STATISTICAL COMPUTING ACTIVITY

3: Some Discrete Distributions

Purpose: The purpose of this activity is to show you how to deal with some discrete random variables in R. You will want to know how to calculate point probabilities, cumulative probabilities, and quantiles. You will also want to be able to generate random variables based on the distribution. This last will help you better understand the variables and functions of the variables.

R Functions: We will see the following functions in R.

- sum
- sample
- dbinom
- rbinom
- dpois
- rpois
- ==
- >
- >=
- <
- <=
- !=
- &
- |
- plot
- points

PROCEDURE

PART 0: THE START

1. Start R.
2. Open a new script.
3. Since we are generating random values, we will not need to load external data.
4. You do still need to source the usual file. You will always need to do that:

```
source("http://rfs.kvasaheim.com/stat200.R")
```

PART I: THE CATEGORICAL RANDOM VARIABLE

This is the most flexible of discrete variables. One is able to approximate (at least) all discrete random variables with the categorical variable. A categorical random variable consists of a sample space and a corresponding set of probabilities.

Because of the need to be very flexible with this most general of discrete random variables, this part offers a glimpse into the general workings of R. It also offers a base from which all other discrete random variables can be examined.

1. Let us assume that our random variable has sample space S = {1, 4, 6} with corresponding probabilities p = {0.25, 0.40, 0.35}. To get R to know this, run the following:

```
x = c(1, 4, 6)
p = c(0.25, 0.40, 0.35)
```

2. This is all R needs to know about your random variable to understand some statistics. From this, we can calculate the expected value, μ or E[X]:

```
sum(p*x)
```

The variance, σ^2 or V[X]:

```
sum( p*(x-sum(p*x))^2 )
```

or

```
sum(p*x^2) - sum(p*x)^2
```

The probability of a 4 coming up, P[X = 4]

```
p[x==4]
```

or

```
sum(p[x==4])
```

The probability of more than a 3 coming up, P[X > 3]

```
sum(p[x>3])
```

The probability of at most a 4 coming up, P[X ≤ 4]

```
sum(p[x<=4])
```

The probability of a 3, 4, or 6 coming up

```
p[x==3] + p[x==4] + p[x==6]
```

or

```
sum(p[x==3], p[x==4], p[x==6])
```

or

```
sum(p[x==3 | x==4 | x==6])
```

3. What does the `sum()` function do? What does the `==` operator do? (As last week, that is two equal signs, not one.) What does the `|` operator do? (That is called the "pipe" operator, or the "bar" operator, or the "or" operator. It is located above the Enter key on your keyboard.)

PART IB: GRAPHING

4. We can also graph the probability mass function (pmf) of X:

```
plot(x,p)
```

or

```
plot(x,p, type="h")
```

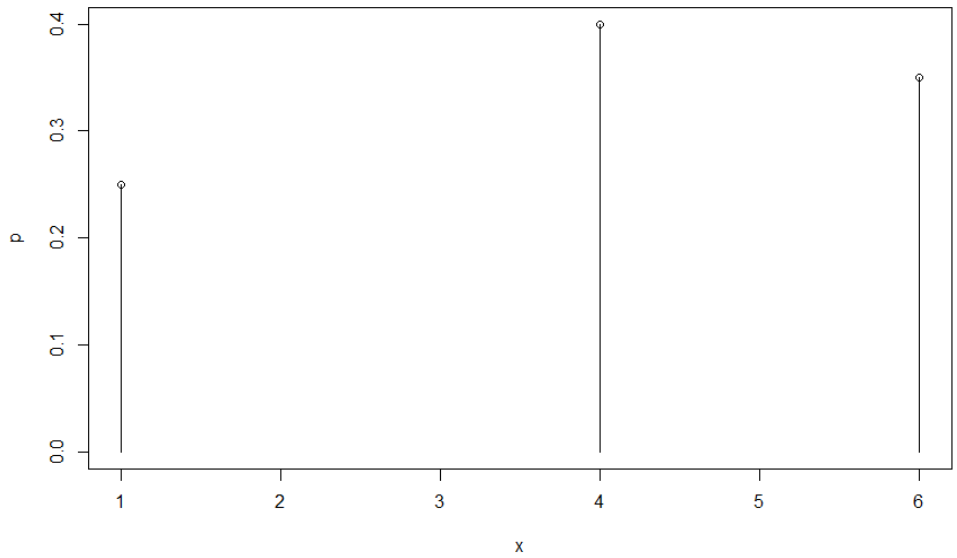
or

```
plot(x,p, type="h", ylim=c(0,1))
```

or

```
plot(x,p, type="h", ylim=c(0,1))
points(x,p)
```

5. How do those four code sets differ? Why might I use one over the others? What does **`ylim`** represent?



This a graphic I made of the probability mass function (pmf). I used a different **`ylim`** value. How does mine differ from yours?

PART IC: RANDOM NUMBERS

6. The next thing to do with this random variable is to generate values from it. This is accomplished with the **`sample()`** function. To generate 100 values from the X distribution described above, we run

```
sample(x, size=100, replace=TRUE, prob=p)
```

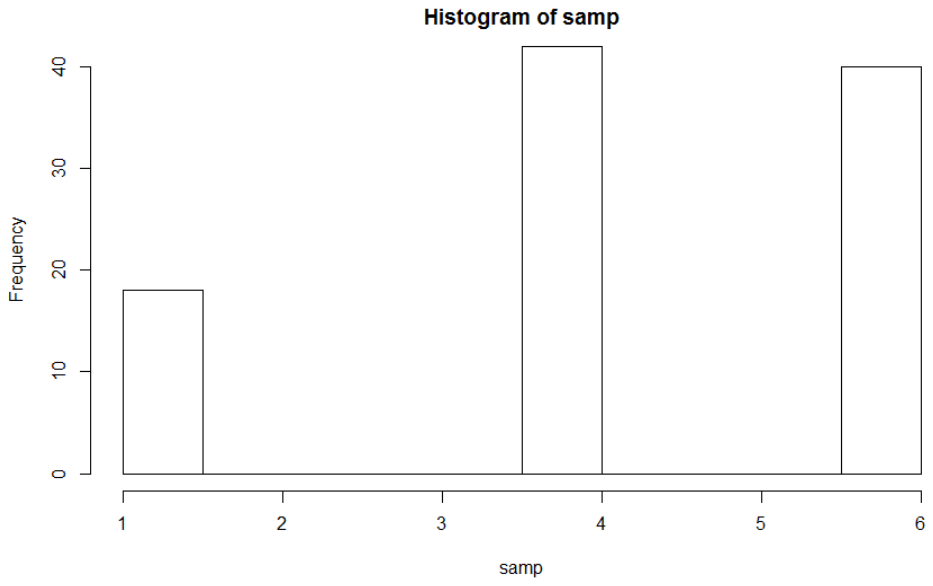
This gives a sample of size 100 from the X distribution. To produce a histogram of this sample, I could run

```
samp = sample(x, size=100, replace=TRUE, prob=p)
hist(samp)
```

Note that the histogram is similar to the pmf graphic we created previously. It is **not** the same, because we are working with a sample and not with the entire population. Were we working with the entire population, we would get a histogram that was ***identical*** to the pmf graphic.

By the way, your sample from the X distribution will be different from my sample from the X distribution. They will be similar, however. This is ***very*** important in understanding probability and statistics. As we are both pulling random samples from a population, the actual statistics calculated on those samples will differ. This is a consequence of random sampling.

Here is the histogram I got. Yours will probably be close to it in terms of bar heights.



PART II: THE BINOMIAL DISTRIBUTION

The Binomial distribution is defined by its two parameters. In the textbook, those are **n** and **p**. In R, they are **size** and **prob**. In this part, we will do for the Binomial distribution what we did for the categorical distribution above.

For this section, let us define X as a random variable that follows a Binomial distribution with 5 trials and a success probability of 0.10. In symbols, we are assuming X ~ Bin(5, 0.10).

1. Remember that the sample space for a Binomial random variable is S = {0, 1, 2, …, n}. Thus, we ***could*** define

   ```
   x = 0:5
   p = dbinom(x, size=5, prob=0.10)
   ```

   Note the new function, **dbinom**. This function calculates P[X = x] for the given value of x, where X ~ Bin(5, 0.10). So, if x = 1, **dbinom(1, size=5, prob=0.10)** calculates the probability P[X = 1].

   This is a d* function. All d* functions calculate P[X = x]. The differences are in what follows the **d**. Here, it is **binom**, which indicates X follows a Binomial distribution.

2. Now, we can calculate the expected value

   ```
   sum(p*x)
   ```

   The variance:

   ```
   sum(p*x^2) - sum(p*x)^2
   ```

   The probability of a 4 coming up

   ```
   p[x==4]
   ```

   or

   ```
   sum(p[x==4])
   ```

   The probability of more than 3 coming up

   ```
   sum(p[x>3])
   ```

   The probability of at most 4 coming up

   ```
   sum(p[x<=4])
   ```

   The probability of a 3, 4, or 6 coming up

   ```
   sum(p[x==3 | x==4 | x==6])
   ```

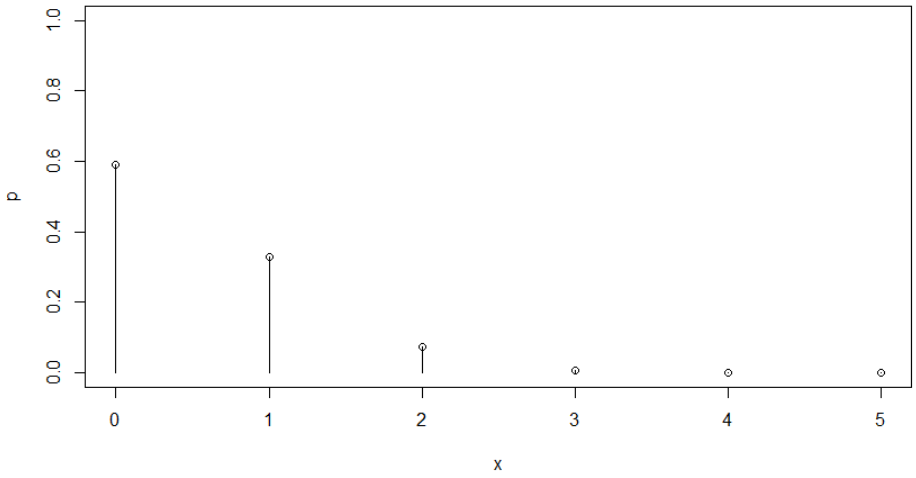
PART IIB: GRAPHING

3. We can also graph the probability mass function (pmf) of X:

```
plot(x,p, type="h", ylim=c(0,1))
points(x,p)
```

Those lines should look familiar. I copy-and-pasted them from Part Ib.

Here is a graphic of the pmf for the Binomial(5, 0.10) distribution. Note that yours will be the same as mine… except for its size and shape.



PART IIC: RANDOM NUMBERS

4. The next thing to do with this random variable is to generate values from it. We could do it as before using the **`sample()`** function.

```
samp = sample(x, size=100, replace=TRUE, prob=p)
```

But, it will be easier to do it using a built-in function

```
samp = rbinom(100, size=5, prob=0.10)
```

Why is this second method better? First, it explicitly tells the reader (and you) the distribution you are drawing your sample from. Second, it is shorter. Third, it is faster.

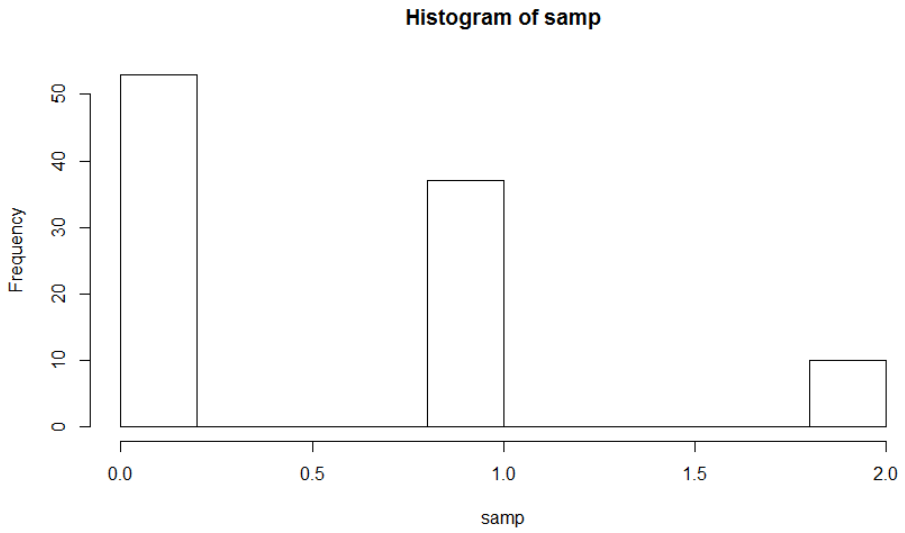
Thus, for the first reason, at least, you should use the second method. All of the important distributions have the r* form that generates the random sample. In all r* forms, the first value is the number of random values desired. Here, that is 100. So, after running this line, the variable **`samp`** will hold 100 values drawn from a Bin(5, 0.10) distribution.

5. To produce a histogram of this sample, I could now run

```
hist(samp)
```

Note that the histogram is similar to the pmf graphic we created previously. It is not the same, because we are working with a sample and not with the entire population. Were we working with the entire population, we would get a histogram that was ***identical*** to the pmf graphic.

By the way, your sample from the X distribution will be different from ***my*** sample from the X distribution. They will be similar, however. This is ***very*** important in understanding probability and statistics. As we are both pulling random samples from a population, the actual statistics calculated on those samples will differ… but usually not by much.



PART III: THE POISSON DISTRIBUTION

The Poisson distribution is defined by its single parameter: lambda, λ. In this part, we will do for the Poisson distribution what we did for the previous two distributions. For this section, let us define X as a random variable that follows a Poisson distribution with mean $\lambda = 9$. In symbols, we are assuming $X \sim \text{Pois}(\lambda=9)$.

1. Remember that the sample space for a Poisson random variable is S = {0, 1, 2, …}. Since the sample space is infinite, the previous methods for calculating the mean and variance will not work. We could ***estimate*** the mean and variance, however (if we forget that the mean and variance are both λ). This method for estimation is called Monte Carlo simulation.
2. The first step is to draw a large number of random values from the specified distribution.

   ```
   samp = rpois(1000000, lambda=9)
   ```

   Notice that we are using the r* form for the Poisson distribution. Refresh your memory as to what the r* form does.
3. Now, we can estimate the following statistics. The expected value and variance:

   ```
   mean(samp)
   var(samp)
   ```

4. We can also estimate probabilities. The probability of a 4 coming up is approximately:

   ```
   mean(samp==4)
   ```

   The probability of more than 3 coming up:

   ```
   mean(samp>3)
   ```

   The probability of at most 4 coming up:

   ```
   mean(samp<=4)
   ```

   The probability of a 3, 4, or 6 coming up:

   ```
   mean(samp==3 | samp==4 | samp==6)
   ```

The script on Moodle has other options for calculating cumulative probabilities. Feel free to see that script when it is posted. You will see the p* and q* functions next week. They make much more sense for continuous variables than for discrete.

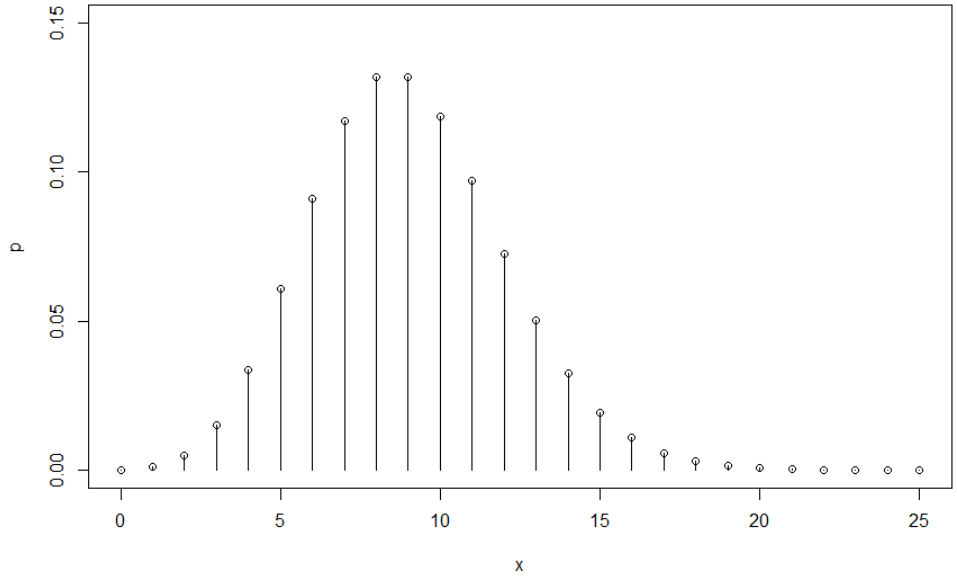
5. Note that the P[X >25] is very small. Thus, we could also estimate these by starting with

```
x = 0:25
p = dpois(x, lambda=9)
```

And using the methods of the previous two parts. This leads to the following pdf graphic for X:

```
plot(x,p, type="h", ylim=c(0,0.15)); points(x,p)
```

This is the pmf graphic I obtained.



6. And now for an extension and some insight into the power of Monte Carlo simulation. Let us know that X and Y are both random variables that follow Poisson distributions. Specifically, let us suppose X ~ Pois(λ=9) and Y ~ Pois(λ=11). What is the distribution of W = X + Y?

7. Here is the approximate distribution of X:

```
X = rpois(1e6, lambda=9)
```

   Here is the approximate distribution of Y:

```
Y = rpois(1e6, lambda=11)
```

   Here is the approximate distribution of W:

```
W=X+Y
```

8. Here is the approximate pmf of W:

```
hist(W)
```

9. Here is the (approximate) mean of W:

```
mean(W)
```

10. Here is the (approximate) variance of W:

```
var(W)
```

11. The true mean and variance of W is 20. How close were the approximations? How could we improve the precision of the estimates? How could we learn about the distribution of `V=X*Y`?