목차

1. R 설치 및 기본 사용법 ····· 1
2. 변수사용법 데이터 처리 유형 ····· 7
3. R에서 데이터유형 ····· 12
4. plor과 boxplot ····· 17
- [미션] plot ····· 22

5. 상관계수 및 패키지 설치 ····· 24
6. 데이터 전처리(그룹합) & plot ····· 27
- [미션] 데이터 전처리 & plot ····· 29

7. 데이터 전처리(NA, 필터) ····· 33

8. 사용자정의 함수 ····· 44
9. 문자열 처리 패키지(stringr) ····· 45
- [미션1]~[미션2] ····· 49

10. 트리맵 패키지 활용 ····· 50
- [미션1]~[미션4] ····· 52

11. 텍스트마이닝(한글) ····· 53
12. 텍스트마이닝(영어) ····· 61

13. 공공데이터 & 지도차트 ····· 66

12. 연관분석 ····· 69

1	변수사용법과 많은 데이터 처리 방법 배우기 R 설치/R에서 사용하는 다양한 데이터 유형등	

1. R 이란?

가) R의 탄생

- 오픈소스 프로그램으로 통계, 데이터마이닝과 그래프를 위한 언어이다.
- 다양한 최신 통계분석과 마이닝 기능을 제공한다.
- 전세계적으로 사용자들이 다양한 예제를 공유한다.
- 다양한 기능을 지원하는 5000개에 일는 패키지가 수시로 업데이트 된다.
- 윈도우, 맥, 리눅스 운영체제 모두 사용가능하다.
- 객체지향 언어를 사용하고 있어 깔끔하고 빠른 코드 수행 속도를 지닌다.

나) 분석도구 비교

항목	SAS	SPSS	오픈소스R
프로그램비용	유료, 고가	유료,고가	오픈소스
설치용량	대용량	대용량	모듈화로 간단
다양한 모듈 지원 및 비용	별도구매	별도구매	오픈소스
최근 알로리즘 및 기술반영	느림	다소느림	매우빠름
학습자료 입수의 편의성	유료 도서 위주	유료 도서 위주	공개 논문 및 자료다수
질의를 위한 공개 커뮤니티	NA	NA	매우 활발

나) R실행 프로그램은 R과 R스튜디오가 있으나 대량 데이터 처리는 R이 더 적합함

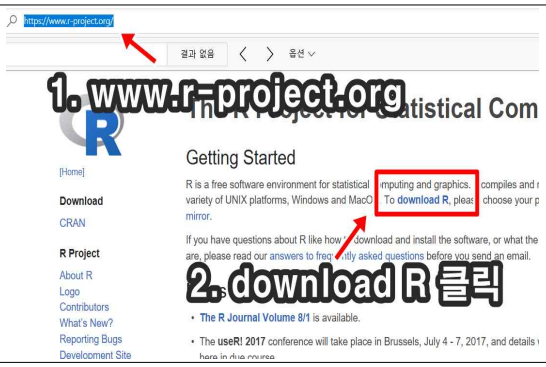
2. R 프로그램 설치법

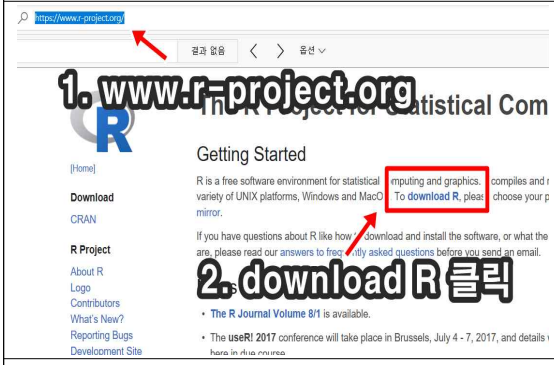
1. 현재 윈도우가 32비트인지 64비트인지 확인

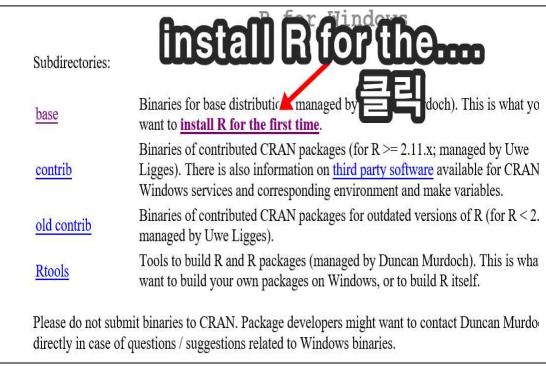
2. http://www.r-project.org 에서 32비트 또는 64비트 컴퓨터에 맞는 R 프로그램 다운로드

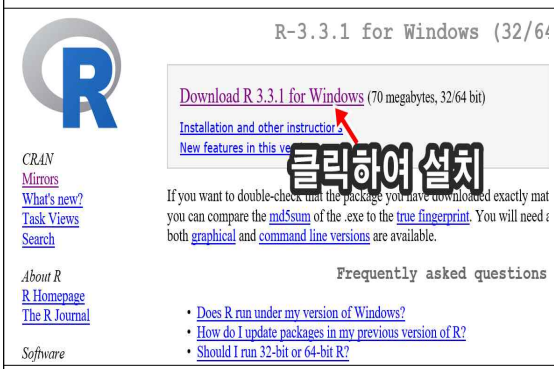
R 동작 기본환경
- 개인용: Win7 32bit·64bit, RAM 4G(8G권장) / 기업용: Linux64bit Dual Core, RAM 32G, Disk 2T 이상권장

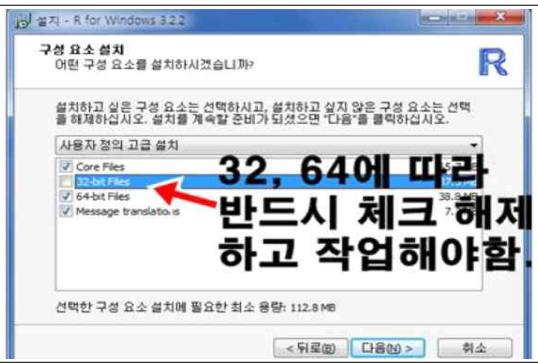
만약 jvm.dll이 없다고 에러가 나오면 java관련 자료 다운로드 받아 설치함.
설치 경로에 한글이 있는 경우 Package가 제대로 설치되지 않는 경우가 있음. 이럴 경우 현재 윈도우에 로그인 되어 있는 계저으이 이름을 영문으로 바꾸어 설치







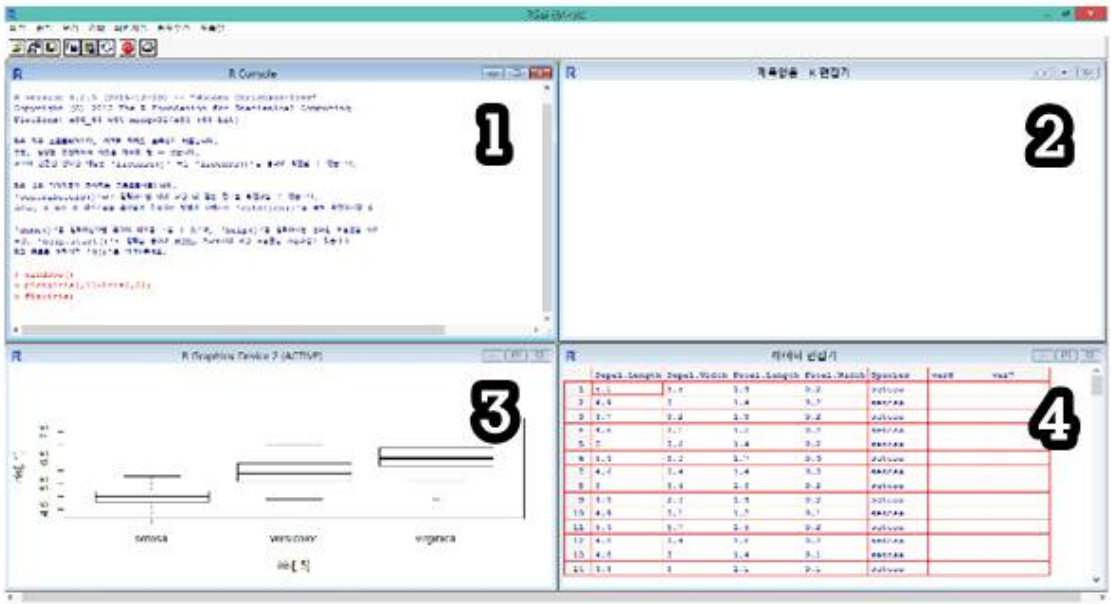




* R실행시 마우스 우측 클릭하여 [관리자 권한으로 실행] 함.

3. R의 기본 사용법 및 데이터유형

【R 작업창 이해】



R은 기본적으로 세 개의 창(R Consloe, R Editor, R Graphics)으로 이루어져 있으며 이후 사용자의 편의를 위하여 추가로 Data Editor창을 부를 수 있다.

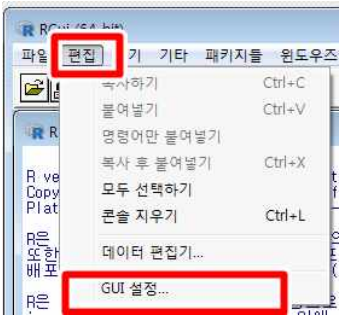
① R Console : R 명령문을 입력하고 실행시키게 된다.
② R Editor : R 명령문 작성과 수정할 수 있다.
③ R Graphics : 사용자가 만든 그래프가 그려져 출력된다.
④ Data Editor : 사용자가 데이터를 수정할 수 있다. 주로 분석에 사용되는 데이터를 확인하기 위해 사용되어진다.

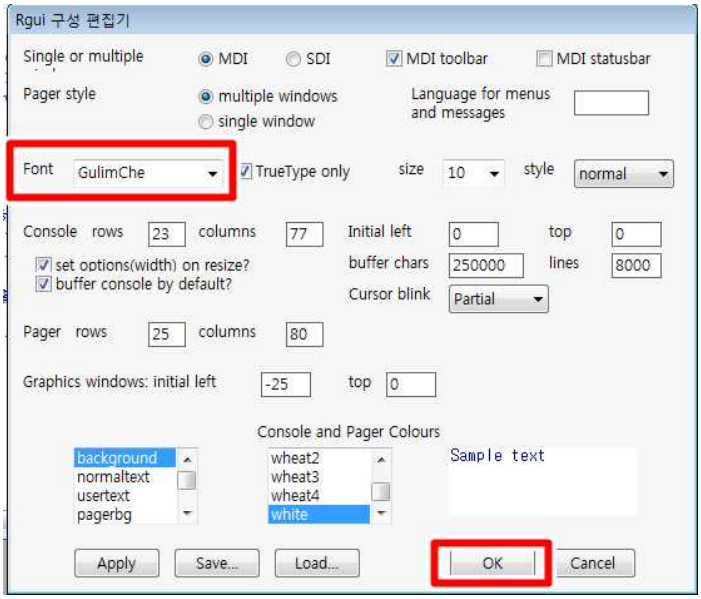
<< 구글 검색 >>

filteyp:hwp R설치
filetype:pdf r 기본
plot in r => R에서 plot와 연관된 자료를 찾기

【R 작업 글꼴 변경】

R실행후
[편집]
-[GUI]설정에서
작업환경 설정





【R 프로그램 특징】

- > 표시가 나와야 명령 입력가능한 인터프리터 언어이다.
- + 표시가 나오면 명령을 이어가거나 ESC로 명령 취소한다.
- > help("plot") help로 도움말을 얻을수 있다.
- > ?plot ? 로 도움알을 얻을수 있다.
- R에서는 ↑ ↓ 방향키로 코딩 작업을 반복한다.
- Ctrl+L 로 화면 삭제가능하다.
- R주석은 #을 입력후 작성한다.
- R 프로그램은 대소문자를 구별한다.

- **[파일]-[새 스트립트]를 열고 자료를 코딩한뒤 Ctrl+R 또는 F5를 누르면 커서가 있는 줄의 명령이 R 창에 프로그램으로 실행됨, 범위지정뒤 실행가능**

【변수작성법】

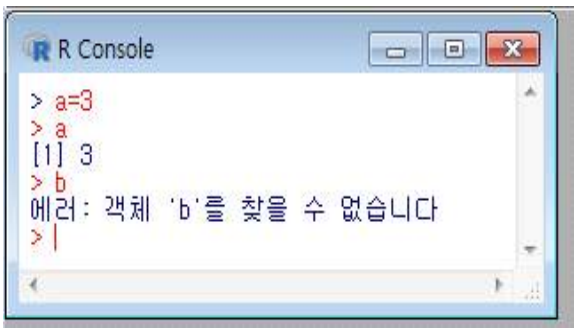
R에서 제공하는 명령어가 아닌 사용자가 필요에 의해서 작성하는 데이터 저장용 객체를 변수라 한다.

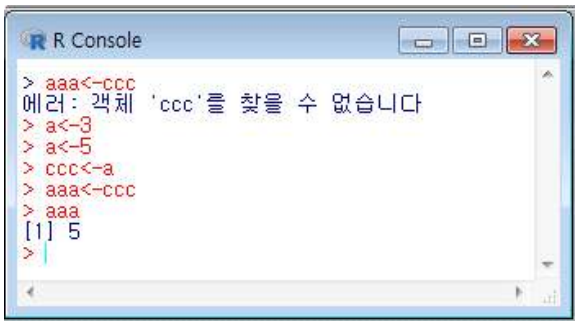
일반적으로 값을 변수에 넣어라는 '변수명=값' 의 형태로서 오른쪽의 값을 왼쪽의 변수명에 할당하는 의미이다.

R에서는 '변수명<-값' 또는 '변수명=값'으로 표시한다. R에서 많은 사용이 있는 통계부분에서의 변수와 구별하기 위하여 R에서는 변수를 객체라고 표현한다.

▪ 변수작성규칙은 아래와 같다.

- 영문(한글), 숫자, underbar(_), 점(.) 으로 구성된다.
- 첫글자는 숫자일수 없다.
- 띄어쓰기 불가능하다.





▪ 변수 관련 R 명령

ls()	#내가 만든 모든 변수(객체)를 확인
ls.str()	# 변수(객체) 할당 값 및 num(숫자), chr(문자)인지 보여줌
rm(객체명, 객체명)	# 변수(객체)삭제
rm(list=ls())	# 모든 변수(객체)삭제

R에서 사용가능한 데이터는 정수(3), 실수(3.5), 문자(“t”), 문자열(“tmp”), 논리값(TRUE,FALSE) 이 있다.

```
> print("Hello World")
[1] "Hello World"
> "Hello World"
[1] "Hello World"
> hello world
에러: 예상하지 못한 기호(symbol)입니다. in "hello world"
> 1*2*3*4
[1] 24
> factorial(4)
[1] 24
> "1*2*3*4"
[1] "1*2*3*4"
```

[실습] 파일-새스크립트를 실행한뒤 아래의 내용을 입력한다.

```
num<-c(1,2,3,4,5,6,7,8)
irum<-c("a","b","c","d","e","f","g","h")
m_f<-c("m","f","m","m","m","m","f","f")
age<-c(20,30,21,22,25,50,22,30)
num
irum

length(num)  #전체 데이터 개수를 보여줌
is(num)  #num변수의 성격을 보여줌
summary(num) #숫자데이터 기술통계

length(irum)
is(irum)  #irum 변수의 성격을 보여줌
summary(irum) #문자데이터 기술통계
irum[3])   #3행의 내용을 보여줌

as.factor(age) #age나이를 그룹 가능한 문자로 보여줌
f_age<-as.factor(age)   #age를 factor화 한후 f_age변수에 할당
summary(f_age)
is(f_age)
data<-data.frame(num,irum,m_f,age)  #데이터 프레임으로 변환
data
data[3,4] ; data[,3]; data[3,];   dim(data) #;으로  한줄에 여러개 명령입력
data$age  #데이터 프레임중 age변수만 보기
plot(data$age) ; barplot(data$age) ; boxplot(data$age)
```

2	변수사용법과 많은 데이터 처리 방법 배우기 변수 개념과 사용방법 배우기	

1. 변수란

'변수명=값' 의 형태로서 오른쪽의 값을 왼쪽의 변수명에 할당하는 의미이다. R에서는 '변수명<-값'으로 표시한다. R에서 많은 사용이 있는 통계부분에서의 변수와 구별하기 위하여 R에서는 변수를 객체라고 표현한다.

오른쪽의 값은 3과 같은 숫자, "a"와 같은 문자, tmp와 같은 변수, 계산식등 다양한 값을 넣을수 있다.

▶ 변수 규칙

. 영문(한글), 숫자, underbar(_), 점(.) 으로 구성된다.
. 첫글자는 숫자일수 없다.
. 띄어쓰기 불가능하다.

잘못된예
올바른예

변수	범주형변수	명목형변수	문자 변수로서 성별, 혈액형등의 카테고리(level)를 가짐 (남,남,남자 는 X)
		순서형변수	순서가 있는 변수 A+,A-,A,B+
	수치형변수	이산형변수	셀 수 있는 정수이며 값들이 서로 이어져 있지 안음. 이항분포를 따르는 확률질량함수를 만들 수 있음.
		연속형변수	변수값이 연속적인 수치. 170.2는 170~171 사이에 무수히 많은 값이 존재하므로 어떤순간을 띄어낼수 없음 독립적이지 않음. 확률밀도함수와 연관됨

2. 변수사용예

가) 변수사용하기

`> aaa`
`에러: 객체 'aaa'를 찾을 수 없습니다`
`> aaa<-5`
`> aaa`
`[1] 5`
 	
`> aaa<-bbb`
`에러: 객체 'bbb'를 찾을 수 없습니다`
`> bbb<-6`
`> aaa<-bbb`
`> aaa`
`[1] 6`
`>`
	aaa에 설정된 값이 없으므로 변수 또는 객체를 찾을수 없음에 대한 메시지 나옴.
	프로그램에 따라 변수에 초기값이 할당되지 않으면 0으로 자동 세팅되는 프로그램도 있음.
	aaa<-bbb에서 bbb값의 초기값이 설정되지 않았으므로 bbb 객체를 찾을수 없음으로 에러
	bbb<-6을 초기값으로 주면
	aaa<-bbb는 aaa에 6을 할당함.
	aaa=5, aaa=bbb 사용가능

`> kor=20`
`> mat=30`
`> tot=kor+mat`
`> tot`
	Kor<-20 kor 변수에 20을 할당
	mat<-30 mat 변수에 30을 할당
	tot 변수에 20+30 , 50을 할당
	tot 50 출력

`> name="bu"`
`> juso="seoul"`
`> name+juso`
`Error in name + juso : 이항연산자에 수치가 아닌 인수입니다`
`> paste(name,juso)`
`[1] "bu seoul"`
`> paste(name,"(",juso,")")`
`[1] "bu ( seoul )"`
`>`
	name+juso 는 문자열+문자열 이므로 계산 불가임. 문자열은 서로 이어서 출력하는 연결 작업을 해야함.
	paste(name,juso)
	paste(name,"(",juso,")")

Tmp<-factorial(3)	변수에 함수를 할당하면 함수의 결과값이 들어감. tmp<-1*2*3
Tmp<-TRUE Tmp<-FALSE	변수에 논리값도 할당 가능하다.

▶ 아래의 command를 직접 작성하면?

. 키변수에 180 할당

. 몸무게변수에 70 할당

. 출력=> H:180, W:70

나) rep함수를 이용한 횟수만큼 출력물 나타나게 하기

“-,-” # - 표시를 20번 나타나게함. rep(x="-", times=20) # times에 들어온 숫자만큼 x값을 반복하자
함수 : rept(x=반복하고자 하는 문자, times=반복횟수) 의미 : x값을 times만큼 반복하여 벡터타입의 시퀀스로 반환

Tmp_Value<-“-” Tmp_Cnt<-20 rep(x=Tmp_Value, times=Tmp_Cnt)
위와 같이 작성한뒤 ↑를 이용하여 Tmp_Value값과 Tmp_Cnt값을 변경한뒤 ↓ 화살표로 rep 명령을 불러내어 재 실행함

▶ 아래 예문의 결과는?.

Tmp_Value<-“3+4” Tmp_cnt<-50 rep(x=Tmp_Value, times=Tmp_Cnt)	Tmp_Value<-3+4 Tmp_cnt<-50 rep(x=Tmp_Value, times=Tmp_Cnt)
Tmp_Value<-3+4 Tmp_cnt<-“3+4” rep(x=Tmp_Value, times=Tmp_Cnt)	Tmp_Value<-3+4 Tmp_cnt<-3+4 rep(x=Tmp_Value, times=Tmp_Cnt)

3) 변수관련 함수

ls() #내가 만든 모든 변수(객체)를 확인

ls.str() # 변수(객체) 할당 값 및 num(숫자), chr(문자)인지 보여줌

rm(객체명, 객체명) # 변수(객체)삭제

rm(list=ls()) # 모든 변수(객체)삭제

실습	R프로그램 제공 데이터 활용

```
data()      #R의 내장 데이터set 리스트 보기
data(mtcars)  # Motor Trend Car Road Tests 데이터 불러오기
View(mtcars) # 테이블 형태로 별도의 창으로 나타내기
str(mtcars)  #데이터 구조 확인
head(mtcars)
tail(mtcars)
head(mtcars,10)
names(mtcars)
nrow(mtcars)    # 행의 개수
length(mtcars)    # 열의 개수
attributes(mtcars)
x<-mtcars$mpg
length(x)            #행의 개수
mean(x)  #평균
median(x);min(x);max(x);min(x)  #여러개 명령은 ; 으로
range(x)
summary(x)
var(x);sd(x)  # 분산, 표준편차
cnt=round(x/10)
mtcars$bun<-rep(c(1:length(x)),len=nrow(mtcars))
  #mtcars의 bun 열추가 이때 1에서 마지막 자료까지, mtcars의 번호1,2,3 넣기
bun_1<-rep(c(1:2),len=nrow(mtcars)) #1,2 반복하여 bun_1에 할당
mtcars<-cbind(mtcars,bun_1)    #기존프레임에 bun_1 열 추가
plot(mtcars$hp,mtcars$mpg)    #마력(hp:house power), plot(mtcars$hp,mtcars$mpg)
attach(mtcars)  # mtcars 데이터 프레임 활성화
plot(hp, mpg)
```

프레임$변수명을 계속 작성하기 번거로울 때 attach()로 메모리에 할당
메모리에 할당된 데이터 프레임을 해제하고자 할때는 detach() 함수 사용

```
> # mpg 라는 동일한 이름의 벡터를 생성 후 동일하게 산포도 명령 실행하면?
  #같은 이름이 있으면 에러
> mpg <- c(20.0, 21.0, 19.2, 18.4, 19.9) # 동일한 이름의 mpg 벡터 신규 생성
> plot(mpg, hp) # 마력(hp), 연비(mpg) plot 그리기 에러발생
Error in xy.coords(x, y, xlabel, ylabel, log) :
  'x' and 'y' lengths differ
```

(해결방법1) '$'로 데이터 프레임을 지정(할당)해주고 변수를 불러오면 됩니다.
(해결방법2) rm()으로 충돌(중복)되는 동일 이름 벡터 삭제 후 attach() 함수 활용

```
> detach(mtcars) # 활성화
> rm(mpg)  # 신규로 생성했었던 mpg 벡터 삭제
> attach(mtcars)
> plot(mpg, hp) # 다시 산점도 그리면 ok
> detach(mtcars)  # 활성화 해제
```

주의) attach()로 데이터 프레임을 메모리로 호출한 이후에 원본 데이터 프레임을 변경한 경우
attach()로 활성화 시켜서 메모리에 데이터가 올라온 상태에서 '$'로 혹은 transform() 함수로 데이터 객체를 변경하였다면 detach() 로 활성화를 해제시켰다가 다시 attach()로 활성화 시켜서 사용할 것.

3 R 데이터 유형

벡터, Matrix, List, Data Frame 개념과 사용방법

1. R 데이터 유형

가. R의 기본 자료형

자료 저장을 위해 R 에서 사용하는 기본 자료형은 하나의 객체이며 그 종류는 다음과 같습니다.

숫자형	객체 이름
정수	integer
실수	numeric
복소수	complex
문자형	character ex) “abc”, “123” 등
논리형	logical ex) TRUE(T), FALSE(F)
NULL	정의되지 않은 값
NA	Missing Value
-Inf, Inf	음과 양의 무한대
NaN	수의 연산에서 불능의 경우 표현 0/0, Inf/Inf 등

2. R의 자료구조

자료객체	구성차원	자료유형	복수 데이터 유형 적용 여부
벡터(vector)	1차원	수치/문자/복소수/논리	불가능
행렬(matrix)	2차원	수치/문자/복소수/논리	불가능
데이터프레임 (dataframe)	2차원	수치/문자/복소수/논리	가능
배열(array)	2차원이상	수치/문자/복소수/논리	불가능
요인(factor)	1차원	수치/문자	불가능
시계열 (timeseries)	2차원	수치/문자/복소수/논리	불가능
리스트(list)	2차원이상	수치/문자/복소수/논리/함수/표현식 /call 등	가능

가. 벡터(Vector)

명령	설명	예시
생성	1. 연산자 (시작:종료) 2. 함수 . c() . seq(시작,종료,by=증가분) · seq(시작,종료,length.out=n) . rep(x,times=n) # times=생략가능 . rep(x,each=n)	x<-1:6 x<-c(1:3,6) x<-seq(0,10,by=2) x<-seq(0,10,length.out=6) rep(1:2,2) rep(1:2,each=2)
검출	. 변수[n] #n번째 값 . 변수[1:5] # 1~5번째 값 . 변수[c(1,3,5)] # 1,3,5번째 값 ex) 변수[1,3,5]는 오류 . 변수[논리 연산] #논리연산의 결과가 True인 값 . 변수[함수] # 함수 결과값의 값	x[2] x[1:5] x[c(1,3,5)] x[x>0] x[length(x)]
삭제	. 변수[-n] # n번째 값 삭제,	a<-x[-2]
수정	. 변수[n]<-값	x[2]<-“a”
변환	. as.vector(x)	

나. 행렬(matrix)

명령	설명	예시
생성	matrix(값, 행개수, 열 개수, byrow=F, dimnames=NULL) - 행이나 열 개수 1나만 적어도 됨 - byrow=T : 행방향으로 데이터 입력됨	x<-matrix(1:10,5,2) x<-matrix(1:10,5) x<-matrix(1:10,5,byrow=T)
검출	변수[행,열] 변수[,n] # n열만 검출 변수[n,] # n행만 검출 변수[2:3,] #2~3행까지 검출	x[1,4] x[,2] x[3,] x[2:3,]
삭제	변수[-2,] #2행전체 삭제 변수[,-1] # 1열 전체 삭제 변수[-(1:3),] # 1~3열까지 삭제 *(주의) 삭제 후 행이나 열이 1개가 되면 matirx속성을 잃고 vector속성이 된다. drop=F 옵션 주어야 함	a<-a[-2,] a<-a[,-2] a<-a[-(1:3),] a<-a[,-2,drop=F]
수정	변수[2,3]<-5	a[2,3]<-5
변환	as.matirx(x)	
정보	class(y) :자료형 정보, dim(y) : 행렬 정보 ncol(y) : 열 개수, nrow(y) : 행 개수 length(y) : 총 원소 개수	

다. 팩터(factor)

팩터는 새로운 데이터형으로 범주형 변수에서 주로 사용된다. 팩터는 입력한 값을 그대로 저장하는 것이 아니라 레벨과 위치 값을 저장한다. 레벨은 입력 값이 중복을 제거한 유일값이다.

명령	설명예시
생성	factor()

예	> a<-c(5,3,6,5)
	> a.f<-factor(a)
	> a.f
	[1] 5 3 6 5
	Levels: 3 5 6 # 입력데이터의 중복값을 제거한 유일 값
	> unclass(a.f)
	[1] 2 1 3 2 #레벨 위치 정보 2는 두 번째 레벨값, 1은 첫번째 레벨값...
	attr(,"levels")
	[1] "3" "5" "6" # 레벨정보가 문자로 입력되어 있다.
	(주의)팩터를 벡터로 바꾸면 실제 데이터가 저장되는 것이 아니라 위치정보만 저장 된다.
	따라서 실제 데이터를 잃어버리므로 주의해야 함
	
	> x<-as.numeric(a.f)
	> x
	[1] 2 1 3 2

table 함수	> a<-c(1,3,1,5,3) > table(a) #a변수의 자료 빈도수를 집계함. 1은 2번나옴, 3은2번나옴 a 1 3 5 2 2 1 > b<-c("a","b","a","b","c") > table(b) b a b c 2 2 1

라. 데이터 프레임(data.frame)

명령	설명
생성	data.frame()

예	> a<-c("김","최","김","이","김")
	> b<-c(12,10,5,11,10)
	> d<-data.frame(kids=a,ages=b)
	#
	> d
	kids ages
	1 김 12
	2 최 10
	3 김 5
	4 이 11
	5 김 10
	> str(d)
	'data.frame': 5 obs. of 2 variables:
	$ kids: Factor w/ 3 levels "김","이","최": 1 3 1 2 1
	$ ages: num 12 10 5 11 10
	> summary(d)
	kids ages
	김:3 Min. : 5.0
	이:1 1st Qu.:10.0
	최:1 Median :10.0
	Mean : 9.6
	3rd Qu.:11.0
	Max. :12.0
	
	> dim(d) # 행과 열의 개수 확인
	[1] 5 2

검출	> d$ages # 결과 벡터 [1] 12 10 5 11 10 > d[["ages"]] #결과 벡터 [1] 12 10 5 11 10 > d[,1] # 행은 모두 보여주고 1열출력, d[1]과 동일 [1] 12 10 5 11 10 > d["ages"] #결과 데이터프레임 ages 1 12 2 10
변환	as.data.frame(x)

마. 리스트(list)

명령	설명	
생성	list()	
예	> j<-list(names="joe",salary=55000,union=T) > j $names [1] "joe" $salary [1] 55000 $union [1] TRUE	> jn<-list('joe',55000,T) > jn [[1]] [1] "joe" [[2]] [1] 55000 [[3]] [1] TRUE
검출	> j$salary [1] 55000 > j$sal # 다른 구성요소 이름과 겹치지 않으면 축약 사용 가능함 [1] 55000 > j[["salary"]] [1] 55000 > j[[2]] [1] 55000	
추가 삭제	z[[4]]<-c(23,4,5) z[5:7]<-c(l,3,4) z$b<-NULL	
변환	as.list(x)	

4	R을 활용한 다양한 그래픽 표현 방법 배우기 plot()함수로 기본적 그래픽 작업 배우기	

고수준 그래프 : 차트 작성 / 저수준 그래프 : 차트안의 옵션

1. plot() 함수 : 선형 그래프 그리기

plot(x 축 데이터 , y 축 데이터 , 옵션)

▶ 옵션

인수	설명
main="메인제목"	제목설정
sub="서브제목"	서브제목설정
xlab="문자", ylab="문자"	x,y축에 사용할 문자열을 지정
ann=F	x,y축 제목을 지정하지 않음
tmag=2	제목등에 사용되는 문자의 확대율 지정
axes=F	x,y축을 표시하지 않음
axis	x,y축을 사용자의 지정값으로 표시

그래프타입	내용	선의모양	내용
type="p"	점모양그래프	lty=0, lty="blank"	투명선
type="l"	꺽은선 그래프	lty=1, lty="solid"	실선
type="b"	점과 선 모양 그래프	lty=2, lty="dashed"	대쉬선
type="c"	"b"에서 점 생략	lty=3, lty="dotted"	점선
type="o"	점과 선을 중첩	lty=4, lty="dotdash"	점선과 대쉬선
type="h"	각 점에서 x 축 까지 수직선 그래프	lty=5, lty="longdash"	긴 대쉬선
type="s"	왼쪽값을 기초로 계단모양으로 연결	lty=6, lty="twodash"	2개의 대쉬선
type="S"	오른쪽값을 기초로 계단모양으로 연결		
type="n"	축만 그리고 그래프는 그리지 않음		

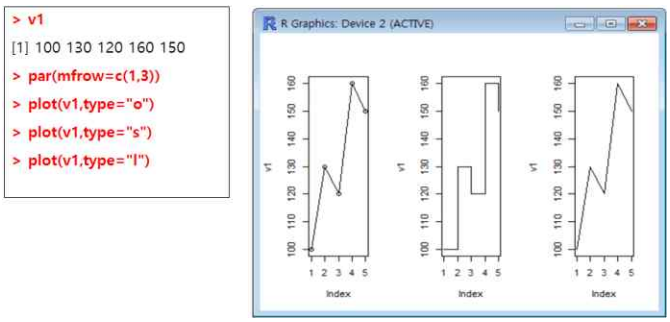
색기호등	설명
col=1, col="blue"	기호의 색지정 1:검정, 2:빨강, 3:초록, 4:파랑, 5:연파랑, 6:보라, 7:노랑, 8:회색
pch=0, pch="문자"	점의 모양 지정
bg ="blue"	그래프의 배경색 지정
lwd="숫자"	선을 그릴때 선의 굵기 지정
cex="숫자"	점이나 문자를 그릴때 점이나 문자의 굵기를 지정

* 그래프 추가 옵션

```
v1 <- c(100,130,120,160,150)
plot(v1,type='o',col='red',ylim=c(0,200),axes=FALSE,ann=FALSE)
axis(1,at=1:5 ,lab=c("MON","TUE","WED","THU","FRI"))    # X축 제목 설정
axis(2,ylim=c(0,200))
title(main="FRUIT" , col.main="red",font.main=4)
title(xlab="DAY", col.lab="black")
title(ylab="PRICE",col.lab="blue")
```

■ 그래프의 배치 조정하기 (mfrow)

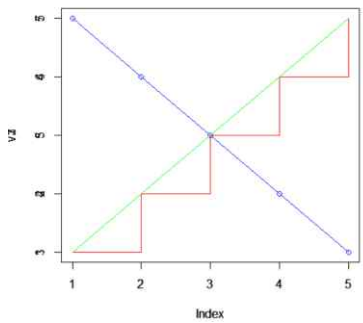
par (mfrow = c(nr,nc)) <-- nr : 행의 갯수 , nc : 열의 개수



■ oma 옵션 테스트하기 <- 그래프 전체의 여백 조정하기 입니다.

```
> par(oma=c(2,1,0,0))
> plot(a,xlab="aaa")
```

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

> v1 <- c(1,2,3,4,5) > v2 <- c(5,4,3,2,1) > v3 <- c(3,4,5,6,7) > plot(v1,type="s",col="red",ylim=c(1,5)) > par(new=T) # 이 부분이 중복 허용 부분입니다 > plot(v2,type="o",col="blue",ylim=c(1,5)) > par(new=T) # 이 부분이 중복 허용 부분입니다 > plot(v3,type="l",col="green")	
> plot(v1,type="s",col="red",ylim=c(1,10)) > lines(v2,type="o",col="blue",ylim=c(1,5)) > lines(v3,type="l",col="green",ylim=c(1,15))	**lines로 추가하는 방법도 있음.**

■ 그래프에 범례 추가하기

legend(x 축 위치 , y 축 위치,내용 , cex=글자크기 , col =색상 , pch=크기 , lty=선 모양

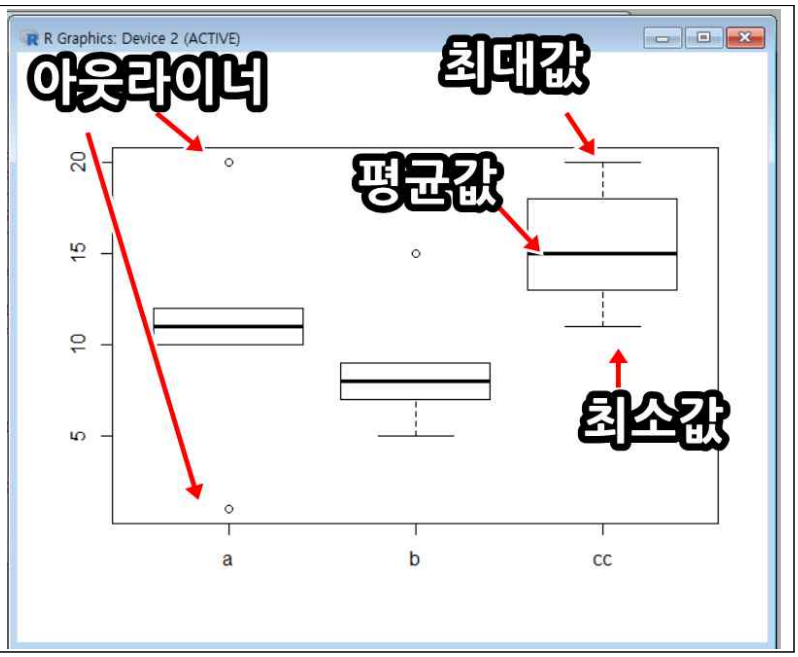
```
> legend(4,9,c("v1","v2","v3"),cex=0.9,col=c("red","blue","green"),lty=1)
```

2. boxplot 개념 이해하기

- **이상치**

통계에서는 데이터 샘플에서 관찰된 한 값이 다른 관측값과 거리가 있을 때 이상치(outlier)라고 한다. 측정에 있어서 데이터들의 가변성, 변동성(variability) 때문일 수 있고 실제로 잘못된 실험에 의한 에러일 수 있다. 후자의 경우에는 분명히 데이터 분석 이전에 outlier를 제거를 해야한다.

```
a<-c(10,12,1,11,20)
b<-c(5,7,15,8,9)
cc<-c(11,20,15,18,13)
ccc<-data.frame(a,b,cc)
boxplot(ccc)
mean(a); mean(b); mean(cc)
[1] 10.8
[1] 8.8
[1] 15.4
```




```
R에서 Outlier검출법  http://sosal.kr/945
fivenum(a)  # 0%, 25%, 50%, 70%, 100% 로 나눔
UpperQ = fivenum(data)[4]
LoLowerQ = fivenum(a) [2]
UpperQ+IQR(a)*1.5     # max 이상값 15
LowerQ - IQR(a)*1.5  #min 이하값  7

DF<-read.csv("example_studentlist.csv")
boxplot(DF$height~DF$bloodtype)
```

3. boxplot 실습

http://blog.naver.com/PostView.nhn?blogId=kist125&logNo=90157263902

* 실습 :

R에서 그래프를 그리는 이유는 크게 세가지 임.

- 빠르게 데이터를 탐색하기 위해 그래프 그리기(EDA-탐색적 자료 분석) => plot
  plot(), barplot(), hist(), boxplot()
- 보다 정교한 데이터의 특징을 나타내기 위해 그래프 그리기 => ggplot
- Repor를 위한 그래프 => rchart패키지
- 그래프에 사용되는 변수는 반드시 명목형이거나 수치형이어야함.
- 만약 에러가 난다면 as.factor()를 사용해 명목형 변수로 변경해야 함.

```
DF<-read.csv("example_studentlist.csv")
View(DF)
str(DF)
names(DF)
attach(DF)
plot(age)
plot(height,weight)
plot(weight~height)
plot(height,sex)
plot(sex,height)
DF2<-data.frame(height,weight)
DF2
plot(DF2)
DF3<-cbind(DF2,age)
DF3
plot(DF3)
plot(DF)
```

Data: DF

	name	sex	age	grade	absence	bloodtype	height	weight
1	김길동	남자	23	3	무	O	165.3	68.2
2	이미린	여자	22	2	무	AB	170.1	53.0
3	홍길동	남자	24	4	무	B	175.0	80.1
4	김철수	남자	23	3	무	AB	182.1	85.7
5	손세수	여자	20	1	무	A	168.0	49.5
6	박미희	여자	21	2	무	O	162.0	52.0
7	강수진	여자	22	1	무	O	155.2	45.3
8	이희수	여자	23	1	무	A	176.9	55.0
9	이철린	남자	23	3	무	B	178.5	64.2
10	방희철	남자	22	2	무	B	176.1	61.3
11	박수호	남자	24	4	무	O	167.1	62.0
12	임동민	남자	22	2	무	AB	180.0	75.8
13	김민수	남자	21	1	무	A	162.2	55.3
14	이희진	여자	23	3	무	O	176.1	53.1
15	김미진	여자	22	2	무	B	158.2	45.2
16	김동수	남자	24	4	무	B	168.6	70.2
17	여수근	남자	21	1	무	A	169.2	62.2

```
plot(weight~height, pch=as.integer(sex))   # Level 별 그래프 보기/여자남자를 동그라미,세모로표시
legend("topleft",c("남","여"),pch=sex)    #왼쪽위 상단에 범례표시
coplot(weight~height | sex)   # Level별 그래프 보임      coplot(종속변수~독립변수 | 명목형변수)
coplot(weight~height | bloodtype)
plot(weight~height,ann=F)     # 고수준 그래프 함수 호출시 다른 인자 없음.
title(main="A 대학 B 학과생 몸무게외 키의 상관관계")    # 제목추가
title(xlab="몸무게")    # x축제목추가
title(ylab="키")    # Y축 제목추가
grid()
abline(v=mean(height), col="red")       # 키의 평균값 위치를 빨간색으로 보여줌
abline(h=mean(weight), col="red")       # 몸무게의 평균값 위치를 빨간색으로 보여줌
plot(bloodtype)    #factor인 경우 데이터 빈도수를 계산함.
```

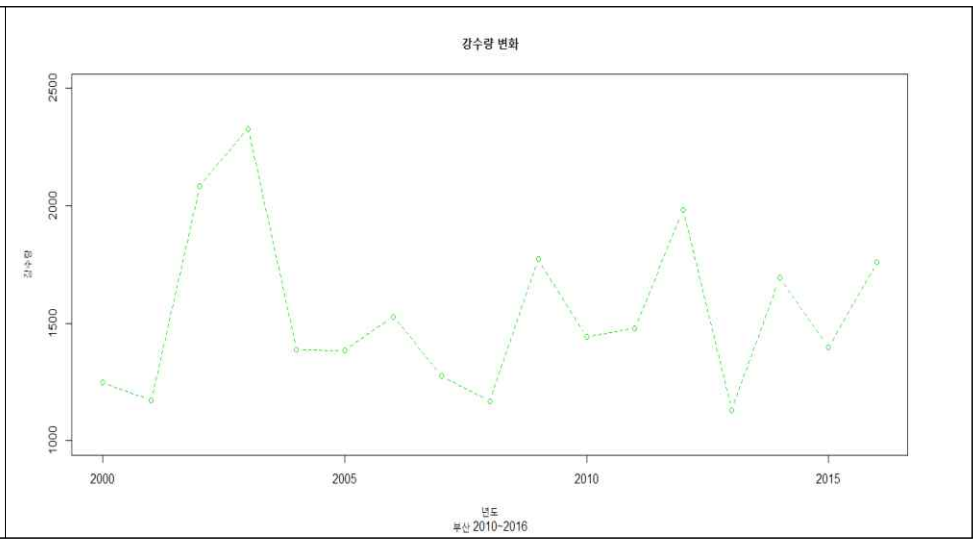
* 실습: 강수량 자료 차트 표현

```
setwd("c:/data_r")      # 자료가있는 폴더를 먼저 C드라이브에세팅함, getwd(), dir()
data<-read.csv("강수량_서울_년.csv")
 # 기상청 공공데이터 2010~2016년 서울 강수량자료임, 년, 지점, 강수량.mm 으로 구성
data;View(data);head(data,3);tail(data,2);data[2,];data[3,];data[2,3]; data[1:5,2:3]
str(data);summary(data);class(data)    # 한줄에 ; 으로 여러명령어 기술 가능
data1<-data.frame(data[1],data[3])
     #data자료의 1열 년, 3열 강수량.mm으로 data 재구성
     # 또는 data<-data.frame(data[-2])
plot(data1)
plot(data1,ylim=c(500,3000))    #y축 최소 500, 최대 3000으로 설정  c=combine
plot(data1,ylim=c(500,3000), xlim=c(2010,2016))  # x축 2010~2016년도 설정
plot(data1,ylim=c(500,3000), xlim=c(2010,2016), main="강수량 변화",
   sub="서울 2010~2016", xlab="년도",ylab="강수량")
plot(data1, type="p")
plot(data1, type="l")
plot(data1, type="b")
plot(data1, type="o")
plot(data1, type="h")
plot(data1, type="s")
plot(data1, type="S")
plot(data1, type="n")
plot(data1, type="l", lty=2)
plot(data1, type="h", lty=3, col=2)
plot(data1, type="b", lty=3, col=2, pch=5)
plot(data1, type="b", lty=3, col=2, pch=10,lwd=3, cex=3)
 name<-"서울"    #Name 변수에 서울 글자 할당  기상청 108번 코드가 서울임.
 data2<-data.frame(data[-2],name)
   # 또는 data2<-data.frame(data[-2],name="서울")
colnames(data2)<-c("year","val","name")  #데이터2의 제목변경
  # 또는 colnames(data2)[2]<-"val"
data2
plot(data2$year,data2$val)  #프레임이름$변수명 으로 사용해야함  또는 data2[1]
attach(data2)  # data2$year 를 year로 사용하기 위하여 메모리에 data 프레임올림
plot(year,val)
plot(year,val, type="s")
savePlot("서울_년_강수량",type="png")      # write.csv(data2,"tmp.csv")
dir()    # 디렉토리 파일 확인
ls()  #메모리 자료 확인
rm(data1) #data1만 메모리에서 삭제
detach(data2) #attach 된 자료 삭제
```

* [미션] plot

[미션] "강수량_부산_년.csv" 자료를 불러와 다음과 같은 차트 출력한후 부산_년_강수량.png 파일로 저장하기

	A	B	C
1	년	지점	강수량(mm)
2	2000	159	1248.5
3	2001	159	1171.3
4	2002	159	2085.2
5	2003	159	2328.3
6	2004	159	1386.5
7	2005	159	1383.9
8	2006	159	1528.3
9	2007	159	1276.5
10	2008	159	1168.3
11	2009	159	1772.9
12	2010	159	1441.9
13	2011	159	1478.6
14	2012	159	1983.3
15	2013	159	1130.1
16	2014	159	1693.1
17	2015	159	1396.7
18	2016	159	1760.2



[미션] 위에서 작성된 서울의 data2 프레임과 부산의 b_data 프레임을 합친후 "년_강수량_서울부산.csv"로 저장

	A	B	C	D
1		year	val	name
2	1	2000	1186.8	서울
3	2	2001	1386	서울
4	3	2002	1388	서울
5	4	2003	2012	서울
6	5	2004	1499.1	서울
7	6	2005	1358.4	서울
8	7	2006	1681.9	서울
9	8	2007	1212.3	서울
10	9	2008	1356.3	서울
11	10	2009	1564	서울
12	11	2010	2043.5	서울
13	12	2011	2039.3	서울
14	13	2012	1646.3	서울
15	14	2013	1403.8	서울
16	15	2014	808.9	서울
17	16	2015	792.1	서울
18	17	2016	991.7	서울
19	18	2000	1248.5	부산
20	19	2001	1171.3	부산
21	20	2002	2085.2	부산
22	21	2003	2328.3	부산
23	22	2004	1386.5	부산
24	23	2005	1383.9	부산
25	24	2006	1528.3	부산
26	25	2007	1276.5	부산
27	26	2008	1168.3	부산

[미션] 메모리에 올려진 모든 자료 제거

답: "강수량_부산_년.csv" 자료를 불러와 다음과 같은 차트 출력하기		
b_data에 csv자료 할당		b_data<-read.csv("강수량_부산_년.csv")
b_data보기		b_data
2번째 코드값 제외하고 마지막에 부산 추가		b_data<-data.frame(b_data[-2],Name="부산")
b_data출력		b_data
제목변경		colnames(b_data)<-c("year","val","name")
attach		attach(b_data)
부산강수량 범위보기 1130.1~2328.3		range(val)
부산강수량데이터 통계보기		summary(val)
차트 작성 plot(year,val,ylim=c(1000,2500), xlim=c(2000,2016), main="강수량 변화", sub="부산 2010~2016", xlab="년도",ylab="강수량", type="b", lty=2, col=3, pch=1)		
차트저장	R	savePlot("부산_년_강수량",type="png")

	R Studio	png("test.png")
		plot(year, val) #data는 저장할 plot명
		dev.off()

		detach(b_data)
서울과 부산 자료 합침		new_data<-rbind(data2,b_data)
new_data를 csv로 저장함		write.csv(new_data,"년_강수량_서울부산.csv")
메모리 모든 변수 지움		rm(list=ls())

5 상관계수 및 패키지 설치

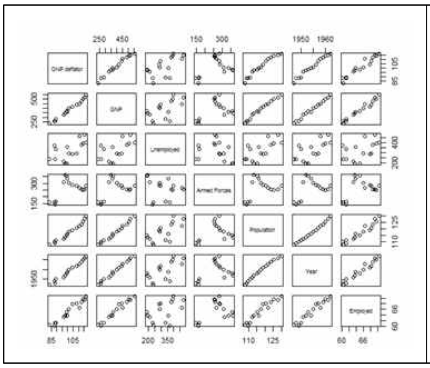
#상관분석은 두변수 또는 데이터 셋간의 통계적 관계를 나타냄.
#예제 : R내에 있는 longley

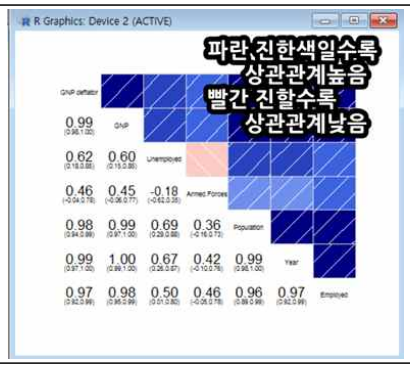
```
> longley
   GNP.deflator     GNP Unemployed Armed.Forces Population Year Employed
1947         83.0 234.289      235.6       159.0    107.608 1947   60.323
1948         88.5 259.426      232.5       145.6    108.632 1948   61.122
1949         88.2 258.054      368.2       161.6    109.773 1949   60.171
1950         89.5 284.599      335.1       165.0    110.929 1950   61.187
1951         96.2 328.975      209.9       309.9    112.075 1951   63.221
1952         98.1 346.999      193.2       359.4    113.270 1952   63.639
1953         99.0 365.385      187.0       354.7    115.094 1953   64.989
1954        100.0 363.112      357.8       335.0    116.219 1954   63.761
1955        101.2 397.469      290.4       304.8    117.388 1955   66.019
1956        104.6 419.180      282.2       285.7    118.734 1956   67.857
1957        108.4 442.769      293.6       279.8    120.445 1957   68.169
1958        110.8 444.546      468.1       263.7    121.950 1958   66.513
1959        112.6 482.704      381.3       255.2    123.366 1959   68.655
1960        114.2 502.601      393.1       251.4    125.368 1960   69.564
                                480.6       257.2    127.852 1961   69.331
                                400.7       282.7    130.081 1962   70.551
> cor(Longley)
            GNP.deflator     GNP Unemployed Armed.Forces Population      Year Employed
GNP.deflator    1.0000000 0.9915892  0.6206334    0.4647442  0.9791634 0.9911492 0.9708985
GNP             0.9915892 1.0000000  0.6042609    0.4464368  0.9910901 0.9952735 0.9835516
Unemployed      0.6206334 0.6042609  1.0000000   -0.1774206  0.6865515 0.6682566 0.5024981
Armed.Forces    0.4647442 0.4464368 -0.1774206    1.0000000  0.3644163 0.4172451 0.4573074
Population      0.9791634 0.9910901  0.6865515    0.3644163  1.0000000 0.9939528 0.9603906
Year            0.9911492 0.9952735  0.6682566    0.4172451  0.9939528 1.0000000 0.9713295
Employed        0.9708985 0.9835516  0.5024981    0.4573074  0.9603906 0.9713295 1.0000000
```

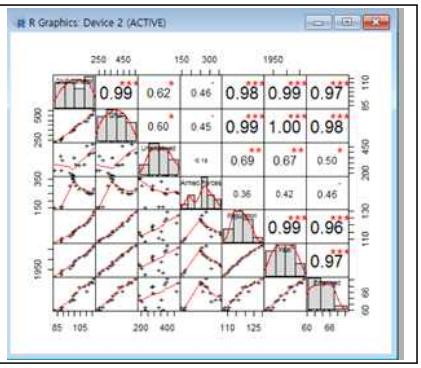
#-상관 분석 의견
데이터상으로
고용율과 GNP는
정상관 관계가 있는것으로 나타남

* 패키지를 이용한 상관계수 시각화

```
result<-longley
plot(result)        #산점도 차트

install.packages("corrgram")     #인터넷연결되어 있어야함
library(corrgram)

corrgram(result)
corrgram(result, upper.panel=panel.conf) #색상과 숫자차트
corrgram(result, lower.panel=panel.conf)

install.packages("PerformanceAnalytics")
library(PerformanceAnalytics)
chart.Correlation(result, histogram=, pch="+") #빨간그래프차트
```



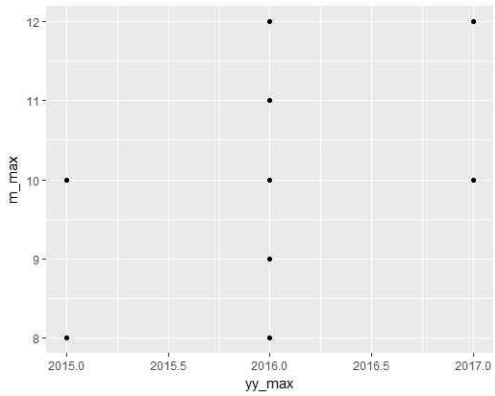




R을 활용한 다양한 그래픽 표현 방법 배우기

ggplot

```
install.packages("ggplot2")
library(ggplot2)
setwd("c:/data_r")
지도<-read.csv("야영장위경도.csv")

지도
ggplot(지도, aes(yy_max, m_max))  + geom_point()
```



 # 동그라미 포인트 작성

```
ggplot(지도, aes(yy_max, m_max))  + geom_point()
```

#동그라미 색상을 주소에 따라 다르게함

```
ggplot(지도, aes(yy_max, m_max))  + geom_point(aes(colour=주소))
```

#동그라미 크기를 g_sum에 따라 다르게함

```
ggplot(지도, aes(yy_max, m_max))  + geom_point(aes(colour=주소, size=g_sum))
```

#겹쳐져 있는 동그라미의 투명도를 조정함. 0~1 (0은 투명, 1은 불투명)

```
ggplot(지도, aes(yy_max, m_max))  + geom_point(aes(colour=주소, size=g_sum),alpha=(0.2))
```

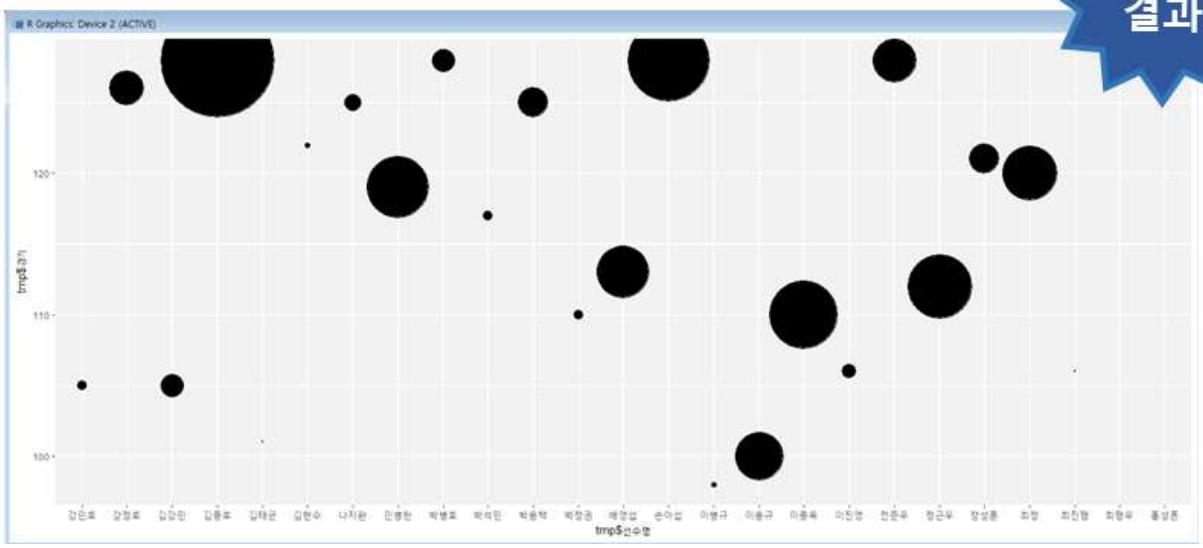
#아래와 같이 추가하여 사용가능함.

```
a<-ggplot(지도, aes(yy_max, m_max))
a
a<-a+ geom_point()
a
```

야구성적.csv를 불러와 선수명별 경기수에
geom_point로 작성
이때 원의 크기는 도루수로 지정

```
tmp<-read.table("야구성적.csv",sep=",",header=T)
tmp<-read.csv("야구성적.csv")
```

미션
결과



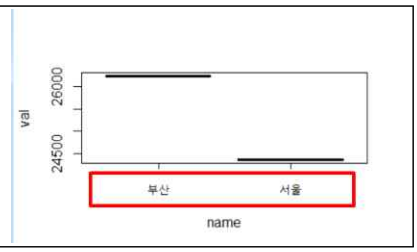

```
tmp
dim(tmp)
a<-ggplot(tmp,aes(x=tmp$선수명, y=tmp$경기))
크기=tmp$도루
크기
b<-a+geom_point(size=크기)
b
```

6	데이터 전처리(그룹합) & plot 그룹합 plot	

1. 데이터 전처리 및 plot 작성

```
data<-read.csv("년_강수량_서울부산.csv")
names(data)
subset(data,val>=1000 & name=="서울")
# 2010년도 이상 자료중 서울자료만 보기
tmp<-ifelse(data$val>mean(data$val),"평균이상","평균이하")
data<-cbind(data,tmp)   # 열추가
aggregate(data$val~data$year,data,sum)   #년도별 강수량집계
aggregate(val~name+year,data,sum) #년도별 지역별 강수량 집계
```

```
plot(aggregate(val~name,data,sum))
#지역별 강수량 집계를 차트로 나타냄
```




```
# ----------------------------------------
tmp<-split(data$val,data$name)
tmp
```

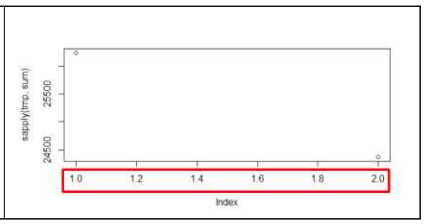
```
$부산
 [1] 1248.5 1171.3 2085.2 2328.3 1386.5 1383.9 1528.3 1276.5 1168.3 1772.9 1441.9 1478.6
1983.3 1130.1 1693.1
[16] 1396.7 1760.2
$서울
 [1] 1186.8 1386.0 1388.0 2012.0 1499.1 1358.4 1681.9 1212.3 1356.3 1564.0 2043.5 2039.3
1646.3 1403.8  808.9
[16]  792.1  991.7
```

sapply(tmp,mean) **# 데이터프레임을 벡터, 또는 행렬의 형태로 반환 (s: simplify)**

```
 부산       서울
1543.153  1433.553
```

plot(sapply(tmp,sum))

#지역별 강수량 집계를 차트로 나타냄. aggregate함수와 다른점은 행열의 배열로 들어가 있어 부산,서울 이 아니라 1,2,로 X축이 구성됨



tapply(data$val,data$name,range)

 # 테이블(행,열) 형태로 저장 출력값, 기준컬럼, 함수

 # tapply=> 입력값을 index에 지정한 factor 값으로분류(그룹화)하여 매개변수로 넘어온 function을 적용하는 함수

tmp_1<-tapply(data$val,data$year,sum)

table(data$name) **# 부산, 서울 글자가 몇 번 나왔는지 출력해주는 함수**

plot(table(data$tmp)) **# 평균이상, 평균이하가 몇 번 나왔는지 출력해주는 함수**

prop.table(table(data$tmp)) **# 평균이상, 평균이하의 비율은 얼마인지 상대도수구하기**

table<-rbind(table(data$tmp),prop.table(table(data$tmp)))

```
> table
        평균이상    평균이하
[1,] 12.0000000 22.0000000
[2,]  0.3529412  0.6470588
```

table<-addmargins(table,margin=2) **# 1행의 도수와 2행의 상대도수에 대한 합 34,1**

```
> table
        평균이상    평균이하 Sum
[1,] 12.0000000 22.0000000  34
[2,]  0.3529412  0.6470588   1
```

[미션]: 데이터 전처리 복습 및 plot 작성

[미션1] "강수량_일별.csv" 자료 불러와 구조 및 내용 파악하기

	A	B	C	D	E	F
1	날짜	월	요일	지점	강수량(mm)	
2	2016-01-01	1	금	108	0	
3	2016-01-02	1	토	108	0	
4	2016-01-03	1	일	108	0	
5	2016-01-04	1	월	108	0	
6	2016-01-05	1	화	108	0	
7	2016-01-06	1	수	108	0	
8	2016-01-07	1	목	108	0	
9	2016-01-08	1	금	108	0	
10	2016-01-09	1	토	108	0	
11	2016-01-10	1	일	108	0	
12	2016-01-11	1	월	108	0	
13	2016-01-12	1	화	108	0	
14	2016-01-13	1	수	108	0.4	
15	2016-01-14	1	목	108	0	

[미션2] ifesle 작성

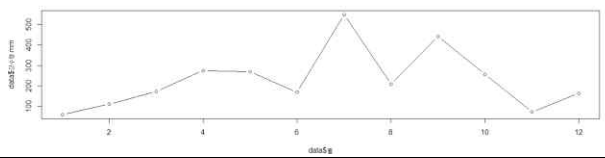
- 토,일 요일은 주말, 그 외는 주중으로 표기하는 tmp변수 작성
- 1,2,3월은 Q1, 4,5,6월은 Q2, 7,8,9월은 Q3, 10,11,12월은 Q4로 표기하는 tmp1변수 작성
- data 변수에 tmp, tmp1 열추가

```
> head(data)
         날짜 월 요일 지점 강수량.mm.  tmp tmp1
1 2016-01-01  1   금  108          0 주중   Q1
2 2016-01-02  1   토  108          0 주말   Q1
3 2016-01-03  1   일  108          0 주말   Q1
4 2016-01-04  1   월  108          0 주중   Q1
5 2016-01-05  1   화  108          0 주중   Q1
6 2016-01-06  1   수  108          0 주중   Q1
> tail(data)
           날짜 월 요일 지점 강수량.mm.  tmp tmp1
727 2016-12-26 12   월  159       22.2 주중   Q4
728 2016-12-27 12   화  159        2.6 주중   Q4
729 2016-12-28 12   수  159        0.0 주중   Q4
730 2016-12-29 12   목  159        0.0 주중   Q4
731 2016-12-30 12   금  159        0.0 주중   Q4
732 2016-12-31 12   토  159        0.0 주말   Q4
```

[미션3] aggregate 작성

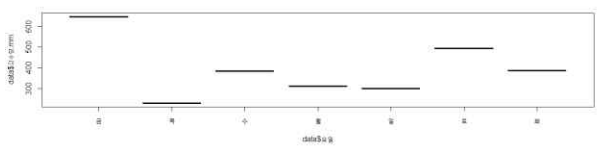
요일별 강수량 집계

	data$요일	data$강수량.mm
1	금	648.3
2	목	227.7
3	수	383.7
4	월	310.0
5	일	300.3
6	토	494.7
7	화	387.2



월별 강수량 집계

	data$월	data$강수량.mm
1	1	60.5
2	2	111.3
3	3	174.0
4	4	275.3
5	5	269.3
6	6	169.6
7	7	547.0
8	8	208.6
9	9	440.9
10	10	257.2
11	11	73.3
12	12	164.9



주말/주중으로 나누어 집계

	data$tmp	data$강수량.mm
1	주말	795.0
2	주중	1956.9

분기별로 나누어 집계

	data$tmp1	data$강수량.mm
1	Q1	171.8
2	Q2	888.2
3	Q3	1196.5
4	Q4	495.4

지점별 주말주중으로 나누어 집계

	data$tmp	지점	data$강수량.mm
1	주말	108	259.4
2	주중	108	732.3
3	주말	159	535.6
4	주중	159	1224.6

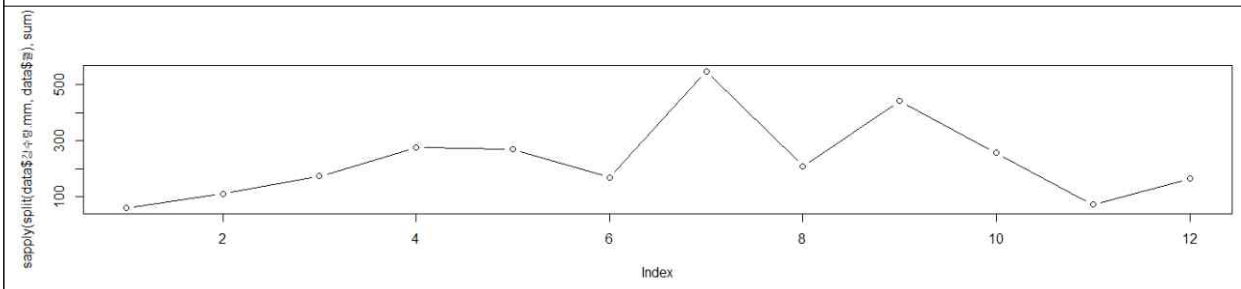
지점별/주말주중/분기별 강수량집계

	data$지점	tmp	tmp1	data$강수량.mm
1	108	주말	Q1	27.9
2	159	주말	Q1	45.2
3	108	주중	Q1	20.7
4	159	주중	Q1	78.0
5	108	주말	Q2	135.8
6	159	주말	Q2	113.4
7	108	주중	Q2	196.4
8	159	주중	Q2	442.6
9	108	주말	Q3	63.9
10	159	주말	Q3	294.6
11	108	주중	Q3	394.4
12	159	주중	Q3	443.6
13	108	주말	Q4	31.8
14	159	주말	Q4	82.4
15	108	주중	Q4	120.8
16	159	주중	Q4	260.4

[미션4] split, sapply 작성

요일별 강수량 집계						
금	목	수	월	일	토	화
648.3	227.7	383.7	310.0	300.3	494.7	387.2

월별 강수량 집계											
1	2	3	4	5	6	7	8	9	10	11	12
60.5	111.3	174.0	275.3	269.3	169.6	547.0	208.6	440.9	257.2	73.3	164.9



주말/주중 별 강수량 집계	
주말	주중
795.0	1956.9

분기별 강수량 집계			
Q1	Q2	Q3	Q4
171.8	888.2	1196.5	495.4

[미션5] table 작성

요일별 도수(횟수) 및 상대도수 구하기

	금	목	수	월	일	토	화
[1,]	106.0000000	104.0000000	104.0000000	104.0000000	104.0000000	106.0000000	104.0000000
[2,]	0.1448087	0.1420765	0.1420765	0.1420765	0.1420765	0.1448087	0.1420765

월별/요일별 빈도수 구하기 (서울,부산 지점코드가 2개 있으므로 빈도수/2 해야함)

	1	2	3	4	5	6	7	8	9	10	11	12	<-월
금	5	4	4	5	4	4	5	4	5	4	4	5	
목	4	4	5	4	4	5	4	4	5	4	4	5	
수	4	4	5	4	4	5	4	5	4	4	5	4	
월	4	5	4	4	5	4	4	5	4	5	4	4	
일	5	4	4	4	5	4	5	4	4	5	4	4	
토	5	4	4	5	4	4	5	4	4	5	4	5	
화	4	4	5	4	5	4	4	5	4	4	5	4	

월별/요일별 빈도수 행열 합구하기

	1	2	3	4	5	6	7	8	9	10	11	12	Sum
금	5	4	4	5	4	4	5	4	5	4	4	5	53
목	4	4	5	4	4	5	4	4	5	4	4	5	52
수	4	4	5	4	4	5	4	5	4	4	5	4	52
월	4	5	4	4	5	4	4	5	4	5	4	4	52
일	5	4	4	4	5	4	5	4	4	5	4	4	52
토	5	4	4	5	4	4	5	4	4	5	4	5	53
화	4	4	5	4	5	4	4	5	4	4	5	4	52
Sum	31	29	31	30	31	30	31	31	30	31	30	31	366

강수량이 0이면 ‘X_비안옴’, 그 외는 ‘O_비옴’ 으로 변수추가한뒤 빈도수 집계

	O_비옴	X_비안옴	Sum
금	14.5	38.5	53.0
목	13.5	38.5	52.0
수	12.5	39.5	52.0
월	15.5	36.5	52.0
일	16.5	35.5	52.0
토	15.0	38.0	53.0
화	18.5	33.5	52.0
Sum	106.0	260.0	366.0

[미션6] 조건에 맞는 자료 추출 및 저장

월요일 자료만 추출하여서 파일 저장 => 강수량_월요일.csv

화요일과 목요일 자료만 추출하여서 파일 저장 => 강수량_화,목요일.csv

비온자료만 추출하여서 파일 저장 => 강수량_비온자료.csv

비온자료중 Q1,Q2,Q4 자료만 추출하여서 파일 저장 => 강수량_비온자료_Q1,2,4.csv

7	데이터 전처리(NA, 필터) 결측치 및 데이터 조건 필터	

1. NA(결측값: 비어있거나 없는값)

(1) 결측값이 포함되어 있는지 확인하는 방법: is.na()

(2) 결측값이 총 몇 개인지 계산하는 방법: sum(is.na())

(3) 데이터 프레임 모든 변수 결측치 합 구하기: colSums()

(4) 결측값을 통계 분석 시 제외(미포함): na.rm = TRUE

(5) 결측값이 들어있는 행 전체를 데이터 셋에서 제거: na.omit()

(6) 특정 행과 열에 결측값이 들어있는 행을 데이터 셋에서 제거 : complete.cases()

(7) 결측값을 다른 값으로 대체: dataset$var[is.na(dataset$var)] <- new_value

(8) 데이터프레임의 각 변수의 결측값을 각 변수 별 평균값으로 일괄 대체

: sapply(dataset, function(x) ifelse(is.na(x), mean(x, na.rm=TRUE), x))

```
> library(MASS)
> is.na(Cars93)
Manufacturer Model Type Min.Price Price Max.Price MPG.city MPG.highway AirBags
DriveTrain Cylinders EngineSize 1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE 2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
FALSE FALSE FALSE FALSE
> sum(is.na(Cars93))
[1] 13
> colSums(is.na(Cars93))
> colSums(is.na(Cars93))
      Manufacturer             Model              Type         Min.Price             Price
                 0                 0                 0                 0                 0
         Max.Price           MPG.city       MPG.highway           AirBags        DriveTrain
                 0                 0                 0                 0                 0
         Cylinders         EngineSize        Horsepower               RPM      Rev.per.mile
                 0                 0                 0                 0                 0
   Man.trans.avail Fuel.tank.capacity        Passengers            Length         Wheelbase
                 0                 0                 0                 0                 0
             Width        Turn.circle    Rear.seat.room      Luggage.room            Weight
                 0                 0                 2                11                 0
            Origin              Make
                 0                 0
> sum(Cars93$Luggage.room)
[1] NA
> mean(Cars93$Luggage.room)
[1] NA
> sum(Cars93$Luggage.room, na.rm = TRUE)
[1] 1139
```

```
> mean(Cars93$Luggage.room, na.rm = TRUE)
[1] 13.89024

> Cars93_1 <- na.omit(Cars93)
> sum(is.na(Cars93_1))
[1] 0

> sum(is.na(Cars93))
[1] 13

> # Cars93 데이터 프레임의 "Rear.seat.room" 칼럼 내 결측값이 있는 행 전체 삭제

> Cars93_2 <- Cars93[ complete.cases(Cars93[ , c("Rear.seat.room")]), ]
> sum(is.na(Cars93_2))
[1] 9
# 결측치 0으로 모두 대체
tmp<-Cars93
sum(is.na(tmp))
tmp[is.na(Cars93)]<-0

# 결측치 평균값으로 대체

> Cars93_7 <- Cars93[1:20,c("Rear.seat.room", "Luggage.room")]

> colSums(is.na(Cars93_7))
Rear.seat.room     Luggage.room
             1                3

> Cars93_7
> Cars93_7 <- Cars93[1:20,c("Rear.seat.room", "Luggage.room")]

> colSums(is.na(Cars93_7))
> Cars93_7

> Cars93_7<-data.frame(sapply(Cars93_7,function(x) ifelse(is.na(x),mean(x,na.rm=T),x))
> Cars93_7
```

[실습하기]

```
> install.packages("data.table")
> library(data.table)
> data<-fread("example_coffee.csv",header=T,stringsAsFactors=T, data.table=F)
  # 데이터량이 많을때는fread 명령으로 가져옴
> Size<-data$sizeOfsite
> summary(Size)          # NA값 확인

   Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
   0.00     28.12     50.00     75.53     93.75  24080.00        19
> plot(Size)
> Size[Size>10000]<-NA      # 조건에 해당하는 자료 NA로 대체
> summary(Size)             # NA갯수 증가 확인

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   0.00   28.12   50.00   75.02   93.75 1406.00      20
> plot(Size)
> Size[Size==0]<-NA
> summary(Size)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   0.25   30.00   51.92   77.23   95.30 1406.00    1361
> Size<-Size[complete.cases(Size)]
> summary(Size)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.25   30.00   51.92   77.23   95.30 1406.00

> table(data$yearOfStart) #년도별 오픈한 커피숍갯수 확인

1964 1966 1967 1968 1969 1970 1971 1972 1974 1975 1976 1979 1980 1981 1982 1983 1984
1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000
   2    2    3    1    2    4    6    3    1    2    5    4    9    8   12    9   11   18
21   21   26   23   25   28   37   50   48   48   41   54   54   46   89
2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
 183  398  799  648  654  863 1233 1579 2489 4172 5942 6315 7270 9905 3650

> aa<-table(data$stateOfbusiness,data$yearOfStart)
> aa

          1964 1966 1967 1968 1969 1970 1971 1972 1974 1975 1976 1979 1980 1981 1982 1983 1984 1985 1986 1987
1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998
  운영중    0    0    2    0    0    2    4    2    1    1    1    2    3    6    2    3    4    5    5    6   11
5    7    7    3   14   14   13   14   21   25
  폐업 등   2    2    1    1    2    2    2    1    0    1    4    2    6    2   10    6    7   13   16   15   15
18   18   21   34   36   34   35   27   33   29

          1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
  운영중    23   26   76  105  163  180  204  314  496  729 1229 2503 3961 4642 6045 9125 3564
  폐업 등   23   63  107  293  636  468  450  549  737  850 1260 1669 1981 1673 1225  780   86

> addmargins(aa,margin=1)
```

2. 데이터 필터

▶ 실습하기

내용	R코드
hflights 패키지 설치	install.packages("hflights")
hflights 라이브러리 설치	library(hflights)
hflights 구조보기	str(hflights)
열의개수 확인	length(hflights)
행의개수 확인	nrow(hflights)
메모리에 hflights 올리기	attach(hflights)
Dest 도착지별 건수합구하여서 Cnt_Dest에 할당	Cnt_Dest<-table(Dest)
Dest 도착지개수 확인(명목형변수세기)	length(Cnt_Dest)
도착지점별 차트그리기	plot(Cnt_Dest)
도착지 횟수 가장 작은값부터 가장 많은값 까지 범위보기	range(Cnt_Dest)
가장적은곳 도착지 확인	Cnt_Dest[Cnt_Dest==1]
가장많은곳 도착지 확인	Cnt_Dest[Cnt_Dest==max(Cnt_Dest)]
Cnt_Dest값중 6000건이상 a에 할당	a<-Cnt_Dest[Cnt_Dest>6000]
a의 오른쪽 끝에 sum	addmargins(a,margin=1)
	plot(a)
	barplot(a)

내용	R코드
hflights값 data로 할당	data<-hflights
NA결측치 모두 제거	data<-data[complete.cases(data),]
메로리에 data 할당 참고: detach(data)	attach(data)
월별 TaxiOut건수합 구하기 tapply함수사용	tmp<-tapply(TaxiOut,Month,sum)
q_1 변수에 월이 7~12월은 하반기, 그 외는 상반기로 텍스트할당	q_1<-ifelse(Month>6,"하반기","상반기")
cc데이터셋에 q_1과 TaxiOut 할당	cc<-data.frame(q_1,TaxiOut)
q변수에 1,2,3월은 Q1, 4,5,6월은 Q2, 7,8,9월은 Q3, 10,11,12월은 Q4 텍스트할당	
q<-ifelse(Month>=10,"Q4",ifelse(Month>=7,"Q3",ifelse(Month>=4,"Q2","Q1")))	
q,q_1변수 data에 열추가	data<-cbind(data,q,q_1)
Month,q,q_1,FlightNum,AirTime,Arrdelay,Dest 만 만들어서 data에 할당	
data<-cbind(Month,q,q_1,FlightNum,AirTime,ArrDelay,Dest)	

[실습: 학생자료]

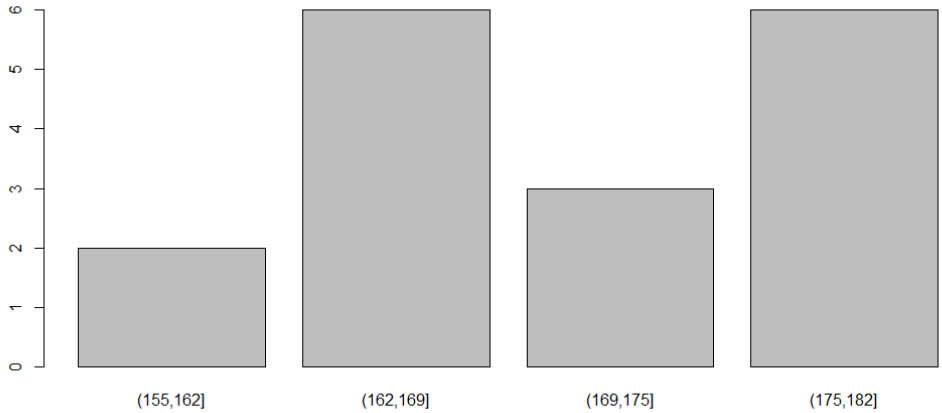
```
> data<-read.csv("example_st.csv")
> data
> subset(data, height>170))    # 키 170초과
> subset(data, height!=170))    #키가 170이 아닌 학생자료
> subset(data, bloodtype!='A'))  #혈액형이 A가 아닌 학생자료
> subset(data,subset=(bloodtype!='A' | grade==3))
    #혈액혁이 A가 아니거나 grade가 3이 아닌 학생 자료
> subset(data,subset=(bloodtype!='A' & grade==3))
   # 혈액형이 A가 아니거나 grade가 3인 학생 자료
> subset(data,select=c(name,height), subset=(height>180))
    name height
4 김철수  182.1
> subset(data,select=c(-height,-weight,-grade))    # - 는 제외하고 출력
> colnames(data)
[1] "name"      "sex"       "age"       "grade"     "absence"   "bloodtype" "height"
[8] "weight"
> colnames(data)[4]<-"jumsoo"    #4번째 열제목 grade를 jumsoo로 변경
> colnames(data)
[1] "name"      "sex"       "age"       "jumsoo"    "absence"   "bloodtype" "height"
[8] "weight"
> colnames(data)<-c("v1","v2","v3","v4","v5","v6","v7","v8")
> colnames(data)
[1] "v1" "v2" "v3" "v4" "v5" "v6" "v7" "v8“
> colnames(data)<-c("name","sex","age","grade","absence","bloodtype","height","weight")
> colnames(data)
[1] "name"      "sex"       "age"       "grade"     "absence"   "bloodtype" "height"
[8] "weight"
> attach(data)
> BMI<-weight/height^2  #체질량지수
```

```
> BMI
 [1] 0.002495966 0.001831754 0.002615510 0.002584407 0.001753827 0.001981405 0.001880679
 [8] 0.001757547 0.002014924 0.001976705 0.002220439 0.002339506 0.002101955 0.001712284
[15] 0.001806032 0.002469574 0.002172649

> data<-cbind(data,BMI)
> head(data)
     name  sex age grade absence bloodtype height weight          BMI
1 김길동 남자  23     3      유         O  165.3   68.2 0.002495966
2 이미린 여자  22     2      무        AB  170.1   53.0 0.001831754
3 홍길동 남자  24     4      무         B  175.0   80.1 0.002615510
4 김철수 남자  23     3      무        AB  182.1   85.7 0.002584407
5 손세수 여자  20     1      유         A  168.0   49.5 0.001753827
6 박미희 여자  21     2      무         O  162.0   52.0 0.001981405

> tmp<-read.csv("example_st_추가.csv")
> tmp      #data의 1,2,3...의 행 자료와 tmp의 1,2,3의 행자료가 순서가다름
      name footsize
1   강수친      245
2   김길동      270
3   김동수      265
4   김미진      235
5   김민수      270
6   김철수      280
7   박미희      240   ......

> data<-merge(data,tmp,by="name")
> data
      name  sex age grade absence bloodtype height weight          BMI footsize
1   강수친 여자  22     1      무         O  155.2   45.3 0.001880679      245
2   김길동 남자  23     3      유         O  165.3   68.2 0.002495966      270
3   김동수 남자  24     4      유         B  168.6   70.2 0.002469574      265
4   김미진 여자  22     2      무         B  158.2   45.2 0.001806032      235
5   김민수 남자  21     1      무         A  162.2   55.3 0.002101955      270
6   김철수 남자  23     3      무        AB  182.1   85.7 0.002584407      280
7   박미희 여자  21     2      무         O  162.0   52.0 0.001981405      240
8   박수호 남자  24     4      유         O  167.1   62.0 0.002220439       NA
9   방희철 남자  22     2      무         B  176.1   61.3 0.001976705      275
10  손세수 여자  20     1      유         A  168.0   49.5 0.001753827      240
11  여수근 남자  21     1      무         A  169.2   62.2 0.002172649      265
12  이미린 여자  22     2      무        AB  170.1   53.0 0.001831754      245
13  이철린 남자  23     3      무         B  178.5   64.2 0.002014924       NA
14  이희수 여자  23     1      무         A  176.9   55.0 0.001757547      245
15  이희진 여자  23     3      무         O  176.1   53.1 0.001712284      245
16  임동민 남자  22     2      무        AB  180.0   75.8 0.002339506      280
17  홍길동 남자  24     4      무         B  175.0   80.1 0.002615510      275
```

```
> tmp_h<-split(height,sex)
> tmp_h
$남자
 [1] 165.3 175.0 182.1 178.5 176.1 167.1 180.0 162.2 168.6 169.2

$여자
[1] 170.1 168.0 162.0 155.2 176.9 176.1 158.2

> tmp_h2<-split(height,bloodtype)
> tmp_h2
$A
[1] 168.0 176.9 162.2 169.2

$AB
[1] 170.1 182.1 180.0

$B
[1] 175.0 178.5 176.1 158.2 168.6

$O
[1] 165.3 162.0 155.2 167.1 176.1

> sapply(tmp_h,mean)
     남자       여자
172.4100  166.6429
> sapply(tmp_h2,mean)
       A       AB        B        O
169.075 177.400 171.280 165.140

> sapply(tmp_h2,range)
          A     AB     B     O
[1,] 162.2 170.1 158.2 155.2
[2,] 176.9 182.1 178.5 176.1

> tapply(height,bloodtype,mean)      #tapply(출력값,기준컬럼,적용함수)
       A       AB        B        O
169.075 177.400 171.280 165.140

> tapply(height,grade,mean)
        1        2        3        4
166.3000 169.2800 175.5000 170.2333
```

```
> freq<-table(bloodtype) #테이블 함수는 명목형변숭의 항목의 개수(빈도수)를 셀수 있음.
> freq
bloodtype
 A AB  B  O
 4  3  5  5

> R_freq<-prop.table(freq)  #상대도수구하기
> R_freq
bloodtype
        A         AB         B         O
0.2352941 0.1764706 0.2941176 0.2941176

> table<-rbind(freq,R_freq)  #행과행끼리 합치기
> table
                A         AB         B         O
freq    4.0000000 3.0000000 5.0000000 5.0000000
R_freq  0.2352941 0.1764706 0.2941176 0.2941176

> table<-addmargins(table,margin=1)  #합구하기, margin =1 이면 열의합
> table
                A         AB         B         O
freq    4.0000000 3.0000000 5.0000000 5.0000000
R_freq  0.2352941 0.1764706 0.2941176 0.2941176
Sum     4.2352941 3.1764706 5.2941176 5.2941176

> table<-addmargins(table,margin=2) #margin =3 이면 행의합, 생략은 행열합
> table
                A         AB         B         O Sum
freq    4.0000000 3.0000000 5.0000000 5.0000000  17
R_freq  0.2352941 0.1764706 0.2941176 0.2941176   1
Sum     4.2352941 3.1764706 5.2941176 5.2941176  18

> ct<-table(absence,bloodtype)
> ct
        bloodtype
absence A AB B O
     무 3  3 4 3
     유 1  0 1 2
```

```
> fac_h<-cut(height, breaks=4)  #height 의 17개값을 4개구간으로 나누자
> plot(fac_h)
```




```
> t_h<-table(fac_h)  # 빈도수
> t_h
fac_h
(155,162] (162,169] (169,175] (175,182]
        2         6         3         6
> a<-rbind(t_h,prop.table(t_h))  # 상대도구
> a
     (155,162] (162,169] (169,175] (175,182]
t_h  2.0000000 6.0000000 3.0000000 6.0000000
     0.1176471 0.3529412 0.1764706 0.3529412

> rownames(a)[2]<-"RelativeFreq"  #2번행의 이름 변경
> a
                (155,162] (162,169] (169,175] (175,182]
t_h             2.0000000 6.0000000 3.0000000 6.0000000
RelativeFreq    0.1176471 0.3529412 0.1764706 0.3529412

> CumuFreq<-cumsum(a[2,])  # 누적상대도수
> CumuFreq
(155,162] (162,169] (169,175] (175,182]
0.1176471 0.4705882 0.6470588 1.0000000
```

```
> a<-rbind(a,CumuFreq)
> a
                  (155,162] (162,169] (169,175] (175,182]
t_h             2.0000000 6.0000000 3.0000000 6.0000000
RelativeFreq 0.1176471 0.3529412 0.1764706 0.3529412
CumuFreq        0.1176471 0.4705882 0.6470588 1.0000000
> rownames(a)<-c("도수","상대도수","누적도수")
> a
              (155,162] (162,169] (169,175] (175,182]
도수       2.0000000 6.0000000 3.0000000 6.0000000
상대도수 0.1176471 0.3529412 0.1764706 0.3529412
누적도수 0.1176471 0.4705882 0.6470588 1.0000000
> a<-addmargins(a,margin=2)
> a
              (155,162] (162,169] (169,175] (175,182]        Sum
도수       2.0000000 6.0000000 3.0000000 6.0000000 17.000000
상대도수 0.1176471 0.3529412 0.1764706 0.3529412  1.000000
누적도수 0.1176471 0.4705882 0.6470588 1.0000000  2.235294

> detach(data)
```

참고) 한글 인코딩

```
# read.csv("파일위치/파일명", fileEncoding="euc-kr")
# read.table("파일위치/파일명", fileEncoding="euc-kr"
```

메모장에서 프로그램상에서 인코딩 정보 확인. 한글이 에디터에서 잘 보이고 트레이에 인코딩이 ANSI, EUC-KR, Windows 949 라고 표현되어 있으면 코드와 같이 EUC-KR을 입력하면 되고, UTF-8로 표시되어 있으면 fileEncoding="EUC-KR"로 지정하면 된다.

CP949, UTF-8, unknown

8	사용자 정의 함수 여러개 plot 그리기	

data() #R 내장데이터 확인/ 'datasets'는 패키지 설치하지 않아도 되나 다른 데이터셋은 상단의 패키지 설치해야함

```
mtcars
dim(mtcars)
nrow(mtcars)  #행개수 확인
length(mtcars) #열개수 확인

par("mar")
par(mar=c(1,1,1,1))
차트행개수=round(length(mtcars)/3)
```

########### 반복 for 구문을 이용한 plot 차트 작성 for(변수 in 시작값:마지막값)

```
 par(mfrow=c(차트행개수,3)) # 열은 무조건 3개, 전체열을 3개로 나누어서행으로 작성

 for(i in 1:length(mtcars)) {
    tmp<-mtcars[,i]
    names<-colnames(mtcars)[1]
    plot(tmp,main=names)
  }
```

############ plot 을 사용자 정의함수로 작성

```
 차트작성함수<-function(x) {
                 차트행개수=round(length(x)/3)
                 par(mfrow=c(차트행개수,3))

                 for(i in 1:length(x)) {
                     tmp<-x[,i]
                     names<-colnames(x)[1]
                     plot(tmp,main=names)
                   }
 }

 차트작성함수(mtcars)
 women                #내장데이터 women 자료 불러오기
 차트작성함수(women)  #미리작성된 차트작성함수에 적용하기
```

9	데이터를 원하는 형태로 변형하는 다양한 방법들 stringr()패키지 사용 방법 배우기	

1. stringr 패키지

문자열을 특정 문자를 기준으로 나누기 위해서는 외부 패키지가 필요하며 'stringr'를 많이 사용한다.

▶ str_split(객체,분리할문자열") => 리스트 형태로 반환함.

```
> fruits <- str_c('apple','/','orange','/','banana')
> View(fruits)
>
```

	x
1	apple/orange/banana

```
> a<-str_split(fruits,"/")
> View(a)
>
```

	c..apple....orange....banana..
1	apple
2	orange
3	banana

▶ str_split_fixed(객체,분리할문자열,나눌갯수)

```
> a<-str_split_fixed(fruits,"/",3)
> a
     [,1]    [,2]     [,3]
[1,] "apple" "orange" "banana"
> View(a)
>
```

Data: a

	V1	V2	V3
1	apple	orange	banana

가) 전국 인구조사 자료 정리하기(전처리 연습)

```
setwd("c:/data_r")
install.packages("stringr")
library("stringr")
```

data<-read.csv("example_population.csv",stringsAsFactor=F)

#stringsAsFactor=T 로 하면 문자열은 fector로 받아들임.

#F는 문자열은 char로 숫자는 num으로 받는 내용임

인구수 숫자에 , 가 있어 vector로 인식하므로 문자열로 처리하기 위해 stringsAsFactors=F로 처리함.

	A	B	C	D	E	F	G
1	City	Population	Households	PersInHou	Male	Female	SexRatio
2	서울특별시 (1100000000)	10,078,850	4,197,478	2.4	4,962,774	5,116,076	0.97
3	서울특별시 종로구 (1111000000)	155,695	72,882	2.14	76,962	78,733	0.98
4	서울특별시 중구 (1114000000)	126,817	59,614	2.13	63,292	63,525	1
5	서울특별시 용산구 (1117000000)	235,186	108,138	2.17	114,119	121,067	0.94
6	서울특별시 성동구 (1120000000)	298,145	126,915	2.35	148,265	149,880	0.99
7	서울특별시 광진구 (1121500000)	362,197	158,769	2.28	177,946	184,251	0.97
8	서울특별시 동대문구 (1123000000)	362,604	160,110	2.26	181,825	180,779	1.01
9	서울특별시 중랑구 (1126000000)	417,976	177,077	2.36	208,657	209,319	1
10	서울특별시 성북구 (1129000000)	464,176	192,670	2.41	227,676	236,500	0.96

str(data)

```
> data<-read.csv("example_population.csv",stringsAsFactor=F)
> str(data)
'data.frame':   281 obs. of  7 variables:
 $ City      : chr  "서울특별시  (1100000000)" "서울특별시 종로구 (1111000000)" "서울특별시 중구 (1114000000)" "서울특별시 용산구 (1117000000)" $
 $ Population: chr  "10,078,850" "155,695" "126,817" "235,186" ...
 $ Households: chr  "4,197,478" "72,882" "59,614" "108,138" ...
 $ PersInHou : num  2.4 2.14 2.13 2.17 2.35 2.28 2.26 2.36 2.41 2.36 ...
 $ Male      : chr  "4,962,774" "76,962" "63,292" "114,119" ...
 $ Female    : chr  "5,116,076" "78,733" "63,525" "121,067" ...
 $ SexRatio  : num  0.97 0.98 1 0.94 0.99 0.97 1.01 1 0.96 0.97 ...
> data1<-read.csv("example_population.csv",stringsAsFactor=T)
> str(data1)
'data.frame':   281 obs. of  7 variables:
 $ City      : Factor w/ 281 levels "강원도  (4200000000)",..: 160 183 184 181 176 166 171 185 177 163 ...
 $ Population: Factor w/ 279 levels "1,015,972","1,072,222",..: 11 35 28 79 120 160 161 181 206 142 ...
 $ Households: Factor w/ 280 levels "1,145,232","1,160,150",..: 195 249 230 22 48 83 86 104 118 70 ...
 $ PersInHou : num  2.4 2.14 2.13 2.17 2.35 2.28 2.26 2.36 2.41 2.36 ...
 $ Male      : Factor w/ 280 levels "1,049,546","1,239,275",..: 207 253 246 25 64 102 108 125 144 82 ...
 $ Female    : Factor w/ 280 levels "1,018,898","1,251,862",..: 224 257 244 35 66 105 104 123 150 86 ...
 $ SexRatio  : num  0.97 0.98 1 0.94 0.99 0.97 1.01 1 0.96 0.97 ...
```

head(data, n=5) #5개 자료만 샘플로 보기

tmp<-str_split_fixed(data[,1],"\\(",2)

View(tmp)

str_split_fixed(문자열,분리할 기준 문자, 분리할 개수)

괄호로 시작하는 숫자부분 삭제, '\\('는 정규식 표현으로 결과적으로는 '('를 알림, (를 삭제하고 2개로 분리

Data: tmp

	V1	V2
1	서울특별시	1100000000)
2	서울특별시 종로구	1111000000)
3	서울특별시 중구	1114000000)
4	서울특별시 용산구	1117000000)
5	서울특별시 성동구	1120000000)

New_city<-str_split_fixed(tmp[,1]," ",2) # 주소의 빈칸을 기준으로 사이띄기

View(New_city)

Data: New_city

	V1	V2
1	서울특별시	
2	서울특별시	종로구
3	서울특별시	중구
4	서울특별시	용산구
5	서울특별시	성동구

colnames(New_city)<-c("Juso_1","Juso_2") # New_city 열의 이름을 변경함.

data<-data.frame(New_city,data[,c(2:7)])

View(data)

Data: data

	Juso_1	Juso_2	Population	Households	PersInHou	Male
1	서울특별시		10,078,850	4,197,478	2.40	4,962,774
2	서울특별시	종로구	155,695	72,882	2.14	76,962
3	서울특별시	중구	126,817	59,614	2.13	63,292
4	서울특별시	용산구	235,186	108,138	2.17	114,119
5	서울특별시	성동구	298,145	126,915	2.35	148,265
6	서울특별시	광진구	362,197	158,769	2.28	177,946

summary(data) #자료요약보기

```
> summary(data)
       Juso_1          Juso_2      Population         Households          PersInHou          Male              Female             SexRatio
 경기도     : 52          : 18   Length:281         Length:281         Min.   :1.860   Length:281         Length:281         Min.   :0.900
 경상북도   : 26   동구    :  6   Class :character   Class :character   1st Qu.:2.180   Class :character   Class :character   1st Qu.:0.980
 서울특별시: 26   중구    :  6   Mode  :character   Mode  :character   Median :2.390   Mode  :character   Mode  :character   Median :1.000
 경상남도   : 24   남구    :  5                                         Mean   :2.373                                         Mean   :1.007
 전라남도   : 23   서구    :  5                                         3rd Qu.:2.550                                         3rd Qu.:1.030
 강원도     : 19   북구    :  4                                         Max.   :2.910                                         Max.   :1.290
 (Other)    :111   (Other):237
>
```

length(data$Juso_2[data$Juso_2==" "]) #Juso_2에 빈셀이 있는 자료가 몇 개 있는지

data[data==" "]<-NA #빈칸에 NA값 넣기

summary(data)

```
> summary(data)
       Juso_1          Juso_2      Population         Households          PersInHou          Male              Female             SexRatio
 경기도     : 52   동구    :  6   Length:281         Length:281         Min.   :1.860   Length:281         Length:281         Min.   :0.900
 경상북도   : 26   중구    :  6   Class :character   Class :character   1st Qu.:2.180   Class :character   Class :character   1st Qu.:0.980
 서울특별시: 26   남구    :  5   Mode  :character   Mode  :character   Median :2.390   Mode  :character   Mode  :character   Median :1.000
 경상남도   : 24   서구    :  5                                         Mean   :2.373                                         Mean   :1.007
 전라남도   : 23   북구    :  4                                         3rd Qu.:2.550                                         3rd Qu.:1.030
 강원도     : 19   (Other):237                                         Max.   :2.910                                         Max.   :1.290
 (Other)    :111   NA's   : 18
```

Data: data

	Juso_1	Juso_2	Population	Households	PersInHou	Male
1	서울특별시	NA	10,078,850	4,197,478	2.40	4,962,774
2	서울특별시	[illegible]	155,695	72,882	2.14	76,962
3	서울특별시	중구	[illegible]	[illegible]	2.13	63,292
4	서울특별시	용산구	[illegible]	[illegible]	2.17	114,119

NA로 변경됨

nrow(data) #281개

data<-data[complete.cases(data),] #NA값이 있는 행 지우기 na.omit()

nrow(data) # 281개-18개=> 263개

length(data) # 8개의 열

```
for(i in 3:8){
data[,i]<-sapply(data[,i],function(x)gsub(",","",x))
data[,i]<-as.numeric(data[,i])
}
```

gsub(찾는글자, 바꿀글자)

```
> str(data)
'data.frame':   263 obs. of  8 variables:
 $ Juso_1    : Factor w/ 17 levels "강원도","경기도",..: 9 9 9 9 9 9 9 9 9 9 ...
 $ Juso_2    : Factor w/ 241 levels " ","가평군 ",..: 188 189 158 105 32 65 190 106 6 63 ...
 $ Population: num  155695 126817 235186 298145 362197 ...
 $ Households: num  72882 59614 108138 126915 158769 ...
 $ PersInHou : num  2.14 2.13 2.17 2.35 2.28 2.26 2.36 2.41 2.36 2.57 ...
 $ Male      : num  76962 63292 114119 148265 177946 ...
 $ Female    : num  78733 63525 121067 149880 184251 ...
 $ SexRatio  : num  0.98 1 0.94 0.99 0.97 1.01 1 0.96 0.97 0.97 ...
```

```
summary(data)
```

데이터처리 복습

```
F_Ratio<-round(data$Female/(data$Male+data$Female),2)
M_Ratio<-round(data$Male/(data$Male+data$Female),2)
data<-data.frame(data,M_Ratio,F_Ratio)
aggregate(Population~Juso_1,data,sum)
aggregate(F_Ratio~Juso_1,data,mean)
tmp_value<-apply(data[3:8],1,sum)    #3~8열까지 숫자가 있는 자료만 갖고가기
View(tmp_value)
tmp_value<-apply(data[3:8],2,sum)
View(tmp_value)
tmp_value<-sapply(data[3:8],sum)
View(tmp_value)
tmp_value<-lapply(data[3:8],sum)
View(tmp_value)
tapply(data$Male,data$Juso_1,sum)
install.packages("plyr")
library(plyr)
ddply(data,'Juso_1',summarise,sum_Male=sum(Male),sum_Female=sum(Female))
```

[미션1]

[미션2]

10	R을 활용한 다양한 그래픽 표현 방법 배우기 트리맵	

* 트리맵 패키지 실행하기

트리맵은 계층 데이터를 중첩된 사각형의 집합으로 표시합니다. 계층의 각 수준은 다른 사각형("잎")를 포함하는 색이 칠해진 사각형("가지"라고도 함)으로 표시됩니다. 각 사각형 안에 공간은 측정된 정량 값을 기반으로 할당되며, 왼쪽 상단(최대)에서 오른쪽 하단(최소)까지 크기 별로 정렬된 사각형으로 표시됩니다.

※ 트리맵을 사용하는 경우

- 많은 양의 계층적 데이터를 표시하는 경우
- 가로 막대형 차트로는 많은 수의 값을 효과적으로 처리할 수 없는 경우
- 각 부분과 전체 간의 비율을 표시하는 경우
- 계층 구조의 각 수준의 범주에 걸쳐 측정값이 분포되는 패턴을 표시하는 경우
- 크기 및 색 구분을 사용하여 특성을 표시하는 경우
- 패턴, 이상값, 가장 중요한 요인 및 예외를 강조하는 경우

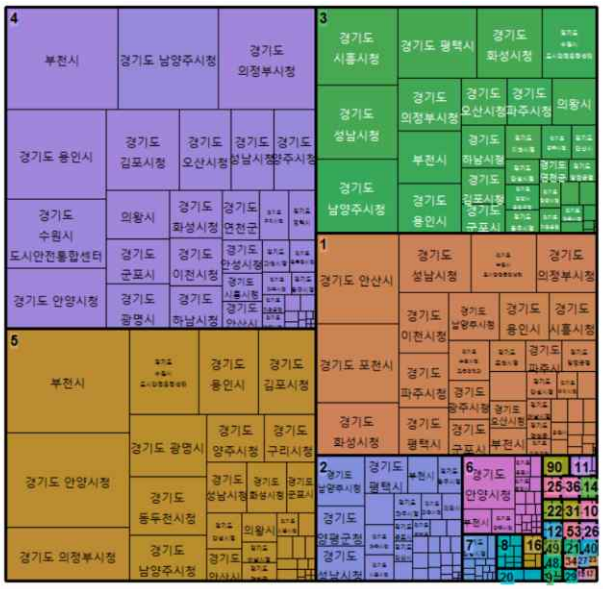
```
install.packages("treemap")
library(treemap)
cctv<-read.csv("cctv.csv")
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	관리기관명	소재지도로	소재지지번주소	설치목적구분	카메라대수	카메라화소	촬영방면정	보관일수	설치년월	관리기관전	위도	경도	데이터기준일자
2	경기도 시흥시청		경기도 시흥시 대	다목적	11	200	주차장내부	30	Jan-13	031-488-6	37.44255	126.7889	2015-08-31
3	경기도 시흥시청		경기도 시흥시 월	재난재해	1	200	월곶해안		Oct-14	031-310-2	37.3883	126.7372	2015-08-31
4	경기도 시흥시청		경기도 시흥시 장	다목적	10	200	주차장내부	30	Jan-13	031-488-6	37.37786	126.7848	2015-08-31
5	경기도 시흥시청		경기도 시흥시 신	다목적	26	41	주차장내부	30	Jan-10	031-488-6	37.43897	126.7851	2015-08-31
6	경기도 시흥시청		경기도 시흥시 대	재난재해	1	41	도로		Apr-09	031-310-2	37.46296	126.7917	2015-08-31
7	경기도 시흥시청		경기도 시흥시 정	다목적	27	41	주차장내부	30	Jan-12	031-488-6	37.34558	126.7344	2015-08-31
8	경기도 시흥시청		경기도 시흥시 정	다목적	20	41	주차장내부	30	Jan-10	031-488-6	37.34803	126.7504	2015-08-31

■ cctv의 '관리기관명','설치목적구분','카메라대수' 갖고와 다양한 트리맵그리기

> treemap(cctv,index="관리기관명",vSize="카메라대수",type="value", vColor="카메라대수")

> treemap(cctv,vSize="카메라대수",index=c("카메라대수","관리기관명"),align.labels=list(c("left", "top"),c("center", "center")))



■ 관리기관명별 카메라대수 집계내어 y에 할당후 트리맵으로 그리기

y<-aggregate(카메라대수~관리기관명,cctv,sum)

treemap(y,vSize="카메라대수",index=c("카메라대수","관리기관명"),align.labels=list(c("left", "top"),c("center", "center")))

■ 관리기관명별, 설치목적구분에 따른 카메라대수 집계내어 y에 할당

> z<-aggregate(카메라대수~관리기관명+설치목적구분,cctv,sum)

> treemap(z,vSize="카메라대수",index=c("카메라대수","관리기관명","설치목적구분"),align.labels=list(c("left", "top"),c("center", "center")))

> treemap(z,vSize="카메라대수",index=c("관리기관명","설치목적구분"),align.labels=list(c("left", "top"),c("center", "center")))

> treemap(z, index=c("관리기관명", "설치목적구분"), vSize="카메라대수", vColor="data.available", type="categorical")

[미션1]

[미션2]

[미션3]

[미션4]

11	데이터를 원하는 형태로 변형하는 다양한 방법들 워드클라우드	

참고) http://www.datamarket.kr/xe/board_AGDR50/240

1. 텍스트마이닝 - 정형데이터 워드클라우드

데이터 시각화 기법 중 하나로, 하나의 텍스트에 출현하는 단어를 빈도에 비례하는 크기로 표출한 그래프로서 텍스트 내 명사(noun)들로 구성된 단어 클라우드는 잠재적 독자에게 경제적이고 효과적인 요약을 제공한다.

꼭 중간에 핵심 단어가 들어가는 것은 아니며 R에서 워드클라우드를 실해하기 위해서는 아래의 패키지가 일부 또는 모두 필요할수 있다.

```
# 워드클라우드 관련 패키지
install.packages("wordcloud")
library(wordcloud)
library(RColorBrewer)

#---------세종 명사 사전 관련 패키지 java 미리 설치되어있어야함--
install.packages("KoNLP")
library(KoNLP)
useSejongDic( )
```

구글검색 " wordcloud in r " 로 검색하면 자세한 정보가 나옴.

- **문법 :**

wordcloud(words,freq,scale=c(4,.5),min.freq=3,max.words=Inf,random.order=TRUE, random.color=FALSE,rot.per=.1,colors="black",ordered.colors=FALSE, use.r.layout=FALSE,fixed.asp=TRUE, ...)

옵션	설명
words	출력할 단어들
freq	언급된 빈도수
scale	글자크기 c(Max,Min)
min.freq	최소언급횟수지정 - 이 값 이상 언급된 단어만 출력
max.words	최대언급횟수지정. 이 값 이상 언급되면 삭제
random.order	출력되는 순서를 임의로 지정. F 는 빈도가 큰 단어를 중심에둠
random.color	글자 색상을 임의로 지정. T 는 실행시마다 색상 변경
rot.per	단어배치를 90 도 각도로 출력. 회전되는 단어의 빈도
colors	출력될 단어들의 색상 지정
ordered.colors	이 값을 true 로 지정할 경우 각 글자별로 색상을 순서대로 지정할 수 있음
use.r.layout	이 값을 false 로 할 경우 R 에서 c++ 코드를 사용할 수 있음

```
setwd("c:/data_r")
dir()
data<-read.csv("영화_역대_박스오피스.csv")
summary(data)
data$매출액<-as.numeric(gsub(",","",data$매출액))
summary(data)
plot(data$영화명,data$매출액)

#-------------트리맵-----------------
install.packages("treemap")
library(treemap)

treemap(data,index="영화명",vSize="매출액")

names(data)
treemap(data,index=c("국적","영화명"),vSize="매출액")

treegraph(data,index=c("국적","영화명"))
```

```
#---------자료 필터--------------------
nrow(data)
head(data)
영화_한국<-subset(data,data$대표국적=="한국" & as.numeric(data$순위)<=50)
영화_한국
nrow(영화_한국)

#---------트리맵 및 트리그래프--------------------
treemap(영화_한국,index=c("배급사","영화명"),vSize="매출액")
treemap(영화_한국,index=c("배급사","영화명"),vSize="매출액",inflate.labels = TRUE)
 #inflate.labels는 타일사이즈에 맞추어서 글자크기 증가

treemap(영화_한국,index=c("배급사","영화명"),vSize="매출액",fontsize.labels=c(20, 14),alpha=0.1)
 #fontsize.labels=c(배급사글자크기,영화명글자크기)

treemap(영화_한국,index=c("배급사","영화명"),vSize="매출액",fontsize.labels=c(20, 14),
      align.labels=list(c("left", "top"), c("center", "center")))

treegraph(영화_한국,index=c("배급사","영화명"),show.labels=T)
#-------------워드클라우드-----------------------
install.packages("wordcloud")
library(wordcloud)
wordcloud(data$영화명,data$매출액)
wordcloud(data$영화명,data$매출액,random.order=F)


wordcloud(data$영화명,data$매출액,random.order=F,rot.per=0)
wordcloud(data$영화명,data$매출액,min.freq=mean(data$매출액))


#--------------aggregate 집계 워드클라우드 -------------
tmp<-aggregate(매출액~배급사,영화_한국,sum)
tmp
plot(tmp$매출액~tmp$배급사)
barplot(tmp$매출액~tm$배급사)
wordcloud(tmp$배급사,tmp$매출액,random.order=F,rot.per=0)
```

#----------색상 워드클라우드 ----------

```
display.brewer.all(n=10, exact.n=FALSE)
 display.brewer.pal(8,"Dark2")
 palete<-brewer.pal(8,"Dark2")

wordcloud(tmp$배급사,tmp$매출액,random.order=F,color=palete,rot.per=0)
```

#----------글꼴 워드클라우드 ----------

```
windowsFonts(word_font=windowsFont("궁서"))
wordcloud(tmp$배급사,tmp$매출액,random.order=F,color=palete,rot.per=0,family="
word_font")
```

#----------table 집계----------

```
sum_배급사<-table(data$배급사)      #배급사 빈도수 집계
names(sum_배급사)
```

배급사이름(열이름) names(sum_배급사)	배급사1	배급사2	배급사3	배급사5
빈도수(값) sum_배급사	3	5	1	8

```
barplot(sum_배급사)
wordcloud(names(sum_배급사),sum_배급사,min.freq=1,random.order=F,color=palet
e,rot.per=0,family="word_font")


sum_배급사2<-tapply(data$매출액,data$배급사,sum)
sum_배급사2
wordcloud(names(sum_배급사2),sum_배급사2,min.freq=1,random.order=F,color=pal
ete,rot.per=0,family="word_font")
```

2. 문자열 처리

1) sub :

문자열에서 특정 패턴을 찾아내여 첫번째에 해당하는 것만 replacement 옵션에 지정된 값으로 바꾸는 함수(대소문자구별, ignore.case=TRUE 사용시만 대소문자 구별안함)

문법 : sub(pattern="찾는글자", replacement="바꿀글자", x=개체명)

a<-"r은 데이터 분석 도구로서 알프로그램 사용시 R은 많은 시각화를 제공합니다."
sub(pattern="R", replacement="알", x=a) **#sub("R","알",a)**

▶ 결과: r은 데이터 분석 도구로서 알프로그램 사용시 알은 많은 시각화를 제공합니다.

문법 : sub(pattern="찾는글자", replacement="바꿀글자", x=개체명, ignore.case=TRUE)

a<-"r은 데이터 분석 도구로서 알프로그램 사용시 R은 많은 시각화를 제공합니다."
sub(pattern="R", replacement="알", x=a, ignore.case=TRUE) **#sub("R","알",a,TRUE)**

▶ 결과: 알은 데이터 분석 도구로서 알프로그램 사용시 R은 많은 시각화를 제공합니다.

2) gsub

찾아낸 모든 pattern에 대하여 replacement모두 적용

a<-"r은 데이터 분석 도구로서 알프로그램 사용시 R은 많은 시각화를 제공합니다." gsub(pattern="r", replacement="알", x=a, ignore.case=TRUE) **#gsub("R","알",a,TRUE)**

▶ 결과: 알은 데이터 분석 도구로서 알프로그램 사용시 알은 많은 시각화를 제공합니다.

3) grep : grep(찾는글자,개체)

grep("한국",data$국적) #data$국적 필드(열이름)에서 "한국" 글자찾기
length(grep("한국",data$국적)) "갯수보여주기"
국적_한국<-data[grep("한국",data$국적),]
국적_한국

3. 정규화(정규식)

a<-"1 a b 2 3 가 나 다"

식	내용	결과
\\d	숫자	뿌 a b 뿌 뿌 가 나 다
	gsub("\\d","뿌",a)	
\\D	숫자를 제외한 모든 문자	1뿌뿌뿌뿌뿌2뿌3뿌뿌뿌뿌뿌뿌"
	gsub("\\D","뿌",a)	
\\s	공백 탭 개행	1뿌a뿌b뿌2뿌3뿌가뿌나뿌다
	gsub("\\s","뿌",a)	
\\S	공백 탭 개행을 제외한 모든 문자	뿌 뿌 뿌 뿌 뿌 뿌 뿌
	gsub("\\S","뿌",a)	
\\w	영소문자, 영대문자, 숫자, _(언더바)	a<-"1,-,_, a b 2 3 가, 나, 다"
	gsub("\\w","뿌",a)	뿌,-,뿌, 뿌 뿌 뿌 뿌 뿌, 뿌, 뿌
\\W	영소문자, 영대문자, 숫자, _(언더바)를 제외한 모든 문자	a<-"1,-,_, a b 2 3 가, 나, 다"
	gsub("\\W","뿌",a)	1뿌뿌뿌_뿌뿌a뿌b뿌2뿌3뿌가뿌뿌나뿌뿌다

```
# ------영화명중 숫자가 있는 이름의 순위 평균 구하기 ---
영화명_숫자<-data[grep("\\d",data$영화명),]
mean(영화명_숫자$순위)
nrow(영화명_숫자)
nrow(영화명_숫자)/nrow(data)*100

# ------영화명중 숫자가 있는 이름에 비고 항목 1 입력---
비고 <- character(length(data$영화명))
비고[grep("\\d",data$영화명)]<-1
비고
data1<-data.frame(data,비고)
names(data1)
```

4. KoNLP를 이용한 한국어 형태소 분석

https://brunch.co.kr/@mapthecity/9

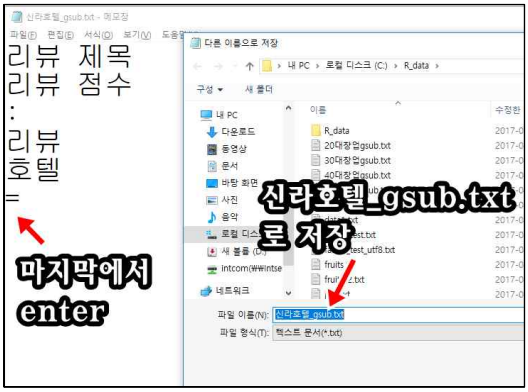
```
#------------KoNLP 형태소 분석기 패키지 설치---------
install.packages("KoNLP")
library(KoNLP)
useSejongDic()

#-----------형태소 나누기---------
aa<-"아버지가 방에 스르륵 들어가신다."
b_1<-extractNoun(aa)
b_1
b_2<-strsplit(aa," ")
b_2

#------------단어 사전에 추가--------------------------
mergeUserDic(data.frame(c("스르륵"),c("mag")))
mergeUserDic(data.frame(c("들어"),c("mag")))
extractNoun(aa)

sample<-"부성순 강사는"
extractNoun(sample)
mergeUserDic(data.frame(c("부성순"), "ncn"))
extractNoun(sample)

#----------문장 분석후 워드클라우드--------------------
aa<-"아버지가 방에 스르륵 들어가신다. 아버지는 방에 들어가신후 한참을 나오지 않았다"
bb<-extractNoun(aa)
bb
str(bb)
cc<-table(bb)
cc
names(cc)
wordcloud(names(cc),cc)
```

5. txt자료 불러와 wordcloud

```
data<-readLines("서울_신라호텔리뷰.txt")
data
data<-gsub("=","",data)
data
data<-gsub("리뷰 제목","",data)
data
data<-gsub("리뷰 점수","",data)
data
data<-gsub(":","",data)
data

data_형태소분석<-extractNoun(data)
data_형태소분석
str(data_형태소분석)


data_형태소_unlist<-unlist(data_형태소분석)
data_형태소_unlist
str(data_형태소_unlist)
data_최종 <- Filter(function(x) {nchar(x) >= 3} ,data_형태소_unlist)
#글자수가 3글자 이상인자료만

data_최종
data_집계<-table(data_최종)
data_집계
data_집계_큰값순<-sort(data_집계, decreasing=T)   #내림차순집계
data_집계_큰값순

data_집계
head(sort(data_최종, decreasing=T),20)

display.brewer.all(n=10, exact.n=FALSE)
display.brewer.pal(8,"Dark2")
palete<-brewer.pal(8,"Dark2")

barplot(data_큰값순)
wordcloud(names(data_큰값순),data_큰값순,random.order=F,color=palete,rot.per=0)
```

6. 여러단어 삭제시 gsub txt 파일로 작업하기

메모장에서 제거할 내용을 입력한뒤 마지막에서 enter 한후
파일-다른이름으로 저장, txt 파일로 저장한다.



gsub 코드
data<-readLines("서울_신라호텔리뷰.txt") data data<-gsub("=","",data) data data<-gsub("리뷰 제목","",data) data data<-gsub("리뷰 점수","",data) data data<-gsub(":","",data) data

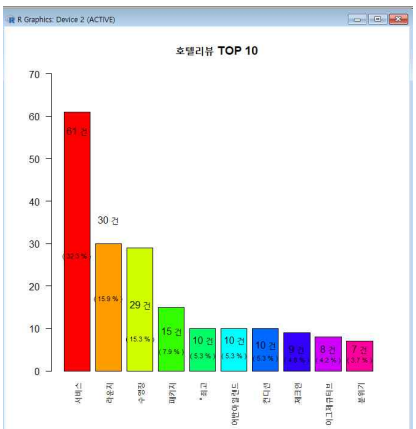
gsub 파일 이용

txt <- readLines("신라호텔_gsub.txt")
txt
cnt_txt <- length(txt)
cnt_txt
for(i in 1:cnt_txt) { #for(시작값 in 시작:마지막){ }
data <-gsub((txt[i]),"",data)
}
data
data<-gsub("\\d","",data)
data

7. 차트 작성

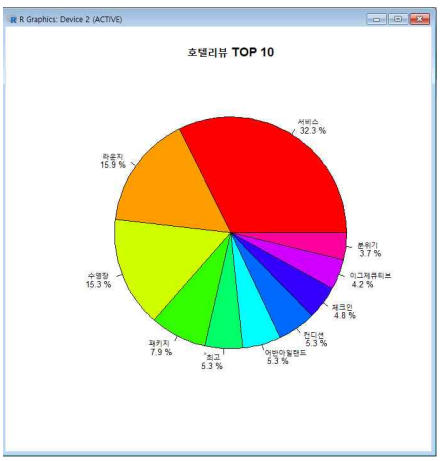
▶ 막대차트

```
data_집계_차트<-head(sort(data_집계, decreasing=T),10)
data_집계_차트

bp <- barplot(data_집계_차트,  main = "호텔리뷰 TOP 10 ", col = rainbow(10),
cex.names=1,  las = 2,ylim=c(0,70))

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

text(x = bp, y = a*1.05, labels = paste(data_집계_차트,"건"), col = "black", cex = 1)
text(x = bp, y = a*0.85, labels = paste("(",pct,"%",")"), col = "black", cex = 0.7)
```



▶ 원차트

```
lab <- paste(names(data_집계_차트),"\n",pct,"%")
pie(data_집계_차트,main="호텔리뷰 TOP 10",col=rainbow(10), cex=0.8,labels = lab)
```

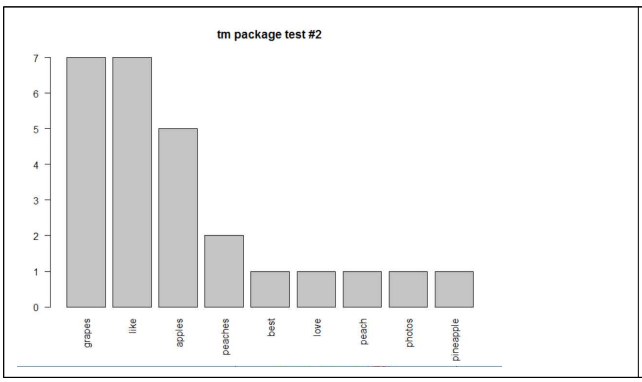


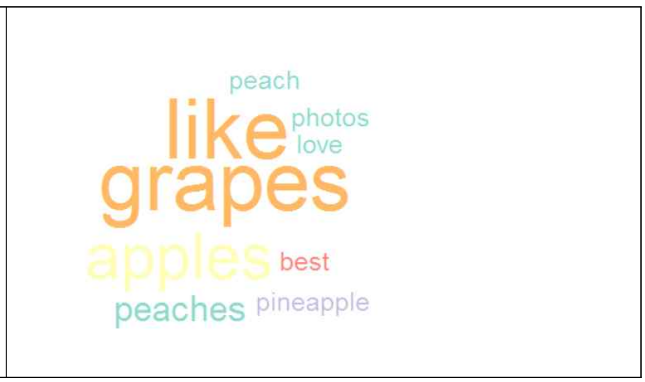
제 목	텍스트 마이닝 - 영어

좋아하는과일_영어 - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
I like apples and photos.
I like grapes grapes grapes grapes grapes are the best!
I like peaches and apples 555.
I like peaches and apples.
I like apples and grapes.
I like pineapple and peach 3333.
I like apples and grapes.
I love it
```






```
> install.packages("tm") : library(tm)
> data1<-readLines("좋아하는과일_영어.txt")
> length(data1) ; nchar(data1)

# Step 4. 위 4 줄을 tm 패키지가 처리할 수 있는 형태인 Corpus (말뭉치) 형태로 변환합니다.
# corp1 명령의 결과에서 documents : 8 부분이 중요합니다.
# document 란 tm 패키지가 작업할 수 있는 특별한 형태를 의미하며 일반적으로는
# 1 줄이 1개의 document 가 됩니다. 위의 경우 원본 파일이 총 8 줄이라 documents : 8 입니다
> corp1 <- Corpus(VectorSource(data1))   # 벡터이므로 VectorSource( ) 함수 사용함
> corp1                                  # Dataframe 의 경우 DataframeSource( ) 함수 씀.
  <<VCorpus>>
  Metadata:  corpus specific: 0, document level (indexed): 0
  Content:  documents: 4
> inspect(corp1)
  <<VCorpus>>
  Metadata:  corpus specific: 0, document level (indexed): 0
  Content:  documents: 8

  [[1]]
  <<PlainTextDocument>>
  Metadata:  7
  Content:  chars: 25

  [[2]]
  <<PlainTextDocument>>
  Metadata:  7
  Content:  chars: 55
```

 # tm 패키지가 분석 할 수 있는 Term-Document 형식의 Matrix 로 변환해야 합니다.

```
> tdm <- TermDocumentMatrix(corp1)
> tdm
<<TermDocumentMatrix (terms: 16, documents: 8)>>
Non-/sparse entries: 32/96
Sparsity           : 75%
Maximal term length: 9
Weighting          : term frequency (tf)
```

- terms : 16 은 총 16 개의 단어를 골랐다는 뜻이고 documents :8 는 소스가 8 개의 문장이라는 뜻입니다.
- sparsity 가 75% 는 tdm 안에 0 인 원소가 75% 라는 의미입니다.
- Term-Document Matrix 는 tm 패키지만 볼 수 있으므로 일반적으로 사용되는 Matrix 로 변환함 그래야 사람이 내용을 확인하기 쉽습니다.

```
> m <- as.matrix(tdm)
> m          # 1~8번행까지의 문장에서 출현되는 라인수, 3333은 6번째 라인에서만 글자가 나옴.
                Docs
Terms          1 2 3 4 5 6 7 8
  3333.        0 0 0 0 0 1 0 0
  555.         0 0 1 0 0 0 0 0
  and          1 0 1 1 1 1 1 0
  apples       1 0 1 0 1 0 1 0
  apples.      0 0 0 1 0 0 0 0
  are          0 1 0 0 0 0 0 0
  best!        0 1 0 0 0 0 0 0
  grapes       0 5 0 0 0 0 0 0
  grapes.      0 0 0 0 1 0 1 0
  like         1 1 1 1 1 1 1 0
  love         0 0 0 0 0 0 0 1
  peach        0 0 0 0 0 1 0 0
  peaches      0 0 1 1 0 0 0 0
  photos.      1 0 0 0 0 0 0 0
  pineapple    0 0 0 0 0 1 0 0
  the          0 1 0 0 0 0 0 0
```

 # 불필요한 and , but , not 같은 전치사 , 접속사 같은 불용어 제거
 # corpus 안에 있는 불용어나 제거 하고 싶은 단어를 제거하는 방법은 tm_map( ) 함수 사용

```
> corp2 <- tm_map(corp1,stripWhitespace) # 여러개의 공백을 하나의 공백으로 변환합니다
> corp2 <- tm_map(corp2,tolower) # 대문자가 있을 경우 소문자로 변환합니다
> corp2 <- tm_map(corp2,removeNumbers) # 숫자를 제거합니다
> corp2 <- tm_map(corp2,removePunctuation)  # 마침표,콤마,세미콜론,콜론 등의 문자 제거
> corp2 <- tm_map(corp2,PlainTextDocument)
> sword2 <- c(stopwords('en'),"and","but","not") # 기본 불용어 외 불용어로 쓸 단어 추가하기
> corp2 <- tm_map(corp2,removeWords,sword2) # 불용어 제거하기 (전치사 , 관사 등)
> tdm2 <- TermDocumentMatrix(corp2)
  # tdm3<-TermDocumentMatrix(corp2,control=list(wordLengths=c(1,Inf)))
```

```
> m2 <- as.matrix(tdm2)
> m2
> dim(m2)
> colnames(m2) <- c(1:8)
> m2
                    Docs
  Terms          1 2 3 4 5 6 7 8
  apples       1 0 1 1 1 0 1 0
  best         0 1 0 0 0 0 0 0
  grapes       0 5 0 0 1 0 1 0
  like         1 1 1 1 1 1 1 0
  love         0 0 0 0 0 0 0 1
  peach        0 0 0 0 0 1 0 0
  peaches      0 0 1 1 0 0 0 0
  photos       1 0 0 0 0 0 0 0
  pineapple    0 0 0 0 0 1 0 0
```

단어별 집계

```
> freq1 <- sort(rowSums(m2),decreasing=T)        # 컬럼별 Sum은 colSums( ) 함수를 사용
> head(freq1,3)
> barplot(freq1)
> pie(freq1)
> barplot(freq1,main="tm package test #2",las=2,ylim=c(0,max(freq1)))
> library(RColorBrewer)
> palete <- brewer.pal(7,"Set3")
> wordcloud(names(freq1),freq=freq1,scale=c(5,1),min.freq=1,colors=palete,random.order=F,
+ random.color=T)
```

13	위도와 경도를 이용하여 지도위에 위치 표시 지도차트	

위도와 경도 데이터를 이용하여 지도에 데이터의 크기를 원의 size에 적용하여 표현하는 지도차트를 작성할수 있다.

서울시에서 제공하는 자전거도로의 위치를 지도위에 표시해보자

서울시에서 제공하는 공공데이터 포털 [서울열린데이터 광장]-[오픈데이터]에 접속한다.

자전거를 검색한다.

서울시 자전거도로 위치정보의 [sheet]를 클릭한다.

csv 파일로 저장한다.

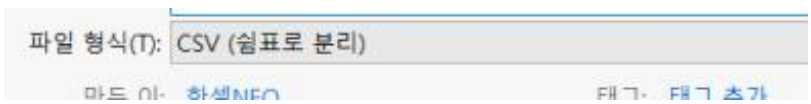
작업중인 R폴더에 자전거.csv로 저장한다.

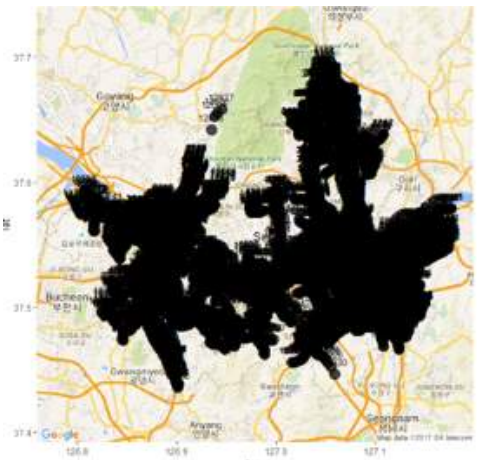
I	J	K	L	M	N	O	P	Q	R
도로종별코	주종코드	도로번호	일방통행여	차선수	도로명	고도값	자전거도로	경도	위도
13	0		1	1		20	3	126.9113	37.55876
13	0		1	1		0	1	127.0334	37.54581

LAT=위도
LON=경도

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

loc <- read.csv("자전거.csv")
loc
kor <- get_map("seoul", zoom=11, maptype = "roadmap")
LAT<-loc$위도
ON<-loc$경도
or.map <- ggmap(kor)+geom_point(data=loc, aes(x=LON, y=LAT),size=5,alpha=0.7)
kor.map + geom_text(data=loc, aes(x = LON, y = LAT+0.01, label=고유번호),size=3)
```

참고) 공공데이터 자료 가져올 때 에러가 생길시에는

	불필요한 열은 모두 제거하고 저장함
	파일-다른이름으로 저장csv로 저장함.

	
자전거도로 표시됨	**길이가 1000 보다 큰 곳만 표시**

```
loc <- read.csv("자전거csv",header=T)
head(loc)
summary(loc)
loc<-subset(loc,loc$길이m>1000)
loc
kor <- get_map("seoul", zoom=11, maptype = “roadmap“)
LAT<<-loc$위도
ON<-loc$경도
or.map <- ggmap(kor)+geom_point(data=loc, aes(x=LON, y=LAT),size=5,alpha=0.7)
kor.map + geom_text(data=loc, aes(x = LON, y = LAT+0.01, label=loc$길이m),size=3)
```

14	연관분석(장바구니 분석)	

예) 기저귀-맥주(미국월마트분석)

1. 고객들은 어떤 상품들을 동시에 구매하는가?
2. 라면을 구매한 고객은 주로 다른 어떤 상품을 구매하는가?

위와 같은 질문에 대한 분석을 토대로 고객들에게 SMS를 보낸다든가, 판촉용 전화를 한다든가 묶음 판매를 기획함.

이와 같은 질문에 대한 답은 연관규칙을 이용하여 구할 수 있습니다. 연관규칙은 상업 데이터베이스에서 가장 흔히 쓰이는 도구로, 어떤 사건이 얼마나 자주 동시에 발생하는가를 표현하는 규칙 또는 조건을 의미.

support: 지지도는 품목 A와 B를 동시에 구매할 확률인 P(A∩B)를 나타냅니다
confidence: 신뢰도는 품목 A가 구매하고나서, 품목 B 가 구매될 확률
lift: 향상도는 A를 구매한 사람이 B를 구매할 확률과 A의 구매와 상관없이 B를 구매할 확률의 비율
lift>1 이면 관련도가 높고 lift<1 이면 A구매자가 B를 구매하지 않을 확률이 높음

***연관분석, 장바구니 분석**

***지지도(Support): 전체 집합군에서 [조건] 자료가 포함된 집합수, 비율,**
[조건1]자료수 / 전체자료수

***신뢰도(Confidence): [조건1]가 있을때 [조건2]도 같이 있는 확률**
[조건1]->[조건2] 라고 하면
[조건1],[조건2] 가 같이나온 자료수/[조건1] 자료수

즉: [조건1],[조건2] 지지도 / [조건1] 지지도

***향상도(Lift:Improvement):**
[조건1][조건2]가 같이 나온 자료수/[조건1]자료수/전체자료수

사과를 구매한 고객이 치즈도 함께구매할 연관성에 대해 분석

지지도=P(A∩B)

▶ 지지도=[사과][치즈]가 같이 나온 자료/전체자료 => 1/4 =>0.25

구매자번호	제품명
1	사과
	치즈
	생수
2	생수
	호두
	치즈
	고등어
3	수박
	사과
	생수
4	생수
	호두
	치즈
	옥수수

신뢰도=P(A∩B)/P(A)

▶ 신뢰도=[사과][치즈]가 같이 나온 자료/[사과]자료 => 1/2 =>0.5

구매자번호	제품명
1	사과
	치즈
	생수
2	생수
	호두
	치즈
	고등어
3	수박
	사과
	생수
4	생수
	호두
	치즈
	옥수수

향상도=신뢰도(A,B)/지지도(B)

▶ 향상도=0.5/0.75=0.6666667

구매자번호	제품명
1	사과
	치즈
	생수
2	생수
	호두
	치즈
	고등어
3	수박
	사과
	생수
4	생수
	호두
	치즈
	옥수수

항목별 지지도[Support]

번호	제품명	지지도(자료수/4)	
1	고등어	1	0.25
2	사과	2	0.5
3	생수	4	1
4	수박	1	0.25
5	옥수수	1	0.25
6	치즈	3	0.75
7	호두	2	0.5

1~4번의 구매자번호별 항목 출현체크(오름차순정렬)

번호	항목, 구매자번호	1	2	3	4
1	고등어				
2	사과				
3	생수				
4	수박				
5	옥수수				
6	치즈				
7	호두				

연관분석 - 화장품전문점 패키지 구성방법?

분류	내용
예제 데이터	▪ B화장품전문점에서 판매된 트랜잭션 데이터
변수명	▪ 단일변수 - Nail Polish(매니큐어), Brushes(브러시), - Concealer(컨실러: 피부 결점을 감추어 주는 화장품) - Bronzer(피부를 햇볕에 그을린 것처럼 보이게 하는 화장품) - Lip liner(입술 라이너), Mascara(마스카라: 속눈썹용 화장품) - Eye shadow(아이섀도: 눈꺼풀에 바르는 화장품) - Foundation(파운데이션: 가루분), Lip Gloss(립글로스: 입술 화장품) - Lipstick(립스틱), Eyeliner(아이 라이너: 눈의 윤곽 그림)
분석문제	▪ 전체 트랜잭션 개수와 상품아이템 유형은 몇 개인가? ▪ 가장 발생빈도가 높은 상품아이템은 무엇인가? ▪ 지지도를 10%로 설정했을 때의 생성되는 규칙의 가지수는? ▪ 상품아이템 중에서 가장 발생확률이 높은 아이템과 낮은 아이템은 무엇인가? ▪ 가장 발생가능성이 높은 <2개 상품간>의 연관규칙은 무엇인가? ▪ 가장 발생가능성이 높은 <2개 상품이상에서> <제3의 상품으로>의 연관규칙은?

판매촉진 - 프로모션 효율화 방안

[우체국 쇼핑부문] 쇼핑몰 이용고객을 위한 추천상품 분석

분류	내용
예제 데이터	▪ 우체국 쇼핑에서 판매된 트랜잭션 데이터파일
변수명	▪ 단일변수: 의류(clothes), 냉동식품(frozen), 주류(alcohol,) 야채(veg), 제과(bakery), 육류(meat), 과자(snack), 생활장식(deco)에 대한 거래처리데이터
분석문제	▪ 전체 트랜잭션 개수와 상품아이템 유형은 몇 개인가? ▪ 가장 발생빈도가 높은 상품아이템은 무엇인가? ▪ 지지도를 10%로 설정했을 때의 생성되는 규칙의 가지수는? ▪ 상품아이템 중에서 가장 발생확률이 높은 아이템과 낮은 아이템은 무엇인가? ▪ 가장 발생가능성이 높은 <2개 상품간>의 연관규칙은 무엇인가? ▪ 가장 발생가능성이 높은 <2개 상품이상에서> <제3의 상품으로>의 연관규칙은?

연관성 분석 - R(연관분석할수 있게 트랜잭션으로 읽기)

```
install.packages(setwd("arules")
libarary(arules)
setwd("c:/data_r")
tr<-read.transactions ("장바구니분석소스.txt",format="basket",sep=",")
tr
class(tr)        # 4행7열로 이루어진 데이터임.
summary(tr)
inspect(tr)      # 트랜잭션 형태의 자료 아이템 확인
tr@itemInfo
tr@data
```

장바구니분석소스.txt - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
사과,치즈,생수
생수,호두,치즈,고등어
수박,사과,생수
생수,호두,치즈,옥수수
```

구매자번호	제품명
1	사과
	치즈
	생수
2	생수
	호두
	치즈
	고등어
3	수박
	사과
	생수
4	생수
	호두
	치즈
	옥수수

tr@data

1~4번의 구매자번호별 항목 출현체크(오름차순정렬)

번호	항목, 구매자번호	1	2	3	4
1	고등어		■		
2	사과	■		■	
3	생수	■	■	■	■
4	수박			■	
5	옥수수				■
6	치즈	■	■		■
7	호두		■		■

연관성 분석- R(연관분석에서 신뢰도 및 향상도)

rules=apriori(tr,parameter=list(supp=0.1,conf=0.1)) #지지도, 향상도 0.1 이상 자료
inspect(rules)

```
>     rules=apriori(tr,parameter=list(supp=0.3,conf=0.3))   #
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target  ext
        0.3    0.1    1 none FALSE            TRUE       5     0.3      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 1

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[7 item(s), 4 transaction(s)] done [0.00s].
sorting and recoding items ... [4 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [15 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>     inspect(rules)
     lhs          rhs     support confidence lift     count
[1]  {}        => {사과} 0.50    0.5000000  1.000000 2
[2]  {}        => {호두} 0.50    0.5000000  1.000000 2
[3]  {}        => {치즈} 0.75    0.7500000  1.000000 3
[4]  {}        => {생수} 1.00    1.0000000  1.000000 4
[5]  {사과}    => {생수} 0.50    1.0000000  1.000000 2
[6]  {생수}    => {사과} 0.50    0.5000000  1.000000 2
[7]  {호두}    => {치즈} 0.50    1.0000000  1.333333 2
[8]  {치즈}    => {호두} 0.50    0.6666667  1.333333 2
[9]  {호두}    => {생수} 0.50    1.0000000  1.000000 2
[10] {생수}    => {호두} 0.50    0.5000000  1.000000 2
[11] {치즈}    => {생수} 0.75    1.0000000  1.000000 3
[12] {생수}    => {치즈} 0.75    0.7500000  1.000000 3
[13] {치즈,호두} => {생수} 0.50    1.0000000  1.000000 2
[14] {생수,호두} => {치즈} 0.50    1.0000000  1.333333 2
[15] {생수,치즈} => {호두} 0.50    0.6666667  1.333333 2
```

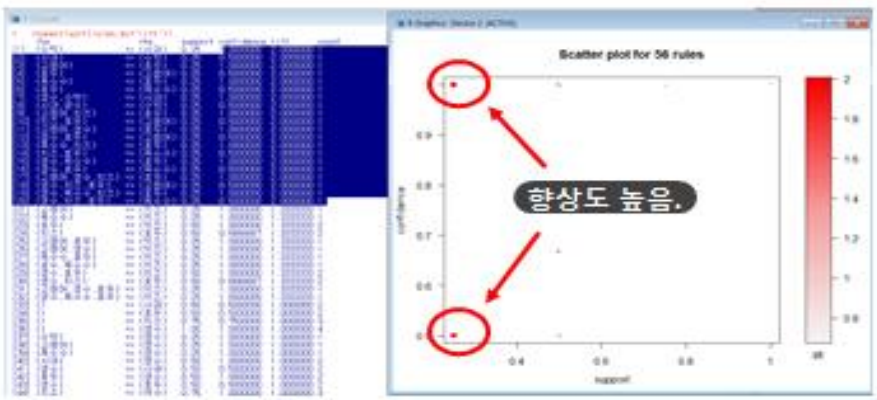
지지도 신뢰도 향상도

지지도, 신뢰도30%이상인 15개의 자료나옴
사과,치즈는 지지도가 0.25 이므로 나타나지 않음

치즈->생수
지지도: 0.75
신뢰도: 0.75
향상도: 1

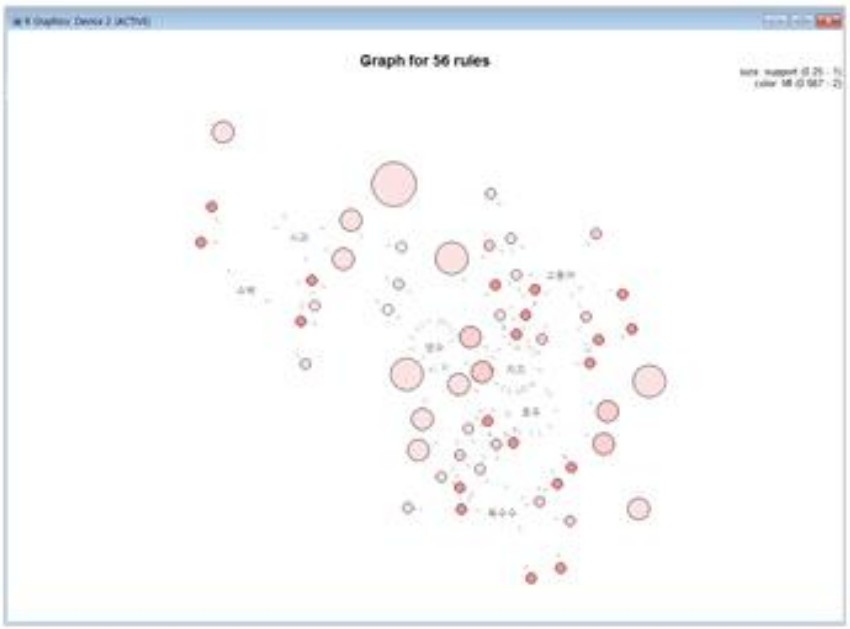
연관성 분석- R(시각화, 차트)

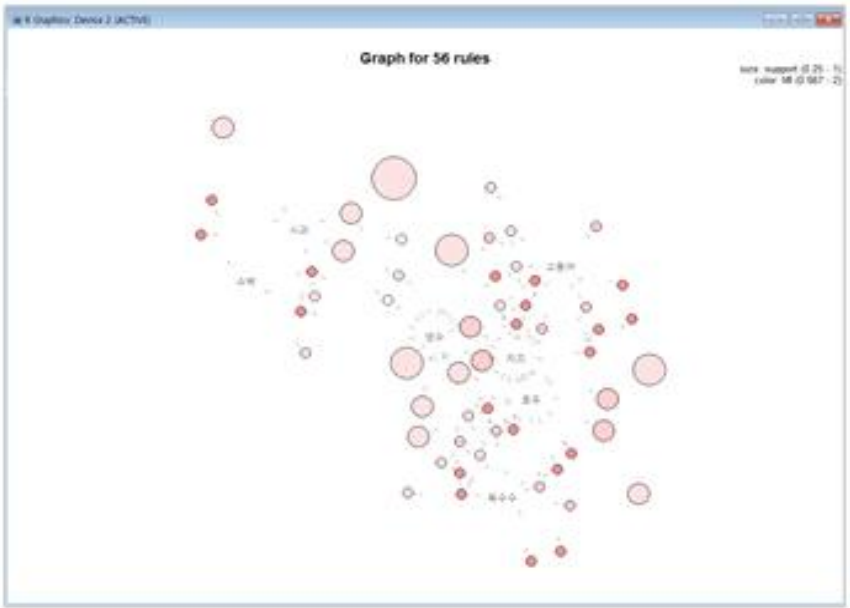
```
install.packages("tidyr")
library(tidyr)
install.packages("arulesViz")
library(arulesViz)
rules=apriori(tr,parameter=list(supp=0.25,conf=0.5))
inspect(rules)
 plot(rules)          # 가로(지지도), 세로(신뢰도), 색상(향상도)
                      #아래 자료는 지지도 0.25, 신뢰도 0.5와 1일때 향상도가 높음, 진한빨강색이 표시됨.
  inspect(sort(rules,by="lift"))
```



연관성 분석- R(시각화, 차트)

plot(rules,method="graph") #네트워크차트
각규칙별로어떤아이템들이 연관되어묶여있는지 보여주는네트워크그래프



연관성 분석- R(시각화, 차트)

plot(rules,method="graph") #네트워크차트
각규칙별로어떤아이템들이 연관되어묶여있는지 보여주는네트워크그래프