

Data Types in R

Elly Ochieng'

2024-10-30

R Markdown

```
# Data Types in R

# R treats everything as an object, and the simplest data objects are known
# as atomic data types.

# Atomic data types allow the creation of atomic vectors.
# Common atomic data types include:
# - Numeric (integer and double)
# - Character
# - Logical
# - Complex
# - Raw

# Checking if an object is atomic
is.atomic(3)          # TRUE, 3 is a numeric atomic vector

## [1] TRUE

is.atomic("R CODER")  # TRUE, "R CODER" is a character atomic vector

## [1] TRUE

# Functions to check data type in R
typeof(1)             # "double", shows the internal type

## [1] "double"

class(2)              # "numeric", shows the object's class

## [1] "numeric"

storage.mode(3)       # "double", shows storage mode

## [1] "double"

mode(4)               # "numeric", another way to check type

## [1] "numeric"

str(5)                # Displays the structure of the object, shows it is
numeric

##  num 5
```

```

# Example of changing an object's class
x <- 1                # Assigning numeric value 1
class(x)              # "numeric", initial class

## [1] "numeric"

class(x) <- "My_class" # Changing class to "My_class"
class(x)              # "My_class", confirms class change

## [1] "My_class"

typeof(x)             # "double", still shows it's a double

## [1] "double"

# Summary of Type Functions
# Outputs of typeof, storage.mode, and mode for various types
# | Function          | logical | integer | double | character | raw |
# |-----|-----|-----|-----|-----|-----|
# | typeof            | logical | numeric | double | character | raw |
# | storage.mode      | logical | numeric | double | character | raw |
# | mode              | logical | integer | double | character | raw |

# Numeric Data Types
# Numeric types consist of double and integer.
mode(55)              # "numeric"

## [1] "numeric"

is.numeric(3)         # TRUE, confirms it's numeric

## [1] TRUE

# Double or Real Data Type
# Double-precision representation is default for all numbers
typeof(2)             # "double"

## [1] "double"

typeof(Inf)           # "double", shows infinity

## [1] "double"

typeof(-Inf)          # "double", shows negative infinity

## [1] "double"

typeof(NaN)           # "double", represents "Not a Number"

## [1] "double"

typeof(3.12e3)        # "double", shows scientific notation

## [1] "double"

```

```

typeof(0xbade)          # "double", hexadecimal notation
## [1] "double"

# Check if an object is double
is.double(2)            # TRUE
## [1] TRUE

is.double(2.8)          # TRUE
## [1] TRUE

# Integer Data Type
# Create integers by appending L to a number
y <- 2L                 # Creates integer
typeof(y)               # "integer"
## [1] "integer"

is.integer(3)           # FALSE, 3 is a double
## [1] FALSE

is.integer(3L)          # TRUE, 3L is an integer
## [1] TRUE

# Logical Data Type
# Composed of TRUE, FALSE, and NA
t <- TRUE               # Assign TRUE
f <- FALSE              # Assign FALSE
n <- NA                 # Assign NA
typeof(t)               # "logical"
## [1] "logical"

typeof(f)               # "logical"
## [1] "logical"

typeof(n)               # "logical"
## [1] "logical"

# Check if an object is logical
is.logical(T)           # TRUE
## [1] TRUE

is.logical(TRUE)        # TRUE
## [1] TRUE

```

```

# Caution with T and F
# Using T and F can override the values
F                                # FALSE, the default

## [1] FALSE

a <- T                          # Assigns TRUE to variable a
F <- a                          # Now F is TRUE

# Complex Data Type
# Includes imaginary numbers
1 + 3i                          # Represents a complex number

## [1] 1+3i

typeof(1 + 3i)                  # "complex"

## [1] "complex"

is.complex(1 + 3i)              # TRUE

## [1] TRUE

# String or Character Data Type
# Character strings are enclosed in quotes
character <- "a"                # Assigns a string
typeof(character)               # "character"

## [1] "character"

is.character(character)         # TRUE

## [1] TRUE

typeof('R CODER')              # "character"

## [1] "character"

typeof("R CODER")              # "character"

## [1] "character"

nchar("A string")               # 8, counts characters including spaces

## [1] 8

# Raw Data Type in R
# Holds raw bytes and is less common
a <- charToRaw("R CODER")       # Converts string to raw bytes
# Outputs raw byte representation
typeof(a)                       # "raw"

## [1] "raw"

```

```

b <- intToBits(3L)      # Converts integer to raw bits
typeof(b)               # "raw"

## [1] "raw"

is.raw(b)               # TRUE

## [1] TRUE

# Date and Time Data Type in R
# Dates can be represented using as.Date()
date_example <- as.Date("2024-10-30") # Example of creating a date object

# Data Types Coercion in R
# Coerce data types using functions that start with as.
# | Function          | Coerced Data Type |
# |-----|-----|
# | as.numeric        | Numeric           |
# | as.integer        | Integer           |
# | as.double         | Double            |
# | as.character      | Character          |
# | as.logical        | Boolean            |
# | as.raw            | Raw                |

# Example of coercing a double to integer
a <- 3                  # double by default
typeof(a)               # "double"

## [1] "double"

a <- as.integer(a)      # coerces to integer
typeof(a)               # "integer"

## [1] "integer"

# Coerce Logical values
b <- TRUE
b <- as.numeric(b)      # Coerces TRUE to 1
b                        # Outputs 1

## [1] 1

c <- FALSE
c <- as.numeric(c)      # Coerces FALSE to 0
c                        # Outputs 0

## [1] 0

d <- TRUE
d <- as.character(d)    # Coerces TRUE to string "TRUE"
d                        # Outputs "TRUE"

```

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

```
# Attempting to coerce incompatible types will yield an error
```

```
as.double("R CODER") # Outputs NA with a warning: NAs introduced by coercion
```

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