



## R for Data Science

### Bài 10: XML & JSON

Phòng LT & Mạng

[https://csc.edu.vn/lap-trinh-va-cSDL/R-Programming-Language-for-Data-Science\\_190](https://csc.edu.vn/lap-trinh-va-cSDL/R-Programming-Language-for-Data-Science_190)

2020



## Nội dung

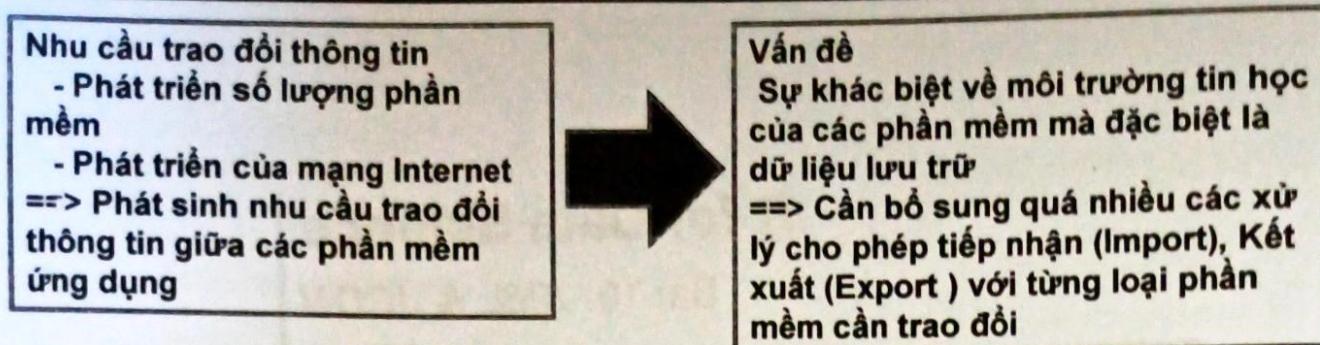


### 1. XML

### 2. JSON



## Giới thiệu



## Giới thiệu

### XML (eXtensible Markup Language) – Ngôn ngữ đánh dấu mở rộng

- Được thiết kế để chuyển và lưu trữ dữ liệu
- Do W3C đề nghị sử dụng
- Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet.



### □ Đặc điểm

- Dễ dàng viết được những chương trình xử lý tài liệu XML
- Dễ đọc và có tính hợp lý cao
- Giảm thiểu những thuộc tính tùy chọn
- Dễ dàng được sử dụng trên Internet
- Hỗ trợ nhiều ứng dụng



### □ Cấu trúc

- Tài liệu XML có cấu trúc cây (tree) bắt đầu bằng một nút gốc (root) và các nhánh lá (leave)
- Tài liệu XML là tài liệu tự mô tả và có cấu trúc đơn giản



## Giới thiệu

### ❑ Cấu trúc

```

    Khai báo phiên bản và loại mã hóa
    <?xml version="1.0" encoding="UTF-8"?>
<note>                                Nút gốc (root element)
    <to>Students</to>
    <from>Teacher</from>                  Nút con (child element)
    <heading>Reminder</heading>
    <body>Please come to class this Monday evening
        on time</body>
</note>                                Kết thúc nút gốc

```



## Giới thiệu

### ❑ Cú pháp

- Có một root Element duy nhất
- Mỗi tag được mở ra cần phải đóng lại
- Tag có phân biệt chữ hoa và chữ thường
- Nút con (Child Element) phải nằm trong nút cha (Parent Element)
- Ghi chú trong XML: <!-- Nội dung ghi chú-->

### ❑ Chú ý: Trong file .xml cần phải có dòng khai báo XML ở đầu file



## Giới thiệu

### ❑ Ví dụ

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Don_hang>
    <Ngay_dat_hang>2017-02-27</Ngay_dat_hang>
    <Ten_khach_hang>Khuát Thuỳ Phương</Ten_khach_hang>
    <Noi_dung>
        <Ma_san_pham>1</Ma_san_pham>
        <So_luong>2</So_luong>
    </Noi_dung>
    <Noi_dung>
        <Ma_san_pham>4</Ma_san_pham>
        <So_luong>1</So_luong>
    </Noi_dung>
</Don_hang>
```

## Giới thiệu

### ❑ Ví dụ

```
<?xml version="1.0" encoding="UTF-8"?>
<DON_VI_S shelf="Danh sách đơn vị">
    <DON_VI ID="1" Ten="Đơn vị A" />
    <DON_VI ID="2" Ten="Đơn vị B" />
    <DON_VI ID="3" Ten="Đơn vị C" />
    <DON_VI ID="4" Ten="Đơn vị D" />
    <DON_VI ID="5" Ten="Đơn vị E" />
</DON_VI_S>
```

## Làm việc với tập tin XML

### □ Để làm việc với tập tin XML cần phải `install.packages("XML")`

```
install.packages("XML")
```

```
Installing package into 'C:/Users/ktphuong/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
```

```
package 'XML' successfully unpacked and MD5 sums checked
```

```
The downloaded binary packages are in
C:\Users\ktphuong\AppData\Local\Temp\Rtmpc520Ad\downloaded_packages
```

### □ Tải package XML: `library("XML")`

### □ Tải các pakage cần thiết khác: `library("methods")`



R programming language for Data Science

11

## Làm việc với tập tin XML



### □ Xuất nội dung xml qua data frame

- Sử dụng: `xmlToDataFrame("Đường dẫn tập tin")`

- Ví dụ:

```
data <- xmlToDataFrame("Du_lieu/cd.xml")
```

data

TITLE	ARTIST	COUNTRY	COMPANY	PRICE	YEAR
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
Eros	Eros Ramazzotti	EU	BMG	9.90	1997
One night only	Bee Gees	UK	Polydor	10.90	1998
Sylvias Mother	Dr.Honk	UK	CBS	8.10	1973



R programming language for Data Science \*\*\*

12

## □ Ghi nội dung xml vào xml file

- Sử dụng:

```
cat(saveXML(xml_data, indent = TRUE,  
prefix = "<?xml version=\"1.0\""  
encoding = "utf-8" standalone = "no"?>\n"),  
file = "../Đường_dẫn_tập_tin")
```

Chú ý: Cần phải chuyển dữ liệu sang xml data trước khi lưu vào tập tin



## □ Ghi nội dung xml vào xml file

- Ví dụ:

```
# create xml file from data frame  
doc = newXMLDoc()  
# Simple creation of an XML tree using these functions  
top = newXMLNode("CATALOG", doc = doc)  
for (row in 1:nrow(CDS_USA)){  
  cdnode = newXMLNode("CD", parent=top)  
  newXMLNode("TITLE", CDS_USA[row, "TITLE"], parent=cdnode)  
  newXMLNode("ARTIST", CDS_USA[row, "ARTIST"], parent=cdnode)  
  newXMLNode("COUNTRY", CDS_USA[row, "COUNTRY"], parent=cdnode)  
  newXMLNode("COMPANY", CDS_USA[row, "COMPANY"], parent=cdnode)  
  newXMLNode("PRICE", CDS_USA[row, "PRICE"], parent=cdnode)  
  newXMLNode("YEAR", CDS_USA[row, "YEAR"], parent=cdnode)  
}  
  
cat(saveXML(doc,  
            indent = TRUE,  
            prefix = "<?xml version=\"1.0\" encoding=\"utf-8\" standalone = \"no\"?>\n"),  
    file = "../R/chuong12/Du_lieu/cd_usa.xml")
```



## Làm việc với tập tin XML

### □ Đọc nội dung xml từ URL

- Download nội dung, sau đó ghi vào tập tin xml với `download.file(url, destfile = "Đường dẫn tập tin")`

- Ví dụ: `url <- 'http://www.w3schools.com/xml/cd_catalog.xml'`  
`download.file(url, destfile = 'cd.xml')`

```
cd <- xmlToDataFrame("cd.xml")
```

```
print(head(cd))
```

	TITLE	ARTIST	COUNTRY	COMPANY	PRICE	YEAR
1	Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
2	Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
3	Greatest Hits	Dolly Parton	USA	RCA	9.90	1982
4	Still got the blues	Gary Moore	UK	Virgin records	10.20	1990
5	Eros	Eros Ramazzotti	EU	BMG	9.90	1997
6	One night only	Bee Gees	UK	Polydor	10.90	1998

## Nội dung

1. XML

2. JSON

## Giới thiệu

# {JSON}

JavaScript Object Notation

- JSON là dạng dữ liệu lightweight text dựa trên open standard được thiết kế để trao đổi dữ liệu.
- JSON format được đưa ra bởi Douglas Crockford, và được mô tả trong tiêu chuẩn RFC 4627.
- JSON được mở rộng từ JavaScript scripting language.
- Tập tin JSON có phần mở rộng là .json
- JSON được sử dụng trong hầu hết các NNLT như PHP, PERL, Python, Ruby, Java, R...
- JSON Internet Media type là application/json.



## Giới thiệu

### □ Đặc điểm

- Rất dễ đọc và viết.
- Là một định dạng trao đổi văn bản siêu nhẹ
- JSON là ngôn ngữ độc lập
- Dễ dàng được sử dụng trên Internet
- Hỗ trợ nhiều ứng dụng



## Giới thiệu

### □ JSON syntax

- Xuất phát từ cú pháp ký hiệu của JavaScript:
  - Dữ liệu theo định dạng cặp name/value như sau: "name" : value
  - Dữ liệu được phân cách nhau bằng dấu phẩy ,
  - Ngoặc nhọn (curly brace) {} chứa các đối tượng
  - Ngoặc vuông (square bracket) [] chứa các danh sách



## Giới thiệu

### □ JSON Syntax

- Ví dụ

```
{
  "books": [
    { "language": "Java" , "edition": "first" },
    { "language": "Python" , "edition": "second" },
    { "language": "Kotlin" , "edition": "third" }
  ]
}
```

Chú ý: trong JSON, name (key) phải là string, được viết trong dấu nháy đôi double quote ("")



## Giới thiệu

### □ Các kiểu dữ liệu hỗ trợ

- string
- number
- object (JSON object)
- array
- boolean
- null

Không hỗ trợ các kiểu: function, date,  
*undefined*



## Nội dung

1. Giới thiệu JSON
2. Làm việc với tập tin JSON



## Làm việc với tập tin JSON

### □ Để làm việc với tập tin JSON cần phải cài và sử dụng cách thư viện

- library("httr")
- library("jsonlite")



## Làm việc với tập tin JSON

### □ Đọc nội dung tập tin json

- Sử dụng: **fromJSON(txt = “Đường dẫn tập tin”)**
- Ví dụ:  
`json_text <- fromJSON("Du_lieu/books.json")`

title	publisher	isbn	pages	author	attribute
Applied Linear Statistical Models	McGraw Hill	9780073108742	1396	Michael Kutner, William Li, Christopher Nachtsheim, John Neter	Exercises, Illustrations, Readability
Mathematical Proofs: A Transition to Advanced Mathematics	Pearson	9780321390530	365	Gary Chartrand, Ping Zhang, Albert Polimeni	Exercises, Readability
Mathematical Statistics with Resampling and R	Wiley	9781118029855	418	Laura Chihara, Tim Hesterberg	Exercises, Illustrations, Readability



## ☐ Xuất nội dung json qua data frame

- Sử dụng: `as.data.frame("nội dung tập tin json")`
- Ví dụ: `books <- as.data.frame(json_text)`

books

Mathematics.book.title	Mathematics.book.publisher	Mathematics.book.isbn	Mathematics.book.pages	Mathematics.book.author	Mathematics.book.attribute
Applied Linear Statistical Models	McGraw Hill	9780073108742	1396	Michael Kutner , William Li , Christopher Nachtsheim , John Neter	Exercises , Illustrations , Readability
Mathematical Proofs: A Transition to Advanced Mathematics	Pearson	9780321390530	365	Gary Chartrand , Ping Zhang , Albert Polimeni	Exercises , Readability
Mathematical Statistics with Resampling and R	Wiley	9781118029855	418	Laura Chihara , Tim Hesterberg	Exercises , Illustrations , Readability



## ☐ Ghi nội dung vào JSON file

- Sử dụng: `write(json_data, file = "Đường dẫn tập tin")`

- Ví dụ:

```
# convert dataframe to JSON data
# save file
json_books <- toJSON(books)
write(json_books, file="books.json")
```

Chú ý: cần phải chuyển dữ liệu sang json data trước khi lưu vào tập tin



## □ Đọc nội dung JSON từ Internet

```
library("httr")
```

```
library("jsonlite")
```

```
url <- "http://phuong13021982.pythonanywhere.com/mystore/product_service/"  
get_url <- GET(url)
```

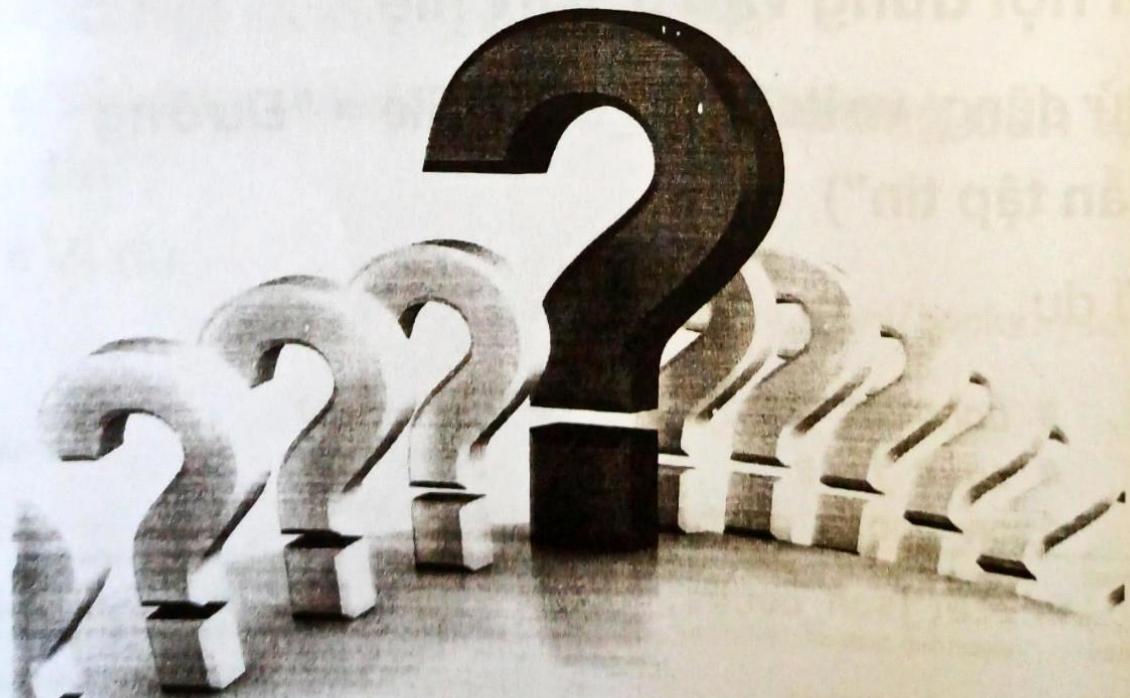
```
content <- rawToChar(get_url$content)
```

```
json <- fromJSON(content)
```

```
tivis <- data.frame(json)
```

```
print(tivis[c(1, 2, 3)])
```

	pk	name	fee
1	2	Asanzo 50 inch	9000000
2	11	Dell Vostro V3568 XF6C61	11999000
3	12	Macbook Air 2017 MQD32	18990000
4	3	Panasonic 40 inch	6590000
5	5	Samsung 32 inch	5999000
6	9	Samsung Galaxy J2 Prime	2690000
7	6	Sharp 45 inch	7490000
8	7	Sony 48 inch	11599000
9	4	Sony 55 inch	20590000
10	8	TCL 55 inch	9900000
11	1	Toshiba 32 inch	4590000
12	10	iPad WiFi 32GB New 2018	8390000



# Chapter 10: Làm việc với tập tin XML, JSON

## Exercise 1: Đọc và hiển thị tập tin XML

- Link: [\(http://www.w3schools.com/xml/plant\\_catalog.xml\)](http://www.w3schools.com/xml/plant_catalog.xml)
- Download nội dung xml từ link trên vào tập tin plant\_catalog.xml
- Đọc nội dung tập tin và chuyển thành dataframe
- Xuất nội dung dataframe
- Liệt kê những cây trồng quanh năm (Annual) và cần có nắng (Sunny)
  - [1] "Plants grow in sunny and annual:"

	COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
23	Black-Eyed Susan	Rudbeckia hirta	Annual	Sunny	\$9.80	061899
26	Butterfly Weed	Asclepias tuberosa	Annual	Sunny	\$2.78	063099

## Exercise 2: Ghi nội dung vào file XML

- Sử dụng dữ liệu mtcars
- Tạo tài liệu xml từ dữ liệu này nhưng chỉ lấy thông tin: names, mpg, wt, gear
- Ghi tài liệu xml này vào tập tin mtcars.xml
- Đọc tập tin vừa ghi để xem kết quả

names	mpg	wt	gear
Mazda RX4	21	2.62	4
Mazda RX4 Wag	21	2.875	4
Datsun 710	22.8	2.32	4
Hornet 4 Drive	21.4	3.215	3
Hornet Sportabout	18.7	3.44	3
Valiant	18.1	3.46	3

## Exercise 3: Đọc, xử lý và ghi nội dung JSON

- Cung cấp tập tin orange.json
- Đọc nội dung tập tin này => kiểm tra kiểu dữ liệu => đưa vào data.frame
- Cho biết cây cam có tuổi thọ cao nhất, thấp nhất
- Cho biết các cây cam có circumference >100 và age >1000. Có tất cả bao nhiêu cây cam?
- Chuyển dữ liệu những cây cam này thành json
- Ghi vào tập tin json
- Đọc nội dung tập tin vừa ghi để kiểm tra kết quả



	Tree	age	circumference
1	1	1004	115
2	1	1231	120
3	1	1372	142
4	1	1582	145
5	2	1004	156
6	2	1231	172
7	2	1372	203
8	2	1582	203
9	3	1004	108
10	3	1231	115
11	3	1372	139
12	3	1582	140
13	4	1004	167
14	4	1231	179
15	4	1372	209
16	4	1582	214
17	5	1004	125
18	5	1231	142
19	5	1372	174
20	5	1582	177

## Exercise 4: Đọc nội dung từ URL và ghi nội dung JSON

- Cung cấp URL: [http://phuong13021982.pythonanywhere.com/mystore/product\\_service/](http://phuong13021982.pythonanywhere.com/mystore/product_service/) ([http://phuong13021982.pythonanywhere.com/mystore/product\\_service/](http://phuong13021982.pythonanywhere.com/mystore/product_service/))
- Đọc nội dung từ URL này => đọc JSON => chuyển thành data.frame tên là Tivis
- Bỏ cột description trong Tivis
- Chuyển Tivis thành json

	pk		name	fee	image
1	2		Asanzo 50 inch	9000000	images/asanzo_4k_50_11690000.jpg
2	11	Dell Vostro V3568	XF6C61	11999000	images/Dell_vostro.jpg
3	12	Macbook Air 2017	MQD32	18990000	images/MacbooAir.jpg
4	3	Panasonic 40 inch		6590000	images/panasonic_40_6590000.jpg
5	5	Samsung 32 inch		5999000	images/samsung_32_5990000.jpg
6	9	Samsung Galaxy J2 Prime		2690000	images/SamsungPrime.jpg
7	6	Sharp 45 inch		7490000	images/sharp_45_7490000.jpg
8	7	Sony 48 inch		11599000	images/sony_48_11599000.png
9	4	Sony 55 inch		20590000	images/sony_55_20590000.jpg
10	8	TCL 55 inch		9900000	images/tcl_55_9900000.jpg
11	1	Toshiba 32 inch		4590000	images/toshiba32_4590000.jpg
12	10	iPad WiFi 32GB New 2018		8390000	images/iPad_Samsung_Wifi.jpg

- Ghi vào tập tin tivis.json
- Đọc nội dung của tập tin vừa ghi và xem kết quả

## Exercise 1: Đọc và hiển thị tập tin XML

```
In [2]: # Load the package required to read XML files.  
library("XML")  
# Also Load the other required package.  
library("methods")
```

```
In [3]: # download file xml  
fileUrl = "http://www.w3schools.com/xml/plant_catalog.xml"  
# tao file co ten la plant_catalog.xml  
download.file(fileUrl, destfile = "Du_lieu/plant_catalog.xml")
```

```
In [14]: # chuyen noi dung file sang data frame  
xmlDataFrame <- xmlToDataFrame("Du_lieu/plant_catalog.xml")  
print("Plants data frame:")  
head(xmlDataFrame)
```

```
[1] "Plants data frame:"
```

COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
Bloodroot	Sanguinaria canadensis	4	Mostly Shady	\$2.44	031599
Columbine	Aquilegia canadensis	3	Mostly Shady	\$9.37	030699
Marsh Marigold	Caltha palustris	4	Mostly Sunny	\$6.81	051799
Cowslip	Caltha palustris	4	Mostly Shady	\$9.90	030699
Dutchman's-Breeches	Dicentra cucullaria	3	Mostly Shady	\$6.44	012099
Ginger, Wild	Asarum canadense	3	Mostly Shady	\$9.03	041899

```
In [8]: #loc bo du Lieu cac cay trong quanh nam va can anh nang  
data_annual_sunny <- subset(xmlDataFrame, xmlDataFrame$ZONE=="Annual" &  
                                xmlDataFrame$LIGHT=="Sunny")  
print("Plants grow in sunny and annual:")  
data_annual_sunny
```

```
[1] "Plants grow in sunny and annual:"
```

	COMMON	BOTANICAL	ZONE	LIGHT	PRICE	AVAILABILITY
23	Black-Eyed Susan	Rudbeckia hirta	Annual	Sunny	\$9.80	061899
26	Butterfly Weed	Asclepias tuberosa	Annual	Sunny	\$2.78	063099

## Exercise 2: Ghi nội dung vào file XML

In [9]: # Load the packages required to read XML files.

```
library("XML")
library("methods")

df <- mtcars
df <- cbind(names = rownames(df), df)
rownames(df) <- c()
print(head(df))
```

	names	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
2	Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
3	Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
4	Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
5	Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
6	Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

In [10]: print("Create xml...")

```
[1] "Create xml..."
```

In [11]: doc = newXMLDoc()

```
# Simple creation of an XML tree using these functions
top = newXMLNode("cars", doc = doc)
for(row in 1:nrow(df)){
  carnode = newXMLNode("car", parent=top)
  newXMLNode("names", df[row, "names"], parent=carnode)
  newXMLNode("mpg", df[row, "mpg"], parent=carnode)
  newXMLNode("wt", df[row, "wt"], parent=carnode)
  newXMLNode("gear", df[row, "gear"], parent=carnode)
}
```

In [12]: # save file

```
print("Saving file...")
cat(saveXML(doc,
            indent = TRUE,
            prefix = "<?xml version=\"1.0\" encoding=\"utf-8\""
                      "standalone=\"no\"?>\n",
            file="Du_lieu/mtcars_new.xml")
print("Complete!")
```

```
[1] "Saving file..."
```

```
[1] "Complete!"
```

```
In [15]: # chuyen noi dung file sang data frame  
xmldataframe <- xmlToDataFrame("Du_lieu/mtcars_new.xml")  
head(xmldataframe)
```

	names	mpg	wt	gear
	Mazda RX4	21	2.62	4
	Mazda RX4 Wag	21	2.875	4
	Datsun 710	22.8	2.32	4
	Hornet 4 Drive	21.4	3.215	3
	Hornet Sportabout	18.7	3.44	3
	Valiant	18.1	3.46	3

## Exercise 3: Đọc, xử lý và ghi nội dung JSON

```
In [16]: # Load the package required to read JSON files.  
library("rjson")
```

```
In [24]: # Give the input file name to the function.  
result <- fromJSON(file = "Du_lieu/orange.json")  
# cho biet kieu du lieu cua orange  
print(paste("Data type:", class(result)))  
# neu khong phai la data frame thi doi thanh data frame  
# in ket qua  
data <- data.frame(result)  
print(head(data))
```

```
[1] "Data type: list"  
Tree age circumference  
1 1 118 30  
2 1 484 58  
3 1 664 87  
4 1 1004 115  
5 1 1231 120  
6 1 1372 142
```

```
In [18]: # cho biet trong nhung cay cam nay cay nao co tuoi tho cao nhat  
data_max_year <- data[which.max(data$age),]  
print("Max year:")  
print(data_max_year)
```

```
[1] "Max year:"  
Tree age circumference  
7 1 1582 145
```

```
In [19]: # cho biet trong nhung cay cam nay cay nao co tuoi tho thap nhat
data_min_year <- data[which.min(data$age),]
print("Min year:")
print(data_min_year)
```

```
[1] "Min year:"
Tree age circumference
1 1 118 30
```

```
In [25]: # danh sach cac cay trong co circumference >100 va age >1000
large_100_age_1000 <- subset(data, data$age>1000 &
                                data$circumference>100)
print(head(large_100_age_1000))
print(class(large_100_age_1000))
print(paste("Number of rows:", nrow(large_100_age_1000)))
```

```
Tree age circumference
4 1 1004 115
5 1 1231 120
6 1 1372 142
7 1 1582 145
11 2 1004 156
12 2 1231 172
```

```
[1] "data.frame"
[1] "Number of rows: 20"
```

```
In [21]: # ghi vao file
# doc file de kiem tra ket qua
large_100_age_1000_json <- toJSON(large_100_age_1000)
write(large_100_age_1000_json, file="Du_lieu/large_100_age_1000_new.json")
```

```
In [26]: # Give the input file name to the function.
result <- fromJSON(file = "Du_lieu/large_100_age_1000_new.json")
#in ket qua
data <- data.frame(result)
print(head(data))
```

```
Tree age circumference
1 1 1004 115
2 1 1231 120
3 1 1372 142
4 1 1582 145
5 2 1004 156
6 2 1231 172
```

## Exercise 4: Đọc nội dung từ URL và ghi nội dung JSON

```
In [27]: library(httr)
library("jsonlite")
```

Attaching package: 'jsonlite'

The following objects are masked from 'package:rjson':

fromJSON, toJSON

```
In [28]: # doc noi dung tu internet
```

```
URL <- "http://phuong13021982.pythonanywhere.com/mystore/product_service/"
getURL <- GET(URL)
content <- rawToChar(getURL$content)
json <- fromJSON(content)
Tivis <- data.frame(json)
print(class(Tivis))
#bo cot description
Tivis$description <-NULL
print(Tivis)
```

[1] "data.frame"

	pk	name	fee	image
1	2	Asanzo 50 inch	9000000	images/asanzo_4k_50_11690000.jpg
2	11	Dell Vostro V3568 XF6C61	11999000	images/Dell_vostro.jpg
3	12	Macbook Air 2017 MQD32	18990000	images/MacbooAir.jpg
4	3	Panasonic 40 inch	6590000	images/panasonic_40_6590000.jpg
5	5	Samsung 32 inch	5999000	images/samsung_32_5990000.jpg
6	9	Samsung Galaxy J2 Prime	2690000	images/SamsungPrime.jpg
7	6	Sharp 45 inch	7490000	images/sharp_45_7490000.jpg
8	7	Sony 48 inch	11599000	images/sony_48_11599000.png
9	4	Sony 55 inch	20590000	images/sony_55_20590000.jpg
10	8	TCL 55 inch	9900000	images/tcl_55_9900000.jpg
11	1	Toshiba 32 inch	4590000	images/toshiba32_4590000.jpg
12	10	iPad WiFi 32GB New 2018	8390000	images/iPad_Samsung_Wifi.jpg

```
In [33]: #ghi noi dung nay vao tap tin tivi.json'
```

```
Tivis_json <- toJSON(Tivis)
write(Tivis_json, file="Du_lieu/tivis_new.json")
```

```
In [35]: # Give the input file name to the function.  
result <- fromJSON(txt= "Du_lieu/tivis_new.json")  
# Print the result.  
print("json data read from file:")  
data <- data.frame(result)  
data
```

```
[1] "json data read from file:"
```

pk	name	fee	image
2	Asanzo 50 inch	9000000	images/asanzo_4k_50_11690000.jpg
11	Dell Vostro V3568 XF6C61	11999000	images/Dell_vostro.jpg
12	Macbook Air 2017 MQD32	18990000	images/MacbooAir.jpg
3	Panasonic 40 inch	6590000	images/panasonic_40_6590000.jpg
5	Samsung 32 inch	5999000	images/samsung_32_5990000.jpg
9	Samsung Galaxy J2 Prime	2690000	images/SamsungPrime.jpg
6	Sharp 45 inch	7490000	images/sharp_45_7490000.jpg
7	Sony 48 inch	11599000	images/sony_48_11599000.png
4	Sony 55 inch	20590000	images/sony_55_20590000.jpg
8	TCL 55 inch	9900000	images/tcl_55_9900000.jpg
1	Toshiba 32 inch	4590000	images/toshiba32_4590000.jpg
10	iPad WiFi 32GB New 2018	8390000	images/iPad_Samsung_Wifi.jpg