

R

2020-11-10

Contents

1		7
2	R	9
2.1	R Rstudio	9
2.2	RStudio	10
2.3	R	11
2.4		13
2.5	—	15
2.6		16
2.7		19
2.8		20
3		25
3.1	atomic R	25
3.2		29
3.3		30
3.4		35
3.5	NA -	37
3.6		39
4	R	41
4.1		41
4.2		43
4.3		43
4.4		46
5	R	49
5.1		49
5.2	R	50
5.3	CRAN	50
5.4		50
5.5	::	51
5.6	c Bioconductor	51
5.7	Github	52

5.8	56
6		57
6.1	RStudio	57
6.2	60
6.3	61
6.4	: Excel, SPSS	62
6.5	62
7		67
7.1	if, else, else if	67
7.2	for	69
7.3	: ifelse() dplyr::case_when() ..	70
8	R	73
8.1	73
8.2	74
8.3	?	76
8.4	76
8.5	apply()	77
9	tidyverse	83
9.1	tidyverse	83
9.2	readr	84
9.3	tibble	85
9.4	magrittr::%>%	86
9.5	tidyverse: dplyr tidyR	87
9.6	88
9.7	98
9.8	: dplyr::mutate() dplyr::transmute()	103
9.9	105
9.10	: dplyr::across()	111
10	tidyverse	117
10.1	117
10.2	Tidy data: tidyR::pivot_longer(), tidyR::pivot_wider() ..	124
11		127
11.1	R	127
11.2	127
11.3	129
11.4	129
11.5	130
11.6	132
11.7	132
11.8	134
11.9	135

11.10	140
11.11	140
11.12	apply()	144
11.13	magrittr::%>%	146
11.14	: dplyr::slice() dplyr::filter()	147
11.15	: dplyr::select()	147
11.16	: dplyr::arrange()	149
11.17	: dplyr::distinct()	151
11.18	: dplyr::mutate() dplyr::transmute()	151
11.19	: dplyr::group_by() %>% summarise()	152
11.20	: across()	153
11.21	: *_join {#task_join}	154
11.22	Tidy data	155
12		159
12.1	R	159
12.2	160
12.3	161
12.4	162
12.5	164
12.6	166
12.7	167
12.8	169
12.9	170
12.10	175
12.11	177
12.12	apply()	181
12.13	magrittr::%>%	184
12.14	: dplyr::slice() dplyr::filter()	184
12.15	: dplyr::select()	185
12.16	: dplyr::arrange()	187
12.17	: dplyr::distinct()	189
12.18	: dplyr::mutate() dplyr::transmute()	190
12.19	: dplyr::group_by() %>% summarise()	191
12.20	: across()	191
12.21	: *_join {#solution_join}	193
12.22	Tidy data	195

Chapter 1

“R”.
!
ivanspozdniakov@gmail.com, VK
gram:@pozdniakovivan. Telegram

Chapter 2

R

2.1 R Rstudio

R .

- R
 - Windows, [Download R \(\) for Windows.](#)
 - Mac, , 5 , *.pkg ,
 - Linux, :

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

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

```
## [1] "R version 4.0.2 (2020-06-22)"
```

R RStudio:

- RStudio
 - (), , :
- RStudio cloud
 - RStudio? , RStudio — (IDE),

RStudio — R, , , .
Jupyter Notebook, R

RMarkdown —
RMarkdown

2.2 RStudio

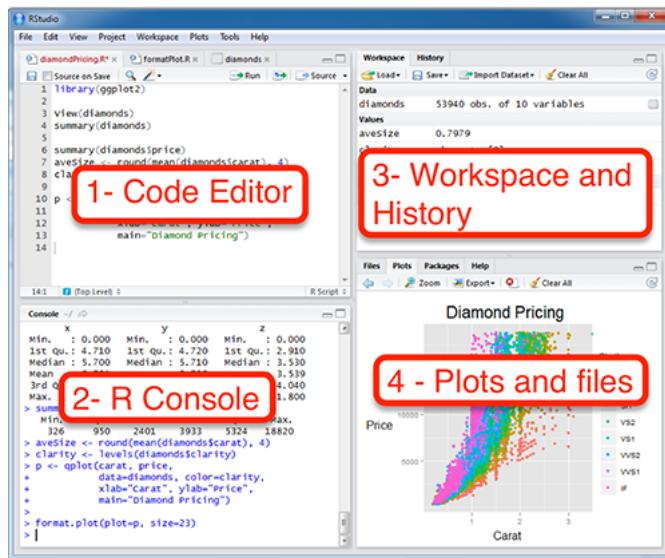


Figure 2.1:

: 1 - Code Editor ()¹ 2 - R
Console ().

2 - R Console

Enter.

1 - Code Editor
Enter macos).

Ctrl + Enter (Cmd +

Ctrl

+ A Windows Linux, Cmd + A macOS².
(),

2 - R Console,

1 RStudio
Script.

File - New File - R

² RStudio
Help.

Help - Keyboard Shortcuts

File - Save
As... R .R,
“ ”

3 - Workspace and History —

4 - Plots and files. (Packages) Help

2.3 R

R — , R
, , R
, , R +, -,
*, /, ^ (), (), ..

40+2

```
## [1] 42
3-2
```

```
## [1] 1
5*6
```

```
## [1] 30
99/9 #
```

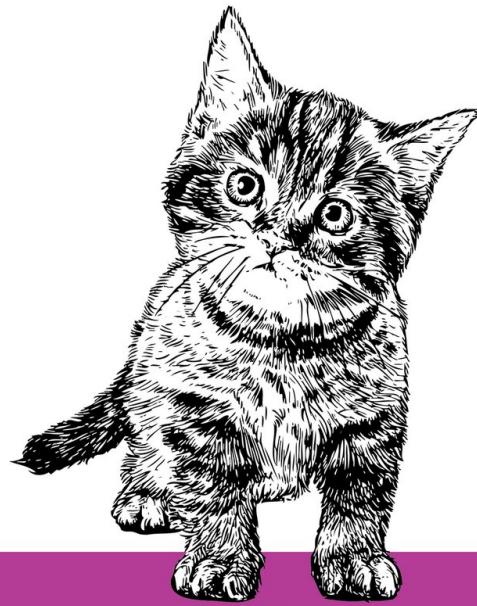
```
## [1] 11
2^3 #
```

```
## [1] 8
13 %/% 3 #
```

```
## [1] 4
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      (#). ,      #
( . - ),      ,
,      “ ” ,      ,
:      #      ,
: Ctrl + Shift + C (Cmd + Shift +
#      .
C   macOS) —
,      14%
:      :
2 + 2 * 2

## [1] 6

,      -      ,      6 (
)
( . . . , operator precedence) R
,
(2+2)*2

## [1] 8

,      ,      ,      ,
?Syntax.

```

2.4

```

,      ,      ,
:
16^0.5

## [1] 4

,      .      :      (
      )      -      ,
      (      )      -      ,
      ,      . . ). -
,
,      ,      :
sqrt(16)

## [1] 4
—————
3
,
!
```

```
R — case-sensitive , ... . SQRT(16) .
:
log(8)

## [1] 2.079442

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

— .
,
?log

RStudio : ;
, , base = (2.7182818...),
.. . R —
, , , ,
log(x = 8, base = 2)

## [1] 3
... ( ):  

log(8,2)

## [1] 3
,
log(8, sqrt(4))

## [1] 3
,
log(base = 2, x = 8)
```

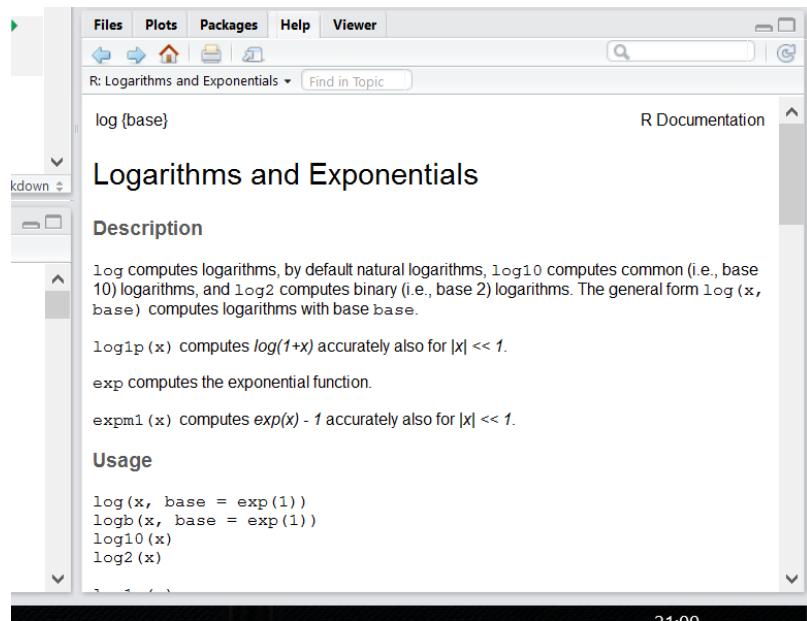


Figure 2.3:

```
## [1] 3
,
:
log(b = 2, x = 8)

## [1] 3
,
:
Python).
R
,
:
: +,-,/,\^ ...
:
'+'(3,4)

## [1] 7
```

2.5



Figure 2.4:

*Stackoverflow*⁴ R- !

Computer Programming To Be Officially Renamed “Googling Stack Overflow”
Source: <http://t.co/xu7acfXvFF> 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

: — , — .

2.6

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

⁴Stackoverflow —

Quora, The Question,

Mail.ru

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

Programmers:



Figure 2.5:

```

,          !
,          ,
a <- 2
a

## [1] 2

<-
5:           ,           ,           ,           ,
<-
           .           .
           :           ,           ,           ,
(           ,           )           ,
           ,
: - new variable - _new_variable - .1var - v-r
: - new_variable - .new.variable - var_2
           !
           ,
           (           -           )
           ,
: some_variable6.

```

Environment RStudio:

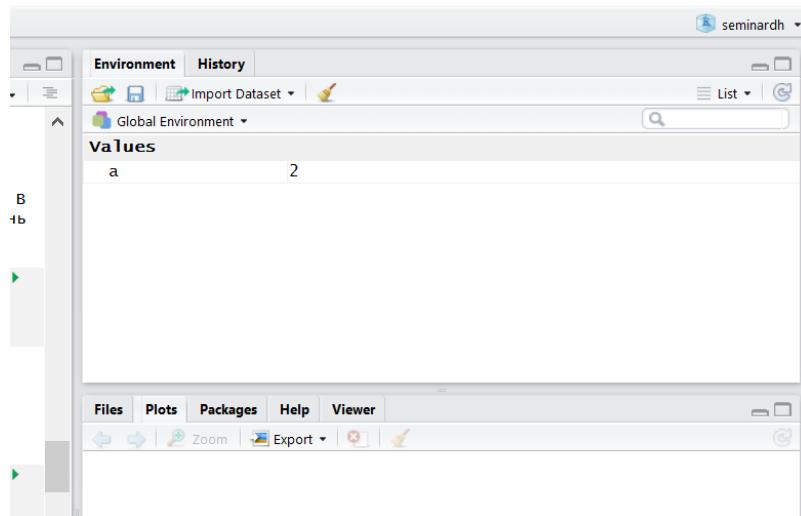


Figure 2.6:

5 ->, , , ,
6 SomeVariable, , , ,

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

## [1] 8
log(b, a)

## [1] 3
```

2.7

```
:
a == b

## [1] FALSE
,
          ==,      =.

a = b
a

## [1] 8
)
(
,
(           =( ).

:
a <- 2
b <- 3

a == b

## [1] FALSE
a != b

## [1] TRUE
R
/
:

a > b

## [1] FALSE
a < b

## [1] TRUE
```

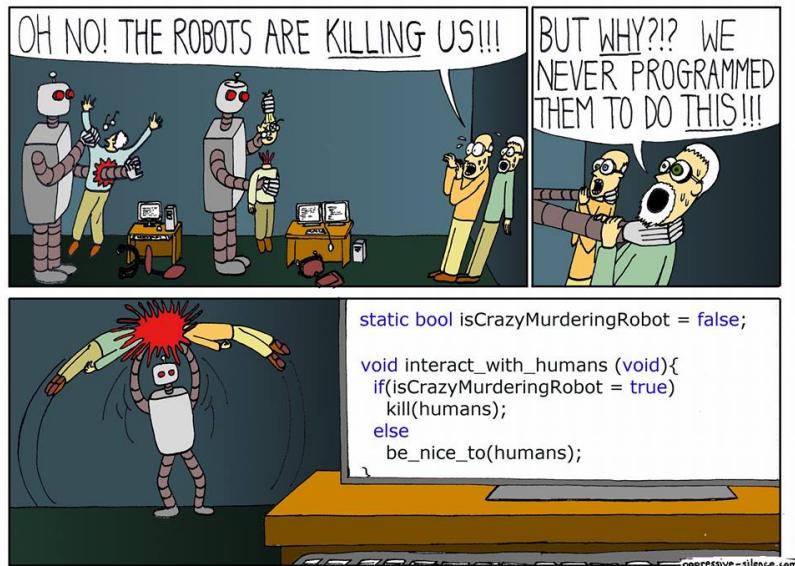


Figure 2.7:

```
a >= b
## [1] FALSE
a <= b
## [1] TRUE
```

!

2.8

```
(numeric):
class(a)
## [1] "numeric"
, R      numeric: integer( ), double( ), complex( )7.
R          numeric ,
```

⁷ $\text{R} : \text{complexnumber} \leftarrow 2+2\text{i}. \text{i} - 1.$

```

as.double() as.complex(), , as.integer(),
integer: L,
is.integer(5)

## [1] FALSE
is.integer(5L)

## [1] TRUE

double . , R
:
sqrt(2)^2 == 2

## [1] FALSE

, R., ,
all.equal():
all.equal(sqrt(2)^2, 2)

## [1] TRUE

R:
1. (character): , , .
s <- " !"
s

## [1] " !"
class(s)

## [1] "character"
", ' ( , - ).
"Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn"

## [1] "Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'nagl fhtagn"

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
rm() : 
, , , , !
comparison <- a == b
comparison
## [1] FALSE
, , , :
—      (!). !    TRUE  FALSE,  FALSE  TRUE:
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
-          (      ) — xor(),
TRUE       ,           TRUE.
,   8.          (      ) - R.           !
.
```


Chapter 3

3.1 atomic R

vector atomic) — (), ., (atomic

{0,0} - , , , {2,3}:

, ., {2,3}:

, (..) R, , , 1. ! ., R

, c():

```
c(4, 8, 15, 16, 23, 42)
```

[1] 4 8 15 16 23 42

```
c(" ", " ", " ")
```

[1] " " " " "

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

[1] TRUE FALSE

c . ? . R . :
(3, 4, 5)

```
## Error in (3, 4, 5): " "
```

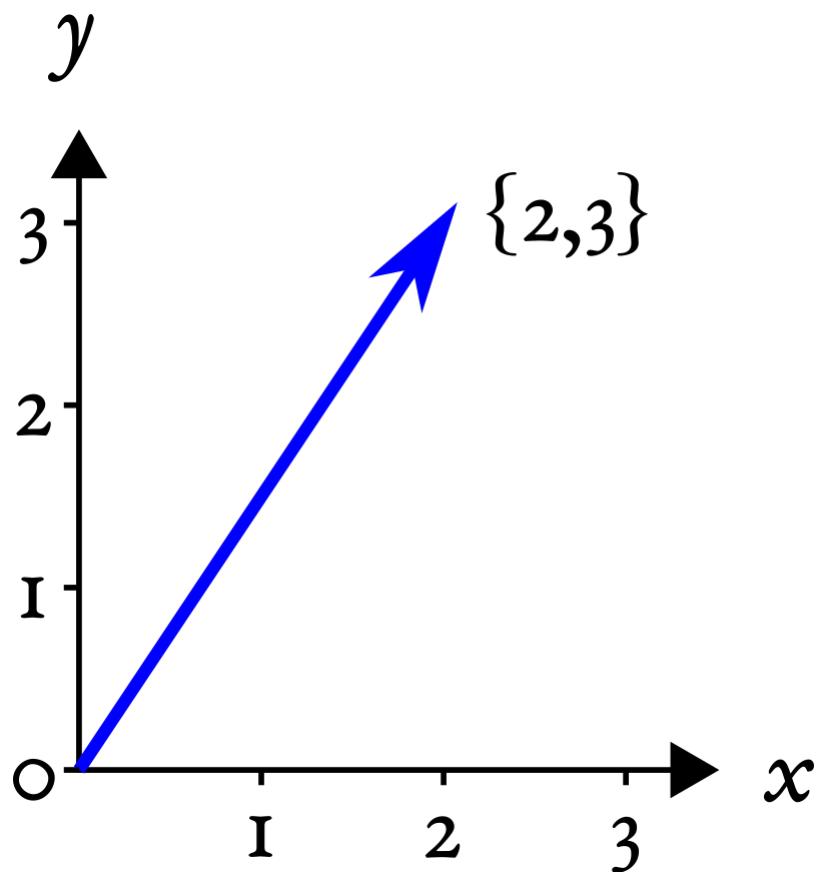


Figure 3.1:

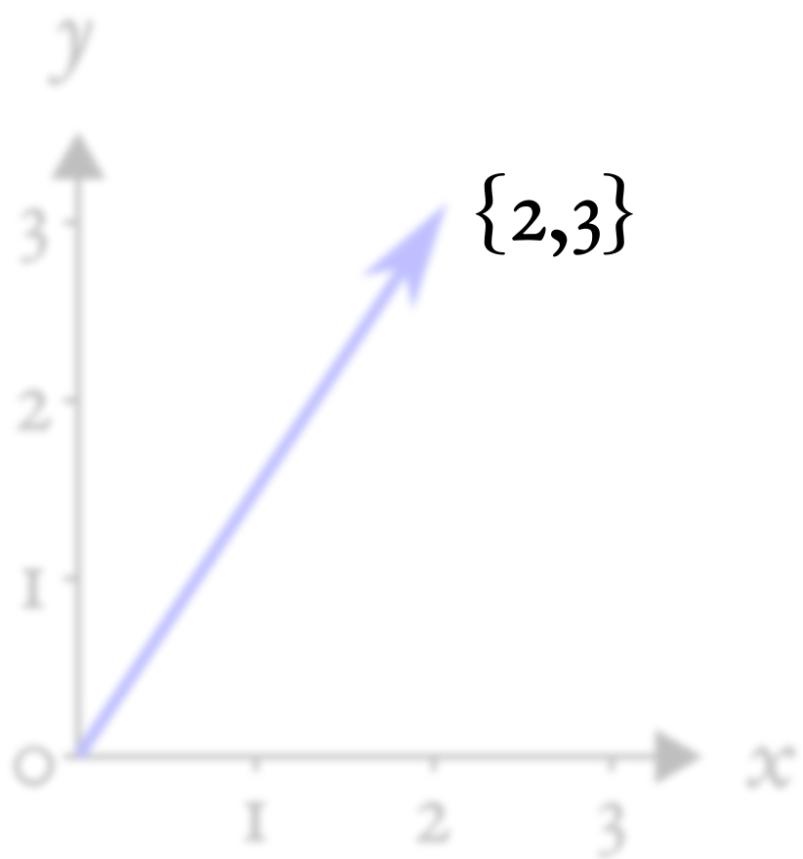


Figure 3.2:

```

:
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
1. , seq():
...
, , seq():

seq(10,100, by = 10)

## [1] 10 20 30 40 50 60 70 80 90 100
, , seq():
seq(1,13, length.out = 4)

## [1] 1 5 9 13
— rep() —
, —
rep(1, 5)

## [1] 1 1 1 1 1
, !
rep(1:3, 3)

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

## [1] 1 2 2 3 3 3
( , , , , 1):
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

3.2

, ? ?
, (atomic )
" ", " "
" ", ( —
). : . R " "
c(FALSE, 2)
## [1] 0 2

FALSE 0 ( TRUE 1),
: . :
2 + TRUE
## [1] 3

(c(TRUE, 3, " ")
(imlicit coercion).
: . :
c(TRUE, 3, " ")
## [1] "TRUE" "3" " "
NULL < raw < logical < integer < double < complex < character <
list < expression.

0 1 , 0 1 , "0" "1". — TRUE FALSE —
as.
(as.numeric(c(TRUE, FALSE, FALSE)))
## [1] 1 0 0
as.character(as.numeric(c(TRUE, FALSE, FALSE)))
```

```

## [1] "1" "0" "0"
,
NA —
as.numeric(c("1", "2", " "))
## Warning: NA
## [1] 1 2 NA
sum() mean()
! TRUE

```

3.3

```

,
,
:
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 7.2.      !
,
for while 8.5.1.

```

3.3.1

```

,           -           ?           ,
(           .           ,           !           ,
,           ,           ,
:
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

```

3.3.2

```

,          ,
R-      !
R —   [] (
— ( , ) — .
— , ..
n <- c(0, 1, 1, 2, 3, 5, 8, 13, 21, 34)
n[1]

## [1] 0
n[10]

## [1] 34
( MATLAB,
0 — ( ,
R           . — 1,
— length().

, : :
n[3] <- 20
n

## [1] 0 1 20 2 3 5 8 13 21 34
, : :
n[4:7]

## [1] 2 3 5 8
n[10:1]

## [1] 34 21 13 8 5 3 2 20 1 0
n[4:6] <- 0
n

## [1] 0 1 20 0 0 0 8 13 21 34
: :

```

```
n[-1]
## [1] 1 20 0 0 0 8 13 21 34
n[c(-4, -5)]
## [1] 0 1 20 0 8 13 21 34
"           ,
,
n[c(TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE, TRUE, FALSE)]
## [1] 0 20 0 8 21
:
TRUE,
FALSE.
```

Python.

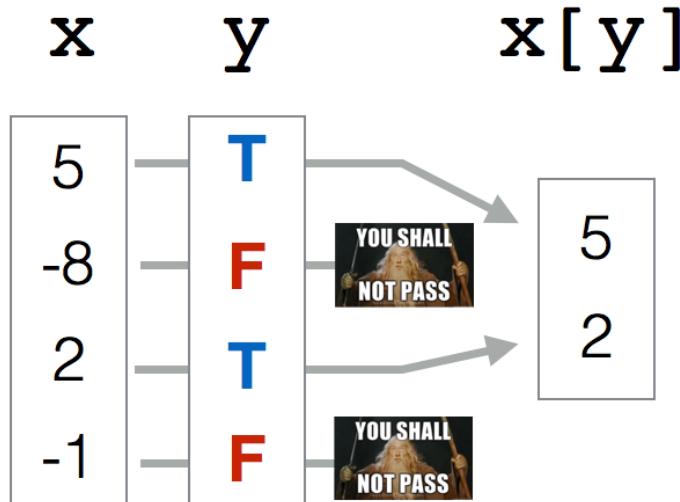


Figure 3.3:

```
(      )
!
n[c(TRUE, FALSE)] #      - recycling rule!
## [1] 0 20 0 8 21
(      ) , : ,
```

```

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]
names(d)

## [1] "a" "b" "c" "d"
d["a"]

## a
## 1

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

n mean():

mean(n)

## [1] 9.7

, ?
— :
larger <- n>mean(n)
larger

## [1] FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE
n:
n[larger]

## [1] 20 13 21 34

:
n[n>mean(n)]

## [1] 20 13 21 34
, R: (subset)

```

3.4

```
eyes <- c("green", "blue", "blue", "brown", "green", "blue")
```

3.4.1 mean() sum() TRUE

```
sum()                                                          TRUE
      "blue"                                                  eyes:
```

```
eyes == "blue"
```

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

```
sum(eyes == "blue")
```

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

```
mean()                                                  TRUE
```

```
eyes == "blue"
```

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

```
mean(eyes == "blue")
```

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

```
100,                                                  :
```

```
mean(eyes == "blue") * 100
```

```
## [1] 50
```

3.4.2 all() any()

```
all()          TRUE                                  TRUE:
```

```
all(eyes == "blue")
```

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

```
any()          TRUE                                  TRUE:
```

```
any(eyes == "blue")
```

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

```
!
```

```
.                                                  ,
```

```
FALSE         ?
```

```

any(!eyes == "blue")

## [1] TRUE
!all(eyes == "blue")

## [1] TRUE
FALSE?

all(!eyes == "blue")

## [1] FALSE
!any(eyes == "blue")

## [1] FALSE

```

3.4.3 : which()

```

, , ,
which()

which(eyes == "blue")

## [1] 2 3 6

```

3.4.4 %in% match()

```

: ,
: ,
eyes[eyes == c("green", "blue")]

## [1] "green" "blue"  "green" "blue"
: , !,
: , , !
eyes[eyes == "green" | eyes == "blue"]

## [1] "green" "blue"  "blue"  "green" "blue"

```

<code>eyes</code>	<code>==</code>	<code>c("green", "blue")</code>	\rightarrow	
"green"	<code>==</code>	<code>("green")</code>		TRUE
"blue"	<code>==</code>	<code>("blue")</code>		TRUE
"blue"	<code>==</code>	<code>("green")</code>	\searrow	FALSE
"brown"	<code>==</code>	<code>("blue")</code>	\searrow	FALSE
"green"	<code>==</code>	<code>("green")</code>	\searrow	TRUE
"blue"	<code>==</code>	<code>("blue")</code>	\searrow	TRUE

Figure 3.4:

```

,           2.           %in%,
,           ;           ,
,           .           ,
eyes[eyes %in% c("green", "blue")]
## [1] "green" "blue"  "blue"  "green" "blue"
           .           %in%           ,
           match()           %in%           ,
           FALSE      match()      NA (           TRUE.           ,
           nomatch =).
match(eyes, c("green", "blue"))

## [1] 1 2 2 NA 1 2
           ?   -   ,           (           NA.
           ),  -   ,           NA.
c("green", "blue")[match(eyes, c("green", "blue"))]
## [1] "green" "blue"  "blue"  NA       "green" "blue"

```

3.5 NA -

- - . - , - R NA (

```

Not Available -      ). NA —      "NA", 0,      ""
FALSE. NA —  NA. ,      NA      NA:
missed <- NA
missed == "NA"

## [1] NA
missed == ""

## [1] NA
missed == NA

## [1] NA

,      NA c NA  NA. : ,      NA,      NA
— ,      ! : NA — ,      ,      ,
.      ( ..      NA), ,      .

NA : 

n[5] <- NA
n

## [1] 0 1 20 0 NA 0 8 13 21 34
mean(n)

## [1] NA

,      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 FALSE TRUE FALSE FALSE FALSE FALSE
is.na(n) FALSE ,      TRUE ,      NA.
n      NA ,      : TRUE,      NA, FALSE,
NA.      ! ( ),

:
n[!is.na(n)]

## [1] 0 1 20 0 0 8 13 21 34
!
```

```

mean(n[!is.na(n)])
## [1] 10.77778
,          (!)
,           ,      NA.               mean():

?mean()
na.rm =,      FALSE.,      !
mean(n, na.rm = T)
## [1] 10.77778
NA          .,      NA -,      NA_integer_,
,           NA_real_, NA_complex_, NA_character_, R
NA.

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

```

3.6

R,
 ,
 R.
 “ ”,
 data.frame.

,
 - ,
 : ,
 (list) (matrix).
 .
 ,
 — .
 R —

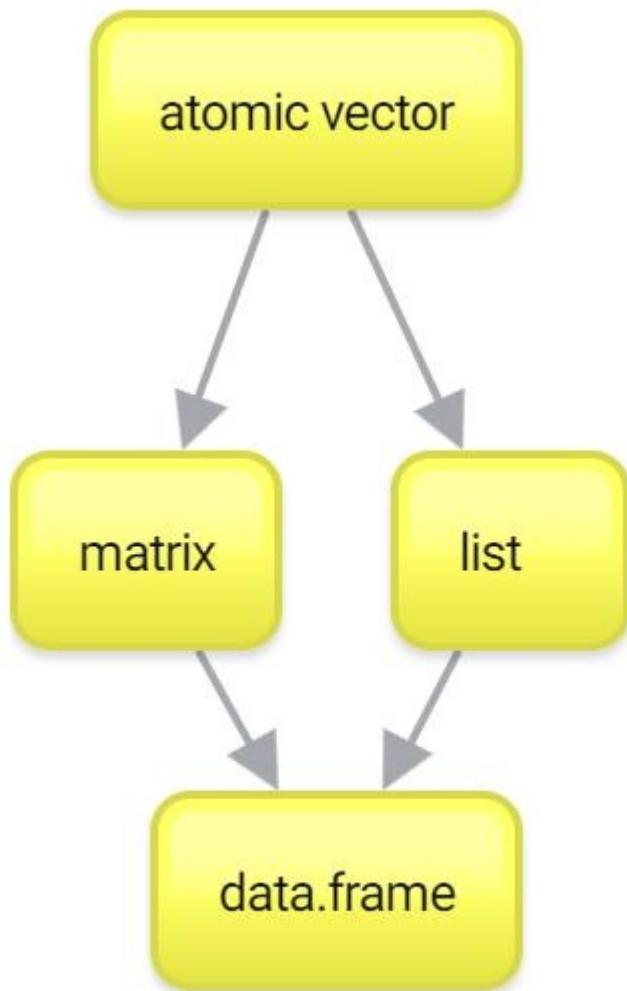


Figure 3.5:

Chapter 4

R

4.1

```
,           .      (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  
  
,           :           ,  
,           ,           ,           ,           :  
.           .           ,           R.  
,           ,           :  
  
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
          — , — , :
          /           ,
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 , , ,
, — , , : R —
, “ ”. dim ( ) dimnames. —
, . .
dim .
. 99-101 “R in a Nutshell” (Adler, 2010).

```

4.2

— !
 (array). : , , ,

```
array_3d <- array(1:12, c(3, 2, 2))
array_3d
```

```

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

```

4.3

(list)!

```
simple_list <- list(42, " ", TRUE)
simple_list

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

```

##  

## [[2]]  

## [1] "  

##  

## [[3]]  

## [1] TRUE  

,  

,  

!  

complex_list <- list(c("Wow", "this", "list", "is", "so", "big"), "16", simple_list)  

complex_list  

## [[1]]  

## [1] "Wow"  "this" "list" "is"   "so"   "big"  

##  

## [[2]]  

## [1] "16"  

##  

## [[3]]  

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

## [1] 42  

##  

## [[3]][[2]]  

## [1] "  

##  

## [[3]][[3]]  

## [1] TRUE  

,  

,  

,  

,  

str():  

str(complex_list)  

## List of 3  

## $ : chr [1:6] "Wow" "this" "list" "is" ...  

## $ : chr "16"  

## $ :List of 3  

## ..$ : num 42  

## ..$ : chr "  

## ..$ : logi TRUE  

,  

-  

:  

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

named_list  

## $age  

## [1] 24

```

```

##  

## $PhDstudent  

## [1] TRUE  

##  

## $language  

## [1] "Russian"  

,  

named_list$age  

## [1] 24  

,  

named_list[1]  

## $age  

## [1] 24  

,  

,—  

, . . . :  

class(named_list)  

## [1] "list"  

class(named_list[1])  

## [1] "list"  

( ,  

:  

named_list[[1]]  

## [1] 24  

class(named_list[[1]])  

## [1] "numeric"  

Indexing lists in #rstats. Inspired by the Residence Inn pic.twitter.com/YQ6axb2w7t  

— Hadley Wickham (?) September 14, 2015  

,  

named_list[['age']]  

## [1] 24  

,—  

,  

§.  

R, , Python. — R,  

,
```

4.4

```
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 : chr "Ivan" "Eugeny" "Lena" "Misha" ...
## $ age  : num  26 34 23 27 26
## $ student: logi FALSE FALSE TRUE TRUE TRUE

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           .           ,
...1           .           ,
,

```

¹ , , , ,

The screenshot shows a table with the following data:

	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 4.1:

Chapter 5

R

5.1

R —
(packages).

— beep(), beep_on_error()
,
— stringi, stringr
().

R,
data.table tidyverse.
R,
, tidyverse - ,
tidyverse.
— mlr3 , “ ”,

5.2 R

```
,           R :      base ,      stats,
utils, graphics.          :
rownames(installed.packages(priority = "base"))

## [1] "base"      "compiler"   "datasets"   "graphics"   "grDevices"  "grid"
## [7] "methods"   "parallel"   "splines"    "stats"     "stats4"    "tcltk"
## [13] "tools"     "utils"
```

5.3

CRAN

```
install.packages()
work (CRAN).          CRAN       Comprehensive R Archive Net-
CRAN:                 16000
,                      ,
remotes.               CRAN
install.packages("remotes")
```

!()

5.4

```
install.packages()
library().
library("remotes")

install.packages(),   library()
,
library(remote)
```

5.5

::

```

  .   ,   ::,   .
  (   ::),   .
    package_deps()           remotes,
  :
remotes::package_deps("tidyverse")

  ::   ,   ,
  ::   tidyverse, dplyr,      filter().
R     stats,               .
  ::   ,   filter()        dplyr,
stats::filter(1:20, rep(1,3))      stats:

## Time Series:
## Start = 1
## End = 20
## Frequency = 1
## [1] NA  6  9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 NA

  ,
  ,   ,   ,
  ,   ,   ,   ::.

  detach().

detach(package::remotes)

```

5.6**c Bioconductor**

, CRAN, — Bioconductor.

Bioconductor

BiocManager CRAN.

```
install.packages("BiocManager")
```

install()

BiocManager

flowCore

```
—
BiocManager::install("flowCore")
```

5.7 Github

```

CRAN,      Bioconductor.          ,
           CRAN.          ,
CRAN (      ,           )          CRAN
,
           Github.

( CRAN,      )      remotes1.
remotes::install_github("dracor-org/rdracor")

,
library(rdracor)
godunov <- play_igraph(corpus = "rus",
                       play = "pushkin-boris-godunov")
plot(godunov)

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <91>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <be>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d1>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <80>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

```

¹ remotes “ ” devtools, remote devtools.
devtools/remotes remotes,
devtools::install_github().

```
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <b8>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <d1>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <81>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0411
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+043e
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0440
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0438
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0441
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <d0>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <9f>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <d0>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <b8>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <d0>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
## <bc>
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 ' ' 'mbcsToSbcs':
```

```
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <b5>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <bd>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 U+041f
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 U+0438
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 U+043c
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 U+0435
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 U+043d
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <93>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d1>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <80>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '      'mbcsToSbcs':
## <b8>
```

```
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <b3>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <be>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <d1>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <80>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <b8>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <d0>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                 '        'mbcsToSbcs':
## <b9>

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0413

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0440

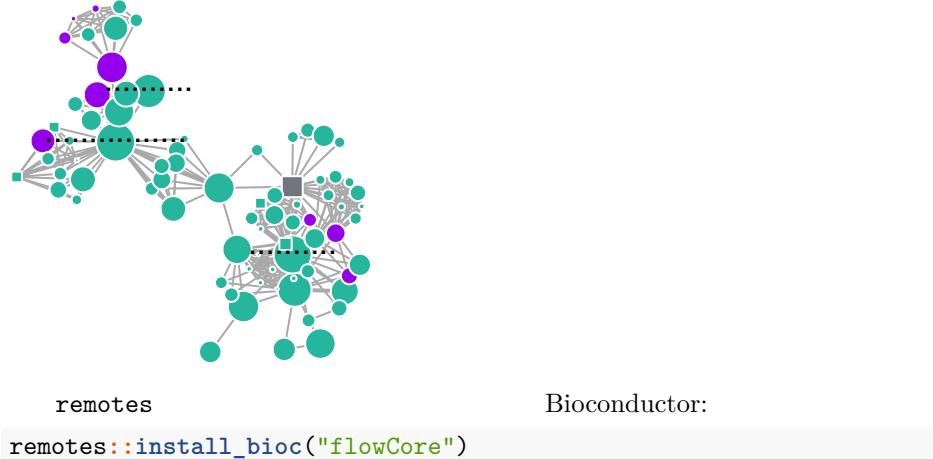
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0438

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :                           U+0433
```

```

## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :
## Warning in text.default(x, y, labels = labels, col = label.color, family =
## label.family, :

```



5.8

, . ?
 , . , R , ,
 ! , , ,
 - . CRAN (Task View)
 , :
<https://cran.r-project.org/web/views/>
 , - , , Task View, Task View,
 — — , , , : R
 . , , , ,

Chapter 6

```
, ) . . , , ( - 1, , , ,  
— . . , , ( : , ,  
, , ).
```

6.1 RStudio

```
, , “ ” ( ,  
read.csv() — ), ,  
:  
read.csv("heroes_information.csv")  
  
## Warning in file(file, "rt"): 'heroes_information.csv': No  
## such file or directory  
## Error in file(file, "rt"):  
 , R . . , .  
, R . .  
• :  
, getwd() ( ),  
:  
—————  
1 , DC Marvel, DC Marvel.
```

```
heroes <- read.csv("heroes_information.csv")
, RStudio , (
"Console"):
```

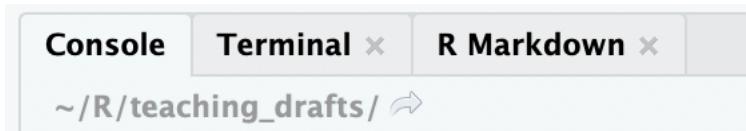


Figure 6.1:

```
• : . .
      setwd() , ,
      :
heroes <- read.csv("heroes_information.csv")
!
,
• : .
heroes <- read.csv("/Users/Username/Some_Folder/heroes_information.csv")
,
,
Windows : / R,
//.
• : Import Dataset.
Environment RStudio "Import Dataset" ,
,
,
R, ,
:
heroes <- read.csv("https://raw.githubusercontent.com/agricolamz/2020-2021-ds4dh/master"
• : RStudio.
,
,
,
,
,
,
data
```

Tool - Global Options...

6.1.1 :

R.

macOS) : “ ” (, TextEdit

Values” (,). .csv — .CSV Microsoft Excel “ ” : (.). .CSV Microsoft Excel !

.csv, ;, (,) .read.csv() .read.csv2() — () “ ”. .CSV, , .read.csv2()

.CSV — .tsv — , .CSV, .read.delim() .read.delim2(). .read.table(). , “ ” —

```

            ,           file =,
stringsAsFactors =      FALSE:
heroes <- read.csv("data/heroes_information.csv", stringsAsFactors = FALSE)

stringsAsFactors =      ,
                    ,           character,
                    ,           character,
factor,           .
.
.
R
4.0,   stringsAsFactors =   FALSE
.
View(heroes):      !
(read.table(), read.delim())
Help.

```

6.2

```

(          ),
,
str():
str(heroes)

## 'data.frame':    734 obs. of  11 variables:
## $ X          : int  0 1 2 3 4 5 6 7 8 9 ...
## $ name        : chr  "A-Bomb" "Abe Sapien" "Abin Sur" "Abomination" ...
## $ Gender      : chr  "Male" "Male" "Male" "Male" ...
## $ Eye.color   : chr  "yellow" "blue" "blue" "green" ...
## $ Race        : chr  "Human" "Ichtyo Sapien" "Ungaran" "Human / Radiation" ...
## $ Hair.color: chr  "No Hair" "No Hair" "No Hair" "No Hair" ...
## $ Height      : num  203 191 185 203 -99 193 -99 185 173 178 ...
## $ Publisher   : chr  "Marvel Comics" "Dark Horse Comics" "DC Comics" "Marvel Comics"
## $ Skin.color: chr  "--" "blue" "red" "--" ...
## $ Alignment   : chr  "good" "good" "good" "bad" ...
## $ Weight      : int  441 65 90 441 -99 122 -99 88 61 81 ...
,
,
?

1.           NA.
"NA",
na.strings =   read.table()
.
```

```

, , , NA.
,
2. , , int num. ,
chr ("character") Factor ( stringsAsFactors =
TRUE), , , NA.
3. . .
, , ( header =
FALSE)
4. . .
"UTF-8" encoding =
heroes <- read.csv("data/heroes_information.csv",
stringsAsFactors = FALSE,
encoding = "UTF-8")
, , .
5. . , , —
sep =.
6. \ ", read.csv(), read.delim(), read.csv2(),
read.delim2()
, ;,
7. . (read.table(), read.csv(), read.delim()), , (read.csv2(),
read.delim2()).
read.csv(),
, . .
-99. , , ( na.strings =
read.csv()):
heroes <- read.csv("data/heroes_information.csv",
stringsAsFactors = FALSE,
na.strings = c("-", "-99"))

```

6.3

, DC .csv.

```
dc <- heroes[heroes$Publisher == "DC Comics",]

write.csv() .csv:
write.csv(dc, "data/dc_heroes_information.csv")

, , row.names = FALSE:
write.csv(dc, "data/dc_heroes_information.csv", row.names = FALSE)

read.csv2(), write.csv2() .csv ;.
write.csv2(dc, "data/dc_heroes_information.csv", row.names = FALSE)
```

6.4 : Excel, SPSS

- ,
- Microsoft Excel. .xlsx ,
- ,
- .csv.
- R. ,
- :
- Microsoft Excel: `readxl (` tidyverse), `(xlsx, openxlsx)`.
 - SPSS, SAS, Stata: — `haven (` tidyverse) `foreign`.

6.5

```
, read.csv(),
,
• readr ( tidyverse)
read.csv(), read.csv2() ,
read_csv() read_csv2(). ,
tidyverse.

readr::read_csv("data/heroes_information.csv",
na = c("-", "-99"))

## Warning: Missing column names filled in: 'X1' [1]
```

```

## 
## -- Column specification -----
## cols(
##   X1 = col_double(),
##   name = col_character(),
##   Gender = col_character(),
##   `Eye color` = col_character(),
##   Race = col_character(),
##   `Hair color` = col_character(),
##   Height = col_double(),
##   Publisher = col_character(),
##   `Skin color` = col_character(),
##   Alignment = col_character(),
##   Weight = col_double()
## )

## # A tibble: 734 x 11
##       X1     name   Gender `Eye color` `Race` `Hair color` `Height` Publisher
##   <dbl>    <chr>   <chr>      <chr>    <chr>      <chr>    <dbl>   <chr>
## 1     0 A-Bo~ Male    yellow   Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male    blue    Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male    blue    Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male    green   Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male    blue    Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male    blue    Human No Hair      193 Marvel C~
## 7     6 Adam~ Male    blue    <NA> Blond        NA NBC - He~
## 8     7 Adam~ Male    blue    Human Blond       185 DC Comics
## 9     8 Agen~ Female  blue    <NA> Blond        173 Marvel C~
## 10    9 Agen~ Male    brown   Human Brown      178 Marvel C~

## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

• vroom -          tidyverse.           readr      tidyverse,
      (             ).

vroom::vroom("data/heroes_information.csv")

## New names:
## * `` -> ...1

## Rows: 734
## Columns: 11
## Delimiter: ","
## chr [8]: name, Gender, Eye color, Race, Hair color, Publisher, Skin color, Alignment
## dbl [3]: ...1, Height, Weight
##
## Use `spec()` to retrieve the guessed column specification
## Pass a specification to the `col_types` argument to quiet this message

```

```

## # A tibble: 734 x 11
##   ...1 name  Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr>     <chr> <chr>      <dbl> <chr>
## 1     0 A-Bo~ Male  yellow  Human No Hair       203 Marvel C~
## 2     1 Abe ~ Male  blue   Ichth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male  blue   Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male  green  Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male  blue   Cosm~ Black        -99 Marvel C~
## 6     5 Abso~ Male  blue   Human No Hair      193 Marvel C~
## 7     6 Adam~ Male  blue   -     Blond        -99 NBC - He-
## 8     7 Adam~ Male  blue   Human Blond       185 DC Comics
## 9     8 Agen~ Female blue   -     Blond        173 Marvel C~
## 10    9 Agen~ Male   brown  Human Brown       178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

•   data.table - , R,
tidyverse. data.table - . data.table
, . : fread()
fwrite(), f fast2.
data.table::fread("data/heroes_information.csv")

##      V1          name Gender Eye color           Race Hair color
## 1:  0       A-Bomb   Male  yellow      Human  No Hair
## 2:  1      Abe Sapien   Male   blue  Ichthyo Sapien  No Hair
## 3:  2      Abin Sur   Male   blue  Ungaran  No Hair
## 4:  3 Abomination   Male  green Human / Radiation  No Hair
## 5:  4      Abraxas   Male   blue Cosmic Entity  Black
## 6:  --- 
## 730: 729 Yellowjacket II Female   blue      Human Strawberry Blond
## 731: 730          Ymir   Male  white   Frost Giant  No Hair
## 732: 731          Yoda   Male  brown  Yoda's species  White
## 733: 732         Zatanna Female  blue      Human  Black
## 734: 733          Zoom   Male   red      -        Brown
## 735: 735          Height Publisher Skin color Alignment Weight
## 1: 203.0  Marvel Comics   -    good    441
## 2: 191.0 Dark Horse Comics  blue  good     65
## 3: 185.0  DC Comics     red  good     90
## 4: 203.0  Marvel Comics   -    bad    441
## 5: -99.0  Marvel Comics   -    bad    -99
## 6:  --- 
## 730: 165.0  Marvel Comics   -    good     52
## 731: 304.8  Marvel Comics  white  good    -99

```

² friendly: fread()

. vroom .

```
## 732: 66.0      George Lucas      green      good      17
## 733: 170.0      DC Comics       -        good      57
## 734: 185.0      DC Comics       -        bad       81
?      3      ,      vroom  data.table.

,      R)  readr::read_csv() (
tidyverse)  .
,      (      !)      :
readr::write_csv(dc, "data/dc_heroes_information.csv")
readr::write_excel_csv(dc, "data/dc_heroes_information.csv") #      Excel
vroom::vroom_write(dc, "data/dc_heroes_information.csv", delim = ",")
data.table::fwrite(dc, "data/dc_heroes_information.csv")

,      readr,      -      : vroom  data.table
,      R.
```


Chapter 7

7.1 if, else, else if

```
— R . . .
c : . . .

if ( . . )
:
number <- 1
if (number > 0) " "
## [1] "
(expression)
,
number <- 1
if (number > 0) {
  "
}
## [1] "
,
1,
2,
else :
if ( . . ) else
——— , ,
1 , TRUE.
2 , FALSE.
```

```

:
number <- -3
if (number > 0) {
  " "
} else {
  " "
}

## [1] ""

               :
else if.

:
number <- 0
if (number > 0) {
  " "
} else if (number < 0){
  " "
} else {
  " "
}

## [1] ""

               , R — , — , ,
R!
number <- -2:2
if (number > 0) {
  " "
} else if (number < 0){
  " "
} else {
  " "
}

## Warning in if (number > 0) {:
##          > 1,
## Warning in if (number < 0) {:
##          > 1,
##

## [1] ""

               R , ?

```

7.2 for

```
- , for. for
for( in )
      for. , :
for (i in number) {
  if (i > 0) {
    print(" ")
  } else if (i < 0) {
    print(" ")
  } else {
    print(" ")
  }
}

## [1] "
## [1] "
## [1] "
## [1] "
## [1] "
for, print().
, for R . for
R , ,
, ,
, ,
, ,
, ,
, ,
, ,
, ,
, cumsum()
cumsum(1:10)

## [1] 1 3 6 10 15 21 28 36 45 55
, apply() ( .
@ref(apply_f) ).

- for , ,
, for? , , , ,
for , , , ,
```

```

for - ( , , )
,
number_descriptions <- character(length(number)) #
for (i in 1:length(number)) {
  if (number[i] > 0) {
    number_descriptions[i] <- " "
  } else if (number[i] < 0) {
    number_descriptions[i] <- " "
  } else {
    number_descriptions[i] <- " "
  }
}
number_descriptions

## [1] " " " "
## [4] " " " "
,
for
,
for , , , ,
,
tidyverse ( . [pipe]).
```

7.3 : **ifelse()**

dplyr::case_when()

```

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

ifelse(number > 0, " ", " ")

## [1] " "
## [3] " "
## [5] " "

: ifelse( , TRUE, FALSE).
TRUE FALSE
TRUE FALSE
,
!
ifelse() : else if.
ifelse() ifelse():

ifelse(number > 0,
" ",
",
```

```
ifelse(number < 0, "", "")  
## [1] ""  
## [4] "  
dplyr ( tidyverse) — case_when(),  
:  
dplyr::case_when(  
  number > 0 ~ "",  
  number < 0 ~ "",  
  number == 0 ~ "")  
## [1] ""  
## [4] "
```


Chapter 8

R

8.1

```
,           R.           ,           ,
,           !           .           ,
,           .           .           ,
function,
,           ,           ,
,           "           "
,           ,
,           ,
,           ,
return().   return()
,           ,
return(),
pow <- function(x, p) {
  power <- x ^ p
  return(power)
}
pow(3, 2)

## [1] 9

,           return()
,           ,
pow <- function(x, p) {
  x ^ p
}
pow(3, 2)

## [1] 9
```

```
, , , , , :  
 , , , , , !  
  
pow <- function(x, p) {  
  power <- x ^ p #  
}  
pow(3, 2) #  
  
,  
pow <- function(x, p) x ^ p  
pow(3, 2)  
  
## [1] 9  
  
, , , , ,  
  
pow <- function(x, p = 2) x ^ p  
pow(3)  
  
## [1] 9  
pow(3, 3)  
  
## [1] 27  
R (lazy evaluations). , ,  
we_will_not_use_this_parameter =,  
pow <- function(x, p = 2, we_will_not_use_this_parameter) x ^ p  
pow(x = 3)  
  
## [1] 9
```

8.2

(sanity check).

```

imt()
weight =)           (      height =
imt <- function(weight, height) weight / height ^ 2

,
:
w <- c(60, 80, 120)
h <- c(1.6, 1.7, 1.8)
imt(weight = w, height = h)

## [1] 23.43750 27.68166 37.03704

,
,
,
,
3.           ,
warning()

imt <- function(weight, height) {
  if (height > 3) warning("           height      3:      ,           ,
  weight / height ^ 2
}
imt(78, 167)

## Warning in imt(78, 167):           height      3:      ,
## [1] 0.002796802

,
,
,
,
imt()           0.           ,
,
imt <- function(weight, height) {
  if (any(weight <= 0 | height <= 0)) stop("           height      3:      ,
  if (height > 3) warning("           height      3:      ,           ,
  weight / height ^ 2
}
imt(-78, 167)

## Error in imt(-78, 167):
,
```

R.

R ,

8.3 ?

? “ ” — , . , , , ,
 : , , , , , , ,
 , , , (, - , , ,
 R), , . , , , ,
 — R.
 , — , , , , ,

8.4

, function — — . function(). , { — ! R —
 — R. , , , ,
 , R (, , ..). , ,
 :
list(mean, min, `{`)
 ## [[1]]
 ## function (x, ...)
 ## UseMethod("mean")
 ## <bytecode: 0x7fa187119808>
 ## <environment: namespace:base>
 ##
 ## [[2]]
 ## function (... , na.rm = FALSE) .Primitive("min")
 ##
 ## [[3]]
 ## .Primitive("{")
 ! — , !
 1 , ,

 1 , , ,

```
, , - ,  
apply() (@ref(apply_f)) tidyverse.
```

Python :
(,)

8.5 **apply()**

8.5.1 **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() ( ).  
apply() ,  
, ,  
,  
, “ ” ( , ) . apply()  
FUN, ...), X — , MARGIN 1 ( ), 2 ( ), c(1,2) : apply(X, MARGIN,  
( .. ), FUN — , ! apply()  
/ X  
,  
,  
:  
apply(A, 1, sum) #  
## [1] 22 26 30  
apply(A, 2, sum) #  
## [1] 6 15 24 33
```

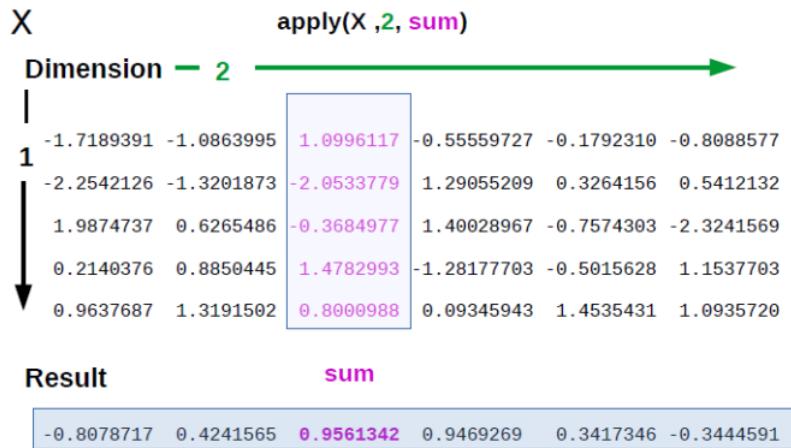


Figure 8.1: apply

```

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

R           colSums(),
rowSums(), colMeans()  rowMeans(),
apply()

,
apply(A, 1, sum, na.rm = TRUE)

## [1] 22 26 30
apply(A, 1, weighted.mean, w = c(0.2, 0.4, 0.3, 0.1))

## [1] 4.9 5.9 6.9

```

8.5.2

, - , ?
, - , ?
- , . , ?
- , . , ?

```

,           :
apply(A, 1, function(x) x - mean(x)) #

##      [,1] [,2] [,3]
## [1,] -4.5 -4.5 -4.5
## [2,] -1.5 -1.5 -1.5
## [3,]  1.5  1.5  1.5
## [4,]  4.5  4.5  4.5

apply(A, 2, function(x) x - mean(x)) #

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

apply(A, c(1,2), function(x) x - mean(x)) , . .

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

,      x , , - , ,
,      . , , , , :
apply(A, 1, function(whatevname) whatevname - mean(whatevname))

##      [,1] [,2] [,3]
## [1,] -4.5 -4.5 -4.5
## [2,] -1.5 -1.5 -1.5
## [3,]  1.5  1.5  1.5
## [4,]  4.5  4.5  4.5

```

8.5.3 *apply()*

```

, apply() . ? ?
lapply( " " ) sapply() - lapply(),
" " .

some_list <- list(some = 1:10, list = letters)
lapply(some_list, length)

## $some
## [1] 10
##
## $list
## [1] 26

```

```

sapply(some_list, length)

## some list
##   10   26

sapply(),
.

sapply(1:10, sqrt)

## [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
## [9] 3.000000 3.162278
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

sapply(),
.
,
sapply(),
for.

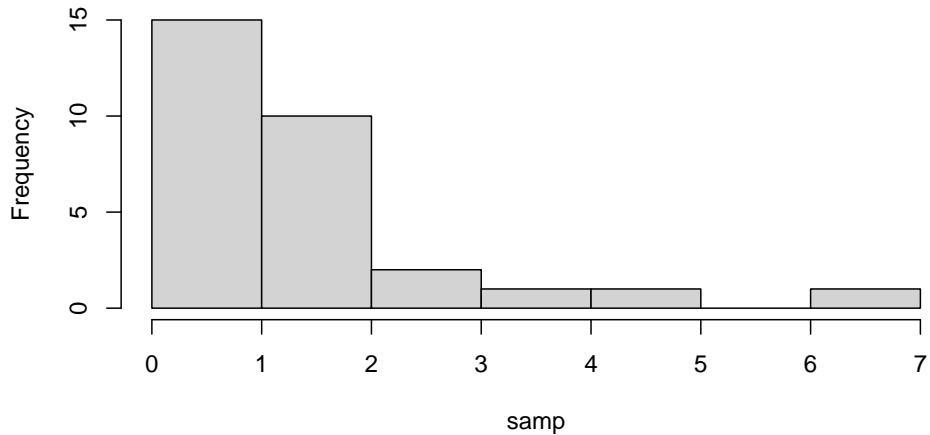
-
Vectorize().
apply().

lapply() sapply()
( . 4.4),
:
heroes <- read.csv("data/heroes_information.csv",
na.strings = c("-", "-99"))
sapply(heroes, class)

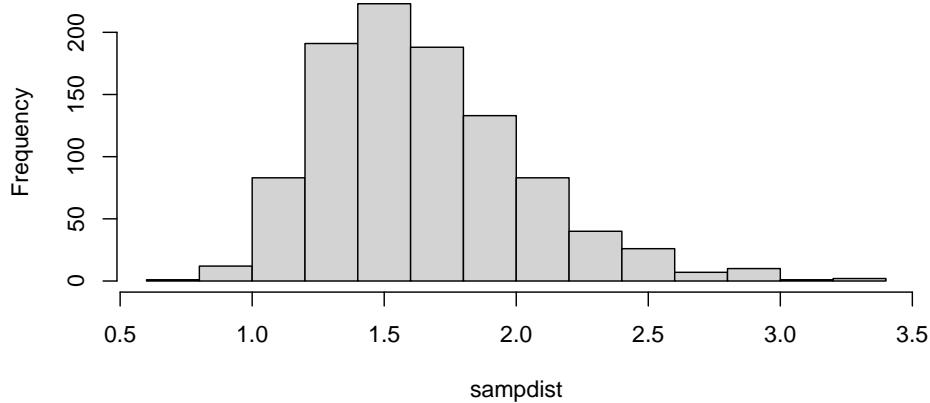
##           X      name    Gender   Eye.color      Race Hair.color
## "integer" "character" "character" "character" "character" "character"
##      Height Publisher Skin.color Alignment      Weight
## "numeric" "character" "character" "character" "integer"

apply() -      replicate() -
:
samp <- rlnorm(30)
hist(samp)

```

Histogram of samp

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

Histogram of sampdist

```
apply(),  
,  
apply() —  
purrr., tidyverse  
purrr., apply(),  
purrr.
```


Chapter 9

tidyverse

9.1 tidyverse

tidyverse - , . . . (), -

tidyverse — :
• *ggplot2*,
• *tibble*,
• *tidyr*, tidy data
• *readr*, R
• *purrr*, (*apply())
• *dplyr*,
• *stringr*,
• *forcats*, -
• *vroom*,
• *readxl*, .xls .xlsx
• *jsonlite*, JSON
• *xml*, XML
• *DBI*,
• *rvest*, -
• *lubridate*,
• *tidytext*,
• *glue*,
• *magrittr*, pipe
• *tidymodels*,

¹ tidyverse, tidymodels —

```

• dplyr,          dplyr           data.table
    tidyverse!      ,               tidyverse.
    tidyverse       ,               tidyverse.
    tidyverse       ,               tidyverse.
    tidyverse       ,               tidyverse.

install.packages("tidyverse")

tidyverse —
library("tidyverse")

## -- Attaching packages ----- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble   3.0.4     v dplyr    1.0.2
## v tidyrr   1.1.2     v stringr  1.4.0
## v readr    1.4.0     vforcats  0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

tidyverse           tidyverse,
.
.
.

```

9.2 readr

```

          .csv      R      read.csv(),
read_csv()      readr.    read_csv()      read.csv():
                  (        URL),
heroes <- read_csv("data/heroes_information.csv",
                    na = c("-", "-99"))

## Warning: Missing column names filled in: 'X1' [1]

##
## -- Column specification -----
## cols(
##   X1 = col_double(),
##   name = col_character(),
##   Gender = col_character(),
##   `Eye color` = col_character(),
##   Race = col_character(),

```

```

## `Hair color` = col_character(),
## Height = col_double(),
## Publisher = col_character(),
## `Skin color` = col_character(),
## Alignment = col_character(),
## Weight = col_double()
## )
,
tidyverse,      @ref(real_data).

```

9.3 tibble

```

read_csv(),          tibble,    data.frame:
class(heroes)

## [1] "spec_tbl_df" "tbl_df"       "tbl"        "data.frame"
(tibble) -      " "           "data.frame.",      "data.frame",
.
.
.

heroes

## # A tibble: 734 x 11
##       X1 name Gender `Eye color` `Race` `Hair color` Height Publisher
##   <dbl> <chr> <chr>   <chr>   <chr>   <dbl> <chr>
## 1     0 A-Bo~ Male   yellow Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male   blue   Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male   blue   Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male   green  Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male   blue   Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male   blue   Human No Hair     193 Marvel C~
## 7     6 Adam~ Male   blue   <NA>  Blond        NA NBC - He~
## 8     7 Adam~ Male   blue   Human Blond       185 DC Comics
## 9     8 Agen~ Female blue   <NA>  Blond       173 Marvel C~
## 10    9 Agen~ Male   brown  Human Brown      178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>
10 ,      ,
.
.

tidyverse
,
.

heroes_df <- as.data.frame(heroes) #
class(heroes_df)

```

```

## [1] "data.frame"
as_tibble(heroes_df) #



## # A tibble: 734 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr>     <chr> <chr>      <dbl> <chr>
## 1     0 A-Bo~ Male  yellow Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male  blue  Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male  blue  Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male  green Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male  blue  Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male  blue  Human No Hair      193 Marvel C~
## 7     6 Adam~ Male  blue  <NA> Blond        NA NBC - He~
## 8     7 Adam~ Male  blue  Human Blond       185 DC Comics
## 9     8 Agen~ Female blue  <NA> Blond       173 Marvel C~
## 10    9 Agen~ Male  brown Human Brown       178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

tidyverse, , ,
tibble(), data.frame():

tibble(
  a = 1:3,
  b = letters[1:3]
)

## # A tibble: 3 x 2
##       a   b
##   <int> <chr>
## 1     1 a
## 2     2 b
## 3     3 c

```

9.4 magrittr::%>%

```
%>%      “ ” (pipe), . “ ”. , ( ), ( ). , , , , ,
```

```

magrittr2.      ,
tidyverse,3

,
.

...
sum(sqrt(abs(sin(1:22))))3

## [1] 16.72656

...
:
1:22 %>%
  sin() %>%
  abs() %>%
  sqrt() %>%
  sum()

## [1] 16.72656

(
).
%,           ,
%>%.

"       !" %>%
  c(" -- ", , " -- ")

## [1] " -- "           "       !" " -- "

```

9.5

tidyverse: dplyr tidyR

dplyr⁴ — tidyverse.

- dplyr dplyr, plyr,
 plyr, , , .
 tidyR dplyr, , , .
-
-
-

, dplyr , , tidyR
 . , , , , ,

² , %>% tidyverse, magrittr
³ , , , : , ,

⁴ , , , : , ,

```

  . , ( ), ,
  ::,
  ,
tidyr — reshape2,
  reshape.      plyr,
  ,
  dplyr  tidyr , . dplyr
tidyr , ,
  ,
  ,

```

9.6

9.6.1 : dplyr::select()

```

dplyr::select() ( ).  

heroes %>%
  select(1,5)

## # A tibble: 734 x 2
##       X1 Race
##   <dbl> <chr>
## 1     0 Human
## 2     1 Icthyo Sapien
## 3     2 Ungaran
## 4     3 Human / Radiation
## 5     4 Cosmic Entity
## 6     5 Human
## 7     6 <NA>
## 8     7 Human
## 9     8 <NA>
## 10    9 Human
## # ... with 724 more rows
heroes %>%
  select(name, Race, Publisher, `Hair color`)

## # A tibble: 734 x 4
##   name      Race      Publisher `Hair color`
##   <chr>     <chr>     <chr>        <chr>
## 1 A-Bomb   Human    Marvel Comics No Hair
## 2 Abe Sapien Icthyo Sapien Dark Horse Comics No Hair
## 3 Abin Sur Ungaran DC Comics    No Hair

```

```

## 4 Abomination Human / Radiation Marvel Comics No Hair
## 5 Abraxas Cosmic Entity Marvel Comics Black
## 6 Absorbing Man Human Marvel Comics No Hair
## 7 Adam Monroe <NA> NBC - Heroes Blond
## 8 Adam Strange Human DC Comics Blond
## 9 Agent 13 <NA> Marvel Comics Blond
## 10 Agent Bob Human Marvel Comics Brown
## # ... with 724 more rows

, , , , ,
(2.6). , ,
( , , ~).

, tidyverse / . , ,
, : 

heroes_some_cols <- heroes %>%
  select(name, Race, Publisher, `Hair color`)
heroes_some_cols

## # A tibble: 734 x 4
##   name      Race     Publisher `Hair color`
##   <chr>     <chr>    <chr>      <chr>
## 1 A-Bomb   Human   Marvel Comics No Hair
## 2 Abe Sapien Icthyo Sapien Dark Horse Comics No Hair
## 3 Abin Sur Ungaran DC Comics No Hair
## 4 Abomination Human / Radiation Marvel Comics No Hair
## 5 Abraxas   Cosmic Entity Marvel Comics Black
## 6 Absorbing Man Human Marvel Comics No Hair
## 7 Adam Monroe <NA> NBC - Heroes Blond
## 8 Adam Strange Human DC Comics Blond
## 9 Agent 13   <NA> Marvel Comics Blond
## 10 Agent Bob Human Marvel Comics Brown
## # ... with 724 more rows

```

9.6.2 - tidyselect

```

(      select(),          tidyverse)
- tidyselect           5. tidyselect
:                   (
1).

heroes %>%
  select(name:Publisher)



---


5      magrittr,    tidyselect          tidyverse,
tidyverse.

```

```

## # A tibble: 734 x 7
##   name     Gender `Eye color` Race      `Hair color` Height Publisher
##   <chr>    <chr>   <chr>     <chr>    <chr>       <dbl> <chr>
## 1 A-Bomb   Male    yellow   Human    No Hair      203 Marvel Comics
## 2 Abe Sapien Male    blue    Icthyo Sapien No Hair    191 Dark Horse C-
## 3 Abin Sur  Male    blue    Ungaran  No Hair      185 DC Comics
## 4 Abominati~ Male    green   Human / Radi~ No Hair    203 Marvel Comics
## 5 Abraxas   Male    blue    Cosmic Entity Black NA Marvel Comics
## 6 Absorbing~ Male    blue   Human    No Hair      193 Marvel Comics
## 7 Adam Monr~ Male    blue   <NA>      Blond      NA NBC - Heroes
## 8 Adam Stra~ Male    blue   Human    Blond      185 DC Comics
## 9 Agent 13   Female  blue   <NA>      Blond      173 Marvel Comics
## 10 Agent Bob  Male   brown  Human    Brown      178 Marvel Comics
## # ... with 724 more rows

heroes %>%
  select(name:`Eye color`, Publisher:Weight)

## # A tibble: 734 x 7
##   name     Gender `Eye color` Publisher      `Skin color` Alignment Weight
##   <chr>    <chr>   <chr>     <chr>    <chr>       <chr>     <dbl>
## 1 A-Bomb   Male    yellow   Marvel Comics <NA>      good      441
## 2 Abe Sapien Male    blue    Dark Horse Com~ blue      good      65
## 3 Abin Sur  Male    blue    DC Comics     red      good      90
## 4 Abomination Male    green   Marvel Comics <NA>      bad      441
## 5 Abraxas   Male    blue    Marvel Comics <NA>      bad      NA
## 6 Absorbing M~ Male    blue   Marvel Comics <NA>      bad      122
## 7 Adam Monroe Male    blue   NBC - Heroes <NA>      good      NA
## 8 Adam Strange Male    blue   DC Comics     <NA>      good      88
## 9 Agent 13   Female  blue   Marvel Comics <NA>      good      61
## 10 Agent Bob  Male   brown  Marvel Comics <NA>      good      81
## # ... with 724 more rows

!

heroes %>%
  select(!X1)

## # A tibble: 734 x 10
##   name   Gender `Eye color` Race  `Hair color` Height Publisher `Skin color`
##   <chr>  <chr>   <chr>     <chr> <chr>       <dbl> <chr>    <chr>
## 1 A-Bo~ Male    yellow   Human  No Hair      203 Marvel C~ <NA>
## 2 Abe ~ Male   blue    Icth~ No Hair    191 Dark Hor~ blue
## 3 Abin~ Male   blue    Unga~ No Hair    185 DC Comics red
## 4 Abom~ Male   green   Huma~ No Hair    203 Marvel C~ <NA>
## 5 Abra~ Male   blue    Cosm~ Black      NA Marvel C~ <NA>
## 6 Abso~ Male   blue    Human No Hair    193 Marvel C~ <NA>
## 7 Adam~ Male   blue    <NA>  Blond      NA NBC - He~ <NA>

```

```

## # A tibble: 734 x 6
##   X1 name     Publisher `Skin color` Alignment Weight
##   <dbl> <chr>    <chr>      <chr>      <chr>    <dbl>
## 1 0 A-Bomb  Marvel Comics <NA>       good     441
## 2 1 Abe Sapien Dark Horse Comics blue     good     65
## 3 2 Abin Sur   DC Comics     red      good     90
## 4 3 Abomination Marvel Comics <NA>      bad     441
## 5 4 Abraxas   Marvel Comics <NA>      bad      NA
## 6 5 Absorbing Man Marvel Comics <NA>      bad     122
## 7 6 Adam Monroe NBC - Heroes <NA>      good      NA
## 8 7 Adam Strange DC Comics     <NA>      good     88
## 9 8 Agent 13   Marvel Comics <NA>      good     61
## 10 9 Agent Bob  Marvel Comics <NA>      good     81
## # ... with 724 more rows
#& |) tidyselect.
#:, tidyselect , tidyselect.
last_col() : 

heroes %>%
  select(name:last_col())

## # A tibble: 734 x 10
##   name Gender `Eye color` Race `Hair color` Height Publisher `Skin color`
##   <chr> <chr>    <chr>   <chr>      <chr>    <dbl> <chr>      <chr>
## 1 A-Bo~ Male   yellow  Human No Hair      203 Marvel C~ <NA>
## 2 Abe ~ Male   blue    Icth~ No Hair     191 Dark Hor~ blue
## 3 Abin~ Male   blue    Unga~ No Hair     185 DC Comics red
## 4 Abom~ Male   green   Huma~ No Hair     203 Marvel C~ <NA>
## 5 Abra~ Male   blue    Cosm~ Black      NA Marvel C~ <NA>
## 6 Abso~ Male   blue    Human No Hair    193 Marvel C~ <NA>
## 7 Adam~ Male   blue    <NA>   Blond      NA NBC - He~ <NA>
## 8 Adam~ Male   blue    Human Blond     185 DC Comics <NA>
## 9 Agen~ Female blue    <NA>   Blond      173 Marvel C~ <NA>
## 10 Agen~ Male   brown   Human Brown     178 Marvel C~ <NA>
## # ... with 724 more rows, and 2 more variables: Alignment <chr>, Weight <dbl>
everything()

```

```

heroes %>%
  select(everything())

## # A tibble: 734 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr>     <dbl> <chr>
## 1     0 A-Bo~ Male  yellow Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male  blue  Icth~ No Hair     191 Dark Hor~
## 3     2 Abin~ Male  blue  Unga~ No Hair    185 DC Comics
## 4     3 Abom~ Male  green Huma~ No Hair    203 Marvel C~
## 5     4 Abra~ Male  blue  Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male  blue  Human No Hair    193 Marvel C~
## 7     6 Adam~ Male  blue  <NA> Blond        NA NBC - He~
## 8     7 Adam~ Male  blue  Human Blond     185 DC Comics
## 9     8 Agen~ Female blue  <NA> Blond     173 Marvel C~
## 10    9 Agen~ Male  brown Human Brown     178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

everything() , everything()
:

heroes %>%
  select(name, Publisher, everything())

## # A tibble: 734 x 11
##       name Publisher X1 Gender `Eye color` Race `Hair color` Height
##   <chr> <chr> <dbl> <chr> <chr> <chr>     <dbl>
## 1 A-Bo~ Marvel C~     0 Male  yellow Human No Hair      203
## 2 Abe ~ Dark Hor~     1 Male  blue  Icth~ No Hair     191
## 3 Abin~ DC Comics     2 Male  blue  Unga~ No Hair    185
## 4 Abom~ Marvel C~     3 Male  green Huma~ No Hair    203
## 5 Abra~ Marvel C~     4 Male  blue  Cosm~ Black        NA
## 6 Abso~ Marvel C~     5 Male  blue  Human No Hair    193
## 7 Adam~ NBC - He~     6 Male  blue  <NA> Blond        NA
## 8 Adam~ DC Comics     7 Male  blue  Human Blond     185
## 9 Agen~ Marvel C~     8 Female blue  <NA> Blond     173
## 10 Agen~ Marvel C~    9 Male  brown Human Brown     178
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

, relocate() (@ref(tidy_relocate))
. , ends_with() , :
:

heroes %>%
  select(ends_with("color"))

```

```

## # A tibble: 734 x 3
##   `Eye color` `Hair color` `Skin color`
##   <chr>        <chr>        <chr>
## 1 yellow       No Hair      <NA>
## 2 blue         No Hair      blue
## 3 blue         No Hair      red
## 4 green        No Hair      <NA>
## 5 blue         Black        <NA>
## 6 blue         No Hair      <NA>
## 7 blue         Blond        <NA>
## 8 blue         Blond        <NA>
## 9 blue         Blond        <NA>
## 10 brown        Brown        <NA>
## # ... with 724 more rows

       ,           starts_with()
contains() —                                     ,  

heroes %>%
  select(starts_with("Eye") & ends_with("color"))

## # A tibble: 734 x 1
##   `Eye color`
##   <chr>
## 1 yellow
## 2 blue
## 3 blue
## 4 green
## 5 blue
## 6 blue
## 7 blue
## 8 blue
## 9 blue
## 10 brown
## # ... with 724 more rows

heroes %>%
  select(contains("eight"))

## # A tibble: 734 x 2
##   Height Weight
##   <dbl>   <dbl>
## 1 203     441
## 2 191     65
## 3 185     90

```

6
matches().

```

### 4    203    441
### 5      NA     NA
### 6    193    122
### 7      NA     NA
### 8    185     88
### 9    173     61
### 10   178     81
### # ... with 724 more rows

,                                     where().
sapply()(@ref(apply_other)) : where , ,
TRUE.

heroes %>%
  select(where(is.numeric))

## # A tibble: 734 x 3
##       X1 Height Weight
##       <dbl>  <dbl>  <dbl>
## 1     0    203    441
## 2     1    191     65
## 3     2    185     90
## 4     3    203    441
## 5     4      NA     NA
## 6     5    193    122
## 7     6      NA     NA
## 8     7    185     88
## 9     8    173     61
## 10    9    178     81
## # ... with 724 more rows

where() . , NA:
heroes %>%
  select(where(function(x) !any(is.na(x))))
```



```

## # A tibble: 734 x 3
##       X1 name      Publisher
##       <dbl> <chr>    <chr>
## 1     0 A-Bomb    Marvel Comics
## 2     1 Abe Sapien Dark Horse Comics
## 3     2 Abin Sur   DC Comics
## 4     3 Abomination Marvel Comics
## 5     4 Abraxas    Marvel Comics
## 6     5 Absorbing Man Marvel Comics
## 7     6 Adam Monroe NBC - Heroes
## 8     7 Adam Strange DC Comics
```

```
## 9     8 Agent 13      Marvel Comics
## 10    9 Agent Bob    Marvel Comics
## # ... with 724 more rows
```

9.6.3 : dplyr::rename()

```
select() , :
heroes %>%
  select(id = X1)

## # A tibble: 734 x 1
##       id
##   <dbl>
## 1     0
## 2     1
## 3     2
## 4     3
## 5     4
## 6     5
## 7     6
## 8     7
## 9     8
## 10    9
## # ... with 724 more rows

dplyr::rename(), .
select(), rename(), .
heroes %>%
  rename(id = X1)

## # A tibble: 734 x 11
##       id name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1     0 A-Bo~ Male  yellow Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male  blue  Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male  blue  Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male  green Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male  blue  Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male  blue  Human No Hair      193 Marvel C~
## 7     6 Adam~ Male  blue  <NA>  Blond        NA NBC - He-
## 8     7 Adam~ Male  blue  Human Blond       185 DC Comics
## 9     8 Agen~ Female blue  <NA>  Blond       173 Marvel C~
## 10    9 Agen~ Male  brown Human Brown       178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>
```

```

tidyselect          (           rename_with().
                      )           ,
                    

heroes %>%
  rename_with(make.names)

## # A tibble: 734 x 11
##       X1 name Gender Eye.color Race Hair.color Height Publisher Skin.color
##   <dbl> <chr> <chr> <chr>    <chr> <chr>     <dbl> <chr>      <chr>
## 1     0 A-Bo~ Male  yellow  Human No Hair     203 Marvel C~ <NA>
## 2     1 Abe ~ Male  blue   Icth~ No Hair    191 Dark Hor~ blue
## 3     2 Abin~ Male  blue   Unga~ No Hair    185 DC Comics red
## 4     3 Abom~ Male  green  Huma~ No Hair    203 Marvel C~ <NA>
## 5     4 Abra~ Male  blue   Cosm~ Black      NA Marvel C~ <NA>
## 6     5 Abso~ Male  blue   Human No Hair   193 Marvel C~ <NA>
## 7     6 Adam~ Male  blue   <NA> Blond      NA NBC - He~ <NA>
## 8     7 Adam~ Male  blue   Human Blond     185 DC Comics <NA>
## 9     8 Agen~ Female blue   <NA> Blond     173 Marvel C~ <NA>
## 10    9 Agen~ Male  brown  Human Brown     178 Marvel C~ <NA>
## # ... with 724 more rows, and 2 more variables: Alignment <chr>, Weight <dbl>

```

9.6.4 : dplyr::relocate()

```

            relocate().

select()  rename()7.  rename(),      relocate()
          :

heroes %>%
  relocate(Publisher)

## # A tibble: 734 x 11
##   Publisher   X1 name Gender `Eye color` Race `Hair color` Height
##   <chr>     <dbl> <chr> <chr>    <chr> <chr>     <dbl>
## 1 Marvel C~     0 A-Bo~ Male  yellow  Human No Hair     203
## 2 Dark Hor~     1 Abe ~ Male  blue   Icth~ No Hair    191
## 3 DC Comics     2 Abin~ Male  blue   Unga~ No Hair    185
## 4 Marvel C~     3 Abom~ Male  green  Huma~ No Hair    203
## 5 Marvel C~     4 Abra~ Male  blue   Cosm~ Black      NA
## 6 Marvel C~     5 Abso~ Male  blue   Human No Hair   193
## 7 NBC - He~     6 Adam~ Male  blue   <NA> Blond      NA
## 8 DC Comics     7 Adam~ Male  blue   Human Blond     185
## 9 Marvel C~     8 Agen~ Female blue   <NA> Blond     173
## 10 Marvel C~    9 Agen~ Male  brown  Human Brown     178
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,

```

⁷relocate()

select() rename()

```

## # Alignment <chr>, Weight <dbl>
relocate() .after = .before =,
.

heroes %>%
  relocate(Publisher, .after = name)

## # A tibble: 734 x 11
##   X1 name Publisher Gender `Eye color` Race `Hair color` Height
##   <dbl> <chr> <chr>    <chr> <chr>    <chr> <chr>      <dbl>
## 1     0 A-Bo~ Marvel C~ Male   yellow Human No Hair     203
## 2     1 Abe ~ Dark Hor~ Male   blue  Icth~ No Hair     191
## 3     2 Abin~ DC Comics Male   blue  Unga~ No Hair     185
## 4     3 Abom~ Marvel C~ Male   green Huma~ No Hair     203
## 5     4 Abra~ Marvel C~ Male   blue  Cosm~ Black       NA
## 6     5 Abso~ Marvel C~ Male   blue  Human No Hair    193
## 7     6 Adam~ NBC - He~ Male   blue  <NA> Blond       NA
## 8     7 Adam~ DC Comics Male   blue  Human Blond     185
## 9     8 Agen~ Marvel C~ Female blue  <NA> Blond     173
## 10    9 Agen~ Marvel C~ Male   brown Human Brown     178
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

relocate() tidyselect. ,
.

heroes %>%
  relocate(Publisher, where(is.numeric), .after = name)

## # A tibble: 734 x 11
##   name Publisher X1 Height Weight Gender `Eye color` Race `Hair color` 
##   <chr> <chr>    <dbl> <dbl> <dbl> <chr> <chr>    <chr> <chr>  
## 1 A-Bo~ Marvel C~     0    203    441 Male   yellow Human No Hair 
## 2 Abe ~ Dark Hor~    1    191     65 Male   blue  Icth~ No Hair 
## 3 Abin~ DC Comics    2    185     90 Male   blue  Unga~ No Hair 
## 4 Abom~ Marvel C~    3    203    441 Male   green Huma~ No Hair 
## 5 Abra~ Marvel C~    4    NA      NA Male   blue  Cosm~ Black  
## 6 Abso~ Marvel C~    5    193    122 Male   blue  Human No Hair 
## 7 Adam~ NBC - He~    6    NA      NA Male   blue  <NA> Blond  
## 8 Adam~ DC Comics    7    185     88 Male   blue  Human Blond  
## 9 Agen~ Marvel C~    8    173     61 Female blue  <NA> Blond  
## 10 Agen~ Marvel C~   9    178     81 Male   brown Human Brown 
## # ... with 724 more rows, and 2 more variables: `Skin color` <chr>,
## #   Alignment <chr>
— pull(). ,
$, ..
.

tidyverse,
:
```

```

heroes %>%
  select(Height) %>%
  pull() %>%
  head()

## [1] 203 191 185 203 NA 193

heroes %>%
  pull(Height) %>%
  head()

## [1] 203 191 185 203 NA 193

pull()      name =,
heroes %>%
  pull(Height, name) %>%
  head()

##      A-Bomb    Abe Sapien     Abin Sur  Abomination    Abraxas
##      203          191          185          203            NA
## Absorbing Man
##          193

R, tidyverse
pull() -

```

9.7

9.7.1 : dplyr::slice()

```

dplyr::slice()

heroes %>%
  slice(1:3)

## # A tibble: 3 x 11
##       X1 name  Gender `Eye color` `Race` `Hair color` Height Publisher
##   <dbl> <chr> <chr>    <chr>    <chr>    <chr>    <dbl> <chr>
## 1     0 A-Bo~ Male    yellow    Human  No Hair      203 Marvel C~
## 2     1 Abe ~ Male    blue     Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male    blue    Unga~ No Hair      185 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
```

9.7.2 : dplyr::filter()

```
dplyr::filter() , slice() .
, ( ) .

heroes %>%
  filter(Publisher == "DC Comics")

## # A tibble: 215 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr>    <chr> <chr>      <dbl> <chr>
## 1     2 Abin~ Male   blue    Unga~ No Hair      185 DC Comics
## 2     7 Adam~ Male   blue   Human Blond      185 DC Comics
## 3    13 Alan~ Male   blue    <NA>  Blond      180 DC Comics
## 4    16 Alfr~ Male   blue   Human Black      178 DC Comics
## 5    19 Amazo Male   red    Andr~ <NA>      257 DC Comics
## 6    27 Anim~ Male   blue   Human Blond      183 DC Comics
## 7    31 Anti~ Male   yellow God ~ No Hair      61 DC Comics
## 8    35 Aqua~ Male   blue    <NA>  Blond       NA DC Comics
## 9    36 Aqua~ Male   blue   Atla~ Black      178 DC Comics
## 10   37 Aqua~ Male   blue   Atla~ Blond      185 DC Comics
## # ... with 205 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>
```

9.7.3 slice()

```
slice() , slice() filter().
, dplyr::slice_max() dplyr::slice_min()
, :

heroes %>%
  slice_max(Weight, n = 3)

## # A tibble: 3 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr>    <chr> <chr>      <dbl> <chr>
## 1 575 Sasq~ Male   red    <NA>  Orange      305 Marvel C-
## 2 373 Jugg~ Male   blue   Human Red      287 Marvel C-
## 3 203 Dark~ Male   red    New ~ No Hair    267 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>

heroes %>%
  slice_min(Weight, n = 3)

## # A tibble: 3 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr>    <chr> <chr>      <dbl> <chr>
```

```

## 1 346 Iron~ Male blue <NA> No Hair NA Marvel C~
## 2 302 Groot Male yellow Flor~ <NA> 701 Marvel C~
## 3 350 Jack~ Male blue Human Brown 71 Dark Hor~
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>

slice_sample()
:
heroes %>%
  slice_sample(n = 3)

## # A tibble: 3 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 415 Long~ Male blue Human Blond 188 Marvel C~
## 2 117 Bliz~ Male brown <NA> Brown 175 Marvel C~
## 3 242 Ener~ Female <NA> <NA> <NA> NA HarperCo~
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>

:
heroes %>%
  slice_sample(prop = .01)

## # A tibble: 7 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 642 Suns~ Male brown Muta~ black 173 "Marvel ~
## 2 511 Pena~ <NA> <NA> <NA> <NA> NA "Marvel ~
## 3 423 Magn~ Male grey Muta~ White 188 "Marvel ~
## 4 657 Than~ Male red Eter~ No Hair 201 "Marvel ~
## 5 579 Scar~ Male blue Human Blond 178 "Marvel ~
## 6 90 Bird~ Male <NA> God ~ <NA> NA "Hanna-B-
## 7 348 Jack~ Male <NA> <NA> <NA> NA ""
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>

prop = 1,
:
heroes %>%
  slice_sample(prop = 1)

## # A tibble: 734 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 509 Para~ <NA> <NA> Para~ <NA> NA DC Comics
## 2 135 Brai~ Male green Andr~ No Hair 198 DC Comics
## 3 165 Ceci~ <NA> brown <NA> Brown 170 Marvel C~
## 4 497 Nova Female white Huma~ Red 163 Marvel C~
## 5 713 Whit~ Female brown Human Black NA DC Comics
## 6 505 Orac~ Female blue Human Red 178 DC Comics

```

```

## 7 635 Star~ Female blue Human Blond 165 DC Comics
## 8 249 Fabi~ <NA> blue <NA> Brown 196 Marvel C~
## 9 36 Aqua~ Male blue Atla~ Black 178 DC Comics
## 10 638 Step~ Female <NA> <NA> Blond NA ABC Stud~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## # Alignment <chr>, Weight <dbl>

```

9.7.4 NA: `tidy::drop_na()`

`tidy::drop_na()`.

```

heroes %>%
  drop_na()

```

```

## # A tibble: 50 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1      1 Abe ~ Male blue Icth~ No Hair 191 Dark Hor-
## 2      2 Abin~ Male blue Unga~ No Hair 185 DC Comics
## 3     34 Apoc~ Male red Muta~ Black 213 Marvel C~
## 4     39 Arch~ Male blue Muta~ Blond 183 Marvel C~
## 5     41 Ardi~ Female white Alien Orange 193 Marvel C~
## 6     56 Azaz~ Male yellow Neya~ Black 183 Marvel C~
## 7     74 Beast Male blue Muta~ Blue 180 Marvel C~
## 8     75 Beas~ Male green Human Green 173 DC Comics
## 9     92 Biza~ Male black Biza~ Black 191 DC Comics
## 10   108 Blac~ Male red Demon White 191 Marvel C~
## # ... with 40 more rows, and 3 more variables: `Skin color` <chr>,
## # Alignment <chr>, Weight <dbl>

```

```

,      NA          (
,      NA          .

```

```

heroes %>%
  drop_na(Weight)

```

```

## # A tibble: 495 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1      0 A-Bo~ Male yellow Human No Hair 203 Marvel C~
## 2      1 Abe ~ Male blue Icth~ No Hair 191 Dark Hor-
## 3      2 Abin~ Male blue Unga~ No Hair 185 DC Comics
## 4      3 Abom~ Male green Huma~ No Hair 203 Marvel C~
## 5      5 Abso~ Male blue Human No Hair 193 Marvel C~
## 6      7 Adam~ Male blue Human Blond 185 DC Comics
## 7      8 Agen~ Female blue <NA> Blond 173 Marvel C~

```

```
## 8     9 Agen~ Male   brown      Human Brown      178 Marvel C~
## 9    10 Agen~ Male   <NA>       <NA> <NA>       191 Marvel C~
## 10   11 Air~~ Male   blue       <NA> White      188 Marvel C~
## # ... with 485 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>
drop_na()      tidyselect,          (9.6.2).
```

9.7.5 : dplyr::arrange()

```
dplyr::arrange() ( - )
```

```
heroes %>%
```

```
arrange(Weight)
```

```
## # A tibble: 734 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1    346 Iron~ Male   blue   <NA> No Hair      NA Marvel C~
## 2    302 Groot Male   yellow Flor~ <NA>       701 Marvel C~
## 3    350 Jack~ Male   blue   Human Brown      71 Dark Hor~
## 4    272 Gala~ Male   black  Cosm~ Black      876 Marvel C~
## 5    731 Yoda  Male   brown  Yoda~ White      66 George L~
## 6    255 Fin ~ Male   red   Kaka~ No Hair     975 Marvel C~
## 7    330 Howa~ Male   brown  <NA> Yellow      79 Marvel C~
## 8    396 Kryp~ Male   blue   Kryp~ White      64 DC Comics
## 9    568 Rock~ Male   brown  Anim~ Brown     122 Marvel C~
## 10   208 Dash  Male   blue   Human Blond     122 Dark Hor~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>
```

```
, desc().
```

```
heroes %>%
```

```
arrange(desc(Weight))
```

```
## # A tibble: 734 x 11
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1    575 Sasq~ Male   red   <NA> Orange      305 Marvel C~
## 2    373 Jugg~ Male   blue  Human Red       287 Marvel C~
## 3    203 Dark~ Male   red   New ~ No Hair    267 DC Comics
## 4    283 Giga~ Female green <NA> Red        62.5 DC Comics
## 5    331 Hulk   Male   green Huma~ Green     244 Marvel C~
## 6    549 Red ~ Male   yellow Huma~ Black     213 Marvel C~
## 7    119 Bloo~ Female blue  Human Brown     218 Marvel C~
```

```

## 8 718 Wolf~ Female green      <NA> Auburn      366  Marvel C~
## 9 657 Than~ Male   red       Eter~ No Hair     201  Marvel C~
## 10 0 A-Bo~ Male   yellow    Human No Hair    203  Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

,
—
heroes %>%
  arrange(Gender, desc(Weight))

## # A tibble: 734 x 11
##       X1 name  Gender `Eye color` Race  `Hair color` Height Publisher
##       <dbl> <chr> <chr>   <chr> <chr>   <dbl> <chr>
## 1 283 Giga~ Female green      <NA> Red        62.5 DC Comics
## 2 119 Bloo~ Female blue      Human Brown     218  Marvel C~
## 3 718 Wolf~ Female green      <NA> Auburn     366  Marvel C~
## 4 591 She~~ Female green     Human Green     201  Marvel C~
## 5 320 Hela  Female green     Asga~ Black     213  Marvel C~
## 6 686 Valk~ Female blue      <NA> Blond      191  Marvel C~
## 7 596 Sif   Female blue     Asga~ Black     188  Marvel C~
## 8 271 Frig~ Female blue      <NA> White      180  Marvel C~
## 9 667 Thun~ Female green      <NA> Red        218  Marvel C~
## 10 592 She~~ Female blue     Huma~ No Hair    183  Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

```

9.8 : dplyr::mutate() dplyr::transmute()

```

dplyr::mutate()
heroes %>%
  mutate(imt = Weight/(Height/100)^2) %>%
  select(name, imt) %>%
  arrange(desc(imt))

## # A tibble: 734 x 2
##       name      imt
##       <chr>     <dbl>
## 1 Utgard-Loki 2510.
## 2 Giganta     1613.
## 3 Red Hulk    139.
## 4 Darkseid    115.
## 5 Machine Man 114.
## 6 Thanos      110.

```

```

## 7 Destroyer      108.
## 8 A-Bomb        107.
## 9 Abomination   107.
## 10 Hulk         106.
## # ... with 724 more rows

dplyr::transmute() -      mutate(),
:
heroes %>%
  transmute(imt = Weight/(Height/100)^2)

## # A tibble: 734 x 1
##       imt
##   <dbl>
## 1 107.
## 2 17.8
## 3 26.3
## 4 107.
## 5 NA
## 6 32.8
## 7 NA
## 8 25.7
## 9 20.4
## 10 25.6
## # ... with 724 more rows

mutate()  transmute()          (
  ),      ,
  , . . .           (3.3.1):
heroes %>%
  transmute(name, weight_mean = mean(Weight, na.rm = TRUE))

## # A tibble: 734 x 2
##       name      weight_mean
##   <chr>        <dbl>
## 1 A-Bomb      112.
## 2 Abe Sapien  112.
## 3 Abin Sur   112.
## 4 Abomination 112.
## 5 Abraxas    112.
## 6 Absorbing Man 112.
## 7 Adam Monroe 112.
## 8 Adam Strange 112.
## 9 Agent 13    112.
## 10 Agent Bob   112.
## # ... with 724 more rows

```

```

    mutate()  transmute()
      1           ,
heroes %>%
  mutate(one_and_two = 1:2)

## Error: Problem with `mutate()`` input `one_and_two`.
## x Input `one_and_two` can't be recycled to size 734.
## i Input `one_and_two` is `1:2`.
## i Input `one_and_two` must be size 734 or 1, not 2.

,   :      dplyr      ,      —      dplyr      rep(),
      .           .           .           .           .
(3.1).

heroes %>%
  mutate(one_and_two = rep(1:2, length.out = nrow(.)))

## # A tibble: 734 x 12
##       X1 name Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr>     <chr> <chr>      <dbl> <chr>
## 1      0 A-Bo~ Male  yellow    Human No Hair      203 Marvel C~
## 2      1 Abe ~ Male  blue     Icth~ No Hair      191 Dark Hor~
## 3      2 Abin~ Male  blue     Unga~ No Hair      185 DC Comics
## 4      3 Abom~ Male  green   Huma~ No Hair      203 Marvel C~
## 5      4 Abra~ Male  blue     Cosm~ Black        NA Marvel C~
## 6      5 Abso~ Male  blue     Human No Hair      193 Marvel C~
## 7      6 Adam~ Male  blue     <NA> Blond        NA NBC - He~
## 8      7 Adam~ Male  blue     Human Blond       185 DC Comics
## 9      8 Agen~ Female blue     <NA> Blond       173 Marvel C~
## 10     9 Agen~ Male  brown   Human Brown       178 Marvel C~

## # ... with 724 more rows, and 4 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>, one_and_two <int>

```

9.9

9.9.1 : summarise()

```

    ,   ,
  .           ,           ,
  dplyr, dplyr
  :     (group_by())           (summarise()).
  dplyr::summarise()8

```

8 dplyr::summarise() dplyr::summarize(),

```

mutate(),           mutate()           ,           ,
      ,           ,   summarise()           ,           ,
1.   , min(), mean(), max() ..           ,           ,
      mutate()).           (           ,

heroes %>%
  mutate(imt = Weight/(Height/100)^2) %>%
  summarise(min(imt, na.rm = TRUE),
            max(imt, na.rm = TRUE))

## # A tibble: 1 x 2
##   `min(imt, na.rm = TRUE)` `max(imt, na.rm = TRUE)`
##   <dbl>                 <dbl>
## 1 0.0814                2510.

dplyr           tidyverse.           ,
  dplyr::nth(), dplyr::first()  dplyr::last(),
  ( -       slice(),           )

heroes %>%
  mutate(imt = Weight/(Height/100)^2) %>%
  arrange(imt) %>%
  summarise(first = first(imt),
            tenth = nth(imt, 10),
            last = last(imt))

## # A tibble: 1 x 3
##   first tenth last
##   <dbl> <dbl> <dbl>
## 1 0.0814 16.7   NA

  mutate(),           summarise()

      ,           ,           .

heroes %>%
  mutate(imt = Weight/(Height/100)^2) %>%
  summarise(imt_range = range(imt, na.rm = TRUE)) #     range()

## # A tibble: 2 x 1
##   imt_range
##   <dbl>
## 1 0.0814
## 2 2510.

:

```

9.9.2 : group_by()

```

dplyr::group_by() -
  summarise().           group_by()           ,

```

```

groups9:
heroes %>%
  group_by(Gender)

## # A tibble: 734 x 11
## # Groups:   Gender [3]
##       X1 name  Gender `Eye color` Race `Hair color` Height Publisher
##       <dbl> <chr> <chr>     <chr> <chr>      <dbl> <chr>
## 1     0 A-Bo~ Male    yellow Human No Hair      203 Marvel C~
## 2     1 Abe ~ Male    blue  Icth~ No Hair      191 Dark Hor~
## 3     2 Abin~ Male    blue  Unga~ No Hair      185 DC Comics
## 4     3 Abom~ Male    green Huma~ No Hair      203 Marvel C~
## 5     4 Abra~ Male    blue  Cosm~ Black        NA Marvel C~
## 6     5 Abso~ Male    blue  Human No Hair      193 Marvel C~
## 7     6 Adam~ Male    blue  <NA> Blond        NA NBC - He-
## 8     7 Adam~ Male    blue  Human Blond       185 DC Comics
## 9     8 Agen~ Female  blue  <NA> Blond        173 Marvel C~
## 10    9 Agen~ Male    brown Human Brown       178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <chr>,
## #   Alignment <chr>, Weight <dbl>

  summarise(),
  .

heroes %>%
  mutate(imt = Weight/(Height/100)^2) %>%
  group_by(Gender) %>%
  summarise(min(imt, na.rm = TRUE),
            max(imt, na.rm = TRUE))

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

## # A tibble: 3 x 3
##   Gender `min(imt, na.rm = TRUE)` `max(imt, na.rm = TRUE)`
##   <chr>          <dbl>           <dbl>
## 1 Female         15.5            1613.
## 2 Male           0.0814          2510.
## 3 <NA>           16.3            114.

:

```

9.9.3 : dplyr::n(), dplyr::count()

n().

⁹ ungroup().

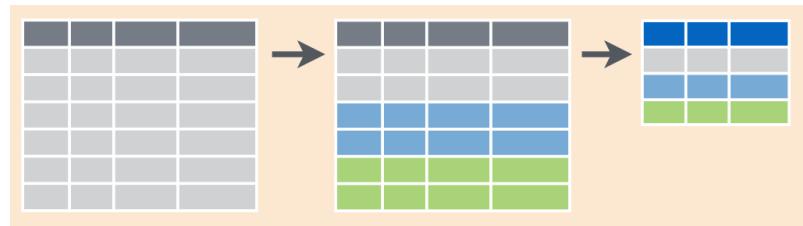


Figure 9.1:

```

heroes %>%
  group_by(Gender) %>%
  summarise(n = n())

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

## # A tibble: 3 x 2
##   Gender     n
##   <chr>   <int>
## 1 Female    200
## 2 Male      505
## 3 <NA>       29

n()      group_by()      filter()      "      "
...

```

```

heroes %>%
  group_by(Race) %>%
  filter(n() > 10) %>%
  select(name, Race)

## # A tibble: 611 x 2
## # Groups:   Race [6]
##   name        Race
##   <chr>        <chr>
## 1 A-Bomb      Human
## 2 Abomination Human / Radiation
## 3 Absorbing Man Human
## 4 Adam Monroe <NA>
## 5 Adam Strange Human
## 6 Agent 13    <NA>
## 7 Agent Bob   Human
## 8 Agent Zero  <NA>
## 9 Air-Walker   <NA>
## 10 Ajax        Cyborg
## # ... with 601 more rows

```

```

    ,
    :
heroes %>%
  group_by(Race) %>%
  filter(n() == 1) %>%
  select(name, Race)

## # A tibble: 34 x 2
## # Groups:   Race [34]
##   name      Race
##   <chr>     <chr>
## 1 Abe Sapien  Icthyo Sapien
## 2 Abin Sur   Ungaran
## 3 Alien      Xenomorph XX121
## 4 Azazel     Neyaphem
## 5 Bizarro    Bizarro
## 6 Boba Fett   Human / Clone
## 7 Darth Maul  Dathomirian Zabrak
## 8 Fin Fang Foom Kakarantharaian
## 9 Gamora     Zen-Whoberian
## 10 Gladiator  Strontian
## # ... with 24 more rows

group_by() summarise(n = n()). count()

:
heroes %>%
  count(Gender)

## # A tibble: 3 x 2
##   Gender   n
##   <chr> <int>
## 1 Female  200
## 2 Male   505
## 3 <NA>    29

,          sort = TRUE.

heroes %>%
  count(Gender, sort = TRUE)

## # A tibble: 3 x 2
##   Gender   n
##   <chr> <int>
## 1 Male   505
## 2 Female 200
## 3 <NA>    29

count(),          tidyverse.

```

9.9.4 : dplyr::distinct()

```
dplyr::distinct() - unique(),
.

heroes %>%
  distinct(Gender)

## # A tibble: 3 x 1
##   Gender
##   <chr>
## 1 Male
## 2 Female
## 3 <NA>

heroes %>%
  distinct(Gender, Race)

## # A tibble: 81 x 2
##   Gender Race
##   <chr>  <chr>
## 1 Male   Human
## 2 Male   Icthyo Sapien
## 3 Male   Ungaran
## 4 Male   Human / Radiation
## 5 Male   Cosmic Entity
## 6 Male   <NA>
## 7 Female <NA>
## 8 Male   Cyborg
## 9 Male   Xenomorph XX121
## 10 Male  Android
## # ... with 71 more rows
```

9.9.5

```
tidyverse          group_by()  mutate() (  summarise()):
heroes %>%
  group_by(Race) %>%
  mutate(Race_n = n()) %>%
  select(Race, name, Gender, Race_n)

## # A tibble: 734 x 4
## # Groups:   Race [62]
##   Race           name      Gender  Race_n
```

```

##   <chr>      <chr>      <chr>    <int>
## 1 Human       A-Bomb     Male     208
## 2 Ichthyosapien Abe Sapien  Male      1
## 3 Ungaran     Abin Sur   Male      1
## 4 Human / Radiation Abomination  Male     11
## 5 Cosmic Entity Abraxas   Male      4
## 6 Human       Absorbing Man Male     208
## 7 <NA>        Adam Monroe Male    304
## 8 Human       Adam Strange Male    208
## 9 <NA>        Agent 13   Female   304
## 10 Human      Agent Bob   Male     208
## # ... with 724 more rows

```



Figure 9.2:

9.10**: dplyr::across()**

```

, , ,
summarise(), : ,
heroes %>%
  group_by(Gender) %>%
  summarise(height = mean(Height, na.rm = TRUE),
            weight = mean(Weight, na.rm = TRUE))

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

## # A tibble: 3 x 3
##   Gender height weight
##   <chr>    <dbl>  <dbl>
## 1 Female    175.   78.8
## 2 Male      192.   126.
## 3 <NA>     177.   129.

```

```

dplyr          : dplyr::across()10.
apply()        tidyselect .
,
across()       :
1.          tidyselect.      ,
(9.6.2).      ,
2.          across().      .col —   ,
tidyselect,     everything(), ..   .fns
—           ,
            across().
3.          summarise()    dplyr.
across().      ,

,
Height  Weight.
heroes %>%
  group_by(Gender) %>%
    summarise(across(c(Height,Weight), mean))

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

## # A tibble: 3 x 3
##   Gender Height Weight
##   <chr>   <dbl>  <dbl>
## 1 Female    NA     NA
## 2 Male     NA     NA
## 3 <NA>    NA     NA

                           :      mean()          NA
NA,                      na.rm =.          apply() .
(@ref(apply_f)),          : 

heroes %>%
  group_by(Gender) %>%
    summarise(across(c(Height, Weight), mean, na.rm = TRUE))

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

## # A tibble: 3 x 3
##   Gender Height Weight
##   <chr>   <dbl>  <dbl>
## 1 Female    175.   78.8
## 2 Male     192.   126.
## 3 <NA>    177.   129.


```

¹⁰ across() dplyr tidyverse
 *_at(), *_if(), *_all(), , summarise_at(), summarise_if(),
 summarize_all(). dplyr, .
 purrr (??) apply() (@ref(apply_f)).

```

    .           across()      tidyselect
    .           ,             numeric
    :

heroes %>%
  drop_na(Height, Weight) %>%
  group_by(Gender) %>%
  summarise(across(where(is.numeric), mean, na.rm = TRUE))

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

## # A tibble: 3 x 4
##   Gender     X1 Height Weight
##   <chr>    <dbl>  <dbl>   <dbl>
## 1 Female    394.   174.   78.3
## 2 Male      369.   193.   126.
## 3 <NA>      375.   182.   129.

,
(@ref(anon_f)).

heroes %>%
  group_by(Gender) %>%
  summarise(across(where(is.character),
                  function(x) mean(nchar(x), na.rm = TRUE)))

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

## # A tibble: 3 x 8
##   Gender   name `Eye color`  Race `Hair color` Publisher `Skin color` Alignment
##   <chr>    <dbl>        <dbl>   <dbl>        <dbl>       <dbl>        <dbl>
## 1 Female    9.04        4.68    6.42        5.05     11.5        4.57     3.88
## 2 Male      9.05        4.53    6.75        5.48     11.4        5.02     3.78
## 3 <NA>      9.48        5.16   10.1        6.44     11.9         4       3.96
,
            summarise()!

heroes %>%
  group_by(Gender) %>%
  summarise(across(where(is.numeric), mean, na.rm = TRUE),
            across(where(is.character),
                  function(x) mean(nchar(x), na.rm = TRUE)))

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

## # A tibble: 3 x 11
##   Gender     X1 Height Weight   name `Eye color`  Race `Hair color` Publisher
##   <chr>    <dbl>  <dbl>   <dbl> <dbl>        <dbl>   <dbl>        <dbl>       <dbl>
## 1 Female    395.   175.   78.8  9.04        4.68    6.42        5.05     11.5
## 2 Male      357.   192.   126.  9.05        4.53    6.75        5.48     11.4

```

```

### 3 <NA>    329    177.  129.   9.48      5.16 10.1      6.44    11.9
## # ... with 2 more variables: `Skin color` <dbl>, Alignment <dbl>

  across(),
  .       ,
  (      -      ).

heroes %>%
  group_by(Gender) %>%
  summarise(across(c(Height, Weight),
                  list(minimum = min,
                        average = mean,
                        maximum = max),
                  na.rm = TRUE))

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

## # A tibble: 3 x 7
##   Gender Height_minimum Height_average Height_maximum Weight_minimum
##   <chr>        <dbl>          <dbl>          <dbl>          <dbl>
## 1 Female       62.5          175.          366            41
## 2 Male         15.2          192.          975             2
## 3 <NA>         108           177.          198            39
## # ... with 2 more variables: Weight_average <dbl>, Weight_maximum <dbl>
## # ... with 2 more variables: Weight_maximum <dbl>,
## #   @ref(functions_objects))!

heroes %>%
  group_by(Gender) %>%
  summarise(across(c(Height, Weight),
                  list(min = function(x) min(x, na.rm = TRUE),
                        mean = function(x) mean(x, na.rm = TRUE),
                        max = function(x) max(x, na.rm = TRUE),
                        na_n = function(x, ...) sum(is.na(x)))
                  )
  )

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

## # A tibble: 3 x 9
##   Gender Height_min Height_mean Height_max Height_na_n Weight_min Weight_mean
##   <chr>     <dbl>      <dbl>      <dbl>      <int>      <dbl>      <dbl>
## 1 Female      62.5      175.      366       56        41       78.8
## 2 Male        15.2      192.      975      147        2       126.
## 3 <NA>        108       177.      198       14        39       129.
## # ... with 2 more variables: Weight_max <dbl>, Weight_na_n <int>

  across() —                         summarise(),
  across()               dplyr.       ,
  mutate():

```

```
heroes %>%
  mutate(across(where(is.character), as.factor))

## # A tibble: 734 x 11
##       X1 name  Gender `Eye color` Race  `Hair color` Height Publisher
##       <dbl> <fct> <fct>    <fct> <fct>      <dbl> <fct>
## 1      0 A-Bo~ Male   yellow  Human No Hair     203 Marvel C~
## 2      1 Abe ~ Male   blue   Icth~ No Hair     191 Dark Hor~
## 3      2 Abin~ Male   blue   Unga~ No Hair     185 DC Comics
## 4      3 Abom~ Male   green  Huma~ No Hair     203 Marvel C~
## 5      4 Abra~ Male   blue   Cosm~ Black        NA Marvel C~
## 6      5 Abso~ Male   blue   Human No Hair    193 Marvel C~
## 7      6 Adam~ Male   blue   <NA>  Blond        NA NBC - He~
## 8      7 Adam~ Male   blue   Human Blond      185 DC Comics
## 9      8 Agen~ Female blue   <NA>  Blond        173 Marvel C~
## 10     9 Agen~ Male   brown  Human Brown     178 Marvel C~
## # ... with 724 more rows, and 3 more variables: `Skin color` <fct>,
## #   Alignment <fct>, Weight <dbl>

  across() -         across()     count()
n_distinct(),
  .

heroes %>%
  select(where(function(x) n_distinct(x) <= 6))

## # A tibble: 734 x 2
##       Gender Alignment
##       <chr>   <chr>
## 1 Male    good
## 2 Male    good
## 3 Male    good
## 4 Male    bad
## 5 Male    bad
## 6 Male    bad
## 7 Male    good
## 8 Male    good
## 9 Female  good
## 10 Male   good
## # ... with 724 more rows

heroes %>%
  count(across(where(function(x) n_distinct(x) <= 6)))

## # A tibble: 11 x 3
##       Gender Alignment     n
##       <chr>   <chr>     <int>
```

```
## 1 Female bad      35
## 2 Female good     161
## 3 Female neutral    4
## 4 Male   bad      165
## 5 Male   good     316
## 6 Male   neutral    18
## 7 Male   <NA>       6
## 8 <NA>   bad       7
## 9 <NA>   good      19
## 10 <NA>  neutral     2
## 11 <NA>  <NA>       1
```

Chapter 10

tidyverse

10.1

```
10.1.1 : bind_rows(), bind_cols()

dc, marvel other_publishers:

dc <- heroes %>%
  filter(Publisher == "DC Comics") %>%
  group_by(Gender) %>%
  summarise(weight_mean = mean(Weight, na.rm = TRUE))

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

## # A tibble: 3 x 2
##   Gender weight_mean
##   <chr>     <dbl>
## 1 Female      76.8
## 2 Male        113.
## 3 <NA>       NaN

marvel <- heroes %>%
  filter(Publisher == "Marvel Comics") %>%
  group_by(Gender) %>%
  summarise(weight_mean = mean(Weight, na.rm = TRUE))

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

## # A tibble: 3 x 2
```

```

##   Gender weight_mean
##   <chr>      <dbl>
## 1 Female     80.1
## 2 Male       134.
## 3 <NA>       129.

other_publishers <- heroes %>%
  filter(!(Publisher %in% c("DC Comics", "Marvel Comics"))) %>%
  group_by(Gender) %>%
  summarise(weight_mean = mean(Weight, na.rm = TRUE))

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

## # A tibble: 3 x 2
##   Gender weight_mean
##   <chr>      <dbl>
## 1 Female     70.8
## 2 Male       111.
## 3 <NA>       NaN

bind_rows().

bind_rows(dc, marvel)

## # A tibble: 6 x 2
##   Gender weight_mean
##   <chr>      <dbl>
## 1 Female     76.8
## 2 Male       113.
## 3 <NA>       NaN
## 4 Female     80.1
## 5 Male       134.
## 6 <NA>       129.

, bind_cols().

bind_cols(dc, marvel)

## New names:
## * Gender -> Gender...1
## * weight_mean -> weight_mean...2
## * Gender -> Gender...3
## * weight_mean -> weight_mean...4

## # A tibble: 3 x 4
##   Gender...1 weight_mean...2 Gender...3 weight_mean...4
##   <chr>      <dbl> <chr>      <dbl>

```

```
## 1 Female           76.8 Female          80.1
## 2 Male              113.  Male          134.
## 3 <NA>             NaN   <NA>          129.

bind_rows() bind_cols() , .
bind_rows(dc, marvel, other_publishers)

## # A tibble: 9 x 2
##   Gender weight_mean
##   <chr>     <dbl>
## 1 Female     76.8
## 2 Male      113.
## 3 <NA>      NaN
## 4 Female     80.1
## 5 Male      134.
## 6 <NA>      129.
## 7 Female     70.8
## 8 Male      111.
## 9 <NA>      NaN

bind_rows() bind_cold() , /
heroes_list_of_df <- list(DC = dc,
                           Marvel = marvel,
                           Other = other_publishers)
bind_rows(heroes_list_of_df)

## # A tibble: 9 x 2
##   Gender weight_mean
##   <chr>     <dbl>
## 1 Female     76.8
## 2 Male      113.
## 3 <NA>      NaN
## 4 Female     80.1
## 5 Male      134.
## 6 <NA>      129.
## 7 Female     70.8
## 8 Male      111.
## 9 <NA>      NaN

, , , ( )
.id =.

bind_rows(heroes_list_of_df, .id = "Publisher")

## # A tibble: 9 x 3
##   Publisher Gender weight_mean
```

```

##   <chr>    <chr>    <dbl>
## 1 DC      Female     76.8
## 2 DC      Male       113.
## 3 DC      <NA>      NaN
## 4 Marvel  Female     80.1
## 5 Marvel  Male       134.
## 6 Marvel  <NA>      129.
## 7 Other   Female     70.8
## 8 Other   Male       111.
## 9 Other   <NA>      NaN

bind_rows(),
,
bind_rows().

```

10.1.2 : *_join()

```

),
, , , ,
“ ”
,
,
,
bind_rows() bind_cols()
(join),
,
,
,
dplyr
*_join().
band_members band_instruments, dplyr
*_join().
band_members

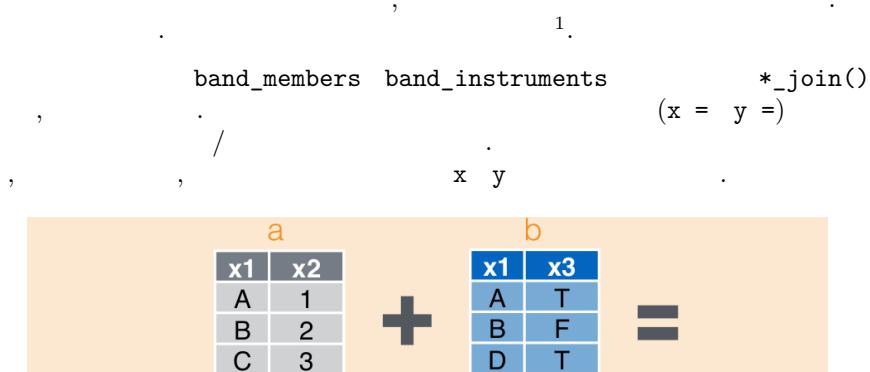
## # A tibble: 3 x 2
##   name   band
##   <chr> <chr>
## 1 Mick  Stones
## 2 John  Beatles
## 3 Paul  Beatles

band_instruments

## # A tibble: 3 x 2
##   name   plays
##   <chr> <chr>
## 1 John  guitar

```

```
## 2 Paul bass
## 3 Keith guitar
```



Mutating Joins

x1	x2	x3
A	1	T
B	2	F
C	3	NA

dplyr::left_join(a, b, by = "x1")

Join matching rows from b to a.

x1	x3	x2
A	T	1
B	F	2
D	T	NA

dplyr::right_join(a, b, by = "x1")

Join matching rows from a to b.

x1	x2	x3
A	1	T
B	2	F

dplyr::inner_join(a, b, by = "x1")

Join data. Retain only rows in both sets.

x1	x2	x3
A	1	T
B	2	F
C	3	NA
D	NA	T

dplyr::full_join(a, b, by = "x1")

Join data. Retain all values, all rows.

Figure 10.1:

- `left_join()`:

```
band_members %>%
  left_join(band_instruments)
```

```
## Joining, by = "name"
```

```
## # A tibble: 3 x 3
##   name   band   plays
```

1 *_join()
 , , ,
 , , ,
 , *_join()

```

## <chr> <chr> <chr>
## 1 Mick Stones <NA>
## 2 John Beatles guitar
## 3 Paul Beatles bass

left_join() -
  " "
  y. , ,
  y, , ,
  - by =,
  . , ,
  : , ,

band_members %>%
  left_join(band_instruments, by = "name")

## # A tibble: 3 x 3
##   name   band   plays
##   <chr> <chr> <chr>
## 1 Mick Stones <NA>
## 2 John Beatles guitar
## 3 Paul Beatles bass

, , ,
: , ,

band_members %>%
  left_join(band_instruments2, by = c("name" = "artist"))

## # A tibble: 3 x 3
##   name   band   plays
##   <chr> <chr> <chr>
## 1 Mick Stones <NA>
## 2 John Beatles guitar
## 3 Paul Beatles bass

• right_join():

band_members %>%
  right_join(band_instruments)

## Joining, by = "name"

## # A tibble: 3 x 3
##   name   band   plays
##   <chr> <chr> <chr>
## 1 John Beatles guitar
## 2 Paul Beatles bass
## 3 Keith <NA>   guitar

```

```

right_join()           x,           y,           y -
left_join()           .
  • full_join():
band_members %>%
  full_join(band_instruments)

## Joining, by = "name"
## # A tibble: 4 x 3
##   name   band   plays
##   <chr> <chr>  <chr>
## 1 Mick  Stones <NA>
## 2 John  Beatles guitar
## 3 Paul  Beatles bass
## 4 Keith <NA>   guitar

full_join()           x   y.      ,
left_join() —       full_join()
  • inner_join():
band_members %>%
  inner_join(band_instruments)

## Joining, by = "name"
## # A tibble: 2 x 3
##   name   band   plays
##   <chr> <chr>  <chr>
## 1 John  Beatles guitar
## 2 Paul  Beatles bass

full_join()           ,           x,     y.
  • semi_join():
band_members %>%
  semi_join(band_instruments)

## Joining, by = "name"
## # A tibble: 2 x 2
##   name   band
##   <chr> <chr>
## 1 John  Beatles
## 2 Paul  Beatles

  • anti_join():
band_members %>%
  anti_join(band_instruments)

```

```
## Joining, by = "name"
## # A tibble: 1 x 2
##   name   band
##   <chr> <chr>
## 1 Mick Stones
  semi_join() anti_join() / (y) .
  -           -           x,       y
(semi_join()) , , y (anti_join()).
```

10.2 Tidy data: `tidyr::pivot_longer()`, `tidyr::pivot_wider()`

tidy data , ,
 , , () , : , ,
 - .

() , “ ” ,
 , - ()? ,
 • :
 -

R	R
70	63
80	74
86	71

• “ ” :
 -

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

, , , ,
 , , , ,
 ,

- `tidyr::pivot_longer()`:

- `tidyr::pivot_wider()`:

```

new_diet <- tibble(
  student = c(" ", " ", " "),
  before_r_course = c(70, 80, 86),
  after_r_course = c(63, 74, 71)
)
new_diet

## # A tibble: 3 x 3
##   student  before_r_course after_r_course
##   <chr>        <dbl>         <dbl>
## 1          70            63
## 2          80            74
## 3          86            71

new_diet -
  new_diet    :
new_diet %>%
  pivot_longer(cols = before_r_course:after_r_course,
                names_to = "measurement_time",
                values_to = "weight_kg")

## # A tibble: 6 x 3
##   student measurement_time weight_kg
##   <chr>    <chr>           <dbl>
## 1          before_r_course    70
## 2          after_r_course     63
## 3          before_r_course    80
## 4          after_r_course     74
## 5          before_r_course    86
## 6          after_r_course     71

:
new_diet %>%
  pivot_longer(cols = before_r_course:after_r_course,
                names_to = "measurement_time",
                values_to = "weight_kg") %>%
  pivot_wider(names_from = "measurement_time",
              values_from = "weight_kg")

## # A tibble: 3 x 3
##   student  before_r_course after_r_course
##   <chr>        <dbl>         <dbl>
## 1          70            63

```

## 2	80	74
## 3	86	71

Chapter 11

11.1 R

```
•         9801   9.  
## [1] 1089  
•             8912162342       6.  
## [1] 12  
•                 10           5.  
## [1] 11.51293  
•             sin()      sin(π),sin(π/2),sin(π/6).  
π -          R      (pi).  
## [1] 1.224647e-16  
## [1] 1  
## [1] 0.5
```

11.2

```
•         2, 30   4000.  
## [1]    2   30 4000  
•             1   20.  
## [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

```

•          20   1.

## [1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
sum() . . . . . . . . . . . . . . . . . . . . . . . . . .
1 100) ( ..

## [1] 5050

•          1 20      1.    20      !
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 19 18 17 16 15
## [26] 14 13 12 11 10 9 8 7 6 5 4 3 2 1
•          5, 4, 3, 2, 2, 3, 4, 5:
## [1] 5 4 3 2 2 3 4 5

•          2, 4, 6, ... , 18, 20.

## [1] 2 4 6 8 10 12 14 16 18 20
•          0.1, 0.2, 0.3, ..., 0.9, 1.

## [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
• 2020 — . . . . . . . . . . . . . . . . . . . . . .
XXI , 2020 . .
2100 XXI , XXII.

## [1] 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060 2064 2068 2072 2076
## [16] 2080 2084 2088 2092 2096 2100

• , 20 “ !”.
## [1] " !" " !" " !" " !" " !" " !" " !" " !" " !" " !" " !
## [11] " !" " !" " !" " !" " !" " !" " !" " !" " !" " !" " !
• , , , 1 10 , sqrt()
roots.

## [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
## [9] 3.000000 3.162278

• , . . . . roots
!

## [1] 1 2 3 4 5 6 7 8 9 10
• , . . . . roots .
## [1] 1 2 3 4 5 6 7 8 9 10
• * , , , , , ..., .

```

```
## [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 9 9
## [39] 9 9 9 9 9 9 9
```

11.3

```
• vec1, 3, " " " "
## [1] "3" " " " "
• TRUE 10.
## [1] 9
• 10 TRUE vec2.
## [1] 10 1
• vec2 "r":
## [1] "10" "1" "r"
• 10, TRUE, "r" .
## [1] "10" "TRUE" "r"
```

11.4

```
• p, 4, 5, 6, 7, q, 0, 1, 2, 3.
## [1] 4 5 6 7
## [1] 0 1 2 3
• p q:
## [1] 4 6 8 10
• p q:
## [1] 4 4 4 4
• p q:
, , 0!
## [1] Inf 5.000000 3.000000 2.333333
• p q:
## [1] 1 5 36 343
• p 10.
```

```

## [1] 40 50 60 70
• 1 10:
## [1] 1 4 9 16 25 36 49 64 81 100
• 0, 2, 0, 4, ..., 18, 0, 20.
## [1] 0 2 0 4 0 6 0 8 0 10 0 12 0 14 0 16 0 18 0 20
• 1, 0, 3, 0, 5, ..., 17, 0, 19, 0.
## [1] 1 0 3 0 5 0 7 0 9 0 11 0 13 0 15 0 17 0 19 0
• * , 20 .
## [1] 2 4 8 16 32 64 128 256 512
## [10] 1024 2048 4096 8192 16384 32768 65536 131072 262144
## [19] 524288 1048576
• * 1, 10, 100, 1000, 10000:
## [1] 1 10 100 1000 10000
• *  $\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{50 \cdot 51}$ .
## [1] 0.9803922
• *  $\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^{20}}$ .
## [1] 1.999999
• *  $1 + \frac{4}{3} + \frac{7}{9} + \frac{10}{27} + \frac{13}{81} + \dots + \frac{28}{19683}$ .
## [1] 3.749174
• *  $1 + \frac{4}{3} + \frac{7}{9} + \frac{10}{27} + \frac{13}{81} + \dots + \frac{28}{19683}$  0.5?
## [1] 3

```

11.5

```

• troiki 3, 6, 9, ..., 24, 27.
## [1] 3 6 9 12 15 18 21 24 27
• 2, 5 7 troiki.
## [1] 6 15 21
• troiki.
## [1] 24
• troiki :

```

```

## [1]  3  6  9 12 15 18 21 27
      vec3:
vec3 <- c(3, 5, 2, 1, 8, 4, 9, 10, 3, 15, 1, 11)

•           vec3.

## [1] 5
•           vec3.

## [1] 5 8
•           vec3:
## [1] NA
•           vec3 .
.
## [1] 3 2 1 8 4 9 10 3 15 1 11
•           vec3 .
.
## [1] 3 2 1 4 9 10 3 15 1 11
•           vec3.

## [1] 11
•           vec3 .
.
## [1] 5 2 1 8 4 9 10 3 15 1
•           vec3,        4.
## [1] 5 8 9 10 15 11
•           vec3,        4,        10.
,          !
## [1] 5 8 9
•           vec3,        4        10.
## [1] 3 2 1 3 15 1 11
•           vec3.

## [1] 9 25 4 1 64 16 81 100 9 225 1 121
• *
      vec3.

- ,        0.5.
## [1] 9.000000 2.236068 4.000000 1.000000 64.000000 2.000000 81.000000
## [8] 3.162278 9.000000 3.872983 1.000000 3.316625

```

```

•           2, 4, 6, ... , 18, 20      2
          , , . , , , ! , ,
## [1] 2 4 6 8 10 12 14 16 18 20

```

11.6

```

•           vec4           300, 15, 8, 2, 0, 1, 110:
vec4 <- c(300, 15, 8, 20, 0, 1, 110)
vec4

## [1] 300 15   8   20   0   1 110
•           vec4,           20   NA.
•           vec4:
## [1] NA 15   8 20   0   1 NA
•           vec4           sum().   NA       !
## [1] 44

```

11.7

```

•           4 4,           .           M1.
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    1    1    1
## [3,]    1    1    1    1
## [4,]    1    1    1    1
•           M1 (           [2,2], [2,3], [3,2]  [3,3])
2.

##      [,1] [,2] [,3] [,4]
## [1,]    1    1    1    1
## [2,]    1    2    2    1
## [3,]    1    2    2    1
## [4,]    1    1    1    1
•           M1.
##      [,1] [,2]
## [1,]    1    1

```

```

## [2,]    2    2
## [3,]    2    2
## [4,]    1    1
•      (==)          M1.

## [1] TRUE TRUE TRUE TRUE

• *      (9 9)          .
               mult_tab.

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]    1    2    3    4    5    6    7    8    9
## [2,]    2    4    6    8   10   12   14   16   18
## [3,]    3    6    9   12   15   18   21   24   27
## [4,]    4    8   12   16   20   24   28   32   36
## [5,]    5   10   15   20   25   30   35   40   45
## [6,]    6   12   18   24   30   36   42   48   54
## [7,]    7   14   21   28   35   42   49   56   63
## [8,]    8   16   24   32   40   48   56   64   72
## [9,]    9   18   27   36   45   54   63   72   81
• *      mult_tab          ,
               6     8     3
7.

## [,1] [,2] [,3] [,4] [,5]
## [1,] 18  24  30  36  42
## [2,] 21  28  35  42  49
## [3,] 24  32  40  48  56
• *      ,   TRUE,          (mult_tab)
               FALSE,
               .

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE FALSE TRUE  TRUE  TRUE  TRUE  TRUE
## [3,] FALSE FALSE FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [4,] FALSE FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [5,] FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [6,] FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [7,] FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [8,] FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [9,] FALSE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
• *      mult_tab2,          tab    10     0.
## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] 0    0    0    0    0    0    0    0    0
## [2,] 0    0    0    0   10   12   14   16   18
## [3,] 0    0    0   12   15   18   21   24   27

```

```
## [4,] 0 0 12 16 20 24 28 32 36
## [5,] 0 10 15 20 25 30 35 40 45
## [6,] 0 12 18 24 30 36 42 48 54
## [7,] 0 14 21 28 35 42 49 56 63
## [8,] 0 16 24 32 40 48 56 64 72
## [9,] 0 18 27 36 45 54 63 72 81
```

11.8

```
list1:
list1 = list(numbers = 1:5, letters = letters, logic = TRUE)
list1

## $numbers
## [1] 1 2 3 4 5
##
## $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"
##
## $logic
## [1] TRUE

• list1.

## $numbers
## [1] 1 2 3 4 5

• list1

## [1] 1 2 3 4 5

## [1] 1 2 3 4 5

• list1.

## [1] 1

• list2, list1. pupa,
— lupa.

## $pupa
## $pupa$numbers
## [1] 1 2 3 4 5
##
## $pupa$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"
```

```

## 
## $lupa$logic
## [1] TRUE
##
##
## $lupa
## $lupa$numbers
## [1] 1 2 3 4 5
##
## $lupa$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"
##
## $lupa$logic
## [1] TRUE

• *           list2,      —          ,      —          .
## [1] "c"

```

11.9

```

•           data(mtcars)
-           ,           -           .           ?mtcars.

data(mtcars)
mtcars

##          mpg cyl disp hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant       18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360    14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D     24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230      22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280      19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE     16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL     17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC    15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4

```

```

## Fiat 128      32.4   4 78.7 66 4.08 2.200 19.47 1 1 4 1
## Honda Civic   30.4   4 75.7 52 4.93 1.615 18.52 1 1 4 2
## Toyota Corolla 33.9   4 71.1 65 4.22 1.835 19.90 1 1 4 1
## Toyota Corona  21.5   4 120.1 97 3.70 2.465 20.01 1 0 3 1
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87 0 0 3 2
## AMC Javelin    15.2   8 304.0 150 3.15 3.435 17.30 0 0 3 2
## Camaro Z28     13.3   8 350.0 245 3.73 3.840 15.41 0 0 3 4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05 0 0 3 2
## Fiat X1-9       27.3   4 79.0 66 4.08 1.935 18.90 1 1 4 1
## Porsche 914-2    26.0   4 120.3 91 4.43 2.140 16.70 0 1 5 2
## Lotus Europa     30.4   4 95.1 113 3.77 1.513 16.90 1 1 5 2
## Ford Pantera L   15.8   8 351.0 264 4.22 3.170 14.50 0 1 5 4
## Ferrari Dino     19.7   6 145.0 175 3.62 2.770 15.50 0 1 5 6
## Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60 0 1 5 8
## Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60 1 1 4 2

•           mtcars      str().

## 'data.frame': 32 obs. of 11 variables:
## $ mpg : num 21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num 6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num 160 160 108 258 360 ...
## $ hp  : num 110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num 3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt  : num 2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num 16.5 17 18.6 19.4 17 ...
## $ vs  : num 0 0 1 1 0 1 0 1 1 1 ...
## $ am  : num 1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num 4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num 4 4 1 1 2 1 4 2 2 4 ...

•           mtcars.

## [1] 93

•           mtcars.

##          mpg cyl disp  hp drat    wt
## Mazda RX4     21.0   6 160 110 3.90 2.620
## Mazda RX4 Wag  21.0   6 160 110 3.90 2.875
## Datsun 710    22.8   4 108  93 3.85 2.320
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215
## Hornet Sportabout 18.7   8 360 175 3.15 3.440
## Valiant       18.1   6 225 105 2.76 3.460

•           wt      mtcars -
## [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440 4.070
## [13] 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520 3.435 3.840
## [25] 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780

```

```

•           mtcars          : hp, mpg, cyl.

##          hp   mpg cyl
## Mazda RX4     110 21.0   6
## Mazda RX4 Wag 110 21.0   6
## Datsun 710      93 22.8   4
## Hornet 4 Drive 110 21.4   6
## Hornet Sportabout 175 18.7   8
## Valiant        105 18.1   6
## Duster 360      245 14.3   8
## Merc 240D       62 24.4   4
## Merc 230         95 22.8   4
## Merc 280        123 19.2   6
## Merc 280C        123 17.8   6
## Merc 450SE       180 16.4   8
## Merc 450SL       180 17.3   8
## Merc 450SLC      180 15.2   8
## Cadillac Fleetwood 205 10.4   8
## Lincoln Continental 215 10.4   8
## Chrysler Imperial 230 14.7   8
## Fiat 128          66 32.4   4
## Honda Civic        52 30.4   4
## Toyota Corolla     65 33.9   4
## Toyota Corona      97 21.5   4
## Dodge Challenger    150 15.5   8
## AMC Javelin        150 15.2   8
## Camaro Z28          245 13.3   8
## Pontiac Firebird     175 19.2   8
## Fiat X1-9          66 27.3   4
## Porsche 914-2       91 26.0   4
## Lotus Europa        113 30.4   4
## Ford Pantera L      264 15.8   8
## Ferrari Dino        175 19.7   6
## Maserati Bora        335 15.0   8
## Volvo 142E          109 21.4   4

•           4          (cyl)          mtcars.

## [1] 11

•           4          (cyl)          mtcars.

## [1] 0.34375

•           100          (hp)          mtcars.

##          mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46  0  1     4     4
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1     4     4

```

```

## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0  3  1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0  3  2
## Valiant            18.1   6 225.0 105 2.76 3.460 20.22  1  0  3  1
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0  3  4
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0  4  4
## Merc 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0  4  4
## Merc 450SE         16.4   8 275.8 180 3.07 4.070 17.40  0  0  3  3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0  3  3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0  3  3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0  3  4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0  3  4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0  3  4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0  3  2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0  3  2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0  3  4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0  3  2
## Lotus Europa         30.4   4  95.1 113 3.77 1.513 16.90  1  1  5  2
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1  5  4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1  5  6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1  5  8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1  4  2
•                               100          (hp)  4          (cyl)
mtcars.

##                   mpg cyl disp hp drat    wt qsec vs am gear carb
## Lotus Europa  30.4   4  95.1 113 3.77 1.513 16.9   1  1  5  2
## Volvo 142E    21.4   4 121.0 109 4.11 2.780 18.6   1  1  4  2
•                               (wt)          ,          max():
## [1] 5.424
•                               (wt)          ,          min():
## [1] 1.513
•                               mtcars
##                   mpg cyl disp hp drat    wt qsec vs am gear carb
## Lotus Europa  30.4   4  95.1 113 3.77 1.513 16.9   1  1  5  2
•                               mtcars          ,
##                   mpg cyl disp hp drat    wt qsec vs am gear carb
## Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46  0  1  4  4
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1  4  4
## Datsun 710    22.8   4 108.0  93 3.85 2.320 18.61  1  1  4  1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0  3  1
## Merc 240D     24.4   4 146.7  62 3.69 3.190 20.00  1  0  4  2
## Merc 230     22.8   4 140.8  95 3.92 3.150 22.90  1  0  4  2

```

```

## Fiat 128      32.4   4 78.7 66 4.08 2.200 19.47 1 1 4 1
## Honda Civic   30.4   4 75.7 52 4.93 1.615 18.52 1 1 4 2
## Toyota Corolla 33.9   4 71.1 65 4.22 1.835 19.90 1 1 4 1
## Toyota Corona  21.5   4 120.1 97 3.70 2.465 20.01 1 0 3 1
## Fiat X1-9     27.3   4 79.0 66 4.08 1.935 18.90 1 1 4 1
## Porsche 914-2  26.0   4 120.3 91 4.43 2.140 16.70 0 1 5 2
## Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90 1 1 5 2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50 0 1 5 4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.50 0 1 5 6
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60 1 1 4 2

```

```

•           .       wt_kg
round().

1 = 0.45359237 .

```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	wt_kg
## Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4	1188
## Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4	1304
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	1052
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1	1458
## Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2	1560
## Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1	1569
## Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	1619
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2	1447
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2	1429
## Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4	1560
## Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4	1560
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3	1846
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3	1692
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3	1715
## Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	2381
## Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4	2460
## Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	2424
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	998
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2	733
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1	832
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1	1118
## Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2	1597
## AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	1558
## Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	1742
## Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2	1744
## Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	878
## Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2	971
## Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2	686
## Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4	1438
## Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	1256

```
## Maserati Bora      15.0   8 301.0 335 3.54 3.570 14.60 0 1 5 8 1619
## Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60 1 1 4 2 1261
```

11.10

- vec5:

```
vec5 <- c(5, 20, 30, 0, 2, 9)
```

- , 10 vec5 "", ,

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

- heroes_information.csv heroes.

```
heroes <- read.csv("data/heroes_information.csv",
  stringsAsFactors = FALSE,
  na.strings = c("-", "-99"))
```

- hair heroes, "Bold" ,
Hair.color "No Hair", "Hairy" .

```
##   X      name Gender Eye.color      Race Hair.color Height
## 1 0     A-Bomb  Male  yellow    Human  No Hair   203
## 2 1     Abe Sapien  Male   blue  Icthyo Sapien  No Hair   191
## 3 2     Abin Sur  Male   blue    Ungaran  No Hair   185
## 4 3 Abomination  Male  green Human / Radiation  No Hair   203
## 5 4     Abraxas  Male   blue  Cosmic Entity  Black    NA
## 6 5 Absorbing Man  Male   blue    Human  No Hair   193
##   Publisher Skin.color Alignment Weight hair
## 1     Marvel Comics <NA>     good   441 Bold
## 2 Dark Horse Comics   blue     good    65 Bold
## 3     DC Comics     red     good    90 Bold
## 4     Marvel Comics <NA>     bad    441 Bold
## 5     Marvel Comics <NA>     bad    NA Hairy
## 6     Marvel Comics <NA>     bad   122 Bold
```

- tall heroes, "tall" ,
Height 190, "short" ,
Height 170, "middle" .

11.11

- plus_one(), + 1.

```

• plus_one()      41.
plus_one(41)

## [1] 42

• circle_area,  $\pi r^2$ .
• 5.

## [1] 78.53982

• cels2fahr(),
• -100, -40 0, cels2fahr()
cels2fahr(c(-100, -40, 0))

## [1] -148 -40 32

• highlight(),
• "***"
highlight(c(" ", " !"))

## [1] "***" " " " !" "***"

• highlight . wrapper =,
• "***". wrapper =
• c(" ", " !").

highlight(c(" ", " !"))

## [1] "***" " " " !" "***"
highlight(c(" ", " !"), wrapper = "_")

## [1] "_" " " " " !" "_"
• trim(),
• , trim():
trim(1:7)

## [1] 2 3 4 5 6
trim(letters)

## [1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [20] "u" "v" "w" "x" "y"
• trim() n = 1.

```

```

•           :
trim(letters)

## [1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [20] "u" "v" "w" "x" "y"
trim(letters, n = 2)

## [1] "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u"
## [20] "v" "w" "x"

•           ,           trim()           n = 0, ...
.

trim(letters, n = 0)

## [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"

• *           :           trim()           ,           n
=           n =           :           :

• *           trim():

trim(1:6, 3)

## Error in trim(1:6, 3): n           !
trim(1:6, -1)

## Error in trim(1:6, -1): n           !
.

•           na_n(),           NA           .
•           na_n()           :           .

na_n(c(NA, 3:5, NA, 2, NA))

## [1] 3

•           factors(),
.

•           factors()           :           : %%.

factors(3)

## [1] 1 3
factors(161)

## [1] 1   7  23 161
factors(1984)

```

```

## [1] 1 2 4 8 16 31 32 62 64 124 248 496 992 1984
• *      is_prime(), , TRUE,
any() - TRUE,
TRUE.

• , :
is_prime(2017)

## [1] TRUE
is_prime(2019)

## [1] FALSE
2019/3 #2019      3

## [1] 673
is_prime(2020)

## [1] FALSE
is_prime(2021)

## [1] FALSE
• *      monotonic(), TRUE, (
-          )
— diff() —
monotonic(1:7)

## [1] TRUE
monotonic(c(1:5,5:1))

## [1] FALSE
monotonic(6:-1)

## [1] TRUE
monotonic(c(1:5, rep(5, 10), 5:10))

## [1] TRUE
+  %in%
!      — ‘’, , ( ), %notin%,
%      TRUE, :

```

```

`%notin%` <- function(x, y) ! (x %in% y)
1:10 %notin% c(1, 4, 5)

## [1] FALSE TRUE TRUE FALSE FALSE TRUE TRUE TRUE TRUE TRUE TRUE
• * %without%,

c(" ", " ", " ", " ", " ", " ", " ") %without% c(" ", " ")

## [1] " " " " " " "
• * %between%, TRUE,
:
1:10 %between% c(1, 4, 5)

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

```

11.12 apply()

```

• M2:
M2 <- matrix(c(20:11, 11:20), nrow = 5)
M2

## [,1] [,2] [,3] [,4]
## [1,] 20 15 11 16
## [2,] 19 14 12 17
## [3,] 18 13 13 18
## [4,] 17 12 14 19
## [5,] 16 11 15 20

• M2
## [1] 20 19 18 19 20

• M2
## [1] 20 15 15 20

• M2
## [1] 15.5 15.5 15.5 15.5 15.5

• M2
## [1] 18 13 13 18

• list3:

```

```

list3 <- list(
  a = 1:5,
  b = 0:20,
  c = 4:24,
  d = 6:3,
  e = 6:25
)

•           list3.

##   a   b   c   d   e
##  5 20 24  6 25

•           list3.

##   a   b   c   d   e
## 15 210 294 18 310

•           list3.

##   a   b   c   d   e
##  5 21 21  4 20

•           max_item(),
          ,      - (    )

          which.max(),
(  ,       ).

•           max_item()      list3.

max_item(list3)

## [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20

•           list4:
list4 <- list(1:3, 3:40, list3)

•           ,
          ,   ..
          list4.

rapply(): recursive lapply

## [[1]]
## [1] 3
##
## [[2]]
## [1] 38
##
## [[3]]
## [[3]]$a

```

```

## [1] 5
##
## [[3]]$b
## [1] 21
##
## [[3]]$c
## [1] 21
##
## [[3]]$d
## [1] 4
##
## [[3]]$e
## [1] 20

• *           heroes      ,      NA
                           na_n().

##          X      name   Gender Eye.color      Race Hair.color      Height
##      0      0       29      172      304      172      217
## Publisher Skin.color Alignment   Weight     hair      tall
##      0      662        7      239      172      217

• *           is_prime(),      prime_numbers(),
.

prime_numbers(200)

## [1]  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71
## [20] 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167
## [39] 173 179 181 191 193 197 199

```

11.13 magrittr::%>%

```

•           ,      %>%:
sqrt(sum(1:10))

## [1] 7.416198
## [1] 7.416198
abs(min(-5:5))

## [1] 5
## [1] 5
c(" ", 2, " ", sqrt(2))

```

```
## [1] " " "2" " " "1.4142135623731"
## [1] " " "2" " " "1.4142135623731"
```

11.14 : dplyr::slice() dplyr::filter()

```
• , 500 .

## # A tibble: 6 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 203 Dark~ Male red New ~ No Hair 267 DC Comics
## 2 283 Giga~ Female green <NA> Red 62.5 DC Comics
## 3 331 Hulk Male green Huma~ Green 244 Marvel C~
## 4 373 Jugg~ Male blue Human Red 287 Marvel C~
## 5 549 Red ~ Male yellow Huma~ Black 213 Marvel C~
## 6 575 Sasq~ Male red <NA> Orange 305 Marvel C~
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
• , - 500 .

## # A tibble: 1 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 283 Giga~ Female green <NA> Red 62.5 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
• , ("Human")
5.

## # A tibble: 5 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 38 Arac~ Female blue Human Blond 175 Marvel C~
## 2 63 Batg~ Female green Human Red 170 DC Comics
## 3 65 Batg~ Female green Human Black 165 DC Comics
## 4 72 Batw~ Female green Human Red 178 DC Comics
## 5 96 Blac~ Female blue Human Blond 165 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
```

11.15 : dplyr::select()

```
• 4 powers.
```

```
## # A tibble: 667 x 4
##   hero_names Agility `Accelerated Healing` `Lantern Power Ring`
```

```

##   <chr>    <lgl>    <lgl>    <lgl>
## 1 3-D Man      TRUE     FALSE    FALSE
## 2 A-Bomb       FALSE     TRUE     FALSE
## 3 Abe Sapien   TRUE     TRUE     FALSE
## 4 Abin Sur     FALSE     FALSE    TRUE
## 5 Abomination  FALSE     TRUE     FALSE
## 6 Abraxas      FALSE     FALSE    FALSE
## 7 Absorbing Man FALSE     FALSE    FALSE
## 8 Adam Monroe  FALSE     TRUE     FALSE
## 9 Adam Strange  FALSE     FALSE    FALSE
## 10 Agent Bob   FALSE    FALSE    FALSE
## # ... with 657 more rows

•             Reflexes  Empathy    powers:

## # A tibble: 667 x 7
##   Reflexes Invulnerability `Energy Constru~ `Force Fields` `Self-Sustenanc~
##   <lgl>    <lgl>    <lgl>    <lgl>    <lgl>
## 1 FALSE    FALSE     FALSE    FALSE    FALSE
## 2 FALSE    FALSE     FALSE    FALSE    TRUE
## 3 TRUE     FALSE     FALSE    FALSE    FALSE
## 4 FALSE    FALSE     FALSE    FALSE    FALSE
## 5 FALSE    TRUE      FALSE    FALSE    FALSE
## 6 FALSE    TRUE      FALSE    FALSE    FALSE
## 7 FALSE    TRUE      FALSE    FALSE    FALSE
## 8 FALSE    FALSE     FALSE    FALSE    FALSE
## 9 FALSE    FALSE     FALSE    FALSE    FALSE
## 10 FALSE   FALSE     FALSE    FALSE   FALSE
## # ... with 657 more rows, and 2 more variables: `Anti-Gravity` <lgl>,
## #     Empathy <lgl>

•             powers    (hero_names):

## # A tibble: 667 x 167
##   Agility `Accelerated He~ `Lantern Power ~ `Dimensional Aw~ `Cold Resistanc~
##   <lgl>    <lgl>    <lgl>    <lgl>    <lgl>
## 1 TRUE     FALSE     FALSE    FALSE    FALSE
## 2 FALSE    TRUE      FALSE    FALSE    FALSE
## 3 TRUE     TRUE      FALSE    FALSE    TRUE
## 4 FALSE    FALSE     TRUE     FALSE    FALSE
## 5 FALSE    TRUE      FALSE    FALSE    FALSE
## 6 FALSE    FALSE     FALSE    TRUE     FALSE
## 7 FALSE    FALSE     FALSE    FALSE    TRUE
## 8 FALSE    TRUE      FALSE    FALSE    FALSE
## 9 FALSE    FALSE     FALSE    FALSE    FALSE
## 10 FALSE   FALSE     FALSE   FALSE    FALSE
## # ... with 657 more rows, and 162 more variables: Durability <lgl>,
## #     Stealth <lgl>, `Energy Absorption` <lgl>, Flight <lgl>, `Danger

```

```

## # Sense` <lgl>, `Underwater breathing` <lgl>, Marksmanship <lgl>, `Weapons
## # Master` <lgl>, `Power Augmentation` <lgl>, `Animal Attributes` <lgl>,
## # Longevity <lgl>, Intelligence <lgl>, `Super Strength` <lgl>,
## # Cryokinesis <lgl>, Telepathy <lgl>, `Energy Armor` <lgl>, `Energy
## # Blasts` <lgl>, Duplication <lgl>, `Size Changing` <lgl>, `Density
## # Control` <lgl>, Stamina <lgl>, `Astral Travel` <lgl>, `Audio
## # Control` <lgl>, Dexterity <lgl>, Omnitrix <lgl>, `Super Speed` <lgl>,
## # Possession <lgl>, `Animal Oriented Powers` <lgl>, `Weapon-based
## # Powers` <lgl>, Electrokinesis <lgl>, `Darkforce Manipulation` <lgl>, `Death
## # Touch` <lgl>, Teleportation <lgl>, `Enhanced Senses` <lgl>,
## # Telekinesis <lgl>, `Energy Beams` <lgl>, Magic <lgl>, Hyperkinesis <lgl>,
## # Jump <lgl>, Clairvoyance <lgl>, `Dimensional Travel` <lgl>, `Power
## # Sense` <lgl>, Shapeshifting <lgl>, `Peak Human Condition` <lgl>,
## # Immortality <lgl>, Camouflage <lgl>, `Element Control` <lgl>,
## # Phasing <lgl>, `Astral Projection` <lgl>, `Electrical Transport` <lgl>,
## # `Fire Control` <lgl>, Projection <lgl>, Summoning <lgl>, `Enhanced
## # Memory` <lgl>, Reflexes <lgl>, Invulnerability <lgl>, `Energy
## # Constructs` <lgl>, `Force Fields` <lgl>, `Self-Sustenance` <lgl>,
## # `Anti-Gravity` <lgl>, Empathy <lgl>, `Power Nullifier` <lgl>, `Radiation
## # Control` <lgl>, `Psionic Powers` <lgl>, Elasticity <lgl>, `Substance
## # Secretion` <lgl>, `Elemental Transmogrification` <lgl>,
## # `Technopath/Cyberpath` <lgl>, `Photographic Reflexes` <lgl>, `Seismic
## # Power` <lgl>, Animation <lgl>, Precognition <lgl>, `Mind Control` <lgl>,
## # `Fire Resistance` <lgl>, `Power Absorption` <lgl>, `Enhanced
## # Hearing` <lgl>, `Nova Force` <lgl>, Insanity <lgl>, Hypnokinesis <lgl>,
## # `Animal Control` <lgl>, `Natural Armor` <lgl>, Intangibility <lgl>,
## # `Enhanced Sight` <lgl>, `Molecular Manipulation` <lgl>, `Heat
## # Generation` <lgl>, Adaptation <lgl>, Gliding <lgl>, `Power Suit` <lgl>,
## # `Mind Blast` <lgl>, `Probability Manipulation` <lgl>, `Gravity
## # Control` <lgl>, Regeneration <lgl>, `Light Control` <lgl>,
## # Echolocation <lgl>, Levitation <lgl>, `Toxin and Disease Control` <lgl>,
## # Banish <lgl>, `Energy Manipulation` <lgl>, `Heat Resistance` <lgl>,
## # `Natural Weapons` <lgl>, ...

```

11.16

: dplyr::arrange()

- heroes name, Gender, Height
Height.

```

## # A tibble: 734 x 3
##   name      Gender Height
##   <chr>     <chr>   <dbl>
## 1 Utgard-Loki  Male    15.2
## 2 Bloodwraith Male    30.5

```

```
## 3 King Kong      Male   30.5
## 4 Anti-Monitor  Male   61
## 5 Giganta       Female 62.5
## 6 Krypto         Male   64
## 7 Yoda           Male   66
## 8 Jack-Jack     Male   71
## 9 Howard the Duck Male  79
## 10 Godzilla      <NA>  108
## # ... with 724 more rows
```

- heroes name, Gender, Height
Height.

```
## # A tibble: 734 x 3
##   name      Gender Height
##   <chr>     <chr>   <dbl>
## 1 Fin Fang Foom  Male    975
## 2 Galactus     Male    876
## 3 Groot        Male    701
## 4 MODOK        Male    366
## 5 Wolfsbane    Female  366
## 6 Onslaught    Male    305
## 7 Sasquatch    Male    305
## 8 Ymir         Male    305.
## 9 Rey          Female  297
## 10 Juggernaut  Male    287
## # ... with 724 more rows
```

- heroes name, Gender, Height
Gender, Height.

```
## # A tibble: 734 x 3
##   name      Gender Height
##   <chr>     <chr>   <dbl>
## 1 Wolfsbane Female  366
## 2 Rey        Female  297
## 3 Bloodaxe   Female  218
## 4 Thundra    Female  218
## 5 Hela       Female  213
## 6 Frenzy     Female  211
## 7 She-Hulk   Female  201
## 8 Ardina     Female  193
## 9 Starfire   Female  193
## 10 Valkyrie  Female  191
## # ... with 724 more rows
```

11.17 : dplyr::distinct()

```

•           Eye color      heroes.

## # A tibble: 23 x 1
##   `Eye color`
##   <chr>
## 1 yellow
## 2 blue
## 3 green
## 4 brown
## 5 <NA>
## 6 red
## 7 violet
## 8 white
## 9 purple
## 10 black
## # ... with 13 more rows

•           Hair color      heroes.

## # A tibble: 30 x 1
##   `Hair color`
##   <chr>
## 1 No Hair
## 2 Black
## 3 Blond
## 4 Brown
## 5 <NA>
## 6 White
## 7 Purple
## 8 Orange
## 9 Pink
## 10 Red
## # ... with 20 more rows

```

11.18 : dplyr::mutate() dplyr::transmute()

```

•           height_m      ,      name

height_m.

## # A tibble: 734 x 2
##   name      height_m
##   <chr>     <dbl>
## 1 A-Bomb    2.03
## 2 Abe Sapien 1.91

```

```

## 3 Abin Sur      1.85
## 4 Abomination   2.03
## 5 Abraxas       NA
## 6 Absorbing Man  1.93
## 7 Adam Monroe    NA
## 8 Adam Strange    1.85
## 9 Agent 13       1.73
## 10 Agent Bob     1.78
## # ... with 724 more rows

      •          hair heroes,           "Bold"
      Hair.color   "No Hair",        "Hairy"
      name, Hair color, hair.

## # A tibble: 734 x 3
##   name     `Hair color` hair
##   <chr>     <chr>     <chr>
## 1 A-Bomb   No Hair   Bold
## 2 Abe Sapien No Hair   Bold
## 3 Abin Sur  No Hair   Bold
## 4 Abomination No Hair   Bold
## 5 Abraxas   Black     Hairy
## 6 Absorbing Man No Hair   Bold
## 7 Adam Monroe Blond     Hairy
## 8 Adam Strange  Blond     Hairy
## 9 Agent 13    Blond     Hairy
## 10 Agent Bob   Brown     Hairy
## # ... with 724 more rows

```

11.19 : dplyr::group_by() %>% summarise()

```

      •          .          5          .

## # A tibble: 5 x 2
##   Race      n
##   <chr>    <int>
## 1 <NA>      304
## 2 Human     208
## 3 Mutant    63
## 4 God / Eternal  14
## 5 Cyborg    11

      •          .          .

## # A tibble: 3 x 2
##   Gender height_mean
##   <chr>     <dbl>
## 1 Male      70.2
## 2 Female    65.5
## 3 Non-Binary 67.7

```

```
## <chr>      <dbl>
## 1 Female    175.
## 2 Male     192.
## 3 <NA>    177.
```

11.20 : across()

- NA , (Gender).

```
## # A tibble: 3 x 11
##   Gender   X1 name `Eye color` `Race` `Hair color` Height Publisher
##   <chr>   <int> <int>       <int> <int>       <int> <int> <int>
## 1 Female    0     0         41    98        38    56       0
## 2 Male     0     0        121   184       123   147       0
## 3 <NA>     0     0         10    22        11    14       0
## # ... with 3 more variables: `Skin color` <int>, Alignment <int>, Weight <int>
```

- NA , "color", (Gender).

```
## # A tibble: 3 x 4
##   Gender `Eye color` `Hair color` `Skin color`
##   <chr>       <int>       <int>       <int>
## 1 Female      41        38       186
## 2 Male       121       123       449
## 3 <NA>       10        11        27
```

- heroes name, Height Weight, " ", "
- .

```
## # A tibble: 734 x 3
##   name      Height      Weight
##   <chr>      <chr>      <chr>
## 1 A-Bomb
## 2 Abe Sapien
## 3 Abin Sur
## 4 Abomination
## 5 Abraxas    <NA>      <NA>
## 6 Absorbing Man
## 7 Adam Monroe <NA>      <NA>
## 8 Adam Strange
## 9 Agent 13
## 10 Agent Bob
## # ... with 724 more rows
```

```

•      heroes           Gender, name, Height  Weight,
      "          ",           "
      ",           .
      ",

## # A tibble: 734 x 4
## # Groups:   Gender [3]
##   Gender name      Height      Weight
##   <chr>  <chr>     <chr>     <chr>
## 1 Male   A-Bomb
## 2 Male   Abe Sapien
## 3 Male   Abin Sur
## 4 Male   Abomination
## 5 Male   Abraxas    <NA>       <NA>
## 6 Male   Absorbing Man
## 7 Male   Adam Monroe <NA>       <NA>
## 8 Male   Adam Strange
## 9 Female Agent 13
## 10 Male  Agent Bob
## # ... with 724 more rows

```

11.21 : *_join {#task_join}

```

•      web_creators,           ,   ,
TRUE      Web Creation      powers.

## # A tibble: 16 x 12
##   X1 name  Gender `Eye color` Race  `Hair color` Height Publisher
##   <dbl> <chr> <chr>    <chr> <chr>    <dbl> <chr>
## 1 33 Anti~ Male   blue     Symb~ Blond      229 Marvel C~
## 2 38 Arac~ Female blue    Human Blond    175 Marvel C~
## 3 161 Carn~ Male   green   Symb~ Red       185 Marvel C~
## 4 335 Hybr~ Male   brown   Symb~ Black     175 Marvel C~
## 5 479 Myst~ Male   brown   Human No Hair 180 Marvel C~
## 6 580 Scar~ Male   brown   Clone Brown   193 Marvel C~
## 7 597 Silk  Female brown  Human Black    NA Marvel C~
## 8 620 Spid~ Female blue  Human Brown   170 Marvel C~
## 9 621 Spid~ Female blue  Human Blond   165 Marvel C~
## 10 622 Spid~ Male   hazel  Human Brown   178 Marvel C~
## 11 623 Spid~ <NA>   red   Human Brown   178 Marvel C~
## 12 624 Spid~ Male   brown  Human Black   157 Marvel C~
## 13 673 Toxin  Male   blue   Symb~ Brown    188 Marvel C~
## 14 674 Toxin  Male   black  Symb~ Blond    191 Marvel C~
## 15 689 Venom  Male   blue   Symb~ Strawberry ~ 191 Marvel C~
## 16 692 Veno~ Male   <NA>   Symb~ <NA>    226 Marvel C~
## # ... with 4 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>,

```

```
## # `Web Creation` <lgl>
• , heroes, powers.

## [1] "Agent 13"      "Alfred Pennyworth" "Arsenal"
## [4] "Batgirl III"    "Batgirl V"        "Beetle"
## [7] "Black Goliath"   "Black Widow II"   "Blaquesmith"
## [10] "Bolt"           "Boomer"          "Box"
## [13] "Box III"         "Captain Mar-vell" "Cat II"
## [16] "Cecilia Reyes"   "Clea"             "Clock King"
## [19] "Colin Wagner"    "Colossal Boy"    "Corsair"
## [22] "Cypher"          "Danny Cooper"    "Darkside"
## [25] "ERG-1"           "Fixer"           "Franklin Storm"
## [28] "Giant-Man"       "Giant-Man II"   "Goliath"
## [31] "Goliath"          "Goliath"          "Guardian"
## [34] "Hawkwoman"        "Hawkwoman II"   "Hawkwoman III"
## [37] "Howard the Duck"  "Jack Bauer"       "Jesse Quick"
## [40] "Jessica Sanders"  "Jigsaw"          "Jyn Erso"
## [43] "Kid Flash II"     "Kingpin"          "Meteorite"
## [46] "Mister Zsasz"     "Mogo"            "Moloch"
## [49] "Morph"            "Nite Owl II"     "Omega Red"
## [52] "Paul Blart"        "Penance"          "Penance I"
## [55] "Plastic Lad"       "Power Man"        "Renata Soliz"
## [58] "Ronin"             "Shrinking Violet" "Snake-Eyes"
## [61] "Spider-Carnage"    "Spider-Woman II" "Stacy X"
## [64] "Thunderbird II"    "Two-Face"          "Vagabond"
## [67] "Vision II"         "Vulcan"           "Warbird"
## [70] "White Queen"       "Wiz Kid"           "Wondra"
## [73] "Wyatt Wingfoot"    "Yellow Claw"      ""

• , powers, heroes.

## [1] "3-D Man"          "Bananaman"        "Bizarro-Girl"
## [4] "Black Vulcan"      "Blue Streak"       "Bradley"
## [7] "Clayface"          "Concrete"          "Dementor"
## [10] "Doctor Poison"     "Fire"              "Hellgramite"
## [13] "Lara Croft"        "Little Epic"       "Lord Voldemort"
## [16] "Orion"             "Peek-a-Boo"        "Queen Hippolyta"
## [19] "Reactron"          "SHDB"              "Stretch Armstrong"
## [22] "TEST"              "Tommy Clarke"      "Tyrant"
```

11.22 Tidy data

- heroes_weight, :

```

heroes_weight <- heroes %>%
  filter(Publisher %in% c("DC Comics", "Marvel Comics")) %>%
  group_by(Gender, Publisher) %>%
  summarise(weight_mean = mean(Weight, na.rm = TRUE)) %>%
  drop_na()
heroes_weight

## # A tibble: 4 x 3
## # Groups:   Gender [2]
##   Gender Publisher    weight_mean
##   <chr>   <chr>        <dbl>
## 1 Female  DC Comics     76.8
## 2 Female  Marvel Comics 80.1
## 3 Male    DC Comics    113.
## 4 Male    Marvel Comics 134.

drop_na() , NA.
•      heroes_weight : 

## # A tibble: 2 x 3
## # Groups:   Gender [2]
##   Gender `DC Comics` `Marvel Comics`
##   <chr>      <dbl>        <dbl>
## 1 Female       76.8        80.1
## 2 Male        113.        134.

•      :

## # A tibble: 4 x 3
## # Groups:   Gender [2]
##   Gender Publisher    weight_mean
##   <chr>   <chr>        <dbl>
## 1 Female  DC Comics     76.8
## 2 Female  Marvel Comics 80.1
## 3 Male    DC Comics    113.
## 4 Male    Marvel Comics 134.

•      powers      : hero_names, power ( )
has ( ). 

## # A tibble: 111,389 x 3
##   hero_names power      has
##   <chr>      <chr>    <lgl>
## 1 3-D Man    Agility   TRUE
## 2 3-D Man    Accelerated Healing FALSE
## 3 3-D Man    Lantern Power Ring FALSE
## 4 3-D Man    Dimensional Awareness FALSE
## 5 3-D Man    Cold Resistance FALSE

```

```

## 6 3-D Man Durability FALSE
## 7 3-D Man Stealth FALSE
## 8 3-D Man Energy Absorption FALSE
## 9 3-D Man Flight FALSE
## 10 3-D Man Danger Sense FALSE
## # ... with 111,379 more rows

• powers , :
- ( - ).

## # A tibble: 167 x 668
##   power `3-D Man` `A-Bomb` `Abe Sapien` `Abin Sur` `Abomination` Abraxas
##   <chr> <lgl>   <lgl>   <lgl>   <lgl>   <lgl>   <lgl>
## 1 Agil~ TRUE    FALSE   TRUE    FALSE   FALSE   FALSE
## 2 Acce~ FALSE   TRUE    TRUE    FALSE   TRUE    FALSE
## 3 Lant~ FALSE   FALSE   FALSE   TRUE    FALSE   FALSE
## 4 Dime~ FALSE   FALSE   FALSE   FALSE   FALSE   TRUE
## 5 Cold~ FALSE   FALSE   TRUE    FALSE   FALSE   FALSE
## 6 Dura~ FALSE   TRUE    TRUE    FALSE   FALSE   FALSE
## 7 Stea~ FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
## 8 Ener~ FALSE   FALSE   FALSE   FALSE   FALSE   FALSE
## 9 Flig~ FALSE   FALSE   FALSE   FALSE   FALSE   TRUE
## 10 Dang~ FALSE  FALSE   FALSE   FALSE   FALSE   FALSE
## # ... with 157 more rows, and 661 more variables: `Absorbing Man` <lgl>, `Adam
## # Monroe` <lgl>, `Adam Strange` <lgl>, `Agent Bob` <lgl>, `Agent Zero` <lgl>,
## # `Air-Walker` <lgl>, Ajax <lgl>, `Alan Scott` <lgl>, `Alex Mercer` <lgl>,
## # `Alex Woolsly` <lgl>, Alien <lgl>, `Allan Quatermain` <lgl>, Amazo <lgl>,
## # Ammo <lgl>, `Ando Masahashi` <lgl>, Angel <lgl>, `Angel Dust` <lgl>, `Angel
## # Salvadore` <lgl>, Angela <lgl>, `Animal Man` <lgl>, Annihilus <lgl>,
## # `Ant-Man` <lgl>, `Ant-Man II` <lgl>, `Anti-Monitor` <lgl>,
## # `Anti-Spawn` <lgl>, `Anti-Venom` <lgl>, Apocalypse <lgl>, Aquababy <lgl>,
## # Aqualad <lgl>, Aquaman <lgl>, Arachne <lgl>, Archangel <lgl>,
## # Arclight <lgl>, Ardina <lgl>, Ares <lgl>, Ariel <lgl>, Armor <lgl>, `Astro
## # Boy` <lgl>, Atlas <lgl>, Atom <lgl>, `Atom Girl` <lgl>, `Atom II` <lgl>,
## # `Atom III` <lgl>, `Atom IV` <lgl>, Aurora <lgl>, Azazel <lgl>,
## # Azrael <lgl>, Aztar <lgl>, Bananaman <lgl>, Bane <lgl>, Banshee <lgl>,
## # Bantam <lgl>, Batgirl <lgl>, `Batgirl IV` <lgl>, `Batgirl VI` <lgl>,
## # Batman <lgl>, `Batman II` <lgl>, Battlestar <lgl>, `Batwoman V` <lgl>,
## # Beak <lgl>, Beast <lgl>, `Beast Boy` <lgl>, `Ben 10` <lgl>, `Beta Ray
## # Bill` <lgl>, Beyonder <lgl>, `Big Barda` <lgl>, `Big Daddy` <lgl>, `Big
## # Man` <lgl>, `Bill Harken` <lgl>, `Billy Kincaid` <lgl>, Binary <lgl>,
## # `Bionic Woman` <lgl>, `Bird-Brain` <lgl>, `Bird-Man` <lgl>, `Bird-Man
## # II` <lgl>, Birdman <lgl>, Bishop <lgl>, Bizarro <lgl>,
## # `Bizarro-Girl` <lgl>, `Black Abbott` <lgl>, `Black Adam` <lgl>, `Black
## # Bolt` <lgl>, `Black Canary` <lgl>, `Black Cat` <lgl>, `Black Flash` <lgl>,
## # `Black Knight III` <lgl>, `Black Lightning` <lgl>, `Black Mamba` <lgl>,
## # `Black Manta` <lgl>, `Black Panther` <lgl>, `Black Vulcan` <lgl>, `Black

```

```
## #  Widow` <lgl>, Blackout <lgl>, Blackwing <lgl>, Blackwulf <lgl>,
## #  Blade <lgl>, `Bling!` <lgl>, Blink <lgl>, Blizzard <lgl>, `Blizzard
## #  II` <lgl>, ...
```

Chapter 12

12.1 R

```
•         9801   9.  
9801/9  
## [1] 1089  
•             8912162342      6.  
log(2176782336, 6)  
## [1] 12  
•                 10          5.  
log(10)*5  
## [1] 11.51293  
•           sin()      sin(π), sin(π/2), sin(π/6).  
π -       R      (pi).  
sin(pi)  
## [1] 1.224647e-16  
sin(pi/2)  
## [1] 1  
sin(pi/6)  
## [1] 0.5
```

12.2

```

•          2, 30  4000.

c(2, 30, 4000)

## [1]    2    30 4000
•          1   20.

1:20

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

20:1

## [1] 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
sum()
      .                               100                   ( ..
1   100).

sum(1:100)

## [1] 5050
•          1   20       1.   20
!c(1:20, 19:1)

## [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 19 18 17 16 15
## [26] 14 13 12 11 10 9 8 7 6 5 4 3 2 1
•          5, 4, 3, 2, 2, 3, 4, 5:
c(5:2, 2:5)

## [1] 5 4 3 2 2 3 4 5
•          2, 4, 6, ..., 18, 20.

seq(2, 20, 2)

## [1] 2  4  6  8 10 12 14 16 18 20
•          0.1, 0.2, 0.3, ..., 0.9, 1.

seq(0, 1, 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
• 2020 — .           4 — 2024 .
                           XXI , 2020 .
2100           XXI , XXII.

```

```

seq(2020, 2100, 4)

## [1] 2020 2024 2028 2032 2036 2040 2044 2048 2052 2056 2060 2064 2068 2072 2076
## [16] 2080 2084 2088 2092 2096 2100

• , 20 “ !”.
rep(" !", 20)

## [1] " !" " !" " !" " !" " !" " !" " !" " !" " !" " !" " !" " !
## [11] " !" " !" " !" " !" " !" " !" " !" " !" " !" " !" " !
• , , ,
. 1 10 , sqrt()
roots.

roots <- sqrt(1:10)
roots

## [1] 1.000000 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427
## [9] 3.000000 3.162278

• , .
!
roots ^ 2

## [1] 1 2 3 4 5 6 7 8 9 10
• , .
roots * roots

## [1] 1 2 3 4 5 6 7 8 9 10
• *
, , , , ..., .
rep(1:9, 1:9)

## [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9
## [39] 9 9 9 9 9 9 9

```

12.3

```

• vec1, 3, " " " ".
vec1 <- c(3, " ", " ")
vec1

## [1] "3" " " " "
• TRUE 10.

```

```
10 - TRUE

## [1] 9
• 10 TRUE vec2.
vec2 <- c(10, TRUE)
vec2

## [1] 10 1
• vec2 "r":
c(vec2, "r")

## [1] "10" "1" "r"
• 10, TRUE, "r" .
c(10, TRUE, "r")

## [1] "10" "TRUE" "r"
```

12.4

```
• p, 4, 5, 6, 7, q, 0, 1, 2, 3.
p <- 4:7
p

## [1] 4 5 6 7
q <- 0:3
q

## [1] 0 1 2 3
• p q:
p + q

## [1] 4 6 8 10
• p q:
p - q

## [1] 4 4 4 4
• p q:
, , 0!
```

```

p / q

## [1]      Inf 5.000000 3.000000 2.333333
•           p                               q:
p ^ q

## [1]   1   5  36 343
•           p   10.
p * 10

## [1] 40 50 60 70
•           1   10:
(1:10)^2

## [1]   1   4   9   16   25   36   49   64   81   100
•           0, 2, 0, 4, ... , 18, 0, 20.
1:20 * 0:1

## [1]   0   2   0   4   0   6   0   8   0 10   0 12   0 14   0 16   0 18   0 20
•           1, 0, 3, 0, 5, ..., 17, 0, 19, 0.
1:20 * 1:0

## [1]   1   0   3   0   5   0   7   0   9   0 11   0 13   0 15   0 17   0 19   0
•           *,                20               .
2 ^ (1:20)

## [1]      2       4       8      16      32      64     128     256     512
## [10]    1024    2048    4096    8192   16384   32768   65536  131072  262144
## [19] 524288 1048576

•           1, 10, 100, 1000, 10000:
10 ^ (0:4)

## [1]      1     10    100   1000  10000
•            $\frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots + \frac{1}{50 \cdot 51}$ .
sum(1 / (1:50 * 2:51))

## [1] 0.9803922
•            $\frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \dots + \frac{1}{2^{20}}$ .

```

```

sum(1 / 2 ^ (0:20))

## [1] 1.999999
• *

$$1 + \frac{4}{3} + \frac{7}{9} + \frac{10}{27} + \frac{13}{81} + \dots + \frac{28}{19683}.$$

sum((3 * (1:10) - 2) / 3 ^ (0:9))

## [1] 3.749174
• *

$$1 + \frac{4}{3} + \frac{7}{9} + \frac{10}{27} + \frac{13}{81} + \dots + \frac{28}{19683}$$
 0.5?
sum((3 * (1:10) - 2) / 3 ^ (0:9) > 0.5)

## [1] 3

```

12.5

• troiki 3, 6, 9, ..., 24, 27.

```

troiki <- seq(3, 27, 3)
troiki

```

```
## [1] 3 6 9 12 15 18 21 24 27
```

• 2, 5 7 troiki.

```
troiki[c(2, 5, 7)]
```

```
## [1] 6 15 21
```

• troiki.

```
troiki[length(troiki) - 1]
```

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

• troiki :

```
troiki[-(length(troiki) - 1)]
```

```
## [1] 3 6 9 12 15 18 21 27
```

vec3:

```
vec3 <- c(3, 5, 2, 1, 8, 4, 9, 10, 3, 15, 1, 11)
```

• vec3.

```
vec3[2]
```

```
## [1] 5
```

• vec3.

```
vec3[c(2, 5)]  
## [1] 5 8  
•  
vec3:  
vec3[100]  
## [1] NA  
•  
vec3  
vec3[-2]  
## [1] 3 2 1 8 4 9 10 3 15 1 11  
•  
vec3  
vec3[c(-2, -5)]  
## [1] 3 2 1 4 9 10 3 15 1 11  
•  
vec3.  
vec3[length(vec3)]  
## [1] 11  
•  
vec3  
vec3[c(-1, -length(vec3))]  
## [1] 5 2 1 8 4 9 10 3 15 1  
•  
vec3, 4.  
vec3[vec3 > 4]  
## [1] 5 8 9 10 15 11  
•  
vec3, 4, 10.  
, !  
vec3[vec3 > 4 & vec3 < 10]  
## [1] 5 8 9  
•  
vec3, 4, 10.  
vec3[vec3 < 4 | vec3 > 10]  
## [1] 3 2 1 3 15 1 11  
•  
vec3.
```

```

vec3 ^ 2

## [1] 9 25 4 1 64 16 81 100 9 225 1 121
• *
      vec3.
      - , 0.5.

vec3 ^ c(2, 0.5)

## [1] 9.000000 2.236068 4.000000 1.000000 64.000000 2.000000 81.000000
## [8] 3.162278 9.000000 3.872983 1.000000 3.316625

• 2, 4, 6, ... , 18, 20 2 .
      , , . , , , !

(1:20) [c(FALSE, TRUE)]

## [1] 2 4 6 8 10 12 14 16 18 20
#(1:10)*2

```

12.6

```

• vec4 300, 15, 8, 2, 0, 1, 110:
vec4 <- c(300, 15, 8, 20, 0, 1, 110)
vec4

## [1] 300 15 8 20 0 1 110
• vec4, 20 NA.
vec4[vec4 > 20] <- NA

• vec4:
vec4

## [1] NA 15 8 20 0 1 NA
• vec4 sum(). NA !
sum(vec4, na.rm = TRUE)

## [1] 44

```

12.7

```

•           4 4,           .           M1.

M1 <- matrix(rep(1, 16), ncol = 4)
M1

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

•           M1 (           [2,2], [2,3], [3,2]  [3,3])
2.

M1[2:3, 2:3] <- 2
M1

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

•           M1.

M1[,2:3]

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

•           (==)           M1.

M1[,2] == M1[2,]

## [1] TRUE TRUE TRUE TRUE

•           (9 9)           .           mult_tab.

mult_tab <- matrix(rep(1:9, rep(9,9))*rep(1:9), nrow = 9)
mult_tab

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,]     1     2     3     4     5     6     7     8     9
## [2,]     2     4     6     8    10    12    14    16    18
## [3,]     3     6     9    12    15    18    21    24    27
## [4,]     4     8    12    16    20    24    28    32    36

```

```

## [5,]   5   10   15   20   25   30   35   40   45
## [6,]   6   12   18   24   30   36   42   48   54
## [7,]   7   14   21   28   35   42   49   56   63
## [8,]   8   16   24   32   40   48   56   64   72
## [9,]   9   18   27   36   45   54   63   72   81

#
#outer(1:9, 1:9, "*")
#1:9 %% 1:9

• *      mult_tab          ,           6   8       3
7.

mult_tab[6:8, 3:7]

## [,1] [,2] [,3] [,4] [,5]
## [1,] 18   24   30   36   42
## [2,] 21   28   35   42   49
## [3,] 24   32   40   48   56

• *
           ,   TRUE,
FALSE,   .

mult_tab >= 10

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

• *      mult_tab2,          tab    10       0.
mult_tab2 <- mult_tab
mult_tab2[mult_tab < 10] <- 0
mult_tab2

## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
## [1,] 0   0   0   0   0   0   0   0   0
## [2,] 0   0   0   0   10  12  14  16  18
## [3,] 0   0   0   12  15  18  21  24  27
## [4,] 0   0   12  16  20  24  28  32  36
## [5,] 0   10  15  20  25  30  35  40  45

```

```
## [6,]    0   12   18   24   30   36   42   48   54
## [7,]    0   14   21   28   35   42   49   56   63
## [8,]    0   16   24   32   40   48   56   64   72
## [9,]    0   18   27   36   45   54   63   72   81
```

12.8

```
list1:
list1 = list(numbers = 1:5, letters = letters, logic = TRUE)
list1

## $numbers
## [1] 1 2 3 4 5
##
## $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"
##
## $logic
## [1] TRUE

•           list1.

list1[1]

## $numbers
## [1] 1 2 3 4 5

•           list1

list1[[1]]

## [1] 1 2 3 4 5

list1$numbers

## [1] 1 2 3 4 5

•           list1.

list1[[1]][1]

## [1] 1

•           list2,           list1,           pupa,
— lupa.

list2 = list(pupa = list1, lupa = list1)
list2
```

```

## $pupa
## $pupa$numbers
## [1] 1 2 3 4 5
##
## $pupa$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"
##
## $pupa$logic
## [1] TRUE
##
##
## $lupa
## $lupa$numbers
## [1] 1 2 3 4 5
##
## $lupa$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"
##
## $lupa$logic
## [1] TRUE

• *           list2,   —      ,   —      .
list2[[1]][[2]][3]

## [1] "c"

```

12.9

```

•           data(mtcars)
-           ,
-           . ?mtcars.

data(mtcars)
mtcars

##                                     mpg cyl disp hp drat    wt  qsec vs am gear carb
## Mazda RX4          21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2

```

```

## Merc 230      22.8   4 140.8  95 3.92 3.150 22.90  1  0  4  2
## Merc 280      19.2   6 167.6 123 3.92 3.440 18.30  1  0  4  4
## Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90  1  0  4  4
## Merc 450SE    16.4   8 275.8 180 3.07 4.070 17.40  0  0  3  3
## Merc 450SL    17.3   8 275.8 180 3.07 3.730 17.60  0  0  3  3
## Merc 450SLC   15.2   8 275.8 180 3.07 3.780 18.00  0  0  3  3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0  3  4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0  3  4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0  3  4
## Fiat 128       32.4   4 78.7  66 4.08 2.200 19.47  1  1  4  1
## Honda Civic     30.4   4 75.7  52 4.93 1.615 18.52  1  1  4  2
## Toyota Corolla 33.9   4 71.1  65 4.22 1.835 19.90  1  1  4  1
## Toyota Corona   21.5   4 120.1  97 3.70 2.465 20.01  1  0  3  1
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0  3  2
## AMC Javelin     15.2   8 304.0 150 3.15 3.435 17.30  0  0  3  2
## Camaro Z28      13.3   8 350.0 245 3.73 3.840 15.41  0  0  3  4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0  3  2
## Fiat X1-9        27.3   4 79.0  66 4.08 1.935 18.90  1  1  4  1
## Porsche 914-2    26.0   4 120.3  91 4.43 2.140 16.70  0  1  5  2
## Lotus Europa     30.4   4 95.1 113 3.77 1.513 16.90  1  1  5  2
## Ford Pantera L   15.8   8 351.0 264 4.22 3.170 14.50  0  1  5  4
## Ferrari Dino     19.7   6 145.0 175 3.62 2.770 15.50  0  1  5  6
## Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1  5  8
## Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60  1  1  4  2

•           mtcars      str().

str(mtcars)

## 'data.frame': 32 obs. of 11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num  16.5 17 18.6 19.4 17 ...
## $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
## $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num  4 4 1 1 2 1 4 2 2 4 ...

•           mtcars.

mtcars[3, 4]

## [1] 93

•           mtcars.

```

```

mtcars[1:6, 1:6]

##                      mpg cyl disp  hp drat    wt
## Mazda RX4        21.0   6 160 110 3.90 2.620
## Mazda RX4 Wag    21.0   6 160 110 3.90 2.875
## Datsun 710       22.8   4 108  93 3.85 2.320
## Hornet 4 Drive   21.4   6 258 110 3.08 3.215
## Hornet Sportabout 18.7   8 360 175 3.15 3.440
## Valiant          18.1   6 225 105 2.76 3.460

•           wt      mtcars -
mtcars$wt

## [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440 4.070
## [13] 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520 3.435 3.840
## [25] 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780

•           mtcars          : hp, mpg, cyl.
mtcars[, c("hp", "mpg", "cyl")]

##                      hp mpg cyl
## Mazda RX4        110 21.0   6
## Mazda RX4 Wag    110 21.0   6
## Datsun 710       93 22.8   4
## Hornet 4 Drive   110 21.4   6
## Hornet Sportabout 175 18.7   8
## Valiant          105 18.1   6
## Duster 360       245 14.3   8
## Merc 240D        62 24.4   4
## Merc 230          95 22.8   4
## Merc 280          123 19.2   6
## Merc 280C         123 17.8   6
## Merc 450SE        180 16.4   8
## Merc 450SL        180 17.3   8
## Merc 450SLC       180 15.2   8
## Cadillac Fleetwood 205 10.4   8
## Lincoln Continental 215 10.4   8
## Chrysler Imperial  230 14.7   8
## Fiat 128           66 32.4   4
## Honda Civic         52 30.4   4
## Toyota Corolla     65 33.9   4
## Toyota Corona       97 21.5   4
## Dodge Challenger    150 15.5   8
## AMC Javelin         150 15.2   8
## Camaro Z28          245 13.3   8
## Pontiac Firebird    175 19.2   8

```

```

## Fiat X1-9      66 27.3   4
## Porsche 914-2  91 26.0   4
## Lotus Europa   113 30.4   4
## Ford Pantera L 264 15.8   8
## Ferrari Dino   175 19.7   6
## Maserati Bora   335 15.0   8
## Volvo 142E     109 21.4   4
•          4      (cyl)      mtcars.

sum(mtcars$cyl == 4)

## [1] 11
•          4      (cyl)      mtcars.

mean(mtcars$cyl == 4)

## [1] 0.34375
•          100      (hp)      mtcars.

mtcars[mtcars$hp >= 100, ]

##          mpg cyl disp hp drat wt qsec vs am gear carb
## Mazda RX4    21.0   6 160.0 110 3.90 2.620 16.46 0 1 4 4
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02 0 1 4 4
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44 1 0 3 1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02 0 0 3 2
## Valiant      18.1   6 225.0 105 2.76 3.460 20.22 1 0 3 1
## Duster 360    14.3   8 360.0 245 3.21 3.570 15.84 0 0 3 4
## Merc 280      19.2   6 167.6 123 3.92 3.440 18.30 1 0 4 4
## Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90 1 0 4 4
## Merc 450SE     16.4   8 275.8 180 3.07 4.070 17.40 0 0 3 3
## Merc 450SL     17.3   8 275.8 180 3.07 3.730 17.60 0 0 3 3
## Merc 450SLC    15.2   8 275.8 180 3.07 3.780 18.00 0 0 3 3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98 0 0 3 4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82 0 0 3 4
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42 0 0 3 4
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87 0 0 3 2
## AMC Javelin     15.2   8 304.0 150 3.15 3.435 17.30 0 0 3 2
## Camaro Z28      13.3   8 350.0 245 3.73 3.840 15.41 0 0 3 4
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05 0 0 3 2
## Lotus Europa     30.4   4  95.1 113 3.77 1.513 16.90 1 1 5 2
## Ford Pantera L   15.8   8 351.0 264 4.22 3.170 14.50 0 1 5 4
## Ferrari Dino     19.7   6 145.0 175 3.62 2.770 15.50 0 1 5 6
## Maserati Bora    15.0   8 301.0 335 3.54 3.570 14.60 0 1 5 8
## Volvo 142E       21.4   4 121.0 109 4.11 2.780 18.60 1 1 4 2
•          100      (hp)  4      (cyl)

```

```

mtcars.

mtcars[mtcars$hp >= 100 & mtcars$cyl == 4, ]

##          mpg cyl disp hp drat    wt qsec vs am gear carb
## Lotus Europa 30.4   4 95.1 113 3.77 1.513 16.9   1   1     5     2
## Volvo 142E   21.4   4 121.0 109 4.11 2.780 18.6   1   1     4     2
•           (wt)      ,           max():

max(mtcars$wt)

## [1] 5.424

•           (wt)      ,           min():

min(mtcars$wt)

## [1] 1.513

•           mtcars      .

mtcars[mtcars$wt == min(mtcars$wt), ]

##          mpg cyl disp hp drat    wt qsec vs am gear carb
## Lotus Europa 30.4   4 95.1 113 3.77 1.513 16.9   1   1     5     2
•           mtcars      ,
mtcars[mtcars$wt < mean(mtcars$wt), ]

##          mpg cyl disp hp drat    wt qsec vs am gear carb
## Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46   0   1     4     4
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02   0   1     4     4
## Datsun 710    22.8   4 108.0  93 3.85 2.320 18.61   1   1     4     1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44   1   0     3     1
## Merc 240D     24.4   4 146.7  62 3.69 3.190 20.00   1   0     4     2
## Merc 230      22.8   4 140.8  95 3.92 3.150 22.90   1   0     4     2
## Fiat 128      32.4   4  78.7  66 4.08 2.200 19.47   1   1     4     1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52   1   1     4     2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90   1   1     4     1
## Toyota Corona   21.5   4 120.1  97 3.70 2.465 20.01   1   0     3     1
## Fiat X1-9       27.3   4  79.0  66 4.08 1.935 18.90   1   1     4     1
## Porsche 914-2    26.0   4 120.3  91 4.43 2.140 16.70   0   1     5     2
## Lotus Europa    30.4   4 95.1 113 3.77 1.513 16.90   1   1     5     2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.50   0   1     5     4
## Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.50   0   1     5     6
## Volvo 142E      21.4   4 121.0 109 4.11 2.780 18.60   1   1     4     2
•           .           wt_kg
round().

1      = 0.45359237 .

```

```

mtcars$wt_kg <- round(mtcars$wt * 1000 * 0.45359237)
mtcars

##          mpg cyl  disp  hp drat    wt  qsec vs am gear carb wt_kg
## Mazda RX4     21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4 1188
## Mazda RX4 Wag 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4 1304
## Datsun 710    22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1 1052
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1 1458
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2 1560
## Valiant       18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1 1569
## Duster 360    14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4 1619
## Merc 240D     24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2 1447
## Merc 230      22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2 1429
## Merc 280      19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4 1560
## Merc 280C     17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4 1560
## Merc 450SE    16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3 1846
## Merc 450SL    17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3 1692
## Merc 450SLC   15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3 1715
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4 2381
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4 2460
## Chrysler Imperial 14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4 2424
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1  998
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2  733
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1  832
## Toyota Corona   21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1 1118
## Dodge Challenger 15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2 1597
## AMC Javelin    15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2 1558
## Camaro Z28     13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4 1742
## Pontiac Firebird 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2 1744
## Fiat X1-9       27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1  878
## Porsche 914-2   26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2  971
## Lotus Europa    30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2  686
## Ford Pantera L  15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4 1438
## Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6 1256
## Maserati Bora    15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8 1619
## Volvo 142E      21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2 1261

```

12.10

- vec5:

```
vec5 <- c(5, 20, 30, 0, 2, 9)
```

- , 10 vec5 “ ”,

```

ifelse(vec5 > 10, "      ", "      ")
## [1] "      "
## [5] "      "
•           heroes_information.csv      heroes.

heroes <- read.csv("data/heroes_information.csv",
                    stringsAsFactors = FALSE,
                    na.strings = c("-", "-99"))

•           hair   heroes,          "Bold"      ,
Hair.color    "No Hair",      "Hairy"      .
heroes$hair <- ifelse(heroes$Hair.color == "No Hair", "Bold", "Hairy")
head(heroes)

##   X      name Gender Eye.color      Race Hair.color Height
## 1 0     A-Bomb  Male   yellow     Human  No Hair   203
## 2 1     Abe Sapien  Male     blue  Ichtyo Sapien  No Hair   191
## 3 2     Abin Sur  Male     blue     Ungaran  No Hair   185
## 4 3 Abomination  Male   green Human / Radiation  No Hair   203
## 5 4     Abraxas  Male     blue   Cosmic Entity Black    NA
## 6 5 Absorbing Man  Male     blue     Human  No Hair   193
##           Publisher Skin.color Alignment Weight hair
## 1     Marvel Comics <NA>      good    441 Bold
## 2 Dark Horse Comics     blue      good     65 Bold
## 3     DC Comics       red      good     90 Bold
## 4     Marvel Comics <NA>      bad    441 Bold
## 5     Marvel Comics <NA>      bad     NA Hairy
## 6     Marvel Comics <NA>      bad    122 Bold

•           tall   heroes,          "tall"      ,
Height      190,      "short"      ,
Height      170,      "middle"      .
# heroes$tall <- dplyr::case_when(
#   heroes$Height > 190 ~ "tall",
#   heroes$Height < 170 ~ "short",
#   TRUE ~ "middle"
# )
heroes$tall <- ifelse(heroes$Height > 190,
                      "tall",
                      ifelse(heroes$Height < 170,
                             "short",
                             "middle"))

```

12.11

```

•           plus_one(),
plus_one <- function(x) x + 1

•           plus_one()      41.
plus_one(41)

## [1] 42

•           circle_area,
circle_area <- function(r) pi * r ^ 2

•           5.
circle_area(5)

## [1] 78.53982

•           cels2fahr(),
cels2fahr <- function(x) x * 9 / 5 + 32

•           -100, -40  0,      cels2fahr()
cels2fahr(c(-100, -40, 0))

## [1] -148  -40   32

•           highlight(),
highlight <- function(x) c("!!!", x, "!!!")

highlight(c(" ", " !"))

## [1] "!!!"    " "      " !"  "!!!"

•           highlight      .,           wrapper =,
"!!!".      wrapper =
highlight <- function(x, wrapper = "!!!") c(wrapper, x, wrapper)

•           c(" ", " !").
highlight(c(" ", " !"))

## [1] "!!!"    " "      " !"  "!!!"

highlight(c(" ", " !"), wrapper = "___")

## [1] "___"    " "      " !"  "___"

```

```

•           trim(),
).
trim <- function(x) x[c(-1, -length(x))]

•           ,           trim()           :
trim(1:7)

## [1] 2 3 4 5 6
trim(letters)

## [1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [20] "u" "v" "w" "x" "y"

•           trim()           n =           1.

trim <- function(x, n = 1) x[c(-1:-n, (-length(x)+n-1):-length(x))]

•           :
trim(letters)

## [1] "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t"
## [20] "u" "v" "w" "x" "y"
trim(letters, n = 2)

## [1] "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s" "t" "u"
## [20] "v" "w" "x"

•           ,           trim()           n = 0, ..

trim <- function(x, n = 1) {
  if (n == 0) return(x)
  x[c(-1:-n, (-length(x)+n-1):-length(x))]
}

trim(letters, n = 0)

## [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"

•           *           :           trim()           ,           n
=           n =           :           :
trim <- function(x, n = 1) {
  if (n < 0) stop("n           !")
  l <- length(x)
  if (n > ceiling(l/2) - 1) stop("n           !")
}

```

```

if (n == 0) return(x)
x[c(-1:-n, -(l+n-1):-1)]
}

• *
      trim():

trim(1:6, 3)

## Error in trim(1:6, 3): n          !
trim(1:6, -1)

## Error in trim(1:6, -1): n          !
•           na_n(),                  NA
na_n <- function(x) sum(is.na(x))

•           na_n()      :
na_n(c(NA, 3:5, NA, 2, NA))

## [1] 3
•           factors(),
: %%
factors <- function(x) (1:x)[x %% (1:x) == 0]

•           factors()
: 
factors(3)

## [1] 1 3
factors(161)

## [1] 1   7  23 161
factors(1984)

## [1] 1   2   4   8   16  31  32  62  64  124 248 496 992 1984
• *           is_prime(),          ,
any() -           TRUE,
TRUE.

is_prime <- function(x) !any(x%%(2:(x-1)) == 0)
#is_prime <- function(x) length(factors(x)) == 2 #           factors()

•           ,           :
is_prime(2017)

```

```

## [1] TRUE
is_prime(2019)

## [1] FALSE
2019/3 #2019      3

## [1] 673
is_prime(2020)

## [1] FALSE
is_prime(2021)

## [1] FALSE
• *           monotonic(),          TRUE,          (
-                   )
— diff() —
monotonic <- function(x) all(diff(x)>=0) | all(diff(x)<=0)

monotonic(1:7)

## [1] TRUE
monotonic(c(1:5,5:1))

## [1] FALSE
monotonic(6:-1)

## [1] TRUE
monotonic(c(1:5, rep(5, 10), 5:10))

## [1] TRUE
+   %in%
!
—
:   ,
TRUE,
`%notin%` <- function(x, y) ! (x %in% y)
1:10 %notin% c(1, 4, 5)

## [1] FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
• *           %without%,
```

```
%> %without% <- function(x, y) x[!x %in% y]
%> c(" ", " ", " ", " ", " ", " ") %without% c(" ", " ")
## [1] " "      " "      " "      " "
• *           %between%,          TRUE,
:
%> %between% <- function(x, y) x >= min(y) & x <= max(y)
%> 1:10 %between% c(1, 4, 5)
## [1] TRUE TRUE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE
```

12.12**apply()**

- M2:

```
M2 <- matrix(c(20:11, 11:20), nrow = 5)
M2
```

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

- M2

```
apply(M2, 1, max)
```

```
## [1] 20 19 18 19 20
```

- M2

```
apply(M2, 2, max)
```

```
## [1] 20 15 15 20
```

- M2

```
apply(M2, 1, mean)
```

```
## [1] 15.5 15.5 15.5 15.5 15.5
```

- M2

```
apply(M2, 2, mean)
```

```

## [1] 18 13 13 18

•           list3:

list3 <- list(
  a = 1:5,
  b = 0:20,
  c = 4:24,
  d = 6:3,
  e = 6:25
)

•           list3.

sapply(list3, max)

##   a   b   c   d   e
##  5 20 24  6 25

•           list3.

sapply(list3, sum)

##   a   b   c   d   e
## 15 210 294 18 310

•           list3.

sapply(list3, length)

##   a   b   c   d   e
##  5 21 21  4 20

•           max_item(), , - ( )

•           which.max(),
( , )..

max_item <- function (x) x[[which.max(sapply(x, length))]]

•           max_item()      list3.

max_item(list3)

## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

•           list4:

list4 <- list(1:3, 3:40, list3)

•           , .. .
,
```

```
rapply(): recursive lapply

rapply(list4, length, how = "list")

## [[1]]
## [1] 3
##
## [[2]]
## [1] 38
##
## [[3]]
## [[3]]$a
## [1] 5
##
## [[3]]$b
## [1] 21
##
## [[3]]$c
## [1] 21
##
## [[3]]$d
## [1] 4
##
## [[3]]$e
## [1] 20

• *           heroes      ,      NA
                           na_n().

sapply(heroes, na_n)

##          X      name   Gender Eye.color      Race Hair.color     Height
##          0        0       29      172       304       172       217
## Publisher Skin.color Alignment    Weight      hair      tall
##          0       662        7      239       172       217

• *           is_prime(),      prime_numbers(),
.
.
is_prime <- function(x) !any(x %% (2:(x - 1)) == 0)
prime_numbers <- function(x) (2:x)[sapply(2:x, is_prime)]

prime_numbers(200)

## [1]  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71
## [20] 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167
## [39] 173 179 181 191 193 197 199
```

```
library(tidyverse)
heroes <- read_csv("data/heroes_information.csv",
                     na = c("-", "-99"))
powers <- read_csv("data/super_hero_powers.csv")
```

12.13 magrittr::%>%

```
• , %>%
sqrt(sum(1:10))

## [1] 7.416198
1:10 %>%
sum() %>%
sqrt()

## [1] 7.416198
abs(min(-5:5))

## [1] 5
-5:5 %>%
min() %>%
abs()

## [1] 5
c(" ", 2, " ", sqrt(2))

## [1] " " "2" " " "1.4142135623731"
2 %>% c(" ", ., " ", sqrt(.))

## [1] " " "2" " " "1.4142135623731"
```

12.14 : dplyr::slice() dplyr::filter()

```
• , 500 .
heroes %>%
filter(Weight > 500)

## # A tibble: 6 x 11
##       X1 name  Gender `Eye color` `Race` `Hair color` Height Publisher
##       <dbl> <chr> <chr>   <chr>    <chr>    <chr>      <dbl> <chr>
```

```

## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
• , - 500 .
heroes %>%
  filter(Weight > 500 & Gender == "Female")

## # A tibble: 1 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 283 Giga~ Female green      <NA> Red           62.5 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>
• , ("Human")
5.

heroes %>%
  filter(Race == "Human" & Gender == "Female") %>%
  slice(1:5)

## # A tibble: 5 x 11
##   X1 name Gender `Eye color` Race `Hair color` Height Publisher
##   <dbl> <chr> <chr> <chr> <chr> <chr> <dbl> <chr>
## 1 38 Arac~ Female blue      Human Blond       175 Marvel C-
## 2 63 Batg~ Female green     Human Red        170 DC Comics
## 3 65 Batg~ Female green     Human Black      165 DC Comics
## 4 72 Batw~ Female green     Human Red        178 DC Comics
## 5 96 Blac~ Female blue      Human Blond       165 DC Comics
## # ... with 3 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>

```

12.15 : dplyr::select()

- 4 powers.

```

powers %>%
  select(1:4)

## # A tibble: 667 x 4
##   hero_names    Agility `Accelerated Healing` `Lantern Power Ring`
##   <chr>          <lgl>   <lgl>            <lgl>
## 1 3-D Man      TRUE    FALSE             FALSE

```

```

### 2 A-Bomb      FALSE  TRUE      FALSE
### 3 Abe Sapien  TRUE   TRUE      FALSE
### 4 Abin Sur    FALSE  FALSE     TRUE
### 5 Abomination FALSE  TRUE      FALSE
### 6 Abraxas     FALSE  FALSE     FALSE
### 7 Absorbing Man FALSE FALSE     FALSE
### 8 Adam Monroe FALSE  TRUE      FALSE
### 9 Adam Strange FALSE FALSE     FALSE
### 10 Agent Bob   FALSE FALSE     FALSE
### # ... with 657 more rows
•             Reflexes  Empathy      powers:
powers %>%
  select(Reflexes:Empathy)

### # A tibble: 667 x 7
###   Reflexes Invulnerability `Energy Constru~ `Force Fields` `Self-Sustenanc~
###   <lgl>     <lgl>          <lgl>        <lgl>        <lgl>
###   1 FALSE    FALSE          FALSE        FALSE        FALSE
###   2 FALSE    FALSE          FALSE        FALSE        TRUE
###   3 TRUE     FALSE          FALSE        FALSE        FALSE
###   4 FALSE    FALSE          FALSE        FALSE        FALSE
###   5 FALSE    TRUE           FALSE        FALSE        FALSE
###   6 FALSE    TRUE           FALSE        FALSE        FALSE
###   7 FALSE    TRUE           FALSE        FALSE        FALSE
###   8 FALSE    FALSE          FALSE        FALSE        FALSE
###   9 FALSE    FALSE          FALSE        FALSE        FALSE
###  10 FALSE   FALSE          FALSE        FALSE        FALSE
### # ... with 657 more rows, and 2 more variables: `Anti-Gravity` <lgl>,
### #   Empathy <lgl>
•             powers      (hero_names):
powers %>%
  select(!hero_names)

### # A tibble: 667 x 167
###   Agility `Accelerated He~ `Lantern Power ~ `Dimensional Aw~ `Cold Resistanc~
###   <lgl>    <lgl>          <lgl>        <lgl>        <lgl>
###   1 TRUE    FALSE          FALSE        FALSE        FALSE
###   2 FALSE   TRUE           FALSE        FALSE        FALSE
###   3 TRUE    TRUE           FALSE        FALSE        TRUE
###   4 FALSE   FALSE          TRUE         FALSE        FALSE
###   5 FALSE   TRUE           FALSE        FALSE        FALSE
###   6 FALSE   FALSE          FALSE        TRUE         FALSE
###   7 FALSE   FALSE          FALSE        FALSE        TRUE
###   8 FALSE   TRUE           FALSE        FALSE        FALSE

```

```

##  9 FALSE  FALSE      FALSE      FALSE      FALSE
## 10 FALSE  FALSE      FALSE      FALSE      FALSE
## # ... with 657 more rows, and 162 more variables: Durability <lgl>,
## #   Stealth <lgl>, `Energy Absorption` <lgl>, Flight <lgl>, `Danger
## #   Sense` <lgl>, `Underwater breathing` <lgl>, Marksmanship <lgl>, `Weapons
## #   Master` <lgl>, `Power Augmentation` <lgl>, `Animal Attributes` <lgl>,
## #   Longevity <lgl>, Intelligence <lgl>, `Super Strength` <lgl>,
## #   Cryokinesis <lgl>, Telepathy <lgl>, `Energy Armor` <lgl>, `Energy
## #   Blasts` <lgl>, Duplication <lgl>, `Size Changing` <lgl>, `Density
## #   Control` <lgl>, Stamina <lgl>, `Astral Travel` <lgl>, `Audio
## #   Control` <lgl>, Dexterity <lgl>, Omnitrix <lgl>, `Super Speed` <lgl>,
## #   Possession <lgl>, `Animal Oriented Powers` <lgl>, `Weapon-based
## #   Powers` <lgl>, Electrokinesis <lgl>, `Darkforce Manipulation` <lgl>, `Death
## #   Touch` <lgl>, Teleportation <lgl>, `Enhanced Senses` <lgl>,
## #   Telekinesis <lgl>, `Energy Beams` <lgl>, Magic <lgl>, Hyperkinesis <lgl>,
## #   Jump <lgl>, Clairvoyance <lgl>, `Dimensional Travel` <lgl>, `Power
## #   Sense` <lgl>, Shapeshifting <lgl>, `Peak Human Condition` <lgl>,
## #   Immortality <lgl>, Camouflage <lgl>, `Element Control` <lgl>,
## #   Phasing <lgl>, `Astral Projection` <lgl>, `Electrical Transport` <lgl>,
## #   `Fire Control` <lgl>, Projection <lgl>, Summoning <lgl>, `Enhanced
## #   Memory` <lgl>, Reflexes <lgl>, Invulnerability <lgl>, `Energy
## #   Constructs` <lgl>, `Force Fields` <lgl>, `Self-Sustenance` <lgl>,
## #   `Anti-Gravity` <lgl>, Empathy <lgl>, `Power Nullifier` <lgl>, `Radiation
## #   Control` <lgl>, `Psionic Powers` <lgl>, Elasticity <lgl>, `Substance
## #   Secretion` <lgl>, `Elemental Transmogrification` <lgl>,
## #   `Technopath/Cyberpath` <lgl>, `Photographic Reflexes` <lgl>, `Seismic
## #   Power` <lgl>, Animation <lgl>, Precognition <lgl>, `Mind Control` <lgl>,
## #   `Fire Resistance` <lgl>, `Power Absorption` <lgl>, `Enhanced
## #   Hearing` <lgl>, `Nova Force` <lgl>, Insanity <lgl>, Hypnokinesis <lgl>,
## #   `Animal Control` <lgl>, `Natural Armor` <lgl>, Intangibility <lgl>,
## #   `Enhanced Sight` <lgl>, `Molecular Manipulation` <lgl>, `Heat
## #   Generation` <lgl>, Adaptation <lgl>, Gliding <lgl>, `Power Suit` <lgl>,
## #   `Mind Blast` <lgl>, `Probability Manipulation` <lgl>, `Gravity
## #   Control` <lgl>, Regeneration <lgl>, `Light Control` <lgl>,
## #   Echolocation <lgl>, Levitation <lgl>, `Toxin and Disease Control` <lgl>,
## #   Banish <lgl>, `Energy Manipulation` <lgl>, `Heat Resistance` <lgl>,
## #   `Natural Weapons` <lgl>, ...

```

12.16

: dplyr::arrange()

- heroes name, Gender, Height
Height.

```

heroes %>%
  select(name, Gender, Height) %>%
  arrange(Height)

## # A tibble: 734 x 3
##   name      Gender Height
##   <chr>     <chr>   <dbl>
## 1 Utgard-Loki  Male    15.2
## 2 Bloodwraith  Male    30.5
## 3 King Kong    Male    30.5
## 4 Anti-Monitor Male    61
## 5 Giganta     Female  62.5
## 6 Krypto       Male    64
## 7 Yoda         Male    66
## 8 Jack-Jack    Male    71
## 9 Howard the Duck Male    79
## 10 Godzilla    <NA>   108
## # ... with 724 more rows

•           heroes      name, Gender, Height
             Height.

heroes %>%
  select(name, Gender, Height) %>%
  arrange(desc(Height))

## # A tibble: 734 x 3
##   name      Gender Height
##   <chr>     <chr>   <dbl>
## 1 Fin Fang Foom  Male    975
## 2 Galactus     Male    876
## 3 Groot        Male    701
## 4 MODOK        Male    366
## 5 Wolfsbane    Female  366
## 6 Onslaught    Male    305
## 7 Sasquatch    Male    305
## 8 Ymir         Male    305.
## 9 Rey          Female  297
## 10 Juggernaut   Male    287
## # ... with 724 more rows

•           heroes      name, Gender, Height
             Gender,           Height.

heroes %>%
  select(name, Gender, Height) %>%
  arrange(Gender, desc(Height))

```

```
## # A tibble: 734 x 3
##   name     Gender Height
##   <chr>    <chr>   <dbl>
## 1 Wolfsbane Female   366
## 2 Rey       Female   297
## 3 Bloodaxe Female   218
## 4 Thundra  Female   218
## 5 Hela      Female   213
## 6 Frenzy    Female   211
## 7 She-Hulk  Female   201
## 8 Ardina    Female   193
## 9 Starfire  Female   193
## 10 Valkyrie Female   191
## # ... with 724 more rows
```

12.17 : dplyr::distinct()

- Eye color heroes.

```
heroes %>%
  distinct(`Eye color`)

## # A tibble: 23 x 1
##   `Eye color`
##   <chr>
## 1 yellow
## 2 blue
## 3 green
## 4 brown
## 5 <NA>
## 6 red
## 7 violet
## 8 white
## 9 purple
## 10 black
## # ... with 13 more rows
```

- Hair color heroes.

```
heroes %>%
  distinct(`Hair color`)

## # A tibble: 30 x 1
##   `Hair color`
##   <chr>
## 1 No Hair
```

```
##  2 Black
##  3 Blond
##  4 Brown
##  5 <NA>
##  6 White
##  7 Purple
##  8 Orange
##  9 Pink
## 10 Red
## # ... with 20 more rows
```

12.18 : dplyr::mutate() dplyr::transmute()

- `height_m`, `name`

```
height_m.
heroes %>%
  mutate(height_m = Height/100) %>%
  select(name, height_m)
```

`## # A tibble: 734 x 2`

##	name height_m
##	<chr> <dbl>
##	1 A-Bomb 2.03
##	2 Abe Sapien 1.91
##	3 Abin Sur 1.85
##	4 Abomination 2.03
##	5 Abraxas NA
##	6 Absorbing Man 1.93
##	7 Adam Monroe NA
##	8 Adam Strange 1.85
##	9 Agent 13 1.73
##	10 Agent Bob 1.78
## #	... with 724 more rows

- `hair heroes, "Bold"`, `Hair.color "No Hair", "Hairy"`, `name, Hair color, hair`.

```
heroes %>%
  mutate(hair = ifelse(`Hair color` == "No Hair", "Bold", "Hairy")) %>%
  select(name, `Hair color`, hair)
```

`## # A tibble: 734 x 3`

##	name `Hair color`	hair
##	<chr>	<chr>

```

## 1 A-Bomb      No Hair      Bold
## 2 Abe Sapien   No Hair      Bold
## 3 Abin Sur     No Hair      Bold
## 4 Abomination  No Hair      Bold
## 5 Abraxas       Black       Hairy
## 6 Absorbing Man No Hair      Bold
## 7 Adam Monroe   Blond       Hairy
## 8 Adam Strange   Blond       Hairy
## 9 Agent 13      Blond       Hairy
## 10 Agent Bob    Brown       Hairy
## # ... with 724 more rows

```

12.19 : dplyr::group_by() %>% summarise()

```

• . 5 .

heroes %>%
  count(Race, sort = TRUE) %>%
  slice(1:5)

## # A tibble: 5 x 2
##   Race          n
##   <chr>        <int>
## 1 <NA>         304
## 2 Human        208
## 3 Mutant       63
## 4 God / Eternal 14
## 5 Cyborg       11
• .

heroes %>%
  group_by(Gender) %>%
  summarise(height_mean = mean(Height, na.rm = TRUE))

## # A tibble: 3 x 2
##   Gender height_mean
##   <chr>        <dbl>
## 1 Female       175.
## 2 Male         192.
## 3 <NA>         177.

```

12.20 : across()

• NA , (Gender).

```

na_n <- function(x) sum(is.na(x))
heroes %>%
  group_by(Gender) %>%
  summarise(across(everything(), na_n))

## # A tibble: 3 x 11
##   Gender     X1 name `Eye color` `Race` `Hair color` Height Publisher
##   <chr>    <int> <int>      <int> <int>      <int> <int> <int>
## 1 Female     0     0         41     98        38     56       0
## 2 Male       0     0        121    184       123    147       0
## 3 <NA>       0     0         10     22        11     14       0
## # ... with 3 more variables: `Skin color` <int>, Alignment <int>, Weight <int>
•           NA , "color",
(Gender).

na_n <- function(x) sum(is.na(x))
heroes %>%
  group_by(Gender) %>%
  summarise(across(ends_with("color"), na_n))

## # A tibble: 3 x 4
##   Gender `Eye color` `Hair color` `Skin color`
##   <chr>     <int>      <int>      <int>
## 1 Female      41        38       186
## 2 Male       121       123       449
## 3 <NA>       10        11        27
•           heroes name, Height  Weight,
" " , " " , ",

higher_than_average <- function(x) ifelse(x > mean(x, na.rm = TRUE),
                                         " " , " ")
                                         " " , " ")

heroes %>%
  transmute(name,
            across(c(Height, Weight),
                   higher_than_average))

## # A tibble: 734 x 3
##   name      Height      Weight
##   <chr>      <chr>      <chr>
## 1 A-Bomb
## 2 Abe Sapien
## 3 Abin Sur
## 4 Abomination
## 5 Abraxas      <NA>      <NA>

```

```

## 6 Absorbing Man
## 7 Adam Monroe <NA> <NA>
## 8 Adam Strange
## 9 Agent 13
## 10 Agent Bob
## # ... with 724 more rows

•      heroes      Gender, name, Height  Weight,
      "      ",      "
      ",      .

heroes %>%
  group_by(Gender) %>%
  transmute(name,
            across(c(Height, Weight),
                    higher_than_average))

## # A tibble: 734 x 4
## # Groups:   Gender [3]
##   Gender name      Height      Weight
##   <chr>  <chr>     <chr>     <chr>
## 1 Male   A-Bomb
## 2 Male   Abe Sapien
## 3 Male   Abin Sur
## 4 Male   Abomination
## 5 Male   Abraxas    <NA>      <NA>
## 6 Male   Absorbing Man
## 7 Male   Adam Monroe <NA>      <NA>
## 8 Male   Adam Strange
## 9 Female Agent 13
## 10 Male  Agent Bob
## # ... with 724 more rows

```

12.21

: *_join {#solution_join}

```

web_creators, , , . . . TRUE
Web Creation powers.

powers_web <- powers %>%
  select(hero_names, `Web Creation`)
web_creators <- left_join(heroes, powers_web, by = c("name" = "hero_names")) %>%
  filter(`Web Creation`)
web_creators

## # A tibble: 16 x 12
##       X1 name  Gender `Eye color` Race  `Hair color` Height Publisher
##       <dbl> <chr> <chr>   <chr> <chr> <chr>   <dbl> <chr>
```

```

##      <dbl> <chr> <chr> <chr>      <chr> <chr>      <dbl> <chr>
##  1    33 Anti~ Male   blue     Symb~ Blond    229 Marvel C~
##  2    38 Arac~ Female blue   Human Blond  175 Marvel C~
##  3   161 Carn~ Male   green   Symb~ Red     185 Marvel C~
##  4   335 Hybr~ Male   brown   Symb~ Black   175 Marvel C~
##  5   479 Myst~ Male   brown   Human No Hair 180 Marvel C~
##  6   580 Scar~ Male   brown   Clone Brown  193 Marvel C~
##  7   597 Silk  Female brown  Human Black  NA Marvel C~
##  8   620 Spid~ Female blue  Human Brown  170 Marvel C~
##  9   621 Spid~ Female blue  Human Blond  165 Marvel C~
## 10   622 Spid~ Male   hazel  Human Brown  178 Marvel C~
## 11   623 Spid~ <NA>   red   Human Brown  178 Marvel C~
## 12   624 Spid~ Male   brown  Human Black  157 Marvel C~
## 13   673 Toxin  Male   blue   Symb~ Brown  188 Marvel C~
## 14   674 Toxin  Male   black  Symb~ Blond  191 Marvel C~
## 15   689 Venom  Male   blue   Symb~ Strawberry ~ 191 Marvel C~
## 16   692 Veno~ Male   <NA>  Symb~ <NA>  226 Marvel C~

## # ... with 4 more variables: `Skin color` <chr>, Alignment <chr>, Weight <dbl>,
## #   `Web Creation` <lgl>

• , heroes, powers.

anti_join(heroes, powers, by = c("name" = "hero_names")) %>%
  pull(name)

## [1] "Agent 13"          "Alfred Pennyworth" "Arsenal"
## [4] "Batgirl III"       "Batgirl V"        "Beetle"
## [7] "Black Goliath"     "Black Widow II"   "Blaquesmith"
## [10] "Bolt"              "Boomer"          "Box"
## [13] "Box III"           "Captain Mar-vell" "Cat II"
## [16] "Cecilia Reyes"    "Clea"             "Clock King"
## [19] "Colin Wagner"     "Colossal Boy"    "Corsair"
## [22] "Cypher"            "Danny Cooper"     "Darkside"
## [25] "ERG-1"              "Fixer"            "Franklin Storm"
## [28] "Giant-Man"         "Giant-Man II"    "Goliath"
## [31] "Goliath"            "Goliath"          "Guardian"
## [34] "Hawkwoman"          "Hawkwoman II"    "Hawkwoman III"
## [37] "Howard the Duck"   "Jack Bauer"       "Jesse Quick"
## [40] "Jessica Sanders"   "Jigsaw"           "Jyn Erso"
## [43] "Kid Flash II"      "Kingpin"          "Meteorite"
## [46] "Mister Zsasz"      "Mogo"             "Moloch"
## [49] "Morph"              "Nite Owl II"      "Omega Red"
## [52] "Paul Blart"         "Penance"          "Penance I"
## [55] "Plastic Lad"        "Power Man"        "Renata Soliz"
## [58] "Ronin"              "Shrinking Violet" "Snake-Eyes"
## [61] "Spider-Carnage"    "Spider-Woman II" "Stacy X"

```

```

## [64] "Thunderbird II"      "Two-Face"           "Vagabond"
## [67] "Vision II"            "Vulcan"             "Warbird"
## [70] "White Queen"          "Wiz Kid"             "Wondra"
## [73] "Wyatt Wingfoot"       "Yellow Claw"

• , powers, heroes.

anti_join(powers, heroes, by = c("hero_names" = "name")) %>%
  pull(hero_names)

## [1] "3-D Man"           "Bananaman"         "Bizarro-Girl"
## [4] "Black Vulcan"       "Blue Streak"        "Bradley"
## [7] "Clayface"           "Concrete"          "Dementor"
## [10] "Doctor Poison"      "Fire"               "Hellgramite"
## [13] "Lara Croft"         "Little Epic"        "Lord Voldemort"
## [16] "Orion"              "Peek-a-Boo"         "Queen Hippolyta"
## [19] "Reactron"           "SHDB"               "Stretch Armstrong"
## [22] "TEST"               "Tommy Clarke"       "Tyrant"

```

12.22 Tidy data

```

• heroes_weight, :

heroes_weight <- heroes %>%
  filter(Publisher %in% c("DC Comics", "Marvel Comics")) %>%
  group_by(Gender, Publisher) %>%
  summarise(weight_mean = mean(Weight, na.rm = TRUE)) %>%
  drop_na()
heroes_weight

## # A tibble: 4 x 3
## # Groups:   Gender [2]
##   Gender Publisher    weight_mean
##   <chr>  <chr>          <dbl>
## 1 Female DC Comics     76.8
## 2 Female Marvel Comics 80.1
## 3 Male   DC Comics     113.
## 4 Male   Marvel Comics 134.

drop_na() , NA.

• heroes_weight, :

heroes_weight %>%
  pivot_wider(names_from = "Publisher", values_from = "weight_mean")

## # A tibble: 2 x 3

```

```

## # Groups:  Gender [2]
##   Gender `DC Comics` `Marvel Comics`
##   <chr>      <dbl>      <dbl>
## 1 Female      76.8       80.1
## 2 Male        113.      134.

•           :

heroes_weight %>%
  pivot_wider(names_from = "Publisher", values_from = "weight_mean") %>%
  pivot_longer(cols = !Gender,
    names_to = "Publisher",
    values_to = "weight_mean")

## # A tibble: 4 x 3
## # Groups:  Gender [2]
##   Gender Publisher weight_mean
##   <chr>  <chr>      <dbl>
## 1 Female DC Comics     76.8
## 2 Female Marvel Comics 80.1
## 3 Male   DC Comics     113.
## 4 Male   Marvel Comics 134.

•           powers          : hero_names, power ( )
  has ( ). 

powers %>%
  pivot_longer(cols = !hero_names,
    names_to = "power",
    values_to = "has")

## # A tibble: 111,389 x 3
##   hero_names power      has
##   <chr>      <chr>      <lgl>
## 1 3-D Man   Agility    TRUE
## 2 3-D Man   Accelerated Healing FALSE
## 3 3-D Man   Lantern Power Ring FALSE
## 4 3-D Man   Dimensional Awareness FALSE
## 5 3-D Man   Cold Resistance FALSE
## 6 3-D Man   Durability   FALSE
## 7 3-D Man   Stealth     FALSE
## 8 3-D Man   Energy Absorption FALSE
## 9 3-D Man   Flight      FALSE
## 10 3-D Man  Danger Sense FALSE
## # ... with 111,379 more rows

•           powers , , :
  - ( ). 

```

```

powers %>%
  pivot_longer(cols = !hero_names,
               names_to = "power",
               values_to = "has") %>%
  pivot_wider(names_from = hero_names,
               values_from = has)

## # A tibble: 167 x 668
##   power `3-D Man` `A-Bomb` `Abe Sapien` `Abin Sur` Abomination Abraxas
##   <chr> <lgl>    <lgl>    <lgl>      <lgl>    <lgl>    <lgl>
## 1 Agil~ TRUE     FALSE    TRUE      FALSE     FALSE    FALSE
## 2 Acce~ FALSE    TRUE     TRUE      FALSE     TRUE     FALSE
## 3 Lant~ FALSE    FALSE    FALSE      TRUE     FALSE    FALSE
## 4 Dime~ FALSE    FALSE    FALSE      FALSE     FALSE    TRUE
## 5 Cold~ FALSE    FALSE    TRUE      FALSE     FALSE    FALSE
## 6 Dura~ FALSE    TRUE     TRUE      FALSE     FALSE    FALSE
## 7 Stea~ FALSE    FALSE    FALSE      FALSE     FALSE    FALSE
## 8 Ener~ FALSE    FALSE    FALSE      FALSE     FALSE    FALSE
## 9 Flig~ FALSE    FALSE    FALSE      FALSE     FALSE    TRUE
## 10 Dang~ FALSE   FALSE    FALSE      FALSE     FALSE    FALSE
## # ... with 157 more rows, and 661 more variables: `Absorbing Man` <lgl>, `Adam
## # Monroe` <lgl>, `Adam Strange` <lgl>, `Agent Bob` <lgl>, `Agent Zero` <lgl>,
## # `Air-Walker` <lgl>, Ajax <lgl>, `Alan Scott` <lgl>, `Alex Mercer` <lgl>,
## # `Alex Woolsly` <lgl>, Alien <lgl>, `Allan Quatermain` <lgl>, Amazo <lgl>,
## # Ammo <lgl>, `Ando Masahashi` <lgl>, Angel <lgl>, `Angel Dust` <lgl>, `Angel
## # Salvadore` <lgl>, Angela <lgl>, `Animal Man` <lgl>, Annihilus <lgl>,
## # `Ant-Man` <lgl>, `Ant-Man II` <lgl>, `Anti-Monitor` <lgl>,
## # `Anti-Spawn` <lgl>, `Anti-Venom` <lgl>, Apocalypse <lgl>, Aquababy <lgl>,
## # Aqualad <lgl>, Aquaman <lgl>, Arachne <lgl>, Archangel <lgl>,
## # Arclight <lgl>, Ardina <lgl>, Ares <lgl>, Ariel <lgl>, Armor <lgl>, `Astro
## # Boy` <lgl>, Atlas <lgl>, Atom <lgl>, `Atom Girl` <lgl>, `Atom II` <lgl>,
## # `Atom III` <lgl>, `Atom IV` <lgl>, Aurora <lgl>, Azazel <lgl>,
## # Azrael <lgl>, Aztar <lgl>, Bananaman <lgl>, Bane <lgl>, Banshee <lgl>,
## # Bantam <lgl>, Batgirl <lgl>, `Batgirl IV` <lgl>, `Batgirl VI` <lgl>,
## # Batman <lgl>, `Batman II` <lgl>, Battlestar <lgl>, `Batwoman V` <lgl>,
## # Beak <lgl>, Beast <lgl>, `Beast Boy` <lgl>, `Ben 10` <lgl>, `Beta Ray
## # Bill` <lgl>, Beyonder <lgl>, `Big Barda` <lgl>, `Big Daddy` <lgl>, `Big
## # Man` <lgl>, `Bill Harken` <lgl>, `Billy Kincaid` <lgl>, Binary <lgl>,
## # `Bionic Woman` <lgl>, `Bird-Brain` <lgl>, `Bird-Man` <lgl>, `Bird-Man
## # II` <lgl>, Birdman <lgl>, Bishop <lgl>, Bizarro <lgl>,
## # `Bizarro-Girl` <lgl>, `Black Abbott` <lgl>, `Black Adam` <lgl>, `Black
## # Bolt` <lgl>, `Black Canary` <lgl>, `Black Cat` <lgl>, `Black Flash` <lgl>,
## # `Black Knight III` <lgl>, `Black Lightning` <lgl>, `Black Mamba` <lgl>,
## # `Black Manta` <lgl>, `Black Panther` <lgl>, `Black Vulcan` <lgl>, `Black
## # Widow` <lgl>, Blackout <lgl>, Blackwing <lgl>, Blackwulf <lgl>,

```

```
## #  Blade <lgl>, `Bling!` <lgl>, Blink <lgl>, Blizzard <lgl>, `Blizzard  
## #  II` <lgl>, ...
```

Bibliography

Adler, J. (2010). *R in a nutshell: A desktop quick reference.* ” O'Reilly Media, Inc.”.