09  -2.365
```

```
Correlation of Fixed Effects:
            (Intr) Offer  CndtnH I(O^2) Off:CH
Offer       -0.892
ConditinHmn -0.768  0.774
I(Offer^2)   0.834 -0.981 -0.723
Offr:CndtnH  0.728 -0.816 -0.948  0.801
CndH:I(O^2) -0.681  0.801  0.886 -0.816 -0.981
```

```
> anova(m1, m4, m5, m6)
```

```
Data: data
Models:
m1: RT ~ Offer + (1 | Subject)
m4: RT ~ Offer + I(Offer^2) + (1 | Subject)
m5: RT ~ Offer * Condition + I(Offer^2) + (1 | Subject)
m6: RT ~ Offer * Condition + I(Offer^2) * Condition + (1 | Subject)
   Df   AIC   BIC  logLik   Chisq Chi Df Pr(>Chisq)
m1  4 10921 10939 -5456.4
m4  5 10898 10920 -5444.1 24.5790      1  7.132e-07 ***
m5  7 10901 10932 -5443.4  1.4566      2    0.48272
m6  8 10897 10933 -5440.6  5.6165      1    0.01779 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

After completing the exercise, we can conclude that the results we reported earlier in the regression section were accurate, even when modeling for the repeated structure of the data.

1.4 p-Values

You may have noticed that there are no p-values associated with the parameter estimates from the model output [1]. While the lme4 package does provide t values, the authors have admitted to not knowing how to calculate exact values and are perplexed as to how to best approximate the degrees of freedom in a mixed model framework, particularly with unbalanced designs and correlated random factors. In SAS there

[1]For a more detailed discussion of this problem see Baayen, Davidson, and Bates (2008) in Journal of Memory and Language

are apparently 6 different df approximations, which lead to different p-values. For example, how does one even go about calculating the number of parameters for a mixed model? In the simple model we used in the example, there are 6, fixed effects values 1 random effect, and 1 value for the variance of the error term. But what about the 18 parameters that were calculated for each participant's intercept? The authors here have chosen to abstain from providing p-values, until they have developed a more accurate method with which they are more comfortable. Unfortunately, most of us work in areas where providing p-values is still customary, which makes this particularly frustrating. However, do not be discouraged there are two approaches which can be taken, both of which, unfortunately, suffer from their own respective problems. First, you can use the t value reported and approximate the degrees of freedom by subtracting `the number of observations - the number of fixed effects parameters - 1`. This is the approach typically taken in standard linear models and happens to be the strategy adopted by SPSS, however, it is likely anticonservative, particularly when the sample size is small. An alternative approach is to use markov chain monte carlo simulations on the parameter estimates and calculate the p-values based on the confidence intervals of the empirically observed distributions. This can be accomplished using the `mcmcsamp()` function included in the `lme4` package. This approach has been made easier with the `pval.fnc()` from the `languageR` package. We can use this approach to examine the p-values on our best fitting model.

```
> library(languageR)

> pvals.fnc(m6)

$fixed
                          Estimate MCMCmean HPD95lower
(Intercept)                2130.50  2132.28     1477.0
Offer                        67.19    65.44     -412.8
ConditionHuman            -1047.95 -1050.16    -1827.4
I(Offer^2)                  -38.63   -38.31     -117.7
Offer:ConditionHuman        780.66   782.41      156.2
ConditionHuman:I(Offer^2)  -116.12  -116.42     -210.9
                          HPD95upper  pMCMC Pr(>|t|)
(Intercept)                  2806.63 0.0001   0.0000
Offer                         561.28 0.8016   0.7889
ConditionHuman               -286.18 0.0062   0.0083
I(Offer^2)                     37.17 0.3396   0.3356
Offer:ConditionHuman         1357.06 0.0088   0.0113
```

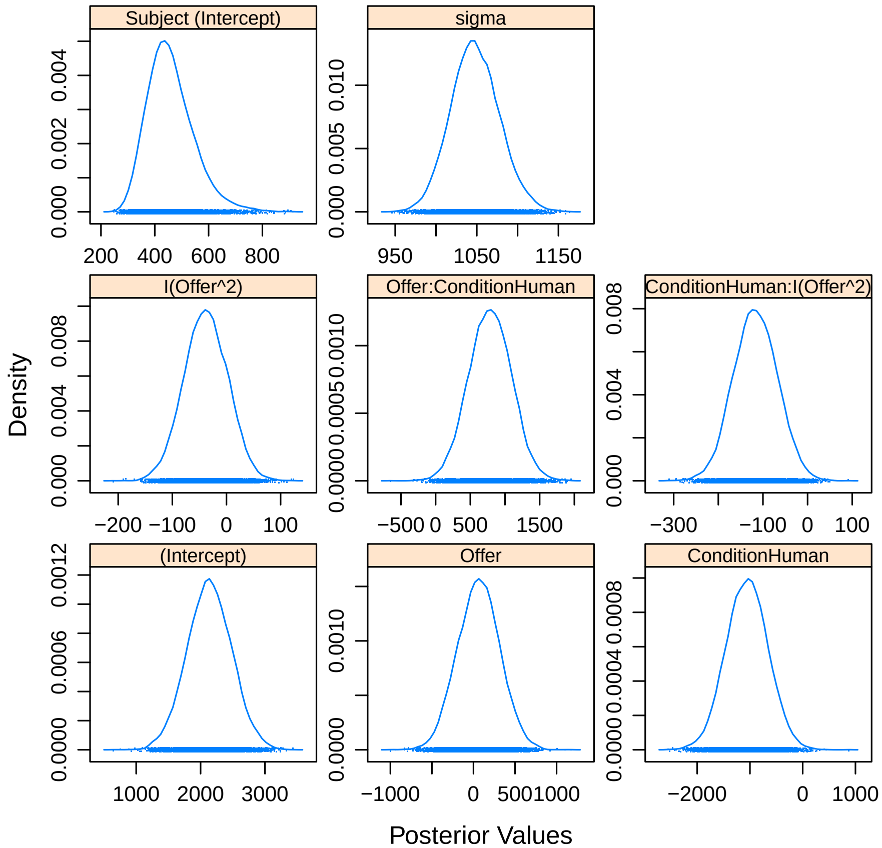

Figure 1: Output from pval.fnc()

```
ConditionHuman:I(Offer^2)     -20.44 0.0146   0.0183

$random
    Groups         Name  Std.Dev. MCMCmedian  MCMCmean
1  Subject (Intercept)  492.3118   450.4866  461.2617
2 Residual             1044.3039  1047.5423 1047.8945
  HPD95lower HPD95upper
1   306.7829   640.7244
2   991.9578  1108.2339
```

We see that the the MCMC derived p-values are slightly more conservative then those from a t distribution, but still fairly similar. This is a promising approach to calculating p-values for mixed models. However, it is important to note that at this time the method only works on models with varying intercepts. There is no current implementation for models with correlated random effects, such as those with varying intercepts and slopes.

1.5 Fitting Generalized Linear Mixed Models

We will conclude this section with a final example from the Ultimatum Game dataset. The previous examples have used linear mixed models to examine decision conflict. However, we are also interested in modeling the participant's actual decisions, which is a binary choice to either accept or reject the offer. Typically, psychologists would create the mean acceptance rate at each level of offer amount and then enter these data into a repeated measures ANOVA. However, this introduces a host of problems, which are discussed in greater detail by Jaeger (2008)[2]. The best analytic approach to this data is to use a mixed logit model to predict participants' decisions. We will use the `glmer()` function and additionally specify that our outcome data comes from a binomial distribution and that we should use a logit link function.

```
> m1 <- glmer(Decision ~ Offer + (1 | Subject),
        data = data, family = binomial(link = "logit"))
> summary(m1)
```

```
Generalized linear mixed model fit by the Laplace approximation
Formula: Decision ~ Offer + (1 | Subject)
   Data: data
   AIC   BIC logLik deviance
 363.3 376.7 -178.7    357.3
Random effects:
 Groups  Name        Variance Std.Dev.
 Subject (Intercept) 7.7136   2.7773
Number of obs: 642, groups: Subject, 18

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.0823     0.8443   -6.02 1.75e-09 ***
Offer         2.7107     0.2491   10.88  < 2e-16 ***
```

[2]This is published in the same special issue of Journal of Memory and Learning as the Baayen et al., 2008 paper

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
      (Intr)
Offer -0.601
```

This model indicates that the probability of accepting an offer increases as the amount of money increases. The parameter estimate for this factor is 2.71 and is in log odds, which is necessary as a result of the additive nature of a linear model. It can be quickly converted into an odds-ratio using the `exp()` command.

We will now fit a number of hypothesis driven models and determine the best model using our model comparison procedure. We'll spare you the boring details and will provide the best fitting model.

```
> m2 <- glmer(Decision ~ Offer + I(Offer^2) + (1 |
        Subject), data = data, family = binomial(link = "logit"))
> m3 <- glmer(Decision ~ Offer * Condition + I(Offer^2) +
        (1 | Subject), data = data, family = binomial(link = "logit"))
> m4 <- glmer(Decision ~ Offer * Condition + I(Offer^2) *
        Condition + (1 | Subject), data = data,
        family = binomial(link = "logit"))
> m5 <- glmer(Decision ~ Offer * Condition + I(Offer^2) *
        Condition + (I(Offer^2) | Subject), data = data,
        family = binomial(link = "logit"))
> anova(m1, m2, m3, m4, m5)
```

```
Data: data
Models:
m1: Decision ~ Offer + (1 | Subject)
m2: Decision ~ Offer + I(Offer^2) + (1 | Subject)
m3: Decision ~ Offer * Condition + I(Offer^2) + (1 | Subject)
m4: Decision ~ Offer * Condition + I(Offer^2) * Condition + (1 |
m4:     Subject)
m5: Decision ~ Offer * Condition + I(Offer^2) * Condition + (I(Offer^2) |
m5:     Subject)
   Df    AIC    BIC  logLik   Chisq Chi Df Pr(>Chisq)
m1  3 363.34 376.73 -178.67
m2  4 351.79 369.65 -171.90 13.5479      1  0.0002326 ***
```

```
m3  6 340.47 367.25 -164.23 15.3240      2  0.0004704 ***
m4  7 342.41 373.66 -164.21  0.0565      1  0.8121355
m5  9 315.02 355.20 -148.51 31.3931      2  1.524e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

> summary(m5)

```
Generalized linear mixed model fit by the Laplace approximation
Formula: Decision ~ Offer * Condition + I(Offer^2) * Condition
+ (I(Offer^2) |      Subject)
   Data: data
 AIC   BIC logLik deviance
 315 355.2 -148.5      297
Random effects:
 Groups  Name        Variance Std.Dev. Corr
 Subject (Intercept) 25.38311 5.03817
         I(Offer^2)   0.36267 0.60222  -0.750
Number of obs: 642, groups: Subject, 18

Fixed effects:
                           Estimate Std. Error z value
(Intercept)               -0.001534   2.355194  -0.001
Offer                     -3.416735   2.269656  -1.505
ConditionHuman            -2.996807   2.384292  -1.257
I(Offer^2)                 1.745145   0.616017   2.833
Offer:ConditionHuman       1.086708   2.627859   0.414
ConditionHuman:I(Offer^2) -0.109385   0.656495  -0.167
                          Pr(>|z|)
(Intercept)                0.99948
Offer                      0.13222
ConditionHuman             0.20879
I(Offer^2)                 0.00461 **
Offer:ConditionHuman       0.67922
ConditionHuman:I(Offer^2)  0.86767
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Correlation of Fixed Effects:
```

```
             (Intr) Offer  CndtnH I(O^2) Off:CH
Offer        -0.820
ConditinHmn -0.596  0.661
I(Offer^2)   0.643 -0.948 -0.576
Offr:CndtnH  0.604 -0.713 -0.966  0.646
CndH:I(O^2) -0.587  0.721  0.911 -0.676 -0.983
```

The best fitting model turns out to be the same best fitting model as the reaction time analyses, except treating the quadratic term as a varying slope. These findings indicate that participants exponentially increase their probability of accepting an offer as it increases in fairness.