



제 1 장

R 패키지 소개

1.1 R 이란?

- 배열 및 행렬로 표현된 데이터에 대하여 효과적인 연산자를 이용해 자료의 분석, 시뮬레이션 및 시각적 표현에 유용한 객체지향적 프로그램이다.(<http://www.r-project.org/>)

1.2 R 소개

- R 프로그래밍 언어는 SAS, SPSS, MINITAB, ... 등과 같은 통계 소프트웨어이며, 통계 계산과 그래픽을 위한 프로그래밍 언어이자 소프트웨어 환경이다. 뉴질랜드 오클랜드 대학의 Robert Gentleman and Ross Ihaka에 의해 시작되어 현재는 R core team에 의해 개발되고 있다. R은 GPL 하에 배포되는 S 프로그래밍 언어의 구현으로 GNU S라고도 한다.

- R은 freeware이며, 사용자에게 의해 새롭게 개발될 수 있고 연구자들의 알고리즘을 획득하기 쉽다는 이유로 많은 학자에 의해 이용되어지고 있다.

- R은 3개의 창, R console, R editor, R graphics 화면으로 구성되어 있다.

1.3 R 설치

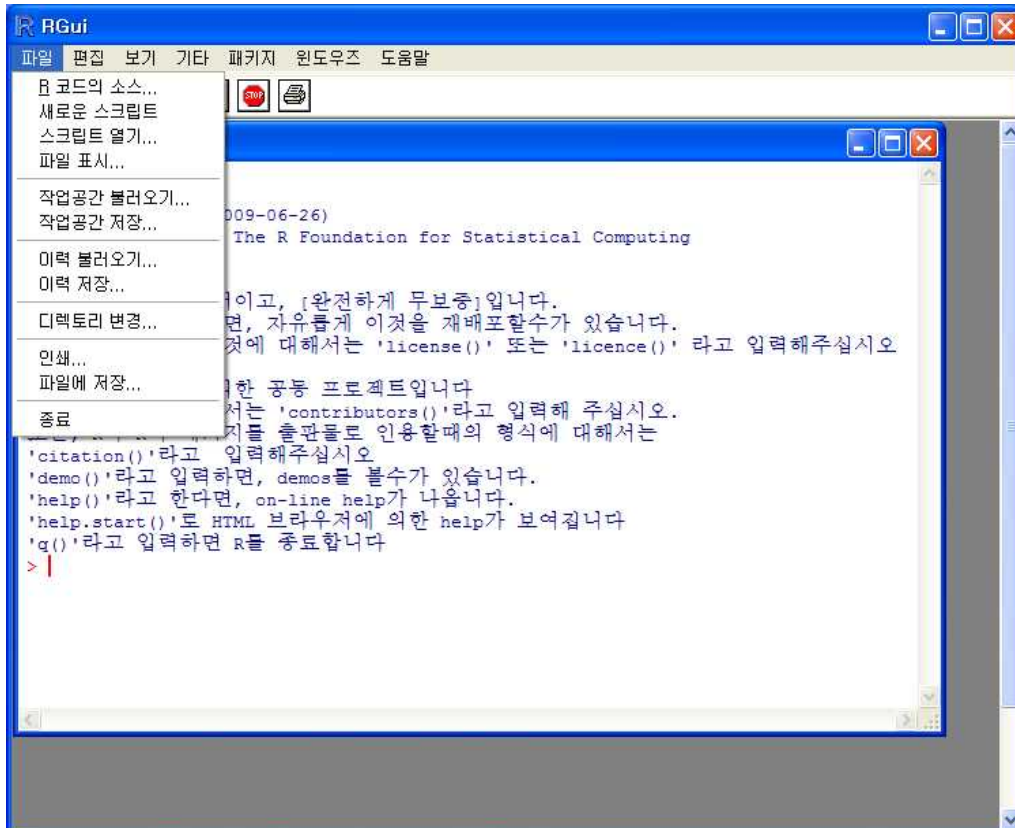
- R의 공식 홈페이지인 <http://www.r-project.org> 에서 무료로 다운받을 수 있으며, MS Windows 버전(한국미러)에서 base를 선택한 다음 R-2.9.1-win32.exe(2009년 7월 현재)를 선택하면 설치가 가능하다.

설치과정 중 Startup options에서 customized startup를 선택하고, Help Style에서 Plain text를 선택 후 설치를 하게 되면 R을 이용할 경우 도움을 받을 수 있다.

- 현재 개인에 의하여 한글과 수정작업이 원활히 이루어지고 있지만, 완벽하다고 할 수 없기 때문에 영문 GUI보다 못하다. 그렇기 때문에 설치과정에서 English로 설치를 하는 것을 권유한다. 또는 한글로 설치 후 편집(Edit) → GUI 설정...클릭 후 언어(Language for menus and messages)를 'English'로 지정한 다음 R을 다시 실행시키면 GUI가 영어로 바뀌게 되는 것을 확인할 수 있다.

1.4 R 프로그램의 구성

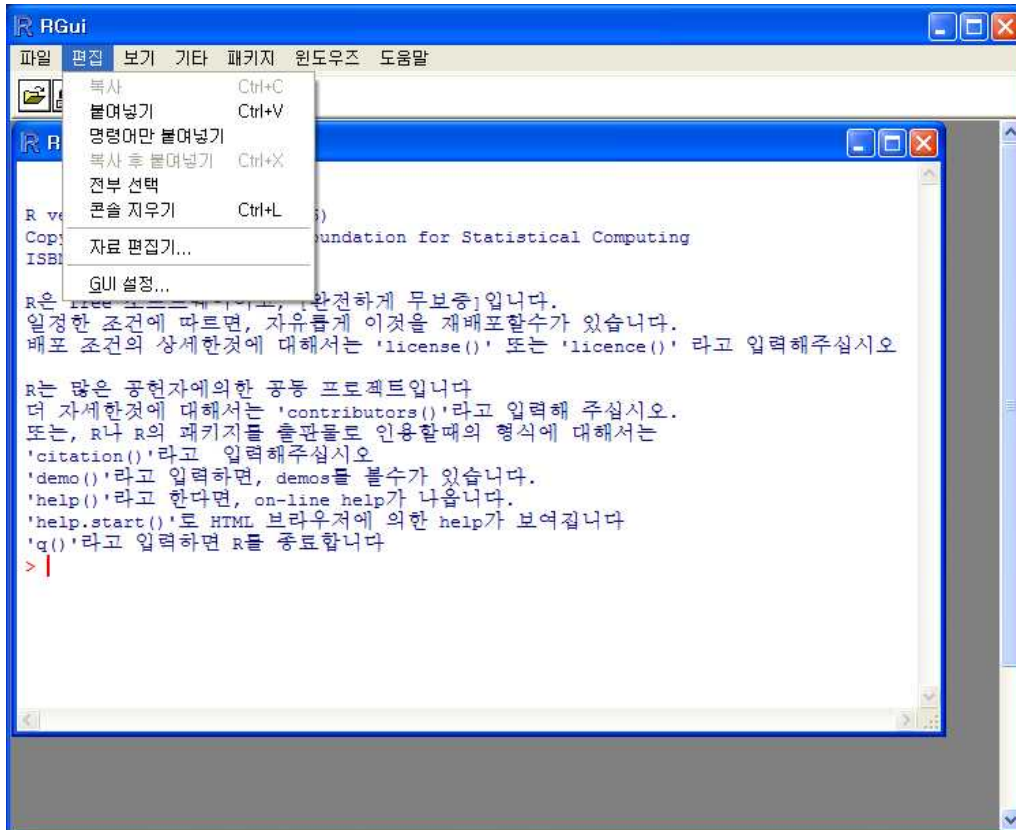
1.4.1 파일



R 코드의 소스	저장된 R 명령문을 R editor 창에 불러옴과 동시에 자동 실행한다.
새로운 스크립트	새로운 R 프로그램을 입력할 R editor 창을 연다.
스크립트 열기	저장된 R 명령문을 R editor 창에 불러오며 자동 실행은 하지 않는다.
파일 표시...	R이나 S 파일 등을 선택하여 새 창에 연다.
작업공간 불러오기	저장된 작업환경에 대한 객체를 불러온다.
작업공간 저장...	현재 작업환경의 객체를 확장자 .Rdata인 텍스트 파일로 저장한다.
이력 불러오기	R console 화면의 프로그램 이력을 불러온다.
이력 저장	R 프로그램을 구동 후 작업된 모든 프로그램 이력을 확장자 .History인 텍스트 파일로 저장한다.
디렉토리 변경...	R 프로그램의 작업 디렉토리를 변경한다.
인쇄...	화면에 보여지는 내용을 출력한다.
파일에 저장...	R console 화면의 내용을 저장하며 이때 이력과는 달리 출력결과를 포함한 화면에 나타난 모든 내용을 텍스트 파일로 저장한다.
종료	R 프로그램을 종료한다.

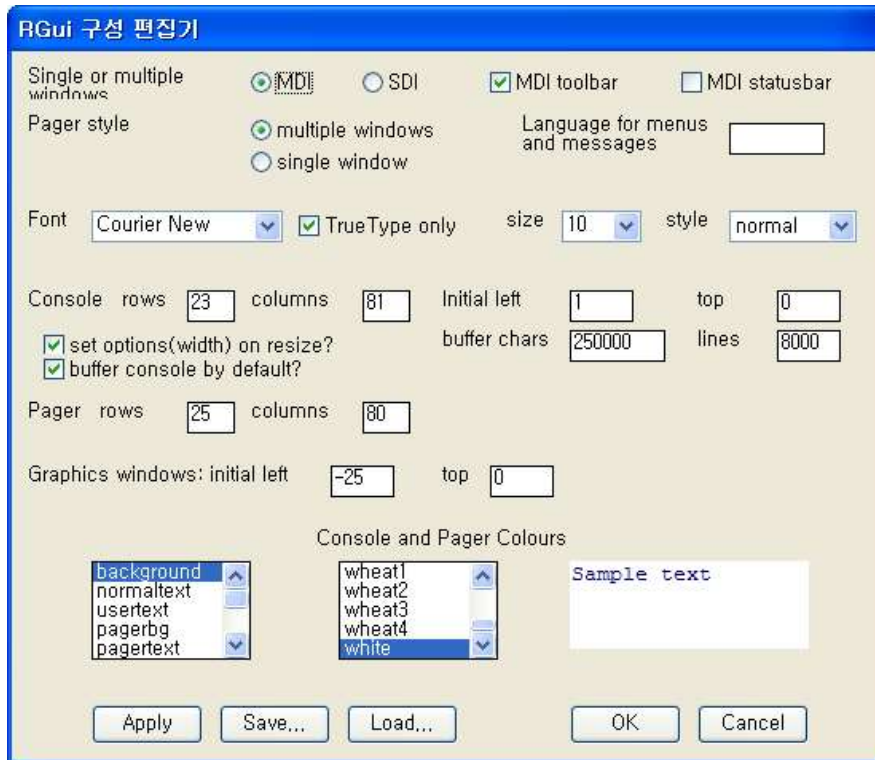
1.4.2 편집

- R console 화면의 편집기능에 대한 부 메뉴들로 구성되어 있다.



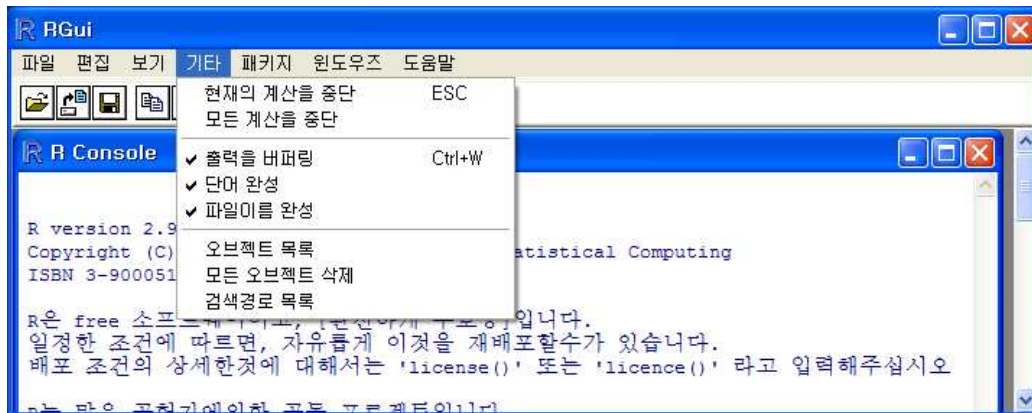
복사	선택된 내용을 복사한다.
붙여넣기	복사된 내용 전체를 붙여 넣는다.
명령어만 붙여넣기	선택된 내용 중 R 프로그램의 명령어만 붙여 넣는다.
복사 후 붙여넣기	선택된 내용을 복사하고 현재 커서가 위치한 곳에 붙여 넣는다.
전부 선택	화면 전체를 선택한다.
콘솔 지우기	현재 R console 화면의 내용을 모두 지운다.
자료 편집기	기존에 정의된 데이터를 에디터 창을 열어 나타낸다.
GUI 선택	화면의 크기와 색, 글꼴 모양 등 R 프로그램의 사용자 환경을 정의한다.

* GUI editor



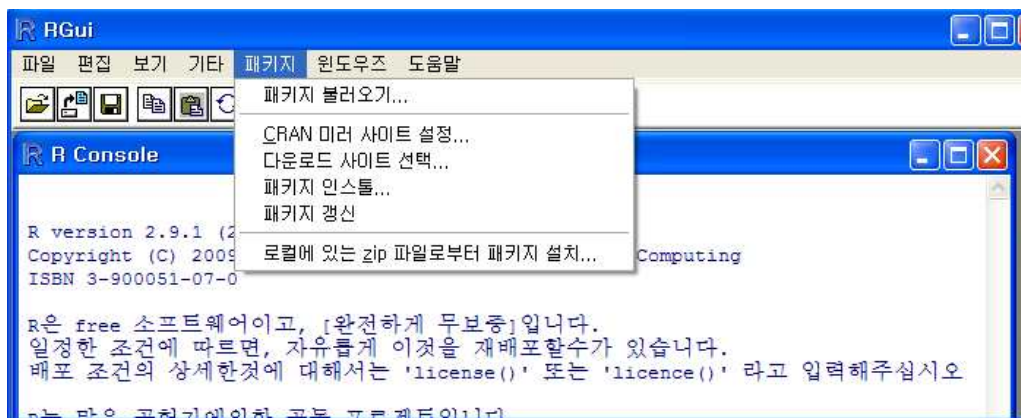
Single or Multiple Windows	R 프로그램의 실행에 따라 열리는 창의 개수에 대한 선택사항이다.
Pager style	R console 화면 이외에 추가로 열리는 화면에 대한 선택사항이다.
Console row	R console 화면의 크기를 지정하는 행수
Console columns	R console 화면의 크기를 지정하는 열수
Initial left	R 프로그램을 시작할 때 R console 화면의 좌우 위치
Initial top	R 프로그램을 시작할 때 R console 화면의 상하 위치
Buffer bytes, lines	R의 계산 효율성과 관련된 선택사항이다.
Pager rows, columns	R console 화면 이외에 추가로 열리는 화면의 크기를 지정하는 행수, 열수의 선택사항이다.
Console and Pager Colours	R console 화면과 다른 추가 화면의 색을 지정하는 것으로 화면의 바탕색, 출력 텍스트, 사용자 입력 프로그램, 추가 페이지의 제목줄의 색을 선택한다.

1.4.3 기타



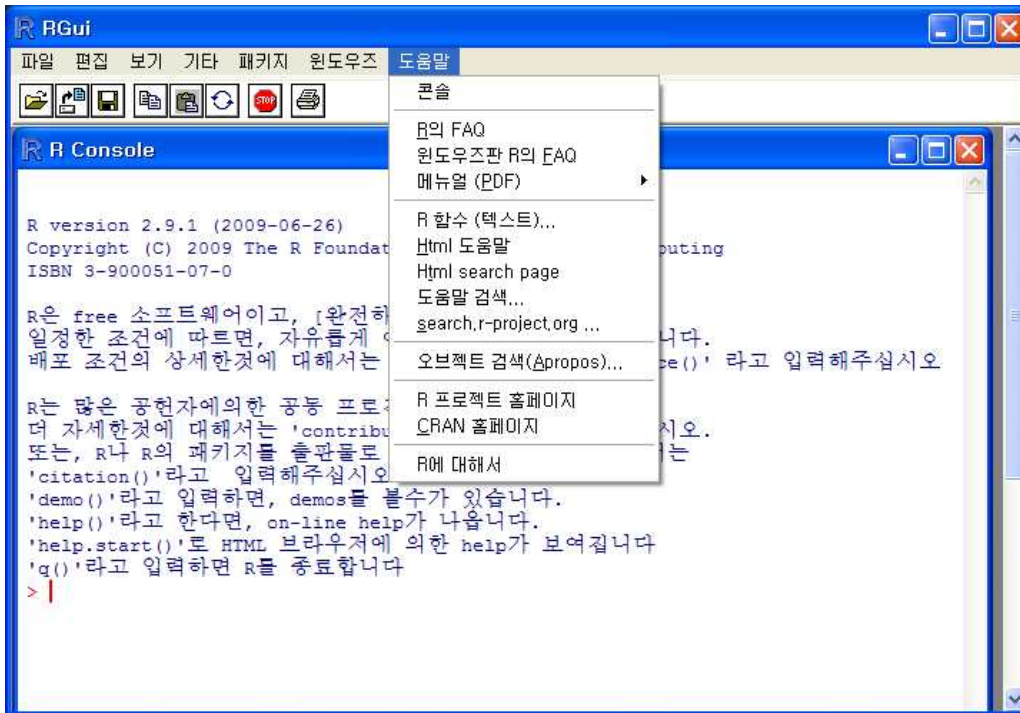
현재의 계산을 중단	현재 실행중인 작업을 중단한다.
출력을 버퍼링	이력의 저장과 관련된 선택메뉴이다.
오브젝트 목록	현재 작업환경의 모든 객체를 출력한다.
모든 오브젝트의 삭제	현재 작업환경의 모든 객체를 삭제한다.
검색경로 목록	현재 사용할 수 있는 객체들의 자원을 나타낸다.

1.4.4 패키지



패키지 불러오기	R 프로그램의 기본 패키지와 추천 패키지 중 불러올 패키지를 선택
CRAN 미러 사이트 설정	다운로드 하지 않은 패키지에 대해 다운로드할 CRAN 미러사이트 선택
다운로드 사이트 선택	분류된 패키지 다운로드 사이트를 선택한다.
패키지 설치	나열되는 패키지 중 설치를 원하는 것을 선택한다.
패키지 갱신	다운로드된 패키지를 최신화 한다. default는 graphics 패키지를 최신화 하는 것으로 함수표현은 update.packages(ask='graphics')이다.
로컬에 있는 zip 파일로부터 패키지 설치	zip파일 형태의 패키지를 선택하여 설치한다.

1.4.5 도움말



콘솔	R console 화면의 편집기능과 관련된 도움말 창을 연다.
R의 FAQ	html 형식의 R FAQ 화면을 연다.
윈도우즈판 R의 FAQ	Windows 운영환경 하에서의 html 형식의 R FAQ 화면을 연다.
메뉴얼 (PDF)	pdf 형식의 메뉴얼 파일을 연다.
R의 함수 (텍스트)	입력된 함수에 대한 텍스트 형식의 도움말 창을 연다.
Html 도움말	주제별로 구분된 html 형식의 도움말을 연다.
Html search page	html 형식의 검색 page를 연다.
도움말 검색	주어진 단어를 포함하는 도움말을 R 정보창을 열고 출력한다.
search.r-project.org	search.r-project.org의 웹사이트를 통해 주어진 단어를 검색한다.
오브젝트 검색	주어진 단어를 포함하는 객체를 R console 화면에 출력한다.
R 프로젝트 홈페이지	웹페이지 http://www.r-project.org/ 를 연다.
CRAN 홈페이지	웹페이지 http://www.cran.r-project.org/ 를 연다
R에 대해서	R 프로그램의 정보를 나타내는 창을 연다.

1.5 기초 연산

- R을 처음 시작하면 >라는 프롬프트(prompt)가 나타난다. 프롬프트는 사용자의 입력을 기다린다는 표시이다. 간단한 명령을 살펴보자.

```
> 1+2
[1] 3
> 3-1
[1] 2
> 1*2
[1] 2
> (1*3)/2
[1] 1.5
> {2*3}^2
[1] 36
> 1+2;3+4
[1] 3
[1] 7
> x=1
> y=2
> x+y
[1] 3
```

- 결과를 살펴보면, +는 덧셈, -는 뺄셈, *는 곱셈, /는 나눗셈이며 ^는 거듭제곱을 뜻한다. 사칙연산에 대한 연산자나 연산자간의 우선순위는 일반적인 프로그래밍 언어와 같으며 소괄호('()')를 통해 제어한다.

- 출력결과에 '[1]'이 항상 표현되고, 이는 출력물인 데이터에 대한 인덱스(index)로 결과의 자리수를 확인할 수 있다.

- 한 명령문의 마침은 ';'을 이용하여 명령문의 종료를 나타내도록 한다.

- '#' 표시가 붙으면 이하의 내용은 주석으로 프로그램 실행에서 무시한다.

- 변수에 값을 할당(assign)할 수 있다. 변수의 이름은 알파벳과 숫자, 밑줄과 마침표로 지을 수 있으나 첫 글자는 알파벳으로 시작해야 한다.

- 대소문자를 구분함으로 주의해서 사용해야 한다.

- 명령어의 길이가 길 경우는 연결 프롬프트로 '>'대신 '+'가 나타난다. 입력받은 명령어가 불완전할 경우 자동적으로 생성한다.

- 주요 연산자

연산자	기능
{	블록정의
(괄호기능
\$	성분추출
[[[첨자표현
+ - * /	더하기, 빼기, 곱하기, 나누기
^ **	제곱 연산자
%*% %/% %%	행렬의 곱, 몫, 나머지 연산자
< > <= => == !=	비교 연산자
!	부정 연산자
& &	논리 연산자
<<-	전역 할당 연산자
<- = ->	할당 연산자

1.6 수학 함수

수학 함수	기능
abs(x)	x 절댓값
ceiling(x)	x보다 큰 수 중 가장 작은 정수
floor(x)	x보다 작은 수 중 가장 큰 정수
trunc(x)	0과 x 사이의 가장 큰 정수를 출력한다.
round(x, digits=y)	x의 소수점 (y+1)자리에서의 반올림
signif(x, digits=y)	10의 지수형태의 표현으로 반올림
exp(x)	지수함수(exponential function)
log(x), log10(x), log2(x), log(x, base=y)	밑(base)이 자연대수 e, 10, 2, y인 로그함수
sign(x)	부호함수 (±)
sqrt(x)	제곱근 함수
factorial(x)	x의 계승 출력(x!)
choose(x,y)	x에서 y를 고르는 조합의 수 출력
pi	원주율
beta(a,b)	베타함수
gamma(x)	감마함수
cos(x), sin(x), tan(x), acos(x), asin(x), atan(x), atan2(y,x)	삼각함수

- 수학 함수를 이용한 연산

```
> abs(-2)
[1] 2
> round(0.6); round(1.25,1)
[1] 1
[1] 1.2
> sqrt(10^2); 10^2;
[1] 10
[1] 100
> log(2); log2(2); log10(2); log(2,base=10);
[1] 0.6931472
[1] 1
[1] 0.30103
[1] 0.30103
> sign(10); -10*sign(10);
[1] 1
[1] -10
> choose(5,2); factorial(5); 5*pi;
[1] 10
[1] 120
[1] 15.70796
```

1.7 할당

할당 : 변수나 객체에 값을 정의하는 것을 말한다.

- 할당 연산자로 '=', '<-', '<<-'를 사용하며, 함수 내에서 '=', '<-'에 의해 할당된 값은 함수가 수행된 후에 저장되지 않으나, '<<-'을 통해 할당된 객체 값은 사라지지 않는다.

```
> exp.x <- exp(x=1); exp.x; x
[1] 2.718282
[1] 1
> exp.x <- exp(x<-1); exp.x; x
[1] 2.718282
[1] 1
> exp.x <- exp(x<<-2); exp.x; x
[1] 7.389056
[1] 2
```

1.8 객체 관리

- ls() : workspace에 있는 모든 객체들의 목록을 보여준다.
- search() : 현재 작업환경에서 구동된 패키지들을 문자 리스트로 출력한다.
- as.environment() : 지정된 패키지의 속성을 나타내는 함수이다.
- rm() : 현재 작업환경의 모든 객체를 제거한다.

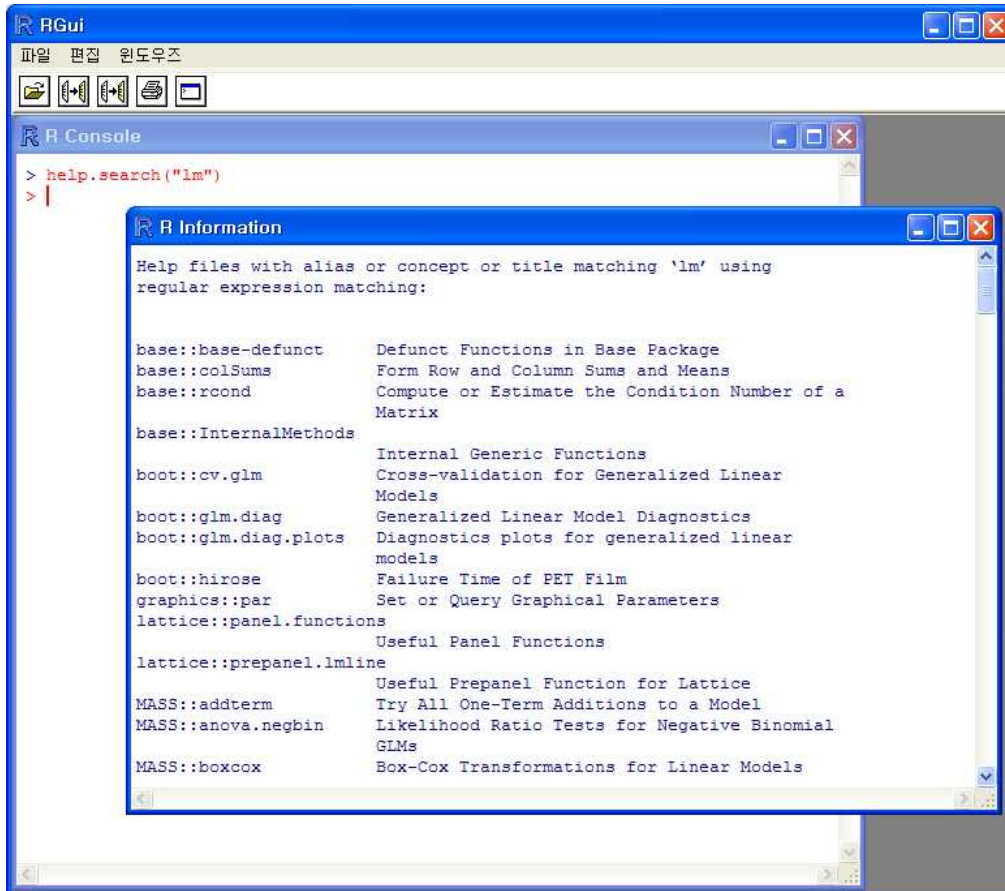
```
> ## 객체관리
> search()
[1] ".GlobalEnv"      "package:stats"    "package:graphics"
[4] "package:grDevices" "package:utils"    "package:datasets"
[7] "package:methods" "Autoloads"        "package:base"
> ls("package:stats", pattern="dist")
[1] "as.dist"          "cooks.distance"  "dist"
> objects("package:stats", pattern="dist")
[1] "as.dist"          "cooks.distance"  "dist"
> ls(pos=2, pattern="dist")
[1] "as.dist"          "cooks.distance"  "dist"
> as.environment(2)
<environment: package:stats>
attr(,"name")
[1] "package:stats"
attr(,"path")
[1] "C:/PROGRA~1/R/R-29~1.1/library/stats"
> ls(as.environment(2), pattern="dist")
[1] "as.dist"          "cooks.distance"  "dist"
> ls()
[1] "변수.x" "exp.x"  "x"      "x.y"
```

1.9 패키지의 이용

- R의 가장 큰 장점은 단시일 내에 업데이트 되는 패키지의 이용 가능하다.
- 패키지 인스톨은 풀다운 메뉴중 패키지 -> package(s) 인스톨을 이용한다.
- R console에서 install.package("패키지명")를 이용하여 직접적인 인스톨이 가능하다.
- 패키지 인스톨 후 library() 함수 또는 require() 함수를 이용해 패키지를 구동한다.
- R 콘솔 화면의 풀다운 메뉴중 패키지 -> 패키지 불러오기를 이용하여 패키지 구동 가능.

1.10 도움말의 이용

- help() 혹은 ? : 함수의 기능이나 필요한 인수들에 대한 내용을 볼 수 있다.
- help.search() : 원하는 단어를 포함하는 내용을 검색하고자 할 때 사용한다.



1.11 R 커맨더 : R을 SPSS처럼 사용하기

1.11.1 소개와 설치

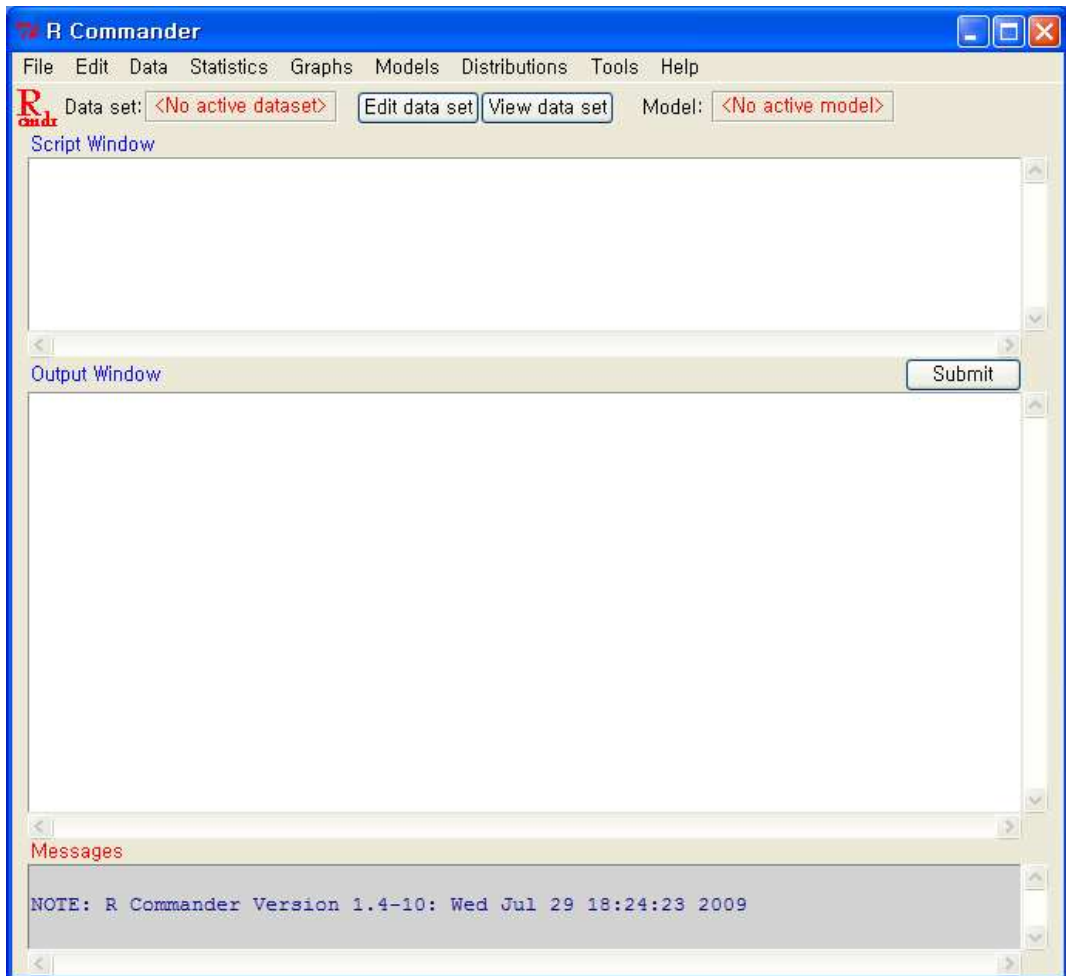
- R의 가장 큰 단점은 불편한 사용법이다. 명령어를 일일이 입력해서 자료를 분석하는 방법은 익숙해지면 메뉴 방식보다 편리하지만 익숙해지는데 걸리는 시간은 길다.

- 'R 커맨더'라는 패키지를 설치하면 R을 SPSS에서처럼 메뉴만 클릭해서 사용할 수 있다. R의 명령 창에 아래와 같이 입력하면 설치된다.

```
## 패키지 설치
> install.packages("Rcmdr",dependencies=T)

## 불러오기
> library(Rcmdr)
```

- 실행화면



1.11.2 사용법

※ R 커맨더는 위에서부터 다섯 부분으로 나뉘져있다.

- 메뉴 : 원하는 작업을 선택한다.
- 자료와 모형 : 작업할 자료나 모형을 선택하고 편집할 수 있다.
- 명령창(script window) : 메뉴를 선택하면 해당하는 R 명령이 이 창에 표시된다. 사용자가 직접 명령을 입력할 수도 있다.
- 결과창(output window) : 실행 결과가 표시된다.
- 메시지창(message window) : 에러 메시지 등이 표시된다.

※ SPSS와 사용법이 거의 동일하다. 단, SPSS는 한 번에 하나의 파일만 불러들일 수 있는 반면, R커맨더는 여러 개의 자료를 불러들일 수 있다. 대신 어떤 자료를 분석할 것인가에 대하여 선택하여야 한다.

1.11.3 직접 명령하기

사용자가 직접 명령을 입력할 때는 다음과 같이 한다.

- R 커맨더의 명령창에 명령을 입력
- 마우스로 드레그하여 선택
- 마우스 오른쪽 버튼을 눌러 "submit"을 선택

그냥 R 의 원래 명령창에 명령을 입력해도 된다.

제 2 장

R의 자료구조

2.1 벡터(vector)

- R은 통계용 언어이므로 하나의 값을 다루는 경우보다 여러 개의 값을 한 번에 다루어야 하는 경우가 더 많다. 여러 개의 값을 나타내는 방법에는 벡터, 리스트, 매트릭스, 데이터프레임 등 다양한 방법이 있다. 가장 기본은 벡터로서 한정된 개수의 값을 표현한다.

- `c()` : 나열된 데이터나 객체들을 하나의 객체로 결합하는 함수이다. 각 원소(element)는 `'`로 구분한다.

- 자료의 입력

```
> ## 자료의 입력
> c(1,2,3)
[1] 1 2 3
> c(1,2,3,4,5)
[1] 1 2 3 4 5
```

- 벡터와 값의 사칙연산

```
> ## vector와 값의 사칙연산
> c(1,2,3) + 2
[1] 3 4 5
```

- 벡터와 벡터의 사칙연산

```
> ## vector & vector
> c(1,2,3) + c(4,5,6)
[1] 5 7 9
```

- **길이가 다른 벡터** : 짧은 쪽을 처음으로 다시 적용한다. 긴 벡터의 길이가 짧은 벡터의 길이의 배수인 것이 원칙이지만 그렇지 않더라도 경고(warning)만 뜰 뿐 계산되어진다.

```
> c(1,2) + c(3,4,5,6)
[1] 4 6 6 8
> c(1,2,3) + c(4,5,6,7)
[1] 5 7 9 8
Warning message:
In c(1, 2, 3) + c(4, 5, 6, 7) :
  longer object length is not a multiple of shorter object length
```

- **‘:’** : 단위가 1인 등차수열을 나타내고자 할 때 ‘:’를 이용하여 간단하게 나타낸다.

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

- **seq()** : 단위에 관계없이 모든 등차수열을 나타낼 수 있다.
seq(끝), seq(시작, 끝), seq(시작, 끝, 간격)과 같은 형태로 사용한다.
seq()에서 by와 length는 동시에 사용할 수 없다.

```
> seq(10)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(1,5)
[1] 1 2 3 4 5
> seq(1,2,0.1)
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0
> seq(10,1)
[1] 10 9 8 7 6 5 4 3 2 1
```

- **rep()** : 주어진 벡터 객체를 반복하여 자료를 생성하는 함수이다.
 - rep(값, 횟수) : 값을 횟수만큼 반복한다.
 - rep(벡터, 횟수) : 벡터를 횟수만큼 반복한다.
 - rep(벡터, 벡터) : 첫 벡터의 값을 다음 벡터에 있는 같은 자리의 횟수만큼 반복한다.
 - length 옵션을 주면 반복해서 해당 길이만큼 되도록 한다.

```

> rep(1,5)
[1] 1 1 1 1 1
> rep(1:3, times=2)
[1] 1 2 3 1 2 3
> rep(c(1,2,3), times=2, each=2)
[1] 1 1 2 2 3 3 1 1 2 2 3 3
> rep(c(1,2,3),c(1,2,3))
[1] 1 2 2 3 3 3
> rep(1:10, times=2, each=3, length.out=5)
[1] 1 1 1 2 2
> rep(1:10, times=2, each=3, length.out=10)
[1] 1 1 1 2 2 2 3 3 3 4
> rep(1:10,3,2)
[1] 1 2

```

- R은 함수의 중첩이 가능하다.

- paste() : 문자형 벡터의 자료 입력에 유용한 함수이다
- sequence() : 주어진 벡터 객체의 각 요소에 seq() 함수를 수행하는 함수로 seq() 안의 숫자를 수열의 상한값으로 하여 수열을 생성한다.

```

> as.character(2:5)
[1] "2" "3" "4" "5"
> paste(1:6)
[1] "1" "2" "3" "4" "5" "6"
> paste("A", 1:3, sep="")
[1] "A1" "A2" "A3"
> paste("A", 1:4, 1:8, sep="_")
[1] "A_1_1" "A_2_2" "A_3_3" "A_4_4" "A_1_5" "A_2_6" "A_3_7" "A_4_8"
> paste(paste(1:3), 1:4, 6:9, sep="+")
[1] "1+ 1+6" "2+ 2+7" "3+ 3+8" "1+ 4+9"
> sequence(1)
[1] 1
> sequence(1:5)
[1] 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5
> sequence(c(1,9,2))
[1] 1 1 2 3 4 5 6 7 8 9 1 2

```


- `scan()` : R console 창이나 파일로부터 벡터나 리스트에 자료를 입력하는 함수이다.

```
> a<-scan()
1: 1
2: 2
3: 1 2 3
Read 5 items
> a
[1] 1 2 1 2 3
> b<-scan(what=numeric(0))
1: 1 NA 3
Read 3 items
> b
[1] 1 NA 3
> x<-scan(what="")
1: Nam Mun Hong
Read 3 items
> x
[1] "Nam" "Mun" "Hong"
> is.character(x)
[1] TRUE
```

- **벡터 객체의 속성**

벡터 객체의 속성은 자료의 유형(mode), 길이(length), 데이터 원소의 이름(names)이 있다.

```
> score <- c("A", "B+", "A", "C", "D")
> score
[1] "A" "B+" "A" "C" "D"
> mode(score)
[1] "character"
> length(score)
[1] 5
> names(score) <- scan(what="")
1: 남 작 홍 문 이
6:
Read 5 items
> names(score)
[1] "남" "작" "홍" "문" "이"
> score
남 작 홍 문 이
"A" "B+" "A" "C" "D"
```

2.2 데이터의 유형

2.2.1 데이터의 기본 유형

- 수치형(numeric) : 숫자로 이루어졌으며 정수형(integer)과 실수형(double)으로 구분한다.
 - 논리형(logical) : 참(TRUE)이나 거짓(FALSE)의 논리값을 나타낸다.
 - 문자형(character) : 문자나 문자열을 나타낸다.
 - 복소수형(complex) : 실수와 허수로 구성된 복소수를 나타낸다.
- mode() : 데이터의 유형을 나타내 주는 함수이다

```
> ## 데이터의 유형
> mode(3)
[1] "numeric"
> mode(pi)
[1] "numeric"
> mode(3>4)
[1] "logical"
> mode(mode(3))
[1] "character"
> mode(TRUE)
[1] "logical"
> mode(FALSE); mode(F)
[1] "logical"
[1] "logical"
> mode(true); mode(False)
이하에 에러mode(true) : 오브젝트 'true'가 없습니다
> mode(1+ 2i); mode(1+ 0i); mode(1+ (2+ 3i))
[1] "complex"
[1] "complex"
[1] "complex"
```

- R은 데이터의 기본적인 속성을 데이터의 유형(mode)과 길이(length)로 나타내며, 여기서 길이는 데이터의 개수를 나타낸다.
 - length(x) : 데이터의 길이를 나타내는 함수이다.
- 데이터의 유형을 검증하는 함수는 is를 이용한다.
- 데이터의 유형이 실수인지를 확인하려면 is.double(x)를 이용하며, 이때 결과는 TRUE 혹은 FALSE로 나타난다.

- 데이터 유형 검증 함수

is.numeric(x)	수치형 여부	is.na(x)	NA 여부
is.double(x)	실수형 여부	is.null(x)	NULL 여부
is.integer(x)	정수형 여부	is.nan(x)	NaN 여부
is.logical(x)	논리형 여부	is.infinite(x)	무한 수치 여부
is.complex(x)	복소수형 여부	is.finite(x)	유한 수치 여부
is.character(x)	문자형 여부		

2.2.2 특수 데이터

- NULL : 비어있는 값으로 데이터 유형도 없으며 자료의 길이도 0임.
- NA : 결측값(missing value).
- NAN : 수학적으로 정의가 불가능한 수 (예: $\sqrt{-3}$).
- Inf, -Inf : 양의 무한대와 음의 무한대.
- 특수형태의 데이터들과의 연산 결과는 보통 특수형태의 데이터들이 되고 데이터의 형식은 연산에 사용된 다른 값들의 유형과 동일하게 된다.

```
> sqrt(-3)
[1] NaN
Warning message:
In sqrt(-3) : NANs가 작성되었습니다
> 1/0
[1] Inf
> mode(NULL); length(NULL)
[1] "NULL"
[1] 0
> mode(NA); length(NA)
[1] "logical"
[1] 1
> mode(NaN); length(NaN)
[1] "numeric"
[1] 1
> mode(NULL+0); length(NULL+0)
[1] "numeric"
[1] 0
> mode(NA+3); length(NA+3)
[1] "numeric"
에러:오브젝트 'Na'가 없습니다
> 1/Inf
[1] 0
```

2.2.3 데이터 유형의 변경

- 서로 다른 유형의 데이터에 대한 연산결과는 R 프로그램에 의해 자동으로 하나의 유형으로 정의된다. 이러한 현상을 데이터 유형의 강제변환이라고 부르며 값의 유형별 우선순위는 다음과 같다.

문자형 > 복소수형 > 수치형 > 논리형

- R에서 유형 변환은 as.~ 함수를 이용하며, 가능한 함수는 다음과 같다.

- as.numeric(x) : 수치형으로 변환
- as.logical(x) : 논리형으로 변환
- as.double(x) : 실수형으로 변환
- as.complex(x) : 복소수형으로 변환
- as.integer(x) : 정수형으로 변환
- as.character(x) : 문자형으로 변환

```
> FALSE + 3
[1] 3
> TRUE+(1+ 2i)
[1] 2+ 2i
> paste(TRUE, "is 1")
[1] "TRUE is 1"
> 3+(5+ 0i)
[1] 8+ 0i
> paste(3,"is three")
[1] "3 is three"
> paste(1+ 2i,"is complex")
[1] "1+ 2i is complex"
> a <- "3"; a
[1] "3"
> mode(a)
[1] "character"
> as.numeric(a)
[1] 3
> mode(as.numeric(a))
[1] "numeric"
> b<-as.numeric(a)
> mode(b)
[1] "numeric"
```

2.3 자료 객체

- R은 여러 가지 특수한 종류의 자료구조를 수용할 수 있도록 다양한 형태의 자료객체를 정의하고 있다. 각 자료 객체는 자료의 구조나 여러 데이터 유형의 포함 여부에 따라 다음과 같이 나누어진다.

자료 객체	구성차원	자료 유형	복수 데이터 유형 적용 여부
벡터(vector)	1차원	수치/문자/복소수/논리	불가능
행렬(matrix)	2차원	수치/문자/복소수/논리	불가능
데이터 프레임 (data frame)	2차원	수치/문자/복소수/논리	가능
배열(array)	2차원 이상	수치/문자/복소수/논리	불가능
요인(factor)	1차원	수치/문자	불가능
시계열(time series)	2차원	수치/문자/복소수/논리	불가능
리스트(list)	2차원 이상	수치/문자/복소수/논리/ 함수/표현식/call 등	가능

- 데이터 분석에 주로 사용되는 자료 객체는 벡터, 행렬, 데이터 프레임 등이며 추가로 배열, 범주형 자료에 대한 요인(factor), 시계열 자료 객체들이 있고 리스트의 경우는 객체 특성의 요약 등에 주로 활용되어 그래프나 출력결과에 대한 객체들이 리스트의 자료 구조를 갖는다.

2.4 행렬(matrix)

- 동일한 유형의 자료 값으로 구성되며 행과 열의 2차원 자료구조를 갖는다.
- 행렬은 수치형, 문자형, 복소수형, 논리형의 자료 유형 중 한 가지 유형에 대한 데이터 입력만 가능하다.

2.4.1 자료의 입력 : 벡터 객체 이용

- 벡터 객체를 이용해 행렬 데이터를 입력하는 방법은 함수 `cbind()`와 `rbind()`가 있다.
- `cbind()` : 각각의 벡터를 열로 결합하여 matrix를 만든다.
- `rbind()` : 각각의 벡터를 행으로 결합하여 matrix를 만든다.

2.4.2 자료의 입력 : `matrix()`, `dim()` 함수 이용

- 함수 `matrix()`와 `dim()`은 주어진 차원에 따라 행렬 객체를 생성하는 함수이다.

* 함수 예제

```
matrix(data=NA, nrow=1, ncol=1, byrow=FALSE, dimnames=NULL)
```

- matrix() 함수는 모두 5개의 인수를 가진다.

-* 인수 설명

data : 벡터 객체 이름

nrow : 양의 정수로 원하는 행의 수(디폴트 1)를 정의한다.

ncol : 양의 정수로 원하는 열의 수(디폴트 1)를 정의한다.

byrow : 논리값으로 FALSE(디폴트)이면 행부터 데이터를 채우고 TRUE이면 열부터 데이터를 채운다.

dimnames : 각 행과 열의 이름을 정의하는 길이 2의 리스트 객체로 정의된다.

- 함수의 인수를 정의하지 않으면 디폴트로 지정된 값이 적용된다.

- 함수의 인수 위치만 맞으면 인수 이름을 사용하지 않아도 무관하다. 하지만 인수의 위치와 맞지 않을 때에는 반드시 인수이름을 같이 지정해 줘야 한다.

```
> ma<-matrix(a, nrow=2); ma
  [,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6
> mb <- matrix(a, ncol=3); mb
  [,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6
> mc <- matrix(a, ncol=3, byrow=T); mc
  [,1] [,2] [,3]
[1,]  1   2   3
[2,]  4   5   6
> dim(a) <- c(2,3); a
  [,1] [,2] [,3]
[1,]  1   3   5
[2,]  2   4   6
```

2.4.3 행렬 객체의 속성

- 행렬 객체의 속성은 자료의 유형, 길이, 차원(dimension), 행과 열의 이름이 있다.
- 자료의 유형은 데이터 원소들의 자료 유형이고 길이는 데이터 원소의 개수, 차원의 행과 열의 크기를 나타낸다.

dimnames() : matrix의 행과 열의 이름을 나타내주는 함수이다. list 객체를 이용한다.
matrix를 생성할시 matrix() 함수의 인수로 지정해 줄 수 있지만 dimnames() 함수는 matrix가 생성되고 난 후 사용되는 함수이다.

colnames() : matrix의 열의 이름을 나타내는 함수이다.

rownames() : matrix의 행의 이름을 나타내는 함수이다.

2.5 데이터 프레임(data frame)

- 데이터 프레임은 변수(필드)와 관찰치(레코드)로 구성된 2차원의 자료 객체이다. 가장 일반적인 데이터 구조이며, 각 변수의 자료 유형은 일치하지 않아도 상관없지만 변수별 관찰치의 수는 동일해야 한다.

2.5.1 자료의 입력: data.frame() 함수

- data.frame() : R 프로그램 내부에서 주어진 자료 객체의 결합으로 데이터 프레임을 생성하는 함수이다.

- * 내장 데이터 : R은 유명한 데이터나 자주 쓰는 문자를 내장하고 있다.

- I() : 벡터들의 묶음으로 데이터 프레임을 생성할시 문자형 벡터는 범주형(factor)으로 변환된다. 이를 방지하기 위해서 I() 함수를 이용한다.

```
> a<-letters[1:5]; a
[1] "a" "b" "c" "d" "e"
> b<-month.abb[1:5]; b
[1] "Jan" "Feb" "Mar" "Apr" "May"
> x<-data.frame(alpha=a, month=b,
+ row.names=c(5,4,3,2,1)); x
  alpha month
5     a   Jan
4     b   Feb
3     c   Mar
2     d   Apr
1     e   May
> x<-data.frame(I(a), I(b)); x
  a b
1 a Jan
2 b Feb
3 c Mar
4 d Apr
5 e May
> colnames(x) <- c("alpha","month")
> rownames(x) <- c(5,4,3,2,1)
> x
  alpha month
5     a   Jan
4     b   Feb
3     c   Mar
2     d   Apr
1     e   May
```

2.6 유용한 함수

함수	기능	함수	기능
sum()	원소의 합	prod()	원소의 곱
max()	최대값	min()	최소값
which.max()	최대값의 인덱스	which.min()	최소값의 인덱스
range()	c(min(),max())	var()	분산
sd()	표준편차	cor()	상관계수
length()	원소의 수	mean()	평균
median()	중위수	rev()	원소의 역순
scale()	표준화	cumsum()	누적 합
cumprod()	누적 곱	cummin()	누적 최소값
cummax()	누적 최대값	subset()	원소의 일부 선택
sample()	복원/비복원의 임의추출	match()	일치하는 원소의 인덱스
which()	조건을 만족하는 원소의 인덱스	gl()	주어진 수준에 따른 요인 생성
diff()	데이터 원소 사이의 차	rank()	원소들의 순위
sort()	원소들의 정렬	order()	rank에 해당하는 인덱스
toupper()	대문자로 변환	tolower()	소문자로 변환
nchar()	문자의 길이	substr()	문자의 일부분을 선택 혹은 변경
sub()	문자 중 정의된 내용과 같은 경우 치환(첫번째만)	gsub()	문자 중 정의된 내용과 같은 경우 치환(모두)

2.7 R 프로그래밍 및 사용자 정의 함수

- R에서 제공되는 연산이나 함수 이외에 조건문, 반복문 등을 이용한 프로그래밍을 통해 다양한 형태의 출력결과를 얻을 수 있는 프로그램을 작성할 수 있다.
- 사용자가 자주 사용하는 프로그램의 경우는 함수와 관련된 인수를 정의하여 선언함으로써 간단히 적용할 수 있으며, 이러한 함수를 사용자 정의 함수라고 한다.

2.7.1 조건문

- R의 조건문과 조건문 기능을 하는 함수로는 다음과 같은 종류가 있다.
 - if (조건) 명령문
 - if (조건) 명령문1 else 명령문2
 - ifelse(조건, 명령문1, 명령문2)
 - switch(기준, 조건1, 명령문1, 조건2, 명령문2, ...)

2.7.2 반복문

- 일정한 조건하에서 정의된 명령문의 반복 수행을 지시하는 R의 반복문의 종류는 다음과 같다.
 - for (반복) 명령문
 - while (조건) 명령문
 - repeat 명령문
- for 문은 정의된 반복 형태에 따라 명령문을 수행하는 것이고 while 문은 조건이 참인 경우에만 명령문을 수행한다. repeat 문은 조건 없이 명령문을 반복 수행하는 것으로 무한 반복 수행을 막기 위해서 무조건 분기문을 반드시 포함해야 한다.
- for 문과 while 문은 조건을 먼저 확인한 후 명령문 수행 여부를 결정하지만 repeat 문은 주어진 명령문을 먼저 수행한다. 따라서 명령문의 순서에 따라 동일한 내용의 프로그램이라도 결과가 다를 수 있다.

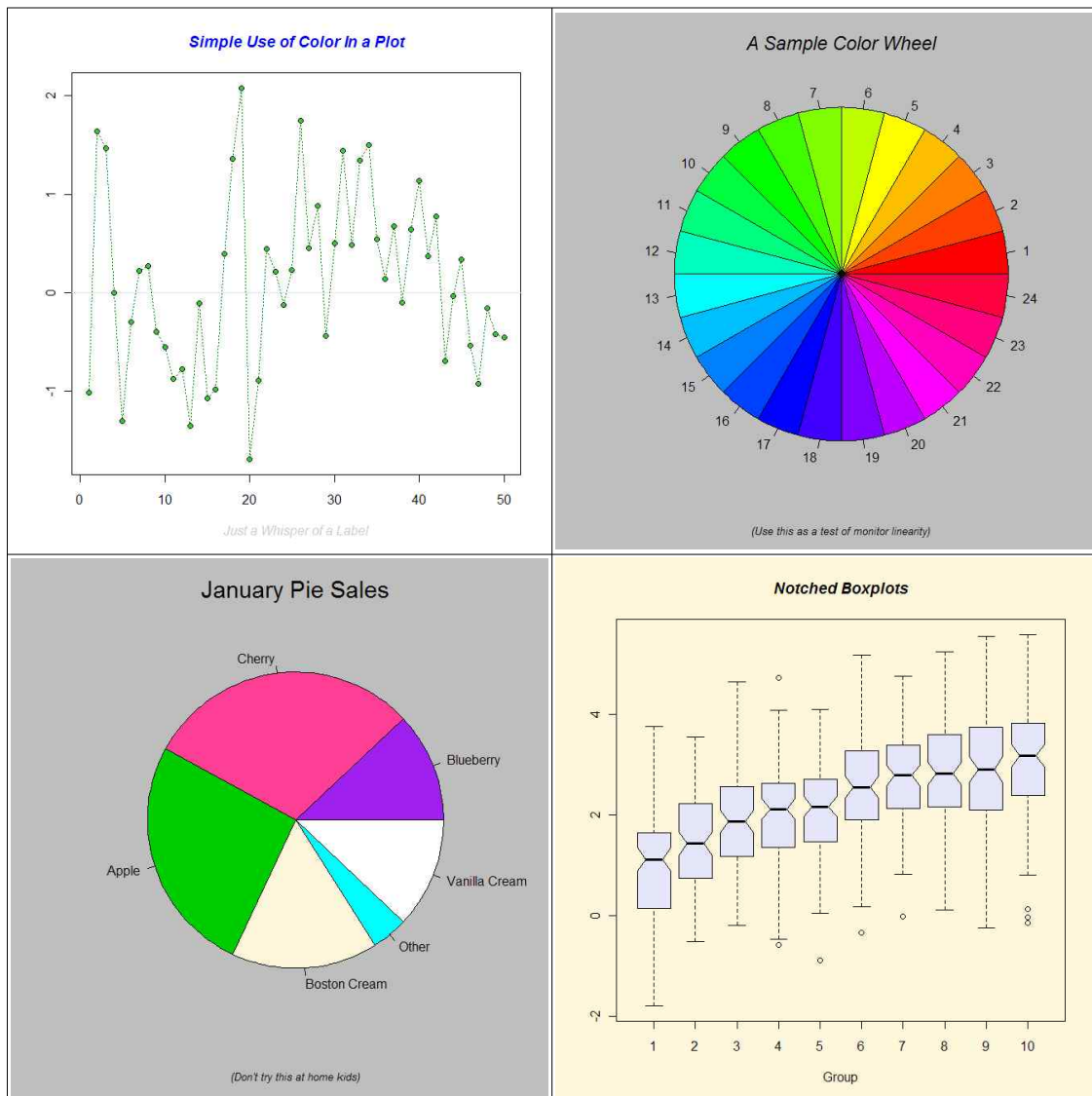
* 무조건 분기문

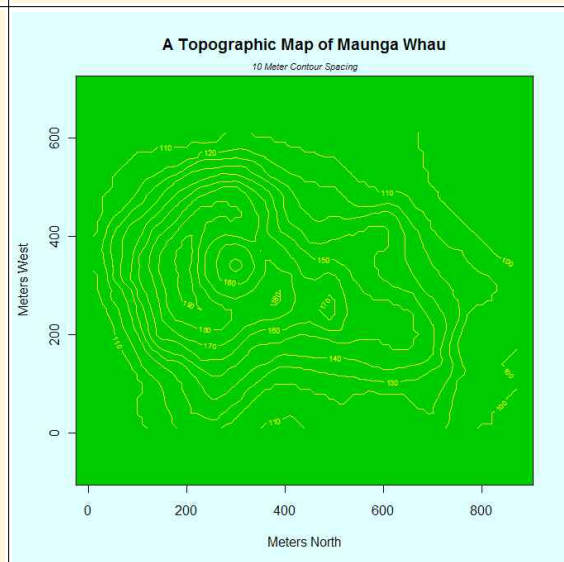
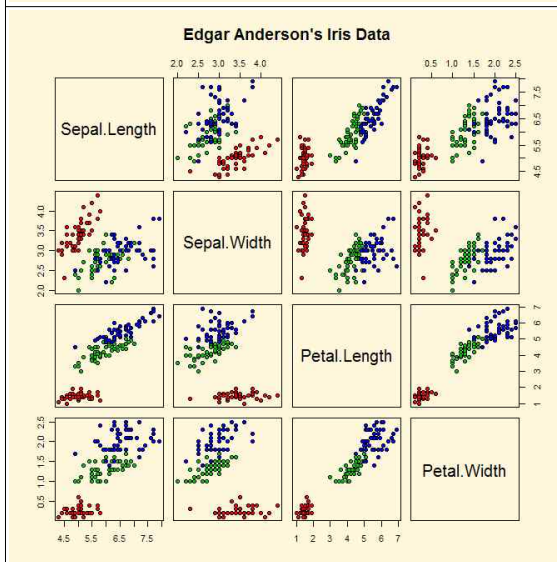
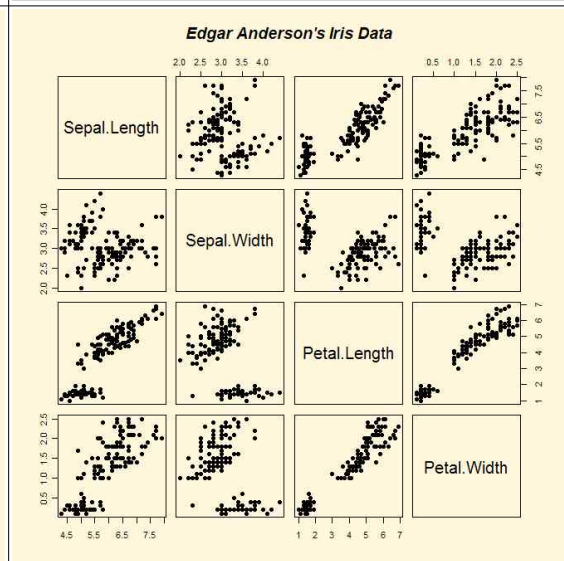
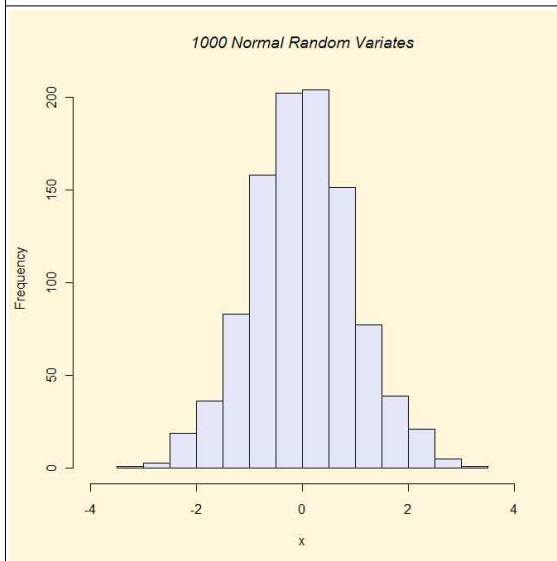
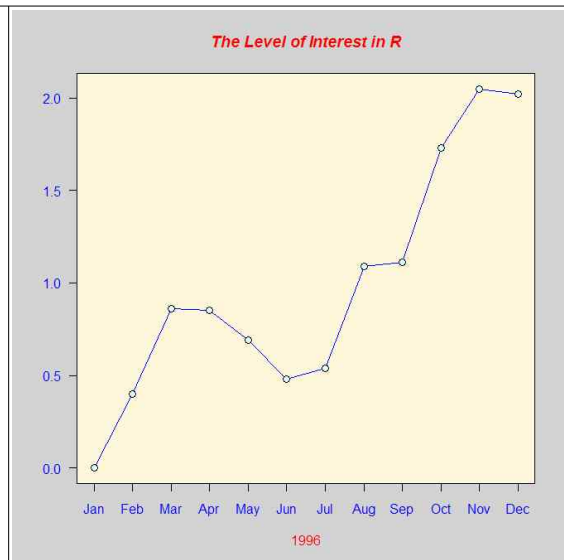
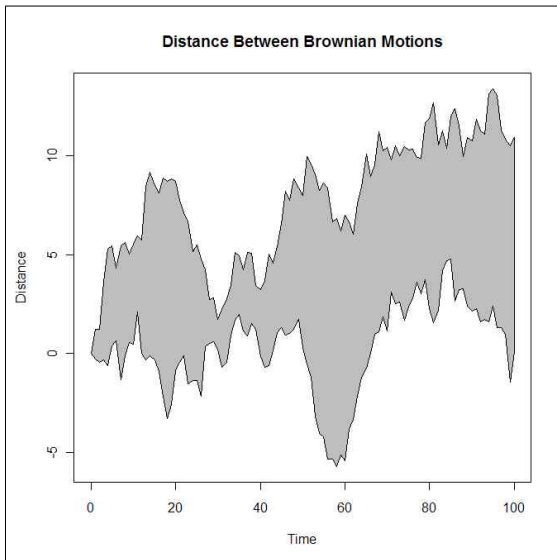
- 무조건 분기문은 주로 반복문의 실행을 제어하는 목적으로 사용되는 명령문으로 반복실행을 무조건 빠져나가도록 하는 break 문과 다음 반복실행 단계로 이동하게 하는 next 문이 있다.

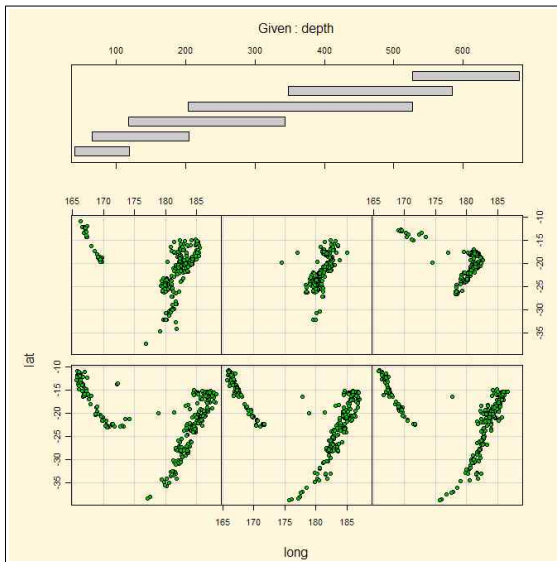
2.8 그래픽(Graphic)

2.8.1 그래픽의 관리

- R은 매우 다양하며 강력한 그래픽 기능을 제공한다.
- `demo(graphics)` : R이 제공하는 다양한 2D 그래픽 기능의 데모이다.
- `demo(persp)` : R이 제공하는 다양한 3D 그래픽 기능의 데모이다.

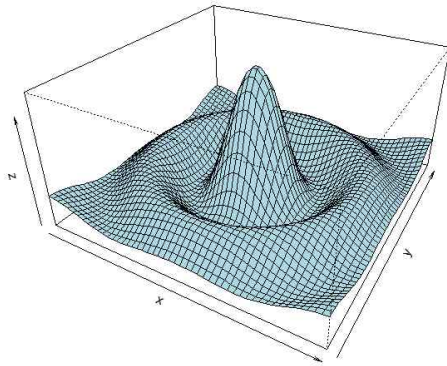




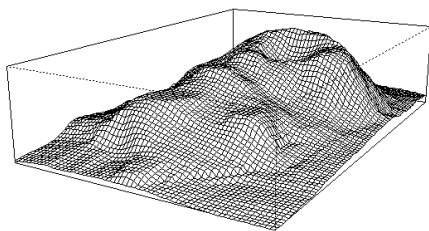
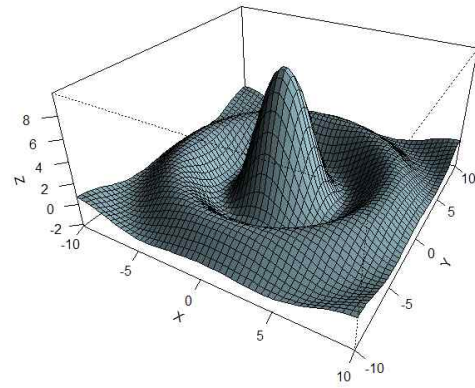


demo(persp)

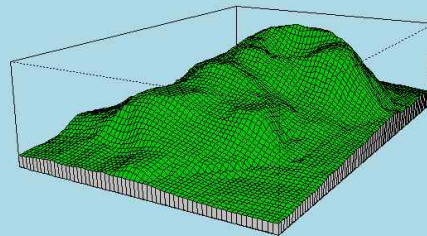
$$z = \text{Sinc}(\sqrt{x^2 + y^2})$$

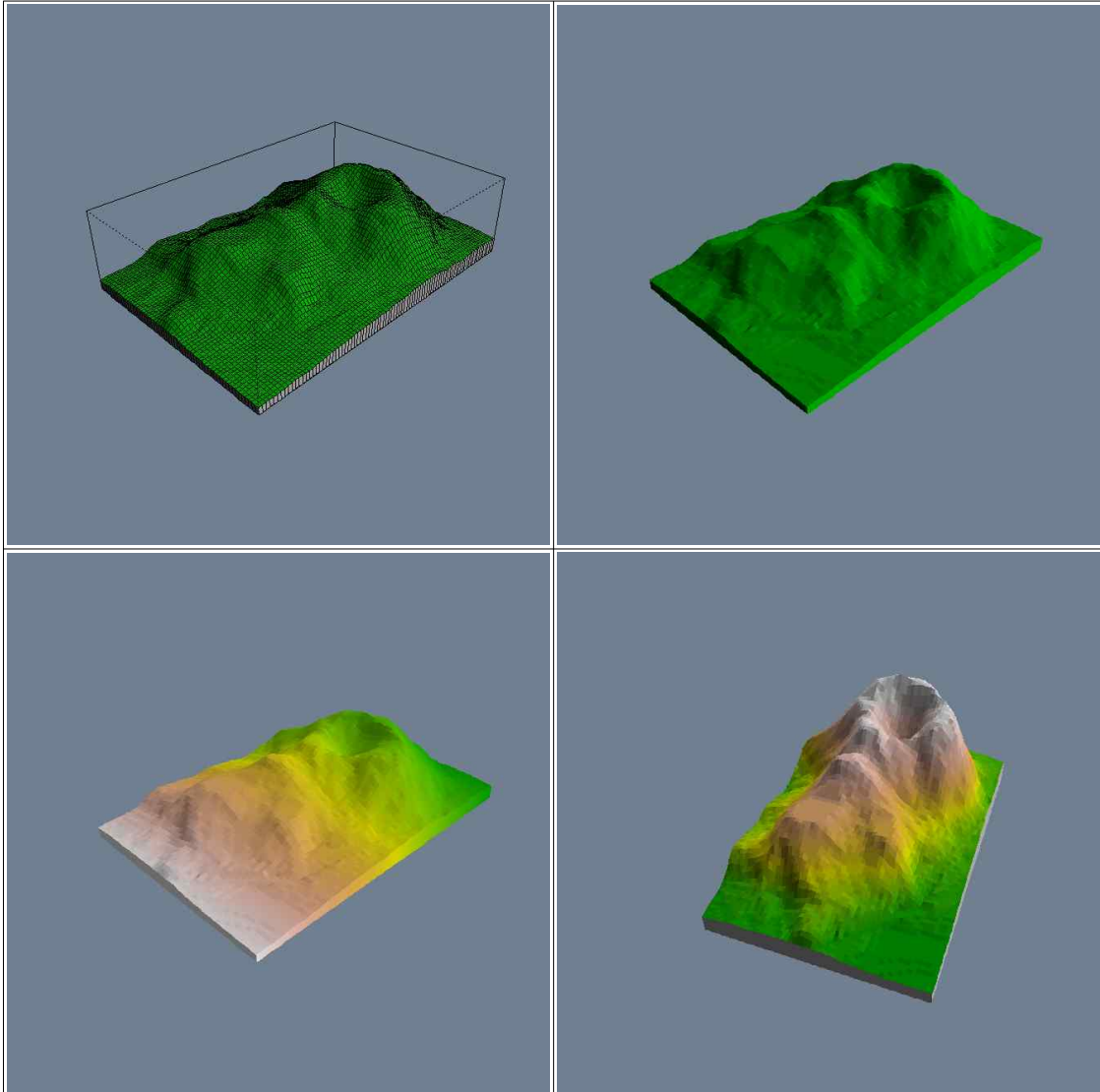


$$z = \text{Sinc}(\sqrt{x^2 + y^2})$$



Maunga Whau
One of 50 Volcanoes in the Auckland Region.





- 그래픽 작성과 관련된 패키지는 `grid`, `lattice` 등과 기본 패키지 내에 `grDevices`, `graphics` 패키지가 있다.

2.8.2 여러 그래픽 장치 열기

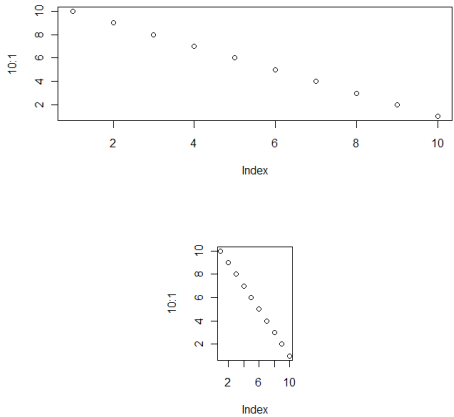
- 그래픽 함수의 결과는 객체로 생성되는 것이 아니라 그래픽 장치(device)로 전해져 표현된다. 이러한 그래픽 장치로는 그래픽을 포함한 윈도우나 그래픽 파일이 있다.
- `bmp(filename = "Rplot%03d.bmp", width = 480, height = 480, units = "px",`
- `pointsize = 12, bg = "white", res = NA, restoreConsole = TRUE)`
- `jpeg(filename = "Rplot%03d.jpg", width = 480, height = 480, units = "px",`
- `pointsize = 12, quality = 75, bg = "white", res = NA, restoreConsole = TRUE)`

- `png(filename = "Rplot%03d.png", width = 480, height = 480, units = "px", pointsize = 12, bg = "white", res = NA, restoreConsole = TRUE)`
 - `savePlot(filename = "Rplot", type = c("wmf", "emf", "png", "jpeg", "jpg", "bmp", "ps", "eps", "pdf"), device = dev.cur(), restoreConsole = TRUE)`
- 그래픽 함수가 수행된 후 열려진 그래픽 장치가 없을 경우, 그래픽 윈도우를 열고 그 위에 그래프를 표현하는데 이때 열려지는 그래픽 윈도우는 수행된 함수에 의존한다.
- `x11()` or `windows()` : 그래픽 윈도우를 여는 함수이다.
 - `postscript()` or `pdf()` or `png()` : 그래픽 파일을 여는 함수이다.
- 그래픽 함수의 결과는 활성화되어 있는 가장 최근에 열린 그래픽 장치에 표현된다.
- `dev.list()` : 열려진 그래픽 장치의 리스트를 출력한다.
 - `dev.cur()` : 현재 활성화된 그래픽 장치를 참조하는 함수이다.
 - `dev.set()` : 인수로 주어지는 그래픽 장치 번호를 기본으로 해당 장치를 활성화한다.
 - `dev.off()` : 인수로 주어지는 번호를 기준으로 열려진 그래픽 장치를 닫는 함수로, 디폴트는 현재 활성화된 장치이다.

2.8.3 그래픽의 분할

- `split.screen()` : 현재 활성화된 그래픽 장치를 분할하는 함수이다. 그래픽 장치의 분할된 구역을 스크린(screen)이라 부른다.
- `layout()` : 현재 활성화된 그래픽 장치의 화면을 분할해 그래프의 출력 결과를 차례로 표현한다. 행렬을 통해 화면을 분할한다.

<pre>nf <- layout(matrix(c(1,1,0,2), 2, 2, byrow=TRUE), respect=TRUE) layout.show(nf)</pre>	<div style="border: 1px solid black; width: 100%; height: 100%; display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; width: 50%; height: 50%; display: flex; align-items: center; justify-content: center;">1</div> <div style="border: 1px solid black; width: 50%; height: 50%; display: flex; align-items: center; justify-content: center;">2</div> </div>
------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

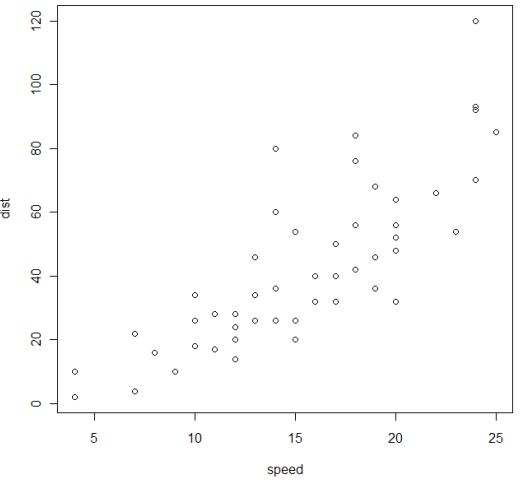
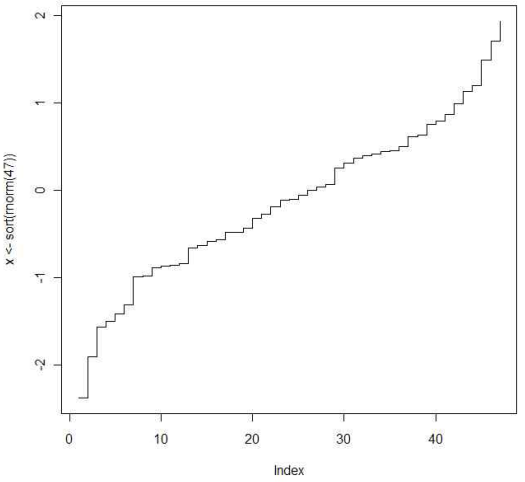
<pre>split.screen(c(2,1)) split.screen(c(1,3), screen = 2) screen(1) plot(10:1) screen(4) plot(10:1) close.screen(all = TRUE)</pre>	
-------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

2.9 R의 고수준 그래픽(Graphic) 함수

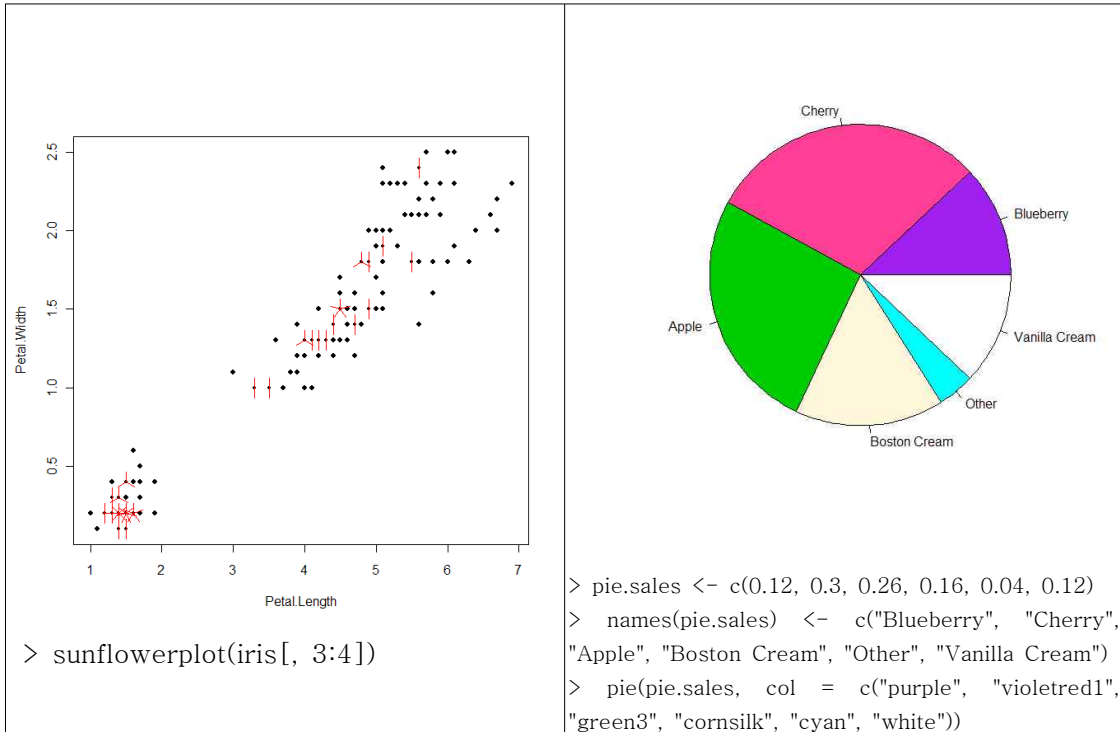
- R의 그래픽 함수는 새로운 그래프를 작성하는 함수와 기존에 존재하는 그래프에 점, 선, 면, 문자, 좌표축, 범례 등의 다양한 그래프의 속성을 정의하는 함수로 크게 구분된다. 이 중 전자를 고수준(high-level) 그래픽 함수라고 하고 후자를 저수준(low-level) 그래픽 함수라 한다.

2.9.1 고수준 그래픽 함수의 종류 및 기능

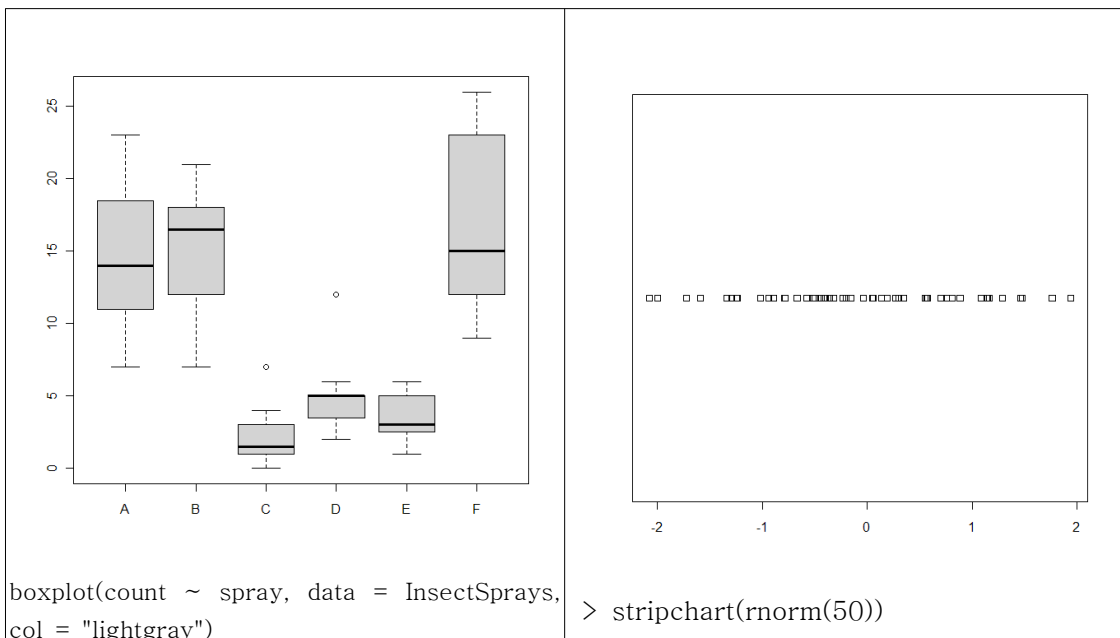
- plot(x) : 가로축에는 인덱스(index)를 표시하고 세로축에 x의 값을 표시한다.
- plot(x,y) : x와 y의 이차원 그래프를 표현한다.

 <pre>plot(cars)</pre>	 <pre>plot(x <- sort(rnorm(47)), type = "s", main = "plot(x, type = 'sW')")</pre>
-----------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

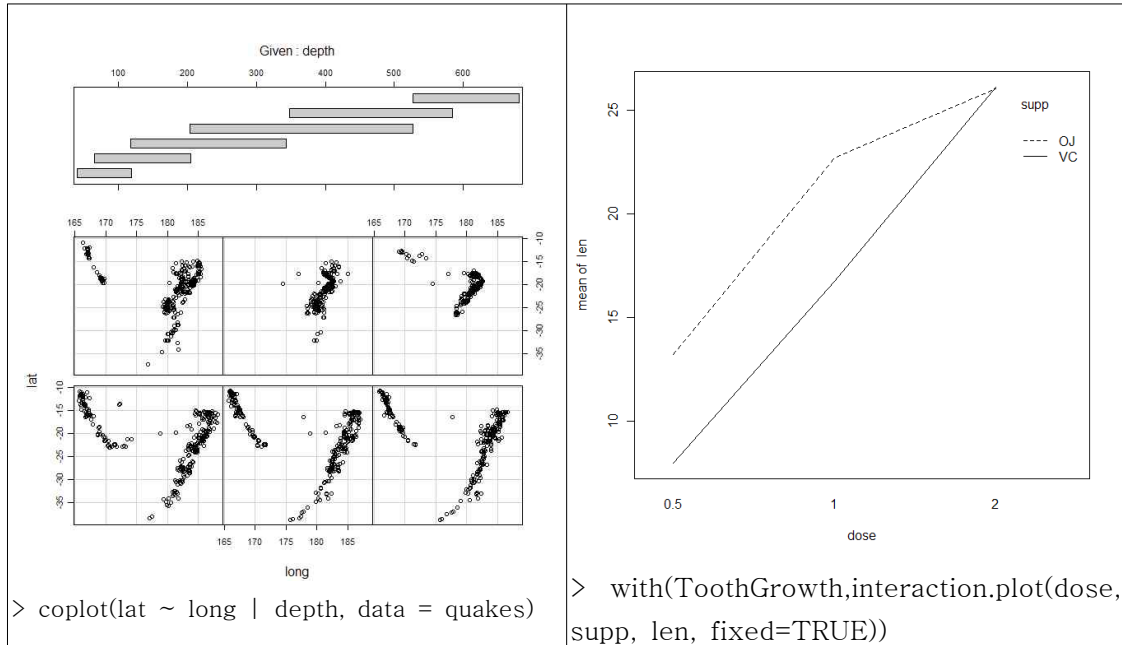
- sunflowerplot(x,y) : x와 y의 이차원 그래프를 표현하며 중첩된 케이스가 있는 경우 자료의 크기를 꽃잎의 형태로 표현한다.
- pie(x) : x에 대한 파이 차트를 작성한다.



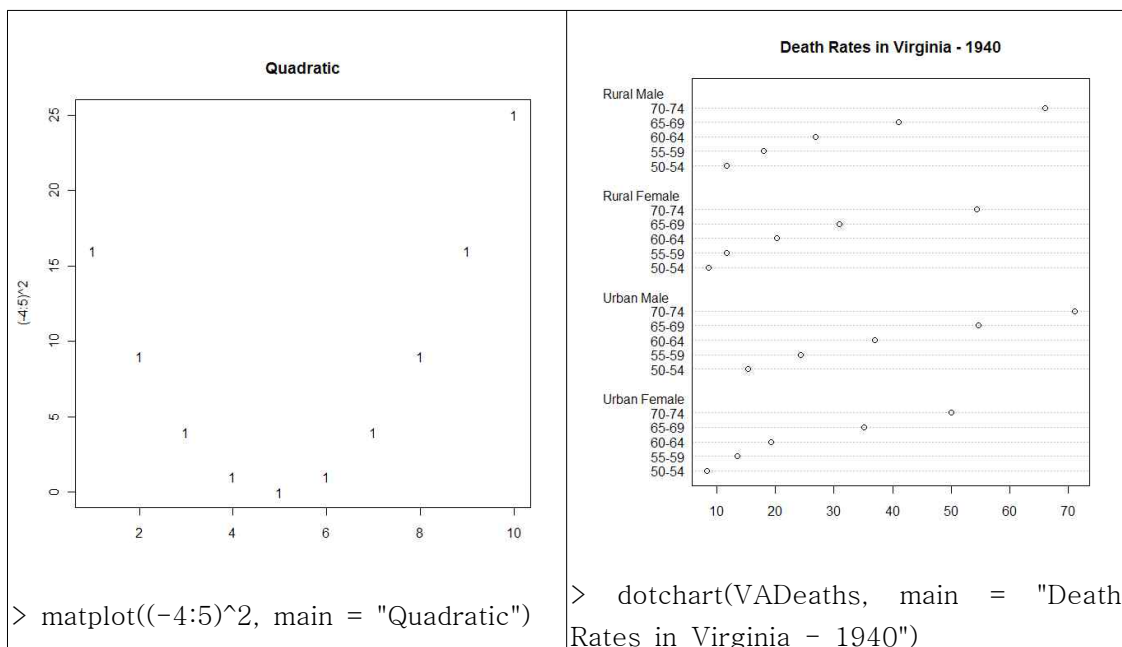
- boxplot(x) : 상자와 수염(box-and-whiskers) 그래프를 작성한다.
- stripchart(x) : x에 대한 일차원 산점도를 표현한다.



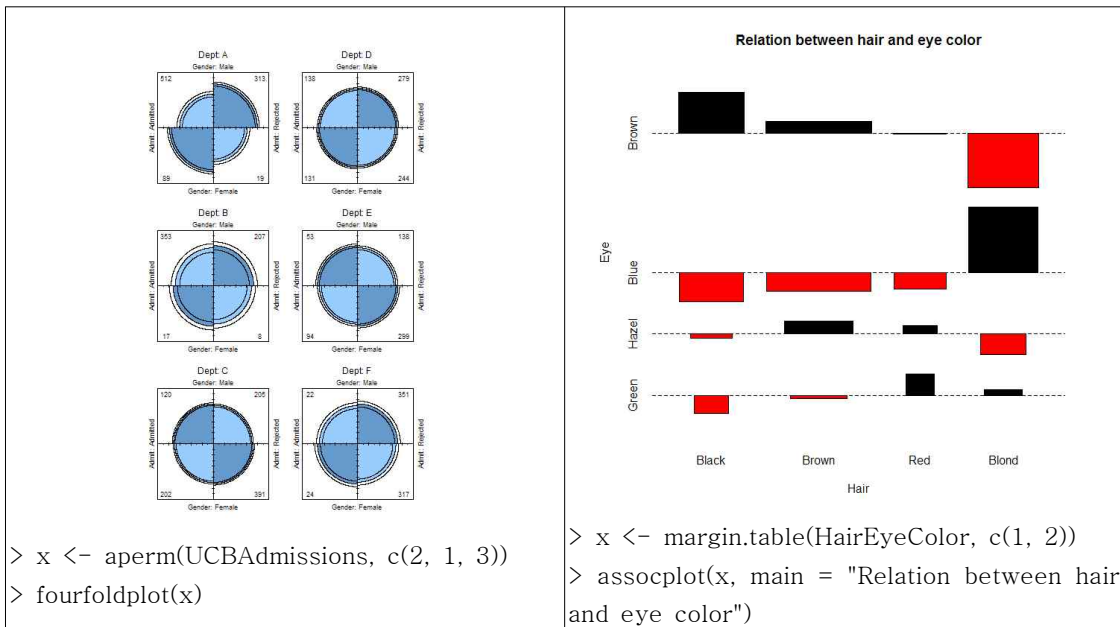
- `coplot(x~y | z)` : z 의 값에 따라 x 와 y 의 2차원 그래프를 표현한다.
- `interaction.plot(f1,f2,y)` : $f1$ 과 $f2$ 가 요인이라면 y 값의 평균을 y 축 위에 표시하여 요인이 $f1,f2$ 인 2원 분산분석의 교호작용 그림을 표현한다.



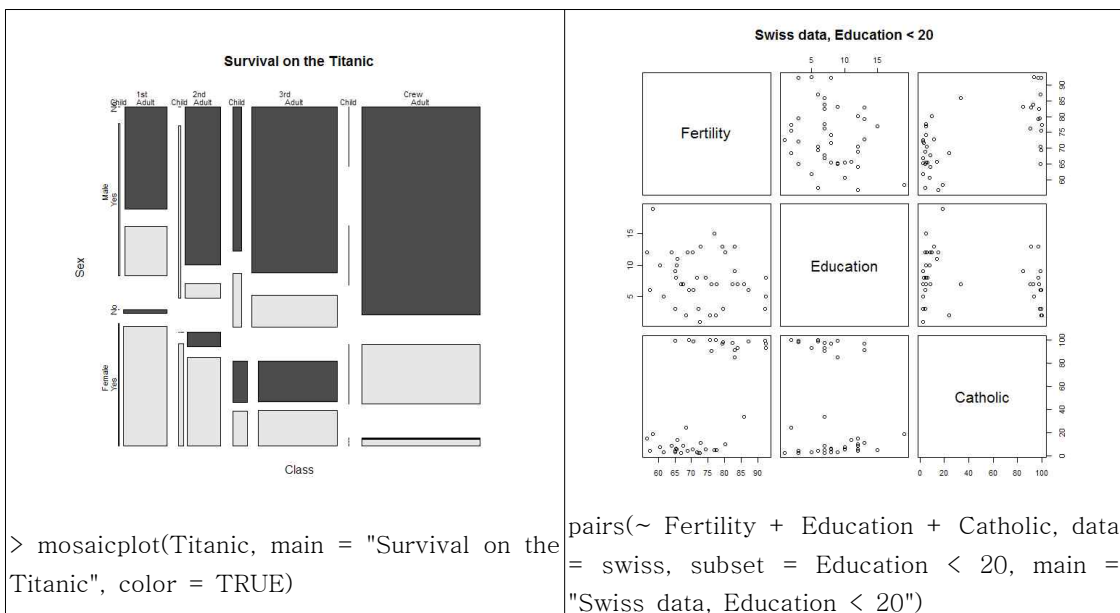
- `matplot(x,y)` : 행렬 데이터 x 의 열별로 y 의 지정된 형식에 따라 2차원 그래프를 표현한다.
- `dotchart(x)` : 데이터 프레임 x 의 행(열) 수준별 산점도를 분석하고자 하는 변수들을 세로로 결합해 표현한다.



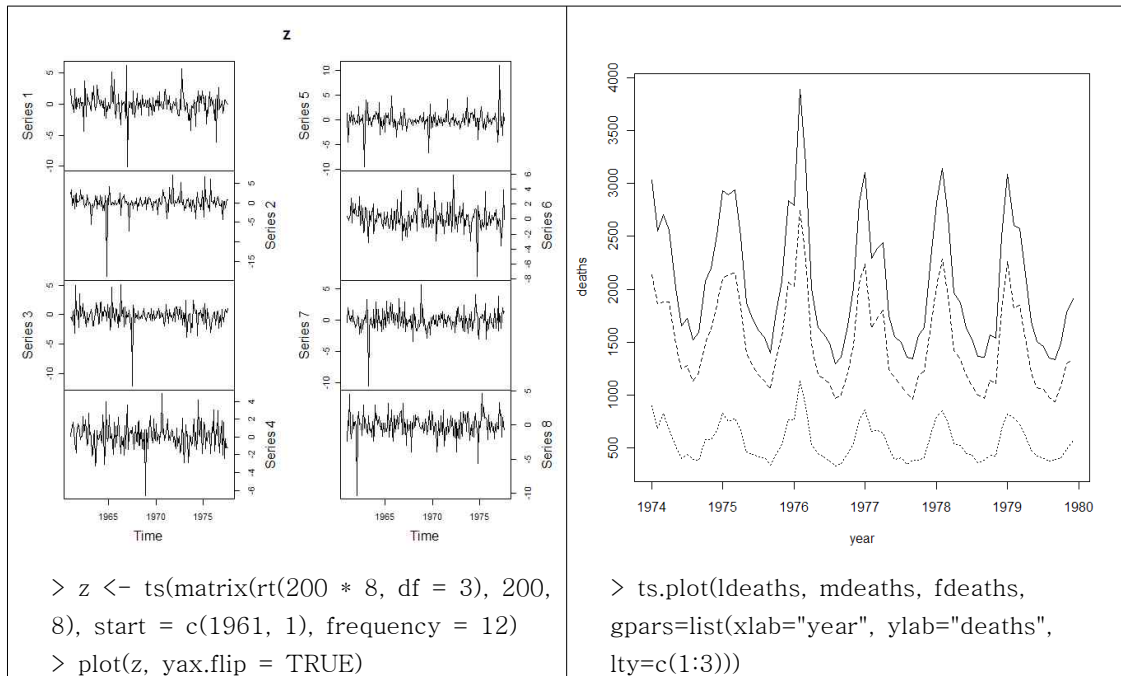
- fourfoldplot(x) : k개의 2x2 2차원 분할표에 대한 결합 형태인 차원(2,2,k)인 x에 대해 두 이항 변수들 간의 연관성을 시각적으로 표현한 1/4원 그래프를 k개 생성한다. 1/4원 그래프는 원을 4개의 균등한 영역으로 나누고 각 영역에 도수, 확률의 추정치와 그 신뢰구간을 표현하는 그래프이다.
- assocplot(x) : 2차원 분할표에 대해 행과 열의 독립성을 나타내는 코헨-프렌들리 (Cohen-Friendly) 그래프를 표현한다.



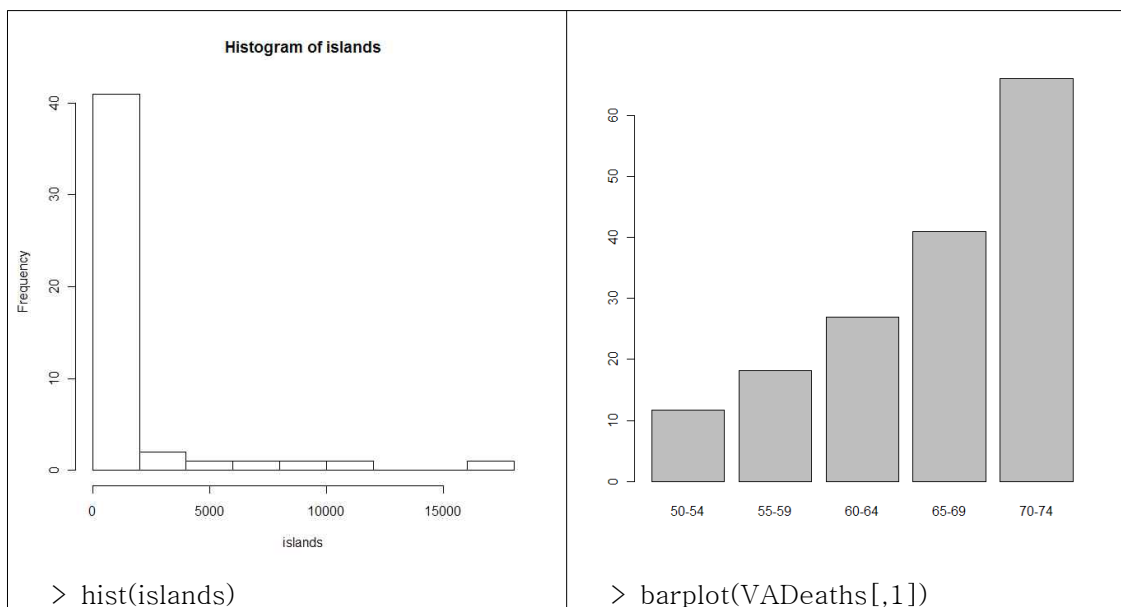
- mosaicplot(x) : 다차원 분할표에 대해 범주별 비율이나 주변 합계를 구분된 사각형의 면적으로 표현한다. 로그-선형 모형으로부터의 잔차의 크기나 부호는 색으로 구분하여 나타낸다.
- pairs(x) : x가 행렬이나 데이터 프레임일 경우, x의 행에 대한 산점도 행렬을 출력한다.



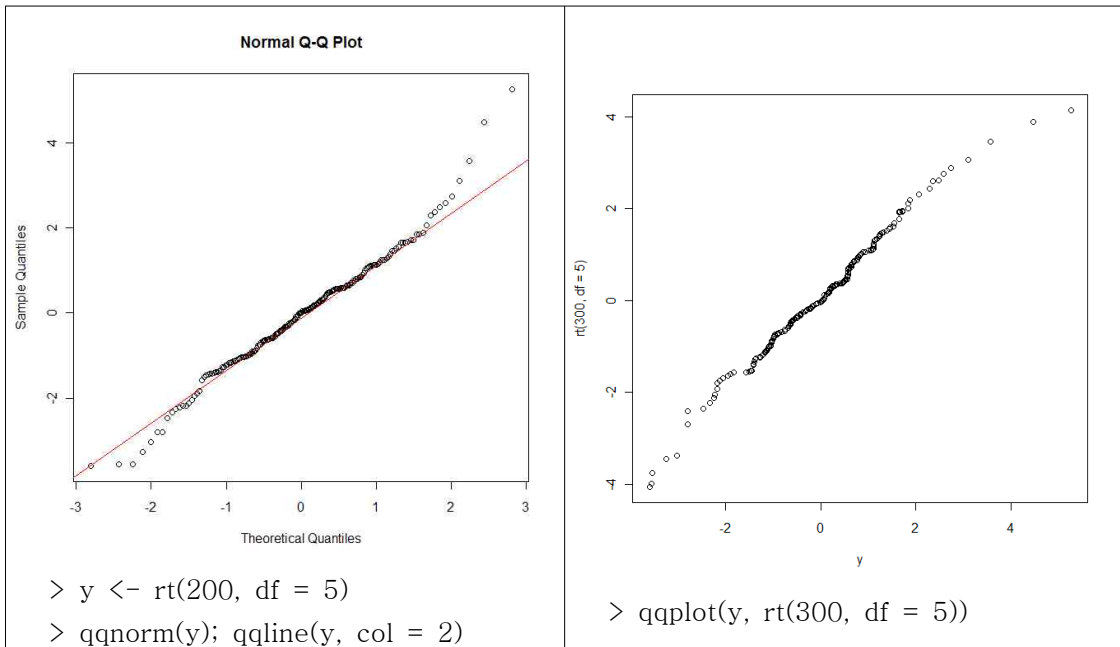
- plot.ts(x) : 시계열 자료에 대한 시계열 그래프를 표현한다.
- ts.plot(x) : 다중 시계열 자료에 대한 시계열 그래프를 표현한다.



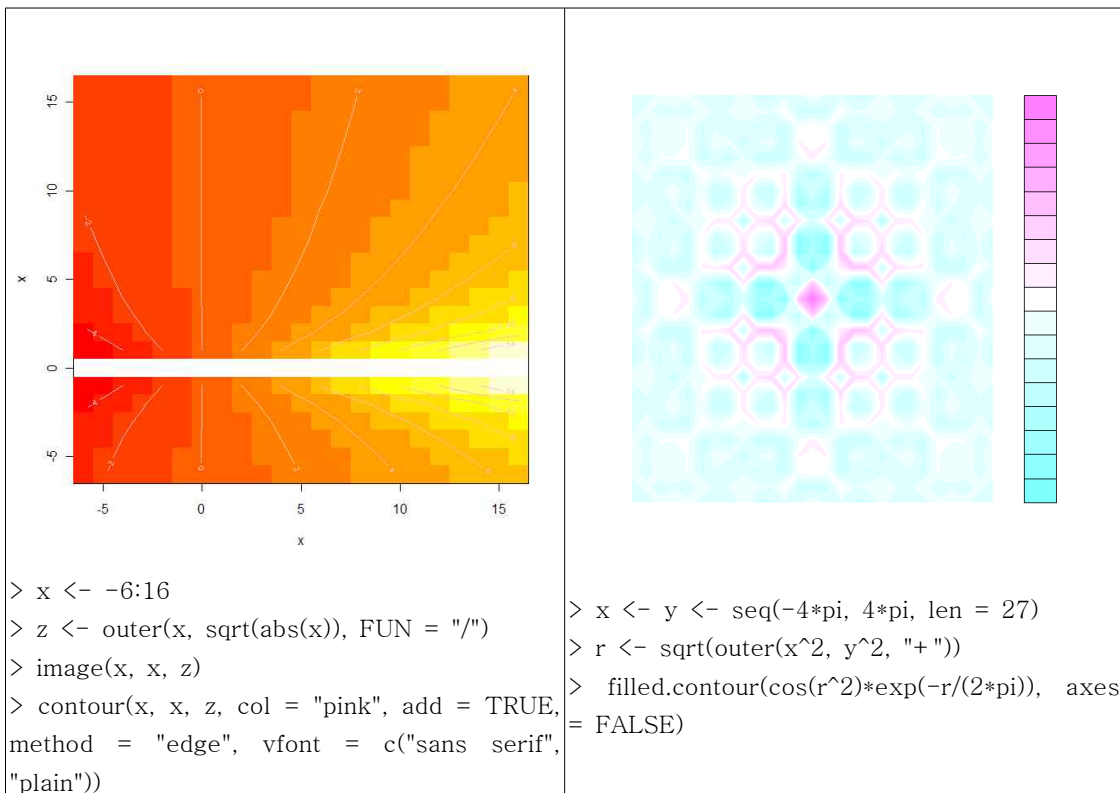
- hist(x) : x의 도수에 대한 히스토그램을 표현한다.
- barplot(x) : x의 값에 대한 막대 그림을 표현한다.



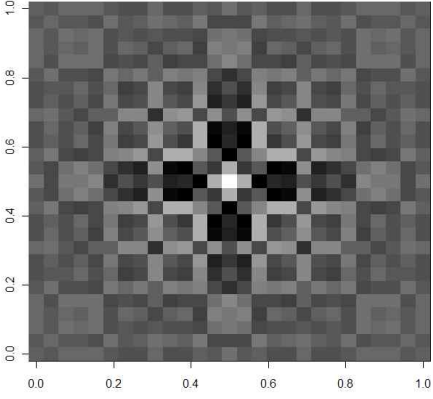
- qqnorm(x) : 누적정규확률 그래프를 출력한다.
- qqline(x) : normal quantile-quantile line을 추가한다.
- qqplot(x,y) : x의 분위수에 대한 y의 분위수를 그래프로 표현한다.



- `contour(x,y,z)` : 벡터인 x , y 와 행렬인 z 에 대해 2차원 등고선 그래프를 출력한다.
- `filled.contour(x,y,z)` : `contour(x,y,z)`의 그래프에 등고선 사이를 색칠하고 그 색에 대한 구분을 범례로 출력한다.



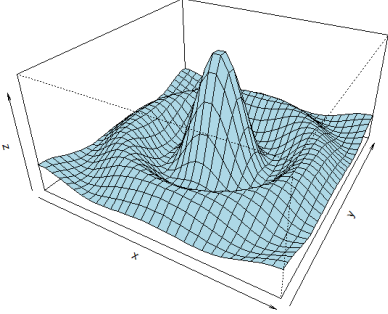
- image(x,y,z) : 3차원 자료나 공간 자료를 유색의 이미지로 표현한다.
- persp(x,y,z) : 3차원의 그래프를 2차원의 xy 평면에 투영하여 투시도를 표현한다.



```

> x <- y <- seq(-4*pi, 4*pi, len=27)
> r <- sqrt(outer(x^2, y^2, "+ "))
> image(z = z <- cos(r^2)*exp(-r/6),
col=gray((0:32)/32))

```



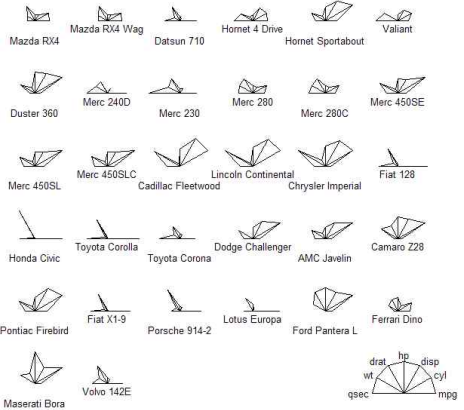
```

> x <- seq(-10, 10, length= 30)
> y <- x
> f <- function(x,y) { r <- sqrt(x^2+y^2); 10
* sin(r)/r }
> z <- outer(x, y, f)
> persp(x, y, z, theta = 30, phi = 30, expand
= 0.5, col = "lightblue")

```

- stars(x) : x에 대한 별 그림을 표현한다.
- lag.plot(x) : 시계열 자료 x에 대한 lag 그래프를 표현한다.

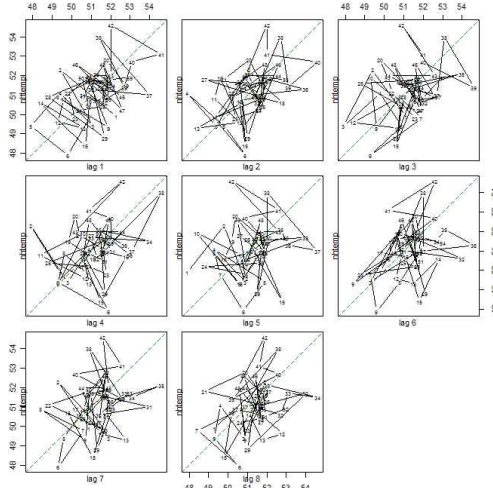
Motor Trend Cars : stars(*, full = F)



```

> stars(mtcars[, 1:7], key.loc = c(14, 2),
main = "Motor Trend Cars : stars(*, full =
F)", full = FALSE)
> lag.plot(nhtemp, 8, diag.col = "forest
green")

```



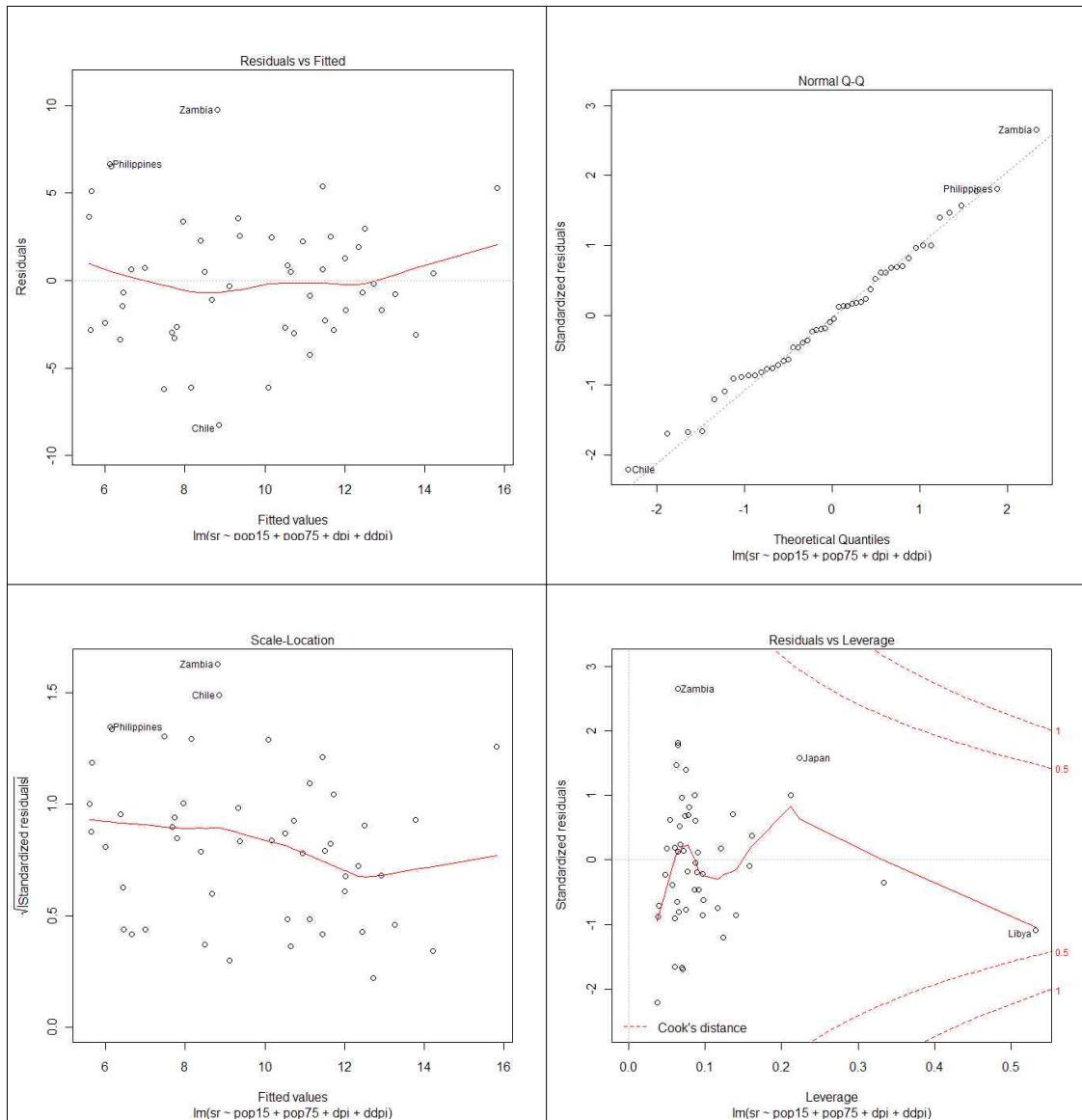
```

> lag.plot(nhtemp, 8, diag.col = "forest
green")

```

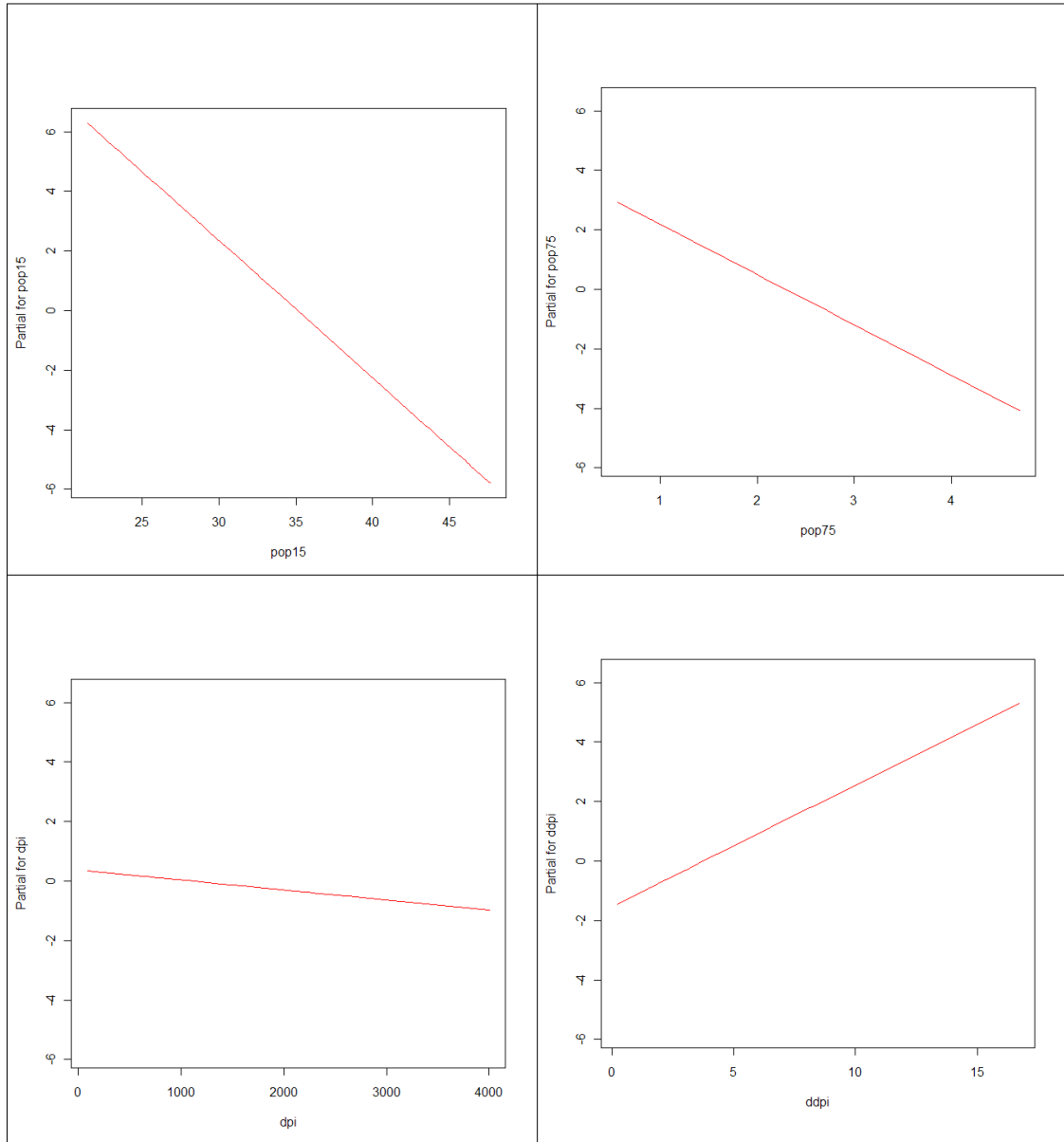
- plot.lm() : 회귀분석의 결과로부터 회귀진단을 위한 그래프를 출력한다.

- plot(lm.SR <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings))

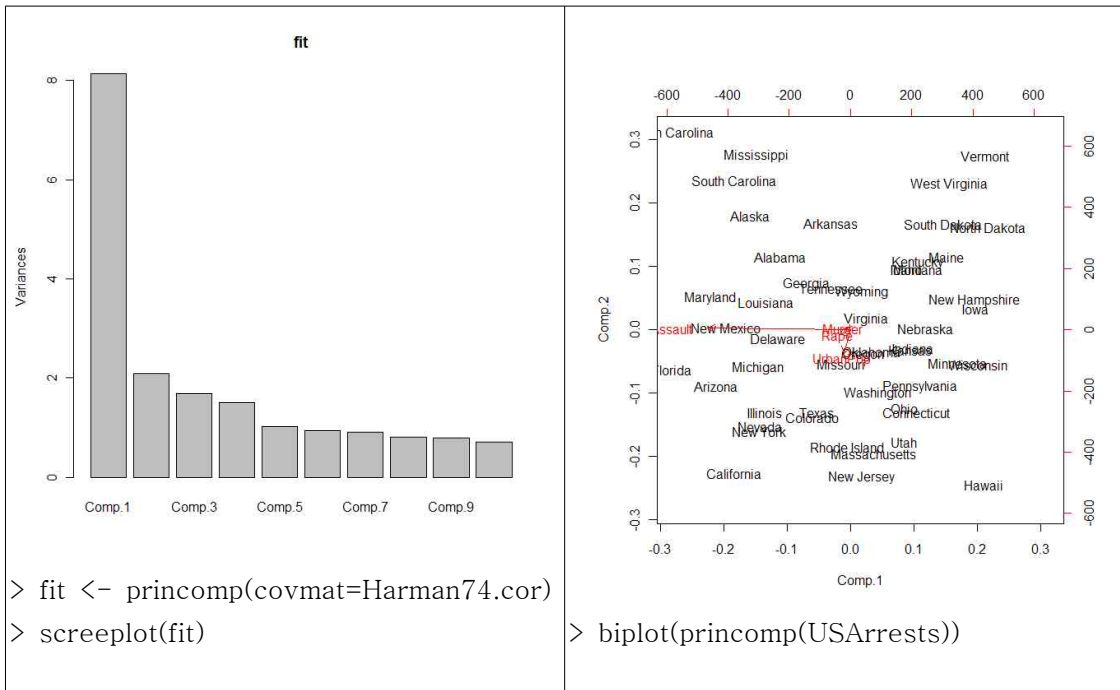


- termplot() : 회귀분석의 결과로 적합된 회귀모형을 이루는 각 항에 대해서 예측값들에 대응하는 회귀선을 출력한다.

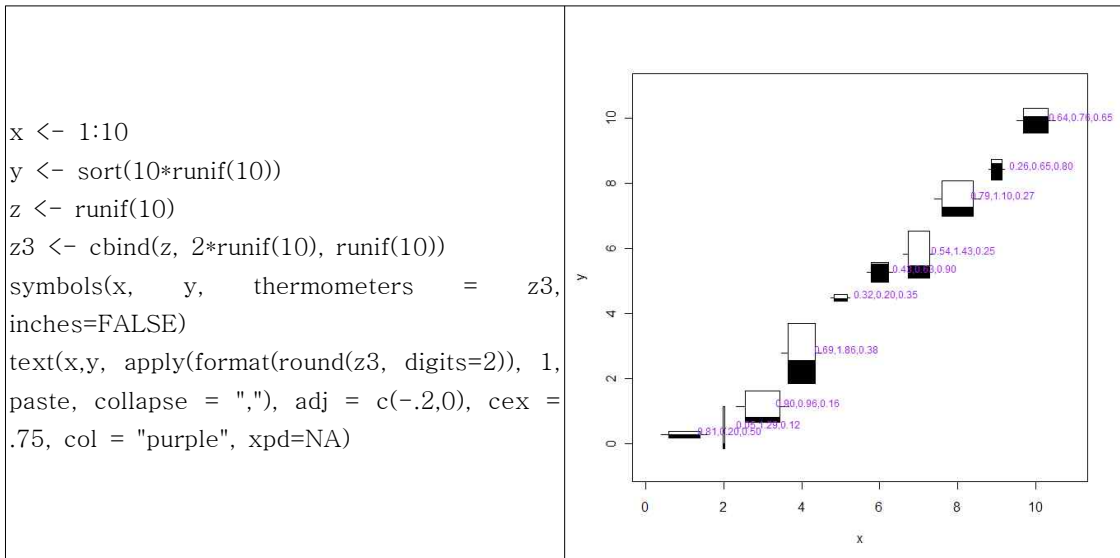
```
-termplot(lm.SR <- lm(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings))
```



- `screeplot()` : 다변량 분석의 일종인 주성분 분석의 결과로부터 산비탈(scree plot) 그림을 출력 한다.



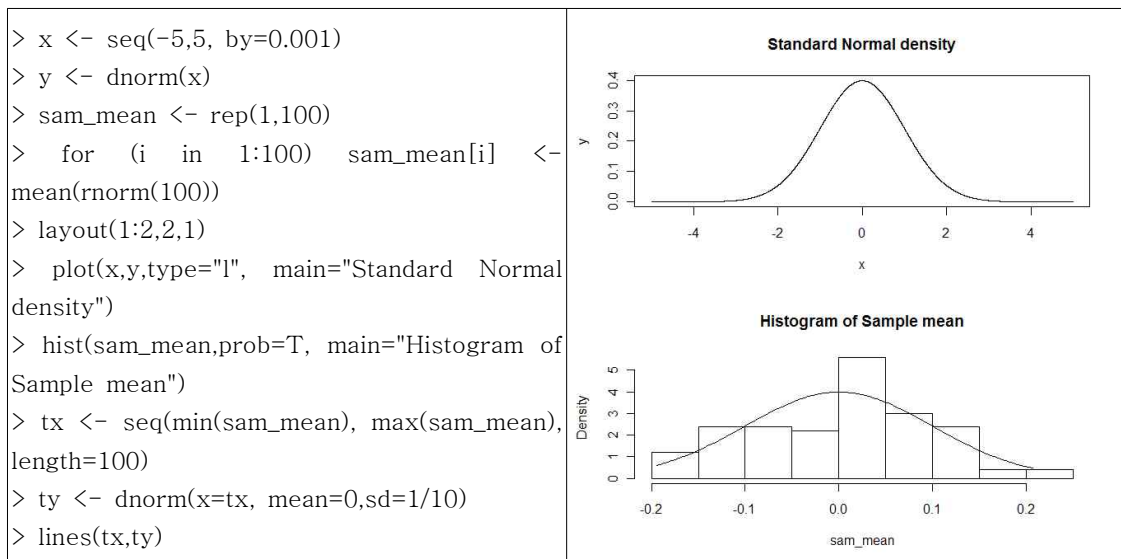
- `symbols(x,y,...)` : (x,y) 좌표에 기호(원, 사각형, 별, 온도계, 상자-수염)가 정의된 인수의 값에 따라 크기, 색 등이 다르게 표현된 심볼 그래프를 표현한다.



2.9.2 공통 선택사항(option)

- 다양한 고수준 그래픽 함수는 고유의 기능을 수행하고 사용자가 원하는 최적의 출력결과를 얻기 위해 정의 가능한 다양한 선택사항들이 있다.

- add=FALSE : TRUE 이면 만약 이전의 출력 그래프에 겹쳐서 출력한다.
- axes=TRUE : FALSE이면 그래프의 축이나 테두리를 출력하지 않는다.
- type=p : 그래프의 출력형식을 정의하는 것으로 p(점), l(줄), b(선위의 점), o(점위의 선), h(수직 선), s(계단 그림으로 수직선은 데이터의 위치 아래로 표현), S(계단 그림으로 수직선은 데이터의 위치 위로 표현)가 가능하다.
- xlim=, ylim= : 문자로 x(y) 축의 상한과 하한을 정의한다.
- xlab=, ylab= : 문자로 x(y) 축의 주석(annotates)을 정의한다.
- main= : 문자로 그래프의 주제목(main title)을 정의한다.
- sub= : 문자로 그래프의 부제목(sub title)을 정의한다.



2.10 저수준 그래픽 함수

2.10.1 주요 저수준 그래픽 함수

- 저수준 그래픽 함수는 기존의 그래프의 속성을 변경하는 함수로 주요 저수준 그래픽 함수는 다음과 같다.

`point(x,y)` : (x,y) 좌표에 포인트를 추가한다.

`lines(x,y)` : (x,y) 좌표를 연결하는 라인을 추가한다.

`curve(f(x),from,to)` : x의 함수 f(x)를 그려준다.

`text(x,y,labels)` : 이름을 추가한다.

`mtext(text,side,line)` : 그래프의 side와 line 인수로 정의된 위치에 text를 출력한다.

`segments(x0,y0,x1,y1)` : (x0,y0) 좌표와 (x1,y1) 좌표 사이를 연결하는 선을 출력한다.

`arrows(x0,y0,x1,y1,angle,code)` : (x0,y0) 좌표와 (x1,y1) 좌표 사이를 연결하는 화살표를 출력한다. 화살표의 위치는 code 인수로 정의되며 angle 인수는 화살표의 모양을 정의한다.

`abline(a,b)` : 기울기 b, 절편 a인 직선을 출력한다.

`abline(h=y)` : y축의 값을 y로 갖는 x축과 평행한 직선을 출력한다.

`abline(v=x)` : x축의 값을 x로 갖는 y축과 평행한 직선을 출력한다.

`abline(lm.obj)` : lm 함수의 객체로부터 주어지는 회귀선을 출력한다.

`rect(x1,y1,x2,y2)` : 좌표 (x1,y1), (x1,y2), (x2,y2), (x2,y1)을 연결하는 사각형을 추가한다.

`polygon(x,y)` : x와 y로부터 주어지는 좌표들을 연결하는 다각형을 추가한다.

`legend(x,y,legend)` : 좌표 (x,y)에 위치하는 legend에 주어진 범례를 출력한다.

`title()` : 주제목과 부제목을 추가한다.

`axis(side, vect)` : 축을 side 인수에 정의된 위치에 추가하고 vect의 값에 따라 `abline(h=y)`, `abline(v=x)`의 결과와 같은 평행선을 굵게 표현한다.

`box()` : 그래프를 감싸는 상자를 추가한다.

`rug(x)` : x의 값을 x 축에 표현한다.

`locator(n, type)` : 그래프 내에서 사용자가 n번 클릭한 위치의 좌표값을 표현하고 type 인수에 따른 점, 선 등을 추가한다. 디폴트는 type="n"로 추가하지 않는 것이다.

2.11 par() 함수

- 그래프 표현의 향상은 저수준 그래픽 함수를 적용하는 것 이외에 그래프를 출력하는 그래픽 장치의 여러 가지 인수를 정의함으로써도 가능하다.

- `par()` : 그래픽 장치의 다양한 속성들을 참조하고 정의하는 기능을 가진 함수이다. `par()` 함수는 그래픽 창을 닫기 전까지 설정이 유지된다. `par()` 함수의 인수는 70여개 정도이며 주요 인수들은 다음과 같다.

- adj : 텍스트의 정렬방식을 정의하는 것으로 0은 좌측정렬, 1은 우측정렬, 0.5는 중앙정렬을 의미한다. 두 값이 주어지면 각각 수평정렬과 수직정렬의 형식을 정의하는 인수로서의 역할을 한다.
- bg : 배경화면의 색을 정의한다.
- bty : 그래프를 둘러싸는 상자의 모양을 정의한다.
- cex : cex(character expansion) 인수는 문자나 점의 크기를 설정하는 역할을 하는 것으로 축, 레이블, 주제목과 부제목에 대해 동일한 역할을 하는 인수로 cex,axis, cex.lab, cex.main, cex.sub 인수가 있다.
- col : 기호의 색을 정의한다. cex 인수와 같이 다른 대상에 대해 동일한 역할을 하는 col.axis, col.lab, col.main, col.sub 인수가 있다.
- font : 텍스트의 글꼴 형식을 정의하는 것으로 1은 보통, 2는 이탤릭, 3은 굵은체, 4는 굵은 이탤릭체를 의미한다. cex 인수와 마찬가지로 font,axis, font.lab, font.main, font.sub 인수가 있다.
- las : 축의 레이블의 표현형식을 정의하는 인수로 정수로 정의된다. 1은 축과 평행, 2는 수평, 3은 축에 수직, 4는 수직을 의미한다.
- lty : 실선, 점선 등 선의 형식을 정의하는 인수이다.
- lwd : 선의 굵기를 정의한다.
- mar : 축과 그래프의 경계사이의 공간을 정의한다.
- mfcol : 그래픽 윈도우를 분할하는 기준을 정의하는 인수로 c(nr,nc)로 정의되면 총 nr x nc 의 분할 화면이 생성된다. 생성되는 그래프는 열을 중심으로 순서대로 자리한다.
- mfrow : mfcol과 동일한 역할을 하는 인수로 생성되는 그래프는 행을 중심으로 순서대로 자리한다.
- pch : 기호의 형태를 정의하는 인수로 1~25 사이의 정수를 갖는다.
- ps : 텍스트의 크기와 기호의 크기를 정의하는 정수값의 인수이다.
- pty : 그래프 영역의 형태를 정의하는 인수로 "s"는 x축과 y축의 비율이 동일하게 설정하고 "m"은 최대 크기로 정의한다.
- tck : 축의 눈금길이를 정의하는 것으로 그래프 영역의 폭과 높이 중 작은 것을 1이라하고 tck 비율에 따라 눈금의 길이가 정의된다.
- tcl : 축의 눈금길이를 정의하는 인수로 tck 인수와 다른 점은 cex=1일 경우, 문자의 길이 1을 기준으로 정의한다.
- xaxt : 축의 눈금 및 값들의 표현 여부를 정의하는 인수로 xaxt=n 이면 x축에는 눈금이나 값들이 표시되지 않는다.
- yaxt : 축의 눈금 및 값들의 표현 여부를 정의하는 인수로 yaxt=n 이면 y축에는 눈금이나 값들이 표시되지 않는다.

제 3 장

데이터파일 작성하기

본 장에서는 R을 이용하여 통계분석에 앞서 분석을 위한 데이터파일을 만들어보고, 외부 데이터를 불러들이는 방법에 대해서 알아보자.

3.1 자료 입력하기

- R을 이용한 통계분석의 첫 단계는 분석의 대상이 되는 자료를 입력하는 것이다. R에서 자료를 불러들이는 방법은 크게 세 가지가 있다.

3.1.1 직접 입력하기

- R에서 자료를 직접입력 하는 것은 앞장의 자료구조를 이용하여 원하는 데이터를 입력하는 것으로 벡터, 행렬, 데이터 프레임, 배열, 리스트, 요인, 시계열 자료 등을 이용한다.

3.1.1.1 배열(array)

- 배열은 자료의 유형이 동일한 데이터 원소들로 구성된 2차원 이상의 자료 구조를 갖는 자료 객체이다. 행렬 역시 배열의 한 종류이며 일반적으로 배열이라 함은 3차원 이상의 자료구조를 갖는 경우를 말한다.

- array() : matrix() 함수와 사용법이 거의 동일하며 속성 역시 행렬과 동일하다.

3.1.1.2 리스트(list)

- 서로 다른 자료 유형으로 구성이 가능한 리스트는 각 차원별 원소의 개수가 동일해야 한다는 제한도 없어 자료 객체 중 가장 자유로운 구조를 허락한다.

- list() : 리스트는 list() 함수를 통해 생성되며, 자료 구조의 유형(mode)과 구성 성분의 수(length), 성분의 이름(names)등의 속성을 가진다.

3.1.1.3 요인(factor)

- 벡터 객체 중 범주형 데이터를 원소로 갖는 자료 객체를 요인(factor)이라 한다.

- 요인에는 범수(level)의 순서가 정의된 순서형 요인과 그렇지 않은 명목형 요인으로 구분된다.

- 요인 객체를 생성하는 함수로 대표적인 것은 factor()와 ordered()가 있다.

- 추가로 수치형 변수에 범위를 지정해 범주형 데이터를 생성하는 함수로는 cut(), split(), findInterval() 등이 있다. 이들에 대한 설명은 help 창을 통해 살펴보자.

3.1.1.4 시계열(time series)

- 일, 월, 년, 시간 등과 같이 일련의 시계열 자료를 표현하는 데이터를 나타내는 자료 객체가 시계열 객체이다.
- 벡터나 행렬의 자료 객체에 관찰치의 시간 정보가 추가되어 생성된다.
- ts() : 시계열 자료를 생성하는 가장 기본적인 방법은 ts() 함수를 이용하는 것이다.
자세한 사항은 help 창을 통해 살펴보자.

3.1.2 복사해서 붙이기

- 복사해서 붙이는 방법은 간단하게 구현할 수 있다.
- ex) 엑셀에서 데이터를 입력한 후 원하는 부분을 선택하고 Ctrl+c를 눌러 복사한다.
- 복사한 내용은 clipboard라는 곳에 임시로 저장된다.
- read.table() : 테이블 형식의 외부 파일로부터 데이터를 읽어 데이터 프레임을 생성한다.
데이터 구분자는 공백을 기본으로 한다.

	A	B	
1	성별	연령	직업
2		1	3
3		1	2
4		1	5
5		1	2
6		1	4
7		2	3
8		1	3
9		1	2

```
> read.table('clipboard',header=T)
  성별  연령  직업
1    1    3
2    1    2
3    1    5
4    1    2
5    1    4
6    2    3
```

- 엑셀만이 아니라 워드, SPSS, 웹페이지, 텍스트파일, CSV 파일도 같은 방법으로 불러들일 수 있다.

3.1.3 파일에서 불러오기

- * file.choose() : 파일을 불러올시 file.choose()를 이용하면 윈도우 탐색기창을 통해 쉽게 선택할 수 있다.
- * read.table(file.choose(), header=T, sep = "", as.is = T, na.strings = ".")

3.1.3.1 텍스트파일 형식의 자료 불러오기.

1) 텍스트파일 만들기

일반적으로 MS Word나 한글, notepad 등에서 자료를 입력한 후 R에서 불러오는 방법을 많이 사용하고 있다. 이 경우 프로그램에서 [파일] → 다른이름으로저장을 선택한 후 텍스트(txt)파일로 저장을 해주어야 한다. [파일] → 저장의 경우 각 프로그램에서 제공하는 기본 확장자로 저장되어 R 프로그램에서 해당 파일을 처리할 수 없게 되는 경우가 생긴다.

2) 텍스트파일 불러오기

- read.table 함수는 텍스트 파일을 읽어서 데이터프레임으로 만들어준다. 기본적인 사용법은 다음과 같이 간단히 구현된다.

- 텍스트 파일의 첫 줄에 변수명이 포함되어 있다면, header를 참(T)으로 설정한다.

test.txt	read.table('test.txt')	test.txt	read.table('test.txt', header=T)
1 홍길동 22 2 가나다 20 3 동상옥 28	V1 V2 V3 1 1 홍길동 22 2 2 가나다 20 3 3 동상옥 28	id name age 1 홍길동 22 2 가나다 20 3 동상옥 28	id name age 1 1 홍길동 22 2 2 가나다 20 3 3 동상옥 28

- 구분자가 쉼표(,)로 구분되어 있을 때, sep=","을 지정한다.

test.txt	read.table('test.txt', header=T, sep=",")
id,name,age 1,홍길동,22 2,가나다,20 3,동상옥,28	id name age 1 1 홍길동 22 2 2 가나다 20 3 3 동상옥 28

3.1.3.2 CSV 형식의 자료 불러오기.

- CSV(Comma-Separated Values)은 엑셀을 비롯한 여러 계산 프로그램들이 지원하는 파일형식이다.

- 변수 사이를 빈 칸 대신 쉼표로 구분자를 이용하는 텍스트 파일이기 때문에 read.table 함수에서 header=T, sep=","으로 설정하면 되지만 간단히 read.csv 함수를 사용해도 된다.

```
>read.csv("test.csv")
```

- read.csv 함수는 read.table 함수를 바탕으로 만들어졌기 때문에 다른 모든 옵션의 사용법이 동일하다.

- read.delim() : read.table()과 기능이 같으며 데이터 구분자는 'Tab(\t)'이다.

3.1.3.3 SPSS 형식의 자료 불러오기.

- foreign 패키지를 이용하면 다른 통계패키지의 데이터 파일을 불러들일 수 있다. SPSS의 SAV파일은 read.spss 함수를 사용한다.

```
> library(foreign)
> dat = read.spss("test.sav")
```

3.1.3.4 Excel 형식의 자료 불러오기.

- ODBC(Open DataBase Connectivity)란 서로 다른 종류의 데이터베이스에 동일한 방법으로 접속할 수 있도록 만든 표준이다. RODBC 패키지를 이용하면 R에서도 DB에 있는 자료를 불러오거나 DB에 자료를 저장할 수 있다. MS 엑셀도 ODBC로 접속할 수 있다.
- 먼저 다음과 같은 test.xls라는 엑셀 파일이 있다고 하자.

	A	B	C	D	E	F	G
1	성별	연령	직업	거주지	워드프린트이용빈도	워드프린트선호도	워드프린트중요도
2		1	3	2	2	1	1
3		1	2	2	2	1	3
4		1	5	2	2	1	1
5		1	2	2	2	1	1
6		1	4	2	2	1	1
7		2	3	2	2	1	1
8		1	3	2	2	1	1
9		1	2	1	2	1	1
10		1	2	1	2	1	2
11		1	4	1	2	1	3
12		1	2	1	2	1	1
13		1	2	1	2	1	1
14		2	2	1	3	1	1
15		2	2	1	3	1	2
16		2	2	1	4	1	3
17		2	2	1	4	1	3
18		2	2	1	4	5	4
19		2	2	1	4	1	1
20		2	2	1	4	1	1

- R에서는 다음과 같은 명령으로 불러들일 수 있다.

```

> library(RODBC)           # RODBC 를 불러들인다.
> xls = odbcConnectExcel("data.xls") # 엑셀 파일에 접속한다.
> data = sqlFetch(xls, "data") # 엑셀 파일의 data 시트를 읽는다.
> data

```

※ Comments

- RODBC package를 추가로 설치해 줘야 한다.
- 엑셀 파일에서 데이터를 불러올 때는 데이터가 입력 되어 있는 엑셀 파일이 작업디렉토리로 설정된 디렉토리 안에 있어야 한다. (RGui에서 메뉴 파일 → 디렉토리 변경)
- 디렉토리 이름이 한글로 되어 있으면 에러가 발생할 수도 있다.
- 작업디렉토리에 작업하고자하는 엑셀파일이 존재해야한다.

3.2 데이터 다루기

3.2.1 결측값(missing value)

- R에서 결측값은 NA(not available)로 표기한다. NaN(not a number)도 수치가 아닌 것을 뜻하며 결측값과 같이 취급한다.
- 결측값을 가지는 자료의 결측값을 제외한 연산을 실행하기 위해서는 na.rm 이라는 키워드와 함께 이용한다.

```
> x <- c(3, 3, 3, NA)
> mean(x)
[1] NA
> mean(x, na.rm=T)
[1] 3
```

3.2.2 자료 변환과 데이터 부분세트

- 데이터파일을 이용하여 통계분석을 시행하다보면 데이터파일의 내용을 수정해야 할 필요가 생긴다. 또한 어떤 경우에는 조사가 추가적으로 이루어져 기존 응답자의 자료에 새로운 응답자의 자료가 합쳐져야 할 상황이 발생하여 데이터파일의 수정이 이루어지게 된다.
- 다음은 새로운 변수를 생성하는 방법이다. transform을 이용하여 기존list를 사용해서 새로운 list를 생성한다.

```
> x <- c(50, 45, 60, 32, 55)
> y <- c(44, 65, 30, 70, 60)
> da <- list(kor = x, eng = y)
> da2 <- transform(da, tot = kor + eng)
> da2
  kor eng tot
1  50  44  94
2  45  65 110
3  60  30  90
4  32  70 102
5  55  60 115
```


- subset : 원하는 부분만 골라내기.
- ex) tot가 100 이상인 데이터만 출력하기.
- []를 이용하여 표현할 수도 있다.

<pre>> subset(da2, tot>100) kor eng tot 2 45 65 110 4 32 70 102 5 55 60 115</pre>	<pre>> da2[da2\$tot>100,] kor eng tot 2 45 65 110 4 32 70 102 5 55 60 115</pre>
-------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------

- 원하는 변수만 출력하기

<pre>> da2.result = subset(da2, select=c(tot)) > da2.result tot 1 94 2 110 3 90 4 102 5 115</pre>	<pre>> da2[c(3)] tot 1 94 2 110 3 90 4 102 5 115</pre>
-------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------

- 두 방법을 함께 이용하는 것도 가능하다.
- 앞의 결과는 list이고, 뒤의 결과는 벡터이다.

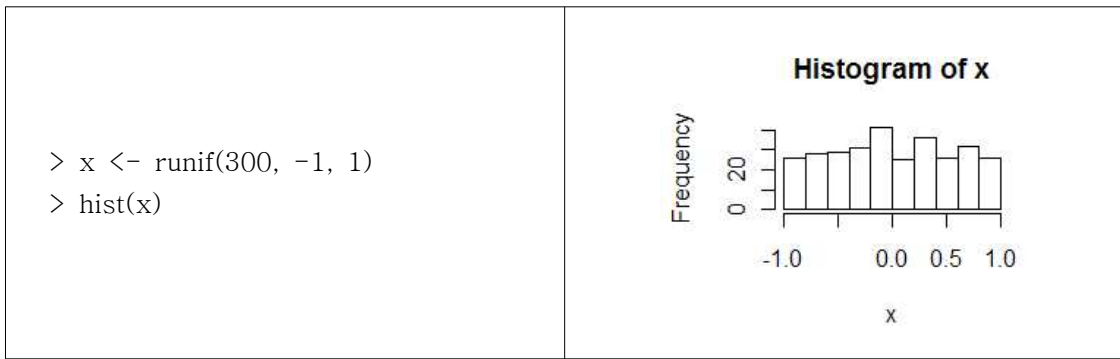
<pre>> subset(da2, kor > 40, select = c(3)) tot 1 94 2 110 3 90 5 115 > da2[da2\$kor > 40, c(3)] [1] 94 110 90 115</pre>

- 자료를 요약하기 : aggregate()
- 만약 전체의 평균이 아니라 각 부분의 평균을 구하고 싶다면 어떻게 할까?
- aggregate(계산할 값, 자료를 여러 부분으로 나누는 기준, 계산할 함수)
- 계산할 값에 벡터가 아니라 데이터프레임이 오면 모든 열에 대하여 함수를 적용한다.

3.2.3 임의의 데이터 생성

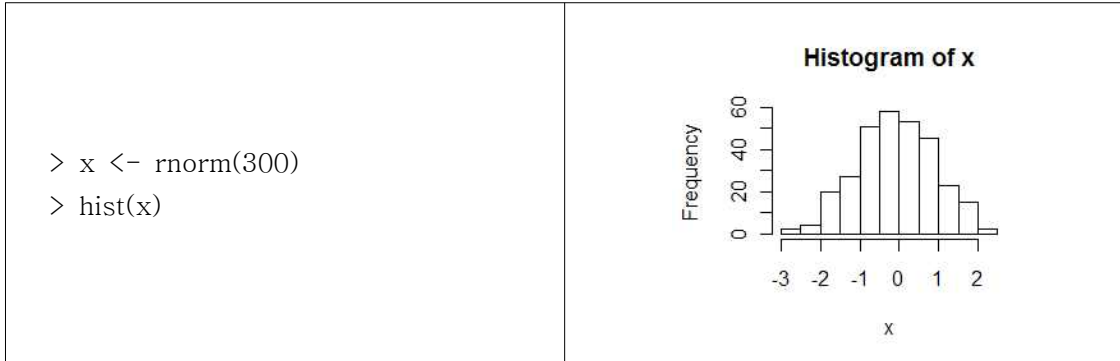
- 균일분포(uniform distribution)

- `runif(n, min, max)` `n` : 생성시킬 임의 수의 개수
- `min` : 균일분포의 최소경계 `min = 0`인 경우 생략 가능
- `max` : 균일분포의 최대경계 `max = 1`인 경우 생략 가능



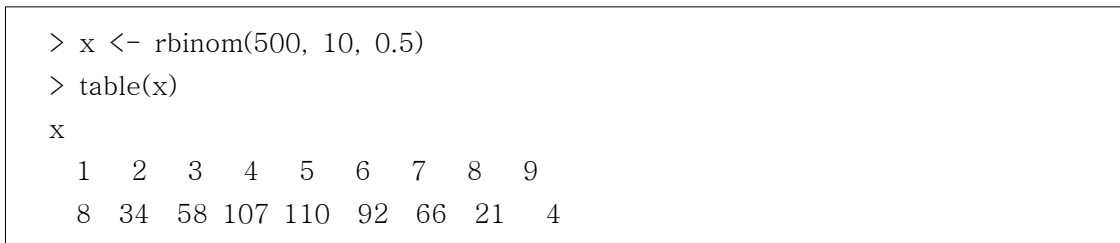
- 정규분포(normal distribution)

- `rnorm(n, mean, sd)` `n` : 생성시킬 임의 수의 개수
- `mean` : 모분포의 평균 `mean = 0`인 경우 생략 가능
- `sd` : 모분포의 표준편차 `sd = 1`인 경우 생략 가능



- 이항분포(binomial distribution)

- `rbinom(n, size, prob)` `n` : 생성시킬 임의 수의 개수
- `size` : 크기
- `prob` : 확률



- 포아송 분포(Poisson distribution)
- rpois(n, lambda) n : 생성시킬 임의 수의 개수
 lambda : 포아송 분포의 평균

```

> x <- rpois(500,5)
> table(x)
x
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14
2 23 52 66 97 81 66 50 34 10 12  4  1  1  1

> x
 [1]  2  2 10  7  2  5  7  8  7  5  6  6  6  4  7  4  3  6  4  3
[21]  3  4  2  8  3  5  6  2  6  4  1  5  4  6  6  4  2  5  3  8
[41]  4  4  6 10  7  4  2  1  4  8  5  5  7  4  6  7  3  6  6  2
[61]  6  5  3  2  7  3  7  6  8  2  2  5  5  3  2  3 10  3  6  5
[81]  5  8  7  1  4  5  7  8  1  2  2  2  7  3  6  8 10  5  7  2
[101] 7  9  4  4  2  4  3 11  2  6  2  5  3  6  4  3  4  9  4  8
[121] 5  3  4  7  3  5  3  6  7  2  7  4  3 11  5  4  6  5  7  7
[141] 6  6  6  8  2  8  5  6  2  3  4  6 11  8  5  2  7  2  1  2
[161] 9  6  0  2  6  5  4  1  1  7  4  2  6  4  8  3  3  1  4  5
[181] 7  7  8  5  8  1  2  6  5  8  7  2  9  7 14  7  3  7  4  5
[201] 8  5  3  5  4  7  4  4 10  5  2  4  8  4  4  5  7  7 13  4
[221] 5  7  4  6  3  3  3  3  2  4  4  5  4  6  7 10  4 10 11  1
[241] 2  0  3  8  5  7  5  6  4  7  8  5  5  2  4  4  3  6  4  2
[261] 1  2  2  5  5  6  6  3  4  2  7  3  5  3  4  5  7  4  8  3
[281] 4  6  6  5  3  6  4  4  3  2  4  5  9 10  4  6  6  3  3  3
[301] 4  4  1  1  6  4  8  4  8  5  3  2  6  3  7  4  3 10  7  3
[321] 6  4  5  7  8 10  4  2  4  7  4  1  3  5  9  2  2  4  6  1
[341] 5  4  5  5  4  3  6  9  6  4  6  4  7  9  3  4  4  5  6  4
[361] 6  8  4  3  8  8  4  5  2  6  4  5  5  1  6  3  7  3  6  4
[381] 5  9  1  3  6  5  8  5  4  1  5  1  8  5  1  6  3  3  6 10
[401] 4  7  5 12  3  4  5  6  2  5  6  4  5  4  5  2  7  5  7  8
[421] 4  5  4  9  8  7  2  3  3  5  3  3  4  5  5  4  5  3  6  5
[441] 2  3  3  1  5  5  4  4  4  3  5  4  5  8  6 10  6  2  2  4
[461] 5  5  3  4  2  8  4  6  5  7  4  4  6  3  4  3  3  7  5  8
[481] 7  4  1  4  4  5  2  5  4  6  1  7  4  6  6  5  3  6  2  5

```

제 4 장

두 집단간 평균과 비율 차이에 대한 검정

4.1 R을 이용한 독립표본 t-test

- 두 집단간 평균 차이의 검정은 t 검정을 사용한다. t 검정은 두 모집단이 정규분포를 따르며, 분산이 같다는 가정을 한 후, 집단간 자료의 평균에서 차이가 있는가를 검정하기 위해 이용되는 통계기법이다. 예를 들어 남학생과 여학생의 학업성취능력에 대한 평균이 같은지, 어느 지역의 A와 B간에 사교육비에서 차이가 있는지를 분석하고자 할 때 활용할 수 있다.

4.1.1 Dataset

- 다음은 어떤 어느 대학에서 중간고사 후의 성적을 나타낸 것이다. A 그룹의 데이터는 A반 학생 20명에 대해서 측정한 것이며, B 그룹의 데이터는 B반 학생 18명에 대해서 조사한 것이다. A반 학생과 B반 학생의 성적에는 차이가 있다고 말할 수 있는지 검토하여야.

A 그룹	80 77 85 95 60 70 65 90 45 70
	87 56 79 84 63 71 85 49 52 93
B 그룹	51 63 88 91 47 76 82 61 59 63
	88 73 76 69 89 67 76 47

4.1.2 가설의 설정

- A반과 B반 학생들의 성적을 각각 μ_1 과 μ_2 라 하면, 다음과 같이 가설이 설정된다. 이와 같은 경우는 양측검정(two-sides test)이 이루어진다.

$$H_0 : \mu_1 = \mu_2$$
$$H_1 : \mu_1 \neq \mu_2$$

- 만약 연구자가 과거의 경험이나 사전조사를 통하여 한 표본집단의 점수(값)가 다른 집단에 비해 높거나 낮다라는 사전지식이 있는 경우 단측검정을 실시할 수도 있다.

4.1.3 검정통계량

- 중심극한정리(CLT : Central Limit Theorem)에 의하면, 표본수 n이 커짐에 따라서 표본 평균 \bar{X} 는 평균이 μ 이고 분산이 σ^2/n 인 정규분포로 근사된다. 이를 표준화하면 표본정규 분포가 되므로, 이를 식으로 표현하면 다음과 같다.

$$\frac{\bar{x} - \mu}{\sigma / \sqrt{n}} \sim N(0,1) \text{ as } n \text{ increases}$$

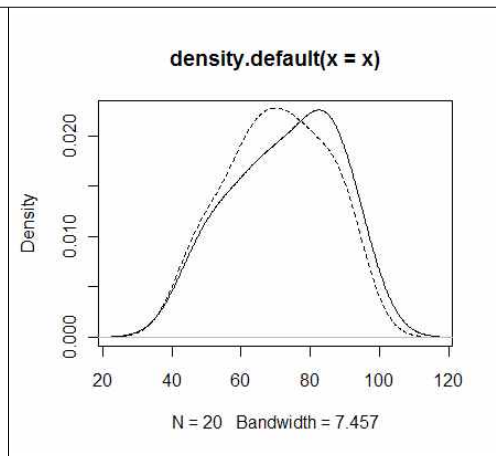
- t 분포는 두 집단의 각 표본수가 30명(개)을 넘는 경우 정규분포와 동일해지지만 엄격한 검정을 위하여 t 검정을 이용하는 것이 일반적이다.

$$t^* = \frac{\bar{X}_1 - \bar{X}_2}{\sigma_{(\bar{X}_1 - \bar{X}_2)}} = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\left(\frac{S_1^2}{n_1}\right) + \left(\frac{S_2^2}{n_2}\right)}}$$

4.1.4 R을 이용한 통계검정

- 데이터의 입력 및 개략적인 정보 확인하기

```
> x=c(80, 77, 85, 95, 60, 70, 65, 90, 45, 70,
+     87, 56, 79, 84, 63, 71, 85, 49, 52, 93)
> y=c(51, 63, 88, 91, 47, 76, 82, 61, 59, 63,
+     88, 73, 76, 69, 89, 67, 76, 47)
> summary(x)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
45.00  62.25   74.00   72.80  85.00   95.00
> summary(y)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
47.00  61.50   71.00   70.33  80.50   91.00
> plot(density(x))
> lines(density(y),lty=2)
```



- A그룹의 경우 평균이 72.80, B그룹의 경우 평균이 70.33인 것을 알 수 있다.
- plot에서 lty=2 명령어는 점선으로 나타내라는 뜻이다.
- x가 실선, y가 점선으로, 확률밀도함수의 개형이 그림과 같음을 알 수 있다.
- var.test(x,y)를 통해서 등분산성 검정이 가능하다.

- R에서 t-test를 수행하는 경우 명령어는 t.test 이다. 두 집단이 등분산일 경우와 등분산이 아닐 경우 두 가지 검정이 가능하다.

- ① 양측검정으로 등분산을 가정했을 경우
 - var.equal=T : 등분산일 경우
 - alt='two.sided' : 양측검정

```
> t.test(x,y,var.equal=T,alt='two.sided')

Two Sample t-test

data: x and y
t = 0.5171, df = 36, p-value = 0.6083
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.20813 12.14146
sample estimates:
mean of x mean of y
 72.80000  70.33333
```

검정결과를 살펴보면 계산되어진 검정통계량값은 0.5171이고 양측검정의 p-value=0.6083으로 반별로 중간고사 성적에는 유의수준 5%에서 유의한 차이가 발생하지 않는 것으로 결론을 내리게 된다.

- ② 단측검정으로 등분산을 가정했을 경우
 - alt='less' : 대립가설에서 왼쪽 집단(x)이 작다고 설정

```
> t.test(x,y,var.equal=T,alt='less')

Two Sample t-test

data: x and y
t = 0.5171, df = 36, p-value = 0.6959
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
 -Inf 10.52050
sample estimates:
mean of x mean of y
 72.80000  70.33333
```

단측검정의 결과 p-value=0.6959로 역시 유의하지 않게 나타났다.

- ③ 양측검정으로 등분산을 가정하지 않을 경우
- var.equal의 옵션을 F로 설정한다.

```
> t.test(x,y,var.equal=F,alt='two.sided')
```

```
Welch Two Sample t-test
```

```
data: x and y  
t = 0.5187, df = 35.912, p-value = 0.6071  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
-7.178187 12.111520  
sample estimates:  
mean of x mean of y  
72.80000 70.33333
```

- ④ 단측검정으로 등분산을 가정하지 않을 경우

```
> t.test(x,y,var.equal=F,alt='less')
```

```
Welch Two Sample t-test
```

```
data: x and y  
t = 0.5187, df = 35.912, p-value = 0.6964  
alternative hypothesis: true difference in means is less than 0  
95 percent confidence interval:  
-Inf 10.49541  
sample estimates:  
mean of x mean of y  
72.80000 70.33333
```

등분산을 가정하지 않았을 때는 Welch test라는 말이 결과에 표시된다. 자유도가 작아지고, 유의확률 또한 높아짐을 알 수 있다.

4.2 R을 이용한 paired t-test

- 앞서 제시한 두 집단간의 평균 차이에 대한 t-test에서는 두 집단이 서로 독립이라 가정하였다. 그러나 동일 표본집단에서는 이러한 가정이 성립하지 않는다.
- paired t-test는 사전사후검정이라 칭하기도 하며, 같은 집단을 시간을 두고 반복 관찰한 값이 달라졌는가를 알아보는 데 많이 사용된다. 예를 들어, 약의 복용 결과나 지지율의 변화 등과 같이 사람의 경우 어떤 사건의 발생 혹은 시간의 변화가 그 사람의 성향을 변화시켰는지를 알아보기 위하여 사용되어진다.

4.2.1 Dataset

- 탈모치료제로 쓰이는 Finasteride 성분의 투약결과이다. (Verzani, J . (2005) p.243)

Group	score								
Finasteride Treatment	5	3	5	6	4	4	7	4	3
placebo	2	3	2	4	2	2	3	4	2

4.2.2 가설의 설정

- 이 경우 가설은 다음과 같이 설정되며, 양측검정을 통하여 검정이 실시된다.

$$H_0 : \mu_d = 0$$

$$H_1 : \mu_d \neq 0$$

(단, μ_d 는 관측값 차이의 평균)

4.2.3 검정통계량

- 대응표본에 대한 차이 검정에 이용되는 검정통계량은 t 통계량이며, 이 t 통계량은 n-1의 자유도를 갖는다.

$$t^* = \frac{\bar{d}}{S_d / \sqrt{n}}$$

- 가설의 채택기준은 유의수준 5%를 이용한다.

4.2.4 R을 이용한 통계검정

- 데이터의 입력 및 출력 결과

- paired=T : paired t-test를 수행하겠다. 반대는 F로 설정.

```
> Finasteride=c(5,3,5,6,4,4,7,4,3)
> placebo=c(2,3,2,4,2,2,3,4,2)

> t.test(Finasteride, placebo, paired=T, alt='two.sided')

Paired t-test

data:  Finasteride and placebo
t = 4.1538, df = 8, p-value = 0.003192
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.8402524 2.9375254
sample estimates:
mean of the differences
      1.888889
```

- 분석결과 먼저 t 검정통계량값, 자유도, p-value가 제시된다. 다음 하한, 상한 값이 제시되며 마지막으로 두 변수간의 차이의 평균을 나타낸다. 위의 결과를 살펴보면 평균의 차이는 1.888889로 Finasteride를 사용했을 때 위약(placebo)를 사용했을 때보다 약 1.89 정도의 모발의 증가가 나타나는 것을 알 수 있다. p-value는 0.003으로 통계적으로 유의한 차이가 발생하는 것으로 결론을 내릴 수 있다.

4.3 R을 이용한 χ^2 독립성 검정

- χ^2 독립성 검정 또는 χ^2 검정은 집단간에 명목 또는 서열척도로 측정된 변수(속성)에 있어 차이가 있는지를 알아보는 데 사용되는 통계기법이다. 예를 들면, 남학생과 여학생에 있어서 애완동물을 기르고 있는 사람의 비율에는 차이가 있는지를 확인하거나 남녀간에 지난 한 달간 A아이스크림 상표 구입여부에 차이가 있는지를 알아보는 데 사용되는 분석방법이다.

4.3.1 Dataset

- A지역의 대학에 다니는 학생들 중 무작위로 남녀 50명씩 추출하여 다음과 같은 설문조사를 실시하였다.

- (설문1) 귀하의 성별을 선택해 주십시오.

A, 남 B, 여

- (설문2) 귀하는 댁에서 애완동물을 키우고 있습니까?

A, 그렇다 B, 그렇지 않다

4.3.2 교차분석 및 가설 설정

- 위의 설문에 의하여 연구자는 남, 여학생에 따라 애완동물을 키우는 빈도에 차이가 있는지를 확인하고자 한다. 먼저 수집된 데이터를 가지고 교차분석표(cross table) 또는 상황표(contingency table)를 작성하여 보자.

- R을 이용한 교차분석표

<pre>> nic = read.table(file="prop.txt", header=T) > y = table(nic\$group, nic\$yn) > y</pre>	<pre> 1 2 1 12 38 2 18 32</pre>
--------------------------------------------------------------------------------------------------------	-----------------------------------

- 작성된 교차분석표

	그렇다	그렇지 않다	계
남학생	12	38	50
여학생	18	32	50
계	30	70	100

- 통계적으로 유의한 차이가 있는지를 확인하기 위하여 다음과 같이 가설을 설정한다.

H_0 (귀무가설) : 남, 여학생 별로 애완동물을 키우는 사람의 비율에는 차이가 없다.

H_1 (대립가설) : 남, 여학생 별로 애완동물을 키우는 사람의 비율에는 차이가 있다.

- χ^2 검정의 귀무가설에는 두 변수간의 관계가 서로 독립적임을 제시하며 대립가설에는 두 변수간에 어떤 관계가 있음을 제시하게 된다.

4.3.3 검정통계량

- 다음과 같은 공식을 사용하여 χ^2 검정통계량을 계산하는데, 이 통계량은 (행의 수-1)*(열의 수-1)의 자유도를 갖는 χ^2 분포를 따르게 된다. 즉 여기서는 (2-1)*(2-1)=1의 자유도를 갖게 된다.

$$\chi^2 = \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

4.3.4 R을 이용한 통계검정

- 데이터의 입력 및 출력 결과

- correct=F : Yates의 보정을 하지 않은 Pearson의 카이제곱검정

```
> x=matrix(y, nc=2) ## nc는 컬럼의 숫자 ex) 2 by 2
> chisq.test(x, correct=F)

Pearson's Chi-squared test

data: x
X-squared = 1.7143, df = 1, p-value = 0.1904

> chisq.test(x) # with Yates' continuity correction

Pearson's Chi-squared test with Yates' continuity correction

data: x
X-squared = 1.1905, df = 1, p-value = 0.2752
```

- 검정결과 χ^2 검정은 일반적으로 Pearson 분석이 널리 이용된다. 본 예에서는 검정통계량 값은 1.7143, p-value는 0.1904로 5% 유의수준에서 귀무가설은 기각되지 않는다. 즉, 남 학생과 여학생에 있어서 애완동물을 키우고 있는 사람의 비율에는 차이가 없다.

- 두 번째 결과는 연속수정 값이며, 2x2 표에 대해서만 계산되어진다.
- 도수가 5 미만일 경우 정밀도가 나빠져 결과를 신용할 수 없다. 이와 같은 경우 Fisher의 직접확률검정(Fisher's exact test)을 실시하는 것이 좋다.

```

> fisher.test(x)

Fisher's Exact Test for Count Data

data: x
p-value = 0.2752
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.2129583 1.4539313
sample estimates:
odds ratio
 0.5646665

```

- 검정결과 여기서의 양쪽검정이 이루어 졌으며, p-value는 0.2752로 유의수준 0.05보다 크기 때문에 귀무가설을 기각하지 못하는 결론을 내리게 된다.

- 만약 두 집단의 비율이 대응이 있다면 맥네마(McNemar)의 검정을 실시한다. 예를 들어,

<p>- (설문1) 귀하는 휴대폰을 가지고 있습니까? A, 예 B, 아니요</p>	<p>- (설문2) 귀하는 호출기를 가지고 있습니까? A, 예 B, 아니요</p>
------------------------------------------------------------------------------------	------------------------------------------------------------------------------------

이와 같은 경우 휴대폰과 호출기의 양쪽을 가지고 있는 사람이 존재한다. 이때 검정하는 방법이 맥네마 검정이다.

```

> mcnemar.test(x, correct=F)

McNemar's Chi-squared test

data: x
McNemar's chi-squared = 7.1429, df = 1, p-value = 0.007526

> mcnemar.test(x) # with continuity correction

McNemar's Chi-squared test with continuity correction

data: x
McNemar's chi-squared = 6.4464, df = 1, p-value = 0.01112

```

- 검정결과 만약 이 데이터가 대응이 있는 설문의 데이터이면, p-value는 0.01112로 귀무가설은 기각된다. 즉, 휴대폰과 호출기를 가지고 있는 사람의 비율에는 차이가 있다.

제 5 장

분산분석

- 여러 집단을 총괄적으로 분석할 수 있는 분산분석(analysis of variance : ANOVA)은 1919년 피셔(R. A. Fisher; 1890-1962)에 의해 고안된 방법으로서, 실험계획법(experimental design)과 회귀분석(regression analysis)에 주로 사용되어 왔다.

5.1 일원배치 분산분석

- 분산분석은 두 표본 이상의 평균치에 대한 차이를 검정하는 통계기법이다. 이 분산분석을 이용하여 표본들이 동일한 평균을 가진 모집단에서 추출된 것인지의 여부를 추론할 수 있다. 예를 들면, 분산분석의 이용은 통계학을 수강한 학생들의 점수[종속변수: 비율척도 또는 등간 척도]에 대해 학년별[독립변수: 명목척도]평균의 차이가 있는지를 살펴볼 수 있다. 그리고 이러한 차이가 통계적으로 유의한 것인지를 파악할 필요가 있는데 이 같은 상황에서 두 집단 이상의 한 변수에 대한 평균의 차이를 검정하고자 할 때 이용한다.

5.1.1 Dataset

- 다이어트 식품으로 알려진 A, B, C, D 네 가지 식품의 콜레스테롤 함유량을 비교하려고 한다. 각 식품별로 세 개의 제품을 추출하여 콜레스테롤 함유량을 측정한 결과 다음과 같다

종류	함유량 (단위 : mg)		
A	3.6	4.1	4.0
B	3.1	3.2	3.9
C	3.2	3.5	3.5
D	3.5	3.8	3.8

5.1.2 가설의 설정

- 분산분석에서 가설검정은 집단간에 종속변수의 평균값이 동일한지를 확인하는 것이다. 식품별 콜레스테롤 함유량에 차이가 있는지를 검정하고자 할 경우 귀무가설과 대립가설은 다음과 같이 제시된다.

$$H_0 = \mu_1 = \mu_2 = \mu_3 = \mu_4$$

(또는 식품별 콜레스테롤 함유량에 차이가 없다.)

H_a : 집단간 평균에 차이가 있다.

(또는 식품별 콜레스테롤 함유량에 차이가 있다.)

5.1.3 검정통계량

- F분포는 분산의 비교를 통해 얻어진 분포비율이다. 이 비율을 이용하여 각 집단의 모집단분산이 차이가 있는지에 대한 검정과 모집단평균이 차이가 있는지 검정하는 방법으로 사용한다. 즉 $F = (\text{군간변동})/(\text{군내변동})$ 이다. 만약 군내변동이 크다면 집단간 평균차이를 확인하는 것이 어렵다. 분산분석에서는 집단간의 분산의 동질성을 가정하고 하기 때문에 만약 분산의 차이가 크다면 그 차이를 유발한 변인을 찾아 제거해야 한다. 그렇지 못하면 분산분석의 신뢰도는 나빠지게 된다.

- 가정

1) 정규성 가정

두 모집단에서 변인 Y는 정규분포를 따른다.

두 모집단에서 Y의 평균은 다를 수 있다.

2) 분산의 동질성 가정

Y의 모집단 분산은 두 모집단에서 동일하다.

3) 관찰의 독립성 가정

두 모집단에서 크기가 각각 n_1, n_2 인 표본들이 독립적으로 표집된다.

- 두 표본에서 산출된 모집단 분산의 추정치의 비율을 구한다. 이를 'F' 또는 'F 통계치'라고 한다. F 값들은 특정한 이론적 확률분포를 따르게 되는데 이것이 F 분포이다.

< 분산분석표 >

변동의 요인	자승합	자유도	자승평균	F
집단간 변이	$SSB = \sum_{j=1}^C n_j (\bar{Y}_j - \bar{Y})^2$	c-1	$MSB = \frac{SSB}{c-1}$	$\frac{MSB}{MSW}$
집단내 변이	$SSW = \sum_{j=1}^C \sum_{i=1}^N (\bar{Y}_{ij} - \bar{Y}_j)^2$	N-c	$MSW = \frac{SSW}{N-c}$	
총변이	$SST = \sum_{j=1}^C \sum_{i=1}^N (Y_{ij} - \bar{Y})^2$	N-1		

- 집단간 차이가 유의하기 위해서는 집단내 변이는 가능한 적어야 하며 집단간 변이는 가능한 커야 하기 때문에 집단간 변이와 집단내 변이의 상대적 비율을 나타내는 F 통계량을 이용하여 집단간 차이에 대한 검정을 한다.

5.1.4 R을 이용한 통계검정

- 데이터의 입력방식은 다음과 같이 두 방법으로 입력가능하다.

```
> ## prepare the data
> ## method 1
> data1 <- c(3.6, 4.1, 4.0, 3.1, 3.2, 3.9, 3.2, 3.5, 3.5, 3.5, 3.8, 3.8)
> group1 <- as.factor(c(rep(1,3), rep(2,3), rep(3,3), rep(4,3)))
> edu <- data.frame(cbind(data1, group1))

> ## method 2
> x1 = c(3.6, 4.1, 4.0)
> x2 = c(3.1, 3.2, 3.9)
> x3 = c(3.2, 3.5, 3.5)
> x4 = c(3.5, 3.8, 3.8)
> data2 = data.frame(x1,x2,x3,x4)
> df=stack(data.frame(x1,x2,x3,x4))
```

- 차이점에 대해서는 비교해 보길 바란다.

```
> oneway.test(values ~ ind, data=df, var.equal=T) ## 등분산을 가정

One-way analysis of means

data: values and ind
F = 2.25, num df = 3, denom df = 8, p-value = 0.1598

> anova(lm(values ~ ind, data=df))
Analysis of Variance Table

Response: values
      Df Sum Sq Mean Sq F value Pr(>F)
ind     3   0.54   0.18   2.25 0.1598
Residuals 8   0.64   0.08
```

- method 1 방법으로 데이터 입력시 아래와 같이 분석한다.

```
- oneway.test(edu$score ~ edu$group, data = edu) # 일원분산분석
- anova(lm(edu$score ~ edu$group, data = edu))
```

- 결과 해석

p-value = 0.1598로 유의수준 0.05 보다 크기 때문에 귀무가설을 기각할 수 없다. 즉, 다이어트 식품별 콜레스테롤 함유량이 다르다고 할 수 없다.

- 사용자 직접 구현

```
> data <- read.table("D:/R/분산분석.txt", header=T) ## 메모장 자료를 불러온다.
> t <- ncol(data) ## 열의 갯수를 t라 하고 행의 갯수를 r이라 한다.
> r <- nrow(data)
> N <- t * r
> totalmean <- sum(data)/N ## 각각의 평균을 구한다.
> meanA <- sum(data[,1])/r
> meanB <- sum(data[,2])/r
> meanC <- sum(data[,3])/r
> meanD <- sum(data[,4])/r
> submean <- c(meanA, meanB, meanC, meanD) ## 구해진 평균을 벡터로 저장.
> SST <- 0
> SSR <- 0
>
> ## for 문을 사용하여 총제곱합과 처리제곱합을 구한다.
> for(j in 1:r){
+ SSR <- r*(submean[j] - totalmean)^2 + SSR
+ for(i in 1:t){
+ SST <- (data[i,j] - totalmean)^2 + SST
+ }
+ }
>
> SSE <- SST - SSR ## 오차 제곱합은 총제곱합에서 처리제곱합을 뺀값
> MSR <- SSR/(t-1)
> MSE <- SSE/(N-t)
> F <- MSR / MSE
> p <- 1-pf(F, t-1, (t*r)-1)
>
> ## 데이터의 출력을 위해 matrix를 만든 후 복차를 만들어준다.
> result <- matrix(c(0),nrow=3,ncol=5,dimnames=list(c("처리","오차","전체"),
+ c("자유도(df)","제곱합(SS)","평균제곱(MS)","분산비(F)","유의확률(p)")))
> result[1,] <- c(t-1,SSR,MSR,F,p)
> result[2,] <- c(N-t,SSE,MSE,0,0)
> result[3,] <- c(N-1,SST,0,0,0)
> result
      자유도(df) 제곱합(SS) 평균제곱(MS) 분산비(F) 유의확률(p)
처리           3      0.51      0.1700  2.344828  0.1290160
오차           8      0.58      0.0725  0.000000  0.0000000
전체          11      1.09      0.0000  0.000000  0.0000000
```

- 결과 해석 : 함수를 이용한 것과 약간의 차이가 있지만 결과는 같은 결과를 나타낸다.

5.2 공변량분석

5.2.1 공변량의 의미

- 공변량(covariance)이란 여러 변인들이 공통적으로 함께 공유하고 있는 변량을 뜻한다. 공변량의 개념을 단일종속변인 변량분석 (univariate analysis of variance)에 적용시키면 독립변인들이 하나의 종속변인에 대해 함께 공유하는 변량, 또는 독립변인과 기타 잡음변인들이 공유하는 변량을 뜻한다. 우리는 어떤 실험을 할 때 구 주요한 목적은 연구하고자 하는 독립변인 이외에 다른 잡다한 요인이 종속변인에 영향을 주는 통제함으로써 가능한 주어진 독립변인의 영향만 순수하게 측정하고자 한다. 그러나 실제적으로 독립변인 이외에 종속변인에 영향을 미칠 수 있는 요인이 적지 않은데 그러한 요인들을 잡음요인 (nuisance factor)이라고 한다. 잡음요인을 통제하는 방법은 실험적 통제와 통계적 통제가 있다.

- 통계적 통제방법이 공변량분석이다. 공변량분석은 독립변인 이외에 다른 잡다한 요인들이 종속변인에 영향을 미치는 것을 통제함으로써 주어진 독립변인의 순수한 영향을 측정하는데 목적이 있다. 이 방법은 변량분석과 회귀분석의 방법을 이용하여 통제하고자 하는 잡음요인을 기준으로 동질적인 사례들을 각 실험집단에 배치하는 결과와 동등한 결과를 가져오도록 하는 데 있다.

- 공변량의 예: 커피의 광고효과를 알기 위한 3가지 내용의 광고 (맛, 향, 분위기)
공변량 - 하루에 마시는 커피의 잔수

- 두 변인 X (커피의 광고를 본 후 마시는 커피의 잔수)와 Y (평소에 마시는 커피의 잔수)의 개별 값들이 평균치 (M_x 와 M_y)로부터 떨어져 있는 거리인 편차 (x 와 y)를 구하고, X와 Y의 서로 대응하는 범주의 x 와 y 값을 곱하여 얻어지는 xy 를 교적 (cross product)이라고 한다. 이 교적을 모두 합한 xy 를 교적화 (sum of cross product)라고 한다. x^2 와 y^2 을 x 와 y 의 자승합 (sum of squares)라고 한다. 이때 xy 를 사례수 N 으로 나눈 값이 공변량이라고 한다.

5.2.2 가설설정

- 공변량분석에서 가설검정은 집단간에 종속변수의 평균값이 동일한지를 확인하고, 공변량의 유의성에 대한 가설을 설정한다. 보관온도와 콜레스테롤 함유량에 차이가 있는지를 검정하고자 할 경우 귀무가설과 대립가설은 다음과 같이 제시된다.

$$H_0 = \mu_1 = \mu_2 = \mu_3 = \mu_4$$

(또는 보관온도와 콜레스테롤 함유량에 차이가 없다.)

H_a : 집단간 평균에 차이가 있다.

(또는 보관온도와 콜레스테롤 함유량에 차이가 있다.)

5.2.3 R을 이용한 공변량분석

```
> data <- read.table("D:/R/ancova.txt", header=T) ## 메모장 자료를 불러온다.
> attach(data)
> group <- factor(group)
> coplot(col~group|temp) ## 시각화 하기위한 표현
> plot(col~group)
> text(temp+0.15,col,group) ## I added 0.15 to offset the label
> plot(col~group,pch=as.numeric(temp))
> ResR1 <- lm(col~group+temp)
> ResSR <- lm(col~group)
> anova(ResSR,ResR1)
```

Analysis of Variance Table

Model 1: col ~ group

Model 2: col ~ group + temp

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	8	0.64000				
2	7	0.57518	1	0.06482	0.7889	0.4039

- 공변량인 보관온도에 대한 가설검정경과에서 F값은 0.7889, p-value=0.4039로 유의하지 않은 것으로 나타났다. 즉, 보관온도에 따라 콜레스테롤 함유량이 다르다고 할 수 없다.

5.3 이원분산분석

- 이원배치법(이원분산분석) : 종속변수가 연속변수이고, 독립변수가 범주변수 2개일 때 사용하는 통계기법이다.
- 이원분산분석에서 두 요인은 행과 열이라고 하며, 각각 A, B로 나타낸다.
- 반복이 없는 것과 반복이 있는 것으로 구분할 수 있다. (교호(상호)작용에서 효율이 있으면 연구가설이 채택, 효율이 없으면 반복 없는 이원배치를 확인함) 따라서, 교호작용의 차이를 먼저 봐야 한다.
- 이원분산분석을 이용하기 위해서는 일원분산분석에서 필요한 가정을 필요로 한다. 즉, 각 표본들은 독립이 되어야 하고, 모든 행과 열의 조합(셀)에 대해 모집단이 정규분포를 이루고, 분산이 같아야 한다. 이원분산분석에서는 일원분산분석과 달리 두 요인, A요인과 B요인이 상호작용을 하지 않는다는 가정을 필요로 한다. 만약 그렇지 않다면 각 요인이 종속변수에 미치는 영향은 명확하게 나타나지 않는다. 따라서 이원분산분석에서는 상호작용이 없다는 가정을 검증할 수 있도록 하고 있다. 이 가정을 만족하는 모형을 부가모형(additive model)이라고 한다. <참고자료 : 이인재 외, 사회복지통계분석, 나남출판>
- 상호작용효과란 한 독립변수가 종속변수에 미치는 효과가 다른 독립변수에 의해 영향을 받는 경우를 말한다.

5.3.1 반복이 없는 이원분산분석

5.3.1.1 가설의 설정

- 독립변수 각각의 주효과와 독립변수들간의 상호작용효과에 대해 이루어지게 된다. 5.1.1 Dataset에서 각 다이어트 식품별로 측정된 세 개의 관측치가 세 곳의 상이한 실험실에서 측정된 결과라고 하자. 이때, 다이어트 식품과 실험실에 따라서 콜레스테롤 함유량에 차이가 있는가를 5% 유의수준으로 검정하여 보자.
- 첫 번째 집단변수에 대한 가설
 - H_{0_1} : 식품에 따라 콜레스테롤 함유량에 차이가 없다.
 - H_{a_1} : 식품에 따라 콜레스테롤 함유량에 차이가 있다.
- 두 번째 집단변수에 대한 가설
 - H_{0_2} : 실험실에 따라 콜레스테롤 함유량에 차이가 없다.
 - H_{a_2} : 실험실에 따라 콜레스테롤 함유량에 차이가 있다.

5.3.1.2 R을 이용한 이원분산분석

```
> data <- read.table("D:/R/twoway.txt", header=T)
> attach(data)
> group <- factor(group)
> class <- factor(class)
> anova(lm(col~group+ class))
Analysis of Variance Table

Response: col
      Df Sum Sq Mean Sq F value Pr(>F)
group   3  0.54000  0.18000   4.9091 0.04692 *
class   2  0.42000  0.21000   5.7273 0.04062 *
Residuals 6  0.22000  0.03667
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- 결과 해석 : 분산분석표에서 식품별 콜레스테롤 평균함유량에 대한 검정에서 p-value = 0.04692로 유의수준 0.05보다 작기 때문에 귀무가설을 기각한다. 즉, 다이어트 식품별 콜레스테롤 평균함유량이 모두 같다고는 할 수 없다.
또한, 실험실별 콜레스테롤 평균함유량에 대한 검정에서 p-value = 0.04062로 유의수준 0.05보다 작기 때문에 귀무가설을 기각한다. 즉, 실험실별로 콜레스테롤 평균함유량이 모두 같다고는 할 수 없다.

5.3.2 반복이 있는 이원분산분석

5.3.2.1 Dataset

- 점포의 크기와 지역에 따라 생활필수품의 가격에 차이가 있는가를 알아보기 위하여 이원 분산분석을 하려고 한다. 점포의 크기를 인자 A로 하고 지역을 인자 B로 할 때 인자수준은 a=2(대, 소), b=3(서울, 중부, 남부)으로 한다. 각 처리에서 표본을 추출한 결과 다음과 같은 데이터를 얻을 수 있었다.

	B1(서울)	B2(중부)	B3(남부)
A1(소)	74 78	78 74	68 72
A2(대)	70 74	68 72	60 64

5.3.2.2 가설의 설정

- 첫 번째 집단변수에 대한 가설
 - H_{0_1} : 점포의 크기에 따라 가격에 차이가 없다.
 - H_{a_1} : 점포의 크기에 따라 가격에 차이가 있다.
- 두 번째 집단변수에 대한 가설
 - H_{0_2} : 지역적 위치가 생활필수품의 가격에 영향을 미치지 않는다.
 - H_{a_2} : 지역적 위치가 생활필수품의 가격에 영향을 미친다.
- 점포의 크기와 지역간에 교호작용에 대한 가설
 - H_{0_3} : 점포의 크기와 지역간에 교호작용이 없다.
 - H_{a_3} : 점포의 크기와 지역간에 교호작용이 있다.

5.3.2.3 R을 이용한 이원분산분석

```
> library(foreign) ## SPSS 파일을 불러오기 위한 옵션
> dat = read.spss("D:/R/twoway2.sav")
> attach(dat)
> size <- factor(size)
> area <- factor(area)
> anova(lm(price~size+ area+ size:area))
Analysis of Variance Table

Response: price
      Df Sum Sq Mean Sq F value Pr(>F)
size    1   108    108    13.5 0.01040 *
area    2   152     76     9.5 0.01382 *
size:area 2     8      4     0.5 0.62974
Residuals 6    48      8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- 결과 해석 :
 - 1) 점포크기의 효과에 대한 검정에서 p-value=0.01040으로 귀무가설을 기각한다. 즉, 점포의 크기에 따라 가격에 차이가 있다.
 - 2) 지역의 효과에 대한 검정에서 p-value=0.01382로 귀무가설을 기각한다. 즉, 지역적 위치가 생활필수품의 가격에 영향을 미친다.
 - 3) 교호작용의 검정에서 p-value=0.62974로 귀무가설을 기각하지 않는다. 즉, 점포의 크기와 지역간에 교호작용은 존재하지 않는다.

5.4 다변량 분산분석

- 다변량 분산분석(multivariate analysis of variance : MANOVA)은 종속변수가 두 개 이상인 경우에 여러 모집단의 평균 벡터를 동시에 비교하는 분석기법이다. 예를 들면, 연령대 별로 국내 프리미엄 아이스크림의 선도 상표인 베스킨라빈스에 대해 서구적이다, 고급스럽다는 이미지 평가점수와 함께 실제 구매의도가격에 차이가 나는지를 동시에 분석할 경우 이용된다. 또한 다변량 분석분석에서는 종속변수의 조합에 대한 효과의 동시검정을 중요시한다. 그 이유는 대부분의 경우에 종속변수들은 서로 독립적이 아니고, 동일한 개체에서 채택되어 상관관계가 있기 때문이다.

5.4.1 Dataset

K기업의 마케팅 담당자는 신제품의 전국적인 판매촉진활동을 계획하고 있다. 그는 가격(price)과 판매점(place)에 따른 신제품 판매량의 차이를 측정하고자 한다. 판매점마다 가격을 서로 다르게 하여 2주일 동안 판매량을 조사한 결과 다음과 같은 데이터를 수집했다. 유의수준 5%로 검정하라.

구분		판매점			
		1	2	3	4
가격	1(상)	(30, 34)	(40, 40)	(27, 26)	(25, 27)
		(34, 31)	(40, 45)	(26, 28)	(23, 25)
	2(중)	(36, 37)	(46, 44)	(27, 26)	(29, 27)
		(34, 34)	(48, 47)	(26, 29)	(26, 26)
	3(하)	(39, 36)	(46, 48)	(32, 34)	(30, 31)
		(39, 40)	(47, 53)	(32, 31)	(33, 31)

(주) 괄호 안의 첫 번째 숫자는 판매촉진 1주일 전의 판매량이고, 두 번째 숫자는 판매촉진 1주일 후의 판매량을 가리킨다.

5.4.2 가설의 설정

- 판매점에 따라 판매촉진 1주일 전의 판매량과 판매촉진 1주일 후의 판매량은 차이가 없을 것이다.
- 가격에 따라 판매촉진 1주일 전의 판매량과 판매촉진 1주일 후의 판매량은 차이가 없을 것이다.
- 판매점과 가격간의 상호작용이 없다.

5.4.3 R을 이용한 다변량 분산분석

```
> library(foreign) ## SPSS 파일을 불러오기 위한 옵션
> dat = read.spss("D:/R/MANOVA.sav")
> names(dat) <- c('place','price','score1','score2')
> attach(dat)
> place=factor(place)
> price=factor(price)
> Y <- cbind(score1, score2)
> fit <- manova(Y ~ place*price)
> summary.aov(fit)
Response score1 :
      Df Sum Sq Mean Sq F value    Pr(>F)
place   3 1105.46   368.49 180.4830 3.019e-10 ***
price   2  175.58    87.79  43.0000 3.371e-06 ***
place:price 6   31.42    5.24   2.5646 0.07783 .
Residuals 12   24.50    2.04
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Response score2 :
      Df Sum Sq Mean Sq F value    Pr(>F)
place   3 1268.83   422.94  84.5889 2.437e-08 ***
price   2  152.33    76.17  15.2333 0.0005091 ***
place:price 6   10.67    1.78   0.3556 0.8931994
Residuals 12   60.00    5.00
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

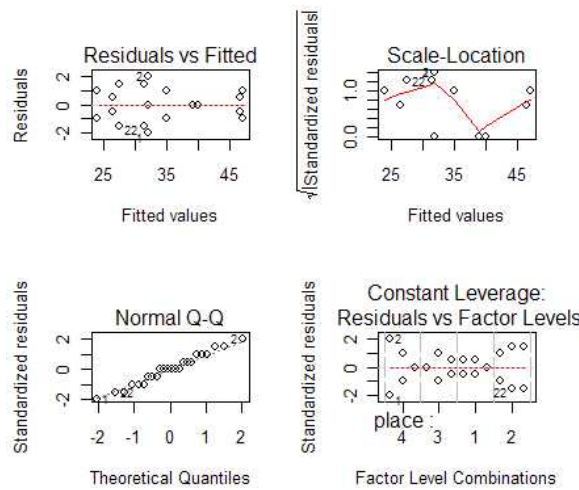
> summary(fit, test="Wilks")
      Df  Wilks approx F num Df den Df    Pr(>F)
place   3  0.0152  26.0761     6   22 6.135e-09 ***
price   2  0.0899  12.8425     4   22 1.533e-05 ***
place:price 6  0.3734   1.1667    12   22   0.3626
Residuals 12
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- 결과 해석 : 다변량 분산분석에서는 연구목적에 따라서 다양한 통계량을 사용한다. 여기서 Wilks 통계량값을 사용한다.

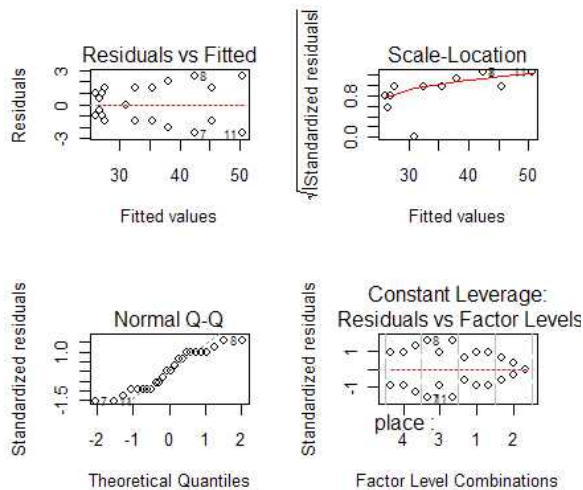
- 만일 람다 값이 작으면 귀무가설을 기각시킨다. 위의 결과 판매점의 람다 값은 0.015이며, 이것을 F값으로 환산하면 26.0761이 된다. 이때, 유의확률은 0.000이므로 평균 벡터가 같다는 귀무가설을 기각시킨다. 따라서 판매점에 따라 신제품 판매량은 차이가 있으며, 또한 가격에 따라서도 차이가 있음을 알 수 있다.
- 교호작용효과를 검정하기 위해 Wilks 람다 값의 F 확률을 살펴보면 0.3626으로 판매점 및 가격의 교호작용효과는 없다고 볼 수 있다.

<참고 사항>

- > ResLM1=lm(score1~place*price)
- > ResLM2=lm(score2~place*price)
- > layout(matrix(c(1,2,3,4),2,2))
- > plot(ResLM1)



- > plot(ResLM2)



제 6 장

상관관계분석

6.1 상관관계의 의미

6.1.1 상관계수

- 상관계수란?

두 변수가 얼마나 서로 관련을 맺고 움직이느냐를 수치화한 것이다.

- 상관계수의 성질

1) 상관계수는 $-1 < r < 1$ 사이의 값을 갖는다.

2) $r > 0$: 양의 상관관계 즉 x, y의 변화 방향이 같을 때.

$r < 0$: 음의 상관관계 즉 x, y의 변화 방향이 반대일대.

3) 상관계수는 r의 절대값 크기에 따라 상관관계의 강도를 알 수 있다. 또한 $r=0$ 일 때, 두 변수의 관계가 전혀 없다는 예기이다.

4) 상관계수를 확인하기 전에 산포도를 통해 시각적으로 그 변수간의 상관관계를 확인하는 것이 중요하고 필요하다.

5) 상관계수 r은 두변수의 표준점수로 나타내어지기 때문에 측정단위에 영향을 받지 않으며 x, y의 값이 서로 바뀌어도 그 값은 같다.

6) 주의) 상관계수는 절대적이지 않다. 이는 어떤 현상이든 항상 교락 인자가 존재하기 때문이다.

실험변수, 즉 제3의 외생변수라 생각하면 되는데, 실제로 종속변수 독립변수 보다 더 직접적으로 영향을 미칠 수 있으며 또한 이상치가 있을 때에는 상관계수 산출에 또한 큰 영향을 미친다.

6.1.2 상관계수의 종류

- 상관계수의 종류는 크게 등간척도 이상으로 측정된 두 변수들간의 상관관계를 측정하는데 사용되는 피어슨 상관계수(Pearson correlation)와 서열척도로 측정된 두 변수들간의 상관관계를 측정하는데 사용되는 스피어만 상관계수(Spearman correlation)로 구분된다. 일반적으로 상관계수라 할 때는 피어슨 상관계수를 지칭한다. 이외에 분산분석에서 공변량(covariate)에 해당되는 외생변수의 효과를 통제한 후 순수하게 두 변수간의 관계를 살펴보는 편상관계수(partial correlation)가 이용되기도 한다.

6.2 피어슨 상관계수

$$- \text{정의} : r = \frac{S_{XY}}{S_X S_Y} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{n-1} \div \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}} \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{n-1}}$$

S_X : X의 표준편차

S_Y : Y의 표준편차

S_{XY} : 두 변수 X, Y의 공분산

- 의미 : 두 변수 X와 Y 사이의 선형적 상관관계를 파악할 수 있다. Pearson의 상관계수는 -1에서 +1 사이의 실수 값을 취하며, 0을 취할 때에는 두 변수는 선형적으로 아무런 관계가 없음(독립)을 의미한다.

- 성질 : 상관계수는 측정 단위에 의존하지 않는다. 예를 들면, 파운드로 기록된 몸무게와 인치로 기록된 Y변수의 상관계수는 킬로그램으로 측정된 X변수와 센티미터로 측정된 Y변수의 상관계수는 같다.

6.2.1 Dataset

- 개인별 산소 소모에 미치는 변수들의 효과를 보기 위해 1.5마일 주행시간(RUNTIME), 연령(age), 몸무게(weight), 주행 중 맥박수(runpulse), 주행 중 최대 맥박수(maxpulse), 휴식 중 최대 맥박수(rstpulse)로 잡았다. 이 변수들 간의 Pearson Correlation을 구하라.

age weight oxy runtime rstpulse runpulse maxpulse													
44	89.47	44.609	11.37	62	178	182	49	81.42	49.156	8.95	44	180	185
40	75.07	45.313	10.07	62	185	185	51	69.63	40.836	10.95	57	168	172
44	85.84	54.297	8.65	45	156	168	51	77.91	46.672	10.00	48	162	168
42	68.15	59.571	8.17	40	166	172	48	91.63	46.774	10.25	48	162	164
38	89.02	49.874	9.22	55	178	180	49	73.37	50.388	10.08	67	168	168
47	77.45	44.811	11.63	58	176	176	57	73.37	39.407	12.63	58	174	176
40	75.98	45.681	11.95	70	176	180	54	79.38	46.080	11.17	62	156	165
43	81.19	49.091	10.85	64	162	170	52	76.32	45.441	9.63	48	164	166
44	81.42	39.442	13.08	63	174	176	50	70.87	54.625	8.92	48	146	155
38	81.87	60.055	8.63	48	170	186	51	67.25	45.118	11.08	48	172	172
44	73.03	50.541	10.13	45	168	168	54	91.63	39.203	12.88	44	168	172
45	87.66	37.388	14.03	56	186	192	51	73.71	45.790	10.47	59	186	188
45	66.45	44.754	11.12	51	176	176	57	59.08	50.545	9.93	49	148	155
47	79.15	47.273	10.60	47	162	164	49	76.32	48.673	9.40	56	186	188
54	83.12	51.855	10.33	50	166	170	48	61.24	47.920	11.50	52	170	176
							52	82.78	47.467	10.50	53	170	172

6.2.2 R을 이용한 통계검정

```

> data <- read.table("D:/R/correlation.txt", header=T)
> attach(data)
> cor(data)
      age      weight      oxy      runtime      rstpulse      runpulse
age      1.0000000 -0.23353903 -0.3045924  0.1887453 -0.16409995 -0.3378703
weight  -0.2335390  1.00000000 -0.1627528  0.1435076  0.04397417  0.1815163
oxy      -0.3045924 -0.16275285  1.0000000 -0.8621949 -0.39935611 -0.3979742
runtime  0.1887453  0.14350758 -0.8621949  1.0000000  0.45038260  0.3136478
rstpulse -0.1640999  0.04397417 -0.3993561  0.4503826  1.00000000  0.3524606
runpulse -0.3378703  0.18151633 -0.3979742  0.3136478  0.35246060  1.0000000
maxpulse -0.4329159  0.24938123 -0.2367402  0.2261030  0.30512400  0.9297538
      maxpulse
age      -0.4329159
weight   0.2493812
oxy      -0.2367402
runtime  0.2261030
rstpulse 0.3051240
runpulse 0.9297538
maxpulse 1.0000000

```

- 위와 같이 cor() 함수를 이용하면 각각의 상관계수가 출력된다. 하지만 t-검정을 통한 유의수준 p-value 값이 제시가 되지 않아 유의한지에 대하여 알 수가 없다.
- cor.test(x,y) 를 이용하기 위해선 변수의 수가 2개 이상일 경우는 분석을 실시 할 수 없다는 단점이 있다.
- 따라서, 보다 정확한 검정을 하기 위해서는 agricolae 라는 패키지를 이용한다.
- 먼저, 패키지 → 패키지 인스톨... 을 선택한 후 CRAN mirror를 지정해 준다.
- 다음으로, 설치할 패키지인 agricolae 를 선택하여 설치해 준다.

```

> library(agricolae)
> data <- read.table("D:/R/correlation.txt", header=T)
> analysis<-correlation(data[,1:7],method="pearson")
> analysis
$correlation

$pvalue

$n.obs
[1] 31

```

- 결과 해석 : oxy와 runtime 상관계수는 -0.86이며 p-value=0.00001로 매우 작은 값이므로 산소 흡입량과 1.5마일 달리는 데 걸리는 시간과 상당히 유의한 것으로 결론을 내릴 수 있다. 서로 음의 상관관계로서 빨리 달리는 사람이 산소 흡입량이 많음을 밝히고 있다.

6.3 스피어만 상관계수

- 스피어만 상관계수는 데이터가 서열척도인 경우 즉 자료의 값 대신 순위를 이용하는 경우의 상관계수로서, 데이터를 작은 것부터 차례로 순위를 매겨 서열 순서로 바꾼 뒤 순위를 이용해 상관계수를 구한다. 두 변수 간의 연관 관계가 있는지 없는지를 밝혀주며 자료에 이상점이 있거나 표본크기가 작을 때 유용하다. 스피어만 상관계수는 -1과 1 사이의 값을 가지는데 두 변수간의 순위가 완전히 일치하면 +1이고, 두 변수의 순위가 완전히 반대이면 -1이 된다. 예를 들어 수학 잘하는 학생이 영어를 잘하는 것과 상관있는지 없는지를 알아보는데 이용할 수 있다.

6.3.1 스피어만 상관계수 계산공식

- 스피어만 상관계수의 계산은 각 응답자(또는 표본)별로 두 변수값의 차이를 새로운 변수 (d_i)로 만든 후 다음과 같은 식을 이용하여 계산한다.

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad \text{여기서 } n \text{은 표본의 수를 의미함.}$$

6.3.2 R을 이용한 통계검정

```
> library(agricolae)
> data <- read.table("D:/R/correlation.txt", header=T)
> analysis<-correlation(data[,1:7],method="spearman")
> analysis
```

- 방법은 피어슨 상관분석과 동일하다.

※ 참고

```
- correlation between age, variable 2 and other elements from data.
data(data)
attach(data)
analysis<-correlation(age,data[,2:7],method="pearson",alternative="less")
analysis
detach(data)
```

6.4 편상관계수

- 편상관계수란 변수의 상관관계를 분석을 할 때, 다른 변수의 효과를 고정시키고 분석하고자 하는 두 변수 사이의 순수한 상관관계를 구하고자 할 때 사용되는 기법이다.

- 외생변수가 두 변수간의 상관관계에 영향을 미칠 것으로 예상되는 경우 외생변수의 효과를 제거한 후 두 변수간의 순수한 상관관계를 계산한 것이 편상관계수(partial correlation coefficient)이다. 편상관계수를 적용하기 위해서는 수집된 자료가 피어슨 상관계수와 마찬가지로 등간척도 이상이어야 하며 모집단 분포에 대해 정규분포를 가정한다.

6.4.1 편상관계수 계산공식

- 변수 x1을 고정시키고 y와 x2의 편상관계수를 구하는 식은 다음과 같다.

$$r_{yx_2, x_1} = \frac{r_{yx_2} - r_{x_1x_2}r_{yx_1}}{\sqrt{1 - r_{x_1x_2}^2} \sqrt{1 - r_{yx_1}^2}}$$

- 변수 x2을 고정시키고 y와 x1의 편상관계수를 구하는 식은 다음과 같다.

$$r_{yx_1, x_2} = \frac{r_{yx_1} - r_{x_1x_2}r_{yx_2}}{\sqrt{1 - r_{x_1x_2}^2} \sqrt{1 - r_{yx_2}^2}}$$

6.4.2 R을 이용한 통계검정

```
> library(agricolae)
> data <- read.table("D:/R/correlation.txt", header=T)
> attach(data)
> a = lm(oxy ~ rstpulse)    ## 회귀분석을 실시하고 잔차를 구한다.
> b = lm(runtime ~ rstpulse)
> r1 = c(a$residuals)      ## 잔차의 값을 새로운 변수에 넣는다.
> r2 = c(b$residuals)
> dat <- cbind(r1,r2)
> detach(data)
```

- 순수한 상관관계를 보고자 하는 개념이기 때문에 회귀 분석을 이용하여 회귀식을 실행한 다음, 서로의 잔차들의 상관계수를 구하면 편상관계수가 된다.

```

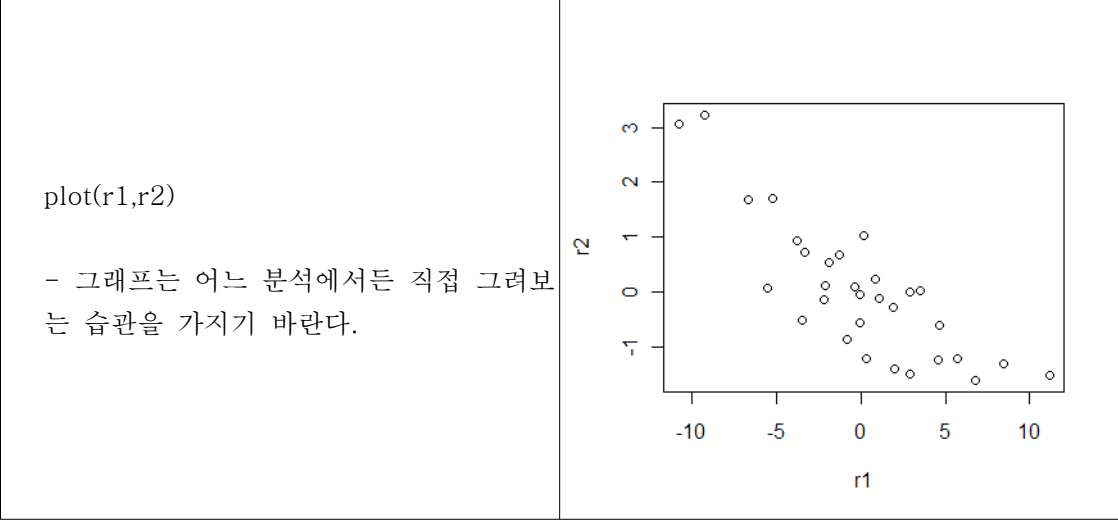
> analysis<-correlation(r1,r2,method="pearson")

Pearson's product-moment correlation

data: r1 and r2
t = -8.126571 , df = 29 , p-value = 5.822853e-09
alternative hypothesis: true rho is not equal to 0
sample estimates:
cor
-0.833588

```

- 결과 해석 : 앞서 설명한 oxy와 runtime 상관계수는 -0.86을 나타냈지만, 휴식 중 최대 맥박수(rstpulse)를 통제한 결과 편상관계수는 -0.833588로 낮아짐을 알 수 있다. 이는 통제하기 전과 마찬가지로 p-value=0.00001로 매우 작은 값이므로 산호 흡입량과 1.5마일 달리는 데 걸리는 시간과 상당히 유의한 것으로 결론을 내릴 수 있다. 또한 서로 음의 상관관계로서 빨리 달리는 사람이 산소 흡입량이 많음을 밝히고 있다. 하지만 통제를 한다면 통제하지 않은 사람보다 산소 흡입량이 작아짐을 알 수 있다.



제 7 장

회귀분석

7.1 회귀분석의 의미

- 회귀분석은 변수들 간의 함수관계를 분석하고 모형화하는 통계적 기법이다.
- 회귀분석의 응용분야는 공학, 자연과학, 경제학, 경영학, 생명과학, 사회과학 등 여러 분야에 적용되고 있으며, 최근에는 컴퓨터 통계 소프트웨어(SAS, SPSS, MINITAB, BMDP 등)의 활용으로 변수들 사이의 복잡한 함수관계 추정하는데 가장 널리 사용되어지는 자료분석 기법이다.

Ex) 어느 회사 제품의 매출액이 광고비 지출액에 따라 변동한다면, 이들 두 변수사이의 함수관계를 추정하여 매출액을 추정할 수 있고, 광고비가 매출액에 미치는 효과를 분석할 수 있을 것이다. 또는 어느 부품조립 생산라인에서 제품의 불량률이 생산라인의 속도와 어떤 관련성이 있다면, 이들 두 변수들 간의 함수관계를 분석함으로써 불량률을 가능한 줄이면서 양질의 제품을 생산할 수 있는 라인속도를 추정할 수 있을 것이다.

- 위의 두 가지 예에서 다른 변수로부터 추정 또는 예측되어야 하는 변수들인 판매량 불량률은 일반적으로 종속변수(dependent variable) 혹은 반응변수(response variable)라 부르고 기호 Y로 표기하고, 광고비 와 생산라인 속도와 같이 종속변수에 영향을 미치는 변수들을 독립변수(independent variable) 혹은 설명변수(explanatory variable)라 하고 X로 표기한다.

- 일반적으로 종속변수에 영향을 주는 독립변수 수는 여러 개가 있을 수 있기 때문에 회귀분석은 하나의 종속변수와 여러 개의 독립변수들 간의 통계적 함수관계를 분석하여 모형화하는데 이용되고 있다.

- 회귀분석을 사용하는 목적

첫째, 종속변수와 독립변수들 사이의 함수관계가 어떠한 형태(선형 또는 비선형)를 가지고 있는지를 파악하는 것이다.

둘째, 종속변수에 영향을 미치는 중요한 독립변수들의 영향을 추정, 검정하는 것이다.

셋째, 추정된 회귀함수를 인용하여 주어진 독립변수의 값에서 종속변수의 평균변화를 추정 혹은 예측하는 것이다.

7.2 단순회귀분석

7.2.1. 단순선형회귀분석의 목적

- 단순선형회귀분석이란 종속변수 y 와 x 간의 어떤 선형적 관계가 있는지를 알아보고 식으로 표현하며, 나아가서는 특정 x 값에 대한 y 값을 알아낼 수 있도록 하는 관계식을 설정함을 목표로 합니다.

7.2.2 단순선형회귀분석의 기본가정

- 이를 위해서는 몇 가지 조건이 필요하다. 보통 단순회귀분석에서 x 와 y 의 관계는 아래와 같이 표현한다.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

즉, y_i 는 y 축의 절편에 해당하는 값, x_i 의 계수, x_i , 그리고 오차항으로 이루어진다.

- 회귀분석에서는 오차항의 독립성과 등분산성, 정규성을 가정한다.

이를 식으로 표현하면 아래와 같다.

$$\epsilon_i \text{ iid} \sim N(0,1)$$

iid 는 identically and independently distributed 즉, 분산이 같고 독립적임을 뜻한다.

7.2.3 회귀계수의 추정식

- 보통 단순선형회귀분석에서 회귀계수를 추정하는 방법은 LSE(Least Square Method)이다. 즉, 오차의 제곱이 최소가 되는 계수를 추정하는 방법이다.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \text{ 이므로, } \epsilon_i = y_i - \beta_0 - \beta_1 x_i \text{ 로 표현 가능하다.}$$

- LSE 방법은 $\sum \epsilon_i^2$ 를 최소화하는 β_0, β_1 의 값을 찾는 것이 목적이므로, 아래의 식을 각 계수 값에 관하여 편미분하고 0으로 놓은 뒤, 연립방정식을 풀면 된다.

7.2.4 제곱합의 증명과정

- y 의 각 관측값과 추정된 회귀직선과의 차이의 제곱합을 SSTO (sum of squares total)라고 하면, 이는 SSR (sum of squares regression)과 SSE(sum of squares error)로 나누어 짐을 밝힐 수 있다. 이를 각각의 자유도로 나눈 MSR과 MSE의 비를 이용한 F 검정을 통해 회귀분석의 모형이 유의한지를 알 수 있다.

7.2.5 회귀계수의 유의성 검정에 대한 제증명

- 추정된 회귀계수 b_1 의 분산은 다음과 같다.

$$Var(b_1) = Var\left(\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}\right) = \frac{Var(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = \frac{\sigma^2}{\sum(x_i - \bar{x})^2}$$

- 오차항이 정규분포를 한다는 가정 하에, y_i 도 정규분포 가정이 성립하고, b_1 이 y_i 의 선형결합이므로, b_1 도 정규분포를 따르고 그 기댓값은 β_1 , 분산은 위 식과 같다. 식으로 표현하면 아래와 같다.

$$\therefore b_1 \sim N\left(\beta_1, \frac{\sigma^2}{\sum(x_i - \bar{x})^2}\right)$$

그런데, 여기서 σ^2 은 모수이므로 불편추정량인 s^2 을 사용한다.

따라서, b_1 은 t분포를 따르게 되고, 이를 표준화한 t통계량을 식으로 표현하면 아래와 같다.

$$t = \frac{\hat{\beta}_1}{\frac{S}{\sqrt{\sum(x_i - \bar{x})^2}}}$$

$\hat{\beta}_1$: 추정된 회귀계수

S : 오차항의 표본표준오차

7.2.6 Dataset

- 다음은 변수 X, Y에 대한 6개의 데이터이다. 이 6개의 데이터 X, Y의 관계를 가장 잘 설명하는 식을 구해보자.

X	1	2	3	4	6	7
Y	21	32	43	56	67	76

7.2.7 R을 이용한 통계분석

```

> # 단순 선형회귀모형
> x = c(1,2,3,4,6,7)
> y = c(21,32,43,56,67,76)
> out = lm(y ~ x)           # 단순선형회귀
> summary(out)             # 기본 분석결과 확인

Call:
lm(formula = y ~ x)

Residuals:
    1     2     3     4     5     6 
-2.7019 -0.6894  1.3230  5.3354 -1.6398 -1.6273

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.7143     2.7816   5.29 0.006129 **
x             8.9876     0.6354  14.15 0.000145 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.291 on 4 degrees of freedom
Multiple R-squared:  0.9804,    Adjusted R-squared:  0.9755
F-statistic: 200.1 on 1 and 4 DF,  p-value: 0.000145

```

```

> jpeg('7_2_6.jpg')
> plot(x, y)           # plotting
> abline(out)         # 회귀선 추가

```

- 결과 해석 : X, Y는 $Y = 14.7143 + 8.9876 * X$ 의 관계를 가진다. 이들 추정된 계수는 t값이 5.29, 14.15로 의미 있는 것으로 나타났다. 이렇게 구한 회귀방정식이 설명하는 정도(R^2)는 R-squared: 0.9804로 98.04%이다. 그리고 모형의 적합도를 나타내는 F값 또한 F-statistic: 200.1 이며 그 확률은 p-value: 0.000145로서 통상적인 유의수준 0.05보다 훨씬 작기 때문에 모형이 상당히 의미 있는 것으로 나타났다.

7.3 다중회귀분석

7.3.1 다중회귀분석의 의미

- 다중회귀분석은 일반적으로 두 변수 이상의 독립변수(영향변수, 원인변수)들이 종속변수(결과변수)에 어떠한 영향을 미치는 가를 알기 위한 분석기법이다. 예를 들면, 골프용품에 대한 전체만족도에 골프용품의 타구감, 방향성, 명성 등의 변수들이 어떠한 영향을 미치는 가를 알기 위해서는 다중회귀분석을 하여야 한다. 따라서 독립변수들이 종속변수에 미치는 상대적 영향력을 알아 볼 수 있으며, 이러한 독립변수 값들의 변화에 따라 종속변수 값이 어떻게 변화하는가를 예측할 수 있다.

- 독립변수와 종속변수에 쓰인 변수들의 측정척도는 등간척도(interval scale)나 비율척도(ratio scale)의 메트릭(metric) 자료이어야 하나, 독립변수가 명목척도일 경우에는 0과 1을 사용한 더미변수(dummy variables)를 이용하여 분석할 수 있다.

- 다중회귀분석은 독립변수를 동시에 모두 투입하는 방식(ENTER), 중요한 변수(설명력이 높은 변수, 통계적으로 유의도가 높은 변수) 순으로 투입되다가 통계적으로 유의성이 없는 변수들만 남게 되면 분석이 중단되는 방식(STEPWISE), 그리고 모든 독립변수를 모두 투입하여 회귀식을 도출하고 나서 중요도가 낮은 변수들을 순차적으로 제거하면서 회귀식을 도출하는 방식(BACKWARD) 등이 있다.

- 다중회귀분석은 회귀분석 중 하나로 하나의 독립변수와 종속변수와의 관계를 알아보기 위한 단순회귀분석과는 구분된다. 일반적으로 다중회귀분석의 회귀식은 다음과 같다.

$$Y = a + \beta_1 X_1 \dots\dots\dots \beta_n X_n + \varepsilon$$

7.3.2 다중공선성

- 다중공선성이란 무엇인가?

- 1) 다중공선성(Multicollinearity)이란 독립변수들 사이에 선형관계가 있는 경우를 의미함
- 2) 한 독립변수의 추정계수가 다른 독립변수의 추정계수에 상당한 영향을 미치게 됨

- 어떤 문제점이 발생하는가?

- 1) 다중공선성이 있는 하나의 변수를 제외시키면 다른 독립변수의 계수가 급격하게 변하게 되는 문제점이 발생함

- 2) 추정된 회귀계수의 분산이 크기 때문에 회귀계수의 추정치의 신뢰도가 떨어짐
 - 모형의 신뢰도를 나타내는 F값은 유의함
 - 문제는 계수의 검정치인 t값이 의미가 없게 되는 경우가 발생함
- 3) 서로 상관이 있는 독립변수들이 공존하므로 어떤 계수는 양의 값을 가져야 될 것 같은데 음의 값을 가지게 되는 경우가 발생함

※ 이와 같이 다중공선성이 있는 모형은 상당히 불안정하다.

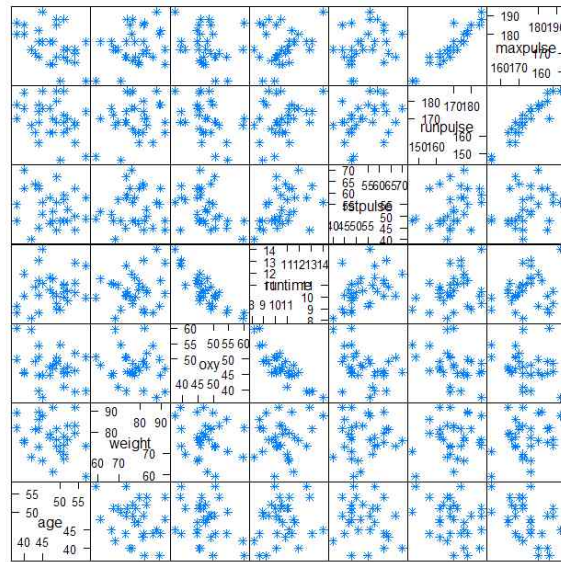
- 해결방안은 무엇인가?
 - 1) 독립변수를 제거한다.
 - 2) 새로운 관측치를 추가한다.
 - 3) 주성분분석, 요인분석을 통하여 공선성이 발생하지 않는 변수변환을 시킨다.
 - 4) Partial Least Square 분석을 수행한다.
- 다중공선성은 어떻게 감지하나?
 - 1) 모형과 각 계수의 통계적 유의성 관찰
 - 2) 분산확대지수(VIF, Variance Inflation Factors)
 - VIF의 계산은 $1/(1-R_i^2)$ 으로 계산함
 - R_i^2 : i번째 독립변수를 종속변수로 정의하고 나머지 독립변수들을 독립변수로 한 모형의 R^2 값이다.
 - VIF의 값이 클수록 다중공선성이 있다는 것을 의심해 봐야 한다.
 - 3) 행렬 $X'X$ 의 구조분석
 - $X'X$ 벡터의 고유치와 고유벡터를 분석하여 고유치의 크기가 0이면 완전한 공선성인 것을 나타내므로 0에 가까운 값은 공선성이 높다는 것을 의미한다.

7.3.3 Dataset

- 개인별 산소 소모에 미치는 변수들의 효과를 보기 위해 1.5마일 주행시간(RUNTIME), 연령(age), 몸무게(weight), 주행 중 맥박수(runpulse), 주행 중 최대 맥박수(maxpulse), 휴식 중 최대 맥박수(rstpulse)로 잡았다. 이 자료는 상관관계분석에서 6.2.1 Dataset으로 사용한 바가 있으니 참고하길 바란다.

7.3.4 R을 이용한 통계분석

```
> library(lattice)
> data <- read.table("D:/R/correlation.txt", header=T)
> splom(~data, pch = 8)      # scatter plot matrix
```



Scatter Plot Matrix

```
> attach(data)
> prod.lm = lm(oxy ~ runtime+ age+ weight+ runpulse+ maxpulse+ rstpulse)
> summary(prod.lm)
```

Call:

```
lm(formula = oxy ~ runtime+ age+ weight+ runpulse+ maxpulse+ rstpulse)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.40256	-0.89908	0.07063	1.04964	5.38469

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	102.93448	12.40326	8.299	1.64e-08 ***
runtime	-2.62865	0.38456	-6.835	4.54e-07 ***
age	-0.22697	0.09984	-2.273	0.03224 *
weight	-0.07418	0.05459	-1.359	0.18687
runpulse	-0.36963	0.11985	-3.084	0.00508 **
maxpulse	0.30322	0.13650	2.221	0.03601 *
rstpulse	-0.02153	0.06605	-0.326	0.74725

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.317 on 24 degrees of freedom
 Multiple R-squared: 0.8487, Adjusted R-squared: 0.8108
 F-statistic: 22.43 on 6 and 24 DF, p-value: 9.715e-09

- 결과 해석 : 먼저 6장 상관관계분석에서의 결과를 살펴보기 바란다. 변수 RUNPULSE와 MAXPULSE의 상관계수는 0.92975로 두 변수가 선형관계에 있다는 것을 확인할 수 있을 것이다. 이 경우 한 독립변수의 추정값이 다른 독립변수의 추정값에 상당한 영향을 미친다. 따라서, 두 회귀계수의 추정치는 불안정하게 된다.

- 다중회귀분석의 결과를 보면, 회귀방정식은 :

$$\text{OXY} = 102.93448 - 2.62865 \cdot \text{runtime} - 0.22697 \cdot \text{age} - 0.07418 \cdot \text{weight} - 0.36963 \cdot \text{runpulse} + 0.30322 \cdot \text{maxpulse} - 0.02153 \cdot \text{rstpulse} \text{ 이다.}$$

- 모형은 R^2 값이 84.87%, 모형의 유의확률이 0.0001로 매우 유의한 것으로 나타났다. 회귀계수의 추정값을 검정해 보면 weight와 rstpulse 변수의 유의확률이 각각 0.18687, 0.74725로 유의하지 않다.

7.3.4 R을 이용한 통계분석 (다중공선성)

```

> library(lattice)
> data <- read.table("D:/R/correlation.txt", header=T)
> attach(data)
> prod.lm = lm(oxy ~ runtime+ age+ weight+ runpulse+ maxpulse+ rstpulse)
> summary(prod.lm)
>
> # To get VIF's, first copy this function (by Bill Venables) into R:
> #####
> vif <- function(object, ...)
+ UseMethod("vif")
>
> vif.default <- function(object, ...)
+ stop("No default method for vif. Sorry.")
>
> vif.lm <- function(object, ...) {
+   V <- summary(object)$cov.unscaled
+   Vi <- crossprod(model.matrix(object))
+   nam <- names(coef(object))
+   if(k <- match("(Intercept)", nam, nomatch = F)) {
+     v1 <- diag(V)[-k]
+     v2 <- (diag(Vi)[-k] - Vi[k, -k]^2/Vi[k,k])
+     nam <- nam[-k]
+   } else {
+     v1 <- diag(V)
+     v2 <- diag(Vi)
+     warning("No intercept term detected. Results may surprise.")
+   }
+   structure(v1*v2, names = nam)
+ }
> #####
> vif(prod.lm)
runtime      age      weight runpulse maxpulse rstpulse
1.590868 1.512836 1.155329 8.437274 8.743848 1.415589

```

- 결과 해석 : 앞의 분석에서 Multiple R-squared: 0.8487이고 $1/(1-R_i^2)=6.609$ 이므로 이 값보다 큰 변수 runpulse와 maxpulse는 다중공선성이 있다는 것을 알 수 있다.

- 혼회 변수의 공차한계값은 0.9를 넘고 VIF값은 1.0에 근접하면, 두 변수간에 상관관계는 낮으며, 추정된 회귀계수는 다중공선성의 영향을 받지 않는 것으로 결론을 내린다.

7.4 더미 변수를 이용한 회귀분석

- 회귀분석의 입력 자료는 대개의 경우 등간척도, 비율척도로 구성된다. 그러나 경우에 따라 명목척도로 측정된 변수를 회귀분석의 독립변수로 하여 분석할 필요가 있는데 이러한 변수를 더미변수라고 한다.

$$\text{더미변수의 수} = \text{범주의 수} - 1$$

- 만약 범주의 수가 두 개인 경우 더미변수의 수는 한 개이며, 한 범주를 1로 다른 범주를 0으로 입력한다. 범주가 세 개인 경우에는 더미변수는 2개이며, 입력방식은 다음과 같다.

범 주	더미변수1	더미변수2
A	0	0
B	1	0
C	0	1

7.4.1 Dataset

- 소비자의 직업이 아이스크림의 구매빈도에 미치는 영향을 알고 싶어 회귀분석을 실시한다. 소비자의 직업을 중고교생, 대학(원)생, 주부, 직장인의 4개 유형으로 분류하였기 때문에 세 개의 더미변수를 필요로 한다.

	D1	D2	D3
중고교생(기준점)	0	0	0
대학(원)생	1	0	0
주부	0	1	0
직장인	0	0	1

- SPSS 'ice.sav' 파일을 참고하기 바란다.

7.4.2 R을 이용한 통계분석

```
> library(foreign) ## SPSS 파일을 불러오기 위한 옵션
> dat = read.spss("D:/R/ice.sav")
> names(dat) <- c('id','a4','d1','d2','d3','d7')
> attach(dat)
>
> df <- data.frame(a4,d3,d7)
> # Create dummy vectors
> df$dd1 <- ifelse(df$d3==2, 1, 0)
> df$dd2 <- ifelse(df$d3==4, 1, 0)
> df$dd3 <- ifelse(df$d3==3, 1, 0)
> detach(dat)
> attach(df)
> prod.lm = lm(a4 ~ dd1 + dd2 + dd3)
> summary(prod.lm)
```

Call:

```
lm(formula = a4 ~ dd1 + dd2 + dd3)
```

Residuals:

Min	1Q	Median	3Q	Max
-5.82	-3.65	-2.36	1.64	53.50

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.6500	0.8894	7.477	7.01e-13 ***
dd1	-0.1469	1.0421	-0.141	0.888
dd2	-0.2900	1.3192	-0.220	0.826
dd3	0.1697	1.2527	0.135	0.892

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.889 on 328 degrees of freedom

Multiple R-squared: 0.0004563, Adjusted R-squared: -0.008686

F-statistic: 0.04991 on 3 and 328 DF, p-value: 0.9852

- 결과 해석 : 전환된 더미변수를 독립변수로 이용한 다중회귀분석을 실시했을 경우 모두 유의하지 않은 것으로 나타났다. 이는 중고교생과 비교했을 때 대학(원)생, 주부, 직자인 모두 아이스크림 구매빈도에 큰 차이가 나지 않음을 의미한다.

7.5 정준상관분석

- 개념 : 몇 개 변수들이 집단으로 관측된 두 집단사이의 연관성(상관성)을 보거나, 두 변수 집단이 서로 독립적인지의 여부를 분석하는 통계분석기법이다.

- Ex) 학생들의 신체조건과 운동능력 사이의 연관성을 보기 위하여 학생 100명을 대상으로 신체조건(X1=키, X2=앉은 키, X3=100m달리기, Y2=넓이 뛰기, Y3=높이 뛰기)

- 기능 : 크게 두 가지의 기능.

*두 변인군간의 관계와 관계 정도의 파악

예를 들어, 어떤 신용카드 회사가 가족의 수와 총소득(독립변수)으로 가족이 보유한 신용카드의 수 및 월 평균 신용카드 지출액(종속변수)을 동시에 예측하려 한다면 정준상관분석을 이용할 수 있다. 이것은 단일한 변수들간의 상관관계가 아니라 여러 개의 독립변수들과 여러 개의 종속변수들 간의 관계가 되며, 상관관계분석이나 회귀분석 등의 방법으로는 문제를 해결할 수 없다.

*독립변수와 종속변수군간의 상관관계가 최대가 되게 하는 각 변수들의 상대적 기여도 파악 위의 예에서 정준상관분석을 사용하면 가족수 와 총소득 중 어느 것이 신용카드의 수 및 지출액 중 어느 변수를 어느 정도 예측할 수 있는지 알 수 있다.

- 주의사항

① 일반적인 상관성 분석과 마찬가지로 정준상관분석은 인과관계의 방향성을 밝히지는 않는다.

② 분석을 위해서는 먼저 여러 변인들을 종속변수군과 독립변수군으로 분류시켜야 한다.

③ 모든 변수들을 등간척도나 비율척도로 측정해야만 한다. 단, 더미변수를 사용하면 명목척도로 된 자료를 가지고서도 분석이 가능하다.

- 정준상관분석 모형

위 예에서 3가지 신체조건과 3가지 운동능력에 있어서 변수 집단 X와 Y를 각각

$$X=(X_1, X_2, X_3), Y=(Y_1, Y_2, Y_3)$$

두 개의 벡터 a와 b를

$$a=(a_1, a_2, a_3), b=(b_1, b_2, b_3)$$

와 같이 정의한 후에 Z1 과 Z2를

$$Z_1 = aX = a_1X_1 + a_2X_2 + a_3X_3$$

$$Z_2 = bY = b_1Y_1 + b_2Y_2 + b_3Y_3$$

와 같이 구한다. 이와 같이 변수 Z1과 Z2를 각각 X와 Y의 선형 결합으로 구한 후에 변수 Z1과 Z2의 상관계수를 정준상관계수라 한다.

7.5.1 R을 이용한 통계분석

- 본 분석에서는 하야시 박사에 의하여 개발된 직적 자료의 수량화에 관한 이론 및 방법론인 수량화방법론 I, II, III, IV의 연습문제를 풀이해보도록 하자.

- 수량화방법 II

연습문제 1. 다음은 일본 T대학 여학생 40명을 상대로 한 앙케이트 조사결과 이다.

외적기준인 혈액형(BLD)과 설명변량인 성격(V1-V12)과의 관계에 대하여 수량화 방법2에 의한 통계분석을 하여라. 이때 제1 정준변량과 제2 정준변량의 공간에 자료를 플롯하고 적절한 해석을 하여라.

< 변수 설명 >

ID = 개인번호	V1 = 합리적인가	V2 = 기분파인가
V3 = 타인을 의식하는 편인가		V4 = 꼭 맞는 것을 좋아 하는가
V5 = 조정반원인가	V6 = 개성적인가	V7 = 현실파인가
V8 = 보수적인가	V9 = 낭만파인가	V10 = 낙천가인가
V11 = 견실파인가	V12 = 중간의식	BLD = 혈액형

자료값 설명 : 1="예", 2="아니오"

```
> library(foreign) ## SPSS 파일을 불러오기 위한 옵션
> dat = read.spss("D:/R/BLD.sav")
> names(dat) <- c('id','v1','v2','v3','v4','v5','v6','v7'
+               , 'v8','v9','v10','v11','v12','BLD','BLD_1')
> dat$v1_1 <- ifelse(dat$v1==1, 1, 0)
> dat$v1_2 <- ifelse(dat$v2==1, 1, 0)
> dat$v1_3 <- ifelse(dat$v3==1, 1, 0)
> dat$v1_4 <- ifelse(dat$v4==1, 1, 0)
> dat$v1_5 <- ifelse(dat$v5==1, 1, 0)
> dat$v1_6 <- ifelse(dat$v6==1, 1, 0)
> dat$v1_7 <- ifelse(dat$v7==1, 1, 0)
> dat$v1_8 <- ifelse(dat$v8==1, 1, 0)
> dat$v1_9 <- ifelse(dat$v9==1, 1, 0)
> dat$v1_10 <- ifelse(dat$v10==1, 1, 0)
> dat$v1_11 <- ifelse(dat$v11==1, 1, 0)
> dat$v1_12 <- ifelse(dat$v12==1, 1, 0)
> dat$intbld1 <- ifelse(dat$BLD_1=='A', 1, 0)
> dat$intbld2 <- ifelse(dat$BLD_1=='B', 1, 0)
> dat$intbld3 <- ifelse(dat$BLD_1=='O', 1, 0)
```

```

> attach(dat)
> x <- data.frame(intbld1,intbld2,intbld3)
> y <- data.frame(v1_1,v1_2,v1_3,v1_4,v1_5,v1_6,v1_7,
+               v1_8,v1_9,v1_10,v1_11,v1_12)
> library(CCA)  ## 정준상관분석을 하기 위한 패키지
> ex1.cancor <- cc(x,y)
> summary(ex1.cancor)
      Length Class  Mode
cor      3    -none- numeric
names    3    -none- list
xcoef    9    -none- numeric
ycoef   36    -none- numeric
scores   6    -none- list
> ex1.cancor$cor
[1] 0.6040876 0.5012448 0.4086025
> ex1.cancor$xcoef
      [,1]      [,2]      [,3]
intbld1 1.6903954 -2.4121157 1.873969
intbld2 0.9927978 -0.5039874 3.658599
intbld3 -0.6122255 -2.5439235 2.480652
> ex1.cancor$ycoef
      [,1]      [,2]      [,3]
v1_1  0.315297751 1.53289746 -0.85152661
v1_2  0.855461212 -0.18862998 -0.70060325
v1_3 -0.067723460 1.15352484 0.14756448
v1_4  0.701465592 -0.72452032 -0.80535567
v1_5  0.110160495 0.94215721 -0.38788935
v1_6 -0.438474595 -0.21783762 -0.52134846
v1_7 -1.570903319 -0.87482711 0.06519743
v1_8 -0.092166052 -0.19419566 0.05211813
v1_9  0.006579956 -0.09245728 -0.32569087
v1_10 0.002970200 0.33229552 -0.76830667
v1_11 0.337655169 -1.09374134 -1.40879781
v1_12 0.202565832 0.19658877 1.01280945

```

제 1축의 정준 변량은

$$Y1 = 1.69 \text{ intbld1} + 0.99 \text{ intbld2} - 0.61 \text{ intbld3}$$

$$X1 = 0.315 \text{ v1}_1 + 0.855 \text{ v1}_2 - 0.068 \text{ v1}_3 + 0.701 \text{ v1}_4 + 0.110 \text{ v1}_5 \\ - 0.438 \text{ v1}_6 - 1.571 \text{ v1}_7 - 0.092 \text{ v1}_8 + 0.007 \text{ v1}_9 + 0.003 \text{ v1}_{10} \\ + 0.338 \text{ v1}_{11} + 0.203 \text{ v1}_{12}$$

이고 이때의 정준상관은 0.604088 이다.

제 2 축의 정준 변량은

$$Y1 = -2.412*INTBLD1 -0.504*INTBLD2 -2.544*INTBLD3$$

$$X1 = -1.533*V1_1 + 0.189*V1_2 - 1.154*V1_3 + 0.725*V1_4 - 0.942*V1_5 \\ + 0.218*V1_6 + 0.875*V1_7 + 0.194*V1_8 + 0.092*V1_9 - 0.332*V1_{10} \\ + 1.094*V1_{11} - 0.197*V1_{12}$$

이때의 정준상관은 정준계수 값인 0.501245 이다.

-각 수량화 값을 중심화 하고 설명변량의 중요도를 볼 수 있도록 수량화 값의 범위를 계산

<제 1 축 원정계수 수량화>

정준상 관값	0.604088	중심화 수량값	*정준상 관값	범위	빈도	가중평균
	Y1					
A	1.69	1.00			A 16	0.69
B	0.99	0.30			B 8	
O	-0.61	-1.30			O 12	
AB	0	-0.69		2.30	AB 4	
	X1					
V1_1	0.32	0.20	0.12		V1_1 1 14 2 26	0.11
	0.00	-0.11	-0.07	0.32		
V1_2	0.86	0.28	0.17		V1_2 1 27 2 13	0.58
	0.00	-0.58	-0.35	0.86		
V1_3	-0.07	-0.01	-0.01		V1_3 1 35 2 5	-0.06
	0.00	0.06	0.04	0.07		
V1_4	0.70	0.32	0.19		V1_4 1 22 2 18	0.39
	0.00	-0.39	-0.23	0.70		
V1_5	0.11	0.07	0.04		V1_5 1 14 2 26	0.04
	0.00	-0.04	-0.02	0.11		
V1_6	-0.44	-0.24	-0.15		V1_6 1 18 2 22	-0.20
	0.00	0.20	0.12	0.44		
V1_7	-1.57	-0.67	-0.40		V1_7 1 23 2 17	-0.90
	0.00	0.90	0.55	1.57		
V1_8	-0.09	-0.05	-0.03		V1_8 1 18 2 22	-0.04
	0.00	0.04	0.03	0.09		
V1_9	0.01	0.00	0.00		V1_9 1 30 2 10	0.00
	0.00	0.00	0.00	0.01		
V1_10	0.00	0.00	0.00		V1_10 1 25 2 15	0.00
	0.00	0.00	0.00	0.00		
V1_11	0.34	0.20	0.12		V1_11 1 16 2 24	0.14
	0.00	-0.14	-0.08	0.34		
V1_12	0.20	0.11	0.07		V1_12 1 18 2 22	0.09
	0.00	-0.09	-0.06	0.20		

<제 2 축 원정계수 수량화>

정준상 관값	0.501245	중심화 수량값	*정준상 관값	범위	빈도	가중평균
	Y2					
A	2.41	0.58			A 16	1.83
B	0.50	-1.32			B 8	
O	2.54	0.72			O 12	
AB	0.00	-1.83		2.54	AB 4	

X2					V1_1		가중평균
V1_1	-1.53	-1.00	-0.50		1	14	
	0.00	0.54	0.27	1.53	2	26	
V1_2					V1_2		0.13
V1_2	0.19	0.06	0.03		1	27	
	0.00	-0.13	-0.06	0.19	2	13	
V1_3					V1_3		-1.01
V1_3	-1.15	-0.14	-0.07		1	35	
	0.00	1.01	0.51	1.15	2	5	
V1_4					V1_4		0.40
V1_4	0.72	0.33	0.16		1	22	
	0.00	-0.40	-0.20	0.72	2	18	
V1_5					V1_5		-0.33
V1_5	-0.94	-0.61	-0.31		1	14	
	0.00	0.33	0.17	0.94	2	26	
V1_6					V1_6		0.10
V1_6	0.22	0.12	0.06		1	18	
	0.00	-0.10	-0.05	0.22	2	22	
V1_7					V1_7		0.50
V1_7	0.87	0.37	0.19		1	23	
	0.00	-0.50	-0.25	0.87	2	17	
V1_8					V1_8		0.09
V1_8	0.19	0.11	0.05		1	18	
	0.00	-0.09	-0.04	0.19	2	22	
V1_9					V1_9		0.07
V1_9	0.09	0.02	0.01		1	30	
	0.00	-0.07	-0.03	0.09	2	10	
V1_10					V1_10		-0.21
V1_10	-0.33	-0.12	-0.06		1	25	
	0.00	0.21	0.10	0.33	2	15	
V1_11					V1_11		0.44
V1_11	1.09	0.66	0.33		1	16	
	0.00	-0.44	-0.22	1.09	2	24	
V1_12					V1_12		-0.09
V1_12	-0.20	-0.11	-0.05		1	18	
	0.00	0.09	0.04	0.20	2	22	

본 분석자는 SAS로 분석하여 제 2축의 원정계수가 양으로 나와 위와 같은 표를 작성하였으며, 여러분은 R로 분석하여 나온 음 값을 이용하여 해보길 바란다. 결과는 같은 결과를 보여주고 있을 것이다.

<정준상관분석에 의한 혈액형 자료의 수량화>

		제 1 축		제 2 축	
		수량화값	범위	수량화값	범위
혈액형	A	1.00		0.58	
	B	0.30		-1.32	
	O	-1.30		0.72	
	AB	-0.69	2.30	-1.83	2.54
합리적인가	예	0.12		-0.50	
	아니오	-0.07	0.32	0.27	1.53
기분파인가	예	0.17		0.03	
	아니오	-0.35	0.86	-0.06	0.19
타인을 의식하는 편인가	예	-0.01		-0.07	
	아니오	0.04	0.07	0.51	1.15
꼭 맞는것을 좋아하는가	예	0.19		0.16	
	아니오	-0.23	0.70	-0.20	0.72
조정반원인가	예	0.04		-0.31	
	아니오	-0.02	0.11	0.17	0.94
개성적인가	예	-0.15		0.06	
	아니오	0.12	0.44	-0.05	0.22
현실파인가	예	-0.40		0.19	
	아니오	0.55	1.57	-0.25	0.87
보수적인가	예	-0.03		0.05	
	아니오	0.03	0.09	-0.04	0.19
낭만파인가	예	0.00		0.01	
	아니오	0.00	0.01	-0.03	0.09
낙천가인가	예	0.00		-0.06	
	아니오	0.00	0.00	0.10	0.33
견실파인가	예	0.12		0.33	
	아니오	-0.08	0.34	-0.22	1.09
중간의식	예	0.07		-0.05	
	아니오	-0.06	0.20	0.04	0.20
정준상관 :		0.604088		0.501245	

1> 외적기준의 제1축 수량화 → A, B : (-) , O, AB : (+)

가장 큰 관련을 가지는 (범위가 가장 큰) 설명변량은 현실파인가(1.57) 이다.

V7의 값이 “예” 응답을 보이는 사례는 “O형”, “AB형”과 관련되어있고, “아니오” 응답을 보이는 사례는 “A형”, “B형”과 관련되어있다.

2> 외적기준의 제 2축 수량화 → AB, B : (-) , A, O : (+)

가장 큰 관련을 가지는 (범위가 가장 큰) 설명변량은 합리적인가(1.53) 이다.

V1의 값이 “예” 응답을 보이는 사례는 “B형”, “AB형”과 관련되어있고, “아니오” 응답을 보이는 사례는 “A형”, “O형”과 관련되어있다.

- 수량화방법 III

연습문제 1. 다음은 12 종류의 음식에 대한 10개 성, 연령 그룹의 반응패턴이다. 수량화방법III에 의한 분석을 하여라. 이 때 제1축과 제2축 수량화 값의 플롯을 제시하여라.

food	man1	man2	man3	man4	man5	wo1	wo2	wo3	wo4	wo5
1	1	1	1	0	0	1	1	1	1	0
2	0	0	0	0	1	1	0	0	0	1
3	1	1	1	1	0	1	1	1	1	0
4	0	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	0	0	0	1	1	0	0	0
7	1	1	1	1	0	1	1	0	1	0
8	0	1	1	1	1	0	1	1	1	1
9	1	1	1	1	1	0	0	0	0	1
10	1	1	0	1	1	1	0	1	1	1
11	0	0	1	1	0	0	1	1	1	1
12	0	1	1	1	1	0	1	1	1	1

food	1) 카레라이스	2) 냉면	3) 튀김국수	4) 된장국	
	5) 스키야키	6) 고로케	7) 햄	8) 생선회	
	9) 장어구이	10) 오뎅	11) 팔보채	12) 두부	
연령	15세이하	16~20세	21~30세	31~40세	41세이상

```

> data <- read.table("D:/R/cca3.txt", header=T)
> attach(data)
> library(ca)
> df <- data.frame(man1, man2, man3, man4, man5, wo1, wo2, wo3, wo4, wo5)
> x <- ca(df)

```

- txt 파일을 불러오며, 대응분석을 하기 위하여 ca라는 패키지를 설치 후 불러온다.
- data에서 분석에 필요한 변수를 frame으로 만들고, x라는 변수에 분석결과를 입력한다.
- 분석결과를 summary()와 x 로 불러온다.

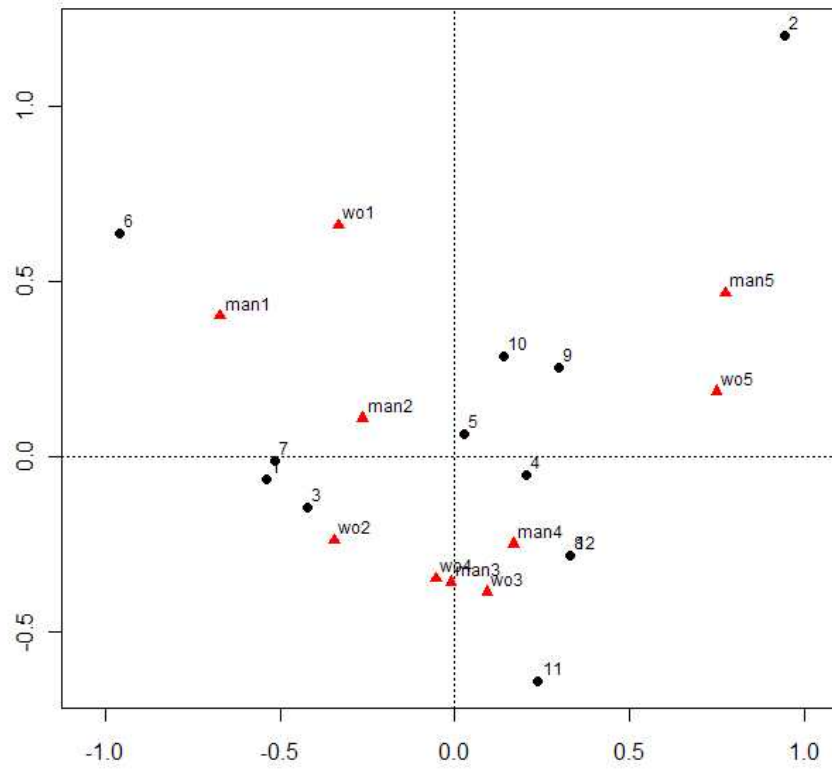

```
> summary(x)
```

```
Principal inertias (eigenvalues):
```

	dim	value	%	cum%	scree plot
[1,]	1	0.176122	40.4	40.4	*****
[2,]	2	0.131847	30.3	70.7	*****
[3,]	3	0.056714	13.0	83.7	*****
[4,]	4	0.026221	6.0	89.7	****
[5,]	5	0.019099	4.4	94.1	***
[6,]	6	0.011446	2.6	96.8	**
[7,]	7	0.010780	2.5	99.2	*
[8,]	8	0.002507	0.6	99.8	
[9,]	9	0.000830	0.2	100.0	
[10,]		-----	-----		
[11,]	Total:	0.435566	100.0		

```
> x
```

```
> plot(x)
```



- plot의 결과이다.

제 8 장

로지스틱 회귀분석과 판별분석

- 회귀분석은 기본적으로 종속변수와 독립변수가 모두 등간척도 이상으로 측정된 경우에 적용되는 통계기법을 말한다. 만약 독립변수가 명목이나 서열척도로 측정된 경우 독립변수를 더미변수로 전환하여 회귀분석을 적용한다. 그러나 종속변수가 질적인(qualitative) 척도, 즉 명목척도로 측정된 경우에는 회귀분석의 적용이 어려워진다.

- 명목척도로 측정된 종속변수를 독립변수(들)를 이용하여 예측하고자 하는 경우 로지스틱 회귀분석(logistic regression)과 판별분석(discriminant analysis)이 사용될 수 있다. 이 두 모형의 기본원리는 관찰값이 어떤 집단에 속하는지를 직접 예측하는 것이 아니라 독립변수들로 구성된 식을 이용하여 종속변수값을 예측하고, 이 값을 토대로 관찰값이 어느 집단에 속하는지를 예측하게 된다.

- 두 모형의 차이점은 판별분석에서는 독립변수들의 정규성(normality)과 각 집단들의 분산, 공분산이 동일해야 한다는 가정을 필요로 하는 반면, 로지스틱 회귀분석은 이러한 가정을 요구하지 않기 때문에 모형의 적용가능성이 높다는 것이다. 또한 로지스틱 회귀분석은 각 관찰치가 특정 집단에 속할 확률을 토대로 그 관찰자가 어느 집단에 속하는지를 예측하는 반면, 판별분석에서는 판별점수를 이용하여 관찰치가 어느 집단에 속하는지를 예측한다.

8.1 로지스틱 회귀분석