R Data Types: integer, character, numeric, logical, factor, complex, and date

Introduction

R supports several fundamental data types that allow you to work with various forms of data. Understanding these data types is crucial for writing efficient and correct R scripts. This section covers `integer`, `character`, `numeric`, `logical`, `factor`, `complex`, and `date` types, along with examples and exercises.

1. Integer

- Used to store whole numbers.
- Created using the `as.integer()` function or by appending an `L` to a number.

Example:

```
x <- 5L       # Integer value
y <- as.integer(10)  # Convert to integer
class(x)      # Output: "integer"
```

2. Character

- Used to store text strings.
- Created by enclosing text in double or single quotes.

Example:

```
name <- "Alice"  # Character string
greeting <- paste("Hello,", name)  # Concatenates strings
```

3. Numeric

- Default data type for numbers in R, including integers and decimals.
- Created by assigning a number to a variable.

Example:

```
x <- 3.14       # Numeric
y <- 100        # Also treated as numeric
class(x)        # Output: "numeric"
```

4. Logical

- Represents TRUE or FALSE values.
- Often used in conditional statements and logical operations.

Example:

```
a <- TRUE
b <- FALSE
result <- a & b  # Logical AND
```

5. Factor

- Used to represent categorical data.
- Levels are assigned to each unique value.

Example:

```
categories <- factor(c("small", "medium", "large"))
levels(categories)  # Output: "large", "medium", "small"
```

6. Complex

- Represents complex numbers with real and imaginary parts.

Example:

```
z <- 3 + 4i  # Complex number
Re(z)        # Real part
Im(z)        # Imaginary part
```

7. Date

- Used to store and manipulate date values.
- Created using as.Date() or strptime().

Example:

```
today <- as.Date("2024-01-01")
formatted_date <- strptime("01-01-2024", "%d-%m-%Y")
```

Exercises

1. Create an integer variable and verify its class.
2. Convert a numeric value to an integer using `as.integer()`.
3. Write a script to concatenate two character strings and print the result.
4. Create a numeric variable with a decimal value and calculate its square.
5. Assign `TRUE` and `FALSE` to two variables and calculate their logical AND.
6. Create a factor variable representing three categories: ”low”, ”medium”, ”high”.
7. Write a script to display the levels of a factor variable.
8. Create a complex number and extract its real and imaginary parts.
9. Write a script to calculate the magnitude of a complex number.
10. Assign the current date to a variable using `Sys.Date()` and print it.
11. Write a script to calculate the number of days between two dates.
12. Create a vector of dates and sort them in ascending order.
13. Convert a date string in the format `"dd-mm-yyyy"` to a `Date` object.
14. Create a character vector and convert it to a factor.
15. Write a script to classify a numeric variable into categories using a factor.

Conclusion

These exercises cover R’s core data types. Mastery of these concepts is essential for working with data in R effectively and efficiently.