LAB 1. PROBABILITY

Goals

The goals of this lab are to (1) familiarize you with the most common probability distribution and teach you how to simulate data using these distributions; (2) teach you how to build new distributions; (3) understand the process for choosing a probability distribution given a data set; (3) understand how to use the methods of moments the derive parameters for probability distributions.

Probability Distributions in R

We will use R to generate random distributions. R is a vector driven language and is ideal for manipulation of random variables that fit different probability functions. Commands are specific to each distribution. You can look up syntax details in R help. R can do many things with these distributions: it will draw random deviates (r prefix), provide the probability of observing a given value (d prefix), calculate the cumulative density function (c or p prefix), and compute quantile functions (q prefix). In this handout, I have provided the R help text for some of the most common distributions you will encounter:

Discrete

Binomial	rbinom(n, size,p) n=number of random draws, size=no of trials, p=prob success of individual trial
Bernoulli	rbern(n,p) n=number of random draws. A binomial distribution with n=1. This distribution is in the package Rlab
Multinomial	rmultinom(n, size=n, prob=c(p1,p2,…pn)) n=number of random draws from a multinomial distribution with n trials, and probability of each outcome =p
Poisson	rpois(n, lambda) n=number of random draws lambda is the expected value of the distribution.
Negative binomial	rnbinom(n, size, prob, mu) Parameterized with prob & size or with mu & k(size). Cannot use prob and size at the same time.

The alternative parameterization (often used in ecology) is by the mean (mu), the predicted value of your scientific model), and size (k), the dispersion parameter, where prob = size/(size+mu). In this parameterization the variance is mu + mu^2/size. (See Lecture notes and R syntax).

Continuous

Normal	rnorm(n, mean=0, sd=1) 0 and 1 are default values
Lognormal	rlnorm(n, meanlog = 0, sdlog = 1) 0 and 1 are default values
Exponential	rexp(n, rate) rate 1 is default value
Weibull	rweibull(n, shape, scale = 1)
Gamma	rgamma(n, shape, rate, scale=1/rate) use either rate or scale
Beta	rbeta (n, shape1, shape2) shape parameters define the distribution

Get familiar with the distributions. Generate and plot random variables for the binomial, Poisson, negative binomial, exponential, gamma, lognormal, normal, and beta distributions. Get familiar with the mathematical formulation of these distributions using your lecture notes and the handout for this lab.

Using the commands above generate variables that have a given distribution. Understand the parameters that R requires because often you will need to estimate them when using likelihood.

Plot the random variables using a histogram (You can change from frequency to probability in the histogram by adding either freq=FALSE or prob=TRUE to the hist command).

It might be useful to plot different distributions on the same graph window to understand how different parameters of a given distribution affect the behavior of the distribution. The number of panels plotted in the same graph windows is given as par (mfrow=(#rows,#cols). For example, you could use par(mfrow=c(3,3) to plot 6 gamma distributions with variable shapes and rates, or you could plot the density of six gamma distribution in the same plot using different type of lines (lty="dashed" or "dotted", or "longdash"…)

Pick a value in the range generated by your choice of parameters and calculate its probability, its cumulative probability, and any quantile (5% and 95% for instance).

EXERCISES (Some materials from Bolker ms**)**

1. UNDERSTANDING DISTRUBUTION FUNCTIONS

1.1 For a binomial distribution with 10 trials and a success probability of p=0.2, generate 8 random values and calculate

- The probability of obtaining X=0, X=2, and X=5.
- The probability of obtaining 5 or more successes

You can also use R to test your distributions and make sure that they match with the theoretical expectations as they should. The results of a large number of draws should have the correct moments (mean and variance) and a histogram of those random effects should match up with the theoretical distribution. For example, draws from a binomial distribution with p = 0.2 and N=40 should have a mean of approximately $N*p$=8 and a variance of $N*p*(1-p)$=6.4 . Test this using R. Are the mean and the variance obtained using the method of moments on random draws close to the theoretical expectations?

We could repeat this exercise for a large number of draws (10,000). The steps in R are:
1. Pick 10,000 random deviates:

```
> x <-rbinom(10000, prob = 0.2, size = 12)
```

2. Tabulate the values, and divide by the number of samples to get a probability distribution:

```
> tx <- table(factor(x, levels = 0:12))/10000
```

(The levels command is necessary in this case because the probability of x = 12 with p = 0.2 and N = 12 is actually so low (8.7×10^{-5}) that there's a reasonable chance that a sample of 10,000 won't include any samples with 12 successes.)

3. Draw a barplot of the values, extending the y-limits a bit to make room for the theoretical values and saving the x locations at which the bars are drawn:

```
> b1 <-barplot(tx, ylim = c(0, 0.3), ylab = "Probability")
```

4. Add the theoretical values, plotting them at the same x-locations as the centers of the bars:

```
> points(b1, dbinom(0:12, prob = 0.2, size = 12), pch = 16)
```

1.2 Pick 10,000 negative binomial deviates with mu = 2, k = 0.5. Draw the histogram. Check that the mean and variance agree reasonably well with the theoretical values. Add points representing the theoretical distribution to the plot.

2. DETERMINING THE ERROR STRUCTURE IN YOUR DATA & METHOD OF MOMENTS

You will develop skill in the choice of distributions over time. Your choice should be based on the nature of your data, the processes that you believe generated the data, and the parameters that you want to estimate. The same data, counts of seeds, may require a binomial distribution if you are interested in understanding the process that determines the fate (survival) of individual seeds with the total number being of less interest, or Poisson, if you are interested in the total amount of seeds.

2.1. Let's look at the dataset "***sapling_growth.txt***". Plot the data (Growth). What distribution would you pick for these data? Why?

```
> growth<-read.table("sapling_growth.txt",header=TRUE)
```

The method of moments is a method of estimation of population parameters such as the mean, variance, median, etc. (which need not be moments), by equating sample moments with unobservable population moments (e.g., other shape parameters) and then solving those equations for the quantities to be estimated.

Let's go back to the sapling growth data. Assume that the data follow a gamma distribution? How would you find the parameters of the distribution? Once you find these parameters, can you plot the results? To do so, you will need to use the curve command which only takes an x argument.

```
>hist(Growth, prob=T)
>x<-seq(1,25,0.1)
>curve (dgamma(x, shape=?, scale=?), add=T)
```

Is there another distribution that can fit the sapling growth data?

2.2. Let's look at some data from Hilborn & Mangel (1997) (Table 4.3). The data is in the file "***HMtab43.txt***". These are the number of accidental bird bycatch (Captures) tabulated by the number of net hauls (e.g., the number of hauls that caught 0 birds and so on). Read the data into your workspace.

```
> tows<-read.table("HMtab43.txt",header=TRUE)
> attach(tows)  #so that you can work with Hauls and Captures directly.
```

Remember to detach!

Take a look at the data:

```
> barplot(Hauls, names=Captures, ylab="#Hauls", xlab="#Captures")

> totHauls <- sum(Hauls)
> Haulfreq <- Hauls/totHauls
> barplot(Haulfreq,names=Captures,ylab="Frequency",xlab="Captures")
```

What distribution would you use for these data? Why? Estimate the expected value and variance of this distribution. Calculate the parameter of the distribution using your estimates of the mean and the variance and the method of moments.

2.3 Plot your data (what you want to work on during the course). What distribution would you choose? Why? Be specific about how you evaluate the fit of the distribution to the data.

3. CREATING NEW DISTRIBUTIONS

(a) Zero-inflated distributions

The general formula for the probability distribution of a zero-inflated distribution, with an underlying distribution P(x) and a zero-inflation probability of pz, is (Recall seed predation data):

$\text{Prob}(0) = pz + (1 - pz)P(0)$ $\quad\quad$ $\text{Prob}(x > 0) = (1 - pz)P(x)$

So, for example, we could define a probability distribution for a zero-inflated negative binomial as follows:

```
> dzinbinom = function(x, mu, size, zprob) {
+ ifelse(x == 0, zprob + (1 - zprob) * dnbinom(0, mu = mu,
+ size = size), (1 - zprob) * dnbinom(x, mu = mu, size = size))
+ }
```

A random variable generator would look like this:

```
> rzinbinom = function(n, mu, size, zprob) {
+ ifelse(runif(n) < zprob, 0, rnbinom(n, mu = mu, size = size))
+ }
```

The command runif(n) picks n random values between 0 and 1; the ifelse command compares them with the value of zprob. If an individual value is less than zprob (which happens with probability zprob=pz), then the corresponding random number is zero; otherwise it is a value picked out of the appropriate negative binomial distribution.

4.1 *Write a density function and random deviate generator for*

- *a zero-inflated Poisson distribution*
- *a normal distribution with variance proportional to mean. Check graphically that these functions actually work.*

(B) Compound distributions

The key to compounding distributions in R is that the functions that generate random deviates can all take a vector of different parameters rather than a single parameter. For example, if you were simulating the number of hatchlings surviving (with individual probability 0.8) from a series of 8 clutches, all of size 10, you would say

```
> rbinom(8, size = 10, prob=0.8)
```

but if you had a series of clutches of different sizes, you could still pick all the random values at the same time:

```
> clutch_size = c(10, 9, 9, 12, 10, 10, 8, 11)
> rbinom(8, size = clutch_size, prob = 0.8)
```

Taking this a step farther, the clutch size itself could be a random variable:

```
> clutch_size = rpois(8, lambda = 10)
> rbinom(8, size = clutch_size, prob = 0.8)
```