Homework # 3:

Note: Please circle your answers when appropriate!

As usual, 3rd edition references are in bold and in [ ]

4th edition references in italics and {}

1) 3.12, p. 103 **[3.15, p. 95]** *{3.4.1, p102}*

But do the following:

(a) between 6 and 14 inches *b) less than 8 inches* *c) more than 4 inches*

2) 3.16, p. 113 **[3.27, p. 110]** *{3.6.2, p. 115}*

Let's consider the world population instead of just the U.S. - World wide 32% of people have blood type A. Now follow the instructions in the text for parts (a - e):

3) 3.17, p. 113 **[3.28, p. 111]** *{3.6.3, p. 115}*

Do (a) and (b) as in the text. For (c) & (d) do the following:

(c) exactly 12 will be cured *(e) exactly 60% will be cured*

4) 3.18, p. 113 **[3.29, p. 111]** *{3.6.4, p. 115}*

But do the following:

(a) 20% *(b) 30%* *(c) 90%*

5) 3.21, p. 114 **[3.33, p. 111]** *{3.6.9, p. 115}. No changes to this problem.*

6) Let's do some R....

Suppose you have the following situation:

You have a large jar of beans, 17% white, 83% black. You take a sample of 12 beans.

a) Use R to figure out the probability for every single possible outcome. In other words, you need to calculate:

Pr{0 black beans in 12 trials}
Pr{1 black bean in 12 trials}
.
. etc.
.
Pr{12 black beans in 12 trials}

b) Once you have this information, you need to plot a histogram that shows you what this distribution looks like (actually, this would be a barplot).

If you don't know, you'll need to use the binomial (in R) to solve this problem. Make sure you present all your results (you should have a list of probabilities and the histogram/barplot).

7) Repeat question 6, but now assume that you have 48*% white beans and 52% black beans in the jar and take a sample of 8 beans*. Obviously you want to use R again here.

In other words repeat (a) and (b) from problem 6 using the new numbers.

Also answer the following:

c) How is this distribution different from the one in question 6?

Some R instructions to help you are on the next page. The graphs will look a lot nicer using the command line, but R-commander is okay. See also the instructions for copying graphs from R (at the end of the R instructions).

Incidentally, never just hand in a R printout: <u>Clean it up. Clearly label everything. You will not do very well if all you do is hand in a printout.</u>

BIOL 214 - problems are due in recitation, Monday, June 17th. **For R problems:** be prepared to sketch, list, or show your results if you are called on to present them. There is a document camera in the room you can use. You will ***not*** have to use your computer during your presentation.

BIOL 312 - problems are due at the beginning of lab, Tuesday, June 18th.

Using R from the command line:

For probabilities:

You want probabilities for 0 through 12 beans. First we need to create y with 13 numbers:

```
y <- c(0:12)
```

This command says to give "y" the numbers 0 - 12 in sequence; technically "0:12" tells R it's a sequence of numbers, and the "c" out front means to combine all the numbers. The "c" isn't really needed (try it!) this time, but it is required in many similar situations so it's a good habit to get into to.

Now we just "feed" our y into the binomial function to get our answers:

```
p <- dbinom( y, size = 12, prob = .83)
```

This should give you "p" with all the binomial probabilities (they'll be in order from 0 to 12; type "p" to get the probabilities)

You should be able to figure out how to do question 6 on your own

To get a barplot:

To get a barplot, you simply do:

```
barplot(p)
```

This won't look very nice, so you should try to improve it:

```
barplot(p,names.arg = y)
```

Here "names.arg" is the variable that holds the labels you want to put on the x axis (remember we put the numbers 0 - 12 into y). Just try it - you'll see how it works.

If you want to improve your graph even more, you can do:

```
barplot(p,names.arg = y, ylab = "frequency", xlab = "number of
dark beans")
```

Or even fancier:

```
barplot(p,names.arg = y, ylab = "frequency", xlab = "number of
dark beans", col = "blue")
```

Using R-commander:

For probabilities:

This is quite simple. Click on the following:

Distributions --> Discrete distributions --> Binomial distribution --> Binomial probabilites

Now fill in the box that pops up. "Binomial trials" is simply N (12 in our case), and "probability of success is" our p (0.83 in our case). Now click OK

To get a barplot:

This turns out to be almost impossible in R commander. You can, however, do a plot that's similar (and will be acceptable) as follows:

Distributions --> Discrete distributions --> Binomial distribution

--> Plot binomial distribution

Enter the same information as above under calculating probabilities, and click OK.

The "barplot" you get will look quite strange, but you should be able to figure it out.

There is no way to easily "annotate" or label the x or y axes in R-commander.

To copy/save graphs generated in R (from the command line or R-commander):

You're going to want to save your graphs or copy them into a word processor so that you can hand them in or refer to them during presentations.

Windows:

In R, right click the graph, select "copy as metafile" (don't use bitmap), then open your favorite word processor and paste the graph.

Mac OS:

You should be able to copy the graph (make sure the graph window is active) from the menu, and then simply paste the graph into Word (or whatever word processor you use). If this doesn't work for some reason, the Linux instructions will work (they'll work with Windows as well).

Linux:

This is a bit complicated. A simple way to do it is to use the print screen key, but it'll look horrible. To get good looking graphs:

1) generate the graph on-screen (just as usual) to make sure it looks right.

2) type "`jpeg()`" in the script window if using R-commander, or in the command line if you're using the command line. If you're using r-commander, highlight this, and click on submit.

3) generate the graph again. *You'll notice that nothing seems to happen*. That's okay. It's writing the graph to a jpeg file.

4) now type "`dev.off()`" in the script window, highlight this, and click on submit (or simply in the command line if you're using that).

5) the graphics file should now be in your home directory. It'll have a name like "Rplotxxx.jpeg", where xxx is some number. You can always sort by modification date in your file browser to find it quickly.

6) You should now be able to insert or copy the file into your text document (e.g. Word, or whatever you're using).

7) The jpeg may not look terrific (it'll look a lot better than "print-screen". You can increase the resolution by doing (in step 2)):

```
jpeg(width = 1000,height = 1000)
```

The jpeg command defaults 480 x 480, which isn't that great. You can also use the jpeg command to give it a filename that makes sense, if you want:

```
jpeg(filename = "your-file-name",width = xxx,
height = xxx)
```