MATH 559: BAYESIAN THEORY AND METHODS

BAYESIAN MODELLING WITH THE STUDENT-T MODEL

A model is to be constructed under an assumption of exchangeability with the following components:

- Finite realization $y_1, \ldots, y_n$ recorded;
- $\mathcal{Y} \equiv \mathbb{R}$;
- $\theta = (\mu, \sigma)$;
- $f_Y(y; \theta) \equiv Student(\nu, \mu, \sigma)$

$$f_Y(y;\theta) = \frac{\Gamma\left(\dfrac{\nu+1}{2}\right)}{\Gamma\left(\dfrac{\nu}{2}\right)\sigma\sqrt{\pi\nu}\left\{1+\dfrac{1}{\nu}\left(\dfrac{y-\mu}{\sigma}\right)^2\right\}^{(\nu+1)/2}}$$

  where $\nu > 0$ is a **fixed constant**.

- $\pi_0(\mu, \sigma)$ a prior density on $\mathbb{R} \times \mathbb{R}^+$.

Consider a data-generating scenario where the true values of the parameters are $\mu_0 = 2, \sigma_0 = 1$ and consider a sample of size $n = 20$ with $\nu = 3$. Note that if $Y \sim Student(\nu, \mu, \sigma)$, then we have that

$$Z = \frac{Y - \mu}{\sigma} \sim Student(\nu, 0, 1) \equiv Student(\nu).$$

Thus Z has the 'standard' Student-t distribution with ν degrees of freedom, and in `R` the functions `rt` and `dt` compute for this standard version.

```
#Data generation
set.seed(234)
n<-20
mu0<-2;sigma0<-1
nu<-3
y<-rt(n,nu)*sigma0+mu0
round(y,4)

+  [1]  2.3469  2.1329  1.8627  3.2638  2.4353  3.0528  0.9068  2.6401  2.1668
+ [10]  2.7738  0.3462 -3.4361  2.5891  1.9087  2.3068  3.0496  1.1758  1.6915
+ [19]  1.1288  4.7298
```

A conjugate analysis is not possible in this model, so we are forced to use numerical methods. We will retain the Normal-Inverse Gamma prior that would be conjugate for a Normal likelihood, with

$$\pi_0(\mu, \sigma^2) = \pi_0(\mu|\sigma^2)\pi_0(\sigma^2)$$

with factors

$$\pi_0(\mu|\sigma^2) \equiv Normal(\eta, \sigma^2/\lambda)$$

and

$$\pi_0(\sigma^2) = InvGamma(\alpha_0/2, \beta_0/2)$$

where we take $\eta = 2, \lambda = 0.1, \alpha_0 = 2$ and $\beta_0 = 4$ in the illustrative analysis.

Approach 1: Direct approach using Gibbs sampler
Here the two parameters are updated recursively from their full conditionals

$$\pi_n(\mu|\sigma^2) \qquad \pi_n(\sigma^2|\mu)$$

using Metropolis-Hastings updates for each, as there is no standard form for either density. Note that if

$$\mathcal{L}_n(\mu,\sigma^2) = \prod_{i=1}^{n} f_Y(y_i;\mu,\sigma^2)$$

represents the likelihood, we have that

$$\pi_n(\mu|\sigma^2) \propto \mathcal{L}_n(\mu,\sigma^2)\pi_0(\mu|\sigma^2)$$

$$\pi_n(\sigma^2|\mu) \propto \mathcal{L}_n(\mu,\sigma^2)\pi_0(\mu|\sigma^2)\pi_0(\sigma^2)$$

```
#Function definitions
dinvgamma<-function(x,a,b,log=FALSE){
    dx<-(b^a/gamma(a))*(1/x)^(a+1)*exp(-b/x)
    if(log){
        dx<-log(dx)
    }
    return(dx)
}

log.like<-function(yv,mv,sv,nuv){
    #Use dt on the standardized data, but remember to include
    #the Jacobian for the scale transform
    xv<-(yv-mv)/sv
    return(sum(dt(xv,nuv,log=T)-log(sv)))
}
```

```
#MCMC collection settings
nsamp<-10000
nburn<-50
nthin<-10
nits<-nburn+nthin*nsamp

#Prior distributions
eta<-2; lambda<-0.1
al<-2; be<-4

#Starting set up
old.mu<-0
old.sigsq<-1
old.like<-log.like(y,old.mu,sqrt(old.sigsq),nu)
old.prior.mu<-dnorm(old.mu,eta,sqrt(old.sigsq/lambda),log=TRUE)
old.prior.sig<-dinvgamma(old.sigsq,al/2,be/2,log=TRUE)
old.post<-old.like+old.prior.mu+old.prior.sig
```

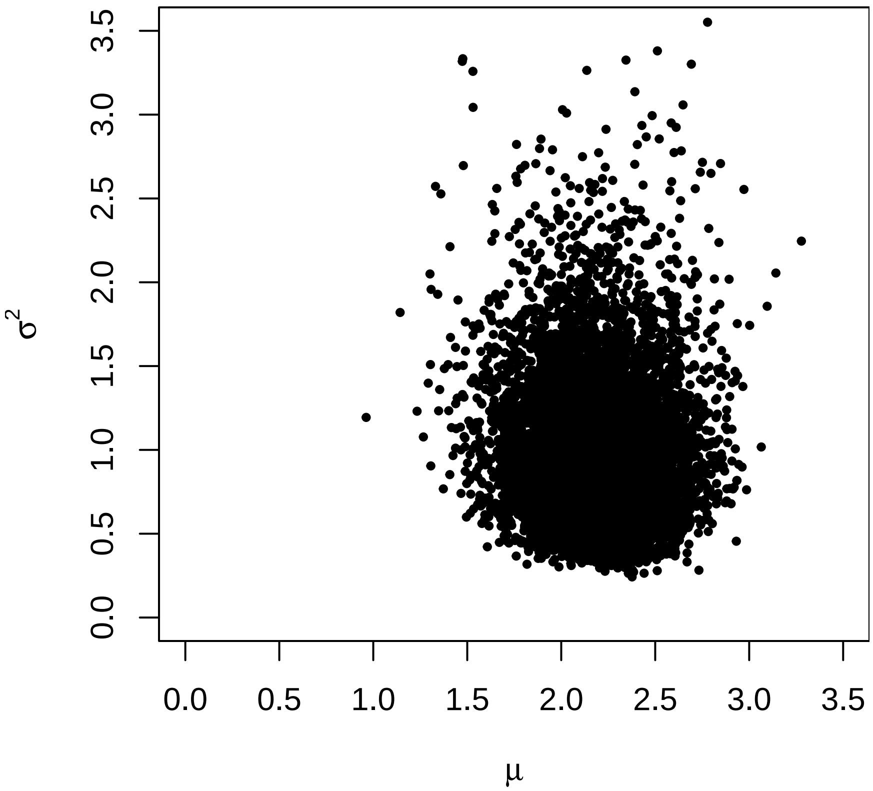
We will run the Gibbs sampler with Metropolis updates, using Normal and reflected Normal proposals for μ and σ respectively, that is, using proposal densities

$$q(\mu,z) \equiv Normal(\mu,\sigma_m^2)$$

$$q(\sigma,z) : Z = |\sigma+\delta|, \delta \sim Normal(0,\sigma_s^2)$$

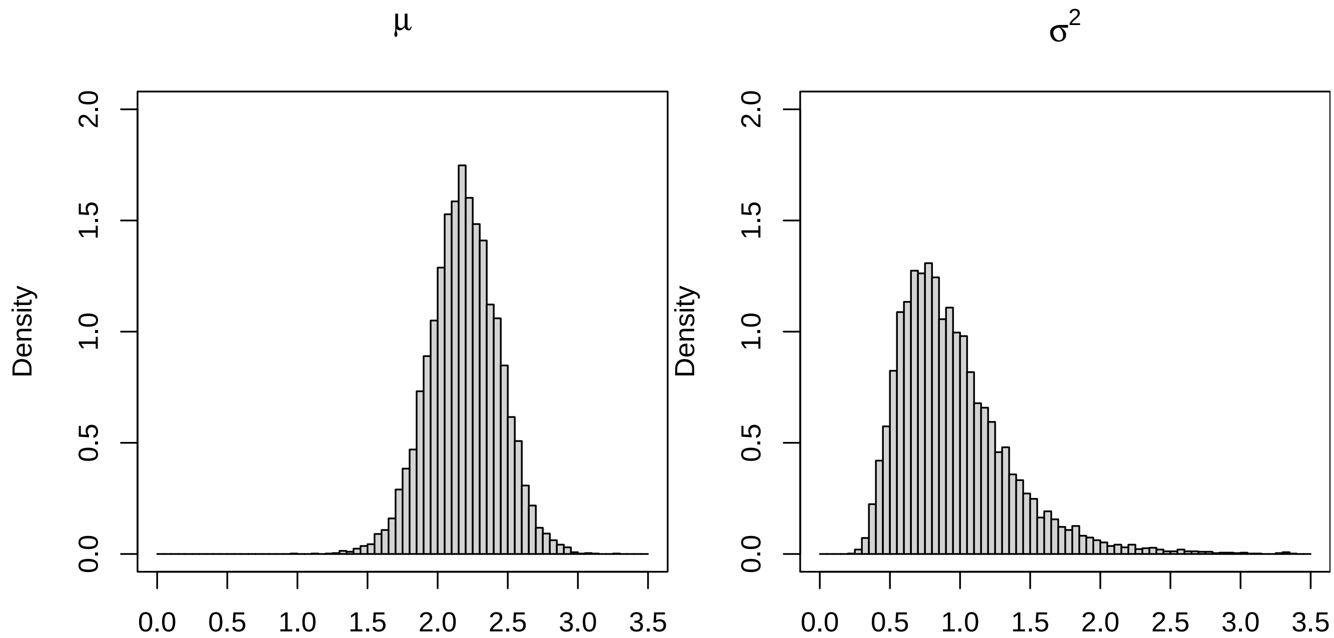
```r
#Gibbs sampler - Metropolis-within-Gibbs

#Proposal parameters
psig.m<-0.2
psig.s<-0.2

#For the accept/reject steps
logu1<-log(runif(nits))
logu2<-log(runif(nits))

#For collecting the samples
ico<-0
par.samp<-matrix(0,nrow=nsamp,ncol=2)
for(iter in 1:nits){

    #Update mu from its full conditional
    new.mu<-rnorm(1,old.mu,psig.m)
    new.like<-log.like(y,new.mu,sqrt(old.sigsq),nu)
    new.prior.mu<-dnorm(new.mu,eta,sqrt(old.sigsq/lambda),log=TRUE)
    new.post<-new.like+new.prior.mu+old.prior.sig
    if(logu1[iter] < new.post-old.post){
        #Accept update
        old.mu<-new.mu
        old.like<-new.like
        old.prior.mu<-new.prior.mu
        old.post<-new.post
    }

    #Update sigma sq from its full conditional
    new.sigsq<-abs(rnorm(1,old.sigsq,psig.s))
    new.like<-log.like(y,old.mu,sqrt(new.sigsq),nu)
    new.prior.mu<-dnorm(old.mu,eta,sqrt(new.sigsq/lambda),log=TRUE)
    new.prior.sig<-dinvgamma(new.sigsq,al/2,be/2,log=TRUE)
    new.post<-new.like+new.prior.mu+new.prior.sig
    if(logu2[iter] < new.post-old.post){
        #Accept update
        old.sigsq<-new.sigsq
        old.like<-new.like
        old.prior.mu<-new.prior.mu
        old.prior.sig<-new.prior.sig
        old.post<-new.post
    }

    #Collect output
    if(iter > nburn & iter %% nthin ==0){
        ico<-ico+1
        par.samp[ico,]<-c(old.mu,old.sigsq)
    }
}
```

```r
par(mar=c(4,4,2,0))
mulab<-expression(mu);siglab<-expression(sigma^2)
plot(par.samp,pch=19,cex=0.5,xlab=mulab,ylab=siglab,xlim=range(0,3.5),ylim=range(0,3.5))
```

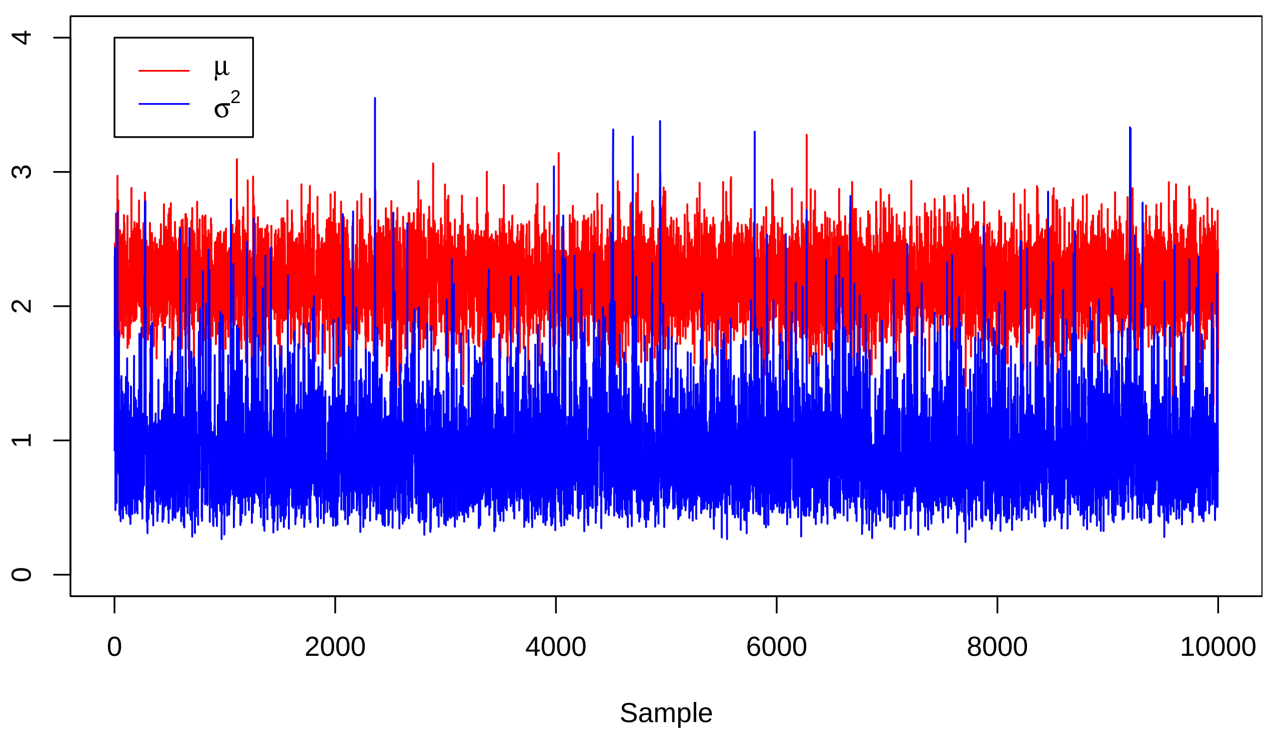



```
#Summaries
mean(par.samp[,1])

+ [1] 2.187971

mean(par.samp[,2])

+ [1] 0.9545342

var(par.samp)

+               [,1]         [,2]
+ [1,]  0.061063005 -0.005058613
+ [2,] -0.005058613  0.157071592
```

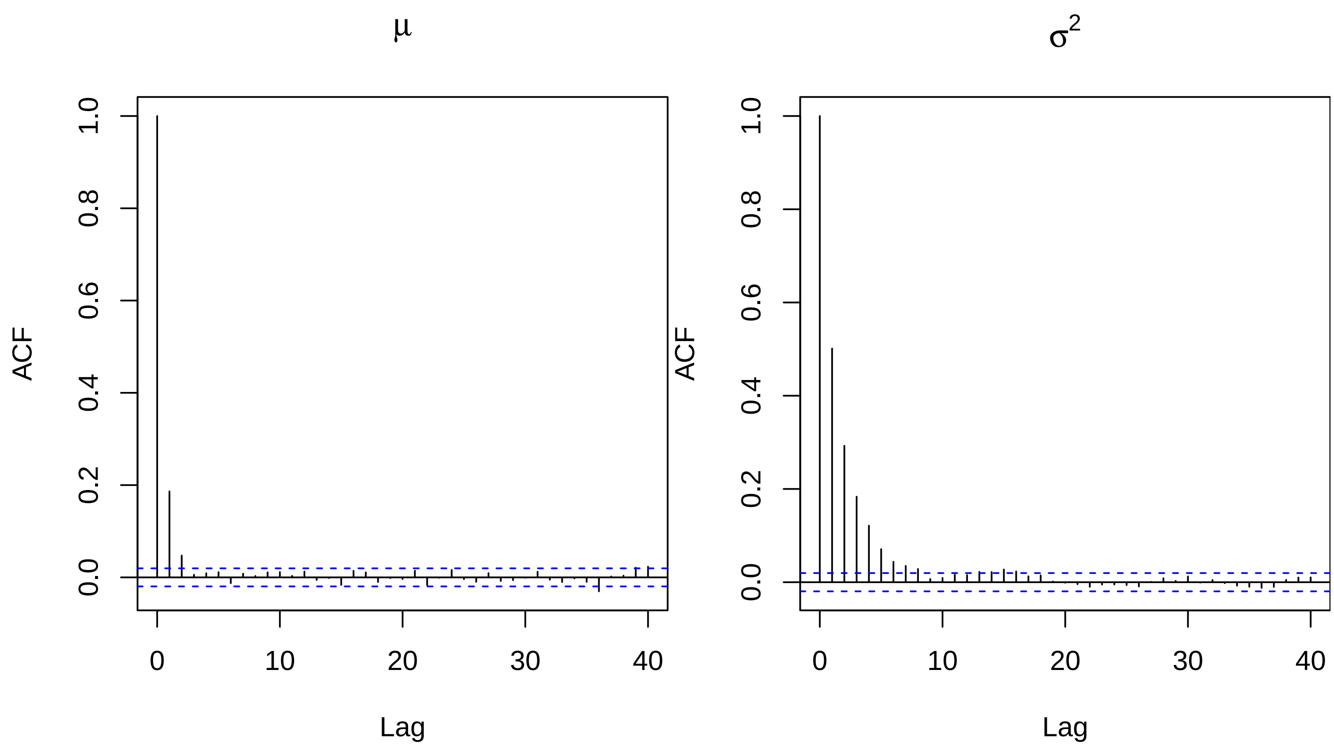
```
#Posterior histograms
par(mar=c(4,4,4,0),mfrow=c(1,2))
hist(par.samp[,1],breaks=seq(0,3.5,by=0.05),ylim=range(0,2),
    freq=FALSE,main=mulab,xlab='');box()
acc<-par.samp[,2]<3.5
hist(par.samp[acc,2],breaks=seq(0,3.5,by=0.05),,ylim=range(0,2),
    freq=FALSE,main=siglab,xlab='');box()
```




```r
#Trace plots
par(mar=c(4,4,1,0))
plot(par.samp[,1],type='l',col='red',xlab='Sample',ylab='',ylim=range(0,4))
lines(par.samp[,2],type='l',col='blue')
legend(0,4,c(mulab,siglab),lty=1,col=c('red','blue'))
```



```
#ACF plots
par(mar=c(4,4,3,0),mfrow=c(1,2))
acf(par.samp[,1],main=mulab)
acf(par.samp[,2],main=siglab)
```




```
#Effective sample size
library(coda)
effectiveSize(par.samp)

+     var1     var2
+ 6856.589 2839.718

#Efficiency
effectiveSize(par.samp)/nsamp

+      var1      var2
+ 0.6856589 0.2839718
```

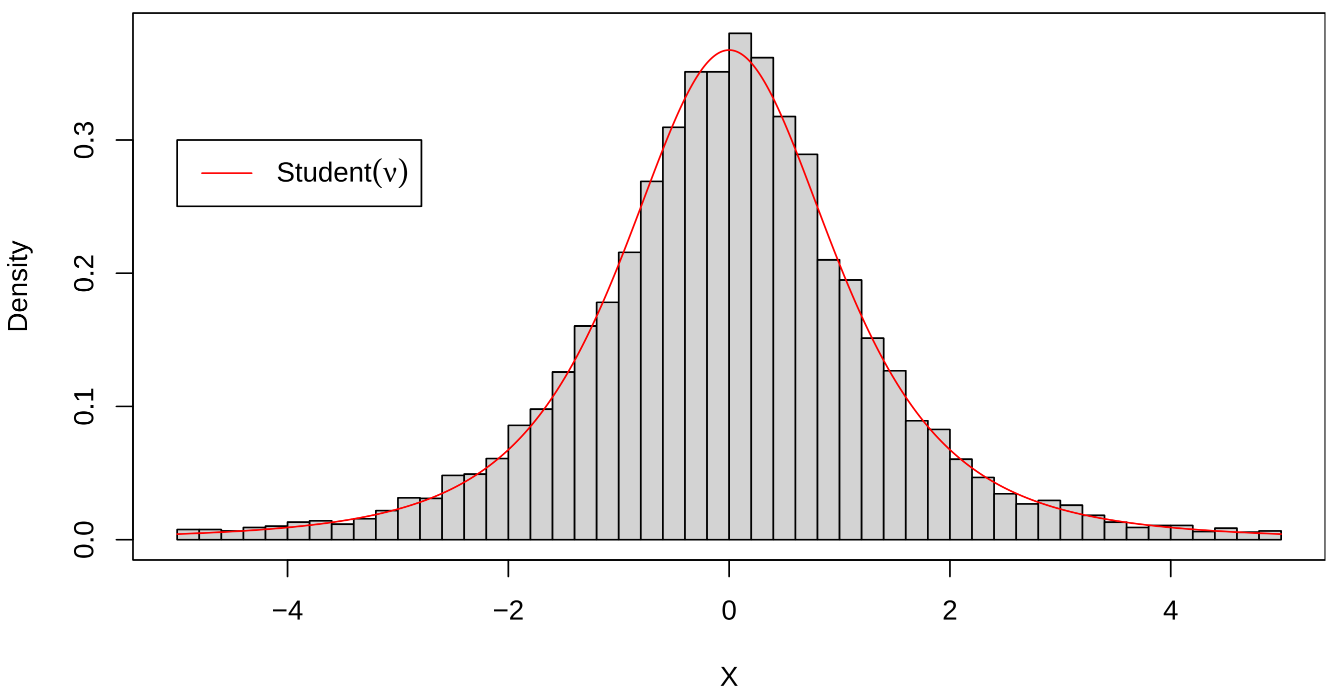
Approach 2: Gibbs sampler with auxiliary variables

For the Student-t model, we can use an auxiliary variable representation to remove the need for a Metropolis-Hastings update and rely on a Gibbs sampler where all full conditionals are standard distributions. The representation arises as if

$$V \sim InvGamma(\nu/2, \nu/2) \qquad\qquad X|V = v \sim Normal(0, v)$$

then a standard calculation verifies that $X \sim Student(\nu)$.

```
set.seed(38)
V<-1/rgamma(10000,nu/2,nu/2)  #Generate the inverse gamma V
X<-rnorm(10000,0,sqrt(V))      #Generate the conditional normal
inc<-(abs(X) < 5)
par(mar=c(4,4,2,0))
hist(X[inc],breaks=seq(-5,5,0.2),freq=FALSE,main='',xlab='X');box()
xv<-seq(-5,5,by=0.001)
yv<-dt(xv,nu)            #Student(nu) density
lines(xv,yv,col='red')
legend(-5,0.3,c(expression(Student(nu))),col='red',lty=1)
```



For our inference problem we may therefore using a likelihood that includes the auxiliary variables based on using the **joint** density

$$f_{Y,V}(y, v; \mu, \sigma^2) = f_{Y|V}(y|v; \mu, \sigma^2) f_V(v).$$

for which the marginal density

$$f_Y(y; \mu, \sigma^2) = \int_0^\infty f_{Y|V}(y|v; \mu, \sigma^2) f_V(v)\, dv$$

is the correct target $Student(\nu, \mu, \sigma)$. Specifically, we now have the auxiliary variable likelihood that includes the auxiliary variables $\mathbf{v} = (v_1, \ldots, v_n)$ as unknown quantities

$$\mathcal{L}_n(\mu, \sigma, \mathbf{v}) = \prod_{i=1}^{n} f_{Y|V}(y_i|v_i; \mu, \sigma^2) f_V(v_i)$$

which takes the form

$$\mathcal{L}_n(\mu,\sigma,\mathbf{v}) = \prod_{i=1}^{n}\left(\frac{1}{2\pi v_i\sigma^2}\right)^{1/2}\exp\left\{-\frac{1}{2v_i\sigma^2}(y_i-\mu)^2\right\}\frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)}\left(\frac{1}{v_i}\right)^{\nu/2+1}\exp\left\{-\frac{v_i\nu}{2}\right\}$$
$$\propto \left(\frac{1}{\sigma^2}\right)^{n/2}\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\frac{1}{v_i}(y_i-\mu)^2\right\}\left\{\prod_{i=1}^{n}\left(\frac{1}{v_i}\right)^{(\nu+1)/2+1}\right\}\exp\left\{-\frac{\nu}{2}\sum_{i=1}^{n}v_i\right\}$$

which combines with the prior

$$\pi_0(\mu|\sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{1/2}\exp\left\{-\frac{\lambda}{2\sigma^2}(\mu-\eta)^2\right\}$$
$$\pi_0(\sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{\alpha_0/2+1}\exp\left\{-\frac{\beta_0}{2\sigma^2}\right\}$$

to yield the joint posterior $\pi_n(\mu,\theta,\mathbf{v})$ up to proportionality. For the Gibbs sampler, we need the full conditional posteriors:

- Full conditional for μ:

$$\pi_n(\mu|\sigma^2,\mathbf{v}) \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\frac{1}{v_i}(y_i-\mu)^2\right\}\exp\left\{-\frac{\lambda}{2\sigma^2}(\mu-\eta)^2\right\}$$
$$\propto \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}\frac{1}{v_i}(y_i-\mu)^2+\lambda(\mu-\eta)^2\right]\right\}$$
$$\propto \exp\left\{-\frac{\lambda_n(\mathbf{v})}{2\sigma^2}(\mu-\eta_n(\mathbf{v}))^2\right\}$$

  where

$$\eta_n(\mathbf{v}) = \frac{\sum_{i=1}^{n}(y_i/v_i)+\lambda\eta}{\sum_{i=1}^{n}(1/v_i)+\lambda} \qquad \lambda_n(\mathbf{v}) = \lambda+\sum_{i=1}^{n}(1/v_i).$$

  Therefore

$$\pi_n(\mu|\sigma^2,\mathbf{v}) \equiv Normal(\eta_n(\mathbf{v}),\sigma^2/\lambda_n(\mathbf{v})).$$

- Full conditional for σ^2:

$$\pi_n(\sigma^2|\mu,\mathbf{v}) \propto \left(\frac{1}{\sigma^2}\right)^{(n+\alpha_0+1)/2+1}\exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i=1}^{n}\frac{1}{v_i}(y_i-\mu)^2+\lambda(\mu-\eta)^2+\beta_0\right]\right\}$$

  so therefore

$$\pi_n(\sigma^2|\mu,\mathbf{v}) \equiv InvGamma(\alpha_n/2,\beta_n(\mu,\mathbf{v})/2)$$

  where

$$\alpha_n = \alpha_0+n+1 \qquad \beta_n(\mu,\mathbf{v}) = \sum_{i=1}^{n}\frac{1}{v_i}(y_i-\mu)^2+\lambda(\mu-\eta)^2+\beta_0.$$

- Full conditional for $v_i, i = 1, \ldots, n$: we can see from the form of $\pi_n(\mu, \sigma^2, \mathbf{v})$ that

$$\pi_n(\mathbf{v}|\mu, \sigma^2) = \prod_{i=1}^{n} \pi_n(v_i|\mu, \sigma^2)$$

  where

$$\pi_n(v_i|\mu, \sigma^2) \propto \left(\frac{1}{v_i}\right)^{(\nu+1)/2+1} \exp\left\{-\frac{1}{2v_i}\left[\nu + \frac{(y_i - \mu)^2}{\sigma^2}\right]\right\}$$

  that is,

$$\pi_n(v_i|\mu, \sigma^2) \equiv InvGamma((\nu + 1)/2, \varphi_i(\mu, \sigma^2)/2)$$

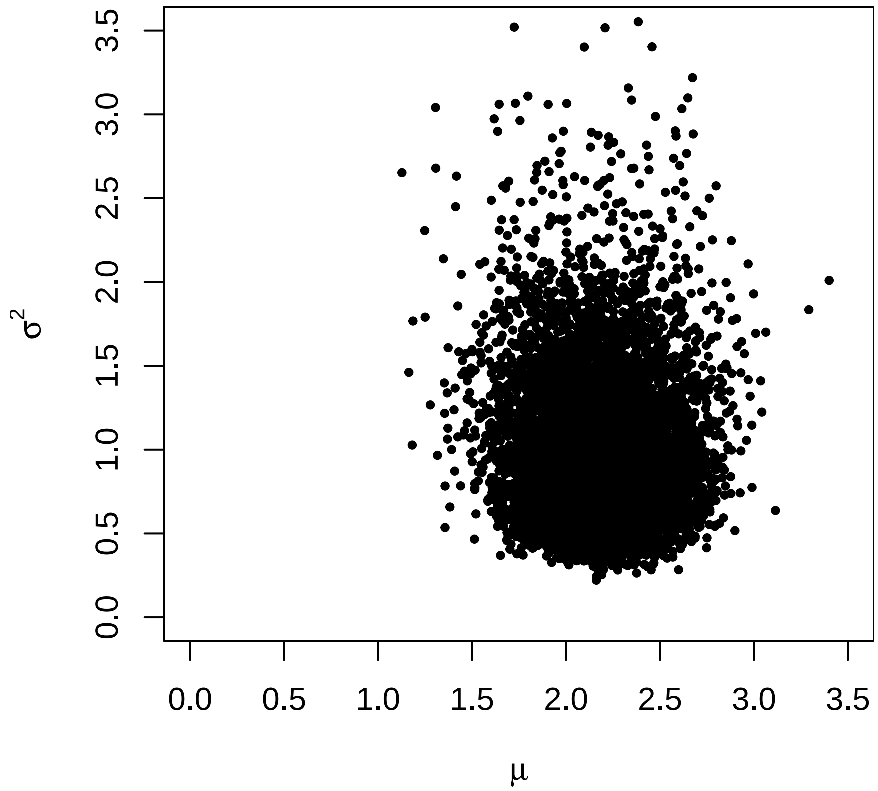
  where

$$\varphi_i(\mu, \sigma^2) = \nu + \frac{(y_i - \mu)^2}{\sigma^2}$$

Therefore all the full conditionals are standard distributions, and the elements of $\mathbf{v}$ can be updated independently in parallel.

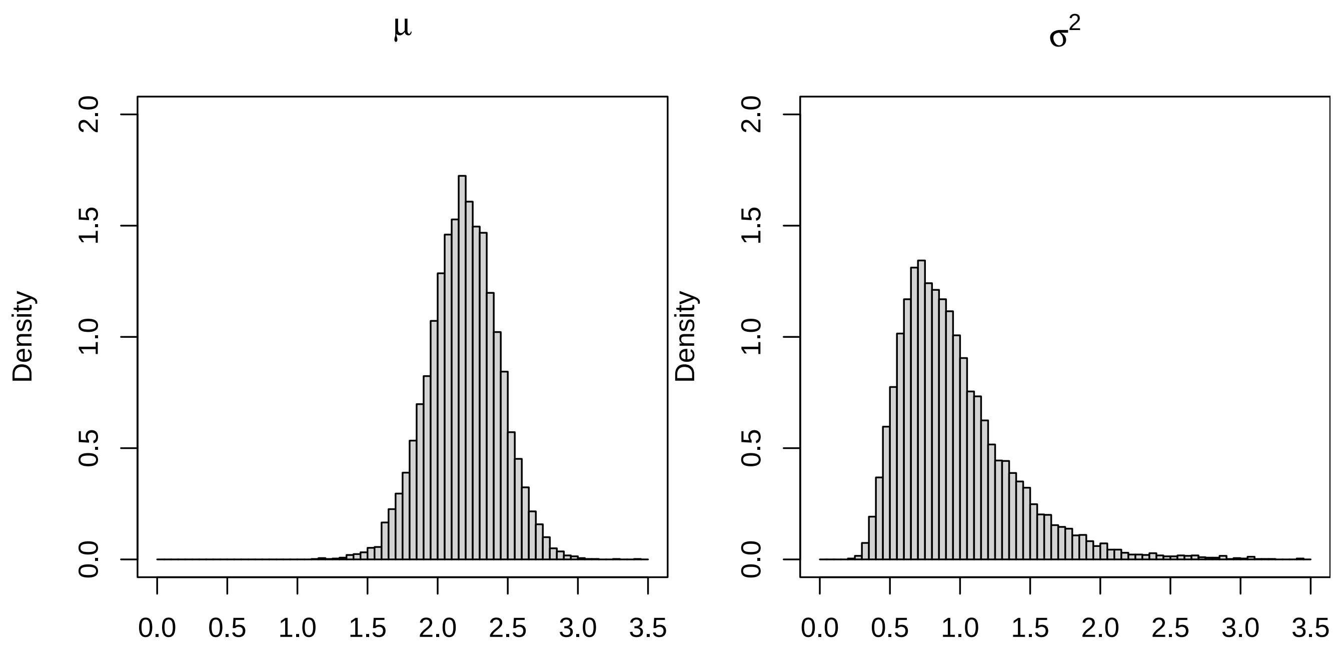
```
old.mu<-0
old.sigsq<-1
old.v<-rep(1,n)
ico<-0
par.samp.A<-matrix(0,nrow=nsamp,ncol=2)
v.samp.A<-matrix(0,nrow=nsamp,ncol=n)

for(iter in 1:nits){

    #Update mu
    lam.n<-sum(1/old.v)+lambda
    eta.n<-(sum(y/old.v)+eta*lambda)/lam.n
    old.mu<-rnorm(1,eta.n,sqrt(old.sigsq/lam.n))

    #Update sigma^2
    al.n<-al+n+1
    be.n<-sum((y-old.mu)^2/old.v)+lambda*(old.mu-eta)^2 + be
    old.sigsq<-1/rgamma(1,al.n/2,be.n/2)

    #Update v
    old.v<-1/rgamma(n,(nu+1)/2,(nu+(y-old.mu)^2/old.sigsq)/2)

    #Collect output
    if(iter > nburn & iter %% nthin ==0){
        ico<-ico+1
        par.samp.A[ico,]<-c(old.mu,old.sigsq)
        v.samp.A[ico,]<-old.v
    }
}
```

```
par(mar=c(4,4,2,0))
plot(par.samp.A,pch=19,cex=0.5,xlab=mulab,ylab=siglab,xlim=range(0,3.5),ylim=range(0,3.5))
```

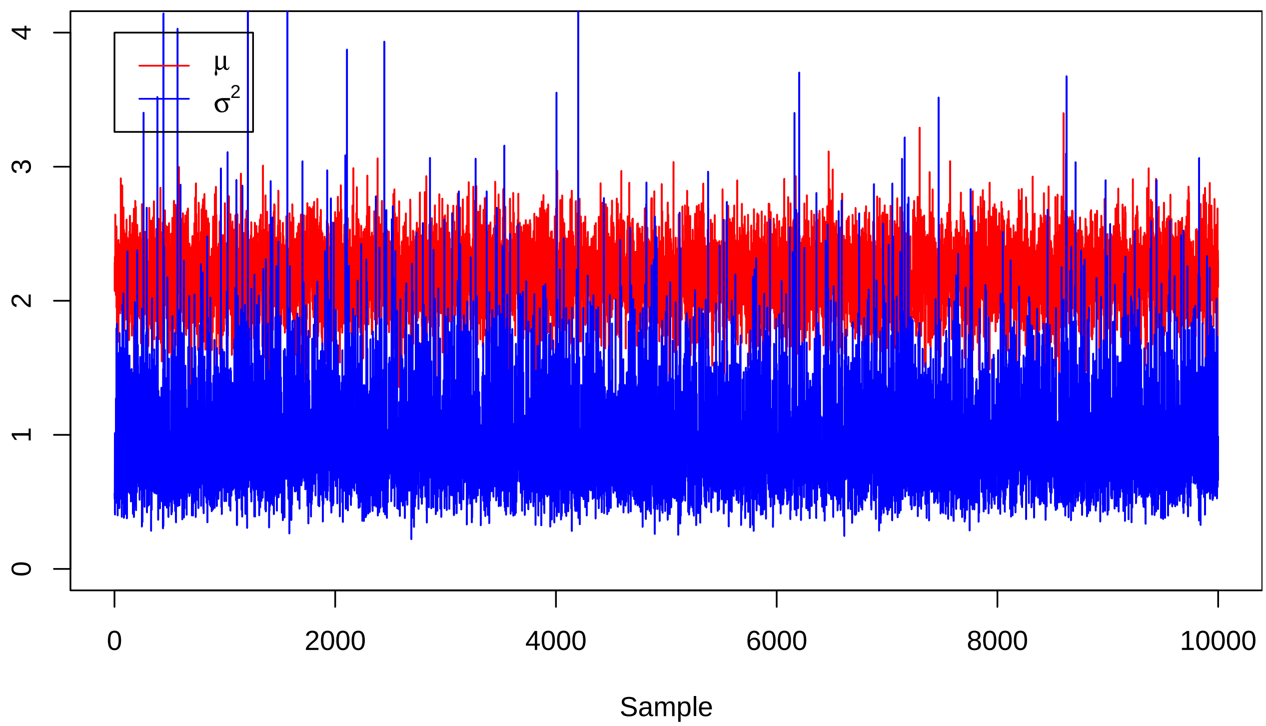



```
#Summaries
mean(par.samp.A[,1])

+ [1] 2.185484

mean(par.samp.A[,2])

+ [1] 0.9666655

var(par.samp.A)

+               [,1]         [,2]
+ [1,]  0.062621136 -0.005672293
+ [2,] -0.005672293  0.174804277
```

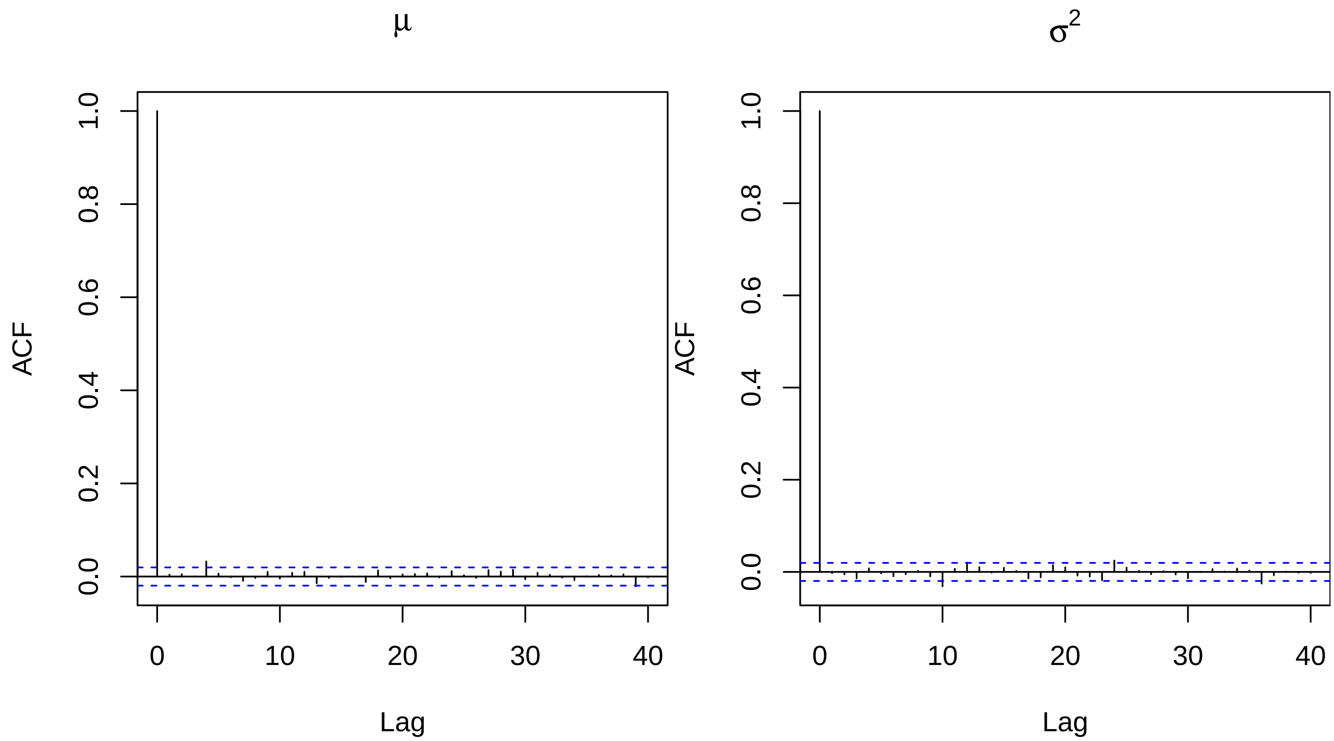
```
#Posterior histograms
par(mar=c(4,4,4,0),mfrow=c(1,2))
inc<-par.samp.A[,2]<3.5
hist(par.samp.A[,1],breaks=seq(0,3.5,by=0.05),ylim=range(0,2),
    freq=FALSE,main=mulab,xlab='');box()
hist(par.samp.A[inc,2],breaks=seq(0,3.5,by=0.05),,ylim=range(0,2),
    freq=FALSE,main=siglab,xlab='');box()
```




```
#Trace plots
par(mar=c(4,4,1,0))
plot(par.samp.A[,1],type='l',col='red',xlab='Sample',ylab='',ylim=range(0,4))
lines(par.samp.A[,2],type='l',col='blue')
legend(0,4,c(mulab,siglab),lty=1,col=c('red','blue'))
```



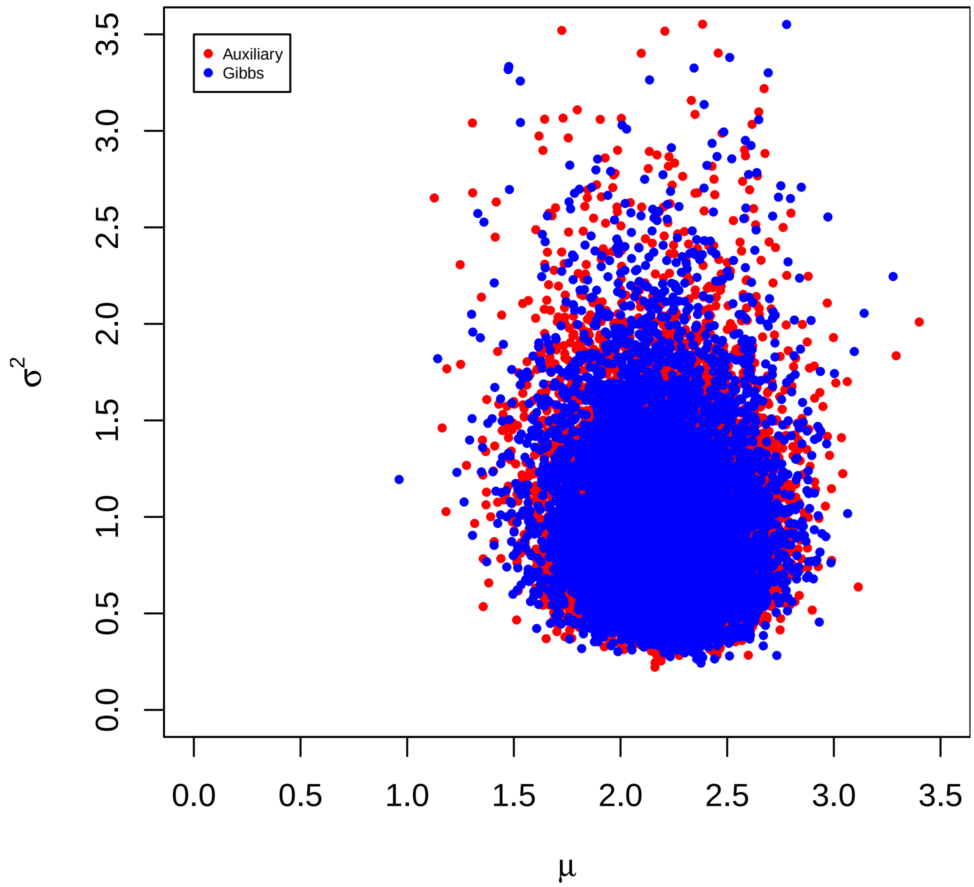
```r
#ACF plots
par(mar=c(4,4,3,0),mfrow=c(1,2))
acf(par.samp.A[,1],main=mulab)
acf(par.samp.A[,2],main=siglab)
```




```r
#Effective sample size
library(coda)
effectiveSize(par.samp.A)

+      var1      var2
+  9202.815 10000.000

#Efficiency
effectiveSize(par.samp.A)/nsamp

+      var1      var2
+ 0.9202815 1.0000000
```

To compare the two MCMC approaches, we may overlay the two posterior samples:

```r
par(mar=c(4,4,2,0))
plot(par.samp.A,pch=19,cex=0.5,col='red',
    xlab=mulab,ylab=siglab,xlim=range(0,3.5),ylim=range(0,3.5))
points(par.samp,col='blue',pch=19,cex=0.5)
legend(0,3.5,c('Auxiliary','Gibbs'),pch=19,cex=0.5,col=c('red','blue'))
```

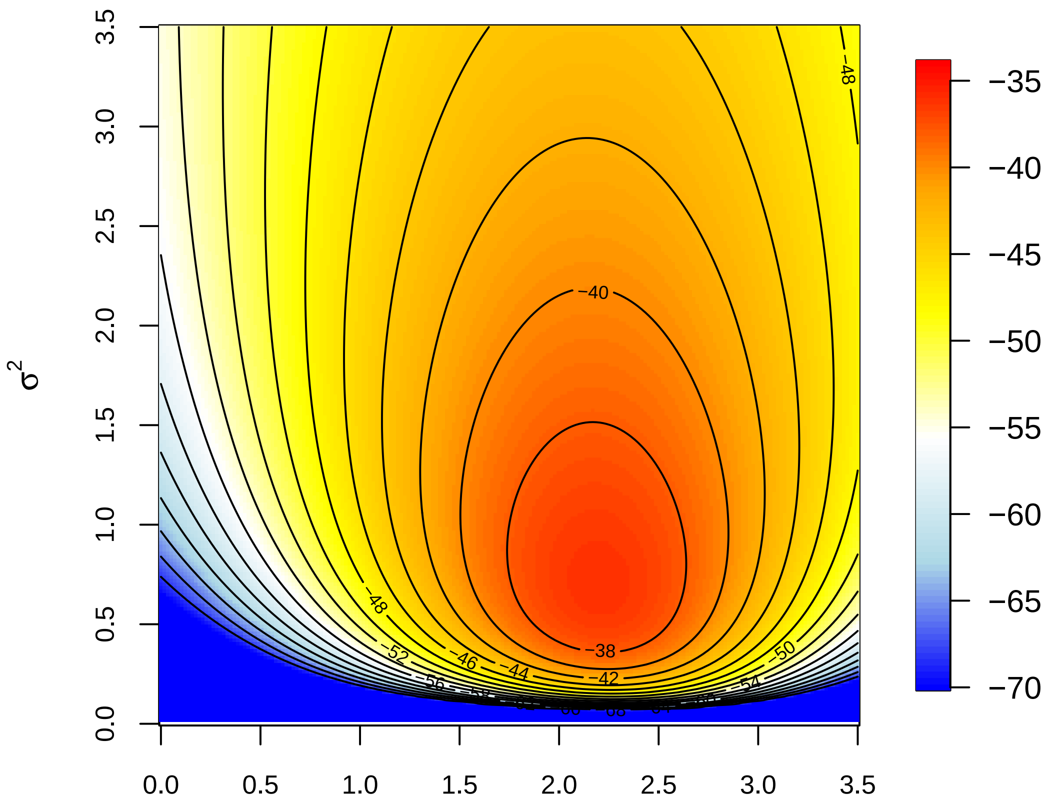


It seems that the auxiliary variable method explores more of the parameter space.

Approach 3: Rejection sampling
For rejection sampling, we may use a bivariate proposal distribution with parameters estimated by finding a Normal approximation near the posterior mode that we find using `optim`.

```
log.post<-function(parv,yv,etv,lamv,a0v,b0v,nuv){
    t1<-log.like(yv,parv[1],sqrt(parv[2]),nuv)
    t2<-dnorm(parv[1],etv,sqrt(parv[2])/sqrt(lamv),log=TRUE)
    t3<-dinvgamma(parv[2],a0v/2,b0v/2,log=TRUE)
    return(t1+t2+t3)
}

pstart<-c(0,1)
olist<-list(fnscale=-1)
post.max<-optim(pstart,fn=log.post,yv=y,etv=eta,lamv=lambda,a0v=al,b0v=be,nuv=nu,
                control=olist,hessian=T)
rej.mu<-post.max$par
rej.mu                           #Proposal mean

+ [1] 2.2005388 0.7029691

rej.Sig<-solve(-post.max$hessian)
rej.Sig                          #Proposal variance matrix

+               [,1]         [,2]
+ [1,]  0.049461119 -0.004161044
+ [2,] -0.004161044  0.067296278
```

```
gpts<-seq(0,1,by=0.005)
mu1<-sig1<-gpts*3.5 #Grids

#Compute the posterior on the log scale
log.post.func1<-function(muv,sigv,yv,etv,lamv,a0v,b0v,nuv){
    parv<-c(muv,sigv)
    t1<-log.post(parv,yv,etv,lamv,a0v,b0v,nuv)
    return(t1)
}
f <- Vectorize(log.post.func1,vectorize.args=c("muv","sigv"))
log.p1<-outer(mu1,sig1,f,yv=y,etv=eta,lamv=lambda,a0v=al,b0v=be,nuv=nu)
log.p1[log.p1 < -70]<--70
par(pty='s',mar=c(2,3,2,2))
colfunc <- colorRampPalette(c("blue","lightblue","white","yellow","orange","red"))
image.plot(mu1,sig1,log.p1,col=colfunc(100),zlim=range(-70,-34),
           xlab=mulab,ylab=siglab,cex.axis=0.8)
contour(mu1,sig1,log.p1,add=T,levels=seq(-70,-34,by=2))
```

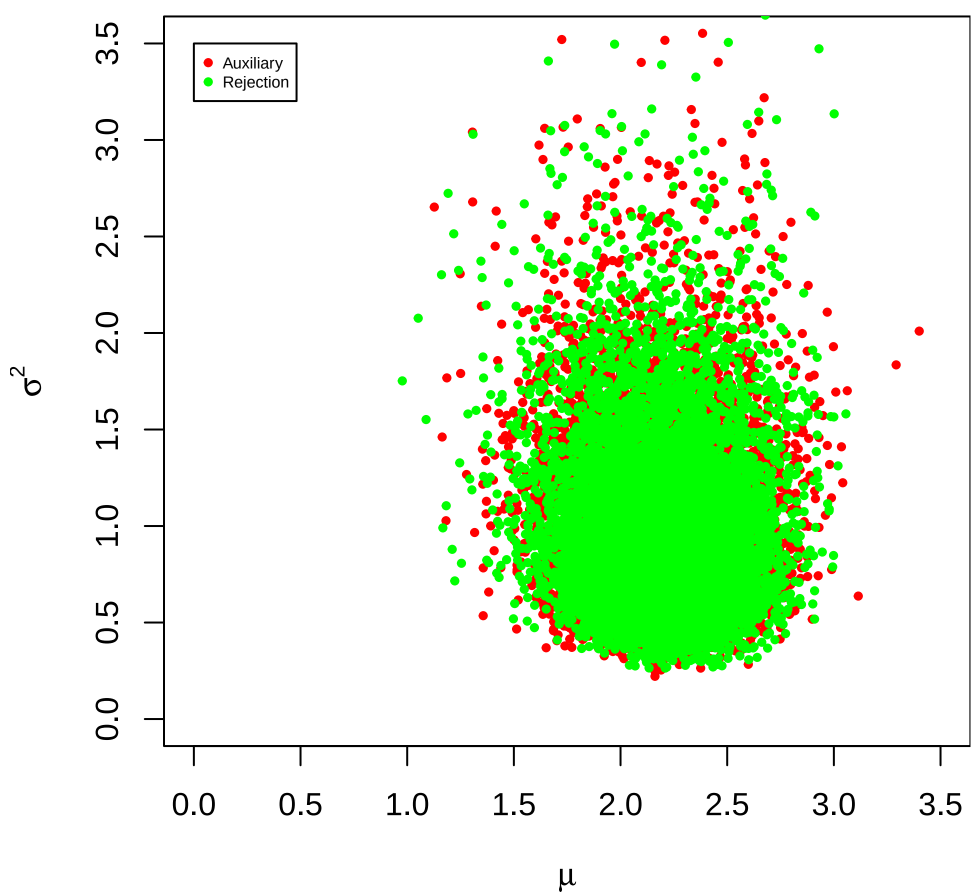


We propose from a bivariate Student-t distribution with degrees of freedom 2 using the mvnfast library functions `dmvt` and `rmvt`.

```
library(mvnfast)

log.rat<-function(parv,yv,etv,lamv,a0v,b0v,nuv,Muv,Sigv){
    if(parv[2]<0){return(-Inf)}
    t1<-log.like(yv,parv[1],sqrt(parv[2]),nuv)
    t2<-dnorm(parv[1],etv,sqrt(parv[2])/sqrt(lamv),log=TRUE)
    t3<-dinvgamma(parv[2],a0v/2,b0v/2,log=TRUE)
    t4<-dmvt(parv,Muv,Sigv,2,log=T)
    return(t1+t2+t3-t4)
}

pstart<-c(0,1)
olist<-list(fnscale=-1)
rat.max<-optim(pstart,fn=log.rat,yv=y,etv=eta,lamv=lambda,a0v=al,b0v=be,nuv=nu,
                Muv=rej.mu,Sigv=rej.Sig,control=olist,hessian=T)
logM<-rat.max$value

nacc<-0
ico<-0
while(nacc < nsamp){
    th<-rmvt(nsamp,rej.mu,rej.Sig,2)
    u<-log(runif(nsamp))
    r<-apply(th,1,log.rat,yv=y,etv=eta,lamv=lambda,a0v=al,b0v=be,nuv=nu,
                Muv=rej.mu,Sigv=rej.Sig)
    acc<-u < r - logM
```

```
    if(nacc==0){
        rej.samp<-th[acc,]
    }else{
        rej.samp<-rbind(rej.samp,th[acc,])
    }
    nacc<-nacc+sum(acc)
    ico<-ico+nsamp

}
c(ico,nacc,nacc/ico)                #Acceptance rate

+ [1] 60000.0000000 11408.0000000     0.1901333

rej.samp<-rej.samp[1:nsamp,]
```

```
par(mar=c(4,4,2,0))
plot(par.samp.A,pch=19,cex=0.5,col='red',
    xlab=mulab,ylab=siglab,xlim=range(0,3.5),ylim=range(0,3.5))
points(rej.samp,col='green',pch=19,cex=0.5)
legend(0,3.5,c('Auxiliary','Rejection'),pch=19,cex=0.5,col=c('red','green'))
```



A comparison shows that the auxiliary variable MCMC sampler matches the exact rejection sampler.

```r
var(par.samp)    #Gibbs sampler

+              [,1]         [,2]
+ [1,]  0.061063005 -0.005058613
+ [2,] -0.005058613  0.157071592

var(par.samp.A)  #Auxiliary variable sampler

+              [,1]         [,2]
+ [1,]  0.062621136 -0.005672293
+ [2,] -0.005672293  0.174804277

var(rej.samp)    #Rejection sampling

+              [,1]         [,2]
+ [1,]  0.064157418 -0.006675872
+ [2,] -0.006675872  0.175015601
```