MATH 598: TOPICS IN STATISTICS

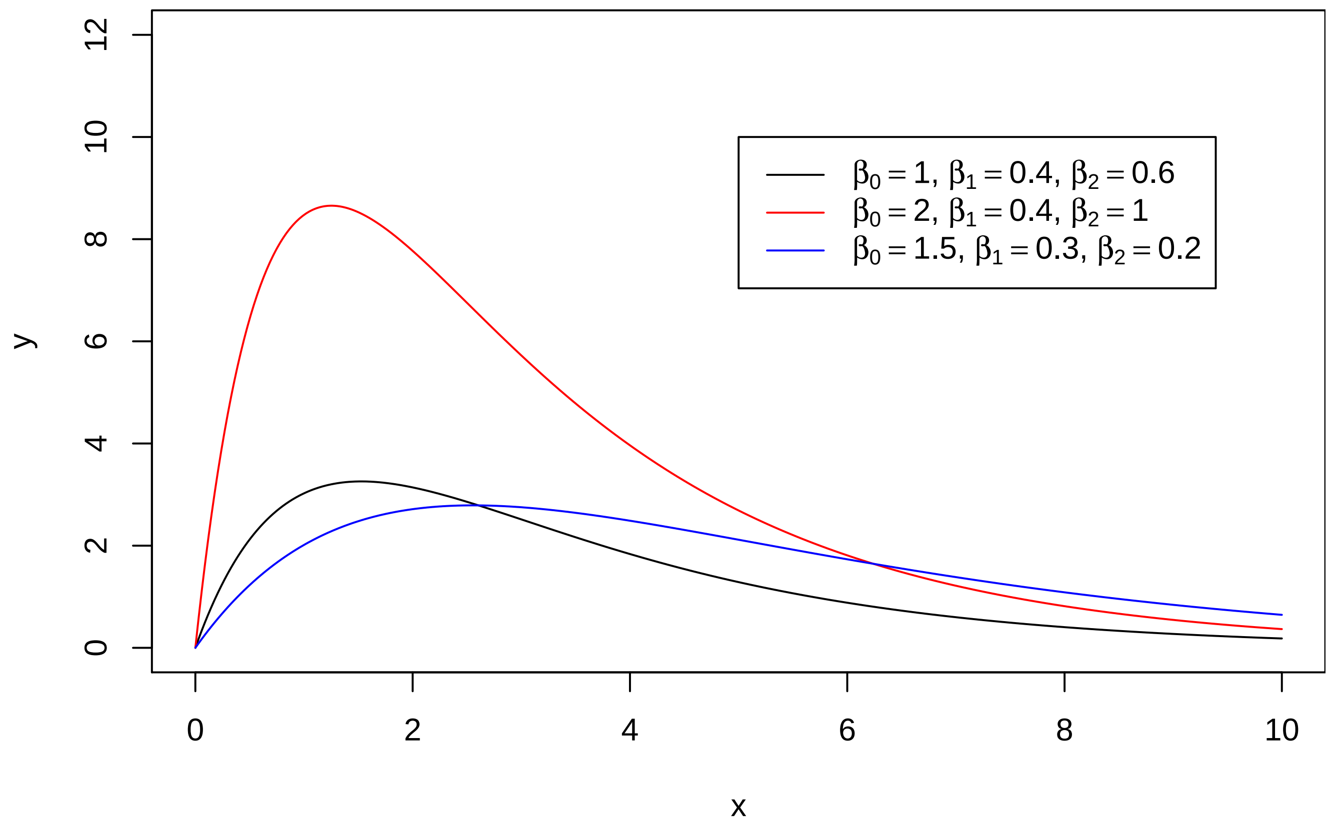
Bayesian Inference for Nonlinear Regression

A non-linear regression model is a model for the conditional expectation of a response variable Y in terms of a function of predictors x that is non-linear in the parameters. For example, in pharmacokinetics, the concentration Y of drug in the bloodstream at time x after an initial dose D can be modelled using ordinary differential equations. One simple 'compartment' model represents the concentration $y(x)$ at time x as

$$y(x) = D\beta_0(\exp\{-\beta_1 x\} - \exp\{-(\beta_1+\beta_2)x\}) = \mu(x, D, \beta_0, \beta_1, \beta_2) \qquad x > 0$$

say, for parameters $\beta_0, \beta_1, \beta_2 > 0$.

```
D<-10
pk.model<-function(xv,b0,b1,b2,Dose){
    yv<-Dose*be0*(exp(-b1*xv)-exp(-(b1+b2)*xv))
    return(yv)
}
x<-seq(0,10,by=0.01)
be0<-1;be1<-0.4;be2<-0.6
y<-pk.model(x,be0,be1,be2,D)
par(mar=c(4,4,1,0))
plot(x,y,type='l',ylim=range(0,12))
be0<-2;be1<-0.4;be2<-1
y<-pk.model(x,be0,be1,be2,D)
lines(x,y,col='red')
be0<-1.5;be1<-0.3;be2<-0.2
y<-pk.model(x,be0,be1,be2,D)
lines(x,y,col='blue')
legend(5,10,c(expression(paste(beta[0]==1.0,', ',beta[1]==0.4,', ',beta[2]==0.6)),
              expression(paste(beta[0]==2.0,', ',beta[1]==0.4,', ',beta[2]==1.0)),
              expression(paste(beta[0]==1.5,', ',beta[1]==0.3,', ',beta[2]==0.2))),
              lty=1,col=c('black','red','blue'))
```

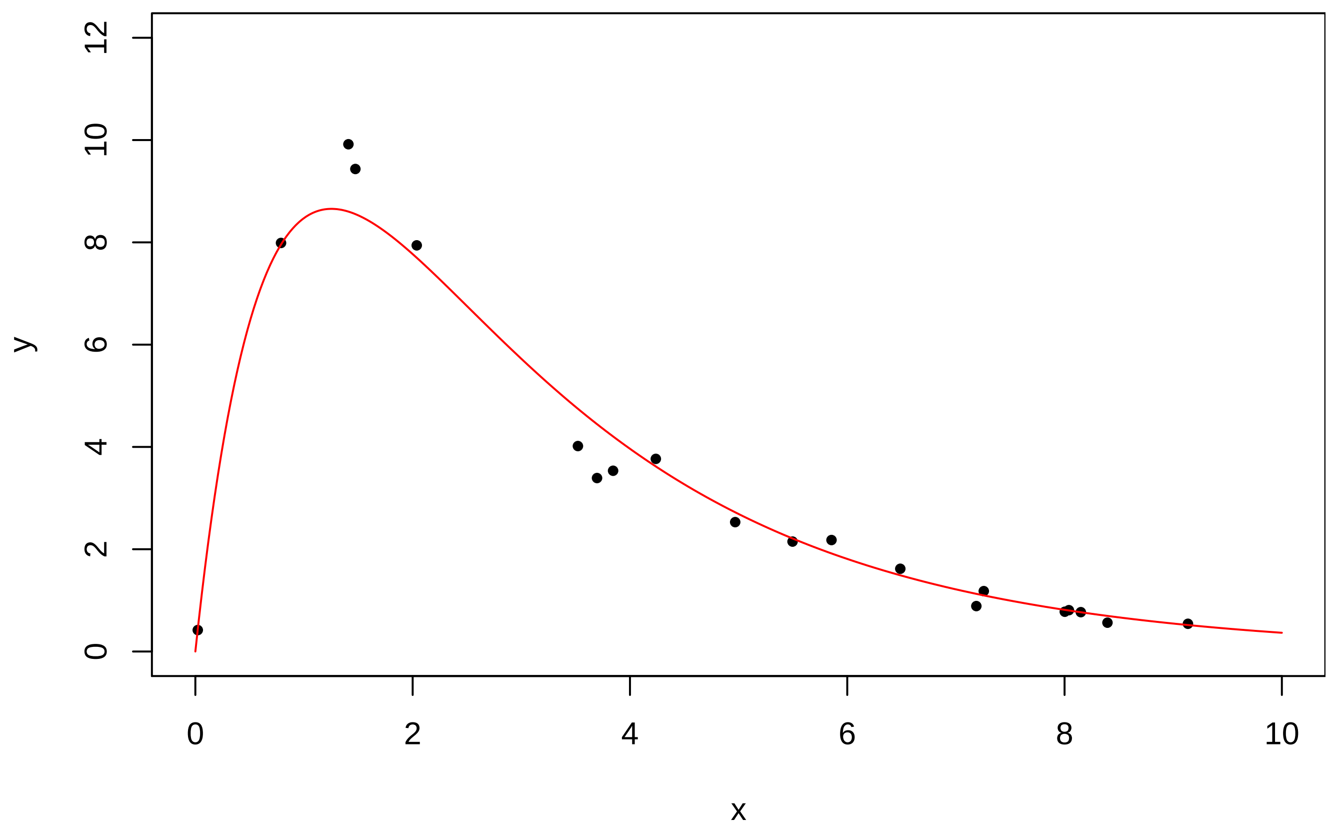


In practice, data are collected from individual patients across time, and these data are measured with observation error. One common model involves representing the data using multiplicative errors, that is

$$Y(x_i) = \mu(x_i, D, \beta_0, \beta_1, \beta_2)\epsilon_i$$

for $i = 1, \ldots, n$, where $\log \epsilon_i \sim Normal(0, \sigma^2)$. Below, $n = 20$ data points gathered uniformly on $(0, 10)$ are simulated according to this model, with $(\beta_0, \beta_1, \beta_2) = (2, 0.4, 1)$ and $\sigma = 0.1$, for $D = 10$.

```
set.seed(37)
D<-10
n<-20
sig<-0.1
x0<-seq(0,10,by=0.01)
x<-sort(runif(n,0,10))
epsilon<-exp(rnorm(n,0,sig))
be0<-2;be1<-0.4;be2<-1
y0<-pk.model(x0,be0,be1,be2,D)
y<-pk.model(x,be0,be1,be2,D)*epsilon
par(mar=c(4,4,1,0))
plot(x,y,xlim=range(0,10),ylim=range(0,12),pch=19,cex=0.6)
lines(x0,y0,col='red')
```



This suggests that the statistical model to be used for inference is

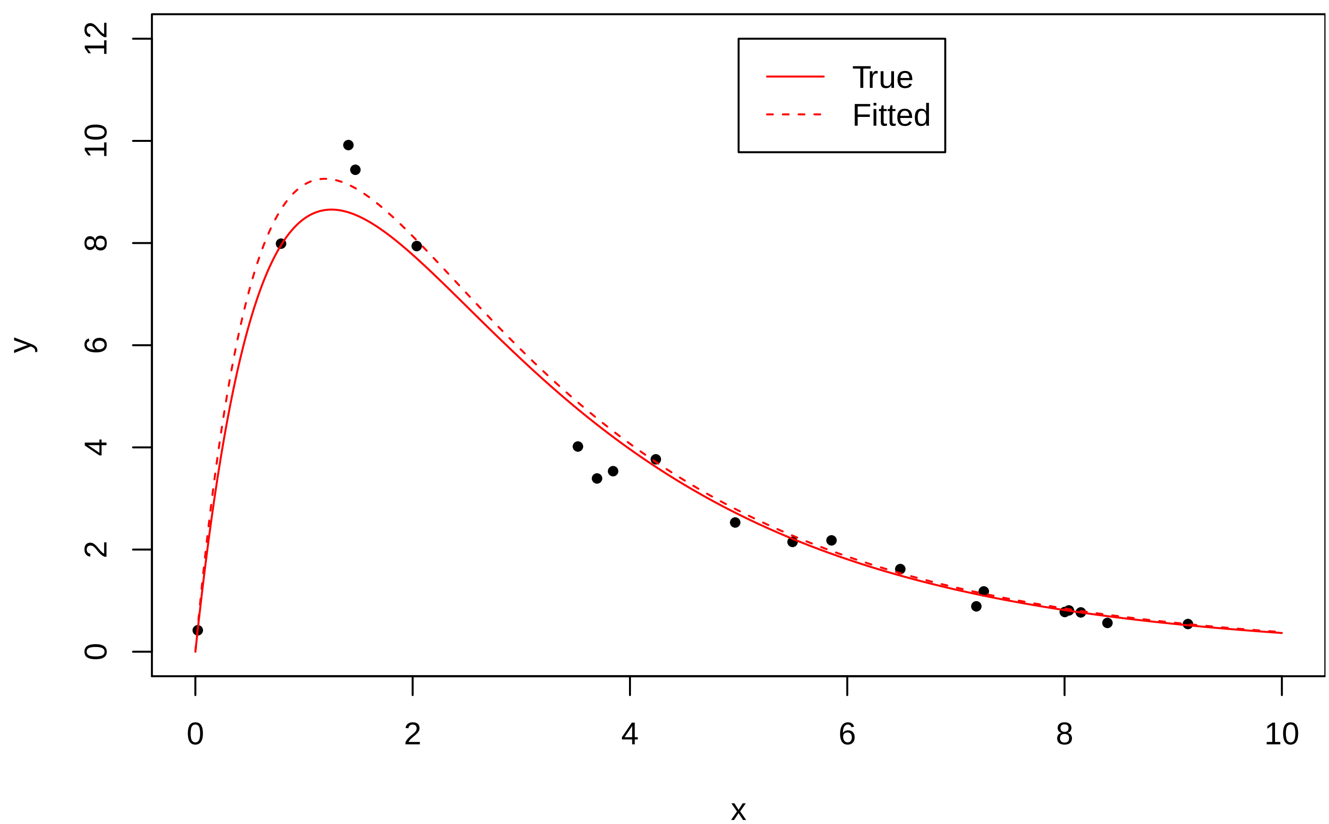
$$\log Y(x_i) = \log \mu(x, D, \beta_0, \beta_1, \beta_2) + \log \epsilon_i$$

for $i = 1, \ldots, n$. Frequentist estimation can be carried out using non-linear least squares. We can achieve the minimization of

$$\sum_{i=1}^{n} (\log y_i - \log \mu(x_i, D, \beta_0, \beta_1, \beta_2))^2$$

using `optim`.

```
pk.like<-function(be,Dv,xv,yv){
    res<-log(yv) - log(Dv*be[1]*(exp(-be[2]*xv)-exp(-(be[2]+be[3])*xv)))
    return(sum(res^2))
}
be.start<-c(1,0.1,0.2)
pk.fit<-optim(be.start,fn=pk.like,Dv=D,xv=x,yv=y)
pk.fit  #par gives parameter estimates

+ $par
+ [1] 1.8600512 0.3953164 1.1350574
+
+ $value
+ [1] 0.2763411
+
+ $counts
+ function gradient
+      152       NA
+
+ $convergence
+ [1] 0
+
+ $message
+ NULL

par(mar=c(4,4,1,0))
plot(x,y,xlim=range(0,10),ylim=range(0,12),pch=19,cex=0.6)
lines(x0,y0,col='red')
be0.hat<-pk.fit$par[1];be1.hat<-pk.fit$par[2];be2.hat<-pk.fit$par[3]
y.hat<-pk.model(x0,be0.hat,be1.hat,be2.hat,D)
lines(x0,y.hat,col='red',lty=2)
legend(5,12,c('True','Fitted'),col='red',lty=c(1,2))
```

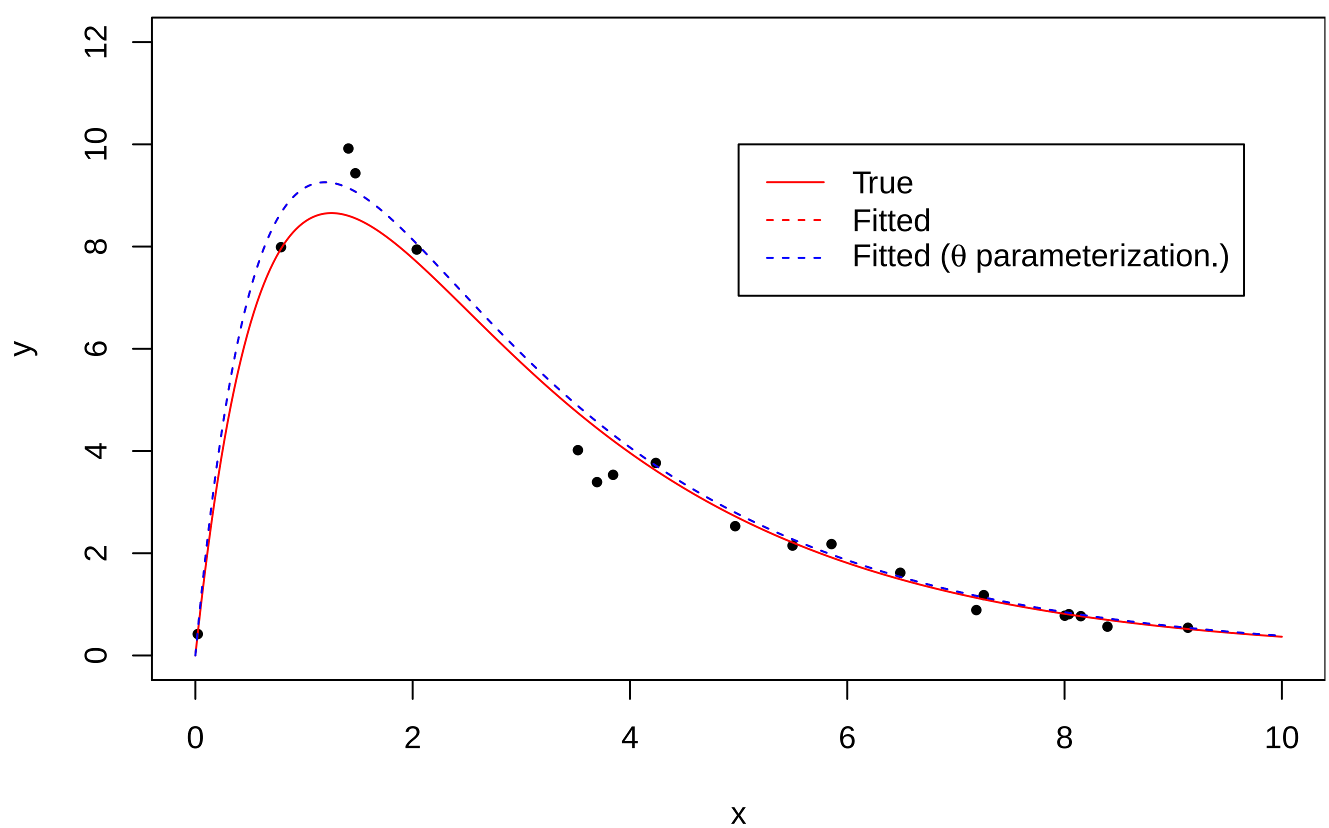


We can carry out the same optimization, but first reparamaterize onto the log scale, say

$$\theta_0 = \log\beta_0 \qquad \theta_1 = \log\beta_1 \qquad \theta_2 = \log\beta_2$$

as this should make no difference in the minimization.

```
pk.like.th<-function(th,Dv,xv,yv){
    be<-exp(th)
    res<-log(yv) - log(Dv*be[1]*(exp(-be[2]*xv)-exp(-(be[2]+be[3])*xv)))
    return(sum(res^2))
}
th.start<-log(be.start)
pk.fit.th<-optim(th.start,fn=pk.like.th,Dv=D,xv=x,yv=y)
pk.fit.th$par       #par gives parameter estimates on theta scale

+ [1]  0.6203688 -0.9282107  0.1267071

exp(pk.fit.th$par) #on the beta scale - identical to previous analysis

+ [1] 1.8596137 0.3952603 1.1350846

par(mar=c(4,4,1,0))
plot(x,y,xlim=range(0,10),ylim=range(0,12),pch=19,cex=0.6)
lines(x0,y0,col='red')
be0.hat2<-exp(pk.fit.th$par[1]);be1.hat2<-exp(pk.fit.th$par[2]);be2.hat2<-exp(pk.fit.th$par[3])
y.hat2<-pk.model(x0,be0.hat2,be1.hat2,be2.hat2,D)
lines(x0,y.hat,col='red',lty=2)
lines(x0,y.hat2,col='blue',lty=2)
legend(5,10,c('True','Fitted',expression(paste('Fitted (',theta,' parameterization.)'))),
            col=c('red','red','blue'),lty=c(1,2,2))
```



For a Bayesian analysis, we use the same model to generate the likelihood: in the θ parameterization, we have

$$\begin{aligned}
\mathcal{L}_n(\theta,\sigma^2) &= \prod_{i=1}^n \left(\frac{1}{2\pi\sigma^2}\right)^{1/2} \exp\left\{-\frac{1}{2\sigma^2}(\log y_i - \log\mu(x_i, D, e^{\theta_0}, e^{\theta_1}, e^{\theta_2}))^2\right\} \\
&= \left(\frac{1}{2\pi}\right)^{n/2}\left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^n(\log y_i - \log\mu(x_i, D, e^{\theta_0}, e^{\theta_1}, e^{\theta_2}))^2\right\} \\
&= \left(\frac{1}{2\pi}\right)^{n/2}\left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)\right\}
\end{aligned}$$

say. For the prior, we may choose independent priors on $\theta_0, \theta_1, \theta_2$ conditional on σ^2. For σ^2, as in the linear regression case, we may choose an Inverse Gamma prior

$$\pi_0(\sigma^2) \equiv InvGamma(a/2, b/2)$$

for suitable constants a and b; we may use the subjective prior knowledge that suggests that σ^2 is no greater than 5, and suggest $a = b = 8$. Given σ^2, for $(\theta_0, \theta_1, \theta_2)$ we may select independent $Normal(0, \sigma^2/\lambda)$ priors; more generally, a multivariate Normal prior

$$\pi_0(\theta_0,\theta_1,\theta_2|\sigma^2) \equiv Normal_3(\mathbf{0}, \sigma^2\mathbf{L}^{-1})$$

for some positive definite matrix $\mathbf{L}^{-1}$ could be considered. For the posterior distribution, then, we have up to proportionality that

$$\begin{aligned}
\pi_n(\theta_0,\theta_1,\theta_2,\sigma^2) \propto\; & \mathcal{L}_n(\theta_0,\theta_1,\theta_2,\sigma^2)\pi_0(\theta_0,\theta_1,\theta_2|\sigma^2)\pi_0(\sigma^2) \\
\propto\; & \left(\frac{1}{\sigma^2}\right)^{n/2} \exp\left\{-\frac{1}{2\sigma^2}S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)\right\}\left(\frac{1}{\sigma^2}\right)^{3/2}\exp\left\{-\frac{1}{2\sigma^2}\theta^\top\mathbf{L}\theta\right\}\left(\frac{1}{\sigma^2}\right)^{a/2+1}\exp\left\{-\frac{b}{2\sigma^2}\right\} \\
\propto\; & \left(\frac{1}{\sigma^2}\right)^{(n+a+3)/2+1} \exp\left\{-\frac{1}{2\sigma^2}\left[S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)+\theta^\top\mathbf{L}\theta+b\right]\right\}.
\end{aligned}$$

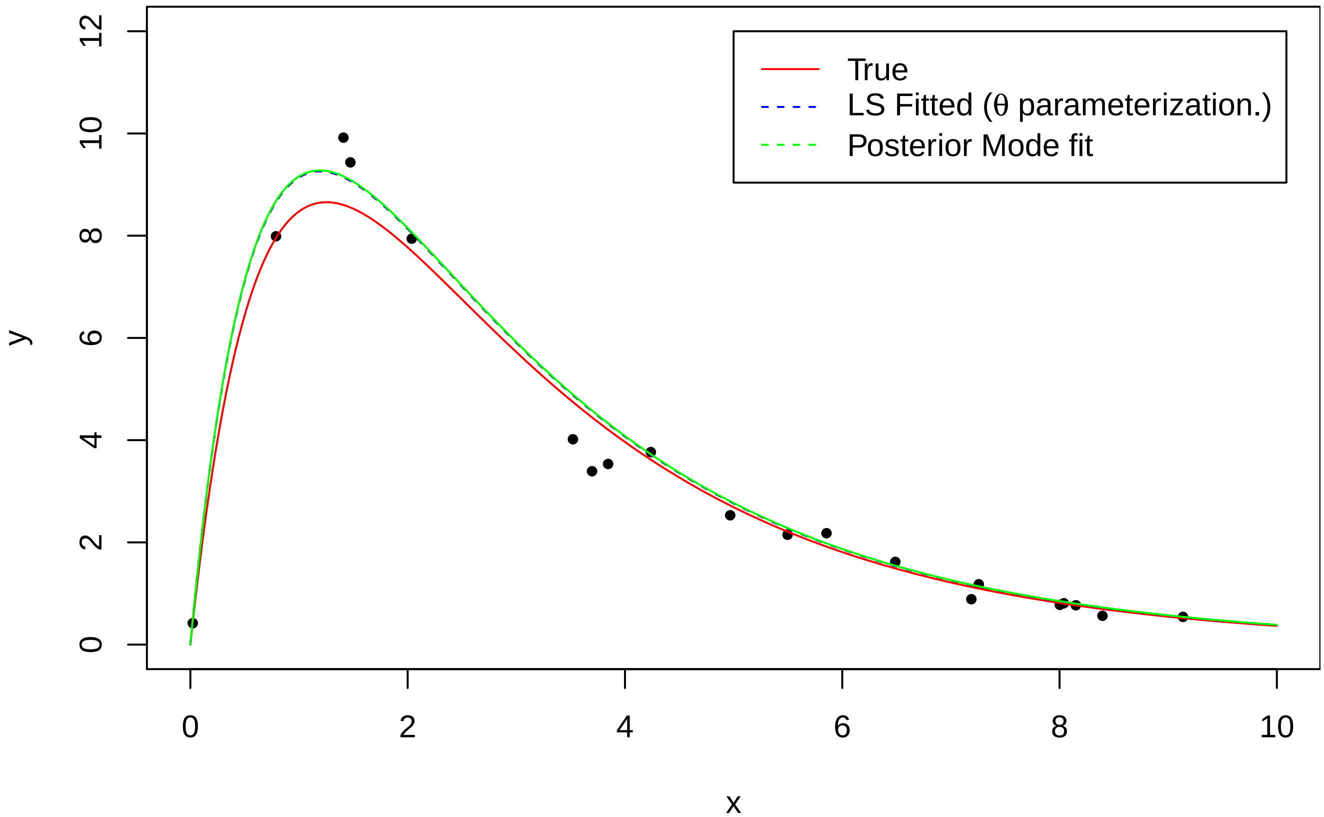
To compute with this posterior distribution, we may first integrate out σ^2 to leave the marginal posterior for $(\theta_0, \theta_1, \theta_2)$. To integrate out σ^2 from $\pi_n(\theta_0, \theta_1, \theta_2, \sigma^2)$, we note that the integrand is proportional to an Inverse Gamma pdf, so therefore

$$\pi_n(\theta_0,\theta_1,\theta_2) \propto \left\{S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)+\theta^\top\mathbf{L}\theta+b\right\}^{-(n+a+3)/2}.$$

This posterior is intractable. We may, however, perform maximization of the posterior again using `optim`.

```
pk.loglike.th<-function(th,Dv,xv,yv,Lm,av,bv){
    be<-exp(th)
    res<-log(yv) - log(Dv*be[1]*(exp(-be[2]*xv)-exp(-(be[2]+be[3])*xv)))
    Sval<-sum(res^2)
    pth<-t(th) %*% (Lm %*% th)
    llike<--(length(xv)+av+3)*log(Sval+pth[1,1]+b)
    return(-llike)   #optim performs minimization - need to multiply log posterior by -1
}
a<-b<-8
L<-diag(rep(1/10,3))
th.start<-pk.fit.th$par
pk.post.th<-optim(th.start,fn=pk.loglike.th,Dv=D,xv=x,yv=y,Lm=L,av=a,bv=b,hessian=T)
pk.post.th$par      #par gives parameter estimates on theta scale

+ [1]  0.6138207 -0.9297518  0.1293539

exp(pk.post.th$par) #on the beta scale - identical to previous analysis
```

```
+ [1] 1.8474766 0.3946516 1.1380928

par(mar=c(4,4,1,0))
plot(x,y,xlim=range(0,10),ylim=range(0,12),pch=19,cex=0.6)
lines(x0,y0,col='red')
be0.post<-exp(pk.post.th$par[1]);be1.post<-exp(pk.post.th$par[2]);be2.post<-exp(pk.post.th$par[3])
y.post<-pk.model(x0,be0.post,be1.post,be2.post,D)
lines(x0,y.hat,col='red',lty=2)
lines(x0,y.hat2,col='blue',lty=2)
lines(x0,y.post,col='green',lty=1)
legend(5,12,c('True',expression(paste('LS Fitted (',theta,' parameterization.)')),
              'Posterior Mode fit'),col=c('red','blue','green'),lty=c(1,2,2))
```



As for the earlier GLM examples, we may attempt to approximate the posterior using a Normal approximation

$$\pi_n(\theta) \approx Normal_3(\widehat{\theta}_n, \widehat{M}_n^{-1})$$

where M_n is computed as the Hessian matrix (matrix of second partial derivatives of the negative log-likelihood evaluated at $\widehat{\theta}_n$).

```
Mn<-pk.post.th$hessian
(pk.post.var<-solve(Mn))

+             [,1]        [,2]        [,3]
+ [1,]  0.07977347  0.02960331 -0.10211777
+ [2,]  0.02960331  0.01232891 -0.03696588
+ [3,] -0.10211777 -0.03696588  0.20235534
```

For an exact analysis using Monte Carlo methods, we aim to use rejection sampling to produce a sample from the exact posterior. To do this we need an easy to sample distribution, $\widetilde{\pi}_n(\theta)$ say, that resembles the true posterior: from the above analysis, that $\widetilde{\pi}_n(\theta) \equiv Normal_3(\widehat{\theta}_n, \widehat{M}_n^{-1})$ distribution seems a sensible place to start. However, recall that rejection sampling requires us first to bound the ratio of the target to the proposal density

$$\frac{\pi_n(\theta)}{\widetilde{\pi}_n(\theta)} < M.$$

In this case, due to the prior assumptions, the posterior itself is approximately Normal in its tails. Therefore, choosing a Normal proposal density may not be feasible as the ratio $\pi_n(\theta)/\widetilde{\pi}_n(\theta)$ may not be bounded in the tails. Instead, we choose a multivariate $Student(5)$ distribution with the same location and scale – this forces the tail behaviour of the proposal density to be `heavier` than the target. The multivariate Student-`t` distribution density and random number generation may be implemented via the `mvnfast` library, and using the `dmvt` and `rmvt` functions.

```
library(mvnfast)
pk.rejection.th<-function(th,Dv,xv,yv,Lm,av,bv,muv,Sigv){
    be<-exp(th)
    res<-log(yv) - log(Dv*be[1]*(exp(-be[2]*xv)-exp(-(be[2]+be[3])*xv)))
    Sval<-sum(res^2)
    pth<-t(th) %*% (Lm %*% th)
    llike<--(length(xv)+av+3)*log(Sval+pth[1,1]+bv)
    lprop<-dmvt(th,muv,Sigv,5,log=T)
    return(-llike+lprop)   #optim performs minimization - need to multiply log posterior by -1
}
a<-b<-8
L<-diag(rep(1/10,3))
th.start<-pk.post.th$par
post.mu<-pk.post.th$par
post.Sig<-pk.post.var
pk.rej.th<-optim(th.start,fn=pk.rejection.th,
                 Dv=D,xv=x,yv=y,Lm=L,av=a,bv=b,muv=post.mu,Sigv=post.Sig)
pk.rej.th

+ $par
+ [1]  0.1074398 -1.2440940  0.6723525
+
+ $value
+ [1] 67.83072
+
+ $counts
+ function gradient
+      175       NA
+
+ $convergence
+ [1] 0
+
+ $message
+ NULL
```

This analysis gives that

$$\log\left(\frac{\pi_n(\theta)}{c\widetilde{\pi}_n(\theta)}\right) < -67.830716$$

– recall that we only have $\pi_n(\theta)$ up to proportionality, so there is an indeterminate constant c; however, we do not need the constant to implement rejection sampling. We aim to produce a sample of size $N = 10000$ from the posterior.

```
set.seed(94)
N<-10000
nsamp<-0
ico<-0
Mval<--pk.rej.th$val
theta.rej<-matrix(0,nrow=N,ncol=3)
while(nsamp < N){
```

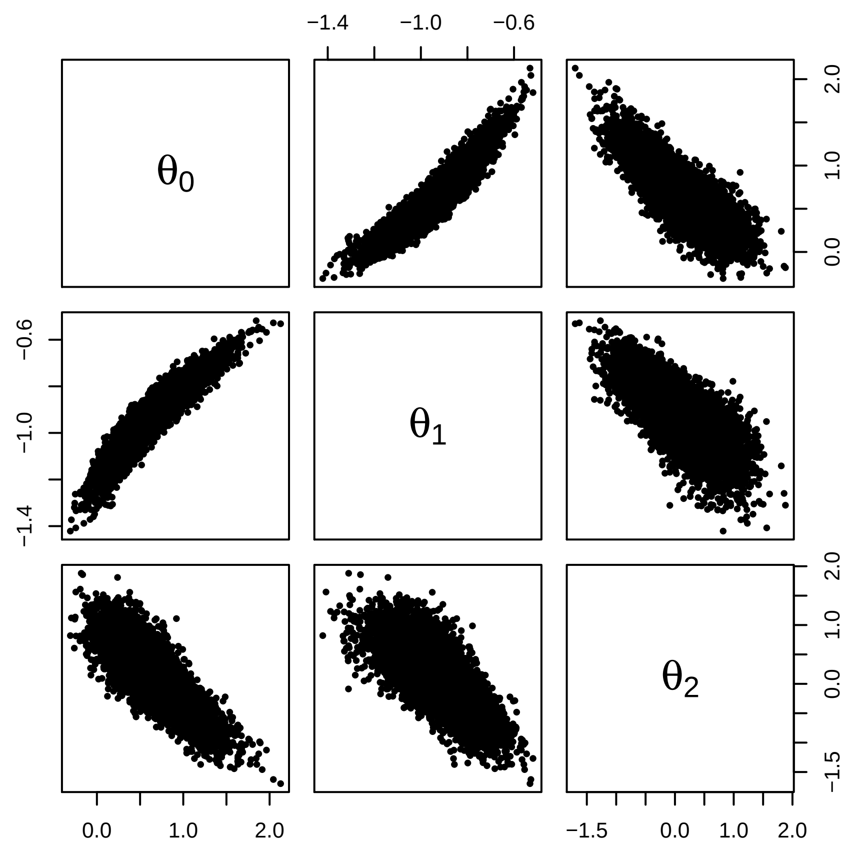
```
    ico<-ico+1
    th<-(rmvt(1,post.mu,post.Sig,5))
    be<-exp(th)
    res<-log(y) - log(D*be[1]*(exp(-be[2]*x)-exp(-(be[2]+be[3])*x)))
    Sval<-sum(res^2)
    pth<-(th) %*% (L %*% t(th))
    llike<--(length(x)+a+3)*log(Sval+pth[1,1]+b)
    lprop<-dmvt(th,post.mu,post.Sig,5,log=T)
    U<-runif(1)
    if(log(U) < llike-lprop-Mval){
        nsamp<-nsamp+1
        theta.rej[nsamp,]<-th
    }
}
print(ico)

+ [1] 34182
```

This analysis tells us that on this run, 34182 proposals were needed to generate the $N = 10000$ samples from the posterior. Thus the acceptance rate is 0.292552, and therefore

$$\max_{\theta} \frac{\pi_n(\theta)}{\widetilde{\pi}_n(\theta)} \simeq \frac{1}{0.292552} = 3.418200$$

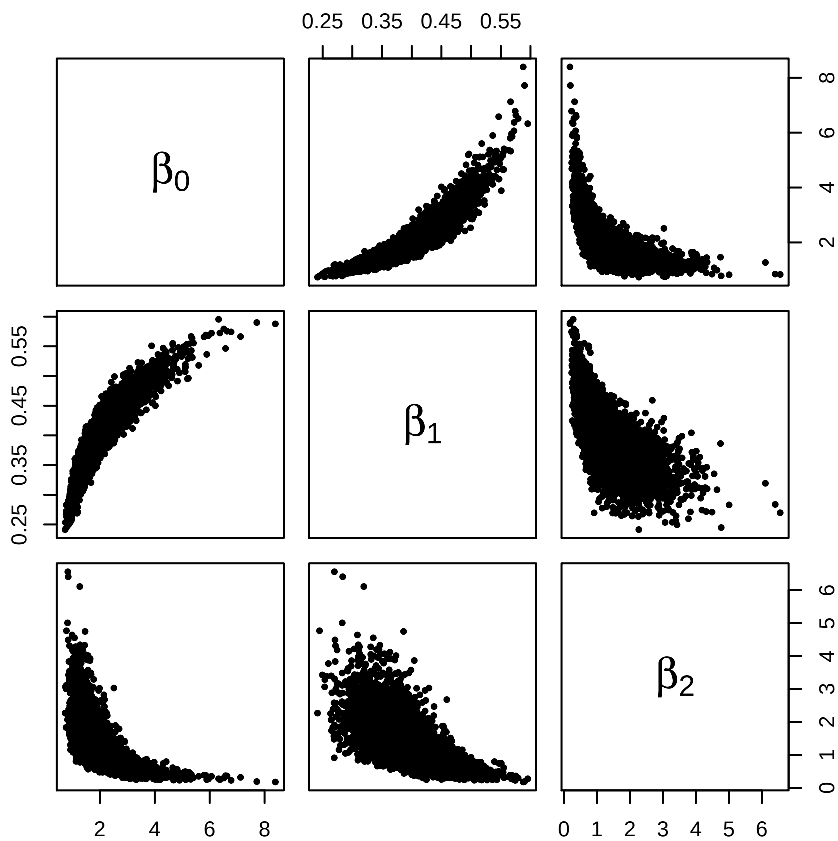
which informs us that the normalizing constant c is given approximately by $c \simeq 3.418200 \exp\{-67.830716\}$.

```
par(mar=c(4,4,2,0));pairs(theta.rej,pch=19,cex=0.5,
    labels=c(expression(theta[0]),expression(theta[1]),expression(theta[2])))
```



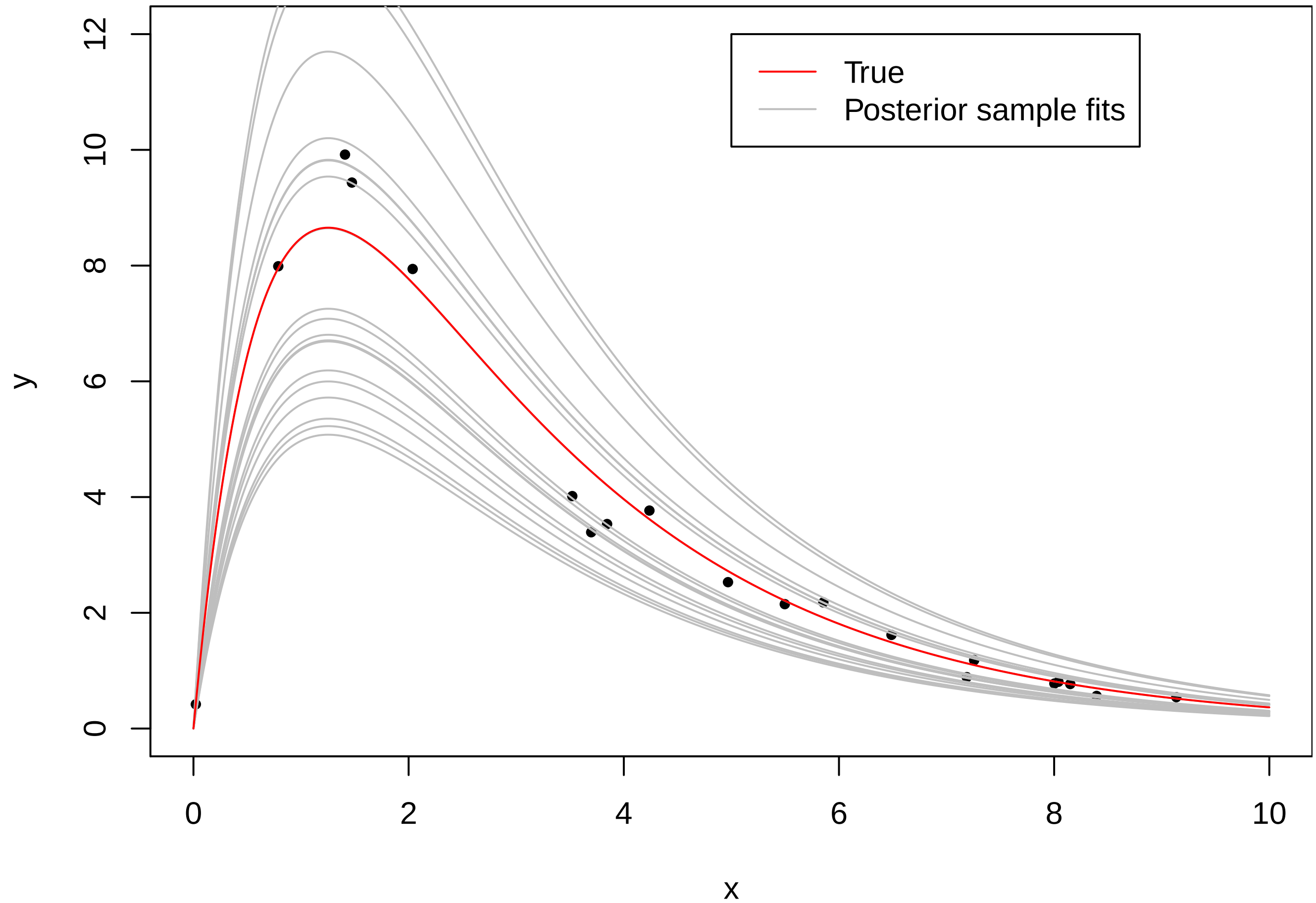
We can obtain samples of the posterior for $\beta_0, \beta_1, \beta_2$ by transformation.

```
beta.rej<-exp(theta.rej)
par(mar=c(4,4,2,0));pairs(beta.rej,pch=19,cex=0.5,labels=
                          c(expression(beta[0]),expression(beta[1]),expression(beta[2])))
```



Having obtained the posterior samples for $\beta_0, \beta_1, \beta_2$, we can compute sampled fitted curves.

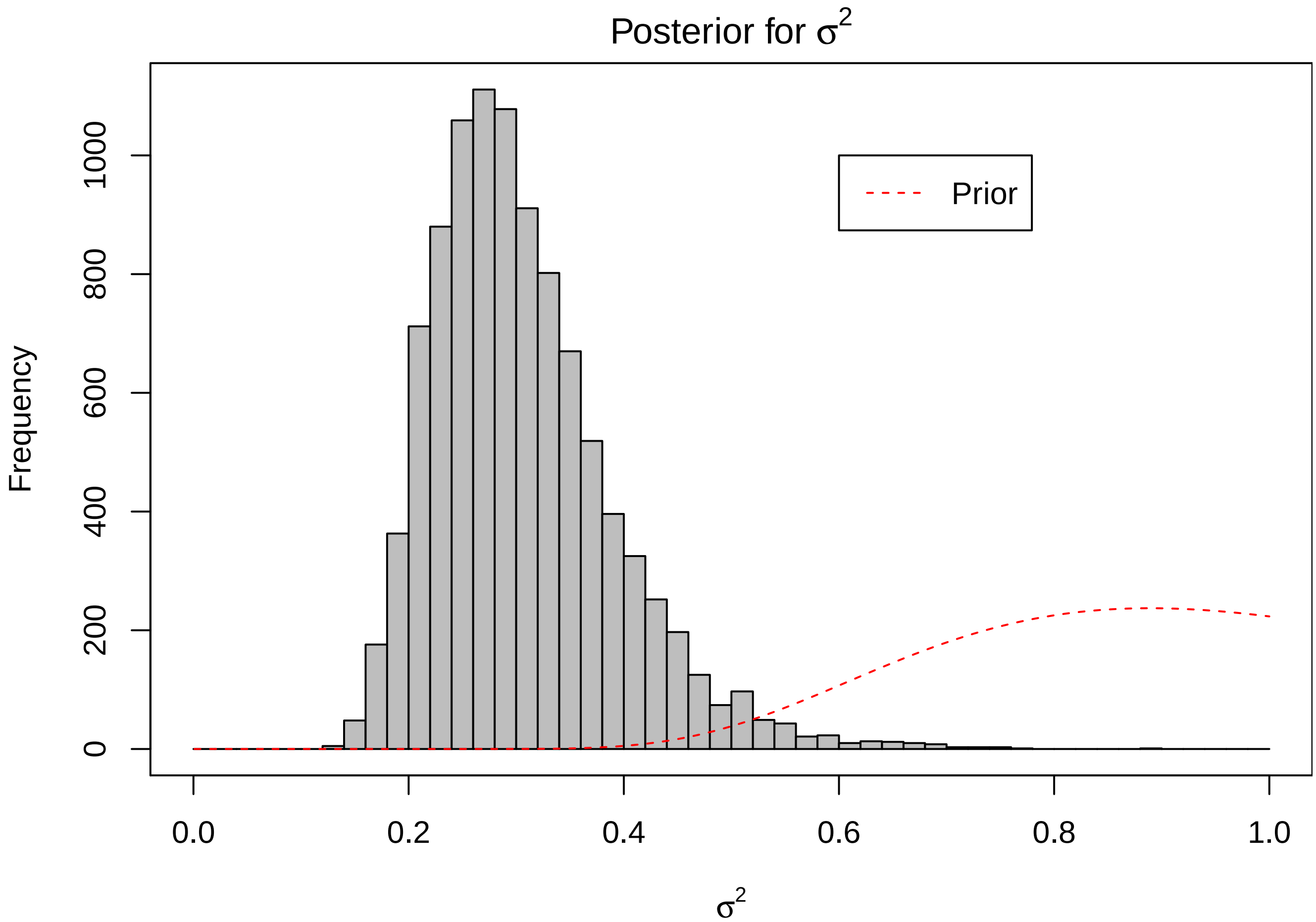
```
par(mar=c(4,4,1,0))
plot(x,y,xlim=range(0,10),ylim=range(0,12),pch=19,cex=0.6)
for(i in 1:20){
    be0<-beta.rej[i,1];be1.post<-beta.rej[i,2];be2.post<-beta.rej[i,3]
    y.post<-pk.model(x0,be0,be1,be2,D)
    lines(x0,y.post,col='gray',lty=1)
}
legend(5,12,c('True','Posterior sample fits'),col=c('red','gray'),lty=c(1,1))
lines(x0,y0,col='red')
```



Finally, having sampled the θ values, we can produce a sample from the posterior for σ^2 by sampling the conditional posterior

$$
\begin{aligned}
\pi_n(\sigma^2|\theta_0,\theta_1,\theta_2) &\propto \pi_n(\theta_0,\theta_1,\theta_2,\sigma^2) \\
&\propto \left(\frac{1}{\sigma^2}\right)^{(n+a+3)/2+1} \exp\left\{-\frac{1}{2\sigma^2}\left[S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)+\theta^\top \mathbf{L}\theta + b\right]\right\} \\
&\equiv InvGamma\left(\frac{(n+a+3)}{2}, \frac{\left(S(\mathbf{y},\mathbf{x},D,\theta_0,\theta_1,\theta_2)+\theta^\top \mathbf{L}\theta + b\right)}{2}\right)
\end{aligned}
$$

```
be.rej<-exp(theta.rej)
sigsq.rej<-rep(0,N)
for(i in 1:N){
    res<-log(y) - log(D*be.rej[i,1]*(exp(-be.rej[i,2]*x)-exp(-(be.rej[i,2]+be.rej[i,3])*x)))
    Sval<-sum(res^2)
    pth<-t(theta.rej[i,]) %*% (L %*% (theta.rej[i,]))
    sigsq.rej[i]<-1/rgamma(1,(length(x)+a+3)/2,(Sval+pth[1,1]+b)/2)
}
par(mar=c(4,4,2,0))
hist(sigsq.rej,breaks=seq(0,1,by=0.02),xlab=expression(sigma^2),
     main=expression(paste('Posterior for ',sigma^2)),col='gray');box()
xv<-seq(0,1,by=0.001)
dinvgamma<-function(x,al,be){return(exp(al*log(be)-lgamma(al)-(al+1)*log(x)-be/x))}
yv<-dinvgamma(xv,8,8)
lines(xv,yv*N*0.02,col='red',lty=2)
legend(0.6,1000,c('Prior'),lty=2,col='red')
```

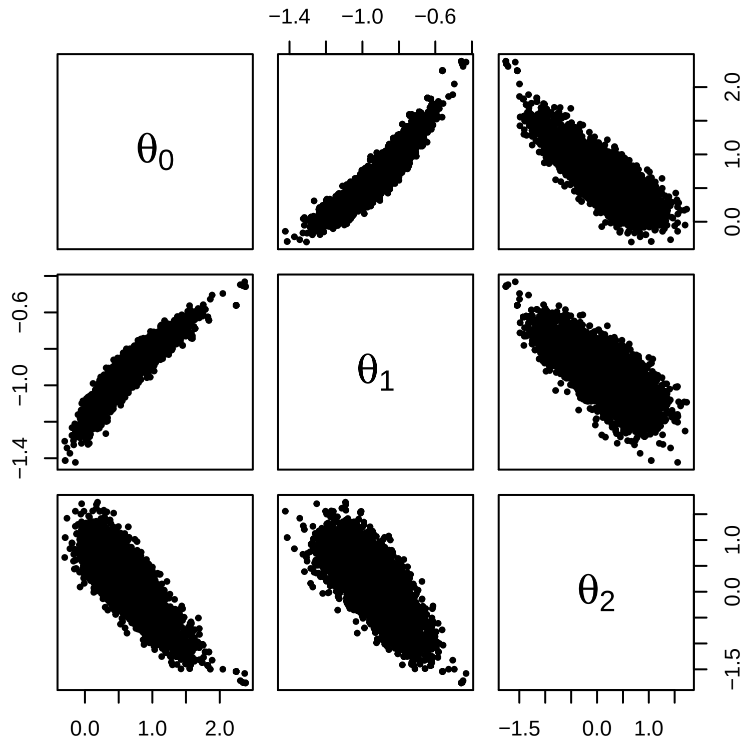


Sampling Importance Resampling: In sampling importance resampling, we generate a large sample $\theta^{(1)}, \ldots, \theta^{(N_0)}$ from a proposal distribution $\widetilde{\pi}_n(\theta)$ and the produce a sample of size N from the target $\pi_n(\theta)$ by resampling from the generated sample with replacement with weights

$$w_i = \frac{\pi_n(\theta^{(i)})/\widetilde{\pi}_n(\theta^{(i)})}{\sum_{j=1}^{N_0}\{\pi_n(\theta^{(j)})/\widetilde{\pi}_n(\theta^{(j)})\}} \qquad i = 1, \ldots, N$$

In the following simulation, we choose the $Student(5)$ proposal above, and set $N_0 = 50000$ and $N = 10000$

```
set.seed(94)
N0<-50000
N<-10000
th0<-rmvt(N0,post.mu,post.Sig,5)
lprop<-dmvt(th0,post.mu,post.Sig,5,log=T)
llike<-rep(0,N0)
be0<-exp(th0)
for(i in 1:N0){
    res<-log(y) - log(D*be0[i,1]*(exp(-be0[i,2]*x)-exp(-(be0[i,2]+be0[i,3])*x)))
    Sval<-sum(res^2)
    pth<-t(th0[i,]) %*% (L %*% (th0[i,]))
    llike[i]<--(length(x)+a+3)*log(Sval+pth[1,1]+b)
}
w<-exp(llike-lprop)
w<-w/sum(w)
th.sir<-th0[sample(1:N0,prob=w,rep=T,size=N),]
par(mar=c(4,4,2,0));pairs(th.sir,pch=19,cex=0.5,
    labels=c(expression(theta[0]),expression(theta[1]),expression(theta[2])))
```



The rejection and SIR samples have very similar statistical summaries:

```
apply(theta.rej,2,mean); apply(th.sir,2,mean) #Means

+ [1]  0.6380792 -0.9315390  0.1204690
+ [1]  0.6684064 -0.9248542  0.0767214

cov(theta.rej);cov(th.sir)                         #Covariances

+              [,1]        [,2]        [,3]
+ [1,]  0.09190180  0.03291669 -0.11467162
+ [2,]  0.03291669  0.01348374 -0.03970165
+ [3,] -0.11467162 -0.03970165  0.22106960
+              [,1]        [,2]        [,3]
+ [1,]  0.11882249  0.03991121 -0.15006692
+ [2,]  0.03991121  0.01523303 -0.04909684
+ [3,] -0.15006692 -0.04909684  0.26581996
```

We can measure the efficiency of the Importance Sampling proposal by computing the *effective sample size* (ESS)

$$\mathrm{ESS} = \left(\sum_{i=1}^{N_0} w_i^2\right)^{-1}$$

which is to be compared against the largest possible value – N_0 – which is obtained when $w_i = 1/N_0$ for all i. Here

```
ESS<-1/sum(w^2)
ESS

+ [1] 19874.68
```