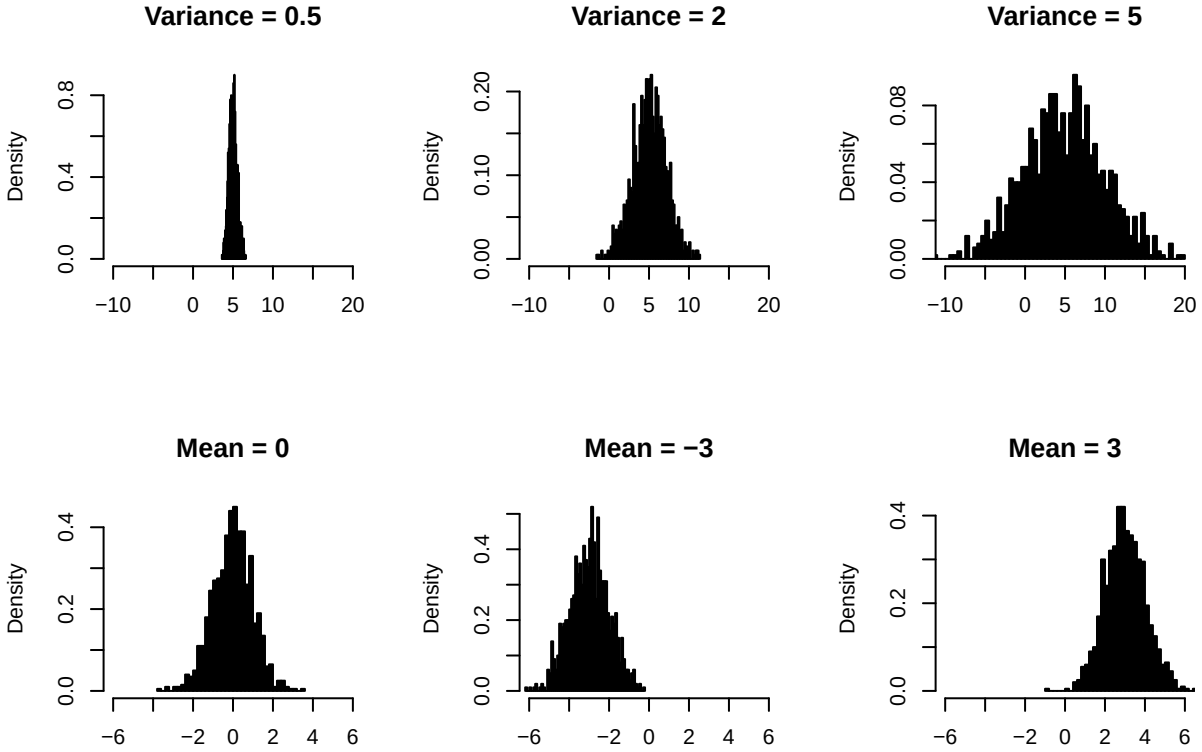
Lecture 1 - Probability Distributions

Sam Mason

1/28/2020

Part 1: Exploring the Shape of Continuous Distributions

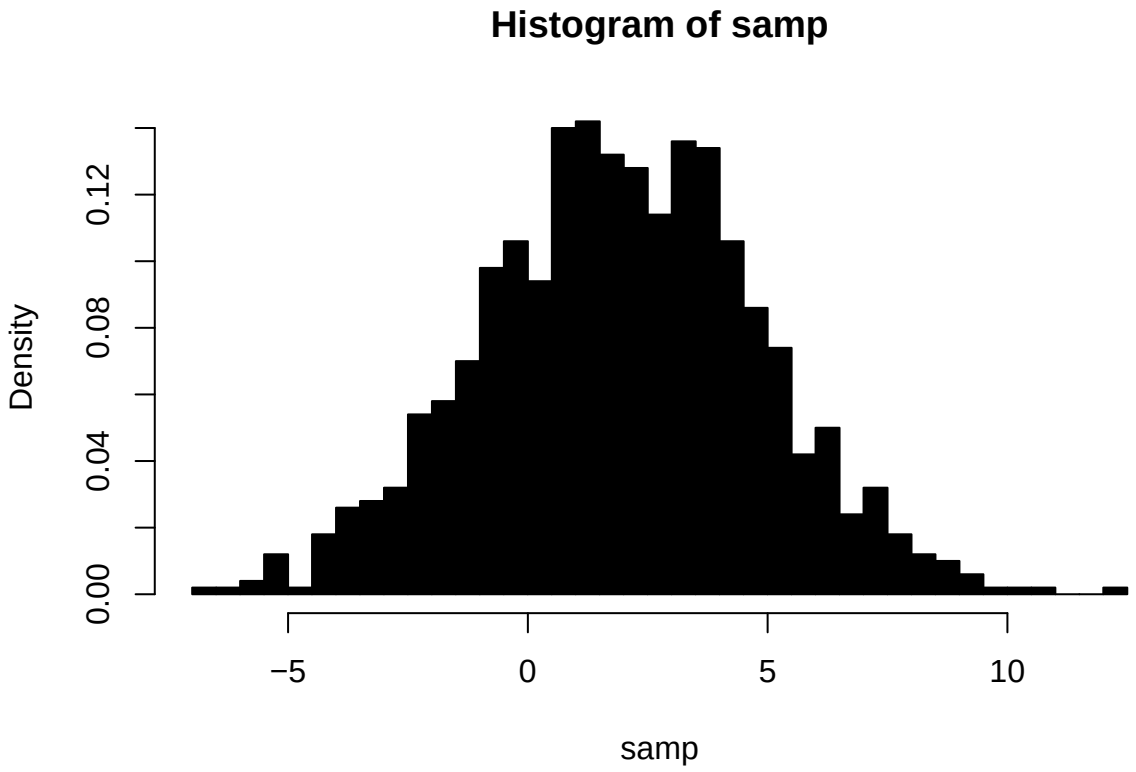
The shape of a normal distribution is given by its first and second moments, the mean μ and variance σ^2. In this figure below, the first row of distributions holds μ constant at 5, and allows σ^2 to vary. The second row features normal distributions with constant σ^2 and variable μ. By definition, the normal distribution has skewness and kurtosis equal to 0.



The normal distribution is appropriate for continuous variables, those which can take on the values of all real numbers (of which there are infinitely many) within a given interval. We call this interval the **support**, and say things things like *the normal distribution has support in the set of all real numbers*. Let's look at another continuous distribution useful to biologists, the beta distribution with support along the interval (0, 1). Unlike the normal distribution, the beta distribution has parameters α and β, which do not explicitly reflect the first and second moments of the distribution. For example, the first moment of a beta distribution is given by the ratio $\frac{\alpha}{\alpha+\beta}$. Let's prove this to ourselves! I'll show you an R proof using the normal distribution, which will be mathemtically unfulfilling, but syntactically helpful.

```
# First we generate a large random sample a normal distribution using the rnorm() function
samp <- rnorm(1000, 2, 3)

# Use the help call ? to determine the parameters defined by each of the three arguments.
# Let's plot this data using the hist() function to visualize it.
hist(samp, breaks = 50, freq = FALSE, col = 'black')
```




```
# What is the "breaks" argument doing here? Looks like the first moment is right where
# it's supposed to be around two. Let's confirm by using R's mean() function.
mean(samp)
```

```
## [1] 1.930893
```

```
# Pretty close! And we'll get closer for larger n. The first moment of a normal
# distribution is equal to it's mean parameter. So here's the big question: is
# two equal to two? Let's submit our query humbly before R:
2 == 2
```

```
## [1] TRUE
```

```
# R HAS SPOKEN! This is an example of a logical expression, and can be evaluated by
# R as either "TRUE" or "FALSE"
```

Alright, you're up! Here are the steps:

1. Create a vector of 1000 random draws from a beta distribution with α and β parameters of your choosing. **HINT:** make sure you're assigning this vector to a new object so you can use it later.
2. Calculate the mean using the `mean()` function.
3. Calculate the mean using the α and β parameters defined in step 1.

Part II: Discrete Distributions and the Probability Mass Function (PMF)

Let's take a look at a discrete distribution, the Bernoulli distribution, with support in the set $\{0, 1\}$. To be clear, Bernoulli data is binomial, and can only take on values of zero and one. Can you think of an example of real-world binomial data? Perhaps the presence (one) or absence (zero) of a particular butterfly species in a sample of salt marsh patches? Let's dig into this concept of binomial data a bit more. Because the data can only exist in two states (zero or one), we say that binomial states are *disjoint*, that is, knowledege of one state implies perfect knowledge of the other. If a patch of salt marsh is occupied by the butterfly species (one), it cannot possibly be unoccupied (zero). Because a patch can only exist as occupied or unoccupied,

if the probability of occupancy is Ψ, then the probability of unoccupancy is $1 - \Psi$. Because of this, when defining the shape of a Bernoulli distribution, we need only specify the probability of one state or the other (conventionally the *probability of success*, $P(1)$).

We can describe the shape of a Bernoulli distribution using the appropriate probability mass function (PMF) presented below:

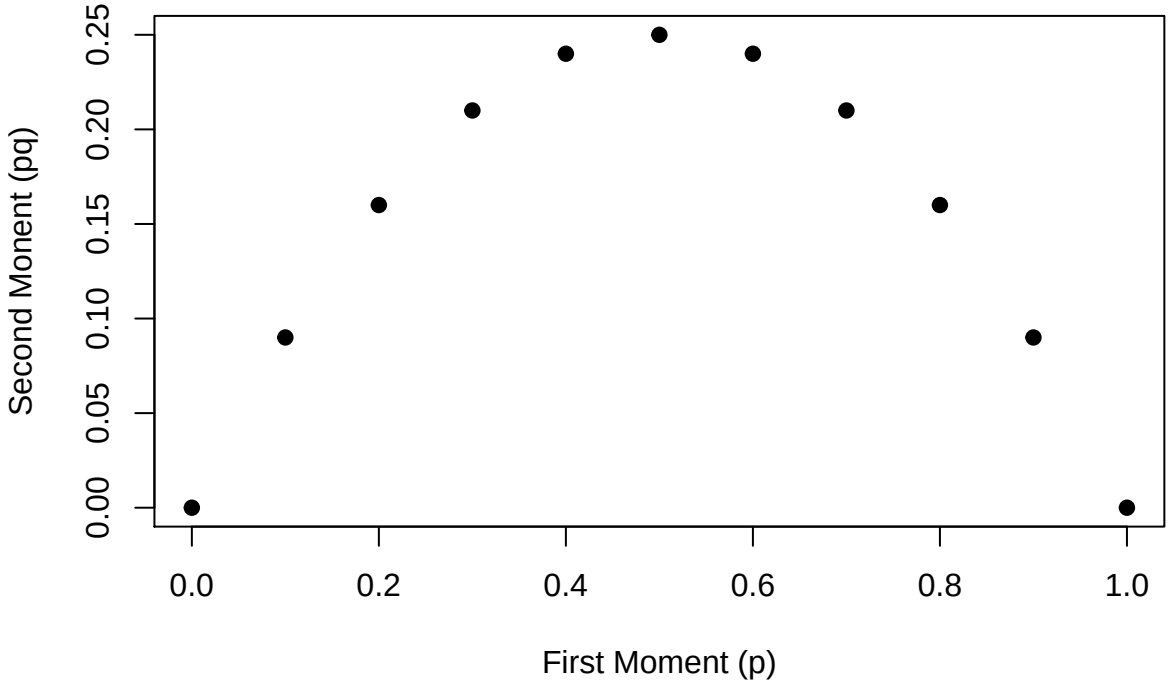
$$PMF = f(k) = \begin{cases} q = 1 - p & if \;\; k = 0 \\ p & if \;\; k = 1 \end{cases}$$

where p is the probability of success ($k = 1$) and q is the probability of failure ($k = 0$). The PMF, $f(k)$ takes some value $k \in 0, 1$ and returns the probability of that value. This is important to note: the PMF of a discrete distribution calculates the *probability* of a discrete value existing in the support of the distribution. In the next section we'll introduce the probability *density* function (PDF) associated with continuous distributions, and compare it to the concept of the PMF.

The first and second moments of the Bernoulli distribution are given by p and pq, respectively. Let's exmaine how these two moments relate to one another. Please create a scatter plot where the first moment is on the x-axis, and the second moment is on the y-axis. Here's how we'll tackle it:

1. Create a vector called "m1" and assign it all values from 0 to 1, incremented by 0.1 using the `seq()` function in R.
2. Create a new vector, "q" equal to $1 - m1$. In R, we can apply arithmetic operators, like substraction, to each element of a vector by simply using the vector in an arithmetic expression. For example, if I wanted to add five to each element of "m1" I could execute the code `m1 + 5`.
3. Create a third vector, "m2", equal to the product of "m1" and "q".
4. Use the plot() function with default arguments to create your scatterplot.

Here's what my plot looks like after I spruced up the formatting a bit:



Part III: Probability Mass vs. Density

We'll start by introducing a new continuous distribution, the gamma distribution, with support $x \in (0, \infty)$, and parameters α (shape) and β (rate). As with any new distribution, let's take a moment to explore it. Please make a 2x3 (row by column) matrix of gamma distributions, holding shape constant in the first row, and rate constant in the second row (similar to what I produced above for the normal distribution). The specific parameter values are up to you. We'll begin to address graphics in R more directly in this exercise. Okay, let's dive in:

1. When composing graphics in R, we often want to set some formatting parameters ahead of time. The function `par()` does the heavy lifting in this regard. This is a very powerful function, but for now we'll use it to create our plotting matrix and define subplot margins. Begin your plotting with the following line: `par(mfrow=c(2, 3), mar = c(3, 3, 3, 3))`. The "mfrow" argument says to R: *I want six subplots, organized as two rows of three plots, and I want new plots to be placed in the matrix by row (i.e., the first plot gets placed in cell (1, 1), the second plot in cell (1, 2), etc.)*. The "mar" argument defines how large the margins around each subplot should be. The margins are presented as a list in the order (*bottom, left, top, right*).
2. Create a vector containing 1000 random draws from a gamma distribution with parameters of your choosing.
3. Using the `hist()` function, plot your first distribution. Use the help call ? to identify the arguments that hist() can take. Some key formatting include: **"main"**, specifying the title of the plot (must put title in quotation marks); **"breaks"**, the number of bins included in your histogram; **"freq"**, a logical argument accepting either TRUE or FALSE (TRUE: plots count on the y-axis, FALSE: plots probability density); **"xlab"**, specifying the x-axis label (quotations required); **"ylab"**, specifying the y-axis label (quotations required); **"xlim"**, determines the extent of the domain (specified as a concated list, for example, `c(0, 10)`); **"col"**, the color of the bars.

HINT: in order to adequately compare distributions, make sure your "xlim" argument is consistent among relevant plots.
HINT: if you mess up a subplot, you'll need to clear the graphics memory using the broom icon in the "Plots" window of RStudio, and then rerun the code, including the `par()` line.