

0. R설치

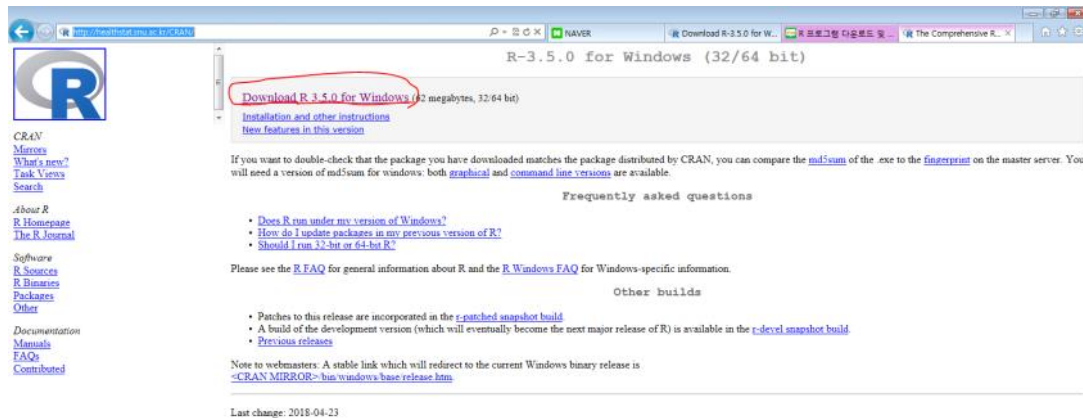
2018년 5월 8일 화요일 오후 2:46

■ R 설치

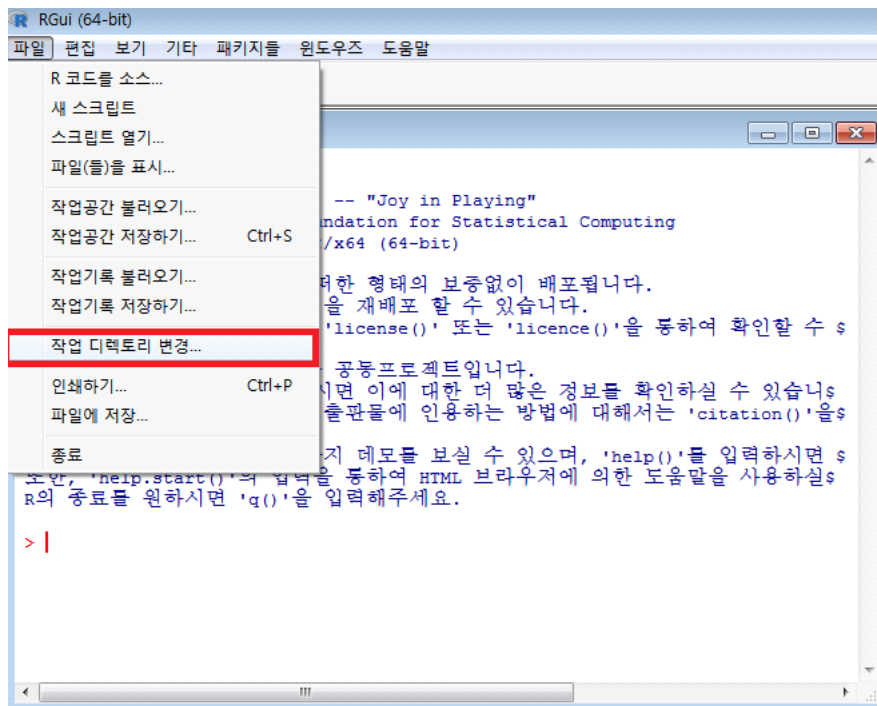
R : <http://healthstat.snu.ac.kr/CRAN/>

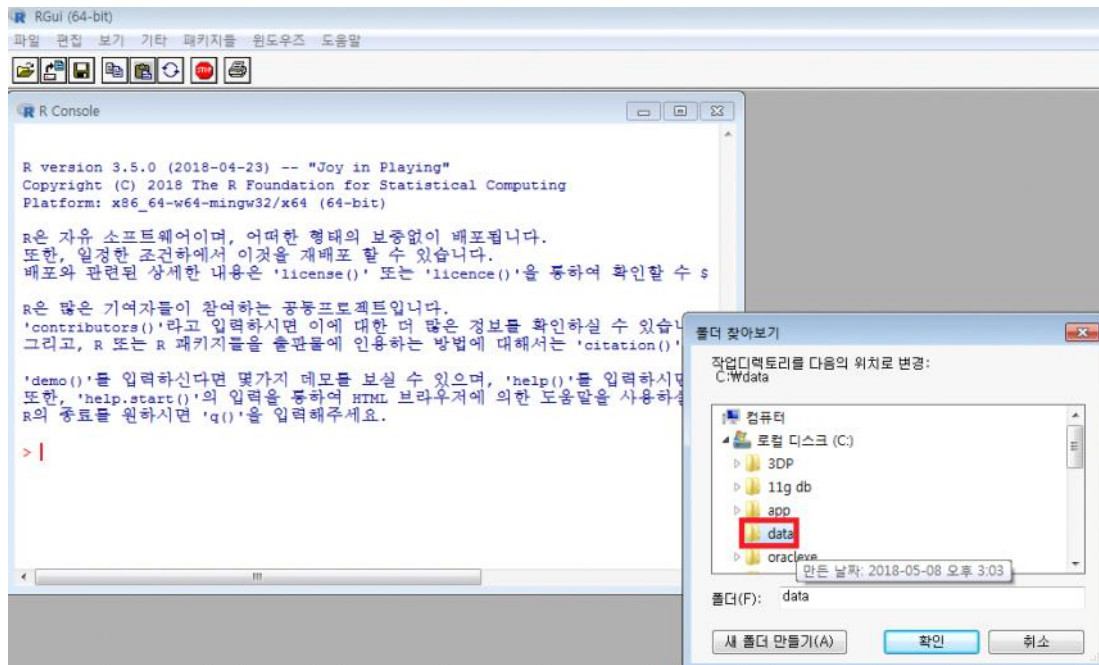
R Studio : <https://www.rstudio.com/products/rstudio/download/>

R스튜디오는 R을 먼저 다운받아야 사용 가능.



다음다음다음 누른다..





* c드라이버에 만든 폴더를 경로로 지정해 주자.

```
> emp<-read.csv("emp.csv", head=T)
> attach(emp)
> tapply(sal,list(deptno,job),sum)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10         NA  1300      2450        5000         NA
20      6000  1900      2975          NA         NA
30         NA   950      2850          NA      5600
70         NA 3200         NA          NA         NA
> |
```

--정상적으로 동작한다.

■ R에서 csv파일 사용하기

1. R실행

2. [menu] -> [change directory]에서 csv파일이 들어있는 디렉토리로 이동

3. [변수명] <- read.csv("[파일명]",header=T) 로 파일을 읽어들인다.

1. 변수명은 말 그대로 변수명이다. 적당한 이름을 지정

2. 파일명은 csv파일명이다. 따옴표 잊지말것

3. header는 열의 이름이다. csv파일에 열의 이름이 있을 경우에는T를 열의 이름이 없이1행부터 데이터가 들어있는 경우에는F로 하면 된다.

예) > input <- read.csv("a.csv",header = T)

1열에 행이름을 넣었을 경우 row.names 옵션을 줘서 이름을 인식시킬 수 있다.

예) > input <- read.csv("a.csv", header = T, row.names=1)

데이터를 읽어들인 후에 attach명령을 사용하면 행이름을 바로 참조해서 사용할 수 있다.

예) > attach(input)

1. R이란?

2018년 5월 8일 화요일 오후 3:13

1. R을 사용하는 이유

1. 무료 사용 가능하다.
2. data 분석을 위해 가장 많이 사용하는 통계 플랫폼이다.
3. 복잡한 데이터를 다양한 그래프로 표현할 수 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 OS에도 설치 가능하다. (심지어 아이폰에서도 설치가 가능)

R은 통계, 기계학습, 금융, 생물정보학, 그래픽스에 이르는 다양한 통계 패키지를 갖추고 있으며 이 모든 것이 무료로 제공된다. 또한 셀 수 없이 많은 R을 사용한 통계분석 서적, 기계학습 서적이 존재한다.

2. R의 특징

- R은 인터프리터 언어 (ex. 컴파일러 방식 = c언어 등)
- 대소문자 구분 (굉장히 예민)
- ↑ 방향으로 이전에 했던 작업 수행 가능
- q() 사용하면 R종료
- 작업하는 내용을 저장하거나 작업용 데이터를 보관하는 작업 디렉토리를 지정하는 것이 좋음
- 스크립트창을 열어서 코드를 입력한 후 해당 부분을 선택한 후 ctrl + R을 눌러서 실행

: 주석

> : 명령 프롬프트

+ : 여러 줄에 명령을 칠 때 다음줄의 가장 왼쪽에 생성됨

3. 기본 자료형

1) 6가지 기본 자료형

- 문자형(character) : 문자, 문자열
- 수치형(numeric)
 - 정수(integer) - 1L, 20L (L부호는 정수형으로 데이터를 저장하도록 R에게 알려준다)
 - 실수(double) - 1, 20, 3.14, 2.1
- 복소수형(complex) : 실수 + 허수
- 논리형 (logical) : TRUE(T, 1), FALSE(F, 0)

2) 특수한 형태의 값

- NULL : 데이터의 값이 존재하지 않는다는 의미이다.

- NA : missing value, 결측값, 손실된값으로 값이 없음을 의미한다. (하나의 요소의 의미)
- NaN(not a number) :수학적으로 정의되지 않은 값
- Inf, -Inf :무한대
- 자료형 확인 함수 : mode(), typeof() --> 자료형 확인

3. 작업 디렉토리 지정하기 & 화면에 결과 출력

■ setwd("디렉토리명")

- R로 작업을 할때 필요한 데이터들을 미리 가져다 두는 약속된 디렉토리
- 작업 후 나오는 결과물들도 기본적으로 저장됨
- 작업하기 전 디렉토리 생성 후 분석할 소스 데이터들을 생성한 디렉토리로 옮겨놓고 작업

■ print() / cat()

- R 콘솔 화면에서 print 명령을 이용하여 내용 출력
- print() : 한 번에 한 가지만 출력 가능 (단점)
- cat() : 여러 개 출력 가능, 복잡한 데이터 형태는 출력 불가(행렬, 리스트 등)
- 여러 개의 명령을 연속적으로 실행하고 싶을 경우 세미콜론(;) 사용

4. 패키지 사용법

구분	설명
패키지 설치	install.packages("패키지명") - 추가 패키지는 인터넷으로 다운받기 때문에 인터넷 연결이 되어야함.
패키지 업데이트	update.packages("패키지명") - 설치되어 있는 패키지를 업데이트 - 패키지명을 입력하지 않으면 모든 패키지 업데이트
설치된 패키지들의 경로 확인	.libpaths() > .libPaths() [1] "C:/Program Files/R/R-3.5.0/library"
설치된 패키지 확인	installed.packages() > installed.packages() <div> Package assertthat "assertthat" base "base" BH "BH" bindr "bindr" bindrcpp "bindrcpp" boot "boot" class "class" cli "cli" cluster "cluster" codetools "codetools" compiler "compiler" </div>
설치 되어있는 패키지 삭제	remove.packages("패키지명")
특정 패키지의 정보 확인	library(help=패키지명)

5. 예제

문제 1. 아래의 결과를 R로 구현 하시오.

```

      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10         NA  1300      2450      5000         NA
20      6000  1900      2975         NA         NA
30         NA   950      2850         NA      5600
70         NA 3200         NA         NA         NA
> |

```

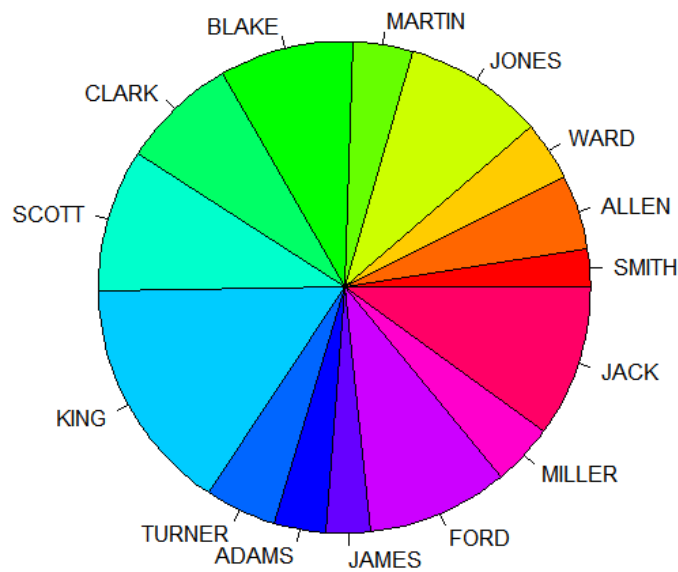
```

> emp<-read.csv("emp.csv", header=T)    -- R스튜디오에선 emp<-read.csv("c:\\data\\emp.csv",header=T)
> attach(emp)
> tapply(sal,list(deptno,job),sum)

```

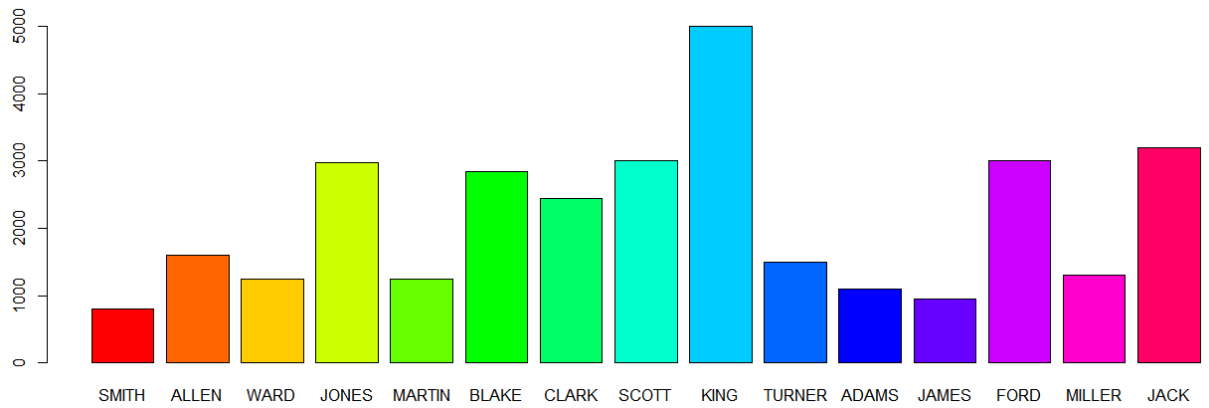
문제 2.사원 테이블의 월급을 시각화 하시오.

```
pie(emp$sal, label=emp$ename, col=rainbow(15))
```



문제 3.사원 테이블의 월급을 막대그래프로 시각화 하시오.

```
barplot(emp$sal, col=rainbow(15),names.arg=emp$ename) -- col=rainbow (15가지 무지개 색으로 표현)
```



문제 4.	누구든지 유용한 패키지를 생성해서 공유할 수 있고 새로운 기능에 대한 전달이 빠르다는 장점을 코드로 구현해 보시오.
-------	--

```
install.packages("networkD3")
install.packages("dplyr")
```

#####전체 코드#####

```
library(networkD3)
```

```
library(dplyr)
```

```
# data set 소설 레미제라블 인물 관계도
```

```
data(MisLinks, MisNodes)
```

```
head(MisNodes)
```

```
head(MisLinks)
```

```
# plot
```

```
D3_network_LM<-forceNetwork(Links = MisLinks, Nodes = MisNodes,
```

```
  Source = 'source', Target = 'target',
```

```
  NodeID = 'name', Group = 'group',opacityNoHover = TRUE,
```

```
  zoom = TRUE, bounded = TRUE,
```

```
  fontSize = 15,
```

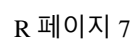
```
  linkDistance = 75,
```

```
  opacity = 0.9)
```

```
D3_network_LM
```

```
# html 발사
```

```
networkD3::saveNetwork(D3_network_LM, "D3_LM.html", selfcontained = TRUE)
```



2. R의 자료구조

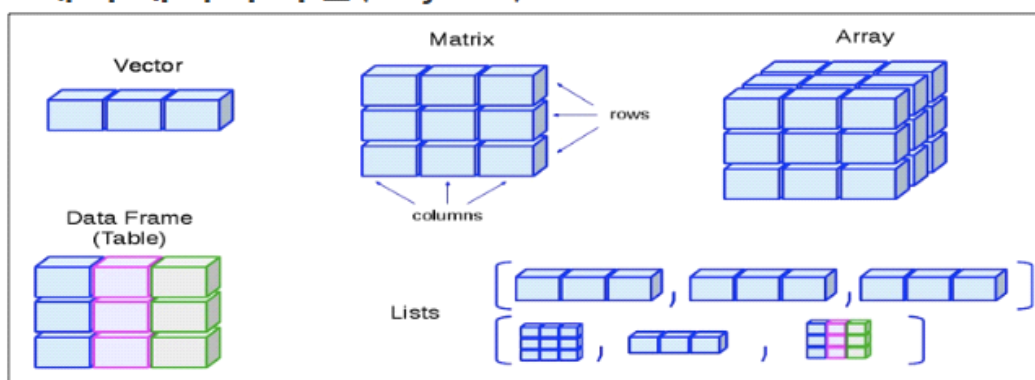
2018년 5월 8일 화요일 오후 4:15

1. 자료구조의 종류

1. **vector** : 같은 데이터 타입을 갖는 1차원 배열 구조
2. **matrix(행렬)** : 같은 데이터 타입을 갖는 2차원 배열 구조
3. **array** : 같은 데이터 타입을 갖는 다차원 배열 구조
4. **data frame** : 각각의 데이터 타입을 갖는 컬럼으로 이루어진 2차원 배열구조
5. **list** : 서로 다른 데이터 구조(vector, data frame, matrix, array)의 데이터 타입이 중첩된 구조



R에서 데이터 타입(Objects)



문제 5. 데이터 프레임이 데이터베이스의 테이블과 유사한 R의 자료형이다. Emp의 구조가 data frame임을 확인하시오.

str(emp) # 오라클의 desc emp 와 같다.


```
> str(emp)
'data.frame': 15 obs. of 8 variables:
 $ empno : int 7369 7499 7521 7566 7654 7698 7782 7788 7839 7844 ...
 $ ename : Factor w/ 15 levels "ADAMS","ALLEN",...: 13 2 15 8 10 3 4 12 9 14 ...
 $ job : Factor w/ 5 levels "ANALYST","CLERK",...: 2 5 5 3 5 3 3 1 4 5 ...
 $ mgr : int 7902 7698 7698 7839 7698 7839 7839 7566 NA 7698 ...
 $ hiredate: Factor w/ 13 levels "1980-12-17","1981-02-20",...: 1 2 3 4 8 5 6 12 9 7 ...
 $ sal : int 800 1600 1250 2975 1250 2850 2450 3000 5000 1500 ...
 $ comm : int NA 300 500 NA 1400 NA NA NA NA 0 ...
 $ deptno : int 20 30 30 20 30 30 10 20 10 30 ...
```

문제 6. Emp 데이터 프레임에서 이름, 월급을 출력 하시오.

`emp [, c("ename","sal")] -- emp[행, 열] c --> combine의 약자`

```
> emp [ , c("ename","sal")]
  ename sal
1 SMITH 800
2 ALLEN 1600
3 WARD 1250
4 JONES 2975
5 MARTIN 1250
6 BLAKE 2850
7 CLARK 2450
8 SCOTT 3000
9 KING 5000
10 TURNER 1500
11 ADAMS 1100
12 JAMES 950
13 FORD 3000
14 MILLER 1300
15 JACK 3200
```

문제 7. 월급이 3000인 직원들의 이름, 월급을 출력 하시오.

`emp[emp$sal==3000, c("ename","sal")]`

```
> emp[emp$sal==3000, c("ename","sal")]
  ename sal
8 SCOTT 3000
13 FORD 3000
```

문제 8. 월급이 2000이상인 직원들의 이름, 월급을 출력 하시오.

`emp[emp$sal >= 2000, c("ename","sal")]`

```
> emp[emp$sal >= 2000, c("ename","sal")]
  ename sal
4 JONES 2975
6 BLAKE 2850
7 CLARK 2450
8 SCOTT 3000
9 KING 5000
13 FORD 3000
15 JACK 3200
```

3. 벡터(vector)

2018년 5월 15일 화요일 오후 8:01

■ 벡터 (vector)

- 여러 개의 동일한 형태의 데이터를 모아서 함께 저장한다.
- **c(combine value)** 함수 또는 **seq(sequence value)** 를 사용해서 생성
- 동일한 데이터형이 저장되어야 함
- 다른 유형의 데이터가 있을 경우 **강제 형변환 또는 에러발생**



```
> c(1,2,3,4,5)
```

```
[1] 1 2 3 4 5
```

```
> # 숫자와 문자데이터가 섞여있을 경우 강제형변환 (모두다 문자로)
```

```
> c(1,2,3,4,"5")
```

```
[1] "1" "2" "3" "4" "5"
```

```
> # 벡터명을 지정하여 데이터 입력
```

```
> vec1 <- c(1,2,3,4,5)
```

```
> vec1
```

```
[1] 1 2 3 4 5
```

■ 사용법

1. 특정 위치 값 제어

- 특정 항목의 요소를 보고싶을 경우 : 벡터명[번호]
- 기존 벡터에 새로운 데이터 추가 : 벡터명 <-데이터

#ex. **vec<-c(vec,1,3)** vec값에 1,3이 추가됨

- Append 함수를 사용하여 데이터 추가 가능 : **append(벡터명, 추가할 데이터, after=3)**

3번째 뒤(4) 자리에 데이터 추가 (만약 사이가 비어있다면 na값이 들어감)

2. 벡터로 연산하기

- 벡터는 여러건의 동일한 데이터가 들어있으므로 자체 연산가능
- 집합이라고도 함

- union : 데이터형이 다를 경우 두 집합의 합집합
- Setdiff : 두 집합간의 차이 값 ex) `setdiff(var1,var2)` # var1에는 있지만 var2에는 없는 요소 출력
- Intersect : 두 집합간에 공통적으로 있는 요소 찾기 (교집합)

3. 벡터의 각 컬럼에 이름지정

- names() 함수를 이용해서 값만 있는 벡터에 컬럼명을 지정 할 수 있다.

```
> fruits <- c(10,20,30)
> fruits
[1] 10 20 30
>
> names(fruits) <- c('apple','banana','peach')
> fruits
apple banana peach
  10    20    30
```

4 벡터에 연속적인 데이터 할당 : seq(), rep() 함수사용

```
# seq()
> var5 <- seq(1,5)
> var5
[1] 1 2 3 4 5

> var6 <- seq(2,-2)
> var6
[1] 2 1 0 -1 -2

> var7 <- seq(1,10,2) # 1부터 10까지 2씩 증가시켜서 값을 출력
> var7
[1] 1 3 5 7 9

# rep()
> var8 <- rep(1:3)
> var8
[1] 1 2 3

> var9 <- rep(1:3,2) # 1부터 3까지 두번출력
> var9
[1] 1 2 3 1 2 3

> var10 <- rep(1:3,each=2) # 1부터 3까지 출력하되 각각2번씩 출력
> var10
[1] 1 1 2 2 3 3
```

5. 벡터의 길이

- length() : 벡터의 요소가 몇 개인지 반환

- `nrow()` : 행렬일 경우 몇 행인지 구하는 함수
- `NROW()` : 배열의 건수를 구할 수 있음

6. 벡터에 특정 문자의 포함여부 찾기

- `%in%` : 특정 데이터의 존재 유무를 검증하는 방법으로 많이 사용

```
> var7
[1] 1 3 5 7 9
> 3 %in% var7
[1] TRUE
> 4 %in% var7
[1] FALSE
```

1.예제

문제 194.	아래의 숫자들을 출력하는 vector를 생성 하시오. 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
---------	---

```
> x<-c(1:20)
> x
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

문제 195.	아래의 숫자들을 출력하는 vector를 생성 하시오. 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4
---------	--

```
> x2<-c(rep(1,5),rep(2,6),rep(3,8),rep(4,6))
> x2
[1] 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4
```

문제 196.	아래와 같은 벡터를 생성 하시오. <pre>> x scott king jones 1 3 4</pre>
---------	--

```
x<-c(1,3,4)
names(x)<-c("scott","king","jones")
x
```

```
> names(x)<-c("scott","king","jones")
> x
scott king jones
1      3      4
```

문제 197.	아래 x의 요소 중에 2번째 것만 빼고 출력 하시오.
---------	-------------------------------

```
> x[-2]  
scott jones  
1      4
```

4. 팩터(factor)

2018년 5월 23일 수요일 오후 1:57

■ 팩터(factor)

1. 범주(값의 목록)를 갖는 vector
2. factor() 함수를 통해서 생성
3. Factor는 명목형(nominal), 순서형(ordinal) 형식 2가지가 존재
4. Nominal은 level의 순서 값이 무의미 하며 알파벳 순서로 정의
5. Ordinal은 level 순서의 값을 직접 정의해서 원하는 순서로 정의할 수 있다.

1. 예제

```
f1 <- c("middle","low","high")
```

```
f1
```

```
class(f1)
```

```
> f1 <- c("middle","low","high")  
> f1  
[1] "middle" "low"    "high"  
> class(f1)  
[1] "character"
```

```
f2<-factor(f1)
```

```
f2
```

```
class(f2)
```

```
[1] middle low    high  
Levels: high low middle  
> class(f2)  
[1] "factor"
```

```
f2<-factor(f1, order=T, levels = c("low","middle","high"))
```

```
f2
```

```
> f2<-factor(f1, order=T, levels = c("low","middle","high"))  
> f2  
[1] middle low    high  
Levels: low < middle < high
```

문제 198.	f2의 값 목록을 출력하는데 high, middle, low 순서로 정렬해서 출력 하시오.
----------------	--

```
sort(f2, decreasing = T)
```

```
> sort(f2, decreasing = T)
[1] high middle low
Levels: low < middle < high
```

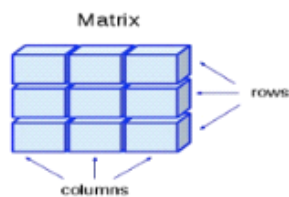
문제 199.	F2의 값 목록을 출력하는데 high, middle, low 순서로 정렬해서 출력 하시오.
----------------	--

5. 행렬(matrix)

2018년 5월 23일 수요일 오후 2:25

■ 행렬 (Matrix)

- Matrix()함수 사용
- 모든 컬럼과 행은 데이터형이 동일 해야함 (같은 타입의 데이터를 갖는 2차원 배열)
- 열 우선으로 입력됨
- nrow = n 값으로 행 값 입력
- byrow = T 가로로 입력을 우선으로
- rbind() : 행 추가
- cbind() : 열 추가



기본 예제	1. matrix의 데이터 조회	2 새로운 행과 열 추가 : rbind(), cbind()
<pre>> mat1 <- matrix(c(1,2,3,4)) > mat1 [1] [1,] 1 [2,] 2 [3,] 3 [4,] 4 > > mat2 <- matrix(c(1,2,3,4), nrow=2) > mat2 [1,] [2,] [1,] 1 3 [2,] 2 4 > > mat3 <- matrix(c(1,2,3,4), nrow=2, byrow=T) > mat3 [1,] [2,] [1,] 1 2 [2,] 3 4</pre>	<pre>> mat3 [1,] [2,] [1,] 1 2 [2,] 3 4 > > mat2[,1] #모든행의 1열값을 출력 [1] 1 2 > > mat3[,1] #모든행의 1열값을 출력 [1] 1 3 > > mat3[1,] #1행의 모든열값을 출력 [1] 1 2 > > mat3[1,1] # 1행의 1열 값을 출력 [1] 1</pre>	<pre>> mat4 <- matrix(c(1,2,3,4,5,6,7,8,9), nrow=3, byrow=T) > mat4 [1,] [2,] [3,] [1,] 1 2 3 [2,] 4 5 6 [3,] 7 8 9 > > # 컬럼의 이름 지정 : colnames > colnames(mat4) <- c('First','Second','Third') > mat4 First Second Third [1,] 1 2 3 [2,] 4 5 6 [3,] 7 8 9 > > # 행추가 > mat4 <- rbind(mat4, c(11,12,13)) > mat4 [1,] [2,] [3,] [1,] 1 2 3 [2,] 4 5 6 [3,] 7 8 9 [4,] 11 12 13</pre>


```
> # 행길이가 다를경우 에러
> mat4 <- rbind(mat4, c(11,12,13,14))

경고메시지:
In rbind(mat4, c(11, 12, 13, 14)) :
  number of columns of result is not a multiple of the number of columns of any of the data sets

> mat4 <- cbind(mat4, c(9,9,9,9))
> mat4
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    9
[2,]    4    5    6    9
[3,]    7    8    9    9
[4,]   11   12   13    9
```

1. 예제

문제 201. 아래와 같은 matrix를 만드시오.

```
matrix(c(1:9),nrow=3,byrow = T)
```

```
> matrix(c(1:9),nrow=3,byrow = T)
      [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

문제 202. 아래와 같은 matrix를 만드시오.

```
x<-matrix(c(1:12),nrow=3,byrow = T)
```

```
> x
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

문제 203. 아래의 행렬합을 구현 하시오.

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} + \begin{array}{ccc} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array} = ?$$

```
matrix(c(1:9),nrow=3,byrow = T) + matrix(c(1:9),nrow=3,byrow = F)
```

```
> matrix(c(1:9),nrow=3,byrow = T) + matrix(c(1:9),nrow=3,byrow = F)
      [,1] [,2] [,3]
[1,]    2    6   10
[2,]    6   10   14
[3,]   10   14   18
```

문제 204. 아래의 행렬의 원소별 곱셈과 행렬곱을 각각 구하시오,

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} + \begin{array}{ccc} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{array} = ?$$

```
a<- matrix(c(1:9),nrow=3,byrow = T)
```

```
b<- matrix(c(1:9),nrow=3,byrow = F)
```

```
a%*%b
```

```
> a<- matrix(c(1:9),nrow=3,byrow = T)
> b<- matrix(c(1:9),nrow=3,byrow = F)
> a%*%b
      [,1] [,2] [,3]
[1,]   14   32   50
[2,]   32   77  122
[3,]   50  122  194
```

6. 배열(array)

2018년 5월 23일 수요일 오후 3:01

1. 같은 데이터 타입을 갖는 다차원 배열 구조
2. matrix는 2차원 행렬이고 array는 3차원 행렬
3. array() 함수를 이용해서 3차원 배열을 생성할 수 있다.

예제 : array(c(1:12), dim=c(3,4))
array(c(1:12), dim=c(2,2,3))
array(c(1:12), dim=c(2,2,2,2))

7. 데이터 프레임(data frame)

2018년 5월 23일 수요일 오후 2:15

■ 데이터 프레임

1. 각기 다른 데이터 타입을 갖는 컬럼으로 이루어진 2차원 테이블 구조
2. data.frame() 함수를 이용하여 생성하며 각 컬럼, 행의 이름을 지정할 수 있다.

1. 신규변수 생성 (transform())

데이터 프레임에서 신규 변수 생성하는 방법은 2가지

1. dataframe\$variable

```
> h_w_d.f
  height weight
1    175     62
2    159     55
3    166     59
4    189     75
5    171     61
6    173     64
7    179     63
8    167     65
9    182     70
10   170     60
```

```
> options(digits=4) # 숫자 개수 지정해주는 옵션. 이거 지정 안해주면 소숫점 5~6자리까지 나옴
> h_w_d.f$bmi_1 <- h_w_d.f$weight/(h_w_d.f$height/100)^2
> h_w_d.f
```

```
  height weight bmi_1
1    175     62 20.24
2    159     55 21.76
3    166     59 21.41
4    189     75 21.00
5    171     61 20.86
6    173     64 21.38
7    179     63 19.66
8    167     65 23.31
9    182     70 21.13
10   170     60 20.76
```

매번 dataframe\$variable 을 입력해줘야만 하는게 꽤 불편하다.

신규 변수를 한 두번 생성 할 시에는 사용하기 괜찮지만, 다수 변수를 이용해서 다수 변수를 신규 생성해야 하는 경우라면 아무래도 손이 많이 가는 방법이다.

2. transform(데이터프레임 , 신규변수명 = 수식, [신규변수명2 = 수식 ..])

```
> h_w_d.f <- transform( h_w_d.f, bmi_2 = weight/(height/100)^2)
```

```
> h_w_d.f
  height weight bmi_1 bmi_2
1    175    62  20.24 20.24
2    159    55  21.76 21.76
3    166    59  21.41 21.41
4    189    75  21.00 21.00
5    171    61  20.86 20.86
6    173    64  21.38 21.38
7    179    63  19.66 19.66
8    167    65  23.31 23.31
9    182    70  21.13 21.13
10   170    60  20.76 20.76
```

dataset\$variable 에서 매번 '\$'를 입력해줘야 하는 번거로움 대비 transform()은 깔끔하다.
거기다가 한꺼번에 여러개의 변수를 생성하는 이점도 있다.

```
> options(digits=3)
> h_w_d.f <- transform(h_w_d.f, bmi_sqrt = sqrt(bmi_2), bmi_log10 = log10(bmi_2) )
> View(h_w_d.f)
```

	height	weight	bmi_1	bmi_2	bmi_sqrt	bmi_log10
1	175	62	20.2	20.2	4.50	1.31
2	159	55	21.8	21.8	4.66	1.34
3	166	59	21.4	21.4	4.63	1.33
4	189	75	21.0	21.0	4.58	1.32
5	171	61	20.9	20.9	4.57	1.32
6	173	64	21.4	21.4	4.62	1.33
7	179	63	19.7	19.7	4.43	1.29
8	167	65	23.3	23.3	4.83	1.37
9	182	70	21.1	21.1	4.60	1.32
10	170	60	20.8	20.8	4.56	1.32

2. 변수 사용 (with(), attach(), detach())

데이터 프레임 변수의 값을 사용하는 방법은 크게 3가지가 있다.

데이터프레임 변수 사용 방법

\$을 이용한 변수 사용	with() 함수 사용	attach() / detach() 사용
df명\$변수명	With(데이터프레임명, 명령어)	attach() Df명 생략 가능 구간 detach()
Ex) max(emp\$sal) # 5000	Ex) with(emp, max(sal)) # 5000	attach(emp) max(sal) # 5000 detach(emp) max(sal) # 오류 발생
<pre>> max(sal) [1] 5000</pre>	<pre>> with(emp,max(sal)) [1] 5000</pre>	<pre>> attach(emp) > max(sal) [1] 5000 > detach(emp) > max(sal) Error: object 'sal' not found</pre>

3. 변수명 변경

1. names() & colnames()

컬럼의 개수만큼 컬럼명을 입력해주지 않으면 나머지 값은 NA로 채워지므로 컬럼의 개수가 얼마 안되거나 컬럼명을 모두 바꾸는 경우에 사용하자.

names(데이터프레임) <- c("변수명1", "변수명2", "변수명3" ...) # 컬럼명의 개수보다 적게 입력하면 나머지는 NA로 채워짐
colnames(데이터프레임) <- c("변수명1", "변수명2", "변수명3" ...) # 컬럼명의 개수보다 더 많이 입력하면 오류발생

```
> names(emp2)<-c("vv1", "vv2", "vv3", "vv4", "vv5")
> emp2
  vv1    vv2    vv3    vv4    vv5    NA    NA    NA
1 7839 KING PRESIDENT NA 1981-11-17 0:00 5000 NA 10
2 7698 BLAKE  MANAGER 7839 1981-05-01 0:00 2850 NA 30
3 7782 CLARK  MANAGER 7839 1981-05-09 0:00 2450 NA 10
4 7566 JONES  MANAGER 7839 1981-04-01 0:00 2975 NA 20
5 7654 MARTIN SALESMAN 7698 1981-09-10 0:00 1250 1400 30
6 7499 ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300 30
7 7844 TURNER SALESMAN 7698 1981-08-21 0:00 1500 0 30
8 7900 JAMES  CLERK 7698 1981-12-11 0:00 950 NA 30
9 7521 WARD   SALESMAN 7698 1981-02-23 0:00 1250 500 30
10 7902 FORD   ANALYST 7566 1981-12-11 0:00 3000 NA 20
11 7369 SMITH  CLERK 7902 1980-12-09 0:00 800 NA 20
12 7788 SCOTT  ANALYST 7566 1982-12-22 0:00 3000 NA 20
13 7876 ADAMS  CLERK 7788 1983-01-15 0:00 1100 NA 20
14 7934 MILLER CLERK 7782 1982-01-11 0:00 1300 NA 10
> colnames(emp2)<-c("mm1", "mm2", "mm3")
> emp2
  mm1    mm2    mm3    NA    NA    NA    NA
1 7839 KING PRESIDENT NA 1981-11-17 0:00 5000 NA 10
2 7698 BLAKE  MANAGER 7839 1981-05-01 0:00 2850 NA 30
3 7782 CLARK  MANAGER 7839 1981-05-09 0:00 2450 NA 10
4 7566 JONES  MANAGER 7839 1981-04-01 0:00 2975 NA 20
5 7654 MARTIN SALESMAN 7698 1981-09-10 0:00 1250 1400 30
6 7499 ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300 30
7 7844 TURNER SALESMAN 7698 1981-08-21 0:00 1500 0 30
8 7900 JAMES  CLERK 7698 1981-12-11 0:00 950 NA 30
9 7521 WARD   SALESMAN 7698 1981-02-23 0:00 1250 500 30
10 7902 FORD   ANALYST 7566 1981-12-11 0:00 3000 NA 20
11 7369 SMITH  CLERK 7902 1980-12-09 0:00 800 NA 20
12 7788 SCOTT  ANALYST 7566 1982-12-22 0:00 3000 NA 20
13 7876 ADAMS  CLERK 7788 1983-01-15 0:00 1100 NA 20
14 7934 MILLER CLERK 7782 1982-01-11 0:00 1300 NA 10
```

2. rename() 함수 이용

rename() 함수를 이용하면 원하는 컬럼명만 바꿀 수 있다. (names()는 컬럼 개수만큼 써주지 않으면 NA값으로 채워짐)

2.1 reshape 패키지의 rename() 함수

install.packages("reshape") #패키지를 추가해주어야 한다.

데이터프레임 <- rename(데이터프레임, c(old변수명1 = "new변수명1", old변수명2 = "new변수명2", ...)

예제

```
install.packages("reshape")
library(reshape)
emp2<-rename(emp2,c(mm2="new변수1"))
```

```
> emp<-rename(emp, c(ename="new이름"))
> emp
  empno new이름 job mgr hiredate sal comm deptno
1 7839 KING PRESIDENT NA 1981-11-17 0:00 5000 NA 10
2 7698 BLAKE  MANAGER 7839 1981-05-01 0:00 2850 NA 30
3 7782 CLARK  MANAGER 7839 1981-05-09 0:00 2450 NA 10
4 7566 JONES  MANAGER 7839 1981-04-01 0:00 2975 NA 20
5 7654 MARTIN SALESMAN 7698 1981-09-10 0:00 1250 1400 30
6 7499 ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300 30
7 7844 TURNER SALESMAN 7698 1981-08-21 0:00 1500 0 30
8 7900 JAMES  CLERK 7698 1981-12-11 0:00 950 NA 30
9 7521 WARD   SALESMAN 7698 1981-02-23 0:00 1250 500 30
10 7902 FORD   ANALYST 7566 1981-12-11 0:00 3000 NA 20
11 7369 SMITH  CLERK 7902 1980-12-09 0:00 800 NA 20
12 7788 SCOTT  ANALYST 7566 1982-12-22 0:00 3000 NA 20
13 7876 ADAMS  CLERK 7788 1983-01-15 0:00 1100 NA 20
14 7934 MILLER CLERK 7782 1982-01-11 0:00 1300 NA 10
```

2.2 plyr 패키지의 rename() 함수

Install.package("plyr") # 패키지를 추가해주어야 한다

Reshape 패키지의 rename과 다른점은 old 변수명도 큰따옴표를 해주어야 한다.

데이터프레임 <- rename(데이터프레임, c(old변수명1 = "new변수명1", old변수명2 = "new변수명2", ...)

예제

```
install.packages("plyr")
```

```
library(plyr)
```

```
> library(plyr)
> emp<-rename(emp,c("job"="직업"))
> emp
  empno new이름      직업 mgr hiredate sal comm deptno
1  7839   KING PRESIDENT  NA 1981-11-17 0:00 5000   NA    10
2  7698  BLAKE  MANAGER  7839 1981-05-01 0:00 2850   NA    30
3  7782  CLARK  MANAGER  7839 1981-05-09 0:00 2450   NA    10
4  7566  JONES  MANAGER  7839 1981-04-01 0:00 2975   NA    20
5  7654 MARTIN SALESMAN  7698 1981-09-10 0:00 1250 1400    30
6  7499  ALLEN SALESMAN  7698 1981-02-11 0:00 1600   300    30
7  7844  TURNER SALESMAN  7698 1981-08-21 0:00 1500    0     30
8  7900   JAMES  CLERK  7698 1981-12-11 0:00  950   NA     30
9  7521   WARD  SALESMAN  7698 1981-02-23 0:00 1250   500    30
10 7902   FORD  ANALYST  7566 1981-12-11 0:00 3000   NA     20
11 7369  SMITH  CLERK  7902 1980-12-09 0:00  800   NA     20
12 7788  SCOTT  ANALYST  7566 1982-12-22 0:00 3000   NA     20
13 7876  ADAMS  CLERK  7788 1983-01-15 0:00 1100   NA     20
14 7934  MILLER  CLERK  7782 1982-01-11 0:00 1300   NA     10
```

4. 컬럼&로우 추가 및 생성(cbind(), rbind())

■ 두 벡터를 각각 컬럼으로 하는 dataframe을 만들고(추가하고) 싶으면? : cbind

예제 1.

명령어	<pre>vec1 <- c('one','two','three') vec2 <- c(1,2,3) cbind(vec1, vec2)</pre>
결과	<pre>## vec1 vec2 ## [1,] "one" "1" ## [2,] "two" "2" ## [3,] "three" "3"</pre>

예제 2.

명령어	<pre>df <- data.frame(cbind(vec1,vec2)) str(df)</pre>
-----	--

결과	<pre>## 'data.frame': 3 obs. of 2 variables: ## \$ vec1: Factor w/ 3 levels "one","three",...: 1 3 2 ## \$ vec2: Factor w/ 3 levels "1","2","3": 1 2 3</pre> <p>위의 방법은 <code>str(df)</code>에서 보듯이, 숫자든, 문자든 모두 factor로 인식하게 됩니다. 그렇다면, <code>cbind.data.frame</code>이라는 함수를 사용</p>
----	---

예제 3.

명령어	<pre>df <- data.frame(cbind(vec1,vec2)) str(df)</pre>
결과	<pre>## 'data.frame': 3 obs. of 2 variables: ## \$ vec1: Factor w/ 3 levels "one","three",...: 1 3 2 ## \$ vec2: Factor w/ 3 levels "1","2","3": 1 2 3</pre> <p>위의 방법은 <code>str(df)</code>에서 보듯이, 숫자든, 문자든 모두 factor로 인식하게 됩니다. 그렇다면, <code>cbind.data.frame</code>이라는 함수를 사용</p>

■ 두 벡터를 각각 **로우**로 하는 dataframe을 만들고(추가하고) 싶으면? : **rbind**

예제 1.

명령어	<pre>vec1 <- c('one','two','three') vec2 <- c(1,2,3) rbind(vec1, vec2)</pre>
결과	<pre>## [,1] [,2] [,3] ## vec1 "one" "two" "three" ## vec2 "1" "2" "3"</pre>

예제 2.

명령어	<pre>data.frame(rbind(vec1,vec2))</pre>
결과	<pre>## x1 x2 x3 ## vec1 one two three ## vec2 1 2 3</pre>

5. 예제

문제 199. 아래의 k1 변수의 결과에서 10만 출력해 보시오.

```
k1 <- data.frame(x=c(1,2,3,4,5),y=c(2,3,4,6,10))
```

```
k1
```

```
k1[5,2]
```

```
> k1[5,2]
[1] 10
> k1
  x y
1 1 2
2 2 3
3 3 4
4 4 6
5 5 10
```

문제 200. 빨간 부분만 출력되게 하시오.

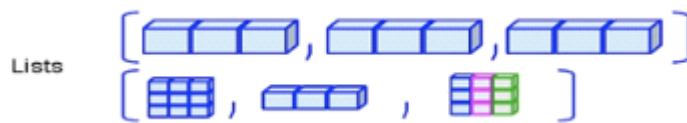
```
> k1[5,2]
[1] 10
> k1
  x y
1 1 2
2 2 3
3 3 4
4 4 6
5 5 10
```

8. 리스트(list)

2018년 5월 15일 화요일 오후 8:27

■ list

- 키, 값 형태로 데이터를 저장하는 일종의 배열
- 서로 다른 데이터 구조(vector, data, frame, array, list)의 중첩된 데이터 구조
- list() 함수를 이용해서 데이터 구조를 중첩할 수 있다.
- 데이터 프레임의 기초



1. List 생성하고 조회하기

- 특정키만 조회하고 싶을 경우 **변수이름\$key값** 형식으로 조회

```
> list1 <- list(name='jenny', address='Seoul', tel='010-xxxx-xxxx', pay=500)
> list1

$name
[1] "jenny"

$address
[1] "Seoul"

$tel
[1] "010-xxxx-xxxx"

$pay
[1] 500

> # 특정키만 조회하고 싶은 경우
> list1$name
[1] "jenny"
>
> list1[1:2]

$name
[1] "jenny"

$address
[1] "Seoul"
```

9. R의 연산자

2018년 5월 8일 화요일 오후 4:25

1. 산술 연산자 : * / + -

2. 비교 연산자 : >, <, >=, <=, ==, !=

3. 논리 연산자 : & (and , 백터화된 연산), && (and , 백터화 되지 않은 연산)
| (or, 백터화된 연산), || (or, 백터화 되지 않은 연산), ! (not)

* 백터화 된 연산

예 : x <- c(1,2,3)

x > c(1,1,1) & x < c(3,3,3)



4. 기타 비교 연산자

SQL	R
In	%in%
Like	grep
Is null	is.na
Between.. And ..	emp\$sal >=1000 & emp\$sal <=3000

1. 예제

문제 9. 직업이 SALESMAN이 아닌 직원들의 이름, 월급, 직업을 출력 하시오.

```
emp[emp$job != 'SALESMAN', c("ename","job")]      # emp [행, 열]
```

```
> emp[emp$job != 'SALESMAN', c("ename","job")]
   ename      job
1  SMITH    CLERK
4  JONES  MANAGER
6  BLAKE  MANAGER
7  CLARK  MANAGER
8  SCOTT  ANALYST
9   KING PRESIDENT
11 ADAMS    CLERK
12 JAMES    CLERK
13  FORD  ANALYST
14 MILLER    CLERK
15  JACK    CLERK
```

문제 10. 1981년 12월 03일에 입사한 직원들의 이름과 입사일을 출력 하시오.

```
emp[emp$hiredate == '1981-12-03' , c("ename","hiredate")]
```

```
> emp[emp$hiredate == '1981-12-03' , c("ename","hiredate")]
  ename hiredate
12 JAMES 1981-12-03
13 FORD 1981-12-03
```

문제 11. 직업이 SALESMAN 이고 월급이 1000이상인 직원들의 이름, 월급, 직업을 출력 하시오.

```
emp[emp$job=='SALESMAN' & emp$sal >= 1000 , c("ename","sal","job")]
```

```
> emp[emp$job=='SALESMAN' & emp$sal >= 1000 , c("ename","sal","job")]
  ename sal job
2 ALLEN 1600 SALESMAN
3 WARD 1250 SALESMAN
5 MARTIN 1250 SALESMAN
10 TURNER 1500 SALESMAN
```

문제 12. 직업이 SALESMAN, ANALYST 인 직원들의 이름, 직업을 출력 하시오.

```
emp[emp$job=='SALESMAN' | emp$job == 'ANALYST' , c("ename","sal","job")]
```

```
> emp[emp$job=='SALESMAN' | emp$job == 'ANALYST' , c("ename","sal","job")]
  ename sal job
2 ALLEN 1600 SALESMAN
3 WARD 1250 SALESMAN
5 MARTIN 1250 SALESMAN
8 SCOTT 3000 ANALYST
10 TURNER 1500 SALESMAN
13 FORD 3000 ANALYST
```

문제 13. 직업이 SALESMAN, ANALYST가 아닌 직원들의 이름, 직업을 출력 하시오.

```
emp[!(emp$job %in% c("SALESMAN","ANALYST")),c("ename","job") ]
```

```
> emp[!emp$job %in% c("SALESMAN","ANALYST"),c("ename","job") ]
  ename job
1 SMITH CLERK
4 JONES MANAGER
6 BLAKE MANAGER
7 CLARK MANAGER
9 KING PRESIDENT
11 ADAMS CLERK
12 JAMES CLERK
14 MILLER CLERK
15 JACK CLERK
```

문제 14. 커미션이 null인 직원들의 이름,월급,커미션을 출력 하시오.

```
emp[is.na(emp$comm) ,c("ename","sal","comm")]
```

```
> emp[is.na(emp$comm) ,c("ename","sal","comm")]
  ename  sal comm
1  SMITH  800  NA
4  JONES 2975  NA
6  BLAKE 2850  NA
7  CLARK 2450  NA
8  SCOTT 3000  NA
9   KING 5000  NA
11 ADAMS 1100  NA
12 JAMES  950  NA
13  FORD 3000  NA
14 MILLER 1300  NA
15  JACK 3200  NA
```

*R에서의 null 값

- 1.NULL : 아무것도 없다 ---> is.null()
- 2.NA : 결손 값 ---> is.na()
- 3.NaN : 비수치 (not a number)---> is.nan()

*null을 활용하는 때

```
x <- NULL
for (i in 1:10)
  x <- append(x,i*i)
X

> x <- NULL
> for (i in 1:10)
+ x <- append(x,i*i)
> x
[1] 1 4 9 16 25 36 49 64 81 100
```

설명 : null (아무것도 없다)를 활용할 때는 반복문으로 처리할 오브젝트의 초기값을 null로 설정할 때 활용한다.

문제 15. 월급이 1000에서 3000 사이인 직원들의 이름,월급을 출력 하시오.

```
emp[emp$sal >= 1000 & emp$sal <=3000 , c("ename","sal")]

> emp[emp$sal >= 1000 & emp$sal <=3000 , c("ename","sal")]
  ename  sal
2  ALLEN 1600
3   WARD 1250
4  JONES 2975
5 MARTIN 1250
6  BLAKE 2850
7  CLARK 2450
8  SCOTT 3000
10 TURNER 1500
11 ADAMS 1100
13  FORD 3000
14 MILLER 1300
```

문제 16. 이름의 첫 글자가 A로 시작하는 직원들의 이름, 월급을 출력 하시오.

```
> emp[grep("^A",emp$ename),c("ename","sal")]
```

```
> emp[grep("^A",emp$ename),c("ename","sal")]
  ename sal
2 ALLEN 1600
11 ADAMS 1100
```

* Grep 관련 설명

^ : 첫 번째

\$: 마지막

. : 한 자리수

* : 와일드 카드

문제 17.	이름의 끝 글자가 T로 끝나는 직원들의 이름, 월급을 출력 하시오.
--------	---------------------------------------

```
> emp[grep("T$",emp$ename), c("ename","sal")]
```

```
> emp[grep("T$",emp$ename), c("ename","sal")]
  ename sal
8 SCOTT 3000
```

문제 18.	이름의 두 번째 철자가 M인 직원들의 이름, 월급을 출력 하시오.
--------	--------------------------------------

```
emp[grep("^..M",emp$ename),c("ename","sal")]
```

```
> emp[grep("^..M",emp$ename),c("ename","sal")]
  ename sal
1 SMITH 800
```

문제 19.	부서번호를 출력하는데 중복제거 해서 출력 하시오.
--------	-----------------------------

```
unique(emp$deptno)
```

```
> unique(emp$deptno)
[1] 20 30 10 70
```

```
> install.packages("data.table")

There is a binary version available but the source version is later:
  binary source needs_compilation
data.table 1.11.0 1.11.2          TRUE

Binaries will be installed
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/data.table_1.11.0.zip'
Content type 'application/zip' length 1825779 bytes (1.7 MB)
downloaded 1.7 MB

package 'data.table' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\Administrator\AppData\Local\Temp\Rtmpe8unLd\downloaded_packages
> library(data.table)
data.table 1.11.0
The fastest way to learn (by data.table authors): https://www.datacamp.com/courses/data-analysis-the-data-table-way
Documentation: ?data.table, example(data.table) and browseVignettes("data.table")
Release notes, videos and slides: http://r-datatable.com
> data.table("부서번호"=unique(emp$deptno))
부서번호
1:      20
2:      30
3:      10
4:      70
```

문제 20. 직업을 출력하는데 중복을 제거해서 출력 하시오.

```
data.table("직업"=unique(emp$job))
```

```
> data.table("직업"=unique(emp$job))
      직업
1:  CLERK
2: SALESMAN
3:  MANAGER
4:  ANALYST
5: PRESIDENT
```

문제 21. 이름과 월급을 출력하는데 월급이 높은 순서대로 출력 하시오.

```
emp[order(emp$sal, decreasing = T), c("ename","sal")]
```

```
> emp[order(emp$sal, decreasing = T), c("ename","sal")]
   ename  sal
9  KING 5000
15 JACK 3200
8  SCOTT 3000
13 FORD 3000
4  JONES 2975
6  BLAKE 2850
7  CLARK 2450
2  ALLEN 1600
10 TURNER 1500
14 MILLER 1300
3   WARD 1250
5  MARTIN 1250
11 ADAMS 1100
12 JAMES  950
1  SMITH  800
```

문제 22. 이름과 입사일을 출력하는데 먼저 입사한 사원부터 출력 하시오.

```
> emp[order(emp$hiredate, decreasing = F), c("ename","hiredate")]
```



```
> emp[order(emp$hiredate, decreasing = F), c("ename","hiredate")]
  ename hiredate
1  SMITH 1980-12-17
2  ALLEN 1981-02-20
3   WARD 1981-02-22
4  JONES 1981-04-02
6  BLAKE 1981-05-01
7  CLARK 1981-06-09
10 TURNER 1981-09-08
5  MARTIN 1981-09-28
9   KING 1981-11-17
12 JAMES 1981-12-03
13  FORD 1981-12-03
14 MILLER 1982-01-23
15  JACK 1982-01-23
8  SCOTT 1987-04-19
11 ADAMS 1987-05-23
```

문제 23. 직업이 SALESMAN인 직원들의 이름, 월급을 출력 하는데 월급이 높은 직원부터 출력 하시오.

```
emp[emp$job == 'SALESMAN' & order(emp$sal, decreasing = T), c("ename","sal","job")]
```

```
> emp[emp$job == 'SALESMAN' & order(emp$sal, decreasing = T), c("ename","sal","job")]
  ename sal      job
2  ALLEN 1600 SALESMAN
3   WARD 1250 SALESMAN
5  MARTIN 1250 SALESMAN
10 TURNER 1500 SALESMAN
```

문제 24. 직업이 SALESMAN인 직원들의 이름, 월급을 출력 하는데 월급이 높은 직원부터 출력 하시오.

```
emp[emp$job == "SALESMAN" & order(emp$sal, decreasing = T), c("ename","sal","job")]
```

```
> emp[emp$job == "SALESMAN" & order(emp$sal, decreasing = T), c("ename","sal","job")]
  ename sal      job
2  ALLEN 1600 SALESMAN
3   WARD 1250 SALESMAN
5  MARTIN 1250 SALESMAN
10 TURNER 1500 SALESMAN
```

*변수에 담아서 정렬하는 방법도 있다.

```
> emp[emp$job == "SALESMAN", c("ename","sal","job")]
  ename sal      job
2  ALLEN 1600 SALESMAN
3   WARD 1250 SALESMAN
5  MARTIN 1250 SALESMAN
10 TURNER 1500 SALESMAN
> x<-emp[emp$job == "SALESMAN", c("ename","sal","job")]
> order(x,decreasing = T)
[1] 2 4 9 10 11 12 3 1 5 8 6 7
> x[order(x$sal,decreasing = T),c("ename","sal")]
  ename sal
2  ALLEN 1600
10 TURNER 1500
3   WARD 1250
5  MARTIN 1250
> x
  ename sal      job
2  ALLEN 1600 SALESMAN
3   WARD 1250 SALESMAN
5  MARTIN 1250 SALESMAN
10 TURNER 1500 SALESMAN
```

*변수 보기 및 삭제

```
> ls()
[1] "emp" "i"   "x"   "x"
> rm(x)
> ls()
[1] "emp" "i"   "x"
>
```

문제 25. 직업이 ANALYST가 아닌 직원들의 이름,월급, 직업을 출력하는데 월급이 높은 순서대로 출력 하시오.

```
> emp[emp$job != 'ANALYST' , c("ename","sal","job")]
  ename  sal      job
1 SMITH  800    CLERK
2 ALLEN 1600  SALESMAN
3 WARD  1250  SALESMAN
4 JONES 2975  MANAGER
5 MARTIN 1250 SALESMAN
6 BLAKE 2850  MANAGER
7 CLARK 2450  MANAGER
9 KING  5000 PRESIDENT
10 TURNER 1500 SALESMAN
11 ADAMS 1100    CLERK
12 JAMES  950    CLERK
14 MILLER 1300    CLERK
15 JACK 3200    CLERK
> x<-emp[emp$job != 'ANALYST' , c("ename","sal","job")]
> x
  ename  sal      job
1 SMITH  800    CLERK
2 ALLEN 1600  SALESMAN
3 WARD  1250  SALESMAN
4 JONES 2975  MANAGER
5 MARTIN 1250 SALESMAN
6 BLAKE 2850  MANAGER
7 CLARK 2450  MANAGER
9 KING  5000 PRESIDENT
10 TURNER 1500 SALESMAN
11 ADAMS 1100    CLERK
12 JAMES  950    CLERK
14 MILLER 1300    CLERK
15 JACK 3200    CLERK
> x[order(x$sal, decreasing = T),c("ename","sal","job")]
  ename  sal      job
9 KING  5000 PRESIDENT
15 JACK 3200    CLERK
4 JONES 2975  MANAGER
6 BLAKE 2850  MANAGER
7 CLARK 2450  MANAGER
2 ALLEN 1600  SALESMAN
10 TURNER 1500 SALESMAN
14 MILLER 1300    CLERK
3 WARD  1250  SALESMAN
5 MARTIN 1250 SALESMAN
11 ADAMS 1100    CLERK
12 JAMES  950    CLERK
1 SMITH  800    CLERK
```

문제 26. 문제 24번을 doBy 패키지의 orderBy 함수를 이용해서 출력 하시오.

```
orderBy(~sal, emp[emp$job != 'ANALYST', c("ename","sal","job")])
orderBy(~sal, emp[emp$job != 'ANALYST', c("ename","sal","job")]) # 컬럼명 앞에 - 가 있으면 내림차순
```

```

> orderBy(~sal, emp[emp$job != 'ANALYST', c("ename","sal","job")])
  ename  sal    job
1  SMITH  800    CLERK
12 JAMES  950    CLERK
11 ADAMS 1100    CLERK
3   WARD 1250  SALESMAN
5  MARTIN 1250  SALESMAN
14 MILLER 1300    CLERK
10 TURNER 1500  SALESMAN
2   ALLEN 1600  SALESMAN
7   CLARK 2450  MANAGER
6   BLAKE 2850  MANAGER
4   JONES 2975  MANAGER
15  JACK 3200    CLERK
9   KING 5000  PRESIDENT
> orderBy(~-sal, emp[emp$job != 'ANALYST', c("ename","sal","job")])
  ename  sal    job
9   KING 5000  PRESIDENT
15  JACK 3200    CLERK
4   JONES 2975  MANAGER
6   BLAKE 2850  MANAGER
7   CLARK 2450  MANAGER
2   ALLEN 1600  SALESMAN
10 TURNER 1500  SALESMAN
14 MILLER 1300    CLERK
3   WARD 1250  SALESMAN
5  MARTIN 1250  SALESMAN
11 ADAMS 1100    CLERK
12 JAMES  950    CLERK
1   SMITH  800    CLERK

```

문제 27.	Crime_loc.csv를 내려받고 R로 로드한 후에 살인이 일어나는 장소와 건수를 살인의 건수가 높은 것부터 출력 하시오.
---------------	---

```

> crime_loc<-read.csv("c:WWdataWWcrime_loc.csv", header = T)
> head(orderBy(~-건수, crime_loc[crime_loc$범죄 == '살인',c("장소","건수")]),10)

# head(결과, 몇 개만 출력할지?)

```

```

> head(orderBy(~-건수, crime_loc[crime_loc$범죄 == '살인',c("장소","건수")]),10)
  장소  건수
83   집   312
85  노상  280
82 아파트 242
108 기타  131
89 병원   87
88 숙박업소 43
90 사무실  40
86 상점   23
101 의료기관 19
91 공장   15

```

10. 문자함수

2018년 5월 9일 수요일 오후 2:08

■ R 함수의 종류

1. 문자함수
2. 숫자함수
3. 날짜함수
4. 변환함수
5. 일반함수

1 문자함수의 종류

SQL	R	설명 및 예시																
upper	toupper	# 문자열을 대문자로 변환 toupper('문자열')																
lower	tolower	# 문자열을 소문자로 변환 tolower('문자열')																
substr	substr(x, start, stop)	#문자형 벡터 x의 start에서 부터 stop 까지만 잘라오기 (부분 선택) substr('문자열', 시작위치, 끝 위치) Ex) v1<-'SCOTT' R : substr(v1, 2, 2) # C 출력 Oracle : substr(v1, 2, 2) # CO 출력																
	sub(old, new, x)	#문자형 벡터 x에서 처음 나오는 old 문자를 new 문자로 한번만 바꾸기 sub('S', '@', emp\$ename) "KING" "BLAKE" "CLARK" "JONE@" "MARTIN" "ALLEN" "TURNER" "JAME@" "WARD" "FORD" "@MITH" "@COTT" "ADAM@" "MILLER"																
replace	gsub(old, new, x)	#문자형 벡터 x 내에 모든 old 문자를 new 문자로 모두 바꾸기 gsub('[0-2]', '*', emp\$sal) # emp\$sal의 0~2값을 *로 치환																
cocat	paste	paste(emp\$ename, '의 직업은', emp\$job) #연결연산자 역할																
To_char	format()	format(as.Date(emp\$hiredate), '%A') # 날짜형으로 변환한 데이터를 %A 형식의 문자형 타입으로 변환하겠다. <table><tr><td>%A</td><td>요일</td><td>%Y</td><td>년도 4자리</td></tr><tr><td>%y</td><td>년도 2자리</td><td>%m</td><td>달</td></tr><tr><td>%d</td><td>일</td><td>%H</td><td>시간</td></tr><tr><td>%M</td><td>분</td><td>%S</td><td>초</td></tr></table>	%A	요일	%Y	년도 4자리	%y	년도 2자리	%m	달	%d	일	%H	시간	%M	분	%S	초
%A	요일	%Y	년도 4자리															
%y	년도 2자리	%m	달															
%d	일	%H	시간															
%M	분	%S	초															
like	grep	emp[grep('COT', emp\$ename),] #SCOTT이 있는 행 출력																
length	nchar(x)	# 문자열의 글자수를 출력																

		x <- c("Seoul", "New York", "London", "1234") nchar(x) # 5 8 6 4
	regexpr()	#text 내에서 패턴이 가장 먼저 나오는 위치 찾기
	gregexpr()	#text 내에서 패턴이 나오는 모든 위치를 찾기

2 예제

문제 27. 이름과 직업을 출력하는데 소문자로 출력 하시오.

```
data.table(이름 = tolower(emp$ename), 직업=tolower(emp$job))
```

```
> data.table(이름 = tolower(emp$ename), 직업=tolower(emp$job))
   이름      직업
1: smith    clerk
2: allen  salesman
3:  ward  salesman
4: jones   manager
5: martin salesman
6: blake   manager
7: clark   manager
8: scott   analyst
9:  king president
10: turner salesman
11: adams   clerk
12: james   clerk
13:  ford   analyst
14: miller   clerk
15:  jack   clerk
>
```

문제 28. 이름이 scott인 사원의 이름과 월급을 조회하는데 scott을 소문자로 조합해도 조회되게 코드를 구현 하시오.

```
emp[emp$ename == toupper('scott') , c("ename","sal")]
```

```
> emp[emp$ename == toupper('scott') , c("ename","sal")]
   ename sal
8 SCOTT 3000
```

문제 29. 이름의 두번째 철자가 M인 사원들의 이름, 월급을 출력하는데 SUBSTR 함수를 이용해서 출력 하시오.

```
emp[substr(emp$ename,2,2) == 'M', c("ename","sal") ]
```

R은 오라클과 달리 substr(2,2) 면 2번째 자리부터 2번째 자리까지

```
> emp[substr(emp$ename,2,1) == 'M', c("ename","sal") ]
[1] ename sal
<0 행> <또는 row.names의 길이가 0입니다>
> emp[substr(emp$ename,2,2) == 'M', c("ename","sal") ]
   ename sal
1 SMITH 800
```

문제 30. 이름을 출력하고 그 옆에 이름의 첫번째 철자부터 세번째 철자까지 출력 하시오.

```
data.table(emp$ename, substr(emp$ename,1,3))
```

```
> data.table(emp$ename, substr(emp$ename,1,3))
      V1 V2
1: SMITH SMI
2: ALLEN ALL
3:  WARD WAR
4: JONES JON
5: MARTIN MAR
6: BLAKE BLA
7: CLARK CLA
8: SCOTT SCO
9:  KING KIN
10: TURNER TUR
11: ADAMS ADA
12: JAMES JAM
13:  FORD FOR
14: MILLER MIL
15:  JACK JAC
```

문제 31. 우리반 테이블을 R로 로드하고 이름과 성씨를 출력 하시오.

```
data.table(성=substr(emp2$ename,1,1), 이름=substr(emp2$ename,2,3))
```

```
> data.table(성=substr(emp2$ename,1,1), 이름=substr(emp2$ename,2,3))
      성 이름
1: 윤 지민
2: 송 영호
3: 은 해찬
4: 김 영도
5: 정 호진
6: 김 지우
7: 정 인중
8: 이 규진
9: 백 광훈
10: 김 원선
11: 김 광훈
12: 장 은희
13: 지 은철
14: 이 상민
15: 윤 동환
16: 이 한새
17: 방 승준
18: 신 영근
19: 김 근마
20: 유 혜린
21: 김 건태
22: 신 현수
23: 이 근호
24: 김 대경
25: 김 동원
26: 노 영
27: 이 찬경
28: 차 호성
29: 한 지원
30: 이 광훈
```

문제 32. 우리반에 성씨가 무엇이 있는지 중복제거해서 출력 하시오.

```
data.table(unique(substr(emp2$ename,1,1)))
> data.table(unique(substr(emp2$ename,1,1)))
   v1
1: 윤
2: 송
3: 은
4: 김
5: 정
6: 이
7: 백
8: 장
9: 지
10: 방
11: 신
12: 유
13: 도
14: 차
15: 한
```

문제 33. 이름, 나이, 통신사를 출력 하시오.

```
emp2[,c("ename","age","telecom")]
> emp2[,c("ename","age","telecom")]
   ename age telecom
1  윤진민  27      sk
2  송윤호  27      kt
3  은해찬  29      lg
4  김영토  28      kt
5  정호진  32      lg
6  김지우  27      kt
7  정민중  27      lg
8  이유헌  25      sk
9  백광흠  26      kt
10 김원섭  26      sk
11 김광록  27      lg
12 장은희  24      kt
13 지윤철  28      kt
14 이상민  26      lg
15 윤동환  25      sk
16 이한새  25      kt
17 방승준  26      kt
18 신영근  25      kt
19 김근마  24      kt
20 유혜린  26      lg
21 김건태  28      sk
22 신현수  27      kt
23 이근호  28      cjh
24 김대경  27      sk
25 김동운  28      sk
26   도웅  28      lg
27 이찬중  28      lg
28 차호성  28      kt
29 한지운  24      sk
30 이광훈  25      lg
```

문제 34. 전공이 경제학과인 학생들의 이름, 나이, 주소를 출력 하시오.

```
emp2[grepl('*경제학과*',emp2$major), c("ename","age","address")]
```

```
> emp2[grepl('*경제학과*',emp2$major), c("ename","age","address")]
  ename age address
1 윤진민 27 경기도 구리시 교문동
22 신현수 27 서울시 관악구 삼성동
28 차호성 28 서울시 관악구 신림동
```

문제 35. 이름, 월급을 출력하는데 월급을 출력할 때에 숫자 0을 *로 출력 하시오.

Gsub : ex) gsub('h','H',text) -- text에서 소문자 h를 대문자 H로 치환한다.

```
data.table(emp$ename, gsub('0','*',emp$sal))
> data.table(emp$ename, gsub('0','*',emp$sal))
   v1    v2
1: SMITH 8**
2: ALLEN 16**
3:  WARD 125*
4: JONES 2975
5: MARTIN 125*
6: BLAKE 285*
7: CLARK 245*
8: SCOTT 3***
9:  KING 5***
10: TURNER 15**
11: ADAMS 11**
12: JAMES  95*
13:  FORD 3***
14: MILLER 13**
15:  JACK 32**
```

문제 36. 이름, 월급을 출력하는데 월급을 출력하라 때에 숫자 0을 *로 출력 하시오.

```
data.table(emp$ename, gsub('[0-2]','*',emp$sal))
> data.table(emp$ename, gsub('[0-2]','*',emp$sal))
   v1    v2
1: SMITH 8**
2: ALLEN *6**
3:  WARD **5*
4: JONES *975
5: MARTIN **5*
6: BLAKE *85*
7: CLARK *45*
8: SCOTT 3***
9:  KING 5***
10: TURNER *5**
11: ADAMS *****
12: JAMES  95*
13:  FORD 3***
14: MILLER *3**
15:  JACK 3***
```

문제 36_2. Paste 함수를 이용해서 이름, 직업을 아래와 같이 출력 하시오.

*paste 함수는 오라클의 연결연산자와 비슷한 기능을 하는 함수

```
data.table(paste(emp$ename, '의 직업은 ',emp$job))
```

```
> data.table(paste(emp$ename, '의 직업은 ',emp$job))
      v1
1: SMITH 의 직업은 CLERK
2: ALLEN 의 직업은 SALESMAN
3: WARD 의 직업은 SALESMAN
4: JONES 의 직업은 MANAGER
5: MARTIN 의 직업은 SALESMAN
6: BLAKE 의 직업은 MANAGER
7: CLARK 의 직업은 MANAGER
8: SCOTT 의 직업은 ANALYST
9: KING 의 직업은 PRESIDENT
10: TURNER 의 직업은 SALESMAN
11: ADAMS 의 직업은 CLERK
12: JAMES 의 직업은 CLERK
13: FORD 의 직업은 ANALYST
14: MILLER 의 직업은 CLERK
15: JACK 의 직업은 CLERK
```

문제 36_3. 이름, 연봉을 아래와 같이 출력 하시오.

SCOTT의 연봉을 3600입니다.

```
x<-emp[emp$ename == 'SCOTT', c("ename","sal")]
```

```
x
```

```
data.table(paste(x$ename , '의 연봉은 ',x$sal*12,'입니다.'))
```

```
> x<-emp[emp$ename == 'SCOTT', c("ename","sal")]
> x
  ename  sal
8 SCOTT 3000
> data.table(paste(x$ename , '의 연봉은 ',x$sal*12,'입니다.'))
      v1
1: SCOTT 의 연봉은 36000 입니다.
```

11. 숫자함수

2018년 5월 9일 수요일 오후 3:33

1. 숫자함수 종류

SQL	R	설명	예시
Round	round	반올림	round(123.5) # 124
trunc	trunc		trunc(3678.78, -1) # 3678
mod	%%		10%%3 # 결과 1
power	^ (ex : 2^3)		3^3 # 결과 27
	ceiling(x)	크거나 같은 정수 (지붕정수)	ceiling(5.88) # 6
	floor(x)	작거나 같은 정수 (바닥정수)	floor(5.88) # 5
	sqrt(x)	X의 제곱근 구하기	sqrt(16) # 4 sqrt(-16) # 허수는 NaN
	abs(x)	X의 절대 값 구하기	abs(-10) #10
	exp(x)	X를 지수 변환	exp(log(10, base=exp(1))) # 10 exp(1) # 2.718282
	factorial(x)	X의 팩토리얼 값 출력	factorial(5) # 120
log(x,n)	log(x, base=n)	X를 밑이 n인 log 취하기	log(10, base=2) # 3.321928

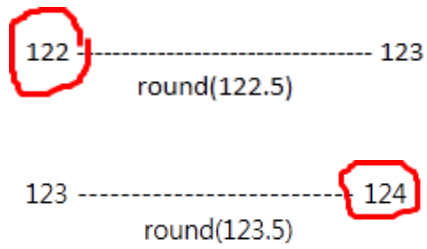
■ rep() ,seq()

Rep()	Seq()
일정한 데이터를 반복할 때 사용	일정한 구조/순차 데이터를 생성할 때 사용
<pre>var <- rep(1:3) # 1, 2, 3 값이 들어감 var <- rep(1:3, 2) # 1,2,3 값이 2번 들어감 var <- rep(1:3, each=2) #1~3값이 2번씩 들어감 (1,1,2,2,3,3) var <- rep(c("a", 1), c(5,10)) # "a"를 먼저 5번 반복하고, "1"을 10번 반복</pre>	<pre>var <- seq(1,5) # 1, 2, 3, 4, 5 값이 들어감 var <- seq(2,-2) # 2, 1, 0, -1, -2 값이 들어감 var <- seq(1,10,2) # 1, 3, 5, 7, 9 값이 들어감 (2씩증가)</pre>

■ R의 round 함수의 특징

R은 짝수를 좋아한다.

 122.5 ----- 123



■ round와 trunc 함수 설명

Ex) 3 0 0 0 . 7 8
~~4~~ ~~3~~ ~~2~~ ~~1~~ ~~0~~ ~~1~~

```
> round(3000.78, 1)
[1] 3000.8
```

```
> round(3678.78, -1)
[1] 3680
```

```
> trunc(3678.78, 1)
[1] 3678
```

```
> trunc(3678.78, -1)
[1] 3678
```

*trunc는 소수점 이하만 가능하다.

2. 예제

문제 37. 6의 9승을 출력 하시오.

```
> 6^9
[1] 10077696
```

문제 38. 10을 3으로 나눈 나머지 값이 무엇인가

```
> 10%%3
[1] 1
```

문제 39. 이름과 연봉을 출력 하는데 연봉이 월급의 12를 곱해서 출력하고 컬럼명을 "이름", "연봉"으로 출력 하시오.

```
data.table(이름=emp$ename, 연봉=emp$sal*12)
```

```
> data.table(이름=emp$ename, 연봉=emp$sal*12)
```

	이름	연봉
1:	SMITH	9600
2:	ALLEN	19200
3:	WARD	15000
4:	JONES	35700
5:	MARTIN	15000
6:	BLAKE	34200
7:	CLARK	29400
8:	SCOTT	36000
9:	KING	60000
10:	TURNER	18000
11:	ADAMS	13200
12:	JAMES	11400
13:	FORD	36000
14:	MILLER	15600
15:	JACK	38400

12. 날짜함수

2018년 5월 9일 수요일 오후 4:20

1. 날짜함수 종류

오라클	R	예시
sysdate	Sys.Date()	Sys.Date() -- "2018-05-09"
add_month	difftime	
Months_between	사용자 정의 함수	
last_day	사용자 정의 함수	
next_day	사용자 정의 함수	
To_Date	as.Date()	format(as.Date(emp\$hiredate),'%A')

날짜관련 패키지 함수

```
> install.packages("lubridate")
```

```
> library(lubridate)
```

```
* floor_date(Sys.Date(),"month") -- 날짜의 첫번째 날짜를 반환
```

```
* ceiling_date(Sys.Date(),"month") -- 날짜의 다음달 첫번째 날짜를 반환
```

2. 예제

문제 40. 오늘의 날짜를 출력 하시오.

```
Sys.Date()
```

```
> Sys.Date()  
[1] "2018-05-09"
```

문제 41. 이름, 입사한 날짜부터 오늘까지 총 몇일 근무 했는지 출력 하시오.

```
> data.table(emp$ename ,Sys.Date()-emp$hiredate)  
Error in as.character.factor(x) : malformed factor  
In addition: warning message:  
In data.table(emp$ename, Sys.Date() - emp$hiredate) :  
  Incompatible methods ("-.Date", "Ops.factor") for "-"
```

*날짜데이터 - 문자데이터(Factor) 라서 오류가 발생 -----> 형변환 해주어야 한다. // **as.Date()**

```
> str(emp$hiredate)
Factor w/ 13 levels "1980-12-17","1981-02-20",...: 1 2 3 4 8 5 6 12 9 7 ...
>
> str(Sys.Date())
Date[1:1], format: "2018-05-09"
```

```
data.table(emp$ename ,Sys.Date()- as.Date(emp$hiredate))
```

```
> data.table(emp$ename ,Sys.Date()- as.Date(emp$hiredate))
      v1      v2
1: SMITH 13657 days
2: ALLEN 13592 days
3:  WARD 13590 days
4: JONES 13551 days
5: MARTIN 13372 days
6: BLAKE 13522 days
7: CLARK 13483 days
8: SCOTT 11343 days
9:  KING 13322 days
10: TURNER 13392 days
11: ADAMS 11309 days
12: JAMES 13306 days
13:  FORD 13306 days
14: MILLER 13255 days
15:  JACK 13255 days
```

문제 42. 이름, 입사한 날짜부터 오늘까지 총 몇 달 근무 했는지 출력 하시오.

----> 함수를 만들어서 사용해야함

문제 43. 오늘 날짜가 속한 달의 마지막 날짜를 출력 하시오.

SQL : select last_day(sysdate) from dual;

R : install.packages("lubridate")
library(lubridate)

```
install.packages("lubridate")
library(lubridate)
```

```
> floor_date(Sys.Date(),"month")
[1] "2018-05-01"
> ceiling_date(Sys.Date(),"month")
[1] "2018-06-01"
> ceiling_date(Sys.Date(),"month")-1
[1] "2018-05-31"
```

* floor_date(Sys.Date(),"month") -- 날짜의 첫번째 날짜를 반환
* ceiling_date(Sys.Date(),"month") -- 날짜의 다음달 첫번째 날짜를 반환

문제 44. last_day라는 함수를 생성 하시오.

SQL : select last_day(sysdate) from dual;

```
R : last_day <- function(x) {  
  ceiling_date(x,"month") - 1  
}  
last_day(Sys.Date())
```

```
> last_day <- function(x){ ceiling_date(x, "month") -1 }  
> last_day(Sys.Date())
```

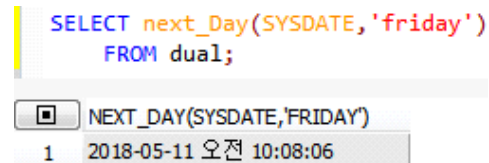
```
> last_day <- function(x){ ceiling_date(x, "month") -1 }  
> last_day(Sys.Date())  
[1] "2018-05-31"
```

문제 45. first_day라는 함수를 생성 하시오.

```
first_day<-function(x){ floor_date(x,"month") }  
first_day(Sys.Date())
```

```
> first_day(Sys.Date())  
[1] "2018-05-01"
```

문제 46. Next_day 함수를 생성 하시오.



The screenshot shows a SQL query in a text editor: `SELECT next_Day(SYSDATE,'friday') FROM dual;`. Below the query, there is a table with one row and one column. The column header is `NEXT_DAY(SYSDATE,'FRIDAY')` and the value in the row is `2018-05-11 오전 10:08:06`.

```
next_day <- function(x,day){  
  for (i in 1:7) {  
    check_date = x+i  
  
    if(format(check_date,'%A')==day){  
      print(check_date)  
    }  
  }  
}
```

```
next_day(Sys.Date(),'금요일')
```

```
> next_day(Sys.Date(), '금요일')  
[1] "2018-05-11"
```

문제 47. 이름, 입사일, 입사한 요일을 출력 하시오.

```
data.table(emp$ename, emp$hiredate, format(as.Date(emp$hiredate), '%A'))
```

```
> data.table(emp$ename, emp$hiredate, format(as.Date(emp$hiredate), '%A'))
      V1      V2      V3
1: SMITH 1980-12-17 수요일
2: ALLEN 1981-02-20 금요일
3:  WARD 1981-02-22 일요일
4: JONES 1981-04-02 목요일
5: MARTIN 1981-09-28 월요일
6: BLAKE 1981-05-01 금요일
7: CLARK 1981-06-09 화요일
8: SCOTT 1987-04-19 일요일
9:  KING 1981-11-17 화요일
10: TURNER 1981-09-08 화요일
11: ADAMS 1987-05-23 토요일
12: JAMES 1981-12-03 목요일
13:  FORD 1981-12-03 목요일
14: MILLER 1982-01-23 토요일
15:  JACK 1982-01-23 토요일
```

문제 48. 내일이 무슨 요일인지 출력 하시오.

```
format(Sys.Date()+1, '%A')
```

```
> Sys.Date()
[1] "2018-05-10"
> Sys.Date()+1
[1] "2018-05-11"
> format(Sys.Date()+1, '%A')
[1] "금요일"
```

문제 49. 오라클의 add_months 함수를 R에서 생성하기 위해 다음 식을 수행 하시오.

```
SELECT ADD_MONTHS(SYSDATE,100)
FROM dual;
```

	ADD_MONTHS(SYSDATE,100)
1	2026-09-10 오전 11:03:52

```
Sys.Date() + months(100)
```

```
> Sys.Date() + months(100)
[1] "2026-09-10"
```

문제 50. 아래와 같이 add_months 함수를 실행하면 100달 뒤의 날짜가 출력되게 함수를 생성 하시오.

```
add_month<-function(x,day){
  check_date = x + months(day)
  print(check_date)
}
```



```
add_month(Sys.Date(),100)
```

```
> add_month<-function(x,day){
+   check_date = x + months(day)
+   print(check_date)
+ }
> add_month(Sys.Date(),100)
[1] "2026-09-10"
```

문제 51. 아래와 같이 add_months 함수를 실행하면 100달 뒤의 날짜가 출력되게 함수를 생성 하시오.

```
SELECT ename, ROUND(months_between(SYSDATE,hiredate))
FROM EMP;
```

	ENAME	ROUND(MONTHS_BETWEEN(SYSDATE,HIREDATE))
1	KING	438
2	BLAKE	444
3	CLARK	444
4	JONES	445
5	MARTIN	440
6	ALLEN	447
7	TURNER	441
8	JAMES	437

-- 오라클에서 이름, 입사한 날짜부터 오늘까지 총 몇 달 근무 했는지 출력

R에서 아래와 같이 수행하면 달수가 출력되는 함수를 생성 하시오.

```
Months_between(Sys.Date(),emp$hiredate)
```

```
months_between<-function(x,y){
  day1 = as.Date(x)
  day2 = as.Date(y)
  (year(day1)-year(day2))*12 + (month(day1)-month(day2))
}
```

```
data.table(emp$ename,months_between(Sys.Date(),emp$hiredate))
```

```
> data.table(emp$ename,months_between(Sys.Date(),emp$hiredate))
      V1  V2
1: SMITH 449
2: ALLEN 447
3:  WARD 447
4: JONES 445
5: MARTIN 440
6: BLAKE 444
7: CLARK 443
8: SCOTT 373
9:  KING 438
10: TURNER 440
11: ADAMS 372
12: JAMES 437
13:  FORD 437
14: MILLER 436
15:  JACK 436
```

문제 52. 오라클 db의 emp 테이블의 data를 csv로 내려서 R에 로드해서 오라클과 R의 emp 테이블의 내용을 일치 시키시오.

```
> emp<-read.csv("c:\\data\\emp.csv", header= T)
> emp
```

	empno	ename	job	mgr	hiredate	sal	comm	deptno
1	7839	KING	PRESIDENT	NA	1981-11-17	5000	NA	10
2	7698	BLAKE	MANAGER	7839	1981-05-01	2850	NA	30
3	7782	CLARK	MANAGER	7839	1981-05-09	2450	NA	10
4	7566	JONES	MANAGER	7839	1981-04-01	2975	NA	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11	950	NA	30
9	7521	WARD	SALESMAN	7698	1981-02-23	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11	3000	NA	20
11	7369	SMITH	CLERK	7902	1980-12-09	800	NA	20
12	7788	SCOTT	ANALYST	7566	1982-12-22	3000	NA	20
13	7876	ADAMS	CLERK	7788	1983-01-15	1100	NA	20
14	7934	MILLER	CLERK	7782	1982-01-11	1300	NA	10

13. 일반함수

2018년 5월 10일 목요일 오전 9:45

1. 일반함수 종류

오라클	R	예시
nvl	is.na	is.na(값) # 값이 NA일 경우 T를 반환
decode	ifelse	ifelse(조건, 조건이 T일경우 실행문, 조건이 F일경우 실행문)
case	ifelse	ifelse(조건, 조건이 T일경우 실행문, 조건이 F일경우 실행문)

2. 예제

ifelse 함수 예제	이름, 월급, 등급을 출력하는데 월급이 1500 이상이면 등급을 A로 출력하고 아니면 B로 출력 하시오.
---------------------	--

```
data.table(emp$ename, emp$sal, ifelse(emp$sal>=1500, 'A','B'))
```

```
> data.table(emp$ename, emp$sal, ifelse(emp$sal>=1500, 'A','B'))
      V1    V2 V3
1:  KING 5000  A
2: BLAKE 2850  A
3: CLARK 2450  A
4: JONES 2975  A
5: MARTIN 1250  B
6: ALLEN 1600  A
7: TURNER 1500  A
8: JAMES  950  B
9:  WARD 1250  B
10:  FORD 3000  A
11: SMITH  800  B
12: SCOTT 3000  A
13: ADAMS 1100  B
14: MILLER 1300  B
```

문제 52.	이름, 월급, 등급을 출력하는데 월급이 3000 이상이면 A를 출력하고 월급이 1500 이상이고 3000보다 작으면 B를 출력하고 나머지 직원들은 C를 출력 하시오.
---------------	--

```
data.table(emp$ename, emp$sal, ifelse(emp$sal>=3000,'A',ifelse(emp$sal>=1500&emp$sal<3000,'B','c')))
```

```
> data.table(emp$ename, emp$sal, ifelse(emp$sal>=3000,'A',ifelse(emp$sal>=1500&emp$sal<3000,'B','c'))
      V1    V2 V3
1:  KING 5000  A
2: BLAKE 2850  B
3: CLARK 2450  B
4: JONES 2975  B
5: MARTIN 1250  C
6: ALLEN 1600  B
7: TURNER 1500  B
8: JAMES  950  C
9:  WARD 1250  C
10:  FORD 3000  A
11: SMITH  800  C
12: SCOTT 3000  A
13: ADAMS 1100  C
14: MILLER 1300  C
```

```
> data.table(emp$ename, emp$sal, ifelse(emp$sal>=3000,'A',ifelse(emp$sal>=1500&emp$sal<3000,'B','c'))))
      V1  V2 V3
1:  KING 5000  A
2:  BLAKE 2850  B
3:  CLARK 2450  B
4:  JONES 2975  B
5: MARTIN 1250  C
6:  ALLEN 1600  B
7:  TURNER 1500  B
8:  JAMES  950  C
9:   WARD 1250  C
10:   FORD 3000  A
11: SMITH  800  C
12: SCOTT 3000  A
13: ADAMS 1100  C
14: MILLER 1300  C
```

문제 53. 이름, 입사일, 보너스를 출력하는데 1980년도에 입사 했으면 보너스를 A로 출력하고 1981년도에 입사했으면 보너스를 B로 출력하고 1982년도에 입사했으면 보너스를 C로 출력하고 나머지 년도는 D로 출력되게 하시오.

```
data.table(이름=emp$ename, 입사일=emp$hiredate, 보너스=ifelse(substr(emp$hire,1,4)=='1980','A',
  ifelse(substr(emp$hiredate,1,4)=='1981','B',ifelse(substr(emp$hiredate,1,4)=='1982','C','D'))))
```

```
> data.table(이름=emp$ename, 입사일=emp$hiredate, 보너스=ifelse(substr(emp$hire,1,4)=='1980','A',
+   ifelse(substr(emp$hiredate,1,4)=='1981','B',ifelse(substr(emp$hiredate,1,4)=='1982','c','d'))))
      이름  입사일 보너스
1:  KING 1981-11-17      B
2:  BLAKE 1981-05-01      B
3:  CLARK 1981-05-09      B
4:  JONES 1981-04-01      B
5: MARTIN 1981-09-10      B
6:  ALLEN 1981-02-11      B
7:  TURNER 1981-08-21      B
8:  JAMES 1981-12-11      B
9:   WARD 1981-02-23      B
10:   FORD 1981-12-11      B
11: SMITH 1980-12-09      A
12: SCOTT 1982-12-22      C
13: ADAMS 1983-01-15      D
14: MILLER 1982-01-11      C
```

문제 54. 이름과 커미션을 출력하는데 is.na 함수를 이용해서 커미션이 NA인 직원들을 출력 하시오.

```
emp[is.na(emp$comm) , c("ename","comm")]
```

```
> emp[is.na(emp$comm) , c("ename","comm")]
      ename comm
1:  KING    NA
2:  BLAKE    NA
3:  CLARK    NA
4:  JONES    NA
8:  JAMES    NA
10:  FORD    NA
11: SMITH    NA
12: SCOTT    NA
13: ADAMS    NA
14: MILLER    NA
```

문제 55. 이름과 커미션을 출력하는데 커미션이 NA인 직원들은 no comm이란 글씨로 출력되게 하시오.

```
data.table(emp$ename, emp$comm, ifelse(is.na(emp$comm),'no comm',emp$comm ))
```

```
> data.table(emp$ename, emp$comm, ifelse(is.na(emp$comm),'no comm',emp$comm ))
```

	V1	V2	V3
1:	KING	NA	no comm
2:	BLAKE	NA	no comm
3:	CLARK	NA	no comm
4:	JONES	NA	no comm
5:	MARTIN	1400	1400
6:	ALLEN	300	300
7:	TURNER	0	0
8:	JAMES	NA	no comm
9:	WARD	500	500
10:	FORD	NA	no comm
11:	SMITH	NA	no comm
12:	SCOTT	NA	no comm
13:	ADAMS	NA	no comm
14:	MILLER	NA	no comm

14. 변환함수/그룹함수

2018년 5월 10일 목요일 오후 2:32

1. 변환함수

" 데이터의 유형을 변경하는 함수 "

오라클	R	예시
To_char	as.character	
To_number	as.integer	
To_date	as.date as.factor format	

■ format 함수 설명

```
format(as.Date(emp$hiredate),'%A')
```

%Y	년도	%m	달
%d	일	%A	요일

예제. 내가 무슨 요일에 태어났는지 출력 하시오.

```
format(as.Date('1993-6-18'),'%A')
```

```
> format(as.Date('1993-6-18'),'%A')  
[1] "금요일"
```

2. 그룹함수

함수명	설명	사용법
aggregate	그룹함수를 세로로 출력한다	aggregate(출력할 값 ~ 그룹할 값 , 데이터 프레임, 그룹함수명)
tapply	그룹함수를 가로로 출력한다	tapply(출력할 값, 기준 값(열) or [list(~별(세로),~별(가로))] , 그룹함수명)

■ 그룹 함수

오라클	R	설명	예시
Max	max(x)	X값의 최대값 출력	max(emp[emp

			<code>\$job=='SALESMAN','sal']</code>)
min	<code>min(x)</code>	X값의 최소값 출력	<code>min(emp[emp\$deptno==30,'sal'])</code>
sum	<code>sum(x)</code>	X값들의 총합 출력	<code>sum(emp\$sal)</code>
avg	<code>mean(x)</code>	X값의 평균 출력	<code>mean(emp\$sal)</code>
count	<code>length(세로)</code> <code>table(가로)</code>	<code>length()</code> : x값의 개수 <code>table()</code> : x값 별 개수	<code>length(emp)</code> # 8 컬럼의 개수 반환 <code>length(emp\$sal)</code> # 14 sal컬럼의 로우 수
	median	벡터 x가 홀수개이면 정 가운데 값을 중앙값을 가져오지만, 위의 case와 같이 x가 짝수개 이면 정가운데의 양쪽 두개의 값을 가져다가 평균을 내서 중앙값을 계산	

`*apply(그룹함수에 쓸 컬럼, 기준 컬럼명, 그룹 함수) ---> 가로로 출력`

3.예제

문제 56.	직업이 SALESMAN인 직원들 중에 최대 월급을 출력 하시오.
--------	-------------------------------------

```
max(emp[emp$job=='SALESMAN','sal'])
```

```
> max(emp[emp$job=='SALESMAN','sal'])
[1] 1600
```

문제 57.	부서번호가 30번인 직원들 중에서 최소 월급을 출력 하시오.
--------	-----------------------------------

```
min(emp[emp$deptno==30,'sal'])
```

```
> min(emp[emp$deptno==30,'sal'])
[1] 950
```

문제 58.	직업, 직업별 최대 월급을 출력 하시오.
--------	------------------------

```
aggregate(sal~job,emp,max)
```

```
> aggregate(sal~job,emp,max)
  job  sal
1 ANALYST 3000
2  CLERK 1300
3  MANAGER 2975
4 PRESIDENT 5000
5 SALESMAN 1600
```

문제 59. 부서번호, 부서번호별 최소 월급을 출력 하시오.

```
aggregate(sal~deptno,emp,min)
```

```
> aggregate(sal~deptno,emp,min)
  deptno  sal
1     10 1300
2     20  800
3     30  950
```

문제 60. 부서번호, 직업, 부서번호별 직업별 토탈 월급을 출력 하시오.

```
aggregate(sal~(emp$job+emp$deptno),emp,sum)
```

```
> aggregate(sal~(emp$job+emp$deptno),emp,sum)
  emp$job emp$deptno  sal
1   CLERK         10 1300
2  MANAGER         10 2450
3 PRESIDENT         10 5000
4  ANALYST         20 6000
5   CLERK         20 1900
6  MANAGER         20 2975
7   CLERK         30  950
8  MANAGER         30 2850
9 SALESMAN         30 5600
```

문제 61. 부서번호, 부서번호별 최대 월급을 출력 하는데 부서번호별 최대 월급이 높은 것부터 출력 하시오.

```
library(dplyr)
order_by(~sal,aggregate(sal~deptno,emp,max))
```

```
> order_by(~sal,aggregate(sal~deptno,emp,max))
  deptno  sal
1     10 5000
2     20 3000
3     30 2850
```

문제 62. 직업, 직업별 인원수를 출력 하시오.

```
aggregate(empno~job,emp,length)
```

```
> aggregate(empno~job,emp,length)
  job empno
1 ANALYST    2
2  CLERK     4
3  MANAGER    3
4 PRESIDENT    1
5 SALESMAN    4
```


문제 63. 직업, 직업별 평균 월급을 출력 하시오.

```
aggregate(sal~job,emp,mean)
```

```
> aggregate(sal~job,emp,mean)
      job      sal
1 ANALYST 3000.000
2  CLERK 1037.500
3  MANAGER 2758.333
4 PRESIDENT 5000.000
5  SALESMAN 1400.000
```

문제 64. 입사한 년도(4자리), 입사한 년도별 토탈 월급을 출력 하시오.

```
aggregate(sal~substr(hiredate,1,4),emp,sum)
```

```
> aggregate(sal~substr(hiredate,1,4),emp,sum)
      substr(hiredate, 1, 4)      sal
1          1980           800
2          1981        22825
3          1982         4300
4          1983         1100
```

문제 65. 입사한 년도(4자리), 입사한 년도별 토탈 월급을 출력 하시오. (가로로 출력 하시오)

```
*tapply(그룹함수에 쓸 컬럼, 기준 컬럼명(열) , 그룹 함수)
```

```
tapply(emp$sal, year(emp$hiredate),sum)
```

```
> tapply(emp$sal, year(emp$hiredate),sum)
1980 1981 1982 1983
 800 22825 4300 1100
```

문제 66. 부서번호, 부서번호별 토탈 월급을 가로로 출력 하시오.

```
tapply (emp$sal, emp$deptno,sum)
```

```
> tapply (emp$sal, emp$deptno,sum)
 10    20    30
8750 10875 9400
```

문제 67. 직업, 직업별 인원수를 출력하는데 세로, 가로로 두 번 출력 하시오.

```
> tapply (emp$empno, emp$job, length)
```

```
> table(emp$job)
```

```
> tapply(emp$empno, emp$job, length)
ANALYST    CLERK    MANAGER  PRESIDENT  SALESMAN
      2         4         3         1         4

> table(emp$job)
ANALYST    CLERK    MANAGER  PRESIDENT  SALESMAN
      2         4         3         1         4
>
```

문제 68. 직업, 직업별 인원수를 출력하는데 세로, 가로로 두 번 출력 하시오.

```
> tapply(emp$sal, list(emp$deptno, emp$job), sum)

> tapply(emp$sal, list(emp$deptno, emp$job), sum)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10         NA  1300     2450       5000         NA
20      6000  1900     2975         NA         NA
30         NA   950     2850         NA      5600
```

문제 69. 아래의 결과에서 NA를 숫자 0으로 출력 하시오.

```
x<-tapply(emp$sal, list(emp$deptno, emp$job), sum)
x[is.na(x)]<-0      -- null 인곳에 0을 넣어라
x

> x<-tapply(emp$sal, list(emp$deptno, emp$job), sum)
> x[is.na(x)]<-0
> x
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10         0  1300     2450       5000         0
20      6000  1900     2975         0         0
30         0   950     2850         0      5600
```

문제 70. 직업, 입사한 년도(4자리), 직업별 입사한 년도별 토탈 월급을 출력 하시오.

```
x<-tapply(emp$sal, list(substr(emp$hiredate, 1, 4), emp$job), sum)
x[is.na(x)]<-0
x

> tapply(emp$sal, list(substr(emp$hiredate, 1, 4), emp$job), sum)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
1980         NA   800         NA         NA         NA
1981      3000   950     8275       5000      5600
1982      3000  1300         NA         NA         NA
1983         NA  1100         NA         NA         NA
>
>
> x<-tapply(emp$sal, list(substr(emp$hiredate, 1, 4), emp$job), sum)
> x[is.na(x)]<-0
> x
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
1980         0   800         0         0         0
1981      3000   950     8275       5000      5600
1982      3000  1300         0         0         0
```

```
> tapply(emp$sal, list(substr(emp$hiredate,1,4), emp$job), sum)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
1980         NA    800         NA         NA         NA
1981      3000    950      8275      5000      5600
1982      3000   1300         NA         NA         NA
1983         NA   1100         NA         NA         NA
>
>
> x<-tapply(emp$sal, list(substr(emp$hiredate,1,4), emp$job), sum)
> x[is.na(x)]<-0
> x
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
1980         0    800         0         0         0
1981      3000    950      8275      5000      5600
1982      3000   1300         0         0         0
1983         0   1100         0         0         0
```

문제 71. 나이, 통신사, 나이별 통신사별 인원수를 아래와 같은 형태로 출력 하시오.

```
x<-tapply(emp2$empno, list(emp2$age,emp2$telecom),length)
x[is.na(x)]<-0
x

> tapply(emp2$empno, list(emp2$age,emp2$telecom),length)
      cjh kt  lg sk
24   NA  2  NA  1
25   NA  2   1  2
26   NA  2   2  1
27   NA  3   2  2
28    1  3   2  2
29   NA NA   1 NA
32   NA NA   1 NA
> x<-tapply(emp2$empno, list(emp2$age,emp2$telecom),length)
> x[is.na(x)]<-0
> x
      cjh kt  lg sk
24     0  2   0  1
25     0  2   1  2
26     0  2   2  1
27     0  3   2  2
28     1  3   2  2
29     0  0   1  0
32     0  0   1  0
```

문제 72. 직업(세로), 입사한 년도 4자리(가로), 직업별 입사한 년도별 토탈 월급을 출력 하시오.

```
x<-tapply(emp$sal, list(emp$job,format(as.Date(emp$hiredate),'%Y')),sum)
x[is.na(x)]<-0
x

> x<-tapply(emp$sal, list(emp$job,format(as.Date(emp$hiredate),'%Y')),sum)
> x[is.na(x)]<-0
> x
      1980 1981 1982 1983
ANALYST     0 3000 3000     0
CLERK      800  950 1300 1100
MANAGER     0 8275     0     0
PRESIDENT   0 5000     0     0
SALESMAN    0 5600     0     0
```

문제 73. 아래의 결과에서 column name과 row name을 각각 출력 하시오.

	1980	1981	1982	1983
ANALYST	0	3000	3000	0
CLERK	800	950	1300	1100
MANAGER	0	8275	0	0
PRESIDENT	0	5000	0	0
SALESMAN	0	5600	0	0

colnames(x)

rownames(x)

```
> colnames(x)
[1] "1980" "1981" "1982" "1983"
> rownames(x)
[1] "ANALYST" "CLERK" "MANAGER" "PRESIDENT" "SALESMAN"
```

문제 74. Crime_loc.csv를 R로 로드하고 범죄 유형이 살인인 범죄의 장소와 건수를 출력 하시오.

```
crime_loc<-read.csv("c:\\data\\crime_loc.csv",header = T)
```

```
head(crime_loc[crime_loc$범죄 == '살인',c("장소","건수")],10)
```

```
> head(crime_loc[crime_loc$범죄 == '살인',c("장소","건수")],10)
      장소 건수
82 아파트 242
83 집 312
84 고속도로 1
85 노상 280
86 상점 23
87 시장노점 5
88 숙박업소 43
89 병원 87
90 사무실 40
91 공장 15
```

문제 75. 장소(세로), 범죄(가로), 장소별 범죄별 건수를 출력 하시오.

```
tapply(crime_loc$건수,list(crime_loc$장소,crime_loc$범죄),sum)
```

	간통	강간	강도	공갈	과실치사상	교통사고처리	도로교통법위반	도박과복표	방화	살인	상해	손괴	실화			
고속도로	0	18	1	3	2	489	211	1	0	1	75	32	2			
공사장	0	12	8	101	12	18	7	3	6	4	374	217	28			
공장	0	26	16	37	20	11	0	41	39	15	560	158	196			
공지	0	11	3	1	1	4	3	3	1	3	53	76	15			
교통	16	551	33	25	11	74	14	6	5	9	378	101	16			
구급장소	0	14	1	1	3	0	0	0	0	0	149	1	0			
금융기관	0	17	23	51	2	0	2	13	13	0	55	176	6			
기타	404	3184	552	898	208	1123	704	4300	298	131	9642	11821	423			
노상	17	3488	1541	2154	463	196218	42824	105	293	280	29219	22193	171			
병원	23	1362	275	382	171	0	2	976	86	87	9568	3539	139			
부대	0	1	0	1	1	1	0	0	0	1	25	1	0			
사무실	40	352	128	489	35	64	76	3706	77	40	2925	1183	53			
산마	2	37	12	7	14	15	1	15	8	8	181	624	49			
상점	4	390	487	152	53	55	25	420	66	23	1973	2054	120			
숙박업소	590	2798	262	58	19	0	2	156	61	43	1434	907	58			
시장노점	1	26	5	43	13	30	6	30	9	5	434	142	7			
마파트	241	2371	372	218	65	142	65	625	354	242	5115	3122	242			
역대합실	0	298	13	27	18	14	2	4	7	8	347	100	10			
유원지	2	227	55	124	49	60	21	47	16	13	835	115	11			
유흥접객업소	1	65	2	19	91	2	0	75	3	1	139	45	1			
의료기관	4	147	24	38	25	16	4	4	25	19	651	194	7			
종교기관	4	64	8	16	5	2	0	4	29	2	196	159	16			
지하철	0	1339	4	12	28	0	0	0	4	1	188	41	0			
집	348	2927	528	180	64	11	15	2693	450	312	5335	3172	354			
창고	0	24	6	4	1	1	0	44	17	4	59	67	94			
학교	1	188	35	190	31	24	5	4	19	8	844	239	27			
해상	0	2	1	2	4	1	0	0	0	0	2	31	87			
	약취와유인	업무상과실치사상	유기	장물	절도	주거침입	체포와감금	폭력행위	등처벌에관한법률위반	폭행	협박					
고속도로	0				3	1	151	0	5			53	107	2		
공사장	0				403	0	41	4003	17	6		237	349	13		
공장	0				337	0	177	3540	72	4		385	480	17		
공지	0				5	1	15	366	2	0		28	31	2		
교통	1				30	1	13	2249	7	19		116	864	12		

*컬럼의 수가 많아서 아래에서 이어서 출력.

문제 76.	서울시 물가 데이터(price.csv)를 R로 로드하고 tapply 함수를 이용해서 전통시장과 대형마트간의 물품별 가격 평균을 출력 하시오.
---------------	--

```
round(tapply(price$a_price,list(price$a_name,price$m_type_name),mean))
```

```
> round(tapply(price$a_price,list(price$a_name,price$m_type_name),mean))
```

	대형마트	전통시장
고등어	3313	3000
고등어 (30cm, 국산)	5020	3140
고등어 (30cm, 수입산)	3980	NA
고등어 (냉동, 국산)	3873	3415
고등어 (냉동, 수입산)	2430	2500
고등어 (생물, 국산)	4701	3308
고등어 (생물, 수입산)	3253	2500
냉동참조기 (20cm, 국산)	1227	684
달걀	3192	2522
달걀(완란)	2719	2098
달걀(특란)	2623	2665
닭고기	6965	5840
닭고기(육계)	7673	5597
닭고기(중간)	6637	NA
동태	2120	3583
돼지고기(삼겹살)	9403	10217
돼지고기(생삼겹살)	11331	9705
명태	0	4667
명태 (45cm, 국산)	0	NA
명태 (45cm, 수입산)	2744	3667
명태 (냉동, 수입산)	3305	2647
명태 (러시아, 냉동)	2323	2671
명태 (생물, 수입산)	2408	4611
명태 (일본산, 냉동)	NA	5033
무	1837	1533
무 (1kg)	1987	1525
무 (세척무)	2284	1333
무 (세척무, 중)	1886	1286
배	4361	2583
배 (신고)	4798	3354
배 (신고), 중급(대)	3626	2800
배 (신고), 중급(중)	3444	2190

*컬럼을 대문자-->소문자로 바꾸는 방법

```
> colnames(price)<-tolower(colnames(price))
```

15. tapply(), sapply(), lapply()

2018년 5월 19일 토요일 오후 5:16

함수	설명	사용법	결과
tapply()	요인(factor)의 수준(level)(or 벡터)별로 특정 벡터에 함수 명령어를 동시에 적용 (그룹함수 역할을 한다)	tapply(벡터, 요인 or 벡터, 함수)	벡터 또는 행렬
sapply()	데이터 프레임 여러 변수에 함수 명령어 동시에 적용	sapply(데이터 프레임, 함수)	벡터 또는 행렬
lapply()	데이터 프레임 여러 변수에 함수 명령어 동시에 적용	lapply(데이터 프레임, 함수)	리스트

1. tapply 예제

```
> str(emp)
'data.frame':  14 obs. of  8 variables:
 $ empno   : int  7839 7698 7782 7566 7654 7499 7844 7900 7521 7902 ...
 $ ename    : Factor w/ 14 levels "ADAMS","ALLEN",...: 8 3 4 7 9 2 13 6 14 5 ...
 $ job      : Factor w/ 5 levels "ANALYST","CLERK",...: 4 3 3 3 5 5 5 2 5 1 ...
 $ mgr      : int   NA 7839 7839 7839 7698 7698 7698 7698 7698 7566 ...
 $ hiredate : Factor w/ 13 levels "1980-12-09 0:00",...: 9 5 6 4 8 2 7 10 3 10 ...
 $ sal      : int  5000 2850 2450 2975 1250 1600 1500 950 1250 3000 ...
 $ comm     : int   NA  NA  NA  NA 1400 300  0  NA  500  NA ...
 $ deptno   : int   10  30  10  20  30  30  30  30  30  20 ...
```

```
tapply(emp$sal, emp$deptno, sum)
```

```
> tapply(emp$sal, emp$deptno, sum)
 10    20    30 
8750 10875  9400
```

```
tapply(emp$sal, list(emp$deptno, emp$job), sum)
```

```
> tapply(emp$sal, list(emp$deptno, emp$job), sum)
      ANALYST CLERK  MANAGER  PRESIDENT  SALESMAN
10      NA    1300    2450      5000      NA
20    6000    1900    2975      NA      NA
30      NA     950    2850      NA    5600
```

2. sapply 예제

```
sapply(emp, class)
```

```
> sapply(emp, class)
      empno      ename      job      mgr      hiredate      sal      comm      deptno
"integer" "factor"  "factor" "integer" "factor" "integer" "integer" "integer"
```

```
class(sapply(emp, class))      # 결과는 벡터 or 행렬 값
```

```
> class(sapply(emp, class))  
[1] "character"
```

* sapply 함수를 사용하지 않고 실행하면 ...

class(empno) .. class(ename) ... 처럼 변수 개수만큼 명령어를 실행 해야한다.

3. lapply 예제

```
lapply(emp, class)
```

```
> lapply(emp, class)  
$`empno`  
[1] "integer"  
  
$ename  
[1] "factor"  
  
$job  
[1] "factor"  
  
$mgr  
[1] "integer"  
  
$hiredate  
[1] "factor"  
  
$sal  
[1] "integer"  
  
$comm  
[1] "integer"  
  
$deptno  
[1] "integer"
```

class(lapply(emp, class)) # 결과는 리스트 -----> 인덱싱 하기 편하다.

```
> class(lapply(emp, class))  
[1] "list"
```


16. 정렬함수 (sort(), order(), order By())

2018년 5월 17일 목요일 오후 7:12

■ 정렬 함수의 종류

1. 벡터에서 정렬 : sort()
2. data frame의 정렬 : order()
3. doBy 패키지의 order By 함수를 사용

	sort()	order()	orderBy()
설명	벡터 정렬	데이터 프레임 정렬 각 요소에 정렬인덱스를 반환	데이터 프레임 정렬 doBy 패키지를 사용해야함 내림차순 하려면 ~뒤에 - 붙임
예제	<code>v1<-c(5,2,7,1)</code> <code>sort(v1,decreasing = F)</code>	<code>emp[order(emp\$sal, decreasing = T),</code> <code>c("ename","sal")]</code>	<code>library(doBy)</code> <code>orderBy(~sal, emp[,c("sal","ename")])</code>
결과	<code>[1] 1 2 5 7</code>	<pre> ename sal 1 KING 5000 10 FORD 3000 12 SCOTT 3000 4 JONES 2975 2 BLAKE 2850 3 CLARK 2450 6 ALLEN 1600 7 TURNER 1500</pre>	<pre> sal ename 11 800 SMITH 8 950 JAMES 13 1100 ADAMS 5 1250 MARTIN 9 1250 WARD</pre>

17. 순위출력

2018년 5월 15일 화요일 오전 11:26

■ 순위 출력을 R로 구현하는 방법 : rank

rank 사용법

rank(데이터, ties.method = "min") # ties.method = "min" 오라클의 rank와 똑같은 결과로 출력
이 옵션을 사용 안하면 같은 순위에 여러 명이 있으면 "순위.5" 로 출력 ex) 12.5

1. 예제

예제	이름, 월급, 월급에 대한 순위를 출력 하시오.
----	----------------------------

```
> library(data.table)
> data.table(emp$ename, emp$sal, rank(-emp$sal, ties.method = "min"))

> data.table(emp$ename, emp$sal, rank(-emp$sal, ties.method = "min"))
   V1    V2 V3
1:  KING 5000  1
2: BLAKE 2850  5
3:  CLARK 2450  6
4:  JONES 2975  4
5: MARTIN 1250 10
6:  ALLEN 1600  7
7: TURNER 1500  8
8:  JAMES  950 13
9:   WARD 1250 10
10:  FORD 3000  2
11: SMITH  800 14
12: SCOTT 3000  2
13: ADAMS 1100 12
14: MILLER 1300  9
```

```
data.table(emp$ename, emp$sal, rank(-emp$sal, ties.methd = 'min'))
```

Min : 오라클의 rank와 같다.

First : 오라클의 rank와 같은데 인덱스가 먼저 나오는 데이터를 높은 순서로 부여한다.

문제 101.	이름, 월급, 월급에 대한 순위를 출력하는데 순위가 1등부터 정렬이 되어서 출력되게 하시오.
---------	---

```
> x<-data.table(emp$ename, emp$sal, rank(-emp$sal, ties.method = "min"))
> x[order(x$rnk, decreasing=F),]
```

```

> x
      ename  sal rnk
1:   KING 5000   1
2:  BLAKE 2850   5
3:  CLARK 2450   6
4:  JONES 2975   4
5: MARTIN 1250  10
6:  ALLEN 1600   7
7: TURNER 1500   8
8:  JAMES  950  13
9:   WARD 1250  10
10:  FORD 3000   2
11: SMITH  800  14
12: SCOTT 3000   2
13: ADAMS 1100  12
14: MILLER 1300   9
>
>
> x[order(x$rnk, decreasing=F),]
      ename  sal rnk
1:   KING 5000   1
2:   FORD 3000   2
3:  SCOTT 3000   2
4:  JONES 2975   4
5:  BLAKE 2850   5
6:  CLARK 2450   6
7:  ALLEN 1600   7
8: TURNER 1500   8
9: MILLER 1300   9
10: MARTIN 1250  10
11:   WARD 1250  10
12: ADAMS 1100  12
13:  JAMES  950  13
14: SMITH  800  14

```

문제 102. 원섭이의 2017년 아파트 매매 데이터를 apartment 라는 변수에 담고 아래의 결과를 출력 하시오.

```

> x<-orderBy(~rnk,data.table(주소=paste(apartment$시군구,apartment$번지),
+                                아파트이름=apartment$단지명,가격=apartment$거래금액,
+                                rnk=rank(-apartment$거래금액,ties.method="min") ))

> x[x$rnk<=20,]

```

```
> apartment<-read.csv(file.choose(),header = T)
> x<-orderBy(~rnk,data.table(주소=paste(apartment$시군구,apartment$번지),
+                                     아파트이름=apartment$단지명,가격=apartment$거래금액,
+                                     rnk=rank(-apartment$거래금액,ties.method="min"))))
> x[x$rnk<=20,]
```

	주소	아파트이름	가격	rnk
1:	경기도 남양주시 별내동 1056	쌍용예가	60000	1
2:	경기도 남양주시 별내동 854	현대아이파크	60000	1
3:	경기도 남양주시 별내동 854	현대아이파크	60000	1
4:	경기도 남양주시 와부읍 도곡리 986-1	덕소두산위브	60000	1
5:	경기도 남양주시 와부읍 도곡리 986-1	덕소두산위브	60000	1
6:	경기도 남양주시 별내동 1056	쌍용예가	60000	1
7:	경기도 남양주시 별내동 854	현대아이파크	60000	1
8:	경기도 남양주시 와부읍 도곡리 986-1	덕소두산위브	60000	1
9:	경기도 남양주시 와부읍 도곡리 986-1	덕소두산위브	60000	1
10:	경기도 남양주시 지금동 3720	지금힐스테이트	60000	1
11:	경기도 남양주시 다산동 3720	지금힐스테이트	60000	1
12:	경기도 남양주시 별내동 1056	쌍용예가	60000	1
13:	경기도 남양주시 와부읍 덕소리 600-15	덕소강변현대	59950	13
14:	경기도 남양주시 별내동 854	현대아이파크	59800	14
15:	경기도 남양주시 와부읍 도곡리 1093	쌍용스윗닷홈리버	59800	14
16:	경기도 남양주시 와부읍 덕소리 612	덕소아이파크	59800	14
17:	경기도 남양주시 다산동 02-Jan	부영e그린4,5차	59800	14
18:	경기도 남양주시 다산동 02-Jan	부영e그린4,5차	59500	18
19:	경기도 남양주시 별내동 1072	KCC스위첸	59500	18
20:	경기도 남양주시 와부읍 도곡리 986-1	덕소두산위브	59500	18
21:	경기도 남양주시 다산동 02-Jan	부영e그린1차	59500	18
22:	경기도 남양주시 와부읍 덕소리 600-15	덕소강변현대	59500	18
23:	경기도 남양주시 별내동 1056	쌍용예가	59500	18
24:	경기도 남양주시 별내동 854	현대아이파크	59500	18

>

문제 103. crime_loc.csv를 R로 로드하고 병원에서 많이 발생하는 범죄유형, 건수, 순위를 출력 하시오.

```
> library(dplyr)
> x<-crime_loc[crime_loc$장소=="병원",]
> colnames(x2)<-c('crime','cnt','rnk')
> x2<-data.table(x$범죄,x$건수,dense_rank(-x$건수))
> x2[order(x2$rnk, decreasing = F)]
> x2
```

	crime	cnt	rnk
1:	절도	16053	1
2:	폭행	14142	2
3:	상해	9568	3
4:	폭력행위등처벌에관한법률위반	7396	4
5:	손괴	3539	5
6:	강간	1362	6
7:	도박과복표	976	7
8:	공갈	382	8
9:	강도	275	9
10:	협박	247	10
11:	주거침입	197	11
12:	과실치사상	171	12
13:	실화	139	13
14:	살인	87	14
15:	방화	86	15
16:	업무상과실치사상	76	16
17:	장물	39	17
18:	체포와감금	31	18
19:	간통	23	19
20:	약취와유인	8	20
21:	도로교통법위반	2	21
22:	유기	1	22
23:	교통사고처리	0	23

문제 104. 카페에서 암 발생 데이터를 내려받고 R로 로드한 후에 여자들이 많이 걸리는 암과 건수와 순위를 출력 하시오.

```
> x2<- data.table(암=x$암종,건수=x$환자수,순위=dense_rank(-x$환자수) )
> colnames(x2)<-c('cancer','cnt','rnk')
> x2[order(x2$rnk,decreasing=F), ]
```

```
> x2[order(x2$rnk,decreasing=F), ]
      cancer      cnt rnk
1:      갑상선 217874   1
2:      유방 131581   2
3:      대장 69971   3
4:      위 69490   4
5:      자궁경부 43523   5
6:      기타 암 37312   6
7:      폐 19058   7
8:      자궁체부 15191   8
9:      난소 14171   9
10:      간 12968  10
11: 비호지킨림프종 12127  11
12:      신장 8464  12
13: 담낭 및 기타담도 7246  13
14:      백혈병 6674  14
15:      구강및인두 5523  15
16:      방광 4743  16
17: 뇌 및 중추신경계 4118  17
18:      췌장 3229  18
19: 다발성 골수종 1800  19
20:      식도 730  20
21: 호지킨림프종 725  21
22:      후두 524  22
      cancer      cnt rnk
```

문제 105. 2009년도에 서울시에서 교통사고가 일어난 장소, 건수, 순위를 출력 하시오.

```
> x<-car[car$기준년도 == 2009 & car$지자체=='서울',]
> colnames(x)<-c('년도','지자체','장소','건수','사고유형')
> x2<-data.table(x$장소,x$건수,dense_rank(-x$건수))
> colnames(x2)<-c("loc",'cnt','rnk')
> x2[order(x2$rnk,decreasing = F), ]
> x2[order(x2$rnk,decreasing = F), ]
      loc      cnt rnk
1:      수유동 먹자골목 49   1
2:      롯데백화점 앞 노상 망우로 29   2
3:      종로2가교차로 횡단보도상 28   3
4:      헬로광 약국 앞 27   4
5:      엘에이모델 25   5
---
1496:      영등포역전앞 버스정류장 2  27
1497:      국민은행 2  27
1498:      서원동(신림본동) 2  27
1499: 방배역 사거리 지하철 4번출구 앞 2  27
1500:      현대인테크 2  27
```

18. IF 문 / LOOP문

2018년 5월 21일 월요일 오후 2:15

1. 조건&반복문의 사용법

구분	문법
if문 사용법	if (조건식) { } else if (조건식) { } else { }
for loop문 문법	for (루프변수 in 리스트) { 반복할 실행문 }
while loop문 문법	while(조건식) { #조건식이 true일 경우 실행된다 반복할 실행문 }

■ 분기문

break문 : Break문을 사용하면 for loop문이나 while loop문 반복문 실행 도중 루프문에서 벗어 날 수 있다.

next문 : loop문에서 next를 만나면 아래의 명령어를 실행하지 않고 즉시 조건을 비교하는 위치부터 실행
(Java의 **continue문**과 동일한 역할)

2. 예제

문제 188.	이름을 물어 보게하고 이름을 입력하면 해당 사원이 고소득자인지 일반 소득자인지 저소득자인지 출력되는 함수를 생성 하시오.
---------	---

```
income <-function(){  
  
  res<-readline(prompt = '이름을 입력하세요')  
  x<-emp[emp$ename == res , "sal"]  
  
  if (x >= 3000)  
    print('고소득자')  
  else if (x >=2000)  
    print('일반 소득자')  
  else  
    print('저소득자')  
}
```

```
> income()  
이름을 입력하세요KING  
[1] "고소득자"
```

문제 189.	이름을 물어 보게하고 이름을 입력하면 해당 사원의 커미션이 null이면 보너스 대상입니다 라는 메시지가 출력되게 하고 null이 아니면 보너스 대상자가 아닙니다라는 메시지가 출력되게 하시오.
----------------	--

```
income2 <-function(){

  res<-readline(prompt = '이름을 입력하세요')

  x<-emp[emp$ename == res , "comm"]

  if (is.na(x))
    print('보너스 대상자입니다.')
  else
    print('보너스 대상자가 아닙니다.')

}
```

```
> income2()
이름을 입력하세요WARD
[1] "보너스 대상자가 아닙니다."
```

문제 190.	아래와 같이 함수를 실행하면 ★가 출력되게 하시오.
----------------	------------------------------

```
func190<-function(x){

  for(i in 1:x){
    s<-"
    for(j in 1:i)
      s<-paste(s,'★')
    print(s)
  }
}
```

```
> func190(6)
[1] " ★"
[1] " ★ ★"
[1] " ★ ★ ★"
[1] " ★ ★ ★ ★"
[1] " ★ ★ ★ ★ ★"
[1] " ★ ★ ★ ★ ★ ★"
```

문제 191.	아래와 같이 팩토리얼 함수를 생성 하시오.
----------------	-------------------------

```
func191 <-function(x){
  s<-1
```

```

for(i in 1:x) s<-s*i
print(s)
}

```

```

> func191(5)
[1] 120

```

문제 192.	문제 191 번의 팩토리얼 구하는 함수를 디버깅 패키지를 사용해서 디버깅 하시오.
----------------	---

```

debug(
  func191_ <-function(x){
    s<-1
    m<-""
    for(i in 1:x){
      s<-s*i
      if (i<x)
        m<-paste(m,i,"x",sep = "")
      else {
        m<-paste(m,i,'=',s,sep = "")
        print(m)
      }
    }
  }
)

```

#최초에 함수를 통째로 넣어줘야 밑에 처럼 간략히 호출가능

```

debug(func191_(5))

```



```

1 function(x){
2
3   s<-1
4   m<-''
5   for(i in 1:x){
6     s<-s*i
7     if (i<x)
8       m<-paste(m,i,"x",sep = "")
9     else {
10      m<-paste(m,i,"=",s,sep = "")
11      print(m)
12    }
13  }
14 }

```

Console Terminal x Deploy x

~f ↻

Next { } < > Continue Stop

```

      m <- paste(m, i, "x", sep = "")
    else {
      m <- paste(m, i, "=", s, sep = "")
      print(m)
    }
  }
}
Browse[2]> |

```

문제 193. 10을 입력하면 55가 출력 되게하는 함수를 while loop문으로 생성 하시오.

```

func193<-function(x){
  s<-0
  i<-1
  while(i<=x){
    s = s+i
    i<-i+1
  }
  print(s)
}

```

```

> func193(10)
[1] 55

```

문제 194. 아래의 power함수를 생성 하시오.
(^를 사용하지 말고, while loop문과 break문으로 구현 하시오.)

```

power<-function(x,y){

  s<-0
  sum<-1

```

```

while(TRUE){

  if(s>=y) break

  sum<-sum*x
  s<-s+1
}
print(sum)
}

> power(2,4)
[1] 16

```

문제 195.	아래와 같이 log7 이라는 로그 함수를 구현 하시오.(while loop문과 break 문으로 구현 하시오) EX. log(2, 16) # 결과는 4입니다.
----------------	---

```

log7 <- function(x,y){
  cnt <-0
  while(TRUE){
    y<-y/x
    cnt<-cnt+1
    if (y<x) break
  }
  print(cnt)
}

> log7(2,16)
[1] 4

```

문제 196.	24와 16의 최대공약수를 구하시오.
----------------	----------------------

```

max_gong_yaksu <- function(x,y){

  n<-1
  if (x>y){
    n<-x%%y
    if (n == 0)
      print(paste('최대 공약수는 ',y,'입니다.'))
    else if (y%%n == 0)
      print(paste('최대 공약수는 ',n,'입니다.'))
    else
      print('최대 공약수는 1 입니다.')
  }
  else{
    n<-y%%x

```

```

if (n==0)
  print(paste('최대 공약수는 ',x,'입니다.))
else if (x%%n==0){
  print(paste('최대 공약수는 ',n,'입니다.))
}else
  print(paste('최대 공약수는 1 입니다.))
}
}

```

```

> max_gong_yaksu(32,4)
[1] "최대 공약수는 4 입니다."
> max_gong_yaksu(24,18)
[1] "최대 공약수는 6 입니다."

```

19. 조인 (merge문)

2018년 5월 11일 금요일 오전 10:57

■ merge문

서로 다른 dataframe의 변수들을 하나의 결과로 출력할 때 사용하는 문법으로 R에서는 merge 함수를 사용해서 구현한다.

* merge문 사용법

```
merge(테이블1(x), 테이블2(y), by='x.컬럼명', by='y.컬럼명')
```

1. 예제

문제 77. Dept.csv를 내려받고 dept라는 변수에 로드 하시오.

```
> dept<-read.csv("c:\\data\\dept.csv",header=T)
>
>
>
> dept
  deptno      dname      loc
1     10 ACCOUNTING NEW YORK
2     20  RESEARCH  DALLAS
3     30    SALES   CHICAGO
4     40 OPERATIONS  BOSTON
> |
```

문제 78. Emp data frame 과 dept data frame 을 merge 해서 이름, 월급, 부서위치를 출력 하시오.

```
merge(emp,dept,by="deptno") [,c("ename","sal","loc")]
```

```
> merge(emp,dept,by="deptno") [,c("ename","sal","loc")]
  ename  sal      loc
1   KING 5000 NEW YORK
2 MILLER 1300 NEW YORK
3  CLARK 2450 NEW YORK
4  JONES 2975  DALLAS
5  SMITH  800  DALLAS
6  SCOTT 3000  DALLAS
7  ADAMS 1100  DALLAS
8   FORD 3000  DALLAS
9  JAMES  950 CHICAGO
10 MARTIN 1250 CHICAGO
11 BLAKE 2850 CHICAGO
12 TURNER 1500 CHICAGO
13  WARD 1250 CHICAGO
14  ALLEN 1600 CHICAGO
```

문제 79. 부서위치가 DALLAS인 직원들의 이름, 월급, 부서위치를 출력 하시오.

```
> merge(emp,dept,by="deptno") [merge(emp,dept)$loc == 'DALLAS',c("ename","sal","loc")]
```

```
> merge(emp,dept,by="deptno") [merge(emp,dept)$loc == 'DALLAS',c("ename","sal","loc")]
  ename sal   loc
4 JONES 2975 DALLAS
5 SMITH  800 DALLAS
6 SCOTT 3000 DALLAS
7 ADAMS 1100 DALLAS
8 FORD  3000 DALLAS
```

문제 80. 월급이 1200이상이고 직업이 SALESMAN인 직원들의 이름, 월급, 직업, 부서위치를 출력 하시오.

```
> merge(emp,dept,by="deptno") [merge(emp,dept)$sal >= 1200 & merge(emp,dept)$job ==
```

```
'SALESMAN',c("ename","sal","job","loc")]
```

```
> merge(emp,dept,by="deptno") [merge(emp,dept)$sal >= 1200 & merge(emp,dept)$job == 'SALESMAN',c("ename","sal","job","loc")]
  ename sal   job   loc
10 MARTIN 1250 SALESMAN CHICAGO
12 TURNER 1500 SALESMAN CHICAGO
13 WARD  1250 SALESMAN CHICAGO
14 ALLEN  1600 SALESMAN CHICAGO
```

-----> 소스가 너무 길어진다. 조인한 결과를 변수에 넣어서 사용하도록 하자!

```
> x<-merge(emp,dept,by="deptno")
```

```
> x[x$sal >= 1200 & x$job == 'SALESMAN',c("ename","sal","job","loc")]
```

```
> x<-merge(emp,dept,by="deptno")
> x
  deptno empno  ename      job mgr  hiredate  sal comm      dname      loc
1      10   7839   KING PRESIDENT  NA 1981-11-17 5000   NA ACCOUNTING NEW YORK
2      10   7934  MILLER      CLERK 7782 1982-01-11 1300   NA ACCOUNTING NEW YORK
3      10   7782   CLARK  MANAGER 7839 1981-05-09 2450   NA ACCOUNTING NEW YORK
4      20   7566   JONES  MANAGER 7839 1981-04-01 2975   NA  RESEARCH  DALLAS
5      20   7369   SMITH      CLERK 7902 1980-12-09  800   NA  RESEARCH  DALLAS
6      20   7788   SCOTT  ANALYST 7566 1982-12-22 3000   NA  RESEARCH  DALLAS
7      20   7876   ADAMS      CLERK 7788 1983-01-15 1100   NA  RESEARCH  DALLAS
8      20   7902   FORD   ANALYST 7566 1981-12-11 3000   NA  RESEARCH  DALLAS
9      30   7900   JAMES      CLERK 7698 1981-12-11  950   NA    SALES  CHICAGO
10     30   7654  MARTIN  SALESMAN 7698 1981-09-10 1250 1400    SALES  CHICAGO
11     30   7698   BLAKE  MANAGER 7839 1981-05-01 2850   NA    SALES  CHICAGO
12     30   7844  TURNER  SALESMAN 7698 1981-08-21 1500   0    SALES  CHICAGO
13     30   7521   WARD   SALESMAN 7698 1981-02-23 1250  500    SALES  CHICAGO
14     30   7499   ALLEN  SALESMAN 7698 1981-02-11 1600  300    SALES  CHICAGO
> x[x$sal >= 1200 & x$job == 'SALESMAN',c("ename","sal","job","loc")]
  ename sal   job   loc
10 MARTIN 1250 SALESMAN CHICAGO
12 TURNER 1500 SALESMAN CHICAGO
13 WARD  1250 SALESMAN CHICAGO
14 ALLEN  1600 SALESMAN CHICAGO
```

문제 81. 커미션이 NA인 직원들의 이름, 부서위치, 커미션을 출력 하시오.

```
x[is.na(x$comm),c("ename","deptno","comm")] ]
```

```
> x[is.na(x$comm),c("ename", "deptno", "comm")]
  ename deptno comm
1   KING     10  NA
2 MILLER     10  NA
3   CLARK     10  NA
4   JONES     20  NA
5   SMITH     20  NA
6   SCOTT     20  NA
7   ADAMS     20  NA
8    FORD     20  NA
9   JAMES     30  NA
11  BLAKE     30  NA
```

문제 82. 이름, 부서위치를 출력하는데 오라클의 outer join과 같은 결과를 출력 하시오.

```
merge(emp,dept,by="deptno", all.y=T)[,c("ename","loc")]
```

emp 위치가 x, dept 위치가 y
all.y = T y쪽이 모두 나오게 해라

```
> merge(emp,dept,by="deptno",all.y=T)[,c("ename", "loc")]
  ename      loc
1   KING NEW YORK
2 MILLER NEW YORK
3   CLARK NEW YORK
4   JONES  DALLAS
5   SMITH  DALLAS
6   SCOTT  DALLAS
7   ADAMS  DALLAS
8    FORD  DALLAS
9   JAMES CHICAGO
10 MARTIN CHICAGO
11  BLAKE CHICAGO
12 TURNER CHICAGO
13   WARD CHICAGO
14  ALLEN CHICAGO
15   <NA>  BOSTON
```

문제 83. 이름, 부서위치를 출력하는데 오라클의 full outer join과 같은 결과를 출력 하시오.

```
merge(emp,dept,by="deptno",all=T)[,c("ename","loc")]
```

```
> merge(emp,dept,by="deptno",all=T)[,c("ename", "loc")]
  ename      loc
1   KING NEW YORK
2 MILLER NEW YORK
3   CLARK NEW YORK
4   JONES  DALLAS
5   SMITH  DALLAS
6   SCOTT  DALLAS
7   ADAMS  DALLAS
8    FORD  DALLAS
9   JAMES CHICAGO
10 MARTIN CHICAGO
11  BLAKE CHICAGO
12 TURNER CHICAGO
13   WARD CHICAGO
14  ALLEN CHICAGO
15   <NA>  BOSTON
```

문제 84. 이름, 자기의 직속 상사의 이름(관리자)를 출력 하시오.

```
merge(emp,emp,by.x="mgr",by.y="empno")[,c("ename.x","ename.y")]
```

```
> merge(emp,emp,by.x="mgr",by.y="empno")[,c("ename.x","ename.y")]
  ename.x ename.y
1    FORD   JONES
2   SCOTT   JONES
3  MARTIN   BLAKE
4   ALLEN   BLAKE
5  TURNER   BLAKE
6   JAMES   BLAKE
7    WARD   BLAKE
8  MILLER   CLARK
9   ADAMS   SCOTT
10  BLAKE    KING
11  CLARK    KING
12  JONES    KING
13  SMITH    FORD
```

문제 85. 문제 84번을 다시 수행하는데, 자신의 월급이 자신의 관리자인 사원의 월급보다 더 많은 월급을 받는 사원들만 출력 하시오.

```
x<-merge(emp,emp,by.x="mgr",by.y="empno")
x[x$sal.x >= x$sal.y , c("ename.x","sal.x")]
```

```
> x<-merge(emp,emp,by.x="mgr",by.y="empno")
> x
  mgr empno ename.x job.x hiredate.x sal.x comm.x deptno.x ename.y job.y mgr.y hiredate.y sal.y comm.y deptno.y
1 7566 7902   FORD ANALYST 1981-12-11 3000    NA      20   JONES  MANAGER 7839 1981-04-01 2975    NA      20
2 7566 7788   SCOTT ANALYST 1982-12-22 3000    NA      20   JONES  MANAGER 7839 1981-04-01 2975    NA      20
3 7698 7654  MARTIN SALESMAN 1981-09-10 1250  1400    30   BLAKE  MANAGER 7839 1981-05-01 2850    NA      30
4 7698 7499   ALLEN SALESMAN 1981-02-11 1600    300    30   BLAKE  MANAGER 7839 1981-05-01 2850    NA      30
5 7698 7844  TURNER SALESMAN 1981-08-21 1500     0     30   BLAKE  MANAGER 7839 1981-05-01 2850    NA      30
6 7698 7900   JAMES CLERK 1981-12-11  950    NA     30   BLAKE  MANAGER 7839 1981-05-01 2850    NA      30
7 7698 7521   WARD SALESMAN 1981-02-23 1250    500    30   BLAKE  MANAGER 7839 1981-05-01 2850    NA      30
8 7782 7934  MILLER CLERK 1982-01-11 1300    NA     10   CLARK  MANAGER 7839 1981-05-09 2450    NA      10
9 7788 7876  ADAMS CLERK 1983-01-15 1100    NA     20   SCOTT  ANALYST 7566 1982-12-22 3000    NA      20
10 7839 7698   BLAKE MANAGER 1981-05-01 2850    NA     30   KING  PRESIDENT NA 1981-11-17 5000    NA      10
11 7839 7782   CLARK MANAGER 1981-05-09 2450    NA     10   KING  PRESIDENT NA 1981-11-17 5000    NA      10
12 7839 7566   JONES MANAGER 1981-04-01 2975    NA     20   KING  PRESIDENT NA 1981-11-17 5000    NA      10
13 7902 7369  SMITH CLERK 1980-12-09  800    NA     20   FORD   ANALYST 7566 1981-12-11 3000    NA      20

> x[x$sal.x >= x$sal.y , c("ename.x","sal.x")]
  ename.x sal.x
1    FORD 3000
2   SCOTT 3000
```

문제 86. 부서위치, 부서위치별 토탈 월급을 출력 하시오.

```
x<-merge(emp,dept,by="deptno")
aggregate(sal~loc,x,sum)
```

```
> aggregate(sal~loc,x,sum)
  loc    sal
1 CHICAGO 9400
2 DALLAS 10875
3 NEW YORK 8750
```

문제 87. 부서위치, 부서위치별 토탈 월급을 출력 하시오. (가로출력)

```
x<-merge(emp,dept,by="deptno")
tapply(x$sal,x$loc,sum)
```

```
> x<-merge(emp,dept,by="deptno")
> tapply(x$sal,x$loc,sum)
BOSTON  CHICAGO  DALLAS  NEW YORK
      NA      9400    10875    8750
```

문제 88. 부서위치(세로), 입사한 년도4자리(가로), 부서위치별 입사한 년도별 인원수를 출력 하시오.

```
tapply(x$empno, list(x$loc,substr(x$hiredate,1,4)), sum)
```

```
> tapply(x$empno, list(x$loc,substr(x$hiredate,1,4)), sum)
      1980  1981  1982  1983
BOSTON    NA    NA    NA    NA
CHICAGO    NA 46116    NA    NA
DALLAS   7369 15468 7788 7876
NEW YORK    NA 15621 7934    NA
```

문제 89. 부서위치(세로), 입사한 년도4자리(가로), 부서위치별 입사한 년도별 인원수를 출력 하시오.

```
y<-tapply(x$empno, list(x$loc,substr(x$hiredate,1,4)), sum)
ifelse(is.na(y),0,y)
```

```
> ifelse(is.na(y),0,y)
      1980  1981  1982  1983
BOSTON     0     0     0     0
CHICAGO     0 46116     0     0
DALLAS   7369 15468 7788 7876
NEW YORK     0 15621 7934     0
```

문제 90. 부서위치(세로), 입사한 년도4자리(가로), 부서위치별 입사한 년도별 인원수를 출력 하시오.

```
rbind(aggregate(sal~deptno,emp,sum),c("",sum(emp$sal)))
```

```
> rbind(aggregate(sal~deptno,emp,sum),c('',sum(emp$sal)))
  deptno  sal
1     10 8750
2     20 10875
3     30  9400
4      NA 29025
```

```
rbind(aggregate(sal~deptno,emp,sum),cbind(deptno="",sal=sum(emp$sal)))
```

```
> rbind(aggregate(sal~deptno,emp,sum),cbind(deptno='',sal=sum(emp$sal)))
  deptno  sal
1     10 8750
2     20 10875
3     30  9400
4      NA 29025
```

문제 91. 아래의 SQL을 R로 구현 하시오.

20. 집합 연산자

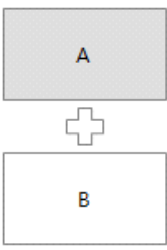
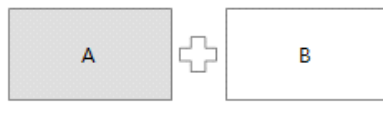
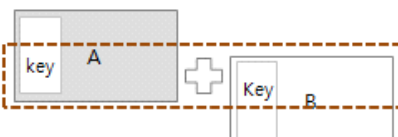
2018년 5월 11일 금요일 오후 2:08

1. 집합연산자 종류

Merge는 양 옆으로 결과를 합치는 것이었다면, 집합 연산자는 위아래로 결과를 합치는 문법이다.

오라클	R	예시
union all	rbind	rbind(데이터1, 데이터2, ...)
Union	rbind+unique	unique(rbind(데이터1, 데이터2, ..))
Intersect	(dplyr 패키지사용) intersect	
Minus	(dplyr 패키지사용) setdiff	

*rbind랑 연관된 다른 문법 : cbind

rbind(A, B)	cbind(A, B)	merge(A, B, by='key')
 <p>행 결합</p>	 <p>열 결합</p>	 <p>동일 key 값 기준 결합</p> <p>http://rfriend.tistory.com</p>

2. 예제

문제 89. 아래의 SQL의 결과를 R로 구현 하시오.

```
SELECT ename, sal, deptno
FROM EMP
WHERE deptno IN (10,20)
UNION all
SELECT ename, sal,deptno
FROM EMP
WHERE deptno=30;
```

ENAME	SAL	DEPTNO
KING	5000	10
CLARK	2450	10
JONES	2975	20
FORD	3000	20
SMITH	800	20
SCOTT	3000	20
ADAMS	1100	20
MILLER	1300	10
BLAKE	2850	30
MARTIN	1250	30
ALLEN	1600	30

```
> rbind(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],emp[emp$deptno == 30,
c("ename","sal","deptno")])
```

```
> rbind(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],emp[emp$deptno == 30, c("ename","sal","deptno")])
  ename sal deptno
1  KING 5000     10
3  CLARK 2450     10
4  JONES 2975     20
10 FORD 3000     20
11 SMITH 800     20
12 SCOTT 3000     20
13 ADAMS 1100     20
14 MILLER 1300    10
2  BLAKE 2850     30
5  MARTIN 1250     30
6  ALLEN 1600     30
7  TURNER 1500     30
8  JAMES 950     30
9  WARD 1250     30
```

문제 90. 아래의 SQL의 결과를 R로 구현 하시오.

```
SELECT deptno, SUM(sal)
  FROM EMP
 GROUP BY deptno
 UNION ALL
SELECT NULL,SUM(sal)
  FROM EMP;
```

	DEPTNO	SUM(SAL)
1	30	9400
2	20	10875
3	10	8750
4	(null)	29025

```
> rbind(aggregate(sal~deptno,emp,sum),cbind(deptno="",sal=sum(emp$sal)))
```

```
> rbind(aggregate(sal~deptno,emp,sum),cbind(deptno=' ',sal=sum(emp$sal)))
  deptno sal
1     10 8750
2     20 10875
3     30 9400
4      29025
```

문제 91. 아래의 SQL의 결과를 R로 구현 하시오.

```
SELECT deptno, SUM(sal)
  FROM EMP
 GROUP BY deptno
 UNION ALL
SELECT NULL,SUM(sal)
  FROM EMP;
```

	DEPTNO	SUM(SAL)
1	30	9400
2	20	10875
3	10	8750
4	(null)	29025

```
> rbind(aggregate(empno~job,emp,length),cbind(job='토탈값',empno=length(emp$ename)))
```

	job	empno
1	ANALYST	2
2	CLERK	4
3	MANAGER	3
4	PRESIDENT	1
5	SALESMAN	4
6	토탈값	14

문제 92. 아래의 SQL의 결과를 R로 구현 하시오.

```
SELECT ename, sal, deptno
  FROM EMP
 WHERE deptno IN (10,20)
 UNION
SELECT ename, sal, deptno
  FROM EMP
 WHERE deptno = 10;
```

	ENAME	SAL	DEPTNO
1	ADAMS	1100	20
2	CLARK	2450	10
3	FORD	3000	20
4	JONES	2975	20
5	KING	5000	10
6	MILLER	1300	10
7	SCOTT	3000	20
8	SMITH	800	20

*중복이 제거되서 출력됨. -- union all 은 중복까지 모두 출력

unique(

```

+ rbind(
+   emp[emp$deptno %in% c(10,20),c('ename','sal','deptno')],
+   emp[emp$deptno == 10, c('ename','sal','deptno')]
+ )
+ )

> unique(
+   rbind(
+     emp[emp$deptno %in% c(10,20),c('ename','sal','deptno')],
+     emp[emp$deptno == 10, c('ename','sal','deptno')]
+   )
+ )
  ename sal deptno
1   KING 5000    10
3  CLARK 2450    10
4  JONES 2975    20
10  FORD 3000    20
11 SMITH  800    20
12 SCOTT 3000    20
13 ADAMS 1100    20
14 MILLER 1300   10

```

문제 93. 아래의 SQL을 R로 구현 하시오.

```

SELECT ename, sal, deptno
FROM EMP
  WHERE deptno IN (10,20)
INTERSECT
SELECT ename, sal, deptno
FROM EMP
  WHERE deptno = 10;

```

	ENAME	SAL	DEPTNO
1	CLARK	2450	10
2	KING	5000	10
3	MILLER	1300	10

```

> x<-intersect(emp[emp$deptno %in% c(10,20),c("ename")],emp[emp$deptno==10,c("ename")])
> emp[emp$ename %in% x, c("ename","sal","deptno")]

```

```

> x<-intersect(emp[emp$deptno %in% c(10,20),c("ename")],emp[emp$deptno==10,c("ename")])
> emp[emp$ename %in% x, c("ename","sal","deptno")]
  ename sal deptno
1   KING 5000    10
3  CLARK 2450    10
14 MILLER 1300    10

```

-----> 더 간단한 방법

```

> library(dplyr)
> intersect(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],
+   emp[emp$deptno == 10 , c("ename","sal","deptno")])

```

```
> intersect(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],
+           emp[emp$deptno ==10 , c("ename","sal","deptno")])
  ename  sal deptno
1  KING 5000     10
2  CLARK 2450     10
3  MILLER 1300     10
```

문제 94. 아래의 SQL을 R로 구현 하시오.

```
SELECT ename, sal, deptno
FROM EMP
WHERE deptno IN (10,20)
MINUS
SELECT ename, sal, deptno
FROM EMP
WHERE deptno = 10;
```

	ENAME	SAL	DEPTNO
1	ADAMS	1100	20
2	FORD	3000	20
3	JONES	2975	20
4	SCOTT	3000	20
5	SMITH	800	20

```
setdiff(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],
+       emp[emp$deptno ==10 , c("ename","sal","deptno")])
```

```
> setdiff(emp[emp$deptno %in% c(10,20), c("ename","sal","deptno")],
+         emp[emp$deptno ==10 , c("ename","sal","deptno")])
  ename  sal deptno
1 JONES 2975     20
2  FORD 3000     20
3  SMITH  800     20
4  SCOTT 3000     20
5  ADAMS 1100     20
```

21. 서브쿼리

2018년 5월 11일 금요일 오후 3:33

*오라클의 서브쿼리

- 1.single row subquery
- 2.multiple row subquery
- 3.multiple column subquery

오라클의 서브쿼리를 R로 구현하는 방법은 변수만 잘 활용하면 된다.

1. 예제

문제 94. JONES의 월급보다 더 많은 월급을 받는 직원들의 이름, 월급을 출력 하시오.

```
SELECT ENAME, SAL
FROM EMP
WHERE sal > (SELECT sal
              FROM EMP
              WHERE ename = 'JONES');
```

	ENAME	SAL
1	KING	5000
2	FORD	3000
3	SCOTT	3000

```
> x <- emp[emp$ename == 'JONES', "sal"]    -- 변수에 2975 저장
> emp[emp$sal > x, c("ename", "sal")]
```

```
> emp[emp$sal > x, c("ename", "sal")]
  ename sal
1  KING 5000
10 FORD 3000
12 SCOTT 3000
```

*서브쿼리 처럼 하는 방법

```
emp[emp$sal > emp$sal[emp$ename == 'JONES'], c('ename', 'sal')]
```

```
> emp[emp$sal > emp$sal[emp$ename == 'JONES'], c('ename', 'sal')]
  ename sal
1  KING 5000
10 FORD 3000
12 SCOTT 3000
```

문제 96. 최대 월급을 받는 사원의 이름, 월급을 출력 하시오.

```
> emp[emp$sal == max(emp$sal),c("ename","sal")]
```

```
> emp[emp$sal == max(emp$sal),c("ename","sal")]
  ename sal
1  KING 5000
```

문제 97. KING에게 보고 하는 직원들의 이름, 월급을 출력 하시오.

```
> emp[emp$mgr == emp$empno[emp$ename == 'KING'],c("ename","sal")]
```

```
> emp[emp$mgr == emp$empno[emp$ename == 'KING'],c("ename","sal")]
  ename sal
NA  <NA>  NA
2  BLAKE 2850
3  CLARK 2450
4  JONES 2975
```

-----> NA 로우를 지우려면 ? na.omit()

```
> na.omit(emp[emp$mgr == emp$empno[emp$ename == 'KING'],c("ename","sal")])
```

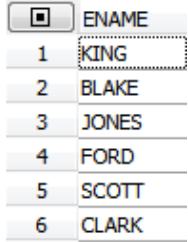
```
> na.omit(emp[emp$mgr == emp$empno[emp$ename == 'KING'],c("ename","sal")])
  ename sal
2  BLAKE 2850
3  CLARK 2450
4  JONES 2975
```

문제 98 관리자인 직원들의 이름을 출력 하시오.

```
SELECT ename
FROM EMP
WHERE empno IN (SELECT mgr FROM emp);
```

	ENAME
1	KING
2	BLAKE
3	JONES
4	FORD
5	SCOTT
6	CLARK

```
> emp[emp$empno %in% emp$mgr,"ename"]
> emp[emp$empno %in% emp$mgr,"ename"]
[1] KING BLAKE CLARK JONES FORD SCOTT
```

문제 99	<p>관리자가 아닌 직원들의 이름을 출력 하시오.</p> <pre>SELECT ename FROM EMP WHERE empno IN (SELECT mgr FROM emp);</pre> 
-------	--

```
> emp[!emp$empno %in% emp$mgr,"ename"]
```

```
> emp[!emp$empno %in% emp$mgr,"ename"]
[1] MARTIN ALLEN TURNER JAMES WARD SMITH ADAMS MILLER
```

문제 100	<p>아래의 수학문제를 R로 구현 하시오.</p> <p>문제</p> <p>흰공 2개, 빨간공 2개가 들어있는 주머니가 있다. 이 주머니에서 임의로 2개의 공을 동시에 꺼낼 때, 꺼낸 2개의 공이 모두 흰공일 확률이 $\frac{q}{p}$ 이다. $p + q$ 를 구하시오</p> <p>방법1. 조합에 관련한 패키지를 찾아서 푼다. 방법2. 함수를 직접 만들어서 구현한다.</p> <p>예: <code>combination(2,2) + combination(6,2) = 16</code></p>
--------	---

```
> combination<- function(x,y){
+
+   za = 1
+   mo = 1
+
+   for (i in (x-y+1):x){
+     za = za*i
+   }
+
+   for (i in 1:y){
+     mo = mo*i
+   }
+
+   return(za/mo)
```



```
+ }  
> combination(2,2) + combination(6,2)  
  
> combination<- function(x,y){  
+  
+   za = 1  
+   mo = 1  
+  
+   for (i in (x-y+1):x){  
+     za = za*i  
+   }  
+  
+   for (i in 1:y){  
+     mo = mo*i  
+   }  
+  
+   return(za/mo)  
+ }  
> combination(2,2) + combination(6,2)  
[1] 16
```

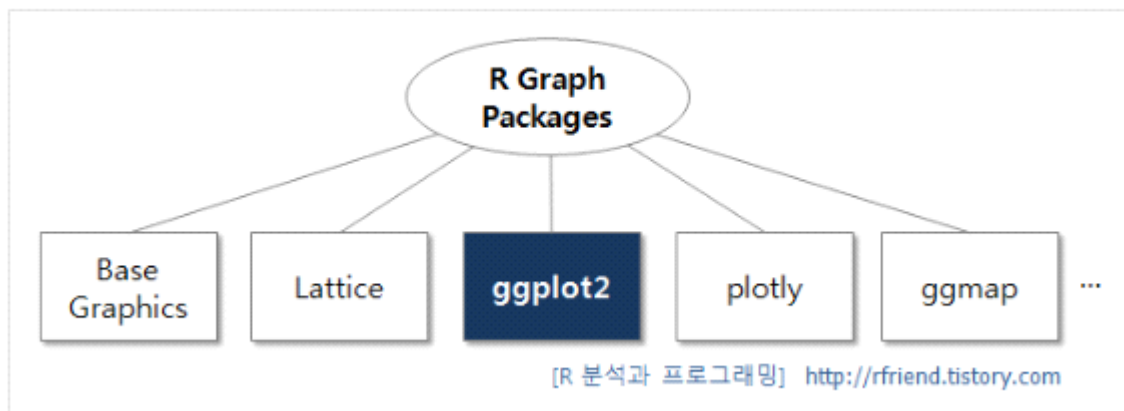
22. R의 그래프 이론

2018년 5월 17일 목요일 오후 1:47

1. R 그래프 패키지 소개 (Graphics, Lattice, ggplot2)

가장 많이 사용되는 패키지 3개를 들자면 Base Graphics package(Base package로서 별도 설치 필요 없음), Lattice package(별도 설치/호출 필요), ggplot2 package(별도 설치/호출 필요) 를 꼽을 수 있다.

[R 그래프 패키지]



package	author	장점	단점
Base Graphics	R Core Team and contributors worldwide	- 별도 설치/호출 필요 없음 - 쉽고 편함 - 사람이 생각하는 방식처럼 순차적으로 그래프를 쌓아감	- 한번 실행하면 취소 못함 - 미리 계획 필요 (예: 세로 축 scale)
Lattice	Deepayan Sarkar	- 전체 데이터를 보고 세로 축, 마진, 여백 자동 계산 편리 - 여러개 그래프를 동시에 하나의 화면에 그릴 때 편리	- 순차적으로 그래프 쌓아가는 것 안됨 - 직관적이지 못함
ggplot2	Hadley Wickham	- Base Graphics 와 Lattice의 장점만 골라냈음 - 그래프 문법에 따라 체계적, 통계적 조건 등 부여하여 고급 그래프 생성 가능	- 처음 배우기가 상대적으로 어려움 (단, 일단 문법이 익으면 그때부터는 생산성 더 높음)

2. 데이터에 맞는 그래프를 선택하려면?

1. 원형 그래프 : 데이터간의 비율을 확인할 때 유용하다.
2. 막대 그래프 : 데이터의 구체적인 수치를 비교 및 확인하고 싶을 때
3. 산포도 그래프 : 시간 흐름에 따른 데이터의 변화를 확인할 때와 데이터 간의 상관관계를 확인할 때

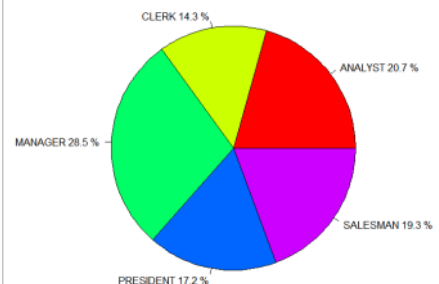
[변수 개수별 형태별 그래프 종류]

변수 개수	변수 형태	그래프
일변량 (변수 1개)	연속형 데이터	<ul style="list-style-type: none"> 히스토그램 (Histogram) 커널 밀도 곡선 (Kernel Density Curve) 박스 그래프 (Box Plot) 바이올린 그래프 (Violin Plot)
	범주형 데이터 (명목형, 순서형)	<ul style="list-style-type: none"> 막대 그림 (Bar Chart) 원 그림 (Pie Chart)
다변량 (변수 2개 이상)	연속형 데이터	<ul style="list-style-type: none"> 산점도 (행렬) 선 그래프 시계열 그래프 (x 시간*y Value)
	범주형 데이터	<ul style="list-style-type: none"> 모자이크 그래프 (Mosaic Plot)

[R 분석과 프로그래밍] <http://rfriend.tistory.com>

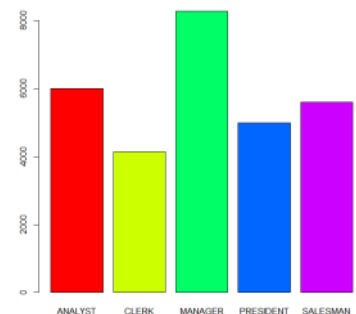
원형(pie) 그래프

- 데이터간의 비율 확인 시 유용
- 확실한 크기차이를 확인하기에는 어렵 다.



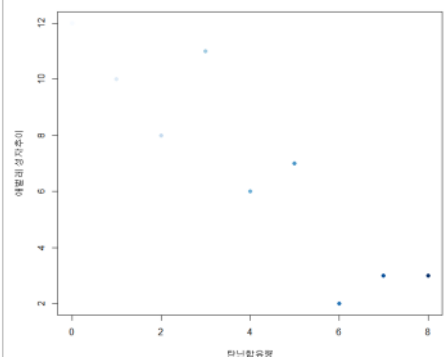
막대(bar) 그래프

- 데이터의 크기 차이를 섬세하게 확인 시 사용
- 원형 그래프의 단점 보완

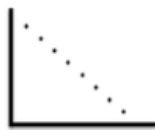


산포도(plot) 그래프

- 데이터 간의 상관관계 확인할 때
- 시간의 흐름에 따른 데이터의 변화를 확인 할 때

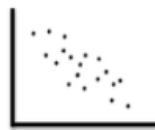


상관계수 : col (x축, y축)



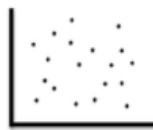
$r = -1$

음의 상관관계가
강하다.



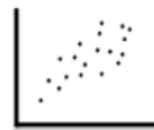
$-1 < r < 0$

음의 상관관계가
있기는 하다.



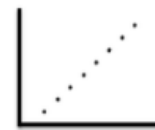
$r = 0$

상관관계가 없다.



$0 < r < 1$

양의 상관관계가
있기는 하다.



$r = +1$

양의 상관관계가
강하다.

0.0 ~ 0.2	상관관계가 거의 없다.
0.2 ~ 0.4	상관관계가 낮다.
0.4 ~ 0.6	상관관계가 있다.
0.6 ~ 0.8	상관관계가 높다.
0.8 ~ 1.0	상관관계가 매우 높다.

23. 막대 그래프 (barplot)

2018년 5월 15일 화요일 오후 2:59

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 지도 그래프 & 소리 시각화
6. 워드 클라우드
7. 사분위수 그래프

■ 막대그래프 (barplot)

* **barplot()** : 막대 그래프 함수 Ex. barplot(데이터, [옵션...])

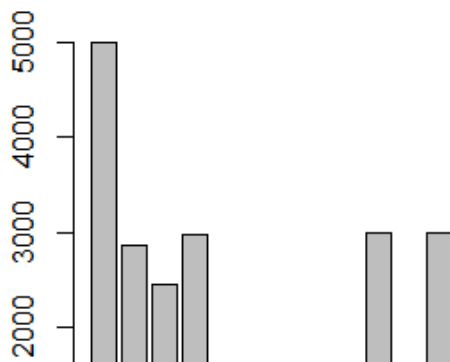
■ 옵션

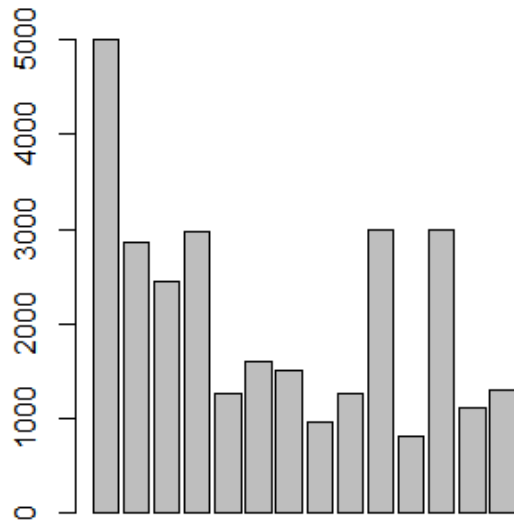
인수	설명
angle, density, col	막대를 칠하는 선분의 각도, 선분의 수, 선분의 색 지정
legend	오른쪽 상단에 범례 추가
names	각 막대의 라벨을 정하는 문자열 벡터를 지정
width	각 막대의 상대적인 폭을 벡터로 지정
space	각 막대사이의 간격을 지정
beside	TRUE를 지정하면 각각의 값마다 막대를 그림
horiz	TRUE를 지정하면 막대를 옆으로 눕혀서 그림

2. 예제

문제 106. Emp 테이블의 월급으로 기본적인 막대 그래프를 그리시오.

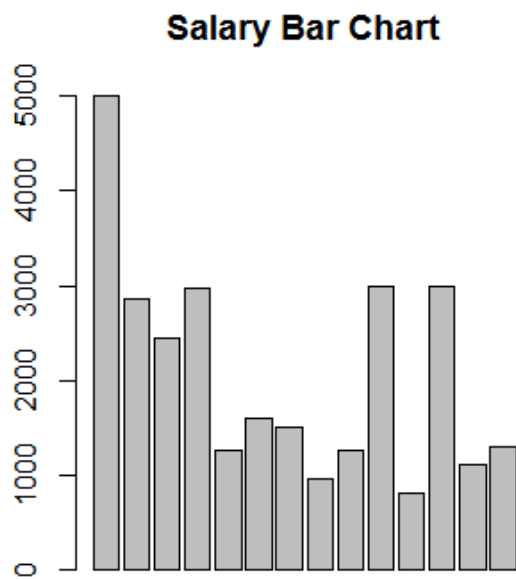
```
> barplot(emp$sal)
```





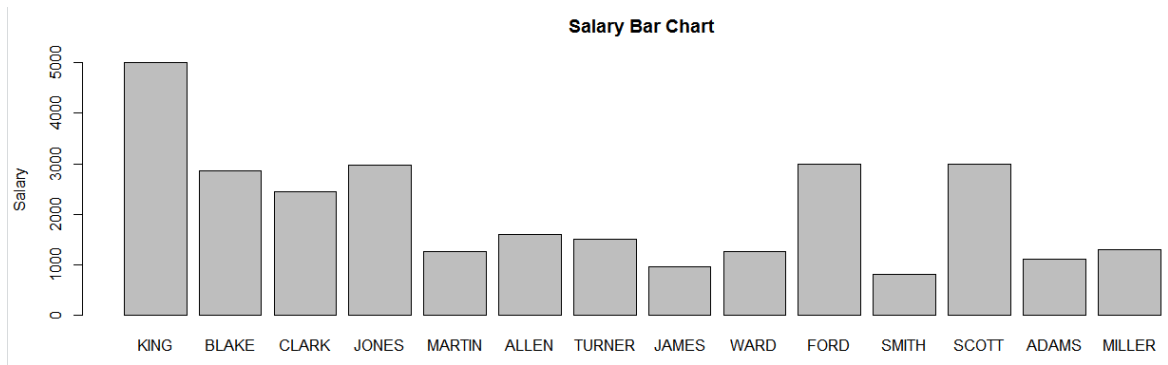
문제 107. 문제 106번 그래프에 제목을 Salary Bar Chart 라고 이름을 붙이시오.

```
> barplot(emp$sal, main='Salary Bar Chart')
```



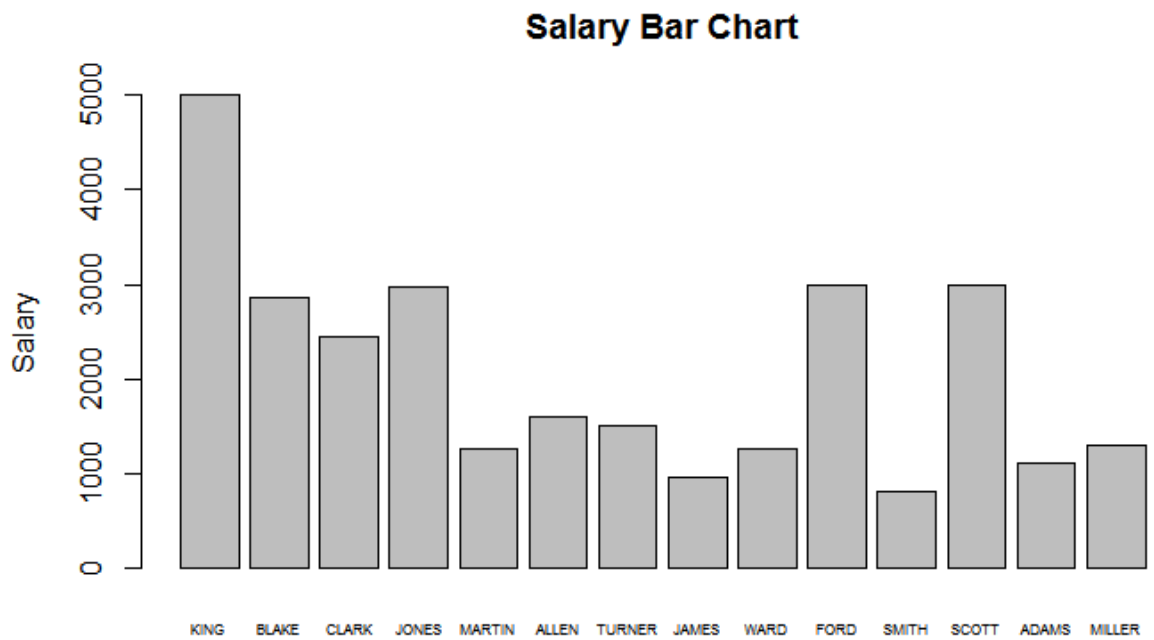
문제 108. 문제 107번 막대 그래프 x축에 사원이름을 출력 하시오.

```
> barplot(emp$sal, main='Salary Bar Chart',names.arg = emp$ename,ylab = 'Salary')
```



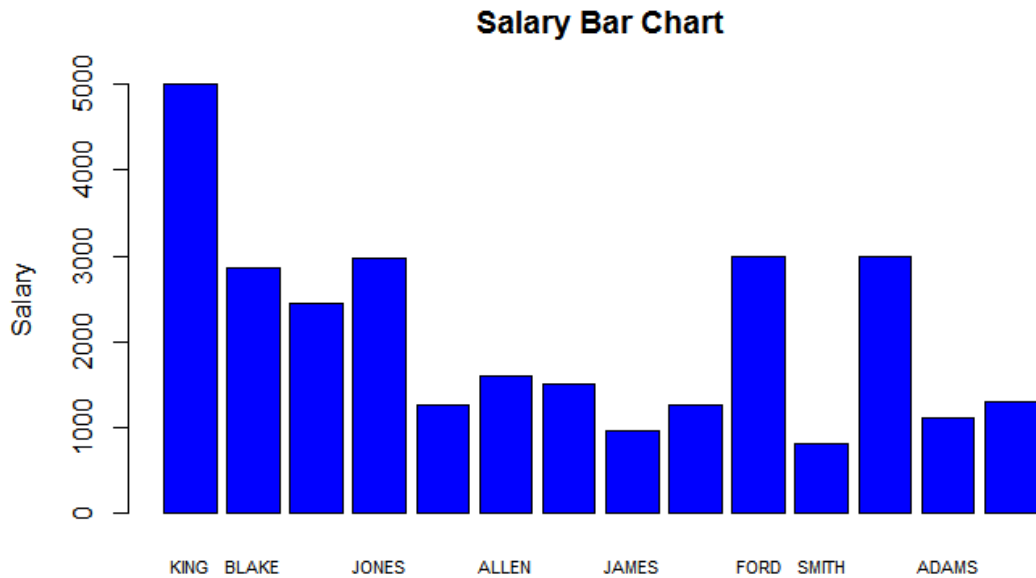
```
> barplot(emp$sal, main='Salary Bar Chart',names.arg = emp$ename ,ylab = 'Salary', cex.names=0.5)
```

*cex.names = 0.5 --글씨 크기를 바꿔준다.



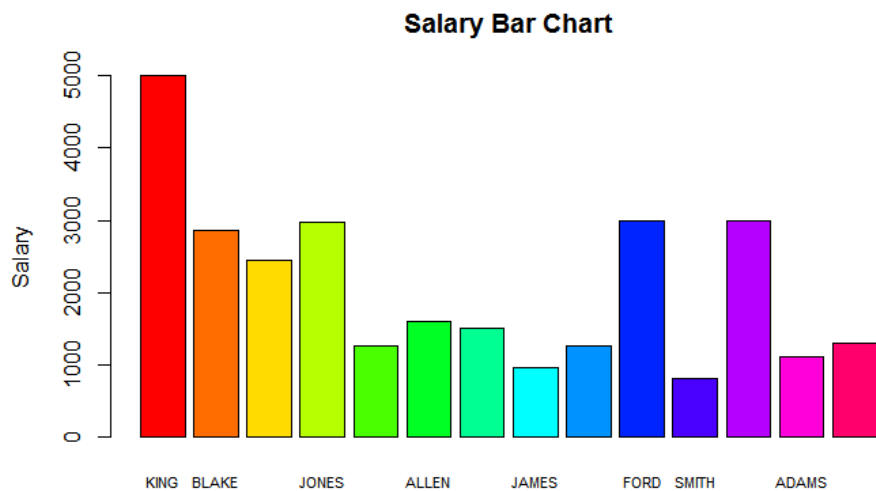
문제 109. 문제 108번 그래프에 색깔을 입히시오.

```
> barplot(emp$sal, main='Salary Bar Chart',names.arg = emp$ename,ylab = 'Salary', cex.names=0.7, col="blue")
```



문제 110. 문제 109번 그래프에 사원별로 유니크한 색깔을 입히시오.

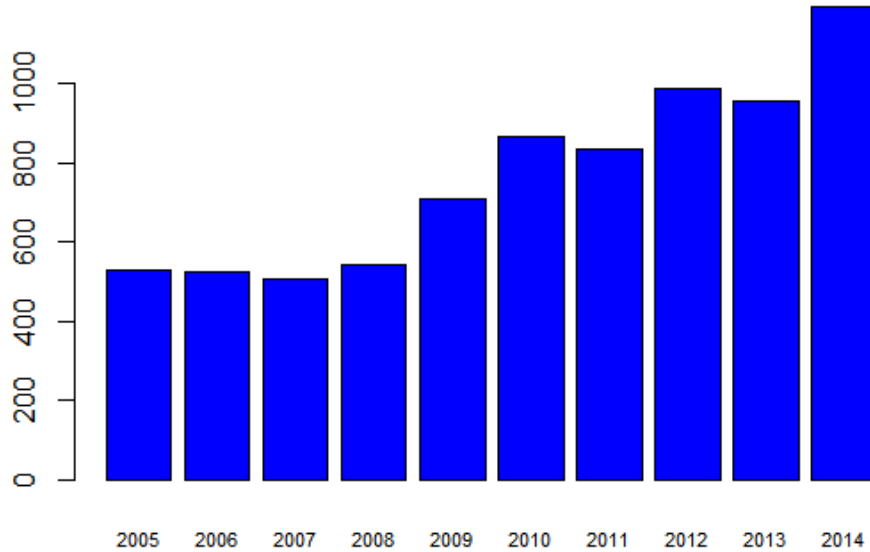
```
> barplot(emp$sal, main='Salary Bar Chart',names.arg = emp$ename,ylab = 'Salary', cex.names=0.7, col=rainbow(14))
```



문제 111. 치킨집 년도별 창업 건수를 막대 그래프로 시각화 하시오.

```
> barplot(create_cnt$치킨, main="년도별 치킨집 창업건수", names.arg=create_cnt$X, col=("blue"), ylim=c(0,1000),
cex.names=0.7)
```

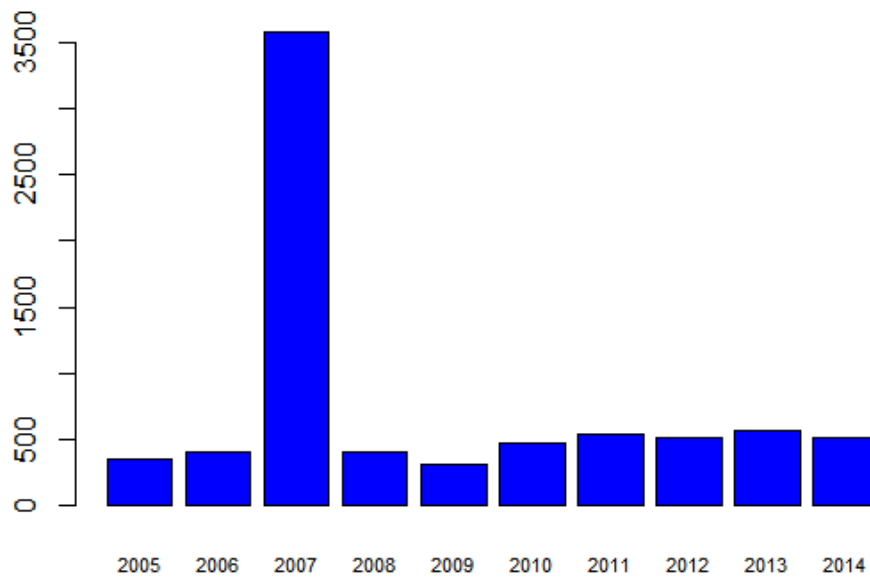

년도별 치킨집 창업건수



문제 112. 치킨집 년도별 폐업 건수를 막대 그래프로 시각화 하시오.

```
> barplot(drop_cnt$치킨, main="년도별 치킨집 폐업건수", names.arg=drop_cnt$X, col=("blue"), ylim=c(0,3500),
cex.names=0.7)
```

년도별 치킨집 폐업건수

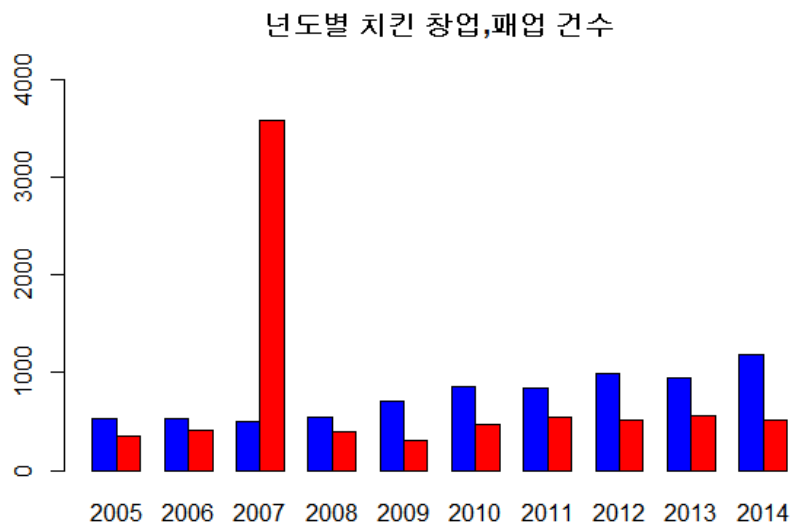


문제 113. 치킨집 년도별 창업건수, 폐업건수를 막대 그래프로 같이 나오게 하시오.

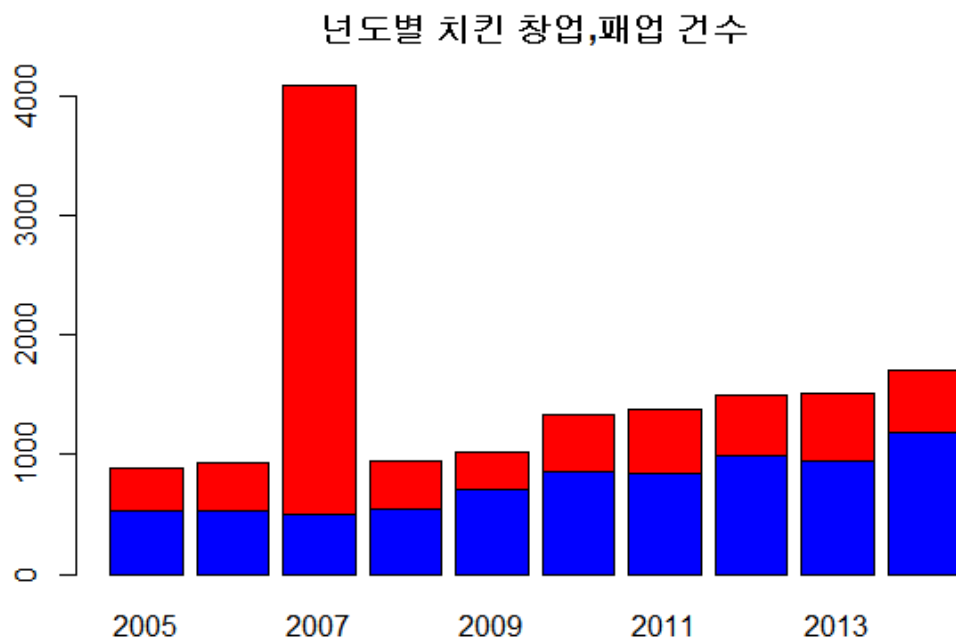
```
> x<-rbind(create_cnt$치킨집,drop_cnt$치킨집)
> x
```

```
> x<-rbind(create_cnt$치킨집,drop_cnt$치킨집)
> x
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  530  525  507  543  711  865  837  986  954  1193
[2,]  353  405 3579  399  308  464  538  510  560   511
```

```
> barplot(x,main="년도별 치킨 창업,폐업 건수",names.arg=create_cnt$X,col=c("blue","red"),ylim=c(0,4000), beside=T)
```



```
> barplot(x,main="년도별 치킨 창업,폐업 건수",names.arg=create_cnt$X,col=c("blue","red"),ylim=c(0,4000))
```

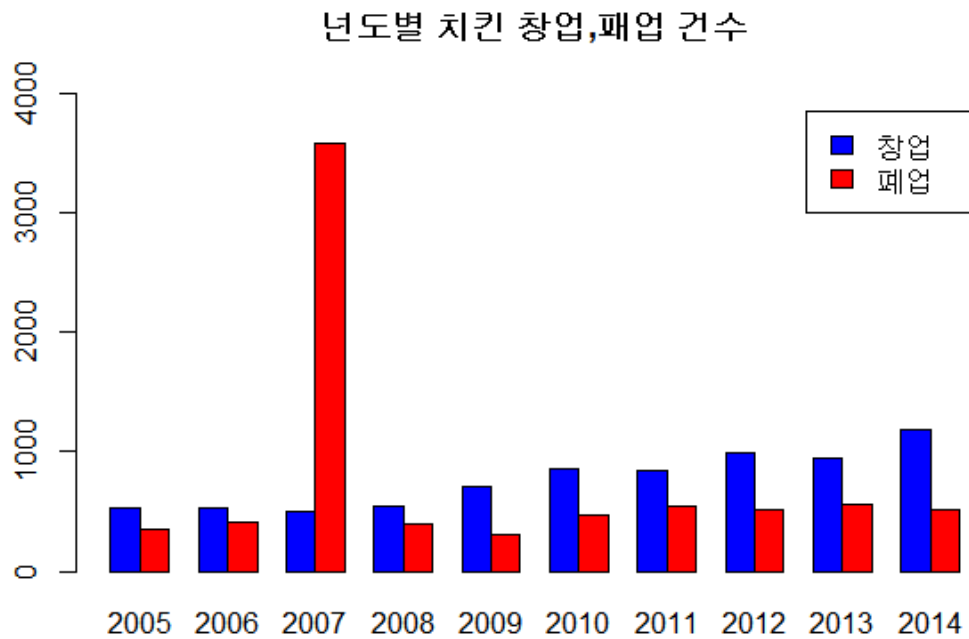


문제 114. 문제 113번에 legend를 달아서 파란색이 창업이고 빨간색이 폐업이다 라고 하시오.

```
> barplot(x,main="년도별 치킨 창업,폐업 건수",names.arg=create_cnt$X,col=c("blue","red"),ylim=c(0,4000), beside=T,
legend=c("창업","폐업"))
```

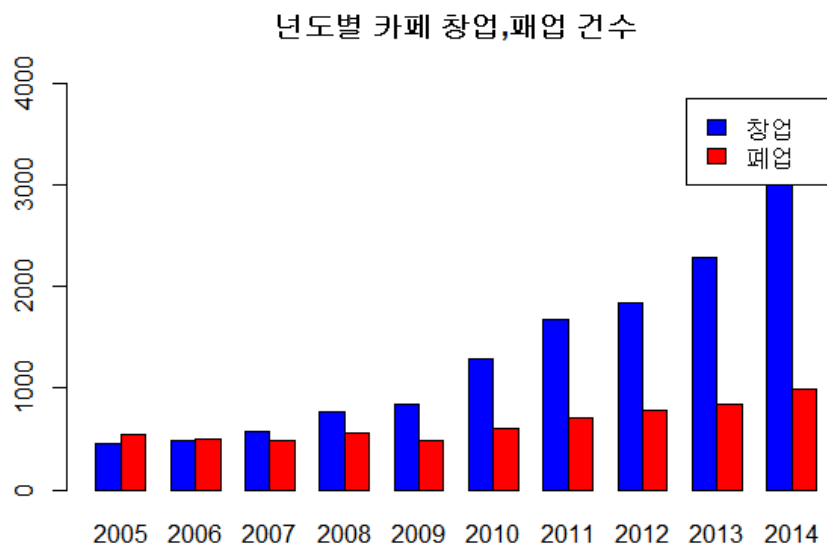
인 수	기 능
angle, density, col	막대를 칠하는 선분의 각도, 선분의 수, 선분의 색을 지정
legend	오른쪽 상단에 범례를 그림
names	각 막대의 라벨을 정하는 문자열 벡터를 지정
width	각 막대의 상대적인 폭을 벡터로 지정
space	각 막대 사이의 간격을 지정
beside	TRUE를 지정하면 각각의 값마다 막대를 그림
horiz	TRUE를 지정하면 막대를 옆으로 눕혀서 그림

년도별 치킨 창업,폐업 건수



문제 115. 카페(커피음료)가 얼마나 창업을 하고 얼마나 폐업을 하는지 막대 그래프로 시각화 하시오.

```
> x<-rbind(create_cnt$커피음료,drop_cnt$커피음료)
> barplot(x,main="년도별 카페 창업,폐업 건수",names.arg=create_cnt$X,col=c("blue","red"),ylim=c(0,4000), beside=T,
legend=c("창업","폐업"))
```



24. 원형 그래프 (pie)

2018년 5월 15일 화요일 오후 4:15

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 지도 그래프 & 소리 시각화
6. 워드 클라우드
7. 사분위수 그래프

■ 원형 그래프

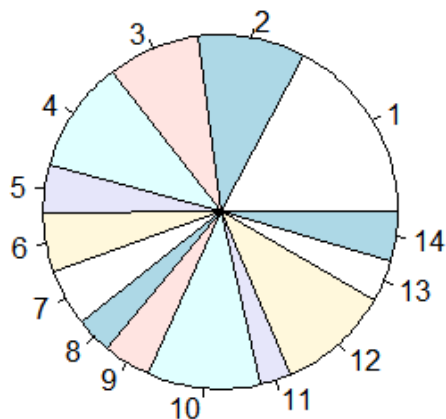
pie() : 파이 (pie) 모양의 차트 , 전체 값이 100이 되어야 하는 경우 서로 비교할 때 사용

***옵션**

인수	설명
Angle, density, col	Pie 부분을 구성하는 각도(angle), 수(density), 색상(col) 지정
labels	각 pie 부분의 이름을 지정하는 문자벡터 지정
radius	원형의 크기를 지정 ex. radius = 0.5
clockwise	시계방향(T), 반시계방향(F) 회전 여부 지정. [기본값 : 반시계]
Init.angle	시작되는 지점의 각도지정

문제 116. 사원 테이블의 월급으로 원형 그래프를 그리시오.

> pie(emp\$sal)

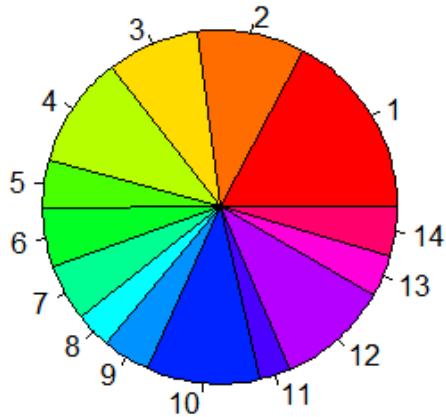


문제 117. 사원 테이블의 월급으로 원형 그래프를 그리시오.

문제 118. 문제 116번을 다시 수행하는데 제목을 Salary Pie Chart 라고 붙이시오.

```
> pie(emp$sal, col=rainbow(14) , main='Salary Pie Chart')
```

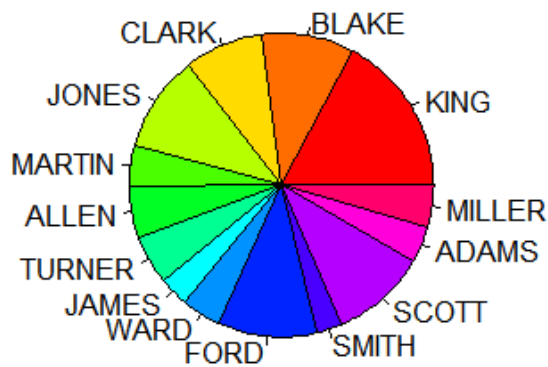
Salary Pie Chart



문제 119. 문제 118번을 다시 수행하는데 누구의 월급인지 이름이 출력되게 하시오.

```
> pie(emp$sal, col=rainbow(14),main='Salary Pie Chart', labels = emp$ename)
```

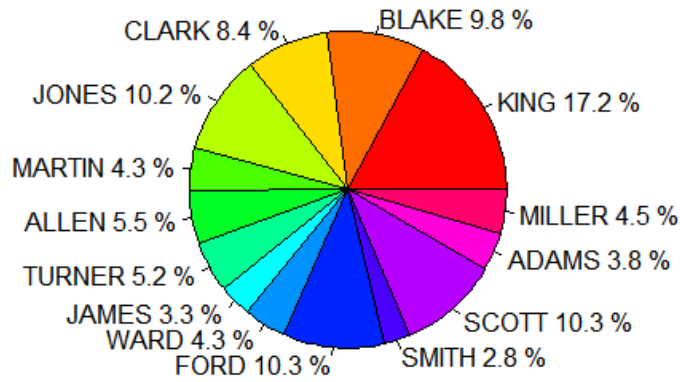
Salary Pie Chart



문제 120. 문제 119번 그래프에 월급의 비율을 붙여서 출력 하시오.

```
> sal_label2<-paste(emp$ename,round(emp$sal/sum(emp$sal)*100,1),"%")
> pie(emp$sal, col=rainbow(14),main='Salary Pie Chart',labels =sal_label2)
```

Salary Pie Chart



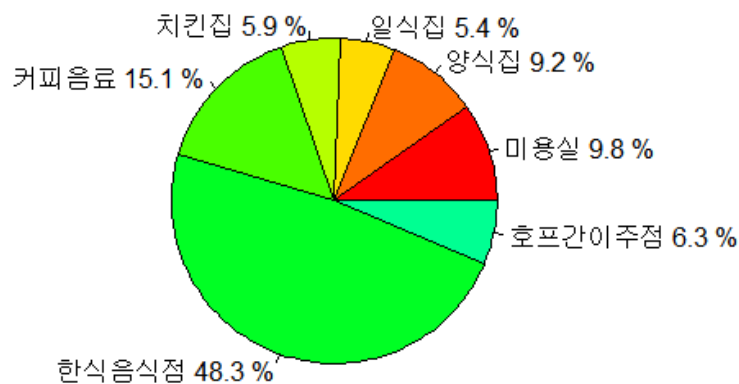
문제 121. 2014년도 업종별 창업 비율을 아래와 같이 원형 그래프로 그리시오.

```
> v<- create_cnt[create_cnt$X == 2014, -1]
> t(v)                                     -- 가로 세로를 변경 .

> v<- create_cnt[create_cnt$X == 2014, -1]
> v
   미용실 양식집 일식집 치킨집 커피음료 한식음식점 호프간이주점
10   1980   1870   1095   1193     3053     9772           1272
> t(v)
           10
미용실      1980
양식집      1870
일식집      1095
치킨집      1193
커피음료     3053
한식음식점   9772
호프간이주점 1272

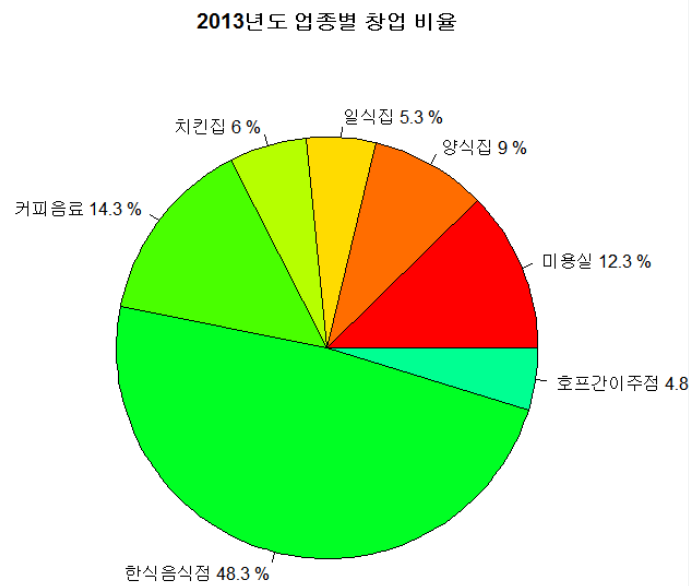
> cnt_label<-paste(colnames(v),round(v/sum(v)*100,1),'%')
> pie( t(v), col=rainbow(14),main='Salary Pie Chart',labels = cnt_label)
```

Salary Pie Chart



문제 122. 2013년도 업종별 창업 비율을 아래와 같이 원형 그래프로 그리시오.

문제 123. 2013년도 업종별 창업 비율을 아래와 같이 원형 그래프로 그리시오. 제목을 2013년도 창업 현황이라고 하시오.



```
> v<- create_cnt[create_cnt$v1 == 2013, -1]
> cnt_label<-paste(colnames(v),round(v/sum(v)*100,1),'%')
> pie(t(v), col=rainbow(14),main='Salary Pie Chart',labels = cnt_label)
```

문제 124. 문제 121의 코드를 가지고 함수를 생성하는데 아래와 같이 년도를 입력 받아 해당 년도의 원형 그래프가 그려지게 하시오.

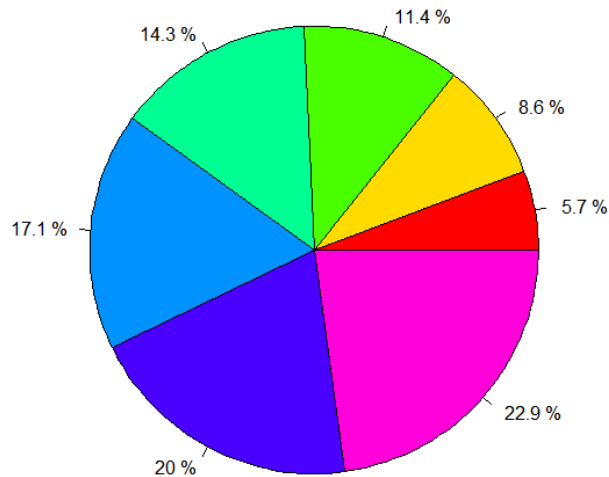
```
show_pie()
2014
```

```
show_pie <- function(){
  graphics.off()
  response <- readline(prompt='년도를 입력하세요~')

  x2<- create_cnt[create_cnt$v1 == response, (2:8)]
  pie(t(x2),col=rainbow(7))
  cnt_label <- round(x2/sum(x2)*100,1)
  cnt_label2<-paste(colnames(cnt_label),t(cnt_label),'%')
  pie(t(x2),col=rainbow(7),labels=cnt_label2, main=paste(response,'년도 업종별 창업현황'))
}
```

```
> show_pie()
년도를 입력하세요~2014
```

2014 년도 업종별 창업현황



문제 125. 아래와 같이 업종을 물어 보게 하고 업종을 입력하면 해당 업종의 창업, 폐업 현황이 막대그래프로 그려지는 함수를 생성 하시오.

```
show_var <- function(){
  graphics.off()
  response<- readline(prompt='업종을 입력하세요~')

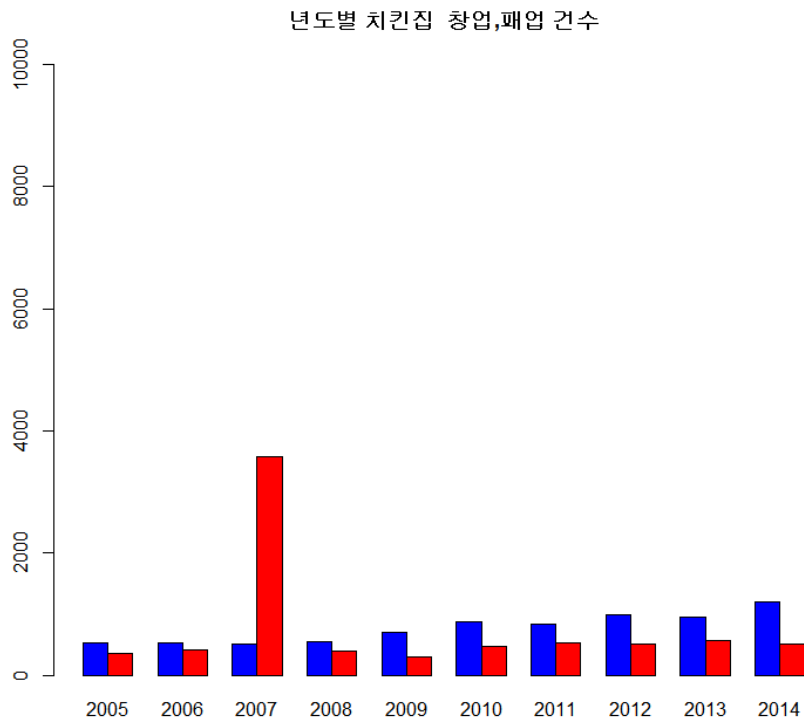
  x<-rbind(create_cnt[,response] , drop_cnt[,response])
  barplot(x,main=paste("년도별",response , " 창업,폐업 건수"),names.arg=create_cnt
$X,col=c("blue","red"),ylim=c(0,10000), beside=T)

}

> show_var()
업종을 입력하세요~치킨집

#데이터 프레임 함수 관련 오류가 발생해서 library 해체해주어야 실행된다...

detach(package:data.table,unload=TRUE)
```

문제 127. 아래의 스크립트를 수행하고 입력했을때의 값이 x7이라는 변수에 입력되게 하시오.

```
x7 <- menu(c("Yes","No"),title="Do yo want this?")

> x7 <- menu(c("Yes","No"),title="Do yo want this?")
Do yo want this?

1: Yes
2: No

선택: 2
> x7
[1] 2
```

문제 128. R의 menu 함수를 이용해서 아래와 같이 업종을 번호로 선택해서 막대그래프가 출력되게 하시오.

```
> show_bar2()
Do yo want this?

1: 미용실
2: 양식집
3: 일식집
4: 치킨집
5: 커피음료
6: 한식음식점
7: 호프간미주점
```

```
show_bar3 <- function(){
  graphics.off()
  response<- menu(colnames(create_cnt[-1]),title="Do yo want this?")

  print(response)
```

```

x<-rbind(create_cnt[,response+1] , drop_cnt[,response+1])
barplot(x,main=paste("년도 별",response , " 창업,패업 건수"),names.arg=create_cnt
$X,col=c("blue","red"),ylim=c(0,10000), beside=T)

}
# menu 에서 선택하면 response에 번호가 들어간다.
#create_cnt[, 번호] 는 번호 번째 열을 출력

```

#switch 함수를 이용한 방법

****switch 함수 (숫자, 숫자가 1일경우, 숫자가 2일경우,)**

```

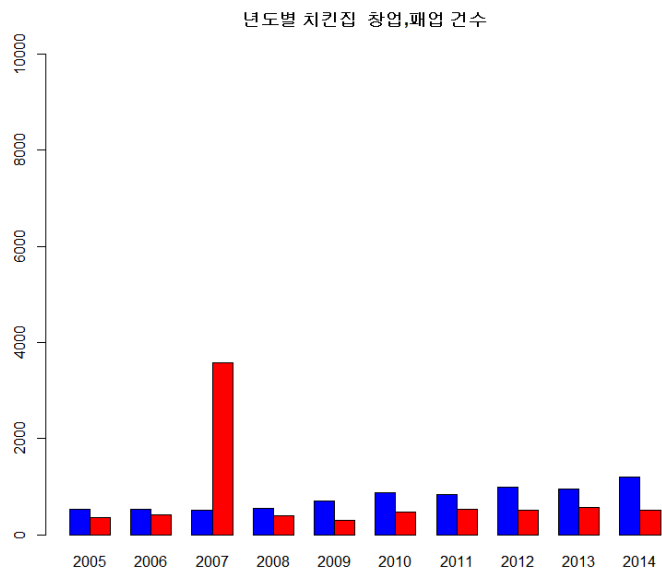
show_bar2 <- function(){
  graphics.off()
  response<- switch(menu(colnames(create_cnt[-1]),title="Do yo want this?"),"미용실","양식집","일식집","치킨집","커피
음료","한식음식점","호프간이주점")

  print(response)

  x<-rbind(create_cnt[,response] , drop_cnt[,response])
  barplot(x,main=paste("년도 별",response , " 창업,패업 건수"),names.arg=create_cnt
$X,col=c("blue","red"),ylim=c(0,10000), beside=T)

}

```



문제 130. 직업, 입사한 년도 (4자리), 직업별 입사한 년도별 토탈 월급을 출력 하시오.

```

emp<-read.csv("c:\\data\\emp.csv", header = T)
library(lubridate)
x<-tapply(emp$sal,list(emp$job,substr(emp$hiredate,1,4)),sum)
ifelse(is.na(x),0,x)

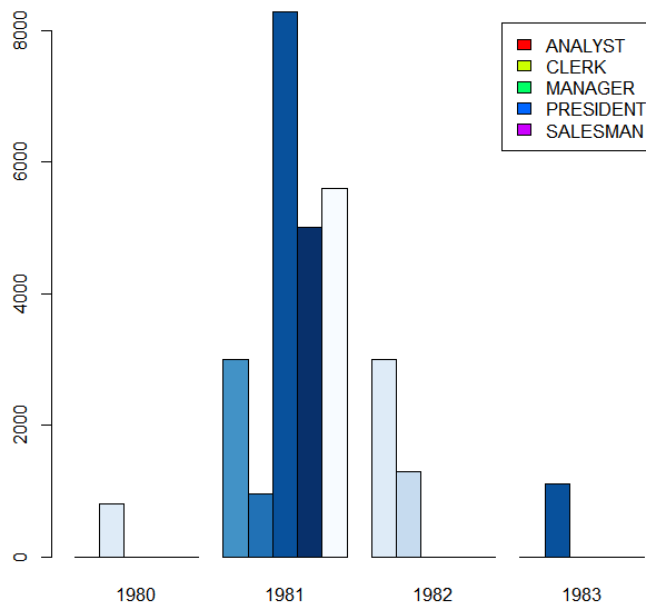
```

```
> ifelse(is.na(x),0,x)
      1980 1981 1982 1983
ANALYST      0 3000 3000    0
CLERK      800  950 1300 1100
MANAGER      0 8275    0    0
PRESIDENT    0 5000    0    0
SALESMAN     0 5600    0    0
```

문제 132. 문제 130번을 막대 그래프로 출력 하시오.

```
library(lubridate)
x<-tapply(emp$sal,list(emp$job,year(emp$hiredate)),sum)
y<-ifelse(is.na(x),0,x)

barplot(y,col=blues9,beside=T)
legend('topright',legend=rownames(y),fill=rainbow(5),inset=.02)
```



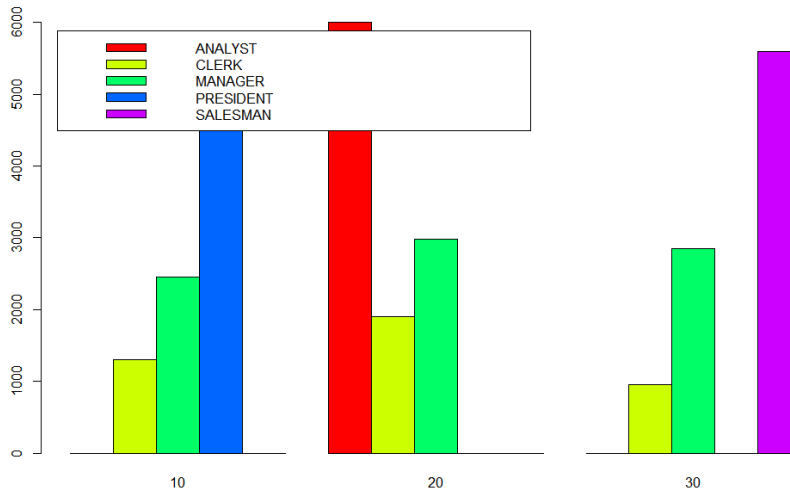
문제 133. 아래와 같이 막대 그래프를 그릴 컬럼을 물어보게 하고 컬럼을 각각 입력하면 막대 그래프가 그려지게 하시오.

```
show_emp_bar <- function(){

  response1<- readline(prompt = '가로가 될 컬럼명을 입력하세요.')
  response2<- readline(prompt = '세로가 될 컬럼명을 입력하세요.')

  x<-tapply(emp$sal,list(emp[,response2],emp[,response1]),sum)
  y<-ifelse(is.na(x),0,x)
  print(y)
  barplot(y, col=rainbow(5),beside=T)
  legend('topleft',legend=rownames(y),fill=rainbow(5),inset=.02)
}
```

```
> show_emp_bar()
가로가 될 컬럼명을 입력하세요. deptno
세로가 될 컬럼명을 입력하세요. job
      10    20    30
ANALYST      0 6000      0
CLERK      1300 1900    950
MANAGER      2450 2975 2850
PRESIDENT  5000      0      0
SALESMAN      0      0 5600
```



문제 134. 아래와 같이 막대 그래프를 그릴 컬럼을 물어 보게하고 컬럼을 각각 입력하면 토탈 월급에 대한 막대 그래프가 그려지게 하시오.

```
show_emp_bar2 <- function(){
  graphics.off()
  library(lubridate)
  res1<-switch(menu(c('job','mgr','year','deptno','grade'),title='가로가 될 컬럼명을 입력하세요'),
    emp$job,emp$mgr,year(emp$hiredate),emp$deptno,emp$grade)
  print(res1)
  res2<-switch(menu(c('job','mgr','year','deptno','grade'),title='세로가 될 컬럼명을 입력하세요'),
    emp$job,emp$mgr,year(emp$hiredate),emp$deptno,emp$grade)

  print(res2)
  x<-tapply(emp$sal,list(res2,res1),sum)
  y<-ifelse(is.na(x),0,x)
  barplot(y,col=blues9,beside=T)
  legend("topleft",rownames(y),fill=blues9)
}
```

#또다른 방법

```
show_emp_bar <- function() {

  graphics.off()

  x1 <- menu( colnames(emp), title = '가로가 될 컬럼을 선택하세요 ~ ')
  x2 <- menu( colnames(emp), title = '세로가 될 컬럼을 선택하세요 ~ ')
```

```

x1 <- colnames(emp)[x1]
x2 <- colnames(emp)[x2]

q <- tapply(emp$sal, list(emp[,x2], emp[,x1]), sum)
q[is.na(q)] <- 0
barplot(q, col = rainbow(nrow(q)), main = paste( x1, '별', x2, '의 월급총합' ), beside = T, ylim = c( 0, max(q)*1.5 ))
legend("topright", rownames(q),title = x2 ,inset = 0,fill = rainbow(nrow(q)),cex=0.8)
}

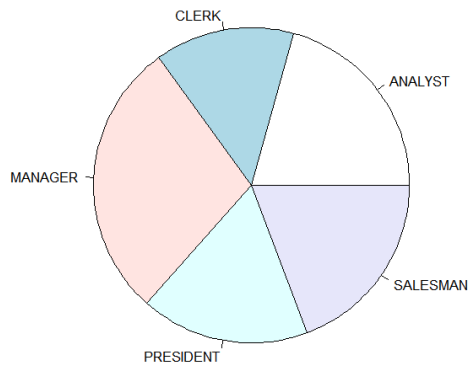
```

문제 135. 직업, 직업별 토탈 월급을 원형(pie) 그래프로 그리시오.

```

x<-tapply(emp$sal, emp$job,sum)
pie(x)

```

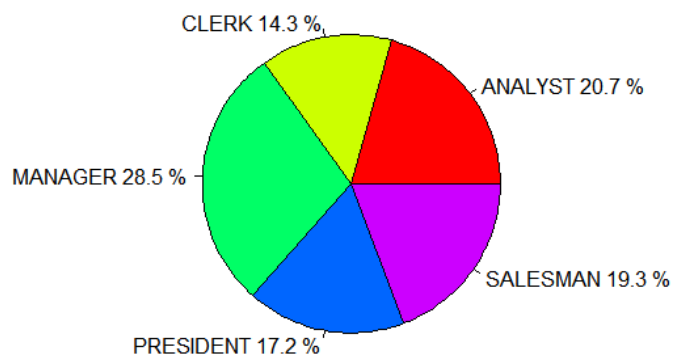


문제 136. 아래 그래프의 직업 옆에 비율을 표시 하시오.

```

x<-tapply(emp$sal, emp$job,sum)
label<-round(x/sum(x)*100,1)
label2<-paste(sort(unique(emp$job)),label)
pie(x,col=rainbow(nrow(x)),label=paste(label2,"%"))

```

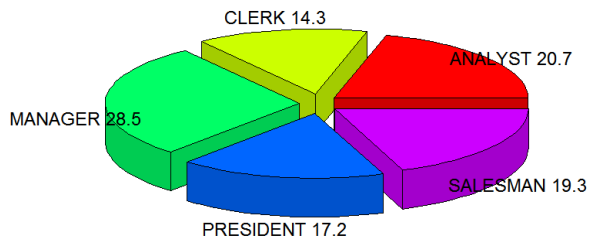


문제 137. 문제 136번 그래프를 3D그래프로 출력 하시오.

```

install.packages("plotrix")
library(plotrix)
pie3D(x,explode = 0.1, labels=label2)

```

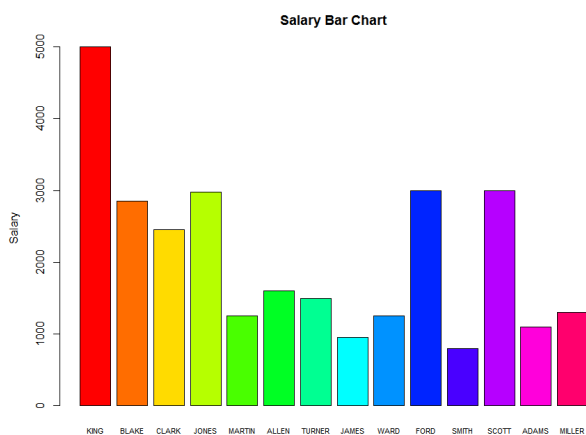


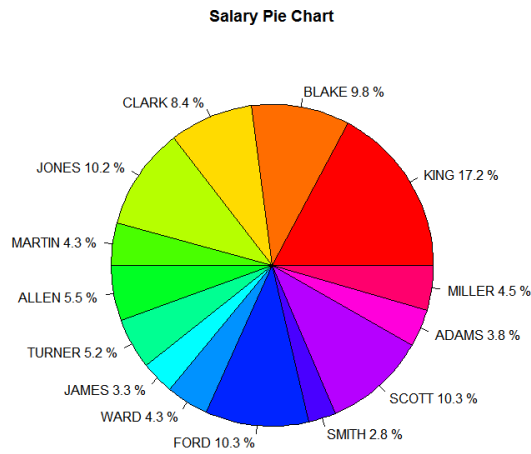
문제 138. Switch문으로 막대그래프와 원형그래프를 선택해서 출력할 수 있도록 하시오.

```
func <- function() {

  x1 <- menu( c("막대그래프","원형그래프") ,
             title ='원하는 그래프의 숫자를 선택하세요 ')

  switch(x1,
        barchart ={
          barplot(emp$sal , main="Salary Bar Chart",
                  names.arg = emp$ename, ylab="Salary",
                  cex.names=0.7 , col = rainbow(14) )
        },
        piechart ={
          sal_label <- round(emp$sal/sum(emp$sal) * 100,1)
          sal_label2 <- paste(emp$ename, sal_label, "%")
          pie(emp$sal , col=rainbow(14),
              main="Salary Pie Chart", labels = sal_label2)
        },
        {
          print('default')
        }
  )
}
```





문제 139. 그래프의 컬럼과 종류를 선택 받아 그래프가 그려지는 함수를 생성 하시오.

```
func <- function() {
  res1<- menu(colnames(emp), title='토달 값을 구할 컬럼번호 입력하세요~')
  res2<- menu(colnames(emp), title='그룹핑할 컬럼번호 입력하세요~')
  x1 <- menu( c("막대그래프","원형그래프") ,title ='원하는 그래프의 숫자를 선택하세요 ')

  r1<-colnames(emp)[res1]
  r2<-colnames(emp)[res2]

  q<-tapply(emp[,r1], emp[,r2],sum)

  switch(x1,
    {
      q[is.na(q)] <- 0
      barplot(q, col = rainbow(nrow(q)), main = paste( r2, '별', r1,'총합' ), beside = T, ylim = c(0,max(q)*1.4))
      legend("topright", rownames(q),title = paste(r2,' 구분' ),inset = 0,fill = rainbow(nrow(q)),cex=0.8)
    },
    {
      label<-paste(unique(emp[,r2]), round(q/sum(q) * 100,1),'%')
      pie(q,col=rainbow(nrow(q)),label=label,main = paste( r2, '별', r1,'총합' ))
    }
  )
}
```

```
> func()
토달 값을 구할 컬럼번호 입력하세요~
```

```
1: empno
2: ename
3: job
4: mgr
5: hiredate
6: sal
7: comm
8: deptno
```

```
선택: 6
그룹핑할 컬럼번호 입력하세요~
```

```
1: empno
2: ename
3: job
4: mgr
5: hiredate
6: sal
7: comm
```

> func()

토탈 값을 구할 컬럼번호 입력하세요~

1: empno
2: ename
3: job
4: mgr
5: hiredate
6: sal
7: comm
8: deptno

선택: 6

그룹핑할 컬럼번호 입력하세요~

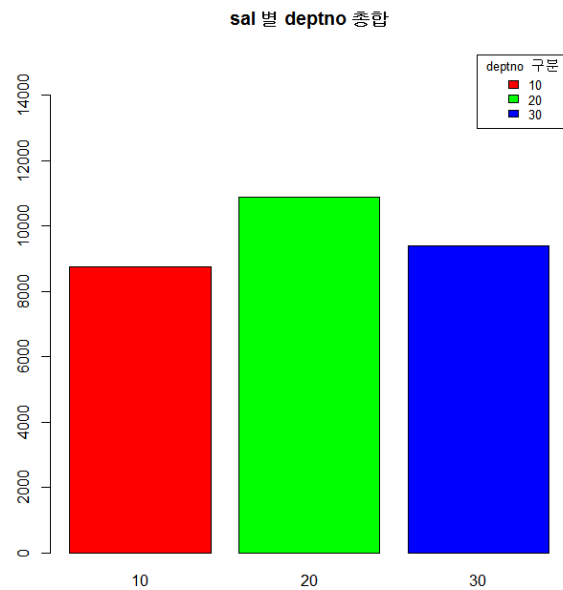
1: empno
2: ename
3: job
4: mgr
5: hiredate
6: sal
7: comm
8: deptno

선택: 8

원하는 그래프의 숫자를 선택하세요

1: 막대그래프
2: 원형그래프

선택: 1



25. 산포도 그래프 (plot)

2018년 5월 17일 목요일 오전 10:00

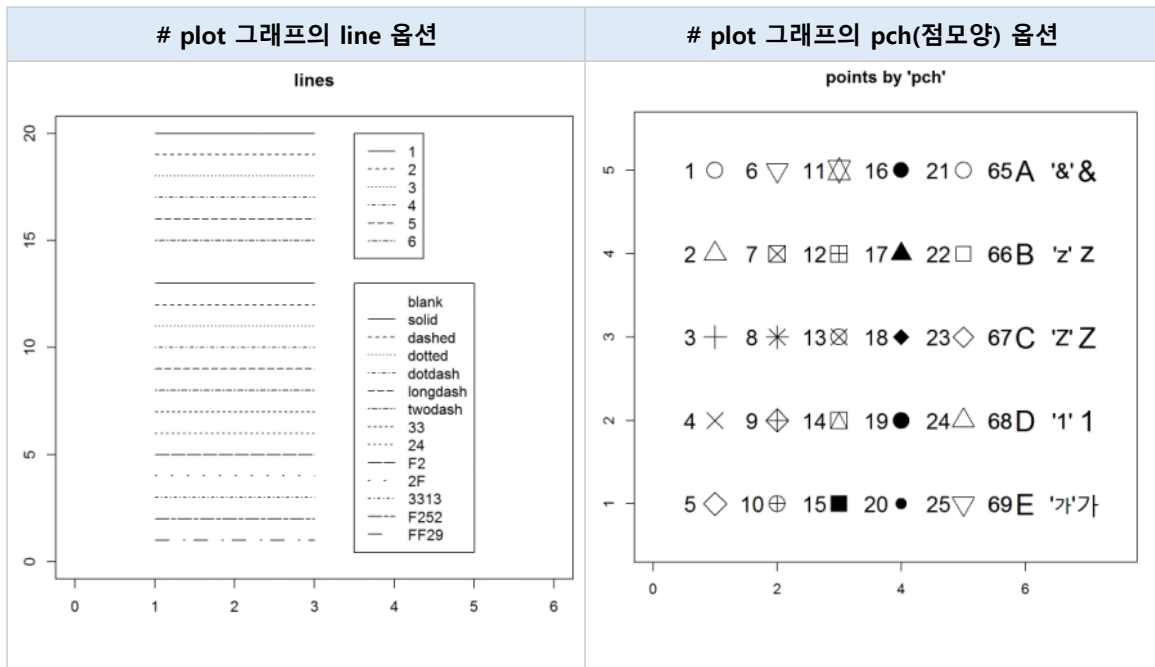
■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 소리를 시각화
6. 지도 그래프
7. 워드 클라우드
8. 사분위수 그래프

인 수	설 명
main = "메인 제목"	제목 설정
sub = "서브 제목"	서브 제목
xlab = "문자", ylab = "문자"	x, y축에 사용할 문자열을 지정합니다.
ann=F	x, y축 제목을 지정하지 않습니다.
tmap=2	제목 등에 사용되는 문자의 확대율 지정
axes=F	x, y축을 표시하지 않습니다.
axis	x, y축을 사용자의 지정값으로 표시합니다.
그래프 타입 선택	
type="p"	점 모양 그래프 (기본값)
type="l"	선 모양 그래프 (꺼은선 그래프)
type="b"	점과 선 모양 그래프
type="c"	"b"에서 점을 생략한 모양
type="o"	점과 선을 중첩해서 그린 그래프
type="h"	각 점에서 x축까지의 수직선 그래프
type="s"	왼쪽값을 기초로 계단모양으로 연결한 그래프
type="S"	오른쪽 값을 기초로 계단모양으로 연결한 그래프
type="n"	축만 그리고 그래프는 그리지 않습니다.

선의 모양 선택	
lty=0, lty="blank"	투명선
lty=1, lty="solid"	실선
lty=2, lty="dashed"	대쉬선
lty=3, lty="dotted"	점선
lty=4, lty="dotdash"	점선과 대쉬선

lty=5, lty="longdash"	긴 대쉬선
lty=6, lty="twodash"	2개의 대쉬선
색, 기호 등	
col=1, col="blue"	기호의 색지정, 1-검정, 2-빨강, 3-초록, 4-파랑, 5-연파랑, 6-보라, 7-노랑, 8-회색
pch=0, pch="문자"	점의 모양을 지정합니다
bg="blue"	그래프의 배경색 지정
lwd="숫자"	선을 그릴 때 선의 굵기를 지정
cex="숫자"	점이나 문자를 그릴 때 점이나 문자의 굵기를 지정



Plot 그래프의 type 옵션

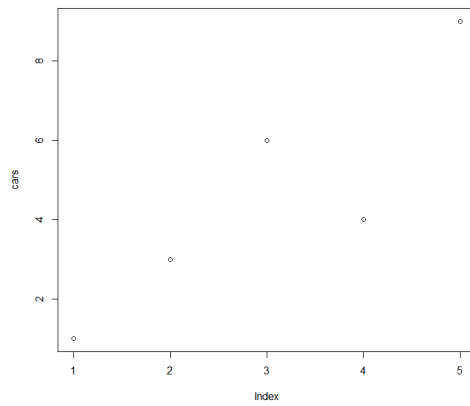
값	설명
"p"	점으로
"l"	선으로
"b"	점과 선 둘다 동시에
"o"	점과 선 둘다 동시에 (단 겹쳐짐 : overplotted)
"h"	히스토그램과 비슷한 형태로 (histogram)
"s"	계단 모양으로 (stair steps)
"S"	계단모양으로 (upper stair steps)
"n"	좌표찍지 않음

1. 예제

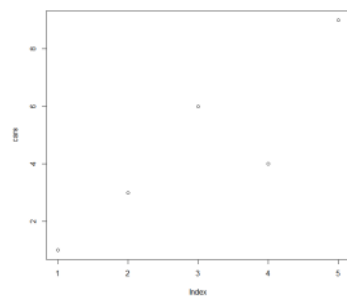
예제	아래의 점(plot) 그래프를 그리시오.
----	------------------------

```
graphics.off()
cars<-c(1,3,6,4,9)
cars

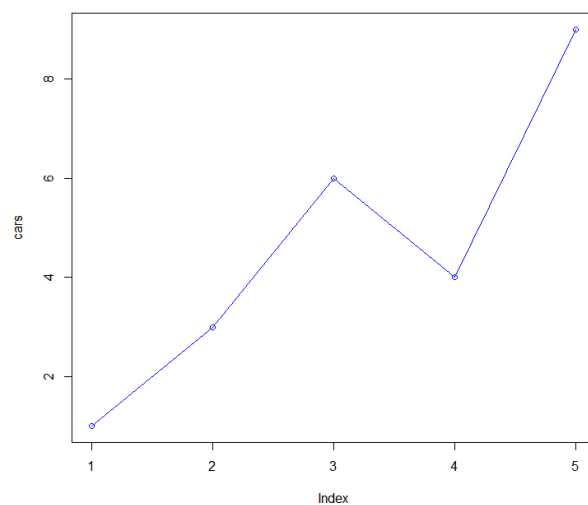
plot(cars)
```



문제 142. 아래의 그래프에 파란색선을 그리시오.



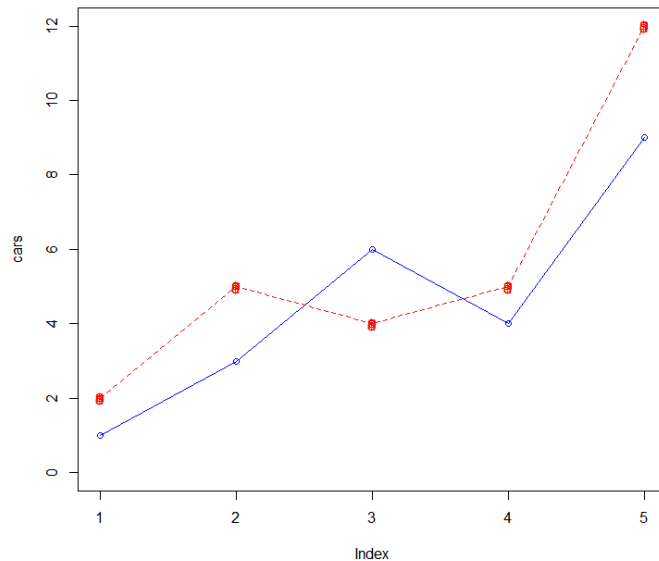
```
plot(cars, type="o", col="blue")
```



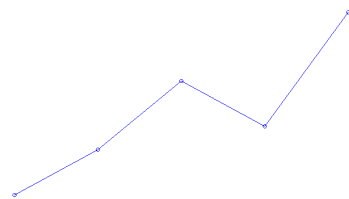
문제 143. 차와 트럭의 팔린 댓수를 라인 그래프로 시각화 하시오.

```
graphics.off()
cars<-c(1,3,6,4,9)
truck<-c(2,5,4,5,12)
plot(cars,type = "o", col="blue", ylim=c(0,12))
```

```
lines(truck, type="o",pch=36, lty=2, col="red")      # type="o" 라인을 그린다 # pch 점모양 옵션
```



문제 144. 아래의 그래프를 예시의 순서대로 시각화 작업을 하시오.



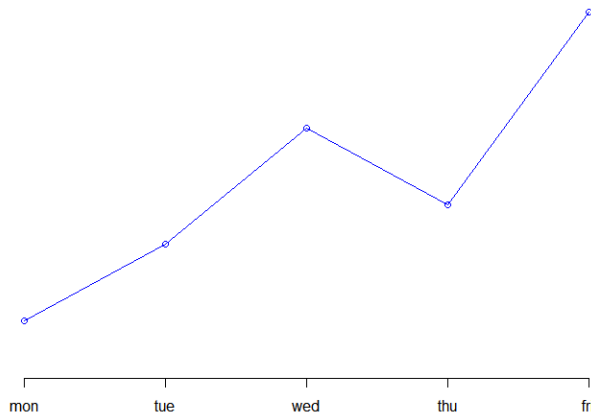
```
graphics.off()
cars<-c(1,3,6,4,9)
trucks<-c(2,5,4,5,12)
g_range<-range(0,cars,trucks)
g_range
plot(cars,type='o',col='blue',ylim=g_range, axes=F,ann=F)
```

```
# axes : 레이아웃 박스를 나타낼지 지정하는 옵션
# ann : x축, y축 이름을 출력할지 지정하는 옵션
```

----> 그래프가 켜져있는 상태에서 아래 명령어를 실행해보자.

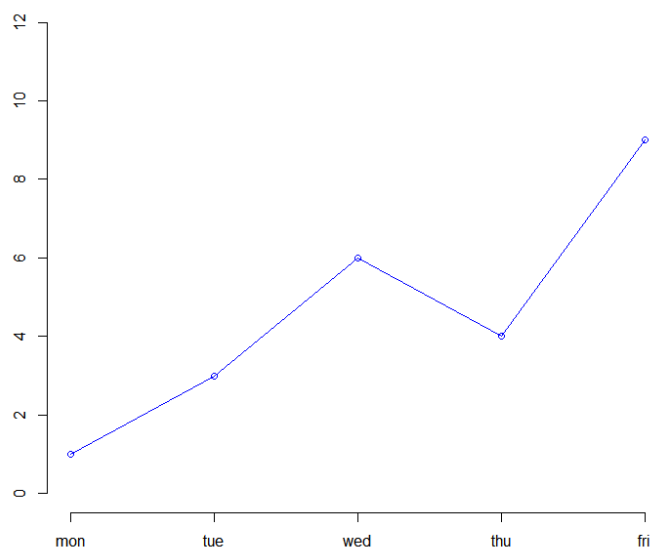
```
> axis(1,at=1:5, lab=c("mon","tue","wed","thu","fri"))      # x축과 이름이 같이 생긴다.
```





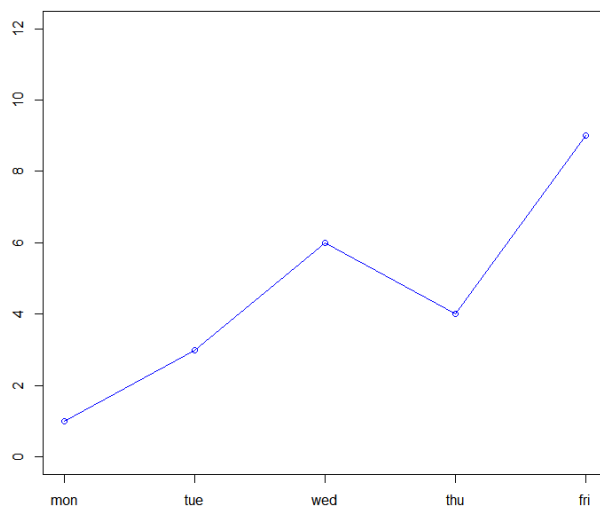
----> 그래프가 커져있는 상태에서 아래 명령어를 실행해보자.

```
> axis(2) #y축이 생긴다
```

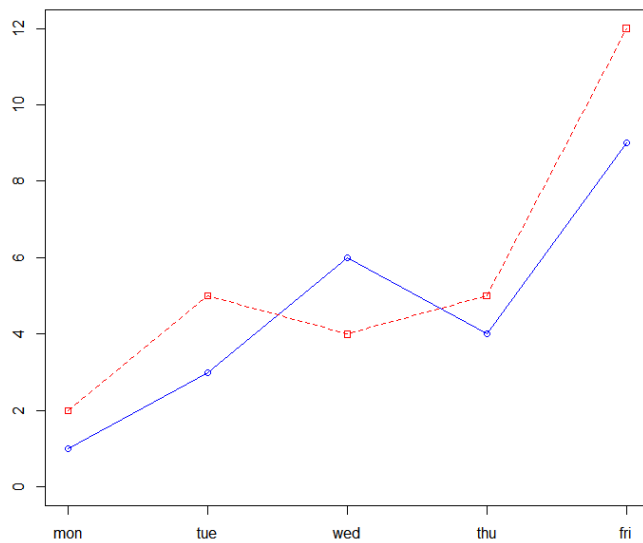


----> 그래프가 커져있는 상태에서 아래 명령어를 실행해보자.

```
> box() # 박스 레이아웃이 생긴다.
```

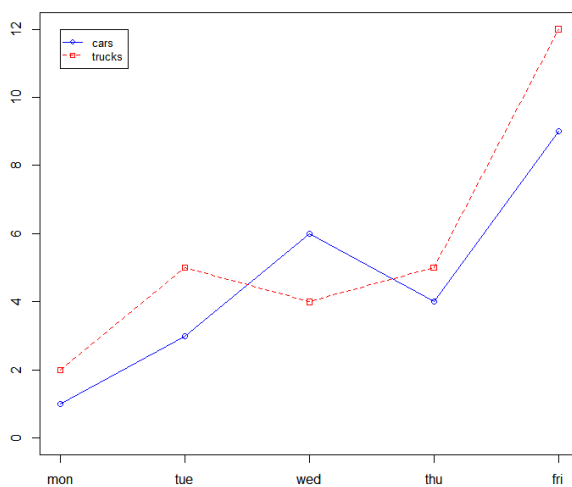


```
> lines(trucks, type = 'o', pch=22, lty=2, col='red')
```



```
> legend(1,12,c("cars","trucks"), col=c("blue","red"),cex=0.8, pch=21:22 , lty=1:2)
```

```
# cex : 글씨크기
# pch : 점모양 ( 21 : 동그라미 , 22: 네모 )
# lty : 선타입 ( 1: 직선 , 2: 점선 )
```



문제 145.	2018년 1월달 우리나라 지하철 총 승차 인원수 데이터를 R로 로드하고 1월달 데이터를 일별로 정렬해서 출력 하시오.
----------------	--

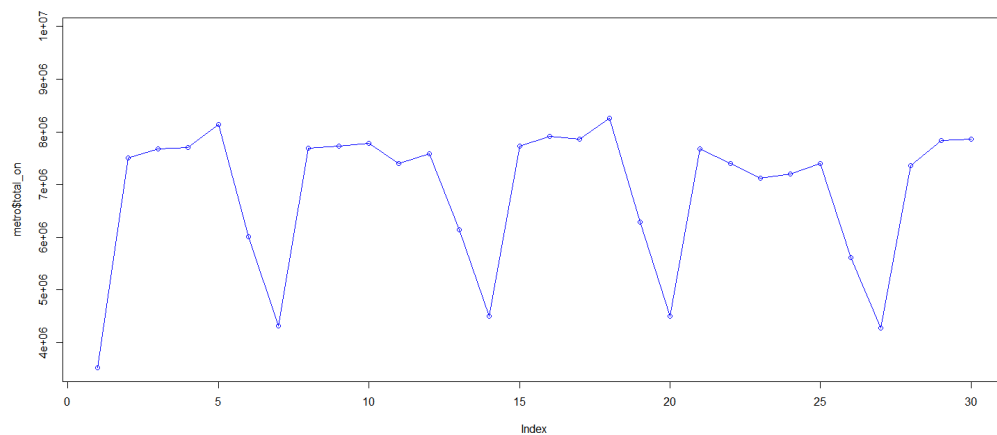
```
metro <- read.csv("c:\wwwdata\wwwmetro.csv", header = T)
metro <- na.omit(metro)
colnames(metro)<-tolower(colnames(metro))
metro[order(metro$usedate,decreasing = F),]
metro<-metro[order(metro$usedate,decreasing = F),]
```

```
# na값을 없앤다.
#컬럼명 소문자로 변경
#order 함수를 사용해서 정렬
```

	usedate	total_on
3	2018-01-01	3520929
1	2018-01-02	7505016
4	2018-01-03	7681469
7	2018-01-04	7705415
19	2018-01-05	8134256
2	2018-01-06	6015107
13	2018-01-07	4312642
14	2018-01-08	7685881
8	2018-01-09	7724445
26	2018-01-10	7786699
9	2018-01-11	7400556
10	2018-01-12	7581428
27	2018-01-13	6144612
15	2018-01-14	4497583
5	2018-01-15	7723671

문제 146. 2018년 1월달 우리나라 지하철 총 승차 인원수 데이터를 plot 그래프로 시각화 하시오.

```
plot(metro$total_on,type='o',col='blue', ylim=c(min(metro$total_on),max(metro$total_on)*1.2))
```



문제 147. Sql 포트폴리오의 데이터를 사용해서 시각화를 하시오.

```
rst123<-read.csv("c:\\data\\rst123.csv",header = T)
```

```
colnames(rst123)<-c('월', '평균미세먼지','상승률','순위')
```

```
graphics.off()
```

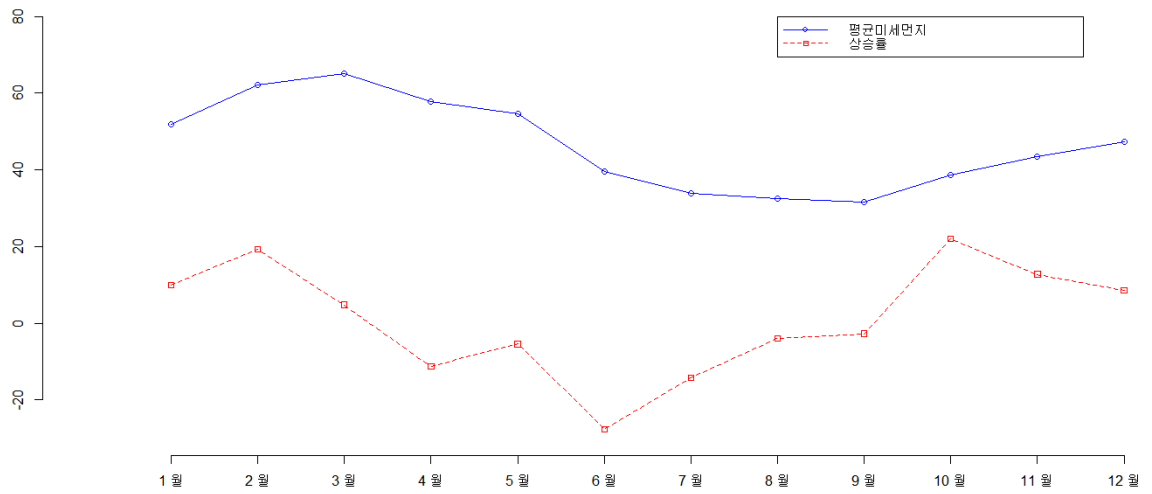
```
plot(rst123$평균미세먼지,type='o',col='blue', ylim=c(-30,80), xlim=c(0,12), main='미세먼지 농도 및 상승률 ',axes=F, ann=F)
```

```
lines(rst123$상승률, type = 'o', pch=22,lty=2,col='red')
```

```
legend(8,80,c("평균미세먼지","상승률"), col=c("blue","red"),cex=0.8, pch=21:22 , lty=1:2)
```

```
axis(1,at=1:12, lab=paste(1:12,'월'))
```

```
axis(2)
```

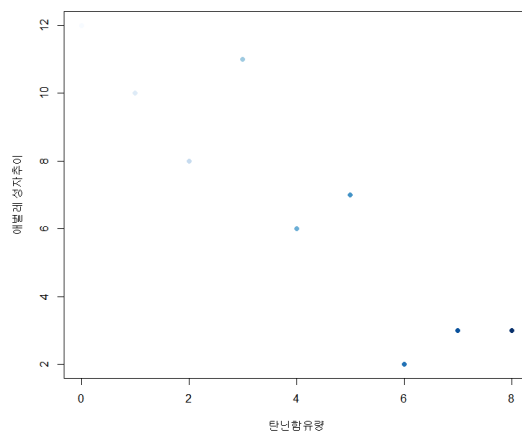


문제 148. 사료의 탄닌 함량 포함에 따른 애벌레 성장 추이에 관한 lavar.csv를 내려받고 탄닌 함량과 성장률간의 상관관계가 어떻게 되는지 시각화하고 상관계수를 구하시오.

```
larva<-read.csv('c:\wwdata\ww\larva.csv',header = T)
larva<-na.omit(larva)
```

```
plot(larva$tannin,larva$growth,col=blues9, pch=16, xlab='탄닌함유량', ylab='애벌레 성장추이')
cor(larva$tannin,larva$growth)
```

```
> cor(larva$tannin,larva$growth)
[1] -0.9031408
```



문제 149. 아래와 같이 함수를 실행하면 x축과 y축 컬럼을 각각 물어보게 하고 산포도 그래프가 그려지게 하시오.

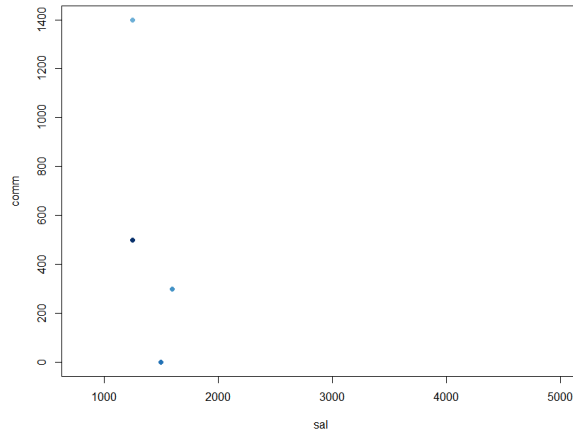
```
plot_func<-function(){

  res1<- readline(prompt='x축 컬럼명을 입력 하세요.')
  res2<- readline(prompt='y축 컬럼명을 입력 하세요.')

  x<-emp[,res1]
  y<-emp[,res2]
```



```
plot(x,y, col=blues9, pch=16, xlab = res1, ylab = res2)
}
```



문제 150. 테이블(변수명)을 먼저 물어보게 하고 변수에서 컬럼명을 추출해서 x축, y축을 물어볼 때 번호를 선택하게 하시오.

```
plot_func<-function(){

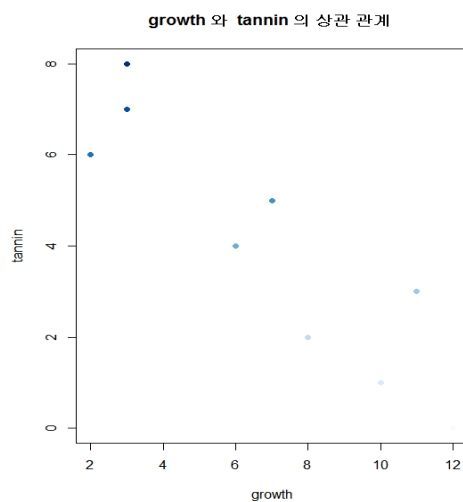
  graphics.off()
  # get(변수명) : 변수안에 들어있는 값을 가져옴

  v_name <- readline(prompt = '테이블명 입력 : ')

  v_name <- get(v_name)

  x <- menu(colnames(v_name), title='x축 컬럼명 선택 : ')
  y <- menu(colnames(v_name), title='y축 컬럼명 선택 : ')

  plot(v_name[,x],v_name[,y],pch=16, col=blues9,xlab = colnames(v_name)[x] ,ylab = colnames(v_name)[y],main =
  paste(colnames(v_name)[x],'와 ',colnames(v_name)[y],'의 상관 관계 '))
}
```



문제 151. 원형그래프와 막대그래프를 자동으로 그리는 함수를 아래와 같이 테이블을 먼저 물어보게 코드를 수정

```
func_151 <- function() {

  res0<- get(readline(prompt = '테이블명 입력 : '))
  res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼번호 입력하세요~')
  res2<- menu(colnames(res0), title='그룹핑할 컬럼번호 입력하세요~')
  x1 <- menu( c("막대그래프","원형그래프") ,title = '원하는 그래프의 숫자를 선택하세요 ')

  r1<-colnames(res0)[res1]
  r2<-colnames(res0)[res2]

  q<-tapply(res0[,r1], res0[,r2],sum)

  switch(x1,
    {
      q[is.na(q)] <- 0
      barplot(q, col = rainbow(nrow(q)), main = paste( r2, '별', r1,'총합' ), beside = T, ylim = c(0,max(q)*1.4))
      legend("topright", rownames(q),title = paste(r2,' 구분') ,inset = 0,fill = rainbow(nrow(q)),cex=0.8)
    },
    {
      label<-paste(unique(emp[,r2]), round(q/sum(q) * 100,1),'%')
      pie(q,col=rainbow(nrow(q)),label=label,main = paste( r2, '별', r1,'총합' ))
    }
  )
}
```

26. 구글에서 제공하는 그래프

2018년 5월 17일 목요일 오후 4:04

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
- 4. 구글에서 제공하는 그래프**
5. 지도 그래프 & 소리 시각화
6. 워드 클라우드
7. 사분위수 그래프

구글 그래프를 사용하기 위해선 --> **googleVis** 패키지 사용

```
install.packages("googleVis")  
library(googleVis)
```

1. 예제

문제 152. 조인을 써서 이름을 출력하고 관리자의 이름을 출력 하시오.

SQL문

```
SELECT e.ename, m.ename  
  From emp e, emp m  
  Where e.mgr = m.empno;
```

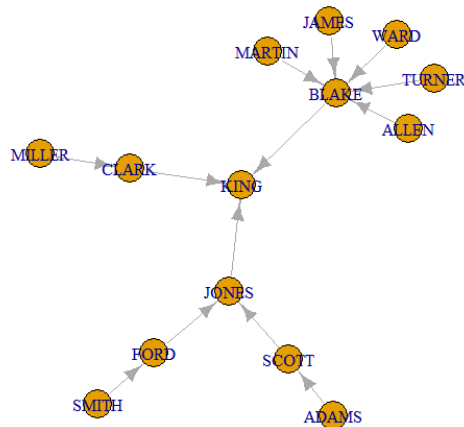
```
m<-merge(emp,emp,by.x="mgr",by.y="empno")  
m[,c("ename.x","ename.y")]
```

```
> m[,c("ename.x","ename.y")]  
  ename.x ename.y  
1    FORD   JONES  
2   SCOTT   JONES  
3  MARTIN   BLAKE  
4   ALLEN   BLAKE  
5  TURNER   BLAKE  
6   JAMES   BLAKE  
7    WARD   BLAKE  
8  MILLER   CLARK  
9   ADAMS   SCOTT  
10  BLAKE    KING  
11  CLARK    KING  
12  JONES    KING  
13  SMITH    FORD
```

문제 153. 문제 152번 결과를 가지고 사원 테이블의 조직도를 그리시오.

```
install.packages("igraph")
library(igraph)
```

```
m<-merge(emp,emp,by.x="mgr",by.y="empno")
k<-m[,c("ename.x","ename.y")]
b<-graph.data.frame(k,directed = T)
plot(b)
```

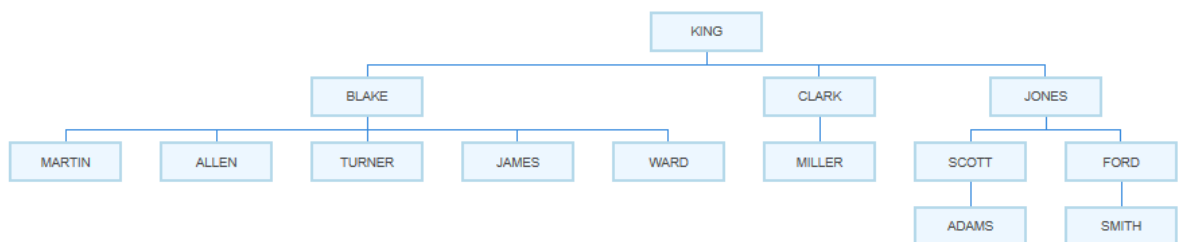


문제 154. 문제 153번의 시각화 결과를 구글의 googleVis를 이용해서 emp 테이블의 관계도를 시각화 하시오.

```
install.packages("googleVis")
library(googleVis)
```

```
a <- merge(emp,emp, by.x="empno",by.y="mgr", all.y=T)

org <- gvisOrgChart(a, idvar="ename.y",parentvar="ename.x",
  options=list(width=600, height=250, size='middle',allowCollapse=T)) #allowCollapse : 접기 가능
plot(org)
```



문제 155. 아래와 같이 함수를 실행하면 바로 emp 테이블의 조직도가 구글 그래프로 그려지게 하시오.

```
emp_org <- function(){

  a <- merge(emp,emp, by.x="empno",by.y="mgr", all.y=T)
```

```
org <- gvisOrgChart(a, idvar="ename.y",parentvar="ename.x",
  options=list(width=600, height=250, size='middle',allowCollapse=T)) #allowCollapse : 접기 가능
plot(org)

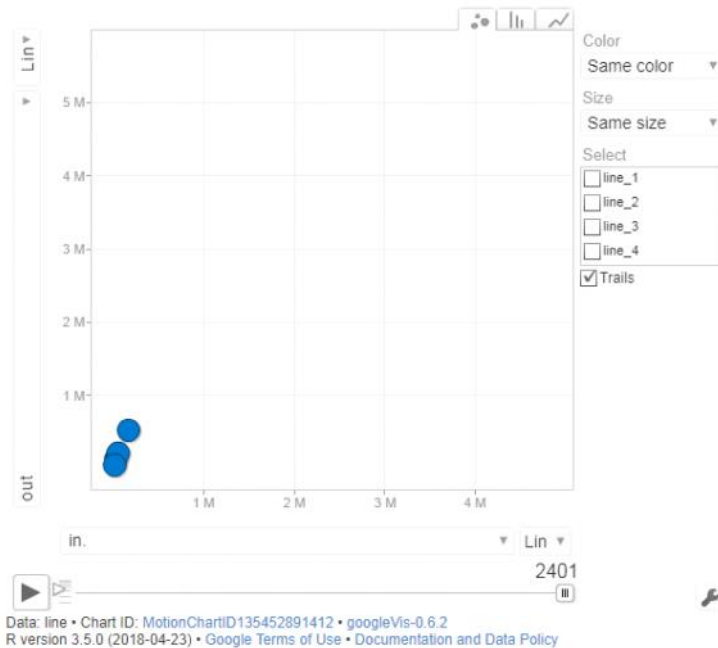
}
```

문제 156. 지하철 1-4호선 승하차 승객수.csv를 R로 로드해서 line no 컬럼과 time 컬럼을 이용해서 구글 모션차트를 그리시오.

```
line<-read.csv(file.choose(),header = T)

t1<-gvisMotionChart(line,idvar = "line_no",timevar = "time")

plot(t1)
```



문제 157. 지하철 5-8호선 승하차 승객수.csv를 R로 로드해서 line no 컬럼과 time 컬럼을 이용해서 구글 모션차트를 그리시오.

```
line2<-read.csv(file.choose(),header = T)

colnames(line2)<-c("line_no","time","in_cnt","out_cnt")

t2<-gvisMotionChart(line2,idvar = "line_no",timevar = "time")

plot(t2)
```

문제 158. 부서위치, 부서위치별 토탈 월급을 세로로 출력하고 막대 그리프를 만드시오.

```
emp<-read.csv("c:\\data\\emp.csv",header = T)
dept<-read.csv("c:\\data\\dept.csv",header = T)
```

```
x<-merge(emp,dept, by="deptno",all.y=T)
x2<-aggregate(x$sal~x$loc,x,sum)
```

x2 # na값을 가진 BOSTON은 출력되지 않는다. ->aggregate에 na.action 옵션 추가하자!

```
x2<-aggregate(x$sal~x$loc,x,sum, na.action = na.pass)
x2
```

```
x2[is.na(x2)==T] <-0
names(x2)<-c("loc","sumsal")
x2
```

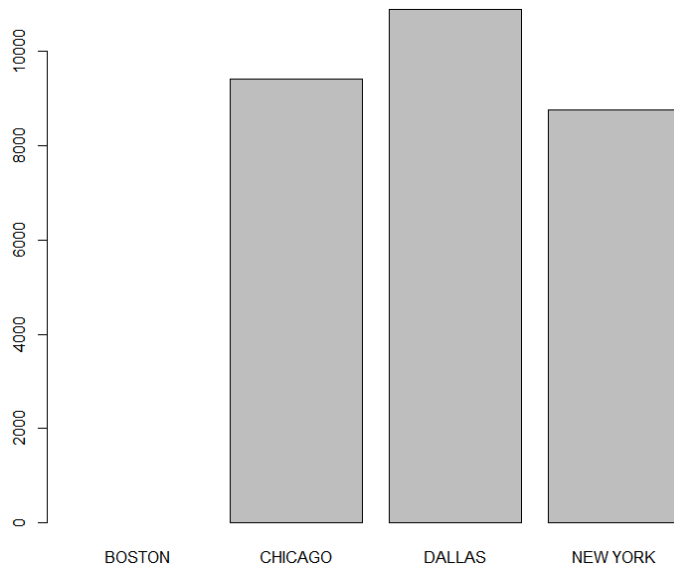
```
> barplot(x2)
Error in barplot.default(x2) :
  'height'는 반드시 벡터 또는 행렬이어야 합니다
```

막대그래프는 가로로 출력해서 그리는게 편하다. (**tapply** 사용)
 굳이 출력하려면 bar(x2\$sal) 로도 가능하다.

```
> x2<-aggregate(x$sal~x$loc,x,sum)
> x2
      x$loc x$sal
1 CHICAGO  9400
2 DALLAS 10875
3 NEW YORK  8750
> x2<-aggregate(x$sal~x$loc,x,sum, na.action = na.pass)
> x2
      x$loc x$sal
1  BOSTON    NA
2 CHICAGO  9400
3 DALLAS 10875
4 NEW YORK  8750
> x2[is.na(x2)==T] <-0
> names(x2)<-c("loc","sumsal")
> x2
      loc sumsal
1  BOSTON      0
2 CHICAGO  9400
3 DALLAS 10875
4 NEW YORK  8750
```

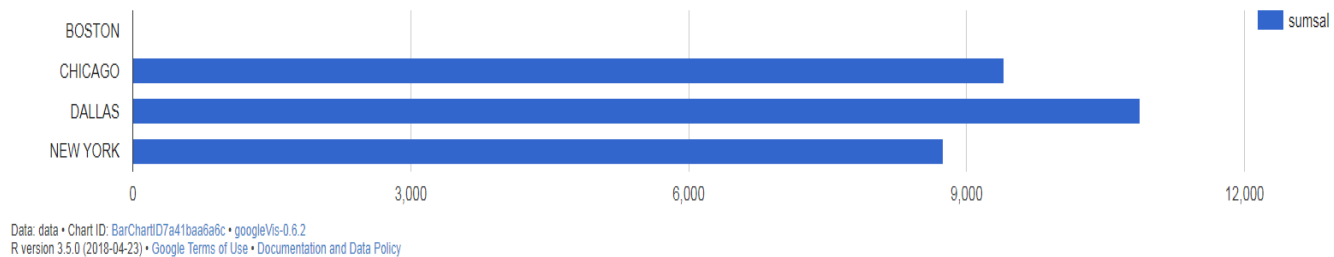
```
x3<-tapply(x$sal,x$loc,sum)
x3
```

```
> x3<-tapply(x$sal,x$loc,sum)
> x3
      BOSTON  CHICAGO  DALLAS NEW YORK
      NA      9400    10875    8750
```



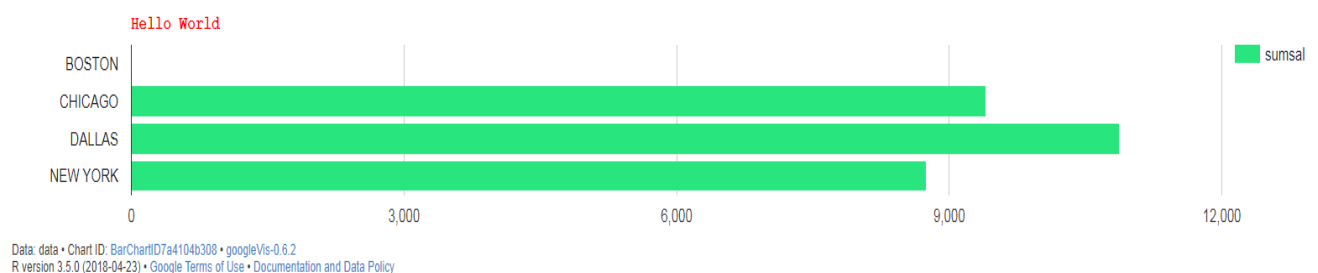
문제 159. 부서위치, 부서위치별 토탈 월급을 구글 막대 그래프로 시각화 하시오.

```
library(googleVis)
x4<-gvisBarChart(x2)
x4
plot(x4)
```



문제 160. 문제 159번에서 만들었던 구글 막대 그래프의 색깔을 변경 하시오.

```
x4<-gvisBarChart(x2, options = list(title="Hello World",
                                     titleTextStyle="{color:'red',fontName:'Courier',fontSize:16}",
                                     bar="{groupWidth:'80%'}",colors="['#29e57d']"))
plot(x4)
```



27. 지도 그래프 & 소리 시각화

2018년 5월 18일 금요일 오전 10:14

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 지도 그래프 & 소리 시각화
6. 워드 클라우드
7. 사분위수 그래프

1. 예제

문제 161.	Maps 패키지를 설치하고 중국지도만 확대해서 출력 하시오.
---------	-----------------------------------

```
install.packages("maps")  
install.packages("mapproj")  
library(maps)  
library(mapproj)  
map("world")  
map("world", "china")
```



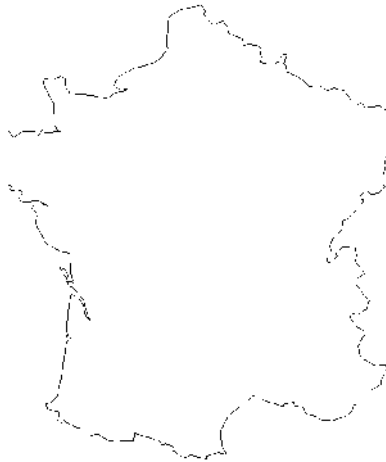
문제 162. 우리나라 지도를 출력 하시오.

```
map("world", "south korea")
```



문제 163. 프랑스 지도를 출력 하시오.

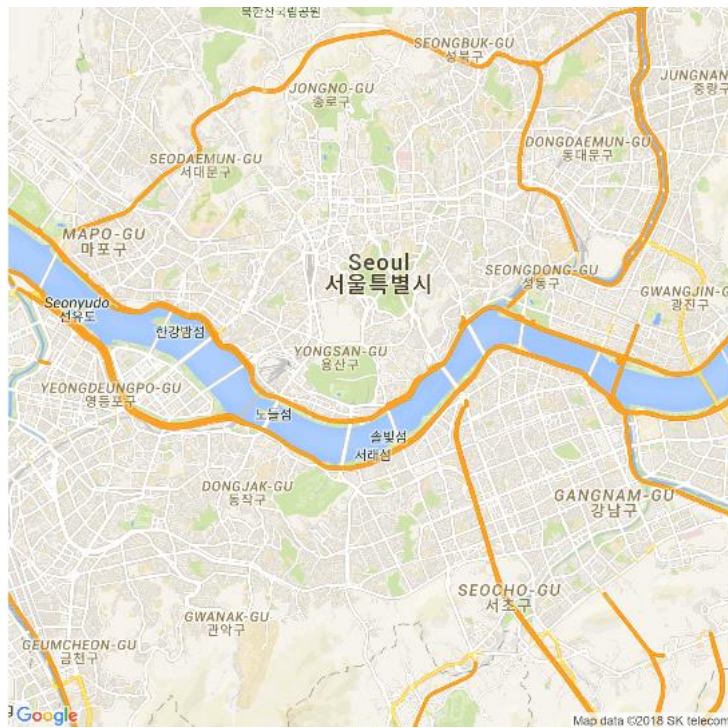
```
map("world", "france")
```



문제 164. 구글 지도 그래프를 이용해서 서울 지역의 지하철 2호선의 그래프를 시각화 하시오.

```
loc<-read.csv(file.choose(), header=T)
center<-c(mean(loc$LON),mean(loc$LAT))
center

kor<-get_map(center, zoom=12, maptype="roadmap")
plot(kor)
```



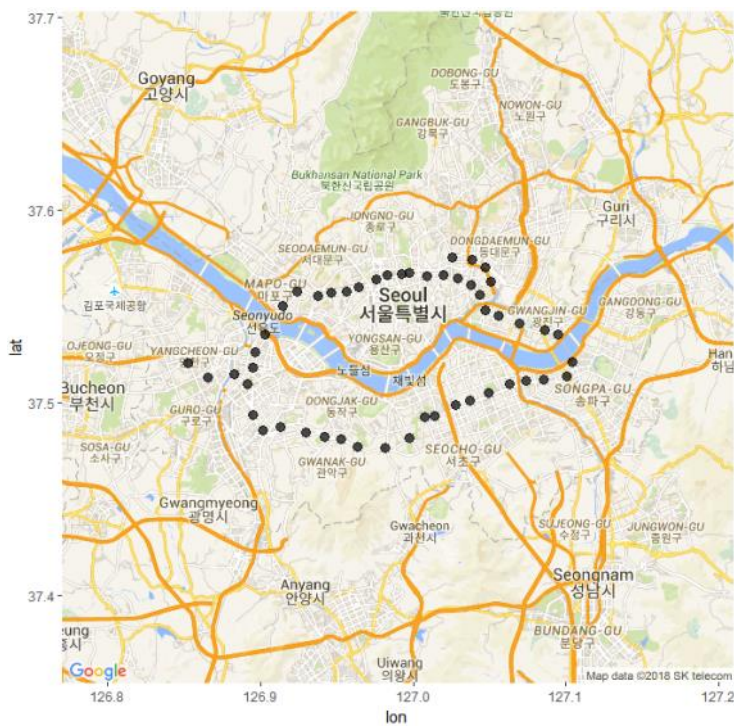
```
loc<-read.csv(file.choose(), header=T)
```

```
center <- c(mean(loc$LON),mean(loc$LAT))
```

```
kor <- get_map(center,zoom=11, maptype="roadmap")
```

```
kor.map <- ggmap(kor) + geom_point(data=loc,aes(x=LON,y=LAT),size=3, alpha=0.7)
```

```
kor.map + geom_text(data=loc, aes(x=LON,y=LAT+0.005,label=역명),size=3)
```

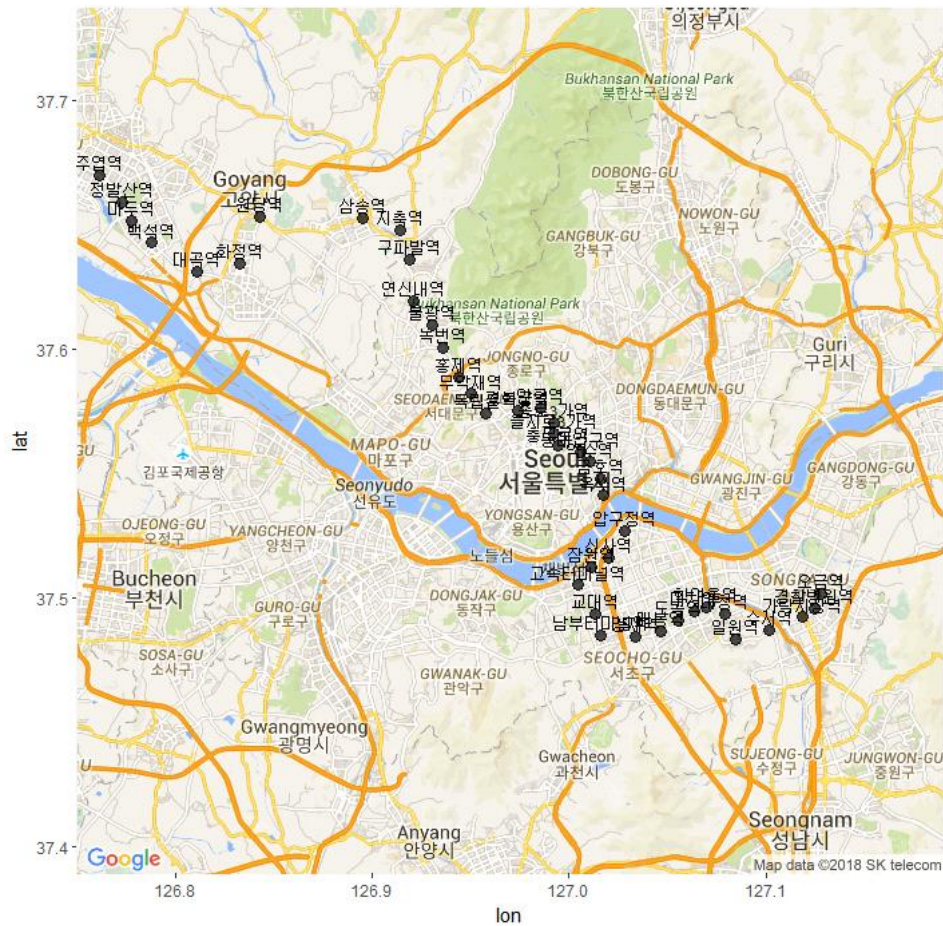


문제 165. 구글 지도 그래프를 이용해서 서울 지역의 지하철 3호선의 그래프를 시각화 하시오.

```
library(ggplot2)
```

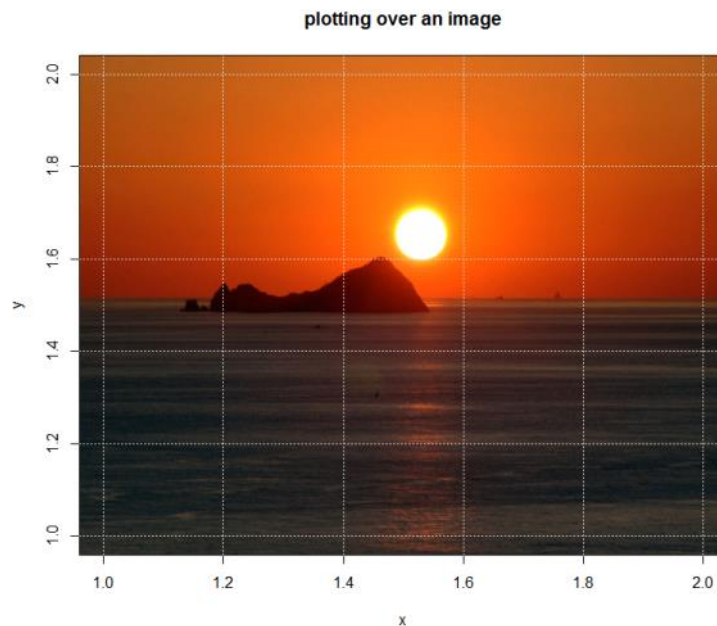
```
library(ggmap)
```

```
center <- c(mean(loc2$LON),mean(loc2$LAT))
kor <- get_map(center,zoom=11, maptype="roadmap")
kor.map <- ggmap(kor) + geom_point(data=loc2,aes(x=LON,y=LAT),size=3, alpha=0.7)
kor.map + geom_text(data=loc2, aes(x=LON,y=LAT+0.005,label=역명),size=3)
```



문제 166. Plot 그래프의 배경을 바꿔보자.

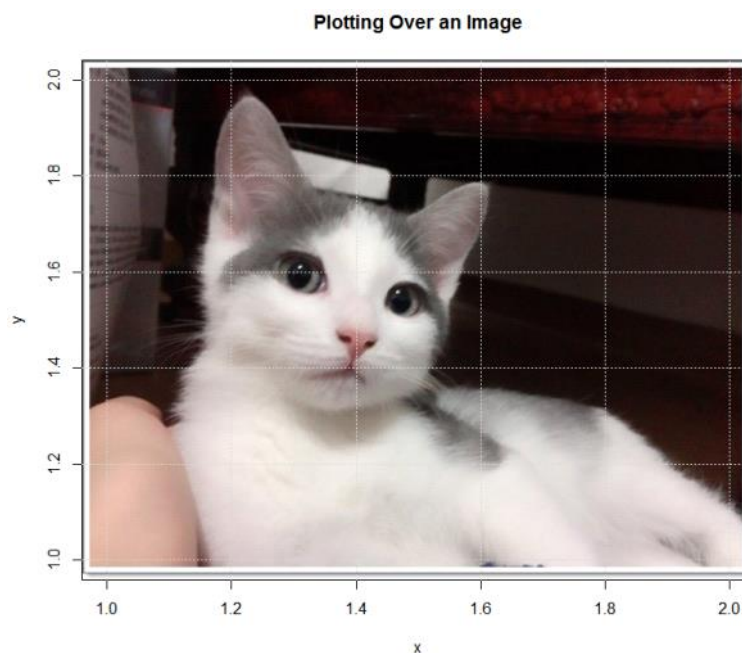
```
rasterImage(ima, lim$usr[1], lim$usr[3], lim$usr[2], lim$usr[4])
lim <- par() #그래프 변경 옵션
grid() # 칸을 그려줌
lines(c(1, 1.2, 1.4, 1.6, 1.8, 2.0), c(1, 1.3, 1.7, 1.6, 1.7, 1.0), type="b", lwd=5, col="white")
```

문제 167. Plot 그래프를 다시 그리는데 배경 사진을 본인이 원하는 사진으로 변경해서 그리시오.

```
ima2<-readJPEG("c:\\data\\cat.jpg")
plot(1:2, type='n', main="Plotting Over an Image", xlab="x", ylab="y")

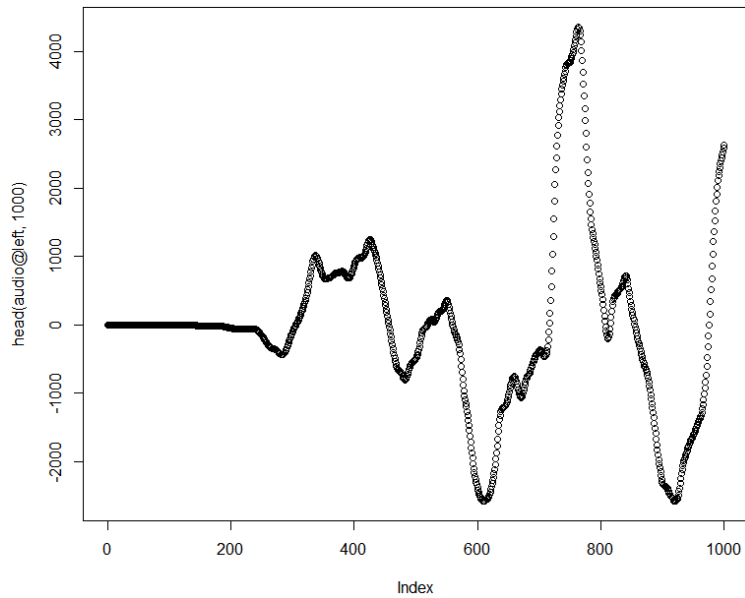
rasterImage(ima2, lim$usr[1], lim$usr[3], lim$usr[2], lim$usr[4])
lim <- par() #그래프 변경 옵션
grid() # 칸을 그려줌
lines(c(1, 1.2, 1.4, 1.6, 1.8, 2.0), c(1, 1.3, 1.7, 1.6, 1.7, 1.0), type="b", lwd=5, col="white")
```



문제 168. Output.wav의 소리를 R로 시각화 하시오.

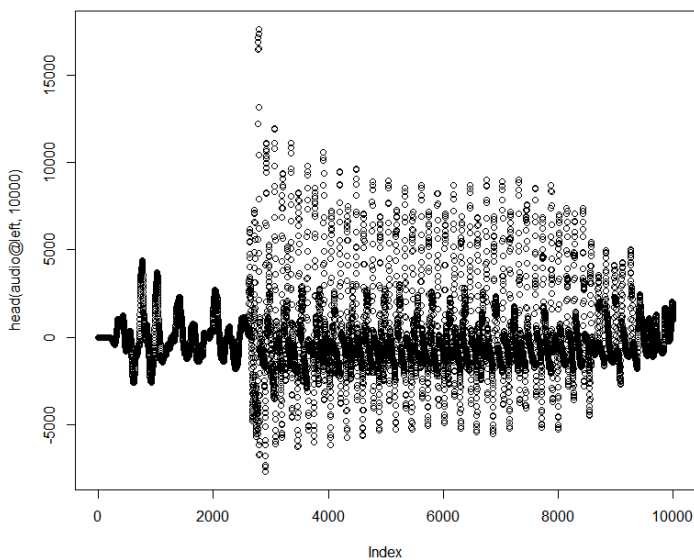
```
install.packages("tuneR")
library(tuneR)
audio<-readWave("c:\\data\\sound\\output.wav")
```

```
play(audio)
head(audio@left,1000)
plot(head(audio@left,1000))
```



문제 169. 원더걸스의 so hot을 시각화 하시오.

```
audio2<-readWave("c:\\data\\sound\\sohot.wav")
play(audio2)
head(audio2@left,10000)
plot(head(audio@left,10000))
```



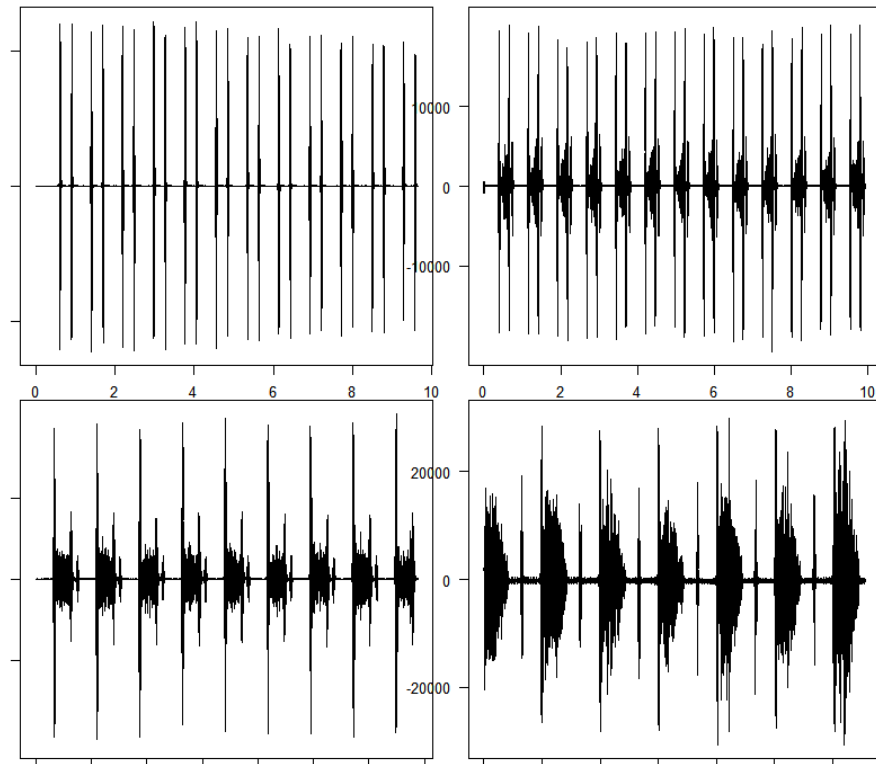
문제 170. 정상적인 심장박동 소리와 질병이 있는 심장박동 소리를 서로 비교할 수 있도록 시각화 하시오.

```
graphics.off()
par(mfrow=c(2,2)) #4개의 그래프를 한화면에 표시 할것 (2행2열)
par(mar=c(1,1,1,1)) #위,아래,좌,우 여백 사이즈
audio1<-readWave("c:\\data\\sound\\normal.wav")
```

```

audio2<-readWave("c:\\data\\sound\\ps.wav")
audio3<-readWave("c:\\data\\sound\\mr.wav")
audio4<-readWave("c:\\data\\sound\\ar.wav")
plot(audio1)
plot(audio2)
plot(audio3)
plot(audio4)

```



문제 171. 테러가 일어난 지역에 대한 위도,경도 정보를 가지고 세계지도를 바탕으로 두고 테러가 일어난 지역에 plot 그래프로 점표시를 하시오.

```

terror<- read.csv('terror_2015.csv')
terror<- na.omit(terror)

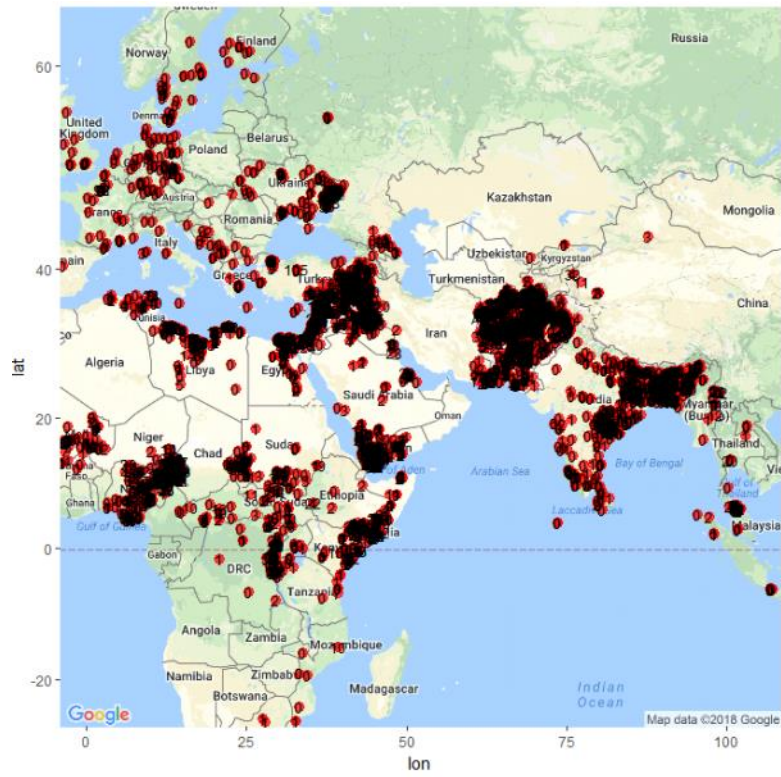
center<- c(mean(terror$longitude), mean(terror$latitude))
center

X11()
ter<- get_map(center,
  zoom = 3,
  maptype = 'roadmap')

ter.map<- ggmap(ter) + geom_point(data = terror
  ,aes(x = longitude,
    y = latitude),
  size = 3,
  alpha = 0.7,
  col = 'red')
ter.map + geom_text(data = terror,
  aes(x = longitude,
    y = latitude + 0.005,
    label = nkill ),

```

size = 3)



28. 워드 클라우드

2018년 5월 18일 금요일 오후 4:29

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 지도 그래프 & 소리 시각화

6. 워드 클라우드

7. 사분위수 그래프

1. 예제

문제 172. 안철수 연설문을 가지고 워드 클라우드 형식으로 시각화 하시오.

```
install.packages("KoNLP")          # 한국어를 R에서 인식할수 있도록 하는 패키지
install.packages("wordcloud")      # 워드클라우드 그리는 패키지
install.packages("plyr")           # 워드클라우드를 그릴때 필요하다

library(wordcloud)
library(KoNLP)
library(plyr)

ahn <- " 연설문 내용 "

useSejongDic()                    # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

mergeUserDic(data.frame(c('안철수', '박근혜', '문재인'), c('nqpc')))) # 세종 사전에 3개의 단어를 추가
                                                                    (안에 존재하지 않는 단어이므로)

nouns <- extractNoun(ahn)         # 연설문에서 명사만 출력
nouns <- nouns[nchar(nouns)>=2]    # 두글자 이상인 명사만 추출
cnouns <- count(nouns)            # 단어와 건수 출력

pal <- brewer.pal(6,"Dark2")      # Dark2라는 색깔을 추가하는 작업
pal <- pal[-(1)]

windowsFonts(malgun=windowsFont("맑은 고딕"))    #맑은 고딕 폰트 추가

wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
           random.order=F, family="malgun")        #워드 클라우드 그리는 문법
```




문제 173. 안철수 연설문의 단어와 건수를 출력하는데 건수가 높은 것부터 출력 하시오.

```
head(cnouns[order(cnouns$freq, decreasing = T)],10)
```

	x	freq
219	저	35
83	들	24
47	국민	20
153	수	17
277	한	15
17	것	14
232	정치	12
135	생각	11
101	미래	9
15	거	6

문제 174. 영화 겨울 왕국 대본을 워드 클라우드 시각화 하시오.

```
graphics.off()
```

```
winter<- readLines('c:\wwdata\ww\winter.txt')
nouns <- extractNoun(winter) # 연설문에서 명사만 출력
nouns<-unlist(nouns) # list형을 char형으로 바꿈
nouns <- nouns[nchar(nouns)>=4] #두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns)) #단어와 건수 출력
```

```
pal <- brewer.pal(6,"Dark2") # Dark2라는 색깔을 추가하는 작업
pal <- pal[-(1)]
```

```
windowsFonts(malgun=windowsFont("맑은 고딕")) #맑은 고딕 폰트 추가
```

```
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
          random.order=F, family="malgun") #워드 클라우드 그리는 문법
# min.freq = 3: 단어의 빈도수가 3개 이상인것만 시각화
# random.order = F : 가장 많은것부터 중앙에서 퍼지게한다.
```

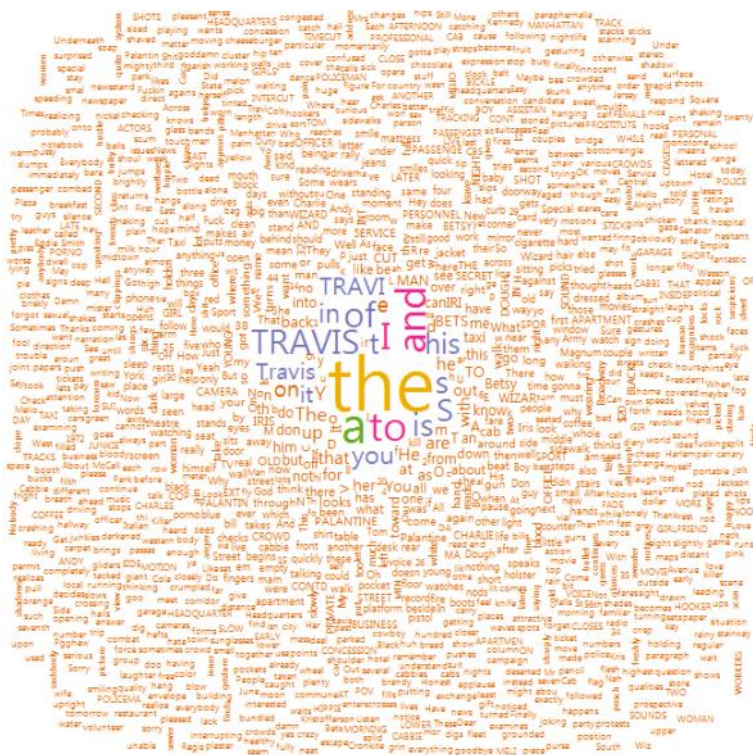

문제 177. 영화 대본을 워드 클라우드로 시각화 하시오.

```
graphics.off()
taxi <- readLines(file.choose())

nouns <- extractNoun(taxi) # 연설문에서 명사만 출력
nouns <- nouns[nchar(nouns)>=2] #두글자 이상인 명사만 추출
cnouns <- count(unlist(nouns)) #단어와 건수 출력

pal <- brewer.pal(6,"Dark2") # Dark2라는 색깔을 추가하는 작업
pal <- pal[(-1)]

windowsFonts(malgun=windowsFont("맑은 고딕")) #맑은 고딕 폰트 추가
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
  random.order=F, family="malgun")
```



문제 178. 텍스트 파일을 물어보게 하고 텍스트 파일명을 입력하면 자동으로 워드 클라우드가 그려지는 함수를 생성 하시오.

```
word_cloud <- function() {

  #install.packages("KoNLP") # 한국어를 R에서 인식할수 있도록 하는 패키지
  #install.packages("wordcloud") # 워드클라우드 그리는 패키지
  #install.packages("plyr") # 워드클라우드를 그릴때 필요하다

  library(wordcloud)
  library(KoNLP)
```



```

q<-tapply(res0[,res1], res0[,res2],sum)
}

switch(x1,
  {#막대그래프
    q[is.na(q)] <- 0
    barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1],'총합' ),
beside = T, ylim = c(0,max(q)*1.4))
    legend("topright", rownames(q),title = paste(colnames(res0)[res2],' 구분' ),inset = 0,fill =
rainbow(nrow(q)),cex=0.8)
  },
  {#원형 그래프
    label<-paste(unique(res0[,res2]), round(q/sum(q) * 100,1),'%')
    pie(q,col=rainbow(nrow(q)),label=label,main = paste( colnames(res0)[res2], '별',colnames(res0)[res1],'총합' ))
  },
  {#산포도 그래프
    res0<- get(readline(prompt = '테이블명 입력 : '))
    x <- menu(colnames(res0), title='x축 컬럼명 선택 : ')
    y <- menu(colnames(res0), title='y축 컬럼명 선택 : ')
    plot(res0[,x],res0[,y],pch=16, col=blues9,xlab = colnames(res0)[x] ,ylab = colnames(res0)[y],main =
paste(colnames(res0)[x],'와 ',colnames(res0)[y],'의 상관 관계 '))

  },
  {#워드클라우드

    library(wordcloud)
    library(KoNLP)
    library(plyr)
    useSejongDic()          # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

    graphics.off()

    res<-readline(prompt = 'c:wwdata 경로에 위치한 txt 파일명 입력 : ')
    word<-readLines(gsub(' ','',paste('c:wwwwdatawwww',res,'.txt'))))

    nouns <- extractNoun(word)  # 연설문에서 명사만 출력
    nouns <- nouns[nchar(nouns)>=2]  #두글자 이상인 명사만 추출
    cnouns <- count(unlist(nouns))  #단어와 건수 출력

    pal <- brewer.pal(6,"Dark2")  # Dark2라는 색깔을 추가하는 작업
    pal <- pal[1:3]
    windowsFonts(malgun=windowsFont("맑은 고딕"))  #맑은 고딕 폰트 추가
    wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
              random.order=F, family="malgun")
  }
)
}

```

그룹함수까지 지정할 수 있는 소스


```

graph_func<-function(){
  graphics.off()
  q3<-menu(c('막대그래프','원형그래프','산포도그래프','워드클라우드'),title='시각화 종류 : ')
  if(q3 <= 3){ q0<-get(readline(prompt='테이블 이름?'))
  q2<-menu(colnames(emp),title='x축: ')
  q1<-menu(colnames(emp),title='y축: ')
  m1<-colnames(q0[q1])
  m2<-colnames(q0[q2])
  }

  # 바/원형그래프

  bbb<-function(){
    x<-data.table(q1,q2)
    xxx<-switch(menu(c('합계','평균','최소','최대','건수'),title='어떻게 그룹화하시겠습니까?'),
      'sum','mean','min','max','length')
    x<-tapply(q0[q1],q0[q2],xxx)
    label<-paste(sort(unique(colnames(t(x)))),'(',round(x/sum(x)*100,1),'%')
    mains<-paste(m2,'별 ',m1, xxx)
    switch(q3,bar=barplot(x,names=colnames(t(x)),col=blues9, ylab=m1,cex.names=0.7, main=mains),
      piee=pie(x,col=blues9,labels=label,main=mains))

  }

  #산포도 그래프
  plotfunc<-function(){
    plot(q0[q2],q0[q1],col=blues9,pch=16,xlab=m2, ylab=m1,
      main=paste(m2,'&',m1,'의 상관관계'))
  }

  #워드클라우드
  wordcloud7<-function(){
    rm(wordcloud)
    library(KoNLP)
    library(wordcloud)
    library(plyr)
    pal <- brewer.pal(6,"Dark2")
    useSejongDic()
    xt<-readline(prompt='워드클라우드를 그릴 원본 텍스트명을 입력하세요!')
    setwd('c:\\data')

    xn<-readLines(paste(xt,'.txt',sep=''))
    x_n<-extractNoun(xn)
    x_n<-unlist(x_n)

    x_n<-x_n[nchar(x_n)>=4]
    x_n<-count(x_n)
    wordcloud(words=x_n$x, x_n$freq, colors=pal, min.freq=3,

```

```
        random.order=F, family="malgun")  
    }  
  
    switch(q3,bbb(),bbb(),plotfunc(),wordcloud7())
```


29. 사분위수 그래프

2018년 5월 21일 월요일 오전 10:06

■ 그래프의 종류

1. 막대 그래프
2. 원형 그래프
3. 산포도(Plot) 그래프
4. 구글에서 제공하는 그래프
5. 지도 그래프 & 소리 시각화
6. 워드 클라우드
7. 사분위수 그래프

■ 평균 값이란?

왜 통계학에서는 average라고 안하고 mean 이라고 할까?

평균 값을 구하는데는 여러가지 방법이 있기 때문이다.

Ex) 우리반 학생들의 나이 평균 값을 구한다고 하면, 지금 사람의 수가 몇 명인지 알고 있지만 새로운 누군가가 들어오면 다시 계산해야하는 번거로움이 생긴다. 이러한 번거로움을 피하기 위한 방법이 무엇인가 ?

----> 통계학자들은 숫자를 문자로 표현함으로써 이러한 번거로움을 해소했다.

합계 : $x_1 + x_2 + x_3 + \dots + x_n$ 이를 표현한 간단한 방법 ? $\sum x$ (시그마 x)

----> 그럼 평균을 구하는 것을 문자로 나타낸다면?

$$\sum x / n = \mu \text{ (뮤)}$$

1. 예제

문제 180. 아래 나이의 평균을 R로 구하시오.

나이	19	20	21
도수	1	3	1

```
age<-mean(c(19,20,20,20,21))
```

```
age<-mean(c(19,rep(20,3),21))
```

rep 함수를 이용해서 더 간단히

```
> age<-mean(c(19,20,20,20,21))
> age
[1] 20
> age<-mean(c(19,rep(20,3),21))
> age
[1] 20
```

문제 181. 아래 쿡푸 교실의 수강생 나이 평균을 R로 구하시오.

나이	19	20	21	145	147
도수	3	6	3	1	1 (이상치)

```
age <- mean(c(rep(19,3),rep(20,6),rep(21,3),145,147))
```

```
> age <- mean(c(rep(19,3),rep(20,6),rep(21,3),145,147))
> age
[1] 38
```

문제 182. 쿡푸교실의 데이터중 이상치를 구하시오.

나이	19	20	21	145	147
도수	3	6	3	1	1 (이상치)

```
install.packages("outliers")
library(outliers)
age <- (c(rep(19,3),rep(20,6),rep(21,3),145,147))
outlier(age)
```

```
- -
> age <- (c(rep(19,3),rep(20,6),rep(21,3),145,147))
> outlier(age)
[1] 147
```

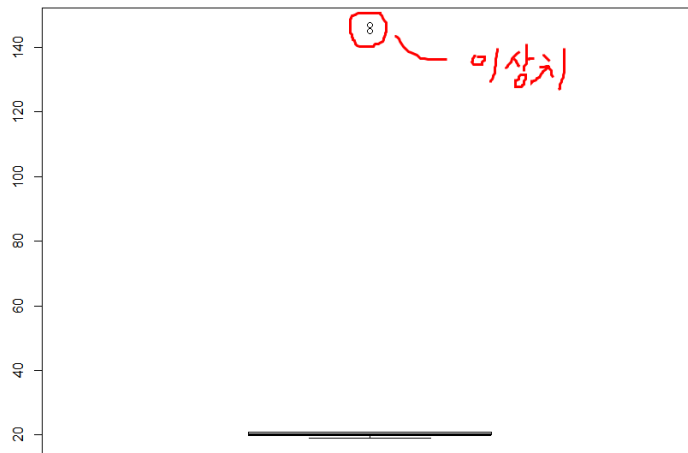
중앙값 (median)

어떤 여학생(25살)이 나랑 비슷한 나이대인 스포츠 센터 교실에 등록하려고 했는데 평균을 보니 38이어서 등록을 안 하게 되었다. 하지만, 대부분의 사람들은 20대인데 이상치 때문에 평균 값이 올라갔다. 이를 해결하기 위한 방법이 무엇인가?

-----> 평균 말고 다른 데이터를 알아야하는데 그것은 무엇일까? = 중앙값 (median)

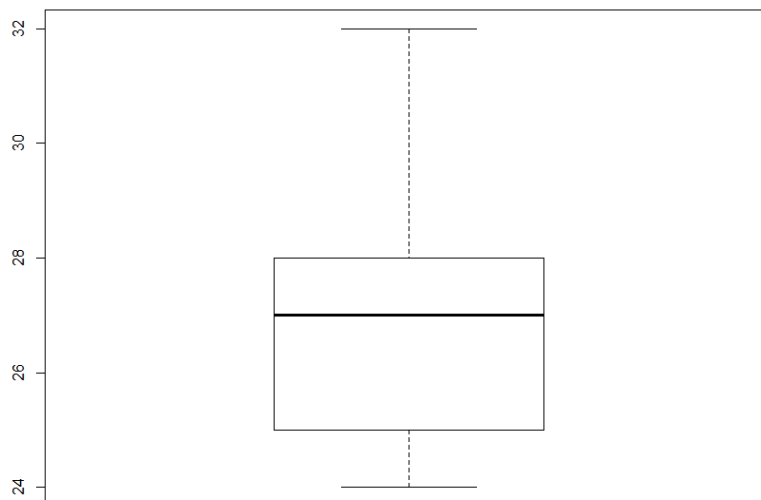
문제 183. age (19 19 19 20 20 20 20 20 20 21 21 21 145 147) 데이터를 boxplot 그래프로 시각화 하시오.

```
a<-boxplot(age)
```



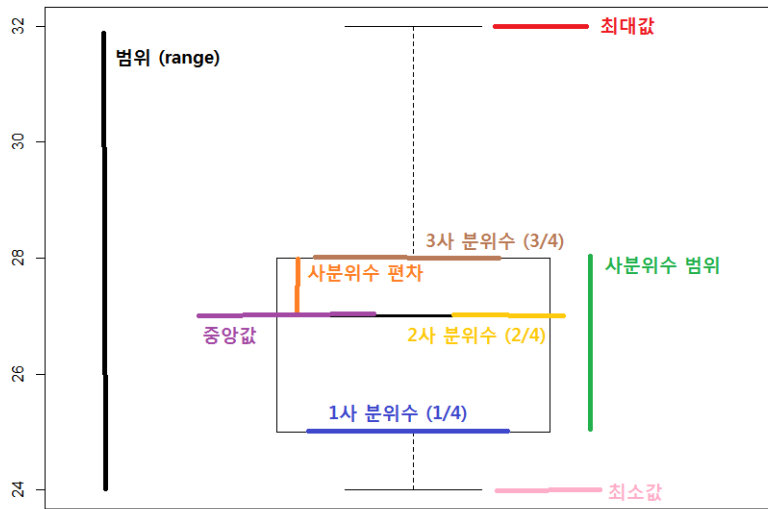
문제 184. 우리반 나이 데이터를 가지고 박스 그래프(사분위수 그래프)를 그리시오.

```
emp2<-read.csv("c:\wwdata\wwemp2.csv", header = T)
emp2_age <- emp2$age
boxplot(emp2_age)
```



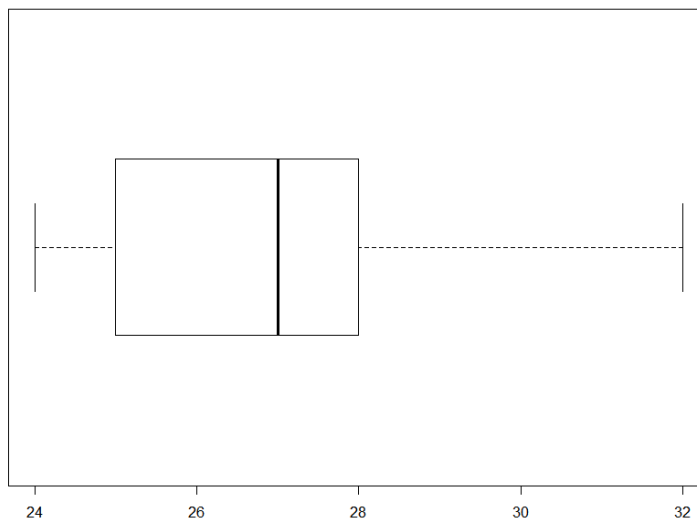
```
summary(emp2$age)
```

```
> summary(emp2$age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 24.00  25.25   27.00   26.70   28.00   32.00
```



문제 185. 문제 184번의 그래프를 옆으로 그려지게 하시오.

```
boxplot(emp2_age, horizontal = T)
```



문제 186. 사분위수 그래프를 자동화 그래프 함수에 5번째로 추가 하시오.

```
baek_func <- function() {

  graphics.off()
  x1 <- menu( c("막대그래프","원형그래프","산포도그래프","워드클라우드","사분위수그래프") ,title ='원하는 그래프의 숫
자를 선택하세요 ')

  res0<- get(readline(prompt = '테이블명 입력 : '))
  if (x1 == 1 | x1 == 2){
    res1<- menu(colnames(res0), title='토탈 값을 구할 컬럼 선택 : ')
    res2<- menu(colnames(res0), title='그룹핑할 컬럼 선택 : ')
    q<-tapply(res0[,res1], res0[,res2],sum)
  }
}
```

```

switch(x1,
  {#막대 그래프
    q[is.na(q)] <- 0
    barplot(q, col = rainbow(nrow(q)), main = paste( colnames(res0)[res2], '별', colnames(res0)[res1], '총합' ),
    beside = T, ylim = c(0,max(q)*1.4))
    legend("topright", rownames(q), title = paste(colnames(res0)[res2], ' 구분' ), inset = 0, fill =
rainbow(nrow(q)), cex=0.8)
  },
  {#원형 그래프
    label <- paste(unique(res0[,res2]), round(q/sum(q) * 100, 1), '%')
    pie(q, col = rainbow(nrow(q)), label = label, main = paste( colnames(res0)[res2], '별', colnames(res0)[res1], '총합' ))
  },
  {#산포도 그래프
    x <- menu(colnames(res0), title = 'x축 컬럼 선택 : ')
    y <- menu(colnames(res0), title = 'y축 컬럼 선택 : ')
    plot(res0[,x], res0[,y], pch=16, col=blues9, xlab = colnames(res0)[x], ylab = colnames(res0)[y], main =
paste(colnames(res0)[x], '와 ', colnames(res0)[y], '의 상관 관계 '))

  },
  {#워드클라우드

    library(wordcloud)
    library(KoNLP)
    library(plyr)
    useSejongDic() # 370957개의 한글 단어가 추가 (전희원 선생님이 만듦)

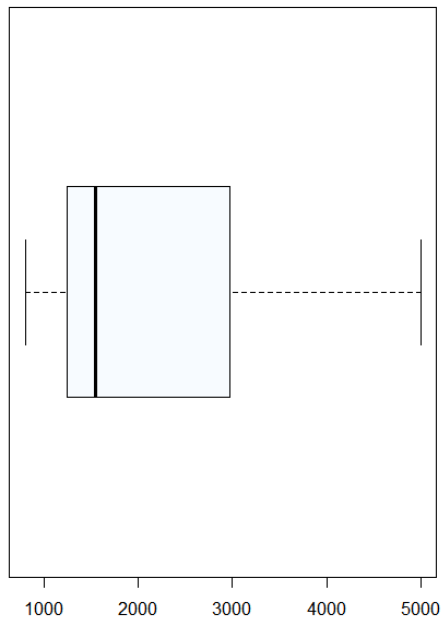
    graphics.off()

    res <- readline(prompt = 'c:\WWWdata 경로에 위치한 txt 파일명 입력 : ')
    word <- readLines(gsub(' ', '', paste('c:\WWWdata\WWW', res, '.txt')))

    nouns <- extractNoun(word) # 연설문에서 명사만 출력
    nouns <- nouns[nchar(nouns) >= 2] # 두글자 이상인 명사만 추출
    cnouns <- count(unlist(nouns)) # 단어와 건수 출력

    pal <- brewer.pal(6, "Dark2") # Dark2라는 색깔을 추가하는 작업
    pal <- pal[-(1)]
    windowsFonts(malgun = windowsFont("맑은 고딕")) # 맑은 고딕 폰트 추가
    wordcloud(words = cnouns$x, freq = cnouns$freq, colors = pal, min.freq = 3,
              random.order = F, family = "malgun")
  },
  {#사분위수 그래프
    res1 <- menu(colnames(res0), title = '컬럼 선택 : ')
    boxplot(res0[,res1], horizontal = T, col = blues9)
  }
)
}

```



문제 187. 웹 브라우저에서 R shiny으로 그래프를 띄우는 것을 구현 하시오.

기술통계 함수

2018년 5월 18일 금요일 오후 8:15

```
x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
y <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

분포 및 중심화 경향

함수명	설명	예제
mean(x)	평균	mean(x) # 5
median(x)	중앙 값 벡터 x가 홀수개이면 정 가운데 값을 중앙값을 가져오지만, 위의 case와 같이 x가 짝수개 이면 정가운데의 양쪽 두개의 값을 가져 다가 평균을 내서 중앙값을 계산합니다.	median(x) # 5.5
min(x)	최소 값	min(x) # 1
max(x)	최대 값	max(x) # 10
range(x)	범위 값 (최소 최대 값 출력)	range(x) # 1 10
IQR(x)	IQR(Inter-Quartile Range)	IQR(x) # 4.5
summary(x)	중심화 경향 및 분포 요약 <pre>> summary(x) Min. 1st Qu. Median Mean 3rd Qu. Max. 1.00 3.25 5.50 5.50 7.75 10.00</pre>	

#퍼짐 정도

함수명	설명	예제
var(x)	분산	var(x) # 9.166667
sd(x)	표준 편차	sd(x) # 3.02765

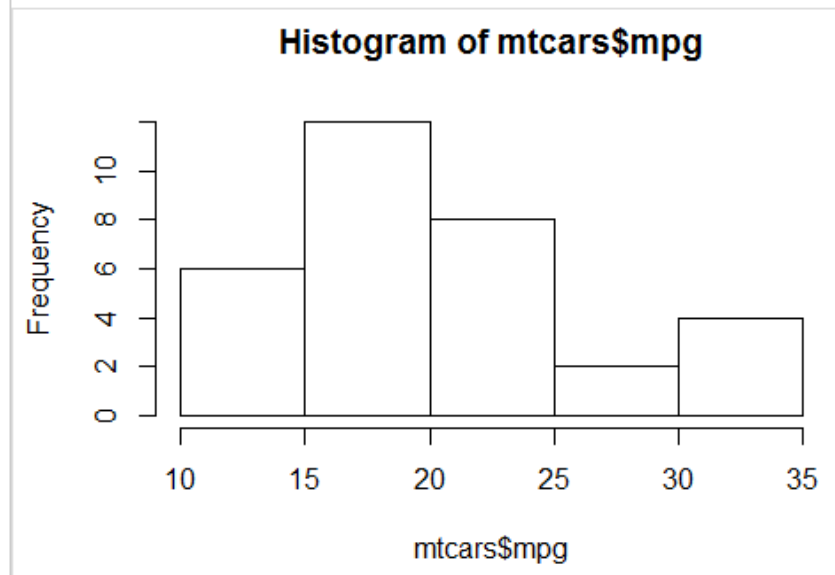
#확률분포의 비대칭 정도

왜도 : skewness(x)

R에 왜도와 첨도를 위한 함수가 내장되어 있지 않기 때문에 별도 패키지(fBasics)를 설치해야 합니다.

자동차 정보가 들어있는 mtcars 데이터 프레임의 연비에 대해서 히스토그램을 그려보니 평균보다 왼쪽으로 치우쳐 있고 오른쪽으로 꼬리가 긴 분포를 띠고 있네요. 그러면 왜도(skewness) 가 '0'보다 크게 나타납니다. (공식이 평균에서 관측치를 뺀 값을 3제곱 하기 때문이에요) 위 예에서는 왜도가 0.61로 '0'보다 크게 나왔지요. 정규분포의 평균과 일치하면 왜도는 '0'이 되고, 반대로 평균보다 오른쪽으로 값이 치우쳐 있고 왼쪽으로 꼬리가 길면 왜도는 '0'보다 작은 값이 나옵니다.

```
> install.packages("fBasics") # 왜도, 첨도 분석 가능한 package 설치
> library(fBasics) # package 호출
> hist(mtcars$mpg)
```



```
> skewness(mtcars$mpg)
[1] 0.610655
attr(,"method")
[1] "moment"
```

첨도 : kurtosis(x)

관측값이 정규분포보다 뾰족한가 아닌가를 가늠하는 척도가 첨도입니다. '3'보다 크면 정규분포보다 더 뾰족한 모양이고, '3'보다 작으면 정규분포보다 덜 뾰족한 모양이라고 해석하면 되겠습니다. (패키지에 따라서는 '3'을 빼서 '0'으로 표준화해서 값을 제시하기도 합니다)

```
> kurtosis(mtcars$mpg)
[1] -0.372766
attr(,"method")
[1] "excess"
```


데이터 편집 edit()

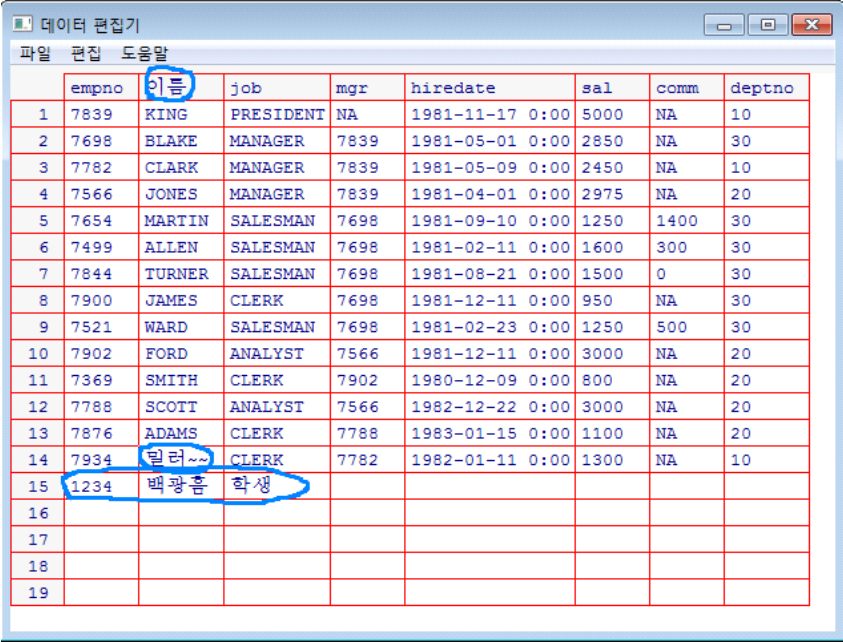
2018년 5월 19일 토요일 오후 4:22

엑셀 처럼 생긴 데이터 편집기 창을 사용하려면 `edit()` 함수를 이용하며, 데이터 프레임 구조로 저장된다. 비교적 소규모의 데이터를 입력하기에는 써볼만 하겠지만, 대용량 데이터를 입력은 무리가 있다.

예제. emp테이블의 구조와 데이터를 수정해보자.

```
> emp2<-edit(emp)
```

```
> emp2
```



	empno	이름	job	mgr	hiredate	sal	comm	deptno
1	7839	KING	PRESIDENT	NA	1981-11-17 0:00	5000	NA	10
2	7698	BLAKE	MANAGER	7839	1981-05-01 0:00	2850	NA	30
3	7782	CLARK	MANAGER	7839	1981-05-09 0:00	2450	NA	10
4	7566	JONES	MANAGER	7839	1981-04-01 0:00	2975	NA	20
5	7654	MARTIN	SALESMAN	7698	1981-09-10 0:00	1250	1400	30
6	7499	ALLEN	SALESMAN	7698	1981-02-11 0:00	1600	300	30
7	7844	TURNER	SALESMAN	7698	1981-08-21 0:00	1500	0	30
8	7900	JAMES	CLERK	7698	1981-12-11 0:00	950	NA	30
9	7521	WARD	SALESMAN	7698	1981-02-23 0:00	1250	500	30
10	7902	FORD	ANALYST	7566	1981-12-11 0:00	3000	NA	20
11	7369	SMITH	CLERK	7902	1980-12-09 0:00	800	NA	20
12	7788	SCOTT	ANALYST	7566	1982-12-22 0:00	3000	NA	20
13	7876	ADAMS	CLERK	7788	1983-01-15 0:00	1100	NA	20
14	7934	밀러~~	CLERK	7782	1982-01-11 0:00	1300	NA	10
15	1234	백광흠	학생					
16								
17								
18								
19								

```
> emp2
  empno  이름      job mgr      hiredate  sal comm deptno
1  7839   KING  PRESIDENT   NA 1981-11-17 0:00 5000   NA     10
2  7698   BLAKE   MANAGER 7839 1981-05-01 0:00 2850   NA     30
3  7782   CLARK   MANAGER 7839 1981-05-09 0:00 2450   NA     10
4  7566   JONES   MANAGER 7839 1981-04-01 0:00 2975   NA     20
5  7654  MARTIN  SALESMAN 7698 1981-09-10 0:00 1250 1400     30
6  7499   ALLEN  SALESMAN 7698 1981-02-11 0:00 1600  300     30
7  7844  TURNER  SALESMAN 7698 1981-08-21 0:00 1500    0     30
8  7900   JAMES    CLERK 7698 1981-12-11 0:00  950   NA     30
9  7521   WARD   SALESMAN 7698 1981-02-23 0:00 1250  500     30
10 7902   FORD   ANALYST 7566 1981-12-11 0:00 3000   NA     20
11 7369   SMITH    CLERK 7902 1980-12-09 0:00  800   NA     20
12 7788   SCOTT   ANALYST 7566 1982-12-22 0:00 3000   NA     20
13 7876  ADAMS    CLERK 7788 1983-01-15 0:00 1100   NA     20
14 7934  밀러~~    CLERK 7782 1982-01-11 0:00 1300   NA     10
15 1234  백광흠     학생   NA      <NA>   NA   NA     NA
```

결측값(NA) 확인 및 처리

2018년 5월 19일 토요일 오후 5:44

#외부 데이터를 불러오고 하는 과정

1. Str() 함수를 이용해서 데이터 구조 파악하기
2. Head() or tail() 함수를 이용해서 데이터 몇 개 미리보기
3. 결측 값 확인 및 처리 (NA 값 처리)
4. 탐색적 데이터 분석 (특이값/ 영향치 확인 및 처리 등)

R에서 결측값이 들어있는 상태에서 통계 분석을 진행하면 NA 라는 결과가 나올 뿐, 원하는 결과를 얻지 못한다. 그리고 대부분의 R 통계 함수에는 옵션으로 "na.rm = TRUE" 라는 옵션을 제공해서 결측값을 통계량 계산할 때 포함하지 않기를 선택할 수 있게 해준다.

결측값이 포함되어있는지 확인 : is.na(x)

```
x <- c(1, 2, 3, 4, NA, 6, 7, 8, 9, NA)
is.na(x)

> x <- c(1, 2, 3, 4, NA, 6, 7, 8, 9, NA)
> is.na(x)
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE
```

위의 벡터처럼 구성요소 갯수가 몇 개 안될 경우 is.na() 한 후에 TRUE, FALSE 논리형 값을 눈으로 보고 확인할 수 있다. 하지만 데이터 프레임처럼 변수 갯수도 많고, 관측치 갯수도 많은 경우 (대부분의 실무에서 쓰는 데이터 셋) is.na() 함수만 가지고서는 아무래도 결측치 현황을 파악하는데 무리가 있다.

결측값이 총 몇 개 인지 계산 : sum(is.na()), colSums(is.na())

```
sum(is.na(x))

> x <- c(1, 2, 3, 4, NA, 6, 7, 8, 9, NA)
> is.na(x)
[1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE
> sum(is.na(x))
[1] 2
```

colSums(is.na(emp)) # colSums() 함수를 이용하면 각 열마다 na값이 몇 개 인지 확인 할 수 있다.

```
> colSums(is.na(emp))
      empno      ename      job      mgr      hiredate      sal      comm      deptno
      0         0         0         1         0         0         10         0
```

결측값을 통계 분석 시 제외(미포함) : na.rm = TRUE

```
sum(emp$comm) # na 값이 포함되어 있어서 na 값이 출력
sum(emp$comm, na.rm = T) # na값을 미포함해서 계산 한다.
```

```
> emp$comm
[1] NA NA NA NA 1400 300 0 NA 500 NA NA NA NA NA
> sum(emp$comm)
[1] NA
> sum(emp$comm, na.rm = T)
[1] 2200
```

결측값이 들어있는 행 전체를 데이터셋에서 제거 : na.omit()

```
> emp
  empno  ename      job mgr    hiredate   sal comm deptno
1  7839   KING PRESIDENT  NA 1981-11-17 0:00 5000   NA     10
2  7698  BLAKE   MANAGER 7839 1981-05-01 0:00 2850   NA     30
3  7782  CLARK   MANAGER 7839 1981-05-09 0:00 2450   NA     10
4  7566  JONES   MANAGER 7839 1981-04-01 0:00 2975   NA     20
5  7654 MARTIN  SALESMAN 7698 1981-09-10 0:00 1250 1400     30
6  7499  ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300     30
7  7844  TURNER SALESMAN 7698 1981-08-21 0:00 1500 0       30
8  7900  JAMES   CLERK   7698 1981-12-11 0:00 950    NA     30
9  7521  WARD    SALESMAN 7698 1981-02-23 0:00 1250 500     30
10 7902  FORD    ANALYST 7566 1981-12-11 0:00 3000   NA     20
11 7369  SMITH   CLERK   7902 1980-12-09 0:00 800    NA     20
12 7788  SCOTT   ANALYST 7566 1982-12-22 0:00 3000   NA     20
13 7876  ADAMS   CLERK   7788 1983-01-15 0:00 1100   NA     20
14 7934  MILLER  CLERK   7782 1982-01-11 0:00 1300   NA     10
```

```
emp2 <- is.omit(emp) #emp 데이터 셋에서 na값이 포함된 행을 제거하고 emp2 에 저장
```

```
> emp2<-na.omit(emp)
> emp2
  empno  ename      job mgr    hiredate   sal comm deptno
5  7654 MARTIN  SALESMAN 7698 1981-09-10 0:00 1250 1400     30
6  7499  ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300     30
7  7844  TURNER SALESMAN 7698 1981-08-21 0:00 1500 0       30
9  7521  WARD    SALESMAN 7698 1981-02-23 0:00 1250 500     30
```

데이터프레임의 모든 행의 결측값을 특정 값으로 일괄 대체 : dataset[is.na(dataset)] <- 특정 값

```
emp[is.na(emp)] <- 0
emp
```

```
> emp[is.na(emp)] <- 0
> emp
  empno  ename      job mgr    hiredate   sal comm deptno
1  7839   KING PRESIDENT  0 1981-11-17 0:00 5000 0       10
2  7698  BLAKE   MANAGER 7839 1981-05-01 0:00 2850 0       30
3  7782  CLARK   MANAGER 7839 1981-05-09 0:00 2450 0       10
4  7566  JONES   MANAGER 7839 1981-04-01 0:00 2975 0       20
5  7654 MARTIN  SALESMAN 7698 1981-09-10 0:00 1250 1400     30
6  7499  ALLEN  SALESMAN 7698 1981-02-11 0:00 1600 300     30
7  7844  TURNER SALESMAN 7698 1981-08-21 0:00 1500 0       30
8  7900  JAMES   CLERK   7698 1981-12-11 0:00 950    0       30
9  7521  WARD    SALESMAN 7698 1981-02-23 0:00 1250 500     30
10 7902  FORD    ANALYST 7566 1981-12-11 0:00 3000 0       20
11 7369  SMITH   CLERK   7902 1980-12-09 0:00 800    0       20
12 7788  SCOTT   ANALYST 7566 1982-12-22 0:00 3000 0       20
13 7876  ADAMS   CLERK   7788 1983-01-15 0:00 1100 0       20
14 7934  MILLER  CLERK   7782 1982-01-11 0:00 1300 0       10
```

연속형->범주형으로 변경

2018년 5월 21일 월요일 오후 5:36

데이터는 크게(1) 명목형 또는 순서형의 범주형 데이터 (categorical data)와 (2) 연속형 데이터 (continuous data) 로 구분할 수 있다. R에서는 범주형 데이터를 요인(factor)형 데이터 구조라고 부르고 있으며, 순서(order)가 있는 경우는 순서형 요인(ordered factor)라고 해서 구분하기도 합니다.

분석하고자 하는 데이터 셋을 받으면 제일 먼저 데이터 구조와 데이터 형태를 탐색하게 된다.
그리고 분석 목적과 시나리오에 따라서 변수를 변환한다.

이번 페이지에선 연속형 변수를 범주형 변수로 변환하는 3가지 방법에 대해서 알아보도록 한다.

통계기법 중 도수분포표, 교차분할표, 카이제곱 검정이라든지, 로지스틱회귀분석, 그래프 중 막대그림, 원그림, 점그림 등의 경우 범주형 변수로 변환을 해야만 하며, 데이터 탐색 시에도 범주형 변수로 변환하여 분포 형태나 집단 간 비교를 하게 되므로 이번 포스팅은 활용도가 매우 높다.

```
> score_d.f
  student_id stat_score
1      s01         56
2      s02         94
3      s03         82
4      s04         70
5      s05         64
6      s06         82
7      s07         78
8      s08         80
9      s09         76
10     s10         78
```

cut() 을 이용한 범주형 변수 변환

옵션을 적절히 사용 해야 되고 직관적인 수식을 적지 않으므로 헷갈리기 쉽다.

```
score_d.f <- transform(score_d.f, stat_score_1 = cut(stat_score, breaks = c(0, 60, 70, 80, 90, 100),
  include.lowest = TRUE,
  right = FALSE,
  labels = c("가", "양", "미", "우", "수")))
```

옵션	설명
Include.lowest =	T일 경우 구성요소 값이 최소값과 같아도 변환 시킨다.
right =	T일 경우 $a < x \leq b$ 와 같이 오른쪽 숫자까지 포함해서 등급 부여 F일 경우 $a \leq x < b$ 와 같이 왼쪽 숫자만 포함해서 등급 부여
labels =	c("등급1","등급2",...) 와 같이 등급을 부여함

ifelse() 을 이용한 범주형 변수 변환

cut() 대비 수식등호와 부등호를 직접 입력하니 직관적으로 분석가가 원하는 범주로 수식을 적을 수 있지만 범주의 수준이 많아질 수록 괄호 수가 많아져서 유의해야 한다.

```
score_d.f <- transform(score_d.f, ifelse = ifelse(stat_score < 60, "가",
                                                ifelse(stat_score >= 60 & stat_score < 70, "양",
                                                ifelse(stat_score >= 70 & stat_score < 80, "미",
                                                ifelse(stat_score >= 80 & stat_score < 90, "우", "수" )))) )
```

within() 을 이용한 범주형 변수 변환

within() 함수를 순서형 요인변수 만들 때 위 셋 중에서 가장 많이 사용하는 편

within() 함수는 먼저 새로 만들 변수 "윗인 = character(0)" 이라고 해서 문자형 변수라고 신규생성/지정을 해 주고 시작한다.

수식 등호, 부등호로 구간 설정하고, 제일 마지막 줄에 factor() 함수를 이용해 level = c("수", "우", "미", "양", "가") 라고 해서 수준을 지정해 줄 수 있다. 성적은 순서(order)가 있으므로 level 에 지정한 순서가 "윗인" 요인 변수의 level 순서가 된다.

```
score_d.f <- within( score_d.f, {
  stat_score_6 = character(0)
  stat_score_6[ stat_score < 60 ] = "가"
  stat_score_6[ stat_score >=60 & stat_score < 70 ] = "양"
  stat_score_6[ stat_score >=70 & stat_score < 80 ] = "미"
  stat_score_6[ stat_score >=80 & stat_score < 90 ] = "우"
  stat_score_6[ stat_score >=90 ] = "수"

  stat_score_6 = factor(stat_score_6, level = c("수", "우", "미", "양", "가"))
})
```

```
> score_d.f
  student_id stat_score stat_score_1 ifelse 윗인
1         s01         56           가     가     가
2         s02         94           수     수     수
3         s03         82           우     우     우
4         s04         70           미     미     미
5         s05         64         양     양     양
6         s06         82           우     우     우
7         s07         78           미     미     미
8         s08         80           우     우     우
9         s09         76           미     미     미
10        s10         78           미     미     미
```

외부데이터 불러오기

2018년 5월 18일 금요일 오후 7:02

■ 외부데이터 불러오는 함수의 종류

1. read.fwf()
2. read.table()
3. read.csv()
4. read.xlsx()
5. readLines()
6. DB 에서 데이터 가져오기

■ 외부 데이터를 불러오고 하는 과정

1. Str() 함수를 이용해서 데이터 구조 파악하기
2. Head() or tail() 함수를 이용해서 데이터 몇 개 미리보기
3. 결측 값 확인 및 처리 (NA 값 처리)
4. 탐색적 데이터 분석 (특이값/ 영향치 확인 및 처리 등)

1. 예제

1. read.fwf()

fwf (Fixed width file)

데이터의 길이가 일정한 간격, 고정된 폭의 외부데이터를 불러올 때

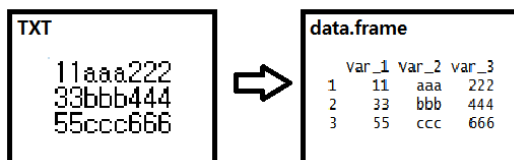
기계에서 일정한 간격으로 흐트러짐 없이 쏟아져 나오는 센서 데이터와 같이 **일정한 간격, 고정된 구조**의 데이터라고 확신이 있을 때만 사용

1.1 사용법

read.fwf("디렉토리 경로", widths = c(간격 설정), col.names = c(변수명 설정))

1.2 예제

data_fwf <- read.fwf("c:\data\Wwf\Wwf.txt", widths = c(2,3,3), col.names = c("Var_1", "Var_2", "var_3"))



2. read.table()

다수의 변수에 대해 다수의 관찰 값이 2차원 형태로 구성된 데이터 파일을 불러오는데 **read.table()** 함수를 사용합니다.

read.csv() 와 read.table() 차이점

read.csv()함수

- sep="," 옵션이 필요 없다는 점과
- 파일명 끝이 "dataset_name.csv"로 끝난다는 점

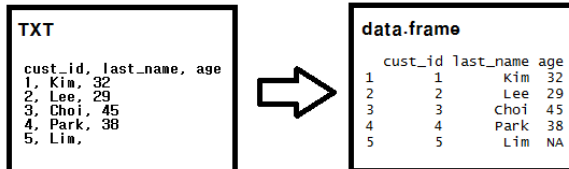
(.csv 파일은 'comma separated values'의 약자로서 콤마로 구분자가 되어 있기 때문)

1.1 사용법

```
read.table("디렉토리 경로", header = T, sep = ",", stringAsFactor = F, na.strings = "na")
```

1.2 예제

```
dataset_1 <- read.table("C:/Users/user/Documents/R/dataset_1.txt",  
+                       header = TRUE, # 변수명  
+                       sep = ",", # 구분자  
+                       stringsAsFactor = FALSE, # 문자형 데이터를 요인으로 인식할지 여부  
+                       na.strings = "" # 결측값 표시  
+                       )
```



3. Csv 파일을 로드하는 방법 :

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

4. Xlsx파일을 로드하는 방법 :

```
install.packages("xlsx")  
library(xlsx)
```

```
dept <- read.xlsx("dept.xlsx",1) # 1 은 엑셀파일의 첫 번째 sheet 를 의미함
```

```
> dept <- read.xlsx("c:\\data\\dept.xls",1)  
> dept  
  DEPTNO      DNAME      LOC  
1      50      DDD NEW YORK  
2      70      aa      bb  
3      20 RESEARCH DALLAS  
4      30      SALES      AA  
5      40 OPERATIONS BOSTON
```

5. Txt 파일을 로드하는 방법 :

```
niv <- readLines("NIV.txt")
```

```
> niv <- readLines("c:\\data\\NIV.txt")  
Warning message:  
In readLines("c:\\data\\NIV.txt") :  
  incomplete final line found on 'c:\\data\\NIV.txt'  
> head(niv)  
[1] "Gen 1:1 In the beginning God created the heavens and the earth."  
[2] "Gen 1:2 Now the earth was formless and empty, darkness was over the surface of the deep, and the Spirit of God was hovering over the waters."  
[3] "Gen 1:3 And God said, \"Let there be light,\" and there was light."  
[4] "Gen 1:4 God saw that the light was good, and he separated the light from the darkness."  
[5] "Gen 1:5 God called the light \"day,\" and the darkness he called \"night.\" And there was evening, and there was morning, the first day."  
[6] "Gen 1:6 And God said, \"Let there be an expanse between the waters to separate water from water.\""
```

6. Database에서 R로 데이터 로드하는 방법 :

```
install.packages('DBI')
install.packages('RJDBC')
library(DBI)
library(RJDBC)

driver <- JDBC('oracle.jdbc.driver.OracleDriver', 'c:\\data\\ojdbc6.jar')
oracle_db <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'scott', 'tiger')
emp_query <- 'select * from emp'
emp_data <- dbGetQuery(oracle_db, emp_query)

emp_data
```

문제 205. Sh 계정의 sales 테이블을 R의 변수로 로드해보자.

```
oracle_db2 <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'sh', 'sh')
```

```
sal_query <- 'select * from sales'
sal_data <- dbGetQuery(oracle_db2, sal_query)
```

```
sal_data
```

```
> oracle_db2 <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'sh', 'sh')
Error in .jcall(drv@jdrv, "Ljava/sql/Connection;", "connect", as.character(url)[1], :
  java.sql.SQLException: ORA-28000: the account is locked
> oracle_db2 <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'sh', 'sh')
> sal_query <- 'select * from sales'
> sal_data <- dbGetQuery(oracle_db2, sal_query)
> sal_data
```

	PROD_ID	CUST_ID	TIME_ID	CHANNEL_ID	PROMO_ID	QUANTITY_SOLD	AMOUNT_SOLD
1	13	987	1998-01-10 00:00:00	3	999	1	1232.16
2	13	1660	1998-01-10 00:00:00	3	999	1	1232.16
3	13	1762	1998-01-10 00:00:00	3	999	1	1232.16
4	13	1843	1998-01-10 00:00:00	3	999	1	1232.16
5	13	1948	1998-01-10 00:00:00	3	999	1	1232.16
6	13	2273	1998-01-10 00:00:00	3	999	1	1232.16

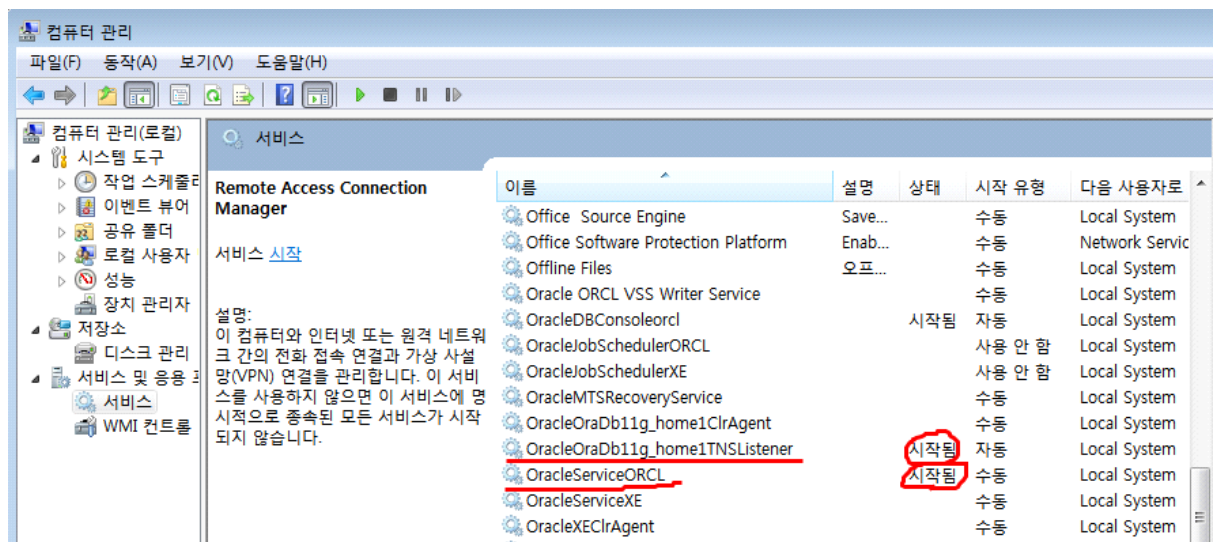
오라클 연동

2018년 5월 16일 수요일 오전 9:48

문제 140. 오라클과 R을 연동 하시오.

1. 오라클 서비스를 올린다.
2. 오라클 리스너의 상태를 확인한다.
3. 리스너를 통해서 오라클에 접속이 되는지 확인한다.
(자바를 설치 해야된다)

에러: JAVA_HOME cannot be determined from the Registry
Error: 패키지 'rJava'는 로드되어질 수 없습니다



```
C:\Users\Administrator>lsnrctl status

LSNRCTL for 64-bit Windows: Version 11.2.0.1.0 - Production on 16-5월 -2018 16:41:56

Copyright (c) 1991, 2010, Oracle. All rights reserved.

<DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC1522)))>에 연결되었습니다
리스너의 상태
-----
별칭          LISTENER
버전          TNSLSNR for 64-bit Windows: Version 11.2.0.1.0 - Production
시작 날짜     16-5월 -2018 09:40:37
업타임       0 일 7 시간. 1 분. 30 초
업타임스 기준 off
보안          ON: Local OS Authentication
SNMP          OFF
리스너 로그 파일 C:\app\Administrator\product\11.2.0\bdhhome_1\network\admin\listener.ora
리스너 로그 파일 c:\app\Administrator\diag\tnslsnr\WMSDN-SPECIAL\listener\alert\log.xml
현재 요약 정보...
<DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(PIPENAME=\\.\pipe\EXTPROC1522ipc)))>
<DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=127.0.0.1)(PORT=1522)))>
서비스 요약...
"CLRExtProc" 서비스는 1개의 인스턴스를 가집니다.
"CLRExtProc" 인스턴스<UNKNOWN 상태>는 이 서비스에 대해 1 처리기를 가집니다.
"orcl" 서비스는 1개의 인스턴스를 가집니다.
"orcl" 인스턴스<READY 상태>는 이 서비스에 대해 1 처리기를 가집니다.
"orclXDB" 서비스는 1개의 인스턴스를 가집니다.
"orcl" 인스턴스<READY 상태>는 이 서비스에 대해 1 처리기를 가집니다.
명령이 성공적으로 수행되었습니다
```

```
C:\Users\Administrator>sqlplus scott/tiger@orcl
```

```
SQL*Plus: Release 11.2.0.1.0 Production on 수 5월 16 16:44:38 2018
```

```
Copyright (c) 1982, 2010, Oracle. All rights reserved.
```

다음에 접속됨:

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

```
install.packages('DBI')
install.packages('RJDBC')
library(DBI)
library(RJDBC)
```

```
driver <- JDBC('oracle.jdbc.driver.OracleDriver', 'c:\\data\\ojdbc6.jar')
```

```
oracle_db <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1522/orcl', 'scott', 'tiger')
```

```
emp_query <- 'select * from emp'
```

```
emp_data <- dbGetQuery(oracle_db, emp_query)
```

```
emp_data
```

	EMPNO	ENAME	JOB	MGR		HIREDATE	SAL	COMM	DEPTNO	GRADE
1	7839	KING	PRESIDENT	NA	1981-11-17	00:00:00	5000	NA	10	A
2	7698	BLAKE	MANAGER	7839	1981-05-01	00:00:00	2850	NA	30	B
3	7782	CLARK	MANAGER	7839	1981-05-09	00:00:00	2450	NA	10	B
4	7566	JONES	MANAGER	7839	1981-04-01	00:00:00	2975	NA	20	B
5	7654	MARTIN	SALESMAN	7698	1981-09-10	00:00:00	1250	1400	30	D
6	7499	ALLEN	SALESMAN	7698	1981-02-11	00:00:00	1600	300	30	C
7	7844	TURNER	SALESMAN	7698	1981-08-21	00:00:00	1500	0	30	C
8	7900	JAMES	CLERK	7698	1981-12-11	00:00:00	950	NA	30	F
9	7521	WARD	SALESMAN	7698	1981-02-23	00:00:00	1250	500	30	D
10	7902	FORD	ANALYST	7566	1981-12-11	00:00:00	3000	NA	20	A
11	7369	SMITH	CLERK	7902	1980-12-09	00:00:00	800	NA	20	F
12	7788	SCOTT	ANALYST	7566	1982-12-22	00:00:00	3000	NA	20	A
13	7876	ADAMS	CLERK	7788	1983-01-15	00:00:00	1100	NA	20	D
14	7934	MILLER	CLERK	7782	1982-01-11	00:00:00	1300	NA	10	D

문제 141. Emp2 테이블의 데이터를 ,emp2_data에 로드하고 통신사, 나이를 가지고 아래의 그래프를 그리시오.

```
func2 <- function() {
```

```
  res1<- menu(colnames(emp2_data), title='토달 값을 구할 컬럼번호 입력하세요~')
```

```
  res2<- menu(colnames(emp2_data), title='그룹핑할 컬럼번호 입력하세요~')
```

```
  x1 <- menu( c("막대그래프","원형그래프"), title='원하는 그래프의 숫자를 선택하세요 ')
```

```
  r1<-colnames(emp2_data)[res1]
```

```
  r2<-colnames(emp2_data)[res2]
```

```
  q<-tapply(emp2_data[,r1], emp2_data[,r2],sum)
```

```
  switch(x1,
```

```
    {
```

```
      q[is.na(q)] <- 0
```

```
      barplot(q, col = rainbow(nrow(q)), main = paste( r2, '별', r1, '총합' ), beside = T, ylim = c(0,max(q)*1.4))
```

```
      legend("topright", rownames(q),title = paste(r2,' 구분' ),inset = 0,fill = rainbow(nrow(q)),cex=0.8)
```

```

    },
    {
      label<-paste(sort(unique(emp2_data[,r2])), round(q/sum(q) * 100,1),'%')
      pie(q,col=rainbow(nrow(q)),label=label,main = paste( r2, '별', r1,'총합' ))
    }
  )
}

```

> func2()

토탈 값을 구할 컬럼번호 입력하세요~

1: EMPNO	2: ENAME	3: AGE	4: BIRTH	5: MAJOR	6: EMAIL	7: MOBILE
8: ADDRESS	9: TELECOM	10: GRADE	11: BIRTH_DAY	12: RNK		

선택: 3

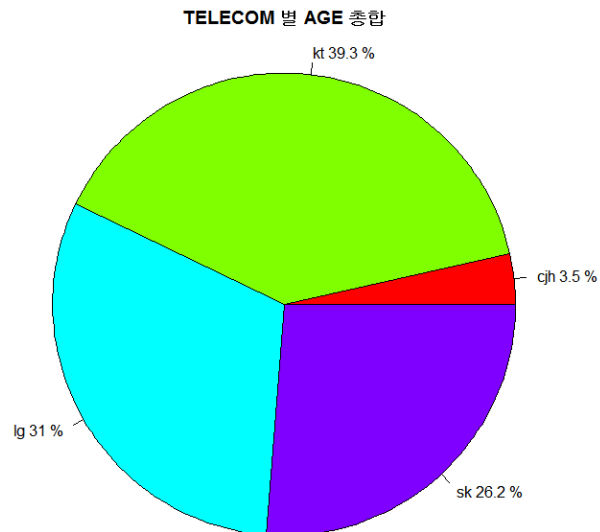
그룹핑할 컬럼번호 입력하세요~

1: EMPNO	2: ENAME	3: AGE	4: BIRTH	5: MAJOR	6: EMAIL	7: MOBILE
8: ADDRESS	9: TELECOM	10: GRADE	11: BIRTH_DAY	12: RNK		

선택: 9

원하는 그래프의 숫자를 선택하세요

- 1: 막대그래프
- 2: 원형그래프



R ggplot2 히스토그램

2018년 5월 22일 화요일 오후 4:53

데이터셋을 받으면 제일 먼저 하는 일이 데이터의 구조를 파악하고, 변수명, 변수별 데이터 유형(숫자형, 문자형, 논리형), 결측값 여부, 이상치/영향치 여부, 데이터의 퍼진 정도/분포 모양 등을 탐색하게 된다.

히스토그램(Histogram)은 연속형 변수를 일정한 구간(binwidth)으로 나누어서 빈도수를 구한 후에 이를 막대그래프로 그린 그래프이다.

2. 예제

예제 1.

ggplot2 패키지를 library()로 호출한 후에 ggplot() 함수의 +geom_histogram() 함수를 사용하여 default 옵션으로 히스토그램을 그리면 아래와 같다.

소스

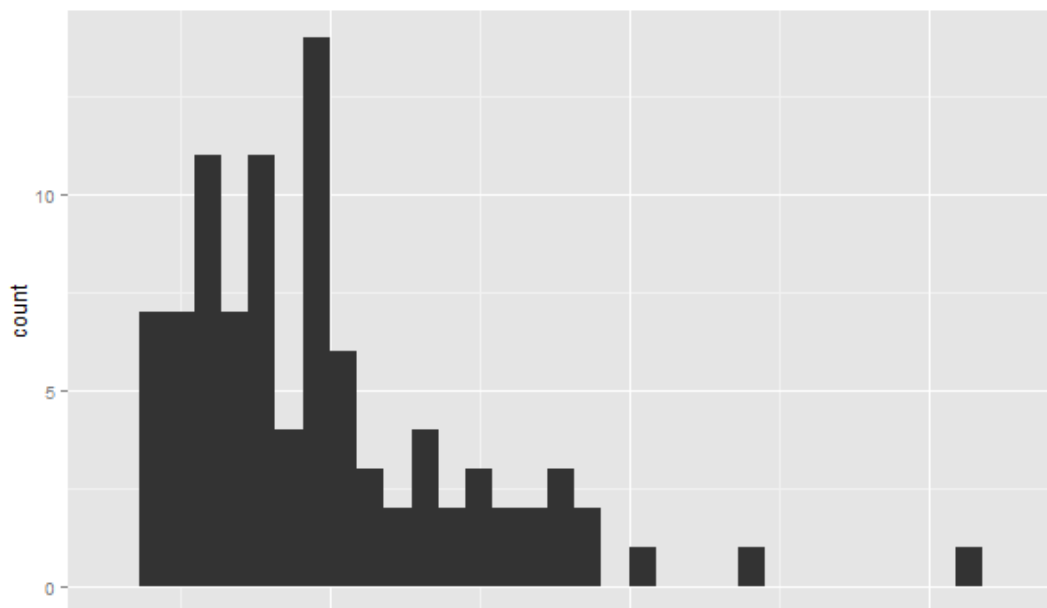
```
install.packages("ggplot2") # ggplot2 패키지 설치
library(ggplot2)

# binwidth defaulted to range/30
ggplot(Cars93, aes(x=Price)) + geom_histogram()
```

결과

```
> ggplot(Cars93, aes(x=Price)) + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

binwidth를 설정하지 않아서 range/30 디폴트 기준으로 binwidth를 계산해서 그렸다는 뜻





예제 2.

Binwidth를 range/30 으로 설정해보자.

소스

```
range(Cars93$Price) # 7.4 ~ 61.9
```

```
[1] 7.4 61.9
```

```
diff(range(Cars93$Price)) # 54.5
```

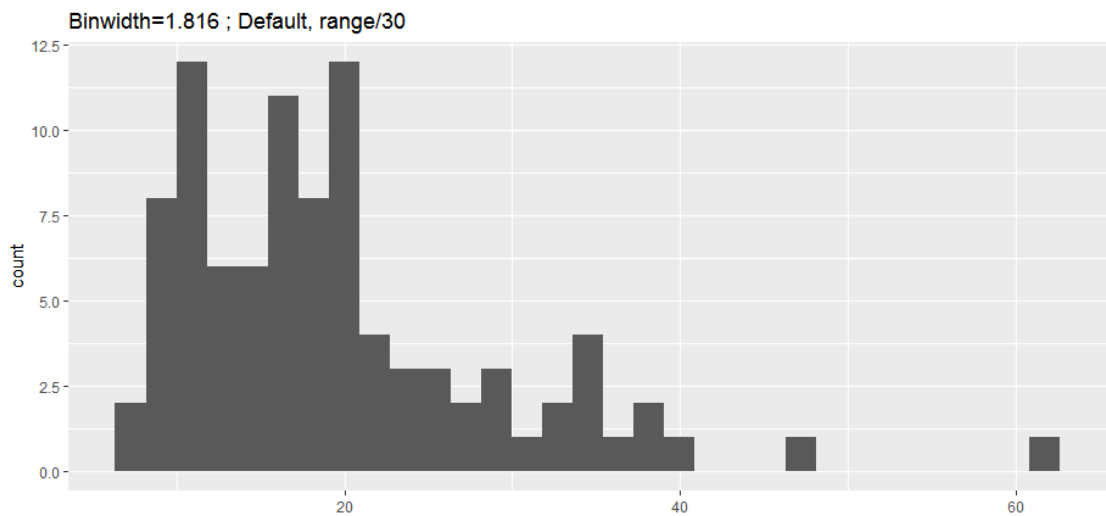
```
[1] 54.5
```

```
diff(range(Cars93$Price))/30 # 1.816
```

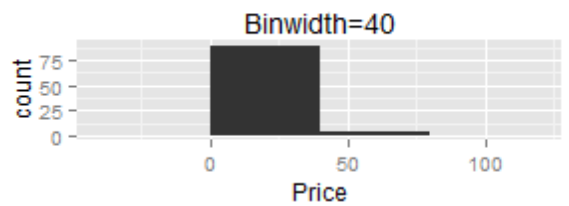
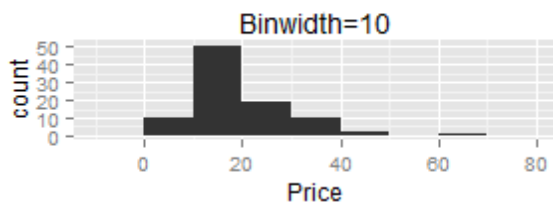
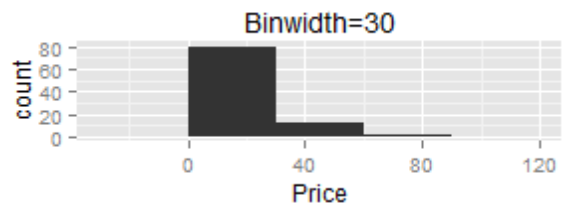
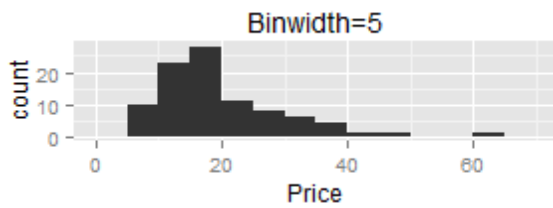
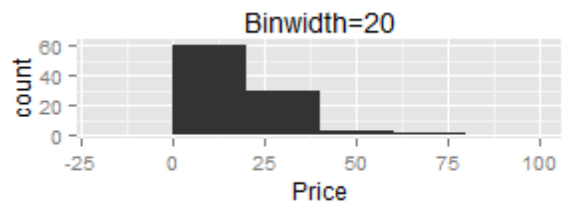
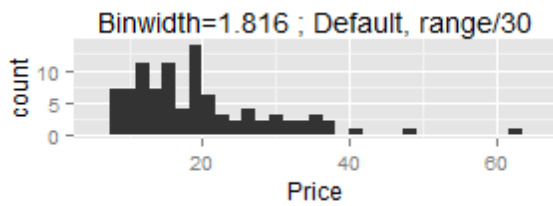
```
[1] 1.816667
```

```
ggplot(Cars93, aes(x=Price)) + geom_histogram(binwidth=1.816) + ggtitle("Binwidth=1.816 ; Default, range/30")
# x축                # binwidth 설정                # 그래프 제목설정
```

결과



히스토그램에서 중요하면서 어려운 문제 중의 하나가 bin 개수를 몇 개로 할 것인가, 다른 말로 binwidth를 몇으로 할 것인가이다. bin 개수가 너무 많으면 (즉, binwidth가 너무 좁으면) 이빨 빠진 머리빗처럼 데이터의 분포 모양을 보기에 부적할 수가 있다. 반면에 bin 개수가 너무 적으면 (즉, binwidth가 너무 넓으면) 너무 많은 도수가 하나의 bin에 통쳐져서 막대기둥 한두개만 덩그러니 서있게 되어 이 또한 데이터의 분포 모양을 파악하는데 도움이 안되게 된다.



예제 3.

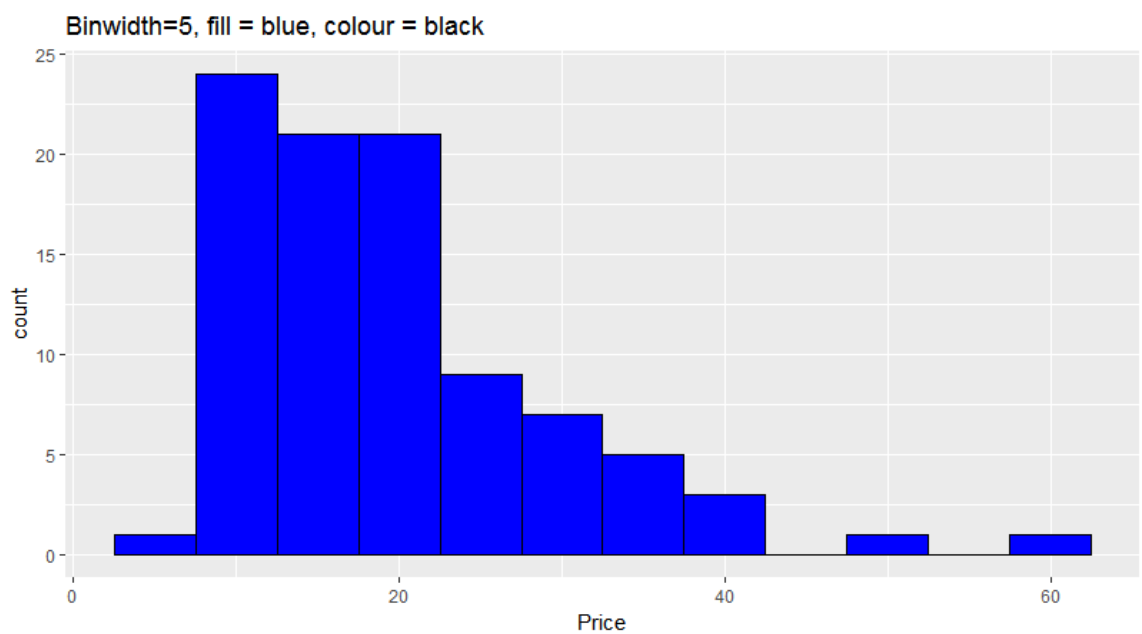
채우기 색, 경계선 색을 설정해보자.

소스

```
# 채우기 색, 경계선 색 : geom_histogram(binwidth, fill, colour)
```

```
ggplot(Cars93, aes(x=Price)) + geom_histogram(binwidth=5, fill = "blue", colour = "black") +  
ggtitle("Binwidth=5, fill = blue, colour = black")
```

결과



예제 4.

facet_grid() 를 써서 요인(factor)/집단/그룹별로 히스토그램을 구분해서 그려보도록 하자.

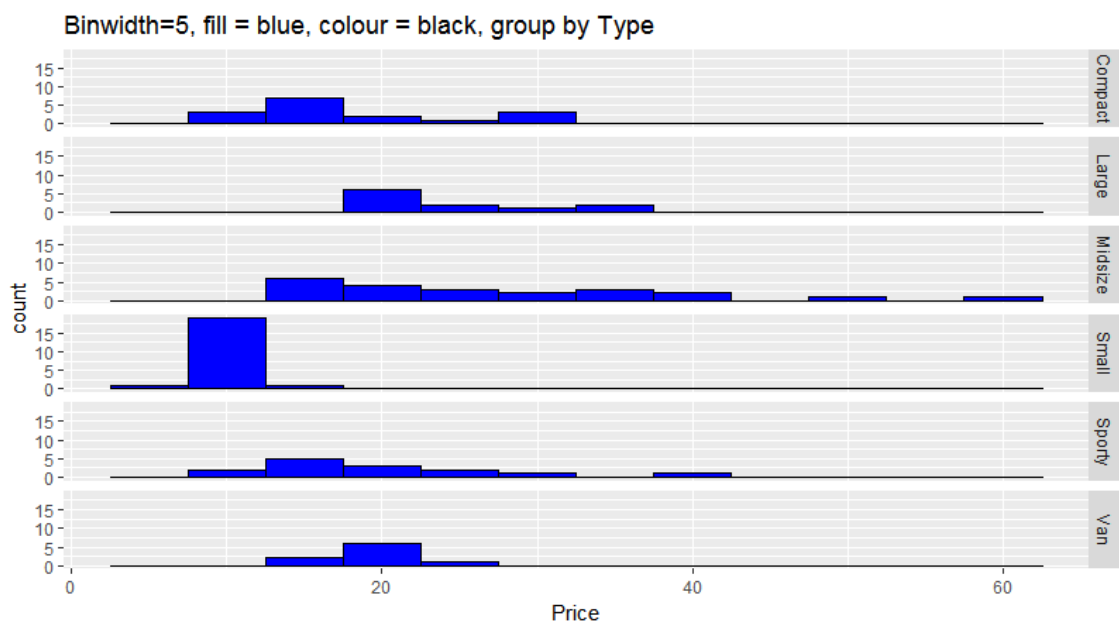
단, facet_grid()에 들어가는 변수는 요인(factor)형 변수이어야 합니다

소스

```
# 요인(factor) 여부 확인, levels 확인
class(Cars93$Type); levels(Cars93$Type)
[1] "factor"
[1] "Compact" "Large" "Midsize" "Small" "Sporty" "Van"

# 요인/집단/그룹(factor)별로 나누어서 히스토그램 그리기
ggplot(Cars93, aes(x=Price)) + geom_histogram(binwidth=5, fill = "blue", colour = "black") +
  ggtitle("Binwidth=5, fill = blue, colour = black, group by Type") + facet_grid(Type ~ .)
```

결과



위의 히스토그램처럼 자동차의 유형(Type)인 'Compact', 'Large', 'Midsize', 'Small', 'Sporty', 'Van' 의 6개 유형별로 가격(Price)의 히스토그램을 그려보면 서로 한눈에 비교가 가능하니 매우 유용하다.

예제 5.

세로로 세워서 그래프를 그린 후에 비교를 하려면 + facet_grid(. ~ Type) 처럼 괄호안의 기입 순서를 바꾸어주면 된다.

소스

```
# 요인/집단/그룹(factor)별로 나누어서 히스토그램 그리기
ggplot(Cars93, aes(x=Price)) + geom_histogram(binwidth=5, fill = "blue", colour = "black") +
  ggtitle("Binwidth=5, fill = blue, colour = black, group by Type") + facet_grid(. ~ Type) # 수직
```

결과

