

1.1 R 개요

통계 계산과 그래프 작성을 위한 언어와 환경인 R 은 뉴질랜드 Auckland 대학의 Ross Ihaka 와 Robert Gentleman 이 1995 년 개발하고 R-Core 팀이 1997 년 결성 2002 년 R Foundation 이 설립되어 전세계에 무료로 배포되고 있다.

1.1.1 GNU(GNU is Not Unix) Project

- ☐ Unix-like Operating System
- ☐ 다양한 인터페이스, DBMS 데이터 수용
- ☐ 그래픽 기능이 강한 S 언어와 환경 하에서 개발
- ☐ S 코드 대부분은 수정 없이 R 에서 실행 가능

1.1.2 S language 기반

- + 설계자 John Chambers, 1998 (ACM award)
- + 데이터 통합, 시각화, 분석을 위한 객체지향형 프로그래밍 언어
- + 통계연구 프로젝트 착수 1976, 버전 4 출시 1998
- + 대화형 기반 기술 데이터 입출력, 변환 용이
- + 데이터 시각적 표현 중심으로 개발
- + Java 의 Eclipse 제공으로 프로그래머 개발 환경 개선
 - S 기반 통계시스템 상용 프로그램
 - S-Plus: 1993 라이선스 (TIBCO 회사 상용 소프트웨어), 2007 S-Plus 8 이 R 과 호환, 2008 Insightful (2001 설립, S-plus 기반) 회사 인수
 - Statistical Sciences (1988), Inc, R. Douglas Martin, 미국 Washington 대학 (Seattle), 통계학과 교수
 - Adopt Trellis 그래프, 1995
 - Embedded S-Plus in Excel, 1998
 - Mathsoft(1993) 현재 버전 8

1.1.3 R 장점

- + 사용 환경 ; small and effective
 - 대화식 프로그램 수행
 - 대용량 데이터 관리 및 처리, 행렬 연산, 그래픽 환경
- + Freeware and Open source

- GPL(General Public License) 개념으로 CRAN (Comprehensive R Archive Network)에서 배포
- 전세계 연구자의 개발 알고리즘, 함수 활용 가능

+다양한 확장성 및 범용성

- UNIX, Mac, Windows 운영체제 사용이 가능함
- Java, C, Fortran 프로그래밍 언어에 인터페이스 제공
- DBMS 데이터 접근 용이

+Bio-informatics, Microarray 분석 표준시스템

+Embedded R in Excel

+R 사용준비

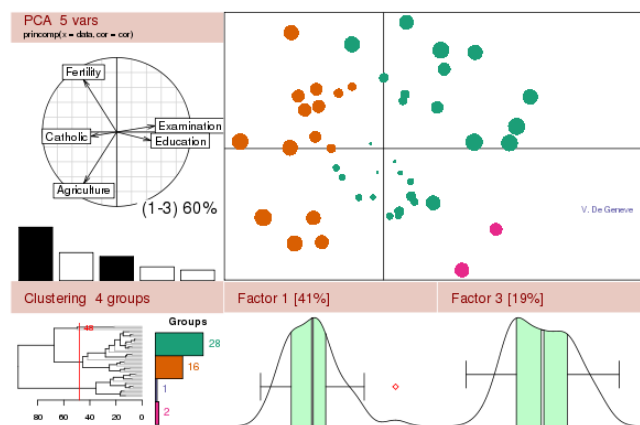
- Be smart: copy-left 소프트웨어 사용자, Bulky hater
- easy to use: 적은 시간 투자로 R 사용자 가능
- Be economical: free and get world-wide help
- Be challenge; 데이터, 프로그램 개별 관리, 행렬 개념의 데이터 관리,
- 데이터 분석: 함수 중심 알고리즘

1.2 R 설치

1.2.1 설치파일 가져오기

- + 설치 파일 웹사이트: <http://www.r-project.org>

The R Project for Statistical Computing



+ 설치 파일 찾기

 <p> About R What is R? Contributors Screenshots What's new? Download, Packages CRAN </p>	<p>첫 화면 왼쪽 메뉴에서 CRAN 링크를 선택하면 우측 화면에 설치 파일이 있는 Mirror 사이트(네트워크 트래픽을 줄이기 위하여 다른 컴퓨터 서버를 복사해 놓은 웹사이트 또는 컴퓨터 파일서버)가 출력된다.</p> <p>대한민국의 R Mirror 사이트는 다음과 같다.</p> <p>Korea http://bibs.snu.ac.kr/R/ </p>
---	--

+ 설치파일

<p>Download and Install R</p> <p>Precompiled binary distributions these versions of R:</p> <ul style="list-style-type: none"> • Linux • MacOS X • Windows 	<p>자신 컴퓨터 운영체계에 적절한 링크를 선택한다.</p> <p>Window XP, Vista, 윈도우 7 등은 Windows 링크를 클릭한다.</p>
<p>Subdirectories:</p> <p> base contrib </p>	<p>Binaries for base distribution Binaries of contributed packa</p>

“base” 링크를 선택하여 R 기본적인 프로그램 설치 파일을 다운로드 링크를 얻는다.

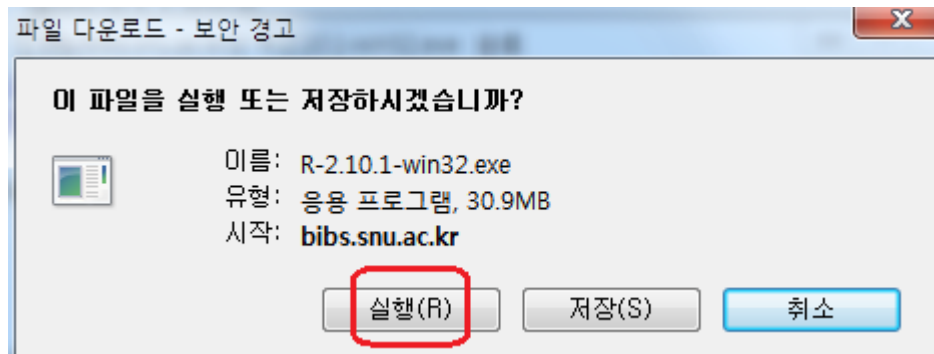
“contrib” 링크에는 부가적인 패키지들이 있다. 부가적 라이브러리 패키지는 필요할 때 R 프로그램 내의 메뉴에서 설치 할 수 있으므로 초기에는 “base”에 있는 아래 프로그램만 설치하면 된다.

[Download R 2.10.1 for Windows](#) (32 megabytes) (2010.01.23 현재)

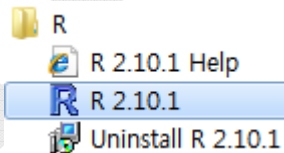
1.2.2 설치하기

+ 메인 R 프로그램 설치하기

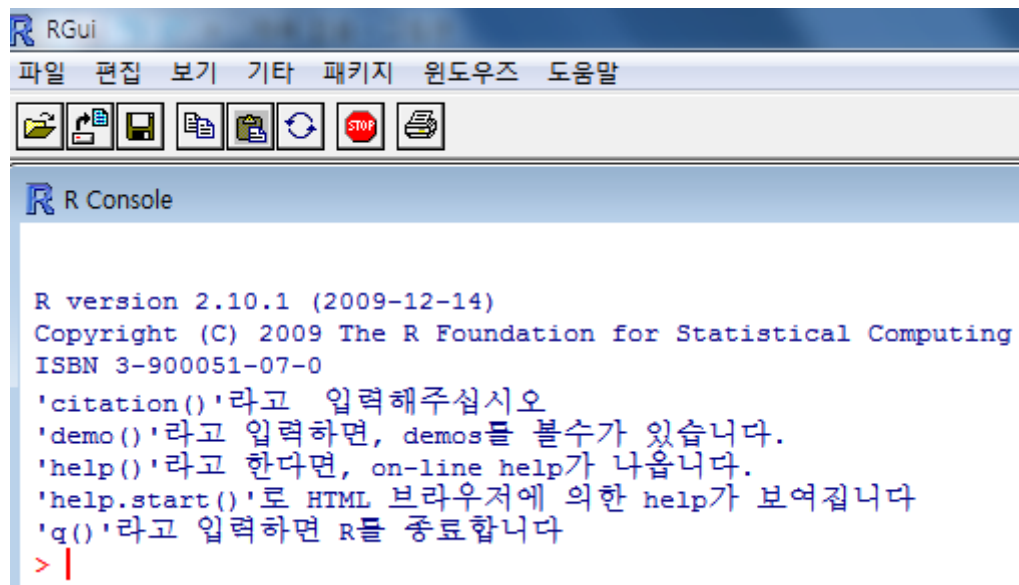
“Download R 2.10 for Windows” 링크를 클릭하면 아래 화면이 나온다. “저장”하여 프로그램을 설치할 수 있으나, 다시 사용할 필요가 없고 언제든지 다운받을 수 있으므로 바로 “실행”을 눌러 프로그램을 설치할 것을 권한다.



“실행”을 눌러 프로그램 설치가 시작되면 설치 마법사가 설치관련 옵션을 창에 띄워 선택하게 한다. 설치 후 환경설정을 수정할 수 있으므로 제시하는 디폴트 옵션으로 설치할 것을 권한다. 다음은 디폴트 옵션에 의해 설치된 R 프로그램의 설치 폴더는 `Main_Drive (C:) > Program Files > R > R-2.10.1`이고 프로그램 메뉴에 다음 아이콘 그룹이 생성된다. 그룹 메뉴에서 **R 2.10.1**을 선택하면 R 프로그램이 실행된다.

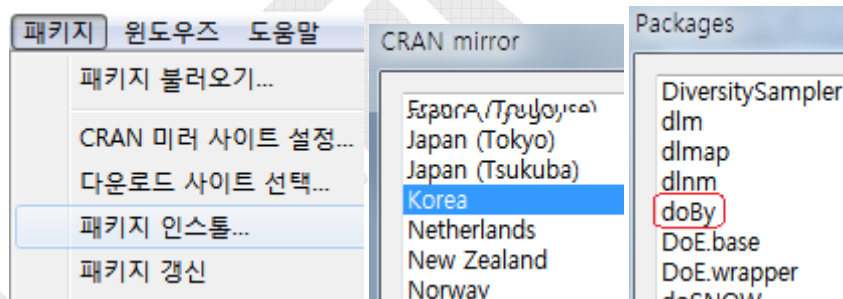


다음은 R의 첫 화면이다. 메뉴 바, 주요 아이콘 바, 콘솔 (console) 창이 나타난다. 콘솔 창에서는 라인 실행이 가능하다. 콘솔에서는 프로그램 (R에서는 Script라 한다) 실행, 결과 출력, 프로그램 오류 등이 출력된다. 그래프 창은 필요 시 팝업되며 적절한 그래프가 그려진다. 프로그램은 콘솔에서 가능하지만 스크립트 (Script, 프로그램) 창에서 작성이 가능하다.



+ 패키지 설치

“base”에 포함되지 않는 함수가 포함된 라이브러리는 “패키지” 메뉴에서 가능하다. 그룹 변수에 의해 측정형 변수의 기초통계량을 구하는 함수 `summaryBy()`는 “base”에 포함되어 있지 않고 “doBy” 라이브러리에 있다. 이를 설치하는 예로 패키지 설치 방법을 설명하겠다.



설치가 완료되면 아래와 같이 콘솔 화면에 출력된다. 패키지는 한 번만 설치하면 지속적으로 사용할 수 있다.

```
> utils::menuInstallPkgs()
이 세션으로 사용에 CRAN mirror를 선택해 주십시오
URL 'http://bibs.snu.ac.kr/R/bin/windows/contrib/2.10/doBy
Content type 'application/zip' length 242849 bytes (237 Kb
열린 URL
downloaded 237 Kb

패키지 'doBy'는 성공적으로 unpack 하였고 MD5 합계가 체크되
다운로드된 패키지는 C:\Users\User\AppData\Local\Temp\Rtmpu
있습니다
> |
```

1.3 One Bite

1.3.1 예제

학생 5 명의 통계학 중간고사 성적 (M, 80), (M, 70), (F, 90), (F, 85), (F, 95)을 입력하고 다음을 구하시오.

- (1) 중간고사 평균과 분산을 구하시오.
- (2) 성별 중간고사 평균과 분산을 구하시오.

EXAMPLE	맞보기
<pre>> score=c(80,70,90,85,95) > gender=c('M','M','F','F','F') > mean(score);sd(score) [1] 84 [1] 9.617692</pre> <p>*) 오브젝트; 데이터 (스칼라, 행렬, 관측치), 함수 결과가 저장되는 공간</p> <p>*) 라인 명령어; 화살표 ↑, ↓ 사용하면 이전, 이후 명령문이 나타난다.</p>	<p>> 함수 c()는 데이터 입력, 문자형 데이터의 경우에는 ‘ ’ 혹은 “ ” 사용</p> <p>> =의 의미: 우측의 결과가 좌측 변수 명으로 (이를 오브젝트, object 라 한다) 저장된다.</p> <p>> 함수 mean(), sd()은 정의된 오브젝트의 평균과 표준편차를 구한다.</p>

EXAMPLE	맞보기 2
<pre>> library(doBy) 요구된 패키지 survival을 로드중 요구된 패키지 splines을 로드중 > ds=data.frame(cbind(score,gender)) > summaryBy(score~gender,data=ds, + FUN=function(x){c(m=mean(x))}) gender score.m 1 F 4.0 2 M 1.5 + FUN=function(x){c(s=sd(x))})</pre> <p>*) 옵션: 내용 및 사용방법은 함수 help() 이용하면 된다.</p>	<p>> 함수 library(): 작업에 필요한 함수가 있는 라이브러리를 로딩함</p> <p>> 함수 cbind(): 오브젝트를 열 합치기</p> <p>> 함수 data.frame(): 오브젝트를 데이터 셋 만들기</p> <p>> 함수 summaryBy(): 범주형 변수에 의해 측정형 변수 기초통계량 구하기, () 안에 정의된 것은 적절한 옵션</p> <p>> +: 프로그램이 완료되지 않음을 의미</p>

```
> help(summaryBy)
```

<http://127.0.0.1:21220/library/doBy/html/summaryBy.html>

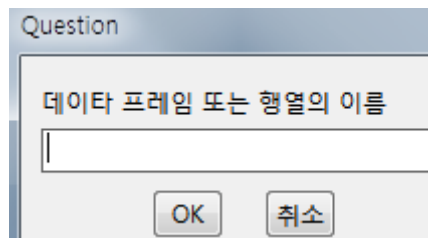
1.4 메뉴

1.4.1 파일 메뉴

R 코드의 소스...	> 스크립트 (script); 작업을 위한 프로그램 모임, 일정한 작업을 프로그램으로 저장해 놓으면 유사 작업 수행 용이
새로운 스크립트	> 작업공간 (workspace); 프로그램 수행 중 만들어진 오브젝트, 데이터 셋 등을 저장해 두면, R 프로그램 재실행 후도 동일하게 사용할 수 있음
스크립트 열기...	> 이력 (history): 작업 수행을 위해 사용하였던 Line 프로그램을 저장, 이전 라인 프로그램으로 가려면 화살표 ↑를 사용하면 된다.
파일 표시...	> 디렉터리 (directory) 변경: 외부 데이터가 있거나 스크립트, 이력 등을 저장하려는 폴더 지정
작업공간 불러오기...	> 파일에 저장; 콘솔 내용을 텍스트로 저장
작업공간 저장...	
이력 불러오기...	
이력 저장...	
디렉토리 변경...	
인쇄...	
파일에 저장...	
종료	

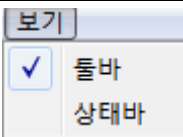
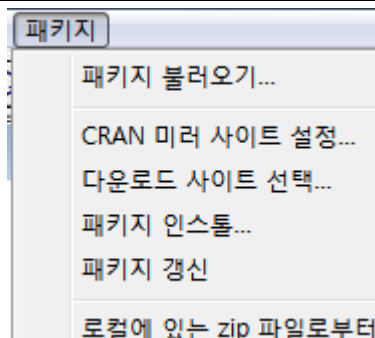
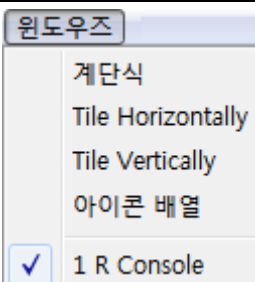
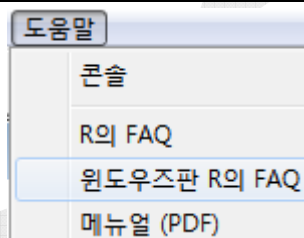
1.4.2 편집메뉴

복사	Ctrl+C	> 프로그램이나 결과 중 필요한 부분 복사, 잘라내기, 붙여 넣기
붙여넣기	Ctrl+V	> 콘솔 지우기; 콘솔에 있는 내용 모두 지우기
명령어만 붙여넣기		> 자료 편집기; 데이터 혹은 행렬 편집, 엑셀과 같은 스프레드시트 형식으로 열림
복사 후 붙여넣기	Ctrl+X	
전부 선택		
콘솔 지우기	Ctrl+L	
자료 편집기...		
GUI 설정...		



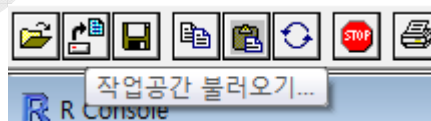
> GUI 설정: R의 GUI (Graphic User Interface) 환경 설정

1.4.3 기타 메뉴

 <p>> 아이콘 바와 아래 상태 바 보기</p>	 <p>> 필요한 패키지 설치 메뉴 (1.3.1 참고)</p>
 <p>> 열린 창 보기 설정</p>	 <p>> 도움말 기능</p>

1.4.4 아이콘 메뉴

메뉴 중 가장 많이 사용되는 기능을 아이콘으로 표현하여 사용을 용이하게 함, 아이콘에 마우스를 눌러 놓으면 작업 내용을 표시한다. 다음은 두 번째 아이콘에 마우스를 올려 놓았을 때 표시된 작업 내용이다.



1.5 알아두기

1.5.1 오브젝트 (object)

오브젝트는 R에서 모든 것이다. (Object is everything) 모든 오브젝트는 속성을 가지고 있고 이를 통하여 R 사용자는 R과 이야기하며, 이곳에 모든 내용을 이곳에 저장한다.

오브젝트의 형식은 숫자형 (numeric), 문자형 (characteristic), 요인 (분산분석의 요인), 논리형 (logical), 실수형 (double), 복소수형 (complex) 등이 있다.

오브젝트에 저장될 수 있는 내용은 벡터, 행렬, 데이터, 배열, 리스트, 함수 (분석) 결과 내용 등이 있으며, 저장하려면 기호 `<-`, `=` (동일한 작업 내용) 을 사용한다. 이를 오브젝트 할당 (assignment)라 한다.

EXAMPLE	오브젝트 활용
<pre>> sqrt(2) [1] 1.414214 > x=sqrt(2) > x [1] 1.414214 > y<-sqrt(2);y [1] 1.414214 > ls() [1] "x" "y"</pre>	<p>> <code>=</code>, <code><-</code>을 사용하면 우측 작업 내용이 좌측 이름으로 오브젝트로 저장된다.</p> <p>> 오브젝트 내용을 보려면 오브젝트 이름을 입력하면 된다.</p> <p>> 오브젝트의 리스트를 보려면 함수 <code>ls()</code> 이용한다.</p> <p>> <code>;</code>은 명령문을 나누는 역할을 한다.</p>

오브젝트 이름은 영어 알파벳, 숫자, 마침표 `.`, under-bar `_` 등의 조합을 사용할 수 있으나, 숫자, `.`, `_` -은 첫 글자로 사용할 수 없다. 그리고 내장 함수 (`sqrt`, `sin`, `pi`) 이름나 명령어로 (`for`, `if`, `matrix`, `read.xls`) 예약된 단어는 오브젝트 이름으로 사용할 수 없다. R에서는 대소문자 구별해야 (case sensitive) 하므로 오브젝트 이름도 동일한 적용을 받아, `ds`와 `Ds`는 서로 다른 오브젝트 이름이다. 이름 길이는 충분히 길어도 되지만 가능하면 짧고 오브젝트 내용을 인지할 수 있으면 된다. 학생들의 성적관련 데이터 셋이면 'scoredataset' 보다는 'score' 혹은 'score.ds'가 더 적절해 보인다.

오브젝트를 향후에도 사용하고 싶다면 **작업공간 저장...** (workspace)을 이용하여 저장해 두었다가 향후에 불러 사용하면 된다. 필요 없는 오브젝트를 지우려면 다음과 같이 함수 `rm()`을 사용하면 된다. 함수 `search()`를 사용하면 현재 작업환경에서 구동 중인 패키지의 이름들이 리스트 된다.

```
> rm(y)
> ls()
[1] "x"

> search()
[1] ".GlobalEnv"
[6] "package:graphics"
[11] "Autoloads"

"package:doBy"
"package:grDevice"
"package:base"
```

1.5.2. 함수 (function)

+ 내장 함수 (built-in function)

R의 라이브러리 패키지에는 여러 함수가 내장되어 있다. 다음 그림은 1.2.2 절에서 설치한 "doBy" 패키지에 내장된 함수와 그에 대한 사용 설명 도움말 (html 폴더 내에 있음) 화면을 캡처한 것이다.

<p>문서 라이브러리</p> <p>doBy</p> <p>이름</p> <ul style="list-style-type: none"> data doc help html Meta R DESCRIPTION INDEX MD5 NAMESPACE 	<h2>Documentation for package 'doBy' versio</h2> <h3>User Guides and Package Vignettes</h3> <h3>Help Pages</h3> <table> <tr> <td>budworm</td><td>Effect of Insecticide on survival of tobacco</td></tr> <tr> <td>codstom</td><td>Diet of Atlantic cod in the Gulf of St. Lawre</td></tr> <tr> <td>dietox</td><td>Growth curves of pigs in a 3x3 factorial exp</td></tr> <tr> <td>doBy</td><td>Various utilities which includes functions for</td></tr> <tr> <td>dose.LD50</td><td>Calculate LD50</td></tr> <tr> <td>esticon</td><td>Contrasts for lm, glm, lme, and geeglm obje</td></tr> <tr> <td>subsetBy</td><td>Finds subsets of a dataframe which is split t</td></tr> <tr> <td>summaryBy</td><td>Function to calculate groupwise summary st</td></tr> <tr> <td>transformBy</td><td>Function to make groupwise transformation</td></tr> </table>	budworm	Effect of Insecticide on survival of tobacco	codstom	Diet of Atlantic cod in the Gulf of St. Lawre	dietox	Growth curves of pigs in a 3x3 factorial exp	doBy	Various utilities which includes functions for	dose.LD50	Calculate LD50	esticon	Contrasts for lm, glm, lme, and geeglm obje	subsetBy	Finds subsets of a dataframe which is split t	summaryBy	Function to calculate groupwise summary st	transformBy	Function to make groupwise transformation
budworm	Effect of Insecticide on survival of tobacco																		
codstom	Diet of Atlantic cod in the Gulf of St. Lawre																		
dietox	Growth curves of pigs in a 3x3 factorial exp																		
doBy	Various utilities which includes functions for																		
dose.LD50	Calculate LD50																		
esticon	Contrasts for lm, glm, lme, and geeglm obje																		
subsetBy	Finds subsets of a dataframe which is split t																		
summaryBy	Function to calculate groupwise summary st																		
transformBy	Function to make groupwise transformation																		

R 에 내장된 함수는 수학함수, 문자함수, 통계분포함수, 통계함수, 데이터 생성 함수 등이 있다. 다음은 통계학에서 주로 사용되는 함수 중심으로 함수명과 기능을 예제 중심으로 설명한 것이다.

+수학함수 (mathematical function)

함수	기능	R 에서 실행
abs(x)	절대값,	<pre>> x=-2.34;y=1.36;z=8 > abs(x);sqrt(z);z^(3);z^(1/3) [1] 2.34 [1] 2.828427 [1] 512 [1] 2</pre>
sqrt(x)	제곱근,	
x^n	n 승 값	
ceiling(x)	올림	<pre>> x=-2.34;y=1.36 > ceiling(x);ceiling(y) [1] -2 [1] 2</pre>
floor(x)	내림	<pre>> floor(x);floor(y) [1] -3 [1] 1</pre>
trunc(x)	소수점 이하 버림	<pre>> trunc(x);trunc(y) [1] -2 [1] 1</pre>

round(x, digits=n)	소수점 n 자리 이하 반올림	<pre>> x=3.678 > round(x,2) [1] 3.68</pre>
cos(), sin(), tan()	삼각함수 값	<pre>> sin(90);sin(pi/2) [1] 0.8939967 [1] 1</pre>
log(), log10(), log2()	자연로그 값, 상용로그 값, 밑이 2 인 로그 값	
exp()	지수함수 값	<pre>e^x exp(3) [1] 20.08554</pre>
prod()	데이터 곱	<pre>> prod(2,3,5) [1] 30</pre>
choose(n, r), factorial(n)	조합, combination 순열, factorial	<pre>> choose(5,3) [1] 10 > factorial(5) [1] 120</pre>

+문자함수 (characteristic function)

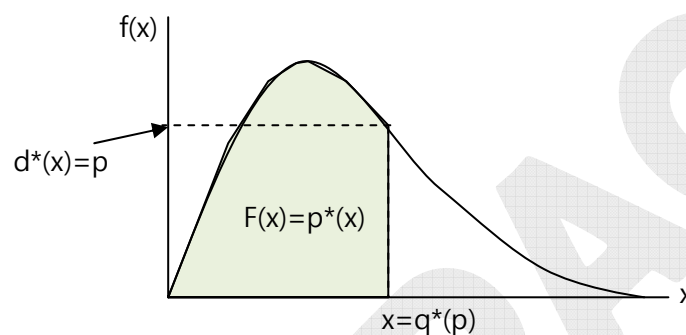
함수	기능	R 에서 실행
substr(x, s, e)	단어 X 에서 s 자리부터 e 자리까지 선택	<pre>> x='statistics' > substr(x,3,6) [1] "atis"</pre>
strsplit(x, 'sp')	sp 단어에서 단어가 나누어짐	<pre>> strsplit(x,"t") [[1]] [1] "s" "a" "is" "ics"</pre>
paste()	단어 붙이기	<pre>> paste('X',1:4,sep=" ") [1] "X1" "X2" "X3" "X4" > paste('X',1:4,sep="i") [1] "Xi1" "Xi2" "Xi3" "Xi4" > paste('Today is',date()) [1] "Today is Wed Jan 27 2010"</pre>
toupper(x) tolower(x)	대문자 만들기 소문자 만들기	<pre>> z=paste('I am','Fine') > toupper(z) [1] "I AM FINE" > tolower(z) [1] "i am fine"</pre>
sub(r, o, x)	단어 x 에서 o 문자를 r 로 대체함	<pre>> z=paste('I am','Fine') > sub('Fine','Great',z) [1] "I am Great"</pre>

+통계함수 (statistical function)

함수	기능	R 에서 실행
데이터		<pre>> x=round(rnorm(10,50,10));x</pre> <pre>[1] 46 53 39 31 60 41 52 50 42 70</pre> <p>함수 rnorm(n, m, s)는 평균이 m, 표준편차 s 인 정규분포를 따르는 데이터 n 개를 랜덤하게 만든다. 랜덤하게 만들 때 seed 는 실행 시각으로 할당되므로 매번 다르게 된다.</p> <p>함수 round()는 값을 정수로 반올림한다.</p>
mean()	평균 10% 절삭평균	<pre>> mean(x)</pre> <pre>[1] 48.4</pre> <pre>> mean(x, rm=0.1)</pre> <pre>[1] 48.4</pre> <p>(상, 하 각각 10% 절삭)</p>
sd(), var()	표준편차, 분산	<pre>> sd(x);var(x)</pre> <pre>[1] 11.20714</pre> <pre>[1] 125.6</pre>
median()	중위수	<pre>> median(x)</pre> <pre>[1] 48</pre>
quantile(x, p)	사분위값	<pre>> quantile(x)</pre> <pre>0% 25% 50% 75% 100%</pre> <pre>31.00 41.25 48.00 52.75 70.00</pre> <pre>> quantile(x,c(0.1,0.9))</pre> <pre>10% 90%</pre> <pre>38.2 61.0</pre> <pre>> quantile(x,0.8)</pre> <pre>80%</pre> <pre>54.4</pre>
range()	범위	<pre>> range(x)</pre> <pre>[1] 31 70</pre>
sum(), prod()	합계, 곱	<pre>> sum(x);prod(x)</pre> <pre>[1] 484</pre> <pre>[1] 5.542629e+16 (= 5.5×10¹⁶)</pre>
diff(,lag=t)	앞의 숫자 - 뒤 숫자 (시차=t)	<pre>> diff(x)</pre> <pre>[1] 7 -14 -8 29 -19 11 -2 -8 28</pre> <pre>> diff(x,2)</pre> <pre>[1] -7 -22 21 10 -8 9 -10 20</pre>
min(), max()	최소값, 최대값	<pre>> min(x);max(x)</pre> <pre>[1] 31</pre> <pre>[1] 70</pre>

+ 통계분포함수 (statistical distribution function)

함수	기능
$d^*(x, \text{모수})$	확률밀도함수 확률 값, $f(x)$
$p^*(x, p, \text{모수})$	분포함수 값, $F(x)$
$q^*(p, \text{모수})$	역분포함수 값, $F^{-1}(p)$
$r^*(n, \text{모수})$	분포함수 따르는 데이터 n 개 랜덤하게 생성



분포 이름	R 함수	모수
beta	beta	shape1, shape2
binomial	binom	size, prob
Cauchy	cauchy	location, scale
chi-squared	chisq	df, ncp
exponential	exp	rate
F	f	df1, df2
gamma	gamma	shape, scale
geometric	geom	prob
hypergeometric	hyper	m, n, k
log-normal	lnorm	meanlog, sdlog
logistic	logis	location, scale
negative binomial	nbinom	size, prob
normal	norm	mean, sd
Poisson	pois	lambda
Student's t	t	df, ncp
uniform	unif	min, max
Weibull	weibull	shape, scale
Wilcoxon	wilcox	m, n

EXAMPLE

사용 방법과 예제는 1.5.5 스크립트에서 설명하기로 한다.

+기타함수 (miscellaneous)

함수	기능	R 에서 실행
seq(s, e, b)	정수 s 부터 e 까지 b 씩 증가	<pre>> seq(1,10,3) [1] 1 4 7 10</pre>
rep(a:b, r)	정수 a 부터 b 까지 r 반복	<pre>> rep(1:3,2) [1] 1 2 3 1 2 3</pre>
rep(a, r)	정수 a 를 r 반복	<pre>> rep(1,6) [1] 1 1 1 1 1 1</pre>

1.5.3 연산자 (operator)

연산자는 스칼라 (실수), 벡터, 행렬, 데이터 오브젝트의 계산을 위한 명령어이다.

+수학 연산자 (arithmetic operator)

함수	기능	R 에서 실행
+, -, *, /	사칙 연산	<pre>> x=7;y=2 > x/y;x*y [1] 3.5 [1] 14</pre>
**, ^	승수 연산	<pre>> 3^(-2) [1] 0.1111111 > 3^(1/2) [1] 1.732051</pre>
%%	나머지	<pre>> x=7;y=2 > x%%y [1] 1</pre>
%/%	나눈 몫	<pre>> x%/y [1] 3</pre>

+논리 연산자 (logical operator)

함수	기능
<, <=, >, >=	조건에서 양변의 크기 대소
==	조건에서 양변 동일, =은 오브젝트 지정에 사용
!=	조건에서 양변 다름
, &	or 양 조건 중 하나 참, and 두 조건 모두 참
isTRUE(x)	X의 논리 참 여부 참이면 TRUE, 거짓이면 FALSE의 값을 저장 <pre>> isTRUE(3>2) [1] TRUE > isTRUE(3==2) [1] FALSE</pre>

1.5.4 제어문 (control statement)

R은 라인 명령문 왼쪽에서 오른쪽으로, 위에서 아래로 순서대로 실행된다. 이를 제어하는 문장을 제어문이라 한다.

+FOR 문장

for(변수 in 연속) {문장}	연속에 지정된 값만큼 변수 값이 변화하면서 '문장'을 반복 실행한다.
--------------------	--

EXAMPLE	for 활용
<pre>> for(i in 1:3){x=i*3;print(x)} [1] 3 [1] 6 [1] 9</pre>	<ul style="list-style-type: none"> > 함수 print()는 괄호 안 화면 출력 > 변수 i=1, 2, 3 인 순차적으로 증가되면서 {} 문장 반복된다.
<pre>> x=c(50,60,80,90,95) > for(i in length(x):1){print(i)} [1] 5 [1] 4 [1] 3 [1] 2 [1] 1</pre>	<ul style="list-style-type: none"> > 함수 length()는 괄호 안의 데이터 크기 > in 에서 앞의 숫자가 더 큰 경우에는 순차적으로 감소한다.

+WHILE 문장

while(조건) {문장}	조건이 만족하는 동안 문장 반복 실행된다.
----------------	-------------------------

EXAMPLE

while 활용

```
> while(i<=3){print(i);i=i+1}
[1] 1
[1] 2
[1] 3
```

- > 값이 설정되지 않으면 R 내부 초기값은 0 이나 실제 존재하지 않는다. 그러므로 0 은 출력되지 않는다.
- > i=i+1 은 i 의 값을 1 씩 증가시킨다.
- > i 가 3 이 되는 순간까지 반복 실행한다.

+IF-ELSE 문장

if (조건) {문장}	조건이 만족하면 문장이 실행된다.
--------------	--------------------

EXAMPLE

if-else 활용

```
> for(i in 1:10){
+ if(i%%2==0){print(i)}}
[1] 2
[1] 4
[1] 6
[1] 8
[1] 10
```

- > '%2'는 i 값을 2 로 나누었을 때 나머지 값
- > 값 i 의 2 로 나눈 나머지가 0 인 경우 i 값을 출력하게 된다.

```
> for(i in 1:10){
+ if(i%%2==0){cat('Even ',i,"\n")}}
+ else{cat(i,'는 홀수','\n')}}
1 는 홀수
Even 2
3 는 홀수
Even 4
5 는 홀수
Even 6
7 는 홀수
Even 8
9 는 홀수
Even 10
```

- > if (조건) {문장 1} else {문장 2}; 조건이 만족하면 문장 1, 그렇지 않으면 문장 2 을 실행
- > 함수 cat()은 print()와 유사하게 화면 출력 기능, 오브젝트 뿐 아니라 문자도 출력 가능
- > '\n'은 출력 시 라인 바꿈, '\b'는 탭 이동을 기능

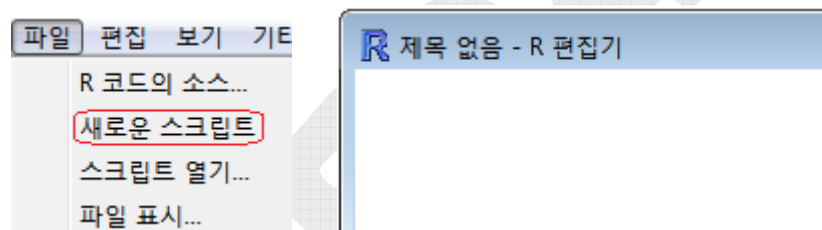
1.5.5 스크립트 (script)

콘솔에서는 라인 타입 명령문 입력과 실행으로 작동하여 Enter 키를 입력하지 않는 한 라인에 작업한 내용은 실행되지 않는다. 그러므로 상이한 명령문을 연속하여 동일 라인에 입력하고자 하면 명령문 사이에 세미콜론 (;)을 사용하면 된다.

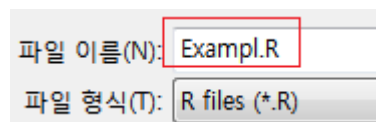
명령문 길이가 길어 한 라인에 들어가기 못하는 경우 (Enter 키를 치기 전까지는 하나의 명령문으로 R은 인지하고 있음) 명령어 중간에서 Enter 키를 치면 커서는 다음 라인에 위치하고 > 대신 플러스 (+) 사인이 라인 첫 머리에 나타난다.

이전 라인 명령문을 재실행하려면 방향 키 ⏮, ⏭ 을 사용하면 된다. 라인 명령문을 향후에도 사용하려면 **이력 저장...**을 이용하여 명령어들을 저장한 후 향후 불러 사용하면 된다.

라인 에디터 작업 환경은 다소 불편하므로 다수의 라인 명령어들 동시에 실행하거나 원하는 함수를 만들기, 작업에 필요한 프로그램 작업을 위해서는 스크립트를 활용하는 것이 좋다. 스크립트는 SAS의 프로그램 작업과 동일하다. 새로운 스크립트 만들기는 다음과 같이 하면 스크립트 창이 열린다.



스크립트 작업이 끝나면 스크립트를 저장하여 **다른 이름으로 저장...** 다시 불러 사용하면 된다. 저장할 때는 R을 확장자 사용하는 것이 적절하다. 이는 R은 다른 소프트웨어와는 달리 확장자 R이 자동으로 붙여지지 않는다. 그러므로 스크립트 이름을 붙일 때는 다음과 같이 이름 뒤에 확장자까지 붙여 저장하기를 권장한다.



스크립트에 작성된 프로그램은 라인 별로 실행할 수도 있으며 전체 프로그램을 한 번에 실행할 수도 있다. 실행된 명령문은 콘솔에 명령문과 함께 결과가 출력된다.

> **라인 별 실행:** 커서를 명령문 라인 아무 곳이나 놓고 ⌘+↵을 치면 라인 명령문이 실행되어 콘솔에 명령문과 결과가 출력되고 커서는 다음 라인 명령문으로 이동한다.

커서가 마지막 라인에 있으면 더 이상 아래로 이동하지 않는다.

> 전체 스크립트 실행: 아무 곳이나 놓고 **⌘+⌘**를 입력하면 스크립트 명령문 전체가 선택된다. **⌘+⌘**를 치면 스크립트 전체가 실행되어 콘솔에 나타난다. 스크립트 일부만 선택하고 싶으면 마우스로 실행 원하는 부분을 선택한 후 **⌘+⌘**를 치면 된다.

스크립트에서 명령문을 실행하지 않으려면 명령문 제일 앞에 '#' 표시를 하여 실행되지 않도록 한다. 이를 주석문 (comment statement)이라 한다.

EXAMPLE	스크립트 활용
<p>R I:\TMP\W3 - R 편집기</p> <pre>#예제 스크립트 x=c(50,60,80,90);x lnx=log(x);lnx mean(x) s=sd(x)</pre> <p>> 라인 실행해 보자. > 전체 실행해 보자.</p>	<p>콘솔 내용</p> <pre>> #예제 스크립트 > x=c(50,60,80,90);x [1] 50 60 80 90 > lnx=log(x);lnx [1] 3.912023 4.094345 4.382027 4.499810 > mean(x) [1] 70 > s=sd(x)</pre>
<p>R 제목 없음 - R 편집기</p> <pre>for(i in 1:10){ if(i%%2==0) {cat('Even ',i,"\n")} else{cat(i,'는 홀수','\n')}} </pre> <p>> 전체 실행하면 오른쪽 결과가 콘솔 화면에 나타난다.</p>	<p>1.5.4 절의 if-else 제어문 예제</p> <pre>> for(i in 1:10){ + if(i%%2==0) + {cat('Even ',i,"\n")} + else{cat(i,'는 홀수','\n')}} 1 는 홀수 Even 2 3 는 홀수 Even 4 5 는 홀수 Even 6 7 는 홀수 Even 8 9 는 홀수 Even 10</pre>

EXAMPLE

스크립트 활용

```
> #유의확률 계산: p*() 함수 이용
>
> ts=2 #검정통계량, n=10
> 1-pnorm(2,0,1) #z-검정
[1] 0.02275013
> 1-pt(2,9) #t-검정
[1] 0.03827641
>
> #기각역 계산: q*() 함수 이용
>
> alpha=0.05 #유의수준, 양측, n=10
> qnorm(0.975)
[1] 1.959964
> qt(0.975,9)
[1] 2.262157
```

R 제목 없음 - R 편집기

#유의확률 계산: p*() 함수 이용

```
ts=2 #검정통계량, n=10
1-pnorm(2,0,1) #z-검정
1-pt(2,9) #t-검정
```

#기각역 계산: q*() 함수 이용

```
alpha=0.05 #유의수준, 양측, n=10
qnorm(0.975)
qt(0.975,9)
```

+PLOT() 함수

plot(x, y, main=' ', sub=' ', xlim=c(a, b), ylab=' ', type=' ')		산점도 그리기
x	x-축 변수 지정	
y	y-축 변수 지정	
main=	그래프 제목	
sub=	그래프 아래 제목	
xlim=	x-축 좌표 눈금 지정, y 축은 ylim	
ylab=	y-축 제목 지정, x-축은 xlab	
type=	p	관측치 점을 ○ 으로 표현
	l	라인으로 그리기 —
	b	점과 라인 모두 그리기 —○—
	c	점선으로 그리기 ---
	o	선과 점을 동시에 그리기 —○—
	h	히스토그램을 선으로 형태
	s	계산 형식의 히스토그램
	n	그래프 사용하지 않음

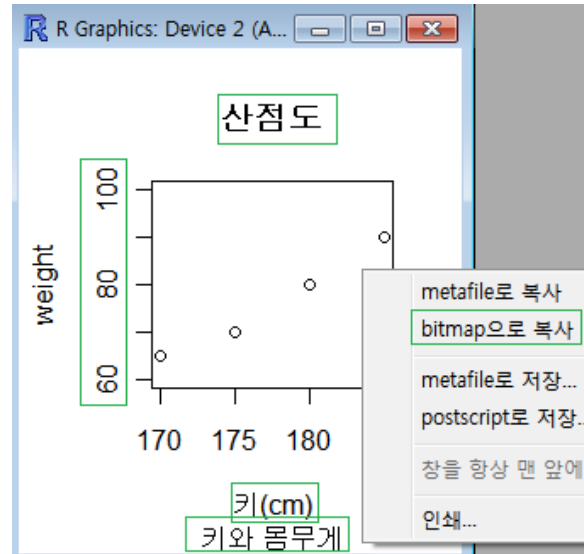
EXAMPLE

산점도 그리기

R 제목 없음 - R 편집기

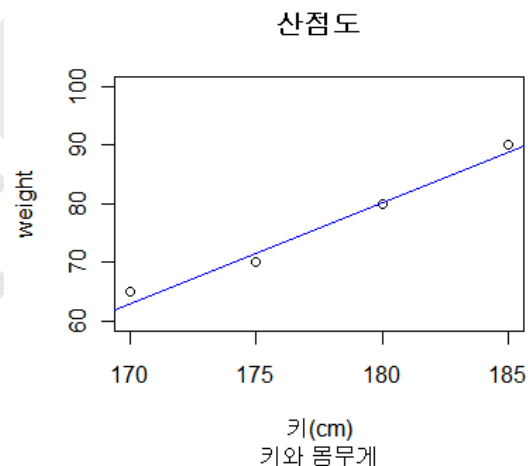
```
#산점도 그리기
weight=c(80,70,65,90,70)
height=c(180,175,170,185,175)
plot(height,weight,type='p',
      main='산점도',sub='키와 몸무게',
      xlab='키 (cm)',ylim=c(60,100))
```

- > 그래프 창이 열리고 산점도가 그려진다.
- > 그래프 창의 크기를 원하는 크기로 조절한 후 오른쪽 마우스 버튼을 눌러 그래프를 저장하여 사용한다.



```
xlab='키 (cm)',ylim=c(60,100))
#직선 그리기
abline(lm(weight~height),
       col='blue')
```

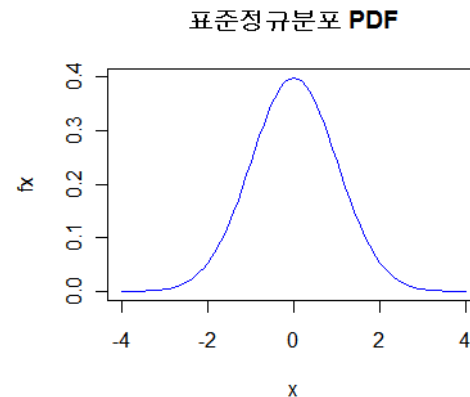
- > 그려진 그래프 위에 회귀직선을 그리려면 함수 abline()을 이용하면 된다. 옵션 col 은 직선 색을 정한다.



EXAMPLE

확률밀도함수 그리기

```
R I:\Books\WR\활용\ch1.R - R 편집기
#확률분포함수 그리기: 표준정규분포
x=seq(-4,4,0.1)
fx=dnorm(x,0,1)
plot(x,fx,type='l',
      main='표준정규분포 PDF',
      col='blue')
```

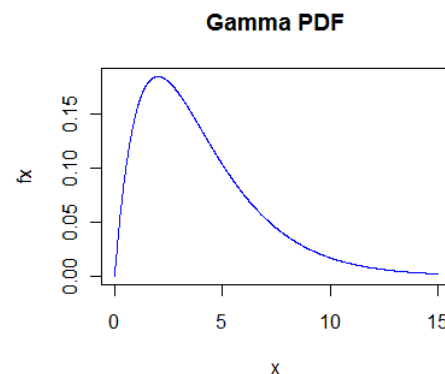


- > $x = -4, -3.9, -3.8, \dots$ 0.1 씩 증가한다. 작은 값으로 증가시킨 이유는 그래프를 보다 smoothing 하게 그리기 위함이다.
- > x 값이 순차적으로 `dnorm()` 함수에 들어가 fx 값이 계산된다.
- > x 의 범위를 $(-4, 4)$ 로 지정한 이유는 표준정규분포의 경우 $(-3, 3)$ 사이의 확률 값이 99.9%이기 때문이다.

EXAMPLE

확률분포함수 그리기

```
R I:\Books\WR\활용\ch1.R - R 편집기
#확률분포함수 그리기
#감마분포PDF (a=2,b=0.5)
x=seq(0,15,0.01)
fx=dgamma(x,2,0.5)
plot(x,fx,type='l',
      main='Gamma PDF',
      col='blue')
```



- > 모수 (α, β) 인 감마분포의 평균은 $\alpha / \beta = 4$, 분산은 $\alpha / \beta^2 = 8$
- > 평균 $\pm 4 \times (\text{표준편차})$ 를 범위로 하면 적절하다. 감마분포 확률변수는 0 보다 큰 값을 가지므로 음수를 구간으로 설정할 필요는 없다.

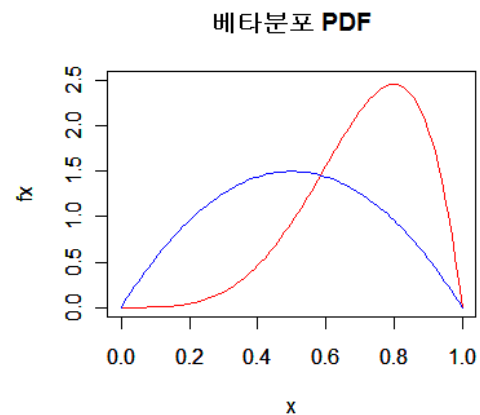
EXAMPLE

2 개 이상 그래프 동일 화면 그리기

```

R I\WBooks\WR활용\ch1.R - R 편집기
#확률분포함수 그리기: 베타분포
x=seq(0,1,0.01)
fx1=dbeta(x,5,2)
fx2=dbeta(x,2,2)
x11()
split.screen(c(1,1))
screen(1)
plot(x,fx1,type='l',
     main='베타분포 PDF',col='red',
     ylab='fx',xlab='x',
     xlim=c(0,1),ylim=c(0,2.5))
screen(1)
plot(x,fx2,type='l',col='blue',
     ylab='fx',xlab='x',
     xlim=c(0,1),ylim=c(0,2.5))

```



- > 함수 windows() 혹은 x11(); 기존의 그래프는 없애고 새로운 그래프 창을 연다.
- > 함수 split.screen(); 그래프 창을 원하는 크기로 나눈다. c(1,2)로 하면 행 1, 열 2 개로 나누어 2 개의 그래프를 한 화면에 그릴 수 있다. 이 경우 화면은 screen(1), screen(2)가 된다.

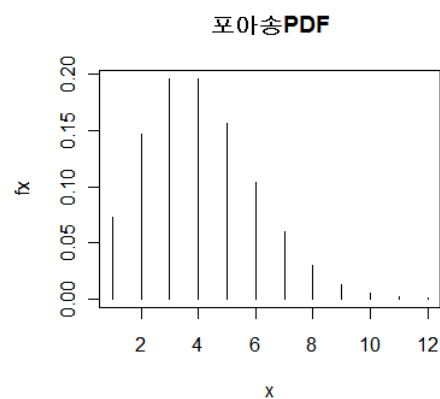
EXAMPLE

이산형 확률밀도함수

```

R I\WBooks\WR활용\ch1.R - R 편집기
#포아송확률밀도함수 그리기
x=seq(1,12,1)
fx=dpois(x,4)
plot(x,fx,main='포아송PDF',
     type='h')

```



- > 함수 dpois(x, λ); 모수 λ (평균=4)인 포아송 분포의 x에서의 확률 값
- > 함수 plot()에서의 h 옵션은 막대 형태의 히스토그램

EXAMPLE

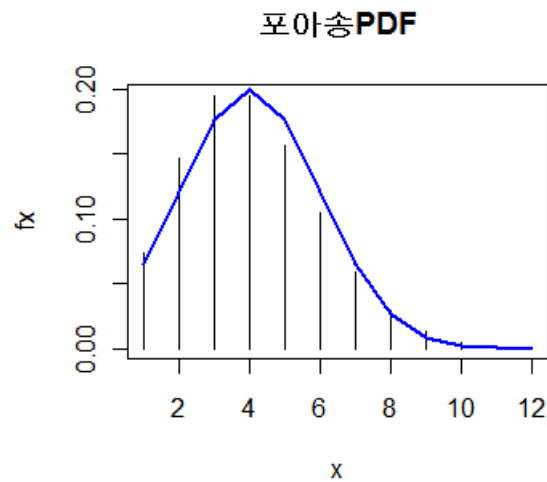
정규분포 겹쳐 그리기

```

R I\#Books\#R활용\#ch1.R - R 편집기
#포아송확률밀도함수 그리기
x=seq(1,12,1)
fx=dpois(x,4)
plot(x,fx,main='포아송PDF',
type='h')
ynorm<-dnorm(x,4,sqrt(4))
lines(x,ynorm, col="blue", lwd=2)

```

> 옵션 lwd 는 라인의 굵기 지정
> x 값이 정수만 가지고 있어
smoothing 이 완전하지 않음.

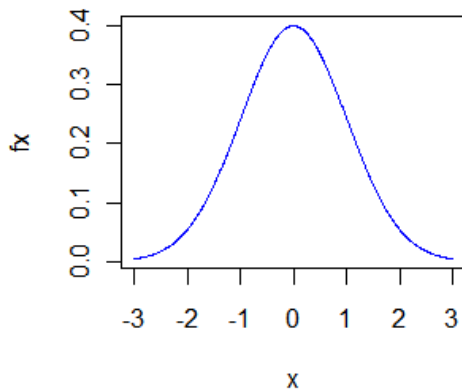


EXERCISE

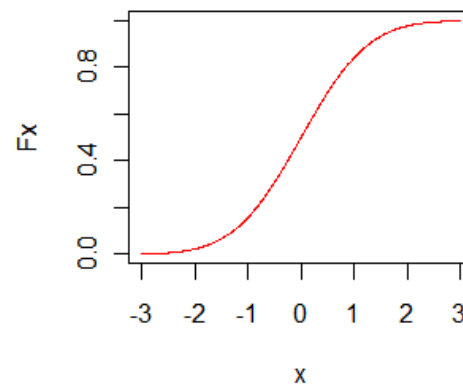
4 개의 그래프(표준정규분포, 누적표준정규분포, 평균 1 인 지수분포의 확률밀도함수와 누적분포함수)를 그리는 스크립트를 작성하시오.

누적분포를 그리기 위해서는 함수 p^* ()를 사용하면 된다.

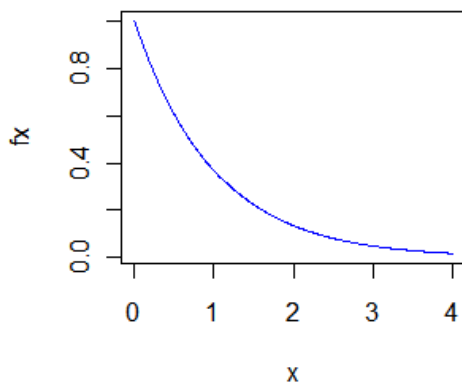
표준정규분포 PDF



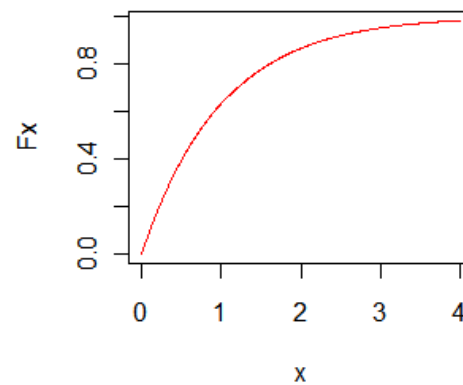
표준정규분포 CDF



지수분포 PDF m=10



지수분포 CDF



1.5.5 함수 만들기

원하는 작업을 하기 위하여 일련의 내장된 함수들을 이용하거나 다수의 명령문이 필요한 경우에는 스크립트를 이용하면 편리함을 1.5.4 절에서 언급하였다. 이렇게 사용자가 작성한 스크립트를 자신의 함수로 만들어 내장된 함수와 동일하게 사용할 수 있다.

함수이름 = function (모수) {문장}

모수는 스크립트 문장 내에 변하는 값

+ 모수 없는 함수

EXAMPLE		게임 함수
<pre> R I:\Books\WR활용\ch1.R - R 편집기 #하이 로우 게임 game=function() { x=round(runif(1,0,1)*100) for(i in 1:10){ cat("Guess? ") g=scan(nmax=1,quiet=T) if (g>x) {cat("Lower. \n")} if (g<x) {cat("Higher. \n")} if (g==x) {cat("Good!");break}} </pre>	<pre> > game() 1 th Guess? 1: 50 Higher. 2 th Guess? 1: 75 Lower. 3 th Guess? 1: 62 Lower. 4 th Guess? 1: 56 Higher. 5 th Guess? 1: 59 Higher. 6 th Guess? 1: 61 Lower. 7 th Guess? 1: 60 Good!> </pre>	

- > 함수 game() 스크립트를 작성한 후 스크립트 전체를 선택한 후 CTRL+R 로 실행한다. 실행된 스크립트는 콘솔에 나타나고 오류 메시지가 있는지 확인한다.
- > 오류 메시지가 없으면 콘솔에서 함수 game()을 치면 스크립트가 실행된다.
- > runif(1,0,1); (0, 1) 균일분포 따르는 데이터 하나를 임의 생성한다.
- > 함수 round(); 정수의 반올림, 그러므로 0 과 1 사이의 값이 x 에 저장된다.
- > for() 제어문; i 는 1 부터 10 까지 증가, 게임은 10 번까지 할 수 있음
- > 함수 cat('문장'); 화면에 문장이 출력된다.
- > scan(); 키보드 입력 값을 하나 (nmax=1) 읽어들이고 후 g 에 저장한다.
- > if(조건) {문장}; 조건이 맞으면 문장이 실행된다.
- > break; 스크립트 실행이 중단된다.

+ 모수 있는 함수

EXAMPLE

확률밀도함수 그리기 함수

```

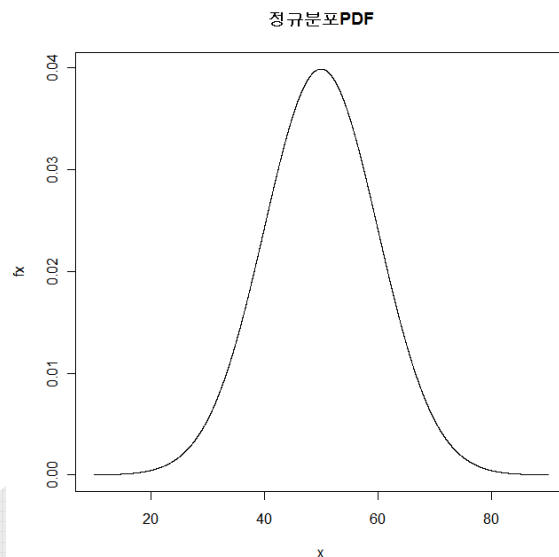
R I:\Books\WR\활용\wch1.R - R 편집기
#정규분포함수 그리기
ng=function(m,s){
  max=m+4*s
  min=m-4*s
  x=seq(min,max,0.01)
  fx=dnorm(x,m,s)
  plot(x,fx,main='정규분포PDF',
  type='l')}

```

```

> ng(0,1)
> ng(50,10)

```



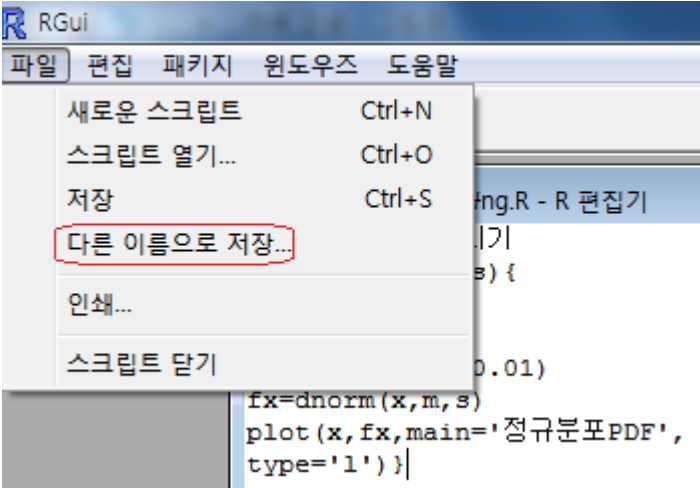
> ng(0,1); 평균 0, 표준편차 1 인 정규분포가 그려진다.

> min, max; 확률변수의 최소값과 최대값 지정, 정규분포의 경우 $\pm 3\sigma$ 영역에 99.9% 확률, 그러므로 여유를 주는 4σ 사용

> dnorm(x,m,s); 평균이 m, 표준편차가 s 인 정규분포의 x에서의 함수값

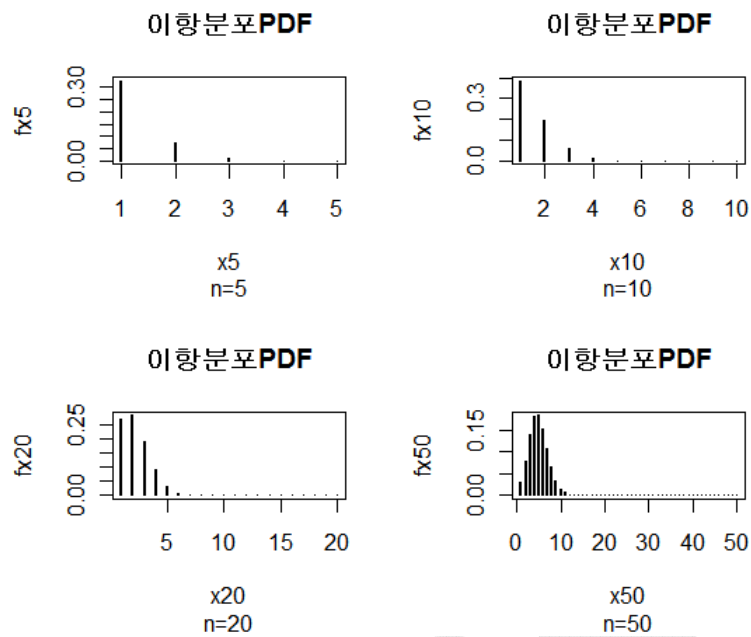
$$dnorm(x,m,s) = f(x;m,s) = \frac{1}{\sqrt{2\pi}s} \exp\left(-\frac{(x-m)^2}{2s^2}\right)$$

+ 함수 저장하기

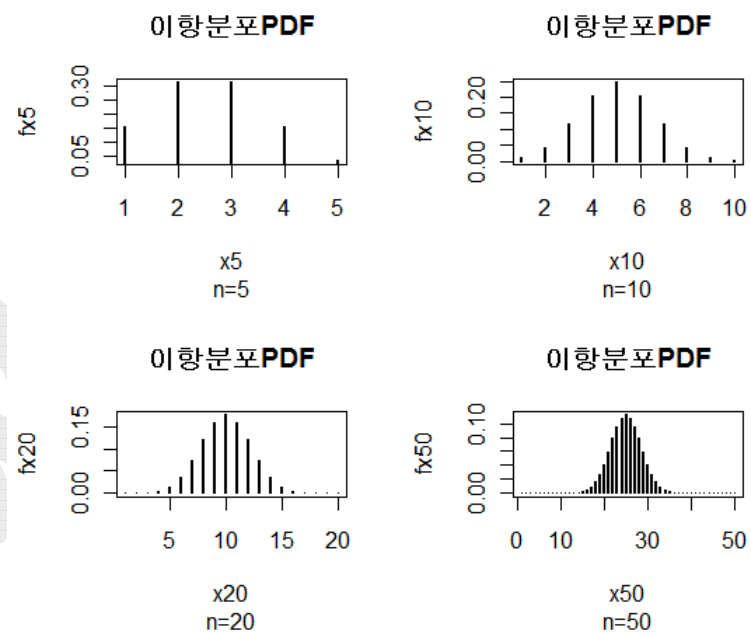
EXAMPLE		함수 저장
		<p>> 스크립트를 저장한다. 예를 들어 작업 디렉터리에 'ng.R'로 저장하였다고 하자.</p>
<pre>> source('ng.R') > ng(3,2) > ng(4,1)</pre>	<p>> R을 실행한 후 함수 source()를 이용하여 사용자가 작성한 함수를 불러온다.</p> <p>> 함수 형식에 맞게 사용하면 된다.</p>	

EXERCISE
<p>이항분포의 정규분포 근사를 보이는 함수 bin2nor(p)를 만드시오. n=5, 10, 20, 50 으로 하여 4개 이항 확률밀도함수가 동일한 그래프에 그려지게 하시오.</p> <pre>> bin2nor(0.1) > bin2nor(0.5)</pre>

이항분포 $B(n, p=0.1)$



이항분포 $B(n, p=0.5)$



1.5.6 에러 메시지

R 은 사용자의 작업 명령에 오류가 발생하면 사용자에게 오류 메시지를 즉시 알려주므로 에러 메시지를 잘 이해하면 프로그램 작업이 용이하다. 다음은 CSV 형식 엑셀 외부 데이터를 읽어 들이는 경우 스크립트에서 발생하는 오류 메시지에 대한 예이다.

```
(1) > ds=read.xls(data.csv)
에러:함수 "read.xls"를 검색해낼수가 없었습니다
> ds=read.xls(data.csv)
에러:함수 "read.xls"를 검색해낼수가 없었습니다
(2) > ds=read.csv(data.csv)
이하에 에러read.table(file = file, header = header, sep =
오브젝트 'data.csv'가 없습니다
(3) > ds=read.csv("data.csv")
이하에 에러file(file, "rt") : 연결로부터 열수가 없습니다
추가정보:Warning message:
In file(file, "rt") :
cannot open file 'data.csv': No such file or directory
```

- (1) 사용한 함수 read.xls()가 R 에 없거나 관련 패키지가 (xlsReadWrite) 설치되지 않은 경우 표시되는 오류 ⇨(해결책) help() 기능을 이용하여 함수가 있는지 확인하거나 관련 패키지를 찾아 설치한다.
- (2) 함수 read.csv() 안의 옵션 설정이 잘못되어 나타나는 오류 ⇨(해결책) 읽어 들일 파일 이름을 지정할 때는 “”, 혹은 ‘’을 사용한다.
- (3) 지정된 폴더에 읽어 들일 파일이 없는 경우 나타나는 오류 ⇨(해결책) **파일** 메뉴에서 **디렉토리 변경...** 메뉴를 선택하여 데이터 파일이 있는 폴더를 지정한다.