Lab: Playing with Probability

36-600

Fall 2022

In lecture, you learned the rudiments of probability theory, and here, you will build intuition about various often-used distributions that you will encounter when doing data analyses.

First, however, we need to describe the functions that `R` provides that relate to probability distributions. We'll start by giving the name of one of these: `dnorm()`. (You may have noticed that I tack on parantheses after function names. This is to make it clear to the reader that what is being referred to is a function and not, say, a variable.) We can split the name into two parts: `- d`: this means that the function is returning $f_X(x)$, i.e., the probability density function value given the coordinate x; and `- norm`: this means that we are working with the normal distribution.

First we'll list the names of the distributions that we listed in the lecture notes:

`R` Name	Distribution	Parameters (See Help Pages For More)
`binom`	binomial	`size` (n) and `prob` (p)
`pois`	Poisson	`lambda` (λ)
`norm`	normal	`mean` (μ) and `sd` (σ)
`t`	t	`df` (ν)
`exp`	exponential	`rate` (β)
`chisq`	chi-square	`df` (ν)

And second we'll list the prefixes that you use to accomplish various tasks:

Prefix	What it Does
`d`	returns the pdf $f_X(x)$ or pmf $p_X(x)$
`p`	returns the cdf $F_X(x)$
`q`	returns the inverse cdf $F_X^{-1}(x)$
`r`	returns a random sample of data

So, what's the cdf? That's the *cumulative distribution function*, and it is defined as

$$F_X(x) = \sum_{y \leq x} p_Y(y)$$

for a discrete distribution, or

$$F_X(x) = \int_{-\infty}^{x} f_Y(y) dy$$

for a continuous distribution. To make the math more intuitively meaningful: let's say you flip a coin four times, and record the number of heads. The possible number of heads is x = 0, 1, 2, 3, or 4, and the probability mass function $p_X(x)$ is given by the binomial distribution. The cdf $F_X(2)$ would be the sum of the probabilities for 0, 1, and 2. $F_X(1)$ would be the sum for 0 and 1. Etc. Why would we use the cdf?

Let's say you want to know the probability of sampling a datum with value between 3 and 8 given a normal distribution with mean 4 and standard deviation 2. That probability would be

```
pnorm(8,mean=4,sd=2) - pnorm(3,mean=4,sd=2)
```

The first term is the integral from $-\infty$ to 8, and the second term is the integral from $-\infty$ to 3, so the difference is the integral from 3 to 8...and that value, which lies between 0 and 1, is the probability we are looking for.

Last: what is an inverse cdf? It takes in the value of the cdf as input and returns the coordinate x associated with that cdf as output. In other words, if you have the equation

$$q = \int_{-\infty}^{x} f_Y(y) dy \,,$$

where *you have set the value* q, you would be solving for the value x. CDF: we set x and solve for q. Inverse CDF: we set q and solve for x.

Assuming a one-to-one relationship between x and $q = F_X(x)$, it is the case that $x = F_X^{-1}(F_X(x))$. So, for instance,

```
qnorm(pnorm(8,mean=4,sd=2),mean=4,sd=2)
```

would return `8` as output.

OK...enough of this. Let's play.

Question 1

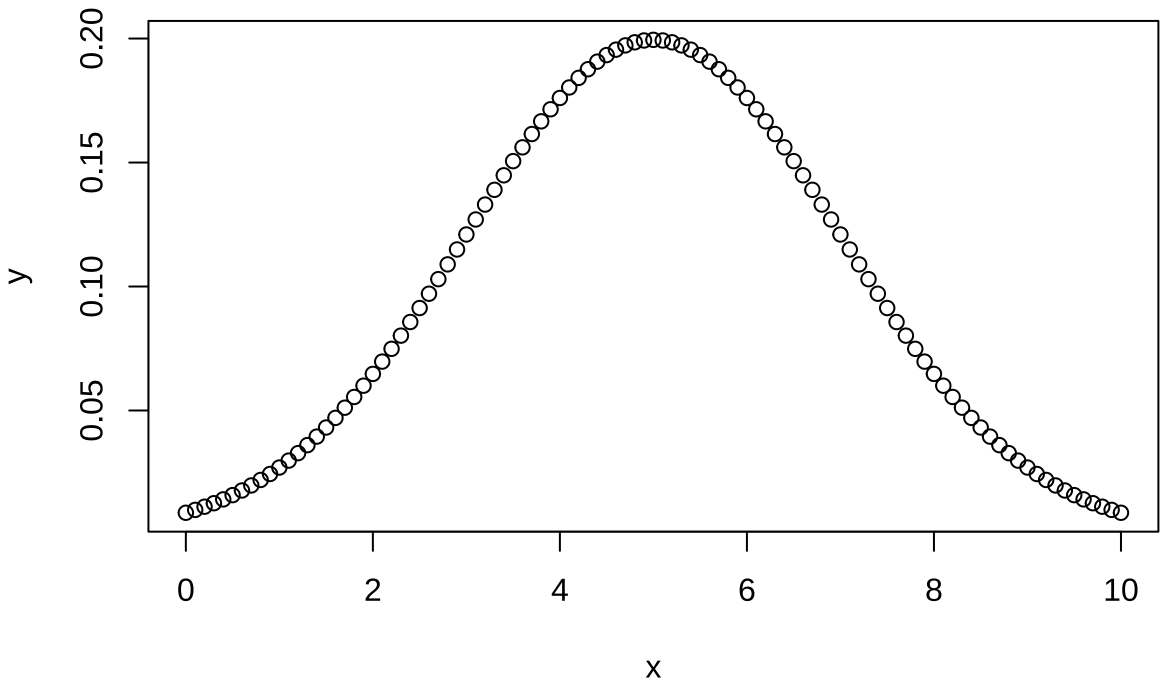
Well, OK, before we play, we still need to lay out a bit more background material. Recall that you can define a vector using the `seq()` function. For instance,

```
seq(0,10,by=0.1)
```

defines the sequence {0,0.1,0.2,...,9.9,10}. If we want to plot distributions, defining a sequence is a good way to start.

So: define an appropriate vector `x` and plot the probability density function $f_X(x)$ on the y-axis versus x on the x-axis for a normal distribution with mean 5 and standard deviation 2. To give you a hint as to how "wide" to define the sequence of values `x`, know that in general, nearly all the probability content of a distribution lies within three standard deviations of the mean. Put a little cushion on this, and define the sequence to stretch over a range of four or five standard deviations on either side of the mean. Oh...for plotting, you can create a basic plot via `plot(x,y)`. (There are lots of arguments to make the plot prettier, but let's start here.)

```
x <- seq(0,10,by=0.1)
y <- dnorm(x,mean=5,sd=2)
plot(x,y)
```



Question 2

Compute the probability that, given the distribution from Question 1, we would sample a datum that has a value less than 5. Does the result make sense to you?

```
pnorm(5,mean=5,sd=2)
```

```
## [1] 0.5
```

```
# The mean is five and the distribution is symmetric around the mean, so half of the probability
# should lay to the left of the mean.
```

Question 3

Sample 100 data from the distribution defined in Question 1. Use `sum()` and a relational operator to display how many of the data are less than 5. Then incorporate `length()` to display the *proportion* of the data that are less than 5. Note that this proportion is a random variable: you will not get the same result as your neighbor, unless dumb luck intervenes, or…see Question 4.

```
x <- rnorm(100,mean=5,sd=2)
sum(x<5)
```

```
## [1] 51
```

```
sum(x<5)/length(x)
```

```
## [1] 0.51
```

Question 4

What you will find is that each time you run the code in the code chunk for Question 3, you will get a different result. The reason is that every time you sample, you get a different dataset. This impedes reproducibility.

Hence, when you sample data, you should always set a *random number seed* first. You do that in `R` by calling the function `set.seed()` and passing in a number (your choice) as an argument.

Below, set the random number seed, sample five numbers from the distribution defined in Question 1, and display your sample. (You can do this by simply saying `x`, or by saying `print(x)`.) Then repeat the process with the same random number seed. You should see that your five numbers are exactly the same.

```
set.seed(101)
rnorm(5,mean=5,sd=2)
```

```
## [1] 4.347927 6.104924 3.650112 5.428719 5.621538
```

```
set.seed(101)
rnorm(5,mean=5,sd=2)
```

```
## [1] 4.347927 6.104924 3.650112 5.428719 5.621538
```

Question 5

Keeping the distribution the same... what is the probability that a sampled datum will be between 3 and 7? How about between 1 and 9? And last, -1 and 11? If you've taken AP Statistics or an introductory stats class in college, these numbers *may* be familiar to you. (Then again... AP Stats would have been so long ago... to which I would ultimately say "bah!" because I can still do calc and I learned it from Newton.)

```
pnorm(7,mean=5,sd=2) - pnorm(3,mean=5,sd=2)   # 68%
```

```
## [1] 0.6826895
```

```
pnorm(9,mean=5,sd=2) - pnorm(1,mean=5,sd=2)   # 95%
```

```
## [1] 0.9544997
```

```
pnorm(11,mean=5,sd=2) - pnorm(-1,mean=5,sd=2) # 100%
```

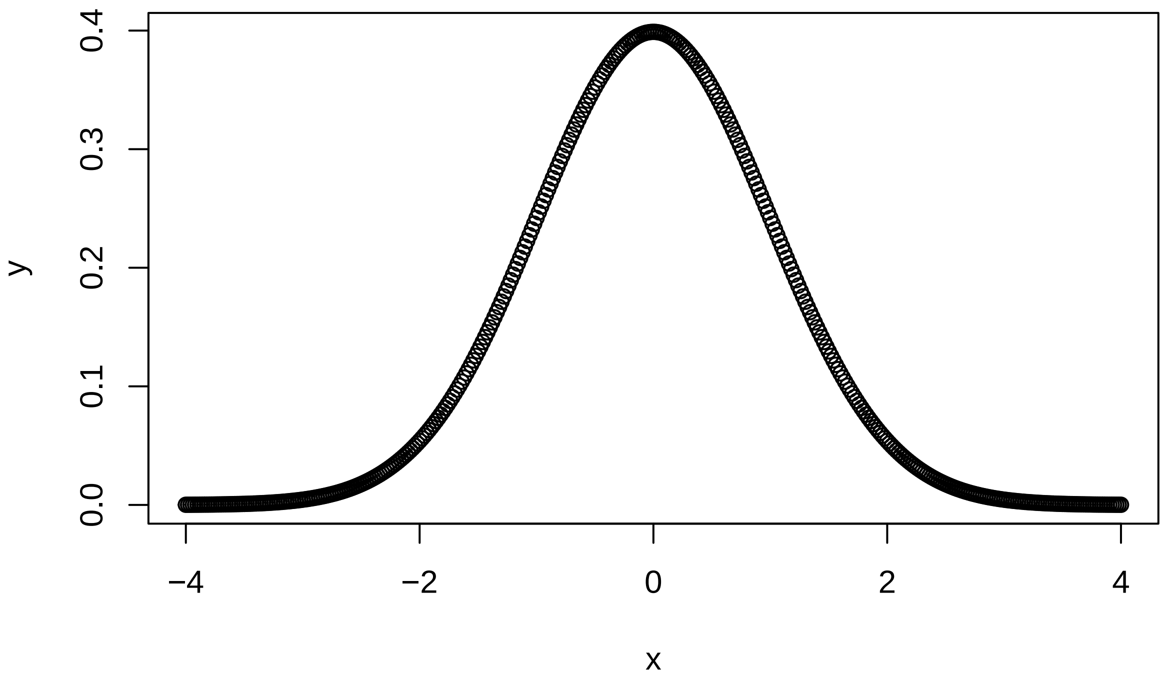
```
## [1] 0.9973002
```

Question 6

Now we will change distributions, to the *standard normal*. This distribution was not listed above. That's because a standard normal is simply a normal with mean $\mu = 0$ and standard deviation $\sigma = 1$. And... these values are the default values for `mean` and `sd` in the `R` functions relating to the normal.

Plot a standard normal distribution. Note that you'll have to change your sequence `x`!

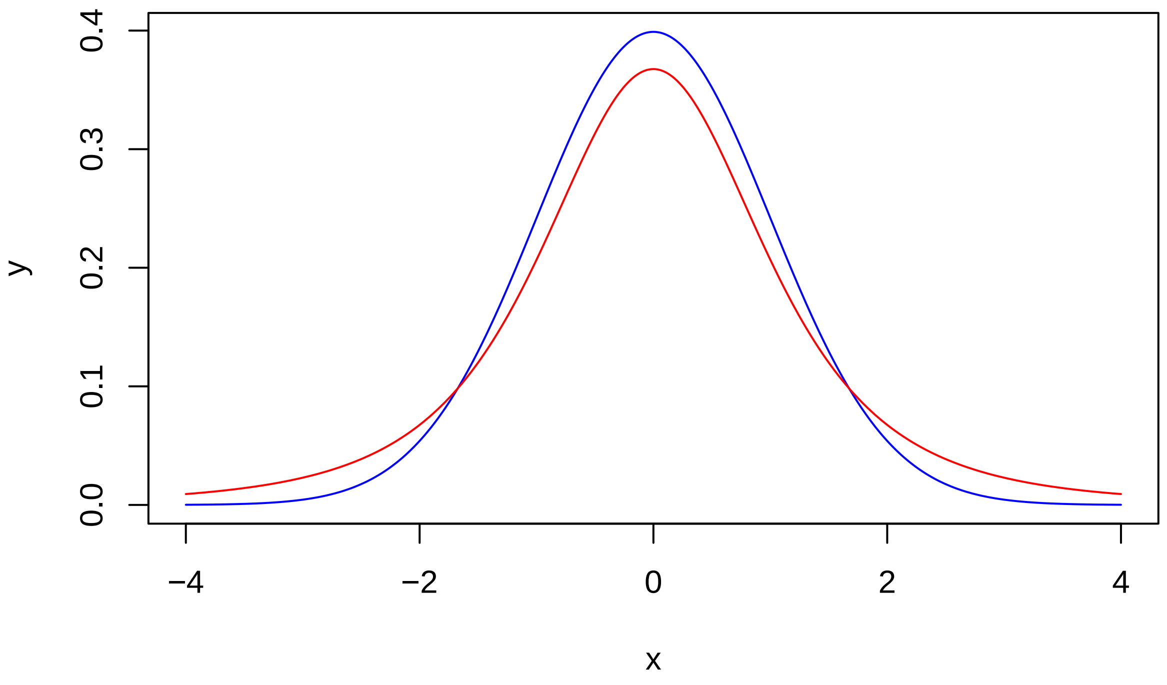
```
x <- seq(-4,4,by=0.02)
y <- dnorm(x)
plot(x,y)
```



Question 7

Plot a standard normal distribution again. To make it look better, include the following arguments in the call to `plot()`: `typ="l"` and `col="blue"`. Then overplot a *t* distribution for 3 degrees of freedom. (Use the same `x`, but now call `dt()`, save the output to the variable `z`, and type `lines(x,z,col="red")`. This last command should plop a red curve down on top of your blue one, in the same plot pane. If things go awry, ask for help!) How does the `t` distribution differ from the standard normal, qualitatively?

```
plot(x,y,typ="l",col="blue")
z <- dt(x,3)
lines(x,z,col="red")
```

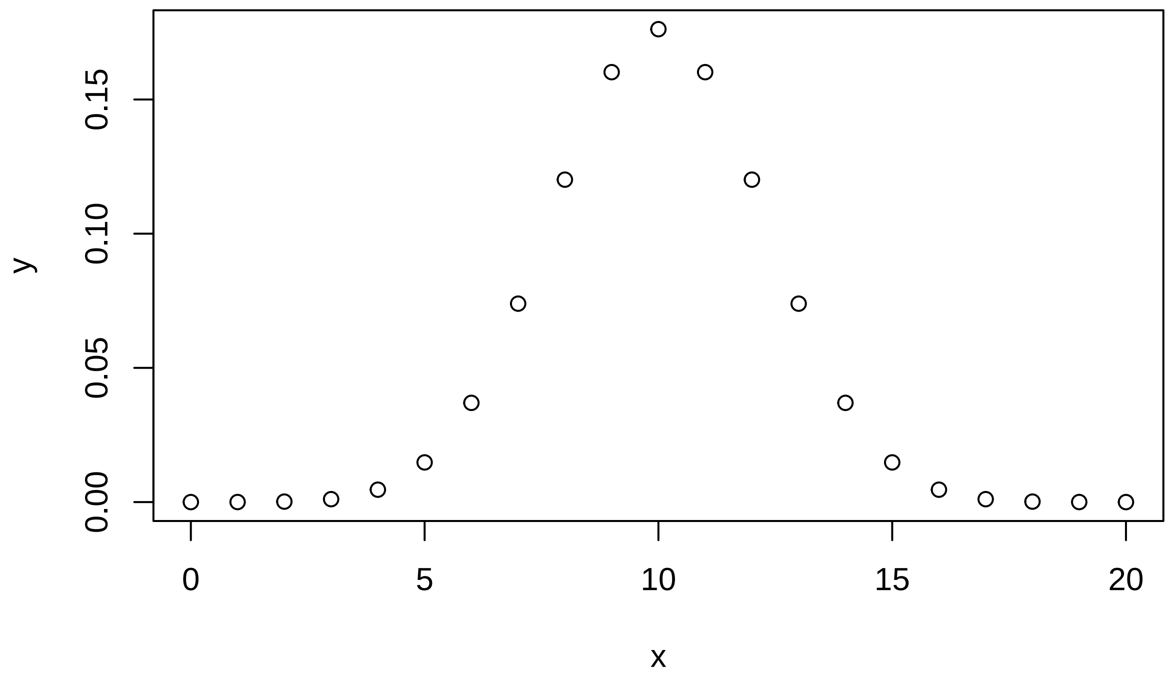


```
# The t distribution is like the standard normal, but shorter and with wider tails.
```

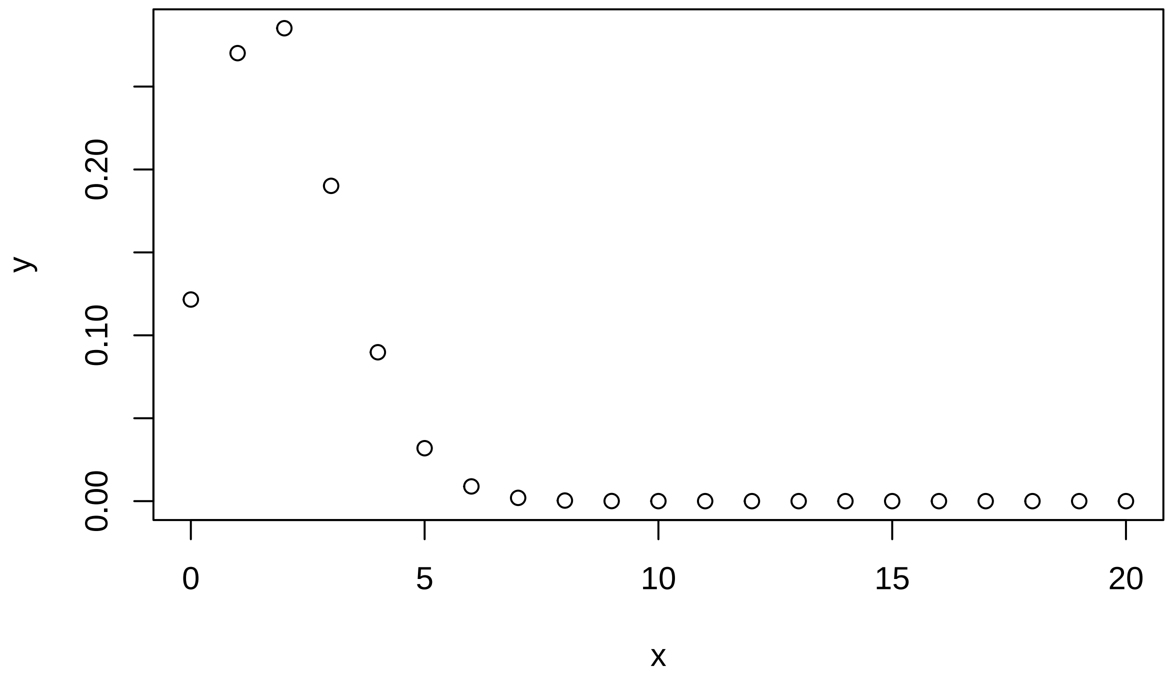
Question 8

Plot the probability mass function $p_X(x)$ for a binomial distribution with $n = 20$ and $p = 0.5$. Does it look like a normal distribution, at least in general shape? Next, make two more pmf plots, one with $p = 0.1$ and one with $p = 0.9$. Do these pmfs have a similar shape to a normal distribution? (Note that what you see here is related to whether or not you can perform a population proportion hypothesis test, which requires that the binomial pmf has, approximately, the shape of a normal distribution.) For the `x` in your plot, use `x <- 0:20`, i.e., `x` has values 0, 1, . . . , 20.

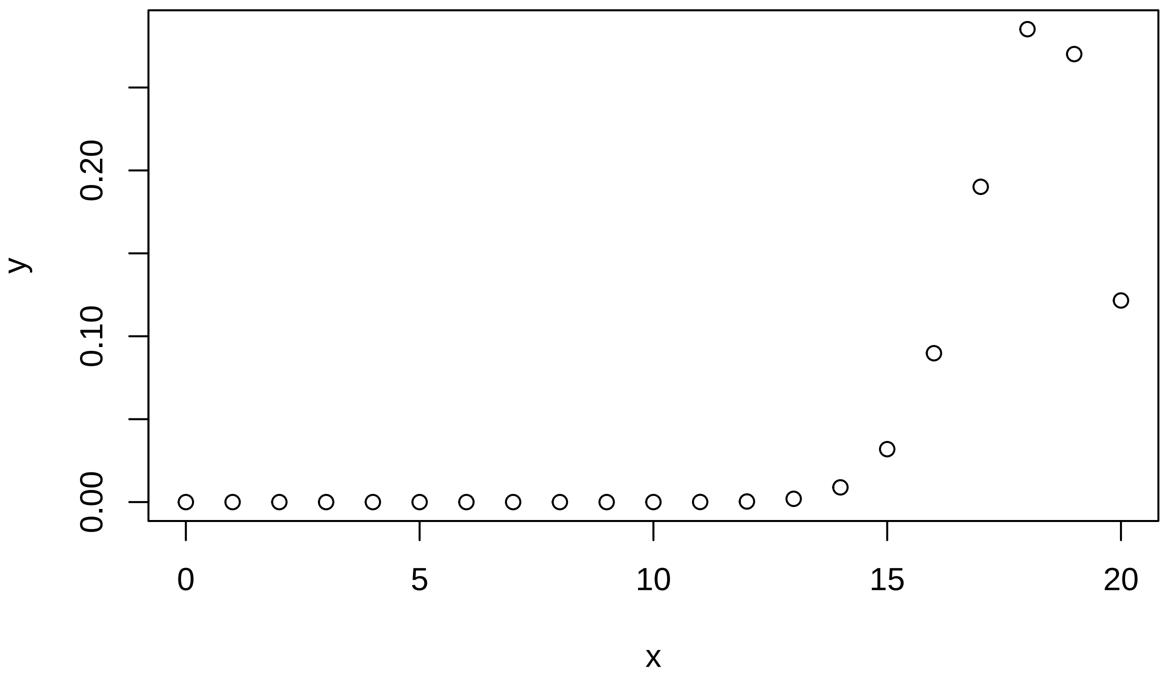
```
x <- 0:20
y <- dbinom(x,20,0.5)
plot(x,y)
```



```
y <- dbinom(x,20,0.1)
plot(x,y)
```



```
y <- dbinom(x,20,0.9)
plot(x,y)
```

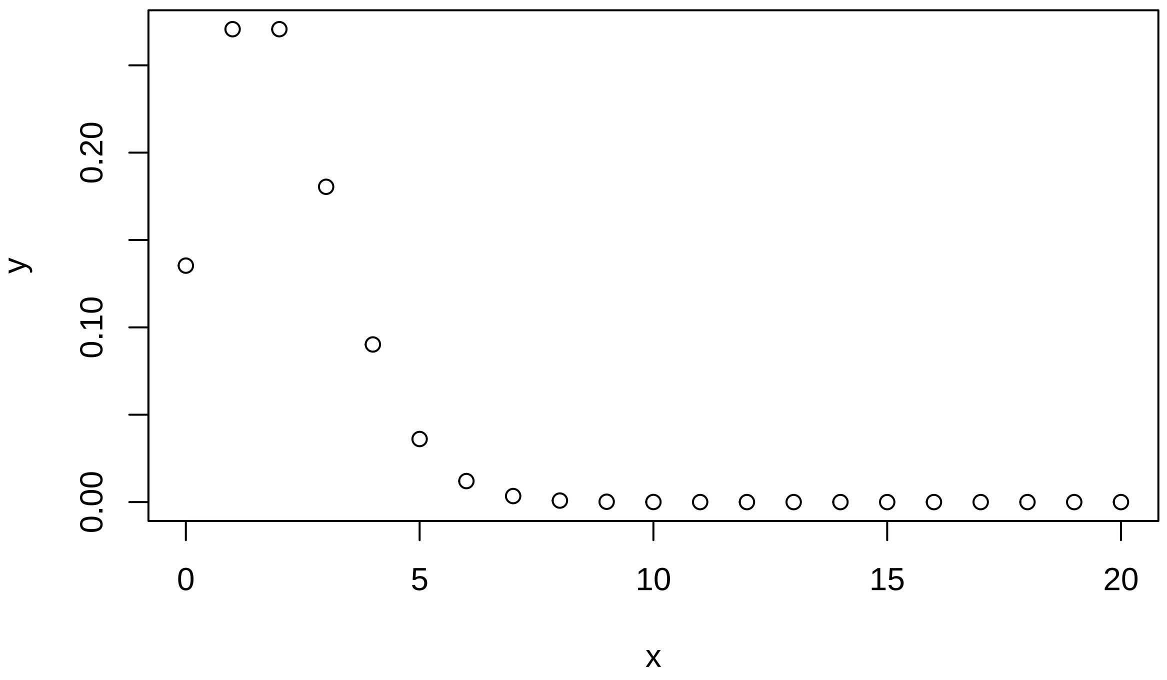


```
# The closer the probability p is to 0.5, the more "normal" the distribution looks.
```

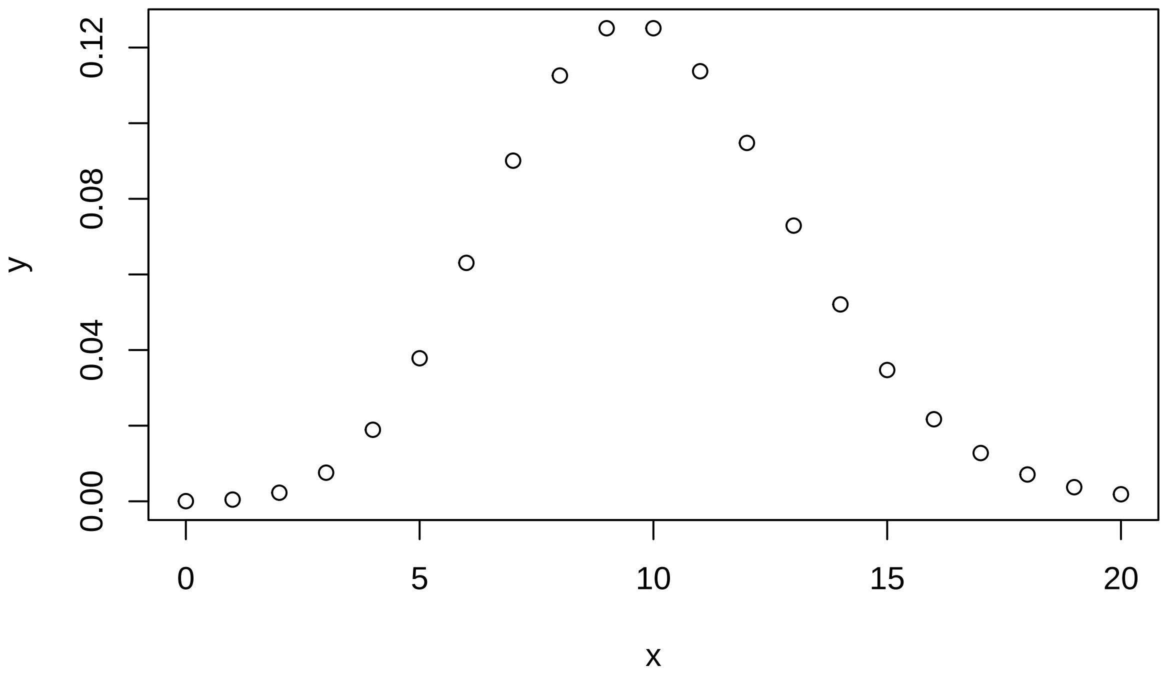
Question 9

Plot the probability mass function $p_X(x)$ for a Poisson distribution with $\lambda = 2$ and $\lambda = 10$. You should see "issues" similar to what you observed with you completed Question 8. (To give away a result: as λ gets larger, the Poisson pmf adopts a shape more and more like that of a normal distribution.) Here, use the same `x` as above (0:20).

```
y <- dpois(x,2)
plot(x,y)
```



```
y <- dpois(x,10)
plot(x,y)
```



Question 10

In any given year, there are, on average, 4.5 earthquakes with magnitude greater than 5 in California. (I *completely* made that up. Sue me.) What is the probability that in a given year, we will actually observe 6 or more such earthquakes? Utilize the Poisson distribution here, and realize you'll need to add a little bit of additional stuff to a bare function call. Not sure what I mean? Call us over…

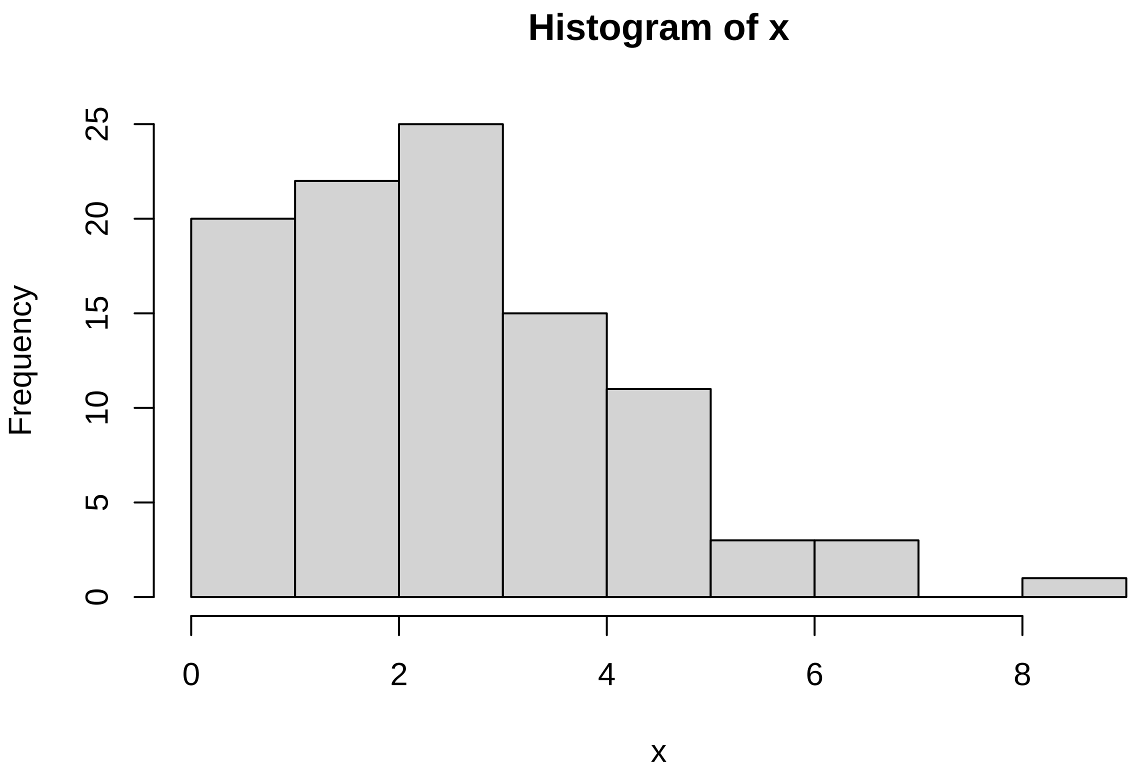
```r
# The probability of "six or more" is one minus the probability of "five or less"
1 - ppois(5,4.5)
```

```
## [1] 0.2970696
```

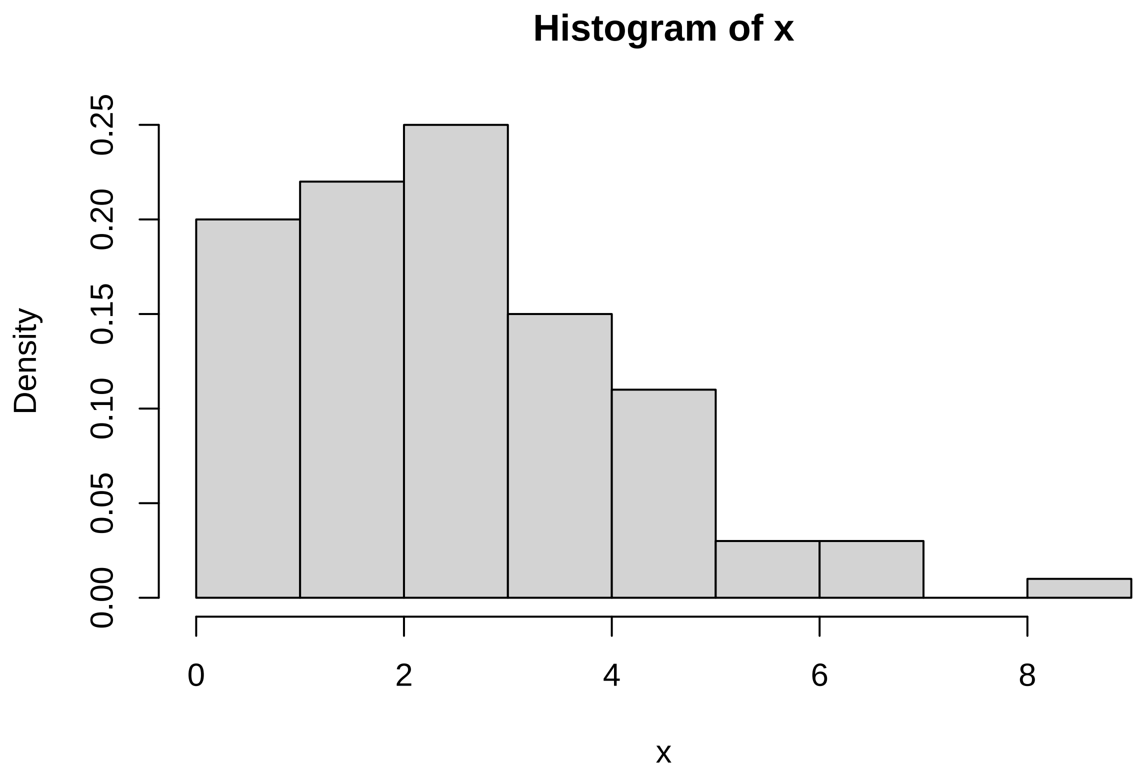
Question 11

Sample 100 data from a Poisson distribution with $\lambda = 3$. Visualize these data via a histogram: if your data are contained in a vector called `x`, then `hist(x)` will create the histogram. By default, what is shown is a *frequency histogram*, meaning the number of data observed within each histogram bin is displayed. If you want to show something that is akin to a probability mass function instead, add the argument `prob=TRUE` to the histogram function call. (Note: you should set the seed here, for reproducibility!)

```r
set.seed(202)
x <- rpois(100,3)
hist(x)
```



```r
hist(x,prob=TRUE)
```



Question 12

(Looking ahead.) In Question 11, you generated a vector with 100 data. Statisticians generally do not like working with entire datasets, when it is possible to make inferences with *summary statistics*. These statistics include ones like the *sample mean*, the *sample standard deviation*, the *sample median*, etc. Below, compute these statistics, via the `mean()`, `sd()`, and `median()` functions. Also, pass your vector to the `summary()` function, which gives a six-number summary showing the minimum value (which you can also find with the function `min()`), the maximum value (`max()`), the mean and median, and the 25th and 75th percentiles (i.e., the first quartile, or "1st Qu.", and the third quartile, or "3rd Qu.", numbers you could also find via the function `quantile()` [yes, "quan" and not "quar"…that's not a typo]).

```
mean(x)
```

```
## [1] 2.98
```

```
sd(x)
```

```
## [1] 1.717292
```

```
median(x)
```

```
## [1] 3
```

```
summary(x)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    2.00    3.00    2.98    4.00    9.00
```

Question 13

(One last question.) So in the lecture notes, we talked about how the expected value $E[X]$ is the average value of the next datum that you sample from a distribution… but nowhere above do we compute one. That's because R is in the business of analyzing data, not generating theoretical results about distributions. Put another way, you are expected to know (or look up) what the mean (i.e., expected value) and standard deviation are for any stated distribution.

But what if you don't want to look it up?

Well, you could get a fairly precise (if not exactly correct) figure via numerical integration.

Here's an example for an exponential distribution, for which the probability density function is

$$f_X(x) = \beta \exp(-\beta x) \,,$$

where `exp()` represents the constant mathematical constant e. (So, e^6 would be computed in R as `exp(6)`.)

```r
integrand = function(x,beta=1) {
  return(x*beta*exp(-x*beta))  # for the mean, we integrate x*f(x)
}
integrate(integrand,0,Inf,beta=2)  # the value should be 1/beta ... which is what is output
```

```
## 0.5 with absolute error < 8.6e-06
```

Try this yourself with the simplest of all continuous distributions, the uniform. The uniform has two parameters (`a`, the minimum value, and `b`, the maximum value), and the pdf $f_X(x)$ is

$$f_X(x) = \frac{1}{b-a} \,.$$

The mean is $(a+b)/2$; if $a = 0$ and $b = 1$, the mean is $1/2$. Note that while mimicking the code above, all you would need is to pass to `integrate` is three arguments: the integrating function (`integrand`) and the numerical values of a and b (which are the bounds of integration). The same three arguments should be passed to `integrand` itself.

```r
integrand = function(x,a=0,b=1) {
  return(x/(b-a))
}
integrate(integrand,0,1)
```

```
## 0.5 with absolute error < 5.6e-15
```

A final point: you are not expected to internalize all the information about probability instantly, and depending on your background in mathematics and statistics you might be confused by some of today's material. That's normal (pun not intended) and fine, and you should feel free to ping the 36-600 team via email or at office hours for more explanation(s) now and throughout the semester.