Lab 2: Descriptive Statistics, Probability Distributions, and Sampling

The goals of this lab are to become familiar with discrete probability functions, specifically the binomial and Poisson distributions. The first part of the lab provides useful R functions, some of which we encountered in the last lab. The second part of the lab provides practice with exploratory statistics. The third part of the lab focuses on discrete probability functions. At the end of the lab, there are four problems to solve. Please show your code, type your answers to each problem, and plot graphs using R Markdown. *Submit your answers to the class Canvas site under the Assignments folder.*

More functions in R

In this lab, we introduce a few new R commands.

- `ls()` - provides a list of all the objects in the workspace
- `rm()` - removes individual objects from the workspace
- `attach()` - attaches a database to the R search path, making it possible to refer to a variable in the data frame by their names alone
- `detach()` - removes databases, usually a data frame that has been attached with `attach` or a package attached by `library` or `require`
- `is.na()` - logical function that identifies all NA's within a vector, returning either `TRUE` or `FALSE` for each value
- `dbinom()` - returns the height of the probability density function of the binomial distribution
- `pbinom()` - returns the cumulative density function of the binomial distribution; given a number, it computes the probability that a random number from a binomial distribution will be less than that number
- `rbinom()` - generates a random number from the binomial distribution defined by the probability (`prob`) of "successes" in a specified number (`size`) of trials
- `dpois()`, `ppois()`, `rpois()` - Same functions as above, but for the Poisson distribution defined by a single parameter, `lambda`, which specifies the mean and variance of the distribution

Download the database `AfrPlots.csv` from Canvas and read it into R. You will need to set a working directory to tell R where you're storing your data and where you want to write results to. You can do this using the `setwd()` function, which takes a directory path as an argument. For example: `setwd("C:/Users/...")`. Your TAs can show you how to do this.

```
afdat <- read.csv("AfrPlots.csv", header = T)
```

The database consists of tree plot data from Africa. In 30 1-ha plots, all the trees $\geq$ 10 cm dbh (diameter-at-breast height) were measured, and from this data the biomass, basal area, and other statistics were calculated for each plot. Note also that there were two census periods: the initial census and then a census four years later.

Sometimes you may want to remove variables from the workspace, particularly when you have created multiple variables and might confuse them. Variables can be removed one at a time using `rm()` as below. Or, multiple variables can be removed with `rm(list=ls(your objects here))`. You can remove *all* the objects in your workspace by leaving the list argument blank: `rm(list=ls())`.

```r
var0 <- seq(1:5)
 var0
  rm(var0)
```

In R, missing values are represented by the symbol NA (not available). Some functions, like `mean()` have arguments to remove NA's from the operation. Note what happens if you set `na.rm` to `TRUE`. `na.rm` is assumed to be FALSE if it is not explicitly set.

```r
var1 <- c(0, 4, NA, 2, NA, 7)
mean(var1)
mean(var1, na.rm = TRUE)
```

Other functions may not have built-in arguments to remove NA's, or you may have a different reason for removing them from a vector. To do this, employ the `is.na()` function. `is.na` is a logical function, meaning that it will return a TRUE or FALSE response for each value in the vector.

```r
is.na(var1)
```

To actually remove the NA's, define `var2` as all the values of `var1` where `is.na` is not (`!`) equal to TRUE. Note that the two lines below do the same thing.

```r
var2 <- var1[!is.na(var1)]

var2a <- var1[is.na(var1) == FALSE]
```

To code an observation as missing, you can simply assign it NA. For example, let's replace the value of 7 in `var2` with NA. The double equal sign (==) is a logical operator that means *exactly equal to*. This line of code says replace all values in `var2` that are exactly equal to 7 with `NA`.

```r
var2[var2 == 7] <- NA
```

Exploratory data analysis in R

Measures of location and spread

We have talked about measures of location and spread in lecture. There are several measures of Central Tendency, including the median, mean, trimmed mean, harmonic mean, and geometric mean. Similarly, there are several measures of spread, including variance, standard deviation, and coefficient of variation.

The variance of a sample of data can be calculated as:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2$$

The standard deviation is the square root of the variance. And, the coefficient of variation is the ratio of the standard deviation to the mean, reported as a percentage.

Use R's functions (`var`, `sd`, `mean`) to find the variance, standard deviation, and coefficient of variation of `BasalArea`.

Discrete probability in R

To explore discrete probability distributions in R, we first need to generate some random numbers.

```
sample(x = 1:100, size = 10)
```

Binomial distribution

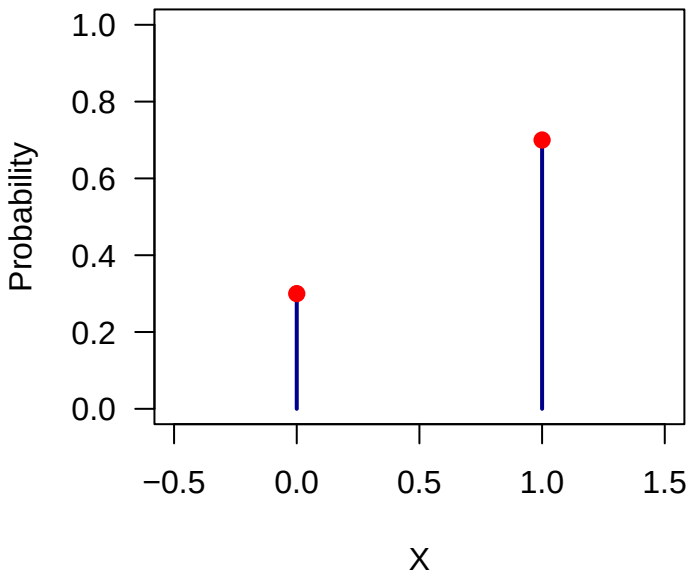
Remember from lecture that a Bernoulli trial is an experiment with two possible outcomes, like flipping a coin or recording whether a species is present or absent. The outcome is referred to as a success or failure. Most often in environmental sciences, we are interested in what happens over a sequence of Bernoulli trials. For our random variable, X, we will count the number of successes: the number of times out of 10 that we flip "heads". The random variable, X, is a binomial random variable.

To simulate a Bernoulli trial, we can write:

```
set.seed(1001)
 dbinom(1, size=1, prob=0.5)
```

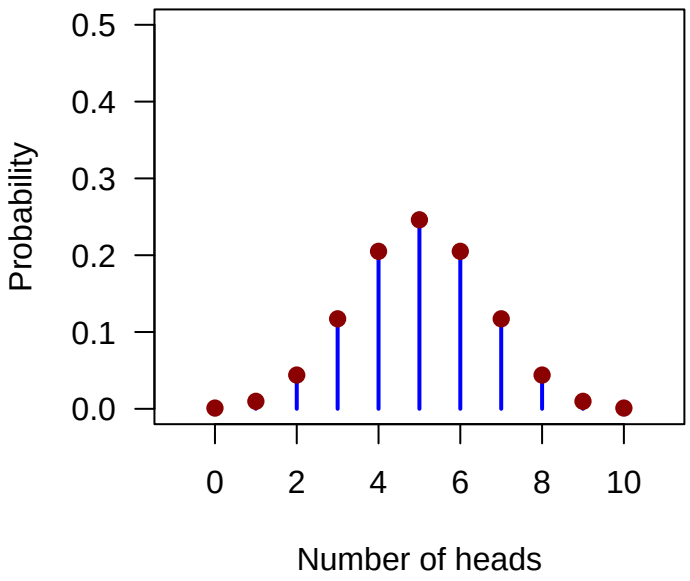
Here `dbinom` is the density of the binomial distribution. For discrete data (not continuous data), it returns the probability of a specific value of X. How is this different from the results we obtained using `rbinom` in the previous lab? Because the values of the distribution are discrete, the probability mass function will not be continuous like a normal distribution, so we visualize it with a sequence of spikes. Let's represent a Bernoulli distribution where the probability of a 0 is 0.3 and the probability of getting a 1 is 0.7:

```
x <- c(0,1)
 p <- c(0.3, 0.7)

plot(x, p, type = "h", las = 1, xlim = c(-0.5, 1.5),
      ylim = c(0, 1), lwd = 2, col = "darkblue",
       ylab = "Probability", xlab = "X")

points(x, p, pch = 16, cex = 1.2, col = "red")
```



Now, let's toss a 'fair' coin ten times. What is the probability of obtaining 3 or fewer heads? First, let's visualize the distribution. Here n is the number of flips; p is the probability of getting a heads, x is all the possible outcomes, and pr determines the probability of getting 0 to 10 heads out of 10 flips of a coin.

```
n <- 10
 p <- 1/2
  x <- 0:10
   pr <- dbinom(x, size = n, prob = p)

 plot(x, pr, type="h",xlim = c(-1, 11), ylim = c(0, 0.5),
     las = 1, lwd = 2, col = "blue",
      ylab = "Probability", xlab = "Number of heads")
points(x, pr, pch = 16, cex = 1.2, col = "dark red")
```



One approach is to realize that the probability of obtaining 3 or fewer heads is the sum of the individual probabilities. Add up the probabilities of obtaining 0, 1, 2, or 3, heads.

```
sum(pr[1:4])
```

which is the same as

```
sum(dbinom(0:3, size = 10, prob = 0.5))
```

So what is the `dbinom()` function doing? It is calculating the binomial probability, using the equation:

$$P(X) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

.

{{To Do}}

Write the equation for binomial probability in R. Use it to calculate the probability of obtaining 3 or fewer heads in ten flips of a fair coin. How does your answer compare to the probability found using `dbinom()`?

Poisson probabilities

Like the binomial distribution, the Poisson distribution is a discrete probability function. However, instead of modeling successes and failures, it models counts of

outcomes. If lambda, λ, is the mean occurrence of a count, then the probability of having x occurrences is given as:

$$P\Big(x\Big) = \frac{\lambda^x}{x!}e^{-\lambda}$$

where $x = 0, 1, 2, 3, ...$

Instead of using `dbinom` to find the probability of an occurrence, we replace it with `dpois` for the Poisson distribution. For example, if the mean number of lightening strikes on top of Mt. Baldy is 3 per year, then the probability of the mountain only being struck once in 2014 is found by:

```
  dpois(x = 1, lambda = 3)
```

What is the probability of the mountain being struck 3 times? 10 times? 30 times?

Normal probabilities

Unlike the binomial and Poisson, the normal distribution is a continuous probabilty function. It can be used to model numbers that can take on infinitely many, uncountable values. The probability ***density*** function is give as:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

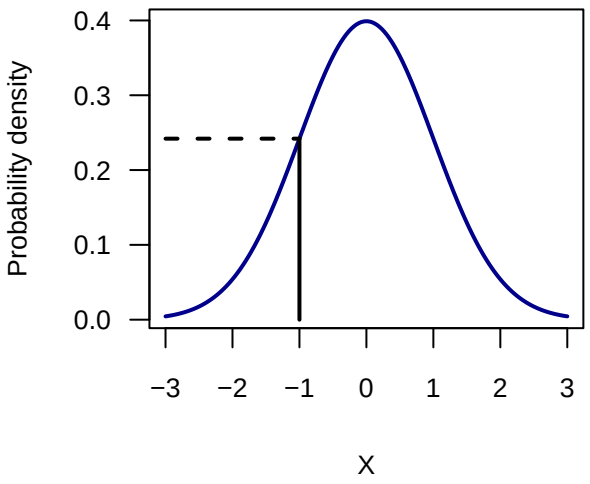
where x is any value of a continuous random variable, σ is the standard deviation and μ is the population mean.

Plugging in a value of x and the mean and standard deviation, we can calculate a height f(x) of the density function. For example, we could use `dnorm` similar to `dbinom` and `dpois` before:

```
dnorm(x = -1, mean = 0, sd = 1)
```

```
  dnorm(x = -1, mean = 0, sd = 1)
```

```
x <- seq(-3, 3, length = 1000)
 y <- dnorm(x, mean = 0, sd = 1)
  plot(x, y, type = "l", lwd = 2, col = "darkblue", xlab = "X",
       ylab = "Probability density", las = 1, cex.axis = 0.8, cex.lab = 0.8)
   segments(-1, 0, -1, dnorm(-1, 0, 1), lwd = 2)
    segments(-3, dnorm(-1, 0, 1), -1, dnorm(-1, 0, 1), lty = 2, lwd = 2)
```




```
   by.hand <- 1/(1*sqrt(2*pi))*exp(-(-1-0)^2/(2*1^2))
```

This statement provides the density of the curve at an x-value of -1, but does not correspond directly to the probability. Again, this is the density or height of the curve at that point, not its probability. We could estimate the probability of getting a -1 by finding the area under the curve between -0.999 and -1.001:

```
dnorm(x = -0.99999, mean = 0, sd = 1) - dnorm(x = -1.00001, mean = 0, sd = 1)
```

This probability is very small, and would get smaller and smaller if we tried to estimate it with more precision. On continuous distributions we find probabilities for area under the curve, not for individual numbers.

But that is okay because we usually want to calculate bigger areas under the Normal curve. For example, we ask questions like 'what is the probability of getting a value smaller than -1'. This can be determined by using the *cumulative probability* function `pnorm`.

```
pnorm(q = -1, mean = 0, sd = 1)
```

This shaded area represents the cumulative probability between -4 and -1 for the distribution.

```
x1 <- seq(-4, 4, length = 1000)
xy.dat <- data.frame(cbind(x = seq(-4, 4, length = 1000),
                           y = dnorm(x1, mean = 0, sd = 1)))

 q <- ggplot(data = xy.dat, aes(x, y)) +
```

```
       geom_line(stat = 'identity', col = dukeblue) +
        ylab("Probability density") + xlab("X") +
         xlim(-4, 4) +
          annotate("text", x = 3, y = 0.38,
                   label = TeX("$X \\sim \\mathrm{N}(0, 1)")) +
         theme_bw()

q1 <- q + stat_function(fun = dnorm, xlim = c(-4, -1), geom = "area",
                        fill = "#001A57", alpha = 0.7) +
            annotate("text", x = -3, y = 0.12,
                     label = TeX("P(X$\\leq -1) = "), size = 3) +
            annotate("text", x = -3, y = 0.1,
                     label = "pnorm(q = -1,", size = 3) +
            annotate("text", x = -3, y = 0.08,
                     label = " mean = 0, sd = 1)", size = 3)

q1
```

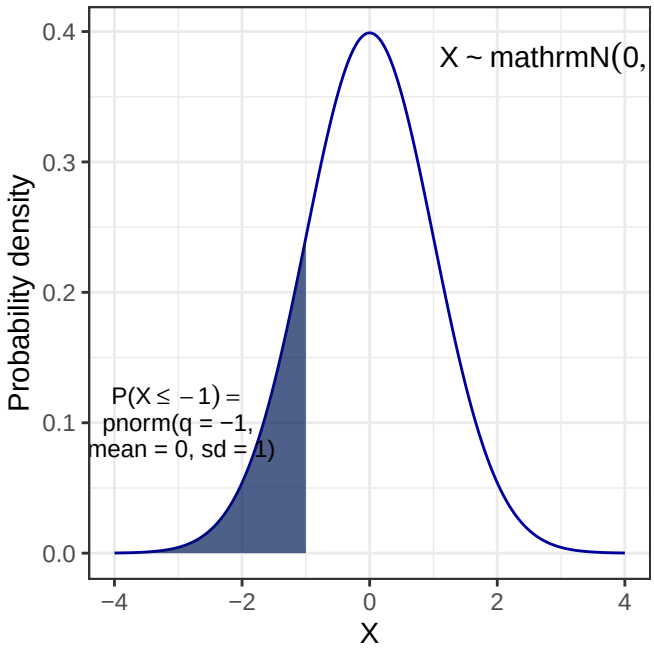


Figure 1: Normal distribution with shading demonstrating that the probability of getting a number less than -1 is 15.9%.

Sampling Distribution

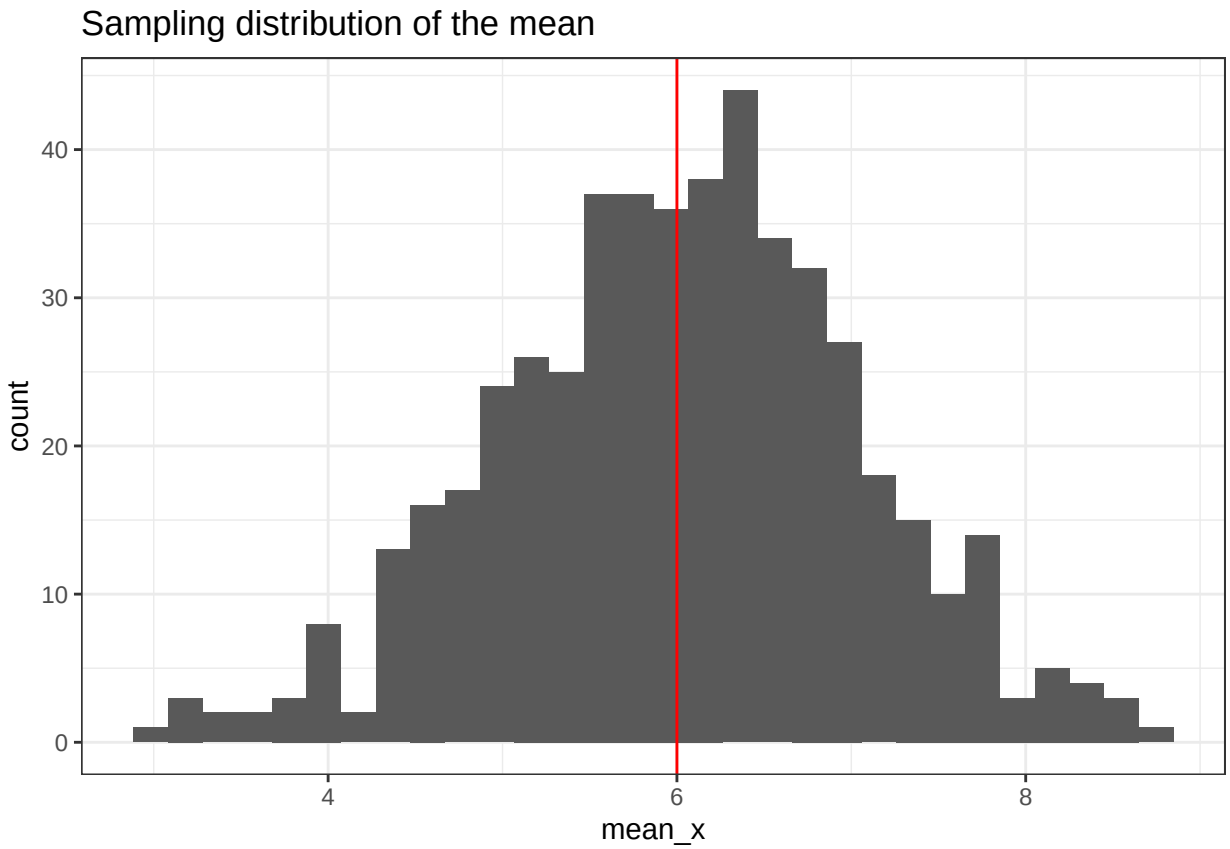
In this class, we have discussed how statistics are: 1) *random variables* and 2) *have probability distributions*. We discussed that the randomness of a given statistic (e.g. a mean) is because they come from a sample, and a different sample would yield a different result. As we've seen in lecture, this randomness is governed by a

probability distribution. The distribution of a statistic is known as the ***sampling distribution***. To see how this works, let's do the same experiment repeatedly in R and look at the sampling distribution of the mean. Our experiment will consist of taking a random sample of size $n = 100$ from a normal population and we will compute the mean. Let's suppose that the mean of the population is 6, and the population standard deviation is 2.

In order to accomplish this, we will need to:

1) Create a vector to store the mean we calculate from each experiment
2) Write a `for loop` in R to automate each experiment
3) Conduct the experiment within the for loop and store the results
4) Plot the results

```
### Step 1: Create storage

# Define number of experiments
nexp <- 1000
sample_means <- matrix(NA,nrow=nexp)

### Step 2-3: Write for loop, conduct experiment, and store results
for (i in 1:nexp){
x <- rnorm(n=500,mean=6,sd=1)
sample_means[i] <- mean(x)
}
### Step 4: Plot the result
x <- data.frame(mean_x = x)
gg <- ggplot(x,aes(mean_x))+
        geom_histogram()+
        geom_vline(xintercept=6,color="red")+
        theme_bw()+
        ggtitle("Sampling distribution of the mean")
gg
```



As you can (hopefully) see, the sampling distribution of our repeated exeriment follows a normal distribution that has a mean of 6. What this means is that we can collect a sample from a population and estimate properties of the probability distribution that gave rise to the data. In other words, the expected value (i.e. the mean) of our sample follows a distribution that is centered around the true population mean. Practically speaking, this is why we believe that a statistic (our sample mean) is a good estimate of a parameter (our population mean). You will show this in this week's lab.

Problems (3 Points)

Problem 1 (0.5 points)

Summarize the data for mean growth `MeanGr` of the plots in `afdat`. Mean growth is the difference in average DBH for each plot between the first tree census and the second census; thus, there are only data for `MeanGr` during the second census. Include the following in your analysis:

a. **Plot a histogram and boxplot of `MeanGr`.**
b. **Report the mean, median, variance, standard deviation and coefficient of variation of the `MeanGr`.**

Problem 2 (0.5 points)

Say you flip a fair coin 20 times. What is the probability of obtaining 5 or fewer heads or more than 5 heads? Show how you would answer this question using the `dbinom` function and calculating it with your binomial equation from above.

Problem 3 (0.5 points)

Suppose there are 20 multiple-choice questions on a Stats quiz. Each question has four possible answers, only one of which is correct. Find the probability of answering 17 or more answers correctly if you attempt to answer them at random.

Problem 4 (0.5 points)

If there are 4 butterflies feeding at a flower per hour on average, find the probability of having 9 or more butterflies feeding at the flower in a particular hour. Although the possible maximum count of butterflies could be much greater, let's limit the maximum number of butterflies to be 13. Do this by (1) writing out the Poisson formula in R, and (2) by using `dpois()`. Please also graph the probability distribution.

Problem 5 (1 point)

You are going to conduct many experiments similar to the one described above under the 'Sampling Distribution' section.

a. **Suppose you are interested in knowing the spatial pattern of a species' distribution (i.e. presence or absence) across an area that is 100km x 100km. You decide to divide the study area into 1km x 1km square grid cells. The 10,000 resulting cells are far too many to realistically sample given the budget that you have to conduct your study, but you have the ability to sample 10% of them. Further, from a prior census, you know that the probability that the species is present is $p = 0.30$. Conduct 1,000 experiments in which you randomly generate a data point that indicates whether or not the species is present in a given grid cell, and estimate the proportion of cells where the**

species is present. Store the resulting estimates and plot them as seen in the example above. Comment on the resulting plot. What does it look like? What distribution does it appear to follow?

b. Now suppose that, across the same area, you are interested in understanding the number of trees in each grid cell. Again, previous research has indicated that the true expected number of trees in each grid cell is 5. Conduct a similar experiment to the one above. Here, you are interested in the average number of trees per grid cell. Store the resulting estimates and plot them as seen in the example above. Again, what does the distribution look like? What distribution does it appear to follow?

c. Now suppose that you are interested in understanding the diameter at breast height (DBH) in feet of the tree population in this area. Again, suppose you know from prior research that the mean diameter is 1 foot, with a standard deviation of 2 inches. Repeat the procedures of a and b (adapted for this experiment), store the estimates, and plot them as before. What does the distribution look like? What distribution does it appear to follow?