HW5

Zach White

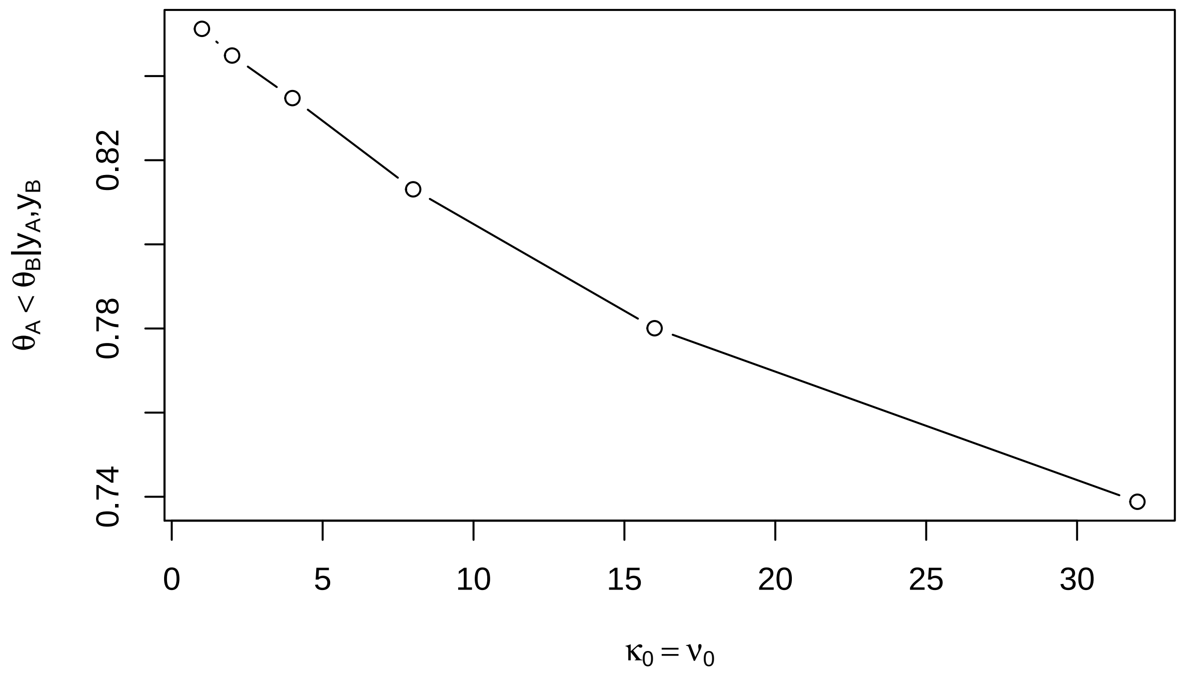
10/14/2016

Exercise 5.2

```
n.a = 16 ; n.b = 16;
k.0 = 2^seq(0,5,by = 1)
nu.0 = 2^seq(0,5,by = 1)

mu.0 = 75
sigma2.0 = 100

y.bar.a = 75.2 ; s.a = 7.3
y.bar.b = 77.5 ; s.b = 8.1

kappa.n.a = k.0 + n.a ; kappa.n.b = k.0 + n.b
mu.n.a = (k.0*mu.0 + n.a*y.bar.a) / kappa.n.a ; mu.n.b = (k.0*mu.0 + n.b*y.bar.b) / kappa.n.b
nu.n.a = nu.0 + n.a; nu.n.b = nu.0 + n.b
sigma2.n.a = (1/nu.n.a)*(nu.0*sigma2.0 + (n.a-1)*s.a^2 + ((k.0*n.a)/(kappa.n.a))*(y.bar.a-mu.0)^2)
sigma2.n.b = (1/nu.n.b)*(nu.0*sigma2.0 + (n.b-1)*s.b^2 + ((k.0*n.b)/kappa.n.b) * (y.bar.b-mu.0)^2)

rounds = length(k.0)
theta.a.draws.mat = matrix(NA,50000,rounds)
theta.b.draws.mat = matrix(NA,50000,rounds)
prob.a.b = rep(NA,6)

for(i in 1:rounds){
  s2a.post.samp = 1/rgamma(50000,nu.n.a/2, sigma2.n.a*nu.n.a/2)
  s2b.post.samp = 1/rgamma(50000,nu.n.b/2, sigma2.n.b*nu.n.b/2)
  theta.a.draws.mat[,i] = rnorm(50000,mu.n.a[i], sqrt(s2a.post.samp / kappa.n.a[i]))
  theta.b.draws.mat[,i] = rnorm(50000,mu.n.b[i], sqrt(s2b.post.samp) / kappa.n.b[i])
  prob.a.b[i] = mean(theta.a.draws.mat[,i] < theta.b.draws.mat[,i])
}

theta.diff = theta.b.draws.mat - theta.a.draws.mat
colMeans(theta.diff > 0)
```

```
## [1] 0.85122 0.84488 0.83476 0.81308 0.78006 0.73882
```

```
plot(k.0,prob.a.b, type = "b", xlab = expression(kappa[0] == nu[0]), ylab = expression(theta[A] < theta
```



It is clear looking at this plot that as (κ_0, ν_0) increase, the $Pr(\theta_A < \theta_B | y_A, y_B)$ decreases also.

Exercise 5.3

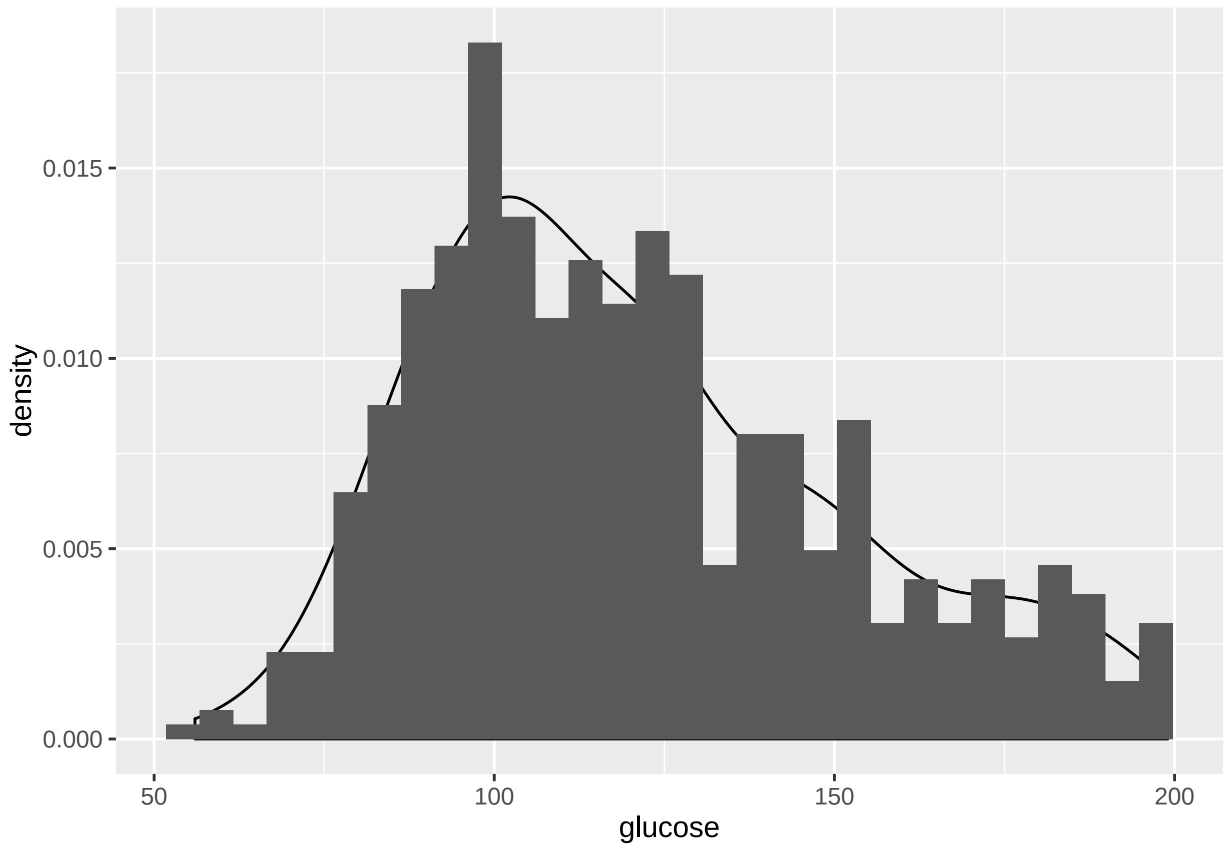
Exercise 6.2

```
glucose = scan("glucose.dat")
```

Part A

```
ggplot(data = as.data.frame(glucose), aes(x = glucose)) + geom_density() + geom_histogram(aes(y = ..dens
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Looking at this plot, it seems as though the area from 50-115 follows a normal curve. However, after that, it doesn't descend as quickly, and so it has a heavier right tail. And in the right tail, there are some interesting features. There are some areas in the tail with higher density than I would have expected.

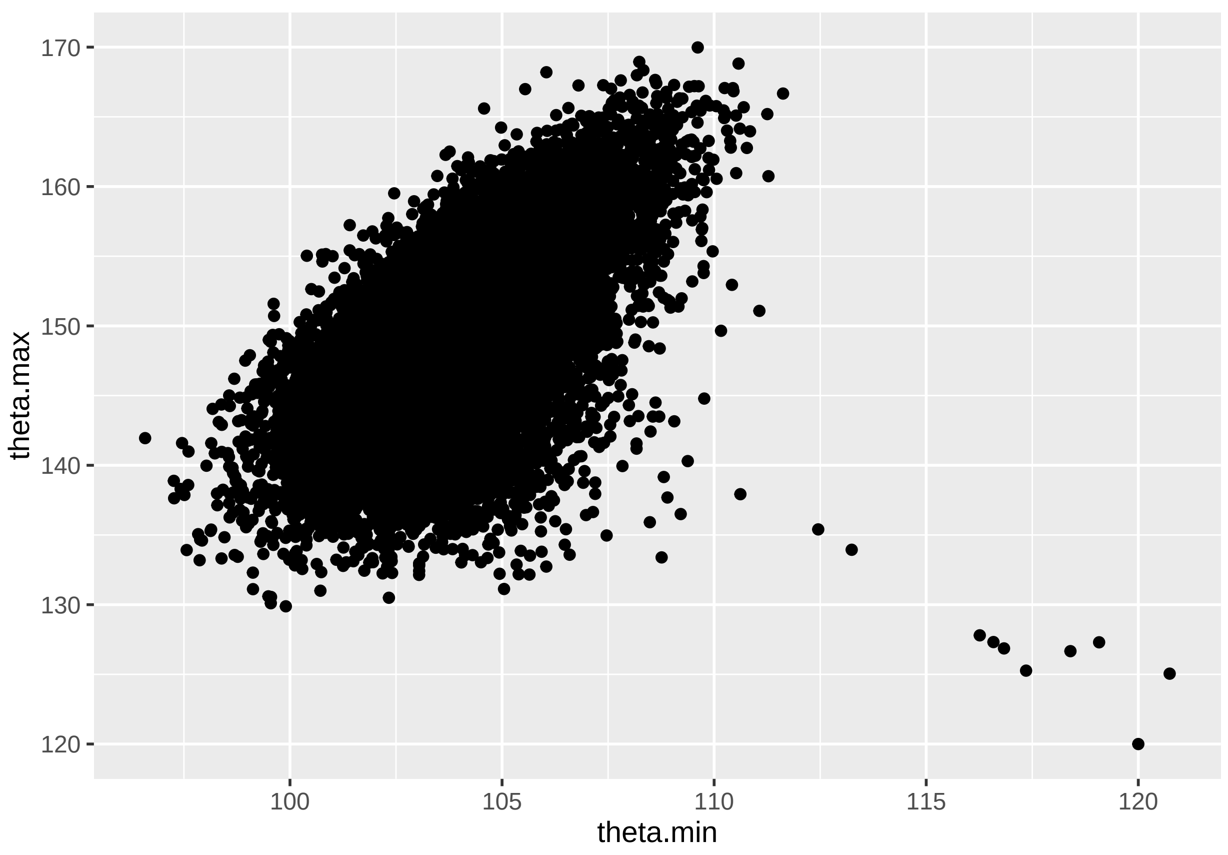
Part B

$$P(X=1|\theta,\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\underline{y}) \sim \frac{p\cdot dnorm(\theta,\mu_1,\sigma_1)}{p\cdot dnorm(\theta,\mu_1,\sigma_1)+(1-p)\cdot dnorm(\theta,\mu_2,\sigma_2)}$$
$$P(X=2|\theta,\mu_1,\mu_2,\sigma_1^2,\sigma_2^2,\underline{y}) \sim \frac{(1-p)\cdot dnorm(\theta,\mu_2,\sigma_2)}{p\cdot dnorm(\theta,\mu_1,\sigma_1)+(1-p)\cdot dnorm(\theta,\mu_2,\sigma_2)}$$
$$p|\theta,\underline{y} \sim \text{beta}(a+n_1,b+n_2)$$
$$\sigma_i^2|\theta,\underline{y} \sim \text{invGamma}(\frac{\nu_{n_i}}{2},\frac{\nu_{n_i}\sigma_{n_i}^2(\theta)}{2}),\nu_{n_i}=\nu_0+n_i,\sigma_{n_i}^2=\frac{1}{\nu_n}[\nu_n\sigma_0^2+ns_n^2(\theta)]$$
$$\theta_i|\sigma_i^2,\underline{y} \sim N(\mu_{n_i},\tau_{n_i}^2),\mu_{n_i}=\frac{\frac{\mu_0}{\tau_o^2}+\frac{n\bar{y}_i}{\sigma_i^2}}{\frac{1}{\tau_0^2}+\frac{n_i}{\sigma_i^2}},\tau_n^2=\frac{1}{\frac{1}{\tau_0^2}+\frac{n_i}{\sigma_i^2}}$$

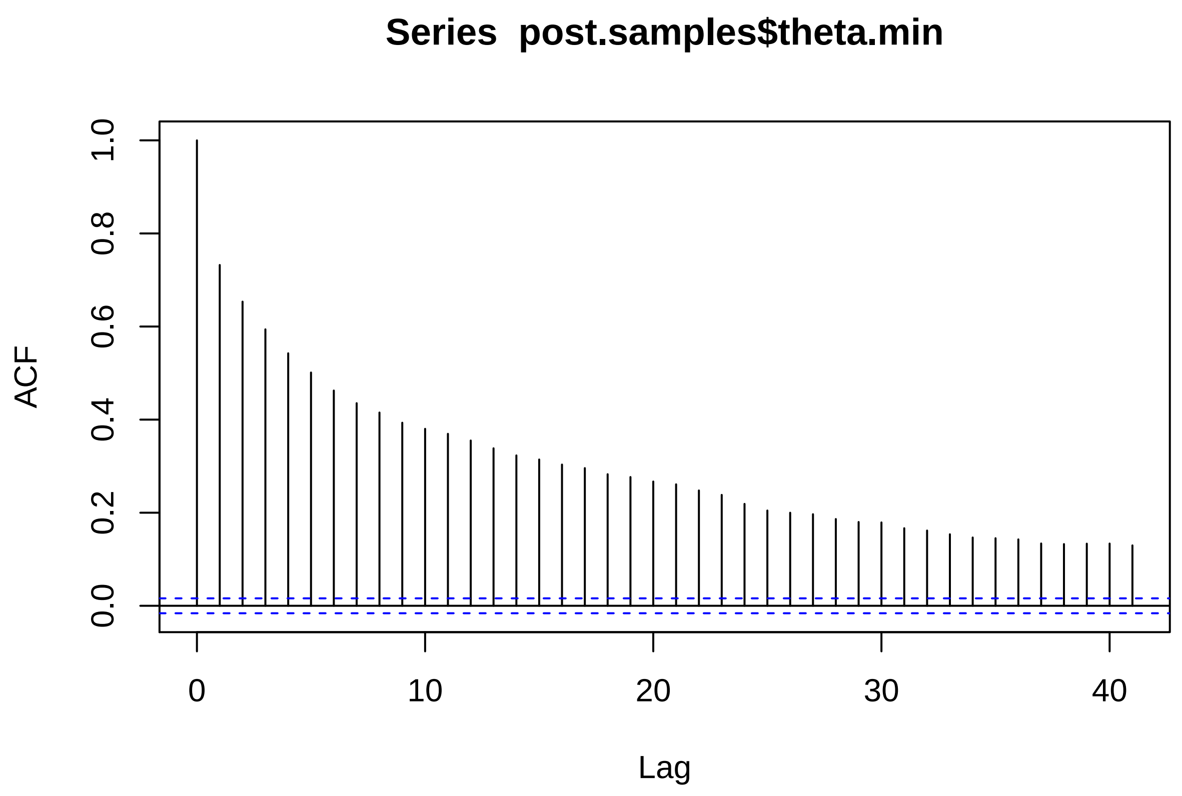
Part C

```
gluc = scan("glucose.dat")

# Hyperparameter values
a = 1; b = 1
```

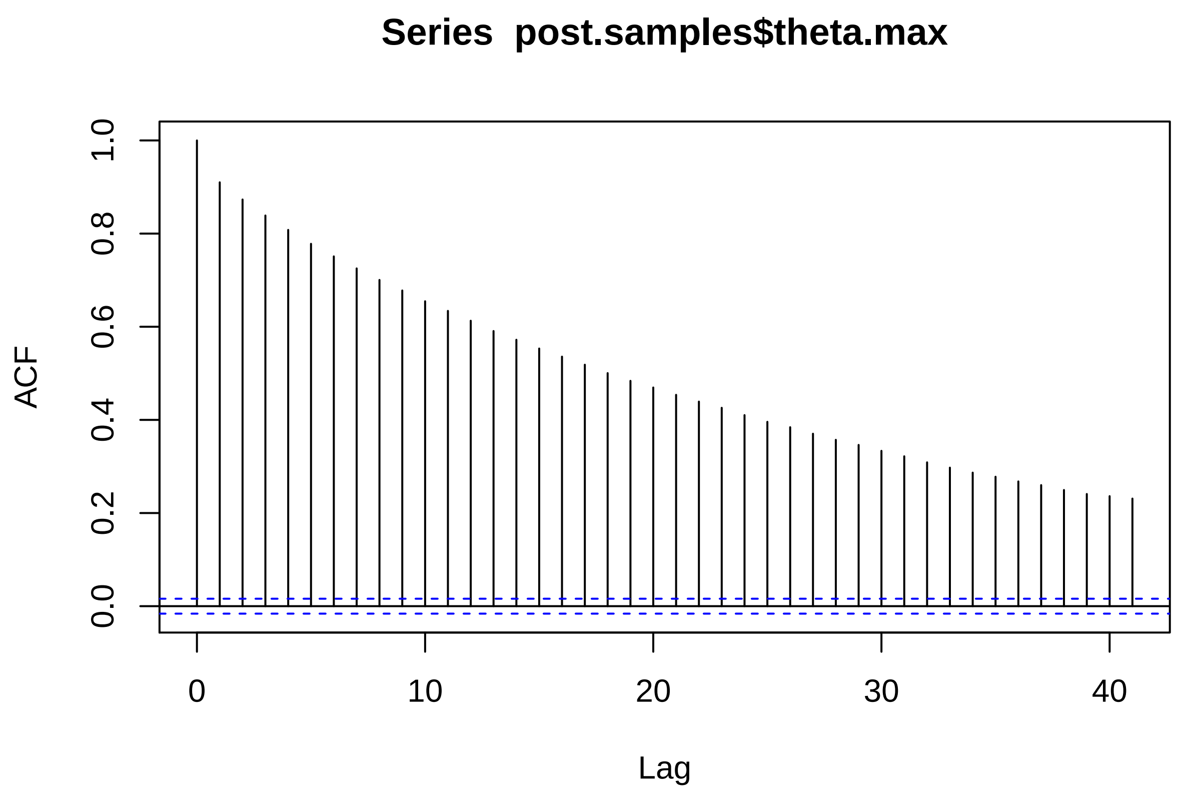
```r
mu0 = 120
tau2.0 = 200
sigma2.0 = 1000
nu0 = 10
n.iter = 15000

x = rep(NA,n.iter)
theta.1 = rep(NA,n.iter)
theta.2 = rep(NA,n.iter)
sigma2.1 = rep(NA,n.iter)
sigma2.2 = rep(NA,n.iter)
p = rep(NA,n.iter)
y.tilde = rep(NA,n.iter)

theta.1[1] = mu0
theta.2[1] = mu0
sigma2.1[1] = sigma2.0
sigma2.2[1] = sigma2.0
p[1] = .5
x[1] = 1
y.tilde[1] = rnorm(1,theta.1[1],sqrt(sigma2.1[1]))

for(i in 2:n.iter){
  fc.p.x.1 = (p[i-1] * dnorm(gluc,theta.1[i-1],sqrt(sigma2.1[i-1]))) /
    (p[i-1] * dnorm(gluc,theta.1[i-1],sqrt(sigma2.1[i-1])) + (1-p[i-1]) * dnorm(gluc,theta.2[i-1],sqrt(
  #Generating actual draws from of x
  x.1 = 2-rbinom(length(fc.p.x.1),1,fc.p.x.1)

  i.1 = (x.1 == 1)
  n.1 = sum(i.1)
  y.1 = gluc[i.1]
  y.bar.1 = mean(y.1)
  s1 = var(y.1)

  i.2 = (x.1 == 2)
  n.2 = sum(i.2)
  y.2 = gluc[i.2]
  y.bar.2 = mean(y.2)
  s2 = var(y.2)
  # Posterior for p
  p[i] = rbeta(1,a+n.1,b+n.2)

  # From theta
  tau2.n.1 = 1 /((1/tau2.0) + n.1/sigma2.1[i-1])
  mu.n.1 = tau2.n.1 * ((mu0/tau2.0) + (n.1*y.bar.1) / sigma2.1[i-1])
  theta.1[i] = rnorm(1,mu.n.1,sqrt(tau2.n.1))

  tau2.n.2 = 1 /((1/tau2.0) + n.2/sigma2.2[i-1])
  mu.n.2 = tau2.n.2 * ((mu0/tau2.0) + (n.2*y.bar.2) / sigma2.2[i-1])
  theta.2[i] = rnorm(1,mu.n.2,sqrt(tau2.n.2))

  # From sigma
  nu.n.1 = nu0 + n.1
  sigma2.n.1 = (1/nu.n.1)*(nu0 * sigma2.0 + (n.1 - 1) * s1 + n.1*(y.bar.1 - theta.1[i])^2)
  sigma2.1[i] = 1 / rgamma(1,nu.n.1/2,nu.n.1*sigma2.n.1 / 2)

  nu.n.2 = nu0 + n.2
  sigma2.n.2 = (1/nu.n.2)*(nu0 * sigma2.0 + (n.2 - 1) * s2 + n.2*(y.bar.2 - theta.2[i])^2)
  sigma2.2[i] = 1 / rgamma(1,nu.n.2/2,nu.n.2*sigma2.n.2 / 2)

  x[i] = 2-rbinom(1,1,p[i])
  if(x[i] == 1){
    y.tilde[i] = rnorm(1,theta.1[i],sqrt(sigma2.1[i]))
  }else{
    y.tilde[i] = rnorm(1,theta.2[i],sqrt(sigma2.2[i]))
  }
}

post.samples = cbind(x,p,theta.1,theta.2,sigma2.1,sigma2.2,y.tilde)
colnames(post.samples) = c("x","p","theta.1","theta.2","sigma2.1","sigma2.2","y.tilde")

post.samples = as.data.frame(post.samples)

post.samples$theta.min = NA
post.samples$theta.max = NA
for(i in 1:n.iter){
  post.samples$theta.min[i] = min(post.samples$theta.1[i],post.samples$theta.2[i])
  post.samples$theta.max[i] = max(post.samples$theta.1[i],post.samples$theta.2[i])
}

ggplot(data = post.samples, aes(x = theta.min,y = theta.max)) + geom_point()
```



```r
acf(post.samples$theta.min)
```



```r
acf(post.samples$theta.max)
```

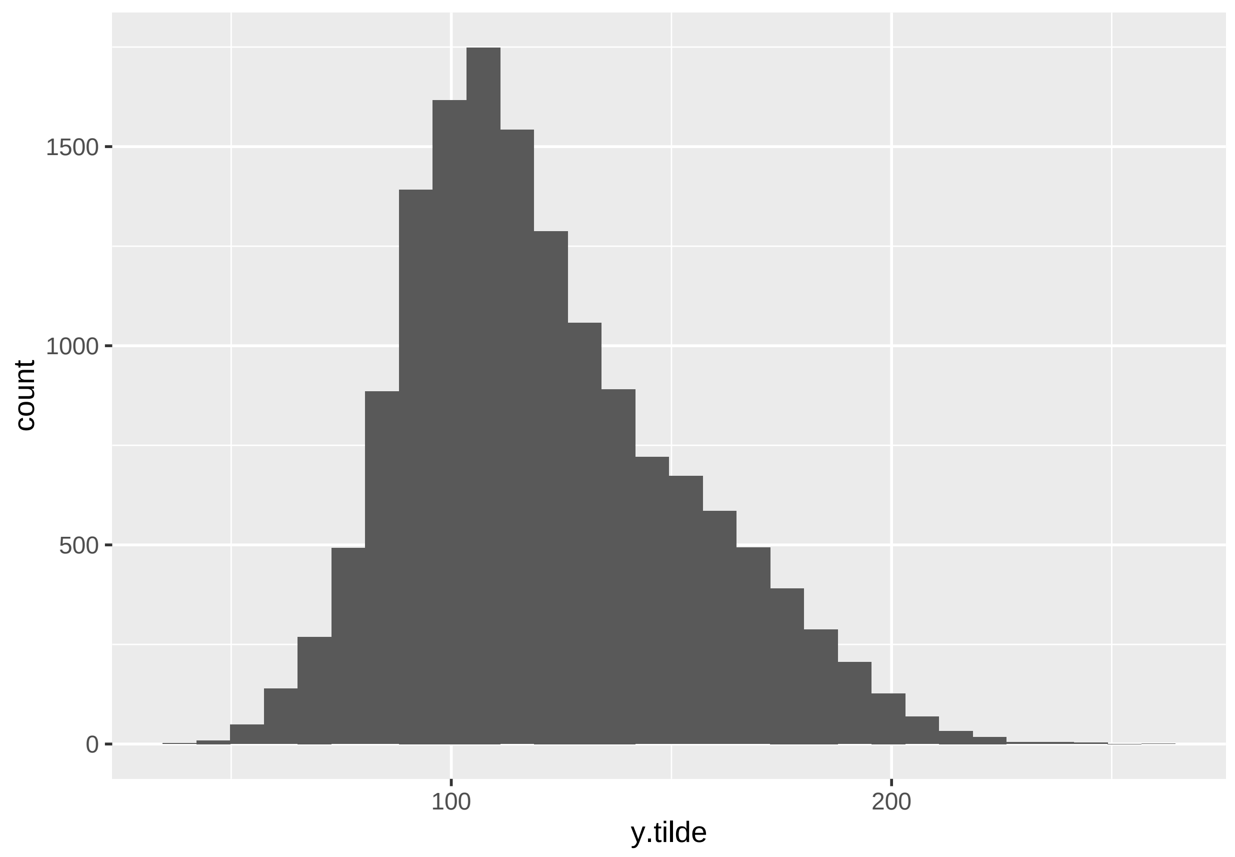


```
eff.theta.min = effectiveSize(post.samples$theta.min)
eff.theta.max = effectiveSize(post.samples$theta.max)
```

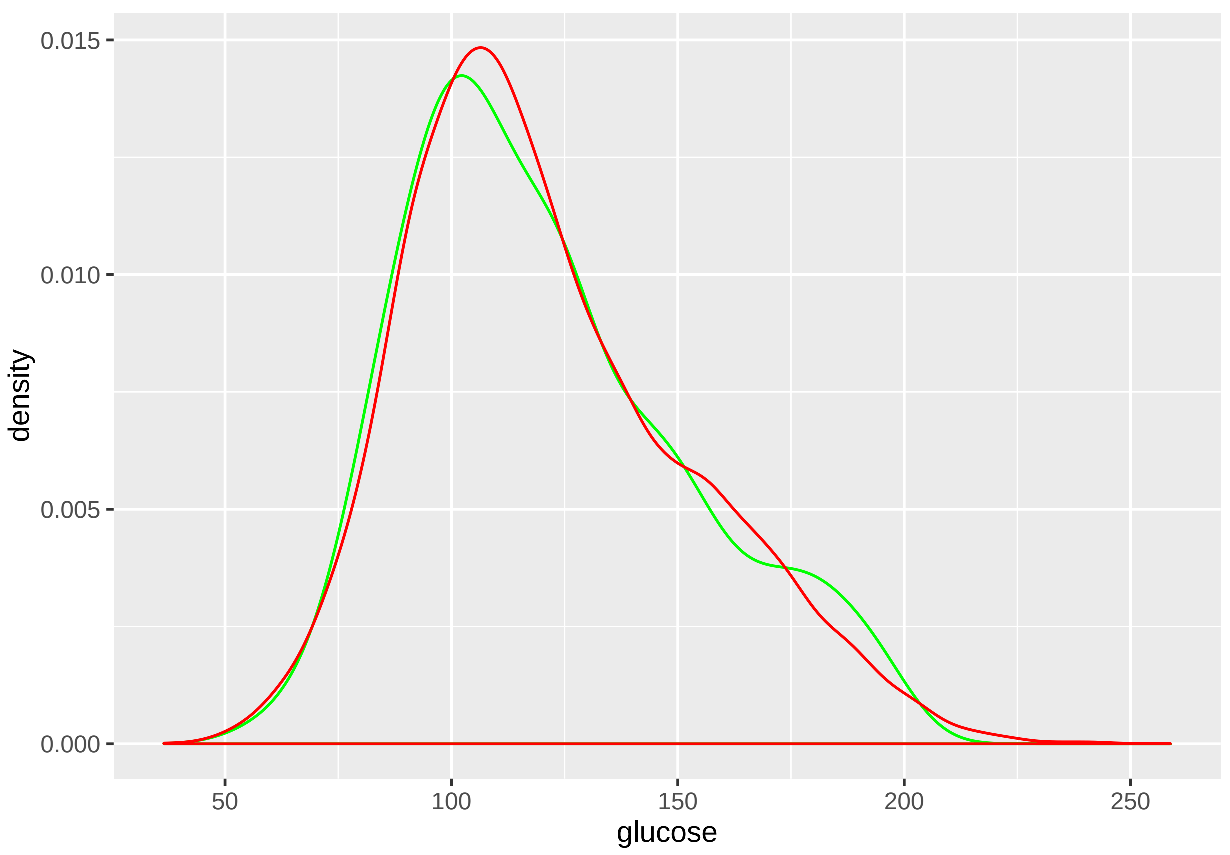
The question asks for at least 10,000 iterations, and I decided to do 15,000 iterations. With my 15000 iterations, I got the effective sample size of the following $\theta_min = 496.49794$ and $\theta_max = 296.5512699$. This certainly isn't idea because it means that these represent the number of independent samples that our posterior draws represent. However, something of note is although the auto-correlation isn't perfect, it does die down relatively quickly, but it's clear that the posterior samples are still dependent.

```
ggplot(data = post.samples, aes(x = y.tilde)) + geom_histogram()
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
ggplot() +
geom_density(data=as.data.frame(glucose), aes(x=glucose), color='green') +
  geom_density(data=post.samples, aes(x= y.tilde), color='red')
```



In this plot, the plot of the data is the green line, while the plot of the posterior predictive is red. It mirrors the distribution well.