

Getting Data with R

Tony Yao-Jen Kuo

How to get data with R

Overview

- ▶ From files

Overview

- ▶ From files
- ▶ From web

Getting data from files

Using read.csv() for CSV files

- ▶ CSV stands for **comma separated values**

```
file_url <- "https://storage.googleapis.com/ds_data_import/
df <- read.csv(file_url, stringsAsFactors = FALSE) # person
dim(df)
```

```
## [1] 15 7
```

Using read.table() for general tabular text files

```
file_url <- "https://storage.googleapis.com/ds_data_import,  
df <- read.table(file_url, header = TRUE, sep = ";")
```

```
## Warning in scan(file = file, what = what, sep = sep, qu  
## dec, : EOF within quoted string
```

```
dim(df)
```

```
## [1] 15 7
```

Using readxl package for Excel spreadsheets

```
install.packages("readxl")
```

```
library(readxl)
```

```
file_path <- "/Users/kuoyaojen/Downloads/fav_nba_teams.xlsx"  
chicago_bulls <- read_excel(file_path)  
head(chicago_bulls)
```

```
## # A tibble: 6 x 7
```

```
##   No. Player      Pos   Ht      Wt `Birth Date`  
##   <dbl> <chr>      <chr> <chr> <dbl> <chr>  
## 1     0 Randy Brown  PG    6-2     190 May 22, 1968  
## 2    30 Jud Buechler  SF    6-6     220 June 19, 1968  
## 3    35 Jason Caffey  PF    6-8     255 June 12, 1973  
## 4    53 James Edwards C      7-0     225 November 22, 1959  
## 5    54 Jack Haley   C     6-10    240 January 27, 1964  
## 6     9 Ron Harper   PG    6-6     185 January 20, 1964
```


Importing other sheets

```
boston_celtics <- read_excel(file_path, sheet = "boston_cel")  
head(boston_celtics)
```

Reading specific cell ranges

```
partial_chi <- read_excel(file_path, range = "B8:C13", col_...  
knitr::kable(partial_chi)
```

Using jsonlite package for JSON files

```
install.packages("jsonlite")
```

```
library(jsonlite)
```

```
file_url <- "https://storage.googleapis.com/ds_data_import/  
chicago_bulls <- fromJSON(file_url)  
class(chicago_bulls)
```

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

A quick review

Source	Format
CSV	<code>data.frame</code>
TXT	<code>data.frame</code>
Spreadsheet	<code>data.frame</code>
JSON	<code>list</code>

Getting data from web

jsonlite for RESTful APIs

```
library(jsonlite)
```

```
web_url <- "https://ecshweb.pchome.com.tw/search/v3.3/all/1"
```

```
macbook <- fromJSON(web_url)
```

```
class(macbook)
```

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

```
names(macbook)
```

```
## [1] "QTime"      "totalRows" "totalPage" "range"      "cat"
```

```
## [7] "subq"      "token"     "prods"
```

rvest for HTML documents

```
install.packages("rvest")
```

The use of %>% operator

- ▶ Originated from `magrittr` package
- ▶ Now an important operator for the tidyverse eco-system
- ▶ Can be generated with: `Ctrl + Shift + m`

How to call a function

```
library(rvest)
```

```
## Loading required package: xml2
```

```
# traditional  
sum(1:10)
```

```
## [1] 55
```

```
# using %>%  
1:10 %>%  
  sum()
```

```
## [1] 55
```

More examples

```
# traditional  
toupper(paste0(strsplit("Jeremy Lin", split = " ")[[1]][2],
```

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

```
# using %>%  
"Jeremy Lin" %>%  
  strsplit(split = " ") %>%  
  `[` (1) %>%  
  `[` (2) %>%  
  paste0("sanity") %>%  
  toupper()
```

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

`read_html()` for reading all html contents

```
library(rvest)

mi_url <- "https://www.imdb.com/title/tt4912910/"
html_doc <- mi_url %>%
  read_html()
```

html_nodes() to locate elements

```
html_doc %>%  
  html_nodes("strong span") # CSS selector
```

```
## {xml_node} (1)  
## [1] <span>8.1</span>
```

html_text() to remove tags

```
html_doc %>%  
  html_nodes("strong span") %>%  
  html_text()
```

```
## [1] "8.1"
```

Data of html document are characters

```
html_doc %>%  
  html_nodes("strong span") %>%  
  html_text() %>%  
  as.numeric()
```

```
## [1] 8.1
```

How to locate elements?

- ▶ By CSS Selectors

How to locate elements?

- ▶ By CSS Selectors
- ▶ By XPath

The use of Chrome plugins

SelectorGadget

A Chrome plugin for CSS selectors: SelectorGadget

How to use SelectorGadget?

XPath Helper

A Chrome plugin for XPath: XPath Helper

How to use XPath Helper?

Practices: Getting genre information from IMDB.com

```
mi_url <- "https://www.imdb.com/title/tt4912910/"
```

```
## [1] "Action"      "Adventure" "Thriller"
```

Practices: Getting cast information from IMDB.com

```
mi_url <- "https://www.imdb.com/title/tt4912910/"
```

```
## [1] " Tom Cruise\n"           " Henry Cavill\n"           " Vin  
## [4] " Simon Pegg\n"           " Rebecca Ferguson\n"       " Sea  
## [7] " Angela Bassett\n"       " Vanessa Kirby\n"           " Mic  
## [10] " Wes Bentley\n"          " Frederick Schmidt\n"      " Ale  
## [13] " Liang Yang\n"          " Kristoffer Joner\n"      " Wo
```

Practices: Getting price ranking from Yahoo! Stock

- ▶ Top 100 for TSE: <https://tw.stock.yahoo.com/d/i/rank.php?t=pri&e=tse&n=100>
- ▶ Top 100 for OTC: <https://tw.stock.yahoo.com/d/i/rank.php?t=pri&e=otc&n=100>