Introductiry Programming in R

By Asef Nazari

asef.nazari@monash.edu

Faculty of IT

Monash university

5. Functions

5.1 User defined Functions

It is easy to define a functon. Here is a simple example.

```
func_name <- function (argument) {
   statement
}
```

In [21]:

```
# your function needs a name
myfunc1 <- function(n){
    n*n
} # the function will return the last value
```

In [22]:

```
# to call a function
myfunc1(5)
```

25

In [23]:

```
t <- 1:5
myfunc1(t)
```

1 4 9 16 25

In [24]:

```
x <- 10
y <- myfunc1(x)
y
```

100

In [25]:

```
# the variable y has a default value. if you don't mention it, it will be 2.
f2 <- function(x,y=2){
    x+y
}
```

In [26]:

```
f2(5,5)
```

10

In [27]:

```
f2(5)
```

7

In [28]:

```
fun3 <- function(x){
    str(x)
}
```

In [29]:

```
fun3(f2)
```

```
function (x, y = 2)
 - attr(*, "srcref")=Class 'srcref'  atomic [1:8] 2 7 4 1 7 1 2 4
  .. ..- attr(*, "srcfile")=Classes 'srcfilecopy', 'srcfile' <environm
ent: 0x1a71ff8>
```

In [30]:

```
col.mean <- function(y, removeNA=TRUE){
    nc <- ncol(y)
    #print(nc)
    means <- numeric(nc) # a vector of size nc containing 0
    for(i in 1:nc){
        means[i] <- mean(y[,i], na.rm=removeNA) # remember maen() is sensitive for
    }
    means
}
#very last expresion is the return value
```

In [31]:

```
col.mean(airquality)
```

```
    42.1293103448276   185.931506849315   9.95751633986928   77.8823529411765
    6.99346405228758   15.8039215686275
```

In [32]:

```
dim(airquality)
```

```
    153   6
```

In [33]:

```
make.power <- function(n){ #a function returns a function
    pow <- function(x){
        x^n
    }
    pow
}
```

In [34]:

```
cube <- make.power(3)
```

In [35]:

```
cube(2)
```

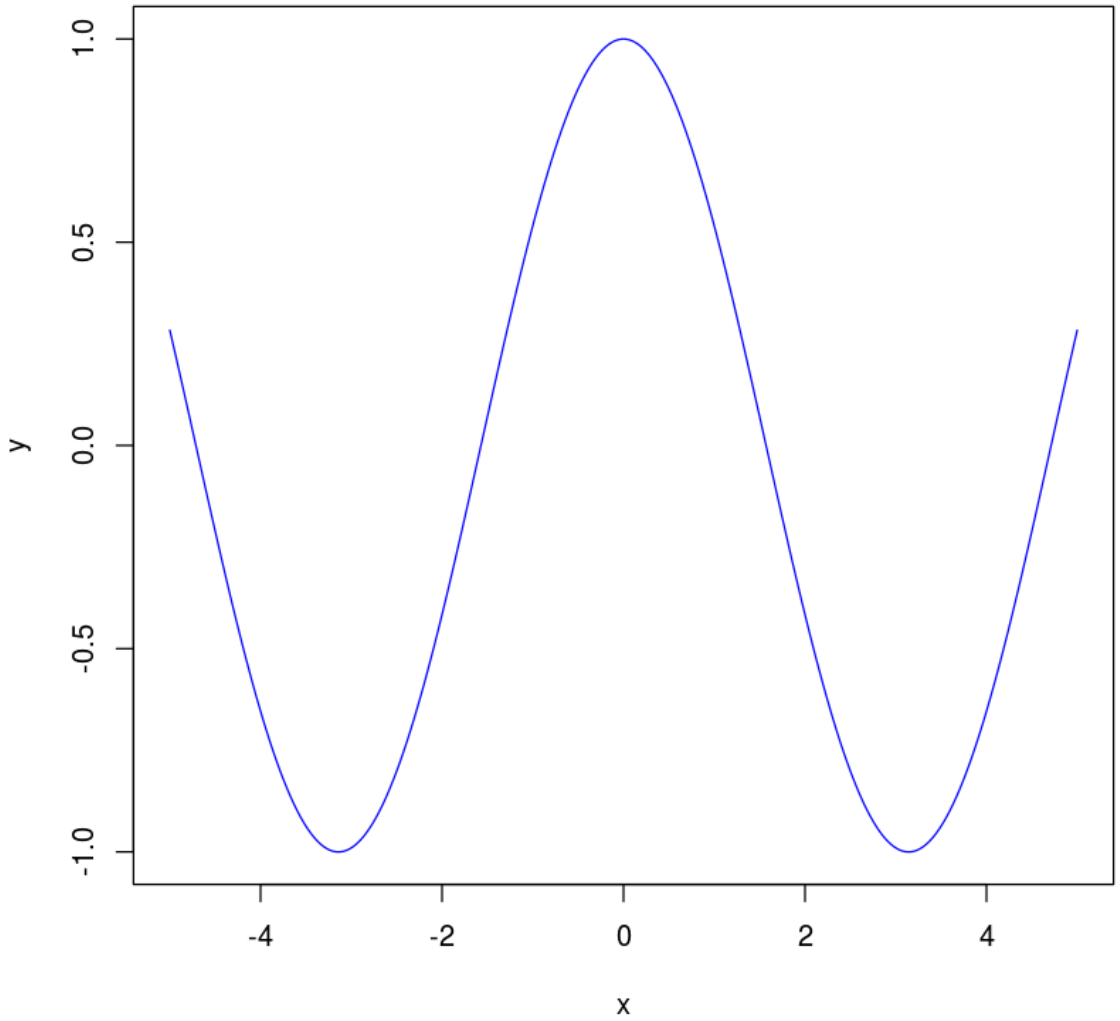
8

In [36]:

```
drawFun <- function(f){
    x <- seq(-5, 5, len=1000)
    y <- sapply(x, f)
    plot(x, y, type="l", col="blue")
}
```

In [37]:

```
drawFun(cos)
```



R Built-in Functions

To use R's built-in functions we need to follow their arguments. A function takes arguments as input and returns an object as output.

In [38]:

```
x <- 1:10
sum(x)
length(x)
median(x)
```

55

10

5.5

In [39]:

```
? seq
```

In [40]:

```
#Type the name of the function without any parentheses or arguments
seq
#if you see UseMethod, there are multiple methods (functions)
#associated with the seq function
### somefunctions might be hidden!
```

```
function (...)
UseMethod("seq")
```

In [41]:

```
methods(seq)
```

```
[1] seq.Date    seq.default seq.POSIXt
see '?methods' for accessing help and source code
```

In [42]:

```
#seq.Date
```

In [43]:

```
seq()
```

```
1
```

In [44]:

```
args(seq)
```

```
function (...)
NULL
```

In [45]:

```
args(round)
```

```
function (x, digits = 0)
NULL
```