R for Data Science

Bài 8: *Data Frame và Tidyverse*

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-csdl/R-Programming-Language-for-Data-Science_190

2020

Nội dung

1. **Data Frame**
2. **Tidyverse**

Data frame

- Là một table hoặc là một mảng hai chiều trong đó mỗi cột chứa các giá trị của một biến và mỗi dòng chứa một bộ các giá trị từ các cột.
- Đặc điểm:
  - Tên cột không để trống
  - Tên dòng là duy nhất.
  - Dữ liệu trong data frame có thể là numeric, factor, character
  - Mỗi cột chỉ chứa cùng một loại dữ liệu

Data frame

- **Sử dụng data.frame để tạo data frame**

  data.frame(col_name_1 = vector_values, …)

  - Ví dụ:

```
students_1 <- data.frame("name" = c('Hellen', 'Jolie', 'Kiwi', 'John', 'Mark', 'Tom'),
                         "age" = c(17, 16, 17, 18, 19, 16),
                         "mark" = c(9, 9, 8, 7.5, 7.5, 8),
                         stringsAsFactors = FALSE
                        )

print(students_1)

    name age mark
1 Hellen  17  9.0
2  Jolie  16  9.0
3   Kiwi  17  8.0
4   John  18  7.5
5   Mark  19  7.5
6    Tom  16  8.0
```

Data frame

Hoặc data.frame(vector1, vector2,…)

- Ví dụ:

```
name <- c('Hellen', 'Jolie', 'Kiwi', 'John', 'Mark', 'Tom')
age <- c(17, 16, 17, 18, 19, 16)
mark <- c(9, 9, 8, 7.5, 7.5, 8)
students <- data.frame(name, age, mark,
                       stringsAsFactors = FALSE)

print(students)
```

```
    name age mark
1 Hellen  17  9.0
2  Jolie  16  9.0
3   Kiwi  17  8.0
4   John  18  7.5
5   Mark  19  7.5
6    Tom  16  8.0
```

Data frame

❑Xem cấu trúc của data frame bằng str()

- Ví dụ:

```
str(students)
```

```
'data.frame':   6 obs. of  3 variables:
 $ name: chr  "Hellen" "Jolie" "Kiwi" "John" ...
 $ age : num  17 16 17 18 19 16
 $ mark: num  9 9 8 7.5 7.5 8
```

Data frame

Data Science

❑Xem tóm tắt thống kê dữ liệu trong data frame với summary()

- Ví dụ:

```
print(summary(students))
```

```
     name                age             mark
 Length:6           Min.   :16.00   Min.   :7.500
 Class :character   1st Qu.:16.25   1st Qu.:7.625
 Mode  :character   Median :17.00   Median :8.000
                    Mean   :17.17   Mean   :8.167
                    3rd Qu.:17.75   3rd Qu.:8.750
                    Max.   :19.00   Max.   :9.000
```

Data frame

Data Science

❑Trích xuất dữ liệu từ data frame

- Ví dụ

```
# get mark column as vector
print(students$mark)
```

```
[1] 9.0 9.0 8.0 7.5 7.5 8.0
```

```
# get age column as data.farme
print(data.frame(students$age))
```

```
  students.age
1           17
2           16
3           17
4           18
5           19
6           16
```

```
# get 2 first rows
print(students[1:2,])
```

```
    name age mark
1 Hellen  17    9
2  Jolie  16    9
```

```
# get rows 3, 5 & columns 1,2
print(students[c(3, 5), c(1,2)])
```

```
  name age
3 Kiwi  17
5 Mark  19
```

Data frame

❑Mở rộng data frame

- Thêm cột: data_frame_name$column_name <- vector_values
- Ví dụ:

```
students$class <- c("k256_S7N", "k256_C7N", "k256_S24", "k256_S7N", "k256_S24", "k256_C7N")
```

```
students
```

name	age	mark	class
Hellen	17	9.0	k256_S7N
Jolie	16	9.0	k256_C7N
Kiwi	17	8.0	k256_S24
John	18	7.5	k256_S7N
Mark	19	7.5	k256_S24
Tom	16	8.0	k256_C7N

Data frame

❑Mở rộng data frame

- Thêm cột: cbind(data_frame, data_frame_new) để nối lại thành một data frame cuối cùng
- Ví dụ:

```
df_POB <- data.frame("POB" = c("UK", "US", "EU", "UK", "US", "EU"))
students<- cbind(students, df_POB)
```

```
students
```

name	age	mark	class	POB
Hellen	17	9.0	k256_S7N	UK
Jolie	16	9.0	k256_C7N	US
Kiwi	17	8.0	k256_S24	EU
John	18	7.5	k256_S7N	UK
Mark	19	7.5	k256_S24	US
Tom	16	8.0	k256_C7N	EU

❑Mở rộng data frame

- Thêm dòng: bằng cách tạo ra một data frame mới có cùng cấu trúc với data frame muốn thêm dòng sau đó dùng rbind(data_frame, data_frame_new) để nối lại thành một data frame cuối cùng

- Ví dụ

```
new_students <- data.frame("name" = c('Sunny', 'Lucy'),
                           "age" = c(17, 18),
                           "mark" = c(8, 9),
                           "class" = c("k256_C7N", "k256_S24"),
                           "POB" = c("EU", "UK"),
                           stringsAsFactors = FALSE)
```

```
print(new_students)
```

```
   name age mark     class POB
1 Sunny  17    8 k256_C7N  EU
2  Lucy  18    9 k256_S24  UK
```

```
students<- rbind(students, new_students)
```

```
students
```

name	age	mark	class	POB
Hellen	17	9.0	k256_S7N	UK
Jolie	16	9.0	k256_C7N	US
Kiwi	17	8.0	k256_S24	EU
John	18	7.5	k256_S7N	UK
Mark	19	7.5	k256_S24	US
Tom	16	8.0	k256_C7N	EU
Sunny	17	8.0	k256_C7N	EU
Lucy	18	9.0	k256_S24	UK
Sunny	17	8.0	k256_C7N	EU
Lucy	18	9.0	k256_S24	UK

❑ Thay đổi tên cột

- names(tên_dataframe)[names(tên_dataframe)=="tên_cột_cũ"] <- tên_cột_mới
- Ví dụ:

```
names(students)[names(students)=="POB"] <- "Place_of_birth"
```

```
head(students)
```

name	age	mark	class	POB
Hellen	17	9.0	k256_S7N	UK
Jolie	16	9.0	k256_C7N	US
Kiwi	17	8.0	k256_S24	EU
John	18	7.5	k256_S7N	UK
Mark	19	7.5	k256_S24	US
Tom	16	8.0	k256_C7N	EU

```
head(students)
```

name	age	mark	class	Place_of_birth
Hellen	17	9.0	k256_S7N	UK
Jolie	16	9.0	k256_C7N	US
Kiwi	17	8.0	k256_S24	EU
John	18	7.5	k256_S7N	UK
Mark	19	7.5	k256_S24	US
Tom	16	8.0	k256_C7N	EU

Nội dung

1. Data Frame
2. Tidyverse

Tidyverse

Data Science

❑ Giới thiệu

- Chuyên dùng cho Data Science
- Phù hợp cho việc quản lý và tiền xử lý dữ liệu

R packages for data science

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

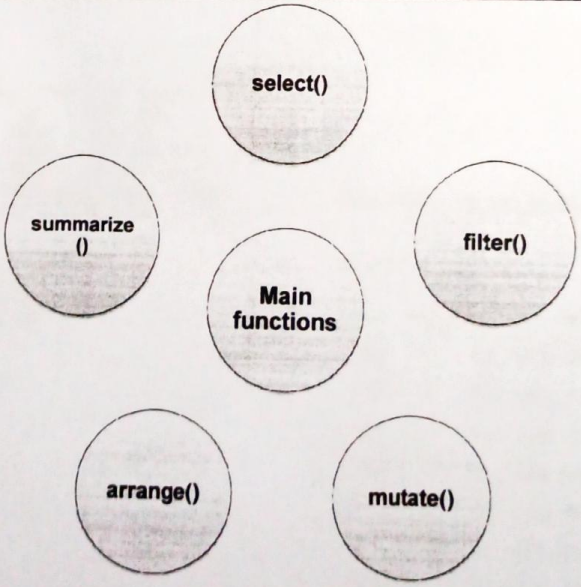
Install the complete tidyverse with:

```
install.packages("tidyverse")
```

https://www.tidyverse.org/

Tidyverse

Data Science



Cú pháp:

```
function_name(dataframe, conditions)
```

Hoặc `dataframe %>% function_name(conditions)`

Tidyverse

Cho dữ liệu pokemon.csv

```
pokemon <- read.csv('pokemon.csv')
```

```
head(pokemon)
```

X.	Name	Type.1	Type.2	Total	HP	Attack	Defense	Sp..Atk	Sp..Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire		309	39	52	43	60	50	65	1	False
5	Charmeleon	Fire		405	58	64	58	80	65	80	1	False

Tidyverse

❑select(): chọn cột

select(dataframe, danh_sách_tên_or_index_cột)

- Ví dụ:

```
pokemon_sub = select(pokemon, c(2, 5, 7, 8))
```

```
head(pokemon_sub)
```

Name	Total	Attack	Defense
Bulbasaur	318	49	49
Ivysaur	405	62	63
Venusaur	525	82	83
VenusaurMega Venusaur	625	100	123
Charmander	309	52	43
Charmeleon	405	64	58

dataframe %>% select(danh_sách_tên_or_index_cột)

- Ví dụ:

```
pokemon_sub = pokemon %>% select(Name, Total, Attack, Defense)
```

```
head(pokemon_sub)
```

Name	Total	Attack	Defense
Bulbasaur	318	49	49
Ivysaur	405	62	63
Venusaur	525	82	83
VenusaurMega Venusaur	625	100	123
Charmander	309	52	43
Charmeleon	405	64	58

❑ select(): chọn cột có điều kiện

select_if(dataframe, conditions)

- Ví dụ:

```
pokemon_numbers = select_if(pokemon, is.numeric)
```

```
head(pokemon_numbers)
```

X.	Total	HP	Attack	Defense	Sp..Atk	Sp..Def	Speed	Generation
1	318	45	49	49	65	65	45	1
2	405	60	62	63	80	80	60	1
3	525	80	82	83	100	100	80	1
3	625	80	100	123	122	120	80	1
4	309	39	52	43	60	50	65	1
5	405	58	64	58	80	65	80	1

❑filter(): chọn dòng

filter(dataframe, conditions)

hoặc **dataframe %>% filter(conditions)**

- Ví dụ:

```
pokemon_500_600 = filter(pokemon_sub, Total >=500, Total <=600)
head(pokemon_500_600)
```

Name	Total	Attack	Defense
Venusaur	525	82	83
Charizard	534	84	78
Blastoise	530	83	100
PidgeotMega Pidgeot	579	80	80
Nidoqueen	505	92	87
Nidoking	505	102	77

❑mutate(): tạo cột mới

mutate(dataframe, conditions)

hoặc **dataframe %>% mutate(conditions)**

- Ví dụ:

```
# Compare: if Attack > Defense => =1 else =0
pokemon_sub = mutate(pokemon_sub, Compare = ifelse(Attack > Defense , 1, 0))

head(pokemon_sub)
```

Name	Total	Attack	Defense	Speed	Compare
Bulbasaur	318	49	49	45	0
Ivysaur	405	62	63	60	0
Venusaur	525	82	83	80	0
VenusaurMega Venusaur	625	100	123	80	0
Charmander	309	52	43	65	1
Charmeleon	405	64	58	80	1

❑arrange(): sắp xếp dữ liệu

arrange(dataframe, conditions)

hoặc **dataframe %>% arrange(conditions)**

- Ví dụ:

```
head(arrange(pokemon_sub, Total))
```

Name	Total	Attack	Defense	Speed	Compare
Sunkern	180	30	30	30	0
Azurill	190	20	40	20	0
Kricketot	194	25	41	25	0
Caterpie	195	30	35	45	0
Weedle	195	35	30	50	1
Wurmple	195	45	35	20	1


```
head(arrange(pokemon_sub, desc(Total)))
```

Name	Total	Attack	Defense	Speed	Compare
MewtwoMega Mewtwo X	780	190	100	130	1
MewtwoMega Mewtwo Y	780	150	70	140	1
RayquazaMega Rayquaza	780	180	100	115	1
KyogrePrimal Kyogre	770	150	90	90	1
GroudonPrimal Groudon	770	180	160	90	1
Arceus	720	120	120	120	0

```
head(arrange(pokemon_sub, desc(Total), desc(Attack)))
```

Name	Total	Attack	Defense	Speed	Compare
MewtwoMega Mewtwo X	780	190	100	130	1
RayquazaMega Rayquaza	780	180	100	115	1
MewtwoMega Mewtwo Y	780	150	70	140	1
GroudonPrimal Groudon	770	180	160	90	1
KyogrePrimal Kyogre	770	150	90	90	1
Arceus	720	120	120	120	0

Tidyverse

- summarize(), group_by(): thống kê dữ liệu theo nhóm

  nhóm = group_by(dataframe, danh_sách_cột nhóm)

  summarize(nhóm, conditions)

Tidyverse

```
group = group_by(pokemon, Type.1)
summary = summarize(group,
                    count = n(),
                    mean.Attack = mean(Attack, na.rm = T),
                    median.Attack = median(Attack, na.rm = T),
                    mean.Defense = mean(Defense, na.rm = T),
                    median.Defense = median(Defense, na.rm = T)
                    )
```

```
summary
```

Type.1	count	mean.Attack	median.Attack	mean.Defense	median.Defense
Bug	69	70.97101	65.0	70.72464	60.0
Dark	31	88.38710	88.0	70.22581	70.0
Dragon	32	112.12500	113.5	86.37500	90.0
Electric	44	69.09091	65.0	66.29545	65.0
Fairy	17	61.52941	52.0	65.70588	66.0

...

❑Count(): đếm tần số xuất hiện

count(dataframe, columns)

Hoặc **dataframe %>% count(columns)**

- Ví dụ:

```
count(pokemon, Type.1)
```

Type.1	n
Bug	69
Dark	31
Dragon	32
Electric	44
Fairy	17
Fighting	27
...	

❑Sample_n(): chọn n mẫu ngẫu nhiên

sample_n(dataframe, n)

Hoặc **dataframe %>% sample_n(n)**

- Ví dụ:

```
pokemon %>% sample_n(5)
```

X.	Name	Type.1	Type.2	Total	HP	Attack	Defense	Sp..Atk	Sp..Def	Speed	Generation	Legendary
55	Golduck	Water		500	80	82	78	95	80	85	1	False
402	Kricketune	Bug		384	77	85	51	55	51	65	4	False
376	Metagross	Steel	Psychic	600	80	135	130	95	90	70	3	False
525	Boldore	Rock		390	70	105	105	50	40	20	5	False
294	Loudred	Normal		360	84	71	43	71	43	48	3	False

❑Sample_frac(): chọn n% mẫu ngẫu nhiên

sample_frac(dataframe, n)

Hoặc **dataframe %>% sample_frac(n)**

- Ví dụ:

```
sub_5_per = pokemon %>% sample_frac(0.05) # 5 percentages

dim(sub_5_per)
```

```
40  13
```

X.	Name	Type.1	Type.2	Total	HP	Attack	Defense	Sp..Atk	Sp..Def	Speed	Generation	Legendary
342	Crawdaunt	Water	Dark	468	63	120	85	90	55	55	3	False
259	Marshtomp	Water	Ground	405	70	85	70	60	70	50	3	False
...												

Chapter 8: Data Frame và Tidyverse

Exercise 1: Tạo dataframe

- Tạo dataframe như hình

	Age	Height	Weight	Sex
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

- In các level của Sex
- Tạo dataframe mới như hình sau (có row.names là Name giống data frame trên):

	Working
Alex	Yes
Lilly	No
Mark	No
Oliver	Yes
Martha	Yes
Lucas	No
Caroline	Yes

- Tạo data frame mới chứa 2 data frame trên

	Age	Height	Weight	Sex	Working
Alex	25	177	57	F	Yes
Lilly	31	163	69	F	No
Mark	23	190	83	M	No
Oliver	52	179	75	M	Yes
Martha	76	163	70	F	Yes
Lucas	49	183	83	M	No
Caroline	26	164	53	F	Yes

- Cho biết data frame mới này có bao nhiêu dòng và bao nhiêu cột?
- Cho biết kiểu dữ liệu của từng cột

Exercise 2: Tạo và làm việc với dataframe

- Cho 3 vector: a <- (floor(runif(10, -10, 10))), b <- letters[4:13], c <- c("yes","no","no","no","no","yes","no","yes","yes","no")
- Tạo data frame từ 3 vector trên, in data frame

```
    a b   c
1  -7 d yes
2  -8 e  no
3  -4 f  no
4  -9 g  no
5   5 h  no
6   0 i yes
7  -7 j  no
8  -8 k yes
9   8 l yes
10 -2 m  no
```

- In data frame với thứ tự giá trị được sắp tăng dần trong cột a

```
    a b   c
4  -9 g  no
2  -8 e  no
8  -8 k yes
1  -7 d yes
7  -7 j  no
3  -4 f  no
10 -2 m  no
6   0 i yes
5   5 h  no
9   8 l yes
```

- Cho ma trận: matrix.data <- matrix(1:40, nrow = 10, ncol = 4)
- Tạo dataframe từ ma trận này
- Đặt tên cho cột và dòng của ma trận như hình dưới và in kết quả:

	variable_1	variable_2	variable_3	variable_4
id_1	1	11	21	31
id_2	2	12	22	32
id_3	3	13	23	33
id_4	4	14	24	34
id_5	5	15	25	35
id_6	6	16	26	36
id_7	7	17	27	37
id_8	8	18	28	38
id_9	9	19	29	39
id_10	10	20	30	40

Exercise 3: Làm việc với dataframe

- Sử dụng dữ liệu cho sẵn của R là state.x77
- In dữ liệu
- Cho biết kiểu của dữ liệu?
- Kiểu này có phải là dataframe không? Nếu không thì chuyển dữ liệu này thành dataframe
- Trong dataframe trên, cho biết có bao nhiêu state có income <4300. Đó là những state nào?
- Cho biết state nào có income cao nhất và là bao nhiêu?
- Cho biết state nào có Life.Exp cao nhất và là bao nhiêu?
- Cho biết state nào có Life.Exp thấp nhất và là bao nhiêu?

```
[1] "state has max income: Alaska 6315"
[1] "state has highest life.exp: Hawaii 73.6"
[1] "state has lowest life.exp: South Carolina 67.96"
```

Exercise 4: Làm việc với dataframe

- Tạo data frame từ các dữ liệu mà R cho sẵn như sau: state.abb, state.area, state.division, state.name, state.region. In head của dataframe này.

	state.abb	state.area	state.division	state.region
Alabama	AL	51609	East South Central	South
Alaska	AK	589757	Pacific	West
Arizona	AZ	113909	Mountain	West
Arkansas	AR	53104	West South Central	South
California	CA	158693	Pacific	West
Colorado	CO	104247	Mountain	West

- Đổi tên cho tất cả các cột trong dataframe với tên chỉ chứa 3 ký tự sau dấu . của các tên cột đang có. In head dataframe sau khi đổi tên.

	abb	are	div	reg
Alabama	AL	51609	East South Central	South
Alaska	AK	589757	Pacific	West
Arizona	AZ	113909	Mountain	West
Arkansas	AR	53104	West South Central	South
California	CA	158693	Pacific	West
Colorado	CO	104247	Mountain	West

- Tạo data frame mới chứa state.x77 và data frame vừa tạo. In head của data frame này.

	Population	Income	Illiteracy	Life Exp	Murder	HS Grad	Frost	Area	abb	are	div	reg
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708	AL	51609	East South Central	South
Alaska	365	6315	1.5	69.31	11.3	66.7	152	566432	AK	589757	Pacific	West
Arizona	2212	4530	1.8	70.55	7.8	58.1	15	113417	AZ	113909	Mountain	West
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945	AR	53104	West South Central	South
California	21198	5114	1.1	71.71	10.3	62.6	20	156361	CA	158693	Pacific	West
Colorado	2541	4884	0.7	72.06	6.8	63.9	166	103766	CO	104247	Mountain	West

- Loại bỏ các cột div, Life Exp, HS Grad, Frost, abb, và are. In head của dataframe sau khi loại bỏ các cột.

	Population	Income	Illiteracy	Murder	Area	reg
Alabama	3615	3624	2.1	15.1	50708	South
Alaska	365	6315	1.5	11.3	566432	West
Arizona	2212	4530	1.8	7.8	113417	West
Arkansas	2110	3378	1.9	10.1	51945	South
California	21198	5114	1.1	10.3	156361	West
Colorado	2541	4884	0.7	6.8	103766	West

- Thêm một cột mới vào data frame với mô tả như sau: tên cột Illiteracy.Levels, giá trị của từng phần tử sẽ là Low nếu Illiteracy <1, là Some nếu Illiteracy <2, còn lại là High (gợi ý: dùng ifelse()). In head của dataframe này.

	Population	Income	Illiteracy	Murder	Area	reg	Illiteracy.Levels
Alabama	3615	3624	2.1	15.1	50708	South	High
Alaska	365	6315	1.5	11.3	566432	West	Some
Arizona	2212	4530	1.8	7.8	113417	West	Some
Arkansas	2110	3378	1.9	10.1	51945	South	Some
California	21198	5114	1.1	10.3	156361	West	Some
Colorado	2541	4884	0.7	6.8	103766	West	Low

- Cho biết trong data frame có bao nhiêu region và đó là những region nào?

```
[1] "Number of Regions: 4 . There are: Northeast, South, North Central, We
```

Exercise 5: Sử dụng function trong package tidyverse

- Cho dữ liệu Obesity_data.csv. Đọc dữ liệu vào dataframe data. In head của data

	id	gender	height	weight	bmi	age	bmc	bmd	fat	lean	pcfat
1	1	F	150	49	21.8	53	1312	0.88	17802	28600	37.3
2	2	M	165	52	19.1	65	1309	0.84	8381	40229	16.8
3	3	F	157	57	23.1	64	1230	0.84	19221	36057	34.0
4	4	F	156	53	21.8	56	1171	0.80	17472	33094	33.8
5	5	M	160	51	19.9	54	1681	0.98	7336	40621	14.8
6	6	F	153	47	20.1	52	1358	0.91	14904	30068	32.2

- Tạo dataframe data1 chỉ chứa các cột id, gender, height, weight, bmi. In head của data1

	gender	height	weight	bmi	age
1	F	150	49	21.8	53
2	M	165	52	19.1	65
3	F	157	57	23.1	64
4	F	156	53	21.8	56
5	M	160	51	19.9	54
6	F	153	47	20.1	52

- Với data1, lọc ra những dòng có bmi >=18.5 và <=24.9 và đưa vào dataframe data2

	gender	height	weight	bmi	age
1	F	150	49	21.8	53
2	M	165	52	19.1	65
3	F	157	57	23.1	64
4	F	156	53	21.8	56
5	M	160	51	19.9	54
6	F	153	47	20.1	52

- Với data1, tạo ta biến mới height_m = height/100

gender	height	weight	bmi	age	height_m
F	150	49	21.8	53	1.50
M	165	52	19.1	65	1.65
F	157	57	23.1	64	1.57
F	156	53	21.8	56	1.56
M	160	51	19.9	54	1.60
F	153	47	20.1	52	1.53

- Với data1, sắp tăng dần theo bmi

gender	height	weight	bmi	age	height_m
M	162	38	14.5	55	1.62
F	162	40	15.2	54	1.62
F	151	35	15.4	33	1.51
F	155	37	15.4	44	1.55
F	150	35	15.6	24	1.50
M	169	45	15.8	50	1.69

- Với data1, Tính giá trị trung bình của height, weight theo gender. Tạo ra các biến mới chứa giá trị trung bình là mean.height, mean.weight

gender	count	mean.height	mean.weight
F	862	153.2912	52.31090
M	355	165.0592	62.02254

- Với data1, đếm số lượng theo gender và age
- Chọn 10 mẫu ngẫu nhiên từ data1 và đưa vào data3
- Chọn 1% mẫu ngẫu nhiên từ data1 và đưa vào data4

Gợi ý

Exercise 1: Tạo dataframe

```
In [1]: Name <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas", "Caroline")
        Age <- c(25, 31, 23, 52, 76, 49, 26)
        Height <- c(177, 163, 190, 179, 163, 183, 164)
        Weight <- c(57, 69, 83, 75, 70, 83, 53)
        Sex <- as.factor(c("F", "F", "M", "M", "F", "M", "F"))
        df <- data.frame (row.names = Name, Age, Height, Weight, Sex)
```

```
In [5]: print(df)

                Age Height Weight Sex
        Alex     25    177     57   F
        Lilly    31    163     69   F
        Mark     23    190     83   M
        Oliver   52    179     75   M
        Martha   76    163     70   F
        Lucas    49    183     83   M
        Caroline 26    164     53   F
```

```
In [3]: print(paste("Levels of Sex:",toString(levels(df$Sex))))

        [1] "Levels of Sex: F, M"
```

In [6]:
```
Working <- c("Yes","No","No","Yes","Yes","No","Yes")
df2 <- data.frame(row.names = Name,Working) #Name has been already defined
print(df2)
```

```
         Working
Alex         Yes
Lilly         No
Mark          No
Oliver       Yes
Martha       Yes
Lucas         No
Caroline     Yes
```

In [7]:
```
# tao data frame moi chua thong tin cua 2 data frame tren
df <- cbind(df,df2)
print("Combining 2 data frame:")
print(df)
```

```
[1] "Combining 2 data frame:"
         Age Height Weight Sex Working
Alex      25    177     57   F     Yes
Lilly     31    163     69   F      No
Mark      23    190     83   M      No
Oliver    52    179     75   M     Yes
Martha    76    163     70   F     Yes
Lucas     49    183     83   M      No
Caroline  26    164     53   F     Yes
```

In [8]:
```
#co bao nhieu dong va bao nhieu cot trong data frame nay?
print(paste("Dong:", dim(df)[1], ", Cot:", dim(df)[2]))
# cho biet kieu du lieu cua tung cot
print(str(df))
```

```
[1] "Dong: 7 , Cot: 5"
'data.frame':   7 obs. of  5 variables:
 $ Age    : num  25 31 23 52 76 49 26
 $ Height : num  177 163 190 179 163 183 164
 $ Weight : num  57 69 83 75 70 83 53
 $ Sex    : Factor w/ 2 levels "F","M": 1 1 2 2 1 2 1
 $ Working: Factor w/ 2 levels "No","Yes": 2 1 1 2 2 1 2
NULL
```

Exercise 2: Tạo và làm việc với dataframe

In [9]:
```
a <- (floor(runif(10, -10, 10)))
b <- letters[4:13]
c <- c("yes","no","no","no","no","yes","no","yes","yes","no")
#tao data frame
df3 <- data.frame(a,b,c)
print("Data frame:")
print(df3)
```

```
[1] "Data frame:"
    a b   c
1  -7 d yes
2  -8 e  no
3  -4 f  no
4  -9 g  no
5   5 h  no
6   0 i yes
7  -7 j  no
8  -8 k yes
9   8 l yes
10 -2 m  no
```

In [10]:
```
#in data frame voi du lieu sap xep tang dan o cot a
print("Data frame with column a in order")
print(df3[with (df3, order(a)),] )
```

```
[1] "Data frame with column a in order"
    a b   c
4  -9 g  no
2  -8 e  no
8  -8 k yes
1  -7 d yes
7  -7 j  no
3  -4 f  no
10 -2 m  no
6   0 i yes
5   5 h  no
9   8 l yes
```

In [11]:
```
#cho ma tran
matrix.data <- matrix(1:40, nrow = 10, ncol = 4)
#tao data frame tu ma tran
print("Data frame form matrix:")
df <- as.data.frame(matrix.data)
#dat ten cot va dong cho ma tran nhu sau
colnames(df) <- sub(" ","",paste("variable_", 1:ncol(df)))
rownames(df) <- sub(" ", "",paste("id_", 1:nrow(df)))
print(df)
```

```
[1] "Data frame form matrix:"
      variable_1 variable_2 variable_3 variable_4
id_1           1         11         21         31
id_2           2         12         22         32
id_3           3         13         23         33
id_4           4         14         24         34
id_5           5         15         25         35
id_6           6         16         26         36
id_7           7         17         27         37
id_8           8         18         28         38
id_9           9         19         29         39
id_10         10         20         30         40
```

Exercise 3: Làm việc với dataframe

In [15]:
```
print(head(state.x77))
print(paste("Data type:",class(state.x77)))
```

```
           Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
[1] "Data type: matrix"
```

In [16]:
```
df <- data.frame(state.x77)
print(head(df))
```

```
           Population Income Illiteracy Life.Exp Murder HS.Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

In [18]:
```
# Find out how many states have an income of less than 4300.
print(paste("Number of states income < 4300:", nrow(df[df$Income < 4300,])))
# what are they?
df[df$Income < 4300,]
```

```
[1] "Number of states income < 4300: 20"
```

	Population	Income	Illiteracy	Life.Exp	Murder	HS.Grad	Frost	Area
Alabama	3615	3624	2.1	69.05	15.1	41.3	20	50708
Arkansas	2110	3378	1.9	70.66	10.1	39.9	65	51945
Georgia	4931	4091	2.0	68.54	13.9	40.6	60	58073
Idaho	813	4119	0.6	71.87	5.3	59.5	126	82677
Kentucky	3387	3712	1.6	70.10	10.6	38.5	95	39650
Louisiana	3806	3545	2.8	68.76	13.2	42.2	12	44930
Maine	1058	3694	0.7	70.39	2.7	54.7	161	30920
Mississippi	2341	3098	2.4	68.09	12.5	41.0	50	47296
Missouri	4767	4254	0.8	70.69	9.3	48.8	108	68995
New Hampshire	812	4281	0.7	71.23	3.3	57.6	174	9027
New Mexico	1144	3601	2.2	70.32	9.7	55.2	120	121412
North Carolina	5441	3875	1.8	69.21	11.1	38.5	80	48798
Oklahoma	2715	3983	1.1	71.42	6.4	51.6	82	68782
South Carolina	2816	3635	2.3	67.96	11.6	37.8	65	30225
South Dakota	681	4167	0.5	72.08	1.7	53.3	172	75955
Tennessee	4173	3821	1.7	70.11	11.0	41.8	70	41328
Texas	12237	4188	2.2	70.90	12.2	47.4	35	262134
Utah	1203	4022	0.6	72.90	4.5	67.3	137	82096
Vermont	472	3907	0.6	71.64	5.5	57.1	168	9267
West Virginia	1799	3617	1.4	69.48	6.7	41.6	100	24070

In [19]:
```
# Find out which is the state with the highest income.
ma <-df[which.max(df$Income),][2]
print(paste("state has max income:",
            row.names(df[which.max(df$Income),]),
            toString(ma)))
# Find out which is the state with the highest Life.Exp.
ma <-df[which.max(df$Life.Exp),][4]
print(paste("state has highest life.exp:",
            row.names(df[which.max(df$Life.Exp),]),
            toString(ma)))
# Find out which is the state with the lowest Life.Exp.
ma <-df[which.min(df$Life.Exp),][4]
print(paste("state has lowest life.exp:",
            row.names(df[which.min(df$Life.Exp),]),
            toString(ma)))
```

```
[1] "state has max income: Alaska 6315"
[1] "state has highest life.exp: Hawaii 73.6"
[1] "state has lowest life.exp: South Carolina 67.96"
```

Exercise 4: Làm việc với dataframe

In [30]:
```
# tao data frame tu state.abb, state.area, state.division, state.name, state.reg
# row names la ten cua states.
df <- data.frame(state.abb, state.area, state.division, state.region,
                 row.names = state.name)
print("Data frame:")
print(head(df))
```

```
[1] "Data frame:"
           state.abb state.area     state.division state.region
Alabama           AL      51609 East South Central        South
Alaska            AK     589757            Pacific         West
Arizona           AZ     113909           Mountain         West
Arkansas          AR      53104 West South Central        South
California        CA     158693            Pacific         West
Colorado          CO     104247           Mountain         West
```

In [31]:
```
# doi ten tat cac cac cot chi chua 3 ky tu sau dau .
colnames(df) <- substr(colnames(df), 7, 9)
# sau khi doi ten
print("After rename:")
print(head(df))
```

```
[1] "After rename:"
           abb    are                div   reg
Alabama     AL  51609 East South Central South
Alaska      AK 589757            Pacific  West
Arizona     AZ 113909           Mountain  West
Arkansas    AR  53104 West South Central South
California  CA 158693            Pacific  West
Colorado    CO 104247           Mountain  West
```

In [32]:
```
# tao data frame chua state.x77 va du lieu moi tao
df_new <- cbind(state.x77, df)
print("New data frame:")
print(head(df_new))
```

```
[1] "New data frame:"
           Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
California      21198   5114        1.1    71.71   10.3    62.6    20 156361
Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
           abb    are                div   reg
Alabama     AL  51609 East South Central South
Alaska      AK 589757            Pacific  West
Arizona     AZ 113909           Mountain  West
Arkansas    AR  53104 West South Central South
California  CA 158693            Pacific  West
Colorado    CO 104247           Mountain  West
```

In [33]:
```
# loai bo cot div
df_new$div <- NULL
# tiep tuc loai bo cac cot Life Exp, HS Grad, Frost, abb, va are
df_new <- subset(df_new,select = -c(4, 6, 7, 9, 10))

print("After drop div, Life Exp, HS Grad, Frost, abb, and are:")
print(head(df_new))
```

```
[1] "After drop div, Life Exp, HS Grad, Frost, abb, and are:"
           Population Income Illiteracy Murder   Area   reg
Alabama          3615   3624        2.1   15.1  50708 South
Alaska            365   6315        1.5   11.3 566432  West
Arizona          2212   4530        1.8    7.8 113417  West
Arkansas         2110   3378        1.9   10.1  51945 South
California      21198   5114        1.1   10.3 156361  West
Colorado         2541   4884        0.7    6.8 103766  West
```

In [34]:
```
# them cot phan loai mu chu Illiteracy.Levels voi mo ta sau: <1 Low, <2 So
# con lai la High
df_new$Illiteracy.Levels <- ifelse(df_new$Illiteracy >= 0 &
                                   df_new$Illiteracy < 1,
                                   "Low",
                                   ifelse(df_new$Illiteracy >= 1
                                          & df_new$Illiteracy < 2,
                                          "Some",
                                          "High"))
print("After insert Illiteracy.Levels:")
print(head(df_new))
```

```
[1] "After insert Illiteracy.Levels:"
           Population Income Illiteracy Murder   Area    reg Illiteracy.Levels
Alabama          3615   3624        2.1   15.1  50708 South              High
Alaska            365   6315        1.5   11.3 566432  West              Some
Arizona          2212   4530        1.8    7.8 113417  West              Some
Arkansas         2110   3378        1.9   10.1  51945 South              Some
California      21198   5114        1.1   10.3 156361  West              Some
Colorado         2541   4884        0.7    6.8 103766  West               Low
```

In [35]:
```
# cho biet co bao nhieu vung va la nhung vung nao?
print(paste("Number of Regions:", nlevels(df_new$reg), ". There are:",
            toString(levels(df_new$reg))))
```

```
[1] "Number of Regions: 4 . There are: Northeast, South, North Central, West"
```

Exercise 5: Sử dụng function trong package tidyverse

In [36]:
```
library(tidyverse)
```

```
Loading tidyverse: ggplot2
Loading tidyverse: tibble
Loading tidyverse: tidyr
Loading tidyverse: readr
Loading tidyverse: purrr
Loading tidyverse: dplyr

Conflicts with tidy packages ----------------------------------------------
-

filter(): dplyr, stats
lag():    dplyr, stats
```

In [37]:
```
data <- read.csv('Obesity_data.csv')
print(head(data))
```

```
  id gender height weight  bmi age  bmc  bmd   fat  lean pcfat
1  1      F    150     49 21.8  53 1312 0.88 17802 28600  37.3
2  2      M    165     52 19.1  65 1309 0.84  8381 40229  16.8
3  3      F    157     57 23.1  64 1230 0.84 19221 36057  34.0
4  4      F    156     53 21.8  56 1171 0.80 17472 33094  33.8
5  5      M    160     51 19.9  54 1681 0.98  7336 40621  14.8
6  6      F    153     47 20.1  52 1358 0.91 14904 30068  32.2
```

In [39]:
```
# c1
data1 = data %>% select(c(2,3,4,5,6))
print(head(data1))
```

```
  gender height weight  bmi age
1      F    150     49 21.8  53
2      M    165     52 19.1  65
3      F    157     57 23.1  64
4      F    156     53 21.8  56
5      M    160     51 19.9  54
6      F    153     47 20.1  52
```

In [51]:
```
# c2
data11 = data %>% select(gender, height, weight, bmi, age)
head(data11)
```

gender	height	weight	bmi	age
F	150	49	21.8	53
M	165	52	19.1	65
F	157	57	23.1	64
F	156	53	21.8	56
M	160	51	19.9	54
F	153	47	20.1	52

In [52]:
```
data2 = data1 %>% filter(bmi >=18.5, bmi <=24.9)
```

In [53]:
```
print(head(data2))
```

```
  gender height weight  bmi age
1      F    150     49 21.8  53
2      M    165     52 19.1  65
3      F    157     57 23.1  64
4      F    156     53 21.8  56
5      M    160     51 19.9  54
6      F    153     47 20.1  52
```

In [54]:
```
print(paste("Rows have bmi >=18.5 and <=24.9: ", nrow(data2)))
```

```
[1] "Rows have bmi >=18.5 and <=24.9:  865"
```

```
In [55]: data1 = mutate(data1, height_m = height/100)
         head(data1)
```

gender	height	weight	bmi	age	height_m
F	150	49	21.8	53	1.50
M	165	52	19.1	65	1.65
F	157	57	23.1	64	1.57
F	156	53	21.8	56	1.56
M	160	51	19.9	54	1.60
F	153	47	20.1	52	1.53

```
In [56]: head(arrange(data1, bmi))
```

gender	height	weight	bmi	age	height_m
M	162	38	14.5	55	1.62
F	162	40	15.2	54	1.62
F	151	35	15.4	33	1.51
F	155	37	15.4	44	1.55
F	150	35	15.6	24	1.50
M	169	45	15.8	50	1.69

```
In [59]: group = group_by(data1, gender)
         summary = summarize(group,
                             count = n(),
                             mean.height = mean(height, na.rm = T),
                             mean.weight = mean(weight, na.rm = T)
                             )
```

```
In [60]: summary
```

gender	count	mean.height	mean.weight
F	862	153.2912	52.31090
M	355	165.0592	62.02254

```
In [62]: count(data1, gender)
```

gender	n
F	862
M	355

```
In [73]: group_gender_age = count(data1, gender, age)
         head(group_gender_age)
```

gender	age	n
F	14	4
F	16	2
F	18	10
F	19	27
F	20	13
F	21	8

```
In [74]: tail(group_gender_age)
```

gender	age	n
M	82	1
M	83	1
M	84	3
M	85	1
M	87	1
M	88	1

```
In [66]: data3 = data1 %>% sample_n(10)
```

```
In [67]: data3
```

	gender	height	weight	bmi	age	height_m
1075	F	152	55	23.8	52	1.52
868	M	171	71	24.3	50	1.71
369	F	158	51	20.4	46	1.58
758	F	136	48	26.0	72	1.36
1198	M	150	45	20.0	59	1.50
318	M	166	52	18.9	26	1.66
789	F	145	45	21.4	80	1.45
672	F	159	50	19.8	22	1.59
57	F	154	43	18.1	48	1.54
181	M	160	54	21.1	65	1.60

```
In [70]: data4 = data %>% sample_frac(0.01) # 1 percentages
```

In [71]: data4

	id	gender	height	weight	bmi	age	bmc	bmd	fat	lean	pcfat
713	719	F	150	46	20.4	30	2055	1.26	15302	28551	33.3
1080	1090	F	139	36	18.6	79	1111	0.89	12075	23204	33.2
288	291	F	152	42	18.2	55	1491	0.96	8986	30893	21.7
76	76	F	151	57	25.0	54	1345	0.85	24067	30986	42.7
787	794	F	155	50	20.8	45	1718	1.05	16897	30004	34.8
842	849	F	161	56	21.6	23	1675	0.96	19429	34252	35.1
147	148	M	167	76	27.3	40	2184	1.08	25043	47546	33.5
776	782	F	155	59	24.6	46	1770	1.12	16907	30175	34.6
709	715	M	174	58	19.2	24	2193	1.12	10680	42376	19.3
628	634	M	183	95	28.4	18	2912	1.21	29944	58217	32.9
129	130	F	151	57	25.0	43	1689	1.02	21875	34088	37.9
91	92	F	151	51	22.4	61	1121	0.78	20464	33941	36.6