

# A Minimal Book Example

Yihui Xie

2020-09-16



# Contents



# Chapter 1

## Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation  $a^2 + b^2 = c^2$ .

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")  
# or the development version  
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading #.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.



# Chapter 2

## 1. R

### 2.1

#### 2.1.1 R Rstudio

R

- R
  - Windows, [Download R \(64-bit\)](#) for Windows.
  - Mac, [Download R \(64-bit\)](#) for Mac OS X.
  - Linux, [Download R \(64-bit\)](#) for Linux.

```
sudo apt-get install r-cran-base
```

R:

```
sessionInfo()$R.version$version.string
```

```
## [1] "R version 3.6.0 (2019-04-26)"
```

R RStudio:

- RStudio
  - [Download RStudio](#) (64-bit) for Windows.
  - [Download RStudio](#) (64-bit) for Mac OS X.
  - [Download RStudio](#) (64-bit) for Linux.
- RStudio cloud
  - [RStudio Cloud](#) (IDE), [RStudio Cloud](#) (IDE), [RStudio Cloud](#) (IDE).

RStudio — R, , , ,  
 - Jupyter Notebook  
 , RStudio — R, , ,  
 Jupyter Notebook, (RNotebook —  
 ), — RMarkdown. !

## 2.1.2 RStudio

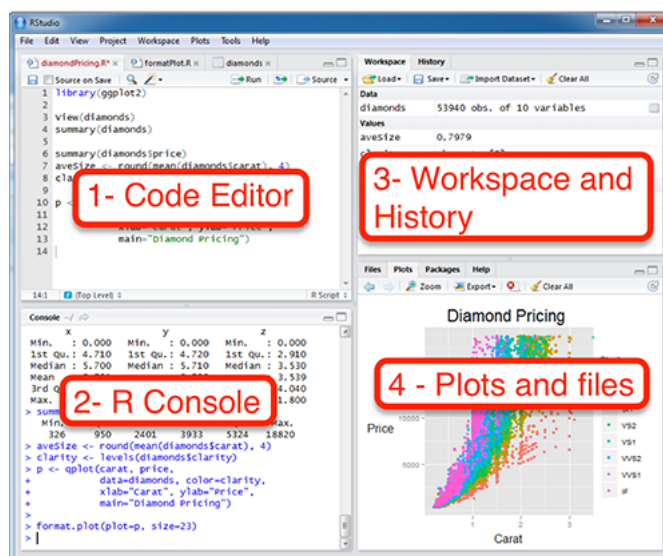


Figure 2.1:

: 1 - Code Editor ( )<sup>1</sup> 2 - R Console ( ).

### 2 - R Console

Enter.

1 - Code Editor  
 Enter macOS).

Ctrl + Enter (Cmd +

Ctrl

<sup>1</sup> RStudio  
 Script.

File - New File - R



+ A Windows Linux, Cmd + A macOS <sup>2</sup>. (

), **2 - R Console**, .

File - Save

As... R .R, ,

- “ ”.

### 3 - Workspace and History —

### 4 - Plots and files.

(Packages) Help

## 2.1.3 R

R — , R . R

, , R

R

\*, /, ^ ( ), () ... +, -, :

```
40+2
```

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

```
3-2
```

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

```
5*6
```

```
## [1] 30
```

```
99/9 #
```

```
## [1] 11
```

```
2^3 #
```

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

```
13 %/% 3 #
```

```
## [1] 4
```

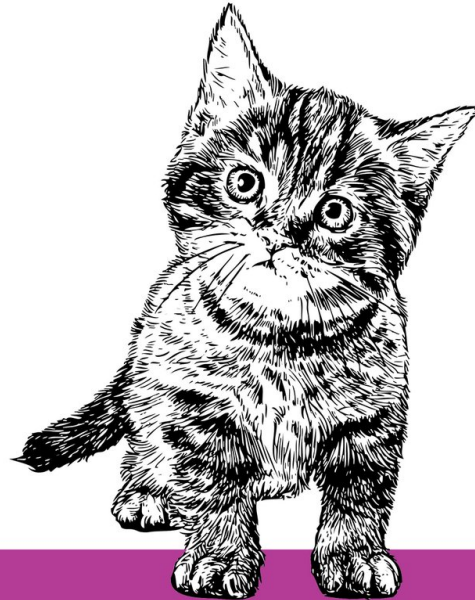
---

<sup>2</sup> RStudio . , Help - Keyboard Shortcuts Help.

```
13 %% 3 #
```

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

*How to actually learn any new programming concept*



*Essential*

## Changing Stuff and Seeing What Happens

O RLY?

@ThePracticalDev

Figure 2.2:

, ?  
R , (#). , #

( - ),<sup>3</sup>  
 , : “ ” , - # ,  
 : Ctrl + Shift + C (Cmd + Shift +  
 C macOS) — #  
 , 14% :

```
2 + 2 * 2
```

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

, ) - , 6 ( ,  
 ( .. , *operator precedence*) R  
 , .

```
(2+2)*2
```

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

, , , ?Syntax.

### 2.1.4

- . , , ,  
 :

```
16^0.5
```

```
## [1] 4
```

, : -  
 . ( ) - , -  
 , - ( , ..).  
 , , :

```
sqrt(16)
```

```
## [1] 4
```

R — case-sensitive , .. . SQRT(16) .  
 :

```
log(8)
```

```
## [1] 2.079442
```

<sup>3</sup>  
 ! , ,

```

, ... - , -
.
!
— , , ,
.
8 2 3:
log2 8 = 3
2 3 8:
23 = 8
- .
, , :
?log

```

RStudio :

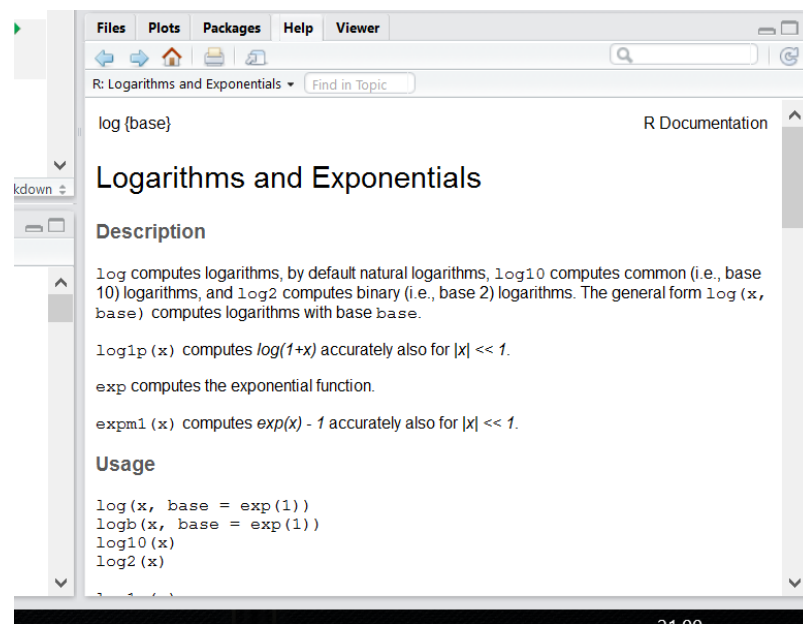


Figure 2.3:

```

, base = (2.7182818...),
. R -
, ,
.

```

```
log(x = 8, base = 2)
```

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

```
... ( ):
```

```
log(8,2)
```

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

```
, :
```

```
log(8, sqrt(4))
```

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

( Python). , — R  
- R , .  
: +, -, /, ^ ..  
:

```
'+'(3, 4)
```

```
## [1] 7
```

### 2.1.5

: <- ( =, R  
Alt + - ( option + - macOS).

```
a <- 2
```

```
a
```

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

```
, ! ,  
,
```

**Environment** RStudio:

```
:
```

```
b <- a ^ a + a * a
```

```
b
```

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

```
log(b, a)
```

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

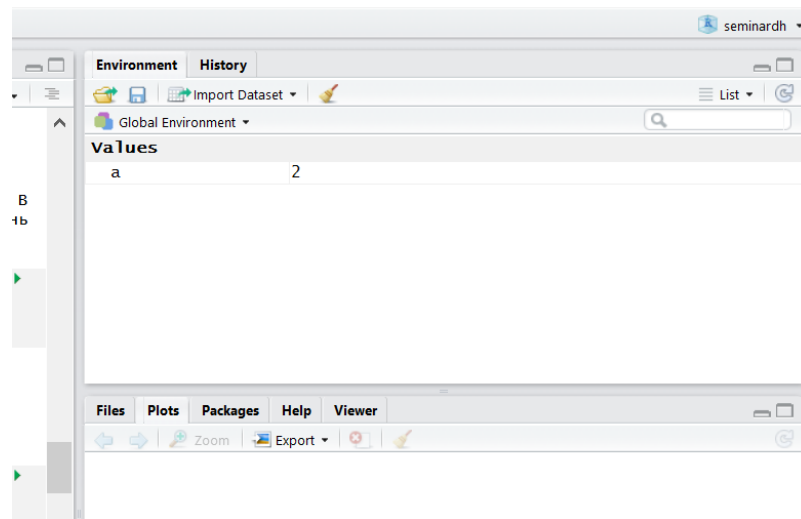


Figure 2.4:

```

:

a == b

## [1] FALSE

, ==, =.

.

a = b
a

## [1] 8

(

)

, ( =().

:

a <- 2
b <- 3

a==b

## [1] FALSE

a!=b

## [1] TRUE

```

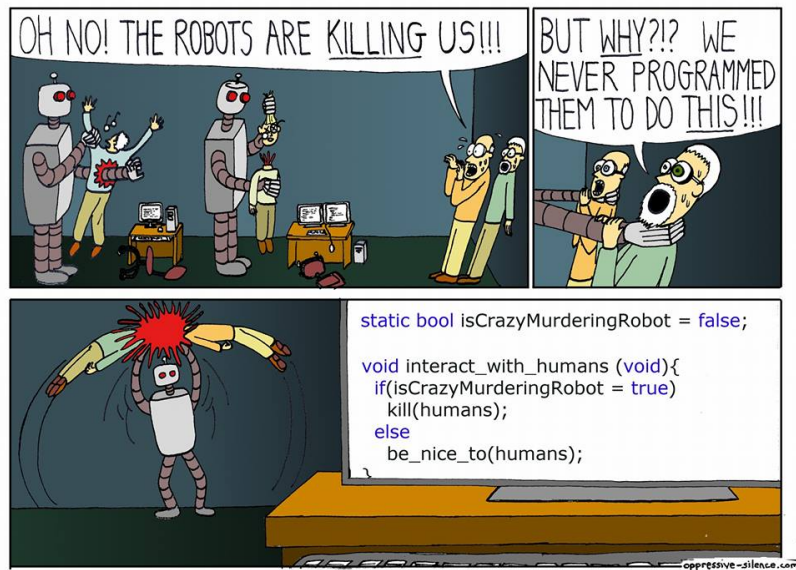


Figure 2.5:

R

/ :

a &gt; b

## [1] FALSE

a &lt; b

## [1] TRUE

a &gt;= b

## [1] FALSE

a &lt;= b

## [1] TRUE

## 2.2

(numeric):

class(a)

## [1] "numeric"

```

    , R      numeric: integer ( ), double ( ), com-
plex ( ).      : complexnumber <- 2+2i      R
    , R
    :
    numeric integer,      R
    R:

1. character:      .      .      ", ' (
    ,      -      ).
s <- "      !"
s
## [1] "      !"
class(s)
## [1] "character"

2. logical:      TRUE      FALSE.
t1 <- TRUE
f1 <- FALSE
t1
## [1] TRUE
f1
## [1] FALSE
    ,      T F (      True False!)
t2 <- T
f2 <- F
    ,      R      TRUE FALSE,      T F
TRUE <- FALSE
## Error in TRUE <- FALSE:      (do_set)
TRUE
## [1] TRUE
T <- FALSE
T
## [1] FALSE
    ,      ,
    !

```



```
comparison <- a == b
comparison
```

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

```
      ,      : -
      ,      .
      — (!): ,
```

```
t1
```

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

```
!t1
```

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

```
!!t1 # !
```

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

```
      ( TRUE TRUE):
```

```
t1 & t2
```

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

```
t1 & f1
```

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

```
      ( TRUE TRUE):
```

```
t1 | f1
```

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

```
f1 | f2
```

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

```
      ,      ( ! ) .
      !      ( ! ) .
```

## 2.3

```
vector atomic) — ( ), (atomic
      {0,0} - , {2,3}:
```

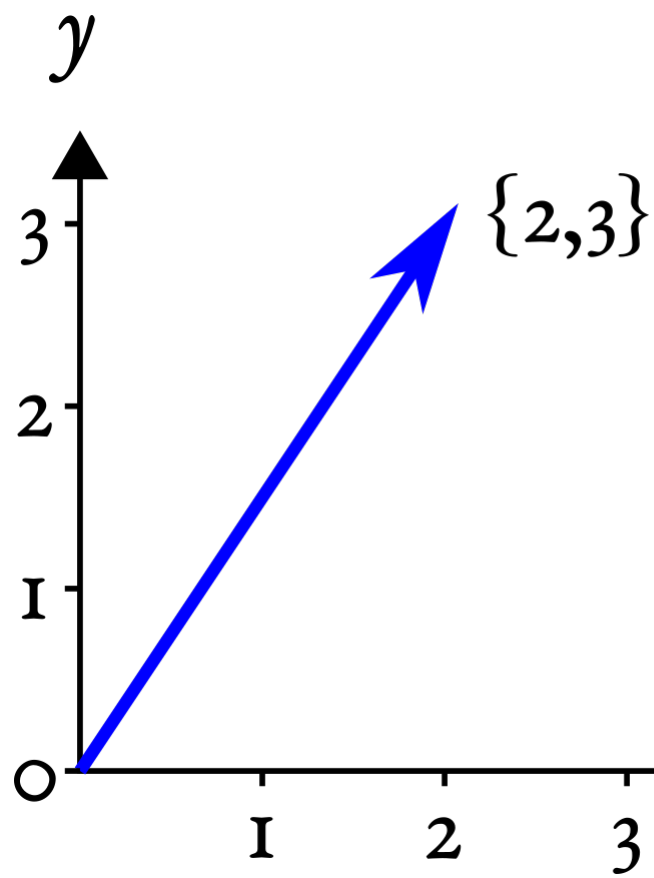


Figure 2.6:

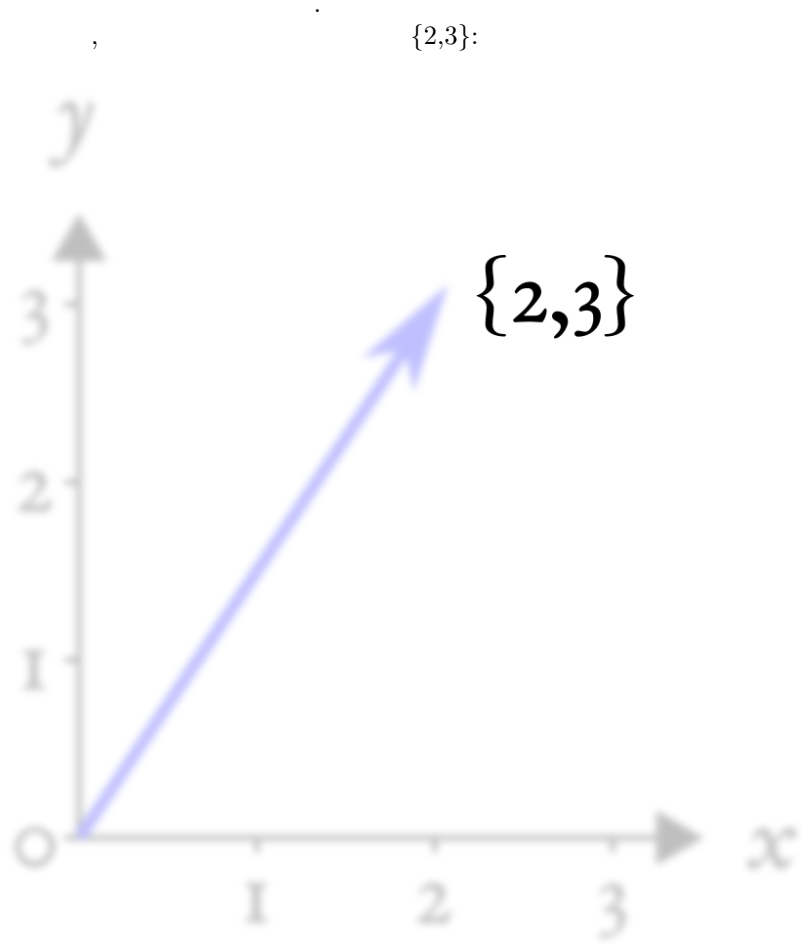


Figure 2.7:

```
,  
(... ) R, , , 1. ! , R  
, c():  
c(4,8,15,16,23,42)  
## [1] 4 8 15 16 23 42  
numeric, character logical:  
c(" ", " ", " ", " ")  
## [1] " " " " " " " "
```

```
c(TRUE, FALSE)
```

```
## [1] TRUE FALSE
```

```
c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
#> [1] 1 2 3 4 5 6 7 8 9 10
```

```
1:10
```

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

```
5:-3
```

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

```
seq(1, 100, by = 10)
#> [1] 1 11 21 31 41 51 61 71 81 91 100
```

```
## [1] 10 20 30 40 50 60 70 80 90 100
```

```
seq(1, 13, length.out = 4)
#> [1] 1 5 9 13
```

```
## [1] 1 5 9 13
```

```
rep(1, 5)
#> [1] 1 1 1 1 1
```

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

```
rep(1:3, 3)
#> [1] 1 2 3 1 2 3 1 2 3
```

```
## [1] 1 2 3 1 2 3 1 2 3
```

```
rep(1:3, 1:3)
#> [1] 1 2 3 2 3 3 1 2 3 3 3
```

```
## [1] 1 2 2 3 3 3
```

```
## [1] 1 2 2 3 3 3
```

```
(c("Hey", "Ho"), c("Let's", "Go!"))
```

```
v1 <- c("Hey", "Ho")
```

```
v2 <- c("Let's", "Go!")
```

```
c(v1, v2)
```

```
## [1] "Hey" "Ho" "Let's" "Go!"
```

```

R . , sum() ( )
mean() ( ):
sum(1:10)

## [1] 55
mean(1:10)

## [1] 5.5

```

### 2.3.1

```

, ? ?
, atomic .
R “ ” .
”, “ ” .
:
c(FALSE, 2)

## [1] 0 2
FALSE 0 ( TRUE 1),
:
2 + TRUE

## [1] 3

```

(implicit coercion).

```

:
c(TRUE, 3, " ")

## [1] "TRUE" "3" " "

```

R :

```

NULL < raw < logical < integer < double < complex < character <
list < expression.

```

```

0 1 , 0 1 "0" "1". — TRUE FALSE —

```

as.

```

(explicit coercion):
as.numeric(c(T, F, F))

## [1] 1 0 0

```

```
as.character(as.numeric(c(T, F, F)))

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

, , NA — ( ).

as.numeric(c("1", "2", " "))

## Warning: NA

## [1] 1 2 NA

sum() mean() TRUE .
```

### 2.3.2

```
n <- 1:4
m <- 4:1
n + m

## [1] 5 5 5 5

n - m

## [1] -3 -1 1 3

n * m

## [1] 4 6 6 4

n / m

## [1] 0.2500000 0.6666667 1.5000000 4.0000000

n ^ m + m * (n - m)

## [1] -11 5 11 7
```

(vectorization).  
 MATLAB  
 m\*n (dot product),  
 % :  
 n %% m  
 ## [,1]  
 ## [1,] 20

R,

, R, , (

, ). “ ”

:

```
sqrt(1:10)
```

```
## [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
```

```
## [9] 3.000000 3.162278
```

R

(C, C++, FORTRAN),

,

R — ,

for while ??.

! —

for while ??.

### 2.3.3 Recycling

— , !

( , *recycling rule*).

,

:

```
n <- 1:4
```

```
m <- 1:2
```

```
n * m
```

```
## [1] 1 4 3 8
```

, ? :

1 , :

```
n * 2
```

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

( , 3, 4), R

.

```
n + c(3,4,5)
```

```
## Warning in n + c(3, 4, 5):
```

```
##
```

```
## [1] 4 6 8 7
```

### 2.3.4

```
n <- 1:10
```

```
n[1]
```

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

```
n[10]
```

```
## [1] 10
```

```
n[3] <- 20
```

```
n
```

```
## [1] 1 2 20 4 5 6 7 8 9 10
```

```
n[4:7]
```

```
## [1] 4 5 6 7
```

```
n[10:1]
```

```
## [1] 10 9 8 7 6 5 4 20 2 1
```

```
n[-1]
```

```
## [1] 2 20 4 5 6 7 8 9 10
```



```

n[c(-4, -5)]

## [1] 1 2 20 6 7 8 9 10

, . :

n[c(T,F,T,F,T,F,T,F,T,F)]

## [1] 1 20 5 7 9

, !

n[c(T,F)] # - recycling rule!

## [1] 1 20 5 7 9

, : . ,

( ) :

my_named_vector <- c(first = 1, second = 2, third = 3)
my_named_vector['first']

## first
## 1

" " names()

d <- 1:4
names(d) <- letters[1:4]
d["a"]

## a
## 1

letters — " " R — a z. !
, LETTERS — , .
pi.

n:

mean(n)

## [1] 7.2

, ?

— :

larger <- n>mean(n)
larger

## [1] FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE

n:

```

```
n[larger]
```

```
## [1] 20 8 9 10
```

```
:
```

```
n[n>mean(n)]
```

```
## [1] 20 8 9 10
```

```
,
```

```
R: (subset)
```

```
.
```

### 2.3.5 NA -

NA — "NA", 0, NA ( *Not Available* - FALSE. NA — NA. NA: NA

```
missed <- NA
```

```
missed == "NA"
```

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

```
missed == ""
```

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

```
missed == NA
```

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

```
: NA c NA NA!
```

```
NA :
```

```
n[5] <- NA
```

```
n
```

```
## [1] 1 2 20 4 NA 6 7 8 9 10
```

```
mean(n)
```

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

```
?
```

```
,
```

```
NA
```

```
.
```

```
:
```

```
n == NA
```

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

```

, , NA c NA NA...
, is.na():
is.na(n)
## [1] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
is.na(n) FALSE , TRUE , NA.
! ( ),
:
n[!is.na(n)]
## [1] 1 2 20 4 6 7 8 9 10
, !
mean(n[!is.na(n)])
## [1] 7.444444
, (!)
, NA. mean():
?mean()
na.rm =, FALSE. , !
mean(n, na.rm = T)
## [1] 7.444444
!
NA
NA: NA, NA_integer_, NA_real_, NA_complex_ and
NA_character_.
NA NaN — . NaN Not a Number
0 / 0. , is.na() TRUE
NaN, is.nan() TRUE NaN FALSE NA:
is.na(NA)
## [1] TRUE
is.na(NaN)
## [1] TRUE
is.nan(NA)
## [1] FALSE

```

```
is.nan(NaN)
```

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

### 2.3.6



Figure 2.8:

*Stackoverflow*<sup>4</sup> R- !

Computer Programming To Be Officially Renamed “Googling Stack Overflow” Source: <http://t.co/xu7acfXvFF> [pic.twitter.com/iJ9k7aAVhd](http://pic.twitter.com/iJ9k7aAVhd)

— Stack Exchange (?) July 20, 2015

Does anyone ever get good at R or do they just get good at googling how to do things in R

— Lauren M. Seyler, Ph.D. (?) May 6, 2019

<sup>4</sup>Stackoverflow — Quora, The Question, Mail.ru

**Doctors: Googling stuff online does not  
make you a doctor.**

**Programmers:**



Figure 2.9:

R. , , —  
 : (list) (matrix).  
 “ ” , R —  
 data.frame.

## 2.4 (matrix)

, — “ ” : ,  
 , . matrix() ,

```
A <- matrix(1:20, nrow=5, ncol=4)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```
A <- matrix(1:20, nrow=5)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```
A[2,3]
```

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

```
A[2:4, 1:3]
```

```
##      [,1] [,2] [,3]
## [1,]    2    7   12
## [2,]    3    8   13
## [3,]    4    9   14
```

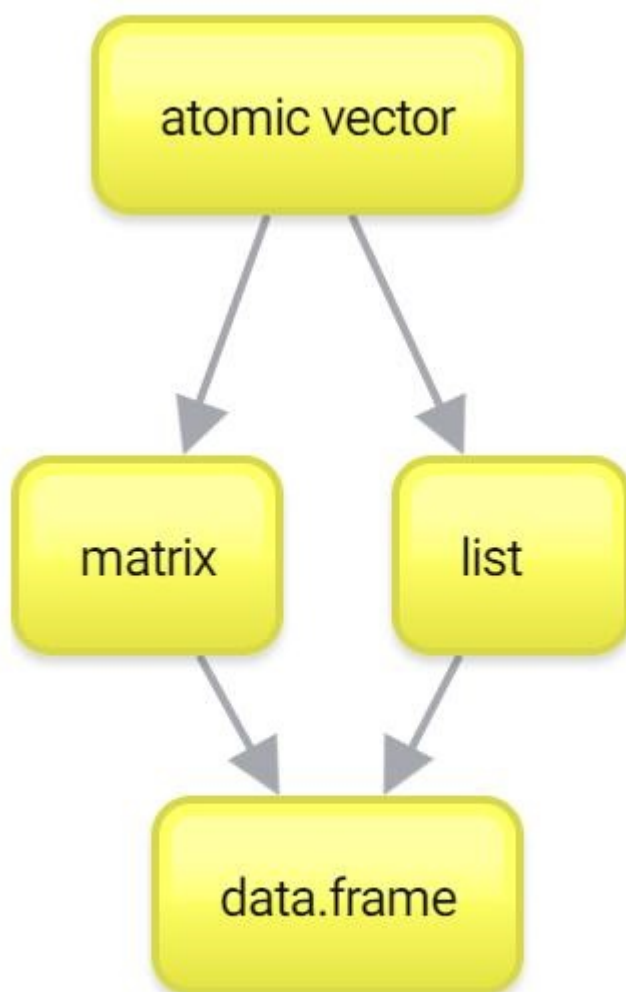


Figure 2.10:

```
A[, 1:3]
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   11
## [2,]    2    7   12
## [3,]    3    8   13
## [4,]    4    9   14
## [5,]    5   10   15
```

```
A[2:4, ]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    2    7   12   17
## [2,]    3    8   13   18
## [3,]    4    9   14   19
```

```
A[, ]
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2    7   12   17
## [3,]    3    8   13   18
## [4,]    4    9   14   19
## [5,]    5   10   15   20
```

```
A[2:4, 2:4] <- 100
```

```
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    6   11   16
## [2,]    2  100  100  100
## [3,]    3  100  100  100
## [4,]    4  100  100  100
## [5,]    5   10   15   20
```

, , R , ,  
, MATLAB. :

, — , R —  
dim ( ) dimnames.

dim  
. 99-101 “R in a Nutshell” (?).



## 2.5 (list)

```
l <- list(42, " ", T)
l
```

```
## [[1]]
## [1] 42
##
## [[2]]
## [1] " "
##
## [[3]]
## [1] TRUE
```

```
lbig <- list(c("Wow", "this", "list", "is", "so", "big"), "16", l)
lbig
```

```
## [[1]]
## [1] "Wow" "this" "list" "is" "so" "big"
##
## [[2]]
## [1] "16"
##
## [[3]]
## [[3]][[1]]
## [1] 42
##
## [[3]][[2]]
## [1] " "
##
## [[3]][[3]]
## [1] TRUE
```

```
str(lbig)
```

```
## List of 3
## $ : chr [1:6] "Wow" "this" "list" "is" ...
## $ : chr "16"
## $ :List of 3
## ..$ : num 42
## ..$ : chr " "
## ..$ : logi TRUE
```

```
named1 <- list(age = 24, PhDstudent = T, language = "Russian")
named1
```

```
## $age
## [1] 24
##
## $PhDstudent
## [1] TRUE
##
## $language
## [1] "Russian"
```

```
named1$age
```

```
## [1] 24
```

```
named1[1]
```

```
## $age
## [1] 24
```

```
class(named1)
```

```
## [1] "list"
```

```
class(named1[1])
```

```
## [1] "list"
```

```
named1[[1]]
```

```
## [1] 24
```

```
class(named1[[1]])
```

```
## [1] "numeric"
```

Indexing lists in #rstats. Inspired by the Residence Inn [pic.twitter.com/YQ6axb2w7t](https://pic.twitter.com/YQ6axb2w7t)

— Hadley Wickham (?) September 14, 2015

```
named1[['age']]
```

```
## [1] 24
```

## 2.6 Data.frame

, . - . (data.frames). , , .

```
name <- c("Ivan", "Eugeny", "Lena", "Misha", "Sasha")
age <- c(26, 34, 23, 27, 26)
student <- c(F, F, T, T, T)
df <- data.frame(name, age, student)
df
```

```
##      name age student
## 1   Ivan  26    FALSE
## 2 Eugeny  34    FALSE
## 3   Lena  23     TRUE
## 4  Misha  27     TRUE
## 5  Sasha  26     TRUE
```

```
str(df)
```

```
## 'data.frame': 5 obs. of 3 variables:
## $ name : Factor w/ 5 levels "Eugeny","Ivan",...: 2 1 3 4 5
## $ age : num 26 34 23 27 26
## $ student: logi FALSE FALSE TRUE TRUE TRUE
```

, , ? , — -  
*atomic* , “ ” 90 , “ ”  
 , “ ” 90 , ,  
 . ( !),  
 , . — numeric, — logical.  
 — character, — numeric, — logical.  
 , :

```
df$age[2:3]
```

```
## [1] 34 23
```

```

      age $.
      2 3.
$
:
```

```
df$lovesR <- T # recycling - ?
df
```

```
##      name age student lovesR
## 1   Ivan  26    FALSE    TRUE
## 2 Eugeny  34    FALSE    TRUE
## 3   Lena  23     TRUE    TRUE
## 4  Misha  27     TRUE    TRUE
## 5  Sasha  26     TRUE    TRUE
```

```
df[3:5, 2:3]
```

```
##      age student
## 3   23     TRUE
## 4   27     TRUE
## 5   26     TRUE
```

```
df[1:2, "age"]
```

```
## [1] 26 34
```

```

! , R ,
df[df$age < mean(df$age), 4]
```

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

```

:
df$lovesR[df$age < mean(df$age)]
```

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

```
df[df$age < mean(df$age), 'lovesR']
```

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

RStudio. View(df)  
( Environment).  
Excel  
5  
...

	name	age	student	lovesR
1	Ivan	26	FALSE	TRUE
2	Eugeny	34	FALSE	TRUE
3	Lena	23	TRUE	TRUE
4	Misha	27	TRUE	TRUE
5	Sasha	26	TRUE	TRUE

Figure 2.11:

, , , . . .

2.7

, ) , , “ ” . . , ,  
...

2.7.1

, read.csv() — “ ”. ( ,  
:  
read.csv("character-deaths.csv")  
## Warning in file(file, "rt"):'character-deaths.csv': No  
## such file or directory  
## Error in file(file, "rt"):  
R  
R  
• :  
5 , , .

```

    , getwd() ( ),
    :
got <- read.csv("character-deaths.csv")
• :
    setwd() , ,
    :
got <- read.csv("character-deaths.csv")
• :
got <- read.csv("/Users/Username/Some_Folder/character-deaths.csv")
    , ,
    Windows : / R,
    //.
• : Import Dataset.
    Environment RStudio Import Dataset. ,
    ,
•
    R, ,
    :
got <- read.csv("https://raw.githubusercontent.com/Pozdniakov/stats/master/data/charac")
• : RStudio.
File - New Project..., New Directory, New Project,
Directory Name Create Project.
- , , ,
    — , ,
    — , , data
    .
    , Git.
    RStudio
File - New Project... Version Control, Git
Subversion RStudio
    Git RStudio.

```

## 2.7.2

```

R.
read.table().
read.csv(), read.csv2()
read.table(),
.csv, Comma Separated
Values ( , ). , " " : ( ..
)
" " .csv, (;),
(,). read.csv() read.csv2() —
" " , — ( ) " ".
read.csv()
read.csv2()
file =,
stringsAsFactors = FALSE:
got <- read.csv("data/character-deaths.csv", stringsAsFactors = FALSE)

read.table() character
(factor). , — , character,
: ,
"male" "female", 1 2, ,
1 2.
character, factor,
.
View(got): ! -
Help.
.csv , .tsv — , .csv,
read.delim() read.delim2().
read.table().

```

```

  ( fread() data.table — !),
  “ ” — .
  .
  , — Microsoft Excel. .xlsx ,
  .csv.
  : readxl, xlsx, openxlsx.
  SPSS, Stata, SAS foreign.
  .

```



## Chapter 3

### 2.

### R

#### 3.1

#### R

```
got <- read.csv("data/character-deaths.csv", stringsAsFactors = FALSE)
```

##### 3.1.1

```
str(got)
```

```
## 'data.frame':  917 obs. of  13 variables:
## $ Name          : chr  "Addam Marbrand" "Aegon Frey (Jinglebell)" "Aegon Targaryen" "Adra
## $ Allegiances    : chr  "Lannister" "None" "House Targaryen" "House Greyjoy" ...
## $ Death.Year     : int   NA 299 NA 300 NA NA 300 300 NA NA ...
## $ Book.of.Death   : int   NA 3 NA 5 NA NA 4 5 NA NA ...
## $ Death.Chapter  : int   NA 51 NA 20 NA NA 35 NA NA NA ...
## $ Book.Intro.Chapter: int  56 49 5 20 NA NA 21 59 11 0 ...
## $ Gender         : int   1 1 1 1 1 1 1 0 1 1 ...
## $ Nobility       : int   1 1 1 1 1 1 1 1 1 0 ...
## $ GoT            : int   1 0 0 0 0 0 1 1 0 0 ...
## $ CoK            : int   1 0 0 0 0 1 0 1 1 0 ...
```

```
## $ SoS      : int  1 1 0 0 1 1 1 1 0 1 ...
## $ FfC      : int  1 0 0 0 0 0 1 0 1 0 ...
## $ DwD      : int  0 0 1 1 0 0 0 1 0 0 ...

:

Name - , - : - , -
, - . Allegiances - .
, , Stark
House Stark . - Death Year, Book.of.Death, Death.Chapter,
Book.Intro.Chapter - ,
( ), . Gender - 1 , 0
. Nobility - . 5 ,
( 5).

- head(): ( 6)
```

```
head(got)
```

```
##           Name      Allegiances Death.Year Book.of.Death
## 1      Addam Marbrand      Lannister      NA      NA
## 2 Aegon Frey (Jinglebell)      None      299      3
## 3      Aegon Targaryen House Targaryen      NA      NA
## 4      Adrack Humble   House Greyjoy      300      5
## 5      Aemon Costayne      Lannister      NA      NA
## 6      Aemon Estermont   Baratheon      NA      NA
## Death.Chapter Book.Intro.Chapter Gender Nobility GoT CoK SoS FfC DwD
## 1      NA      56      1      1 1 1 1 1 0
## 2      51      49      1      1 0 0 1 0 0
## 3      NA      5      1      1 0 0 0 0 1
## 4      20      20      1      1 0 0 0 0 1
## 5      NA      NA      1      1 0 0 1 0 0
## 6      NA      NA      1      1 0 1 1 0 0
```

```
tail(). , .
```

```
table():
```

```
table(got$Allegiances)
```

```
##
##      Arryn      Baratheon      Greyjoy      House Arryn House Baratheon
##      23      56      51      7      8
## House Greyjoy House Lannister House Martell House Stark House Targaryen
##      24      21      12      35      19
## House Tully House Tyrell Lannister Martell Night's Watch
##      8      11      81      25      116
##      None      Stark      Targaryen      Tully      Tyrell
```

3.1. *R* 43

```
##           253           73           17           22           15
##      Wildling
##           40
```

```
!           ,           ?      table()
:
```

```
table(got$Allegiances, got$Gender)
```

```
##
##           0   1
##   Arryn           3  20
##   Baratheon        6  50
##   Greyjoy          4  47
##   House Arryn       3   4
##   House Baratheon   0   8
##   House Greyjoy     1  23
##   House Lannister   2  19
##   House Martell     7   5
##   House Stark       6  29
##   House Targaryen   5  14
##   House Tully       0   8
##   House Tyrell      4   7
##   Lannister        12  69
##   Martell          7  18
##   Night's Watch     0 116
##   None            51 202
##   Stark            21  52
##   Targaryen         1  16
##   Tully             2  20
##   Tyrell            6   9
##   Wildling         16  24
```

### 3.1.2 Subsetting

```

: [ _ _ _ , _ _ _ ] ,
got[100:115, 1:2]
```

```
##           Name   Allegiances
## 100      Blue Bard House Tyrell
## 101  Bonifer Hasty   Lannister
## 102         Borcas Night's Watch
## 103 Boremund Harlaw   Greyjoy
## 104   Boros Blount   Baratheon
## 105         Borroq   Wildling
```

```
## 106      Bowen Marsh Night's Watch
## 107      Bran Stark   House Stark
## 108      Brandon Norrey      Stark
## 109      Brenett      None
## 110      Brienne of Tarth    Stark
## 111      Bronn      Lannister
## 112      Brown Bernarr Night's Watch
## 113      Brusco      None
## 114      Bryan Fossoway      Baratheon
## 115      Bryce Caron      Baratheon
```

```
:
```

```
got[508:515, "Name"]
```

```
## [1] "Mance Rayder"      "Mandon Moore"      "Maric Seaworth"    "Marei"
## [5] "Margaery Tyrell"  "Marillion"         "Maris"             "Marissa Frey"
```

```
!
```

```
got[508:515, c("Name", "Allegiances", "Gender")]
```

```
##           Name      Allegiances Gender
## 508      Mance Rayder      Wildling      1
## 509      Mandon Moore      Baratheon      1
## 510      Maric Seaworth House Baratheon      1
## 511      Marei      None      0
## 512      Margaery Tyrell House Tyrell      0
## 513      Marillion      Arryn      1
## 514      Maris      Wildling      0
## 515      Marissa Frey      None      0
```

```
:
```

```
houses <- got$Allegiances
```

```
unique(houses) # - table()
```

```
## [1] "Lannister"      "None"      "House Targaryen" "House Greyjoy"
## [5] "Baratheon"      "Night's Watch" "Arryn"      "House Stark"
## [9] "House Tyrell"    "Tyrell"      "Stark"      "Greyjoy"
## [13] "House Lannister" "Martell"      "House Martell" "Wildling"
## [17] "Targaryen"      "House Arryn"  "House Tully"  "Tully"
## [21] "House Baratheon"
```

```
, -
- Allegiances "Night's
Watch"
```

```
vectornight <- got$Allegiances == "Night's Watch"
head(vectornight)
```

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

```
TRUE FALSE
```

```
?
```

```
nightswatch <- got[vectornight, ]
head(nightswatch)
```

```
##              Name      Allegiances Death.Year Book.of.Death
## 7  Aemon Targaryen (son of Maekar I) Night's Watch      300      4
## 10              Aethan Night's Watch      NA      NA
## 13              Alan of Rosby Night's Watch      300      5
## 16              Albett Night's Watch      NA      NA
## 24              Alliser Thorne Night's Watch      NA      NA
## 49              Arron Night's Watch      NA      NA
##  Death.Chapter Book.Intro.Chapter Gender Nobility GoT CoK SoS FfC DwD
## 7              35              21      1      1  1  0  1  1  0
## 10             NA              0      1      0  0  0  1  0  0
## 13              4              18      1      1  0  1  1  0  1
## 16             NA              26      1      0  1  0  0  0  0
## 24             NA              19      1      0  1  1  1  0  1
## 49             NA              75      1      0  0  0  1  0  1
```

```
!
```

```
:
```

```
nightswatch <- got[got$Allegiances == "Night's Watch", ]
```

```
!
```

```
(Wildling)
```

```
, | ( )
```

```
:
```

```
nightwatch_wildling <-
  got[got$Allegiances == "Night's Watch" | got$Allegiances == "Wildling", ]
head(nightwatch_wildling)
```

```
##              Name      Allegiances Death.Year Book.of.Death
## 7  Aemon Targaryen (son of Maekar I) Night's Watch      300      4
## 10              Aethan Night's Watch      NA      NA
## 13              Alan of Rosby Night's Watch      300      5
## 16              Albett Night's Watch      NA      NA
## 24              Alliser Thorne Night's Watch      NA      NA
## 49              Arron Night's Watch      NA      NA
##  Death.Chapter Book.Intro.Chapter Gender Nobility GoT CoK SoS FfC DwD
## 7              35              21      1      1  1  0  1  1  0
## 10             NA              0      1      0  0  0  1  0  0
## 13              4              18      1      1  0  1  1  0  1
## 16             NA              26      1      0  1  0  0  0  0
```

```
## 24      NA      19      1      0      1      1      1      0      1
## 49      NA      75      1      0      0      0      1      0      1

      : got[got$Allegiances == c("Night's
Watch", "Wildling"),].
      .
      .      :      recycling.

      %in%,

      %in%,      - TRUE:
1:6 %in% c(1, 4, 5)

## [1] TRUE FALSE FALSE TRUE TRUE FALSE
nightwatch_wildling <- got[got$Allegiances %in% c("Night's Watch", "Wildling"), ]
head(nightwatch_wildling)

##              Name      Allegiances Death.Year Book.of.Death
## 7  Aemon Targaryen (son of Maekar I) Night's Watch      300      4
## 10              Aethan Night's Watch      NA      NA
## 13      Alan of Rosby Night's Watch      300      5
## 16      Albett Night's Watch      NA      NA
## 24      Alliser Thorne Night's Watch      NA      NA
## 49      Arron Night's Watch      NA      NA
##      Death.Chapter Book.Intro.Chapter Gender Nobility GoT CoK SoS FfC DwD
## 7      35      21      1      1      1      0      1      1      0
## 10      NA      0      1      0      0      0      1      0      0
## 13      4      18      1      1      0      1      1      0      1
## 16      NA      26      1      0      1      0      0      0      0
## 24      NA      19      1      0      1      1      1      0      1
## 49      NA      75      1      0      0      0      1      0      1
```

### 3.1.3

```
, , ( ). , ,
      ,
      " "
Death.Year, Death.Chapter Book.of.Death      NA
"Arya Stark",      :
got[got$Name == "Arya Stark", ]

##              Name Allegiances Death.Year Book.of.Death Death.Chapter
## 56 Arya Stark      Stark      NA      NA      NA
##      Book.Intro.Chapter Gender Nobility GoT CoK SoS FfC DwD
## 56      2      0      1      1      1      1      1      1
```

```

      ,      Book.of.Death      NA,      ,
      .

```

```

      Is.Alive:
got$Is.Alive <- is.na(got$Book.of.Death)

```

```

!      ,      ,      ?      ,
      .

```

## 3.2 , ,

### 3.2.1 If, else, else if

“ ” , R if-else statements.

```

:
na_slovah <- "      "
if (na_slovah == "      "){
  na_dele = "      "
} else {na_dele = na_slovah}
na_dele

```

```

## [1] "      "

      if -      .      TRUE,
      ,      ,      else (      else      ).

```

```

:
na_slovah <- "      "
if (na_slovah == "      "){
  na_dele = "      "
} else if (na_slovah == "      "){
  na_dele = "      "
} else {na_dele = na_slovah}
na_dele

```

```

## [1] "      "

      , if, else, else if      -      ,
      .      !      ifelse()      .

```

### 3.2.2 ifelse()

```

        ifelse()      - 1)      ( ..,      ,      TRUE
FALSE), 2)      TRUE, 3)      FALSE.

```

Is.Alive.                      NULL:

```
got$Is.Alive <- NULL
```

```

, ifelse():

```

```
got$Is.Alive <- ifelse(is.na(got$Book.of.Death), "Alive", "Dead")
```

```

, else if . ,
  ifelse() ifelse()

```

### 3.2.3 For loops

( for while) - R. R ,

 $(\quad, \quad), \quad \mathbb{R}.$ 

, , ! R .

```
got$Is.Alive <- NULL
```

```
got$Is.Alive <- character(nrow(got)) #
```

```
for (i in 1:nrow(got)) {
  if (is.na(got$Book.of.Death[i])) {
    got$Is.Alive[i] <- "Alive"
  } else {
    got$Is.Alive[i] <- "Dead"
  }
}
```

$$\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$$
$$\left( \begin{array}{c} \vdots \\ \vdots \\ \vdots \end{array} \right), \quad \text{for}$$

•





```
sumofsquares <- function(x) {
  centralized_x <- x - mean(x)
  squares <- centralized_x ^ 2
  sum(squares)
}
sumofsquares(1:10)
```

```
## [1] 82.5
```

```
sumofsquares <- function(x) {
  sum((x - mean(x)) ^ 2)
}
sumofsquares(1:10)
```

```
## [1] 82.5
```

```
sumofsquares <- function(x) sum((x - mean(x)) ^ 2)
sumofsquares(1:10)
```

```
## [1] 82.5
```

```

,
.
? “ ” -
. , ,
,
.
,
,
,
R),
,
— R.
```

The reason for writing a function is not to reuse its code, but to name the operation it performs.

— Tim “Agile Otter” Ottinger (?) January 22, 2013

### 3.2.5 C apply()

```
? , : apply(), lapply(), sapply(), vapply(), tapply(), mapply(),
rapply()... , .
```

```

A <- matrix(1:12, 3, 4)
A

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

, " " - ( , ) . apply()

...), X — , MARGIN 1 ( ), 2 ( ), c(1,2)
(..), FUN — , (!)
apply() / X .

```

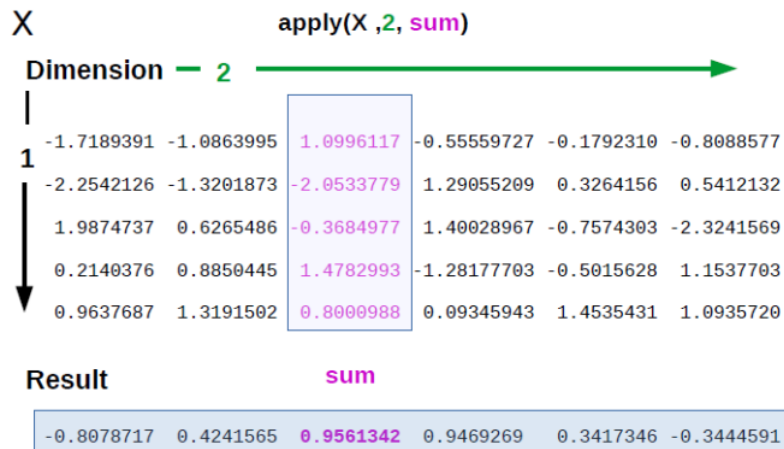


Figure 3.1: apply

```

:

apply(A, 1, sum) #

## [1] 22 26 30

apply(A, 2, sum) #

## [1] 6 15 24 33

apply(A, c(1,2), sum) # ...

##      [,1] [,2] [,3] [,4]
## [1,]    1    4    7   10
## [2,]    2    5    8   11

```

```
## [3,] 3 6 9 12
      ,      (      !)
```



Figure 3.2:

### 3.2.6

```
,      :      apply()
      ,      ...      ?
      !
```

FUN:

```
apply(A, 1, weighted.mean, w = c(0.2, 0.4, 0.3, 0.1))
```

```
## [1] 4.9 5.9 6.9
```

```
-      ,
      -      .      ,
```

```
,      :
```

```
apply(A, 1, function(x) sum((x-mean(x))^2))
```

```
## [1] 45 45 45
```

```
apply(A, 2, function(x) sum((x-mean(x))^2))
```

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

```
apply(A, c(1,2), function(x) sum((x-mean(x))^2))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    0    0    0
## [2,]    0    0    0    0
## [3,]    0    0    0    0
```

```
apply(A, 1, function(whatevername) sum((whatevername-mean(whatevername))^2))
```

```
## [1] 45 45 45
```

```
, apply()
lapply ( " " ) sapply() - lapply(),
( got$Is.Alive), sapply():
got$Is.Alive <- NA
got$Is.Alive <- sapply(got$Book.of.Death, function (x) ifelse(is.na(x), "Alive", "Dead"))
```

```
lapply() sapply()
( . ??), :
```

```
lapply(got, class)
```

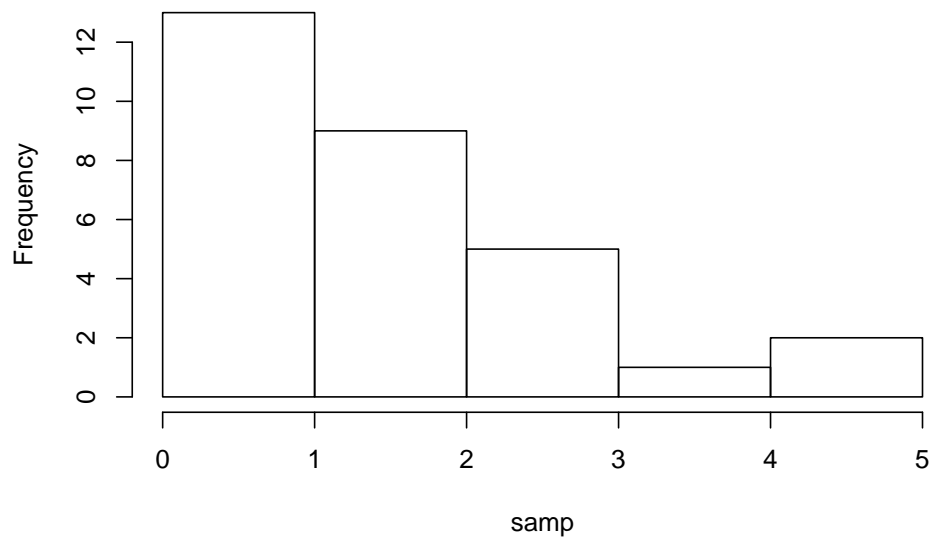
```
## $Name
## [1] "character"
##
## $Allegiances
## [1] "character"
##
## $Death.Year
## [1] "integer"
##
## $Book.of.Death
## [1] "integer"
##
## $Death.Chapter
## [1] "integer"
##
## $Book.Intro.Chapter
## [1] "integer"
##
```

```
## $Gender
## [1] "integer"
##
## $Nobility
## [1] "integer"
##
## $GoT
## [1] "integer"
##
## $CoK
## [1] "integer"
##
## $SoS
## [1] "integer"
##
## $FfC
## [1] "integer"
##
## $DwD
## [1] "integer"
##
## $Is.Alive
## [1] "character"
```

`apply()` - `replicate()` -

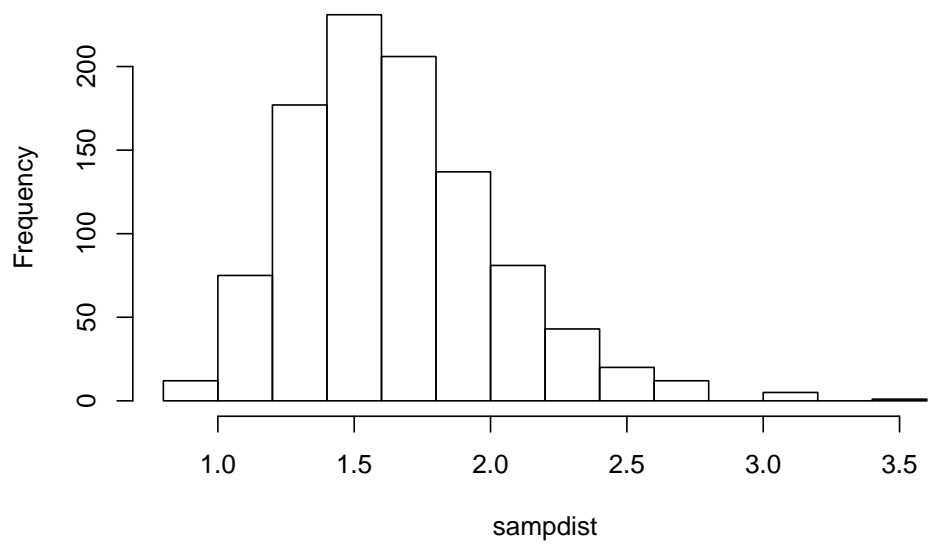
:

```
set.seed(1) #
samp <- rlnorm(30)
hist(samp)
```

**Histogram of samp**

1000 :

```
sampdist <- replicate(1000, mean(rlnorm(30)))  
hist(sampdist)
```

**Histogram of sampdist**

```
apply() , .
```

### 3.3

```

- . R !. -
- “ ” (regular expressions, regex,
regex). R.
, , .
- Name ( ) Allegiances ( ,
/ ).
Allegiances "House Stark", , "Stark".
grep(). , - , ,
( ).
fixed = TRUE.
( , R ).
, - - “ .
- ”. , “ ”!
.
grep()
fixed = TRUE.

```

```
grep("Stark", got$Allegiances, fixed = TRUE)
```

```
## [1] 17 25 29 30 47 53 56 65 69 85 90 91 107 108 110 127 128 133
## [19] 141 155 161 175 183 194 198 200 209 217 218 227 250 260 262 265 272 286
## [37] 326 328 340 342 343 346 348 353 362 367 381 392 397 398 405 411 413 414
## [55] 417 419 448 464 465 467 471 489 500 518 533 534 539 550 561 570 576 581
## [73] 590 607 613 623 645 647 664 686 697 698 699 702 705 706 709 713 717 726
## [91] 744 775 783 789 799 817 820 856 872 876 879 881 894 896 897 898 899 912
```

```

— , , :
starks <- got[grep("Stark", got$Allegiances, fixed = TRUE), ]
table(starks$Allegiances)

```

```
##
## House Stark Stark
## 35 73
```

```
!
```

```

stringsAsFactors = FALSE,
“ ” , R.
( ),
droplevels(). character .

```



```

c      "House Stark"      "Stark",
      "House "           ""      gsub().
      grep(),             ("House "),      (""),
      got$Allegiances
(      got$Houses):
got$Houses <- gsub("House ", "", got$Allegiances, fixed = TRUE)
table(got$Allegiances)

##
##      Arryn      Baratheon      Greyjoy      House Arryn House Baratheon
##      23          56          51          7          8
## House Greyjoy House Lannister House Martell House Stark House Targaryen
##      24          21          12          35          19
## House Tully   House Tyrell   Lannister      Martell   Night's Watch
##      8          11          81          25          116
##      None      Stark      Targaryen      Tully      Tyrell
##      253        73          17          22          15
##      Wildling
##      40

      : nchar() -
      , ,
      :
max(nchar(got$Name))

## [1] 33
33 ! , ?
longest <- which.max(nchar(got$Name)) #index of the longest name
got[longest, 1:2]

##
##      Name      Allegiances
## 7 Aemon Targaryen (son of Maekar I) Night's Watch
, , ,
      substr().      "      ":
      :
aemon <- substr(got$Name[longest], 1, 15)
aemon

## [1] "Aemon Targaryen"
got$Name[longest] <- aemon

, , ,
paste() paste0(). paste()      sep =,

```

```

, paste0() - paste() :
paste("R", "is", "love")

## [1] "R is love"
paste0("R", "is", "love")

## [1] "Rislove"

: paste() , .
- collapse
= ( NULL):
phrase <- paste(c("All", "you", "need", "is", "love"), collapse = " <3 ")
phrase

## [1] "All <3 you <3 need <3 is <3 love"

strsplit() - :
strsplit(phrase, split = " <3 ")

## [[1]]
## [1] "All" "you" "need" "is" "love"

, C format (printf-style formatting), R
sprintf():
sprintf("%i ; %s ", 20000, " ")

## [1] "20000 ; "

. , .
- . , R
, . , ,

```

### 3.4

```

R - ( . . ) . R , R
15000 ( 2020 ), Comprehensive
R Archive Network (CRAN) install.packages(),
install.packages(c("data.table", "dplyr"))

```

!

15000+  
( , Hmisc). ,  
Packages

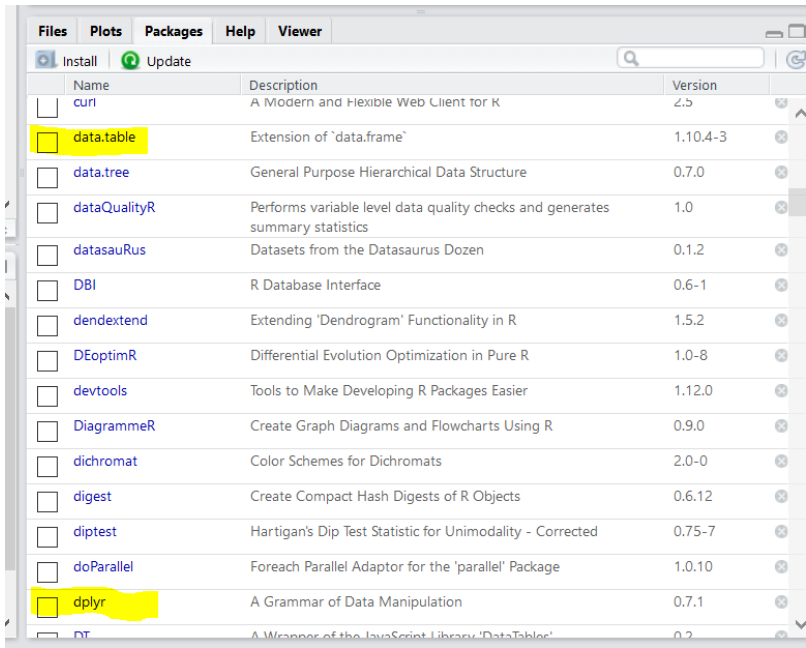


Figure 3.3:

```
library("dplyr")
```

3.5

```
bat <- read.csv("data/battles.csv")
```

, , - , ,  
:  
,

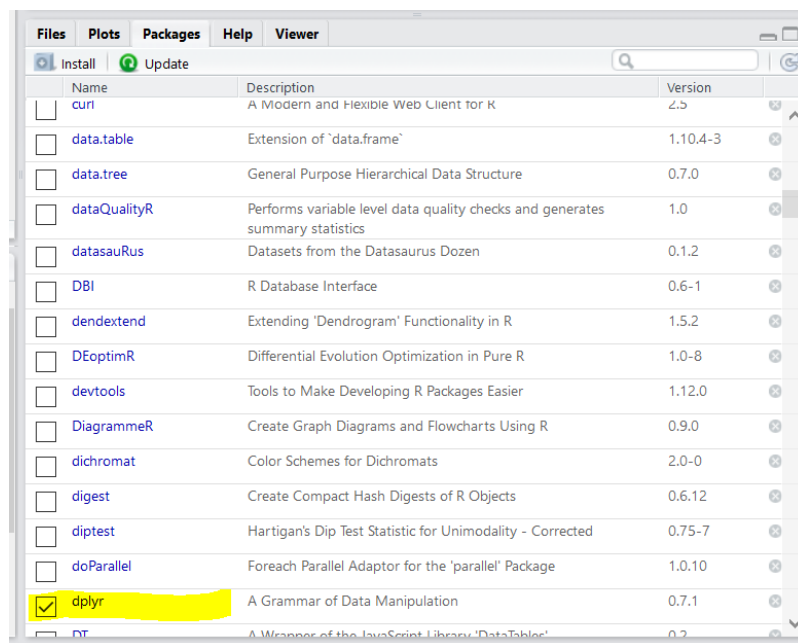


Figure 3.4:

```
mean(bat[bat$year == 298, "attacker_size"], na.rm = T)
```

```
## [1] 11175
```

```
mean(bat[bat$year == 299, "attacker_size"], na.rm = T)
```

```
## [1] 5134.308
```

```
mean(bat[bat$year == 300, "attacker_size"], na.rm = T)
```

```
## [1] 19333.33
```

```

,
? , ! ,
, , , ,
,
R, , :

```

```
sapply(unique(bat$year), function(x) mean(bat$attacker_size[bat$year == x], na.rm = T))
```

```
## [1] 11175.000 5134.308 19333.333
```

```

,
- aggregate(), split(),
.
,
- ,
,

```

? , - , ... , -  
 - ,  
 ( , ) .

### 3.5.1 data.table vs. dplyr

```

dplyr      ggplot2 ( tidy, stringr, lubridate, devtools,
httr, readr R) .

RStudio, . , Help
- Cheatsheets: dplyr. data.table =)

R, . , ,
, .

library("dplyr")
bat %>% group_by(year) %>% summarise(mean(attacker_size, na.rm = T))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 3 x 2
##   year `mean(attacker_size, na.rm = T)`
##   <int>           <dbl>
## 1   298           11175
## 2   299           5134.
## 3   300          19333.

, : , -
.

%>% “ ” (pipe), .. “ ”. ,
,
.
, magrittr. , dplyr,
, data.table!

dplyr (magrittr, purrr, stringr, readr, tidy,
tibble ..), “tidyverse”. R.
, stringr ( ),
readr , purrr - apply(). ,
!

- data.table. R, ,
“ ” . “ ” ,
. data.table “ ” :
```

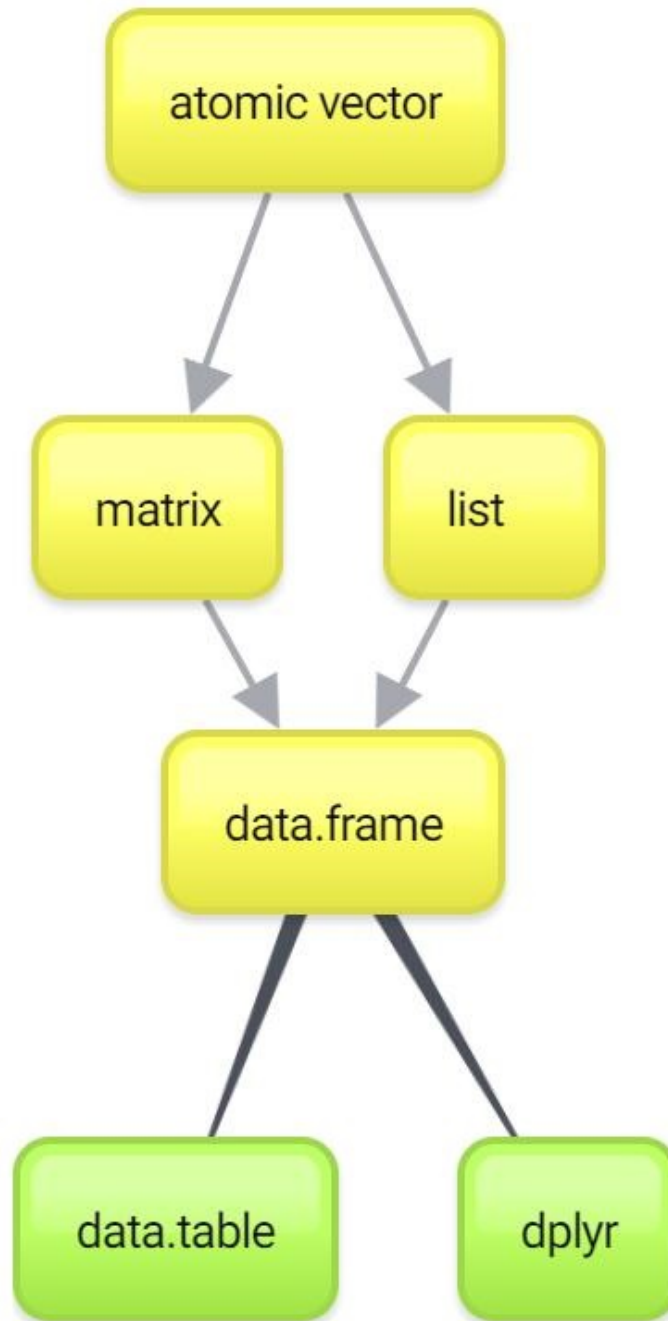


Figure 3.5:

```
library("data.table")
batdt <- as.data.table(bat)
batdt[,mean(attacker_size, na.rm = T), by = year]
```

```
##      year      V1
## 1:  298 11175.000
## 2:  299  5134.308
## 3:  300 19333.333
```

```
data.frame,
by = - ( )
```

### 3.5.2 ?

```
data.table,
-
,
```

### 3.5.3 data.table

```
fread(). read.table(),
( ! ) ( ).
- !
```

```
batdt <- fread("data/battles.csv")
```

```
,
,
,
:
```

```
class(batdt)
```

```
## [1] "data.table" "data.frame"
```

```
! , ( ).
,
data.table ,
data.table = FALSE -
" " , R ,
:: :
```

```
batdataframe <- data.table::fread("data/battles.csv", data.table = FALSE)
```

```
::
,
,
```

### 3.5.3.1 data.table

```
data.table(      ,      SQL (      ,      ,      ;
      ,      ).      :
```

```
DT[i, j, by]
```

```
i -      ,      .      data.frame,      ? j -      ,
.      -      .      -      -
j, ..      ! by -      .
```

“General form: DT[i, j, by] “Take DT, subset rows using i, then calculate j grouped by by” ( data.table).

SQL, i = WHERE, j = SELECT | UPDATE, by = GROUP BY.

```
,      i, j by.
```

```
.      ,      ,      .
.      ,      i, ..      .
-      ,      ,      (i),
(j).
```

```
,      ,      ,      :
batdt[attacker_outcome == "win", mean(attacker_size, na.rm = TRUE), by = region]
```

```
##      region      V1
## 1: The Westerlands 9000.000
## 2: The Riverlands 4425.000
## 3: The North 1107.667
## 4: The Stormlands 3500.000
## 5: The Reach      NaN
!
```

```
:
```

- i: , attacker\_outcome "win". ,  
( ), !
- j: j . - .
- by: .

```
!
```

```
V1 (      ,      V2      .),
:
```

```
batdt[attacker_outcome == "win",
      .(mean_attack = mean(attacker_size, na.rm = TRUE)),
      by = region]
```



```
##           region mean_attack
## 1: The Westerlands    9000.000
## 2: The Riverlands    4425.000
## 3:      The North    1107.667
## 4: The Stormlands    3500.000
## 5:      The Reach      NaN
```

```
.() -           , list().           ,
      :
```

```
batdt[attacker_outcome == "win",
      .(mean_attack = mean(attacker_size, na.rm = TRUE),
        max_attacker = max(attacker_size, na.rm = TRUE)),
      by = region]
```

```
## Warning in gmax(attacker_size, na.rm = TRUE): No non-missing values found in at
## least one group. Coercing to numeric type and returning 'Inf' for such groups to
## be consistent with base
```

```
##           region mean_attack max_attacker
## 1: The Westerlands    9000.000    15000
## 2: The Riverlands    4425.000    15000
## 3:      The North    1107.667     4500
## 4: The Stormlands    3500.000     5000
## 5:      The Reach      NaN      -Inf
```

```
, .() by:
```

```
batdt[,.(mean_attack = mean(attacker_size, na.rm = TRUE)), by = .(region, attacker_outcome)]
```

```
##           region attacker_outcome mean_attack
## 1: The Westerlands      win    9000.000
## 2: The Riverlands      win    4425.000
## 3: The Riverlands      loss   19000.000
## 4:      The North      win    1107.667
## 5: The Stormlands      win    3500.000
## 6: The Crownlands      loss   12000.000
## 7: Beyond the Wall      loss  100000.000
## 8:      The Reach      win         NaN
## 9:      The North      win    5000.000
```

### 3.5.3.2

```
data.table           :=
      ,           ,
      :
```

```
batdt[,outcome:=ifelse(attacker_outcome == "win",
                        "win",
                        "loss")]
```

```

        ifelse(attacker_outcome == "loss",
               " ",
               " "))]]

    :=
    , ..
    .

    ,
    : batdt

    .

    , := :

batdt[, ':='(all_army = attacker_size + defender_size,
            ratio_army = attacker_size / defender_size)]

```

### 3.5.3.3 Chaining

*Chaining* ( ) - - , ,

data.table

“ ” - data.table

data.table.

,

,

data.table .N,

length() - :

```

batdt[.N,]

##           name year battle_number   attacker_king
## 1: Siege of Winterfell 300           38 Stannis Baratheon
##           defender_king attacker_1 attacker_2 attacker_3 attacker_4
## 1: Joffrey/Tommen Baratheon Baratheon Karstark  Mormont  Glover
##   defender_1 defender_2 defender_3 defender_4 attacker_outcome battle_type
## 1:   Bolton   Frey      NA      NA
##   major_death major_capture attacker_size defender_size attacker_commander
## 1:      NA      NA      5000      8000 Stannis Baratheon
##   defender_commander summer   location   region note      outcome
## 1:   Roose Bolton    0 Winterfell The North
##   all_army ratio_army
## 1:   13000    0.625

, .N j(..). .N i,

, - :

batdt[,.N, by = region]

```

```
##           region  N
## 1: The Westerlands 3
## 2: The Riverlands 17
## 3:      The North 10
## 4: The Stormlands 3
## 5: The Crownlands 2
## 6: Beyond the Wall 1
## 7:      The Reach 2
```

```
table()!
```

```
, " "
```

```
batdt[,.N, by = region][order(-N),]
```

```
##           region  N
## 1: The Riverlands 17
## 2:      The North 10
## 3: The Westerlands 3
## 4: The Stormlands 3
## 5: The Crownlands 2
## 6:      The Reach 2
## 7: Beyond the Wall 1
```

```
order(),
```

```
“ N,
```

```
batdt[,.N, by = region][order(-N),][N>2,]
```

```
##           region  N
## 1: The Riverlands 17
## 2:      The North 10
## 3: The Westerlands 3
## 4: The Stormlands 3
```

```
, “ ” , N
```

```
:
```

```
batdt[,.N, by = region
][order(-N),
][N>2,]
```

```
##           region  N
## 1: The Riverlands 17
## 2:      The North 10
## 3: The Westerlands 3
## 4: The Stormlands 3
```

3.5.4

– ( – ? ) , “ . ” ,  
 , – ( – ) ? ,  
“ ” , “ ” .

3.5.4.1 “ ”

R		R
70		63
80		74
86		71

3.5.4.2 “ ”

( )	
R	70
R	80
R	86
R	63
R	74
R	71

3.5.5 data.table: melt() dcast()

, data.table :  
• melt() (= “ ”):  
• dcast()(= “ ”):

3.5.5.1 1: melt()

```
batdt      ,      : attacker_size  
defender_size.  
head(batdt[, .(name, year, attacker_size, defender_size)])  
##           name year attacker_size defender_size
```

```
## 1: Battle of the Golden Tooth 298 15000 4000
## 2: Battle at the Mummer's Ford 298 NA 120
## 3: Battle of Riverrun 298 15000 10000
## 4: Battle of the Green Fork 298 18000 20000
## 5: Battle of the Whispering Wood 298 1875 6000
## 6: Battle of the Camps 298 6000 12625
```

```

      :
      " " battle_role, army_size:
batlong <- melt(batdt,
               measure.vars = c("attacker_size", "defender_size"),
               variable.name = "battle_role",
               value.name = "army_size")

```

```

      batlong
defender_size      , attacker_size
      battle_role.

      melt():
• data - data.table
• id.vars - id. , " "
• measure.vars - ( .. !), Note:
  melt() , id.vars measure.vars.
• variable.name - " "
• value.name -

```

### 3.5.5.2 2: dcast()

```

      !
dcast() . , ,
      ,
      R ,
class(y ~ x1 + x2 * x3)

## [1] "formula"

      (~ - " "),

      dcast()

batwide <- dcast(batlong,
                 ... ~ battle_role,
                 value.var = "army_size")

      batdt,

```

### 3.5.6 rbind(), cbind() merge()

```
bat_at <- batlong[battle_role == "attacker_size",]
bat_def <- batlong[battle_role == "defender_size",]
```

!

: rbind(), cbind() merge().

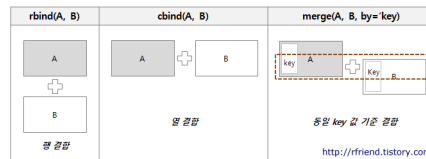


Figure 3.6:

```
verylong_bat <- cbind(bat_at, bat_def) #c stands for columns
h_bat
verywide_bat <- rbind(bat_at, bat_def) #r stands for rows
```

```
batdt[order(-N),][N>2,]

##           region  N
## 1: The Riverlands 17
## 2:      The North 10
## 3: The Westerlands  3
## 4: The Stormlands  3

hot_regions.
```

```
hot_regions <- batdt[,.N, by = region][order(-N),][N>2,]
```

```
merge():
```

```
subset_batdt <- merge(hot_regions, batdt,
  by = "region",
  all.x = TRUE, all.y = FALSE)
```

```
!
merge() - . by =
.
. by.x = by.y =
- all.x = all.y =
( - ) TRUE FALSE,
4 :
```

- all = T: " " : - ,
  - all.x = T, all.y = F: ,
  - all.x = F, all.y = T: ,
  - all.x = F, all.y = F: ,
- batdt. hot\_regions ,

## 3.6

```
, R ( , ).
, , ... , , melt(), dcast()
merge() , - ,
.
data.table ,
```





## Chapter 4

### 3.

#### 4.1

(*descriptive statistics*) (inferential statistics). (sample, . . . , , (population) - - “ ” . , (sample) (statistics), (Population) (Parameters). . (point estimators) , , ( ) - ( , , ), .

survival, pbc.

data.table

```
library(survival)
library(data.table)
```

```
data(pbc)
pbc<dt <- as.data.table(pbc)
```

424 - . , .

“This data is from the Mayo Clinic trial in primary biliary cirrhosis (PBC) of the liver conducted between 1974 and 1984. A total of 424

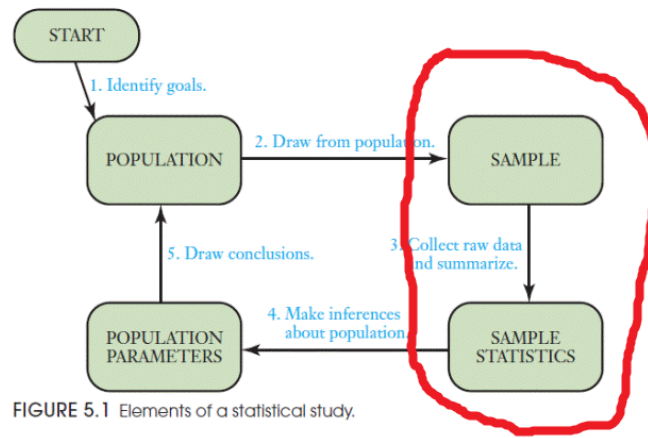


Figure 4.1:

PBC patients, referred to Mayo Clinic during that ten-year interval, met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine. The first 312 cases in the data set participated in the randomized trial and contain largely complete data.”

Help.

```

( , , ).

: complete.cases() , is.na(),
NA, TRUE , FALSE ,

pbcddt <- pbcddt[complete.cases(pbc),]

a

a <- pbcddt$age

```

#### 4.1.1

## 4.1.1.1

```
mean().
```

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$$

$$\sum_{i=1}^n$$

```
- i = 1 n. - for! mymean() c sum()
length(). NA ! mean().
mean(a)
```

```
## [1] 49.79966
```

## 4.1.1.2

```
- . , ( )
. , .
median():
median(a)
```

```
## [1] 49.70979
```

```
. , " "
- .
, - .
, !
.
, 8000-
:
mean(c(a, 8000))
```

```
## [1] 78.50075
```

```
median(c(a, 8000))
```

```
## [1] 49.76318
```

“ ” , .

#### 4.1.1.3 (trimmed mean)

, ( ) , ,

, , mean(),

```
trim =:
```

```
mean(a, trim = 0.1)
```

```
## [1] 49.57392
```

```
trim = 0.1 , 10% 10% . trim 0
0.5. , trim = 0?
```

```
mean(a, trim = 0)
```

```
## [1] 49.79966
```

```
! trim = 0.5?
```

```
mean(a, trim = 0.5)
```

```
## [1] 49.70979
```

```
!
```

#### 4.1.1.4

(mode) - , R ( ).

```
mymode <- function(x){names(which.max(table(x)))}
mymode(pbcddt$sex)
```

```
## [1] "f"
```

#### 4.1.2

1 .

#### 4.1.2.1 {range}

- `range()` returns the range of the values in the vector `a`. In R:

```
range(a)
```

```
## [1] 26.27789 78.43943
```

```
diff(range(a))
```

```
## [1] 52.16153
```

#### 4.1.2.2

`var()` returns the variance of the values in the vector `a`:

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

`myvar()`!

```
myvar <- function(x) mean((x - mean(x))^2)
```

, R `var()`. , , :

```
myvar(a)
```

```
## [1] 110.334
```

```
var(a)
```

```
## [1] 110.7353
```

, `var()`  $n$ ,  $n - 1$ . , , .

#### 4.1.2.3

, , (standard deviation):

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

```
sd():
```

```
sd(a)
```

```
## [1] 10.52308
```

```
, : 
```

```
sqrt(var(a))
```

```
## [1] 10.52308
```

#### 4.1.2.4

( , ) - (median absolute deviation):

$$mad = median(|x_i - median(x)|)$$

```
mad():
```

```
mad(a)
```

```
## [1] 10.63291
```

#### 4.1.2.5

*IQR*). (interquartile range, 25% , 75% , 1 - , 63% .

```
IQR(a)
```

```
## [1] 15.07187
```

```
- !
```

```
1 — , 63% , 63% .
```