

# Comparison to other methods

Terry Therneau, Steve Weigand, Clifford Jack

June 25, 2021

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

One missing section in our document is a comparison to other methods. Competitors are

- Simple linear mixed effects models
- Li, Iddi, Thompson, and Donohue, Bayesian Latent time joint mixed effects model. Also Donhue xxx Stat Medthods Med Research 2017. <https://doi.org/10.1177/0962280217737566>
- Kosick, . . . , Johnoson, group trajectory model. Neuroimage 2019, doi: 10.1002/dad2.12007
- Kiddle, Parodi, Johnson, Wallace, and Dobson. Heterogeneity of cognitive decline in dementia: taking into account variable time-zero severity. <https://doi.org/10.1101/060830>
- Proust-Lima, Joint modeling of repeated multivariate cognitive measures and competing risks of dementia and death: a latent process and latent class approach. <https://doi.org/10.1002/sim.6731>

The short take: a simple linear mixed model does not extrapolate well; it is modeling the wrong biology. The Donohue model is very ambitious with 7 distinct endpoints. They replace all endpoints by ranks and assume a multivariate linear model. The Kosick approach has several ad hoc steps, using a 2 stage analysis. Proust-Limma takes into account the competing risk of death, which adds a major complication to the methods, but is something we should persue.

Cécile Proust-Limma has a very impressive software package `lcmm` which I had been aware of previously but had not yet explored in sufficient depth. In looking at the comparison, I came to realize that it can fit all the above models.

## 1 Linear Mixed Effects

The linear mixed effects model is the simplest of the set.

$$\begin{aligned}y_{ij}(t) &= X\beta + b_{i0} + b_{i1}t + \epsilon \\ b &\sim N(0, \Sigma)\end{aligned}$$

where  $y(t)$  is the response, with  $i = 1, \dots, n$  indexing the subject,  $j = 1, \dots, p$  indexes the response, and  $t$  is time. In all of our models this last will be age. The  $X$  matrix contains fixed covariates, which includes an intercept, age, APOE positivity, and male sex. The per

subject/outcome random intercepts  $b_0$  and random slopes  $b_1$  can be correlated or not. The Donohue paper refers to this as a mixed model (MM) in the former case and a joint mixed model (JMM) in the latter.

Start by fitting a linear mixed model to only the amyloid values, i.e.,  $j = 1$ . The fit can be done using standard software.

```
lmefit1 <- lmer(lpib ~ ns(age.vis, 4) + male + apoepos + (1 | ptnum),
               data=d3, REML = FALSE)
lmefit2 <- lmer(lpib ~ ns(age.vis, 4) + male + apoepos + (1+ age.vis | ptnum),
               data=d3, REML= FALSE)
print(lmefit2, cor=FALSE)
Linear mixed model fit by maximum likelihood ['lmerMod']
Formula: lpib ~ ns(age.vis, 4) + male + apoepos + (1 + age.vis | ptnum)
Data: d3
      AIC      BIC    logLik deviance df.resid
-4910.684 -4845.010  2466.342 -4932.684      2883
Random effects:
Groups      Name      Std.Dev. Corr
ptnum      (Intercept) 0.81600
           age.vis     0.01183 -0.97
Residual                    0.03439
Number of obs: 2894, groups:  ptnum, 1285
Fixed Effects:
      (Intercept) ns(age.vis, 4)1 ns(age.vis, 4)2 ns(age.vis, 4)3
           0.24133           0.16985           0.33614           0.65824
ns(age.vis, 4)4           male           apoepos
           0.67827          -0.04908           0.16629
convergence code 0; 1 optimizer warnings; 0 lme4 warnings
table(table(d3$ptnum))

  1   2   3   4   5   6
253 605 314  76  33   4
```

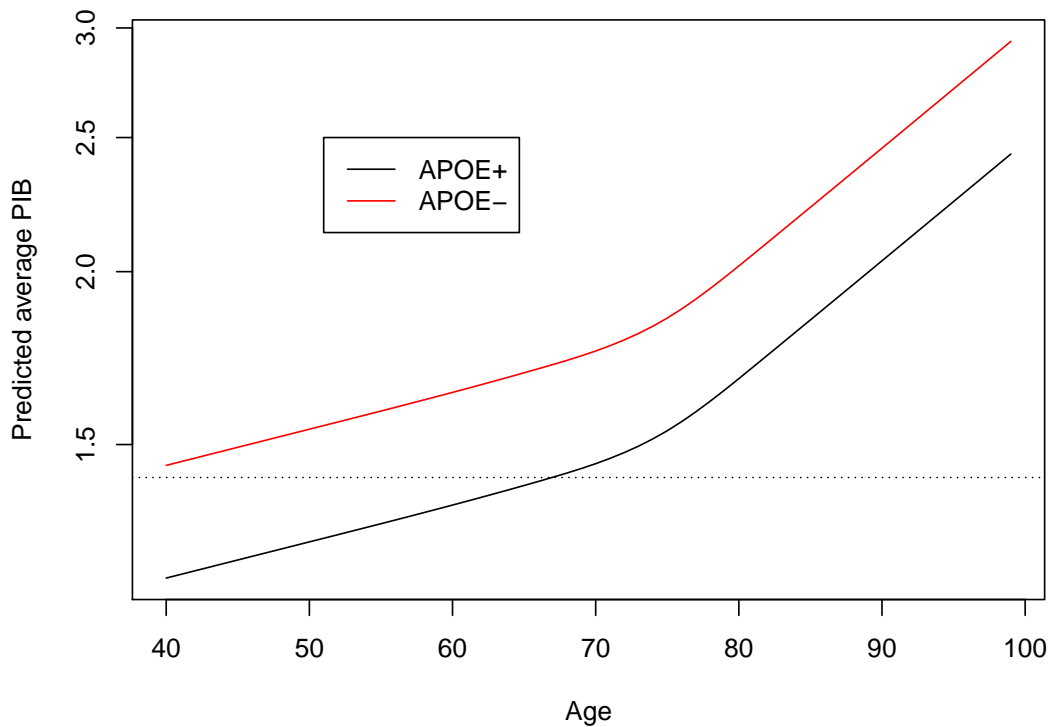
The second fit gives a warning about lack of convergence. Examining the output, the correlation of the random intercept and slope is -.97, which indicates why the model has problems. The final tally shows that the majority of subjects only have 1 or 2 PIB values, so we should not be surprised: Estimation of both a random intercept and a random slope per subject based on only 2 points is too ambitious.

The next figure shows the overall shape of the curve for the first fit, along with the effect of APOE, the strongest covariate. APOE subjects have, on average, a higher amyloid at any age. The problem with the fit is that it is assumed to always be higher, with the “average” APOE positive subject expected to become PIB positive before age 40. The standard LME assumption of vertical up/down shifts simply isn’t feasible. A joint LME fit for both PIB and tau markers will have the same issue, so we won’t pursue it.

```

dummy1 <- expand.grid(age.vis=40:99, apoepos=0:1, male=0)
yhat <- predict(lmefit1, newdata=dummy1, re.form=NA)
matplot(40:99, exp(matrix(yhat, ncol=2)), type='l', lty=1, col=1:2,
        log='y', xlab="Age", ylab= "Predicted average PIB")
legend(51, 2.5, c("APOE+", "APOE-"), lty=1, col=1:2)
abline(h=1.42, lty=3)

```



## 2 Latent time model

Li, Thompson and Donohue add a per subject shift to the LME models and label it as a latent time mixed model (LTMM), then compare this to the simple mixed model (LME) above. They fit the model using both simulation data and ADNI. (No information on the ADNI sample size that I could find.)

$$y_{ij}(t) = X\beta + \gamma_i(td_i) + b_{i0} + b_{i1}t + \epsilon \quad (1)$$

$$b \sim N(0, \Sigma) \quad (2)$$

$$d \sim N(0, \tau^2) \quad (3)$$

Again, Greek letters are fixed effects and Roman ones are random effects. Allowing correlation between the random effects of the two endpoints leads to joint models, JLTMM and JMM.

The paper uses  $j = 7$  joint endpoints, each of which is pre-transformed using *normal scores*. Figure ?? shows the mapping between the SUVR values and normal scores in our data. The LME model is then fit on the transformed scale. A result of the NS transform is that the lower half of the values for each scale has been expanded so that they cover the same range as the upper half; i.e., PIB SUVR values from 1–1.4 map to -3.6–0 and those 1.4–3.5 map to 0–3.6. Such a transformation is often done to dampen the effect of large outliers, but may have surprising effect on the fit. Essentially, the fitting routine has been told that the error in measurement on this new scale is constant, and it will act accordingly — it is now as important for the fitting routine to separate a PIB of 1.1 from 1.2 as to separate a 1.4 from 2.2, as illustrated by the two dashed triangles on the plot; as they have the same vertical rise. The transform has declared these two changes to be “of the same size”. If the drivers of changes in the lower part of PIB range are different than for the upper part, e.g., perhaps the lower region is more subject to small changes in the reference region, then the fitting routine may end up pursuing a red herring. For tau, 82% of the data lies below the “elbow” (red horizontal segment on the figure, SUVR = 1.3), which means that any estimated coefficients are dominated by a search for effects among those values that are  $< 1.3$ . This may not be the effect we were looking for. The table below shows the coefficients of a LME fit to the scored data, with age in decades. Note the attenuation of the tau coefficients.

```
# fit an lme model to the data, a la Li
dz1 <- with(d3, data.frame(ptnum=ptnum, age= age.vis, apoepos= apoepos,
                           male = male, y = rz.pib, tau =0))
dz2 <- with(d3, data.frame(ptnum=ptnum, age= age.vis, apoepos= apoepos,
                           male = male, y = rz.tau, tau =1))
dz3 <- rbind(subset(dz1, !is.na(y)), subset(dz2, !is.na(y)))
lm3 <- lmer(y ~ (age + male + apoepos)*tau + (1| ptnum), dz3)
temp <- fixef(lm3)
tcoef <- rbind(pib= temp[1:4], tau= temp[1:4] + temp[5:8])
tcoef[,2] <- tcoef[,2]*10 # age in decades
round(tcoef, 2)
      (Intercept)  age  male apoepos
pib          -3.31 0.44 -0.23    0.64
tau          -1.51 0.20 -0.11    0.32
# This is a version with correlated random effects
if (FALSE) {
lm3b <- lmer(y ~ (age + male + apoepos)*tau + (1+ tau | ptnum), dz3)
npib <- with(dz3, tapply(tau==0, ptnum, sum))
ntau <- with(dz3, tapply(tau==1, ptnum, sum))
```

```

ztrans <- function(x) {
  n <- length(x)
  qx <- (rank(x)-.5)/n
  qnorm(qx)
}

keep <- !is.na(d3$pib)
d3$rz.pib <- NA
d3$rz.pib[keep] <- ztrans(d3$pib[keep])

keep <- !is.na(d3$tau)
d3$rz.tau <- NA
d3$rz.tau[keep] <- ztrans(d3$tau[keep])

with(d3, matplot(cbind(pib, tau), cbind(rz.pib, rz.tau), type='n',
      xlab="SUVR", ylab="Normal scores"))
lines(rz.pib~pib, data=d3, subset=(order(d3$pib)), lwd=2)
lines(rz.tau~tau, data=d3, subset=(order(d3$tau)), lwd=2, col=2)
lines(c(1.25, 1.45), c(.92, .92), col=2)
lines(c(1.4, 1.6), c(.25, .25), col=1)

# add triangles
xhat <- with(d3, approx(rz.pib, pib, c(-3, -2, 0, 1))$y)
lines(xhat[c(1,2,2)], c(-3, -3, -2), lty=2)
lines(xhat[c(3,4,4)], c(0,0,1), lty=2)

```

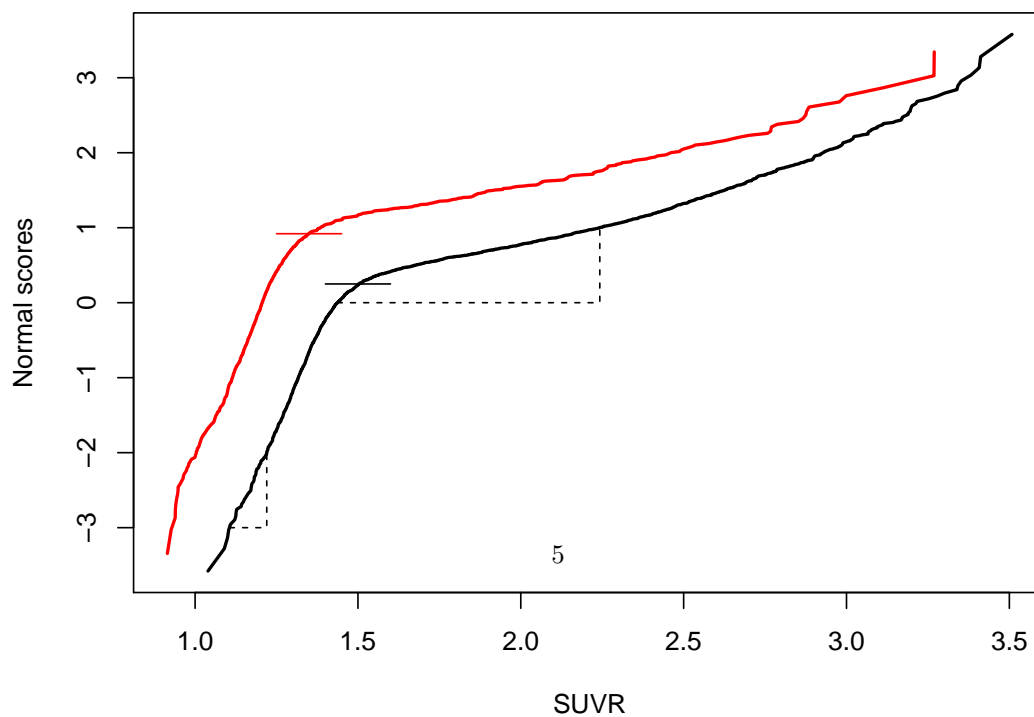


Figure 1: Observed PIB (1.1-1.6) and tau (1.4-1.6) SUVR and corresponding normal distributions of

```

ran3 <- ranef(lm3b)$ptnum
plot(ran3[ntau>0, 1], ran3[ntau>0, 2] + ran3[ntau>0, 1],
     xlab="Amyloid random effect", ylab="Tau random effect")
abline(0, 1, col=2)
}

```

The second is that due to the pronounced elbow in the two curves, when we back transform the model result to the original scale (flip the x and y axes) the predicted curves will have an elbow as well. A plot of two predicted curves for age vs SUVR, for subjects who differ in APOE, say, will look remarkably like a left-right shift. The third consequence is that the lower half of the values has been expanded so that they cover the same range as the upper half; the horizontal line at 0 intersects at the median values of 1.4 for PIB and 1.2 for tau. This expansion of the lower tail may have a surprising effect on the fit. Essentially, the fitting routine has been told that the error in measurement on this new scale is constant and will act accordingly — it is now as important for the fitting routine to separate a PIB of 1.1 from 1.2 as to separate a 1.4 from 2.2, as illustrated by the two red triangles on the plot; as they have the same vertical rise. The transform has declared these two changes to be “of the same size”. If the drivers of changes in the lower part of PIB range are different than for the upper part, e.g., perhaps the lower region is more subject to small changes in the reference region, then the fitting routine may end up pursuing a red herring. For tau, a greater majority of the data lie below the “elbow” making the risk even higher. At the end the predictions will be translated back to the original scale and differences at the lower end will again become minimal, but the damage may already be done wrt the predicted effect of APOE and other covariates.

It should be noted that multi-response mixed models will encounter numerical round-off issues if the responses are of very different magnitudes, say one ranges from 0 to 3 and another from 0 to 3 million, and students are taught that they should renormalize to a common scale to avoid this. (That is, this may be an example reaping the consequences of our own statistical teaching.) For all the responses in our dementia studies this is not an issue, however.

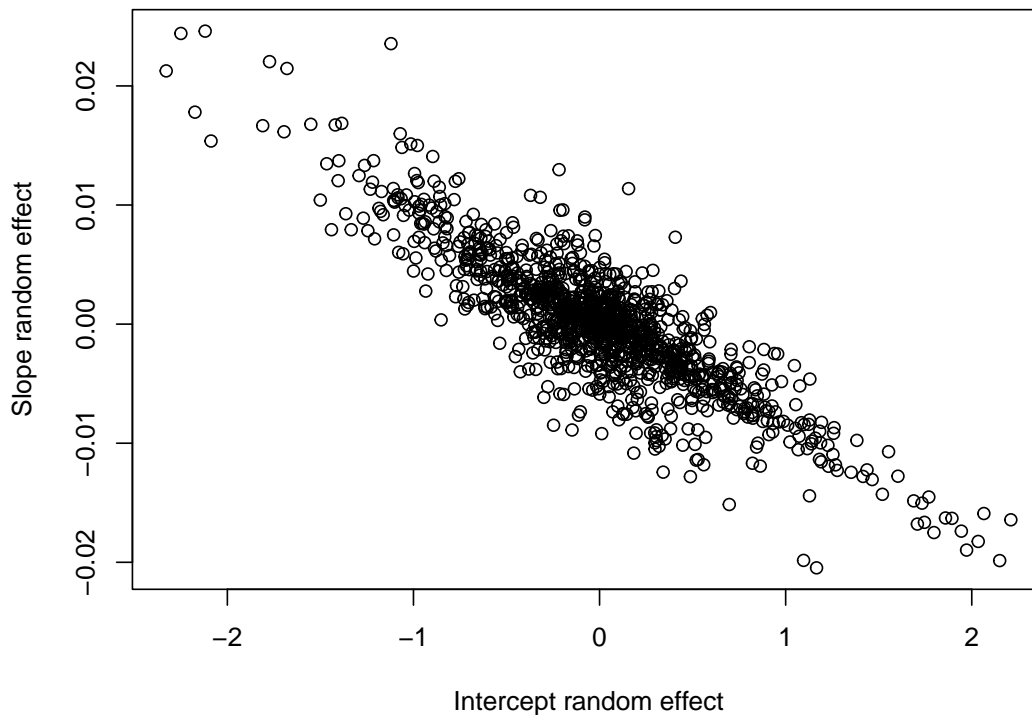
We will fit 3 latent time models to the Mayo data using the `lcmm` software package. The first using the rank-Z transformed data, a second with the raw data, and a third where the code is allowed to pick an optimal transform. First, a test fit.

```

library(lcmm)    # this will mask the ranef and fixef functions of lme

lcfit0 <- mlcmm(rz.pib + rz.tau ~ age.vis + contrast(male) + contrast(apoeapos),
               random= ~age.vis, randomY=FALSE,
               subject="ptnum", ng=1, data=d3, link="linear")
Be patient, multlcmm is running ...
The program took 7.59 seconds
plot(age.vis ~ intercept, data = lcfit0$predRE,
     xlab="Intercept random effect", ylab="Slope random effect")

```



The graph shows that the slope and intercept are very highly correlated. As we saw in the simple lme model, calculation of both a random intercept and random slope per subject is not feasible, they really cannot be disentangled. Even though the Li and Donohue paper includes such a term, we will omit it.

```
lcf1t1 <- mlcmm(rz.pib + rz.tau ~ age.vis + contrast(male) + contrast(apoeapos),
  random= ~1, randomY=TRUE,
  subject="ptnum", ng=1, data=d3, link="linear")
Be patient, multlcmm is running ...
The program took 4.01 seconds
lcf1t2 <- mlcmm(pib + tau ~ age.vis + contrast(male) + contrast(apoeapos),
  random= ~1, randomY=TRUE,
  subject="ptnum", ng=1, data=d3, link="linear")
Be patient, multlcmm is running ...
The program took 2.87 seconds
lcf1t3 <- mlcmm(pib + tau ~ age.vis + contrast(male) + contrast(apoeapos),
  random= ~1, randomY=TRUE,
  subject="ptnum", ng=1, data=d3,
```

```

link="6-manual-splines", intnodes=rep(c(1.4, 1.6, 1.8, 2.5),2))
Be patient, multlcmm is running ...
The program took 22.18 seconds
#plot(lcfit3, 'fit', var.time="age.vis")
#plot(lcfit3, 'link')

```

The graphs below show the predicted values from the rz + linear model fit and the full lcmm fit, for a range of ages, male sex, and APOE-+. The second and fourth plots are quite similar in shape. (I wonder if I've made an intercept error in the fourth one.)

```

# One has to compare the result of coef() to the printout from
# summary() to decode
cfun <- function(fit, newdata) {
  cmat <- cbind(pib= coef(fit)[1:3], tau = coef(fit)[1:3])
  cmat[-1, 1] <- cmat[-1,1] + coef(fit)[4:5]
  cmat[-1, 2] <- cmat[-1,2] - coef(fit)[4:5]

  if (missing(newdata)) cmat
  else list(cmat=cmat, eta = as.matrix(newdata) %*% cmat)
}

eta.to.y <- function(fit, eta) {
  temp <- fit$estimlink
  cbind(pib = approx(temp[,2], temp[,1], eta[,1])$y,
        tau = approx(temp[,4], temp[,3], eta[,2])$y)
}

dummy <- expand.grid(age.vis=50:120, male=0, apoepos=0:1)
yhat1 <- eta.to.y(lcfit1, cfun(lcfit1, dummy)$eta)
yhat2 <- eta.to.y(lcfit2, cfun(lcfit2, dummy)$eta)
yhat3 <- eta.to.y(lcfit3, cfun(lcfit3, dummy)$eta)
ytrans <- matrix(approx(d3$rz.pib, d3$pib, yhat1)$y, ncol=4)

# The yhat1 needs to be back-translated from rz to real units
oldpar <- par(mfrow=c(2,2), mar=c(5,5,1,1))
matplot(50:120, matrix(yhat1, ncol=4), type='l', col=c(1,1,2,2),
        lty=1:2, xlab="Age", ylab="PET, linear on NS scale")

matplot(50:120, ytrans, type='l', col=c(1,1,2,2),
        lty=1:2, xlab="Age", ylab="PET, back-transformed NS")
legend("topleft", c("A+ PIB", "A- PIB", "A+ Tau", "A- Tau"),
        lty=c(2,1,2,1), col=c(1,1,2,2), bty='n')

matplot(50:120, matrix(yhat2, ncol=4), type='l', col=c(1,1,2,2),
        lty=1:2, xlab="Age", ylab="PET, linear")

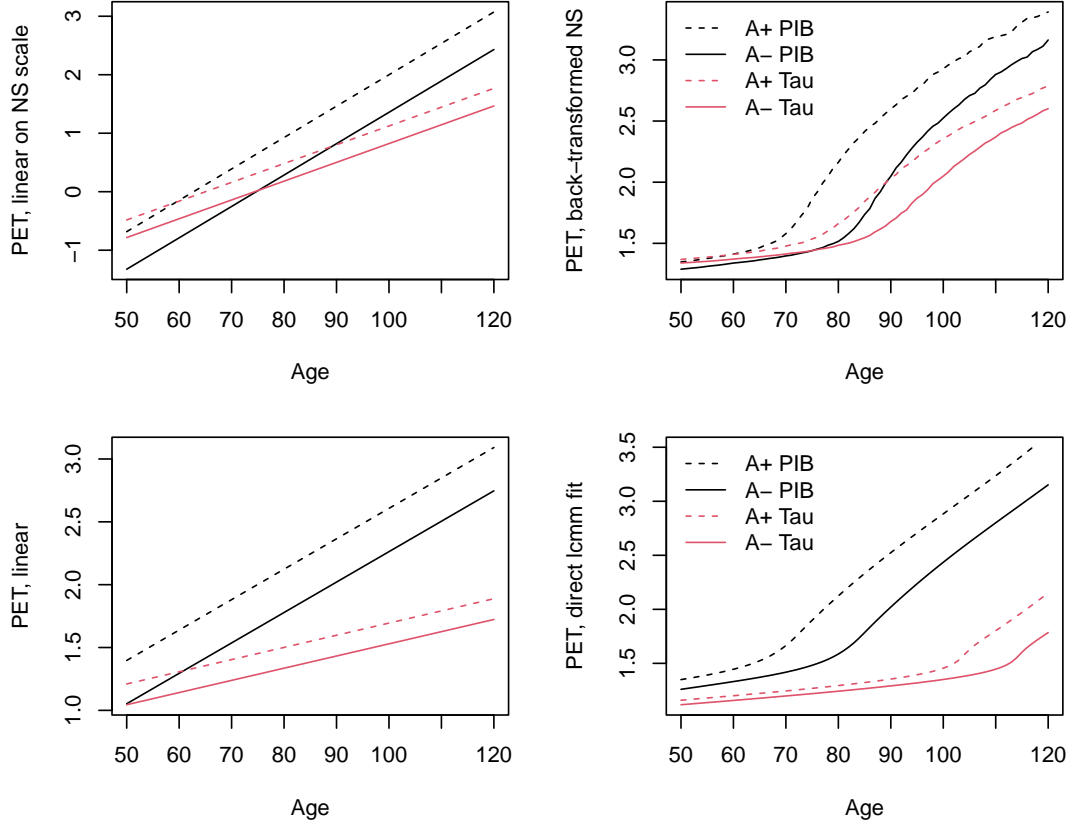
```



```

matplot(50:120, matrix(yhat3, ncol=4), type='l', col=c(1,1,2,2),
        lty=1:2, xlab="Age", ylab="PET, direct lcmm fit")
legend("topleft", c("A+ PIB", "A- PIB", "A+ Tau", "A- Tau"),
        lty=c(2,1,2,1), col=c(1,1,2,2), bty='n')

```



```

par(oldpar)

```

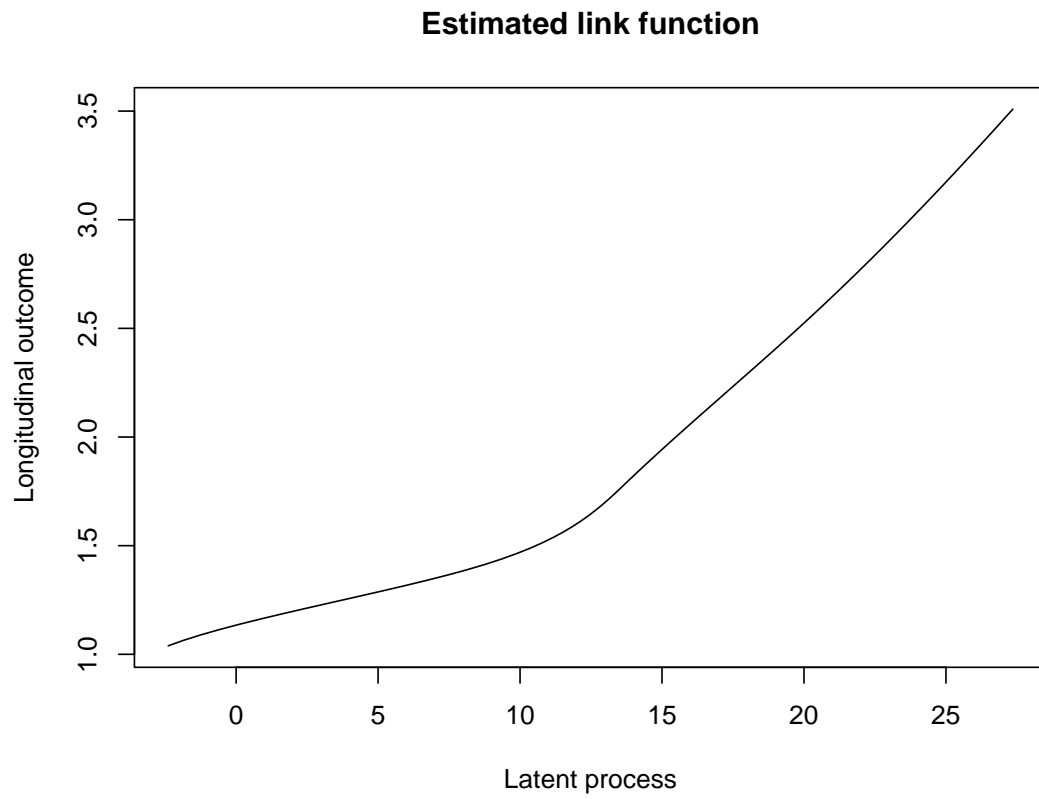
### 3 Trajectory models

The Kosciak paper uses a 3 stage fitting process. First, they fit a trajectory model to all the amyloid data, using subjects who have at least 2 amyloid values. Group based trajectory modeling (GBTM) has become popular in the social sciences; it is distributed as a add on procedure to PC SAS named Proc TRAJ. There are some shortcomings, however. In particular, covariates are not included, so the trajectories are built using only age and response, and secondly the trajectories are simple polynomials. In order to achieve mononicity and non-crossing, the authors had to play with parameters, and chose a final fit based on “reasonableness of the results”.

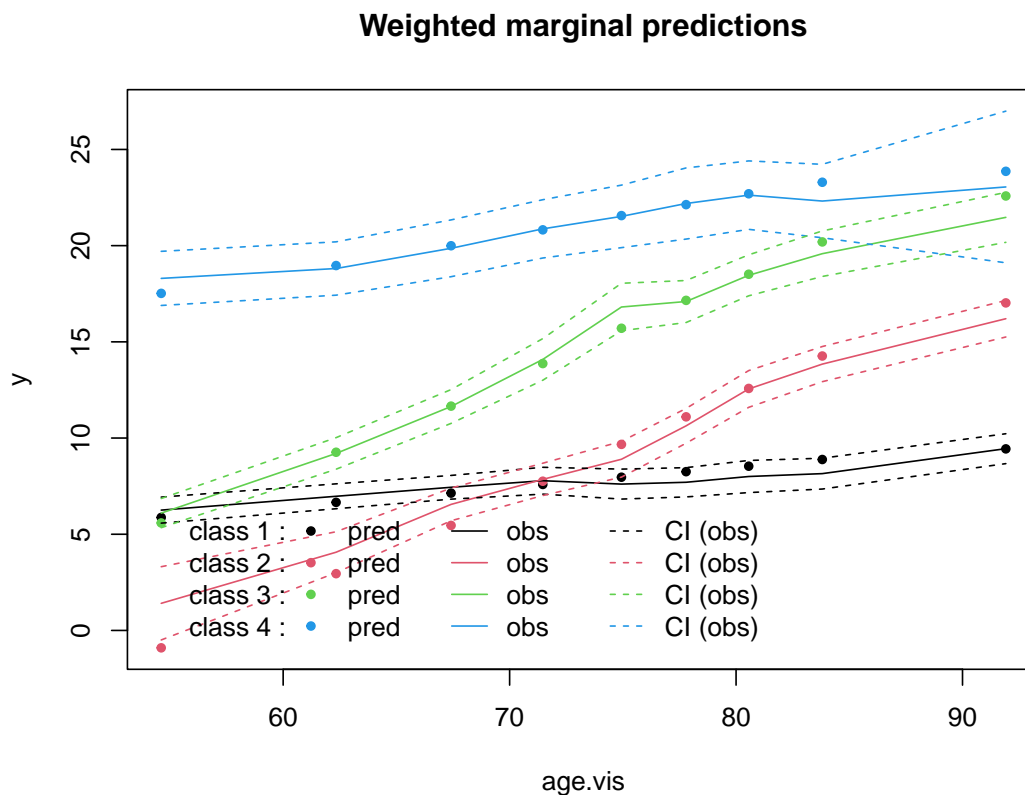
There were 257 subjects with at least 1 PIB scan, of which 171 had at least 2 scans (a requirement of GTBM), and these split into 4 trajectories, along with a classification of each subject's membership as a probability vector, e.g., subject 1 might be classified as (.6, .3, .1), 60% likely to be a sample from trajectory 1, 30% from trajectory 2, and 10% from number 3. The most probable category was 1,2,3, and 4 for 125, 21, 14, and 11 of the 171 subjects. As a second step they re-capitulated the calculation of the probabilities, given the categories, and used that to classify subjects with only a single amyloid result. From each curve they then extracted the point at which it crossed the arbitrary threshold of 1.2, each subject's crossing point was then computed as a weighted sum of the curve crossings. "Chronicity" is defined as the number of years since crossing. The third step was to use subjects' chronicity value as an outcome, and assess its relationship to sex, APOE status, and tau.

Fit the first step of this using lcmm and only amyloid.

```
kfit1 <- lcmm(pib ~ age.vis, random= ~1, subject="ptnum", ng=4,
             link = "6-manual-splines", mixture= ~ age.vis,
             intnodes = rep(c(1.4, 1.6, 1.8, 2.5),1), data = d3)
Be patient, lcmm is running ...
The program took 8.32 seconds
plot(kfit1, 'link')
```



```
kpred <- plot(kfit1, 'fit', var.time="age.vis")
```

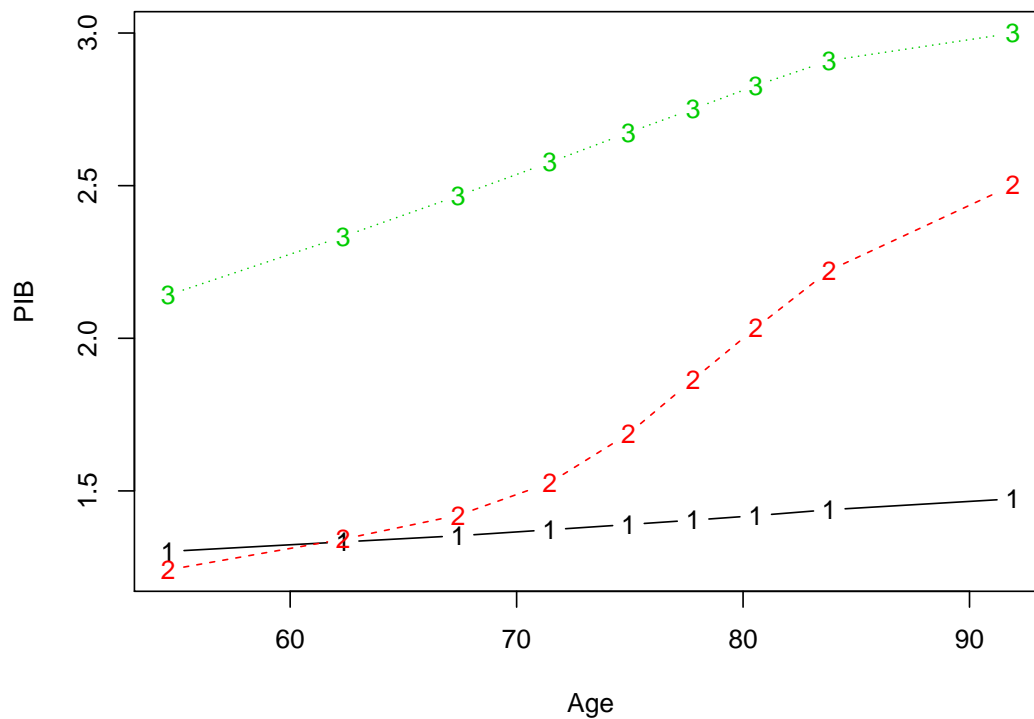


```
temp <- table(class=kfit1$pprob$class, fdata$study)
temp <- cbind(temp, Pct= 100 *rowSums(temp)/sum(temp))
round(temp)
```

	ADRC	MCSA	pop	Pct
1	5	747	59	
2	2	200	16	
3	17	194	16	
4	85	35	9	

The lmm function breaks the the trajectory into two portions, a common non-linear aspect, which we have forced to be monotone, and per group linear trajetories. In this case the function has created 3 groups, two which increase slowly and one which changes rapidly. Replot this on age scale.

```
temp <- approx(kfit1$estimlink[,2], kfit1$estimlink[,1], kpred[,2:4])$y
matplot(kpred[,1], matrix(temp, ncol=3), type='b', xlab="Age", ylab="PIB")
```

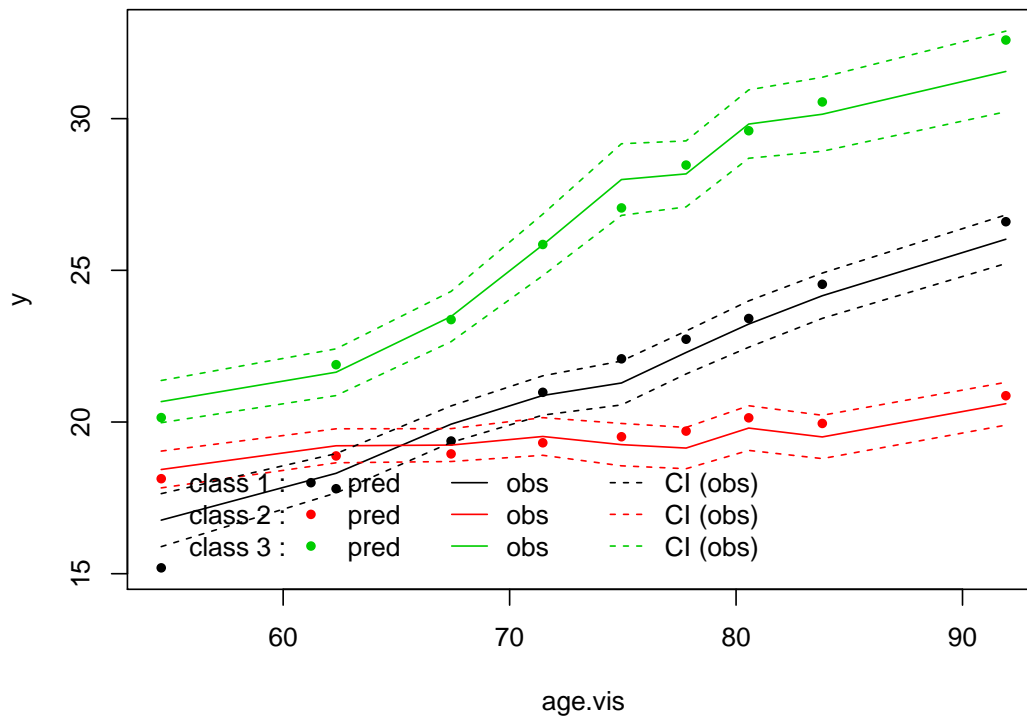


We can't use the chronicity trick of Kosciak, since the group 3 curve never crosses the threshold; it includes most of the MCSA people.

We can also allow covariates in the model.

```
kfit2 <- lcmm(pib ~ age.vis + male + apoepos + study,
             mixture= ~ age.vis + study + male + apoepos,
             random= ~1, subject="ptnum", ng=3,
             link = "6-manual-splines",
             intnodes = rep(c(1.4, 1.6, 1.8, 2.5),1), data = d3)
Be patient, lcmm is running ...
The program took 26.78 seconds
plot(kfit2, 'fit', var.time="age.vis")
```

# Weighted marginal predictions



```
plot(kfit2, 'link')
```

**Estimated link function**

