

R basic

Jan 27 2022

Write on R version 4.1

Ch.1 변수와 데이터의 형태

R은 "값"을 "객체"에 대입하여 코드에 활용함.

대입을 하기 위한 방법으로는 `<-` 혹은 `=` 를 활용할 수 있음.

다만. `=` 를 사용할 경우 R에서 다양하게 활용되는 수식과 혼동 될 수 있기 때문에 `<-` 를 주로 사용함.

아래는 객체 값을 대입하여 그 결과를 확인한 예제임.

첫번째와 같이 a 에 아무런 값을 입력하지 않고 a를 출력하였을때는 객체 "a" 를 찾을 수 없지만 a 와 b 에 각각 다른 방법으로 대입하게 되면, 값을 얻어낼 수 있음.

In [2]:

```
a
```

```
Error in eval(expr, envir, enclos): 객체 'a'를 찾을 수 없습니다
Traceback:
```

In [3]:

```
a <- 2
a
```

2

In [4]:

```
b = 2
b
```

2

사칙연산, 비교연산, 논리연산자

객체를 다양한 방식으로 연산하기 위해 기본적인 연산자를 활용할 수 있다.

연산자의 종류는 아래와 같다.

특히 논리 연산자는 다양하게 활용된다.

사칙연산: `+` `-` `*` `/`

비교 연산자: `>` `<` `>=` `<=`

논리연산자 1: `==` (equal, 수학의 `=`, 대입문을 `=` 을 활용하기 때문에 equal은 `==` 을 사용), `!=` (같지 않다)

논리 연산자 2: `&` (AND) | (OR)

```
In [5]: #논리 연산자 사용 예시
a <- 2
b <- 2
a == b
```

TRUE

```
In [6]: #논리 연산자 사용 예시
a <- 2
b <- 3
a != b

a >= b
a <= b
```

TRUE

FALSE

TRUE

자료의 형태 (수식 연산과 문자 연산)

프로그래밍 언어에서의 데이터는 "숫자" 와 "문자" 가 있다.

숫자의 종류는 정수형(numeric) 과 실수형(double) 이 있으며, 문자의 종류는 string 과 character가 있다.

숫자는 사칙연산과 비교 연산이 가능하지만

문자는 논리 연산만 가능하다

```
In [7]: # 숫자의 연산과 문자의 연산 예시
a <- 1
b <- 1

a
b
a == b
a + b
```

1

1

TRUE

2

```
In [8]: # 숫자의 연산과 문자의 연산 예시
a <- "1"
b <- "1"

a
b
a == b
a + b
```

'1'

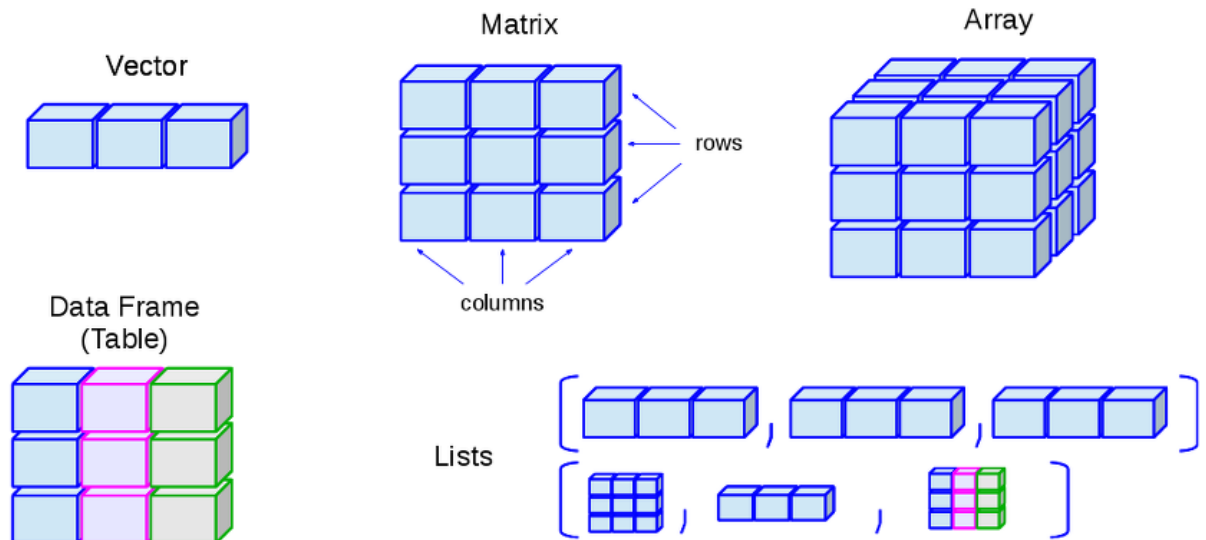
'1'

TRUE

Error in a + b: 이항연산자에 수치가 아닌 인수입니다

Traceback:

데이터의 형태



다양한 방법으로 벡터 만들기, 합치기, 조합하기

```
In [9]: vector_1 <- c(1,2,3,4,5)
vector_1
```

1 · 2 · 3 · 4 · 5

```
In [10]: a <- 1
b <- 2
c <- 2

vector_2 <- c(a,b,c,4,5)
vector_2

vector_3 <- c(a,b,4,5,c)
vector_3
```

1 · 2 · 2 · 4 · 5

1 · 2 · 4 · 5 · 2

```
In [11]: a <- "1"
          b <- "2"
          c <- "2"

          vector_4 <- c(a,b,c,4,5)
          vector_4
```

'1' '2' '2' '4' '5'

```
In [12]: class(vector_2)
          class(vector_3)
```

'numeric'

'numeric'

```
In [13]: class(vector_4)
```

'character'

```
In [14]: # 1부터 10까지 있는 벡터 생성
          x1 <- seq(1,10)

          # 1부터 10까지 2씩 증가하는 수열의 벡터 생성
          x2 <- seq(1,10,2)

          x1
          x2
```

1 2 3 4 5 6 7 8 9 10

1 3 5 7 9

```
In [15]: # 1이 10개 있는 벡터 생성
          rep(1,10)
          # 벡터 1과 2가 5번 반복되는 벡터 생성
          rep(c(1,2),5)
```

1 1 1 1 1 1 1 1 1 1

1 2 1 2 1 2 1 2 1 2

```
In [16]: wt <- rep("WT",5)
          ko <- rep("KO",5)

          # 벡터 합치기
          Conditions <- c(wt, ko)
          Conditions
```

'WT' 'WT' 'WT' 'WT' 'WT' 'KO' 'KO' 'KO' 'KO' 'KO'

TSV 파일 읽어들이어서 데이터프레임으로 불러오기

```
In [17]: # TSV 파일 읽어오는 방법 1

example <- read.table("./Example_data.txt", sep='\t')
head(example)
```

Warning message in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
"따옴표로 묶인 문자열내에 EOF가 있습니다"

Warning message in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
"읽어들인 항목의 개수가 열의 개수의 배수가 아닙니다"

A data.fi

	V1	V2	V3	V4	V5	V6	V7	
	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	<chr>	
1	Entry	Entry_name	Gene_name	Status	Accession	MCF7_1	MCF7_2	
2	Q09666	AHNAK_HUMAN	AHNAK PM227	reviewed	Q09666	4984.800507	4959.724695	48.
3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-3	18.44385675	20.20028963	17.
4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149	3202.244428	3335.804133	315
5	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-8	13.18007585	13.66558497	13.
6	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-4	12.86596234	12.27708927	12

```
In [18]: # TSV 파일 읽어오는 방법 2

example <- read.table("./Example_data.txt", sep='\t', header = T)
head(example)
```

```
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"다음표로 묶인 문자열내에 EOF가 있습니다"
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"읽어들인 항목의 개수가 열의 개수의 배수가 아닙니다"
```

A data.fr

	Entry	Entry_name	Gene_name	Status	Accession	MCF7_1	MCF7_2	
	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	
1	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	Q09666	4984.80051	4959.72469	488
2	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-3	18.44386	20.20029	1
3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149	3202.24443	3335.80413	3156
4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-8	13.18008	13.66558	13
5	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-4	12.86596	12.27709	1
6	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	P21333-2	3657.09665	4067.28070	377

```
In [19]: # TSV 파일 읽어오는 방법 3 row.names 안되는 이유!
```

```
example <- read.table("./Example_data.txt", sep = '\t', header = T, row.names =
head(example))
```

```
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"다음표로 묶인 문자열내에 EOF가 있습니다"
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"읽어들인 항목의 개수가 열의 개수의 배수가 아닙니다"
```

```
Error in read.table("./Example_data.txt", sep = "\t", header = T, row.names
= 1): duplicate 'row.names' are not allowed
Traceback:
```

```
1. read.table("./Example_data.txt", sep = "\t", header = T, row.names = 1)
2. stop("duplicate 'row.names' are not allowed")
```

```
In [26]: # TSV 파일 읽어오는 방법 3 - 이슈 해결
```

```
example <- read.table("./Example_data.txt", sep = '\t', header = T, row.names =
head(example))
```

```
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"다음표로 묶인 문자열내에 EOF가 있습니다"
Warning message in scan(file = file, what = what, sep = sep, quote = quote,
dec = dec, :
"읽어들인 항목의 개수가 열의 개수의 배수가 아닙니다"
```

A data.frame

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Q09666	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	4984.80051	4959.72469	4885.114
Q15149-3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	18.44386	20.20029	17.780
Q15149	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	3202.24443	3335.80413	3156.994
Q15149-8	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	13.18008	13.66558	13.508
Q15149-4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	12.86596	12.27709	12.162
P21333-2	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	3657.09665	4067.28070	3777.620

CSV 파일 읽어들이어서 데이터프레임으로 불러오기

```
In [27]: example <- read.csv <- read.csv("../Example_data.csv")
head(example)
```

A data.frame

	Entry	Entry_name	Gene_name	Status	Accession	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
1	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	Q09666	4984.80051	4959.72469	4885.114
2	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-3	18.44386	20.20029	17.780
3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149	3202.24443	3335.80413	3156.994
4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-8	13.18008	13.66558	13.508
5	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-4	12.86596	12.27709	12.162
6	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	P21333-2	3657.09665	4067.28070	3777.620

```
In [28]: example <- read.csv <- read.csv("../Example_data.csv", header = T)
head(example)
```

A data.frame

	Entry	Entry_name	Gene_name	Status	Accession	MCF7_1	MCF7_2	
	<chr>	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	
1	Q09666	AHNAK_HUMAN	AHNAK PM227	reviewed	Q09666	4984.80051	4959.72469	488
2	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-3	18.44386	20.20029	17.780
3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149	3202.24443	3335.80413	3156.994
4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-8	13.18008	13.66558	13.508
5	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	Q15149-4	12.86596	12.27709	12.162
6	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	P21333-2	3657.09665	4067.28070	3777.620

```
In [29]: example <- read.csv <- read.csv("../Example_data.csv", header = T, row.names =
head(example)
```

A data.frame

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Q09666	Q09666	AHNAK_HUMAN	AHNAK PM227	reviewed	4984.80051	4959.72469	4885.114
Q15149-3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	18.44386	20.20029	17.780
Q15149	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	3202.24443	3335.80413	3156.994
Q15149-8	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	13.18008	13.66558	13.508
Q15149-4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	12.86596	12.27709	12.162
P21333-2	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	3657.09665	4067.28070	3777.620

```
In [30]: class(example)
```

```
'data.frame'
```

Ch.2 데이터 프레임 핸들링

```
In [31]: head(example)
```


A data.frame

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Q09666	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	4984.80051	4959.72469	4885.1141
Q15149-3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	18.44386	20.20029	17.780
Q15149	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	3202.24443	3335.80413	3156.9941
Q15149-8	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	13.18008	13.66558	13.5081
Q15149-4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	12.86596	12.27709	12.1621
P21333-2	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	3657.09665	4067.28070	3777.6201

```
In [32]: # 데이터프레임의 크기 확인하기

# row 길이 보기
nrow(example)

# column 길이 보기
ncol(example)

# 행렬 보기
dim(example)
```

7755

13

7755 · 13

```
In [50]: # 데이터 프레임 열 조회하기

cat("조회하는 방법 1")
head(example[,1])

cat("조회하는 방법 1-2")
head(example[1,])

cat("조회하는 방법 1-3")
head(example[1])

cat("조회하는 방법 2")
head(example$Entry)
```

조회하는 방법 1

'Q09666' · 'Q15149' · 'Q15149' · 'Q15149' · 'Q15149' · 'P21333'

조회하는 방법 1-2

A data.frame: 1 × 1

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3	Doxi
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>	
Q09666	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	4984.801	4959.725	4885.115	

조회하는 방법 1-3

A data.frame: 6 × 1

Entry
<chr>
Q09666 Q09666
Q15149-3 Q15149
Q15149 Q15149
Q15149-8 Q15149
Q15149-4 Q15149
P21333-2 P21333

조회하는 방법 2

'Q09666' · 'Q15149' · 'Q15149' · 'Q15149' · 'Q15149' · 'P21333'

In [37]:

```
# 필요한 데이터만 추출하기1
```

```
expression <- example[,5:10]
head(expression)
```

A data.frame: 6 × 6

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.019521
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.50858
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.91327
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.932638

In [54]: `head(example)`

A data.frame

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Q09666	Q09666	AHNK_HUMAN	AHNAK PM227	reviewed	4984.80051	4959.72469	4885.11485
Q15149-3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	18.44386	20.20029	17.78025
Q15149	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	3202.24443	3335.80413	3156.99450
Q15149-8	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	13.18008	13.66558	13.50839
Q15149-4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	12.86596	12.27709	12.16204
P21333-2	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	3657.09665	4067.28070	3777.62035

In [57]: `# 필요한 데이터만 추출하기2`

```
fold <- example[,12:13]
head(fold)
```

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.25644317
Q15149-3	0.340135415	0.08117139
Q15149	0.000319444	0.91329164
Q15149-8	0.006280590	-0.37684673
Q15149-4	0.000223258	1.38131662
P21333-2	0.146827666	-0.11675658

In [61]:

```
#데이터프레임에 신규 열 추가하기
head(expression)
cat("nrow is", nrow(expression))
head(example$Gene_name)
cat("length is", length(example$Gene_name))

expression$Gene_name <- example$Gene_name
head(expression)
```

A data.frame: 6 × 7

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.019521
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.50858
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.91327
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.932638

nrow is 7755

'AHNAK PM227' · 'PLEC PLEC1' · 'PLEC PLEC1' · 'PLEC PLEC1' · 'PLEC PLEC1' ·
'FLNA FLN FLN1'

length is 7755

A data.frame: 6 × 7

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.019521
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.50858
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.91327
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.932638

In [76]:

```
# 데이터 프레임 합치기
```

```
# 행 개수가 똑같은 두개의 데이터프레임을 양옆으로 합치려면: cbind()
```

```
# 열 개수가 똑같은 두개의 데이터프레임을 위아래로 합치려면: rbind()
```

```
merge_table<- cbind(expression,fold)
```

```
head(merge_table)
```

A data.frame: 6 × 8

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.019521
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.50858
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.91327
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.932638

subset 함수를 활용한 데이터 프레임 추출하기

Column 이름으로 추출하기

In [62]: `head(example)`

A data.frame

	Entry	Entry_name	Gene_name	Status	MCF7_1	MCF7_2	MCF7_3
	<chr>	<chr>	<chr>	<chr>	<dbl>	<dbl>	<dbl>
Q09666	Q09666	AHNK_HUMAN	AHNK PM27	reviewed	4984.80051	4959.72469	4885.11485
Q15149-3	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	18.44386	20.20029	17.78025
Q15149	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	3202.24443	3335.80413	3156.99450
Q15149-8	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	13.18008	13.66558	13.50839
Q15149-4	Q15149	PLEC_HUMAN	PLEC PLEC1	reviewed	12.86596	12.27709	12.16204
P21333-2	P21333	FLNA_HUMAN	FLNA FLN FLN1	reviewed	3657.09665	4067.28070	3777.62035

In [66]: `expression <- subset(example,select=c("MCF7_1","MCF7_2","MCF7_3","Doxorubicin_1","Doxorubicin_2","Doxorubicin_3"))`
`head(expression)`

A data.frame: 6 × 6

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.019521
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.508587
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.913274
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.932638

```
In [67]: fold <- subset(example,select=c("P.adjust","log2FC"))
         head(fold)
```

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.25644317
Q15149-3	0.340135415	0.08117139
Q15149	0.000319444	0.91329164
Q15149-8	0.006280590	-0.37684673
Q15149-4	0.000223258	1.38131662
P21333-2	0.146827666	-0.11675658

조건에 맞는 열로 추출하기

```
In [68]: head(fold)
```

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.25644317
Q15149-3	0.340135415	0.08117139
Q15149	0.000319444	0.91329164
Q15149-8	0.006280590	-0.37684673
Q15149-4	0.000223258	1.38131662
P21333-2	0.146827666	-0.11675658

```
In [69]: # P-value로 데이터 자르기
fold.pval.05 <- subset(fold, fold$P.adjust <= 0.05)
head(fold.pval.05)
nrow(fold.pval.05)
```

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.2564432
Q15149	0.000319444	0.9132916
Q15149-8	0.006280590	-0.3768467
Q15149-4	0.000223258	1.3813166
Q14204	0.011834786	0.4533949
P78527	0.000873574	0.5018461

5623

```
In [73]: # P-value & Fold change로 데이터 자르기
fold.pval.05.Fold1.2 <- subset(fold, fold$P.adjust <= 0.05 & (fold$log2FC >= 1.2))
head(fold.pval.05.Fold1.2)
nrow(fold.pval.05.Fold1.2)
```


A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.2564432
Q15149	0.000319444	0.9132916
Q15149-8	0.006280590	-0.3768467
Q15149-4	0.000223258	1.3813166
Q14204	0.011834786	0.4533949
P78527	0.000873574	0.5018461

4741

Ch.3 Library 활용하여 데이터 프레임 핸들링 하기

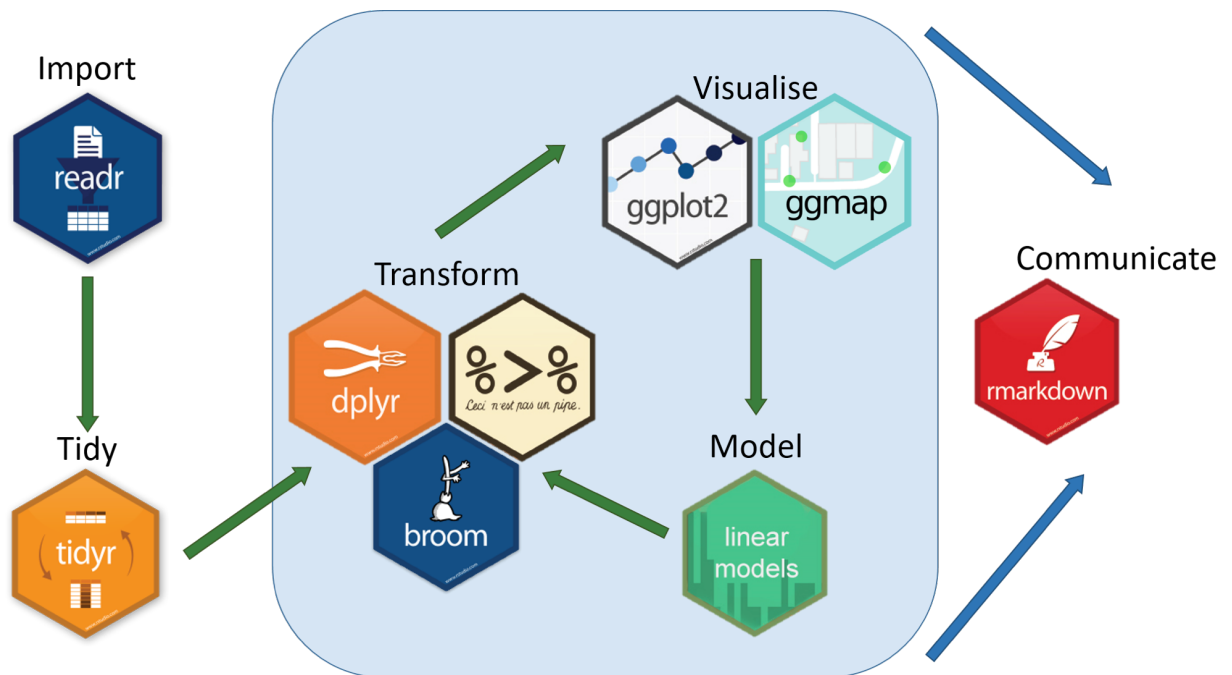
```
In [78]: install.packages()
#install.packages("tidyverse")
# OR

#install.packages("devtools")
#devtools::install_github("tidyverse/tidyverse")
```

Error in install.packages(): no packages were specified
Traceback:

```
1. install.packages()
2. stop("no packages were specified")
```

Tidyverse란?



```
In [80]: library("tidyverse")
library("reshape2")
```

Attaching package: 'reshape2'

The following object is masked from 'package:tidyr':

smiths

stat 패키지 살펴보기

[stat 패키지 함수 목록 바로가기](#)

패키지를 활용해 데이터 조작하기

```
In [83]: head(fold)
fold <- rownames_to_column(fold, var="Accession")
head(fold)
```

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.25644317
Q15149-3	0.340135415	0.08117139
Q15149	0.000319444	0.91329164
Q15149-8	0.006280590	-0.37684673
Q15149-4	0.000223258	1.38131662
P21333-2	0.146827666	-0.11675658

A data.frame: 6 × 3

	Accession	P.adjust	log2FC
	<chr>	<dbl>	<dbl>
1	Q09666	0.002141511	1.25644317
2	Q15149-3	0.340135415	0.08117139
3	Q15149	0.000319444	0.91329164
4	Q15149-8	0.006280590	-0.37684673
5	Q15149-4	0.000223258	1.38131662
6	P21333-2	0.146827666	-0.11675658

```
In [85]: head(fold)
fold <- column_to_rownames(fold, var="Accession")
head(fold)
```

A data.frame: 6 × 3

	Accession	P.adjust	log2FC
	<chr>	<dbl>	<dbl>
1	Q09666	0.002141511	1.25644317
2	Q15149-3	0.340135415	0.08117139
3	Q15149	0.000319444	0.91329164
4	Q15149-8	0.006280590	-0.37684673
5	Q15149-4	0.000223258	1.38131662
6	P21333-2	0.146827666	-0.11675658

A data.frame: 6 × 2

	P.adjust	log2FC
	<dbl>	<dbl>
Q09666	0.002141511	1.25644317
Q15149-3	0.340135415	0.08117139
Q15149	0.000319444	0.91329164
Q15149-8	0.006280590	-0.37684673
Q15149-4	0.000223258	1.38131662
P21333-2	0.146827666	-0.11675658

데이터 melt 시켜 tidy 형태의 데이터로 만들기

In [86]: `head(expression)`

A data.frame: 6 × 6

	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorubicin_3
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.01952
Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.96094
Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.50858
Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.87697
Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.91327
P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.93263

```
In [91]: expression <- rownames_to_column(expression, var="Accession")
         head(expression)
```

A data.frame: 6 × 7

	Accession	MCF7_1	MCF7_2	MCF7_3	Doxorubicin_1	Doxorubicin_2	Doxorub
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
1	Q09666	4984.80051	4959.72469	4885.11485	12114.33764	11727.54481	11587.0
2	Q15149-3	18.44386	20.20029	17.78025	20.96845	19.76066	18.9
3	Q15149	3202.24443	3335.80413	3156.99450	6151.74000	5983.79064	6123.5
4	Q15149-8	13.18008	13.66558	13.50839	10.36080	10.83965	9.8
5	Q15149-4	12.86596	12.27709	12.16204	32.36789	32.90083	31.9
6	P21333-2	3657.09665	4067.28070	3777.62035	3550.46341	3597.42159	3459.9

```
In [92]: expression_melt <- melt(expression)
         head(expression_melt)
```

Using Accession as id variables

A data.frame: 6 × 3

	Accession	variable	value
	<chr>	<fct>	<dbl>
1	Q09666	MCF7_1	4984.80051
2	Q15149-3	MCF7_1	18.44386
3	Q15149	MCF7_1	3202.24443
4	Q15149-8	MCF7_1	13.18008
5	Q15149-4	MCF7_1	12.86596
6	P21333-2	MCF7_1	3657.09665

```
In [93]: subset(expression_melt, expression_melt$Accession == "Q09666")
```

A data.frame: 6 × 3

	Accession	variable	value
	<chr>	<fct>	<dbl>
1	Q09666	MCF7_1	4984.801
7756	Q09666	MCF7_2	4959.725
15511	Q09666	MCF7_3	4885.115
23266	Q09666	Doxorubicin_1	12114.338
31021	Q09666	Doxorubicin_2	11727.545
38776	Q09666	Doxorubicin_3	11587.020

In []: