Mixed Effects

Phil Shea

2023-03-09

Personally, I found most of the descriptions of the mixed effects models difficult to understand. This was pretty frustrating because I understood the theory, but not the words describing how R handles the theory. They are not wrong, but it seems that one had to understand how R dealt with the models already. Hopefully the examples below will help someone else too.

Simple Linear Models

The `Orthodont` data displays many problems common in analysis. The data frame has 108 rows and 4 columns of the change in an orthdontic measurement over time for 27 young subjects at ages of 8, 10, 12, & 14 years.

`lm` estimates *fixed values* (or fixed effects). These are unknown constants whose measurements are corrupted by noise, with an equation like $y = mx + b + \epsilon$, where x is the independent variable, m is the slope, b is the intercept, ϵ is measurement error (assumed normally distributed with zero mean), and y is the dependent variable. As we add independent variables, we add slopes and intercepts for each. The point is, there is one source of error (ϵ) and everything else is a constant to be estimated (or a *fixed effect*).

The original data is grouped in a way which solves many problems already, so a new data frame with just the raw data is created so we can solve the problems ourselves. Since the slope will be calculated at ages 8 to 14, an intercept at zero is a bit odd. The code below will change the `age` variable to be centered on 11 years.

```
O <- as.data.frame( Orthodont) #simplify data
O$Subject <- factor( O$Subject, ordered=FALSE) # remove order of factor.
O$age <- O$age - 11 # center the independent variable.
names(O)
```

```
## [1] "distance" "age"      "Subject"  "Sex"
```

```
flm1 <- lm( distance ~ age, data=O)
(sflm1 <- summary( flm1))
```

```
##
## Call:
## lm(formula = distance ~ age, data = O)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5037 -1.5778 -0.1833  1.3519  6.3167
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.0231     0.2441  98.400  < 2e-16 ***
## age           0.6602     0.1092   6.047 2.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.537 on 106 degrees of freedom
## Multiple R-squared:  0.2565, Adjusted R-squared:  0.2495
## F-statistic: 36.56 on 1 and 106 DF,  p-value: 2.248e-08
```

```
plot( O$age, O$distance, type='b', lty='dotted',
      pch=c( 1, 2)[O$Sex], col=c('blue', 'red')[O$Sex], xlab="Age - 11 years",
      ylab="mm", main="Orthodont data", sub="Distance vs. age")
abline( flm1, lwd=2)
legend( x="topleft", legend=c("boys", "girls"), col=c('blue', 'red'),
        pch=c( 1, 2) )
```

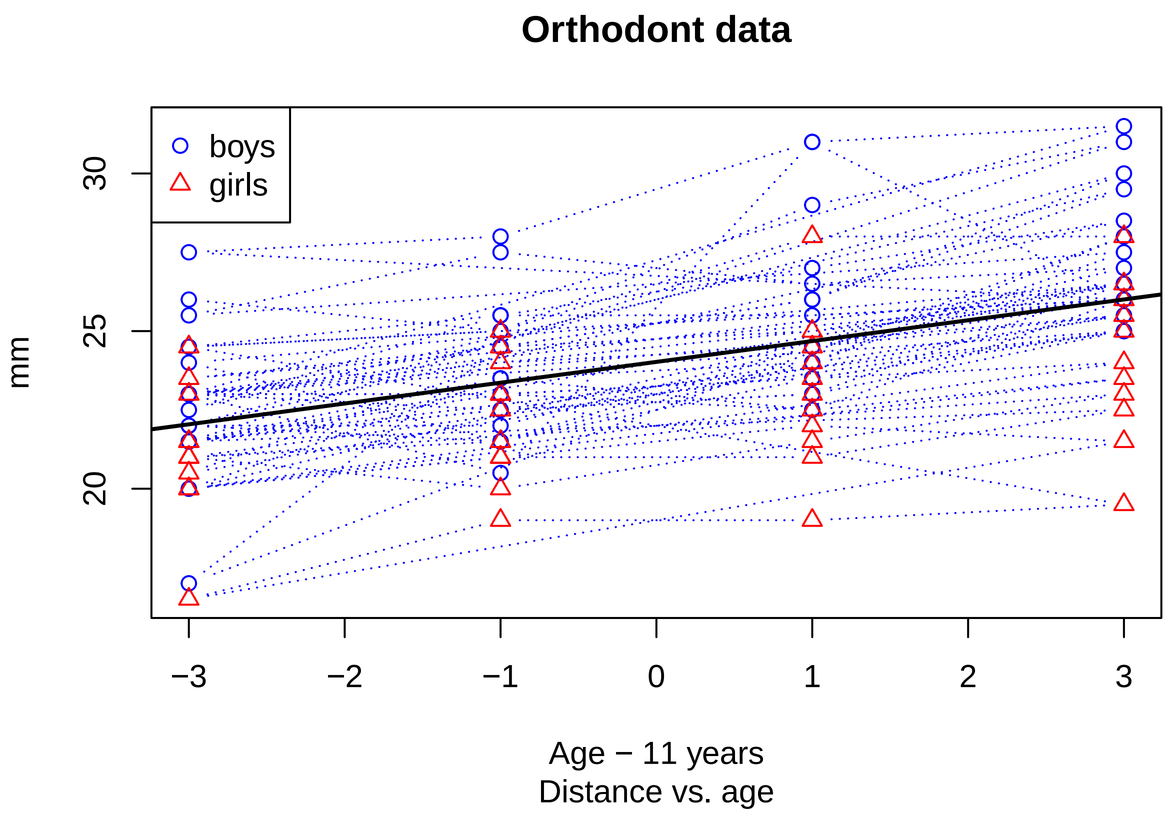


Figure 1: All data with single fit line.

Unsurprisingly, the girls are smaller. The line fits the whole dataset, but rather poorly. We can use `lmList` to run `lm` on each `Sex` independently (that is, as if there is nothing in common between the sexes).

```r
flm2 <- lmList( distance ~ age | Sex, data=O)
(sflm2 <- summary( flm2))
```

```
## Call:
##   Model: distance ~ age | Sex
##    Data: O
##
## Coefficients:
##    (Intercept)
##        Estimate Std. Error  t value      Pr(>|t|)
## Male   24.96875  0.2821186 88.50444 9.969844e-100
## Female 22.64773  0.3402478 66.56244  4.397790e-87
##    age
##         Estimate Std. Error  t value     Pr(>|t|)
## Male   0.7843750  0.1261673 6.216945 1.069216e-08
## Female 0.4795455  0.1521635 3.151515 2.122079e-03
##
## Residual standard error: 2.256949 on 104 degrees of freedom
```

```r
sflm2$adj.r.squared
```

```
## Male :
## [1] 0.3613751
##
## Female :
## [1] 0.1856477
```

```r
plot( augPred( flm2, primary=~ age))
```

This is closer to what we want, but it isn't clear that the slopes are different. The adjusted r-squared is better for the boys, but worse for the girls. Indeed, the slopes are within each others standard error:

```
## , , (Intercept)
##
##        Estimate Std. Error  t value      Pr(>|t|)
## Male   24.96875  0.2821186 88.50444 9.969844e-100
## Female 22.64773  0.3402478 66.56244  4.397790e-87
##
## , , age
##
##         Estimate Std. Error  t value     Pr(>|t|)
## Male   0.7843750  0.1261673 6.216945 1.069216e-08
## Female 0.4795455  0.1521635 3.151515 2.122079e-03
```

```
## Male Intr - 1 SD:  24.68663
```

```
## Female Intr + 1 SD:  22.98798
```

```
## Male slope - 1 SD:  0.6582077
```

```
## Female slope + 1 SD:  0.6317089
```

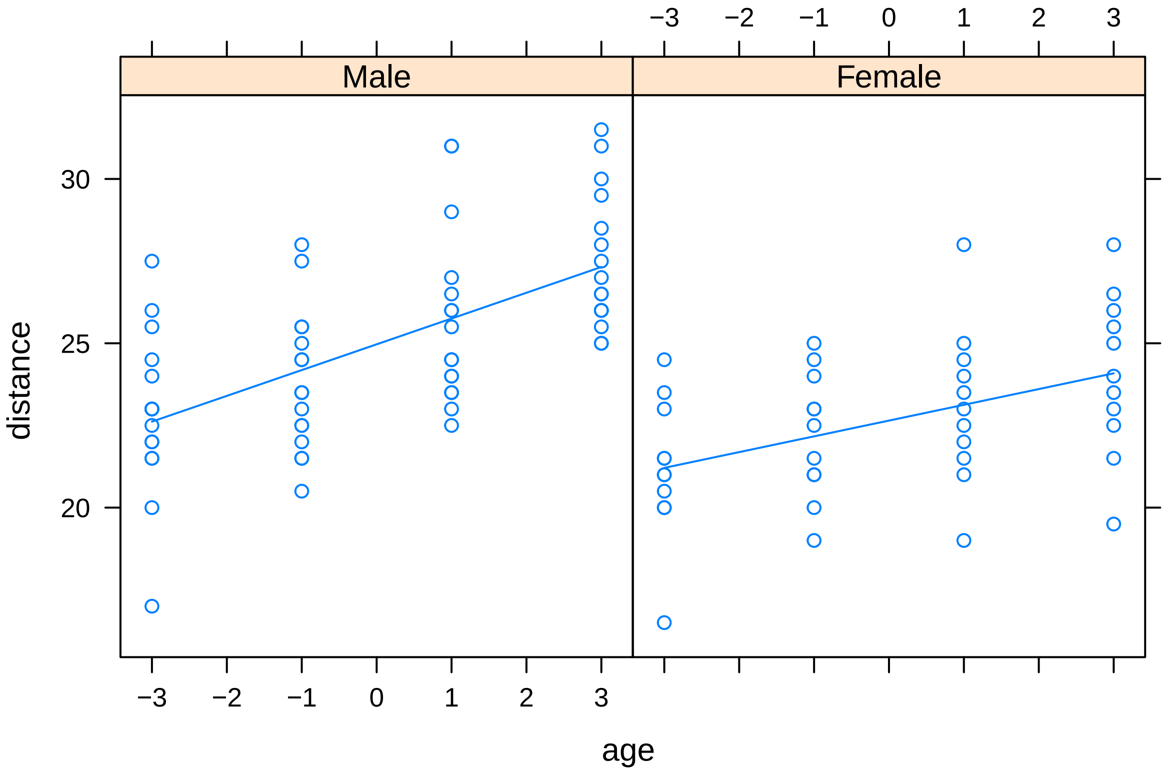


Figure 2: Individual fits by Sex.

This can be plotted easier (here with 90% confidence intervals):

```
plot( intervals( flm2))
```

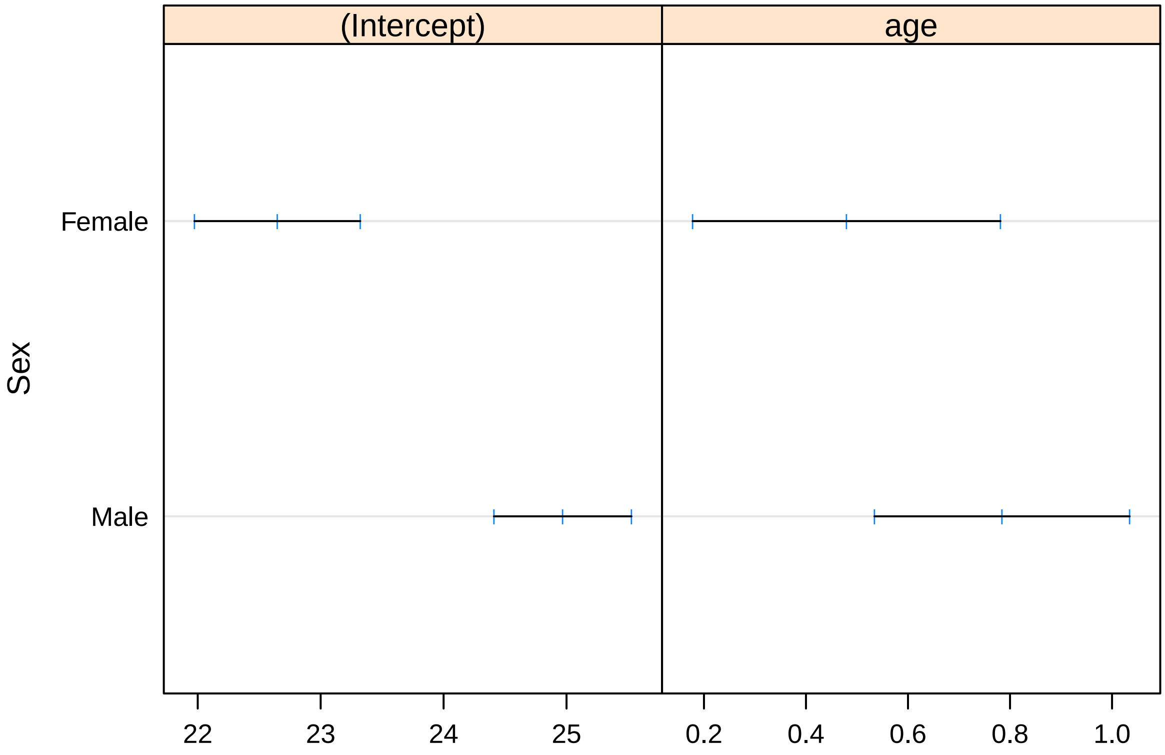


Figure 3: 90% Confidence intervals for flm2 parameters.

This indicates that the intercepts look different, and the slopes overlap.

The following will give an estimate of the difference in the girls' intercept by adding `Sex` to the equation:

```
flm3 <- lm( distance ~ age + Sex, data=O)
dummy.coef( flm3)
```

```
## Full coefficients are
##
## (Intercept):     24.96875
## age:            0.6601852
## Sex:                 Male    Female
##                  0.000000 -2.321023
```

```
(sflm3 <- summary( flm3))
```

```
##
## Call:
```

```
## lm(formula = distance ~ age + Sex, data = O)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9882 -1.4882 -0.0586  1.1916  5.3711
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.96875    0.28396  87.929  < 2e-16 ***
## age          0.66019    0.09776   6.753 8.25e-10 ***
## SexFemale   -2.32102    0.44489  -5.217 9.20e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.272 on 105 degrees of freedom
## Multiple R-squared:  0.4095, Adjusted R-squared:  0.3983
## F-statistic: 36.41 on 2 and 105 DF,  p-value: 9.726e-13
```

With this treatment, it indicates that the girls' intercept is -2.321 mm smaller than the boys', and that the distance increases 0.66 mm per year on average when estimated for both. Note that the rsquared increased from 0.25 to 0.4 using only one additional degree of freedom, so this is a much better fit. Perhaps the girls should be modeled with a different slope and intercept:

```
flm4 <- lm( distance ~ age * Sex, data=O)
(sflm4 <- summary( flm4))
```

```
##
## Call:
## lm(formula = distance ~ age * Sex, data = O)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.6156 -1.3219 -0.1682  1.3299  5.2469
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    24.9687     0.2821  88.504  < 2e-16 ***
## age             0.7844     0.1262   6.217 1.07e-08 ***
## SexFemale      -2.3210     0.4420  -5.251 8.05e-07 ***
## age:SexFemale  -0.3048     0.1977  -1.542    0.126
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.257 on 104 degrees of freedom
## Multiple R-squared:  0.4227, Adjusted R-squared:  0.4061
## F-statistic: 25.39 on 3 and 104 DF,  p-value: 2.108e-12
```

That shows that a seperate slope for girls is not statistically significant.

Mixed Effect Models

The issue with the above treatment is that the model is ignorant of the fact that each child is unique. While we clearly showed that the girls were smaller than the boys, a small boy's data is regressed with a larger boy’s data. Really, we ought to treat each child as a distinct experiment. Mixed effects models allow us to assume that some of the parameters may themselves be random variables. This makes more sense, as each child is unique, and we can consider that we don’t have the whole population of children, but a representative sample (27 in this data set).

Last Try with `lm` & `lmList`

We can try adding in the `Subject` to `lm` to see if this helps the situation.

```
flm5 <- lm( distance ~ age + Sex + Subject, data=O)
(sflm5 <- summary( flm5))
```

```
##
## Call:
## lm(formula = distance ~ age + Sex + Subject, data = O)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.2694 -0.7734 -0.0194  0.7870  5.2148
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.300e+01  7.158e-01  32.132  < 2e-16 ***
## age          6.602e-01  6.161e-02  10.716  < 2e-16 ***
## SexFemale    3.375e+00  1.012e+00   3.334 0.001299 **
## SubjectM05   3.666e-15  1.012e+00   0.000 1.000000
## SubjectM02   3.750e-01  1.012e+00   0.370 0.712028
## SubjectM11   6.250e-01  1.012e+00   0.617 0.538716
## SubjectM07   7.500e-01  1.012e+00   0.741 0.460926
## SubjectM08   8.750e-01  1.012e+00   0.864 0.389965
## SubjectM03   1.250e+00  1.012e+00   1.235 0.220510
## SubjectM12   1.250e+00  1.012e+00   1.235 0.220510
## SubjectM13   1.250e+00  1.012e+00   1.235 0.220510
## SubjectM14   1.875e+00  1.012e+00   1.852 0.067680 .
## SubjectM09   2.125e+00  1.012e+00   2.099 0.038952 *
## SubjectM15   2.875e+00  1.012e+00   2.840 0.005717 **
## SubjectM06   3.375e+00  1.012e+00   3.334 0.001299 **
## SubjectM04   3.625e+00  1.012e+00   3.581 0.000586 ***
## SubjectM01   4.750e+00  1.012e+00   4.692 1.10e-05 ***
## SubjectM10   6.500e+00  1.012e+00   6.421 8.97e-09 ***
## SubjectF10  -7.875e+00  1.012e+00  -7.779 2.18e-11 ***
## SubjectF09  -5.250e+00  1.012e+00  -5.186 1.58e-06 ***
## SubjectF06  -5.250e+00  1.012e+00  -5.186 1.58e-06 ***
## SubjectF01  -5.000e+00  1.012e+00  -4.939 4.22e-06 ***
## SubjectF05  -3.750e+00  1.012e+00  -3.704 0.000388 ***
## SubjectF07  -3.375e+00  1.012e+00  -3.334 0.001299 **
## SubjectF02  -3.375e+00  1.012e+00  -3.334 0.001299 **
## SubjectF08  -3.000e+00  1.012e+00  -2.964 0.004004 **
## SubjectF03  -2.625e+00  1.012e+00  -2.593 0.011305 *
## SubjectF04  -1.500e+00  1.012e+00  -1.482 0.142324
## SubjectF11          NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
##
## Residual standard error: 1.432 on 80 degrees of freedom
## Multiple R-squared:  0.8213, Adjusted R-squared:  0.761
## F-statistic: 13.62 on 27 and 80 DF,  p-value: < 2.2e-16
```

Unsurprisingly, only a subset of subjects are statistically significant. R-squared looks improved, but this result is clearly nonsense. It is difficult to get a good plot of these intervals as there is no `intervals` method for `lm`, but the below is at least partially satisfactory. (Yes, this is sloppy, but I spent too much time just getting here, and this is not the point of this paper.)

```
tmp <- confint( flm5)
tmp <- na.omit( tmp[ grep( "Subject", rownames( tmp)), ])
tmp <- tmp[ order( tmp[ ,1]), ]
barplot( height=tmp[ ,2]-tmp[,1], offset=tmp[, 1], horiz=TRUE,
         xlim=c( min( tmp[ , 1]), max( tmp[ , 2])),
         panel.first=grid())
```

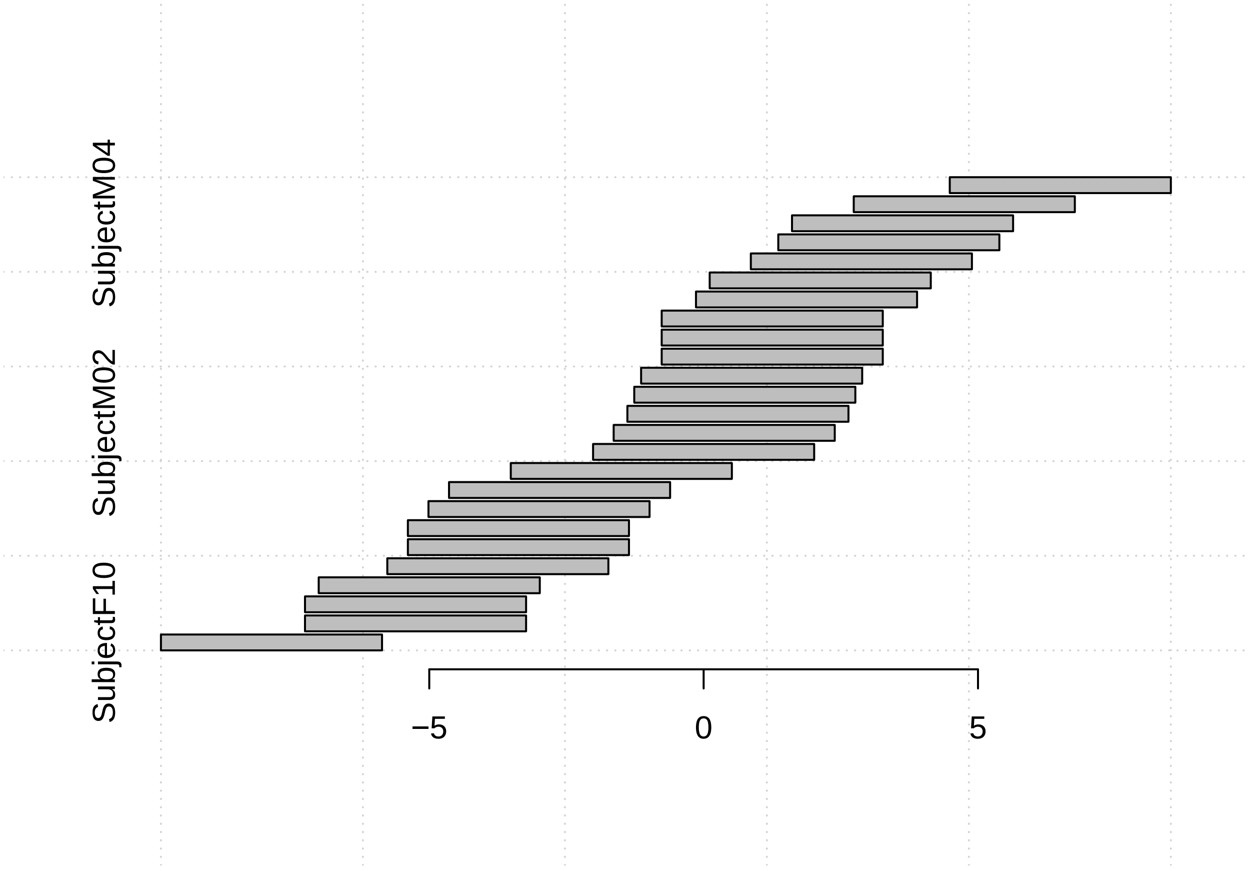


Figure 4: flm5 Subject 95% conf intervals

Note that all it did was find a distinct intercept for most subjects (as there are two other intercepts, the overall `(Intercept)` and `SexFemale`, subject `F11` did not get an intercept).

The following will simply fit each subject.

```
flm6 <- lmList( distance ~ age  | Subject, data=O)
#(sflm6 <- summary( flm6))
plot( augPred( flm6, primary=~ age))
```

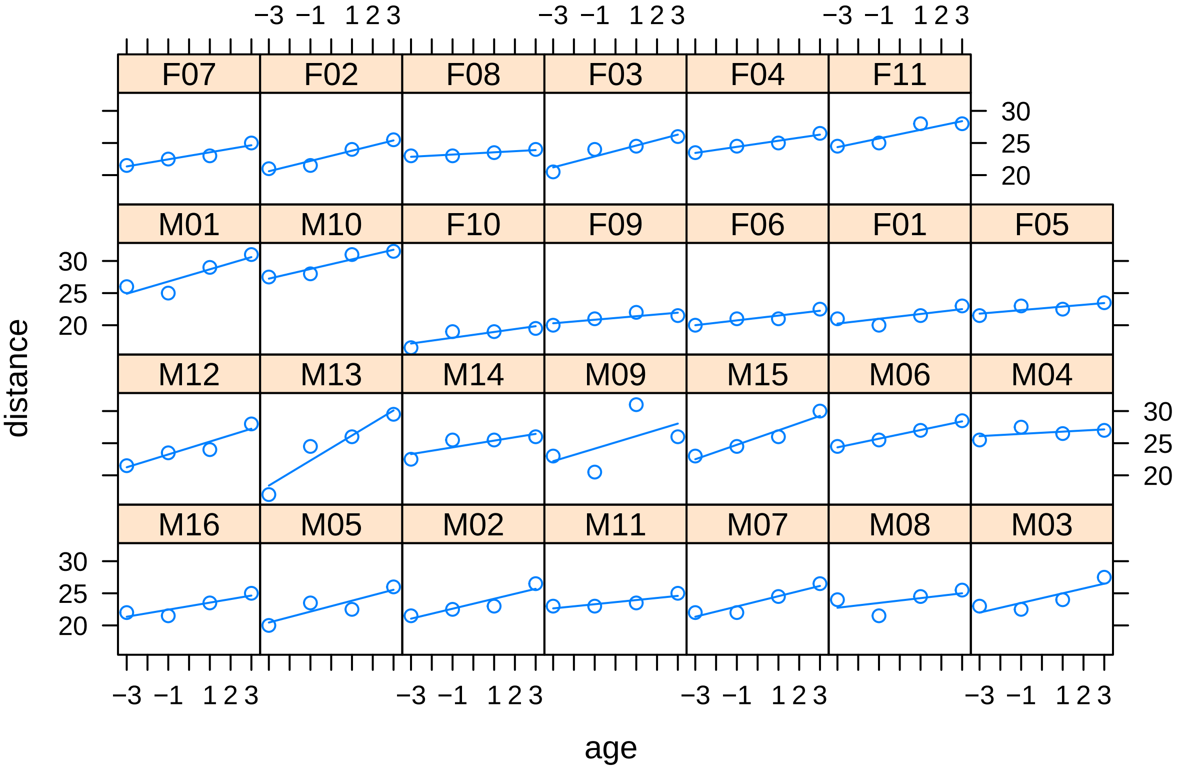


Figure 5: Subject by Subject fits.

Each fit looks pretty good, but this is not useful. Perhaps it gives a range of equation parameters, but it does not give us any statistics on boys vs. girls. We can look at the boys vs girls via an `intervals` plot (available for `lmList`).

```
plot( intervals( flm6))
```

So while the boys and girls appear quite different (and hooray for that), the ranges cover each other.

Using Mixed Effects

Mixed effect will allow us to estimate the differences between boys and girls (thank heaven), while still modelling the individual growth. The mixed effects model assumes that some of the things we are estimating are themselves random variables. A *random effect* formula is then a specification of which model parameter is actually a random variable. The model will be something like this (1):

$$y_{i,j} = (m + m_j)x_{i,j} + b + b_j + \epsilon \tag{1}$$

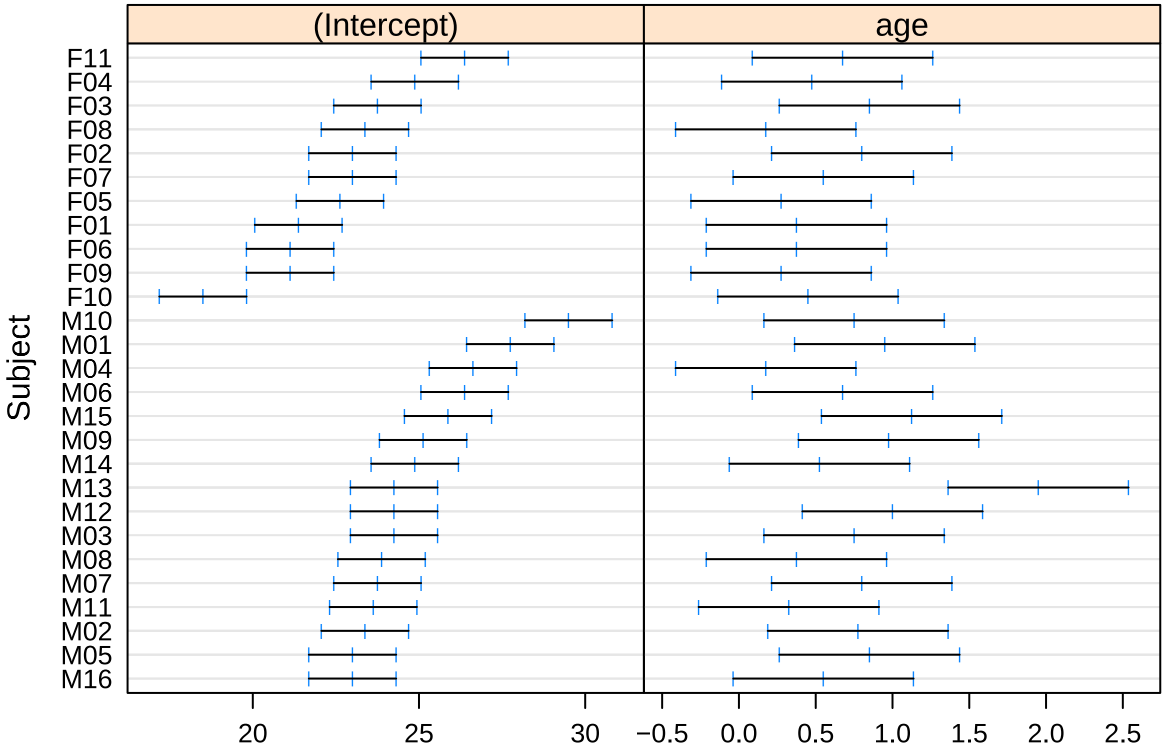


Figure 6: flm6 (lmList) intervals

where the jth subset has "random" slope and intercept, modeled by `lme` as normal distributions with zero mean: these are perturbations to the overall mean and intercept (the *fixed effects*). In our example, we could have a random sample for `Sex`, for `Subject`, and we can even combine `Sex` and `Subject`.

```
fme1 <- lme(distance ~ age, random= ~ age | Subject, data = O, method="ML")
# Use "ML" to compare to lm results.
(sfme1 <- summary( fme1))
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   451.2116 467.3044 -219.6058
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 2.0906362 (Intr)
## age         0.2149246 0.521
## Residual    1.3100396
##
## Fixed effects:  distance ~ age
##                 Value Std.Error DF  t-value p-value
## (Intercept) 24.023148 0.4255878 80 56.44699       0
## age          0.660185 0.0705779 80  9.35400       0
##  Correlation:
##     (Intr)
## age 0.294
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -3.305968026 -0.487429822  0.007598022  0.482236921  3.922790678
##
## Number of Observations: 108
## Number of Groups: 27
```

```
# we must tell augPred what variable drives the model. Level 0 is the overall
# fixed effect, level 1 the subjects.
plot( augPred( fme1, primary= ~ age, level=c(0,1)))
```

The above formula `distance ~ age` will estimate a slope and intercept based on `age`, and the specification `random = ~ age | Subject` says that each subject will have its own random variables for slope and intercept (reminder: `~ age` is equivalent to `~ age + 1`). In effect, the slope and intercept estimated is the mean slope and intercept, and the random effects will estimate the variance of these by subject. A close examination of the figure above will reveal that the `Subject` fits all have a different slope and intercept.

Is it a better fit than `lm` gave us? The coefficients are identical:

```
paste( " flm1: ")
```

```
## [1] " flm1: "
```

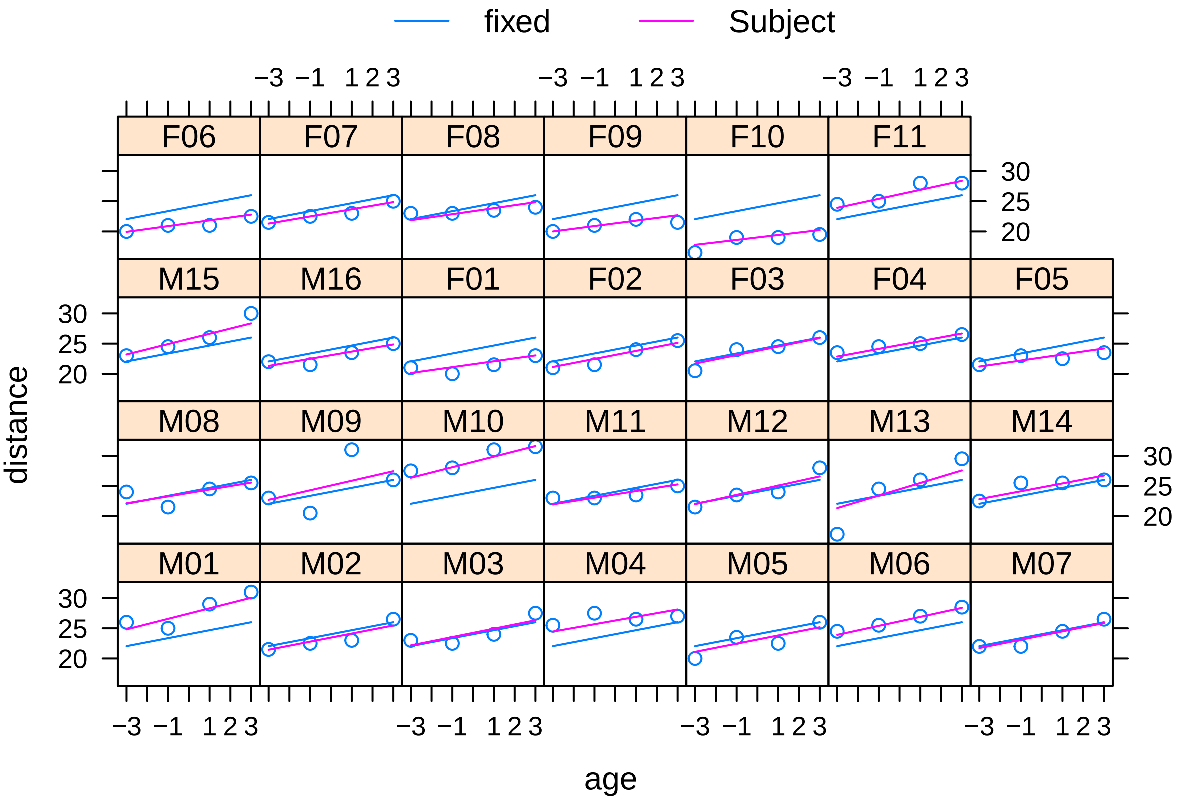


Figure 7: Regression on age alone, with random effects of slope and intercept by subject.

```
coef( flm1)
```

```
## (Intercept)         age
##  24.0231481   0.6601852
```

```
confint( flm1)
```

```
##                  2.5 %     97.5 %
## (Intercept) 23.5391219 24.5071743
## age          0.4437221  0.8766483
```

```
cat( "\n fme1: ")
```

```
##
##  fme1:
```

```
fixed.effects( fme1)
```

```
## (Intercept)         age
##  24.0231481   0.6601852
```

```
intervals( fme1)
```

```
## Approximate 95% confidence intervals
##
##  Fixed effects:
##                  lower       est.     upper
## (Intercept) 23.1840803 24.0231481 24.862216
## age          0.5210373  0.6601852  0.799333
##
##  Random Effects:
##   Level: Subject
##                            lower      est.     upper
## sd((Intercept))       1.55887191 2.0906362 2.8037966
## sd(age)               0.09290933 0.2149246 0.4971790
## cor((Intercept),age) -0.28997177 0.5205713 0.8962415
##
##  Within-group standard error:
##    lower     est.    upper
## 1.084860 1.310040 1.581959
```

So the intercept estimated by `fme1` has a wider range, but the coefficient of `age` is tighter. We can use `anova` to compare them, but we must put the `lme` fit first to get the `lme` method to execute.

```
anova( fme1, flm1)
```

```
##      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fme1     1  6 451.2116 467.3044 -219.6058
## flm1     2  3 511.5770 519.6234 -252.7885 1 vs 2 66.36537  <.0001
```

`fme1` is a substantially better fit than `flm1`, unsurprising since it uses six parameters vs. `lm`'s three (overall slope & intercept, overall variance, a variance each of random slopes and intercepts by `Subject`, and a variance by `Subject`).

Which Mixed Effects?

That successfully modeled the overall data. The slope and intercept are close to those estimated by `lm` modeling with `Sex` (`flm3`), although `Sex` was not modeled here. The specification `random = ~ age | Subject` caused `lme` to model each subject independently with a slope and intercept. We did not model `Sex`, so we have seven choices to go on modeling. (We could have many more, but we will see that modeling without a random intercept is fruitless, and we will also avoid crossing `Sex` and `Subject`, as each `Subject` has only one `Sex`.)

Model ($x = age$)	Random	by	model
$a + bx$	a & b	`Subject`	`fme1`
$a + bx$	a	`Subject`	`fme2`
$a + bx$	b	`Subject`	`fme2b`
$a + bx + c_{Sex}$	a	`Subject`	`fme3`
$a + bx + c_{Sex}$	a & b	`Subject`	`fme4`
$a + (b + d_{Sex})x + c_{Sex}$	a	`Subject`	`fme5`
$a + (b + d_{Sex})x + c_{Sex}$	a & b	`Subject`	`fme6`

`update` will allow a simpler call.

```
fme2 <- update( fme1, random = ~ 1 | Subject) # just intercept
(sfme2 <- summary( fme2))
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   451.3895 462.1181 -221.6948
##
## Random effects:
##  Formula: ~1 | Subject
##         (Intercept) Residual
## StdDev:    2.072142 1.422728
##
## Fixed effects:  distance ~ age
##                 Value Std.Error DF  t-value p-value
## (Intercept) 24.023148 0.4255878 80 56.44699       0
## age          0.660185 0.0617993 80 10.68272       0
##  Correlation:
##     (Intr)
## age 0
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.68695130 -0.53862941 -0.01232442  0.49100161  3.74701483
##
## Number of Observations: 108
## Number of Groups: 27
```

```
intervals( fme2)
```

```
## Approximate 95% confidence intervals
```

```
##
##  Fixed effects:
##                  lower       est.      upper
## (Intercept) 23.1840802 24.0231481 24.8622161
## age          0.5383446  0.6601852  0.7820257
##
##  Random Effects:
##   Level: Subject
##                     lower     est.    upper
## sd((Intercept)) 1.537069 2.072142 2.793482
##
##  Within-group standard error:
##    lower     est.    upper
## 1.219675 1.422728 1.659585
```

```
anova( fme2, fme1)
```

```
##      Model df       AIC      BIC    logLik   Test  L.Ratio p-value
## fme2     1  4 451.3895 462.1181 -221.6948
## fme1     2  6 451.2116 467.3044 -219.6058 1 vs 2 4.177941  0.1238
```

```
plot( augPred( fme2, primary= ~ age, level=c(0,1)))
```

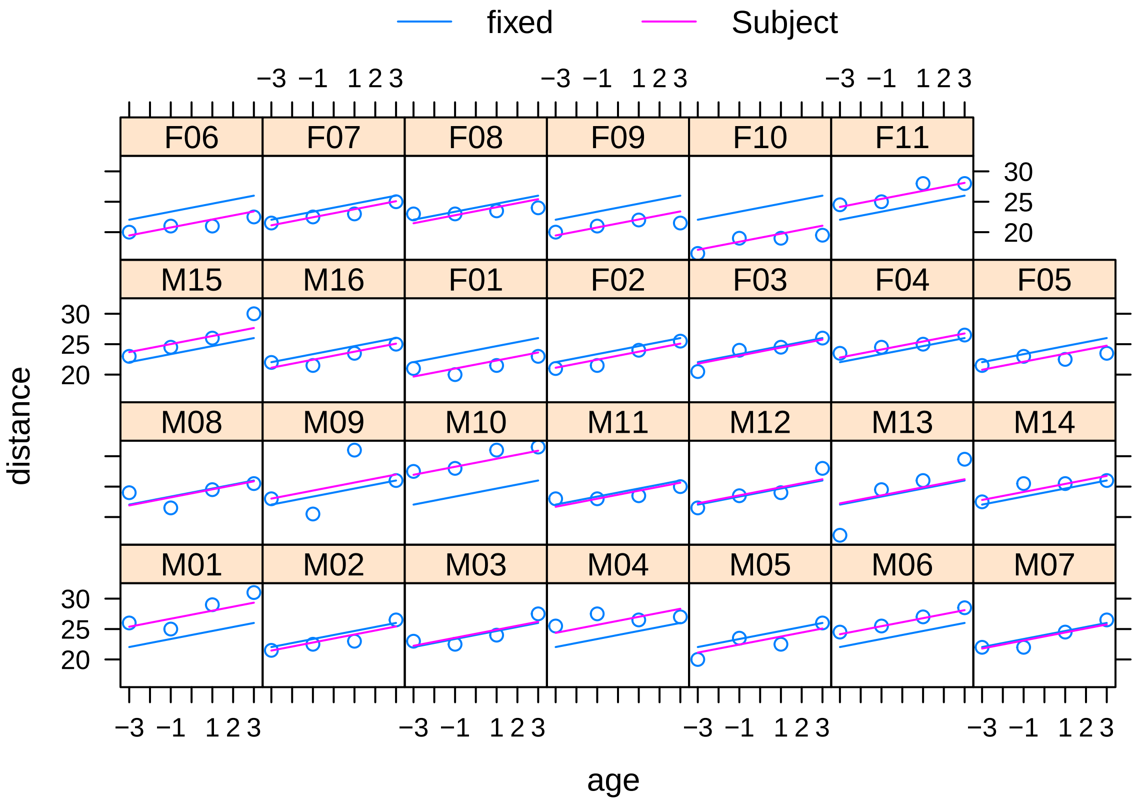


Figure 8: Random effect of intercept only by subject.

Since `fme1` and `fme2` estimated the same parameters with different random effects, it makes sense to compare the resulting parameters. The `coef` function returns the coefficients for each group, combining the random coefficient with the overall coefficient. Since `fme2` only included random effects for the intercept, all of the `age` coefficients in `fme2` are the same.

```
anova( fme2, fme1)
```

```
##      Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fme2     1  4 451.3895 462.1181 -221.6948                        
## fme1     2  6 451.2116 467.3044 -219.6058 1 vs 2 4.177941  0.1238
```

This indicates that the two fits are virtually equivalent, but `fme2` only has four parameters in its model. (So, here is a little mystery: there is no estimate of error by `Subject`. Since there is no slope by `Subject`, there is no real basis to estimate this.)

```
cfme12 <- compareFits( coef( fme1), coef( fme2))
pairs( cfme12)
```

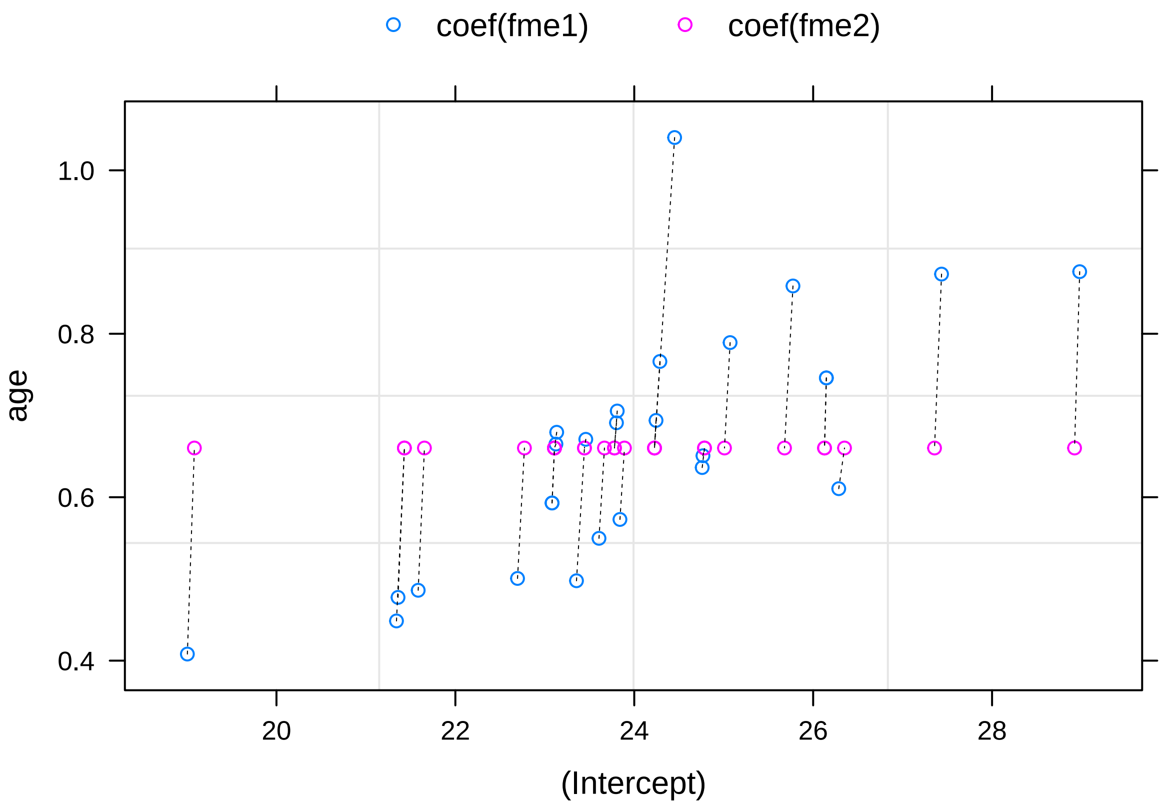


Figure 9: Comparing coefficients when `age` is and is not a random effect

It is clear that if the intercept is below the mean (24.0231481) in `fme2`, the corresponding coefficient of `age` in `fme1` is much smaller than that estimated in `fme2`.

```
plot( as.data.frame( coef( fme1)))
```

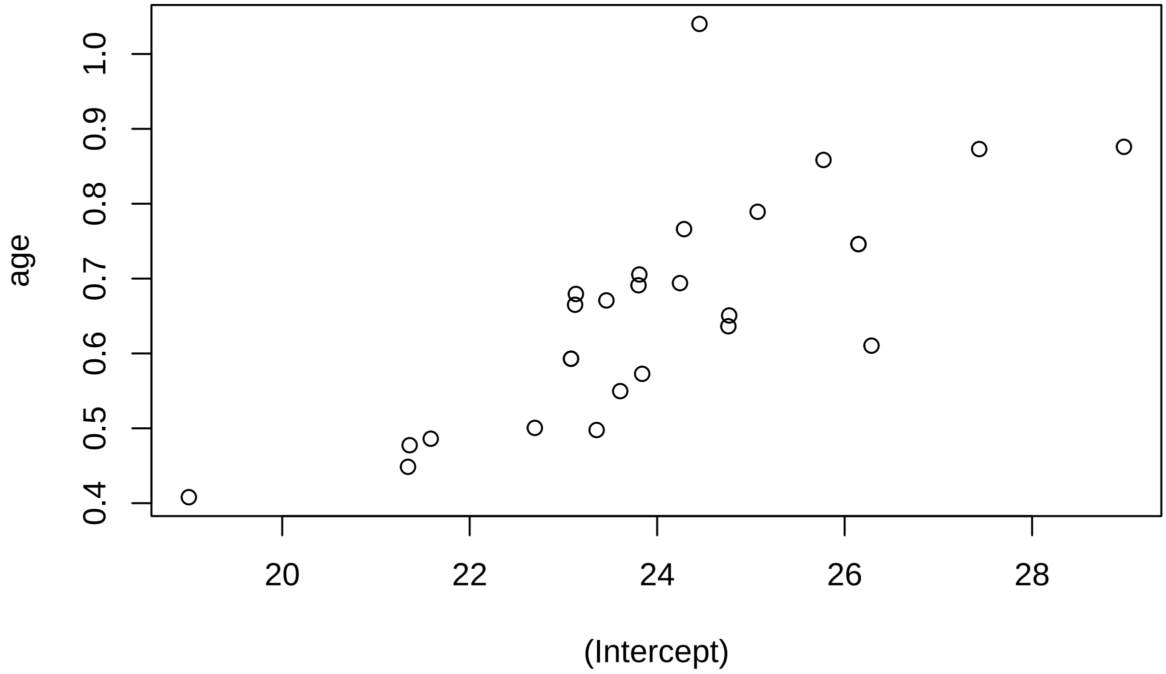


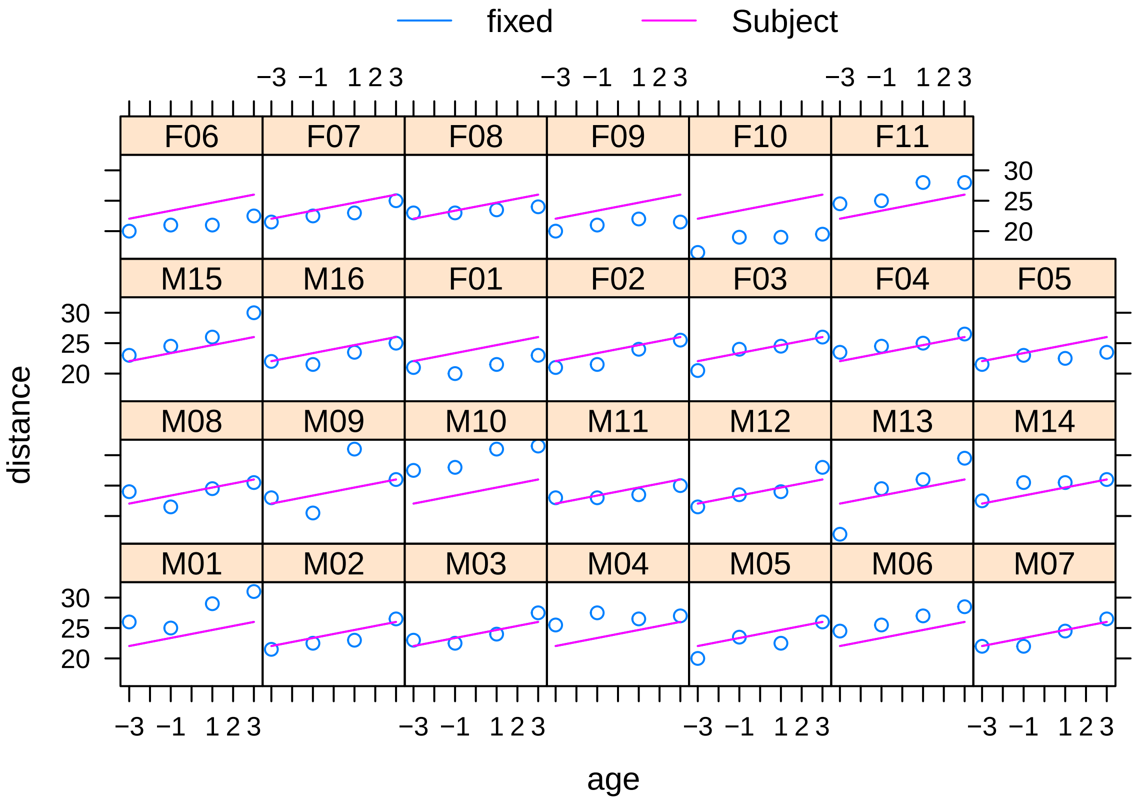
Figure 10: slope vs. intercept for fme1.

So, when estimating random effects of slope and intercept, they end up being strongly correlated. Next, we try `age` as only random effect.

```
fme2b <- update( fme1, random = ~ age - 1 | Subject)
summary( fme2b)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   513.577 524.3055 -252.7885
##
## Random effects:
##  Formula: ~age - 1 | Subject
##                   age Residual
## StdDev: 3.720836e-05 2.513549
##
## Fixed effects:  distance ~ age
##                   Value Std.Error DF  t-value p-value
## (Intercept) 24.023148 0.2441374 80 98.40010       0
## age          0.660185 0.1091816 80  6.04667       0
##  Correlation:
##     (Intr)
## age 0
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.58745882 -0.62770926 -0.07293805  0.53782601  2.51304728
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme2b, primary= ~ age, level=c(0,1)))
```



What happened here? Actually, this did exactly what we would expect, but the slope by individual was a small perturbation. The fixed and `Subject` lines in the figure substantially overlap so it looks like there is only one line. Note that the AIC is much higher, so as expected it is not a useful fit.

Next, continue with `Intercept` as only random effect, but look for an intercept by `Sex`.

```
fme3 <- update( fme2, distance ~ age + Sex) # fme2 only had random intercepts.
# just intercept for each group
summary(fme3)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   444.8565 458.2671 -217.4282
##
## Random effects:
##  Formula: ~1 | Subject
##         (Intercept) Residual
## StdDev:    1.730079 1.422728
##
## Fixed effects:  distance ~ age + Sex
##                 Value Std.Error DF  t-value p-value
## (Intercept) 24.968750 0.4742882 80 52.64468  0.0000
## age          0.660185 0.0620929 80 10.63221  0.0000
## SexFemale   -2.321023 0.7430668 25 -3.12357  0.0045
##  Correlation:
##           (Intr) age
## age        0.000
## SexFemale -0.638  0.000
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.77682007 -0.55426744 -0.01578248  0.45835495  3.68124620
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme3, primary= ~ age, level=c(0,1)))
```

All the fixed effects look significant, and the AIC is lower. Let's add the slope random effect back in.

```
fme4 <- update( fme3, random = ~ age | Subject)
summary(fme4)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   446.8352 465.6101 -216.4176
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 1.7543093 (Intr)
## age         0.2149246 0.202
## Residual    1.3100394
##
## Fixed effects:  distance ~ age + Sex
##                 Value Std.Error DF  t-value p-value
## (Intercept) 24.897236 0.4735834 80 52.57202  0.0000
## age          0.660185 0.0709132 80  9.30977  0.0000
## SexFemale   -2.145489 0.7391993 25 -2.90245  0.0076
##  Correlation:
##           (Intr) age
## age        0.086
## SexFemale -0.636  0.000
```

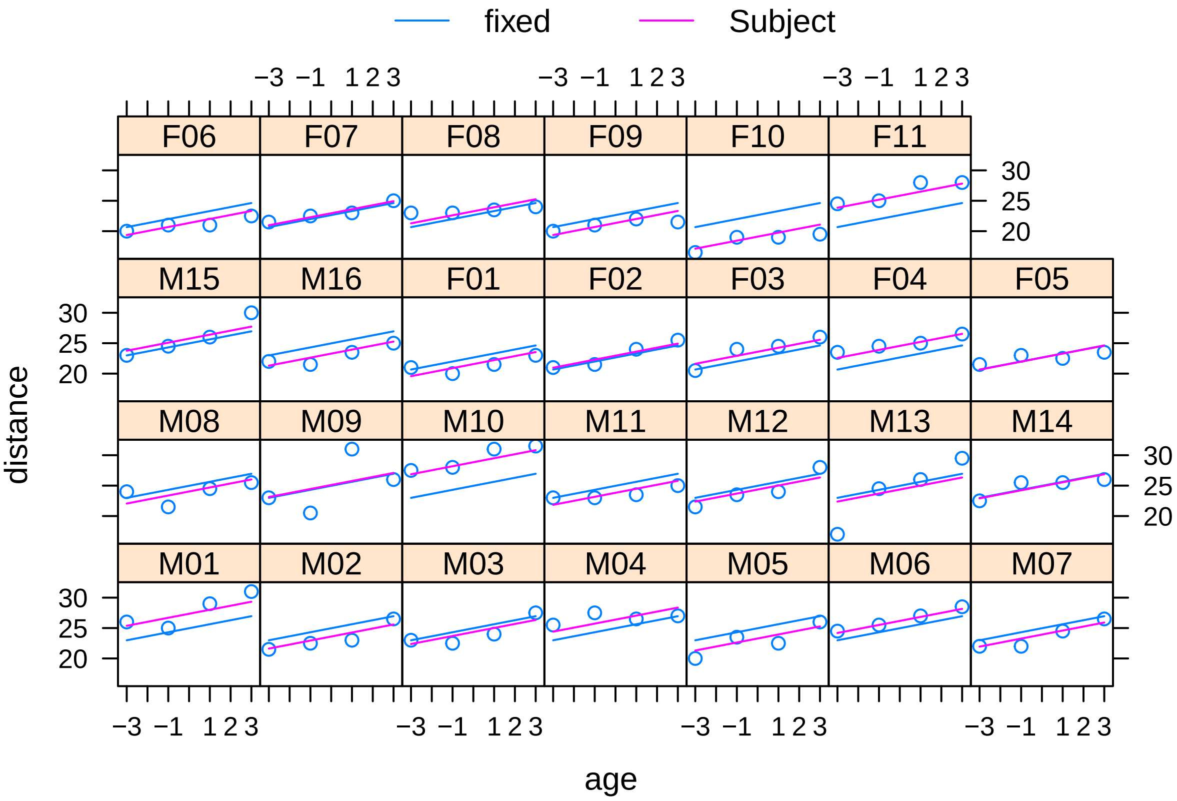


Figure 11: Regression on age with Sex intercept, random effect of intercept only by subject.

```
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.16563254 -0.45463473  0.01446408  0.44559461  3.90045159
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme4, primary= ~ age, level=c(0,1)))
```

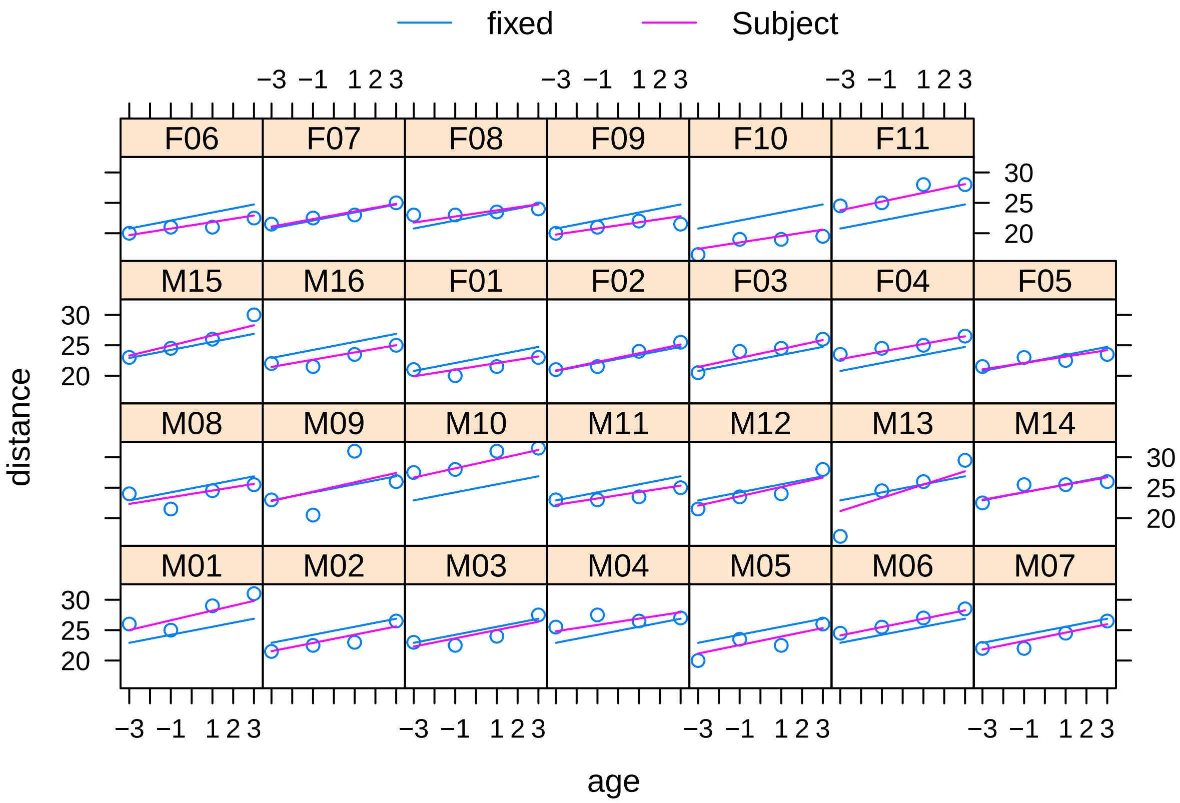


Figure 12: Regression on age and Sex, with random effect of slope and intercept by subject.

Unsurprisingly, the AIC went up. Note though that the fits do appear better when the slope and intercept are random effects.

It seemed reasonable to add `Sex` as an intercept. Let's compare the two ways of specifying random effects.

```
cfme34 <- compareFits( coef( fme3), coef( fme4))
pairs( cfme34)
```

Now we want to look back at only a random intercept, but let there be a different slope by `Sex`.

```
fme5 <- update( fme3, distance ~ age * Sex)
summary( fme5)
```

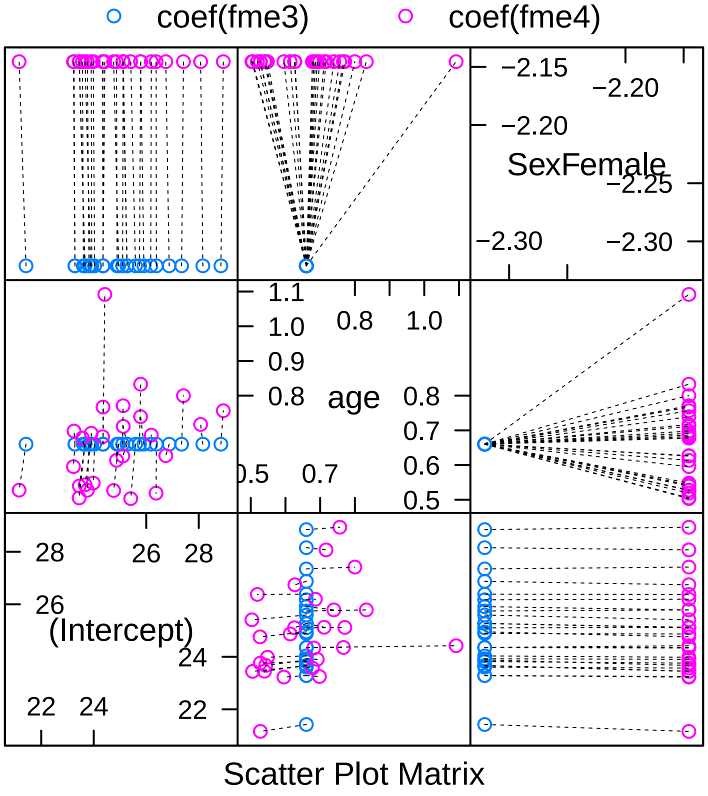


Figure 13: Camparison of fits fme3 & fme4.

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##        AIC      BIC    logLik
##   440.6391 456.7318 -214.3195
##
## Random effects:
##  Formula: ~1 | Subject
##         (Intercept) Residual
## StdDev:    1.740851 1.369159
##
## Fixed effects:  distance ~ age + Sex + age:Sex
##                     Value Std.Error DF  t-value p-value
## (Intercept)   24.968750 0.4765629 79 52.39339  0.0000
## age            0.784375 0.0779963 79 10.05656  0.0000
## SexFemale     -2.321023 0.7466306 25 -3.10866  0.0046
## age:SexFemale -0.304830 0.1221968 79 -2.49458  0.0147
##  Correlation:
##               (Intr) age    SexFml
## age            0.000
## SexFemale     -0.638  0.000
## age:SexFemale  0.000 -0.638  0.000
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.64686407 -0.46341443  0.01556892  0.52172245  3.73335102
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme5, primary= ~ age, level=c( 0, 1)))
```

Lower AIC again. Just to be complete, let us add the slope back in as a random effect.

```
fme6 <- update( fme5, random = ~ age | Subject)
summary( fme6)
```

```
## Linear mixed-effects model fit by maximum likelihood
##   Data: O
##       AIC     BIC   logLik
##   443.806 465.263 -213.903
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 1.7521863 (Intr)
## age         0.1541393 0.234
## Residual    1.3100398
##
## Fixed effects:  distance ~ age + Sex + age:Sex
##                     Value Std.Error DF  t-value p-value
## (Intercept)   24.968750 0.4765627 79 52.39342  0.0000
## age            0.784375 0.0843295 79  9.30132  0.0000
```

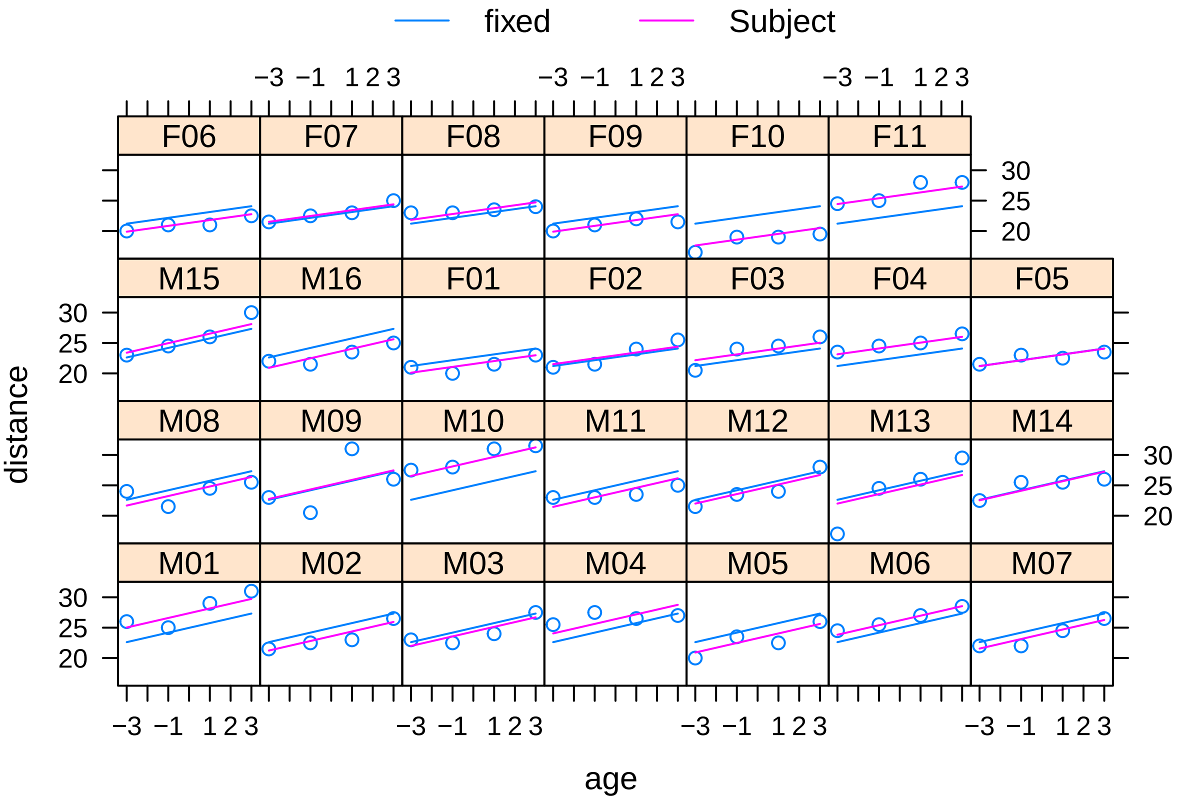


Figure 14: Slope by age and Sex, random intercept by Subject only.

```
## SexFemale     -2.321023 0.7466303 25 -3.10866  0.0046
## age:SexFemale -0.304830 0.1321189 79 -2.30724  0.0237
##  Correlation:
##               (Intr) age    SexFml
## age            0.102
## SexFemale     -0.638 -0.065
## age:SexFemale -0.065 -0.638  0.102
##
## Standardized Within-Group Residuals:
##        Min         Q1        Med          Q3         Max
## -3.3360287 -0.4153976  0.0103922  0.4916951  3.8581927
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme6, primary= ~ age, level=c(0,1)))
```

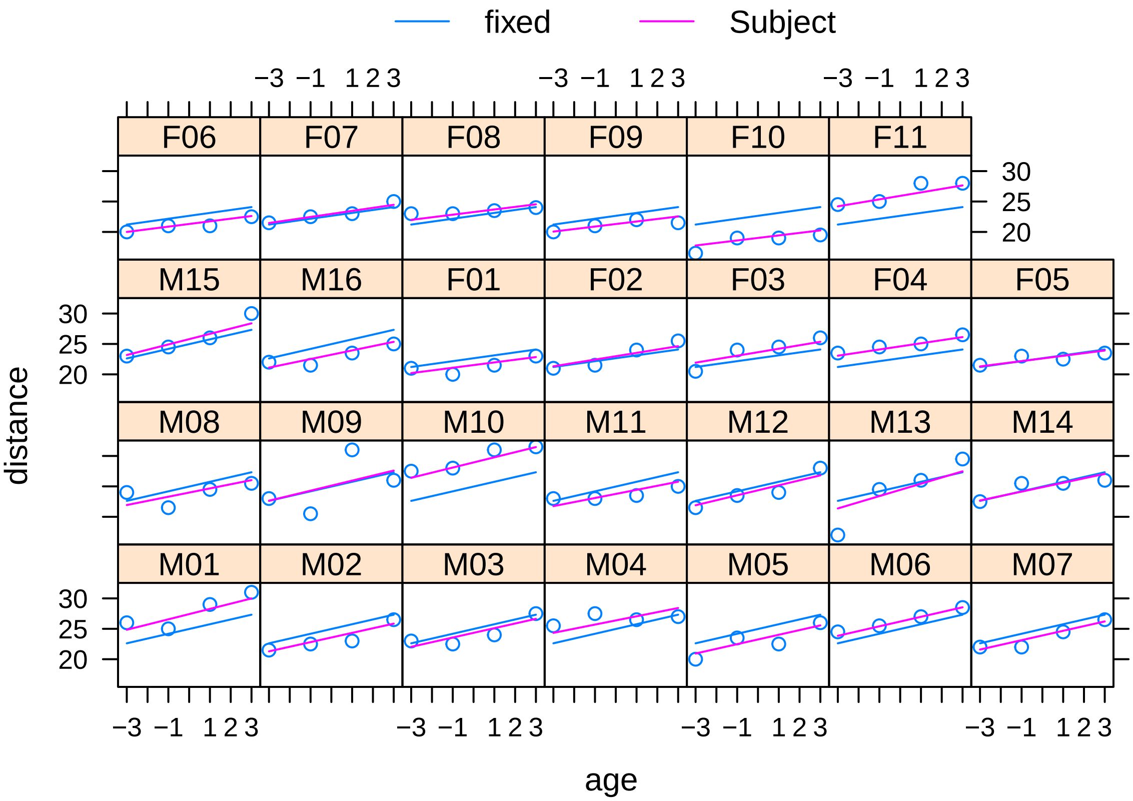


Figure 15: Slope by age and Sex, random slope & intercept by subject.

That generated the same coefficients, only the significance (Std. error) changed.

Finding the Best Fit

The Akaike "An Information Criterion" will discern between these six estimates:

```
AIC( fme1, fme2, fme3, fme4, fme5, fme6)
```

```
##      df      AIC
## fme1  6 451.2116
## fme2  4 451.3895
## fme3  5 444.8565
## fme4  7 446.8352
## fme5  6 440.6391
## fme6  8 443.8060
```

`fme5` has the lowest AIC, with slope for `age`, and `age:Sex`, and an intercept for `Sex`, and treating the intercepts as random by `Subject`.

```
anova( fme2, fme4, fme5)
```

```
##      Model df      AIC      BIC    logLik   Test   L.Ratio p-value
## fme2     1  4 451.3895 462.1181 -221.6948
## fme4     2  7 446.8352 465.6101 -216.4176 1 vs 2 10.554381  0.0144
## fme5     3  6 440.6391 456.7318 -214.3195 2 vs 3  4.196103  0.0405
```

All of these models have only random intercepts. `fme5` is the clear winner.

Now we can look at the normality assumptions.

```
qqnorm( fme5, abline=c(0,1), id=0.05, idLabels=paste(O$Subject,O$age))
```

Disappointing at best, and we can see that Subject M09 and M13 are outliers.

```
plot(Orthodont[O$Subject %in% c('M09', 'M13'),])
```

`M13` grows very fast (perhaps this is *normal*), but `M09` shrinks, twice. We may have reason to drop this datum, as it is the only one which shows a decrease. If we had more data, such as who made the measurements, we might justify removing the point. The point adds noise to the estimates, but this noise could be representative of the noise present in all the measurements.

The following will show the `qqplots` by `Sex`.

```
qqnorm(fme5, ~ residuals(., type = "pearson") | Sex, abline = c(0, 1))
```

The random effects are also assumed to be normal:

```
qqnorm(fme5, ~ranef(., standard=TRUE), abline=c(0,1))
```

We can compare the magnitude of the residuals against the fitted value.

```
plot( fme5, abs( resid(.)) ~ fitted(.), type=c("p", "smooth"))
```

That looks pretty random. Those outliers make the Males look more random than the Females.

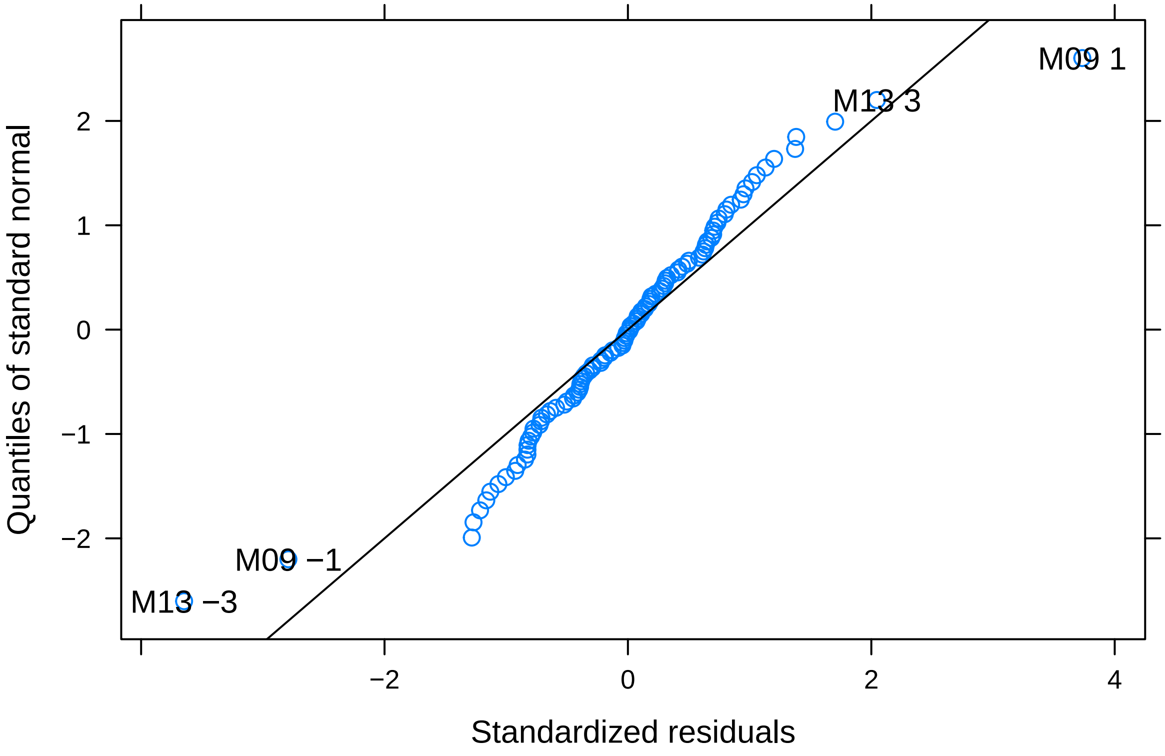


Figure 16: qqplot of fme5

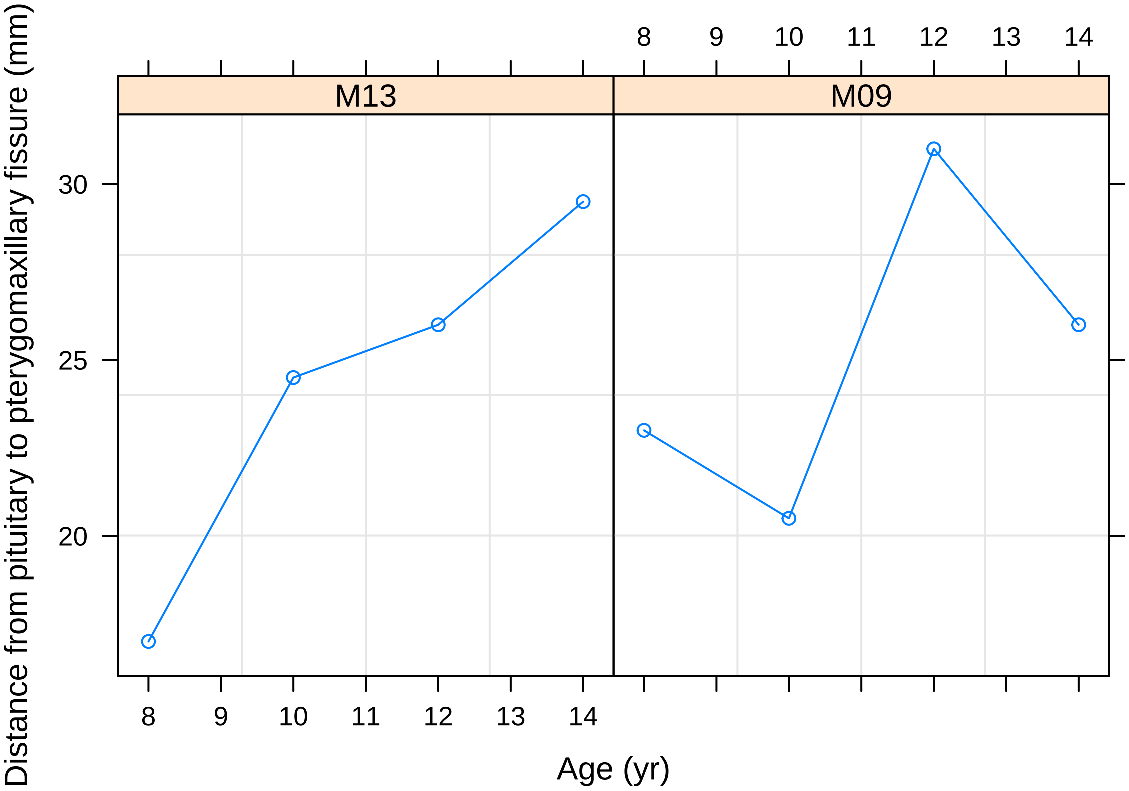


Figure 17: Outlier Subjects

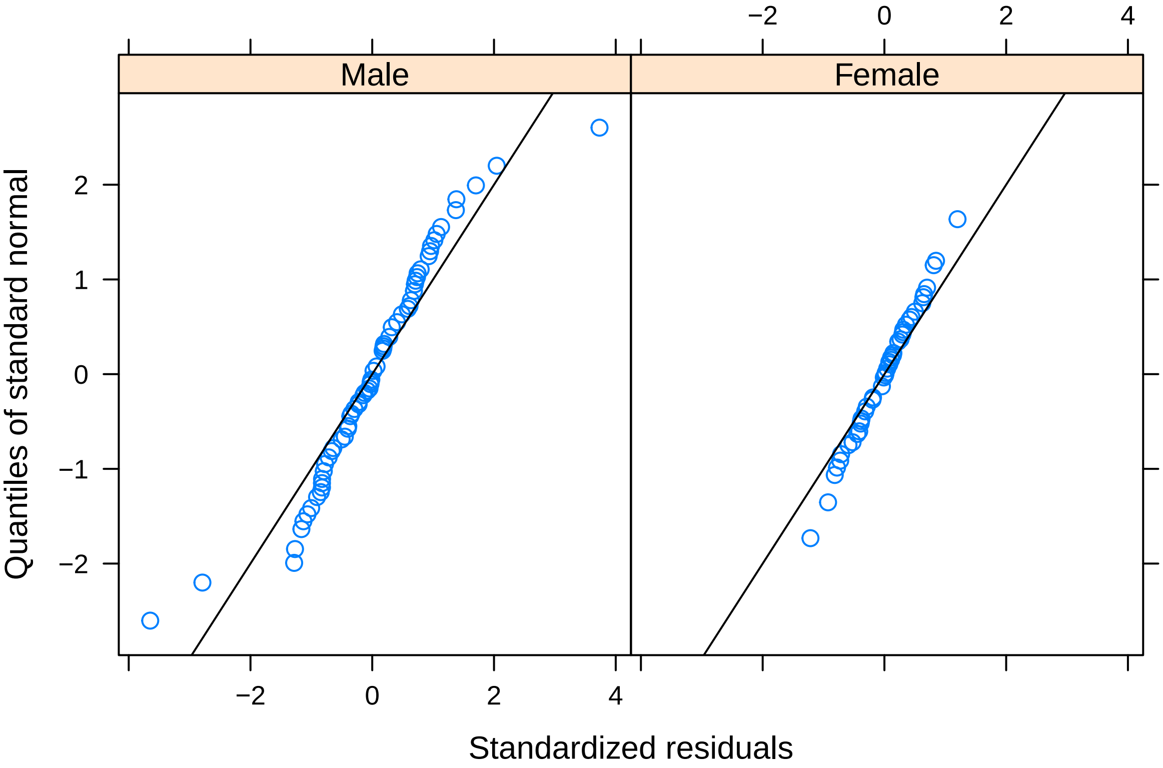


Figure 18: Residuals by `Sex`

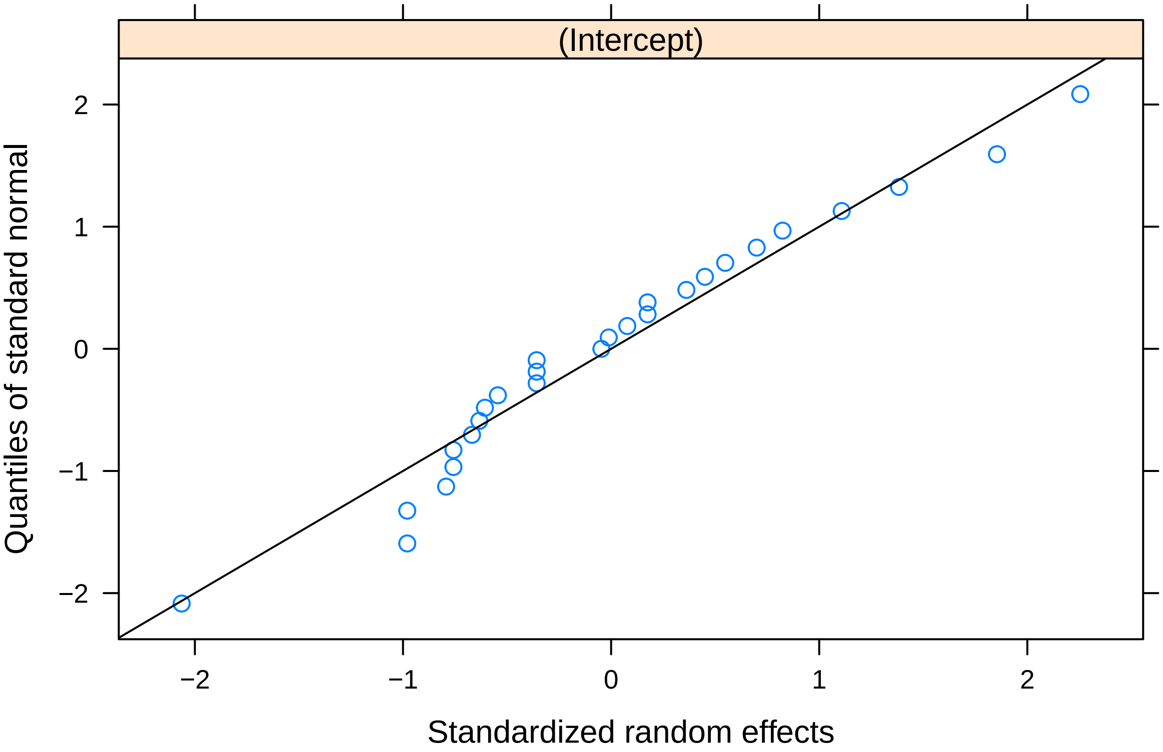


Figure 19: Normal assumption of random effects.

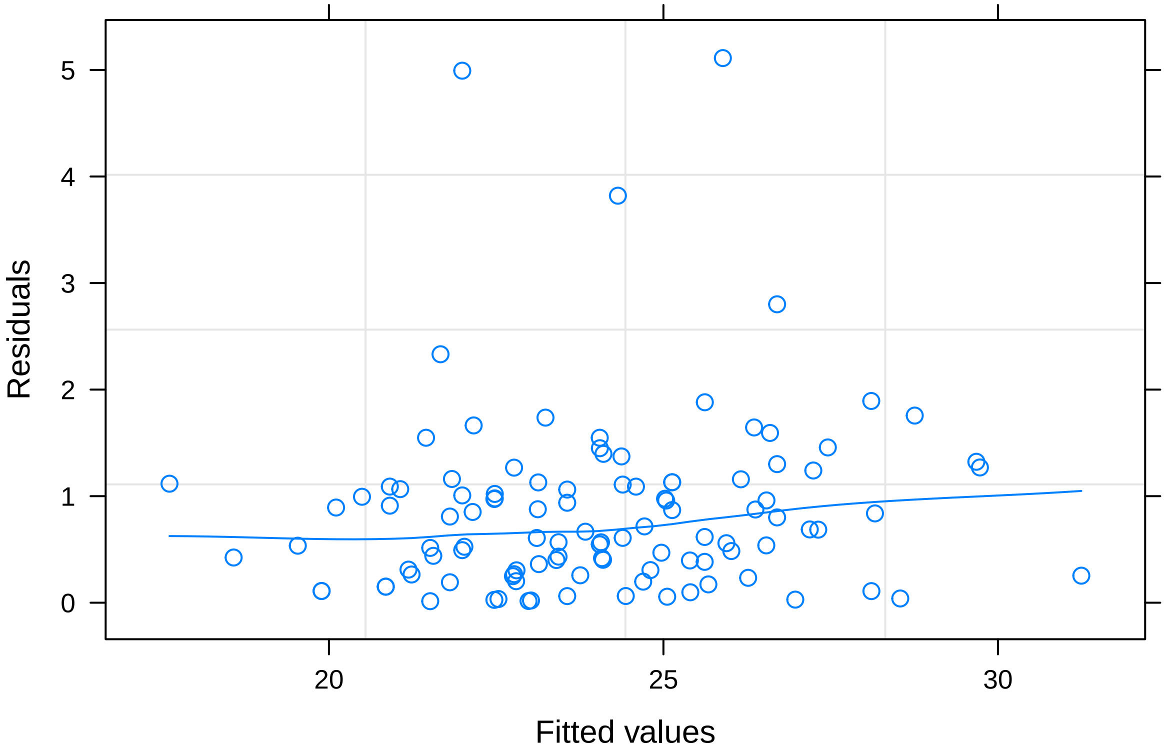


Figure 20: Error magnitude against fitted values

```
plot( fme5, resid(.) ~ fitted(.) | Sex, type=c("p", "smooth"))
```

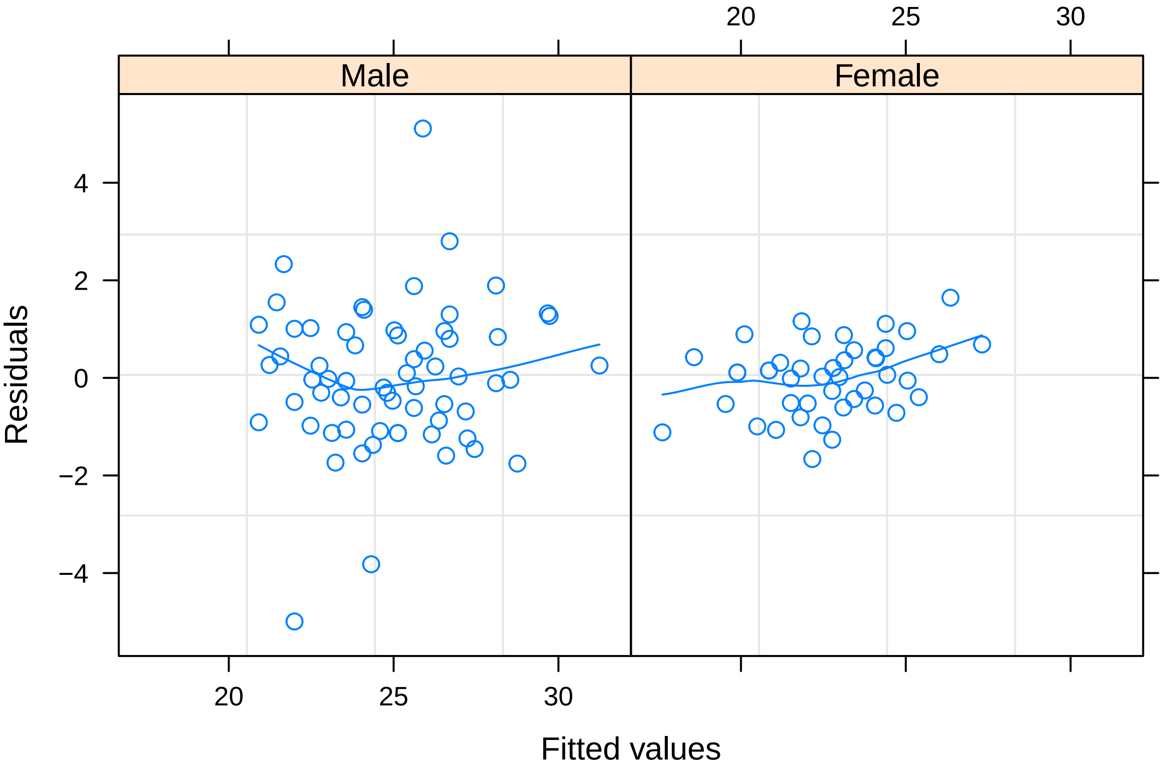


Figure 21: Error magnitude against fitted values

```
plot( fme5, Subject ~ resid(.))
```

Subject M09 and M13 stand out here too. Comparing the measured vs. the fitted values can show oddities too.

```
plot( fme5, distance ~ fitted(.) | Subject, abline=c( 0, 1))
```

Fitted Parameters

Recall that the mixed effects model is the following (1):

$$y_{i,j} = (m + m_j)x_{i,j} + b + b_j + \epsilon$$

where m and b are unknown constants to be estimated, ϵ is $\mathcal{N}(o, \sigma_\epsilon)$, and the m_j and b_j are $\mathcal{N}(o, \sigma_j)$. Therefore the parameters are $m, b, \sigma_j, \&\sigma_\epsilon$.

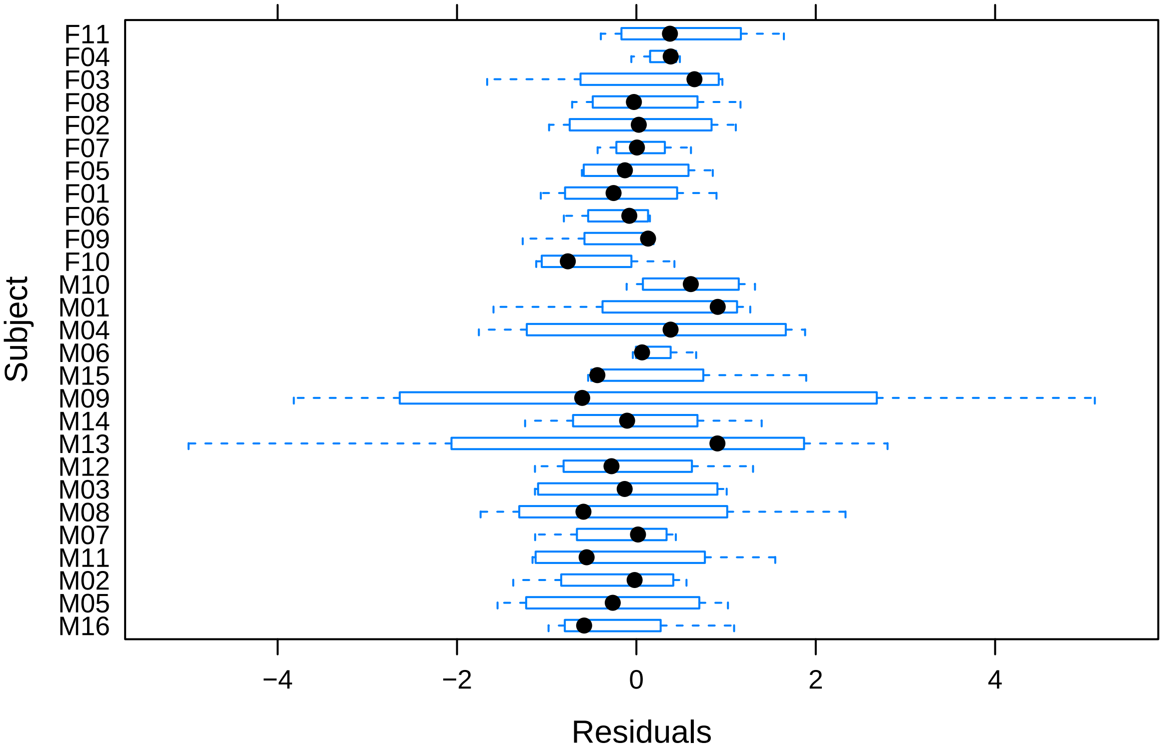


Figure 22: Bar plot of the residuals by Subject. Note that there are only four samples for each subject.

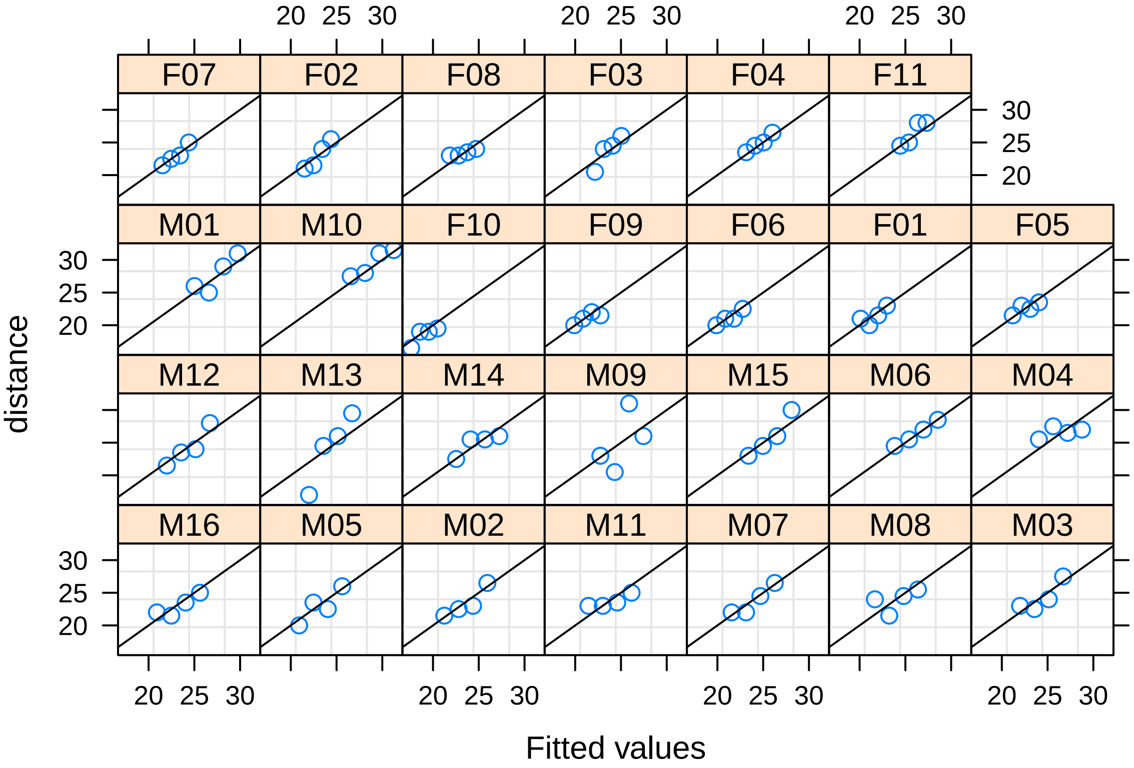


Figure 23: Measured vs. fitted for fme5.

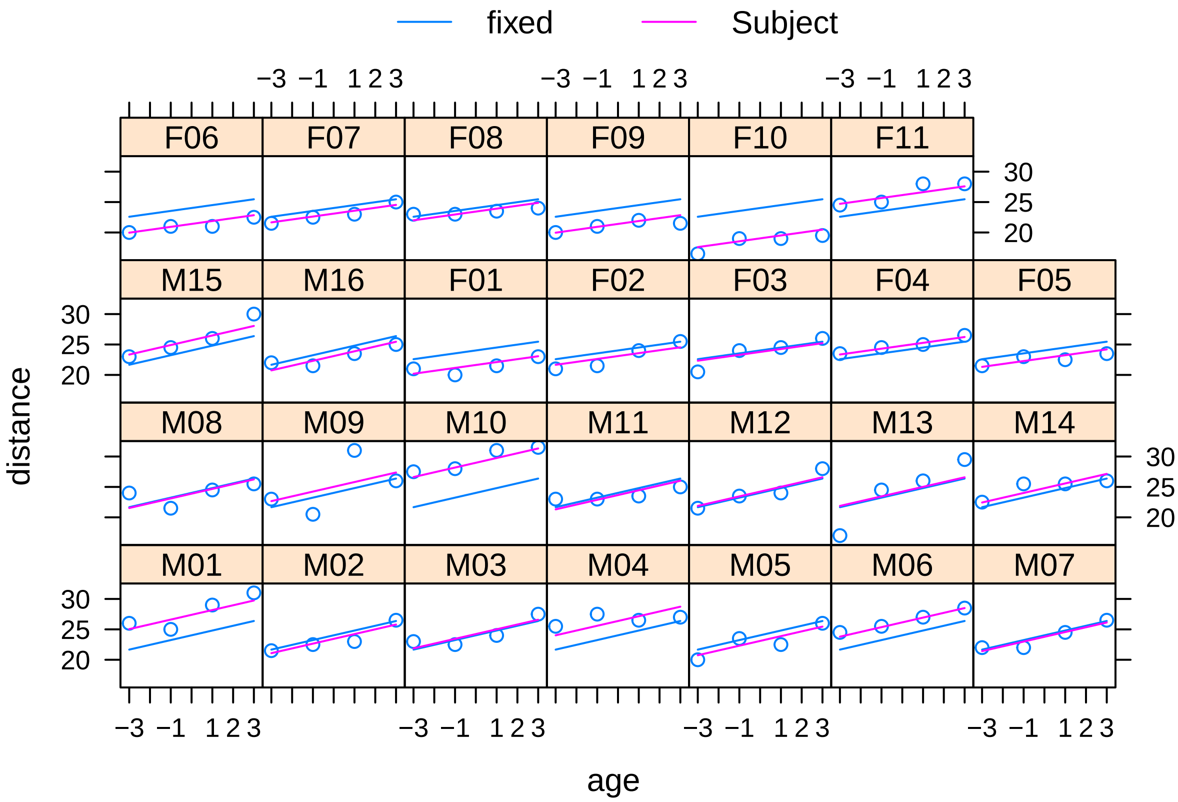
A few more fits

This will be like `fme5` but without the intercept for `Sex`.

```
fme7 <- lme( distance ~ age + age:Sex, data = O, random = ~ 1 | Subject)
summary( fme7)
```

```
## Linear mixed-effects model fit by REML
##   Data: O
##        AIC      BIC    logLik
##   453.2473 466.5171 -221.6236
##
## Random effects:
##  Formula: ~1 | Subject
##         (Intercept) Residual
## StdDev:    2.122241 1.386382
##
## Fixed effects:  distance ~ age + age:Sex
##                     Value Std.Error DF  t-value p-value
## (Intercept)   24.023148 0.4296605 79 55.91193  0.0000
## age            0.784375 0.0775011 79 10.12082  0.0000
## age:SexFemale -0.304830 0.1214209 79 -2.51052  0.0141
##  Correlation:
##               (Intr) age
## age            0.000
## age:SexFemale  0.000 -0.638
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.51635125 -0.51115029  0.02470155  0.50922913  3.74849595
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme7, primary= ~ age, level=c(0,1)))
```

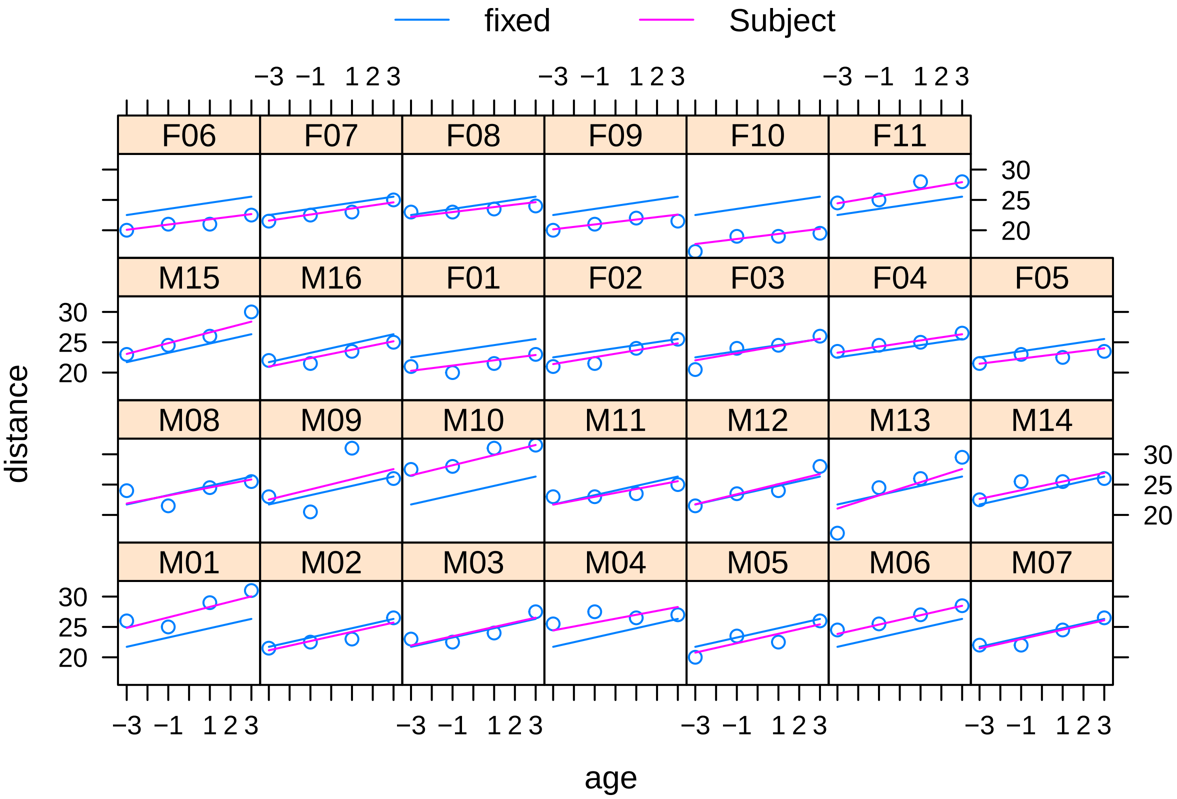


Now like `fme6` but without intercept for `Sex` (or, like `fme7` with randeom effect for intercept by `Subject`).

```
fme8 <- lme( distance ~ age + age:Sex, data = O, random = ~ age | Subject)
summary( fme8)
```

```
## Linear mixed-effects model fit by REML
##   Data: O
##        AIC      BIC    logLik
##   456.0717 474.6494 -221.0358
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 2.1343310 (Intr)
## age         0.1814285 0.232
## Residual    1.3100396
##
## Fixed effects:  distance ~ age + age:Sex
##                    Value Std.Error DF  t-value p-value
## (Intercept)    24.023148 0.4296605 79 55.91193  0.0000
## age             0.767325 0.0859019 79  8.93257  0.0000
## age:SexFemale  -0.262979 0.1340340 79 -1.96203  0.0533
##  Correlation:
##               (Intr) age
## age            0.090
## age:SexFemale  0.000 -0.636
##
## Standardized Within-Group Residuals:
##           Min            Q1           Med            Q3           Max
## -3.0908839327 -0.4553307438 -0.0001948845  0.4720029381  3.9083215943
##
## Number of Observations: 108
## Number of Groups: 27
```

```
plot( augPred( fme8, primary= ~ age, level=c(0,1)))
```



The AIC's for both are a bit larger.

Simulated Data

If we simulate our data we can control the experiment completely and use very large samples that should converge to the correct values. Following equation 1 we can have a number of different random effects (like `Sex` and `Subject` from `Orthodont`), and we will call this number M and will range from `1:M`; we will use the index j. Each effect may have K different values, and thus `K` intercepts and `K` means. `K` could be a random number for each effect (thus there could be a K_j), and there is no particular reason why it cannot be large, but we will fix it. If K is large enough, we should be able to get a very precises estimate of the mean and variance of each of the M random effects. Each sample (row of final data frame) will have its response y given by:

$$y_i = \left(m + \sum_{j=1}^{M} m_{i,j} \right) x_i + b + \sum_{j=1}^{M} b_{i,j} \tag{2}$$

$$y_i = \left(m + \sum_{j=1}^{M} m_{j,k_i} \right) x_i + b + \sum_{j=1}^{M} b_{j,k_i}$$

where the quantities m_{j,k_i} and b_{j,k_i} is meant to indicate the slope and intercept that belongs of the group k that sample i belongs. This create a data frame with $2M + 2$ columns: a column for each of the M factors, one for y and one for x.

Making the dataframe is a little tricky. The data frame will have a column for x, y, and M columns for the factors. Making y will require that we create the random effects for each of the $2M$ effects.

```
require( mvtnorm)
```

```
## Loading required package: mvtnorm
```

```
M <- 3 # Number of random effects factors
K <- rep( 100, M) # number of samples of each factor.
N <- max( K) * 1000 # Ensures we have many samples of each factor.
mx <- -2 # slope & intercept of fixed effect
bx <- 1
sx <- 1 # std dev of noise on y.
mu <- 0:(2*M-1) # the means and intercepts of random effects will be increasing.
Sigma <- diag( 1/( 1 + mu)) # variances are decreasing, and independent.
re <- rmvnorm( n=max(K), mean=mu, sigma=Sigma, method="chol")
colMeans( re)
```

```
## [1] -0.05882303  1.07331612  1.94287514  3.01502188  3.95989590  4.99036205
```

```
var( re)
```

```
##              [,1]        [,2]         [,3]         [,4]         [,5]         [,6]
## [1,] 0.88900878 0.050863921  0.034120865  0.017419569  0.020352560  0.002795770
## [2,] 0.05086392 0.426028584  0.086172160  0.006744660  0.040984566  0.001006058
## [3,] 0.03412087 0.086172160  0.348745673 -0.025430802  0.006798832 -0.001160388
## [4,] 0.01741957 0.006744660 -0.025430802  0.269928340 -0.003215526 -0.029764542
## [5,] 0.02035256 0.040984566  0.006798832 -0.003215526  0.240621182  0.027073928
## [6,] 0.00279577 0.001006058 -0.001160388 -0.029764542  0.027073928  0.185302609
```

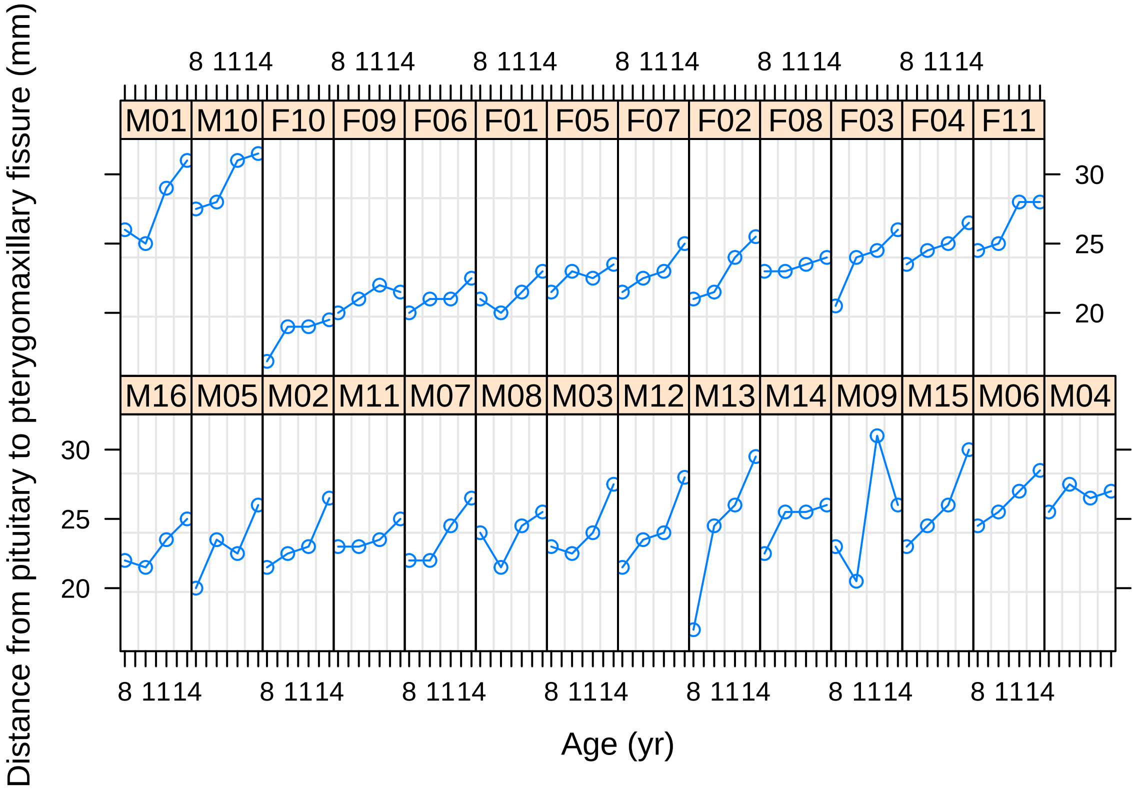
```
# if each k is not the same length, we will just ignore the extra.
df <- data.frame( x=rnorm( N, mean=0, sd=sx))
```

Appendix

The `Orthodont` data frame is used in many R function examples, and some of those are expanded below.

GroupedData

```
Orth.new <-  # create a new copy of the groupedData object
  groupedData( distance ~ age | Subject,
              data = as.data.frame( Orthodont ),
              FUN = mean,
              outer = ~ Sex,
              labels = list( x = "Age",
                y = "Distance from pituitary to pterygomaxillary fissure" ),
              units = list( x = "(yr)", y = "(mm)") )
plot( Orth.new )         # trellis plot by Subject
```



```
formula( Orth.new )       # extractor for the formula
```

```
## distance ~ age | Subject
```

```
gsummary( Orth.new )      # apply summary by Subject
```

```
##      distance age Subject    Sex
## M16    23.000  11     M16   Male
## M05    23.000  11     M05   Male
## M02    23.375  11     M02   Male
## M11    23.625  11     M11   Male
## M07   23.750  11     M07   Male
## M08   23.875  11     M08   Male
## M03   24.250  11     M03   Male
## M12   24.250  11     M12   Male
## M13   24.250  11     M13   Male
## M14   24.875  11     M14   Male
## M09   25.125  11     M09   Male
## M15   25.875  11     M15   Male
## M06   26.375  11     M06   Male
## M04   26.625  11     M04   Male
## M01   27.750  11     M01   Male
## M10   29.500  11     M10   Male
## F10   18.500  11     F10 Female
## F09   21.125  11     F09 Female
## F06   21.125  11     F06 Female
## F01   21.375  11     F01 Female
## F05   22.625  11     F05 Female
## F07   23.000  11     F07 Female
## F02   23.000  11     F02 Female
## F08   23.375  11     F08 Female
## F03   23.750  11     F03 Female
## F04   24.875  11     F04 Female
## F11   26.375  11     F11 Female
```

```
fm1 <- lme( Orth.new )   # fixed and groups formulae extracted from object
Orthodont2 <- update(Orthodont, FUN = mean)
```

```
summary(fm1)
```

```
## Linear mixed-effects model fit by REML
##   Data: Orth.new
##        AIC      BIC    logLik
##   454.6367 470.6173 -221.3183
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite
##             StdDev    Corr
## (Intercept) 2.3270339 (Intr)
## age         0.2264276 -0.609
## Residual    1.3100399
##
## Fixed effects:  distance ~ age
##                 Value Std.Error DF   t-value p-value
## (Intercept) 16.761111 0.7752461 80 21.620375       0
## age          0.660185 0.0712533 80  9.265334       0
##  Correlation:
##     (Intr)
## age -0.848
##
## Standardized Within-Group Residuals:
##         Min           Q1          Med           Q3          Max
## -3.223106016 -0.493760867  0.007316632  0.472151090  3.916032742
```

```
##
## Number of Observations: 108
## Number of Groups: 27
```

gsummary

```
gsummary(Orthodont)  # default summary by Subject
```

```
##     distance age Subject    Sex
## M16   23.000  11     M16   Male
## M05   23.000  11     M05   Male
## M02   23.375  11     M02   Male
## M11   23.625  11     M11   Male
## M07   23.750  11     M07   Male
## M08   23.875  11     M08   Male
## M03   24.250  11     M03   Male
## M12   24.250  11     M12   Male
## M13   24.250  11     M13   Male
## M14   24.875  11     M14   Male
## M09   25.125  11     M09   Male
## M15   25.875  11     M15   Male
## M06   26.375  11     M06   Male
## M04   26.625  11     M04   Male
## M01   27.750  11     M01   Male
## M10   29.500  11     M10   Male
## F10   18.500  11     F10 Female
## F09   21.125  11     F09 Female
## F06   21.125  11     F06 Female
## F01   21.375  11     F01 Female
## F05   22.625  11     F05 Female
## F07   23.000  11     F07 Female
## F02   23.000  11     F02 Female
## F08   23.375  11     F08 Female
## F03   23.750  11     F03 Female
## F04   24.875  11     F04 Female
## F11   26.375  11     F11 Female
```

```
## gsummary with invariantsOnly = TRUE and omitGroupingFactor = TRUE
## determines whether there are covariates like Sex that are invariant
## within the repeated observations on the same Subject.
gsummary(Orthodont, inv = TRUE, omit = TRUE)
```

```
##        Sex
## M16   Male
## M05   Male
## M02   Male
## M11   Male
## M07   Male
## M08   Male
## M03   Male
## M12   Male
## M13   Male
## M14   Male
## M09   Male
## M15   Male
## M06   Male
## M04   Male
## M01   Male
## M10   Male
## F10 Female
## F09 Female
## F06 Female
## F01 Female
## F05 Female
## F07 Female
## F02 Female
## F08 Female
## F03 Female
## F04 Female
## F11 Female
```

lme.lmList

```
fm1 <- lmList(Orthodont)
fm2 <- lme(fm1)
summary(fm1)
```

```
## Call:
##   Model: distance ~ age | Subject
##    Data: Orthodont
##
## Coefficients:
##    (Intercept)
##     Estimate Std. Error   t value     Pr(>|t|)
## M16    16.95   3.288173 5.1548379 3.695247e-06
## M05    13.65   3.288173 4.1512411 1.181678e-04
## M02    14.85   3.288173 4.5161854 3.458934e-05
## M11    20.05   3.288173 6.0976106 1.188838e-07
## M07    14.95   3.288173 4.5465974 3.116705e-05
## M08    19.75   3.288173 6.0063745 1.665712e-07
## M03    16.00   3.288173 4.8659237 1.028488e-05
## M12    13.25   3.288173 4.0295930 1.762580e-04
## M13     2.80   3.288173 0.8515366 3.982319e-01
## M14    19.10   3.288173 5.8086964 3.449588e-07
## M09    14.40   3.288173 4.3793313 5.509579e-05
## M15    13.50   3.288173 4.1056231 1.373664e-04
## M06    18.95   3.288173 5.7630783 4.078189e-07
## M04    24.70   3.288173 7.5117696 6.081644e-10
## M01    17.30   3.288173 5.2612799 2.523621e-06
## M10    21.25   3.288173 6.4625549 3.065505e-08
## F10    13.55   3.288173 4.1208291 1.306536e-04
## F09    18.10   3.288173 5.5045761 1.047769e-06
## F06    17.00   3.288173 5.1700439 3.499774e-06
## F01    17.25   3.288173 5.2460739 2.665260e-06
## F05    19.60   3.288173 5.9607565 1.971127e-07
## F07    16.95   3.288173 5.1548379 3.695247e-06
## F02    14.20   3.288173 4.3185072 6.763806e-05
## F08    21.45   3.288173 6.5233789 2.443813e-08
## F03    14.40   3.288173 4.3793313 5.509579e-05
## F04    19.65   3.288173 5.9759625 1.863600e-07
## F11    18.95   3.288173 5.7630783 4.078189e-07
##    age
##     Estimate Std. Error   t value     Pr(>|t|)
## M16    0.550  0.2929338 1.8775576 6.584707e-02
## M05    0.850  0.2929338 2.9016799 5.361639e-03
## M02    0.775  0.2929338 2.6456493 1.065760e-02
## M11    0.325  0.2929338 1.1094659 2.721458e-01
## M07    0.800  0.2929338 2.7309929 8.511442e-03
## M08    0.375  0.2929338 1.2801529 2.059634e-01
## M03    0.750  0.2929338 2.5603058 1.328807e-02
## M12    1.000  0.2929338 3.4137411 1.222240e-03
## M13    1.950  0.2929338 6.6567951 1.485652e-08
## M14    0.525  0.2929338 1.7922141 7.870160e-02
## M09    0.975  0.2929338 3.3283976 1.577941e-03
## M15    1.125  0.2929338 3.8404587 3.247135e-04
## M06    0.675  0.2929338 2.3042752 2.508117e-02
## M04    0.175  0.2929338 0.5974047 5.527342e-01
## M01    0.950  0.2929338 3.2430540 2.030113e-03
## M10    0.750  0.2929338 2.5603058 1.328807e-02
## F10    0.450  0.2929338 1.5361835 1.303325e-01
## F09    0.275  0.2929338 0.9387788 3.520246e-01
## F06    0.375  0.2929338 1.2801529 2.059634e-01
## F01    0.375  0.2929338 1.2801529 2.059634e-01
## F05    0.275  0.2929338 0.9387788 3.520246e-01
## F07    0.550  0.2929338 1.8775576 6.584707e-02
## F02    0.800  0.2929338 2.7309929 8.511442e-03
## F08    0.175  0.2929338 0.5974047 5.527342e-01
## F03    0.850  0.2929338 2.9016799 5.361639e-03
## F04    0.475  0.2929338 1.6215270 1.107298e-01
## F11    0.675  0.2929338 2.3042752 2.508117e-02
##
## Residual standard error: 1.31004 on 54 degrees of freedom
```

```
summary(fm2)
```

```
## Linear mixed-effects model fit by REML
##   Data: Orthodont
##        AIC      BIC    logLik
##   454.6367 470.6173 -221.3183
##
## Random effects:
##  Formula: ~age | Subject
##  Structure: General positive-definite, Log-Cholesky parametrization
##             StdDev    Corr
## (Intercept) 2.3270354 (Intr)
## age         0.2264279 -0.609
## Residual    1.3100397
```

```
##
## Fixed effects:  distance ~ age
##                   Value Std.Error DF   t-value p-value
## (Intercept) 16.761111 0.7752462 80 21.620373       0
## age           0.660185 0.0712533 80  9.265331       0
##  Correlation:
##     (Intr)
## age -0.848
##
## Standardized Within-Group Residuals:
##          Min           Q1          Med           Q3          Max
## -3.223105153 -0.493761169  0.007316599  0.472151011  3.916033284
##
## Number of Observations: 108
## Number of Groups: 27
```

anova.gls

```
# Pinheiro and Bates, p. 251-252
fm1Orth.gls <- gls(distance ~ Sex * I(age - 11), Orthodont,
correlation = corSymm(form = ~ 1 | Subject),
weights = varIdent(form = ~ 1 | age))
fm2Orth.gls <- update(fm1Orth.gls,
corr = corCompSymm(form = ~ 1 | Subject))
anova(fm1Orth.gls, fm2Orth.gls)
```

```
##             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fm1Orth.gls     1 14 452.5468 489.5683 -212.2734
## fm2Orth.gls     2  9 449.9724 473.7719 -215.9862 1 vs 2 7.425576  0.1909
```

anova.lme

```
## Pinheiro and Bates, pp. 251-254 ------------------------------------------
fm1Orth.gls <- gls(distance ~ Sex * I(age - 11), Orthodont,
correlation = corSymm(form = ~ 1 | Subject),
weights = varIdent(form = ~ 1 | age))
fm2Orth.gls <- update(fm1Orth.gls,
corr = corCompSymm(form = ~ 1 | Subject))
## anova.gls examples:
anova(fm1Orth.gls, fm2Orth.gls)
```

```
##             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fm1Orth.gls     1 14 452.5468 489.5683 -212.2734
## fm2Orth.gls     2  9 449.9724 473.7719 -215.9862 1 vs 2 7.425576  0.1909
```

```
fm3Orth.gls <- update(fm2Orth.gls, weights = NULL)
anova(fm2Orth.gls, fm3Orth.gls)
```

```
##              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fm2Orth.gls     1  9 449.9724 473.7719 -215.9862
## fm3Orth.gls     2  6 445.7572 461.6236 -216.8786 1 vs 2 1.784873  0.6182
```

```
fm4Orth.gls <- update(fm3Orth.gls, weights = varIdent(form = ~ 1 | Sex))
anova(fm3Orth.gls, fm4Orth.gls)
```

```
##              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## fm3Orth.gls     1  6 445.7572 461.6236 -216.8786
## fm4Orth.gls     2  7 436.1887 454.6994 -211.0943 1 vs 2 11.56859   7e-04
```

```
# not in book but needed for the following command
fm3Orth.lme <- lme(distance ~ Sex*I(age-11), data = Orthodont,
random = ~ I(age-11) | Subject,
weights = varIdent(form = ~ 1 | Sex))
# Compare an "lme" object with a "gls" object (test would be non-sensical!)
anova(fm3Orth.lme, fm4Orth.gls, test = FALSE)
```

```
##              Model df      AIC      BIC    logLik
## fm3Orth.lme     1  9 429.5225 453.3220 -205.7612
## fm4Orth.gls     2  7 436.1887 454.6994 -211.0943
```

as/matrix.corrStruc

```
cst1 <- corAR1(form = ~1|Subject)
cst1 <- Initialize(cst1, data = Orthodont)
as.matrix(cst1)
```

```
## $M01
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M02
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M03
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M04
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M05
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M06
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M07
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M08
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M09
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M10
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M11
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

```
##
## $M12
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M13
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M14
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M15
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $M16
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F01
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F02
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F03
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F04
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F05
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F06
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F07
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F08
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F09
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F10
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
##
## $F11
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

as.matric.reStruct

```
rs1 <- reStruct(pdSymm(diag(3), ~age+Sex, data = Orthodont))
as.matrix(rs1)
```

```
## [[1]]
##             (Intercept) age SexFemale
## (Intercept)           1   0         0
## age                   0   1         0
## SexFemale             0   0         1
```