

Monte Carlo Sampling

Ryan G. McClarren

For a QoI we can sample each of the parameters from their given distribution and compute the QoI for each sampled point in parameter space. This gives us samples from the output distribution of the QOI.

For a simple example suppose you need to construct a 5 cm sphere of Uranium-235. For safety purposes, it cannot be within 3 cm of the critical radius. Given uncertainties in material properties, what is the distribution of masses and spheres?, Are they safe?

Here are the formulae for the mass and critical radius:

$$\text{Mass} = \frac{4}{3}\pi r^3 \rho,$$
$$R_c = \frac{2.2698}{N_p \sigma_t} \sqrt{\frac{\sigma_t}{3\sigma_f(\nu - 1)}},$$
$$N_p = \frac{\rho}{235 \times 1.660538782 \times 10^{-24}}.$$

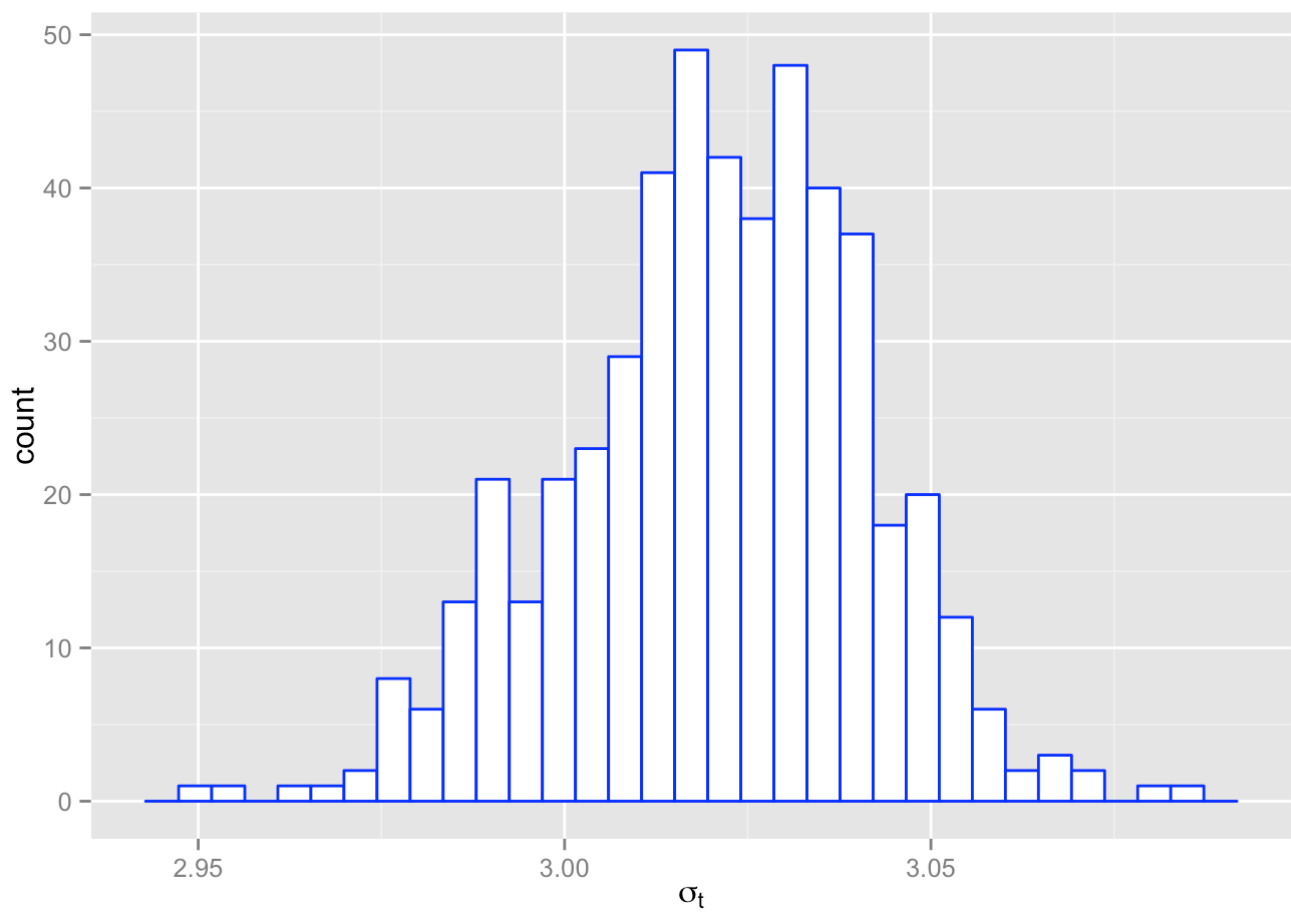
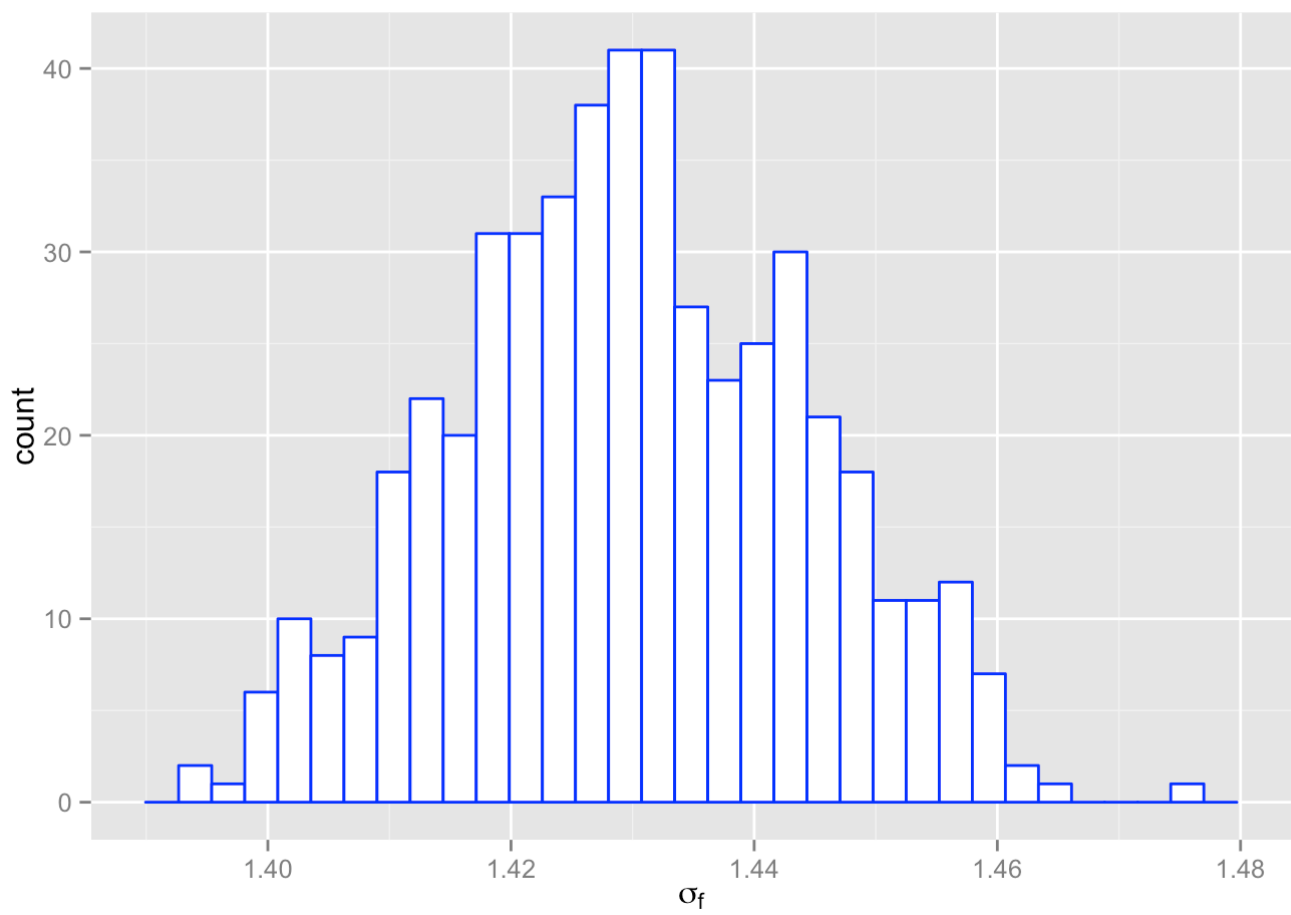
```
r = 5
massSphere <- function(rho) 4/3*pi*r^3 * rho
critRadius <- function(rho, sigf, sigt, nu) {
  Np = rho /235 / 1.660538782e-24
  2.2698/ (Np * sigt * 10^-24) *sqrt(sigt/(3 * sigf * (nu-1)))
}
```

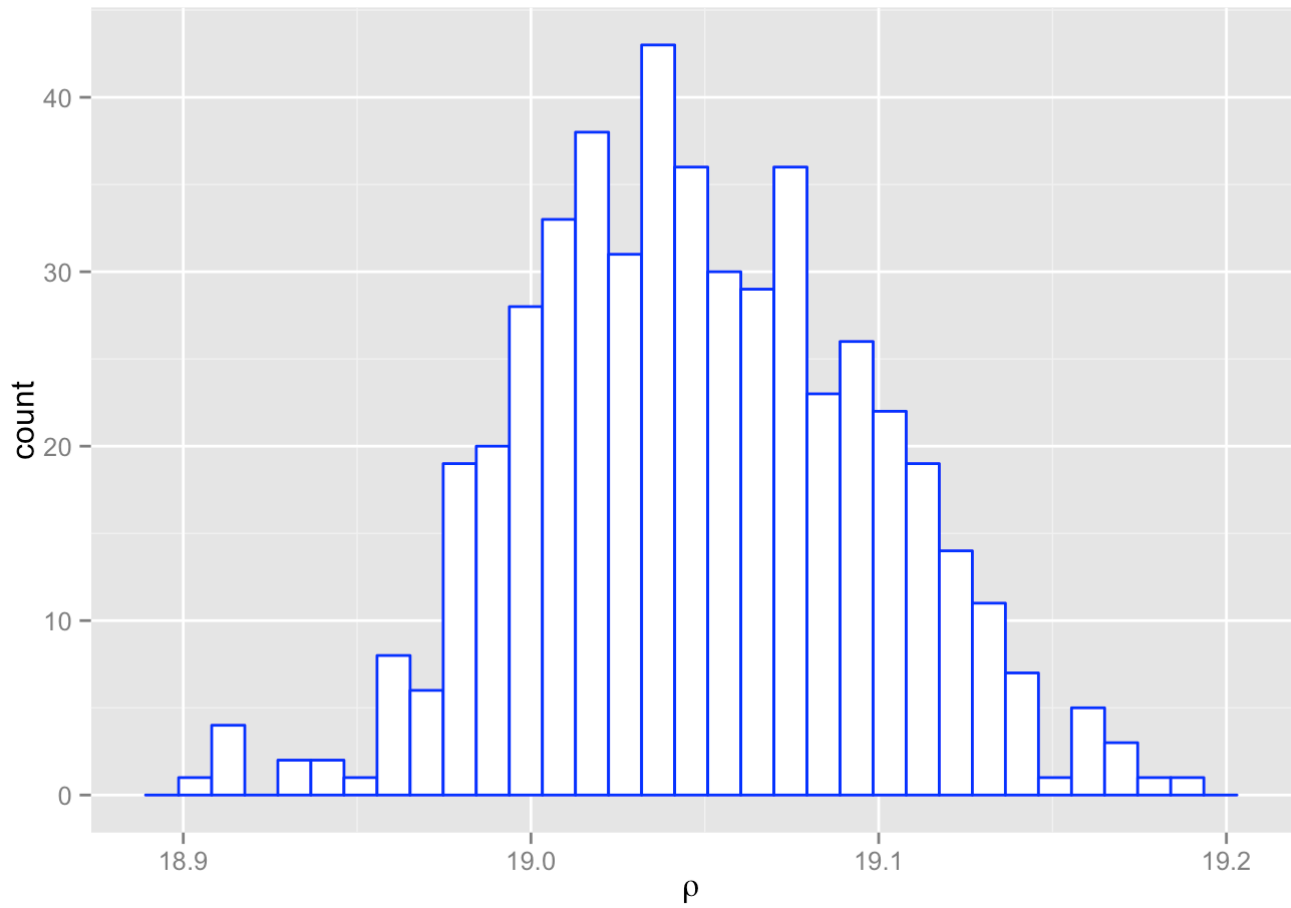
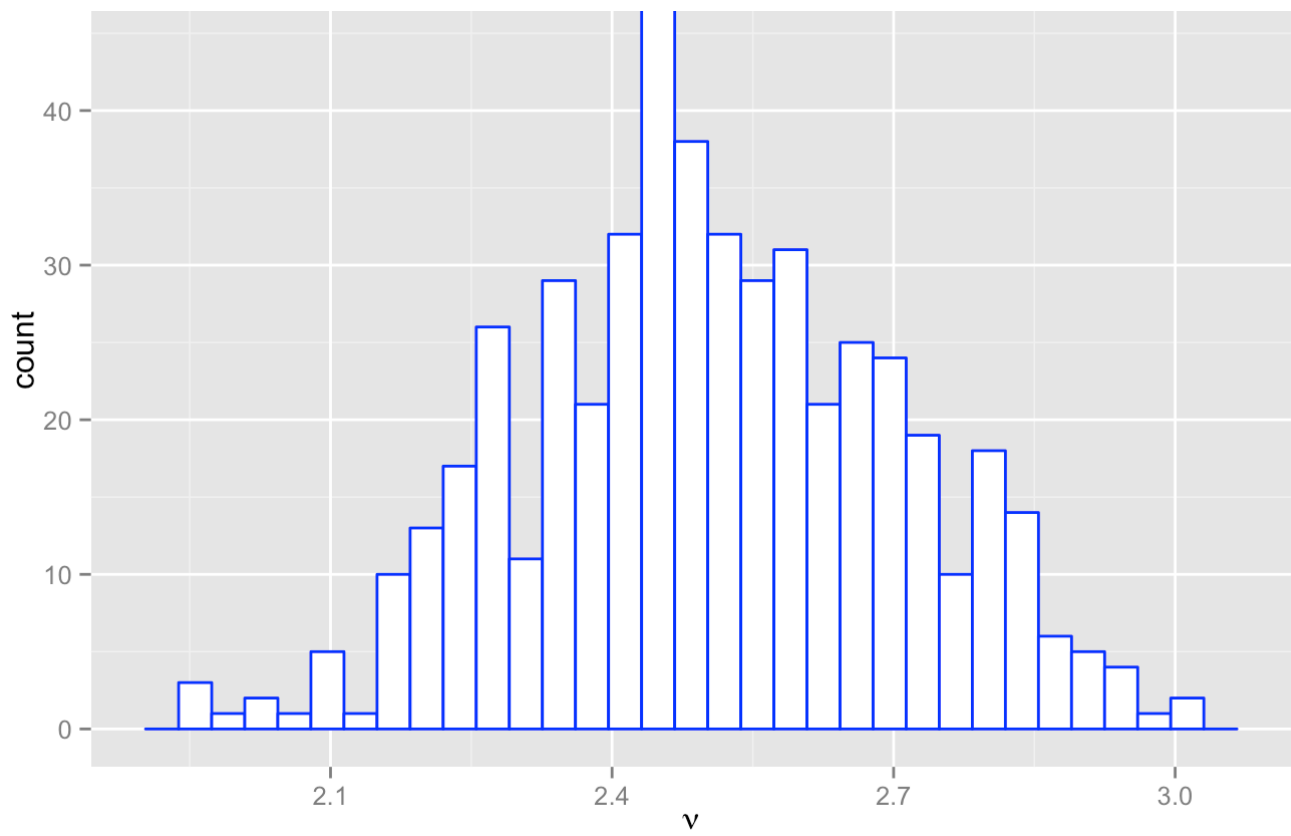
Let's generate samples for each uncertain parameter, assuming each is normally distributed.

```
NumSamps = 500
meanSigf = 1.43; sdSigf = 0.015
meanSigt = 3.02; sdSigt = 0.02
meannu = 2.5; sdnu = 0.2
meanrho = 19.05; sdrho = 0.05

samples <- data.frame(sigf = rnorm(NumSamps,meanSigf,sdSigf),
                      sigt = rnorm(NumSamps,meanSigt,sdSigt),
                      nu = rnorm(NumSamps,meannu,sdnu),
                      rho = rnorm(NumSamps,meanrho,sdrho))
samples$mass <- massSphere(samples$rho)
samples$critRadius <- with(samples,critRadius(rho,sigf,sigt,nu))
```

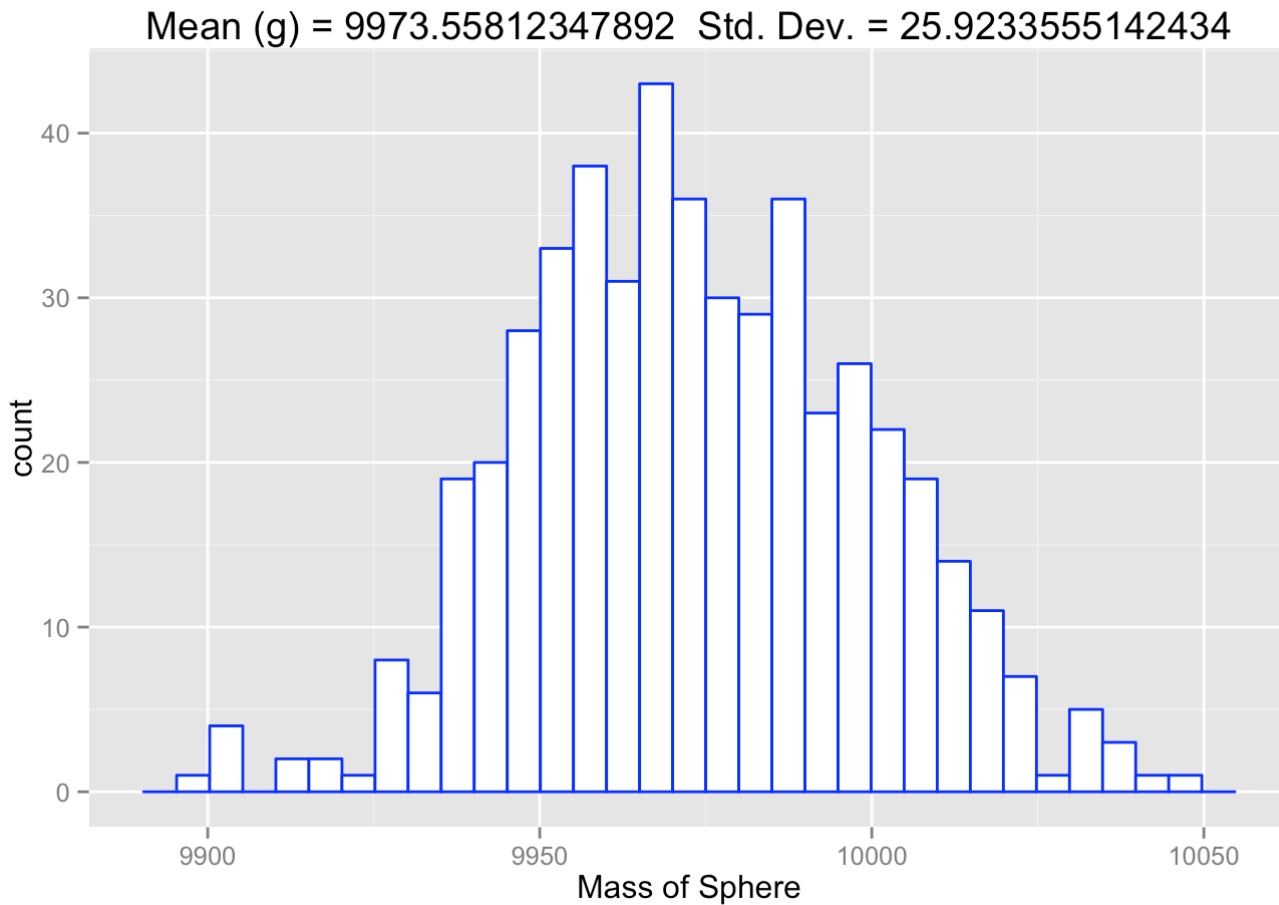
Here is a plot of each variable's histogram





For the mass of the sphere we have the following distribution:

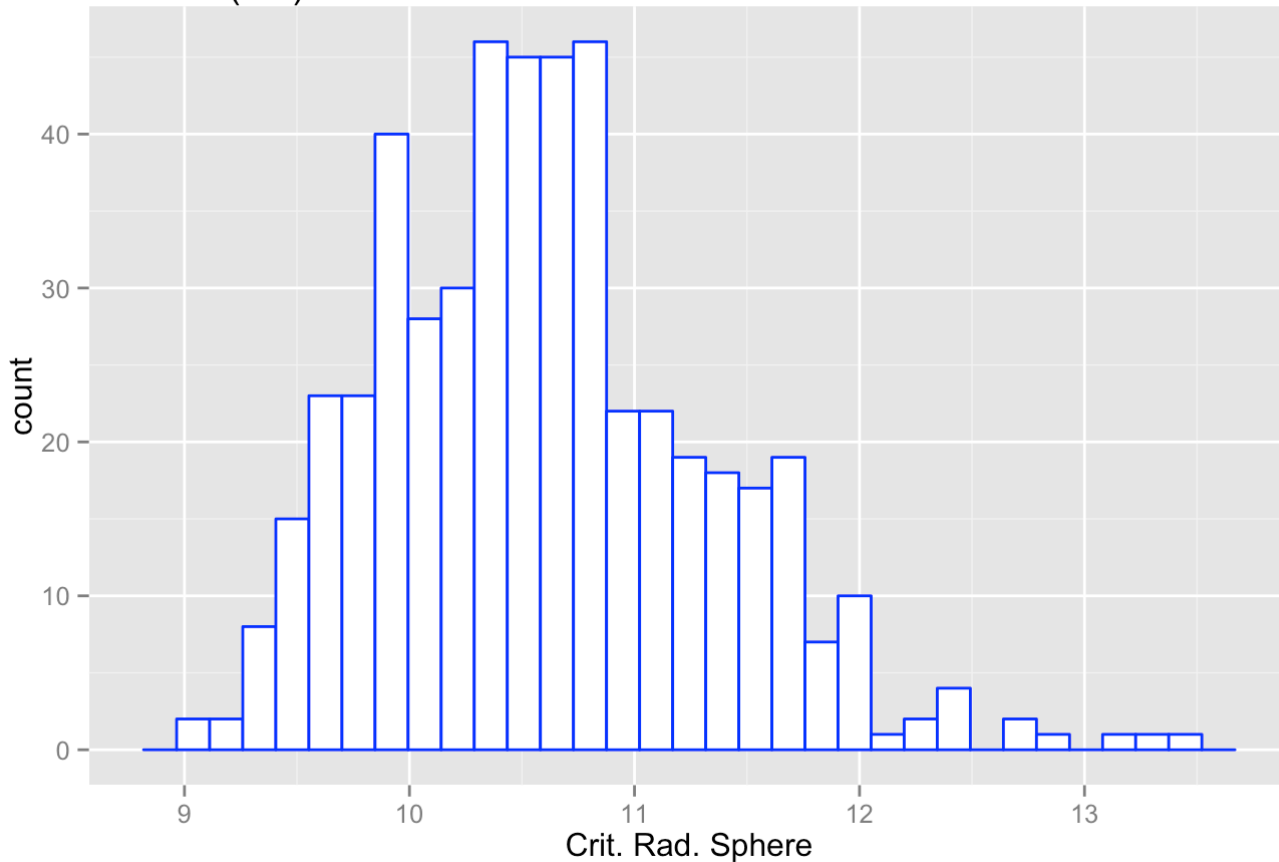
```
ggplot(samples, aes(x=mass)) + geom_histogram(fill="white",color="blue", binwidth = (max(samples$mass)-min(samples$mass))/30) + scale_x_continuous("Mass of Sphere") + ggtitle(paste("Mean (g) =",mean(samples$mass)," Std. Dev. =",sd(samples$mass)))
```



The critical radius of a sphere made out of the material has a distribution given by

```
ggplot(samples, aes(x=critRadius)) + geom_histogram(fill="white",color="blue", binwidth = (max(samples$critRadius)-min(samples$critRadius))/30) + scale_x_continuous("Crit. Rad. Sphere") + ggtitle(paste("Mean (cm) =",mean(samples$critRadius)," Std. Dev. =",sd(samples$critRadius)))
```

Mean (cm) = 10.6011475313925 Std. Dev. = 0.725252295962339



The probability, based on these samples, of the critical radius being 8 or less is 0. The minimum critical radius is 9.0113.

Cauchy RV

Let's change our distribution of random variables to a Laplace (double Exponential) distribution. This has the PDF

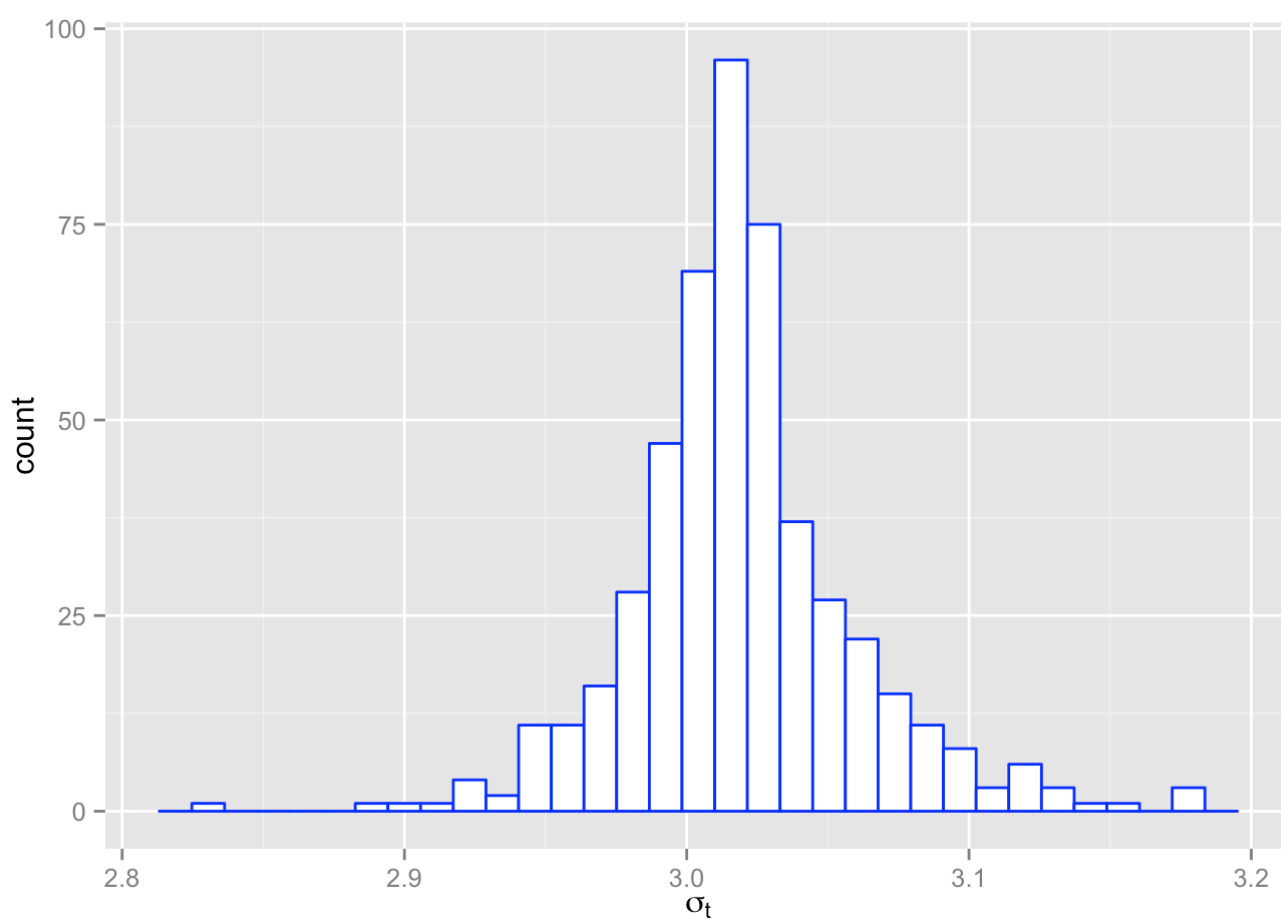
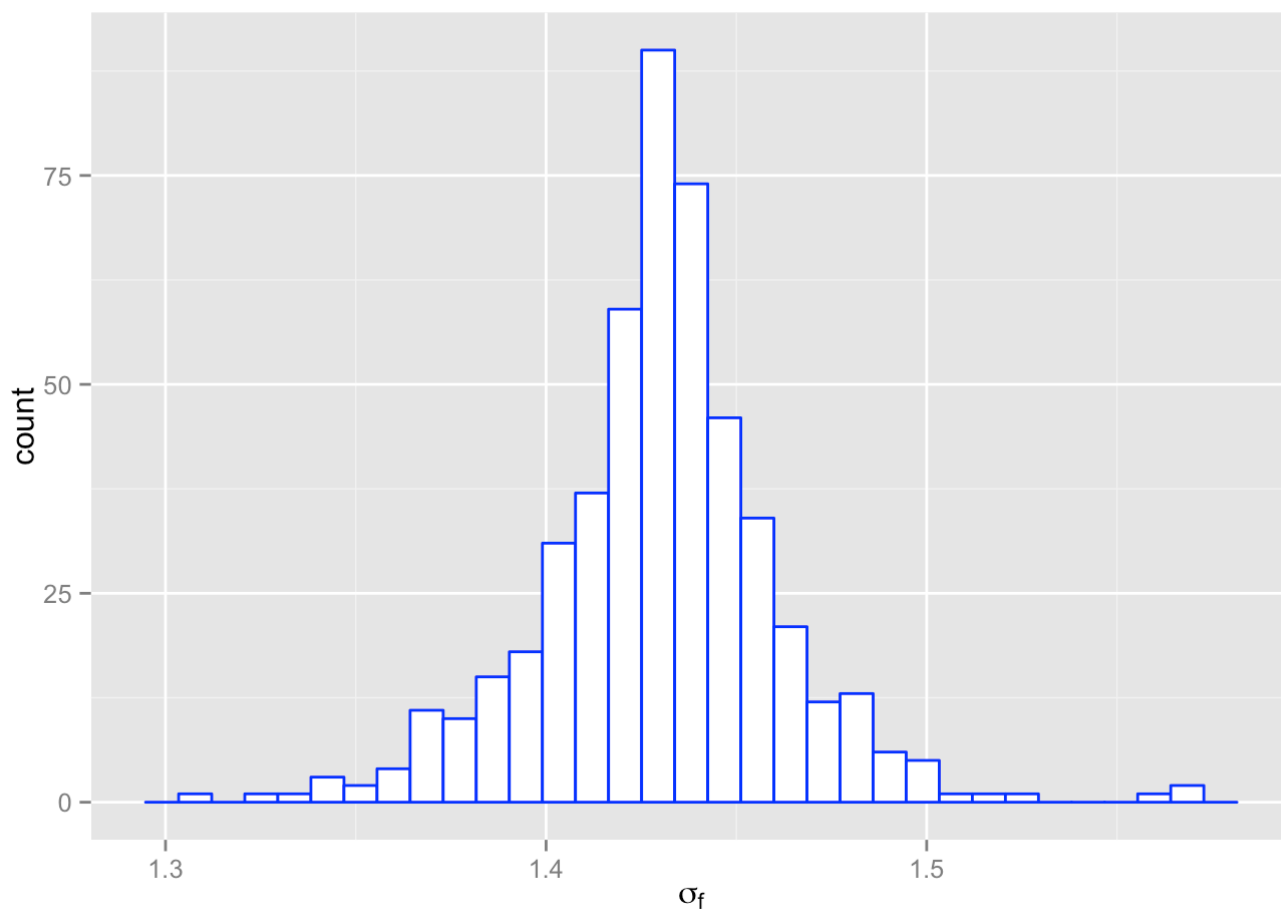
$$f(x|\theta, \lambda) = \frac{1}{2\lambda} \exp\left(-\frac{|x - \theta|}{\lambda}\right).$$

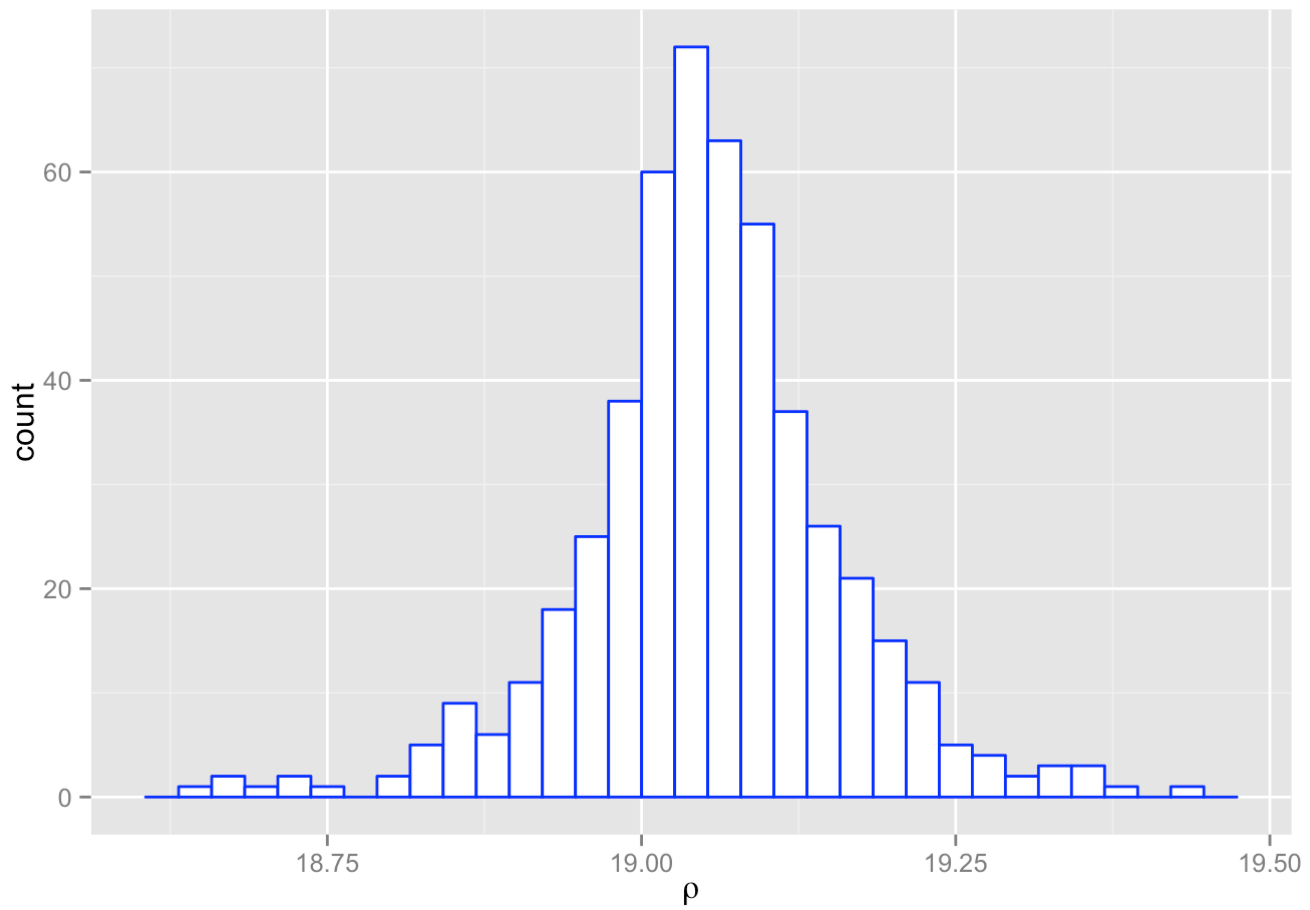
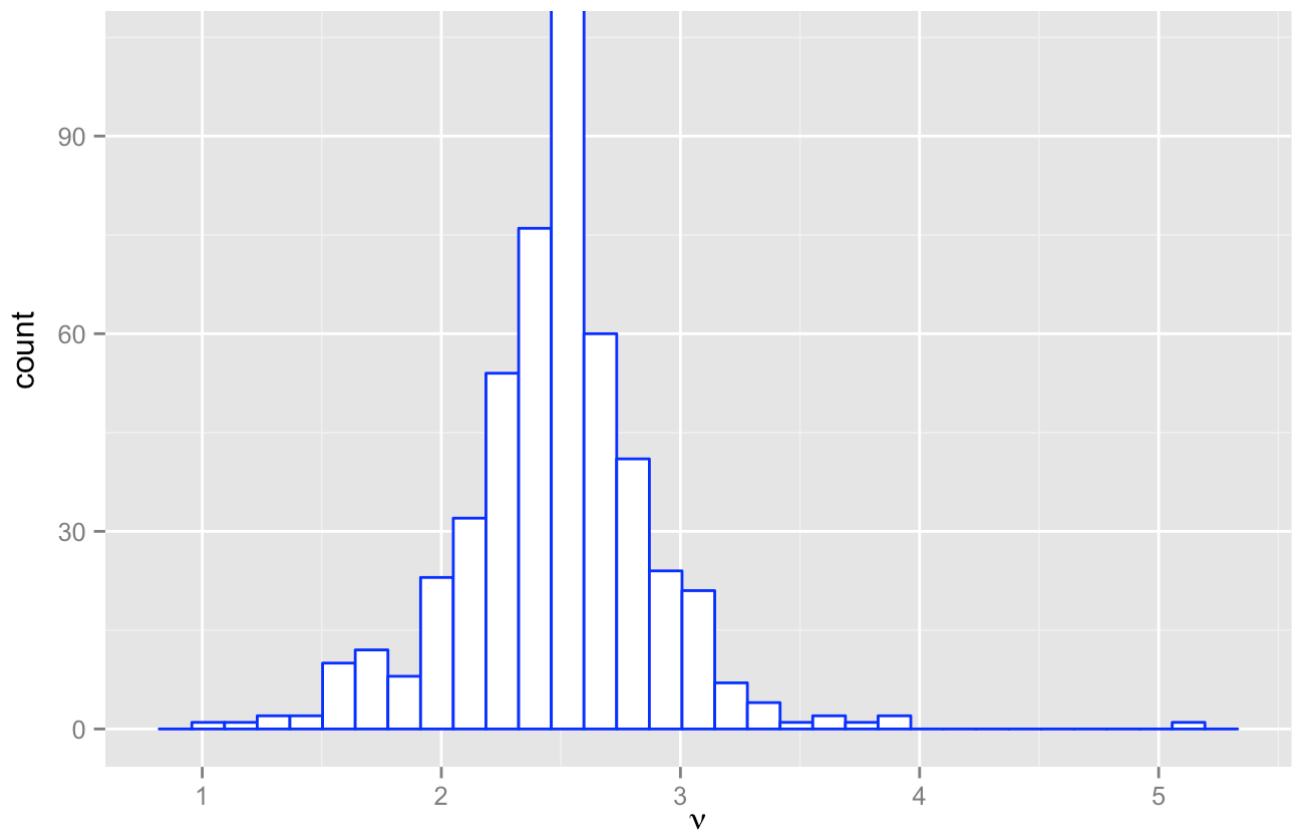
With a mean equal to θ and standard deviation of $\sqrt{2}\lambda$. Therefore, we can change our samples accordingly. Note we have to make sure nu stays larger than 1.

```
NumSamps = 500
meanSigf = 1.43; sdSigf = 0.015
meanSigt = 3.02; sdSigt = 0.02
meannu = 2.5; sdnu = 0.2
meanrho = 19.05; sdrho = 0.05

samples <- data.frame(sigf = rdouplex(NumSamps,meanSigf,sqrt(2)*sdSigf),
                      sigt = rdouplex(NumSamps,meanSigt,sqrt(2)*sdSigt),
                      nu = rdouplex(NumSamps,meannu,sqrt(2)*sdnu),
                      rho = rdouplex(NumSamps,meanrho,sqrt(2)*sdrho))
samples$nu[ samples$nu<=1] <- 1.00001
samples$mass <- massSphere(samples$rho)
samples$critRadius <- with(samples,critRadius(rho,sigf,sigt,nu))
```

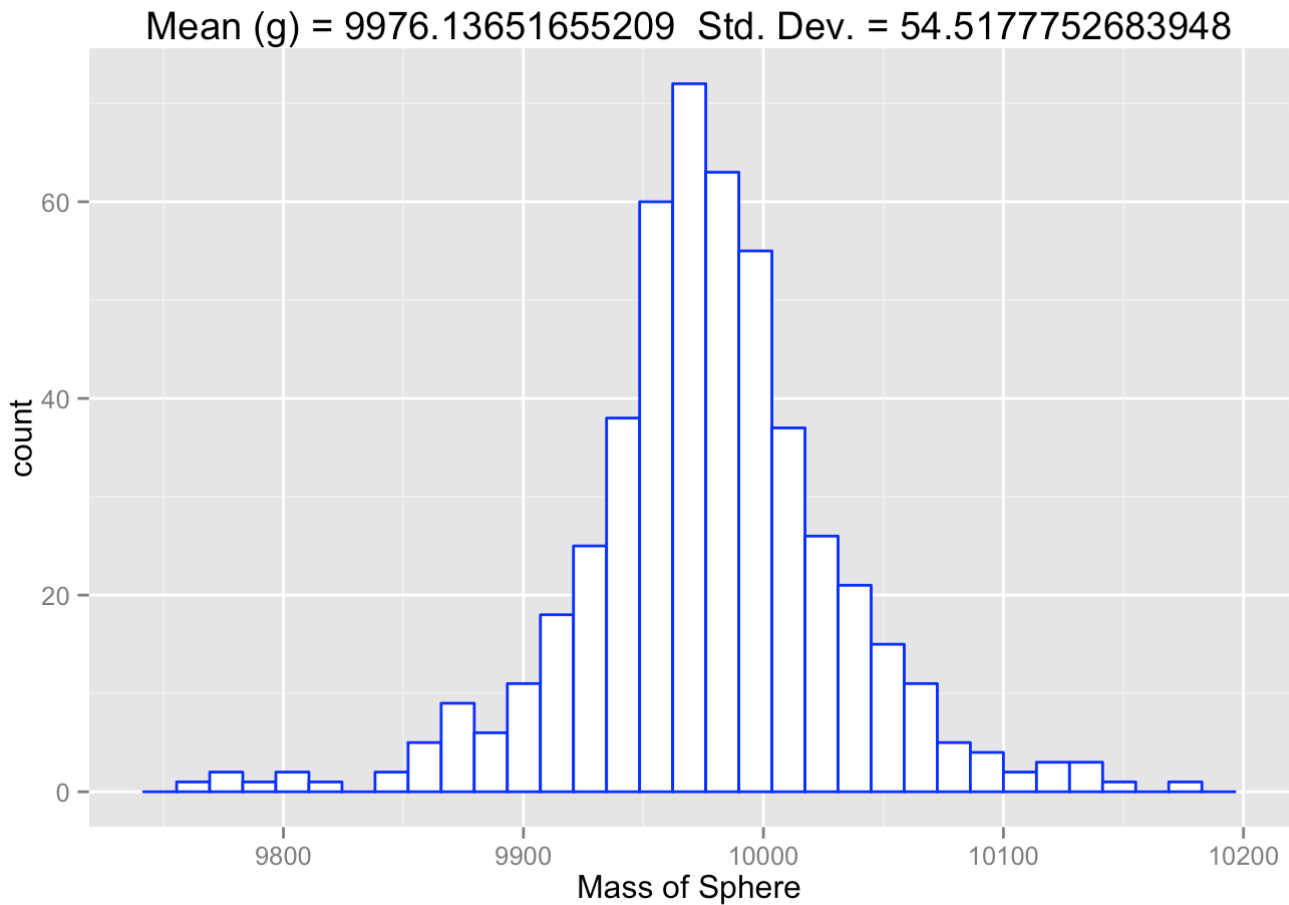
Here is a plot of each variable's histogram





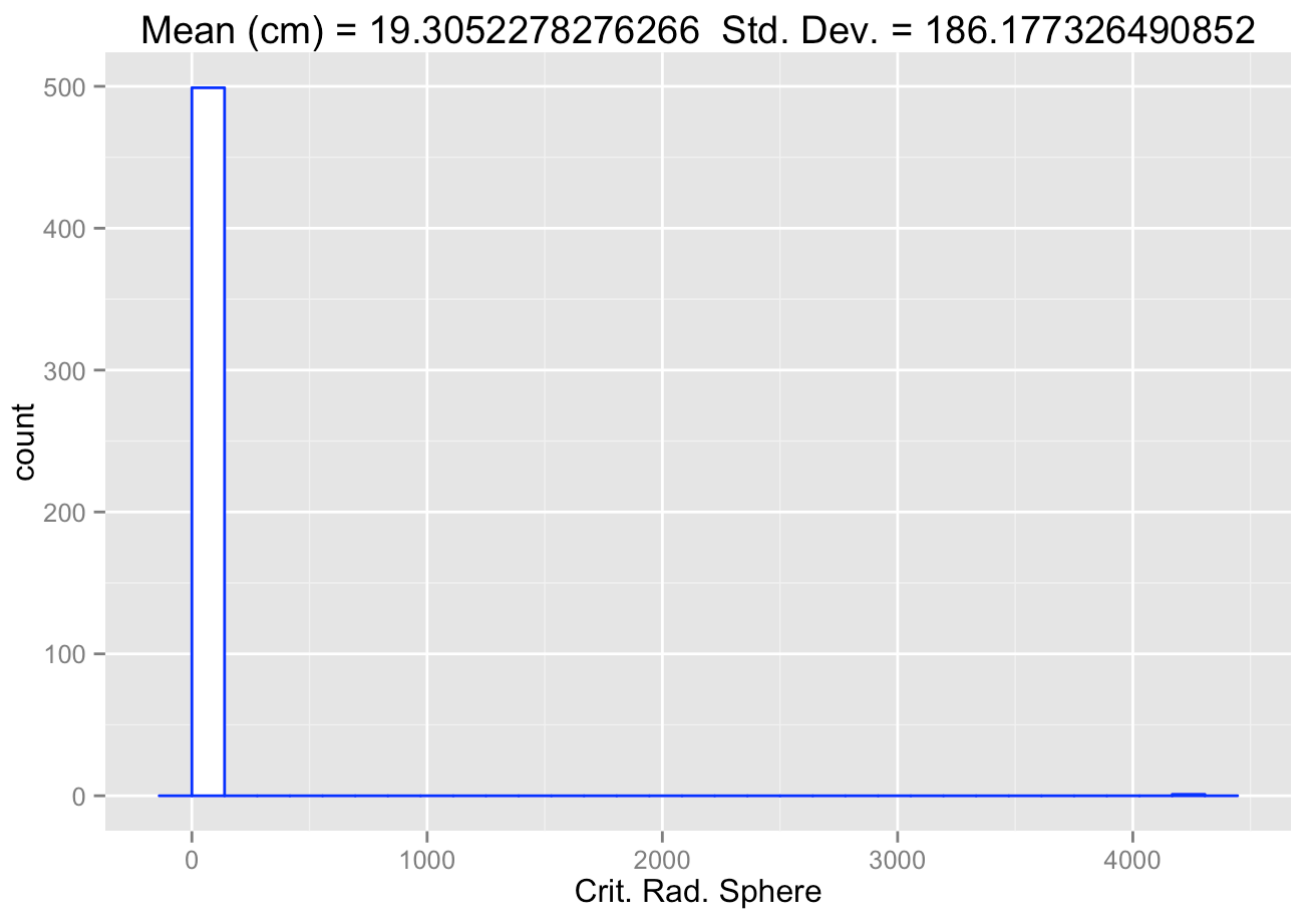
For the mass of the sphere we have the following distribution:

```
ggplot(samples, aes(x=mass)) + geom_histogram(fill="white",color="blue", binwidth = (max(samples$mass)-min(samples$mass))/30) + scale_x_continuous("Mass of Sphere") + ggtitle(paste("Mean (g) =",mean(samples$mass)," Std. Dev. =",sd(samples$mass)))
```



The critical radius of a sphere made out of the material has a distribution given by

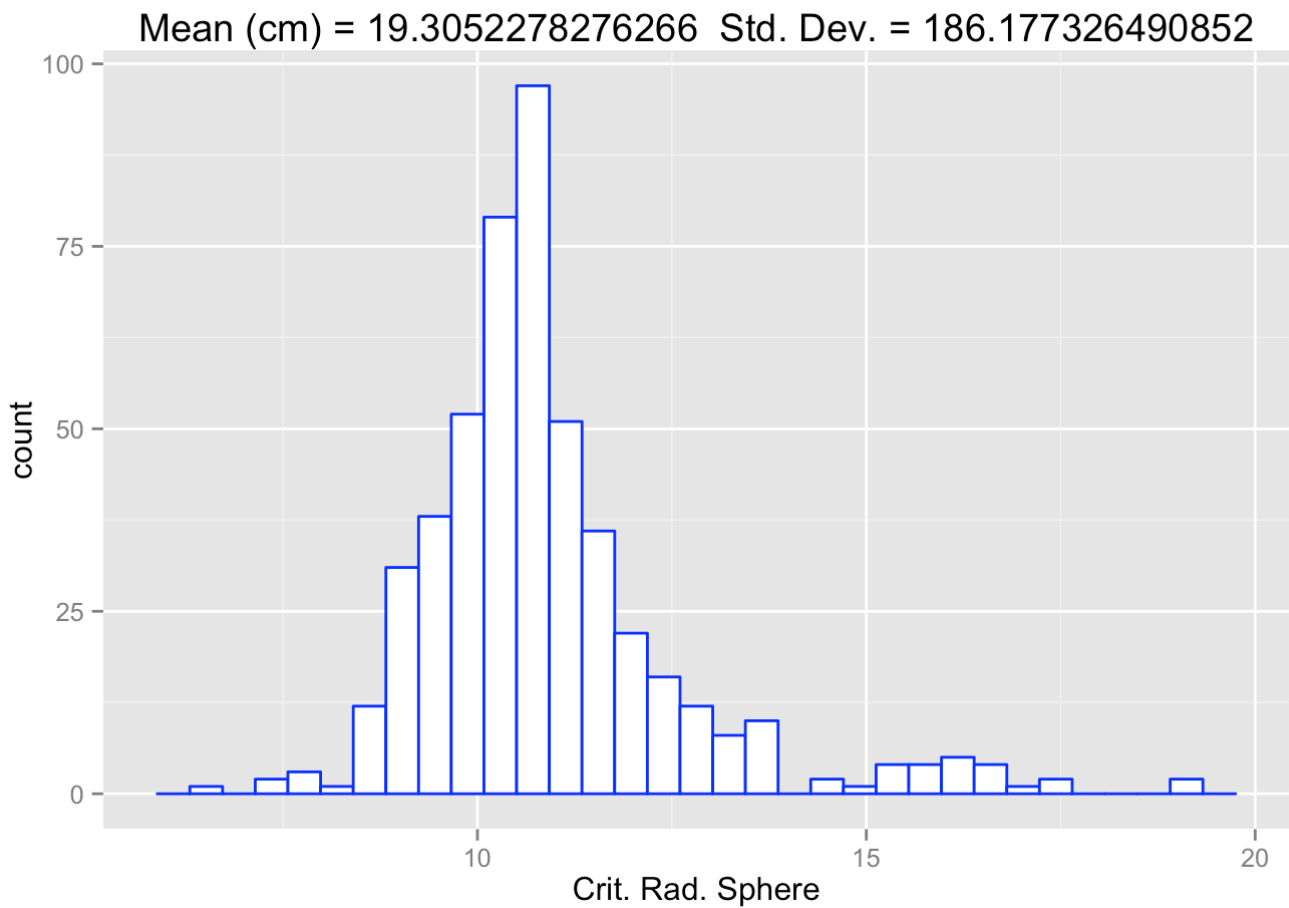
```
ggplot(samples, aes(x=critRadius)) + geom_histogram(fill="white",color="blue", binwidth = (max(samples$critRadius)-min(samples$critRadius))/30) + scale_x_continuous("Crit. Rad. Sphere") + ggtitle(paste("Mean (cm) =",mean(samples$critRadius)," Std. Dev. =",sd(samples$critRadius)))
```



Let's zoom in

```
ggplot(samples[samples$critRadius<20,], aes(x=critRadius)) + geom_histogram(fill="white",color="blue") + scale_x_continuous("Crit. Rad. Sphere") + ggtitle(paste("Mean (cm) =",mean(samples$critRadius)," Std. Dev. =",sd(samples$critRadius)))
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



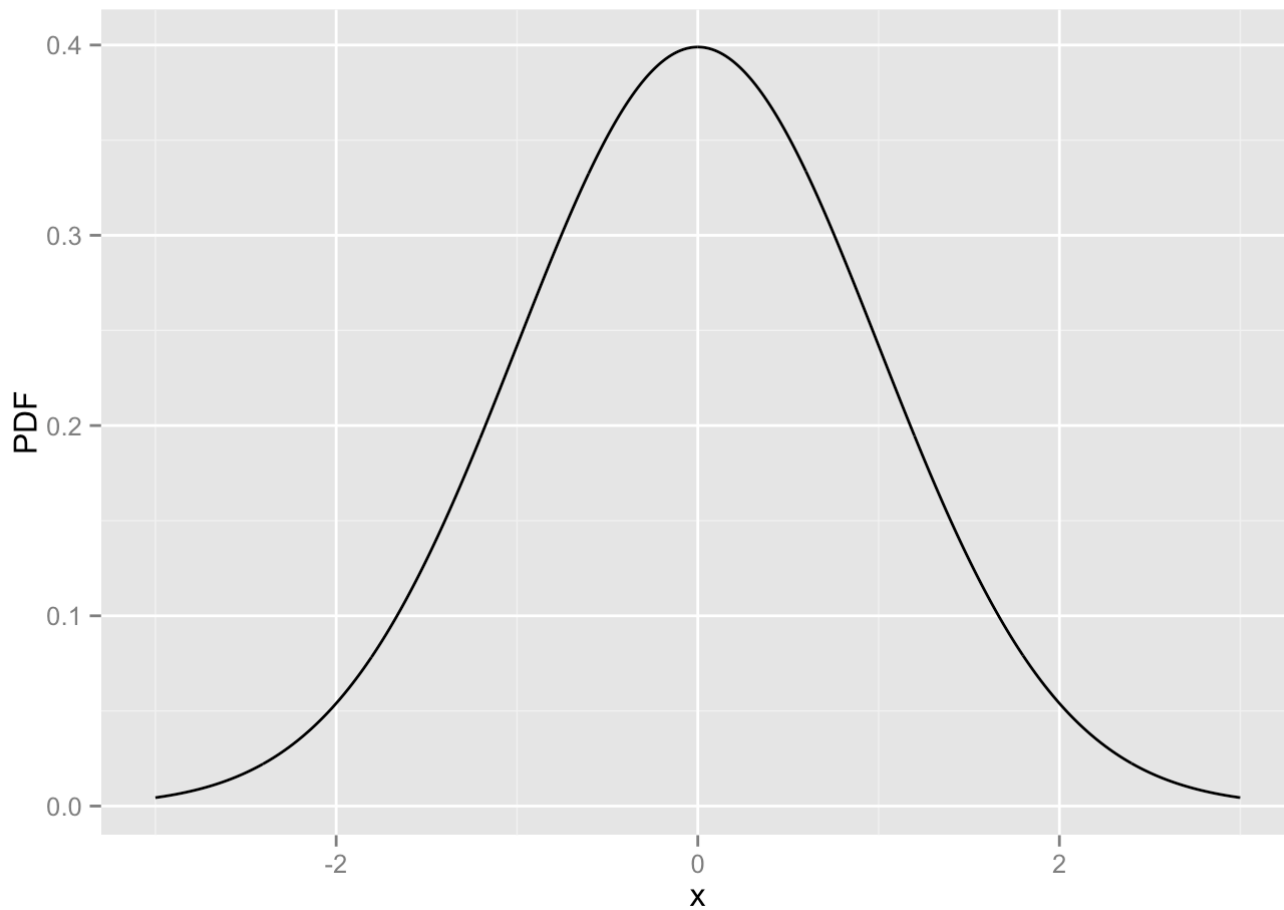
The probability, based on these samples, of the critical radius being 8 or less is 0.012. The minimum critical radius is 6.4603.

Stratified Sampling

It takes a lot of points to get samples in the “tails” of a distribution. To force our samples to spread out we can use stratified sampling. That is we can divide our PDF into equal-probable bins and then pick a bin randomly, and a point inside the bin.

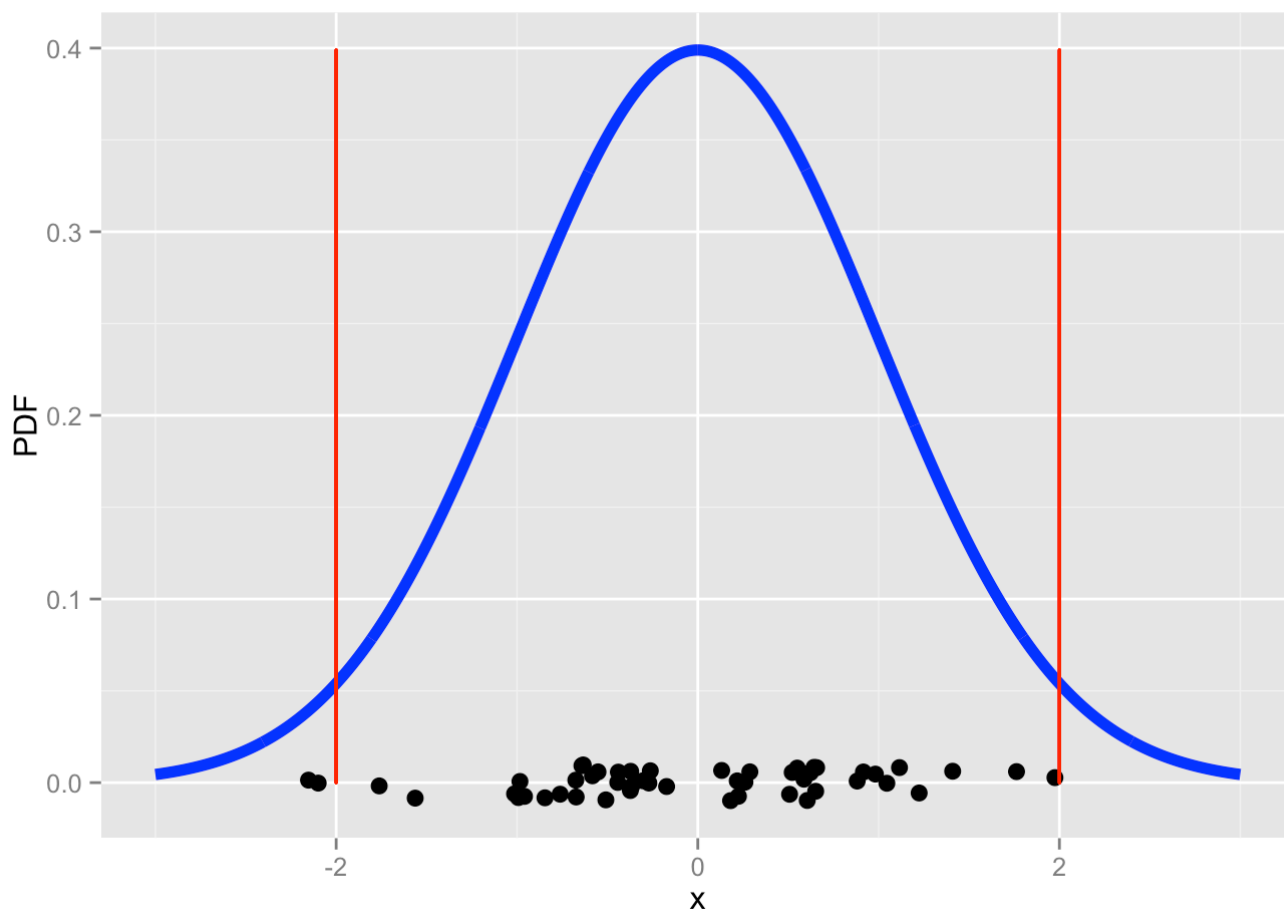
Let's do an example with a standard normal. The PDF looks like:

```
plotDF <- data.frame(x=seq(-3,3,length.out=1000))
plotDF$PDF <- dnorm(plotDF$x)
ggplot(plotDF,aes(x=x, y=PDF)) + geom_line()
```



If we generate points from this distribution randomly we will see that we don't get too many points beyond 2 standard deviations:

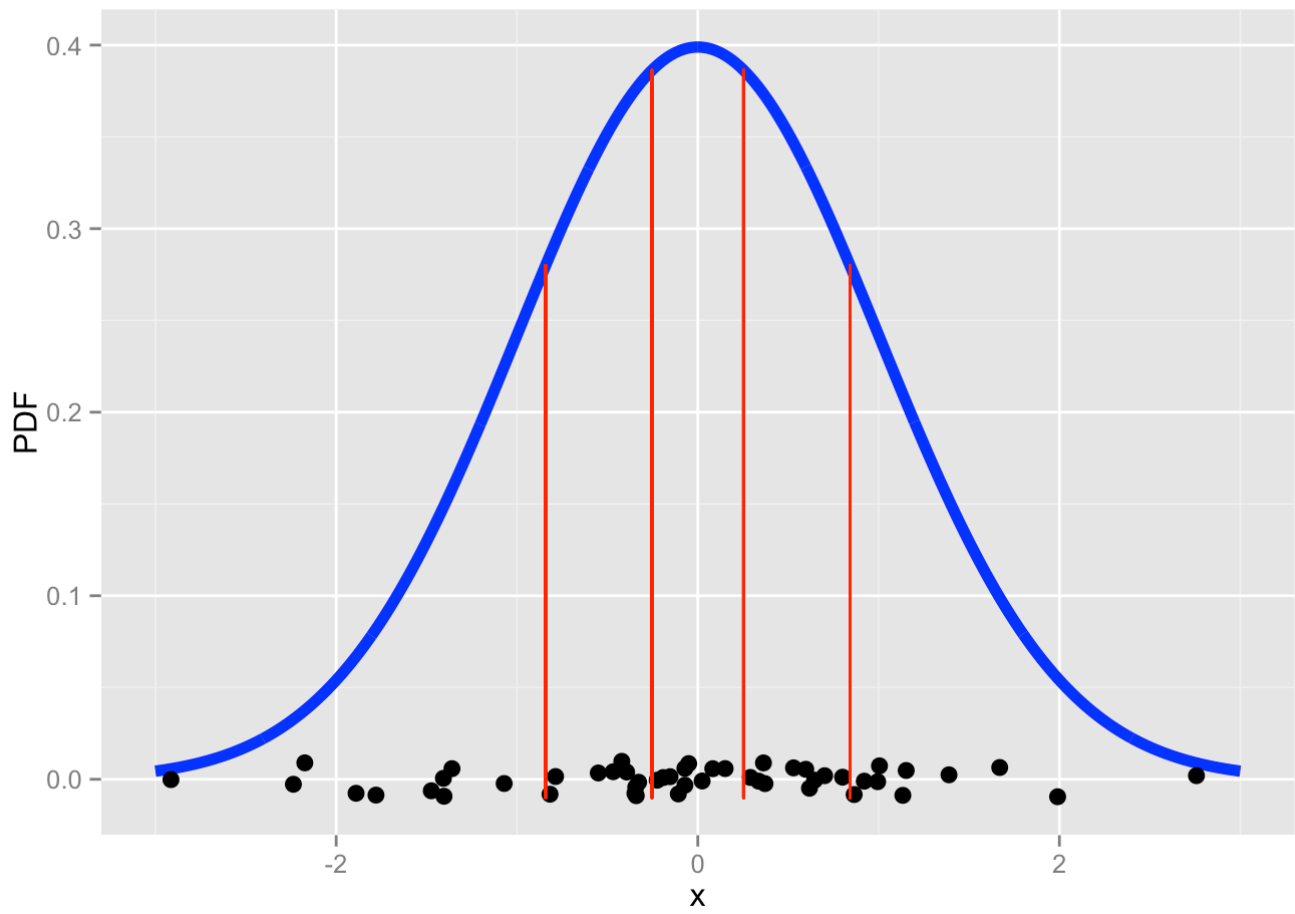
```
Nsamps = 50
normSamps <- data.frame(x=rnorm(Nsamps),y=0)
ggplot(plotDF,aes(x=x, y=PDF)) + geom_line(size=2,color="blue") + geom_jitter(data =normSamps,
aes(x=x,y=y),position = position_jitter(height = .01),size=3) +
  geom_line(aes(x=c(-2,-2),y=c(0,max(plotDF$PDF))), color="red") + geom_line(aes(x=c(2,2),y=c(0,max(plotDF$P
DF))), color="red")
```



Instead, let's divide the distribution into five intervals of equal probability. The first goes from negative infinity to where the CDF, $F(x)$, equals .2, call this point $x_{.2}$. The second bin goes from $x_{.2}$ to $x_{.4}$; the third bin goes from $x_{.4}$ to $x_{.6}$; the third bin goes from $x_{.6}$ to $x_{.8}$. The final bin extends from $x_{.8}$ to ∞ .

$$x_{.2} = -0.8416 \quad x_{.4} = -0.2533 \quad x_{.6} = 0.2533 \quad x_{.8} = 0.8416$$

Now what we is put $N/5$ points in each bin, and then pick a point inside that bin.



If we make 10 bins, then it spreads out even more.

