# Post #2: The Famous Normal Applied in R

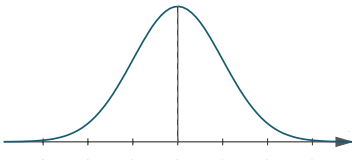
*Deborah Chang*

*12/03/2017*

```
knitr::opts_chunk$set(echo = TRUE)
```

## Introduction

The ability of R to generate data analyses by utilizing the seemingly endless possibilities of functions and packages have helped users have less tedious work. R is defined as a statistical language and hence must contain the most common tests and distributions. In elementary school, you may have learned the general summary statistics such as mean, median, and mode. Later, more complex summaries such as standard deviation, linear regression, and correlation may arise. After we transition into different distributions and probabilities. You may have seen a table of a bunch of numeric values towards the end of your statistics or math textbooks; this is the normal table. A particularly famous distribution that you may have heard of is the normal distribution, which we will cover in this post. This distribution has many applications, including its common use in course grading. Below is an example of a normal curve:



## Motivation

You may not be surprised that R has the ability to generate a normal curve. In this post, we will get an overview of the normal distribution, its purpose, and its application in R.

## Background

So what exactly is a normal distribution, and what does it do? Let's first go into the history of the normal. How was it even created? According to Wikipedia, the normal distribution can be called Gaussian or referred to as the bell curve. Below is the formula for graphing the curve in the image above:

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

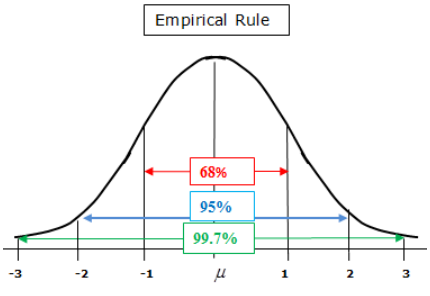
Here, $\mu$ is the mean, $\sigma$ is the standard deviation, and the square of that ($\sigma^2$) is the variance. When the mean is 0 and the variance is 1, we call the normal distribution standard. Abraham de Moivre was a statistician who grew tired of calculating probabilities using the binomial formula, as shown below:

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} a^{n-k} b^k$$

$$= a^n + \binom{n}{1} a^{n-1}b + \binom{n}{2} a^{n-2}b^2 + \binom{n}{3} a^{n-3}b^3 + \cdots + \binom{n}{n-1} ab^{n-1} + \binom{n}{n} b^n$$

$$= a^n + na^{n-1}b + \frac{n(n-1)}{2!} a^{n-2}b^2 + \frac{n(n-1)(n-2)}{3!} a^{n-3}b^3 + \cdots + nab^{n-1} + b^n$$

As you can see above, if n was really large, this formula would become really complicated. de Moivre found a quicker formula to approximate a probability when n was large, and hence the normal distribution was born. There are common probability levels noted as in the curve below, which are the 68%, 95%, and 99.7% levels.




Number of Standard Deviations Above or Below the Mean

Let's do a probability example using the normal distribution.

Suppose we have a set of data of 100 consecutive values with a mean of 50 and a standard deviation of 29.30017. What is the probability that a randomly selected value is less than 75?

```
x <- seq(0,100, by = 1)
mean(x)
```

```
## [1] 50
```

```
sd(x)
```

```
## [1] 29.30017
```

First, I would rescale the value of 1, so that I can be able to use the normal approximation table that one would find at the back of a textbook.

The formula for rescaling the value is:

```
z <- (75-50) / sd(x)
```

$$z = \frac{x - \mu}{\sigma}$$

$$\mu = \text{Mean}$$

$$\sigma = \text{Standard Deviation}$$

After rescaling our value, we get a z-score of 0.8532374, and we can use the normal approximation table to see that the probability that the value is less than 75 is 0.80323672. Note that the table can be used only for “less than” probabilities. If we want to find a “greater than” probability, we would subtract the probability of that z-score from 1 to get the inverse. In this example, the probability that the value is greater than 8 will be:

```
1-0.80323672
```

```
## [1] 0.1967633
```

# Examples

In R, we can conduct a normal approximation efficiently. The functions are:

1. pnorm(vector, mean, sd, lower.tail): This will calculate the probability that we found in our z-score table.

Arguments: vector: a vector of quantiles mean: the average sd: standard deviation lower.tail: if “TRUE (default),” the “less than” probability is calculated

Let’s use pnorm() to solve the example above.

```
pnorm(75, mean(x), sd(x))
```

```
## [1] 0.8032362
```

We do not need to rescale, as pnorm() factors in the mean and standard deviation of our data.

2. qnorm(p, mean, sd, lower.tail): This calculates the unscaled z-score from the probability, also known as the inverse value.

Arguments: p: probability value mean: the average sd: standard deviation lower.tail: if “TRUE (default),” the “less than” probability is calculated

Let’s use qnorm() to find the value that we want to calculate the probability for:

```
qnorm( 0.8032362, mean(x), sd(x))
```

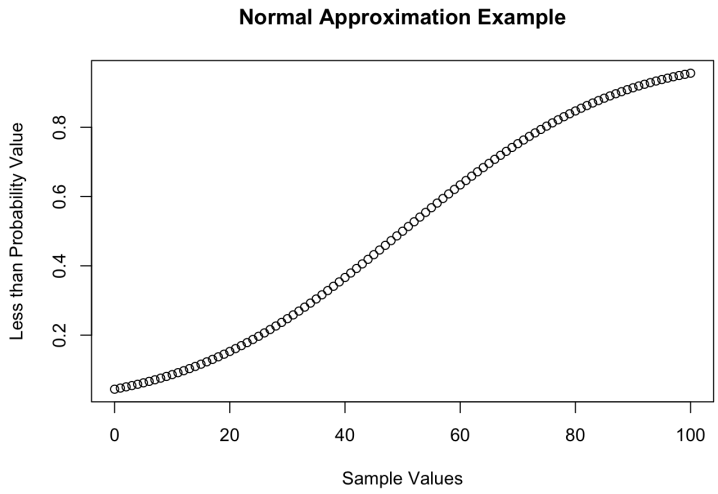
```
## [1] 75
```

3. dnorm(x, mean, sd): This function calculates the value of the normal approximation formula. Effective for graphing the normal curve.

Arguments: x: numeric value mean: the average sd: standard deviation

Let’s graph our data using the normal approximation:

```
xvar <- x
yvar <- pnorm(x, mean(x), sd(x))

plot(xvar, yvar, main = "Normal Approximation Example", xlab = "Sample Values", ylab = "Less than Probability Valu
e")
```




Let’s do another example, but this time, with a sample R dataset: PlantGrowth.

```
data("PlantGrowth")
```

For a randomly selected plant, what is the probability that its weight is over 6?

We want to calculate a “more than” probability. If we hand calculate, we will subtract the probability of our z-score. In addition, our R function pnorm() will have the mandatory argument lower.tail = FALSE.

First, let’s find the mean and standard deviation of our plant data:

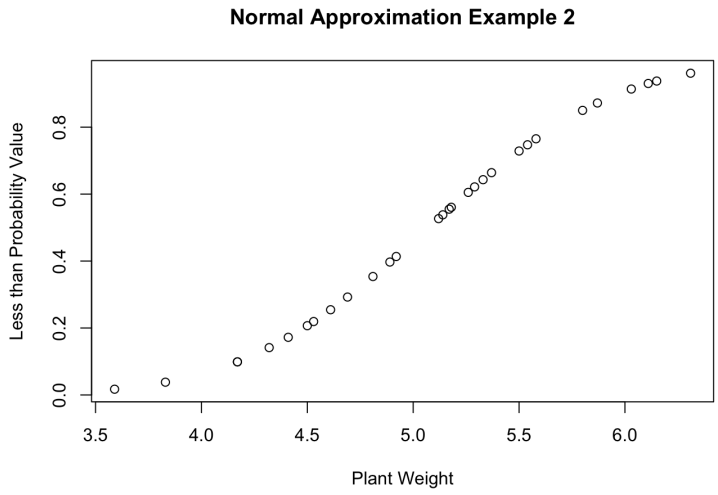
```
plantdata <- as.numeric(unlist(PlantGrowth[1]))
meanPlant <- mean(plantdata)
sdPlant <- sd(plantdata)

pnorm(6, meanPlant, sdPlant, lower.tail = FALSE)
```

```
## [1] 0.09307829
```

Hence, there is around a 9% possibility that our randomly selected plant will have a weight of over 6.

Let’s graph our plant data with the normal distribution:

```
xplant <- plantdata
yplant <- pnorm(xplant, mean(plantdata), sd(plantdata))
plot(xplant, yplant, main = "Normal Approximation Example 2", xlab = "Plant Weight", ylab = "Less than Probability Value")
```




# Discussion

The normal approximation does wonders to speeding up the process of calculating probabilities, especially in R. Our handy functions pnorm() and qnorm() can save us time in finding the appropriate z-scores as well as factoring in nonstandard data when necessary. The development of new normal distribution functions and packages could continue to beautify our understanding and appreciation for the features of R.

# Conclusion

We can continue to visualize the effects of using the normal approximation and distribution in R and compare it with other distributions. Some are more comparable when there is a limit of the number of values and the standard deviation. Hopefully the normal distribution has sparked some interest in the useRs reading this, enjoy normalizing!

# Take-Home Message

In this post, we learned the purpose of the normal distribution in relation to others in the world of statistics. We then went over the characteristics of a normal distribution and solved examples using both hand calculations and R functions. Lastly, we discussed the application of the normal distribution for new packages and functions. Note that the version of RStudio being used is 1.0.153 and that the plant growth data is already built into this platform.

# References


- http://www.mhnederlof.nl/images/normalpdf.jpg
- https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Normal.html
- https://fthmb.tqn.com/4wJnVsIDcaghwNgvSpSI_AFGaTY=/400x0/zscore-56a8fa785f9b58b7d0f6e87b.GIF
- http://www.softschools.com/math/probability_and_statistics/images/the_normal_distribution_empirical_rule_8.png
  http://tutorial.math.lamar.edu/Classes/Calcl/DerivativeProofs_files/eq0035P.gif
- https://www.google.com/search?
  tbm=isch&source=hp&biw=1306&bih=635&ei=7PkNWuy1O8iZ0wL59oeQBg&q=normal+curve&oq=normal+curve&gs_l=img.3..0l10.634.1551.0.1748.13.11.0.0.0.0.124.7
  1.1.64.img..4.9.733.0…0.-kt8mMUvvGw
- https://en.wikipedia.org/wiki/Normal_distribution
- http://onlinestatbook.com/2/normal_distribution/history_normal.html
- ://stat.ethz.ch/R-manual/R-devel/library/stats/html/Normal.html