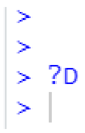
輔仁大學管理學院微積分主題式教材
－程式介紹

1、 **微分函數 D(expr, name)**

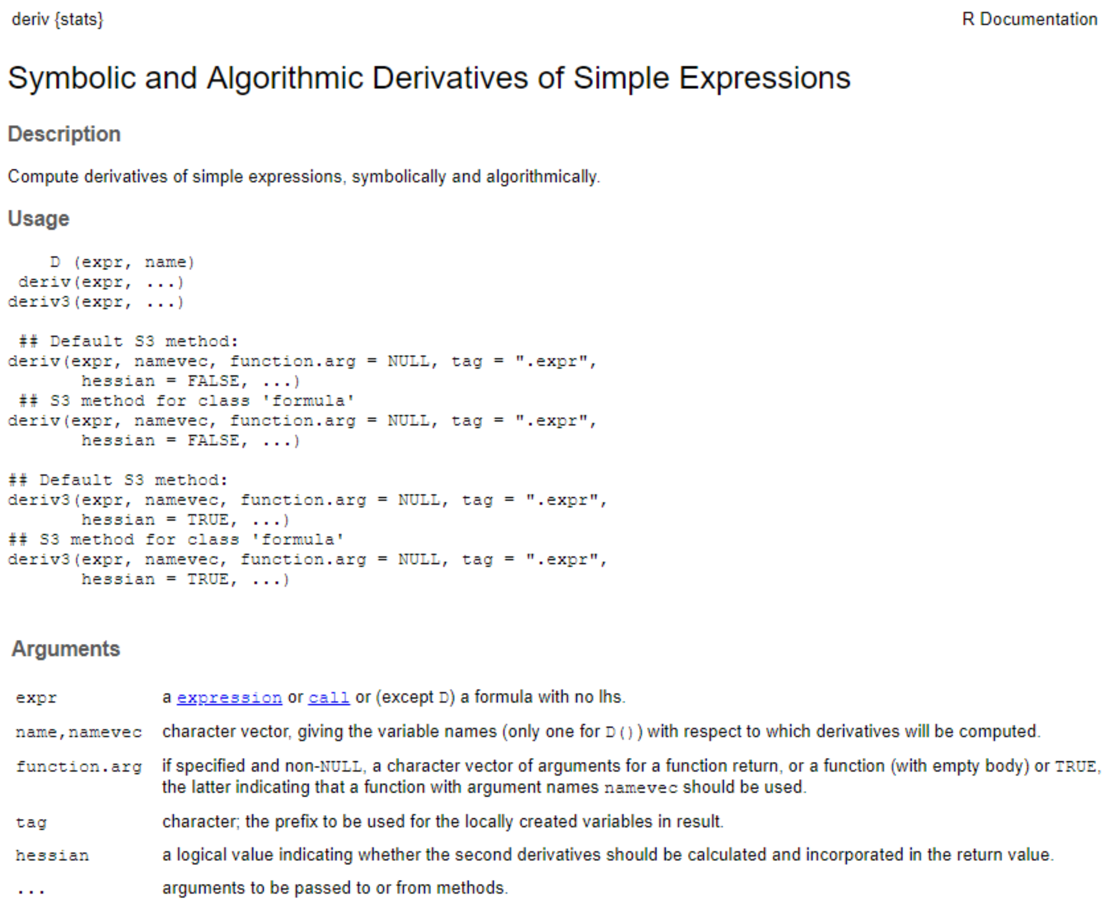
程式部分主要分為微分函數以及繪圖函數兩部分進行解說，首先要介紹的是微分函數 D()，而 Rstudio 提供函數查詢的功能，只要在函數前方加上問號，Rstudio 就會在 help 的部分顯示相關的文件 ：

(1)查詢指令



```
>
>
> ?D
>
```

(2)相關解說文件



deriv {stats}　　　　R Documentation

Symbolic and Algorithmic Derivatives of Simple Expressions

Description

Compute derivatives of simple expressions, symbolically and algorithmically.

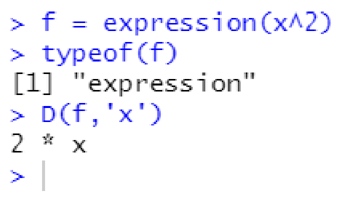
Usage

```
    D (expr, name)
 deriv(expr, ...)
deriv3(expr, ...)

 ## Default S3 method:
deriv(expr, namevec, function.arg = NULL, tag = ".expr",
       hessian = FALSE, ...)
 ## S3 method for class 'formula'
deriv(expr, namevec, function.arg = NULL, tag = ".expr",
       hessian = FALSE, ...)

## Default S3 method:
deriv3(expr, namevec, function.arg = NULL, tag = ".expr",
       hessian = TRUE, ...)
## S3 method for class 'formula'
deriv3(expr, namevec, function.arg = NULL, tag = ".expr",
       hessian = TRUE, ...)
```

Arguments

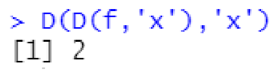
`expr`	a expression or call or (except `D`) a formula with no lhs.
`name,namevec`	character vector, giving the variable names (only one for `D()`) with respect to which derivatives will be computed.
`function.arg`	if specified and non-`NULL`, a character vector of arguments for a function return, or a function (with empty body) or `TRUE`, the latter indicating that a function with argument names `namevec` should be used.
`tag`	character; the prefix to be used for the locally created variables in result.
`hessian`	a logical value indicating whether the second derivatives should be calculated and incorporated in the return value.
`...`	arguments to be passed to or from methods.

D(expr, name)主要分為函式(expr)，以及變數(name)兩部分，函式部分要先用 expression 來將數學式轉換成 expression 的型態，再將轉換後的 expression 放入 D()，並設定要進行微分的變數，即可完成一階微分：



```
> f = expression(x^2)
> typeof(f)
[1] "expression"
> D(f,'x')
2 * x
> 
```

若需進行二次微分，只需再加上一層 D()函數即可，以此類推：



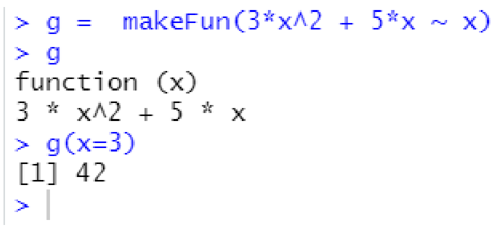
```
> D(D(f,'x'),'x')
[1] 2
```

2、 繪圖函數

而繪圖函數則是採用 mosaic 套件中的 makeFun 函數以及 plotFun 函數，makeFun 函數主要用途為建立方程式，例如：makeFun(3*x^2 + 5*x ~ x)，即為以下方程式:

$$f(x) = 3x^2 + 5x$$

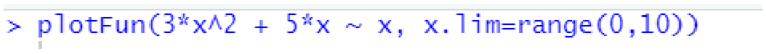
並可利用變數存取此方程式，以便用來給定 x 的數值:



```
> g =  makeFun(3*x^2 + 5*x ~ x)
> g
function (x)
3 * x^2 + 5 * x
> g(x=3)
[1] 42
> 
```

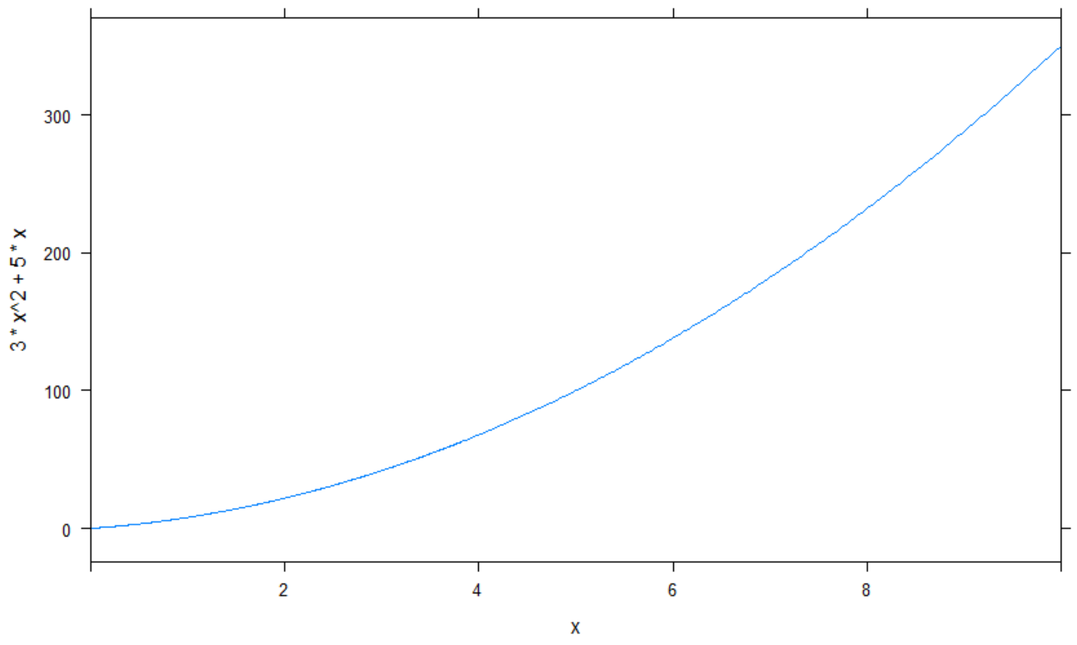
此例即為將方程式儲存成變數 g，並且將變數 g 帶入 x = 3，經由方程式的計算後結果為 42。

plotFun 函數則是用來繪製方程式的圖形，例如：



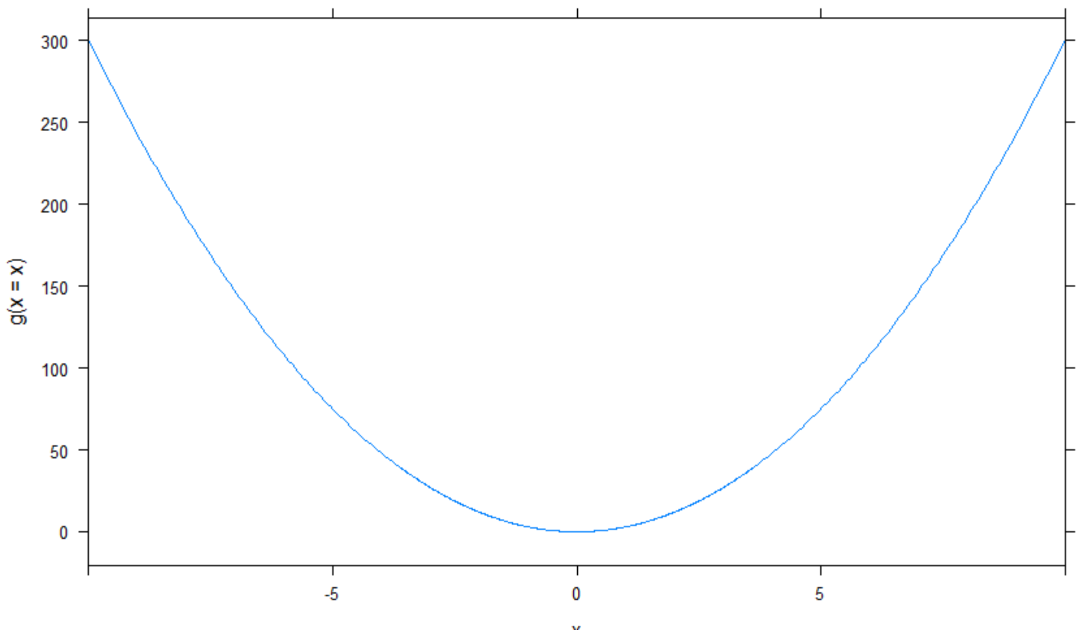
```
> plotFun(3*x^2 + 5*x ~ x, x.lim=range(0,10))
```

可繪製出上述方程式$f(x) = 3x^2 + 5x$帶入 x 範圍 0~10 之圖形 ：



也可帶入經由 makeFun 所產生的方程式 ：

```
> g =  makeFun(3*x^2 ~ x)
> plotFun(g(x=x) ~ x, x.lim=range(-10,10))
```


300
250
200
150
100
50
0
g(x = x)
-5
0
5
x