

R programming

Vinish Shrestha

10/18/2020

Version0

This is intro.

We will keep building.

R programming install (ubuntu)

1. install dependencies necessary for adding new repository
 - ▶ `sudo apt install dirmngr gnupg apt-transport-https ca-certificates software-properties-common`
2. Add Comprehensive R Archive Network (CRAN) repository to the system source list
 - ▶ `sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9`
 - ▶ `sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/'`
3. install R
 - ▶ `sudo apt install r-base`
4. `R --version`

Update R

- ▶ `$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9`
- ▶ `$ sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu focal-cran40/'`
- ▶ `$ sudo apt install r-base`

Install R studio

- ▶ `sudo apt-get install gdebi-core`
- ▶ `wget https://download2.rstudio.org/server/bionic/amd64/rstudio-server-1.4.1717-amd64.deb`
- ▶ `sudo gdebi rstudio-server-1.4.1717-amd64.deb`

Install for windows

- ▶ visit this site
- ▶ click download R for windows
- ▶ click on base
- ▶ Then click "Download R 4.1.1 for Windows"

Next, you'd wanna install RStudio IDE, for better environment

- ▶ go to rstudio page
- ▶ pick the appropriate OS and download
- ▶ The current file for windows is RStudio-1.4.1717.exe

Use this [link](#) as a helpful resource Watch this [video](#) as a guide if you have problems redo.

Helpful resources to learn

- ▶ Venables and friends
- ▶ Grolemund
- ▶ stackover flow
- ▶ CRAN <https://CRAN.R-project.org>

1. Numbers and vectors

```
x <- c(7.4, 5.2, 10.2, 3.3)
print(x)
```

```
## [1] 7.4 5.2 10.2 3.3
```

- ▶ x is a variable (object), c() is a function that can store vector arguments
- ▶ <- tells that object c receives the designated values; can be replaced by "=" in most instances

```
length(x)
```

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

```
y <- 2*x - 4 #basic arithmetic
print(y)
```

```
## [1] 10.8 6.4 16.4 2.6
```

2. Regular sequences

```
even_n <- seq(2, 20, 2)
odd_n  <- seq(1, 19, 2)
even_n
```

```
## [1] 2 4 6 8 10 12 14 16 18 20
```

```
odd_n
```

```
## [1] 1 3 5 7 9 11 13 15 17 19
```

Repetition

```
rep1 <- rep(1:3, 2)
rep2 <- rep(1:3, each = 2)
rep1
```

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

```
rep2
```

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


3. Missing values

- ▶ Often values of variables in observational data are missing
- ▶ missing data is represented by NA

```
x <- c(seq(2, 6, 2), NA)
x
```

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

Say now you want to drop the missing observation from the vector x and store it in vector y

```
y <- x[!is.na(x)]
y
```

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

4. Location of a vector

- ▶ use $[i]$, where i is the integer of interest to find out i^{th} element of a vector

```
even_n <- seq(2, 20, 2)
paste("The second element of even_n is", even_n[2],
      sep = " ")
```

```
## [1] "The second element of even_n is 4"
```

- ▶ here paste joins the character or string "The second element of even_n is" with the object even_n[2]

Quotes

```
x <- "long\tlines can be\nbroken with newlines"  
print(x)
```

```
## [1] "long\tlines can be\nbroken with newlines"
```

```
writeLines(x) # see also ?strwrap
```

```
## long lines can be
```

```
## broken with newlines
```

► see ?Quotes

5. R Objects and Class

1. Character vector

► entered by "

```
a <- c("A", "B", "C")  
a
```

```
## [1] "A" "B" "C"
```

```
class(a)
```

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

```
is.vector(a) #tests whether a is vector
```

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

```
length(a) #returns length
```

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

5. R Objects and Class

2. Numeric

```
x <- seq(1, 10, 1)  
class(x)
```

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

```
length(x) #know the length
```

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

```
x[1] #first element of vector x
```

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

5. R Objects and Class

3. Logicals

```
a <- 4  
l <- a>3  
class(l)
```

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

5. R Objects and Class

4. Matrix

```
mat <- matrix(0, nrow = 3, ncol = 3) #three by three matrix  
mat #display the matrix
```

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

```
mat[,1] #display the first column of the matrix
```

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

5.

5. arrays

```
ar <- array(c(11:14, 21:24, 31:34), dim = c(2, 2, 3))
ar
```

```
## , , 1
```

```
##
```

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

```
## [1,]    11    13
```

```
## [2,]    12    14
```

```
##
```

```
## , , 2
```

```
##
```

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

```
## [1,]    21    23
```

```
## [2,]    22    24
```

```
##
```

```
## , , 3
```

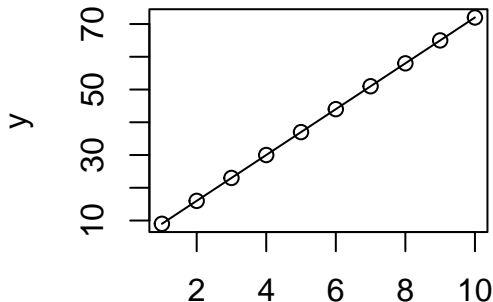
```
##
```


Other objects

- ▶ matrices: multi-dimensional vectors
- ▶ factors: efficient way to handle categorical variables
- ▶ lists: general form of vector that handles objects of different classes
- ▶ data.frames: typical data as in excel
- ▶ functions: used to manipulate data

A Simple Plot

```
x <- seq(1, 10, 1)
y <- 2 + x*7
plot(x, y)
lines(x, y)
```



```
## End Sept 8; Next time
## 1. dataframe
## 2. manipulating dataframe
## 3. running simple regression
```

practice

practice

practice