

R

Jeff Lin

2020-09-16



# Contents

		<b>5</b>
<b>1</b>	<b>R</b>	<b>7</b>
1.1	R . . . . .	7
1.2	RSudio . . . . .	7
1.3	. . . . .	8
1.4	Object . . . . .	9
1.5	. . . . .	9
1.6	. . . . .	11
1.7	. . . . .	11
1.8	. . . . .	11
1.9	RStudio . . . . .	12
1.10	Function . . . . .	13
1.11	Packages . . . . .	14
1.12	. . . . .	14
1.13	. . . . .	15
<b>2</b>	<b>Vector</b>	<b>17</b>
2.1	Vector . . . . .	17
2.2	. . . . .	17
2.3	. . . . .	20
2.4	. . . . .	21
2.5	. . . . .	23
2.6	. . . . .	24
2.7	. . . . .	25
2.8	Inxex . . . . .	26
2.9	( ) Missing Values . . . . .	27
2.10	Factor . . . . .	28
<b>3</b>		<b>33</b>
3.1	Matrix . . . . .	33
3.2	Matrix Index . . . . .	36
3.3	List . . . . .	41

3.4	Data Frame . . . . .	45
3.5	data.frame() . . . . .	45
<b>4</b>		<b>49</b>
4.1	. . . . .	49
4.2	ASCII R : . . . . .	50
4.3	ASCII R : . . . . .	52
4.4	R . . . . .	53
4.5	{R} . . . . .	54
4.6	{R} . . . . .	54
<b>5</b>		<b>55</b>
5.1	. . . . .	55
5.2	ggplot2 . . . . .	55
5.3	. . . . .	57
5.4	. . . . .	69
5.5	. . . . .	87
5.6	. . . . .	108
<b>6</b>		<b>113</b>
6.1	. . . . .	113
6.2	. . . . .	114
6.3	. . . . .	115
6.4	: rep() . . . . .	116
6.5	Arithmetic Computing Function . . . . .	117
6.6	: choose() factorial() . . . . .	121
6.7	: all(), any(), which() . . . . .	122
6.8	Ranking and Sorting . . . . .	123
6.9	. . . . .	125
<b>7</b>		<b>127</b>
7.1	. . . . .	128
7.2	. . . . .	129
7.3	. . . . .	137
7.4	sample() . . . . .	140
<b>8</b>		<b>143</b>
8.1	readr . . . . .	144
8.2	Tidy Data . . . . .	145
8.3	Tibble Data Frame . . . . .	145
8.4	Pipe . . . . .	146
8.5	. . . . .	147
8.6	dplyr . . . . .	147
8.7	. . . . .	152

R, , , R . R (open-source, GNU General Public License), R {R} (R core-development team) , .

R , R Base , , ggplot2, tidyverse . .

R , R Base . R , , ggplot2, tidyverse . , . R Base , R .

R , , , . , R . R **bookdown** package (Xie, 2020), .

, R , . R base R base , R , ggplot2, tidyverse , <https://r4ds.had.co.nz/> <https://rstudio-education.github.io/hopr/> , .



# Chapter 1

## R

R Ross Ihaka Robert Gentleman S , . R , R ,  
 , . S 1980 , AT&T , Rick Becker, John Chambers, Allan  
Wilks , 1990 , Insightful S , , Splus. R S ( Splus)  
 , R S , , S , SAS, SPSS .  
R (open-source, GNU General Public License), R {R}  
(R core-development team) , , {R} , .

### 1.1 R

R , Microsoft Window, Unix/Linux, Apple Mac OS , windows  
 . windows , , “ ” (User), , . R ,  
 , .

R (Reproducible Research, Dynamic Documentation),  
Rtools, RStudio, Tex System, Pandoc, Git . (PATH) , PATH  
 . R , .

R , R Rtools : 1. <http://www.r-project.org> 2. (Link)  
Download CRAN. 3. CRAN Mirrors (CRAN Mirrors), <https://cloud.r-project.org/> 4. Download R for Windows. 5. R for  
Windows base. 6. , Download R X.Y.Z for Windows, X.Y.Z R ,  
 . 7. , R-X.Y.Z-win.exe, . 8. , 64 . 9.  
base , Rtools, Rtoolsxx.exe. , . 10. Windows, Mac  
Linux , google Youtube , .

### 1.2 RStudio

{R} {R} . , , , {R} . {R} / , RStudio {R}  
, RStudio {R} , . <http://www.rstudio.com/>, Product

## 1.3

```
## demo
demo(graphics)
demo(image)
example(contour)
demo(persp)
example(persp3d)
demo(plotmath)
demo(Hershey)
install.packages("lattice") # install package
library("lattice")          # load package
demo(lattice)
```



```
example(wireframe)
install.packages("rgl")
library("rgl")
demo(rgl) # Interact using your mouse.
```

```
{R}      ,      ,      ,      .      ,      ,
{R}      ,      {R} ,      ,      0.5$ $1.0 ,      ,      ,
      .      ,      google,      . {R}      :

•      .
•      .
•      , Big5   utf-8 .
•      : / , , $, }, ], ).
•      / , , : , , Tab .
•      PDF   Web   copy .
•      .

      ,      ,      ,      ,      ,      .
```

## 1.4 Object

{R} **S** , (Object-Oriented Programming Language), {R} , ,  
(**object**). {R} (vector), (matrix), (array), (Lists), (data  
frames) (function) .

{R} , . , {R} , **s** **S** . , (**object**  
**name**) . ( ) , . , , (A-Z a-z), (0-9), /,  
., \_ (underscore) -, . .

{R} , , c, s, C, T, F , (reserved names). :

```
FALSE Inf NA NaN NULL TRUE
break else for function if in next repeat while
F T
c q s t C D I
diff mean pi range rank var
```

```
      ,      ,      ,      ,      ,      .
```

## 1.5

{R} , 2 , (expression), ,

```
1+2
log(x)
mean(x)
```

(assignment), ,

```
x <- 1+2
x = 4-5
```

```
{R}      ,      (prompt symbol),      > ( ).      ,
{R}      .      , {R}      ,      ( )      ,      {R}      . {R}
```

```
options(prompt = "R>")
```

```
> R>.
```

```
(assignment symbol) <- `` , , x <- 1 + 2, x`` ''
$(1 + 2)$ {R} , =( ) ** ** , x = 1 + 2, {R}
,=( ) , <- =, {R} <- '.
```

```
, , print(), , .
```

```
## assign
x <- 1 # assign object x
x      # show x
## [1] 1
print(x) # print()
## [1] 1
msg <- "hello"
msg      # show x
## [1] "hello"
```

```
, {R} , <Enter> , {R} , + ( ), + ,
, {R} . , <Enter> , {R} . , ; ( ) ,
. ,
```

```
## input at the same line, use ;
x <- 1 + 2; y <- 3 + 4
## input 2 lines separately
x <- 1 + 2
y <- 3 + 4
```

```
, { \; }, , (compound expression), .
```

```
{R} , , , (commands), # ( ) , , {R} ,
. , , ##, #.
```

```
## This is my R code
log(pi)
## [1] 1.145
## simple calculation
3+4 # calculator: two plus one
## [1] 7
```

```
{R} Console , , ↑ ( ) , , , <DEL>
. , {R} .
```

```
# This is my R code
x = 1 + 2 # one plus two
x
## [1] 3
x + 4
## [1] 7
x - 1
## [1] 2
```

## 1.6

```
{R} (object), , , , . {R} object() ls() {R}
.

## show objects
object() # show all objects
ls()     # show all objects
ls(x, y) # show x and y object

rm(), , ,

## delete objects: x.vec and y.vec
rm(x.vec, y.vec)

x.vec y.vec.
```

## 1.7

```
, {R} . , <Esc> . ,

for (i in 1:1000000) print (i) # press <Esc>

<Esc> .
```

## 1.8

```
{R} , (working directory). {R} (PATH) //
( , C://RData//) / ( , C:/RData/). Windows \\ ( , C:\\RData).
getwd(), . setwd(), . .

getwd() # show your current working directory
setwd("C:/RData/")
getwd()
## [1] "C:/RData"
```



```
Rlab00.r . , , {RStudio} Consol .
• {RStudio}, {RStudio} .
• File → New File, → R Notebook R Markdown, {RStudio}
  (template). (chunk) ```{r} ``` R . ,
```{r}
2.4*3.8
x.vec = rnorm(50)
y.vec = rnorm(50)
plot(x.vec, y.vec)
```

R Notebook R Markdown , , copy $\rightarrow$
paste word . , knit, , . R Notebook R
Markdown , knit .
```

## 1.10 Function

{R} (function), , , , , , {R} .  
(argument).

{R} (base) , , {R} (contribution) , {R} . ,  
mean(), var(), sd(), log() . R .

```
## function
## function c() = concatenate elements, return a vector x.vec
x.vec = c(1:5)
x.vec # show x.vec
## [1] 1 2 3 4 5
mean(x = x.vec) # function mean() calculate mean, return a scalar
## [1] 3
var(x = x.vec) # function mean() calculate variance
## [1] 2.5
log(x = x.vec) # take log for all elements in vector x.vec
## [1] 0.0000 0.6931 1.0986 1.3863 1.6094
```

(argument) , , (formals). , , (required  
argument), , (optional argument), (ellipsis argument)  
, , , {R} . , log() :

```
log(x, base = exp(1))
```

```
log() {R} , x , . base = exp(1) , ,
log() e , , , 2 , log(x, base = 2).
```

```
## log function
x.vec <- c(1, 2, 3, 4, 5)
log(x = x.vec)
## [1] 0.0000 0.6931 1.0986 1.3863 1.6094
```

```
log(x = x.vec, base = 2)
## [1] 0.000 1.000 1.585 2.000 2.322
```

## 1.11 Packages

```
. } (package). , {R} , (package),
, survival , , tidyverse .
{R} , . {R} , ( ) base {R}, {R} , {R}
, . , {R} , (contributed package).
{R} , . , (1) {RStudio} . {RStudio}
} . Packages → Install. , , tidyverse, MASS .
(2) install.packages() .
```

```
install.packages("PackageName", dependencies = TRUE)
```

```
PackageName` . , `Console` .
```

```
install.packages("survival")
```

```
library(survival)
```

```
{R} , , ,
```

- .
- library() require() .

```
library(), library() = loads a package, , require() =
tries to load a package, , error , , foo() paa ,
require() pbb, pbb coo() foo(), paa paa ,
coo() , error , error , my.obj, , .
```

```
library(package.name) , package.name function.name().
, , {::} package.name function.name() :
```

```
package.name::function.name()
```

```
package.name function.name().
```

```
ggplot2::ggplot()
```

```
ggplot2 ggplot().
```

## 1.12

```
{R} , Google {R} . {R} help.start(). Console
```

```
help.start()
```

```

    funName , {R} , help(funName), ?funName, help.search("funName"),
apropos("funName") . , mean() . {R} .

help(mean)
?mean
help.search("mean")
apropos("mean")

, args("funName").

```

## 1.13

```

{R} , {R} {R} , {R} , {R}
. sessionInfo() {R} .

sessionInfo()

version() {R} version[['version.string']], Sys.getlocale()
{R} LC_COLLATE=Chinese (Traditional)_Taiwan.950;LC_CTYPE=Chinese
(Traditional)_Taiwan.950;LC_MONETARY=Chinese (Traditional)_Taiwan.950;LC_NUMERIC=C;LC_TIME=C
(Traditional)_Taiwan.950. cp950 (big5) . Sys.timezone()
{R} Asia/Taipei. {R} (local time) NA, ,
, , Sys.setlocale("LC_TIME", "C"), UTC
(Universal Time, Coordinated). ISO .

# ->
Sys.setlocale("LC_CTYPE", "en_US.UTF-8")
# system("defaults write org.R-project.R force.LANG en_US.UTF-8") # linux/mac
# ->
Sys.setlocale(category = "LC_ALL", locale = "cht")
# system("defaults write org.R-project.R force.LANG zh_TW.UTF-8") # linux/mac

, {R} .

```





# Chapter 2

## Vector

`{R}` , `{R}` , (object), , (vector), (matrix),  
 (array), (Lists), (data frames) . `{R}` , , , . ,  
 . `{R}` , `{R}` .

### 2.1 Vector

`{R}` , (mode) . `{R}` (basic mode) numeric,  
 integer, logical, complex, character. .  
 (scalar), , (double) ({numerical vector}). ,  
`{R}` , (scalar) 1 , 1- , `{R}` (no dimension).  
 , `{R}` `x.vec <- c(1, 2, 3)`,  $1 \times 3$  ,  $3 \times 1$  , , `x.vec` / ,  
`x.vec` , `{R}` , .

### 2.2

`{R}` , (mode) . `{R}` (basic mode) numeric,  
 integer, logical, complex, character, `class()` .

- numeric, ( ), single double .

```
# numeric
x1 <- 10.1
x1
## [1] 10.1
class(x1)
## [1] "numeric"
x2 <- 10
x2
## [1] 10
```

```
class(x2)
## [1] "numeric"
is.numeric(x2)
## [1] TRUE
```

- **integer**, ( 1L, 2L, ...).

```
# integer
y1 <- 1L
y1
## [1] 1
class(y1)
## [1] "integer"
is.integer(y1)
## [1] TRUE
is.numeric(y1)
## [1] TRUE
```

- **logical**, (true or false), **TRUE (T)** **FALSE (F)** , 1 0  
T F.

```
# logic
yes_id <- TRUE
yes_id
## [1] TRUE
no_id <- FALSE
no_id
## [1] FALSE
class(no_id)
## [1] "logical"
is.logical(no_id)
## [1] TRUE
2 == 3
## [1] FALSE
2 != 3
## [1] TRUE
2 > 3
## [1] FALSE
2 <= 3
## [1] TRUE
4 >= 1
## [1] TRUE
TRUE + 5
## [1] 6
TRUE * 5
## [1] 5
FALSE * 5
```

```
## [1] 0
TRUE + FALSE
## [1] 1
TRUE * FALSE
## [1] 0
```

- complex, .

```
x = 3+5i
x
## [1] 3+5i
class(x)
## [1] "complex"
```

- character, , ("").

```
# character
ca <- "yes"
ca
## [1] "yes"
cb <- "this is a book."
cb
## [1] "this is a book."
class(cb)
## [1] "character"
is.character(cb)
## [1] TRUE
"abc" > "abd"
## [1] FALSE
"date" < "dates"
## [1] TRUE
```

- Date, POSIXct POSIXt , , Sys.Date()‘ .

```
Sys.Date()
## [1] "2020-09-16"
date1 <- as.Date("2020-09-17")
date1
## [1] "2020-09-17"
class(date1)
## [1] "Date"
as.numeric(date1)
## [1] 18522
date2 <- as.POSIXct("2020-09-17 18:30")
class(date2)
## [1] "POSIXct" "POSIXt"
as.numeric(date2)
```

```
## [1] 1.6e+09
```

### 2.2.1 c()

```
, c() . c() concatenate ( ), .

## c()
## numerical
x.vec <- c(1/1, 1/2, 1/3, 1/4, 1/5)
x.vec
## [1] 1.0000 0.5000 0.3333 0.2500 0.2000
## integer
x.vec <- c(1L, 2L, 3L)
x.vec
## [1] 1 2 3
## character
flavors.vec <- c("chocolate", "vanilla", "strawberry") # character
flavors.vec
## [1] "chocolate" "vanilla" "strawberry"
y.vec <- c("Hello", "What's your name?", "Your email?")
y.vec
## [1] "Hello" "What's your name?" "Your email?"
## logical
z.vec <- c(F, T, T, F, F)
z.vec
## [1] FALSE TRUE TRUE FALSE FALSE
## complex
x.complex.vec <- c(8+3i, 9+0i, 2+4i)
x.complex.vec
## [1] 8+3i 9+0i 2+4i
## numerical
x.vec <- c(1/1, 1/2, 1/3, 1/4, 1/5)
y.vec <- c(1, 2, 3, 4, 5)
z.vec <- c(x.vec, 11, 12, y.vec)
z.vec
## [1] 1.0000 0.5000 0.3333 0.2500 0.2000 11.0000 12.0000 1.0000 2.0000
## [10] 3.0000 4.0000 5.0000
```

## 2.3

{R} (basic operators), C (arithmetic operator),  
(relation/comparison operator), (logical operator). {R} (program-  
ming language), (if-else), (switch), (loop) (function) , ,

Table 2.1:

|      |   |
|------|---|
| -    | (Substraction, can be unary or binary)                  |
| +    | (Addition, can be unary or binary)                      |
| !    | (Unary not)   |
| *    | (Multiplication, binary)                                |
| /    | (Division, binary)                                      |
| ^    | (Exponentiation, binary)                                |
| %%   | (Modulus, binary)                                       |
| %/%  | (Integer divide, binary)                                |
| %*%  | (Matrix product, binary)                                |
| %o%  | (Outer product, binary)                                 |
| %x%  | Kronecker (Kronecker product, binary)                   |
| %in% | (Matching operator, binary, in model formulae: nesting) |
| <    | Less than, binary                                       |
| >    | Greater than, binary                                    |
| ==   | Equal to, binary  |
| !=   | Not equal to  |
| >=   | Greater than or equal to, binary                        |
| <=   | Less than or equal to, binary                           |
| &    | , (Logical AND, binary, vectorized)                     |
| &&   | , (Logical AND, binary, not vectorized)                 |
|      | , (Logical OR, binary, vectorized)                      |
|      | , (Logical OR, binary, not vectorized)                  |
| xor  | “ ”, , 1 TRUE   |

## 2.4

{R} (arithmetic operator) , +, -, !, \*, /, \^, %, %/%, %\*%, %o%, %x%, %in% : , , , , , .

```
## Arithmetic Operator
```

```
1 + 2
```

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

```
1 + 2 + 3
```

```
## [1] 6
```

```
3 * 7 * 2
```

```
## [1] 42
```

```
4/2
```

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

```
4/3
```

```
## [1] 1.333
```

```

2 * 3 + 4
## [1] 10
2 * (3 + 4)
## [1] 14
(3 + 11 * 2)/4
## [1] 6.25
#
x.complex <- (8+3i)+(1+2i)
x.complex
## [1] 9+5i
#
x.vec <- 1:5
y.vec <- c(-1, -2, 0, 2, 4)
z.vec <- c(2, 2, 3, 3, 4)
x.vec + y.vec
## [1] 0 0 3 6 9
x.vec - y.vec
## [1] 2 4 3 2 1
#
x.vec * 2
## [1] 2 4 6 8 10
x.vec * y.vec
## [1] -1 -4 0 8 20
x.vec/2
## [1] 0.5 1.0 1.5 2.0 2.5
x.vec/y.vec
## [1] -1.00 -1.00 Inf 2.00 1.25
#
x.vec^2
## [1] 1 4 9 16 25
x.vec^z.vec
## [1] 1 4 27 64 625
y.vec/2
## [1] -0.5 -1.0 0.0 1.0 2.0
y.vec/x.vec
## [1] -1.0 -1.0 0.0 0.5 0.8
#
y.vec %% 3 # modular arithmetic remainder
## [1] 2 1 0 2 1
y.vec %/% 3 # integer division
## [1] -1 -1 0 0 1
y.vec %/% x.vec
## [1] -1 -1 0 0 0

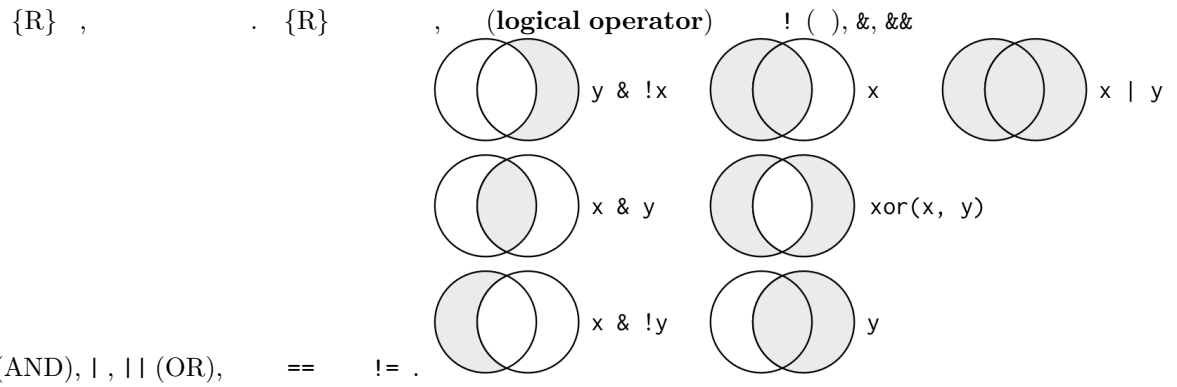
```

## 2.5

(logic vector)      TRUE, FALSE.      T    F.    {R}    ,  
 (relation/comparison operator)      .      <, <=, >, >=.

```
## Relation/Comparison Operator
x.vec <- 1:5
y.vec <- (x.vec > 2)
y.vec
## [1] FALSE FALSE TRUE TRUE TRUE
any(x.vec > 2)
## [1] TRUE
all(x.vec > 2)
## [1] FALSE
#
x.vec <- 1:5
y.vec <- c(0, 2, 4, 6, 8)
#
x.vec < 2
## [1] TRUE FALSE FALSE FALSE FALSE
x.vec <= 2
## [1] TRUE TRUE FALSE FALSE FALSE
x.vec == 2
## [1] FALSE TRUE FALSE FALSE FALSE
x.vec != 2
## [1] TRUE FALSE TRUE TRUE TRUE
#
x.vec < y.vec
## [1] FALSE FALSE TRUE TRUE TRUE
x.vec < (y.vec - 2)
## [1] FALSE FALSE FALSE FALSE TRUE
x.vec <= y.vec
## [1] FALSE TRUE TRUE TRUE TRUE
x.vec <= (y.vec - 2)
## [1] FALSE FALSE FALSE TRUE TRUE
#
x.vec == y.vec
## [1] FALSE TRUE FALSE FALSE FALSE
x.vec == (y.vec - 2)
## [1] FALSE FALSE FALSE TRUE FALSE
x.vec != y.vec
## [1] TRUE FALSE TRUE TRUE TRUE
x.vec != (y.vec - 2)
## [1] TRUE TRUE TRUE FALSE TRUE
```

## 2.6



```
## Logical Operator: AND OR XOR
x.vec <- 1:5
y.vec <- c(0, 2, 4, 6, 8)
(x.vec > 0) & (y.vec > 0) # return vector AND
## [1] FALSE TRUE TRUE TRUE TRUE
(x.vec > 0) && (y.vec > 0) # return scalar AND
## [1] FALSE
#
(x.vec > 0) & ((y.vec - 3) > 0) # return vector AND
## [1] FALSE FALSE TRUE TRUE TRUE
((x.vec-2) > 0) && ((y.vec - 3) > 0) # return scalar AND
## [1] FALSE
#
(x.vec > 0) & ((y.vec + 3) > 0) # return vector AND
## [1] TRUE TRUE TRUE TRUE TRUE
((x.vec-2) > 0) && ((y.vec + 3) > 0) # return scalar AND
## [1] FALSE
#
(x.vec > 0) | (y.vec > 0) # return vector OR
## [1] TRUE TRUE TRUE TRUE TRUE
((x.vec-2) > 0) | ((y.vec - 3) > 0)
## [1] FALSE FALSE TRUE TRUE TRUE
#
(x.vec > 0) || (y.vec > 0) # return scalar OR
## [1] TRUE
((x.vec-2) > 0) || ((y.vec - 3) > 0)
## [1] FALSE
#
(x.vec > 0) || ((y.vec + 3) > 0) # return scalar OR
## [1] TRUE
((x.vec-2) > 0) || ((y.vec + 3) > 0)
```



```
## [1] TRUE
#
xor((x.vec > 0), (y.vec > 0)) # return vector exclusive OR
## [1] TRUE FALSE FALSE FALSE FALSE
xor(((x.vec - 2) > 0), ((y.vec - 3) > 0))
## [1] FALSE FALSE FALSE FALSE FALSE
xor(((x.vec - 2) > 0), ((y.vec + 3) > 0))
## [1] TRUE TRUE FALSE FALSE FALSE
#
xx.vec <- (x.vec <= 3)
yy.vec <- (y.vec >= 4)
xx.vec
## [1] TRUE TRUE TRUE FALSE FALSE
yy.vec
## [1] FALSE FALSE TRUE TRUE TRUE
#
xx.vec && yy.vec
## [1] FALSE
xx.vec & yy.vec
## [1] FALSE FALSE TRUE FALSE FALSE
xx.vec || yy.vec
## [1] TRUE
xx.vec | yy.vec
## [1] TRUE TRUE TRUE TRUE TRUE
xor(xx.vec, yy.vec)
## [1] TRUE TRUE FALSE TRUE TRUE
```

## 2.7

```
, names() , unname() ,
names(x.vec) <- NULL .
## vector names
x.vec <- c(
  age = 50,
  chol = 220,
  dbp = 84,
  sbp = 132
) # directly
x.vec
## age chol dbp sbp
## 50 220 84 132
names(x.vec)
## [1] "age" "chol" "dbp" "sbp"
#
```

```

x.vec <- c(55, 236, 80, 140)
names(x.vec) <- c("age", "chol", "sbp", "dbp")
#
y.vec.name <- names(x.vec)
y.vec <- c(60, 214, 90, 144)
names(y.vec) <- y.vec.name
y.vec
## age chol sbp dbp
## 60 214 90 144

```

## 2.8 Inxex

(length) , (index) , (index) [i],  
 ( ) . , , , .

```

## Vector Indexing
## positive integer
x.vec <- 1:50
x.vec[7]
## [1] 7
x.vec[11:15]
## [1] 11 12 13 14 15
y.vec <- x.vec[11:15]
y.vec
## [1] 11 12 13 14 15
## negative integer
z.vec <- 6:10
z.vec[-c(2, 4)]
## [1] 6 8 10
## character string
fruit.vec <- c(5, 10, 1, 20)
fruit.vec
## [1] 5 10 1 20
names(fruit.vec) <- c("orange", "banana", "apple", "peach")
fruit.vec
## orange banana apple peach
##      5      10      1      20
lunch.vec <- fruit.vec[c("apple", "orange")]
lunch.vec
## apple orange
##      1      5
## logical index
x.vec <- c(NA, -2, -1, NA, 1, 2, NA) # NA = missing value
x.vec
## [1] NA -2 -1 NA 1 2 NA

```

```

y.vec <- x.vec[!is.na(x.vec)] # !is.na() = check missing value
y.vec
## [1] -2 -1  1  2
z.vec <- x.vec[x.vec > 0 & !is.na(x.vec)]
z.vec
## [1] 1 2
x.vec[x.vec < 0] # Note: NA
## [1] NA -2 -1 NA NA
y.vec[y.vec < 0]
## [1] -2 -1
z.vec[z.vec < 0]
## numeric(0)

```

## 2.9 ( ) Missing Values

, (missing value, incomplete data\*\*), R , , NA , (NA = Not Available), R NaN = Not a Number , NULL 0. (NA), (NA) . is.na(), is.nan() . . , na.omit(), na.fail(), na.exclude(), na.action() . complete.cases() . , R .

```

## missing value
z.vec <- c(1:2, NA)
is.na(z.vec)
## [1] FALSE FALSE TRUE
log(z.vec)
## [1] 0.0000 0.6931 NA
z.vec / 0
## [1] Inf Inf NA
0 / 0
## [1] NaN
Inf - Inf
## [1] NaN
#
is.na(z.vec)
## [1] FALSE FALSE TRUE
is.nan(z.vec)
## [1] FALSE FALSE FALSE
is.nan(0 / 0)
## [1] TRUE
#
x.vec <- c(1, 2, NA, 4, NA, 5, 6)
bad <- is.na(x.vec)
x.vec[!bad]
## [1] 1 2 4 5 6

```

```

#
x.vec <- c(1, 2, NA, 4, NA, 5, 6)
y.vec <- c("a", "b", NA, "d", NA, "f", "g")
good <- complete.cases(x.vec, y.vec)
good
## [1] TRUE TRUE FALSE TRUE FALSE TRUE TRUE
x.vec[good]
## [1] 1 2 4 5 6
y.vec[good]
## [1] "a" "b" "d" "f" "g"
#
data(airquality)
airquality[1:6,]
##      Ozone Solar.R Wind Temp Month Day
## 1      41      190  7.4   67     5    1
## 2      36      118  8.0   72     5    2
## 3      12      149 12.6   74     5    3
## 4      18      313 11.5   62     5    4
## 5      NA       NA 14.3   56     5    5
## 6      28       NA 14.9   66     5    6
good <- complete.cases(airquality)
airquality[good,][1:6,]
##      Ozone Solar.R Wind Temp Month Day
## 1      41      190  7.4   67     5    1
## 2      36      118  8.0   72     5    2
## 3      12      149 12.6   74     5    3
## 4      18      313 11.5   62     5    4
## 7      23      299  8.6   65     5    7
## 8      19       99 13.8   59     5    8

```

## 2.10 Factor

(factor) (categorical data),  
 (nominal variable) (ordinal variable),  
 1, 0; 1 = , 2 = ,  
 3 = , 4 = , 5 = , 2 , 0 1, ,  
 (dichotomous variable, binary variable),  
 4 : , , , I, II, III, IV 4 . 1, 2, 3, 4, ...  
 , , , \*\* \*, \*\* \*, \*\* \* {R} (factor) ,  
 , , , {R} (levels), ,  
 , {R} , " , {R} .  
 {R} , factor() .

```
factor(x = character(), levels, labels = levels,
       exclude = NA, ordered = is.ordered(x), nmax = NA)
```

- x , , {R} .
- levels .
- labels .
- exclude = NA .
- ordered = is.ordered(x) , .
- nmax = NA .

```
## factor()
sex <- c("male", "female", "male", "male", "female")
sex
## [1] "male" "female" "male" "male" "female"
class(sex)
## [1] "character"
sex <- factor(sex)
sex
## [1] male female male male female
## Levels: female male
class(sex)
## [1] "factor"
## factor() + levels
sex <- c("male", "female", "male", "male", "female")
sex <- factor(sex, levels = c("female", "male"))
sex
## [1] male female male male female
## Levels: female male
## factor() + levels + labels
x.chr = c("male", "male", "female", "female")
factor(x.chr, levels = c("male", "female", "bisex"))
## [1] male male female female
## Levels: male female bisex
factor(x.chr, levels = c("male", "female", "bisex"),
      labels = c("m", "f", "b"))
## [1] m m f f
## Levels: m f b
## factor() + exclude
## factor() + exclude
pain <- c("none", "mild", "moderate", "severe", NA)
factor(pain) # NA is NOT a level.
## [1] none mild moderate severe <NA>
## Levels: mild moderate none severe
factor(pain, exclude = NA) # NA is NOT a level.
## [1] none mild moderate severe <NA>
```

```
## Levels: mild moderate none severe
factor(pain, exclude = c(NA)) # NA is NOT a level.
## [1] none      mild      moderate severe  <NA>
## Levels: mild moderate none severe
factor(pain, exclude = NULL) # NA is a level.
## [1] none      mild      moderate severe  <NA>
## Levels: mild moderate none severe <NA>
factor(pain, exclude = "mild") # NA is a level.
## [1] none      <NA>      moderate severe  <NA>
## Levels: moderate none severe <NA>
pain <- factor(pain, exclude = c("mild", NA))
pain # mild and NA are NOT levels.
## [1] none      <NA>      moderate severe  <NA>
## Levels: moderate none severe
```

```
{R}  factor()      (unordered factor),      (nominal variable),
      (level),      ,      {R}      ,      ,      levels()      ;
      levels()      , {R}      ,      levels()      ,      .

      ,
      (reference level),
      (contrast comparison).  relevel(),      .
```

```
## unordered
## level()
gender <- c("M", "F", "M", "M", "F")
gender <- factor(gender)
gender
## [1] M F M M F
## Levels: F M
levels(gender)
## [1] "F" "M"
levels(gender) <- c("Female", "Male")
gender
## [1] Male  Female Male  Male  Female
## Levels: Female Male
hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
hypertension <- factor(hypertension)
hypertension
## [1] Lo  Mod Hi  Mod Lo  Hi  Lo
## Levels: Hi Lo Mod
# relevel()
relevel(hypertension, ref = "Lo") # reset a reference level
## [1] Lo  Mod Hi  Mod Lo  Hi  Lo
## Levels: Lo Hi Mod
```

```
as.integer()      ,      1      ,      ,      .
```

```

## convert to numerical values
hypertension <- c("Lo", "Mod", "Hi", "Mod", "Lo", "Hi", "Lo")
hypertension <- factor(hypertension)
levels(hypertension)
## [1] "Hi" "Lo" "Mod"
hypertension
## [1] Lo  Mod Hi  Mod Lo  Hi  Lo
## Levels: Hi Lo Mod
as.integer(hypertension)
## [1] 2 3 1 3 2 1 2
#
levels(hypertension) <- list("Low" = "Lo",
                             "Moderate" = "Mod",
                             "High" = "Hi")

hypertension
## [1] Low      Moderate High      Moderate Low      High      Low
## Levels: Low Moderate High
as.integer(hypertension)
## [1] 1 2 3 2 1 3 1
#
## convert to numerical values
pain <- c(7, 8, 6, 6, 8, 7)
pain <- factor(pain)
pain
## [1] 7 8 6 6 8 7
## Levels: 6 7 8
as.integer(pain)
## [1] 2 3 1 1 3 2
pain.chr = as.character(pain)
pain.chr
## [1] "7" "8" "6" "6" "8" "7"
pain.num = as.integer(pain.chr)
pain.num
## [1] 7 8 6 6 8 7

```





# Chapter 3

{R} , {R} , (object), (vector), (matrix),  
 (array), (Lists), (data frames) . {R} , , ,  
 , {R} , (matrix), (array), (Lists),  
 (data frames) .

## 3.1 Matrix

(matrix) ( , mode) 2- (2-dimension) , (dimension) ,  
 dim() . , 2- (array).

### 3.1.1 matrix()

, × ( × ), , matrix().

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

- nrow = r , (row numbers).
- ncol = c , (column number).
- byrow = FALSE: {R} , ( ) (column) . , byrow = TRUE.
- dimnames = obj.list .

dim() .

```
## numeric
x.mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2) # one row first
x.mat
##      [,1] [,2] [,3]
## [1,]  1   3   5
## [2,]  2   4   6
```

```

dim(x.mat)
## [1] 2 3
y.mat <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2)
y.mat
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6
z.mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, byrow = T)
z.mat
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
p.mat <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2, byrow = T)
p.mat
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
## [3,]    5    6
w.mat <- matrix(c(1:18), nrow = 3)
w.mat
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    4    7   10   13   16
## [2,]    2    5    8   11   14   17
## [3,]    3    6    9   12   15   18
dim(y.mat)
## [1] 3 2
# character
x.vec <- c("a", "b", "c", "d", "e", "f")
x.vec
## [1] "a" "b" "c" "d" "e" "f"
y.mat <- matrix(x.vec, nrow = 2, ncol = 3) # byrow = F
y.mat
##      [,1] [,2] [,3]
## [1,] "a"  "c"  "e"
## [2,] "b"  "d"  "f"
y.mat <- matrix(x.vec,
                nrow = 2,
                ncol = 3,
                byrow = T)
y.mat
##      [,1] [,2] [,3]
## [1,] "a"  "b"  "c"
## [2,] "d"  "e"  "f"
dim(y.mat)

```

```
## [1] 2 3
# dim
m.vec.mat <- 1:10
dim(m.vec.mat) <- c(2, 5)
m.vec.mat
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    3    5    7    9
## [2,]    2    4    6    8   10
dim(m.vec.mat)
## [1] 2 5
```

### 3.1.2 dimnames()

, (column name) (row name), dimnames() .  
 dimnames() matrix . (row name) (column name),  
 rownames() colnames().

```
# dimnames
x.mat <- matrix(1:6, nrow = 2, ncol = 3)
dimnames(x.mat) <- list(c("A1", "A2"),
                        c("B1", "B2", "B3"))

x.mat
##      B1 B2 B3
## A1  1  3  5
## A2  2  4  6
dim(x.mat)
## [1] 2 3
dimnames(x.mat)
## [[1]]
## [1] "A1" "A2"
##
## [[2]]
## [1] "B1" "B2" "B3"
rownames(x.mat)
## [1] "A1" "A2"
colnames(x.mat)
## [1] "B1" "B2" "B3"
#
m.mat <- matrix(
  c(1, 2, 3, 11, 12, 13),
  nrow = 2,
  ncol = 3,
  byrow = TRUE,
  dimnames = list(c("row1", "row2"),
                  c("C1", "C2", "C3"))
)
```

```

m.mat
##      C1 C2 C3
## row1  1  2  3
## row2 11 12 13
dim(m.mat)
## [1] 2 3
dimnames(m.mat)
## [[1]]
## [1] "row1" "row2"
##
## [[2]]
## [1] "C1" "C2" "C3"
rownames(m.mat)
## [1] "row1" "row2"
colnames(m.mat)
## [1] "C1" "C2" "C3"

```

## 3.2 Matrix Index

$(\text{index})$  ,  $2-$  ,  $\{R\}$  ,  $[m, ]$  ,  $n$  (column) .  
 $\text{matrix.name}[i, j]$   $[i, j]$  ;  $\text{matrix.name}[i, ]$   $i$  (ith row),  
 $\text{matrix.name}[ , j]$   $j$  (ith column).  $\{R\}$  ,  $[m, ]$  ,  
 $m$  (row) ;  $[ , n]$  ,  $n$  (column) .

```

## matrix index
x.mat <- matrix(c(1:12), 3, 4)
x.mat
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11
## [3,]    3    6    9   12
x.mat[2, 3] <- 30
x.mat
##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5   30   11
## [3,]    3    6    9   12
x.mat[2,]
## [1]  2  5 30 11
x.mat[, 3]
## [1]  7 30  9
x.mat[c(1, 3), c(2, 4)]
##      [,1] [,2]
## [1,]    4   10
## [2,]    6   12

```

```
#
m.mat <- matrix(
  c(1, 2, 3, 11, 12, 13),
  nrow = 2,
  ncol = 3,
  byrow = TRUE,
  dimnames = list(c("row1", "row2"),
                  c("C1", "C2", "C3"))
)
m.mat
##      C1 C2 C3
## row1  1  2  3
## row2 11 12 13
m.mat[, c("C1", "C2")]
##      C1 C2
## row1  1  2
## row2 11 12
m.mat[c("row2"),]
## C1 C2 C3
## 11 12 13
m.mat[c("row1"), c("C1", "C3")]
## C1 C3
##  1  3

1 1 , , drop = FALSE.
```

```
## dimension reduction
x.mat <- matrix(1:8, 2, 4)
x.mat[1,] # reduces to a vector
## [1] 1 3 5 7
x.mat[1, , drop = FALSE] # remains as a matrix
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
```

### 3.2.1 : rbind() cbind()

$\{R\}$  (no dimension),  $1 \times k$  / ,  $k \times 1$  / , ,  
 / , ,  $\{R\}$  / ,  $1 \times k$   $\{R\}$  , , ,  
 , ,  $1-k$  ,  $\{R\}$   $1 \times k$   $k \times 1$  , , ,  
 (row number) (column number) , recycle .

```
## matrix cbind() and rbind()
x.vec <- c(1, 2, 3)
y.vec <- c(8, 9, 10)
rbind(x.vec, y.vec) # vector as row vector
##      [,1] [,2] [,3]
```

```
## x.vec    1    2    3
## y.vec    8    9   10
cbind(x.vec, y.vec) # vector as col vector
##          x.vec y.vec
## [1,]      1     8
## [2,]      2     9
## [3,]      3    10
#
x.mat <- matrix(c(11:16), 2, 3)
rbind(x.mat, x.vec) # vector as row vector
##          [,1] [,2] [,3]
##          11  13  15
##          12  14  16
## x.vec    1    2    3
cbind(x.mat, y.vec) # warning
## Warning in cbind(x.mat, y.vec): number of rows of result is not a multiple of
## vector length (arg 2)
##          y.vec
## [1,] 11 13 15     8
## [2,] 12 14 16     9
#
x.vec <- c(1, 2)
y.vec <- c(8, 9)
rbind(x.vec, y.vec) # vector as row vector
##          [,1] [,2]
## x.vec    1    2
## y.vec    8    9
cbind(x.vec, y.vec) # vector as col vector
##          x.vec y.vec
## [1,]      1     8
## [2,]      2     9
#
x.mat <- matrix(c(11:14), 2, 2)
z.mat <- rbind(x.mat, x.vec) # vector as row vector
z.mat
##          [,1] [,2]
##          11  13
##          12  14
## x.vec    1    2
cbind(x.mat, y.vec) # vector as col vector
##          y.vec
## [1,] 11 13     8
## [2,] 12 14     9
rbind(z.mat, y.vec) # vector as row vector
##          [,1] [,2]
```

```
##          11  13
##          12  14
## x.vec    1   2
## y.vec    8   9
cbind(z.mat, y.vec) # warning
## Warning in cbind(z.mat, y.vec): number of rows of result is not a multiple of
## vector length (arg 2)
##          y.vec
##          11 13   8
##          12 14   9
## x.vec    1  2   8
```

### 3.2.2 Array

(array) (mode)  $p$  ,  $p$  . array() .  
 {R} 3-  $m \times n \times k$ , [m, , ] ,  $m$  (row) ; [ , n, ]  
 ,  $n$  (column) , . [ , , k] 3- 1, 2- .  
 , , dimnames() . dimnames() array .  
 (index) , , ([i, j, k]). 1 ( , row name) { 2  
 ( , column name\*\*), rownames() colnames().

```
## array()
a.vec <- 1:24
a.vec
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
b.array <- array(a.vec, dim = c(4, 3, 2),
  dimnames = list(c("x1", "x2", "x3", "x4"),
    c("y1", "y2", "y3"),
    c("z1", "z2")))
b.array
## , , z1
##
## y1 y2 y3
## x1 1 5 9
## x2 2 6 10
## x3 3 7 11
## x4 4 8 12
##
## , , z2
##
## y1 y2 y3
## x1 13 17 21
## x2 14 18 22
## x3 15 19 23
## x4 16 20 24
```

```

mode(b.array)
## [1] "numeric"
dim(b.array)
## [1] 4 3 2
length(b.array)
## [1] 24
dimnames(b.array)
## [[1]]
## [1] "x1" "x2" "x3" "x4"
##
## [[2]]
## [1] "y1" "y2" "y3"
##
## [[3]]
## [1] "z1" "z2"
rownames(b.array)
## [1] "x1" "x2" "x3" "x4"
colnames(b.array)
## [1] "y1" "y2" "y3"
# array index
a.vec
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
b.array <- array(a.vec, dim = c(4, 3, 2),
                 dimnames = list(c("x1", "x2", "x3", "x4"),
                                c("y1", "y2", "y3"),
                                c("z1", "z2")))

b.array
## , , z1
##
##      y1 y2 y3
## x1  1  5  9
## x2  2  6 10
## x3  3  7 11
## x4  4  8 12
##
## , , z2
##
##      y1 y2 y3
## x1 13 17 21
## x2 14 18 22
## x3 15 19 23
## x4 16 20 24
b.array[3, 2, 1]
## [1] 7
b.array[4, 3, 2]

```



```
## [1] 24
b.array[2, c(1, 3), 1]
## y1 y3
## 2 10
b.array[3, c(2, 3), 1]
## y2 y3
## 7 11
b.array[2, ,]
## z1 z2
## y1 2 14
## y2 6 18
## y3 10 22
b.array[, 2,]
## z1 z2
## x1 5 17
## x2 6 18
## x3 7 19
## x4 8 20
b.array[, , 2]
## y1 y2 y3
## x1 13 17 21
## x2 14 18 22
## x3 15 19 23
## x4 16 20 24
```

### 3.3 List

(list) , (mode) (complex mode) ,  
 “ ’”, (component), (order sequence), , .  
 , .

#### 3.3.1 list()

list() . {R} , , , , , .

```
## list()
## list w/o component names
x.vec <- 1:4
y.vec <- c("Male", "Female")
z.mat <- matrix(1:9, nrow = 3, ncol = 3)
xyz.list <- list(x.vec, y.vec, z.mat)
xyz.list
## [[1]]
## [1] 1 2 3 4
##
```

```

## [[2]]
## [1] "Male"   "Female"
##
## [[3]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
mode(xyz.list)
## [1] "list"
length(xyz.list)
## [1] 3
dim(xyz.list)
## NULL
names(xyz.list)
## NULL
class(xyz.list)
## [1] "list"
## list w/ component names
x.num <- c(1, 3, 6)
y.str <- c("chocolate", "vanilla", "strawberry")
xy.list <- list(x.num.var = x.num, y.str.var = y.str)
xy.list
## $x.num.var
## [1] 1 3 6
##
## $y.str.var
## [1] "chocolate" "vanilla"   "strawberry"
# list = data matrix
id.vec <- c(1, 2, 3, 4)
age.vec <- c(35, 55, 45, 25)
sex.vec <- c("Male", "Male", "Female", "Female")
disease.vec <- c("Yes", "No", "No", "Yes")
x.list <- list(
  id = id.vec,
  age = age.vec,
  sex = sex.vec,
  disease = disease.vec
)
x.list
## $id
## [1] 1 2 3 4
##
## $age
## [1] 35 55 45 25

```

```
##
## $sex
## [1] "Male" "Male" "Female" "Female"
##
## $disease
## [1] "Yes" "No" "No" "Yes"
```

### 3.3.2 List Index

```
, List.Name , list i.number ,
List.Name[[3]]. , [[i.number]] [i.number] .
```

```
## list index
## list w/o component names
x.vec <- 1:4
y.vec <- c("Male", "Female")
z.mat <- matrix(1:9, nrow = 3, ncol = 3)
xyz.list <- list(x.vec, y.vec, z.mat)
xyz.list
## [[1]]
## [1] 1 2 3 4
##
## [[2]]
## [1] "Male" "Female"
##
## [[3]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
xyz.list[1]
## [[1]]
## [1] 1 2 3 4
xyz.list[[1]]
## [1] 1 2 3 4
xyz.list[2]
## [[1]]
## [1] "Male" "Female"
xyz.list[[3]]
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
xyz.list[3]
## [[1]]
##      [,1] [,2] [,3]
```

```
## [1,] 1 4 7
## [2,] 2 5 8
## [3,] 3 6 9
```

```
(component) comp.name, List.Name$comp.name ,
List.Name[[comp.name]] . List.Name$comp.name List.Name[[comp.name]]
, List.Name$comp.name . [[i.number]] , $ .
```

```
# list w/ component names
x.vec <- 1:4
y.vec <- c("Male", "Female")
z.mat <- matrix(1:9, nrow = 3, ncol = 3)
xyz.list <- list(class = x.vec,
                 gender = y.vec,
                 score = z.mat)

xyz.list
## $class
## [1] 1 2 3 4
##
## $gender
## [1] "Male" "Female"
##
## $score
##      [,1] [,2] [,3]
## [1,] 1 4 7
## [2,] 2 5 8
## [3,] 3 6 9
xyz.list$class
## [1] 1 2 3 4
xyz.list[["class"]]
## [1] 1 2 3 4
xyz.list[["class"]][2]
## [1] 2
#
xyz.list$gender
## [1] "Male" "Female"
xyz.list[["gender"]][1]
## [1] "Male"
#
xyz.list$score
##      [,1] [,2] [,3]
## [1,] 1 4 7
## [2,] 2 5 8
## [3,] 3 6 9
xyz.list[["score"]][2, 3]
```

```
## [1] 8
```

### 3.4 Data Frame

`(data frame)` is a `list()` of `{R}` objects, where each object is a `{R}` object.  
`(data matrix)` is a `{R}` object, where each object is a `{R}` object.

### 3.5 data.frame()

`{R}` `data.frame()` is a `{R}` object, where each object is a `{R}` object.

```
## data frame
id.vec <- c(1, 2, 3, 4)
age.vec <- c(35, 55, 45, 25)
sex.vec <- c("Male", "Male", "Female", "Female")
disease.vec <- c("Yes", "No", "No", "Yes")
x.df <- data.frame(
  id = id.vec,
  age = age.vec,
  sex = sex.vec,
  disease = disease.vec
)
mode(x.df)
## [1] "list"
class(x.df)
## [1] "data.frame"
x.df
##   id age  sex disease
## 1  1 35  Male    Yes
## 2  2 55  Male    No
## 3  3 45 Female    No
## 4  4 25 Female    Yes
x.df$age
## [1] 35 55 45 25
x.df$disease
## [1] "Yes" "No" "No" "Yes"
```

#### 3.5.1 Data Frame Index

`(index)` is a `{R}` object, where each object is a `{R}` object.  
`dataframe.name[i, j]` is a `{R}` object, where each object is a `{R}` object.  
`dataframe.name[i, ]` is a `{R}` object, where each object is a `{R}` object.  
`dataframe.name[, j]` is a `{R}` object, where each object is a `{R}` object.

```

        (index) ,      data.Name ,      'i.number'
( ), data.Name[[3]]. , [[i.number]] [i.number] . variable.name,
  dataframe.Name$variable.name , dataframe.Name[[variable.name]]
. dataframe.Name$variable.name dataframe.Name[[variable.name]]
  [i] , dataframe.Name$variable.name . [[i.number]] , $
.

```

```

## data frame index
data(Puromycin)
Puromycin
##      conc rate      state
## 1  0.02   76   treated
## 2  0.02   47   treated
## 3  0.06   97   treated
## 4  0.06  107   treated
## 5  0.11  123   treated
## 6  0.11  139   treated
## 7  0.22  159   treated
## 8  0.22  152   treated
## 9  0.56  191   treated
## 10 0.56  201   treated
## 11 1.10  207   treated
## 12 1.10  200   treated
## 13 0.02   67 untreated
## 14 0.02   51 untreated
## 15 0.06   84 untreated
## 16 0.06   86 untreated
## 17 0.11   98 untreated
## 18 0.11  115 untreated
## 19 0.22  131 untreated
## 20 0.22  124 untreated
## 21 0.56  144 untreated
## 22 0.56  158 untreated
## 23 1.10  160 untreated
Puromycin$rate
## [1] 76 47 97 107 123 139 159 152 191 201 207 200 67 51 84 86 98 115 131
## [20] 124 144 158 160
Puromycin$state
## [1] treated treated treated treated treated treated treated
## [8] treated treated treated treated treated untreated untreated
## [15] untreated untreated untreated untreated untreated untreated untreated
## [22] untreated untreated
## Levels: treated untreated
Puromycin[1]
##      conc
## 1  0.02

```

```

## 2 0.02
## 3 0.06
## 4 0.06
## 5 0.11
## 6 0.11
## 7 0.22
## 8 0.22
## 9 0.56
## 10 0.56
## 11 1.10
## 12 1.10
## 13 0.02
## 14 0.02
## 15 0.06
## 16 0.06
## 17 0.11
## 18 0.11
## 19 0.22
## 20 0.22
## 21 0.56
## 22 0.56
## 23 1.10
Puromycin[1][[1]]
## [1] 0.02 0.02 0.06 0.06 0.11 0.11 0.22 0.22 0.56 0.56 1.10 1.10 0.02 0.02 0.06
## [16] 0.06 0.11 0.11 0.22 0.22 0.56 0.56 1.10
Puromycin$state[1:3]
## [1] treated treated treated
## Levels: treated untreated
Puromycin[1:3, 1:2]
##   conc rate
## 1 0.02 76
## 2 0.02 47
## 3 0.06 97

```





# Chapter 4

{R} , , (vector), (matrix), (array), (Lists), (data frames) . , {R} , , , {R} , {R} .  
{R} ASCII , {R} , SAS, SPSS, STATA, EXCEL, , web open data (XML, HTML JSON), image, texts, stock market, social media . , .  
{R} , , , , {R} , , {R} , {R} ASCII , ASCII .

## 4.1

, , {R} (data frame). SAS, STATA dataset . , (cross table).  
{R} , , , , , , (mode) .

Table 4.1: DMTKRtabsep.txt DMTKRblanksep.txt:

| No | age | sex | DM | DMyr | preAC | prePC | postAC | postPC | Med | SIDE | PREKS | POSKS | ABS | I |
|----|-----|-----|----|------|-------|-------|--------|--------|-----|------|-------|-------|-----|---|
| 1  | 67  | 0   | 0  | 10   | 120   | 160   | 140    | 180    | 0   | 0    | 56    | 92    | 1   | 0 |
| 2  | 67  | 0   | 0  | 11   | 100   | 150   | 150    | 220    | 0   | 1    | 62    | 62    | 0   | 1 |
| 3  | 72  | 1   | 0  | 4    | 150   | 200   | 120    | 150    | 2   | 0    | 60    | 94    | 1   | 0 |
| 4  | 82  | 1   | 0  | 8    | 150   | 200   | 160    | 250    | 0   | 1    | 47    | 90    | 1   | 0 |
| 5  | 73  | 1   | 0  | 3    | 85    | 110   | 140    | 200    | 0   | 0    | 44    | 88    | 0   | 0 |

, .

- `1 ( , row), (variable names)`
- `1 (row), (column label), 2 (row) .`
- `1 ( , row).`
- `1 ( , column) (label, identification), (row label).`
- `( ) , , .`
- `(column) (row label).`
- `, , , underscore _.`
- `, , , , , .`
- `, , , , , , .`
- `, {R} , , .`

## 4.2 ASCII R :

- ASCII , ASCII (raw data) ,  
 {R} ASCII , {R} ASCII .
- {R} (data frame) , {R} `read.table()` `read.csv()` ,  
 . , , `scan()` , {R} {R} , ASCII  
 , :
- , , .
  - (the first row) (variable names) , (column name)  
 (column label).
  - (the first column) (row label) (row name).
  - (row), .
  - () (blank space) , 'Tab' .
  - , , , .
  - ASCII , .dat, .prn .txt.
  - , ( ) ASCII , comma-separated-variable format CSV  
 format, .csv .
  - (variable name) , , . ( , dot), \_ (underscore). , .  
 (observed value).
  - Tab , 2 ASCII . , CSV format.

```
# DMTKRblanksep.txt = " " single space separate
Rblanksep.df = read.table("C:/RData/DMTKRblanksep.txt",
```

```

        header = TRUE,
        row.names = NULL,
        dec = ".")

head(Rblanksep.df)
##   No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
## 1  1  67  0  0  10  120  160  140  180  0  0  56  92  1  0
## 2  2  67  0  0  11  100  150  150  220  0  1  62  62  0  1
## 3  3  72  1  0  4  150  200  120  150  2  0  60  94  1  0
## 4  4  82  1  0  8  150  200  160  250  0  1  47  90  1  0
## 5  5  73  1  0  3  85  110  140  200  0  0  44  88  0  0
## 6  6  76  0  0  1  120  150  120  200  0  1  52  94  1  0

str(Rblanksep.df)
## 'data.frame':      78 obs. of  15 variables:
##  $ No      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age     : int  67 67 72 82 73 76 76 77 64 64 ...
##  $ sex     : int  0 0 1 1 1 0 0 0 0 0 ...
##  $ DM      : int  0 0 0 0 0 0 0 1 0 0 ...
##  $ DMyr    : int  10 11 4 8 3 1 1 35 5 5 ...
##  $ preAC   : int  120 100 150 150 85 120 120 200 130 130 ...
##  $ prePC   : int  160 150 200 200 110 150 150 250 180 180 ...
##  $ postAC  : int  140 150 120 160 140 120 120 230 100 100 ...
##  $ postPC  : int  180 220 150 250 200 200 200 300 150 150 ...
##  $ Med     : int  0 0 2 0 0 0 0 1 0 0 ...
##  $ SIDE    : int  0 1 0 1 0 1 0 1 0 1 ...
##  $ PREKS   : int  56 62 60 47 44 52 48 42 40 45 ...
##  $ POSKS   : int  92 62 94 90 88 94 96 90 94 96 ...
##  $ ABS     : int  1 0 1 1 0 1 0 1 1 0 ...
##  $ INFECT  : int  0 1 0 0 0 0 0 0 0 0 ...
## complete read.table
## DMTKRblanksep.txt = " " single space separate
Rblanksep.df = read.table("C:/RData/DMTKRblanksep.txt",
        header = TRUE,
        sep = " ",
        quote = "\"\"",
        dec = ".",
        row.names = NULL,
        # col.names,
        as.is = TRUE,
        # as.is = !stringsAsFactors,
        na.strings = c(".", "NA"))

head(Rblanksep.df)
##   No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
## 1  1  67  0  0  10  120  160  140  180  0  0  56  92  1  0
## 2  2  67  0  0  11  100  150  150  220  0  1  62  62  0  1
## 3  3  72  1  0  4  150  200  120  150  2  0  60  94  1  0

```

```
## 4 4 82 1 0 8 150 200 160 250 0 1 47 90 1 0
## 5 5 73 1 0 3 85 110 140 200 0 0 44 88 0 0
## 6 6 76 0 0 1 120 150 120 200 0 1 52 94 1 0
str(Rblanksep.df)
## 'data.frame':      78 obs. of  15 variables:
## $ No      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ age     : int  67 67 72 82 73 76 76 77 64 64 ...
## $ sex     : int  0 0 1 1 1 0 0 0 0 0 ...
## $ DM      : int  0 0 0 0 0 0 0 1 0 0 ...
## $ DMyr    : int  10 11 4 8 3 1 1 35 5 5 ...
## $ preAC   : int  120 100 150 150 85 120 120 200 130 130 ...
## $ prePC   : int  160 150 200 200 110 150 150 250 180 180 ...
## $ postAC  : int  140 150 120 160 140 120 120 230 100 100 ...
## $ postPC  : int  180 220 150 250 200 200 200 300 150 150 ...
## $ Med     : int  0 0 2 0 0 0 0 1 0 0 ...
## $ SIDE    : int  0 1 0 1 0 1 0 1 0 1 ...
## $ PREKS   : int  56 62 60 47 44 52 48 42 40 45 ...
## $ POSKS   : int  92 62 94 90 88 94 96 90 94 96 ...
## $ ABS     : int  1 0 1 1 0 1 0 1 1 0 ...
## $ INFECT  : int  0 1 0 0 0 0 0 0 0 0 ...
```

### 4.3 ASCII R :

ASCII , (comma) ASCII , csv format (comma-separated-variable format), .csv. read.table() ,  
 ,. read.csv() , .

```
## read data file: DMTKRcsv.csv
read_table.df <- read.table("C:/RData/DMTKRcsv.csv",
                           header = TRUE,
                           row.names = NULL,
                           sep = ",",
                           dec = ".")

head(read_table.df, n = 3)
## No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
## 1 1 67 0 0 10 120 160 140 180 0 0 56 92 1 0
## 2 2 67 0 0 11 100 150 150 220 0 1 62 62 0 1
## 3 3 72 1 0 4 150 200 120 150 2 0 60 94 1 0
# simple one
read_csv.df <- read.csv("C:/RData/DMTKRcsv.csv")
head(read_csv.df, n = 3)
## No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
## 1 1 67 0 0 10 120 160 140 180 0 0 56 92 1 0
## 2 2 67 0 0 11 100 150 150 220 0 1 62 62 0 1
## 3 3 72 1 0 4 150 200 120 150 2 0 60 94 1 0
```

```
#
read_csv.df <- read.csv("C:/RData/DMTKRcsv.csv",
                        header = TRUE,
                        row.names = NULL,
                        sep = ",",
                        dec = ".")

head(read_csv.df, n = 3)
##   No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
## 1  1  67   0  0  10  120  160  140  180  0  0  56  92  1  0
## 2  2  67   0  0  11  100  150  150  220  0  1  62  62  0  1
## 3  3  72   1  0   4  150  200  120  150  2  0  60  94  1  0
```

## 4.4 R

```
{R} , (contributed packages) , data() {R} ,
library(help = "datasets") {R} .
```

```
data(package = "package.name") package.name , ,
data(data.name) {R} data.name , data(package.data.name,
package = "package.name") package.name , pack.data.name .
```

```
# data() # check names of datasets
data(Orange) # use {R} build-in dataset = Orange
# help(Orange)
head(Orange)
## Grouped Data: circumference ~ age | Tree
##   Tree age circumference
## 1    1  118           30
## 2    1  484           58
## 3    1  664           87
## 4    1 1004          115
## 5    1 1231          120
## 6    1 1372          142
#
library(MASS)
# help(package = MASS)
# data(package = "MASS") # check MASS package data set
data(VA, package = "MASS") # use MASS package dataset = VA
# help(VA)
head(VA)
##   stime status treat age Karn diag.time cell prior
## 1    72      1    1  69  60      7    1    0
## 2   411      1    1  64  70      5    1   10
## 3   228      1    1  38  60      3    1    0
## 4   126      1    1  63  60      9    1   10
```

```
## 5 118 1 1 65 70 11 1 10
## 6 10 1 1 49 20 5 1 0
```

## 4.5 {R}

```
{R} , . write.table() write.csv() .
```

- `x = {R}`
- `file =`
- `append = FALSE`
- `quote = "\""`
- `sep = " "`
- `eol = "\n"`
- `na = NA NA`
- `dec = '.'`
- `row.names = TRUE` row names
- `col.names = TRUE` (column names)
- `qmethod = c("escape", "double")`
- `fileEncoding = ""`

```
write.csv() write.table() , sep = ",".
```

## 4.6 {R}

```
saveRDS() {R} . readRDS() {R} . {R} , save()
data frame {R} . readRDS() , . {R}
load() , .
```

```
## saveRDS() and save()
x <- c(1:5)
saveRDS(x, file = "C:/RData/x.Rds")
save(x, file = "C:/RData/x.Rda") # working directory
## readRDS()
new_x <- readRDS(file = "C:/RData/x.Rds")
new_x
## [1] 1 2 3 4 5
## load() -- note the result
new_x <- load(file = "C:/RData/x.Rda")
new_x
## [1] "x"
x
## [1] 1 2 3 4 5
```

# Chapter 5

, \ Leland Wilkinson (1999), **The Grammar of Graphics**.  
{R} , {R} , , .  
, (interactive) {R} , .  
{R} , :  
• (high-level plotting functions): , , , .  
• (low-level plotting functions): , , .  
, , **graphic device**), {R} , pdf,  
ps, jpg, png .  
{R} , grid , Splus Trellis . grid , lattice,  
ggplot2 . tidyverse , ggplot2 .  
ggplot2 , .

## 5.1

Edward Tufte (2006) Beautiful Evidence .

- .
- .
- .
- .
- .
- .

## 5.2 ggplot2

ggplot2 , ggplot2 , ggplot2 R base .  
, , <https://www.r-graph-gallery.com/index.html>.

```
ggplot2      ,      ,      ,      ,      .      . ggplot2      ,      +
(layers),      .
```

- data: .
- mapping (aes):
  - x- , y- , treat, fill, shape, size, etc.
- geoms: geometric object
  - point, line, bar, shapes, ribbon, polygon, smooth, text etc.
- stat: / , statistics
- position: position adjustments.

Table: ggplot2

```
ggplot()      .
ggplot(data = data_name,
      aes(x = variable_name,
          y = variable_name,
          ... <other variable_name mappings>)) +
geom_<type>() +
...
```

Prentice (1973) , , , % Veteran's Administration  
 , , , . **survVATrial.csv.**

---

|                 |                              |
|-----------------|------------------------------|
| treat (therapy) | : 0 = ; 1 =                  |
| cellcode        | ; 1 = ; 2 = ; 3 = ; 4 =      |
| time            | , ,                          |
| censor          | : 0 = ; 1 =                  |
| diagtime        | Karnofsky performance score, |
| diagtime        | ,                            |
| age             | ( )                          |
| prior           | ; 0 = ; 1 =                  |

---

```
dd <- read.table("./Data/survVATrial.csv",
      header = TRUE,
      sep = ",",
      quote = "\"\"",
      dec = ".",
      row.names = NULL,
      # col.names,
      as.is = TRUE,
      # as.is = !stringsAsFactors,
      na.strings = c(".", "NA"))
head(dd)
##   treat cellcode time censor diagtime kps age prior
```



```
## 1      0      1    72      1      60    7  69      0
## 2      0      1   411      1      70    5  64     10
## 3      0      1   228      1      60    3  38      0
## 4      0      1   126      1      60    9  63     10
## 5      0      1   118      1      70   11  65     10
## 6      0      1    10      1      20    5  49      0
str(dd)
## 'data.frame':      137 obs. of  8 variables:
## $ treat      : int  0 0 0 0 0 0 0 0 0 0 ...
## $ cellcode: int  1 1 1 1 1 1 1 1 1 1 ...
## $ time       : int  72 411 228 126 118 10 82 110 314 100 ...
## $ censor     : int  1 1 1 1 1 1 1 1 1 0 ...
## $ diagtime: int  60 70 60 60 70 20 40 80 50 70 ...
## $ kps        : int  7 5 3 9 11 5 10 29 18 6 ...
## $ age        : int  69 64 38 63 65 49 69 68 43 70 ...
## $ prior      : int  0 10 0 10 10 0 10 0 0 0 ...
dd$treat <- factor(dd$treat, labels = c("placebo", "test"))
dd$cellcode <- factor(dd$cellcode,
                      labels = c("squamous", "small", "adeno", "large"))
dd$censor <- factor(dd$censor, labels = c("survival", "dead"))
dd$prior <- factor(dd$prior, labels = c("no", "yes"))
head(dd)
##      treat cellcode time censor diagtime kps age prior
## 1 placebo squamous  72   dead      60    7  69   no
## 2 placebo squamous 411   dead      70    5  64   yes
## 3 placebo squamous 228   dead      60    3  38   no
## 4 placebo squamous 126   dead      60    9  63   yes
## 5 placebo squamous 118   dead      70   11  65   yes
## 6 placebo squamous  10   dead      20    5  49   no
str(dd)
## 'data.frame':      137 obs. of  8 variables:
## $ treat      : Factor w/ 2 levels "placebo","test": 1 1 1 1 1 1 1 1 1 1 ...
## $ cellcode: Factor w/ 4 levels "squamous","small",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ time       : int  72 411 228 126 118 10 82 110 314 100 ...
## $ censor     : Factor w/ 2 levels "survival","dead": 2 2 2 2 2 2 2 2 2 1 ...
## $ diagtime: int  60 70 60 60 70 20 40 80 50 70 ...
## $ kps        : int  7 5 3 9 11 5 10 29 18 6 ...
## $ age        : int  69 64 38 63 65 49 69 68 43 70 ...
## $ prior      : Factor w/ 2 levels "no","yes": 1 2 1 2 2 1 2 1 1 1 ...
```

## 5.3

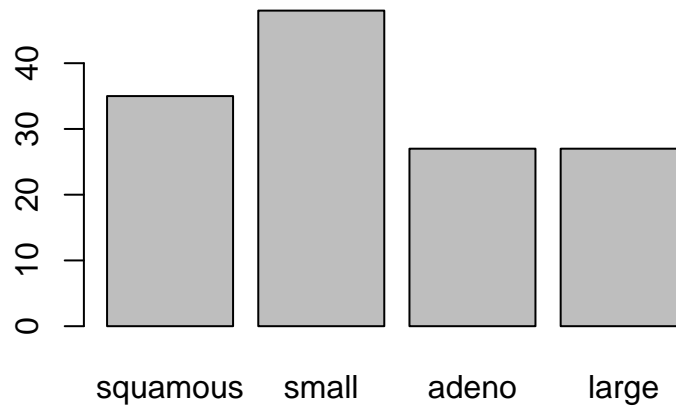
(bar plot) (distribution), . (frequency table),  
(pie chart).

, ., (Table) .

### 5.3.1

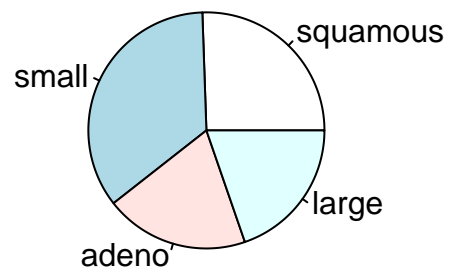
• : .

```
## pie chart: ggplot2 do not have a simple geom_pie()
## use R base pie()
cellcode.tab <- table(dd$cellcode)
cellcode.tab
##
## squamous      small      adeno      large
##          35          48          27          27
prop.table(cellcode.tab)
##
## squamous      small      adeno      large
##    0.2555    0.3504    0.1971    0.1971
barplot(cellcode.tab)
round(barplot(cellcode.tab), 4)
```

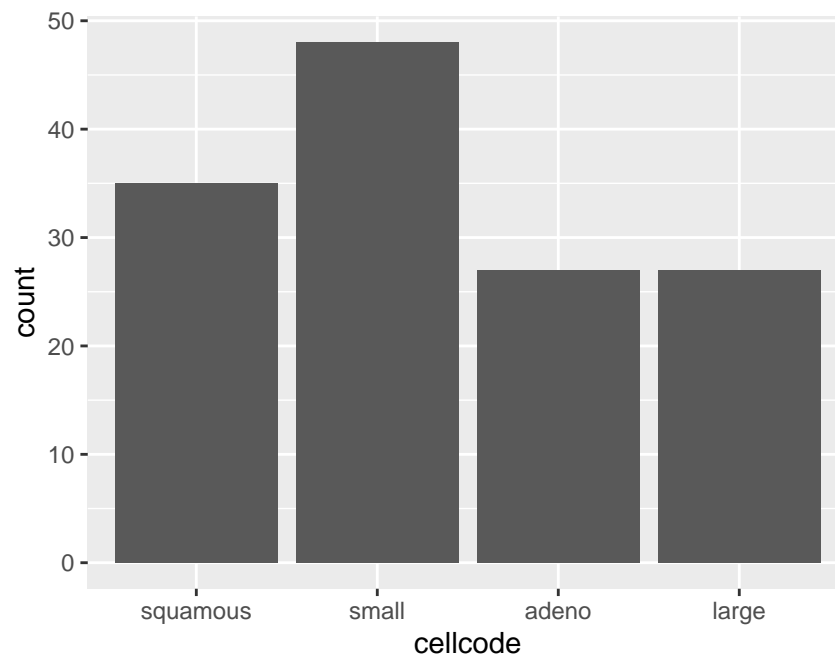


```
##      [,1]
## [1,]  0.7
## [2,]  1.9
## [3,]  3.1
## [4,]  4.3
```

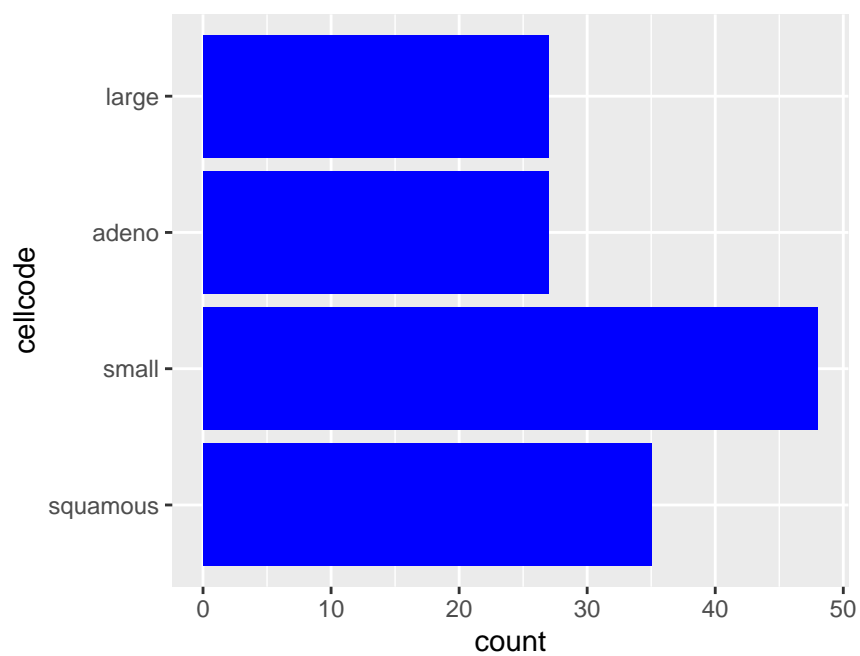
```
pie(cellcode.tab)
```



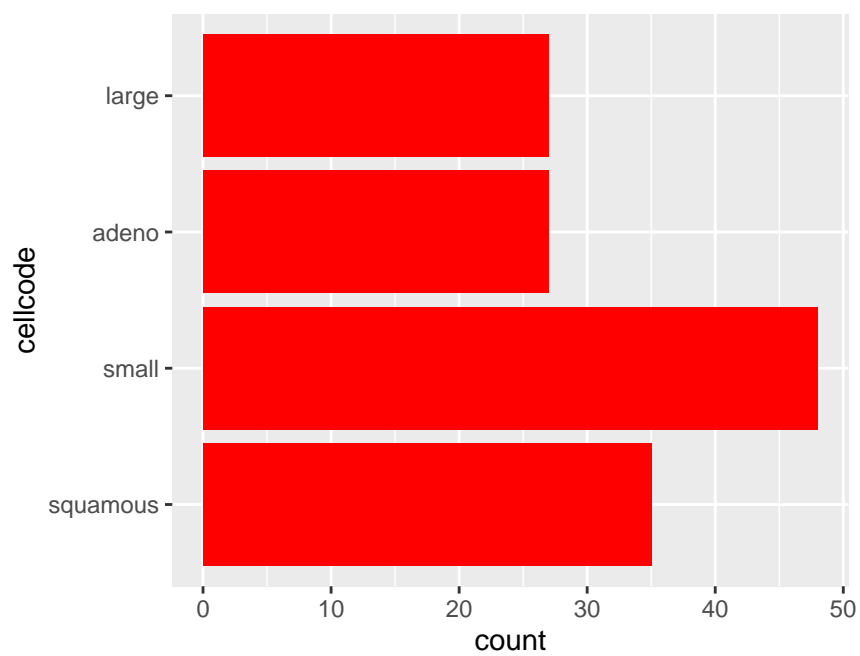
```
library(ggplot2)
## bar chart
ggplot(data = dd, aes(x = cellcode)) +
  geom_bar()
```



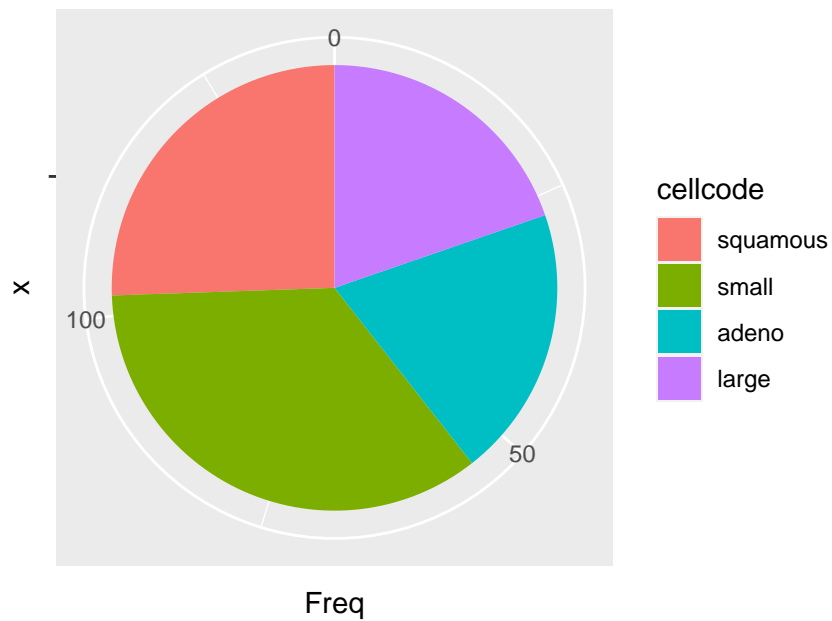
```
ggplot(data = dd, aes(x = cellcode)) +  
  geom_bar(fill = "blue") +  
  coord_flip()
```



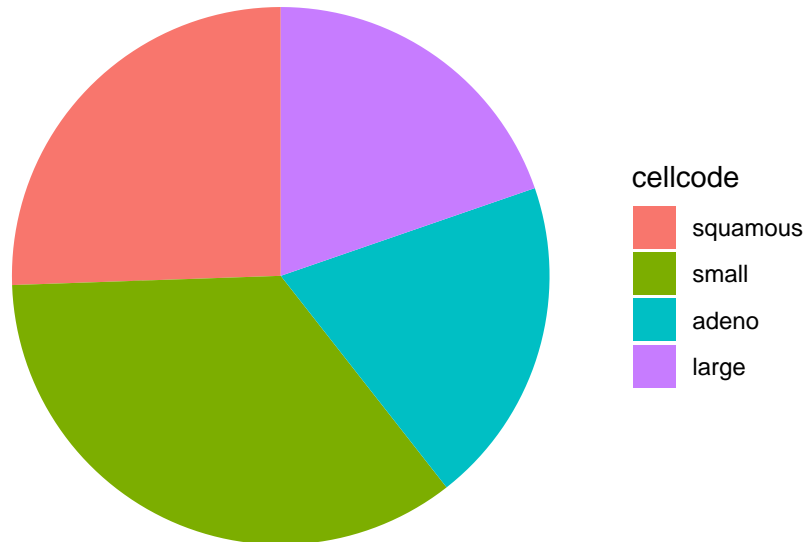
```
ggplot(data = dd, aes(y = cellcode)) +  
  geom_bar(fill = "red")
```



```
# pie chart: no simple solution
clar.freq <- data.frame(cellcode.tab)
names(clar.freq)[1] <- "cellcode"
clar.freq
##   cellcode Freq
## 1 squamous  35
## 2   small  48
## 3   adeno  27
## 4   large  27
ggplot(data = clar.freq, aes(x = "", y = Freq, fill = cellcode)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar("y", start = 0)
```



```
ggplot(data = clar.freq, aes(x = "", y = Freq, fill = cellcode)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y", start = 0) +
  theme_void() # remove background
```



### 5.3.2

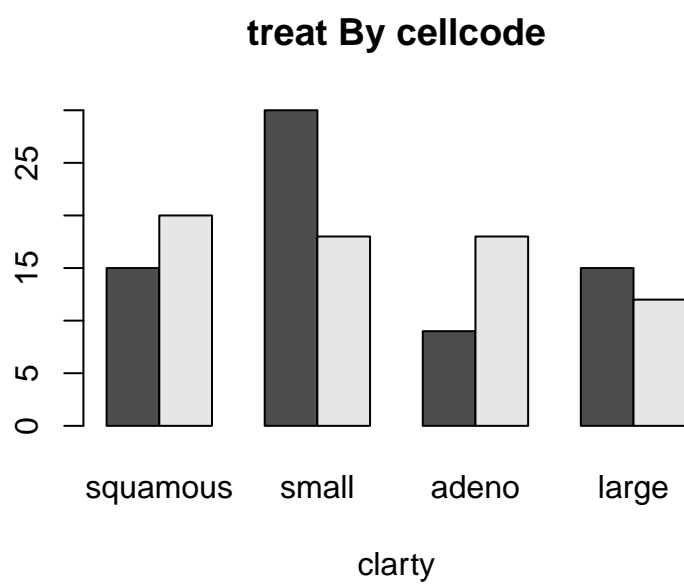
```

• : , .
## two categorical vtriables
table(dd$treat)
##
## placebo    test
##      69     68
table(dd$cellcode)
##
## squamous    small    adeno    large
##      35      48      27      27
twoway.tab <- table(dd$treat, dd$cellcode)
twoway.tab
##
##           squamous small adeno large
## placebo      15    30    9    15
## test        20    18   18   12
## # cell proportion
cell.prop <- prop.table(twoway.tab, margin=NULL)
round(cell.prop, 3)
##
##           squamous small adeno large

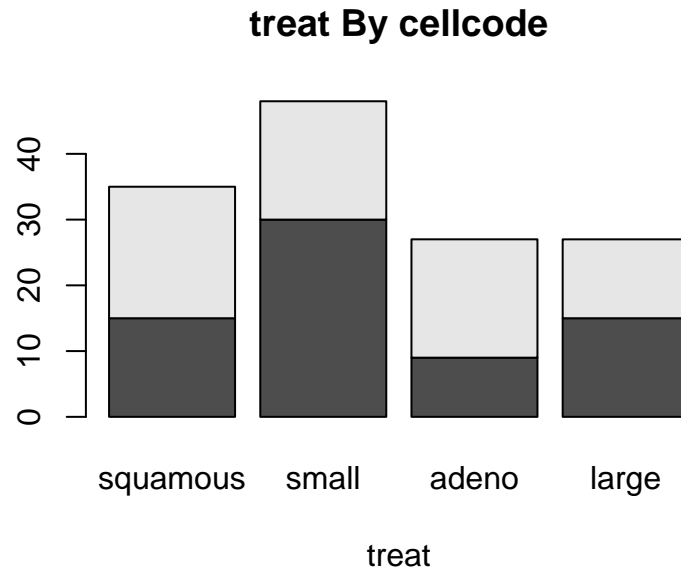
```

```
## placebo 0.109 0.219 0.066 0.109
## test 0.146 0.131 0.131 0.088
## conditional on row sum to 1
cond_row_prop <- prop.table(twoway.tab, margin = 1)
round(cond_row_prop, 3)
##
## squamous small adeno large
## placebo 0.217 0.435 0.130 0.217
## test 0.294 0.265 0.265 0.176
apply(cond_row_prop, 1, sum) # rows sum to 1
## placebo test
## 1 1
## conditional on column sum to 1
cond_col_prop <- prop.table(twoway.tab, margin = 2)
round(cond_col_prop, 3)
##
## squamous small adeno large
## placebo 0.429 0.625 0.333 0.556
## test 0.571 0.375 0.667 0.444
apply(cond_col_prop, 2, sum) # cols sum to 1
## squamous small adeno large
## 1 1 1 1
## side-by-side bar plot
barplot(twoway.tab,
        beside = TRUE,
        main = "treat By cellcode",
        xlab = "clarty")
```

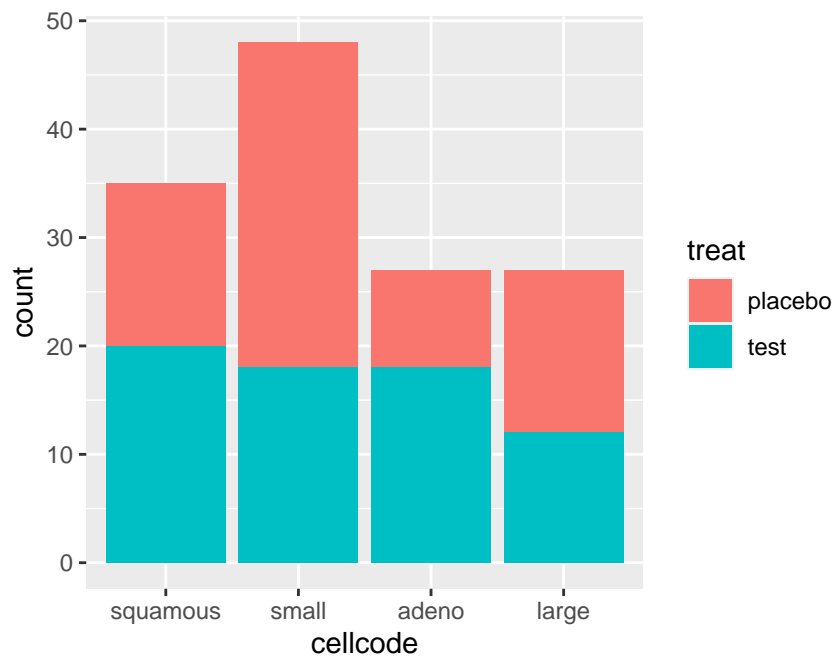




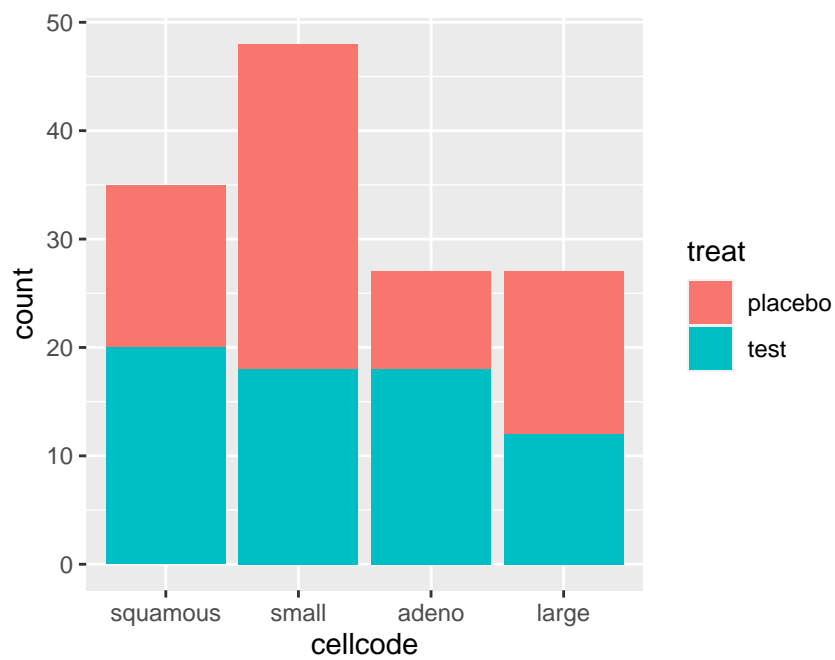
```
# Stacked Bar Plot  
barplot(twoway.tab,  
        beside = FALSE,  
        main = "treat By cellcode",  
        xlab = "treat")
```



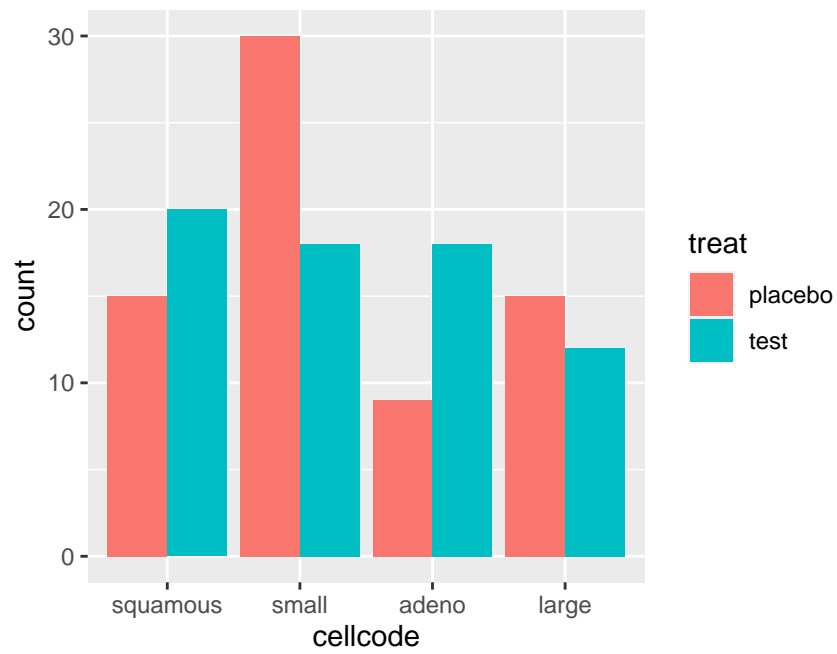
```
## ggplot2
## Automatically stack
library(ggplot2)
ggplot(data = dd, aes(x = cellcode, fill = treat)) +
  geom_bar()
```



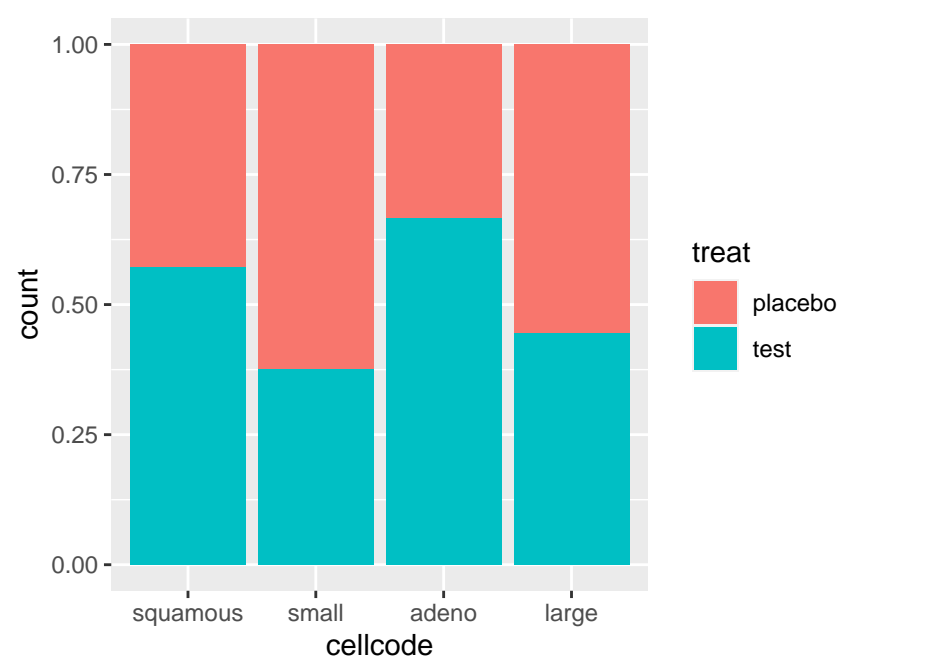
```
ggplot(data = dd, aes(x = cellcode, fill = treat)) +  
  geom_bar(position = "stack")
```



```
## side-by-side  
ggplot(data = dd, aes(x = cellcode, fill = treat)) +  
  geom_bar(position = "dodge")
```



```
ggplot(data = dd, aes(x = cellcode, fill = treat)) +  
  geom_bar(position = "fill")
```

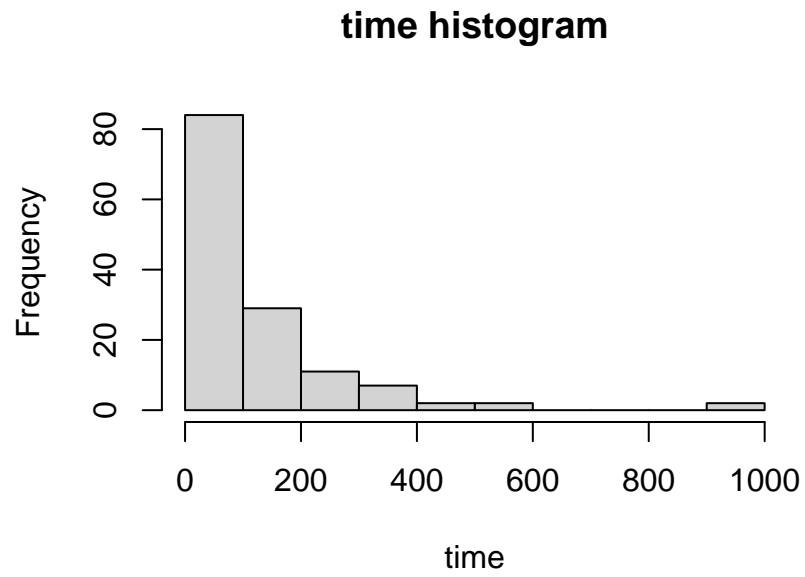


5.4

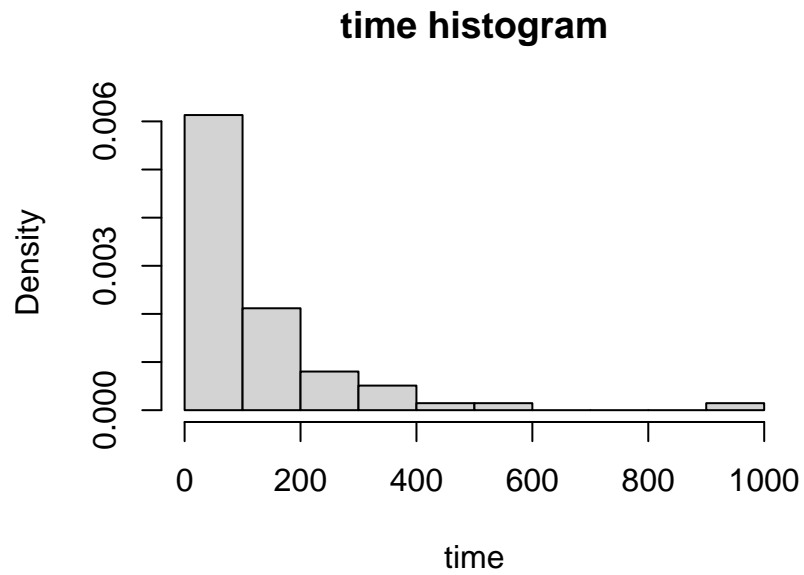
(distribution), (dot plot), (stem-and-leaf), (histogram), (box plot), (density plot), .

5.4.1

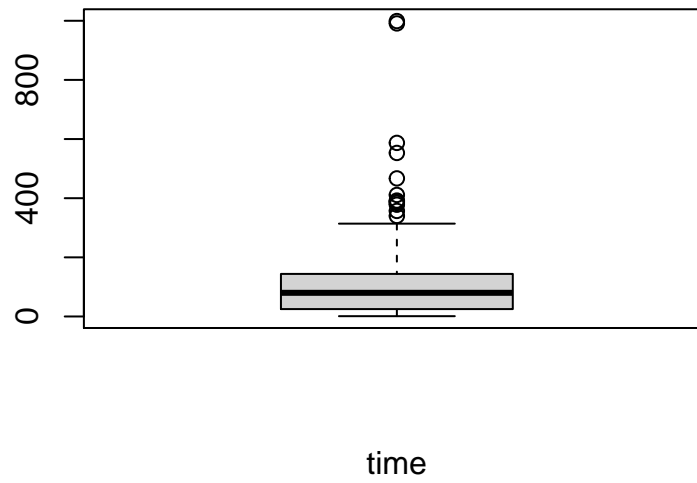
```
• : , , .  
## use R base pie()  
## histogram  
hist(dd$time,  
      freq = TRUE,  
      main = "time histogram",  
      xlab = "time")
```



```
hist(dd$time,  
     freq = FALSE,  
     main = "time histogram",  
     xlab = "time")
```

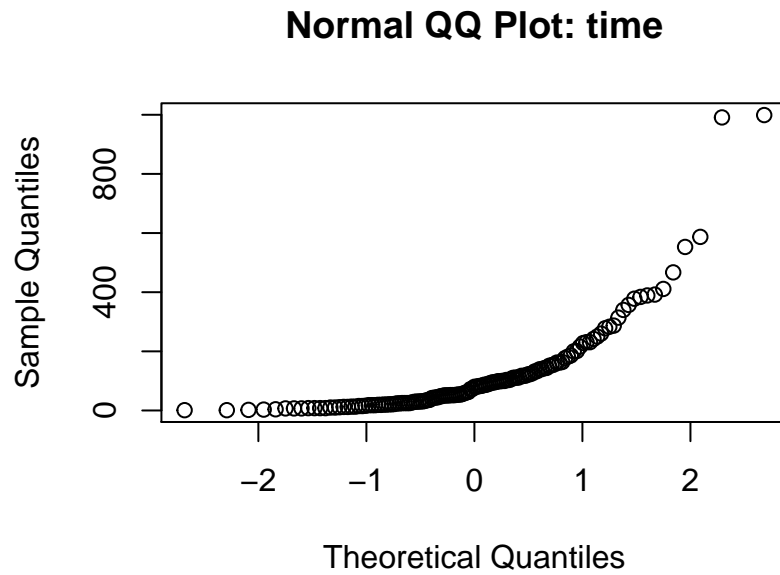


```
# box plot  
boxplot(dd$time,  
        xlab = "time")
```



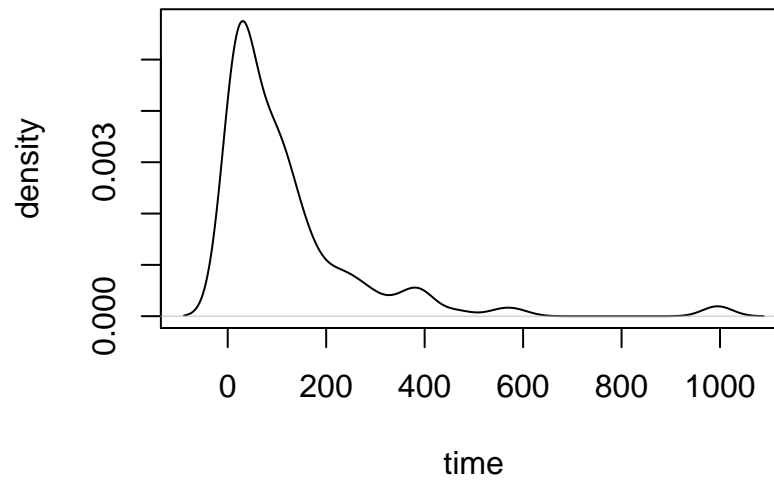
```
# QQ plot  
qqnorm(dd$time,  
        main = "Normal QQ Plot: time")
```



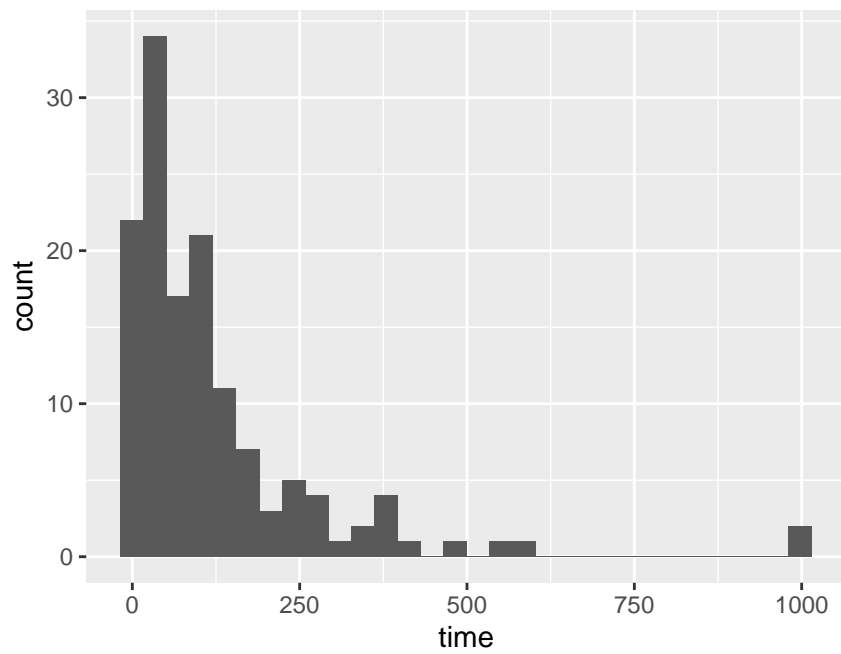


```
# density plot
plot(density(dd$time),
     pch = 16,
     main = "Density Plot",
     xlab = "time",
     ylab = "density")
```

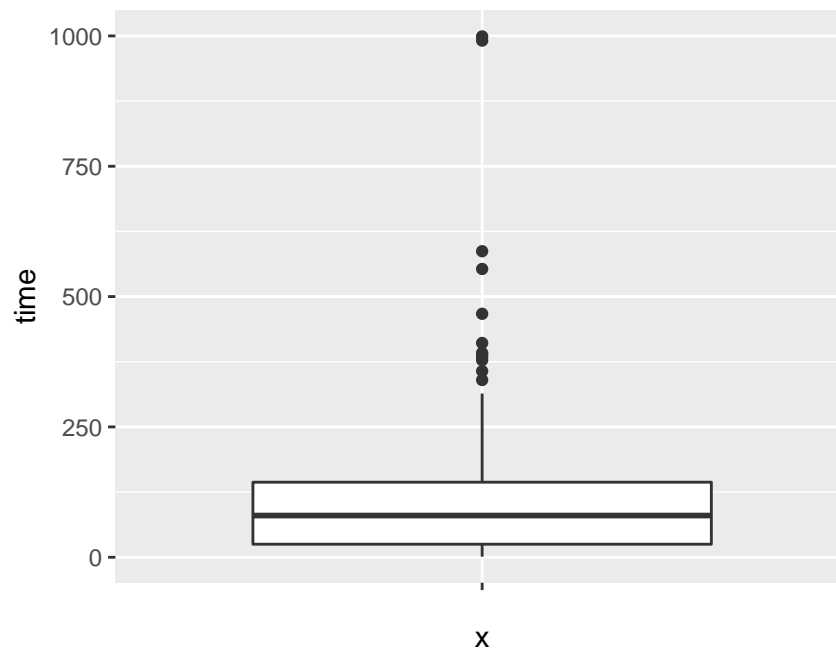
### Density Plot



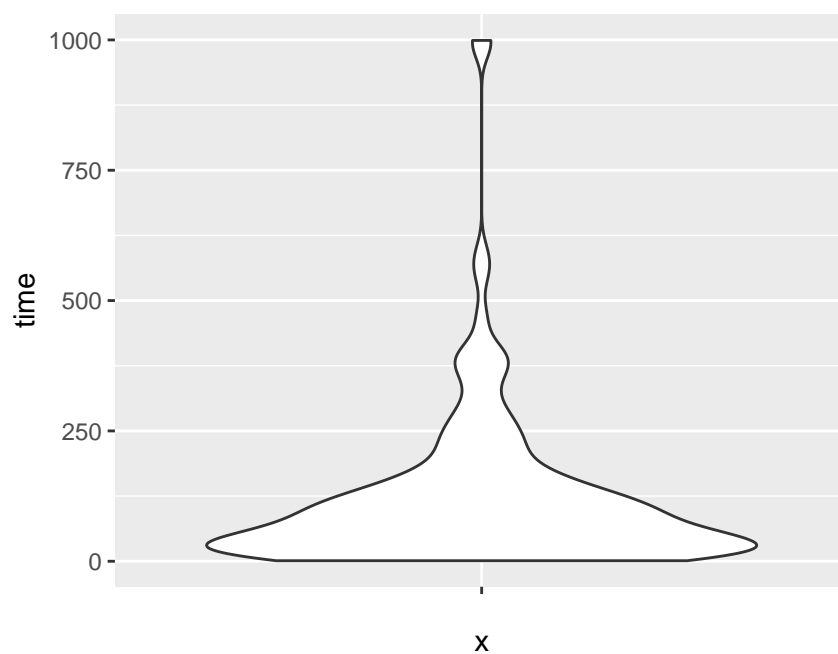
```
## ggplot2
## histogram
ggplot(data = dd, aes(x = time)) +
  geom_histogram()
```



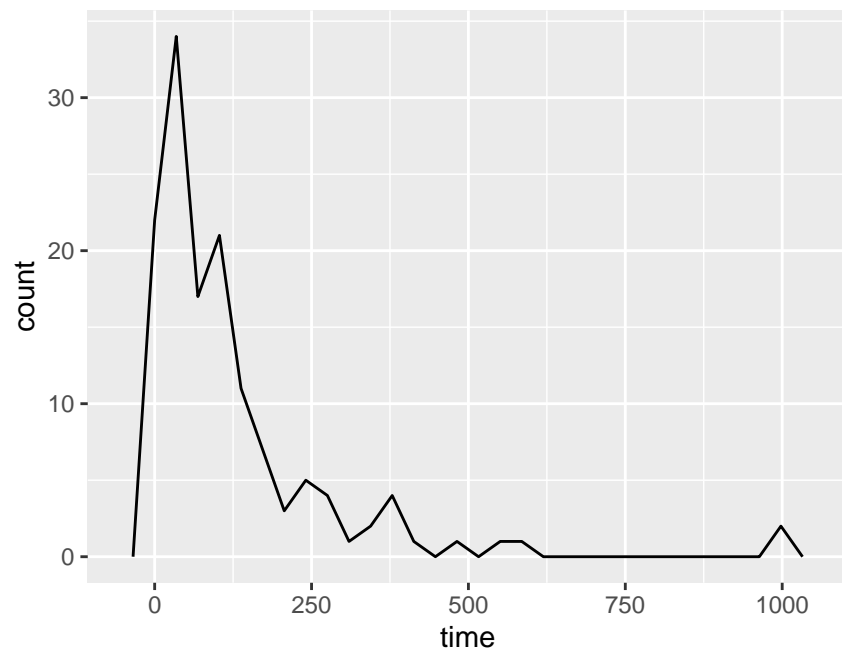
```
## box plot
ggplot(dd, aes(x = "", y = time)) +
  geom_boxplot()
```



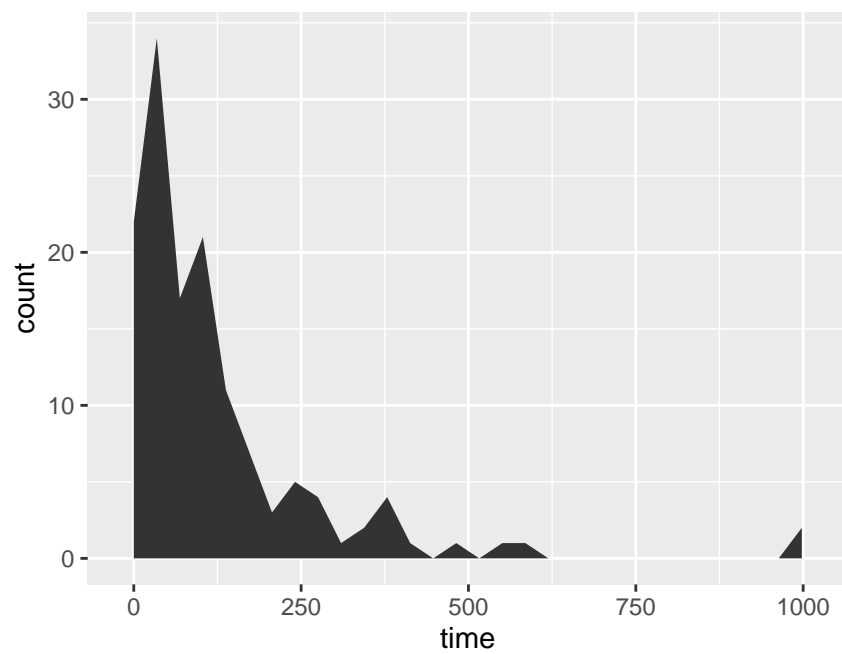
```
## violin plot  
ggplot(dd, aes(x = "", y = time)) +  
  geom_violin()
```



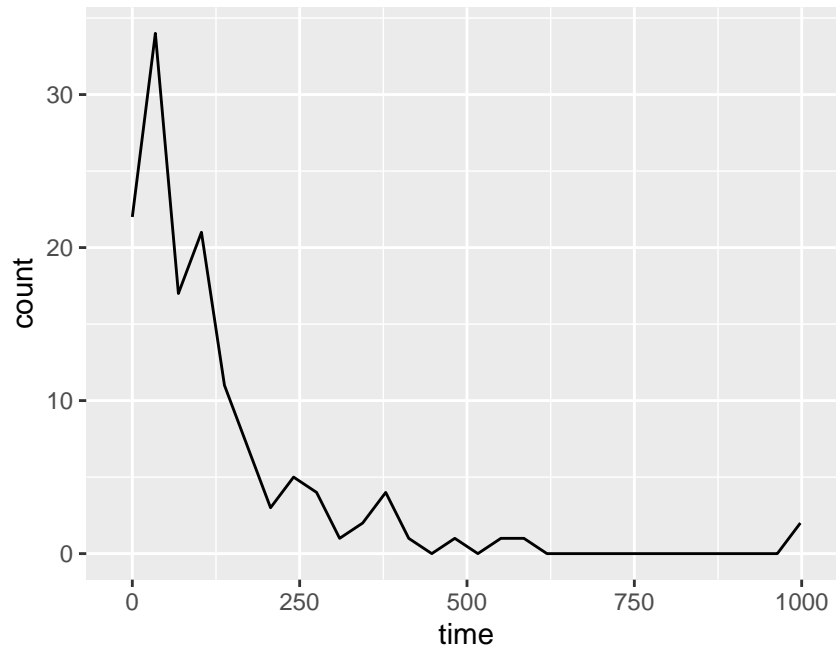
```
## density plot  
ggplot(data = dd, aes(x = time)) +  
  geom_freqpoly()
```



```
ggplot(data = dd, aes(x = time)) +  
  stat_bin(geom = "area")
```



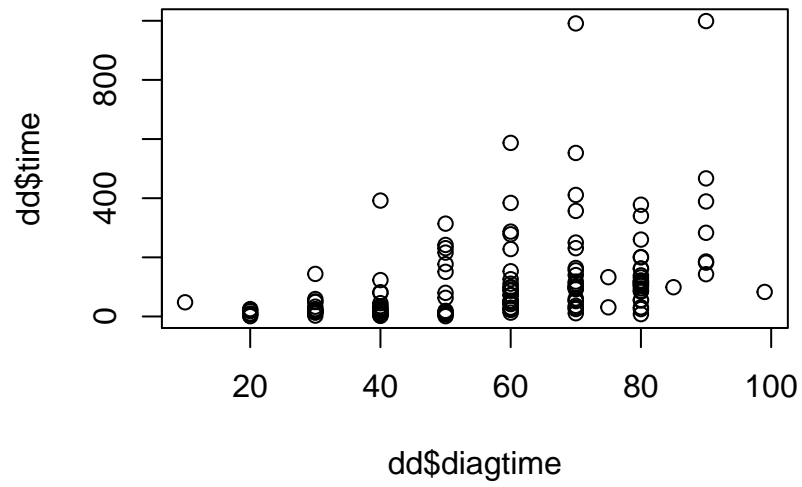
```
ggplot(data = dd, aes(x = time)) +  
  stat_bin(geom = "line")
```



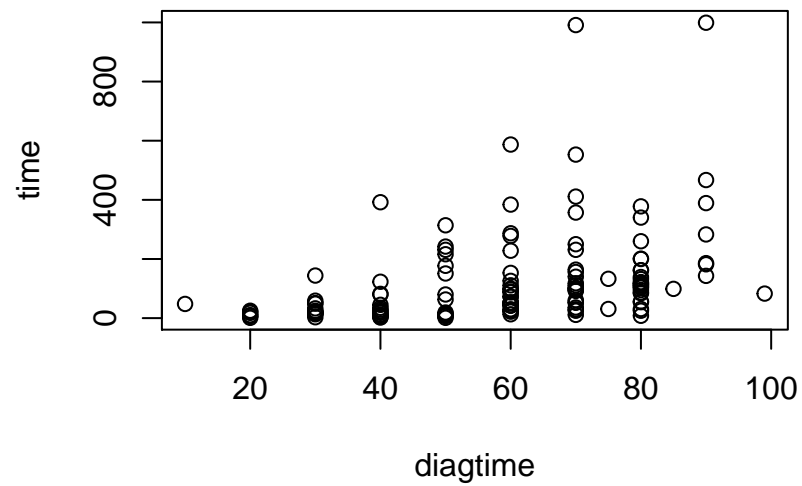
### 5.4.2

- scatter plot = X & Y =
- :

```
## R base  
## scatter plot  
## basic  
plot(x = dd$diagtime, y = dd$time)
```

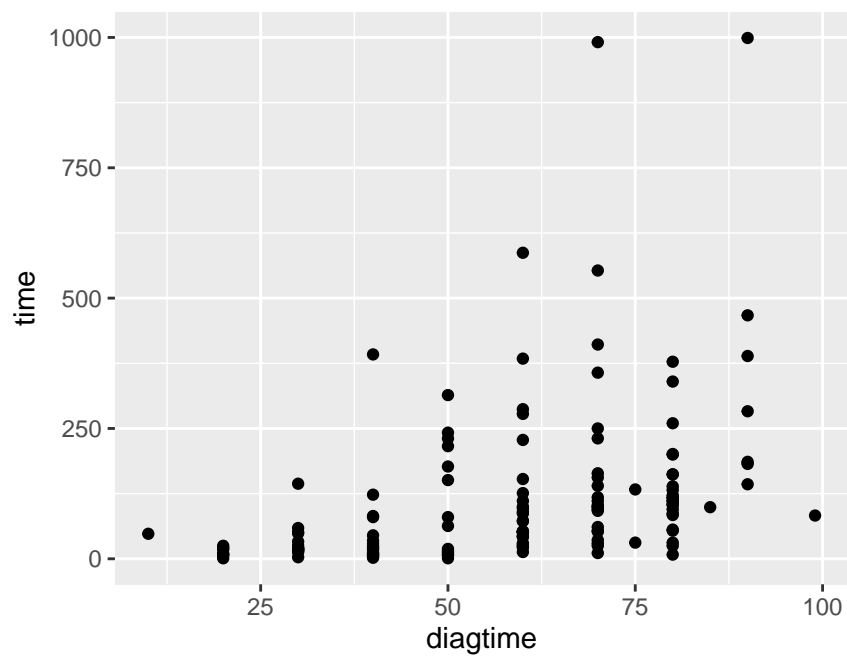


```
## formulat y ~ x, data = data_name)
plot(time ~ diagtime, data = dd)
```

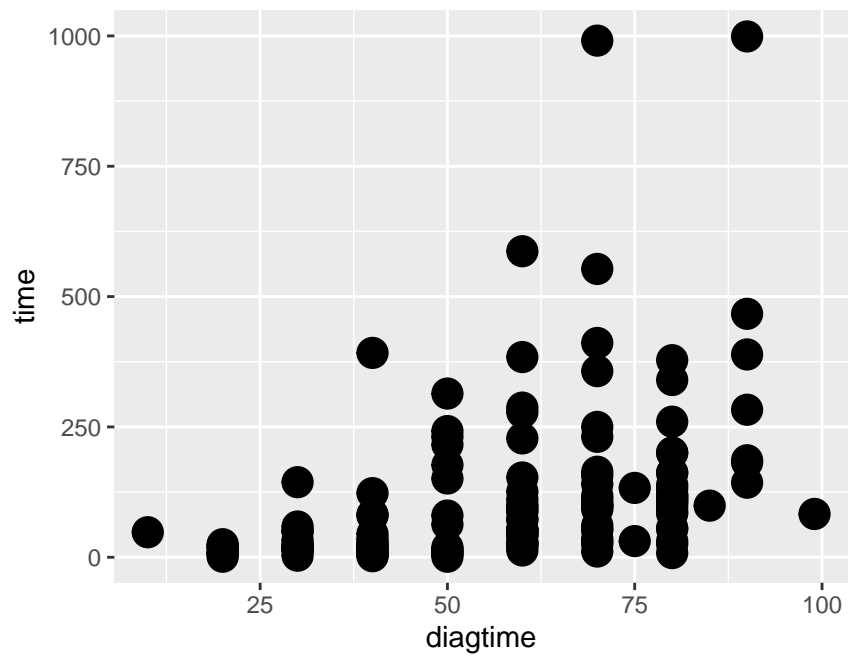




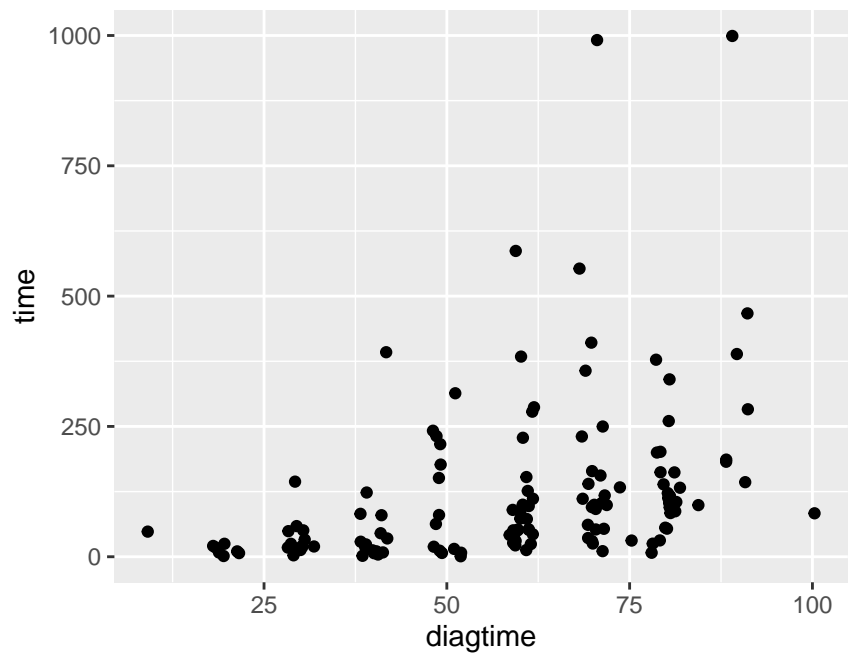
```
## ggplot
ggplot(data = dd, aes(x = diagtime, y = time)) +
  geom_point()
```



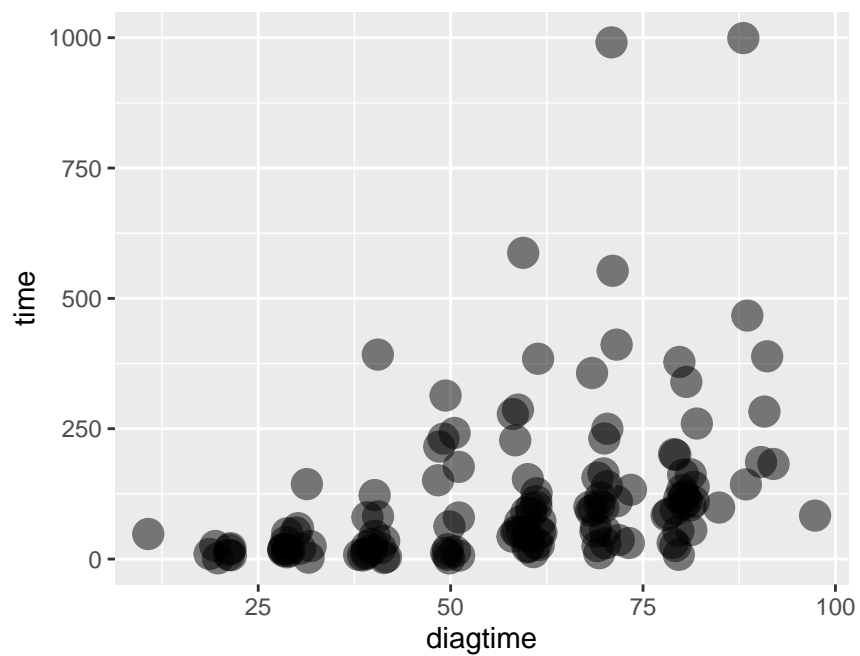
```
ggplot(data = dd, aes(x = diagtime, y = time)) +
  geom_point(size = 5)
```



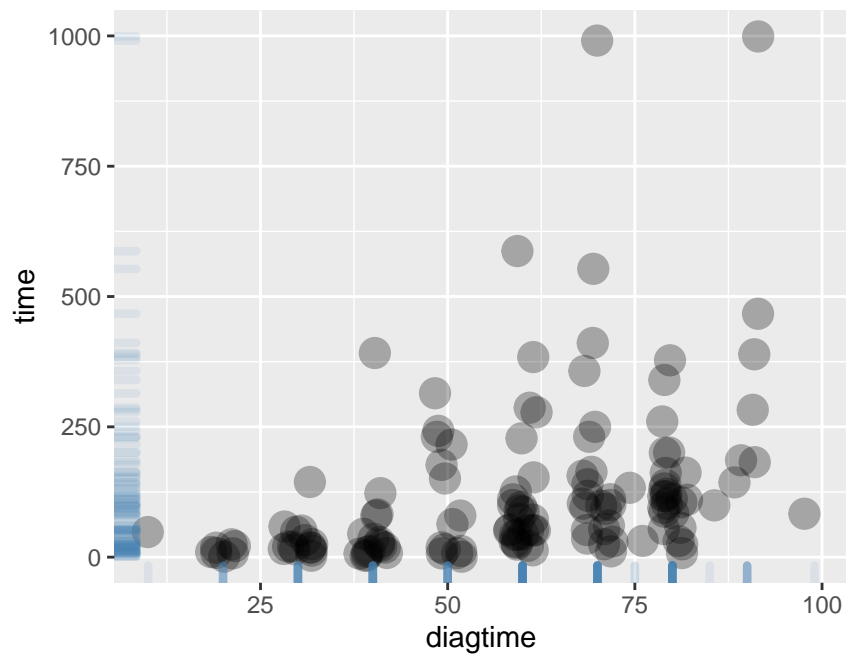
```
ggplot(data = dd, aes(x = diagtime, y = time)) +  
  geom_jitter()
```



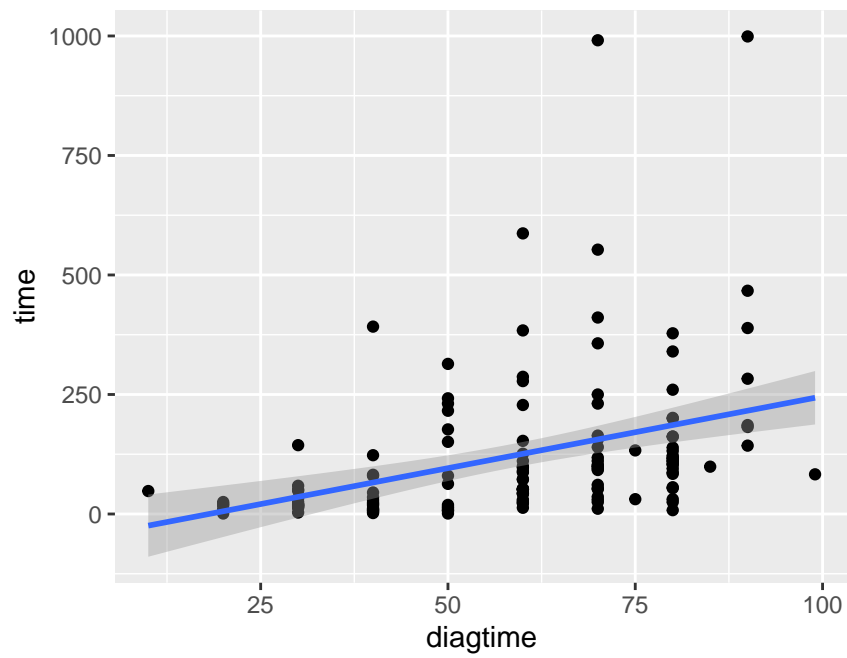
```
ggplot(data = dd, aes(x = diagtime, y = time)) +  
  geom_jitter(size = 5, alpha = 1/2)
```



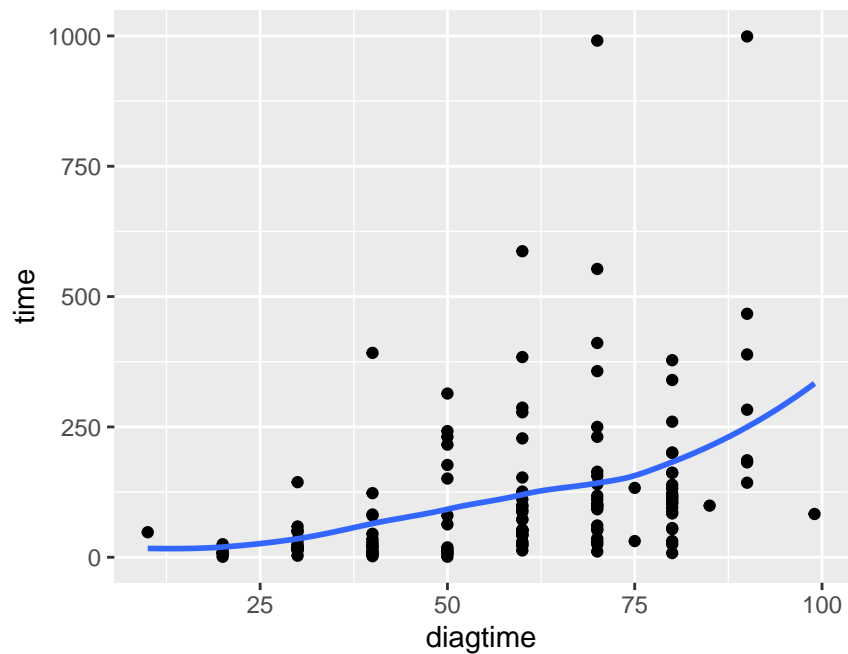
```
ggplot(data = dd, aes(x = diagtime, y = time)) +  
  geom_jitter(size = 5, alpha = 0.3) +  
  geom_rug(col = "steelblue", alpha = 0.1, size = 1.5)
```



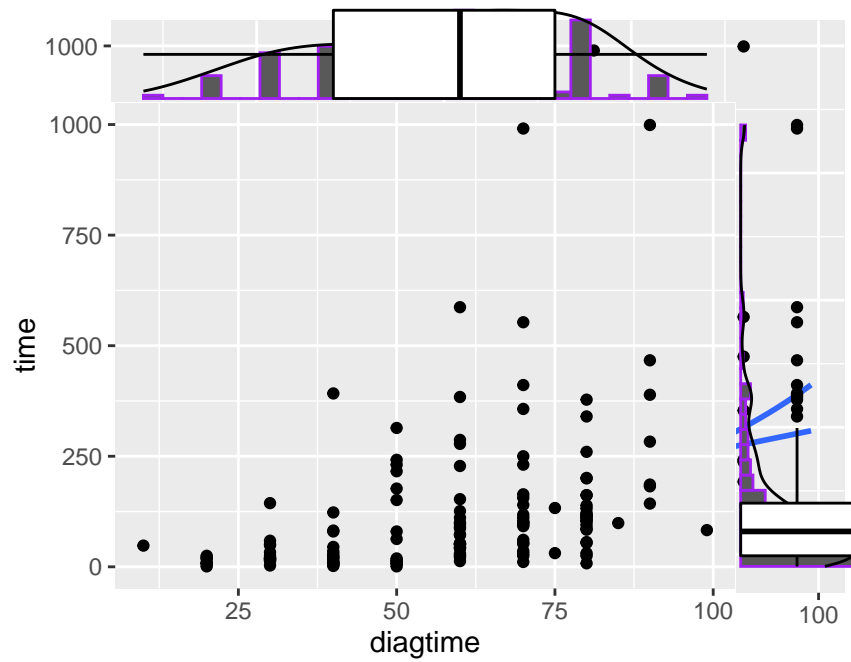
```
# add linear line or smoothing line  
ggplot(data = dd, aes(x = diagtime, y = time)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```



```
ggplot(data = dd, aes(x = diagtime, y = time)) +  
  geom_point() +  
  geom_smooth(se = FALSE)
```



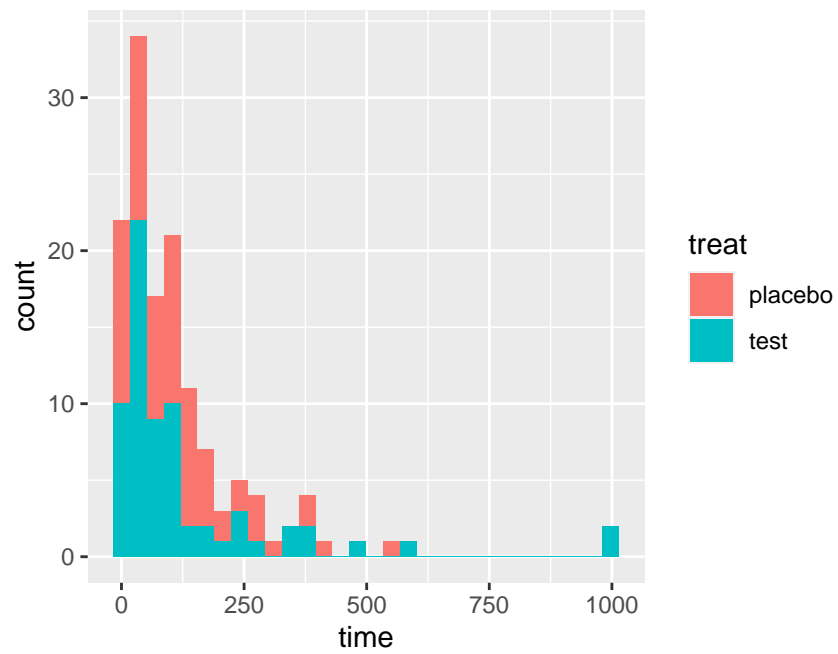
```
ggplot(data = dd, aes(x = diagtime, y = time)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_smooth(se = FALSE)
## scatter plot + marginal distribution
library(ggExtra)
# classical
p <- ggplot(dd, aes(x = diagtime, y = time)) +
  geom_point() +
  theme(legend.position = "none")
# scatter plot + marginal histogram
ggMarginal(p, type = "histogram", color = "purple")
# scatter plot + marginal density
ggMarginal(p, type = "density")
# scatter plot + marginal boxplot
ggMarginal(p, type = "boxplot")
```



## 5.5

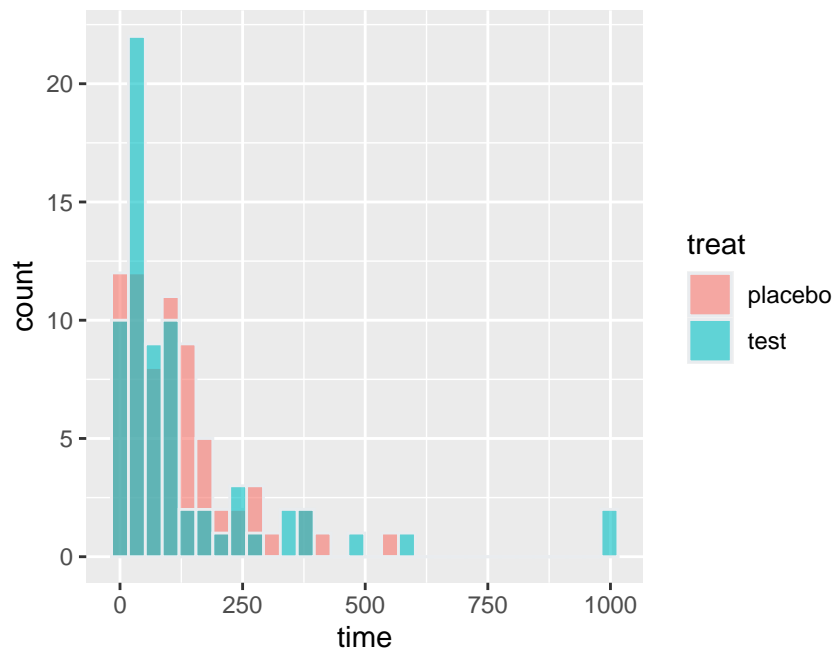
• +  
 • +  
 • + = +  
 •

```
# one continuous + one categorical
ggplot(data = dd, aes(x = time)) +
  geom_histogram(aes(fill = treat))
```

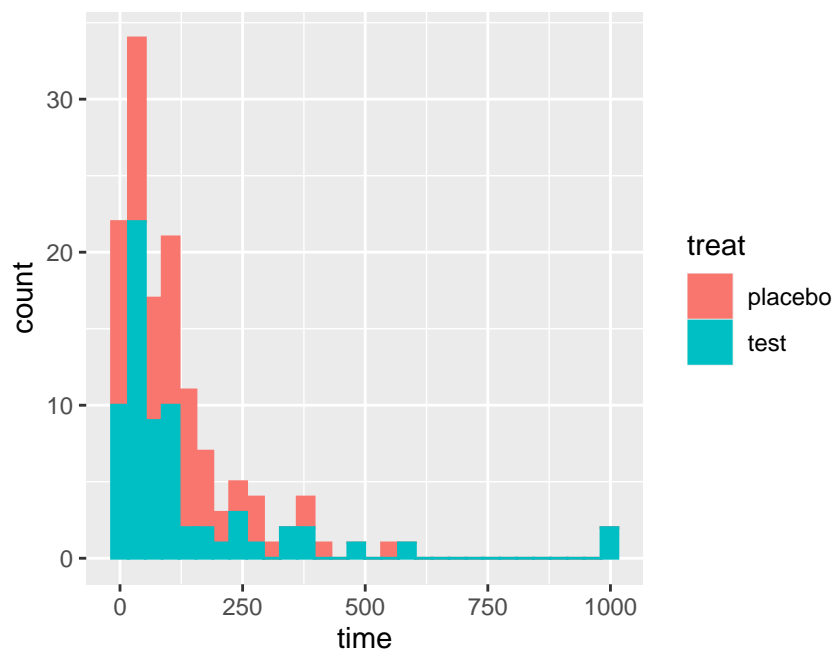


```
ggplot(data = dd, aes(x = time, fill = treat)) +  
  geom_histogram( color = "#e9ecef",  
                  alpha = 0.6,  
                  position = 'identity')
```

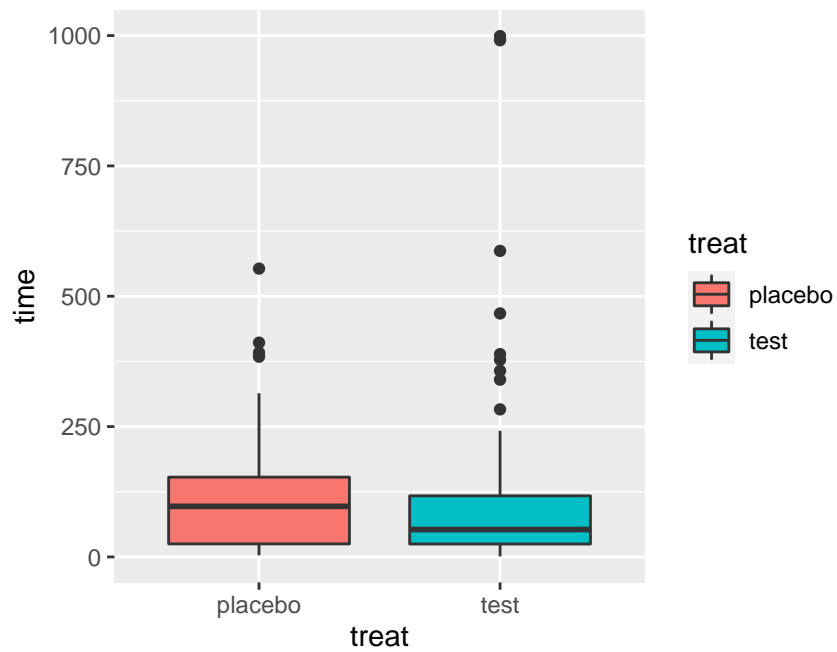




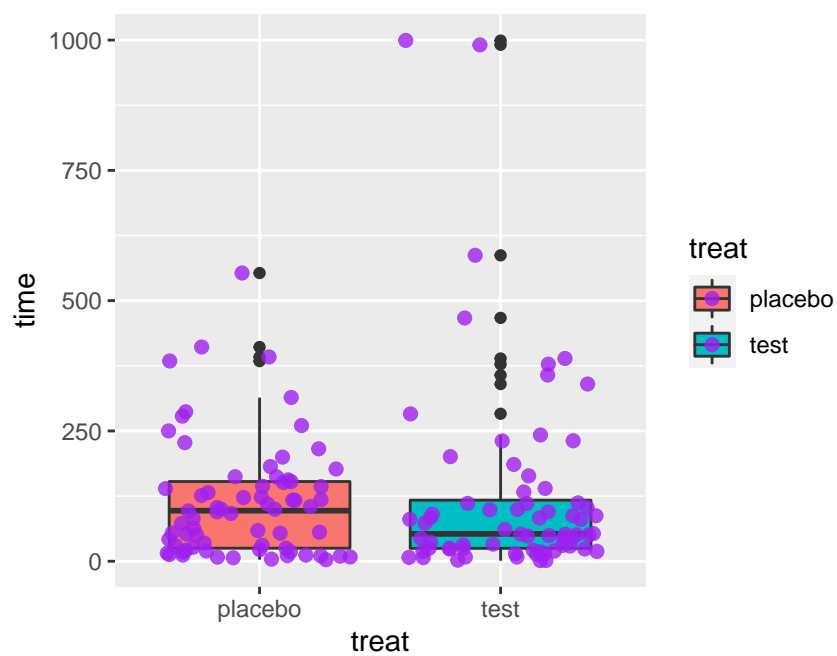
```
ggplot(data = dd, aes(x = time, color = treat, fill = treat)) +  
  geom_histogram()
```



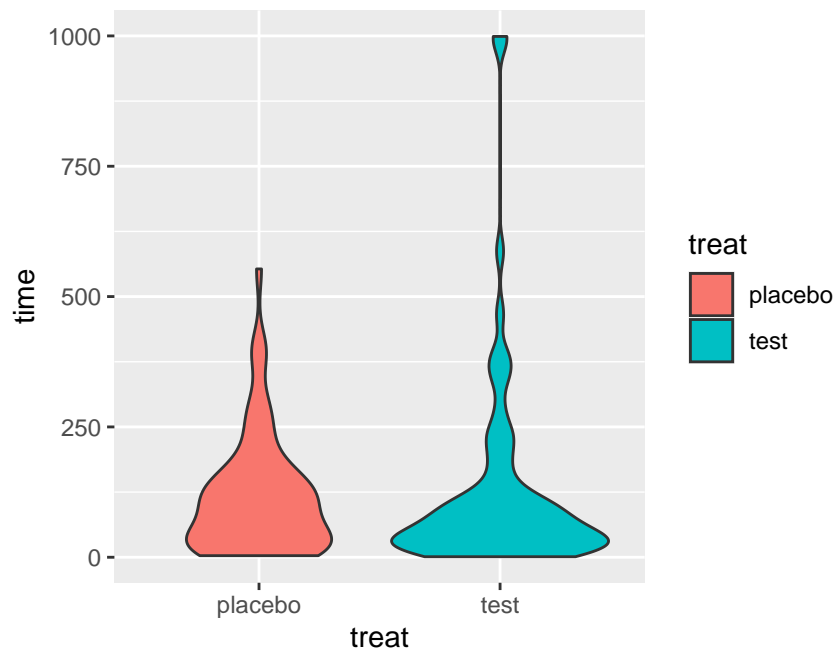
```
#  
ggplot(data = dd, aes(x = treat, y = time, fill = treat)) +  
  geom_boxplot()
```



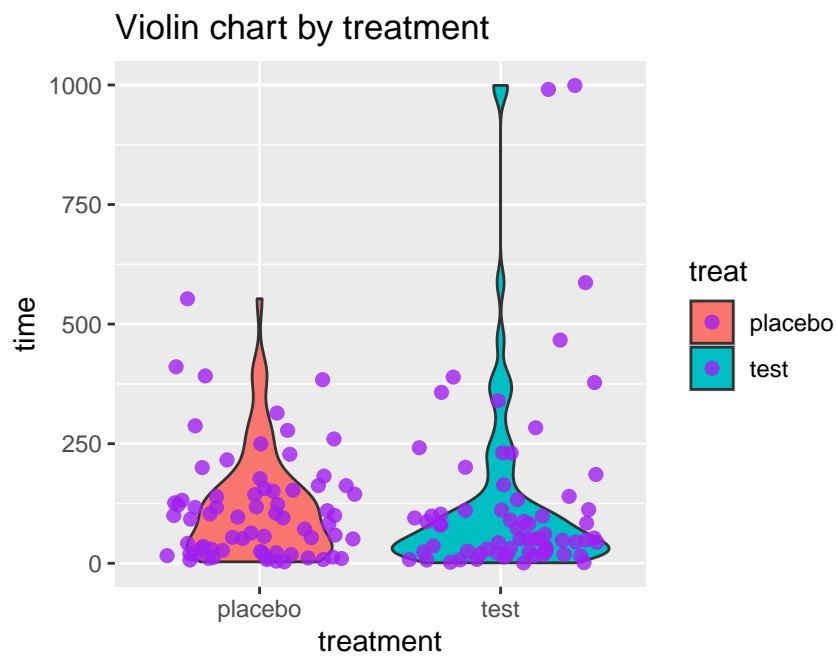
```
ggplot(data = dd, aes(x = treat, y = time, fill = treat)) +  
  geom_boxplot() +  
  geom_jitter(color = "purple", size = 2, alpha = 0.8)
```



```
#  
ggplot(data = dd, aes(x = treat, y = time, fill = treat)) +  
  geom_violin()
```

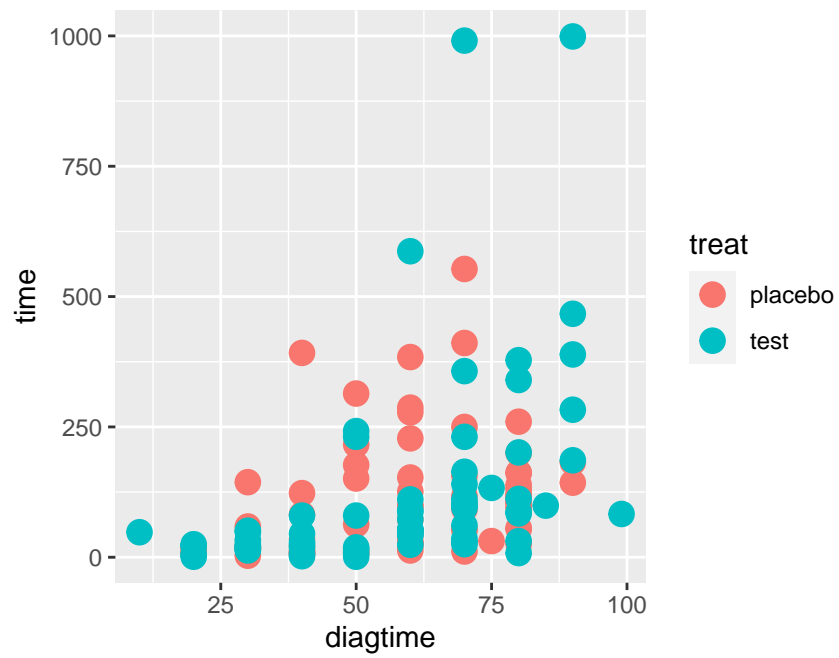


```
ggplot(data = dd, aes(x = treat, y = time, fill = treat)) +  
  geom_violin() +  
  geom_jitter(color = "purple", size = 2, alpha = 0.8) +  
  ggtitle("Violin chart by treatment") +  
  xlab("treatment")
```

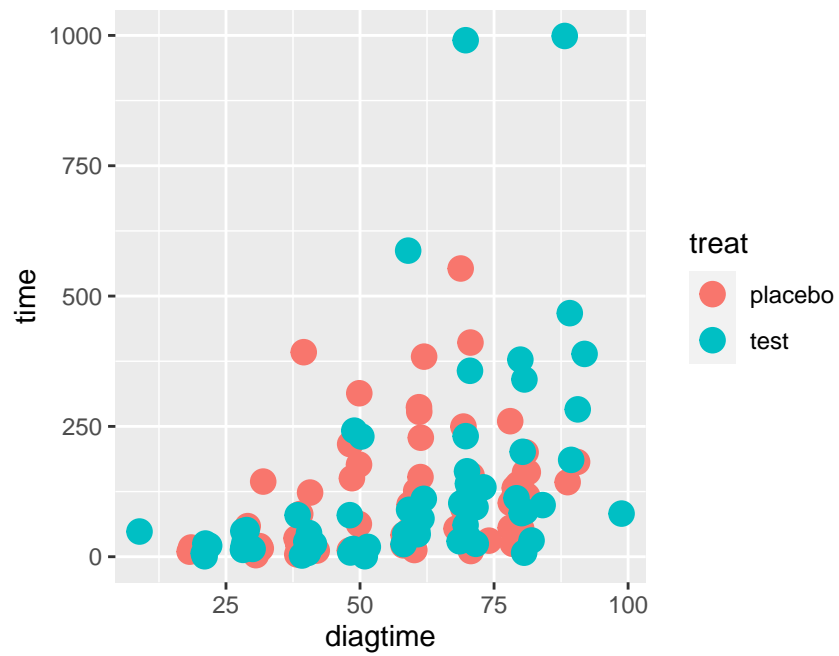


```
## ggplot2
## two continuous + one categorical
ggplot(data = dd, aes(x = diagtime, y = time, color = treat)) +
  geom_point(size = 4)
```

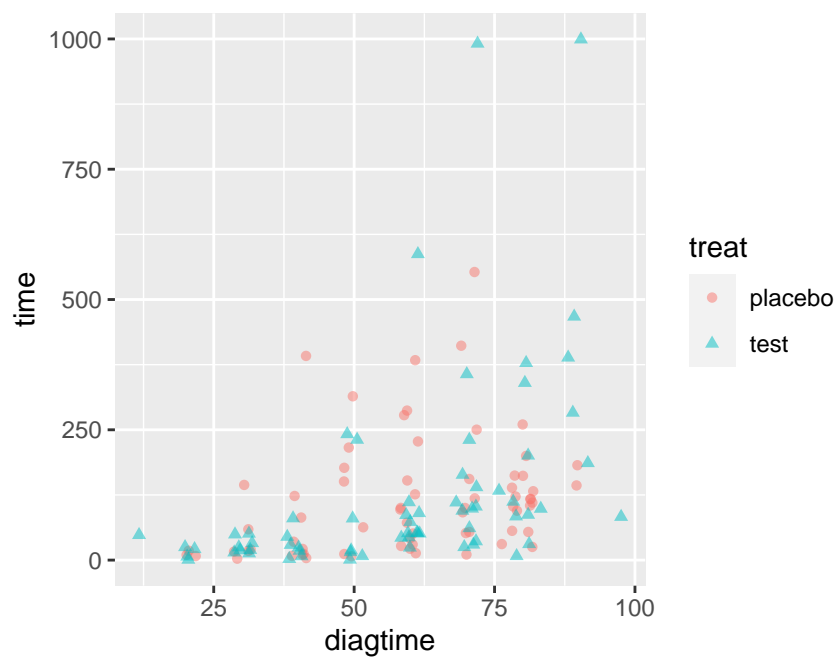
5.5.2 +



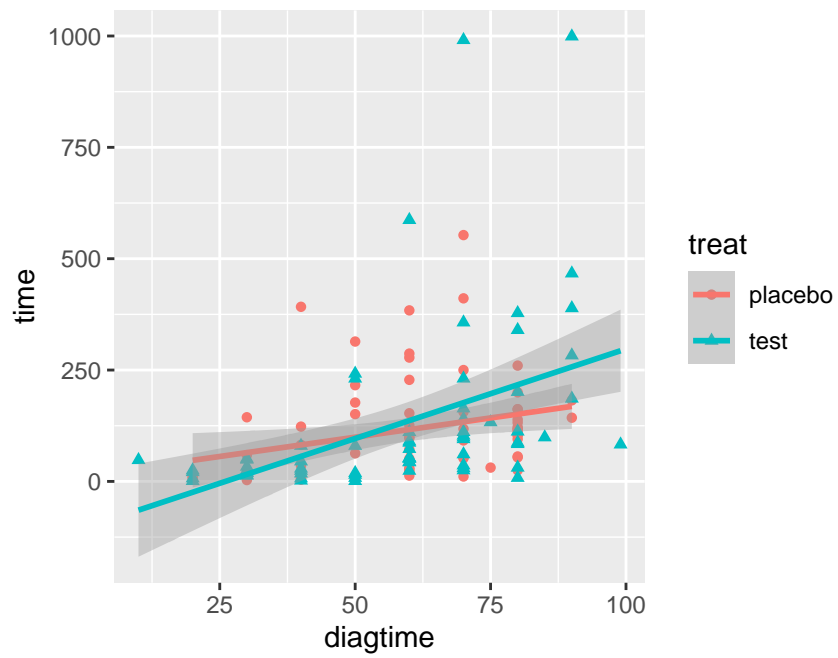
```
ggplot(data = dd, aes(x = diagtime, y = time, color = treat)) +  
  geom_jitter(size = 4)
```



```
ggplot(data = dd, aes(x = diagtime, y = time,
                      color = treat, shape = treat)) +
  geom_jitter(alpha = 1/2)
```

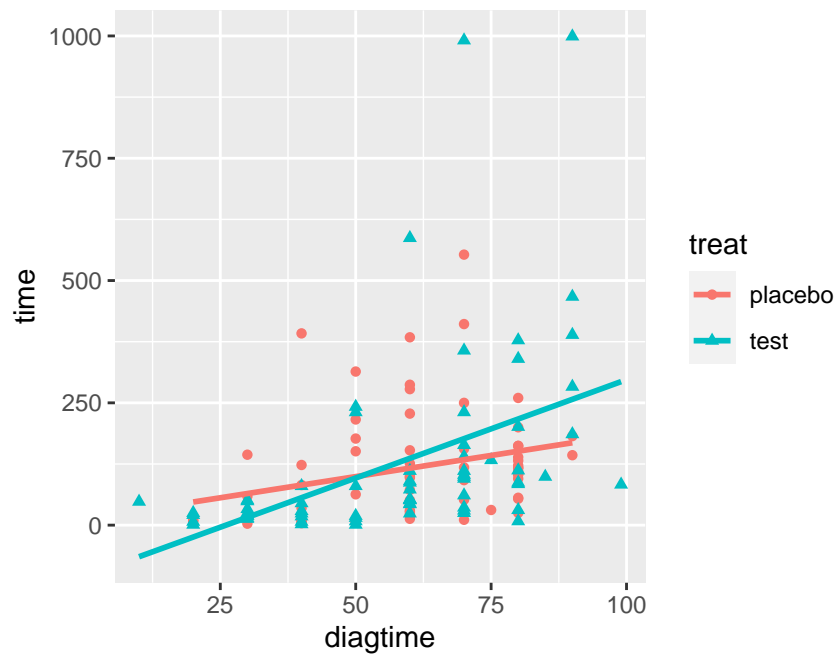


```
# add linear line or smoothing line
ggplot(data = dd, aes(x = diagtime, y = time,
                      color = treat, shape = treat)) +
  geom_point() +
  geom_smooth(method = "lm")
```

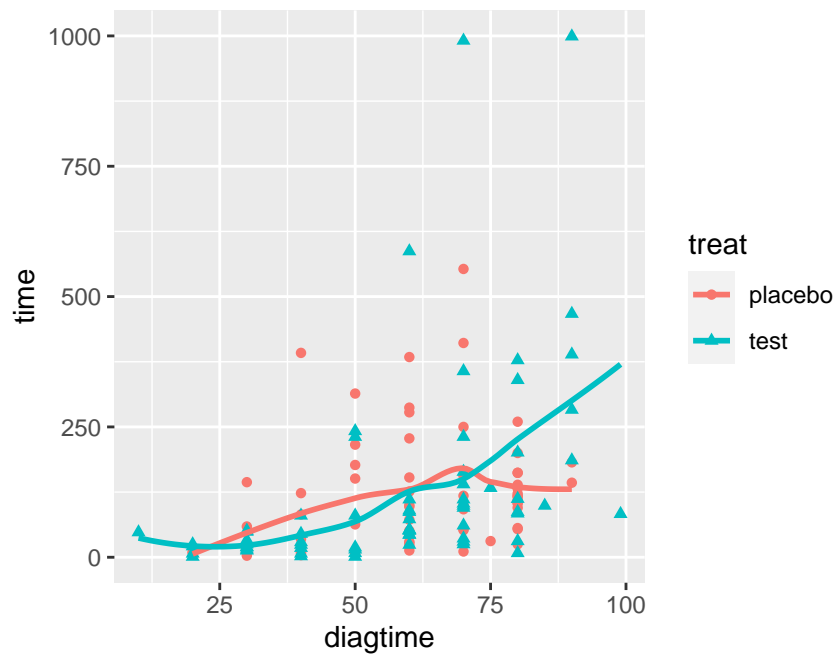


```
ggplot(data = dd, aes(x = diagtime, y = time,
                      color = treat, shape = treat)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

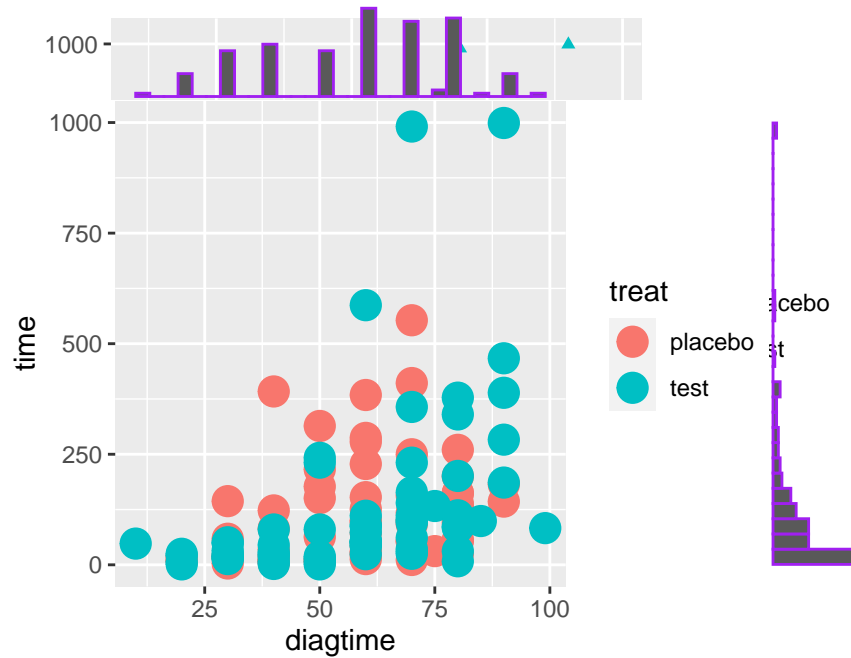




```
#  
ggplot(data = dd, aes(x = diagtime, y = time,  
                      color = treat, shape = treat)) +  
  geom_point() +  
  geom_smooth(se = FALSE)
```



```
# BAD! too many lines
ggplot(data = dd, aes(x = diagtime, y = time,
                      color = treat, shape = treat)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  geom_smooth(se = FALSE)
# classical
p <- ggplot(dd, aes(x = diagtime, y = time, color = treat)) +
  geom_point(size = 5)
# scatter plot + marginal histogram
ggExtra::ggMarginal(p, type = "histogram", color = "purple")
```

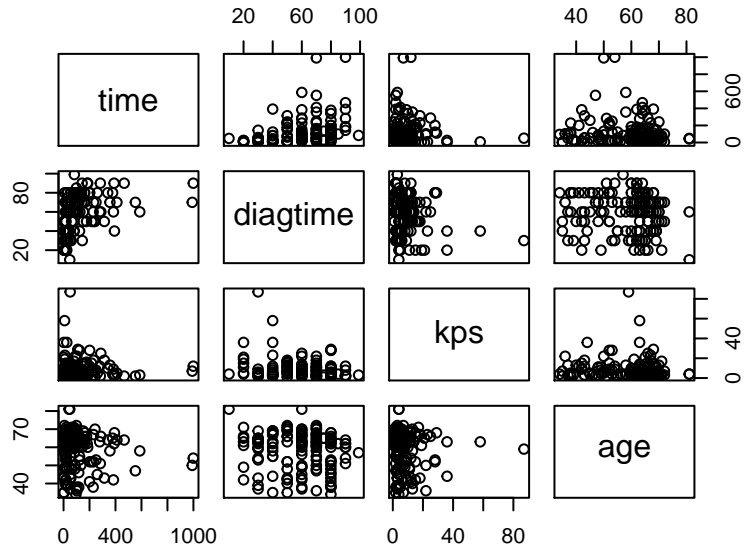


## 5.5.3

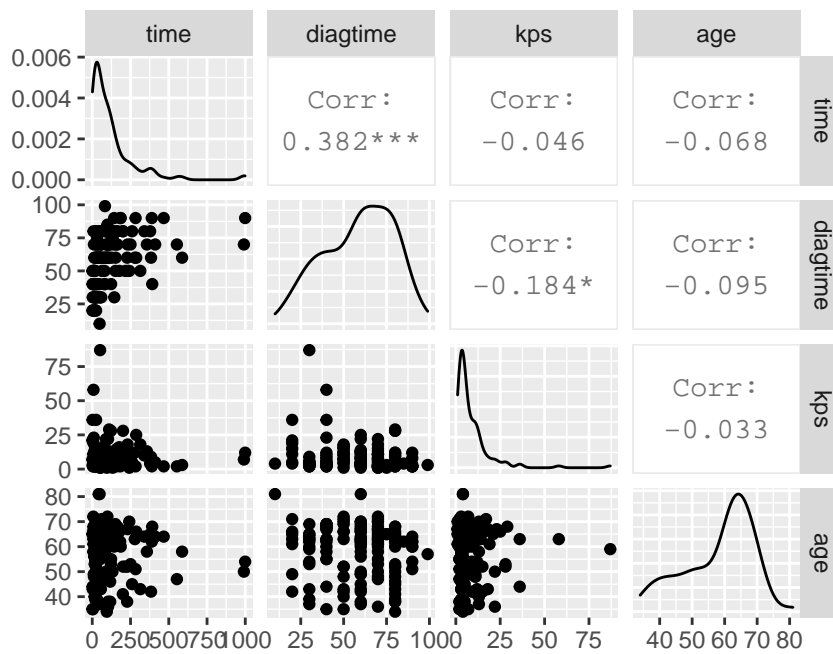
```

• : , , , .
## pairwise scatter plot
## R base
con.df = dd[, c("time", "diagtime", "kps", "age")]
cor.mat = cor(con.df, use = "complete", method = "pearson")
round(cor.mat, 3)
##           time diagtime    kps    age
## time      1.000    0.382 -0.046 -0.068
## diagtime  0.382    1.000 -0.184 -0.095
## kps      -0.046   -0.184  1.000 -0.033
## age      -0.068   -0.095 -0.033  1.000
pairs(con.df)

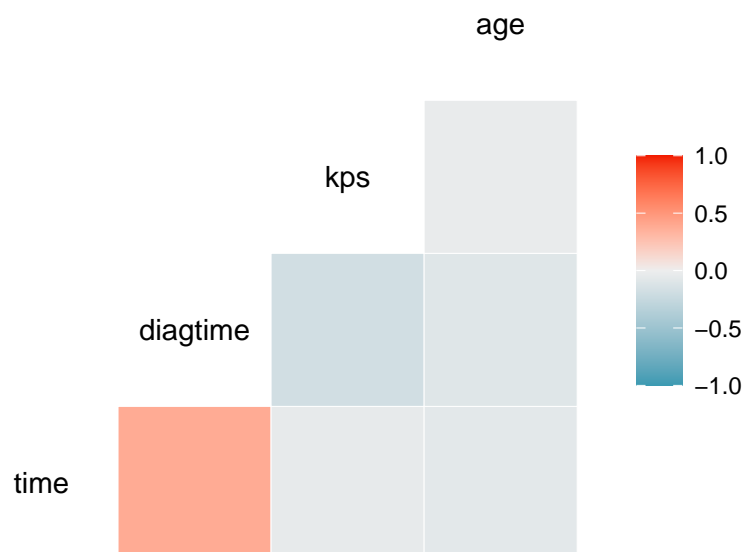
```



```
## ggplot2
library(GGally)
GGally::ggpairs(data = con.df)
```

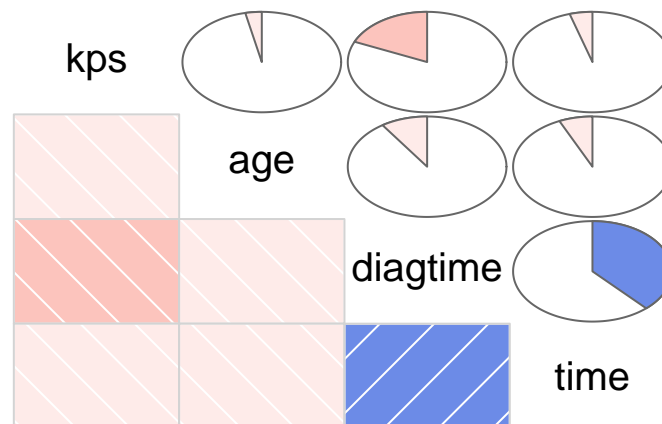


```
Ggally::ggcorr(data = con.df,
  method = c("complete", "pearson"))
```



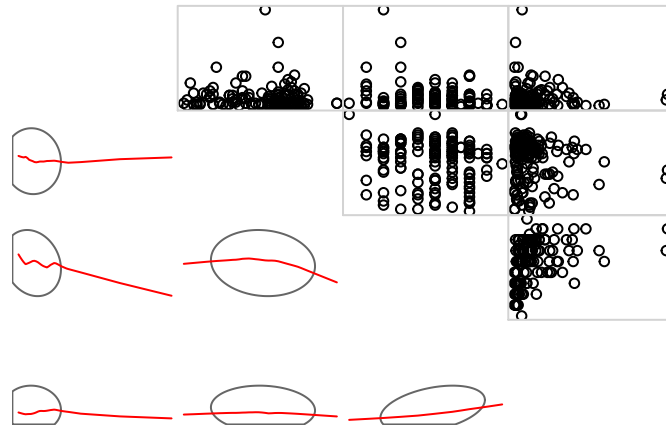
```
## Correlogram
library(corrgram)
corrgram(x = dd,
  order = TRUE,
  lower.panel = panel.shade,
  upper.panel = panel.pie,
  text.panel = panel.txt,
  main = "1. VA Lung Cancer Trial")
```

## 1. VA Lung Cancer Trial



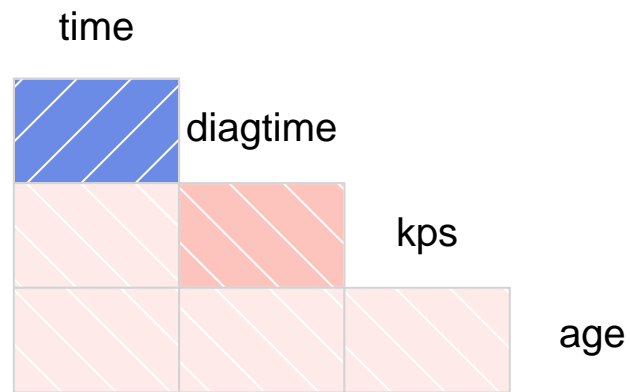
```
corrgram(x = dd,
  order = TRUE,
  lower.panel = panel.ellipse,
  upper.panel = panel.pts,
  text.panel = panel.minmax,
  main = "2. VA Lung Cancer Trial")
```

## 2. VA Lung Cancer Trial



```
corrgram(x = dd,
  order = NULL,
  lower.panel = panel.shade,
  upper.panel = NULL,
  text.panel = panel.txt,
  main = "3. VA Lung Cancer Trial")
```

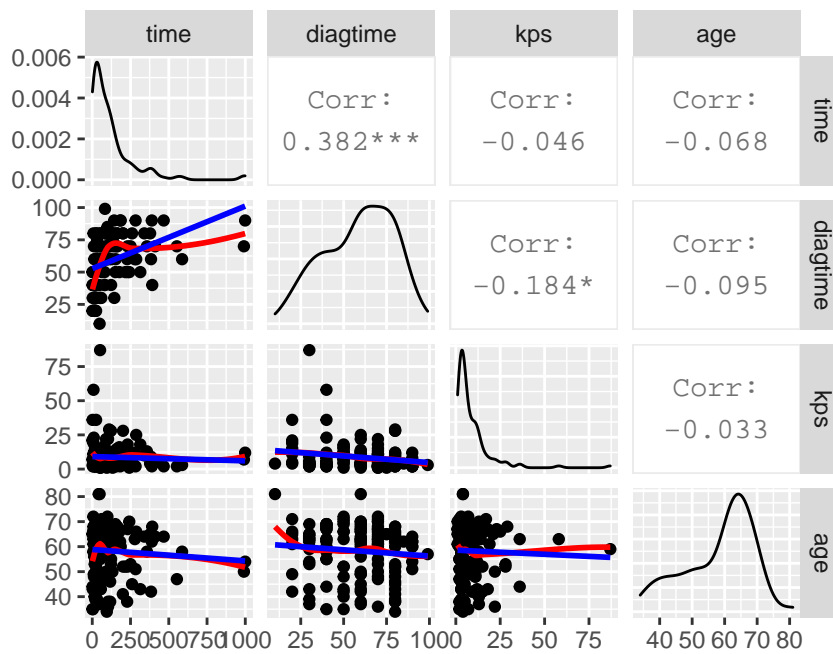
### 3. VA Lung Cancer Trial



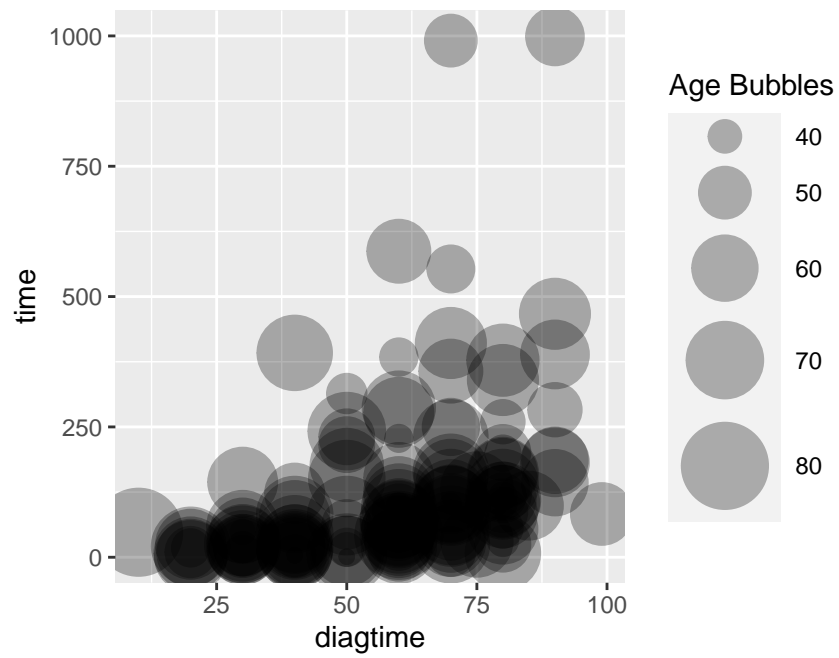
- Try by yourself!

```
# more advanced
my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method = loess, se = FALSE, fill = "red", color = "red", ...) +
    geom_smooth(method = lm, se = FALSE, fill = "blue", color = "blue", ...)
  p
}
GGally::ggpairs(data = con.df,
  lower = list(continuous = my_fn))
```

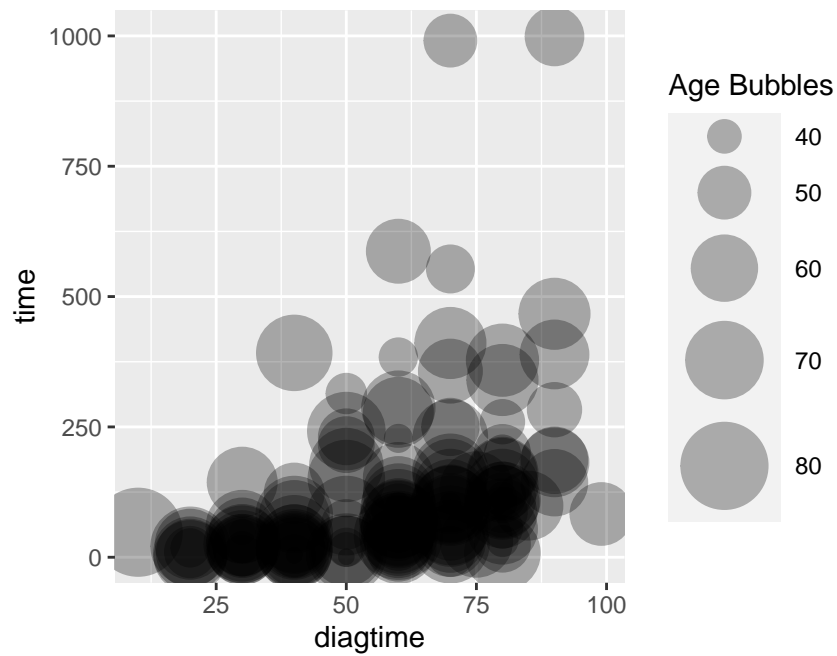




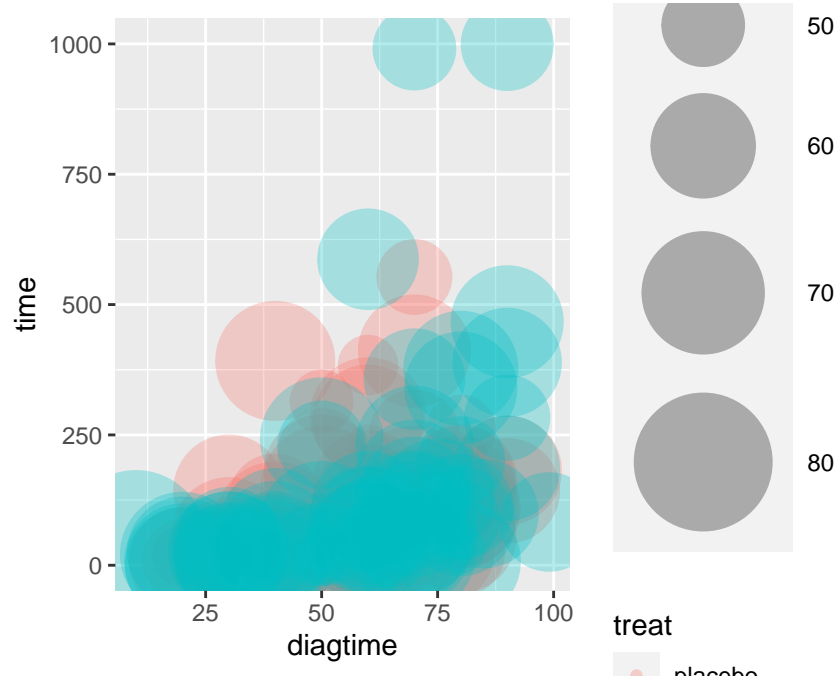
```
## Bubble plot
ggplot(data = dd, aes(x = diagtime, y = time, size = age)) +
  geom_point(alpha = 0.3) +
  scale_size(range = c(.1, 15), name = "Age Bubbles")
```



```
ggplot(data = dd, aes(x = diagtime, y = time, size = age)) +  
  geom_point(alpha = 0.3) +  
  scale_size(range = c(.1, 15), name = "Age Bubbles")
```

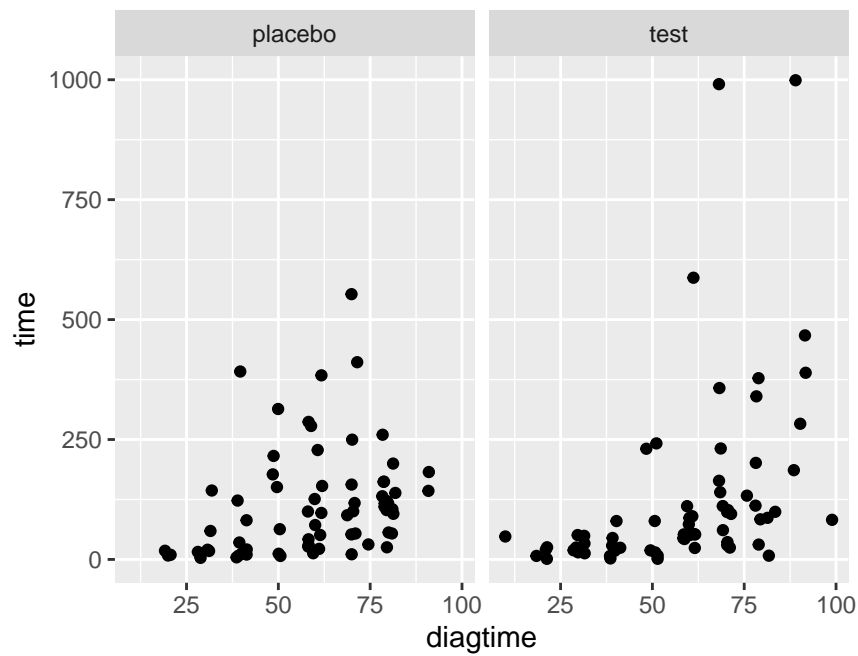


```
ggplot(data = dd, aes(x = diagtime, y = time, size = age, color = treat)) +  
  geom_point(alpha = 0.3) +  
  scale_size(range = c(.1, 24), name = "")
```

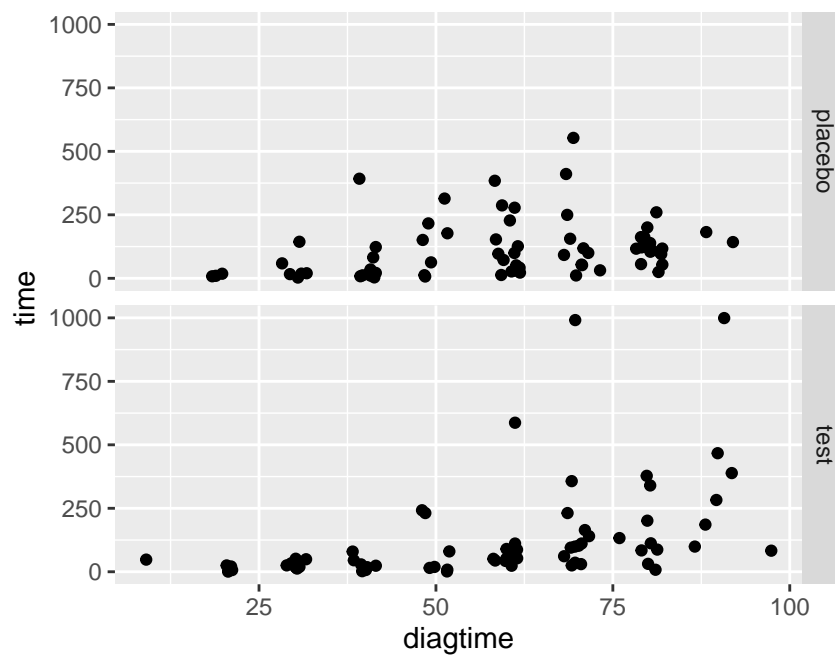


## 5.6

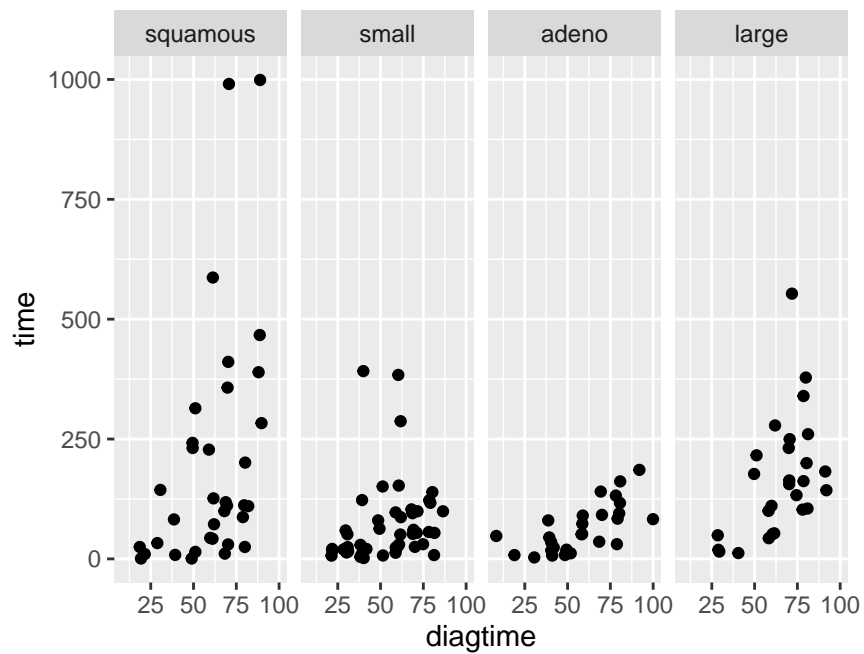
```
•
•
•
# plot by treat
ggplot(data = dd, aes(x = diagtime, y = time)) + geom_jitter() +
  facet_grid(. ~ treat)
```



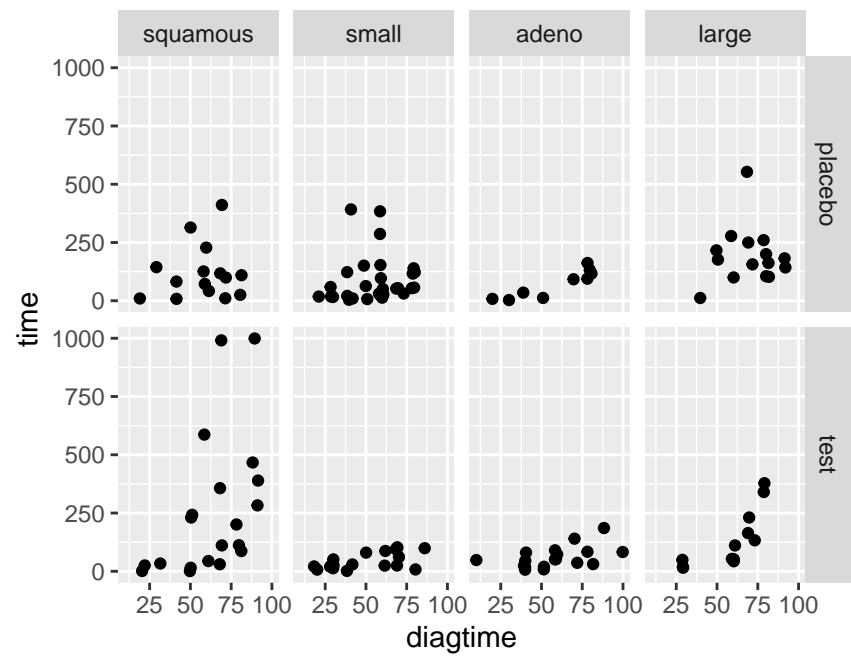
```
ggplot(data = dd, aes(x = diagtime, y = time)) + geom_jitter() +  
  facet_grid(treat ~ .)
```



```
# plot by cellcode  
ggplot(data = dd, aes(x = diagtime, y = time)) + geom_jitter() +  
  facet_grid(. ~ cellcode)
```



```
# two factors  
ggplot(data = dd, aes(x = diagtime, y = time)) + geom_jitter() +  
  facet_grid(treat ~ cellcode)
```







# Chapter 6

`{R}` (function), , , , , , , `{R}` .  
 (argument).  
`{R}` (base) , , `{R}` (contribution) , `{R}` . ,  
`mean()`, `var()`, `sd()`, `log()` .

## 6.1

(argument) , , (formals). , , (required  
 argument), , (optional argument), (ellipsis argument)  
 , , , `{R}` . , `log()` :

`log(x, base = exp(1))`

`log()` `{R}` , `x` , . `base = exp(1)` , ,  
`log()` `e` , , , `2` , `log(x, base = 2)`.

```
## basic function
x.vec = c(1:5)
x.vec          # show x.vec
## [1] 1 2 3 4 5
mean(x = x.vec) # function mean() calculate mean, return a scalar
## [1] 3
var(x = x.vec)  # function mean() calculate variance
## [1] 2.5
sd(x.vec)       # function mean() calculate standard deviation
## [1] 1.581
summary(x.vec)  # summarized statistics
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         1       2       3       3       4       5
log(x = x.vec)  # take log for all elements in vector x.vec
```

```
## [1] 0.0000 0.6931 1.0986 1.3863 1.6094
## log function
x.vec <- c(1, 2, 3, 4, 5)
log(x = x.vec)
## [1] 0.0000 0.6931 1.0986 1.3863 1.6094
log(x = x.vec, base = 2)
## [1] 0.000 1.000 1.585 2.000 2.322
```

## 6.2

```
{R}          S3 classes      S4 classes      ,      ,      {R}          ,      S3 classes
.            , function.name(),                .            methods("function.name"),
getAnywhere("function.name"), stats::function.name ,      sd,
sd          .

## methods()
sd
## function (x, na.rm = FALSE)
## sqrt(var(if (is.vector(x) || is.factor(x)) x else as.double(x),
##      na.rm = na.rm))
## <bytecode: 0x0000000014558130>
## <environment: namespace:stats>
t
## function (x)
## UseMethod("t")
## <bytecode: 0x0000000010d8dd38>
## <environment: namespace:base>
methods(t)
## [1] t,ANY-method          t,CsparseMatrix-method t,dgCMatrix-method
## [4] t,dgeMatrix-method      t,diagonalMatrix-method t,dppMatrix-method
## [7] t,dsCMatrix-method      t,dspMatrix-method      t,dsTMatrix-method
## [10] t,dsyMatrix-method      t,dtpMatrix-method      t,dtrMatrix-method
## [13] t,dtTMatrix-method      t,indMatrix-method      t,lgeMatrix-method
## [16] t,lspMatrix-method      t,lsTMatrix-method      t,lsyMatrix-method
## [19] t,ltpMatrix-method      t,ltrMatrix-method      t,ltTMatrix-method
## [22] t,Matrix-method         t,ngeMatrix-method      t,nsdMatrix-method
## [25] t,nsTMatrix-method      t,nsyMatrix-method      t,ntpMatrix-method
## [28] t,ntrMatrix-method      t,ntTMatrix-method      t,pMatrix-method
## [31] t,RsparseMatrix-method  t,sparseVector-method   t,TsparseMatrix-method
## [34] t.data.frame            t.default                t.fractions*
## [37] t.gtable*               t.trellis*               t.ts*
## [40] t.vctrs_sclr*           t.vctrs_vctr*
## see '?methods' for accessing help and source code
methods(class = "ts")
## [1] [          [<-          aggregate      as.data.frame as_tibble
```

```
## [6] cbind      coerce      cycle      diff      diffinv
## [11] filter     initialize kernapply  lines     Math
## [16] Math2      monthplot  na.omit    Ops       plot
## [21] print      show       slotsFromS3 t         time
## [26] window     window<-
## see '?methods' for accessing help and source code
```

```
S4 classes , showClass("function.name"), showMethods("function.name"),
getMethod("function.name"), selectMethod(), existsMethod(),
hasMethod(), removeClass(), removeMethod(), getClass(), getSlots(),
slotNames(), slot(). , .
```

```
download.packages(pkgs = "package.name",
                  destdir = "C:/RData",
                  type = "source")
```

## 6.3

```
{R} , : , sequence(), rep() .
```

### 6.3.1 : seq() sequence()

```
, , [1,2,3,4,5], [1,3,5,7,9] , : ( ), seq() sequence()
.
```

```
## :
1:5
## [1] 1 2 3 4 5
5:1
## [1] 5 4 3 2 1
- 1:3
## [1] -1 0 1 2 3
```

```
seq() sequence() , , .
```

```
seq(from = 1, to = 1,
    by = ((to - from)/(length.out - 1)),
    length.out = NULL,
    along.with = NULL, ...)
```

- from = 1
- to = 1
- by
- length.out ( )

```
## seq()
seq(from = 1, to = 5, by = 0.5)
```

```
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
seq(1, 5, 0.5)
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
seq(1, 5, length = 3)
## [1] 1 3 5
seq(from = 0, to = 1, by = 0.1)
## [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
seq(from = 0, to = 2, by = 0.33)
## [1] 0.00 0.33 0.66 0.99 1.32 1.65 1.98
sequence(c(3, 4, 5))
## [1] 1 2 3 1 2 3 4 1 2 3 4 5
```

## 6.4 : rep()

```
seq() rep(), . :
rep(x, times = 1, length.out = NA, each = 1)
```

- x .
- times x .
- each x .
- length.out = NA x .

```
## rep()
rep(0, times = 3)
## [1] 0 0 0
rep(1, 5)
## [1] 1 1 1 1 1
x.vec <- c(4, 5, 6)
rep(x.vec, times = 2)
## [1] 4 5 6 4 5 6
rep(x.vec, each = 2)
## [1] 4 4 5 5 6 6
rep(x.vec, each = 2, times = 3)
## [1] 4 4 5 5 6 6 4 4 5 5 6 6 4 4 5 5 6 6
rep(x.vec, times = c(2, 2, 2))
## [1] 4 4 5 5 6 6
rep(x.vec, times = c(1, 2, 3))
## [1] 4 5 5 6 6 6
rep(x.vec, each = 2, len = 4) # first 4 only.
## [1] 4 4 5 5
```

## 6.5 Arithmetic Computing Function

{R} (arithmetic function), , , Gamma , Beta , , , , .

Table 6.1: Arithmetic Computing Function

|                         |  |
|-------------------------|--|
| -                       | (Substraction, can be unary or binary)                       |
| +                       | (Addition, can be unary or binary)                           |
| !                       | (Unary not)  |
|                         | (Multiplication, binary)                                     |
| /                       | (Division, binary)   |
| ^                       | (Exponentiation, binary)                                     |
| %%                      | (Modulus, binary)  |
| %/%                     | (Integer divide, binary)                                     |
| %*%                     | (Matrix product, binary)                                     |
| %o%                     | (Outer product, binary)                                      |
| %x%                     | Kronecker (Kronecker product, binary)                        |
| %in%                    | (Matching operator, binary, in model formulae: nesting)      |
| round(x, digits = 0)    | ( )  |
| signif(x, digits = 6)   | ( )  |
| trunc(x)                | $x$ , 0  |
| ceiling(x)              | $x$  |
| floor(x)                | $x$  |
| sign(x)                 | $x$ , 1, 0, -1.  |
| abs(x)                  | $x$  |
| sqrt(x)                 | $\sqrt{x}$   |
| exp(x)                  | $e^x$  |
| expm1(x)                | $ x  \ll 1$ , $e^x - 1$                                      |
| log(x)                  | $\log(x)$  |
| log10(x)                | $\log_{10}(x)$   |
| log2(x)                 | $\log_2(x)$  |
| logb(x, base = z)       | $\log_z(x)$  |
| log1p(x)                | $ x  \ll 1$ , $\log(1 + x)$                                  |
| gamma(x)                | $\Gamma(x) = (x - 1)! = \int_0^\infty t^{(x-1)} \exp(-t) dt$ |
| lgamma(x)               | $\log_e[\Gamma(x)]$  |
| beta(a, b)              | $B(a, b) = (\Gamma(a)\Gamma(b)) / (\Gamma(a + b))$           |
|                         | $\$ = \int_0^1 t^{a-1} (1-t)^{b-1} dt$                       |
| lbeta(a, b)             | $\log_e[B(a, b)]$  |
| digamma(x)              | $\frac{d}{dx} \log_e[\Gamma(x)]$                             |
| trigamma(x)             | $\frac{d^2}{dx^2} \log_e[\Gamma(x)]$                         |
| psigamma(x, deriv = 0)  | $\frac{d^p}{dx^p} \log_e[\Gamma(x)]$                         |
| sin(x) cos(x) tan(x)    | (trigonometric functions)                                    |
| asin(x) acos(x) atan(x) | (inverse functions)  |

---

|                       |                       |                       |                                |
|-----------------------|-----------------------|-----------------------|--------------------------------|
| <code>sinh(x)</code>  | <code>cosh(x)</code>  | <code>tanh(x)</code>  | (hyperbolic functions)         |
| <code>asinh(x)</code> | <code>acosh(x)</code> | <code>atanh(x)</code> | (inverse hyperbolic functions) |

---

```
## Arithmetic Computing
## rounding
(x.vec <- 0.5 + c(-2:2))
## [1] -1.5 -0.5 0.5 1.5 2.5
round(x.vec) # IEEE rounding
## [1] -2 0 0 2 2
(y.vec <- seq(-2, 2, by = 0.5))
## [1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
(y.round <- round(y.vec)) # IEEE rounding
## [1] -2 -2 -1 0 0 0 1 2 2
(y.trunc <- trunc(y.vec))
## [1] -2 -1 -1 0 0 0 1 1 2
(y.signif <- signif(y.vec))
## [1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0
(y.ceil <- ceiling(y.vec))
## [1] -2 -1 -1 0 0 1 1 2 2
(y.floor <- floor(y.vec))
## [1] -2 -2 -1 -1 0 0 1 1 2
cbind(y.vec, y.round, y.trunc, y.signif, y.ceil, y.floor)
##      y.vec y.round y.trunc y.signif y.ceil y.floor
## [1,] -2.0     -2     -2     -2.0     -2     -2
## [2,] -1.5     -2     -1     -1.5     -1     -2
## [3,] -1.0     -1     -1     -1.0     -1     -1
## [4,] -0.5      0      0     -0.5      0     -1
## [5,]  0.0      0      0      0.0      0      0
## [6,]  0.5      0      0      0.5      1      0
## [7,]  1.0      1      1      1.0      1      1
## [8,]  1.5      2      1      1.5      2      1
## [9,]  2.0      2      2      2.0      2      2
#
(x.vec <- 0.5 + c(-2:3))
## [1] -1.5 -0.5 0.5 1.5 2.5 3.5
round(x.vec) # IEEE rounding
## [1] -2 0 0 2 2 4
(y.vec <- seq(-2, 3, by = 0.5))
## [1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
(y.round <- round(y.vec)) # IEEE rounding
## [1] -2 -2 -1 0 0 0 1 2 2 2 3
(y.trunc <- trunc(y.vec))
## [1] -2 -1 -1 0 0 0 1 1 2 2 3
```

```

(y.signif <- signif(y.vec))
## [1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
(y.ceil <- ceiling(y.vec))
## [1] -2 -1 -1 0 0 1 1 2 2 3 3
(y.floor <- floor(y.vec))
## [1] -2 -2 -1 -1 0 0 1 1 2 2 3
cbind(y.vec, y.round, y.trunc, y.signif, y.ceil, y.floor)
##      y.vec y.round y.trunc y.signif y.ceil y.floor
## [1,] -2.0      -2      -2      -2.0     -2      -2
## [2,] -1.5      -2      -1      -1.5     -1      -2
## [3,] -1.0      -1      -1      -1.0     -1      -1
## [4,] -0.5       0       0      -0.5      0      -1
## [5,] 0.0       0       0       0.0      0       0
## [6,] 0.5       0       0       0.5      1       0
## [7,] 1.0       1       1       1.0      1       1
## [8,] 1.5       2       1       1.5      2       1
## [9,] 2.0       2       2       2.0      2       2
## [10,] 2.5      2       2       2.5      3       2
## [11,] 3.0      3       3       3.0      3       3
#
(y.vec <- seq(-2, 3, by = 0.5))
## [1] -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0
y.vec[trunc(y.vec) != floor(y.vec)]
## [1] -1.5 -0.5
y.vec[round(y.vec) != floor(y.vec + 0.5)]
## [1] -1.5 0.5 2.5
#
(z.vec <- pi * 100 ^ (-1:3))
## [1] 3.142e-02 3.142e+00 3.142e+02 3.142e+04 3.142e+06
round(z.vec, 3)
## [1] 3.100e-02 3.142e+00 3.142e+02 3.142e+04 3.142e+06
signif(z.vec, 3)
## [1] 3.14e-02 3.14e+00 3.14e+02 3.14e+04 3.14e+06
#
## sign() abs()
sign(pi) # == 1
## [1] 1
sign(-2:3) # -1 -1 0 1 1 1
## [1] -1 -1 0 1 1 1
abs(-2:3)
## [1] 2 1 0 1 2 3
#
## log(), exp() calculation
(x.vec <- 1:3)
## [1] 1 2 3

```

```

log(exp(x.vec))
## [1] 1 2 3
(y.vec <- 10 ^ (x.vec))
## [1] 10 100 1000
log10(y.vec)
## [1] 1 2 3
log10(1e7) # = 7
## [1] 7
#
## options(digits, scipen)
options(digits = 4, scipen = 0)
z.vec <- pi * 100^(-1:3)
print(z.vec / 1000, digits = 4)
## [1] 3.142e-05 3.142e-03 3.142e-01 3.142e+01 3.142e+03
options(digits = 4, scipen = 100)
print(z.vec / 1000, digits = 4)
## [1] 0.00003142 0.00314159 0.31415927 31.41592654 3141.59265359
#
options(digits = 4, scipen = 100)
x.vec <- 100 ^ -(1 + 2 * 1:3)
cbind(
  x = x.vec,
  log1px = log(1 + x.vec),
  log1p = log1p(x.vec),
  exp = exp(x.vec) - 1,
  expm1 = expm1(x.vec)
)
##               x               log1px               log1p
## [1,] 0.000001000000000 0.000000999999499918 0.00000099999950
## [2,] 0.00000000010000 0.000000000100000008 0.00000000010000
## [3,] 0.000000000000001 0.0000000000000009992 0.000000000000001
##               exp               expm1
## [1,] 0.000001000000499962 0.000001000000050
## [2,] 0.000000000100000008 0.00000000010000
## [3,] 0.0000000000000009992 0.000000000000001
#
options(digits = 4, scipen = 0)
x.vec <- 100^(-(1 + 2 * 1:3))
cbind(
  x = x.vec,
  log1px = log(1 + x.vec),
  log1p = log1p(x.vec),
  exp = exp(x.vec) - 1,
  expm1 = expm1(x.vec)
)

```



```
##          x      log1px log1p      exp expm1
## [1,] 1e-06 1.000e-06 1e-06 1.000e-06 1e-06
## [2,] 1e-10 1.000e-10 1e-10 1.000e-10 1e-10
## [3,] 1e-14 9.992e-15 1e-14 9.992e-15 1e-14
```

## 6.6 : choose() factorial()

{R} choose(), lchoose(), factorial(), lfactorial(), .

- $\text{choose}(n, k) = \binom{n}{k}$
- $\text{factorial}(x) = x!$
- $k$  .
- $x \geq n$  .
- $\text{factorial}()$ ,  $\text{lfactorial}()$  .

```
## combination
## choose()
choose(n = 5, k = 2)
## [1] 10
log(choose(n = 5, k = 2))
## [1] 2.303
lchoose(n = 5, k = 2)
## [1] 2.303
for (n in 0:5)
  print(choose(n, k = 0:n))
## [1] 1
## [1] 1 1
## [1] 1 2 1
## [1] 1 3 3 1
## [1] 1 4 6 4 1
## [1] 1 5 10 10 5 1
## factorial
factorial(x = 100)
## [1] 9.333e+157
log(factorial(x = 100))
## [1] 363.7
lfactorial(x = 100)
## [1] 363.7
lfactorial(x = 10000)
## [1] 82109
factorial(x = c(1, 3, 5))
## [1] 1 6 120
```

## 6.7 : all(), any(), which()

```

all(x) any(x) obj.vec , TRUE FALSE. which()
obj.vec , , . which.max() which.min()
which() .

```

```

all(..., na.rm = FALSE)
any(..., na.rm = FALSE)
which(x, arr.ind = FALSE, useNames = TRUE)

```

```

... . all(x) any(x) (scalar) TRUE FALSE. all(x)
x TRUE? , any(x) x TRUE? which(x) , x
TRUE (index). which(x) arr.ind = TRUE x array (matrix) ,
array (matrix) .

```

```

## all(), any(), which()
(x.vec <- c(-1:2))
## [1] -1 0 1 2
all(x.vec > 0)
## [1] FALSE
any(x.vec > 0)
## [1] TRUE
which(x.vec > 0)
## [1] 3 4
which.max(x.vec)
## [1] 4
which.min(x.vec)
## [1] 1
#
(x.mat <- matrix(c(2, -1, -3,
                  -1, 2, 4,
                  -3, 4, 9),
                nrow = 3, byrow = T))
##      [,1] [,2] [,3]
## [1,]  2  -1  -3
## [2,] -1   2   4
## [3,] -3   4   9
all(x.mat > 0)
## [1] FALSE
any(x.mat > 0)
## [1] TRUE
which(x.mat > 0)
## [1] 1 5 6 8 9
#
which(x.mat %% 2 == 0)
## [1] 1 5 6 8
which(x.mat %% 2 == 0, arr.ind = TRUE)

```

```
##      row col
## [1,]    1  1
## [2,]    2  2
## [3,]    3  2
## [4,]    2  3
```

## 6.8 Ranking and Sorting

`{R}` , `rev()`, `sort()`, `order()` `rank()`.

Table 6.2:

|                       |                |  |
|-----------------------|----------------|--|
| <code>rev(x)</code>   | <code>x</code> | (reverse order)                          |
| <code>rank(x)</code>  | <code>x</code> | (returns the sample ranks of the values) |
|                       |                | <code>ties.method = "average"</code>     |
| <code>sort(x)</code>  | <code>x</code> | (sort a vector or factor, partially)     |
|                       |                | into ascending or descending order).     |
| <code>order(x)</code> | <code>x</code> |  |

```
rev(x)
sort(x, decreasing = FALSE, na.last = NA, ...)
rank(x, na.last = TRUE,
      ties.method = c("average", "first", "last", "random", "max", "min"))
order(x, ..., na.last = TRUE, decreasing = FALSE,
      method = c("shell", "radix"))
```

- :
- `x` `x`.
- `decreasing`:
  - `decreasing = FALSE` `{R}` .
  - `decreasing = TRUE` .
- `na.last`:
  - `na.last = TRUE` `{R}` `NA` .
  - `na.last = FALSE` `{R}` `NA` .
  - `na.last = NA` `{R}` `NA` .
- `rev(x)` `z`, `x` .
- `sort(x)` `z`, `x` .
- `%>% rank(x)` `z`, `x` , `x` (rank).

```

• "average":      .
• "first":        .
• "last":         .
• "random":       .
• "max":          .
• "min":          .

## reverse, rank, sort and order
## rev(): reverse elements
x.vec <- c(7, 7, 7, 6, 10, 9, 9, 9, NA, 8)
rev(x.vec)
## [1] 8 NA 9 9 9 10 6 7 7 7
## sort(): from the smallest to the largest
sort(x.vec)
## [1] 6 7 7 7 8 9 9 9 10
## rank():
rank(x.vec, na.last = TRUE)
## [1] 3 3 3 1 9 7 7 7 10 5
rank(x.vec, na.last = FALSE)
## [1] 4 4 4 2 10 8 8 8 1 6
set.seed(1)
rank(x.vec, ties.method = "average")
## [1] 3 3 3 1 9 7 7 7 10 5
rank(x.vec, ties.method = "first")
## [1] 2 3 4 1 9 6 7 8 10 5
rank(x.vec, ties.method = "last")
## [1] 4 3 2 1 9 8 7 6 10 5
rank(x.vec, ties.method = "random")
## [1] 2 3 4 1 9 7 8 6 10 5
rank(x.vec, ties.method = "max")
## [1] 4 4 4 1 9 8 8 8 10 5
rank(x.vec, ties.method = "min")
## [1] 2 2 2 1 9 6 6 6 10 5
## order(): retrun index
## x.vec[] is the smallest one
order(x.vec)
## [1] 4 1 2 3 10 6 7 8 5 9
x.vec[order(x.vec)]
## [1] 6 7 7 7 8 9 9 9 10 NA
## rank(): ties.method = "average"
x <- c(7, 9, 6, 7, 8, NA)
sort(x, na.last = FALSE)
## [1] NA 6 7 7 8 9

```

```
rank(x, ties.method = "average", na.last = TRUE)
## [1] 2.5 5.0 1.0 2.5 4.0 6.0
(x.ord <- order(x, na.last = FALSE))
## [1] 6 3 1 4 5 2
x[x.ord] # = sort(x)
## [1] NA 6 7 7 8 9
```

```
{R}      NA      , na.last = TRUE {R}      ,      NA      .
```

## 6.9

```
{R}      is.object(), is.na(), is.vector() ,      .
```

```
{R}      as.object(), as.vector(), as.matrix() ,      .
```

```
## is() and as()
# vector
x.vec <- c(1 / 1, 1 / 2, 1 / 3, 1 / 4, 1 / 5)
x.vec
## [1] 1.0000 0.5000 0.3333 0.2500 0.2000
is.vector(x.vec)
## [1] TRUE
is.character(x.vec)
## [1] FALSE
x.vec <- as.character(x.vec)
x.vec
## [1] "1"                "0.5"                "0.3333333333333333"
## [4] "0.25"               "0.2"
##
b.df <- as.data.frame(matrix(c(1:24), nrow = 6, byrow = T))
is.matrix(b.df)
## [1] FALSE
b.mat <- as.matrix(b.df)
b.mat
##      V1 V2 V3 V4
## [1,]  1  2  3  4
## [2,]  5  6  7  8
## [3,]  9 10 11 12
## [4,] 13 14 15 16
## [5,] 17 18 19 20
## [6,] 21 22 23 24
b.mat <- as.vector(b.mat)
b.mat
## [1]  1  5  9 13 17 21  2  6 10 14 18 22  3  7 11 15 19 23  4  8 12 16 20 24
```



# Chapter 7

{R} , , {R} , , sum(), cumsum(), diff(), prod(), cumprod(); , , mean(), median(), var(), sd(), range(), min(), max(), quantile(), sample() .

{R} , , {R} , , na.omit() , , , mean(na.omit(x)). , , na.rm = T , mean(x, na.rm = T) , , .

Table 7.1:

|             |                          |   |
|-------------|--------------------------|---|
| sum(x)      | (scalar)                 | $y = \sum_i x_i$  |
| cumsum(x)   | (vector)                 | $z_j = \sum_{i \leq j} x_i$                                 |
| diff(x)     | <code>x[i+1]-x[i]</code> | $z_i = x_{i+1} - x_i$                                       |
| lag(x, k)   | <code>x[i-k]</code>      | $z_i = x_{i-k}, \quad \text{x[i]} \quad \text{x[i-k]}$      |
| lead(x, k)  | <code>x[i+k]</code>      | $z_i = x_{i+k}, \quad \text{x[i]} \quad \text{x[i+k]}$      |
| prod(x)     | (product)                | $y = \prod_i x_i$   |
| cumprod(x)  |                          | $\$z\_j = \_ \{i \ j\} \ x\_i$                              |
| mean(x)     | (mean)                   | $\bar{x} = \frac{1}{n} \sum_i x_i$                          |
| median(x)   | (median)                 | 0.5 quantile, 50 <sup>th</sup> percentile                   |
| var(x)      | ,                        | $s^2 = \frac{1}{n-1} \sum_i (x_i - \bar{x})^2$              |
| sd(x)       | (SD)                     | $s = \sqrt{s^2}$  |
| range(x)    | (range)                  | (min( $x$ ), max( $x$ ))                                    |
| min(x)      |                          | min( $x$ )  |
| max(x)      |                          | max( $x$ )  |
| quantile(x) |                          |   |
| fivenum(x)  |                          | (five-number summary)<br>(min, $Q_1$ , median, $Q_3$ , max) |
| sample(x)   |                          | random sample   |

## 7.1

```

{R}      ,      . z <- range(x)      ,      (min(x),max(x));      min(x),
max(x);      quantile(),      quantile(x, probs = c(0.05, 0.25, 0.5,
0.75, 0.95). fivenum(x)  x  (min, Q1, median, Q3, max).

## basic descriptive statistics
x <- seq(-2, 3, 0.3)
x
## [1] -2.0 -1.7 -1.4 -1.1 -0.8 -0.5 -0.2  0.1  0.4  0.7  1.0  1.3  1.6  1.9  2.2
## [16]  2.5  2.8
sum(x)
## [1] 6.8
cumsum(x)
## [1] -2.0 -3.7 -5.1 -6.2 -7.0 -7.5 -7.7 -7.6 -7.2 -6.5 -5.5 -4.2 -2.6 -0.7  1.5
## [16]  4.0  6.8
diff(x)
## [1] 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3
prod(x)
## [1] -0.7138
cumprod(x)
## [1] -2.00000  3.40000 -4.76000  5.23600 -4.18880  2.09440 -0.41888 -0.04189
## [9] -0.01676 -0.01173 -0.01173 -0.01525 -0.02440 -0.04635 -0.10197 -0.25493
## [17] -0.71381
mean(x)
## [1] 0.4
median(x)
## [1] 0.4
var(x)
## [1] 2.295
sd(x)
## [1] 1.515
range(x)
## [1] -2.0  2.8
min(x)
## [1] -2
max(x)
## [1] 2.8
## quantile
y <- quantile(x, probs = c(0.05, 0.25, 0.5, 0.75, 0.95))
y
##      5%      25%      50%      75%      95%
## -1.76 -0.80  0.40  1.60  2.56
# IQR: inter-quantile range
iqr = y[4] - y[2]
iqr

```



```
## 75%
## 2.4
## five number summary
fivenum(x)
## [1] -2.0 -0.8 0.4 1.6 2.8
# missing values
x[3] <- NA
x[7] <- NA
x
## [1] -2.0 -1.7 NA -1.1 -0.8 -0.5 NA 0.1 0.4 0.7 1.0 1.3 1.6 1.9 2.2
## [16] 2.5 2.8
mean(x)
## [1] NA
mean(na.omit(x))
## [1] 0.56
mean(x, na.rm = T)
## [1] 0.56
var(x, na.rm = T)
## [1] 2.338
```

## 7.2

, (contingency table), {R} , (contingency table),  
 , table(), xtabs(), as.table(), is.table(); ftable(), read.ftable(),  
 write.ftable(); as.data.frame(); margin.table(), prop.table(),  
 addmargins() . , {R} , , xtable, vcd\*\*, reshape2, plyr, dplyr, tidyr, tidyverse ,  
 . {R} , , \*\* \*\* . , Epi, epiBasix, epiDisplay( epicalc), epifit, epiR, epiTools, RCO  
 , , , .  
 2 , (a) (individual data, micro data, case data); (b)  
 (aggregated data, macro data, summarized data, ecological data).  
 (subject, individual), , (raw data,  
 primary data, original data). , , , , ,  
 (secondary data).

, , , BMI , , (ecological fallacy),  
 ( ) , ( ) , .

### 7.2.1 : table(), xtabs()

```
table(), xtabs(), , , , table()
_contingency table_ {R} (class) table . as.table()
. as.matrix() . as.data.frame()
as.data.frame() xtabs() . is.table() . table()
```

```

      xtabs()      ,      (model formula)      .      as.data.frame()
xtabs()      ,      .
table(variable_name, ...)
xtabs(formula, data)

```

```

• formula:      .
• data:      .
• na.action = "na.omit":      .
• exclude:      ,      .
• useNA:      .
  - "no":      .
  - "ifany":      ,      (count)      .
  - "always":      1      .      (count) 0      1      .
Prentice (1973)      ,      ,      ,      %      Veteran's Administration
      ,      ,      ,      .      survVATrial.csv.

```

---

|                 |                              |
|-----------------|------------------------------|
| treat (therapy) | : 0 = ; 1 =                  |
| cellcode        | ; 1 = ; 2 = ; 3 = ; 4 =      |
| time            | , ,                          |
| censor          | : 0 = ; 1 =                  |
| diagtime        | Karnofsky performance score, |
| diagtime        | ,                            |
| age             | ( )                          |
| prior           | ; 0 = ; 1 =                  |

---

```

dd <- read.table("./Data/survVATrial.csv",
  header = TRUE,
  sep = ",",
  quote = "\"\"",
  dec = ".",
  row.names = NULL,
  # col.names,
  as.is = TRUE,
  # as.is = !stringsAsFactors,
  na.strings = c(".", "NA"))
head(dd)
##   treat cellcode time censor diagtime kps age prior
## 1     0         1   72      1       60  7  69     0
## 2     0         1  411      1       70  5  64     10
## 3     0         1  228      1       60  3  38     0
## 4     0         1  126      1       60  9  63     10

```

```
## 5      0      1 118      1      70 11 65      10
## 6      0      1 10      1      20 5 49      0
str(dd)
## 'data.frame':      137 obs. of  8 variables:
## $ treat   : int  0 0 0 0 0 0 0 0 0 0 ...
## $ cellcode: int  1 1 1 1 1 1 1 1 1 1 ...
## $ time    : int  72 411 228 126 118 10 82 110 314 100 ...
## $ censor  : int  1 1 1 1 1 1 1 1 1 0 ...
## $ diagtime: int  60 70 60 60 70 20 40 80 50 70 ...
## $ kps     : int  7 5 3 9 11 5 10 29 18 6 ...
## $ age     : int  69 64 38 63 65 49 69 68 43 70 ...
## $ prior   : int  0 10 0 10 10 0 10 0 0 0 ...
dd$treat <- factor(dd$treat, labels = c("placebo", "test"))
dd$cellcode <- factor(dd$cellcode,
                      labels = c("squamous", "small", "adeno", "large"))
dd$censor <- factor(dd$censor, labels = c("survival", "dead"))
dd$prior <- factor(dd$prior, labels = c("no", "yes"))
head(dd)
##      treat cellcode time censor diagtime kps age prior
## 1 placebo squamous  72   dead      60   7  69   no
## 2 placebo squamous 411   dead      70   5  64   yes
## 3 placebo squamous 228   dead      60   3  38   no
## 4 placebo squamous 126   dead      60   9  63   yes
## 5 placebo squamous 118   dead      70  11  65   yes
## 6 placebo squamous  10   dead      20   5  49   no
str(dd)
## 'data.frame':      137 obs. of  8 variables:
## $ treat   : Factor w/ 2 levels "placebo","test": 1 1 1 1 1 1 1 1 1 1 ...
## $ cellcode: Factor w/ 4 levels "squamous","small",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ time    : int  72 411 228 126 118 10 82 110 314 100 ...
## $ censor  : Factor w/ 2 levels "survival","dead": 2 2 2 2 2 2 2 2 2 1 ...
## $ diagtime: int  60 70 60 60 70 20 40 80 50 70 ...
## $ kps     : int  7 5 3 9 11 5 10 29 18 6 ...
## $ age     : int  69 64 38 63 65 49 69 68 43 70 ...
## $ prior   : Factor w/ 2 levels "no","yes": 1 2 1 2 2 1 2 1 1 1 ...
## one-way table
## table()
table(dd$censor)
##
## survival      dead
##          9      128
table(dd$cellcode)
##
## squamous      small      adeno      large
##          35          48          27          27
```

```
## xtabs()
xtabs(~ censor, data = dd)
## censor
## survival      dead
##          9      128
## two-way table()
## table()
dd.2tab = table(dd$cellcode, dd$censor)
dd.2tab
##
##          survival dead
## squamous         4   31
## small            3   45
## adeno            1   26
## large            1   26
class(dd.2tab)
## [1] "table"
## xtabs()
dd.2xtabs = xtabs(~ cellcode + censor, data = dd)
dd.2xtabs
##          censor
## cellcode survival dead
## squamous         4   31
## small            3   45
## adeno            1   26
## large            1   26
class(dd.2xtabs)
## [1] "xtabs" "table"
## three-way table()
## table()
dd.3tab = table(dd$treat, dd$censor, dd$cellcode)
dd.3tab
## , , = squamous
##
##          survival dead
## placebo         2   13
## test            2   18
##
## , , = small
##
##          survival dead
## placebo         2   28
## test            1   17
```

```
##
## , , = adeno
##
##
##      survival dead
## placebo      0   9
## test        1  17
##
## , , = large
##
##
##      survival dead
## placebo      1  14
## test        0  12
## xtabs()
dd.3xtabs = xtabs(~ treat + censor + cellcode, data = dd)
dd.3xtabs
## , , cellcode = squamous
##
##      censor
## treat      survival dead
## placebo      2  13
## test        2  18
##
## , , cellcode = small
##
##      censor
## treat      survival dead
## placebo      2  28
## test        1  17
##
## , , cellcode = adeno
##
##      censor
## treat      survival dead
## placebo      0   9
## test        1  17
##
## , , cellcode = large
##
##      censor
## treat      survival dead
## placebo      1  14
## test        0  12
```

### 7.2.2 : ftable()

```

table() xtabs() list , , ftable(), , , ,
(flat contingency table), ftable (class) , ( , column) ,
, (row) (level). ftable() ftable , {R} , table()
xtabs() _contingency table_ .

```

```

## three-way table()
## ftable()
dd.3ftab = ftable(dd$cellcode, dd$treat, dd$censor)
dd.3ftab
##                survival dead
##
## squamous placebo      2    13
##          test       2    18
## small  placebo      2    28
##          test       1    17
## adeno  placebo      0     9
##          test       1    17
## large  placebo      1    14
##          test       0    12

```

### 7.2.3 : margin.table(), prop.table()

```

margin.table() (class) table , (array, matrix)
(marginal total). prop.table() (array, matrix) , (relative
frequency). addmargins() .

```

```

margin.table() prop.table() (class) ftable .
addmargins() (class) table ftable .

```

```

margin.table(x, margin = NULL)
prop.table(x, margin = NULL)
addmargins(A, margin, ...)

```

- x: table .
- A: table ftable .
- margin: (index/vector), .
  - margin = NULL: (cell count/proportion).
  - margin = 1: (row) (row marginal total/proportion).
- margin = 2: (column) (column marginal total/proportion).
- margin = k: .

```

## one-way table
## table()
dd.1tab = table(dd$cellcode)
dd.1tab

```

```
##
## squamous      small      adeno      large
##          35          48          27          27
margin.table(dd.1tab)
## [1] 137
prop.table(dd.1tab)
##
## squamous      small      adeno      large
##    0.2555    0.3504    0.1971    0.1971
## xtabs()
dd.1xtabs = xtabs(~ censor, data = dd)
margin.table(dd.1xtabs)
## [1] 137
prop.table(dd.1xtabs)
## censor
## survival      dead
##    0.06569    0.93431
## two-way table()
## table()
dd.2tab = table(dd$cellcode, dd$censor)
dd.2tab
##
##          survival dead
## squamous          4   31
## small             3   45
## adeno              1   26
## large              1   26
## cell count total and proportion
margin.table(dd.2tab)
## [1] 137
prop.table(dd.2tab)
##
##          survival      dead
## squamous 0.029197 0.226277
## small    0.021898 0.328467
## adeno     0.007299 0.189781
## large     0.007299 0.189781
## condition on row
margin.table(dd.2tab, margin = 1)
##
## squamous      small      adeno      large
##          35          48          27          27
prop.table(dd.2tab, margin = 1)
##
##          survival      dead
```

```
## squamous 0.11429 0.88571
## small 0.06250 0.93750
## adeno 0.03704 0.96296
## large 0.03704 0.96296
## condition on column
margin.table(dd.2tab, margin = 2)
##
## survival dead
## 9 128
prop.table(dd.2tab, margin = 2)
##
## survival dead
## squamous 0.4444 0.2422
## small 0.3333 0.3516
## adeno 0.1111 0.2031
## large 0.1111 0.2031
## xtabs()
dd.2xtabs = xtabs(~ cellcode + censor, data = dd)
dd.2xtabs
## censor
## cellcode survival dead
## squamous 4 31
## small 3 45
## adeno 1 26
## large 1 26
## cell count total and proportion
margin.table(dd.2xtabs)
## [1] 137
prop.table(dd.2xtabs)
## censor
## cellcode survival dead
## squamous 0.029197 0.226277
## small 0.021898 0.328467
## adeno 0.007299 0.189781
## large 0.007299 0.189781
## condition on row
margin.table(dd.2xtabs, margin = 1)
## cellcode
## squamous small adeno large
## 35 48 27 27
prop.table(dd.2xtabs, margin = 1)
## censor
## cellcode survival dead
## squamous 0.11429 0.88571
## small 0.06250 0.93750
```



```
##      adeno      0.03704 0.96296
##      large      0.03704 0.96296
## condition on column
margin.table(dd.2xtabs, margin = 2)
## censor
## survival      dead
##           9      128
prop.table(dd.2xtabs, margin = 2)
##           censor
## cellcode survival  dead
## squamous  0.4444 0.2422
## small     0.3333 0.3516
## adeno     0.1111 0.2031
## large     0.1111 0.2031
```

7.3

{R} \ref{tab:RDistFun\*\*},  $X$  (random variable), .

$f = f(X = x) =$   $F(x) = \int f(x)dx =$   
 $p = F(q) = P(X \leq q) =$  , cumulative distribution function  
 $q = Q(u) = F^{-1}(p) =$  , quantile function,  $p \leq P(X \leq q)$   
 $d = f(x) = F'(x) = P(X = x) =$  , probability density function  
 $r = R(r) = f^{-1}(x) =$  , random number,

{R} , (probability function), , ProbFun, , ProbFun, , 4 ,  
p, q, d, r, . , fProbFun, , . (non-centrality parameter),  
ncp , ptukey() qtukey() Studentized Range Distribution.

Table 7.3:

|                | {R}    | (ProbFun)           |
|----------------|--------|---------------------|
| beta           | beta   | shape1, shape2, ncp |
| binomial       | binom  | size, prob          |
| Cauchy         | cauchy | location, scale     |
| chi-squared    | chisq  | df, ncp             |
| exponential    | exp    | rate                |
| F              | f      | df1, df1, ncp       |
| gamma          | gamma  | shape, scale        |
| geometric      | geom   | prob                |
| hypergeometric | hyper  | m, n, k             |
| log-normal     | lnorm  | meanlog, sdlog      |

|                   | {R}     | (ProbFun)       |
|-------------------|---------|-----------------|
| logistic          | logis   | location, scale |
| negative binomial |         | nbinom          |
| normal            | norm    | mean, sd        |
| Poisson           | pois    | lambda          |
| Student's         | t       | t df, ncp       |
| uniform           | unif    | min, max        |
| Weibull           | weibull | shape, scale    |
| Wilcoxon          | wilcox  | m, n            |

- `p` (cumulative distribution function, CDF).
- `q` (quantile),  $u \leq P(X \leq x)$   $x$ .
- `d` (probability density function, pdf).
- `r`, (pseudo-random number generation function, random number).
- `dProbFun`  $x$ .
- `pProbFun`  $q$ .
- `qProbFun`  $p$ .
- `rProbFun`  $n$ , .
- `pProbFun` `qProbFun` `lower.tail** log.p`.
  - `lower.tail = TRUE` (default),  $P(X \leq x)$ .
  - `lower.tail = FALSE`  $P(X > x)$ .
  - `log.p = TRUE`,  $p$  `log(p)` .
- `dProbFun` `log`, .

```
# normal distribution
pnorm(1.96)
## [1] 0.975
qnorm(0.975)
## [1] 1.96
dnorm(1.96)
## [1] 0.05844
# Poisson distribution
rpois(10, 1)
## [1] 0 0 0 1 1 2 1 1 4 1
rpois(10, 2)
## [1] 3 4 1 2 0 1 1 0 1 4
rpois(10, 20)
## [1] 18 21 16 23 22 24 23 20 11 22
## Cumulative distribution
## Pr(x <= 2)
ppois(2, 2)
## [1] 0.6767
ppois(4, 2)
## [1] 0.9473
```

```

ppois(6, 2)
## [1] 0.9955
# t distribution
qt(0.995, df = 2)
## [1] 9.925
2*pt(-1.96, df = 2)
## [1] 0.1891
2*pt(-1.96, df = 30)
## [1] 0.05934
# upper 1% point for an F(1, 2) distribution
sqrt(qf(0.99, 1, 2))
## [1] 9.925

```

```

, , {R} , , (current time), (seed) ,
(uniform random number), {R}, , , , ,
set.seed(), , .

```

```

## generate random number
## set.seed(): set initial value
## Caution use set.seed() everytime!
## uniform
runif(5)
## [1] 0.86121 0.43810 0.24480 0.07068 0.09947
runif(5)
## [1] 0.3163 0.5186 0.6620 0.4068 0.9129
set.seed(10)
runif(5)
## [1] 0.50748 0.30677 0.42691 0.69310 0.08514
set.seed(10)
runif(5)
## [1] 0.50748 0.30677 0.42691 0.69310 0.08514
# norm
rnorm(5)
## [1] -0.7540 -0.6059 -0.1772 0.1706 0.2428
rnorm(5)
## [1] -0.1794 -0.6305 0.9787 0.2933 -0.3703
set.seed(10)
rnorm(5)
## [1] 0.01875 -0.18425 -1.37133 -0.59917 0.29455
set.seed(10)
rnorm(5)
## [1] 0.01875 -0.18425 -1.37133 -0.59917 0.29455
## normal + uniform
set.seed(10)
runif(5)
## [1] 0.50748 0.30677 0.42691 0.69310 0.08514

```

```

rnorm(5)
## [1] -0.7540 -0.6059 -0.1772  0.1706  0.2428
set.seed(10)
runif(5)
## [1] 0.50748 0.30677 0.42691 0.69310 0.08514
rnorm(5)
## [1] -0.7540 -0.6059 -0.1772  0.1706  0.2428
set.seed(10)
rnorm(5)
## [1]  0.01875 -0.18425 -1.37133 -0.59917  0.29455
runif(5)
## [1] 0.6517 0.5677 0.1135 0.5959 0.3580

```

## 7.4 sample()

{R} `sample()`, `size`, `replace`, `prob` (seed).

```
sample(x, size, replace = FALSE, prob = NULL)
```

- `x` is a vector of length `n`.
- `size = k` is the number of elements to sample.
- `prob` is a vector of probabilities for each element in `x`.
- `replace = FALSE` means sampling without replacement.

```

## random sampling
letters
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
sample(letters, 5)
## [1] "g" "j" "b" "m" "h"
sample(letters, 5)
## [1] "n" "g" "f" "y" "v"
set.seed(1)
sample(letters, 5)
## [1] "y" "d" "g" "a" "b"
sample(letters, 5)
## [1] "w" "k" "n" "r" "s"
set.seed(1)
sample(letters, 5)
## [1] "y" "d" "g" "a" "b"
sample(letters, 5)
## [1] "w" "k" "n" "r" "s"
## sampling 5 subjects from 10 subjects
## without or with replacement
set.seed(1)

```

```

x <- 1:10
sample(x, size = 5, replace = FALSE) # (a) no resampling
## [1] 9 4 7 1 2
sample(x, size = 5, replace = TRUE) # (b) resampling
## [1] 7 2 3 1 5
# permutation
set.seed(1)
x <- 1:10
sample(x, size = 10, replace = FALSE) # no resampling
## [1] 9 4 7 1 2 5 3 10 6 8
# equal probability
set.seed(1)
x <- 1:10
sample(x, size = 5, replace = FALSE, prob = c(1:10))
## [1] 9 8 6 2 10
sample(x, size = 5, replace = FALSE, prob = c(rep(1, 10) / 10.0))
## [1] 10 1 7 6 2
# unequal probability
set.seed(1)
x <- 1:10
(prob.rs = c(seq(1, 10) / sum(seq(1, 10))))
## [1] 0.01818 0.03636 0.05455 0.07273 0.09091 0.10909 0.12727 0.14545 0.16364
## [10] 0.18182
sum(prob.rs)
## [1] 1
sample(x, size = 5, replace = TRUE, prob = seq(1, 10))
## [1] 9 8 7 3 9

```

```

, , sample(), , . , (seed).

## clinical trials or experiments
## randomization
## random assign to two groups, total 20 subjects
## random assigning treatment groups
## 20 Bernoulli trials
set.seed(1)
sample(c(0, 1), size = 20, replace = TRUE)
## [1] 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1 1 1 0
sample(2, size = 20, replace = TRUE)
## [1] 1 1 1 1 1 1 2 1 1 2 2 2 1 2 1 1 2 1 2 2
# random choose 10 subjects to group 1
set.seed(1)
sample(20, size = 10, replace = FALSE)
## [1] 4 7 1 2 13 19 11 17 14 3
# block randomization
# total 5 blocks, block size 4, choose 2 subjects to group 1

```

```
set.seed(1)
replicate(5, sample(c(1:4), size = 2, replace = FALSE))
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    3    2
## [2,]    3    2    3    2    3
```

## Chapter 8

tidyverse , .

- tidyverse
  - ggplot2 .
  - purrr .
  - tibble .
  - dplyr .
  - tidyr ,
  - stringr .
  - readr .
  - forcats (factors).
- import
  - readxl excel .
  - haven SPSS, Stata SAS .
  - jsonlite JSON .
  - xml2 XML .
  - httr web APIs .
  - rvest web scraping .
- DBI , RSQLite, RPostgres odbc.
- tidy/wrangle
  - stringr .
  - lubridate .
  - forcats (factors).
  - hms .
  - blob .
- program
  - rlang tidyverse.

```

    - magrittr      %>%
    - glue          .
  • model
    - broom        .
    - modelr       .

```

## 8.1 readr

```

tidyverse  readr      . read_csv() .csv , read_excel excel
, read_delim()      . (help(read_delim)).

```

```

• file =
• delim =
• quote = ( )
• escape_backslash = FALSE,
• escape_double = TRUE,
• col_names = (T F)
• col_types =
• na = NA
• comment = ,
• trim_ws =
• skip = (row)
• n_max =

```

```

# .csv
library(tidyverse)
library(readr)
dd <- readr::read_csv("C:/RData/DMTKAInfMo.csv")
print(dd, n = 5, width = Inf)
## # A tibble: 78 x 16
##       No    age  sex    DM  DMyr preAC prePC postAC postPC medication  SIDE PREKS
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl> <dbl> <dbl>
## 1     1     67     0     0    10   120   160   140   180         0     0    56
## 2     2     67     0     0    11   100   150   150   220         0     1    62
## 3     3     72     1     0     4   150   200   120   150         2     0    60
## 4     4     82     1     0     8   150   200   160   250         0     1    47
## 5     5     73     1     0     3    85   110   140   200         0     0    44
##   POSKS  ABS  INFECT  INFMO
##   <dbl> <dbl> <dbl> <dbl>
## 1    92     1     0     0
## 2    62     0     1     2
## 3    94     1     0     0
## 4    90     1     0     0
## 5    88     0     0     0
## # ... with 73 more rows
# .xls

```



```
library(readxl)
dd <- readxl::read_excel("C:/RData/DMTKAInfMo.xls")
print(dd, n = 5, width = Inf)
## # A tibble: 78 x 16
##       No    age  sex    DM  DMyr preAC prePC postAC postPC medication  SIDE PREKS
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1     67     0     0    10   120   160   140   180         0     0    56
## 2     2     67     0     0    11   100   150   150   220         0     1    62
## 3     3     72     1     0     4   150   200   120   150         2     0    60
## 4     4     82     1     0     8   150   200   160   250         0     1    47
## 5     5     73     1     0     3    85   110   140   200         0     0    44
##   POSKS  ABS INFECT IOFECTMO
##   <dbl> <dbl> <dbl> <dbl>
## 1    92     1     0     0
## 2    62     0     1     2
## 3    94     1     0     0
## 4    90     1     0     0
## 5    88     0     0     0
## # ... with 73 more rows
```

## 8.2 Tidy Data

tidyverse readr, tidyverse (data frame). SAS, STATA dataset .  
 tidyverse (cross table), (\*\* data table). tidyverse tidyverse  
 \*\* (tidy data) . 1 ( , row), ( , row) .  
 , , . , . .  
 •  
 • (EXCEL sheet).  
 • ( , Column) , .  
 • , (inxex) (id) .  
 . , EXCEL sheel , . , / .  
 DMTKAORI.xls. , , .

## 8.3 Tibble Data Frame

```
tidyverse readr, tidyverse (tibble), tbl_df, tbl,
tidyverse tibble, tibbles, data.frame, tidyverse
. , {R} base tibble, as.data.frame()
{R} base data.frame, tidyverse data.frame,
as_tibble() data.frame tibbles' . 5 survVATrial.csv .
## data frame object
dd <- read.table("./Data/survVATrial.csv",
```

```

header = TRUE,
sep = ",",
quote = "\"",
dec = ".",
row.names = NULL,
# col.names,
as.is = TRUE,
# as.is = !stringsAsFactors,
na.strings = c(".", "NA"))

class(dd)
## [1] "data.frame"
library(tibble)
dd <- as_tibble(dd)
class(dd)
## [1] "tbl_df"      "tbl"        "data.frame"

```

```

tibble  tibble()      {R} base data.frame(). tibble()      {R} base
. tibble  data.frame      2  .

```

1. `tibble` , `print()` 10 rows, (columns) , `str()` . `n = k` `k` rows, `width = Inf` . `options(tibble.print_min = Inf)`, rows, `options(tibble.width = Inf)` columns.
2. {R} base ( ), `tibble` .

## 8.4 Pipe

```

tidyverse  magrittr      %>%  pipe, , . , , %>% ,
, , , %>%  fun_name(), %>%  fun_name() . Unix/Linux
pipe , magrittr , %>%  %>% ,
pipe' , , .

```

```

## short and clean
log(mean(c(1:10)))
## [1] 1.705
## easily read
x <- c(1:10)
x.mean <- mean(x)
log.mean <- log(x.mean)
log.mean
## [1] 1.705
## pipe %>%
library(magrittr)
c(1:10) %>% mean() %>% log()
## [1] 1.705

```

## 8.5

```
{R} , , . 5 survVATrial.csv .
```

```
dd = dd %>% as_tibble()
print(dd, n = 5, width = Inf)
## # A tibble: 137 x 8
##   treat cellcode time censor diagtime kps age prior
##   <int>   <int> <int> <int>   <int> <int> <int> <int>
## 1     0       1    72     1      60     7    69     0
## 2     0       1   411     1      70     5    64    10
## 3     0       1   228     1      60     3    38     0
## 4     0       1   126     1      60     9    63    10
## 5     0       1   118     1      70    11    65    10
## # ... with 132 more rows
```

```
137 , 8 . time, diagtime, kps, age . treat, cellcode, censor,
prior , , . 5 survVATrial.csv .
```

```
dd$treat <- factor(dd$treat, labels = c("placebo", "test"))
dd$cellcode <- factor(dd$cellcode,
                      labels = c("squamous", "small", "adeno", "large"))
dd$censor <- factor(dd$censor, labels = c("survival", "dead"))
dd$prior <- factor(dd$prior, labels = c("no", "yes"))
print(dd, n = 5, width = Inf)
## # A tibble: 137 x 8
##   treat cellcode time censor diagtime kps age prior
##   <fct> <fct>   <int> <fct>   <int> <int> <int> <fct>
## 1 placebo squamous    72 dead      60     7    69 no
## 2 placebo squamous   411 dead      70     5    64 yes
## 3 placebo squamous   228 dead      60     3    38 no
## 4 placebo squamous   126 dead      60     9    63 yes
## 5 placebo squamous   118 dead      70    11    65 yes
## # ... with 132 more rows
```

## 8.6 dplyr

tidyverse dplyr , .

- %>% = pipe
- rename() = (column)
- filter() = (rows)
- arrange() =
- select() = (variables) (columns)
- mutate() =
- summarise() =

- `group_by()` =
- `%>%` = pipe

```
1      ,      ( ),      group_by() %>% .
```

### 8.6.1 filter()

, . `filter()` . 5 `survVATrial.csv` , `treat`  
`placebo`, `cellcode` `large`.

```
dd %>%
  filter(treat == 'placebo', cellcode == 'large')
## # A tibble: 15 x 8
##   treat cellcode time censor diagtime kps age prior
##   <fct> <fct>    <int> <fct>    <int> <int> <int> <fct>
## 1 placebo large    177 dead      50    16    66 yes
## 2 placebo large    162 dead      80     5    62 no
## 3 placebo large    216 dead      50    15    52 no
## 4 placebo large    553 dead      70     2    47 no
## 5 placebo large    278 dead      60    12    63 no
## 6 placebo large     12 dead      40    12    68 yes
## 7 placebo large    260 dead      80     5    45 no
## 8 placebo large    200 dead      80    12    41 yes
## 9 placebo large    156 dead      70     2    66 no
## 10 placebo large    182 survival    90     2    62 no
## 11 placebo large    143 dead      90     8    60 no
## 12 placebo large    105 dead      80    11    66 no
## 13 placebo large    103 dead      80     5    38 no
## 14 placebo large    250 dead      70     8    53 yes
## 15 placebo large    100 dead      60    13    37 yes
```

`treat` , `age` 50, `kps` 7

```
dd %>%
  filter(treat == 'test', age > 50, kps <= 7)
## # A tibble: 35 x 8
##   treat cellcode time censor diagtime kps age prior
##   <fct> <fct>    <int> <fct>    <int> <int> <int> <fct>
## 1 test squamous  112 dead      80     6    60 no
## 2 test squamous  242 dead      50     1    70 no
## 3 test squamous  111 dead      70     3    62 no
## 4 test squamous  587 dead      60     3    58 no
## 5 test squamous  389 dead      90     2    62 no
## 6 test squamous   33 dead      30     6    64 no
## 7 test squamous  467 dead      90     2    64 no
## 8 test squamous  283 dead      90     2    51 no
## 9 test small     25 dead      30     2    69 no
```

```
## 10 test small 21 dead 20 4 71 no
## # ... with 25 more rows
```

### 8.6.2 arrange()

```
, arrange()
survVATrial.csv , age , time , , desc() , 5
sort() rank() .
```

```
dd %>%
  arrange(age, desc(time))
## # A tibble: 137 x 8
##   treat   cellcode time censor diagtime kps age prior
##   <fct>   <fct>   <int> <fct>   <int> <int> <int> <fct>
## 1 placebo adeno     95 dead      80 4 34 no
## 2 placebo small      4 dead      40 2 35 no
## 3 test squamous      1 dead      50 7 35 no
## 4 test small    103 survival    70 22 36 yes
## 5 placebo large    100 dead      60 13 37 yes
## 6 test large      49 dead      30 3 37 no
## 7 placebo squamous 228 dead      60 3 38 no
## 8 placebo adeno    117 dead      80 2 38 no
## 9 placebo large    103 dead      80 5 38 no
## 10 test adeno      31 dead      80 3 39 no
## # ... with 127 more rows
```

### 8.6.3 select()

```
, , , RAM . 5 survVATrial.csv
, treat, cellcode, censor .
```

```
dd %>%
  arrange(treat, cellcode, censor)
## # A tibble: 137 x 8
##   treat   cellcode time censor diagtime kps age prior
##   <fct>   <fct>   <int> <fct>   <int> <int> <int> <fct>
## 1 placebo squamous 100 survival    70 6 70 no
## 2 placebo squamous 25 survival    80 9 52 yes
## 3 placebo squamous 72 dead      60 7 69 no
## 4 placebo squamous 411 dead      70 5 64 yes
## 5 placebo squamous 228 dead      60 3 38 no
## 6 placebo squamous 126 dead      60 9 63 yes
## 7 placebo squamous 118 dead      70 11 65 yes
## 8 placebo squamous 10 dead      20 5 49 no
## 9 placebo squamous 82 dead      40 10 69 yes
## 10 placebo squamous 110 dead      80 29 68 no
```

```
## # ... with 127 more rows
```

### 8.6.4 mutate()

```
5 survVATrial.csv
, time , diagtime * age / 100.
```

```
dd %>%
  mutate(
    log_age = log(age),
    diag_age = diagtime * age / 100
  )
## # A tibble: 137 x 10
##   treat   cellcode time censor  diagtime  kps  age prior log_age diag_age
##   <fct>   <fct>   <int> <fct>    <int> <int> <int> <fct>   <dbl>   <dbl>
## 1 placebo squamous    72 dead      60    7   69 no     4.23    41.4
## 2 placebo squamous   411 dead      70    5   64 yes    4.16    44.8
## 3 placebo squamous   228 dead      60    3   38 no     3.64    22.8
## 4 placebo squamous   126 dead      60    9   63 yes    4.14    37.8
## 5 placebo squamous   118 dead      70   11   65 yes    4.17    45.5
## 6 placebo squamous    10 dead      20    5   49 no     3.89     9.8
## 7 placebo squamous    82 dead      40   10   69 yes    4.23    27.6
## 8 placebo squamous   110 dead      80   29   68 no     4.22    54.4
## 9 placebo squamous   314 dead      50   18   43 no     3.76    21.5
## 10 placebo squamous   100 survival    70    6   70 no     4.25     49
## # ... with 127 more rows
```

### 8.6.5 summarise()

```
5 survVATrial.csv , n(), age .
```

- Center: mean(), median()
- Spread: sd(), IQR(), mad(). range()
- Range: min(), max(), quantile()
- Position: first(), last(), nth()
- Count: n(), n\_distinct()
- Logical: any(), all()

```
5 survVATrial.csv , n(), age .
```

```
dd %>%
  summarize(
    count = n(),
    age_mean = mean(age, na.rm = TRUE),
    age_sd = sd(age, na.rm = TRUE)
  )
## # A tibble: 1 x 3
```

```
##   count age_mean age_sd
##   <int>   <dbl> <dbl>
## 1   137    58.3   10.5
```

### 8.6.6 group\_by()

```
##   ,   . group_by()   ,   .   5 survVATrial.csv
##   ,   diagtime     ..
```

```
dd %>%
  group_by(treat) %>%
  summarise(
    diagtime_mean = mean(diagtime),
    diagtime_sd = sd(diagtime)
  )
## # A tibble: 2 x 3
##   treat   diagtime_mean diagtime_sd
##   <fct>         <dbl>     <dbl>
## 1 placebo      59.2       18.7
## 2 test         57.9       21.4
```

### 8.6.7

```
summarise()   ,   ,   .
```

- summarise\_all()
- summarise\_each() ,
- summarise\_at()
- summarise\_if()

```
dd %>%
  select(time, diagtime, kps, age) %>%
  summarise_all(mean, na.rm = TRUE)
## # A tibble: 1 x 4
##   time diagtime kps age
##   <dbl>   <dbl> <dbl> <dbl>
## 1  122.    58.6  8.77  58.3
dd %>% select(time, diagtime, kps, age) %>%
  summarise_all(list(mean, sd), na.rm = TRUE)
## # A tibble: 1 x 8
##   time_fn1 diagtime_fn1 kps_fn1 age_fn1 time_fn2 diagtime_fn2 kps_fn2 age_fn2
##   <dbl>         <dbl>   <dbl>   <dbl>   <dbl>         <dbl>   <dbl>   <dbl>
## 1   122.         58.6    8.77    58.3    158.         20.0    10.6    10.5
dd %>%
  summarise_each(list(mean, sd), time, age) # not so useful
## # A tibble: 1 x 4
```

```
##   time_fn1 age_fn1 time_fn2 age_fn2
##   <dbl>   <dbl>   <dbl>   <dbl>
## 1    122.    58.3    158.    10.5
dd %>%
  summarise_at(c("time", "age"), mean, na.rm = TRUE)
## # A tibble: 1 x 2
##   time age
##   <dbl> <dbl>
## 1  122.  58.3
dd %>%
  summarise_at(vars(time, age), mean, na.rm = TRUE)
## # A tibble: 1 x 2
##   time age
##   <dbl> <dbl>
## 1  122.  58.3
dd %>%
  summarise_if(is.numeric, list(mean, sd), na.rm = TRUE)
## # A tibble: 1 x 8
##   time_fn1 diagtime_fn1 kps_fn1 age_fn1 time_fn2 diagtime_fn2 kps_fn2 age_fn2
##   <dbl>         <dbl>   <dbl>   <dbl>   <dbl>         <dbl>   <dbl>   <dbl>
## 1    122.         58.6    8.77    58.3    158.         20.0    10.6    10.5
```

## 8.7



# Bibliography

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2020). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.20.