Homework Assignment 2

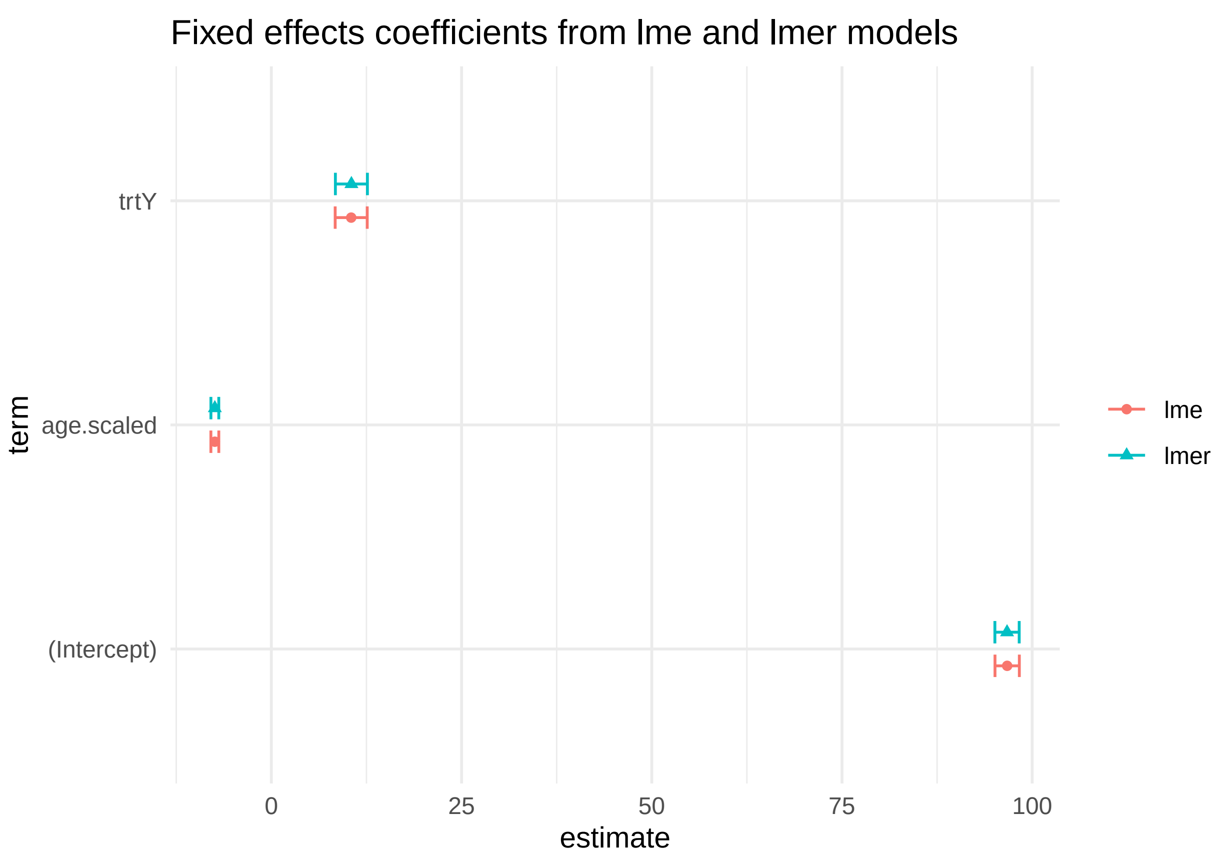
Greg Forkutza
Student ID: 400277514

07 November, 2023

a

Comparing the log-likelihoods, they are essentially equivalent (Mean relative difference: 0.0007723927), however `lmer.llk` > `lme.llk`.

b



We can see above above that the difference between `mod.lme` and `mod.lmer` for the estimated denominator degrees of freedom are different (tolerance > 0.1) for the `intercept` and `age`. They are very similar (10^{-4}

Table 1: Difference in Estimated Fixed Effects Coefficents for mod.lme vs. mod.lmer

term	fec_estimate	std_error
(Intercept)	Mean relative difference: 0.0001675275	Mean relative difference: 0.0005631294
age.scaled	Mean relative difference: 3.348062e-15	Mean relative difference: 0.008810136
trtY	Mean relative difference: 0.002741822	Mean relative difference: 0.001043202

Table 2: Estimated Fixed Effects with Wald CI's for mod.lme and mod.lmer

term	(Intercept)	age.scaled	trtY
estimate_lme	96.709120	-7.427879	10.493574
lower_limit_lme	93.574308	-8.449433	6.364412
upper_limit_lme	99.843932	-6.406325	14.622736
df_lme	205	205	101
estimate_lmer	96.692919	-7.427879	10.522346
lower_limit_lmer	93.559871	-8.440433	6.397492
upper_limit_lmer	99.825966	-6.415325	14.647200
df_lmer	103.6754	108.8338	101.0571

Table 3: Difference between mod.lme vs. mod.lmer for estimated denominator degrees of freedom and lower/upper limits for Wald CIs.

	Result
diff_df_intcpt	Mean relative difference: 0.4942664
diff_df_age	Mean relative difference: 0.4691034
diff_df_trt	Mean relative difference: 0.0005655288
diff_est_ll	Mean relative difference: 0.0005214151
diff_est_ul	Mean relative difference: 0.000425494

$<$ tolerance $< 10^{-2}$) for `trt`. The difference in the lower and upper limit for the Wald CI's are very similar.

C

Table 4: Comparison of estimated degrees of freedom for the lmer fit with the Satterthwaite vs. Kenward-Roger approximations.

	(Intercept)	age.scaled	trt
df_lmer_S	103.675394874025	108.833808095417	101.057118409738
df_lmer_KR	103.730963744733	102	100.999999999998
Differeance df S vs KR	Mean relative difference: 0.000535989	Mean relative difference: 0.06279122	Mean relative difference: 0.0005652092

The difference for `intercept` and `trt` are very similar. The difference for `age` is slightly different (0.01 $<$ tolerance $<$ 0.1). The main roll of the ddf is in determining the critical values from t-distributions for hypothesis tests and constructing confidence intervals. We can see the model summaries for the two methods of computing the ddf below:

```r
print((summary(mod.lmer, ddf="Satterthwaite"))[[10]])
```

```
##              Estimate Std. Error       df    t value     Pr(>|t|)
## (Intercept) 96.692919  1.5984934 103.6754  60.490034 1.099536e-82
## age.scaled  -7.427879  0.5166093 108.8338 -14.378137 6.517701e-27
## trtY        10.522346  2.1045175 101.0571   4.999885 2.419320e-06
```

```r
print((summary(mod.lmer, ddf = "Kenward-Roger"))[[10]])
```

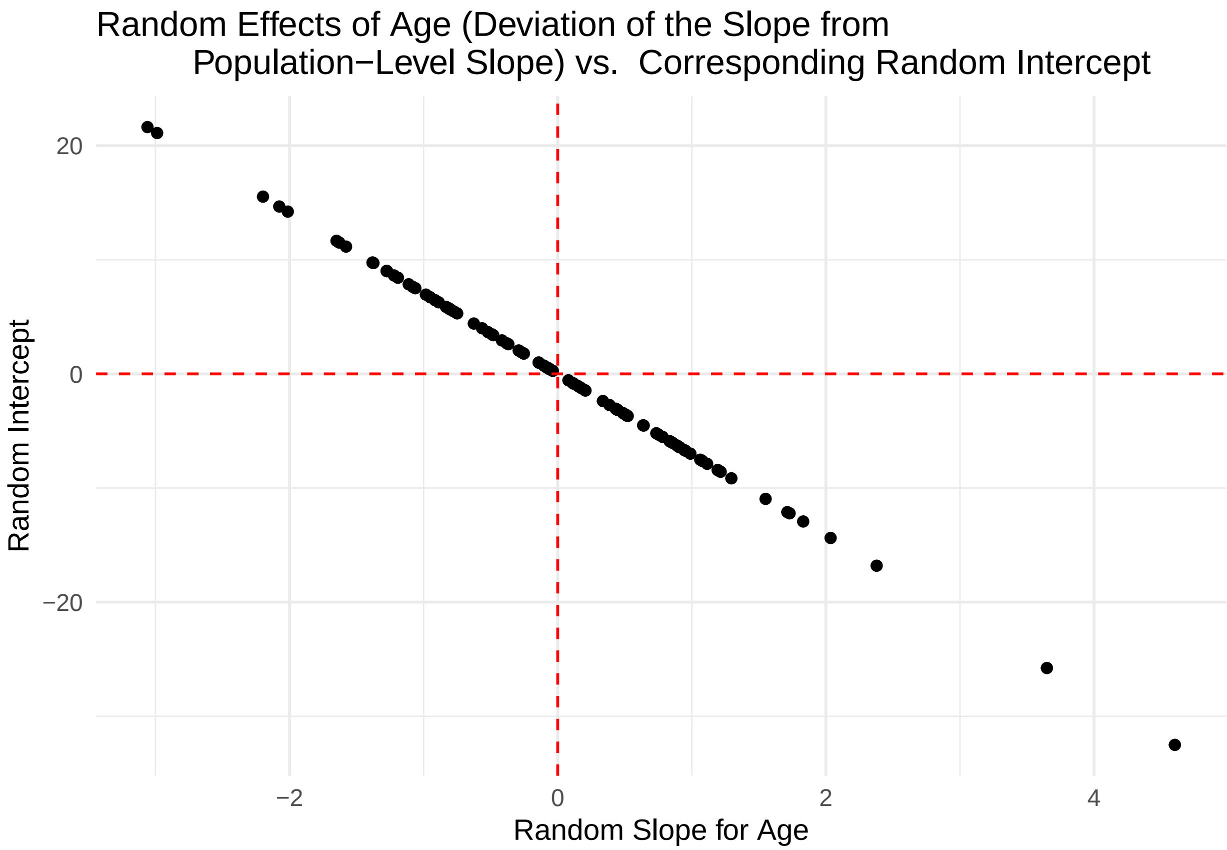
```
##              Estimate Std. Error      df    t value     Pr(>|t|)
## (Intercept) 96.692919  1.6071685 103.731  60.163521 1.764533e-82
## age.scaled  -7.427879  0.5166093 102.000 -14.378137 2.829141e-26
## trtY        10.522346  2.1252522 101.000   4.951105 2.963396e-06
```

We compute the Wald CI's below. We see that the difference in the width of the CI's for `intercept` and `trt` are slightly different. The difference in width for `age`is also slightly different but bordering on very similar. Given the magnitude of the parameter estimates it is not very important which method you choose for this example for computing the ddf.

Table 5: Comparison of Wald CI's based on the estimated degrees of freedom for the lmer fit with the Satterthwaite vs. Kenward-Roger approximations.

Term	Lower_95_CI_KR	Upper_95_CI_KR	Lower_95_CI_SW	Upper_95_CI_SW	Width_CI_KR	Width_CI_SW	Diff_Width
(Intercept)	93.505746	99.880091	93.522929	99.862908	6.374346	6.339978	0.0343673
age.scaled	-8.452571	-6.403187	-8.451799	-6.403959	2.049384	2.047841	0.0015434
trtY	6.306417	14.738274	6.347578	14.697114	8.431857	8.349536	0.0823212

d



e

`trt` is a treatment that is a applied to 58/103 infants. The treatment is deliberately applied to test its effect. It is not a naturally occurring variation in the population. This is a deliberate manipulation and not a random sample. If `trt` were a random effect, it would imply we were interested in generalizing the results to an entire population of potential treatments, which is not the case here asa we wish to generalize about the effect of the treatment itself. Therefore treatment should be considered a fixed effect because it allows for the estimation of the specific effect of the early childhood intervention program on the numeric cognitive score of infants.

f

Fitting a model like `cog ~ 1 +trt + (1 + age | id)` implies we are not including a fixed effect of age. There is no baseline or average effect of `age` on `cog` for the population. The random slope would instead represent random variation around an unspecified (probably implicitly zero?) average slope. It is unclear what the individual deviations are relative to. It also means that some of the variation due to fixed effect of age is being incorrectly attributed to the random effect of `age` and the fixed effect of `trt`. Therefore by not including a fixed effect for `age` we would be saying that there is no average effect of `age` on `cog`. This is a strongly counter intuitive statement about the nature of the relationship between age and cognitive ability.

g

By fitting a reduced model with random intercept only, we are testing the assumption that while the cognitive ability may start at different baselines for different individuals (captured by the random intercepts), it changes with age at the same rate across all individuals (since there's no random slope for age).

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

```
## Data: data
## Models:
## mod.reduced.int.only: cog ~ age.scaled + trt + (1 | id)
## mod.reduced.ind: cog ~ age.scaled + trt + (1 | id) + (0 + age.scaled | id)
## mod.lmer: cog ~ age.scaled + trt + (1 + age.scaled | id)
##                      npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## mod.reduced.int.only    5 2389.8 2408.5 -1189.9   2379.8
## mod.reduced.ind         6 2391.7 2414.1 -1189.9   2379.7 0.0943  1    0.75883
## mod.lmer                7 2388.3 2414.4 -1187.2   2374.3 5.4447  1    0.01963 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The comparison of `mod.reduced.ind` with `mod.reduced.int.only` yields a very high p value (0.75883), suggesting that the added independent random slopes for `age` does not significantly improve the model fit. The comparison of `mod.lmer` with `mod.reduced.ind` shows a significant improvement in fit with a very small p-value (0.01963), indicating that allowing random slopes for `age` to correlate with the intercepts significantly improves the model. `mod.lmer` also has the largest AIC value but with a very small increase of 0.3 over the independent model and an increase of 5.9 compared to the intercept only model. Therefore based on the above output, the model with the correlated random slopes and intercepts `mod.lmer` is the best fitting model according to both AIC and the LRT.

I tried to write my own function to do parametric bootstrapping but the p-value didn't agree with the results of `pbkrtest::PBmodcomp` below. I couldn't think of how to debug this. If there are any obvious issues I would love to know. You can run bootstrap1 and bootstrap2 to see but it is very slow. Also curious how you do this faster.

```
set.seed(123)

# Define function to implement parametric bootstrap for LRT
lrt.bootstrap <- function(mod.full, mod.reduced, data, response, n) {
  response <- deparse(substitute(response))
  bootstrap_lrt <- numeric(n)
  for (i in 1:n) {
    # Simulate data
    sim_data <- simulate(mod.full)

    # Update data with simulated response
    new_data <- data
    new_data[[response]] <- sim_data[[1]]

    # Fit both models to simulated data
    mod.full.boot <- update(mod.full, data = new_data)
    mod.reduced.boot <- update(mod.reduced, data = new_data)

    # Compute and store LRT
    bootstrap_lrt[i] <- 2 * (logLik(mod.full.boot) - logLik(mod.reduced.boot))
  }
#  Calculate the  original LRT statistic
original_lrt <- 2 * (logLik(mod.full) - logLik(mod.reduced))

# Compute p-value
p_value <- mean(bootstrap_lrt >= original_lrt)

# Define output as list
output <- list(bootstrap_lrt, original_lrt, p_value)

  return(output)
}

# Bootstrap of correlated model with independent intercept and slope model
bootstrap1 <- lrt.bootstrap(mod.lmer,
                                      mod.reduced.ind,
```

```r
                                        data = data,
                                        response = cog,
                                        n = 1000)

# Bootstrap of independent model and int only model
bootstrap2 <- lrt.bootstrap(mod.reduced.ind,
                                        mod.reduced.int.only,
                                        data = data,
                                        response = cog,
                                        n = 1000)
```

```r
PB1 <- PBmodcomp(mod.lmer, mod.reduced.ind, 1000)
PB2 <- PBmodcomp(mod.reduced.ind, mod.reduced.int.only, 1000)
print(PB1)
```

```
## Bootstrap test; time: 12.40 sec; samples: 1000; extremes: 16;
## large : cog ~ age.scaled + trt + (1 + age.scaled | id)
## cog ~ age.scaled + trt + (1 | id) + (0 + age.scaled | id)
##          stat df p.value
## LRT    5.4447  1 0.01963 *
## PBtest 5.4447    0.01698 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
print(PB2)
```

```
## Bootstrap test; time: 9.47 sec; samples: 1000; extremes: 350;
## Requested samples: 1000 Used samples: 513 Extremes: 350
## large : cog ~ age.scaled + trt + (1 | id) + (0 + age.scaled | id)
## cog ~ age.scaled + trt + (1 | id)
##          stat df p.value
## LRT    0.0943  1  0.7588
## PBtest 0.0943     0.6829
```

In the second comparison, the model with the independent slope/intercept is very close to a boundary condition where the variance of the random slope could be zero, indicated by a high number of extreme values (extremes = 351). This suggests a skewed or highly variable bootstrap test statistic distribution. This skewness implies that the random slopes contribute little additional variance. Standard LRTs assume a chi-squared distribution for the test statistic, which isn't valid at this boundary since variance cannot be negative. Consequently, the test statistic distribution becomes a skewed mixture, violating Wilk's Theorem. If the actual variance component is zero or close to it, the LRT statistic's distribution may deviate significantly from the chi-squared distribution, often resulting in a skewed, heavier-tailed distribution.

In the first comparison, both models include random slopes so they are less likely to be on the boundary of the parameter space where the random slope variance is zero.