Differentiation and Integration in R

Dr. Noah Mutai

1 Introdutcion to Calculus

Calculus is a branch of mathematics that involves the study of rates of change.

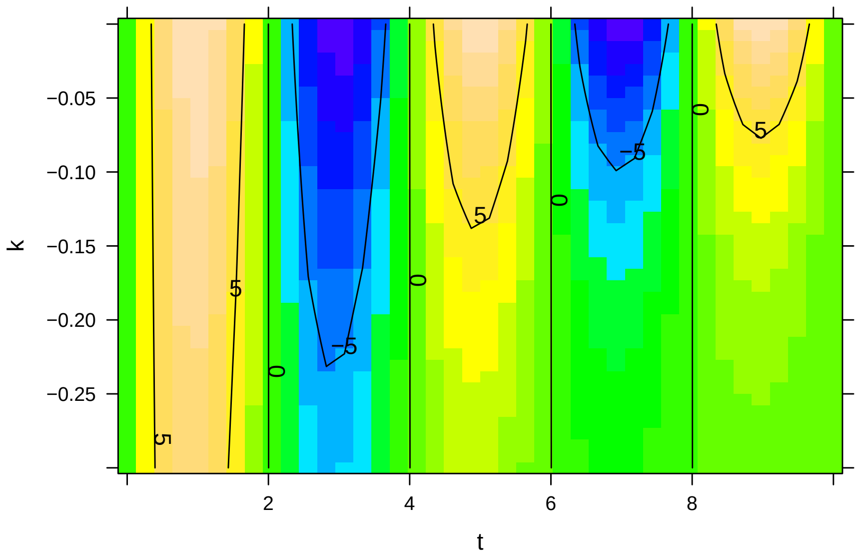
1.1 Functions in R

Create simple mathematical functions with makeFun()

```
f <- makeFun(m * x + b ~ x, m = 3.5, b = 10)
f
# when x = 2
f(x = 2)

g <- makeFun(A * x * cos(pi * x * y) ~ x + y, A = 3)
g

function(x, y, A = 3) {
  A * x * cos(pi * x * y)
}

# when x = 1 and y = 2
g(x = 1, y = 2)
```

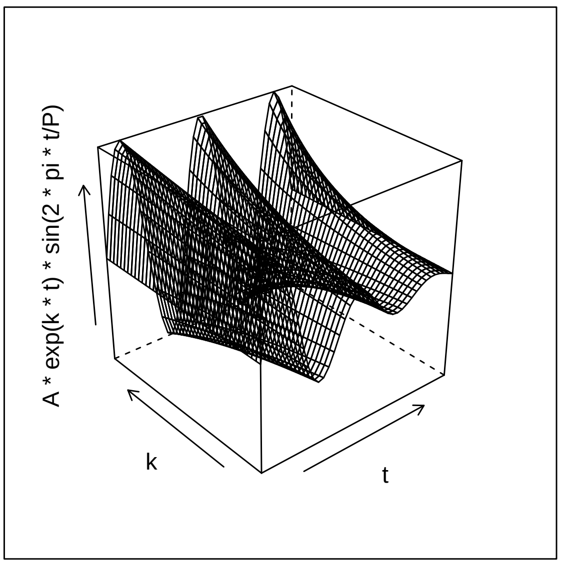
Let us try plotting the calculus function

```
plotFun(A * exp(k * t) * sin(2 * pi * t / P) ~ t + k, t.lim = range(0, 10),
        k.lim = range(-0.3, 0), A = 10, P = 4)
```



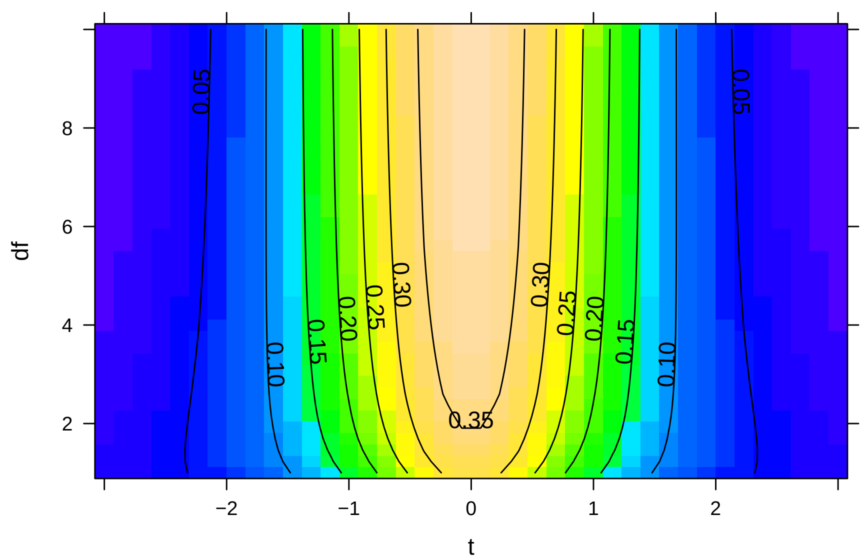
Let us try do do another function in R:

```
plotFun(A * exp(k * t) * sin(2 * pi * t / P) ~ t + k, t.lim = range(0, 10),
        k.lim = range(-0.3, 0), A = 10, P = 4, surface = TRUE)
```



Let us try do do a third function in R:

```
plotFun(dt(t, df) ~ t + df, t.lim = range(-3, 3), df.lim = range(1, 10))
```



1.2 Derivative Functions in R

As with all computations, the operator for taking derivatives, D() takes inputs and produces an output.

Input: an expression using the ~ notation. Examples: x^2~x or sin(x^2)~x or y*cos(x)~y

On the left of the ~ is a mathematical expression, written in correct R notation, that will evaluate to a number when numerical values are available for all of the quantities referenced. On the right of the ~ is the variable with respect to which the derivative is to be taken. By no means need this be called x or y; any valid variable name is allowed.

The output produced by D() is a function. The function will list as arguments all of the variables contained in the input expression. You can then evaluate the output function for particular numerical values of the arguments in order to find the value of the derivative function.

Now let us try to do the derivative of a function in r:

```
D(sin(x) ~ x)
function(x) {
```

```
  cos(x)
}
```

Let us try to do a second derivative in R:

```
D(A * x^2 * sin(y) ~ x)
function(x, A, y) {
  A * (2 * x) * sin(y)
}
```

Let us try to do a third derivative in R:

```
D(A * x^2 * sin(y) ~ x + y)
function(x, y, A) {
  A * (2 * x) * cos(y)
}
```

Solving in R The findZeros() function will locate zeros of a function in a flexible way that's easy to use. The syntax is very similar to that of plotFun() and D().

Let us try solving in R:

```
findZeros(sin(t) ~ t, t.lim = range(-5, 1))
```

Let us Find the nearest several zeros to a point:

```
findZeros(sin(t) ~ t, nearest = 5, near = 10)
```

Let us Specify a range via a center and width and solve this:

```
findZeros(sin(t) ~ t, near = 0, within = 8)
```

Let us solve a solution to 4 sin(3x) = 2:

```
solve(4 * sin(3 * x) == 2 ~ x, near = 0, within = 1)
```

Let us findZeros() for nonlinear functions:

```
findZeros(x * y^2 - 8 ~ x & y, sin(x * y) - 0.5 ~ x & y)
```

Let us try findZeros() for another nonlinear function:

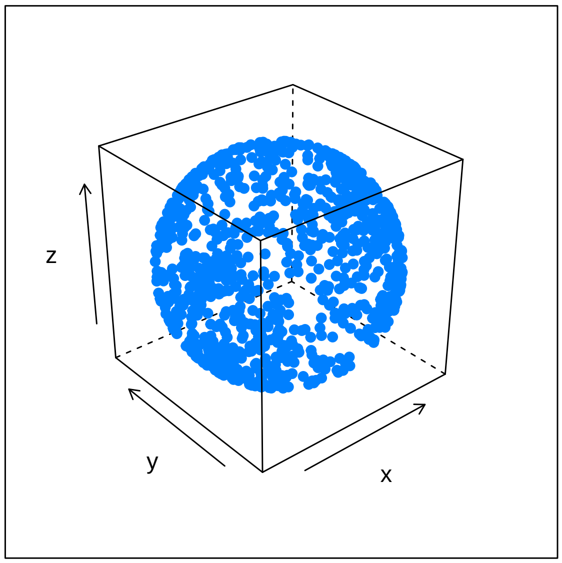
```
findZeros(x * y^2 - 8 ~ x & y, sin(x * y) - 0.5 ~ x & y,
  near = c(x = 20, y = 0),
  within = c(x = 5, y = 1)
)
```

Let us solve the zeros of the function x^2 + y^2 + z^2 -10 in the x, y, z space:

```
findZeros(x^2 + y^2 + z^2 - 10 ~ x & y & z, near = 0, within = 4)
```

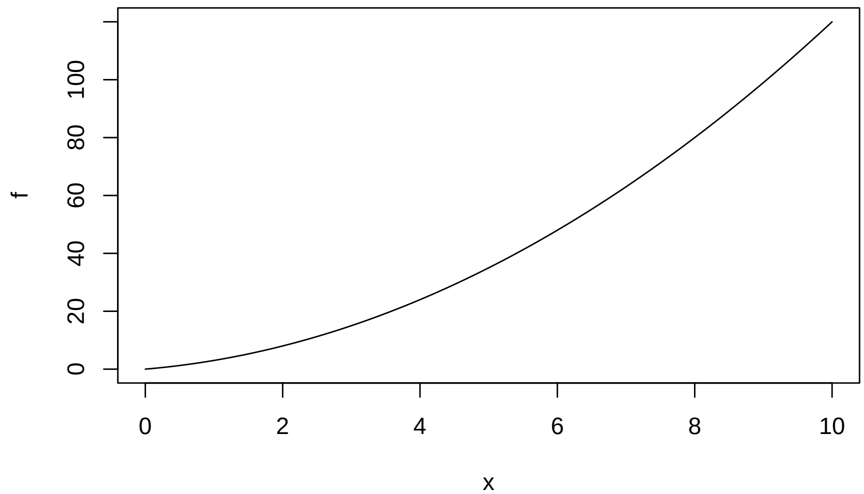
If we want to to show the surface nicely, we need lots of points. Let us try 1000 points that as a rule of thumb are requested:

```
s1 <- findZeros(x^2 + y^2 + z^2 - 10 ~ x & y & z, near = 0, within = 10,
                nearest = 1000)
cloud(z ~ x + y, data = s1, pch = 19)
```



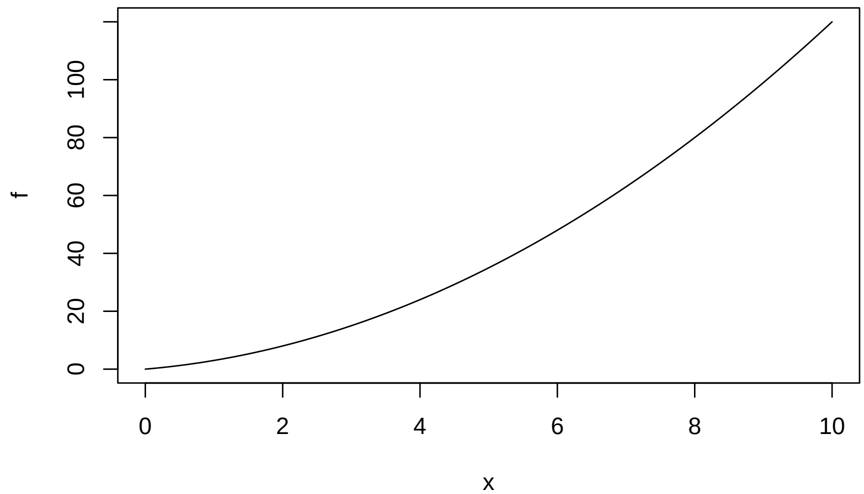
Plotting Derivative Functions in R Let us now try plotting some functions before taking the derivative of the function.

```
D <- function(f, delta = .000001) {
  function(x) {
    (f(x + delta) - f(x - delta)) / (2 * delta)
  }
}
f <- function(x) {
  x^2 + 2 * x
}
plot(f, 0, 10)
```

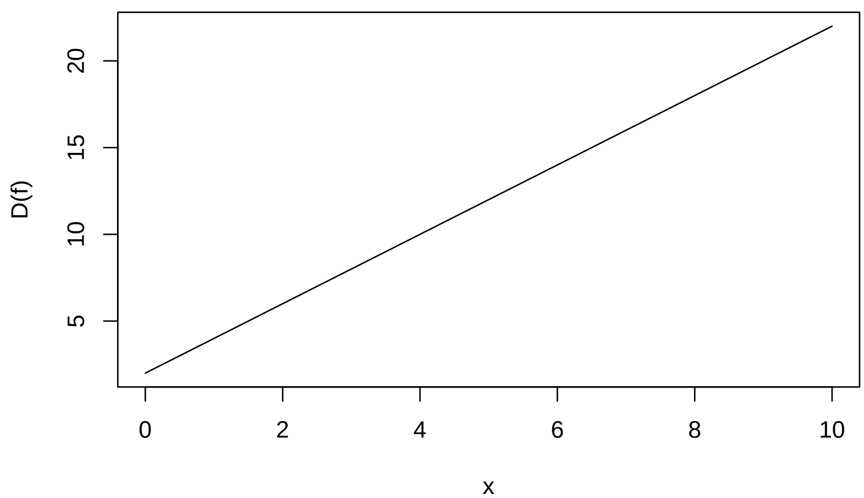


Now let us try plotting a function and the derivative of the function:

```
f <- function(x) {
  x^2 + 2 * x
}
plot(f, 0, 10)
```

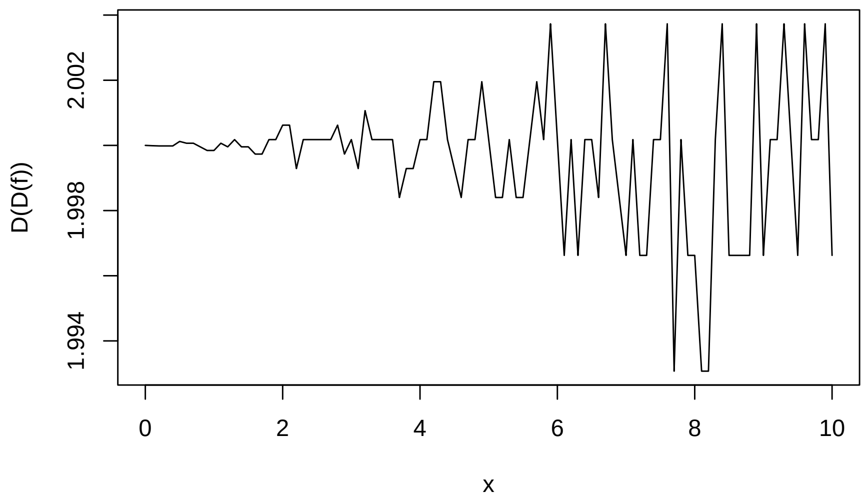



```
plot(D(f), 0, 10)
```



Now let us try plotting the derivative of the derivative function:

```
plot(D(D(f)), 0, 10)
```



2 Integration

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.

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
```

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
  )
}
```

Integrals of exponential distributions in R

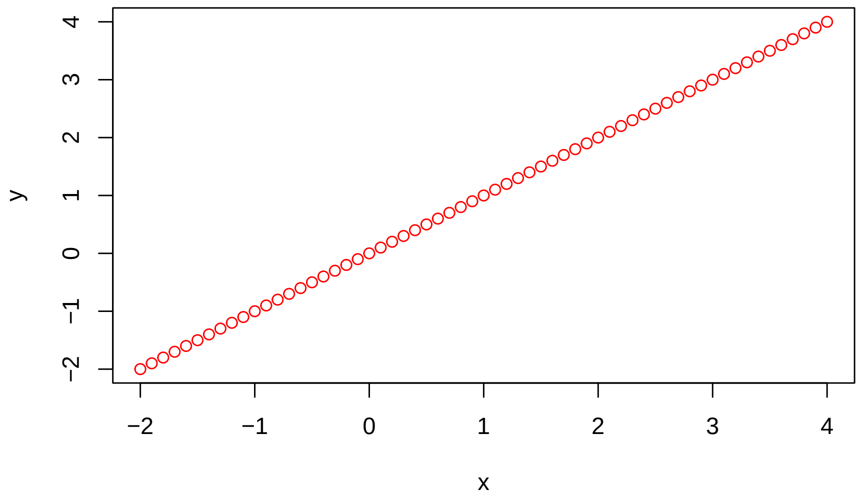
```
F <- antiD(x * dexp(x, rate = rate) ~ x)
F(x = Inf, rate = 10) - F(x = 0, rate = 10)
F(x = Inf, rate = 100) - F(x = 0, rate = 100)
```

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)
```

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")
```



3 Exercises

(1) Using D(), find the derivative of 3 * x ^ 2 - 2*x + 4 ~ x. What is the value of the derivative at x=0 ? {-6,-4,-3,-2,0,2,3,4,6}

(2) Using D(), find the derivative of 5 * exp(0.2 * x) ~ x.

(a) What is the value of the derivative at x=0 ? {-5,-2,-1,0,1,2,5}.

(b) Plot out both the original exponential expression and its derivative. How are they related to each other?