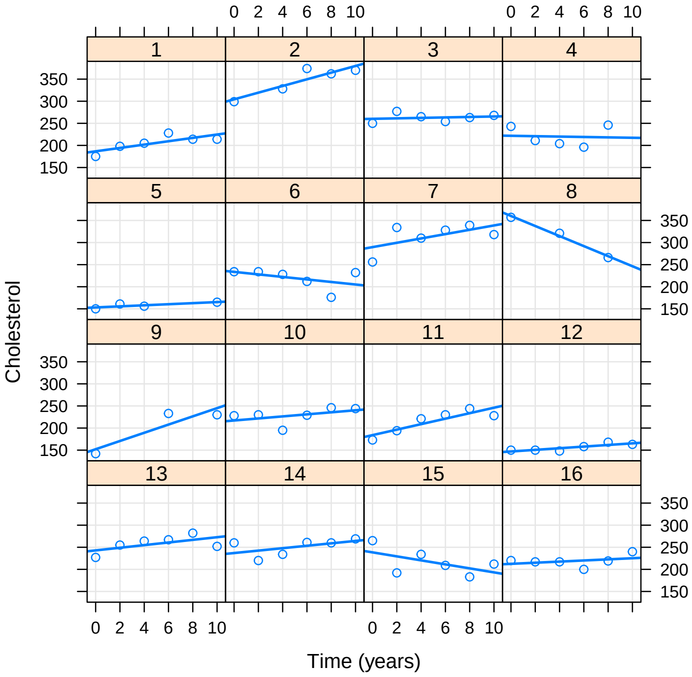
1 Prediction theory

There is a growing interest in applying these models to clinical prediction and personalized medicine. The goal of these applications is that model based predictions increase the chances of survival of the patients. Furthermore, these predictions become more accurate when more information becomes available. A typical application is to tailor the medical treatment based on the survival probabilities derived from our jointmodel, that is, to intervene when the medical prognosis deteriorates.

$$\beta$$

is my favourite section of the r language documentation.

```
> library(lme4)
> library(smoothLME)
> library(lattice)
> library(lcmm)
> data(Cholesterol, package = "qrLMM")
> Cholesterol$resp <- Cholesterol$cholst/100
> Cholesterol$time <- (Cholesterol$year - 5)/10
> Cholesterol.samp <- subset(Cholesterol, newid %in% 1:16)
> Cholesterol.samp$newid <- as.factor(Cholesterol.samp$newid)
> xyplot(cholst ~ year | newid, data = Cholesterol.samp,
+        type = c("p", "r"), lwd = 2, layout = c(4, 4),
+        as.table = TRUE, ylab = "Cholesterol", grid = TRUE,
+        xlab = "Time (years)")
>
```



2 Joint model

We fit the joint model to the pbc data.

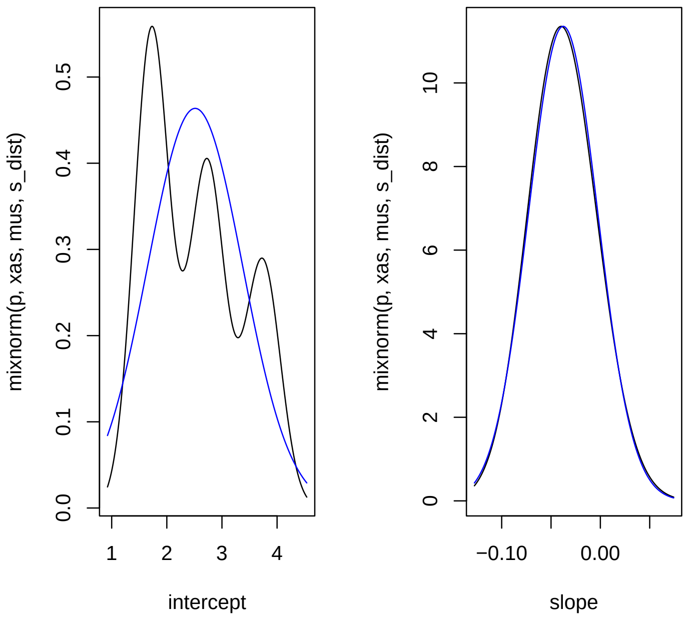
```
> library(smoothJM)
> data("pbc2", package = "JM")
> pbc2$lsgot <- log(pbc2$SGOT)
> pbc2$id <- as.numeric(pbc2$id)
> model1 <- smoothJM(lsgot ~ year +  (1 + year | id), maxiter = 250,
+ survival = Surv(years, status2) ~ factor(drug) + factor(sex), data = pbc2, n1 = 3, n2 = 3,
> model1$survmodel

              coefficient        SE          z      p.value
drugD-penicil  0.03069024 0.1714289  0.1790261 0.857917216
sexfemale     -0.67415357 0.2195564 -3.0705258 0.002136822
bm_intercept   1.02607070 0.1213681  8.4542062 0.000000000
bm_slope       1.52222208 0.1292386 11.7783890 0.000000000
```

this is the end of the document and a test.

3 Aids example

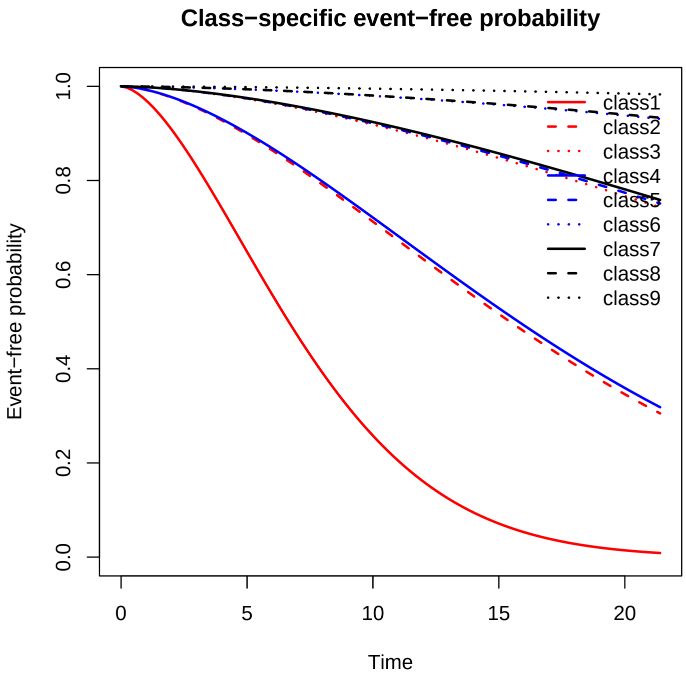
```
> data(aids, package="JM")
> aids$CD4W <- sqrt(aids$CD4)
> aids$patient <- as.numeric(aids$patient)
> col_v <- vector(length=9)
> lty_v <- vector(length=9)
> c1 <- c("red","blue","black")
> for( i in 1:3){
+    for( j in 1:3){
+       col_v[(i-1)*3+j]<-c1[i]
+       lty_v[(i-1)*3+j]<-j
+    }
+ }
> examp <- smoothJM(CD4W~ obstime + (1+obstime|patient), survival = Surv(Time, death) ~ fact
> oldpar <- par()
> par(mfrow=c(1,2))
> plotrandomeffects(examp, which = "intercept", xlab = "intercept")
> plotrandomeffects(examp, which = "slope", xlab = "slope")
> par(mfrow=oldpar$mfrow)
> examp$crosstable
```

```
          [,1]       [,2]       [,3]
[1,] 0.2726611 0.12111436 0.05379824
[2,] 0.1958540 0.08699714 0.03864359
[3,] 0.1406831 0.06249055 0.02775791
```



Survival plot

```
> plot(examp$jlcmm, which = 'survival',col = col_v,lty = lty_v, lwd = 2)
```


Class−specific event−free probability
Event−free probability
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
Time
class1
class2
class3
class4
class5
class6
class7
class8
class9