R

뉴질랜드 AUKLAND 대학의 Ross Lhaka, Robert Clifford Gentleman가 1995년에 개발한 소프트웨어이고 데이터 분석을 위한 통계 및 그래픽스를 지원하는 무료 소프트웨어이다.

R을 사용해야 하는 이유

1.R은 무료소프트웨어이다.

2.데이터 분석을 위해서 가장 많이 쓰는 통계 플랫폼이다.

3.복잡한 데이터를 다양한 그래프로 표현할 수가 있다.

4.분석을 위한 데이터를 쉽게 저장하고 조작할 수 있다.

5.누구든지 유용한 패키지를 생성해서 공유할 수 있고 새로운 기능에 대한 전달이 빠르다.

6.어떠한 OS에서도 설치가 가능하다.

R 설치

<https://www.r-project.org/> <https://www.rstudio.com/>

변수

•변수 이름은 알파벳, 숫자, \_, .(마침표) 사용할 수 있다.

•변수 이름의 첫글자는 알파벳 또는 .(마침표)로 시작할 수 있다.

•.(마침표)로 시작 할 경우에는 바로 뒤에 숫자를 입력할 수 있다. ◦변수 이름으로 가능한 예: a, i, x2, .y

◦변수이름으로 사용이 불가능한 예: 1a, .2, k-j

x <- 1

print(x)

## [1] 1

y <<-2

print(y)

## [1] 2

z = 3

print(z)

## [1] 3

# 전역변수 표현 방식(x변수는 글로벌변수처럼 처리된다.)

sum(x <- c(1,2,3,4,5))

## [1] 15

x

## [1] 1 2 3 4 5

# 지역변수 표현 방식(y변수는 로컬변수처럼 처리된다.)

sum(y = c(1,2,3,4,5))

## [1] 15

y

## [1] 2

숫자(정수, 실수)

x <- 2

print(x)

## [1] 2

# 변수의 타입을 return해주는 함수

class(x)

## [1] "numeric"

**R에서는 기본적으로 숫자를 실수로 표현한다.**

y <- 2L

class(y)

## [1] "integer"

대문자 L(RESERVED WORD)을 사용하면 정확하게 정수로 표현된다.

z <- x+y

# class와 비슷하게 타입과 값을 동시에 return해준다.

str(z)

## num 4

is.type와 같은 형식(is.numeric,is.integer,is.character,is.logical 등)은 변수의 타입을 확인하는 방법으로 BOOLEAN형식으로 표현해준다.

is.numeric(z)

## [1] TRUE

is.integer(y)

## [1] TRUE

**숫자(지수표기법)**

# 1 \* 10^2

1e2

## [1] 100

# 5 \* 10^-1

5e-1

## [1] 0.5

**문자열**

문자열을 입력할 때는 ’’(Single quation mark)나 “”(double quation mark)를 사용하면 된다.

s1 <- 'hello'

class(s1)

## [1] "character"

s2 <- "안녕하세요"

class(s2)

## [1] "character"

is.character(s2)

## [1] TRUE

Boolean(진리값)

AND는 &, OR은 |로 표현한다.

**TRUE & TRUE**

## [1] TRUE

**TRUE | FALSE**

## [1] TRUE

# 약어로도 표현할 수 있다.

**T & F**

## [1] FALSE

하지만 약어 사용은 변수로 선언되었으면 헷갈릴 수 있다. 처리하는 우선순위가 진리값보다 변수가 더 높으므로 약어 사용을 지양하자.

NA(Not Available)

결측치(결측값)으로 데이터 입력 중 실수로 값이 입력되지 않는 경우에 발생한다. (NULL과 다름을 주의)

a <- 100; b <- 90; c <- NA;

is.na(a)

## [1] FALSE

is.na(c)

## [1] TRUE

# 결측값이 있는 변수로 연산 작업을 하면 NA로 표시된다.

a+b+c

## [1] NA

NULL

NULL은 Undefined값을 표현한다. 변수가 초기화되지 않을 때 사용한다. 즉, 선언 시에 숫자를 넣을지 문자를 넣을지 모르지만 변수를 선언하고 싶을 때 사용한다. (NA과 다름을 주의)

# R은 대소문자를 구분하기에 대문자 NULL을 사용해야 한다.

x <- NULL

is.null(x)

## [1] TRUE

is.na(x)

## Warning in is.na(x): is.na()는 유형이 'NULL'인 (리스트 또는 벡터)가 아닌 것

## 에 적용되었습니다

## logical(0)

y <- 100

# 아래 결과를 보고 하나의 값이 없다는 것을 알아야 한다.

x + y

## numeric(0)

산술연산자

기본 사칙연산이 가능하다.

# 몫만 추출하고 싶을 시

100 %/% 3

## [1] 33

# 나머지만 추출하고 싶을 시

100 %% 3

## [1] 1

# 거듭제곱(10\*\*2로도 사용 가능)(참고, plsql에서는 power(10,2))

10^2

## [1] 100

비교연산자

==(같다), !=(다르다), >(크다), <(작다) 등으로 표현 할 수 있다.

10 != 5 & 10>= 10

## [1] TRUE

(10 < 15) | (10 == 5)

## [1] TRUE

자료형

1.vector(벡터) ◦같은 데이터 타입을 갖는 1차원 배열 구조(R의 기본데이터 구조)

◦c(): combine value

◦벡터는 중첩이 불가능하다.

◦벡터는 단일 데이터 타입만 가능

◦데이터 변환규칙: integer < double < character

x <- c(1,2,3,4,5)

mode(x)

## [1] "numeric"

class(x)

## [1] "numeric"

str(x)

## num [1:5] 1 2 3 4 5

벡터는 단일 데이터 타입만 가능하다는 것을 기억하자. 그래서 데이터 변환 규칙에 따라 아래 변수는 캐릭터형으로 바뀐다.

x <- c(1,2,3,4,**'5'**)

**mode(x)**

**## [1] "character"**

**class(x)**

**## [1] "character"**

**str(x)**

**## chr [1:5] "1" "2" "3" "4" "5"**

벡터는 중첩이 불가능하여서 중첩해서 쓸 시 중첩 벡터를 풀어버린다.

x <- c(1,2,3,c(4,5))

print(x)

## [1] 1 2 3 4 5

아래 코드를 보고 함수를 쓰고 싶을 때에 주의하자.

# dataframe의 column이름을 표현

> x<-c(1,2,3,sum=c(4,5))

> x

sum1 sum2

1 2 3 4 5

# sum 사용

> y<-c(1,2,3,sum(c(4,5)))

> y

[1] 1 2 3 9

>

---------------------------------------------------------------------------------------------------------z<- c(과목 = c(80,90,96))

> z

과목1 과목2 과목3

80 90 96

>

넘버링이 자동으로 된다.

z의 셀 이름 바꾸기 names( )

**names(z) <-c("국어","영어","수학")**

> z

국어 영어 수학

80 90 96

셀 이름 지우기

**names(z) <- NULL**

[1] 80 90 96

셀이름을 모두 NA로 표현

**names(z) <- NA**

<NA> <NA> <NA>

80 90 96

names :벡터의 각 셀에 이름을 설정

length(y)

NROW(y)

벡터안에 들어있는 갯수 (plsql의 배열.이름.count랑 똑같음)

y라는 백터변수안에 5개가 들어있는것이다.

plsql에서 어떤특정한값을 하려면 plsql요소를 사용했다.

알에서의 벡터는 네스티드배열처럼 1번부터 시작한다.

벡터의 요소번호를 이용하는 방법

y[1] 1번방

y[2]2번방

어떤요소를 제외시켜서 보려고한다. -1

y[-1]

y[c(-1,-3)] 1번과 3번을 제외

y["하나"]

y[c("하나","다섯")]

y[-1,:-4] 1번요소부터 4번요소까지 다제외시키겠다.

**x<- 1:100**

1부터 100까지 x라는 벡터변수를만들어서 시퀀스하게 만들어진다.

얘네는 1씩증가하지만 2씩증가, 5씩증가하게끔만들어보자.

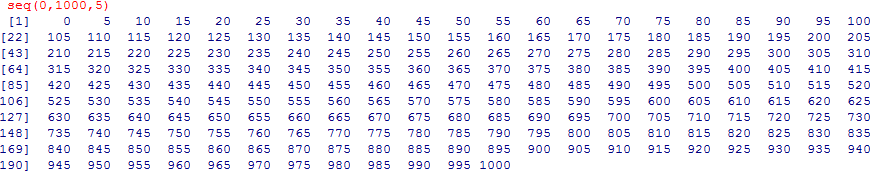
sequence : 자동일련번호를 생성

**seq라는 메쏘드**

**seq(1,5,1)**

[1] 1 2 3 4 5

**seq(0,1000,5)**



**10:1**



리버스 10부터1

**seq(10,0,-1)**



> x<- c(2,4,6,8,10)

> x

[1] 2 4 6 8 10

이벡터안에들어있는 갯수만큼 시퀀스하게 만들고 싶다.

**> seq\_along(x) x변수에있는 갯수만큼 내가 일련번호를 만드는것**

[1] 1 2 3 4 5

**> 1:NROW(x)**

[1] 1 2 3 4 5

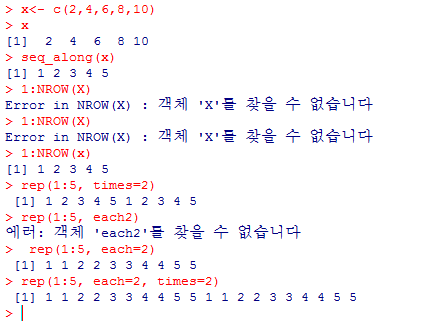
rep(1:5, times=2)

1부터 5까지 두번반복

rep(1:5, each=2)

각각의 값을 두개씩

rep(1:5, each=2, times=2)



**벡터의 값 수정**

> x<- c(1:5) 1부터 5까지 할당

**> x[2] <- 8 2번째의 요소값을 8로 수정**

> x

[1] 1 8 3 4 5

>

**> x[3:5] <-c(30,40,50) 3부터 5까지 수정**

> x

[1] 1 8 30 40 50

**벡터의 값을 추가**

**x[6] <- 60** 6번방을 만들어서 그 곳에 값을 넣겠습니다.

> x[6] <- 60

> x

[1] 1 8 30 40 50 60

x[8] <-80 8번방만들어서 그곳에 값을 넣겠습니다.

> x[8] <-80

> x

[1] 1 8 30 40 50 60 NA 80 7번방은 NA로 결측치가되어있음

7번방 결측치에 70을 넣음

> x[7] <- 70

> x

[1] 1 8 30 40 50 60 70 80

>

x라는 변수에 90을 추가하겠습니다. 8번방뒤에

> append(x,90,after=8)

[1] 1 8 30 40 50 60 70 80 90

8번방 뒤에다가 값을 넣어줌 하지만 xㅇ값확인해보면안되어있음

> x

[1] 1 8 30 40 50 60 70 80

>

그래서 그냥어펜드하지말고

> x<- append(x,90,after=8)

> x

[1] 1 8 30 40 50 60 70 80 90

이렇게 x값에 적용시켜줘야한다.

c.f before는 없다.

> x<- append(x,10, after=3)

[1] 1 8 30 10 40 50 60 70 80 90

>

방과방사이에 값을 끼워넣어줘야하면 append를 써줘야한다.

**벡터연산**

> x<- c(1:5)

> x

[1] 1 2 3 4 5

> x+10

[1] 11 12 13 14 15

> x\*10

[1] 10 20 30 40 50

> x/10

[1] 0.1 0.2 0.3 0.4 0.5

> x-10

[1] -9 -8 -7 -6 -5

>

벡터 변수끼리의 비교

**배열비교**

> x <-c(1,2,3)

> y<-c(1,2,3)

> z<-c(1,2,4)

> x==y

[1] TRUE TRUE TRUE

> x==z

[1] TRUE TRUE FALSE

>

각각의 배열요소 끼리 비교 한다

> identical(x,y)

[1] TRUE

> identical(x,z)

[1] FALSE

> w<-c(1:5)

> x==w

[1] TRUE TRUE TRUE FALSE FALSE

경고메시지(들):

In x == w : 두 객체의 길이가 서로 배수관계에 있지 않습니다

> identical(x,w)

[1] FALSE

>

벡터끼리 값에대한 요소끼리 비교해야할 때는 ==

각각의 값들이 일치가되야하는지

전체를 가지고 비교할때는 identical

> x <- c(1,2,3,4)

> y <-c(1,2,3,4,4)

> x==y

[1] TRUE TRUE TRUE TRUE FALSE

경고메시지(들):

In x == y : 두 객체의 길이가 서로 배수관계에 있지 않습니다

> identical(x,y)

[1] FALSE

>

setequal(x,y) 이라는 메쏘드

> setequal(x,y)

[1] TRUE

**identical(x,y) : 두벡터의 값이 동일한지 판단**

**setequal(x,y) : 두벡터의 같은 집합인지 판단**

union(x,y) :합집합

> union(x,y)

[1] 1 2 3 4

> 중복제거해서 합쳐진다.

intersect(x,y);교집합

> intersect(x,y)

[1] 1 2 3 4

setdiff(x,y) :차집합

> setdiff(x,y)

numeric(0)

>

> x<-c(1,2,3,4)

> y<-c(1,4,6)

> union(x,y)

[1] 1 2 3 4 6

> intersect(x,y)

[1] 1 4

> setdiff(x,y)

[1] 2 3

>

x변수에 값이 있으면 TURE 없으면 FALSE

> 1 **%in%** x

[1] TRUE

> 5%in%x

[1] FALSE

r은 sort라는 메소드로 정렬을 제공한다.

u<-sort(union(x,y),decreasing = TRUE)

[문제1] x변수에 1,3,5,7,9 값을 입력, y 변수에 1,2,3,4,5 값을 입력하세요.

> x <- c(1,3,5,7,9)

> x

[1] 1 3 5 7 9

> y <- c(1,2,3,4,5)

> y

[1] 1 2 3 4 5

[문제2] x 변수와 y 변수를 중복성 없이 하나로 합친후에 u 변수에 넣어 주세요.

> u <- union(x,y)

> u

[1] 1 3 5 7 9 2 4

> u <- sort(union(x,y))

> u

[1] 1 2 3 4 5 7 9

> u <- sort(union(x,y),decreasing = TRUE)

> u

[1] 9 7 5 4 3 2 1

[문제3] x 변수와 y 변수의 값들중에 중복성만 추출해서 i 변수에 넣어주세요.

> i <- intersect(x,y)

> i

[1] 1 3 5

[문제4] x 변수의 값과 y 변수의 값중에 순수하게 x 변수에 들어 있는 값만 추출해서 m 변수에 넣어 주세요.

> m <- setdiff(x,y)

> m

[1] 7 9

>

[문제5] x 변수의 값과 y 변수의 값이 일치가 되면 TRUE 아니면 FALSE를 출력해주세요.

> setequal(x,y)

[1] FALSE

[문제6] x 변수에 값들을 10을 곱한 결과를 x 변수에 적용하세요.

> x

[1] 1 3 5 7 9

> x <- x \* 10

> x

[1] 10 30 50 70 90

[문제7] x 변수에 있는 50을 5로 수정하세요.

> x[3] <- 5

> x

[1] 10 30 5 70 90

[문제8] x 변수에 있는 10 30 5 70 90을 원래의 값으로 1,3,5,7,9로 되돌려 주세요.단 union, 정수 나누기, sort 만 사용하세요

> x <- sort(union(x[-3] %/% 10,x[3]))

> x

[1] 1 3 5 7 9

[문제9] x변수에 11숫자를 제일 뒤에 입력하세요. 단 append와 length를 이용하세요.

> x<-append(x,11,after=length(x))

> x

[1] 1 3 5 7 9 11

[문제10] x 변수에 제일 뒤에 있는 값을 NA로 수정하세요. 단 length를 이용하세요.

> x[length(x)] <- NA

> x

[1] 1 3 5 7 9 NA

> x<-c('b','a','d','a',NA)

> x

[1] "b" "a" "d" "a" NA

> 'a'%in% x

[1] TRUE

있는지만알지 어디에있는지는 알 수가 없다.

> x=='a'

[1] FALSE TRUE FALSE TRUE NA

요소부분에 있으면 TURE 없으면FALSE

> x[x=='a']

[1] "a" "a" NA

NA값 가치나옴

조건에 해당하는 요소번호 찾기

> which('a' ==x)

[1] 2 4

이 값이 이 벡터안에 몇번째에위치하는지 리턴

> x[which('a'==x)]

[1] "a" "a"

>

NA는 없음

which 정말 많이씀 중요!!

> is.na(x)

[1] FALSE FALSE FALSE FALSE TRUE

na가있으면 TRUE 없으면FALSE

> which(NA ==x )

integer(0)

안돌아 간것. is null, is not null하듯 이하 처럼 사용

> which(is.na(x))

[1] 5

> x[which('a'==x)]

[1] "a" "a"

> x[which('a'==x)]<-'aa'

> x

[1] "b" "aa" "d" "aa" NA

그 값을 찾아서 수정해줘야할 때

> x<-c(1:5)

> x

[1] 1 2 3 4 5

> y<-c(1,2,3,4,5)

> y

[1] 1 2 3 4 5

> setequal(x,y)

[1] TRUE

> identical(x,y)

[1] FALSE

값이같은데 TRUE FALSE다르게나오는 이유

구조확인하기

**str(x)**

**str(y)**

**> str(x)**

**int [1:5] 1 2 3 4 5 인티저**

**> str(y)**

**num [1:5] 1 2 3 4 5 루매닉**

**setequal은 정수든 실수든 값이 같으면 TRUE**

**identical은 타입도 완전히 같아야한다.**

**x는 integer 정수형**

**y는 rumanic 실수형**

형변환함수를 배울 것이다.

> as.integer(y)

[1] 1 2 3 4 5

루매닉을 인티저형으로 바꿔주기 미리보기임 적용시켜줘야함

> y<-as.integer(y)

> identical(x,y)

[1] TRUE

R은 헬프기능이잘되어있다.

help(identical)

?(identical)

**벡터: 1차원배열을 표현하는 하나의 방법**

**동일한타입의 데이터값을 표현**

**문자> 실수> 정수 순으로 우선순위**

**list**

-서로 다른 데이터 타입을 갖는 벡터들을 저장하거나 또다른 리스트 저장가능한 구조이다.

-list(키 = 값, 키=값)

> x<- list(name = '홍길동', addr = '서울시', pn = '010-1111-1234')

> x

$`name`

[1] "홍길동"

$addr

[1] "서울시"

$pn

[1] "010-1111-1234"

> str(x)

List of 3

$ name: chr "홍길동"

$ addr: chr "서울시"

$ pn : chr "010-1111-1234"

> class(x)

[1] "list"

> mode(x)

[1] "list"

x$key이름 x에있는 값을 뽑아낸다.

> x$name

[1] "홍길동"

> x$addr

[1] "서울시"

> x$pn

[1] "010-1111-1234"

> x[1] --key하고 value둘다볼때

$`name`

[1] "홍길동"

> x[[1]] --value값만보고싶을때

[1] "홍길동"

> x[1:3]

$`name`

[1] "홍길동"

$addr

[1] "서울시"

$pn

[1] "010-1111-1234"

**list 요소 추가 x% key <- value**

**> x$sal <-10000**

> x

$`name`

[1] "홍길동"

$addr

[1] "서울시"

$pn

[1] "010-1111-1234"

$sal

[1] 10000

**list 요소 제거**

**> x$sal <-NULL**

> x

$`name`

[1] "홍길동"

$addr

[1] "서울시"

$pn

[1] "010-1111-1234"

**list 요소값을 수정**

**> x$pn <- '010-1234-1004'**

> x

$`name`

[1] "홍길동"

$addr

[1] "서울시"

$pn

[1] "010-1234-1004"

**list 중첩 list는 중첩을 할 수 있다. (벡터는 중첩안됨)**

> y <- list(a=list(val=c(1,2,3)), b= list(val=c(1,2,3,4)))

> y

$`a`

$`a`$`val`

[1] 1 2 3

$b

$b$`val`

[1] 1 2 3 4

> y$a

$`val`

[1] 1 2 3

> y$b

$`val`

[1] 1 2 3 4

data frame덕분에 별로쓸일은없다. 알에서 벡터, 데이터프레임, 행렬을 많이 쓴다.

[문제11] lst 변수에 name = 'king' , height = 180, weight = 70 값을 넣어 주세요.

> lst <- list(name = 'king' , height = 180, weight = 70)

> lst

$name

[1] "king"

$height

[1] 180

$weight

[1] 70

[문제12] lst 변수에 blood = 'A' 추가하세요.

> lst$blood <- 'A'

> lst

$name

[1] "king"

$height

[1] 180

$weight

[1] 70

$blood

[1] "A"

[문제13] lst 변수에 name의 값을 'scott'로 수정하세요.

> lst$name

[1] "king"

> lst$name <- 'scott'

> lst$name

[1] "scott"

또는

> lst[1] <- 'scott'

[문제14] lst변수에 2번인덱스 값만 출력해주세요.

> lst[[2]]

[1] 180

[문제15] lst변수에 blood 이름을 blood type 이름으로 수정하세요.

> names(lst)

[1] "name" "height" "weight" "blood"

> names(lst)[4]

[1] "blood"

> names(lst)[4] <- "blood type"

> names(lst)

[1] "name" "height" "weight" "blood type"

> lst

$name

[1] "scott"

$height

[1] 180

$weight

[1] 70

$`blood type`

[1] "A"

[문제16] x 변수에 벡터값 1,2,3,5,6 을 입력한 후 3번째 요소 뒤에 4를 입력하세요.

> x<-c(1,2,3,5,6)

> x

[1] 1 2 3 5 6

> x<- append(x,4,after=3)

> x

[1] 1 2 3 4 5 6

[문제17] 1부터 3씩 증가하는 10 이하의 정수값을 출력하세요.

> seq(1,10,3)

[1] 1 4 7 10

seq(1,10,by=3)

**참고: 알아두자. seq(1,10,length.out=100)**

1부터10까지 균일하게 100개로쪼개주는..

[문제18] 10 부터 20 까지의 값을 x 변수에 생성한 후 원소의 값이 15 이상이고 18 이하인 값들만 출력하세요.

> x

[1] 10 11 12 13 14 15 16 17 18 19 20

> x>=15 &x<=18

[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE FALSE FALSE

> x[x>=15&x<=18]

[1] 15 16 17 18

[문제19] 10 부터 20 까지의 값을 x 변수에 생성한 후 원소의 값이 15 이상이고 18 이하인 값들만 2곱한 값으로 수정하세요 > x[x>=15&x<=18]\*2

[1] 30 32 34 36

----------------------------------------------------------------------------------------------------------

**### 행렬(MATRIX)**

- 벡터처럼 한가지 유형의 스칼라 값만 저장

- MATRIX함수를 이용해서 행렬을 생성

- 행과 열을 지정해주어야 한다.

x<-c(1:9)

> x<-matrix(c(1:9),nrow=3) #nrow=3행의수

> x

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

> x<-matrix(c(1:9),ncol=3) #열의수

> x

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

> x<-matrix(c(1:9),nrow=3, ncol=3)

> x<-matrix(c(1:9),ncol=1)

> x

[,1]

[1,] 1

[2,] 2

[3,] 3

[4,] 4

[5,] 5

[6,] 6

[7,] 7

[8,] 8

[9,] 9

**> x<-matrix(c(1:9),ncol=1)**

> x

[,1]

[1,] 1

[2,] 2

[3,] 3

[4,] 4

[5,] 5

[6,] 6

[7,] 7

[8,] 8

[9,] 9

>

**> nrow(x) #행의수**

[1] 9

**> ncol(x) #열의수**

[1] 1

**> dim(x) #행, 열의수**

[1] 9 1

**채우는 순서**

> x<-matrix(c(1:9),nrow=3,byrow=TRUE) #행부터 값을 채움

> x

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

[3,] 7 8 9

x<-matrix(c(1:9),nrow=3,byrow=FALSE) #열부터 값을 채움 (기본값-byrow안써주면이렇게 )

> x<-matrix(c(1:9),nrow=3,byrow=FALSE

> x

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

x <- matrix(c(1:9), ncol = 3)

dimnames(x) <- list(c("row1","row2","row3"), c("col1","col2","col3"))

**#행과 열의 이름을 만드는 방법**

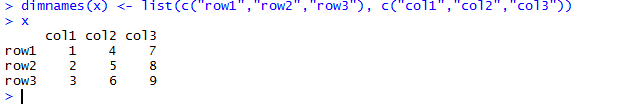
> x

col1 col2 col3

row1 1 4 7

row2 2 5 8

row3 3 6 9



x <- matrix(c(1:9), ncol =3)

dimnames(x)<- list(c("a1","a2","a3"),c("c1","c2","c3"))

리스트안을 벡터로표현해야한다.

rownames(x) #행 이름보는 법

rownames(x)<-c("r1","r2","r3") #행이름 수정

colnames(x) # 열이름

colnames(x) <-c("b1","b2","b3") #열이름 수정하는법

한꺼번에수정하면 dimnames

따로따로만들어서 합치기

> cells<-c(1:9)

> rname <- c("r1","r2","r3")

> cname <-c("c1","c2","c3")

> x<-matrix(cells, nrow=3,byrow=TRUE,dimnames=list(rname,cname))

> x

c1 c2 c3

r1 1 2 3

r2 4 5 6

r3 7 8 9

>

> **(**x<-matrix(cells, nrow=3,byrow=TRUE,dimnames=list(rname,cname))**)**

c1 c2 c3

r1 1 2 3

r2 4 5 6

r3 7 8 9

>

맨앞뒤로 괄호를 또묶어주면 생성과동시에출력된다.

> class(x)

[1] "matrix"

> mode(x)

[1] "numeric"

class는 자료형을표현

mode는 타입을 표현

행렬이름[행인덱스,열인덱스]

**> x[1,1] # 1행1열**

[1] 1

**> x[2,1] #2행1열**

[1] 4

> x[3,2]

[1] 8

**> x[1,]**

c1 c2 c3

1 2 3

**> x[,2]**

r1 r2 r3

2 5 8

**> x[1,-2] #음수로표현하면 그것만 제외**

c1 c3

1 3

**> x[1,2:3]**

c2 c3

2 3

**> x[c(1,3),c(1,2)] #인덱스번호를 벡터로구성해서 보면 원하는값뽑음**

c1 c2

r1 1 2

r3 7 8

**> x["r1",] #로우이름으로 찾기**

c1 c2 c3

1 2 3

**> x[,"c2"]**

r1 r2 r3

2 5 8

>

**수정할때**

**> x[1,1]<-10**

> x[1,1]

[1] 10

**행렬의 연산**

**> x<- matrix(c(1:4), ncol=2)**

**> x**

[,1] [,2]

[1,] 1 3

[2,] 2 4

**> x + 10**

[,1] [,2]

[1,] 11 13

[2,] 12 14

**> x - 10**

[,1] [,2]

[1,] -9 -7

[2,] -8 -6

**> 10 - x**

[,1] [,2]

[1,] 9 7

[2,] 8 6

**> x / 2**

[,1] [,2]

[1,] 0.5 1.5

[2,] 1.0 2.0

**> x \* 10**

[,1] [,2]

[1,] 10 30

[2,] 20 40

**> x + x**

[,1] [,2]

[1,] 2 6

[2,] 4 8

**> x - x**

[,1] [,2]

[1,] 0 0

[2,] 0 0

**> x \* x**

[,1] [,2]

[1,] 1 9

[2,] 4 16

우리가생각하는 행렬의 곱은 아니다.

**> x %\*% x #우리가생각하는 행렬의 곱**

[,1] [,2]

[1,] 7 15

[2,] 10 22

**> t(x) #전치행렬(행과 열의 위치를 바꾼다) 1행2열 <-> 2행1열**

[,1] [,2]

[1,] 1 2

[2,] 3 4

**> solve(x) #역행렬**

[,1] [,2]

[1,] -2 1.5

[2,] 1 -0.5

**> x %\*% solve(x) #단위행렬**

[,1] [,2]

[1,] 1 0

[2,] 0 1

**> x<- matrix(c(1:6),ncol=3)**

> x

[,1] [,2] [,3]

[1,] 1 3 5

[2,] 2 4 6

**> dim(x) #행과열에대한모양**

[1] 2 3 #2행3열

이걸이용해 행과열의 모양을 바꿀 수 있다.

> dim(x) <-c(3,2)

> x

[,1] [,2]

[1,] 1 4

[2,] 2 5

[3,] 3 6

**행렬을 합치는 법**

> x <- matrix(c(1:9),nrow=3)

> y <- matrix(c(1:9),nrow=3)

> x

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

> y

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

**> cbind(x,y) 열기준으로 합침 (옆으로붙임)**

[,1] [,2] [,3] [,4] [,5] [,6]

[1,] 1 4 7 1 4 7

[2,] 2 5 8 2 5 8

[3,] 3 6 9 3 6 9

**> rbind(x,y) 행기준으로 합침(밑으로 붙임)**

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

[4,] 1 4 7

[5,] 2 5 8

[6,] 3 6 9

[문제20] x 변수에 행렬을 구성하세요. 값은 1부터 10까지 입력하시고 5행 2열으로 만들면서 값은 열을 기준으로 생성하세요.

> x<-matrix(c(1:10),nrow=5,ncol=2,byrow=FALSE)

> x

[,1] [,2]

[1,] 1 6

[2,] 2 7

[3,] 3 8

[4,] 4 9

[5,] 5 10

[문제21] x 변수에 열을 기준으로 11,12,13,14,15 **값을 추가**하세요.

> x<- cbind(x,c(11,12,13,14,15))

> x

[,1] [,2] [,3]

[1,] 1 6 11

[2,] 2 7 12

[3,] 3 8 13

[4,] 4 9 14

[5,] 5 10 15

되게많이쓴다.!! 기억해두기!!! 행의갯수만큼 벡터값이 표현되어있어야함

[문제22] x 변수에 행을 기준으로 16,17,18 값을 추가하세요.

> x<- rbind(x,c(16,17,18))

> x

[,1] [,2] [,3]

[1,] 1 6 11

[2,] 2 7 12

[3,] 3 8 13

[4,] 4 9 14

[5,] 5 10 15

[6,] 16 17 18

[문제23] x변수에 6행의 값을 20,21,22 로 수정하세요.

> x[6,]<- c(20,21,22) #c[6,]6행모든값

> x

[,1] [,2] [,3]

[1,] 1 6 11

[2,] 2 7 12

[3,] 3 8 13

[4,] 4 9 14

[5,] 5 10 15

[6,] 20 21 22

[문제24] x 변수에 6행을 제거해주세요.

> x<-x[-6,]

> x

[,1] [,2] [,3]

[1,] 1 6 11

[2,] 2 7 12

[3,] 3 8 13

[4,] 4 9 14

[5,] 5 10 15

**array(배열)**

-같은 데이터 타입을 갖는 3차원 배열구조

-matrix는 2차원 행렬, array는 3차원 행렬

-array 함수를 이용해서 배열 생성

**> x <- array(c(1:6), dim=c(2,3))**

**> x**

[,1] [,2] [,3]

[1,] 1 3 5

[2,] 2 4 6

**> x<- array(c(1:24), dim=c(2,3,4))**

**> x**

, , 1

[,1] [,2] [,3]

[1,] 1 3 5

[2,] 2 4 6

, , 2

[,1] [,2] [,3]

[1,] 7 9 11

[2,] 8 10 12

, , 3

[,1] [,2] [,3]

[1,] 13 15 17

[2,] 14 16 18

, , 4

[,1] [,2] [,3]

[1,] 19 21 23

[2,] 20 22 24

**x[1,1,]**

[1] 1 7 13 19

**> x[1,,]**

[,1] [,2] [,3] [,4]

[1,] 1 7 13 19

[2,] 3 9 15 21

[3,] 5 11 17 23

>

**> x[,,4]**

[,1] [,2] [,3]

[1,] 19 21 23

[2,] 20 22 24

> dimnames(x)

NULL

**> dimnames(x)<-list(c("r1","r2"),c("c1","c2",'c3"')) #dimnames로 행이름 보고,수정**

> x

, , 1

c1 c2 c3"

r1 1 3 5

r2 2 4 6

, , 2

c1 c2 c3"

r1 7 9 11

r2 8 10 12

, , 3

c1 c2 c3"

r1 13 15 17

r2 14 16 18

, , 4

c1 c2 c3"

r1 19 21 23

r2 20 22 24

**matrix에서 rownames(x), colnames(x)와 비슷**

matrix와 array사용법 비슷

**> class(x)**

[1] "array"

**> mode(x)**

[1] "numeric"

**> str(x)**

int [1:2, 1:3, 1:4] 1 2 3 4 5 6 7 8 9 10 ...

- attr(\*, "dimnames")=List of 3

..$ : chr [1:2] "r1" "r2"

..$ : chr [1:3] "c1" "c2" "c3\""

..$ : NULL

**> is.matrix(x) #이게 matrix냐?**

[1] FALSE

**> is.array(x) #이게 array냐?**

[1] TRUE

[문제25] x 배열을 생성하세요. 1부터 12까지 값을 가지고있는 배열을 생성하세요. 면은 3개가 만들어지도록하세요.

x<-array(c(1:12),dim=c(2,2,3))

[문제26] x 배열 변수에 컬럼이름은 'a','b'로 설정하세요.

[문제27] x 배열 변수에 행이름은 'row1','row2'로 설정하세요.

> dimnames(x) <- list(c('row1','row2'),c('a','b'))

> x

, , 1

a b

row1 1 3

row2 2 4

, , 2

a b

row1 5 7

row2 6 8

, , 3

a b

row1 9 11

row2 10 12

rowmanes(x) <-c('row1','row2')

[문제28] x 배열 변수에 면을 2로 수정하세요.

> dim(x)<-c(2,3,2)

> x

, , 1

[,1] [,2] [,3]

[1,] 1 3 5

[2,] 2 4 6

, , 2

[,1] [,2] [,3]

[1,] 7 9 11

[2,] 8 10 12

>

factor(팩터)

- 범주형 : 데이터를 미리 정해진 유형으로 분류

- sql hr employees 테이블의 job\_id나 department\_id같은경우. 이걸 변수처리한다.

- level : A,B,C,D,E , "좋음","보통","나쁨"

- 종류: 순서형(ORDINAL), 명목형(nominal)

- 순서형 (ordinal) : 데이터간 순서를 둘 수 있는 경우(ABCD..)

- 명목형 (nominal) : 데이터간 크기 비교가 불가능한 경우(남여)

> x<- factor("좋음",c("좋음","보통","나쁨")) #벡터안은 레벨이고 그 앞에 “좋음”은 선택한 것

> x

[1] 좋음

Levels: 좋음 보통 나쁨

> str(x)

Factor w/ 3 levels "좋음","보통",..: 1

> class(x)

[1] "factor"

> mode(x)

[1] "numeric" #factor형인데 문자형같지만 숫자형으로표현됨

> x<- factor("좋음",c("좋음","보통","나쁨"),**ordered=TRUE) #순서형**

> x

[1] 좋음

Levels: 좋음 < 보통 < 나쁨

> str(x)

Ord.factor w/ 3 levels "좋음"<"보통"<..: 1

> class(x)

[1] "ordered" "factor"

> mode(x)

[1] "numeric"

데이터프레임의 특정한자료대부분은 factor형으로 되어있다.

그런데 factor형은 분류되어져있는 값이기때문에 처리하기 까다롭다.

**> nlevels(x) #레벨의 개수**

[1] 3

**> levels(x) #레벨의 목록 확인**

[1] "좋음" "보통" "나쁨"

> levels(x)[1]

[1] "좋음"

> levels(x)[2]

[1] "보통"

> levels(x)[3]

[1] "나쁨"

> levels(x) <- c("good","normal","bad") # 레벨의목록바꾸기

> x

[1] good

Levels: good < normal < bad

> is.factor(x) #factor형이냐?

[1] TRUE

> is.ordered(x) #순서형이냐?

[1] TRUE

> gender<-factor(c("male","male","female"),c("male","female"))

> gender

[1] male male female

Levels: male female

**> x<- ordered(c("a","b"),c("a","b","c"))**

> x

[1] a b

Levels: a < b < c

> is.ordered(x)

[1] TRUE

> is.factor(x)

[1] TRUE

**> x<-factor(c("large","medium","small","small","large","medium"),levels=c("small","medium","large"))**

> x

[1] large medium small small large medium

Levels: small medium large

여기에서 6번째 뒤에 추가해보자.

> x<- append(x,"tiny", after=6)

> x

[1] "3" "2" "1" "1" "3" "2" "tiny"

이렇게됨

**나머지는 루메닉으로 바뀌어짐 factor요소번호바꿀때는 중요**

**x<-as.vector(x) #형변환함수 이 x형을 이용해서 형을변환한다.**

**벡터형으로바꿔서 append하게넣어주고 다시 factor로바꿔준다.**

> x<- factor(c("large","medium","small","small","large","medium"),levels=c("small","medium","large"))

> x

[1] large medium small small large medium

Levels: small medium large

> x<- append(x,"tiny", after=6)

> x

[1] "3" "2" "1" "1" "3" "2" "tiny"

> x<- factor(c("large","medium","small","small","large","medium"),levels=c("small","medium","large"))

> x

[1] large medium small small large medium

Levels: small medium large

> x<-as.vector(x)

> x

[1] "large" "medium" "small" "small" "large" "medium"

> x<- append(x,"tiny", after=6)

> x

[1] "large" "medium" "small" "small" "large" "medium" "tiny"

> as.factor(x)

[1] large medium small small large medium tiny

Levels: large medium small tiny

> x

[1] "large" "medium" "small" "small" "large" "medium" "tiny"

>

분석하면서 보통대부분 데이터프레임을 쓰는데 대부분은 factor형이다.

[문제29] 벡터에 있는 값 "large", "medium", "small", "small", "large", "medium" 을 factor 변수로 구성하세요. 변수이름은 x로 생성하시고 levels samall, medium, large 순으로 지정하세요.

x <- factor(c("large", "medium", "small", "small", "large", "medium"),

levels = c("small","medium","large"))

x

[문제30] x factor형 목록이름중에 small 을 s로 수정하세요.

**levels(x)[levels(x) == "small"] <- c("s")**

x

factor하다보면 잘안되는게 많은 데 vector로 바꿔서as.vector(x) 처리한후 factor로 다시바꿈

as.factor(x)

as.ordered(x)

자료형에 따라 메쏘드가 다르다. 이런거 구분 잘해놔야한다.

**#level값 추가하는 방법**

x<- as.vector(x)

x<-factor(x,c("tiny","small","medium","large"))

> x

[1] large medium small small large medium

Levels: tiny small medium large

#데이터 분석시 자료형을 잘 쓰는 것이 중요함, 자료형에 따라 method틀림

**data frame(데이터프레임)**

-각기 다른 데이터 타입을 갖는 컬럼으로 이루어진 2차원 테이블 구조(DB의 TABLE과 유사하다)

-data.frame( ) 함수를 이용해서 각 컬럼, 행을 구성한다.

> df <- data.frame(x=c(1,2,3,4,5),y=c(6,7,8,9,10))

> df

x y

1 1 6

2 2 7

3 3 8

4 4 9

5 5 10

> mode(df)

[1] "list"

> class(df)

[1] "data.frame"

> str(df)

'data.frame': 5 obs. of 2 variables:

$ x: num 1 2 3 4 5

$ y: num 6 7 8 9 10

>

**서로다른 값을 처리할 수 있는 list들이 모여서 데이터프레임을 만드는 것임.**

이데이터프레임안에 저장되어있는 row의수, coloum의 수

분석하기위해서는 중요하다.

variable이 컬럼과 같은 의미라고보면된다. 설문지 항목들과 같은 존재임.

> df$x

[1] 1 2 3 4 5

> df$y

[1] 6 7 8 9 10

변수이름, 컬럼이름이 곧 name

> df <- data.frame(name=c('scott','harden','curry'),sql=c(90,80,70),plsql=c(70,80,90))

> df

name sql plsql

1 scott 90 70

2 harden 80 80

3 curry 70 90

> str(df)

'data.frame': 3 obs. of 3 variables:

$ name : **Factor** w/ 3 levels "curry","harden",..: 3 2 1

$ sql : num 90 80 70

$ plsql: num 70 80 90

이름이 팩터형으로 되어있다. '', "" 이걸로 요소값을 넣으면 팩터형으로 되어서 수정작업하기 힘들게 된다. csv파일 끌어와서 분석해보면 모든 문자열은 factor로되어있어서 좋지않다.

데이터프레임을 만들 때 옵션이있다.

> df <- data.frame(name=c('scott','harden','curry'),sql=c(90,80,70),plsql=c(70,80,90), **stringsAsFactors=FALSE**) #대소문자구분조심

> str(df)

'data.frame': 3 obs. of 3 variables:

$ name : **chr** "scott" "harden" "curry"

$ sql : num 90 80 70

$ plsql: num 70 80 90

**stringsAsFactors=FALSE 옵션으로 인해서 chr형으로 되었다.**

> df[1,1]

[1] "scott"

> df[1,1]<-'james'

특정컬럼을 확인 **변수$컬럼명**

df$sql

df$name

df$plsql

**> df$sql**

[1] 90 80 70

**> df$name**

[1] "james" "harden" "curry"

**> df$plsql**

[1] 70 80 90

**특정컬럼 추가**

df$r<-c(80,70,60)

**> df$r<-c(80,70,60)**

> df

name sql plsql r

1 james 90 70 80

2 harden 80 80 70

3 curry 70 90 60

**특정컬럼 삭제**

**> df$r<-NULL**

> df

name sql plsql

1 james 90 70

2 harden 80 80

3 curry 70 90

> df

name sql plsql

1 james 90 70

2 harden 80 80

3 curry 70 90

> df[1,]

name sql plsql

1 james 90 70

> df[,2]

[1] 90 80 70

> df[c(1,3),2]

[1] 90 70

> df[-1,-2]

name plsql

2 harden 80

3 curry 90

> df[,c("sql","plsql")]

sql plsql

1 90 70

2 80 80

3 70 90

> df[,"sql"]

[1] 90 80 70

> df[,"sql", drop=FALSE] #drop=FALSE 옵션을 넣어주면 세로로보여줌

sql

1 90

2 80

3 70

> df$sql[1]

[1] 90

> df$sql[2]

[1] 80

> x<-data.frame(1:3)

> x

X1.3

1 1

2 2

3 3

> colnames(x)

[1] "X1.3"

> rownames(x)

[1] "1" "2" "3"

> colnames(x)<-c("val") #컬럼이름 지정

> rownames(x)<-c("a","b","c") #행이름 지정

> x

val

a 1

b 2

c 3

> names(x)

[1] "val"

> colnames(x)

[1] "val"

두개같다.

> names(x)<-c("col")

> names(x)

[1] "col"

> colnames(x)

[1] "col"

> d<- data.frame(a=1:3,b=4:6,c=7:9)

> d

a b c

1 1 4 7

2 2 5 8

3 3 6 9

> names(d)%in% c("b","c")

[1] FALSE TRUE TRUE

어떤컬럼들이 있는지 빨리 볼 수있다. 그위치에있으면 TRUE 없으면 FALSE

이때

> **d[,**names(d)%in% c("b","c")**] #특정컬럼만 뽑을 수 있다.**

b c

1 4 7

2 5 8

3 6 9

> d[,**!**names(d)%in% c("b","c")] **# 특정컬럼을 제외시켜서 뽑을 수 있다. b,c제외**

[1] 1 2 3

> d[,-c(2,3)] **#두번째 세번째 컬럼을 빼고 리턴 b,c제외**

[1] 1 2 3

**행 추가**

> d[4,]<-c(7,7,7)

> d

a b c

1 1 4 7

2 2 5 8

3 3 6 9

4 7 7 7

**행 삭제**

> d<-d[-4,]

> d

a b c

1 1 4 7

2 2 5 8

3 3 6 9

x<-data.frame(x=1:1000)

x

1부터 1000까지 보여줘

**> head(x) #앞부분만 보여줌**

x

1 1

2 2

3 3

4 4

5 5

6 6

**> tail(x) #뒷부분만 보여줌**

x

995 995

996 996

997 997

998 998

999 999

1000 1000

[문제31] 아래와 같은 모양의 변수를 생성하세요. 변수 이름은 df로 하세요.

<화면출력>

> df

name sql plsql

1 king 96 75

2 smith 82 91

3 jane 78 86

df<-data.frame("name"=c("king","smith","jane"),

"sql"=c(96, 82, 78),

"plsql"=c(75,91,86),

**stringsAsFactors = FALSE**)>

df

name sql plsql

1 king 96 75

2 smith 82 91

3 jane 78 86

[문제32] df변수에 james, 90, 80 추가 해주세요.

<화면출력>

> df

name sql plsql

1 king 96 75

2 smith 82 91

3 jane 78 86

4 james 90 80

df[4,]<-c('james',90,80)

또는

df<-rbind(df,c('james',90,80))

또는

df <- rbind(df,data.frame(name='james',sql=90,plsql=80)

[문제33] james에 대한 row 정보만 출력하세요.

<화면출력>

name sql plsql r

4 james 90 80 60

> df$name=='james'

[1] FALSE FALSE FALSE TRUE

> df[df$name=='james',]

name sql plsql

4 james 90 80

#컴마뒤에 원하는 컬럼만보고싶다. 이렇게써줌

> df[df$name=='james',c(1,3)]

name plsql

4 james 80

R 데이터 자료형

1. vector : 같은 데이터 타입을 갖는 1차원 배열

2. list: : 서로 다른 데이터 타입을 갖는 1차원배열, 중첩가능

3. matrix : 같은 데이터 타입을 갖는 2차원 배열

4. array : 같은 데이터 타입을 갖는 3차원 배열

5. factor : 목록, 범주형 데이터

6. data.frame : 서로 다른 데이터 타입을 갖는 컬럼으로 이루어진 2차원 배열

7. table : data.frame과 동일한 구조를 갖는데 속도가 빠르다.

> a<- c(1,2)

> b<- list(c('king', 100))

> c<- matrix(c(1,2))

> d <- array(1:12, dim=c(2,2,3))

> e <- factor(c('male', 'female'))

> f<- data.frame(x=c(1,2))

> class(a)

[1] "numeric"

> class(b)

[1] "list"

> class(c)

[1] "matrix"

> class(d)

[1] "array"

> class(e)

[1] "factor"

> class(f)

[1] "data.frame"

> mode(a)

[1] "numeric"

> mode(b)

[1] "list"

> mode(c)

[1] "numeric"

> mode(d)

[1] "numeric"

> mode(e)

[1] "numeric"

> mode(f)

[1] "list"

mode는 조금 다르다. 매트릭스와 어레이가

class는 명확히 구조를 보여주고

mode는 그구조안에 들어가있는 타입을 보여준다.

> str(f)

'data.frame': 2 obs. of 1 variable:

$ x: num 1 2

구조를보여줌

**> is.numeric(a)**

[1] TRUE

> is.character(a)

[1] FALSE

> is.integer(a)

[1] FALSE

**벡터값을 확인할 때 나름 이 3가지는 기억해야한다.**

> is.list(b)

[1] TRUE

> is.matrix(c)

[1] TRUE

> is.factor(e)

[1] TRUE

> is.array(d)

[1] TRUE

> is.data.frame(f)

[1] TRUE

조건제언문에서 이런자료일땐 이렇게처리 저런자료일땐 저렇게처리.->boolean

**as.**

형변환

메쏘드가 너무 많아서 다기억할 수 없다. 이런게있구나 생각을 가지고 자주쓰는것은 기억하면서 헬프기능을 활용하자.

emp.csv파일을 C:\data로 넣어보자.

**R로 csv파일을 읽어들여보자. csv import**

**read.csv**

- csv파일을 데이터프레임으로 읽어들이는 함수

물리적으로 어느디렉토리에있는지 알아야한다.

**getwd()**

[1] "C:/Users/stu/Documents"

**> read.csv**

function (file, header = TRUE, sep = ",", quote = "\"", dec = ".",

fill = TRUE, comment.char = "", ...)

read.table(file = file, header = header, sep = sep, quote = quote,

dec = dec, fill = fill, comment.char = comment.char, ...)

<bytecode: 0x0000000009b72b98>

<environment: namespace:utils>

**> getwd()**

[1] "C:/Users/stu/Documents"

**> setwd("c:/data") #물리적주소세팅**

**> emp<- read.csv("emp.csv") #emp 불러왔음**

**> str(emp)**

'data.frame': 107 obs. of 11 variables:

$ EMPLOYEE\_ID : int 198 199 200 201 202 203 204 205 206 100 ...

$ FIRST\_NAME : Factor w/ 91 levels "Adam","Alana",..: 21 22 39 59 66 83 31 76 90 80 ...

$ LAST\_NAME : Factor w/ 102 levels "Abel","Ande",..: 69 37 101 41 28 63 5 42 36 50 ...

$ EMAIL : Factor w/ 107 levels "ABANDA","ABULL",..: 24 20 52 67 77 93 32 89 105 90 ...

$ PHONE\_NUMBER : Factor w/ 107 levels "011.44.1343.329268",..: 98 99 36 40 62 41 44 42 43 37 ...

$ HIRE\_DATE : int 20070621 20080113 20030917 20040217 20050817 20020607 20020607 20020607 20020607 20030617 ...

$ JOB\_ID : Factor w/ 19 levels "AC\_ACCOUNT","AC\_MGR",..: 17 17 3 10 11 8 12 2 1 4 ...

$ SALARY : int 2600 2600 4400 13000 6000 6500 10000 12008 8300 29040 ...

$ COMMISSION\_PCT: num NA NA NA NA NA NA NA NA NA NA ...

$ MANAGER\_ID : int 124 124 101 100 201 101 101 101 205 NA ...

$ DEPARTMENT\_ID : int 50 50 10 20 20 40 70 110 110 90 ...

문제점: factor형으로되어있음 모든 고유한 이름들이 레벨로되어있다.

**> emp<-read.csv("emp.csv",header=F)**

**> str(emp)**

'data.frame': 108 obs. of 11 variables:

$ V1 : Factor w/ 108 levels "100","101","102",..: 108 99 100 101 102 103 104 105 106 107 ...

$ V2 : Factor w/ 92 levels "Adam","Alana",..: 26 21 22 40 60 67 84 32 77 91 ...

$ V3 : Factor w/ 103 levels "Abel","Ande",..: 55 70 37 102 41 28 64 5 42 36 ...

$ V4 : Factor w/ 108 levels "ABANDA","ABULL",..: 28 24 20 53 68 78 94 33 90 106 ...

$ V5 : Factor w/ 108 levels "011.44.1343.329268",..: 108 98 99 36 40 62 41 44 42 43 ...

$ V6 : Factor w/ 99 levels "20010113","20020607",..: 99 81 90 10 15 38 2 2 2 2 ...

$ V7 : Factor w/ 20 levels "AC\_ACCOUNT","AC\_MGR",..: 10 18 18 3 11 12 8 13 2 1 ...

$ V8 : Factor w/ 59 levels "10000","10500",..: 59 15 15 32 7 35 39 1 6 52 ...

$ V9 : Factor w/ 9 levels "","0.1","0.15",..: 9 1 1 1 1 1 1 1 1 1 ...

$ V10: Factor w/ 20 levels "","100","101",..: 20 12 12 3 2 18 3 3 3 19 ...

$ V11: Factor w/ 13 levels "","10","100",..: 13 8 8 2 5 5 7 10 4 4 ...

>

csv파일 첫줄에 컬럼이름이 있다면 꼭 header를 TRUE로해줘야한다.

**> emp<-read.csv("emp.csv",header=T)**

**> str(emp)**

'data.frame': 107 obs. of 11 variables:

$ EMPLOYEE\_ID : int 198 199 200 201 202 203 204 205 206 100 ...

$ FIRST\_NAME : Factor w/ 91 levels "Adam","Alana",..: 21 22 39 59 66 83 31 76 90 80 ...

$ LAST\_NAME : Factor w/ 102 levels "Abel","Ande",..: 69 37 101 41 28 63 5 42 36 50 ...

$ EMAIL : Factor w/ 107 levels "ABANDA","ABULL",..: 24 20 52 67 77 93 32 89 105 90 ...

$ PHONE\_NUMBER : Factor w/ 107 levels "011.44.1343.329268",..: 98 99 36 40 62 41 44 42 43 37 ...

$ HIRE\_DATE : int 20070621 20080113 20030917 20040217 20050817 20020607 20020607 20020607 20020607 20030617 ...

$ JOB\_ID : Factor w/ 19 levels "AC\_ACCOUNT","AC\_MGR",..: 17 17 3 10 11 8 12 2 1 4 ...

$ SALARY : int 2600 2600 4400 13000 6000 6500 10000 12008 8300 29040 ...

$ COMMISSION\_PCT: num NA NA NA NA NA NA NA NA NA NA ...

$ MANAGER\_ID : int 124 124 101 100 201 101 101 101 205 NA ...

$ DEPARTMENT\_ID : int 50 50 10 20 20 40 70 110 110 90 ...

문자는 문자형으로 보겠다.

**> emp<-read.csv("emp.csv",header=T,stringsAsFactors=F)**

**> str(emp)**

'data.frame': 107 obs. of 11 variables:

$ EMPLOYEE\_ID : int 198 199 200 201 202 203 204 205 206 100 ...

$ FIRST\_NAME : chr "Donald" "Douglas" "Jennifer" "Michael" ...

$ LAST\_NAME : chr "OConnell" "Grant" "Whalen" "Hartstein" ...

$ EMAIL : chr "DOCONNEL" "DGRANT" "JWHALEN" "MHARTSTE" ...

$ PHONE\_NUMBER : chr "650.507.9833" "650.507.9844" "515.123.4444" "515.123.5555" ...

$ HIRE\_DATE : int 20070621 20080113 20030917 20040217 20050817 20020607 20020607 20020607 20020607 20030617 ...

$ JOB\_ID : chr "SH\_CLERK" "SH\_CLERK" "AD\_ASST" "MK\_MAN" ...

$ SALARY : int 2600 2600 4400 13000 6000 6500 10000 12008 8300 29040 ...

$ COMMISSION\_PCT: num NA NA NA NA NA NA NA NA NA NA ...

$ MANAGER\_ID : int 124 124 101 100 201 101 101 101 205 NA ...

$ DEPARTMENT\_ID : int 50 50 10 20 20 40 70 110 110 90 ...

**R에서의 NA는 sql에서의 NULL와 똑같다. 아직할당하지않음.**

**> names(emp) #emp의 컬럼확인**

[1] "EMPLOYEE\_ID" "FIRST\_NAME" "LAST\_NAME" "EMAIL" "PHONE\_NUMBER" "HIRE\_DATE"

[7] "JOB\_ID" "SALARY" "COMMISSION\_PCT" "MANAGER\_ID" "DEPARTMENT\_ID"

**> emp$EMPLOYEE\_ID**

[1] 198 199 200 201 202 203 204 205 206 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118

[29] 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146

[57] 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174

[85] 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197

**> emp$EMPLOYEE\_ID==100**

[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[19] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[55] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[91] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

**같다같다 두번쓰면 요렇게**

emp라는 것안에 이값이 있는 부분을 뽑는 것

,열표현

**> emp[emp$EMPLOYEE\_ID==100,]**

EMPLOYEE\_ID FIRST\_NAME LAST\_NAME EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID SALARY COMMISSION\_PCT MANAGER\_ID

10 100 Steven King SKING 515.123.4567 20030617 AD\_PRES 29040 NA NA

DEPARTMENT\_ID

10 90

**이것과 셀렉문을 비교해보자.**

select last\_name, salary from emp where employee\_id = 100;

emp[emp$EMPLOYEE\_ID==100, c("LAST\_NAME","SALARY")]

---- ----------------------------- -----------------------------

> emp[emp$EMPLOYEE\_ID==100, c("LAST\_NAME","SALARY")]

LAST\_NAME SALARY

10 King 29040

[문제34]. emp 변수에 있는 데이터 중에 급여가 3000 인 사원들의 last\_name, salary를 출력하세요. 단 emp 변수에 컬럼정보를 확인하시고 수행하세요.

names(emp)

str(emp)

emp[emp$SALARY==3000, c("LAST\_NAME","SALARY")]

[문제35] 급여가 2000 이상인 사원들의 last\_name, salary를 출력하세요.

emp[emp$SALARY>=2000, c("LAST\_NAME","SALARY")]

[문제36] job이 ST\_CLERK인 사원들의 이름과 월급과 직업을 출력하세요.

emp[emp$JOB\_ID=="ST\_CLERK", c("LAST\_NAME","SALARY","JOB\_ID")]

[문제37] job이 ST\_CLERK이 아닌 사원들의 이름과 월급과 직업을 출력하세요.

emp[emp$JOB\_ID!="ST\_CLERK", c("LAST\_NAME","SALARY","JOB\_ID")]

[문제38] 오라클의 in 연산자와 비슷한 R연산자는?

%in%

[문제39] job이 AD\_ASST, MK\_MAN 인 사원들의 employee\_id,last\_name,job\_id를 출력하세요.

emp[emp$JOB\_ID %in% c("AD\_ASST","MK\_MAN"),c("EMPLOYEE\_ID","LAST\_NAME","JOB\_ID")]

[문제40] job이 ST\_CLERK, SH\_CLERK, SA\_REP 아닌 사원들의 employee\_id,last\_name,job\_id를 출력하세요.

emp[!emp$JOB\_ID %in% c("ST\_CLERK","SH\_CLERK","SA\_REP"),c("EMPLOYEE\_ID","LAST\_NAME","JOB\_ID")]

[문제41] 부서번호가 10번,20번인 사원들의 last\_name, salary, department\_id를 출력하세요.

emp[emp$DEPARTMENT\_ID %in% c(10,20),c("LAST\_NAME","SALARY","DEPARTMENT\_ID")]

[문제42] 오라클의 연결 연산자 와 비슷한 R 연산자는?

오라클 R

|| paste(,,,)

[문제43] 아래결과와 같이 출력되도록하세요.

Grant 의 직업은 SH\_CLERK 입니다.

paste(emp[,"LAST\_NAME"], "의 직업은 ", emp[,"JOB\_ID"],"입니다.")

[문제42] 오라클의 연결 연산자 와 비슷한 R 연산자는?

오라클 R

|| paste(,,,)

[문제43] 아래결과와 같이 출력되도록하세요.

Grant 의 직업은 SH\_CLERK 입니다.

paste(emp[,"LAST\_NAME"], "의 직업은 ", emp[,"JOB\_ID"],"입니다.")

paste(emp$LAST\_NAME,'의 직업은 ', emp$JOB\_ID, '입니다.')

paste0(emp$LAST\_NAME,'의 직업은 ', emp$JOB\_ID, '입니다.')

paste(emp$LAST\_NAME,'의 직업은 ', emp$JOB\_ID, '입니다.', sep="")

paste(emp$LAST\_NAME,'의 직업은 ', emp$JOB\_ID, '입니다.', sep=",")

emp$LAST\_NAME

emp$JOB\_ID

[문제44] R에서 NA(결측치)를 체크하는 함수는?

is.na()

!is.na()

[문제45] commission\_pct에 NA 인 사원들의 last\_name, salary, commission\_pct를 출력하세요.

emp[is.na(emp$COMMISSION\_PCT),c("LAST\_NAME", "SALARY", "COMMISSION\_PCT")]

[문제46] department\_id에 NA 인 사원들의 last\_name, salary, department\_id를 출력하세요.

emp[is.na(emp$DEPARTMENT\_ID),c("LAST\_NAME", "SALARY", "DEPARTMENT\_ID")]

[문제47] commission\_pct에 NA가 아닌 사원들의 last\_name, salary, commission\_pct를 출력하세요.

emp[!is.na(emp$COMMISSION\_PCT),c("LAST\_NAME", "SALARY", "COMMISSION\_PCT")]

[문제48] 30번 부서 사원들이면서 급여는 3000이상인 사원들의 employee\_id, salary, department\_id를 출력하세요.

emp[emp$DEPARTMENT\_ID==30&emp$SALARY>=3000, c("EMPLOYEE\_ID","SALARY","DEPARTMENT\_ID")]

24 114 11000 30

25 115 3100 30

NA NA NA NA

NA가 나오는게이상함.

NA가 있는 row를 제거하고싶을 때 na.omit

na.omit(emp[emp$DEPARTMENT\_ID==30&emp$SALARY>=3000, c("EMPLOYEE\_ID","SALARY","DEPARTMENT\_ID")])

비정형데이터이면서 제약조건이 없음.

N/A값이 하나라도 있으면 NA 로우가나온다. 혹시나 후에 이곳에 그 값이 있을 수 있다는 뜻.

[문제49] 20번부서 사원이면서 급여는 10000를 초과한 사원 또는 급여가 2500 미만의 사원들의 employee\_id, salary, department\_id를 출력하세요.

emp[(emp$DEPARTMENT\_ID==20&emp$SALARY>10000)|emp$SALARY<2500, c("EMPLOYEE\_ID","SALARY","DEPARTMENT\_ID")]

**■ grep함수 : 문자 패턴을 찾을 때 사용되는 함수 (sql에서의 like연산자)**

**^ : 첫번째**

**$ : 마지막**

**. : 한자리수**

**\* : wild card(%)**

help(grep)

<예>

emp[grep(**"aa"**, emp$LAST\_NAME),c("LAST\_NAME","SALARY")] #위치상관없이 aa가있는거모두

emp[grep(**"^aa"**, emp$LAST\_NAME),c("LAST\_NAME","SALARY")] #맨앞에 aa인

emp[grep(**"aa$"**, emp$LAST\_NAME),c("LAST\_NAME","SALARY")] #맨뒤가 aa

emp[grep("**[x-z]**", emp$LAST\_NAME, ignore.case = TRUE),c("LAST\_NAME","SALARY")]

# x, y, z 하나라도 들어가는애들

ignore.case = TRUE 대소문자 구분안한다.

ignore.case = FALSE 대소문자 구분한다.

[문제50]last\_name의 첫번째 글자가 A 로 시작하는 사원들의 last\_name, salary를 출력하세요.

emp[grep("^A.\*", emp$LAST\_NAME),c("LAST\_NAME","SALARY")]

[문제51]last\_name의 끝글자가 g 로 끝나는 사원들의 last\_name, salary를 출력하세요.

emp[grep("g$", emp$LAST\_NAME),c("LAST\_NAME","SALARY")]

[문제52]last\_name의 z 를 포함하고 있는 사원들의 last\_name, salary를 출력하세요.

emp[grep("z", emp$LAST\_NAME),c("LAST\_NAME","SALARY")]

emp[grep("^.\*.z.\*$", emp$LAST\_NAME),c("LAST\_NAME","SALARY")]

이두가지가같음

[문제53]last\_name의 두번째 철자가 u 인 사원들의 last\_name, salary를 출력하세요.

답1

emp[grep("^.u",emp$LAST\_NAME,ignore.case=FALSE),c("LAST\_NAME","SALARY")]

답2

emp[grep ('^.u.\*$', emp$LAST\_NAME,ignore.case=FALSE)) ,c("LAST\_NAME","SALARY")]

**문자함수**

**1. nchar : 문자수를 리턴하는 함수 (오라클의 length)**

> nchar('R Developer') #공백포함 11

[1] 11

> nchar('R Developer',type="chars") #글자수로 표현 (기본값)

[1] 11

> nchar('R Developer', type ="bytes") # 바이트수로 (바이트단위로볼때 타입조정)

[1] 11

> nchar('빅데이터')

[1] 4

> nchar('빅데이터',type="chars")

[1] 4

> nchar('빅데이터', type ="bytes")

[1] 8

**2. strsplit : 부분문자로 분리하는 함수**

strsplit('R Developer')

strsplit('R Developer',split=character(0))

**> strsplit('R Developer')**

Error in strsplit("R Developer") :

argument "split" is missing, with no default #**옵션이 필수인 메소드들도있다**.

**> strsplit('R Developer',split=character(0))**

[[1]]

[1] "R" " " "D" "e" "v" "e" "l" "o" "p" "e" "r"

**> strsplit('R Developer', split=" ") #공백문자를 기준으로 스플릿**

[[1]]

[1] "R" "Developer"

**> strsplit('R Developer', split=",") #컴마를기준으로 스플릿**

[[1]]

[1] "R Developer"

**> str(strsplit('R Developer', split=",")) #구조가 리스트형**

List of 1

$ : chr "R Developer"

**> str(strsplit('R Developer',split = character(0))) #구조가 리스트형**

List of 1

$ : chr [1:11] "R" " " "D" "e" ...

**> str(unlist(strsplit('R Developer',split = character(0)))) #list 형에서 벡터로바꿔줌**

chr [1:11] "R" " " "D" "e" "v" "e" "l" "o" "p" "e" "r"

> unlist(strsplit('R Developer', split=","))

[1] "R Developer"

> str(unlist(strsplit('R Developer', split=",")))

chr "R Developer"

> unlist(strsplit('R Developer',split = character(0)))

[1] "R" " " "D" "e" "v" "e" "l" "o" "p" "e" "r"

> str(unlist(strsplit('R Developer',split = character(0))))

chr [1:11] "R" " " "D" "e" "v" "e" "l" "o" "p" "e" "r"

**빈도수 체크할 때 유용하다.**

**알파벳빈도수->한글자씩 뽑아내야함**

**3. toupper :대문자**

**> toupper('r developer')**

**[1] "R DEVELOPER"**

**4.tolower: 소문자**

> tolower('R DEVELOPER')

[1] "r developer"

**5. substr: 문자열 추출 substr(' ',시작점, 끝점)** 오라클 sub이랑 다르다.

> substr('R Developer',1,1)

[1] "R"

> substr('123456789',1,1)

[1] "1"

> substr('123456789',1,2)

[1] "12"

> substr('123456789',5,5)

[1] "5"

[문제54] developer글자를 첫글자 대문자, 뒤글자는 소 문자로 변환하세요.

paste(toupper(substr('developer',1,1)),substr ('developer',2,nchar('developer')),sep="")

library(tools)

toTitleCase("developer")

library(stringr)

str\_to\_title('developer')

**누가이미 패키지를 만들어놨다. ->인스톨이필요하다.**

**install.packages("stringr")**

> **library(tools)**

**> toTitleCase("developer")**

[1] "Developer"

**> library(stringr)**

**> str\_to\_title('developer')**

[1] "Developer"

**6. sub( ) : 첫번째 일치하는 문자만 바꾸는 함수**

첫번째: 찾아야할 문자 세번째원본글자 안에서 찾아야할 대상

두번째: 바꿀글자

세번째: 원본글자

첫번째일치하는 글자만 바꿈

**> sub('R', 'Python', 'R Programmer R Developer')**

[1] "Python Programmer R Developer"

'R Programmer R Developer' -> "Python Programmer R Developer"

**7. gsub : 일치하는 문자 모두 바꾸는 함수**

> gsub ('R', 'Python', 'R Programmer R Developer')

[1] "Python Programmer Python Developer"

> gsub('[0-2]','\*','120304')

[1] "\*\*\*3\*4"

**숫자함수**

**1. round 반올림**

> round(45.9226)

[1] 46

> round(45.9226,**1**)

[1] 45.9

> round(45.9226,**2**)

[1] 45.92

> round(45.9226,**3**)

[1] 45.923

> round(45.9226,0)

[1] 46

> round(45.9226,**-1**)

[1] 50

**2. trunc**

> trunc(45.926)

[1] 45

> trunc(45.926,1)

[1] 45

> trunc(45.926,2)

소수점 인수값이 의미가 없다.

**3. singif**

> signif(45.9261)

[1] 45.9261

> signif(45.9261,1) 첫번째글자4를 기준으로 다음글자에서 반올림해

[1] 50

> signif(45.9261,2) 두번째글자2를 기준으로 다음글자에서 반올림해

[1] 46

> signif(45.9261,3)

[1] 45.9

> signif(45.9261,4)

[1] 45.93

**4. floor : 소수점 완전절삭 버림**

> floor(45.926)

[1] 45

인수값없음

[문제55] last\_name의 글자의 수가 10이상인 사원의 employee\_id, last\_name 출력하세요.

emp[nchar(emp$LAST\_NAME)>=10,c("EMPLOYEE\_ID", "LAST\_NAME")]

[문제56] last\_name, last\_name의 첫번째 철자부터 세번째 철자까지 함께 출력하세요.

paste(emp$LAST\_NAME, substr(emp$LAST\_NAME,1,3))

[문제57] last\_name의 두번째 철자가 m 인 사원들의 last\_name, salary를 출력하세요.

emp[substr(emp$LAST\_NAME,2,2)=='m',c('LAST\_NAME','SALARY')]

[문제58] last\_name의 두번째 철자가 m 또는 g 인 사원들의 last\_name, salary를 출력하세요.

emp[substr(emp$LAST\_NAME,2,2)=='m'|substr(emp$LAST\_NAME,2,2)=='g',c('LAST\_NAME','SALARY')]

emp[substr(emp$LAST\_NAME,2,2)%in%c('m',"g"),c('LAST\_NAME','SALARY')]

[문제59] last\_name, salary값을 화면에 출력할때 0은 \* 로 출력하세요.

paste(emp$LAST\_NAME,gsub('0','\*',emp$SALARY))

[문제60] last\_name의 제일 뒷글자만 대문자 앞글자들은 소문자로 출력하세요.

paste(tolower(substr(emp$LAST\_NAME,1,nchar(emp$LAST\_NAME)-1)), toupper(substr(emp$LAST\_NAME, nchar(emp$LAST\_NAME),nchar(emp$LAST\_NAME))),sep="")

[문제61] s 백터 변수를 생성해서 "BIG DATA MARKETING" 입력한 후 단어로 분리해주세요.

s<-"BIG DATA MARKETING"

strsplit(s,split=" ")

**■날짜함수**

**1.현재 날짜 시간**

Sys.Date()

Sys.time()

date()

날짜는 데이트형이기는 하지만 그 내부적으로는 숫자값으로 저장되는 형식이다.

**2. 형을바꿔주는 함수**

as.Date : 문자날짜를 날짜형으로 변환하는 함수

as.Date('2018-07-26')

as.Date('2018/07/26')

as.Date(d'20180726',format='%Y%m%d')

> as.Date('2018-07-25')

[1] "2018-07-25"

> as.Date('2018/07/25')

[1] "2018-07-25"

이 두개가 기본요소

> as.Date('20180725')

Error in charToDate(x) : 문자열이 표준서식을 따르지 않습니다

이런형식으로 만들고싶으면 알에서도 날짜요소를 포함시켜줘야한다.

> as.Date('20180725', format='%Y%m%d')

[1] "2018-07-25"

**3. format 날짜모델요소**

%Y : 세기를 포함한 년도(4자리)

%y : 세기를 생략한 년도(2자리)

%m : 숫자달

%B : 문자달

%b : 문자달의 약어

%d : 일

%A : 요일

%a : 요일의 약어

%u : 숫자요일 1~7 월요일이 1 일요일이 7

%w : 숫자요일 0~6 일요일이 0 월요일이 1

%H : 시

%M : 분

%S : 초

%z : timezone 시간

%Z : timezone name

> format(Sys.time(),'%y%m%d %z%Z')

[1] "180726 +0900KST"

> format(Sys.time(),'%A%a%u%w')

[1] "목요일목44"

**4. weekdays : 요일을 출력하는 함수**

weekdays(Sys.Date())

> weekdays(as.Date('1988/04/10'))

[1] "일요일"

> weekdays(as.Date('1988-04-10',format='%Y-%m-%d'))

[1] "일요일"

format(as.Date('1988/04/10'), '%A')

**5. 날짜 계산**

Sys.Date() + 100 날짜+숫자

Sys.Date() - 206 날짜-숫자

as.Date('2018-07-26', format = '%y-%m-%d')

as.Date('2018-07-26', format = '%Y-%m-%d') +120

> as.Date('2018-11-22', format= '%Y-%m-%d') - as.Date('2018-05-24',format='%Y-%m-%d')

Time difference of 182 days

> **as.numeric(**as.Date('2018-11-22', format= '%Y-%m-%d') - as.Date('2018-05-24',format='%Y-%m-%d')**)**

[1] 182

**6. dfftime함수 : 두 날짜간의 일수를 표현하는 함수**

difftime('2018-11-22',Sys.Date())

원래는 형변환 함수를 써야하지만 잘돌아갔다. 쓰는 습관을 갖자.!!

> difftime('2018-11-22',Sys.Date())

Time difference of 118.625 days

> as.numeric(difftime('2018-11-22',Sys.Date()))

[1] 118.625

**7. as.difftime 함수 : 시간의 차이 (형변환함수라고생각하면안된다)**

> as.difftime('09:30:00')-as.difftime('18:20:00')

Time difference of -8.833333 hours

> as.difftime('18:20:00')-as.difftime('09:30:00')

Time difference of 8.833333 hours

시분초다넣어줘야한다.

**8. lubridate**

install.packages("lubridate") #R 커맨드라인에서 설치

library(lubridate)

로컬에 패키지저장되서 계속 로드할 수있다.

쓸때 로드해야한다.

c.f) 이 패키지를 로드시키지않으면 사용할 수없다.

#현재시간

> now()

[1] "2018-07-26 12:05:52 KST"

#년도만뽑아내기

> year(now())

[1] 2018

# 달 숫자뽑아냄

> month(now())

[1] 7

> date<-now()

> month(date)

[1] 7

> month(date,label=T)

[1] 7

Levels: 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < 10 < 11 < 12

(factor형으로 되어있다.)

> month(date, label=F)

[1] 7

> format(Sys.time(),'%m')

[1] "07"

#일

> day(now())

[1] 26

> day(date)

[1] 26

> format(Sys.time(),'%d')

[1] "26"

#요일

format(Sys.time(),'%A')

wday(now())

wday(now(),week\_start=1) #월요일 기준

wday(no(),week\_star=7) #일요일 기준

> format(Sys.time(),'%A')

[1] "목요일"

> wday(now())

[1] 5

**> wday(now()) #목요일인데 5로나옴 일요일부터 1**

**[1] 5**

> format(Sys.time(),'%u')

[1] "4"

> format(Sys.time(),'%w')

[1] "4"

> wday(now(),week\_start=7,label=T)

[1] 목

Levels: 일 < 월 < 화 < 수 < 목 < 금 < 토

> years(10)

[1] "10y 0m 0d 0H 0M 0S"

> months(100)

[1] "100m 0d 0H 0M 0S"

오라클의 to\_yminterval('10-00')

**> now()+years(10)**

[1] "2028-07-26 14:05:50 KST"

**> now() +months(100)**

[1] "2026-11-26 14:05:59 KST"

**> now() +days(100)**

[1] "2018-11-03 14:07:18 KST"

**> now() + hours(3)**

[1] "2018-07-26 17:09:29 KST"

**> now() + minutes(100)**

[1] "2018-07-26 15:50:07 KST"

**> now() +seconds(100)**

[1] "2018-07-26 14:11:55 KST"

**> now() +years(1) + months(1) +days(1)+hours(10) + minutes(20) +seconds(60)**

[1] "2019-08-28 00:33:15 KST"

**> hm('08:00')**

[1] "8H 0M 0S"

**> now() + hm('08:00') #날짜에 시간을더하는 방법**

[1] "2018-07-26 22:13:02 KST"

**> now()+hms('02:30:59') #날짜에 시간을더하는 방법**

[1] "2018-07-26 16:44:15 KST"

시개월쪽 분석하러가면 많이쓴다.

> date<- now()

> date

[1] "2018-07-26 14:16:47 KST"

> year(date) <-2017 #년도수정

> date

[1] "2017-07-26 14:16:47 KST"

> month(date)<-1 #달 수정

> date

[1] "2017-01-26 14:16:47 KST"

> day(date)<-1 #일 수정

> hour(date)<-01 #시 수정

> minute(date)<- 00 #분 수정

> second(date)<- 00 #초 수정

> date

[1] "2017-01-01 01:00:00 KST"

**9. 분기**

> quarters(Sys.Date())

[1] "Q3" #3분기

integer타입 날짜를 바로 날짜로 바꿀 수 없으므로 as.character로 변환한다음 as.date로 날짜로다시변환한다.

[문제62] 2002-06-07에 입사한 사원들의 last\_name, hire\_date를 출력하세요.

> emp <-read.csv("c:/data/emp.csv",header=T, stringsAsFactors = FALSE)

> str(emp)

'data.frame': 107 obs. of 11 variables:

$ EMPLOYEE\_ID : int 198 199 200 201 202 203 204 205 206 100 ...

$ FIRST\_NAME : chr "Donald" "Douglas" "Jennifer" "Michael" ...

$ LAST\_NAME : chr "OConnell" "Grant" "Whalen" "Hartstein" ...

$ EMAIL : chr "DOCONNEL" "DGRANT" "JWHALEN" "MHARTSTE" ...

$ PHONE\_NUMBER : chr "650.507.9833" "650.507.9844" "515.123.4444" "515.123.5555" ...

$ HIRE\_DATE : int 20070621 20080113 20030917 20040217 20050817 20020607 20020607 20020607 20020607 20030617 ...

$ JOB\_ID : chr "SH\_CLERK" "SH\_CLERK" "AD\_ASST" "MK\_MAN" ...

$ SALARY : int 2600 2600 4400 13000 6000 6500 10000 12008 8300 29040 ...

$ COMMISSION\_PCT: num NA NA NA NA NA NA NA NA NA NA ...

$ MANAGER\_ID : int 124 124 101 100 201 101 101 101 205 NA ...

$ DEPARTMENT\_ID : int 50 50 10 20 20 40 70 110 110 90 ...

emp[emp$HIRE\_DATE==20020607, c("LAST\_NAME","JOB\_ID", "HIRE\_DATE")]

emp[as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d') == as.Date('2002-06-07'), c("LAST\_NAME","JOB\_ID", "HIRE\_DATE")]

[문제63] 사원의 last\_name, 근무일수를 출력하세요.

data.frame(last\_name = emp$LAST\_NAME, working\_days=as.numeric(Sys.Date()-as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d')))

data.frame(last\_name = emp$LAST\_NAME, working\_days=as.numeric(difftime(Sys.Date(),as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d'))))

[문제64] 사원의 last\_name, 입사한 요일을 출력하세요.

emp$HIRE\_DATE <- as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d')

data.frame(emp$LAST\_NAME, weekdays(emp$HIRE\_DATE))

[문제65] 오늘 날짜를 기준으로 100개월 되는 날짜의 요일을 출력하세요.

> format( Sys.Date() + months(100),'%A')

> weekdays( Sys.Date() + months(100))

> wday(Sys.Date() + months(100),label=T)

-----------------------------------------------------------------------------------------------------------------------------------------------------

class(Sys.Date())

mode(Sys.Date())

> class(Sys.Date())

[1] "Date"

> mode(Sys.Date())

[1] "numeric"

> class(Sys.time())

[1] **"POSIXct" "POSIXt"**

**POSIX(Portable Operating System Interface)**

**UNIX 간 소통 가능한 프로그램 인터페이스의 규약**

O.S들끼리 날짜를 다루는 형식이다르면 문제가되니까..

POSIX**ct** (continuous)

POSIX**t** (posixlt) (list time)

r은 날짜 시간 데이터를 이 두가지 POSIX**ct** (continuous), POSIX**t** (posixlt) (list time)클래스를 사용한다.

> Sys.time()

[1] "2018-07-26 16:13:20 KST"

> as.numeric(Sys.time())

[1] 1532589200

> time<- as.POSIXlt(Sys.time())

> time #벡터처럼보이지만 리스트형이다.

[1] "2018-07-26 16:14:40 KST"

> unlist(time) #리스트풀어보자.

sec min hour mday mon

"40.6440498828888" "14" "16" "26" "6"

year wday yday isdst zone

"118" "4" "206" "0" "KST"

gmtoff

"32400"

sec : 초

min : 분

hour : 시

mday : 그달의 일

mon : 1월을 0으로 시작한다.

year : 1900년을 0으로 1900 + 118= 2018년 (유닉스공부하면 유닉스의 시간정보가 이런식으로 되어있다는 것을배움)

wday : 일요일을 0으로 기준 %u랑 같음

yday : 1월 1일을 0으로 기준으로 하여 며칠까지왔는지

isdst : 서머타임(각지역마다 서머타임설정, 0은 서머타임 설정 x)

zone : timezone name

gmtoff : timezone 시(초) 한국은 + 9지역 (9시간 \* 60분 \* 60초 그릴천문대를 기준으로 )

> date <- '2018-07-26'

> as.Date(date,format='%Y-%m-%d')

[1] "2018-07-26"

> class(as.Date(date,format='%Y-%m-%d'))

[1] "Date"

> strptime(date,format='%Y-%m-%d')

[1] "2018-07-26 KST"

> class(strptime(date,format='%Y-%m-%d'))

[1] "POSIXlt" "POSIXt"

strptime()형변환함수는 POSIX형식으로바꿔준다.

**■중복을 제거하는 방법**

오라클의 distinct

**> unique(emp$JOB\_ID)**

[1] SH\_CLERK AD\_ASST MK\_MAN MK\_REP HR\_REP PR\_REP AC\_MGR AC\_ACCOUNT

[9] AD\_PRES AD\_VP IT\_PROG FI\_MGR FI\_ACCOUNT PU\_MAN PU\_CLERK ST\_MAN

[17] ST\_CLERK SA\_MAN SA\_REP

19 Levels: AC\_ACCOUNT AC\_MGR AD\_ASST AD\_PRES AD\_VP FI\_ACCOUNT FI\_MGR HR\_REP IT\_PROG MK\_MAN ... ST\_MAN

[문제66] 부서번호를 중복제거해주세요.

**> unique(emp$DEPARTMENT\_ID)**

[1] 50 10 20 40 70 110 90 60 100 30 80 NA

**> na.omit(unique(emp$DEPARTMENT\_ID))**

[1] 50 10 20 40 70 110 90 60 100 30 80

attr(,"na.action")

[1] 12

attr(,"class")

[1] "omit"

**> as.integer(na.omit(unique(emp$DEPARTMENT\_ID)))**

[1] 50 10 20 40 70 110 90 60 100 30 80

중복제거하려면 정렬필요.

**■정렬(sort)**

x<- c(3,2,4,8,6,5,10,NA,1,11,NA,15)

> x<- c(3,2,4,8,6,5,10,NA,1,11,NA,15)

**> sort(x) #오름차순 기본값**

[1] 1 2 3 4 5 6 8 10 11 15

**> sort(x, decreasing=FALSE) #오름차순**

[1] 1 2 3 4 5 6 8 10 11 15

**> sort(x, decreasing=TRUE) #내림차순**

[1] 15 11 10 8 6 5 4 3 2 1

**> sort(x,decreasing=FALSE,na.last=NA)**

[1] 1 2 3 4 5 6 8 10 11 15

**> sort(x,decreasing=FALSE,na.last=NA) #NA 출력안함(기본값)**

[1] 1 2 3 4 5 6 8 10 11 15

**> sort(x,decreasing=FALSE,na.last=TRUE) # NA 마지막에 출력**

[1] 1 2 3 4 5 6 8 10 11 15 NA NA

**> sort(x,decreasing=FALSE,na.last=FALSE)# NA 처음에 출력**

[1] NA NA 1 2 3 4 5 6 8 10 11 15

**> rev(sort(x)) # sort한걸 리버스**

[1] 15 11 10 8 6 5 4 3 2 1

**order :정렬의 색인을 반환(위치정보반환)**

> x<-c(30,50,**10**,40,20)

> sort(x)

[1] **10** 20 30 40 50

> order(x)

[1] **3** 5 1 4 2 #10이 3번째에 ho:소트랑 항상같이써야겠네..

인덱스번호를 리턴

**> x[order(x)]**

[1] 10 20 30 40 50

> x[order(x,decreasing=TRUE,na.last=NA)]

[1] 50 40 30 20 10

> x[order(x)]

[1] 10 20 30 40 50

> x[order(x,decreasing=TRUE,na.last=NA)]

[1] 50 40 30 20 10

> x[order(x, decreasing=TRUE,na.last=TRUE)]

[1] 50 40 30 20 10

> x[order(x, decreasing=FALSE,na.last=FALSE)]

[1] 10 20 30 40 50

**색인으로 바라봐야하면 ORDER를 사용해야한다.**

**doBy : 데이터 프레임에 정렬**

**install.packages("doBy")**

#기존의 소트와 오더로는 데이터프레임 정렬을 할 수 없기 때문에 패키지 인스톨 한다.

**library(doBy)**

**> orderBy(~SALARY,emp[,c("LAST\_NAME","SALARY")]) #샐러리기준 오름차순**

LAST\_NAME SALARY

42 Olson 2100

38 Markle 2200

46 Philtanker 2200

37 Landry 2400

45 Gee 2400

29 Colmenares 2500

41 Marlow 2500

50 Patel 2500

54 Vargas 2500

92 Sullivan 2500

101 Perkins 2500

1 OConnell 2600

2 Grant 2600

28 Himuro 2600

53 Matos 2600

36 Mikkilineni 2700

49 Seo 2700

27 Tobias 2800

.

.

.

> orderBy(~**-**SALARY,emp[,c("LAST\_NAME","SALARY")]) # -표시 내림차순으로 정렬

LAST\_NAME SALARY

10 King 29040

11 Kochhar 17000

12 De Haan 17000

55 Russell 14000

56 Partners 13500

4 Hartstein 13000

8 Higgins 12008

18 Greenberg 12008

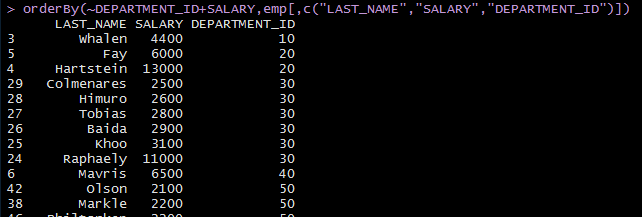
57 Errazuriz 12000

.

.

.

> orderBy(~**DEPARTMENT\_ID+SALARY**,emp[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID")])



.

.

.

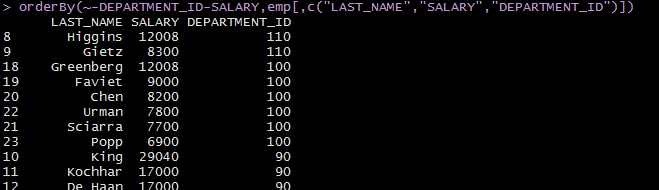
select last\_name, salary, department\_id

from emp

order by department\_id, salary

이거와 같다.

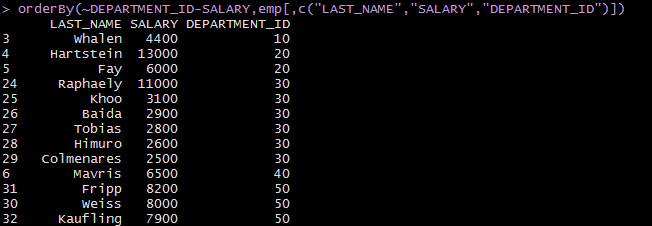
> orderBy(~**-**DEPARTMENT\_ID**-**SALARY,emp[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID")])



내림차순

> orderBy(~DEPARTMENT\_ID-SALARY,emp[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID")])

부서아이디는 오름차순 부서내 샐러리는 내림차순



emp <-read.csv("c:/data/emp.csv",header=T, stringsAsFactors = FALSE)

> str(emp)

'data.frame': 107 obs. of 11 variables:

$ EMPLOYEE\_ID : int 198 199 200 201 202 203 204 205 206 100 ...

$ FIRST\_NAME : chr "Donald" "Douglas" "Jennifer" "Michael" ...

$ LAST\_NAME : chr "OConnell" "Grant" "Whalen" "Hartstein" ...

$ EMAIL : chr "DOCONNEL" "DGRANT" "JWHALEN" "MHARTSTE" ...

$ PHONE\_NUMBER : chr "650.507.9833" "650.507.9844" "515.123.4444" "515.123.5555" ...

$ HIRE\_DATE : int 20070621 20080113 20030917 20040217 20050817 20020607 20020607 20020607 20020607 20030617 ...

$ JOB\_ID : chr "SH\_CLERK" "SH\_CLERK" "AD\_ASST" "MK\_MAN" ...

$ SALARY : int 2600 2600 4400 13000 6000 6500 10000 12008 8300 29040 ...

$ COMMISSION\_PCT: num NA NA NA NA NA NA NA NA NA NA ...

$ MANAGER\_ID : int 124 124 101 100 201 101 101 101 205 NA ...

$ DEPARTMENT\_ID : int 50 50 10 20 20 40 70 110 110 90 ...

>

[문제67] last\_name, hire\_date를 출력하는데 먼저 입사한 사원부터 출력하세요.

**행과 열이 나와야할 때 행에는 인덱스번호가 나와야 하니까 order를 써야한다.**

emp[order(emp$HIRE\_DATE, decreasing=F), c("LAST\_NAME","HIRE\_DATE") ]

emp$HIRE\_DATE <- as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d')

emp[order(emp$HIRE\_DATE, decreasing=F), c("LAST\_NAME","HIRE\_DATE") ]

[문제68] df이름의 data frame 변수를 선언합니다.

id 컬럼의 값은 100,101,102,103,104,

weight 컬럼의 값은 60,90,75,95,65,

size 컬럼의 값은 small, large, medium,large,small 값으로 생성하세요.

> df <- data.frame (id=100:104,

weight=c(60,90,75,95,65),

size=c("small", "large", "medium", "large","small"))

> df

id weight size

1 100 60 small

2 101 90 large

3 102 75 medium

4 103 95 large

5 104 65 small

[문제69] df변수에 weight 컬럼을 기준으로 오름차순 정렬해서 df 변수에 값을 출력하세요.(order 함수를 이용하세요)

> df[order(df$weight), ]

id weight size

1 100 60 small

5 104 65 small

3 102 75 medium

2 101 90 large

4 103 95 large

[문제70] df변수에 size, weight 컬럼을 기준으로 오름차순 정렬하세요.(order 함수를 이용하세요)

select \* from df order by size, weight desc;

> df[ order(df$size, -df$weight), ]

id weight size

2 101 90 large

4 103 95 large

3 102 75 medium

1 100 60 small

5 104 65 small

>

[문제71] df 변수에 있는 weight 컬럼을 기준으로 내림차순 정렬하세요.(order 함수를 이용하세요)

> df[ order(df$weight, decreasing=TRUE), ]

id weight size

4 103 95 large

2 101 90 large

3 102 75 medium

5 104 65 small

1 100 60 small

> df[ order(-df$weight), ]

id weight size

4 103 95 large

2 101 90 large

3 102 75 medium

5 104 65 small

1 100 60 small

[문제72] 30번 부서 사원들의 last\_name, salary를 출력하세요.

단 salary를 기준으로 내림차순정렬하세요.

x <- emp[emp$DEPARTMENT\_ID ==30, c("LAST\_NAME", "SALARY")]

str(x)

na.omit(x[order(x$SALARY, decreasing=T),])

> library(doBy)

> orderBy(~-SALARY,emp[emp$DEPARTMENT\_ID ==30,c('LAST\_NAME','SALARY')])

LAST\_NAME SALARY

24 Raphaely 11000

25 Khoo 3100

26 Baida 2900

27 Tobias 2800

28 Himuro 2600

29 Colmenares 2500

NA <NA> NA

>

> na.omit(orderBy(~-SALARY,emp[emp$DEPARTMENT\_ID ==30,c('LAST\_NAME','SALARY')]))

LAST\_NAME SALARY

24 Raphaely 11000

25 Khoo 3100

26 Baida 2900

27 Tobias 2800

28 Himuro 2600

29 Colmenares 2500

[문제73] job\_id가 ST\_CLERK 가 아닌 사원들의 last\_name, salary, job\_id를 출력하는데 급여가 높은 사원부터 출력되게하세요.(orderBy 함수를 이용하세요)

orderBy(~-SALARY, emp[emp$JOB\_ID !='ST\_CLERK', c("LAST\_NAME","SALARY","JOB\_ID") ])

[문제74] 사원 last\_name, salary, commission\_pct를 출력하는데 commission\_pct를 기준으로 오름차순정렬하세요.(orderBy를 이용하세요)

orderBy(~COMMISSION\_PCT, emp[, c("LAST\_NAME","SALARY","COMMISSION\_PCT")])

[문제75] commission\_pct를 받고 있는 사원들의 last\_name, salary, commission\_pct를 출력하는데 commission\_pct를 기준으로 오름차순정렬하세요.(orderBy를 이용하세요)

orderBy(~COMMISSION\_PCT,emp[!is.na(emp$COMMISSION\_PCT),c("LAST\_NAME","SALARY","COMMISSION\_PCT")])

[문제76] 6의 9승을 출력하세요.

6^9

[문제77] 10을 3으로 나눈 나머지값을 출력하세요.

10%%3

[문제78] last\_name, salary에 12를 곱해서 출력하고 컬럼명이 한글로 연봉으로 데이터 프레임으로 출력하세요.

data.fram(emp$LAST\_NAME , 연봉=emp$SALARY\*12)

[문제79] last\_name과 연봉을 출력하는데 연봉이 높은것부터 출력하세요.

library(doBy)

orderBy(~-연봉,data.frame(emp$LAST\_NAME , 연봉=emp$SALARY\*12))

[문제80] 문제79를 round 함수를 이용해서 아래와 같이 백단위에서 반올림되게 하세요.

4: JONES 35700 -----> 36000

library(doBy)

orderBy(~-연봉,data.frame(emp$LAST\_NAME , 연봉=round(emp$SALARY\*12),-3)

**■ 그룹함수**

select sum(salalry), avg(salary) from x;

x <- c(100,90,80,70)

sum(x) #합

mean(x) # 평균

var(x) #분산

sd(x) # 표준편차

max(x) #최대값

min(x) #최소값

length(x) #행의 수

NROW(x) #갯수

**> x <- c(100,90,80,70,NA)**

**> sum(x)**

[1] NA

**> sum(x,na.rm=TRUE)**

[1] 340

**> sum(x,na.rm=FALSE)**

[1] NA

**> mean(x,na.rm=TRUE)**

[1] 85

**> var(x,na.rm=TRUE)**

[1] 166.6667

**> sd(x,na.rm=TRUE)**

[1] 12.90994

**> max(x,na.rm=TRUE)**

[1] 100

**> min(x,na.rm=TRUE)**

[1] 70

**> length(x)**

[1] 5

**> NROW(x)**

[1] 5

NA는 하나의 개수 행으로봄

**> length(na.omit(x)) #NA를 제거하고 갯수를 세겠다.**

[1] 4

NA포함인지 제외인지 결측치를 어떻게 해야할지 확인해야한다.

**■aggregate함수**

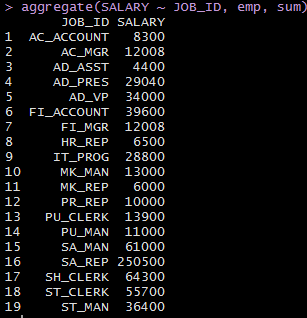
데이터를 분할하고 각 그룹으로 묶은 후 그룹함수를 적용한다.

(오라클의 group by를 함수로제공)

aggregate(계산될 컬럼~분할해야할 기준컬럼, 데이터 함수)

**aggregate(SALARY ~ JOB\_ID, emp, sum)**

> aggregate(SALARY ~ JOB\_ID, emp, sum)



=

**select job\_id, sum(salalry) from emp group by job\_id;**

[문제81] 최대월급을 출력하세요

max(emp$SALARY)

[문제82] 직업이 ST\_CLERK 인 사원들중에 최대월급을 출력하세요

x <- emp[emp$JOB\_ID == 'ST\_CLERK', c('SALARY')]

max(x)

max( emp[emp$JOB\_ID == 'ST\_CLERK', c('SALARY')] )

[문제83] 부서번호별로 급여에 총액을 출력하세요.

aggregate(SALARY~DEPARTMENT\_ID,emp,sum)

[문제84] 부서번호, 직업별로 급여에 총액을 출력하세요.

aggregate(SALARY~DEPARTMENT\_ID**+**JOB\_ID,emp,sum)

**분할해야할 컬럼이 여러개면 + 로 묶어주면 된다. department\_id + job\_id이런식으로**

[문제85] 부서번호별 최대월급을 출력하는데 최대월급이 높은것부터 출력하세요.

library(doBy)

orderBy(~-SALARY, aggregate(SALARY~DEPARTMENT\_ID,emp,max))

[문제86] 직업별 인원수를 출력하세요.

aggregate(EMPLOYEE\_ID ~ JOB\_ID, emp, length)

**apply**

- 행렬, 배열, 데이터프레임에 함수를 적용한 결과를 벡터, 리스트, 배열 형태로 리턴한다.

- 행렬에서 행이나 열의 방향으로 함수를 적용

- 적용하는 대상의 종류에 따라서 사용되는 함수가 달라지게 된다. 예를 들면, array가 적용대상인 경우는 apply함수, list인 경우에는 lapply함수, table인 경우에는 tapply함수가 된다.

applyt(x,,MARGIN, FUN)

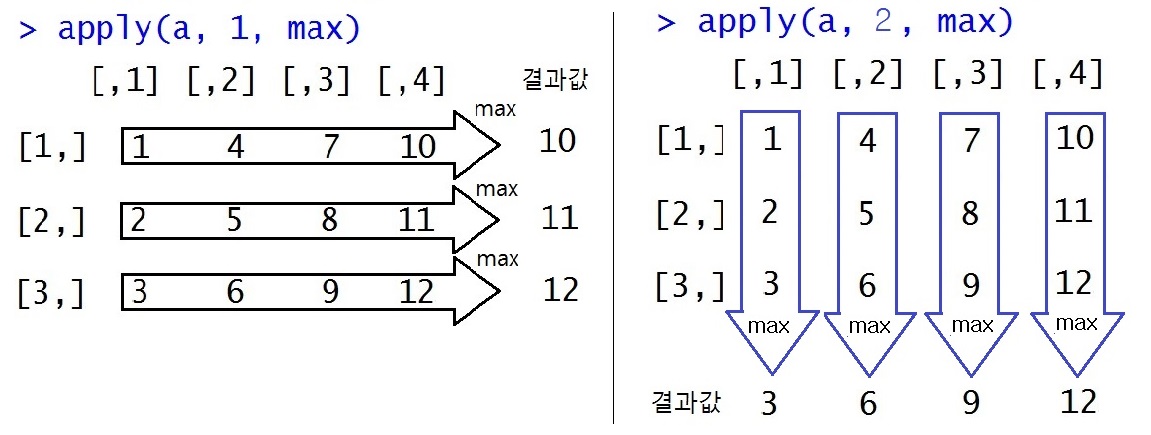
x : 행렬, 배열, 데이터프레임

MARGIN : 함수를 적용할 때 방향을 지정

1:행방향의미, 2:열방향의미, c(1,2): 행,열

FUN: 적용할 함수 (sum, mean, var, sd, max, min)

apply계열 많이 쓴다. 혼동되기 쉬우니 테스트를 많이해보자.



> m<- matrix(1:4, ncol=2)

> m

[,1] [,2]

[1,] 1 3

[2,] 2 4

> dim(m)

[1] 2 2

> apply(m,1,sum) #1+3=4, 2+4=6

[1] 4 6

> apply(m,2,sum) # 1+2=3, 3+4=7

[1] 3 7

1이 행방향 2가 열방향

> df<-data.frame(name=c('king','smith','jane'),sql=c(90,NA,70),python=c(75,90,NA))

> df

name sql python

1 king 90 75

2 smith NA 90

3 jane 70 NA

> apply(df[,2],1,sum)

Error in apply(df[, 2], 1, sum) :

dim(X)는 반드시 양의 값을 가지는 길이를 가져야 합니다

**NA값 처리해줘야함**

**어플라이 모양은 행렬모양, 배열모양, 데이터프레임모양일 경우에만돌아간다.**

**벡터형태 특정단일값을 뽑으려면 굳이 어플라이계열을 이용할 필요없이sum max이런함수쓰면되니까..**

#행기준으로

> apply(df[,c(2:3)],1,sum)

[1] 165 NA NA

> apply(df[,c(2:3)],1,sum,na.rm=T)

[1] 165 90 70

#열기준으로

> apply(df[,c(2:3)],**2**,sum)

sql python

NA NA

> apply(df[,c(2:3)],**2**,sum,na.rm=T)

sql python

160 165

**rowSums() : 배열, 행렬, 데이터프레임의 행의 합**

**rowMeans() : 배열, 행렬, 데이터프레임의 행의 평균**

이 두가지는 제공 나머지는 apply써야함

**> rowSums(df[,2:3],na.rm=T)**

[1] 165 90 70

**> rowMeans(df[,2:3],na.rm=T)**

[1] 82.5 90.0 70.0

**> rowMeans(df[,c(2,3)],na.rm=T)**

[1] 82.5 90.0 70.0

**위에 3번째것 Margin1과 같음**

**colSums() : 배열, 행렬, 데이터프레임의 열의 합**

**colMeans() :배열, 행렬, 데이터프레임의 열의 평균**

**> colSums(df[,2:3],na.rm=T)**

sql python

160 165

**> colMeans(df[,2:3],na.rm=T)**

sql python

80.0 82.5

**위에 Margin2와 같음**

**lapply**

-벡터, 리스트, 데이터프레임에 함수를 적용하고 그 결과를 리스트로 리턴하는 함수

-리스트: 서로 다른 데이터타입에 값을 저장하는 자료형이다.

> x<-list(a=1:3,b=4:6)

> x

$a

[1] 1 2 3

$b

[1] 4 5 6

> str(x)

List of 2

$ a: int [1:3] 1 2 3

$ b: int [1:3] 4 5 6

리스트지만 한줄만뽑아서 평균구하는거->벡터와마찬가지 ->그냥 mean하면되징

하지만

**mean(x$a)**

**mean(x$b)**

두번해야한다.

**> lapply(x,mean)**

$a

[1] 2

$b

[1] 5

한번에나온다.

**apply vs. laapply**

**> lapply(df[,2:3],mean,na.rm=T) #리스트에 적합**

$sql

[1] 80

**$python**

[1] 82.5

**> apply(df[,2:3],2,mean,na.rm=T) #array형에 사용적합**

sql python

80.0 82.5

**> colMeans(df[,2:3],na.rm=T)**

sql python

80.0 82.5

자료형에따라 적절히 선택하자ㅏ.

> unlist(lapply(df[,2:3],mean,na.rm=T))

sql python

80.0 82.5

모양이 열기준같지만 모양자체는 백터모양이다.

**리스트형을 데이터프레임 형식으로 바꾸고 싶다면..**

**> matrix(unlist(lapply(df[,2:3],mean,na.rm=T)),ncol=2, byrow=TRUE) #** 열은 2개, 로우행 방향 값을 채워넣어라.

[,1] [,2]

[1,] 80 82.5

**> x<- data.frame(matrix(unlist(lapply(df[,2:3],mean,na.rm=T)),ncol=2, byrow=TRUE)) #dataframe으로 만들어줌**

> x

X1 X2

1 80 82.5

**1. 리스트를 벡터형으로 바꾼다.**

**2. 그 벡터를 매트릭스모양으로 바꾼다.**

**3. 다시 그 매트릭스를 데이터프레임형으로 모양을 만든다.**

**많이쓴다. 중요**

> x<-as.data.frame(matrix(unlist(lapply(df[,2:3],mean,na.rm=T)),ncol=2, byrow=TRUE))

> x

V1 V2

1 80 82.5

> str(x)

'data.frame': 1 obs. of 2 variables:

$ V1: num 80

$ V2: num 82.5

> class(x)

[1] "data.frame"

> mode(x)

[1] "list"

> names(x)<-c('sql','python')

> x

sql python

1 80 82.5 서로다른 열들을 데이터프레임으로 만들고 그안의 모습은 리스트형으로 되어있다.

클래스하고 모드 틀린경우가있는 이유

전체자료형은 dataframe

속안은list

[정리]

**lapply(df[,2:3],mean,na.rm=T) #리스트형의 결과**

**1.벡터형으로 변환**

**unlist(lapply(df[,2:3],mean,na.rm=T))**

**2.matrix변환**

**matrix(unlist(lapply(df[,2:3],mean,na.rm=T)),ncol=2, byrow=TRUE)**

**3.data.frame 변환**

**x<-data.frame(matrix(unlist(lapply(df[,2:3],mean,na.rm=T)),ncol=2, byrow=TRUE))**

**자료형을 잘봐야한다.**

**sapply**

-벡터, 리스트, 데이터프레임에 함수를 적용하고 그 결과를 벡터로 리턴하는 함수

> x<- sapply(df[,2:3],mean,na.rm=T)

> str(x)

Named num [1:2] 80 82.5

- attr(\*, "names")= chr [1:2] "sql" "python"

> class(x)

[1] "numeric"

> mode(x)

[1] "numeric"

벡터형 뉴메릭

**> as.data.frame(sapply(df[,2:3],mean,na.rm=T))**

sapply(df[, 2:3], mean, na.rm = T)

sql 80.0

python 82.5

데이터프레임으로 하면 이렇게된다.

**이런모양으로나오지만 데이터프레임이아니다.(벡터다.)**

데이터 프레임 형태로 만들어보자.(형변환해야할 경우가 있음)

**> y<-as.data.frame(matrix(sapply(df[,2:3],mean,na.rm=T),ncol=2,byrow=TRUE))**

> names(y)<-c('sql','python')

> y

sql python

1 80 82.5

**tapply**

-벡터, 데이터프레임에 저장된 데이터를 주어진 기준에 따라 그룹으로 묶은 뒤 그룹함수를 적용하고 그 결과를 **array 형식으로** 리턴하는 함수

-tapply는 그룹함수와 비슷하지만 array모양으로 표현해줌

**> tapply(emp$SALARY, emp$DEPARTMENT\_ID, sum) #비교 emp$있음**

10 20 30 40 50 60 70 80 90 100 110

4400 19000 24900 6500 156400 28800 10000 304500 63040 51608 20308

**> aggregate(SALARY~DEPARTMENT\_ID, emp, sum) #비교 emp$없음,**

DEPARTMENT\_ID SALARY

1 10 4400

2 20 19000

3 30 24900

4 40 6500

5 50 156400

6 60 28800

7 70 10000

8 80 304500

9 90 63040

10 100 51608

11 110 20308

**> aggregate(SALARY~DEPARTMENT\_ID+JOB\_ID,emp,sum)**

DEPARTMENT\_ID JOB\_ID SALARY

1 110 AC\_ACCOUNT 8300

2 110 AC\_MGR 12008

3 10 AD\_ASST 4400

4 90 AD\_PRES 29040

5 90 AD\_VP 34000

6 100 FI\_ACCOUNT 39600

7 100 FI\_MGR 12008

8 40 HR\_REP 6500

9 60 IT\_PROG 28800

10 20 MK\_MAN 13000

11 20 MK\_REP 6000

12 70 PR\_REP 10000

13 30 PU\_CLERK 13900

14 30 PU\_MAN 11000

15 80 SA\_MAN 61000

16 80 SA\_REP 243500

17 50 SH\_CLERK 64300

18 50 ST\_CLERK 55700

19 50 ST\_MAN 36400

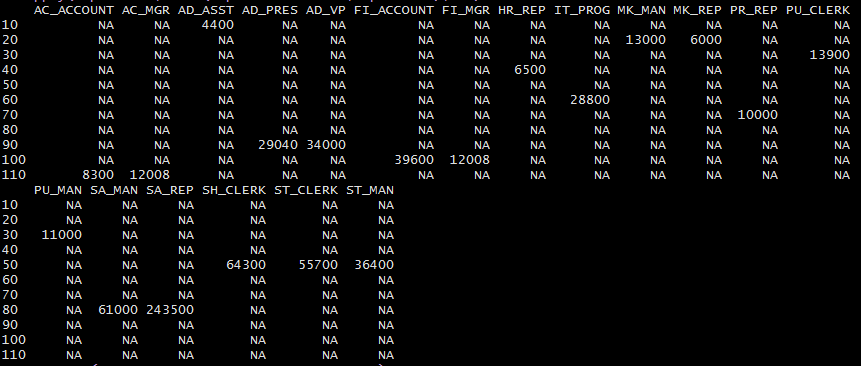
> tapply(emp$SALARY, **c**(emp$DEPARTMENT\_ID,emp$JOB\_ID),sum)

Error in tapply(emp$SALARY, c(emp$DEPARTMENT\_ID, emp$JOB\_ID), sum) :

인자들은 반드시 같은 길이를 가져야 합니다

**안된다.**

**> tapply(emp$SALARY, list(emp$DEPARTMENT\_ID,emp$JOB\_ID),sum)**

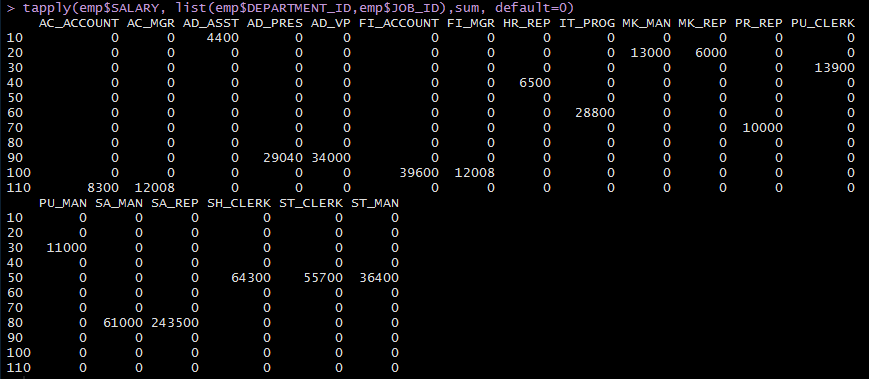


새로로만들려면 tapply, list형으로 묶어주면된다. c로묶으면 안됨

tapply(emp$SALARY, **data.frame**(emp$DEPARTMENT\_ID,emp$JOB\_ID),sum)

이것도 같은 값나오는데 그래도 내부에서는 list로 똑같음

> tapply(emp$SALARY, list(emp$DEPARTMENT\_ID,emp$JOB\_ID),sum, **default=0**)



tapply많이 쓴다. 중요하다.

자료분석하기에 더적합한형태여서 aggregate도 이렇게바꿔달라고 한다.

c.f) **R의 변수 제거 방법**

> **ls()** #변수체크

[1] "a" "b" "c" "cell" "cells" "cname" "d" "df"

[9] "dimnames" "emp" "F" "f100" "gender" "i" "m" "member"

[17] "rname" "s1" "s2" "salary" "T" "total" "u" "w"

[25] "x" "y" "z"

> **rm(y)** #y라는 변수 지우기

> **rm(list=ls())** #전부다지우기

> **ls()**

character(0)

**제어문**

조건의 흐름을 제어

**1.if 문**

: 조건에 따라 서로 다른 코드를 수행하도록 하는 문장

if(조건){

조건에 참일때 수행하는 문장

} else

{

조건이 거짓일때 수행하는 문장

}

예)

> if(1>2){

+ print("1이 2보다 크다")

+ }

> if(2>1){

+ print("2가 1보다 크다")

+ }

[1] "2가 1보다 크다"

+ print("1이 2보다 크다")

+ }else

+ {

+ print("1은 2보다 작다")

+ }

[1] "1은 2보다 작다"

**2. Ifelse함수**

ifelse(조건, 참, 거짓)

**3.swtich**

switch(변수, 실행문1, 실행문2,...) 변수값에 따라 조건에 맞는 실행문을 수행한다.

> x<-1

> switch(x,'남은 기간 최선을다하자','행복하자','건강하게살자')

[1] "남은 기간 최선을다하자"

> x<-2

> switch(x,'남은 기간 최선을다하자','행복하자','건강하게살자')

[1] "행복하자"

> x<-3

> switch(x,'남은 기간 최선을다하자','행복하자','건강하게살자')

[1] "건강하게살자"

변수는 1,2,3값으로..

> x<-'산'

> switch(x,'산'='한라산 가고싶다.','바다'='함덕해수욕장 가고싶다.',

+ paste0(x,'그냥방콕하세요'))

[1] "한라산 가고싶다."

변수바꿔줄 수 있다.

[문제87] 아래화면의 결과 처럼 데이터 프레임을 생성한 후

total 컬럼을 생성해서 sql과 python 의 합을 구하세요.(단 apply함수를 이용하세요)

name sql python

king 96 75

smith NA 91

jane 78 86

scott 90 NA

df<-data.frame(name=c('king','smith','jane','scott'),sql=c(96,NA,78,90),python=c(75,91,86,NA),stringsAsFactors=F)

df<-cbind(df,total=apply(df[,c(2,3)],1,sum,na.rm=TRUE))

df

name sql python total

1 king 96 75 171

2 smith NA 91 91

3 jane 78 86 164

4 scott 90 NA 90

[문제88] 입사한 년도별 평균월급을 세로(aggregate), 가로(tapply)로 출력하세요.

emp<-read.csv("c:/data/emp.csv",header=T,stringsAsFactors=FALSE)

str(emp) # emp$HIRE\_DATE가 int형임을 알 수 있음

emp$HIRE\_DATE<-as.Date(as.character(emp$HIRE\_DATE),format='%Y%m%d')

str(emp) # emp$HIRE\_DATE를 문자형으로 바꿔주고 데이트형식으로 바꿔줬음

aggregate(SALARY~format(emp$HIRE\_DATE,'%Y'),emp,mean)

tapply(emp$SALARY,format(emp$HIRE\_DATE,'%Y'), mean)

[문제89] 부서별 인원수를 세로(aggregate), 가로(tapply)로 출력하세요.

aggregate(EMPLOYEE\_ID ~ DEPARTMENT\_ID, emp, length)

DEPARTMENT\_ID EMPLOYEE\_ID

1 10 1

2 20 2

3 30 6

4 40 1

5 50 45

6 60 5

7 70 1

8 80 34

9 90 3

10 100 6

11 110 2

tapply(emp$EMPLOYEE\_ID, emp$DEPARTMENT\_ID, length)

10 20 30 40 50 60 70 80 90 100 110

1 2 6 1 45 5 1 34 3 6 2

[문제90]job\_id, hire\_date(년도4자리)별 총액 급여를 aggregate함수를 이용해서 생성하세요.

aggregate(SALARY**~JOB\_ID+format(emp$HIRE\_DATE,'%Y')**,emp,sum)

[문제91] job\_id, hire\_date(년도4자리) 총액 급여를 tapply함수를 이용해서 생성하세요.

단 NA 대신에 0 으로 출력하세요.

x<-tapply(emp$SALARY,**list(emp$JOB\_ID,format(emp$HIRE\_DATE,'%Y')),**sum)

x[is.na(x)==T]<-0

x

x<-tapply(emp$SALARY,list(emp$JOB\_ID,format(emp$HIRE\_DATE,'%Y')),sum,default=0)

x

x<-tapply(emp$SALARY,data.frame(emp$JOB\_ID,format(emp$HIRE\_DATE,'%Y')),sum,default=0)

x

[문제92] 변수에 2를 입력한후 그 변수에 값이 2의 배수면 "2의 배수" 출력

아니면 "2의 배수가 아니다" 출력해주세요. (if문, ifelse문)

if (x%%2==0){

print("2의배수")

} else

{

print("2의 배수가 아니다")

}

[1] "2의배수"

ifelse(x%%2==0,"2의 배수", "2의 배수가 아니다")

[1] "2의 배수"

[문제93] emp 변수에 있는 데이터를 가지고 새로운 df변수를 생성하세요.

last\_name, salary, 급여가 10000 이상이면 A, 5000이상 10000보다 작으면 B 나머지는 C가 입력되어 있는 새로운 컬럼을 생성하세요.

컬럼이름은 name, sal, level 로 설정하세요.

emp <-read.csv("c:/data/emp.csv",header=T, stringsAsFactors = FALSE)

df<-data.frame(emp$LAST\_NAME, emp$SALARY,

ifelse(emp$SALARY >= 10000, 'A',

ifelse(emp$SALARY >= 5000 & emp$SALARY < 10000, 'B','C')))

names(df)<-c("name","sal","level")

[문제94] x 변수에 1부터 100까지 입력한 후 짝수값은 자신의 값에 10을 곱한 값으로 수정하세요.

x<-c(1:100)

x<-ifelse(x%%2==0,x\*10,x)

x

[1] 1 20 3 40 5 60 7 80 9 100 11 120 13 140 15 160 17 180 19

[20] 200 21 220 23 240 25 260 27 280 29 300 31 320 33 340 35 360 37 380

[39] 39 400 41 420 43 440 45 460 47 480 49 500 51 520 53 540 55 560 57

[58] 580 59 600 61 620 63 640 65 660 67 680 69 700 71 720 73 740 75 760

[77] 77 780 79 800 81 820 83 840 85 860 87 880 89 900 91 920 93 940 95

[96] 960 97 980 99 1000

x<-1:100

x[x%%2==0] <- x[x%%2==0] \*10

x

[1] 1 20 3 40 5 60 7 80 9 100 11 120 13 140 15 160 17 180 19

[20] 200 21 220 23 240 25 260 27 280 29 300 31 320 33 340 35 360 37 380

[39] 39 400 41 420 43 440 45 460 47 480 49 500 51 520 53 540 55 560 57

[58] 580 59 600 61 620 63 640 65 660 67 680 69 700 71 720 73 740 75 760

[77] 77 780 79 800 81 820 83 840 85 860 87 880 89 900 91 920 93 940 95

[96] 960 97 980 99 1000

[문제95] x <- c(2,10,6,4,3,NA,7,9,1) x변수에 NA가 있는지를 검사하세요.

x <- c(2,10,6,4,3,NA,7,9,1)

is.na(x)

[문제96] x 변수에 NA가 있는 인덱스 번호를 찾아 주세요.

which(is.na(x))

[문제97] x 변수에 NA가 있으면 0으로 설정하세요

x[which(is.na(x))]=0

[문제98] last\_name, salary, commission\_pct,

commission\_pct NA 면 salary \* 12,

아니면 (salary \* 12) + (salary \* 12 \* commission\_pct)을 수행하세요.

> head(df)

name sal comm ann\_sal

1 OConnell 2600 NA 31200

2 Grant 2600 NA 31200

3 Whalen 4400 NA 52800

4 Hartstein 13000 NA 156000

5 Fay 6000 NA 72000

6 Mavris 6500 NA 78000

>

답1

df<-data.frame('name'=emp$LAST\_NAME,'sal'=emp$SAL,'comm'=emp$COMMISSION\_PCT,'ann\_sal'=ifelse(is.na(emp$COMMISSION\_PCT)==T,emp$SALARY\*12,((emp$SALARY \* 12) + (emp$SALARY \* 12 \* emp$COMMISSION\_PCT))))

답2

df<-data.frame(emp$LAST\_NAME,emp$SAL,emp$COMMISSION\_PCT,ifelse(is.na(emp$COMMISSION\_PCT)==T,emp$SALARY\*12,((emp$SALARY \* 12) + (emp$SALARY \* 12 \* emp$COMMISSION\_PCT))))

names(df)<-c('name','sal','comm','ann\_sal')

c.f) R에서의 NULL: 아직초기값을 할당하지 않은 값

NA: 아직정해지지않은 알수없는 값

> x<-1

> if (x==1){'남은 기간 최선을 다하자'

+ } else{if(x==2){'내년에도 다른 학원또 다니지뭐'

+ }else{if(x==3){'그냥 사는거지뭐'}}}

[1] "남은 기간 최선을 다하자"

> x<-2

> if (x==1){'남은 기간 최선을 다하자'

+ } else{if(x==2){'내년에도 다른 학원또 다니지뭐'

+ }else{if(x==3){'그냥 사는거지뭐'}}}

[1] "내년에도 다른 학원또 다니지뭐"

> x<-3

> if (x==1){'남은 기간 최선을 다하자'

+ } else{if(x==2){'내년에도 다른 학원또 다니지뭐'

+ }else{if(x==3){'그냥 사는거지뭐'}}}

[1] "그냥 사는거지뭐"

좀더 편리한 방법을 제공

**■반복문**

**1. for문**

for(변수 in 데이터변수){

반복수행할 문장

}

변수:카운트변수생각하면됨

데이터변수: 값, 얼마나수행해야하는지

> for(i in 1:10){

+ print(i)

+ }

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

[1] 6

[1] 7

[1] 8

[1] 9

[1] 10

짝수의 합

hap<-0

x<-c(2,4,6,8,10)

for(i in x){hap<-hap+i}

hap

[문제99] x변수에 1부터 100까지 입력한 후

1은 합을, 2는 평균, 3은 분산, 4는 표준편차를

구하는 switch문을 생성하세요.

x<-1:100

y<-1

switch(y,sum(x),avg(x),var(x),sd(x))

[문제100] 1부터 10까지 합을 for 문을 이용해서 구하세요.

hap<-0

x<-1:10

for(i in x){hap<-hap+i}

hap

[1] 55

[문제 101] 1부터 100까지 전체합, 짝수합, 홀수합을 출력하세요.

odd<-0

even<-0

total<-0

for(i in 1:100)

{

total<-total+i

if (i %% 2==0)

{

even<-even+i

}else{

odd<-odd+i

}

}

print(odd)

[1] 2500

print(even)

[1] 2550

print(total)

[1] 5050

[문제 102] 1부터 100까지의 짝수합, 홀수합을 tapply를 이용해서 구하세요.

hap<-tapply(1:100,1:100%%2==0,sum)

hap

FALSE TRUE

2500 2550

rownames(hap)=c('홀수','짝수')

hap

홀수 짝수

2500 2550

[문제103] 1부터100까지 홀수만 x 변수에 입력해주세요.(for문을 이용하세요)

s<-seq(1,100,2)

s

x<-NULL; z<-NULL; y<-NULL; r<-NULL; #세미콜론으로 구분지어주면 변수선언을한줄에가능!

for(i in 1:100)

{

if(i%%2!=0)

{

print(paste(i,'홀수'))

x<-append(x,i)

z<-c(z,i)

y<-cbind(y,i)

r<-rbind(r,i)}

}

print(x)

print(z)

print(y)

print(r)

x,y,z,r 4가지방법으로 가능

**2. while**

while(조건){

반복수행할 문장

}

> i<-1

> while(i<= 10){

+ print(i)

+ i<-i+1

+ }

[1] 1

[1] 2

[1] 3

[1] 4

[1] 5

[1] 6

[1] 7

[1] 8

[1] 9

[1] 10

[문제104] while문을 이용해서 2단을 출력하세요.

[1] "2 x 1 = 2"

[1] "2 x 2 = 4"

[1] "2 x 3 = 6"

[1] "2 x 4 = 8"

[1] "2 x 5 = 10"

[1] "2 x 6 = 12"

[1] "2 x 7 = 14"

[1] "2 x 8 = 16"

[1] "2 x 9 = 18"

> i<-1

> while(i<=9){

+ print(paste("2x" ,i, "=" ,2\*i))

+ i<-i+1}

[1] "2x 1 = 2"

[1] "2x 2 = 4"

[1] "2x 3 = 6"

[1] "2x 4 = 8"

[1] "2x 5 = 10"

[1] "2x 6 = 12"

[1] "2x 7 = 14"

[1] "2x 8 = 16"

[1] "2x 9 = 18"

plsql처럼 r에도 기본루프가존재한다.

repeat

**3.repeat**

repeat{

반복수행할 문장

break

}

무한대로돌아감

break를 만나는 순간 repeat을 끝내(plsql에서도 exit썼듯이..)

for문, while문 에서도 break를 쓸 수 있다.(plsql에서도 exit썼듯이..)

repeat{

print(i)

if(i==10){break}

i<-i+1

}

[문제105] repeat문을 이용해서 2단을 출력하세요.

i<-1

while(i<=9){

print(paste("2x" ,i, "=" ,2\*i))

i<-i+1}

i<-1

repeat{print(paste("2x" ,i, "=" ,2\*i))

if(i==10){break}

i<-i+1

}

[문제106] x변수에 1부터 10까지 입력한 후 홀수인지 짝수인지를 출력하세요.

1 2 3 4 5 6 7 8 9 10

"홀수" "짝수" "홀수" "짝수" "홀수" "짝수" "홀수" "짝수" "홀수" "짝수"

x<-1:10

for (i in x) {

if (i%%2==0)

{print("짝수")}

else (i%%2!=0)

{print("홀수")}

}

샘답:

x<-1:10

x

ifelse(i%%2==0, '짝수','홀수')

[문제107] mysentence 이름의 벡터 변수에 'Well begun is half done' 이 값을 입력한 후 공백문자를 기준으로 분리해서 word 변수에 입력하세요. word 변수에 타입을 체크한 후 word 변수에 입력된 값의 수를 출력하세요.

mysentence = 'Well begun is half done'

mysentence

[1] "Well begun is half done"

word <- strsplit(mysentence,split=" ")

word

[[1]]

[1] "Well" "begun" "is" "half" "done"

word[[1]][1]

[1] "Well"

word[[1]][2]

[1] "begun"

class(word)

[1] "list"

length(word)

[1] 1

sapply(word,length)

[1] 5

>

[문제108] 문제107에서 만든 word변수에 있는 값을 리스트 변수 letters에 하나씩 저장하세요.

> letters <- list(rep(NA,5))

> letters

[[1]]

[1] NA NA NA NA NA

> for(i in 1:sapply(word,length)){

+ letters[[i]] <- word[[1]][i]

+ }

> letters

[[1]]

[1] "Well"

[[2]]

[1] "begun"

[[3]]

[1] "is"

[[4]]

[1] "half"

[[5]]

[1] "done"

> length(letters)

[1] 5

[문제109] 문제108에서 생성한 letters 변수에 있는 값을 myword변수에 하나의 문장으로 넣어 주세요.

> myword

[1] "Well begun is half done"

> myword <- NULL

> for(i in 1:length(letters)){

+ myword <- paste(c(myword,letters[[i]][1]),collapse=' ')

+ }

> myword

[1] "Well begun is half done"

**■함수(function)**

-사용자가 정의하는 함수를 생성할 수 있다.

-자주 반복되어 사용하는 기능을 정의하는 프로그램

-코드가 간단해진다.

R에서는 프로시져 없고 함수만만든다.

어디에 따로 저장할 수 없기 때문에 매번 수행하려면 소스코드를 가지고 있어야한다. DB와 일반 프로그램의 차이점.

함수이름<-function(){

함수 수행해야할 코드

return(반환값) #선택 R에서의 리턴값은 옵셔널하다.

}

Sys.Date()

date1<-function(){

return(Sys.Date())

}

date1()

date1이라는 함수를 만들었음

time<-function(){

Sys.time()

}

time()

time이라는 함수를 만들었음 return값없이

hap<-function**(x,y)**{

res<-**x+y**

return(res)

}

hap(1,2)

hap이라는 변수.

x,y 형식매개변수 함수호출할때 이곳에 값을 넣어주는거.

res로컬변수(지역변수) 이함수에서만 사용하는 변수이다.

[문제110] hap함수에 인자값을 입력하게되면 1부터해서 입력숫자까지 누적합을 구하세요.

hap(10)

> hap<-function(x){

+ res<-sum(1:x)

+ return(res)

+ }

> hap(10)

[1] 55

hap<-function(x)

선생님 닶

> hap<-function(x){

+ res<-0

+ for(i in 1:x){

+ res<-res+i

+ }

+ return(res)

+ }

> hap(10)

[1] 55

hap(10)

hap(1,2)

hap(1,2,3,)

hap(10,20,30,40)

인수값의 갯수가 가변이다.

이런걸 표현해야할 때 어떻게 만들어야하는가

->가변인수

**가변인수**

fff <- function(...)

{

x<-list(...)

for (i in x)

{print(i)}

}

점세개: 가변인수를 의미

갯수대로 list 만들어줘서 변수값넣어주는거임. c( ) 컴바인으로묶어줘도된다.

f(1,2)

f(1,2,3)

f('a','b','c','d')

> fff <- function(...)

+ {

+ x<-list(...)

+ for (i in x)

+ {print(i)}

+ }

> fff(1,2)

[1] 1

[1] 2

> fff(1,2,3)

[1] 1

[1] 2

[1] 3

어차피 벡터모양으로 출력됨

[문제 111] 인수 값에 따라 합을 구하세요.

hap(1,2)

hap(1,2,3)

hap(10,20,30,40)

hap(1,2)

hap(1,2,3)

hap(10,20,30,40)

hap(1:100)

hap<-function(...){

x<-c(...)

res<-0

for (i in x){

res<-res+i

}

return(res)

}

> hap<-function(...){

+ x<-c(...)

+ res<-0

+ for (i in x){

+ res<-res+i

+ }

+ return(res)

+ }

> hap(1,2)

[1] 3

> hap(1,2,3)

[1] 6

> hap(10,20,30,40)

[1] 100

> hap(1:100)

[1] 5050

[문제112] x변수에 1:5 까지 입력되어 있다 아래 화면 처럼 출력하세요.

<화면결과>

[[1]]

[1] "홀수"

[[2]]

[1] "짝수"

[[3]]

[1] "홀수"

[[4]]

[1] "짝수"

[[5]]

[1] "홀수"

as.list(ifelse(x%%2==0,"짝수","홀수"))

lapply(x, **function(x){ifelse(x%%2==0,"짝수","홀수")**})

**lapply에도 우리가 정의한 함수를 사용 할 수 있다.**

apply계열에 그대로쓸 수 있다.

lapply안 그자리에서 바로정의해버렸음

[문제113] 사원 번호를 입력 값으로 받아서 사원의 LAST\_NAME, SALARY를 출력하는 함수를 생성하세요.

> find(100)

LAST\_NAME SALARY

10 King 24000

find<-function(x){emp[emp$EMPLOYEE\_ID==x,c('emp$LAST\_NAME','emp$SALARY')]}

**중첩함수**

f<- function(x,y){

print(x)

**f2<-function(y){**

**y<-x\*y**

**print(y)**

**}**

**f2(y)**

}

f(10,20)

함수는 호출하지않으면 수행이안된다. 안에서 수행하려면 호출을 다시해줘야한다.

함수는 이런식으로 여러번 중첩가능하다.ㅣ

**전역변수(global variable)**

함수에 상관없이 프로그램 전체에서 사용할 수 있는 변수

**지역변수(local variable)**

함수내에서 정의되고 사용하는 변수

**매개변수(parameter variable)**

함수의 인수에서 받아서 사용하는 변수

> x<-1; y<-2; z<-3;

> f<-function(x){ #x 매개변수

+ y<-x\*10

+ print(x); print(y); print(z) #x 매개변수, y지역변수 z는 글로벌변수가 출력

+ }

> f(x)

[1] 1

[1] 10

[1] 3

1,10,3이 출력

다시 xyz출력해보면

> x

[1] 1

> y

[1] 2

> z

[1] 3

y값이 바뀜

> x<-1; y<-2; z<-3;

> f<-function(x){ #x 매개변수

+ y**<<**-x\*10

+ print(x); print(y); print(z) #x 매개변수, y지역변수 z는 글로벌변수가 출력

+ }

> f(x)

[1] 1

[1] 10

[1] 3

1,10,3이 출력

다시 xyz출력해보면

> x<-1; y<-2; z<-3;

> f<-function(x){

+ y<-x\*10

+ print(x); print(y); print(z)

+ }

> f(x)

[1] 1

[1] 10

[1] 3

화살표다르게 y<<- 하면 전역변수

> x<-1; y<-2; z<-3;

> f<-function(x){

+ y**<<-**x\*10 #전역변수가됨 위에건지역변수

+ print(x); print(y); print(z)

+ }

> f(x)

[1] 1

[1] 10

[1] 3

> x

[1] 1

> y

[1] 10

> z

[1] 3

>

값이 달라졌다.

-----------------정리

> x<-1; y<-2; z<-3;

> f<-function(x){

+ y<-x\*10 #지역

+ print(x); print(y); print(z)

+ }

> f(x)

[1] 1

[1] 10

[1] 3

> x

[1] 1

> y

[1] 2

> z

[1] 3

> x<-1; y<-2; z<-3;

> f<-function(x){

+ y<<-x\*10 #전역변수 프로그램밖에서 정의해놓은 글로벌변수에영향을줌

+ print(x); print(y); print(z)

+ }

> f(x)

[1] 1

[1] 10

[1] 3

> x

[1] 1

> y

[1] 10

> z

[1] 3

> x<-1; y<-2; z<-3;

> f<-function(x){

+ y=x\*10 #지역

+ print(x); print(y); print(z)

+ }

> f(x)

[1] 1

[1] 10

[1] 3

> x

[1] 1

> y

[1] 2

> z

[1] 3

sum(x1<-c(1,2,3,4,5))

x1

sum(y1<<-c(1,2,3,4,5))

y1

sum(z1=c(1,2,3,4,5))

z1

merge(join)

-두 데이터프레임의 **공통된 값을 기준으로 병합**한다.

x1<-data.frame(id=c(100,200,300),sql=c(70,90,80))

y1<-data.frame(id=c(100,200,300),python=c(80,70,60))

**> x1<-data.frame(id=c(100,200,300),sql=c(70,90,80))**

**> y1<-data.frame(id=c(100,200,300),python=c(80,70,60))**

**> cbind(x1,y1)**

id sql id python

1 100 70 100 80

2 200 90 200 70

3 300 80 300 60

**> rbind(x1,y1)**

Error in match.names(clabs, names(xi)) :

names do not match previous names

**> merge(x1,y1)**

id sql python

1 100 70 80

2 200 90 70

equi조인하고 같음 키값이 일치되는 것만 뽑아냄

**merge(x1,y1,all=T)**

**> merge(x1,y1,all=T) # full outer join**

id sql python

1 100 70 80

2 200 90 70

3 300 80 NA

4 500 NA 60

**> merge(x1,y1,all.x=T) # x1 left outer join**

id sql python

1 100 70 80

2 200 90 70

3 300 80 NA

**> merge(x1,y1,all.y=T) # y1 right outer join**

id sql python

1 100 70 80

2 200 90 70

3 500 NA 60

> x1<-data.frame(**id**=c(100,200,300),sql=c(70,90,80))

> x2<-data.frame(id=c(100,200,300),sql=c(80,70,60))

> x3<-data.frame(**no**=c(100,200,500),python=c(80,60,70))

**> merge(x1,x3,by.x='id', by.y='no') #특정컬럼을 유지하자. join on 절 처럼 만든 것**

id sql python

1 100 70 80

2 200 90 60

**> x4<-merge(x1,x3,by.x='id', by.y='no')**

> x4

id sql python

1 100 70 80

2 200 90 60

**> merge(x4, x2, by.x='id', by.y='id')**

id sql.x python sql.y

1 100 70 80 80

2 200 90 60 70

> x4<-merge(x1,x3,by.x='id', by.y='no', **all=T**)

> x4

id sql python

1 100 70 80

2 200 90 60

3 300 80 NA

4 500 NA 70

> merge(x4, x2, by.x='id', by.y='id', **all=T**)

id sql.x python sql.y

1 100 70 80 80

2 200 90 60 70

3 300 80 NA 60

4 500 NA 70 NA

dept csv

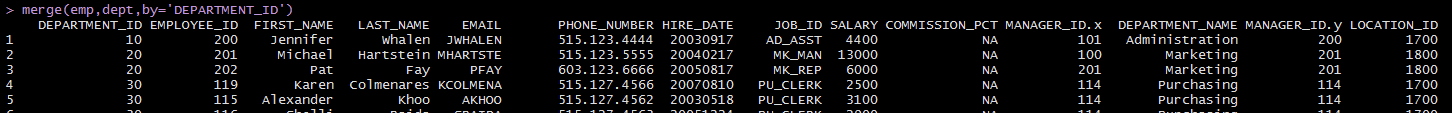
merge(emp,dept,by='DEPARTMENT\_ID')

=

select last\_name, department\_id

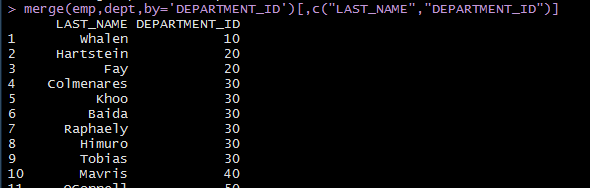
from emp join dept

using(department\_id);



특정칼럼을보고싶음 뒤에대괄호

merge(emp,dept,by='DEPARTMENT\_ID')[,c("LAST\_NAME","DEPARTMENT\_ID")]



[문제 114] 20번 부서에 소속되어 있는 사원의 LAST\_NAME, SALARY, JOB\_ID, DEPARTMENT\_NAME을 출력하세요.

select last\_name, salary, job\_id, department\_name

from emp join dept

using(department\_id)

where department\_id =20;

> emp\_20 <- emp[emp$DEPARTMENT\_ID == 20,]

> merge(emp\_20, dept, by="DEPARTMENT\_ID")[ , c("LAST\_NAME","SALARY","DEPARTMENT\_NAME")]

select e.last\_name, e.salary, e.job\_id, d.department\_name

from emp e, dept d

where e.department\_id=20

and d.department\_id=20;

merge(emp[emp$DEPARTMENT\_ID==20, c('LAST\_NAME','SALARY','DEPARTMENT\_ID')], dept[dept$DEPARTMENT\_ID==20,])[,c('LAST\_NAME','SALARY','JOB\_ID',DEPARTMENT\_NAME')]

merge(emp[emp$DEPARTMENT\_ID == 20,][,c('LAST\_NAME','SALARY','DEPARTMENT\_ID')], dept, by="DEPARTMENT\_ID")[ , c("LAST\_NAME","SALARY","JOB\_ID","DEPARTMENT\_NAME")]

[문제115] salary가 3000 이상이고 job\_id는 ST\_CLERK인 사원들의 employee\_id, salary, job\_id, department\_id,department\_name을 출력하세요.

공통이면 by.x by.y 이런식으로 안쓰고 그냥 by=으로 써도된다.

x <- emp[emp$SALARY>=3000 & emp$JOB\_ID=='ST\_CLERK',c('EMPLOYEE\_ID','SALARY','JOB\_ID','DEPARTMENT\_ID')]

merge(x, dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME')])

merge(emp[emp$SALARY>=3000 & emp$JOB\_ID=='ST\_CLERK',c('EMPLOYEE\_ID','SALARY','JOB\_ID','DEPARTMENT\_ID')], dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME')])

[문제116] 커미션이 NA 인 사원들의 last\_name, commission\_pct, department\_id, department\_name을 출력하세요.

x <- emp[is.na(emp$COMMISSION\_PCT),c("LAST\_NAME","COMMISSION\_PCT","DEPARTMENT\_ID") ]

merge(x,dept,by='DEPARTMENT\_ID')[,c('LAST\_NAME','COMMISSION\_PCT','DEPARTMENT\_ID','DEPARTMENT\_NAME')]

[문제117] 커미션이 NA가 아닌 사원들의 last\_name, commission\_pct,department\_id, department\_name을 출력하세요.

x <- emp[!is.na(emp$COMMISSION\_PCT),c("LAST\_NAME","COMMISSION\_PCT","DEPARTMENT\_ID") ]

y <- dept[,c("DEPARTMENT\_ID","DEPARTMENT\_NAME")]

merge(x,y,by='DEPARTMENT\_ID')[,c('LAST\_NAME','COMMISSION\_PCT','DEPARTMENT\_ID','DEPARTMENT\_NAME')]

[문제118]커미션이 NA가 아닌 사원들의 last\_name, commission\_pct, department\_id, department\_name을 출력하세요.

단 department\_id가 NA인 사원도 출력해주세요.

x1 <- emp[!is.na(emp$COMMISSION\_PCT),c("LAST\_NAME","COMMISSION\_PCT","DEPARTMENT\_ID") ]

y1 <- dept[,c("DEPARTMENT\_ID","DEPARTMENT\_NAME")]

merge(x1,y1,all.x=T)[,c('LAST\_NAME','COMMISSION\_PCT','DEPARTMENT\_ID','DEPARTMENT\_NAME')]

레프트아우터조인

[문제119] 사원의 last\_name, 관리자 last\_name을 출력해주세요. 관리자가 없는 사원도 출력해주세요.

셀프조인

w <- emp[,c('EMPLOYEE\_ID','LAST\_NAME','MANAGER\_ID')]

m <- emp[,c('EMPLOYEE\_ID','LAST\_NAME')]

merge(w,m,by.x='MANAGER\_ID',by.y='EMPLOYEE\_ID',all.x=T)[,c('LAST\_NAME.x','LAST\_NAME.y')]

x축 매니저아이디와 y쪽 임플로이아이디

사원아이디는 있는데 매니저아이디 없는 사람도있으니 left outer조인

[문제120] 부서이름별 총액 급여를 출력하세요.

x <- aggregate(SALARY ~ DEPARTMENT\_ID, emp, sum) #부서아이디별로 그루핑 샐러리에대한 sum

x

names(x) <- c("DEPARTMENT\_ID","SUM\_SAL") # 직접 이름 바꿔줌

x

merge(x, dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","SUM\_SAL")]

select d.department\_name, x.sum\_sal

from (select department\_id, sum(salary) sum\_sal

from emp

group by department\_id)x, dept d

where x.department\_id = d.department\_id;

aggregate(SALARY ~ DEPARTMENT\_NAME, merge(emp[,c('SALARY','DEPARTMENT\_ID')], dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME')]), sum)

select d.department\_name, sum(e.salary)

from emp e, dept d

where e.department\_id = d.department\_id

group by d.department\_name;

[문제121] 부서이름별 소속사원들의 인원수를 출력하세요.

제한할 것이 있으면 머지하기전에 제한하는 것이좋다.

x<-aggregate(emp$EMPLOYEE\_ID~emp$DEPARTMENT\_ID,emp,length)

names(x)<-c('DEPARTMENT\_ID','count noses')

x

merge(x, dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","count noses")]

선생님 답

x <- aggregate(SALARY ~ DEPARTMENT\_ID+JOB\_ID, emp, sum)

x

merge(x, dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","JOB\_ID","SALARY")]

aggregate(SALARY ~ DEPARTMENT\_NAME + JOB\_ID, merge(emp[,c('SALARY','DEPARTMENT\_ID','JOB\_ID')], dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME')]), sum)

merge(x, dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","count noses")]

[문제122] 최고 급여를 받는 사원의 이름, 급여, 부서코드, 부서이름를 출력하세요.

merge(emp,dept,by='DEPARTMENT\_ID')[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID","DEPARTMENT\_NAME")]

선생님답

merge(emp[emp$SALARY==max(emp$SALARY), c("LAST\_NAME","SALARY","DEPARTMENT\_ID")],dept)[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID","DEPARTMENT\_NAME")]

[문제123] 부서이름,직업별 급여의 총액을 구하세요.

aggregate(emp$SALARY~emp$DEPARTMENT\_ID+emp$JOB\_ID,emp,sum)

선생님답

x <- aggregate(SALARY ~ DEPARTMENT\_ID+JOB\_ID, emp, sum)

x

merge(x, dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","JOB\_ID","SALARY")]

aggregate(SALARY ~ DEPARTMENT\_NAME + JOB\_ID, merge(emp[,c('SALARY','DEPARTMENT\_ID','JOB\_ID')], dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME')]), sum)

[문제124] loc.csv 파일을 loc 변수로 로드하세요.

Toronto 지역에 근무하는 사원들의 LAST\_NAME,SALARY,DEPARTMENT\_ID,DEPARTMENT\_NAME,STREET\_ADDRESS 정보를 출력하세요.

Toronto 지역에 근무하는 사원들의 LAST\_NAME,SALARY,DEPARTMENT\_ID,DEPARTMENT\_NAME,STREET\_ADDRESS 정보를 출력하세요.

loc<-read.csv('c:/data/loc.csv',header = T,stringsAsFactors = FALSE)

loc

x<-loc[loc$CITY=='Toronto', ]

merge(emp,merge(x,dept))

선생님답

loc <- read.csv("c:/data/loc.csv",header=T,stringsAsFactors=F)

t <- merge(loc[loc$CITY=='Toronto',],dept,by='LOCATION\_ID')

merge(emp, t, by='DEPARTMENT\_ID')[, c('LAST\_NAME','SALARY','DEPARTMENT\_ID','DEPARTMENT\_NAME','STREET\_ADDRESS')]

merge(emp, merge(loc[loc$CITY=='Toronto',],dept,by='LOCATION\_ID'), by='DEPARTMENT\_ID')[, c('LAST\_NAME','SALARY','DEPARTMENT\_ID','DEPARTMENT\_NAME','STREET\_ADDRESS')]

l <- loc[loc$CITY=='Toronto',c('LOCATION\_ID','STREET\_ADDRESS')]

d <- dept[,c('DEPARTMENT\_ID','DEPARTMENT\_NAME','LOCATION\_ID')]

e <- emp[,c('LAST\_NAME','SALARY','DEPARTMENT\_ID')]

t <- merge(l,d,by='LOCATION\_ID')

merge(e,t, by='DEPARTMENT\_ID')[, c('LAST\_NAME','SALARY','DEPARTMENT\_ID','DEPARTMENT\_NAME','STREET\_ADDRESS')]

[문제125] 아래 화면의 결과처럼 출력해주세요.

부서이름 부서별급여

Administration 4400

Marketing 19000

Purchasing 24900

Human Resources 6500

Shipping 156400

IT 28800

Public Relations 10000

Sales 304500

Executive 63040

Finance 51608

Accounting 20308

소속부서X 7000

사원총급여 696456

x<-rbind(merge(aggregate(SALARY ~ DEPARTMENT\_ID, emp, sum), dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","SALARY")],list("소속부서X",sum(emp[is.na(emp$DEPARTMENT\_ID),'SALARY'])),list("사원총급여",sum(emp$SALARY)))

names(x)<-c("부서이름","부서별급여")

x

--------------------------쪼개서보자.

x<-merge(aggregate(SALARY ~ DEPARTMENT\_ID, emp, sum), dept, by="DEPARTMENT\_ID")[,c("DEPARTMENT\_NAME","SALARY")]

y<-list("소속부서X",sum(emp[is.na(emp$DEPARTMENT\_ID),'SALARY']))

z<-list("사원총급여",sum(emp$SALARY))

s<-rbind(x,y,z)

#문자하고숫자라서 list형으로 만들어줌 c()벡터로하면 형은 단하나 우선순위가 문자실수정수순이라 전부 문자형으로바뀜. 그래서 list형으로 해주는게 더 좋다.

#키컬럼이없으면 merge하기힘듬-> 열만맞으면 rbind로 묶어버리면됨

c.f

sum(emp$SALARY)-sum(aggregate(SALARY~DEPARTMENT\_ID,emp,sum))

#이렇게하면 안된다. 부서 아이디가 숫자형이어서 부서아이디까지다더해짐

sum(emp$SALARY)-sum(aggregate(SALARY~DEPARTMENT\_ID,emp,sum)**[2]**)

#이렇게하면 두번째컬럼인 샐러리값만 합을구해줄 수있다.

------------------------------

**subset()**

조건에 만족하는 데이터를 선택

**> emp[emp$DEPARTMENT\_ID==20,]**

EMPLOYEE\_ID FIRST\_NAME LAST\_NAME EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID SALARY COMMISSION\_PCT MANAGER\_ID

4 201 Michael Hartstein MHARTSTE 515.123.5555 20040217 MK\_MAN 13000 NA 100

5 202 Pat Fay PFAY 603.123.6666 20050817 MK\_REP 6000 NA 201

NA NA <NA> <NA> <NA> <NA> NA <NA> NA NA NA

DEPARTMENT\_ID

4 20

5 20

NA NA

**> subset(emp,DEPARTMENT\_ID==20) #NA값을 없애줌**

EMPLOYEE\_ID FIRST\_NAME LAST\_NAME EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID SALARY COMMISSION\_PCT MANAGER\_ID

4 201 Michael Hartstein MHARTSTE 515.123.5555 20040217 MK\_MAN 13000 NA 100

5 202 Pat Fay PFAY 603.123.6666 20050817 MK\_REP 6000 NA 201

DEPARTMENT\_ID

4 20

5 20

**na.omit(emp[emp$DEPARTMENT\_ID==20,]) #na값이 있으면 그 행전체를 삭제**

> na.omit(emp[emp$DEPARTMENT\_ID==20,]) #na값이 있으면 그 행전체를 삭제

[1] EMPLOYEE\_ID FIRST\_NAME LAST\_NAME EMAIL PHONE\_NUMBER HIRE\_DATE JOB\_ID

[8] SALARY COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID

<0 행> <또는 row.names의 길이가 0입니다>

뒤에 컬럼선택을해줘야함

emp[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID")]

subset(emp,select=c(LAST\_NAME,SALARY,DEPARTMENT\_ID))

names(emp)

emp[,c("LAST\_NAME","SALARY","DEPARTMENT\_ID")]

제외하고 봐야할 컬럼을 이용해서 많은 컬럼을 한번에도출하기

emp[,names(emp)%in% c("LAST\_NAME","SALARY","DEPARTMENT\_ID")]

emp[,**!names(emp)**%in% c("LAST\_NAME","SALARY","DEPARTMENT\_ID")]

names를 컬럼이름을 다 타이핑하긴 싫으니 제외시킬 영역만나열하고!을붙이면 많은 필요한 컬럼들을 뽑을 수 있음

원래c()벡터앞에 -는 안되지만 subset안에 c()에서 -(마이너스)된다.

subset(emp, select=c(LAST\_NAME, SALARY, DEPARTMENT\_ID))

subset(emp, salary>=10000, select=**-c**(LAST\_NAME, SALARY, DEPARTMENT\_ID))

**■sqldf**

sql을 이용해서 데이터를 처리한다.

패키지깔아줘야함

install.packages("sqldf")

library(sqldf)

**한번씩 해보기**

sqldf("select \* from emp ")

sqldf("select \* from dept")

sqldf("select \* from loc")

sqldf("select job\_id from ")

sqldf("select distinct job\_id from ")

sqldf("select \* from emp where department\_id=20 ")

sqldf("select \* from emp limit 10 ")

sqldf("select \* from ")

length(emp)

NROW(emp)

sum(emp$SALARY)

mean(emp$SALARY)

sd(emp$SALARY)

var(emp$SALARY)

max(emp$SALARY)

min(emp$SALARY)

sqldf("select count(employee\_id),sumsalary, avg(salary), variance(salary), stdev(salary), max(salary), min(salary) from emp")

sqldf("select department\_id, sum(salary)

from emp

group by department\_id")

sqldf("select department\_id, sum(salary)

from emp

group by department\_id

having sum(salary)>=20000")

sqldf("select department\_id, job\_id, sum(salary)

from emp

group by department\_id, job\_id

having sum(salary)>=20000")

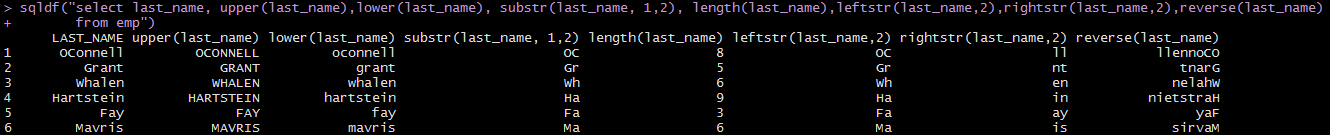
sqldf("select department\_id, job\_id, sum(salary)

from emp

group by department\_id, job\_id

having sum(salary)>=20000")

sqldf("select last\_name, upper(last\_name),lower(last\_name), substr(last\_name, 1,2), length(last\_name),leftstr(last\_name,2),rightstr(last\_name,2),reverse(last\_name) from emp")



sqldf("select salary/3, round(salary/3,0) from emp")

sqldf("select \* from emp where department\_id is null")

sqldf("select \* from emp where department\_id is not null")

#널값 추출할 때 NA안씀

sqldf("select \* from emp where department\_id in (10,20)")

sqldf("select \* from emp where salary between 10000 and 20000")

sqldf("select e. last\_name, d.department\_name

from emp e join dept d

on e.department\_id = d.department\_id")

sqldf("select e. last\_name, d.department\_name

from emp e join dept d

using(department\_id")

sqldf("select e. last\_name, d.department\_name

from emp e left outer join dept d

on e.department\_id = d.department\_id")

#right outer join 은 없다. from절 서로 위치만바꾸면 되니까

sqldf("select e. last\_name, d.department\_name

from dept d left outer join emp e

on e.department\_id = d.department\_id")

full outer join도 syntax없음 left랑 자리바꾼 left랑 서로 union쓰면되니까

sqldf("select e. last\_name, d.department\_name

from emp e left outer join dept d

on e.department\_id = d.department\_id

union

select e. last\_name, d.department\_name

from dept d left outer join emp e

on e.department\_id = d.department\_id")

차집합

sqldf("select e. last\_name, d.department\_name

from emp e left outer join dept d

on e.department\_id = d.department\_id

except

select e. last\_name, d.department\_name

from dept d left outer join emp e

on e.department\_id = d.department\_id")

위 minus 밑

서브쿼리

sqldf("select \* from emp

where salary > (select salary

from emp

where employee\_id = 150)")

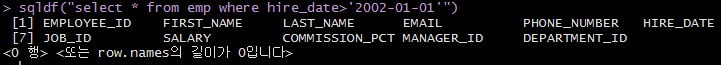
c.f)지금 날짜다루기는 잘안된다.

문자형 날짜는 잘 된다.

hire\_date date형식이면 date형식 똑같이 넣어도 잘 안된다.

hire\_date를 char로 바꾼 다음에 비교해보자.

sqldf("select \* from emp where hire\_date>'2002-01-01'") #안된다. 이런거 체크해보자.



[문제134] 자신의 관리자 보다 더 많은 급여를 받는 사원이름, 사원급여, 관리자이름, 관리자급여를 출력하세요.

emp<-read.csv('c:/data/emp.csv',header = T,stringsAsFactors = FALSE)

library(sqldf)

선생님답

sqldf("select w.last\_name, w.salary, m.last\_name, m.salary

from emp w, emp m

where w.manager\_id = m.employee\_id

and w.salary>m.salary

")

[문제135] 자신의 부서 평균 급여보다 많이 받는 사원들의 정보를 출력하세요.

sqldf(" select \*

from emp e

where salary > (select avg(salary)

from emp

department\_id = e.department\_id)

")

선생님답

sqldf("

select e2.employee\_id, e2.salary, e2.department\_id,e1.avgsal

from (select department\_id, avg(salary) avgsal

from emp

group by department\_id) e1, emp e2

where e1.department\_id = e2.department\_id

and e2.salary > e1.avgsal

")

[문제136] fruits\_sales.csv file 읽어 들인 후 과일 이름별 판매량, 판매합계를 구하세요.(tapply를 이용하세요)

qty price

apple 44 66000

banana 31 70000

berry 52 95100

orange 21 83000

a<-tapply(fruits\_sales$qty,fruits\_sales$name,sum)

b<-tapply(fruits\_sales$price, fruits\_sales$name, sum)

cbind(qty=a, price=b)

티어플라이는 데이터프레임, 벡터를 어레이형으로리턴해준다.

c.f)

rownames(as.data.frame())이런식으로하면 어떤열을 행이름으로바꿔줄수있음

[문제137] fruits\_sales.csv file 읽어 들인 후 과일 이름별 판매량, 판매합계를 구하세요.(aggregate를 이용하세요)

name qty price

apple 44 66000

banana 31 70000

berry 52 95100

orange 21 83000

x<-aggregate(qty~name,fruits\_sales,sum)

y<-aggregate(price~name,fruits\_sales,sum)

merge(x,y)

[문제138] fruits\_sales.csv file 읽어 들인 후 과일 이름별 판매량, 판매합계를 구하세요.(sqldf를 이용하세요)

qty price

apple 44 66000

banana 31 70000

berry 52 95100

orange 21 83000

sqldf(" select name, sum(qty) qty, sum(price) price

from fruits\_sales

group by name

")

[문제139] fruits\_sales.csv file 읽어 들인 후 년도별로 판매량 중에 가장 많은 판매를 한 년도를 출력해주세요.(tapply 를 이용하세요)

> a<-tapply(fruits\_sales$qty, fruits\_sales$year, sum)

> names(a)

[1] "2014" "2015" "2016" "2017"

> rownames(a)

[1] "2014" "2015" "2016" "2017"

> names(a[1])

[1] "2014"

> rownames(a)[1]

[1] "2014"

> #특정행을 뽑아낼 때는 이렇게 달라진다.

> names(a[a==max(a)])

[1] "2017"

> rownames(a)[a==max(a)]

[1] "2017"

티어플라이는 어레이형식으로나옴->행의연도만뽑기

[문제140] fruits\_sales.csv file 읽어 들인 후 년도별로 판매량 중에 가장 많은 판매를 한 년도를 출력해주세요. (aggregate를 이용하세요)

b<-aggregate(qty ~year, fruits\_sales, sum)

b[b$qty==max(b$qty),'year']

[문제141] fruits\_sales.csv file 읽어 들인 후 년도별로 판매량 중에 가장 많은 판매를 한 년도를 출력해주세요.(sqldf 를 이용하세요)

--------------

stack overflow <https://stackoverflow.com/>

aggregate(SALARY~DEPARTMENT\_ID,emp,mean)

aggregate(SALARY~DEPARTMENT\_ID+JOB\_ID,emp,mean)

**ddply**

데이터프레임을 분할하고 함수를 적용한 뒤 **데이터프레임으로** 결과를 반환하는 함수

install.packages("plyr")

library(plyr)

**ddply(data, 기준컬럼, 함수) #기준컬럼을 기준으로 모으는 것**

**ddply(emp,'DEPARTMENT\_ID', summarise,avg\_sal=mean(SALARY))**

DEPARTMENT\_ID avg\_sal

1 10 4400.000

2 20 9500.000

3 30 4150.000

4 40 6500.000

5 50 3475.556

6 60 5760.000

7 70 10000.000

8 80 8955.882

9 90 21013.333

10 100 8601.333

11 110 10154.000

12  **NA** 7000.000

#na값도나온다.

**ddply(emp, c('DEPARTMENT\_ID','JOB\_ID'),summarise,avg\_sal=mean(SALARY))**

**ddply(emp, 'DEPARTMENT\_ID', summarise, sum\_sal=sum(SALARY), avg\_sal =mean(SALARY))**

**ddply(emp, 'DEPARTMENT\_ID', summarise, cn=length(EMPLOYEE\_ID),sum\_sal=sum(SALARY), avg\_sal =mean(SALARY))**

**summarise : 기준컬럼의 데이터끼리 모은 후 함수에 적용 (grouping개념)**

[문제142] fruits\_sales.csv file 읽어 들인 후 과일 이름별 판매량, 판매합계를 구하세요.

ddply함수를 이용하세요.

ddply(fruits\_sales,'name',summarise, sum\_qty=sum(qty),sum\_price=sum(price))

ddply(fruits\_sales,'year',summarise, sum\_qty=sum(qty),sum\_price=sum(price))

ddply(fruits\_sales,'name',summarise, max(qty),min(price))

ddply(fruits\_sales,'year',summarise, max(qty),min(price))

**transform : 각 행별로 연산을 수행해서 행당값을 출력**

**이름별로 쭉 판매량 정렬해줌**

ddply(fruits\_sales,'name', **transform**, sum\_qty=sum(qty))

year name qty price sum\_qty

1 2014 apple 6 16000 44

2 2015 apple 10 10000 44

3 2016 apple 13 23000 44

4 2017 apple 15 17000 44

5 2014 banana 2 10000 31

6 2015 banana 7 25000 31

7 2016 banana 10 5000 31

8 2017 banana 12 30000 31

9 2014 berry 9 9000 52

10 2015 berry 15 35000 52

11 2016 berry 11 1100 52

12 2017 berry 17 50000 52

13 2014 orange 7 35000 21

14 2015 orange 3 15000 21

15 2016 orange 5 2500 21

16 2017 orange 6 30500 21

> ddply(fruits\_sales,'year',**transform**, sum\_qty=sum(qty))

year name qty price sum\_qty

1 2014 apple 6 16000 24

2 2014 banana 2 10000 24

3 2014 orange 7 35000 24

4 2014 berry 9 9000 24

5 2015 apple 10 10000 35

6 2015 banana 7 25000 35

7 2015 orange 3 15000 35

8 2015 berry 15 35000 35

9 2016 apple 13 23000 39

10 2016 banana 10 5000 39

11 2016 orange 5 2500 39

12 2016 berry 11 1100 39

13 2017 apple 15 17000 50

14 2017 banana 12 30000 50

15 2017 orange 6 30500 50

16 2017 berry 17 50000 50

판매량 계산

ddply(fruits\_sales**,'name',transform**, s\_qty=sum(qty),**pct\_qty=(100\*qty)/sum(qty))**

year name qty price s\_qty pct\_qty

1 2014 apple 6 16000 44 13.636364

2 2015 apple 10 10000 44 22.727273

3 2016 apple 13 23000 44 29.545455

4 2017 apple 15 17000 44 34.090909

5 2014 banana 2 10000 31 6.451613

6 2015 banana 7 25000 31 22.580645

7 2016 banana 10 5000 31 32.258065

8 2017 banana 12 30000 31 38.709677

9 2014 berry 9 9000 52 17.307692

10 2015 berry 15 35000 52 28.846154

11 2016 berry 11 1100 52 21.153846

12 2017 berry 17 50000 52 32.692308

13 2014 orange 7 35000 21 33.333333

14 2015 orange 3 15000 21 14.285714

15 2016 orange 5 2500 21 23.809524

16 2017 orange 6 30500 21 28.571429

> ddply(fruits\_sales**,'year',transform**, s\_qty=sum(qty),pct\_qty=(100\*qty)/sum(qty))

year name qty price s\_qty pct\_qty

1 2014 apple 6 16000 24 25.000000

2 2014 banana 2 10000 24 8.333333

3 2014 orange 7 35000 24 29.166667

4 2014 berry 9 9000 24 37.500000

5 2015 apple 10 10000 35 28.571429

6 2015 banana 7 25000 35 20.000000

7 2015 orange 3 15000 35 8.571429

8 2015 berry 15 35000 35 42.857143

9 2016 apple 13 23000 39 33.333333

10 2016 banana 10 5000 39 25.641026

11 2016 orange 5 2500 39 12.820513

12 2016 berry 11 1100 39 28.205128

13 2017 apple 15 17000 50 30.000000

14 2017 banana 12 30000 50 24.000000

15 2017 orange 6 30500 50 12.000000

16 2017 berry 17 50000 50 34.000000

**dplyr**

install.packages("dplyr")

library(dplyr)

emp[emp$DEPARTMENT\_ID==20,]

str(emp[emp$DEPARTMENT\_ID==20,])

**filter : 조건을 주어서 필터링하는 함수**

filter(emp, DEPARTMENT\_ID ==20) #이거하면 na안뜸

filter(emp, DEPARTMENT\_ID ==20)[,c('LAST\_NAME','SALARY')] #이렇게 뒤에서 조건제한

filter(emp, DEPARTMENT\_ID==20)[,2:5]

filter(emp, DEPARTMENT\_ID==30&SALARY>=3000)[,1:5]

**select : 여러컬럼이 있는 데이터 프레임에서 특정 컬럼만 선택사용**

select(emp,LAST\_NAME,SALARY)

select(emp,1,2)

select(emp,1:5)

select(emp,2,4,6)

select(emp,-SALARY,-COMMISSION\_PCT) #이 두컬럼만 제외시키고 보기

**%>%: 여러문장을 조합해서 사용하는 방법**

emp%>%

select(LAST\_NAME, JOB\_ID, SALARY)%>%

filter(SALARY>=20000)

emp%>%

select(LAST\_NAME, JOB\_ID, SALARY)%>%

filter(SALARY>=20000)

arrange(SALARY) #오름차순정렬

emp%>%

select(LAST\_NAME, JOB\_ID, SALARY)%>%

filter(SALARY>=20000)

arrange(desc(SALARY)) #내림차순정렬

주의점: 꼭 package 로드시키고 수행

**mutate:새로운 컬럼을 추가**

emp$SAL<-emp$SALARY\*12

emp$SAL<-NULL

mutate(emp,sal=SALARY\*12) #미리보기다.

df<-mutate(emp,sal=SALARY\*12) #실제적용시키려면 이런식으로 테이블명 잡아줘야한다.

str(df)

**emp%>%**

**select(LAST\_NAME, JOB\_ID,SALARY,COMMISSION\_PCT)%>%**

**mutate(ANNUAL\_SAL=SALARY\*12+**

**ifelse(is.na(COMMISSION\_PCT),0,SALARY\*COMMISSION\_PCT))%>%**

**arrange(desc(ANNUAL\_SAL))**

**emp%>%**

**select(LAST\_NAME, JOB\_ID,SALARY,COMMISSION\_PCT,DEPARTMENT\_ID)%>%**

**mutate(ANNUAL\_SAL=SALARY\*12+**

**ifelse(is.na(COMMISSION\_PCT),0,SALARY\*COMMISSION\_PCT))%>%**

**arrange(DEPARTMENT\_ID, desc(ANNUAL\_SAL))**

[문제143] emp 데이터 프레임을 새로운 df 이름으로 복제하세요.

df 데이터 프레임에 새로운 comm 컬럼을 생성하는데 COMMISSION\_PCT 값을 기반으로 값을 입력하시고

결측값은 기존 COMMISSION\_PCT의 평균 값으로 입력해주세요.(단 mutate함수를 이용하세요)

df<-emp

df<-df%>% mutate(comm=ifelse(is.na(COMMISSION\_PCT), mean(COMMISSION\_PCT,na.rm=T),COMMISSION\_PCT))

df

**summarise : 주어진 데이터 집계**

emp%>%

summarise(sum\_sal==sum(SALARY), mean\_sal(SALARY))

**> emp%>%**

**+ summarise(sum\_sal=sum(SALARY), mean\_sal=mean(SALARY))**

sum\_sal mean\_sal

1 696456 6508.935

**> emp%>%**

**+ summarise(max\_sal=max(SALARY),min\_sal=min(SALARY))**

max\_sal min\_sal

1 29040 2100

**> emp%>%**

**+ group\_by(DEPARTMENT\_ID)%>%**

**+ summarise(sum\_sal=sum(SALARY))**

# A tibble: 12 x 2 #데이터 프레임이다. 속성의 값 몇행 몇렬짜리이다.

DEPARTMENT\_ID sum\_sal

<int> <int>

1 10 4400

2 20 19000

3 30 24900

4 40 6500

5 50 156400

6 60 28800

7 70 10000

8 80 304500

9 90 63040

10 100 51608

11 110 20308

12 NA 7000

**> emp%>%**

**+ group\_by(JOB\_ID)%>%**

**+ summarise(sum\_sal=sum(SALARY))**

# A tibble: 19 x 2

JOB\_ID sum\_sal

<chr> <int>

1 AC\_ACCOUNT 8300

2 AC\_MGR 12008

3 AD\_ASST 4400

4 AD\_PRES 29040

5 AD\_VP 34000

6 FI\_ACCOUNT 39600

7 FI\_MGR 12008

8 HR\_REP 6500

9 IT\_PROG 28800

10 MK\_MAN 13000

11 MK\_REP 6000

12 PR\_REP 10000

13 PU\_CLERK 13900

14 PU\_MAN 11000

15 SA\_MAN 61000

16 SA\_REP 250500

17 SH\_CLERK 64300

18 ST\_CLERK 55700

19 ST\_MAN 36400

**#summarise\_at: 두번 따로 summarise해야할 것을 한번에 함**

**emp%>%**

**summarise\_at(c("SALARY","COMMISSION\_PCT"),sum, na.rm=TRUE)**

> emp%>%

+ summarise\_at(c("SALARY","COMMISSION\_PCT"),sum, na.rm=TRUE)

SALARY COMMISSION\_PCT

1 696456 7.8

**#summraise\_if: 조건에 만족하면 summarise실행**

**emp%>%**

**summarise\_if(is.numeric,sum,na.rm=TRUE) #numeric에 관해서는 모두 sum**

> emp%>%

+ summarise\_if(is.numeric,sum,na.rm=TRUE)

EMPLOYEE\_ID HIRE\_DATE SALARY COMMISSION\_PCT MANAGER\_ID DEPARTMENT\_ID

1 16371 2145889701 696456 7.8 13225 6700

**emp%>%**

**summarise\_if(is.numeric,c(sum,mean),na.rm=TRUE)**

> emp%>%

+ summarise\_if(is.numeric,c(sum,mean),na.rm=TRUE)

EMPLOYEE\_ID\_.Primitive("sum") HIRE\_DATE\_.Primitive("sum") SALARY\_.Primitive("sum")

1 16371 2145889701 696456

COMMISSION\_PCT\_.Primitive("sum") MANAGER\_ID\_.Primitive("sum")

1 7.8 13225

DEPARTMENT\_ID\_.Primitive("sum") EMPLOYEE\_ID\_function (x, ...) ...

1 6700 153

HIRE\_DATE\_function (x, ...) ... SALARY\_function (x, ...) ...

1 20055044 6508.935

COMMISSION\_PCT\_function (x, ...) ... MANAGER\_ID\_function (x, ...) ...

1 0.2228571 124.7642

DEPARTMENT\_ID\_function (x, ...) ...

1 63.20755

sales<-read.csv('c:/data/fruits\_sales.csv',header = T,stringsAsFactors = FALSE)

str(sales)

#가로 모양으로 나옴 이걸 분석하기 편하도록 세로모양으로 만들어 주는 패키지 reshape

**install.packages("reshape2")**

**library(reshape2)**

**#melt() : 컬럼이 많은 형태(wide)를 세로 방향으로 긴(long)형태로 변경**

melt(sales,id='year')

sales

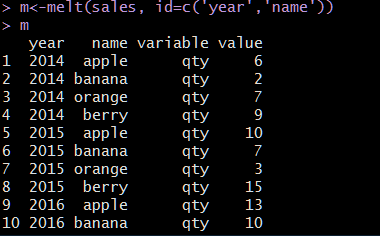
melt(sales,id='year')#year를 기준으로해서

melt(sales,id='name')#name을 기준으로해서

melt(sales,id='year','name')

m<-melt(sales, id=c('year','name'))

m



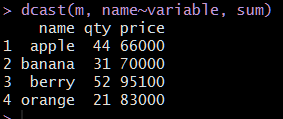
value 컬럼이 하나생성되었다.

**#dcast() : 세로(long)을 가로(wide) 형태로 변경**

dcast(m,year+name~variable)

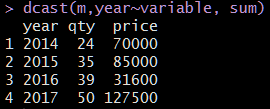


dcast(m, name~variable, sum)



dcast 에서 sum도됨

dcast(m,year~variable, sum)



**■grep: 동일한 문자열을 문자열 벡터에서 찾아서 인덱스 번호를 리턴하는 함수**

text<-c('a','ab','acb','accb','accccb')

**> grep('a',text)**

[1] 1 2 3 4 5

**> grep('ab',text)**

[1] 2

**> grep('acb',text)**

[1] 3

**> grep('c',text, value=TRUE)**

[1] "acb" "accb" "accccb"

**> grep('ac\*b',text,value=TRUE) # \* 적어도 0번 이상 매칭되면 출력**

[1] "ab" "acb" "accb" "accccb"

**> grep('ac+b',text,value=TRUE) # + 적어도 1번이상 매칭되면 출력**

[1] "acb" "accb" "accccb"

**> grep('ac?b',text,value=TRUE) # ? 0번 또는 1번 매칭되면 출력**

[1] "ab" "acb"

**> grep('ac{2}b',text,value=TRUE) # {n} n번 매칭되면 출력**

[1] "accb"

**> grep('ac{2,}b',text,value=TRUE) # {n,} n번이상 매칭되면 출력**

[1] "accb" "accccb"

**> grep('ac{2,3}b',text,value=TRUE) #{n,m} n 부터 m번 매칭되면 출력**

[1] "accb"

**> grep('^ab',text,value=TRUE) # ^ ab로시작**

[1] "ab"

**> grep('ab$',text, value=TRUE) # $ ab로끝**

[1] "ab"

**> grep('\\bab',text,value=TRUE) #\\b ab시작되는 문자를 찾는데 빈문자열 뒤에 시작되는 ab도 찾는다. 공백을의미**

[1] "ab"

**> grep('\\bab',text,value=TRUE)**

[1] "abcd" "c abd"

**> text<-c('^ab','ab','abc','abd','abe','ab 12')**

**> grep('ab',text,value=TRUE)**

[1] "^ab" "ab" "abc" "abd" "abe" "ab 12"

**> grep('ab.',text,value=TRUE) # . 어떤 문자 하나 매칭**

[1] "abc" "abd" "abe" "ab 12"

**> grep('ab[c,e]',text,value=TRUE) # [..]리스트 안에 있는 문자 매칭**

[1] "abc" "abe"

**> grep('ab[c,d,e]',text,value=TRUE)**

[1] "abc" "abd" "abe"

**> grep('ab[c-e]',text,value=TRUE) #[n-m] n부터 m까지 문자 매칭**

[1] "abc" "abd" "abe"

**> grep('ab[^c]',text,value=TRUE) #[^] NOT의 의미 이 글자만아니면돼**

[1] "abd" "abe" "ab 12"

**> grep('\\^',text,value=TRUE) #** [오류! 하이퍼링크 참조가 잘못되었습니다.](\\\\^) **^시작글자를 찾는다.**

[1] "^ab"

**> grep('$',text,value=TRUE) # $가 끝의 의미라 전부다나옴**

[1] "^ab" "ab" "abc" "abd" "abe" "ab 12"

**text<-c('^ab','$ab','abc','abd','abe','ab 12')**

**> grep('\\$',text,value=TRUE) #$를 문자로 본다.**

[1] "$ab"

**text<-c('sql','SQL','Sql100','PLSQL','plsql','R','r','r0','python','PYTHON','pyth0n','python#')**

**> grep('[0-9]',text,value=T)**

[1] "Sql100" "r0" "pyth0n"

**> grep('[[:digit:]]',text,value=T)**

[1] "Sql100" "r0" "pyth0n"

둘이 똑같다.

**> grep('[[:digit:]]',text,value=T)**

[1] "Sql100" "r0" "pyth0n"

**> grep('[[:upper:]]',text,value=T) #대문자만**

[1] "SQL" "Sql100" "PLSQL" "R" "PYTHON"

**> grep('[[:lower:]]',text,value=T) #소문자만**

[1] "sql" "Sql100" "plsql" "r" "r0" "python" "pyth0n"

[8] "python#"

text<-c('sql','SQL','Sql100','PLSQL','plsql','R','r','r0','python','PYTHON','pyth0n','python#','100')

grep('[0-9]',text,value=T)

**> grep('[[:alpha:]]',text,value=T) #문자만 포함**

[1] "sql" "SQL" "Sql100" "PLSQL" "plsql" "R" "r" "r0"

[9] "python" "PYTHON" "pyth0n" "python#"

**> grep('[[:alnum:]]',text,value=T) #문자 숫자 다포함**

[1] "sql" "SQL" "Sql100" "PLSQL" "plsql" "R" "r" "r0"

[9] "python" "PYTHON" "pyth0n" "python#" "100"

**> grep('[[:punct:]]',text,value=T) #구두점, 특수문자 포함하는애들 뽑기**

[1] "python#"

text<-c('sql','SQL','Sql100','PLSQL','plsql','R','r','r0','python','PYTHON','pyth0n','python#','100','\*','^')

**> grep('[[:punct:]]',text,value=T) #둘다나온다.**

[1] "python#" "\*" "^"

**#순수하게 기호만 뽑아내는 법**

**> gregexpr("[\*|$|^]",text)==1**

[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

[14] TRUE TRUE

**> which(gregexpr("[\*|$|^]",text)==1)**

[1] 14 15

**> text[gregexpr("[\*|$|^]",text)==1]**

[1] "\*" "^"

**grep('Steven',emp$FIRST\_NAME,value=T)**

**grep('Stephen',emp$FIRST\_NAME,value=T)**

**grep('Steven|stephen',emp$FIRST\_NAME,value=T)**

**EMP[gregexpr('Ste[v|ph]en',emp$FIRST\_NAME)==1,]**

**grep('Steven', emp$FIRST\_NAME, value=T)**

**emp[gregexpr('Ste(v|ph)en', emp$FIRST\_NAME)==1,]**

**x<-c('Steven','Stephen')**

**grep(paste(x,collapse='|'),emp$FIRST\_NAME,value = T)**

Steven과 Stephen을 'Steven|Stephen'으로 해서 찾아야한다면 이런식으로해주면된다.

x값안에 엄청 많은 값이 존재하면 일일이 수작업 할 수 없으니까.

**library(stringr)**

**text**

[1] "sql" "SQL" "Sql100" "PLSQL" "plsql"

[6] "R" "r" "r0" "python" "PYTHON"

[11] "pyth0n" "python#" "100" "\*" "^"

**#str\_detect : 특정 문자가 있는지를 검사해서 TRUE/FALSE를 출력하는 함수**

**> str\_detect(text,'SQL')#대문자 SQL찾기**

[1] FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE

[10] FALSE FALSE FALSE FALSE FALSE FALSE

**> text[str\_detect(text,'SQL')]**

[1] "SQL" "PLSQL"

**> which(str\_detect(text,'SQL'))**

[1] 2 4

**> text[str\_detect(text,'^s')]**

[1] "sql"

**> text[str\_detect(text,'n$')]**

[1] "python" "pyth0n"

**> text[str\_detect(text,'^[sS]')]**

[1] "sql" "SQL" "Sql100"

**> text[str\_detect(text,'[qQ]')]**

[1] "sql" "SQL" "Sql100" "PLSQL" "plsql"

**> text[str\_detect(text,'[sS]')]**

[1] "sql" "SQL" "Sql100" "PLSQL" "plsql"

**str\_count#주어진 단어에서 해당 글자가 몇개 존재하는지 알려주는 함수**

**> str\_count(text,'s')**

[1] 2 3

**> str\_count(text,'S')**

[1] 0 0

**> str\_count('[sS]')**

[1] 4

**#str\_c( ) : 문자열 합쳐서 출력하는 함수**

**str\_c('R','빅데이터 분석')**

> str\_c('R','빅데이터 분석')

[1] "R빅데이터 분석"

**> str\_c(text,' 은 데이터 분석 하기 위해 좋은 언어는',text,'이다')**

[1] "sqls 은 데이터 분석 하기 위해 좋은 언어는sqls이다"

[2] "ssqls 은 데이터 분석 하기 위해 좋은 언어는ssqls이다"

**> str\_c('R','빅데이터 분석',collapse=' ')**

[1] "R빅데이터 분석"

#변화없다. 직접은안되고 벡터변수에모양 만들어놓고해야한다.

**text<-c('R','빅데이터분석')**

**> str\_c(text,collapse=',')**

[1] "R,빅데이터분석"

**> str\_dup('파도 소리를 듣고싶다.',10)**

[1] "파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다.파도 소리를 듣고싶다."

**#str\_length( ) : 주어진 문자열의 길이를 출력하는 함수**

str\_length('해운대가고싶다')

[1] 7

**#str\_locate() :주어진 문자열에서 특정 문자가 처음으로 나오는 위치**

**> str\_locate('january','a')**

start end

[1,] 2 2

**> str\_locate\_all('january','a') #모든 a에 관련 정보를..**

[[1]]

start end

[1,] 2 2

[2,] 5 5

**> str\_locate\_all('janauary','a')**

[[1]]

start end

[1,] 2 2

[2,] 4 4

[3,] 6 6

기존의 sub

**str\_replace() : 주어진 문자열에서 변경전 문자를 변경후 문자로 바꾸는 함수**

**> str\_replace('빅데이터 분석','빅데이터','가치') # (원본, 치환할단어, 치환단어 )**

[1] "가치 분석"

**> str\_replace('banana','a','\*') #첫번째 a만 치환**

[1] "b\*nana"

**> str\_replace\_all('banana','a','\*') # 모든 a 치환**

[1] "b\*n\*n\*"

데이터 정제작업을 위해서 이런함수들을 쓴다.

**#str\_split( ) : 주어진 데이터셋에서 지정된 기호를 기준으로 분리하는 함수**

**> str<-str\_c('sql','/','plsql','/','r')**

**> str**

[1] "sql/plsql/r"

**> str\_split(str,'/')**

[[1]]

[1] "sql" "plsql" "r"

**#str\_sub() : 주어진 문자열에서 지정된 길이 만큼의 문자를 잘라내는 함수**

**> str\_sub('행복하게 살자', start=1,end=2)**

[1] "행복"

**> str\_sub('행복하게 살자', start=-2) #뒤에서부터 두글자**

[1] "살자"

**> str\_sub('행복하게 살자', start=1)**

[1] "행복하게 살자"

**> str\_sub('행복하게 살자', start=1,end=1)**

[1] "행"

**#' R ' 이런문자에서 공백문자를 제거하고싶을때**

**str\_trim( ) : 접두, 접미 부분에 공백문자를 제거하는 함수**

**> str\_trim(' R ')**

[1] "R"

그래프를 그릴 때 무작정 그리는 것이아니다.

자료에 따라 그래프의 유형이 달라진다.

**통계학**

- 관심대상에 대해 관련된 **데이터**를 **수집**하고 그 데이터를 **요약, 정리** 하여 이로부터 불확실한 사실에 대한 **결론**이나 일반적인 **규칙성을 찾는 학문**이다.

- 해결하고 싶은 문제가 있기에 그에 따른 답을 찾기위해서 분석을 한다.

**자료**

- 문제해결을 위한 원재료로서 처리되지 않은 숫자, 문자, 일련의 사실이나 기록들의 모임

- 어떠한 갖치판단을 할 수 있는 근거가 되는 재료

**양적자료**

- 숫자, 크기에 관심, 측정되는 값

- 연속형자료 : 키, 몸무게 등 무언가 소수점있는 그런 자료 딱떨어지지 않는 값

- 이산형자료 : 출생아수, 남학생 수, 왼손잡이의 수, 오른손잡이의 수 딱떨어지는값

**질적자료**

- 숫자가 내포하는 의미가 있다.

- 순위형 자료: 학점(A,B,C,D,F), 매우그렇다, 보통이다. 의미가있으면서 순서가있는 것

- 명목형 자료: 성별(남,여, 남자는 1로, 여자는 2로), 거주지역, 혈액형

요약방법 자료정리 그래프

---------------------------------------------------------------------------------------------------------질적자료 도표, 그래프 도수분포표,

분할표

---------------------------------------------------------------------------------------------------------

양적자료 수치,그래프 평균, 분산, 표준편차 히스토그램

중앙값,최소값,최대값 상자도표

시계열도표

산점도

**기술통계**

- 자료 수집, 정리

- 자료 형태를 표현하는 것

- 자료의 특성값 도출

**추측통계**

- 표본으로부터 관찰하고자 하는 특성값을 도출해내는 방법

- 이를 바탕으로 모집단의 특성 파악

**<pie chart>**

- 질적자료에 대한 상대도수분포를 나타내기 위해 일반적으로 사용되는 그래프

- 원을 그린 후 그 원에 각 계급의 상대도수에 대응하는 면적 또는 부분으로 나눈다.

A회사 : 100억

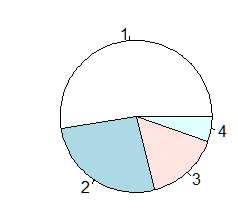
B회사 : 50억

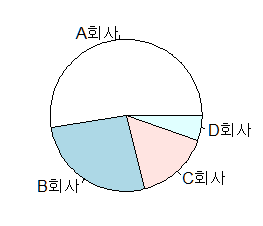
C회사 : 30억

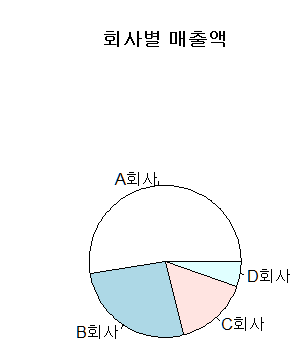
D회사 : 10억

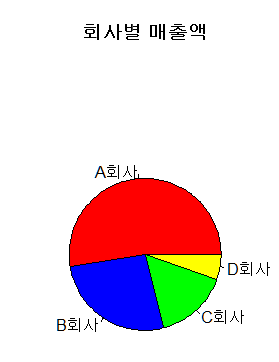
s<- c(100,50,30,10)

company<-c('A회사','B회사','C회사','D회사')

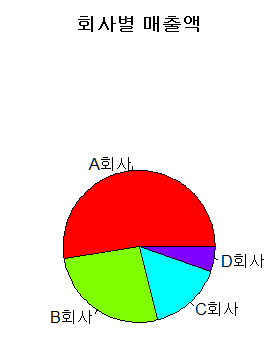
> pie(s)

pie(s, labels=company)

pie(s, labels=company,main='회사별 매출액')



pie(s, labels=company,main='회사별 매출액', col=c("red","blue","green","yellow"))



pie(s, labels=company,main='회사별 매출액', col=c(**rainbow**(length(s))))

rainbow : 무지개색

heat.colors(12) : 적색, 황색에 치우친 색

terrain.colors(12) : 지구 지형색

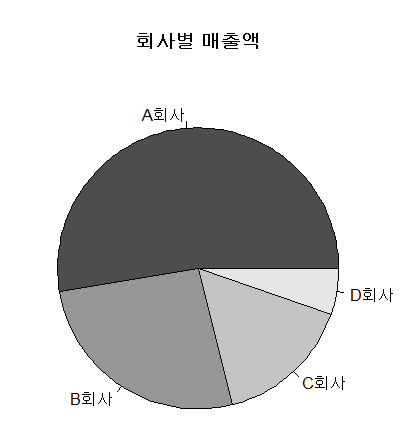
topo.colors(12) : 청색에 가까운색

cm.colors(12) : 핑크, 블루

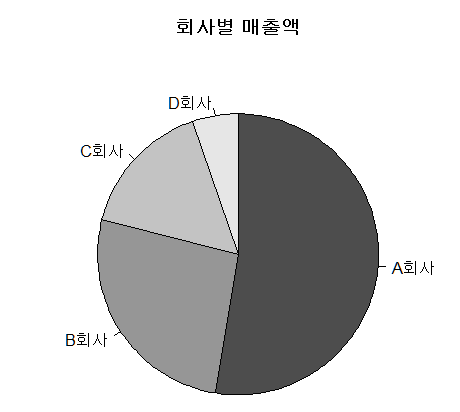
?pie 에서 색깔옵션보면된다.

#clockwise

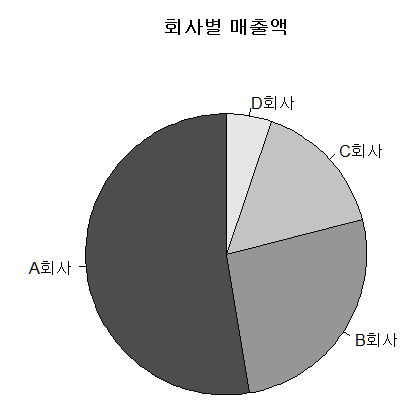
시계방향(TRUE)으로 회전할지 반시계방향(FALSE)으로 회전할지 지정, F가 기본값



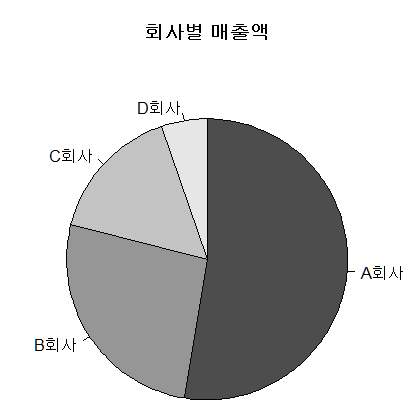
pie(s, labels=company, main='회사별 매출액', col=gray.colors(length(s)), clockwise=F)



pie(s, labels=company, main='회사별 매출액', col=gray.colors(length(s)), clockwise=T)



pie(s, labels=company, main='회사별 매출액', col=gray.colors(length(s)), clockwise=F, **init.angle=90**)



pie(s, labels=company, main='회사별 매출액', col=gray.colors(length(s)), clockwise=T , init.angle=90)

**> p<-round(s/sum(s)\*100)**

**> p**

[1] 53 26 16 5 #퍼센트로나온다.

**> label<-paste(company,p)**

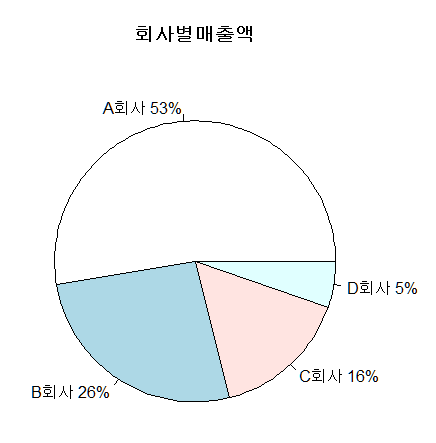
**> label**

[1] "A회사 53" "B회사 26" "C회사 16" "D회사 5"

**> label<-paste(label,'%',sep="")**

**> label**

[1] "A회사 53%" "B회사 26%" "C회사 16%" "D회사 5%"



> pie(s, labels=label, main="회사별매출액")

이런식으로 가공할 수 있다.

[문제144] 부서별 급여의 총액을 pie chart를 생성하세요.

emp<-read.csv("c:/data/emp.csv",header = T,stringsAsFactors = FALSE)

dp.sum\_sal<-tapply(emp$SALARY,emp$DEPARTMENT\_ID,sum)

dp\_group<-(emp$DEPARTMENT\_ID)

pie(dp.sum\_sal, labels=dp\_group, main = '부서별 급여 총액',col=c(rainbow(length(s))), clockwise=F, init.angle=180)

선생님답

df<-aggregate(SALARY~DEPARTMENT\_ID, emp, sum)

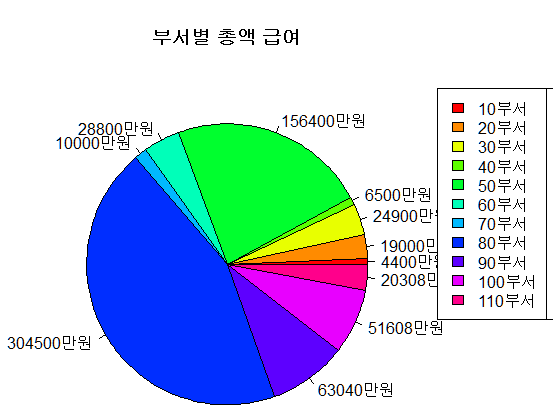
label1<-paste(df$DEPARTMENT\_ID,"부서",sep="")

label2<-paste(df$SALARY,"만원",sep="")

pie(df$SALARY,labels=label2,main="부서별 총액 급여",col=rainbow(length(df$SALARY)))

**legend(1,1,label1,fill=rainbow(length(df$SALARY)))**

범례표시가능

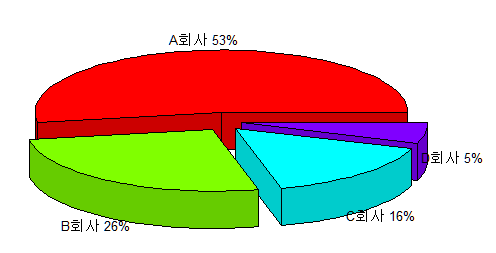


3d 파이

install.packages("plotrix")

library(plotrix)

**pie3D(s, labels=label, explode=0.1,labelcex=0.8)**



explode : 부채꼴들의 간격

labelcex : label의 문자 크기

**#막대그래프 barplot**

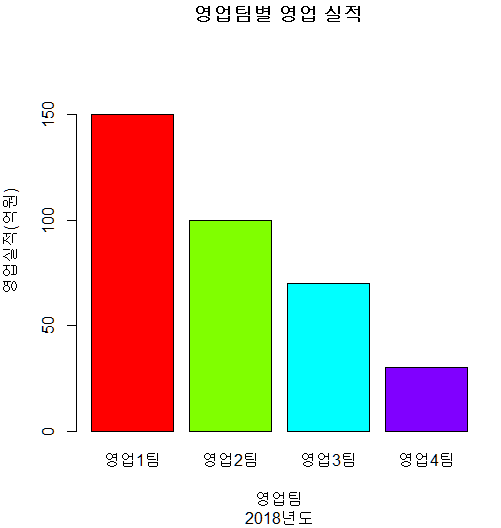
- 질적자료를 표현하는 그래프

- 각 계급 이름 위에 고정 너비의 막대를 그리고 도수에 따라 적절하게 막대의 길이를 표현한다.

c.f) 히스토그램은 막대의 너비가 달라진다.

**sales<-c(150,100,70,30)**

**team<-c("영업1팀","영업2팀","영업3팀","영업4팀")**



**barplot(height=sales, width=0.5,names.arg=team,horiz = FALSE, col=rainbow(length(sales)),main="영업팀별 영업 실적",xlab="영업팀",ylab="영업실적(억원)",ylim=c(0,180),sub="2018년도")**

height: 막대 크기를 나타내는 벡터(숫자)

width : 막대너비

names.arg: 막대 아래 출력되는 이름

col : 막대 색상

main: wpahr

sub: 부제목

horiz: TRUE(수평막대), FALSE(수직막대)

xlab : x축 이름

ylab : y축 이름

ylim : y축 크기

xlim : x축 크기

막대그래프 위에 값을 넣어주기

**bp<-**barplot(height=sales, width=0.5,names.arg=team,horiz = FALSE, col=rainbow(length(sales)),main="영업팀별 영업 실적",xlab="영업팀",ylab="영업실적(억원)",ylim=c(0,180),sub="2018년도")

**text(x=bp, y= sales, labels =round(sales),pos=3)**



pos = 1 : 막대 끝선의 아래쪽

2 : 막대 끝 선의 왼쪽

3: 막대 끝 선의 위쪽

4: 막대 끝 선의 오른쪽

label <- paste(sales,'억원',sep="") 이거해주고 albels = label 해주면 막대그래프에 억원들어감

**#stacked bar chart(스택형 바차트)**

> barplot(data, names.arg=name, main= "영업팀별 실적", xlab="영업팀",ylab="판매실적(억원)", ylim=c(0,30), col = c("darkblue","red"))

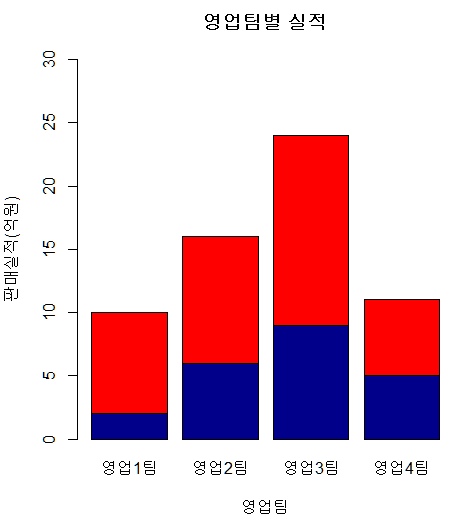
> x1<-c(2,6,9,5)

> x2<-c(8,10,15,6)

> data<- rbind(x1,x2)

> name<-c("영업1팀","영업2팀","영업3팀","영업4팀")

> barplot(data, names.arg=name, main= "영업팀별 실적", xlab="영업팀",ylab="판매실적(억원)", ylim=c(0,30), col = c("darkblue","red"))

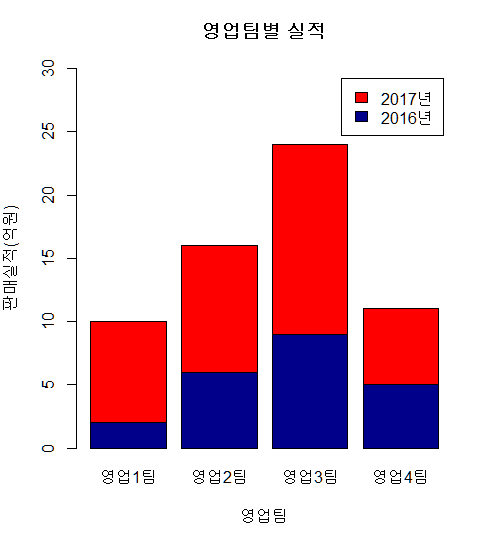


rbind로 쌓여있는 모양을 만들어 줘야한다.

범례추가해주기

**label<-c('2016년','2017년')**

barplot(data, names.arg=name, main= "영업팀별 실적", xlab="영업팀",ylab="판매실적(억원)", ylim=c(0,30), col = c("darkblue","red"),**legend.text=label**)



정리되어있는 데이터가 중요한 것이다. 시각화한것일 뿐

[문제145] 성별 현황을 조사 자료를 이용하여 성별 인구 비율을 원형 차트로 만드세요.

성별 현황

구분 조사수

---- ------

남자 226965

여자 241319

qty<-c(226965,241319)

gen<-c('남자','여자')

label1<-paste(qty,gen)

pie(qty,labels=label1, main="성별 현황 조사 자료",col=rainbow(length(qty)))

선생님답

survey<-c(226965,241319)

gender<-c('남자','여자')

proportion<-round((survey/sum(survey)\*100))

label<-paste(proportion,'%',sep="")

colors()

pie(survey,labels=label, main="성별 현황", col=c("blue","red"))

legend("topright", legend=gender, cex=0.6,fill=c("blue","red"))

library(plotrix)

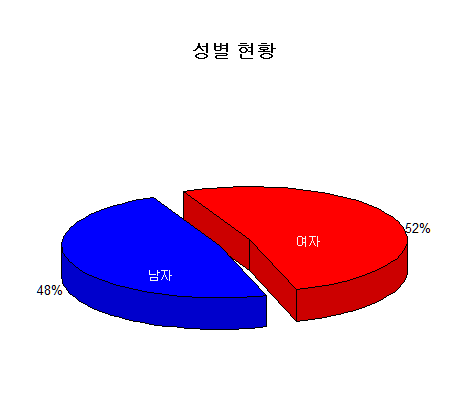
pie3D(proportion,labels=label, start=2, explode=0.1, labelcex = 0.8,main="성별 현황",col=c ("blue","red"))

legend("topright", legend=gender, cex=0.6,fill=c("blue","red"))

#원형안에다가 남자, 여자를 넣고싶을 때

p<-pie3D(proportion,labels=label, start=2, explode=0.1, labelcex = 0.8,main="성별 현황",col=c ("blue","red"))

pie3D.labels(p,labels=gender,labelrad=0.5,labelcex = 0.7,labelcol = "whitesmoke") #p를 수행할건데 labels=gender로



[문제146] 성별 현황을 조사 자료를 이용하여 성별 인구수를 막대그래프로 만드세요.

성별 현황

구분 조사수

---- ------

남자 226965

여자 241319

barplot(height=qty,width=0.2,names.arg=gen,horiz=FALSE,col=rainbow(length(qty)),main="성별현황조사자료")

선생님답

barplot(height=survey, width=1,names.arg=gender, horiz=FALSE, col=c("blue","red"),main="성별 현황",xlab="성별",axes=TRUE,ylim=c(0,291319))

# axes=FALSE y축을 없앰

bp<-barplot(height=survey, width=1,names.arg=gender, horiz=FALSE, col=c("blue","red"),main=" 성별현황",xlab="성별",axes=TRUE,ylim=c(0,291319))

label<-paste(survey,"명",sep="")

text(x=bp,y=survey,labels=label, pos=3)

[문제147] 부서별 인원수 막대그래프를 생성하세요. 단 부서없는 사원들의 인원수도 포함하세 요.

x<-tapply(emp$EMPLOYEE\_ID,ifelse(is.na(emp$DEPARTMENT\_ID),'NA',emp$DEPARTMENT\_ID), NROW)

bp2<-barplot(height = x, width=1,names.arg =names(x), horiz=FALSE,col = rainbow(length(x)),main='부서별인원수',xlab='부서',ylab='인원수',ylim=c(0,50))

text(x=bp2,y=x,labels=paste(x,'명'), pos=3)

종현이 답

library(plyr)

a<-ddply(emp,'DEPARTMENT\_ID',summarise,emp\_cn=length(EMPLOYEE\_ID))

a$DEPARTMENT\_ID<-ifelse(is.na(a$DEPARTMENT\_ID),'NULL',a$DEPARTMENT\_ID)

b<-barplot(height=a$emp\_cn,names.arg=a$DEPARTMENT\_ID,horiz=FALSE,xlab='부서번호',col=rainbow(length(a$DEPARTMENT\_ID)),axes=TRUE,ylim=c(0,50),ylab='인원수',main='부서별 인원수')

c<-text(x=b,y=a$emp\_cn,labels=a$emp\_cn,pos=3)

선생님답

dept\_cn<-aggregate(EMPLOYEE\_ID~ifelse(is.na(DEPARTMENT\_ID),'부서없음',DEPARTMENT\_ID),emp,length)

names(dept\_cn)<-c('dept\_id','cn')

bp<-barplot(dept\_cn$cn, names.arg=dept\_cn$dept\_id,main="부서별 인원수", xlab="부서", col=rainbow(length(dept\_cn$dept\_id)),cex.name=0.7,axes=FALSE)

label<-paste(dept\_cn$cn,'명',sep="")

text(x=bp,y=dept\_cn$cn, labels=label, cex=0.9,pos=3)

--------------------

**#stacked bar chart**

x1<-c(2,6,9,5)

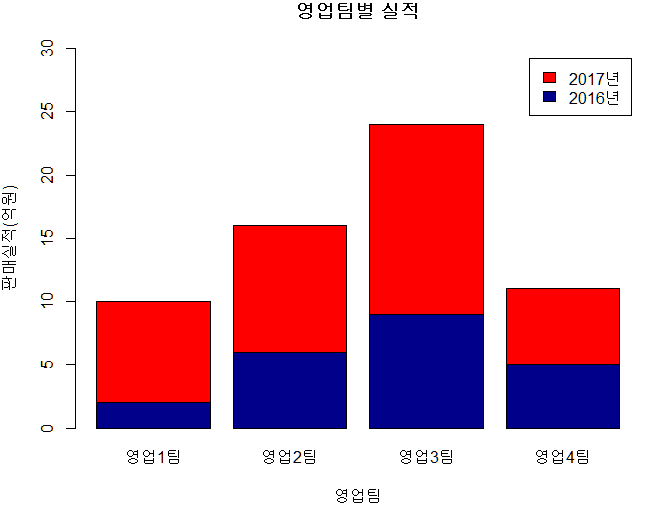
x2<-c(8,10,15,6)

data<-rbind(x1,x2)

name<-c("영업1팀","영업2팀","영업3팀","영업4팀")

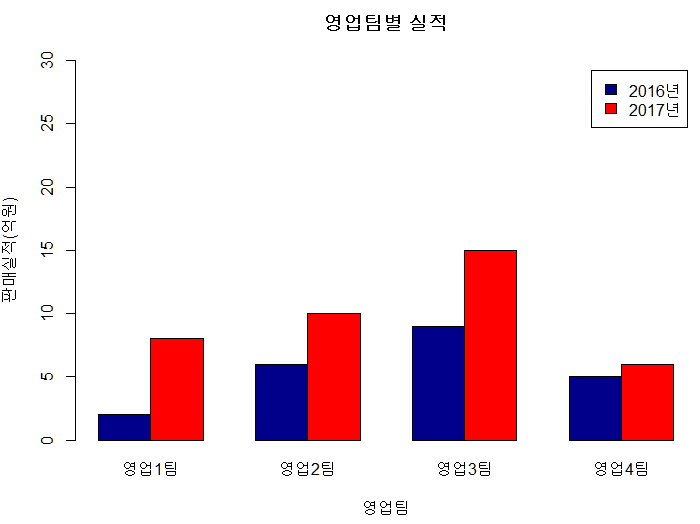
label<-c("2016년","2017년")

barplot(data,names.arg=name, main="영업팀별 실적",xlab="영업팀",ylab="판매실적(억원)",ylim=c(0,30),legend.text=label, col=c("darkblue","red"))



**#group bar chart**

barplot(data,names.arg=name, main="영업팀별 실적",xlab="영업팀",ylab="판매실적(억원)",ylim=c(0,30),legend.text=label, col=c("darkblue","red"),**beside=T**)



그룹바차트

[문제148] 부서 이름별 급여 총액에 대해서 막대그래프를 생성하세요.단 부서없는 사원 들의 인원수도 포함하세요.

선생님답

dept\_sal <- sqldf('SELECT department\_name dept\_name, sum(salary) sum\_sal

FROM emp e LEFT OUTER JOIN dept d

ON e.department\_id=d.department\_id

GROUP BY department\_name')

bp <- barplot(dept\_sal$sum\_sal,

names.arg=dept\_sal$dept\_name,

col=rainbow(length(dept\_sal$dept\_name)),

main='Salary Bar Chart',

xlab='Dept Name', ylab='Salary',

ylim=c(0,350000),

cex.names=0.5,las=2) #las=2 : 이거 써주면 x축 이름을 세로로 써줄 수있다.

text(x=bp, y=dept\_sal$sum\_sal,

labels=dept\_sal$sum\_sal, pos=1, cex=0.5)

text(x=bp[is.na(dept\_sal$dept\_name)],

y=dept\_sal$sum\_sal[is.na(dept\_sal$dept\_name)],

labels='부서(x)',

pos=3,

col='red')

text(x=bp[dept\_sal$sum\_sal==max(dept\_sal$sum\_sal)],

y=dept\_sal$sum\_sal[dept\_sal$sum\_sal==max(dept\_sal$sum\_sal)],

labels='최고급여',

pos=3,

col='red')

[문제149] fruits\_sales.csv을 읽어 들인 후 년도별, 과일이름별 판매량을 그룹형 막대 그 래프로 만드세요.

sales<-read.csv('c:/data/fruits\_sales.csv',header=T,stringsAsFactors = FALSE)

names(sales)

syn<-tapply(sales$qty,list(sales$name,sales$year),sum)

syn

rownames(syn)

synbp<-barplot(syn, beside = T, main="년도별 과일 판매량", xlab="년도별 과일분류", ylab="qty", ylim=c(0,20), col=rainbow(4),legend.text=rownames(syn))

text(x=synbp, y=syn, label=syn, pos=3 )

선생님답

f\_sale <- read.csv('c:/data/fruits\_sales.csv', stringsAsFactors=F)

f\_sale

x<-tapply(f\_sale$qty, list(f\_sale$name, f\_sale$year), sum)

bp<-barplot(x,

xlab='년도', ylab='판매량',

col=c('red','yellow','green','orange'),

beside=TRUE,

ylim=c(0,20))

legend('topright',

**title='과일',**

legend=rownames(x),

**pch=15**, col=c('red','yellow','green','orange'),

cex=0.7)

title(main='년도별 과일판매량')

**library(plotrix)**

#barlabels(bp,matrix(x),matrix(x),cex=0.8,prop=1.04)

boxed.labels(bp,matrix(x),matrix(x),cex=0.8,border=FALSE,srt=0,bg=c ('red','yellow','green','orange'))

#plotrix 패키지안의 메소드들. matrix를 사용해서 순서대로 쭉 펼침

srt각도

bg 색깔

#한화면에 그래프를 여러개 표현

graphics.off()

par(mfrow=c(1,1))

par(mfrow=c(2,2))

?barlabels

Usage

barlabels(xpos,ypos,labels=NULL,cex=1,prop=0.5,miny=0,offset=0,...)

[문제150] exam.csv file에는 학생들의 시험점수가 있습니다. 학생들의 SQL 점수를 막대 그래프로 출력해주세요.

exam<-read.csv('c:/data/exam.csv',header = T,stringsAsFactors = FALSE)

library(sqldf)

sql\_ng<-sqldf('select name, grade

from exam

where subject = "SQL"')

bp<-barplot(sql\_ng$grade, names.arg= sql\_ng$name, xlab='name',ylab='SQL\_grade',ylim=c (0,100),col='gray',las= 2)

text(x=bp, y=sql\_ng$grade, labels =sql\_ng$grade,pos=3,cex=0.5 )

[문제151] exam.csv file에는 학생들의 시험점수가 있습니다. 학생들의 R 점수를 막대그 래프로 출력해주세요.

r\_ng<-sqldf('select name, grade

from exam

where subject = "R"')

bp<-barplot(r\_ng$grade, names.arg= r\_ng$name, xlab='name',ylab='SQL\_grade',ylim=c (0,100),col='gray',las= 2)

text(x=bp, y=r\_ng$grade, labels =r\_ng$grade,pos=3,cex=0.5 )

[문제152] exam.csv file에는 학생들의 시험점수가 있습니다. 학생들의 PYTHON 점수를 막대그래프로 출력해주세요.

python\_ng<-sqldf('select name, grade

from exam

where subject = "PYTHON"')

bp<-barplot(python\_ng$grade, names.arg= python\_ng$name, xlab='name',ylab='SQL\_grade',ylim=c(0,100),col='gray',las= 2)

text(x=bp, y=python\_ng$grade, labels =python\_ng$grade,pos=3,cex=0.5 )

[문제153] exam.csv file에는 학생들의 시험점수가 있습니다. 학생들의 과목 총 점수를 막 대그래프로 출력하세요.

sum\_ng<-sqldf('select name, sum(grade) SG

from exam

group by "name"')

bp<-barplot(sum\_ng$SG, names.arg= sum\_ng$name, xlab='name',ylab='SQL\_grade',ylim=c (0,300),col='gray',las= 2)

text(x=bp, y=sum\_ng$SG, labels =sum\_ng$grade,pos=3,cex=0.5 )

[문제154] 학생들의 이름을 기준으로 과목점수를 스택형 막대그래프로 생성하세요.

t <- tapply(exam$grade, list(exam$subject, exam$name), sum)

bp <- barplot(t, names.arg=names(t), ylim=c(0,350),

xlab='이름', ylab='성적',col=c('blue','green','purple'),

main='과목별 점수',las=2)

legend('topright',legend=rownames(t), title='과목', pch=15,

col=c('blue','green','purple'),

cex=0.9, pt.cex=1)

abline(h=seq(100,300,100),col='red',lty=2)

library(plotrix)

barlabels(bp,t,bg=c('blue','green','purple'),border=c('blue','green','purple'))

total<-apply(t,2,sum)

text(bp,y=total\*1.1,labels=paste(total,'점',sep=""))

[문제155] 학생들의 이름을 기준으로 과목점수를 그룹형 막대그래프로 생성하세요.

bp <- barplot(t, names.arg=names(t), beside=TRUE, ylim=c(0,110),

xlab='이름', ylab='성적',

col=c('blue','green','purple'),

main='과목별 점수',

las=2)

legend('topright',

legend=rownames(t),

title='과목',

pch=15,

col=c('blue','green','purple'),

cex=0.8, pt.cex=0.6)

[문제156] 창업건수.csv 파일에 데이터 중에 년도별 치킨집 창업 건수를 막대그래프로 생성하세요.

ot<-tapply(open$치킨집, open$년도, max)

barplot(ot, names.arg=names(ot), ylim=c(0,1200), main='년도별치킨집창업건수',xlab = '년 도', ylab='창업건수')

[문제157] 년도별 치킨집 창업, 폐업 건수를 그룹형 막대그래프로 생성하세요.

ct<-tapply(close$치킨집,close$년도,max)

ct

x<-rbind(ot,ct)

barplot(x,main="년도별 치킨집 창업, 폐업", names.arg=ot, col=c("blue","red"), ylim=c (0,4000),beside=T,legend=c("창업","폐업"))

[문제158] 2014 년도 업종별 창업 비율을 원형 그래프로 생성하세요.

open[open$년도==2014,-1]

x<-open[open$년도==2014,-1]

x

y<-round(x/(sum(x)-2014)\*100)

y

label=paste(colnames(x),'\n',paste(y,'%',sep=""))

library(plotrix)

pie3D(unlist(y),labels=label,start=2,explode=0.1,labelcex=0.8,main='2014Hell\_Gate',col = rainbow(length(y)))

'\n' 이거 넣어주면 네이밍이 엔터쳐저서 두줄로 표현가능

선생님답

x\_2014M-create\_cnt[create\_cnt$년도=='2014',-1]

cnt\_labels<-round(x\_2014/sum(x)2014)\*100,1)

cnt\_labels2<-paste(colnames(cnt\_labels),t(cnt\_labels),'%')

pie(t(cnt\_labels),

col=rainbow(7),

labels=cnt\_labels2,

main="2014년 업종별 창업비율")

pie3D(t(cnt\_labels),

col=rainbow(7),

labels=cnt\_labels2,

main="2014년 업종별 창업비율",labelcex=0.7,explode=0.1)

[문제159] 년도를 입력하면 해당 년도의 원형 그래프 생성할 수 있는 함수를 생성하세요 .

show\_pie(2006)

show\_pie(2006)

show\_pie<-function(x){

graphics.off()

x2<-create\_cnt[create\_cnt$년도==x,(2:8)]

cnt\_labels<-round(x2/sum(x2)\*100,1)

cnt\_labels2<-paste(colnames(cnt\_labels),'\n',t(cnt\_labels),'%')

pie(t(cnt\_labels),col=rainbow(7),labels=cnt\_labels2)

}

'\n' 이거 넣어주면 네이밍이 엔터쳐저서 두줄로 표현가능

[문제160] 한 주간 영업 판매량을 막대그래프를 생성하데 막대그래프의 색상은 판매량 에 따라 다르게 해야 합니다.

판매량이 200 이상이면 red, 199 ~ 150 이면 yellow, 149 ~ 100 이면 blue, 99 ~ 50 pink, 49 ~ black색으로 출력하세요.

월요일 : 70

화요일 : 151

수요일 : 210

목요일 : 180

금요일 : 250

토요일 : 300

일요일 : 40

sales <- c(100,151,80,30,250,300,40)

colors <- NULL

for (i in 1:length(sales)){

if(sales[i]>=200){

colors <- c(colors,"red")

} else if(sales[i]>= 150 & sales[i]<=199){

colors <- c(colors,"yellow")

} else if(sales[i]>=100 & sales[i]<=149){

colors <- c(colors,"blue")

} else if(sales[i]>=50 & sales[i]<=99){

colors <- c(colors,"pink")

} else {

colors <- c(colors,"black")

}

}

colors

barplot(sales, main = "판매량",

names.arg=c("월","화","수","목","금","토","일"),

col=colors, xlab="요일", ylab="판매량", ylim=c(0,350))

abline(h=seq(0,350,10),lty=3)

[문제161] itwill hotel 투수객 조사를 통해 고객들의 평가에 대한 정보는 survey.csv에 저 장 되어 있습니다.

파일 내용을 통해서 호텔에 대한 고객들의 반응을 그래프로 생성하세요.

survey <- read.csv("c:/data/survey.csv",stringsAsFactors = F,head = F)

# 결과 집합 생성

result <- matrix(0, ncol = 5, nrow = 1)

colnames(result) <- c("매우좋음","좋음","보통","나쁨","매우나쁨")

# 결과집합 데이터입력

for(i in 1:length(survey)){

if(survey[i] == "매우좋음"){

result[,1] <- result[,1] +1

}else if(survey[i] == "좋음"){

result[,2] <- result[,2] + 1

}else if(survey[i] == "보통"){

result[,3] <- result[,3] + 1

}else if(survey[i] == "나쁨"){

result[,4] <- result[,4] + 1

}else{

result[,5] <- result[,5] + 1

}

}

survey <- read.csv("c:/data/survey.csv",stringsAsFactors = F,head = F)

# 결과 집합 생성

result <- matrix(0, ncol = 5, nrow = 1)

colnames(result) <- c("매우좋음","좋음","보통","나쁨","매우나쁨")

# 결과집합 데이터입력

for(i in 1:length(survey)){

if(survey[i] == "매우좋음"){

result[,1] <- result[,1] +1

}else if(survey[i] == "좋음"){

result[,2] <- result[,2] + 1

}else if(survey[i] == "보통"){

result[,3] <- result[,3] + 1

}else if(survey[i] == "나쁨"){

result[,4] <- result[,4] + 1

}else{

result[,5] <- result[,5] + 1

}

}

# 막대그래프

bp <-

barplot(height = as.numeric(result), names.arg = colnames(result),

ylim = c(0,10),

col = c("green","forestgreen","yellow","orange","red"),

main = "투숙객 설문조사표")

text(bp,result,labels = paste(result,"명",sep=""), pos = 3)

# 원형그래프

pie(as.numeric(result), labels = paste(result/sum(result)\*100,"%",sep=""),

col = c("green","forestgreen","yellow","orange","red"),

main = "투숙객 설문조사표")

legend(1,1,legend = colnames(result), pch = 15,

col = c("green","forestgreen","yellow","orange","red"),

title = "만족도")

선생님답

survey <- read.csv("c:/data/survey.csv", header=F, stringsAsFactors = F)

survey

s <- t(survey)

u <- unique(s)

u

u <- factor(u, levels=c('매우좋음','좋음','보통','나쁨','매우나쁨'))

**#이렇게 숫자아니면서 레벨있어야할땐 factor가 유횽하다.**

u

result <- data.frame(eval = sort(u), cn = 0) **#도수분포표**

result

s

for(i in s){

result[result$eval==i,'cn'] <- result[result$eval==i,'cn'] +1

}

result

proportion <- round((result$cn/sum(result$cn))\*100)

proportion

label <- paste(result$eval,'\n',proportion,"%",sep="")

pie(result$cn,

init.angle = 90,

labels=label,

main="itwill hote 품질 조사",

col = rainbow(length(result$eval)))

bp<-barplot(height = result$cn,

width = 0.5, # 폭

names.arg= result$eval,

horiz = FALSE,

col = rainbow(length(result$eval)),

main = "itwill hotel 품질 조사",

xlab = "등급",

ylab = "도수",

ylim = c(0,10))

text(x=bp, y=result$cn,labels=paste(result$cn,'명',sep=''), pos=3, col='red')

**분할표(contingency table)**

-명목형(categorical), 순서형(순위형)(ordinal)데이터의 도수(frequency)를 표 형태로 나타내는 것을 의미한다.

x<-c('A','B','A','B','B','C')

**#데이터의 빈도수 (도수)**

table(x)

**> x<-c('A','B','A','B','B','C')**

**> table(x)**

x

A B C

2 3 1

**#최대값이 저장된 위치의 색인을 반환**

**> which.max(table(x))**

B

2

**#최소값이 저장된 위치의 색인을 반환**

**> which.min(table(x))**

C

3

**> names(table(x)) #셀목록**

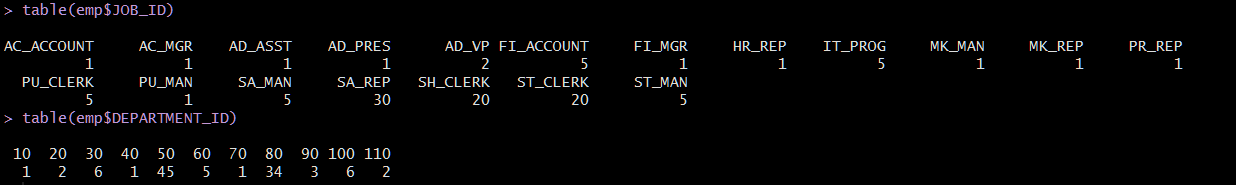
[1] "A" "B" "C"

**> names(table(x))[2]**

[1] "B"

**table(emp$JOB\_ID)**

**table(emp$DEPARTMENT\_ID)**



#포물러를 사용해 분할표 작성

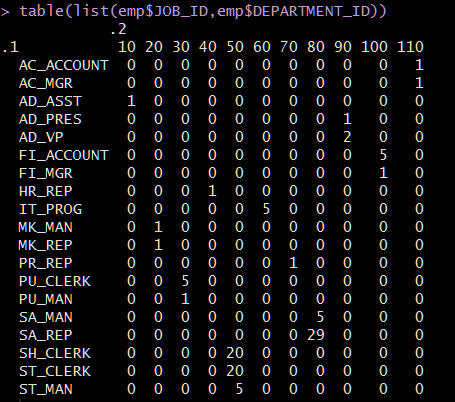
**xtabs(~JOB\_ID,emp)**

**xtabs(~DEPARTMENT\_ID,emp)**

**table(list(emp$JOB\_ID,emp$DEPARTMENT\_ID))**

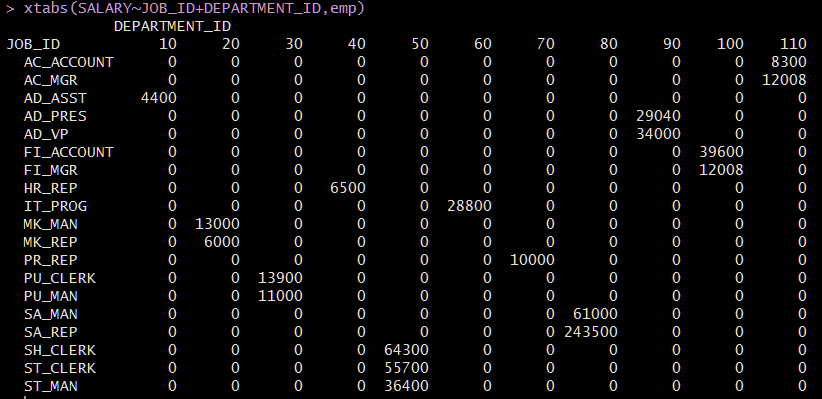
**xtabs(~JOB\_ID+DEPARTMENT\_ID,emp)**

크로스테이블이라고도한다.



**xtabs(SALARY~JOB\_ID+DEPARTMENT\_ID,emp)**

#JOB\_ID,DEPARTMENT\_ID크로스 되는 부분에SALARY에 대한 합이들어가진다.



SALARY부분이 옵션자리.

열은 부서아이디

행은 잡아이디

잡아이디를 기준으로해서 행의 합을 구하고싶음

rowsum, colsum, apply의 margin1,2 처럼

분할표에대한 행의합과 열의 합을 구할때쓰는 method

**x<-xtabs(SALARY~JOB\_ID+DEPARTMENT\_ID,emp)**

**> margin.table(x,1)**

JOB\_ID

AC\_ACCOUNT AC\_MGR AD\_ASST AD\_PRES AD\_VP FI\_ACCOUNT FI\_MGR HR\_REP

8300 12008 4400 29040 34000 39600 12008 6500

IT\_PROG MK\_MAN MK\_REP PR\_REP PU\_CLERK PU\_MAN SA\_MAN SA\_REP

28800 13000 6000 10000 13900 11000 61000 243500

SH\_CLERK ST\_CLERK ST\_MAN

64300 55700 36400

**> margin.table(x,1) #분할표의 행방향 합**

**> margin.table(x,2) #분할표의 열방향 합**

**> margin.table(x,2)**

DEPARTMENT\_ID

10 20 30 40 50 60 70 80 90 100 110

4400 19000 24900 6500 156400 28800 10000 304500 63040 51608 20308

#분할표 행의 비율 (한 행 전체가 1)

**> prop.table(x,1)**

DEPARTMENT\_ID

JOB\_ID 10 20 30 40 50 60 70 80 90 100 110

AC\_ACCOUNT 0 0 0 0 0 0 0 0 0 0 1

AC\_MGR 0 0 0 0 0 0 0 0 0 0 1

AD\_ASST 1 0 0 0 0 0 0 0 0 0 0

AD\_PRES 0 0 0 0 0 0 0 0 1 0 0

AD\_VP 0 0 0 0 0 0 0 0 1 0 0

FI\_ACCOUNT 0 0 0 0 0 0 0 0 0 1 0

FI\_MGR 0 0 0 0 0 0 0 0 0 1 0

HR\_REP 0 0 0 1 0 0 0 0 0 0 0

IT\_PROG 0 0 0 0 0 1 0 0 0 0 0

MK\_MAN 0 1 0 0 0 0 0 0 0 0 0

MK\_REP 0 1 0 0 0 0 0 0 0 0 0

PR\_REP 0 0 0 0 0 0 1 0 0 0 0

PU\_CLERK 0 0 1 0 0 0 0 0 0 0 0

PU\_MAN 0 0 1 0 0 0 0 0 0 0 0

SA\_MAN 0 0 0 0 0 0 0 1 0 0 0

SA\_REP 0 0 0 0 0 0 0 1 0 0 0

SH\_CLERK 0 0 0 0 1 0 0 0 0 0 0

ST\_CLERK 0 0 0 0 1 0 0 0 0 0 0

ST\_MAN 0 0 0 0 1 0 0 0 0 0 0

**#분할표 열의 비율값 (한 열 전체가 1)**

**> prop.table(x,2)**

DEPARTMENT\_ID

JOB\_ID 10 20 30 40 50 60 70

AC\_ACCOUNT 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

AC\_MGR 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

AD\_ASST 1.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

AD\_PRES 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

AD\_VP 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000

.

.

.

**#분할표 전체의 비율(마진삭제) 분할표전체를 1로봄**

**> prop.table(x)**

DEPARTMENT\_ID

JOB\_ID 10 20 30 40 50 60

AC\_ACCOUNT 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

AC\_MGR 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

AD\_ASST 0.006381843 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

AD\_PRES 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

AD\_VP 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

FI\_ACCOUNT 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

FI\_MGR 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000

HR\_REP 0.000000000 0.000000000 0.000000000 0.009427723 0.000000000 0.000000000

IT\_PROG 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000 0.041772064

MK\_MAN 0.000000000 0.018855445 0.000000000 0.000000000 0.000000000 0.000000000

[문제162] itwill hotel 투수객 조사를 통해 고객들의 평가에 대한 정보는 survey.csv에 저 장 되어 있습니다.

파일 내용을 통해서 호텔에 대한 고객들의 반응을 그래프로 생성하세요.(table함수를 이 용해서 해결해주세요)

survey

x<-t(survey)

table(x)

sum(table(x))

prp<-(prop.table(table(x))\*100)

label=paste(names(table(x)),paste(prp,'%',sep = ""))

label

pie(prp,

init.angle = 90,

labels=label,

main="itwill hotel 품질 조사",

col = rainbow(length(x)))

선생님 답

survey <- read.csv("c:/data/survey.csv", header=F, stringsAsFactors = F)

x<-table(t(survey))

x

library(reshape)

result <- melt(x)

names(result) <- c('eval','cn')

result

#proportion <- round((result$cn/sum(result$cn))\*100)

proportion<-prop.table(result$cn)\*100

proportion

label <- paste(result$eval,'\n',proportion,"%",sep="")

pie(result$cn,

init.angle = 90,

labels=label,

main="itwill hote 품질 조사",

col = rainbow(length(result$eval)))

result$eval <- factor(result$eval, levels=c('매우좋음','좋음','보통','나쁨','매우나쁨'))

result

result[order(result$eval),2]

#의미에맞게 순서를 맞추기위해서 factor쓰고, order순서 맞춰줬음

bp<-barplot(height = result[order(result$eval),2],

width = 0.5,

names.arg= result[order(result$eval),1],

horiz = FALSE,

col = rainbow(length(result$eval)),

main = "itwill hotel 품질 조사",

xlab = "등급",

ylab = "도수",

ylim = c(0,10))

text(x=bp, y=result[order(result$eval),2],labels=paste(result[order(result$eval),2],'명',sep=''), pos=3, col='red')

**산점도(scatter plot)**

-주어진 데이터를 점으로 표시해 흩뿌리듯이 시각화한 그래프

-데이터의 실제값들이 표시 되므로 데이터의 분포를 한눈에 살펴보는데 유용하다.

data() R에 내장되어있는 모든 데이터 셋을 확인 할 수 있다.

women # 70년대 여성 키 몸무게 데이터 셋

help(women)

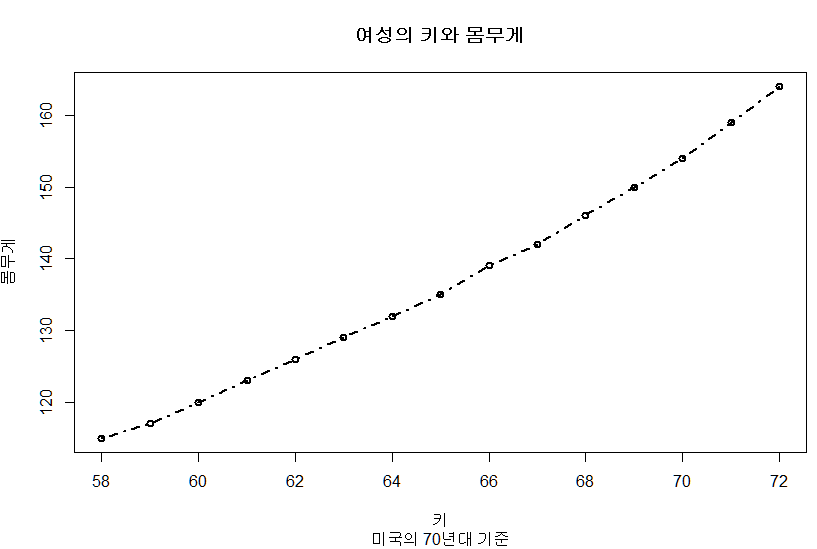
str(women)

**<x-y plotting>**

plot(women$weight)

plot(women$height)

**plot(x=women$height,y=women$weight,xlab='키',ylab='몸무게',main='여성의 키와 몸무게',sub='미국의 70년대 기준',type='o',lty=4, lwd=2,pch=1,cex=1)**

****

#sub 부제목

#type p: 점, l:선, b:점,선, c: b의 선, o: 점위의 선, h:수직선,s: 계단, n:나타나지않음

# lty 선의 유형 (1~6)

0:그리지 않음

1:실선(기본값)

2:대시(-)

3:점

4:점과 대시

5:긴 대시

6:두개의 대시

#lwd 선의 굵기 (기본값1)

#pch 점의 종류 0~25

#cex 점의 크기 (기본값 1)

상관분석 할 때 많이 쓰인다. 산포도, 두변수의 상관관계를 볼 수있다.

data.go.kr 공공데이터 얻는 곳

[문제163] 1군전염병발병현환\_년도별.csv 에 데이터를 가지고 그래프를 생성하세요.

d<-read.csv("c:/data/1군전염병발병현황\_년도별.csv",header = T,stringsAsFactors = F)

d

plot(d$콜레라,ylim=c(0,6000),axes=FALSE,col="violet",type="o", lwd=1,main="1군전염병발병현황-년도별(단위:건수) 출처:통계청")

axis(1,at=1:11, labels=d$년도별,las=2) #axis 축을만들어주고싶을때 써주는 것

axis(2)

#이미 plot이만들어져있기 때문에 그다음 그래프는 lines()를 사용한다.

lines(d$장티푸스,col="blue",type="o",lwd=1)

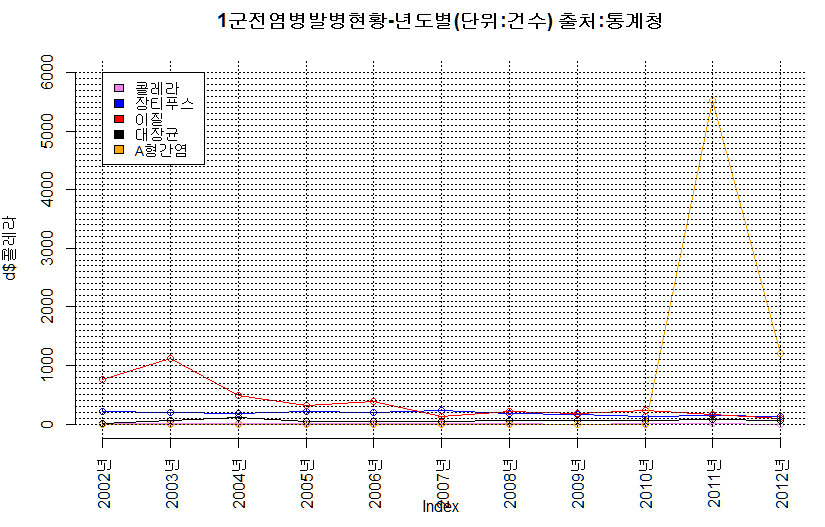
lines(d$이질,col="red",type="o",lwd=1)

lines(d$대장균,col="black",type="o",lwd=1)

lines(d$A형간염,col="orange",type="o",lwd=1)

abline(h=seq(0,6000,100),v=seq(1,11,1),lty=3,lwd=0.2)

legend(x=1,y=6000,legend=names(d[2:6]),cex=0.8,fill=c("violet","blue","red","black","orange"))



**score<-scan() # 입력해주는기능**

**Read Data Values**

**Read data into a vector or list from the console or file.**

score

90 88 78 65 80 94 69 72 83 64 95 68 87 69 82 91 63 70 81 67

> class(score)

[1] "numeric"

> mode(score)

[1] "numeric"

> str(score)

num [1:20] 90 88 78 65 80 94 69 72 83 64 ...

[문제164] 조사된 데이터 값들이 있습니다. 도수분포표를 만드세요.

90 88 78 65 80 94 69 72 83 64 95 68 87 69 82 91 63 70 81 67

계급 도수 상대도수 누적도수

1 90점이상 4 0.20 4

2 80점이상~90점미만 6 0.30 10

3 70점이상~80점미만 3 0.15 13

4 60점이상~70점미만 7 0.35 20

5 전체 20 1.00 NA

ft<- data.frame(계급= c("90점이상","80점이상~90점미만","70점이상~80점미만","60점이상~70점미만"),

도수=c(0,0,0,0))

for(i in score){

if(i>=90){

ft**[1,2]**<- ft[1,2]+1

}else if(i>=80 & i<90){

ft[2,2]<-ft[2,2]+1

}else if(i>=70 & i<80){

ft[3,2]<-ft[3,2]+1

}else{ft[4,2] <- ft[4,2]+1}

}

ft

ft <- **rbind**(ft,data.frame(계급="전체",도수=sum(ft[,2])))

ft

ft$상대도수 <- ft$도수/**ft[ft$계급=="전체","도수"]**

ft

for(i in (1:NROW(ft))){

if(i==1){

ft$누적도수[i] <- ft$도수[i]

} else if(i == NROW(ft)){

ft$누적도수[i]<- NA

} else {

ft$누적도수[i]<- ft$누적도수[i-1]+ft$도수[i]

}

}

ft

도수분포표

미리 구간을 설정해 놓고 각 구간의 범위안에 조사된 데이터 값들이 몇개씩 속하는가를 표시하는 표를 도수분포표(frequency distribution),frequency table

계급: 각 구간

도수: 각 구간(계급)에 속한 데이터 값들의 수

상대도수: 어느 계급에 속한 도수가 전체 도수에 차지하는 비율

누적도수: 어느 계급의 도수와 그 위의 계급에 속한 도수를 모두 합하여 구하며 이러한 누적도수를 각 계급 별로 표시

계급의 수 결정

계급의 수 = 자료의 측정값들의 갯수^(1/3) (절대적인것은 아니지만 공식)

round(length(score)^(1/3)) =3 계급의 수를 3개정도만들면 된다.

급간=(가장 큰 측정값 - 가장 작은 측정값)/계급 수

(max(score)-min(score))/round(length(score)^(1/3)) =10

도수분포표

**#연속형변수를 범주형 변수로 변환**

cut

score<-scan()

90 88 78 65 80 94 69 72 83 64 95 68 87 69 82 91 63 70 81 67

score

c<-cut(score,breaks=c(60,70,80,90,100),right=FALSE,labels=c("60점이상~70점미만","70점이상~80점미만","80점이상~90점미만","90점이상"))

c

table(c)

cut:: right= FALSE(60<=~<70)[60,70) #[: 크거나같다. ):작다

TRUE(60<~<=70) (60,70]

c<-cut(score,breaks=seq(60,100,by=10),right=FALSE,labels=c("60점이상~70점미만","70점이상~80점미만","80점이상~90점미만","90점이상"))

c

table(c)

도수분포표

c.table<-table(c)

상대도수분포표

c.prop<-prop.table(c.table)

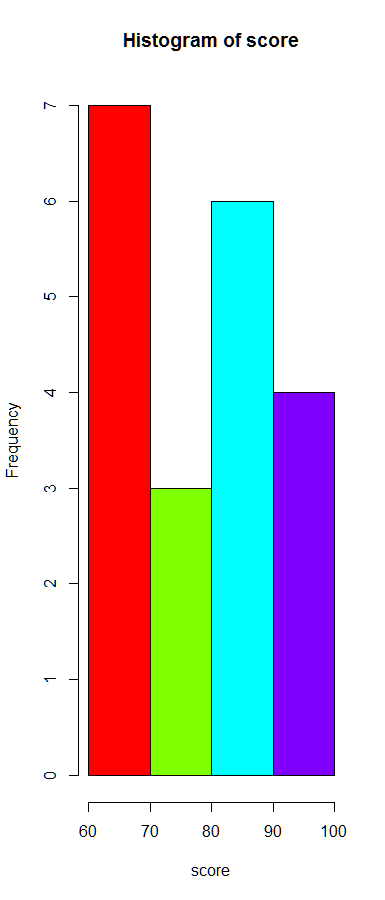
cbind(c.table,c.prop)

도수분포표는 자료의 중요한 특성을 서술적 방법으로 나타내는데 효과적이고 이를 시각화하려면 히스토그램을 사용하면 된다.

**값의 분포도** - 도수분포표

도수분포표의 시각화 - 히스토그램

hist(score,breaks=seq(60,100,by=10),right=FALSE, col=rainbow(4))

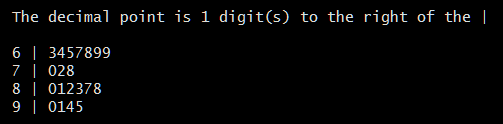


줄기잎그림(stem and leaf diagram)

서술적인면과 그래프의 시각적인 면을 동시에 고려하여 자료의 특성을 나타낼 때 사용

-자료를 구성하는 각각의 데이터 값들의 줄기(stem)와 잎(leaf)으로 구분

stem(score)



줄기 | 잎

63,64,65,67,68,69,69 이런식으로

숫자만입력된다.

소수점은 반올림된다.

[문제165] itwill 자동차 수리점 부품 비용 자료 입니다. 분석하세요.

91 78 93 57 75 52 99 80 97 62

71 69 72 89 66 75 79 75 72 76

104 74 62 68 97 105 77 65 80 109

85 97 88 68 83 68 71 69 67 74

62 82 98 101 79 105 79 69 62 73

read.table

[문제166] 학생들의 몸무게 자료를 이용해서 도수분포표,그래프를 생성하세요.

weight <- read.table("c:/data/weight.txt")

weight <- read.table("c:/data/weight.txt")

w<-unlist(weight)

w

class(w)

max(weight)#93

min(weight)#52

round((length(weight)\*nrow(weight))^1/3) #17

(93-52)/17 #2.4

length(seq(52,94,by=3))

seq(52,93,by=3)

cw<-cut(w, breaks=seq(50,100,by=10),right=FALSE,labels=c('50~59','60~69','70~79','80~89','90~'))

cw

cwt<-table(cw)

cwp<-prop.table(table(cw))

result<-cbind(cwt,cwp)

colnames(result)<-c("도수", "상대도수")

result

도수 상대도수

50~59 2 0.04

60~69 15 0.30

70~79 21 0.42

80~89 10 0.20

90~ 2 0.04

hist(x=w,breaks=seq(50,100,by=10),xlab='weight',ylab='cn',ylim =c(0,25),right=FALSE,main='weight of students',col=rainbow(5))

지도를 배워보자.

```{r}

install.packages("ggmap")

library(ggmap)

```

geocode("지역명 또는 주소") 위도 경도값을 반환하는 함수

```{r}

geocode(enc2utf8("서울시")) #enc2utf8:한글을 utf8형식으로 변환 함수.

```

longitude경도 latitude위도

```{r}

geocode(enc2utf8("서울시 강남구 역삼로7길 17"))

gc<-geocode(enc2utf8("서울시 강남구 역삼로7길 17"))

get\_googlemap(center=as.numeric(gc),zoom=18,maptype="satellite")

#numeric형식으로 넣어줘야한다.

map<-get\_googlemap(center=as.numeric(gc),zoom=18,maptype="satellite")

ggmap(map)

```

center : 지도 좌표값

zoom:지도 크기 기본값(10 도시),3(대륙) ~21(빌딩)

size: 지도 가로 세로 픽셀 크기 기본값640 x 640, 바꾸는법 size=c(640,640)

maptype: 지도 유형

terrain(지형정보기반),satellite(위성지도),roadmap(도로명표시),hybrid(위성,도로명)

marker: 위도 경도 위치에 마커 출력

ggmap:지도를 출력하는 함수

선생님답

weight <- read.table("c:/data/weight.txt")

str(weight)

weight <- as.matrix(weight)

dim(weight) <- c(50,1)

weight

max(weight)

min(weight)

w\_ct <- cut(weight,breaks=c(50,60,70,80,90,101),right=FALSE,include.lowest = FALSE,

labels=c("50이상~60미만","60이상~70미만","70이상~80미만","80이상~90미만","90이상"))

w\_ct

table(w\_ct)

prop.table(w\_t)

c\_df <- cbind(table(w\_ct),prop.table(w\_t))

c\_df

colnames(c\_df) <- c("도수","상대도수")

c\_df

hist(x=weight, breaks=seq(50,100,by=10), right=FALSE, main="학생들의 몸무게", xlab="몸무게", ylab="인원수", ylim=c(0,25),

col=rainbow(5),labels=T)

par(new=T)

plot(density(weight),type="l", col="darkblue",lwd=2,axes=FALSE,xlab="",ylab="",main="")

stem(weight)

names<-c("1.협재해수욕장","2.함덕해수욕장")

addr<-c("제주특별자치도 제주시ㅐ 한림읍 협재리 2497-1",

"제주특별자치도 제주시 조천읍 함덕리 1008")

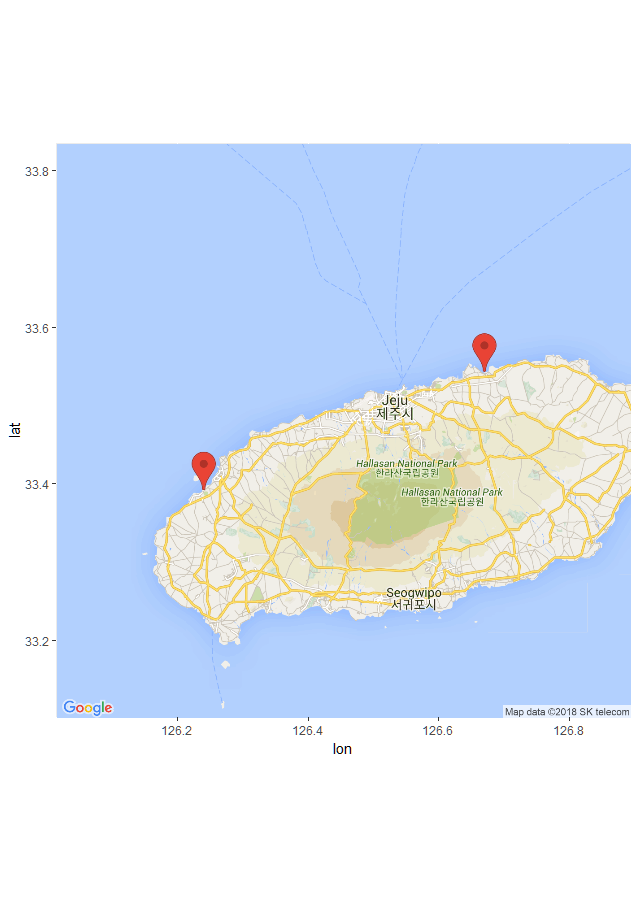
gc<-geocode(enc2utf8(addr))

df<-data.frame(name=names, lon=gc$lon, lat=gc$lat)

cen<-c(mean(df$lon),mean(df$lat)) #중심좌표값을 만들어줌 이렇게하면좋음

map<-get\_googlemap(center=cen, maptype="roadmap",zoom=10,marker=gc)

ggmap(map)



geocode(enc2utf8("서울시 강남구 역삼로7길 17"))

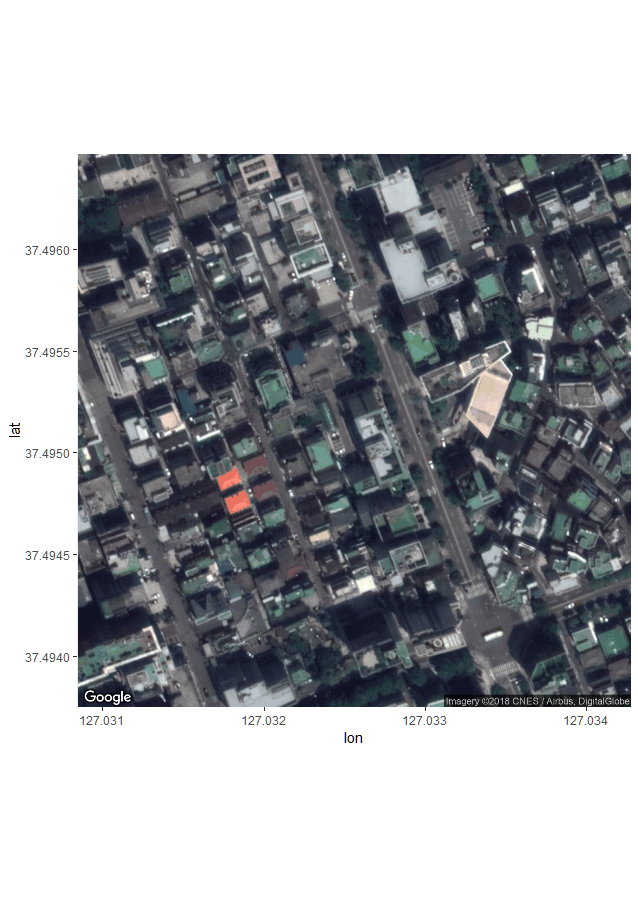
gc<-geocode(enc2utf8("서울시 강남구 역삼로7길 17"))

get\_googlemap(center=as.numeric(gc),zoom=18,maptype="satellite")

#numeric형식으로 넣어줘야한다.

map<-get\_googlemap(center=as.numeric(gc),zoom=18,maptype="satellite")

ggmap(map)



경도위도 알고찍을 경우에는 이렇게 찍어야한다.

cen<-c(2.294481,48.85837)

ggmap(get\_googlemap(center=cen,maptype="roadmap",zoom=15,marker=data.frame(lon=2.294481,lat=48.85837)))



#강남구 씨씨티비점찍기

install.packages("ggplot2")

library(ggplot2) # geom\_point

df <- read.csv("c:/data/서울특별시\_강남구\_CCTV\_20180510.csv", header=TRUE, stringsAsFactors=F)

str(df)

head(df)

df <- df[grep('테헤란로',df$소재지도로명주소),]

df <- data.frame(name=df$소재지도로명주소,lat=df$위도, lon=df$경도)

head(df)

tail(df)

cen <- c(mean(df$lon),mean(df$lat))

ggmap(get\_googlemap(center=cen,maptype = "roadmap",zoom=13,markers=data.frame(df$lon,df$lat)))

map <- get\_googlemap(center=cen,maptype = "roadmap",zoom=14)

cctv<-read.csv(choose.files(),header=T ) # csv불러올때 파일선택하기

[문제167] 2016년\_서울\_주요구별\_병원현황.csv file을 읽어 들인후 구별로 진료과목별 병원현황을 그룹막대형 그래프로 만드세요.

data1 <- read.csv("c:/data/2016년\_서울\_주요구별\_병원현황.csv" , header=T)

str(data1)

par(mfrow=c(1,1))

barplot(as.matrix(data1[1:9,2:11]),

main=paste("서울시 주요 구별 과목별 병원현황-2016년","\n",

"출처[국민건강보험공단]") , ylab="병원수", beside=T ,

col=rainbow(length(data1$표시과목)),ylim=c(0,350))

abline(h=seq(0,350,10),lty=3,lwd=0.2)

legend('topright',

legend=data1$표시과목,

cex=0.8,

fill=rainbow(length(data1$표시과목)),

bg="yellow")

[문제168] 강남구 과목별 병원수만 막대형 그래프를 생성하세요.(단 10개 1개로 표현하세요.)

barplot(data1$강남구\*0.1,

main="강남구 병원현황",

axes=F,

ylab="병원수(단위:10개)",

cex.names=0.8,

las=2,

ylim=c(0,40),

col=rainbow(length(data1$표시과목)),

border="white",

names.arg=data1$표시과목)

axis(2,ylim=seq(0,40,10))

abline(h=seq(0,40,5),lty=2)

[문제169] 2016년\_서울\_주요구별\_병원현황.csv file을 읽어 들인후 구별 진료과목에 해당하는 병원수를

막대형 그래프로 만드세요.

단 막대높이는 10개당 1개로 만드시고 막대그래프 프로그램은 한개를 가지고 모든 구별 막대그래프가 생성되도록 만드세요.

data1 <- read.csv("c:/data/2016년\_서울\_주요구별\_병원현황.csv" , header=T)

gu <- colnames(data1[2:11])

par(mfrow=c(2,5))

for(i in 1:10){

barplot(data1[1:9,i+1] \* 0.1 ,

main=paste(gu[i],"병원현황"),

axes=F,

ylab="병원수(단위:10개)",

xlab="",

cex.names=0.85,

las=2,

ylim=c(0,40),

col=rainbow(8),

border="white",

names.arg=data1$표시과목)

axis(2,ylim=seq(0,40,10))

abline(h=seq(0,40,5),lty=2)

}

-----------------------------------------------------------------------------------------------------

**write.csv()**

csv파일로 저장하기. 데이터정제작업할 때 요긴하다.

**data1 <- read.csv("c:/data/시군구별\_표시과목별\_의원현황\_20180813100616.csv" ,**

**header=T, skip=1,stringsAsFactor=F)**

skip 옵션:

**data1<-data1[,-1]** #첫번째 행 필요없음. 이렇게 뺄 수 있따.

행과열을 바꿀땐 t( )

**data1<-t(data1)**

**data1**

행렬로 구성됨, 행렬은 단일데이터 타입만 사용하기 때문에 전부문자형으로되어있음

이걸 os에 별도의 파일로 떨어뜨려 저장해보자.

다시읽으면 숫자형으로 돌아올테니.

**write.csv(data1,"c:/data/시군구별\_표시과목별\_의원현황\_20180813100616.csv")**

**write.csv(data1,"c:/data/시군구별\_표시과목별\_의원현황\_20180813100616.csv",quote=FALSE)**

quote= # 더블쿼텐션마크("") 유무 옵션

**data1 <- read.csv("c:/data/시군구별\_표시과목별\_의원현황\_20180813100616.csv" ,**

**header=T, skip=1, stringsAsFactor=F)**

skip으로 다시 첫번째 행 필요없는 첫번째 행 제거

**barplot(data1[,3],beside=T)**

**t( )로 행과 열을 바꾸어 줬을 때 자료의 타입문제발생 -> write로 다시 스킵할거하고 수정해주면 정리됨**

**-----------------------------------------------------------------------------------------------------**

**Animaiton**

**install.packages("animation")**

**library(animation)**

**myAni<-function(){**

**for (i in 10:0){**

**plot.new()**

**rect(0,0,1,1,col="gold")**

**text(0.5,0.5,i, cex=10,col=rgb(0,0,1,0.5))**

**ani.pause()**

**}**

**}**

**myAni()**

rgb(red,green,blue) # 빛의 3원색 0.5: 투명도 0.5는 안써도됨

**myAni<-function(){**

**n<-ani.options("nmax")**

**x<-sample(1:n)**

**y<-sample(1:n)**

**for(i in 1:n){**

**plot(x[i],y[i],cex=3, col="red", lwd=2,ylim=c(0,50),xlim=c(0,50))**

**ani.pause()**

**}**

**}**

**myAni()**

n<-ani.options("nmax") : 애니매이션의 옵션을 설정하는 함수,애니매이션에대한 프레임

interval : animation의 시간 간격(초단위) 기본값 1초

nmax : animation 프레임을 만들기 위한 반복수(1초의 30개 프레임)스틸컷, 기본값 50

ani.width() : 프레임 가로크기(픽셀)

ani.height() : 프레임 세로 크기(픽셀)

ani.pause() : 주어진 시간동에 대기하고 현재 화면을 지운다.

ani.puase('interval')기본값

-------------------------

[문제170] 모기업의 영업팀은'A팀', 'B팀','C팀','D팀', 'E팀'이 있습니다.

영업팀별 영업 매출액은 90,45,70,67,85 입니다.

이 자료를 이용해서 barplot, pie, pie3D그래프를 생성하세요.

y<-c(90,45,70,67,85)

x<-c('A팀', 'B팀','C팀','D팀', 'E팀')

#

barplot(y,names.arg=x,xlab='팀명',ylab='영업 매출액', col=rainbow(length(x)),ylim=c(0,100),main='영업매출액 BARPLOT',sub='2017년도')

#

p<-round(y/sum(y)\*100)

pie(y,labels=paste(x,p,'%'),main='영업매출액 pie차트',col=rainbow(length(y)))

**library(plotrix)**

pie3D(y,labels=paste(x,p,'%'),main='영업매출액 pie3D차트',col=rainbow(length(y)),explode=0.1,labelcex=1.5)

#

pie3D(proportion,labels=label, start=2, explode=0.1, labelcex = 0.8,main="성별 현황",col=c("blue","red"))

이걸로 애니메이션만들기

**install.packages("jpeg")**

**library(jpeg)**

**jpeg('c:/data/ex170\_1.jpg',width=400,height=300,pointsize = 12)**

y<-c(90,45,70,67,85)

x<-c('A팀', 'B팀','C팀','D팀', 'E팀')

barplot(y,names.arg=x,xlab='팀명',ylab='영업 매출액', col=rainbow(length(x)),ylim=c(0,100),main='영업매출액 BARPLOT',sub='2017년도')

**dev.off()**

#

**jpeg('c:/data/ex170\_2.jpg',width=400,height=300,pointsize = 12)**

p<-round(y/sum(y)\*100)

pie(y,labels=paste(x,p,'%'),main='영업매출액 pie차트',col=rainbow(length(y)))

**dev.off()**

#

library(plotrix)

**jpeg('c:/data/ex170\_3.jpg',width=400,height=300,pointsize = 12)**

pie3D(y,labels=paste(x,p,'%'),main='영업매출액 pie3D차트',col=rainbow(length(y)),explode=0.1,labelcex=1.5)

**dev.off()**

**for(i in 1:3){**

**img<- paste("c:/data/ex170\_",i,".jpg",sep="")**

**img<-readJPEG(img)**

**plot.new()**

**rect(0,0,1,1,col="white",border="white")**

**rasterImage(img,0,0,1,1)**

**ani.pause()**

**}**

rasterImage: 화면에 이미지 출력하는 함수

rasterImage(image,xleft,ybottom,xright,ytop)

image : 출력할 이미지 파일 이름

xleft : 이미지 출력 위치(x축 왼쪽)

ybottom : 이미지 출력 위치(y축 하단)

xright : 이미지 출력 위치(x축 오른쪽)

ytop : 이미지 출력 위치(y축 상단)

인터벌 옵션넣어줘도된다.

[문제171] 문제169번을 수행하면서 구이름.jpg형식으로 파일이 생성하도록 한후

구이름.jpg를 이용해서 애니메이션을 수행하세요.

[문제171] 문제169번을 수행하면서 구이름.jpg형식으로 파일이 생성하도록 한후

구이름.jpg를 이용해서 애니메이션을 수행하세요.

data1 <- read.csv("c:/data/2016년\_서울\_주요구별\_병원현황.csv" , header=T)

gu <- colnames(data1[2:11])

par=c(2,5)(mfrow)

jpeg('c:/data/goo1.jpg',width=400,height=300,pointsize=12)

for(i in 1:10){

barplot(data1[1:9,i+1] \* 0.1 ,

main=paste(gu[i],"병원현황"),

axes=F,

ylab="병원수(단위:10개)",

xlab="",

cex.names=0.85,

las=2,

ylim=c(0,40),

col=rainbow(8),

border="white",

names.arg=data1$표시과목)

axis(2,ylim=seq(0,40,10))

abline(h=seq(0,40,5),lty=2)

}

dev.off()

jpeg('c:/data/goo2.jpg',width=400,height=300,pointsize=12)

for(i in 1:2){

barplot(data1[1:9,i+1] \* 0.1 ,

main=paste(gu[i],"병원현황"),

axes=F,

ylab="병원수(단위:10개)",

xlab="",

cex.names=0.85,

las=2,

ylim=c(0,40),

col=rainbow(8),

border="white",

names.arg=data1$표시과목)

axis(2,ylim=seq(0,40,10))

abline(h=seq(0,40,5),lty=2)

}

dev.off()

for(i in 1:2){

img<- paste("c:/data/goo",i,".jpg",sep="")

img<-readJPEG(img)

plot.new()

rect(0,0,1,1,col="white",border="white")

rasterImage(img,0,0,1,1)

ani.pause()

}

선생님답

library(jpeg)

data1 <- read.csv("c:/data/2016년\_서울\_주요구별\_병원현황.csv" , header=T)

gu <- colnames(data1[2:11])

for(i in 1:10){

jpeg(paste('c:/data/',gu[i],'.jpg',sep=""),width=400,height=300,pointsize=12)

barplot(data1[1:9,i+1] \* 0.1 ,

main=paste(gu[i],"병원현황"),

axes=F,

ylab="병원수(단위:10개)",

xlab="",

cex.names=0.85,

las=2,

ylim=c(0,40),

col=rainbow(length(data1$표시과목)),

border="white",

names.arg=data1$표시과목)

axis(2,ylim=seq(0,40,10))

abline(h=seq(0,40,5),lty=2)

dev.off()

}

library(animation)

for (i in gu) {

img <- paste("c:/data/",i,".jpg",sep="")

img <- readJPEG(img)

plot.new()

rasterImage(img,0,0,1,1)

ani.pause()

}

무한반복은 어떻게하지?

jdk-8u102-windows-x64 설치

jar 파일 c:/data안에넣기

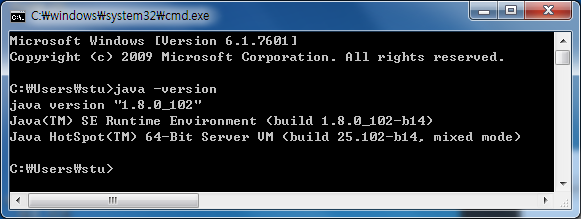
C:\Program Files\Java\jre1.8.0\_102\bin

path걸기

: 시스템설정(컴퓨터오른쪽버튼) -고급시스템설정- 환경변수-편집- path란 맨뒤에 ;C:\Program Files\Java\jre1.8.0\_102\bin 이어붙여넣기 -재부팅

cmd

java -version



path 완료

install.packages("RJDBC")

library(RJDBC)

jdbcDriver <- JDBC(driverClass="oracle.jdbc.OracleDriver", classPath**="C:/data/ojdbc6.jar"**)

conn <- dbConnect(jdbcDriver, "jdbc:oracle:thin**:@localhost:1521/xe**", "hr", "hr")

#[접속해야할 db아이피주소]:[포트번호]/[sid(db의 인스턴스이름)], "오라클이름", "계정"

emp <- dbGetQuery(conn, "select \* from employees")

dept <- dbGetQuery(conn, "select \* from departments")

> class(emp)

[1] "data.frame"

<https://www.java.com/ko/>

에서 java설치

jar 파일이 꼭 컴퓨터에저장되어있어야한다.

**이렇게하면 db에서 직접 data를 가져와서 가공할 수 있다.**

[문제172] R에서 오라클 접속한 후 부서별 인원수에 대한 막대그래프 그리세요.

GD<-sort(tapply(emp$EMPLOYEE\_ID, ifelse(is.na(emp$DEPARTMENT\_ID),'NA', emp $DEPARTMENT\_ID), NROW ),decreasing=TRUE)

barplot(GD,col=rainbow(length(GD)),ylim=c(0,50),las=2,)

선생님답

library(RJDBC)

jdbcDriver <- JDBC(driverClass="oracle.jdbc.OracleDriver", classPath="C:/r/ojdbc6.jar")

conn <- dbConnect(jdbcDriver, "jdbc:oracle:thin:@localhost:1521/xe", "hr", "hr")

df <- dbGetQuery(conn, "SELECT nvl(department\_name,'소속부서(x)') dept\_name, count(\*) cnt

FROM employees e , departments d

WHERE e.department\_id = d.department\_id(+)

GROUP BY nvl(department\_name,'소속부서(x)')")

bp <- barplot(df$CNT,

names.arg=df$DEPT\_NAME,

main='부서별 사원 수',

col=terrain.colors(length(df$DEPT\_NAME)),

ylab='사원수(명)',

ylim=c(0,50),

las=2)

text(x=bp, y=df$CNT, labels=df$CNT, pos=3)

df <- dbGetQuery(conn, "SELECT nvl(department\_name,'소속부서(x)') dept\_name, count(\*) cnt

FROM employees e LEFT OUTER JOIN departments d

ON e.department\_id=d.department\_id

GROUP BY nvl(department\_name,'소속부서(x)')

ORDER BY cnt DESC")

bp <- barplot(df$CNT,

names.arg=df$DEPT\_NAME,

main='부서별 사원 수',

col=terrain.colors(length(df$DEPT\_NAME)),

ylab='사원수(명)',

ylim=c(0,50),

las=2)

text(x=bp, y=df$CNT, labels=df$CNT, pos=3)

emp <- dbGetQuery(conn, "select \* from employees")

**dept <- dbReadTable(conn, "DEPARTMENTS")**

#dbReadTable 테이블전체가져오기

library(sqldf)

res\_sqldf <- sqldf('SELECT department\_name dept\_name, count(\*) cnt

FROM emp e LEFT OUTER JOIN dept d ON e.department\_id=d.department\_id

GROUP BY department\_name')

res\_sqldf$dept\_name[is.na(res\_sqldf$dept\_name)] <- '소속부서(x)'

res\_sqldf$dept\_name

bp <- barplot(res\_sqldf$cnt,

names.arg=res\_sqldf$dept\_name,

main='부서별 사원 수',

col=terrain.colors(length(res\_sqldf$dept\_name)),

ylab='사원수(명)',

ylim=c(0,50),

las=2)

text(x=bp, y=res\_sqldf$cnt, labels=res\_sqldf$cnt, pos=3)

[문제173] 아래와 같은 결과를 출력하도록 SQL문을 작성하세요.

Job Dept 10 Dept 20 Dept 30 Dept 40 Dept 50 Dept 60 Dept 70 Dept 80 Dept 90 Dept 100

-------------------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ---------- ----------

AC\_MGR

AC\_ACCOUNT

IT\_PROG 28800

ST\_MAN 36400

AD\_ASST 0

PU\_MAN 11000

SH\_CLERK 64300

AD\_VP 35700

FI\_ACCOUNT 39600

MK\_MAN 13000

PR\_REP 10000

FI\_MGR 12008

PU\_CLERK 13900

SA\_MAN 61000

MK\_REP 6000

AD\_PRES 26400

SA\_REP 243500

HR\_REP 6500

ST\_CLERK 55700

SELECT job\_id "Job",

SUM(DECODE(department\_id , 10, salary)) "Dept10",

SUM(DECODE(department\_id , 20, salary)) "Dept20",

SUM(DECODE(department\_id , 30, salary)) "Dept30",

SUM(DECODE(department\_id , 40, salary)) "Dept40",

SUM(DECODE(department\_id , 50, salary)) "Dept50",

SUM(DECODE(department\_id , 60, salary)) "Dept60",

SUM(DECODE(department\_id , 70, salary)) "Dept70",

SUM(DECODE(department\_id , 80, salary)) "Dept80",

SUM(DECODE(department\_id , 90, salary)) "Dept90",

SUM(DECODE(department\_id , 100, salary)) "Dept100"

FROM employees

GROUP BY job\_id;

SELECT \*

FROM (SELECT department\_id, job\_id, salary

FROM employees)

**PIVOT (SUM(salary) FOR department\_id IN (10,20,30,40,50,60,70,80,90,100,110));**

SELECT \*

FROM (SELECT nvl(department\_id,0) dept\_id, job\_id, salary

FROM employees)

**PIVOT (SUM(salary) FOR dept\_id IN (10,20,30,40,50,60,70,80,90,100,110,0));**

**#PIVOT : 이 결과셋을 가지고 크로스테이블을만든다. 뒤에 열을 나열**

[문제174] 문제 173번에서 만든 SQL문을 R에서 오라클로 접속하셔서 수행하세요.

emp\_pivot <- dbGetQuery(conn,'SELECT \*

FROM (SELECT nvl(department\_id,0) dept\_id, job\_id, salary

FROM employees)

PIVOT (SUM(salary) FOR dept\_id IN (10,20,30,40,50,60,70,80,90,100,110,0))')

[문제175] R에서 오라클에 있는 EMPLOYEES 테이블을 읽어 들인 후 R함수를 이용하셔서

문제 173번과 동일한 결과를 출력하세요.

empdf <- dbReadTable(conn, "EMPLOYEES")

tapply(empdf$SALARY, list(empdf$JOB\_ID, empdf$DEPARTMENT\_ID), sum, **default=0)**

**#default :NA기본값을 0으로**

**SELECT \***

**FROM (SELECT department\_id, job\_id**

**FROM EMPLOYEES)**

**PIVOT(COUNT(\*) FOR DEPARTMENT\_ID IN(10,20,30,40,50,60,70,80,90,100,110));**

이렇게도된다.

**-------------**

**emp\_itwill<-dbGetQuery(conn,'**

**SELECT job\_id "Job",**

**SUM(DECODE(department\_id , 10, salary)) "Dept10",**

**SUM(DECODE(department\_id , 20, salary)) "Dept20",**

**SUM(DECODE(department\_id , 30, salary)) "Dept30",**

**SUM(DECODE(department\_id , 40, salary)) "Dept40",**

**SUM(DECODE(department\_id , 50, salary)) "Dept50",**

**SUM(DECODE(department\_id , 60, salary)) "Dept60",**

**SUM(DECODE(department\_id , 70, salary)) "Dept70",**

**SUM(DECODE(department\_id , 80, salary)) "Dept80",**

**SUM(DECODE(department\_id , 90, salary)) "Dept90",**

**SUM(DECODE(department\_id , 100, salary)) "Dept100"**

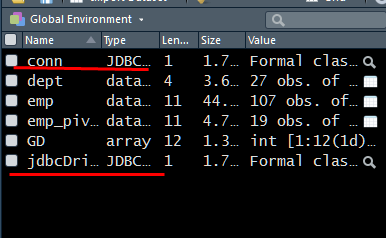
**FROM employees**

**GROUP BY job\_id')**

**emp\_itwill<-replace(emp\_itwill,is.na(emp\_pivot),0)**

**str(emp\_itwill)**

db에 create해보자.



**dbWriteTable(conn,"emp\_itwill",emp\_itwill) #R에서 DB로 테이블로생성**

**dbReadTable(conn,"EMP\_ITWILL") #R에서 DB 테이블읽어오기**

**DBSENDuPDATE(CONN,"DROP TABLE EMP\_ITWILL") #R에서 DB에 업데이트하기**

---------------------------------

library(ggplot2)

df<-read.csv("c:/data/exam.csv",header = T,stringsAsFactors = F)

str(df)

x<-df[df$subject=='SQL',]

x

> x

name subject grade

1 Carrick SQL 90

2 Pogba SQL 70

3 Mata SQL 92

4 Smalling SQL 76

5 Lukaku SQL 97

6 Herrera SQL 80

7 Young SQL 35

8 Lingard SQL 50

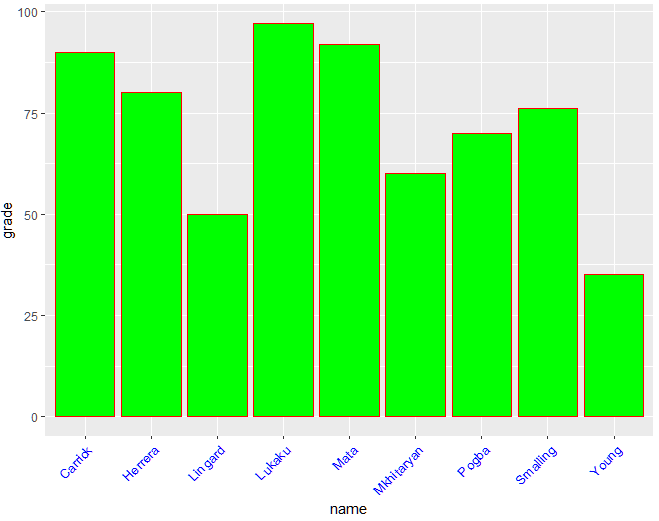
9 Mkhitaryan SQL 60

ggplot(x,aes(x=name,y=grade)) #그래프 좌표만나옴

ggplot(x,aes(x=name,y=grade))+

geom\_bar(stat="identity",fill="green",colour="red")+

theme(axis.text.x=element\_text(angle=45,hjust=1,vjust=1,colour="blue",size=10))

****

df

**library(plyr)**

**df1<-arrange(df,name,subject)**

**library(doBy)**

**df2<-orderBy(~name+subject,df)**

스택형 막대그래프, 그룹형막대그래프를 표현하고싶다면 이런모양으로 데이터 셋이 되어있어야하니까. 이렇게 쓴다.

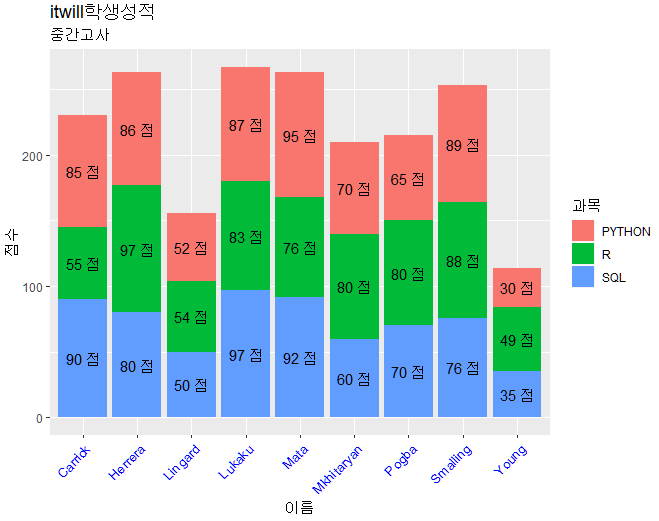
**ggplot(df2, aes(x=name, y=grade, fill=subject))+geom\_bar(stat='identity')+**

**geom\_text(aes(y=grade, label=paste(grade, '점')), col='black', size=4, position=position\_stack(vjust=0.5))+**

**theme(axis.text.x=element\_text(angle=45, hjust=1, vjust=1, colour='blue', size=10))+**

**ggtitle("itwill학생성적",subtitle="중간고사")+**

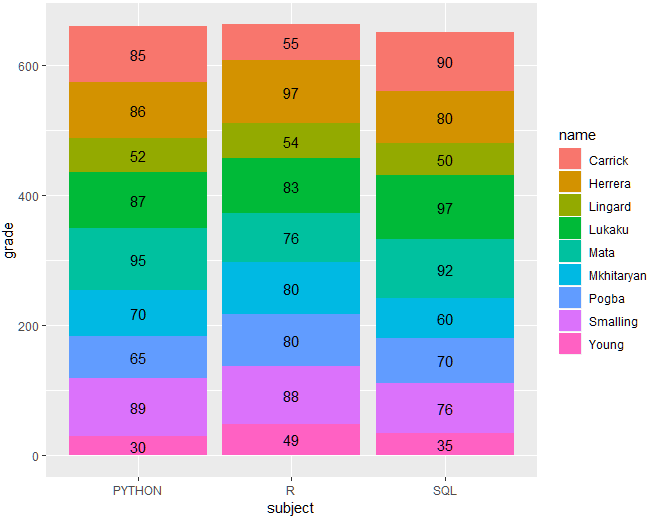
**labs(x="이름",y="점수",fill="과목")**

****

**ggplot(data=df,aes(subject, grade, group=name))+**

**geom\_col(aes(fill=name),position="stack")+**

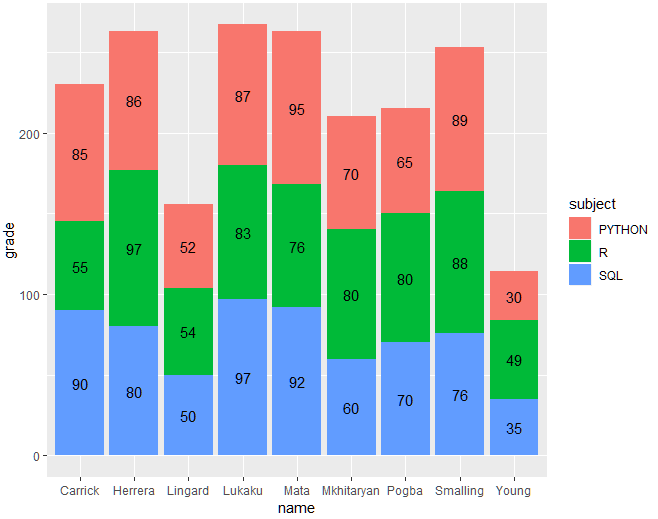
**geom\_text(aes(label=grade),position=position\_stack(vjust=0.5))**

****

**ggplot(data=df,aes(name, grade, group=subject))+**

**geom\_col(aes(fill=subject),position="stack")+**

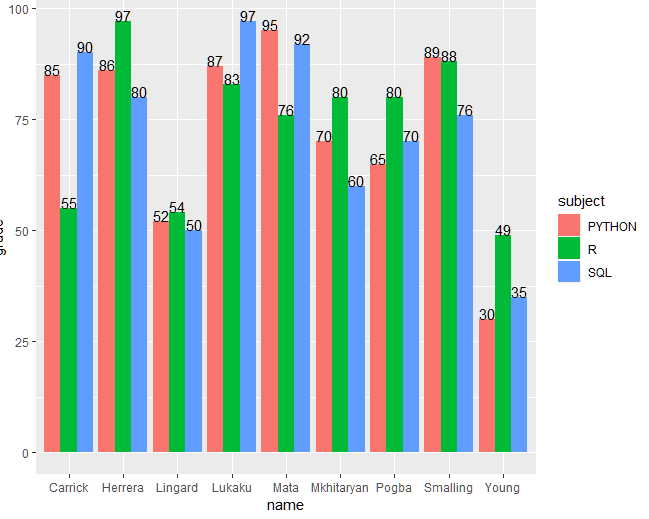
**geom\_text(aes(label=grade),position=position\_stack(vjust=0.5))**

****

**ggplot(data=df,aes(name, grade, group=subject))+**

**geom\_col(aes(fill=subject),position="dodge")+**

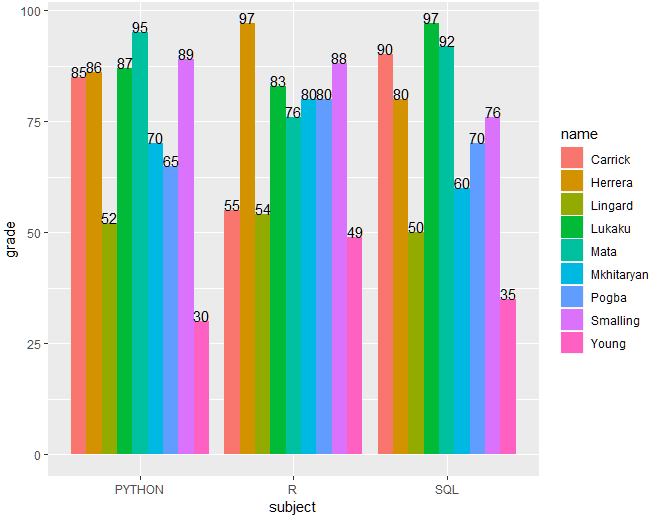
**geom\_text(aes(label=grade),position=position\_dodge(0.9),vjust=0)**

****

**ggplot(data=df,aes(subject, grade, group=name))+**

**geom\_col(aes(fill=name),position="dodge")+**

**geom\_text(aes(label=grade),position=position\_dodge(0.9),vjust=0,size=4)**

****

[문제176] 부서 인원수를 ggplot를 이용해서 막대그래프를 출력하세요.

data<-sqldf("select DEPARTMENT\_ID, count(EMPLOYEE\_ID)

from emp

group by DEPARTMENT\_ID")

ifelse(is.na(data$DEPARTMENT\_ID),'부서없긔',data$DEPARTMENT\_ID) #이걸 어떻게넣어주지?

ggplot(data,aes(x=data$DEPARTMENT\_ID,y=data$`count(EMPLOYEE\_ID)`, group=data$DEPARTMENT\_ID))+

geom\_bar(stat='identity')+

labs(x="부서번호",y="명")

emp <- dbGetQuery(conn, "select nvl(department\_id,0) dept\_id, count(\*) cn from employees group by nvl(department\_id,0) order by cn desc")

emp

ggplot(emp, aes(x=factor(DEPT\_ID),y=CN))+

geom\_bar(stat="identity", fill= rainbow(12))+

geom\_text(aes(label=CN),vjust=-0.1,size=5)+

labs(title='부서별 인원 현황', x='부서번호', y='인원수(명)')+

theme(plot.title=element\_text(face='bold', color='darkblue', hjust=0.5))+

theme(axis.title.x=element\_text(face='bold.italic', color='brown',size=6))+

theme(axis.title.y=element\_text(face='bold.italic', color='brown'))

[문제177] 년도별 입사한 수를 ggplot를 이용해서 막대그래프를 출력하세 요.

year<-substr(emp$HIRE\_DATE,1,4)

table(year)

cn<-table(year)

str(cn)

names(cn)

cn2<-as.data.frame(cn)

names(cn2)<-c('year','peoples')

cn2

ggplot(cn2,aes(x=year,y=peoples))+

geom\_bar(stat="identity",fill="black",colour="white")+

theme(axis.text.x=element\_text (angle=0,hjust=1,vjust=1,colour="black",size=10))

선생님답

**library(JDBC)**

**library(ggplot2)**

**library(lubridate) #year::lubridate**

**library(plyr) #count::plyr**

emp <- dbGetQuery(conn, "select \* from employees")

ggplot(emp,aes(x=**year(HIRE\_DATE)**))+

geom\_bar()

**#year() lubridate패키지의 날짜뽑기**

ggplot(emp, aes(x=factor(year(HIRE\_DATE)), fill=factor(year(HIRE\_DATE))))+

geom\_bar()+

labs(title = "년도별 입사현황", x="년도", y="인원수", fill="년도")+

theme(plot.title=element\_text(face='bold', color='darkblue', hjust=0.5))+

theme(axis.title.x=element\_text(face='bold', color='darkblue'))+

theme(axis.title.y=element\_text(face='bold', color='darkblue'))+

**theme(legend.title.align=0.5, legend.box.background = element\_rect(),**

**legend.box.margin = margin(t=0.1,r=0.1, b=0.1, l=0.1,unit='cm'))**

**# 범례 옵션들**

year\_cn <- **count(emp,"year(HIRE\_DATE)")**

colnames(year\_cn) <- c("year","cn")

year\_cn

**#count( ) plyr 패키지**

ggplot(year\_cn,aes(x=year,y=cn,fill=factor(year)))+

geom\_bar(stat="identity")+

geom\_text(aes(label=cn))+

ggtitle("년도별 입사현황",subtitle="신입사원")+

theme(plot.**title** = element\_text(lineheight=.8, face="bold",color="darkblue",hjust=0.5))+

theme(plot.**subtitle** = element\_text(lineheight=.8, face="bold",color="darkblue",hjust=0.5))+

labs(caption = "2001년 ~ 2008년")+

labs(x="년도",y="인원수")+

guides(fill=guide\_legend(title="년도"))+

theme(legend.title.align=0.5, legend.box.background = element\_rect(),

legend.box.margin = margin(t=0.1,r=0.1, b=0.1, l=0.1,unit='cm'))

hire\_cn <- dbGetQuery(conn, "select to\_char(hire\_date,'yyyy') year,count(\*) cn from employees group by to\_char(hire\_date,'yyyy')")

hire\_cn

ggplot(hire\_cn,aes(x=YEAR,y=CN,fill=YEAR))+

geom\_bar(stat="identity")+

geom\_text(aes(label=CN))+

labs(title = "년도별 입사현황", x="년도", y="인원수", fill="년도")+

theme(plot.title=element\_text(face='bold', color='darkblue', hjust=0.5))+

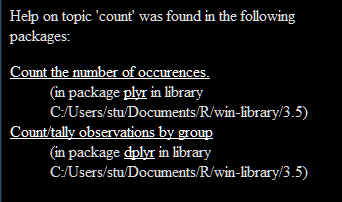
theme(axis.title.x=element\_text(face='bold', color='darkblue'))+

theme(axis.title.y=element\_text(face='bold', color='darkblue'))+

theme(legend.title.align=0.5, legend.box.background = element\_rect(),

legend.box.margin = margin(t=0.1,r=0.1, b=0.1, l=0.1,unit='cm'))

**?count**

****

어떤패키지안에있는지모를 때 이런식으로 찾아보면 된다.

[문제178] 학생들의 과목 점수를 각각으로 막대그래프를 그리세요.

df <- read.csv("c:/r/exam.csv", header=TRUE, stringsAsFactors=F)

df

ggplot(df, aes(x=name, y=grade))+

geom\_bar(stat="identity",fill="red")+

theme(axis.text.x = element\_blank())+

**facet\_wrap(~ name+subject)**

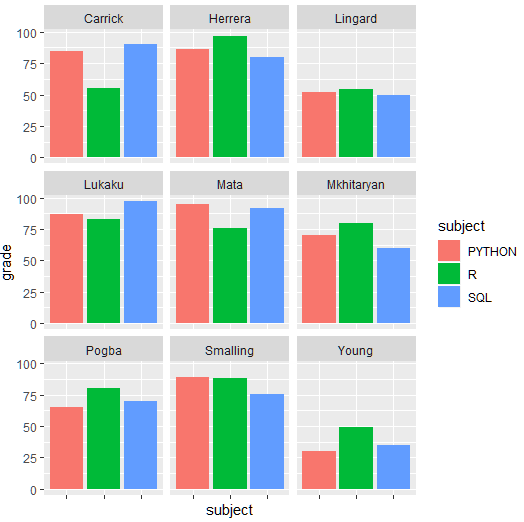
ggplot(df, aes(x=subject, y=grade, group=name))+

geom\_col(aes(fill=subject))+

theme(axis.text.x = element\_blank())+

**facet\_wrap(~ name)**

#이름을 기준으로 해서 그룹핑을하는 메소드

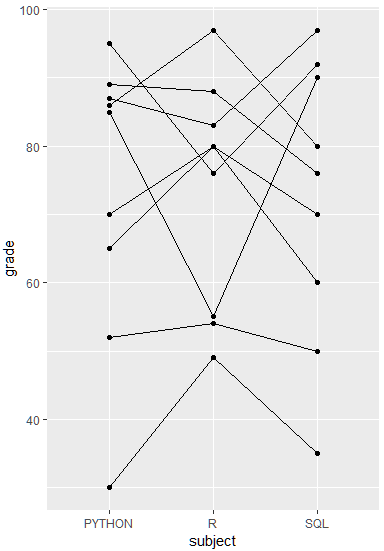


**ggplot 선과 점으로 그래프찍기**

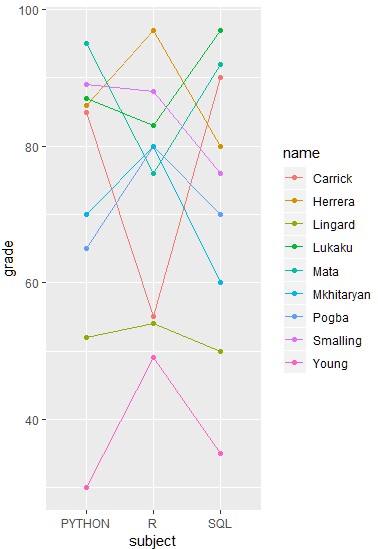
**df<-read.csv("c:/data/exam.csv",header=TRUE,stringsAsFactors = F)**

**library(ggplot2)**

**ggplot(df,aes(x=subject,y=grade, group=name))+geom\_line()+geom\_point()**

****

**ggplot(df,aes(x=subject, y=grade, color=name,group=name))+geom\_line()+geom\_point()**



#geom\_line(), geon\_point()로 전보다 간단하게 만들 수 있따.

[문제179] 'Mata','Young' 학생의 과목점수를 선과점 그래프로 생성하세요.

library(sqldf)

df2<-sqldf('select \*

from df

where name = "Mata"

or name = "Young"

order by name')

ggplot(df,aes(x=subject, y=grade, color=name,group=name))+geom\_line()+geom\_point()

선생님 답

ggplot(df[df$name %in% c('Mata','Young')],aes(x=subject, y=grade, color=name,group=name))+geom\_line()+geom\_point()

word crowd?

text1<-"R은 오픈소스로 통계, 기계학습, 금융, 생물정보학, 그래픽스에 이르는 다양한 통계패키지를 갖추고있는 좋은 프로그램이다."

#이 문장에서 단어만 추출해주세요.

library(stringr)

strsplit(text1," " )

#조사부분까지 다가지고있다. 명사만 뽑아내고싶다. 워드클라우드

#한글 워드클라우드 자바베이스로 되어있다. 1차적으로 전시간에했던 jar 파일과 이 패키지가 있어야 한다.

**install.lpackages("rJava")**

**library(rJava)**

**install.packages("KoNLP")**

**library(KoNLP)**

#KoNLP에서 제공하는 명사만추출하는기능

**extractNoun(text1) #한글 명사를 추출하는 함수: 한나눔분석기**

**> extractNoun(text1)**

[1] "R" "오픈" "소스" "통계" "기계" "학습" "금융" "생물"

[9] "정보학" "그래픽스" "다양" "한" "통계" "패키지" "프로그램"

> strsplit(text1," " )

[[1]]

[1] "R은" "오픈소스로" "통계," "기계학습," "금융,"

[6] "생물정보학," "그래픽스에" "이르는" "다양한" "통계패키지를"

[11] "갖추고있는" "좋은" "프로그램이다."

#이것과 비교해서 좋네..

오픈 소스/ 기계 학습/ 생물 정보학/다양 한/

KoNLP에서 갖고있는 사전에는 오픈소스, 기계학습 등의 이런 말들이 없어서 이렇게나오는 것임

품사까지 적용시켜 분석결과를 요청 해보자.

**> SimplePos09(text1) #9개의 품사 태그를 달아주는**

$`R은`

[1] "R/F+은/J"

$오픈소스로

[1] "오픈소스/N+로/J"

$`통계,`

[1] "통계/N+,/S"

$`기계학습,`

[1] "기계학습/N+,/S"

$`금융,`

[1] "금융/N+,/S"

$`생물정보학,`

[1] "생물정보학/N+,/S"

$그래픽스에

[1] "그래픽스/N+에/J"

$이르는

[1] "이르/P+는/E"

$다양한

[1] "다양한/N"

$통계패키지를

[1] "통계패키지/N+를/J"

$갖추고있는

[1] "갖추/P+고/E+있/P+는/E"

$좋은

[1] "좋/P+은/E"

$프로그램이다

[1] "프로그램/N+이/J+다/E"

$.

[1] "./S"

**> SimplePos22(text1) #22개 품사 태그를 달아주는 함수**

$`R은`

[1] "R/F+은/JX"

$오픈소스로

[1] "오픈소스/NC+로/JC"

$`통계,`

[1] "통계/NC+,/SP"

$`기계학습,`

[1] "기계학습/NC+,/SP"

$`금융,`

[1] "금융/NC+,/SP"

$`생물정보학,`

[1] "생물정보학/NC+,/SP"

$그래픽스에

[1] "그래픽스/NC+에/JC"

$이르는

[1] "이르/PA+는/ET"

$다양한

[1] "다양한/NC"

$통계패키지를

[1] "통계패키지/NC+를/JC"

$갖추고있는

[1] "갖추/PV+고/EC+있/PX+는/ET"

$좋은

[1] "좋/PA+은/ET"

$프로그램이다

[1] "프로그램/NC+이/JP+다/EF"

$.

[1] "./SF"

**text2**

**grep("/N",text2)**

**text2[grep("/N",text2)]**

**> library(stringr)**

**> str\_match(text2,'([A-z가-힣]+)/N')**

[,1] [,2]

[1,] NA NA

[2,] "오픈소스/N" "오픈소스"

[3,] "통계/N" "통계"

[4,] "기계학습/N" "기계학습"

[5,] "금융/N" "금융"

[6,] "생물정보학/N" "생물정보학"

[7,] "그래픽스/N" "그래픽스"

[8,] NA NA

[9,] "다양한/N" "다양한"

[10,] "통계패키지/N" "통계패키지"

[11,] NA NA

[12,] NA NA

[13,] "프로그램/N" "프로그램"

[14,] NA NA

품사들은 빼고 N만 나온다.

**text\_noun<-str\_match(text2,'([A-z가-힣]+)/N')**

**text\_noun[,2]**

**na.omit(text\_noun[,2])**

**as.vector(na.omit(text\_noun[,2]))**

**> text\_noun[,2]**

[1] NA "오픈소스" "통계" "기계학습" "금융" "생물정보학" "그래픽스"

[8] NA "다양한" "통계패키지" NA NA "프로그램" NA

**> na.omit(text\_noun[,2])**

[1] "오픈소스" "통계" "기계학습" "금융" "생물정보학" "그래픽스" "다양한"

[8] "통계패키지" "프로그램"

attr(,"na.action")

[1] 1 8 11 12 14

attr(,"class")

[1] "omit"

**> as.vector(na.omit(text\_noun[,2]))**

[1] "오픈소스" "통계" "기계학습" "금융" "생물정보학" "그래픽스" "다양한"

[8] "통계패키지" "프로그램"

**> extractNoun(text1)**

[1] "R" "오픈" "소스" "통계" "기계" "학습" "금융" "생물"

[9] "정보학" "그래픽스" "다양" "한" "통계" "패키지" "프로그램"

**> .libPaths() #패키지가 어느 경로에 있는 지 알 수 있다.**

[1] "C:/Users/stu/Documents/R/win-library/3.5" "C:/Program Files/R/R-3.5.1/library"

C:\Users\stu\Documents\R\win-library\3.5\KoNLP\_dic\current\dic\_user에 사전 있음

**useSejongDic()**

이거하면 사전추가업뎃됨

> extractNoun(text1)

[1] "R" "오픈" "소스" "통계" "기계" "학습" "금융" "생물"

[9] "정보학" "그래픽스" "다양" "한" "통계" "패키지" "프로그램"

**> buildDictionary(ext\_dic="sejong",user\_dic=data.frame(c("오픈소스","기계학습","생물정보학","다양한"),c("ncn")),replace\_usr\_dic=TRUE)**

370961 words dictionary was built.

#수동으로 단어 등록시켜주는 방법 이렇게하면 파일안에 추가가된다.

질적자료를 분석하려면 자료형을 정리해야한다.

table() #빈도수

**readLines("c:/data/취임사.txt")**

이렇게 하면 문장별로 나누어진다.

[124] "이 길에 함께해주십시오."

[125] "저의 신명을 바쳐 일하겠습니다."

[126] "감사합니다."

**text<-readLines("c:/data/취임사.txt")**

**text**

**table(unlist(extractNoun(rl)))**

단어별 빈도수

**install.packages("wordcloud")**

**library(wordcloud)**

**pal<-brewer.pal(8,"Dark2")**

**wordcloud(names(word\_cn),freq=word\_cn,scale = c(5,0.5), min.freq=2,random.order=F,rot.per=.1,colors=pal)**

#scale = c(5,0.5) 크게는 5부터 작게는 0.5

#min.freq=2 최소 2번이상나오는애들만추출

#random.order=F 랜덤하게추출여부

#rot.per=.1

nmaes:출력할 단어들

freq:빈도수

scale:글자의 크기c(큰값, 작은값)

min.freq : 최소 빈도수를 지정

max.words :이 값 이상의 빈도수면 삭제

random.order : 출력되는 순서를 임의로 지정

rot.per:단어배치

colors:출력될 단어들의 색상



**install.packages("wordcloud2")**

**library(wordcloud2)**

**wordcloud2(word\_cn, shape="pentagon")**



kaifa<-readLines("c:/data/kaifa.txt")

kaifa

extractNoun(kaifa)

table(extractNoun(kaifa))

cn<-table(extractNoun(kaifa))

wordcloud2(cn)



[문제180] seoul.txt 파일을 단어별 빈도수를 확인 하시고 wordcloud를 그리세요.

선생님답

library(KoNLP)

library(wordcloud)

data1 <- readLines("c:/data/seoul.txt")

useSejongDic()

buildDictionary(ext\_dic="sejong",

user\_dic=data.frame(c("초등학교","시장님","어린이집"),c

("ncn")),replace\_usr\_dic=T) **#사전추가**

help("buildDictionary")

data2 <- extractNoun(data1)

data2

head(unlist(data2), 30)

data3 <- unlist(data2)

data3 <- Filter(function(x) {nchar(x) >= 2} ,data3)

data3

data3 <- gsub("\\d+","", data3)

data3 <- gsub("서울시","", data3)

data3 <- gsub("서울","", data3)

data3 <- gsub("요청","", data3)

data3 <- gsub("제안","", data3)

data3 <- gsub(" ","", data3)

data3 <- gsub("-","",data3)

data3 <- gsub("O+","",data3)

data3

write(unlist(data3),"c:/data/seoul\_2.txt")

data4 <- read.table("c:/data/seoul\_2.txt")

data4

nrow(data4)

wordcount <- table(data4)

wordcount

head(sort(wordcount, decreasing=T),50)

library(RColorBrewer)

palete <- brewer.pal(9,"Set3")

wordcloud(names(wordcount),

freq=wordcount,

scale=c(2,0.5),

random.order=F,

random.color=T,

colors=palete)

library(wordcloud2)

wordcloud2(wordcount)



**display.brewer.all() #색상표보기**

**palete<- brewer.pal(9,"set3") #brewer.pal(팔레트의색의수, 팔레트 이름)**

Dark2(8),Accent(8),paired(12),pastel1(9),pastel2(8),Set1(9), set2(8),set3(12) #괄호안은 색상갯수

**display.brewer.pal(8,"Dark2") #어떤색상인지 찝어서확인하기**

[문제181] jeju.txt 분석하세요.

**#사전으로 등록시켜야 할 단어들 파일로 떨어뜨려놓고 사전에추가한다. 벡터형으로 일일이 넣을 필요없이**

**>useSejongDic()**

**>buildDictionary(ext\_dic="sejong",**

**user\_dic=data.frame(readLines("c:/data/제주도관광지.txt"),"ncn"),**

**replace\_usr\_dic=T)**

library(ggmap)

names<-unique(total)

da<-data.frame(lon=c(),lat=c())

for (i in 2:length(names)){

da[i,]<-geocode(enc2utf8(names[i]))

}

da<-na.omit(da)

da<-da[-c(6,26),]

names<-names[-c(6,23,27)]

cen<-c(mean(da$lon),mean(da$lat))

map<-get\_googlemap(center=cen,maptype="roadmap",zoom=11,markers=da)

ggmap(map,extent="device")

gmap<-ggmap(map,extent="device")

gmap+geom\_text(data=da,aes(x=lon,y=lat),size=3,label=names)

#------------

[문제] 하나 이상의 연이어 있는 숫자를 찾기

jeju<-readLines("c:/data/jeju.txt")

grep("[[:digit:]]{1,}",data,value=T) #숫자 나오는데 1번이상나오는 거.

library(stringr)

table(unlist(str\_extract\_all(data,"[[:digit:]]{1,}"))) #digit 숫자만 뽑은거

#숫자 자리에 왼쪽 오른쪽으로 문자들까지 보여줘

[문제] 숫자 앞 뒤에 문자가 있거나 없는 경우

table(unlist(str\_extract\_all(data,"[[:alpha:]]{0,}[[:digit:]]{1,}[[:alpha:]]")))

알파

#{0,} : 있거나 없거나

#어떤 문자가 오는지 패턴을 알아볼 수 있다.

#어떤게 오는지 알고싶으면 찾고자하는 문자를 찾아야한다. 텍스트마이닝에서 많이 쓰임

**> table(unlist(str\_extract\_all(data,"[[:alpha:]]{0,}[[:digit:]]{1,}[[:alpha:]]")))**

000원 00까 00출 08시 09시 1000원 10분 10시 10월 1100고 1100도

1 1 1 2 1 2 3 4 2 1 1

1112번 11시 11일 12시 12인 15분 16도 17시 18일 1곳 1박

1 2 1 1 1 2 1 1 1 1 1

1시 1위 1일 2000원 2014년 20도 20분 20일 21일 21코 22일

1 1 8 1 1 1 1 1 2 1 1

23일 24평 2박 2시 2일 2층 30까 30분 31일 3곳 3박

1 1 4 2 9 1 1 5 1 1 2

3월 3일 40분 4500원 4W 4륜 4박 4시 4일 500원 5500원

2 12 1 2 1 1 1 3 8 1 1

5곳 5분 5시 5월 5일 6000원 6분 6시 6월 6일 700원

1 1 2 7 4 1 3 3 1 1 1

78시 7시 7월 7일 8시 8월 9시 매표9시 오전10시 오후5시 오후6시

1 3 1 2 3 1 2 1 2 1 1

오후7시 올레7코 한라산1100고

1 2 1

**#<우도 앞뒤로 문자있는 것 찾기>**

**> unlist(str\_extract\_all(data,"[[:alpha:]]{0,}우도[[:alpha:]]{0,}"))**

[1] "우도" "우도" "우도" "우도가시기"

[5] "우도" "우도" "우도를" "우도"

[9] "우도는" "우도유람선" "우도에" "우도관광은"

[13] "우도관광" "우도관광" "우도항출발" "우도봉"

[17] "우도박물관" "우도까지" "우도안에" "우도"

[21] "우도" "우도" "우도" "우도는"

[25] "우도" "우도항" "우도는" "우도"

[29] "우도" "우도" "우도키다리아저씨"

#중문앞뒤로 문자열있는거 찾기

**> unlist(str\_extract\_all(data,"[[:punct:]]{0,}중문[[:punct:]]{0,}"))**

[1] "중문" ",중문" "(중문)" "중문" "중문" "중문" "-중문" "-중문" "중문" "중문"

[11] "-중문" "-중문" "-중문/" "-중문" "중문" "-중문" "중문" "중문" "중문" "중문"

[21] "중문" "-중문" "중문" ",중문" "중문" ",중문" "중문" "중문" "중문" "중문"

[31] "중문"

#문자있는 거

**> unlist(str\_extract\_all(data,"[[:punct:]]{0,}중문[[:punct:]]{0,}[[:alpha:]]{0,}"))**

[1] "중문랜드" ",중문흑돼지맛집" "(중문)"

[4] "중문관광단지" "중문해수욕장" "중문관광단지"

[7] "-중문단지내" "-중문관광단지" "중문관광단지"

[10] "중문해수욕장" "-중문관광단지" "-중문관광단지"

[13] "-중문/믿거나말거나박물관" "-중문단지내" "중문단지"

[16] "-중문단지내" "중문" "중문관광단지"

[19] "중문에는" "중문" "중문관광단지"

[22] "-중문관광단지" "중문랜드" ",중문흑돼지맛집"

[25] "중문랜드" ",중문흑돼지맛집" "중문"

[28] "중문관광단지" "중문에는" "중문"

[31] "중문에"

**> unlist(str\_extract\_all(data,"[[:alpha:]]{0,}[' ']{0,}폭포[[:punct:]]{0,}[[:alpha:]]{0,}"))**

**#띄워쓰기 체크 ex)천지연 폭포**

[1] "천제연폭포" "천제연폭포" "정방폭포"

[4] "천지연폭포-이중섭미술거리" "천지연 폭포.새연교" "천지연폭포-숙소"

[7] "천지연폭포-쇠소깍" "천지연폭포-중문관광단지" "천지연폭포-중문관광단지"

[10] "천지연폭포-숙소" "천지연 폭포.새연교" "천제연폭포,선임교"

[13] "정방폭포" "바다와 폭포의" "천지연폭포보다는"

[16] "정방폭포가" "폭포만" "천지연폭포"

[19] "천지연 폭포는" "천제연폭포와" "천지연폭포를"

[22] "입장하셔서 폭포까지" "천지연 폭포[새연교" "천지연폭포/새연교"

[25] "천지연폭포-숙소" "정방폭포-쇠소깍" "천제연폭포"

[28] "천지연폭포" "천지연폭포"

#이런식으로 정규식 표현식을 사용하여 바꿔버리면 gsub을 많이 사용할 필요가 없다.

20180820

# 스크롤링

중앙일보사이트 들어가기 - 빅데이터 검색

인터넷 창에서 오른쪽클릭(또는 F12) - 요소검사

요소선택 - 원하는 제목 클릭-아래창에서 열리는 주소 더블클릭해서 복사하기

-> 근데 하나하나 다 클릭해서 복사하고 붙여넣기하면 힘들지 않을까,,

중앙일보사이트 들어가기 - 빅데이터 검색 - 주소 복사,붙여넣기

<https://search.joins.com/joongangnews?keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&searchcategorytype=JoongangNews>

2페이지 갔다가 다시 1페이지 오기 - 주소 복사,붙여넣기

<https://search.joins.com/JoongangNews?page=1&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews>

주소가 바뀜(page=1추가됨)

2페이지 - 주소 복사,붙여넣기

<https://search.joins.com/JoongangNews?page=2&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews>

페이지별로 page=1, page=2만 다름

-> for문을 돌리면 주소를 한번에 불러올수있겠다!

# 중앙일보-빅데이터 1페이지 R에서 불러오기

**install.packages('rvest')**

**library(rvest)**

html<-read\_html('https://search.joins.com/JoongangNews?page=1&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews')

**>html**

{xml\_document}

<html>

[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=utf-8">\n<meta charset="utf-8">\n<m ...

[2] <body class="aside\_off">\r\n <input type="hidden" value=""><!-- preLoad Image --><img src="https ...

중앙일보사이트 들어가기 - 빅데이터 검색

- 인터넷 창에서 오른쪽클릭(또는 F12) - 요소검사

- 요소선택 - 원하는 제목 클릭-

url 뽑기

**url<-html\_nodes(html,".list\_default .headline")%>%**

**html\_nodes("a")%>%**

**html\_attr("href")**

url

[1] "<https://news.joins.com/article/22897894>" "<https://news.joins.com/article/22896642>"

[3] "<https://news.joins.com/article/22896160>" "<https://news.joins.com/article/22894890>"

[5] "<https://news.joins.com/article/22891818>" "<https://news.joins.com/article/22888630>"

[7] "<https://news.joins.com/article/22887668>" "<https://news.joins.com/article/22885090>"

[9] "<https://news.joins.com/article/22883527>" "<https://news.joins.com/article/22883485>"

addr<-read\_html("http://news.baidu.com/ns?word=80%E5%90%8E%2090%E5%90%8E%20%E6%B6%88%E8%B4%B9&pn=0&cl=2&ct=1&tn=news&rn=20&ie=utf-8&bt=0&et=0")

**url<-html\_nodes(**addr**,".list\_default .headline")%>%**

**html\_nodes("a")%>%**

**html\_attr("href")**

html<-read\_html("https://news.joins.com/article/22896642")

html

{xml\_document}

<html class="no-js" xmlns:fb="http://ogp.me/ns/fb#" lang="ko" xml:lang="ko">

[1] <head>\n<title>'폭염'도 견디는 에너지 자립도시 만든다...전기도 알아서 생산</title>\n<meta charset="utf-8">\n<meta http-equiv="C ...

[2] <body class="aside\_off">\r\n <input type="hidden" value=""><!-- preLoad Image --><img src="https://i ...

txt<-html\_nodes(html,"#article\_body")%>%

html\_text()

**10번 페이지까지 100개의 url을 긁어다가 변수에 전부 넣어라**

**그리고 그변수안에 있는 문서들을 이용해서 단어 빈도수 체크**

**wordcloud 그려보기**

**-> for문 이용하기**

html<-NULL

url1<-NULL

for(i in 1:10){html1<-read\_html(paste0("https://search.joins.com/JoongangNews?page=",

i,

"&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews"))

url1<-append(url1,html\_nodes(html1,".list\_default .headline")%>%

html\_nodes("a")%>%

html\_attr("href"))}

html1

{xml\_document}

<html>

[1] <head>\n<meta http-equiv="Content-Type" content="text/html; charset=utf-8">\n<meta charset="utf-8">\n<meta name="viewport" ...

[2] <body class="aside\_off">\r\n <input type="hidden" value=""><!-- preLoad Image --><img src="https://images.joins.com/ ...

url1

[1] "<https://news.joins.com/article/22897894>" "<https://news.joins.com/article/22896642>"

[3] "<https://news.joins.com/article/22896160>" "<https://news.joins.com/article/22894890>"

[5] "<https://news.joins.com/article/22891818>" "<https://news.joins.com/article/22888630>"

[7] "<https://news.joins.com/article/22887668>" "<https://news.joins.com/article/22885090>"

[9] "<https://news.joins.com/article/22883527>" "<https://news.joins.com/article/22883485>"

[11] "<https://news.joins.com/article/22883482>" "<https://news.joins.com/article/22883441>"

[13] "<https://news.joins.com/article/22883430>" "<https://news.joins.com/article/22881715>"

[15] "<https://news.joins.com/article/22880232>" "<https://news.joins.com/article/22880222>"

[17] "<https://news.joins.com/article/22879465>" "<https://news.joins.com/article/22879051>"

[19] "<https://news.joins.com/article/22876278>" "<https://news.joins.com/article/22874636>"

[21] "<https://news.joins.com/article/22873347>" "<https://news.joins.com/article/22871963>"

[23] "<https://news.joins.com/article/22871130>" "<https://news.joins.com/article/22871134>"

[25] "<https://news.joins.com/article/22870226>" "<https://news.joins.com/article/22870209>"

[27] "<https://news.joins.com/article/22869564>" "<https://news.joins.com/article/22869542>"

[29] "<https://news.joins.com/article/22864665>" "<https://news.joins.com/article/22864411>"

[31] "<https://news.joins.com/article/22863788>" "<https://news.joins.com/article/22863646>"

[33] "<https://news.joins.com/article/22863533>" "<https://news.joins.com/article/22863144>"

[35] "<https://news.joins.com/article/22859990>" "<https://news.joins.com/article/22857226>"

[37] "<https://news.joins.com/article/22857089>" "<https://news.joins.com/article/22856346>"

[39] "<https://news.joins.com/article/22854243>" "<https://news.joins.com/article/22853066>"

[41] "<https://news.joins.com/article/22853065>" "<https://news.joins.com/article/22851397>"

[43] "<https://news.joins.com/article/22850772>" "<https://news.joins.com/article/22849907>"

[45] "<https://news.joins.com/article/22847745>" "<https://news.joins.com/article/22846572>"

[47] "<https://news.joins.com/article/22846463>" "<https://news.joins.com/article/22846454>"

[49] "<https://news.joins.com/article/22845072>" "<https://news.joins.com/article/22844465>"

[51] "<https://news.joins.com/article/22842218>" "<https://news.joins.com/article/22842021>"

[53] "<https://news.joins.com/article/22842017>" "<https://news.joins.com/article/22841703>"

[55] "<https://news.joins.com/article/22841114>" "<https://news.joins.com/article/22838774>"

[57] "<https://news.joins.com/article/22838305>" "<https://news.joins.com/article/22838194>"

[59] "<https://news.joins.com/article/22838188>" "<https://news.joins.com/article/22834802>"

[61] "<https://news.joins.com/article/22831484>" "<https://news.joins.com/article/22831273>"

[63] "<https://news.joins.com/article/22831149>" "<https://news.joins.com/article/22831066>"

[65] "<https://news.joins.com/article/22827272>" "<https://news.joins.com/article/22823772>"

[67] "<https://news.joins.com/article/22823744>" "<https://news.joins.com/article/22823718>"

[69] "<https://news.joins.com/article/22822931>" "<https://news.joins.com/article/22818857>"

[71] "<https://news.joins.com/article/22817999>" "<https://news.joins.com/article/22817215>"

[73] "<https://news.joins.com/article/22817199>" "<https://news.joins.com/article/22817068>"

[75] "<https://news.joins.com/article/22816998>" "<https://news.joins.com/article/22816534>"

[77] "<https://news.joins.com/article/22815343>" "<https://news.joins.com/article/22812078>"

[79] "<https://news.joins.com/article/22810768>" "<https://news.joins.com/article/22810661>"

[81] "<https://news.joins.com/article/22810642>" "<https://news.joins.com/article/22810636>"

[83] "<https://news.joins.com/article/22807594>" "<https://news.joins.com/article/22807583>"

[85] "<https://news.joins.com/article/22807001>" "<https://news.joins.com/article/22806090>"

[87] "<https://news.joins.com/article/22805002>" "<https://news.joins.com/article/22803464>"

[89] "<https://news.joins.com/article/22803411>" "<https://news.joins.com/article/22802489>"

[91] "<https://news.joins.com/article/22802109>" "<https://news.joins.com/article/22801154>"

[93] "<https://news.joins.com/article/22791421>" "<https://news.joins.com/article/22789657>"

[95] "<https://news.joins.com/article/22786652>" "<https://news.joins.com/article/22785017>"

[97] "<https://news.joins.com/article/22783685>" "<https://news.joins.com/article/22783582>"

[99] "<https://news.joins.com/article/22783507>" "<https://news.joins.com/article/22783508>"

txt1<-NULL

for(i in 1:length(url1)){txt1<-append(txt1,html\_nodes(read\_html(url1[i]),"#article\_body")%>%

html\_text())}

txt1

library(stringr)

unlist(str\_extract\_all(txt1,

'[[:alpha:]]{0,}[[:alnum:]]{0,}[" "]{0,}기술[" "]{0,}[[:alnum:]]{0,}[[:punct:]]{0,}'))

txt1<-gsub('[[:punct:]]',' ', txt1)

library(KoNLP)

t<-extractNoun(txt1)

t

t1 <- Filter(function(x){nchar(x)>=2},unlist(t))

head<-head(sort(table(unlist(t1)),decreasing = T),1000)

library(wordcloud)

wordcloud(names(head),head,

colors=brewer.pal(8,'Dark2') ,

scale=c(5,0.5),

random.order=T,random.color=T)

-------------------------------------------------

영화평점란을 분석해보자.

1번페이지 url

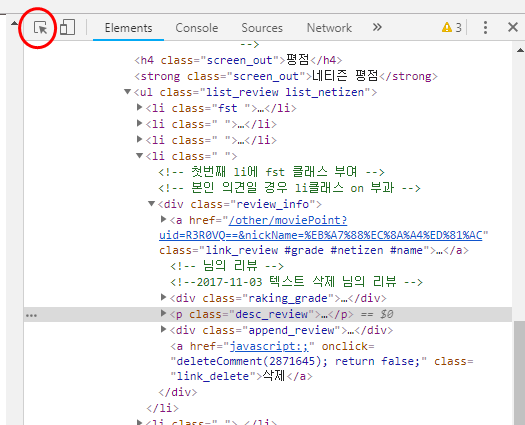
<https://movie.daum.net/moviedb/grade?movieId=120166&type=netizen&page=1>

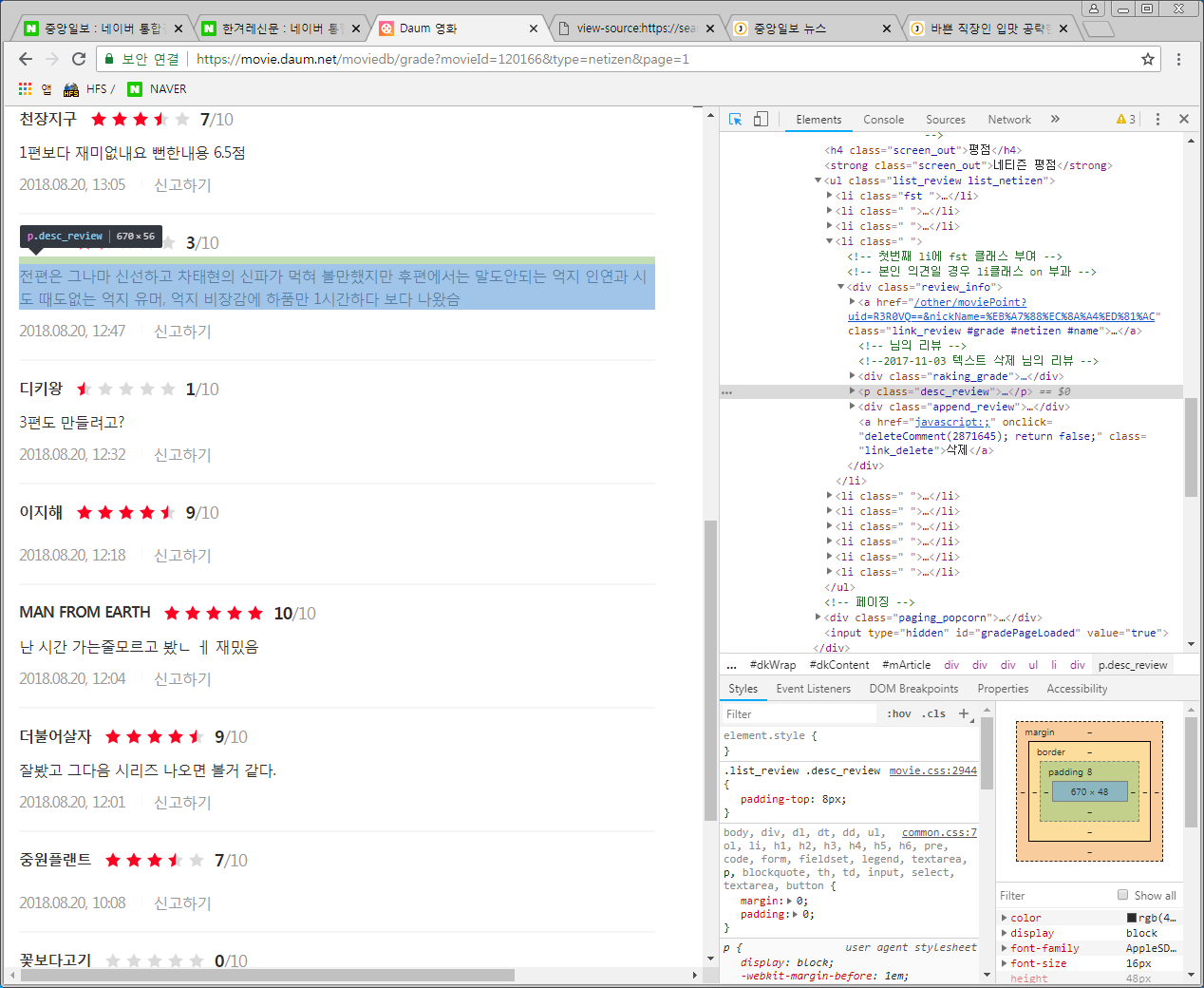
2번페이지 url

<https://movie.daum.net/moviedb/grade?movieId=120166&type=netizen&page=2>

url에서 보면 다똑같은데 페이지번호만 다르다.

요소확인으로 리뷰 어디있는지 눌러보기





이런식으로 위치를 찾는다.

---------------------

8-20

참고

<http://blog.naver.com/PostView.nhn?blogId=nife0719&logNo=220982416637>

<http://blog.eairship.kr/23?category=409828> -html 에 대해서

--html 에 관하여

- read\_html(url): url로부터 html을 읽어옵니다.

1)html(Hyper Text Markup Language) : www 을 통해 볼수 있는 문서를 만들 때 사용하는 언어,

문서의 글자 크기, 글자색,글자모양,문서의 이동을 정의하는 명령어로서 홈페이지를 작성하는데 사용한다, html.명령어= tag(태그)이며 (<>) 꺽쇠를 사용하여 표현한다.

대소문자 구분이 없다.

2) tag: html에서 사용하는 명령어, 꺽쇠로표현,

시작태그와 종료태그('/'로 시작)로 이루어져있다.

(elements : 요소 : 시작,종료태그로 이루어진 모든 명령어를 나타낸다.

속성attribute: 명령을 구체화시키는 것으로 요소의 시작태그 안에서 사용된다.

-추가적인 정보를 전달하고 이름='값' 형식으로 사용된다.

예를 들면>

<div='my-class'></div> : div태그가 class(attribute)라는 값이 'my-class'(attribute의 value)인attribute를 가지고 있다.

라고 해석가능하다.>

변수arguments : 속성과 관련된 값)

시작태그 앞에 /를 넣어서 종료가능 기본형식이다.<a></a>로

3) 문서의 기본형식

<html>

<head> #머리글

<title></title>#제목

<head>

<body>#본문 : 대부분의 태그가 여기에 들어감

</body>

</html>

<html>문서내용</html> : 문서의 시작과끝

<head></head> : 문서의 head 내용

<title></title> : 문서의 제목 >page이름 위의 진짜 창이름이 된다.

<body></body> : 문서의 body?진짜 문서의 내용을 나타낸다.

<http://blog.naver.com/PostView.nhn?blogId=phm0515&logNo=220651225137&widgetTypeCall=true>

>>존잼 참곸ㅋㅋ

- html\_nodes(data, find\_class): data에서 find\_class에 해당하는 부분(노드)을 추출합니다.

tag,class,id(id만은 html\_node()로 찾는다.)를 모두를 찾을수있다.

- html\_text(): html\_nodes뒤에 쓰이며, 그 안에서 text 부분만 추출

-html\_attrs() : attribute을 추출

- paste0(): 문자열을 붙여줍니다.

- strsplit(): 문자열을 특정 패턴으로 잘라줍니다.

- write.csv(): matrix나 data frame을 csv 파일 형태로 저장해줍니다.

/html

제가 공부하면서 겪은 R크롤링에서의 유형 두 가지는,

1. url 주소창에 page주소가 그대로 나오는 유형 (비교적 쉬움)

2. url 주소창에 page주소가 그대로 나오지 않고 진짜 주소가 숨겨진 유형 (꽁꽁 숨기면 짜증남)

중앙일보 들어가기

-------------

Q중앙일보 기사중에 빅데이터에 대해 모아서 분석해주세요

--익스플로어로 통일!

- 사이트에 들어가서 돋보기 모양(검색창)에서 빅데이터 검색!

더보기해서,

유니코드,아스트릭코드값으로 변형해서 검색,

우리는 이거를 긁어야한다.

쫘악 복사해서해야하는데, 이건

요소 검사 ( F12번키도 가능)

-페이지 내 어디부분인지를 크롬, 익스플로우에서 제공해준다.

요소 선택 Ctrl+B

원하는거 클릭

>공통적인 태그나 검색된다. 열린다.

(먼소리냐)

<https://news.joins.com/article/22896642>

주소 복사

href의

url이다.

우리가 이 페이지에서 복사할 것은 url이다.

전부 복사를 개별 불가능

웹 스콜링 : 전부 긁어 버리기 라는 것이다.

- 스콜링은 곧 정보 = 돈이기에 >막는 곳도 있다.

변수에 url를 담아놓고

~~~

안들림...

-------------

-------------

Q가전제풀 살때 가장먼저 하는일은?

네이버검색

가격비교사이트 ? 그럼 가격정보를 어떻게 얻는가?

스콜링을 통한 가격정보 확면에 뿌리는것

우리도 스콜링을 통해서 분석을 할 수 있어야 한다.

-이것을 우리는 포토폴리오로 만들어야 한다.

-------------

-------------

빅데이터 검색한 url를 r이 알아 듣게 하려면

1.install.packages('rvest')

library(rvest)

<https://search.joins.com/joongangnews?keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&searchcategorytype=JoongangNews>

<https://search.joins.com/JoongangNews?page=2&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews>

<https://search.joins.com/JoongangNews?page=3&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews>

page=2>만 다르다.

count변수와 sep=''로 포문을 돌려주면 된다.

-우리는 html 변수에 주소를 넣어볼 것이다.

저거 처음들어가면 페이지 없고 두번째세번째 페이지 누르고 1페이지 돌아오면 페이지 생겨 page=1생겨!

<https://search.joins.com/joongangnews?page=1&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&searchcategorytype=JoongangNews>

html<-read\_html('https://search.joins.com/joongangnews?page=1&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&searchcategorytype=JoongangNews')

html

> html

{xml\_document}

<html>

[1] <head>\n<meta http-equiv="Content-Type" content="text/htm ...

[2] <body class="aside\_off">\r\n <input type="hidden" ...

-사이트에 들어가서 1번째 기사와 2번째 차이점

-a href의 태그를 찾아야한다. >>클릭하고 오른쪽을 보면된다.

..진심 뭔말..

</a>이게 태그 끝을 알리는거임

근데 제목들을 다 클릭해보면

태그이름이다 똑같아 그래서 저 사이트에a태그의 href내용을 긁어오는거야

url<-html\_nodes(html,'.list\_default .headline')%>% #여기서 상위은 어떤것이 와도 된다.

html\_nodes('a')%>% #'a'태그의

html\_attr('href') #속성 href의 값을 불러들인다.

url

> url

[1] "<https://news.joins.com/article/22897894>"

[2] "<https://news.joins.com/article/22896642>"

[3] "<https://news.joins.com/article/22896160>"

[4] "<https://news.joins.com/article/22894890>"

[5] "<https://news.joins.com/article/22891818>"

[6] "<https://news.joins.com/article/22888630>"

[7] "<https://news.joins.com/article/22887668>"

[8] "<https://news.joins.com/article/22885090>"

[9] "<https://news.joins.com/article/22883527>"

[10] "<https://news.joins.com/article/22883485>"

-------------

"<https://news.joins.com/article/22897894>" 여기 주소로 들어가서,

그 주소에서 F12를 클릭한뒤 본문을 요소선택하면 맨아래에 div#article\_body가 마지막태그임을 알수있다.

#공통적 태그를 찾기위해하는것

두번째 기사도 div#article\_body가 맨 마지막 태그임을 알수 잇다.

우리는 공통태그가 div#article\_body임을 알수 잇따.

루프구조를 이용해서

html<-read\_html('https://news.joins.com/article/22896642')

html

여기서 찾고 싶은 것은 article\_body이다.

txt<-html\_nodes(html,'#article\_body')%>%

html\_text()

-변수data html에서 #article\_body의 class, id, tag와, 그 안에 txt를 리턴해주시오

txt

> txt

[1] "\r\n 폭염ㆍ가뭄 등 글로벌 기후변화에 대응하기 위한 도시 내 에너지 자립 계획을 정부가 내놨다. 도시 내에서 직접 에너지를 생산해 기후변화에 대응하자는 취지다. 프랑스 가르단 지방에 설치된 태양광 패널의 모습. 과기정통부는 19일, '도시 발전 시범프로젝트' 추진을 검토한다고 밝혔다. 폭염 등 기후변화를 극복하기 위한 도시 내 에너지 자립 계획이 프로젝트안에 포함됐다. [로이터=연합뉴스] 과학기술정보통신부(이하 과기정통부)는 19일 기후변화 대응기술을 기반으로, 이상기후에 대응하기 위한 '도시 발전(發電) 시범프로젝트' 추진을 검토한다고 발표했다. 그간 정부가 지원해온 기후변화 대응 원천기술의 성과를 실제 도시의 에너지 생산에 적용한다는 계획이다. ━ 지구면적의 2% 차지하는 도시, CO2배출은 70% '도시 발전'이란 태양광ㆍ풍력ㆍ연료전지 등 다양한 신재생 에너지원을 이용해 도시 내에서 에너지를 자립적으로 생산하고, 인공지능(AI)ㆍ빅데이터 등을 활용해 에너지의 공급 및 수요를 최적화하는 것을 말한다. 정부는 지구 면적의 2%에 불과한 도시에 전 세계 인구의 절반 이상이 거주하면서 세계 에너지의 78%를 소비하고, 이산화탄소 배출랴의 70%를 배출하고 있다며 '도시 발전'의 필요성을 언급했다. 도시의 특성상 열섬현상으로 인한 전력 소비가 과다할 수 밖에 없고 이것이 기후변화에 악영향을 미치는 악순환이 반복되고 있다고 정부는 설명했다. [과학기술정보통신부] 과기정통부는 \"지구면적의 2%에 불과한 도시가 세계 에너지의 78%를 소비하고, 세계 이산화탄소 배출량의 70%를 차지하고 있다\"며 도시 발전의 필요성을 강조했다. 인공 건축물과 아스팔트ㆍ자동차가 많은 특성상, 도시 중심부의 기온이 주변 지역보다 현저하게 올라가는 '열섬현상'이 그간 빈번하게 발생했고, 이로 인해 에어컨 등 전력 소비가 급증해 기후 변화에 악영향을 미치는 등 악순환이 반복됐다는 이유다. ━ 기존 기술혁신 통해 에너지 직접 생산ㆍ전달ㆍ소비하는 도시 만든다 정부는 태양전지ㆍ에너지저장(ESS) 기술을 비롯한 5개 기후기술 개발을 지원하고 2025년에는 출연연을 중심으로 도시발전 R&D 성과를 실생활에 적용한 실증단지를 조성해 나가겠다고 밝혔다. 사진은 미국항공우주국이 개발한 '종이접기 활용 태양전지판'의 프로토타입.[사진제공=미국항공우주국] 이를 극복하기 위해 정부는 도시 발전에 먼저 필요한 5개 분야의 기후기술 개발을 지원하기로 했다. 태양전지ㆍ에너지저장(ESS)ㆍ수소연료전지ㆍ에너지 하베스팅ㆍ신재생 하이브리드 기술이 그것이다. 태양전지는 기존 상용화된 실리콘 소재의 한계를 극복할 페로브스카이트ㆍ염료 감응 태양전지 등 차세대 기술을 개발하기로 했다. 대규모 공간을 차지했던 기존과 달리, 형태를 곡면으로 만들어 다양한 장소에 설치할 수 있도록 하는 것이 목표다. 정부가 제시한 도시발전 개념도(정부과천청사). 기존의 도시 에너지 공급방식은 도시 외부 대규모 중앙 집중형 발전소에서 일방적으로 수급받는 방식이었다면, 도시 발전은 차세대 신재생에너지 분산발전을 중심으로 도시 내에서 에너지를 직접 생산ㆍ전달ㆍ소비하는 방식이다. [과학기술정보통신부] 또 그간 ESS용 핵심 소재로 주목받아 왔던 '리튬 이온 전지'가 폭발 위험성이 있는 것을 고려해, 이를 극복하기 위한 차세대 기술을 개발할 예정이다. 압력ㆍ진동ㆍ빛 등 일상에서 버려지는 에너지를 수확해 전력으로 변환하는 '에너지 하베스팅'에 쓰일 고효율 소재도 만들 계획이다. ━ 진천 친환경 에너지타운 벤치마킹한 '도시발전 연구개발 실증단지' 조성 이런 기술은 내년부터 정부출연연구소 건물을 중심으로 실제 적용된다. 정부는 이러한 내용의 '소규모 선도사업'을 추진하고 이 성과를 바탕으로 2025년에는 '도시발전 연구개발 실증단지'를 조성한다는 계획이다. 충북 진천에 들어선 친환경에너지타운. 건물 지붕위에 태양광 시설이 마련돼 있다. [미래창조과학부] 이진규 과기정통부 제1차관은 지난 17일 충북 진천 친환경 에너지타운에서 신재생에너지 융·복합 기술 발전 방안에 대한 산학연 간담회를 개최하고, \"진천 친환경 에너지타운의 성과를 벤치마킹하면서 복잡한 도시 구조물에 기후 기술을 적용할 수 있도록 해야한다\"고 강조했다. 진천 친환경 에너지타운은 2016년 11월 완성된 미래형 에너지 커뮤니티로, 태양광ㆍ지열ㆍ하수 폐열 등 다양한 신재생에너지를 활용해, 도시가 필요한 에너지를 공급하고 있다. 이 차관은 \"우리의 목표는 혁신적 기후기술이 국민 생활에 적용될 수 있도록 하는 것\"이라며 \"차세대 기후기술이 결집된 도시발전을 통해 폭염ㆍ혹한 등 기후변화를 완화할 수 있도록 노력하겠다\"고 밝혔다. 허정원 기자 heo.jeongwon@joongang.co.kr \r\n\r\n\r\n"

------------------------------------------

1.page 1~10까지

1page당 10 총 100의 url를 긁어다가 변수에 모든것을 넣고

변수의 문서들의 단어를 분석해 wordcloud를만들고해라

-for문을 돌려... page 와 url에 100개가 들어가게

1.for문

#url 추출

url<-c()

for(i in 1:10){

html<- read\_html(paste('https://search.joins.com/joongangnews?page=',i,'&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&searchcategorytype=JoongangNews',seq=''))

url <-c(url,html\_nodes(html,'.headline')%>%

html\_nodes('a')%>%

html\_attr('href')

)

}

#본문 추출

news<-c() #담아야 하니까 만들어 준다.

for(i in 1: length(url)){

html<-read\_html(url[i]) #url를 순차적으로 불러들인다.

temp<-html\_node(html,'#article\_body')%>%html\_text()

news<-c(news,temp)

}

news

--------쌤답------------------------

url <- c()

for (i in 1:10){

html<- read\_html(paste("https://search.joins.com/JoongangNews?page=",i,"&Keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&SortType=New&SearchCategoryType=JoongangNews",sep=""))

url <- c(url, html\_nodes(html,".headline") %>%

html\_nodes('a')%>%

html\_attr('href'))

}

news <- c()

for (i in 1:length(url)){

html<-read\_html(url[i])

temp <- html\_node(html, "#article\_body")%>%

html\_text()

news <- c(news,temp)

}

news

---------------------조선일보---------------------

빅데이터 검색

조선일보가 아닌 다른 뉴스사이트가 걸린다.

조선일보만주의 해서 해야한다.

dl.search\_news에서도

동일한 url를 같은 페이지 안에서도 여러개이다.

그니까 dt,dd에 같은 유알엘이 존재하는것이다. 결국 애네들의 가장 상위인,

dd.thumb이다.

search\_news\_box하면불필요하게 밑에 잇는데도 뽑아버린다.

그래서 dd class='thumb' >> 가 적격이다.

태그의 a href를 하면 된다............ 일단...... 넘어가자...........

-url를 모아 저장,

-본문들어가서 text 긁어오기

-본문을 클릭하면 par가 분단 별로 나뉜다.

상위인 artcle.news\_article 에서

div.news\_body\_all 로 해보세요...

1.url만들기

html<- read.html('http://search.chosun.com/search/news.search?query=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&pageno=1&orderby=news&naviarraystr=&kind=11000&cont1=&cont2=&cont5=&categoryname=%EC%A1%B0%EC%84%A0%EC%9D%BC%EB%B3%B4&categoryd2=&c\_scope=paging&sdate=&edate=&premium=')

html<-read.html('http://search.chosun.com/search/news.search?query=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&pageno=2&orderby=news&naviarraystr=&kind=11000&cont1=&cont2=&cont5=&categoryname=%EC%A1%B0%EC%84%A0%EC%9D%BC%EB%B3%B4&categoryd2=&c\_scope=paging&sdate=&edate=&premium=')

#1,2페이지의 차이

pageno=1,2로 구분된다.

> 5페이지 까지만 for문을 돌려보면

url<-c()

for(i in 1:5){

html<-read.html(paste('http://search.chosun.com/search/news.search?query=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&pageno=',i,'&orderby=news&naviarraystr=&kind=11000&cont1=&cont2=&cont5=&categoryname=%EC%A1%B0%EC%84%A0%EC%9D%BC%EB%B3%B4&categoryd2=&c\_scope=paging&sdate=&edate=&premium=',sep=''))

url<-c(url,html\_nodes(html,'.search\_news'))%>%html\_nodes('a')%>%html\_attr('href')

}

---------------------한겨레---------------------

여기선 pageseq=~로 표기된다.

<http://search.hani.co.kr/Search?command=query&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&media=news&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.20&pageseq=0>

<http://search.hani.co.kr/Search?command=query&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&media=news&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.20&pageseq=1>

html<- read\_html('http://search.hani.co.kr/Search?command=query&keyword=%EB%B9%85%EB%8D%B0%EC%9D%B4%ED%84%B0&media=news&sort=d&period=all&datefrom=2000.01.01&dateto=2018.08.20&pageseq=0

')

이런거 상위항목 찾을때 열린거를 닫아보자, .search-result..

그다음 title은 걍 head니까.. dl로 왔을때 안에 항목을 하나하나 확인해서 주소가 중복되는지 확인해야한다.

-------다음 영화 신과함께 영화 평점--------------

<https://movie.daum.net/moviedb/grade?movieId=120166&type=netizen&page=1>

<https://movie.daum.net/moviedb/grade?movieId=120166&type=netizen&page=2>

<https://movie.daum.net/moviedb/grade?movieId=120166&type=netizen&page=3>

Q이 영화에 대해서만 분석하고 싶을때

-현재 문장을 전부 긁어오고 싶을때, 문장에 대해서 요소선택을 통해 클릭했을때,

나온다. 뭐가???

젤밑에 보더라도

리뷰\_인포 > desc\_리뷰 일때

둘다 사용한다고??

>>>>>?????태그의 기준이 뭐냐..

화면 F12 눌러서 리퓨 클릭

>review\_info

>desc\_review

---------------------naver\_영화----------------------------

<https://movie.naver.com/movie/bi/mi/point.nhn?code=167697>

Q코드 가 안바뀜

-요소선택을 해보자

paging - div - a id 'pagerTagAnchor1'

-------여기서는 page번호 생김-=----

<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=152661&target=after&page=1>

<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=152661&target=after&page=2>

<https://movie.naver.com/movie/point/af/list.nhn?st=mcode&sword=152661&target=after&page=3>

>평가는

list\_netizen-tbody-tr-td.title-a class

>>처음부터 타이틀로 가면 안될수도있기에 타이틀의 상위개념으로 가서

리스트네티즌도 써준다.

우리가 좋아하는 영화에 대한 리뷰를 분석해서 할수 있다.

-기사검색

-영화리뷰

-내가 좋아하는 회사에 대해서

>> 제출한 목적으로 리포트를 작성하시오,,(작은프로젝트)

corpus(말뭉치)를 사용하여 분석해야할 경우도있다.

말뭉치는 자연언어 연구를 위해 특정한 목적을 가지고 언어의 표본을 추출한 집합이다.

**install.packages("tm")**

**library(tm)**

**> data1<-readLines("c:/data/tm\_example.txt")**

**> data1**

[1] "I like apple and banana, but hate orang." "I love banana, but not mango"

[3] "I hate peach, but like cherry" "I want to eat strawberry ~~!"

**> Corpus(VectorSource(data1))**

<<SimpleCorpus>>

Metadata: corpus specific: 1, document level (indexed): 0

Content: documents: 4

**> corp1<-Corpus(VectorSource(data1))**

**#documents : tm 패키지가 작업할 수 있는 특별한 형태를 의미 일반적으로 1줄이 1개의document가 된다.**

#하나의 줄을 documents라고 한다. 하나의 뭉치

#네개의 뭉치가있는 거임

**> summary(corp1) #corpus 내용 보는 방법**

Length Class Mode

1 2 PlainTextDocument list

2 2 PlainTextDocument list

3 2 PlainTextDocument list

4 2 PlainTextDocument list

**> inspect(corp1) #**

<<SimpleCorpus>>

Metadata: corpus specific: 1, document level (indexed): 0

Content: documents: 4

[1] I like apple and banana, but hate orang. I love banana, but not mango

[3] I hate peach, but like cherry I want to eat strawberry ~~!

**> corp1[[1]] #첫번째 방에있는 리스트를 보여줘**

<<PlainTextDocument>>

Metadata: 7

Content: chars: 42

메타데이터는 공백을 포함해서 7개의 덩어리

42개의 글자수

**> corp1[[1]]$meta #1번방의 메타자료를 보여줘**

author : character(0)

datetimestamp: 2018-08-24 01:04:11

description : character(0)

heading : character(0)

id : 1

language : en

origin : character(0)

논문 주석처리같은거 할때 이렇게 본다.(?)

**> corp1[[1]]$content #1번방의 문서내용**

[1] "I like apple and banana, but hate orang."

문장들을 개별로분석. 단어를 추출해서 첫번째 문서에도있었나 두번째문서에도있었나...이런식으로 보고싶을 때.

이걸 matrix형으로 바꿔줘야한다.

**## tm 패키지가 분석할 수 있는 Term-Dcoument 형식의 matrix로 변환해야함**

**> TermDocumentMatrix(corp1)**

<<TermDocumentMatrix (terms: 15, documents: 4)>>

Non-/sparse entries: 20/40

Sparsity : 67%

Maximal term length: 10

Weighting : term frequency (tf)

**tdm<-TermDocumentMatrix(corp1)**

**m<-as.matrix(tdm) #변환완료**

**> m**

Docs

Terms 1 2 3 4

and 1 0 0 0

apple 1 0 0 0

banana 1 1 0 0

but 1 1 1 0

hate 1 0 1 0

like 1 0 1 0

orang 1 0 0 0

love 0 1 0 0

mango 0 1 0 0

not 0 1 0 0

cherry 0 0 1 0

peach 0 0 1 0

eat 0 0 0 1

strawberry 0 0 0 1

want 0 0 0 1

1,2,3,4 가 document번호

**tdm<-TermDocumentMatrix(corp1,control=list(wordLengths=c(1,Inf)))**

**원래 기본값은 두 글자** 부터 분석하는데 이걸 제어하기위해 wordLengths를 사용했음 inf=Infinite

**> tdm<-TermDocumentMatrix(corp1,control=list(wordLengths=c(1,Inf)))**

**> m<-as.matrix(tdm)**

**> m**

Docs

Terms 1 2 3 4

and 1 0 0 0

apple 1 0 0 0

banana 1 1 0 0

but 1 1 1 0

hate 1 0 1 0

**i 1 1 1 1**

like 1 0 1 0

orang 1 0 0 0

love 0 1 0 0

mango 0 1 0 0

not 0 1 0 0

cherry 0 0 1 0

peach 0 0 1 0

eat 0 0 0 1

strawberry 0 0 0 1

to 0 0 0 1

want 0 0 0 1

#옵션은 help 기능 이용하면서 쓰자. ?TermDocumentMatrix

**corpus에서 정제작업을 할때 gsub( )을 쓸 수 없다.**

corpus에서 제공하는 연산자를 쓰거나 방법을 바꿔서 써야한다.

**#여러개의 공백을 하나의 공백으로 변환**

**corp2 <- tm\_map(corp1,stripWhitespace)**

**#대문자가 있을 경우 소문자로 변환**

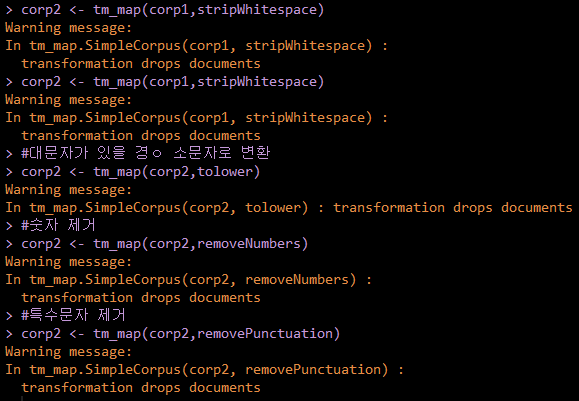
**corp2 <- tm\_map(corp2,tolower)**

**#숫자 제거**

**corp2 <- tm\_map(corp2,removeNumbers)**

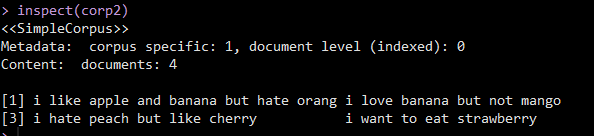
**#특수문자 제거**

**corp2 <- tm\_map(corp2,removePunctuation)**

****

에러뜨긴했는데 다 됬네

**inspect(corp2)**



> corp3 <- gsub('~','',corp1)

> corp3

[1] "c(\"I like apple and banana, but hate orang.\", \"I love banana, but not mango\", \"I hate peach, but like cherry\", \"I want to eat strawberry !\")"

[2] "list(language = \"en\")"

[3] "list()"

**#기존의 메소드들을 그대로 쓰면 말뭉치가 망가진다. 사용할 수 없다.**

**별도의 함수를 만들어 줘야한다.**

**tostring <- content\_transformer(function(x,from,to)gsub(from,to,x))**

함수의 인수값(말뭉치, 변환해야할 문자, 변환할 문자)

tostring이라는 함수를 **function(x,from,to)gsub(from,to,x) 이런식으로 만든것이다.**

이 규칙을 따라서 만들어야한다.

**corp3 <- tm\_map(corp1, tostring,"~","")** #물결모양을 찾아서 없는걸로바꿔주세요

**corp3 <- tm\_map(corp1, tostring,"!","")**

**corp3 <- tm\_map(corp1, tostring,",","")**

**#불용어를 제거(전치사, 관사 등)**

**stopwords('en') #불용어사전에 의거해서 자동삭제**

**sword <- c('and','but','not') #불용어를 따로지정해줌**

**corp2 <- tm\_map(corp2,removeWords,sword)**

**sword <- c(stopwords('en')'and','but','not') #불용어사전도 포함해서 지우고자하면 이렇게 추가 해줄 수있다.**

**corp2 <- tm\_map(corp2,removeWords,sword)**

inspect(corp2)

**> inspect(corp2)**

<<SimpleCorpus>>

Metadata: corpus specific: 1, document level (indexed): 0

Content: documents: 4

[1] i like apple banana hate orang i love banana mango i hate peach like cherry

[4] i want to eat strawberry

tdm2 <- TermDocumentMatrix(corp2) #tdm 이런식으로나오는애들은 보통 말뭉치관련된 것들

m2 <- as.matrix(tdm2)

m2

**> tdm2 <- TermDocumentMatrix(corp2) #tdm 이런식으로나오는애들은 보통 말뭉치관련된 것들**

**> m2 <- as.matrix(tdm2)**

**> m2**

Docs

Terms 1 2 3 4

apple 1 0 0 0

banana 1 1 0 0

hate 1 0 1 0

like 1 0 1 0

orang 1 0 0 0

love 0 1 0 0

mango 0 1 0 0

cherry 0 0 1 0

peach 0 0 1 0

eat 0 0 0 1

strawberry 0 0 0 1

want 0 0 0 1

**#행단위 단어의 수**

**> freq1 <- sort(rowSums(m2),decreasing=T)**

**> freq1**

banana hate like apple orang love mango cherry peach

2 2 2 1 1 1 1 1 1

eat strawberry want

1 1 1

두번이상 언급된 단어의 수 3개

**#열단위 단어의 수**

**> freq2 <- sort(colSums(m2),decreasing=T)**

**> freq2**

1 3 2 4

5 4 3 3

**#특정 횟수 이상 언급된 단어들만 출력**

**> findFreqTerms(tdm2,2)**

[1] "banana" "hate" "like"

**#특정 단어와 상관관계를 찾고 싶을 경우**

**> findAssocs(tdm,"apple",0.5)**

$apple

orang banana hate like

1.00 0.58 0.58 0.58

library(RColorBrewer)

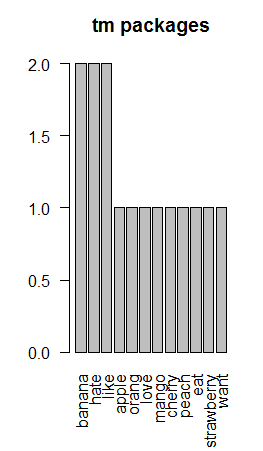
library(wordcloud)

palete <- brewer.pal(7,"Set3")

wordcloud(names(freq1),freq=freq1,min.freq=1,colors=palete)



barplot(freq1,main="tm packages",las=2)



ggplot()

행렬의 곱

**mm <- m2%\*%t(m2)**

**mm**

Terms

Terms apple banana hate like orang love mango cherry peach eat strawberry want

apple 1 1 1 1 1 0 0 0 0 0 0 0

banana 1 2 1 1 1 1 1 0 0 0 0 0

hate 1 1 2 2 1 0 0 1 1 0 0 0

like 1 1 2 2 1 0 0 1 1 0 0 0

orang 1 1 1 1 1 0 0 0 0 0 0 0

love 0 1 0 0 0 1 1 0 0 0 0 0

mango 0 1 0 0 0 1 1 0 0 0 0 0

cherry 0 0 1 1 0 0 0 1 1 0 0 0

peach 0 0 1 1 0 0 0 1 1 0 0 0

eat 0 0 0 0 0 0 0 0 0 1 1 1

strawberry 0 0 0 0 0 0 0 0 0 1 1 1

want 0 0 0 0 0 0 0 0 0 1 1 1

관계도를 그린 것

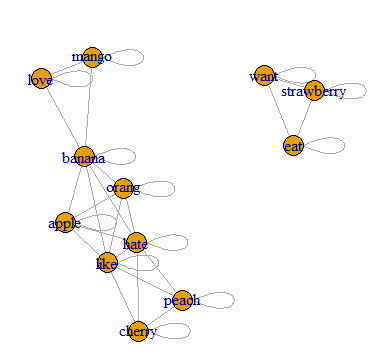
이걸로 그래프를 그려보자.

**install.packages("igraph")**

**library(igraph)**

**> g<-graph.adjacency(mm,weighted = T,mode = "undirected")**

**> plot(g)**



**> g<-graph.adjacency(mm,weighted = T,mode = "undirected")**

**> g2 <- simplify(g)**

**> plot(g2)**

