Data Types

Jayant

2/21/2020

Atomic classes in numeric, logical, character, integer, and complex vectors. # Vectors

```
x <- c(1.5,3.2)             #Numeric
x
```

```
## [1] 1.5 3.2
```

```
x <- c("HA","HA")           #Character
x
```

```
## [1] "HA" "HA"
```

```
x <- 1:4                    #Integer
x
```

```
## [1] 1 2 3 4
```

```
x <- 1:4                    #Logical
x
```

```
## [1] 1 2 3 4
```

```
x <- c(1+0.5i,6-0.3i)       #Complex
x
```

```
## [1] 1+0.5i 6-0.3i
```

```
x <- vector()
x
```

```
## logical(0)
```

```
x <- vector("numeric",10)  #Numeric vector of length 10
x
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0
```

If a vector has different classes it will take the least level one as its own class . levels are as follows:- **Character<Numeric<Logic**

Coercion

explicitly coerce objects from one class to another using functions that usually start with the word "as."

```
x <- 0:6
x
```

```
## [1] 0 1 2 3 4 5 6
```

```
class(x)
```

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

```
as.numeric(x)      # Converts to Numeric
```

```
## [1] 0 1 2 3 4 5 6
```

```
as.character(x)    # Converts to Character
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

```
as.logical(x)      # Converts to Logic
```

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

```
as.complex(x)      # Converts to Complex
```

```
## [1] 0+0i 1+0i 2+0i 3+0i 4+0i 5+0i 6+0i
```

```
x <- c("a","b","c")
as.numeric(x)
```

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

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

Lists

They are an exception of vectors.

```
x <- list(1,"a",TRUE,1+0.9i)
x
```

```
## [[1]]
## [1] 1
##
## [[2]]
## [1] "a"
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] 1+0.9i
```

Matrix

```
m <- matrix(nrow = 2,ncol = 2)
m
```

```
##      [,1] [,2]
## [1,]   NA   NA
## [2,]   NA   NA
```

```
m <- matrix(1:4,nrow = 2,ncol = 2)
m
```

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

```
dim(m)
```

```
## [1] 2 2
```

```
attributes(m)
```

```
## $dim
## [1] 2 2
```

```
m <- 1:6
m
```

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

```
dim(m) <- c(2,3)
m
```

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

```
x <- 0:2
x
```

```
## [1] 0 1 2
```

```
y <- 10:12
y
```

```
## [1] 10 11 12
```

```r
rbind(x,y)
```

```
##   [,1] [,2] [,3]
## x    0    1    2
## y   10   11   12
```

```r
cbind(x,y)
```

```
##      x  y
## [1,] 0 10
## [2,] 1 11
## [3,] 2 12
```

Values are entered column wise when we convert array to matrix

Factors

Used to represent categorical data by using descreptive words instead of using integers

```r
x <- factor(c("y","y","n","y"))
x
```

```
## [1] y y n y
## Levels: n y
```

```r
unclass(x)
```

```
## [1] 2 2 1 2
## attr(,"levels")
## [1] "n" "y"
```

```r
x <- factor(c("y","y","n","y"),levels = c("y","n"))
x
```

```
## [1] y y n y
## Levels: y n
```

Naming Rows and Columns (Data Frames)

```r
x <- data.frame(foo=1:4,bar = c("h","e","l","l"))
x
```

```
##   foo bar
## 1   1   h
## 2   2   e
## 3   3   l
## 4   4   l
```

```r
x <- 1:3
x
```

```
## [1] 1 2 3
```

```r
names(x) <- c("A","B","C")
x
```

```
## A B C
## 1 2 3
```

```r
names(x)
```

```
## [1] "A" "B" "C"
```

```r
x <- list(A = 1:3, B = 4:6, C = 7:9)
x
```

```
## $A
## [1] 1 2 3
##
## $B
## [1] 4 5 6
##
## $C
## [1] 7 8 9
```

```r
m <- matrix(1:4,nrow = 2,ncol = 2)
dimnames(m) <- list(c("A","B"),c("C","D"))
m
```

```
##   C D
## A 1 3
## B 2 4
```

Summary

So far, we've talked about the atomic classes in numeric, logical, character, integer, and complex vectors. We talked about how vectors can only have elements of the same class and the main exception to that is lists which can have elements of different classes. There are factors which are used for, for coding categorical data, with ordered and unordered data. There are missing values that are represented by NAs, and NANs. Data frames are used to store tabular data or each COM can be of a different class. And finally, all our R objects can have names which mean usul, which can be useful for creating self-describing data.