

Kyun-Seop Bae, Sungpil Han

R Programming - Lecture Notes



R

PROGRAMMING

KYUN-SEOP BAE
SUNGPIIL HAN

Contents

List of Tables	v
List of Figures	vii
Preface	ix
1 R language	1
2 Graphics	3
2.1 Introduction	3
2.2 상위수준 그림 함수	3
2.2.1 상위수준 그림 함수의 주요 인자 (arguments)	3
2.2.2 scatter plot	4
2.2.3 Histogram	7
2.2.4 Box-Whisker Plot	11
2.2.5 Bar Plot	13
2.2.6 pie chart	17
2.2.7 matplot 함수	18
2.2.8 Scatter plot matrices (pairs plots)	19
2.3 하위수준 그림 함수	21
2.3.1 점, 선, 설명 추가 하기 {add}	21
2.3.2 polygon 함수	23
2.4 그림 출력하기	24
2.4.1 pdf graphics devices	24
2.4.2 PNG graphics devices	25
3 Data Import / Export	27
3.1 Read.csv	27
3.2 Theoph 데이터	27
3.3 lattice	28
3.4 Subsetting and write.csv	31
4 Frequently Used Functions	39
4.1 Command	39
4.2 The basics	67
4.3 Common data structures	67
4.4 Statistics	67

4.5	Working with R	67
4.6	I/O	67
5	Rstudio and some useful packages 2 - ggplot2	79
5.1	Introduction	79
Appendix		85
A	As-is R Files	85
A.1	Lecture 3	85
A.2	Lecture 4	91
A.3	Lecture 5	92
B	R Tips	99
B.1	Using Coursera	99
C	Acknowledgement	101
Bibliography		103
Index		105

List of Tables

4.1	The basics - The first functions to learn	66
4.2	The basics - Important operators and assignment	66
4.3	The basics - Comparison	67
4.4	The basics - Basic math	68
4.5	The basics - Functions to do with functions	69
4.6	The basics - Logical - sets	69
4.7	The basics - Vectors and matrices	70
4.8	The basics - Making vectors	70
4.9	The basics - Lists - data.frames	71
4.10	The basics - Control flow	71
4.11	The basics - Apply - friends	71
4.12	Common data structures - Date time	72
4.13	Common data structures - Character manipulation	72
4.14	Common data structures - Factors	72
4.15	Common data structures - Array manipulation	73
4.16	Statistics - Ordering and tabulating	73
4.17	Statistics - Linear models	74
4.18	Statistics - Miscellaneous tests	74
4.19	Statistics - Random variables	74
4.20	Statistics - Matrix algebra	75
4.21	Working with R - Workspace	75
4.22	Working with R - Help	75
4.23	Working with R - Debugging	76
4.24	I/O - Output	76
4.25	I/O - Reading and writing data	77
4.26	I/O - Files and directories	77



List of Figures

1	Creative Commons License	x
2	Syllabus page 1	xiv
3	Syllabus page 2	xv



Preface

안녕하십니까?

2017년 1학기 울산대학교 의학과 대학원 수업 R Programming 과목 담당교수 배균섭입니다.

R은 <http://cran.r-project.org> 에서 다운로드받아 설치할 수 있습니다. 역시 같은 사이트에서 Manual이 나와 있으니 참고하시기 바랍니다. 구글에서 'R Programming pdf' 와 같은 키워드로 검색하시면 많은 자료를 보실 수 있습니다.

첨부한 R.stx¹ 파일은 AcroEdit이라는 editor에서 사용할 syntax highlighting용 구문 파일입니다. <http://www.acrosoft.pe.kr> 에서 다운로드 받아 설치하시기 바랍니다. AcroEdit 대신 notepad++를 선호하시는 분은 그대로 사용하셔도 됩니다.

저는 RStudio, tinnR 등을 이용해서 강의하지 않습디만, 필요하신 분은 쓰셔도 괜찮습니다. 향후 R package 작성을 위해서는 MiKTeX와 Rtools를 설치하십시오.

추가로 말씀드리자면, <http://www.coursera.org> 에 많은 R 강좌가 개설되어 있습니다. Specialization course로 들어가면 유료이지만, (Specialization course는 여러 개의 과목이 합쳐져 있는 것입니다.) 개별 과목을 검색해서 들어가면, 무료로도 볼 수 있습니다. (대신 시험을 칠 수 없거나, certificate를 받을 수 없습니다.)

좋은 강좌가 많으니 많이 활용하시기 바랍니다.

강의 장소에 불편함이 많은 것으로 생각되어, 다음과 같이 Skype 모임을 개설하였습니다. 사정상 원거리에서 오시기 불편한 분들은 활용하시기 바랍니다. 출석은 화면을 캡처하거나 휴대폰으로 찍은 뒤 sec@acp.kr², shan@acp.kr³ 보내주시면 출석으로 인정해드립니다.

Skype 모임 참가 <https://meet.lync.com/uucp-acp/ksbae/SKGJ3BNQ>

2017년 3월, 배균섭 배상

The online version of this book is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License⁴.

¹<https://groups.google.com/a/acp.kr/group/r/attach/409db97bf453a/R.stx?part=0.1&authuser=0>

²<mailto:sec@acp.kr>

³<mailto:shan@acp.kr>

⁴<http://creativecommons.org/licenses/by-nc-sa/4.0/>



FIGURE 1: Creative Commons License

Teaching Assistant

안녕하십니까? 서울아산병원 임상약리학과 전공의 한성필입니다. 수업과 관련된 여러 제반 업무를 담당하고 있습니다. 언제든지 의문사항 있으면 r@acr.kr⁵로 전체 메일 보내 시거나 교수님 k@acr.kr⁶ 혹은 제 개인 메일 shan@acp.kr⁷ 연락해 주십시오.

교수님께서 세우신 방침에 따라 수업시간에 출석을 부르지 않을 예정입니다. 수강하시는 화면 (Skype)을 휴대폰으로 사진 찍으시거나 강의실의 스크린을 사진으로 촬영하셔서 sec@acp.kr⁸ / shan@acp.kr⁹로 동시에 보내주시면 됩니다. 가급적 “2017-03-31 한성필 출석”과 같은 식의 제목을 유지해 주시면 처리하는데 큰 도움이 될 것 같습니다.

출석 체크를 위해 전체메일을 사용하지 말아주십시오!

아울러 수업 중에 사용한 코드/스크립트를 사용하여 R의 패키지인 bookdown을 사용해 웹북을 제작 중에 있습니다. (Xie, 2016) 여러분이 읽고 있는 이 책 자체가 R 코드의 일종인 Rmarkdown의 결과물이라고 보시면 됩니다. Github 저장소¹⁰가 있으니 소스 코드를 보실 수 있습니다. 누구나 소스를 편집하여 Pull Request를 요청할 수 있으므로 혹시 Github를 사용하셔서 웹북의 질을 높이하고자 하시는 수강생 선생님들께서는 도움을 주십시오.

감사합니다.

2017년 3월, 한성필 올림

FAQ

접속 관련

⁵<mailto:r@acr.kr>

⁶<mailto:k@acr.kr>

⁷<mailto:shan@acp.kr>

⁸<mailto:sec@acp.kr>

⁹<mailto:shan@acp.kr>

¹⁰<https://github.com/asancpt/Rprogramming>

Q. 스카이프를 한번도 안써봐서 이참에 사용법을 배우고있는데, 수업시작 시에 상대방을 어떻게 검색해서 들어가면 될지 알려주시면 감사하겠습니다.

Q. 온라인 수강시 접속하는 스카이프 주소는 무엇인지요?

<https://meet.lync.com/uucp-acp/ksbae/SKGJ3BNQ>

Chrome 등 웹브라우저에서 위 주소를 입력하면 직접 대화방으로 연결됩니다. (검색할 필요 없습니다.) 처음 설치시에는 Add-on 이 설치될 수 있습니다. MacOS Sierra, Win7, Win10에서 Chrome, Internet Explorer 등을 사용하여 테스트해 보았고 모두 잘 동작하였습니다. 대부분의 경우 Skype For Business 계정이 없을 것으로 생각되는데 따로 로그인할 필요 없습니다.

수업 시작 30분 전부터 대화방을 개설해 놓도록 하겠습니다.

https://groups.google.com/a/acr.kr/d/msg/r/nUkrE37W2kQ/waG-FkM_BgAJ 교수님께서 처음 보낸 메일을 참고해 주십시오.

Q. 앞으로 수업은 지난 첫수업처럼 계속 온라인 수강이 가능한 것인가요?

네, 계속 온라인으로 가능합니다.

Q. 저도 웹캠을 설치하여야 하여야 하나요?

설치할 필요 없습니다. 오히려 수강자의 웹캠의 전원을 꺼두시길 권고드립니다.

Q. 수강전 온라인 강의 테스트 해볼 수 있나요?

수업 시작 30분 전부터 대화방을 개설하여 놓도록 하겠습니다.

출석관련

Q. 미국학회 참석으로 수업시간이 귀국행 비행기 기내에 있을거같아 출석이 안될것 같습니다. 방법이 있을까요?

결석 사유서를 제출해 주시면 출석 처리 하겠습니다. 대학원 홈페이지 참고 바랍니다.¹¹ 이 링크로 들어가시면 가장 위에 있습니다. (결석사유서.hwp) 참고로 수업 영상은 녹화하여 Youtube에 비공개 링크를 만들 예정이라서 추후에 관련 영상을 시청할 수 있을 것 같습니다. 결석사유서를 제출한다고 100% 출석이 인정되는 것은 아닙니다. 이것이 기본적으로는 offline강의이기 때문에 강의시간에 강의실에 있든지, 또는 온라인으로 접속해 있어야 합니다. 출석사유서를 제출하거나, 추후 동영상 시청을 해서 그 증거(사진)를 제출하는 경우에 감점을 줄어드릴 수 있습니다. 예를 들어, 결석시에는 2점 감점인데, 결석사유서를 제출하면 1점만 감점한다는지, 동영상을 보면 0.5점만 감점한다는지 하는 것입니다. 결석 사유서 제출 시 출석 처리 원칙에 대한 설명을 드리오니, 참고하시길 바랍니다.

과제 관련

Q. 과제물이 있다고 들었는데 언제 assign 하게 되는지요?

과제물은 빨라야 5주차 이후에 나갑니다.

¹¹<http://www.medulsan.ac.kr/graduate/?mid=72&curpage=files>

Coursera 관련

Q. 첫 수업 때, certification 관련 말씀을 하셨는데, 정확히 coursera 사이트에서 어떤 것을 듣고, 제출을 해야하는지 궁금합니다. (비슷한 내용이 많아, 어떤것을 들어야하는지 헷갈립니다.)

Coursera는 꼭 어느 것을 들어야 하는 것은 아니고, R programming과 관련된 것이라면 자유로이 골라서 들으면 됩니다. 대표적인 두 가지만 들자면 다음과 같습니다.

- <https://www.coursera.org/learn/r-programming>
 - <https://www.coursera.org/learn/r-programming-environment>
-

Q. Coursera 강의를 듣고 증명서를 내면 출석을 얼마나 커버할 수 있을런지요?

Coursera는 출석 커버보다는 grade를 올려 주기 위한 것입니다. 출석은 Skype로 커버해야 합니다. 출석의 성적 반영비율은 25%이지만, 규정상 4회 이상 결석이면 성적이 나갈 수 없습니다.

Syllabus

2017-1학기 수업계획서(Course Outline)

년도-학기 (year-semester)	2017-1	과목명 (course name)	R 프로그래밍 R Programming					
과목번호-분반 (course-no-class)	W45493 - 01	학점(강의-실습) (credit)	3학점(3-0)	ABEEK(설계학점) (Abeek credit)				
담당교수 (professor)	배근섭 Bae, Kyun Seop	연구실 번호 (office phone)	02-3010-4611	학부(과)사무실 (Dept.office phone)	02-3010-4217			
개설학과-학년 (department-year)	의학과			이수구분 (type of course requirement)	공통 common			
E-MAIL	ksbae@amc.seoul.kr	강좌구분 (type of lecture)	일반강좌 general lecture					
홈페이지 (Homepage)		성적평가방법(method of grade evaluation)	절대평가 / 등급 absolute evaluation					
1.교과목 개요(course description)								
Data Science의 가장 기본적인 tool인 R로 어떻게 프로그래밍을 하는지에 기술을 익힌다. 자신의 세부 전공분야에 상관없이 적용할 수 있는 공통적인 부분을 중심으로 학습할 것이다. Students will learn how to program using R which is the basic tool for the data science. The subjects will be common ones regardless of their specific majors.								
2.교수목표(goal of instruction)								
1.R을 이용하여 자신의 문제를 해결할 수 있다. 2.R package를 개발하여 다른 사람의 문제 해결을 돕는다. 3.정답이 알려지지 않은 문제에 대하여, 체계적인 시행착오로 최적의 해답을 찾는 습관을 들인다. 4.자신이 가진 Tool들의 한계와 장단점을 이해하고 자신의 문제에 적용한다.								
3.주요 학습내용 및 수업진행방법(main contents & methods of teaching)								
강의, 동영상 시청								
4.학습 성과 평가방법(evaluation criteria)								
과제 (중간, 기말고사 대신 과제를 제출해야 하며, 중간, 기말고사 기간에도 강의가 있습니다.)								
평가항목(evaluation)	출석 (attendance)	중간고사(mid-term exam)	기말고사 (final exam)	리포트 (report)	발표(presentation)	퀴즈 (quiz)	Term Project	기타 (etc.)
점수비율 (percentage)	25.00	25.00	25.00	25.00	0.00	0.00	0.00	0.00
5.교재 및 참고 문헌(textbook & reference books)								
1. [부교재] [도서판] (데이터 고급 분석과 통계 프로그래밍을 위한) 벡터데이터 분석 도구 R 프로그래밍 노란 데브토프 지: 홍길 의왕: 에이콘 9788960773332: 9788960772793(set) 2. [부교재] [도서판] R Cookbook: 데이터 분석과 통계, 그래프를 위한 실전 예제 풀 터서:이재원 홍길 서울: 인/ 9788966280379 3. [부교재] [수기입력] Software for Data Analysis Chambers JM Springer 4. [부교재] [수기입력] Advanced R Wickham H CRC Press 5. [부교재] [도서판] [ebook]The Basics of S-PLUS [electronic resource] Krause A, Olson M New York, NY: S, New York 9780387283906 6. [부교재] [도서판] [ebook]Introduction to Scientific Programming and Simulation Using R, Second Edition [electronic resource] Jones O, Mailardet R Hoboken: CRC Press 9781466570016 7. [부교재] [수기입력] R Programming for Data Science Peng R, lulu.com 8. [주교재] [수기입력] The R Manuals R Core Team http://cran.r-project.org								
6.주별 진도계획 및 학습자료(weekly plan & study materials)								
제1주(week 1)	[주별진도(topic)] Course introduction: textbooks, resources, installation, ...					[학습자료(materials)] R-admin		
제2주(week 2)	[주별진도(topic)] Data objects: basic types, vector, matrix, list, data frame					[학습자료(materials)] R-intro		
제3주(week 3)	[주별진도(topic)] Plotting and graphics					[학습자료(materials)] Handout		
제4주(week 4)	[주별진도(topic)] Data manipulation: loading, subsetting, merging, saving					[학습자료(materials)] R-data		

울산대학교 UWIN

1/1

FIGURE 2: Syllabus page 1

제5주 (week 5)	[주별진도(topic)] Basic commands and functions	[학습자료(materials)] R-lang
제6주 (week 6)	[주별진도(topic)] Elements of programming style	[학습자료(materials)]
제7주 (week 7)	[주별진도(topic)] Developing R package without Rstudio	[학습자료(materials)] R-exts
제8주 (week 8)	[주별진도(topic)] Some useful packages 1 ? RODBC, rtf	[학습자료(materials)]
제9주 (week 9)	[주별진도(topic)] Rstudio and some useful packages 2 ? ggplot2	[학습자료(materials)]
제10주 (week 10)	[주별진도(topic)] (책가단신원 휴무)	[학습자료(materials)]
제11주 (week 11)	[주별진도(topic)] Some useful packages 3 ? dplyr, tidyr	[학습자료(materials)]
제12주 (week 12)	[주별진도(topic)] Some useful packages 4	[학습자료(materials)]
제13주 (week 13)	[주별진도(topic)] Handling date, time, and string	[학습자료(materials)]
제14주 (week 14)	[주별진도(topic)] Functional and object-oriented programming in R	[학습자료(materials)]
제15주 (week 15)	[주별진도(topic)] 보강주	[학습자료(materials)]
제16주 (week 16)	[주별진도(topic)] Pitfalls and limitations of R	[학습자료(materials)]

FIGURE 3: Syllabus page 2



1

R language

2017-03-15 배균섭 교수님 강의

R Language Definition¹의 초반 내용에 대해 설명하였습니다.

¹<https://cran.r-project.org/doc/manuals/r-release/R-lang.pdf>



2

Graphics

2017-03-22 임형석 교수님 강의

R을 사용해 그림 그리는 방법에 대해 알아보겠습니다.

2.1 Introduction

- 상위수준 그림 함수는 그림을 생성한다.
 - 하위수준 그림 함수는 기존의 그림에 그림을 추가한다.
-

2.2 상위수준 그림 함수

2.2.1 상위수준 그림 함수의 주요 인자 (arguments)

- main: 제목
- xlab/ylab: x축 및 y축 레이블
- xlim/ylim: x축 및 y축 범위
- col: 색깔
- lty: 선 모양
- pch: 점 모양
- cex: 그림 성분의 크기
- lwd: 선 굵기
- type: 그림 타입

```
dta <- read.csv("PK.csv")
head(dta)
```

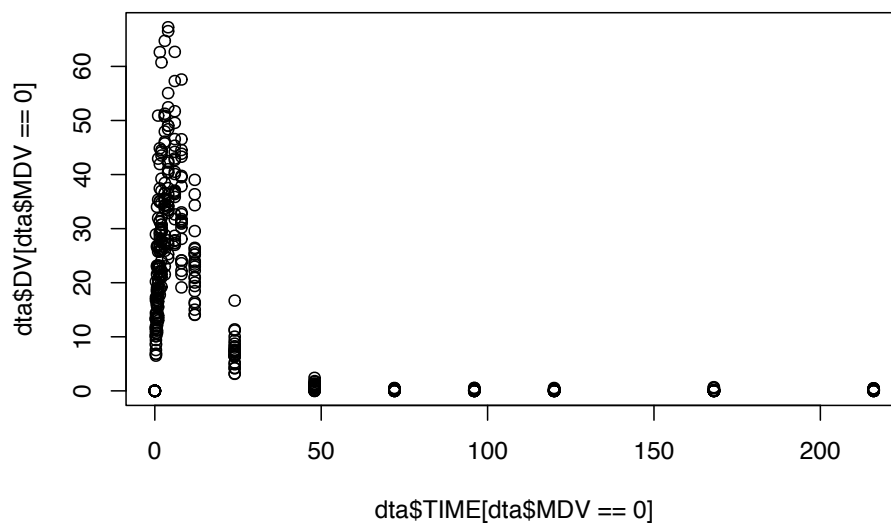
```
##   ID TIME AMT   DV MDV
## 1  1 0.00  0 0.00  0
## 2  1 0.00  4 0.00  1
## 3  1 0.33  0 9.40  0
## 4  1 0.66  0 13.71 0
## 5  1 1.00  0 16.52 0
## 6  1 1.50  0 29.36 0
```

```
str(dta)
```

```
## 'data.frame':   456 obs. of  5 variables:
##  $ ID   : num  1 1 1 1 1 1 1 1 1 ...
##  $ TIME : num  0 0 0.33 0.66 1 1.5 2 3 4 6 ...
##  $ AMT  : num  0 4 0 0 0 0 0 0 0 ...
##  $ DV   : num  0 0 9.4 13.7 16.5 ...
##  $ MDV  : num  0 1 0 0 0 0 0 0 0 ...
```

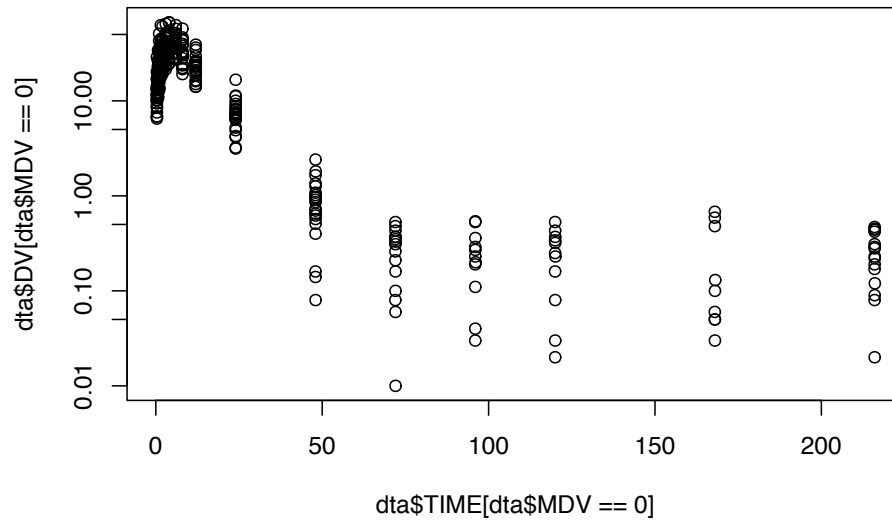
2.2.2 scatter plot

```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0])
```

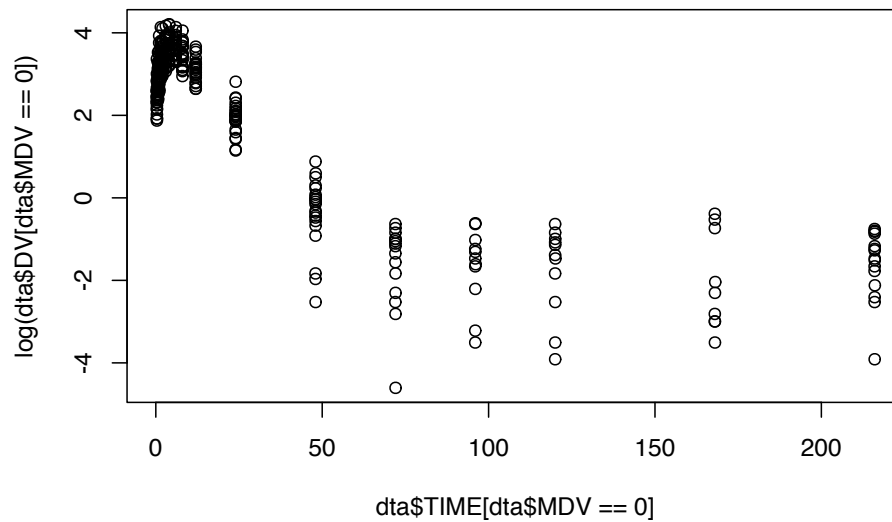


```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], log="y")
```

```
## Warning in xy.coords(x, y, xlabel, ylabel, log): 86 y
## values <= 0 omitted from logarithmic plot
```



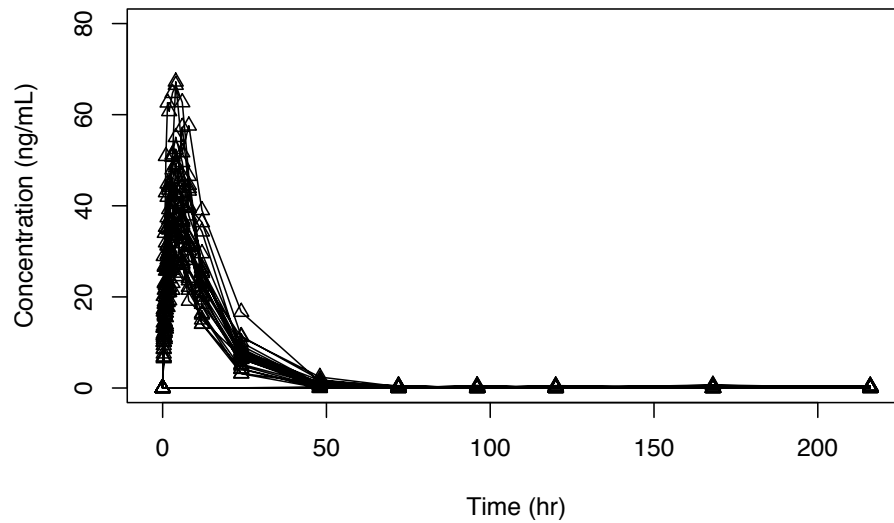
```
plot(dta$TIME[dta$MDV==0], log(dta$DV[dta$MDV==0]))
```



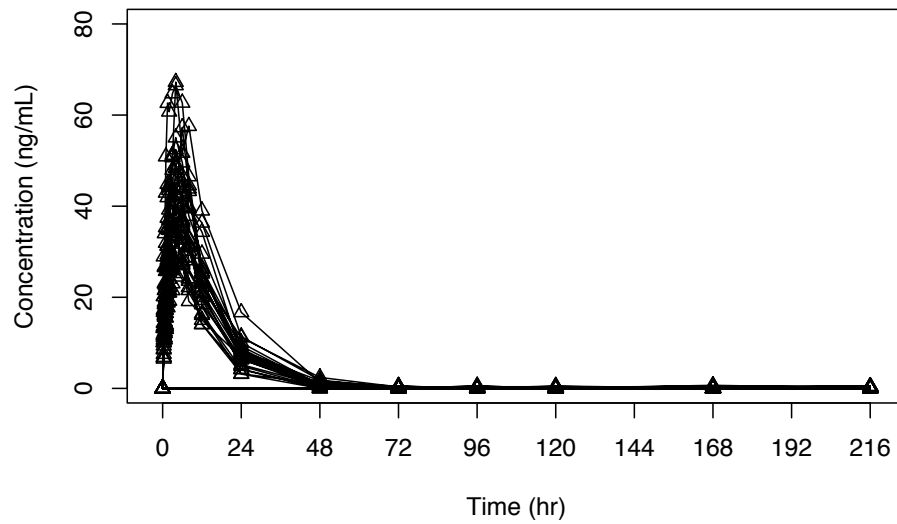
```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0],
     , xlabel="Time (hr)", ylabel="Concentration (ng/mL)")
```

```
, type="o", pch=2, col=1, main="PK time-course of Drug X"  
, xlim =c(-2,218), ylim=c(0,80))
```

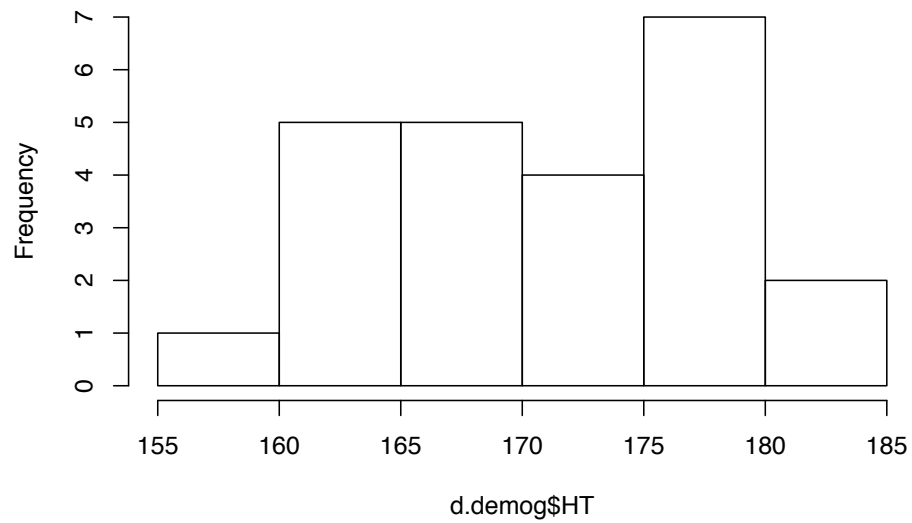
PK time-course of Drug X



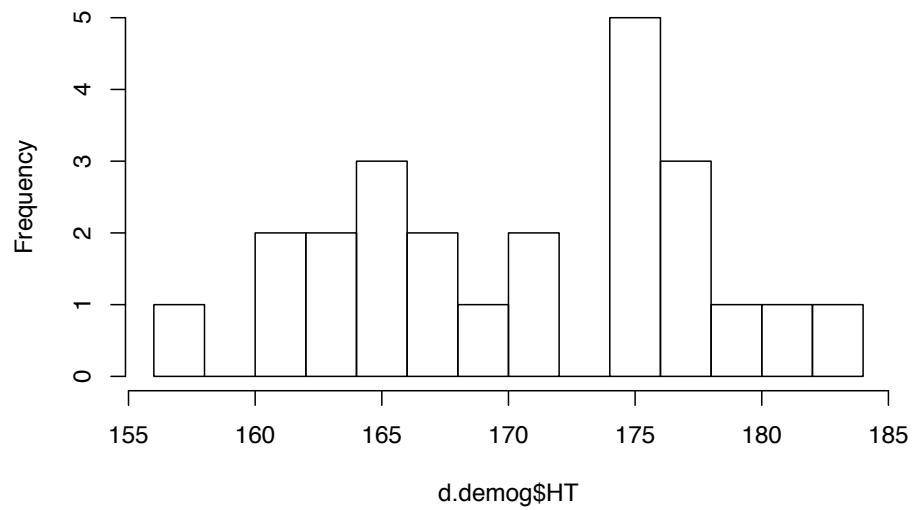
```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], axes=F,  
      , xlab="Time (hr)", ylab="Concentration (ng/mL)"  
      , type="o", pch=2, col=1, main="PK time-course of Drug X"  
      , xlim =c(-2,218), ylim=c(0,80))  
axis(1, at=seq(0, 218, 24))  
axis(2)  
box()
```

PK time-course of Drug X**2.2.3 Histogram**

```
d.demog <- read.csv("DEMOG.csv")  
  
hist(d.demog$HT)
```

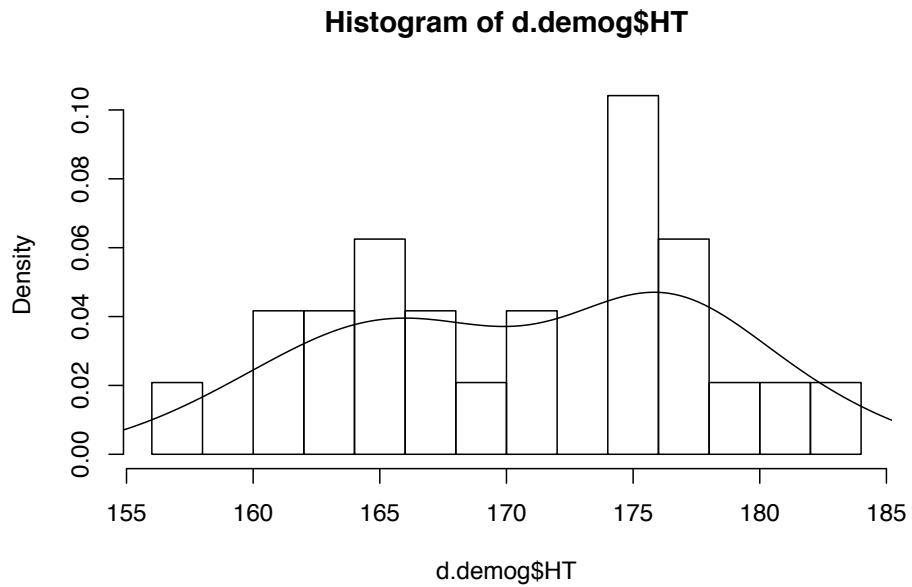
Histogram of d.demog\$HT

```
hist(d.demog$HT, breaks=10)  
hist(d.demog$HT, nclass=10)
```

Histogram of d.demog\$HT

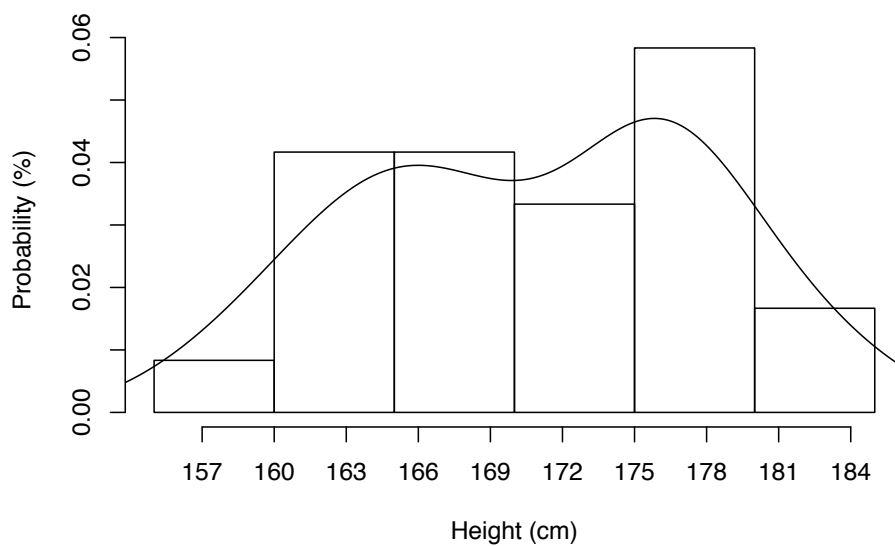
2.2.3.1 with density line

```
hist(d.demog$HT, probability=TRUE, breaks=10)  
lines(density(d.demog$HT))
```



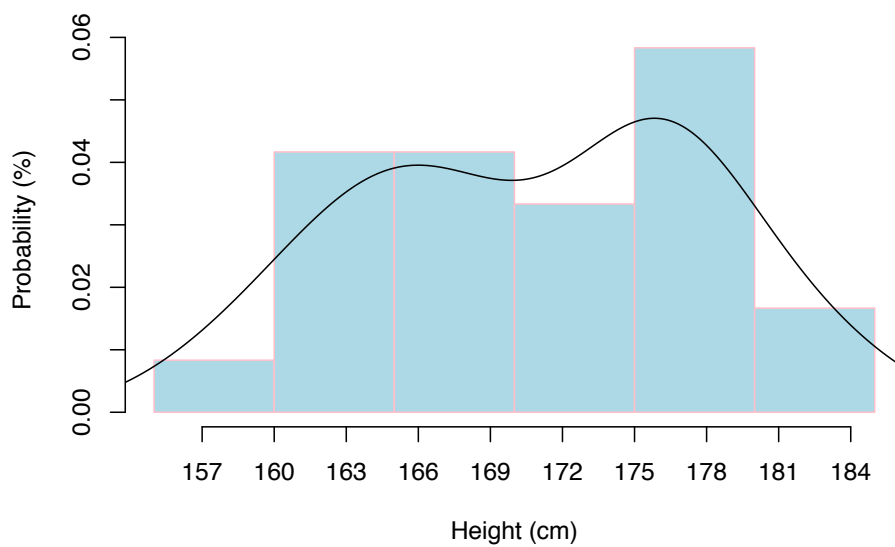
```
hist(d.demog$HT, probability=TRUE, breaks=9, xaxt="n"  
      , main="Histogram for Height", xlab="Height (cm)", ylab="Probability (%)")  
axis(1, at=seq(min(d.demog$HT), max(d.demog$HT), 3))  
lines(density(d.demog$HT))
```

Histogram for Height



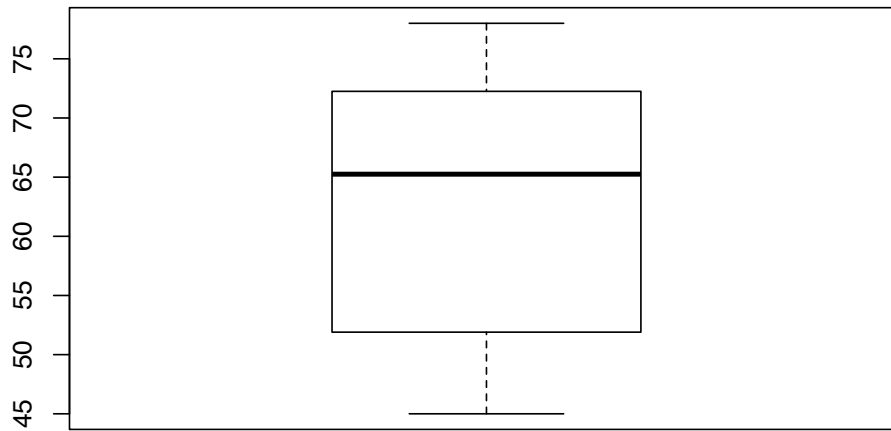
```
hist(d.demog$HT, probability=TRUE, breaks=9, xaxt="n"
     , main="Histogram for Height", xlab="Height (cm)", ylab="Probability (%)")
     , col = "lightblue", border = "pink")
axis(1, at=seq(min(d.demog$HT), max(d.demog$HT), 3))
lines(density(d.demog$HT))
```

Histogram for Height



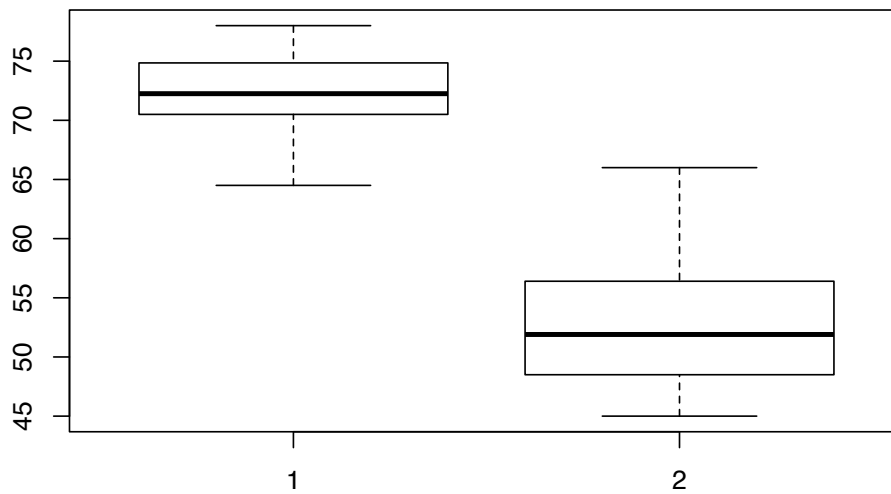
2.2.4 Box-Whisker Plot

```
boxplot(d.demog$WT)
```



```
boxplot(d.demog$WT ~ d.demog$SEX)
```

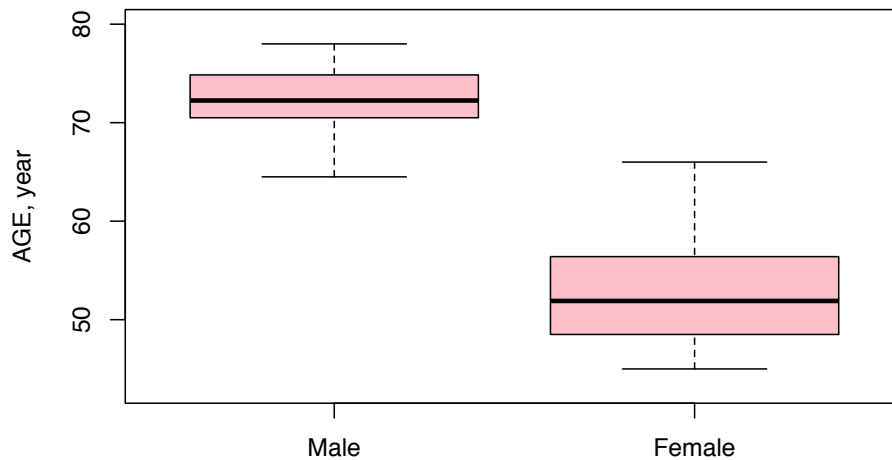
```
boxplot(split(d.demog$WT, d.demog$SEX))
```



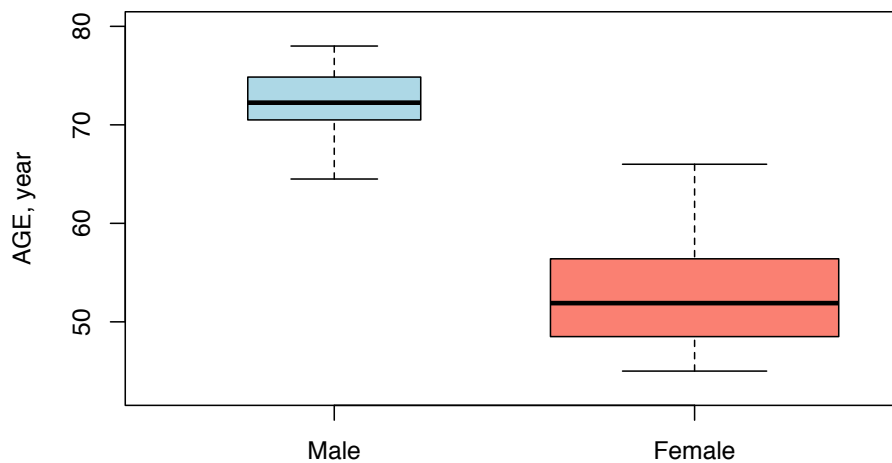
```
boxplot(WT ~ SEX, data=d.demog)
```

```
boxplot(d.demog$WT ~ d.demog$SEX)
```

```
, names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
, col="pink")
```

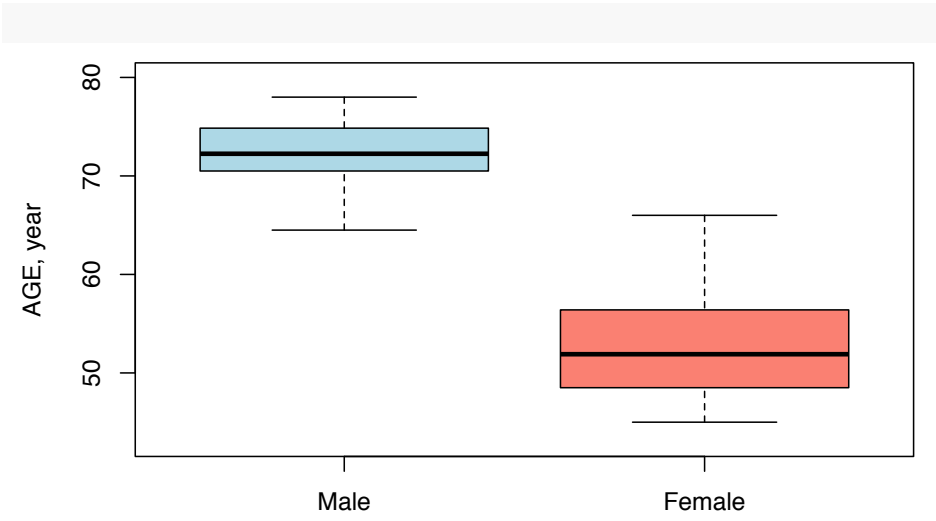


```
boxplot(d.demog$WT ~ d.demog$SEX
, names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
, col=c("lightblue", "salmon"), width=c(0.6, 1))
```

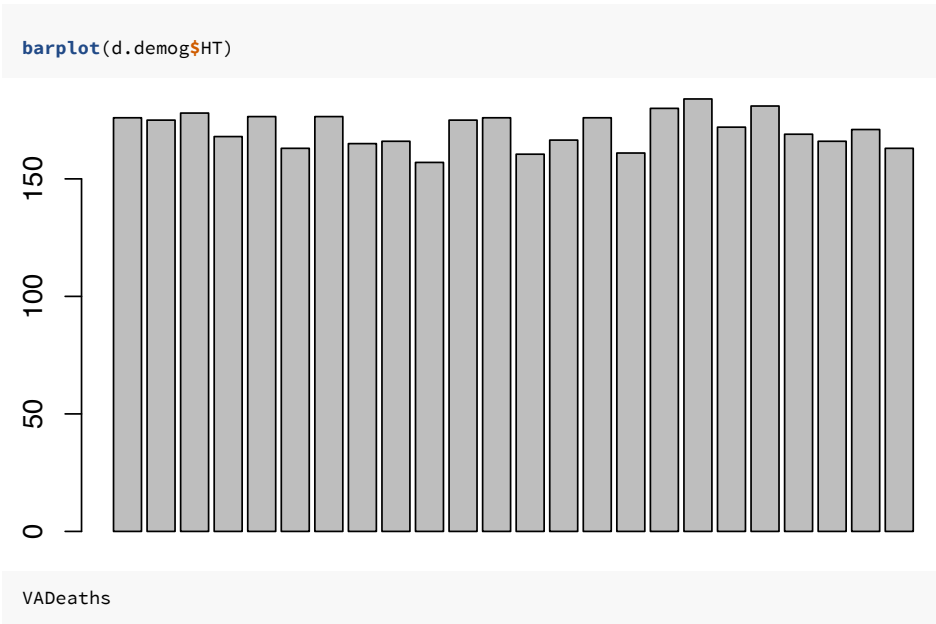


-varwidth: if varwidth is TRUE, the boxes are drawn with widths proportional to the square-roots of the number of observations in the groups.

```
boxplot(d.demog$WT ~ d.demog$SEX
, names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
, col=c("lightblue", "salmon")
, varwidth=TRUE)
```



2.2.5 Bar Plot

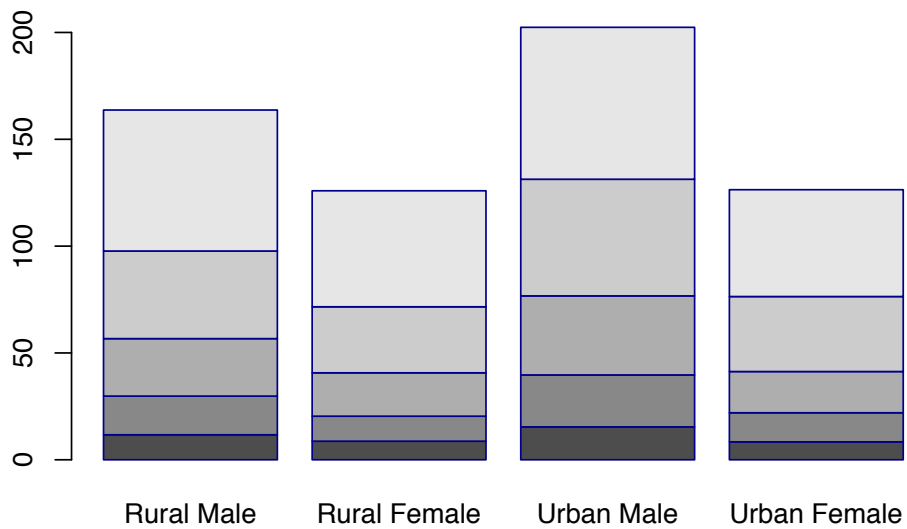


VADeaths

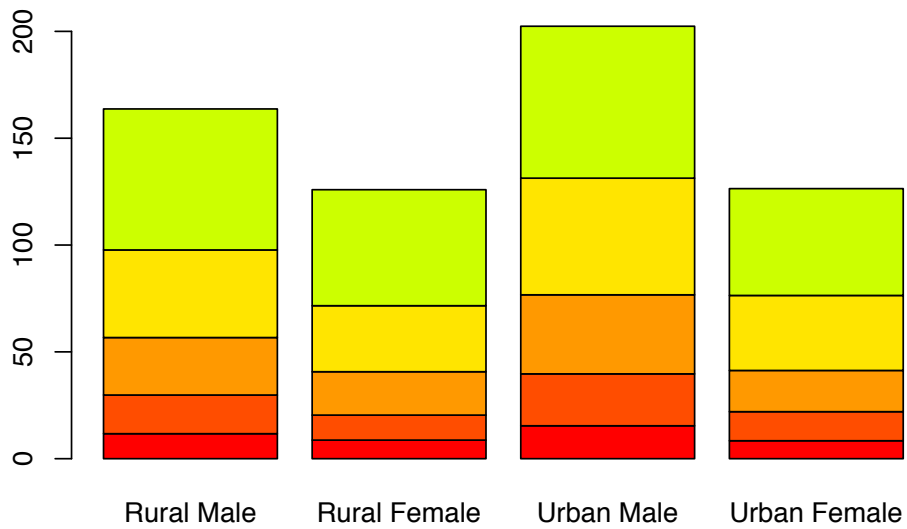
##	Rural Male	Rural Female	Urban Male	Urban Female
## 50-54	11.7	8.7	15.4	8.4
## 55-59	18.1	11.7	24.3	13.6
## 60-64	26.9	20.3	37.0	19.3

## 65-69	41.0	30.9	54.6	35.1
## 70-74	66.0	54.3	71.1	50.0

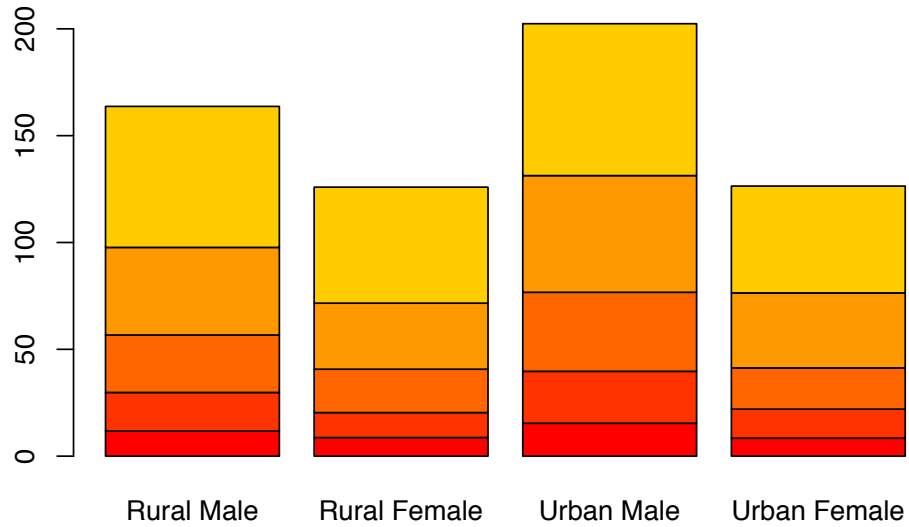
```
barplot(VADeaths, border = "dark blue")
```



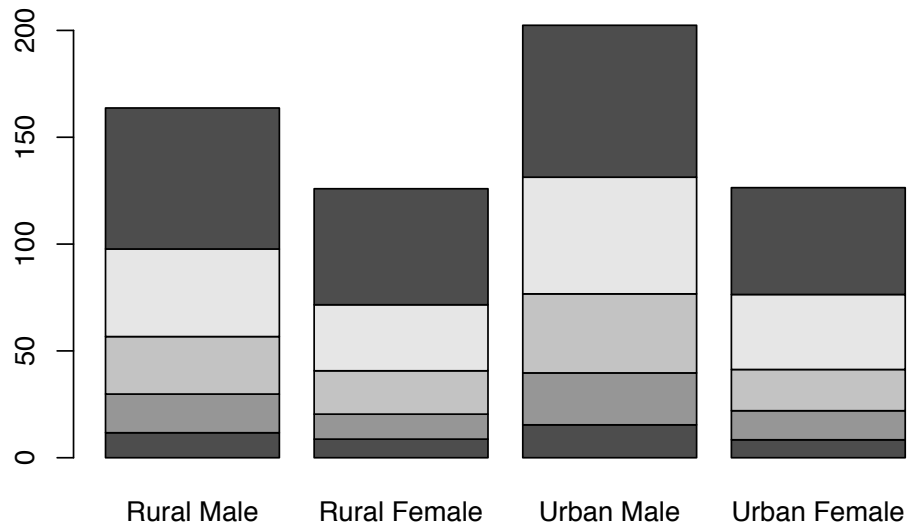
```
barplot(VADeaths, col = rainbow(20))
```



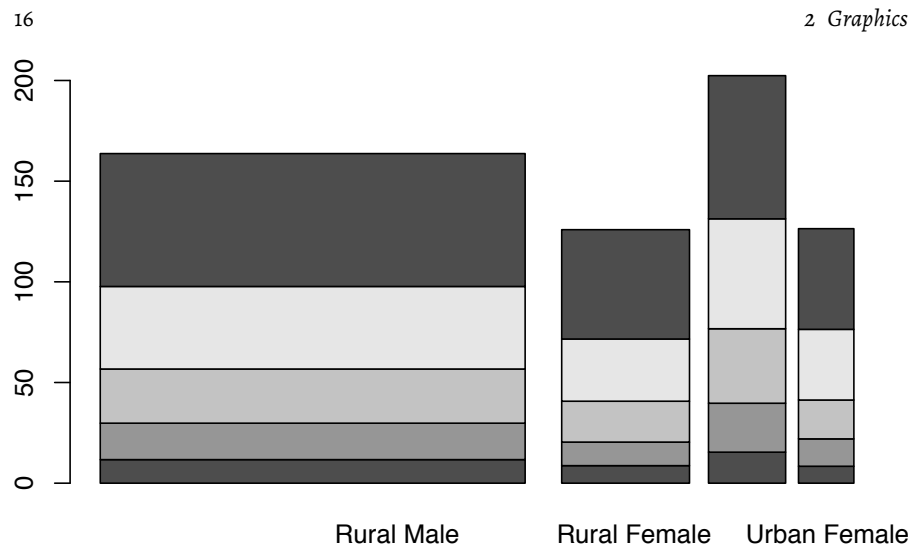
```
barplot(VADeaths, col = heat.colors(8))
```



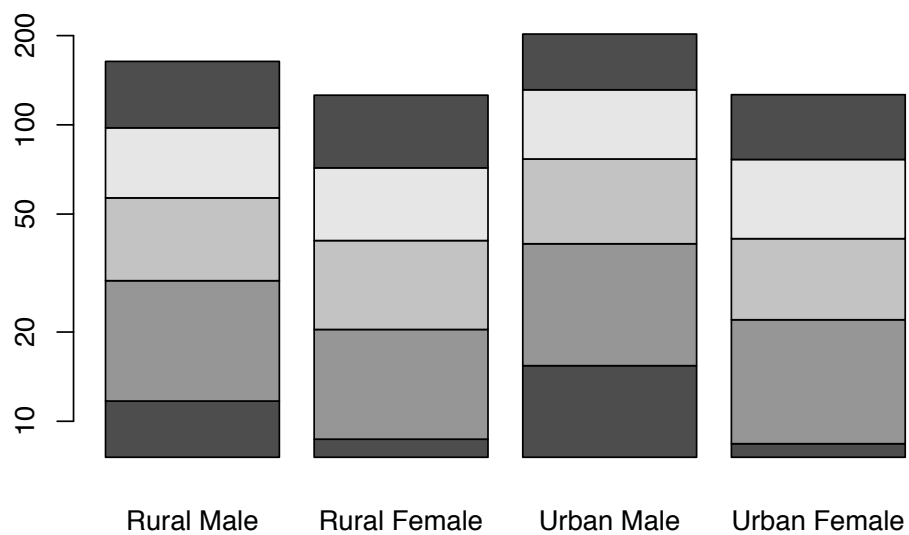
```
barplot(VADeaths, col = gray.colors(4))
```



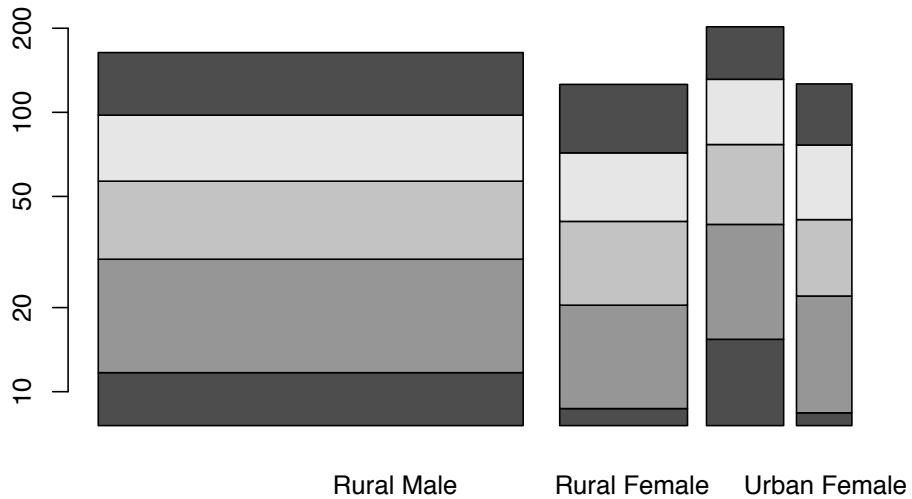
```
barplot(VADeaths, col = gray.colors(4), log="x")
```



```
barplot(VADeaths, col = gray.colors(4), log="y")
```

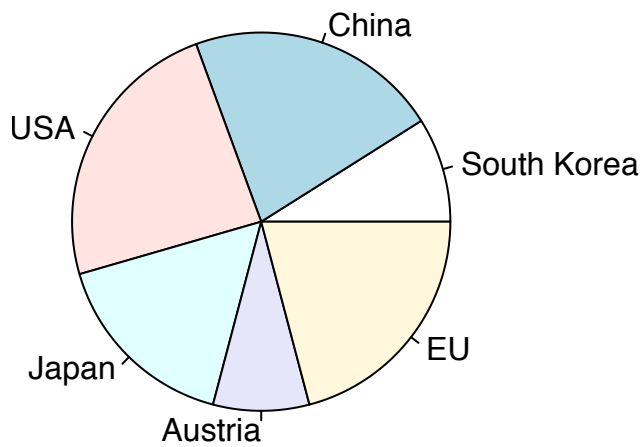


```
barplot(VADeaths, col = gray.colors(4), log="xy")
```

2.2.6 pie chart

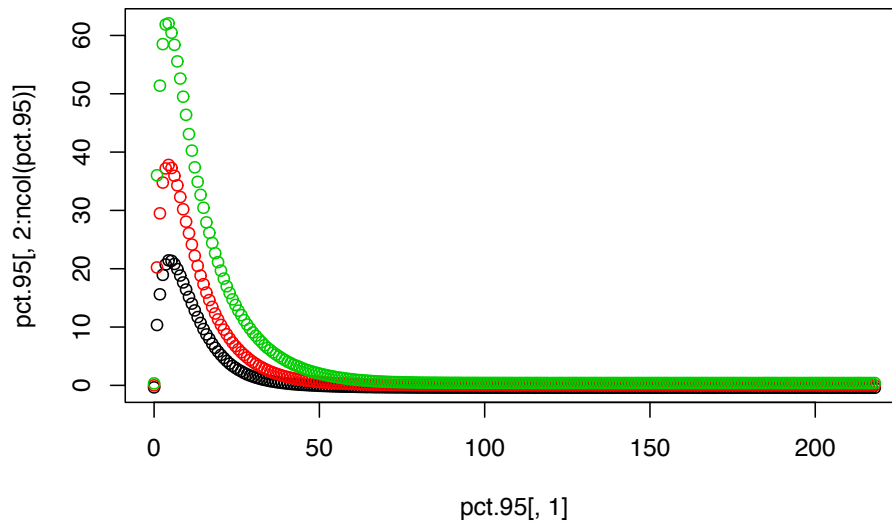
```
drug.X.market <- c(0.12, 0.29, 0.32, 0.22, 0.11, 0.28)
names(drug.X.market) <- c("South Korea", "China", "USA", "Japan", "Austria", "EU")
pie(drug.X.market)
```



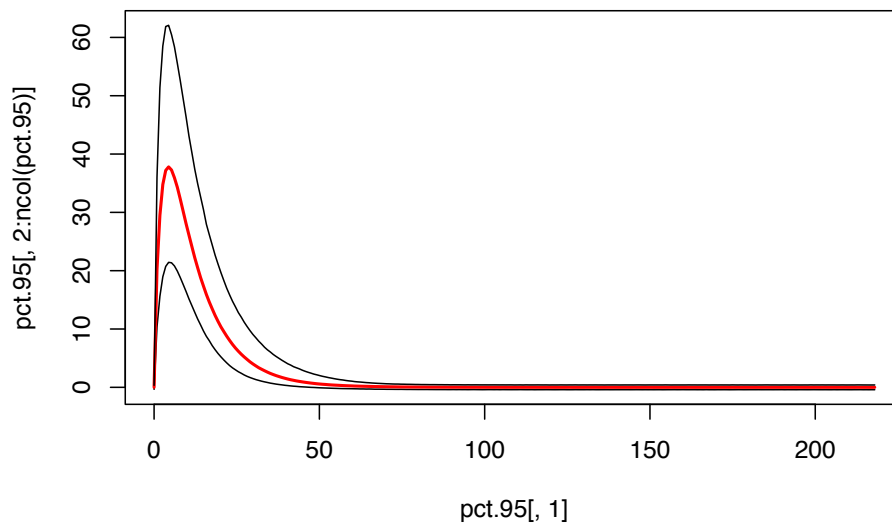
2.2.7 matplot 함수

2.2.7.1 matrix와 column 사이의 그림

```
pct.95 <- read.csv("pct95.csv")
matplot(pct.95[,1], pct.95[,2:ncol(pct.95)], pch=1)
```

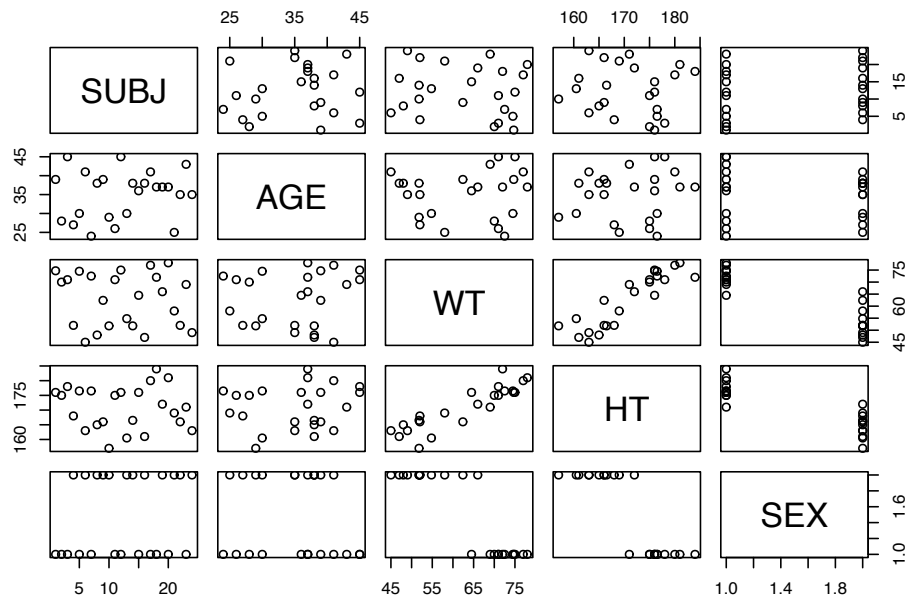


```
matplot(pct.95[,1], pct.95[,2:ncol(pct.95)], pch=1, col=c(1,2,1), type="l", lty=1, lwd=c(1,2,1))
```



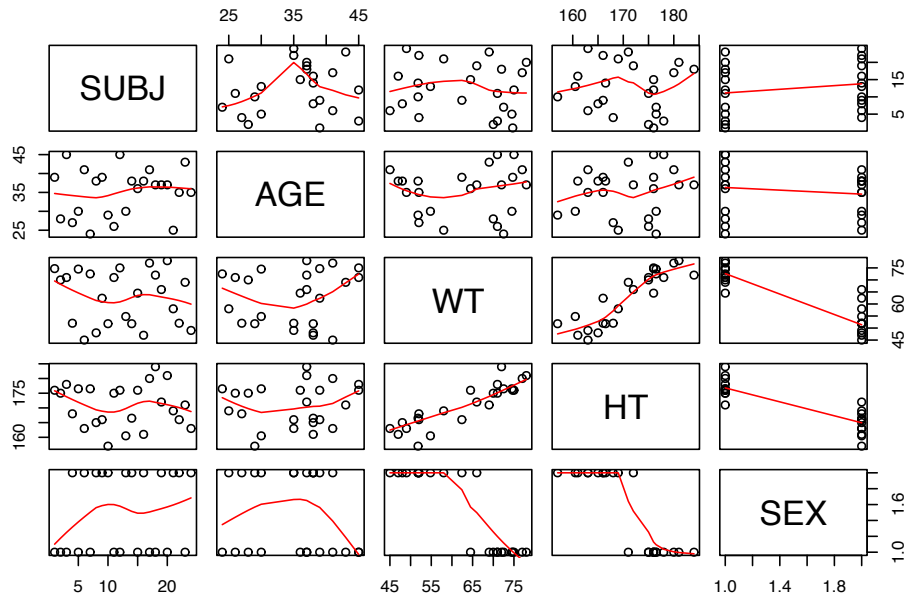
2.2.8 Scatter plot matrices (pairs plots)

```
pairs(d.demog)
```



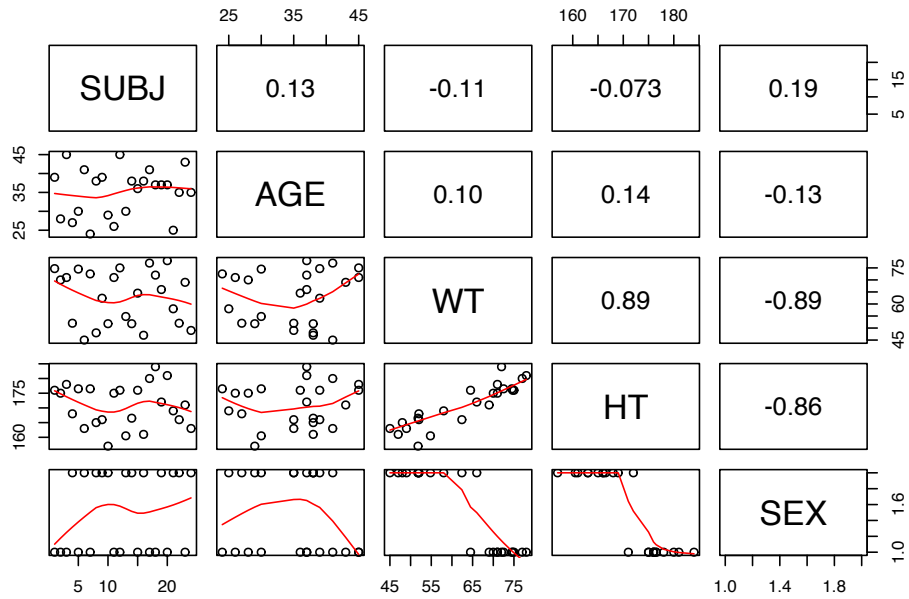
2.2.8.1 add a loess smoother, type

```
pairs(d.demog, panel = panel.smooth)
```



```
panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r = (cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex <- 1.5
  text(0.5, 0.5, txt, cex = 1.5)
}

pairs(d.demog, lower.panel=panel.smooth, upper.panel=panel.cor)
```



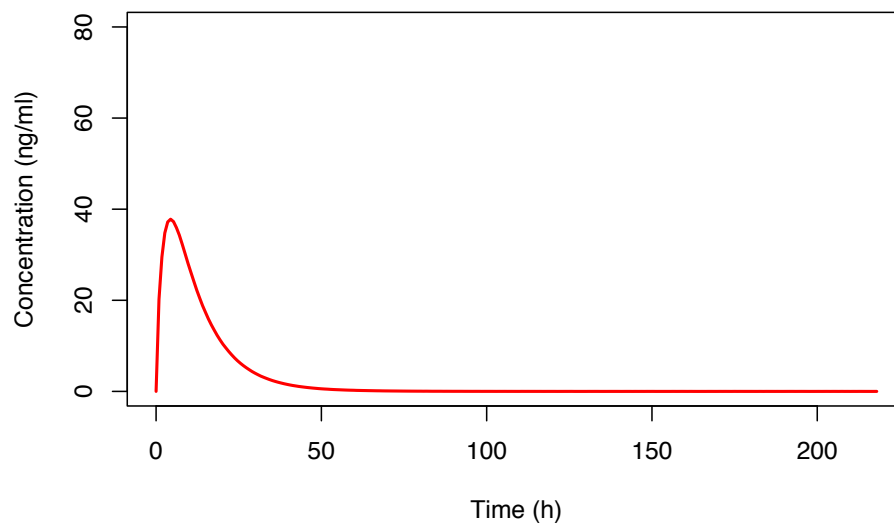
2.3 하위수준 그림 함수

- points : 점 추가
- lines : 선 추가
- abline : 기준선 추가
- mtext : 텍스트 추가
- legend : 설명 (legend) 추가
- polygon : polygon 추가

2.3.1 점, 선, 설명 추가 하기 {add}

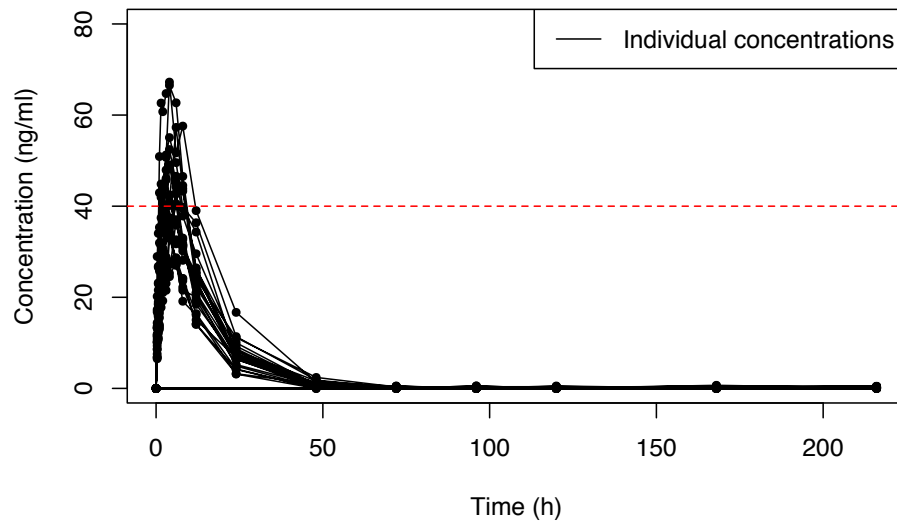
```
plot(pct.95$TIME, pct.95$PCT50, main="PK of Drug X"
     , type="l", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80), lty=1, col="red", lwd=2)
```

PK of Drug X



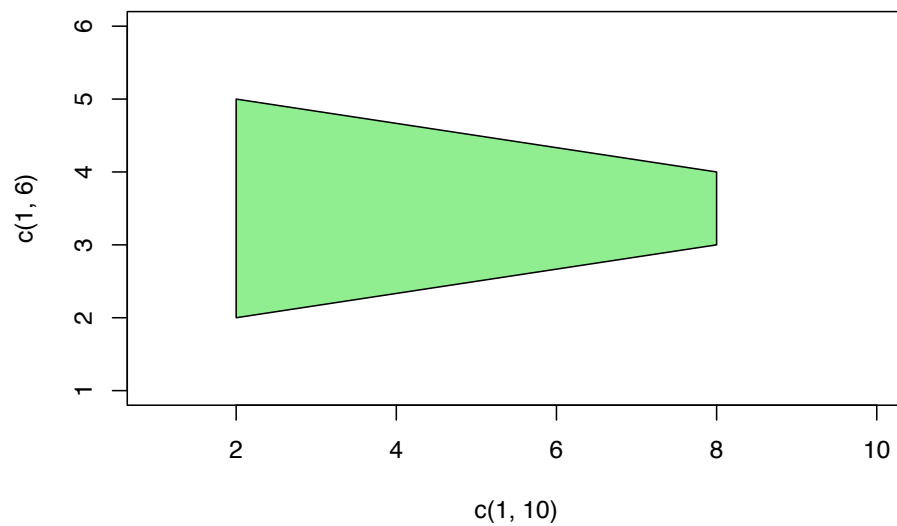
```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) # abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
      , lty=1, col="black")
```

PK of Drug X

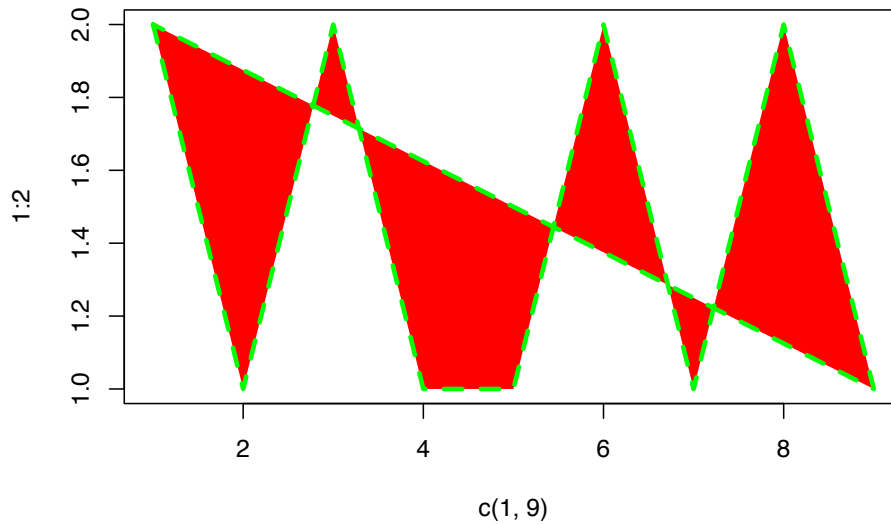


2.3.2 polygon 함수

```
plot(c(1, 10), c(1, 6), type = "n")
polygon(c(2,8,8,2), c(5,4,3,2), col="lightgreen")
```



```
plot(c(1, 9), 1:2, type = "n")
polygon(1:9, c(2,1,2,1,1,2,1,2,1),
       col = c("red", "blue"),
       border = c("green", "yellow"),
       lwd = 3, lty = c("dashed", "solid"))
```



2.4 그림 출력하기

2.4.1 pdf graphics devices

```
pdf("PK_of_Drug_X.pdf")

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
      , lty=1, col="black")
```



```
dev.off()
```

```
## cairo_pdf  
##      2
```

2.4.2 PNG graphics devices

```
png("PK_of_Drug_X.png")  
  
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"  
      , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"  
      , ylim=range(0,80))  
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)  
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)  
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x  
legend("topright", legend=c("Individual concentrations")  
      , lty=1, col="black")  
  
dev.off()  
  
## cairo_pdf  
##      2
```



3

Data Import / Export

2017-03-29 배균섭 교수님 강의

이번 시간에는 자료를 불러오고 조작을 가한 뒤 저장하는 방법에 대해 알아보겠습니다.

3.1 Read.csv

setwd 명령어를 통해서 자료가 있는 작업 공간을 설정할 수 있습니다. 설정 후에는 dir()을 통해 파일의 이름을 확인 할 수 있습니다. read.csv를 통해서 자료를 R에서 사용할 수 있게 됩니다.

```
setwd("D:/Rt")
dir()
mydata <- read.csv("MyData2017.csv", as.is=TRUE)
```

3.2 Theoph 데이터

R에 기본적으로 들어있는 Theoph 약동학 자료에 대해 살펴보겠습니다.

```
head(Theoph, n = 11)
```

```
##      Subject    Wt Dose  Time  conc
## 1           1  79.6  4.02   0.00  0.74
```

```
## 2      1 79.6 4.02 0.25 2.84
## 3      1 79.6 4.02 0.57 6.57
## 4      1 79.6 4.02 1.12 10.50
## 5      1 79.6 4.02 2.02 9.66
## 6      1 79.6 4.02 3.82 8.58
## 7      1 79.6 4.02 5.10 8.36
## 8      1 79.6 4.02 7.03 7.47
## 9      1 79.6 4.02 9.05 6.89
## 10     1 79.6 4.02 12.12 5.94
## 11     1 79.6 4.02 24.37 3.28
```

```
tail(Theoph, n = 11)
```

```
##      Subject  Wt Dose  Time conc
## 122      12 60.5  5.3  0.00 0.00
## 123      12 60.5  5.3  0.25 1.25
## 124      12 60.5  5.3  0.50 3.96
## 125      12 60.5  5.3  1.00 7.82
## 126      12 60.5  5.3  2.00 9.72
## 127      12 60.5  5.3  3.52 9.75
## 128      12 60.5  5.3  5.07 8.57
## 129      12 60.5  5.3  7.07 6.59
## 130      12 60.5  5.3  9.03 6.11
## 131      12 60.5  5.3 12.05 4.57
## 132      12 60.5  5.3 24.15 1.17
```

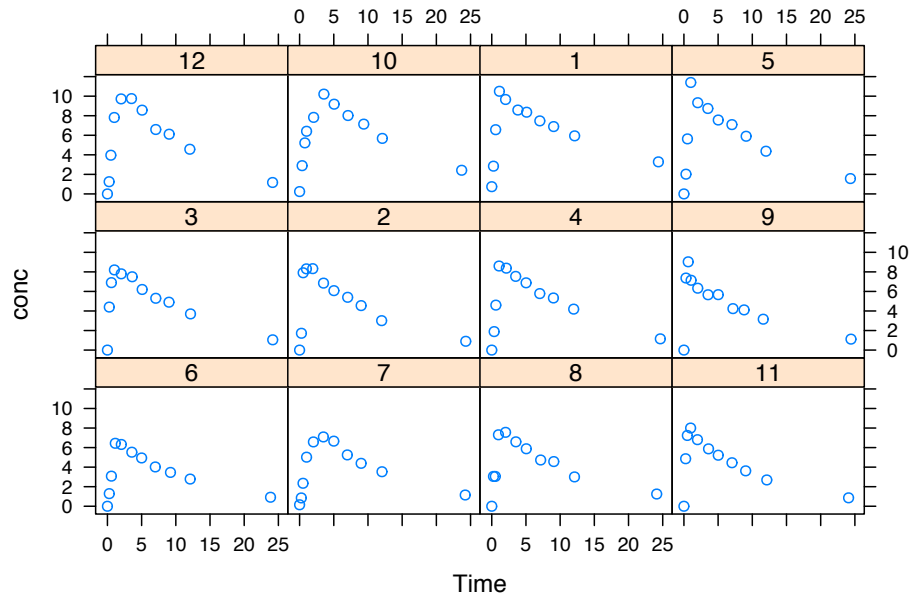
R console에서 ?Theoph를 타이핑 치면 좀 더 자세한 정보를 얻을 수 있습니다.

3.3 lattice

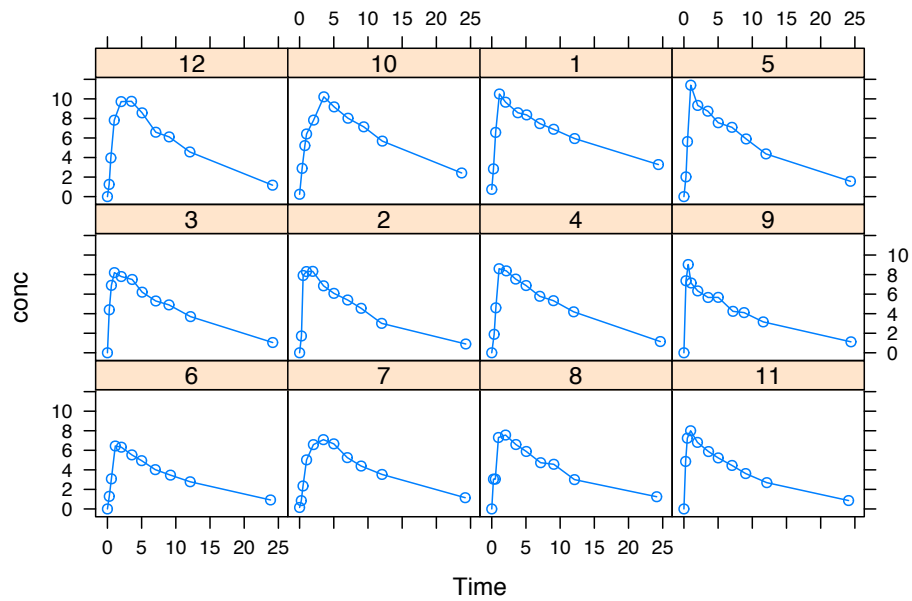
lattice 패키지를 불러온 뒤 그림을 그려보겠습니다. (Sarkar, 2017)

```
library(lattice) # trellis

xyplot(conc ~ Time | Subject, data=Theoph)
```

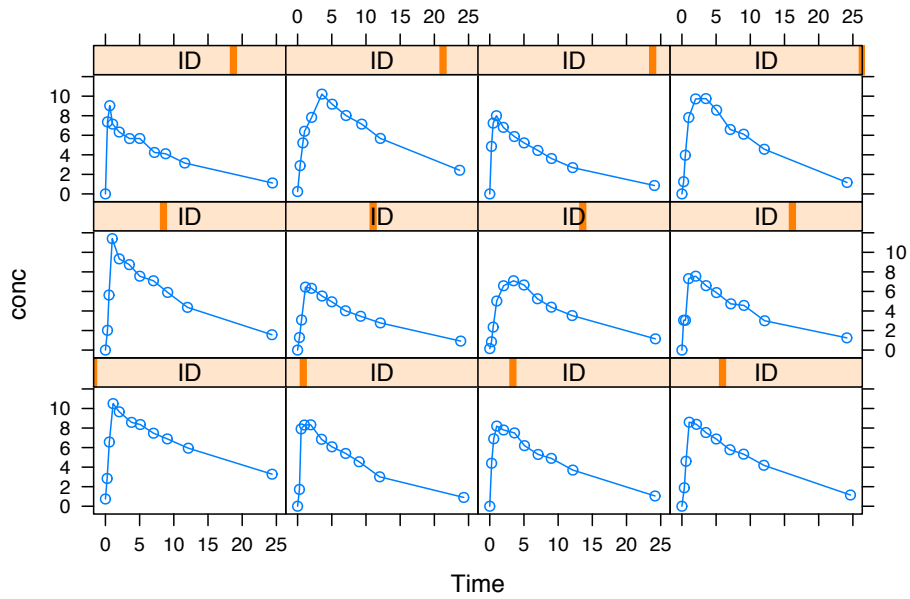


```
xyplot(conc ~ Time | Subject, data=Theoph, type="b")
```

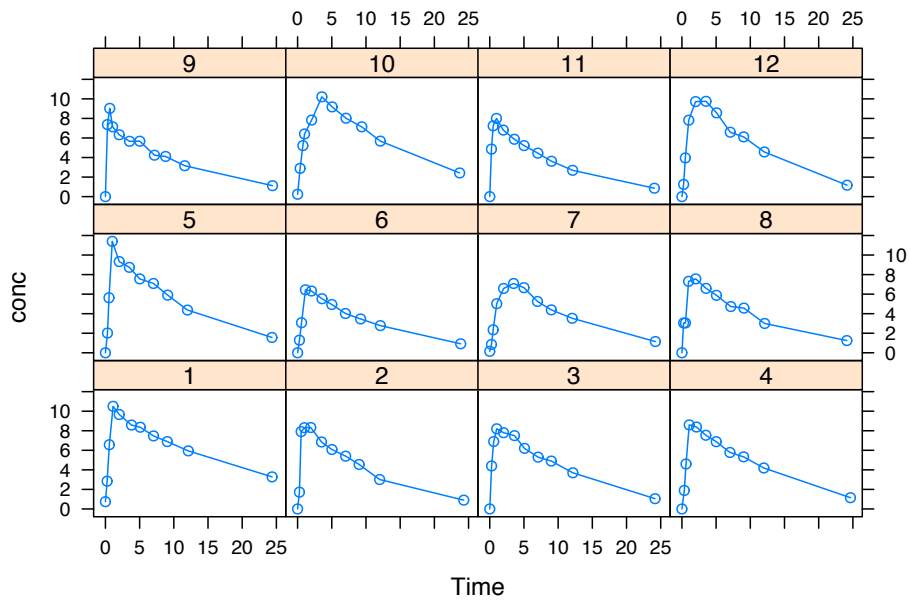


```
Theoph[, "ID"] = as.numeric(as.character(Theoph[, "Subject"]))
```

```
xyplot(conc ~ Time | ID, data=Theoph, type="b")
```



```
xyplot(conc ~ Time | as.factor(ID), data=Theoph, type="b")
```



```
write.csv(Theoph, "Theoph.csv", row.names=FALSE, quote=FALSE, na="")
```

3.4 Subsetting and write.csv

자료를 편집하고, subset을 만들고 각각을 파일로 저장하는 방법에 대해 알아보겠습니다.

```
IDs = sort(unique(Theoph[, "ID"])) ; IDs
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12
```

```
nID = length(IDs) ; nID
```

```
## [1] 12
```

```
demog = unique(Theoph[, c("ID", "Wt")])
colnames(demog) = c("ID", "BWT")
write.csv(demog, "1-demog.csv", row.names=FALSE, quote=FALSE, na="")
```

```
DV = Theoph[, c("ID", "Time", "conc")]
colnames(DV) = c("ID", "TIME", "DV")
write.csv(DV, "3-DV.csv", row.names=FALSE, quote=FALSE, na="")
```

```
adm = cbind(IDs, rep(0, nID), rep(320, nID))
colnames(adm) = c("ID", "TIME", "AMT")
write.csv(adm, "2-adm.csv", row.names=FALSE, quote=FALSE, na="")
```

```
demog = read.csv("1-demog.csv", as.is=TRUE)
adm = read.csv("2-adm.csv", as.is=TRUE)
dv = read.csv("3-dv.csv", as.is=TRUE)
```

```
AdmDv = merge(adm, dv, by=intersect(colnames(adm), colnames(dv)), all=TRUE)
AdmDv
```

```
##      ID  TIME AMT    DV
## 1    1  0.00 320  0.74
## 2    1  0.25  NA  2.84
## 3    1  0.57  NA  6.57
## 4    1  1.12  NA 10.50
## 5    1  2.02  NA  9.66
```

## 6	1	3.82	NA	8.58
## 7	1	5.10	NA	8.36
## 8	1	7.03	NA	7.47
## 9	1	9.05	NA	6.89
## 10	1	12.12	NA	5.94
## 11	1	24.37	NA	3.28
## 12	2	0.00	320	0.00
## 13	2	0.27	NA	1.72
## 14	2	0.52	NA	7.91
## 15	2	1.00	NA	8.31
## 16	2	1.92	NA	8.33
## 17	2	3.50	NA	6.85
## 18	2	5.02	NA	6.08
## 19	2	7.03	NA	5.40
## 20	2	9.00	NA	4.55
## 21	2	12.00	NA	3.01
## 22	2	24.30	NA	0.90
## 23	3	0.00	320	0.00
## 24	3	0.27	NA	4.40
## 25	3	0.58	NA	6.90
## 26	3	1.02	NA	8.20
## 27	3	2.02	NA	7.80
## 28	3	3.62	NA	7.50
## 29	3	5.08	NA	6.20
## 30	3	7.07	NA	5.30
## 31	3	9.00	NA	4.90
## 32	3	12.15	NA	3.70
## 33	3	24.17	NA	1.05
## 34	4	0.00	320	0.00
## 35	4	0.35	NA	1.89
## 36	4	0.60	NA	4.60
## 37	4	1.07	NA	8.60
## 38	4	2.13	NA	8.38
## 39	4	3.50	NA	7.54
## 40	4	5.02	NA	6.88
## 41	4	7.02	NA	5.78
## 42	4	9.02	NA	5.33
## 43	4	11.98	NA	4.19
## 44	4	24.65	NA	1.15
## 45	5	0.00	320	0.00
## 46	5	0.30	NA	2.02
## 47	5	0.52	NA	5.63
## 48	5	1.00	NA	11.40
## 49	5	2.02	NA	9.33
## 50	5	3.50	NA	8.74


```
## 51  5  5.02 NA  7.56
## 52  5  7.02 NA  7.09
## 53  5  9.10 NA  5.90
## 54  5 12.00 NA  4.37
## 55  5 24.35 NA  1.57
## 56  6  0.00 320  0.00
## 57  6  0.27 NA  1.29
## 58  6  0.58 NA  3.08
## 59  6  1.15 NA  6.44
## 60  6  2.03 NA  6.32
## 61  6  3.57 NA  5.53
## 62  6  5.00 NA  4.94
## 63  6  7.00 NA  4.02
## 64  6  9.22 NA  3.46
## 65  6 12.10 NA  2.78
## 66  6 23.85 NA  0.92
## 67  7  0.00 320  0.15
## 68  7  0.25 NA  0.85
## 69  7  0.50 NA  2.35
## 70  7  1.02 NA  5.02
## 71  7  2.02 NA  6.58
## 72  7  3.48 NA  7.09
## 73  7  5.00 NA  6.66
## 74  7  6.98 NA  5.25
## 75  7  9.00 NA  4.39
## 76  7 12.05 NA  3.53
## 77  7 24.22 NA  1.15
## 78  8  0.00 320  0.00
## 79  8  0.25 NA  3.05
## 80  8  0.52 NA  3.05
## 81  8  0.98 NA  7.31
## 82  8  2.02 NA  7.56
## 83  8  3.53 NA  6.59
## 84  8  5.05 NA  5.88
## 85  8  7.15 NA  4.73
## 86  8  9.07 NA  4.57
## 87  8 12.10 NA  3.00
## 88  8 24.12 NA  1.25
## 89  9  0.00 320  0.00
## 90  9  0.30 NA  7.37
## 91  9  0.63 NA  9.03
## 92  9  1.05 NA  7.14
## 93  9  2.02 NA  6.33
## 94  9  3.53 NA  5.66
## 95  9  5.02 NA  5.67
```

```
## 96  9  7.17 NA  4.24
## 97  9  8.80 NA  4.11
## 98  9 11.60 NA  3.16
## 99  9 24.43 NA  1.12
## 100 10 0.00 320 0.24
## 101 10 0.37 NA  2.89
## 102 10 0.77 NA  5.22
## 103 10 1.02 NA  6.41
## 104 10 2.05 NA  7.83
## 105 10 3.55 NA 10.21
## 106 10 5.05 NA  9.18
## 107 10 7.08 NA  8.02
## 108 10 9.38 NA  7.14
## 109 10 12.10 NA 5.68
## 110 10 23.70 NA  2.42
## 111 11 0.00 320 0.00
## 112 11 0.25 NA  4.86
## 113 11 0.50 NA  7.24
## 114 11 0.98 NA  8.00
## 115 11 1.98 NA  6.81
## 116 11 3.60 NA  5.87
## 117 11 5.02 NA  5.22
## 118 11 7.03 NA  4.45
## 119 11 9.03 NA  3.62
## 120 11 12.12 NA  2.69
## 121 11 24.08 NA  0.86
## 122 12 0.00 320 0.00
## 123 12 0.25 NA  1.25
## 124 12 0.50 NA  3.96
## 125 12 1.00 NA  7.82
## 126 12 2.00 NA  9.72
## 127 12 3.52 NA  9.75
## 128 12 5.07 NA  8.57
## 129 12 7.07 NA  6.59
## 130 12 9.03 NA  6.11
## 131 12 12.05 NA  4.57
## 132 12 24.15 NA  1.17
```

자료를 병합(merge)해 보겠습니다.

```
DataAll = merge(demog, AdmDv, by=c("ID"), all=TRUE)
DataAll
```

```
##      ID  BWT  TIME AMT    DV
## 1     1  79.6   0.00 320  0.74
```

```
## 2    1 79.6 0.25 NA 2.84
## 3    1 79.6 0.57 NA 6.57
## 4    1 79.6 1.12 NA 10.50
## 5    1 79.6 2.02 NA 9.66
## 6    1 79.6 3.82 NA 8.58
## 7    1 79.6 5.10 NA 8.36
## 8    1 79.6 7.03 NA 7.47
## 9    1 79.6 9.05 NA 6.89
## 10   1 79.6 12.12 NA 5.94
## 11   1 79.6 24.37 NA 3.28
## 12   2 72.4 0.00 320 0.00
## 13   2 72.4 0.27 NA 1.72
## 14   2 72.4 0.52 NA 7.91
## 15   2 72.4 1.00 NA 8.31
## 16   2 72.4 1.92 NA 8.33
## 17   2 72.4 3.50 NA 6.85
## 18   2 72.4 5.02 NA 6.08
## 19   2 72.4 7.03 NA 5.40
## 20   2 72.4 9.00 NA 4.55
## 21   2 72.4 12.00 NA 3.01
## 22   2 72.4 24.30 NA 0.90
## 23   3 70.5 0.00 320 0.00
## 24   3 70.5 0.27 NA 4.40
## 25   3 70.5 0.58 NA 6.90
## 26   3 70.5 1.02 NA 8.20
## 27   3 70.5 2.02 NA 7.80
## 28   3 70.5 3.62 NA 7.50
## 29   3 70.5 5.08 NA 6.20
## 30   3 70.5 7.07 NA 5.30
## 31   3 70.5 9.00 NA 4.90
## 32   3 70.5 12.15 NA 3.70
## 33   3 70.5 24.17 NA 1.05
## 34   4 72.7 0.00 320 0.00
## 35   4 72.7 0.35 NA 1.89
## 36   4 72.7 0.60 NA 4.60
## 37   4 72.7 1.07 NA 8.60
## 38   4 72.7 2.13 NA 8.38
## 39   4 72.7 3.50 NA 7.54
## 40   4 72.7 5.02 NA 6.88
## 41   4 72.7 7.02 NA 5.78
## 42   4 72.7 9.02 NA 5.33
## 43   4 72.7 11.98 NA 4.19
## 44   4 72.7 24.65 NA 1.15
## 45   5 54.6 0.00 320 0.00
## 46   5 54.6 0.30 NA 2.02
```

```
## 47 5 54.6 0.52 NA 5.63
## 48 5 54.6 1.00 NA 11.40
## 49 5 54.6 2.02 NA 9.33
## 50 5 54.6 3.50 NA 8.74
## 51 5 54.6 5.02 NA 7.56
## 52 5 54.6 7.02 NA 7.09
## 53 5 54.6 9.10 NA 5.90
## 54 5 54.6 12.00 NA 4.37
## 55 5 54.6 24.35 NA 1.57
## 56 6 80.0 0.00 320 0.00
## 57 6 80.0 0.27 NA 1.29
## 58 6 80.0 0.58 NA 3.08
## 59 6 80.0 1.15 NA 6.44
## 60 6 80.0 2.03 NA 6.32
## 61 6 80.0 3.57 NA 5.53
## 62 6 80.0 5.00 NA 4.94
## 63 6 80.0 7.00 NA 4.02
## 64 6 80.0 9.22 NA 3.46
## 65 6 80.0 12.10 NA 2.78
## 66 6 80.0 23.85 NA 0.92
## 67 7 64.6 0.00 320 0.15
## 68 7 64.6 0.25 NA 0.85
## 69 7 64.6 0.50 NA 2.35
## 70 7 64.6 1.02 NA 5.02
## 71 7 64.6 2.02 NA 6.58
## 72 7 64.6 3.48 NA 7.09
## 73 7 64.6 5.00 NA 6.66
## 74 7 64.6 6.98 NA 5.25
## 75 7 64.6 9.00 NA 4.39
## 76 7 64.6 12.05 NA 3.53
## 77 7 64.6 24.22 NA 1.15
## 78 8 70.5 0.00 320 0.00
## 79 8 70.5 0.25 NA 3.05
## 80 8 70.5 0.52 NA 3.05
## 81 8 70.5 0.98 NA 7.31
## 82 8 70.5 2.02 NA 7.56
## 83 8 70.5 3.53 NA 6.59
## 84 8 70.5 5.05 NA 5.88
## 85 8 70.5 7.15 NA 4.73
## 86 8 70.5 9.07 NA 4.57
## 87 8 70.5 12.10 NA 3.00
## 88 8 70.5 24.12 NA 1.25
## 89 9 86.4 0.00 320 0.00
## 90 9 86.4 0.30 NA 7.37
## 91 9 86.4 0.63 NA 9.03
```

```
## 92  9 86.4  1.05 NA  7.14
## 93  9 86.4  2.02 NA  6.33
## 94  9 86.4  3.53 NA  5.66
## 95  9 86.4  5.02 NA  5.67
## 96  9 86.4  7.17 NA  4.24
## 97  9 86.4  8.80 NA  4.11
## 98  9 86.4 11.60 NA  3.16
## 99  9 86.4 24.43 NA  1.12
## 100 10 58.2  0.00 320  0.24
## 101 10 58.2  0.37 NA  2.89
## 102 10 58.2  0.77 NA  5.22
## 103 10 58.2  1.02 NA  6.41
## 104 10 58.2  2.05 NA  7.83
## 105 10 58.2  3.55 NA 10.21
## 106 10 58.2  5.05 NA  9.18
## 107 10 58.2  7.08 NA  8.02
## 108 10 58.2  9.38 NA  7.14
## 109 10 58.2 12.10 NA  5.68
## 110 10 58.2 23.70 NA  2.42
## 111 11 65.0  0.00 320  0.00
## 112 11 65.0  0.25 NA  4.86
## 113 11 65.0  0.50 NA  7.24
## 114 11 65.0  0.98 NA  8.00
## 115 11 65.0  1.98 NA  6.81
## 116 11 65.0  3.60 NA  5.87
## 117 11 65.0  5.02 NA  5.22
## 118 11 65.0  7.03 NA  4.45
## 119 11 65.0  9.03 NA  3.62
## 120 11 65.0 12.12 NA  2.69
## 121 11 65.0 24.08 NA  0.86
## 122 12 60.5  0.00 320  0.00
## 123 12 60.5  0.25 NA  1.25
## 124 12 60.5  0.50 NA  3.96
## 125 12 60.5  1.00 NA  7.82
## 126 12 60.5  2.00 NA  9.72
## 127 12 60.5  3.52 NA  9.75
## 128 12 60.5  5.07 NA  8.57
## 129 12 60.5  7.07 NA  6.59
## 130 12 60.5  9.03 NA  6.11
## 131 12 60.5 12.05 NA  4.57
## 132 12 60.5 24.15 NA  1.17
```



4

Frequently Used Functions

2017-04-05 배균섭 교수님 강의

자주 쓰는 함수 및 명령어에 대해 알아보겠습니다.

4.1 Command

```
# 2017-04-05 R-intro.pdf Chapter 08
```

```
pois
```

```
## Error in eval(expr, envir, enclos): 객체 'pois'를 찾을 수 없습니다
```

```
?dbeta
```

```
dnorm(0)
```

```
## [1] 0.3989
```

```
pnorm(0)
```

```
## [1] 0.5
```

```
1 - pnorm(1.96)
```

```
## [1] 0.025
```

```
?pnorm  
pnorm(1.96, lower.tail=FALSE)
```

```
## [1] 0.025
```

```
qnorm(0.5)
```

```
## [1] 0
```

```
qnorm(0.975)
```

```
## [1] 1.96
```

```
format(qnorm(0.975), digits=22)
```

```
## [1] "1.959963984540053605343"
```

```
rnorm(5)
```

```
## [1] 1.2477 -0.9699 0.4265 -0.7116 0.4842
```

```
rnorm(5, 10, 1)
```

```
## [1] 10.498 10.926 9.292 10.089 10.487
```

```
x = rnorm(100, 10, 1)  
mean(x)
```

```
## [1] 10.02
```

```
sd(x)
```

```
## [1] 0.8721
```

```
2*pt(-2.43, df = 13)
```

```
## [1] 0.03033
```

```
2*pt(-2.43, df = 1000)
```

```
## [1] 0.01527
```



```
qnorm(0.995)
```

```
## [1] 2.576
```

```
qf(0.01, 2, 7, lower.tail = FALSE)
```

```
## [1] 9.547
```

```
?fivenum
```

```
faithful
```

```
##      eruptions waiting
## 1      3.600      79
## 2      1.800      54
## 3      3.333      74
## 4      2.283      62
## 5      4.533      85
## 6      2.883      55
## 7      4.700      88
## 8      3.600      85
## 9      1.950      51
## 10     4.350      85
## 11     1.833      54
## 12     3.917      84
## 13     4.200      78
## 14     1.750      47
## 15     4.700      83
## 16     2.167      52
## 17     1.750      62
## 18     4.800      84
## 19     1.600      52
## 20     4.250      79
## 21     1.800      51
## 22     1.750      47
## 23     3.450      78
## 24     3.067      69
## 25     4.533      74
## 26     3.600      83
## 27     1.967      55
## 28     4.083      76
## 29     3.850      78
## 30     4.433      79
## 31     4.300      73
## 32     4.467      77
```

## 33	3.367	66
## 34	4.033	80
## 35	3.833	74
## 36	2.017	52
## 37	1.867	48
## 38	4.833	80
## 39	1.833	59
## 40	4.783	90
## 41	4.350	80
## 42	1.883	58
## 43	4.567	84
## 44	1.750	58
## 45	4.533	73
## 46	3.317	83
## 47	3.833	64
## 48	2.100	53
## 49	4.633	82
## 50	2.000	59
## 51	4.800	75
## 52	4.716	90
## 53	1.833	54
## 54	4.833	80
## 55	1.733	54
## 56	4.883	83
## 57	3.717	71
## 58	1.667	64
## 59	4.567	77
## 60	4.317	81
## 61	2.233	59
## 62	4.500	84
## 63	1.750	48
## 64	4.800	82
## 65	1.817	60
## 66	4.400	92
## 67	4.167	78
## 68	4.700	78
## 69	2.067	65
## 70	4.700	73
## 71	4.033	82
## 72	1.967	56
## 73	4.500	79
## 74	4.000	71
## 75	1.983	62
## 76	5.067	76
## 77	2.017	60

## 78	4.567	78
## 79	3.883	76
## 80	3.600	83
## 81	4.133	75
## 82	4.333	82
## 83	4.100	70
## 84	2.633	65
## 85	4.067	73
## 86	4.933	88
## 87	3.950	76
## 88	4.517	80
## 89	2.167	48
## 90	4.000	86
## 91	2.200	60
## 92	4.333	90
## 93	1.867	50
## 94	4.817	78
## 95	1.833	63
## 96	4.300	72
## 97	4.667	84
## 98	3.750	75
## 99	1.867	51
## 100	4.900	82
## 101	2.483	62
## 102	4.367	88
## 103	2.100	49
## 104	4.500	83
## 105	4.050	81
## 106	1.867	47
## 107	4.700	84
## 108	1.783	52
## 109	4.850	86
## 110	3.683	81
## 111	4.733	75
## 112	2.300	59
## 113	4.900	89
## 114	4.417	79
## 115	1.700	59
## 116	4.633	81
## 117	2.317	50
## 118	4.600	85
## 119	1.817	59
## 120	4.417	87
## 121	2.617	53
## 122	4.067	69

## 123	4.250	77
## 124	1.967	56
## 125	4.600	88
## 126	3.767	81
## 127	1.917	45
## 128	4.500	82
## 129	2.267	55
## 130	4.650	90
## 131	1.867	45
## 132	4.167	83
## 133	2.800	56
## 134	4.333	89
## 135	1.833	46
## 136	4.383	82
## 137	1.883	51
## 138	4.933	86
## 139	2.033	53
## 140	3.733	79
## 141	4.233	81
## 142	2.233	60
## 143	4.533	82
## 144	4.817	77
## 145	4.333	76
## 146	1.983	59
## 147	4.633	80
## 148	2.017	49
## 149	5.100	96
## 150	1.800	53
## 151	5.033	77
## 152	4.000	77
## 153	2.400	65
## 154	4.600	81
## 155	3.567	71
## 156	4.000	70
## 157	4.500	81
## 158	4.083	93
## 159	1.800	53
## 160	3.967	89
## 161	2.200	45
## 162	4.150	86
## 163	2.000	58
## 164	3.833	78
## 165	3.500	66
## 166	4.583	76
## 167	2.367	63

## 168	5.000	88
## 169	1.933	52
## 170	4.617	93
## 171	1.917	49
## 172	2.083	57
## 173	4.583	77
## 174	3.333	68
## 175	4.167	81
## 176	4.333	81
## 177	4.500	73
## 178	2.417	50
## 179	4.000	85
## 180	4.167	74
## 181	1.883	55
## 182	4.583	77
## 183	4.250	83
## 184	3.767	83
## 185	2.033	51
## 186	4.433	78
## 187	4.083	84
## 188	1.833	46
## 189	4.417	83
## 190	2.183	55
## 191	4.800	81
## 192	1.833	57
## 193	4.800	76
## 194	4.100	84
## 195	3.966	77
## 196	4.233	81
## 197	3.500	87
## 198	4.366	77
## 199	2.250	51
## 200	4.667	78
## 201	2.100	60
## 202	4.350	82
## 203	4.133	91
## 204	1.867	53
## 205	4.600	78
## 206	1.783	46
## 207	4.367	77
## 208	3.850	84
## 209	1.933	49
## 210	4.500	83
## 211	2.383	71
## 212	4.700	80

## 213	1.867	49
## 214	3.833	75
## 215	3.417	64
## 216	4.233	76
## 217	2.400	53
## 218	4.800	94
## 219	2.000	55
## 220	4.150	76
## 221	1.867	50
## 222	4.267	82
## 223	1.750	54
## 224	4.483	75
## 225	4.000	78
## 226	4.117	79
## 227	4.083	78
## 228	4.267	78
## 229	3.917	70
## 230	4.550	79
## 231	4.083	70
## 232	2.417	54
## 233	4.183	86
## 234	2.217	50
## 235	4.450	90
## 236	1.883	54
## 237	1.850	54
## 238	4.283	77
## 239	3.950	79
## 240	2.333	64
## 241	4.150	75
## 242	2.350	47
## 243	4.933	86
## 244	2.900	63
## 245	4.583	85
## 246	3.833	82
## 247	2.083	57
## 248	4.367	82
## 249	2.133	67
## 250	4.350	74
## 251	2.200	54
## 252	4.450	83
## 253	3.567	73
## 254	4.500	73
## 255	4.150	88
## 256	3.817	80
## 257	3.917	71

```
## 258    4.450    83
## 259    2.000    56
## 260    4.283    79
## 261    4.767    78
## 262    4.533    84
## 263    1.850    58
## 264    4.250    83
## 265    1.983    43
## 266    2.250    60
## 267    4.750    75
## 268    4.117    81
## 269    2.150    46
## 270    4.417    90
## 271    1.817    46
## 272    4.467    74
```

```
str(faithful)
```

```
## 'data.frame':    272 obs. of  2 variables:
## $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
## $ waiting : num  79 54 74 62 85 55 88 85 51 85 ...
```

```
eruptions
```

```
## Error in eval(expr, envir, enclos): 객체 'eruptions'를 찾을 수 없습니다
```

```
attach(faithful)
```

```
eruptions
```

```
## [1] 3.600 1.800 3.333 2.283 4.533 2.883 4.700 3.600
## [9] 1.950 4.350 1.833 3.917 4.200 1.750 4.700 2.167
## [17] 1.750 4.800 1.600 4.250 1.800 1.750 3.450 3.067
## [25] 4.533 3.600 1.967 4.083 3.850 4.433 4.300 4.467
## [33] 3.367 4.033 3.833 2.017 1.867 4.833 1.833 4.783
## [41] 4.350 1.883 4.567 1.750 4.533 3.317 3.833 2.100
## [49] 4.633 2.000 4.800 4.716 1.833 4.833 1.733 4.883
## [57] 3.717 1.667 4.567 4.317 2.233 4.500 1.750 4.800
## [65] 1.817 4.400 4.167 4.700 2.067 4.700 4.033 1.967
## [73] 4.500 4.000 1.983 5.067 2.017 4.567 3.883 3.600
## [81] 4.133 4.333 4.100 2.633 4.067 4.933 3.950 4.517
## [89] 2.167 4.000 2.200 4.333 1.867 4.817 1.833 4.300
## [97] 4.667 3.750 1.867 4.900 2.483 4.367 2.100 4.500
## [105] 4.050 1.867 4.700 1.783 4.850 3.683 4.733 2.300
## [113] 4.900 4.417 1.700 4.633 2.317 4.600 1.817 4.417
```

```
## [121] 2.617 4.067 4.250 1.967 4.600 3.767 1.917 4.500
## [129] 2.267 4.650 1.867 4.167 2.800 4.333 1.833 4.383
## [137] 1.883 4.933 2.033 3.733 4.233 2.233 4.533 4.817
## [145] 4.333 1.983 4.633 2.017 5.100 1.800 5.033 4.000
## [153] 2.400 4.600 3.567 4.000 4.500 4.083 1.800 3.967
## [161] 2.200 4.150 2.000 3.833 3.500 4.583 2.367 5.000
## [169] 1.933 4.617 1.917 2.083 4.583 3.333 4.167 4.333
## [177] 4.500 2.417 4.000 4.167 1.883 4.583 4.250 3.767
## [185] 2.033 4.433 4.083 1.833 4.417 2.183 4.800 1.833
## [193] 4.800 4.100 3.966 4.233 3.500 4.366 2.250 4.667
## [201] 2.100 4.350 4.133 1.867 4.600 1.783 4.367 3.850
## [209] 1.933 4.500 2.383 4.700 1.867 3.833 3.417 4.233
## [217] 2.400 4.800 2.000 4.150 1.867 4.267 1.750 4.483
## [225] 4.000 4.117 4.083 4.267 3.917 4.550 4.083 2.417
## [233] 4.183 2.217 4.450 1.883 1.850 4.283 3.950 2.333
## [241] 4.150 2.350 4.933 2.900 4.583 3.833 2.083 4.367
## [249] 2.133 4.350 2.200 4.450 3.567 4.500 4.150 3.817
## [257] 3.917 4.450 2.000 4.283 4.767 4.533 1.850 4.250
## [265] 1.983 2.250 4.750 4.117 2.150 4.417 1.817 4.467
```

```
waiting
```

```
## [1] 79 54 74 62 85 55 88 85 51 85 54 84 78 47 83 52
## [17] 62 84 52 79 51 47 78 69 74 83 55 76 78 79 73 77
## [33] 66 80 74 52 48 80 59 90 80 58 84 58 73 83 64 53
## [49] 82 59 75 90 54 80 54 83 71 64 77 81 59 84 48 82
## [65] 60 92 78 78 65 73 82 56 79 71 62 76 60 78 76 83
## [81] 75 82 70 65 73 88 76 80 48 86 60 90 50 78 63 72
## [97] 84 75 51 82 62 88 49 83 81 47 84 52 86 81 75 59
## [113] 89 79 59 81 50 85 59 87 53 69 77 56 88 81 45 82
## [129] 55 90 45 83 56 89 46 82 51 86 53 79 81 60 82 77
## [145] 76 59 80 49 96 53 77 77 65 81 71 70 81 93 53 89
## [161] 45 86 58 78 66 76 63 88 52 93 49 57 77 68 81 81
## [177] 73 50 85 74 55 77 83 83 51 78 84 46 83 55 81 57
## [193] 76 84 77 81 87 77 51 78 60 82 91 53 78 46 77 84
## [209] 49 83 71 80 49 75 64 76 53 94 55 76 50 82 54 75
## [225] 78 79 78 78 70 79 70 54 86 50 90 54 54 77 79 64
## [241] 75 47 86 63 85 82 57 82 67 74 54 83 73 73 88 80
## [257] 71 83 56 79 78 84 58 83 43 60 75 81 46 90 46 74
```

```
stem(waiting)
```

```
##
## The decimal point is 1 digit(s) to the right of the |
##
```



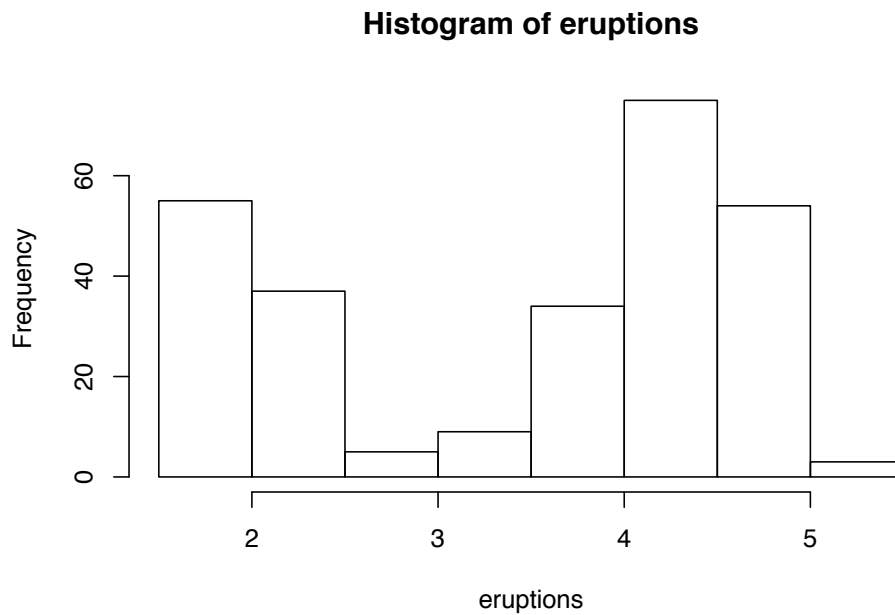
```
## 4 | 3
## 4 | 55566666777788899999
## 5 | 000001111122222333333444444444
## 5 | 5555566667778889999999
## 6 | 0000022223334444
## 6 | 555667899
## 7 | 0000111112333333444444
## 7 | 55555556666666677777777778888888888888889999999999
## 8 | 000000001111111111122222222223333333333344444444444
## 8 | 5555566666677888888999
## 9 | 0000012334
## 9 | 6
```

```
sort(eruptions)
```

```
## [1] 1.600 1.667 1.700 1.733 1.750 1.750 1.750 1.750
## [9] 1.750 1.750 1.783 1.783 1.800 1.800 1.800 1.800
## [17] 1.817 1.817 1.817 1.833 1.833 1.833 1.833 1.833
## [25] 1.833 1.833 1.850 1.850 1.867 1.867 1.867 1.867
## [33] 1.867 1.867 1.867 1.867 1.883 1.883 1.883 1.883
## [41] 1.917 1.917 1.933 1.933 1.950 1.967 1.967 1.967
## [49] 1.983 1.983 1.983 2.000 2.000 2.000 2.000 2.017
## [57] 2.017 2.017 2.033 2.033 2.067 2.083 2.083 2.100
## [65] 2.100 2.100 2.133 2.150 2.167 2.167 2.183 2.200
## [73] 2.200 2.200 2.217 2.233 2.233 2.250 2.250 2.267
## [81] 2.283 2.300 2.317 2.333 2.350 2.367 2.383 2.400
## [89] 2.400 2.417 2.417 2.483 2.617 2.633 2.800 2.883
## [97] 2.900 3.067 3.317 3.333 3.333 3.367 3.417 3.450
## [105] 3.500 3.500 3.567 3.567 3.600 3.600 3.600 3.600
## [113] 3.683 3.717 3.733 3.750 3.767 3.767 3.817 3.833
## [121] 3.833 3.833 3.833 3.833 3.850 3.850 3.883 3.917
## [129] 3.917 3.917 3.950 3.950 3.966 3.967 4.000 4.000
## [137] 4.000 4.000 4.000 4.000 4.033 4.033 4.050 4.067
## [145] 4.067 4.083 4.083 4.083 4.083 4.083 4.100 4.100
## [153] 4.117 4.117 4.133 4.133 4.150 4.150 4.150 4.150
## [161] 4.167 4.167 4.167 4.167 4.183 4.200 4.233 4.233
## [169] 4.233 4.250 4.250 4.250 4.250 4.267 4.267 4.283
## [177] 4.283 4.300 4.300 4.317 4.333 4.333 4.333 4.333
## [185] 4.333 4.350 4.350 4.350 4.350 4.366 4.367 4.367
## [193] 4.367 4.383 4.400 4.417 4.417 4.417 4.417 4.433
## [201] 4.433 4.450 4.450 4.450 4.467 4.467 4.483 4.500
## [209] 4.500 4.500 4.500 4.500 4.500 4.500 4.500 4.517
## [217] 4.533 4.533 4.533 4.533 4.533 4.550 4.567 4.567
## [225] 4.567 4.583 4.583 4.583 4.583 4.600 4.600 4.600
## [233] 4.600 4.617 4.633 4.633 4.633 4.650 4.667 4.667
```

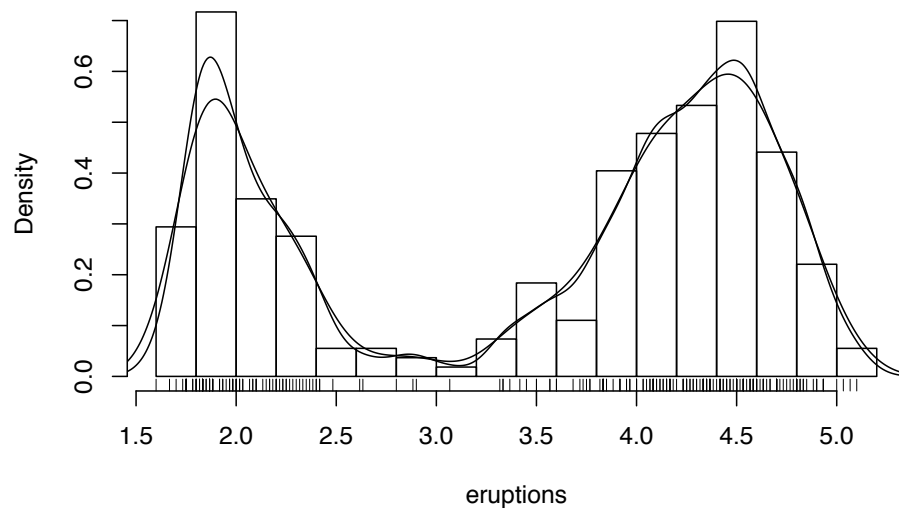
```
## [241] 4.700 4.700 4.700 4.700 4.700 4.700 4.716 4.733
## [249] 4.750 4.767 4.783 4.800 4.800 4.800 4.800 4.800
## [257] 4.800 4.817 4.817 4.833 4.833 4.850 4.883 4.900
## [265] 4.900 4.933 4.933 4.933 5.000 5.033 5.067 5.100
```

```
hist(eruptions)
```



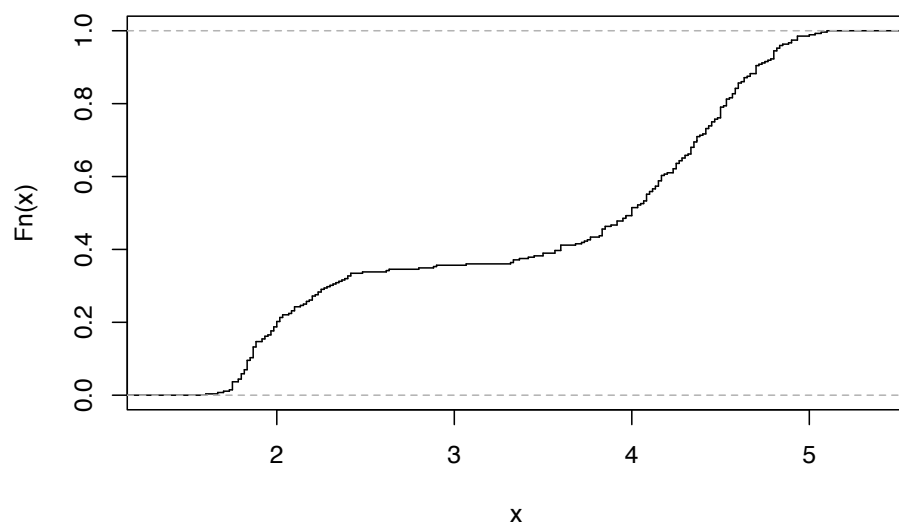
```
hist(eruptions, seq(1.6, 5.2, 0.2), prob=TRUE)
lines(density(eruptions, bw=0.1))
rug(eruptions)
?hist
?density
lines(density(eruptions, bw="SJ"))
```

Histogram of eruptions



```
plot(ecdf(eruptions), do.points=FALSE, verticals=TRUE)
```

ecdf(eruptions)



```
?plot  
ecdf(eruptions)
```

```
## Empirical CDF
```

```
## Call: ecdf(eruptions)
## x[1:126] = 1.6, 1.7, 1.7, ..., 5.1, 5.1
```

```
x = ecdf(eruptions)
x
```

```
## Empirical CDF
## Call: ecdf(eruptions)
## x[1:126] = 1.6, 1.7, 1.7, ..., 5.1, 5.1
```

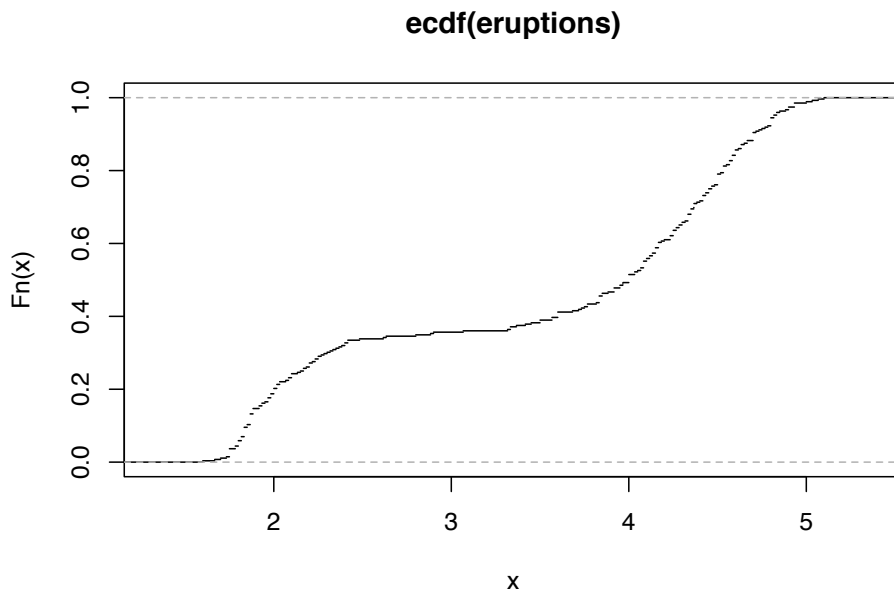
```
str(x)
```

```
## function (v)
## - attr(*, "class")= chr [1:3] "ecdf" "stepfun" "function"
## - attr(*, "call")= language ecdf(eruptions)
```

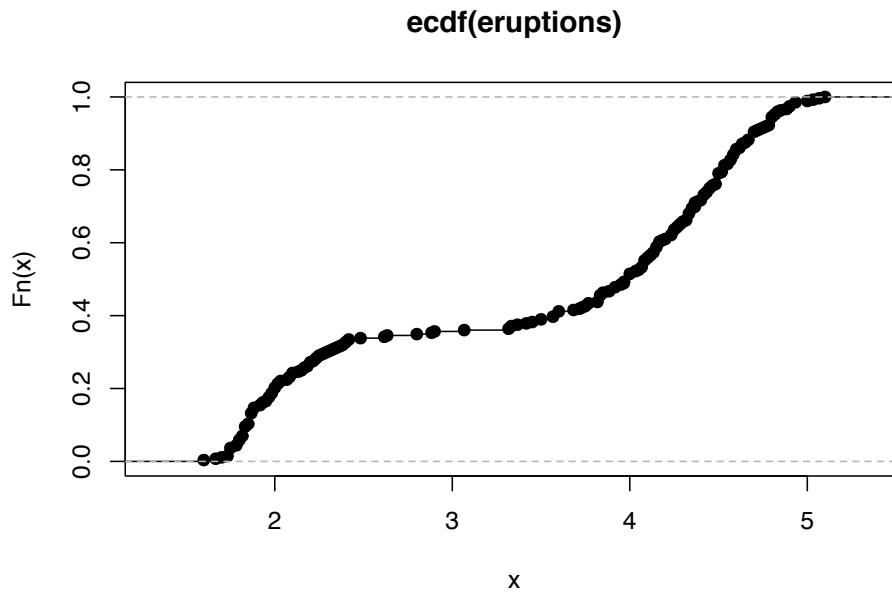
```
x()
```

```
## Error in .approxfun(x, y, v, method, yleft, yright, f): 기본값이 없는 인수 "v"가 누락되어 있습니다
```

```
plot(ecdf(eruptions), do.points=FALSE)
```



```
plot(ecdf(eruptions))
```



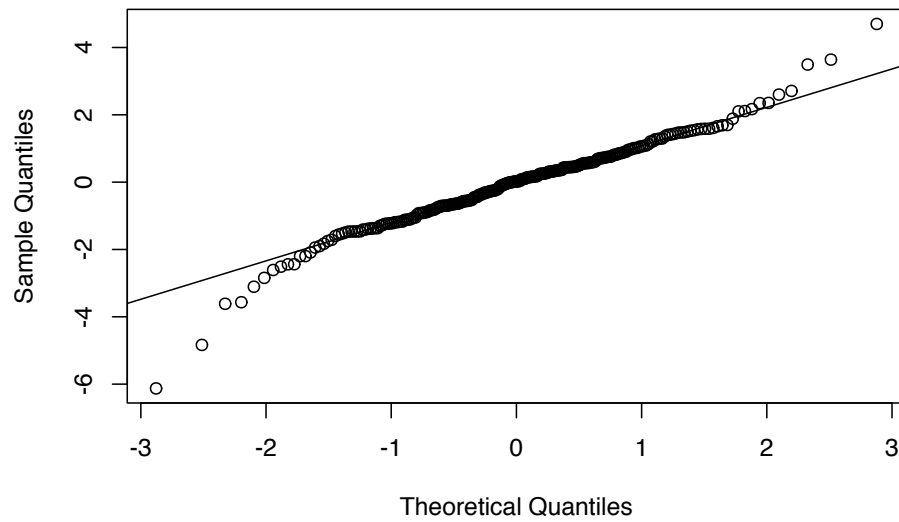
```
long <- eruptions[eruptions > 3]
x <- seq(3, 5.4, 0.01)
pnorm(x, mean=mean(long), sd=sqrt(var(long)))
```

```
## [1] 0.0008362 0.0009084 0.0009864 0.0010704 0.0011610
## [6] 0.0012585 0.0013635 0.0014764 0.0015978 0.0017282
## [11] 0.0018682 0.0020185 0.0021797 0.0023524 0.0025375
## [16] 0.0027356 0.0029476 0.0031743 0.0034165 0.0036752
## [21] 0.0039514 0.0042460 0.0045601 0.0048947 0.0052511
## [26] 0.0056304 0.0060338 0.0064627 0.0069183 0.0074020
## [31] 0.0079152 0.0084596 0.0090365 0.0096475 0.0102944
## [36] 0.0109788 0.0117024 0.0124670 0.0132746 0.0141269
## [41] 0.0150260 0.0159739 0.0169725 0.0180241 0.0191306
## [46] 0.0202945 0.0215177 0.0228028 0.0241519 0.0255674
## [51] 0.0270518 0.0286074 0.0302366 0.0319421 0.0337262
## [56] 0.0355915 0.0375406 0.0395759 0.0417001 0.0439157
## [61] 0.0462253 0.0486315 0.0511367 0.0537436 0.0564547
## [66] 0.0592723 0.0621991 0.0652374 0.0683897 0.0716581
## [71] 0.0750451 0.0785529 0.0821835 0.0859391 0.0898217
## [76] 0.0938331 0.0979753 0.1022500 0.1066587 0.1112030
## [81] 0.1158842 0.1207037 0.1256626 0.1307619 0.1360025
## [86] 0.1413850 0.1469102 0.1525783 0.1583896 0.1643443
## [91] 0.1704423 0.1766832 0.1830667 0.1895922 0.1962589
## [96] 0.2030658 0.2100116 0.2170952 0.2243149 0.2316689
## [101] 0.2391554 0.2467722 0.2545170 0.2623872 0.2703803
```

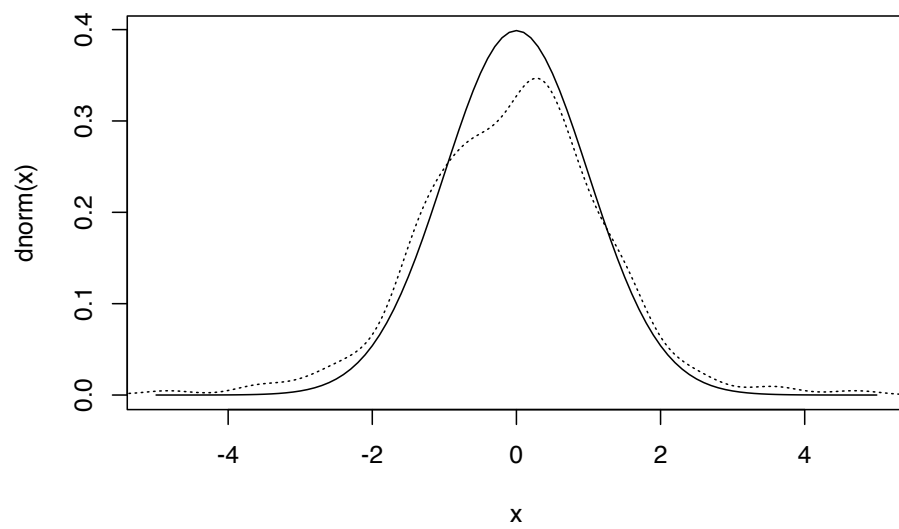
```
## [106] 0.2784932 0.2867229 0.2950662 0.3035195 0.3120794
## [111] 0.3207419 0.3295032 0.3383590 0.3473052 0.3563373
## [116] 0.3654507 0.3746407 0.3839025 0.3932310 0.4026213
## [121] 0.4120681 0.4215661 0.4311100 0.4406943 0.4503134
## [126] 0.4599618 0.4696339 0.4793239 0.4890262 0.4987349
## [131] 0.5084444 0.5181489 0.5278427 0.5375199 0.5471751
## [136] 0.5568024 0.5663963 0.5759512 0.5854617 0.5949224
## [141] 0.6043279 0.6136730 0.6229527 0.6321619 0.6412957
## [146] 0.6503494 0.6593184 0.6681982 0.6769845 0.6856732
## [151] 0.6942601 0.7027416 0.7111139 0.7193735 0.7275172
## [156] 0.7355417 0.7434443 0.7512220 0.7588724 0.7663930
## [161] 0.7737818 0.7810366 0.7881558 0.7951377 0.8019809
## [166] 0.8086842 0.8152465 0.8216671 0.8279453 0.8340805
## [171] 0.8400726 0.8459213 0.8516267 0.8571890 0.8626087
## [176] 0.8678862 0.8730222 0.8780176 0.8828733 0.8875905
## [181] 0.8921703 0.8966142 0.9009236 0.9051002 0.9091456
## [186] 0.9130615 0.9168500 0.9205130 0.9240526 0.9274708
## [191] 0.9307700 0.9339522 0.9370200 0.9399756 0.9428215
## [196] 0.9455601 0.9481939 0.9507254 0.9531571 0.9554916
## [201] 0.9577315 0.9598792 0.9619375 0.9639088 0.9657957
## [206] 0.9676007 0.9693264 0.9709753 0.9725498 0.9740525
## [211] 0.9754857 0.9768519 0.9781534 0.9793926 0.9805717
## [216] 0.9816930 0.9827587 0.9837709 0.9847318 0.9856434
## [221] 0.9865077 0.9873267 0.9881024 0.9888365 0.9895310
## [226] 0.9901874 0.9908077 0.9913933 0.9919460 0.9924672
## [231] 0.9929585 0.9934212 0.9938569 0.9942668 0.9946523
## [236] 0.9950145 0.9953548 0.9956741 0.9959737 0.9962546
## [241] 0.9965177
```

```
?par
x <- rt(250, df = 5)
qqnorm(x); qqline(x)
```

Normal Q-Q Plot



```
curve(dnorm, -5, 5)
y = density(x)
lines(y, lty=3)
```



```
?ppoints
ppoints(250)
```

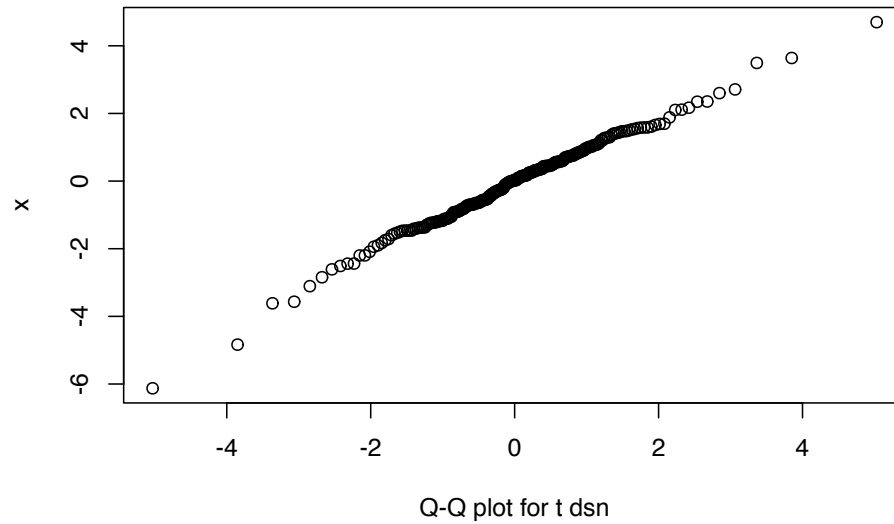
```
## [1] 0.002 0.006 0.010 0.014 0.018 0.022 0.026 0.030
```

```
## [9] 0.034 0.038 0.042 0.046 0.050 0.054 0.058 0.062
## [17] 0.066 0.070 0.074 0.078 0.082 0.086 0.090 0.094
## [25] 0.098 0.102 0.106 0.110 0.114 0.118 0.122 0.126
## [33] 0.130 0.134 0.138 0.142 0.146 0.150 0.154 0.158
## [41] 0.162 0.166 0.170 0.174 0.178 0.182 0.186 0.190
## [49] 0.194 0.198 0.202 0.206 0.210 0.214 0.218 0.222
## [57] 0.226 0.230 0.234 0.238 0.242 0.246 0.250 0.254
## [65] 0.258 0.262 0.266 0.270 0.274 0.278 0.282 0.286
## [73] 0.290 0.294 0.298 0.302 0.306 0.310 0.314 0.318
## [81] 0.322 0.326 0.330 0.334 0.338 0.342 0.346 0.350
## [89] 0.354 0.358 0.362 0.366 0.370 0.374 0.378 0.382
## [97] 0.386 0.390 0.394 0.398 0.402 0.406 0.410 0.414
## [105] 0.418 0.422 0.426 0.430 0.434 0.438 0.442 0.446
## [113] 0.450 0.454 0.458 0.462 0.466 0.470 0.474 0.478
## [121] 0.482 0.486 0.490 0.494 0.498 0.502 0.506 0.510
## [129] 0.514 0.518 0.522 0.526 0.530 0.534 0.538 0.542
## [137] 0.546 0.550 0.554 0.558 0.562 0.566 0.570 0.574
## [145] 0.578 0.582 0.586 0.590 0.594 0.598 0.602 0.606
## [153] 0.610 0.614 0.618 0.622 0.626 0.630 0.634 0.638
## [161] 0.642 0.646 0.650 0.654 0.658 0.662 0.666 0.670
## [169] 0.674 0.678 0.682 0.686 0.690 0.694 0.698 0.702
## [177] 0.706 0.710 0.714 0.718 0.722 0.726 0.730 0.734
## [185] 0.738 0.742 0.746 0.750 0.754 0.758 0.762 0.766
## [193] 0.770 0.774 0.778 0.782 0.786 0.790 0.794 0.798
## [201] 0.802 0.806 0.810 0.814 0.818 0.822 0.826 0.830
## [209] 0.834 0.838 0.842 0.846 0.850 0.854 0.858 0.862
## [217] 0.866 0.870 0.874 0.878 0.882 0.886 0.890 0.894
## [225] 0.898 0.902 0.906 0.910 0.914 0.918 0.922 0.926
## [233] 0.930 0.934 0.938 0.942 0.946 0.950 0.954 0.958
## [241] 0.962 0.966 0.970 0.974 0.978 0.982 0.986 0.990
## [249] 0.994 0.998
```

```
ppoints(10)
```

```
## [1] 0.06098 0.15854 0.25610 0.35366 0.45122 0.54878
## [7] 0.64634 0.74390 0.84146 0.93902
```

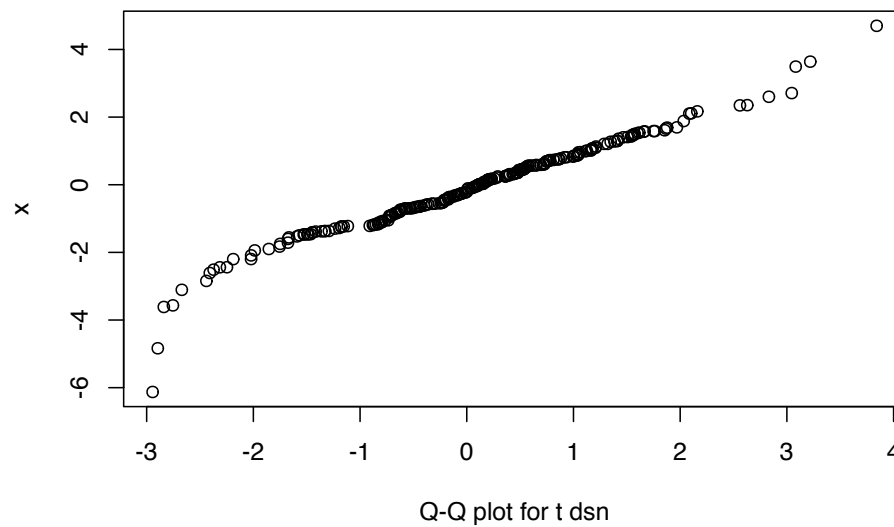
```
qqplot(qt(ppoints(250), df = 5), x, xlab = "Q-Q plot for t dsn")
```

```
windows()
```

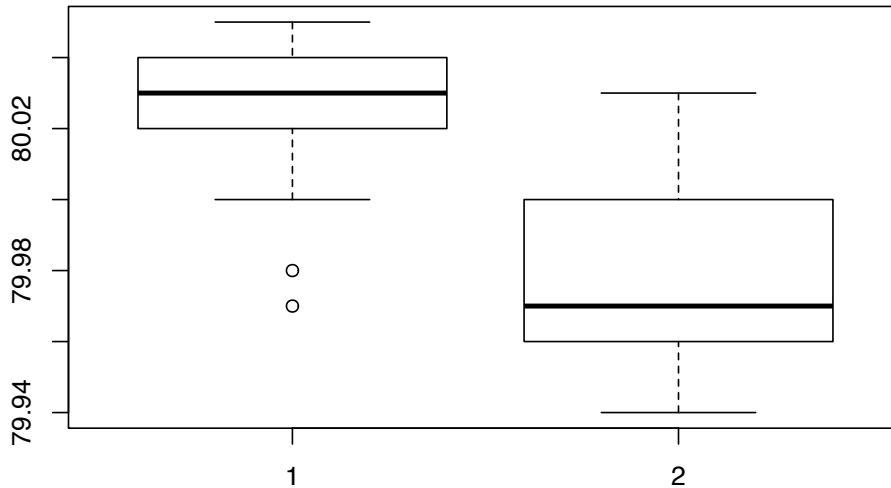
```
## Error in eval(expr, envir, enclos): 함수 "windows"를 찾을 수 없습니다
```

```
qqplot(qt(runif(250), df = 5), x, xlab = "Q-Q plot for t dsn")
```



```
?shapiro.test  
?ks.test  
?t.test
```

```
A = c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.04, 79.97, 80.05, 80.03, 80.02, 80.00, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03, 79.95, 79.97)
boxplot(A, B)
```



```
t.test(A, B)
```

```
##
## Welch Two Sample t-test
##
## data: A and B
## t = 3.2, df = 12, p-value = 0.007
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 0.01386 0.07018
## sample estimates:
## mean of x mean of y
## 80.02 79.98
```

```
var.test(A, B)
```

```
##
## F test to compare two variances
##
## data: A and B
## F = 0.58, num df = 12, denom df = 7, p-value =
## 0.4
```

```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.1251 2.1053
## sample estimates:
## ratio of variances
##           0.5837
```

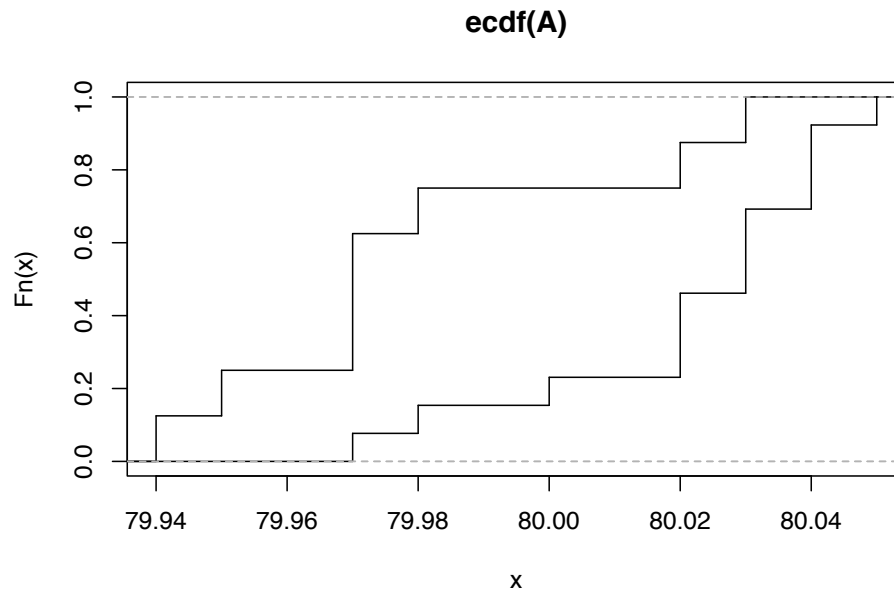
```
t.test(A, B, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: A and B
## t = 3.5, df = 19, p-value = 0.003
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01669 0.06735
## sample estimates:
## mean of x mean of y
##      80.02      79.98
```

```
wilcox.test(A, B)
```

```
## Warning in wilcox.test.default(A, B): cannot compute
## exact p-value with ties
##
## Wilcoxon rank sum test with continuity
## correction
##
## data: A and B
## W = 89, p-value = 0.007
## alternative hypothesis: true location shift is not equal to 0
```

```
plot(ecdf(A), do.points=FALSE, verticals=TRUE, xlim=range(A, B))
plot(ecdf(B), do.points=FALSE, verticals=TRUE, add=TRUE)
```



```
ks.test(A, B)
```

```
## Warning in ks.test(A, B): cannot compute exact p-value
## with ties

##
## Two-sample Kolmogorov-Smirnov test
##
## data: A and B
## D = 0.6, p-value = 0.06
## alternative hypothesis: two-sided
```

```
# Chapter 9 Grouping, loops and conditional execution
```

```
# { } does grouping
# Usefulness of loops: for >> while >> repeat
for (i in 1:10) {
  print(2*i)
}
```

```
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 10
## [1] 12
```

```
## [1] 14
## [1] 16
## [1] 18
## [1] 20
```

```
# # if ~ else ~
# if ( ) {
# # Statements 1
# } else {
# # Statements 2
# }
#
# if ( ) {
# # Statements 1
# } else if ( ) {
# # Statements 2
# } else if ( ) {
# # Statements 3
# } else {
# # Statements 4
# }

#
#

# Chapter 10 Writing your own functions

twosam = function(y1, y2)
{
  n1 = length(y1)
  n2 = length(y2)
  yb1 = mean(y1)
  yb2 = mean(y2)
  s1 = var(y1)
  s2 = var(y2)
  s = ((n1 - 1)*s1 + (n2 - 1)*s2)/(n1 + n2 - 2)
  tst = (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))
  return (tst)
}

x = rnorm(10)
y = rt(10, 5)
```

```
twosam(x, y)
```

```
## [1] 1.708
```

```
T.test = function(y1, y2)
{
  n1 = length(y1)
  n2 = length(y2)
  yb1 = mean(y1)
  yb2 = mean(y2)
  s1 = var(y1)
  s2 = var(y2)
  s = ((n1 - 1)*s1 + (n2 - 1)*s2)/(n1 + n2 - 2)

  tst = (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))
  DF = n1 + n2 - 2
  p.val = pt(tst, df=DF)

  Res = list(tst, DF, p.val)
  names(Res) = c("t", "df", "p-value")

  return (Res)
}
```

```
T.test(x, y)
```

```
## $t
## [1] 1.708
##
## $df
## [1] 18
##
## $`p-value`
## [1] 0.9475
```

```
bslash = function(X, y)
{
  X = qr(X)
  return (qr.coef(X, y))
}

regcoeff = bslash(Xmat, yvar)
```

Error in qr(X): 객체 'Xmat'를 찾을 수 없습니다

```
"%^%" = function(S, pow) with(eigen(S), vectors %*% (abs(values)^pow * t(vectors)))
```

```
M = matrix(c(2,1,1,2), nrow=2) ; M
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    1    2
```

```
M %%% 0.5
```

```
##      [,1] [,2]
## [1,] 1.366 0.366
## [2,] 0.366 1.366
```

```
sqrtM = M%%0.5 ; sqrtM
```

```
##      [,1] [,2]
## [1,] 1.366 0.366
## [2,] 0.366 1.366
```

```
sqrtM %*% sqrtM
```

```
##      [,1] [,2]
## [1,]    2    1
## [2,]    1    2
```

```
area = function(f, a, b, eps=1.0e-06, lim=10)
{
  fun1 = function(f, a, b, fa, fb, a0, eps, lim, fun)
  {
    ## function 'fun1' is only visible inside 'area'
    d = (a + b)/2
    h = (b - a)/4
    fd = f(d)
    a1 = h * (fa + fd)
    a2 = h * (fd + fb)
    if (abs(a0 - a1 - a2) < eps || lim == 0)
      return (a1 + a2)
    else {
      return (fun(f, a, d, fa, fd, a1, eps, lim - 1, fun) + fun(f, d, b, fd, fb, a2, eps, lim - 1, fun))
    }
  }
}
```

```

fa = f(a)
fb = f(b)
a0 = ((fa + fb) * (b - a))/2
fun1(f, a, b, fa, fb, a0, eps, lim, fun1)
}

area(dnorm, 0, 1)

```

```
## [1] 0.3413
```

```
integrate(dnorm, 0, 1)
```

```
## 0.3413 with absolute error < 3.8e-15
```

```
pnorm(1) - pnorm(0)
```

```
## [1] 0.3413
```

```

f = function(x)
{
  y = 2*x
  print(x)
  print(y)
  print(z)
}

f(1)

```

```
## [1] 1
```

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

```
## Error in print(z): 객체 'z'를 찾을 수 없습니다
```

```

z = 3
f(1)

```

```
## [1] 1
```

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

```
## [1] 3
```

```

cube = function(n) {
  sq = function() n*n
  n*sq()
}

```



```
}
```

```
cube(5)
```

```
## [1] 125
```

```
open.account = function(total)
{
  list(
    deposit = function(amount)
    {
      if(amount <= 0)
        stop("Deposits must be positive!\n")
      total <- total + amount
      cat(amount, "deposited. Your balance is", total, "\n\n")
    },
    withdraw = function(amount)
    {
      if(amount > total)
        stop("You don't have that much money!\n")
      total <- total - amount
      cat(amount, "withdrawn. Your balance is", total, "\n\n")
    },
    balance = function()
    {
      cat("Your balance is", total, "\n\n")
    }
  )
}
```

```
ross = open.account(100)
robert = open.account(200)
```

```
ross$balance()
```

```
## Your balance is 100
```

```
robert$balance()
```

```
## Your balance is 200
```

```
ross$deposit(50)
```

```
## 50 deposited. Your balance is 150
```

TABLE 4.1: The basics - The first functions to learn

Keyword	Bae Freq	Essential	Comment
?	H	Y	
str	M	Y	strucutre

TABLE 4.2: The basics - Important operators and assignment

Keyword	Bae Freq	Essential	Comment
%in%	M	Y	Value Matching
match	M	N	Value Matching
=	H	Y	
<-	L	N	
«-	M	Y	
head	H	N	
tail	M	N	
subset	L	N	Subsetting Vectors, Matrices and Data Frames
with	L	N	Evaluate an Expression in a Data Environment
assign	L	N	Assign a Value to a Name
get	L	N	Return the Value of a Named Object

```
ross$balance()
```

```
## Your balance is 150
```

```
ross$withdraw(500)
```

```
## Error in ross$withdraw(500): You don't have that much money!
```

TABLE 4.3: The basics - Comparison

Keyword	Bae Freq	Essential	Comment
all.equal	L	N	Test if Two Objects are (Nearly) Equal
identical	L	N	Test Objects for Exact Equality
!=, ==, >, >=, <, <=	H	Y	Comparison Operator
is.na	H	Y	
complete.cases	L	N	Find Complete Cases
is.finite	M	Y	

4.2 The basics

4.3 Common data structures

4.4 Statistics

4.5 Working with R

4.6 I/O

TABLE 4.4: The basics - Basic math

Keyword	Bae Freq	Essential	Comment
*, +, -, /, ^	H	Y	Math operator
%%	M	Y	Modulus
/%	L	N	Integer division
abs	H	Y	
sign	M	N	
acos	L	Y	
asin	L	Y	
atan	L	Y	
atan2	L	Y	
sin	L	Y	
cos	L	Y	
tan	L	Y	
ceiling	H	Y	
floor	H	N	
round	H	Y	
trunc	H	N	
signif	M	Y	rounds the values in its first argument to the specified number of significant digits
exp	H	Y	
log	H	Y	
log10	L	Y	
log2	L	Y	
sqrt	H	N	
max	H	Y	
min	H	Y	
prod	L	N	
sum	H	Y	
cummax	L	N	
cummin	L	N	
cumprod	L	N	
cumsum	L	N	
diff	L	N	
pmax	L	N	pairwise max
pmin	L	N	pairwise min
range	L	N	
mean	H	Y	
median	H	Y	
cor	H	Y	
sd	H	Y	
var	H	Y	
rle	L	N	Run Length Encoding

TABLE 4.5: The basics - Functions to do with functions

Keyword	Bae Freq	Essential	Comment
function	H	Y	
missing	M	Y	Does a Formal Argument have a Value?
on.exit	L	Y	
return	H	N	
invisible	L	N	Change the Print Mode to Invisible

TABLE 4.6: The basics - Logical - sets

Keyword	Bae Freq	Essential	Comment
&, , !	H	Y	
xor	L	Y	
all	L	Y	Are All Values True?
any	L	Y	Are Some Values True?
intersect	M	Y	
union	M	Y	
setdiff	L	Y	
setequal	L	Y	
which	L	N	Which indices are TRUE?

TABLE 4.7: The basics - Vectors and matrices

Keyword	Bae Freq	Essential	Comment
c	H	Y	
matrix	H	Y	
# automatic coercion rules character > numeric > logical	H	Y	
length	H	Y	
dim	H	Y	
ncol	H	N	
nrow	H	N	
cbind	H	Y	
rbind	H	Y	
names	M	Y	
colnames	H	Y	
rownames	M	Y	
t	H	Y	
diag	H	Y	
sweep	L	N	Sweep out Array Summaries
as.matrix	H	Y	
data.matrix	L	N	Convert a Data Frame to a Numeric Matrix

TABLE 4.8: The basics - Making vectors

Keyword	Bae Freq	Essential	Comment
c	H	Y	
rep	H	Y	
rep_len	L	N	Replicate Elements of Vectors and Lists with length.out
seq	M	Y	
seq_len	L	N	
seq_along	L	N	
rev	M	Y	
sample	H	Y	
choose	H	Y	
factorial	M	Y	
combn	L	N	Generate All Combinations of n Elements, Taken m at a Time
(is/as).(character/numeric/logical/...)	H	Y	

TABLE 4.9: The basics - Lists - data.frames

Keyword	Bae Freq	Essential	Comment
list	H	Y	Flatten Lists
unlist	L	Y	
data.frame	H	Y	
as.data.frame	H	Y	
split	H	Y	
expand.grid	L	N	Create a Data Frame from All Combinations of Factor Variables

TABLE 4.10: The basics - Control flow

Keyword	Bae Freq	Essential	Comment
if	H	Y	Conditional Element Selection
&&	L	Y	
(short circuiting)	L	Y	
for	H	Y	
while	L	N	
next	M	Y	
break	M	Y	
switch	L	Y	
ifelse	L	N	

TABLE 4.11: The basics - Apply - friends

Keyword	Bae Freq	Essential	Comment
lapply	L	N	Apply a Function over a List or Vector
sapply	L	N	user-friendly version and wrapper of lapply
vapply	L	N	similar to sapply, but has a pre-specified type of return value
apply	M	N	Apply Functions Over Array Margins
tapply	L	N	Apply a Function Over a Ragged Array
replicate	L	N	Apply a Function over a List or Vector

TABLE 4.12: Common data structures - Date time

Keyword	Bae Freq	Essential	Comment
ISOdate	L	N	
ISOdatetime	L	N	
strftime	H	Y	Date-time Conversion Functions to and from Character
strptime	H	Y	Date-time Conversion Functions to and from Character
date	M	Y	
difftime	H	Y	
julian	L	Y	Extract Parts of a POSIXt or Date Object
months	L	N	Extract Parts of a POSIXt or Date Object
quarters	L	N	Extract Parts of a POSIXt or Date Object
weekdays	L	N	Extract Parts of a POSIXt or Date Object
library(lubridate)	L	N	

TABLE 4.13: Common data structures - Character manipulation

Keyword	Bae Freq	Essential	Comment
grep	H	Y	Pattern Matching and Replacement
agrep	L	N	Approximate String Matching (Fuzzy Matching)
gsub	M	Y	or use sub
strsplit	H	Y	
chartr	L	N	Character Translation and Casefolding
nchar	M	Y	
tolower	M	Y	
toupper	H	Y	
substr	H	Y	
paste	H	Y	
library(stringr)	L	N	

TABLE 4.14: Common data structures - Factors

Keyword	Bae Freq	Essential	Comment
factor	M	Y	
levels	M	Y	
nlevels	L	N	
reorder	L	N	Reorder Levels of a Factor
relevel	L	N	Reorder Levels of Factor
cut	L	Y	
findInterval	L	N	Find Interval Numbers or Indices
interaction	L	N	Compute Factor Interactions
options(stringsAsFactors = FALSE)	L	N	

TABLE 4.15: Common data structures - Array manipulation

Keyword	Bae Freq	Essential	Comment
array	L	N	Multi-way Arrays
dim	H	Y	
dimnames	M	Y	
aperm	L	N	Array Transposition
library(abind)	L	N	

TABLE 4.16: Statistics - Ordering and tabulating

Keyword	Bae Freq	Essential	Comment
duplicated	L	Y	Determine Duplicated Elements
unique	H	Y	
merge	L	N	
order	H	Y	
rank	L	Y	
quantile	L	Y	Flat Contingency Tables
sort	H	Y	
table	M	Y	
ftable	L	Y	

TABLE 4.17: Statistics - Linear models

Keyword	Bae Freq	Essential	Comment
fitted	L	Y	Extract Model Fitted Values
predict	H	Y	
resid	L	Y	Extract Model Residuals
rstandard	L	Y	
lm	H	Y	Regression Deletion Diagnostics
glm	H	Y	
hat	L	Y	Regression Deletion Diagnostics
influence.measures	M	Y	
logLik	L	Y	
df	M	Y	
deviance	M	Y	
formula	H	Y	
~	H	Y	
I	H	Y	
anova	H	Y	
coef	M	Y	
confint	M	Y	
vcov	H	Y	
contrasts	L	Y	

TABLE 4.18: Statistics - Miscellaneous tests

Keyword	Bae Freq	Essential	Comment
apropos("\\.test\$")	L	Y	

TABLE 4.19: Statistics - Random variables

Keyword	Bae Freq	Essential	Comment
(q, p, d, r) * (beta, binom, cauchy, chisq, exp, f, gamma, geom,	H	Y	

TABLE 4.20: Statistics - Matrix algebra

Keyword	Bae Freq	Essential	Comment
crossprod	L	Y	Matrix Crossproduct
tcrossprod	L	N	Matrix Crossproduct
eigen	H	Y	
qr	L	Y	
svd	L	Y	
%%%	H	Y	
%o%	L	Y	Outer Product of Arrays
outer	H	Y	
rcond	L	N	Compute or Estimate the Condition Number of a Matrix
solve	H	Y	Solve a System of Equations

TABLE 4.21: Working with R - Workspace

Keyword	Bae Freq	Essential	Comment
ls	H	Y	List Objects
exists	M	Y	
rm	M	Y	
getwd	H	Y	
setwd	H	Y	
q	L	Y	
source	H	Y	
install.packages	H	Y	
library	H	Y	
require	H	Y	

TABLE 4.22: Working with R - Help

Keyword	Bae Freq	Essential	Comment
help	L	N	
?	H	Y	
help.search	L	N	
apropos	L	Y	
RSiteSearch	L	N	Search for Key Words or Phrases in Documentation
citation	L	Y	
demo	L	Y	
example	L	Y	
vignette	L	Y	View, List or Get R Source of Package Vignettes

TABLE 4.23: Working with R - Debugging

Keyword	Bae Freq	Essential	Comment
traceback	L	Y	
browser	L	Y	Environment Browser
recover	L	Y	Browsing after an Error
options(error =)	L	Y	
stop	L	Y	Stop Function Execution
warning	H	Y	
message	L	Y	
tryCatch	L	Y	
try	L	Y	

TABLE 4.24: I/O - Output

Keyword	Bae Freq	Essential	Comment
print	H	Y	
cat	H	Y	
message	L	Y	
warning	H	Y	
dput	L	N	Write an Object to a File or Recreate it
format	H	Y	
sink	L	Y	Send R Output to a File
capture.output	L	Y	Send Output to a Character String or File

TABLE 4.25: I/O - Reading and writing data

Keyword	Bae Freq	Essential	Comment
data	L	N	Loads specified data sets, or list the available data sets.
count.fields	L	N	
read.csv	H	Y	
write.csv	H	Y	
read.delim	L	N	
write.delim	L	N	Serialization Interface for Single Objects
read.fwf	L	N	
readLines	M	Y	
writeLines	M	Y	
readRDS	L	N	
saveRDS	L	N	
load	L	Y	
save	L	Y	
library(foreign)	L	N	

TABLE 4.26: I/O - Files and directories

Keyword	Bae Freq	Essential	Comment
dir	H	Y	removes all of the path up to and including the last path separator
basename	L	Y	
dirname	L	Y	
tools::file_ext	L	Y	
file.path	L	Y	
path.expand	L	Y	Expand File Paths
normalizePath	L	Y	Express File Paths in Canonical Form
file.choose	L	Y	
file.copy	L	Y	
file.create	L	Y	
file.remove	L	Y	
file.rename	L	Y	
dir.create	L	Y	
file.exists	L	Y	
file.info	L	Y	
tempdir	L	Y	
tempfile	L	Y	
download.file	L	Y	
library(downloader)	L	N	



5

Rstudio and some useful packages 2 - ggplot2

9주차 강의 예정 자료입니다.

이번 시간에는 ggplot2의 사용법을 예제를 통해 알아보겠습니다. (Wickham and Chang, 2016) <https://rpubs.com/kimwoohyung/ggplot2>의 자료를 많이 참고하였습니다. <https://rpubs.com/mccannecology/53464>도 좋은 자료입니다.

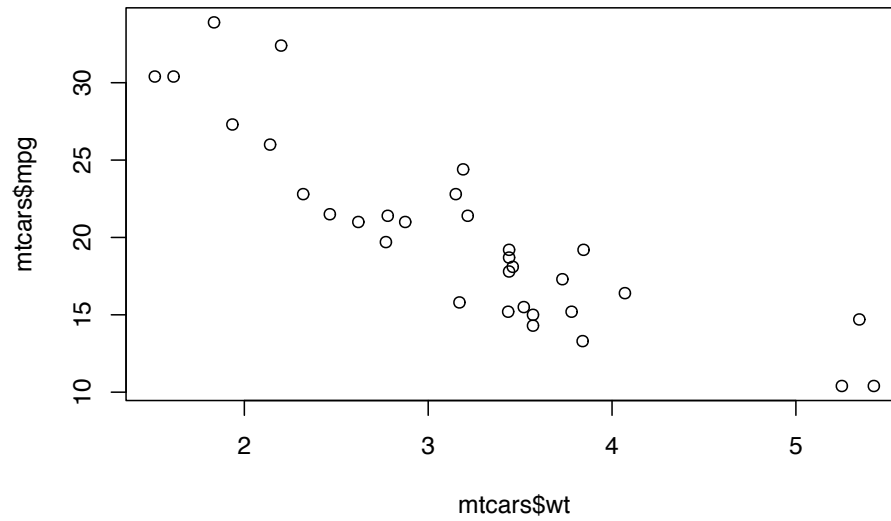
5.1 Introduction

먼저 다음과 같은 패키지가 필요합니다.

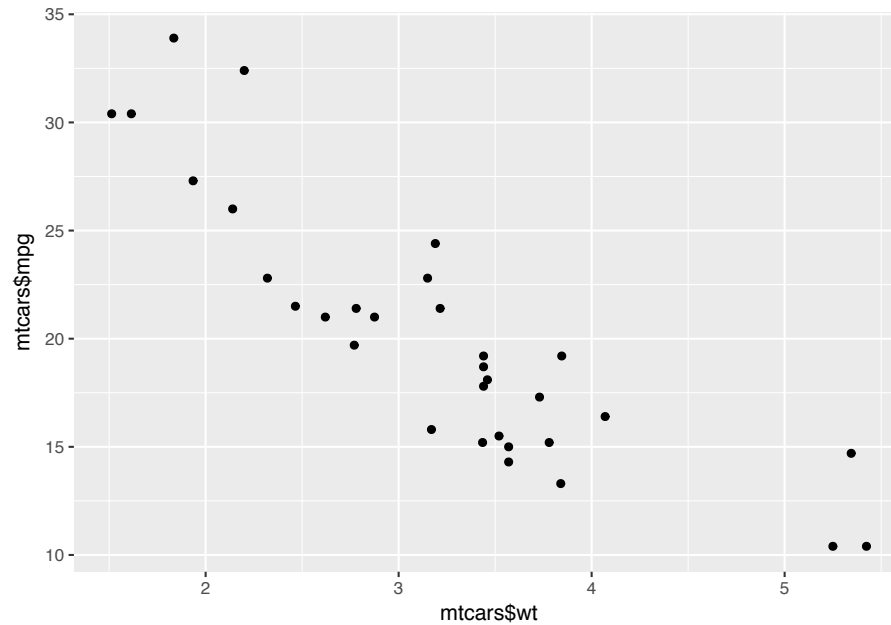
```
library(ggplot2)
library(gcookbook)
library(plyr)
library(reshape2)
```

Scatter plot을 geom_point()를 사용해서 그려보도록 하겠습니다.

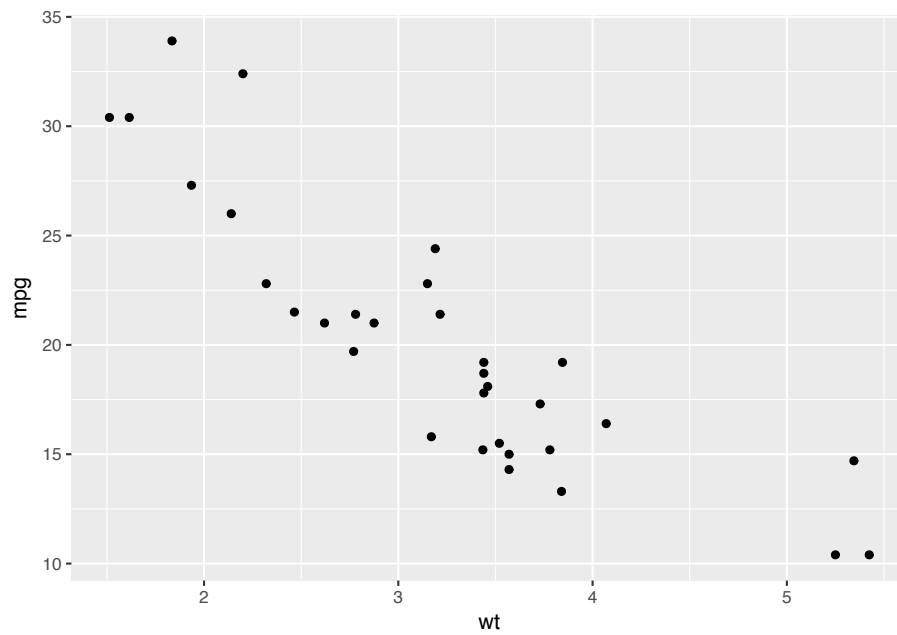
```
plot(mtcars$wt, mtcars$mpg)
```



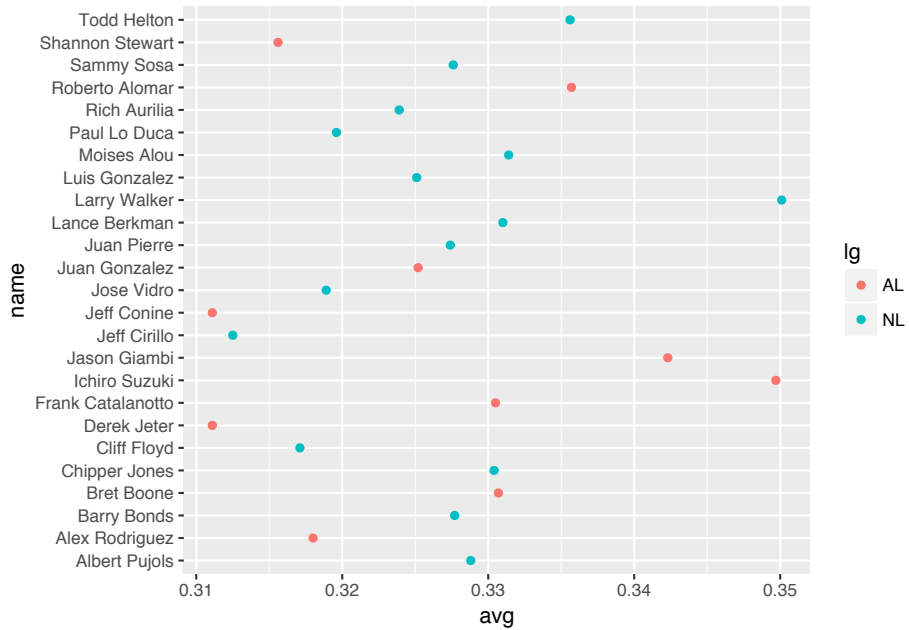
```
qplot(mtcars$wt, mtcars$mpg)
```



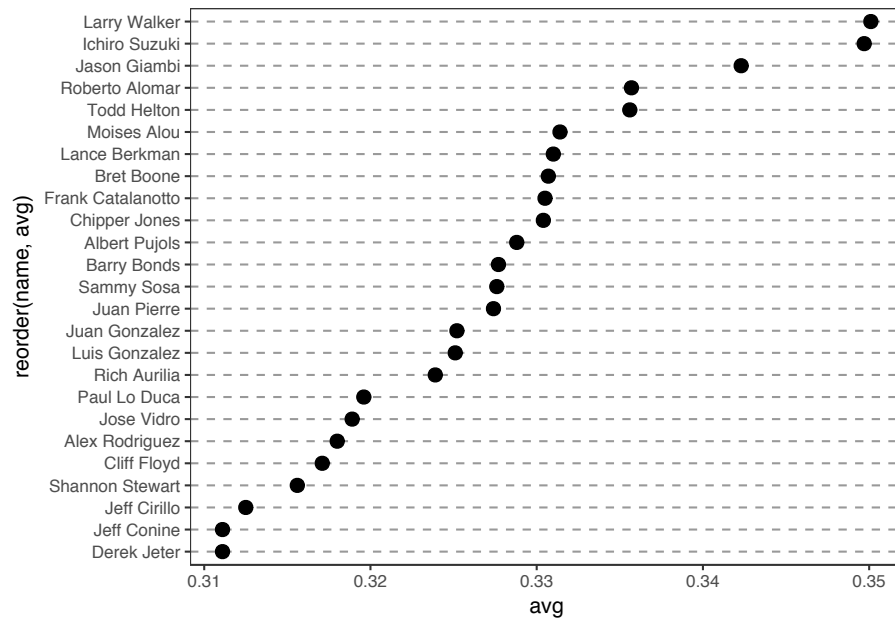
```
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()
```

```
tophit <- tophitters2001[1:25,]  
tophit <- tophit[,c("name", "lg", "avg")]  
  
# y축 이산형 그래프  
ggplot(tophit, aes(x = avg, y = name, col=lg)) + geom_point()
```



```
# 그래프 정렬하기 & 그래프 격자 없애기 & 수평선 점선으로 바꾸기
ggplot(tophit, aes(x = avg, y = reorder(name, avg))) + geom_point(size = 3) + theme_bw() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line(color = "grey60", linetype = "dashed"))
```





A

As-is R Files

교수님께서 주신 원본 R 파일입니다.

A.1 Lecture 3

```
#####  
##-----##  
##                               ##  
##                               Graphics                               ##  
##-----##  
#####  
  
# 상위수준 그림 함수는 그림을 생성한다.  
# 하위수준 그림 함수는 기존의 그림에 그림을 추가한다.  
  
## 상위수준 그림 함수의 주요 인자 (arguments) ###  
  
# main : 제목  
# xlab/ylab : x축 및 y축 레이블  
# xlim/ylim : x축 및 y축 범위  
# col : 색깔  
# lty : 선 모양  
# pch : 점 모양  
# cex : 그림 성분의 크기  
# lwd : 선 굵기  
# type : 그림 타입  
  
#####  
#####      상위수준 그림 함수      #####  
#####
```

```

WD <- "D:\\AMC\\Education\\UU\\2017\\R\\Graphics\\"

setwd(WD)

dta <- read.csv("PK.csv")
head(dta)
str(dta)

##### scatter plot #####

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0])

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], log="y")

plot(dta$TIME[dta$MDV==0], log(dta$DV[dta$MDV==0]))

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0]
      , xlab="Time (hr)", ylab="Concentration (ng/mL)"
      , type="o", pch=2, col=1, main="PK time-course of Drug X"
      , xlim =c(-2,218), ylim=c(0,80))

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], axes=F,
      , xlab="Time (hr)", ylab="Concentration (ng/mL)"
      , type="o", pch=2, col=1, main="PK time-course of Drug X"
      , xlim =c(-2,218), ylim=c(0,80))
axis(1, at=seq(0, 218, 24))
axis(2)
box()

##### Histogram #####

d.demog <- read.csv("DEMOG.csv")

# histogram
hist(d.demog$HT)

hist(d.demog$HT, breaks=10)
hist(d.demog$HT, nclass=10)

# with density line
hist(d.demog$HT, probability=TRUE, breaks=10)
lines(density(d.demog$HT))

```

```

hist (d.demog$HT, probability=TRUE, breaks=9, xaxt="n"
      , main="Histogram for Height", xlab="Height (cm)", ylab="Probability (%)")
axis(1, at=seq(min(d.demog$HT), max(d.demog$HT), 3))
lines(density(d.demog$HT))

hist (d.demog$HT, probability=TRUE, breaks=9, xaxt="n"
      , main="Histogram for Height", xlab="Height (cm)", ylab="Probability (%)")
      , col = "lightblue", border = "pink")
axis(1, at=seq(min(d.demog$HT), max(d.demog$HT), 3))
lines(density(d.demog$HT))

##### Box-Whisker Plot #####

# Box-and-Whisker Plot

boxplot(d.demog$WT)

boxplot(d.demog$WT ~ d.demog$SEX)

boxplot(split(d.demog$WT, d.demog$SEX))

boxplot(WT ~ SEX, data=d.demog)

boxplot(d.demog$WT ~ d.demog$SEX
      , names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
      , col="pink")

boxplot(d.demog$WT ~ d.demog$SEX
      , names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
      , col=c("lightblue", "salmon"), width=c(0.6, 1))

#varwidth: if varwidth is TRUE, the boxes are drawn with widths proportional
#to the square-roots of the number of observations in the groups.

boxplot(d.demog$WT ~ d.demog$SEX
      , names=c("Male", "Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)+2)
      , col=c("lightblue", "salmon")
      , varwidth=TRUE)

```

```
##### Bar Plot #####

barplot(d.demog$HT)

VADeaths

barplot(VADeaths, border = "dark blue")

barplot(VADeaths, col = rainbow(20))

barplot(VADeaths, col = heat.colors(8))

barplot(VADeaths, col = gray.colors(4))

barplot(VADeaths, col = gray.colors(4), log="x")
barplot(VADeaths, col = gray.colors(4), log="y")
barplot(VADeaths, col = gray.colors(4), log="xy")

##### pie chart #####

drug.X.market <- c(0.12, 0.29, 0.32, 0.22, 0.11, 0.28)
names(drug.X.market) <- c("South Korea", "China", "USA", "Japan", "Austria", "EU")
pie(drug.X.market)

##### matplot 함수 #####

# matrix와 column 사이의 그림

pct.95 <- read.csv("pct95.csv")
matplot(pct.95[,1], pct.95[,2:ncol(pct.95)], pch=1)

matplot(pct.95[,1], pct.95[,2:ncol(pct.95)], pch=1, col=c(1,2,1), type="l", lty=1, lwd=c(1,2,1))

##### Scatter plot matrices (pairs plots) #####

pairs(d.demog)

#add a loess smoother, type:
```



```

pairs(d.demog, panel = panel.smooth)

panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r = (cor(x, y))
  txt <- format(c(r, 0.123456789), digits=digits)[1]
  txt <- paste(prefix, txt, sep="")
  if(missing(cex.cor)) cex <- 1.5
  text(0.5, 0.5, txt, cex = 1.5)
}

pairs(d.demog, lower.panel=panel.smooth, upper.panel=panel.cor)

#####
##          하위수준 그림 함수          ##
#####

# points : 점추가
# lines : 선 추가
# abline : 기준선 추가
# mtext : 텍스트 추가
# legend : 설명(legend) 추가
# polygon : polygon 추가

##### 점, 선, 설명 추가 하기 #####

plot(pct.95$TIME, pct.95$PCT50, main="PK of Drug X"
     , type="l", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80), lty=1, col="red", lwd=2)

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
     , lty=1, col="black")

```

```
##### polygon 함수 #####

plot(c(1, 10), c(1, 6), type = "n")
polygon(c(2,8,8,2), c(5,4,3,2), col="lightgreen")

plot(c(1, 9), 1:2, type = "n")
polygon(1:9, c(2,1,2,1,1,2,1,2,1),
        col = c("red", "blue"),
        border = c("green", "yellow"),
        lwd = 3, lty = c("dashed", "solid"))

##### 그림 출력하기 #####

##--pdf graphics devices
pdf("PK_of_Drug_X.pdf")

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
      , lty=1, col="black")

dev.off()

##--PNG graphics devices
png("PK_of_Drug_X.png")

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
      , lty=1, col="black")
```

```

dev.off()

#--Windows graphics devices
win.metafile("PK_of_Drug_X.wmf")

plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
      , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
      , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40, 0, col="red", lty=2) #abline(a,b): y=a+b*x
legend("topright", legend=c("Individual concentrations")
      , lty=1, col="black")

dev.off()

```

A.2 Lecture 4

```

# 2017-03-29

setwd("D:/Rt")
dir()

mydata = read.csv("MyData2017.csv", as.is=TRUE)

Theoph
library(lattice) # trellis

xyplot(conc ~ Time | Subject, data=Theoph)

xyplot(conc ~ Time | Subject, data=Theoph, type="b")

Theoph[, "ID"] = as.numeric(as.character(Theoph[, "Subject"]))

xyplot(conc ~ Time | ID, data=Theoph, type="b")

```

```

xyplot(conc ~ Time | as.factor(ID), data=Theoph, type="b")

write.csv(Theoph, "Theoph.csv", row.names=FALSE, quote=FALSE, na="")

IDs = sort(unique(Theoph[, "ID"])) ; IDs
nID = length(IDs) ; nID

demog = unique(Theoph[, c("ID", "Wt")])
colnames(demog) = c("ID", "BWT")
write.csv(demog, "1-demog.csv", row.names=FALSE, quote=FALSE, na="")

DV = Theoph[, c("ID", "Time", "conc")]
colnames(DV) = c("ID", "TIME", "DV")
write.csv(DV, "3-DV.csv", row.names=FALSE, quote=FALSE, na="")

adm = cbind(IDs, rep(0, nID), rep(320, nID))
colnames(adm) = c("ID", "TIME", "AMT")
write.csv(adm, "2-adm.csv", row.names=FALSE, quote=FALSE, na="")

demog = read.csv("1-demog.csv", as.is=TRUE)
adm = read.csv("2-adm.csv", as.is=TRUE)
dv = read.csv("3-dv.csv", as.is=TRUE)

AdmDv = merge(adm, dv, by=intersect(colnames(adm), colnames(dv)), all=TRUE)

DataAll = merge(demog, AdmDv, by=c("ID"), all=TRUE)

```

A.3 Lecture 5

```
# 2017-04-05 R-intro.pdf Chapter 08
```

```

pois
?dbeta
dnorm(0)
pnorm(0)
1 - pnorm(1.96)

```

```
?pnorm
pnorm(1.96, lower.tail=FALSE)
qnorm(0.5)
qnorm(0.975)
format(qnorm(0.975), digits=22)
rnorm(5)
rnorm(5, 10, 1)
x = rnorm(100, 10, 1)
mean(x)
sd(x)

2*pt(-2.43, df = 13)

2*pt(-2.43, df = 1000)

qnorm(0.995)
qf(0.01, 2, 7, lower.tail = FALSE)

?fivenum
faithful
str(faithful)
eruptions
attach(faithful)
eruptions
waiting

stem(waiting)
sort(eruptions)

hist(eruptions)
hist(eruptions, seq(1.6, 5.2, 0.2), prob=TRUE)
lines(density(eruptions, bw=0.1))
rug(eruptions)
?hist
?density
lines(density(eruptions, bw="SJ"))
plot(ecdf(eruptions), do.points=FALSE, verticals=TRUE)
?plot
ecdf(eruptions)
x = ecdf(eruptions)
x
str(x)
x()
plot(ecdf(eruptions), do.points=FALSE)
```

```

plot(ecdf(eruptions))
long <- eruptions[eruptions > 3]
x <- seq(3, 5.4, 0.01)
pnorm(x, mean=mean(long), sd=sqrt(var(long)))

?par
x <- rt(250, df = 5)
qqnorm(x); qqline(x)
curve(dnorm, -5, 5)
y = density(x)
lines(y, lty=3)
?ppoints
ppoints(250)
ppoints(10)

qqplot(qt(ppoints(250), df = 5), x, xlab = "Q-Q plot for t dsn")
windows()
qqplot(qt(runif(250), df = 5), x, xlab = "Q-Q plot for t dsn")
?shapiro.test
?ks.test
?t.test

A = c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.04, 79.97, 80.05, 80.03, 80.02, 80.00, 80.02)
B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03, 79.95, 79.97)
boxplot(A, B)
t.test(A, B)

var.test(A, B)
t.test(A, B, var.equal=TRUE)
wilcox.test(A, B)
plot(ecdf(A), do.points=FALSE, verticals=TRUE, xlim=range(A, B))
plot(ecdf(B), do.points=FALSE, verticals=TRUE, add=TRUE)
ks.test(A, B)

# Chapter 9 Grouping, loops and conditional execution

# { } does grouping
# Usefulness of loops: for >> while >> repeat
for (i in 1:10) {
  print(2*i)
}

# # if ~ else ~

```

```
# if ( ) {  
# # Statements 1  
# } else {  
# # Statements 2  
# }  
#  
# if ( ) {  
# # Statements 1  
# } else if ( ) {  
# # Statements 2  
# } else if ( ) {  
# # Statements 3  
# } else {  
# # Statements 4  
# }  
  
#  
#  
  
# Chapter 10 Writing your own functions  
  
twosam = function(y1, y2)  
{  
  n1 = length(y1)  
  n2 = length(y2)  
  yb1 = mean(y1)  
  yb2 = mean(y2)  
  s1 = var(y1)  
  s2 = var(y2)  
  s = ((n1 - 1)*s1 + (n2 - 1)*s2)/(n1 + n2 - 2)  
  tst = (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))  
  return (tst)  
}  
  
x = rnorm(10)  
y = rt(10, 5)  
  
twosam(x, y)  
  
T.test = function(y1, y2)  
{  
  n1 = length(y1)  
  n2 = length(y2)
```

```

yb1 = mean(y1)
yb2 = mean(y2)
s1 = var(y1)
s2 = var(y2)
s = ((n1 - 1)*s1 + (n2 - 1)*s2)/(n1 + n2 - 2)

tst = (yb1 - yb2)/sqrt(s*(1/n1 + 1/n2))
DF = n1 + n2 - 2
p.val = pt(tst, df=DF)

Res = list(tst, DF, p.val)
names(Res) = c("t", "df", "p-value")

return (Res)
}

T.test(x, y)

bslash = function(X, y)
{
  X = qr(X)
  return (qr.coef(X, y))
}

regcoeff = bslash(Xmat, yvar)

"%^%" = function(S, pow) with(eigen(S), vectors %*% (abs(values)^pow * t(vectors)))

M = matrix(c(2,1,1,2), nrow=2) ; M
M %^% 0.5
sqrtM = M%^%0.5 ; sqrtM
sqrtM %*% sqrtM

area = function(f, a, b, eps=1.0e-06, lim=10)
{
  fun1 = function(f, a, b, fa, fb, a0, eps, lim, fun)
  {
    ## function 'fun1' is only visible inside 'area'

```



```

    d = (a + b)/2
    h = (b - a)/4
    fd = f(d)
    a1 = h * (fa + fd)
    a2 = h * (fd + fb)
    if (abs(a0 - a1 - a2) < eps || lim == 0)
        return (a1 + a2)
    else {
        return (fun(f, a, d, fa, fd, a1, eps, lim - 1, fun) + fun(f, d, b, fd, fb, a2, eps, lim - 1, fun))
    }
}
fa = f(a)
fb = f(b)
a0 = ((fa + fb) * (b - a))/2
fun1(f, a, b, fa, fb, a0, eps, lim, fun1)
}

area(dnorm, 0, 1)
integrate(dnorm, 0, 1)
pnorm(1) - pnorm(0)

f = function(x)
{
    y = 2*x
    print(x)
    print(y)
    print(z)
}

f(1)
z = 3
f(1)

cube = function(n) {
    sq = function() n*n
    n*sq()
}

cube(5)

open.account = function(total)
{

```

```
list(  
  deposit = function(amount)  
  {  
    if(amount <= 0)  
      stop("Deposits must be positive!\n")  
    total <- total + amount  
    cat(amount, "deposited. Your balance is", total, "\n\n")  
  },  
  withdraw = function(amount)  
  {  
    if(amount > total)  
      stop("You don't have that much money!\n")  
    total <- total - amount  
    cat(amount, "withdrawn. Your balance is", total, "\n\n")  
  },  
  balance = function()  
  {  
    cat("Your balance is", total, "\n\n")  
  }  
)  
}  
  
ross = open.account(100)  
robert = open.account(200)  
  
ross$balance()  
robert$balance()  
ross$deposit(50)  
ross$balance()  
ross$withdraw(500)
```

B

R Tips

- Handbook of statistical distributions with applications¹ (Krishnamoorthy, 2006)
- Materials
 - <https://cran.r-project.org/manuals.html>
 - <https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>
- Changing default R console size and etc : 배균섭 교수님께서 알려주신 tip을 참고하여 video clip을 만들었습니다. 매일같이 마주하게 되는 R console이 너무 작게 느껴졌다면 다음의 동영상을 참고하셔서 초기 세팅 (Rconsole 파일)을 바꿔서 해결할 수 있습니다. <https://youtu.be/uSunEN8W5Mo>

B.1 Using Coursera

PAGK에 보낸 이메일을 그대로 옮겼습니다.

배균섭 교수님의 추천을 받아 다음과 같은 강의와 책을 공유하고자 합니다. Coursera.com에 유익한 R 강좌가 열렸습니다. “Mastering Software Development in R Specialization”이란 제목의 강좌인데 4개 Course를 무료로 들을 수 있게 되어있습니다. <https://www.coursera.org/specializations/r> 이것이 본래의 Link인데 여기서 각각의 Course를 찾거나 혹은 아래의 링크를 각각 클릭하여 하단에 나오는 “Audit”을 클릭하면 무료로 들을 수 있습니다.

Audit 버튼이 안보인다고 하신 분들이 몇분 계셔서 첨언합니다. Coursera 회원가입하시고 로그인 한 뒤, Enroll Now를 누르시면 Audit 혹은 청강하기 라디오버튼을 보실 수 있습니다. 앱에서도 마찬가지입니다. 이외에도 코세라에는 많은 유익한 강의가 있는 것 같습니다. 다만 코스(Course)의 묶음인 “Specialization”에서는 유료등록(Enroll) 밖에

¹<http://www.stat.rice.edu/~dobelman/textfiles/DistributionsHandbook.pdf>

없으므로 Certificate가 필요하지 않다면, 각각의 코스를 구글검색 혹은 코세라 내에서 검색해서 “Audit(청강)” 하시면 무료로 강의를 들을 수 있습니다.

- <https://www.coursera.org/learn/r-programming-environment>
- <https://www.coursera.org/learn/advanced-r>
- <https://www.coursera.org/learn/r-packages>
- <https://www.coursera.org/learn/r-data-visualization>

<https://bookdown.org/rdpeng/RProgDA/> 이 링크는 무료로 공개된 강의 책자입니다. 강의를 듣지 않고 책으로 보고 싶으신 분은 참고하시면 됩니다.

C

Acknowledgement

이 웹북을 만드는데 도움을 주신 분들은 다음과 같습니다.

1. Dr. Jekyll
2. Hyde



Bibliography

Krishnamoorthy, K. (2006). *Handbook of statistical distributions with applications*. Chapman & Hall/CRC, Boca Raton.

Sarkar, D. (2017). *lattice: Trellis Graphics for R*. R package version 0.20-35.

Wickham, H. and Chang, W. (2016). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 2.2.1.

Xie, Y. (2016). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.3.16.



Index

bookdown, [x](#)

Coursera, [xiii](#)

Github, [x](#)

lattice, [28](#)