Chapter-3-Exercises

Gavin McCorry

2024-02-26

Atomic Vectors

Example is the die vector

```r
die <- c(1, 2, 3, 4, 5, 6)
die
```

```
## [1] 1 2 3 4 5 6
```

```r
is.vector(die)
```

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

Vectors of length 1

```r
five <- 5
five
```

```
## [1] 5
```

```r
is.vector(five)
```

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

```r
length(five)
```

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

```r
length(die)
```

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

Integer and Character Vectors

```r
int <- 1L # Make Integeer Vector by adding L
text <- "ace" # make Character vector by surrounding in quotes

int <- c(1L, 5L)
text <- c("ace", "hearts")

sum(int)
```

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

```r
# sum(text)
# Error in `sum()`:
# ! invalid 'type' (character) of argument
# Execution halted
```

Doubles

```r
#EX:
die <- c(1, 2, 3, 4, 5, 6)
die
```

```
## [1] 1 2 3 4 5 6
```

```r
typeof(die)
```

```
## [1] "double"
```

Integers

```r
int <- c(1L, 2L, 4L)
int
```

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

```r
typeof(int)
```

```
## [1] "integer"
```

```r
sqrt(2)^2 - 2
```

```
## [1] 4.440892e-16
```

Characters

```r
text <- c("Hello", "World")
text
```

```
## [1] "Hello" "World"
```

```r
typeof(text)
```

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

```r
typeof("World")
```

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

Logicals

```r
3 > 4
```

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

```r
logic <- c(TRUE, FALSE, TRUE)
logic
```

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

```r
typeof(logic)
```

```
## [1] "logical"
```

```r
typeof(F)
```

```
## [1] "logical"
```

Complex and Raw

```r
comp <- c(1 +1i, 1 + 2i, + 3i)
comp
```

```
## [1] 1+1i 1+2i 0+3i
```

```r
typeof(comp)
```

```
## [1] "complex"
```

```r
raw(3)
```

```
## [1] 00 00 00
```

```r
typeof(raw(3))
```

```
## [1] "raw"
```

Exercise

Create an atomic vector that stores just the face names of the cards in a royal flush,

for exampl, the ace of spades, king of spads, queen of spades, jack of spades, and ten of spades.

The face of the ace of spades would be "ace", and "spades" is the suit

```r
hand <- c("ace", "king", "queen", "jack", "ten")
hand
```

```
## [1] "ace"   "king"  "queen" "jack"  "ten"
```

```r
typeof(hand)
```

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

Attributes

Names - Most common attribute

```r
names(die)
```

```
## NULL
```

```r
names(die) <- c("one", "two", "three", "four", "five", "six")

names(die)
```

```
## [1] "one"   "two"   "three" "four"  "five"  "six"
```

```r
attributes(die)
```

```
## $names
## [1] "one"    "two"    "three" "four"  "five"  "six"
```

```r
die
```

```
##   one   two three  four  five   six
##     1     2     3     4     5     6
```

```r
die + 1 # names wont effect the actuak value of the vector
```

```
##   one   two three  four  five   six
##     2     3     4     5     6     7
```

```r
# you can rename the elements vector
names(die) <- c("uno", "dos", "tres", "cuatro", "cinco", "seis")
die
```

```
##    uno    dos   tres cuatro  cinco   seis
##      1      2      3      4      5      6
```

```r
# can remove all names as well
names(die) <- NULL
die
```

```
## [1] 1 2 3 4 5 6
```

Dim

```r
# can transform an atomic vector into an n-dimensional array

# 2x3 matrix
dim(die) <- c(2, 3)
die
```

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

```r
# 3x2 matrix
dim(die) <- c(3, 2)
die
```

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

```r
# 1 x 2 x 3 hyperecube
dim(die) <- c(1, 2, 3)
die
```

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

Matrices

```r
m <- matrix(die, nrow = 2)
m
```

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

```r
m <- matrix(die, nrow = 2, byrow = TRUE)
m
```

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

Arrays

```r
ar <- array(c(11:14, 21:24, 31:34), dim = c(2, 2, 3))
ar
```

```
## , , 1
##
##      [,1] [,2]
## [1,]   11   13
## [2,]   12   14
##
## , , 2
##
```

```
##      [,1] [,2]
## [1,]   21   23
## [2,]   22   24
##
## , , 3
##
##      [,1] [,2]
## [1,]   31   33
## [2,]   32   34
```

Exercise 2

```
hand1 <- c("ace", "king", "queen", "jack", "ten", "spades", "spades", "spades", "spades", "spades")

matrix(hand1, nrow = 5)
```

```
##      [,1]    [,2]
## [1,] "ace"   "spades"
## [2,] "king"  "spades"
## [3,] "queen" "spades"
## [4,] "jack"  "spades"
## [5,] "ten"   "spades"
```

Class

```
dim(die) <c(2, 3)
```

```
## Warning in dim(die) < c(2, 3): longer object length is not a multiple of
## shorter object length
```

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

```
typeof(die)
```

```
## [1] "double"
```

```
class(die)
```

```
## [1] "array"
```

```
attributes(die)
```

```
## $dim
## [1] 1 2 3
```

```r
class("Hello")
```

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

```r
class(5)
```

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

Dates and Times

```r
now <- Sys.time()
now
```

```
## [1] "2024-02-26 18:30:14 CST"
```

```r
typeof(now)
```

```
## [1] "double"
```

```r
class(now)
```

```
## [1] "POSIXct" "POSIXt"
```

```r
unclass(now)
```

```
## [1] 1708993814
```

```r
# What day is it a million seconds after 12:00 am han 1, 1970?
mil <- 1000000
mil
```

```
## [1] 1e+06
```

```r
class(mil) <- c("POSIXct", "POSIXt")
mil
```

```
## [1] "1970-01-12 07:46:40 CST"
```

Factors

```r
gender <- factor(c("male", "female", "female", "male"))
typeof(gender)
```

```
## [1] "integer"
```

```r
attributes(gender)
```

```
## $levels
## [1] "female" "male"
##
## $class
## [1] "factor"
```

```r
unclass(gender)
```

```
## [1] 2 1 1 2
## attr(,"levels")
## [1] "female" "male"
```

```r
gender
```

```
## [1] male    female female male
## Levels: female male
```

```r
as.character(gender)
```

```
## [1] "male"   "female" "female" "male"
```

Exercise 3

```r
card <- c("ace", "hearts", 1)
card
```

```
## [1] "ace"    "hearts" "1"
```

```r
# you cant put multiple types of data in an atomic vector
```

Coercion

```r
# R uses the same coercion ruls when you try to do math with logic values.So:
sum(c(TRUE, TRUE, FALSE, FALSE))
```

```
## [1] 2
```

```r
# is the same as:
sum(c(1, 1, 0, 0))
```

```
## [1] 2
```

```r
as.character(1)
```

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

```r
as.logical(1)
```

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

```r
as.numeric((FALSE))
```

```
## [1] 0
```

Lists

```r
list1 <- list(100:300, "R", list(TRUE, FALSE))
list1
```

```
## [[1]]
##   [1] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
##  [19] 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135
##  [37] 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153
##  [55] 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171
##  [73] 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189
##  [91] 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207
## [109] 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
## [127] 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243
## [145] 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261
## [163] 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279
## [181] 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297
## [199] 298 299 300
##
## [[2]]
## [1] "R"
##
## [[3]]
## [[3]][[1]]
## [1] TRUE
##
## [[3]][[2]]
## [1] FALSE
```

Exercise 4

```r
# make one card using a list
card <- list("ace", "hearts", 1)
card
```

```
## [[1]]
## [1] "ace"
##
## [[2]]
## [1] "hearts"
##
## [[3]]
## [1] 1
```

Data Frames

```
df <- data.frame(fac = c("ace", "two", "six"), suit = c("clubs", "clubs", "clubs"), value = c(1, 2, 6))
df
```

```
##   fac  suit value
## 1 ace clubs     1
## 2 two clubs     2
## 3 six clubs     6
```

```
typeof(df)
```

```
## [1] "list"
```

```
class(df)
```

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

```
str(df)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ fac  : chr  "ace" "two" "six"
##  $ suit : chr  "clubs" "clubs" "clubs"
##  $ value: num  1 2 6
```

```
df <- data.frame(fac = c("ace", "two", "six"), suit = c("clubs", "clubs", "clubs"), value = c(1, 2, 6),
df
```

```
##   fac  suit value
## 1 ace clubs     1
## 2 two clubs     2
## 3 six clubs     6
```

Loading Data

load deck.csv from http://bit.ly/deck_CSV

```
# head(deck)
```

Saving Data

```
# write.csv(deck, file = "cards.csv", row.names = FALSE)
```