STA 3100 Programming With Data in R: Data Structures

Data Structures in R

Overview

Data structures are the way data is organized and stored in a computer. In R, there are several data structures that can be used to store and manipulate data. These data structures include vectors, matrices, arrays, lists, and data frames. Each of these data structures has its own set of characteristics and functions that allow for efficient data manipulation. Understanding how to use each of these data structures is essential for programming with data in R.

Vectors

A vector is a one-dimensional array of elements. Vectors can contain numeric, character, or logical values. Vectors are created using the `c()` function.

```
Start of Code
# Create a vector of numbers
x <- c(1, 2, 3, 4, 5)
# Create a vector of characters
y <- c("a", "b", "c")
# Create a vector of logical values
z <- c(TRUE, FALSE, TRUE)
End of Code
```

Vectors can be combined using the `c()` function as well.

```
Start of Code
# Combine two vectors
combined_vector <- c(x, y, z)
End of Code
```

Vectors can also be subsetted using the `[]` operator.

```
Start of Code
# Subset the first three elements of the combined vector
subset_vector <- combined_vector[1:3]
End of Code
```

Matrices

A matrix is a two-dimensional array of elements. Matrices can contain numeric, character, or logical values. Matrices are created using the `matrix()` function.

```
Start of Code
# Create a matrix of numbers
x <- matrix(1:9, nrow = 3, ncol = 3)
# Create a matrix of characters
y <- matrix(c("a", "b", "c", "d", "e", "f", "g", "h", "i"), nrow = 3, ncol = 3)
# Create a matrix of logical values
z <- matrix(c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE), nrow = 3, ncol = 3)
End of Code
```

Matrices can be combined using the `cbind()` and `rbind()` functions.

```
Start of Code
# Combine two matrices
combined_matrix <- cbind(x, y, z)
End of Code
```

Matrices can also be subsetted using the `[]` operator.

```
Start of Code
# Subset the first three elements of the combined matrix
subset_matrix <- combined_matrix[1:3, 1:3]
End of Code
```

Arrays

An array is a multi-dimensional array of elements. Arrays can contain numeric, character, or logical values. Arrays are created using the `array()` function.

```
Start of Code
# Create an array of numbers
x <- array(1:27, dim = c(3, 3, 3))
# Create an array of characters
y <- array(c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z"), dim = c(3, 3, 3))
# Create an array of logical values
z <- array(c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE), dim = c(3, 3, 3))
End of Code
```

Arrays can be combined using the `abind()` function.

```
Start of Code
```

```
# Combine two arrays
combined_array <- abind(x, y, z)
End of Code
```

Arrays can also be subsetted using the `[]` operator.

```
Start of Code
# Subset the first three elements of the combined array
subset_array <- combined_array[1:3, 1:3, 1:3]
End of Code
```

Lists

A list is a collection of elements of different types. Lists can contain numeric, character, or logical values. Lists are created using the `list()` function.

```
Start of Code
# Create a list
x <- list(1, "a", TRUE)
End of Code
```

Lists can also be combined using the `c()` function.

```
Start of Code
# Combine two lists
combined_list <- c(x, list(2, "b", FALSE))
End of Code
```

Lists can also be subsetted using the `[[]]` operator.

```
Start of Code
# Subset the first three elements of the combined list
subset_list <- combined_list[[1:3]]
End of Code
```

Data Frames

A data frame is a two-dimensional array of elements with labeled rows and columns. Data frames can contain numeric, character, or logical values. Data frames are created using the `data.frame()` function.

```
Start of Code
# Create a data frame
x <- data.frame(name = c("John", "Jane", "Joe"), age = c(20, 21, 22))
End of Code
```

Data frames can also be combined using the `rbind()` and `cbind()` functions.

```
Start of Code
# Combine two data frames
combined_df <- rbind(x, data.frame(name = "Jill", age = 23))
End of Code
```

Data frames can also be subsetted using the `[]` operator.

```
Start of Code
# Subset the first three elements of the combined data frame
subset_df <- combined_df[1:3, ]
End of Code
```

Practice Multiple Choice Questions

Q1. What is the correct way to create a vector in R?

A. `vector()`

B. `c()`

C. `array()`

D. `list()`

Answer: B. `c()`