Advanced R: Data Structures

dbl\_var <- c(1, 2.5, 4.5)  
int\_var <- c(1L, 6L, 10L)  
log\_var <- c(TRUE, FALSE, T, F)  
chr\_var <- c('these are', 'some strings')  
chr\_var

## [1] "these are" "some strings"

c(1, c(2, c(3,4)))

## [1] 1 2 3 4

#same as  
c(1,2,3,4)

## [1] 1 2 3 4

int\_var <- c(1L, 6L, 10L)  
typeof(int\_var)

## [1] "integer"

is.integer(int\_var)

## [1] TRUE

is.atomic(int\_var)

## [1] TRUE

dbl\_var <- c(1, 2.5, 4.5)  
typeof(dbl\_var)

## [1] "double"

is.double(dbl\_var)

## [1] TRUE

is.atomic(dbl\_var)

## [1] TRUE

is.factor(dbl\_var)

## [1] FALSE

is.numeric(int\_var)

## [1] TRUE

is.numeric(dbl\_var)

## [1] TRUE

#coercion  
str(c('a', 1))

## chr [1:2] "a" "1"

x <- c(FALSE, FALSE, TRUE)  
as.numeric(x)

## [1] 0 0 1

sum(x)

## [1] 1

mean(x)

## [1] 0.3333333

x <- list(1:3, 'a', c(TRUE, FALSE, TRUE), c(2.3, 5.9))  
str(x)

## List of 4  
## $ : int [1:3] 1 2 3  
## $ : chr "a"  
## $ : logi [1:3] TRUE FALSE TRUE  
## $ : num [1:2] 2.3 5.9

x <- list(list(list(list())))  
str(x)

## List of 1  
## $ :List of 1  
## ..$ :List of 1  
## .. ..$ : list()

is.recursive(x)

## [1] TRUE

x <- list(list(1,2), c(3,4))  
y <- c(list(1,2), c(3,4))  
str(x)

## List of 2  
## $ :List of 2  
## ..$ : num 1  
## ..$ : num 2  
## $ : num [1:2] 3 4

str(y)

## List of 4  
## $ : num 1  
## $ : num 2  
## $ : num 3  
## $ : num 4

is.list(mtcars)

## [1] TRUE

mod <- lm(mpg ~ wt, mtcars)  
is.list(mod)

## [1] TRUE

y <- 1:10  
attr(y, 'my\_attribute') <- 'This is a vector'  
attr(y, 'my\_attribute')

## [1] "This is a vector"

str(attributes(y))

## List of 1  
## $ my\_attribute: chr "This is a vector"

attributes(y[1])

## NULL

attributes(sum(y))

## NULL

y <- c(a=1,2,3)  
names(y)

## [1] "a" "" ""

z <- c(1,2,3)  
names(z)

## NULL

x <- factor(c('a', 'b', 'b', 'a'))  
x

## [1] a b b a  
## Levels: a b

class(x)

## [1] "factor"

levels(x)

## [1] "a" "b"

x[2] <- 'c'

## Warning in `[<-.factor`(`\*tmp\*`, 2, value = "c"): invalid factor level, NA  
## generated

x

## [1] a <NA> b a   
## Levels: a b

c(factor('a'), factor('b'))

## [1] 1 1

sex\_char <- c('m', 'm', 'm')  
sex\_factor <- factor(sex\_char, levels = c('m', 'f'))  
table(sex\_char)

## sex\_char  
## m   
## 3

table(sex\_factor)

## sex\_factor  
## m f   
## 3 0

z <- read.csv(text = 'value\n12\n1\n.\n9')  
typeof(z$value)

## [1] "integer"

as.double(z$value)

## [1] 3 2 1 4

class(z$value)

## [1] "factor"

as.double(as.character(z$value))

## Warning: NAs introduced by coercion

## [1] 12 1 NA 9

z <- read.csv(text = 'value\n12\n1\n.\n9', na.strings='.')  
typeof(z$value)

## [1] "integer"

class(z$value)

## [1] "integer"

z$value

## [1] 12 1 NA 9

a <- matrix(1:6, ncol = 3, nrow = 2)  
b <- array(1:12, c(2,3,2))  
b

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

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

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

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

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

length(a)

## [1] 6

nrow(a)

## [1] 2

ncol(a)

## [1] 3

rownames(a) <- c("A", "B")  
colnames(a) <- c('a', 'b', 'c')  
a

## a b c  
## A 1 3 5  
## B 2 4 6

length(b)

## [1] 12

dim(b)

## [1] 2 3 2

dimnames(b) <- list(c('one', 'two'), c('a', 'b', 'c'), c('A', 'B'))  
b

## , , A  
##   
## a b c  
## one 1 3 5  
## two 2 4 6  
##   
## , , B  
##   
## a b c  
## one 7 9 11  
## two 8 10 12

str(1:3)

## int [1:3] 1 2 3

str(matrix(1:3, ncol=1))

## int [1:3, 1] 1 2 3

str(matrix(1:3, nrow=1))

## int [1, 1:3] 1 2 3

str(array(1:3, 3))

## int [1:3(1d)] 1 2 3

l <- list(1:3, 'a', TRUE, 1.0)  
dim(l) <- c(2,2)  
l

## [,1] [,2]  
## [1,] Integer,3 TRUE  
## [2,] "a" 1

df <- data.frame(x = 1:3, y = c('a', 'b', 'c'))  
str(df)

## 'data.frame': 3 obs. of 2 variables:  
## $ x: int 1 2 3  
## $ y: Factor w/ 3 levels "a","b","c": 1 2 3

df <- data.frame(  
 x = 1:3, y = c('a', 'b', 'c'), stringsAsFactors = FALSE  
)  
str(df)

## 'data.frame': 3 obs. of 2 variables:  
## $ x: int 1 2 3  
## $ y: chr "a" "b" "c"

typeof(df)

## [1] "list"

class(df)

## [1] "data.frame"

is.data.frame(df)

## [1] TRUE

cbind(df, data.frame(z=3:1))

## x y z  
## 1 1 a 3  
## 2 2 b 2  
## 3 3 c 1

rbind(df, data.frame(x = 10, y = 'z'))

## x y  
## 1 1 a  
## 2 2 b  
## 3 3 c  
## 4 10 z

df <- data.frame(x = 1:3)  
df$y <- list(1:2, 1:3, 1:4)  
print(df)

## x y  
## 1 1 1, 2  
## 2 2 1, 2, 3  
## 3 3 1, 2, 3, 4

bad <- data.frame(cbind(a=1:2, b = c('a', 'b')))  
str(bad)

## 'data.frame': 2 obs. of 2 variables:  
## $ a: Factor w/ 2 levels "1","2": 1 2  
## $ b: Factor w/ 2 levels "a","b": 1 2

#Make sure all inputs are of the same type!  
good <- data.frame(a=1:2, b = c('a', 'b'), stringsAsFactors=FALSE)  
str(good)

## 'data.frame': 2 obs. of 2 variables:  
## $ a: int 1 2  
## $ b: chr "a" "b"

# I() causes a lsit to be treated as a single unit  
df1 <- data.frame(x = 1:3, y = I(list(1:2, 1:3, 1:4)))  
str(df1)

## 'data.frame': 3 obs. of 2 variables:  
## $ x: int 1 2 3  
## $ y:List of 3  
## ..$ : int 1 2  
## ..$ : int 1 2 3  
## ..$ : int 1 2 3 4  
## ..- attr(\*, "class")= chr "AsIs"

df1[2, 'y']

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

dfm <- data.frame(x = 1:3, y = I(matrix(1:9, nrow = 3)))  
str(dfm)

## 'data.frame': 3 obs. of 2 variables:  
## $ x: int 1 2 3  
## $ y: 'AsIs' int [1:3, 1:3] 1 2 3 4 5 6 7 8 9

dfm[2, 'y']

## [,1] [,2] [,3]  
## [1,] 2 5 8