R Programming Reference Sheet

Atomic Types in R

Atomic types are the most basic data types in R. Everything in R is an object, and atomic types are the building blocks of data objects.

Numeric: Default type for numbers (floating-point)
Integer: Whole numbers (e.g., 2L)
Character: Strings or textual data
Logical: Boolean values: TRUE, FALSE
Complex: Complex numbers (e.g., 1+2i)

Examples - Atomic Types

```
x <- 3.14        # Numeric
typeof(x)        # "double"

y <- 5L          # Integer
typeof(y)        # "integer"

z <- "Hello"     # Character
typeof(z)        # "character"

a <- TRUE        # Logical
typeof(a)        # "logical"

c <- 1 + 2i      # Complex
typeof(c)        # "complex"
```

Data Structures in R

Vectors: Homogeneous, created using c()
Lists: Heterogeneous, hold various types
Matrices: 2D homogeneous, created using matrix()
Arrays: n-dimensional generalization of matrices
Data Frames: 2D heterogeneous, created using data.frame()
Factors: Categorical data, stored as integers with labels

Examples - Data Structures

```
v <- c(1, 2, 3)
lst <- list(1, "a", TRUE, 1+2i)
```

R Programming Reference Sheet

```
m <- matrix(1:9, nrow=3, ncol=3)
a <- array(1:8, dim = c(2, 2, 2))
df <- data.frame(Name=c("Tom", "Anna"), Age=c(25, 28), IsStudent=c(TRUE, FALSE))
f <- factor(c("low", "medium", "high", "medium"))
```

Type Checking and Coercion

Type Checking Functions:
is.numeric(), is.integer(), is.character(), is.logical(), is.complex(),
is.vector(), is.matrix(), is.list(), is.data.frame()

Type Coercion Hierarchy:
Logical -> Integer -> Numeric -> Complex -> Character

```
v <- c(TRUE, 2L, 3.5, "text")  # All elements become character
```

Explicit Conversion:
as.numeric("5"), as.integer("5"), as.logical(1), as.character(3.14)

Summary Table

Vector: 1D, Homogeneous, Single Type
List: 1D, Heterogeneous, Multiple Types
Matrix: 2D, Homogeneous, Single Type
Array: nD, Homogeneous, Single Type
Data Frame: 2D, Heterogeneous (column-wise)
Factor: 1D, Categorical (internally integers)