Integration is the process of evaluating integrals. It is one of the two central ideas of calculus and is the inverse of the other central idea of calculus, differentiation. Generally, we can speak of integration in two different contexts: the indefinite integral, which is the anti-derivative of a given function; and the definite integral, which we use to calculate the area under a curve.

The fundamental concepts and theory of integration, primarily the relationship between differentiation and integration, as well as their application to the solution of applied problems, were developed in the works of P. de Fermat, I. Newton, and G. Leibniz at the end of the 17th century. Their investigations were the beginning of intensive development of mathematical analysis. The development of the theory and methods of integral calculus took place at the end of the 19th century and in the 20th century simultaneously with research into measure theory (cf. Measure), which plays an essential role in integral calculus.

By means of integral calculus, it became possible to solve by a unified method many theoretical and applied problems, both new ones which earlier had not been amenable to solution, and old ones that had previously required special artificial techniques. The basic notions of integral calculus are two closely related notions of the integral, namely the indefinite and the definite integral.

Integration is used in a multitude of fields that you wouldn’t ordinarily think would make use of its concepts. Among them are physics, engineering, economics, statistics, and medicine. Integration is also used in such disparate areas as space travel, as well as determining how medications interact with the body, and even how to build safer structures. You’ll understand why Integration is useful in so many areas if you know a bit about its history as well as what it is designed to do and measure.

If you want to understand what’s going on under the hood in your machine learning work as a data scientist, you’ll need to have a solid grasp of the fundamentals of integration. While you cannot do a lot of data science with just integration, the topic is essential for more advanced topics in data science such as machine learning, algorithms, and advanced statistics.

Let’s get started using Integration with R:

**Integration function in R**

Let us start by making an integration function in R:

F = antiD(a \* x^2 ~ x, a = 1)

F

*function (x, a = 1, C = 0)*

*a \* 1/(3) \* x^3 + C*

Let us try to evaluate the integration at the endpoints of the interval of integration and subtract the two result:

F(x = 3) - F(x = 1) # Should be 9 - 1/3

*[1] 8.667*

**Symbolically integrals in R**

It is also possible to use antiD() for performing integrals symbolically:

antiD(1/(a \* x + b) ~ x)

*function (x, a, b, C = 0)*

*1 \* 1/(a) \* log(((a \* x + b))) + C*

When antiD() cannot find a symbolic form, the anti-derivative will be based on a process of numerical integration:

P = antiD(exp(x^2) ~ x)

P

*function (x, C = 0)*

*{*

*numerical.integration(.newf, .wrt, as.list(match.call())[-1],*

*formals(), from, ciName = intC, .tol)*

*}*

**Mathematical points of integrals in R**

The numerical integration method doesn’t discover the small range over which the integrand is non-zero. For instance, the following integral should be 1:

F(Inf) - F(-Inf)

*[1] 0*

Even integrals out to innity can often be handled with great precision. Here, for instance, is a calculation of the mean of a normal distribution via integration from (-inf) to (inf):

F = antiD(x \* dnorm(x, mean = 3, sd = 2) ~ x, lower.bound = -Inf)

F(x = Inf)

*[1] 3*

**Integrals of exponential distributions in R**

Here are some examples using exponential distributions:

F = antiD(x \* dexp(x, rate = rate) ~ x)

F(x = Inf, rate = 10) - F(x = 0, rate = 10)

*[1] 0.1*

F(x = Inf, rate = 100) - F(x = 0, rate = 100)

*[1] 0.01*

**Integrals of integrals in R**

It’s also possible to take the integral of a function that is itself an integral:

one = makeFun(1 ~ x + y)

by.x = antiD(one(x = x, y = y) ~ x)

by.xy = antiD(by.x(x = sqrt(1 - y^2), y = y) ~ y)

by.xy(y = 1) - by.xy(y = -1)

*[1] 1.571*

**Plotting Integration function in R**

Let is try plotting a integration function in R:

# intergrand f

f <- function(r,x) x\*exp(-r) # order of arguments reversed

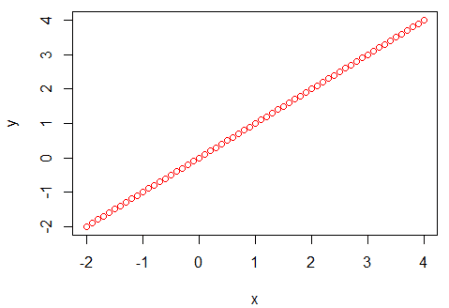
# integral

h <- function(x) integrate(f, lower=0, upper=Inf, x=x)$value

g <- Vectorize(h)

x <- seq(-2,4,.1)

plot(x,g(x), xlim=c(-2,4), xlab="x", ylab="y", col="red")

The above coding gives us the following plot:  
[](https://i2.wp.com/datascienceplus.com/wp-content/uploads/2019/07/IG1.png?ssl=1)