aneur: generalized additive mixture model (GAMM)

Aortic Aneurysm Progression Data

This dataset contains longitudinal measurements of grades of aortic aneurysms, measured by ultrasound examination of the diameter of the aorta.

A data frame containing 4337 rows, with each row corresponding to an ultrasound scan from one of 838 men over 65 years of age.

- ptnum (numeric) Patient identification number
- age (numeric) Recipient age at examination (years)
- diam (numeric) Aortic diameter
- state (numeric) State of aneurysm.

The states represent successive degrees of aneurysm severity, as indicated by the aortic diameter.

- State 1 Aneurysm-free < 30 cm
- State 2 Mild aneurysm 30-44 cm
- State 3 Moderate aneurysm 45-54 cm
- State 4 Severe aneurysm > 55 cm

683 of these men were aneurysm-free at age 65 and were re-screened every two years. The remaining men were aneurysmal at entry and had successive screens with frequency depending on the state of the aneurysm. Severe aneurysms are repaired by surgery.

```
data(aneur)
attach(aneur)
head(aneur)
```

```
##   ptnum      age diam state
## 1     1 60.00000   29     1
## 2     1 65.47671   29     1
## 3     1 67.50411   29     1
## 4     1 70.04384   29     1
## 5     1 72.07671   29     1
## 6     1 74.08767   29     1
```

```
tail(aneur)
```

```
##      ptnum      age diam state
## 4332   838 73.40822   43     2
## 4333   838 73.61644   43     2
## 4334   838 73.87671   42     2
## 4335   838 74.05753   43     2
## 4336   838 74.31507   41     2
## 4337   838 74.56712   40     2
```

```r
#help(aneur)
dim(aneur)
```

```
## [1] 4337    4
```

```r
(N = n_distinct(aneur$ptnum))    # subjects
```

```
## [1] 838
```

```r
(K = max(table(aneur$ptnum)))    # times
```

```
## [1] 21
```

```r
table(table(aneur$ptnum))
```

```
##
##   2   3   4   5   6   7   8   9  10  11  12  14  15  16  17  18  19  21
## 121 107  99  96 260  97  12  12   9   5   2   5   5   3   1   2   1   1
```

```r
J = 4   # categories
### data having width format representation
Y_diam = array(NA,dim=c(N,K))
Y_state = array(NA,dim=c(N,K))
X_age = array(NA,dim=c(N,K))
Ki = table(aneur$ptnum)
Ni = c(0,cumsum(Ki))+1
for(i in 1:N){
    aneur_i = aneur[aneur$ptnum==i,]
    for(k in 1:Ki[i]){
        Y_diam[i,k] = aneur_i$diam[k]
        Y_state[i,k] = aneur_i$state[k]
        X_age[i,k] = aneur_i$age[k]
    }
}
```

```r
### see some data having width format representation
(Y_diam[16:18,1:8])
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]   29   29   29   29   29   29   29   NA
## [2,]   29   29   29   29   NA   NA   NA   NA
## [3,]   29   29   34   NA   NA   NA   NA   NA
```

```r
(Y_state[16:18,1:8])
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    1    1    1    1    1    1    1   NA
## [2,]    1    1    1    1   NA   NA   NA   NA
## [3,]    1    1    2   NA   NA   NA   NA   NA
```

```r
(X_age[16:18,1:8])
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7] [,8]
## [1,]   60 65.40822 67.40822 70.04932 72.07123 74.06575 76.09041   NA
## [2,]   60 65.38082 67.38082 70.02192       NA       NA       NA   NA
## [3,]   60 65.47123 67.47123       NA       NA       NA       NA   NA
```

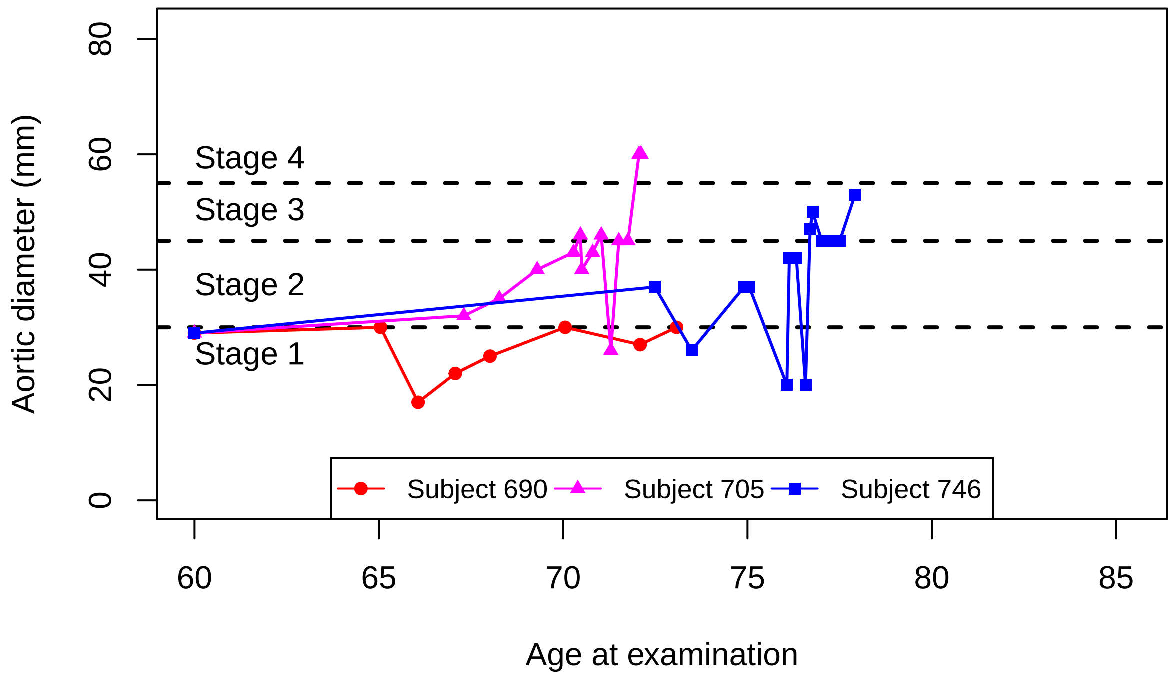
```r
(Ki[16:18])
```

```
## 
## 16 17 18 
##  7  4  3
```

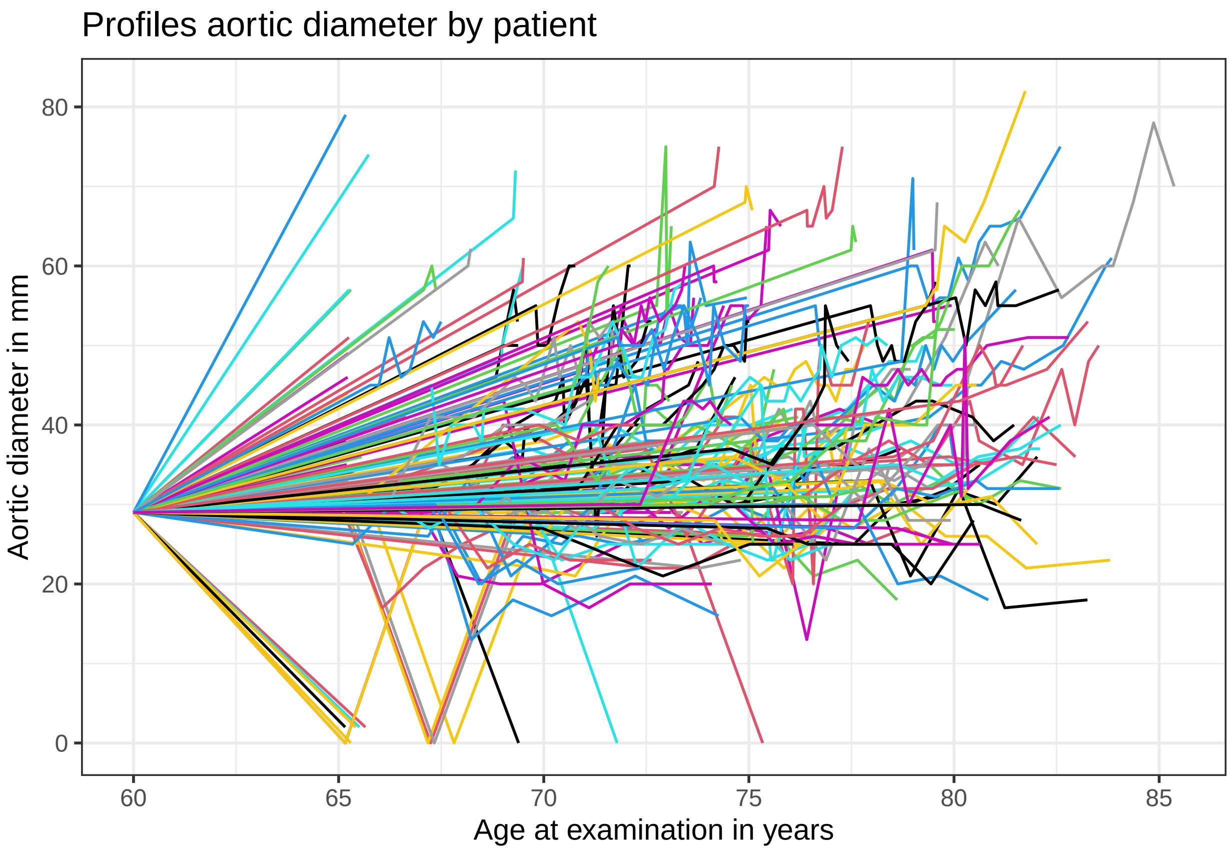
```r
### Considering only data having more than one screen (diam!=29, or diam<29 & dim>29)
idx3 = c()
for(i in 1:N){
  if( min(Y_diam[i,1:Ki[i]])!=max(Y_diam[i,1:Ki[i]])){
    idx3 = c(idx3,i)
  }
}
Y3_diam = Y_diam[idx3,]
Y3_state = Y_state[idx3,]
X3_age = X_age[idx3,]
N3 = length(idx3)
Ki3 = Ki[idx3]
### data used for the analysis
aneur3 = aneur%>%filter(aneur$ptnum%in%idx3)
```

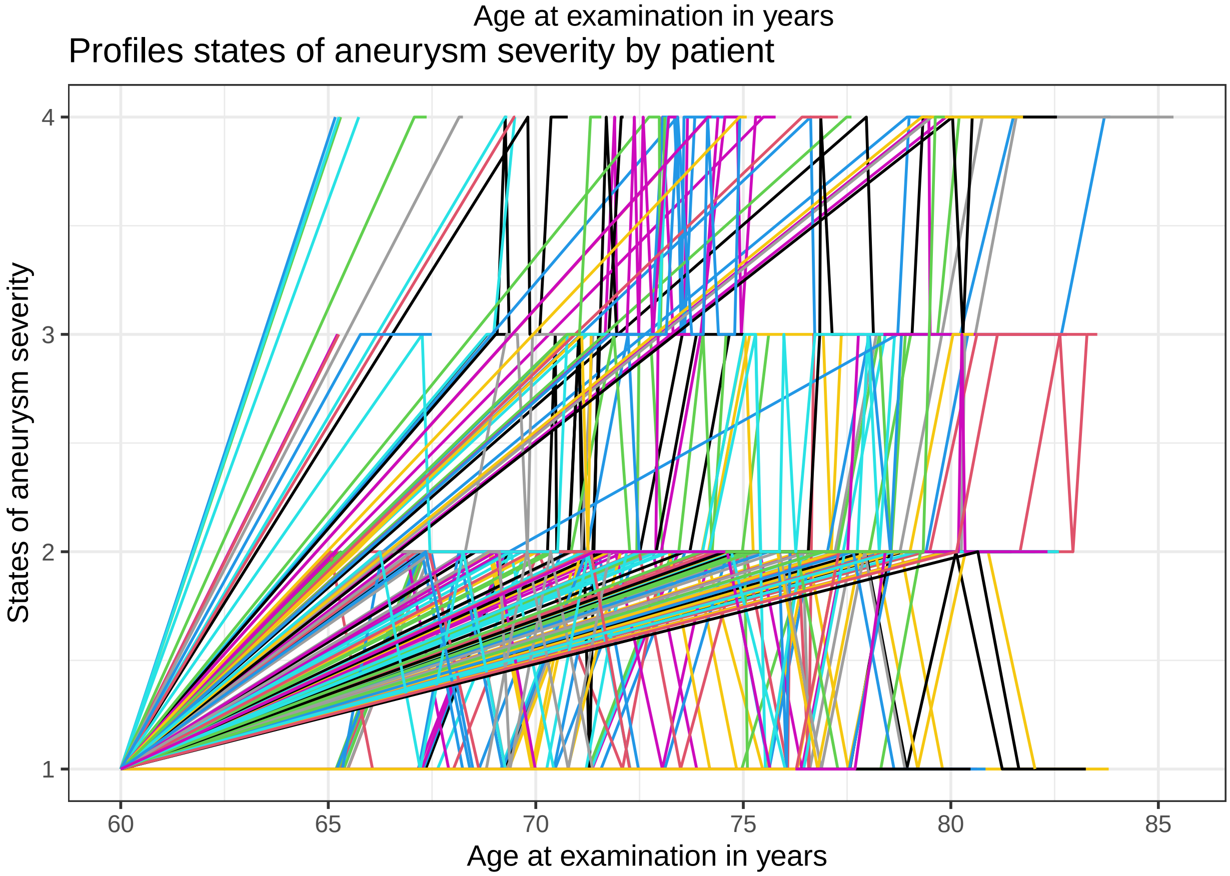
```r
### plot some subjects
Ki3[c(81,96,137)]
```

```
## 
## 690 705 746 
##   8  15  15
```




```
ggplot(data=aneur3, mapping=aes(x=age,y=diam,group=ptnum)) +
  geom_line(color=aneur3$ptnum) + theme_bw() +
  xlab("Age at examination in years") + ylab("Aortic diameter in mm") +
  ggtitle("Profiles aortic diameter by patient")
ggplot(data=aneur3, mapping=aes(x=age,y=state,group=ptnum)) +
  geom_line(color=aneur3$ptnum) + theme_bw() +
  xlab("Age at examination in years") + ylab("States of aneurysm severity") +
  ggtitle("Profiles states of aneurysm severity by patient")
```


Profiles aortic diameter by patient
Aortic diameter in mm
0
20
40
60
80
60
65
70
75
80
85
Age at examination in years


Profiles states of aneurysm severity by patient
States of aneurysm severity
1
2
3
4
60
65
70
75
80
85
Age at examination in years

Generalized additive mixture model (GAMM)

It is possible to consider a continuous response variable (`diam'') or an ordinal response variable` (state“), where the explanatory variable is continuous (“age”). Then, the GAMM models are the following.

The GAMM for the continuous response variable $diam_{it}$, with random slope:

$$\begin{aligned} diam_{it} &= \beta_0 + f_1(age_{it}) + age_{it} \times b_{1i} + \varepsilon_{it}, \\ & \quad b_{1i} \sim \mathrm{N}(0, \psi^2), \qquad \varepsilon_{it} \sim \mathrm{N}(0, \sigma^2), \end{aligned}$$

where β_0 is the intercept and f_1 is the smoothing function for the common fixed effects; b_{1i} is the random slope to consider that subjects have different growth rates; ψ^2 is the variance for the random slope, and σ^2 is the variance for the errors. Note that random intercepts are not needed to model the data, and observations for the same subject are independent, i.e., ε_{it_1} is independent of ε_{it_2} for $\varepsilon_{it_1} \neq \varepsilon_{it_2}$.

The ordinal response $state_{it}$ is modelled in terms of the cumulative probabilities $P(state_{it} \leq j|b_i)$ by using the proportional odds model,

$$P(state_{it} \leq j|b_i) \quad = \quad \eta_{it,j},$$

with $j = 1, 2, 3$, subject to

$$\eta_{it,j} \quad = \quad \kappa_j + f_1(age_{it}) + age_{it} \times b_{1i}, \qquad b_{1i} \sim \mathrm{N}(0, \psi^2),$$

where the constraints are such that f_1 is a non-decreasing smoothing function and $b_{1i} > 0$, and for the breakpoints $\kappa_j < \kappa_{j+1}$.

```
### center variables
y = aneur3$diam -29
x1 = aneur3$age -60
x2 = aneur3$age -60
id = as.numeric(as.factor(aneur3$ptnum))

n = length(y)
N = n_distinct(id)
Ni = c(0,cumsum(table(id)))+1
k1 = 4
k2 = 4
knots1 = quantile(x1, c(0.33,0.67))
knots2 = quantile(x2, c(0.33,0.67))
```

Generate the design matrix X for the penzalized B-splines

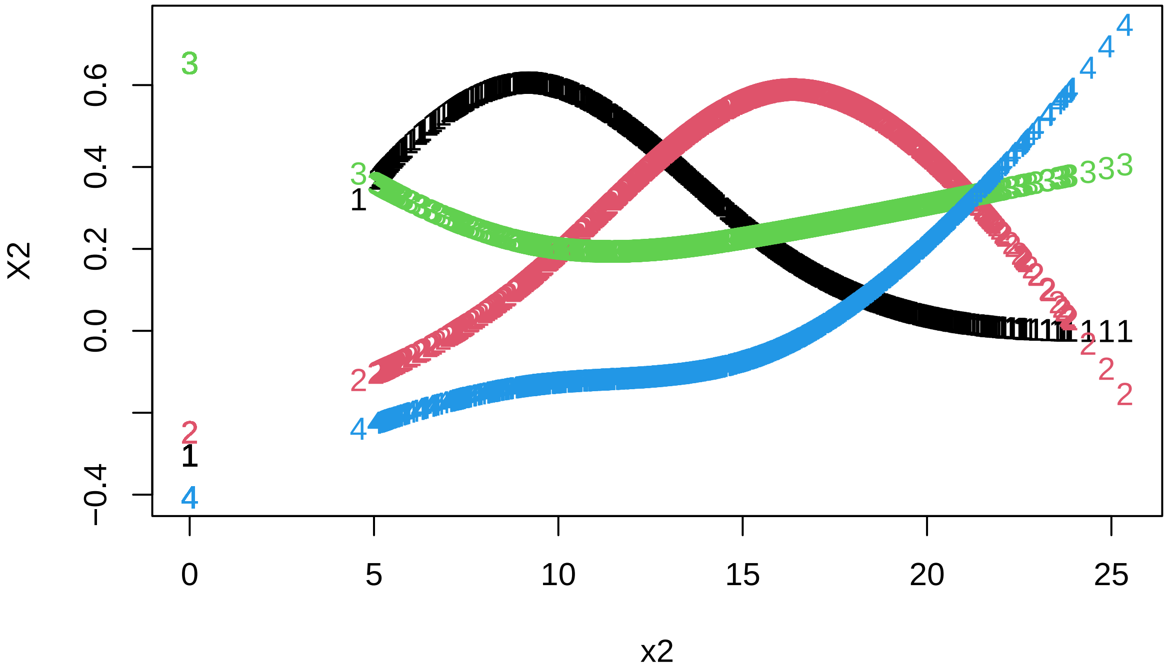
Note that the smoothing function $f(x)$ is represented as:

$$f(x) \quad = \quad \sum_{j=1}^{h_1} \beta_{1j} I_{1j}(x)$$

for β_{1j} unknown parameters.

The number of knots K is chosen a priori.

```
# Generate a basis matrix for Natural Cubic Splines
X2 <- ns(x = x2, knots = knots2, intercept = TRUE)
###X2 = (X2-mean(X2))/sd(X2)
matplot(x2, X2)
```

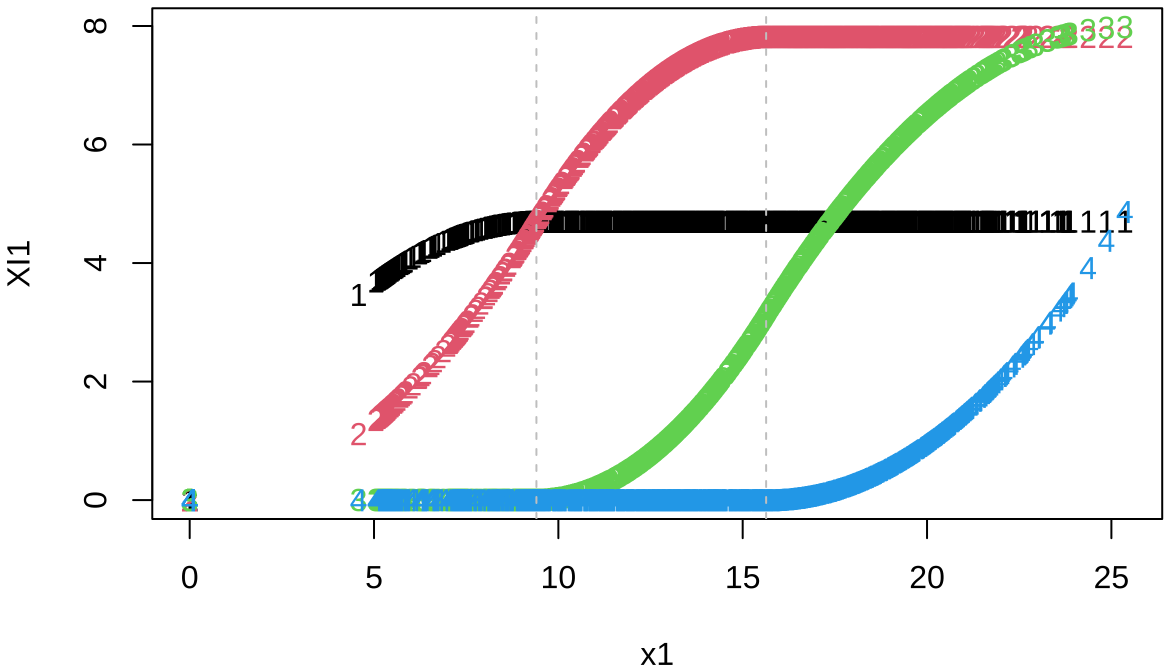


Generate the design matrix XI for the penzalized I-splines

Note that the smoothing function $f(x)$ is represented as:

$$
\begin{aligned}
f_1(x_1) & \quad \sum_{j=1}^{h_1} \beta_{1j} I_{1j}(x_1) \\
I_{1j}(x_1) & = \int_{x_0}^{x_1} B_{1j}(u) d_u
\end{aligned}
$$

```
### ibs: integrated basis splines
### degree = 3 cubic splines
XI1 <- ibs(x1, knots = knots1, degree = 1, intercept = TRUE)
###XI1 = (XI1-mean(XI1))/sd(XI1)
matplot(x1, XI1)
abline(v = knots1, h = knots1, lty = 2, col = "gray")
```



Define the penalizations $S1$ and $S2$

The flexibility of f is controlled by K, from a quadratic penalization as:

$$\sum_j \lambda_j \beta^T S_j \beta$$

where the S_j are matrix with known coefficients, and parameters λ_j are smoothing parameters that should be estimated.

```
#No es el óptimo, pero funciona.
#"k" number of b-splines
#"d" order of difference

# Produce the matrix of differences:
diffMatrix = function(k, d = 2){
  if( (d<1) || (d %% 1 != 0) )stop("d must be a positive integer value");
  if( (k<1) || (k %% 1 != 0) )stop("k must be a positive integer value");
  if(d >= k)stop("d must be lower than k");
  out = diag(k);
  for(i in 1:d){
    out = diff(out);
  }
  return(out)
}
(D1 = diffMatrix(k=k1, d=2))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1   -2    1    0
## [2,]    0    1   -2    1
```

```r
(D2 = diffMatrix(k=k2, d=2))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1   -2    1    0
## [2,]    0    1   -2    1
```

```r
### Matrix of penalization
(S1 = t(D1)%*%D1 + diag(1,k1)*10e-4)
```

```
##        [,1]   [,2]   [,3]   [,4]
## [1,]  1.001 -2.000  1.000  0.000
## [2,] -2.000  5.001 -4.000  1.000
## [3,]  1.000 -4.000  5.001 -2.000
## [4,]  0.000  1.000 -2.000  1.001
```

```r
(S2 = t(D2)%*%D2 + diag(1,k2)*10e-4)
```

```
##        [,1]   [,2]   [,3]   [,4]
## [1,]  1.001 -2.000  1.000  0.000
## [2,] -2.000  5.001 -4.000  1.000
## [3,]  1.000 -4.000  5.001 -2.000
## [4,]  0.000  1.000 -2.000  1.001
```

GAMM with monotone constrains

Data, inits and parameters

```
data.lme.add.incr <- list( y = y , id = id ,
               n = length(y) , N = N , Ni = Ni, k1=k1,
              XI1 = XI1, x1 = x1,
              zero = rep(0,1+k1),  S1 = S1  )
inits.lme.add.incr <- function(){   list(
  "b1" = abs(rnorm(k1,0,0.1)),
  "invtau2" = rgamma(1,1,1) ,
  "lambda" = rgamma(1,1,1) ,
  "invsig2" = rgamma(1,1,1)
)   }
param.lme.add = c("b0","b1", "invtau2","tau2","tau", "lambda","rho", "invsig2","sig2","sigma")
```

Fit the model

```
fit.lme.add.incr.reslope <- stan("gamm_aneur_lme_add_incr_reslope.stan",
            data=data.lme.add.incr,
            chains=3,warmup=300,iter=600,thin=2,cores=4,
            init= inits.lme.add.incr)
```

Results

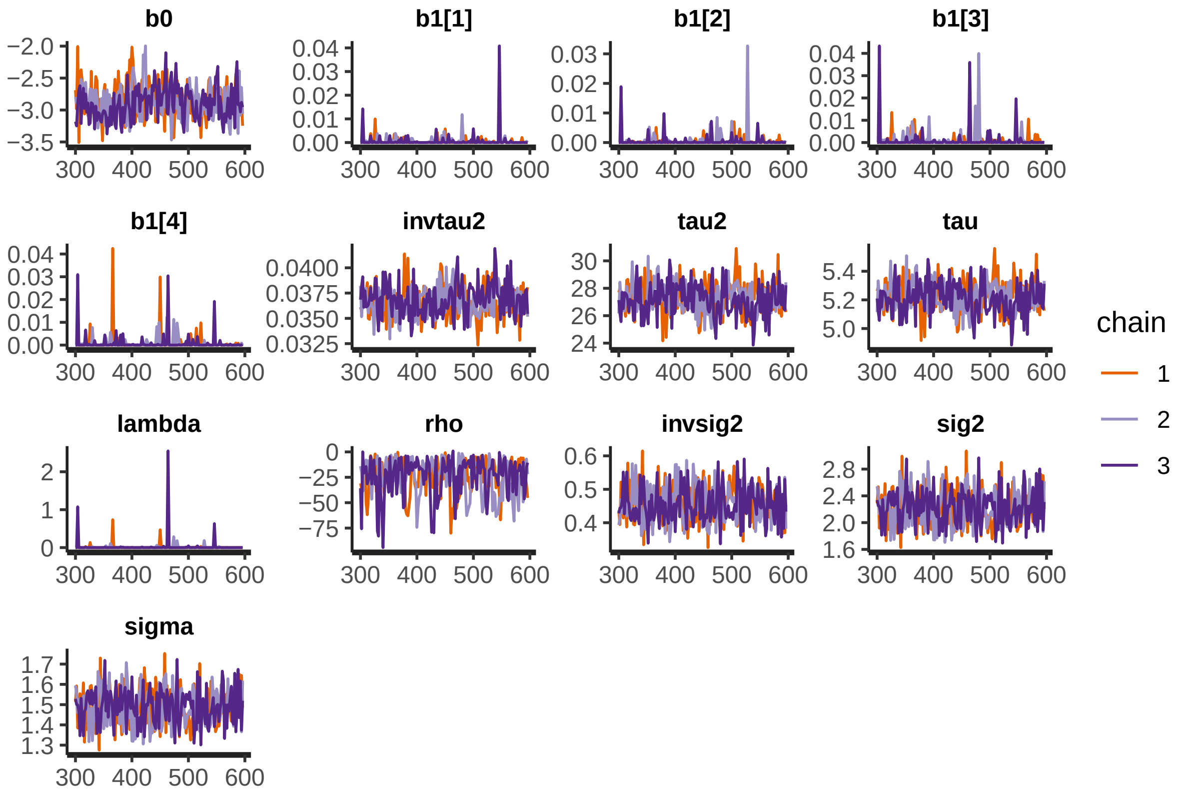
```
print(fit.lme.add.incr.reslope, pars=param.lme.add)
```

```
## Inference for Stan model: gamm_aneur_lme_add_incr_reslope.
## 3 chains, each with iter=600; warmup=300; thin=2;
## post-warmup draws per chain=150, total post-warmup draws=450.
##
##            mean se_mean    sd   2.5%    25%    50%    75% 97.5% n_eff Rhat
## b0        -2.87    0.02  0.26  -3.33  -3.06  -2.89  -2.70 -2.34   246 1.01
## b1[1]      0.00    0.00  0.00   0.00   0.00   0.00   0.00  0.00   478 1.00
## b1[2]      0.00    0.00  0.00   0.00   0.00   0.00   0.00  0.00   456 1.00
## b1[3]      0.00    0.00  0.00   0.00   0.00   0.00   0.00  0.01   414 1.00
## b1[4]      0.00    0.00  0.00   0.00   0.00   0.00   0.00  0.01   483 1.00
## invtau2    0.04    0.00  0.00   0.03   0.04   0.04   0.04  0.04   277 1.01
## tau2      27.25    0.07  1.12  25.07  26.50  27.21  28.01 29.50   281 1.01
## tau        5.22    0.01  0.11   5.01   5.15   5.22   5.29  5.43   280 1.01
## lambda     0.02    0.01  0.14   0.00   0.00   0.00   0.00  0.05   465 1.00
## rho      -24.31    1.31 17.44 -65.68 -33.59 -20.07 -10.78 -3.07   177 1.01
## invsig2    0.46    0.00  0.05   0.36   0.42   0.46   0.49  0.57   444 1.00
## sig2       2.22    0.01  0.26   1.76   2.03   2.20   2.38  2.75   464 0.99
## sigma      1.49    0.00  0.09   1.33   1.42   1.48   1.54  1.66   458 0.99
##
## Samples were drawn using NUTS(diag_e) at Fri Oct 27 10:22:46 2023.
## For each parameter, n_eff is a crude measure of effective sample size,
```

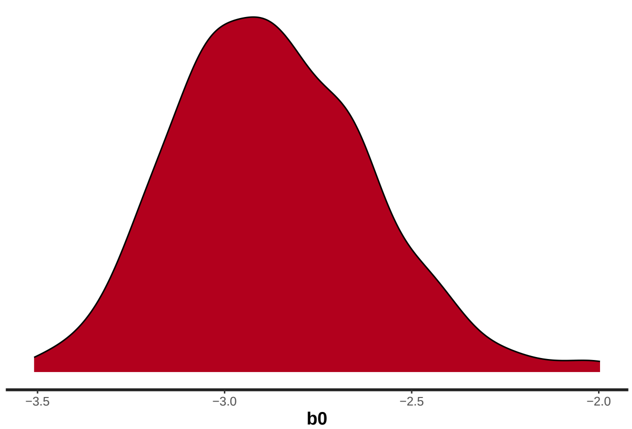
```
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

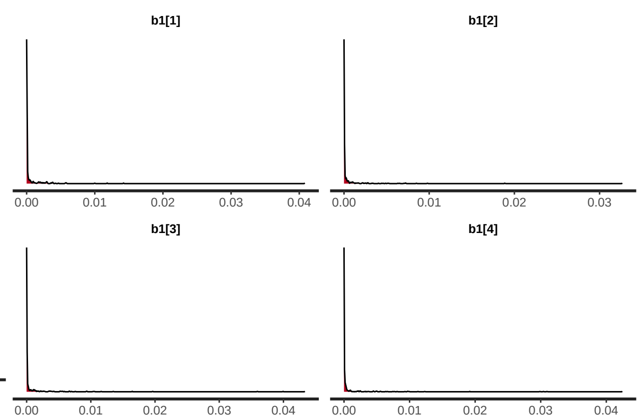
Plots

```
stan_trace(fit.lme.add.incr.reslope, pars=param.lme.add)
```

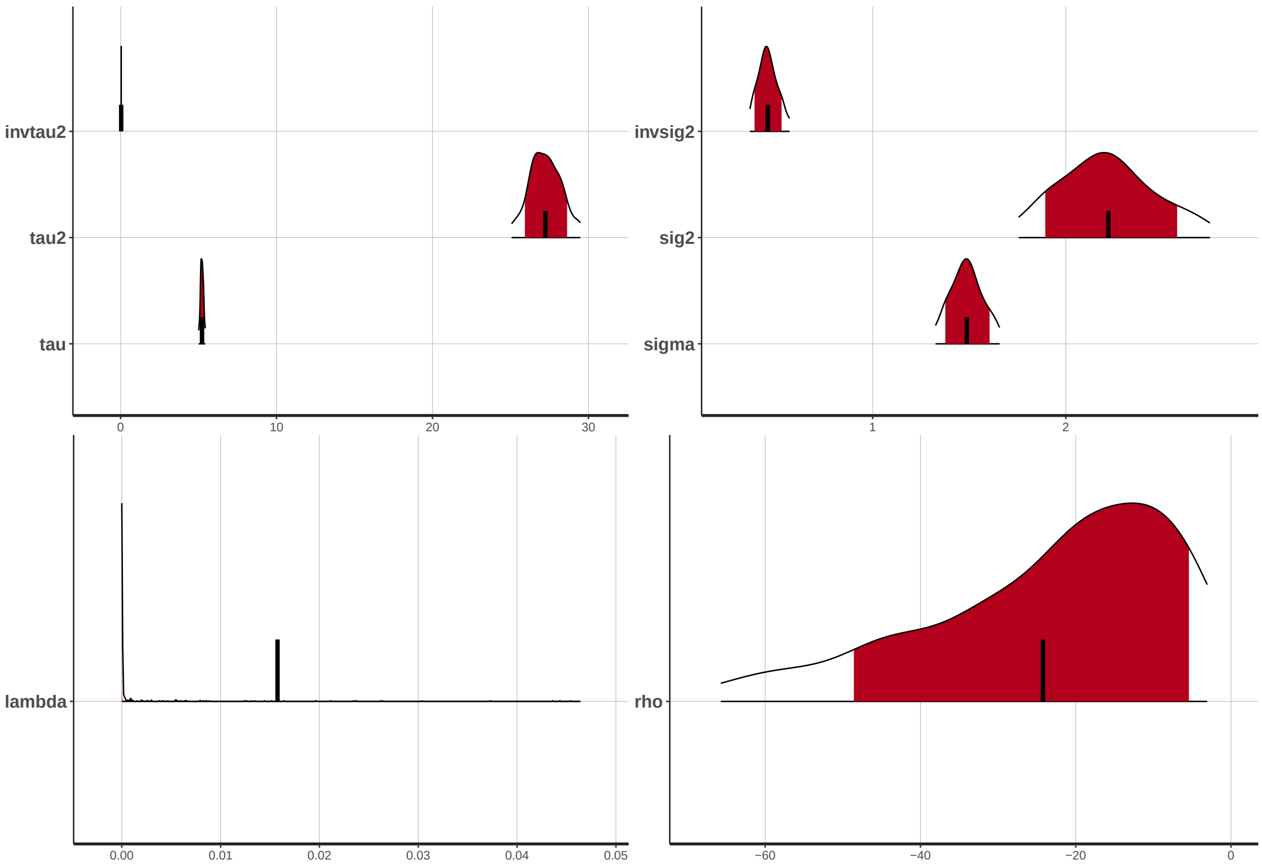



```
stan_dens(fit.lme.add.incr.reslope, pars=c("b0"))
stan_dens(fit.lme.add.incr.reslope, pars=c("b1"))
```

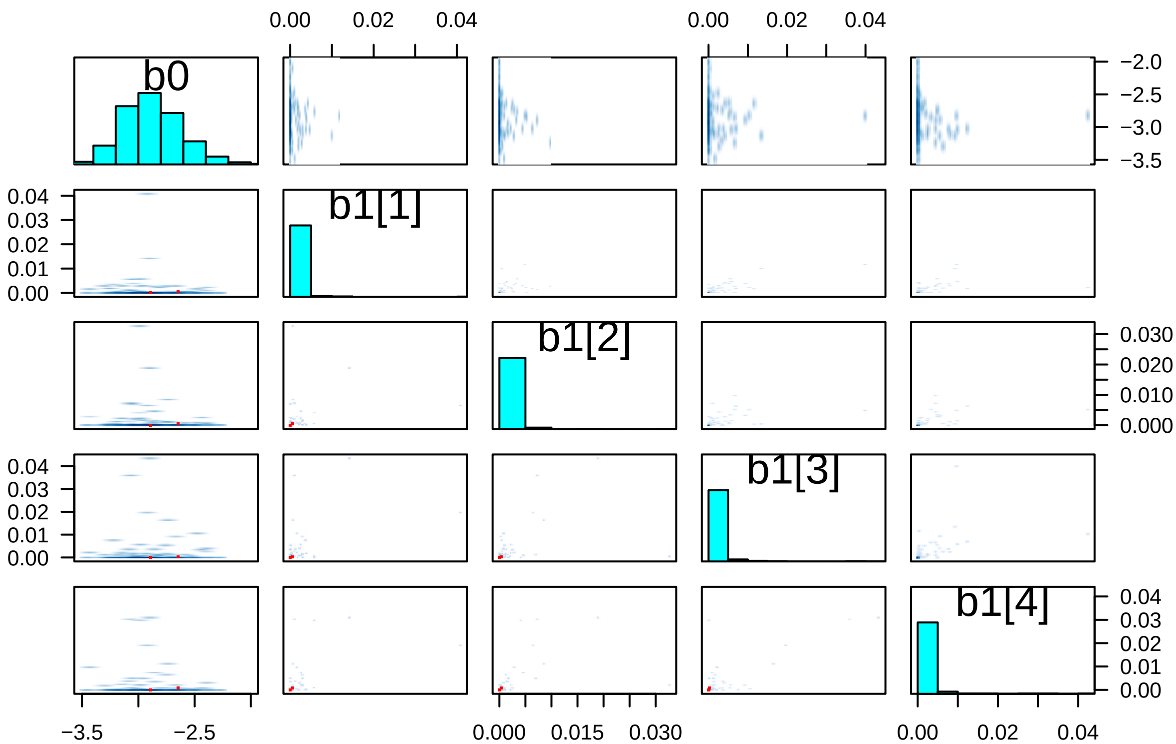




```r
stan_plot(fit.lme.add.incr.reslope, point_est = "mean", show_density = TRUE,
          pars=c("invtau2","tau2","tau") )
stan_plot(fit.lme.add.incr.reslope, point_est = "mean", show_density = TRUE,
          pars=c("invsig2","sig2","sigma") )
stan_plot(fit.lme.add.incr.reslope, point_est = "mean", show_density = TRUE,
          pars=c( "lambda") )
stan_plot(fit.lme.add.incr.reslope, point_est = "mean", show_density = TRUE,
          pars=c("rho") )
```

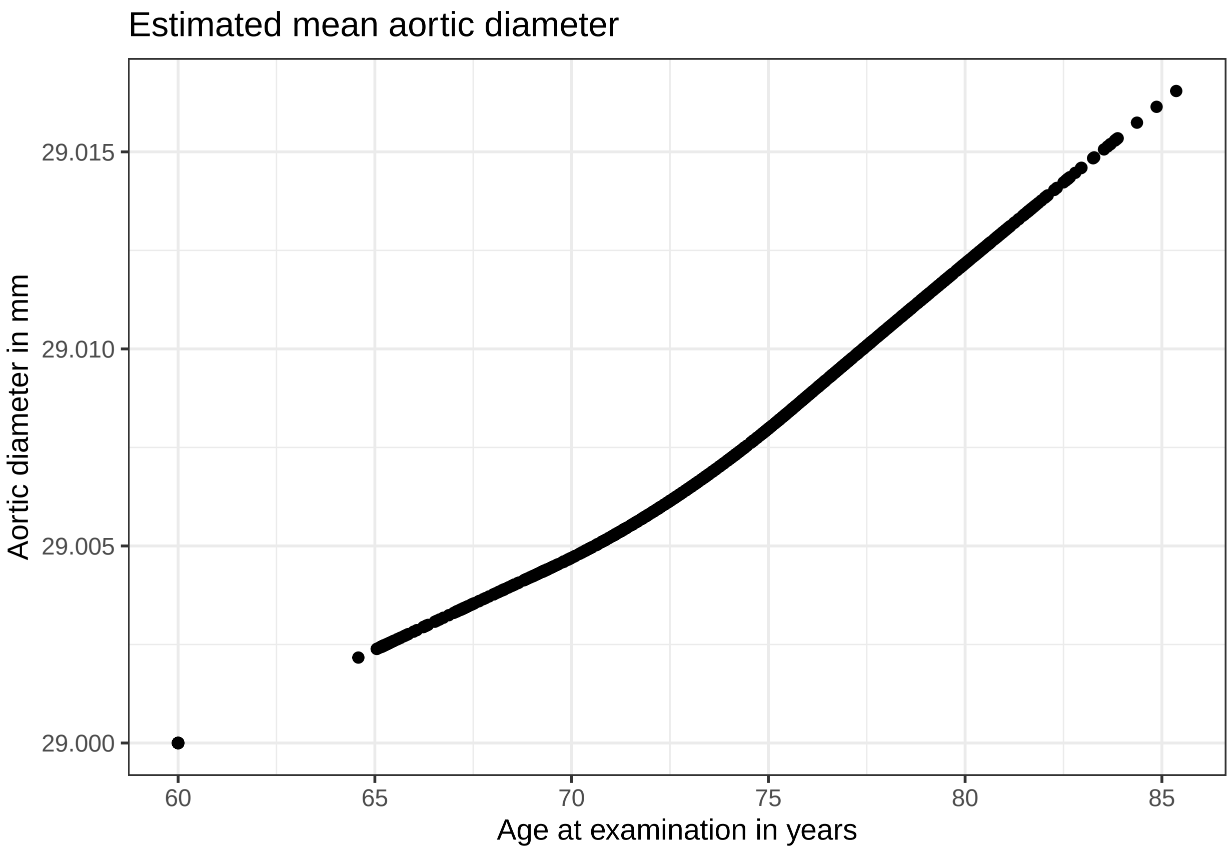



```r
pairs(fit.lme.add.incr.reslope, pars = c("b0","b1"), las = 1)
```



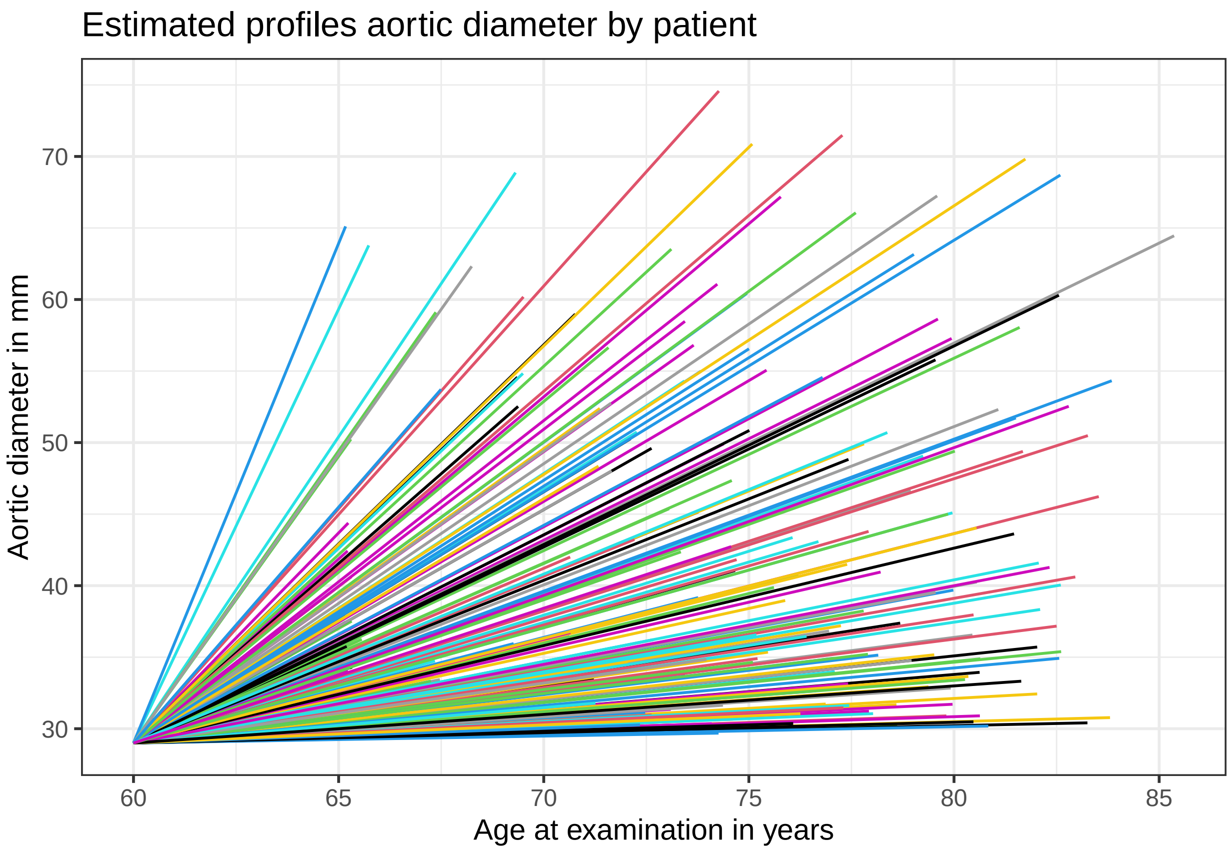

```
mu1 = get_posterior_mean(fit.lme.add.incr.reslope,"mu1")
aneur3$mu1_all = 29 + mu1[,"mean-all chains"]

ggplot(data=aneur3,
       mapping=aes(x=age,y=mu1_all,group=ptnum)) +
  geom_point() +
  theme_bw() +
  xlab("Age at examination in years") + ylab("Aortic diameter in mm") +
  ggtitle("Estimated mean aortic diameter")
```




```r
mu2 = get_posterior_mean(fit.lme.add.incr.reslope,"mu2")
aneur3$mu2_all = 29 + mu2[,"mean-all chains"]

ggplot(data=aneur3,
       mapping=aes(x=age,y=mu2_all,group=ptnum)) +
  geom_line(color=aneur3$ptnum) +
  theme_bw() +
  xlab("Age at examination in years") + ylab("Aortic diameter in mm") +
  ggtitle("Estimated profiles aortic diameter by patient")
```



Information criteria

```
### fitted model
loo_sample_lme = fit.lme.add.incr.reslope
### we have to extract those log-likelihood terms that we so carefully had Stan calculate for us:
log_lik_lme =extract_log_lik(loo_sample_lme, merge_chains = F)
r_eff_lme =relative_eff(log_lik_lme)
###  look at the results for each model, first the one with mu estimated:
(loo_lme <- loo(log_lik_lme, r_eff=r_eff_lme))
```

```
## Warning: Some Pareto k diagnostic values are too high. See help('pareto-k-diagnostic') for details.
```

```
##
## Computed from 450 by 1387 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -4352.2  59.2
## p_loo       150.8  13.0
## looic      8704.4 118.4
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
```

```
## (-Inf, 0.5]   (good)     1279  92.2%   77
##  (0.5, 0.7]   (ok)         73   5.3%   35
##    (0.7, 1]   (bad)        28   2.0%   8
##    (1, Inf)   (very bad)    7   0.5%   2
## See help('pareto-k-diagnostic') for details.
```