MATH 598: TOPICS IN STATISTICS

FLEXIBLE REGRESSION USING REVERSIBLE JUMP MCMC

A flexible linear regression model for response data Y is a model where the conditional mean of Y is assumed to take the form

$$\mathbb{E}[Y|x] = \beta_{K0} + \sum_{k=1}^{K} \beta_{Kk} g(x; \eta_{Kk})$$

for regression coefficients $\beta_K = (\beta_{K0}, \beta_{K1}, \ldots, \beta_{KK})$ where $g(.;.)$ is a *basis* function. For example, we might choose

$$g(x;\eta) = \begin{cases} (x-\eta)^r & x \geq \eta \\ 0 & x < \eta \end{cases}$$

for *knot*-point η and $r > 0$. This formulation ensures a continuous model for the conditional mean.

With the usual assumption of Normal homoscedastic additive errors with variance σ^2, we have a standard linear model where

$$\mathbf{Y}|\mathbf{x}, \beta, \eta, \sigma^2 \sim Normal(\mathbf{X}_K(\eta)\beta, \sigma^2 \mathbf{I}_n)$$

where $\mathbf{X}_K(\eta)$ is the $n \times (K+1)$ design matrix from the linear model, which is a function of the K knot-points $(\eta_{K1}, \ldots, \eta_{KK})$. This model can easily be fit using least squares: using standard theory, the estimates of β_K are

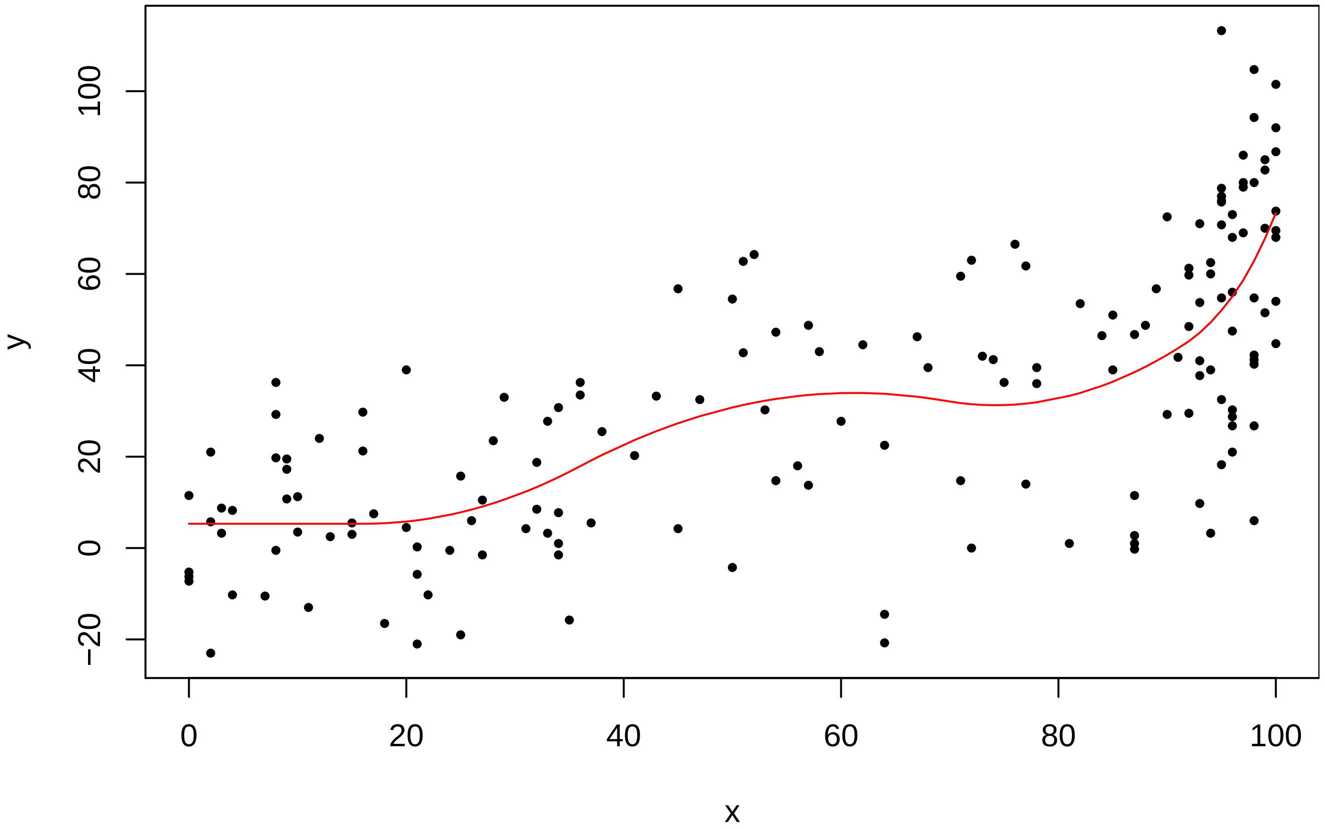
$$\widehat{\beta}_K = (\mathbf{X}_K(\eta)^\top \mathbf{X}_K(\eta))^{-1} \mathbf{X}_K(\eta)^\top \mathbf{y}$$

with fitted values

$$\widehat{\mathbf{y}} = \mathbf{X}_K(\eta)\widehat{\beta}_K.$$

In the example below, we use the cholostyramine data from the `bootstrap` package in `R`. For illustration, we choose $K = 5$ and $r = 2$, with knot-points chosen as quintiles of the observed x values.

```
library(bootstrap)
par(mar=c(4,4,2,0))
plot(cholost,type='p',pch=19,cex=0.5,xlab='x')
y<-cholost$y[order(cholost$z)]
x<-cholost$z[order(cholost$z)]
n<-length(y)

lam<-10
K<-5
eta<-quantile(x,prob=(1:K)/(K+1))
r<-2
Xmat<-rep(1,n)
for(k in 1:K){
        Xmat<-cbind(Xmat,(x-eta[k])^r*(x>eta[k]))
}
np<-ncol(Xmat)
XTX<-t(Xmat)%*%Xmat
be.hat<-solve(XTX) %*% (t(Xmat) %*% y)
y.fit<-Xmat %*% be.hat
sigsq.hat<-sum((y-y.fit)^2)/(n-np)
sig.hat<-sqrt(sigsq.hat)
lines(x,y.fit,col='red')
```



For a Bayesian analysis, and under the usual conjugate prior

$$\pi_0(\beta_K|\sigma^2) \equiv Normal(\xi, \sigma^2 \mathbf{L}^{-1}) \qquad \pi_0(\sigma^2) \equiv InverseGamma(a/2, b/2)$$

for positive definite matrix $\mathbf{L}$, we have that in the posterior

$$\pi_n(\beta_K|\sigma^2) \equiv Normal(\xi_n, \sigma^2 \mathbf{L}_n^{-1}) \qquad \pi_n(\sigma^2) \equiv InverseGamma(a_n/2, b_n/2)$$

where

$$\xi_n = \mathbf{L}_n^{-1}(\mathbf{L}\xi + \mathbf{X}^\top \mathbf{y}) \qquad \mathbf{L}_n = \mathbf{X}^\top \mathbf{X} + \mathbf{L}$$

and

$$a_n = a + n \qquad b_n = b + \mathbf{y}^\top \mathbf{y} + \xi^\top \mathbf{L}\xi - \xi_n^\top \mathbf{L}_n \xi_n.$$

In the analysis, we choose (exchangeable) independent priors on the β_K parameters and set

$$\mathbf{L} = \lambda \mathbf{I}_{K+1}$$

with $\lambda = 0.01$, and then choose $\xi = \mathbf{0}$ and $a = 10$, $b = 100$. Furthermore, we can compute the marginal likelihood

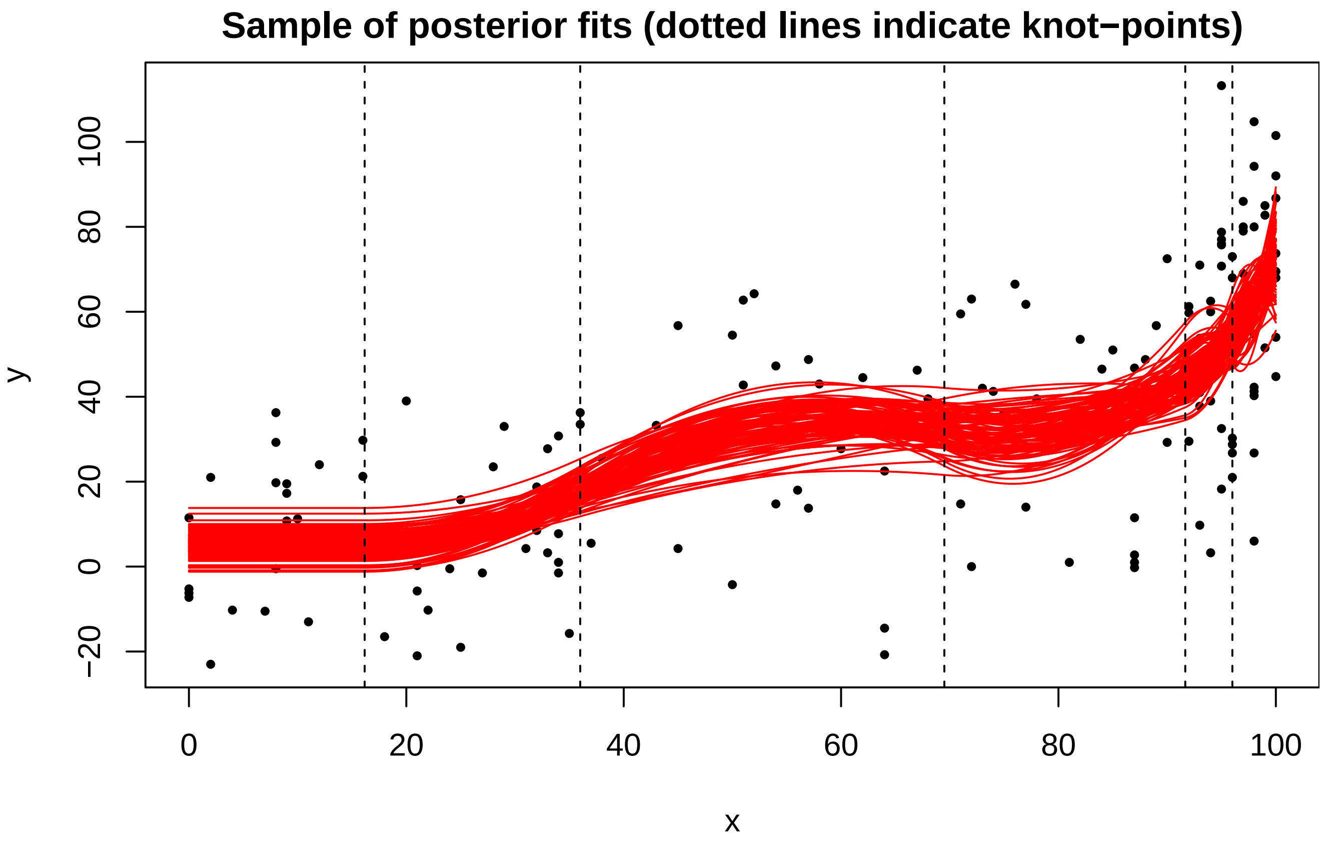
$$f(\mathbf{y}; \xi, \mathbf{L}, a, b) = \int \mathcal{L}_n(\beta_K, \sigma^2)\pi_0(\beta_K, \sigma^2)\, d\beta_K d\sigma^2$$

analytically for this model (see earlier `knitr` sheets). Up to a multiplicative constant that does not depend on K, the marginal likelihood takes the form

$$\frac{|\mathbf{L}|^{1/2}}{|\mathbf{L}_n|^{1/2}} \left\{ \frac{b_n}{2} \right\}^{-a_n/2}$$

The posterior distribution is analytically available, but we can easily sample from it to produce sampled fits.

```
nsamp<-100
library(mvnfast)
prior.L<-diag(rep(1/100,np))
prior.be0<-rep(0,np)
prior.a<-10
prior.b<-100
post.L<-XTX+prior.L
post.var<-solve(post.L)
post.mean<-post.var %*% (t(Xmat) %*% y + prior.L %*% prior.be0)
ssq<-(t(y) %*% y + t(prior.be0) %*% prior.L %*% prior.be0 -
        t(post.mean) %*% post.L %*% post.mean)[1,1]
post.a<-prior.a+n
post.b<-prior.b+ssq
sig.samp<-1/sqrt(rgamma(nsamp,post.a/2,post.b/2))
be.samp<-rmvn(nsamp,mu=rep(0,np),sigma=post.var)*sig.samp+
            t(matrix(post.mean,nrow=np,ncol=nsamp,byrow=F))
par(mar=c(4,4,2,0))
plot(cholost,type='p',pch=19,cex=0.5,xlab='x')
title('Sample of posterior fits (dotted lines indicate knot-points)')
nx<-1001
xvec<-seq(0,100,length=nx)
Xm<-rep(1,nx)
for(k in 1:K){
        Xm<-cbind(Xm,(xvec-eta[k])^r*(xvec>eta[k]))
}
y.fit.b<-t(Xm %*% t(be.samp[1:100,]))
for(i in 1:100){
        lines(xvec,y.fit.b[i,],col='red')
}
abline(v=eta,col='black',lty=2)
```



We can perform a more flexible analysis by allowing the number and position of the knot points to vary.

- We place a $Poisson(\gamma)$ prior on K; if $K = 0$ we have an intercept only model.
- Conditional on having K knots, we now construct a prior for $\eta_{K1}, \ldots, \eta_{KK}$ using the same construction as Green (1995): this helps to keep a more uniform spacing of the knots.
- We assume that the K knot points are the even order statistics derived from $2K+1$ points sampled uniformly on $(0, x_{\max})$.
- The density for this prior model is derived as follows: by standard results, if $X_1, \ldots, X_{2K+1}$ are independent $Uniform(0, x_{\max})$, the joint pdf of the full collection of order statistics is

  $$\frac{(2K+1)!}{x_{\max}^{2K+1}}$$

  on the support

  $$0 < x_{(1)} < x_{(2)} < \cdots < x_{(2K+1)} < x_{\max}$$
- We obtain the marginal distribution of the even order statistics by integrating out over $x_{(1)}, x_{(3)}, \ldots, x_{(2K+1)}$ to obtain the desired prior density

  $$\frac{(2K+1)!}{x_{\max}^{2K+1}} \times x_{(2)} \times \prod_{k=2}^{K} (x_{(2k)} - x_{(2(k-1))}) \times (x_{\max} - x_{(2K)})$$

Thus the prior on $\eta_{K1}, \ldots, \eta_{KK}$ that is used is

$$\frac{(2K+1)!}{x_{\max}^{2K+1}} \times \eta_{K1} \times \prod_{k=2}^{K} (\eta_{Kk} - \eta_{K(k-1)}) \times (\eta_{\max} - \eta_{KK}).$$

We carry out a transdimensional analysis using reversible jump MCMC, by considering Birth and Death moves as a pair.

- **Birth Move:** For a Birth move, if there are currently K knots, we
  - pick one of the $K + 1$ between-knot intervals uniformly at random, k say, between $\eta_{K(k-1)}$ and η_{Kk} (with $\eta_{K0} = 0$ and $\eta_{K(K+1)} = x_{\max}$).
  - pick a new knot position uniformly on $(\eta_{K(k-1)}, \eta_{Kk})$
  - increase the number of knots to $K + 1$.
- **Death Move:** If there are currently $K + 1$ knots,
  - pick one of the $K + 1$ knots, k say, and remove it;
  - decrease the number of knots to K.

Note that after each dimension changing move the design matrix changes, and increases or decreases by one column. We carry out the Birth/Death moves using the marginal likelihood form. For the Birth move, the acceptance probability is the minimum of 1 and

$$\frac{\pi_n(K+1, \eta_{K+1})}{\pi_n(K, \eta_K) \times \frac{1}{(\eta_{Kk} - \eta_{K(k-1)})}}$$

where $\pi_n(K, \eta_K)$ is the product

$$\text{Marginal likelihood} \times \text{Prior on } \eta_K \times \text{Prior on } K$$

and the term

$$\frac{1}{(\eta_{Kk} - \eta_{K(k-1)})}$$

arises from the selection of the position for the new knot. The Death move from $K + 1$ knots has acceptance probability that is given by the reciprocal of the expression above. In addition to the dimension-changing move, we also allow a knot relocation move

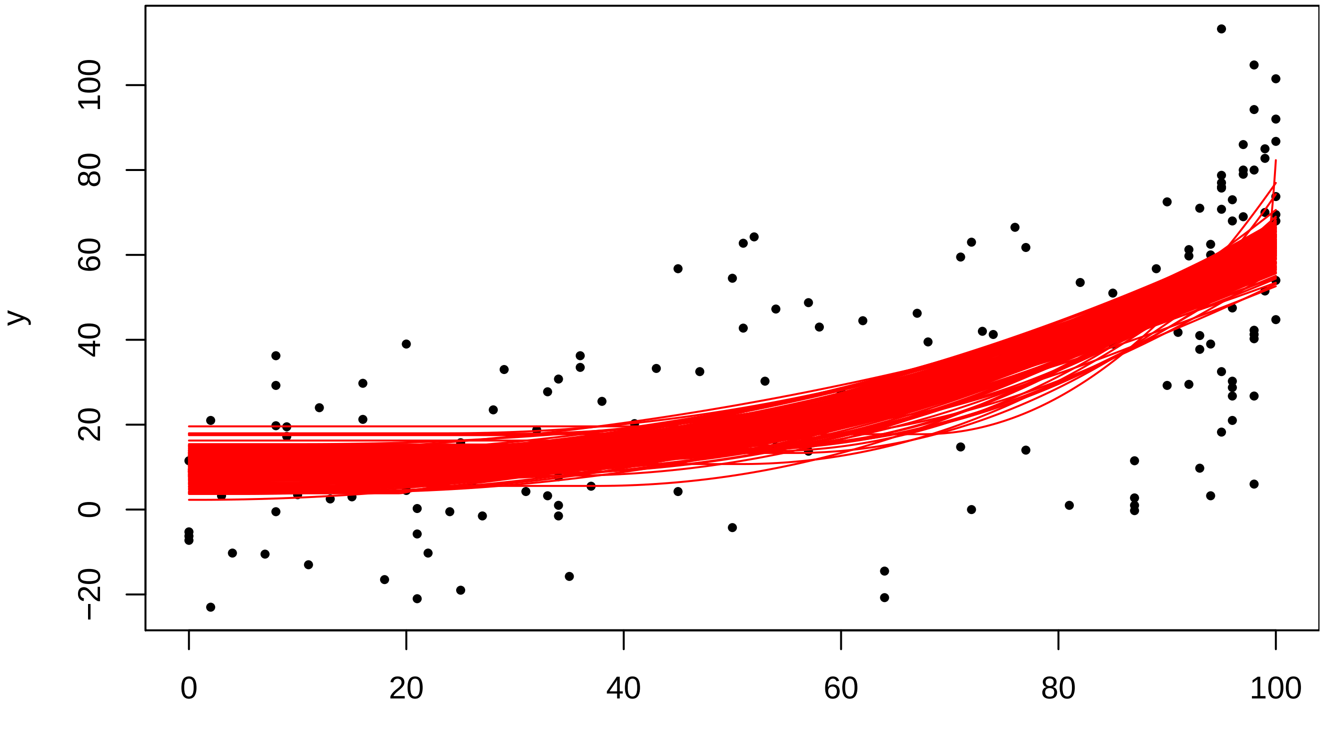
- **Knot relocation Move:** For a relocation move, if there are currently K knots, we
  - select one knot, k, uniformly from the K knots.
  - propose a new knot position uniformly on $(\eta_{K(k-1)}, \eta_{K(k+1)})$
  - form the new knot vector η_K^{new}.

The acceptance probability is the minimum of 1 and

$$\frac{\pi_n(K, \eta_K^{\text{new}})}{\pi_n(K, \eta_K)}.$$

```r
eta.density<-function(evec,Kv,xm){
        dvec<-diff(c(0,evec,xm))
        return(lgamma(2*Kv+2)+sum(log(dvec))-(2*Kv+1)*log(xm))
}

y<-bootstrap::cholost$y[order(cholost$z)]
x<-bootstrap::cholost$z[order(cholost$z)]
n<-length(y)
xmax<-100

prior.lam<-0.01
prior.a<-10
prior.b<-100
prior.gam<-10

old.K<-5  #This is the number of breaks, number of segments is K+1
eta<-sort(runif(2*old.K+1,0,100))
old.eta<-eta[2*(1:old.K)]

old.Xmat<-rep(1,n)
for(k in 1:old.K){
        old.Xmat<-cbind(old.Xmat,(x-old.eta[k])^r*(x>old.eta[k]))
}

old.np<-ncol(old.Xmat)
old.prior.L<-prior.lam*diag(1,old.np)
old.prior.be0<-rep(0,old.np)

XTX<-t(old.Xmat)%*%old.Xmat
be.hat<-solve(XTX) %*% (t(old.Xmat) %*% y)
y.fit<-old.Xmat %*% be.hat
sigsq.hat<-sum((y-y.fit)^2)/(n-old.np)
sig.hat<-sqrt(sigsq.hat)

old.be<-as.numeric(be.hat)
old.yfit<-old.Xmat %*% old.be
old.sig<-sig.hat

ysq<-t(y) %*% y
old.c<-(ysq + t(old.prior.be0) %*% old.prior.L %*% old.prior.be0 -
                t(post.mean) %*% post.L %*% post.mean)[1,1]

old.marg.like<-0.5*log(det(old.prior.L))-0.5*log(det(XTX+old.prior.L))-
                0.5*(n+prior.a)*log(0.5*(prior.b+old.c))

old.prior.K<-dpois(old.K,prior.gam,log=T)

old.prior.eta<-eta.density(old.eta,old.K,xmax)
```

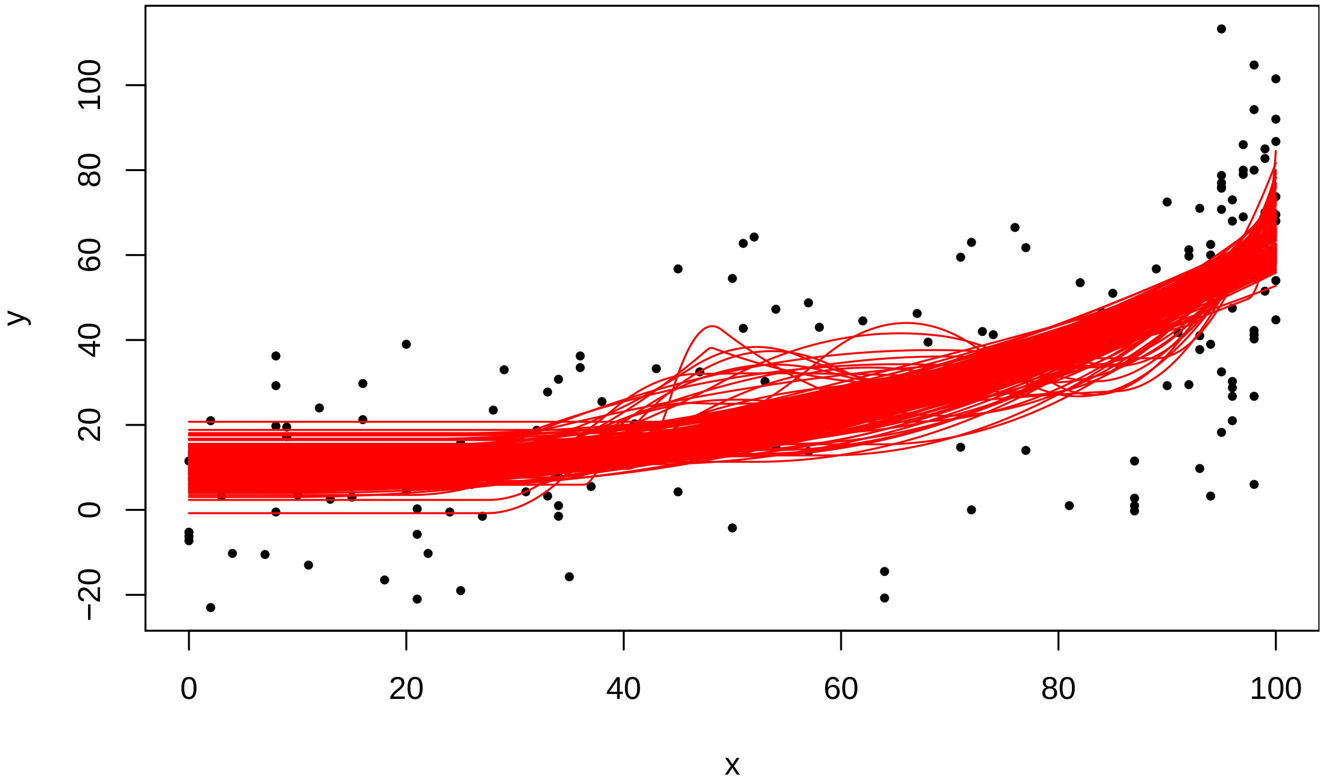
```
nburn<-1000
nsamp<-10000
nthin<-20
nits<-nburn+nthin*nsamp

sig.post<-rep(0,nsamp)
be.post<-eta.post<-matrix(0,nrow=nsamp,ncol=100)
K.post<-rep(0,nsamp)
ico<-0

par(mar=c(4,4,2,0))
plot(x,y,type='p',pch=19,cex=0.5)
for(iter in 1:nits){
        #Dimension changing on the marginal likelihood
        if(runif(1)<0.5){
                #Birth
                new.K<-old.K+1
                k<-sample(1:(old.K+1),size=1)
                tvec<-c(0,old.eta,100)
                left.end<-tvec[k]
                right.end<-tvec[k+1]
                new.val<-runif(1,left.end,right.end)
                new.eta<-sort(c(old.eta,new.val))
                new.Xmat<-rep(1,n)
                for(k in 1:new.K){
                        new.Xmat<-cbind(new.Xmat,(x-new.eta[k])^r*(x>new.eta[k]))
                }
                new.np<-ncol(new.Xmat)
                new.prior.L<-prior.lam*diag(1,new.np)
                new.prior.be0<-rep(0,new.np)
                XTX<-t(new.Xmat)%*%new.Xmat
                post.L<-XTX+new.prior.L
                post.var<-solve(post.L)
                post.mean<-post.var %*% (t(new.Xmat) %*% y + new.prior.L %*% new.prior.be0)
                new.c<-(ysq + t(new.prior.be0) %*% new.prior.L %*% new.prior.be0 -
                                t(post.mean) %*% post.L %*% post.mean)[1,1]

                new.marg.like<-0.5*log(det(new.prior.L))-0.5*log(det(XTX+new.prior.L))-
                                        0.5*(n+prior.a)*log(0.5*(prior.b+new.c))

                new.prior.K<-dpois(new.K,prior.gam,log=T)
                new.prior.eta<-eta.density(new.eta,new.K,xmax)

                new.q<--log(right.end-left.end)

                if(log(runif(1)) < (new.marg.like+new.prior.K+new.prior.eta) -
                                (old.marg.like+old.prior.K+old.prior.eta)-new.q){
                        old.K<-new.K
                        old.prior.L<-new.prior.L
            old.prior.be0<-new.prior.be0
            old.eta<-new.eta
                        old.Xmat<-new.Xmat
                        old.np<-new.np
                        old.c<-new.c
                        old.marg.like<-new.marg.like
                        old.prior.K<-new.prior.K
                        old.prior.eta<-new.prior.eta
                }

        }else{
                #Death
                if(old.K == 0) break #Prior prob on fewer than 0 breaks is zero
                new.K<-old.K-1
                k<-sample(1:old.K,size=1)
                new.eta<-old.eta[-k]
                if(new.K == 0){
                        new.Xmat<-matrix(1,ncol=1,nrow=n)
                }else{
                        new.Xmat<-rep(1,n)
                        for(k in 1:new.K){
                                new.Xmat<-cbind(new.Xmat,(x-new.eta[k])^r*(x>new.eta[k]))
                        }
                }
                new.np<-ncol(new.Xmat)
                new.prior.L<-prior.lam*diag(1,new.np)
                new.prior.be0<-rep(0,new.np)
                XTX<-t(new.Xmat)%*%new.Xmat
                post.L<-XTX+new.prior.L
                post.var<-solve(post.L)
                post.mean<-post.var %*% (t(new.Xmat) %*% y + new.prior.L %*% new.prior.be0)
                new.c<-(ysq + t(new.prior.be0) %*% new.prior.L %*% new.prior.be0 -
                                t(post.mean) %*% post.L %*% post.mean)[1,1]

                new.marg.like<-0.5*log(det(new.prior.L))-0.5*log(det(XTX+new.prior.L))-
                                0.5*(n+prior.a)*log(0.5*(prior.b+new.c))

                new.prior.K<-dpois(new.K,prior.gam,log=T)
                new.prior.eta<-eta.density(new.eta,new.K,xmax)

                evec<-c(0,old.eta,xmax)
                new.q<--log(evec[k+2]-evec[k])

                if(log(runif(1)) < (new.marg.like+new.prior.K+new.prior.eta) -
                                        (old.marg.like+old.prior.K+old.prior.eta)+new.q){
                        old.K<-new.K
                        old.prior.L<-new.prior.L
                old.prior.be0<-new.prior.be0
                        old.eta<-new.eta
                        old.Xmat<-new.Xmat
                        old.np<-new.np
                        old.c<-new.c
                        old.marg.like<-new.marg.like
                        old.prior.K<-new.prior.K
                        old.prior.eta<-new.prior.eta
                }
        }

        #Shift one of the knots
        if(old.K >0){
                new.eta<-old.eta
                evec<-c(0,old.eta,xmax)
                k<-sample(1:old.K,size=1)
                new.eta[k]<-runif(1,evec[k],evec[k+2])
                new.Xmat<-rep(1,n)
                for(k in 1:old.K){
                        new.Xmat<-cbind(new.Xmat,(x-new.eta[k])^r*(x>new.eta[k]))
                }
                XTX<-t(new.Xmat)%*%new.Xmat
                post.L<-XTX+old.prior.L
                post.var<-solve(post.L)
                post.mean<-post.var %*% (t(new.Xmat) %*% y + old.prior.L %*% old.prior.be0)
                new.c<-(ysq + t(old.prior.be0) %*% old.prior.L %*% old.prior.be0 -
                                t(post.mean) %*% post.L %*% post.mean)[1,1]
                new.marg.like<-0.5*log(det(old.prior.L))-0.5*log(det(XTX+old.prior.L))-
                                0.5*(n+prior.a)*log(0.5*(prior.b+new.c))
                new.prior.eta<-eta.density(new.eta,old.K,xmax)
                if(log(runif(1)) < (new.marg.like+new.prior.eta) -
                                                (old.marg.like+old.prior.eta)){
                        old.eta<-new.eta
                        old.Xmat<-new.Xmat
                        old.c<-new.c
                        old.marg.like<-new.marg.like
                        old.prior.eta<-new.prior.eta
                }
        }

        if(iter > nburn & iter %% nthin == 0){
                ico<-ico+1
                #Sample the parameters given old.eta
                XTX<-t(old.Xmat)%*%old.Xmat
                post.L<-XTX+old.prior.L
                post.var<-solve(post.L)
                post.mean<-post.var %*% (t(old.Xmat) %*% y + old.prior.L %*% old.prior.be0)
                old.be<-rmvn(1,mu=post.mean,sigma=old.sig^2*post.var)

                old.yfit<-old.Xmat %*% t(old.be)
                old.ssq<-sum((y-old.yfit)^2)
                post.a<-prior.a+n
                post.b<-prior.b+old.ssq

                old.sig<-1/sqrt(rgamma(1,post.a/2,post.b/2))

                old.c<-(ysq + t(old.prior.be0) %*% old.prior.L %*% old.prior.be0 -
                                t(post.mean) %*% post.L %*% post.mean)[1,1]

                old.marg.like<-0.5*log(det(old.prior.L))-0.5*log(det(XTX+old.prior.L))-
                                        0.5*(n+prior.a)*log(0.5*(prior.b+old.c))

                be.post[ico,1:(old.K+1)]<-old.be
                sig.post[ico]<-old.sig
                K.post[ico]<-old.K
                eta.post[ico,1:old.K]<-old.eta
        }

        if(iter %% 1000 ==0){
                Xm<-rep(1,nx)
                for(k in 1:old.K){
                        Xm<-cbind(Xm,(xvec-old.eta[k])^r*(xvec>old.eta[k]))
                }
                XTX<-t(old.Xmat)%*%old.Xmat
                post.L<-XTX+old.prior.L
                post.var<-solve(post.L)
                post.mean<-post.var %*% (t(old.Xmat) %*% y + old.prior.L %*% old.prior.be0)
                old.be<-rmvn(1,mu=post.mean,sigma=old.sig^2*post.var)
                y.fit.b<-Xm %*% t(old.be)
                lines(xvec,y.fit.b,col='red')
        }
}
```



In this analysis, under the chosen prior, the posterior on the number of knots has support only on 1,2, and 3 knots.

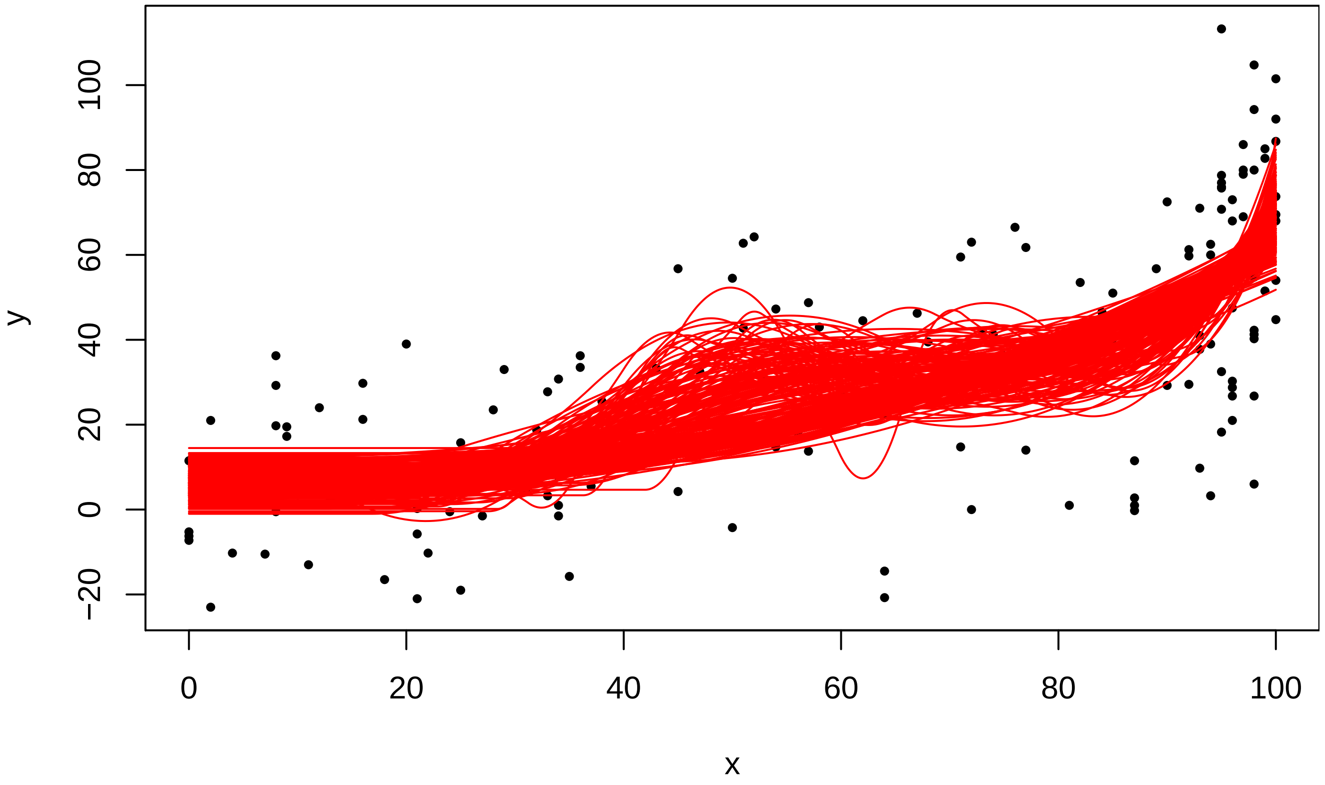
```
table(K.post)/nsamp

+ K.post
+      1      2      3
+ 0.9854 0.0141 0.0005
```

If we change the hyperparameter λ to be $\lambda = 1$ the results change: this is a more restrictive prior, and a greater number of knots is needed.



```
table(K.post)/nsamp

+ K.post
+      1      2      3      4      5
+ 0.8174 0.0935 0.0821 0.0067 0.0003
```

If we set $\lambda = 10$ the results change further.



```
table(K.post)/nsamp

+ K.post
+      1      2      3      4      5      6      7      8     11 
+ 0.3123 0.0956 0.4577 0.1030 0.0233 0.0063 0.0014 0.0002 0.0002 
```