Common Data Objects in R

Vectors, Lists, Data Frames and Tibbles

by Martin Frigaard

Written: October 03 2022

Updated: December 02 2022

Created using the "λέξις" theme

Materials

The slides are in the `slides.pdf` file

The materials for this training are in the `worksheets` folder:

```
worksheets
├── import.Rmd
├── export.Rmd
├── objects.Rmd
├── rmd-basic.Rmd
├── rmd-tables.Rmd
└── rmd-visualizations.Rmd
```

Outline

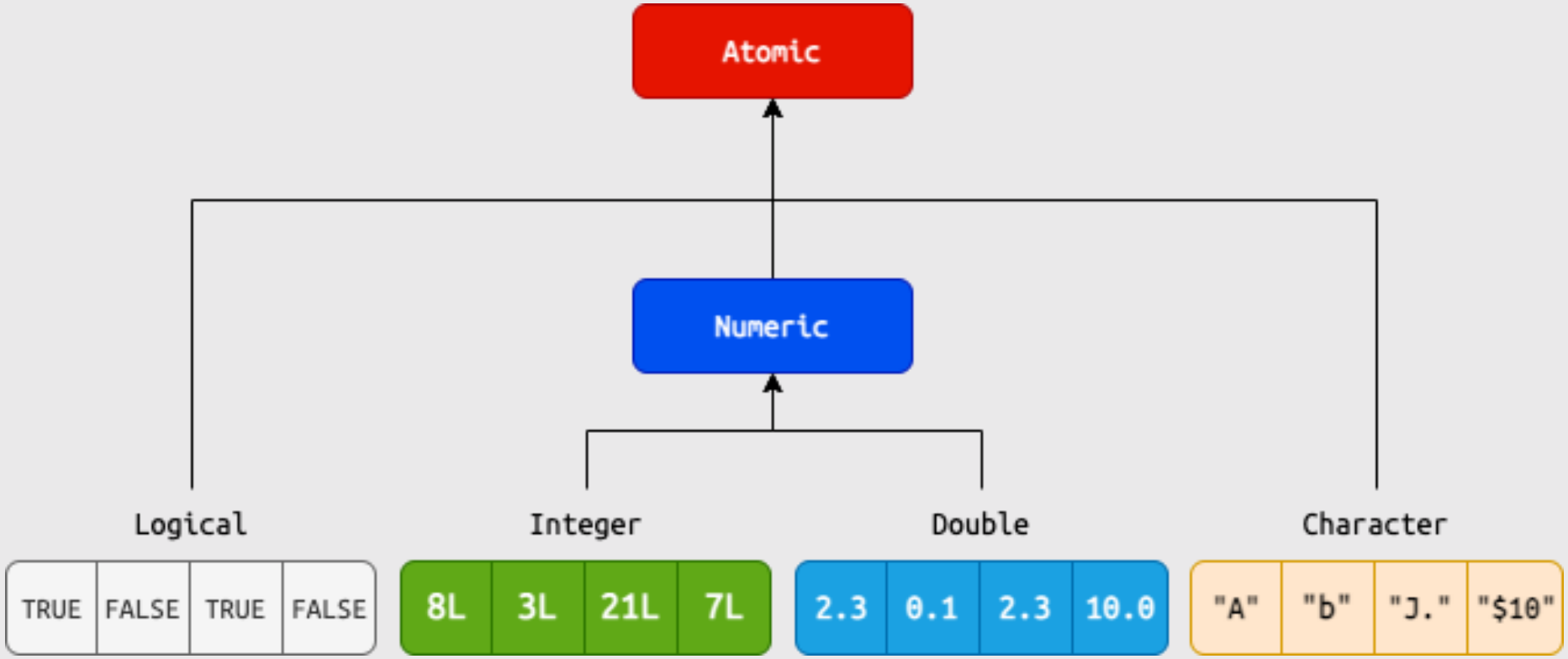
1. Importing data
2. Common Data Objects
3. R Markdown
4. R Markdown Data Visualizations
5. R Markdown Tables
6. Exporting Data

Common Data Objects

Open `objects.Rmd` to follow along

Data Objects: vector

Vectors are the fundamental data object in R



Data Objects: creating vectors

`c()` is used to combine (or concatenate) a variety of elements

`<-` is referred to as the assignment operator, and it's used with `c()` to assign elements to a designated object

Earlier we used `<-` to create the `medical` dataset

Create logical and integer vectors (`log_vec` and `int_vec`)

```
log_vec <- c(TRUE, FALSE)
int_vec <- c(4L, 7L)
```

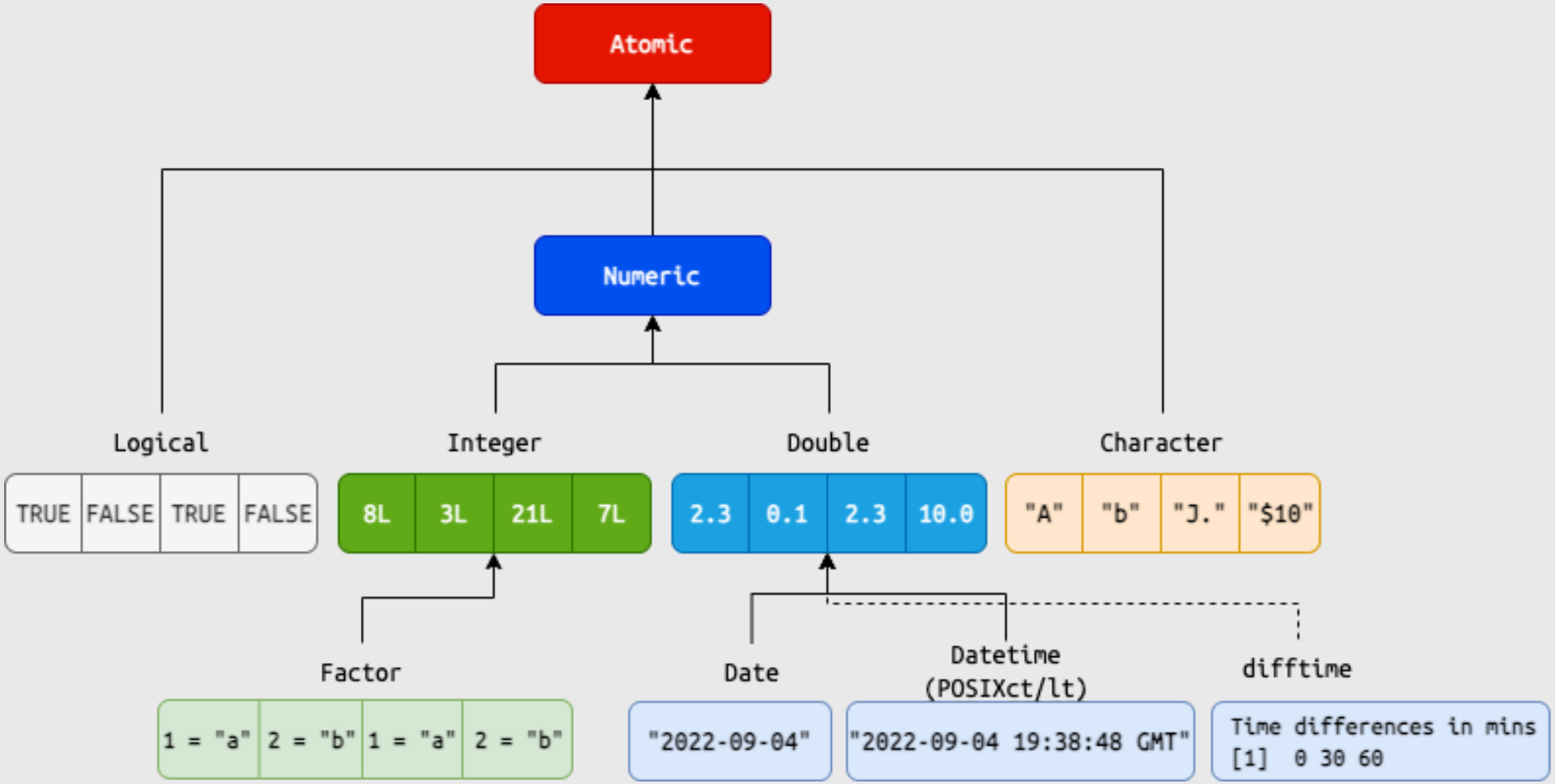
Create double and character vectors (`dbl_vec` and `chr_vec`)

```
dbl_vec <- c(2.2, 8.09)
chr_vec <- c("A", "D")
```

Data Objects: atomic vectors

Print Atomic Vectors	Check with `typeof()`	Check `class()`
`log_vec`	`typeof(log_vec)`	`class(log_vec)`
`[1]  TRUE FALSE`	`[1] "logical"`	`[1] "logical"`
`int_vec`	`typeof(int_vec)`	`class(int_vec)`
`[1] 4 7`	`[1] "integer"`	`[1] "integer"`
`dbl_vec`	`typeof(dbl_vec)`	`class(dbl_vec)`
`[1] 2.20 8.09`	`[1] "double"`	`[1] "numeric"`
`chr_vec`	`typeof(chr_vec)`	`class(chr_vec)`
`[1] "A" "D"`	`[1] "character"`	`[1] "character"`

Data Objects: S3 vectors



Data Objects: S3 vectors

Create S3 Vectors

```
fct_vec <- factor(
        x = c("Medium", "Low", "High"),
        levels = c("Low", "Medium", "High"))
date_vec <- c(Sys.Date(), Sys.Date() + 1)
dt_vec <- c(Sys.time(), Sys.time() + (86400*365))
difft_vec <- difftime(
            time1 = Sys.time(),
            time2 = Sys.time() + (86400*365),
            units = "days")
```

View S3 vectors

```
fct_vec
```

```
  [1] Medium Low    High
 Levels: Low Medium High
```

```
date_vec
```

```
  [1] "2022-12-02" "2022-12-03"
```

```
dt_vec
```

```
  [1] "2022-12-02 09:33:48 PST" "2023-12-02 09:33:48 PST"
```

```
difft_vec
```

```
  Time difference of -365 days
```

Data Objects: S3 vectors

Check `typeof()`

```
typeof(fct_vec)
```

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

```
typeof(date_vec)
```

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

```
typeof(dt_vec)
```

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

```
typeof(difft_vec)
```

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

Check `class()`

```
class(fct_vec)
```

```
  [1] "factor"
```

```
class(date_vec)
```

```
  [1] "Date"
```

```
class(dt_vec)
```

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

```
class(difft_vec)
```

```
  [1] "difftime"
```

Data Objects: S3 vectors

S3 vectors have additional `attributes()`

Factor attributes

```
attributes(fct_vec)
```

```
$levels
[1] "Low"    "Medium" "High"

$class
[1] "factor"
```

Date/Datetime attributes

```
attributes(date_vec)
```

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

```
attributes(dt_vec)
```

```
$class
[1] "POSIXct" "POSIXt"

$tzone
[1] ""
```

Difftime attributes

```
attributes(difft_vec)
```

```
$class
[1] "difftime"

$units
[1] "days"
```

Data Objects: lists

Vectors have to be the same type, or `class`

Lists can contain objects of different `class`es

```
atomic_list <- list(
    'logical vector' = log_vec,
    'integer vector' = int_vec,
    'double vector' = dbl_vec,
    'character vector' = chr_vec
)
```

```
atomic_list
```

```
$`logical vector`
[1]  TRUE FALSE

$`integer vector`
[1] 4 7

$`double vector`
[1] 2.20 8.09

$`character vector`
[1] "A" "D"
```

Data Objects: lists

Lists can even contain other lists!

Create list of date vectors

```
s3_list <- list(
    'date vector' = date_vec,
    'datetime vector' = dt_vec,
    'difftime vector' = difft_vec
)
```

Create list of lists

```
vector_list <- list(
    'S3 list' = s3_list,
    'Atomic list' = atomic_list
)
```

```
vector_list
```

```
$`S3 list`
$`S3 list`$`date vector`
[1] "2022-12-02" "2022-12-03"

$`S3 list`$`datetime vector`
[1] "2022-12-02 09:33:48 PST" "2023-12-02 09:33:48 PST"

$`S3 list`$`difftime vector`
Time difference of -365 days


$`Atomic list`
$`Atomic list`$`logical vector`
[1]  TRUE FALSE

$`Atomic list`$`integer vector`
[1] 4 7

$`Atomic list`$`double vector`
[1] 2.20 8.09

$`Atomic list`$`character vector`
[1] "A" "D"
```

Data Objects: data.frames

A `data.frame` is a rectangular list

Create `data.frame`

```
my_df <- data.frame(
    log_col = log_vec,
    int_col = int_vec,
    dbl_col = dbl_vec,
    chr_col = chr_vec,
    date_col = date_vec,
    dt_col = dt_vec
)
```

View `data.frame`

```
my_df
```

```
  log_col int_col dbl_col chr_col   date_col              dt_col
1    TRUE       4    2.20       A 2022-12-02 2022-12-02 09:33:48
2   FALSE       7    8.09       D 2022-12-03 2023-12-02 09:33:48
```

Data Objects: data.frames

Check the structure of the `data.frame`

```
str(my_df)
```

```
'data.frame':    2 obs. of  6 variables:
 $ log_col : logi  TRUE FALSE
 $ int_col : int  4 7
 $ dbl_col : num  2.2 8.09
 $ chr_col : chr  "A" "D"
 $ date_col: Date, format: "2022-12-02" "2022-12-03"
 $ dt_col  : POSIXct, format: "2022-12-02 09:33:48" "2023-12-02 09:33:48"
```

Check the `class` and `typeof()` for the a `data.frame`

```
class(my_df)
```

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

```
typeof(my_df)
```

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

Data Objects: tibbles

A tibble is a modern reimagining of the `data.frame`

They are created just like `data.frame`s

Create `tibble`

```
my_tbl <- tibble(
    log_col = log_vec,
    int_col = int_vec,
    dbl_col = dbl_vec,
    chr_col = chr_vec,
    date_col = date_vec,
    dt_col = dt_vec
)
```

View `tibble`

```
my_tbl
```

```
# A tibble: 2 × 6
  log_col int_col dbl_col chr_col date_col   dt_col
  <lgl>     <int>   <dbl> <chr>   <date>     <dttm>
1 TRUE          4    2.2  A       2022-12-02 2022-12-02 09:33:48
2 FALSE         7    8.09 D       2022-12-03 2023-12-02 09:33:48
```

Data Objects: data.frames & tibbles

`tibble`s print a little nicer than `data.frame`s, and we'll primarily be using them because they work well with other functions for tables and visualizations.

```
my_df
```

```
  log_col int_col dbl_col chr_col   date_col             dt_col
1    TRUE       4    2.20       A 2022-12-02 2022-12-02 09:33:48
2   FALSE       7    8.09       D 2022-12-03 2023-12-02 09:33:48
```

```
my_tbl
```

```
# A tibble: 2 × 6
  log_col int_col dbl_col chr_col date_col   dt_col
  <lgl>     <int>   <dbl> <chr>   <date>     <dttm>
1 TRUE          4    2.2  A       2022-12-02 2022-12-02 09:33:48
2 FALSE         7    8.09 D       2022-12-03 2023-12-02 09:33:48
```