

R Programming - Lecture Notes



R

PROGRAMMING

KYUN-SEOP BAE
SUNGPIL 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	49
4.3 Common data structures	49
4.4 Statistics	49

4.5	Working with R	49
4.6	I/O	49
5	Rstudio and some useful packages 2 - ggplot2	59
5.1	Introduction	59
5.2	Part1	62
5.3	Part2	79
5.4	Part3	92
5.5	Part4	105
	Appendix	131
.1	Assignment 1	131
.1.1	concUnitConv-test.R	131
.1.2	concUnitConv-test.Rout	131
A	As-is R Files	135
A.1	Lecture 3	135
A.2	Lecture 4	141
A.3	Lecture 5	142
B	R Tips	151
B.1	Using Coursera	151
C	Acknowledgement	153
	Bibliography	155
	Index	157

List of Tables

4.1	The basics - The first functions to learn	46
4.2	The basics - Important operators and assignment	47
4.3	The basics - Comparison	47
4.4	The basics - Basic math	48
4.5	The basics - Functions to do with functions	49
4.6	The basics - Logical - sets	49
4.7	The basics - Vectors and matrices	50
4.8	The basics - Making vectors	50
4.9	The basics - Lists - data.frames	51
4.10	The basics - Control flow	51
4.11	The basics - Apply - friends	51
4.12	Common data structures - Date time	52
4.13	Common data structures - Character manipulation	52
4.14	Common data structures - Factors	52
4.15	Common data structures - Array manipulation	53
4.16	Statistics - Ordering and tabulating	53
4.17	Statistics - Linear models	54
4.18	Statistics - Miscellaneous tests	54
4.19	Statistics - Random variables	54
4.20	Statistics - Matrix algebra	55
4.21	Working with R - Workspace	55
4.22	Working with R - Help	55
4.23	Working with R - Debugging	56
4.24	I/O - Output	56
4.25	I/O - Reading and writing data	57
4.26	I/O - Files and directories	57

List of Figures

1	Creative Commons License	x
2	Syllabus page 1	xiv
3	Syllabus page 1	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 않습니다만, 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 인정해 드립니다.

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.kr⁵ 로 전체 메일 보내 시거나 교수님 k@acr.kr⁶ 혹은 제 개인 메일 shan@acr.kr⁷ 연락해 주십시오.

교수님께서 세우신 방침에 따라 수업시간에 출석을 부르지 않을 예정입니다. 수강하시는 화면 (Skype) 을 휴대폰으로 사진 찍으시거나 강의실의 스크린을 사진으로 촬영하셔서 sec@acr.kr⁸ / shan@acr.kr⁹ 로 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@acr.kr>

⁸<mailto:sec@acr.kr>

⁹<mailto:shan@acr.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-on이 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 출석
이 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 녹화
하여 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)
을 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 assign하게 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

2017년 4월 10일 개정된 수업계획서입니다.

2017-1학기 수업계획서(Course Outline)

년도-학기 (year-semester)	2017-1	과목명 (course name)	R 프로그래밍 R Programming					
과목번호-분반 (courseNo-classNo)	WA5493 - 01	학점(강의-실습) (credit)	3학점(3-0)	ABEEK(설계학점) (Abeek credit)				
담당교수 (professor)	배균섭 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 알려져 있지 않는 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.)
성적비율(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) 빅데이터 분석 도구 R 프로그래밍 노만 메트로프 지음 옮김 의왕 : 에이콘 9788960773332; 9788960772793(set) 2. [부교재] [도서판] R Cookbook : 데이터 분석과 통계, 그래픽스를 위한 실전 예제 폴 터터 지음;이재원 옮김 서울: 인 9788966260379 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 : SI New York 9780387283906 6. [부교재] [도서판] [ebook]Introduction to Scientific Programming and Simulation Using R, Second Edition [electronic resource] Jones O, Maillardet 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		

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)] (석가탄신일 휴무)	[학습자료(materials)]
제10주 (week 10)	[주별진도(topic)] Rstudio and some useful packages 2 - ggplot2	[학습자료(materials)]
제11주 (week 11)	[주별진도(topic)] Some useful packages 3 - dplyr, tidyr	[학습자료(materials)]
제12주 (week 12)	[주별진도(topic)] Some useful packages 4 - lubridate, stringr	[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)] 과제 solution 예시 & feedback	[학습자료(materials)]
제16주 (week 16)	[주별진도(topic)] Pitfalls and limitations of R	[학습자료(materials)]

FIGURE 3: Syllabus page 1

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

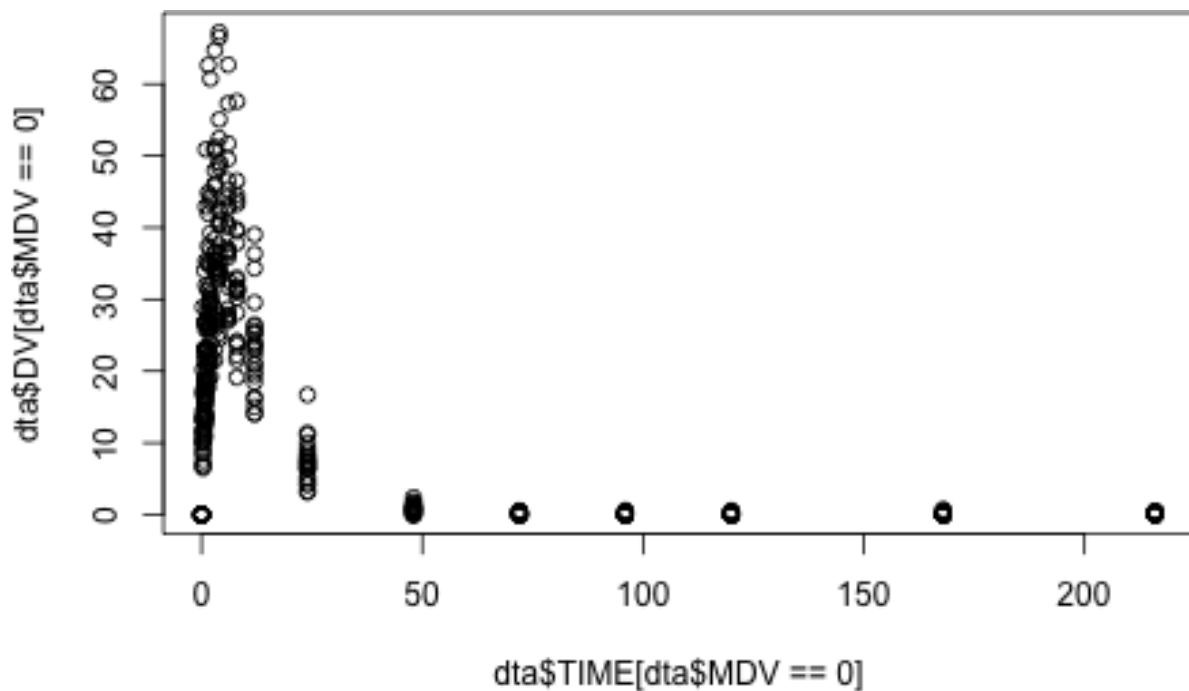
2.2.1 상위수준 그림 함수의 주요 인자 (arguments)

- 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 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 0 ...
##  $ DV   : num  0 0 9.4 13.7 16.5 ...
##  $ MDV  : num  0 1 0 0 0 0 0 0 0 0 ...
```

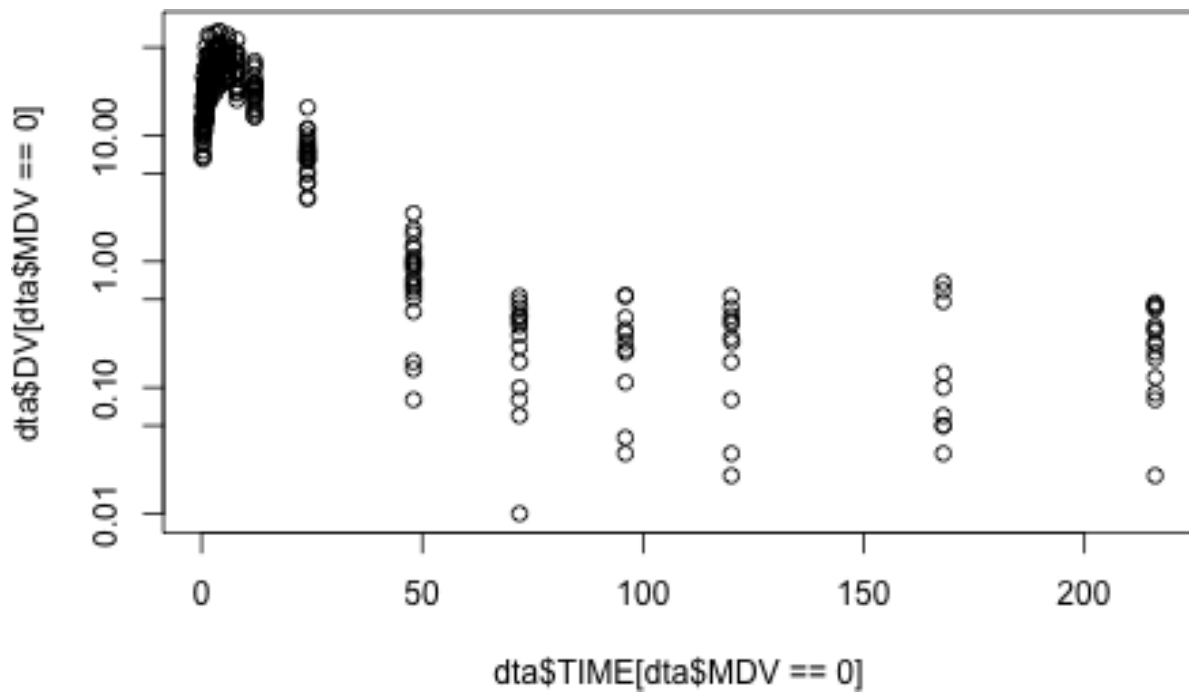
2.2.2 scatter plot

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

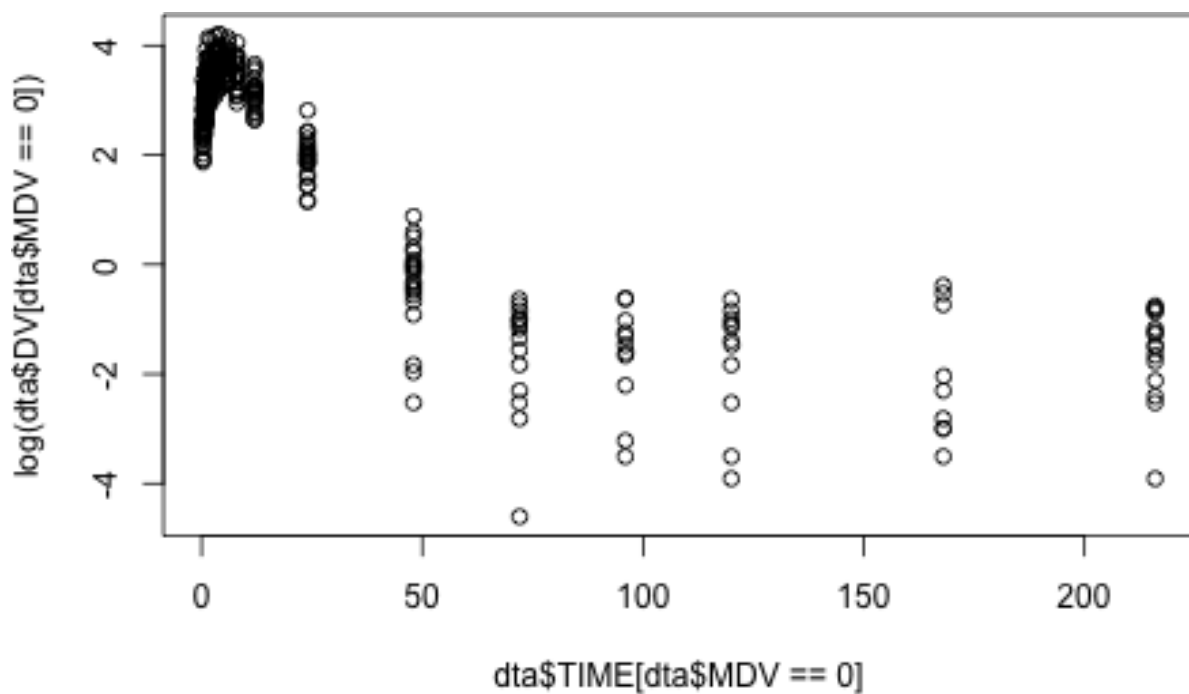


```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], log="y")
```

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

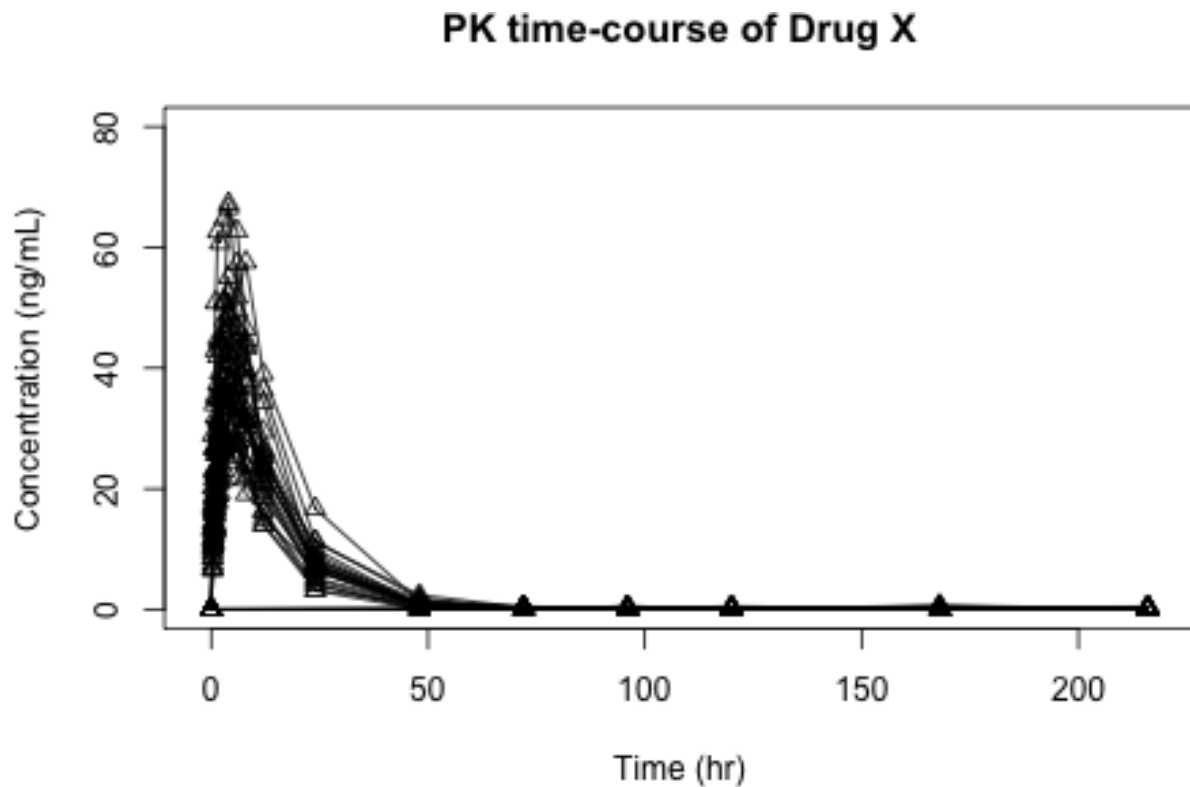


```
plot(dta$TIME[dta$MDV==0], log(dta$DV[dta$MDV==0]))
```

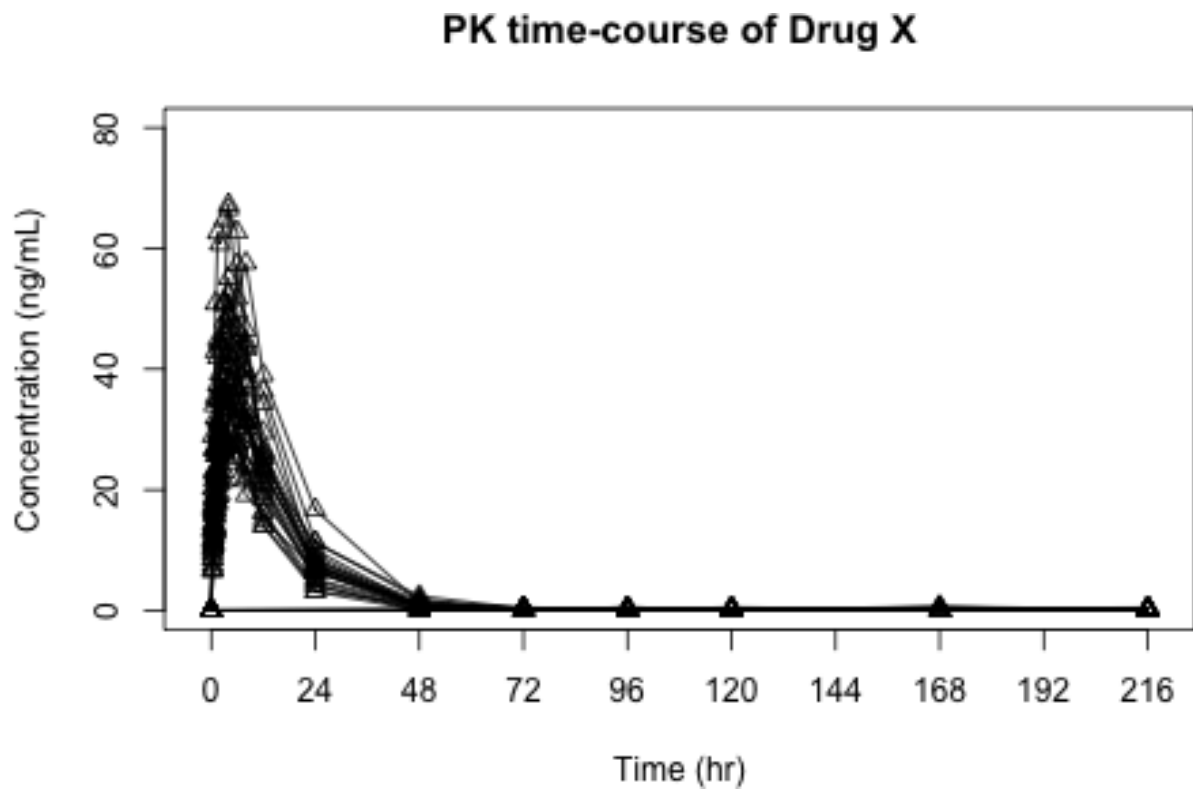


```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0],
     , xlab="Time (hr)", ylab="Concentration (ng/mL)"
```

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



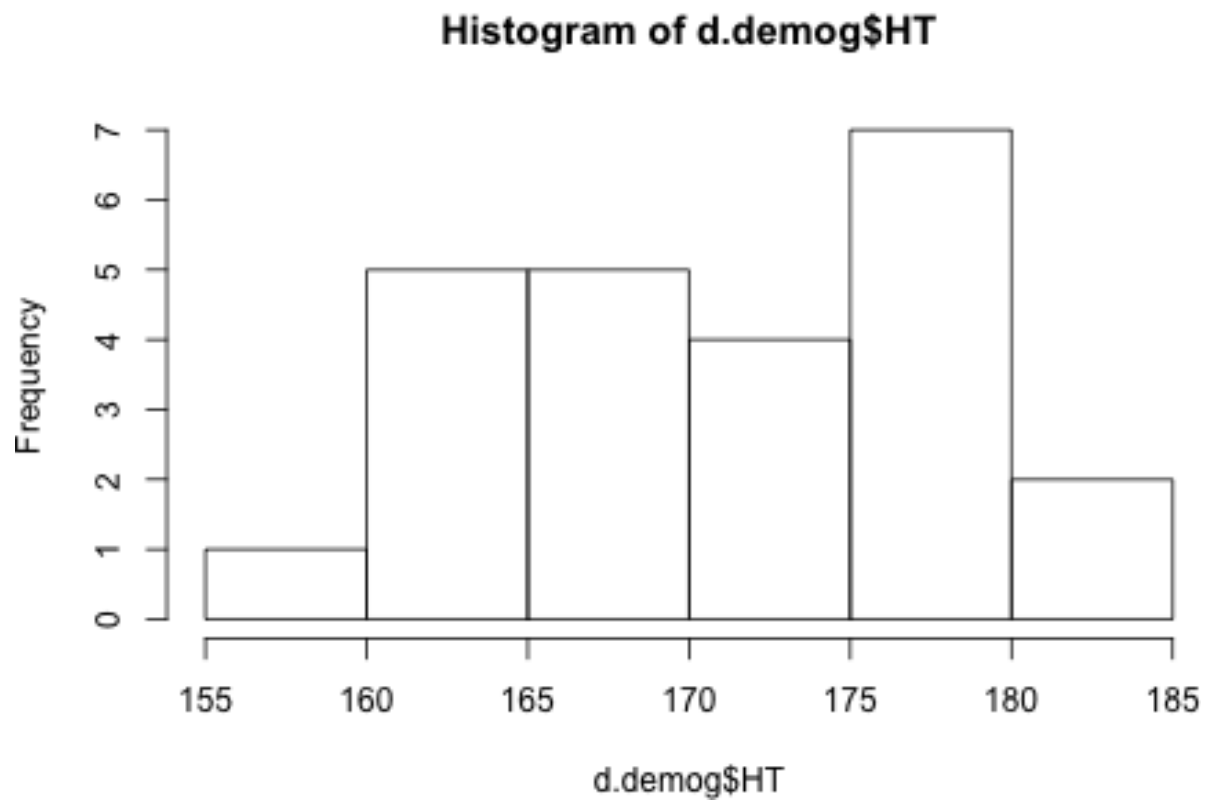
```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], axes=F,
, xlab="Time (hr)", ylab="Concentration (ng/mL)"
, type="o", pch=2, col=1, main="PK time-course of Drug X"
, xlim =c(-2,218), ylim=c(0,80))
axis(1, at=seq(0, 218, 24))
axis(2)
box()
```



2.2.3 Histogram

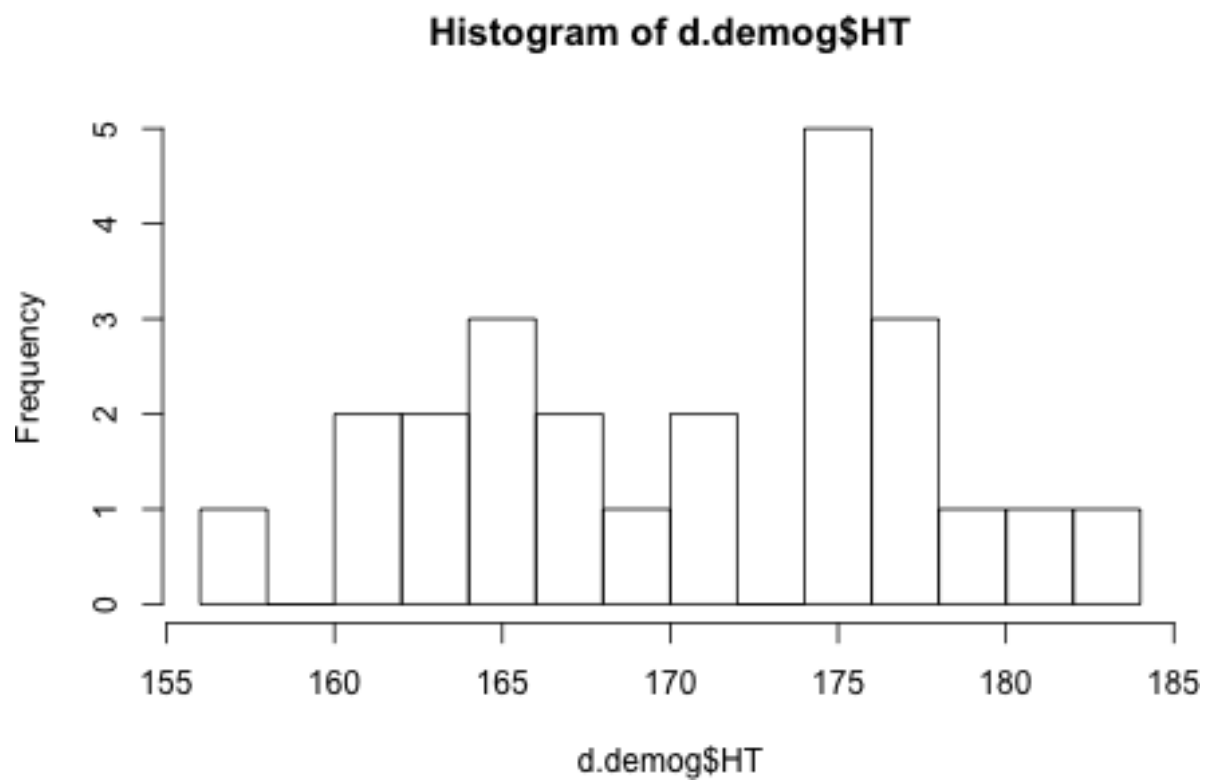
```
d.demog <- read.csv("DEMOG.csv")
```

```
hist(d.demog$HT)
```



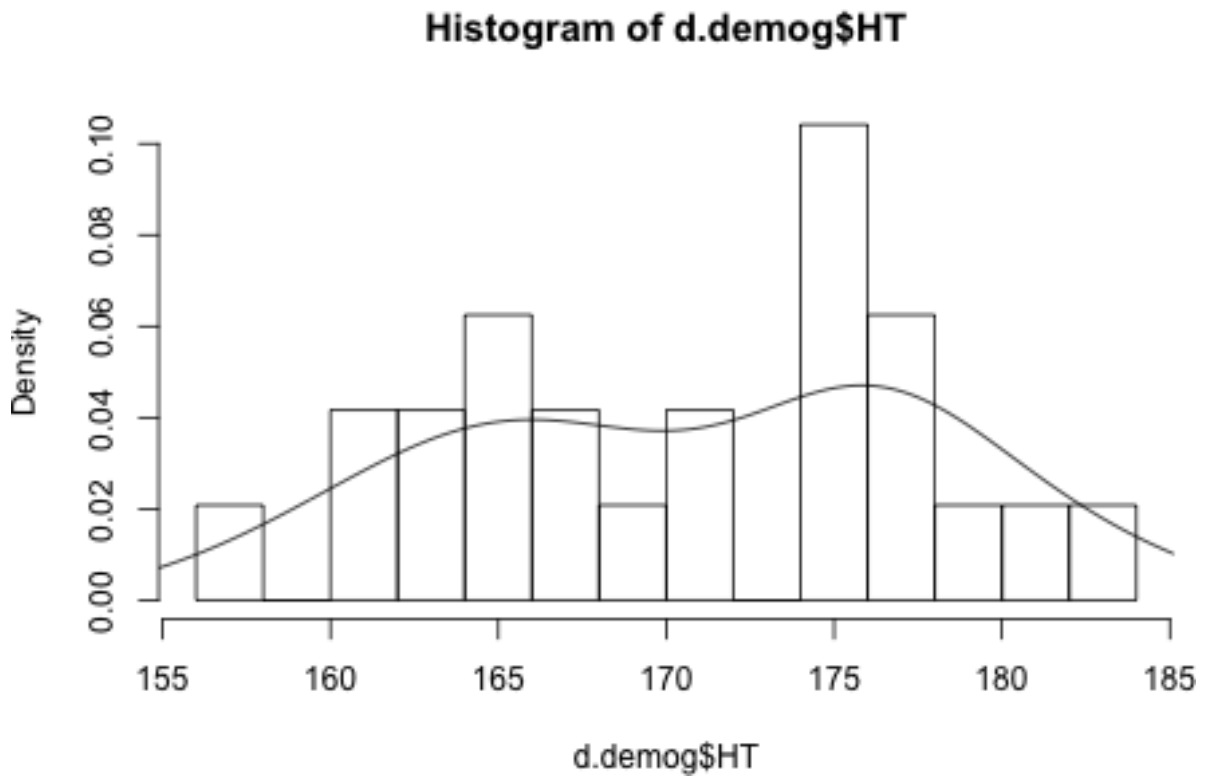
```
hist(d.demog$HT, breaks=10)
```

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

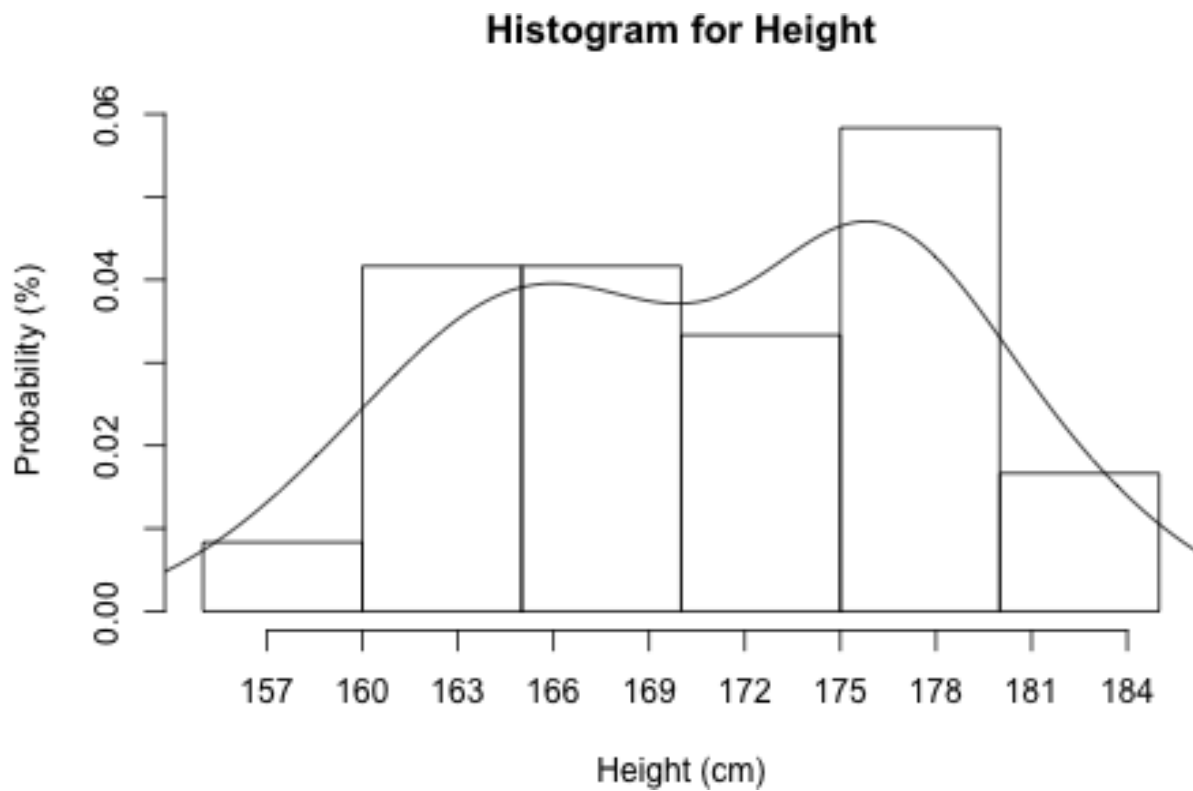


2.2.3.1 with density line

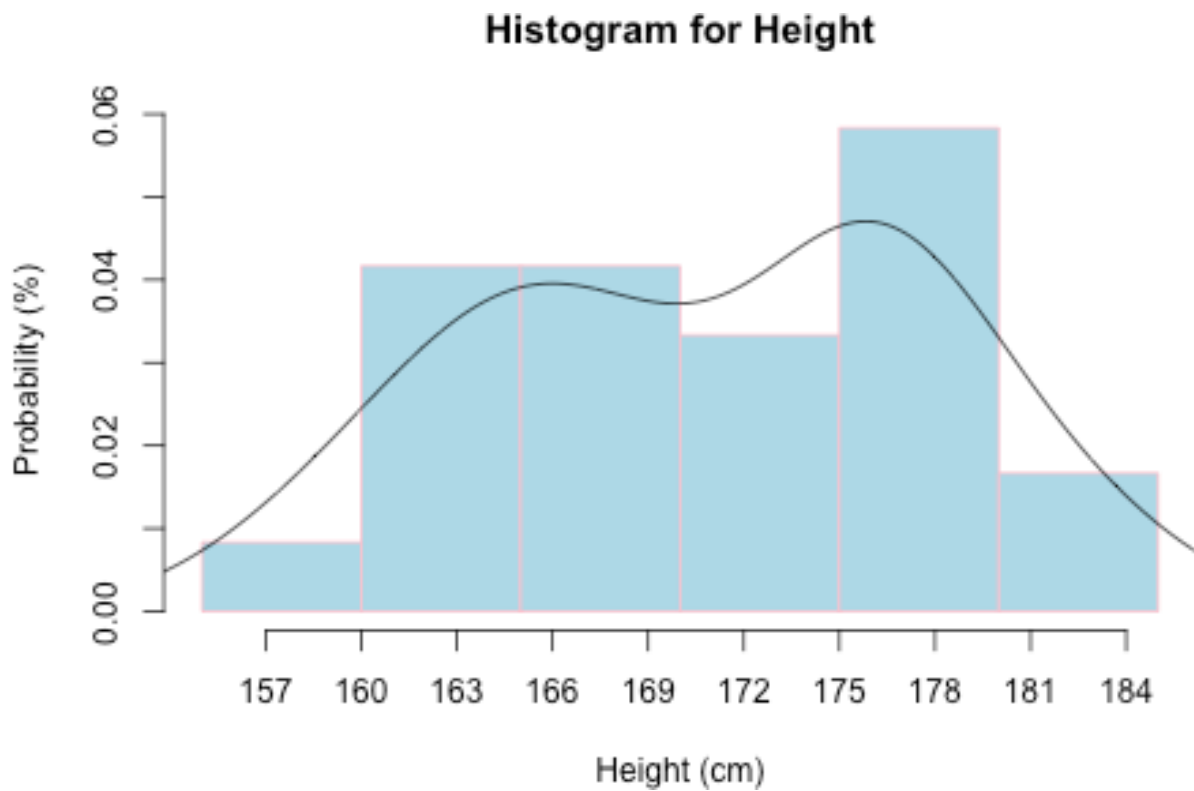
```
hist (d.demog$HT, probability=TRUE, breaks=10)
lines(density(d.demog$HT))
```



```
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))
```

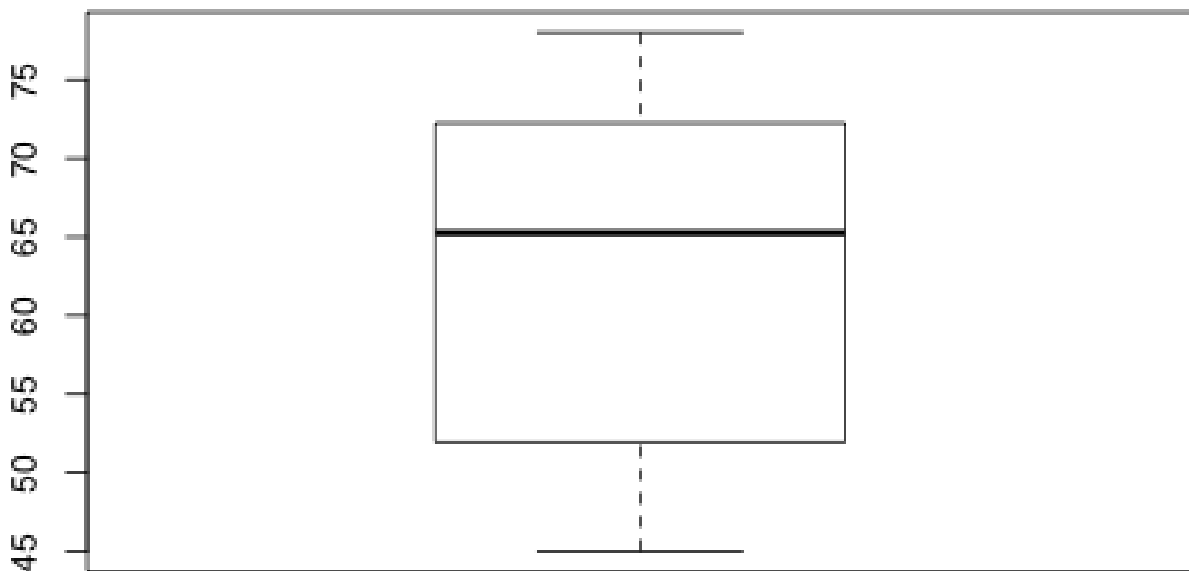


```
hist (d.demog$HT, probability=TRUE, breaks=9, xaxt="n"  
      , main="Histogram for Height", xlab="Height (cm)", ylab="Probability (%)"  
      , col = "lightblue", border = "pink")  
axis(1, at=seq(min(d.demog$HT), max(d.demog$HT), 3))  
lines(density(d.demog$HT))
```



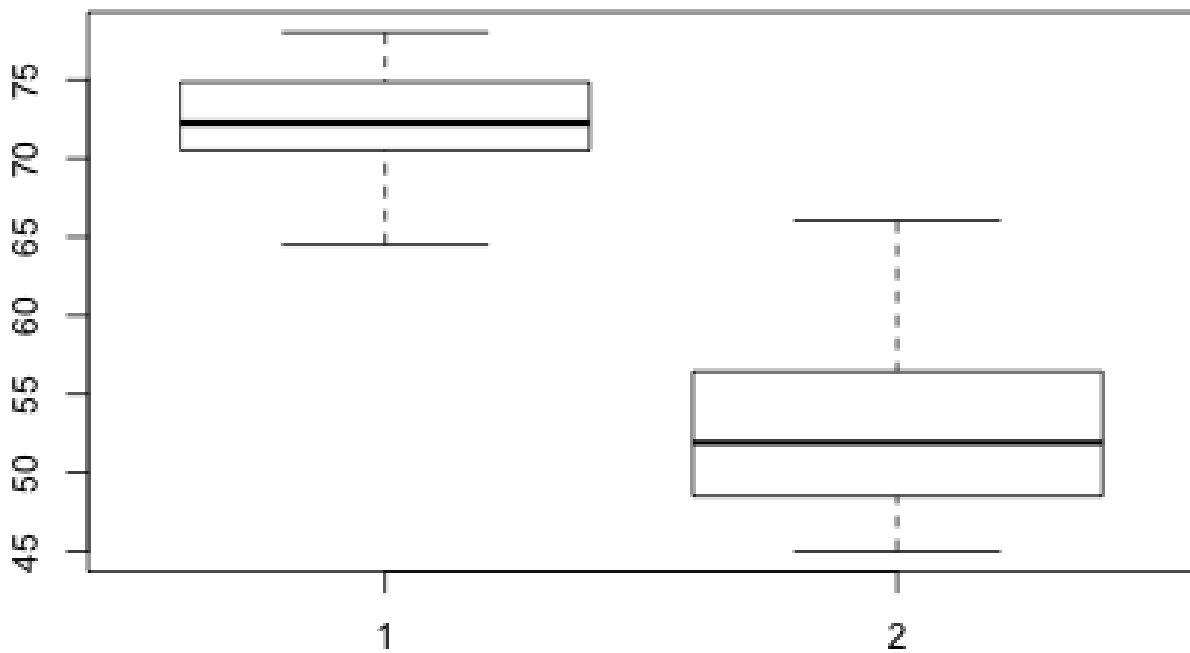
2.2.4 Box-Whisker Plot

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



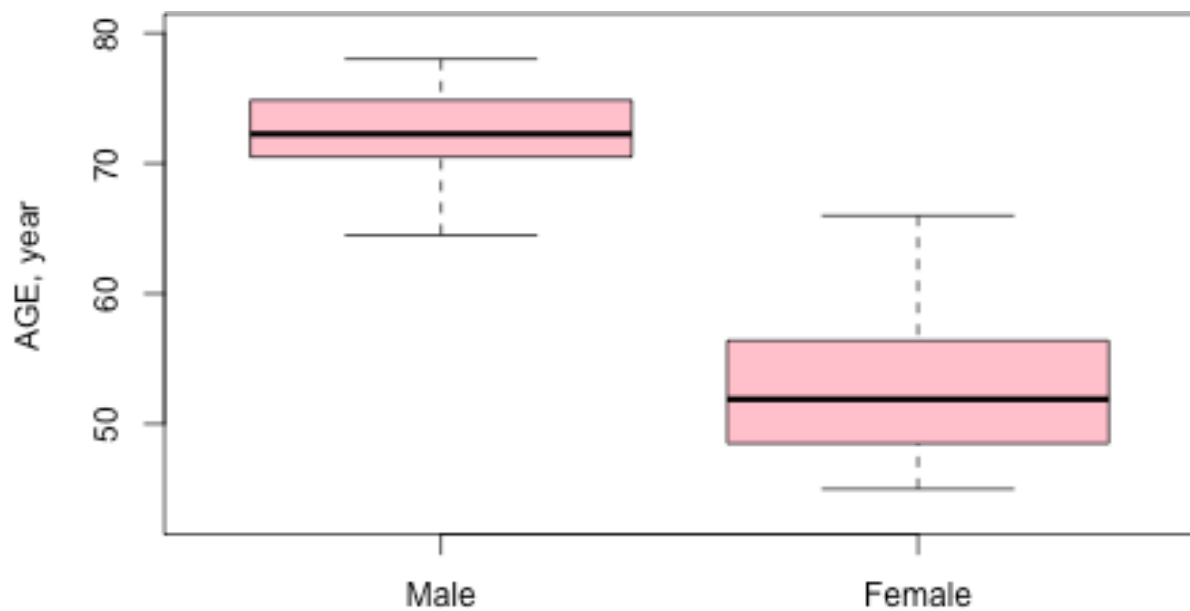
```
boxplot(d.demog$WT ~ d.demog$SEX)
```

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

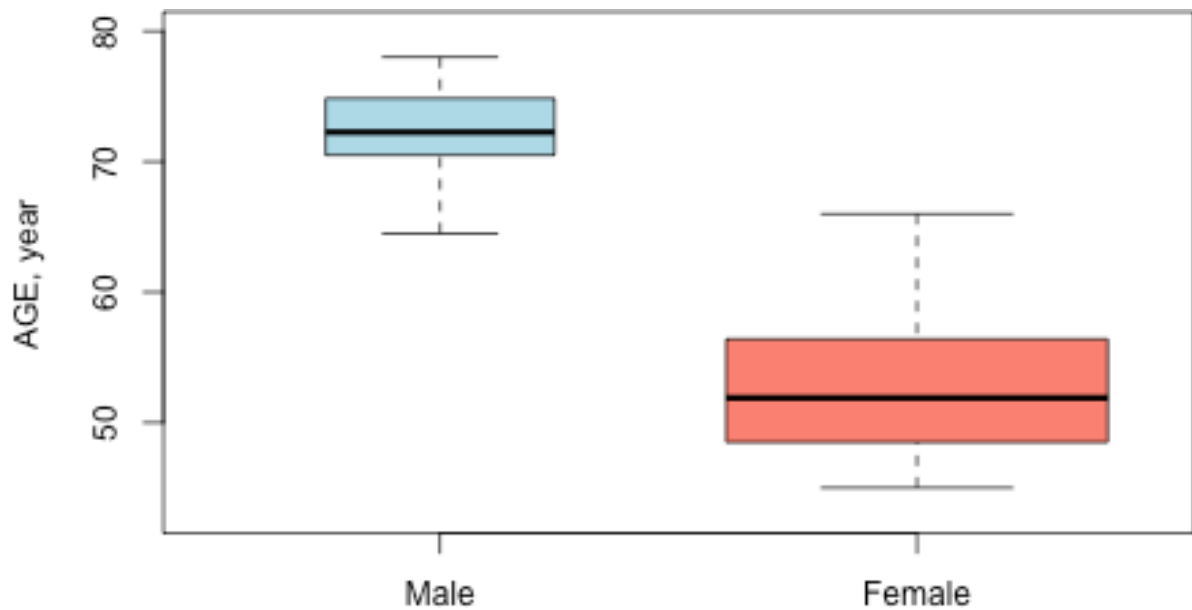


```
boxplot(WT ~ SEX, data=d.demog)
```

```
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)
, col="pink")
```

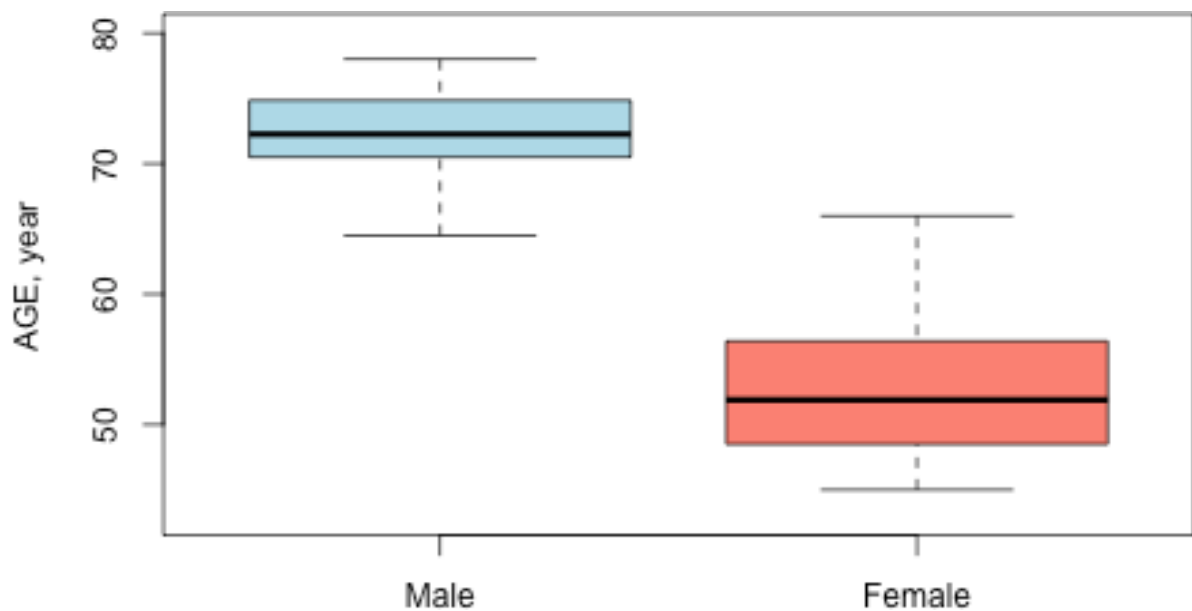


```
boxplot(d.demog$WT ~ d.demog$SEX
, names=c("Male","Female"), ylab="AGE, year", ylim=c(min(d.demog$WT)-2, max(d.demog$WT)
, col=c("lightblue", "salmon"), width=c(0.6, 1))
```



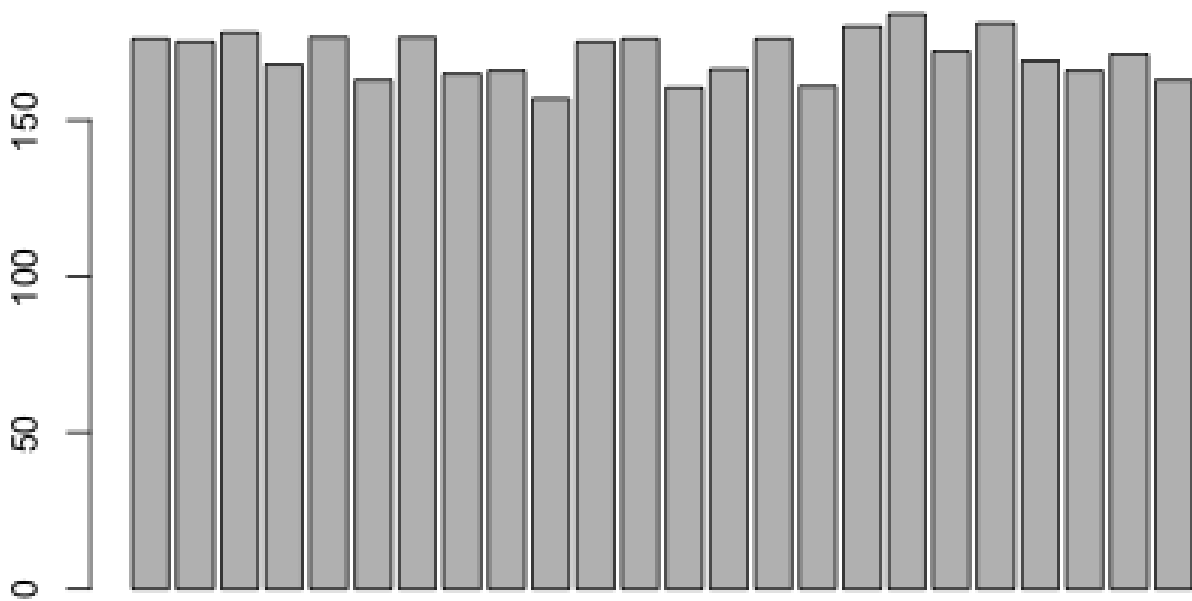
-varwidth: if varwidth is TRUE, the boxes are drawn with widths proportional to the square-roots of the number of observations in the groups.

```
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))
        , col=c("lightblue", "salmon")
        , varwidth=TRUE)
```



2.2.5 Bar Plot

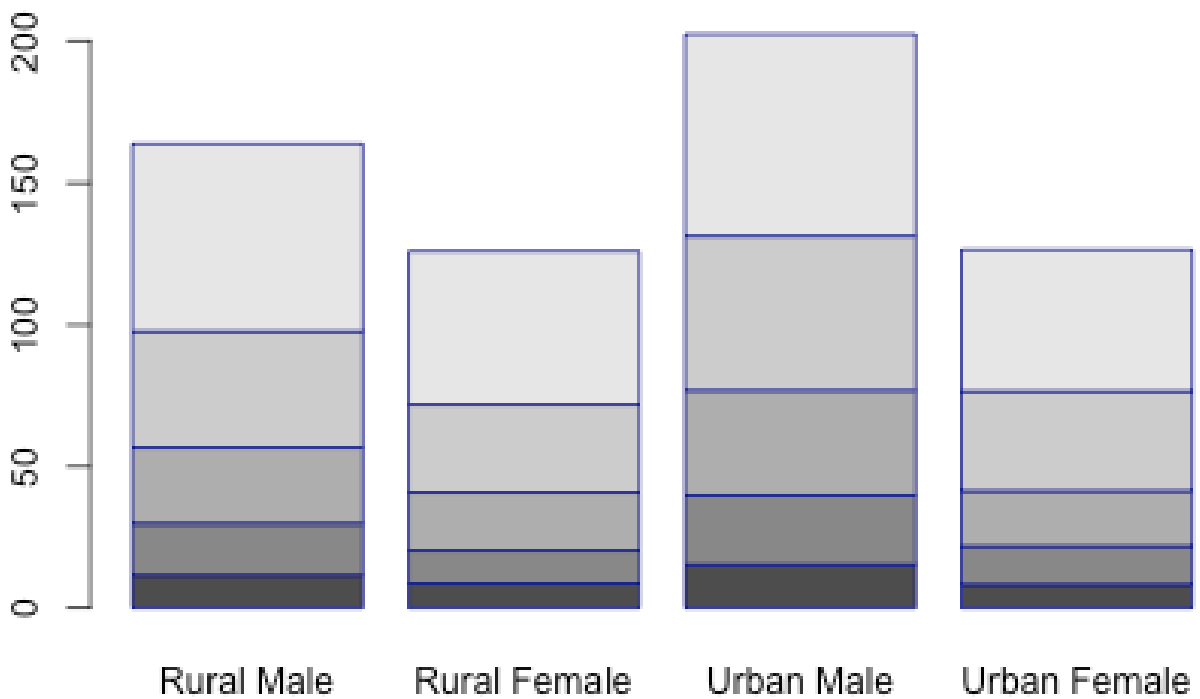
```
barplot(d.demog$HT)
```



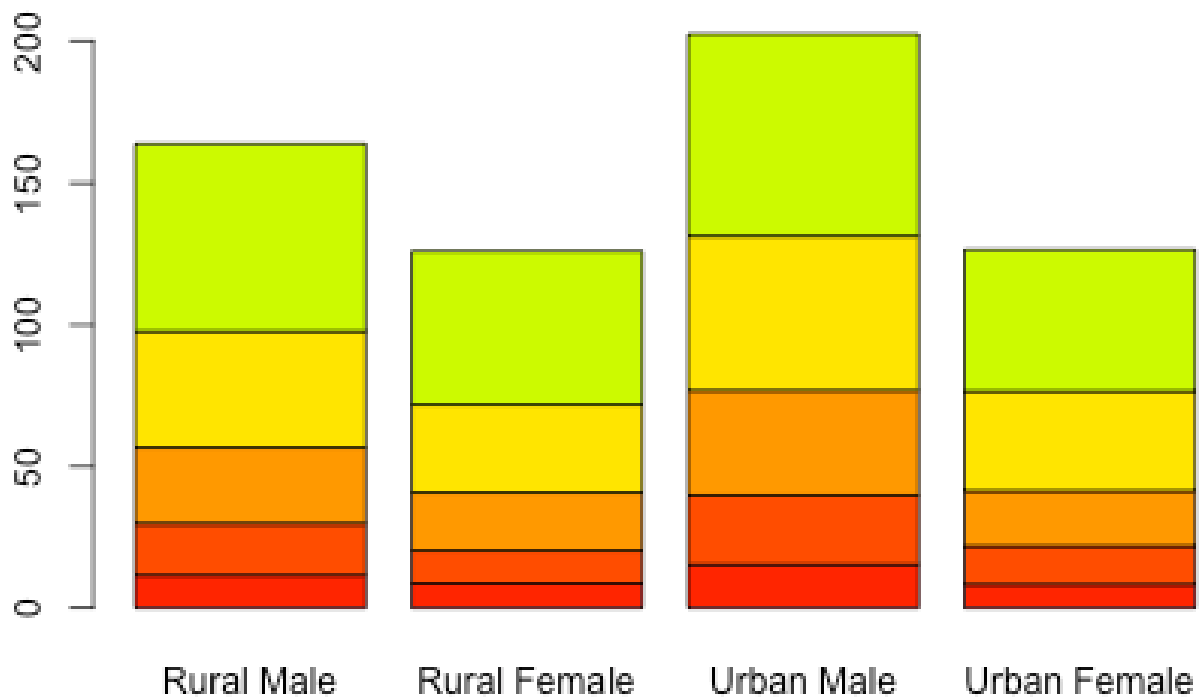
VADeaths

##	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

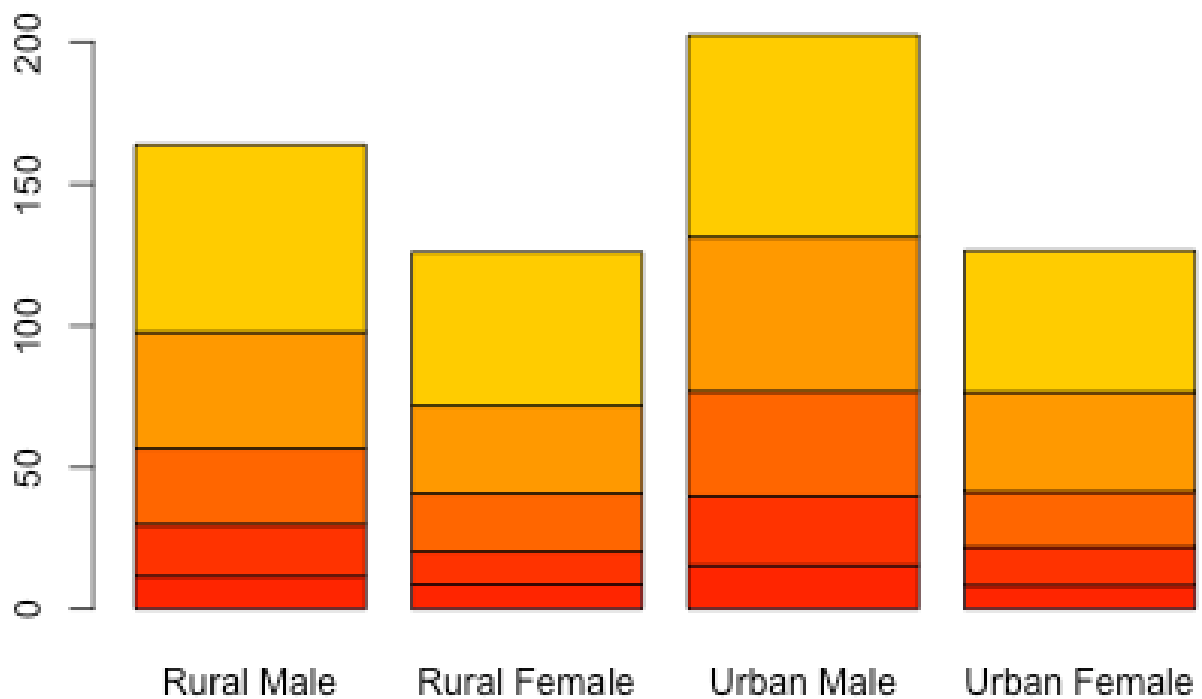
```
barplot(VADeaths, border = "dark blue")
```



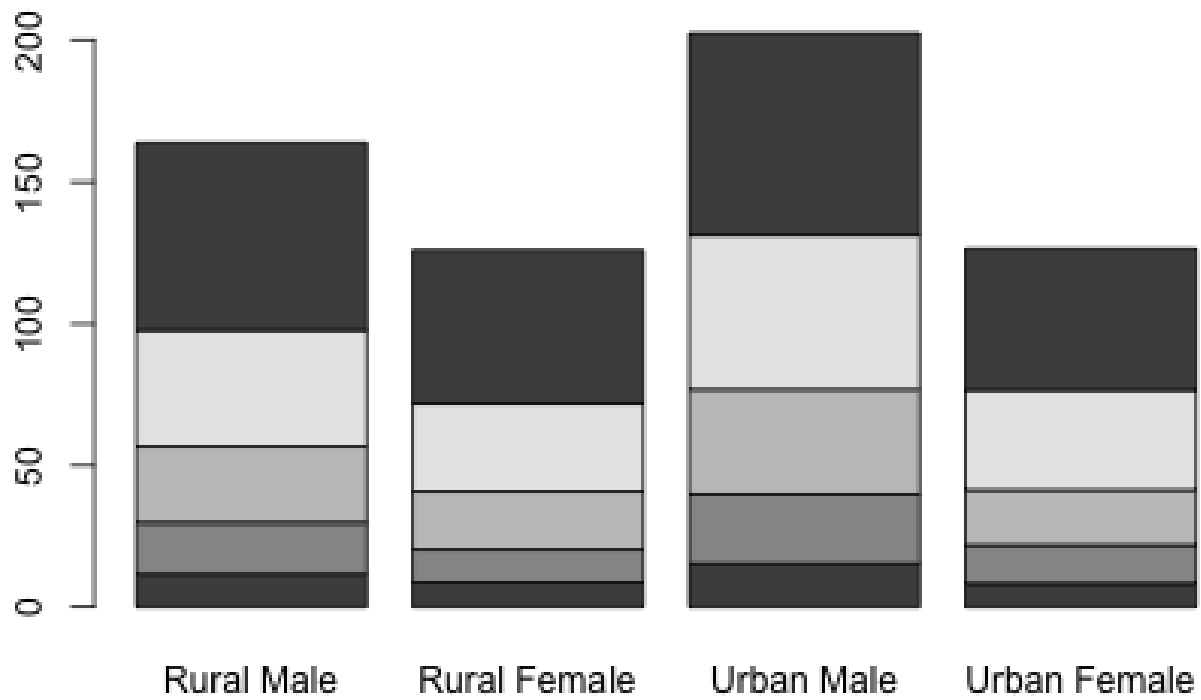
```
barplot(VADeaths, col = rainbow(20))
```



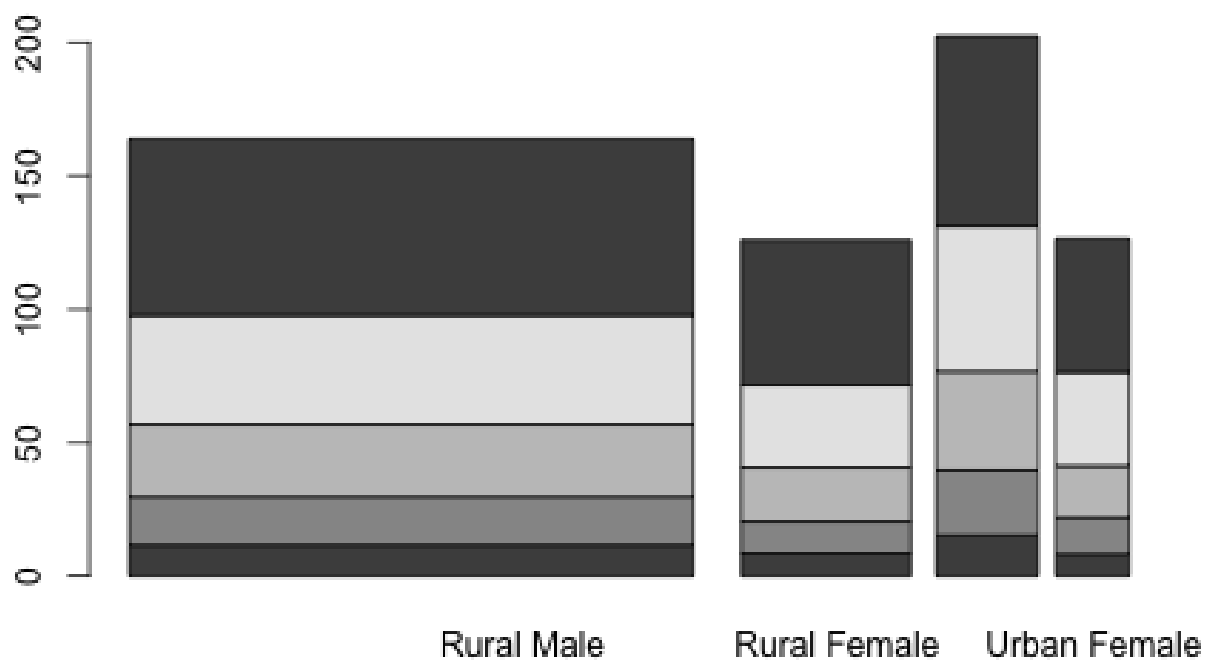
```
barplot(VADeaths, col = heat.colors(8))
```



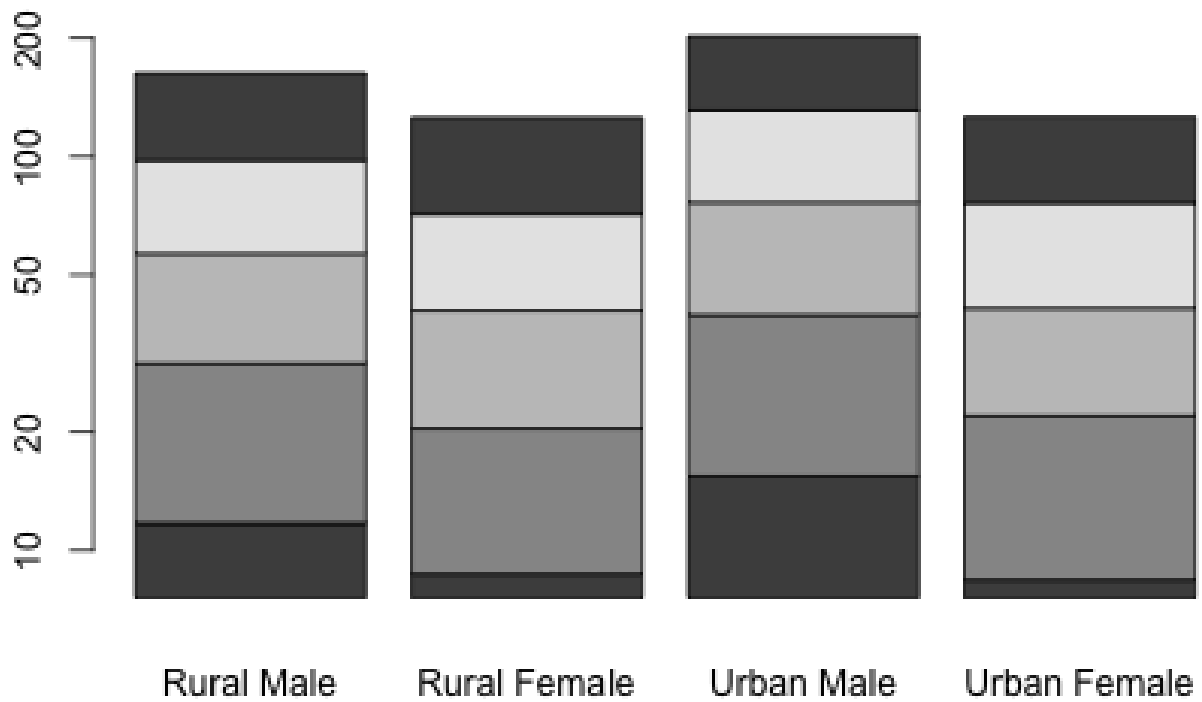
```
barplot(VADeaths, col = gray.colors(4))
```



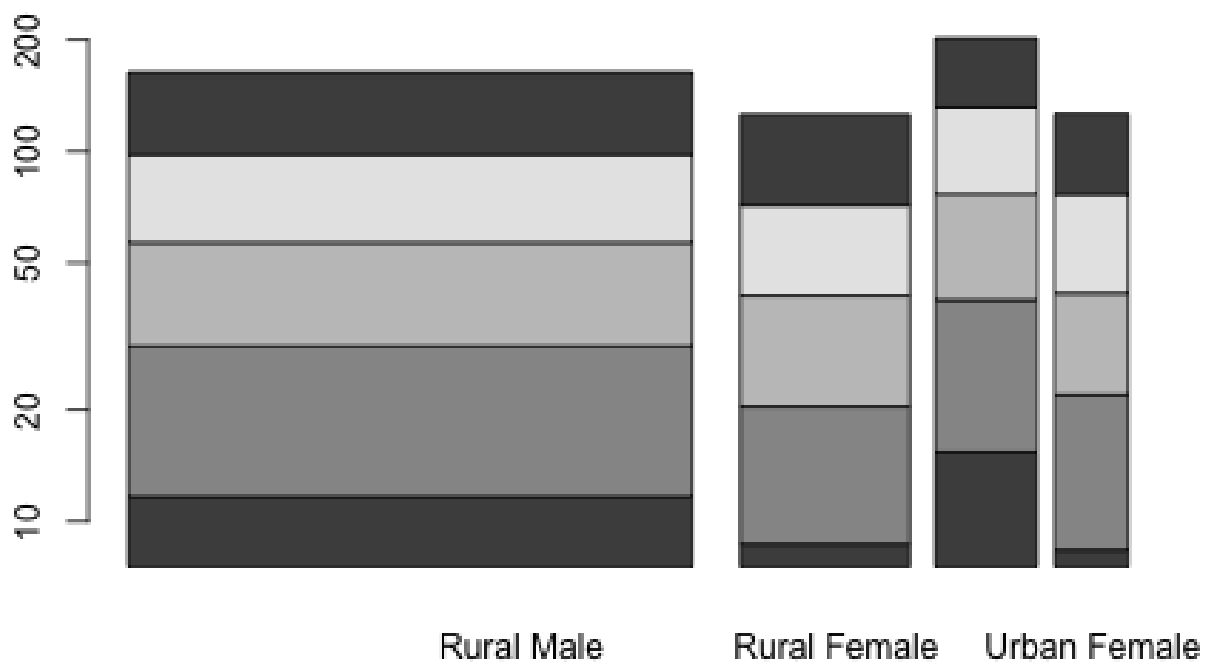
```
barplot(VADeaths, col = gray.colors(4), log="x")
```



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

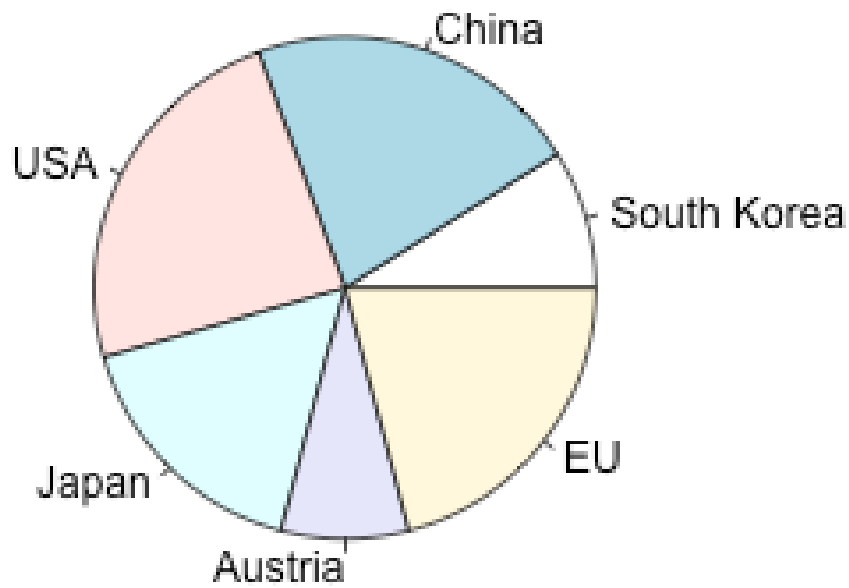



```
barplot(VADeaths, col = gray.colors(4), log="xy")
```



2.2.6 pie chart

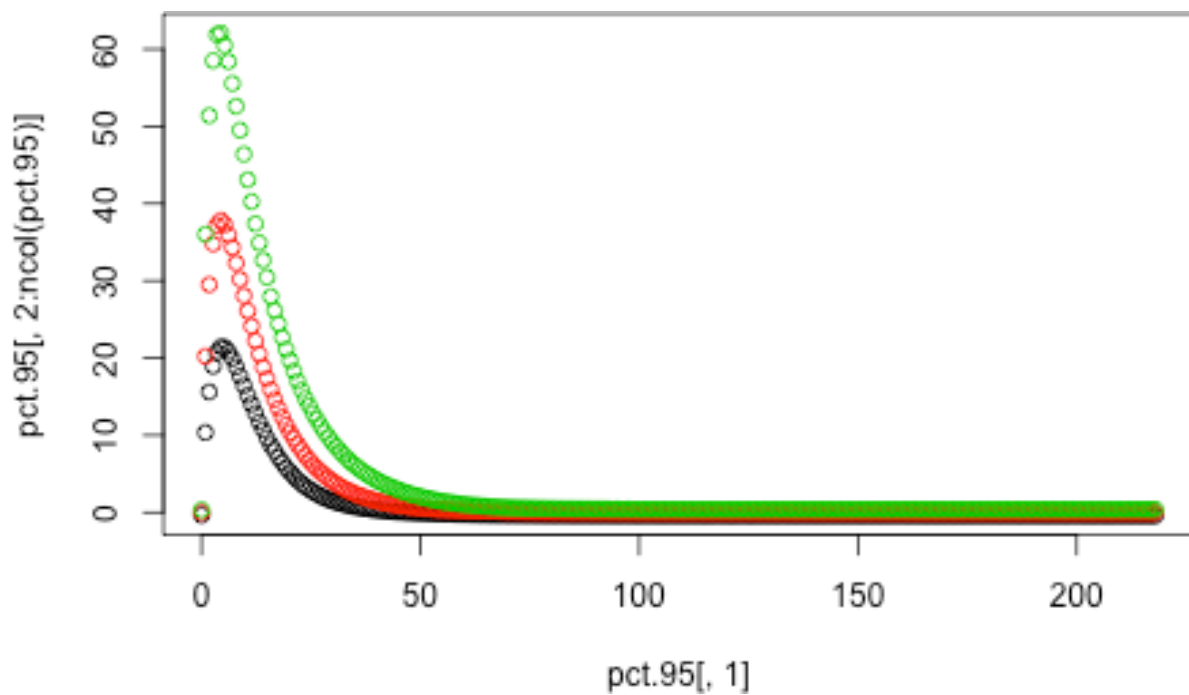
```
drug.X.market <- c(0.12, 0.29, 0.32, 0.22, 0.11, 0.28)
names(drug.X.market) <- c("South Korea", "China", "USA", "Japan", "Austria", "EU")
pie(drug.X.market)
```



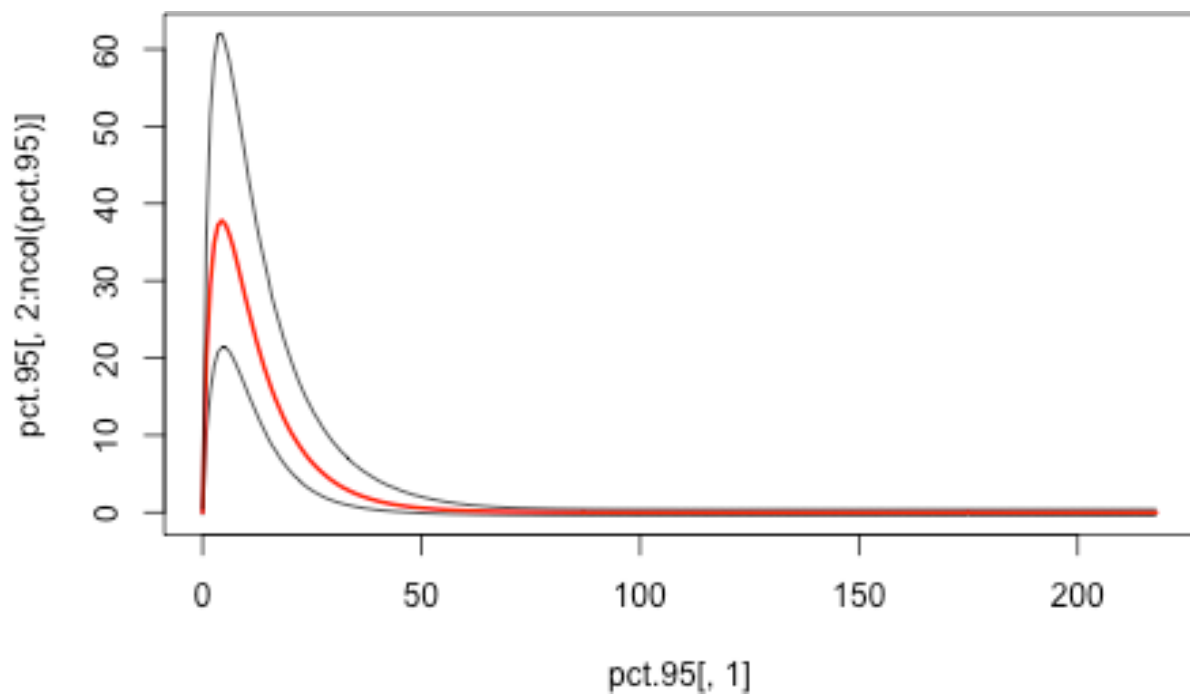
2.2.7 matplot 함수

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)
```

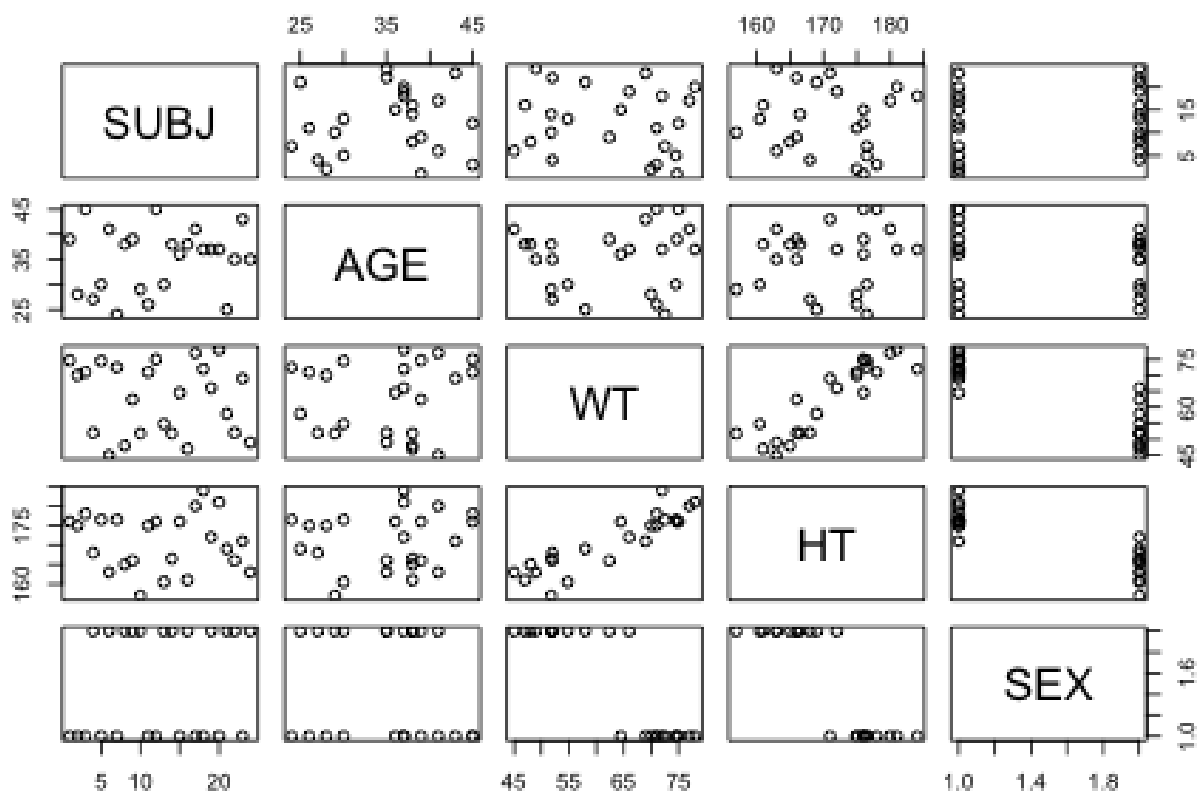


```
matplot(pct.95[,1], pct.95[,2:ncol(pct.95)], pch=1, col=c(1,2,1), type="l", lty=1, lwd=c(1,2,1))
```



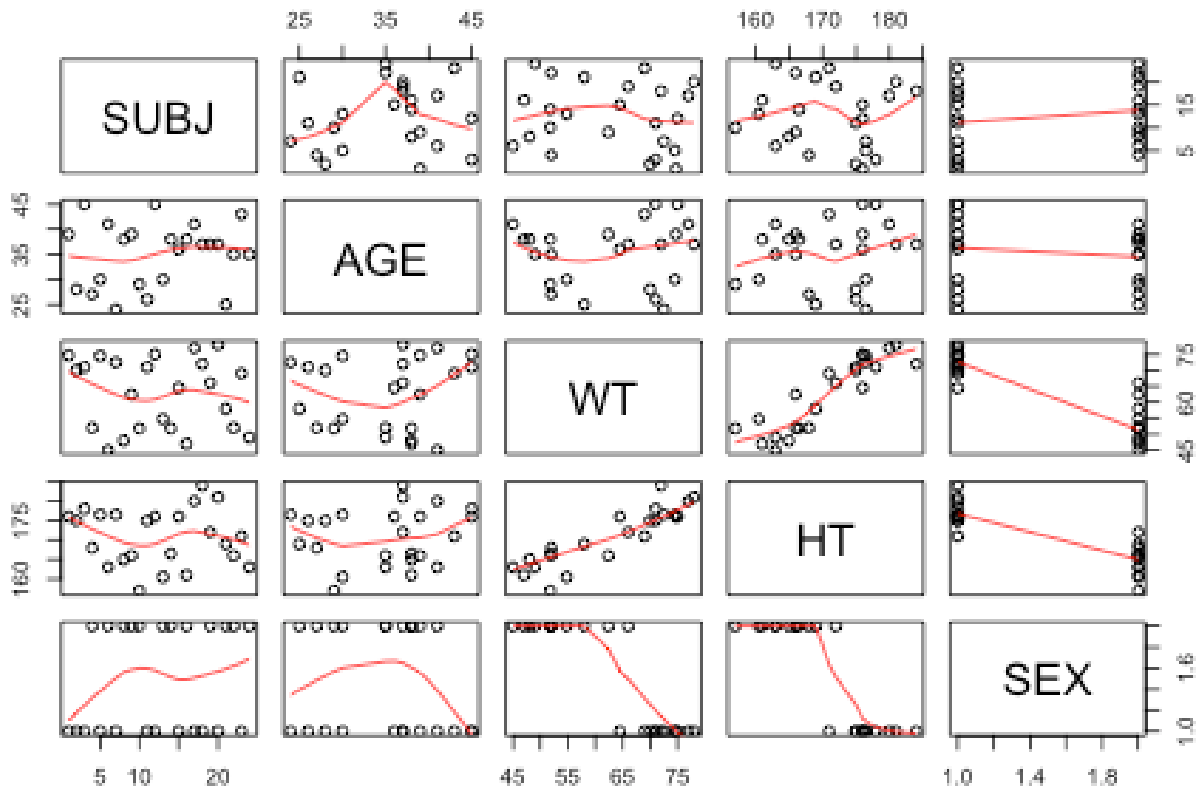
2.2.8 Scatter plot matrices (pairs plots)

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



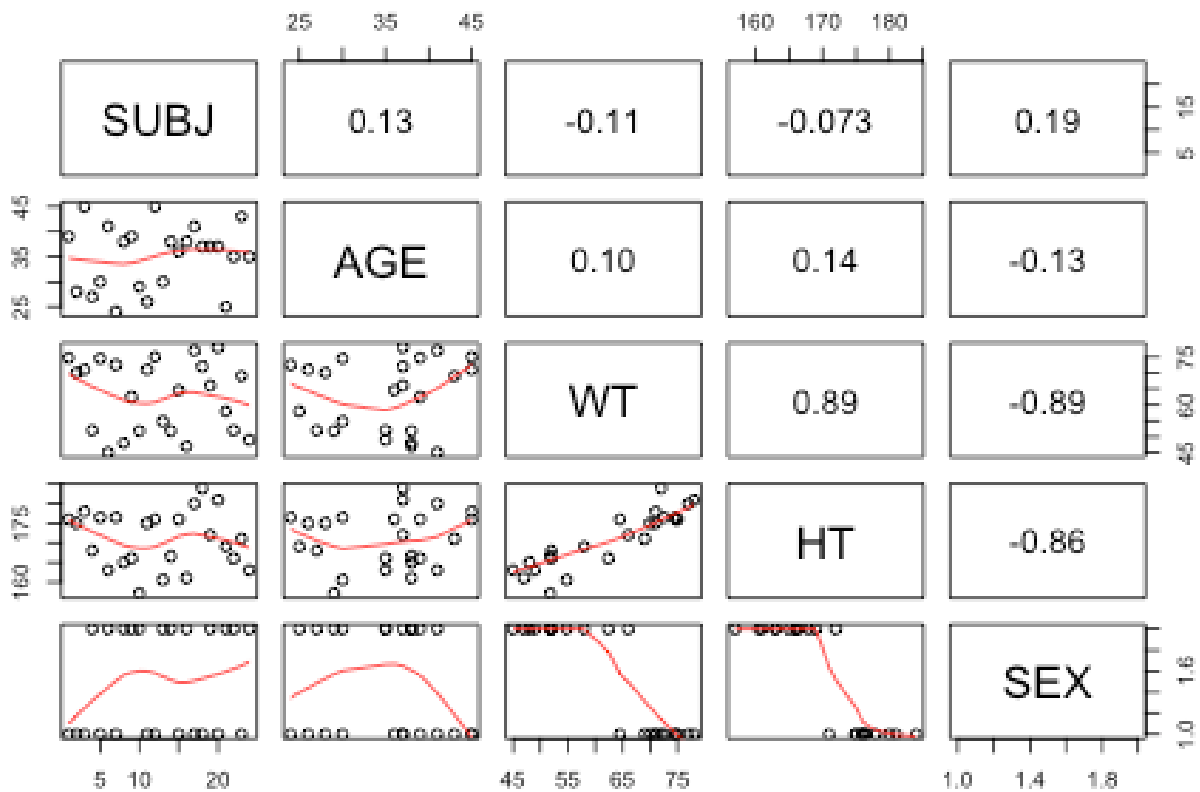
2.2.8.1 add a loess smoother, type

```
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)
```

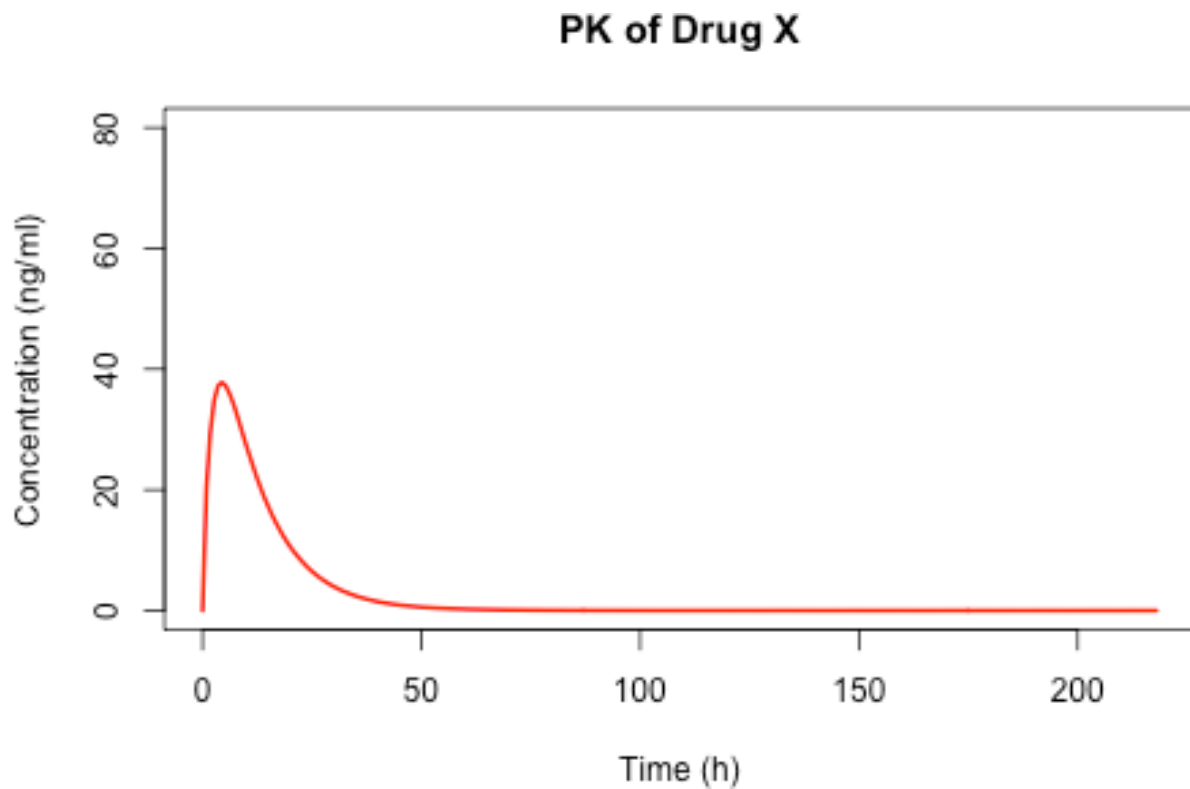


2.3 하위수준 그림 함수

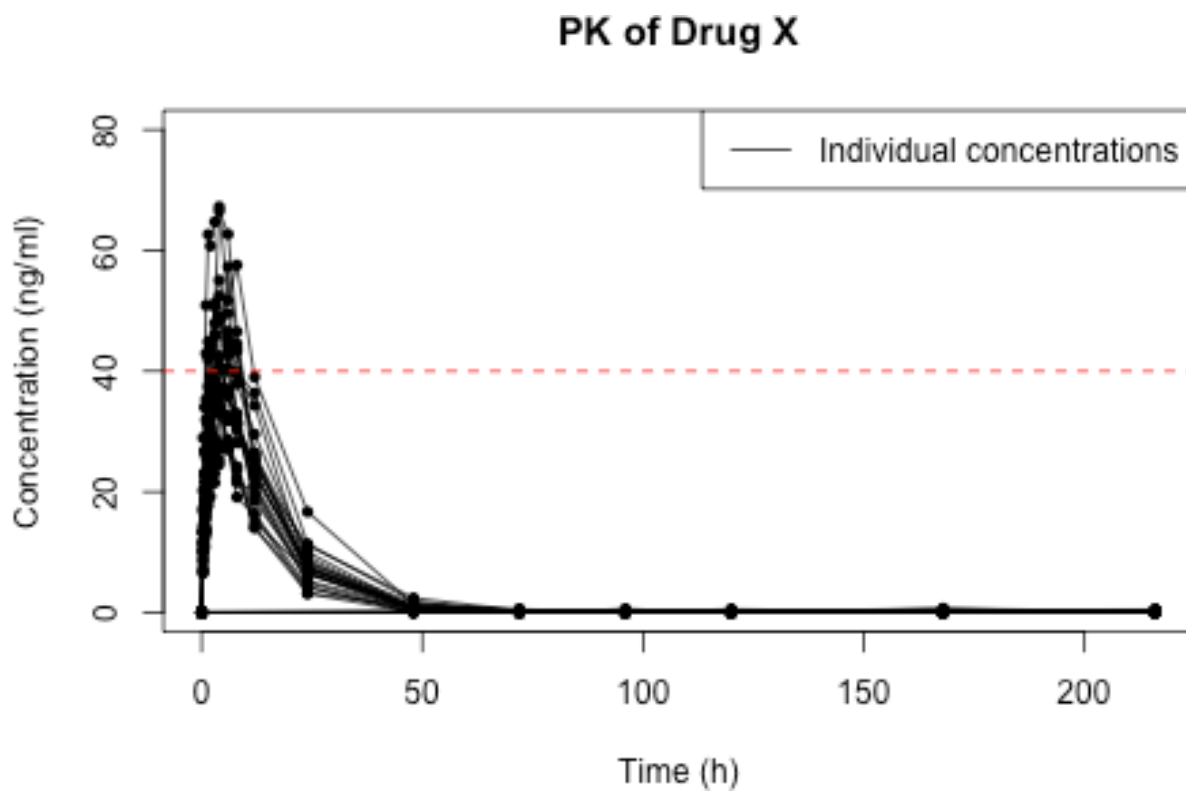
- points : 점 추가
- lines : 선 추가
- abline : 기준선 추가
- mtext : 텍스트 추가
- legend : 설명 (legend) 추가
- polygon : polygon 추가

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)
```

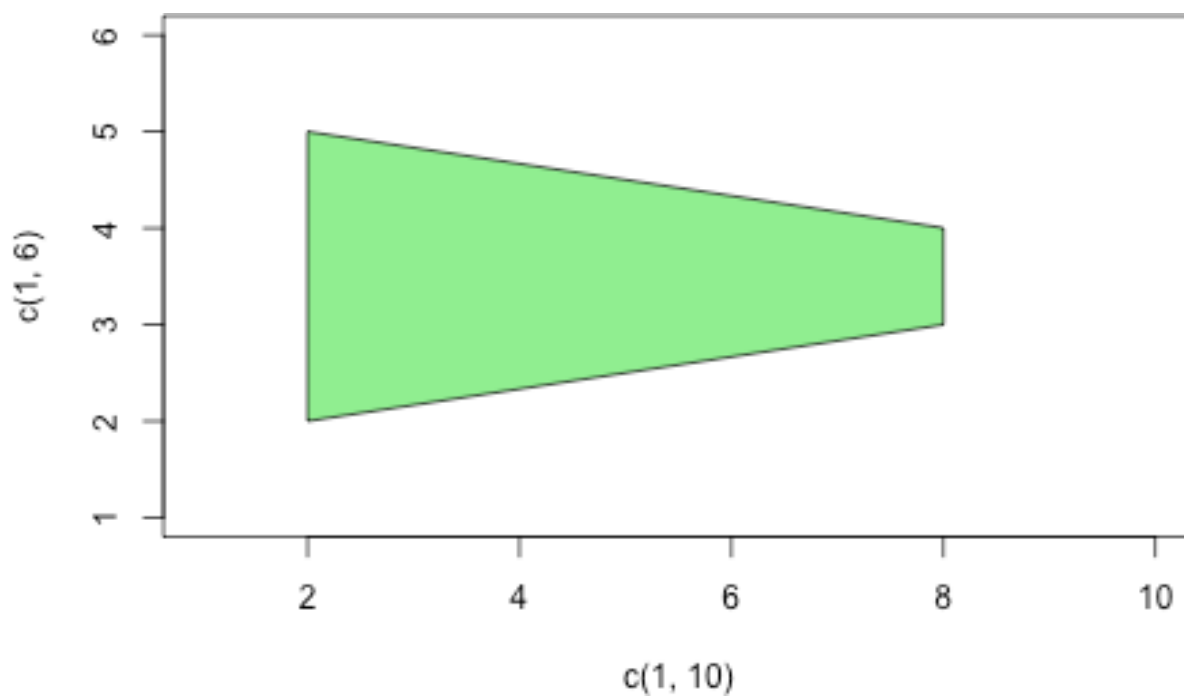


```
plot(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], main="PK of Drug X"
     , type="n", xlab="Time (h)", ylab="Concentration (ng/ml)"
     , ylim=range(0,80))
points(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], pch = 16, cex=0.8)
lines(dta$TIME[dta$MDV==0], dta$DV[dta$MDV==0], col="black", lwd=1)
abline(40,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

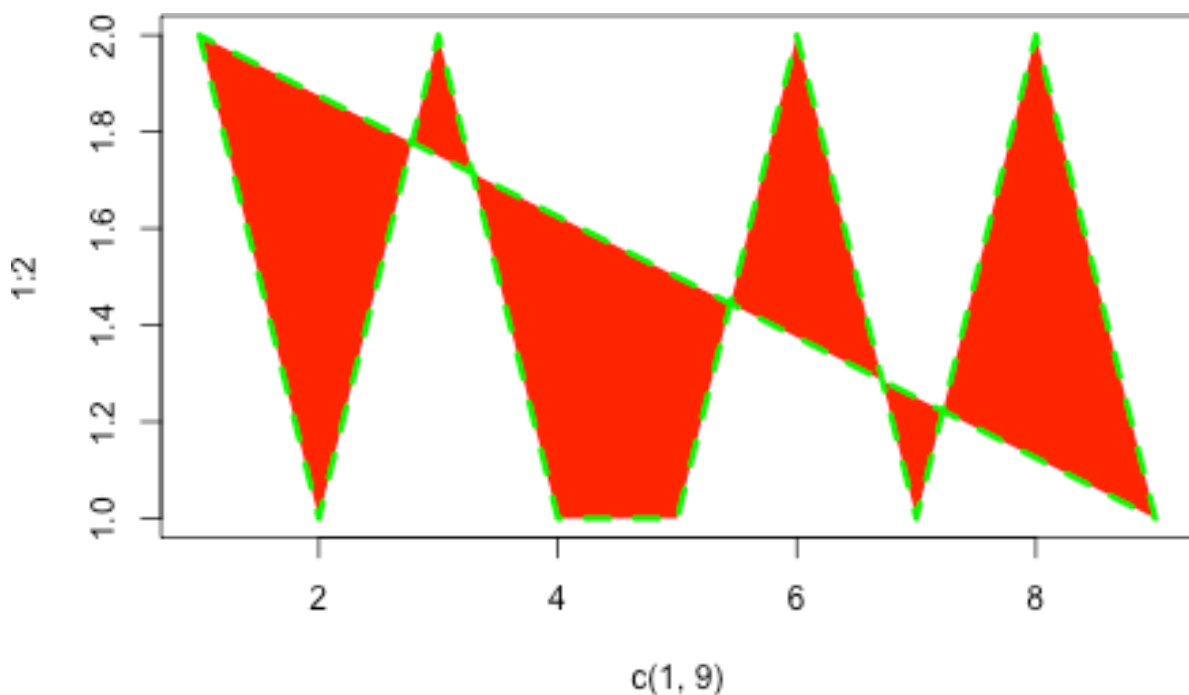


2.3.2 polygon 함수

```
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()
## quartz_off_screen
##                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()
## quartz_off_screen
##                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
```

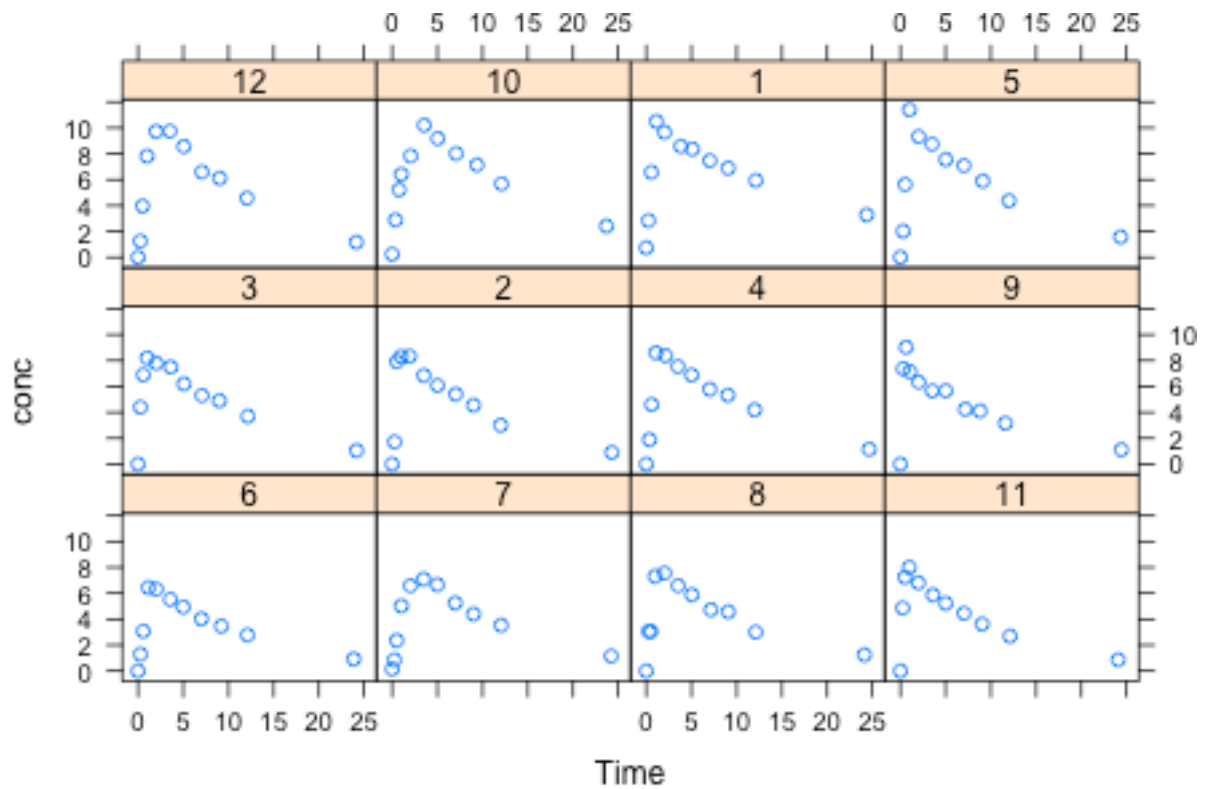
R console에서 ?Theoph를 타이핑 치면 좀 더 자세한 정보를 얻을 수 있습니다.

3.3 lattice

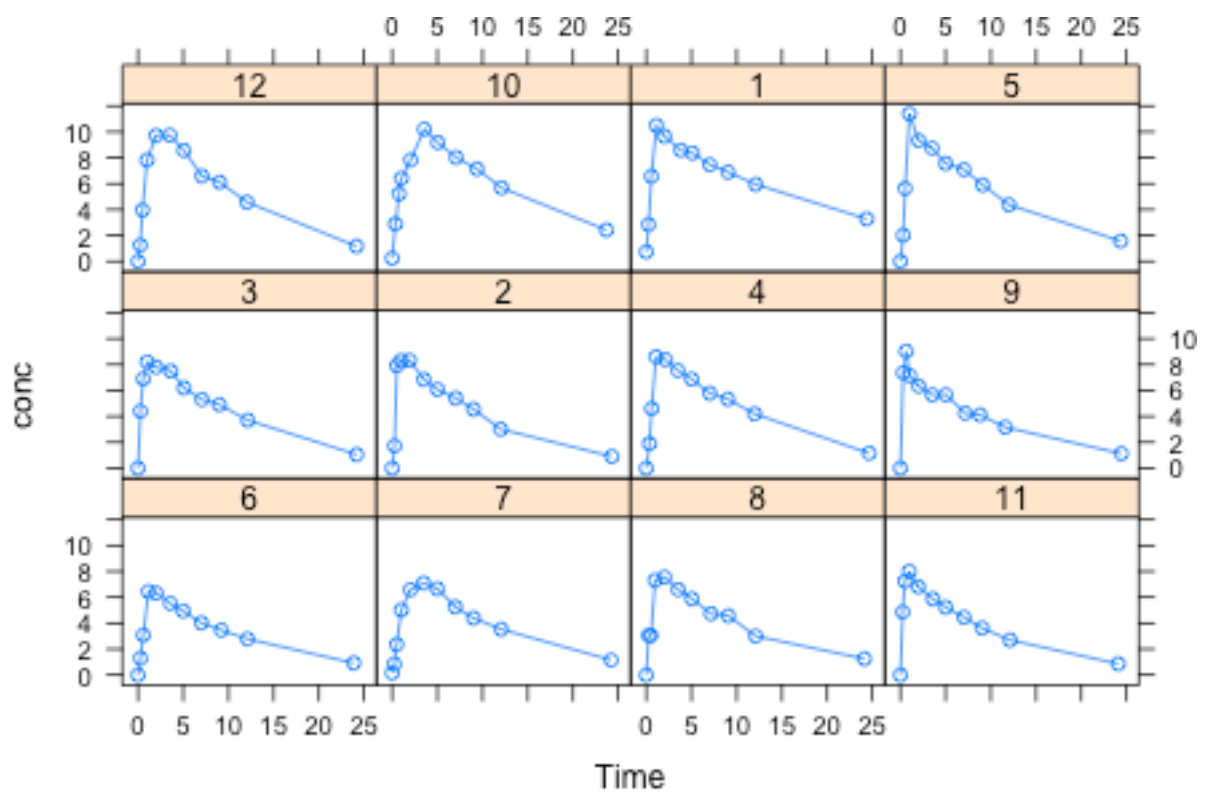
lattice 패키지를 불러온 뒤 그림을 그려보겠습니다. (Sarkar, 2017)

```
library(lattice) # trellis
```

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

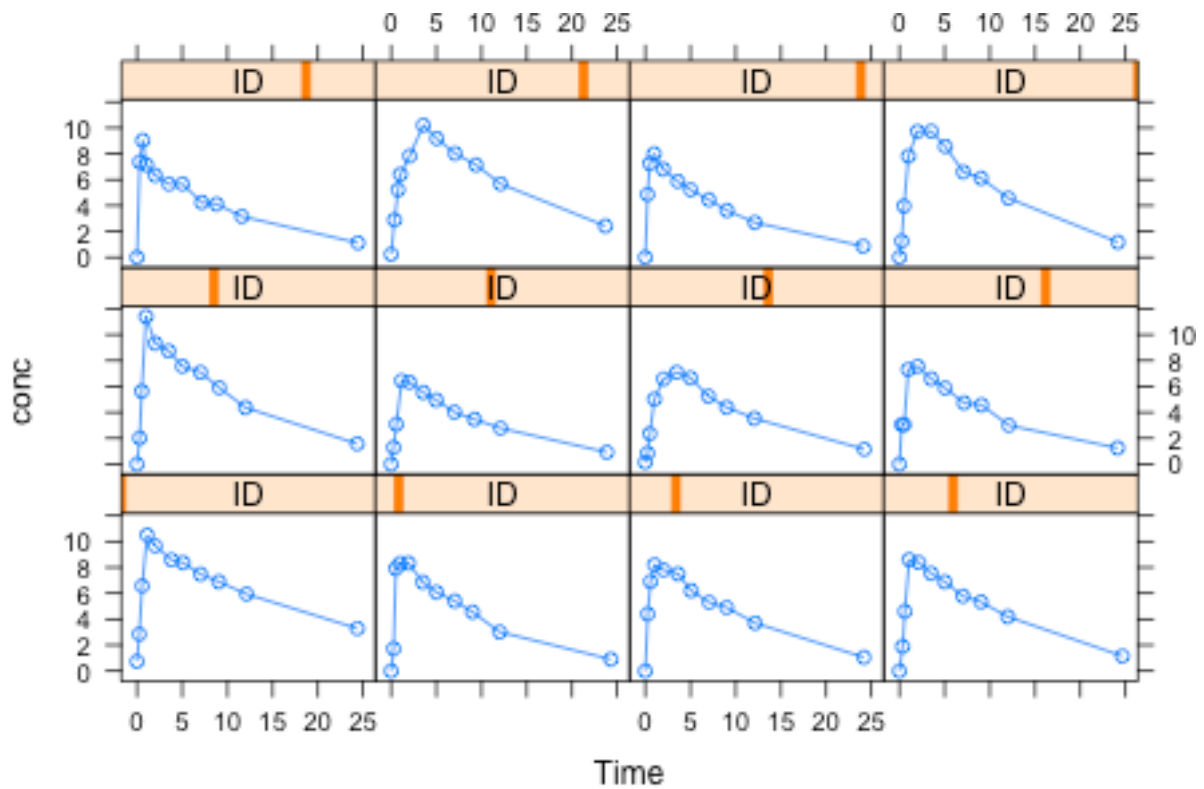


```
xyplot(conc ~ Time | Subject, data=Theoph, type="b")
```

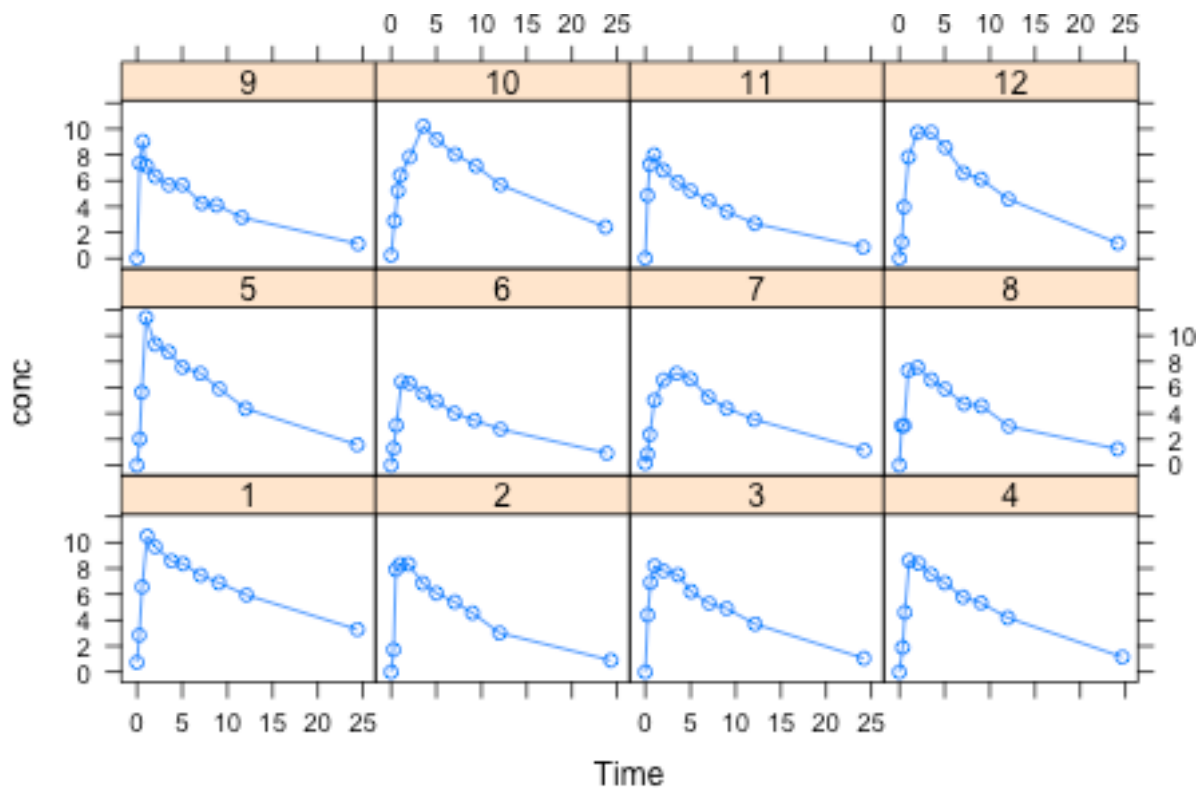


```
Theoph[, "ID"] = as.numeric(as.character(Theoph[, "Subject"]))
```

```
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
# ?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"), lty=3)
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)
}

for (i in 1:10) print(2*i)

while ( ) {
  # Statements
}

# # if ~ else ~
# if ( ) {
# # Statements 1
# } else {
# # Statements 2

```

```

# }
#
# if ( ) # Statement1
# else # Statement2
#
# if ( ) {
# # Statements 1
# } else if ( ) {
# # Statements 2
# } else if ( ) {
# # Statements 3
# } else {
# # Statements 4
# }

#
#

# Chapter 10 Writing your own functions

Square = function(x=0)
{
  return(x*x)
}

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 = 2*(1 - pt(abs(tst), df=DF))

  Res = list(tst, DF, p.val, yb1, yb2)
  names(Res) = c("t", "df", "p-value", "mean of x", "mean of y")

  return (Res)
}

res = T.test(x, y)
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)

# More basic keywords and functions
1 %in% c(1,2,3,4)
5 %in% c(1,2,3,4)
is.finite(Inf)
prod(1:3)
cummax(1:10)

```

TABLE 4.1: The basics - The first functions to learn

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

```

cummax(10:1)
# ?xor
x = 11:20
x
which(x==3)
which(x==13)

length(x)
y = "my string"
length(y)
nchar(y)
strsplit(y, " ")
strsplit(y, " ")[[1]]
substr(y, 4, 5)

sample(1:10)
sample(1:10, 20)
sample(1:10, 20, replace=TRUE)
sample(rep(1:10,2))

## Error: <text>:99:12: 예기치 않은 ')'입니다
## 98:
## 99: while (    )
##                ^

```

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

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	

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 figures
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?

4.2 The basics

4.3 Common data structures

4.4 Statistics

4.5 Working with R

4.6 I/O

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 Summ
as.matrix	H	Y	
data.matrix	L	N	Convert a Data Frame

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 w
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
(is/as).(character/numeric/logical/...)	H	Y	

TABLE 4.9: The basics - Lists - data.frames

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

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	Regression Deletion Diagnostics
lm	H	Y	
glm	H	Y	Regression Deletion Diagnostics
hat	L	Y	
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	
eigen	H	Y	
qr	L	Y	
svd	L	Y	
%%*	H	Y	Outer Product of Arrays
%o%	L	Y	
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	Search for Key Words or Phrases in Documentation
?	H	Y	
help.search	L	N	
apropos	L	Y	
RSiteSearch	L	N	
citation	L	Y	View, List or Get R Source of Package Vignettes
demo	L	Y	
example	L	Y	
vignette	L	Y	

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	Serialization Interface for Single Objects
read.delim	L	N	
write.delim	L	N	
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	Expand File Paths
path.expand	L	Y	
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	

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)
library(moonBook)
```

```
[1] "age"           "sex"           "cardiogenicShock" "entry"         "Dx"
[6] "EF"           "height"        "weight"          "BMI"           "obesity"
[11] "TC"           "LDLC"          "HDL"             "TG"            "DM"
[16] "HBP"          "smoking"
```

```
> str(acs)
'data.frame': 857 obs. of 17 variables:
 $ age      : int  62 78 76 89 56 73 58 62 59 71 ...
 $ sex      : chr  "Male" "Female" "Female" "Female" ...
 $ cardiogenicShock: chr  "No" "No" "Yes" "No" ...
```

```

$ entry      : chr  "Femoral" "Femoral" "Femoral" "Femoral" ...
$ Dx         : chr  "STEMI" "STEMI" "STEMI" "STEMI" ...
$ EF         : num   18 18.4 20 21.8 21.8 22 24.7 26.6 28.5 31.1 ...
$ height     : num   168 148 NA 165 162 153 167 160 152 168 ...
$ weight     : num   72 48 NA 50 64 59 78 50 67 60 ...
$ BMI        : num   25.5 21.9 NA 18.4 24.4 ...
$ obesity    : chr   "Yes" "No" "No" "No" ...
$ TC         : num   215 NA NA 121 195 184 161 136 239 169 ...
$ LDLC       : int   154 NA NA 73 151 112 91 88 161 88 ...
$ HDLC       : int   35 NA NA 20 36 38 34 33 34 54 ...
$ TG         : int   155 166 NA 89 63 137 196 30 118 141 ...
$ DM         : chr   "Yes" "No" "No" "No" ...
$ HBP        : chr   "No" "Yes" "Yes" "No" ...
$ smoking    : chr   "Smoker" "Never" "Never" "Never" ...

```

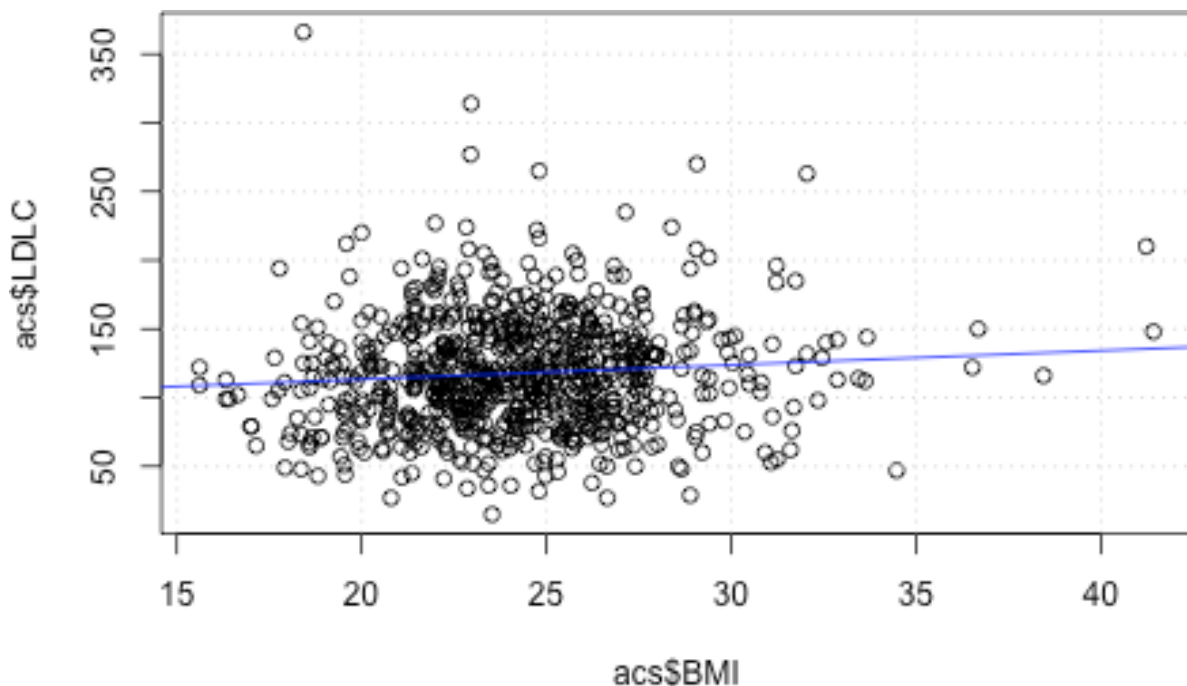
Scatter plot을 `geom_point()` 를 사용해서 그려보도록 하겠습니다.

```

plot(acs$BMI, acs$LDLC)
grid()          # Add gridlines
# Create a linear regression model
model <- lm(formula = LDLC ~ BMI, # 순서에 조심!
            data = acs)

abline(model, col = 'blue')      # Add regression to plot

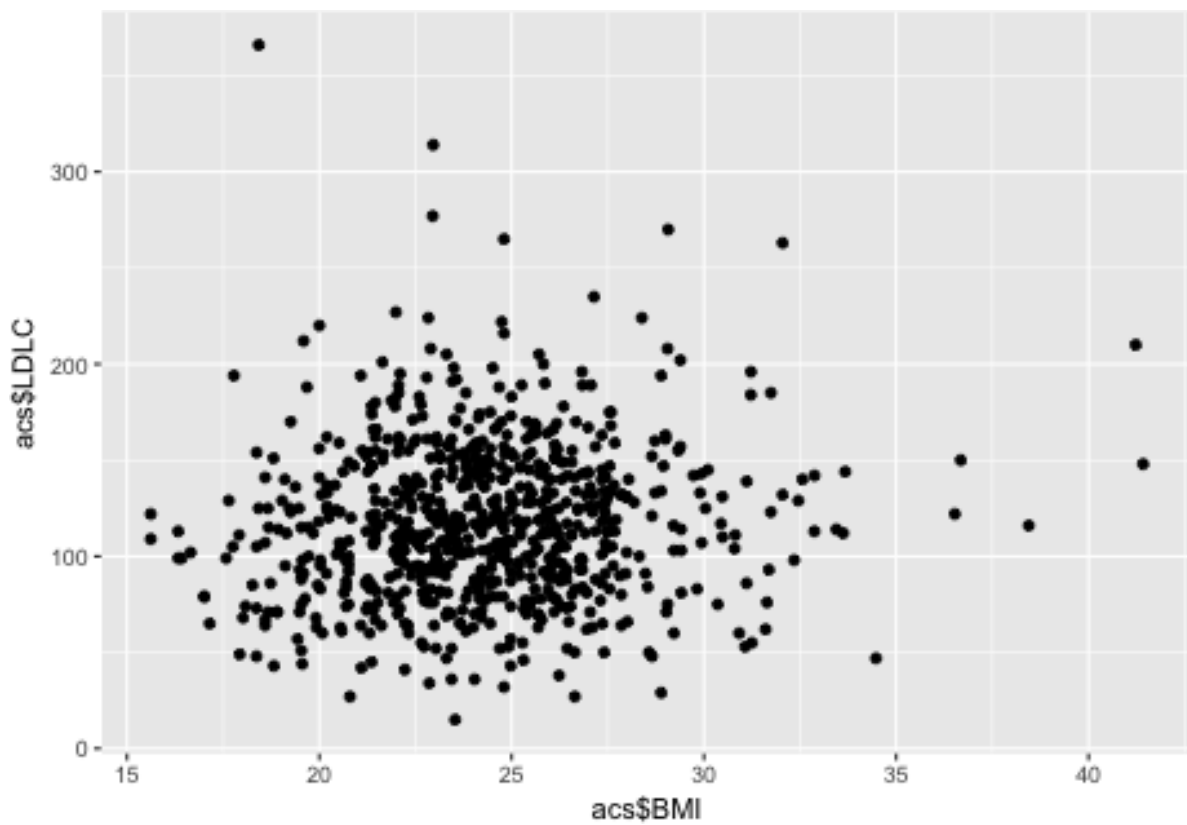
```



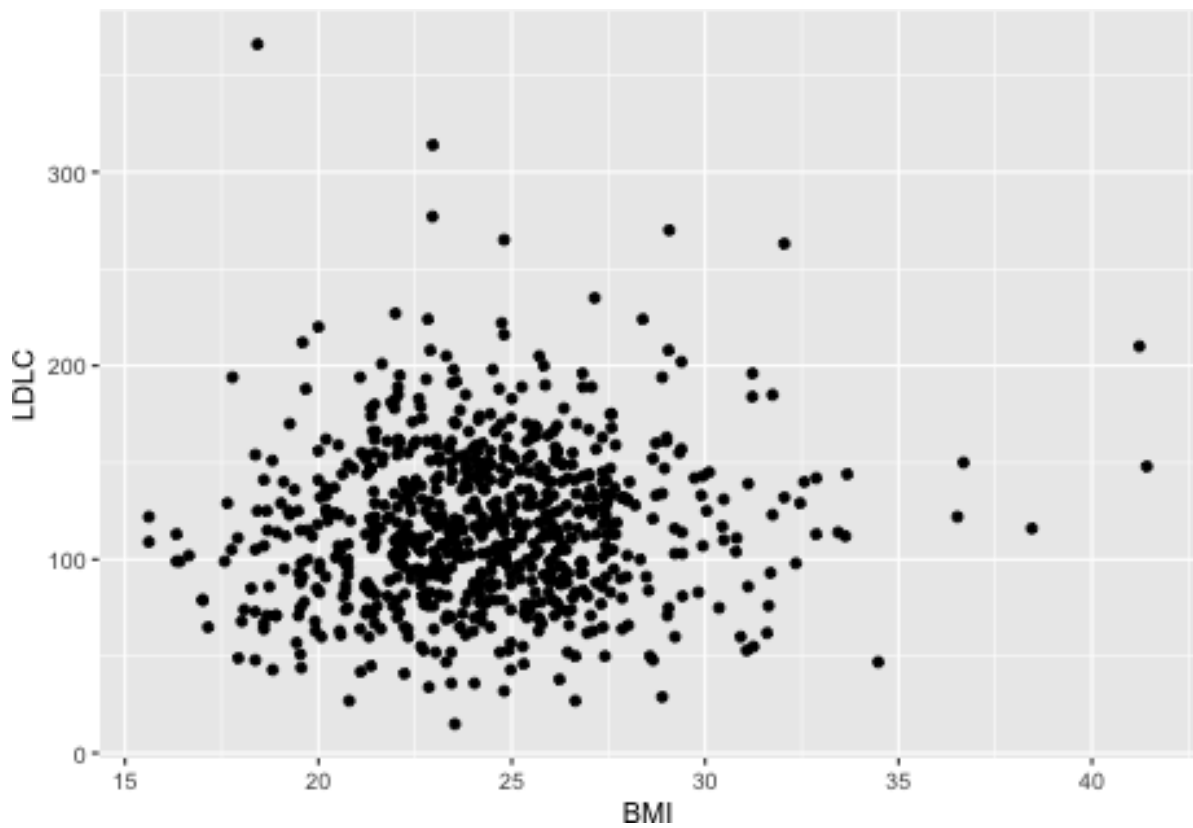
```

qplot(acs$BMI, acs$LDLC)
## Warning: Removed 106 rows containing missing values
## (geom_point).

```



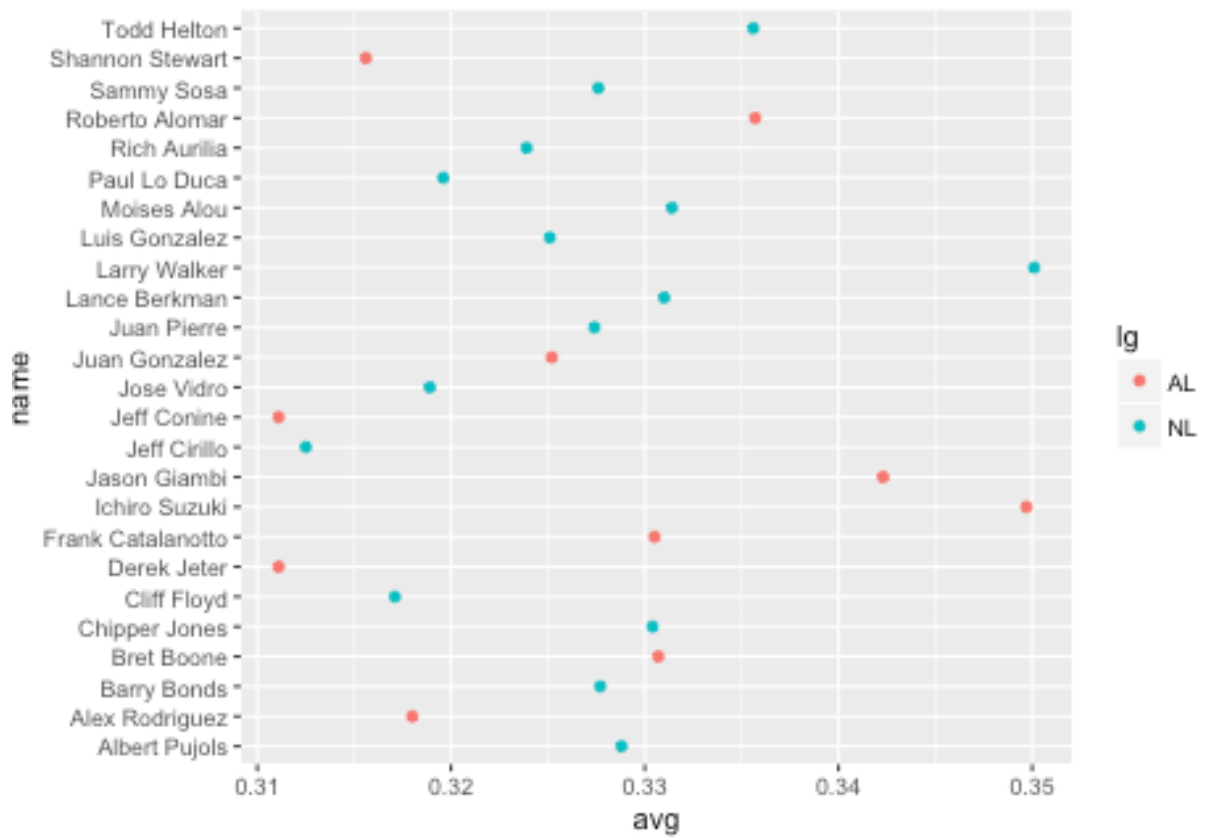
```
ggplot(acs, aes(x = BMI, y = LDLC)) + geom_point()  
## Warning: Removed 106 rows containing missing values  
## (geom_point).
```



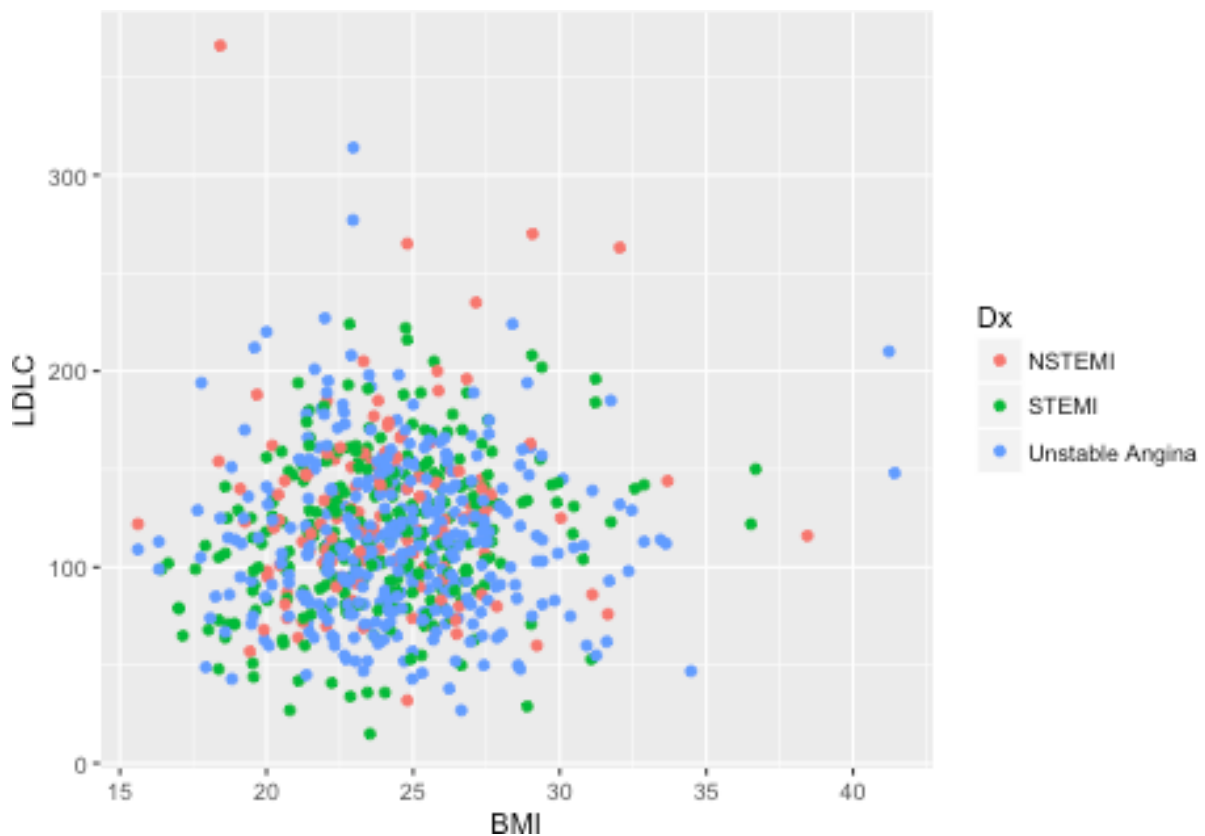
5.2 Part1

```
tophit <- tophitters2001[1:25,]
tophit <- tophit[,c("name", "lg", "avg")]

# y축 이산형 그래프
ggplot(tophit, aes(x = avg, y = name, col=lg)) + geom_point()
```

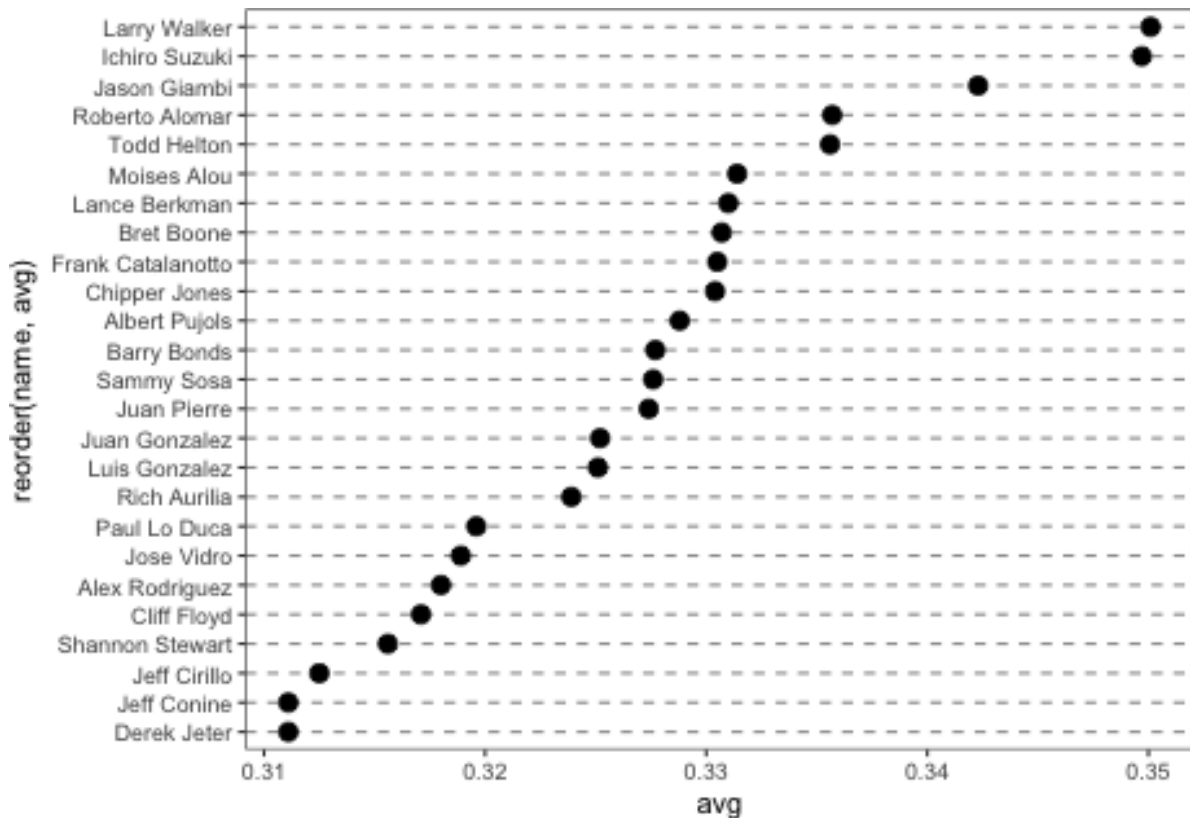


```
ggplot(acs, aes(x = BMI, y = LDLC, col=Dx)) + geom_point()
## Warning: Removed 106 rows containing missing values
## (geom_point).
```



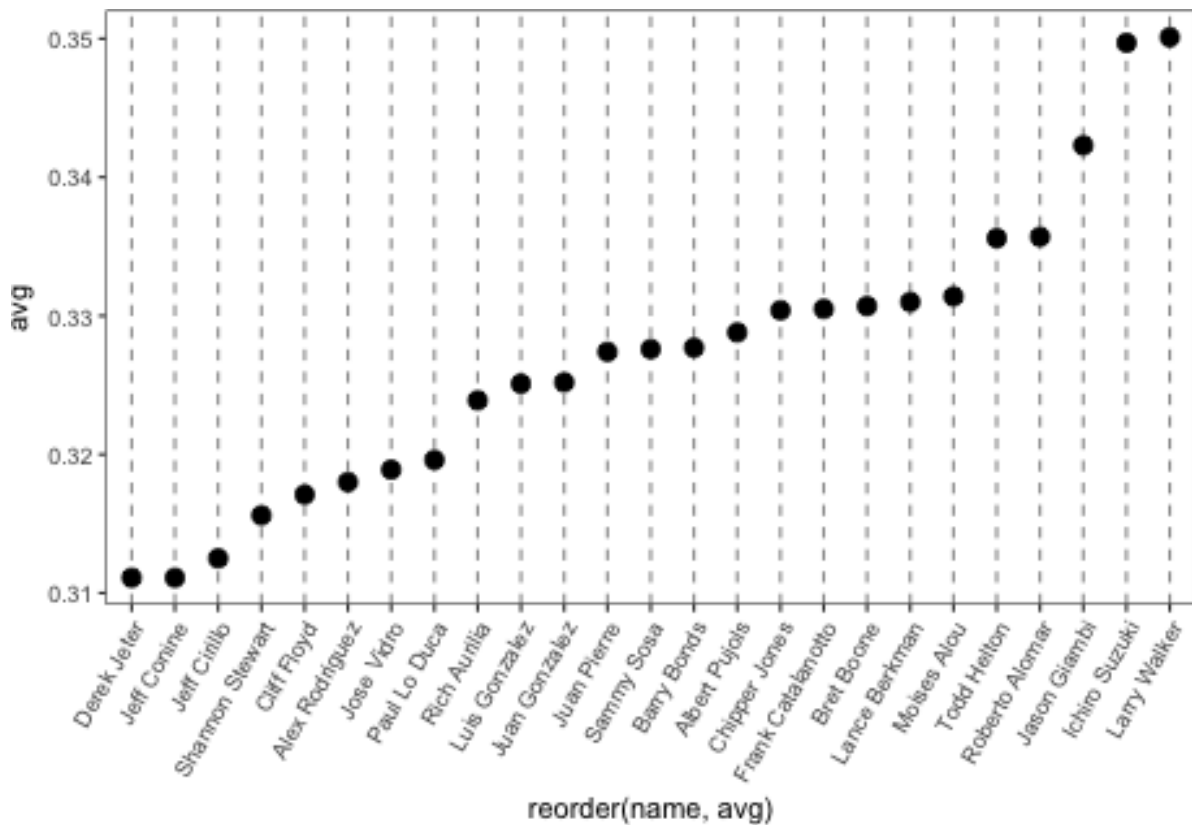
그래프 정렬하기 & 그래프 격자 없애기 & 수평선 점선으로 바꾸기

```
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"))
```



x, y축 바꿔서 그치기 & 그래프 격자 없애기 & 수직선 점선으로 바꾸기 & x축 값 정의 및 회전

```
ggplot(tophit, aes(x = reorder(name, avg), y = avg)) + geom_point(size = 3) + theme_bw() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_line(color = "grey60", linetype = "dashed"))
```

두개 변수로 정의하여 그리기

```
nameorder <- tophit$name[order(tophit$lg, tophit$avg)]
```

```
tophit$name <- factor(tophit$name, levels = nameorder)
```

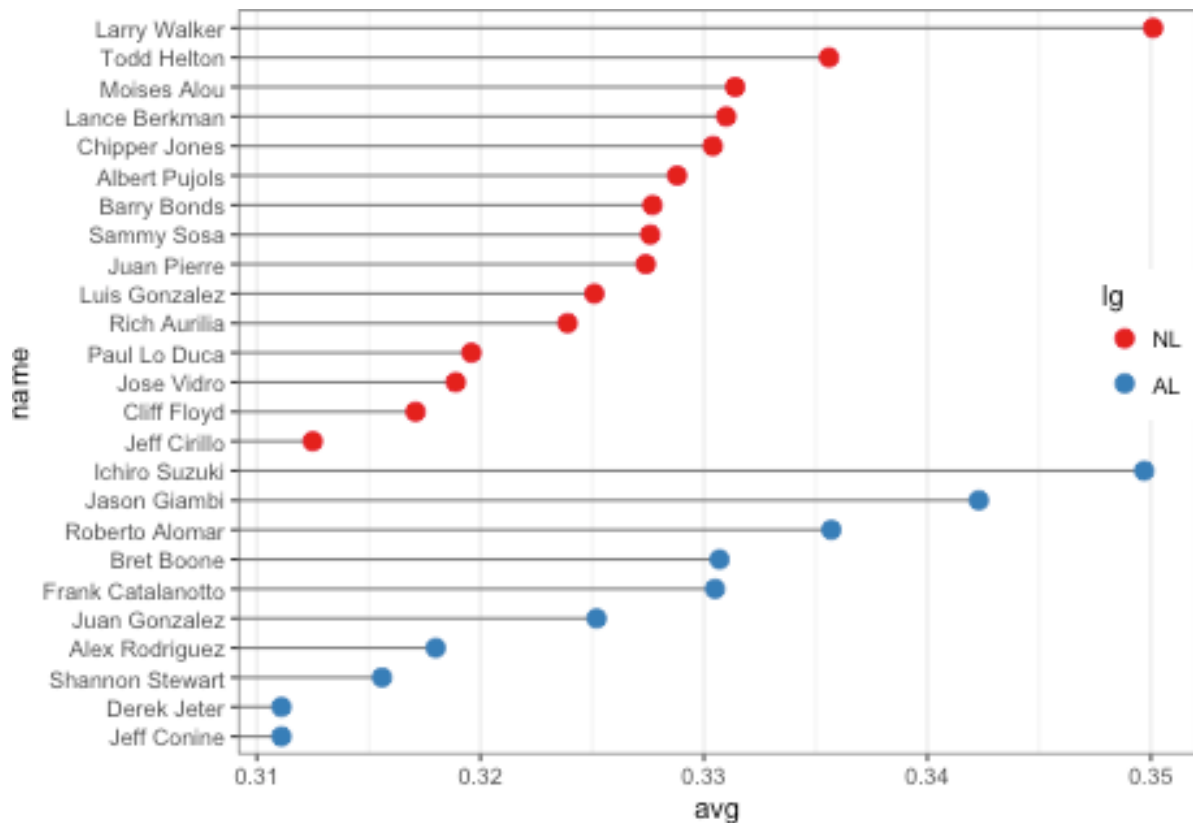
격자 선이 그래프의 끝에서 끝까지 횡단하지 않고, 점까지만 가도록 표현 [geom_segment]

```
ggplot(tophit, aes(x = avg, y = name)) +
```

```
  geom_segment(aes(yend = name), xend = 0, color = "grey50") + geom_point(size = 3, aes(color =
```

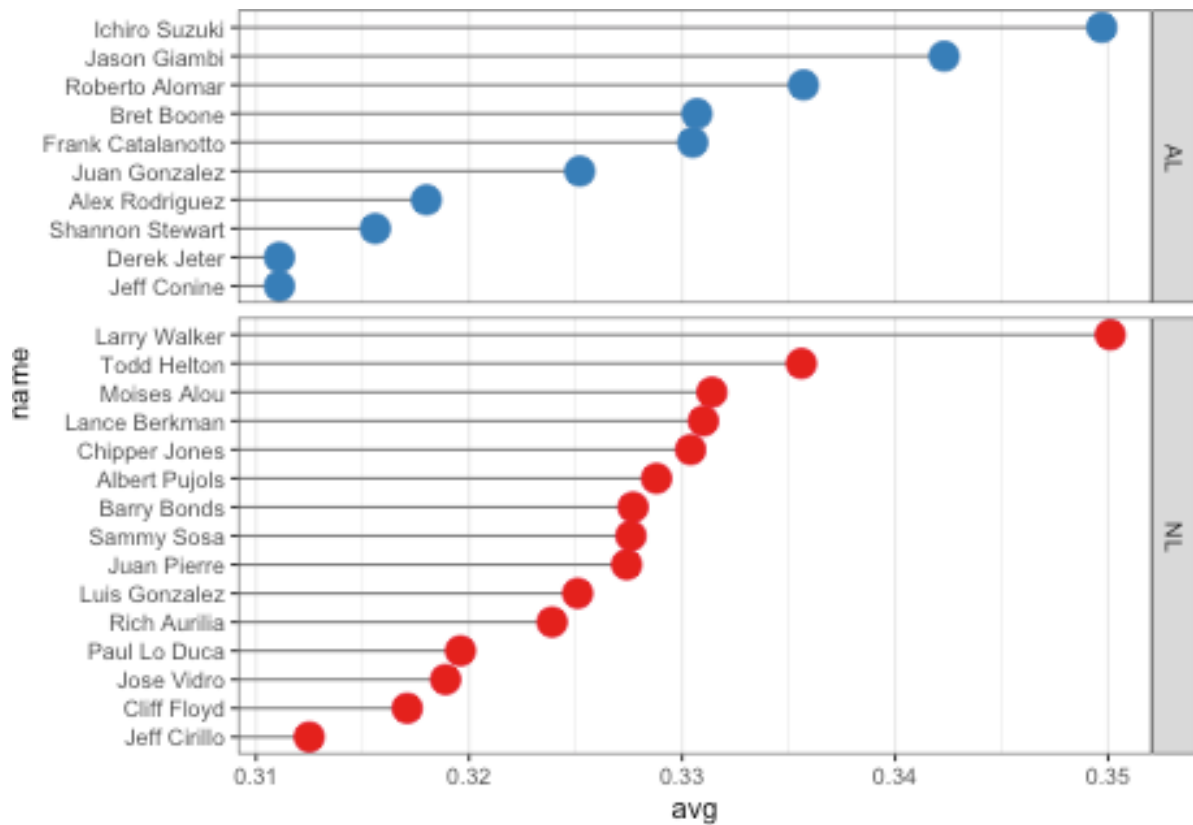
```
  scale_color_brewer(palette = "Set1", limits = c("NL", "AL"))) + theme_bw() +
```

```
  theme(panel.grid.major.y = element_blank(), legend.position = c(1, 0.55), # 범례를 그래프 안쪽  
    legend.justification = c(1, 0.5))
```



그룹 별 그래프 분할

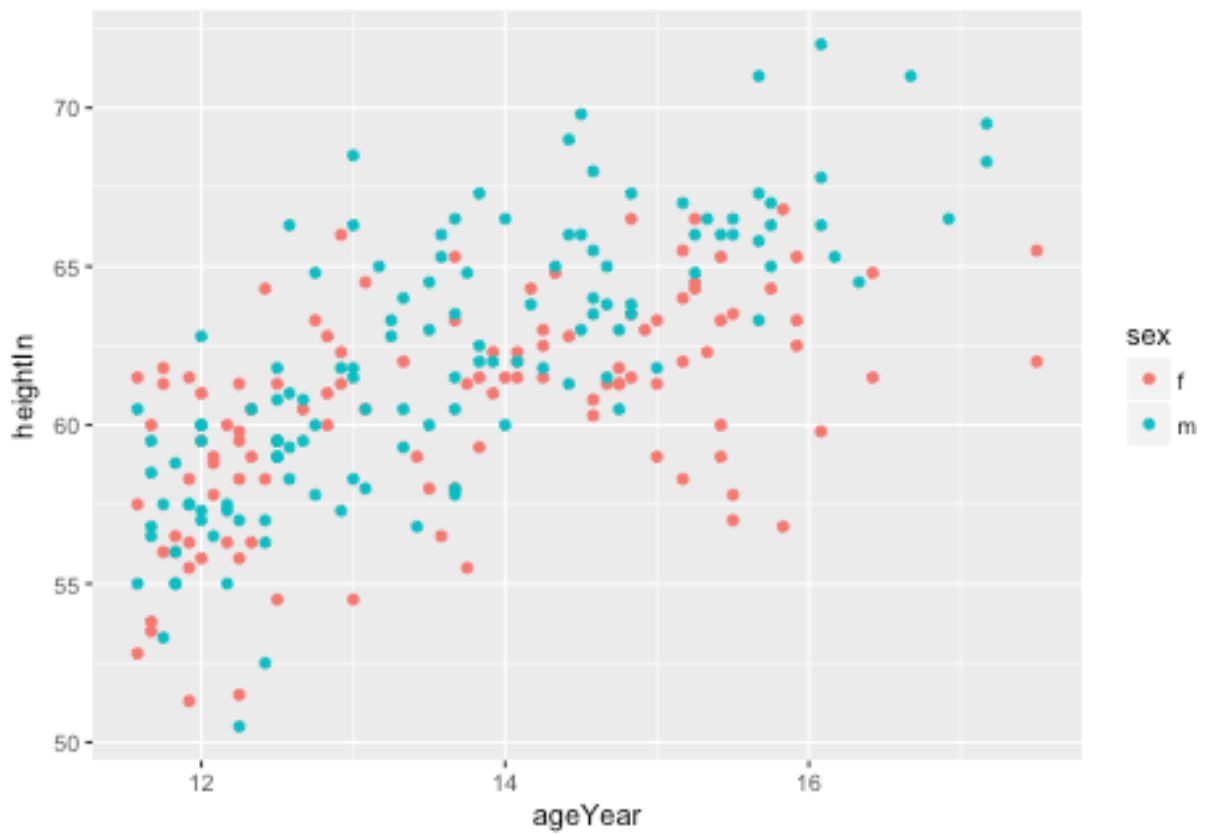
```
ggplot(tophit, aes(x = avg, y = name)) +
  geom_segment(aes(yend = name), xend = 0, color = "grey50") + geom_point(size = 5, aes(color = lg)) +
  scale_color_brewer(palette = "Set1", limits = c("NL", "AL"), guide = FALSE) + theme_bw() +
  theme(panel.grid.major.y = element_blank()) + facet_grid(lg ~ ., scales = "free_y", space = "fill")
```



그룹 별 데이터 구별

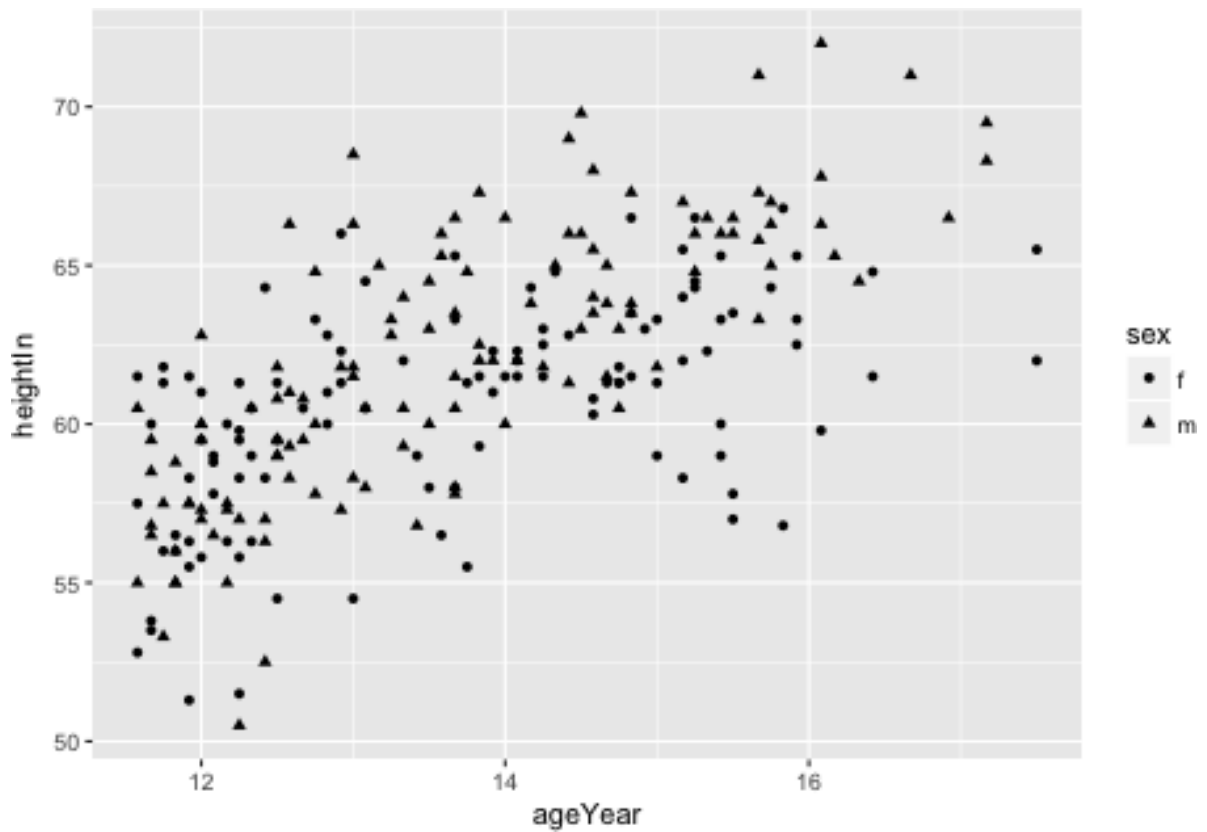
색상기준

```
ggplot(heightweight, aes(x = ageYear, y = heightIn, color = sex)) + geom_point()
```

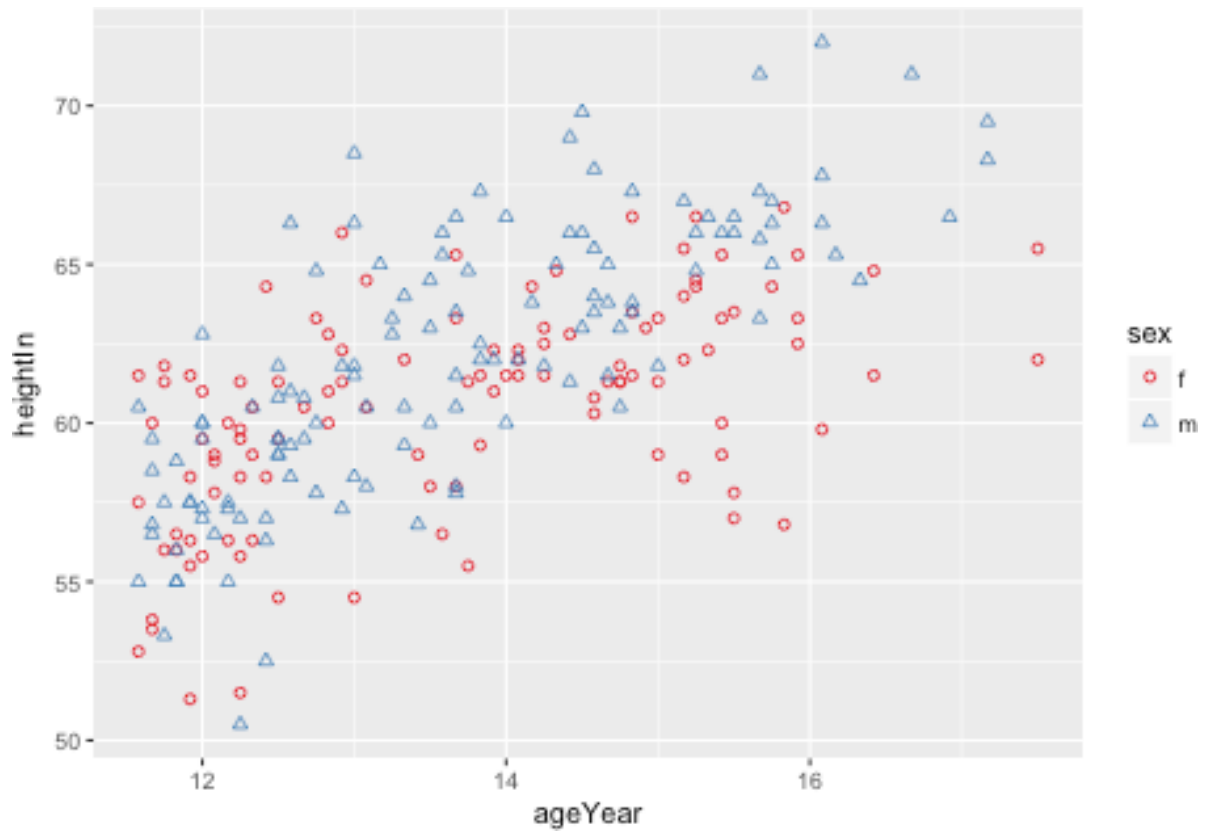


점 모양 기준

```
ggplot(heightweight, aes(x = ageYear, y = heightIn, shape = sex)) + geom_point()
```



```
ggplot(heightweight, aes(x = ageYear, y = heightIn, color = sex, shape = sex)) + geom_point() +  
  scale_shape_manual(values = c(1,2)) + scale_color_brewer(palette = "Set1")
```



```
ggplot(heightweight, aes(x = ageYear, y = heightIn, color = sex, shape = sex)) + geom_point() +  
  scale_shape_manual(values = c(3,2)) + scale_color_brewer(palette = "Set1")
```



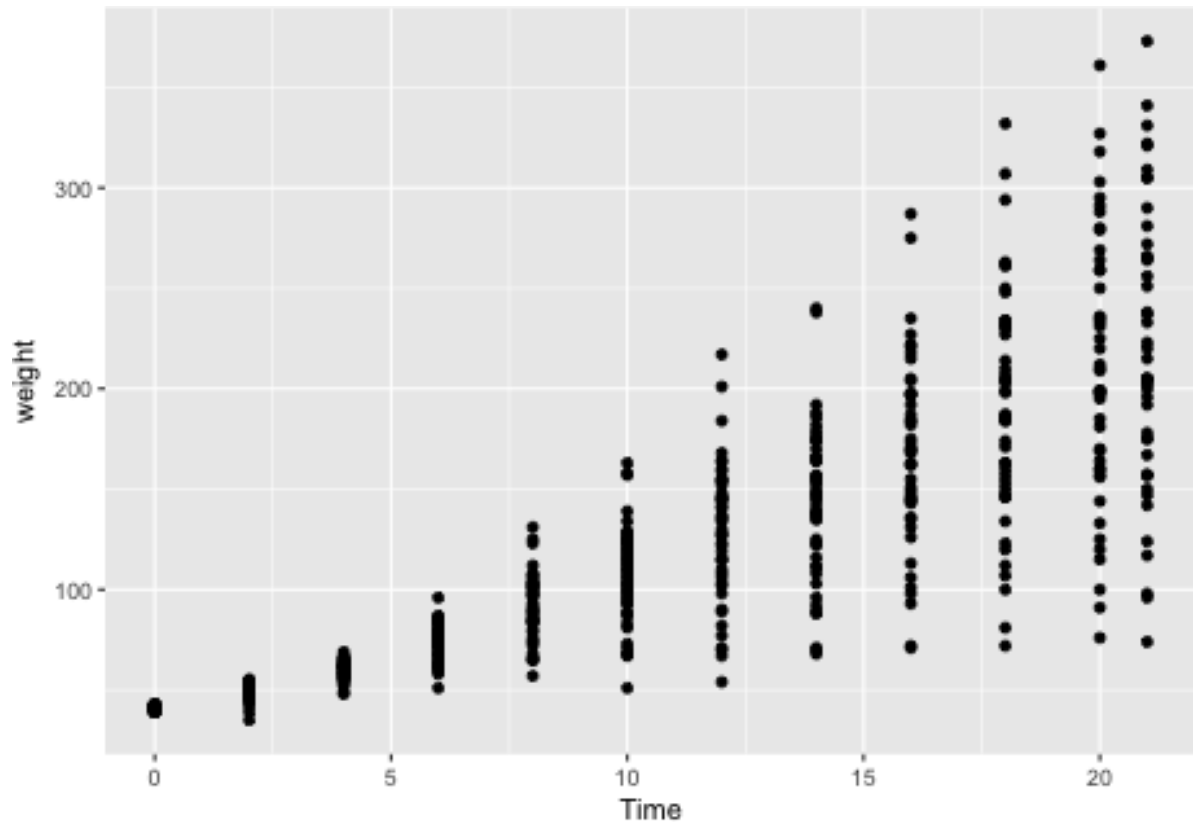
기준 정의에 따라 구별

```
hw <- heightweight
```

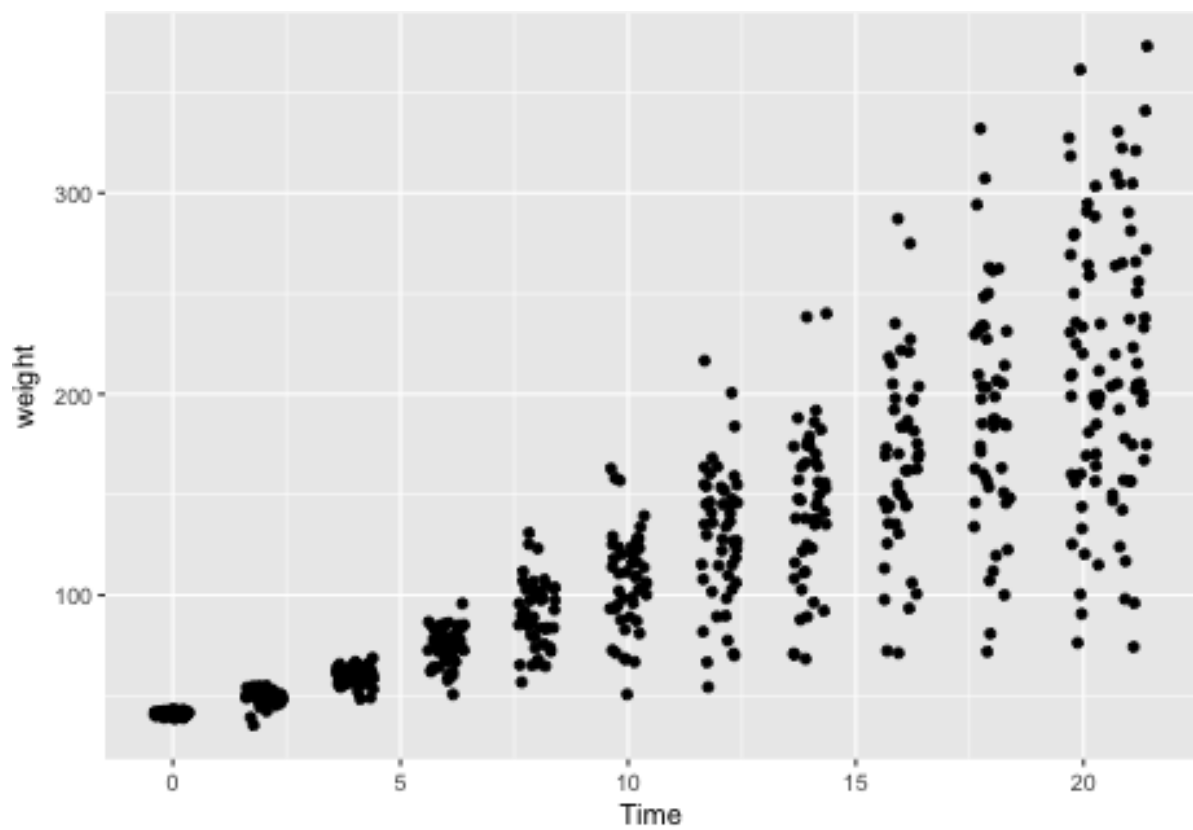
```
hw$weightGroup <- cut(hw$weightLb, breaks = c(-Inf, 100, Inf), labels = c("< 100", ">= 100"))
```

x축이 이산형일 때 점들을 랜덤하게 조금씩 이동시켜 표현

```
ggplot(ChickWeight, aes(x = Time, y = weight)) + geom_point()
```

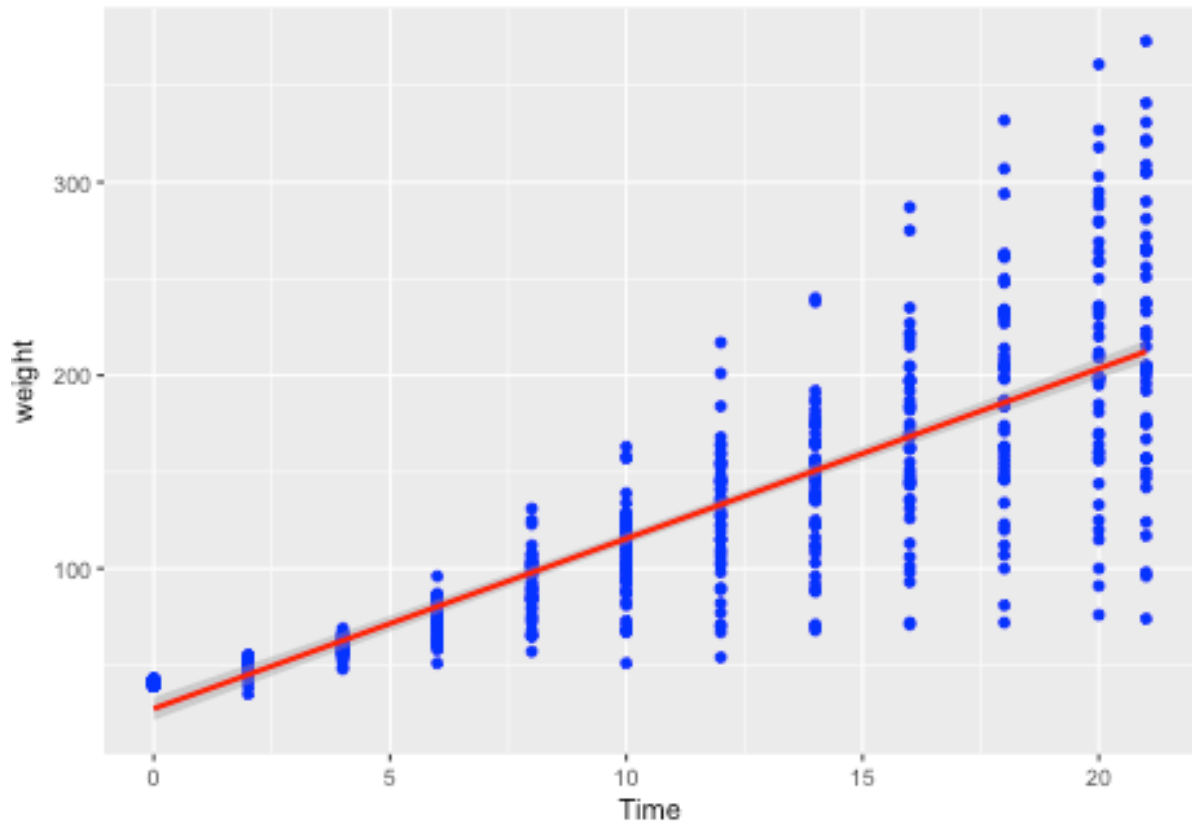


```
ggplot(ChickWeight, aes(x = Time, y = weight)) + geom_jitter() # 이동 적용
```

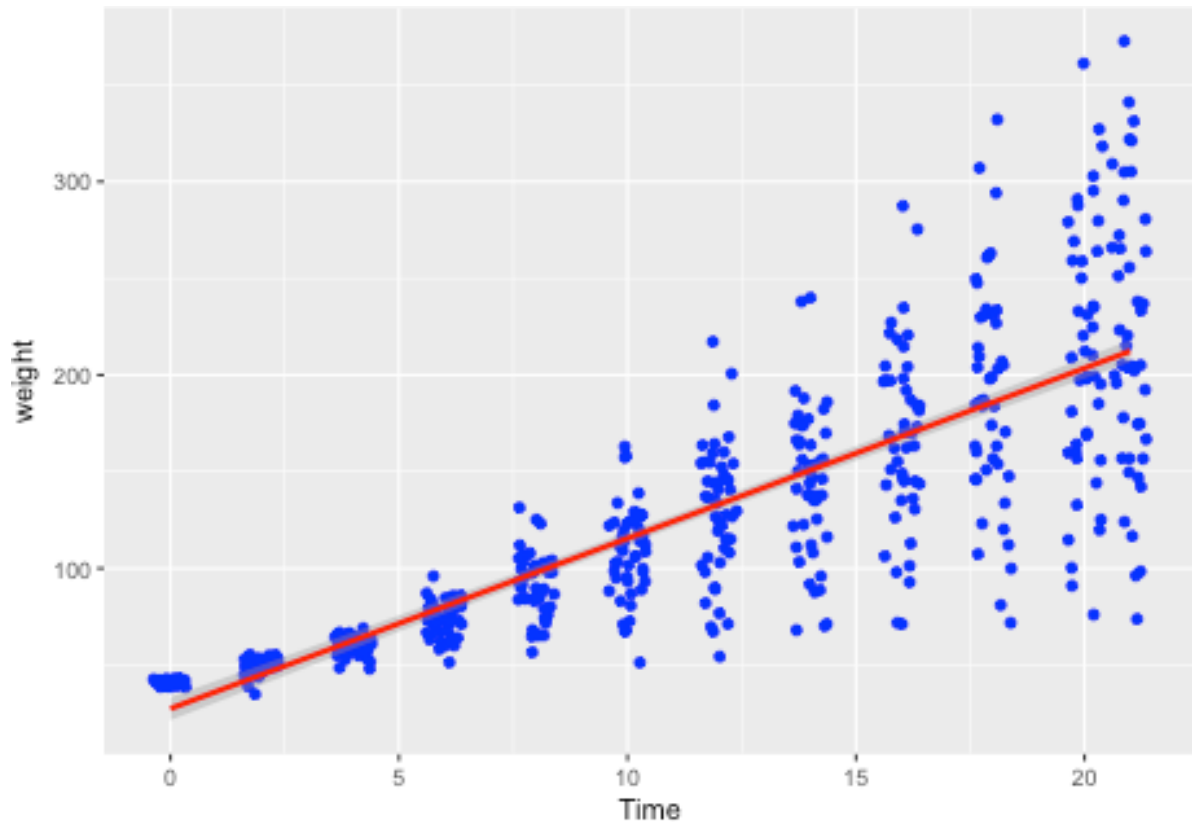


적합한 회귀선 추가하기

```
sp <- ggplot(ChickWeight, aes(x = Time, y = weight))  
sp + geom_point(color = "blue") + stat_smooth(method = lm, se = TRUE, color = "red")
```

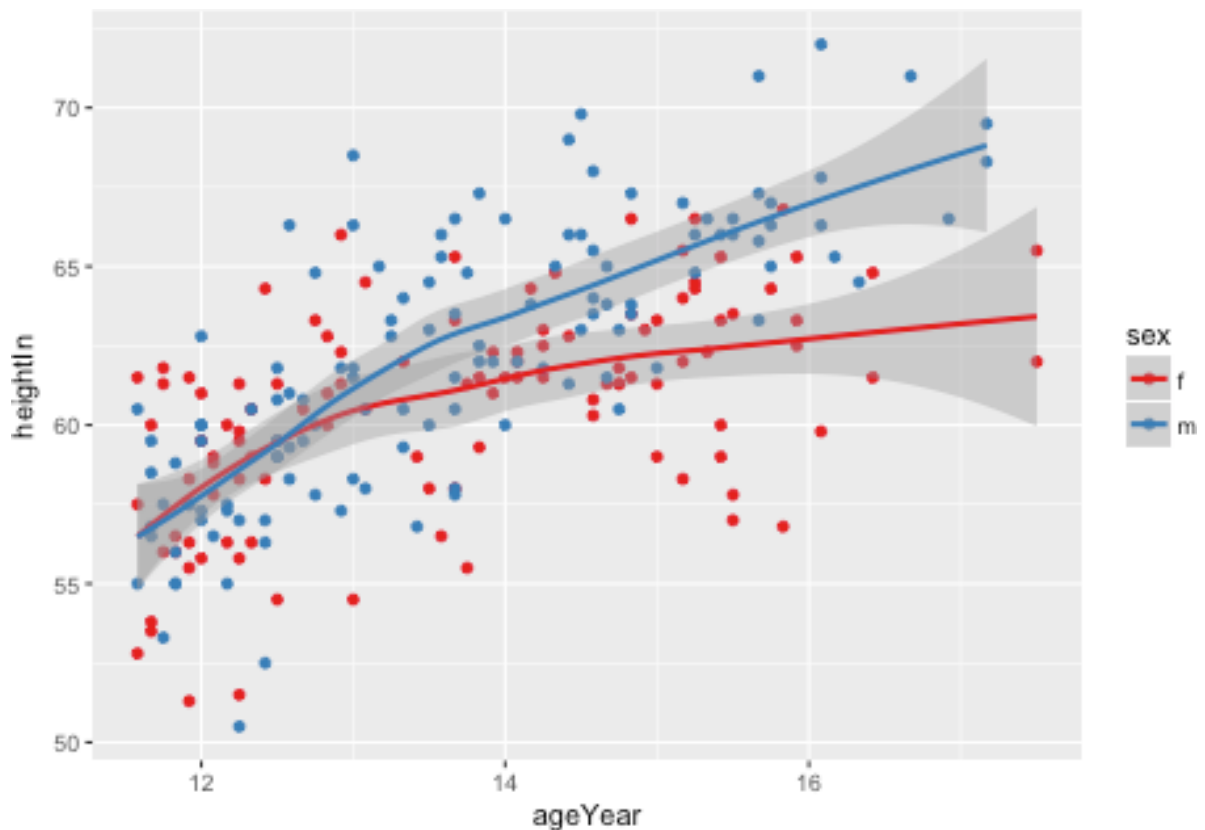


```
sp + geom_jitter(color = "blue") + stat_smooth(method = lm, se = TRUE, color = "red")
```

#그룹 별 회귀선 추가하기

```
sps <- ggplot(heightweight, aes(x = ageYear, y = heightIn, color = sex)) +  
  geom_point() + scale_color_brewer(palette = "Set1")  
sps + geom_smooth()  
## `geom_smooth()` using method = 'loess'
```



예측값 실제값 그래프로 표현하기 (함수)

```
predictvals <- function(model, xvar, yvar, xrange = NULL, sample = 100, ...){
  if(is.null(xrange)){
    if(any(class(model) %in% c("lm", "glm"))){
      xrange <- range(model$model[[xvar]])
    } else if(any(class(model) %in% "loess")){
      xrange <- range(model$x)
    }
  }
}
```

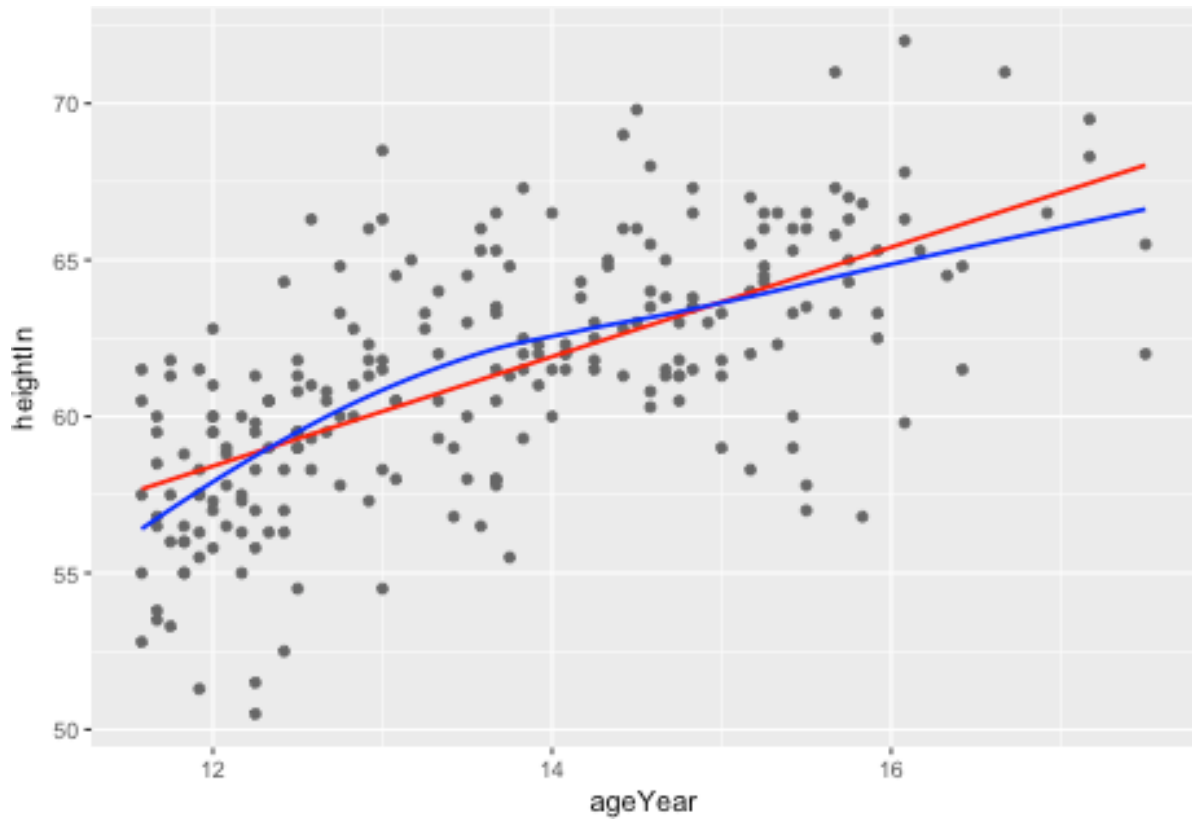
```
newdata <- data.frame(x = seq(xrange[1], xrange[2], length.out = sample))
names(newdata) <- xvar
newdata[[yvar]] <- predict(model, newdata = newdata, ...)
newdata
```

```
modlinear <- lm(heightIn ~ ageYear, heightweight)
modloess <- loess(heightIn ~ ageYear, heightweight)
```

```
lm_predicted <- predictvals(modlinear, "ageYear", "heightIn")
loess_predicted <- predictvals(modloess, "ageYear", "heightIn")
```

```
sp <- ggplot(heightweight, aes(x = ageYear, y = heightIn)) +
```

```
geom_point(color = "grey40")
sp + geom_line(data = lm_predicted, color = "red", size = 0.8) +
  geom_line(data = loess_predicted, color = "blue", size = 0.8)
```

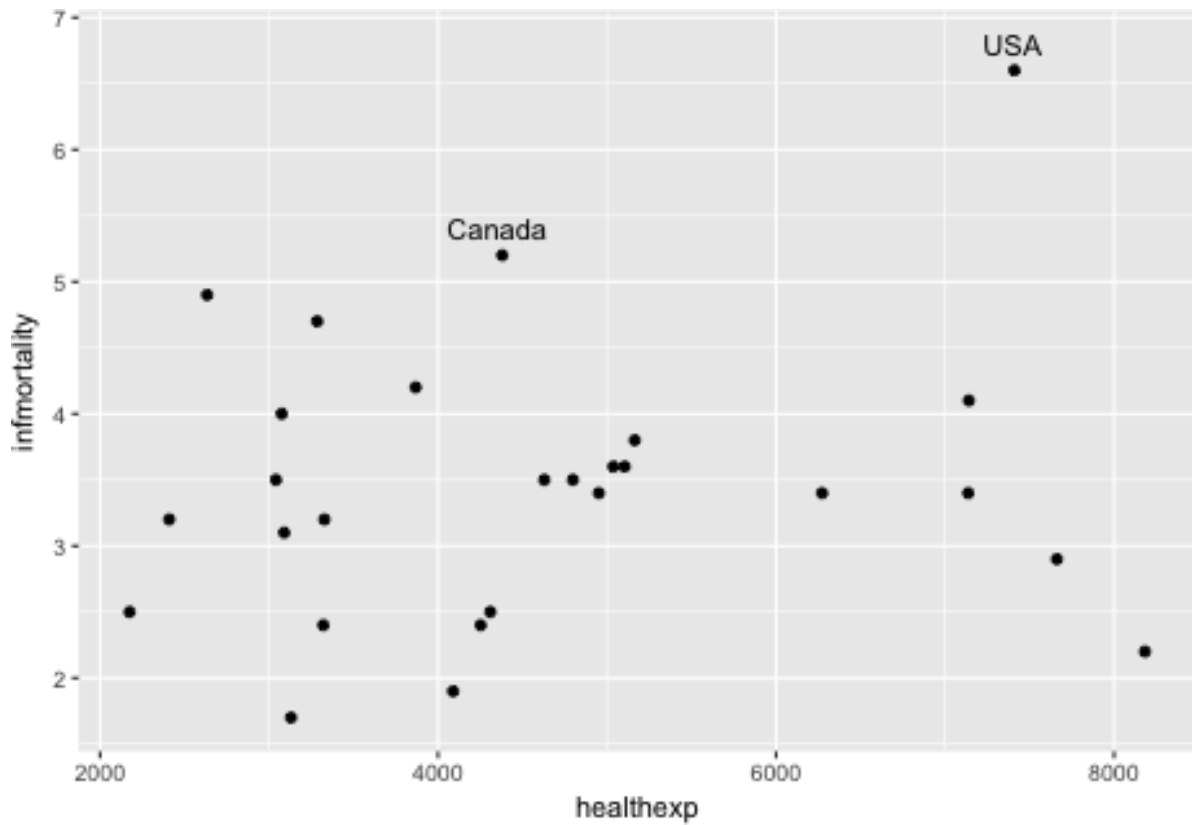


#산점도의 점에 라벨 붙이기

```
sp <- ggplot(subset(countries, Year == 2009 & healthexp > 2000),
  aes(x = healthexp, y = infmortality)) + geom_point()
```

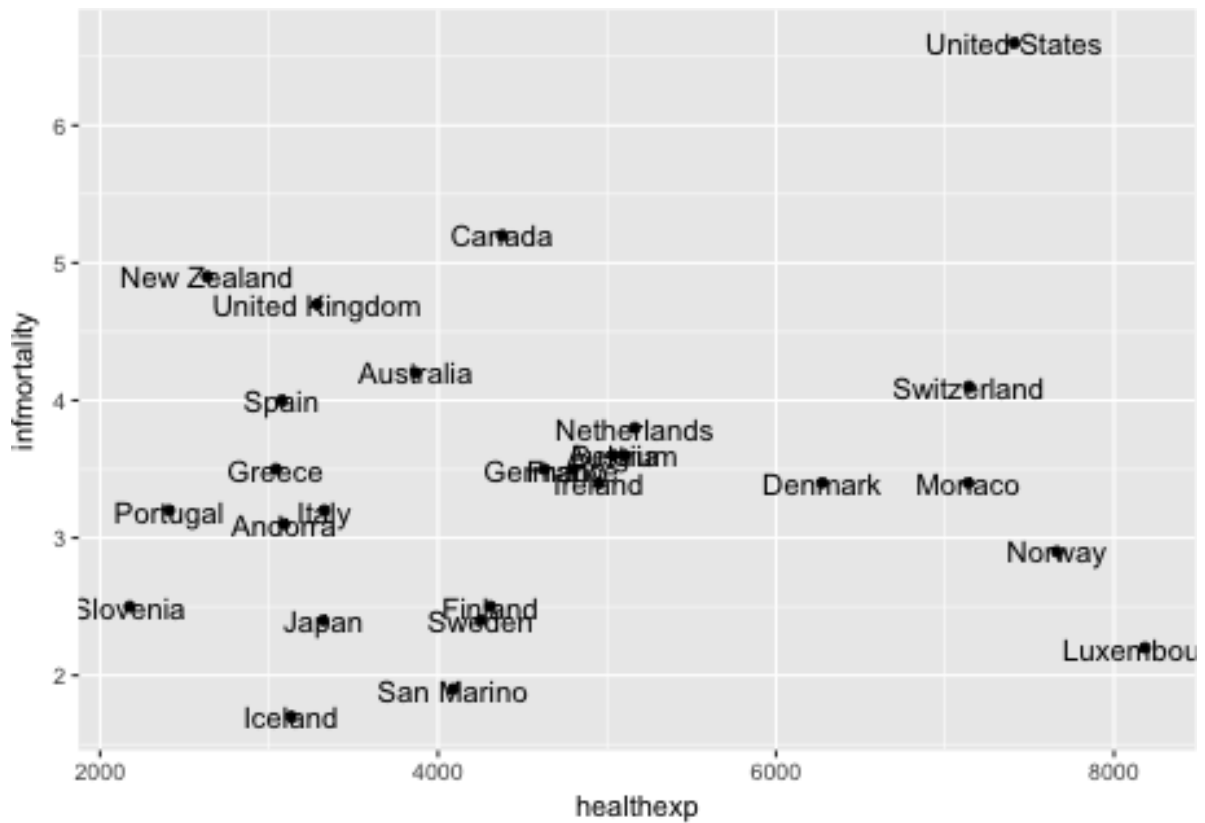
특정 값에 특정 단어로 라벨 붙이기

```
sp + annotate("text", x = 4350, y = 5.4, label = "Canada") +
  annotate("text", x = 7400, y = 6.8, label = "USA")
```



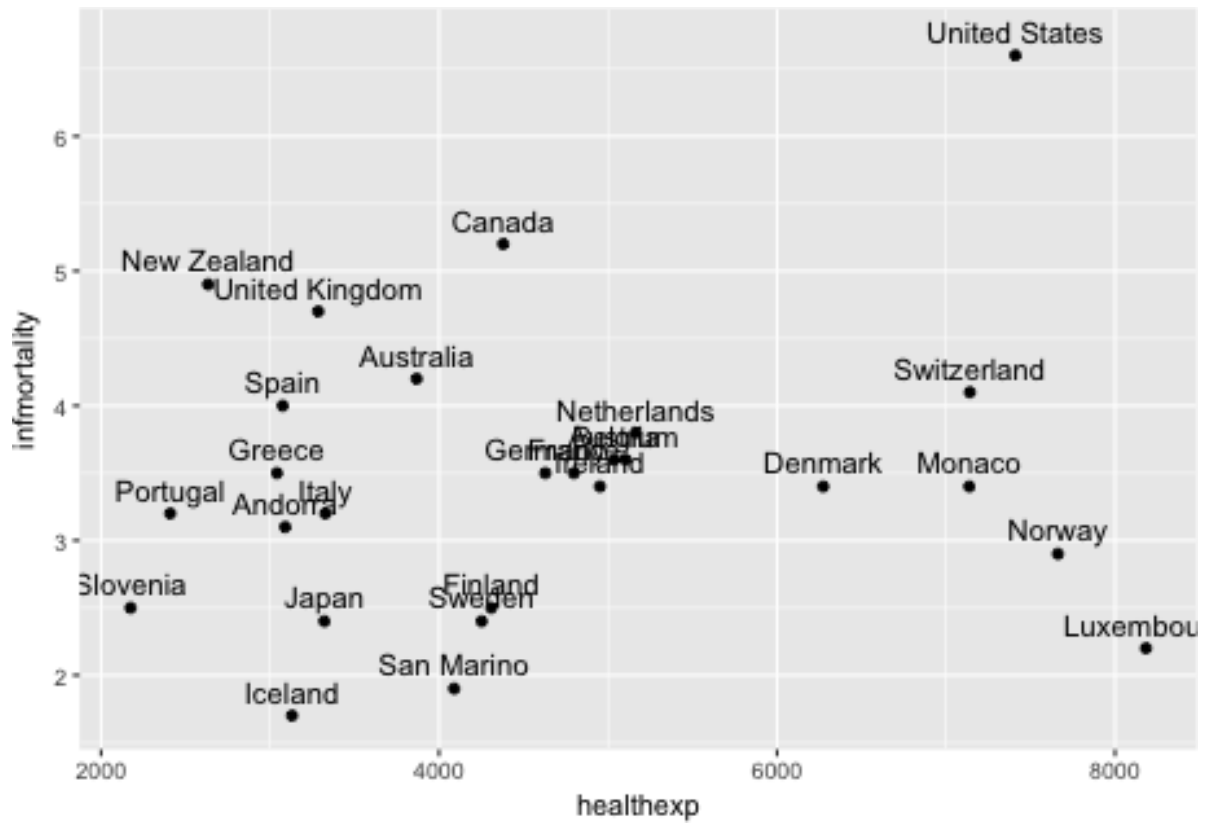
데이터 값을 라벨로 붙이기

```
sp + geom_text(aes(label = Name), size = 4)
```

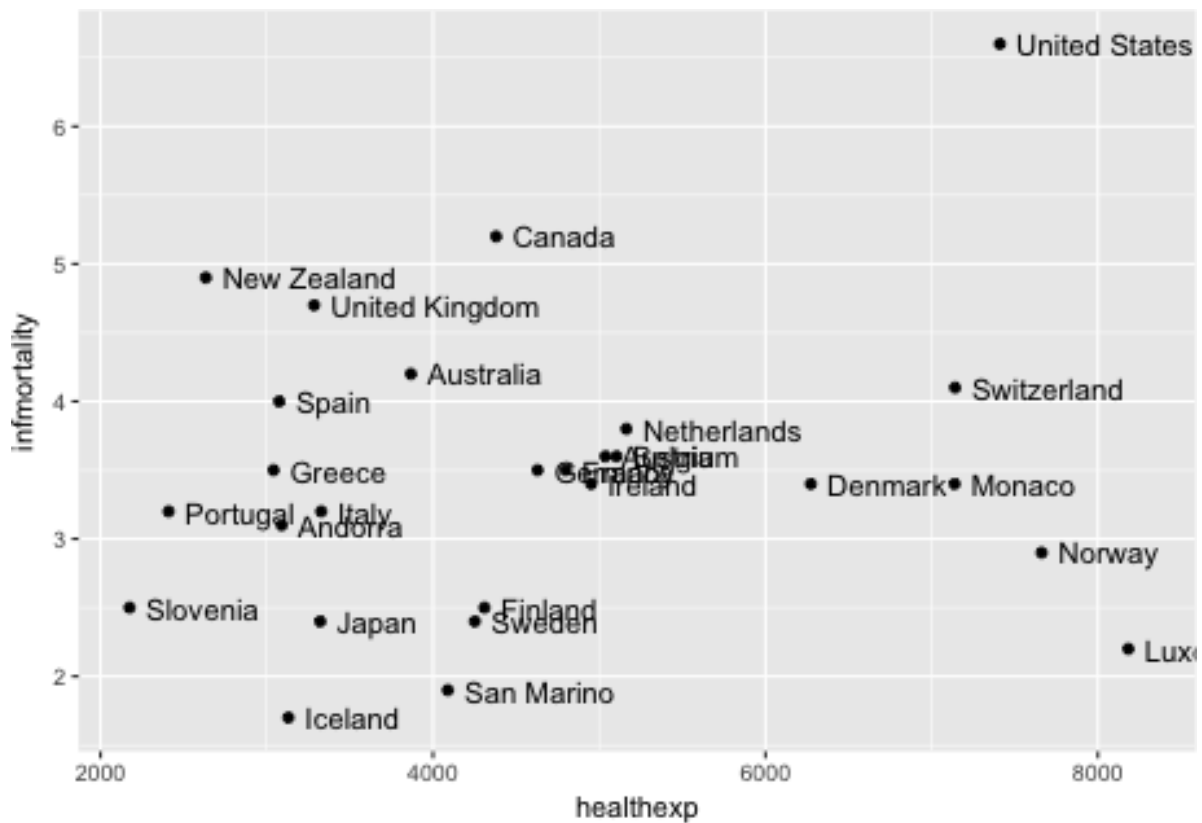


라벨의 위치를 데이터값보다 조금 크게 설정

```
sp + geom_text(aes(y = infmortality + 0.1, label = Name), size = 4, vjust = 0)
```



```
sp + geom_text(aes(x = healthexp + 100, label = Name), size = 4, hjust = 0)
```



특정 값만 라벨 붙이기

```
cdat <- subset(countries, Year == 2009 & healthexp > 2000)
```

```
cdat$Name1 <- cdat$Name
```

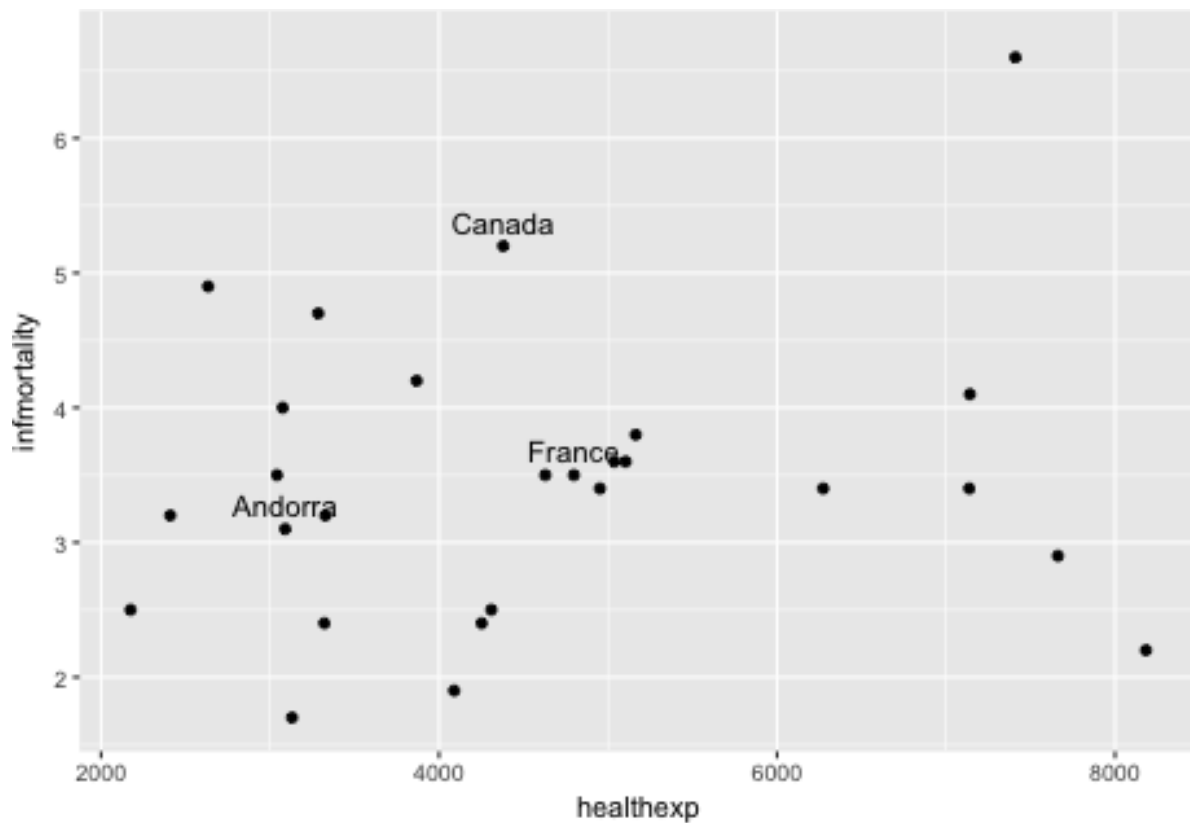
```
idx <- cdat$Name %in% c("Andorra", "France", "Canada")
```

```
cdat$Name1[!idx] <- NA
```

```
ggplot(cdat, aes(x = healthexp, y = infmortality)) + geom_point() + geom_text(aes(y = infmortality))
```

```
## Warning: Removed 24 rows containing missing values
```

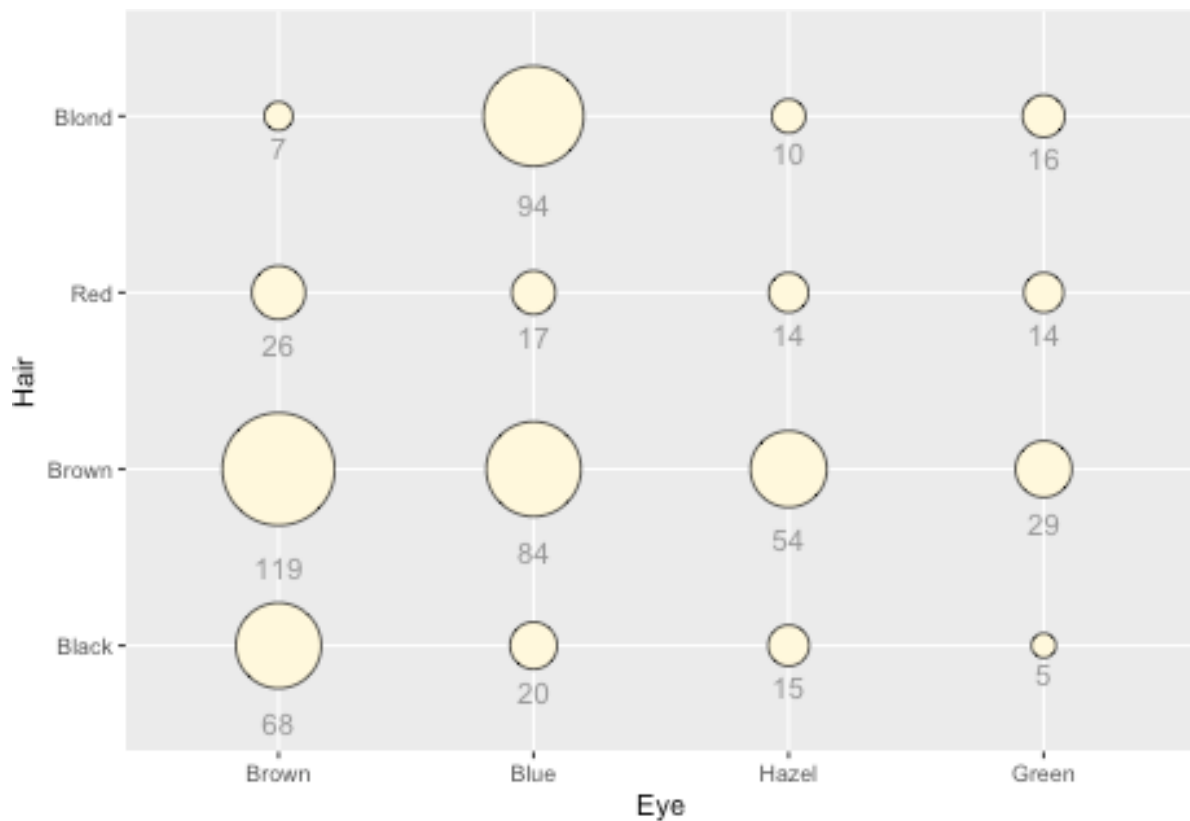
```
## (geom_text).
```



```
## Warning: Removed 24 rows containing missing values (geom_text).
```

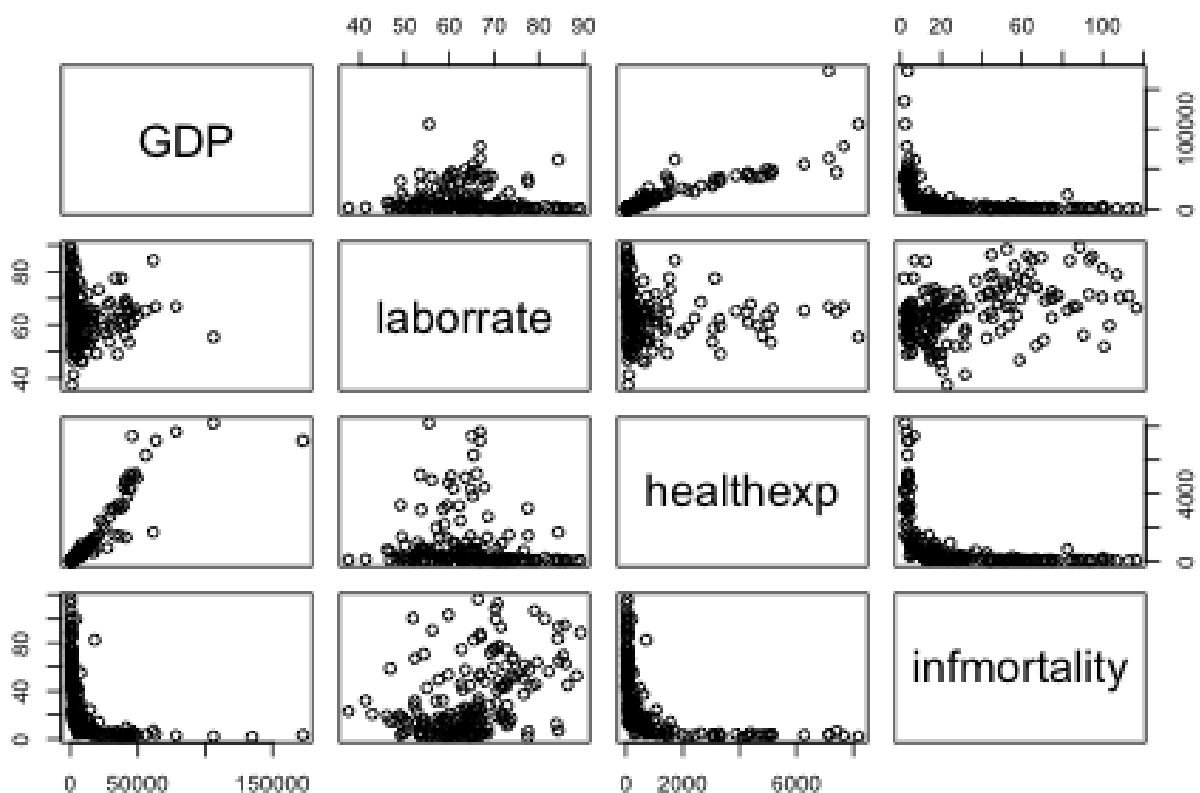
5.3 Part2

```
# 거품그래프(balloon plot)
hec <- HairEyeColor[,,"Male"] + HairEyeColor[,,"Female"]
hec <- melt(hec, value.name = "count")
ggplot(hec, aes(x = Eye, y = Hair)) + geom_point(aes(size = count), shape = 21, color = "black") +
  scale_size_area(max_size = 20, guide = FALSE) +
  geom_text(aes(y = as.numeric(Hair) - sqrt(count)/22, label = count), vjust = 1, color = "grey60")
```



산점도 행렬 만들기

```
c2009 <- subset(countries, Year == 2009, select = c(Name, GDP, laborrate, healthexp, infmortality))
plot(c2009[,2:5])
```




```

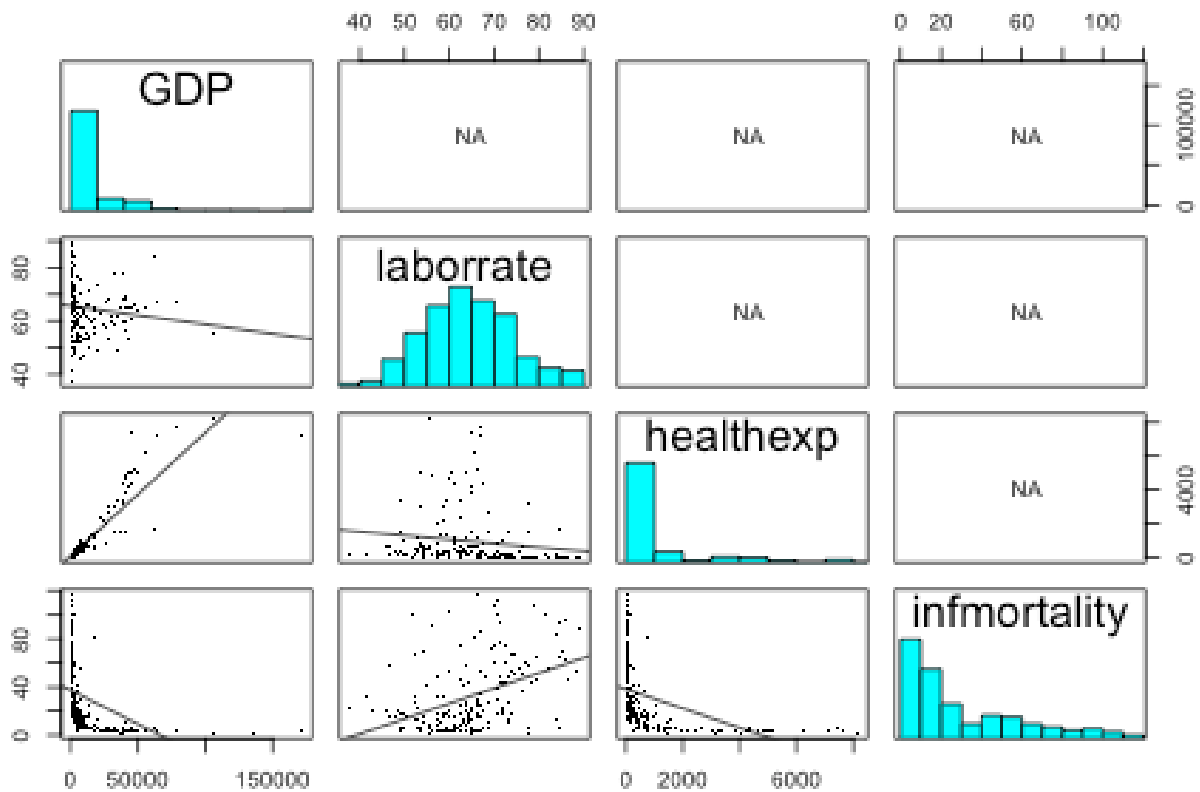
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

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 <- abs(cor(x, y))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

panel.lm <- function(x, y, col = par("col"), bg = NA, pch = par("pch"), cex = 1, col.smooth = "red",
  points(x, y, pch = pch, col = col, bg = bg, cex = cex)
  abline(stats::lm(y ~ x), col = col.smooth, ...)
}

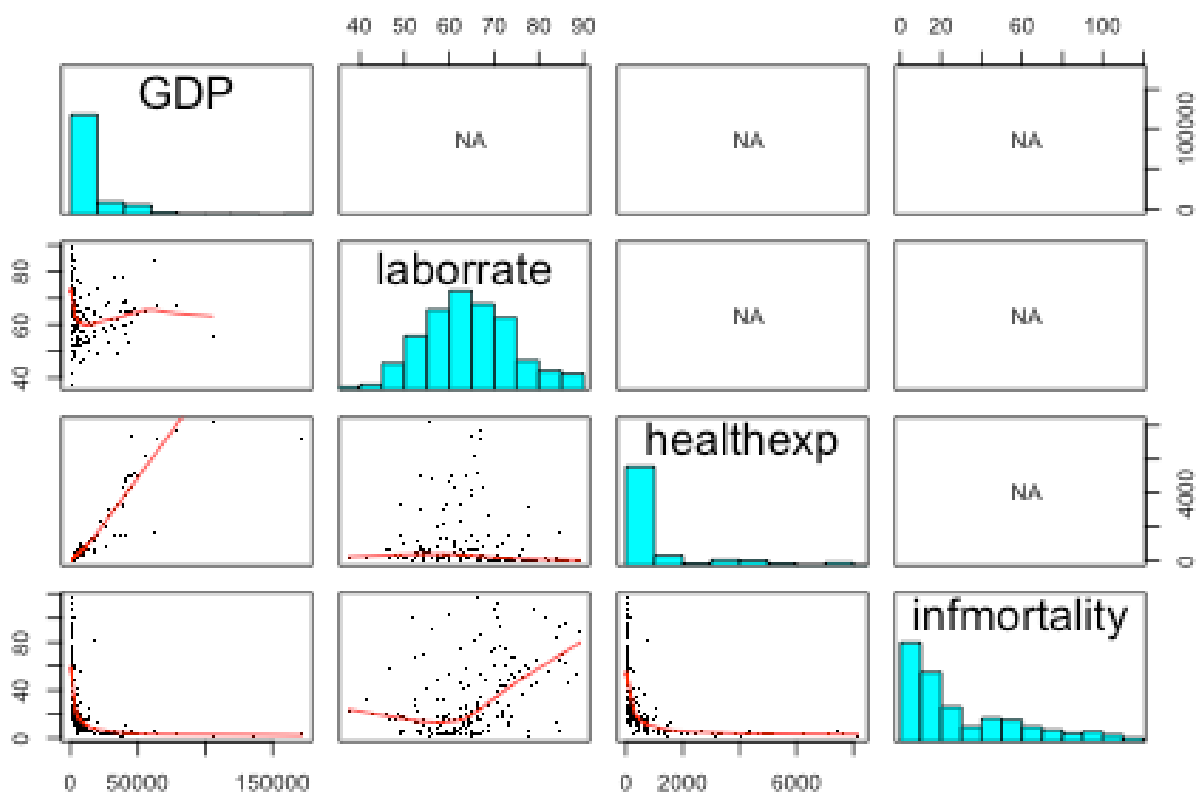
# 데이터에 대한 LOWESS선(평활선) 추가
pairs(c2009[,2:5], pch = ".", upper.panel = panel.cor, diag.panel = panel.hist, lower.panel = panel.lm)

```

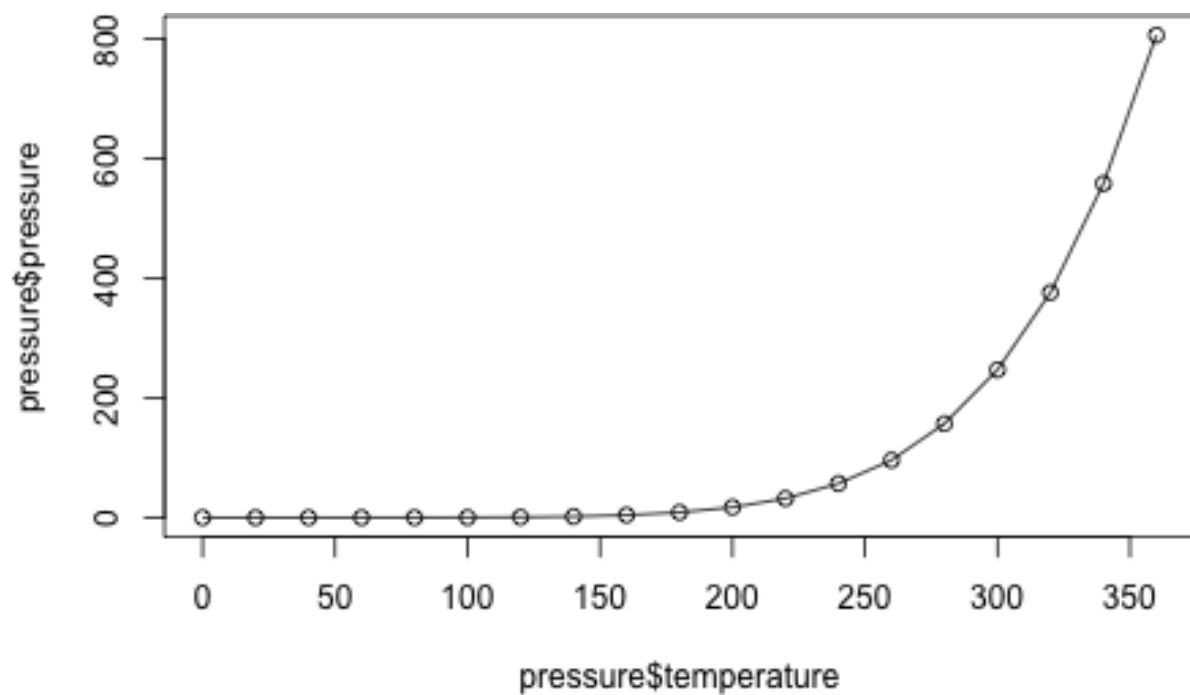


데이터에 대한 회귀적합선 추가

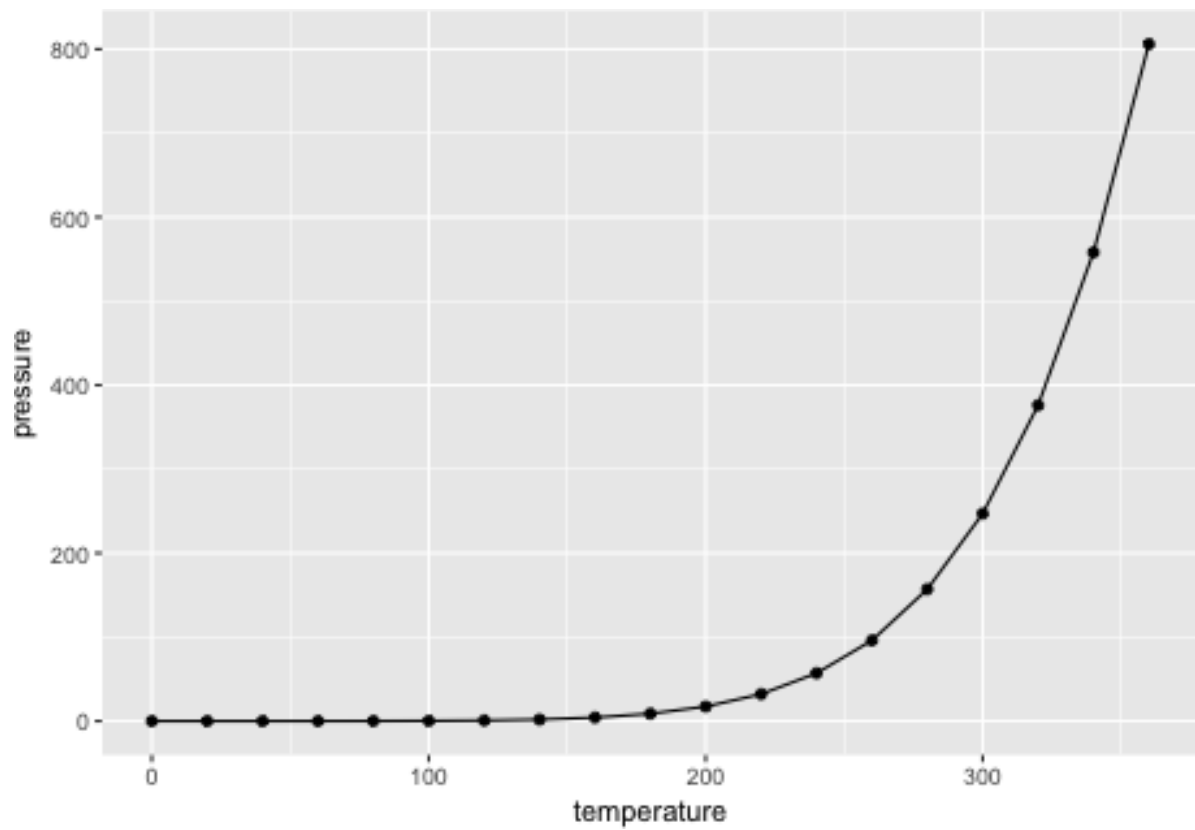
```
pairs(c2009[,2:5], pch = ".", upper.panel = panel.cor, diag.panel = panel.hist, lower.panel = p
```



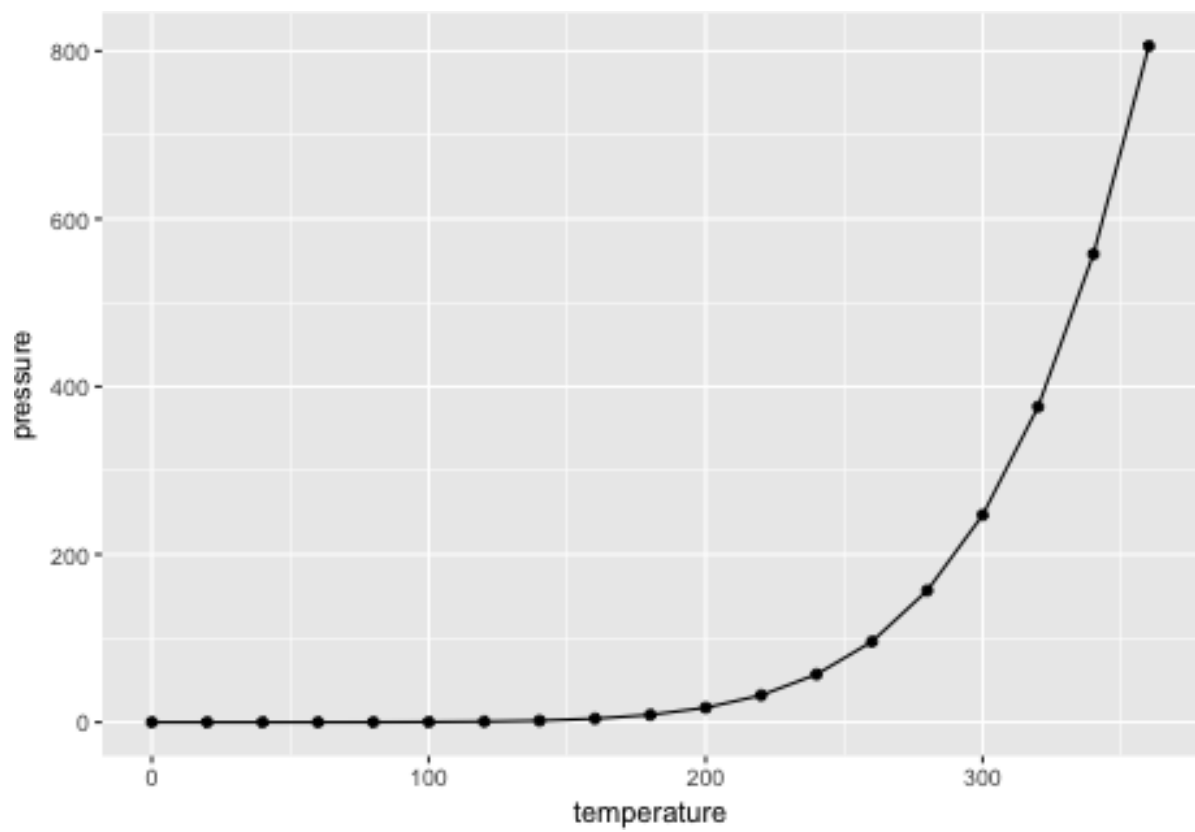
```
#####  
## scatter + line plot (geom_point/ geom_line)  
#####  
plot(pressure$temperature, pressure$pressure, type = "l")  
points(pressure$temperature, pressure$pressure)
```



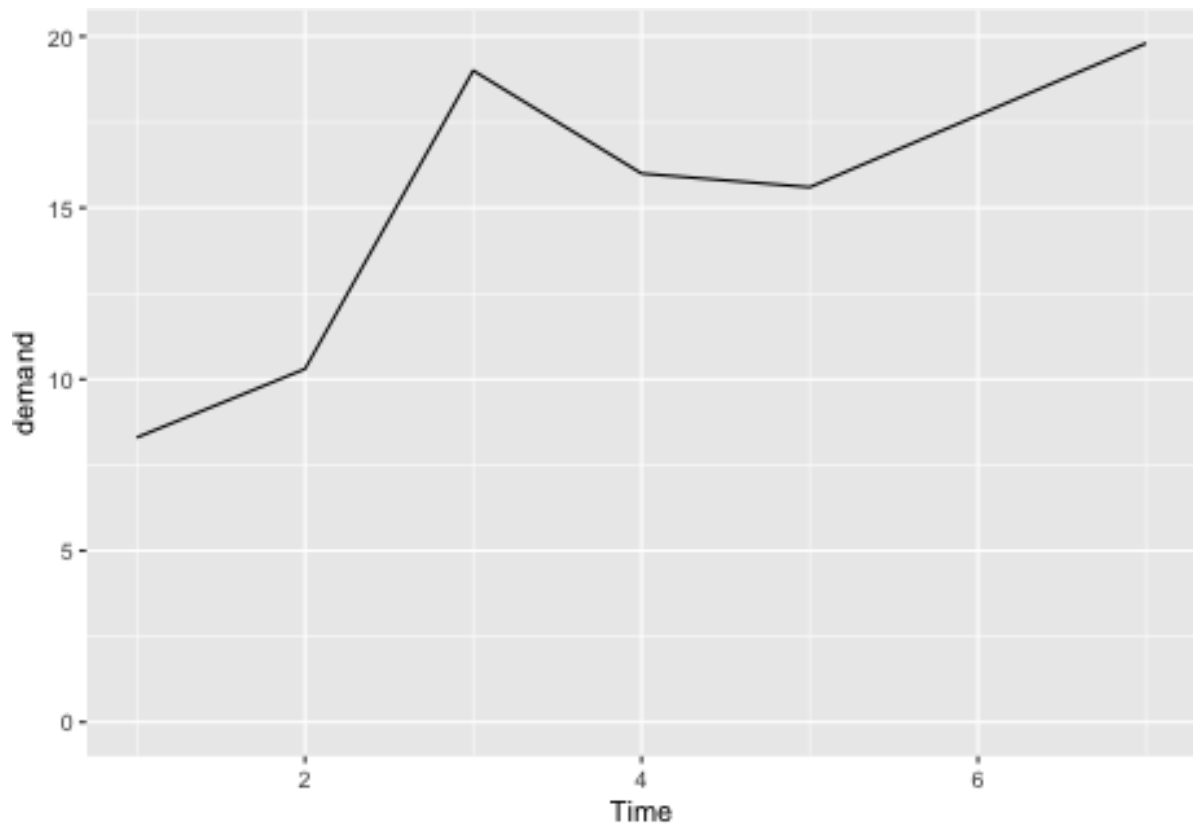
```
qplot(temperature, pressure, data = pressure, geom = c("line", "point"))
```



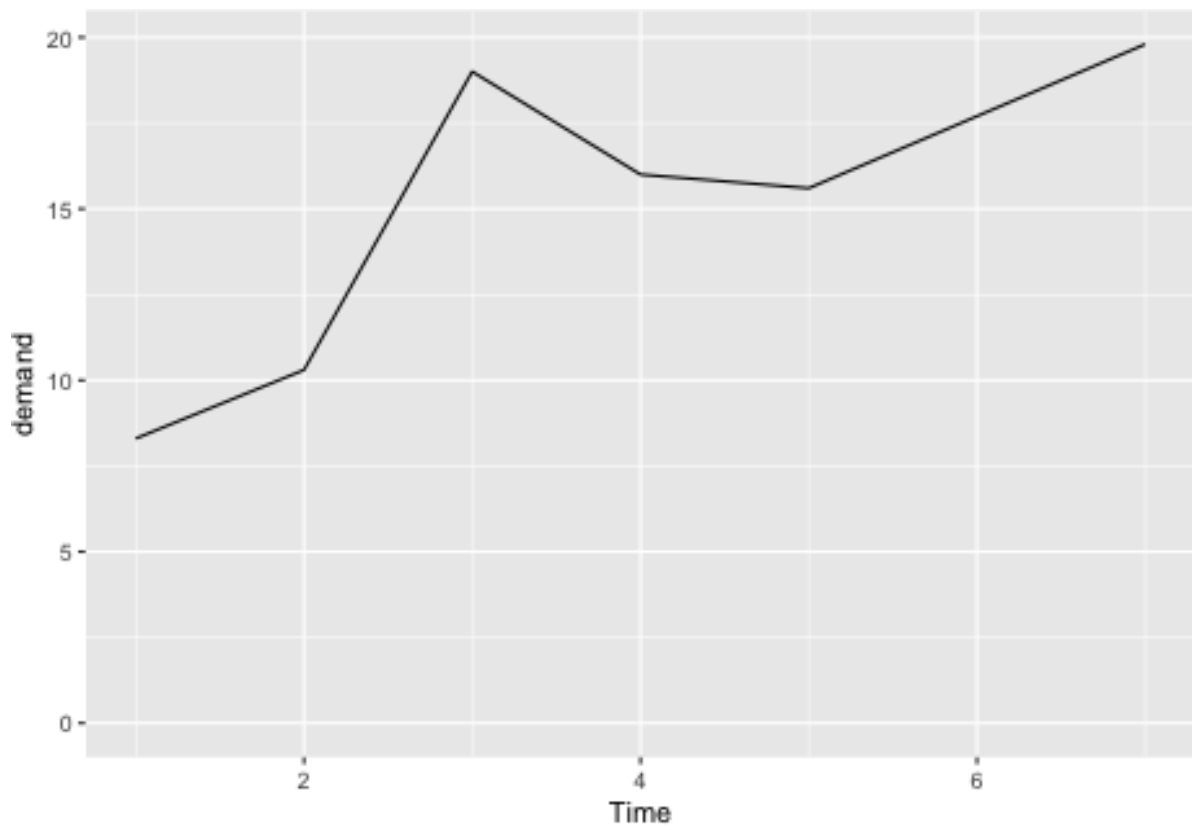
```
ggplot(pressure, aes(x = temperature, y = pressure)) + geom_point() + geom_line()
```



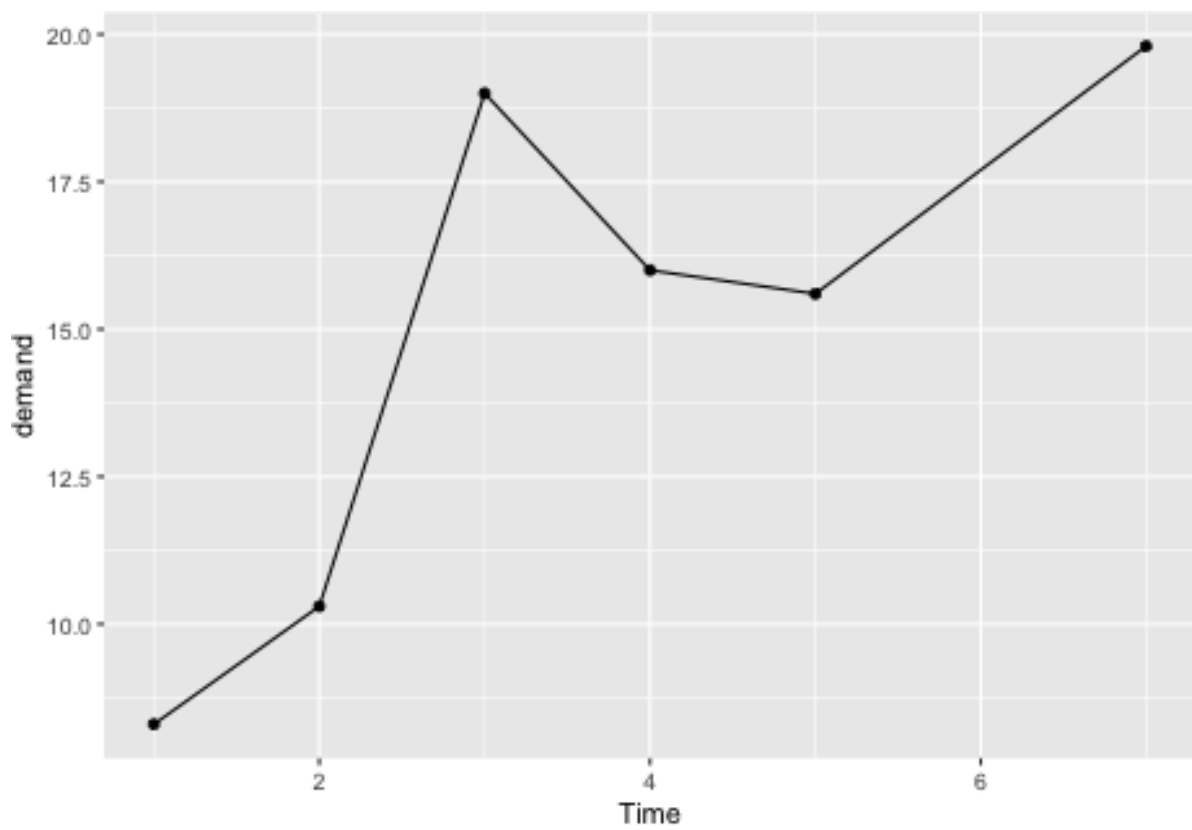
```
#####  
## line plot (geom_line)  
#####  
ggplot(BOD, aes(x = Time, y = demand)) + geom_line() + ylim(0, max(BOD$demand))
```



```
ggplot(BOD, aes(x = Time, y = demand)) + geom_line() + expand_limits(y = 0)
```



```
ggplot(BOD, aes(x = Time, y = demand)) + geom_line() + geom_point()
```

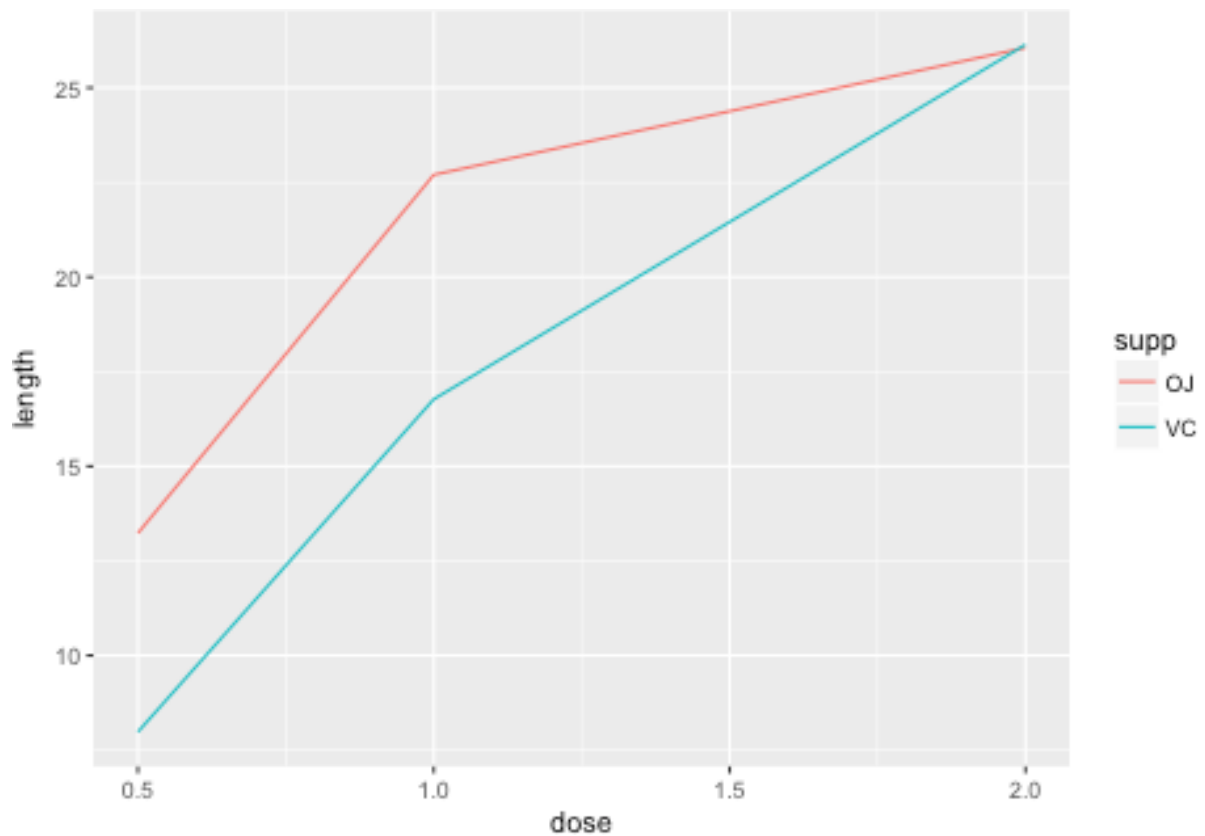


이산형 변수값에 따른 구분

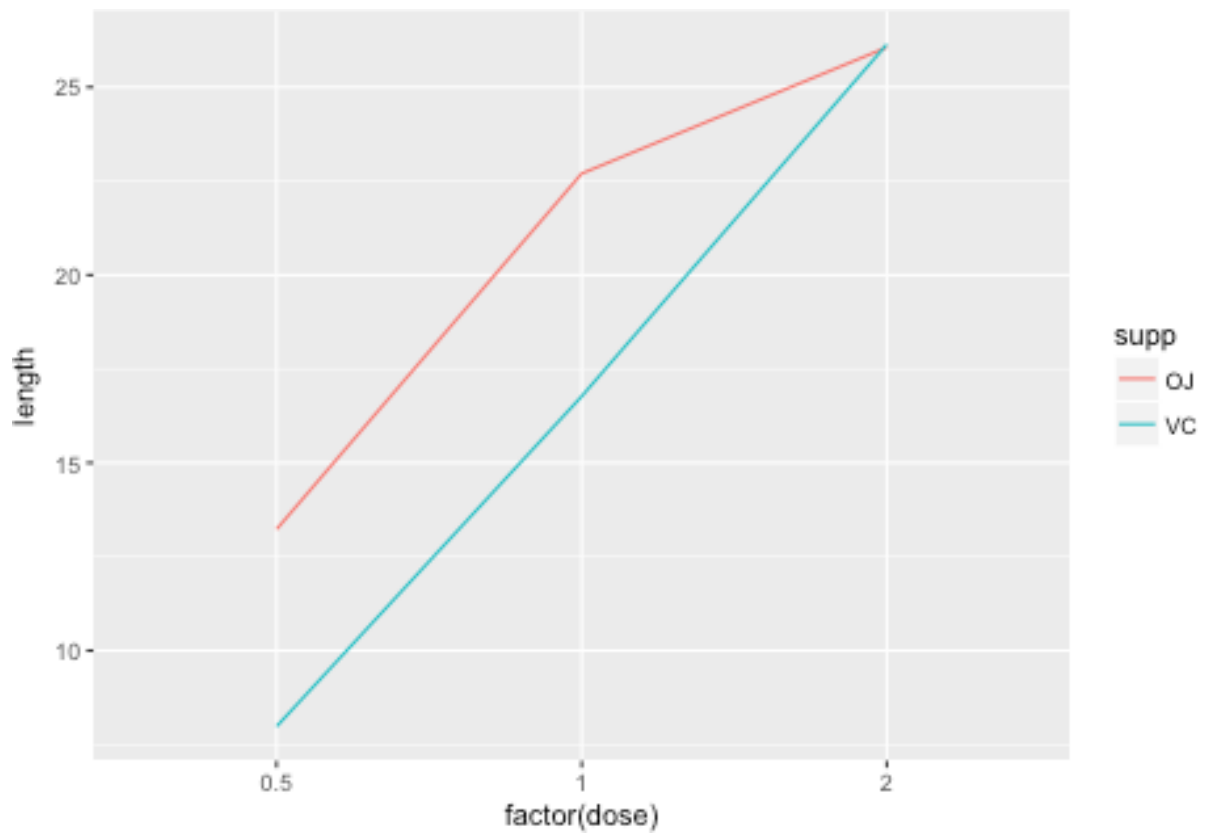
```
tg <- ddply(ToothGrowth, c("supp", "dose"), summarize, length = mean(len))
```

색상으로 구분

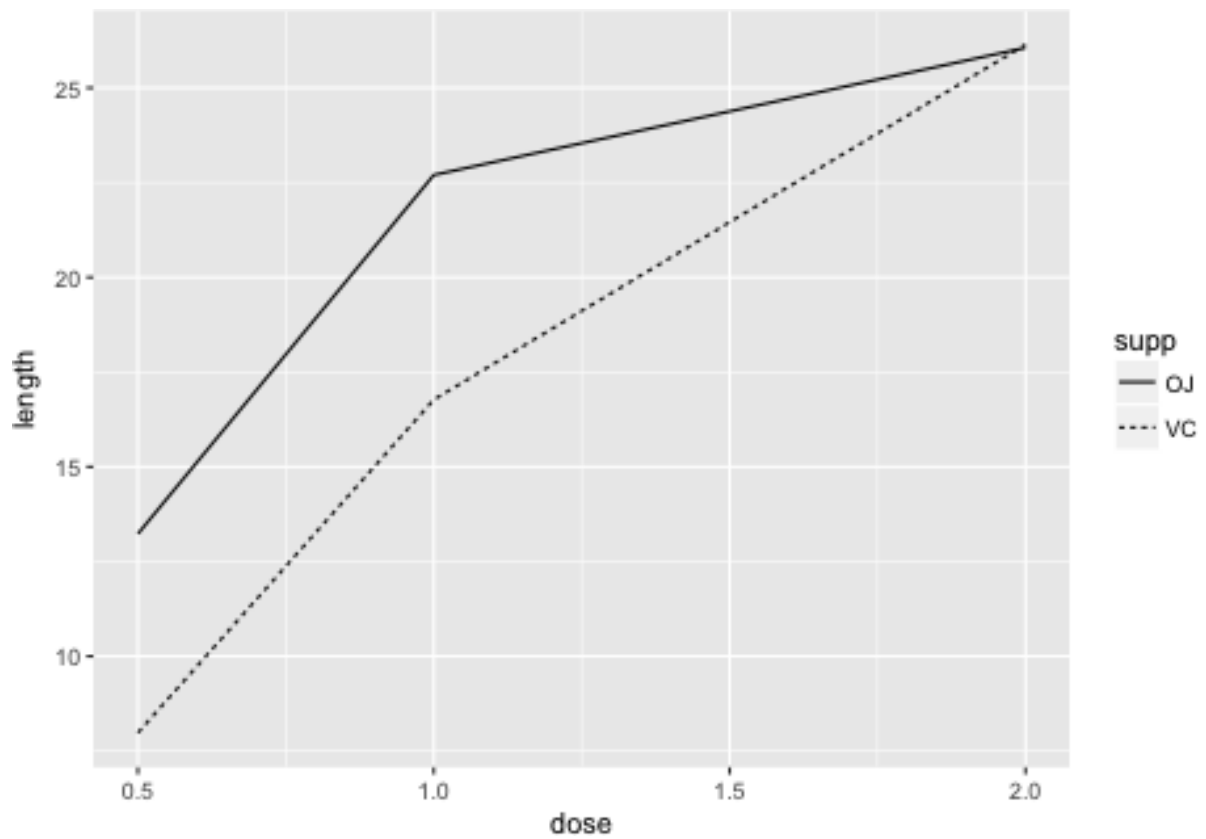
```
ggplot(tg, aes(x = dose, y = length, color = supp)) + geom_line()
```



```
ggplot(tg, aes(x = factor(dose), y = length, color = supp, group = supp)) + geom_line()
```

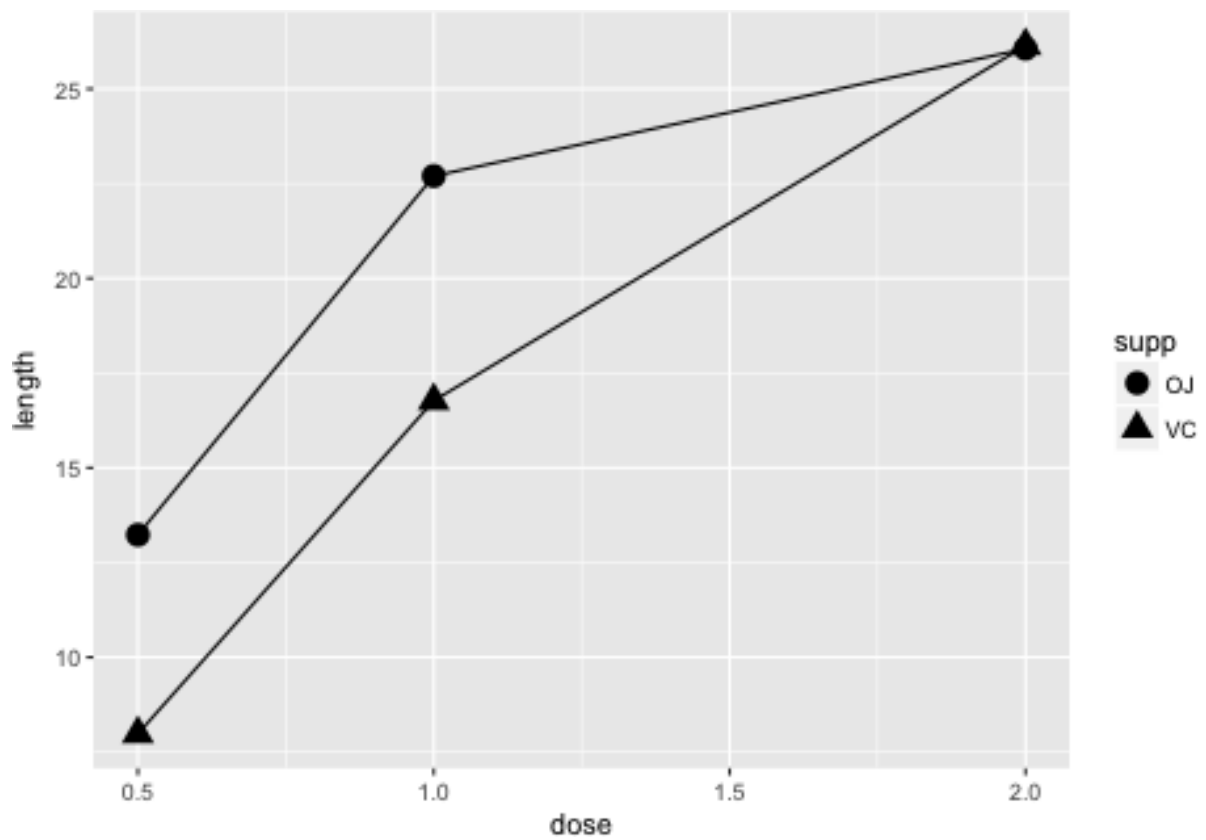


```
# group = supp 주의! : 이 명령문이 없으면 데이터를 어떻게 묶어서 그릴지 모름  
# Line type으로 구분  
ggplot(tg, aes(x = dose, y = length, linetype = supp)) + geom_line()
```

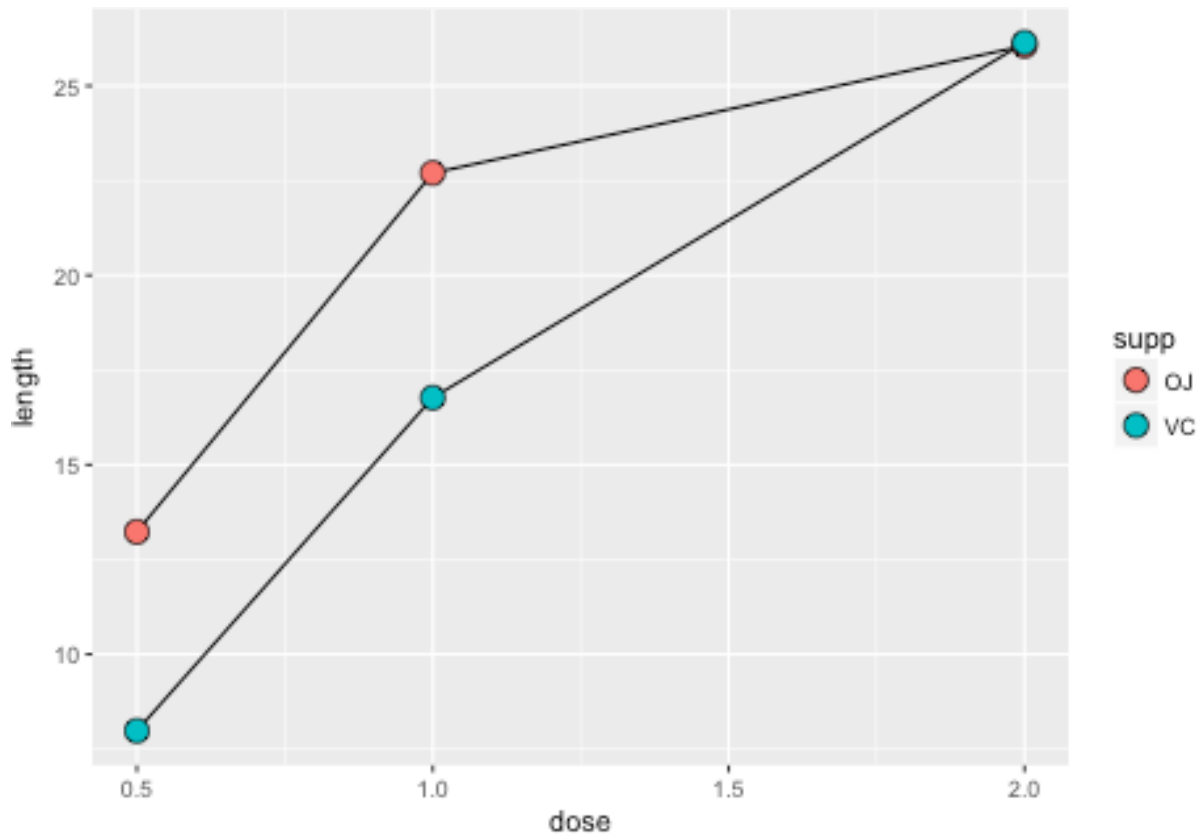
점 형태로 구분

```
ggplot(tg, aes(x = dose, y = length, shape = supp)) + geom_line() + geom_point(size = 4)
```



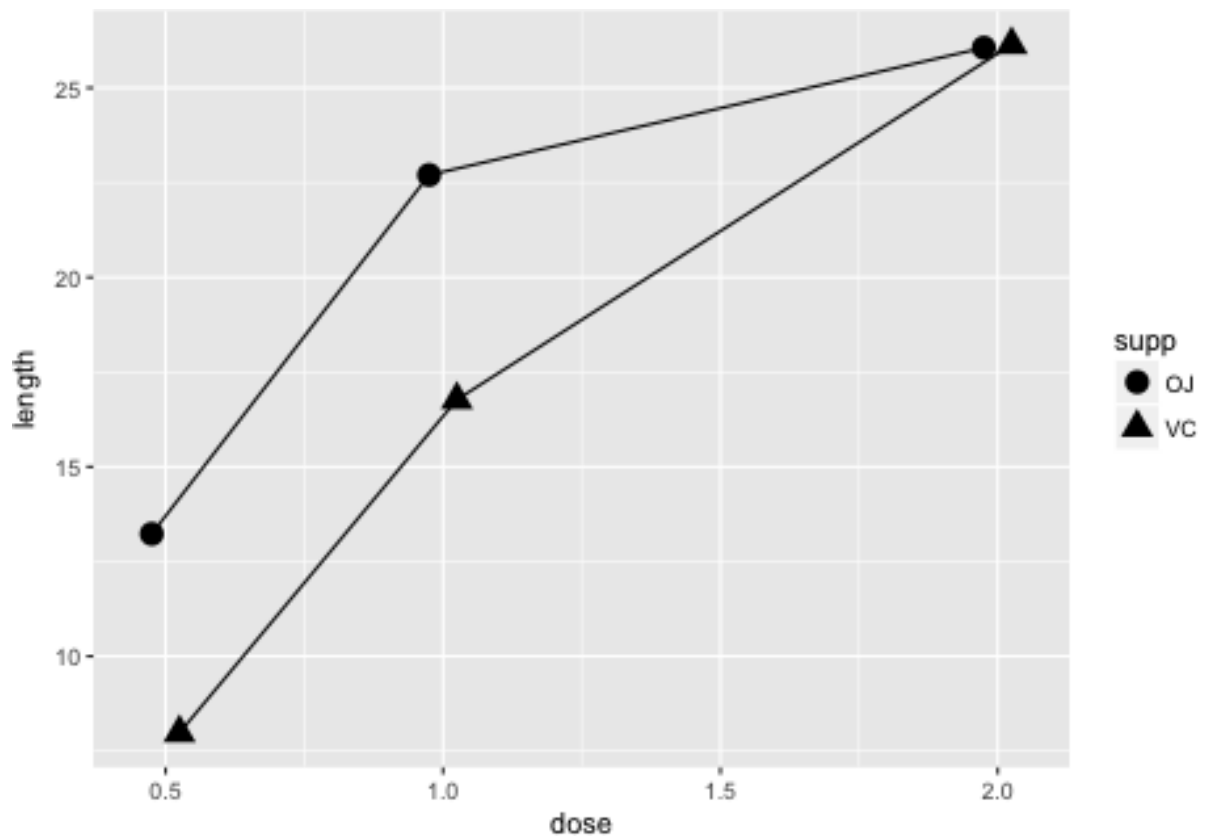
점 색상으로 구분

```
ggplot(tg, aes(x = dose, y = length, fill = supp)) + geom_line() + geom_point(size = 4, shape =
```

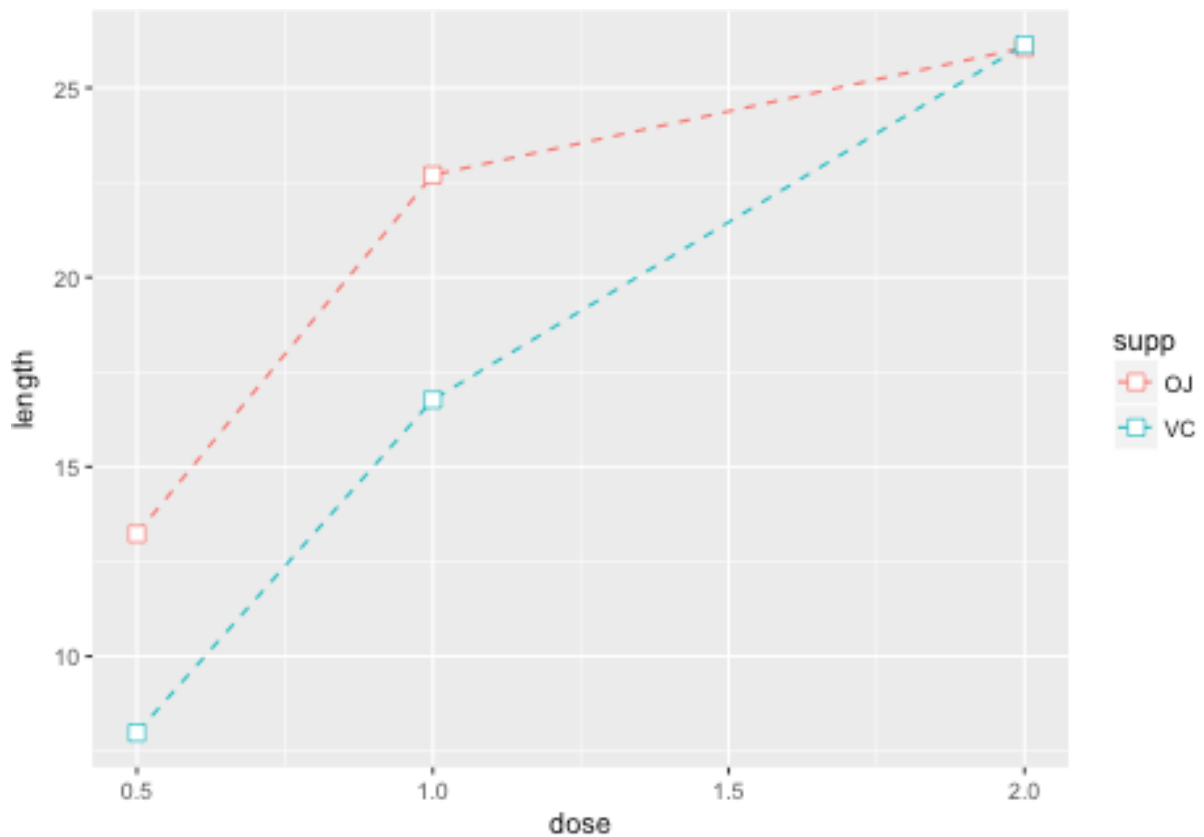


두 선이 겹칠때 하나의 선을 옆으로 이동시켜 표현

```
ggplot(tg, aes(x = dose, y = length, shape = supp)) + geom_line(position = position_dodge(0.1)) +  
  geom_point(position = position_dodge(0.1), size = 4)
```



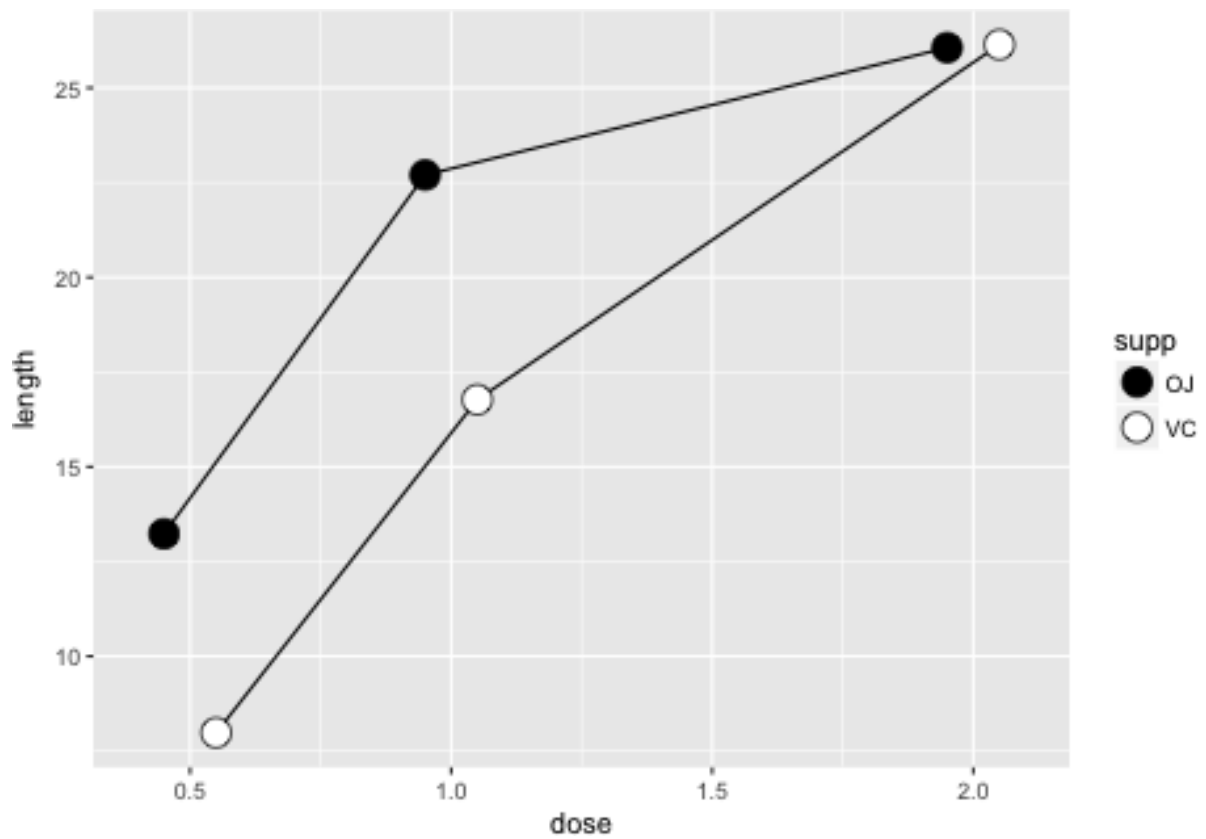
```
# 선 형태 바꾸기 [ linetype ]  
ggplot(tg, aes(x = dose, y = length, color = supp)) + geom_line(linetype = "dashed") +  
  geom_point(shape = 22, size = 3, fill = "white")
```



5.4 Part3

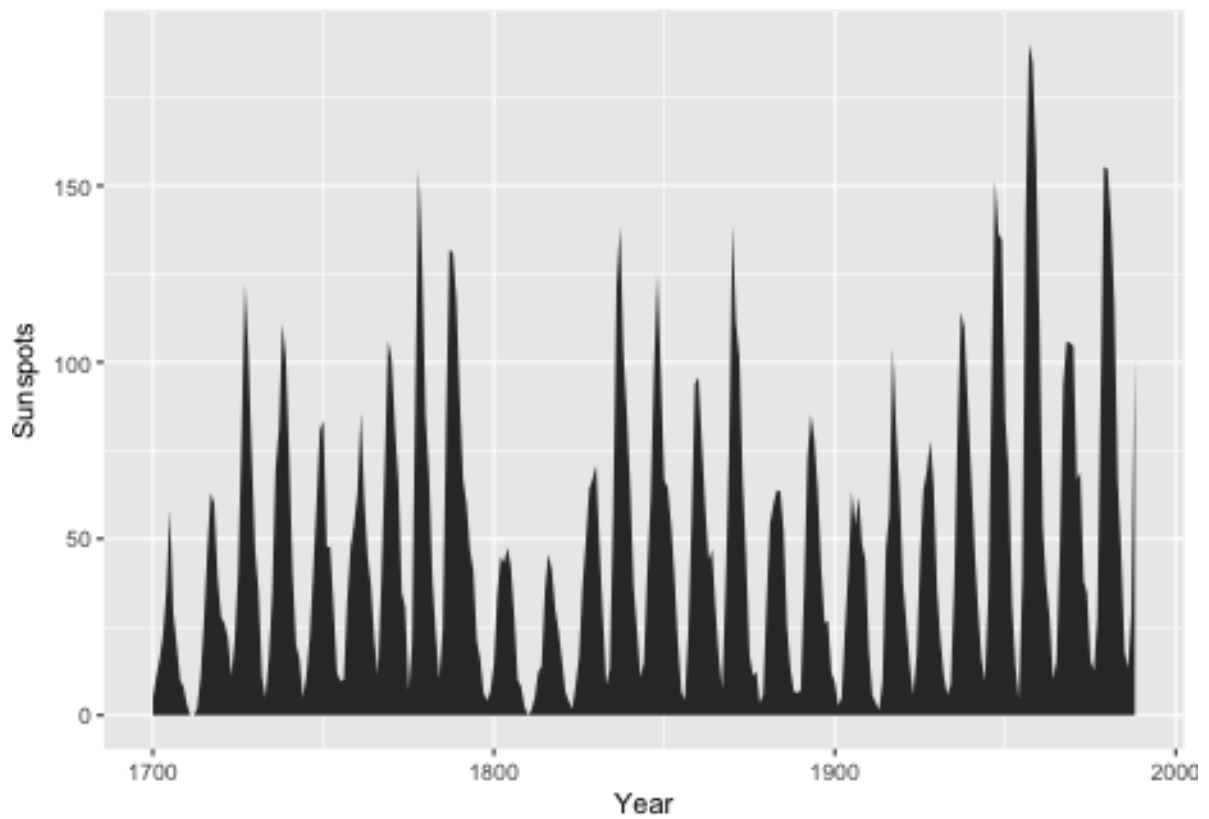
```
# 점 형태 바꾸기 [ shape ]
#####
pd <- position_dodge(0.2)      #
#####

ggplot(tg, aes(x = dose, y = length, fill = supp)) + geom_line(position = pd) +
  geom_point(shape = 21, size = 5, position = pd) +
  scale_fill_manual(values = c("black", "white"))
```



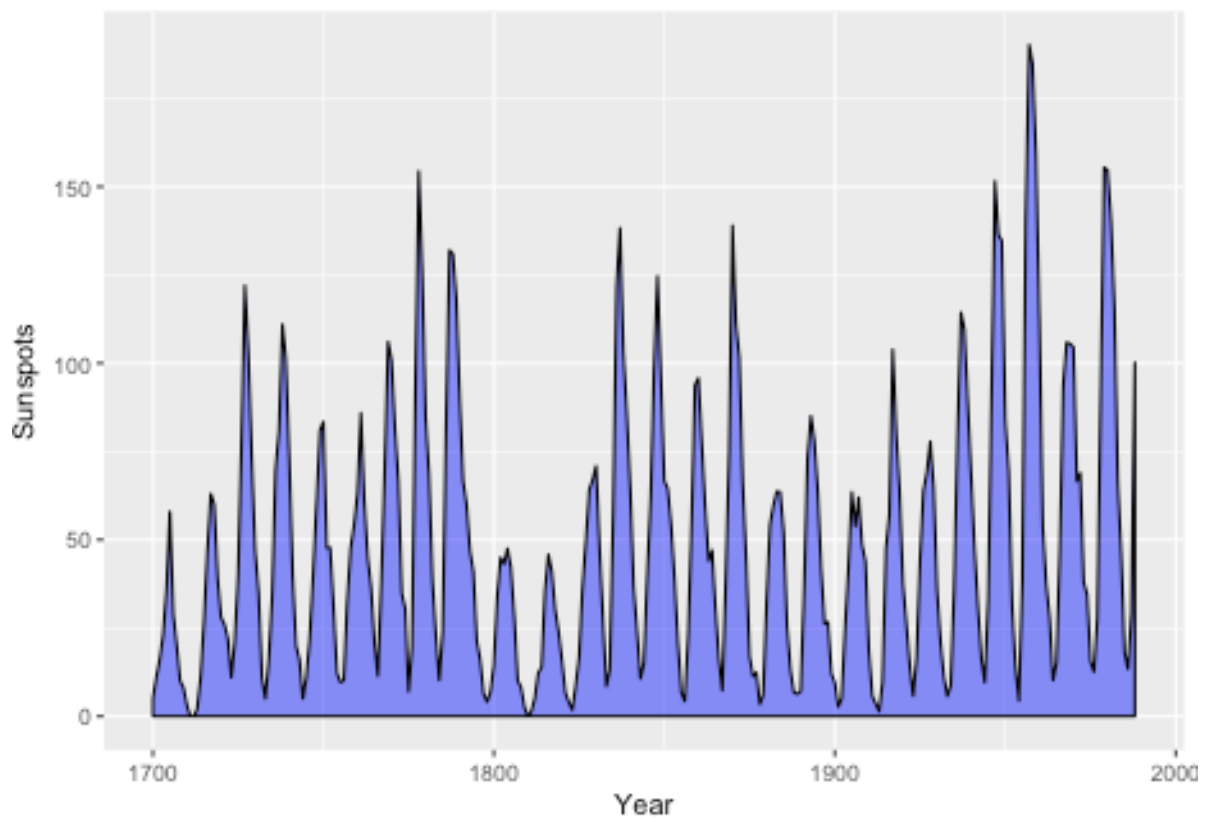
```
# sample Data
sunspotyear <- data.frame(Year      = as.numeric(time(sunspot.year)),
                          Sunspots = as.numeric(sunspot.year))

# 음영 영역 그래프 그리기
ggplot(sunspotyear, aes(x = Year, y = Sunspots)) + geom_area()
```



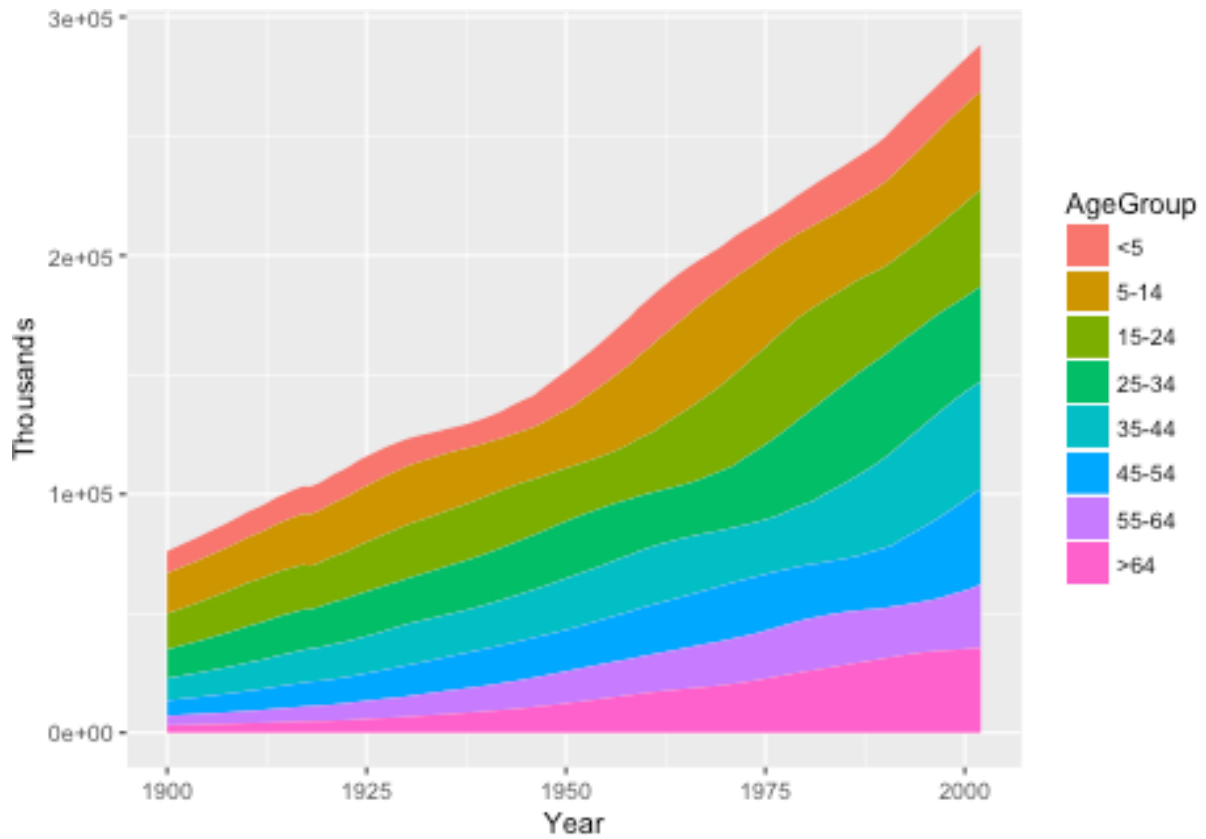
음영 투명도 설정하기 [alpha]

```
ggplot(sunspotyear, aes(x = Year, y = Sunspots)) + geom_area(color = "black", fill = "blue", alpha = 0.5)
```



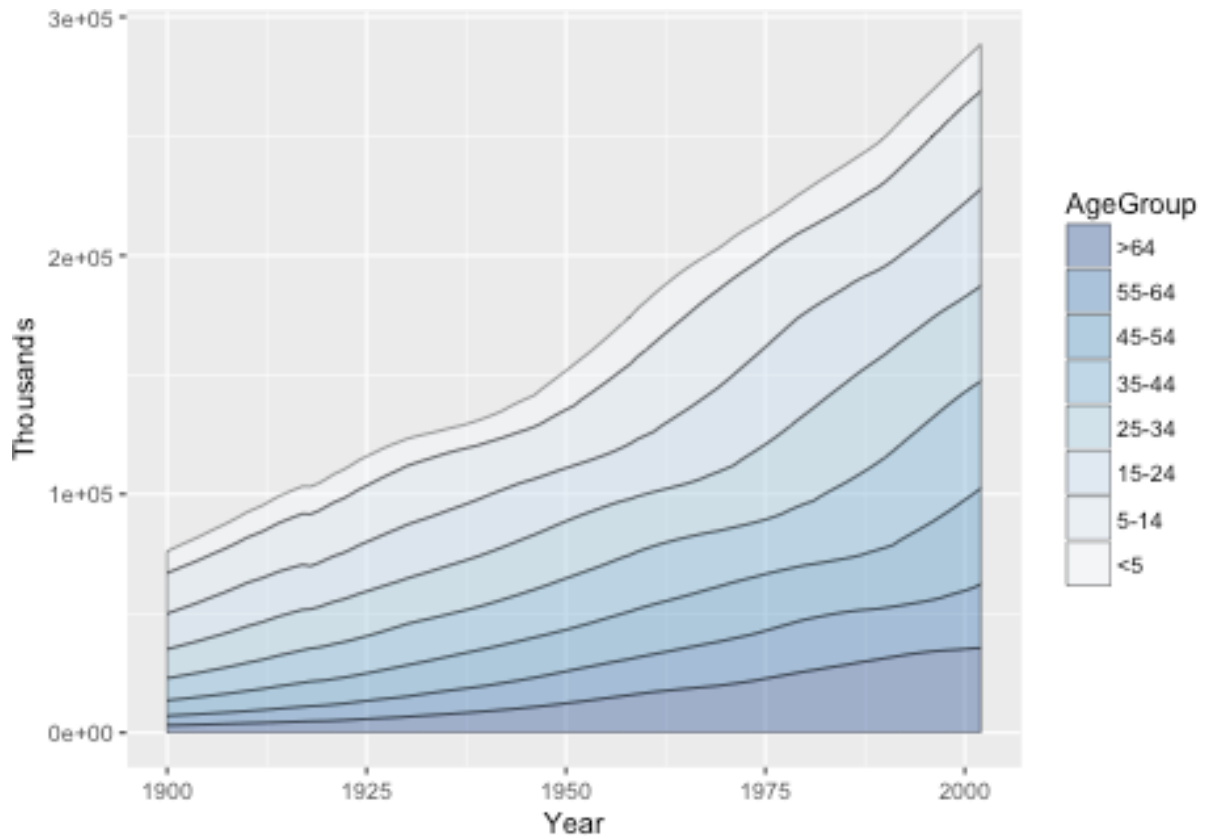
누적 영역 그래프 그리기

```
ggplot(uspopage, aes(x = Year, y = Thousands, fill = AgeGroup)) + geom_area()
```



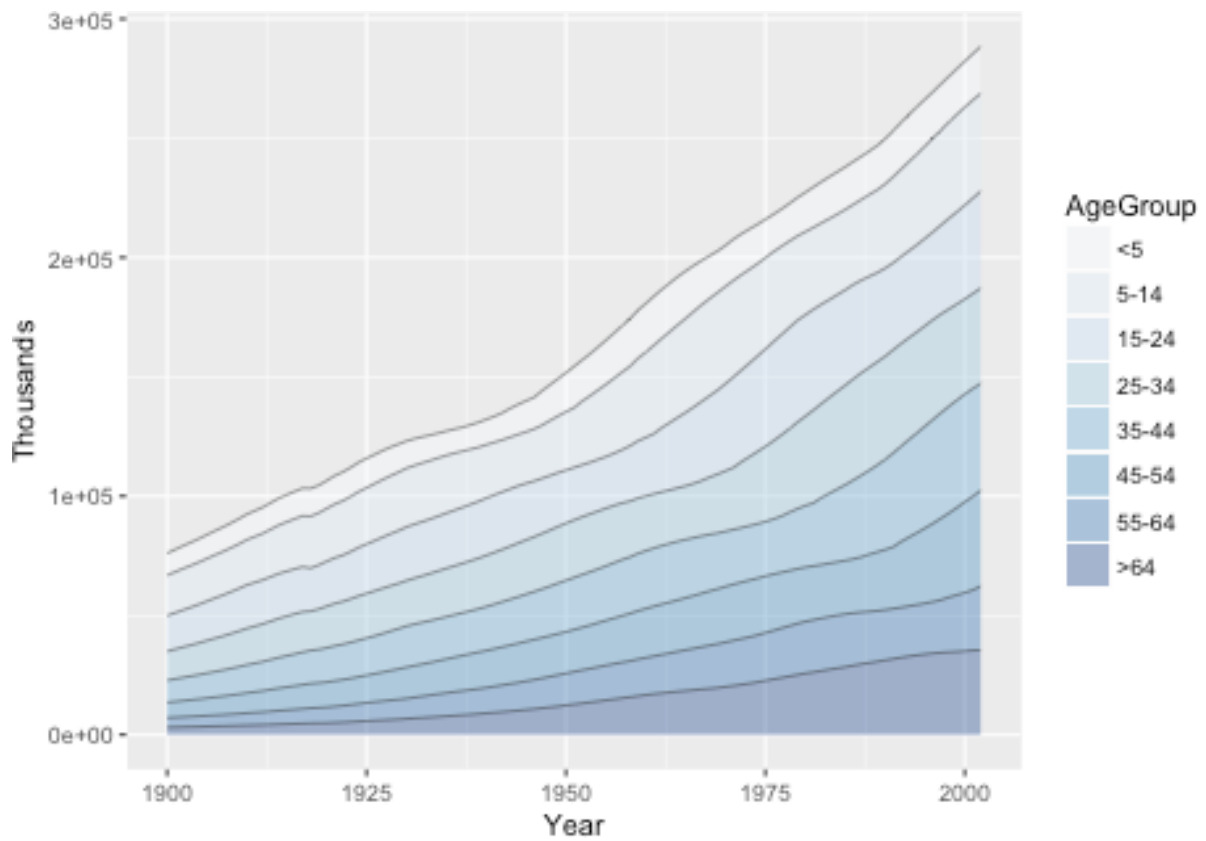
영역색상 그라데이션 넣기

```
ggplot(uspopage, aes(x = Year, y = Thousands, fill = AgeGroup)) + geom_area(color = "black", stroke = "black", size = 1) +  
  scale_fill_brewer(palette = "Blues", breaks = rev(levels(uspopage$AgeGroup)))
```



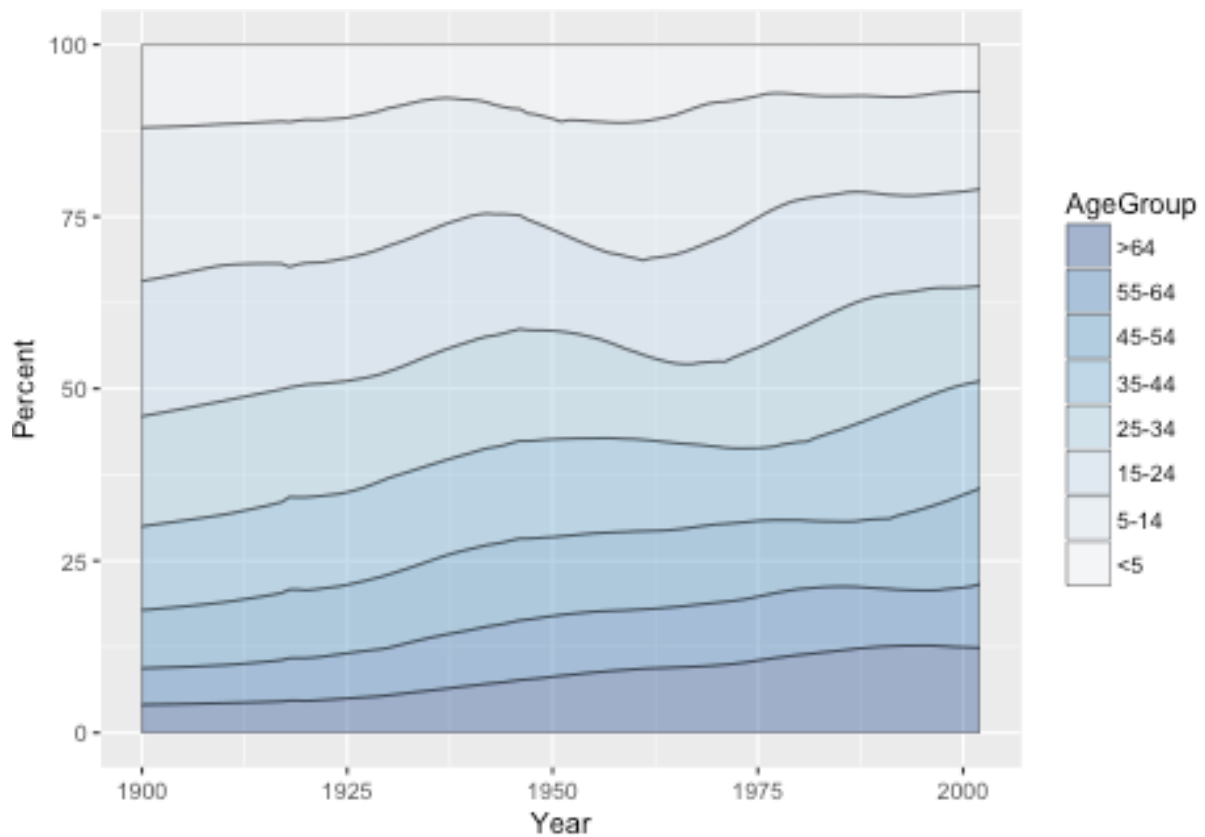
데이터 순서정렬하기 & 양쪽 테두리 지우기

```
ggplot(uspope, aes(x = Year, y = Thousands, fill = AgeGroup, order = desc(AgeGroup))) +  
  geom_area(color = NA, alpha = 0.4) + scale_fill_brewer(palette = "Blues") + geom_line(position = "dodge")
```

비율 누적 영역 그래프 그리기

```
uspopage_prop <- ddply(uspopage, "Year", transform, Percent = Thousands / sum(Thousands) * 100)
ggplot(uspopage_prop, aes(x = Year, y = Percent, fill = AgeGroup)) + geom_area(color = "black",
  scale_fill_brewer(palette = "Blues", breaks = rev(levels(uspopage$AgeGroup)))
```

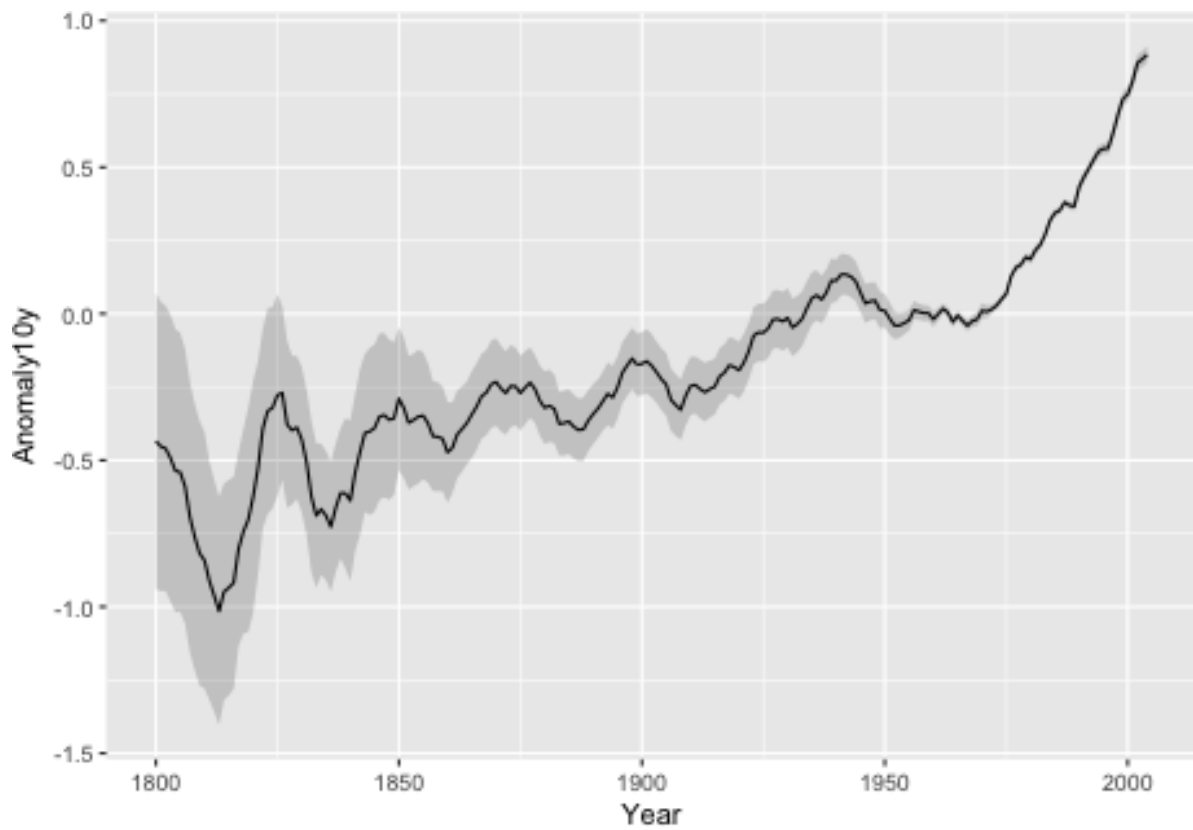


그래프에 신뢰 영역 추가하기

```
clim <- subset(climate, Source == "Berkeley", select = c("Year", "Anomaly10y", "Unc10y"))
```

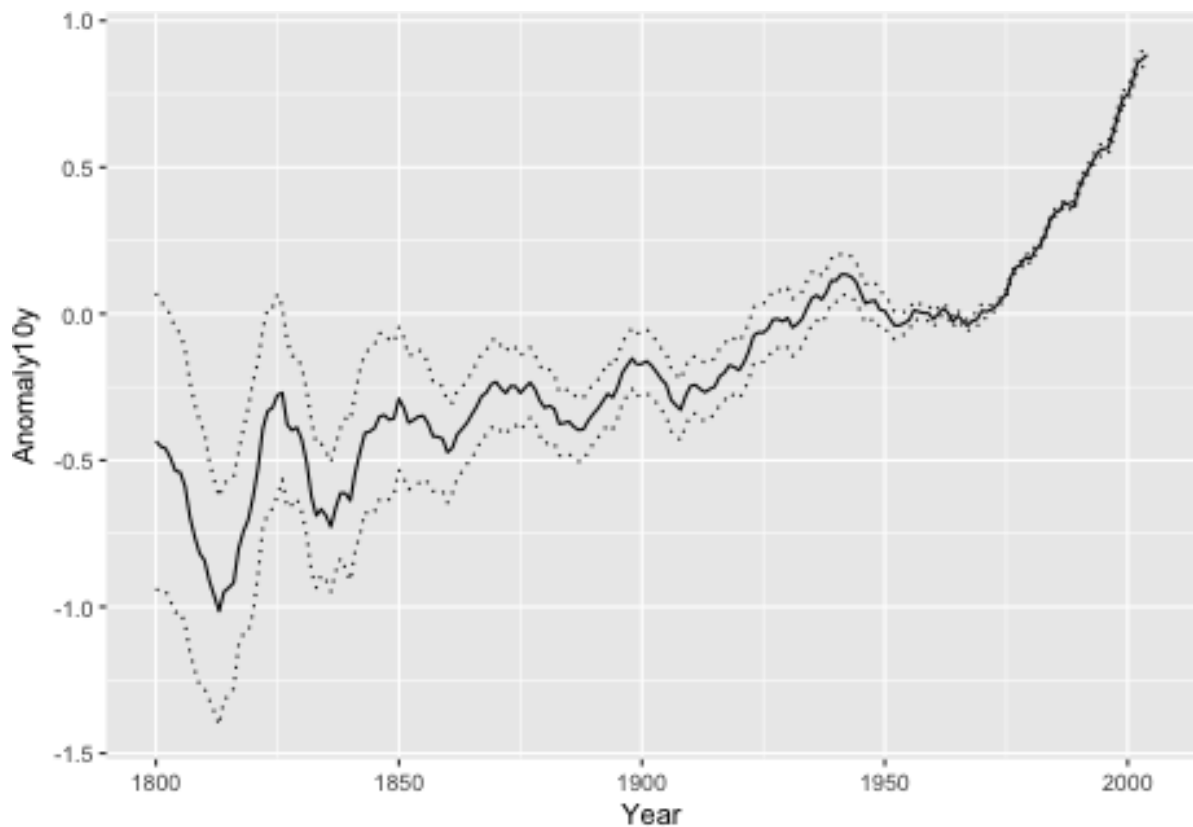
신뢰영역 음영으로 표현

```
ggplot(clim, aes(x = Year, y = Anomaly10y)) +  
  geom_ribbon(aes(ymin = Anomaly10y - Unc10y, ymax = Anomaly10y + Unc10y), alpha = 0.2) +  
  geom_line()
```

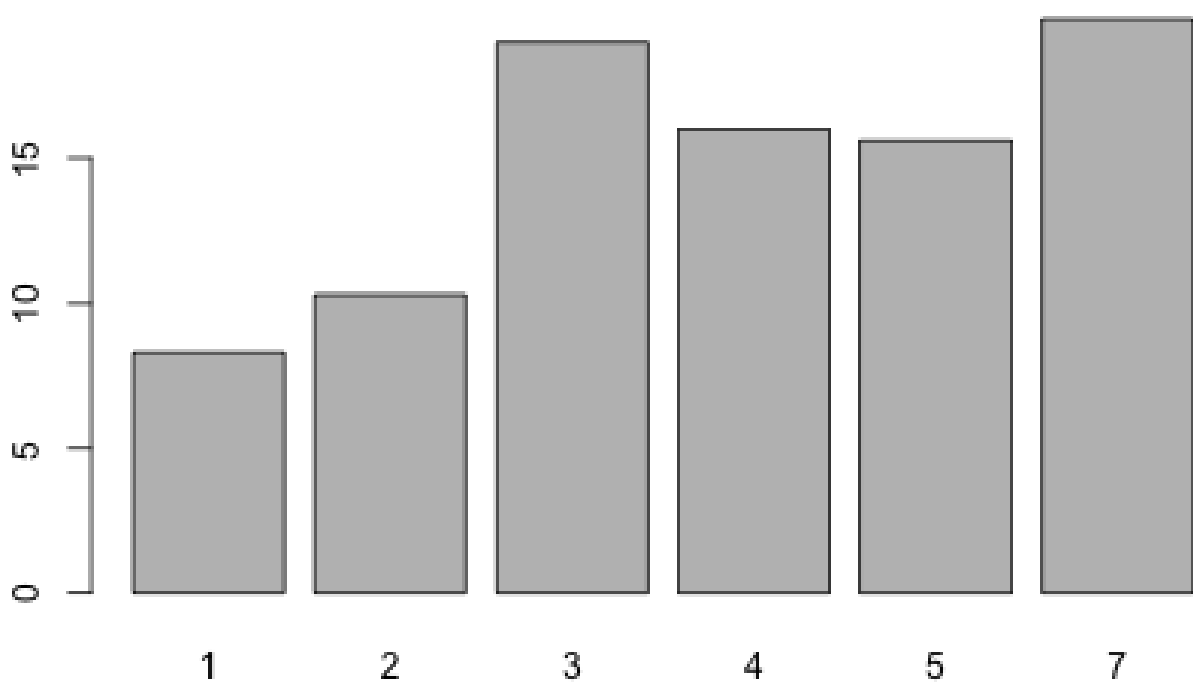


신뢰영역 점선으로 표현

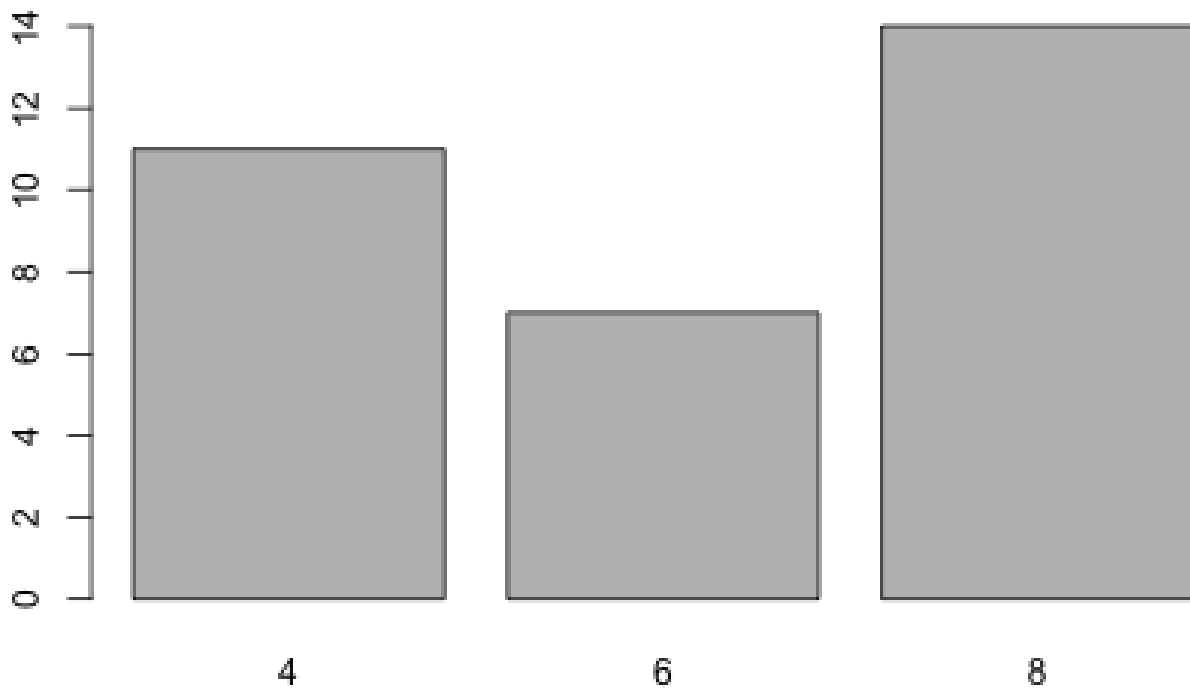
```
ggplot(clim, aes(x = Year, y = Anomaly10y)) +  
  geom_line(aes(y = Anomaly10y - Unc10y), linetype = "dotted") +  
  geom_line(aes(y = Anomaly10y + Unc10y), linetype = "dotted") +  
  geom_line()
```



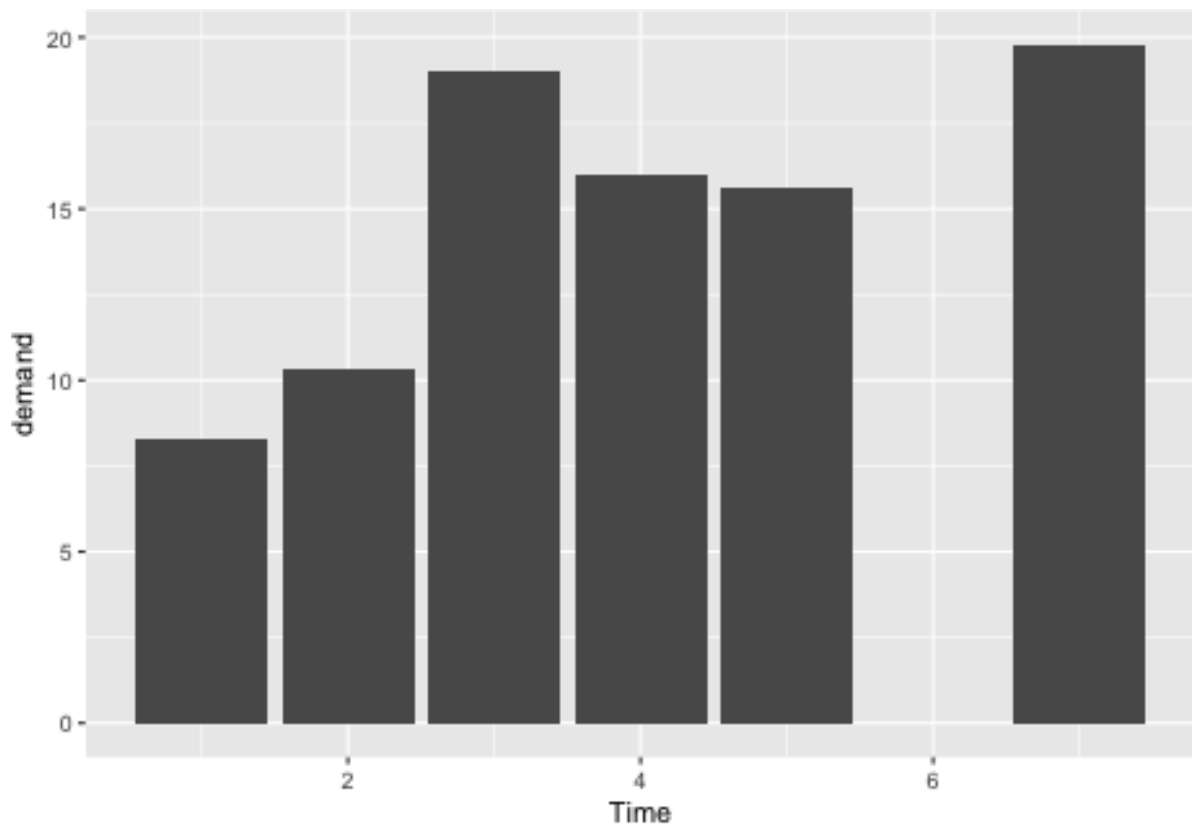
```
#
#####
## barplot (geom_bar)
#####
barplot(BOD$demand, names.arg = BOD$Time)
```



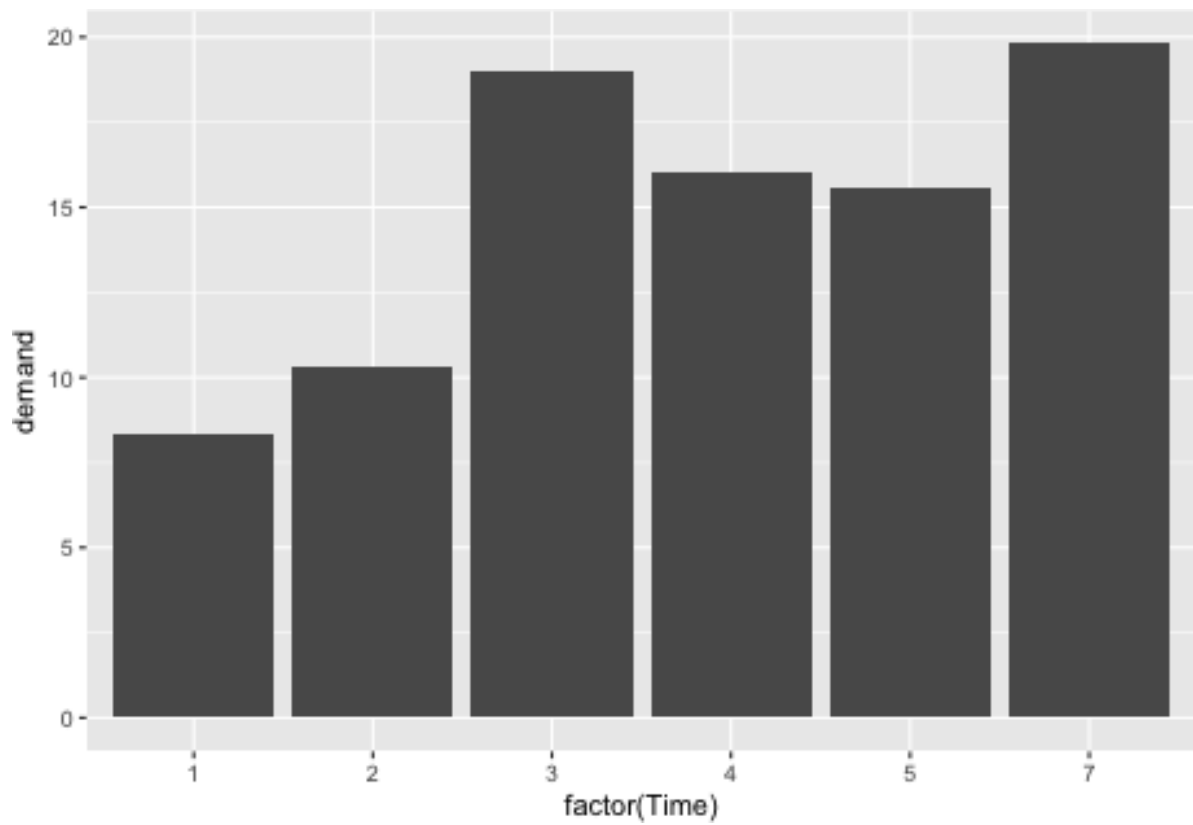
```
barplot(table(mtcars$cyl))
```



```
# x값을 숫자로 인식  
# qplot(BOD$Time, BOD$demand, geom = "bar", stat = "identity")  
ggplot(BOD, aes(x = Time, y = demand)) + geom_bar(stat = "identity")
```

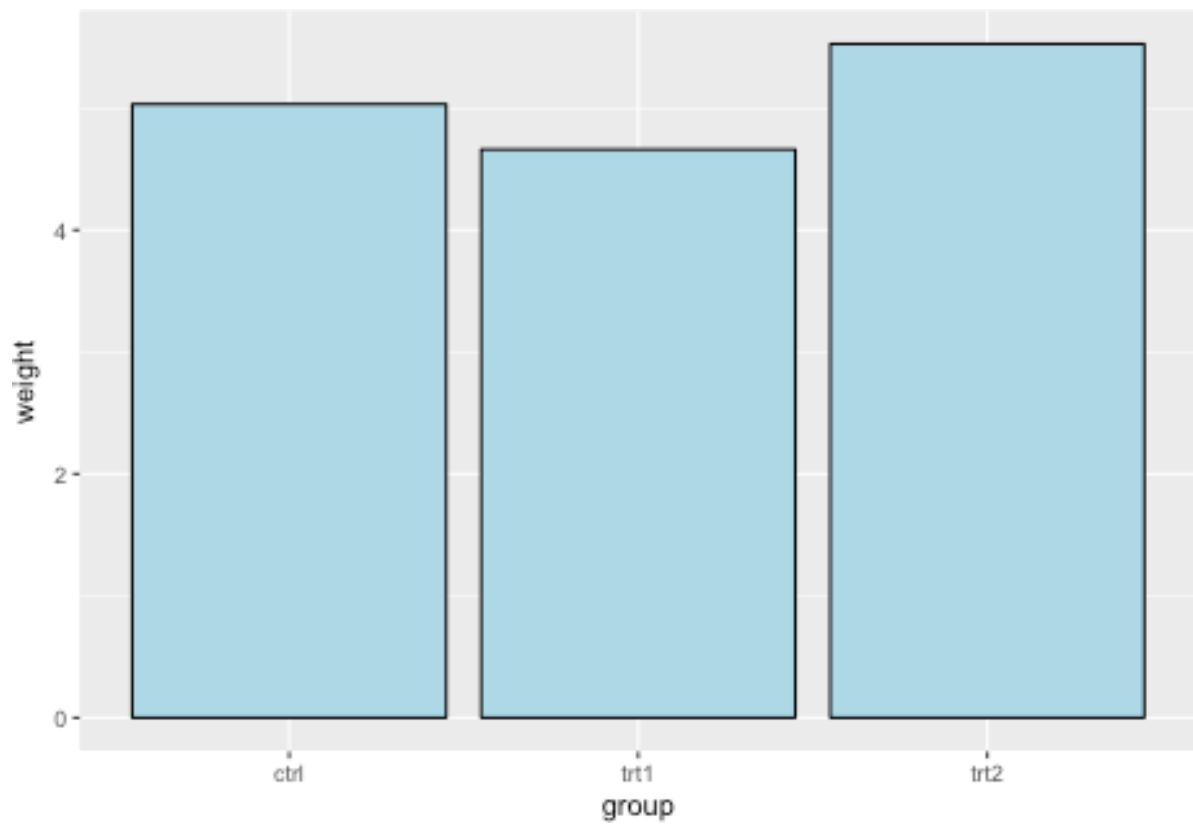


```
# x값을 요인으로 변환  
# qplot(as.factor(BOD$Time), BOD$demand, geom = "bar", stat = "identity")  
ggplot(BOD, aes(x = factor(Time), y = demand)) + geom_bar(stat = "identity")
```



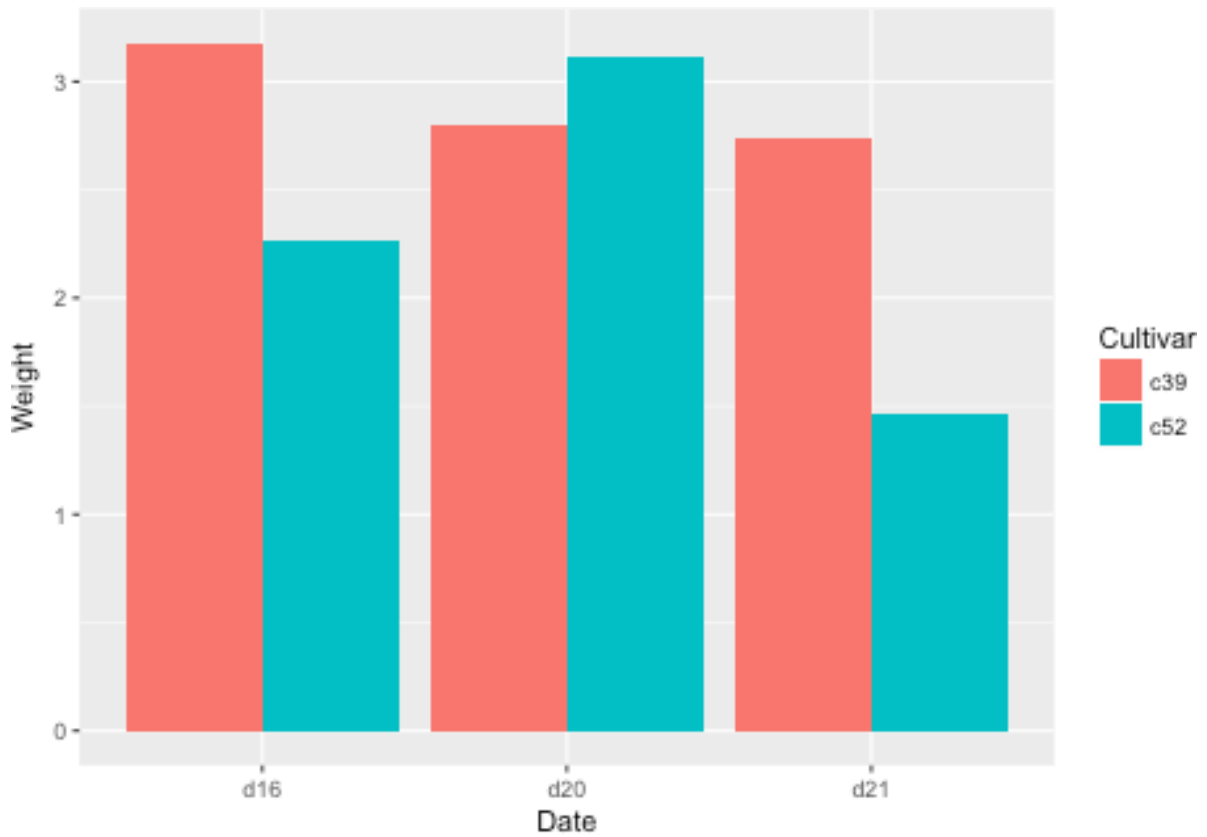
막대 색상 채우기/테두리 설정하기 (fill : 채우기/ colour(or color) : 테두리)

`ggplot(pg_mean, aes(x = group, y = weight)) + geom_bar(stat = "identity", fill = "lightblue", color = "black")`



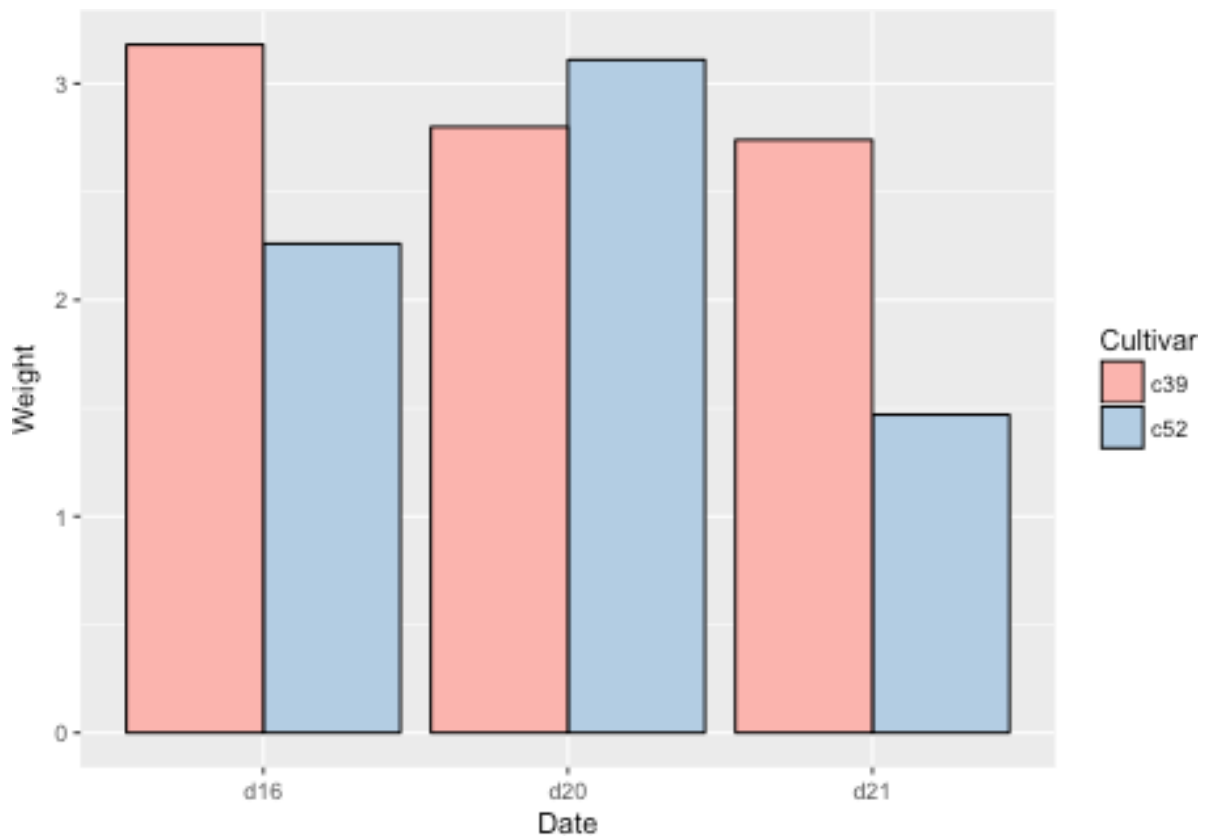
막대 묶어서 표현하기(나누어 표현하고 싶은 변수를 색상으로 지정)

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity", p
```



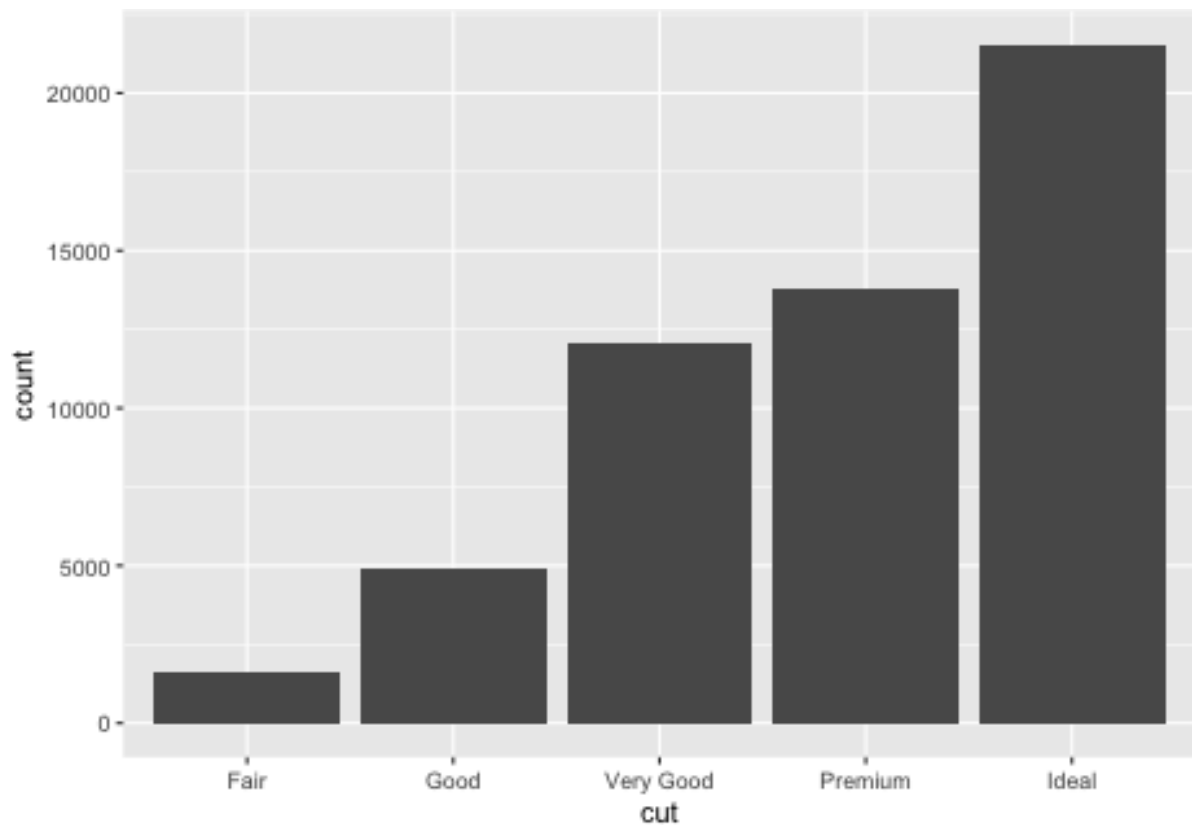
dodge : "피하다"라는 의미로 막대를 새로운 값을 기준으로 나누어 표현

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity", p
  scale_fill_brewer(palette = "Pastel1")
```

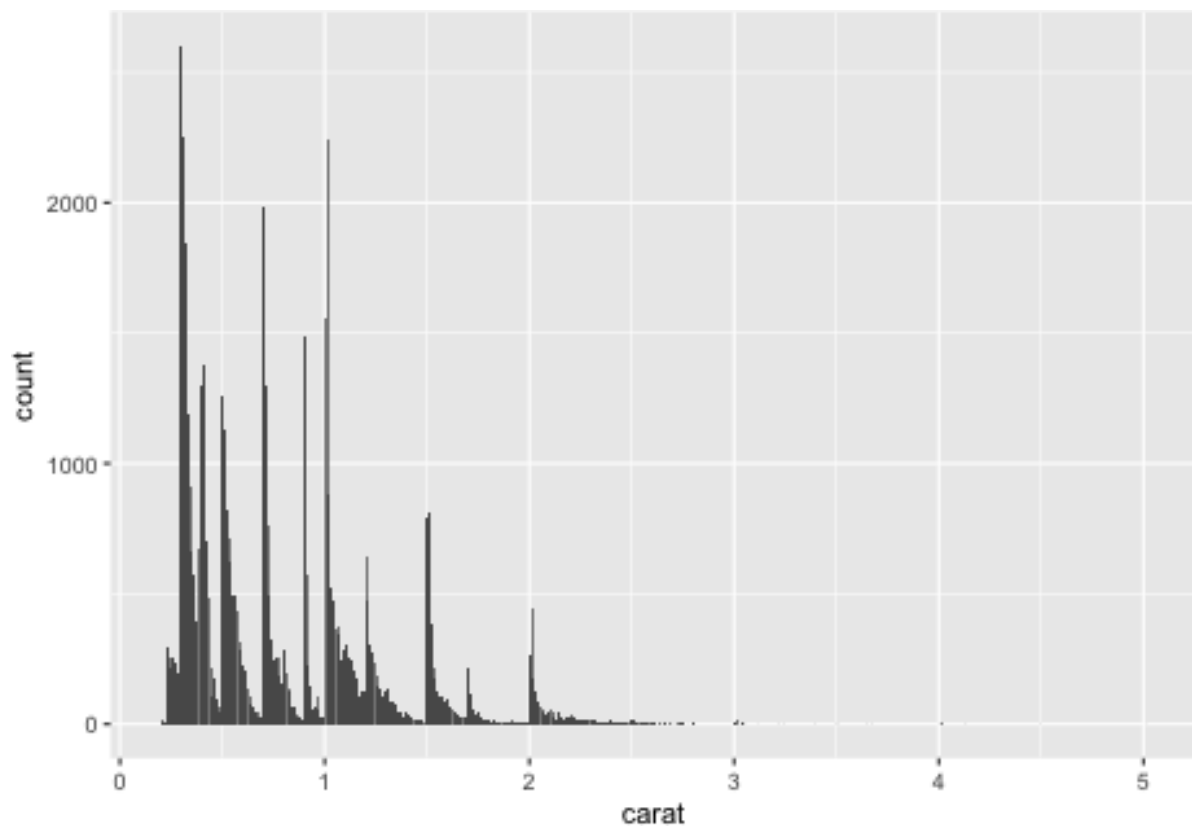
5.5 Part4

```
# 빈도수 막대 그래프 그리기  
# x가 이산형  
ggplot(diamonds, aes(x = cut)) + geom_bar()
```

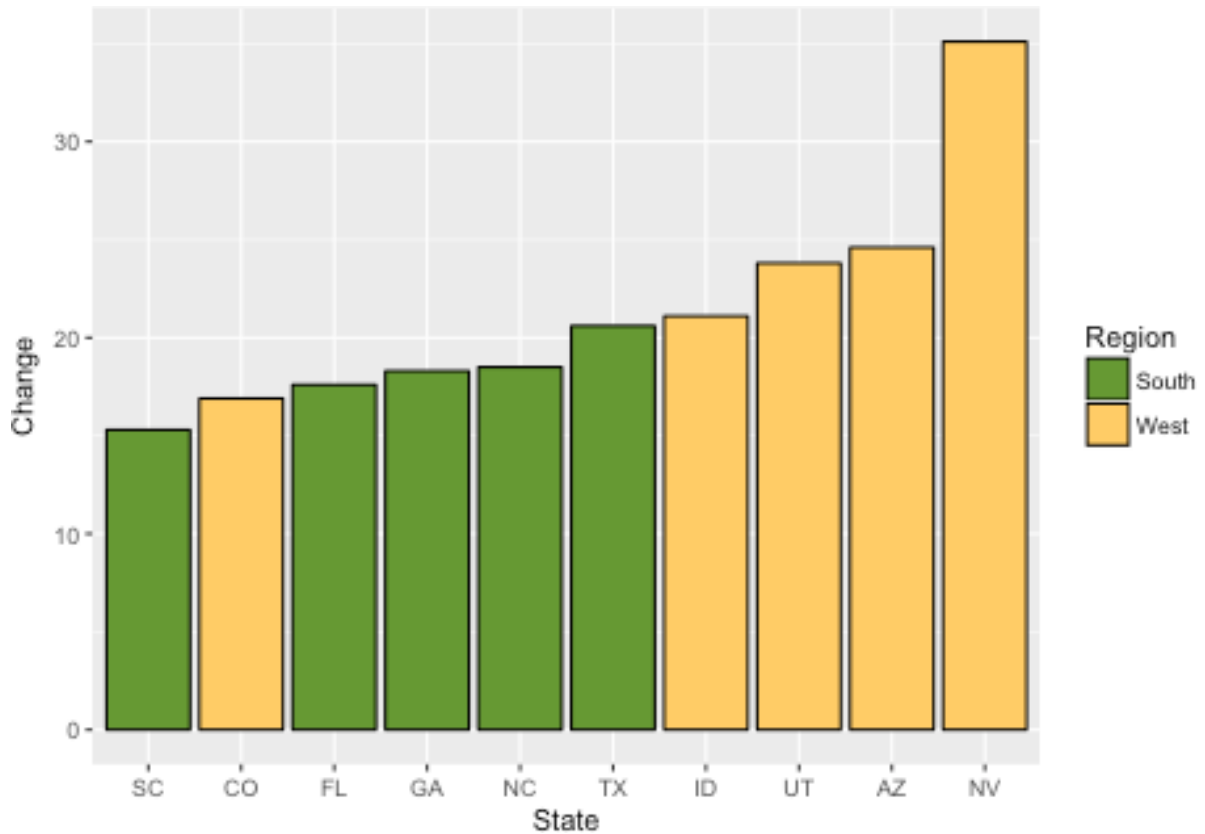


x가 연속형

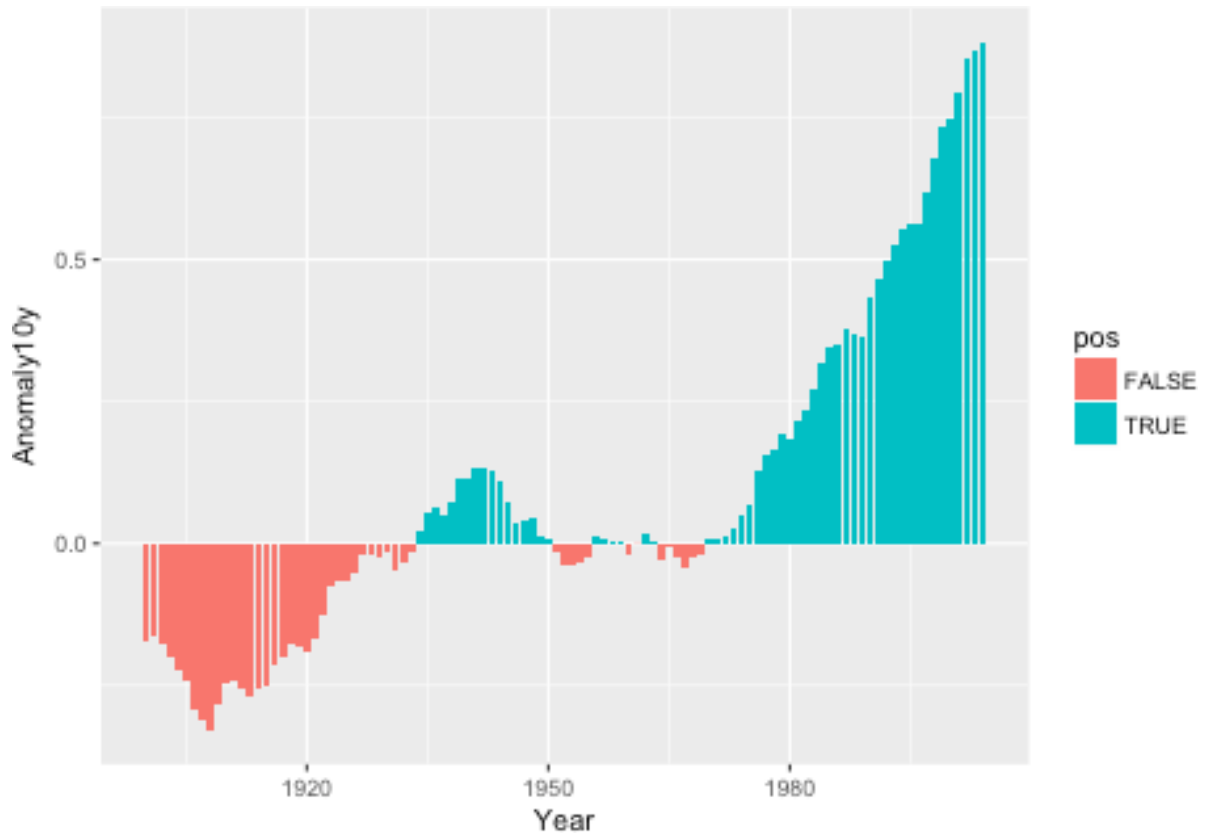
```
ggplot(diamonds, aes(x = carat)) + geom_bar()
```



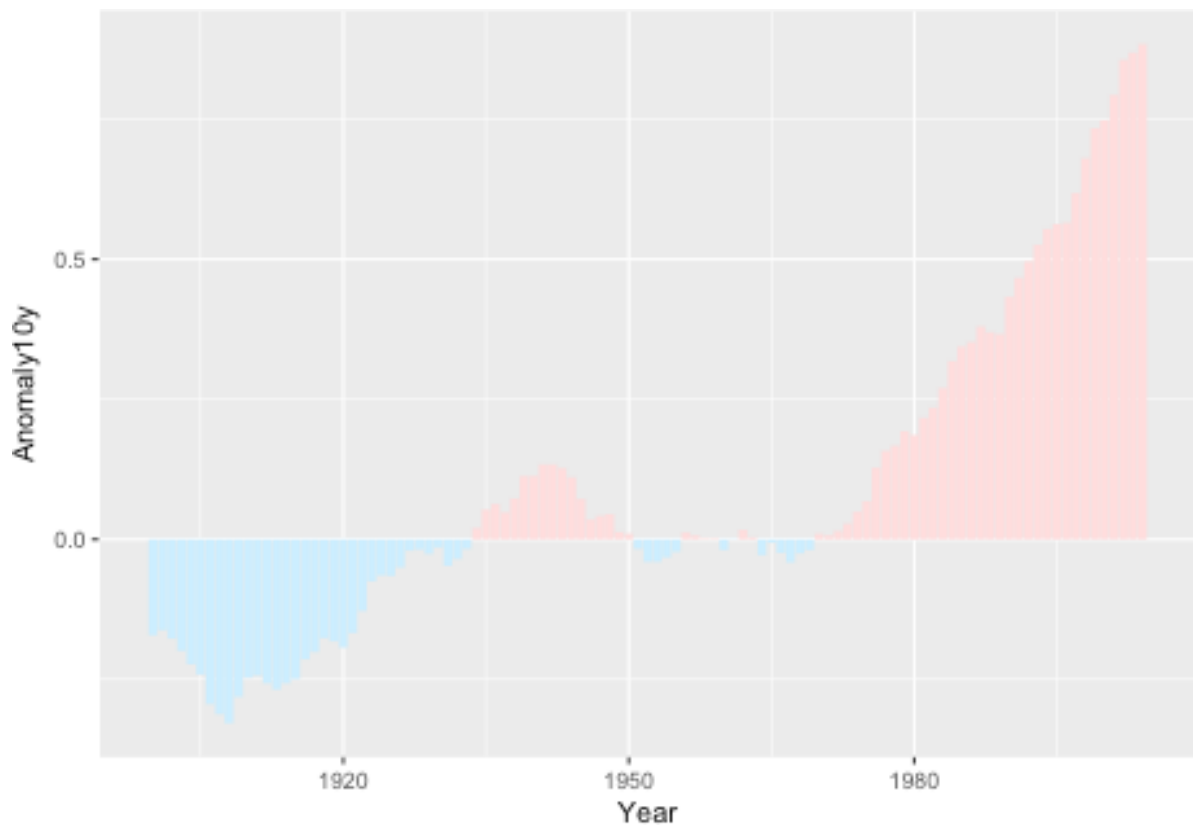
```
# 막대 색상 넣기/ 축 이름 정의하기(reorder)
upc <- subset(uspopenchange, rank(Change) > 40)
ggplot(upc, aes(x = reorder(Abb, Change), y = Change, fill = Region)) + geom_bar(stat = "identity")
scale_fill_manual(values = c("#669933", "#FFCC66")) + xlab("State")
```



```
# 양수/음수 다른 색상으로 표현 (구분 inde를 만들어 색상 변수로 지정)
csub <- subset(climate, Source == "Berkeley" & Year >= 1900 )
csub$pos <- csub$Anomaly10y >= 0
ggplot(csub, aes(x = Year, y = Anomaly10y, fill = pos)) + geom_bar(stat = "identity", position = "dodge")
```

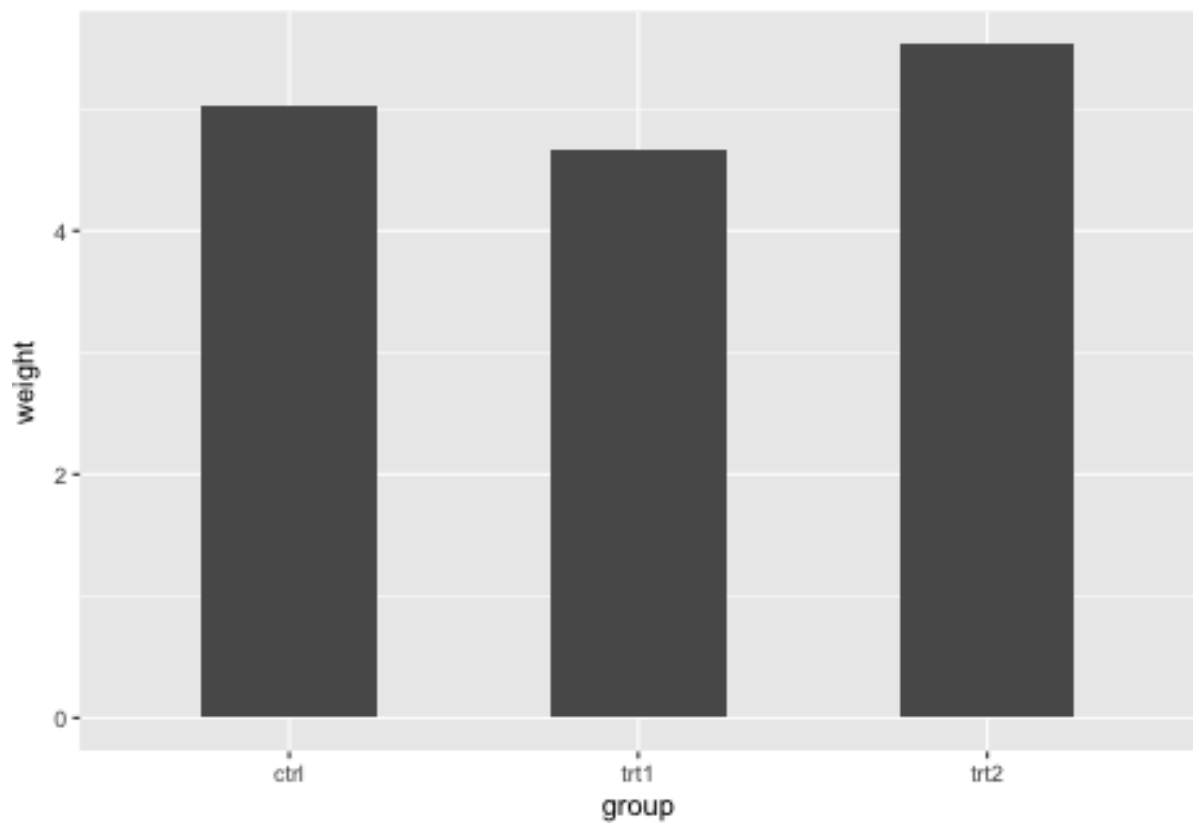


```
# 막대 그래프 테두리 두께 설정(size) / 범례 지우기(guide = FALSE)
ggplot(csub, aes(x = Year, y = Anomaly10y, fill = pos)) + geom_bar(stat = "identity", position
  scale_fill_manual(values = c("#CCEEFF", "#FFDDDD"), guide = FALSE)
```



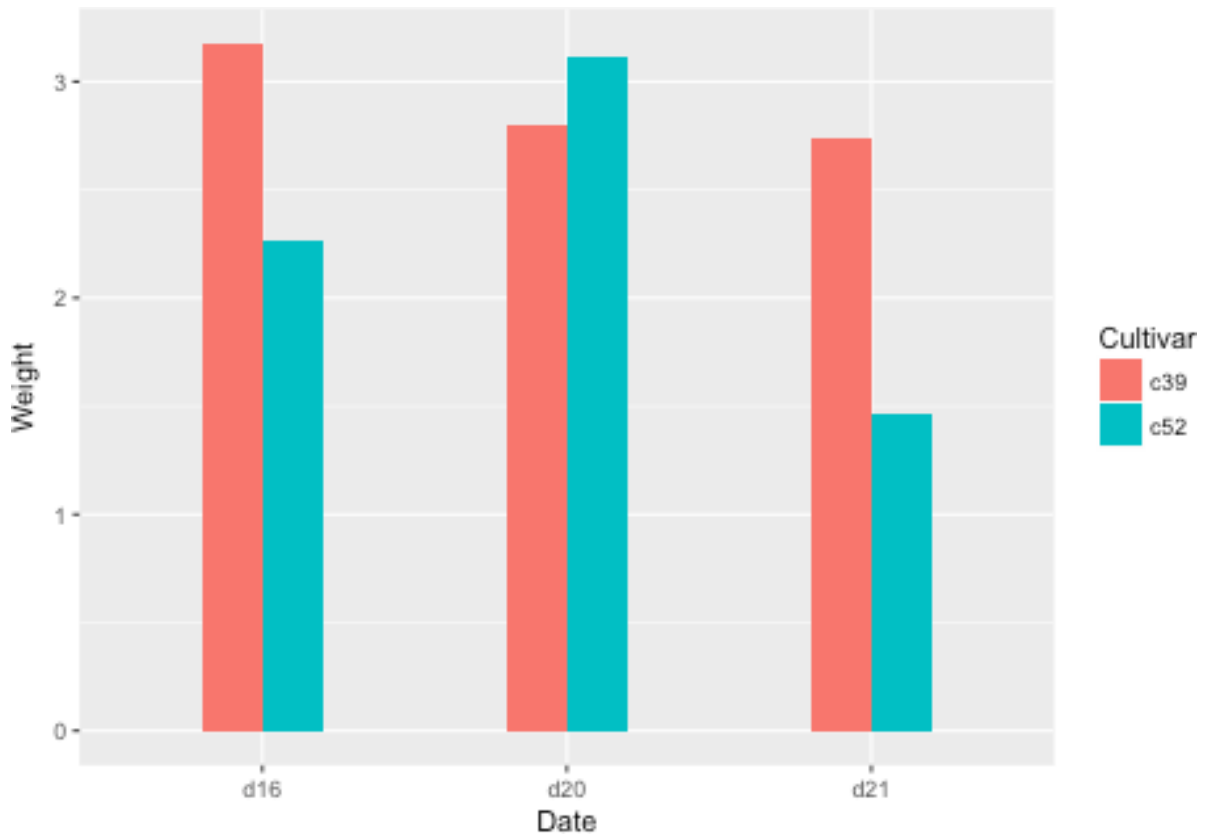
막대 너비/ 간격 조절하기(width : 최대 너비는 1)

```
ggplot(pg_mean, aes(x = group, y = weight)) + geom_bar(stat = "identity", width = 0.5)
```



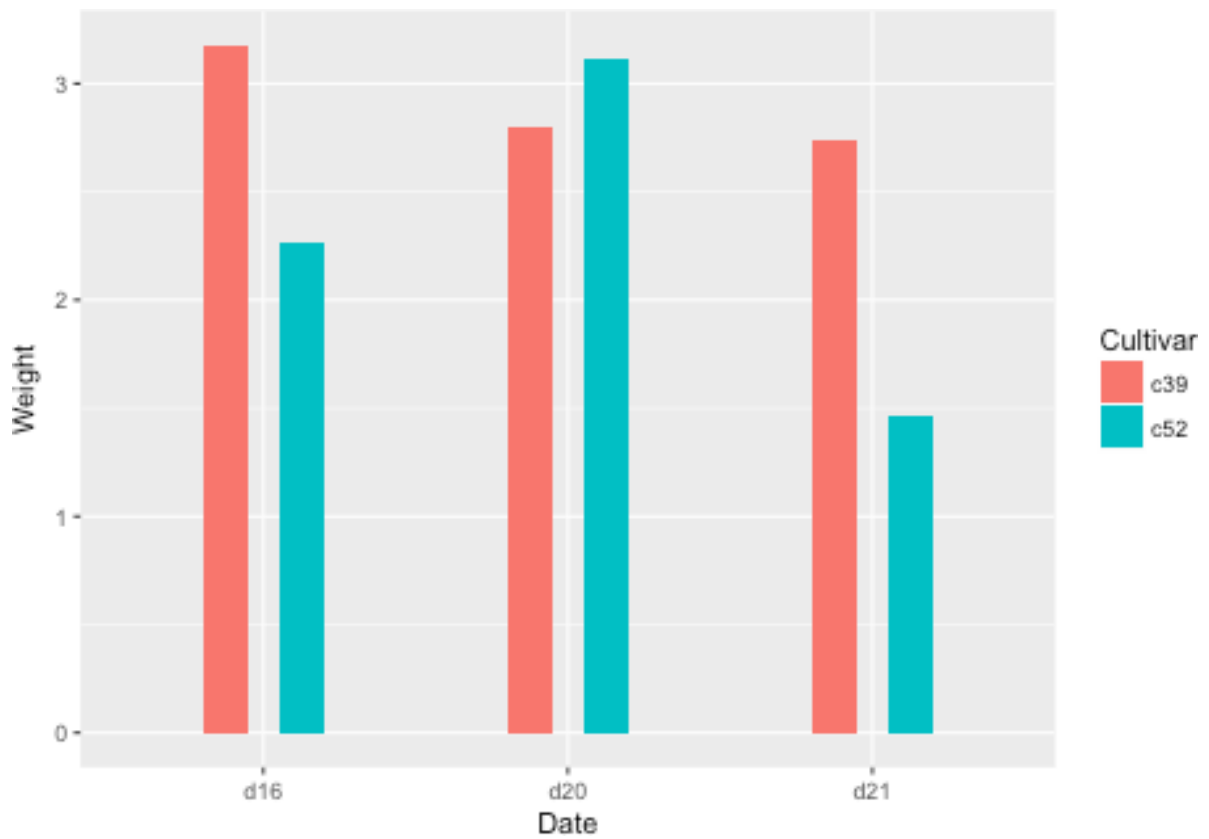
```
# 막대 그룹 간의 간격 조절하기 (default : 0.9)
```

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity", w
```



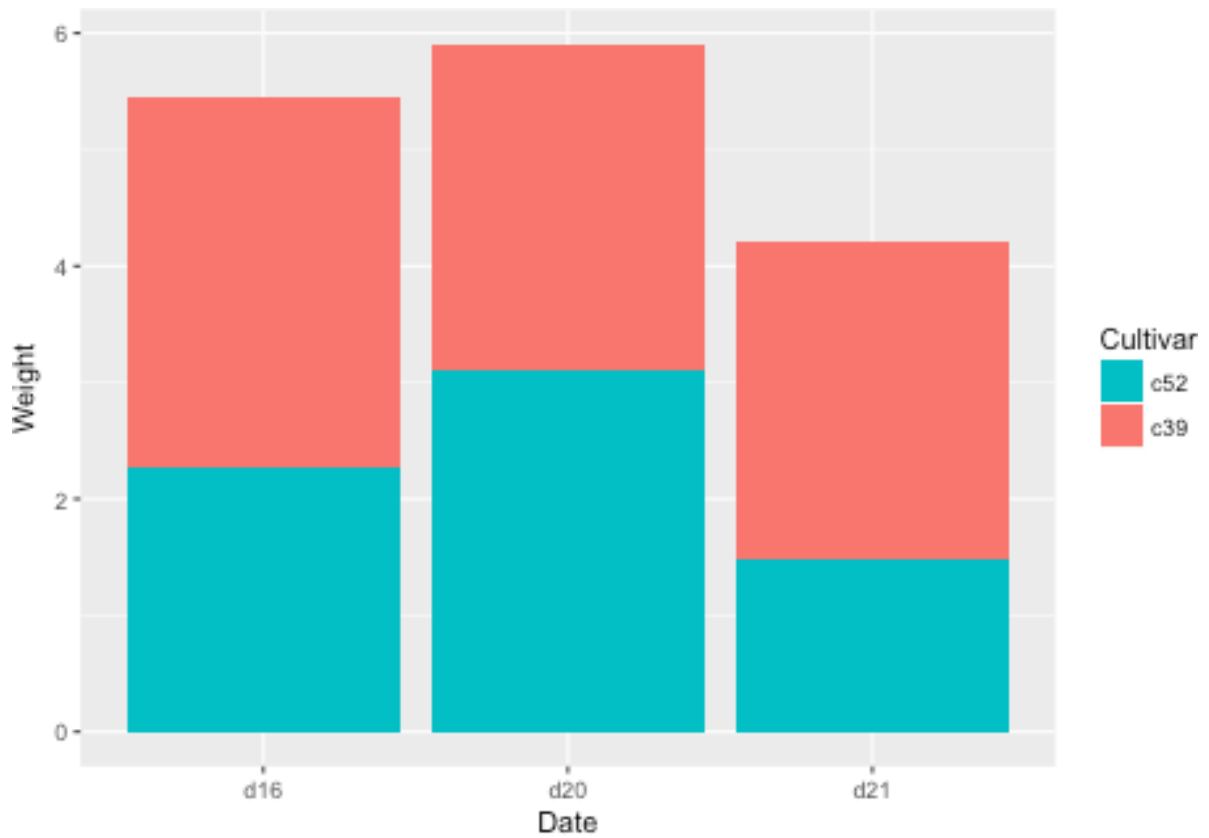
```
# 막대 그룹 내부간의 간격 조절 (default : 0.9)
```

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity", w
```



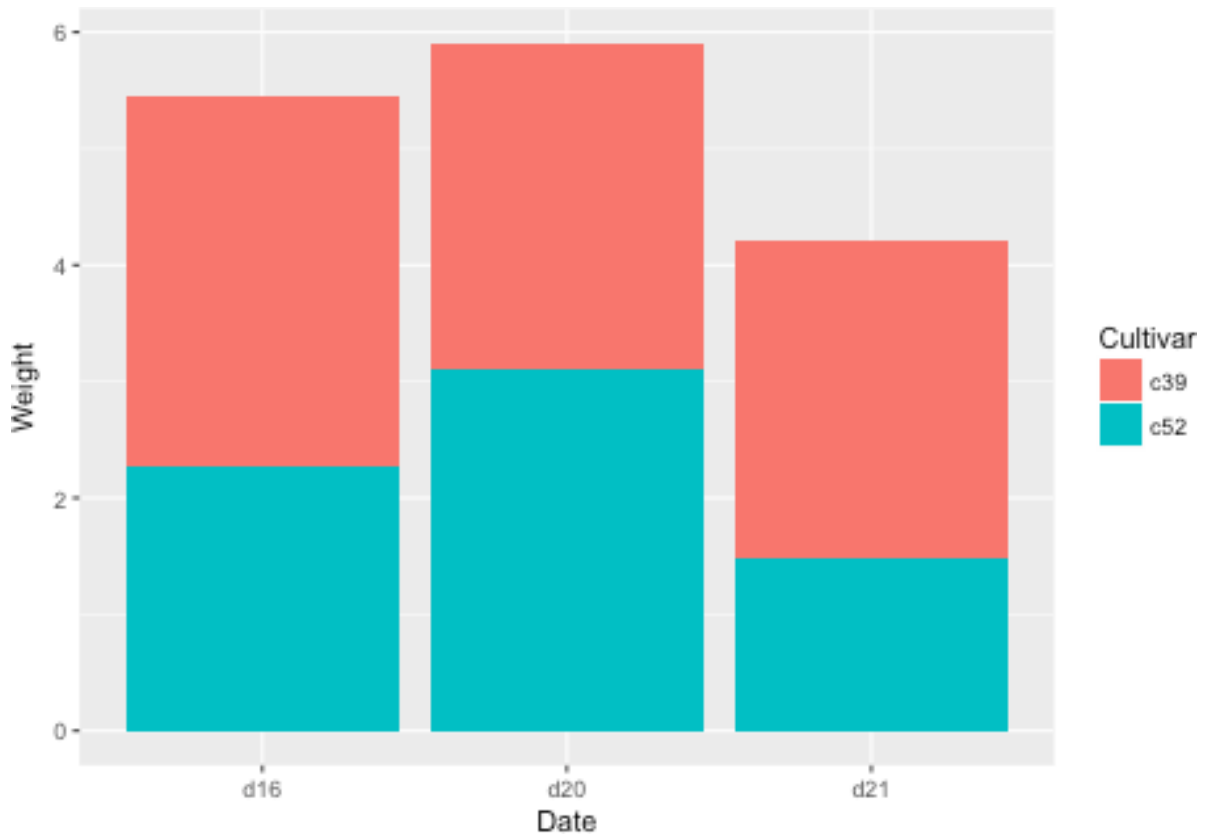
```
# 범례 순서 바꾸기 (reverse = TRUE)
```

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity") +
```



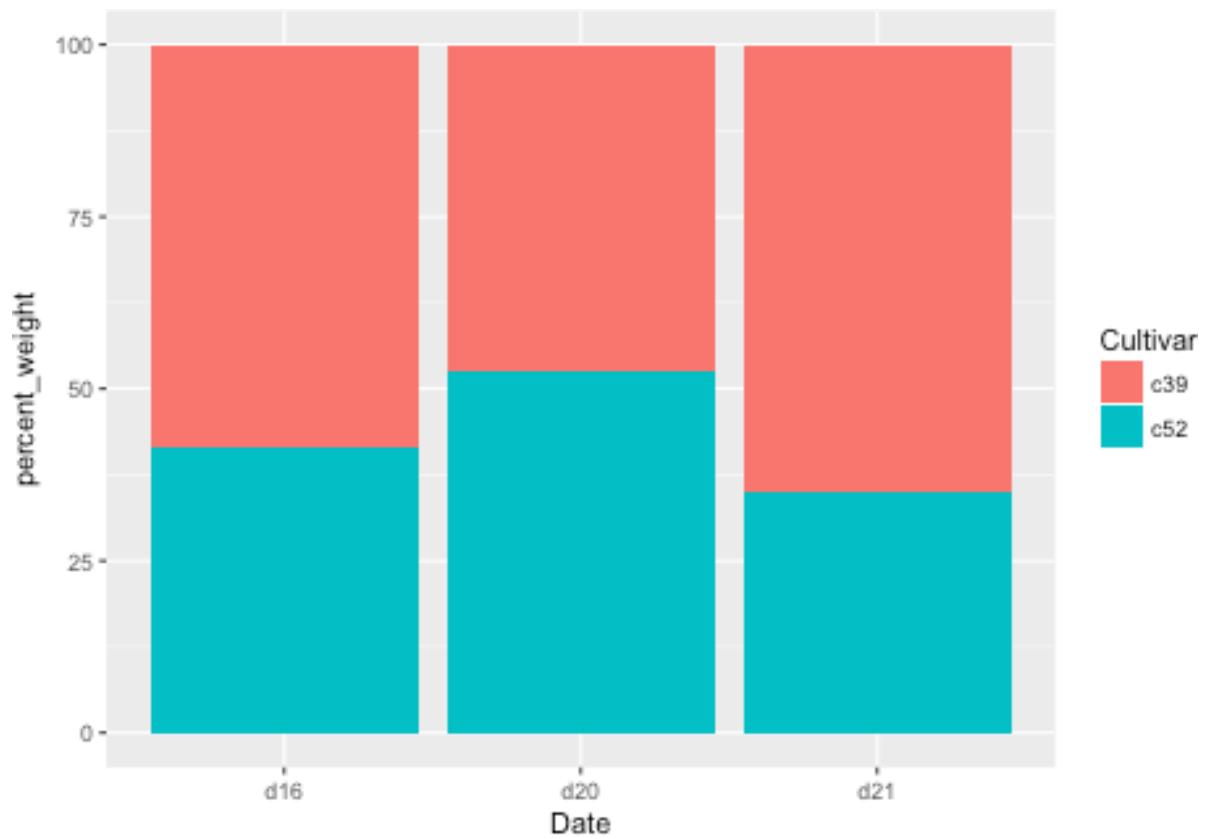
막대 쌓는 순서 바꾸기

```
ggplot(cabbage_exp, aes(x = Date, y = Weight, fill = Cultivar, order = desc(Cultivar))) + geom_bar()
```

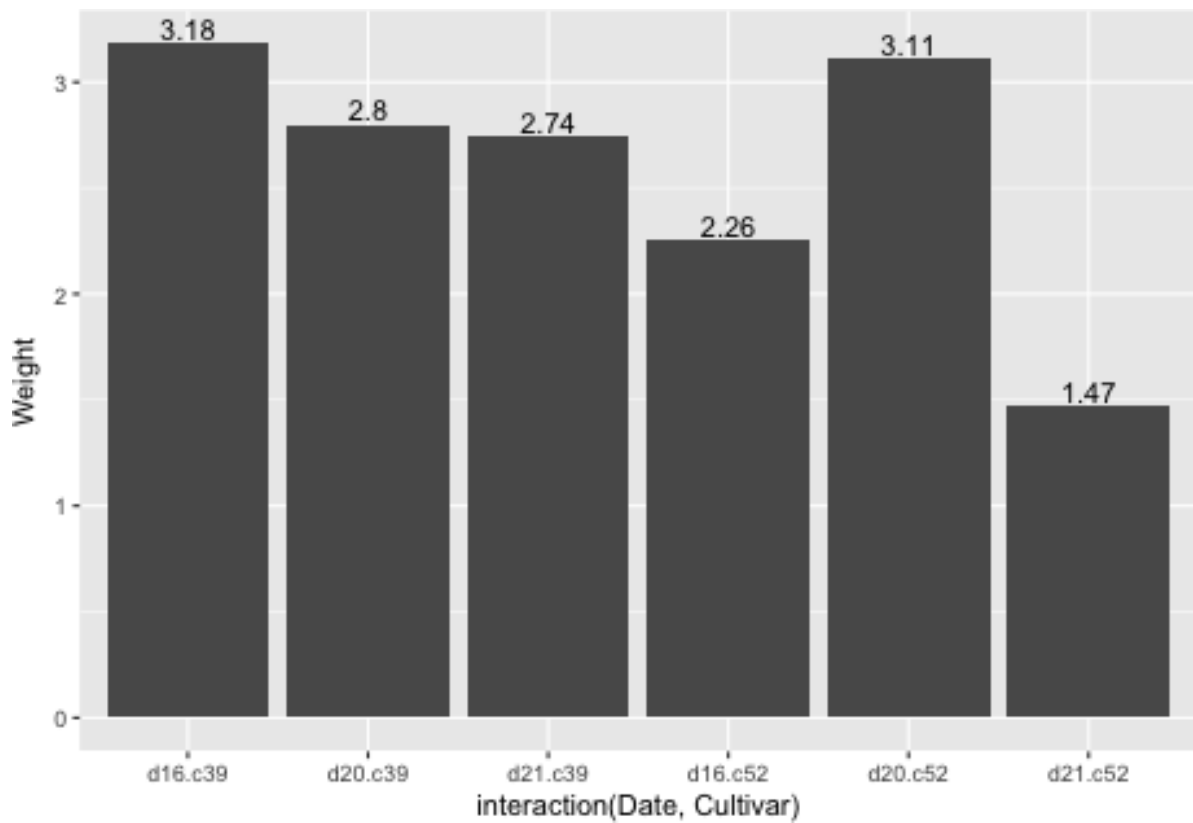


비율 누적 막대 그래프 그리기(막대 전체가 100%가 되도록)

```
ce <- ddply(cabbage_exp, "Date", transform, percent_weight = Weight / sum(Weight) * 100)
ggplot(ce, aes(x = Date, y = percent_weight, fill = Cultivar)) + geom_bar(stat = "identity")
```

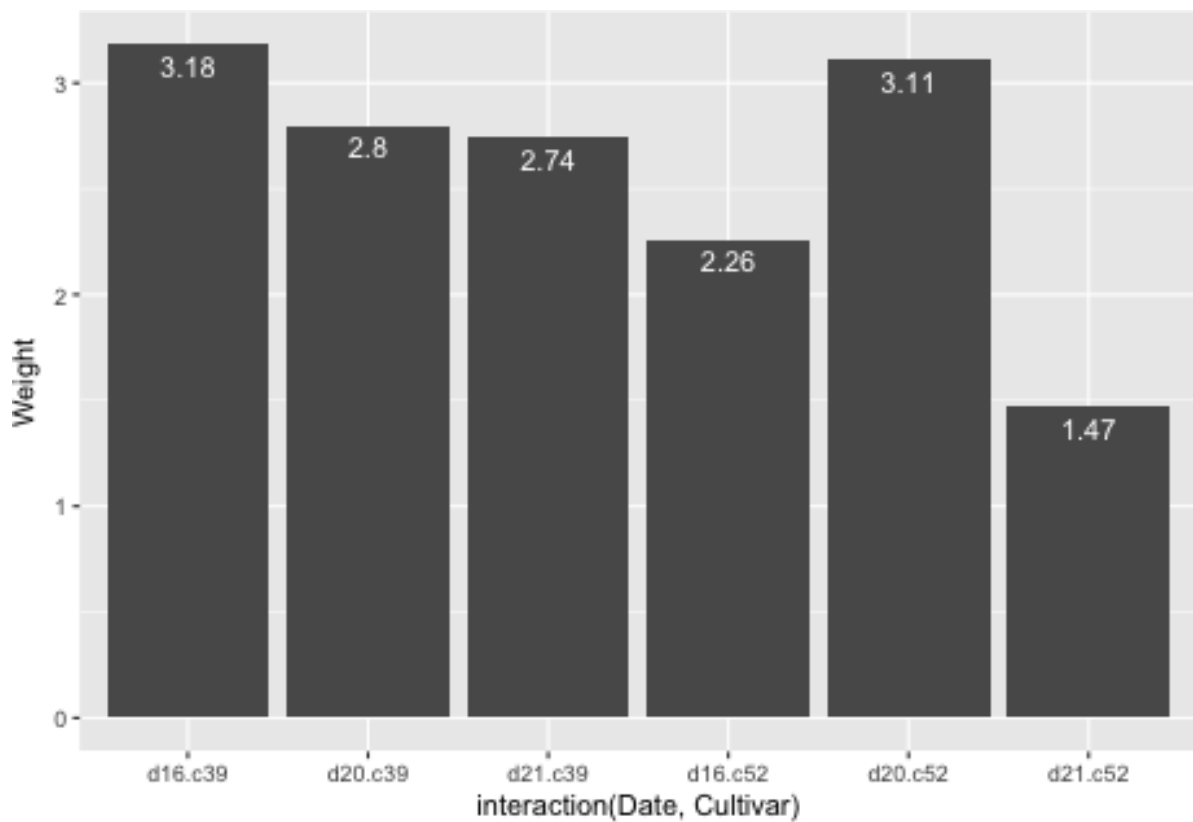



```
# 막대에 라벨 추가하기 [ geom_text ]
# 선 상단 [ vjust < 0 ]
ggplot(cabbage_exp, aes(x = interaction(Date, Cultivar), y = Weight)) + geom_bar(stat = "identity") +
  geom_text(aes(label = Weight), vjust = -0.2)
```

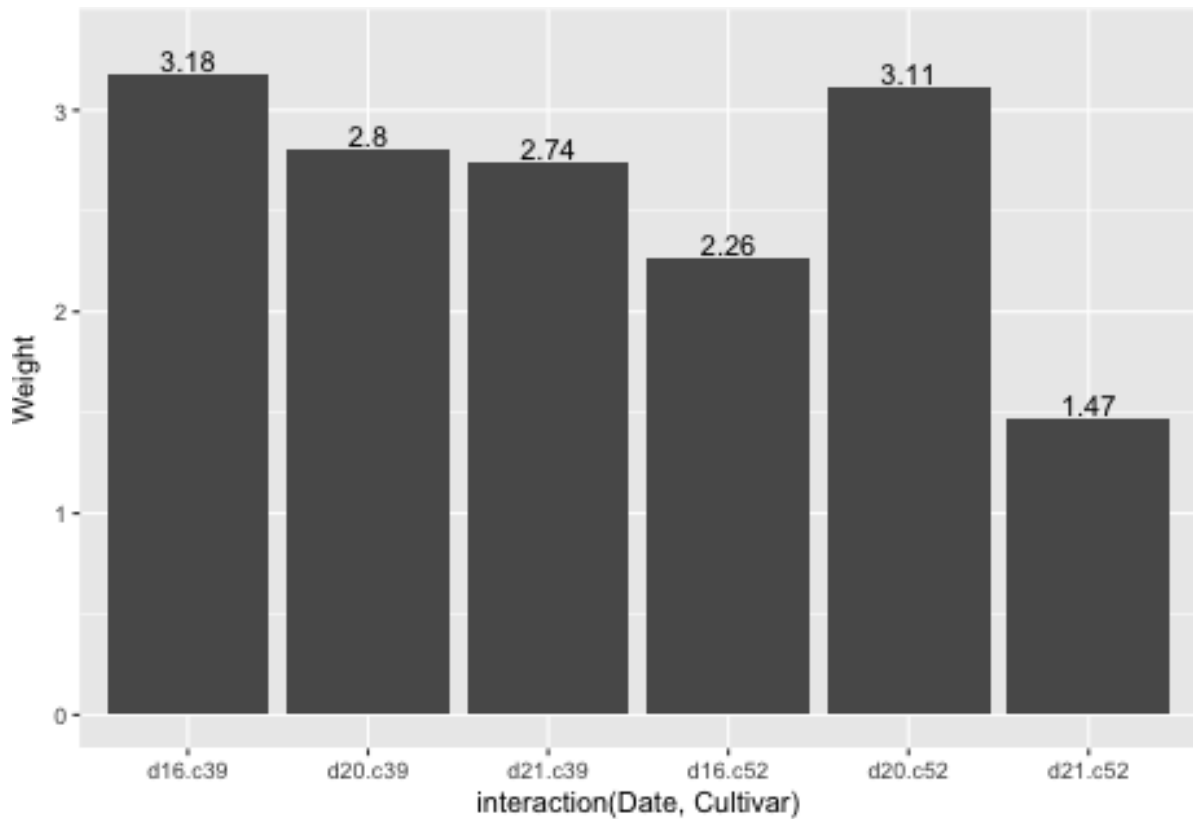


```
# 선 하단 [ vjust > 0 ]
```

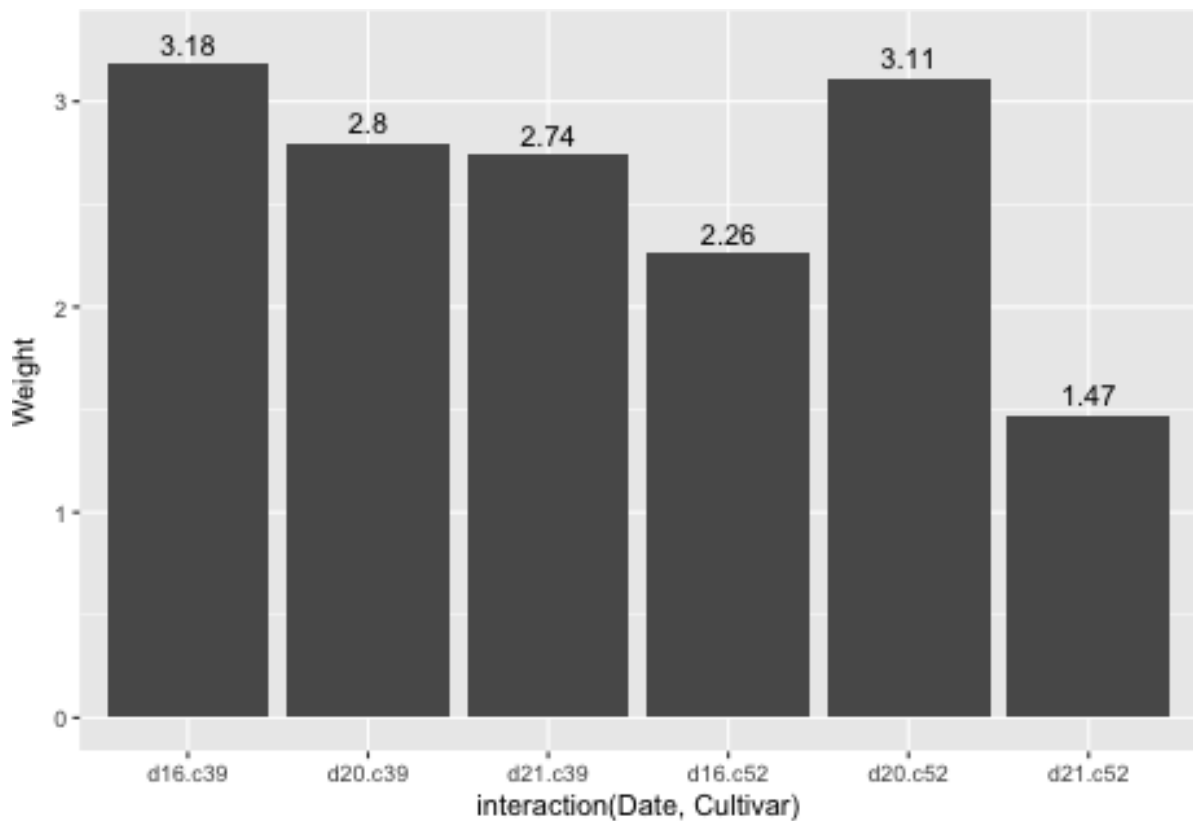
```
ggplot(cabbage_exp, aes(x = interaction(Date, Cultivar), y = Weight)) + geom_bar(stat = "identity") +  
  geom_text(aes(label = Weight), vjust = 1.5, color = "white")
```



```
# 그래프 범위 설정
# max(데이터)로 지정
ggplot(cabbage_exp, aes(x = interaction(Date, Cultivar), y = Weight)) + geom_bar(stat = "identity") +
  geom_text(aes(label = Weight), vjust = -0.2) + ylim(0, max(cabbage_exp$Weight) * 1.05)
```



```
# 막대의 상단보다 조금 높은 위치로 지정(데이터에 따라 자동 조정)
ggplot(cabbage_exp, aes(x = interaction(Date, Cultivar), y = Weight)) + geom_bar(stat = "identity") +
  geom_text(aes(y = Weight + 0.1, label = Weight))
```



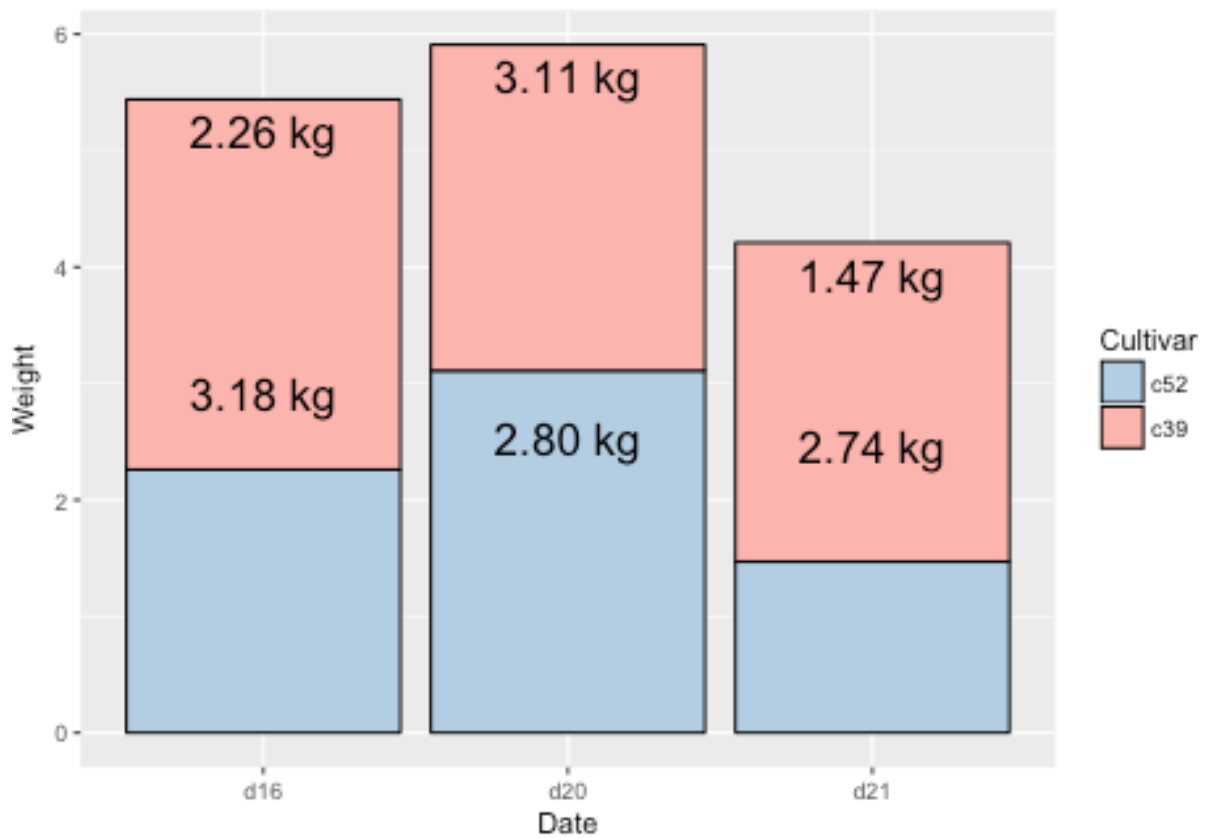
누적 합계 그래프 그리기

```
ce <- arrange(cabbage_exp, Date, Cultivar)
```

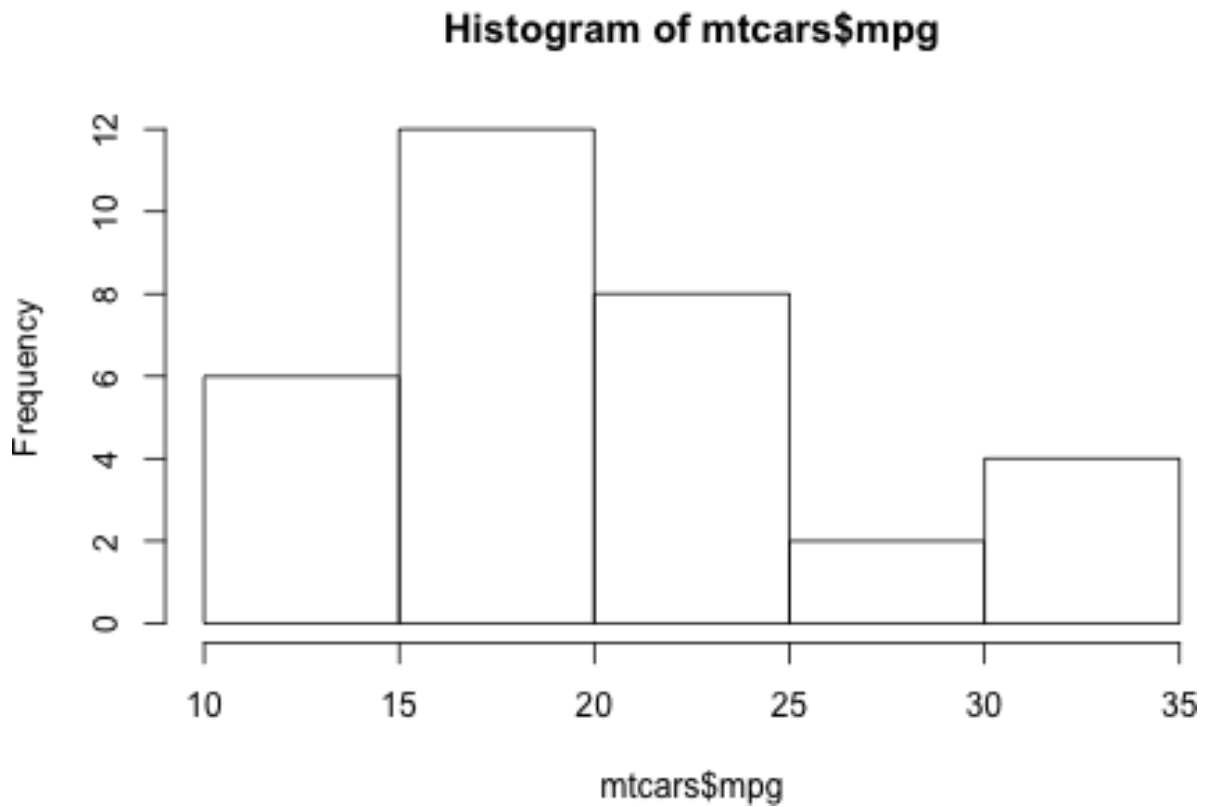
```
ce <- ddply(ce, "Date", transform, label = cumsum(Weight))
```

ddply(data.frame, 그룹 기준 마지막 변수)

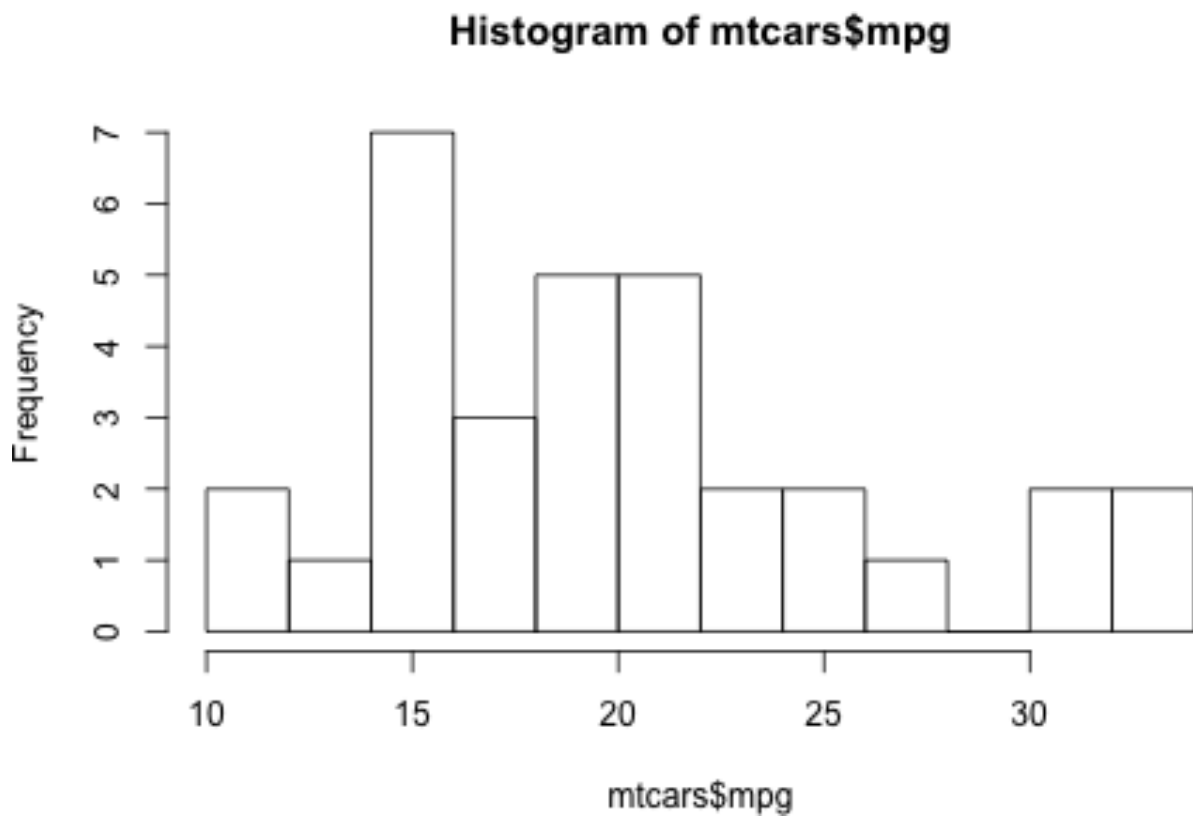
```
ggplot(ce, aes(x = Date, y = Weight, fill = Cultivar)) + geom_bar(stat = "identity", color = "black") +  
  geom_text(aes(y = label, label = paste(format(Weight, nsmall=2), "kg")), size = 6, vjust = 1.1) +  
  guides(fill = guide_legend(reverse = TRUE)) + scale_fill_brewer(palette = "Pastel1")
```



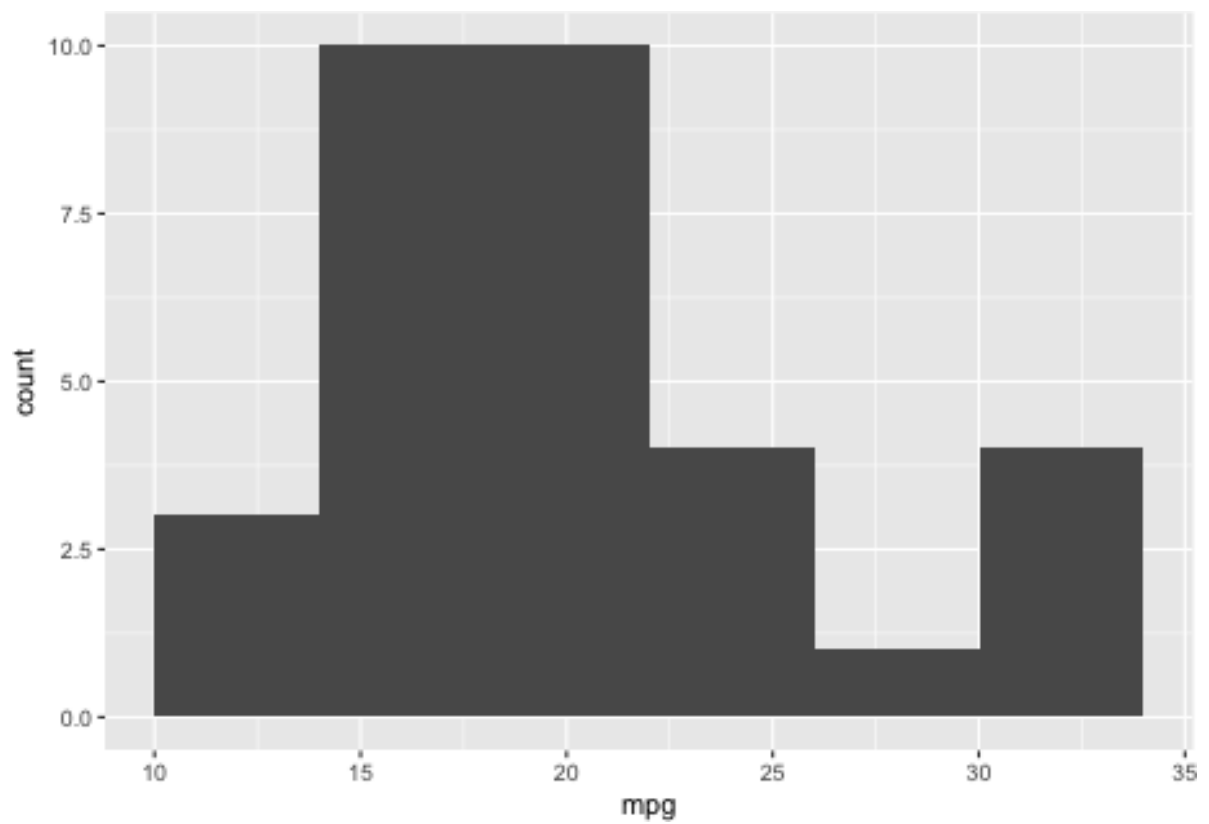
```
#####
## histogram (geom_histogram)
#####
hist(mtcars$mpg)
```



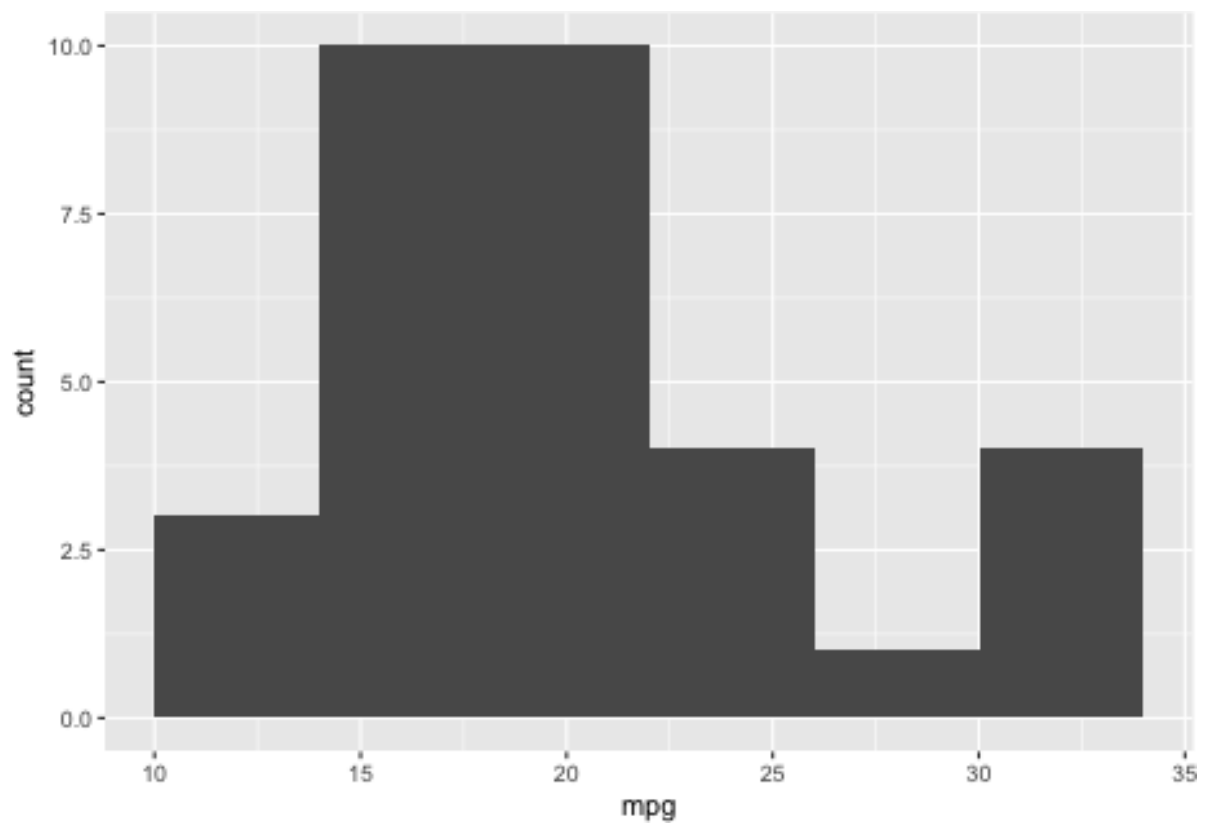
```
# 상자(bin) 개수 지정  
hist(mtcars$mpg, breaks = 10)
```



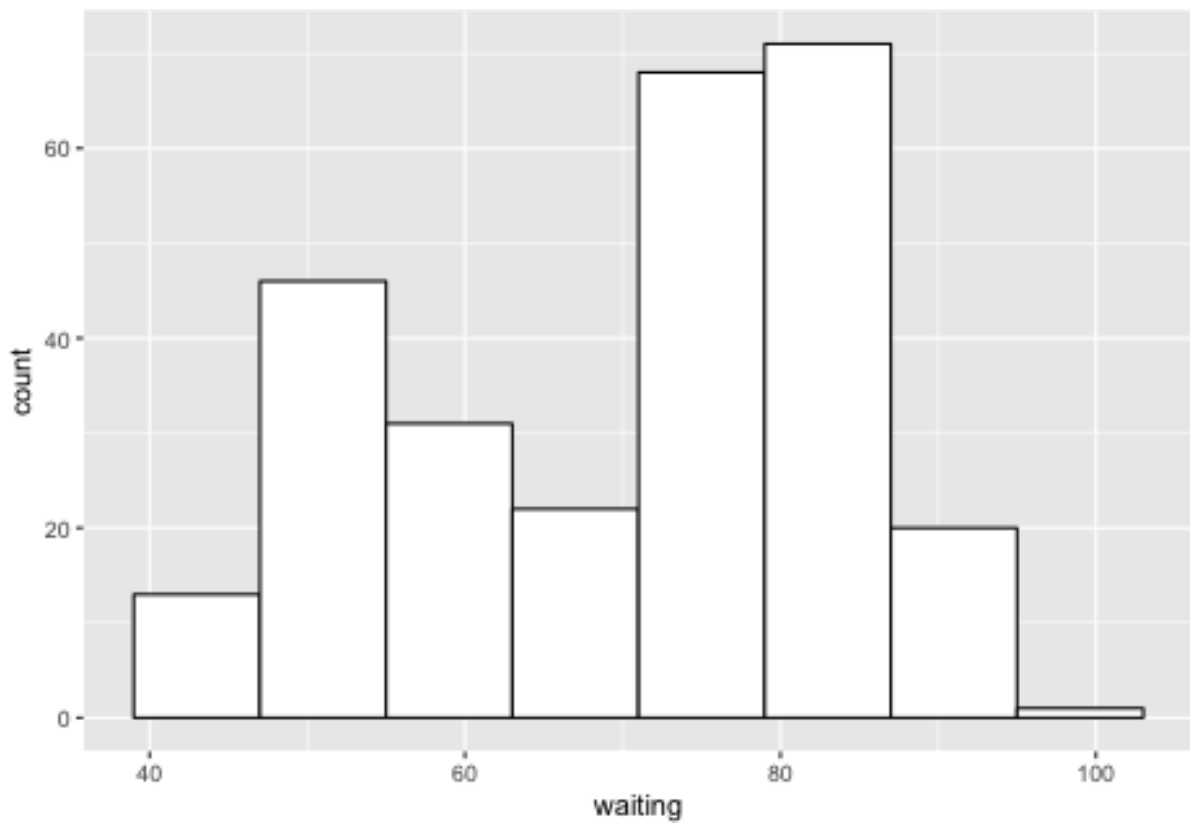
```
qplot(mpg, data = mtcars, binwidth = 4)
```



```
# ggplot(mtcars, aes(x = mpg)) + geom_histogram()  
ggplot(mtcars, aes(x = mpg)) + geom_histogram(binwidth = 4)
```



```
h <- ggplot(faithful, aes(x = waiting))
h + geom_histogram(binwidth = 8, fill = "white", color = "black", origin = 31)
## Warning: `origin` is deprecated. Please use `boundary`
## instead.
```

```
## Warning: `origin` is deprecated. Please use `boundary` instead.
```

```
# 그룹화된 데이터로 여러개의 히스토그램 만들기
```

```
library(MASS)
```

```
##
```

```
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
```

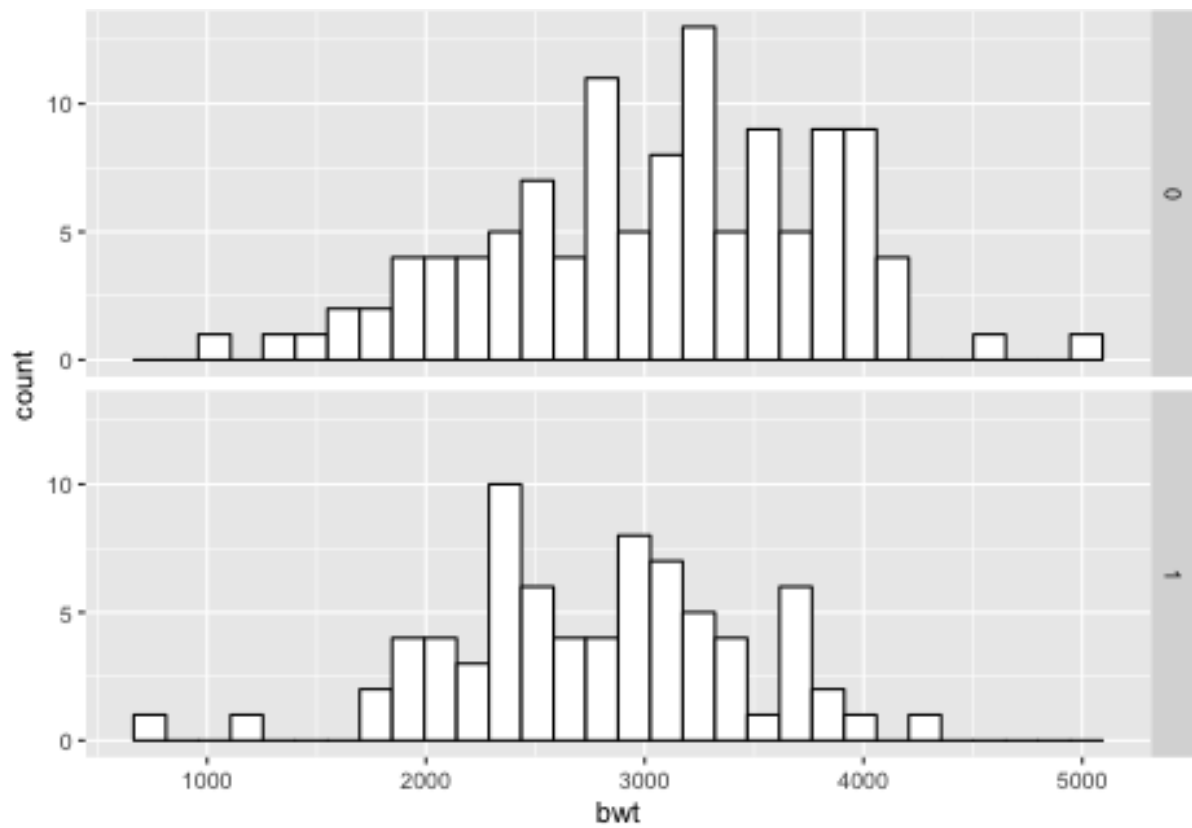
```
##
```

```
## select
```

```
ggplot(birthwt, aes(x = bwt)) + geom_histogram(fill = "white", color = "black") + facet_grid(sr
```

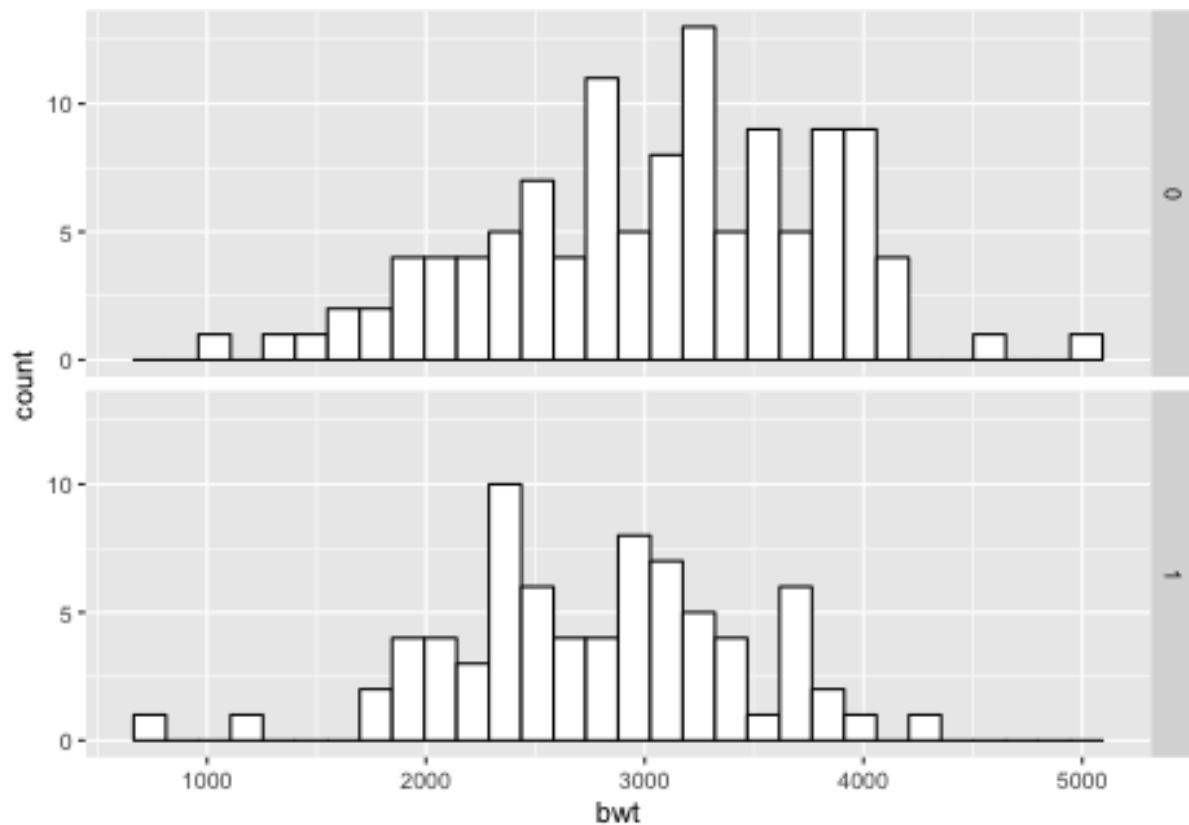
```
## `stat_bin()` using `bins = 30`. Pick better value
```

```
## with `binwidth`.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
ggplot(birthwt, aes(x = bwt)) + geom_histogram(fill = "white", color = "black") + facet_grid(sr
## `stat_bin()` using `bins = 30`. Pick better value
## with `binwidth`.
```



```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

그룹 라벨 정의

```
birthwt1 <- birthwt
```

```
birthwt1$smoke <- factor(birthwt1$smoke)
```

```
levels(birthwt1$smoke)
```

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

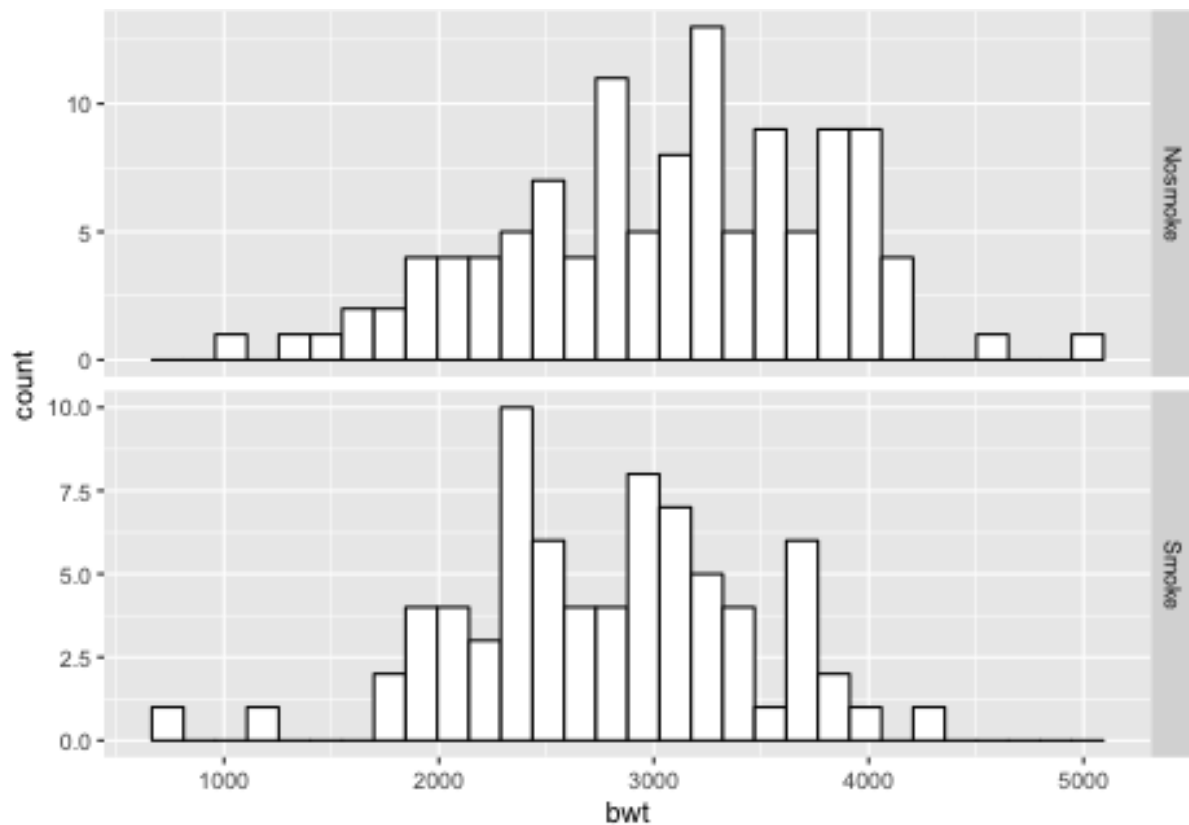
```
## [1] "0" "1"
```

```
birthwt$smoke <- revalue(birthwt1$smoke, c("0" = "Nosmoke", "1" = "Smoke"))
```

```
ggplot(birthwt, aes(x = bwt)) + geom_histogram(fill = "white", color = "black") + facet_grid(smoke ~ 1)
```

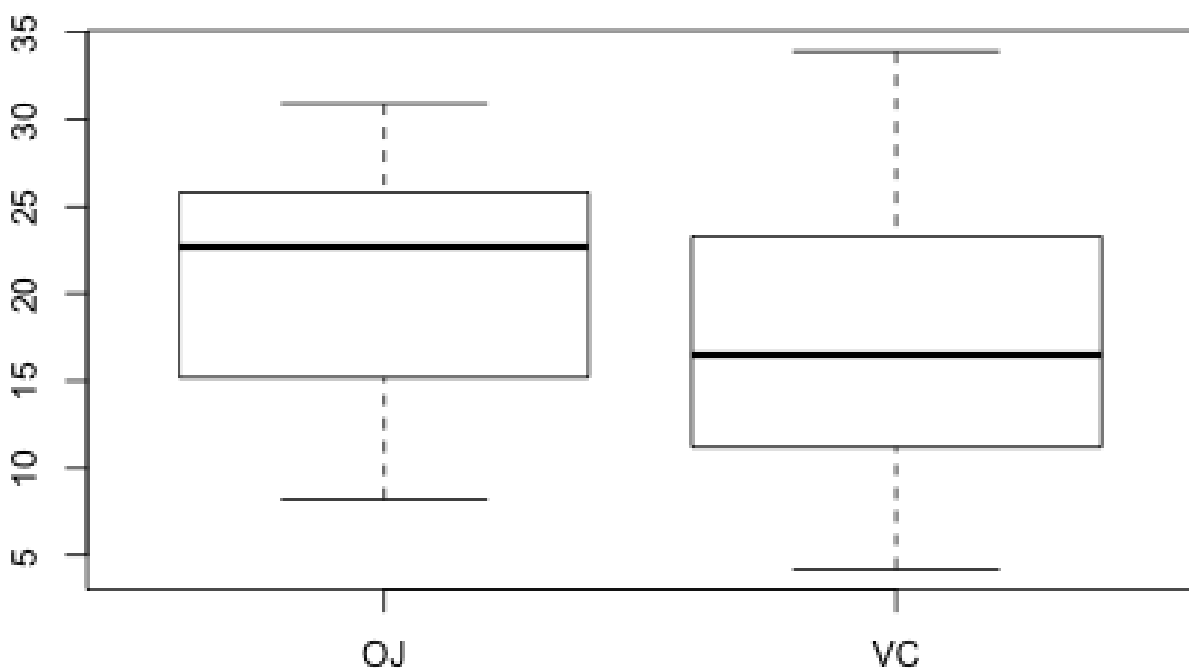
```
## `stat_bin()` using `bins = 30`. Pick better value
```

```
## with `binwidth`.
```

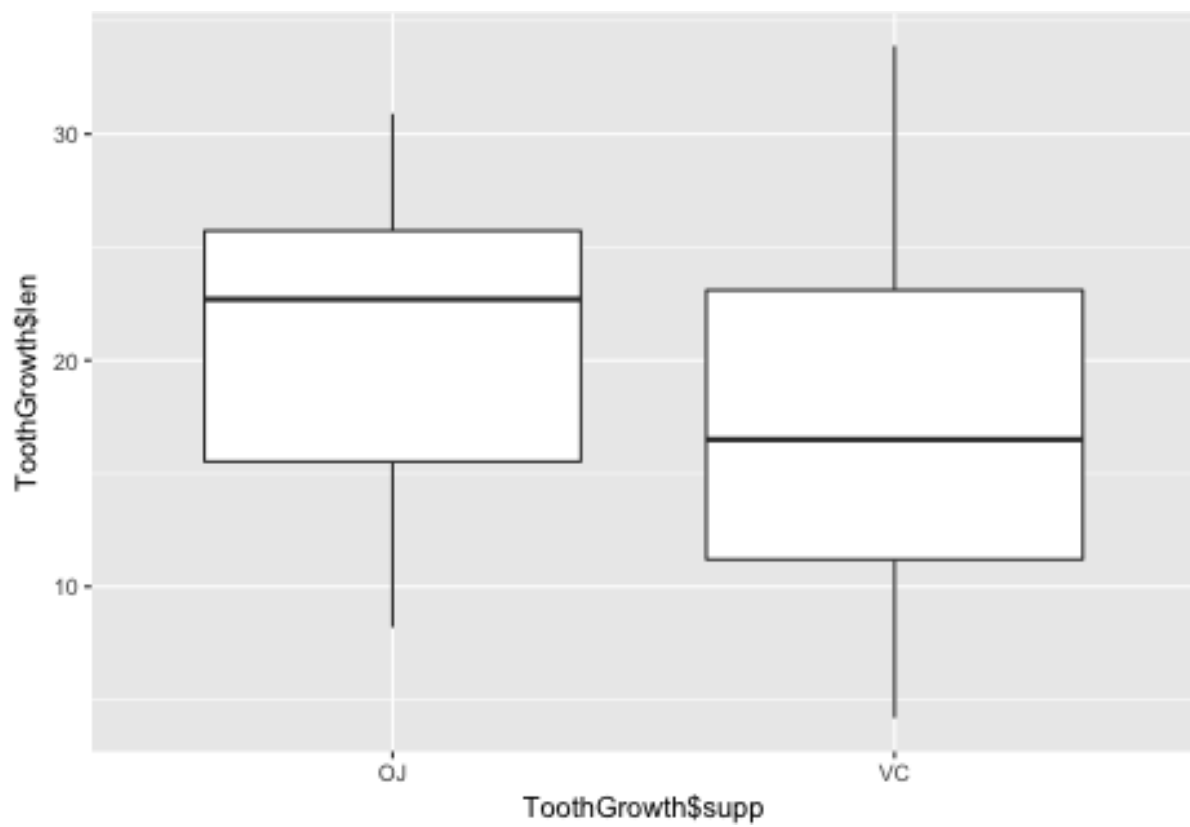


```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

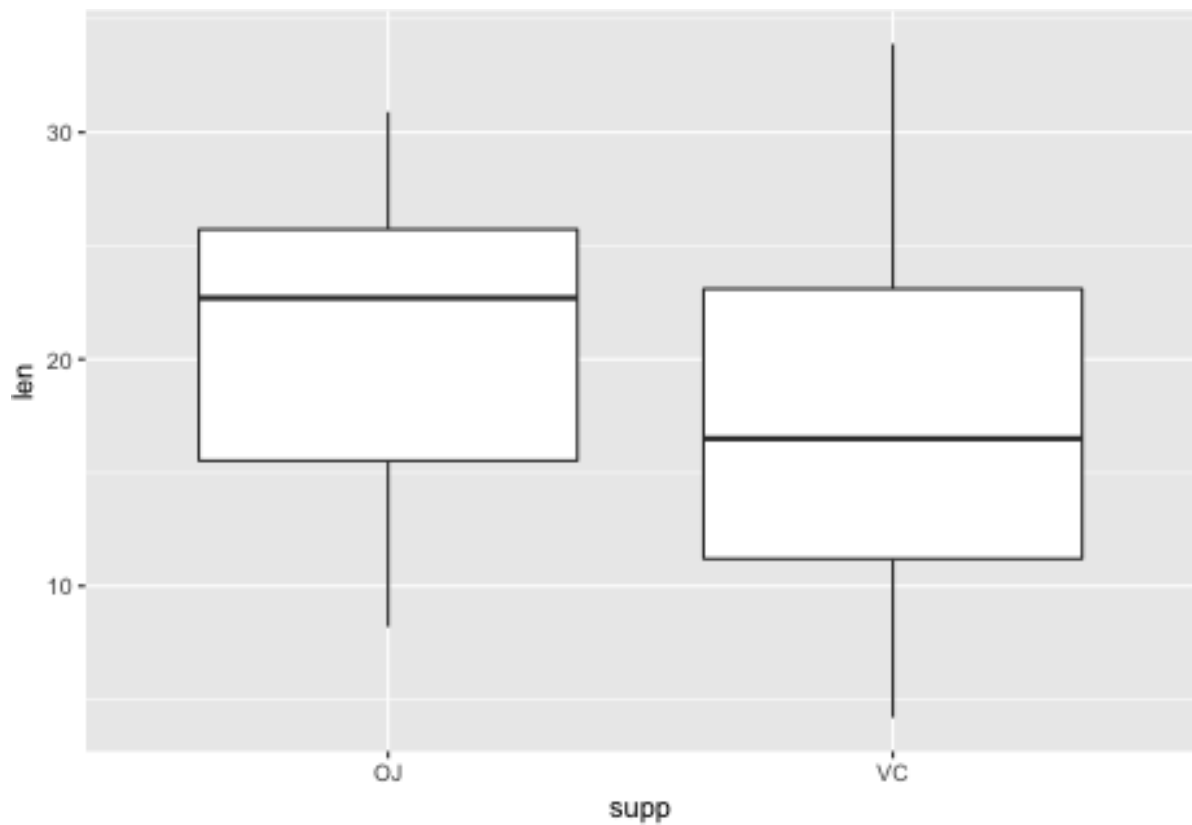
```
#####
## boxplot (geom_boxplot)
#####
boxplot(len ~ supp, data = ToothGrowth)
```



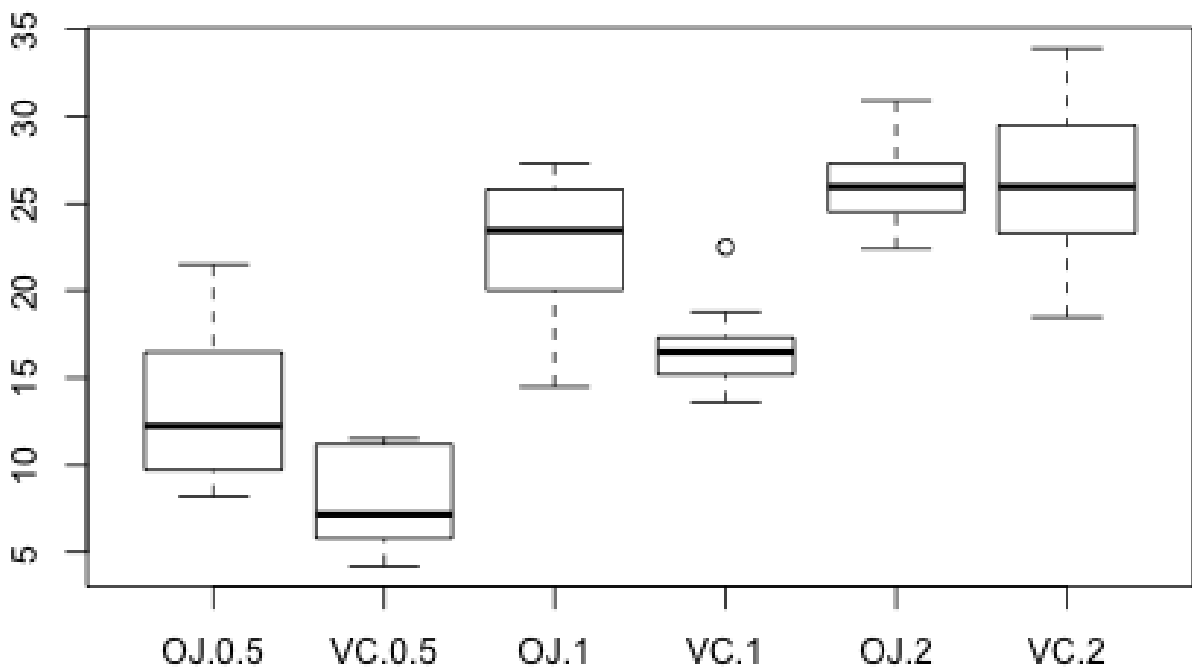
```
qplot(ToothGrowth$supp, ToothGrowth$len, geom = "boxplot")
```



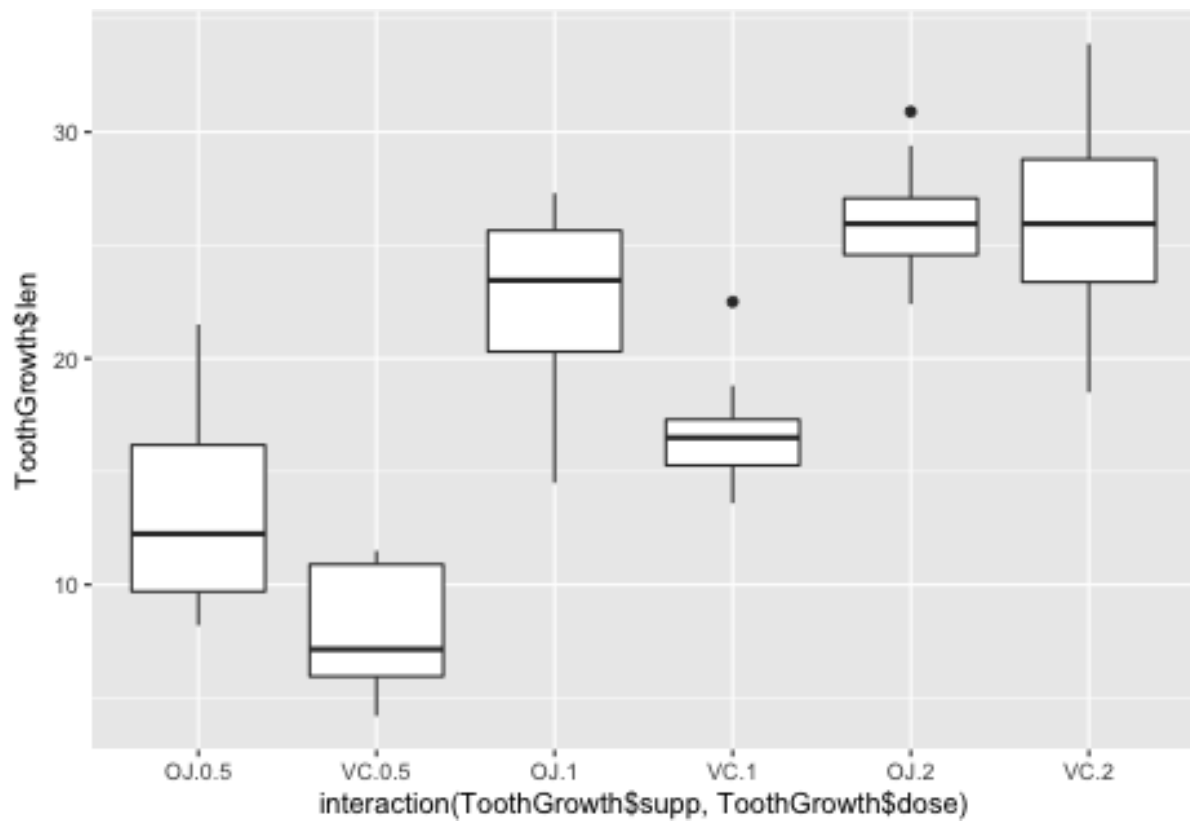
```
ggplot(ToothGrowth, aes(x = supp, y = len)) + geom_boxplot()
```



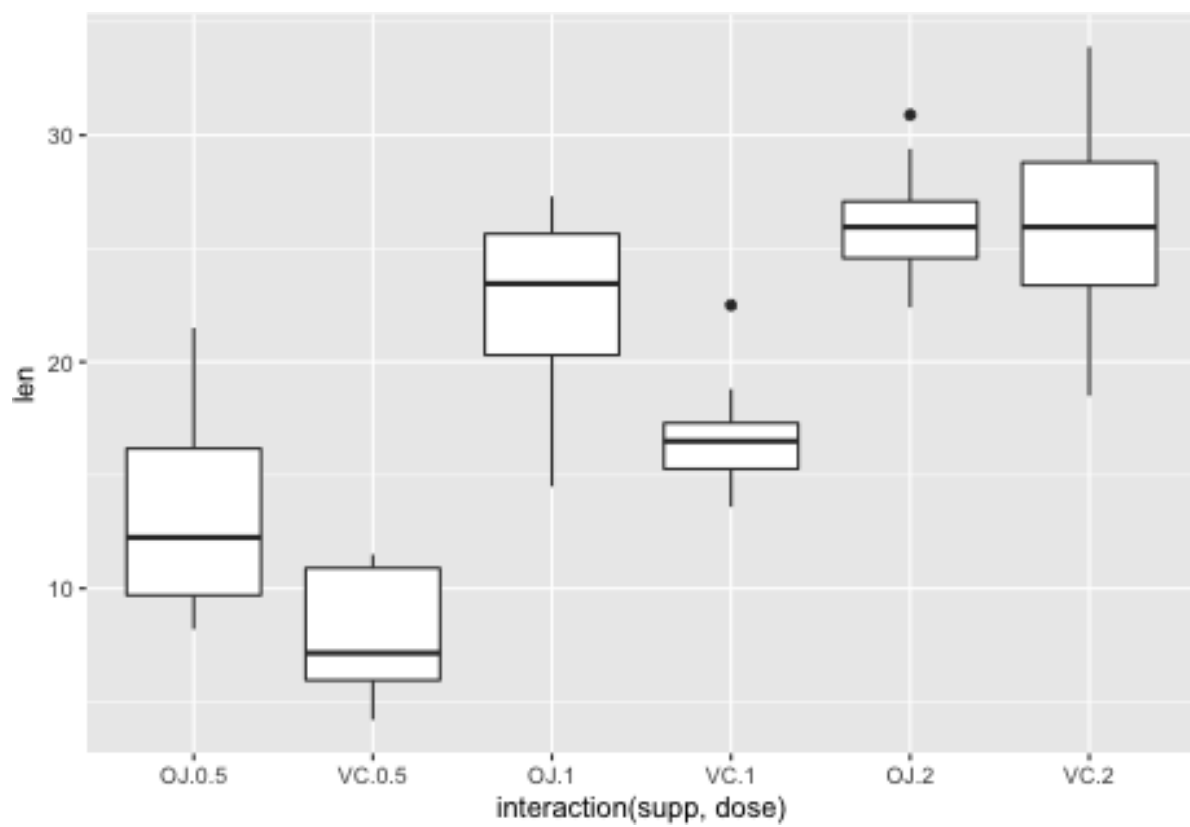
```
# 두 변수의 상호작용을 x축으로
boxplot(len ~ supp + dose, data = ToothGrowth)
```



```
qplot(interaction(ToothGrowth$supp, ToothGrowth$dose), ToothGrowth$len, geom = "boxplot")
```

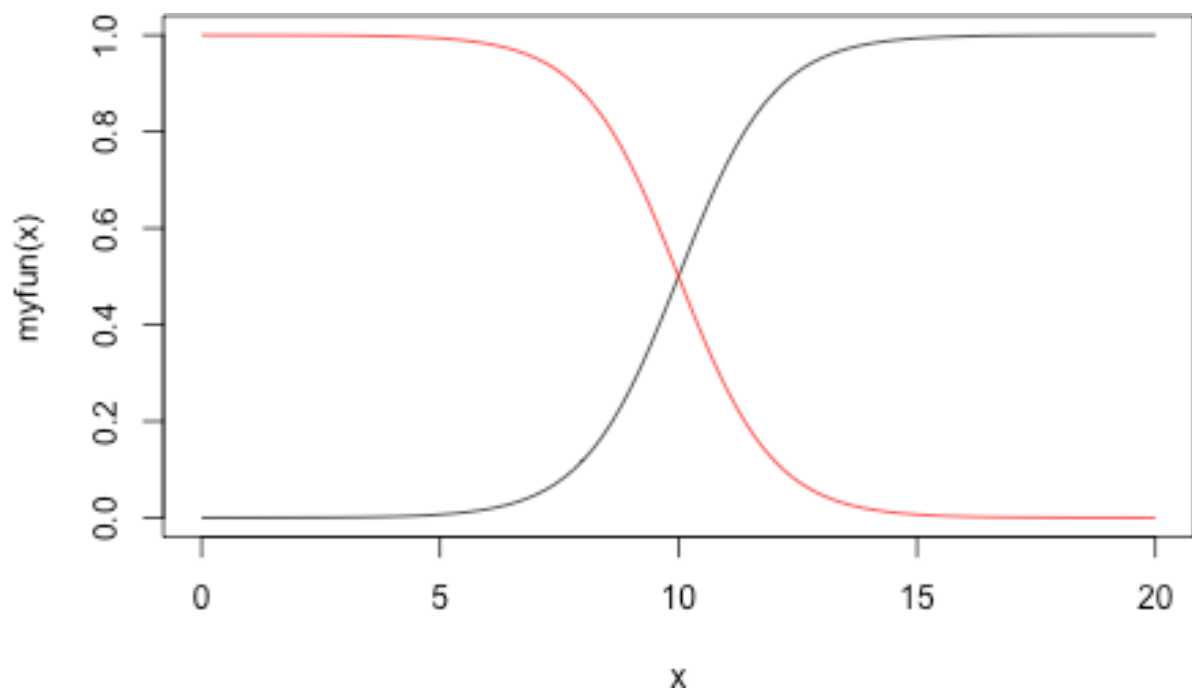


```
ggplot(ToothGrowth, aes(x = interaction(supp, dose), y = len)) + geom_boxplot()
```

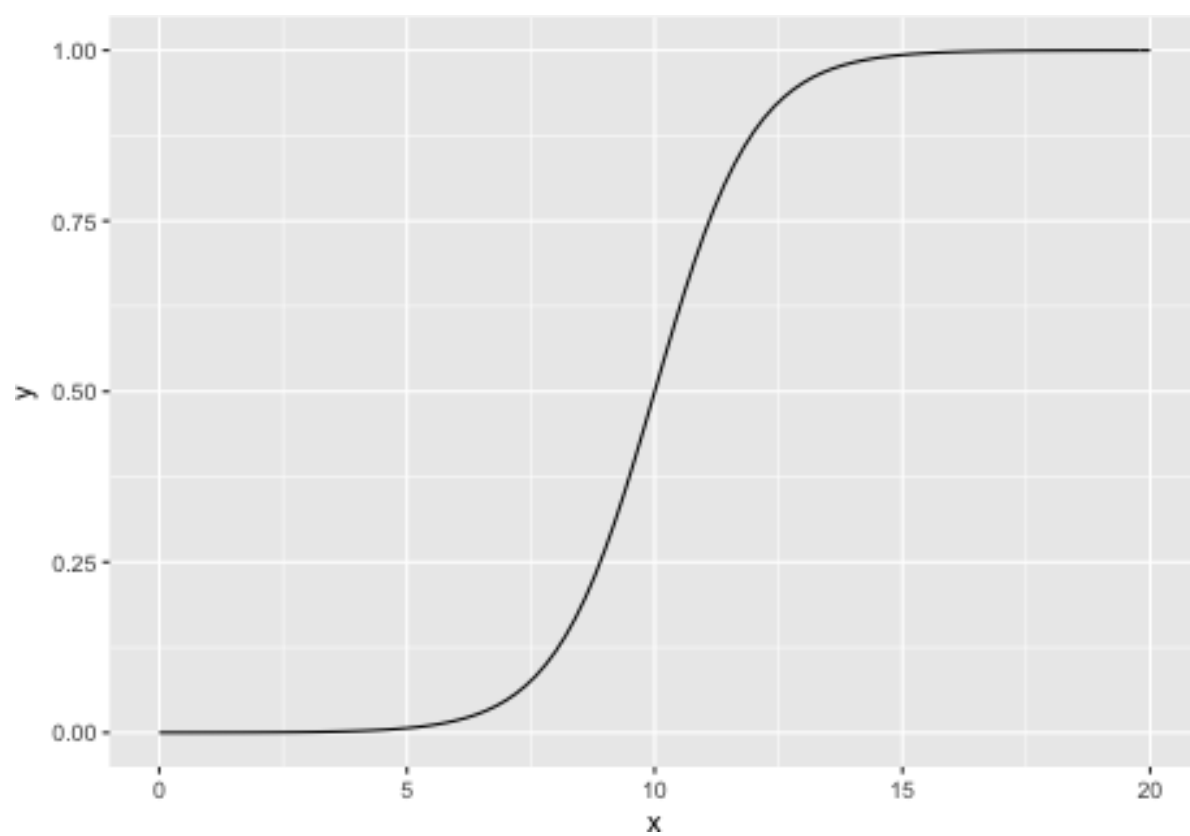


```
#####
## 함수곡선 그리기 (stat_function)
#####
# 사용자함수 정의
myfun <- function(xvar){
  1/(1 + exp(-xvar + 10))
}

curve(myfun(x), from = 0, to = 20)
curve(1-myfun(x), add = TRUE, col = "red")
```



```
# qplot(c(0,20), fun = myfun, stat = "function", geom = "line")
ggplot(data.frame(x = c(0,20)), aes(x = x)) + stat_function(fun = myfun, geom = "line")
```

.1 Assignment 1

첨부한 concUnitConv-test.R과 유사한 R script를 실행하였을 때, concUnitConv-test.Rout과 유사한 결과나 나오는 concUnitConv.R 파일을 작성하시오.

- 제출기한: 2017-05-10 18:00
- 제출방법: R script와 output을 k@acr.kr¹, shan@acp.kr², sec@acp.kr³ 로 제출

.1.1 concUnitConv-test.R

```
source("D:/G/Desk/R/concUnitConv.R")

concUnitConv() # Wrong input
concUnitConv("kg/L", "g/L") # Wrong input
concUnitConv("g/kL", "g/L") # Wrong input

concUnitConv("mg/L", "ug/mL")
Theoph$conc * concUnitConv("mg/L", "ug/L")
Theoph$conc * concUnitConv("mg/L", "mg/mL")
Theoph$conc * concUnitConv("mg/L", "mmol/L") # Wrong input
Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=-100) # Wrong input
Theoph$conc * concUnitConv("mg/L", "mM", MW=180.164) # Wrong input
Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=180.164)
Theoph$mM = Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=180.164)
Theoph$mM * concUnitConv("mmol/L", "ug/L", MW=180.164)
Theoph$mM * concUnitConv("mmol/L", "ug/mL", MW=180.164)
```

.1.2 concUnitConv-test.Rout

```
> source("D:/G/Desk/R/concUnitConv.R")
>
> concUnitConv() # Wrong input
Error in concUnitConv() : Source concentration unit is not valid.
> concUnitConv("kg/L", "g/L") # Wrong input
```

¹<mailto:k@acr.kr>

²<mailto:shan@acp.kr>

³<mailto:sec@acp.kr>

```

Error in concUnitConv("kg/L", "g/L") : Source amount is not supported.
> concUnitConv("g/kL", "g/L") # Wrong input
Error in concUnitConv("g/kL", "g/L") : Volume unit is not supported.
>
> concUnitConv("mg/L", "ug/mL")

1
> Theoph$conc * concUnitConv("mg/L", "ug/L")
 [1] 740 2840 6570 10500 9660 8580 8360 7470 6890 5940 3280 0 1720 7910 8310
[28] 7500 6200 5300 4900 3700 1050 0 1890 4600 8600 8380 7540 6880 5780 5330
[55] 1570 0 1290 3080 6440 6320 5530 4940 4020 3460 2780 920 150 850 2350
[82] 7560 6590 5880 4730 4570 3000 1250 0 7370 9030 7140 6330 5660 5670 4240
[109] 5680 2420 0 4860 7240 8000 6810 5870 5220 4450 3620 2690 860 0 1250
> Theoph$conc * concUnitConv("mg/L", "mg/mL")
 [1] 0.00074 0.00284 0.00657 0.01050 0.00966 0.00858 0.00836 0.00747 0.00689 0.00594 0.00328 0
[21] 0.00301 0.00090 0.00000 0.00440 0.00690 0.00820 0.00780 0.00750 0.00620 0.00530 0.00490 0
[41] 0.00578 0.00533 0.00419 0.00115 0.00000 0.00202 0.00563 0.01140 0.00933 0.00874 0.00756 0
[61] 0.00553 0.00494 0.00402 0.00346 0.00278 0.00092 0.00015 0.00085 0.00235 0.00502 0.00658 0
[81] 0.00731 0.00756 0.00659 0.00588 0.00473 0.00457 0.00300 0.00125 0.00000 0.00737 0.00903 0
[101] 0.00289 0.00522 0.00641 0.00783 0.01021 0.00918 0.00802 0.00714 0.00568 0.00242 0.00000 0
[121] 0.00086 0.00000 0.00125 0.00396 0.00782 0.00972 0.00975 0.00857 0.00659 0.00611 0.00457 0
> Theoph$conc * concUnitConv("mg/L", "mmol/L") # Wrong input
Error in concUnitConv("mg/L", "mmol/L") :
  Positive molecular weight should be given.
> Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=-100) # Wrong input
Error in concUnitConv("mg/L", "mmol/L", MW = -100) :
  Positive molecular weight should be given.
> Theoph$conc * concUnitConv("mg/L", "mM", MW=180.164) # Wrong input
Error in concUnitConv("mg/L", "mM", MW = 180.164) :
  Target concentration unit is not valid.
> Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=180.164)
 [1] 0.0041073688 0.0157634156 0.0364667747 0.0582802336 0.0536178149 0.0476232766 0.046402160
[13] 0.0095468573 0.0439044426 0.0461246420 0.0462356520 0.0380209143 0.0337470305 0.029972691
[25] 0.0382984392 0.0455140872 0.0432938878 0.0416287383 0.0344130903 0.0294176417 0.027197442
[37] 0.0477342865 0.0465131769 0.0418507582 0.0381874292 0.0320818810 0.0295841567 0.023256588
[49] 0.0517861504 0.0485113563 0.0419617682 0.0393530339 0.0327479408 0.0242556782 0.008714282
[61] 0.0306942563 0.0274194623 0.0223130037 0.0192047246 0.0154303856 0.0051064586 0.000832574
[73] 0.0369663196 0.0291401168 0.0243666881 0.0195932595 0.0063830732 0.0000000000 0.016929020
[85] 0.0262538576 0.0253657778 0.0166514953 0.0069381230 0.0000000000 0.0409071735 0.050121000
[97] 0.0228125486 0.0175395751 0.0062165582 0.0013321196 0.0160409405 0.0289736018 0.035578695
[109] 0.0315268311 0.0134322062 0.0000000000 0.0269754224 0.0401856087 0.0444039875 0.037798894
[121] 0.0047734287 0.0000000000 0.0069381230 0.0219799738 0.0434048978 0.0539508448 0.054117359
> Theoph$mM = Theoph$conc * concUnitConv("mg/L", "mmol/L", MW=180.164)
> Theoph$mM * concUnitConv("mmol/L", "ug/L", MW=180.164)

```

```

[1] 740 2840 6570 10500 9660 8580 8360 7470 6890 5940 3280 0 1720 7910 8310
[28] 7500 6200 5300 4900 3700 1050 0 1890 4600 8600 8380 7540 6880 5780 5330
[55] 1570 0 1290 3080 6440 6320 5530 4940 4020 3460 2780 920 150 850 2350
[82] 7560 6590 5880 4730 4570 3000 1250 0 7370 9030 7140 6330 5660 5670 4240
[109] 5680 2420 0 4860 7240 8000 6810 5870 5220 4450 3620 2690 860 0 1250
> Theoph$MM * concUnitConv("mmol/L", "ug/mL", MW=180.164)
[1] 0.74 2.84 6.57 10.50 9.66 8.58 8.36 7.47 6.89 5.94 3.28 0.00 1.72 7.91 8.31
[28] 7.50 6.20 5.30 4.90 3.70 1.05 0.00 1.89 4.60 8.60 8.38 7.54 6.88 5.78 5.33
[55] 1.57 0.00 1.29 3.08 6.44 6.32 5.53 4.94 4.02 3.46 2.78 0.92 0.15 0.85 2.35
[82] 7.56 6.59 5.88 4.73 4.57 3.00 1.25 0.00 7.37 9.03 7.14 6.33 5.66 5.67 4.24
[109] 5.68 2.42 0.00 4.86 7.24 8.00 6.81 5.87 5.22 4.45 3.62 2.69 0.86 0.00 1.25
>
>

```


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))
      , 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))
      , 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))
      , 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"), lty=3)
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)
}

```



```

for (i in 1:10) print(2*i)

while ( ) {
# Statements
}

# # if ~ else ~
# if ( ) {
# # Statements 1
# } else {
# # Statements 2
# }
#
# if ( ) # Statement1
# else # Statement2
#
# if ( ) {
# # Statements 1
# } else if ( ) {
# # Statements 2
# } else if ( ) {
# # Statements 3
# } else {
# # Statements 4
# }

#
#

# Chapter 10 Writing your own functions

Square = function(x=0)
{
  return(x*x)
}

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 = 2*(1 - pt(abs(tst), df=DF))

  Res = list(tst, DF, p.val, yb1, yb2)
  names(Res) = c("t", "df", "p-value", "mean of x", "mean of y")

  return (Res)
}

res = T.test(x, y)
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)

# More basic keywords and functions
1 %in% c(1,2,3,4)
5 %in% c(1,2,3,4)
is.finite(Inf)
prod(1:3)
cummax(1:10)
cummax(10:1)
# ?xor
x = 11:20
x
which(x==3)
which(x==13)

length(x)
y = "my string"
length(y)
nchar(y)
strsplit(y, " ")
strsplit(y, " ")[[1]]
substr(y, 4, 5)

sample(1:10)
sample(1:10, 20)
sample(1:10, 20, replace=TRUE)
sample(rep(1:10,2))
```


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 느껴지셨다면 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 안보이신다고 하신 분들이 몇분 계셔서 첨언합니다. Coursera 회원가입하고 로그인 한 뒤, Enroll Now를 누르시면 Audit 혹은 청강하기 라디오버튼을 보실 수 있습니다. 앱에서도 마찬가지입니다. 이외에도 코세라에는 많은 유익한 강의를 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](#)