

# NHẬP MÔN PHÂN TÍCH DỮ LIỆU

Các thao tác với dữ liệu

Khoa Toán - Cơ - Tin học  
Trường Đại học Khoa học Tự nhiên  
Đại học Quốc gia Hà Nội

Ngày 20 tháng 10 năm 2024

- 1 Nhập xuất dữ liệu
- 2 Xử lý dữ liệu
- 3 Thao tác với dữ liệu
- 4 Xử lý dữ liệu bị thiếu
- 5 Lập trình cơ bản trong R

1 Nhập xuất dữ liệu

2 Xử lý dữ liệu

3 Thao tác với dữ liệu

4 Xử lý dữ liệu bị thiếu

5 Lập trình cơ bản trong R

## 1.1. Nhập dữ liệu trực tiếp bằng lệnh c()

### Nhập dữ liệu

```
> age <- c(50, 62, 60, 40, 48, 47, 57, 70, 48)
> insulin <- c(16.5, 10.8, 32.3, 19.3, 14.2, 11.3, 15.5, 15.8, 16.2)
> data = data.frame(age,insulin)
> data
```

	age	insulin
1	50	16.5
2	62	10.8
3	60	32.3
4	40	19.3
5	48	14.2
6	47	11.3
7	57	15.5
8	70	15.8
9	48	16.2

## 1.1. Nhập dữ liệu trực tiếp bằng lệnh c()

### Nhập dữ liệu

```
> age <- c(50, 62, 60, 40, 48, 47, 57, 70, 48)
> insulin <- c(16.5, 10.8, 32.3, 19.3, 14.2, 11.3, 15.5, 15.8, 16.2)
> data = data.frame(age,insulin)
> data
```

	age	insulin
1	50	16.5
2	62	10.8
3	60	32.3
4	40	19.3
5	48	14.2
6	47	11.3
7	57	15.5
8	70	15.8
9	48	16.2

### Xuất dữ liệu

```
> setwd("D:\\Monhoc\\NhapmonPTDL")
> save(data, file = "Tuan4.rda")
```

## 1.2. Đọc dữ liệu từ tệp

### Nhập dữ liệu từ tệp excel

```
> setwd("D:\\Monhoc\\NhapmonPTDL")  
> data = read.csv("dulieu.csv", header = TRUE)  
> View(data)
```

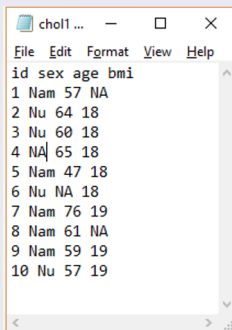
## 1.2. Đọc dữ liệu từ tệp

### Nhập dữ liệu từ tệp excel

```
> setwd("D:\\Monhoc\\NhapmonPTDL")  
> data = read.csv("dulieu.csv", header = TRUE)  
> View(data)
```

	OrderDate	Region	City	Category	Product	Quantity	UnitPrice	TotalPrice
1	1/1/2020	East	Boston	Bars	Carrot	33	1.77	58.41
2	4/1/2020	East	Boston	Crackers	Whole Wheat	87	3.49	303.63
3	7/1/2020	West	Los Angeles	Cookies	Chocolate Chip	58	1.87	108.46
4	10/1/2020	East	New York	Cookies	Chocolate Chip	82	1.87	153.34
5	13/1/2020	East	Boston	Cookies	Arrowroot	38	2.18	82.84
6	16/1/2020	East	Boston	Bars	Carrot	54	1.77	95.58
7	19/1/2020	East	Boston	Crackers	Whole Wheat	149	3.49	520.01
8	22/1/2020	West	Los Angeles	Bars	Carrot	51	1.77	90.27
9	25/1/2020	East	New York	Bars	Carrot	100	1.77	177.00
10	28/1/2020	East	New York	Snacks	Potato Chips	28	1.35	37.80
11	31/1/2020	East	Boston	Cookies	Arrowroot	36	2.18	78.48
12	3/2/2020	East	Boston	Cookies	Chocolate Chip	31	1.87	57.97

## Nhập dữ liệu từ tệp .txt



id	sex	age	bmi
1	Nam	57	NA
2	Nu	64	18
3	Nu	60	18
4	NA	65	18
5	Nam	47	18
6	Nu	NA	18
7	Nam	76	19
8	Nam	61	NA
9	Nam	59	19
10	Nu	57	19

```
> chol1 <- read.table("chol1.txt", header=TRUE)
> chol1.new <- na.omit(chol1)
> chol1
  id sex age bmi
1  1  Nam  57  NA
2  2   Nu  64  18
3  3   Nu  60  18
4  4 <NA>  65  18
5  5  Nam  47  18
6  6   Nu   NA  18
7  7  Nam  76  19
8  8  Nam  61  NA
9  9  Nam  59  19
10 10   Nu  57  19
> chol1.new
  id sex age bmi
2  2   Nu  64  18
3  3   Nu  60  18
5  5  Nam  47  18
7  7  Nam  76  19
9  9  Nam  59  19
10 10   Nu  57  19
```



## 1.3. Lấy dữ liệu từ cơ sở dữ liệu

### Lấy dữ liệu từ package

```
> library(MASS)
> data(Boston)
> dim(Boston)
[1] 506 14
> View(Boston)
```

## 1.3. Lấy dữ liệu từ cơ sở dữ liệu

### Lấy dữ liệu từ package

```
> library(MASS)
> data(Boston)
> dim(Boston)
[1] 506 14
> View(Boston)
```

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
1	0.00632	18.0	2.31	0	0.5380	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
2	0.02731	0.0	7.07	0	0.4690	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
3	0.02729	0.0	7.07	0	0.4690	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
4	0.03237	0.0	2.18	0	0.4580	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
5	0.06905	0.0	2.18	0	0.4580	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
6	0.02985	0.0	2.18	0	0.4580	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7
7	0.08829	12.5	7.87	0	0.5240	6.012	66.6	5.5605	5	311	15.2	395.60	12.43	22.9
8	0.14455	12.5	7.87	0	0.5240	6.172	96.1	5.9505	5	311	15.2	396.90	19.15	27.1
9	0.21124	12.5	7.87	0	0.5240	5.631	100.0	6.0821	5	311	15.2	386.63	29.93	16.5
10	0.17004	12.5	7.87	0	0.5240	6.004	85.9	6.5921	5	311	15.2	386.71	17.10	18.9
11	0.22489	12.5	7.87	0	0.5240	6.377	94.3	6.3467	5	311	15.2	392.52	20.45	15.0
12	0.11747	12.5	7.87	0	0.5240	6.009	82.9	6.2267	5	311	15.2	396.90	13.27	18.9
13	0.09378	12.5	7.87	0	0.5240	5.889	39.0	5.4509	5	311	15.2	390.50	15.71	21.7

- 1 Nhập xuất dữ liệu
- 2 **Xử lý dữ liệu**
- 3 Thao tác với dữ liệu
- 4 Xử lý dữ liệu bị thiếu
- 5 Lập trình cơ bản trong R

## 2.1. Mã hóa dữ liệu

Trong suốt phần này ta sẽ sử dụng dữ liệu "df=mtcars" được lấy từ tạp chí *Motor Trend US* năm 1974. Trong data set này, biến "am" là kiểu hộp số được mã hóa với 2 giá trị 0 (số tự động), 1(số điều khiển bằng tay). Chúng ta sẽ mã hóa lại biến số am thành dạng factor với tên lần lượt là "automatic" và "manual" và gán cho một biến số mới có tên là "trans".

## 2.1. Mã hóa dữ liệu

Trong suốt phần này ta sẽ sử dụng dữ liệu "df=mtcars" được lấy từ tạp chí *Motor Trend US* năm 1974. Trong data set này, biến "am" là kiểu hộp số được mã hóa với 2 giá trị 0 (số tự động), 1(số điều khiển bằng tay). Chúng ta sẽ mã hóa lại biến số am thành dạng factor với tên lần lượt là "automatic" và "manual" và gán cho một biến số mới có tên là "trans".

### Sử dụng lệnh factor

```
> df <- mtcars
> df$trans <- factor(df$am, levels=c(0, 1), labels=c("Automatic", "Manual"))
> df$trans
Levels: Automatic Manual
```

## 2.1. Mã hóa dữ liệu

Trong suốt phần này ta sẽ sử dụng dữ liệu "df=mtcars" được lấy từ tạp chí *Motor Trend US* năm 1974. Trong data set này, biến "am" là kiểu hộp số được mã hóa với 2 giá trị 0 (số tự động), 1(số điều khiển bằng tay). Chúng ta sẽ mã hóa lại biến số am thành dạng factor với tên lần lượt là "automatic" và "manual" và gán cho một biến số mới có tên là "trans".

### Sử dụng lệnh factor

```
> df <- mtcars
> df$trans <- factor(df$am, levels=c(0, 1), labels=c("Automatic", "Manual"))
> df$trans
Levels: Automatic Manual
```

### Sử dụng lệnh match

```
> oldvalues <- c("0", "1")
> newvalues <- factor(c("Automatic", "Manual"))
> df$trans2 <- newvalues[match(df$am, oldvalues)]
```

## 2.1. Mã hóa dữ liệu

Trong suốt phần này ta sẽ sử dụng dữ liệu "df=mtcars" được lấy từ tạp chí *Motor Trend US* năm 1974. Trong data set này, biến "am" là kiểu hộp số được mã hóa với 2 giá trị 0 (số tự động), 1(số điều khiển bằng tay). Chúng ta sẽ mã hóa lại biến số am thành dạng factor với tên lần lượt là "automatic" và "manual" và gán cho một biến số mới có tên là "trans".

### Sử dụng lệnh factor

```
> df <- mtcars  
> df$trans <- factor(df$am, levels=c(0, 1), labels=c("Automatic", "Manual"))  
> df$trans  
Levels: Automatic Manual
```

### Sử dụng lệnh Replace

```
> df$trans3 <- df$am  
> df$trans3 <- replace(df$trans3, df$am == 1, "Manual")  
> df$trans3 <- replace(df$trans3, df$am == 0, "Automatic")
```

Giả sử chúng ta muốn mã hóa dữ liệu công suất của 32 dòng ô tô thành 3 nhóm: cao (H), trung bình (M), thấp (L) và gán các giá trị này vào một biến số mới có tên "power".



Giả sử chúng ta muốn mã hóa dữ liệu công suất của 32 dòng ô tô thành 3 nhóm: cao (H), trung bình (M), thấp (L) và gán các giá trị này vào một biến số mới có tên "power".

### Sử dụng R's built-in function

```
> df$power[df$hp < 96.5] <- "L"  
> df$power[df$hp > 180] <- "H"  
> df$power[96.5 <= df$hp & df$hp <= 180] <- "M"
```

Giả sử chúng ta muốn mã hóa dữ liệu công suất của 32 dòng ô tô thành 3 nhóm: cao (H), trung bình (M), thấp (L) và gán các giá trị này vào một biến số mới có tên "power".

### Sử dụng R's built-in function

```
> df$power[df$hp < 96.5] <- "L"  
> df$power[df$hp > 180] <- "H"  
> df$power[96.5 <= df$hp & df$hp <= 180] <- "M"
```

### Sử dụng hàm cut

```
> df$power <- cut(df$hp,  
  breaks=c(-Inf, 96.5, 180, Inf),  
  labels=c("L", "M", "H"))  
> df$power
```

## 2.2. Chuyển đổi giữa các loại vector

### Dạng số sang Character và Factor

```
> df$vs <- factor(df$vs)
# Chúng ta cũng có thể chỉ định thứ tự của các mức khi chúng ta tạo các biến hoặc
chuyển đổi nó
> df$vs <- factor(mtcars$vs, levels=c("0", "1"))
> df$vs1 <- factor(mtcars$vs, levels=c("1", "0"))
> df$vs2 <- as.character(mtcars$vs)
> str(df)
```

## 2.2. Chuyển đổi giữa các loại vector

### Dạng số sang Character và Factor

```
> df$vs <- factor(df$vs)
# Chúng ta cũng có thể chỉ định thứ tự của các mức khi chúng ta tạo các biến hoặc chuyển đổi nó
> df$vs <- factor(mtcars$vs, levels=c("0", "1"))
> df$vs1 <- factor(mtcars$vs, levels=c("1", "0"))
> df$vs2 <- as.character(mtcars$vs)
> str(df)
```

```
## 'data.frame': 32 obs. of 13 variables:
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num 160 160 108 258 360 ...
## $ hp : num 110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs : Factor w/ 2 levels "0","1": 1 1 2 2 1 2 1 2 2 2 ...
## $ am : num 1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...
## $ vs1 : Factor w/ 2 levels "1","0": 2 2 1 1 2 1 2 1 1 1 ...
## $ vs2 : chr "0" "0" "1" "1" ...
```

## 2.2. Chuyển đổi giữa các loại vector

### Character sang Factor và dạng số

Đổi dữ liệu từ dạng factor sang dạng character rất dễ dàng bằng lệnh `as.character()`.

```
> df$vs2 <- as.numeric(df$vs2)  
> df$vs2 <- as.factor(as.character(mtcars$vs))
```

## 2.2. Chuyển đổi giữa các loại vector

### Character sang Factor và dạng số

Đổi dữ liệu từ dạng factor sang dạng character rất dễ dàng bằng lệnh `as.character()`.

```
> df$vs2 <- as.numeric(df$vs2)
> df$vs2 <- as.factor(as.character(mtcars$vs))
```

### Factor sang character và dạng numeric

Biến đổi từ dạng factor sang dạng numeric phức tạp hơn. Nếu chỉ sử dụng `as.numeric()` thì chỉ có số mã hóa cho factor được chuyển sang dạng số. Do vậy, trước hết ta biến đổi biến factor sang dạng character sau đó mới chuyển về dạng số.

```
> df$vs1 <- as.character(df$vs)
> df$vs2 <- as.numeric(as.character(df$vs2))
> str(df)
```

## 2.3. Sắp xếp dữ liệu

Sắp xếp véc tơ dạng số bằng lệnh sort

```
> v <- sample(100:110)
> sort(v, decreasing=TRUE)
[1] 110 109 108 107 106 105 104 103 102 101 100
```

1 Nhập xuất dữ liệu

2 Xử lý dữ liệu

3 Thao tác với dữ liệu

4 Xử lý dữ liệu bị thiếu

5 Lập trình cơ bản trong R



## 3.1. Trích dữ liệu

### Trích dữ liệu bằng lệnh Subset

```
> Automatic = subset(df, df$am == 0) # trích dữ liệu con gồm các hàng có am = 0
> Manual = subset(df, df$am == 1) # trích dữ liệu con gồm các hàng có am = 1
> dim(Manual)
[1] 13 11
> Manual3 = subset(df, df$am == 1 & df$wt > 2)
> dim(Manual3)
[1] 9 11
```

## 3.1. Trích dữ liệu

### Trích dữ liệu bằng lệnh Subset

```
> Automatic = subset(df, df$am == 0) # trích dữ liệu con gồm các hàng có am = 0
> Manual = subset(df, df$am == 1) # trích dữ liệu con gồm các hàng có am = 1
> dim(Manual)
[1] 13 11
> Manual3 = subset(df, df$am == 1 & df$wt > 2)
> dim(Manual3)
[1] 9 11
```

### Sử dụng R's built-in function

```
> data1 = df[3,] # Trích véc tơ hàng 3 của dữ liệu df
> data2 = df[,3] # Trích véc tơ cột 3 của dữ liệu df
> data4 = df[c(1,2,3), c(4,5,6)] # Trích dữ liệu con gồm hàng 1, 2, 3 và cột 4, 5, 6 của dữ liệu df.
> Manual = df[df$am == 1,] # trích dữ liệu con gồm các hàng có am = 0
> Manual3 = df[df$am == 1 & df$wt > 2,]
> dim(Manual3)
[1] 9 11
```

## 3.2. Ghép dữ liệu

### Sử dụng lệnh merge()

```
stories2 <- read.table(header=TRUE, text='
  id      title
  1      lions
  2      tigers
  3      bears
')
data <- data <- read.table(header=TRUE, text='
  subject storyid rating
      1      1    6.7
      1      2    4.5
      1      3    3.7
      2      2    3.3
      2      3    4.1
      2      1    5.2
')
merge(x=stories2, y=data, by.x="id", by.y="storyid")
```

## 3.2. Ghép dữ liệu

### Kết quả

##	id	title	subject	rating
## 1	1	lions	1	6.7
## 2	1	lions	2	5.2
## 3	2	tigers	1	4.5
## 4	2	tigers	2	3.3
## 5	3	bears	1	3.7
## 6	3	bears	2	4.1

## 3.2. Ghép dữ liệu

### Sử dụng lệnh merge()

Ngoài ra lệnh merge cũng cho phép nối nhiều cột trong 2 data frame với nhau.

```
animals <- read.table(header=T, text='
  size type      name
small  cat       lynx
big    cat       tiger
small  dog       chihuahua
big    dog       "great dane"
cat    dog       dog
')

observations <- read.table(header=T, text='
  number size type
1    big  cat
2 small  dog
3 small  dog
4    big  dog
')
merge(animals, observations, c("size", "type"))
```

## 3.2. Ghép dữ liệu

### Kết quả

##	size	type	name	number
## 1	big	cat	tiger	1
## 2	big	dog	great dane	4
## 3	small	dog	chihuahua	2
## 4	small	dog	chihuahua	3

### 3.3. Sinh dữ liệu ngẫu nhiên

#### Sinh dữ liệu đơn giản

```
> x = 1:200 # Lấy liên tiếp từ 1 đến 200  
> y = sample(c(2,3,4), 200, replace = TRUE) # Sinh ra véc tơ có độ dài 200 được lấy  
từ véc tơ (2, 3, 4) có lặp lại  
> z = sample(seq(1,10, 0.5), 200, replace = T)  
> t = sample(c("Nam", "Nu"), 200, replace = T) # Sinh dữ liệu định tính  
> data = data.frame(x,y,z,t)
```

### 3.3. Sinh dữ liệu ngẫu nhiên

#### Sinh ra các phân bố

Phân bố	Hàm mật độ	H. Phân bố	Mô phỏng
Chuẩn	<code>dnorm(x,mean,sd)</code>	<code>pnorm(x,mean,sd)</code>	<code>rnorm(n,mean,sd)</code>
Nhi phân	<code>dbinom(k,n,p)</code>	<code>pbinom(k,n,p)</code>	<code>rbinom(k,n,prob)</code>
Poisson	<code>dpois(k, lambda)</code>	<code>ppois(q, lambda)</code>	<code>rpois(n, lambda)</code>
Uniform	<code>dunif(x, min, max)</code>	<code>punif(q, min, max)</code>	<code>runif(x, min, max)</code>
Nhi thức âm	<code>dnbinom(x,k,p)</code>	<code>pnbinom(q,k,p)</code>	<code>rnbino(m,n,prob)</code>
Beta	<code>dbeta(x,shape1,shape2)</code>	<code>pbeta(q,shape1,shape2)</code>	<code>rbeta(n,shape1,shape2)</code>
Khi bình phương	<code>dchisq(x,df)</code>	<code>pchisq(q,df)</code>	<code>rchisq(x,df)</code>
Hình học	<code>dgeom(x,p)</code>	<code>pgeom(q,p)</code>	<code>rgeom(n,prob)</code>
Mũ	<code>dexp(x,rate)</code>	<code>pexp(q,rate)</code>	<code>rexp(n,rate)</code>
Student	<code>dt(x,df)</code>	<code>pt(q,df)</code>	<code>rt(n,df)</code>
Fisher	<code>df(x,df1,df2)</code>	<code>pf(q,df1,df2)</code>	<code>rf(n,df1,df2)</code>



## Bài tập 1

- Sinh ra một bộ dữ liệu điểm của sinh viên 234 hàng và bao gồm: mã sinh viên được đánh số từ 1000 đến 2345, Điểm thường xuyên, điểm giữa kỳ, điểm cuối kỳ.
- Tính điểm trung bình theo công thức 20-20-60 và ghép vào dữ liệu ban đầu.
- Phân loại thành điểm chữ và ghép vào dữ liệu ban đầu.

## Bài tập 2

Dữ liệu "Product.csv" được cung cấp bao gồm 24 hàng và 8 biến, đọc tệp đã cho và đặt tên nó là dat. Thực hiện các yêu cầu sau

- Trích ra một bộ dữ liệu con bao gồm các mặt hàng được bán tại thành phố "Boston" và đặt tên nó là dat1.
- Tính tổng giá của mặt hàng "Carrot".
- Tính trung bình số lượng sản phẩm "Carrot" Được bán tại thành phố Boston.

- 1 Nhập xuất dữ liệu
- 2 Xử lý dữ liệu
- 3 Thao tác với dữ liệu
- 4 Xử lý dữ liệu bị thiếu**
- 5 Lập trình cơ bản trong R

## 4.1. Xóa bỏ dữ liệu bị thiếu

### Sử dụng lệnh na.omit

```
> train = fread("Train_UWu5bXk.csv")  
> test = fread("Test_u94Q5KV.csv")  
> dim(train)  
[1] 8523 12
```

## 4.1. Xóa bỏ dữ liệu bị thiếu

### Sử dụng lệnh na.omit

```
> train = fread("Train_UWu5bXk.csv")
> test = fread("Test_u94Q5KV.csv")
> dim(train)
[1] 8523 12
> train = na.omit(train) # Xóa các hàng chứa dữ liệu thiếu
> dim(train)
[1] 7060 12
```

## 4.2. Thay thế dữ liệu định lượng bị thiếu

### Tìm thông tin dữ của liệu bị thiếu

Ta sẽ sử dụng hai bộ dữ liệu trên tập "train" và tập "test"

```
> train = fread("Train_UWu5bXk.csv")  
> test = fread("Test_u94Q5KV.csv")  
> test[,Item_Outlet_Sales := NA]  
> combi = rbind(train, test) # combining train and test datasets  
> colSums(is.na(combi))
```

## 4.2. Thay thế dữ liệu định lượng bị thiếu

### Tìm thông tin dữ của liệu bị thiếu

Ta sẽ sử dụng hai bộ dữ liệu trên tập "train" và tập "test"

```
> train = fread("Train_UWu5bXk.csv")
> test = fread("Test_u94Q5KV.csv")
> test[,Item_Outlet_Sales := NA]
> combi = rbind(train, test) # combining train and test datasets
> colSums(is.na(combi))
```

Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility
0	2439	0	0
Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year
0	0	0	0
Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	0	0	5681

## 4.2. Thay thế dữ liệu định lượng bị thiếu

### Thay thế dữ liệu thiếu bằng giá trị trung bình theo nhóm

```
> missing_index = which(is.na(combi$Item_Weight))  
> for(i in missing_index) {  
  item = combi$Item_Identifier[i]  
  combi$Item_Weight[i] = mean(combi$Item_Weight[combi$Item_Identifier  
    == item], na.rm = T) }  
> colSums(is.na(combi))
```

## 4.2. Thay thế dữ liệu định lượng bị thiếu

### Thay thế dữ liệu thiếu bằng giá trị trung bình theo nhóm

```
> missing_index = which(is.na(combi$Item_Weight))
> for(i in missing_index) {
  item = combi$Item_Identifier[i]
  combi$Item_Weight[i] = mean(combi$Item_Weight[combi$Item_Identifier
    == item], na.rm = T) }
> colSums(is.na(combi))
```

Item_Identifier	Item_Weight	Item_Fat_Content	Item_Visibility
0	0	0	0
Item_Type	Item_MRP	Outlet_Identifier	Outlet_Establishment_Year
0	0	0	0
Outlet_Size	Outlet_Location_Type	Outlet_Type	Item_Outlet_Sales
0	0	0	5681



- 1 Nhập xuất dữ liệu
- 2 Xử lý dữ liệu
- 3 Thao tác với dữ liệu
- 4 Xử lý dữ liệu bị thiếu
- 5 Lập trình cơ bản trong R**

## 5.1. Câu lệnh if, for

### Cấu trúc câu lệnh

```
> if("Điều kiện"){ Kết quả } else { Kết quả trái lại }  
> for(i in "véctơ của chỉ số i") { nội dung của hàm for }
```

## 5.1. Câu lệnh if, for

### Cấu trúc câu lệnh

```
> if("Điều kiện"){ Kết quả } else { Kết quả trái lại }  
> for(i in "véctơ của chỉ số i") { nội dung của hàm for }
```

### Ví dụ 1

So sánh hay giá trị trung bình.

```
> x = c( 1, 2, 3, 5, 6, 3, 2, 1)  
> y = seq(1, 6, 0.5)  
> a = mean(x)  
> b = mean(y)  
> if(a>=b) {  
    print("Giá trị trung bình của x lớn hơn hoặc bằng y") } else {  
    print("Giá trị trung bình của x nhỏ hơn y") }  
[1] "Giá trị trung bình của x nhỏ hơn y"
```

## 5.1. Câu lệnh if, for

### Ví dụ 2

Sinh ngẫu nhiên hai vec tơ cùng độ dài 10, tính tổng theo công thức  $tong = x^2 + y$ .

```
> x = sample(c(2,7,4,6), 10, replace = T)
```

```
> y = sample(20:50, 10, replace = T)
```

```
> tong = 0
```

```
> for(i in 1:length(x)) {  
    tong = tong + x[i]^2 + y[i]  
}
```

```
> tong
```

```
[1] 606
```

## 5.2. Viết hàm trong R

### Cấu trúc hàm

```
>tenham = function("Biến vào") {  
  Nội dung của hàm  
}
```

## 5.2. Viết hàm trong R

### Cấu trúc hàm

```
>tenham = function("Biến vào") {  
    Nội dung của hàm  
}
```

### Ví dụ 3

Viết hàm truyền vào một véc tơ  $x$  và tính tổng theo công thức  $tong = \sum_{i=1}^{length(x)} x_k^k$  với  $x_k$  là vị trí thứ  $k$  của véc tơ  $x$ . (Ví dụ  $x = (2, 3, 4, 5)$ ,  $tong = 2^1 + 3^2 + 4^3 + 5^4$ )

```
> tinh tong = function(x){  
    > tong = 0  
    > for(i in 1:length(x)){  
        tong = tong + (x[i])^i  
    }  
    return(tong)  
}  
> x = c(2, 3, 4, 5)  
> tinh tong(x)  
[1] 700
```

## Bài tập 1

Từ dữ liệu “WHO1”, ta có mức thu nhập quốc dân (GNI) của 195 quốc gia. Với những quốc gia có GNI lớn hơn 20 nghìn ta xét vào nhóm quốc gia phát triển, những quốc gia có GNI nằm trong khoảng từ 10 – 20 nghìn được xếp vào nhóm quốc gia đang phát triển, các nước có GNI nhỏ hơn 10 nghìn được xếp vào nhóm các quốc gia chưa phát triển, còn lại là các quốc gia khác. Sử dụng R và thực hiện các yêu cầu sau

- Nhập dữ liệu “WHO1” vào R. Loại bỏ các dữ liệu trống bằng cách bổ sung bằng giá trị median của dữ liệu khi xóa bỏ các vị trí trống.
- Từ dữ liệu trích ra dữ liệu của hai nước “Bahrain” và “Mexico” và so sánh tuổi thọ của hai nước này.
- Hãy cho biết tỷ lệ sinh sản của nước “Canada”.
- Viết thêm cột phân nhóm các quốc gia vào dữ liệu. Hãy cho biết nước “China” thuộc nhóm nào?
- Hãy viết một hàm để tìm nước có dân số cao nhất. Sử dụng hàm với dữ liệu WHO1.
- Hãy viết một hàm để kiểm tra tên một nước sẽ thuộc nhóm nào theo phân loại ở câu d. Sử dụng hàm với dữ liệu WHO1.