2. Hafta: R'da Veri Türleri ve Temel Fonksiyonlar

Hakan Mehmetcik

2024-10-09

1. Değişkenler ve Veri Türleri

R dilinde, verileri saklamak ve işlemek için değişkenler kullanılır. Değişkenler, belirli bir değeri tutar ve bu değer üzerinde işlemler yapmanıza olanak tanır.

Değişken Tanımlama (<- Operatörü) R'da bir değişken tanımlamak için <- operatörü kullanılır. Bu operatör, sağdaki değeri soldaki değişkene atar.

```
x <- 10   # x değişkenine 10 değerini atar
y <- "Merhaba"   # y değişkenine bir karakter dizisi atar
```

R'da Veri Türleri

R'da değişkenlerin farklı veri türleri olabilir. R'ın sık kullanılan veri türleri şunlardır:

- **Sayısal (Numeric)**: Ondalıklı veya tam sayı olarak saklanan sayılar.

  ```
  a <- 5.7 # Ondalıklı bir sayısal veri
  b <- 10 # Tam sayı olarak bir sayısal veri
  ```

- **Karakter (Character)**: Metin veya harf dizisi olarak saklanan veriler.

  ```
  c <- "R programlama dili" # Karakter veri türünde bir veri
  ```

- **Mantıksal (Logical)**: TRUE veya FALSE değerlerini saklayan değişkenler.

  ```
  durum <- TRUE   # Logical
  ```

Değişkenlerle Matematiksel İşlemler

Sayısal veri türüne sahip değişkenlerde matematiksel işlemler yapabilirsiniz. Örneğin:

```
a <- 5
b <- 3
```

Toplama

```
toplam <- a + b
print(toplam) # Çıktı: 8
```

```
[1] 8
```

Çıkarma

```
cikarma <- a - b
print(cikarma) # Çıktı: 2
```

```
[1] 2
```

Çarpma

```
a*b # Çıktı: 15
```

```
[1] 15
```

Bölme

```
bolme <- a / b
print(bolme) # Çıktı: 1.666667
```

```
[1] 1.666667
```

Üs Alma

```r
us <- a^b
print(us) # Çıktı: 125
```

```
[1] 125
```

3. R'da Temel Fonksiyonlar

R programlama dilinde temel fonksiyonlar, veri analizi ve istatistiksel hesaplamalar için yaygın olarak kullanılır. İşte bazı temel fonksiyonlar ve örnek kullanımları:

R'ın Yerleşik Fonksiyonları Yerleşik fonksiyonlar, R dilinin varsayılan olarak sunduğu işlevlerdir. Örneğin, mean() fonksiyonu bir vektörün ortalamasını hesaplar, sum() toplamını verir ve length() bir vektörün kaç eleman içerdiğini gösterir.

Örnekler:

1. print() Bir nesneyi konsola yazdırır.

   ```r
   print("Merhaba Dünya")
   ```

   ```
   [1] "Merhaba Dünya"
   ```

2. sum(): Bir vektörün elemanlarının toplamını hesaplar.

   ```r
   toplam <- sum(c(1, 2, 3, 4, 5))
   ```

3. mean(): Bir vektörün elemanlarının ortalamasını hesaplar.

   ```r
   ortalama <- mean(c(1, 2, 3, 4, 5))
   ```

4. sd(): Bir vektörün standart sapmasını hesaplar.

   ```r
   standart_sapma <- sd(c(1, 2, 3, 4, 5))
   ```

5. length(): Bir vektörün uzunluğunu döndürür.

   ```r
   uzunluk <- length(c(1, 2, 3, 4, 5))
   ```

6. seq(): Belirli bir aralıkta ardışık sayılar oluşturur.

   ```r
   dizi <- seq(1, 10, by=2)
   ```

7. rep(): Bir değeri belirli sayıda tekrar eder

   ```r
   tekrar <- rep(1, times=5)
   ```

8. c(): Vektör oluşturur.

```r
vektor <- c(1, 2, 3, 4, 5)
```

9. data.frame(): Veri çerçevesi oluşturur.

```r
veri <- data.frame( isim = c("Ali", "Ayşe", "Fatma"), yas = c(25, 30, 35) )
```

10. summary(): Bir nesnenin özet istatistiklerini döndürür.

```r
ozet <- summary(c(1, 2, 3, 4, 5))
```

Parametrelerle Fonksiyon Kullanımı

Fonksiyonlar, belirli bir işlevi gerçekleştirmek için parametre alabilir. Parametreler, fonksiyonun nasıl çalışacağını belirler. Örneğin, mean() fonksiyonuna na.rm = TRUE parametresi eklenerek eksik değerleri göz ardı edebilirsiniz.

```r
sayilar <- c(10, 20, NA, 40, 50)
ortalama <- mean(sayilar, na.rm = TRUE)   # NA değerini göz ardı ederek ortalama hesaplar
```

Fonksiyon Oluşturma

R dilinde kendi fonksiyonlarınızı oluşturmak oldukça basittir. Fonksiyonlar, belirli bir işlevi gerçekleştirmek için bir dizi komut içerir ve gerektiğinde parametreler alabilir. Fonksiyonlar, function anahtar kelimesi kullanılarak tanımlanır.

Fonksiyon Oluşturma Bir fonksiyon oluşturmak için aşağıdaki adımları izleyebilirsiniz:

- function anahtar kelimesini kullanarak fonksiyonunuzu tanımlayın.
- Fonksiyonun alacağı parametreleri parantez içinde belirtin.
- Fonksiyonun gövdesinde, fonksiyonun gerçekleştireceği işlemleri yazın.
- return() fonksiyonunu kullanarak fonksiyonun döndüreceği değeri belirtin (isteğe bağlı).

```r
# Toplama fonksiyonu oluşturma
topla <- function(a, b) {
  sonuc <- a + b
  return(sonuc)
}

# Fonksiyonu çağırma
toplam <- topla(5, 3)
print(toplam)  # 8
```

```
[1] 8
```

R Fonksiyonları İçin Yardım Alma (? ve help() Komutları)

Bir R fonksiyonu hakkında bilgi almak için ? veya help() komutlarını kullanabilirsiniz. Bu komutlar, ilgili fonksiyonun nasıl kullanılacağı ve parametrelerinin ne olduğu hakkında detaylı bilgi verir.

Örnekler:

```
?mean   # mean() fonksiyonu hakkında yardım alır
help(sum)  # sum() fonksiyonu hakkında yardım alır
```

Bu komutlar, fonksiyonun açıklamasını, hangi argümanları kabul ettiğini ve nasıl çalıştırılacağını gösteren bir dokümantasyon penceresi açar.

Belgeleme ve Fonksiyonların Anlamı

R'ın tüm fonksiyonlarının nasıl çalıştığını, örnekler ve açıklamalarla birlikte görebilirsiniz. R'da her fonksiyonun geniş bir açıklaması bulunur ve bu belgeleme, fonksiyonların nasıl kullanılacağını öğrenmenin etkili bir yoludur.

Örnek:

```
help(mean)  # Ortalama hesaplayan mean() fonksiyonunu anlamak için yardım alabiliriz
```

4. Basit Veri Yapıları

R dilinde, veriler çeşitli veri yapıları kullanılarak organize edilir. Temel veri yapıları, vektörler, listeler ve matrislerdir.

Vektörler

R dilinde en temel veri yapılarından biri vektörlerdir. Vektörler, aynı veri türüne sahip bir dizi elemanı saklar. Vektör oluşturmak için c() fonksiyonu kullanılır.

```r
# Sayısal vektör oluşturma
numeric_vector <- c(1, 2, 3, 4, 5)

# Karakter vektör oluşturma
character_vector <- c("a", "b", "c", "d")

# Mantıksal vektör oluşturma
logical_vector <- c(TRUE, FALSE, TRUE)

# Vektör elemanlarına erişim
print(numeric_vector[1])  # 1
```

```
[1] 1
```

```r
print(character_vector[3])  # "c"
```

```
[1] "c"
```

```r
# Vektör elemanlarını güncelleme
numeric_vector[2] <- 10

# Vektör toplama
sum_vector <- numeric_vector + c(1, 1, 1, 1, 1)

# Vektör elemanlarının toplamı
total_sum <- sum(numeric_vector)

# Güncellenmiş vektörleri yazdırma
print(numeric_vector)
```

```
[1]  1 10  3  4  5
```

```r
print(sum_vector)
```

```
[1]  2 11  4  5  6
```

```r
print(total_sum)
```

```
[1] 23
```

Listeler

Listeler, farklı türdeki verileri bir arada saklayabilen veri yapılarıdır. Bir liste içinde sayılar, karakterler, mantıksal değerler ve hatta başka vektörler bulunabilir.

```
# Liste oluşturma
my_list <- list(number = 5, greeting = "Hello", flag = TRUE, numbers = c(1, 2, 3))

# Liste elemanlarına erişim
print(my_list$number)  # 5
```

```
[1] 5
```

```
print(my_list$greeting)  # "Hello"
```

```
[1] "Hello"
```

```
print(my_list$flag)  # TRUE
```

```
[1] TRUE
```

```
print(my_list$numbers)  # c(1, 2, 3)
```

```
[1] 1 2 3
```

```
# Liste elemanlarını güncelleme
my_list$number <- 10
my_list$greeting <- "Hi"

# Listeye yeni eleman ekleme
my_list$new_element <- "New Element"

# Liste elemanlarını silme
my_list$new_element <- NULL

# Güncellenmiş listeyi yazdırma
print(my_list)
```

```
$number
[1] 10

$greeting
[1] "Hi"

$flag
[1] TRUE

$numbers
[1] 1 2 3
```

Matrisler

Matrisler, satırlar ve sütunlar şeklinde organize edilmiş sayısal veri yapılarıdır. Tüm elemanlar aynı türde olmalıdır.

```r
# 2x3 boyutunda bir matris oluşturma
my_matrix <- matrix(1:6, nrow = 2, ncol = 3)

# Matris elemanlarına erişim
print(my_matrix[1, 2])  # 3
```

```
[1] 3
```

```r
print(my_matrix[2, ])  # c(4, 5, 6)
```

```
[1] 2 4 6
```

```r
print(my_matrix[, 3])  # c(5, 6)
```

```
[1] 5 6
```

```r
# Matris elemanlarını güncelleme
my_matrix[1, 2] <- 10

# Matris toplama
matrix1 <- matrix(1:4, nrow = 2)
matrix2 <- matrix(5:8, nrow = 2)
```

```r
sum_matrix <- matrix1 + matrix2

# Matris çarpma
product_matrix <- matrix1 %*% matrix2

# Güncellenmiş matrisleri yazdırma
print(my_matrix)
```

```
     [,1] [,2] [,3]
[1,]    1   10    5
[2,]    2    4    6
```

```r
print(sum_matrix)
```

```
     [,1] [,2]
[1,]    6   10
[2,]    8   12
```

```r
print(product_matrix)
```

```
     [,1] [,2]
[1,]   23   31
[2,]   34   46
```

Data Frameler

Data frameler, R dilinde kullanılan ve farklı veri türlerine sahip sütunları içerebilen iki boyutlu veri yapılarıdır. Data frameler, genellikle tablo şeklinde verileri saklamak için kullanılır ve her sütun bir vektör olarak temsil edilir. Data frameler, data.frame() fonksiyonu kullanılarak oluşturulur.

```r
# Data frame oluşturma
my_data_frame <- data.frame(
  numbers = c(1, 2, 3, 4, 5),
  letters = c("a", "b", "c", "d", "e"),
  logicals = c(TRUE, FALSE, TRUE, FALSE, TRUE)
)

# Data frame elemanlarına erişim
print(my_data_frame[1, 2])  # "a"
```

```
[1] "a"
```

```r
print(my_data_frame[2, ])  # c(2, "b", FALSE)
```

```
  numbers letters logicals
2       2       b    FALSE
```

```r
print(my_data_frame[, 3])  # c(TRUE, FALSE, TRUE, FALSE, TRUE)
```

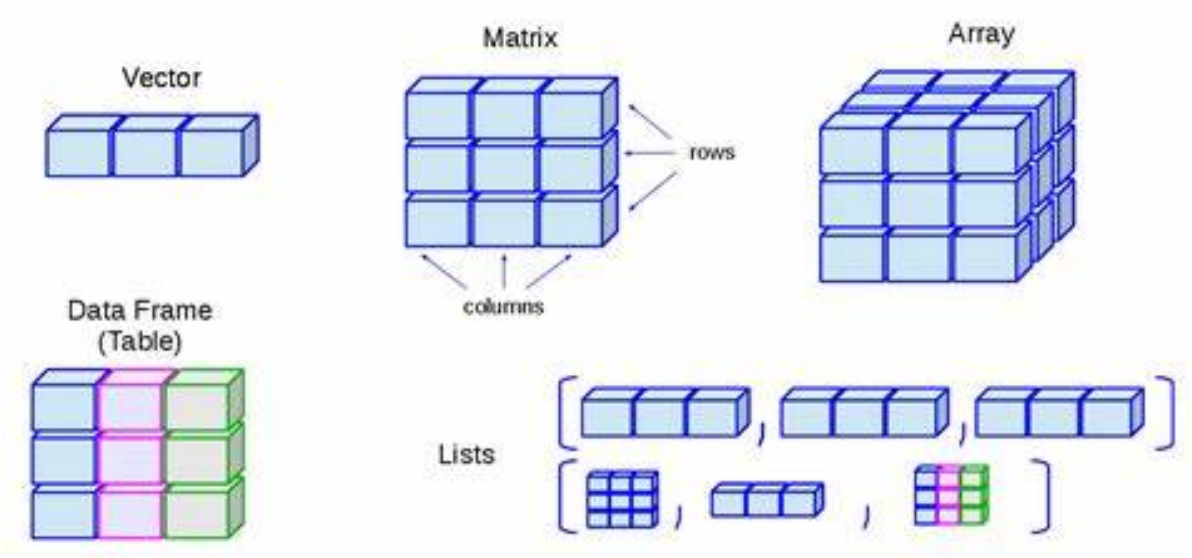
```
[1]  TRUE FALSE  TRUE FALSE  TRUE
```

```r
print(my_data_frame$numbers)  # c(1, 2, 3, 4, 5)
```

```
[1] 1 2 3 4 5
```

```r
# Data frame elemanlarını güncelleme
my_data_frame[1, 2] <- "z"
my_data_frame$numbers[2] <- 10

# Yeni bir sütun ekleme
my_data_frame$new_column <- c(10, 20, 30, 40, 50)

# Satır ekleme
new_row <- data.frame(numbers = 6, letters = "f", logicals = FALSE, new_column = 60)
my_data_frame <- rbind(my_data_frame, new_row)

# Sütun ekleme
new_column <- c(100, 200, 300, 400, 500, 600)
my_data_frame <- cbind(my_data_frame, new_column)

# Güncellenmiş data frame'i yazdırma
print(my_data_frame)
```

```
  numbers letters logicals new_column new_column
1       1       z     TRUE         10        100
2      10       b    FALSE         20        200
3       3       c     TRUE         30        300
4       4       d    FALSE         40        400
5       5       e     TRUE         50        500
6       6       f    FALSE         60        600
```



5. Veri İçe Aktarma ve Kaydetme

Veri içe aktarma ve temel manipülasyon, veri analizi sürecinin önemli adımlarıdır. R dilinde veri içe aktarma ve temel manipülasyon işlemleri için çeşitli fonksiyonlar ve paketler kullanılır.

Veri İçe Aktarma

R dilinde veri içe aktarmak için read.csv(), read.table(), read_excel() gibi fonksiyonlar kullanılır.

```
# CSV dosyasını içe aktarma
data <- read.csv("https://raw.githubusercontent.com/datasciencedojo/datasets/refs/heads/maste
```

```
# readxl paketini yükleme
# install.packages("readxl")
library(readxl)

# Excel dosyasını içe aktarma
data <- read_excel("~/Desktop/IST2083/Data/sample_excel.xlsx")
```

Veri Kaydetme

R dilinde veri kaydetmek için çeşitli fonksiyonlar kullanabilirsiniz. En yaygın kullanılan dosya formatları arasında CSV, Excel ve RDS dosyaları bulunur.

CSV Dosyasına Veri Kaydetme Veri çerçevesini CSV dosyasına kaydetmek için write.csv() fonksiyonunu kullanabilirsiniz:

```
# Veri çerçevesini CSV dosyasına kaydetme
# write.csv(data, "path/to/your/file.csv", row.names = FALSE)
```

Excel Dosyasına Veri Kaydetme Excel dosyasına veri kaydetmek için writexl paketini kullanabilirsiniz:

```
# writexl paketini yükleme
# install.packages("writexl")
# library(writexl)

# Veri çerçevesini Excel dosyasına kaydetme
# write_xlsx(data, "path/to/your/file.xlsx")
```

Veri Yapısını İnceleme

Veri inceleme, veri analizi sürecinin önemli bir parçasıdır. R dilinde veri incelemek için çeşitli fonksiyonlar ve paketler kullanabilirsiniz. Bu işlemler arasında veri yapısını kontrol etme, özet istatistikler oluşturma ve veri görselleştirme gibi işlemler bulunur. Veri yapısını incelemek için str(), summary(), head(), ve tail() fonksiyonlarını kullanabilirsiniz.

```
# Veri yapısını kontrol etme
str(data)
```

```
tibble [46 x 8] (S3: tbl_df/tbl/data.frame)
 $ Product ID                   : chr [1:46] "P101" "P102" "P103" "P104" ...
 $ Product Name                 : chr [1:46] "Laptop" "Monitor" "Keyboard" "Headphones" ...
 $ Opening
Stock            : num [1:46] 50 40 60 30 70 45 55 25 35 40 ...
 $ Purchase/
Stock in        : num [1:46] 20 15 25 10 30 18 22 12 15 20 ...
 $ Number of
Units Sold     : num [1:46] 10 5 15 3 20 8 12 5 7 10 ...
 $ Hand-In-
```

```
Stock            : num [1:46] 60 50 70 37 80 55 65 32 43 50 ...
 $ Cost Price
Per Unit (USD): num [1:46] 1200 500 50 100 900 700 150 200 80 60 ...
 $ Cost Price
Total (USD)    : num [1:46] 72000 25000 3500 3700 72000 38500 9750 6400 3440 3000 ...
```

```r
# Özet istatistikler
summary(data)
```

```
  Product ID        Product Name       Opening \r\nStock Purchase/\r\nStock in
 Length:46          Length:46          Min.   :15.00     Min.   : 6.00
 Class :character   Class :character   1st Qu.:25.00     1st Qu.:10.00
 Mode  :character   Mode  :character   Median :35.00     Median :15.00
                                       Mean   :35.98     Mean   :15.37
                                       3rd Qu.:43.75     3rd Qu.:20.00
                                       Max.   :70.00     Max.   :30.00
 Number of \r\nUnits Sold Hand-In-\r\nStock Cost Price \r\nPer Unit (USD)
 Min.   : 2.000           Min.   :17.00     Min.   :   5.0
 1st Qu.: 4.000           1st Qu.:31.00     1st Qu.:  20.0
 Median : 6.000           Median :41.00     Median :  60.0
 Mean   : 6.826           Mean   :43.57     Mean   : 155.8
 3rd Qu.: 9.000           3rd Qu.:54.25     3rd Qu.: 150.0
 Max.   :20.000           Max.   :80.00     Max.   :1200.0
 Cost Price\r\nTotal (USD)
 Min.   :  145.0
 1st Qu.:  757.5
 Median : 3000.0
 Mean   : 7820.9
 3rd Qu.: 4760.0
 Max.   :72000.0
```

```r
# İlk birkaç satırı görüntüleme
head(data)
```

|Product ID |Product Name | Opening Stock| Purchase/ Stock in| Number of Units Sold| Hand-In- Stock| Cost Price Per Unit (USD)| Cost Price Total (USD)| |:———-|:————|———-:|————:|————:|———-:|————————:|————————:| |P101 |Laptop | 50| 20| 10| 60| 1200| 72000| |P102 |Monitor | 40| 15| 5| 50| 500| 25000| |P103 |Keyboard | 60| 25| 15| 70| 50| 3500| |P104 |Headphones | 30| 10| 3| 37| 100| 3700| |P105 |Smartphone | 70| 30| 20| 80| 900| 72000| |P106 |Tablet | 45| 18| 8| 55| 700| 38500|

```
# Son birkaç satırı görüntüleme
tail(data)
```

Product ID	Product Name	Opening Stock	Purchase/ Stock in	Number of Units Sold	Hand-In- Stock	Cost Price Per Unit (USD)	Cost Price Total (USD)
P141	Anti-Glare Screen Protector	25	8	3	28	10	280
P142	USB-C Adapter	20	10	4	24	15	360
P143	Laptop Sleeve	30	12	5	37	20	740
P144	Wireless Charger	40	18	7	51	30	1530
P145	USB-C Cable	50	20	9	61	8	488
P146	Gaming Desk	25	10	3	28	150	4200