Lab 2 - Data Structures in R

Environmental Data Analytics

John Fay & Luana Lima

Fall 2024

Objective

Discuss and navigate different data types in R - Vectors - Matrices - Arrays - Data frames

---

Data Types in R

R treats objects differently based on their characteristics. For more information, please see: https://www.statmethods.net/input/datatypes.html.

- **Vectors** 1 dimensional structure that contains elements of the same type.
- **Matrices** 2 dimensional structure that contains elements of the same type.
- **Arrays** Similar to matrices, but can have more than 2 dimensions. We will not delve into arrays in depth.
- **Lists** Ordered collection of elements that can have different modes.
- **Data Frames** 2 dimensional structure that is more general than a matrix. Columns can have different modes (e.g., numeric and factor). When we import csv files into the R workspace, they will enter as data frames.

Vectors

Creating Vectors

Add comments below that specify what kind of vectors are created. (First answer is provided.)

```
# Vectors ----
vector1 <- c(1,2,5.3,6,-2,4) # Vector type: Numeric
vector1
```

```
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
```

```r
class(vector1)
```

```
## [1] "numeric"
```

```r
vector2 <- c("one","two","three") # Vector type: Character
vector2
```

```
## [1] "one"   "two"   "three"
```

```r
vector3 <- c(TRUE,TRUE,TRUE,FALSE,TRUE,FALSE) # Vector type: Logical
vector3
```

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

Vector Operations

Add comments below to indicate what each operation does.(First answer is provided.)

```r
# Returns the 3rd item in the vector
vector1[3]
```

```
## [1] 5.3
```

```r
# Gives the maximum value
max(vector1)
```

```
## [1] 6
```

```r
#Create another vector of the values 10 through 15
vector4 <- c(10:15)

# Creates a new vector with the values of vector 1 followed by vector 4
c(vector1, vector4)
```

```
##  [1]  1.0  2.0  5.3  6.0 -2.0  4.0 10.0 11.0 12.0 13.0 14.0 15.0
```

```r
# multiplies the values in the same index in vector 1 and 4 together
vector1 * vector4
```

```
## [1]  10.0  22.0  63.6  78.0 -28.0  60.0
```

```r
# multiplies each value in vector 1 by 10
vector1 * 10
```

```
## [1]  10  20  53  60 -20  40
```

```r
# changes all characters to uppercase
toupper(vector2)
```

```
## [1] "ONE"   "TWO"   "THREE"
```

```r
# Gives the opposite value of vector 3 - can only be applied to logical objects
!vector3
```

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

Exercise 1a: Vector

Find a 5-day forecast of temperatures (Fahrenheit) for Durham, North Carolina. Create a vector representing the high temperature on each of the five days. https://www.wunderground.com/forecast/us/nc/durham/KNCDURHA138

```r
#Assign to the variable 'temps_F' a vector of high temperatures in F
temps_F <- c(80,82,80,75,74)

#What is the average of these temperatures: 78.2
temps_avg <- mean(temps_F)

#Compute the difference between each temperature and the average calculated above
temps_diff <- temps_F - temps_avg

#Subtract 32 and multiply by 5/9ths to create a vector in Celsius
temps_C <- (temps_F-32)* 5/9
```

Matrices

###Creating Matrices Complete the comments below:

```r
# Matrices ----
matrix1 <- matrix(1:20, nrow = 5,ncol = 4) #
matrix1
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```r
#matrix1 has 5 rows and 4 columns

matrix2 <- matrix(1:20, nrow = 5, ncol = 4, byrow = TRUE) #
matrix2
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
```

```
## [2,]    5    6    7    8
## [3,]    9   10   11   12
## [4,]   13   14   15   16
## [5,]   17   18   19   20
```

```
#matrix2 has 5 rows and 4 columns

matrix3 <- matrix(
  data = 1:20,
  nrow = 5,
  ncol = 4,
  byrow = TRUE, # return after comma continues the line
  dimnames = list(
    c("uno", "dos", "tres", "quatro", "cinco"), #names rows
    c("un", "deux", "trois", "quatre") #names columns
    )
  )
```

Matrix Operations

Add comments in the code chunk below to indicate type of vectors created and what each operation does.

```
matrix1[4, 1] # provide the value in the 4th row and 1st column of matrix1
```

```
## [1] 4
```

```
matrix1[4,  ] # provide all values in the 4th row of matrix1
```

```
## [1]  4  9 14 19
```

```
matrix1[ , 3] # provide all values in the 3rd column of matrix1
```

```
## [1] 11 12 13 14 15
```

```
matrix1[c(12, 14)] # provide the 12th and 14th value in matrix1
```

```
## [1] 12 14
```

```
matrix1[c(12:14)] # provide the 12th to 14th values in matrix1
```

```
## [1] 12 13 14
```

```
matrix1[2:4, 1:3] # provide all values in rows 2-4 within columns 1-3 of matrix1
```

```
##      [,1] [,2] [,3]
## [1,]    2    7   12
## [2,]    3    8   13
## [3,]    4    9   14
```

```r
#Create a vector of 6 values
values <- c(1, 26, 24, 68, 3, 44)
#Create a vector of row names
rnames <- c("R1", "R2")
#Create a vector of column names
cnames <- c("C1", "C2", "C3")

matrix4 <- matrix(
  data = values,
  nrow = 2,
  ncol = 3,
  byrow = TRUE,
  dimnames = list(
    rnames,
    cnames)
  ) # creates a matrix out of the vector values with row and column names from the vectors rnames and c

matrix4 # gets the value that is in row 2 and column 1 by calling on the dimension names
```

```
##    C1 C2 C3
## R1  1 26 24
## R2 68  3 44
```

```r
matrix4['R2','C1']
```

```
## [1] 68
```

Exercise 1b: Matrices

Compute the mean of column "C2" in matrix4. (Should be 14.5)

```r
mean_m4c2 <- mean(matrix4[,'C2'])
```

Lists

Creating Lists

```r
# Lists ----
# Creating Lists
list1 <- list(
  name = "Maria",
  mynumbers = vector1,
  mymatrix = matrix1,
  age = 5.3)
```

List Operations

```r
# List Operations

# show the list
list1
```

```
## $name
## [1] "Maria"
##
## $mynumbers
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
##
## $mymatrix
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
##
## $age
## [1] 5.3
```

```r
# Get the age object from the list
list1$age
```

```
## [1] 5.3
```

```r
# Get the names of the objects in the list
names(list1)
```

```
## [1] "name"      "mynumbers" "mymatrix"  "age"
```

```r
# Get the first two objects in the list
list1[1:2]
```

```
## $name
## [1] "Maria"
##
## $mynumbers
## [1]  1.0  2.0  5.3  6.0 -2.0  4.0
```

```r
# Change the name object in the list to Janelle
list1$name = "Janelle"

# changes the value within object 4 of the list (which is age) to 44. Same as above.
list1[[4]] = 44
```

Data Frames

Creating dataframes

```
# Data Frames ----
#Create three vectors of equal length
d <- c(1, 2, 3, 4) # What type of vector?
e <- c("red", "white", "red", NA) # What type of vector?
f <- c(TRUE, TRUE, TRUE, FALSE) # What type of vector?

#Combine the vectors into a dataframe
dataframe1 <- data.frame(d,e,f)

#Set the column names & View
names(dataframe1) <- c("ID","Color","Passed"); # semicolon is a command termination - it shows where on

#Add a new column to the dataframe
dataframe1$month <- c('Jan','Feb', 'Mar', 'Apr')
```

Dataframe operations

```
#Extract a column from the dataframe f
dataframe1$Color
```

```
## [1] "red"   "white" "red"   NA
```

```
#Extract a row
dataframe1[2,]
```

```
##   ID Color Passed month
## 2  2 white   TRUE   Feb
```

```
#Extract several rows
dataframe1[1:2,]
```

```
##   ID Color Passed month
## 1  1   red   TRUE   Jan
## 2  2 white   TRUE   Feb
```

```
#Extract a column
dataframe1[,1]
```

```
## [1] 1 2 3 4
```

```
#Extract a column/columns from their name
out1 <- dataframe1['Color']; class(out1)
```

```
## [1] "data.frame"
```

```r
out2 <- dataframe1$Color; class(out2)
```

```
## [1] "character"
```

```r
out3 <- dataframe1[c("ID","Passed")]; class(out3)
```

```
## [1] "data.frame"
```