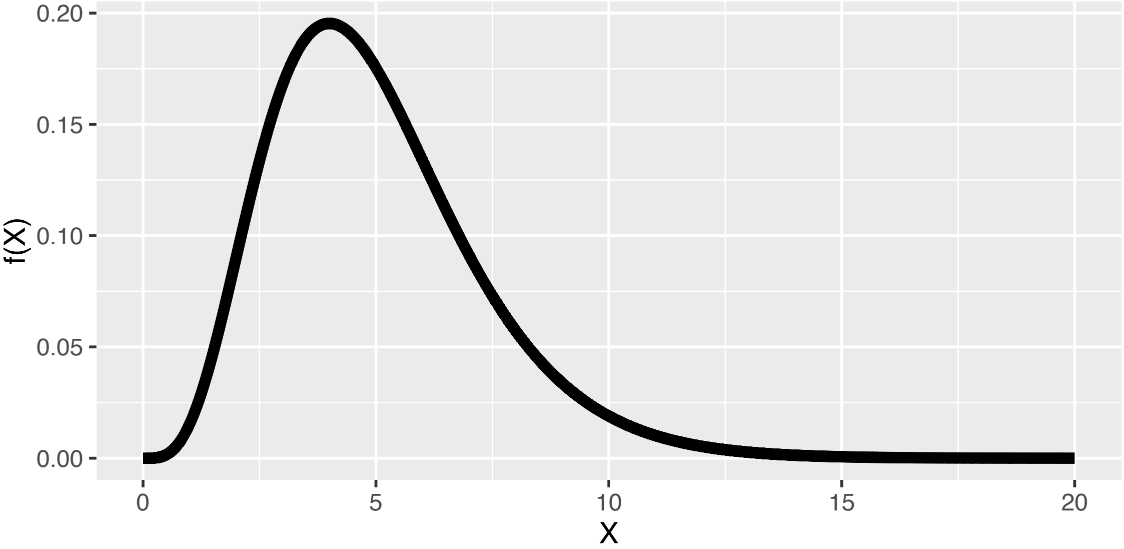
Module 3 Lab

In this lab we will explore sampling distributions. We start by loading `ggplot2`.

```
library("ggplot2")
```

Consider a Gamma(5, 1) distribution, which has an expected value of five.



We can draw `n` random variables from this distribution with the command `rgamma(n, 5, 1)`. Remember, if you want more information about this function, run `?rgamma`. Let's go ahead and try this.

```
rgamma(10, 5, 1) # Generate 10 Gamma(5,1) random variables
```

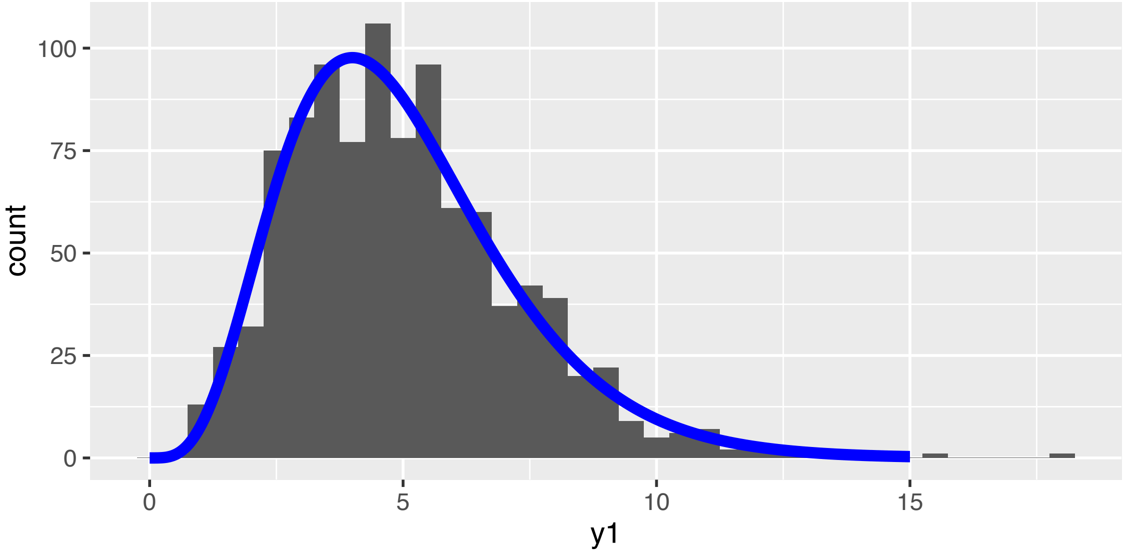
```
##  [1] 7.038159 2.962818 2.975134 1.974349 3.102149 2.227713 2.965589
##  [8] 8.381643 3.090611 6.508384
```

So there they are, but the list is not particularly insightful. Let's generate 1000, and look at a histogram to make sure they are actually coming from the same distribution as the density function above. Don't worry about the extra lines of code; they are there to add the Gamma(5,1) curve over the histogram.

```
y1 <- rgamma(1000, 5, 1) # Generate 1000 Gamma(5, 1) random variables

# Create a set of x, y points to draw the pdf of Gamma(5, 1)
x1 <- seq(0, 15, length = 1000)
y2 <- dgamma(x1, 5, 1)

qplot(y1, binwidth = 0.5) +
  # add the pdf to histogram
  geom_line(aes(y = 1000*0.5*y2, x = x1), size = 2, color = "blue")
```



Looks pretty similar!

Okay, now we will use this distribution to explore sampling distributions. First, we draw a sample of size ten from a Gamma(5,1), and find the sample mean. Remember that the sample mean is an estimate of the true mean of a Gamma(5,1), which is five.

```
draws <- rgamma(10, 5, 1) # Sample of size 10 from Gamma(5,1), stored as "draws"
draws # Show the sample
```

```
##  [1] 8.258371 7.601965 3.074824 2.770866 1.607407 2.881101 2.912440
##  [8] 5.527795 8.191465 9.033992
```

```
avg <- mean(draws) # store the mean of the sample of 10
avg # show the mean
```

```
## [1] 5.186023
```

Make sense? Let's cut down on the typing a bit, by combining these steps into one line of code.

```
mean(rgamma(10,5,1)) # Same process, all in one line of code!
```

Remember, you will get a different value than above, because the function is randomly generating Gamma random variables each time it is called. To understand this nested code, work from the inside out. First R executes `rgamma(10, 5, 1)`, then it takes these draws and executes `mean(draws)`.

Back to sampling distributions. We want to know–how much can we trust this estimate? Further, how does this estimate behave in the long run; in other words, what is its distribution? To find out, lets draw a **sample of size ten** ($n = 10$) and find the mean of the sample—10,000 times. Sometimes we say this is 10,000 replications, in this case of the procedure: draw a sample of size ten and calculate the mean.

```
x <- replicate(10000, mean(rgamma(10, 5, 1))) # Sample size = 10; 10,000 replications
qplot(x, binwidth=.05, xlim=c(3, 7)) # Create a histogram of results
```

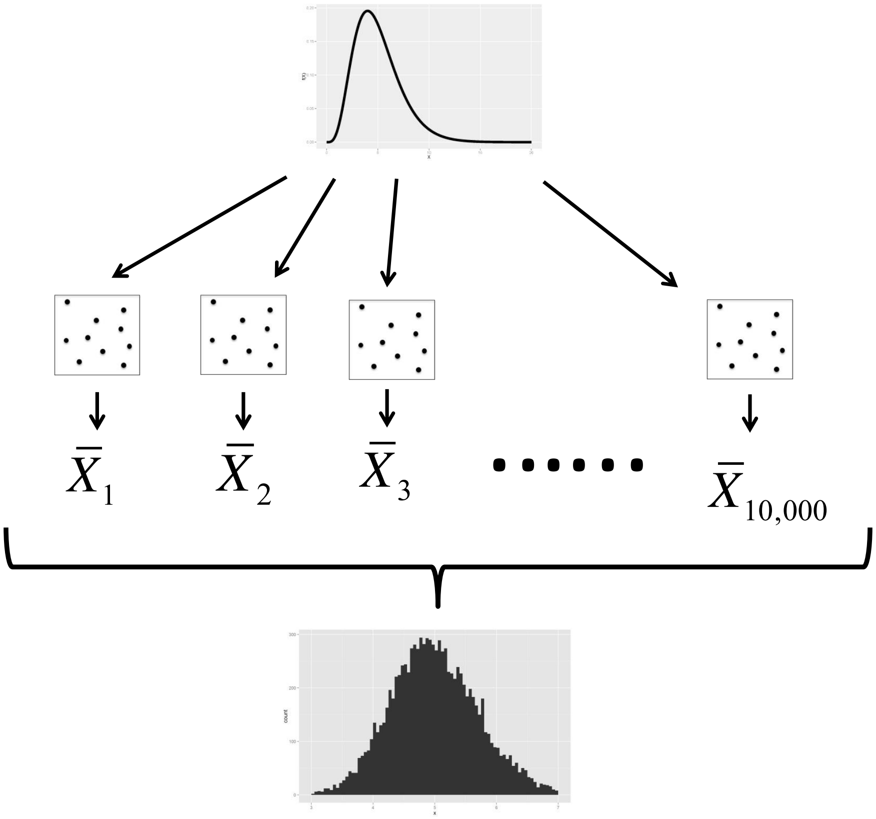
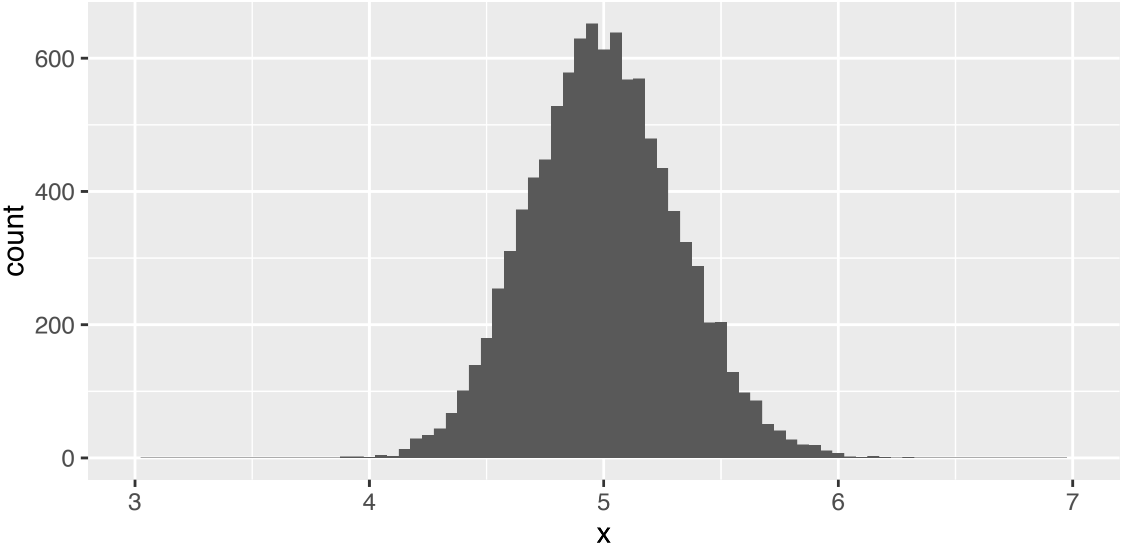


Figure 1

This histogram is the result of **10,000 replications** of taking the mean of **samples of size ten**. The graphic might help visualize what we are doing. The top curve is a Gamma(5,1), the distribution from which we are drawing samples of size ten. The next row, of boxes filled with ten dots, represent the 10,000 samples of size ten we will draw. The row of $\overline{X}$s shows that we calculate a mean for each individual size ten. Finally, the histogram summarizes the 10,000 sample means we calculated. The histogram is centered over five, so *on average* a size ten gives us the true population mean. But quite often the estimate will be above six, or below four. That is a pretty big miss. If we want more confidence in our estimate, we have to take larger samples. Instead of taking the mean of ten draws from the Gamma(5,1) as our estimator, we will try taking the mean of samples of size 50.

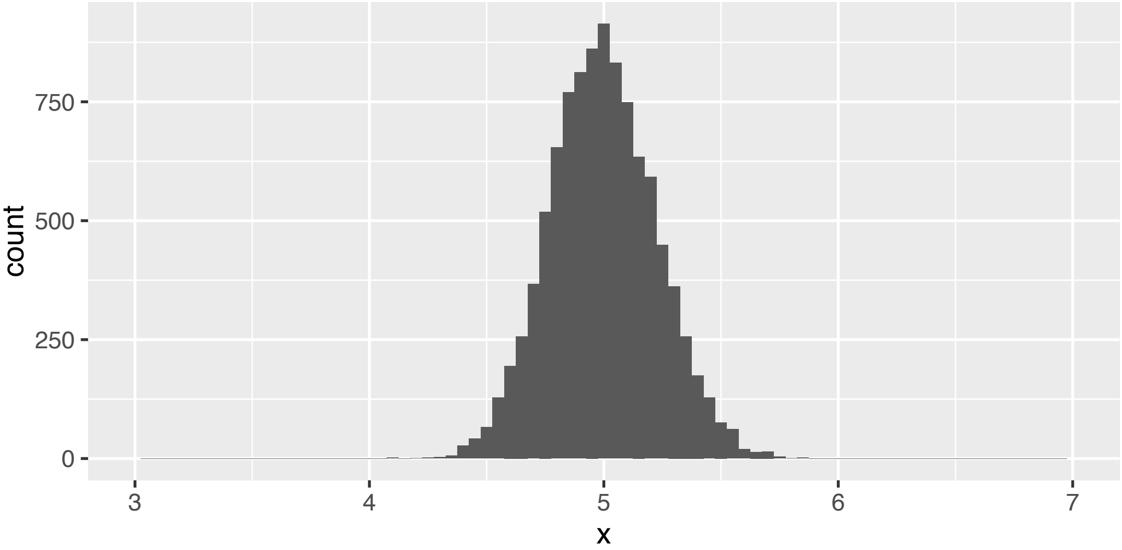
```
x <- replicate(10000, mean(rgamma(50, 5, 1))) # n = 50; 10,000 replications
qplot(x, binwidth = 0.05, xlim=c(3,7)) # Create histogram
```



We used the **same number of replications** (10,000), but now at each replication we calculated the mean of a **sample of size 50**. The histogram is still centered over five (the true population mean), but now the histogram is much narrower, meaning that the sample means are varying less. In other words, our estimate of the population mean is getting much better when our **sample size** increases.

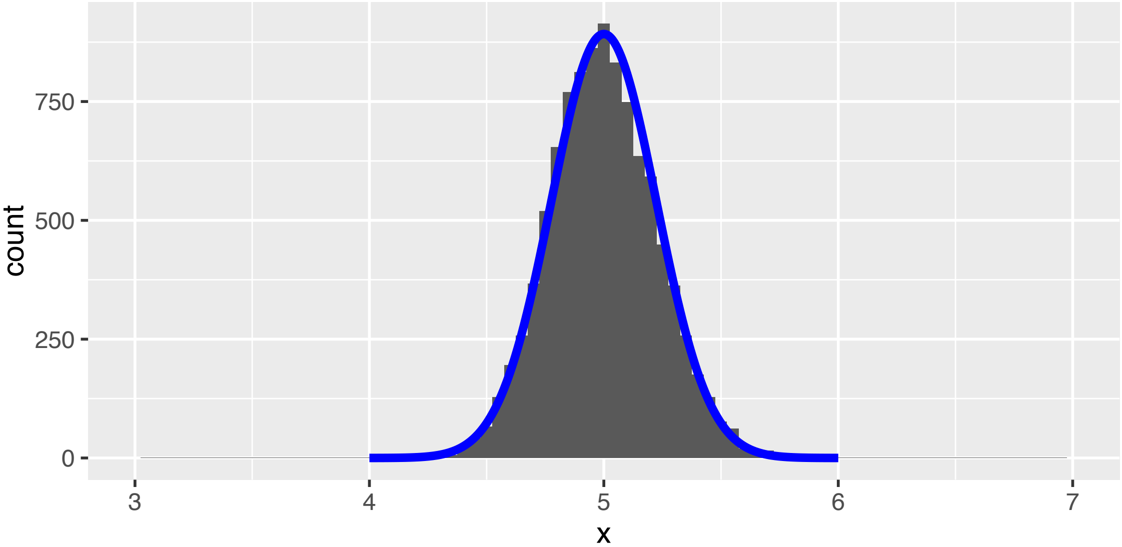
Let's try one more. This time we will find the mean of samples of size 100 from the Gamma(5,1) distribution.

```
x <- replicate(10000, mean(rgamma(100, 5, 1))) # n = 100; 10,000 replications
qplot(x, binwidth = 0.05, xlim=c(3,7)) # Create histogram
```



Still centered over five, the true mean of a Gamma(5,1) distribution, but the histogram is even narrower! The variance of the sample mean is decreasing, as the sample size increases. What does this remind you of? I'll give you a hint. The following is a histogram of 10,000 replications of sample means with n = 100, with a $Normal(5, \sqrt{5/100})$ superimposed (the variance of a Gamma(5,1) random variable is 5).

```
x1 <- seq(4, 6, length = 1000); y1 <- dnorm(x1, 5, sqrt(5)/10)
qplot(x, binwidth = .05, xlim=c(3,7)) +
  geom_line(aes(y = 10000*0.05*y1, x = x1), size = 1.5, color = "blue")
```


count
x