# Distribution Functions in R

*Hassan Nasif*

## Overview

**R** has a family of functions that allow you to analyze the properties of various known probability distributions easily. In this explanation, we will focus on this family of functions for the normal distribution, but note that these commands analogously exist for most distributions, including (but not limited to) the Exponential, Binomial, Poisson, and T distributions. You may simply substitute the suffix _norm with the appropriate abbreviation of the desired distribution. Please refer to this page for a full list of the probability distributions included in base **R** and their abbreviations.

The four functions we will go over are **dnorm**, **pnorm**, **qnorm**, and **rnorm**.

### dnorm

This function returns the value corresponding to the probability *density (mass)* function for continuous (discrete) distributions. For the normal distribution, it returns the y-value on the bell curve when given a value for $x$ and parameters $\mu$ and $\sigma$. In other words, it plugs x into the following density function for the normal distribution, given values for $\mu$ and $\sigma$:

$$f_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

As a result, **dnorm** has 3 main inputs: x, mean, sd. x must be an array of numerics corresponding to the values you want plugged into the density function. In discrete distributions, x must be an array of integers. Mean corresponds to $\mu$ above, and sd corresponds to $\sigma$ above, both numerics. Note that,if sd is less than or equal to 0, you will get an error. In R, the default values for mean and sd are 0 and 1,respectively, in all of the *norm* functions. This corresponds to the standard normal distribution. The output for **dnorm** is an array of the same size/shape as the input $x$.

Let's find the density value for $x = 0$ in the standard normal distribution:

$$f_{0,1}(0) = \textbf{dnorm}(x = 0, \text{mean} = 0, \text{sd} = 1) \tag{1}$$
$$= 0.39894 \tag{2}$$

### pnorm

This function returns the value of the *cumulative distribution* function for a probability distribution. For continuous distributions, this is the definite integral taken from the minimum of the density function's support to a point $x$. For discrete distributions, the integral is replaced with a summation. More specifically, for the normal distribution, this is:

$$F_{\mu,\sigma}(x) = \int_{-\infty}^{x} f_{\mu,\sigma}(x)dx \tag{3}$$

As you may remember from calculus, this computes the area under the curve of the density function $f$, and this area is found from the minimum up until the point $x$. We refer to this area as the quantile for the point $x$. For some intuition, a return of 0 indicates that input corresponds to the minimum of the distribution. Similarly, a return of 1 indicates that the input corresponds to the maximum of the distribution.

Similar to **dnorm**, the three inputs to **pnorm** are an array of numerics $x$ (integers for discrete distributions), the mean ($\mu$), and the sd ($\sigma$). The output is similarly an array of the same shape/size of the input $x$, where the values are always numerics between 0 and 1.

Let's find the quantile value for $x = 0$ in the standard normal distribution:

$$F_{0,1}(0) = \textbf{pnorm}(0, \text{mean} = 0, \text{sd} = 1) \tag{4}$$
$$= 0.5 \tag{5}$$

The value 0.5 means that 0 is the median of the standard normal distribution.

**qnorm**

This returns the value of the *quantile* function at a given quantile value. This can be thought of as the inverse of the **pnorm** function, where you have the quantile value and you want to know what value of $x$ corresponds to that quantile. The input is an array of quantile values (numerics between 0 and 1) and the output is an array of numerics (integers for discrete distributions) of the same size/shape as the input.

Let's say we didn't know what the median of the standard normal distribution was. We can use qnorm to help us find it:

$$F^{-1}_{0,1}(0.5) = \textbf{qnorm}(0.5, \text{mean} = 0, \text{sd} = 1) \tag{6}$$
$$= 0 \tag{7}$$

We've confirmed that 0 is indeed the median of the standard normal distribution, but we already knew that from our previous **pnorm** example. Note that **pnorm** and **qnorm** are great "by **R**" substitutes for the tables you commonly use when working on your homeworks!

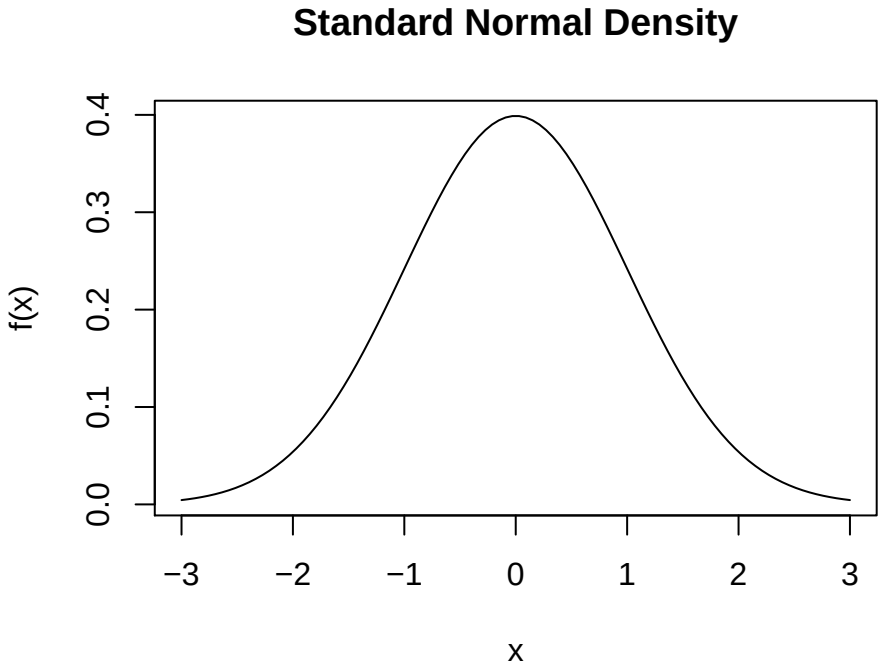
**rnorm**

This is the most unique of the functions. **rnorm** generates a random sample from a normal distribution. The mean and sd inputs remain the same, but the primary input $n$ is an integer that represents the size of the random sample desired. The output is thus an array of length $n$.

This function generates random samples from the normal distribution using a technique called Markov Chain Monte Carlo. For a reasonably large $n$, if you were to produce the histogram of $x$, it would look like the shape of the normal distribution (density) curve. That is what we mean when we say "take a sample of size $n$ from a normal distribution."

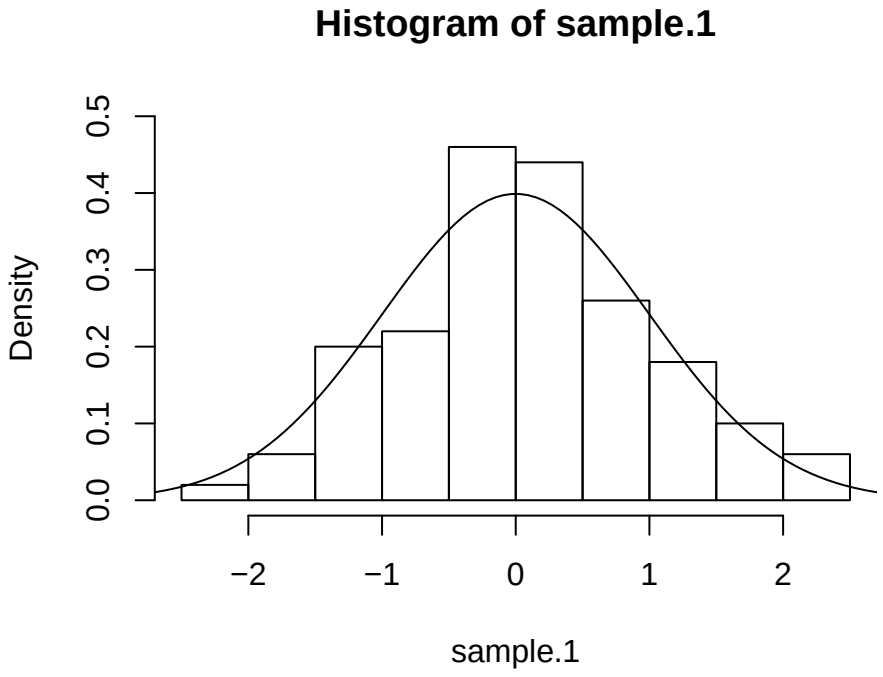
Let's see this in action with the standard normal distribution. We'll first plot the true density curve by using the **dnorm** function.

```r
x <- seq(-3, 3, length = 100)
true_density <- dnorm(x, mean = 0, sd = 1)
plot(x, true_density, ylab = "f(x)", main = "Standard Normal Density", type = "l")
```

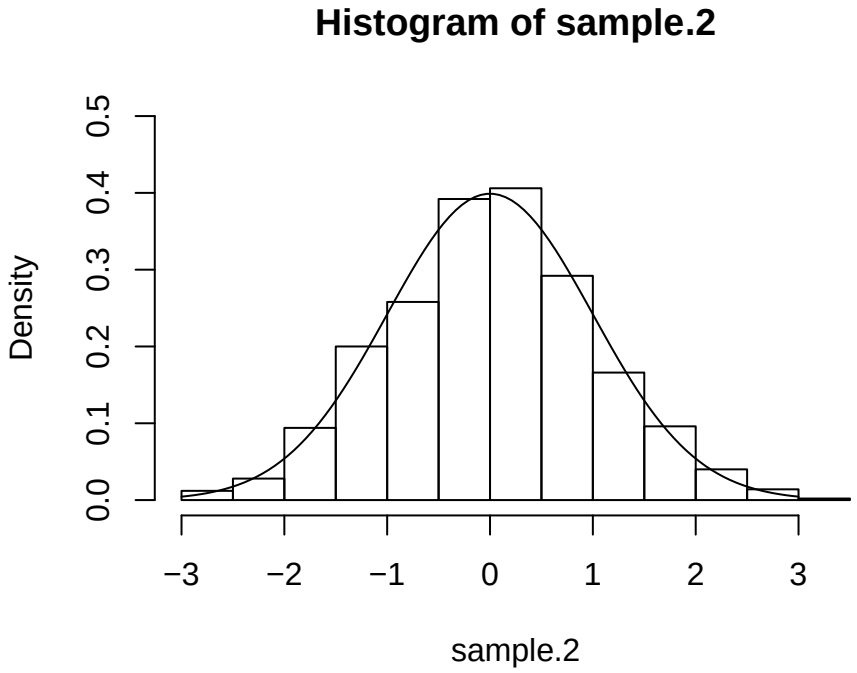



Now let's compare random samples generated by **rnorm** with varying sizes:

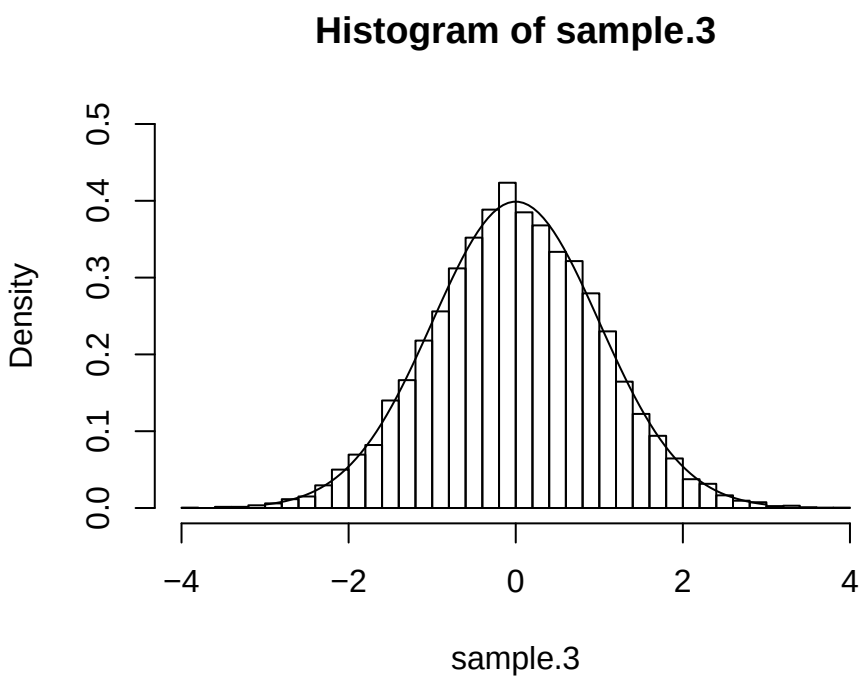
```r
set.seed(123) # IMPORTANT for reproducing the same results shown here

sample.1 <- rnorm(100, mean = 0, sd = 1) # Sample size 100
hist(sample.1, prob = TRUE, ylim = c(0, 0.5), breaks = 10)
lines(x, true_density)
```



```r
sample.2 <- rnorm(1000, mean = 0, sd = 1) # Sample size 1,000
hist(sample.2, prob = TRUE, ylim = c(0, 0.5), breaks = 20)
lines(x, true_density)
```




```r
sample.3 <- rnorm(10000, mean = 0, sd = 1) # Sample size 10,000
hist(sample.3, prob = TRUE, ylim = c(0, 0.5), breaks = 50)
lines(x, true_density)
```




Thus we see that the greater $n$ is, the more closely it approximates the density curve. As you may have realized, the output of **rnorm** is different with each run. It is best to set a seed so that your results are reproducible, which is especially important when publishing results that rely on a random number generator,

or even when debugging your code.

### Other Distributions

Note that in the normal distribution, the mean ($\mu$) and standard deviation ($\sigma$) uniquely define the distribution. However, in other distributions, other parameters must be specified which then uniquely define the distribution (such as $p$ and $n$ for the Binomial distribution). The parameters that define the distribution are the always inputs, in place of mean and sd in the examples above.

### More Information

Our goal is that this guide will allow you to better understand and use these families of functions in **R**. If you're curious about a specific function, please use the R help pages. These can easily be accessed by inserting a question mark before a function, such as:

```
?pnorm
```