jinha

8/8/23

Table of contents

. . , .

, , , , . , , .
.

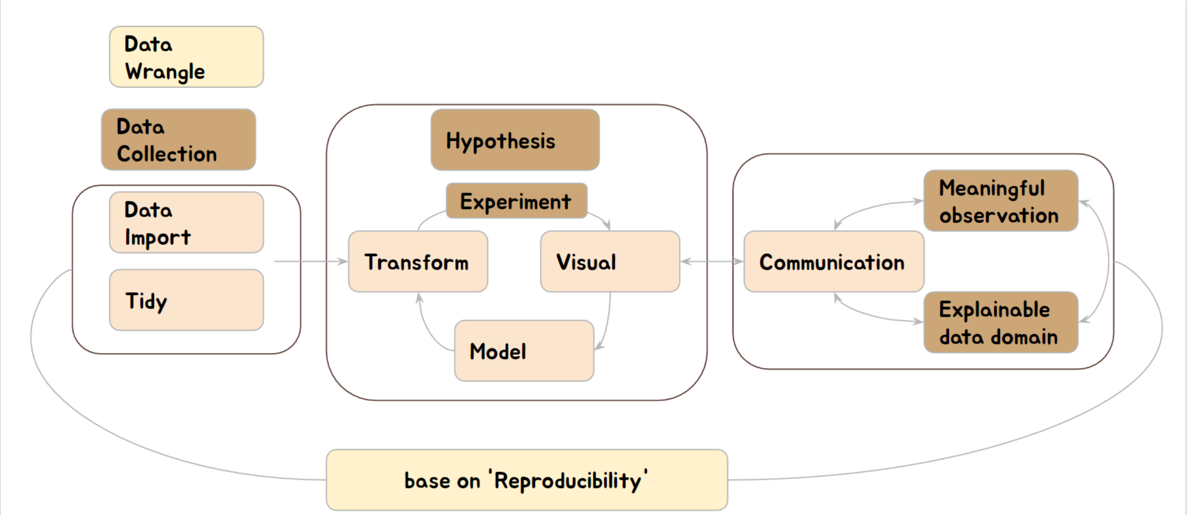
:

- :
  - , , , . .
- :
  - , .
- :
  - . .
- :
  - , .
- :
  - , . , .

,

. .

- :
  - , . , .
- :
  - , , .
- :
  - , , .
- :
  - .
- :
  - , .
- :
  - , . .
- :
  - . . .


Data Wrangle
Data Collection
Data Import
Tidy
Hypothesis
Experiment
Transform
Visual
Model
Communication
Meaningful observation
Explainable data domain
base on 'Reproducibility'

R, Rstudio, markdown, Shiny server, PostgreSQL github . R , . Rafael A. Irizarry Hadley Wickham . .

books

title	authos	url
Introduction to Data Science with R	Rafael A. Irizarry	https://rafalab.github.io/dsbook/
R for Data Science	Garrett Grolemund, Hadley Wickham	https://r4ds.had.co.nz/index.html

- I hope you will get valuable experience with me.

  .!!! **jinha**

Part I

I. R

1 R & R studio

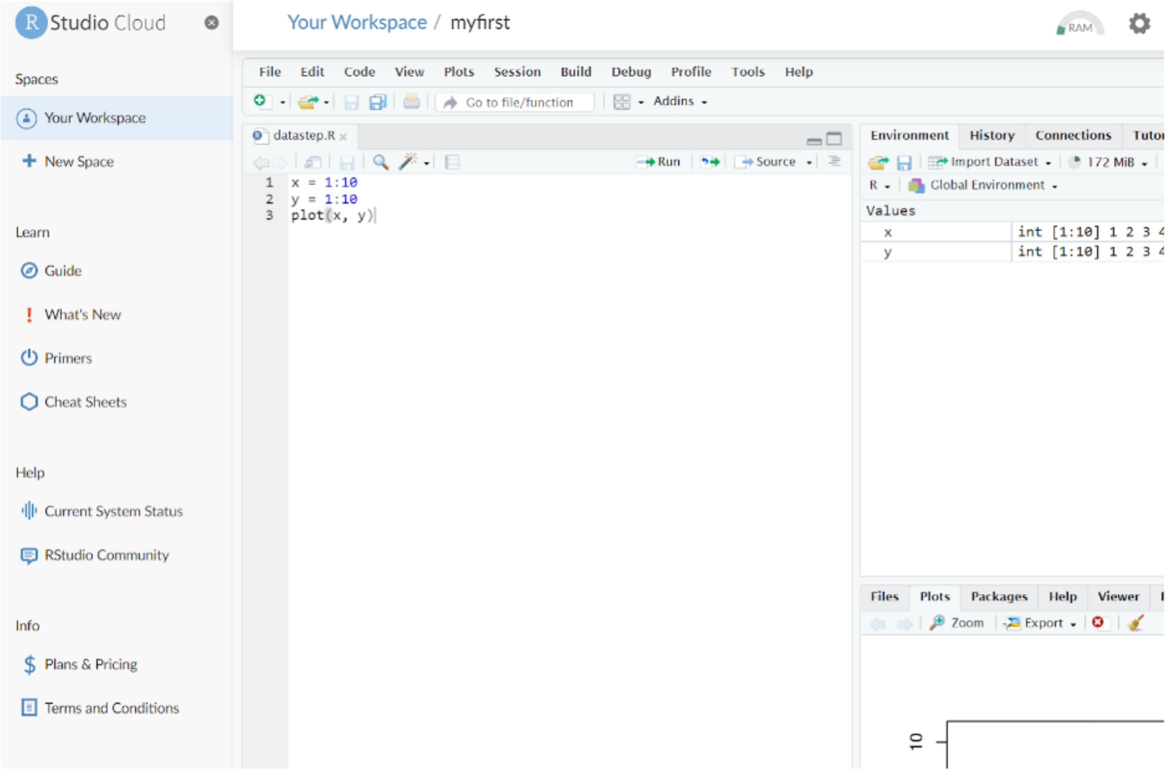
R R studio . rstudio cloud .

1.1 Rstudio Cloud

R studio cloud R R studio . , .

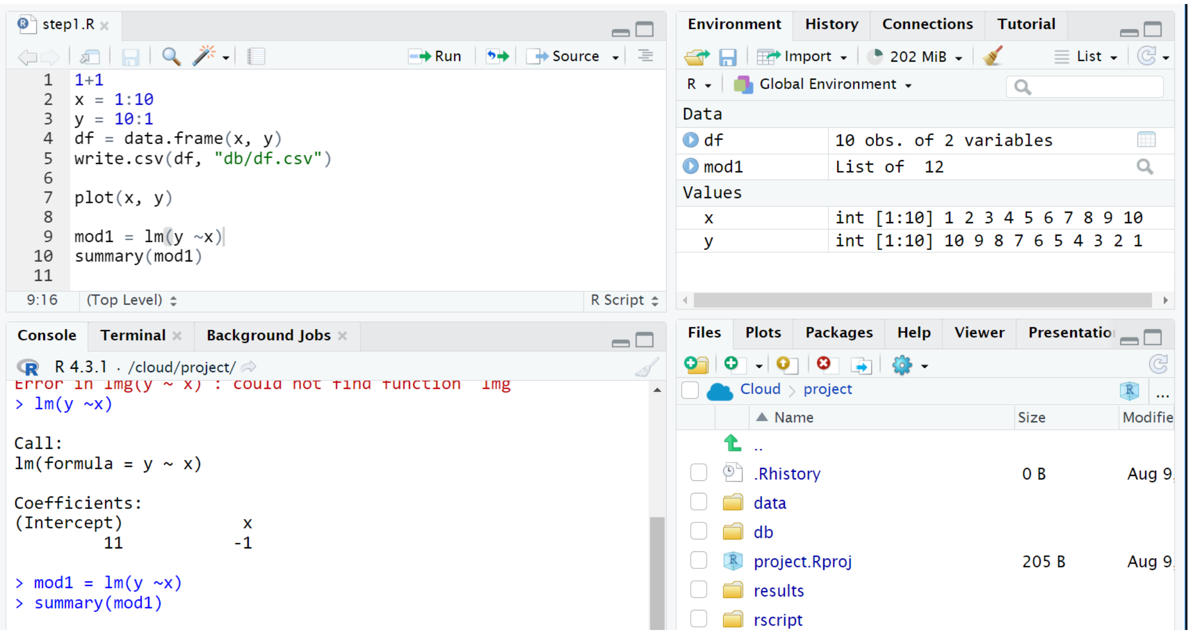
- Rstudio cloud
  - RStudio Cloud .
  - .
  - New Project .
  - .

.



```
x = 1:10
y = 1:10
plot(x, y)
```

```
Values
  x    int [1:10] 1 2 3 4
  y    int [1:10] 1 2 3 4
```


step1.R
Run
Source
9:16 (Top Level)
R Script
Console Terminal Background Jobs
R 4.3.1 · /cloud/project/
Environment History Connections Tutorial
Import
202 MiB
List
Global Environment
Files Plots Packages Help Viewer Presentatio
Cloud > project
Name Size Modifie

```
1+1
x = 1:10
y = 10:1
df = data.frame(x, y)
write.csv(df, "db/df.csv")

plot(x, y)

mod1 = lm(y ~x)
summary(mod1)
```

```
Error in lmg(y ~ x) : could not find function "lmg"
> lm(y ~x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
         11           -1

> mod1 = lm(y ~x)
> summary(mod1)
```

```
Data
  df       10 obs. of 2 variables
  mod1     List of  12
Values
  x        int [1:10] 1 2 3 4 5 6 7 8 9 10
  y        int [1:10] 10 9 8 7 6 5 4 3 2 1
```

```
..
.Rhistory        0 B     Aug 9
data
db
project.Rproj    205 B   Aug 9
results
rscript
```

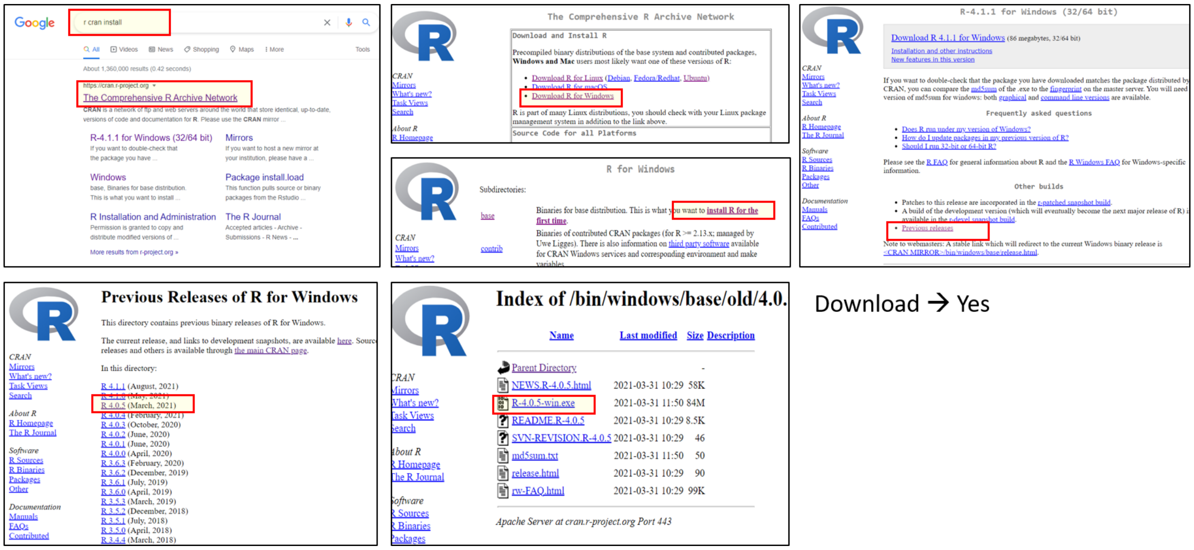
1.2 R and R studio on Window system

Windows R R .

1. R cran .

- .

2. R .
3. Windows R
4. Base → Install R for the first Time
5. **Download R * for Windows**



r cran install

About 1,360,000 results (0.42 seconds)

https://cran.r-project.org

The Comprehensive R Archive Network

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror ...

R-4.1.1 for Windows (32/64 bit) — If you want to double-check that the package you have ...

Mirrors — If you want to host a new mirror at your institution, please have a ...

Windows — base, Binaries for base distribution. This is what you want to install ...

Package install.load — This function pulls source or binary packages from the Rstudio ...

R Installation and Administration — Permission is granted to copy and distribute modified versions of ...

The R Journal — Accepted articles - Archive - Submissions - R News - ...

More results from r-project.org »

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

R for Windows

Subdirectories:

base — Binaries for base distribution. This is what you want to install R for the first time.

contrib — Binaries of contributed CRAN packages (for R >= 2.13.x; managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.

R-4.1.1 for Windows (32/64 bit)

Download R 4.1.1 for Windows (86 megabytes, 32/64 bit)

Installation and other instructions

New features in this version

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the fingerprint on the master server. You will need version of md5sum for windows: both graphical and command line versions are available.

Frequently asked questions

- Does R run under my version of Windows?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

Other builds

- Patches to this release are incorporated in the r-patched snapshot build.
- A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.
- Previous releases

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.html.

Previous Releases of R for Windows

This directory contains previous binary releases of R for Windows.

The current release, and links to development snapshots, are available here. Source releases and others is available through the main CRAN page.

In this directory:

R 4.1.1 (August, 2021)
R 4.1.0 (May, 2021)
R 4.0.5 (March, 2021)
R 4.0.4 (February, 2021)
R 4.0.3 (October, 2020)
R 4.0.2 (June, 2020)
R 4.0.1 (June, 2020)
R 4.0.0 (April, 2020)
R 3.6.3 (February, 2020)
R 3.6.2 (December, 2019)
R 3.6.1 (July, 2019)
R 3.6.0 (April, 2019)
R 3.5.3 (March, 2019)
R 3.5.2 (December, 2018)
R 3.5.1 (July, 2018)
R 3.5.0 (April, 2018)
R 3.4.4 (March, 2018)

Index of /bin/windows/base/old/4.0.

Name	Last modified	Size	Description
Parent Directory		-	
NEWS.R-4.0.5.html	2021-03-31 10:29	58K	
R-4.0.5-win.exe	2021-03-31 11:50	84M	
README.R-4.0.5	2021-03-31 10:29	8.5K	
SVN-REVISION.R-4.0.5	2021-03-31 10:29	46	
md5sum.txt	2021-03-31 11:50	50	
release.html	2021-03-31 10:29	90	
rw-FAQ.html	2021-03-31 10:29	99K	

Apache Server at cran.r-project.org Port 443

Download → Yes

Figure 1.1: R

1.2.1 R studio install

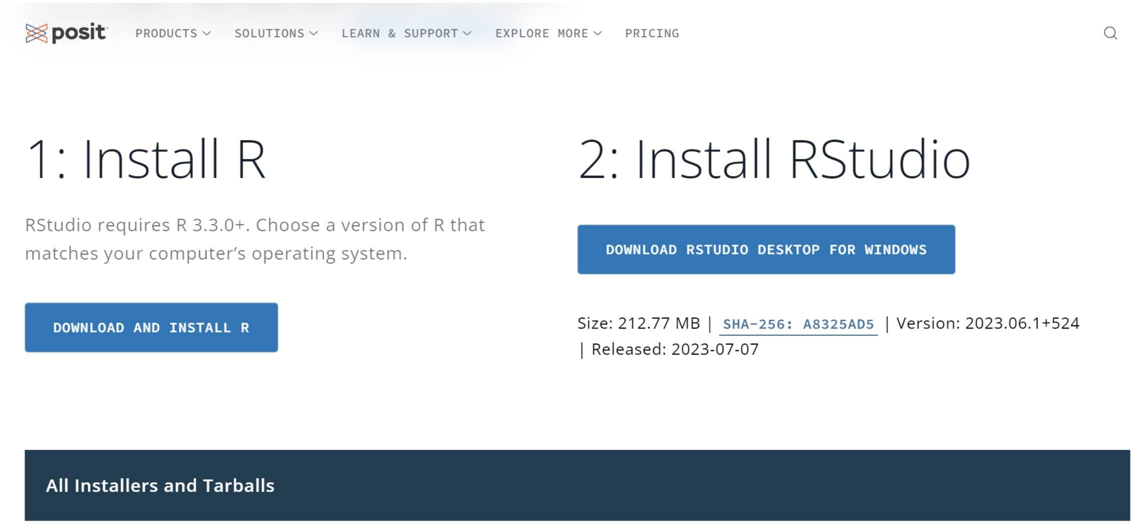
R . Rstudio . https://posit.co/download/rstudio-desktop/

R studio .

1.2.2 R studio project

R . . .

. .



1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

2: Install RStudio

Size: 212.77 MB | SHA-256: A8325AD5 | Version: 2023.06.1+524 | Released: 2023-07-07

Figure 1.2: R

Windows .

, R .

1.3 R and R studio on Ubuntus

R . , . . , docs .

Google Doc Google Doc Download

R

vector matirx list

objects

(value) (variable) . .

```
a <-1
b <-2
c <- -1

a*b+c
```

```
[1] 1
```

. (numeric), (character), (logical) .

```
student_a_age  <- 43
student_a_name <- '  '
student_a_pass <- FALSE
```

, 0 .

(Vectors), (factors)

. `c()` `c` **concatenate** C . (missing value) `NA` `NAN` . , `my_vector[i]` `i` .

```
student_all_age <- c(24, 31, 40, 16)
student_all_age
```

```
[1] 24 31 40 16
```

.

```
student_all_age[1]
student_all_age[1:2]
student_all_age[c(1, 3)]
student_all_age[-4]
```

. max(), min(), range(), length(), sum(), mean(), prod(), sd(), var(), sort() .

```
max(student_all_age)
```

```
[1] 40
```

```
length(student_all_age)
```

```
[1] 4
```

```
student_all_name <- c(' ', ' ', ' ', ' ')
summary(student_all_name)
```

```
   Length     Class      Mode 
        4 character character 
```

```
student_all_class <- c('group1','group2', 'group1', 'group2' )
summary(student_all_class)
```

```
   Length     Class      Mode 
        4 character character 
```

student_all_class group 1 2 . . sutdent_all_class .

```
student_all_group <-factor(c('group1','group2', 'group1', 'group2' ))
summary(student_all_group)
```

```
group1 group2
     2      2
```

group .

```
tapply(student_all_age, student_all_group, mean)
```

```
group1 group2
  32.0   23.5
```

*apply . , 1 . 2

(Matrix), (data frame), (list)

. (row) (column) , . .

```
A1 = matrix(
     c(1, 2, 3, 4, 5, 6, 7, 8),  #
     nrow=2,                     #
     ncol=4,                     #
     byrow = TRUE)               #
A1 #
```

1, 2, 3, 4, 5, 6, 7, 8

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

(column) .

```
dspub_class <- data.frame(
  'name'  = student_all_name,
  'age'   = student_all_age,
  'group' = student_all_group
)
dspub_class
```

```
  name age  group
1    24 group1
2    31 group2
3    40 group1
4    16 group2
```

tidyverse . %>% pipe . mutate
.

```
#install.packages("tidyverse")
library(tidyverse)
```

```
dspub_class %>%
  group_by(group) %>%
  summarize(avg = mean(age))
```

```
# A tibble: 2 x 2
  group    avg
  <fct>  <dbl>
1 group1  32
2 group2  23.5
```

. DSpub group1 gropu2 . / , / . .

```
homework1 <- dspub_class %>%
  group_by(group) %>%
  mutate(avg_age = mean(age))
homework2 <- dspub_class %>%
  group_by(group) %>%
  mutate(max_age = max(age))
```

```
homework1
```

```
# A tibble: 4 x 4
# Groups:   group [2]
  name    age group  avg_age
  <chr> <dbl> <fct>    <dbl>
1       24 group1    32
2       31 group2    23.5
3       40 group1    32
4       16 group2    23.5
```

```
homework2
```

```
# A tibble: 4 x 4
# Groups:   group [2]
  name    age group  max_age
  <chr> <dbl> <fct>    <dbl>
1         24 group1      40
2         31 group2      31
3         40 group1      40
4         16 group2      31
```

, list .

```
second_week_dspub <-
  list(
       student_all_age,
       student_all_class,
       student_all_group,
       student_all_name,
       dspub_class,
       homework1,
       homework2
       )
second_week_dspub
```

```
[[1]]
[1] 24 31 40 16

[[2]]
[1] "group1" "group2" "group1" "group2"

[[3]]
[1] group1 group2 group1 group2
Levels: group1 group2

[[4]]
[1] " " " " " " " "

[[5]]
  name age  group
1     24 group1
```

```
2    31 group2
3    40 group1
4    16 group2

[[6]]
# A tibble: 4 x 4
# Groups:   group [2]
  name    age group  avg_age
  <chr> <dbl> <fct>    <dbl>
1       24 group1    32
2       31 group2    23.5
3       40 group1    32
4       16 group2    23.5

[[7]]
# A tibble: 4 x 4
# Groups:   group [2]
  name    age group  max_age
  <chr> <dbl> <fct>    <dbl>
1       24 group1      40
2       31 group2      31
3       40 group1      40
4       16 group2      31
```

, list . list .

```
second_week_dspub[[7]]
```

```
# A tibble: 4 x 4
# Groups:   group [2]
  name    age group  max_age
  <chr> <dbl> <fct>    <dbl>
1       24 group1      40
2       31 group2      31
3       40 group1      40
4       16 group2      31
```

. **1+2** , **log2(10)** . Rsutdio .

```
3+4;4-3;4/3;3*4
log2(10)
abs(-4)
sqrt(4)
```

.

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
ˆ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/%2 is 2

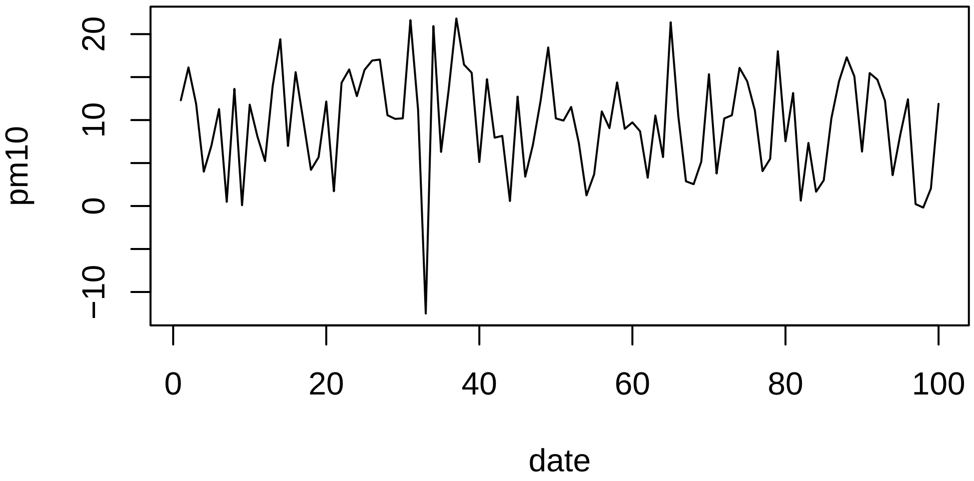
Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
!x	Not x
x	y
x & y	x AND y
isTRUE(x)	test if X is TRUE

Operator	Description
Logarithms and exponentials	log2(x), log10(x), exp(x)
Trigonometric functions	cos(x), sin(x), tan(x), acos(x), asin(x), atan(x)
Others	abs(x): absolute value; sqrt(x): square root.

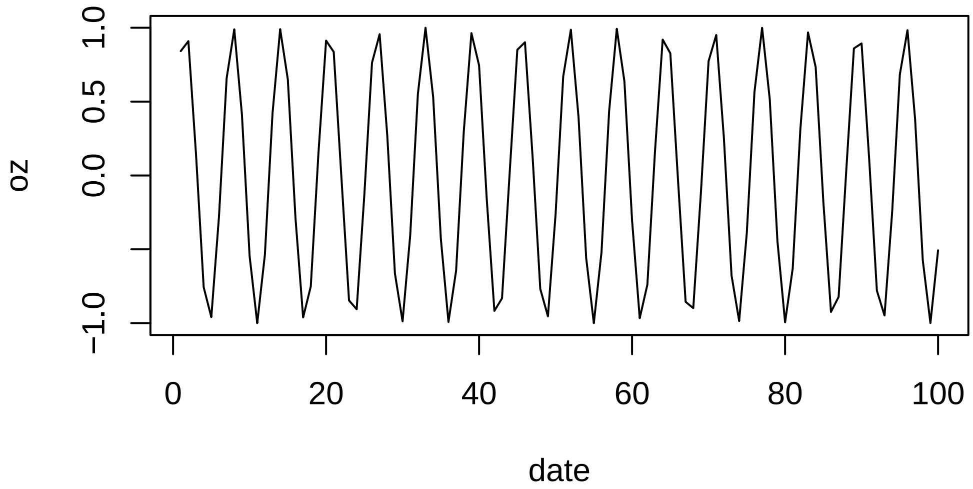
.

```
pm10 = rnorm(n=100, mean = 10, sd = 5) #    10       5  100         pm10
date= rep(1:100)  # 1   100
```

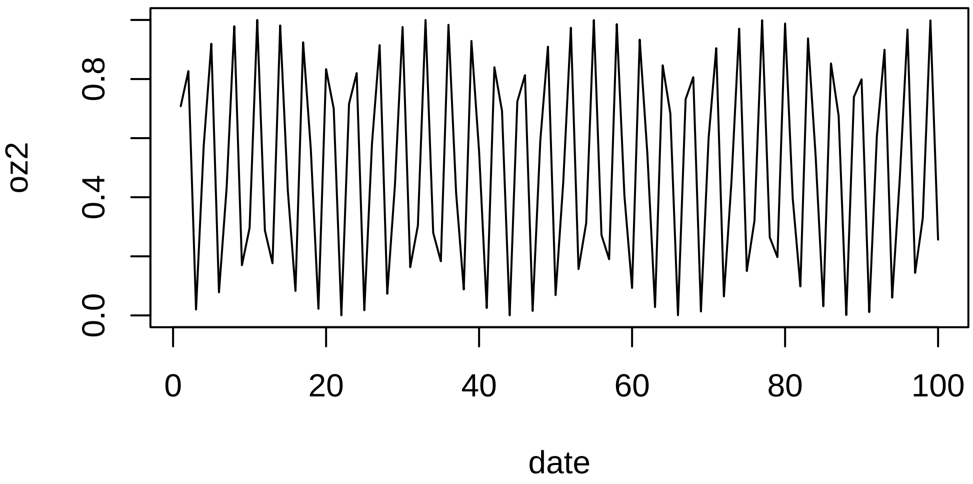
```
plot(x=date, y=pm10, type = "l") # "l" line        .
```



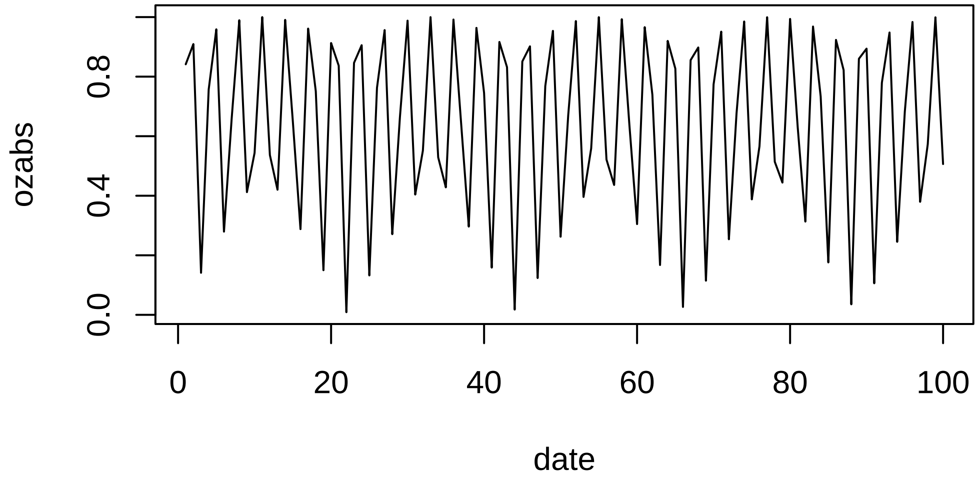
```
oz = sin(date) #                     . sine
plot(x=date, y = oz, type = "l") #
```



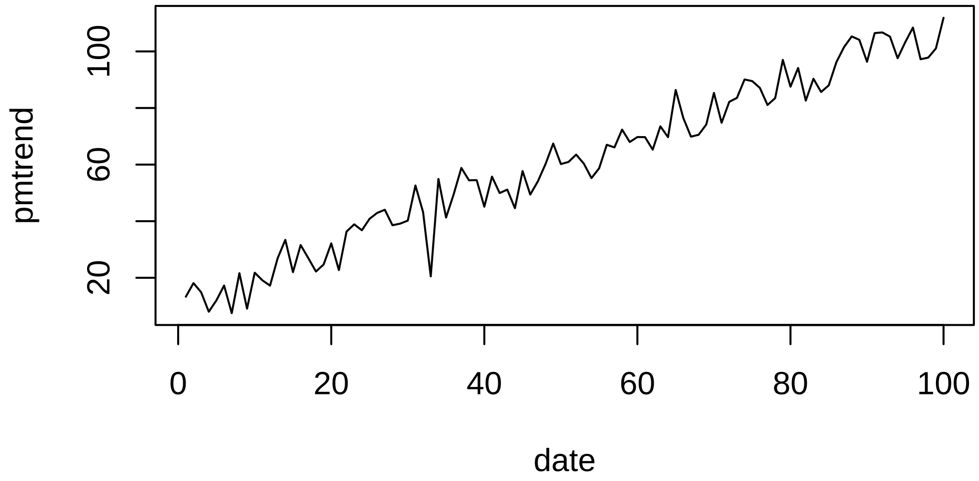
```
oz2 = oz**2 # -            .
plot(x=date, y = oz2, type = "l") #
```



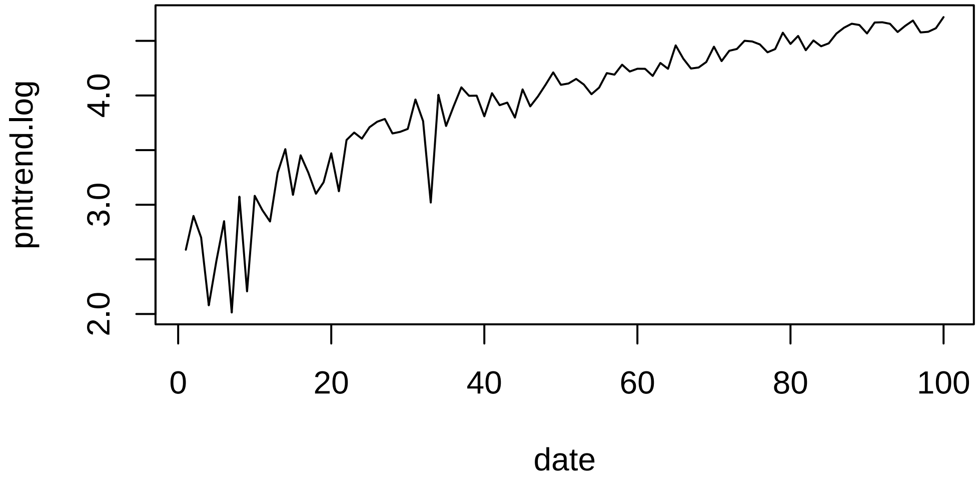
```
ozabs = abs(oz) #          ?
plot(x=date, y = ozabs, type = "l") #
```



```
pmtrend = pm10 + date  #      pm10
plot(x=date, y = pmtrend, type="l")
```



```
pmtrend.log = log(pmtrend) #       .               .
plot(x=date, y = pmtrend.log, type="l") #
```



`if-else` . 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 5 A B .

```
nums<- 1
if (nums <5) {
        chars = 'A'
} else{
        chars = 'B'
}
chars
```

```
[1] "A"
```

```
nums <- 6
if (nums <5) {
        chars = 'A'
} else{
        chars = 'B'
}
chars
```

```
[1] "B"
```

.

```
a<-round(rnorm(10)*10)
a
```

```
 [1]  -1   8  -4   4  -1  -3   5 -13   6 -17
```

```
tab <- ifelse(a>0, ' ', ' ')
tab
```

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

```
data.frame(a, tab)
```

```
     a  tab
1   -1
2    8
3   -4
4    4
5   -1
6   -3
7    5
8  -13
9    6
10 -17
```

any() all() . any() TRUE TRUE , all() TRUE TRUE .

```r
new.var <- c(1, 2, NA)
is.na(new.var)
```

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

```r
any(is.na(new.var))
```

```
[1] TRUE
```

```r
all(is.na(new.var))
```

```
[1] FALSE
```

index IF (Where) . iris Sepal.Length , Sepal.Length Species . ?

```r
head(iris)
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

```r
table(iris$Species)
```

```
    setosa versicolor  virginica 
        50         50         50 
```

```r
max(iris$Sepal.Length)
```

```
[1] 7.9
```

```r
max.length <- which.max(iris$Sepal.Length)
iris$Species[max.length]
```

```
[1] virginica
Levels: setosa versicolor virginica
```

iris Sepal.Length , Sepal.Length Species .

```r
min(iris$Sepal.Length)
```

```
[1] 4.3
```

```r
min.length <- which.min(iris$Sepal.Length)
iris$Species[min.length]
```

```
[1] setosa
Levels: setosa versicolor virginica
```

R . , , , . 2

```r
addtive.function = function(x, y ){
  x + y
}
```

```
addtive.function(100, 2)
```

```
[1] 102
```

2 abs . #-# .

```
abs.function= function(x, y ){
  #--#
}
```

avg . length , .

```
my_vector<- 1:50
avg <- function(x){
        sum(x)/length(x)
}
avg(my_vector)
```

```
[1] 25.5
```

, , , .

```
tabs <- function(x){
        data.frame( ' '      = mean(x),
                    '   '  = length(x),
                    '  '    = max(x),
                    '  '    = min(x)
        )
}

tabs(my_vector)
```

```
1 25.5        50      50       1
```

,

```r
avg <- function(x, arithmetic = TRUE){
  n <- length(x)
  ifelse(arithmetic, sum(x)/n, prod(x)^(1/n))
}
```

, vectorization, functionals

for loop

1a, 2a, 3a, 4a, 5a, 6a, 7a, 8a, 9a, 10a , ?

```r
c('1a', '2a', '10a') #
```

```
[1] "1a"  "2a"  "10a"
```

, .

```r
for (i in 1:10){
  print(paste0(i, 'a'))
}
```

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

, .

```r
paste0(1:10, "a")
```

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

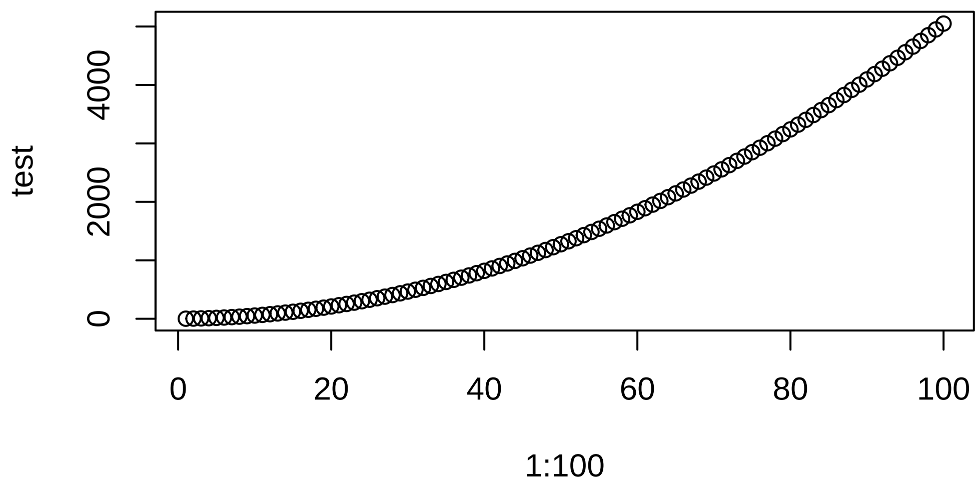
`1:n` , 1 100

```
compute <- function(n){ sum(1:n)}
compute(10)
```

```
[1] 55
```

```
test <-c()
for (n in 1:100){
  test[n] <- compute(n)
}

plot(1:100, test)
```



vectorization apply

ifelse . . ifesel 10 1 . apply . **past0(1:10,“a”)** . 2 3 . **for if** . .

```
n2 <- c(1:9*2)
n3 <- c(1:9*3)
n2*n3
```

```
[1]   6  24  54  96 150 216 294 384 486
```

2 3

```
new.function<-function(n2){
  c(1:9*2) * c(1:9*n2)
}

new.function( 4)
```

```
[1]   8  32  72 128 200 288 392 512 648
```

3 1, 2, 3, 4, 5, 6, 7, 8, 9

```
sapply(1:9, new.function)
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
 [1,]    2    4    6    8   10   12   14   16   18
 [2,]    8   16   24   32   40   48   56   64   72
 [3,]   18   36   54   72   90  108  126  144  162
 [4,]   32   64   96  128  160  192  224  256  288
 [5,]   50  100  150  200  250  300  350  400  450
 [6,]   72  144  216  288  360  432  504  576  648
 [7,]   98  196  294  392  490  588  686  784  882
 [8,]  128  256  384  512  640  768  896 1024 1152
 [9,]  162  324  486  648  810  972 1134 1296 1458
```

ifelse for . apply . **tidyverse** `apply`, `lapply`, `sapply` .

iris data apply

R . . iris .

```
data("iris")
```

iris . 3 150 . :

- Sepal.Length ( )

- Sepal.Width (   )
- Petal.Length (   )
- Petal.Width (   )
- Species ( ):   .
    - ‘setosa’(   ), ‘versicolor’(   ), ‘virginica’(   )   3   .

```
library(tidyverse)
iris %>% head()
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

`Species`   ,   .   .   , ,   .

apply   .   apply   .

```
iris_num=iris[, c(1:4)] # Species
iris_num=iris %>% select(1:4) # tidyverse
#        , sd       .
iris_num %>% pull(Sepal.Length) %>% mean()
```

```
[1] 5.843333
```

```
iris_num %>% pull(Sepal.Width)  %>% mean()
```

```
[1] 3.057333
```

```
iris_num %>% pull(Petal.Length) %>% mean()
```

```
[1] 3.758
```

```
iris_num %>% pull(Petal.Width)  %>% mean()
```

```
[1] 1.199333
```

```
iris_num %>% pull(Sepal.Length) %>% sd()
```

```
[1] 0.8280661
```

```
iris_num %>% pull(Sepal.Width)  %>% sd()
```

```
[1] 0.4358663
```

```
iris_num %>% pull(Petal.Length) %>% sd()
```

```
[1] 1.765298
```

```
iris_num %>% pull(Petal.Width)  %>% sd()
```

```
[1] 0.7622377
```

median . , . , . apply . apply , lapply list . sapply lapply .

```
apply(iris_num, 2, mean)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
    5.843333     3.057333     3.758000     1.199333 
```

```
lapply(iris_num, mean)
```

```
$Sepal.Length
[1] 5.843333

$Sepal.Width
[1] 3.057333

$Petal.Length
[1] 3.758

$Petal.Width
[1] 1.199333
```

```
sapply(iris_num, mean)
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width 
    5.843333     3.057333     3.758000     1.199333 
```

, , . . stat_smry

```
stat_smry = function(x){
  list(
    mean  =mean(x),
    median=median(x),
    std   = sd(x)
  )
}
```

sapply .

```
sapply(iris_num, stat_smry)
```

```
       Sepal.Length Sepal.Width Petal.Length Petal.Width
mean   5.843333     3.057333    3.758        1.199333   
median 5.8          3           4.35         1.3        
std    0.8280661    0.4358663   1.765298     0.7622377  
```

lapply . . . lapply tt list `do.call(rbidn, .)` .rbind row bind list , . , .

```
tt = lapply(iris_num, stat_smry)
```

```
names(tt) #
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width" 
```

```
tt[[1]]    #       "Sepal.Length"
```

```
$mean
[1] 5.843333

$median
[1] 5.8

$std
[1] 0.8280661
```

```
do.call(rbind, tt) #
```

```
             mean     median std
Sepal.Length 5.843333 5.8    0.8280661
Sepal.Width  3.057333 3      0.4358663
Petal.Length 3.758    4.35   1.765298
Petal.Width  1.199333 1.3    0.7622377
```

2 Summary

In summary, this book has no content whatsoever.

```
1 + 1
```

```
[1] 2
```