

1. 선 그래프 그리기**1) plot() 함수****plot(x축 데이터, y축 데이터, 옵션)**

- plot함수는 기본적으로 인수 값이 하나만 지정되면 y축으로 표현됨
- x축 데이터를 명시하지 않으면 자동으로 y축 값의 개수만큼 1,2,3 으로 지정
- 나중에 다중공선성 낮출 때 사용(산점도 ~ 상관관계 확인)
- 그냥 dataframe 형태로 넣으면 모든 column간 (모든 경우의 수) 상관관계 보여줌

인수	설명
main = "메인제목"	제목 설정
sub = "서브 제목"	서브 제목
xlab = "문자", ylab = "문자"	x, y 축에 사용할 문자열을 지정
tmag = 2	제목 등에 사용되는 문자의 확대율 지정
ann = F	x, y축 제목 지정 안 함 (여러 차트 겹칠 때 축 이름 겹침 방지)
axes = F	축을 표시하지 않음 (like ann=F)
axis	축의 범위 지정

그래프 타입 선택	설명
type="p"	점 모양 그래프 (기본값)
type="l"	선 모양 그래프 (꺼은선 그래프)
type="b"	점과 선 모양 그래프 (점이 선 통과 X)
type="c"	"b"에서 점을 생략한 모양
type="o"	점과 선을 중첩해서 그린 그래프(점이 선 통과)
type="h"	각 점에서 x축까지의 수직선 그래프
type="s"	왼쪽값을 기초로 계단모양으로 연결한 그래프
type="S"	오른쪽 값을 기초로 계단모양으로 연결한 그래프
type="n"	축만 그리고 그래프는 그리지 않음

선의 모양 선택	
lty=0, lty="blank"	투명선
lty=1, lty="solid"	실선
lty=2, lty="dashed"	대쉬선
lty=3, lty="dotted"	점선
lty=4, lty="dotdash"	점선과 대쉬선
lty=5, lty="longdash"	긴 대쉬선
lty=6, lty="twodash"	2개의 대쉬선

색, 기호 등	
col=1, col="blue"	기호의 색지정 1:8 순서대로 "black" "red" "green3" "blue" "cyan" "magenta" "yellow" "gray"
pch=0, pch="문자"	점의 모양을 지정
bg="blue"	그래프의 배경색 지정
lwd=숫자	선의 굵기 지정
cex=숫자	점/문자 그릴 때 굵기 지정

참조 - 학교 수업

pch : 점 기호의 종류를 나타내는 것으로, 0~25 중 하나의 숫자를 선택하거나 인용부호 안에 하나의 문자를 지정할 수 있음 (예 : pch=2, pch=*)

lty : 선의 종류를 나타내는 것으로 1~6중 하나의 숫자를 선택

cex : 기호의 크기를 나타내는 것으로 cex=1은 디폴트 크기, cex=2는 2배 확장된 크기로 점 기호가 표기됨

lwd : 선의 굵기를 나타내는 것으로 lwd=1은 디폴트 굵기, lwd=2는 2배로 확장된 굵기로 선이 그려짐

##옵션 pch와 lty의 선택 가능한 형태

○ 1	△ 2	+ 3	× 4	◇ 5	1 ——— 1 ——— 1 <
▽ 6	⊠ 7	※ 8	⊕ 9	⊗ 10	2 - - - - - 2 - - - - - 2 -
⊗ 11	⊞ 12	⊠ 13	⊞ 14	■ 15	3 3 - - - - - 3 <
● 16	▲ 17	◆ 18	● 19	● 20	4 4 - - - - -
○ 21	□ 22	◇ 23	△ 24	▽ 25	5 - - - - - 5 ——— ———
					6 - - - - - 6 ——— ———
					lines lines
					lty lwd

4.2.2 색깔과 관련된 모수

★★

-col : 그래프에 그려지는 기호, 선, 문자 등의 색깔을 지정

-col.lab : X,Y축 라벨의 색깔

-col.name : 그래프 주 제목의 색깔

-col.sub : 그래프 부 제목의 색깔

##모수에 색깔 지정하는 방법

1. 숫자 이용

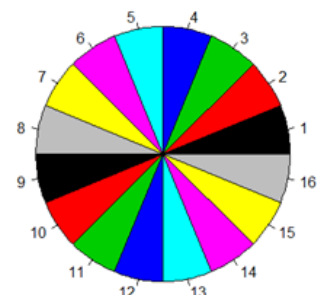
예 : col=2

0. 흰색 1. 검은색 2. 빨간색 3. 녹색 4. 파란색 5. 청록색

6. 자주색 7. 노란색 8. 회색

9 이상의 숫자는 검은색부터 순환

> pie(rep(1,16),col=1:16)



palette()

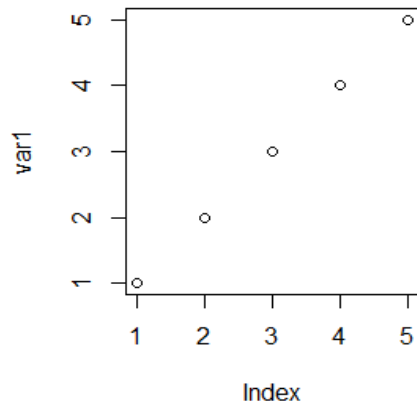
```
[1] "black" "red" "green3" "blue" "cyan" "magenta" "yellow"
"gray"
```

+ 사용자 정의함수를 통해 팔레트도 정의할 수 있겠다. (ex. 몇 이상은 무슨색, ...)

(1) 기본 값으로 그래프 만들기

```
var1<-c(1,2,3,4,5)
```

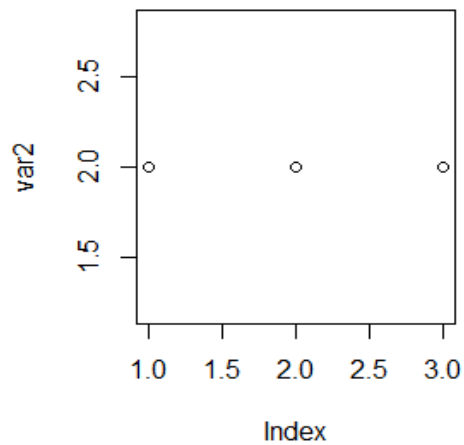
```
plot(var1)
```



(2) y축 값을 동일하게 설정해서 출력하기

```
var2<-c(2,2,2)
```

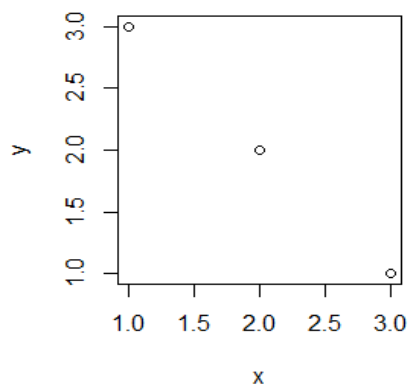
```
plot(var2)
```



(3) x, y축의 값을 다 지정해서 출력하기

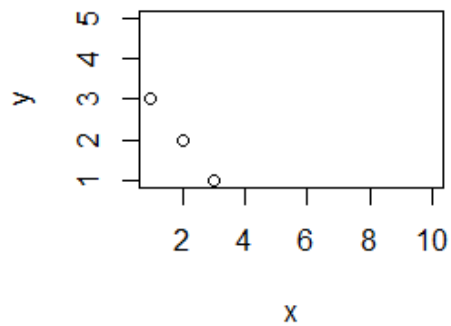
```
x<-1:3;y<-3:1
```

```
plot(x,y)
```



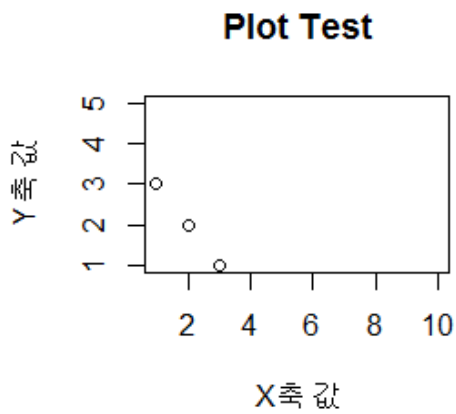
(4) x, y축의 최대 한계값 지정하기 (각 범주 설정)

```
plot(x,y,xlim=c(1,10),ylim=c(1,5))
```



(5) x축과 y축 제목, 그래프 제목 지정해서 출력하기

```
plot(x,y,xlim=c(1,10),ylim=c(1,5),xlab="X축 값", ylab="Y축 값", main="Plot Test")
```

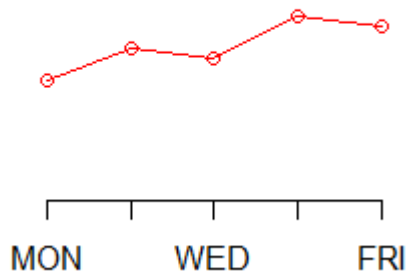


(6) 조금 더 멋진 그래프 예

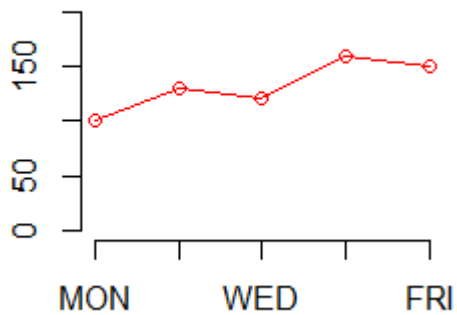
```
> plot(v1, type='o', col='red', ylim=c(0,200), axes=FALSE, ann=F)
```



```
> axis(1, at=1:5, lab=c("MON", "TUE", "WED", "THU", "FRI"))
```

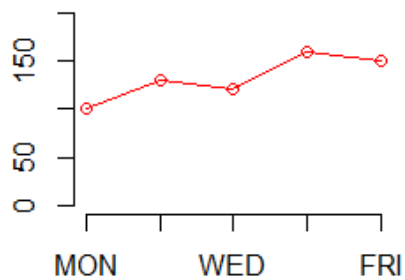


```
> axis(2, ylim=c(0,200))
```

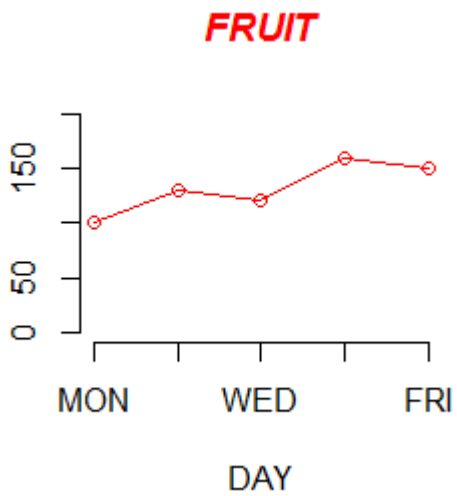


```
> title(main="FRUIT", col.main="red", font.main=4)
```

FRUIT



```
> title(xlab="DAY",col.lab="black")
```



```
> title(ylab="PRICE", col.lab="blue")
```

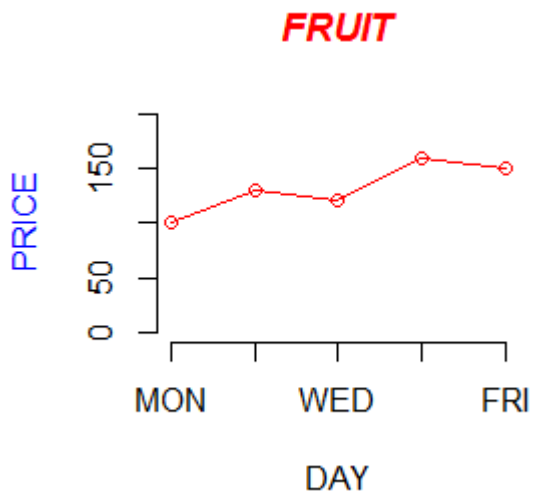


차트 밖에서 옵션 수정

axis - 축에 관련한 것 수정

title - 제목에 관한 것 수정

- 1. plot 차트 안에서 옵션 수정 가능 - 이 plot()에 관한 옵션만 수정
- 2. plot 차트 함수 밖에서 옵션 설정 - 도화지에 있는 모든 그래프에 공통적으로 적용시킴

plot.new() 현재 창에 그래프 새로 그리기

dev.new() 새로운 창에 그래프 새로 그리기 cf. in python - figure

(7) 그래프의 배치 조정하기 ★★★★★mfrow★★★★★

par(mfrow=c(nr,nc))

#nr : 행의 개수 / nc : 열의 개수

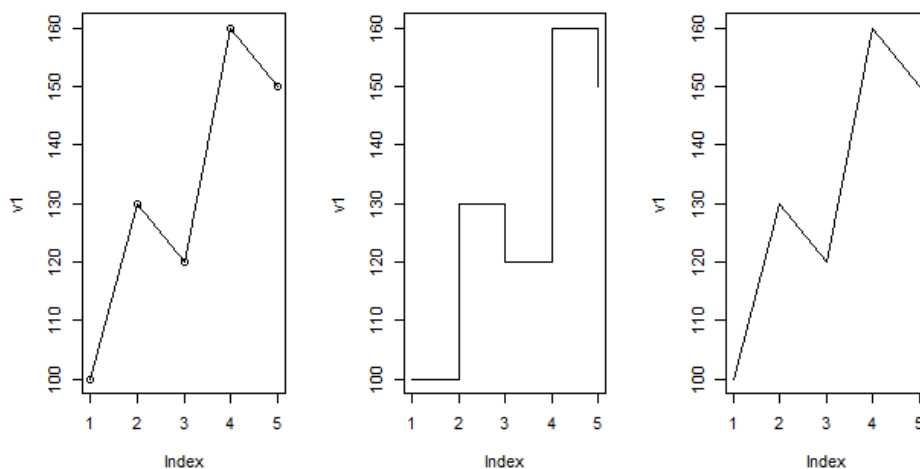
- 한 창을 분할해서 여러 그래프 그리기 cf. python : subplot ~ figure
- par() 여러 개에 전달시..!

par(mfrow=c(1,3))

plot(v1,type="o")

plot(v1,type="s")

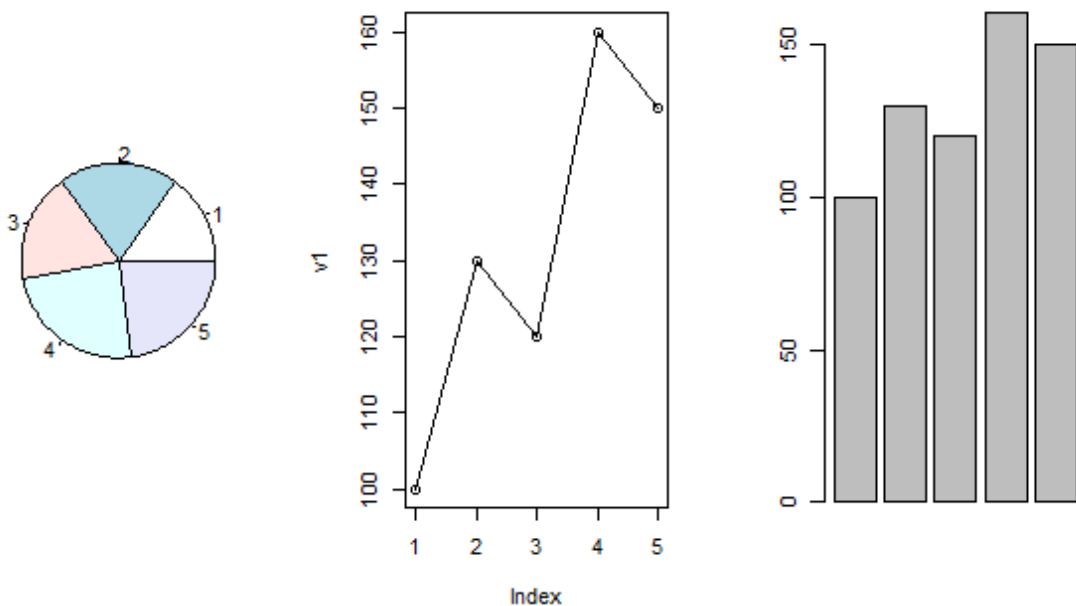
plot(v1,type="l")



pie(v1)

plot(v1,type="o")

barplot(v1)

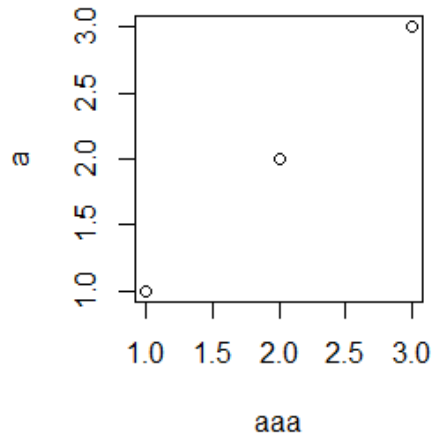


(8) 그래프 그릴 때 여백 조정하기

1) mgp() 함수

```
a <- c(1,2,3)
```

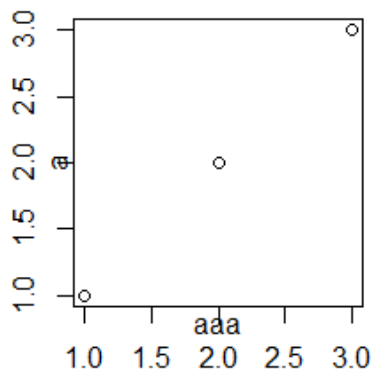
```
plot(a,xlab="aaa")
```



제목 위치 조정

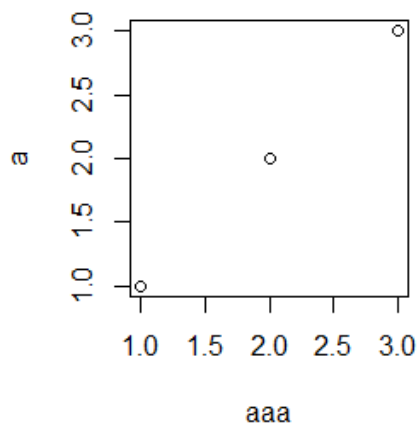
```
par(mgp=c(0,1,0)) # mgp=c(제목위치, 지표값위치, 지표선 위치)
```

```
plot(a,xlab="aaa")
```



```
par(mgp=c(3,1,0))
```

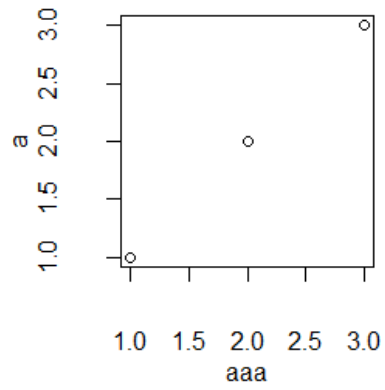
```
plot(a,xlab="aaa")
```



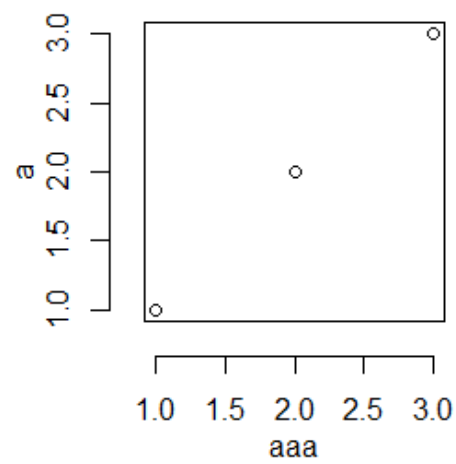
지표값 위치 조정

```
par(mgp=c(3,2,0))
```

```
plot(a,xlab="aaa")
```



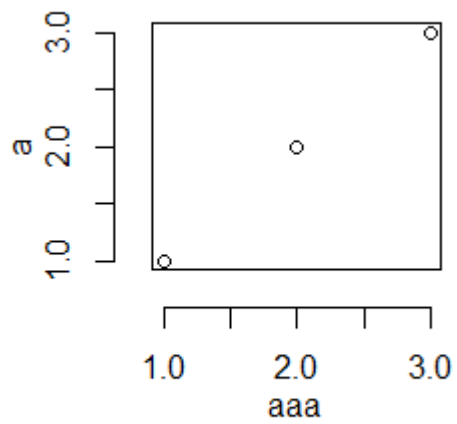
지표선 위치 조정



2) oma() 함수

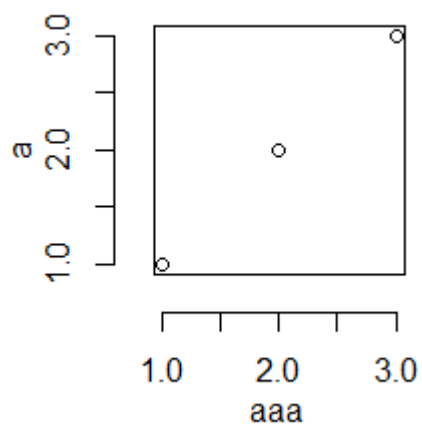
```
par(oma=c(2,1,0,0)) #oma=c(하,좌,상,우)
```

```
plot(a,xlab="aaa")
```



```
par(oma=c(0,2,0,0))
```

```
plot(a,xlab="aaa")
```



cf. par() 함수 - 그 창에서만!

oma ~ 그래프 한 창에 여러 개 그릴 때, 각각의 여백을 조정할 때 사용

★ ★ (9) 여러 개의 그래프를 중첩으로 그리기 ★ ★

1) `par(new=T)`

`par(mfrow=c(1,1))` # 이전 실습에서 3개로 출력하게 한 것을 1개로 만들기 위해 사용

`v1<-c(1:5)`

`v2<-c(5:1)`

`v3<-c(3:7)`

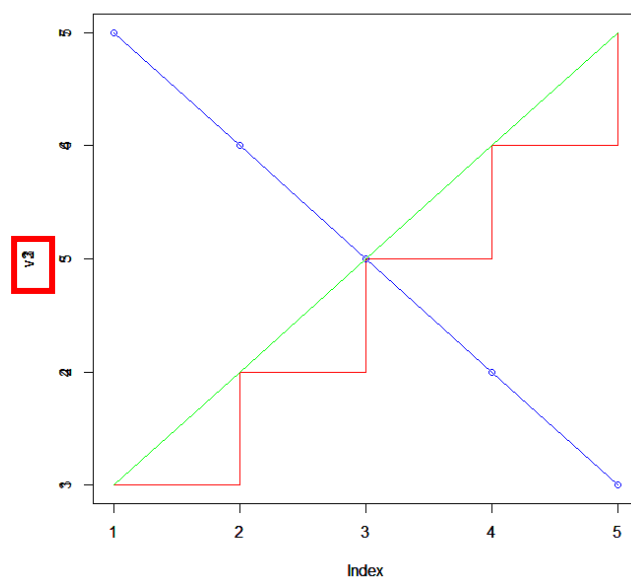
`plot(v1, type="s", col="red", ylim=c(1,5))`

`par(new=T)` # 다음 함수 중복을 허용함

`plot(v2,type="o", col="blue", ylim=c(1,5))`

`par(new=T)` # 다음 함수 중복을 허용함

`plot(v3, type="l", col="green")`



`par(new=T)`의 불편한 점

- 그래프 추가로 그리기 전에 `par(new=T)` 작성해야 함
- 축 이름이 겹침

2) lines() 함수

```
v1 <- c(1:5)
```

```
v2 <- c(5:1)
```

```
v3 <- c(3:7)
```

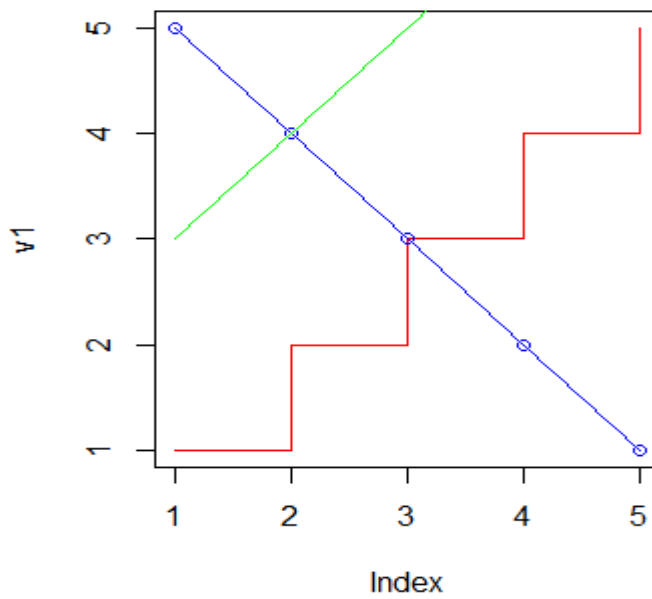
```
plot(v1, type="s", col="red", ylim=c(1,5))
```

```
lines(v2, type="o", col="blue", ylim=c(1,5))
```

#아무리 lines()를 써서 옵션을 써도

```
lines(v2, type="l", col="green", ylim=c(1,15))
```

1st 그래프에서 쓴 옵션만 적용됨



- lines()로 그래프 추가 시 첫번째 그래프의 축 이름, 범주를 그대로 사용함

(10) 범례 추가하기

legend(x축 위치, y축 위치, 내용, cex=글자크기, col=색상, pch=크기, lty=선모양)

x축 위치 : 범례를 시작할 x축 위치

y축 위치 : 범례를 마칠 y축 위치

내용을 주면 순차적으로 !

cf. in python - loc=best (최상의 조건 알아서 맞춰줌!)

```
v1<-c(1:5)
```

```
v2<-c(5:1)
```

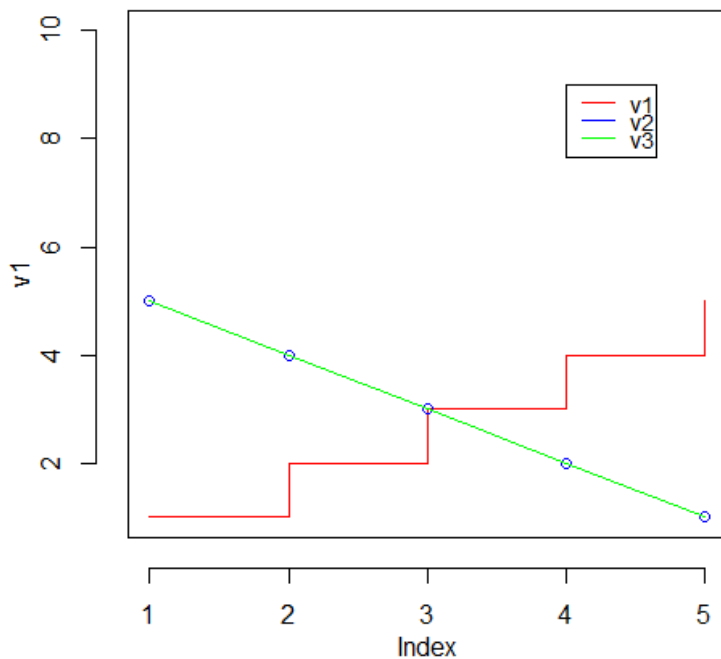
```
v3<-c(3:7)
```

```
plot(v1, type="s", col="red", ylim=c(1,10))
```

```
lines(v2, type="o", col="blue", ylim=c(1,5))
```

```
lines(v2, type="l", col="green", ylim=c(1,15))
```

```
legend(4, 9, c("v1","v2","v3"), cex=0.9, col=c("red","blue","green"), lty=1)
```



plot 문제

각 사원별 월별 성취도 추이 겹쳐서 그리기

```
d2<-read.csv("상반기사원별월별실적현황_new.csv",stringsAsFactor=F)
```

-----dcast로 나누기로 더 권장! python에서 쓰임-----

(마지막 컬럼에 있는 성취도가 값으로 지정될 것임 - 중간 순서 column을 value로 하고 싶으면 따로 뺀 후에)

★bar chart에서는 dcast를 꼭 해야!!!! 로우색인 안 됨★

강사님 방법

★ 겹쳐그릴 때 **ann=F** 쓰기!!★ -> 뒤에서 새로 title()함수

★ 틀린 경우에는 다시 처음부터 써야 함★

```
library(reshape2)
```

```
d3<-dcast(d2, 월 ~ 이름)
```

```
plot(d3$박동주, type="o",col="red",ylim=c(0.6,1.2),ann=F)
```

```
lines(d3$최경우, type="o",col="blue")
```

```
lines(d3$주시현, type="o", col="green")
```

```
lines(d3$이동근, type="o", col="black")
```

```
lines(d3$윤정웅, type="o", col="yellow")
```

```
lines(d3$박지영, type="o", col="magenta")
```

```
lines(d3$김지수, type="o", col="grey")
```

```
title(main="월별 성취도 추이", xlab="월", ylab="성취도")
```

```
_name <- colnames(d3)[-1]      # 자주 쓰일 방법!
```

```
legend(4,1.2,_name, col=c(1:7), lty=1, cex=0.7)
```

#legend - x, y축 눈금에 따라! x : 시작하는 눈금, y : 끝나는 눈금, cex 글씨 크기 조절)

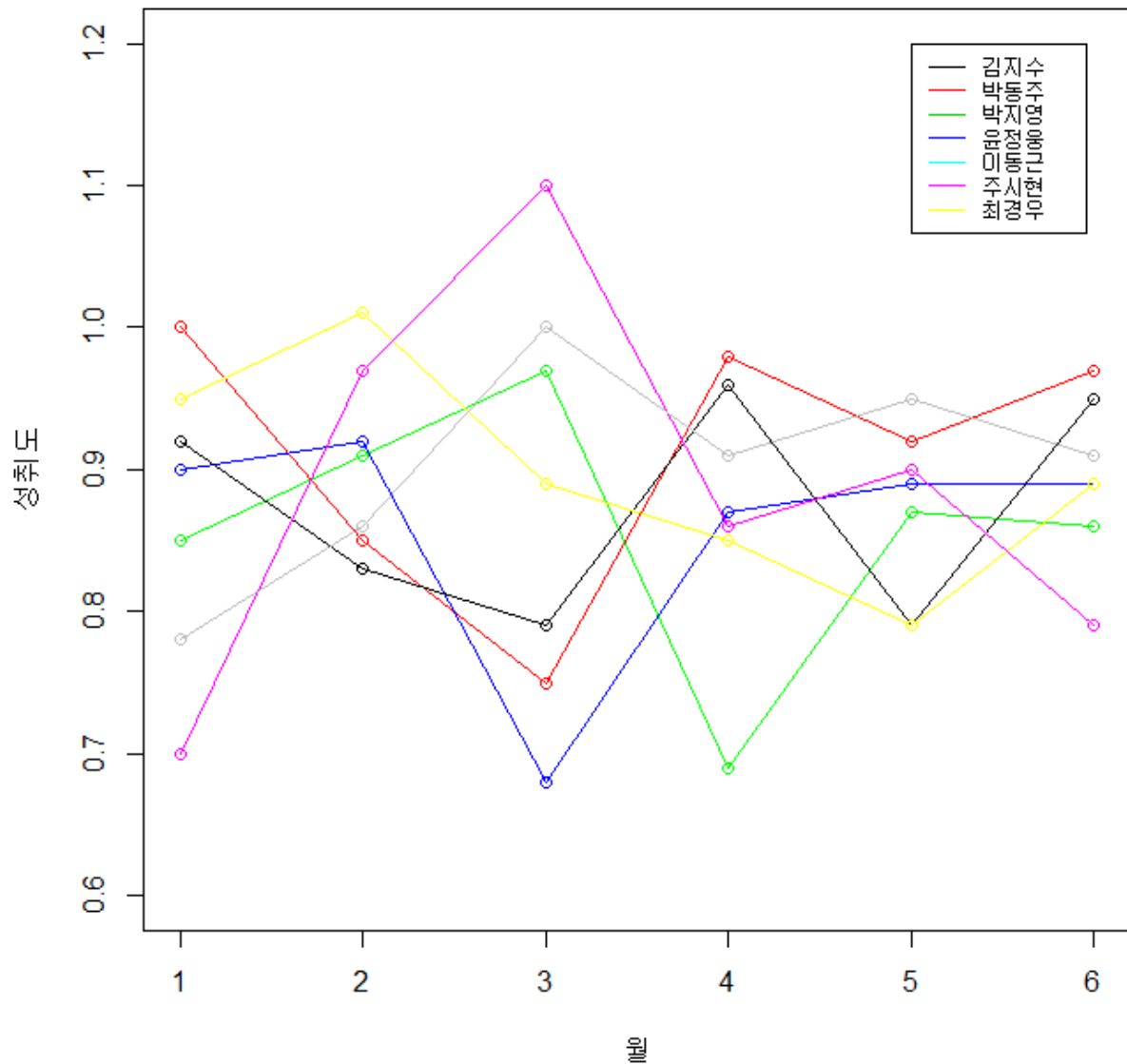
#cf. dcast - 행고정 column ~ 열고정 column

unstack - value column ~ 열고정 column

in unstack - 단 두 가지만 선택 가능 (열 컬럼 + value) -> 월의 의미가 사라짐

```
unstack(d2, 성취도 ~ 이름)
```

월별 성취도 추이



내 흔적

#dcast

이번 경우는 월마다 데이터가 하나여서 단순 나눔! 만일 grouping인 경우 그룹별 함수(mean, ...) 지정 가능

```
library(reshape2)
```

```
d3<-dcast(d2, 월 ~ 이름)
```

```
plot(d3$월,d3$박동주, type="o",col=1,ylim=c(0.6,1.1),ylab="성취도",xlab="월")
```

```
lines(d3$월,d3$최경우, type="o",col=2)
```

```
lines(d3$월,d3$주시현, type="o", col=3)
```

```
lines(d3$월,d3$이동근, type="o", col=4)
lines(d3$월,d3$윤정웅, type="o", col=5)
lines(d3$월,d3$박지영, type="o", col=6)
lines(d3$월,d3$김지수, type="o", col=7)
```

생각해보니 dcast는 월별 순서대로 정리되어서 d3\$월은 없어도 되겠군...

~ row색인 / 쉬운 방법

```
plot(d2[d2$이름=="박동주",2],d2[d2$이름=="박동주",c("성취도",
)],type="o",col=1,ylim=c(0.6,1.1),ylab="성취도",xlab="월")
lines(d2[d2$이름=="최경우",2],d2[d2$이름=="최경우",c("성취도")],type="o",col=2)
lines(d2[d2$이름=="주시현",2],d2[d2$이름=="주시현",c("성취도")],type="o",col=3)
lines(d2[d2$이름=="이동근",2],d2[d2$이름=="이동근",c("성취도")],type="o",col=4)
lines(d2[d2$이름=="윤정웅",2],d2[d2$이름=="윤정웅",c("성취도")],type="o",col=5)
lines(d2[d2$이름=="박지영",2],d2[d2$이름=="박지영",c("성취도")],type="o",col=6)
lines(d2[d2$이름=="김지수",2],d2[d2$이름=="김지수",c("성취도")],type="o",col=7)
```

#[,] 벡터 출력

앗 강사님은

```
plot(d2[d2$이름=="박동주",c("성취도")],type="o",col="red",ylim=c(0.6,1.1),ylab="성취도",xlab="월")
```

바로 성취도...!!!!

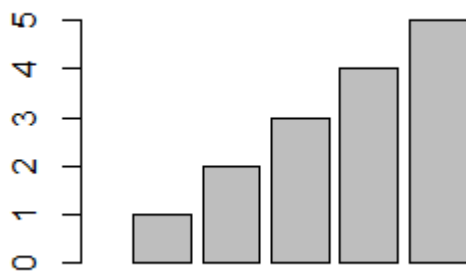
그런데 만약에 월이 1,2,3,4,5,6 순서로 되어있지 않고 4,3,5,2,1,6 이런 순서였으면 다르게 나타났을 것!

2) barplot() : 막대그래프 그리기

인수	기능
angle, density, col	막대를 칠하는 선분의 각도, 선분의 수, 선분의 색 지정
legend	범례 그리기
names	각 막대의 라벨을 정하는 문자열 벡터를 지정
width	각 막대의 상대적 폭을 벡터로 지정
space	각 막대 사이의 간격을 지정
beside	TRUE를 지정하면 각각의 값마다 막대를 그림
horiz	TRUE를 지정하면 막대를 옆으로 눕혀서 그림

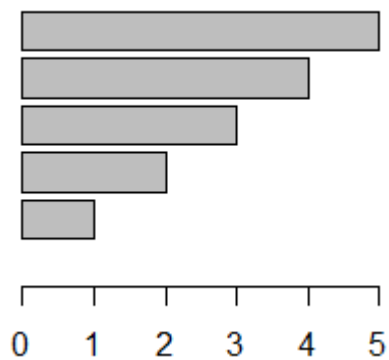
(1) 기본 bar 그래프 그리기

```
x<-c(1:5)
barplot(x)
```



(2) 그래프를 가로로 출력하기

```
x<-c(1:5)
barplot(x, horiz=T)
```



(3) ★ ★ ★ ★ 그룹(열별)으로 묶어서 출력시키기 - beside=T 사용 ★ ★ ★ ★

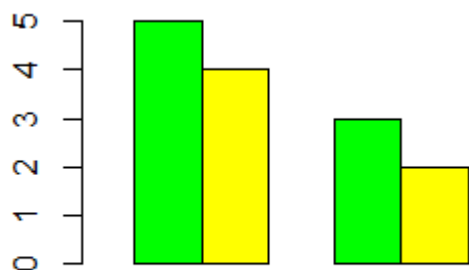
(★ ★ ★ ★ matrix 형태의 데이터만 가능!!!! ★ ★ ★ ★)

```
> x<-matrix(c(5,4,3,2),nrow=2)
> barplot(x, beside=T, names=c(1,2),col=c("green","yellow"))
> x
      [,1] [,2]
[1,]    5    3
[2,]    4    2
```

beside = T -> 열별로 묶음

names -> 열별 이름 (in matrix) / 막대그룹 이름

col -> 행별(in matrix) 색깔 지정



1

2

1st열- 한파트 / 1

2nd열- 한파트/2

	[,1]	[,2]	
[1,]	5	3	1st 행(초록색)
[2,]	4	2	2nd 행(노란색)

cf. 연습문제 참조

in bar plot 표시 (월별)

데이터 형태

1월 2월 3월 4월 ... -> 열

박동주

최경우

...

-> 행

in plot 표시 (월별)

박동주 최경우 ... -> 열

1월

2월

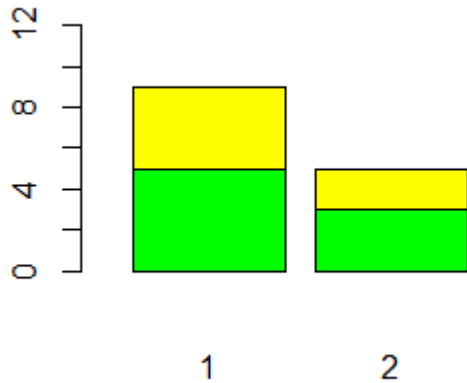
...

-> 열

(4) 하나의 막대 그래프로 출력하기 - row별 portion 의미

```
x<-matrix(c(5,4,3,2),nrow=2)
```

```
barplot(x,names=c(1,2),col=c("green","yellow"),ylim=c(0,12))
```

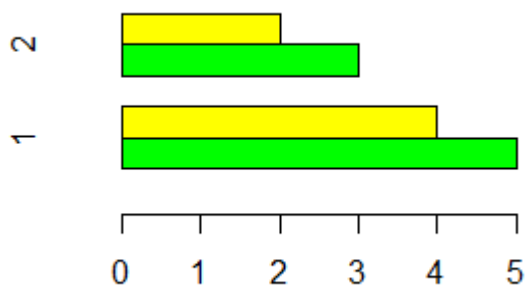


(5) 그룹으로 묶어서 가로로 출력시키기 - beside, horiz = T 사용

```
x<-matrix(c(5,4,3,2),nrow=2)
```

```
par(oma=c(1,0.5,1,0.5))
```

```
barplot(x,names=c(1,2),beside=T,col=c("green","yellow"),horiz=T)
```

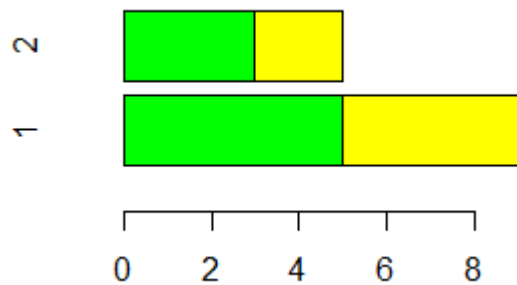


oma부분은 여백 표시, c(하, 왼쪽, 상, 오른쪽) / 그래프의 여백이 많을 경우 요긴하게 사용 가능

(6) 하나의 막대 그래프에 두 개 합쳐서 넓혀서 출력하기

```
x<-matrix(c(5,4,3,2),nrow=2)
```

```
barplot(x,names=c(1,2),horiz=T,col=c("green","yellow"))
```



(7) 여러 막대 그래프를 그룹으로 묶어서 한꺼번에 출력하기

```
1<-c(100,120,140,160,180)
```

```
v2<-c(120,130,150,140,170)
```

```
v3<-c(140,170,120,110,160)
```

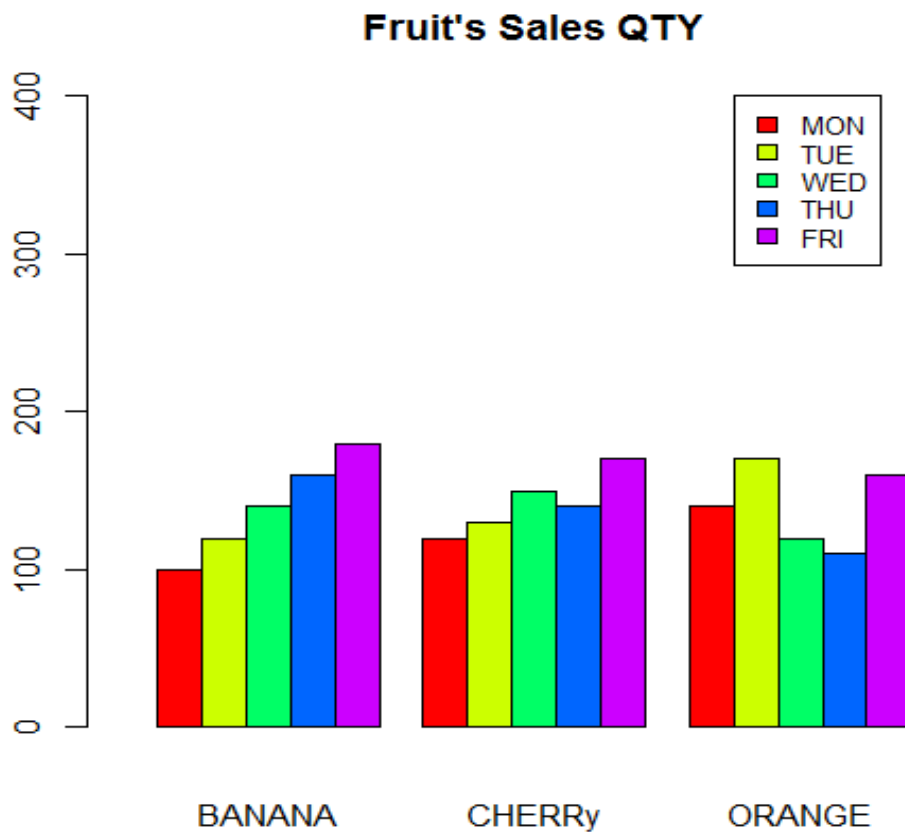
```
qty<-data.frame(BANANA=v1,CHERRY=v2,ORANGE=v3)
```

```
barplot(as.matrix(qty),main="Fruit's Sales QTY", beside=T, col=rainbow(nrow(qty)),ylim=c(0,400))
```

```
legend(14,400,c("MON","TUE","WED","THU","FRI"),cex=0.8,fill=rainbow(nrow(qty)))
```

fill - 범례 채우는 형태로 표현할 때

cf. lty - 선



- beside=T일 때만 전제조건 : matrix 형태! -----시험문제-----

(8)하나의 막대 그래프에 여러 가지 내용을 한꺼번에 출력하기

```
barplot(t(qty),main="Fruit's Sales QTY", ylim=c(0,900), col=rainbow(nrow(qty)),
        space=0.1, cex.axis=0.8, las=1, names.arg=c("MON","TUE","WED","THU","FRI"), cex=0.8)
legend(0.2, 800, names(qty), cex=0.7, fill=rainbow(length(qty)))
```

t(qty) 주목!!!! col별 막대그래프!!

#names - col이름 출력 in dataframe

```
> qty
  BANANA CHERRY ORANGE
1    100    120    140
2    120    130    170
3    140    150    120
4    160    140    110
5    180    170    160
```

```
> t(qty)
      [,1] [,2] [,3] [,4] [,5]
BANANA  100  120  140  160  180
CHERRY  120  130  150  140  170
ORANGE  140  170  120  110  160
```

(9) 조건을 주고 그래프 그리기

- peach값이 200 이상일 경우는 "red", 180 - 199는 "yellow",
그 이하는 "green"으로 출력하라는 코드

```
peach<-c(180,200,250,198,170)
```

```
colors<-c()
```

```
for( i in 1:length(peach)){
```

```
  if (peach[i] >= 200) {  
    colors<-c(colors,"red")
```

```
  }
```

```
  else if (peach[i]>=180){
```

```
    colors<-c(colors,"yellow")
```

```
  }
```

```
  else{
```

```
    colors<-c(colors,"green")
```

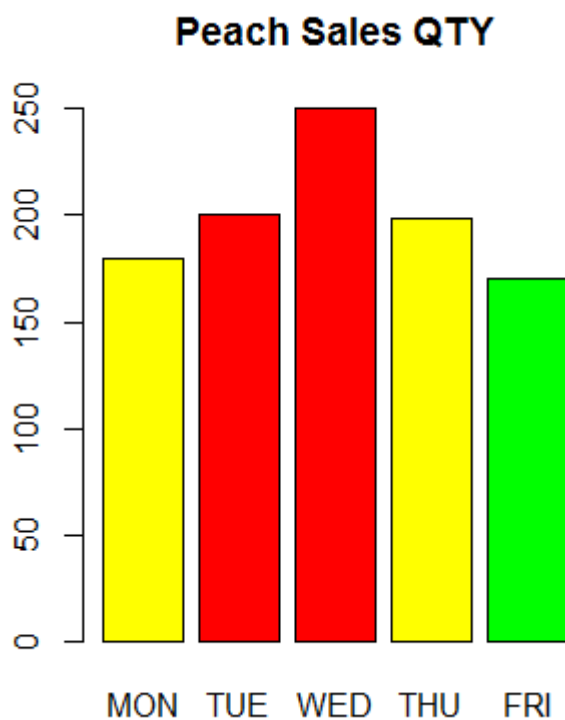
```
  }
```

```
}
```

```
barplot(peach, main="Peach Sales QTY", names.arg=c("MON","TUE","WED","THU","FRI"),col=colors)
```

```
# 자꾸만 i in 1:m 에서 1:을 빼먹음 ! ππ
```

```
# 팔레트 ~ 사용자 정의함수 + 빈 벡터 + 조건에 따른 색 설정
```



연습문제

1. 상반기 사원별 실적현황 데이터를 기반으로 각 월별 학생의 성취율을 막대그래프로 표현

```
d1<-read.csv("상반기사원별월별실적현황_new.csv",stringsAsFactors=F)
```

```
library(reshape2)
```

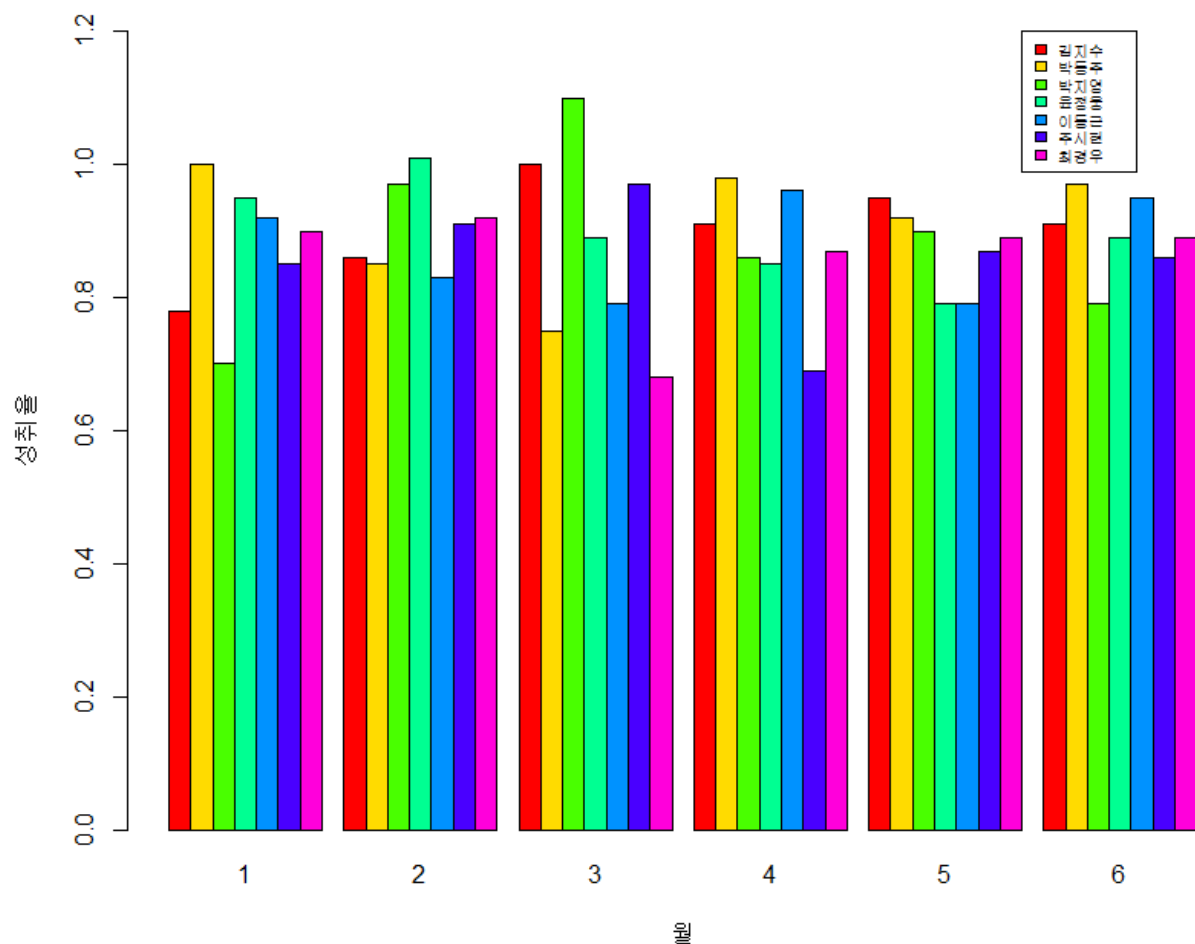
```
d2<-dcast(d1, 이름 ~ 월)
```

cf. 앞 문제랑 비교!!! in barplot ~ x축 : 열

```
barplot(as.matrix(d2[,-1]), beside=T, names=colnames(d2)[-1], col=rainbow(nrow(d2)), ylim=c(0,1.2))
```

```
legend(40,1.2,c(d2$이름), cex=0.6, fill=rainbow(nrow(d2)))
```

```
title(xlab="월",ylab="성취율")
```



cf. unstack(d1, 성취도 ~ 월) # 포물러 : value컬럼 ~ 열고정 컬럼

cf. 계속 안 풀렸던 것!

처음시도

barplot(as.matrix(d2)_[-1], names=r_names, beside=TRUE) ---> **안됨@@@**

안 되는 이유

dataframe -> matrix 하는 중, matrix는 모든 데이터 타입이 동일해야하므로
이름+숫자가 있던 원 dataframe을 matrix로 바꾸면 모두 "문자"로 바뀜!

그래서 1. 문자를 뺀 상태에서(d2[-1])

2. 1을 **as.matrix(d2[-1])**로 바꿔야 **원래의 숫자 타입**으로 남게 할 수 있다.

2. subway2 파일의 데이터를 기반으로 승차가 가장 많은 top 5개의 역을 구하고 각 역의 시간대별 승차의 증감추세를 도표화 하여라

1) top 5 구하기

강사님-----훨씬 간단하였다..!

cf. skip옵션 쓰면 첫 행 빼고 불러옴! -> 나머지 숫자 되었다

```
subway <- read.csv("subway2.csv", stringsAsFactors =F, skip=1)
str(subway)
subway[subway$구분 == "승차",-2]
```

#승차데이터만 뽑기

```
subway2<-subway[subway$구분 == "승차",-2]
rowsum<-order(apply(subway2[,-1],1,sum))
order(rowsum, decreasing=T) #TOP5는 오름차순 X, 내림차순!(역순)
```

```
rownum <- order(apply(subway2[,-1],1,sum), decreasing=T)[1:5]
#rowsum 함수는 R에서만 제공되는 간편함수 / 웬만하면 apply 관련 함수 쓸 것!
```

```
total<-subway2[rownum,]
```

--->강남 데이터가 열에 있고 시간에 대한 정보가 행에 있어야 표현 가능(파이썬도)
--행렬전치필요

```
rownames(total) <- total[,1]
# t(total) 을 하면 첫 번째 열 값인 역 이름이 들어가면서 숫자도 문자화됨(매트릭스 형태)
역의 이름이 row이름인 채로 전치할 수 있도록 하기
```

```
plot_data <- t(total[,-1])
# 첫 번째 행을 제외한 나머지를 전치하기 // 전치 전에 꼭 행 이름 설정하고 전치하기!
(파이썬에서 로우색인 잘 해야 한다! 컬럼별 행별로 x축 계산, 컬럼 값으로 자동 범례화)
```

+ rownames 예쁘게 만들 필요 있음 (범위 이름)
시간대별 범위(ex. 앞 두 숫자) 지정하도록 설정하려고 rownames 변경

```
rownames(plot_data)
```

```
library(stringr)
xname<-str_sub(rownames(plot_data),2,3) # 끝 위치! 글자 개수 X
```

보기 쉽기 위해 만으로 나눠줌

```
plot(plot_data[,1]/10000, type="o", col=1, ylim=c(0, 50), ann=F, axes=F)
lines(plot_data[,2]/10000, type="o", col=2)
lines(plot_data[,3]/10000, type="o", col=3)
lines(plot_data[,4]/10000, type="o", col=4)
lines(plot_data[,5]/10000, type="o", col=5)
```

```
axis(1, at=1:nrow(plot_data), lab=xname)
```

#x축 범위 지정

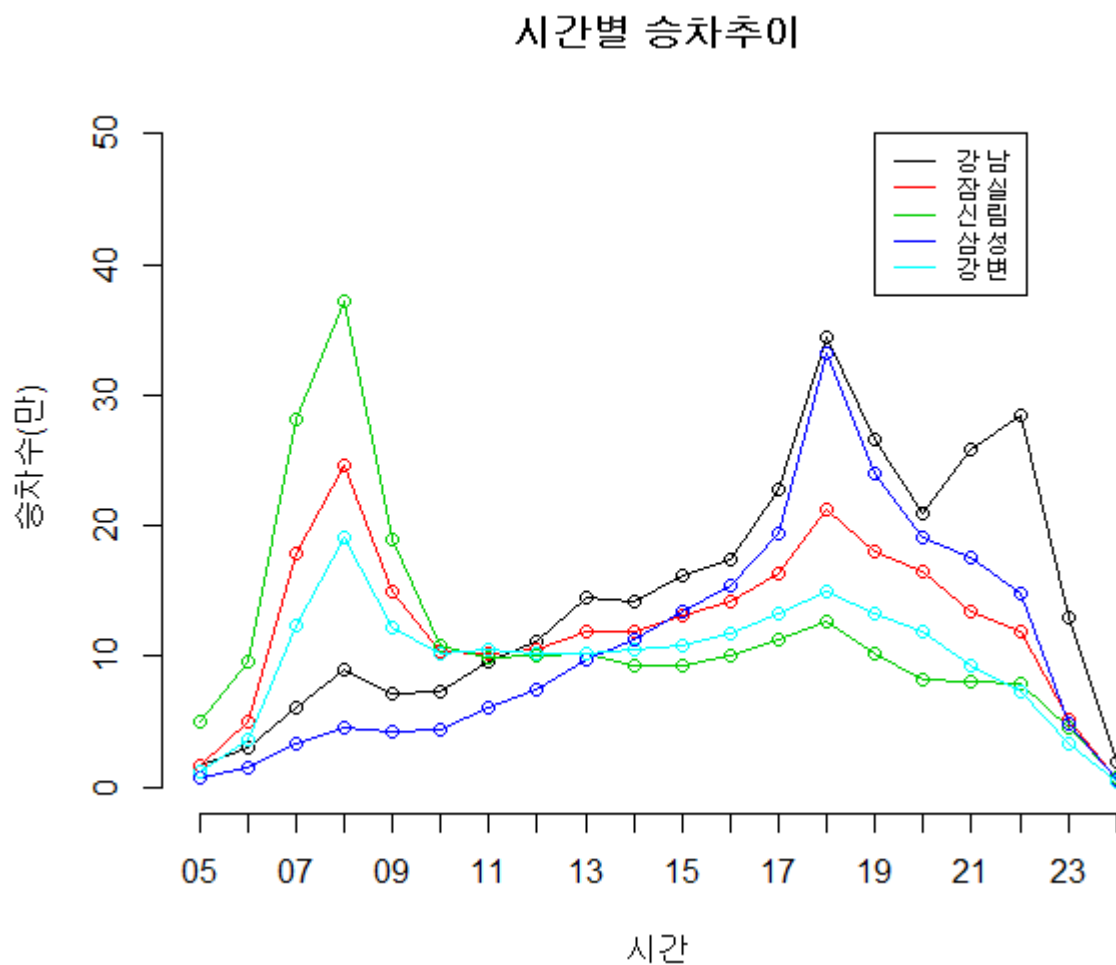
```
axis(2, ylim=c(0,50))
```

#y축 범위 지정 (이름 X!)

```
title(main="시간별 승차추이", xlab="시간", ylab="승차수(만)")
```

#제목 설정

```
legend(15, 50, colnames(plot_data), col=c(1:5), lty=1, cex=0.7)
```



내 방법

```
s1 <- read.csv("subway2.csv", stringsAsFactors = F)
s2 <- s1[-1,]
str(s2) # 숫자들이 문자형태였음
colnames(s2) <- s1[1,]
s2[,c(3:ncol(s2))] <- sapply(s2[,c(3:ncol(s2))], as.numeric) #열단위 데이터 타입 변환-sapply!!
str(s2)
s3 <- melt(s2, id.vars=c("전체", "구분")) #전체만 하면 승차/하차 따로 되고 시간별 따로 되어서!
# 역별 + 승차별 인원 합 구하기 위해 전체 + 구분 id.vars
str(s3)
s4 <- dcast(s3, 전체 ~ 구분, sum)
# 전체(역) : 고정 / 구분(승차,하차) : 열별, 시간 무시 -> 나머지 합
s4[order(s4$승차,decreasing=T),][1:5,]
```

결과 -> 강남, 잠실, 신림, 삼성, 강변

```
> s4[order(s4$승차,decreasing=T),][1:5,]
전체   승차 하차
2  강 남 2910770    0
96 잠 실 2491504    0
64 신 림 2317361    0
46 삼 성 2153134    0
3  강 변 1958187    0
```

```
s5 <- s4[order(s4$승차,decreasing=T),][1:5,] 결과를 다시 새로운 변수로 지정함
```

```
#####
```

```
s6 <- s2[s2$전체 %in% s5$전체,] # 2) 문제는 원데이터를 써서 그래프를 그려야하는데,
# s5는 승/하차가 열로 가서 처음 데이터의 행 번호랑 다름.
# 따라서 top5만 추린 s5의 "전체"열의 역 이름과 같은 데이터를 s2에서 찾는 연산
```

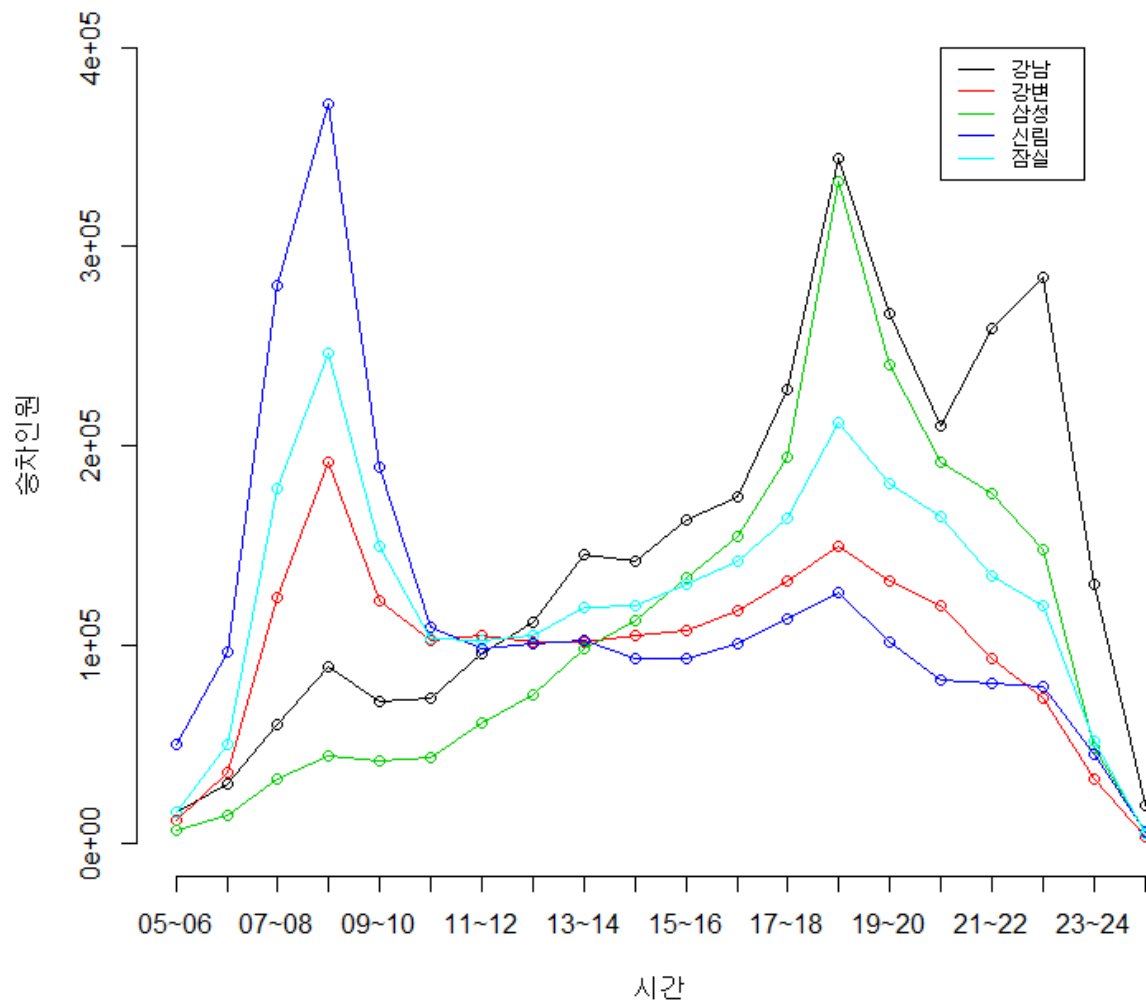
```
#####
```

```
2) s7 <- melt(s6, id.vars=c("전체","구분")) # 년도를 열 -> 행으로 바꾸기 전 단계! 정리가 안됨
s8 <- dcast(s7, variable ~ 전체) # 년도 : 고정 col, 역 : 펼치는 col (역별 표시-> 역이 펼치는 col)
library(stringr)
colnames(s8) <- str_replace(colnames(s8), " ", "") # 이름 선택하려고 하는데 띄어쓰기 거슬려서
```

```
plot(s8$강남, type="o", col=1, ann=F, ylim=c(0,400000), axes = F)
lines(s8$강변, type="o", col=2)
lines(s8[, 4], type="o", col=3) # $__ 형태도 되고 [, ] 형태도 가능! 대신 열 순서 재확인!
lines(s8$신림, type="o", col=4)
lines(s8$잠실, type="o", col=5)
```

```
axis(1, at=1:nrow(s8),lab=s8$variable)
```

```
axis(2, ylim=c(0,400000)) # 이것보다 plot에서 정의한 함수가 먼저 순서!
title(xlab="시간",ylab="승차인원", title="승차인원 TOP5 시간별 승차인원")
legend(16, 400000, colnames(s8)[-1], cex=0.7, col=c(1,2,3,4,5), lty=1)
```



3. employment 데이터를 기반으로 각 근로자 별로 월급여액의 년도 별 증감 추이를 그래프로 나타내어라

강사님 방법

```
emp<-read.csv("고용형태별_취업자현황_2007_2013년.csv",stringsAsFactors=F)
```

월급여액 컬럼추출 - .4로 끝나는 것!! 패턴 찾는 함수 **str_detect**

```
library(stringr)
```

```
emp2<-emp[,str_detect(colnames(emp),'[.4]$')]
```

근로자별 정보가 컬럼화 되기 위해 전치 - 그 전에 **rownames()** 지정 필요

```
rownames(emp2)<-emp[,1]
```

전치를 위한 rownames 설정

```
total<- t(emp2[-1,])    # 원하던 형태! but ',' 때문에 숫자가 문자로 인식됨, 그래프 적용 불가  
                        # 첫 번째 행 필요 없음!
```

',' 제거 예시

```
ex1<- c('1,111', '2,222')
```

```
str_replace(ex1, ",", "")          #str_replace 벡터 연산 기능이 중요! 0!
```

#but 모든 데이터프레임에 적용할 때엔 **apply ~ c(1,2) 개별전달**

```
total2<-apply(total, c(1,2), str_replace, ',', '')
```

cf. 벡터별 연산 되지 않는 함수를 apply를 통해 1(행), 2(열)로 한다면 불가능..

- 파이썬은 문자열 치환함수가 절대 벡터연산이 안 됨! 하나씩 받고 치환 (1:1 치환만 가능)
- col별, row별 전달 불가, 원소 하나씩 전달하는 적용함수를 선택해야 함. **벡터의 원소별 전달 X**

+ **apply -> data를 벡터 형태로 적용 절대 불가능!**

```
total3 <- apply(total2, 2, as.numeric)          # as.numeric 역시 벡터 연산 가능
```

★★★★ **sapply로 데이터프레임을 부르면!! -> 결과가 벡터 / 리스트로 나와서 문제생김!**

=> sapply[,] 이런 식으로 벡터로 불러서 수정하면 문제 X!

/ 따라서 간단하게 쓸 수 있고 결과가 행렬로 나올 수 있는 apply 사용!!!★★★★

```

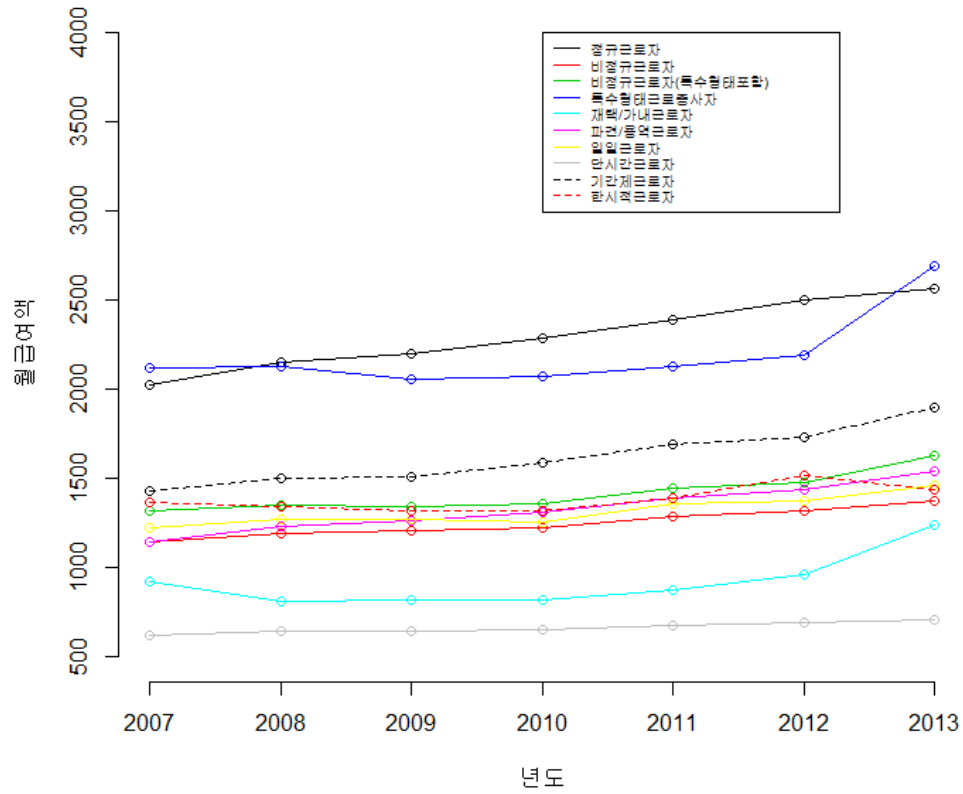
plot(total3[,3], type="o", col=1, ylim=c(500,4000), ann=F, axes=F)
lines(total3[,4], type="o", col=2)
lines(total3[,5], type="o", col=3)
lines(total3[,6], type="o", col=4)
lines(total3[,7], type="o", col=5)
lines(total3[,8], type="o", col=6)
lines(total3[,9], type="o", col=7)
lines(total3[,10], type="o", col=8)
lines(total3[,11], type="o", col=9, lty=2)      #col는 8 이후로 다시 1부터 반복->선 모양으로 구
분
lines(total3[,12], type="o", col=10, lty=2)

axis(1, at=1:nrow(total3), lab= str_sub(rownames(total),2,5))
axis(2, ylim=c(500, 4000))
total3
title(main="근로자별 월급여액 증감추이", xlab="년도", ylab="월급여액")
legend(4, 4000, colnames(total3[,3:ncol(total3)]), col=c(1:10), cex=0.6, lty=c(rep(1,8),rep(2,2)))

```

[1] [2] 는 전체 데이터여서 빼고

근로자별 월급여액 증감추이



강사님 형식으로 잠깐 간단히 다시 해보기 -- 다시 해보기!!

```
e1<-read.csv("고용형태별_취업자현황_2007_2013년.csv",stringsAsFactors=F)
dd<-c()
for(i in 0:6){
  i <- i*9
  dd <-c(dd,i)
}
print(dd)
dd
ddd<-dd+6
e2<-e1[,c(1,ddd)][-1,]
library(stringr)
e2[,2:8]<-sapply(e2[,2:8],str_replace","", "")
e2[,2:8]<-sapply(e2[,2:8],as.numeric)
# 여기 다시 하다가 문제 생긴 이유!!
```

★★★★ sapply로 데이터프레임을 부르면!! -> 결과가 벡터 / 리스트로 나와서 문제생김!

=> sapply[,] 이런 식으로 벡터로 불러서 수정하면 문제 X!

/ 따라서 간단하게 쓸 수 있고 결과가 행렬로 나올 수 있는 apply 사용!!!★★★★

```
rownames(e2)<-e2[,1]
plot_emp<-t(e2[-1])
```

#그리고 1, 2는 전체데이터여서 넣지 않음

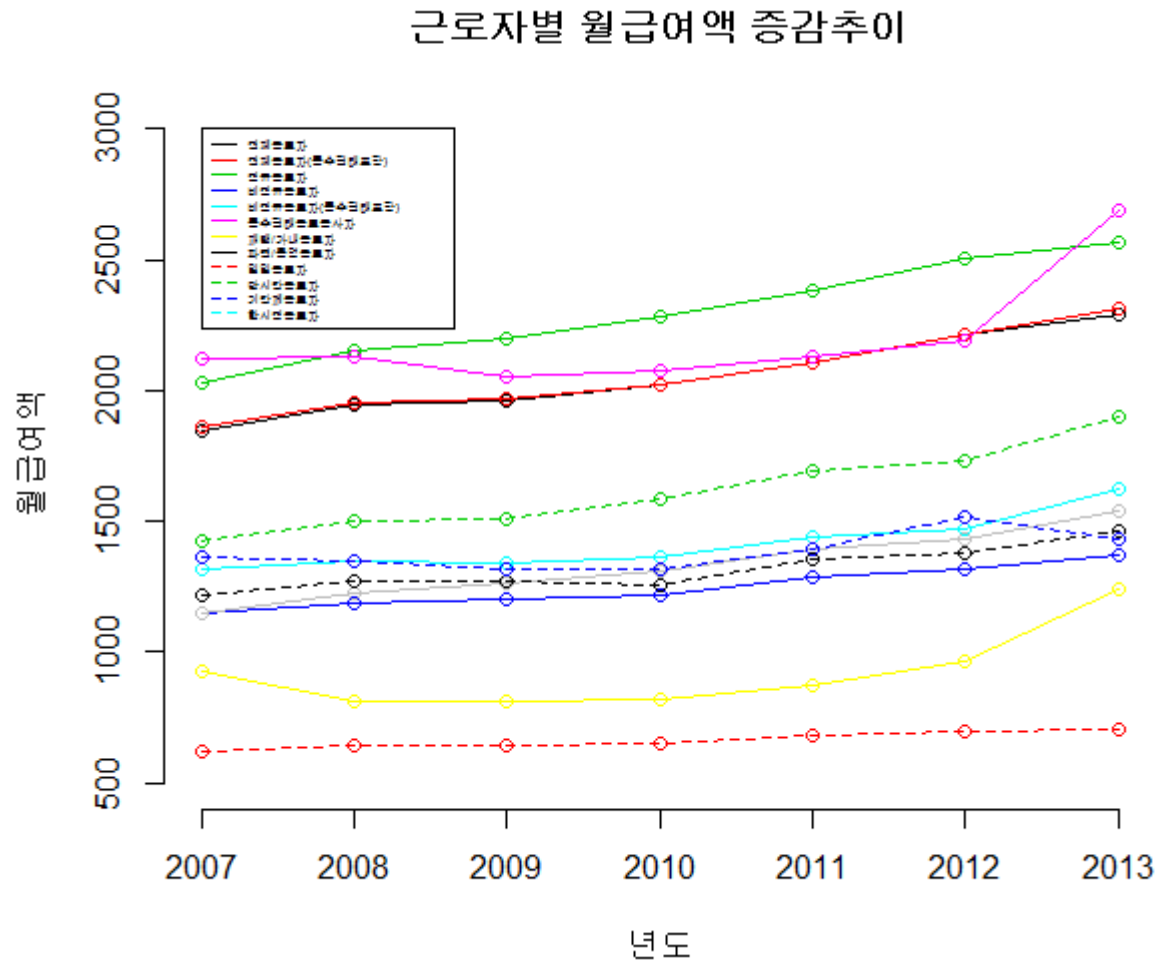
```
plot(plot_emp[,1], type="o", col=1, ylim=c(500,3000), ann=F, axes=F)
lines(plot_emp[,2], type="o", col=2)
lines(plot_emp[,3], type="o", col=3)
lines(plot_emp[,4], type="o", col=4)
lines(plot_emp[,5], type="o", col=5)
lines(plot_emp[,6], type="o", col=6)
lines(plot_emp[,7], type="o", col=7)
lines(plot_emp[,8], type="o", col=8)
lines(plot_emp[,9], type="o", col=9, lty=2)      #col는 8 이후로 다시 1부터 반복->선 모양으로
구분
lines(plot_emp[,10], type="o", col=10, lty=2)
lines(plot_emp[,11], type="o", col=11, lty=2)
lines(plot_emp[,12], type="o", col=12, lty=2)
axis(1, at=1:nrow(plot_emp), lab=c(2007:2013))
```



```
axis(2, ylim=c(500, 3000))
```

```
title(main="근로자별 월급여액 증감추이", xlab="년도", ylab="월급여액")
```

```
legend(1, 3000, colnames(plot_emp), col=c(1:nrow(plot_emp)), cex=0.4, lty=c(rep(1,8),rep(2,4)))
```



나

```
e1<-read.csv("고용형태별_취업자현황_2007_2013년.csv",stringsAsFactors=F)
str(e1)
ncol(e1)
# 여러개가 섞여있어서 월급여액 col만 빼기로 함. 숫자 계산 귀찮아서 함수돌림
dd<-c()
for(i in 0:6){
  i <- i*9
  dd <-c(dd,i)
}
print(dd)
dd
ddd<-dd+6
# 월급여액만 해당하는 열
```

```
e1[,c(1,ddd)]
e2<-e1[,c(1,ddd)][-1,]
#첫번째 행 필요 없어서 지움 - 월급여액 문자열
str(e2)
#월급여액들의 데이터타입이 숫자가 아닌 문자였음 (왜냐면 ",", " 때문에)
library(stringr)
e2[,2:8]<-sapply(e2[,2:8],str_replace,",","")
#먼저 숫자열 "," 지워줌
e2[,2:8]<-sapply(e2[,2:8],as.numeric)
#각 열별로 숫자형태로 만들어줌
str(e2)
```

```
e3<-melt(e2,id.vars="고용형태")
#년도를 열로 만들기 1단계
```

```
e4<-dcast(e3, variable ~ 고용형태)
#년도를 열로 만들기 2단계
```

```
str(e4)
#혹시나 다시 확인
```

```
plot(e4$기간제근로자, type="o", col=1, ylim=c(500,3000), ann=F, axes=F)
lines(e4$단시간근로자, type="o", col=2)
lines(e4[,4], type="o", col=3)
lines(e4[,5], type="o", col=4)
lines(e4[,6], type="o", col=5)
lines(e4[,7], type="o", col=6)
lines(e4[,8], type="o", col=7)
lines(e4[,9], type="o", col=8)
lines(e4[,10], type="o", col=9, lty=2)
#col은 8 이후로 다시 1부터 반복->선 모양으로 구분
lines(e4[,11], type="o", col=10, lty=2)
```

```
lines(e4[,12], type="o", col=11, lty=2)
```

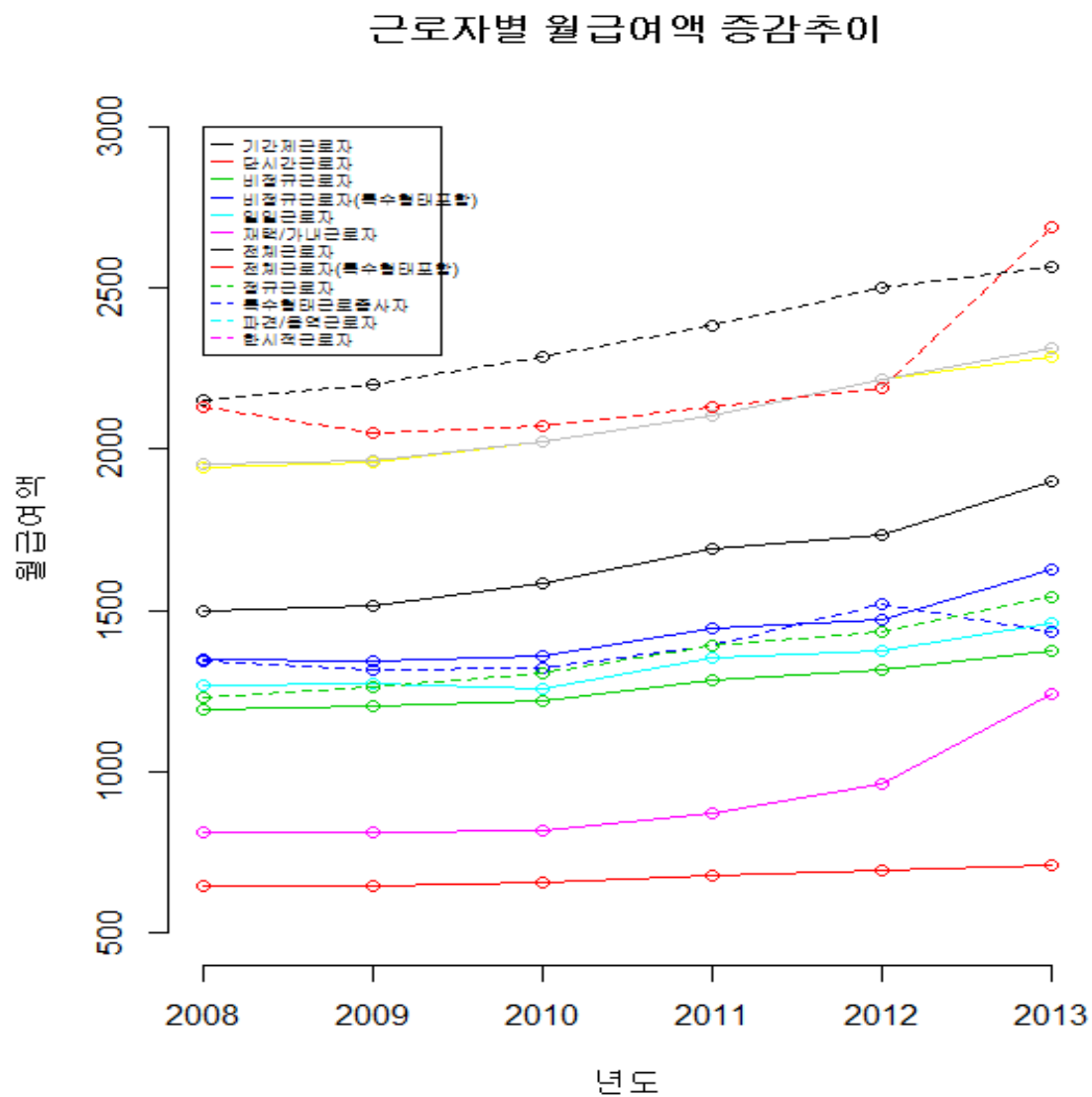
```
lines(e4[,13], type="o", col=12, lty=2)
```

```
axis(1, at=1:nrow(e4), lab=c(2008:2013))
```

```
axis(2, ylim=c(500, 3000))
```

```
title(main="근로자별 월급여액 증감추이", xlab="년도", ylab="월급여액")
```

```
legend(1, 3000, colnames(e4)[-1], col=c(1:nrow(e4)), cex=0.55, lty=c(rep(1,8),rep(2,4)))
```



느낀점

1. str() 데이터 타입 확인 꼭 하고 시작!

-> 처음 read. ~~ 옵션에서 skip=행번호 로 불러도 됨!

2. melt + dcast 세트 엄청 잘 쓰인다. 행 이름을 하나의 열로 바꿀 때!

-> rownames 설정하고 [-1] 형태로 전치 t() !!!!! + apply 관련 함수 적용!!

3. colnames(~~~)[-1] 형태도 잘 쓰인다. -> 범례 쓸 때 하나하나 안 쓰고!

— melt + dcast 이후 variable을 뺀 나머지의 행이름으로 범례 작성

- > 전치 t() 하면 첫열 안 빼도 바로! [1] 1열부터 시작할 수 있음

4. nrow, ncol, length 잘 쓰인다. 직접 세지 말고 함수를 활용!

5. apply 굉장히 많이 쓰이는 군!!

cf. sapply 하려면 [,] 형태로 벡터를 고집어내서 벡터로 바꿔주게 해야 함.

데이터 프레임을 sapply로 불러오면 결과 -> 벡터!

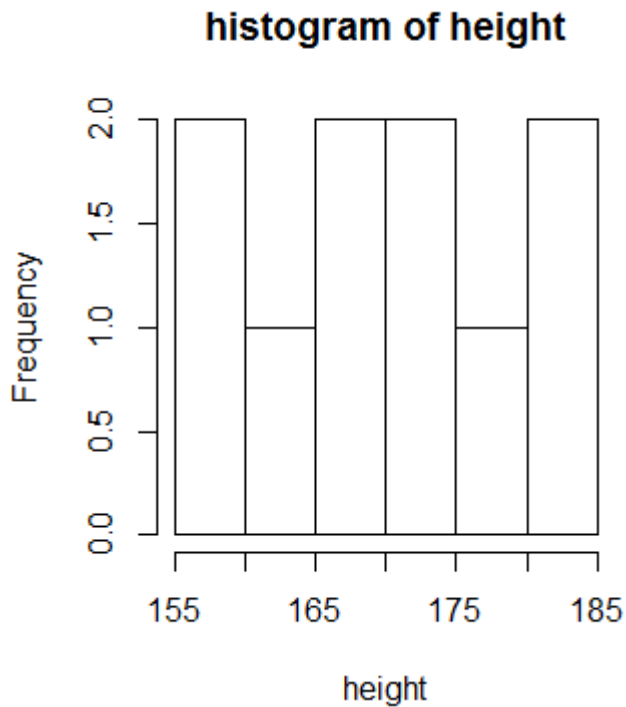
2018.05.29 histogram

3) 히스토그램 그래프 그리기 : hist()

- 빈도수 자동 출력
- 숫자빈도수만 출력되기 때문에 잘 안 쓰임 (문자빈도수 -> barplot)

```
height<-c(182,175,167,172,163,178,181,166,159,155)
```

```
hist(height,main="histogram of height")
```

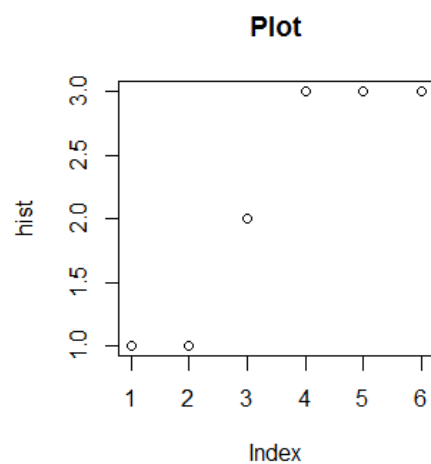
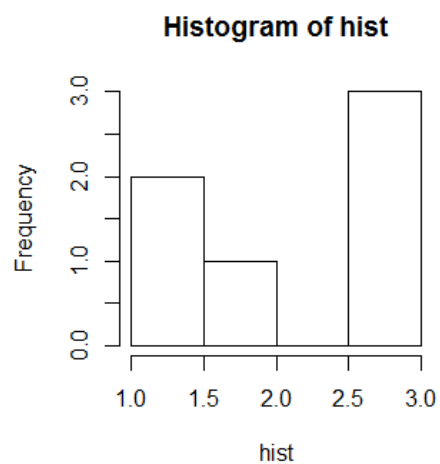


```
par(mfrow=c(1,2), oma=c(2,2,0.1,0.1))
```

```
hist<-c(1,1,2,3,3,3)
```

```
hist(hist)
```

```
plot(hist,main="Plot")
```

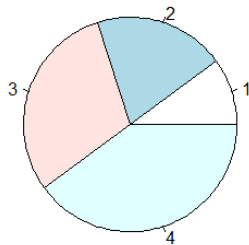


4) 파이(pie)모양의 차트 그리기 : pie()

인수	기능
angle, density, col	파이부분을 구성하는 각도,기울기(angle) / 수 (density) / 색상(col) 지정
labels	각 pie부분의 이름을 지정하는 문자벡터 지정
radius	원형의 크기 지정
clockwise	시계방향(T)으로 회전할 지 반시계방향(F)로 회전할 지 지정. 기본은 반시계
init.angle	시작되는 지점의 각도지정

(1) 기본적인 pie chart

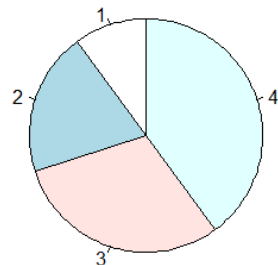
```
par(mfrow=c(1,1),oma=c(0.5,0.5,0.1,0.1))
p1<-c(10,20,30,40)
pie(p1,radius=1)
```



/ 라벨 없으면 자동으로 1, 2, 3, 4, ...

(2) 시작 각도를 90도로 지정하기

```
pie(p1, radius=1, init.angle = 90)
```



(3) 색깔과 label명 지정하기

```
pie(p1, radius=1, init.angle=90, col=rainbow(length(p1)), label=c("W1","W2","W3","W4"))
```



파이차트는 바로 라벨링! /

cf. 각 portion마다 색 기준 하고 싶으면 사용자지정함수+빈벡터 -> 팔레트 -> col 지정

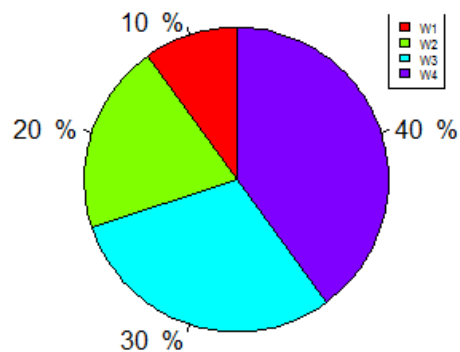
(4) 수치 값을 함께 출력하기

```
pct<-round(p1/sum(p1)*100,1)
```

```
lab <- paste(pct, " %")
```

```
pie(p1, radius=1, init.angle=90, col=rainbow(length(p1)), label=lab)
```

```
legend(1,1.1,c("W1","W2","W3","W4"),cex=0.5,fill=rainbow(length(p1)))
```



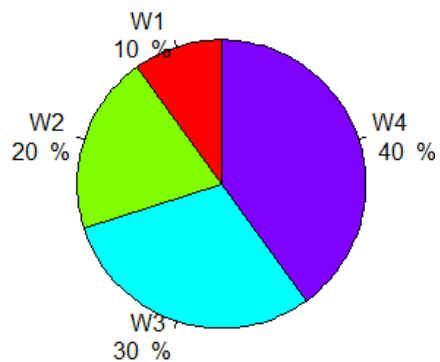
(5) 범례를 생략하고 그래프에 바로 출력하기

```
pct<-round(p1/sum(p1)*100,1)
```

```
lab1 <- c("W1","W2","W3","W4")
```

```
lab2 <- paste(lab1,"Wn",pct," %") # "Wn" <- 엔터를 나타내줌!! / paste - concate()
```

```
pie(p1, radius=1, init.angle=90, col=rainbow(length(p1)), label=lab2)
```



6) pie3D() 함수 -- 패키지설치 필요

```
install.packages("plotrix")
```

```
library(plotrix)
```

```
p2<-c(10,20,30,40,50)
```

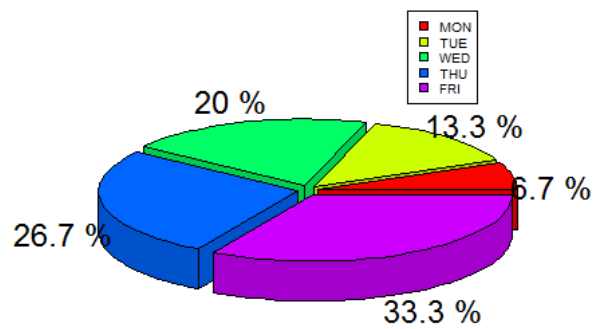
```
f_day<-round(p2/sum(p2)*100,1)
```

```
f_label<-paste(f_day,"%")
```

```
pie3D(p2, main="3D Pie Chart", col=rainbow(length(p2)),cex=0.5, labels=f_label, explode=0.05)
```

```
legend(0.5, 1, c("MON","TUE","WED","THU","FRI"),cex=0.4, fill=rainbow(length(p2)))
```

3D Pie Chart



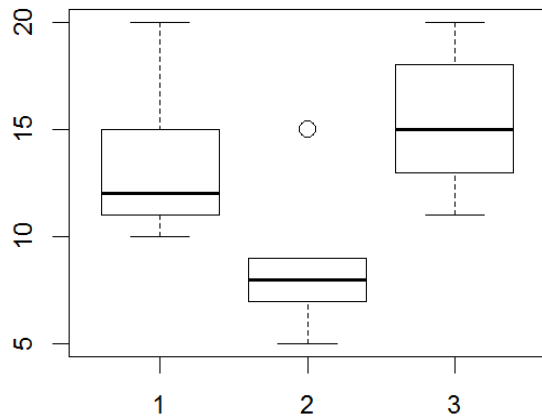
7) 상자차트 - boxplot()

```
v1<-c(10,12,15,11,20)
```

```
v2<-c(5,7,15,8,9)
```

```
v3<-c(11,20,15,18,13)
```

```
boxplot(v1,v2,v3)
```



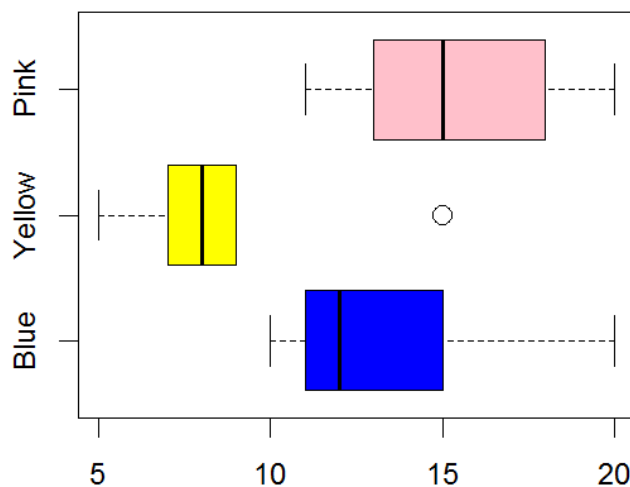
옵션	의미
col	박수 내부의 색깔 지정
names	각 막대의 라벨을 지정할 문자벡터를 지정
range	박스의 끝에서 수염까지의 길이를 지정. 기본은 1.5
width	박스의 폭을 지정
notch	TRUE로 지정할 경우 상자의 허리 부분을 가늘게 표시
horizontal	TRUE로 지정할 경우 상자를 수평으로 그림. 아래부터 차례로 나열됨.

```
v1<-c(10,12,15,11,20)
```

```
v2<-c(5,7,15,8,9)
```

```
v3<-c(11,20,15,18,13)
```

```
boxplot(v1,v2,v3, col=c("blue","yellow","pink"), names=c("Blue","Yellow","Pink"), horizontal=T)
```



연습문제

1. 일자별 총 통화건수와 전일대비 증감률을 구하여라

강사님

```
dd<-read.csv("delivery.csv",stringsAsFactors = F,  
colClasses = c(시간대="character",일자="character"))
```

cf. colClasses 옵션 ~ 원하는 컬럼 데이터타입 불러올 때 설정 가능

시간대 컬럼 숫자로 인식 -> 불러올 때 문자열로 인식하도록 변경

일자 컬럼 숫자 -> 문자 -> 날짜로 변경해야 함!

그래서 불러올 때 먼저 문자열로 인식하도록 변경

특정 컬럼만 데이터 타입 변경하고 싶을 땐 벡터형식으로 전달해야 함

```
library(plyr)
```

```
ddply(dd,.(일자), summarise, total=sum(통화건수))
```

#ddply - 적용할 수 있는 옵션이 많음!

사용자 정의함수를 통한 이전 데이터 불러오기

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

```
  test<-c()
```

```
  for (i in 1:length(x)){
```

```
    test<-c(test, x[max(i-1,1)])
```

```
  }
```

```
  return(test)
```

```
}
```

#사용자 정의함수 - 벡터 전달받을 건지, 원소 하나하나를 받을 건지 생각해야 함 - 형태 달라짐

#if문 안 쓰고 **max(i-1, 1)** 형태로 쓰면 오류 안 뜨고 간단하게 할 수 있음!(i=1이면 오류)

#함수는 **return**으로 객체 활성화!!!! (not print!) - 잊지 말기!

cf. 만일 전전 행을 부르려면 x[max(i-2,1)] 형태로 부를 수 있음

-----**shift 통해서도 가능함**-----

shift(x, n=1L, fill=NA, type=c("lag", "lead"), give.names=FALSE)

type = "lag" - 이전 정보 가져오기 (기본값) / "lead" - 이후 정보 가져오기

fill - 처음 or 끝 (연산X) - 값을 무엇으로 채울 지

#help(shift) 기본패키지 아니어서 안 뜸, ??shift로 검색하면 함수 포함된 패키지 정보 뜸

data.table 패키지의 shift를 사용한 이전 값 가져오기

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

```
shift(d1$total, n=1, fill=0)
```

```
d1$before<-f_shift(d1$total)
```

전일대비 증감률 구하기

$(d1\$total - d1\$before) / d1\$before * 100$

내가 푼 방법

```
dd<-read.csv("delivery.csv",stringsAsFactors = F)
str(dd)
head(dd)
```

#1) 총 통화 건수와 전일대비 증감률

1. 일자별 총 통화 건수

```
dd2<-aggregate(통화건수 ~ 일자, dd, sum)
```

```
#####
```

```
library(plyr)
ddply(dd, .(일자), summarise, sum(통화건수))
#####
```

#2. 전일대비 증감률

증감률 = (오늘 - 어제) / 어제 * 100

-----고민해보니까 된당!-----

```
dd2<-aggregate(통화건수 ~ 일자, dd, sum)
```

```
sal<-c()
pmpm<-function(x){
  for(i in 2:length(x)){
    sal<-c(sal,(x[i]-x[i-1])/x[i-1]*100)
  }
  return(sal)
}
```

#1은 i-1하면 없으니까..! 원래 안 생기니까 2부터 함

```
pmpm(dd2[,2])
```

-----안 되어서 했던 것,,따쉬-----

```
tod<-dd2[,2][-1]
yesterd<-dd2[,2][1:length(tod)]
```

```
(tod-yesterd)/yesterd*100
```

부족한 흔적들..따쉬

1-1~~~~~따쉬 너무 갔ㄷ....ㅏ 그래프를 그리라곤 안 했음

```
rownames(dd2)<-dd2[,1]
```

```
dd3<-t(dd2)[-1,]
```

```
dd3
```

```
plot(dd3,type="o",col=1,axes=0,ann=F, ylim=c(0,60000))
```

```
axis(1, at=1:length(dd3))
```

```
axis(2, ylim=c(0,60000))
```

```
title(main="2월 일자별 총 통화건수", xlab="일", ylab= "통화건수")
```

~~~~~

~~~~~

1-2 함수로 뭔가 해보려 했으나.. 실패!

```
aaa<-c()
```

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

```
  if(i==1)
```

```
    for(i in 1:NROW(x)){
```

```
      if(i==1){
```

```
        aaa<-c(aaa,0)
```

```
      }
```

```
      else{
```

```
        aaa<-c(aaa,(x[i]-x[i-1])/x[i-1]*100)
```

```
      }
```

```
      return(aaa)
```

```
    }
```

```
  }
```

```
aa(dd2[,2])
```

~~~~~

2. 요일별로 각 업종별 통화건수를 확인하고 막대그래프로 표현하여라

### 강사님 방법

-----요일 표시 방법-----

1. lubridate 패키지 사용- **wday + label=T**

```
library(lubridate)
```

```
wday(as.Date(dd$일자, format="%Y%m%d"), label=T)
```

2. **as.character()** 사용 - 표현식 **"%a"**

```
as.character(as.Date(dd$일자, format="%Y%m%d"), "%a")
```

-----

```
dd$요일 <- as.character(as.Date(dd$일자, format="%Y%m%d"), "%a")
```

```
library(plyr)
```

```
data2 <- ddply(dd, .(요일, 업종), summarise, total=sum(통화건수))
```

```
library(reshape2)
```

```
data3<-dcast(data2, 업종 ~ 요일)
```

#변수화 시킨다는 건 메모리를 계속 사용 -> 속도 문제.. !!! 좋은 표현식이 아님.

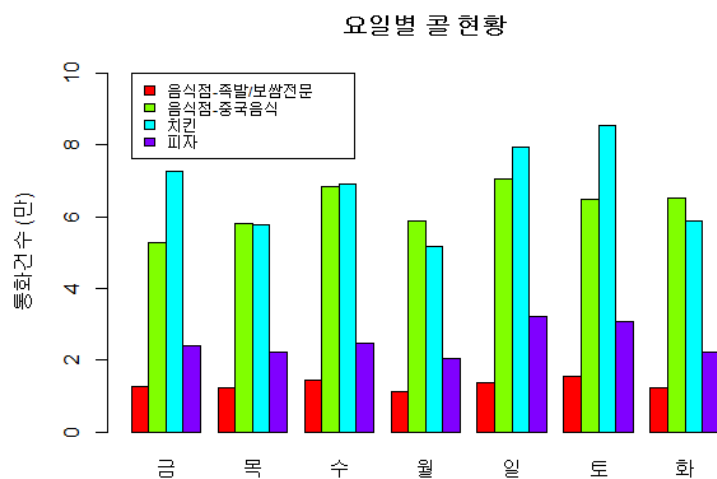
한 라인에 짝 표현하는 법이 더 추천..! in python, R 모두!

# 아니면 만들고 기존 객체 지우기!

```
barplot(as.matrix(data3[,-1])/10000, beside=T, col=rainbow(4), ylim=c(0,10))
```

```
title(main="요일별 콜 현황", ylab="통화건수 (만)")
```

```
legend(1,10,data3[1,],fil=rainbow(4),cex=0.7)
```



## 내가 푼 방법

```
dd$일자<-as.Date(as.character(dd$일자),"%Y%m%d")
```

```
library("lubridate")
```

```
dd$요일<-wday(dd$일자,label=T)
```

```
library(plyr)
```

```
yus<-ddply(dd,.(요일, 업종), summarise, 통화합=sum(통화건수))
```

```
library(reshape2)
```

```
bar_total<-dcast(yus,업종 ~ 요일)
```

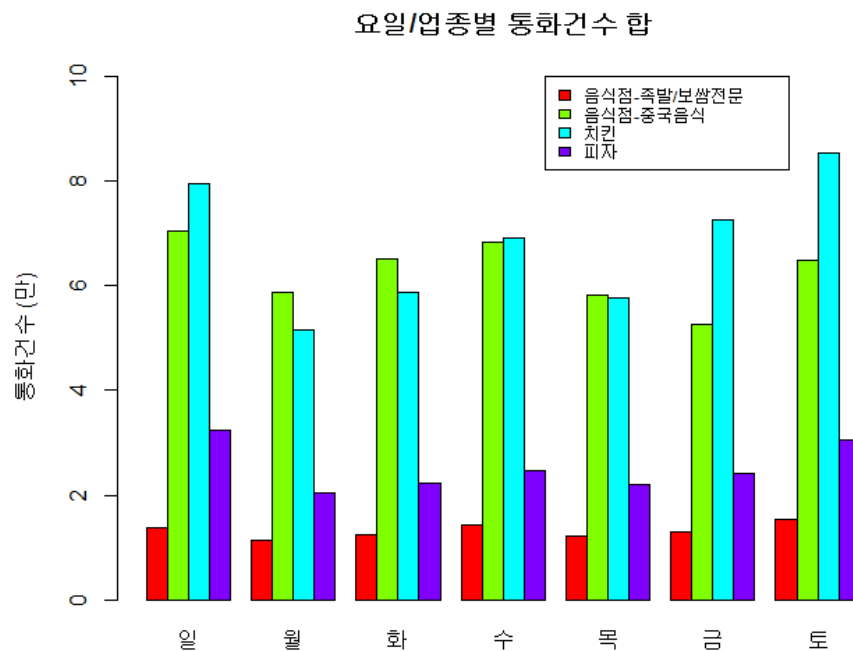
```
rownames(bar_total)<-bar_total[,1]
```

```
bar_total2<-as.matrix(bar_total[,-1])
```

```
barplot(bar_total2/10000, beside=T, col=rainbow(nrow(bar_total2)),ylim=c(0,10))
```

```
legend(20,10,rownames(bar_total2),fill=rainbow(nrow(bar_total2)),cex=0.7)
```

```
title(main="요일/업종별 통화건수 합", ylab="통화건수 (만)")
```



### 3. 요일별 전체 통화건수를 파이 차트로 표현하여라.

앞에서 한 data2 (요일 / 업종별 sum) 사용

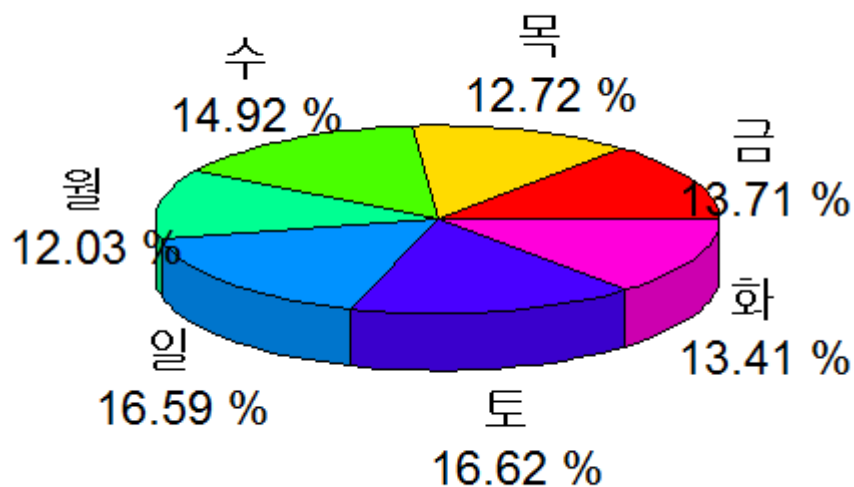
```
data4<-ddply(data2, .(요일), summarise, v1=sum(total))
```

```
library(plotrix)
```

```
pct<-round(data4[,2]/sum(data4$v1)*100,2)
```

```
pie3D(data4[,2], col=rainbow(7), labels=paste(data4$요일,"₩n",pct,"%"))
```

# "₩n" : 엔터의 뜻



#### 내 방법

```
dd$일자<-as.Date(as.character(dd$일자),"%Y%m%d")
```

```
library("lubridate")
```

```
dd$요일<-wday(dd$일자,label=T)
```

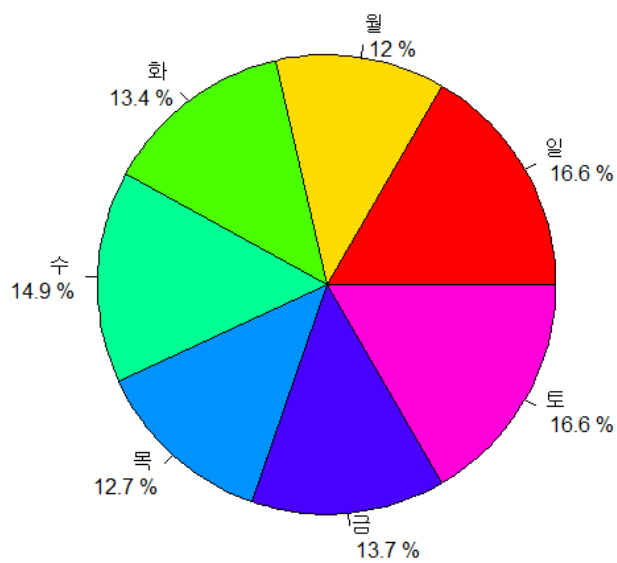
```
daysum<-aggregate(통화건수 ~ 요일, dd, sum)
```

```
barplot(daysum[,2], ylim=c(0,230000), col=rainbow(7), names.arg=daysum[,1])
```

```
legend(1.5,230000,daysum[,1],fill=rainbow(7),cex=0.7)
```

```
pct<-round(daysum[,2]/sum(daysum[,2])*100,1)
```

```
pie(daysum[,2],radius=1,col=rainbow(7), label=paste(daysum[,1],"₩n",pct,"%"))
```

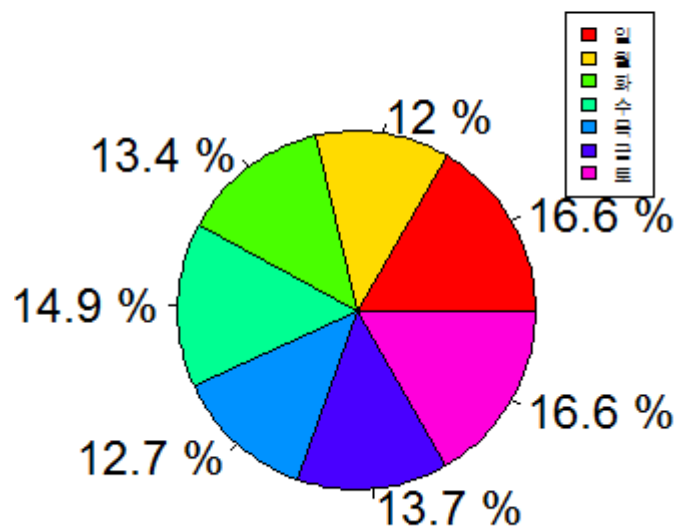


+222222222222222222

pie(daysum[,2],radius=0.6,col=rainbow(7), label=paste(pct,"%"),main="요일별 통화건수")

legend(0.7,1,daysum[,1],fill=rainbow(7),cex=0.4)

## 요일별 통화건수



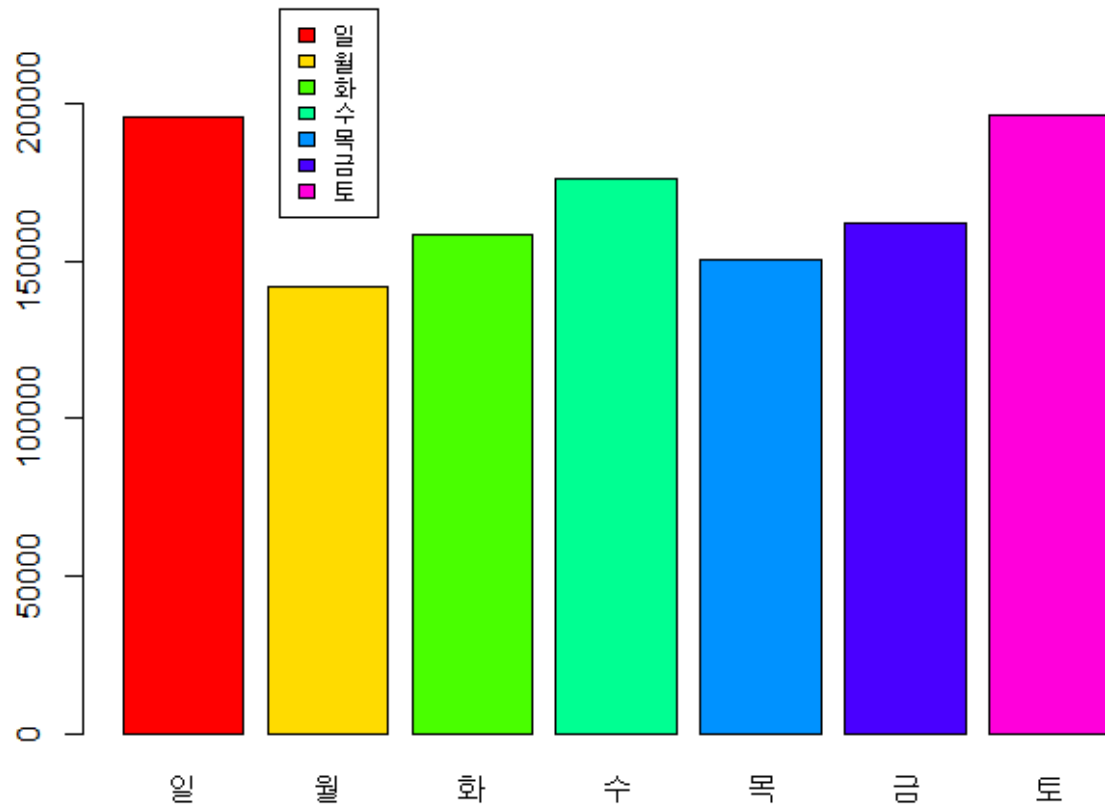


(((((((문제 잘못봤다...))))))

```
daysum<-aggregate(통화건수 ~ 요일, dd, sum)
```

```
barplot(daysum[,2], ylim=c(0,230000), col=rainbow(7), names.arg=daysum[,1])
```

```
legend(1.5,230000,daysum[,1],fill=rainbow(7),cex=0.7)
```



#### 4. 시간대별로 가장 인기있는 음식업종을 출력하여라

```
data5<-ddply(dd, .(시간대, 업종), summarise, total=sum(통화건수))
```

```
ddply(data5, .(시간대), subset, total==max(total))
```