Vectors

Types of Vectors

In this chapter, I would learn about the type of vectors. There are basically 4 types of vectors. That is logical (TRUE- FALSE), Integer (non decimal values, NaN and Inf, -Inf), Double (decimal values) and Character.

Vector that is the same type of elements is Atomic Vector. Otherwise it is a list.

```
lgl_var <- c(TRUE, FALSE)
int_var <- c(1L, 6L, 10L)
dbl_var <- c(1, 2.5, 4.5)
chr_var <- c("these are", "some strings")
```

To get the type of the vector, we would use the function typeof()

```
typeof(lgl_var)
```

```
## [1] "logical"
```

```
typeof(int_var)
```

```
## [1] "integer"
```

```
typeof(dbl_var)
```

```
## [1] "double"
```

```
typeof(chr_var)
```

```
## [1] "character"
```

Missing values

Calculation with NA are mostly NA. Excepting some cases

```
NA + 10
```

```
## [1] NA
```

```
NA *10
```

```
## [1] NA
```

```
NA < 10
```

```
## [1] NA
```

```
!NA
```

```
## [1] NA
```

NA is something between TRUE and FALSE. Let's see it. When doing with TRUE and FALSE, | returns in TRUE & returns in FALSE. NA is in between.

```
NA| TRUE
```

```
## [1] TRUE
```

```
NA & TRUE
```

```
## [1] NA
```

```r
NA | FALSE
```

```
## [1] NA
```

```r
NA & FALSE
```

```
## [1] FALSE
```

```r
NA ^ 0
```

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

Checking if NA equal NA or other numbers. Returns all in NA.

```r
x <- c(NA, 1, NA, 2)
x == NA
```

```
## [1] NA NA NA NA
```

Testing and Coercion

We can use the function is.*() to test the type of a vector. But it applies to is.logical(), is.integer(), is.double() and is.character()

```r
is.logical(lgl_var)
```

```
## [1] TRUE
```

```r
is.integer(lgl_var)
```

```
## [1] FALSE
```

```r
is.double(dbl_var)
```

```
## [1] TRUE
```

```r
is.character(chr_var)
```

```
## [1] TRUE
```

Check carefully if you use these tests below by reading their documentation:

```r
is.vector(lgl_var)
```

```
## [1] TRUE
```

```r
is.numeric(chr_var)
```

```
## [1] FALSE
```

```r
is.atomic(dbl_var)
```

```
## [1] TRUE
```

When combining different type of vector into one vector, the type of the vector will be corerced into this order: character > double > integer > logical.

```r
typeof(c(lgl_var, dbl_var))
```

```
## [1] "double"
```

```r
typeof(c(lgl_var, chr_var))
```

```
## [1] "character"
```

```
typeof(c(chr_var, int_var))
```

```
## [1] "character"
```

Coercing into integer will make the element that is character becomes NA

```
mixed_var <- c(chr_var, int_var)
typeof(mixed_var)
```

```
## [1] "character"
```

```
as.integer(mixed_var)
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA NA  1  6 10
```

Attributes

We can set name pair attributes for any variable. Attribute can be strings or numeric.

```
a <- 1:3
attr(a, "x") <- "abcdef"
attr(a, "x")
```

```
## [1] "abcdef"
```

```
attr(a, "y") <- 4:6
str(attributes(a))
```

```
## List of 2
##  $ x: chr "abcdef"
##  $ y: int [1:3] 4 5 6
```

```
# Or equivalently:
a <- structure(1:3, x = "abcdef", y = 4:6)
str(attributes(a))
```

```
## List of 2
##  $ x: chr "abcdef"
##  $ y: int [1:3] 4 5 6
```

There are 2 attributes that is generally preserved. It is names (character vector giving each elements a name) and dim (short for dimension, used to turn vectors into matrices or arrays).

Names

We have a few way to creates name

```
# Create name at the beginning
x <- c("a" = 1, "b" = 2, "c" = 3)
x
```

```
## a b c
## 1 2 3
```

```
# Create a names vector
y = 1:3
names(y) = c("a", "b", "c")
y
```

```
## a b c
## 1 2 3
```

```
# Inline with setNames()
z <- setNames(1:3, c("a", "b", "c"))
z
```

```
## a b c
## 1 2 3
```

Dimensions

```
# setting matrix
x <- matrix(1:6, nrow = 2, ncol = 3)
x
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

```
y <- array(1:12, c(2, 3, 2))
y
```

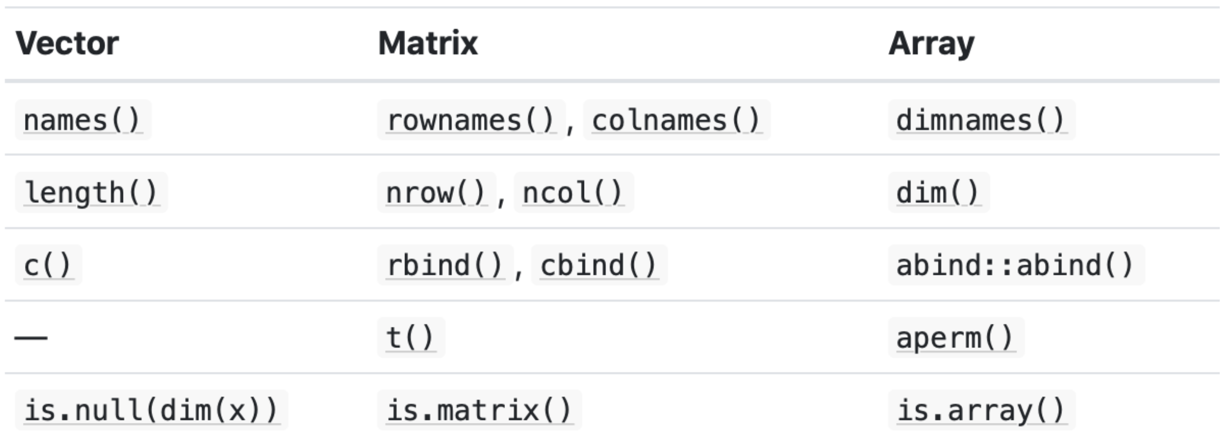
```
## , , 1
##
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
##
## , , 2
##
##      [,1] [,2] [,3]
## [1,]    7    9   11
## [2,]    8   10   12
```

We can also modify the dim to set the object

```
z <- 1:6
dim(z) <- c(2, 3)
z
```

```
##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

Vector, Matrix and Array have the functions as below.



Vector	Matrix	Array
`names()`	`rownames()`, `colnames()`	`dimnames()`
`length()`	`nrow()`, `ncol()`	`dim()`
`c()`	`rbind()`, `cbind()`	`abind::abind()`
—	`t()`	`aperm()`
`is.null(dim(x))`	`is.matrix()`	`is.array()`

Figure 1: Formula