

Advanced R

Chapter 3: Vectors

Vajresh Balaji

@bvajresh

2020-08-13

Outline

- 3.2 Atomic Vectors
- 3.3 Attributes
- 3.4 S3 Atomic Vectors
- 3.5 Lists
- 3.6 Dataframes and Tibbles
- 3.7 NULL

Vectors

- 2 types of vectors
 - Atomic
 - List

Atomic Vectors

- 4 common types

```
a <- c(1,2,3,4) #Integer  
b <- c(TRUE, FALSE, T, F) #Logical  
c <- c(1.2, 2.3, 5.0) #Double  
d <- c("apple", "banana") #Character
```

- Rare types of Atomic Vectors: Raw and Complex

Missing Values

- R uses `NA` to represent missing values.

```
x <- c(NA, 5, NA, 10)
x == NA
```

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

- Use `is.na()` to check for missing values

```
is.na(x)
```

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

Testing

- Type of vectors can be tested with `is.*()` function.
- `is.logical()`, `is.integer()`, `is.double()`, and `is.character()`
- Avoid using `is.vector()`, `is.atomic()`, and `is.numeric()`

Coercion

- For atomic vectors, type is a property of the entire vector.
- When attempting to combine different types of elements, they will be coerced in a fixed order.
- character -> double -> integer -> logical

```
str(c("a", 1))
```

```
## chr [1:2] "a" "1"
```

- Coercion happens automatically.

```
x <- c(FALSE, FALSE, TRUE)  
as.numeric(x)
```

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

```
sum(x) #Total number of TRUEs
```

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

- Using `as.*()` allows us to deliberately coerce.
 - `as.logical()`, `as.integer()`, `as.double()` and `as.character()`

Attributes

- Attributes can be added to atomic vectors to build data structures like Arrays, Matrices, Factors or date-times.
- They can be individually set and retrieved using `attr()`

```
car = "CR-V"  
attr(car, 'manufacturer') <- 'Honda'  
attr(car, 'manufacturer')
```

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

- Set multiple attributes using

```
car2 <- structure("Model S", manufacturer = "Tesla", year = 2020)
```

- You can retrieve multiple attributes by using

```
attributes(car2)
```

```
## $manufacturer  
## [1] "Tesla"  
##  
## $year  
## [1] 2020
```


Names

```
x <- c(apple = 'a', banana = 'b') # 1  
x
```

```
##  apple banana  
##    "a"      "b"
```

```
y <- c('a', 'b')  
names(y) <- c('apple', 'banana') # 2  
y
```

```
##  apple banana  
##    "a"      "b"
```

```
setNames(y, c('apple', 'banana')) # 3
```

```
##  apple banana  
##    "a"      "b"
```

Source: TonyElHabr Chapter 3 slide 8

Dimensions

- Allows vector to behave like a matrix or an array.

```
a <- matrix(1:6, nrow = 2, ncol = 3)
a
```

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

```
b <- matrix(1:6, c(1, 3, 2))
b
```

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

Unusual Behaviour

- Vectors without `dim` are thought of as 1-dimensionals but are **NULL**
- Matrices with 1 row or col and 1-dimensional arrays print the same but behave differently
- Use `str()` to reveal the differences.

S3 Atomic Vectors

- Objects that have a `class` attribute.
- 4 types of S3 vectors
 - factor (categorical)
 - Date (date)
 - POSIXct (date-time)
 - duration (difftime)

Factors

- Vectors that only contain pre-defined values
- Used for Categorical Data

```
sex_char <- c("m", "m", "m")  
sex_factor <- factor(sex_char, levels = c("m", "f"))  
table(sex_factor)
```

```
## sex_factor  
## m f  
## 3 0
```

```
grade <- ordered(c("b", "b", "a", "c"), levels = c("c", "b", "a"))  
grade
```

```
## [1] b b a c  
## Levels: c < b < a
```

- Many base R functions automatically convert character vectors into factors.
- Use `stringsAsFactors = FALSE` to suppress this behaviour.

Dates, POSIXct & Duration

- All built on top of double vectors.
- Dates

```
today <- Sys.Date()  
typeof(today)
```

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

```
attributes(today)
```

```
## $class  
## [1] "Date"
```

- Date-times
 - Two types of storing date-time: POSIXct & POSIXlt
 - Underlying value represents number of seconds since Jan 1, 1970.
 - **tz** attribute

- Duration
 - Amount of time between date/date-time pairs.
 - Stored in `difftime`s
 - `units` attribute to determine how integer should be interpreted.

```
one_week_1 <- as.difftime(1, units = "weeks")  
one_week_1
```

```
## Time difference of 1 weeks
```

```
typeof(one_week_1)
```

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

```
attributes(one_week_1)
```

```
## $class  
## [1] "difftime"  
##  
## $units  
## [1] "weeks"
```

Lists

- Can be of any atomic type or contain other lists.

```
e <- list(1, TRUE, 1.2, "apple", list(2, 4, 6))
```

- Elements of a list are references.
- `c()` combines several lists into one if there given a combination of atomic vectors and lists.

```
l4 <- list(list(1, 2), c(3, 4))  
str(l4)
```

```
## List of 2  
## $ :List of 2  
## ..$ : num 1  
## ..$ : num 2  
## $ : num [1:2] 3 4
```


Data Frames

- S3 Vectors that are built on top of lists.

```
df1 <- data.frame(x = 1:3, y = letters[1:3])  
typeof(df1)
```

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

```
attributes(df1)
```

```
## $names  
## [1] "x" "y"  
##  
## $class  
## [1] "data.frame"  
##  
## $row.names  
## [1] 1 2 3
```

- Constraint
 - Length of each vector must be the same

Tibble

- Share the same structure as data frames
- class vectors are longer
- Default behaviour: `stringsAsFactors = FALSE`
- Discourages rownames
- "Nicer" Printing

NULL

- NULL has a unique type

```
typeof(NULL)
```

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

```
length(NULL)
```

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

- Common uses
 - Represent an empty vector

```
c()
```

```
## NULL
```

- Represent an absent vector