

R 교재 x 목차

2018년 6월 22일 금요일 오후 5:28

- [R 설치](#)
- [R 을 왜 사용해야하는가?](#)
- [R](#)
- [R의 자료구조](#)
- [R의 연산자](#)
- [R의 함수](#)
 - [1. 문자함수](#)
 - [2. 숫자 함수](#)
 - [3. 날짜함수](#)
 - [4. 일반 함수](#)
 - [5. 변환 함수](#)
 - [6. 그룹 함수](#)
- [R의 조인](#)
- [R에서의 서브쿼리](#)
- [R에서 그래프 그리는 방법](#)
 - [1. 막대 그래프](#)
 - [2. 원형 그래프](#)
 - [3. Plot 그래프](#)
- [R의 막대 그래프 이론 설명](#)
 - [4. 구글에서 제공하는 그래프](#)
 - [5. 지도 그래프](#)
 - [6. 소리 시각화](#)
 - [8. 사분위수 그래프](#)
 - [7. 워드 클라우드](#)
 - [8. 사분위수 그래프](#)
- [R의 막대 그래프 이론 설명](#)
 - [4. 구글에서 제공하는 그래프](#)
 - [5. 지도 그래프](#)

- [6. 소리 시각화](#)
 - [8.사분위수 그래프](#)
 - [7. 워드 클라우드](#)
 - [8. 사분위수 그래프](#)
-
- [IF 문의 문법](#)
-
- [R shiny](#)

R 설치

2018년 5월 8일 화요일 오전 9:57

1. <http://healthstat.snu.ac.kr/CRAN/> 접속
2. [Download R for Windows](#) 클릭
3. [install R for the first time.](#) 클릭
4. Download R 3.5.0 for Windows 클릭
5. c나 D 드라이브 밑에 data라는 폴더를 만들고
카페에있는
dept.csv
emp.csv
salgrade.csv 를 넣어둔다.
6. R에서 csv 불러온다.
- 파일 -> 작업 디렉토리 변경 -> data 파일 지정
emp <- read.csv("emp.csv", header=T) 입력

attach(emp) 한번 치고,
tapply(sal,list(deptno,job),sum) 입력

결과)

```
> emp <- read.csv("emp.csv", header=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
> |
```

R studio 는 약간 sql gate라고 보며되고 R은 CMD창이라고 보며된다.

R 을 왜 사용해야하는가?

2018년 5월 8일 화요일 오후 3:13

1. R is free
2. data 분석을 위해서 가장 많이 쓰는 통계 플랫폼
3. 복잡한 데이터를 다양한 그래프로 표현 할 수 있다.
4. 분석을 위한 데이터를 쉽게 저장하고 조작 할 수 있다.
5. 누구든지 유용한 패키지를 생성해서 공유 할 수 있고,
새로운 기능에 대한 전달이 빠르다.
6. 어떠한 os에도 설치가 가능하다.

문제1. 아래의 결과를 R로 구현하시오.

	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

답)

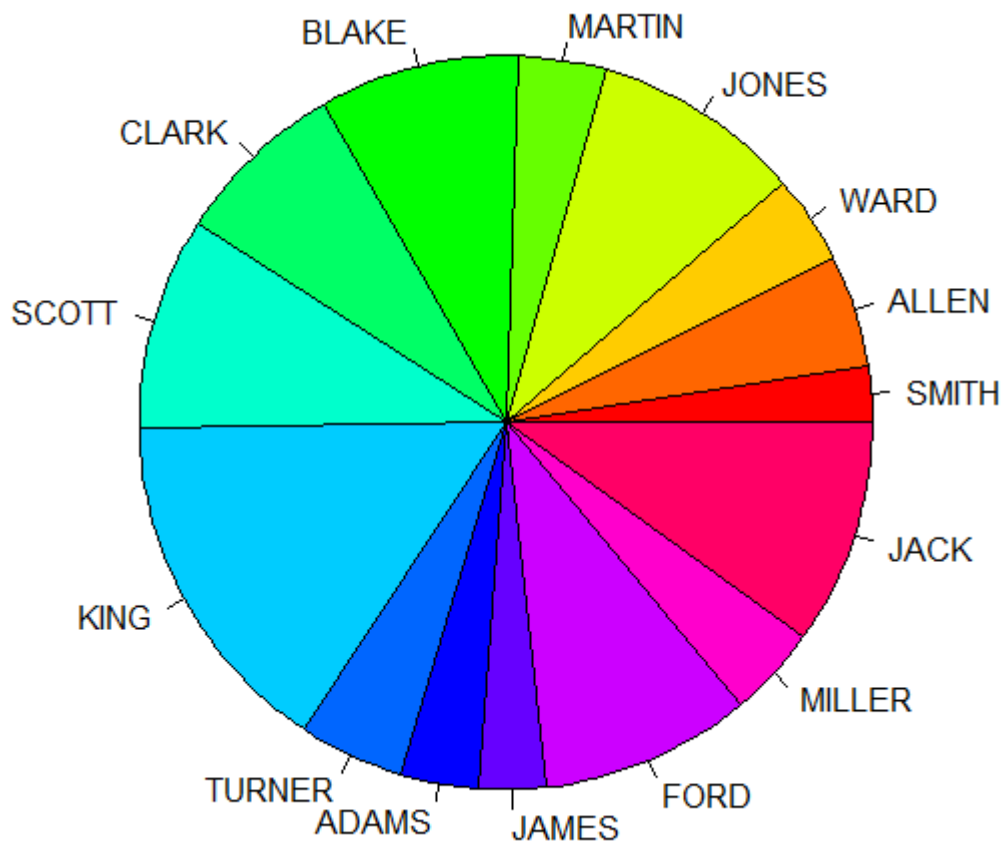
```
emp <- read.csv("emp.csv", header=T)
```

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

문제2. 사원테이블의 월급을 시각화 하시오.

답)

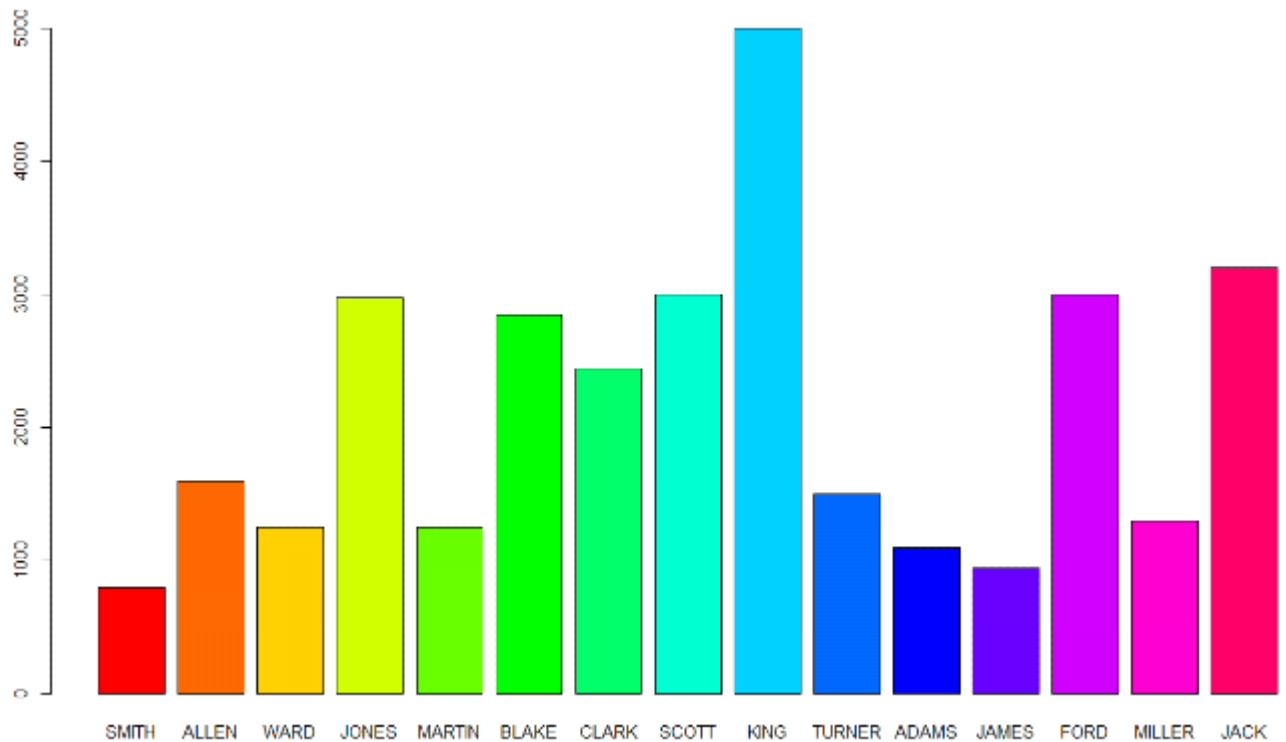
```
pie(emp$sal, labels=emp$ename, col=rainbow(15) )
```



문제3. 사원테이블의 월급을 막대 그래프로 시각화 하시오.

답)

```
barplot(emp$sal, col=rainbow(15), names.arg=emp$ename)
```



문제4. 'R에서 제공하는 유용한 패키지를 생성해서 공유할 수 있고 새로운 기능에 대한 전달이 빠르다' 라는

장점을 코드로 구현해 보시오.

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

입력하고 cran 을 korea 선택

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

```
library(networkD3)
```

```
library(dplyr)
```

<-- library를 사용하겠다

카페 연결망 분석-조대현 들어가서 소스 붙여넣으셈

```
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',
```

```

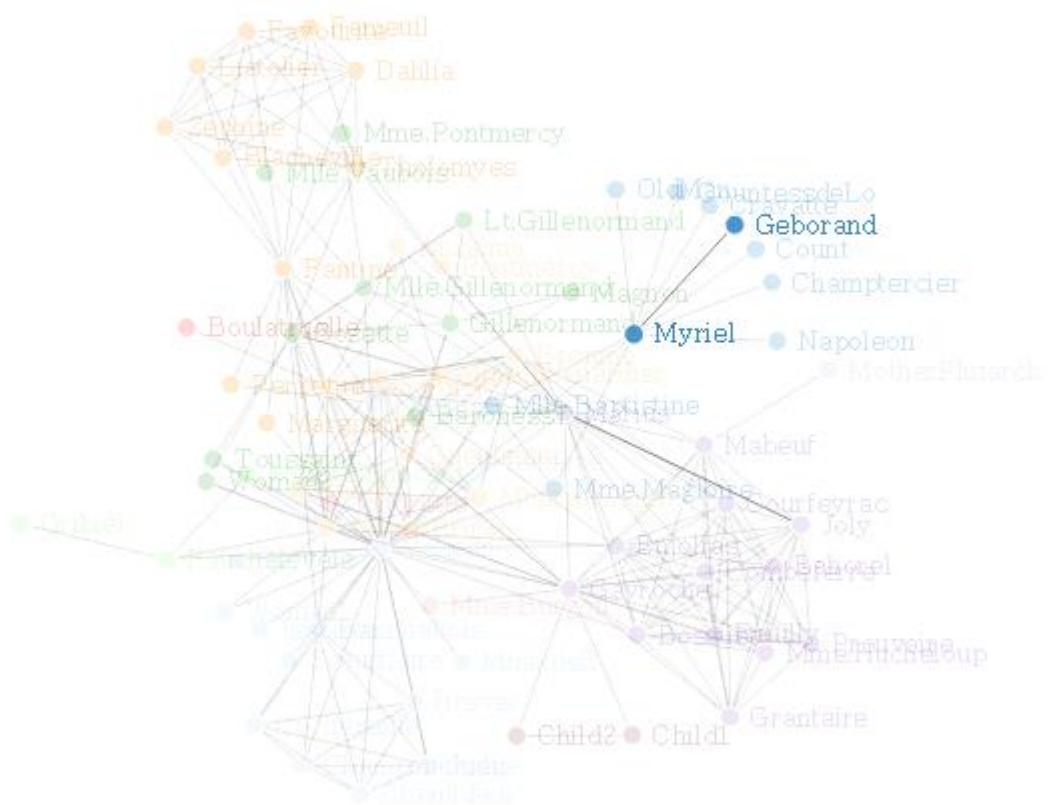
D3_network_LM = forceNetwork(links = linklinks, nodes = linknodes,
                             Source = 'source', Target = 'target',
                             NodeID = 'name', Group = 'group', opacityNoHover = TRUE,
                             zoom = TRUE, bounded = TRUE,
                             fontSize = 15,
                             linkDistance = 75,
                             opacity = 0.9)

```

```
D3_network_LM
```

```
# html 발사
```

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



R studio

```

> emp <- read.csv("c:\\data\\emp.csv", header=T);
> emp
  empno  ename      job mgr  hiredate  sal comm deptno
1   7369  SMITH    CLERK 7902 1980-12-17  800   NA      20
2   7499  ALLEN  SALESMAN 7698 1981-02-20 1600  300      30
3   7521  WARD    SALESMAN 7698 1981-02-22 1250  500      30
4   7566  JONES   MANAGER 7839 1981-04-02 2975   NA      20
5   7654  MARTIN  SALESMAN 7698 1981-09-28 1250 1400      30

```

```

> emp <- read.csv("c:\\data\\emp.csv", header=T);
> emp
  empno  ename      job mgr  hiredate  sal comm deptno
1   7369 SMITH    CLERK 7902 1980-12-17   800   NA     20
2   7499 ALLEN  SALESMAN 7698 1981-02-20  1600   300     30
3   7521  WARD  SALESMAN 7698 1981-02-22  1250   500     30
4   7566 JONES   MANAGER 7839 1981-04-02  2975   NA     20
5   7654 MARTIN  SALESMAN 7698 1981-09-28  1250  1400     30
6   7698 BLAKE   MANAGER 7839 1981-05-01  2850   NA     30
7   7782 CLARK   MANAGER 7839 1981-06-09  2450   NA     10
8   7788 SCOTT   ANALYST 7566 1987-04-19  3000   NA     20
9   7839 KING    PRESIDENT  NA 1981-11-17  5000   NA     10
10  7844 TURNER  SALESMAN 7698 1981-09-08  1500    0     30
11  7876 ADAMS    CLERK 7788 1987-05-23  1100   NA     20
12  7900 JAMES    CLERK 7698 1981-12-03   950   NA     30
13  7902 FORD     ANALYST 7566 1981-12-03  3000   NA     20
14  7934 MILLER   CLERK 7782 1982-01-23  1300   NA     10
15  9292 JACK     CLERK 7782 1982-01-23  3200   NA     70

```


R의 자료구조

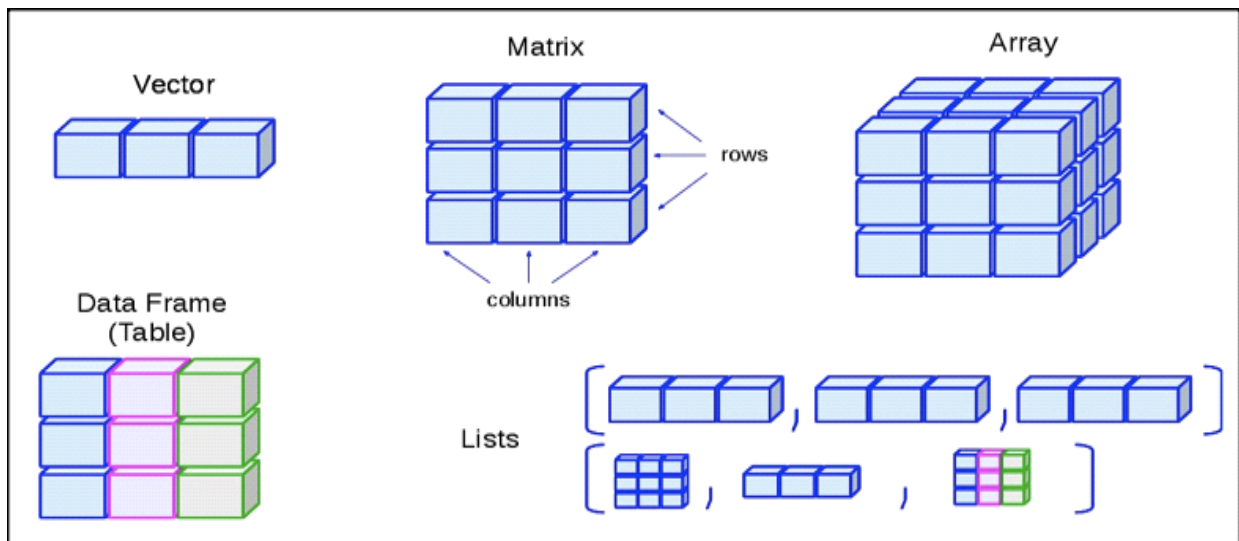
2018년 5월 8일 화요일 오후 4:26

1. **vector** : 같은 데이터 타입을 갖는 1차원 배열구조
2. **matrix** : 같은 데이터 타입을 갖는 2차원 배열구조
3. **array** : 같은 데이터 타입을 갖는 다차원 배열구조
4. **data.frame** : 각각의 데이터 타입을 갖는 컬럼으로 이루어진 2차원 배열구조

(오라클의 테이블과 유사)

5. **list** : 서로 다른 데이터 구조(vector, data, frame, matrix, array)인 데이터

타입이 중첩된 구조



문제5. 데이터 프레임이 데이터 베이스의 테이블과 유사한 R의 자료형이다. emp구조가 data frame임을

확인하시오.

답)

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 ...
```

문제6. emp 데이터 프레임에서 이름과 월급을 출력하시오.

emp[행, 열]

emp[,c("ename","sal")]

c : combine

```
> 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
```

문제7. 월급이 3000인 직원들의 이름과 월급을 출력하시오.

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

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

문제8. 월급이 2000 이상인 직원들의 이름과 월급을 출력하시오.

```
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
```

R의 연산자

2018년 5월 9일 수요일 오전 9:48

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

백터화된 연산 vs 백터화되지 않은 연산

```
ex) x <- c(1,2,3)
     x > c(1,1,1) & x < c(3,3,3)
> x <- c(1,2,3)
>
>
> str(x)
num [1:3] 1 2 3

x <-1
x >-2 && x<2
```

기타 비교 연산자

SQL	vs	R
1. in		1. %in%
2. like		2. grep
3. is null		3. is.na
4. between .. and		4. emp\$sal >= 1000 & emp\$sal <=3000

문제9. 직업이 SALESMAN 이 아닌 직원들의 이름과 월급과 직업을 출력하시오.

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

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

문제10. 1981년 12월 03일에 입사한 직원들의 이름과 입사일을 출력하시오.

```
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
> |
```

문제11. 직업이 SALESMAN이고 월급이 1000이상인 직원들의 이름, 월급, 직업을 출력하시오.

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

  ename      job  sal
2  ALLEN SALESMAN 1600
3   WARD SALESMAN 1250
5  MARTIN SALESMAN 1250
10 TURNER SALESMAN 1500
```

문제12. 직업이 SALESMAN, ANALYST인 직원들의 이름과 직업을 출력하시오.

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

  ename      job
2  ALLEN SALESMAN
3   WARD SALESMAN
5  MARTIN SALESMAN
8   SCOTT ANALYST
10 TURNER SALESMAN
13  FORD ANALYST
```

이방법과 %in%을 사용할 수도 있다

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

문제13. 직업이 SALESMAN, ANALYST가 아닌 직원들의 이름과 직업을 출력하시오.

```
emp[emp$job != "SALESMAN" & emp$job != '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

이 방법과

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

문제14. 커미션이 null 인사원들의 이름과 월급과 커미션을 출력하시오.

?is.na < - is.na의 사용 방법을 알 수 있다.

```
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

R에서 NULL 값 3가지

1. NULL (아무것도 없다) -> is.null()

2. NA(결손값) -> is.na()

NA : Not Available

3. NAN(비수치) -> is.nan()

NAN : Not A Number

NULL(아무것도 없다) 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 활용하는 때는 반복문으로 처리할 오브젝트의 초기값을 NULL로 설정 할때

문제15. 월급이 1000에서 3000 사이인 직원들의 이름, 월급을 출력하시오.

```
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
```

- 위에 결과들을 보면 &(벡터화 된 연산)을 쓰고 있는데 현재 emp테이블을 보면 이미 벡터화 되었어서 &을 쓴다.

문제16. 이름의 첫 글자가 A로 시작하는 직원들의 이름과 월급을 출력하시오.

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

^ : 첫 번째

\$: 마지막,

. : 한자리수

* : wild card

```
      ename  sal
2   ALLEN 1600
11  ADAMS 1100
```

문제17. 이름의 끝 글자가 T로 끝나는 직원들의 이름과 월급을 출력하시오.

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

```
      ename  sal
8   SCOTT 3000
```

문제18. 이름의 두번째 철자가 M인 직원들의 이름과 월급을 출력하시오.

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

```
      ename  sal
1  SMITH 800
```

• 중복제거

SQL
distinct

vs

R
unique

문제19. 부서번호를 중복제거해서 출력하시오.

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

그냥 이렇게하면 가로로 출력되는데

```
install.packages("data.table")      패키지를 설치하고,  
library(data.table)                 라이브러리 하고,  
data.table(unique(emp$deptno))      이렇게 출력
```

```
      v1  
1:  20  
2:  30  
3:  10  
4:  70
```

만약 v1의 컬럼명을 바꾸고 싶으면

```
data.table("부서번호" = unique(emp$deptno))  
      부서번호  
1:          20  
2:          30  
3:          10  
4:          70
```

문제20. 직업을 중복제거해서 출력하시오.

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

• 정렬 작업

SQL	vs	R
order by		1. data.frame 의 order 옵션 2. doBy 패키지를 설치하고 order By 함수를 사용 (대소문자 구별!!)

문제21. 이름과 월급을 출력하는데, 월급이 높은 직원부터 출력하시오.

```
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

낮은 거부터 나오게 하려면 F 로하면된다.

문제22. 이름과 입사일을 출력하는데 먼저 입사한 직원부터 출력하시오.

```
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

문제23. 직업이 SALESMAN인 직원들의 이름과 월급을 출력하는데, 월급이 높은 직원부터 출력하시오.
(먼저 직업이 SALESMAN 먼저 뽑고 정렬한다. 오라클 실행순서랑 같다고 생각하면된다)

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

x에다가 SALESMAN을 담는다

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

	ename	sal
2	ALLEN	1600
10	TURNER	1500
3	WARD	1250
5	MARTIN	1250

- x 라는 변수를 지우고 싶을때

1. 변수 리스트 확인 `ls()`
2. `rm(x)` 하면 x 라는 변수가 사라진다.

문제24. (점심시간)직업이 ANALYST가 아닌 직원들의 이름, 월급, 직업을 출력하는데 월급이 높은 직원부터 출력하시오.

```
x <- emp[emp$job != 'ANALYST', c("ename", "sal", "job")]
x[order(x$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

문제25. 문제24번을 doBy 패키지의 orderBy 함수를 이용해서 출력하시오.

```
install.packages("doBy")           패키지 설치
library(doBy)
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

~- : 높은거 부터, ~ : 낮은거 부터

문제26. 카페에서 crime_loc.csv를 내려받고 R로 로드한 후에 살인이 일어나는 장소와 건수를 출력하는데, 살인이 일어나는 건수가 높은 것부터 출력하시오.

```
crime_loc <- read.csv("c:\\wwdata\\ww\\crime_loc.csv", header=T)
head(crime_loc, 10) <-- 위에 열개만 보겠다
```

```
> head(crime_loc, 10)
```

	범죄	장소	건수
1	절도	아파트	25389
2	절도	집	37787
3	절도	고속도로	151
4	절도	노상	62560
5	절도	상점	29977
6	절도	시장노점	1239
7	절도	숙박업소	9203
8	절도	병원	16053
9	절도	사무실	8416
10	절도	공장	3540

```
library(dplyr)
```

```
orderBy(~-건수, crime_loc[crime_loc$범죄 == "살인", c("장소", "건수")])
```

	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
98	유원지	13
96	교통	9
94	역대합실	8

R의 함수

2018년 5월 9일 수요일 오후 2:09

R함수의 종류

1. 문자 함수
2. 숫자 함수
3. 날짜 함수
4. 일반 함수
5. 변환 함수
6. 그룹 함수

1. 문자함수

SQL	vs	R
upper		toupper
lower		tolower
substr		substr
replace		gsub
concat		paste

문제27. 이름과 직업을 출력하는데 소문자로 출력하시오.

```
library(data.table)
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:	milller	clerk
15:	jack	clerk

문제28. 이름이 scott인 사원의 이름과 월급을 조회하는데 scott을 소문자로 조회해도 조회되게 코드를 구현

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

문제29. 이름의 두번째 철자가 M인 사원들의 이름과 월급을 출력하는데, substr 함수를 이용해서 출력

```
emp[substr(emp$ename,2,2) == 'M', c("ename","sal")] <-- 오라클은 2번째 자리부터 2자리인데
                                                    R은 2번째자리부터 2번째자리까지 가져옴
```

12345
 ename sal
 1 SMITH 800

문제30. 이름을 출력하고 그 옆에 이름의 첫번째 철자부터 세번째 철자까지 출력하시오.

결과를 세로로 출력하려면 data.table을 사용한다

```
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

문제31. 우리반 테이블을 R로 로드하고 이름과 성씨를 출력하시오.

1. SQL gate 접속

2. select * from emp2
order by empno; 결과를 csv로(쉼표) 내보내기
3. R에서 read해오기.

```
data.table(emp2$ename, substr(emp2$ename,1,1))
```

	v1	v2
1:	윤진민	윤
2:	송윤호	송
3:	은해찬	은
4:	김영토	김
5:	정호진	정
6:	김지우	김
7:	정인중	정
8:	이유진	이
9:	백광흠	백
10:	김원섭	김
11:	김광록	김
12:	장은희	장
13:	지윤철	지
14:	이상미	이

문제32. 우리반의 어떤 성씨가 있는지 중복제거해서 출력하시오.

```
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

문제33. 이름, 나이, 통신사를 출력하시오.

```
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	정인중	27	lg
8	이유진	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

문제34. 전공이 경제학과인 학생들의 이름, 나이, 주소를 출력하시오.

```
emp2[emp2$major == '경제학', 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

문제35. 이름과 월급을 출력하는데 월급을 출력할때에 숫자 0을 *로 출력하시오.

gsub : gsub('h','H', text) = 특정 텍스트에서 h를 H로 변경하겠다.

```
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**

문제36. 이름과 월급을 출력하는데 월급을 출력할때에 숫자 0,1,2를 *로 출력하시오.

```
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***

paste : 오라클의 연결 연산자와 비슷한 기능을 하는 함수

문제36-2. paste 함수를 이용해서 이름과 직업을 아래와 같이 출력하시오.

```
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
```

```
library(data.table)                                <-- data.table 함수를 선언해주고
data.table(paste(emp$ename, '의 직업은', emp$job))
```

문제36-3. 이름과 연봉을 아래와 같이 출력하시오.

SCOTT의 연봉은 36000 입니다.

```
-----
x<-emp[emp$ename == 'SCOTT', c("ename", "sal")]    <-- SCOTT의 이름과 월급을 x에다 담고,
data.table(paste(x$ename, '의 연봉은', x$sal*12, '입니다'))
```

```
1: SCOTT 의 연봉은 36000 입니다
```

만약 컴바인 안에서 가공한것을 가져오고 싶지만 컴바인안에는 테이블 안에 있는 결과만 가져올 수 있으므로 "sal *12" 를 할 수 없다.

2. 숫자 함수

SQL	vs	R
round		round
trunc		trunc
mod		%%
power		^ (ex: $2^3 = 8$)

문제37. 6의 9승을 출력하시오.

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

문제38. 10을 3으로 나눈 나머지값이 무엇인지 출력하시오.

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

문제39. 이름과 연봉을 출력하는데 연봉이 월급에 12를 곱해서 출력하고 컬럼명이 한글로 이름, 연봉으로 출력되게 하시오.

```
data.table(이름 = emp$ename, 월급 = emp$sal, 연봉 = emp$sal*12)  
   이름 월급 연봉  
1: SMITH   800 9600  
2: ALLEN  1600 19200  
3:  WARD  1250 15000  
4: JONES  2975 35700  
5: MARTIN 1250 15000  
6: BLAKE  2850 34200  
7: CLARK  2450 29400  
8: SCOTT  3000 36000  
9:  KING  5000 60000  
10: TURNER 1500 18000  
11: ADAMS  1100 13200  
12: JAMES   950 11400  
13:  FORD  3000 36000  
..
```

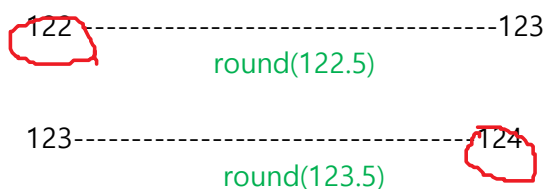
• R의 round 함수의 특징

round(122.5) = 122

round(123.5) = 124

왜그럴까??

R은 짝수를 좋아한다



- **round와 trunc 함수 설명**

3 6 7 8 . 7 8

3	6	7	8	.	7	8
-4	-3	-2	-1	0	1	2

`round(3678.78,1) = 3678.8`

`round(3678.78,-1) = 3680`

`trunc(3678.78,1) = 3678`

`trunc(3678.78,-1) = 3678` 으로 나오는데 이유는? **trunc는 소수점 이하만 가능하다.**

3. 날짜함수

오라클	vs	R
<code>sysdate</code>		<code>Sys.Date()</code>
<code>add_months</code>		<code>difftime</code>
<code>months_between</code>		사용자 정의 함수
<code>last_day</code>		사용자 정의 함수
<code>next_day</code>		사용자 정의 함수

문제40. 오늘 날짜를 출력하시오.

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

문제41. 이름, 입사한 날짜부터 오늘까지 총 몇일 근무했는지 출력하시오.

```
str(emp)
$ hiredate: Factor
hiredate는 문자형으로 적혀있다.
```

```
str(Sys.Date)
> str(Sys.Date)
function ()
```

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

	V1	V2	V3
1:	SMITH	1980-12-17	13657 days
2:	ALLEN	1981-02-20	13592 days
3:	WARD	1981-02-22	13590 days
4:	JONES	1981-04-02	13551 days
5:	MARTIN	1981-09-28	13372 days
6:	BLAKE	1981-05-01	13522 days
7:	CLARK	1981-06-09	13483 days
8:	SCOTT	1987-04-19	11343 days
9:	KING	1981-11-17	13322 days
10:	TURNER	1981-09-08	13392 days
11:	ADAMS	1987-05-23	11309 days
12:	JAMES	1981-12-03	13306 days
13:	FORD	1981-12-03	13306 days
14:	MILLER	1982-01-23	13255 days
15:	JACK	1982-01-23	13255 days

날짜로 형 변환하는 함수 = `as.Date` (오라클에서 `to_date`)

문제42. 이름, 입사한 날짜부터 오늘까지 총 몇달 근무했는지 출력시오.

(사용자 정의함수를 만들어서 해야하는데 나중에 하기로..)

문제43. 오늘 날짜의 달의 마지막 날짜를 출력하시오.(SQL은 `select last_day(sysdate) from dual;`)

`install.packages("lubridate")` 먼저 패키지를 설치

`library(lubridate)`

`?lubridate`

```
> ceiling_date(Sys.Date())
[1] "2018-05-09 00:00:01 UTC"
> ceiling_date(Sys.Date(), "month")
[1] "2018-06-01"
```

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

답)

`ceiling_date(Sys.Date(), "month") - 1`

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

문제44. `last_day`라는 함수를 생성하시오.

```
last_day <- function(x) {
  ceiling_date(x, "month") - 1
}
```

함수를 생성하고,

```
last_day(Sys.Date())
> last_day(Sys.Date())
```

```
last_day(Sys.Date())
> last_day(Sys.Date())
[1] "2018-05-31"
```

문제45. (마지막문제) first_day 라는 함수를 생성하시오.

```
first_day <- function(x){
  floor_date(x,"month")
}
first_day(Sys.Date())
> first_day <- function(x){
+   floor_date(x,"month")
+ }
> first_day(Sys.Date())
[1] "2018-05-01"
```

문제46. next_day 함수를 생성하시오.

오라클

```
select next_day(sysdate,'friday')
from dual;
```

	NEXT_DAY(SYSDATE,'FRIDAY')
1	2018-05-11 오전 10:09:29

R

next_day(Sys.Date(),'금요일') 라고 치면
2018-05-11 <-나오도록

답)

```
next_daty <- function(x,day){
  for (i in 1:7) {
    check_date = as.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"
```

문제47. emp 테이블의 입사일을 요일로 출력하시오.

```
format(as.Date(emp$hiredate),'%A') <- format = oracle에서 to_char라고 보면된다.
%A = 요일을 출력한다.
%Y = 연도를 출력한다.(4자리)
```

%y = 연도를 출력한다.(2자리)

%m = 달을 출력한다.

%d = 일을 출력한다.

%H = 시간을 출력한다.

%S = 초를 출력한다.

%M = 분을 출력한다.

```
> format(as.Date(emp$hiredate), '%A')
[1] "수요일" "금요일" "일요일" "목요일" "월요일" "금요일" "화요일" "일요일"
[9] "화요일" "화요일" "토요일" "목요일" "목요일" "토요일" "토요일"
> |
```

문제48. 내일이 무슨요일인지 출력하시오.

```
format(Sys.Date()+1, '%A')
> format(Sys.Date()+1, '%A')
[1] "금요일"
```

add_months

문제49. 오라클의 add_months 함수를 R에서 생성하기 위해 다음식을 수행하시오.

오늘부터 100달뒤의 날짜를 오라클에서 출력하시오.

```
select add_months(sysdate,100)
from dual;
```

ADD_MONTHS(SYSDATE,100)
1 2026-09-10 오전 11:04:07

- R에서 오늘부터 100달뒤 날짜를 출력하시오.

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

문제50. 아래와 같이 add_months 함수를 실행하면 100달뒤에 날짜가 출력되게 함수를 생성하시오.

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

add_months <- function(x,y){
  x + months(y)
}
> add_months(Sys.Date(),100)
[1] "2026-09-10"
```

months_between

문제51. 오라클의 months_between 함수를 R에서 생성하기 위해서 아래의 식을 구현하시오.

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

```
select ename, hiredate, round(months_between(sysdate,hiredate))
```

from emp;

1	KING	1981-11-17 오전 12:00:00	438
2	BLAKE	1981-05-01 오전 12:00:00	444
3	CLARK	1981-05-09 오전 12:00:00	444
4	JONES	1981-04-01 오전 12:00:00	445
5	MARTIN	1981-09-10 오전 12:00:00	440
6	ALLEN	1981-02-11 오전 12:00:00	447
7	TURNER	1981-08-21 오전 12:00:00	441
8	JAMES	1981-12-11 오전 12:00:00	437
9	WARD	1981-02-23 오전 12:00:00	447
10	FORD	1981-12-11 오전 12:00:00	437
11	SMITH	1980-12-09 오전 12:00:00	449
12	SCOTT	1982-12-22 오전 12:00:00	425
13	ADAMS	1983-01-15 오전 12:00:00	424
14	MILLER	1982-01-11 오전 12:00:00	436

- R에서 아래와 같이 수행하면 달수가 출력되는 함수를 생성하시오.

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

```
year(as.Date(emp$hiredate))
```

<- emp테이블은 팩터라서 Date로 바꿔줘야함

```
months(as.Date(emp$hiredate))
```

- 아래 두 날짜 사이의 개월 수를 출력하시오.

1981-11-17 ~ 2018-05-10

1. 1981년도만 빼고 2018년도만 빼보시오

```
year(as.Date('1981-11-17'))
```

```
year(as.Date('2018-05-10'))
```

```
> year(as.Date('1981-11-17'))
```

```
[1] 1981
```

```
> year(as.Date('2018-05-10'))
```

```
[1] 2018
```

```
(year(as.Date('2018-05-10'))- year(as.Date('1981-11-17')))*12
```

```
> (year(as.Date('2018-05-10'))- year(as.Date('1981-11-17')))*12
```

```
[1] 444
```

2. 두개의 날짜에서 개월 수만 빼보시오.

```
(month(as.Date('2018-05-10'))- month(as.Date('1981-11-17')))
```

```
> (month(as.Date('2018-05-10'))- month(as.Date('1981-11-17')))
```

```
[1] -6
```

문자가 기니까 year를 date1에 담고 month를 date2에 담아준다.

```
date1 <-(year(as.Date('2018-05-10'))- year(as.Date('1981-11-17')))*12
```

```
date2 <-(month(as.Date('2018-05-10'))- month(as.Date('1981-11-17')))
```

3. 두개를 더한다.

```
date1 + date2
```

```
> date1 + date2
[1] 438
```

이제 문제를 다시 풀어보세요.

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

```
months_between(Sys.Date(), emp$hiredate)
> months_between(Sys.Date(), emp$hiredate)
[1] 449 447 447 445 440 444 443 373 438 440 372 437 437 436 436
```

점심시간 문제

오라클 db의 emp테이블의 data를 csv로 내려서 R 에 로드해서 오라클과 R과 emp테이블의 내용을 일치시키시오.

gate에서 csv 로 내보내고 c : data에 다시 저장한후,

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

```
> 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
```

4. 일반 함수

오라클	vs	R
nvl		is.na
decode		ifelse
case		ifelse

ex) 이름, 월급, 등급을 출력하는데 월급이 1500이상이면 등급을 A로 출력하고 아니면 B로 출력하시오.

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

	이름	월급	등급
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

만약 sal >= 이상이면 A를 출력
아니면 B로 출력
오라클에
decode같이 해석

문제52. 이름, 월급, 등급을 출력하는데, 월급이 3000이상이면 A를 출력하고 월급이 1500 이상이고 3000보다 작으면 B를 출력하고 나머지 직원들은 C로 출력하시오. (ifelse안에 ifelse를 쓸 수 있다.)

```
data.table( 이름 = emp$ename, 월급 = emp$sal,
  등급 = ifelse(emp$sal>=3000, 'A',ifelse(emp$sal>=1500, 'B','C')))
```

	이름	월급	등급
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

문제53. 이름, 입사일, 보너스를 출력하는데 1980년도에 입사했으면 보너스를 A로 출력하고 1981년도에 입사했으면 보너스를 B로 출력하고 1982년도에 입사했으면 보너스를 C로 출력하고 나머지 년도는 D로 출력하시오.

```
data.table( 이름 = emp$ename, 월급 = emp$sal,
  보너스 = ifelse(year(emp$hiredate) == '1980', 'A',
    ifelse(year(emp$hiredate) == '1981', 'B',
      ifelse(year(emp$hiredate) == '1982','C','B'))))
```

	이름	월급	보너스
1:	KING	5000	B
2:	BLAKE	2850	B
3:	CLARK	2450	B
4:	JONES	2975	B
5:	MARTIN	1250	B
6:	ALLEN	1600	B
7:	TURNER	1500	B
8:	JAMES	950	B
9:	WARD	1250	B
10:	FORD	3000	B
11:	SMITH	800	A
12:	SCOTT	3000	C
13:	ADAMS	1100	B
14:	MILLER	1300	C

문제54. 이름과 커미션을 출력하는데 is.na 함수를 이용해서 커미션이 NA인 직원들을 출력하시오.

```
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

문제55. 이름과 커미션을 출력하는데 커미션이 NA인 직원들은 no comm 이란 글씨로 출력되게하시오.

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

1:	KING	NA
2:	BLAKE	NA
3:	CLARK	NA
4:	JONES	NA
5:	MARTIN	1400
6:	ALLEN	300
7:	TURNER	0
8:	JAMES	NA
9:	WARD	500
10:	FORD	NA
11:	SMITH	NA
12:	SCOTT	NA
13:	ADAMS	NA
14:	MILLER	NA

5. 변환 함수

: 데이터의 유형을 변경하는 함수

오라클

vs

R

to_char

as.character

to_number	as.integer
to_date	as.Date
	as.factor <-- R에만있음 (factor로 변환)
	format <-- R에만 있음

• format 함수 설명

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

%A = 요일을 출력한다.

%Y = 연도를 출력한다.(4자리)

%y = 연도를 출력한다.(2자리)

%m = 달을 출력한다.

%d = 일을 출력한다.

%H = 시간을 출력한다.

%S = 초를 출력한다.

%M = 분을 출력한다.

ex) 내가 무슨 요일에 태어났는지 출력하시오.

6. 그룹 함수

오라클	vs	R
max		max
min		min
sum		sum
avg		mean
count		length(세로), table(가로)

ex) `max(emp$sal)`

```
> max(emp$sal)
[1] 5000
```

문제56. 직업이 SALESMAN인 직원들중에 최대월급을 출력하시오.

`max(emp[emp$job=='SALESMAN',"sal"]) <-- 월급 하나만 가져오는거라 컴바인 안써도 된다.`

문제57. 30번 부서번호인 직원들 중에서 최소월급을 출력하시오.

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

문제58. 직업, 직업별 최대월급을 출력하시오.

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

```
   job  sal
1 ANALYST 3000
2  CLERK 1300
3  MANAGER 2975
4 PRESIDENT 5000
```

기준컬럼

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

부서번호를 기준컬럼으로 월급의 최소값을 emp로 찾아서 출력해라

	job	sal
1	ANALYST	3000
2	CLERK	1300
3	MANAGER	2975
4	PRESIDENT	5000
5	SALESMAN	1600

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

부서번호를 기준컬럼으로 월급의 최소값을
emp로 찾아서 출력해라

문제59. 부서번호, 부서번호별 최소월급을 출력하시오.

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

	deptno	sal
1	10	1300
2	20	800
3	30	950

문제60. 부서번호, 직업, 부서번호별 직업별 토탈 월급을 출력하시오.

```
select deptno, job, sum(sal)
from emp
group by deptno, job; 을 R로
```

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

	deptno	job	sal
1	20	ANALYST	6000
2	10	CLERK	1300
3	20	CLERK	1900
4	30	CLERK	950
5	10	MANAGER	2450
6	20	MANAGER	2975
7	30	MANAGER	2850
8	10	PRESIDENT	5000
9	30	SALESMAN	5600

문제61. 부서번호, 부서번호별 최대월급을 출력하는데, 부서번호별 최대월급이 높은 것부터 출력하시오.

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

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

문제62. 직업, 직업별 인원수를 출력하시오.

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

<-- 사원번호를 가지고 length를 구할 것 이다.

문제63. 직업, 직업별 평균월급을 출력하시오.

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

	job	sal
1	ANALYST	3000.000
2	CLERK	1037.500
3	MANAGER	2758.333

	job	sal
1	ANALYST	3000.000
2	CLERK	1037.500
3	MANAGER	2758.333
4	PRESIDENT	5000.000
5	SALESMAN	1400.000

문제64. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하시오.

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

문제65. 입사한 년도(4자리), 입사한 년도별 토탈월급을 출력하시오.(가로로 출력하시오)

```
tapply(emp$sal, year(emp$hiredate),sum)
> tapply(emp$sal, year(emp$hiredate),sum)
1980 1981 1982 1983
 800 22825 4300 1100
==> 설명 : tapply(그룹함수에 쓸 컬럼, 기준 컬럼명, 그룹함수)
```

문제66. 부서번호, 부서번호별 토탈월급을 가로로 출력하시오.

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

attach(emp)

tapply(sal, deptno, sum)

할 수 도있는게 가능한데 안쓰는게 좋다.

해제방법

detach(emp)

문제67. 직업과 직업별 인원수를 출력하는데 세로로도 출력하고 가로로도 출력하시오.

세로

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

job	empno
ANALYST	2
CLERK	4
MANAGER	3
PRESIDENT	1
SALESMAN	4

가로

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

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

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

문제68. 부서번호별 직업별 토탈월급을 출력하시오.

```
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
```

문제69. 문제68번 결과에서 NA를 숫자 0으로 출력하시오.

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

x에다 결과를 담고
is.na 함수를 써준다
출력
만약 na가 아닌것을 0으로 할려고 하면 x[!is.na(x)]

문제70. 직업, 입사한 년도(4자리), 직업별 입사한 년도별 토탈월급을 출력하시오.

```
SALESMAN ANALYST CLERK PRESIDENT
1980
1981
1982
1983
```

```
x<-tapply(emp$sal,list(year(emp$hiredate),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

문제71.(마지막문제) 나이, 통신사, 나이별 통신사별 인원수를 아래와 같은 형태로 출력하시오.

(나이 세로, 통신사 가로)

```
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

문제72. 직업(세로), 입사한 년도 4자리(가로), 직업별 입사한 년도별 토달월급을 출력하시오.

```
library(data.table)
library(lubridate)

x <- tapply(emp$sal, list(emp$job, year(emp$hiredate)), 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

문제73. 아래의 결과에서 컬럼명과 로우명을 각각 출력하시오.

	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)
> colnames(x)
[1] "1980" "1981" "1982" "1983"
```

```
rownames(x)
> rownames(x)
[1] "ANALYST" "CLERK" "MANAGER" "PRESIDENT" "SALESMAN"
```

문제74. crime_loc.csv를 R로 로드를 하고 범죄유형이 살인인 범죄의 장소와 건수를 출력하시오.

```
crime_loc[crime_loc$범죄 == '살인', c("장소", "건수")]
      장소  건수
82   아파트  242
83     집   312
84 고속도로    1
85     노상  280
86     상점   23
87 시장노점    5
88 숙박업소  43
```

	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
92	공사장	4
93	창고	4
94	역대합실	8

문제75. 장소(세로), 범죄(가로), 장소별 범죄별 건수를 출력하시오.

절도 살인 강도 공갈

아파트

집

고속도로

노상

상점

.

.

.

답)

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

문제76. 서울시 물가 데이터(price.csv)를 R로 로드하고 tapply 함수를 이용해서 전통시장과 대형마트간의 물품별 가격 평균을 출력하시오.

	대형마트	전통시장
고등어	720	500
무	100	120
.	.	.
.	.	.

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

```
colnames(price) <- tolower(colnames(price)) <-- price의 컬럼명을 소문자로 바꾼다.
```

```
head(price,5)
```

답)

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

```
x[is.na(x)]<-0
```

```
head(x,5)
```

고등어 (냉동, 수입산)

대형마트	전통시장
고등어	3000
고등어 (30cm, 국산)	3140
고등어 (30cm, 수입산)	0
고등어 (냉동, 국산)	3415
고등어 (냉동, 수입산)	2500

R의 조인

2018년 5월 11일 금요일 오전 9:46

서로 다른 dataframe의 변수들을 하나의 결과로 출력할때 사용하는 문법으로 R에서는 **merge**함수를 사용해서 구현한다.

문제77. dept.csv를 내려받고 dept라는 변수에 로드하세요.

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

문제78. emp data frame과 dept data frame을 merge해서 이름과 월급과 부서위치를 출력하시오.

```
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
merge(테이블명1, 테이블명2, by = '공통된 컬럼명') [행,열]
```

문제79. 부서위치가 DALLAS인 직원들의 이름과 월급과 부서위치를 출력하시오.

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

```
x<-merge(emp, dept, by='deptno')
x[x$loc == 'DALLAS', c("ename","sal","loc")]
변수에 담아서 하는방법도있다.
```



```

      ename sal      loc
4 JONES 2975 DALLAS
5 SMITH  800 DALLAS
6 SCOTT 3000 DALLAS
7 ADAMS 1100 DALLAS
8  FORD 3000 DALLAS
> |

```

문제80. 월급이 1200이상이고 직업이 SALESMAN인 직원들의 이름과 월급과 직업과 부서위치를 출력하시오.

```

x<-merge(emp,dept,by='deptno')
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

```

문제81. 커미션이 NA인 직원들의 이름과 부서위치와 커미션을 출력하시오.

```

merge(emp,dept)[is.na(merge(emp,dept)$comm),c("ename","loc","comm")]
      ename      loc comm
1    KING NEW YORK  NA
2  MILLER NEW YORK  NA
3   CLARK NEW YORK  NA
4   JONES  DALLAS  NA
5   SMITH  DALLAS  NA
6   SCOTT  DALLAS  NA
7   ADAMS  DALLAS  NA
8    FORD  DALLAS  NA
9   JAMES CHICAGO  NA
11  BLAKE CHICAGO  NA

```

문제82. 이름과 부서위치를 출력하는데 오라클의 **outer join**과 같은 결과를 출력하시오.

```

SQL > select e.ename, d.loc
      from emp e, dept d
      where e.deptno(+) = d.deptno;
-----

```

```

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

```

```

      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

```

	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

문제83. 이름과 부서위치를 출력하는데 오라클의 full outer join과 같은 결과를 출력하시오.

```
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

문제84. 이름과 자기의 직속상사의 이름(관리자)를 출력하시오.

```
SQL>select e.ename, m.ename
      from emp e, emp m
      where e.mgr = m.empno;
```

```
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

문제85. 문제84번을 다시 수행하는데 자신의 월급이 자신의 관리자인 사원의 월급보다 더 많은 월급을 받는
사원들만 출력하시오.

```
x85<-merge(emp, emp, by.x="mgr", by.y="empno")
x85[x85$sal.x > x85$sal.y, c("ename.x")]
[1] FORD SCOTT
```

문제86. 부서위치, 부서위치별 토탈월급을 출력하시오.

```
SQL>select d.loc, sum(e.sal)
      from emp e, dept d
      where e.deptno = d.deptno
      group by d.loc;
```

```
x86<-merge(emp,dept, by="deptno")[,c("loc","sal")]
```

먼저 x86에 데이터를 넣어주고

```
aggregate(sal~loc, x86, sum)
      loc    sal
1 CHICAGO  9400
2 DALLAS  10875
3 NEW YORK  8750
```

문제87. 부서위치, 부서위치별 토탈월급을 출력하시오.(가로)

```
tapply(x86$sal, x86$loc,sum)
> tapply(x86$sal, x86$loc,sum)
BOSTON CHICAGO DALLAS NEW YORK
NA      9400   10875    8750
```

문제88. 부서위치(세로), 입사한 년도4자리(가로), 부서위치별 입사한 년도별인원수를 출력하시오.

```
x88<-merge(emp,dept,by="deptno")

y88<-tapply(x88$sal ,list(x88$loc, year(x88$hiredate)), length)
y88[is.na(y88)]<-0
y88
```

	1980	1981	1982	1983
BOSTON	0	0	0	0
CHICAGO	0	6	0	0
DALLAS	1	2	1	1
NEW YORK	0	2	1	0

• 집합 연산자

merge는 양옆으로 결과를 합치는 것이지만, 집합연산자는 위아래로 결과를 합치는 문법.

오라클	vs	R
union all		rbind
union		rbind+unique
intersect(교집합)		intersect(dplyr 패키지)
minus		setdiff(dplyr 패키지)

rbind(오른쪽으로 붙임)와 연관된 다른 문법 : cbind(양옆으로 붙임)

rbind

문제89. 아래의 SQL을 R로 구현하시오.

```
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
1	KING	5000	10
2	CLARK	2450	10
3	JONES	2975	20
4	FORD	3000	20
5	SMITH	800	20
6	SCOTT	3000	20
7	ADAMS	1100	20
8	MILLER	1300	10
9	BLAKE	2850	30
10	MARTIN	1250	30
11	ALLEN	1600	30
12	TURNER	1500	30
13	JAMES	950	30
14	WARD	1250	30

```
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

문제90. 아래의 SQL의 결과를 R로 구현하시오.

```
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),
      c(' ', sum(emp$sal)))
```

	deptno	sal
1	10	8750
2	20	10875
3	30	9400
4		29025

문제91. 아래의 SQL의 결과를 R로 구현하시오.

```
select nvl(job,'토탈값') job, count(*)
  from emp
  group by rollup(job);
```

	JOB	COUNT(*)
1	ANALYST	2
2	CLERK	4
3	MANAGER	3
4	PRESIDENT	1
5	SALESMAN	4
6	토탈값	14

```
rbind(aggregate(empno~as.character(job), emp, length),  
c("토탈값", length(emp$empno)))
```

"토탈값" 은 그냥 factor이다.

```
as.character(job) empno
1 ANALYST 2
2 CLERK 4
3 MANAGER 3
4 PRESIDENT 1
5 SALESMAN 4
6 토탈값 14
```

factor와 character의 차이점

factor는 순서(a,b,c)가있다

rbind+unique

문제92. 아래의 SQL의 결과를 R로 구현하시오.

```
select ename, sal, deptno
  from emp
 where deptno in(10,20)
union
select ename, sal, deptno
  from emp
 where deptno = 10;
```

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
```

문제93. 아래의 SQL의 결과를 R로 구현하시오.

```
select ename, sal, deptno
  from emp
 where deptno in(10,20)
intersect
select ename, sal, deptno
  from emp
 where deptno = 10;
```

```
x93 <- intersect(emp[emp$deptno %in% c(10,20),c("ename")],
emp[emp$deptno == 10,c("ename")])
```

```
emp[emp$ename %in% x93, c("ename", "sal", "deptno")]
```

	ename	sal	deptno
1	KING	5000	10
3	CLARK	2450	10
14	MILLER	1300	10

그냥 **intersect**를 사용하려면 컬럼의 개수가
같아야 쓸 수 있다.

하지만!!

library(dplyr) 을 써주면 그냥 사용할 수 있다.

```
library(dplyr)
```

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

문제94. 아래의 SQL의 결과를 R로 구현하시오.

```
select ename, sal, deptno
  from emp
 where deptno in(10,20)
minus
select ename, sal, deptno
  from emp
 where deptno = 10;
```

```
library(dplyr)
```

```
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

R에서의 서브쿼리

2018년 5월 11일 금요일 오전 9:46

- 오라클의 서브쿼리

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

오라클의 서브쿼리의 서브쿼리를 R로 구현하는 방법은 변수만 잘 활용하면 된다.

문제95. JONES의 월급보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오.

SQL>

```
select ename,sal
from emp
where sal > (select sal from emp where ename='JONES');
```

x95 <- emp[emp\$ename == 'JONES', "sal"] 변수에다가 넣어주고

emp[emp\$sal > x95, c("ename","sal")] 그냥 출력하면된다.

```
  ename  sal
1   KING 5000
10  FORD 3000
12 SCOTT 3000
```

문제96. 최대월급을 받는 직원의 이름과 월급을 출력하시오.

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

```
  ename  sal
1   KING 5000
```

변수에 넣어서 하는 방법

```
x95 <- max(emp$sal)
```

```
emp[emp$sal ==x95, c("ename","sal")]
```

문제97. KING에게 보고하는 직원들의 이름과 월급을 출력하시오.

SQL >

```
select ename,sal
from emp
where mgr = (select empno from emp where ename = 'KING');
```

x97<-emp[emp\$ename == 'KING', "empno"]

x97에 킹의 직원번호를 담고
출력한다.

(emp[emp\$mgr == x97, c("ename","sal")])

그런데 그냥 출력하면


```

      ename  sal
NA   <NA>   NA
2    BLAKE 2850
3    CLARK 2450
4    JONES 2975

```

NA가 나오는데

```
x97<-emp[emp$ename == 'KING', "empno"]
```

`na.omit(emp[emp$mgr == x97, c("ename","sal")])` `na.omit`을 써주면 없앨 수 있다.

```

      ename  sal
2    BLAKE 2850
3    CLARK 2450
4    JONES 2975

```

문제98. 관리자인 직원들의 이름을 출력하시오.

```

SQL>select ename
      from emp
      where empno in (select mgr from emp);

```

```

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

```

문제99. 관리자가 아닌 직원들의 이름을 출력하시오.

```

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

```

문제100. 아래의 문제를 R로 구현하시오.

문제

흰공 2개, 빨간공 4개가 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

방법1. 조합에 관련한 패키지를 찾아서 아래식을 푼다.

방법2. 함수를 만들어서 구현한다.

ex) combination(2,2)+combination(6,2)=16

```
combi <- function(x,y){
```

```
  a = 1
```

```
  b = 1
```

```
  for ( i in (x-y+1):x){
```

```
    a = a*i
```

```

}
for (i in 1:y){
  b = b*i
}
return(a/b) }
combi(2,2) + combi(6,2)
> combi(2,2) + combi(6,2)
[1] 16

```

• 순위 출력을 R로 구현하는 방법

오라클 vs R

dense_rank rank

ex) 이름, 월급, 월급에 대한 순위를 출력하시오.

```

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

```

min : 오라클의 rank와 같다

first : 오라클의 rank와 같은데 인덱스가 먼저 나오는 데이터를 높은순위로 부여한다.

문제101. 이름, 월급, 월급에 대한 순위를 출력하는데 순위가 1등부터 정렬하시오.

컬럼 이름 변경하는 방법

```
colnames(x101) <- c("ename", "sal", "rnk")
```

x101

```

> x101
      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

```

```
> x101
      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
```

```
x101<-data.table(emp$ename, emp$sal, rank(-emp$sal,ties.method = "min"))
```

```
x101[order(x101$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
```

문제102. 원섭이의 2017년 아파트 매매 데이터를 partment라는 변수에 담고, 아래의 결과를 출력하시오.

	주소	아파트 이름	2010년 아파트 가격(만원)	2017년 아파트 가격(만원)	층수	가격 증감률	DENSE_RANK()OVER(ORDERBYGAINDESC)
1	경기도 남양주시 호평동	호평마을우미아노스빌	11430	28500	3	2.4934	1
2	경기도 남양주시 호평동	호평마을우미아노스빌	11430	28300	3	2.4759	2
3	경기도 남양주시 다산동	롯데낙천대	12000	29700	5	2.475	3
4	경기도 남양주시 다산동	롯데낙천대	12000	29550	5	2.4625	4
5	경기도 남양주시 다산동	롯데낙천대	12000	29300	5	2.4417	5
6	경기도 남양주시 오남읍	한신(2차)	10000	20800	1	2.08	6
7	경기도 남양주시 오남읍	동부	11500	23900	16	2.0783	7
8	경기도 남양주시 호평동	호평마을우미아노스빌	11430	23700	3	2.0735	8
9	경기도 남양주시 오남읍	한신(2차)	11450	23700	7	2.0699	9
10	경기도 남양주시 오남읍	두산2	10900	22300	1	2.0459	10
11	경기도 남양주시 평내동	평내마을주공	11171	22700	8	2.032	11
12	경기도 남양주시 평내동	평내마을주공	11141	22500	5	2.0196	12
13	경기도 남양주시 평내동	평내마을주공	11171	22500	8	2.0141	13
14	경기도 남양주시 평내동	평내마을주공	11141	22200	5	1.9926	14
15	경기도 남양주시 오남읍	두산2	11000	21900	2	1.9909	15
16	경기도 남양주시 와부읍	덕소주공3	15000	29850	4	1.99	16
17	경기도 남양주시 오남읍	오남1형구	11950	23700	1	1.9833	17
18	경기도 남양주시 와부읍	진도	19950	39500	1	1.9799	18
19	경기도 남양주시 와부읍	한강우성	19000	37500	8	1.9737	19
20	경기도 남양주시 평내동	평내마을주공	11141	21900	5	1.9657	20

```
x102 <- data.table(paste(apartment$시군구, apartment$번지),이름 = apartment$단지명, 가격=apartment$금액,rank(-apartment$금액, ties.method = "min"))
```

```
head(x102[order(apartment$금액, decreasing = T)],20 )
```

문제103. crime_loc.csv를 R로 로드하고 병원에서 많이 발생하는 범죄유형, 건수, 순위를 출력하시오.

```
x103 <- crime_loc[crime_loc$장소 == '병원', ]
x1032 <- data.table(x103$범죄, x103$건수, rank(-x103$건수, ties.method = "min"))
colnames(x1032) <- c('crime','cnt','rnk')
x1032[order(x1032$rnk,decreasing = F), ]
```

	crime	cnt	rnk
1:	절도	16053	1
2:	폭행	14142	2
3:	상해	9568	3
4:	폭력행위 등 처벌에 관한 법률위반	7396	4
5:	손괴	3539	5
6:	강간	1362	6
7:	도박과 복표	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

```
library(dplyr)
```

```
x1032 <- data.table(x103$범죄, x103$건수, dense_rank(-x103$건수))
colnames(x1032) <- c('crime','cnt','rnk')
x1032[order(x1032$rnk,decreasing = F), ]
```

dense_rank로도 할 수 있다.

문제103-2. crime_loc.csv를 R로 로드하고 학교에서 많이 발생하는 범죄유형, 건수, 순위를 출력하시오.

```
x104 <- crime_loc[crime_loc$장소 == '학교', ]
x1042 <- data.table(x103$범죄, x103$건수, rank(-x103$건수, ties.method = "min"))
colnames(x1042) <- c('crime','cnt','rnk')
x1042[order(x1042$rnk,decreasing = F), ]
```

		type	cnt	rnk
1:		절도	3203	1
2:		상해	844	2
3:	폭력행위 등 처벌에 관한 법률 위반	755	3	
4:		폭행	590	4
5:		손괴	239	5
6:		공갈	190	6
7:		강간	188	7
8:		주거침입	56	8
9:		강도	35	9
10:		과실치사상	31	10
11:	업무상과실치사상	28	11	
12:		실화	27	12
13:	교통사고처리	24	13	

문제104. 카페에서 암 발생 데이터를 내려받고 R로 로드한 후에 여자들이 많이 걸리는 암과 건수와 순위 출력.

```
x105 <- cancer[cancer$성별 == '여자',]
x1052 <- data.table(x105$암종, x105$환자수, dense_rank(-x105$환자수))
colnames(x1052) <- c('cancer','cnt','rnk')
x1052[order(x1052$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

문제105. 2009년도에 서울시에서 교통사고가 일어난 장소와 건수와 순위를 출력하시오.

```
car_accident <- read.csv("C:\\WWdata\\WWcar_accident.csv", header = T)
x106 <- car_accident[car_accident$기준년도 == 2009 &
                     car_accident$지자체 == '서울', ]
x1062 <- data.table(x106$지점, x106$발생건수.건., dense_rank(-x106$발생건수.건.))
colnames(x1062) <- c('loc','cnt','rnk')
x1062[order(x1062$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:		엘에이모텔	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

car_accident 테이블의 데이터가 많아서 **fread**로 읽어준다.

```
car <- fread("C:\\data\\car_accident.csv", header = T)
```

```
nrow(car)
```

```
> nrow(car)
```

```
[1] 85843
```

R에서 그래프 그리는 방법

2018년 5월 15일 화요일 오후 2:59

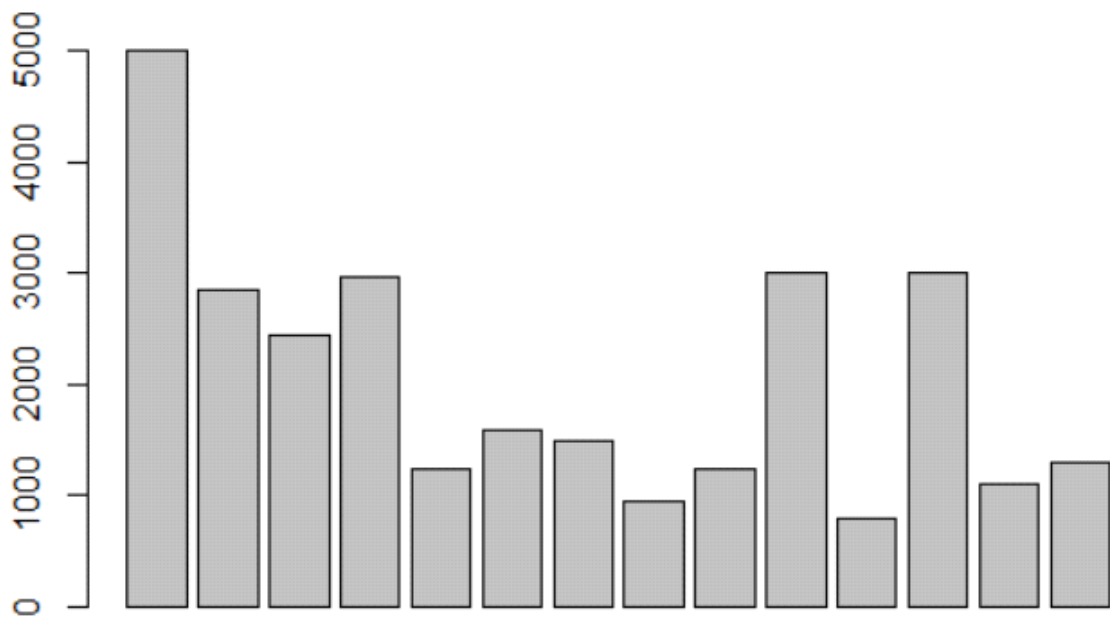
• 그래프의 종류

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

1. 막대 그래프

문제106. emp 테이블의 월급으로 기본적인 막대 그래프를 그리시오.

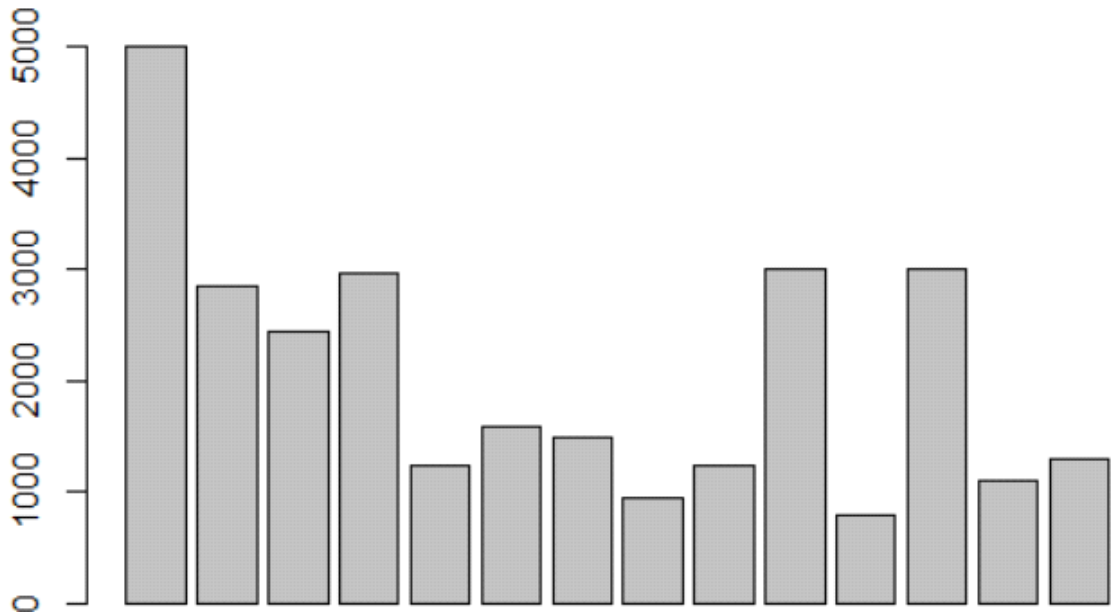
```
barplot(emp$sal)
```



문제107. 문제106번 그래프에 제목을 Salary Bar Chart 라고 이름을 붙이시오.

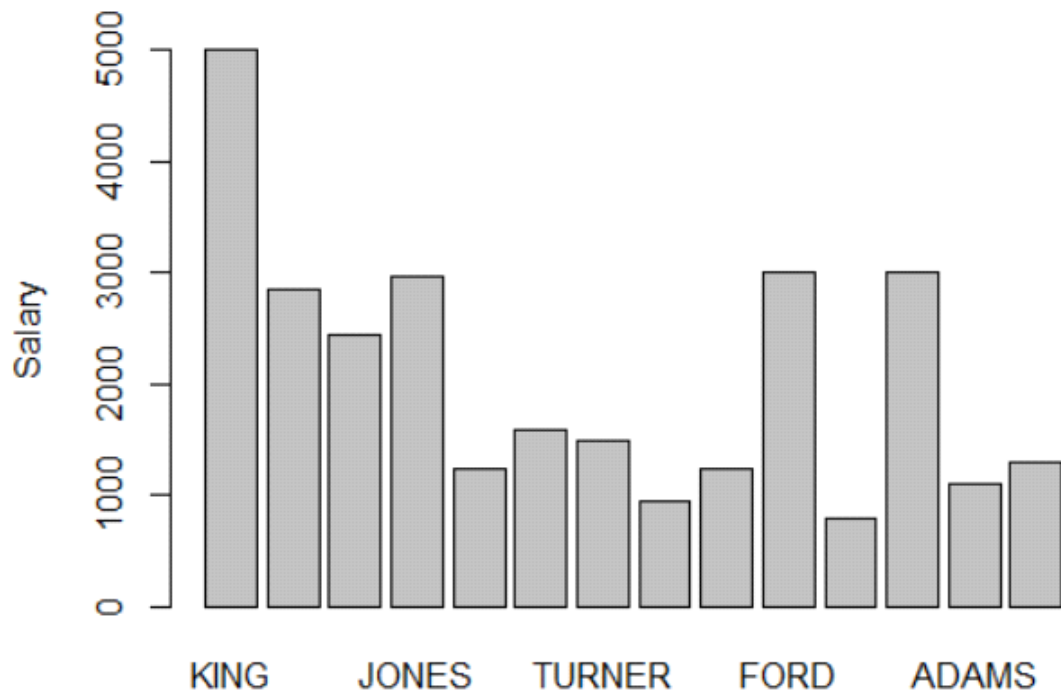
```
barplot(emp$sal, main = "Salary Bar Chart")
```

Salary Bar Chart



문제108. 문제107번 막대 그래프 x축에 사원 이름을 출력하시오.

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

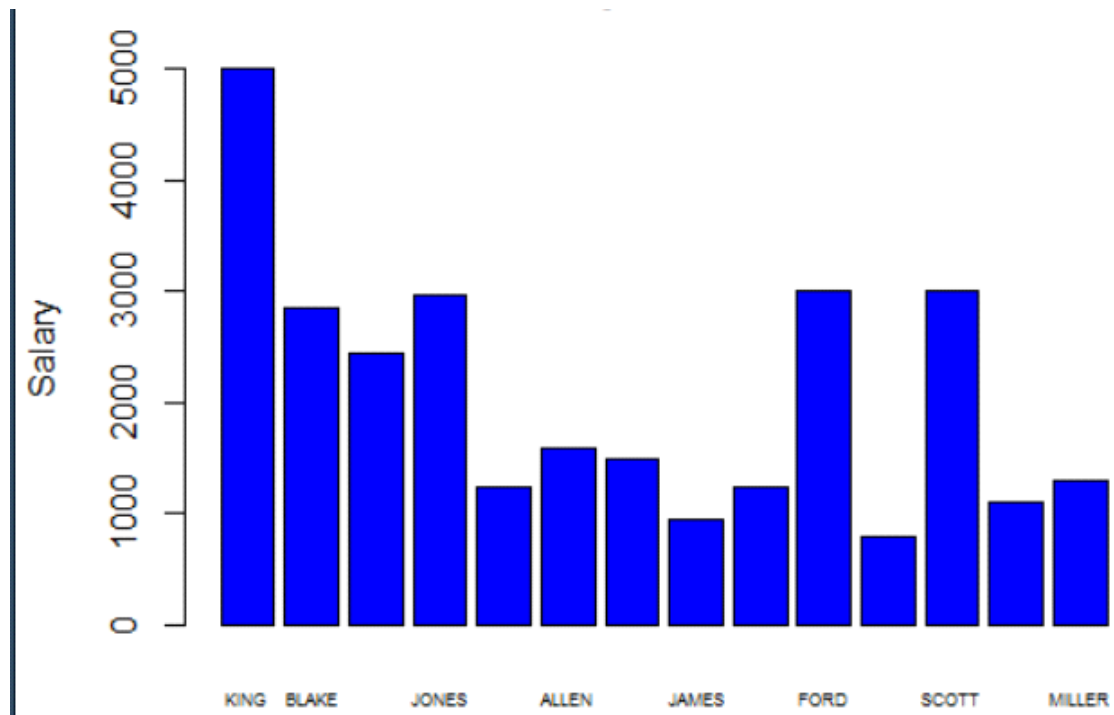


글씨 크기를 작게하면 전부나온다.

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

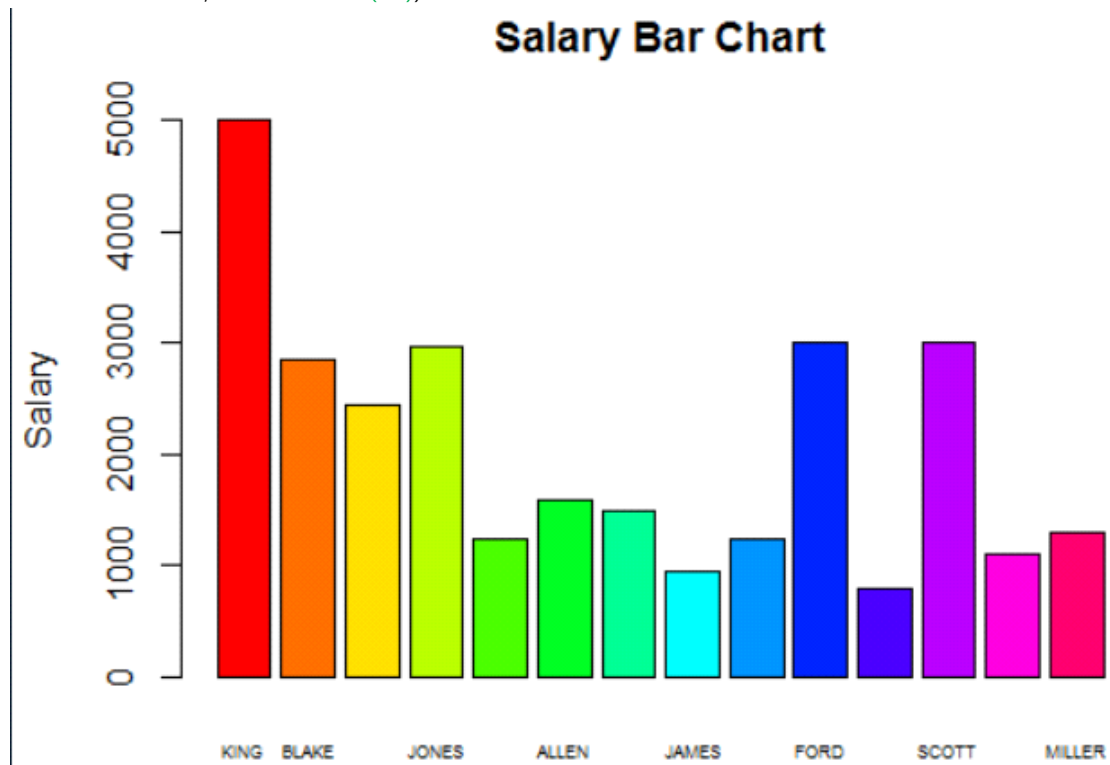
문제109. 문제108번 그래프의 색깔을 입히시오.

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

문제110. 문제109번의 색깔을 사원별로 unique하게 하시오.

```
barplot(emp$sal, main = "Salary Bar Chart", names.arg = emp$ename, ylab = "Salary",
cex.names = 0.5, col=rainbow(14))
```

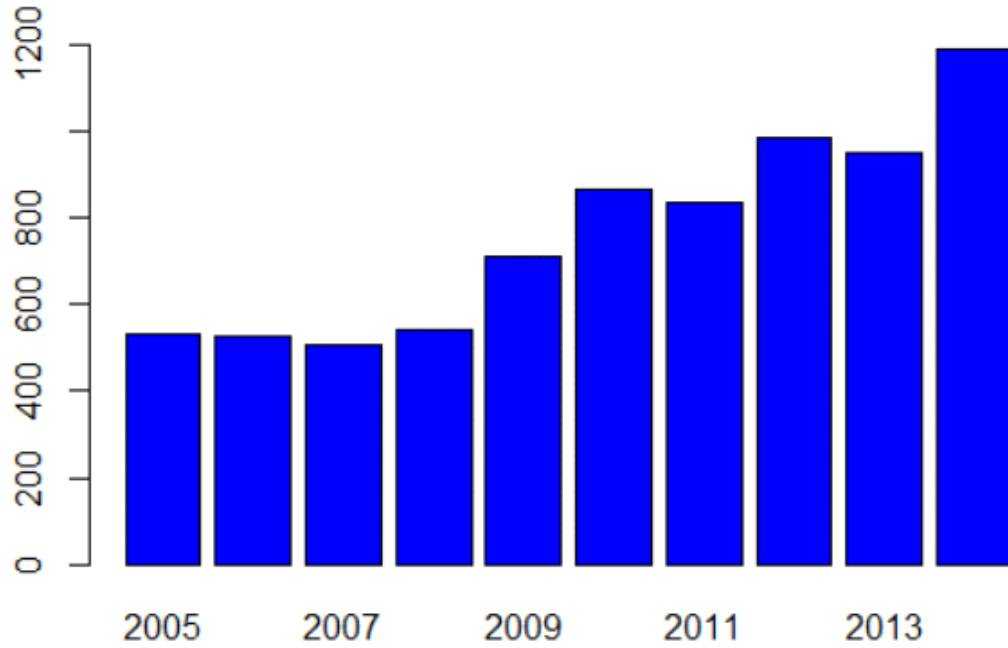


문제111. 치킨집 년도별 창업건수를 막대 그래프로 시각화 하시오.

```
barplot(create_cnt$치킨집, main = "년도별 치킨집 창업건수", names.arg = create_cnt$X,
col=('blue'),ylim=c(0,1300))
```

<- ylim : y축의 리미트

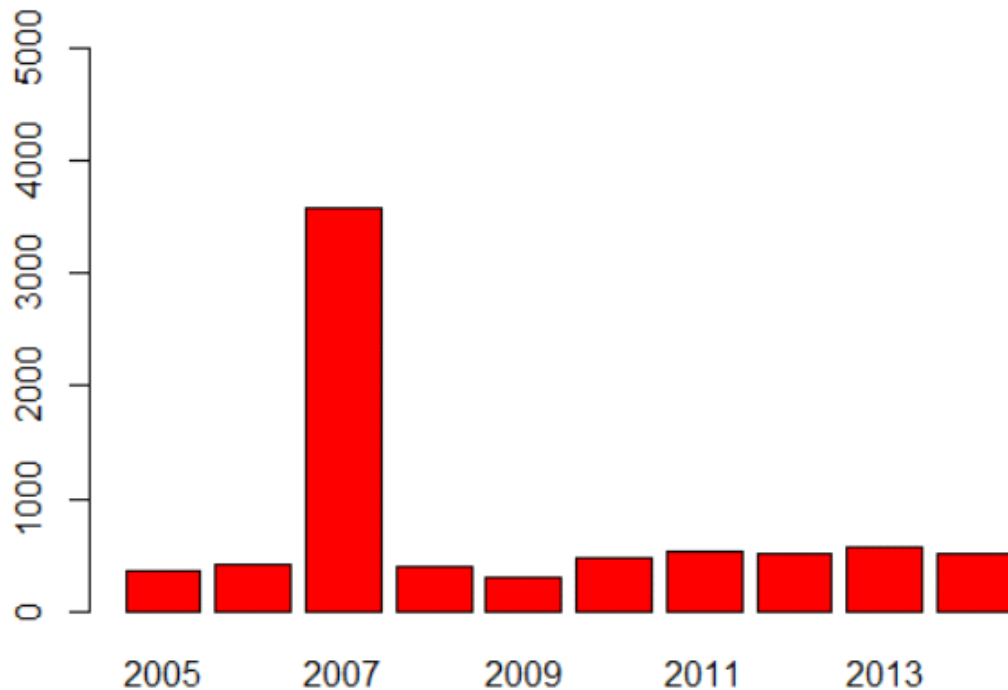
년도별 치킨집 창업건수



문제112. 치킨집 년도별 폐업건수를 막대 그래프로 시각화 하시오.

```
barplot(drop_cnt$치킨집, main = "년도별 치킨 폐업건수", names.arg = drop_cnt$X, col=('red'),ylim=c(0,5000))
```

년도별 치킨 폐업건수

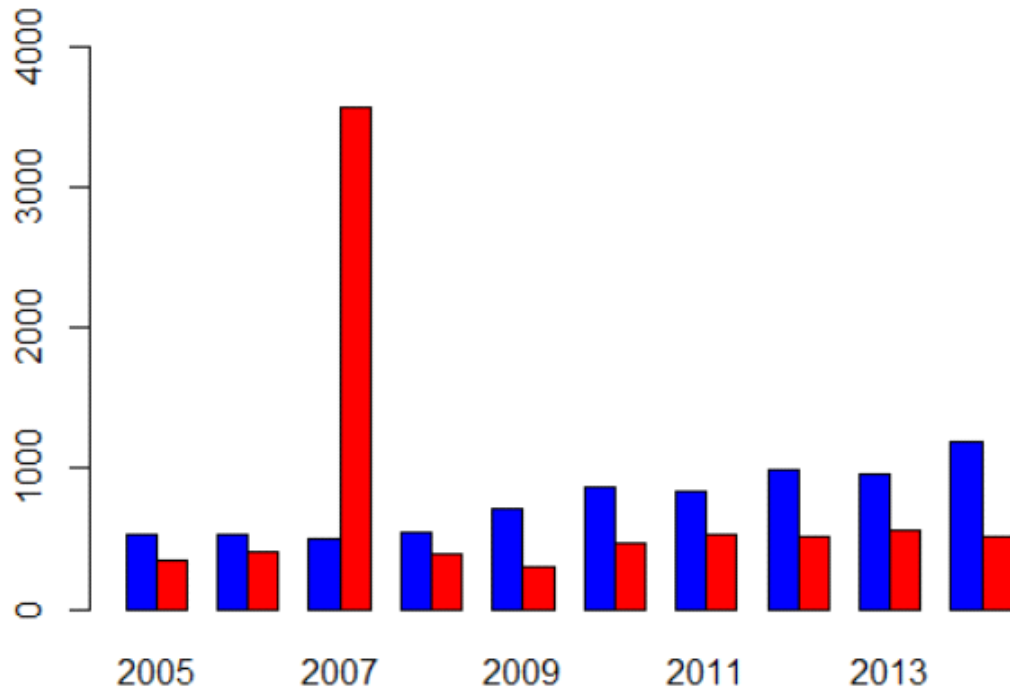


```
graphics.off() <-- 그래프 없애기
```

문제113. 치킨집 년도별 창업건수, 폐업건수를 막대 그래프로 그리는데 같이 출력되게 하시오.

```
x <- rbind(create_cnt$치킨집, drop_cnt$치킨집)
barplot( x, main ="년도별 치킨 창업,폐업건수",names.arg=create_cnt$X,
        col=c('blue','red'), ylim= c(0,4000), beside = T)
```

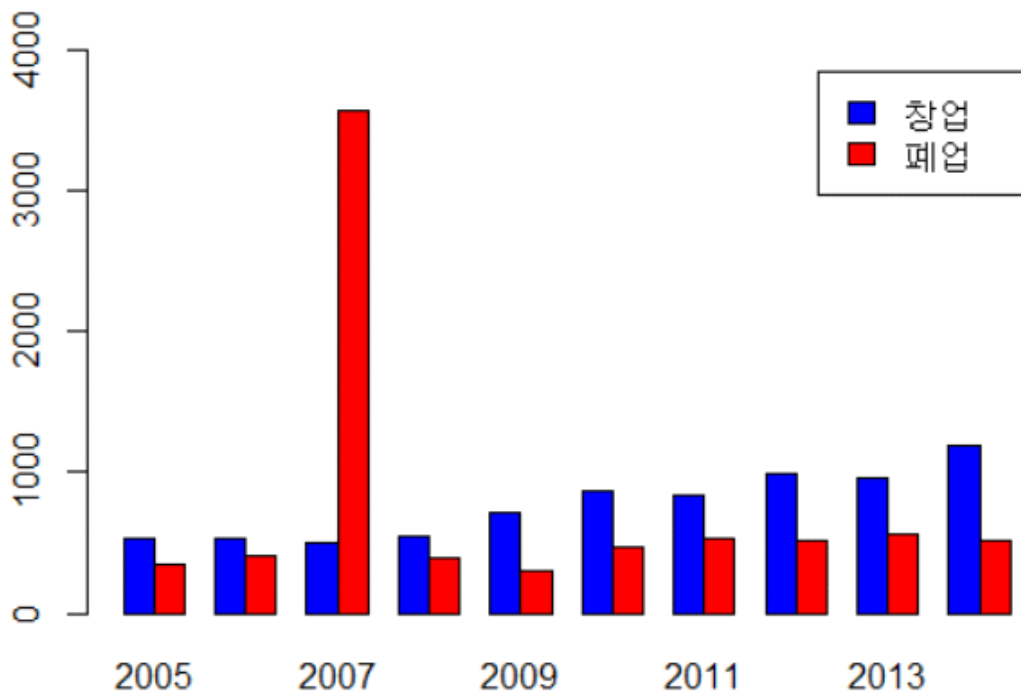
년도별 치킨 창업,폐업건수



문제114. 문제113번의 legend를 달아서 파란색이 창업이고, 빨간색이 폐업이다 라고 하시오.

```
barplot( x, main ="년도별 치킨 창업,폐업건수",names.arg=create_cnt$X,
         col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업") )
```

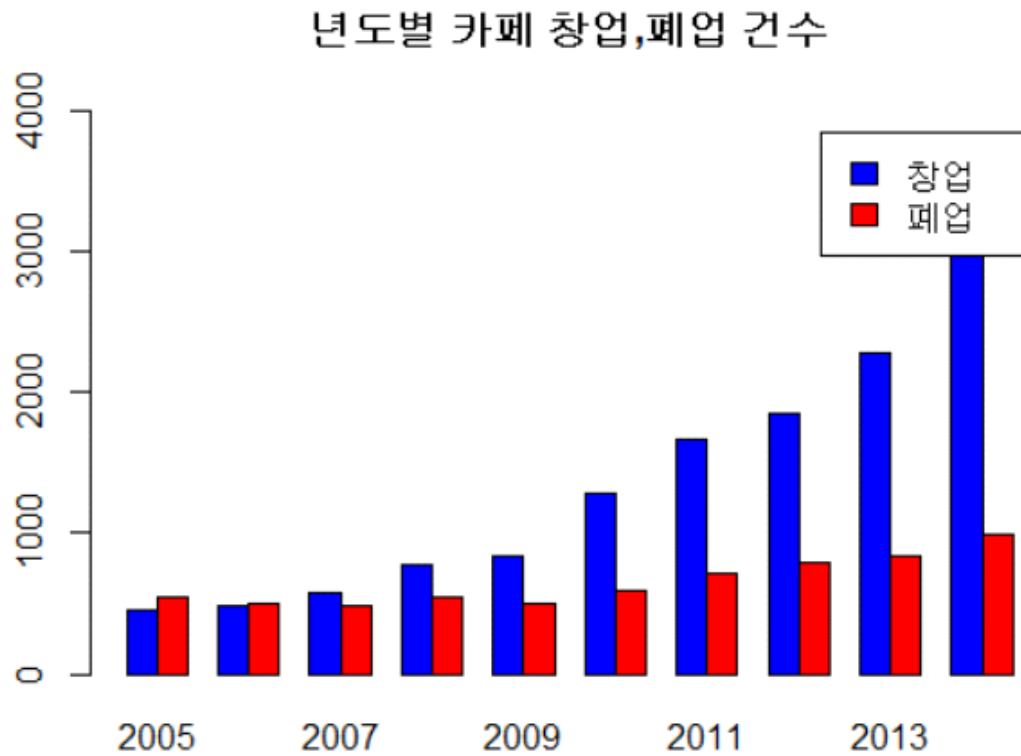
년도별 치킨 창업,폐업건수



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

문제115. 카페(커피음료)가 얼마나 창업을 하고 얼마나 폐업을 하는지 막대 그래프로 시각화하시오.

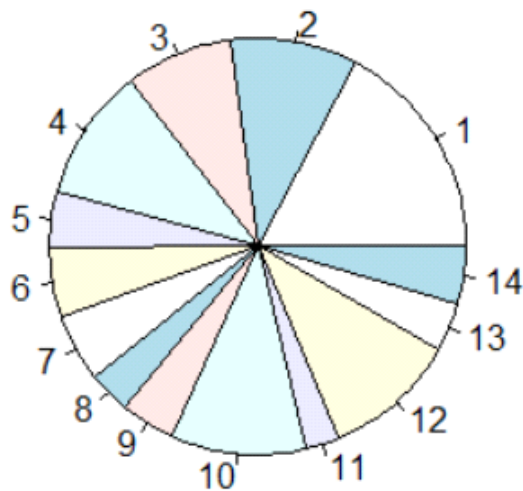
```
x <- rbind(create_cnt$커피음료, drop_cnt$커피음료)
barplot(x, main="년도별 카페 창업,폐업 건수", names.arg = create_cnt$X,
        col = c('blue','red'), ylim=c(0,4000), beside = T, legend = c("창업","폐업"))
```



2. 원형 그래프

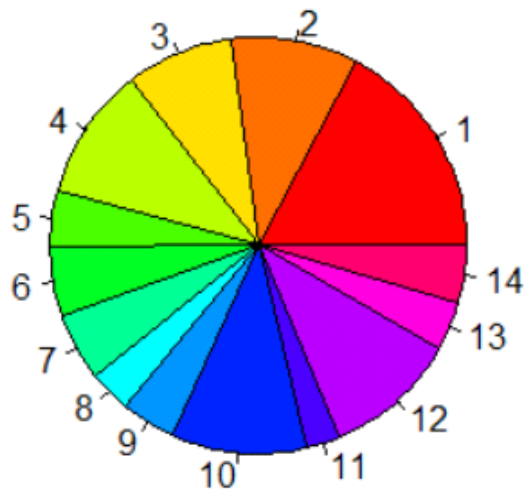
문제116. 사원 테이블의 월급으로 원형 그래프를 그리시오.

```
pie(emp$sal)
```



문제117. 문제116번을 다시 수행하는데 색깔을 좀더 화려하게 시각화 하시오.

```
pie(emp$sal, col =rainbow(14))
```



문제118. 문제117번을 다시 수행하는데 제목을 Salary Pie Chart라고 붙이시오.

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

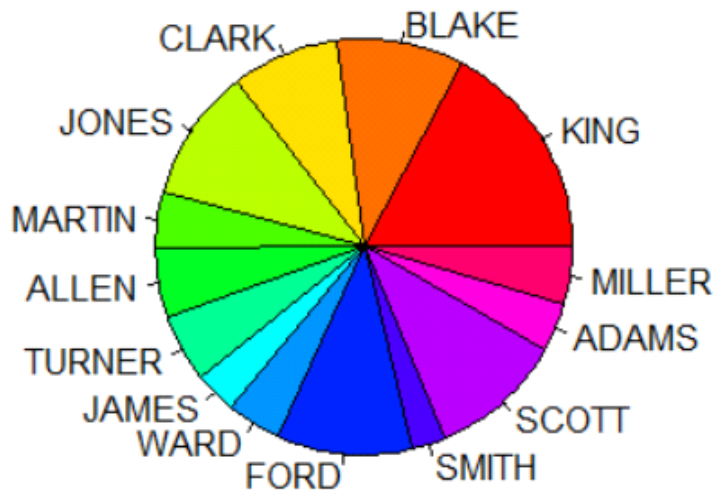
Salary Pie Chart



문제119. 문제118번을 다시 수행하는데 누구의 월급인지 이름이 출력되게 하시오.

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

Salary Pie Chart



문제120. 문제119번 그래프에 월급에 비율을 붙여서 출력하시오.

```
sal_label <- round(emp$sal/sum(emp$sal)*100,1)
```

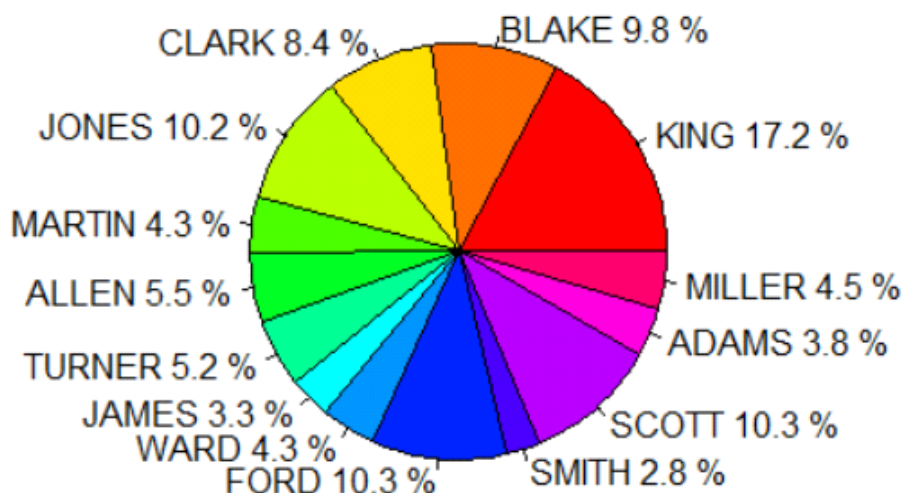
```
sal_label2 <- paste(emp$ename, sal_label, "%")
```

```
sal_label2
```

```
[1] "KING 17.2 %" "BLAKE 9.8 %" "CLARK 8.4 %"
[6] "ALLEN 5.5 %" "TURNER 5.2 %" "JAMES 3.3 %"
[11] "SMITH 2.8 %" "SCOTT 10.3 %" "ADAMS 3.8 %"
```

```
pie(emp$sal, col =rainbow(14), main = "Salary Pie Chart", label = sal_label2)
```

Salary Pie Chart



문제121. 2014년도 업종별 창업 비율을 아래와 같이 원형그래프로 그리시오.

```
x <- create_cnt[create_cnt$X == '2014', (2:8)] <- 두번째 컬럼부터 8번째까지 출력
```

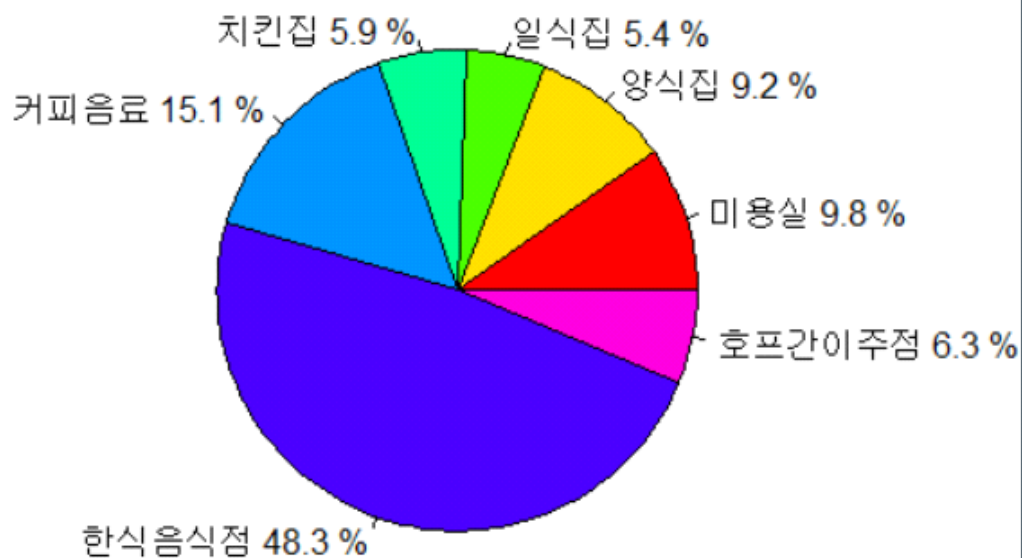
파이를 그릴려면 출력한 것이 가로로 나오면 안되고 세로로 나와야지 할 수 있다.

```
: t(x)
```

```
> t(x)
      10
미용실 1980
양식집 1870
일식집 1095
치킨집 1193
커피음료 3053
한식음식점 9772
호프간이주점 1272
```

```
x <- t(x)
cnt_label <- round(x/sum(x)*100,1)
cnt_label2 <- paste(colnames(t(cnt_label)), t(cnt_label), '%') <- 컬럼명만 가져온다.
cnt_label2
```

```
pie(x,col=rainbow(7), labels=cnt_label2)
```



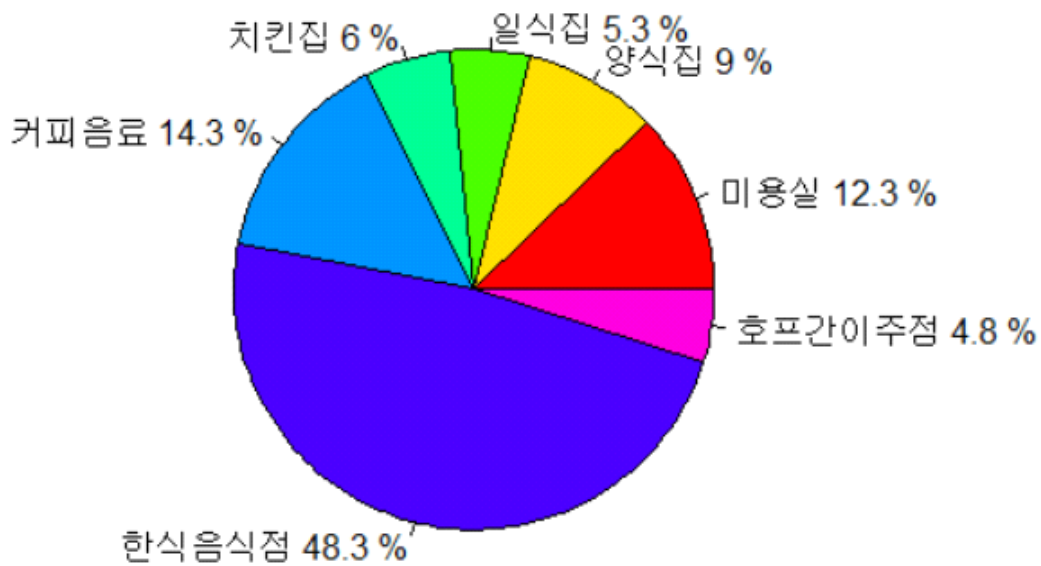
문제122. 2013년도 업종별 창업 비율을 아래와 같이 원형그래프로 그리는데 제목을 '2013년도 창업 현황'이라고 출력되게하시오.

```
x122 <- create_cnt[create_cnt$X == '2013', (2:8)]
```

```
cnt_label <- round(x122/sum(x122)*100,1)
cnt_label2 <- paste(colnames(cnt_label), t(cnt_label), '%')
cnt_label2
```

```
pie(t(x122),col=rainbow(7), labels=cnt_label2, main = '2013년도 업종별 창업 비율')
```

2013년도 업종별 창업 비율



함수 그래프 생성

원형그래프

문제123. 문제122번 코드를 가지고 함수를 생성하는데 아래와 같이 함수를 실행해서 년도를 입력하면 해당 년도의 원형 그래프가 그려지게 하시오.

보기)

show_pie()

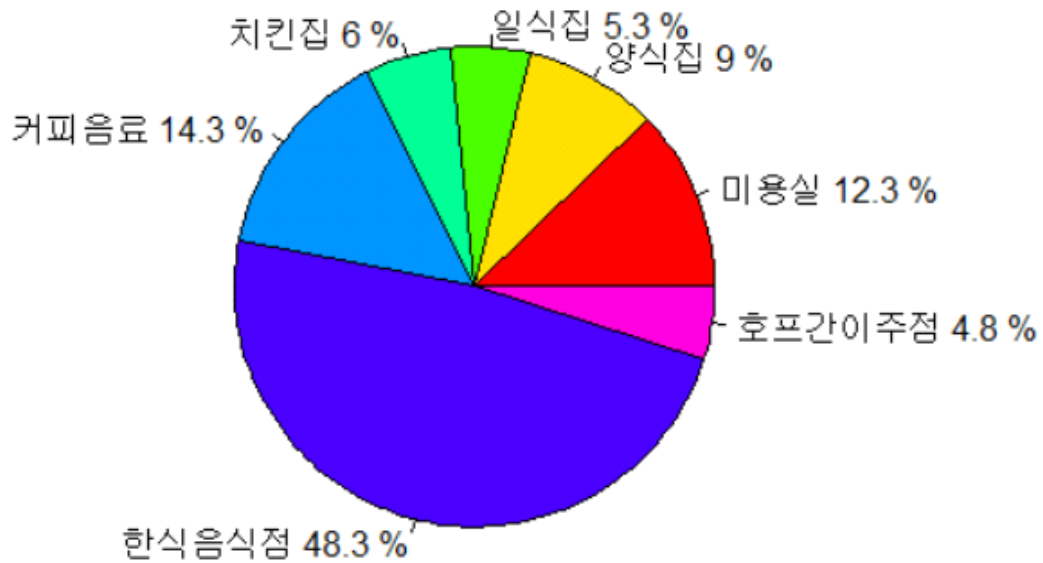
년도를 입력하세요. 2013

하면 2013년도 그래프가 그려지도록

답)

```
show_pie() <- function() {  
  graphics.off()  
  response <- readline(prompt = '년도를 입력하세요')  
  x123 <- create_cnt[create_cnt$X == response, (2:8)]  
  
  cnt_label <- round(x123/sum(x123)*100,1)  
  cnt_label2 <- paste(colnames(cnt_label), t(cnt_label), '%')  
  cnt_label2  
  
  pie(t(x123),col=rainbow(7), labels=cnt_label2, main = '2013년도 업종별 창업 비율')  
}
```


년도 업종별 창업 비율



문제124. 문제123번을하는데 제목도 변경되게 하시오.

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

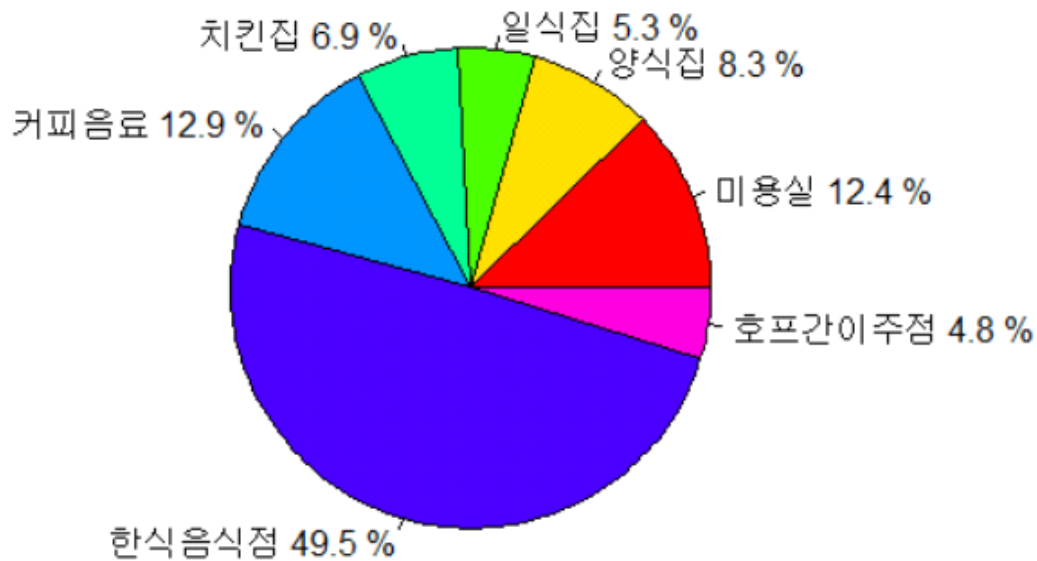
  x123 <- create_cnt[create_cnt$X == response, (2:8)]

  cnt_label <- round(x123/sum(x123)*100,1)
  cnt_label2 <- paste(colnames(cnt_label), t(cnt_label), '%')
  cnt_label2

  pie(t(x123),col=rainbow(7), labels=cnt_label2, main = paste(response, '년도 업종별 창업 비율') )
}

show_pie()
```

2012 년도 업종별 창업 비율



막대 그래프

문제125. 아래와 같이 업종을 물어보게하고 업종을 입력하면 해당 업종의 창업, 폐업 현황이 막대 그래프로 그려지는 함수를 생성하시오.

보기)

`show_bar()`

업종을 입력하세요. 치킨집

하면 치킨 그래프가 나오도록

답)

```
show_var <- function() {
```

```
  graphics.off()
```

```
  response <- readline(prompt = '업종을 입력하세요')
```

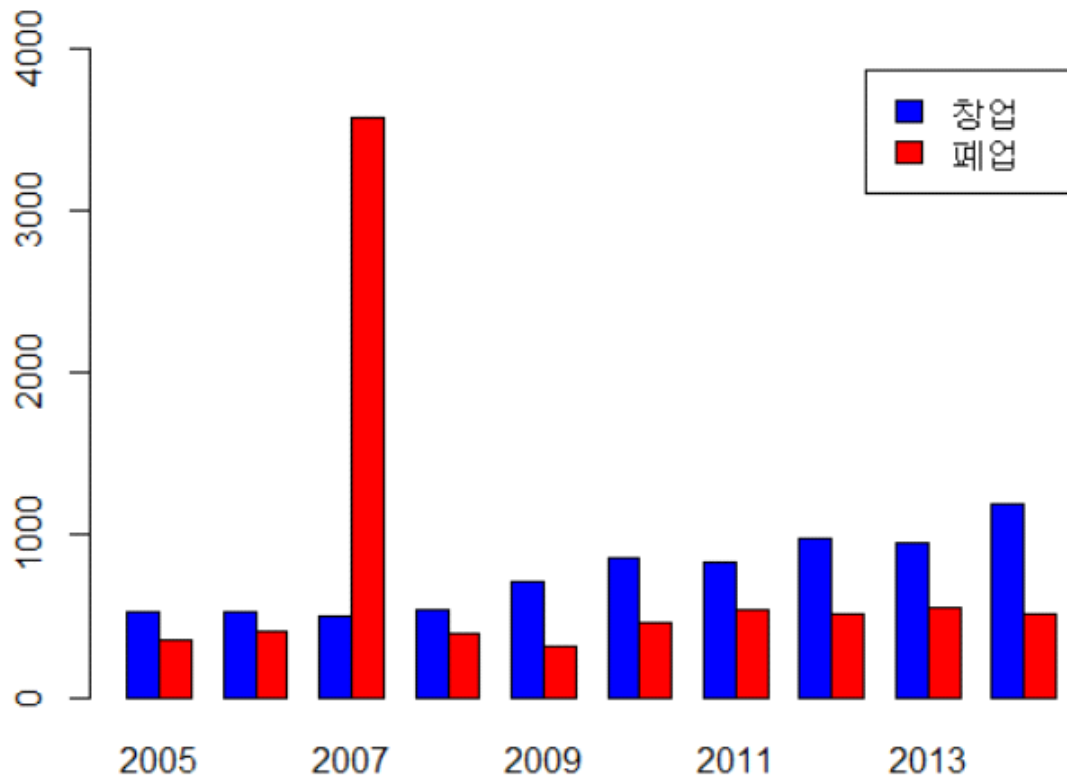
```
  x125 <- rbind(create_cnt[,response], drop_cnt[,response]) <-- $response를 쓰면 안된다.
```

```
  barplot( x125, main = paste('년도별',response," 창업,폐업건수"), names.arg=create_cnt$X,
    col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업"))
```

```
}
```

```
show_var()
```

년도별 치킨집 창업,폐업건수



설명)

	x	미용실	양식집	일식집	치킨집	커피음료	한식음식점	호프간이주점	
1	2005	2196	1034	540	530	454	5994	635	
2	2006	2028	950	577	525	483	5504	591	
3	2007	1802	1036	620	507	575	6148	544	
4	2008	1691	1127	561	543	772	6036	525	
5	2009	1826	1086	645	711	845	6577	627	
6	2010	1798	1105	669	865	1291	6689	553	
7	2011	1688	1199	736	837	1671	6900	638	
8	2012	1767	1183	753	986	1847	7082	687	
9	2013	1965	1432	839	954	2287	7708	769	
10	2014	1980	1870	1095	1193	3053	9772	1272	

행은 전부 나오게하고 response이라는 열을 가져온다.

문제126. 문제125번을 다시하는데 제목도 바뀌게 하시오.

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

  x125 <- rbind(create_cnt[,response], drop_cnt[,response])
  barplot(x125, main = paste('년도별',response," 창업,폐업건수"), names.arg=create_cnt$X,
          col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업"))
}
show_var()
```

문제127. 아래의 스크립트를 수행하고 입력했을때 값이 x7 이라는 변수에 입력되게 하시오.

```
menu(c("Yes","No"), title="Do you want this?")
```

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

```
x7
```

```
> x7
[1] 1
```

문제128. R의 menu 함수를 이용해서 아래와 같이 업종을 번호로 선택해서 막대 그래프가 출력되게 하시오.

보기)

show_bar() 를 치면

Do you want this? 가 나오고

1. 미용실
2. 양식집
- .
- .
7. 호프간이주점

답)

```
show_var <- function(){
  graphics.off()
  response <- menu(c("X","미용실","양식집","일식집","치킨집","커피집","한식음식점","호프간이주점"), title = 'Do
you want this?')
```

```

x128 <- rbind(create_cnt[,response], drop_cnt[,response])
barplot( x128, main = paste('년도별',response," 창업,폐업건수"), names.arg=create_cnt$X,
        col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업"))
}
```

```
show_var()
```

```
Do you want this?

1: X
2: 미용실
3: 양식집
4: 일식집
5: 치킨집
6: 커피집
7: 한식음식점
8: 호프간이주점
```

이렇게 할 수도 있고,

```
show_var <- function(){
  graphics.off()
  response <- menu(colnames(create_cnt[, 2:7]), title = 'Do you want this?')

  x128 <- rbind(create_cnt[,response+1], drop_cnt[,response+1])
  barplot( x128, main = paste('년도별',response," 창업,폐업건수"), names.arg=create_cnt$X,
        col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업"))
```

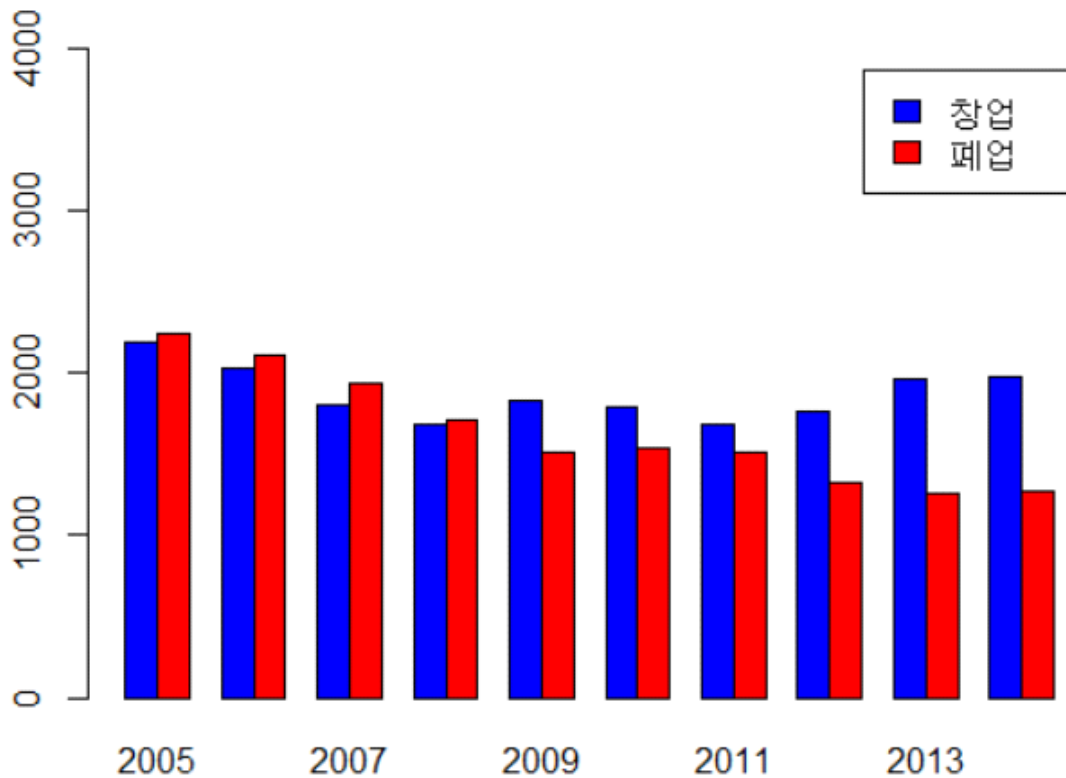
```

}
show_var()
> show_var()
Do you want this?

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

```

년도별 1 창업,폐업건수



문제129. 제목도 깔끔하게 바꾸시오.

ex) `colnames(create_cnt)[2]` <- [] 번째 컬럼 뽑아내기

```

> colnames(create_cnt)[2]
[1] "미용실"

```

이걸 활용하면 된다.

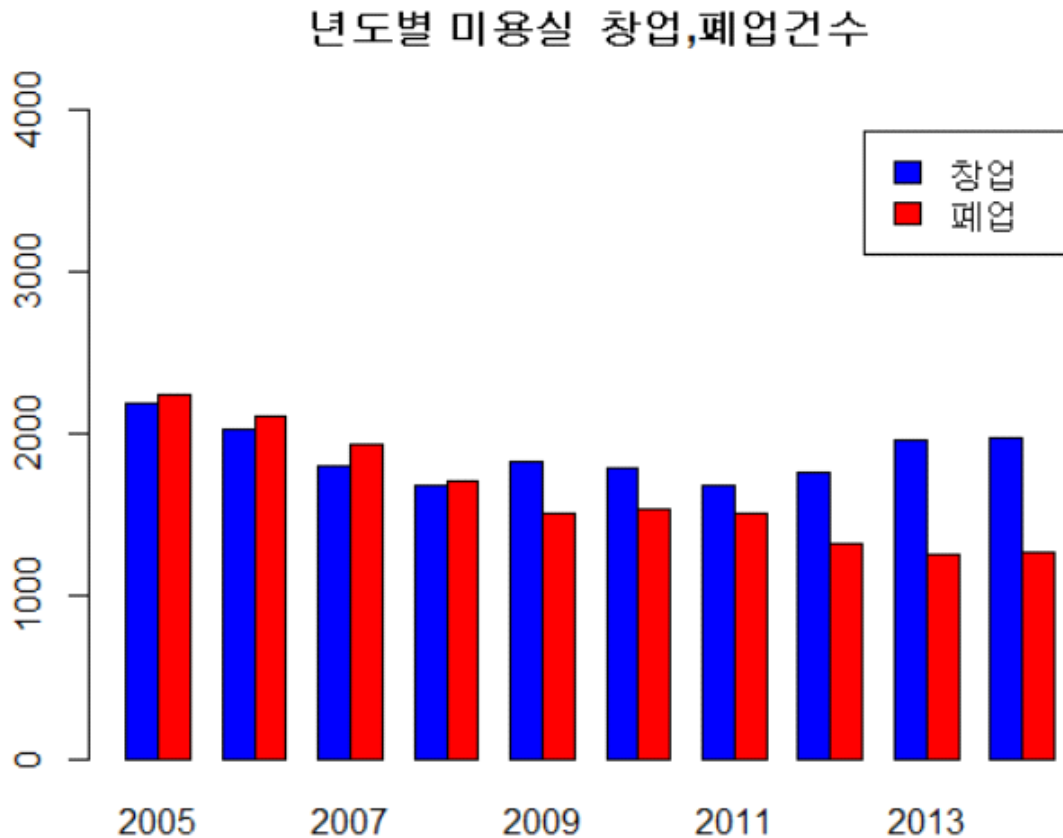
```

show_var <- function(){
  graphics.off()
  response <- menu(colnames(create_cnt[,2:7]), title = 'Do you want this?')

  x129 <- rbind(create_cnt[,response+1], drop_cnt[,response+1])
  barplot( x129, main = paste('년도별',colnames(create_cnt)[response+1]," 창업,폐업건수"),
names.arg=create_cnt$X,
          col=c('blue','red'), ylim= c(0,4000), beside = T, legend = c("창업","폐업"))
}

```

show_var()



문제130. 직업, 입사한 년도(4자리), 직업별 입사한 년도별 토달월급을 출력하시오.

```
x130 <- tapply(emp$sal, list(emp$job, year(emp$hiredate)), sum)
x130[is.na(x130)]<-0
x130
```

```
> x130
      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
```

문제131. 아래의 결과에서 column name과 row name을 각각 출력하시오.

```
> x130
      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(x130)
```

```
rownames(x130)
```

```
> colnames(x130)
[1] "1980" "1981" "1982" "1983"
> rownames(x130)
[1] "ANALYST" "CLERK" "MANAGER" "PRESIDENT" "SALESMAN"
```

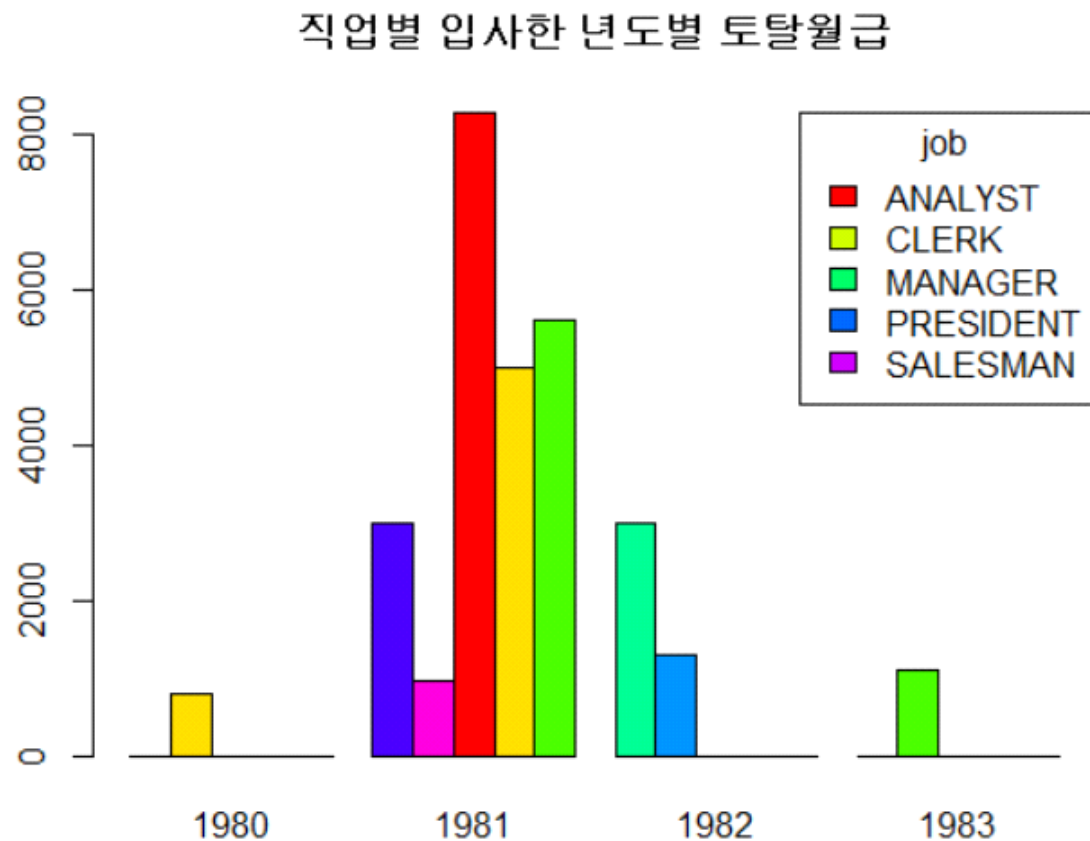
문제132. 아래의 데이터 결과를 막대 그래프로 시각화 하시오.

```
> x130
```

	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

```
barplot(x130, col=rainbow(7),
        beside = T,
        main = '직업별 입사한 년도별 토탈월급' )
```

legend("topright", rownames(x130), fill= rainbow(5), title = 'job') <- 레전드 위치 조정



x11() <- 그래프 그릴 창의 크기를 설정하기 위해 뒤편

문제133. 아래와 같이 막대 그래프를 그릴 컬럼을 물어보게하고 컬럼을 각각 입력되면 막대 그래가 그려지게 하시오.

보기)

show_emp_bar()

가로가 될 컬럼명을 입력하세요 job

세로가 될 컬럼명르 입력하세요 deptno

답)

```
show_var <- function() {
```

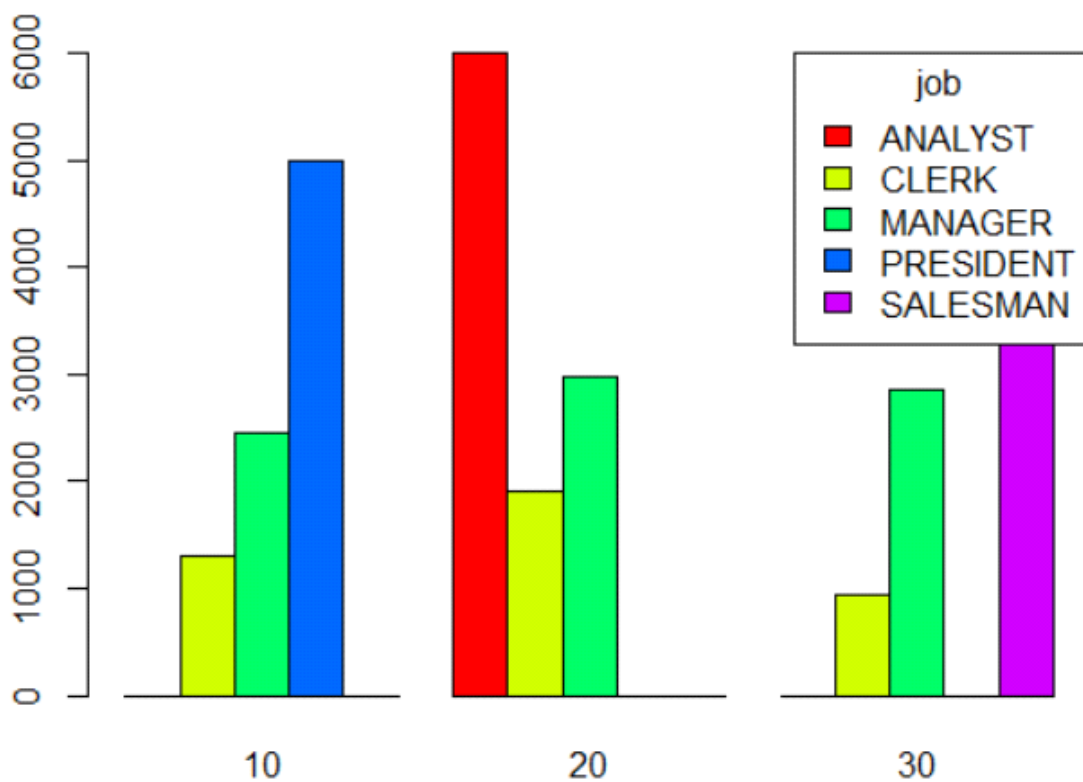
```

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

x133 <- tapply(emp$sal, list(emp[,response2], emp[,response1]), sum)
x133[is.na(x133)]<-0

barplot(x133, col=rainbow(5),
        beside = T)
legend("topright", rownames(x133), fill= rainbow(5), title = 'job')
}
show_var()

```



문제134. 아래와 같이 막대 그래프를 그릴 컬럼을 물어보게하고 컬럼을 각각 입력되면 막대 그래프가 그려지게 하시오.

```

보기)
show_emp_bar()

```

가로가 될 컬럼명을 선택하세요 2

1. job
2. deptno

세로가 될 컬럼명을 입력하세요 1

1. job
2. deptno

답)


```

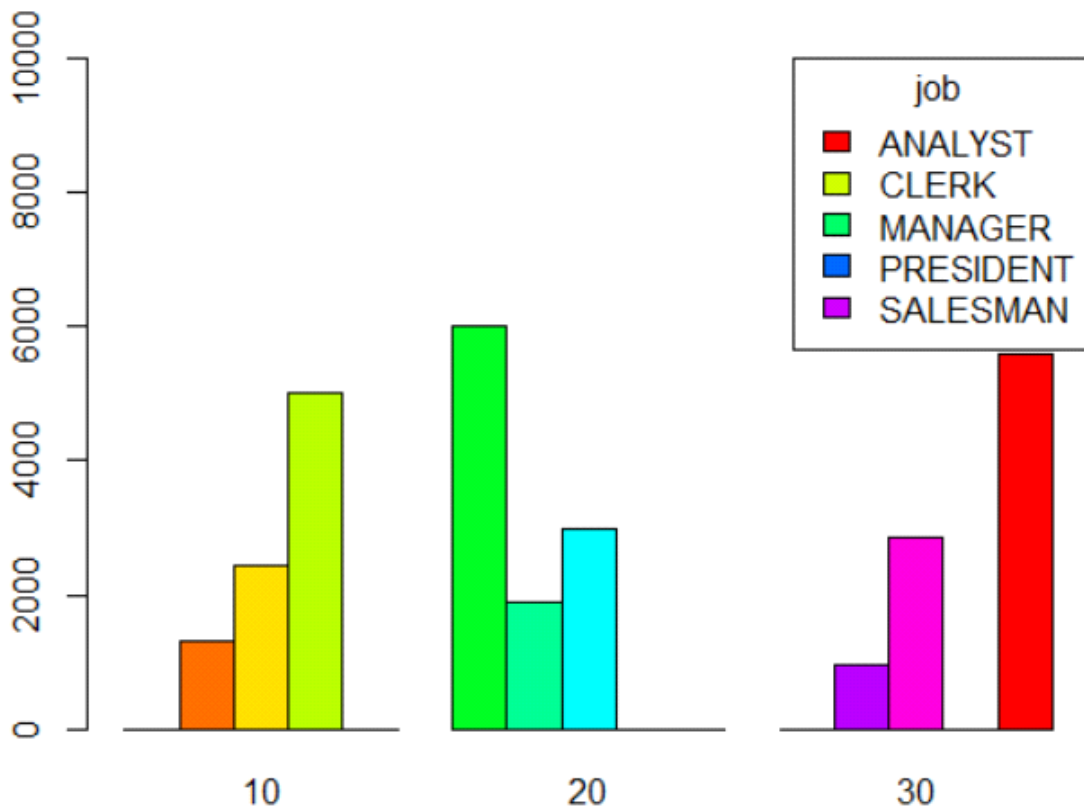
show_bar <- function() {
  graphics.off()
  response1 <- menu(colnames(emp),title = '가로가 될 컬럼명을 선택하시오')
  response2 <- menu(colnames(emp),title = '세로가 될 컬럼명을 선택하시오')

  x134 <- tapply(emp$sal, list(emp[,response2], emp[,response1]), sum)
  x134[is.na(x134)]<-0

  barplot(x134, col=rainbow(14),
          beside = T, ylim = c(0,10000) )
  legend("topright", rownames(x134), fill= rainbow(5), title = 'job')
}

```

show_bar()



#####legend

inset = legend의 여백 조절

cex = 글씨 크기 조절

다른답)

```

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]

```

```

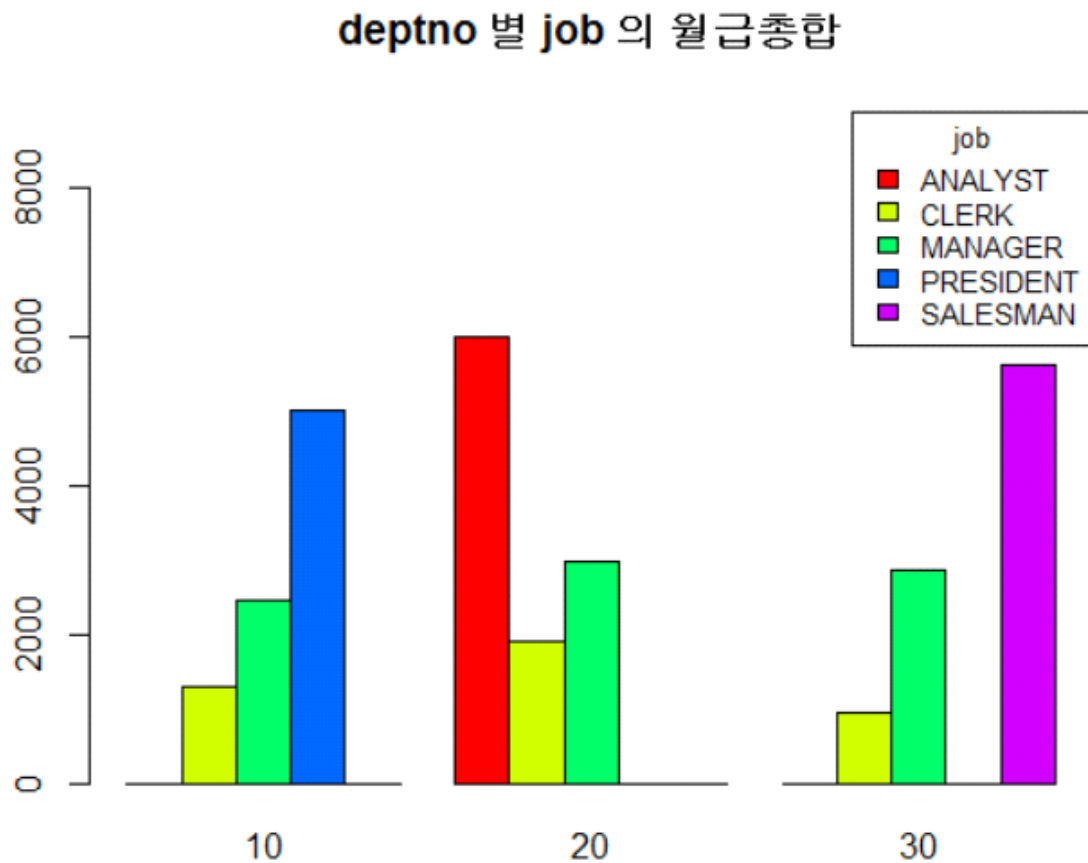
x2 <- colnames(emp)[x2]

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)
}
show_emp_bar()

```

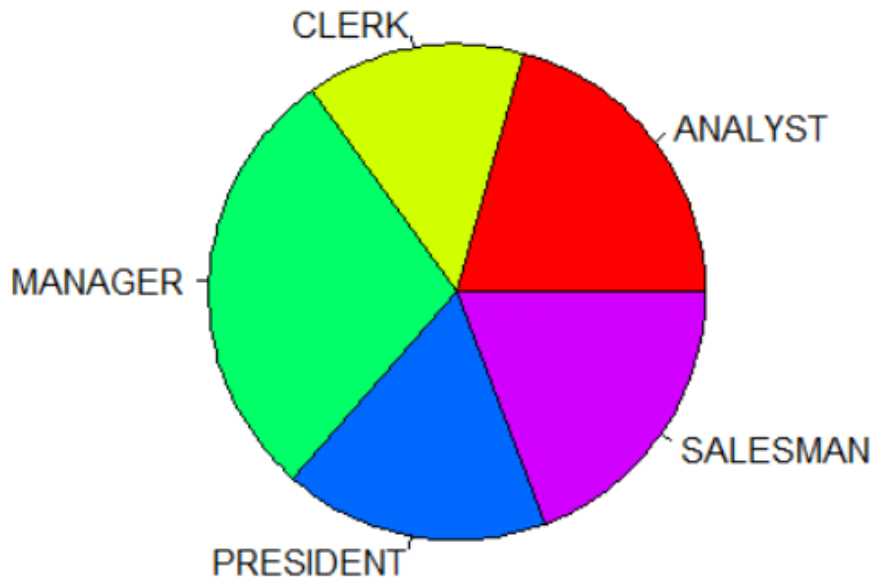


문제135. 직업, 직업별 토탈월급을 원형 그래프로 그리시오.

```

x135 <- tapply(emp$sal, emp$job, sum)
pie(x135, col = rainbow(5))

```

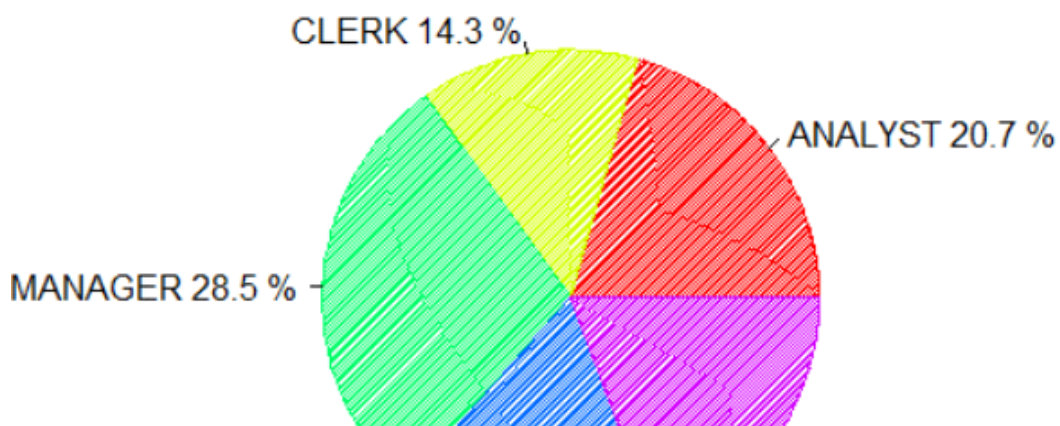
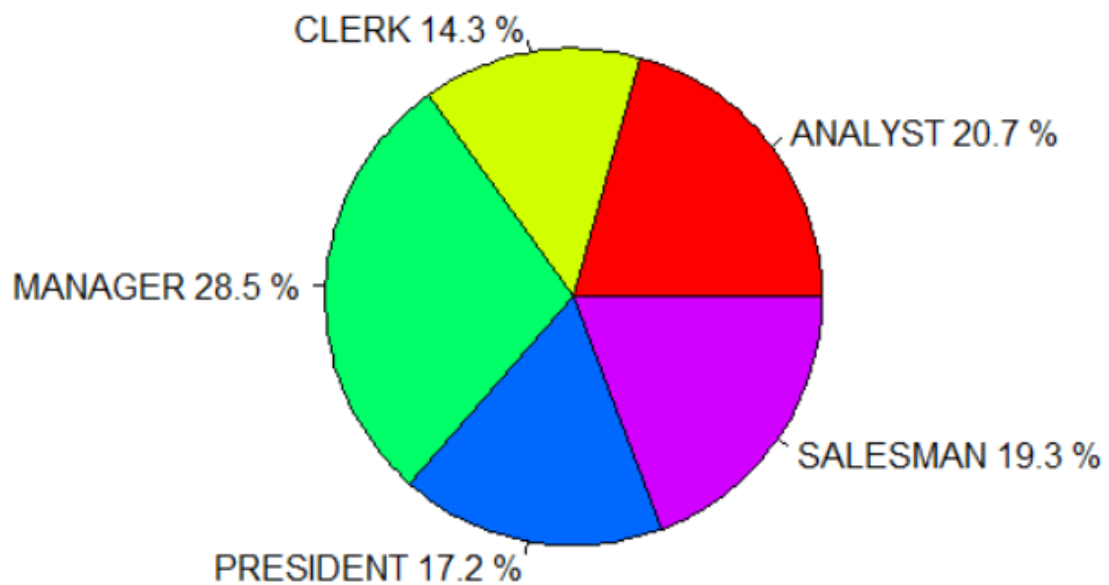


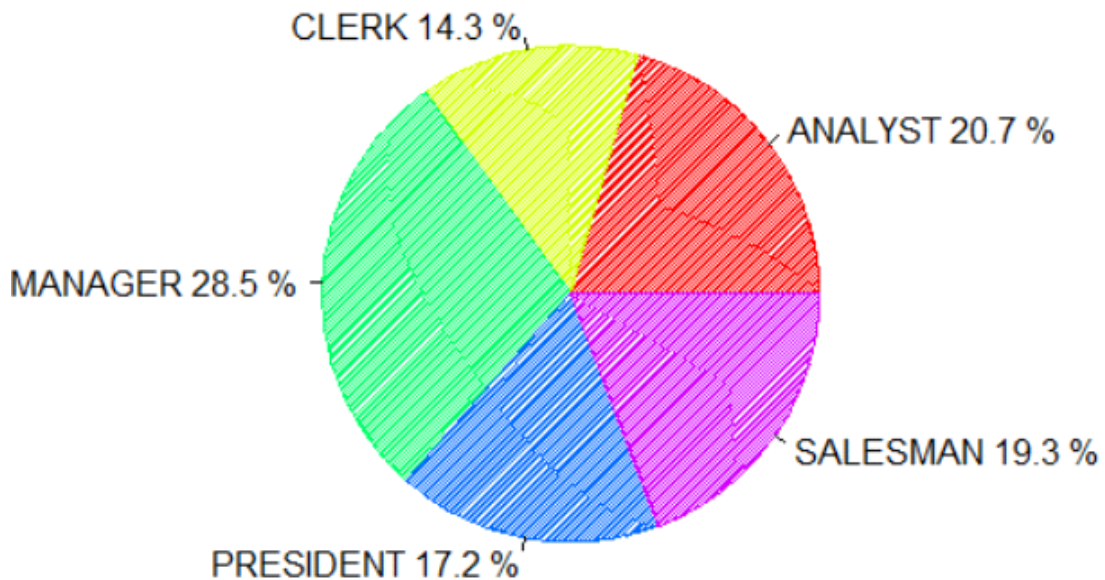
문제136. 위의 그래프에 직업 옆에 비율을 표시하시오.

```
label <- round(x135/sum(x135)*100,1)
```

```
label2 <- paste(sort(unique(emp$job)), label, '%') <-- sort : abc순서대로
```

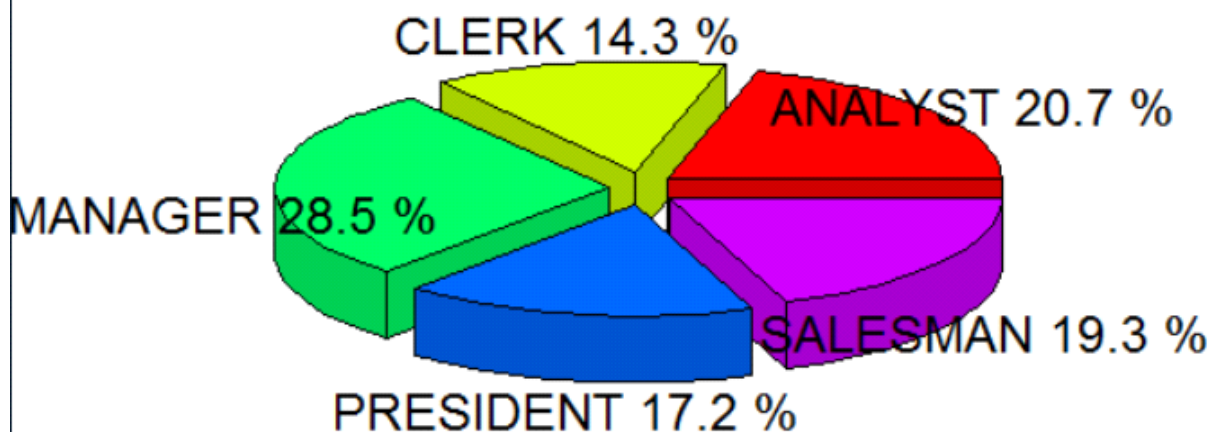
```
pie(x135, col = rainbow(5), labels=label2, density=80) <-- 파스텔처럼나옴
```





문제137. 문제136번 그래프를 3D로 출력하시오.

```
install.packages("plotrix")
library(plotrix)
pie3D(x135, explode = 0.1, labels=label2)
```



문제138. 아래의 switch문을 구현해보시오.

```
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", #문제110
        cex.names = 0.5, col=rainbow(14))
    },
    piechart={
      sal_label <- round(emp$sal/sum(emp$sal)*100,1)
```

```
sal_label2 <- paste(emp$ename, sal_label, "%") #문제120
```

```
sal_label2
```

```
pie(emp$sal, col =rainbow(14), main = "Salary Pie Chart",  
    label = sal_label2)
```

```
},  
{  
  print('default')  
}
```

```
)
```

```
}
```

```
func()
```

```
> graphe()
```

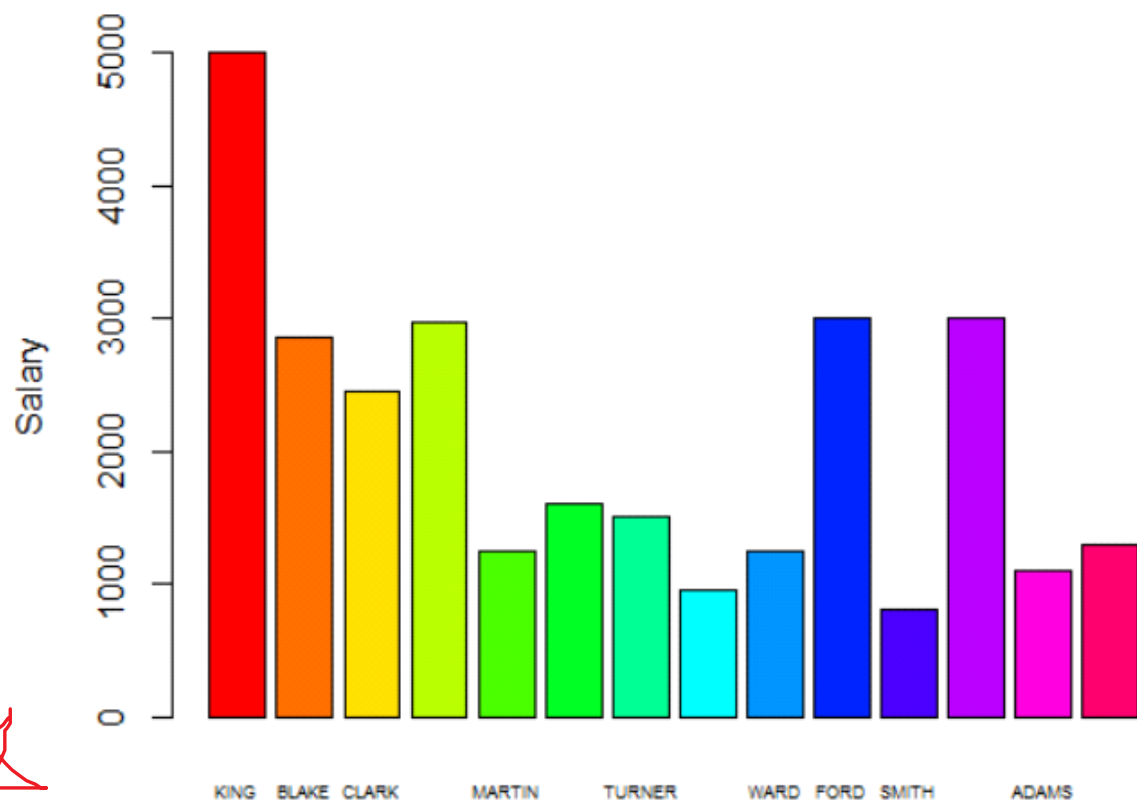
원하는 그래프의 숫자를 선택하세요

1: 막대그래프

2: 원형그래프

선택: 1

Salary Bar Chart



문제139. 아래와 같이 그래프를 그릴 컬럼을 물어보게하고, 그래프의 종류도 물어보게해서 그래프가 그려지는 함수를 생성하시오.

보기)

func()

토탈값을 구할 컬럼번호를 입력하세요~

1. empno
2. ename

.

.

그룹핑할 컬럼번호를 입력하세요~

1. empno
2. ename

.

.

그래프의 종류를 입력하세요~

1. 막대 그래프
2. 원형 그래프

```
graphics.off()
```

```
x1 <- menu(colnames(emp), title='토탈값을 구할 컬럼번호를 입력하시오~')
```

```
x2 <- menu(colnames(emp), title = '그룹핑할 컬럼번호를 입력하시오~')
```

```
x3 <- menu(c("막대그래프","원형그래프"), title = '그래프 종류를 입력하시오!')
```

```
y1<-colnames(emp[x1])
```

```
y2<-colnames(emp[x2])
```

```
x1 <-emp[, x1]
```

```
x2 <-emp[, x2]
```

```
x<- tapply(x1,x2,sum)
```

```
label<-paste(sort(unique(colnames(t(x)))), round(x/sum(x)*100,1),'%')
```

```
mains <- paste(y2, '별', y1, '합계')
```

```
switch(x3,
```

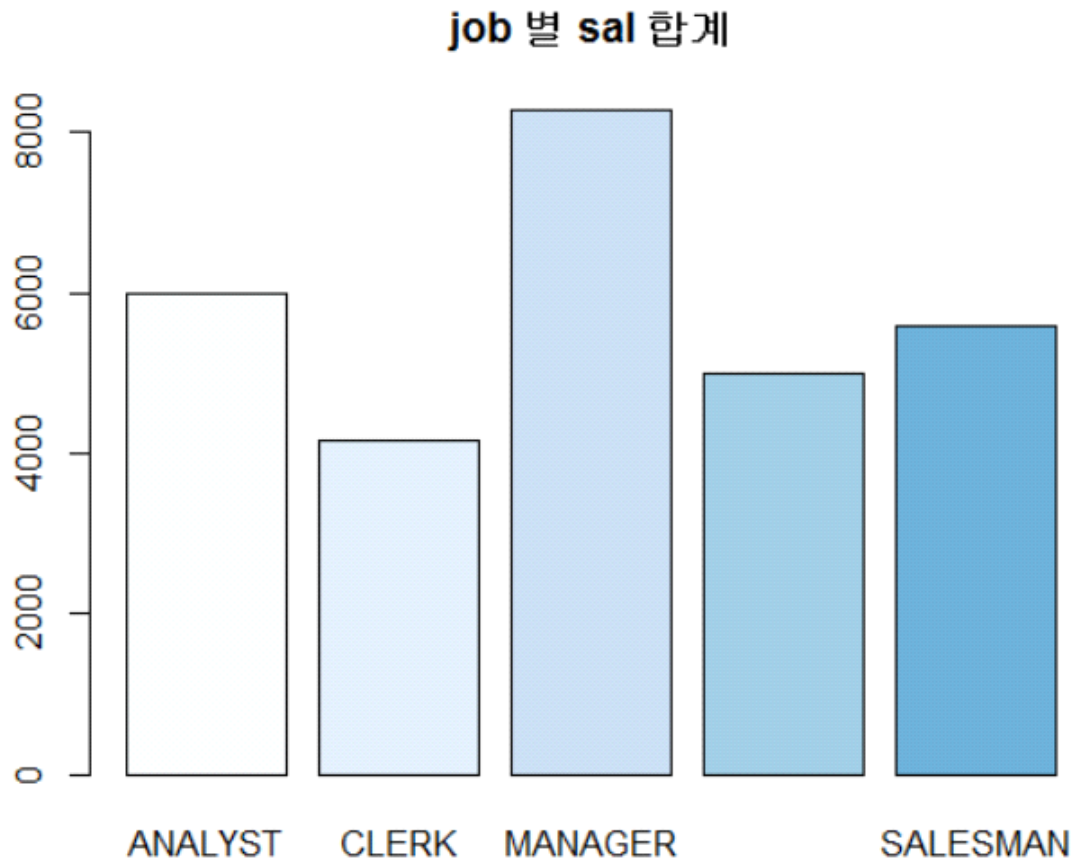
```
  bar=barplot(x, names.arg=colnames(t(x)),col=blues9,ylab = y1,main = mains),
```

```
  piee=pie(x, col = blues9, labels = label, main = mains)
```

```
)
```

```
}
```

```
func()
```



문제140. 오라클과 R을 연동하시오.(오라클 서비스 작동 시키고)

사전 작업

1. 오라클 서비스를 올린다.
2. 오라클 리스너의 상태를 확인한다.

OracleOraDb11g_home1TNSListener 시작됨
OracleServiceORCL 시작됨

cmd에 `lsnrctl status` 친다.

```
"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
```

확인하면 이렇게 나오면 경비원아저씨가 잘 있는거라고 생각

`sqlplus scott/tiger@orcl <- ex` 경비원 아저씨를 통해서 접속

3. 리스너 통해서 오라클에 접속이 되는지 확인한다.

```
install.packages('DBI')
install.packages('RJDBC')
library(RJDBC)
library(DBI)
```

---Rjava 오류시 해결방법

java깎면된다.

카페에서 ojdbc6를 받고 이름을 .jar로 바꿔준다

```
driver <- JDBC('oracle.jdbc.driver.OracleDriver', 'ojdbc6.jar')
```

```
oracle_db <- dbConnect(driver, 'jdbc:oracle:thin:@//127.0.0.1:1521/orcl', 'scott', 'tiger')
```

```
emp_query <- "select * from emp"
```

```
emp_data <- dbGetQuery(oracle_db, emp_query)
```

```
emp_data
```

ex. 직업, 직업별 토달월급을 출력하는데 직업별 토달월급들의 평균값보다 더 큰 값만 출력되게 하시오!

```
# select *
# from (select job,
#       sum_sal,
#       avg(sum_sal) over() avg_sal
#       from (select job,
#               sum(sal) sum_sal
#               from emp
#               group by job))
# where sum_sal > avg_sal;
```

문제141. emp2 테이블의 데이터를 emp2_data에 로드하고 통신사, 나이를 가지고 아래의 그래프를 그리시오.

```
func<-function(){
  graphics.off()
  q1<-menu(colnames(emp2),title='토달값을 구할 컬럼번호를 입력하여라')
  q2<-menu(colnames(emp2),title='그룹핑할 컬럼번호를 선택하여라.')
  q3<-menu(c('막대그래프','원형그래프'),title='그래프의 종류를입력하여라')

  m1<-colnames(emp2[q1])
  m2<-colnames(emp2[q2])

  q1<-emp2[,q1]
  q2<-emp2[,q2]

  x<-tapply(q1,q2,sum)

  label<-paste(sort(unique(colnames(t(x)))),(' ',round(x/sum(x)*100,1),'%'))
  mains<-paste(m2,'별 ',m1,' 합계')

  switch(q3,
    bar=barplot(x,names.arg=colnames(t(x)),col=blues9, ylab=m1,cex.names=0.7,
                main=mains),
    pie=pie(x,col=blues9,labels=label,main=mains))
```



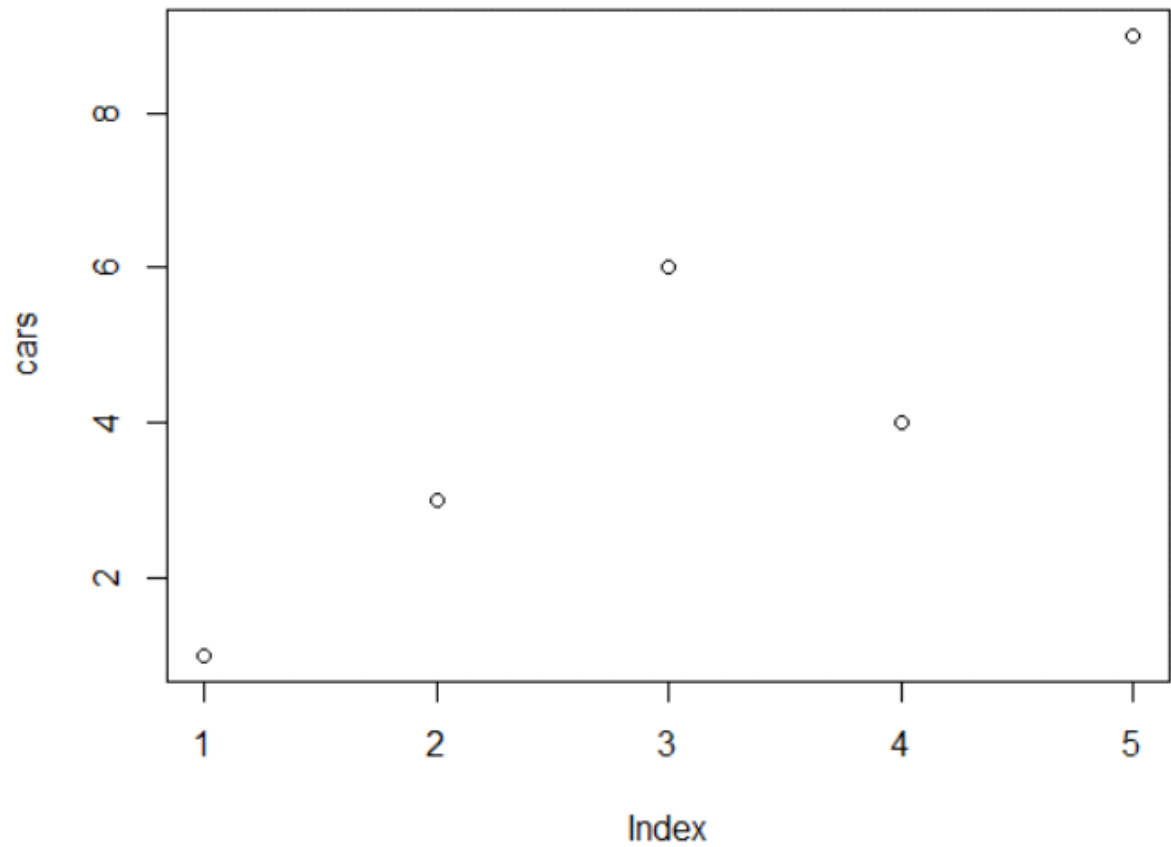
```
}  
func()
```

3. Plot 그래프

ex) 아래의 점(plot) 그래프를 그리시오.

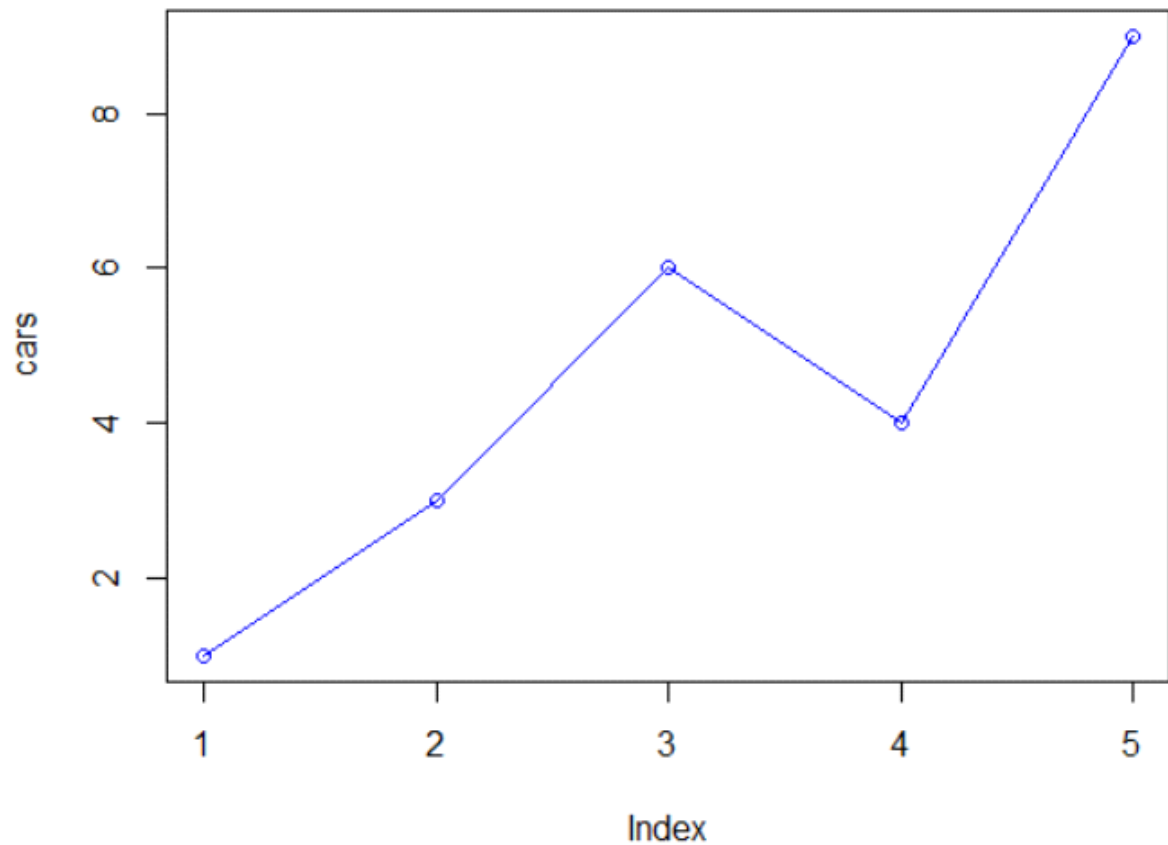
```
graphics.off()  
cars <- c(1,3,6,4,9)  
cars
```

```
plot(cars)
```



문제142. 아래의 그래프에 파란색 선을 그리시오.

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

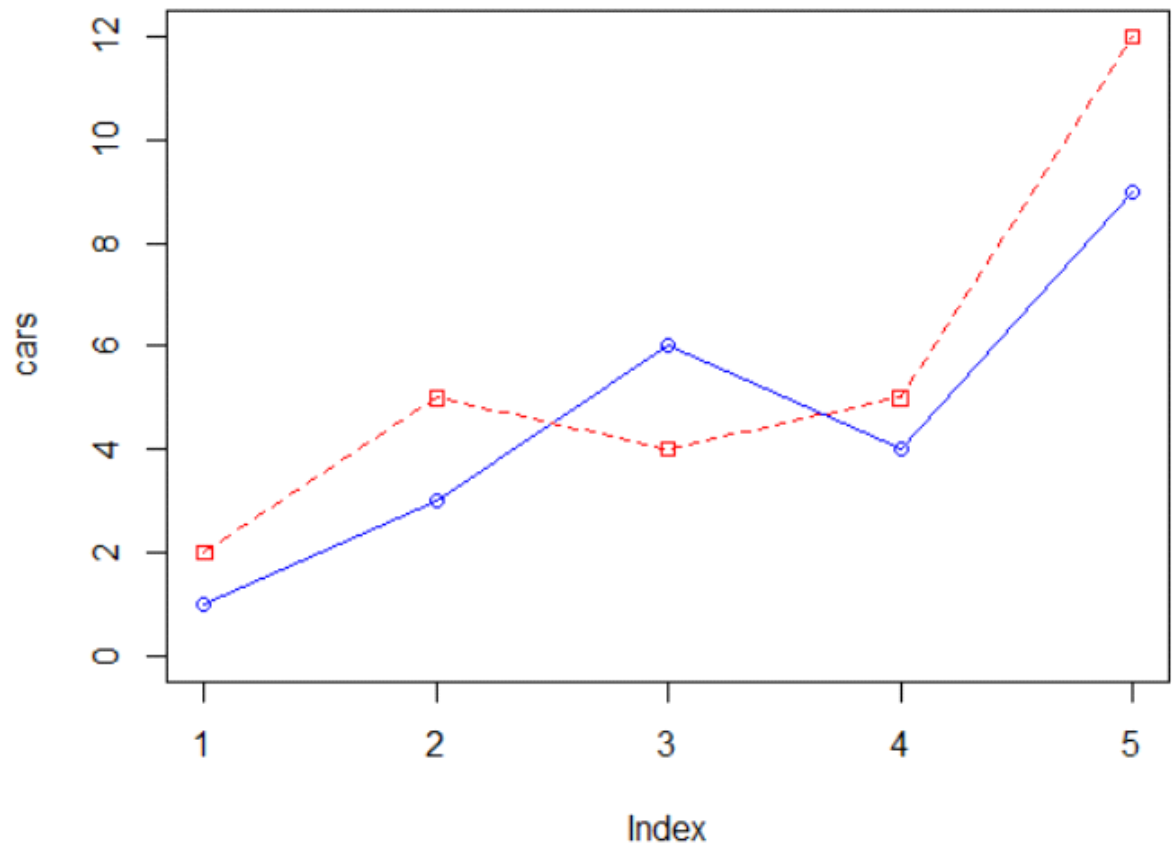


plot type 옵션

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"	계단모양으로 (stair steps)
"S"	계단모양으로 (upper stair steps)
"n"	좌표찍지 않음

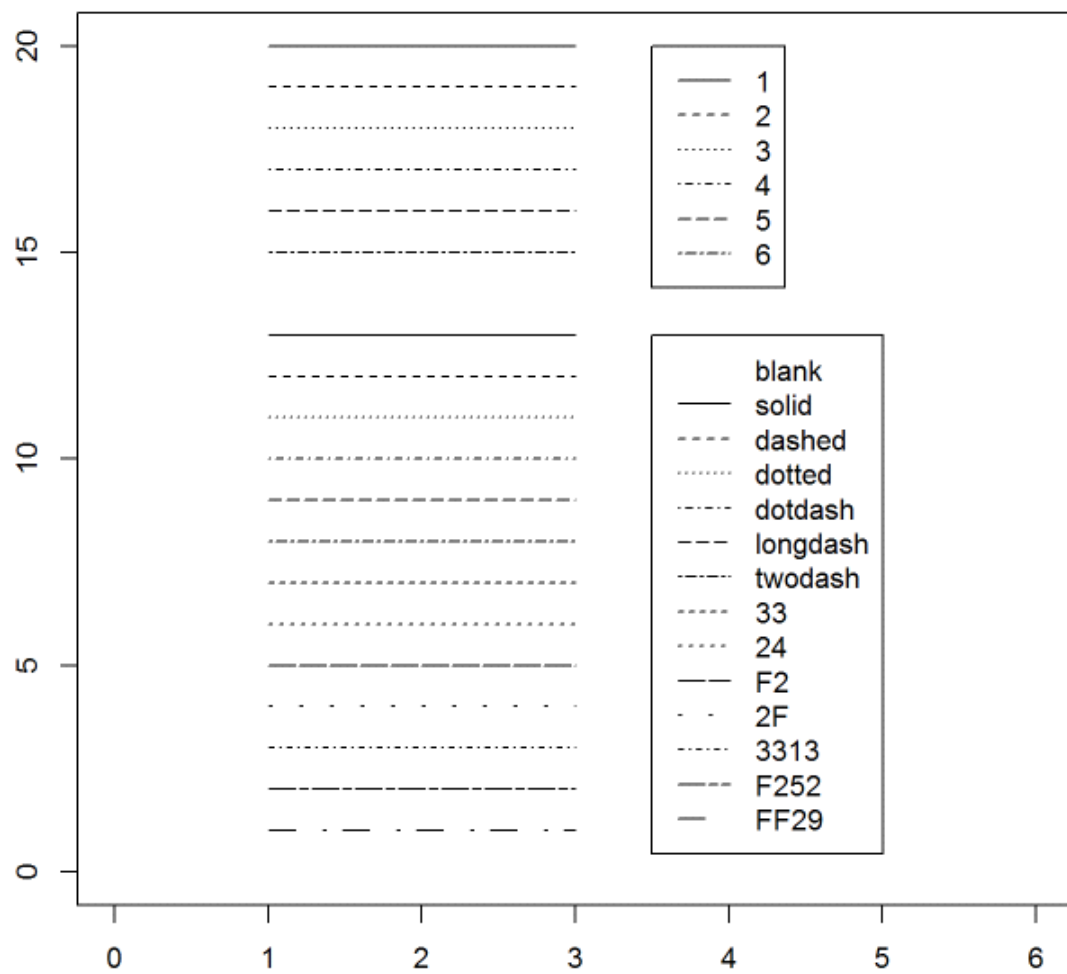
문제143. 차와 트럭의 팔린 댓수를 라인 그래프로 시각화 하시오.

```
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))
#그래프 창을 닫지 말고 바로 이어서
lines(truck,type = 'o',pch=22, lty=2, col='red')    <- pch = 22 : 네모, lty = 2 : 점선
```



plot 그래프에 pch옵션

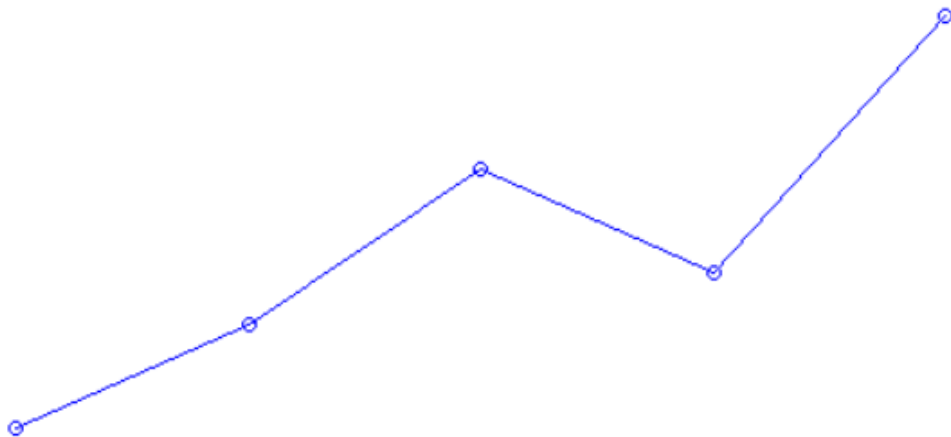
lines



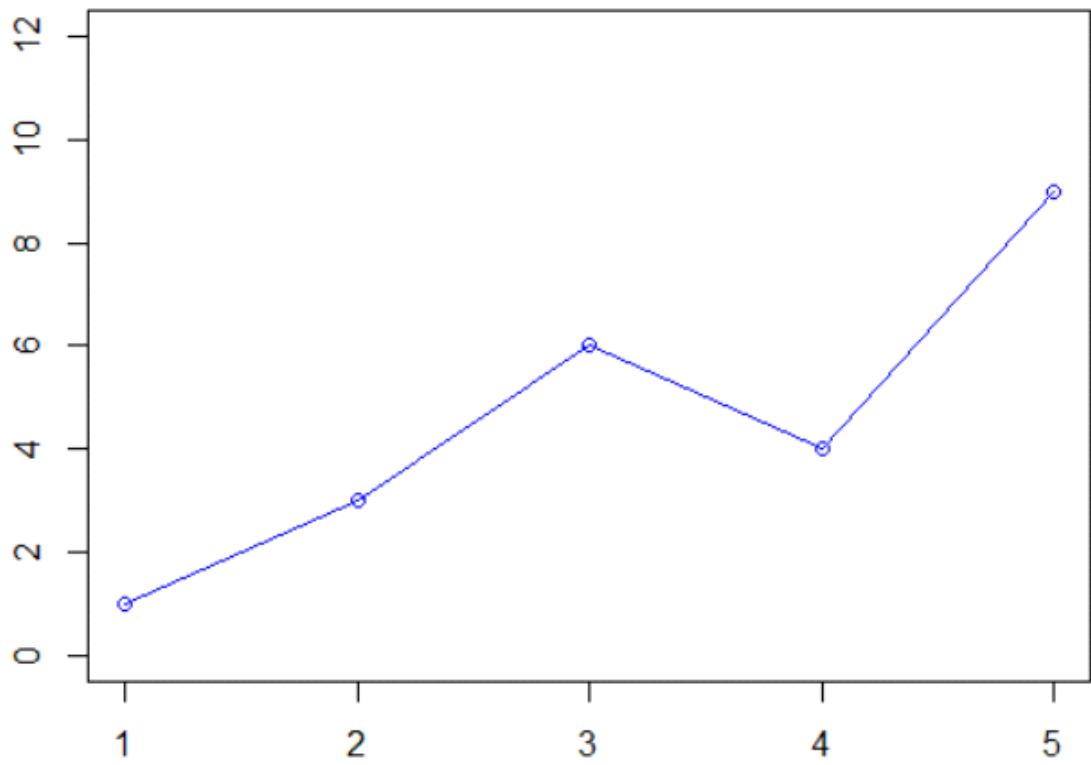
- lty = 0, lty = "blank" : 그리지 않음
- lty = 1, lty = "solid" : 실선 (기본값)
- lty = 2, lty = "dashed" : 대시
- lty = 3, lty = "dotted" : 점
- lty = 4, lty = "dotdash" : 점과 대시
- lty = 5, lty = "longdash" : 긴 대시

문제144. 다시 앞의 그래프를 아래의 순서대로 시각화 작업을 하시오.

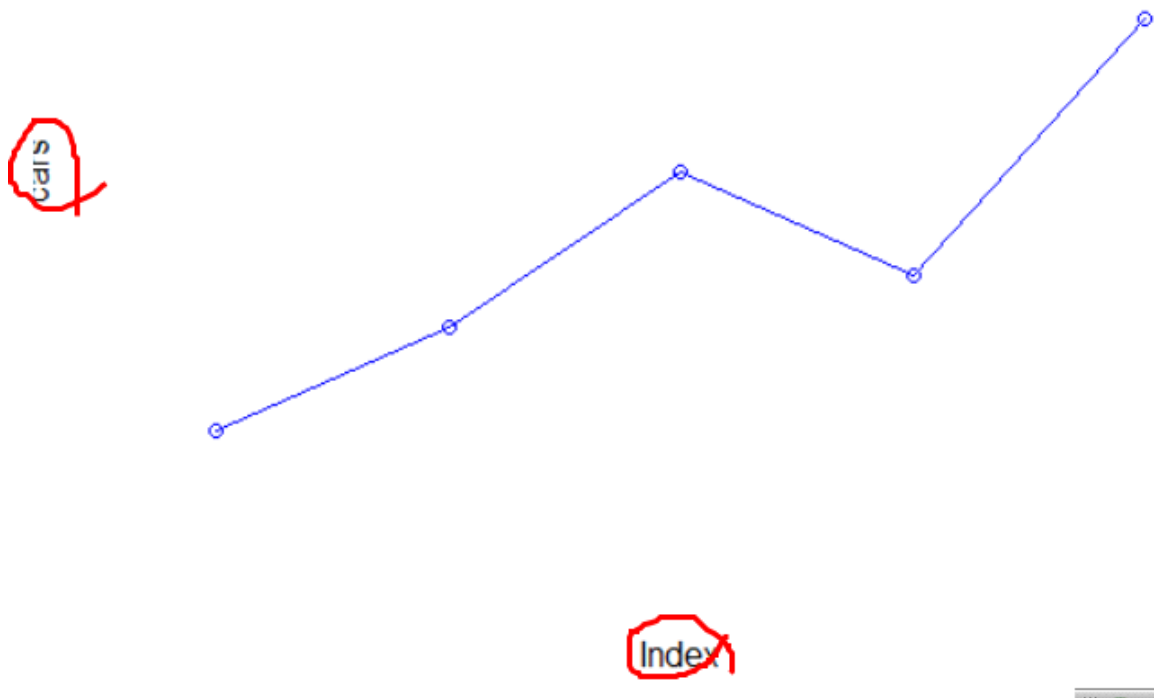
```
graphics.off()
cars <- c(1,3,6,4,9)
trucks <- c(2,5,4,5,12)
g_range <- range(0,cars,trucks) #0부터 cars,trucks중의 최대값
g_range
plot(cars,type = 'o', col='blue',ylim=g_range, axes=FALSE,ann=FALSE)
```



axes = 박스(축)

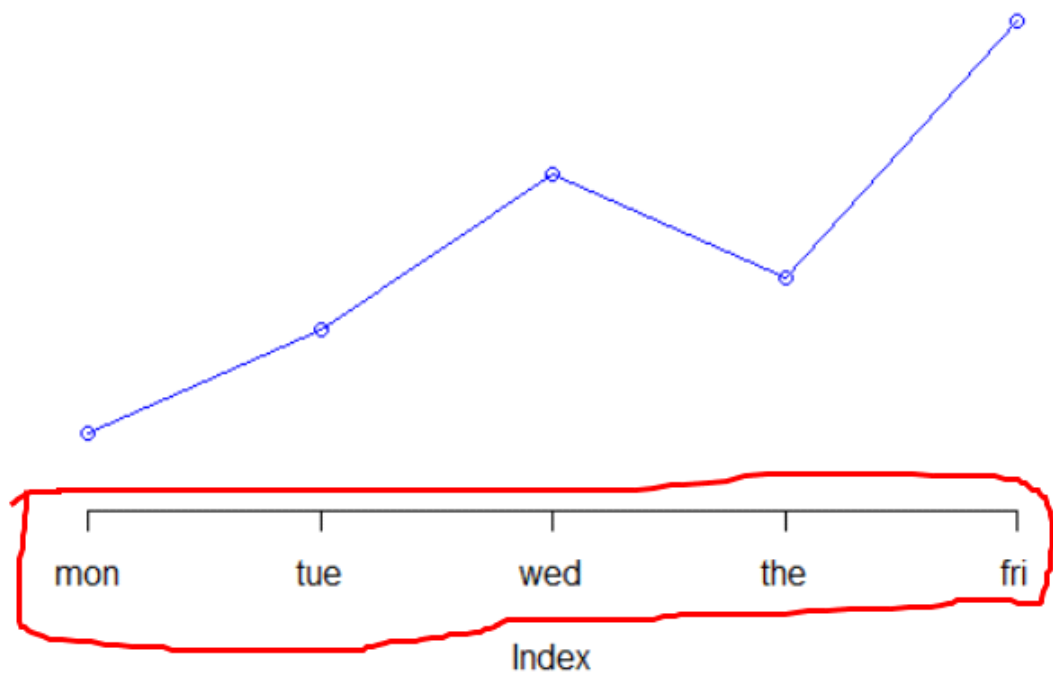


ann = 축의 제목

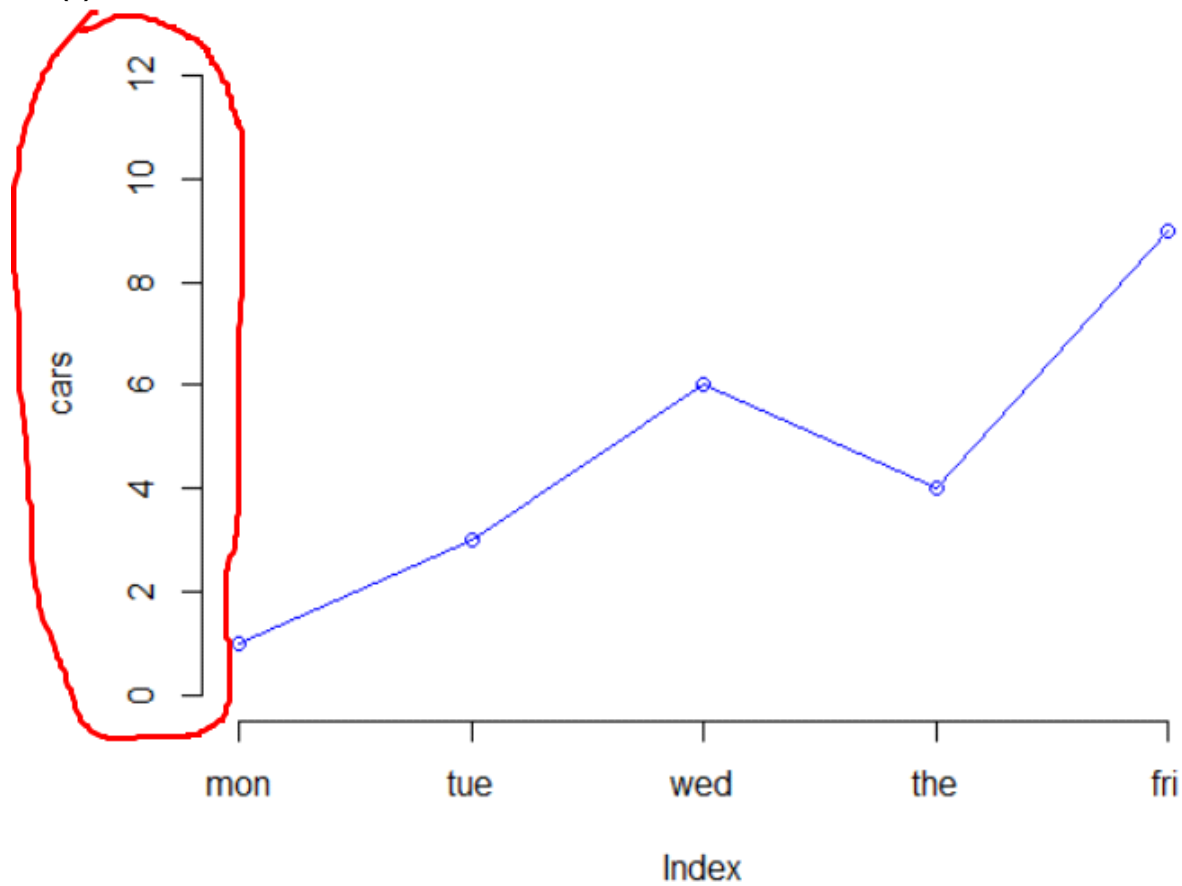


`axis(1, at=1:5, lab=c('mon','tue','wed','the','fri'))`

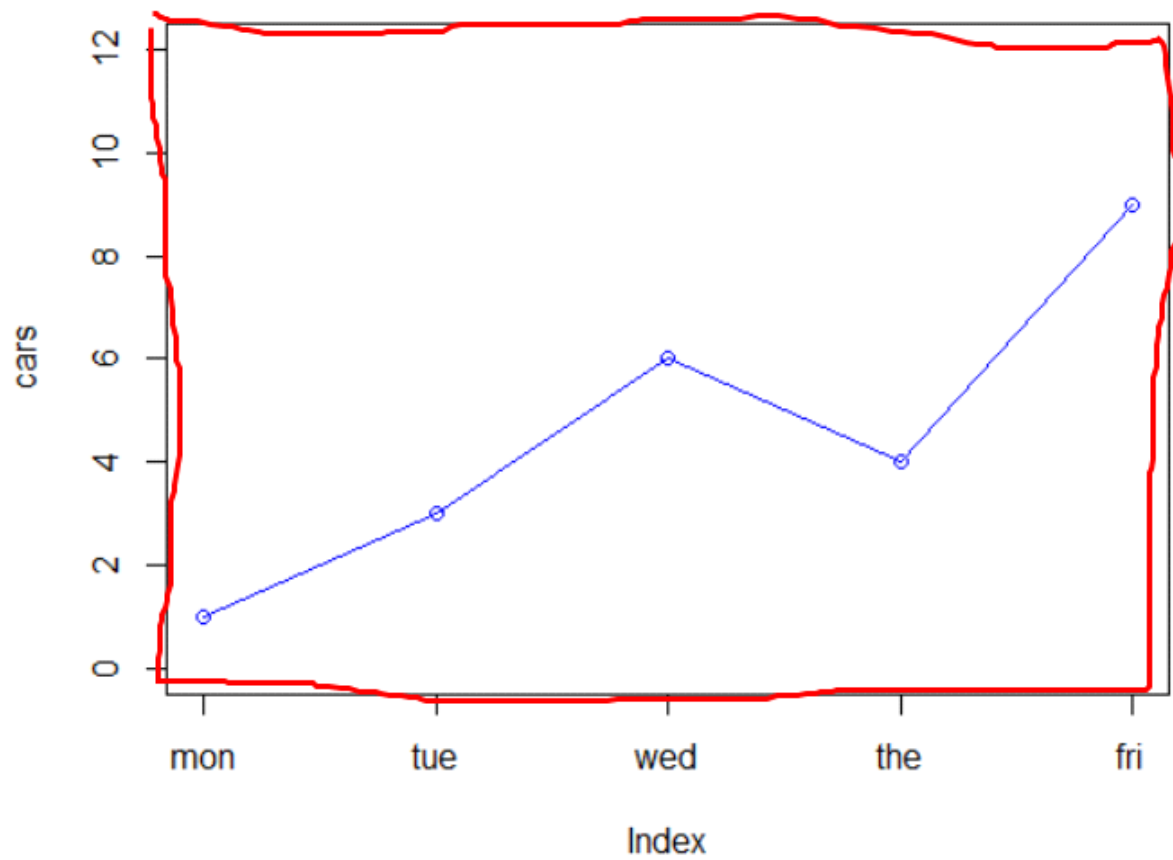
cars



axis(2)

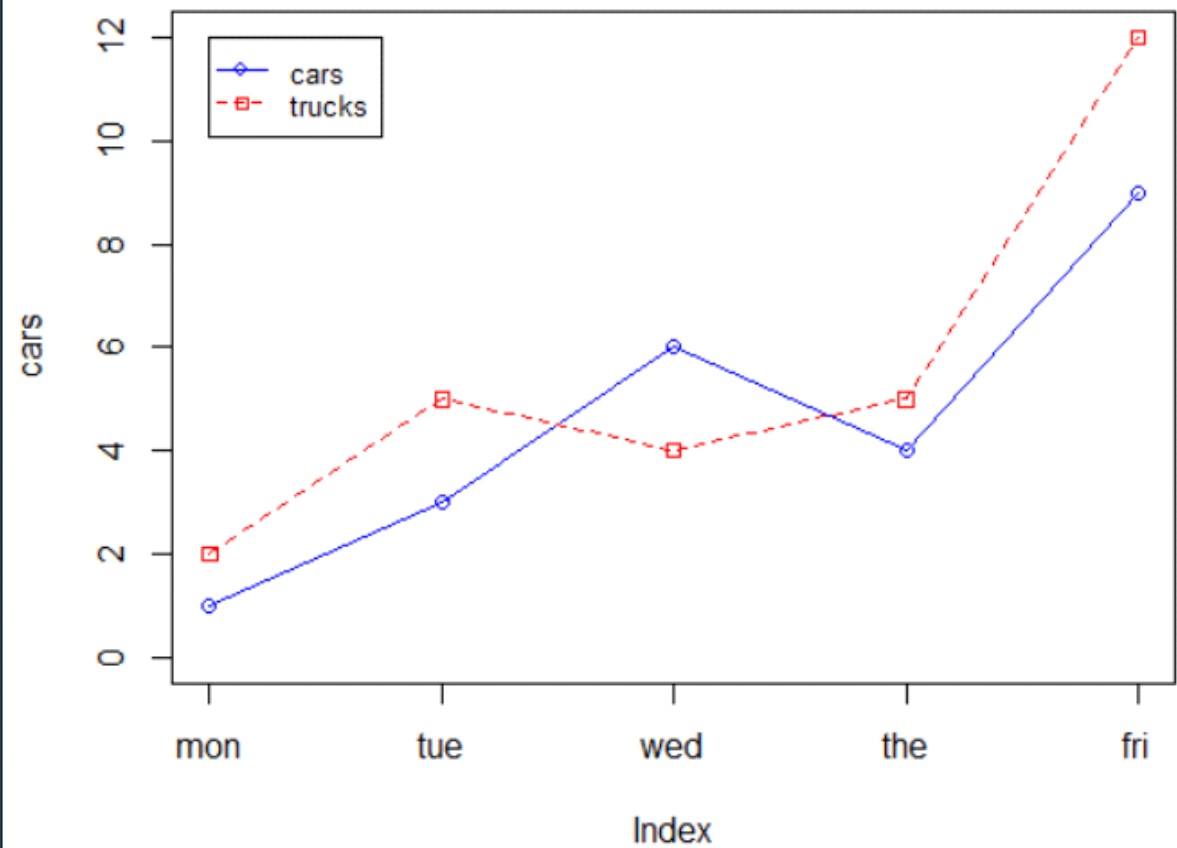


box()



```
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)
```



pch = 21 : 동그라미 lty=1 : 직선
 pch = 22 : 네모 lty = 2 : 점선

문제145. 2018년 1월달 우리나라 지하철 총승차인원수 데이터를 R로 로드하고 1월달 데이터를 일별로 정렬해서 출력하시오.

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

```
metro <- na.omit(metro)
```

```
metro
```

```
library(dplyr)
```

```
colnames(metro)<-tolower(colnames(metro)) <-컬럼명을 소문자로 바꿔준다
```

```
metro2 <- metro[order(metro$usedate, decreasing = F),]
```

```
> metro[order(metro$usedate, decreasing = F),]
  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
```

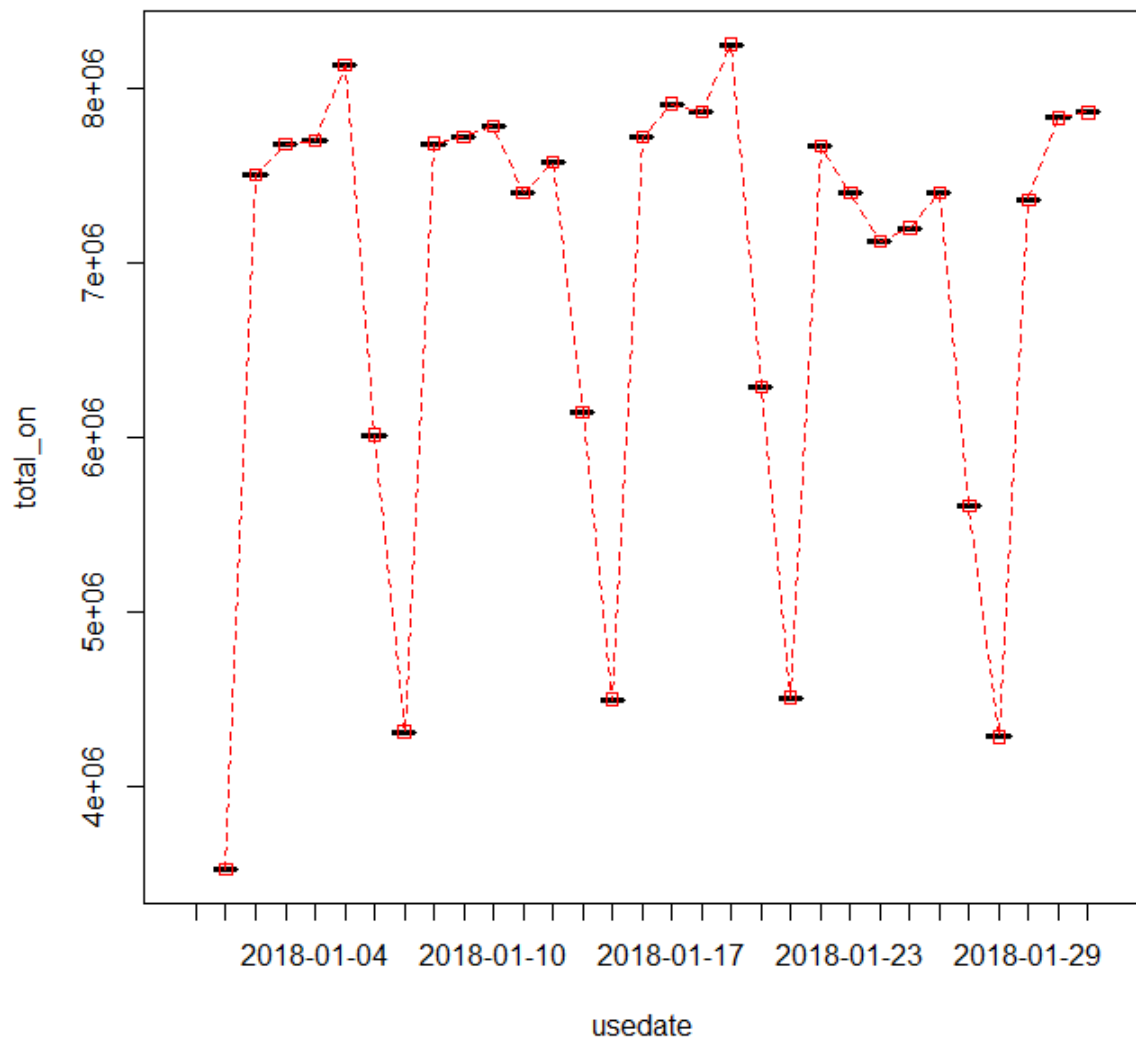
문제146. 2018년 1월달 우리나라 지하철 총승차인원수 데이터를 plot 그래프로 시각화 하시오.

```
graphics.off()
```

```
x11()
```

```
plot(metro2, type="o", col="blue")
```

```
lines(metro2, type="o", pch=22, lty=2, col="red")
```

문제147. SQL 발표 포트폴리오 데이터를 가지고 plot 그래프로 시각화 할 수 있도록 x축 y축에 해당하는 데이터를 선 별해서 변수에 저장하시오.

```

kospi <- read.csv("C:\\data\\kospi.csv", header = T)
kospi
kospi2008 <- kospi[format(as.Date(kospi$date), '%Y') == "2008", "close"]

usdkrw <- read.csv("C:\\data\\usdkrw.csv", header = T)
usdkrw2 <- tapply(usdkrw$close, format(as.Date(usdkrw$date), '%Y'), mean)

kospi2 <- tapply(kospi$close, format(as.Date(kospi$date), '%Y'), mean)

graphics.off()

kospi2_range <- range(min(usdkrw2):max(kospi2))

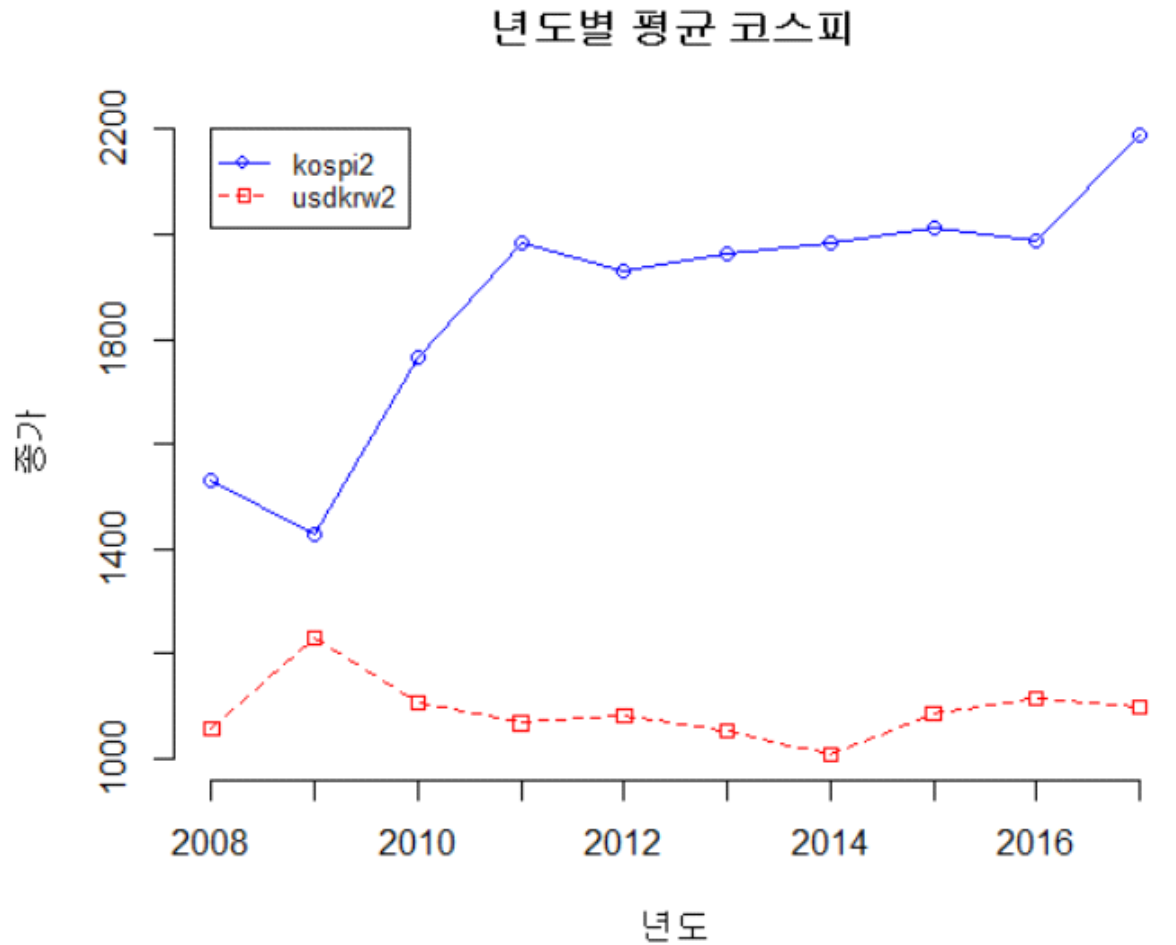
plot(kospi2, type = 'o', axes=FALSE, main = '년도별 평균 코스피', col = 'blue', ylim=kospi2_range,
      ylab = '종가', xlab = '년도')
axis(1, at=1:10,
      lab=c('2008','2009','2010','2011','2012','2013','2014','2015','2016','2017'))

```

```
axis(2)
```

```
lines(usdkrw2, type = 'o',pch=22, lty=2, col='red')
```

```
legend(1,2200,c("kospi2","usdkrw2"),col=c("blue","red"),cex=0.8,pch=21:22, lty=1:2)
```



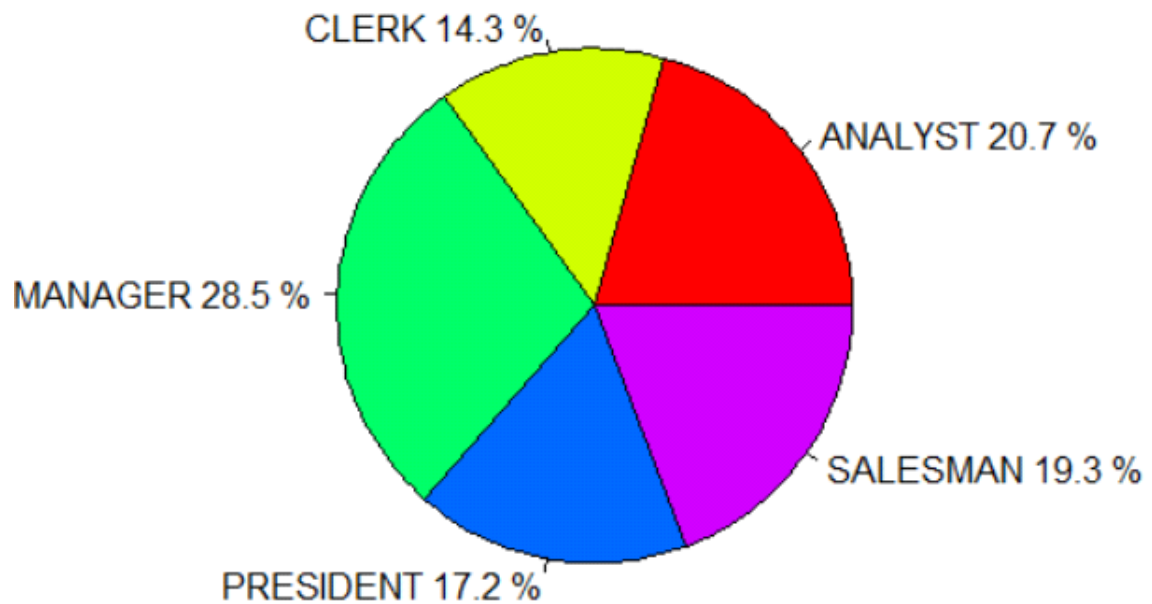
R의 막대 그래프 이론 설명

2018년 5월 17일 목요일 오후 1:47

1. 원형 그래프는 어떤 결과를 볼때 유용한지

: 데이터간의 비율을 확인할 때

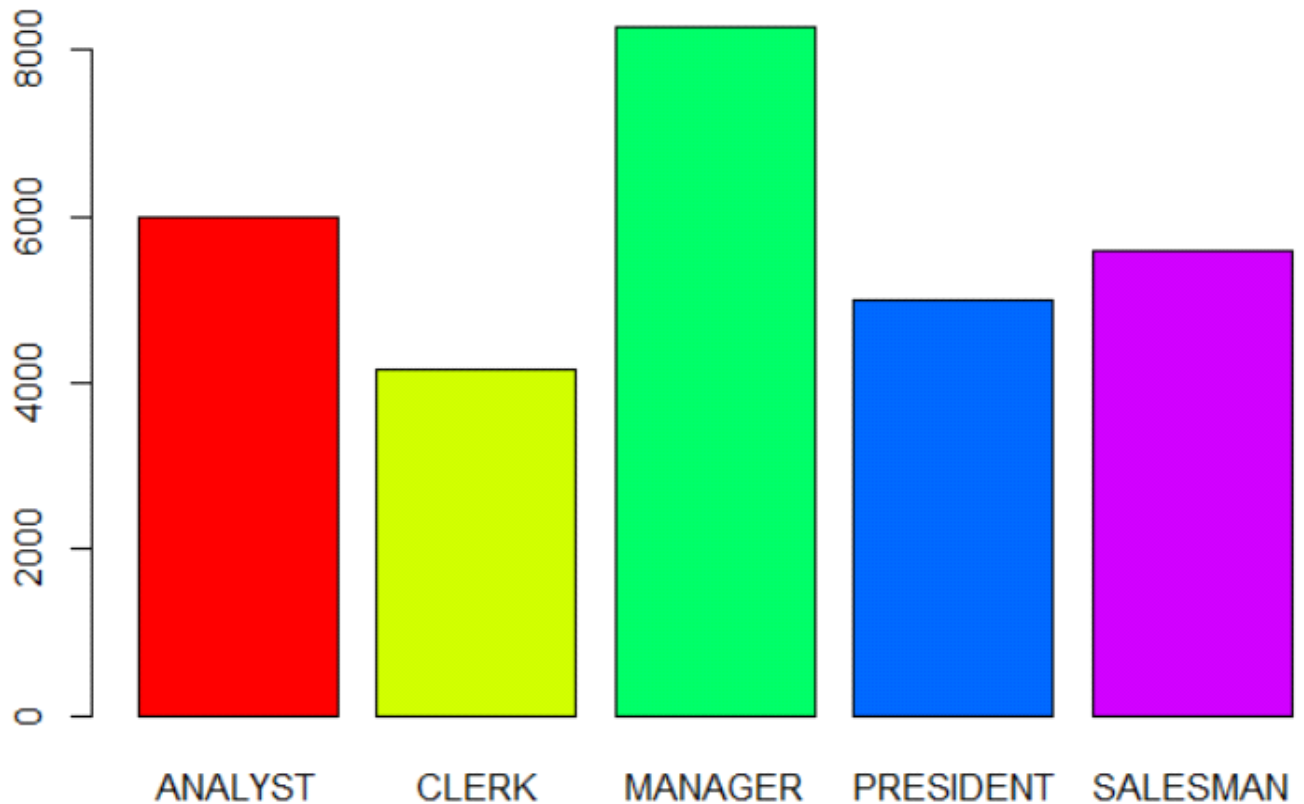
```
x<-tapply(emp$sal, emp$job,sum)
label <- round(x/sum(x)*100,1)
label2 <- paste(sort(unique(emp$job)),label, '%')
pie(x, col=rainbow(5), labels = label2)
```



2. 막대 그래프는 어떤 결과를 볼때 유용한지

: 데이터의 크기의 차이를 섬세하게 확인하고 싶을때

```
barplot(x, col = rainbow(5))
```



3. 산포도(plot) 그래프는 어떤 결과를 볼때 유용한지

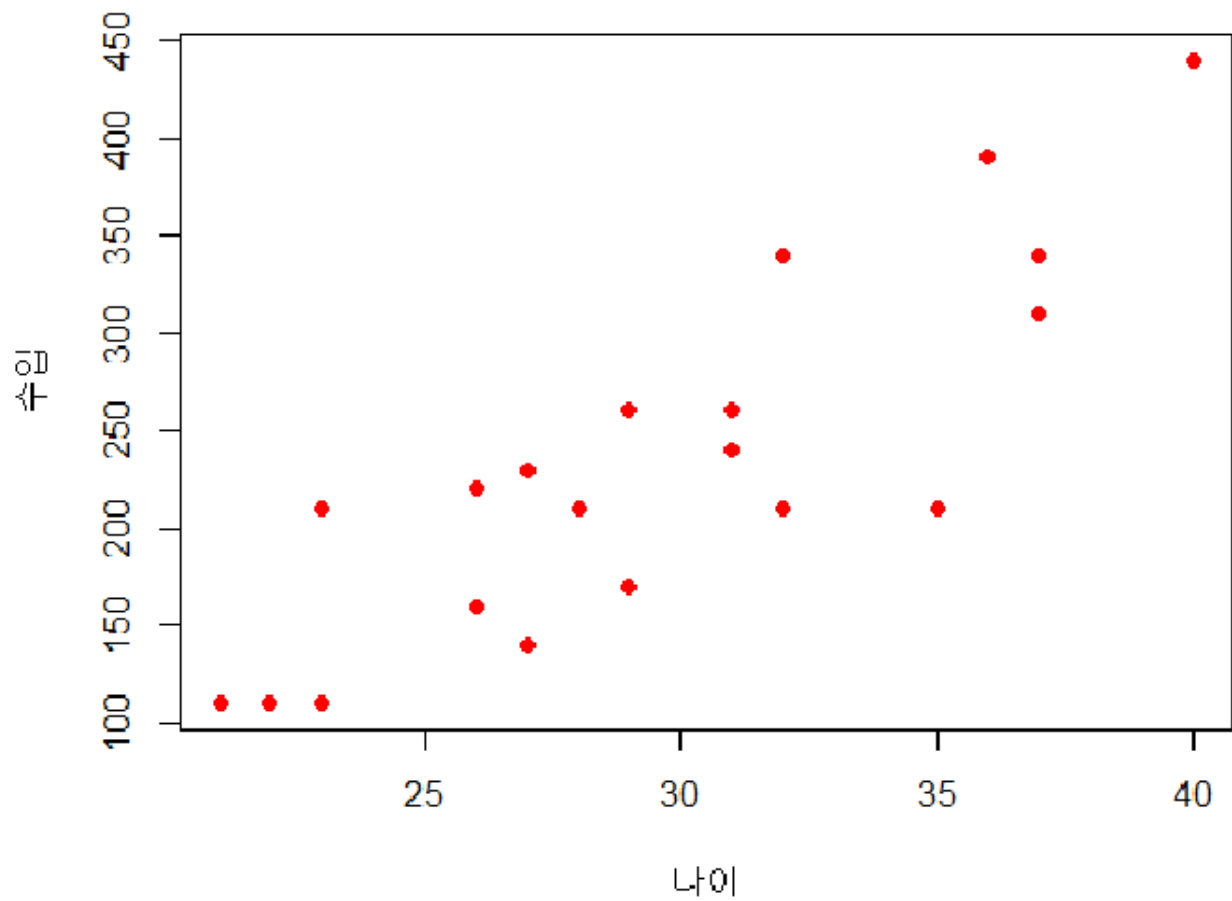
: 시간의 흐름에 따른 데이터의 변화를 확인할 때,
데이터간의 상관관계를 확인할 때

아래의 명령어로 plot 그래프를 그려보시오.

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

```
age_income
  age month_income
1  26          160
2  35          210
3  26          220
4  29          260
5  22          110
6  32          210
7  37          310
8  21          110
9  28          210
10 31          260
11 36          390
12 23          110
13 32          340
14 29          170
15 37          340
```

```
plot(age_income$age, age_income$month_income,
     xlab='나이', ylab='수입',
     col = 'red', pch = 16)
```



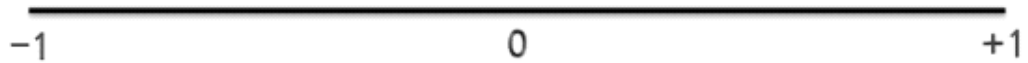
위의 그래프를 보면 나이와 소득이 상관이 있어 보이는데 그럼 상관관계가 얼마나 되는지 확인하려면?

```
cor(age_income$age, age_income$month_income)
```

```
> cor(age_inco  
[1] 0.8553024
```



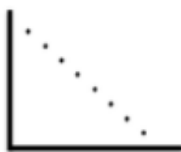
상관관계가 점점 약해진다.



음의 상관관계가
점점 강해진다.

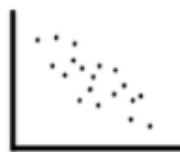


양의 상관관계가
점점 강해진다.



$r=-1$

음의 상관관계가
강하다.



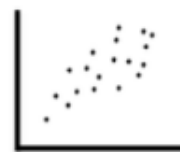
$-1 < r < 0$

음의 상관관계가
있기는 하다.



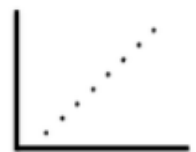
$r=0$

상관관계가 없다.



$0 < r < 1$

양의 상관관계가
있기는 하다.



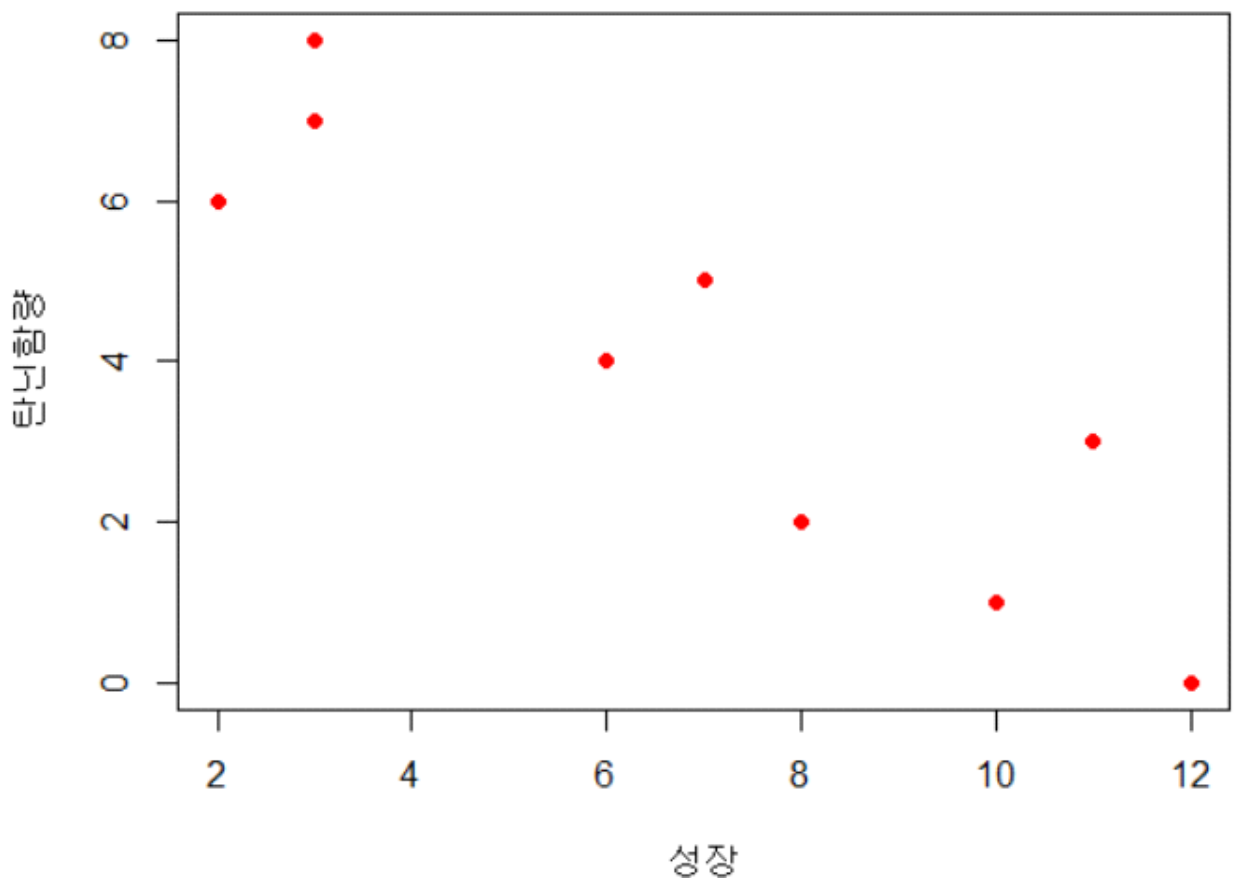
$r=+1$

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.

문제148. 카페에서 사료의 탄닌 함량 포함에 따른 애벌레 성장 추이의 관한 larva.csv를 받고 탄닌 함량과 성장률간의 상관관계가 어떻게 되는지 시각화하고, 상관계수를 구하시오.

```
larva <- read.csv("C:\\data\\larva.csv", header = T)
larva <- na.omit(larva)
plot(larva$growth, larva$tannin,
      xlab='성장',ylab='탄닌함량',
      col = 'red', pch=16)
cor(larva$growth, larva$tannin)
```

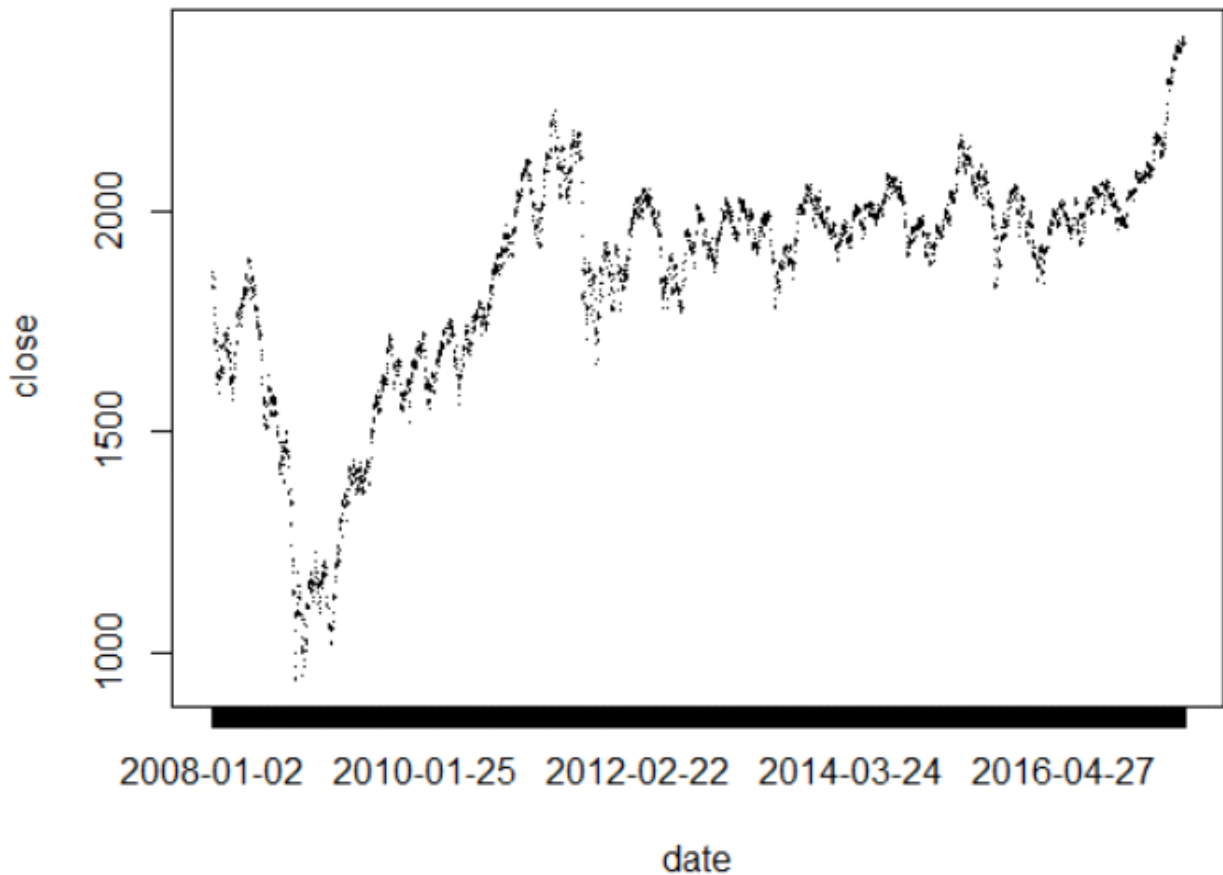


```
> cor(larva$gro
[1] -0.9031408
```

시간의 흐름에 따른 데이터의 변화를 확인하고자 할때 산포도 그래프를 확인하는데, 2008년도

부터 2017년까지 코스피 등락폭을 확인하시오.

```
plot(kospi)
```



문제149. 아래와 같이 함수를 실행하면 x축과 y축 컬럼을 각각 물어보게 하고 산포도 그래프가 그려지게 하시오.

```
plot_func()
```

x축 컬럼명을 입력하세요.~ sal

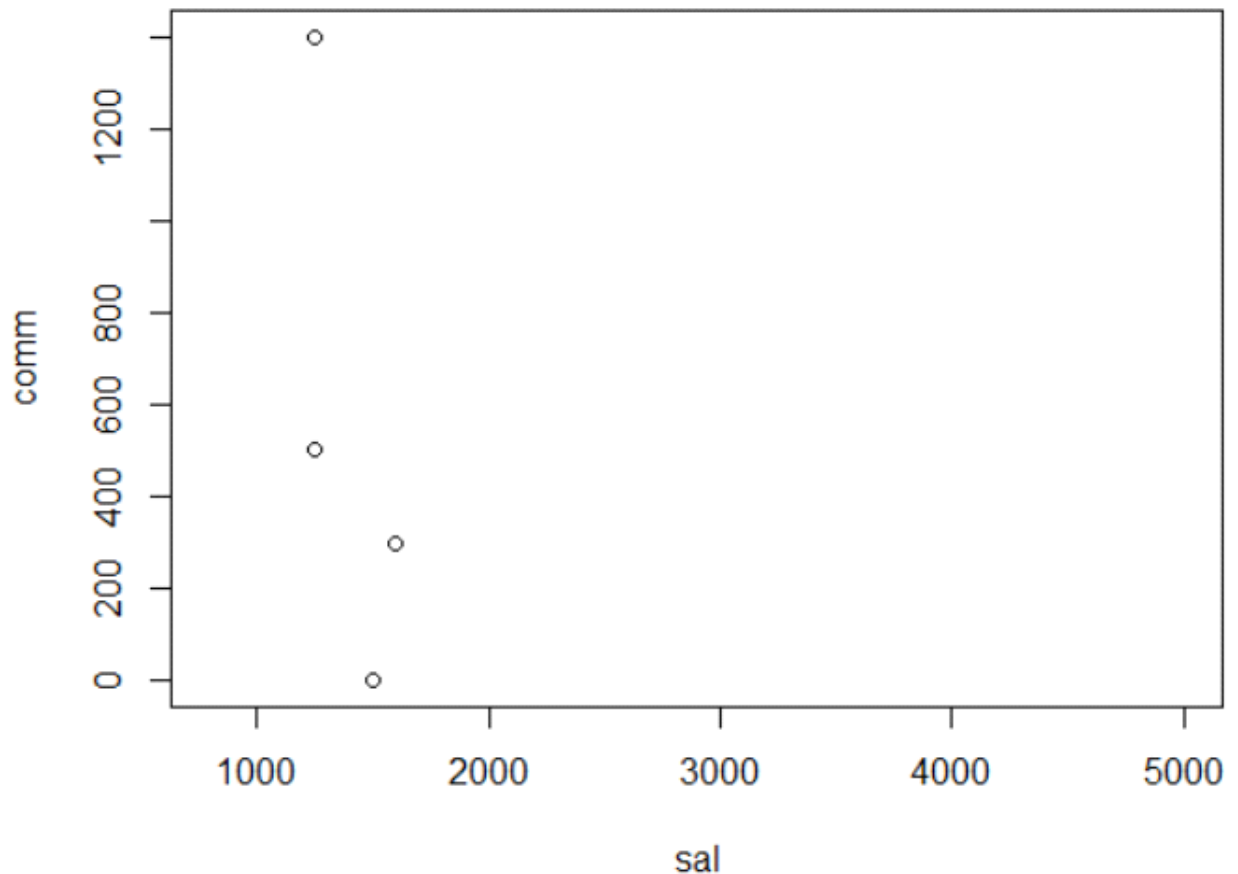
y축 컬럼명을 입력하세요.~ comm

```
plot_func <- function(){
  res1 <- readline(prompt = 'x축 컬럼명을 입력하세요~')
  res2 <- readline(prompt = 'y축 컬럼명을 입력하세요~')

  plot(emp[,res1],emp[,res2], xlab=res1, ylab=res2)
}
```

```
plot_func()
```

```
> plot_func()
x축 컬럼명을 입력하세요~sal
y축 컬럼명을 입력하세요~comm
```

문제150. 변수를 먼저 물어보게하고 변수에서 컬럼명을 추출해서 x축, y축 물어볼때 번호를 선택하게 하시오.
plot_func()

그래프 그릴 테이블을 입력하세요 ~ emp

x축 컬럼명을 입력하세요~ 6

1.empno

2.ename

.

.

y축 컬럼명을 입력하세요~ 7

1.empno

2.ename

.

.

```
plot_func <- function(){
```

```
  graphics.off()
```

```
  res1 <- readline(prompt = '그래프 그릴 테이블을 입력하세요')
```

```
  res2 <- menu(colnames(emp), title = 'x축 컬럼명을 입력하세요')
```

```
  res3 <- menu(colnames(emp), title = 'y축 컬럼명을 입력하세요')
```

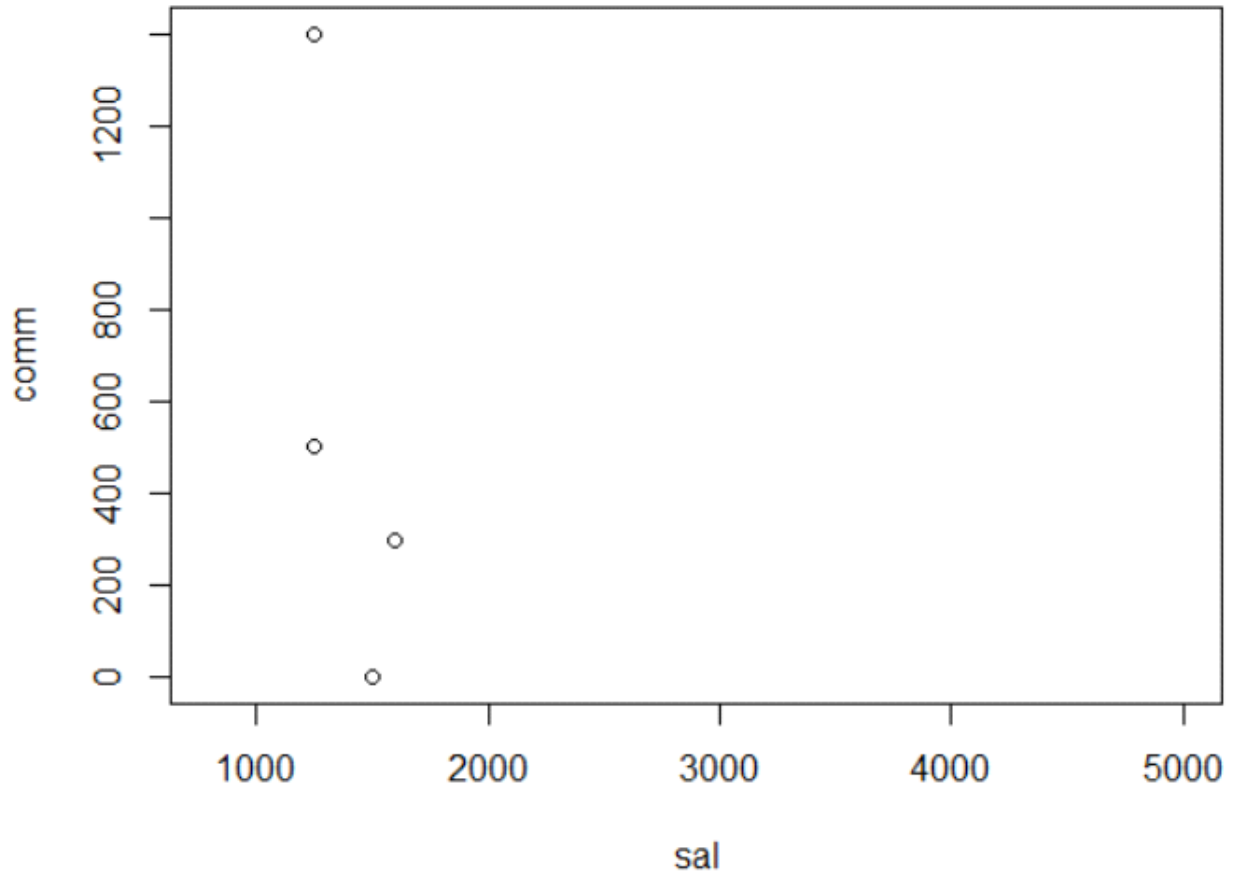
```
  res1_table <- get(res1)    ## talbe안에있는 값을 가져오려면 get을 써줘야한다
```

```
  res2 <- colnames(res1_table[res2])
```

```
  res3 <- colnames(res1_table[res3])
```

```
q <- res1_table[,c(res2,res3)]

plot(q, xlab=res2, ylab=res3)
}
plot_func()
```



문제151. 원형 그래프와 막대 그래프를 자동으로 그리는 함수를 아래와 같이 테이블을 먼저 물어보게 코드를 수정하시오.

```
data_v_f()
```

그래프 그릴 테이블을 입력하세요~ emp

그룹핑할 컬럼명을 입력하세요~ 6

1.empno

2.ename

.

토탈값을 구할 컬럼명을 입력하세요~ 5

1.empno

2.ename

.

막대 그래프를 그릴지 원형 그래프를 그릴지 선택하세요~

1. 막대 그래프

2. 원형 그래프

```

data_v_f<-function(){
  graphics.off()
  g1<-readline(prompt = '그래프 그릴 테이블을 입력하세요')

  q1<-menu(colnames(emp),title='토탈값을 구할 컬럼번호를 입력하여라')
  q2<-menu(colnames(emp),title='그룹핑할 컬럼번호를 선택하여라.')
  q3<-menu(c('막대그래프','원형그래프'),title='그래프의 종류를입력하여라')

  g1_table <- get(g1)

  m1<-colnames(g1_table[q1])
  m2<-colnames(g1_table[q2])

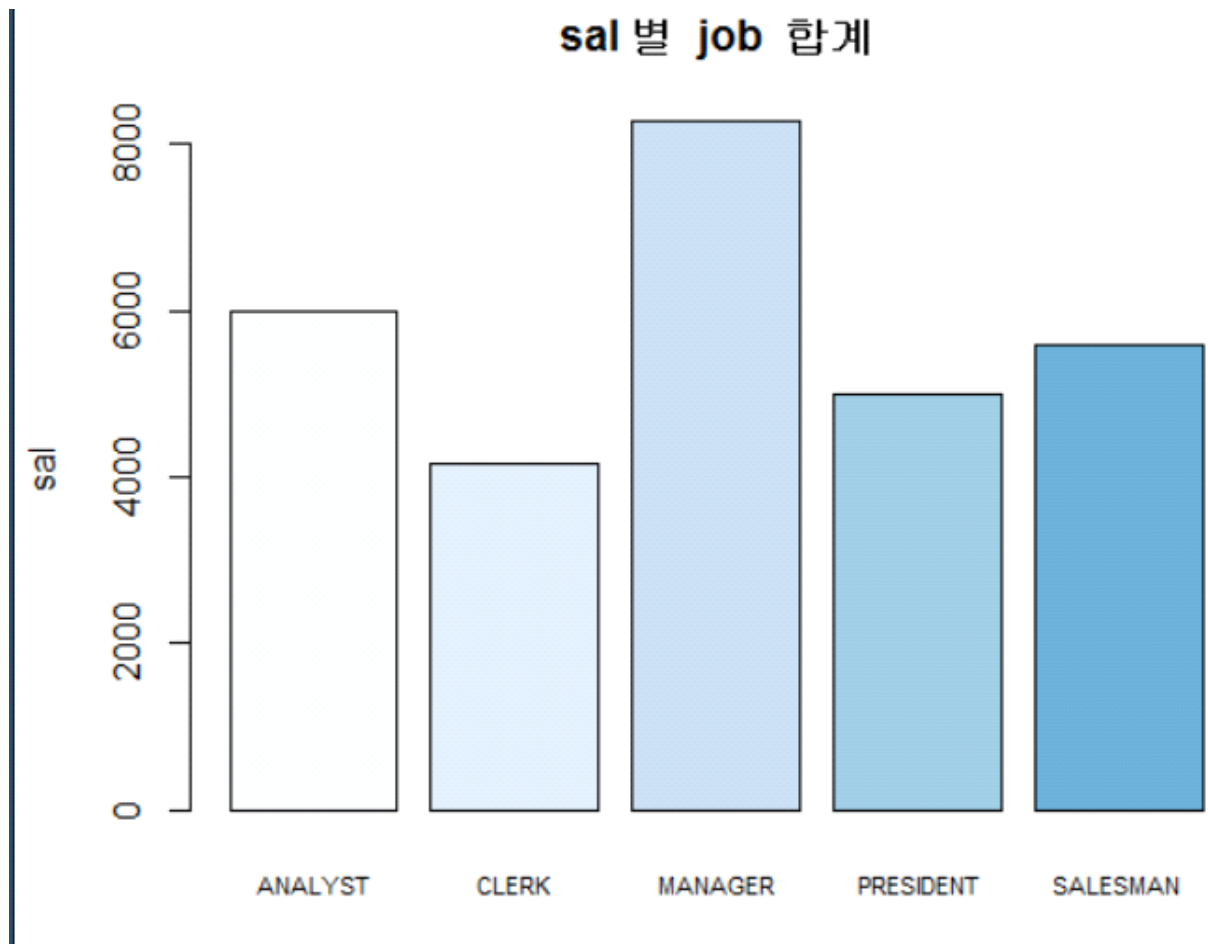
  q1<-g1_table[,q1]
  q2<-g1_table[,q2]

  x<-tapply(q1,q2,sum)

  label<-paste(sort(unique(colnames(t(x)))),
                '(',round(x/sum(x)*100,1),'%')
  mains<-paste(m1,'별 ',m2,' 합계')

  switch(q3,
        bar=barplot(x,names.arg=colnames(t(x)),col=blues9, ylab=m1,cex.names=0.7,
                     main=mains),
        piee=pie(x,col=blues9,labels=label,main=mains) )
}
data_v_f()

```



4. 구글에서 제공하는 그래프

문제152. 조인을 써서 이름을 출력하고 관리자의 이름을 출력하시오.

```
SQL> 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")]
```

	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

문제153. 문제152번 결과를 가지고 사원 테이블의 조직도를 그리시오.

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

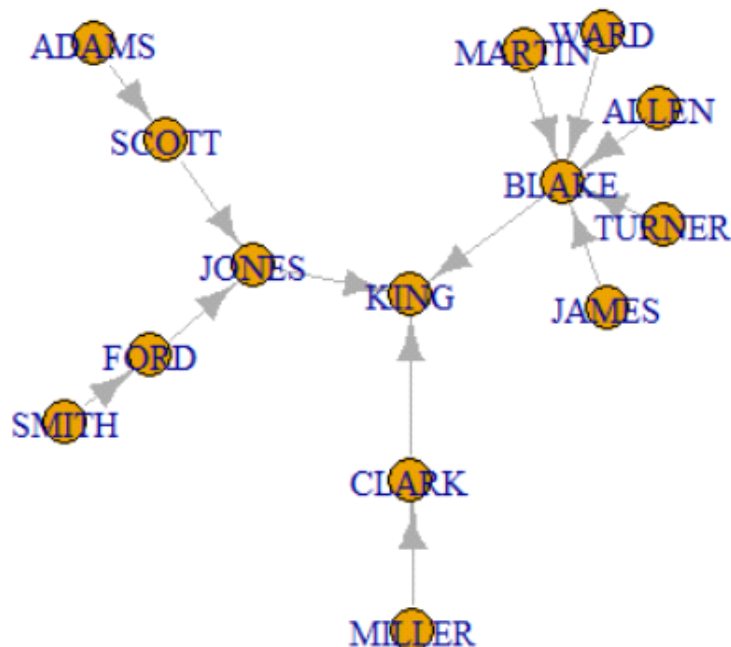
```
library(igraph)
```

```
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)
```



문제154. 문제153번 시각화 결과를 구글의 googleVis를 이용해서 emp 테이블의 관계도를 시각화 하시오.

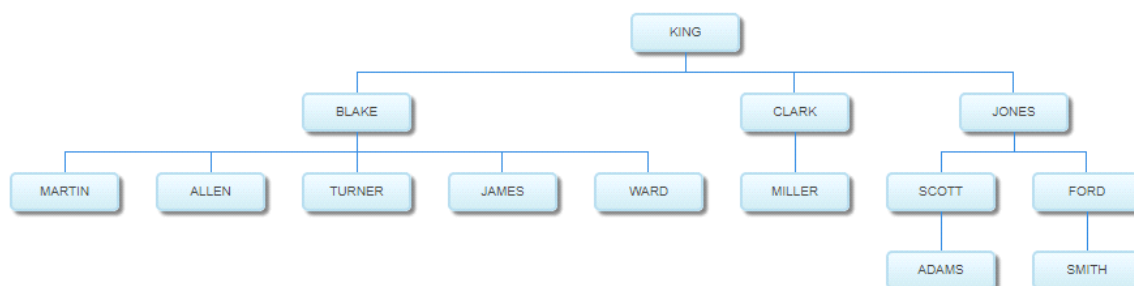
```
install.packages("googleVis")
```

```
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) ) #T를 해주면 접힌다
```

```
plot(org)
```



문제155. 아래와 같이 함수를 실행하면 바로 emp테이블의 조직도가 구글 그래프로 그려지게 하시오.

```
emp_org()
```

```

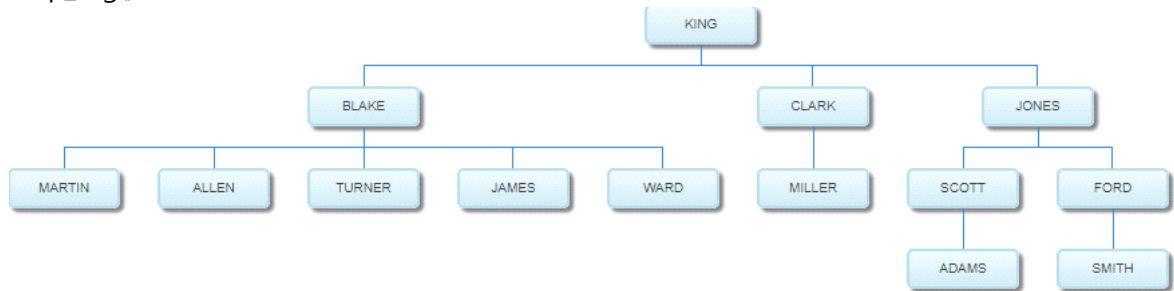
-----
emp_org <- function(){
  a <- merge(emp,emp, by.x="empno",by.y="mgr", all.y=T)

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

  plot(org)
}

emp_org()

```

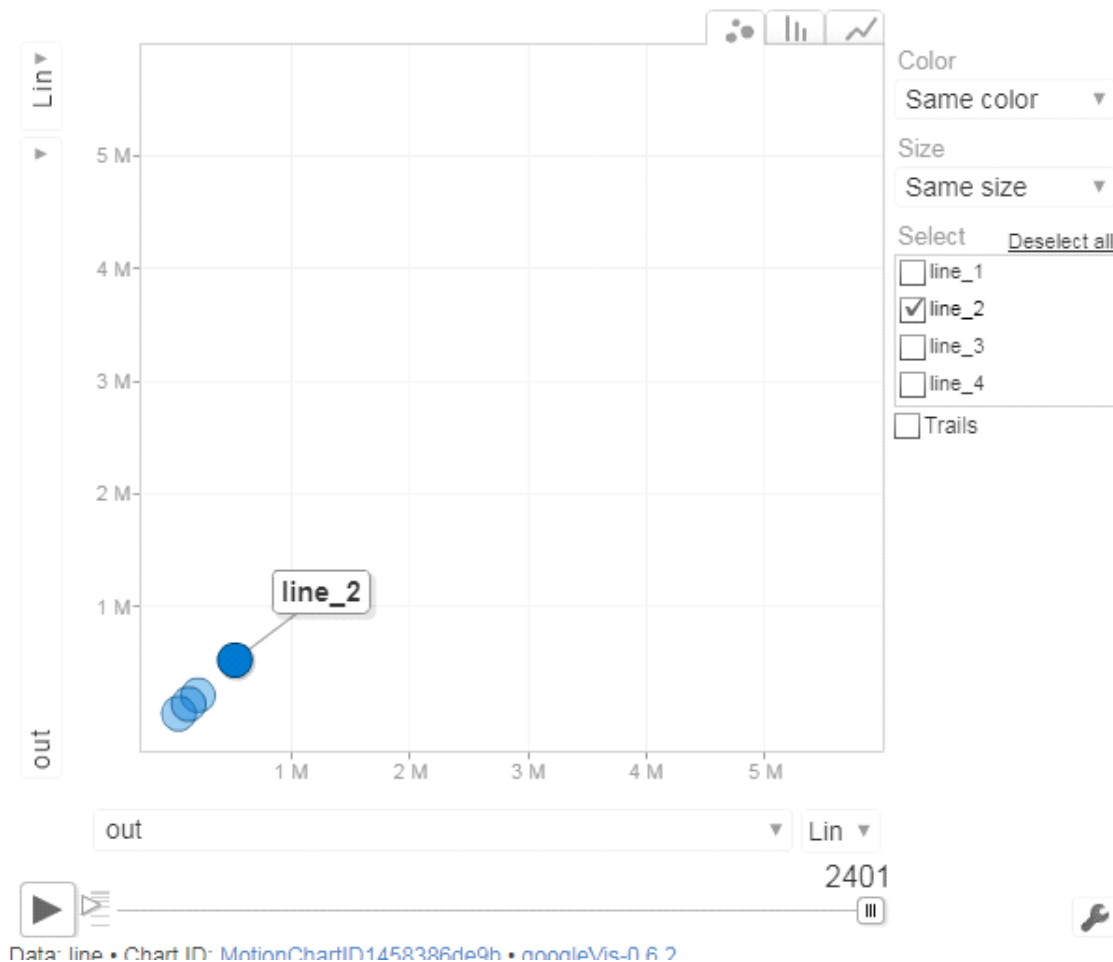


문제156. 지하철 1~4호선 승하차 승객수.csv를 R로 로드하고, line_no 컬럼과, time컬럼을 이용해서, 구글 모션차트를 그리시오.

```

line <- read.csv("C:\wwdata\ww1-4호선승하차승객수.csv", header =T)
plot(t1)

```

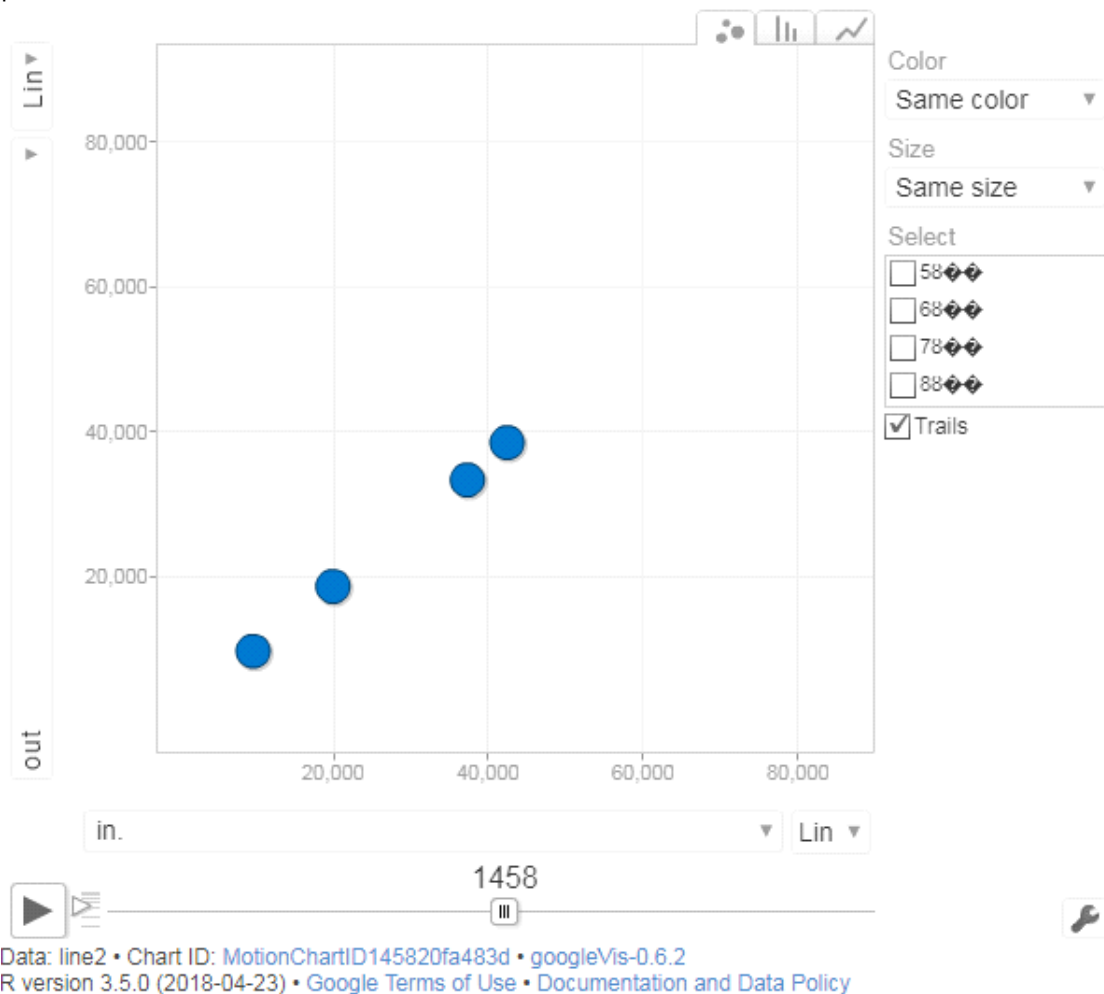


문제157. 5~8호선으로 해보시오

```
line2 <- read.csv("C:\\data\\서울지하철_5-8호선_이용현황_시간대별.csv", header = T)
```

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

```
plot(t2)
```



문제158. 부서위치를 뽑고 부서위치별 토탈월급을 세로로 출력하시오.

```
x <- merge(emp,dept, by = "deptno", all.y=T)
```

```
x
```

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

```
x2[is.na(x2)==T] <- 0
```

```
names(x2) <- c("loc","sumsal")
```

```
x2
```

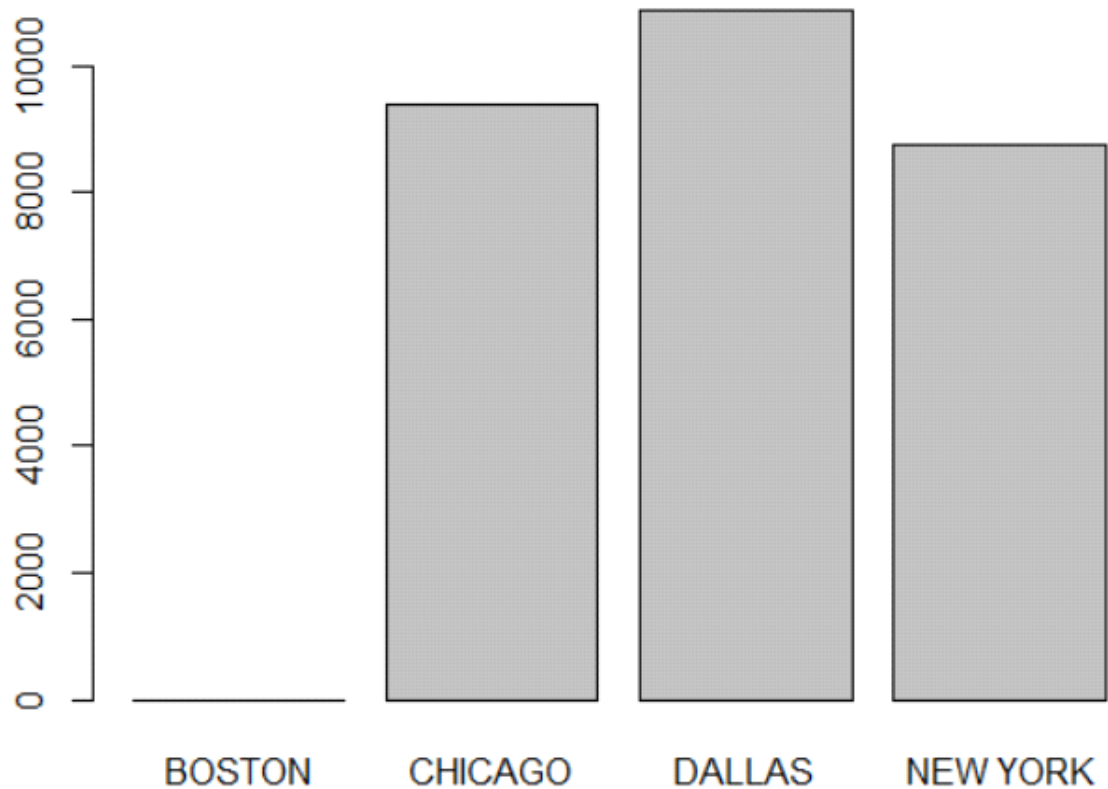
```
> x2
  loc sumsal
1 BOSTON      0
2 CHICAGO  9400
3 DALLAS  10875
4 NEW YORK  8750
```

그래프로 그릴때는 tapply로 하는게 편하다.

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

```
x3[is.na(x3)==T] <- 0
```

```
barplot(x3)
```

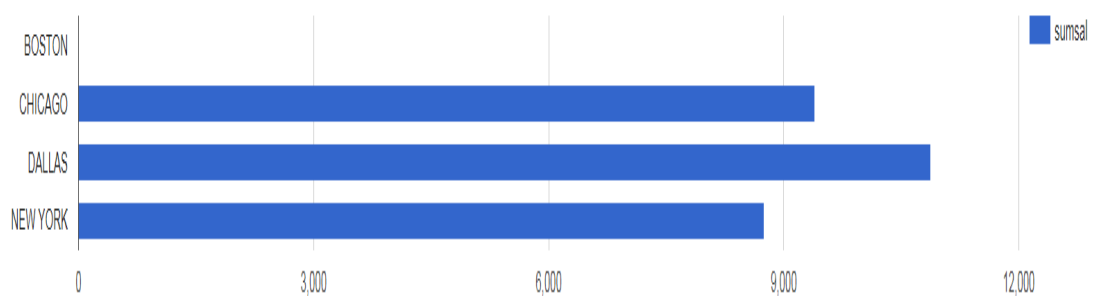


문제159. 부서위치, 부서위치별 토달월급을 구글 막대 그래프로 시각화 하시오.

```
library(googleVis)
```

```
x4<-gvisBarChart(x2)
```

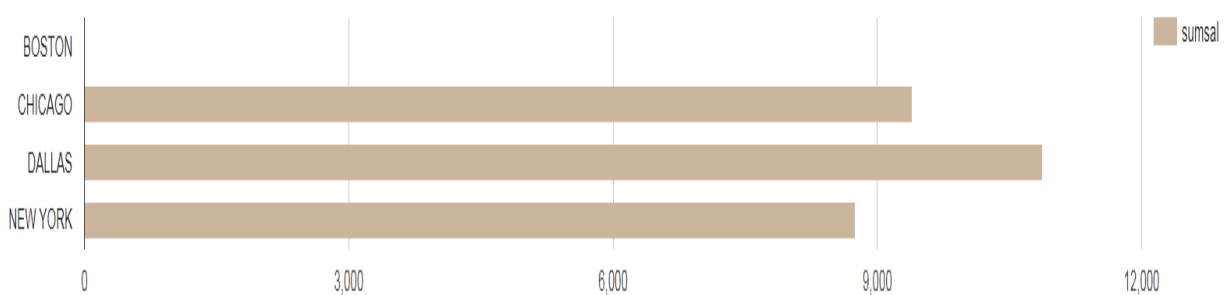
```
plot(x4)
```



문제160. 문제159번에서 만들었던 구글 막대그래프의 색깔을 변경하시오.

```
x4<-gvisBarChart(x2, option =list(colors=["#cbb69d"]) )
```

```
plot(x4)
```



5. 지도 그래프

문제161. **maps** 패키지를 설치하고 중국지도만 확대해서 출력하시오.

```
install.packages("maps")  
install.packages("mapproj")  
library(maps)  
library(mapproj)
```

```
map("world")  
map("world", "china")
```



문제162. 우리나라 지도를 출력하시오.

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



문제163. 프랑스 지도를 출력하시오.

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



문제164. 구글 지도 그래프를 이용해서 서울 지역의 지하철 2호선의 그래프를 시각화 하시오.

```
install.packages("ggplot2")
```

```
install.packages("ggmap")
```

```
library(ggplot2)
```

```
library(ggmap)
```

```
loc <- read.csv("C:\\data\\서울지하철2호선위경도정보.csv",header=T)
```

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

```
center
```

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

문제165. 구글 지도 그래프를 이용해서 서울 지역의 지하철 3호선의 그래프를 시각화 하시오.

```
loc <- read.csv("C:\\data\\서울지하철3호선위경도정보.csv",header=T)
```

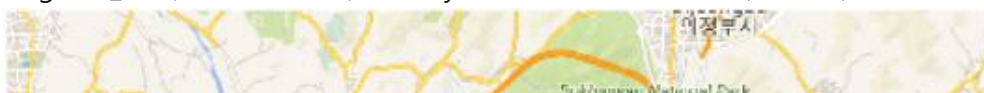
```
center <- c(mean(loc$LON), mean(loc$LAT))
```

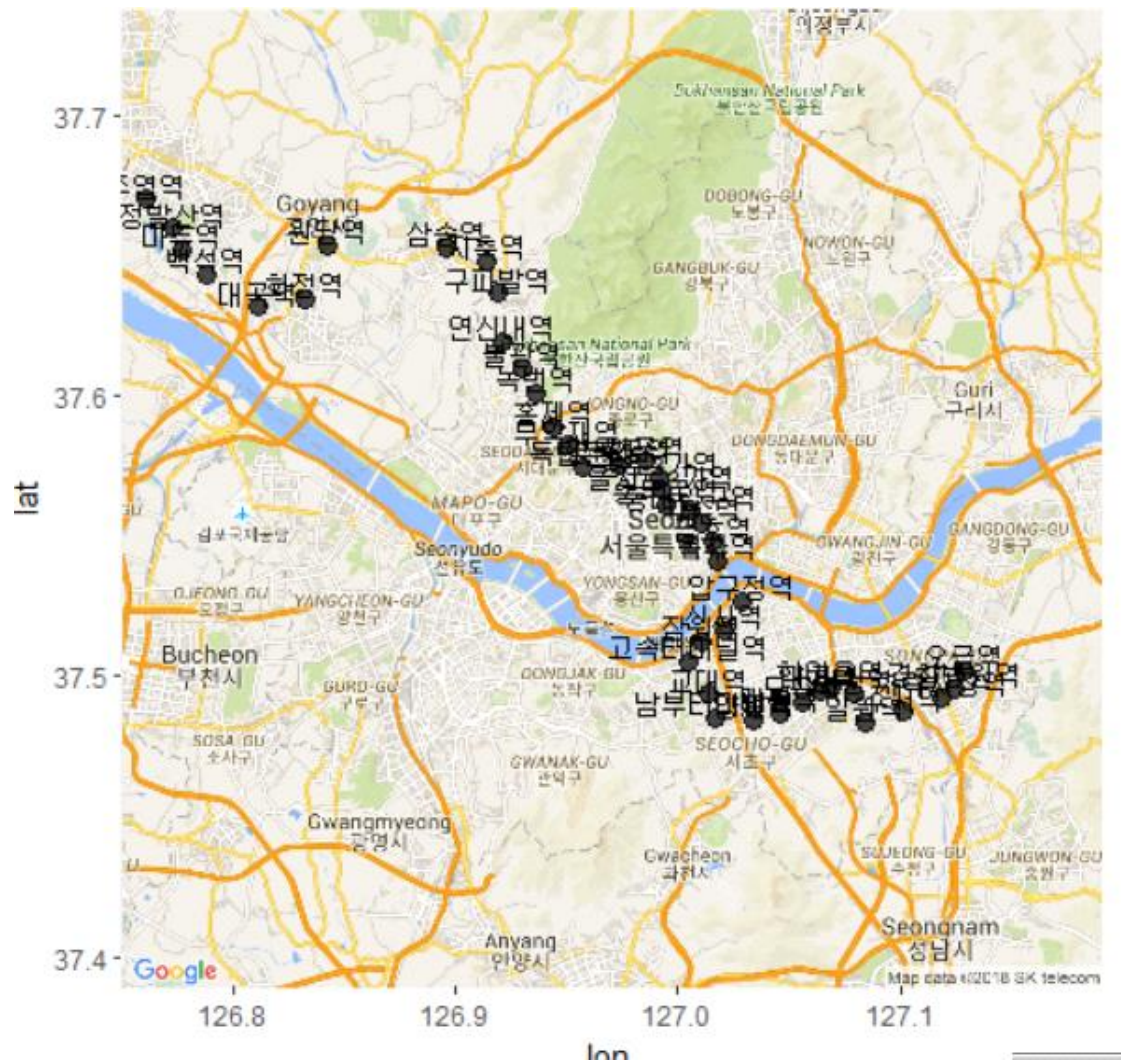
```
kor <- get_map(center, zoom=11, maptype = "roadmap")
```

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

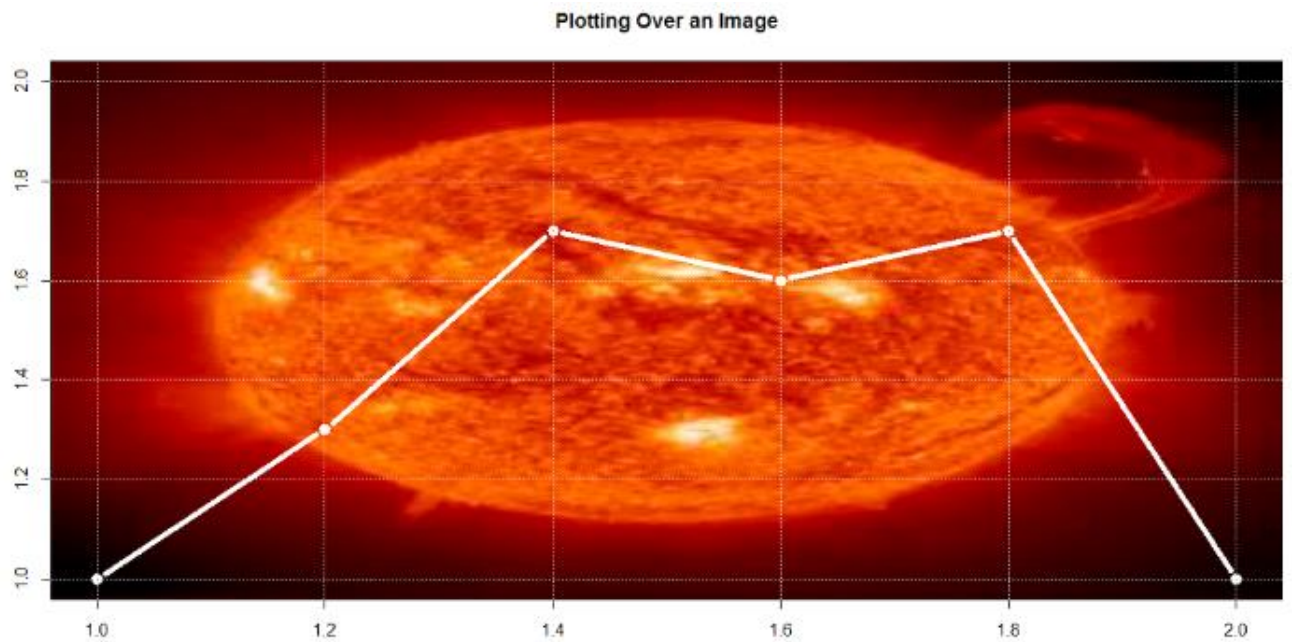
```
kor.map <- ggmap(kor) + geom_point(data=loc,aes(x=LON,y=LAT),size=3, alpha=0.7)
```

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





문제166. plot 그래프의 배경화면을 변경하시오.



```
install.packages("jpeg")
```

```
library(jpeg)
```

```
ima <- readJPEG("C:\\data\\sun.jpg")
```

```

plot(1:2, type='n', main="Plotting Over an Image", xlab="x", ylab="y")

#Get the plot information so the image will fill the plot box, and draw it

lim <- par()

rasterImage(ima, lim$usr[1], lim$usr[3], lim$usr[2], lim$usr[4])

grid()

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")

```

문제167. plot 그래프를 다시 그리는데 배경사진을 고양이 사진으로 변경해서 그리시오.

```

ima <- readJPEG("C:\\data\\wcat3.jpg")

plot(1:2, type='n', main="Plotting Over an Image", xlab="x", ylab="y")

#Get the plot information so the image will fill the plot box, and draw it

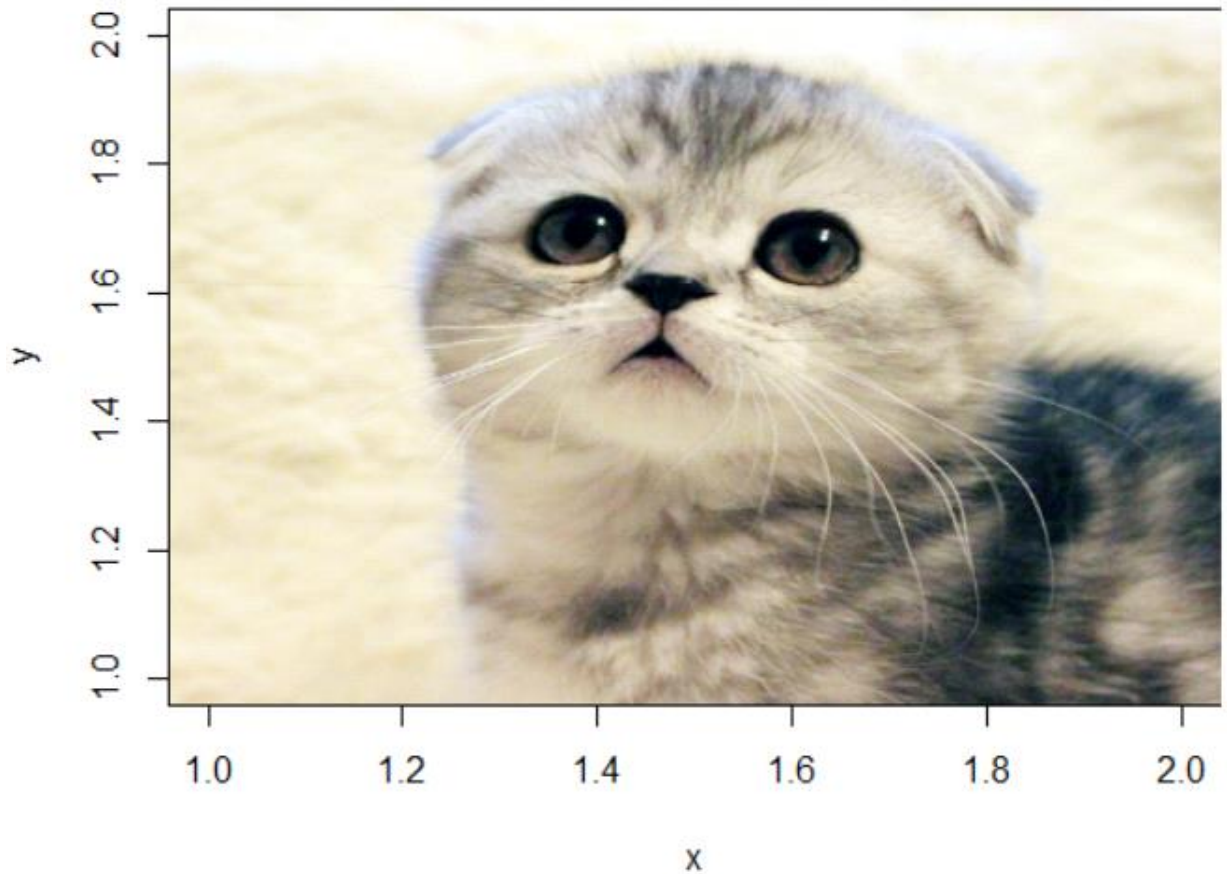
lim <- par()

rasterImage(ima, lim$usr[1], lim$usr[3], lim$usr[2], lim$usr[4])

grid()

```

Plotting Over an Image



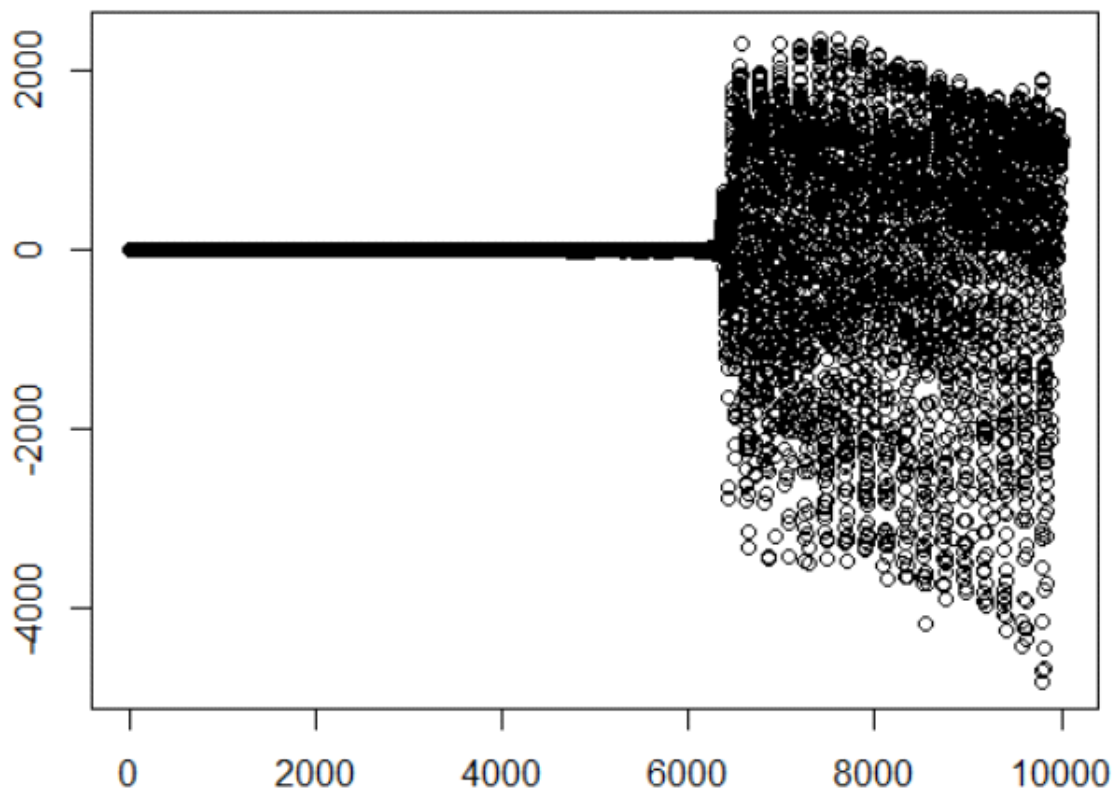
6. 소리 시각화

문제168. `output.wav`의 소리를 R로 시각화 하시오.

```
install.packages("tuneR")  
library(tuneR)  
audio <- readWave("C:\\data\\output.wav")  
play(audio)  
head(audio@left,1000)  
plot(head(audio@left,1000))
```

문제169. 원더걸스의 `so hot` 을 시각화 하시오.

```
audio <- readWave("C:\\data\\sohot.wav")  
play(audio)  
head(audio@left,1000)  
plot(head(audio@left,10000))
```

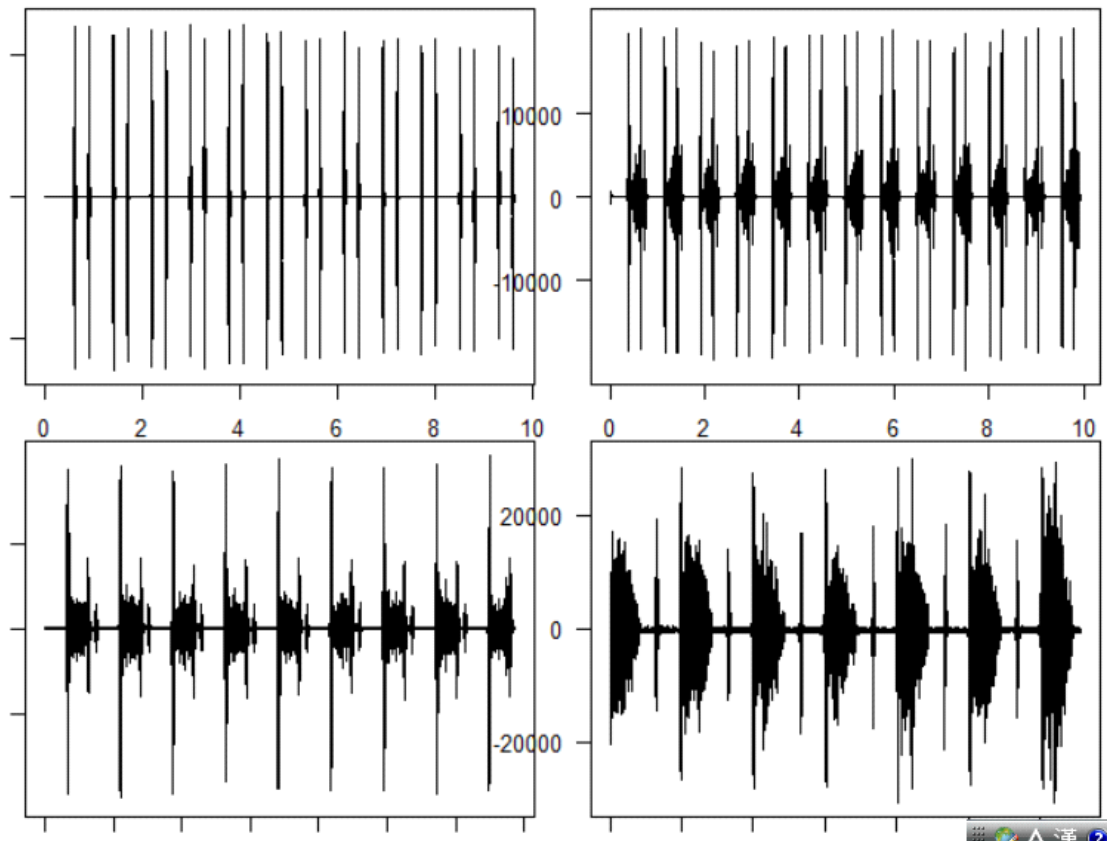
8.사분위수 그래프

문제170. 정상적인 심장박동 소리와 질병이 있는 심장박동소리를 서로 비교할 수 있도록 plot 그래프로 시각화 하시오.

```
par(mfrow=c(2,2)) #4개의 그래프를 한화면에 표시
par(mar=c(1,1,1,1)) #위,아래,좌,우 여백 사이즈
graphics.off()
```

```
audio1 <- readWave("C:\data\normal.wav")
audio2 <- readWave("C:\data\ps.wav")
audio3 <- readWave("C:\data\mr.wav")
audio4 <- readWave("C:\data\ar.wav")
```

```
plot(audio1)
plot(audio2)
plot(audio3)
plot(audio4)
```



문제171. 인증이가 올려준 테러가 일어난 지역에 대한 위도, 경도 정보를 가지고 세계지도를 바탕으로 두고 테러가 일어난 지역에 plot그래프로 점표시를 하시오.

```

terror <- read.csv('C:\ww\data\ww\terror_2015.csv')
terror <- na.omit(terror)

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

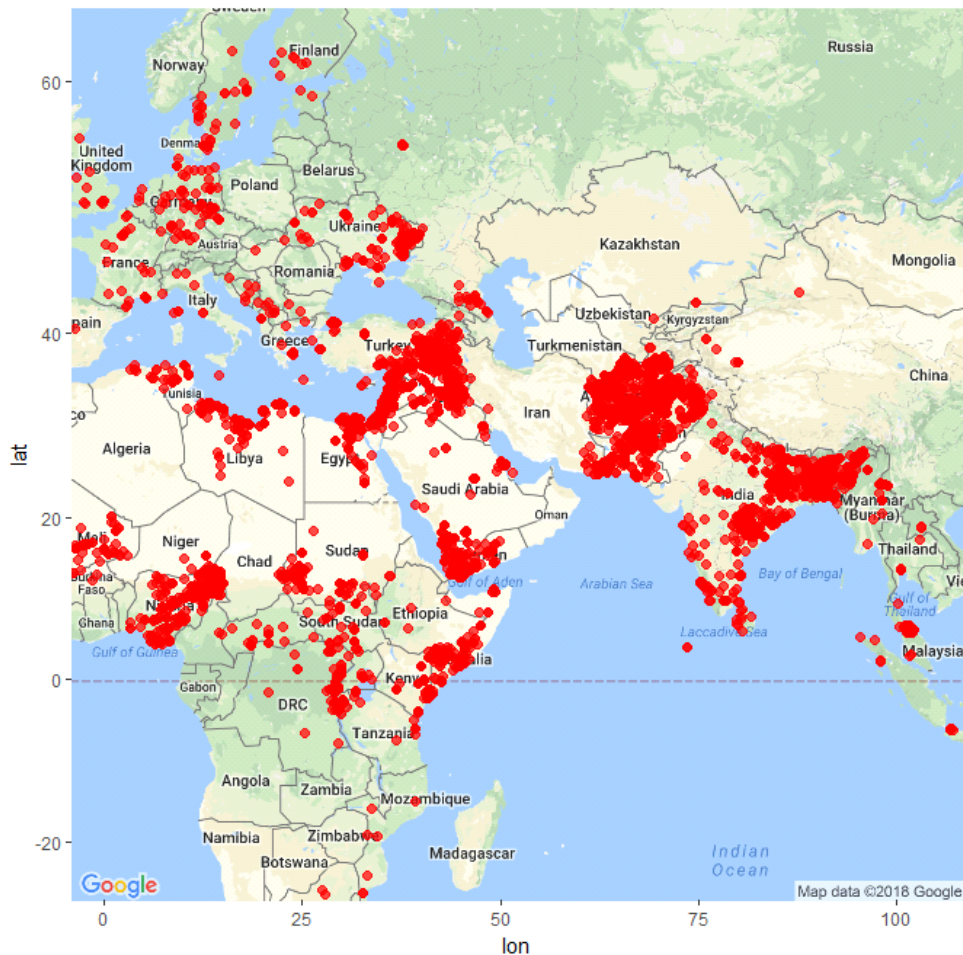
X11()
ter <- get_map(center,
               zoom = 3,      #zoom=3 부터 이 지역 대륙만 잡힌다
               maptype = 'roadmap')

ter.map <- ggmap(ter) + geom_point(data = terror
                                   ,aes(x = longitude,
                                         y = latitude),
                                   size = 2,
                                   alpha = 0.7, #채도
                                   col = 'red')

ter.map ##+ geom_text(data = terror,
                      ##    aes(x = longitude,
                      ##        y = latitude + 0.005,
                      ##        label = nkill ),

```

size = 3) 넣어주면 point 안에 nkill에대한 숫자가 나온다.



7. 워드 클라우드

문제172. 안철수 연설문을 워드 클라우드로 시각화 하시오.

```
library(KoNLP)           #한국어를 R 에서 인식할 수 있도록 설치해야하는 패키지
library(wordcloud)       #워드 클라우드 그리는 패키지
library(plyr)            #워드 클라우드 그릴때 plyr 패키지가 필요하고 나와서 설치해야함
```

```
install.packages("KoNLP")
install.packages("wordcloud")
```

```
useSejongDic()    #370957개의 한글 단어가 추가 (전희원 선생님)
```

```
mergeUserDic(data.frame(c('안철수', '박근혜', '문재인'), c('nqpc'))) # 추가 시키고 싶은
#세종 사전에 3개의 단어를 추가
```

```
nouns <- extractNoun(ahn) #연설문에서 명사만 추출
```

```
nouns <- nouns[nchar(nouns)>=2]    #철자가 2글자 이상인 것만 추출
```

```
cnouns <- count(nouns)
```



```
pal <- brewer.pal(6,"Dark2") #DARK2 라는 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 중앙에서부터 퍼지게한다.
```



문제173. 안철수 연설문의 단어와 건수를 출력하는데 건수가 높은것부터 나오게하시오.

```
head(orderBy(~-freq, cnouns),10)
```

	x	freq
41	국민	20
192	정치	12
106	생각	11
80	미래	9
78	문제	6
100	사람	6
161	우리	6
193	정치가	6
20	경제	5
63	대통령	5

문제174. 영화 겨울왕국 대본을 워드클라우드로 시각화 하시오.

```
useSejongDic()
```

```
winter <- readLines("C:\\\\data\\\\winter.txt")
```

```
nouns <- extractNoun(winter)
```



```
nouns <- nouns[nchar(nouns)>=2]

cnouns <- count(nouns)

pal <- brewer.pal(6,"Dark2")
pal <- pal[-(1)]

windowsFonts(malgun=windowsFont("맑은 고딕"))

wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
  random.order=F, family="malgun")
```



문제175-2. 동윤이의 유튜브 포트폴리오를 워드클라우드로 시각화 하시오.

```
useSejongDic()
youtube <- readLines("C:\\data\\YOUTUBE_18y.txt")

nouns <- extractNoun(youtube)

nouns <- unlist(nouns)

nouns <- nouns[nchar(nouns)>=2]

cnouns <- count(nouns)

pal <- brewer.pal(6,"Dark2")
pal <- pal[-(1)]
```

```
wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
          random.order=F, family="malgun")
graphics.off()
```



문제176. 텍스트 파일을 물어보게하고 텍스트 파일명을 입력하면 자동으로 워드 클라우드가 그려지는 함수를 생성하시오.

wordcloud()

워드 클라우드를 그릴 원본 텍스트명을 입력하세요~ winter.txt

```
wordcloudd <- function(){
```

```
library(KoNLP)
```

```
library(wordcloud)
```

```
library(plyr)
```

```
useSejongDic()
```

```
graphics.off()
```

```

setwd("C:\\data") #가져올 폴더를 지정

res1 <- readline(prompt = '워드클라우드 그릴 원본 텍스트명을 입력하세요~')

winter <- readLines(res1)

nouns <- extractNoun(winter)

nouns <- unlist(nouns)

nouns <- nouns[nchar(nouns)>=2]

cnouns <- count(nouns)

pal <- brewer.pal(6,"Dark2")
pal <- pal[-(1)]

windowsFonts(malgun=windowsFont("맑은 고딕"))

wordcloud(words=cnouns$x, freq=cnouns$freq, colors=pal, min.freq=3,
           random.order=F, family="malgun")

}
wordcloudd()

```

```
워드클라우드 그릴 원본 텍스트명을 입력하세요~winter.txt
```



```

}

#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())
}
graph_func()

```

8. 사분위수 그래프

• 평균 값이란?

왜 통계학에서는 average라고 안하고 mean이라고 할까?

- 평균 값을 구하는데는 여러가지 방법이 있기 때문이다.

ex) 우리반의 학생들의 나이의 평균 값을 구한다고 했을 때 지금은 사람의 수가 몇 명인지 알고 있지만 새로운 누군가가 들어오면 다시 계산해야 하는 번거로움이 생긴다. 이걸 피하기 위한 방법이 무엇?

답) 통계학자들은 숫자를 문자로 표현함으로써 해결하였다.

$x_1 + x_2 + x_3 + x_4 + \dots + x_n$

이를 표현하는 간단한 방법은? $\sum x$

그럼 평균을 구하는 것을 문자로 나타내면?

$\sum x/n = \mu$

문제177. 아래의 나이의 평균을 R로 구하시오.

보기)

나이 19 20 21

도수 1 3 1

방법1)

```
x <- c(19,20,20,20,21)
```

```
table(x)
```

```
mean(x)
```

방법2)

```
x<-c(19, rep(20,3),21)
```

```
mean(x)
```

```
> mean(x)
[1] 20
```

문제178. 아래의 나이의 평균을 R로 구하시오.

보기)

나이 19 20 21 145 147

도수 3 6 3 1 1

```
x <- c(rep(19,3),rep(20,6),rep(21,3),145,147)
```

```
mean(x)
```

```
> mean(x)
[1] 38
```

이상치 때문에 평균 나이가 올라갔다.

문제179. 문제178번의 이상치를 구하시오.

```
install.packages("outliers")
```

```
library(outliers)
```

```
outlier(x)
```

```
> outlier(x)
[1] 147
```

어떤 여학생(25살)이 나랑 비슷한 나이대인 스포츠 센터 교실에 등록하려고 했는데, 평균을 보니 38살이어서 등록을 안하게 되었다. 대부분의 데이터는 20대인데 이상치 때문에 평균값이 올라갔다. 이를 해결하기 위한 방법이 무엇이나?

즉, 평균 값 말고 다른 데이터를 알아야 하는데 그것이 무엇인가?

중앙 값(median)

```
median(x)
```

```
> median(x)
[1] 20
```

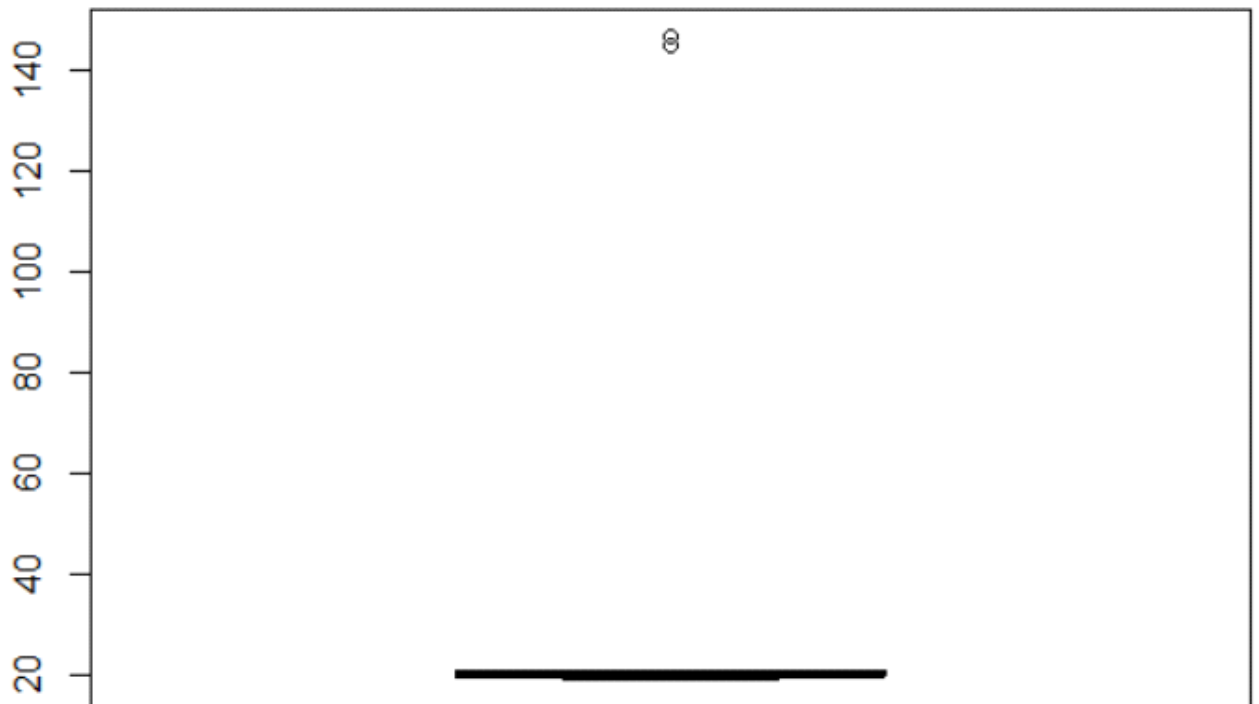
문제180. x에있는 데이터를 boxplot 그래프로 시각화 하시오.

```
a<-boxplot(x)
```

x

```
> a<-boxplot(x)
> x
[1] 19 19 19 20 20 20 20 20 20 21 21 21 145 147
```

boxplot(x)

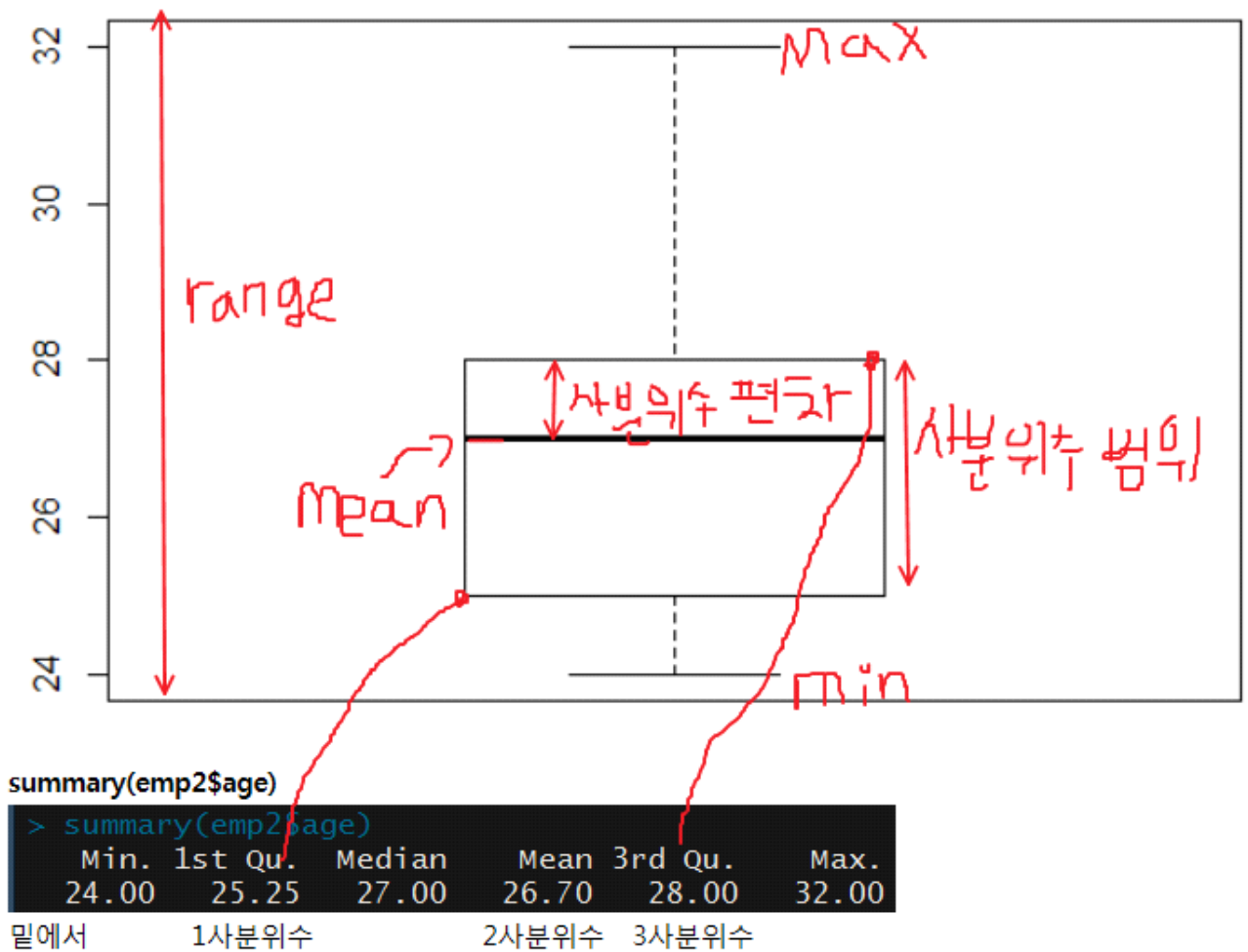


문제181. 우리반 나이 데이터를 가지고 박스 그래프(사분위수 그래프를 그리시오).

```
emp2 <- read.csv("C:\\data\\emp2.csv", header = T)
```

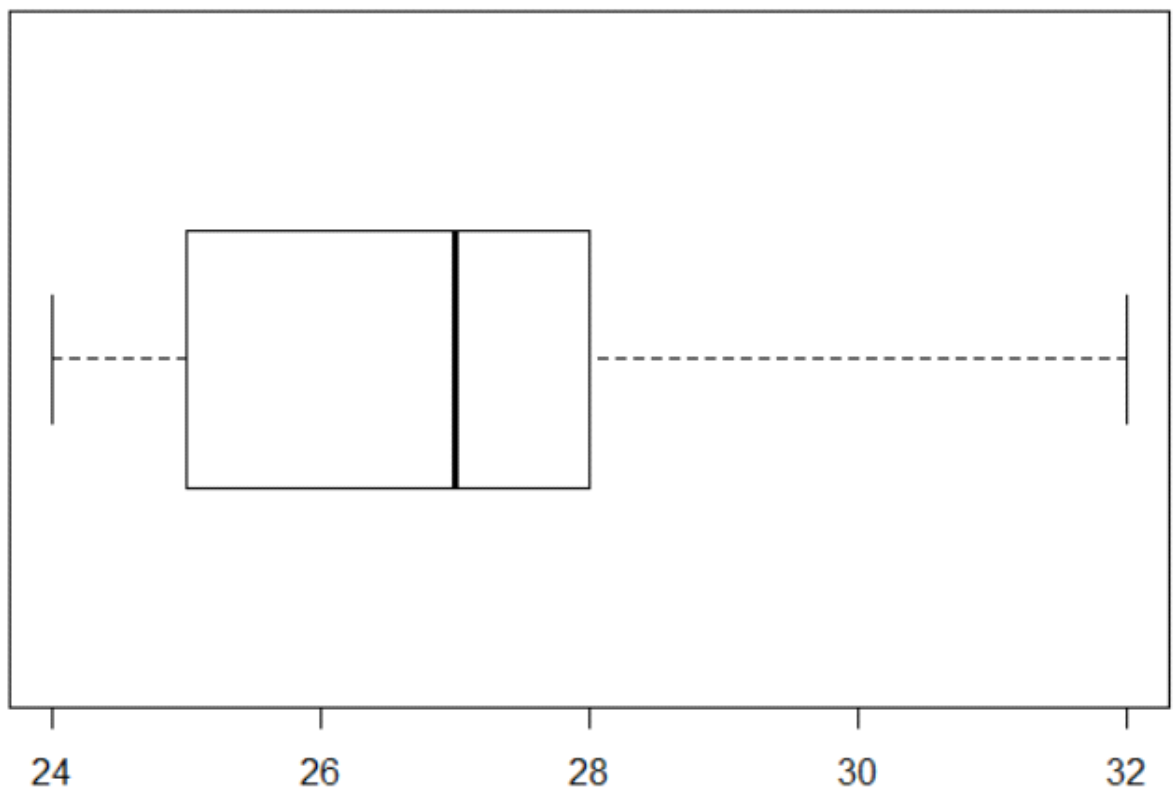
```
boxplot(emp2$age)
```

```
summary(emp2$age)
```



문제182. 방금 그린 사분위수 그래프를 옆으로 그려지게 하시오.

`boxplot(emp2$age, horizontal = T)`



문제183. 사분위수 그래프를 자동화 그래프 함수에 5번째로 추가하시오.

5번을 누르고 나면
테이블명을 입력하세요~ emp
컬럼명을 입력하세요~ sal

```
-----  
#사분위수 그래프  
boxplot_func<-function(){  
  x<-readline(prompt='테이블명을 입력하세요')  
  table_in<-get(x)  
  
  y<-menu(colnames(table_in),title='컬럼명을 입력하세요')  
  
  x2 <-colnames(table_in)[y]  
  
  boxplot(table_in[,x2], horizontal = T, col='green')  
}  
  
switch(q3,bbb(),bbb(),plotfunc(),wordcloud7(),boxplot_func() )  
  
}
```

graph_func()

문제184. 반장형이 구현해온 웹브라우저에서 R shiny로 그래프를 띄우는 것을 구현하시오.

IF 문의 문법

2018년 5월 21일 월요일 오후 2:16

구조)

```
if(조건식) {  
    조건식이 true일때 실행되는 문장 }  
  
else if (조건식) {  
    조건식이 true일때 실행되는 실행문  
}  
else {  
    위의 조건식들에 만족하지 않는 경우 실행되는 실행문  
}
```

문제185. 이름을 물어보게하고 이름을 입력하면 해당 사원이 고소득자인지 일반 소득자인지 저소득자인지

출력되는 함수를 생성하시오.

```
income()  
이름을 입력하세요~ SCOTT  
고소득자 입니다.  
-----기준  
월급 3000 >= 고소득자  
3000 > 월급 >= 2000 일반 소득자  
월급 > 2000 저소득자
```

답)

```
income <- function(){  
  name <- readline(prompt = '이름 입력')  
  salary <- emp[emp$ename==toupper(name),"sal"]  
  if (salary >= 3000)  
  {print("고소득자 입니다")}  
  else if (salary >= 2000)  
  { print("일반 소득자 입니다")}  
  else { print("저소득자 입니다")}  
}
```

문제186. 이름을 물어보게 하고 이름을 입력해서 해당 사원의 커미션이 null이면 '보너스 대상입니다'

라는 메시지가 출력되게하고 null이 아니면 '보너스 대상자가 아닙니다' 라는 메시지가 출력되게 하시오.

```
ename <- function() {  
  name <- readline(prompt = '이름 입력')  
  comm <- emp[emp$ename==toupper(name), "comm"]  
  if (is.na(comm)){print("보너스 대상입니다")}  
  else{print("보너스 대상이 아닙니다.")}  
}  
ename()
```

• for loop 문의 문법

```
for (루프변수 in 리스트) {  
    반복할 문장  
}  
ex )  
aaa <- function(x) {  
  for (i in 1:x) {  
    print(i)  
  }  
}  
aaa(10)
```

```
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5  
[1] 6  
[1] 7  
[1] 8  
[1] 9  
[1] 10
```

문제187. 아래와 같이 함수를 실행하면 ★가 출력되도록 하시오.

```
func187(6)  
★  
★★  
★★★  
★★★★  
★★★★★  
★★★★★★  
★★★★★★★  
-----
```

문제188. 아래와 팩토리얼 함수를 생성하시오.

```
func188(5)
120
-----
func188 <- function(x) {
  fac <- 1
  for (i in 1:x) {
    fac <- fac*i
  }
  print(fac)
}
func188(5)
```

```
> func188(5)
[1] 120
```

문제189. 문제188번의 팩토리얼 구하는 함수를 디버깅 패키지를 사용해서 디버깅하시오.

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

debug(func188 <- function(x) {
  fac <- 1
  for (i in 1:x) {
    fac <- fac*i
  }
  print(fac)
})
func188(5)
```

```
Browse[2]>
debug at #4: fac <- fac * i
Browse[2]>
debug at #4: fac <- fac * i
Browse[2]>
debug at #4: fac <- fac * i
Browse[2]>
debug at #4: fac <- fac * i
Browse[2]>
debug at #4: fac <- fac * i
```

- while loop 문의 문법

while (조건식) { #조건식이 true이면 실행된다

반복할 실행문

```
}
```

```
ex) x <- 0
     while(x <= 5) {
       x <- x + 1
     }
```

```
      x
> x
[1] 6
```

문제190. 10을 입력하면 55가 출력되게하는 함수를 while loop문으로 생성하시오.

```
func190(10)
```

```
55
```

```
-----
func190 <- function(x){
```

```
  y <- 0
```

```
  while (x>0){
```

```
    y <- y + x
```

```
    x <- x - 1
```

```
  }
```

```
  print(y)
```

```
}
```

```
> func190(10)
[1] 55
```

• break 문

: break 문을 사용하면 for loop문이나 while loop문 반복문 실행 도중에 루프문에서 벗어날 수 있다.

```
ex)
```

```
x <- 1:5
```

```
for(i in x){
```

```
  if(i == 3){
```

```
    break
```

```
  }
```

```
  print(i)
```

```
}
```

```
[1] 1
[1] 2
```

문제191. 아래의 power 함수를 생성하시오.(^쓰지말고 while loop와 break로)

```
보기>
```

```
power(2,4)
```



```

-----
powerr <- function(x,y){
  result <- x
  cnt <- 1
  while(TRUE) {
    if(cnt == y) break
    result <- x*result
    cnt <- cnt +1
  }
  print(result)
}
powerr(2,4)
> powerr(2,4)
[1] 16

```

문제192. 아래와 같이 log7 이라는 로그 함수를 구현하시오.(while loop문과 break 문으로 구현하시오.)

log7(2,16)

결과는 4입니다.

```

-----
log7 <- function(x,y){
  cnt = 1
  while(TRUE){
    y=y/x
    cnt=cnt+1
    if(y==x) {
      break
    }
  }
  print(cnt)
}
> log7(2,16)
[1] 4

```

문제193. 24와 18의 최대 공약수를 구하시오.

최대 공약수를 알아야하는 이유? 약분을 빨리 하기 위해서

빵 24개를 18명이 똑같이 나눠 먹으려면 어떻게 배분하면 되는거?

빵 4개를 3명에게 나눠 주려면 한 사람당 몇개씩?

```

-----
max_gong_yaksu(24,18)

```

최대공약수는 6입니다.

```

max_gong_yacksoo <- function(x,y){
  cnt <- 0

  if(x!=0 && y!=0 && x>y){ cnt <- x
    if(x!=0 && y!=0){
      for(i in cnt:0){
        if(x%%cnt==0 && y%%cnt == 0){ break }
        cnt <- cnt - 1
      }
      print(paste("최대공약수는",cnt,"입니다~"))
    }
  }

  else if(x!=0 && y!=0 && y>x){cnt <- y
    if(x!=0 && y!=0){
      for(i in cnt:0){
        if(x%%cnt==0 && y%%cnt==0 ) { break }
        cnt <- cnt - 1
      }
      print(paste("최대공약수는",cnt,"입니다~"))
    }
  }

  else if(x==y){
    print(paste("최대공약수는",x,"입니다~"))
  }

  else if(x == 0 || y == 0){
    print("최대 공약수는 0 입니다~")
  }

  else{
    print("최대공약수는 1 입니다~")
  }
}

```

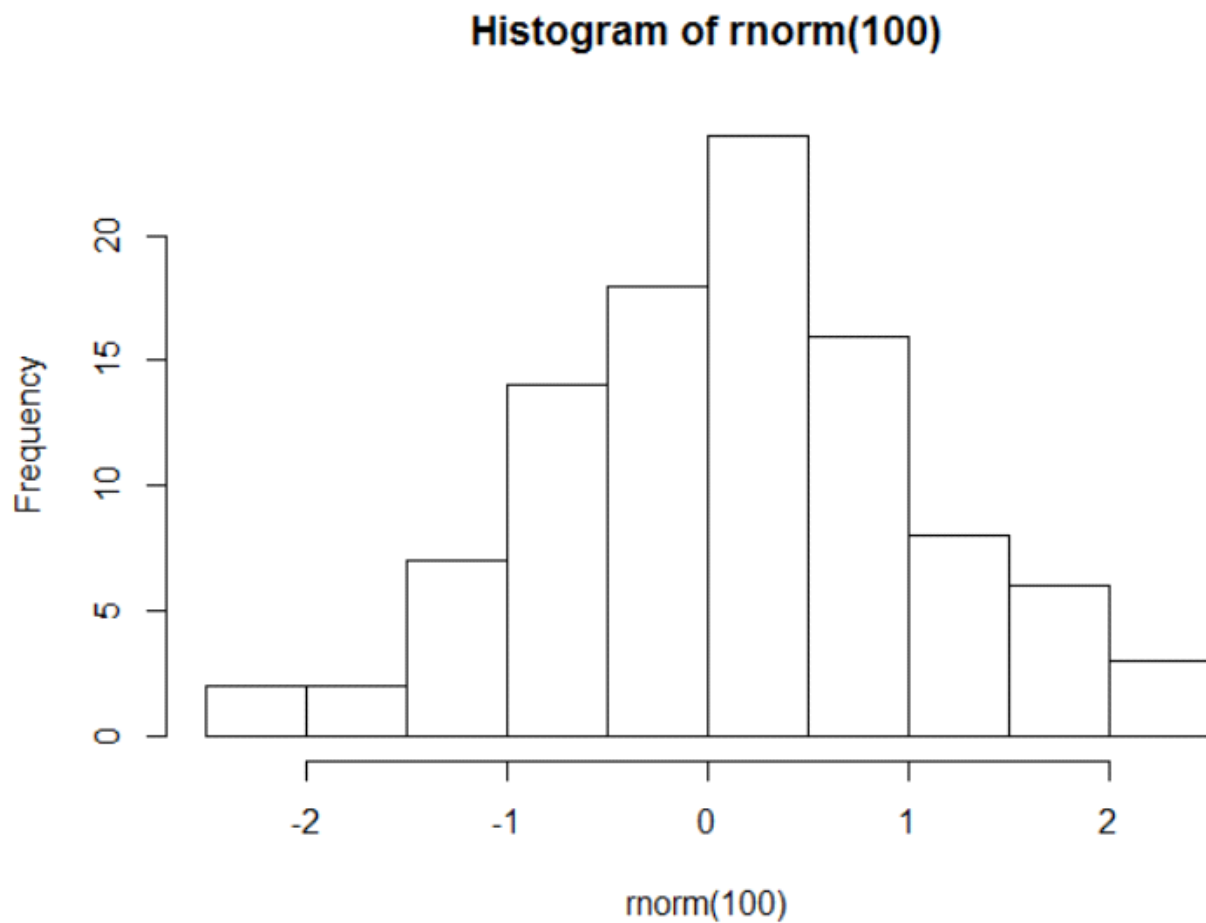
```

> max_gong_yacksoo(24,18)
[1] "최대공약수는 6 입니다~"

```

ex)

hist(rnorm(100)) #rnorm 랜덤으로 100개 추출



숫자를 크게 넣을수록 종모양에 가까워진다.

```
library(shiny)
```

```
library(rsconnect)
```

```
ui <- fluidPage(
```

```
  sliderInput(inputId = 'num', #input 변수
    label = "choose a number",
    value = 25, min=1,
    max=100),
```

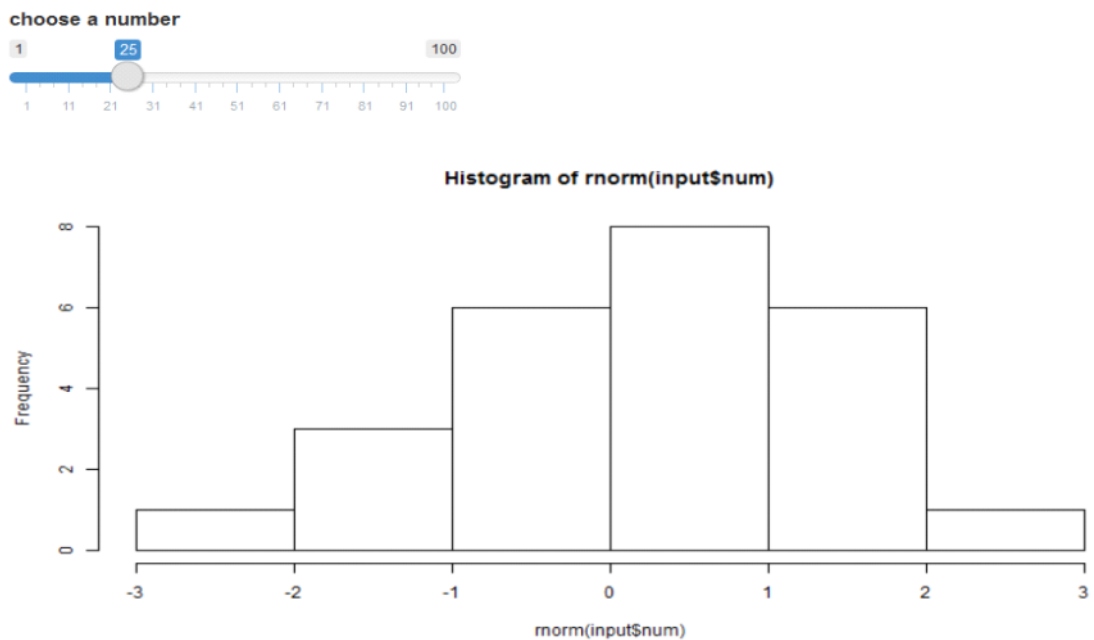
```
  plotOutput("hist")
```

```
)
```

```
server <- function(input, output) {

  output$hist <- renderPlot({#renderPlot이라던가 Reactive variable 을 계속 입력받고 변
    hist(rnorm(input$num)) 하는 변수를 써줘야한다
  })
}

shinyApp(ui = ui, server = server)
```



제목 넣기-----

```
ui <- fluidPage(

  sliderInput(inputId = 'num',
    label = "choose a number",
    value = 25, min=1,
    max=100),
  textInput(inputId = "title",
    label = "Write a title",
    value = "Histogram of Random normal values"
  ), #개체들은 , 로 구분해줘야함.
  plotOutput("hist")
)

server <- function(input, output) {

  output$hist <- renderPlot({ #렌더 플롯으로 한번 묶어줘야한다
    hist(rnorm(input$num),main = input$title)
```

```

    })
  }
  shinyApp(ui = ui, server = server)

```

```

-----
library(shiny)
ui <- fluidPage(

  sliderInput(inputId='num',
              label = "choose a number",
              value = 25, min=1, max=100), # value 값 초기값
  textInput(inputId = "title",
            label = "Write a title",
            value = "Histogram of Random normal values"), # value 값 초기값
  plotOutput("plot"),
  verbatimTextOutput("stats") #출력하는 함수
)

```

```

server <- function(input, output) {

```

```

  data <- reactive({
    rnorm(input$num)
  })

```

```

  output$plot <- renderPlot({
    boxplot(data())
  })

```

```

  output$stats <- renderPrint({
    summary(data())
  })
}

```

```

shinyApp(ui = ui, server = server)

```

#제목 고정-----

```

ui <- fluidPage(

  sliderInput(inputId = 'num',
              label = "choose a number",
              value = 25, min=1,

```

```

        max=100),
textInput(inputId = "title",
          label = "Write a title",
          value = "Histogram of Random normal values"
),
plotOutput("hist")
)

```

```
server <- function(input, output) {
```

```

  output$hist <- renderPlot({
    hist(rnorm(input$num),main = isolate({input$title}))
  })
}

```

```
shinyApp(ui = ui, server = server)
```

#액션버튼 생성-----

```

ui <- fluidPage(

  actionButton(inputId = "clicks",
               label = "Click me")

)

```

```

server <- function(input, output) {
  observeEvent(input$clicks,
               {print(as.numeric(input$clicks)) }
  )
}

```

```
shinyApp(ui = ui, server = server)
```

```

[1] 30
[1] 31
[1] 32
[1] 33
[1] 34
[1] 35
[1] 36
[1] 37
[1] 38
[1] 39
[1] 40

```

클릭하면 올라감

```
library(shiny)
```

```

ui <- fluidPage(

  actionButton(inputId = "clicks",
               label = "Click me")
)

server <- function(input, output) {
  observe({as.numeric(
    print(input$clicks) )})
}

shinyApp(ui = ui, server = server)

```

```

[1] 0
attr(,"class")
[1] "integer"           "shinyActionButtonValue"
[1] 1
attr(,"class")
[1] "integer"           "shinyActionButtonValue"
[1] 2
attr(,"class")
[1] "integer"           "shinyActionButtonValue"

```

자세한 정보가 나온다.