**Introduction to Working with the Binomial and R**

In class we were introduced to the binomial distribution. Recall that the binomial distribution is a discrete distribution which requires the following conditions to hold:

1) The experiment consists of trials where the only outcomes are success or failure.

2) The number of trials (*n*) is fixed.

3) The probability of success (*p*) is the same for each trial.

4) All trials are independent.

As long as these 4 conditions are met, the probability of *x* successes in *n* trials is given by the following probability mass function:



Clearly, as long as you know the values for *n* and *p,* you can compute various probabilities for the binomial distribution.

As we did for the normal distribution, let’s first do a refresher by hand:

**Your turn:**

a) Find the probability that you get exactly 3 sixes if you roll 7 fair dice.

b) Find the probability that you get at least 6 heads if you flip 8 fair coins.

Um…. Jeff?

Yes, Trent?

Are you making them calculations by hand in a computer lab again?

Ummm…. No?

**The Binomial Distribution in R**

Similar to the normal distribution, there are four basic commands used in R to work with the binomial distribution. As before, these commands are all based around four letters: **d**, **p**, **q**, and **r**. Recall that each of these commands serves a particular purpose for which the letter is an abbreviation:

d = density

p = (cumulative) probability

q = quantile

r = random

**dbinom(x,n,p)**

The dbinom command gives the value of the probability mass function of the binomial for a particular value x. Unlike the normal case, the density function in R will actually be useful for us this time because the binomial is a discrete distribution while the normal was a continuous distribution. Thus, the dbinom command will give us the probability of exactly *x* successes, given by the formula on the first page of this lab.

Note that the two other arguments that are needed with the dbinom command are *n* and *p*, the number of trials and probability of success respectively. These are the two *parameters* used to define the binomial distribution. (Recall in the normal case that we had to provide  and , the mean and standard deviation respectively.) In the other three commands for the binomial, we’ll also have to provide these two key values before R will perform the desired calculation.

**Your turn:**

Repeat the calculation in part (a) on page 1, using R to verify the answer you found by hand.

**pbinom(x,n,p)**

The pbinom command allows us to calculate the cumulative probability of the binomial distribution. This is the answer to the following sum:



The beauty of this command is that it allows us to calculate the probability without having to sum a bunch of these binomial formulas by hand.

**Your turn:**

Repeat the calculation in part (b) on page 1, using R to verify the answer you found by hand.

**qbinom(q,n,p)**

Like the normal case, the qbinom function is used to perform the inverse calculation of the cumulative probability. However, the calculation is a little different for a discrete distribution as compared to a continuous distribution. This occurs because the cdf is a step function that has “jumps” corresponding to the values where there is actual mass or positive probability. As a result, the *100qth* percentile is defined as the largest value *c* such that



If you were doing this by hand, you would have to guess and check to determine if your cumulative probability for a value *c* had exceed *q* or not. Fortunately, R removes the guesswork.

**Your turn:**

50% of people who roll 7 fair dice will get no more than how many sixes?

**rbinom(size,n,p)**

Similar to the normal case, we may wish to generate a random draw from a binomial distribution. (You definitely will for the second half of this lab.) The command is similar, with the three arguments needed for the rbinom command being the sample size desired, the number of trials *n*, and the probability of success for each trial *p*.

**Your turn:**

Generate 1000 observations from a binomial distribution with 100 trials and probability of success 0.5. (This is the equivalent of asking 1000 people to flip 100 fair coins each and record the number of heads.) Create a histogram of your data. Recognize the shape?

Now that you have a basic idea of the commands for working with the binomial, let’s see them in action.

**Your turn:**

Suppose that a standard treatment for a disease is successful in 60% of all cases. If 200 patients are given the treatment, find the following:

(i) The probability that exactly 120 patients are cured.

(ii) The probability that no more than 110 patients are cured.

(iii) The probability that between 115 and 125 patients are cured. (Careful here!)

(iv) 75% of the time, at most how many patients are cured?

**Normal Approximation to the Binomial**

Let’s revisit your randomly generated data from the rbinom command. If you did it right, your histogram should have looked very similar to a normal distribution. Let’s see how close it actually was by superimposing the appropriate normal distribution curve on the histogram.

Recall the mean of a binomial is equal to *np* and the standard deviation is . We’ll use this information with our normal distribution.

n = 100 # Here’s our number of trials.

p = 0.5 # Here’s our probability of success.

mu = n\*p # Finding the mean.

sigma = sqrt(n\*p\*(1-p)) # Finding the standard deviation.

data=rbinom(1000,n,p) # Generating random binomial data.

x=0:n

y=dnorm(x,mu,sigma) # Finding the pdf of the binomial.

hist(data,freq=FALSE,breaks=20) # Generate a histogram using proportions

# instead of counts.

lines(x,y) # Draw the normal curve.

Note how close the histogram and the normal distribution match. This suggests that the normal distribution would be a good approximation to the binomial (and normal calculations are a lot easier to do than binomial calculations).

**Your turn:**

Does the normal distribution always do a good job of approximating the binomial? Run the same code as above, except change the values for *n* and *p.* Specifically, explore values of *p* that are very close to 0 or 1. Also, check what happens when *n* is very small or rather large. Under what conditions does the approximation no longer work?

Now that we know that the normal distribution does a fairly good job of approximating the binomial distribution under certain conditions, let’s use our knowledge of R commands for the normal distribution from our previous lab to redo the earlier problems.

**Your turn:**

Suppose that a standard treatment for a disease is successful in 60% of all cases. If 200 patients are given the treatment, approximate the following using the normal distribution. (Hint: The commands that you will need are pnorm and qnorm.) For each of these problems, discuss how close the approximation from the normal distribution is to the actual value obtained using the appropriate command from the binomial distribution. Also, one of these problems cannot be solved using the normal approximation. Which one is it?

(i) The probability that exactly 120 patients are cured.

(ii) The probability that no more than 110 patients are cured.

(iii) The probability that between 115 and 125 patients are cured.

(iv) 75% of the time, at most how many patients are cured?

**The Continuity Correction for the Normal Approximation**

One of the problems with using the normal approximation for the binomial is the fact that the normal distribution is a continuous distribution and the binomial distribution is discrete. Thus, for the binomial distribution,  because there is actual mass or probability on the point *x*. On the other hand, for the normal distribution because any particular value has probability 0.

To correct for this difference, a small correction is suggested when using the normal approximation to the binomial. When I wish to find the probability of  for the binomial, I can approximate it using the normal distribution by finding . Similarly, if I want to find  for the binomial, I can approximate it using the normal distribution by finding . This accounts for the fact that there is actual mass on the point . Depending on whether or not I want to include this point, I look either 0.5 to the right or 0.5 to the left. (Imagine a histogram bar of width 1 centered on the point *x* which I’m either including or excluding.)

Let’s use the continuity correction to improve our normal approximation for two earlier problems.

**Your turn:**

Suppose that a standard treatment for a disease is successful in 60% of all cases. If 200 patients are given the treatment, approximate the following using the normal distribution, using the continuity correction. Discuss how the continuity correction affected your normal approximation compared to the actual value obtained using the appropriate command from the binomial distribution.

(i) The probability that no more than 110 patients are cured.

(ii) The probability that between 115 and 125 patients are cured.