

# R for data analysis

---

Clinical Research Support Unit





# Contents

- 1 Objectives
- 2 R interface
- 3 Installation
- 4 Enter data
- 5 Create dataset
- 6 Import dataset
- 7 Save/export dataset
- 8 Select variables
- 9 Data Visualization

# Objectives



**R Introduction**



**Data  
Management**

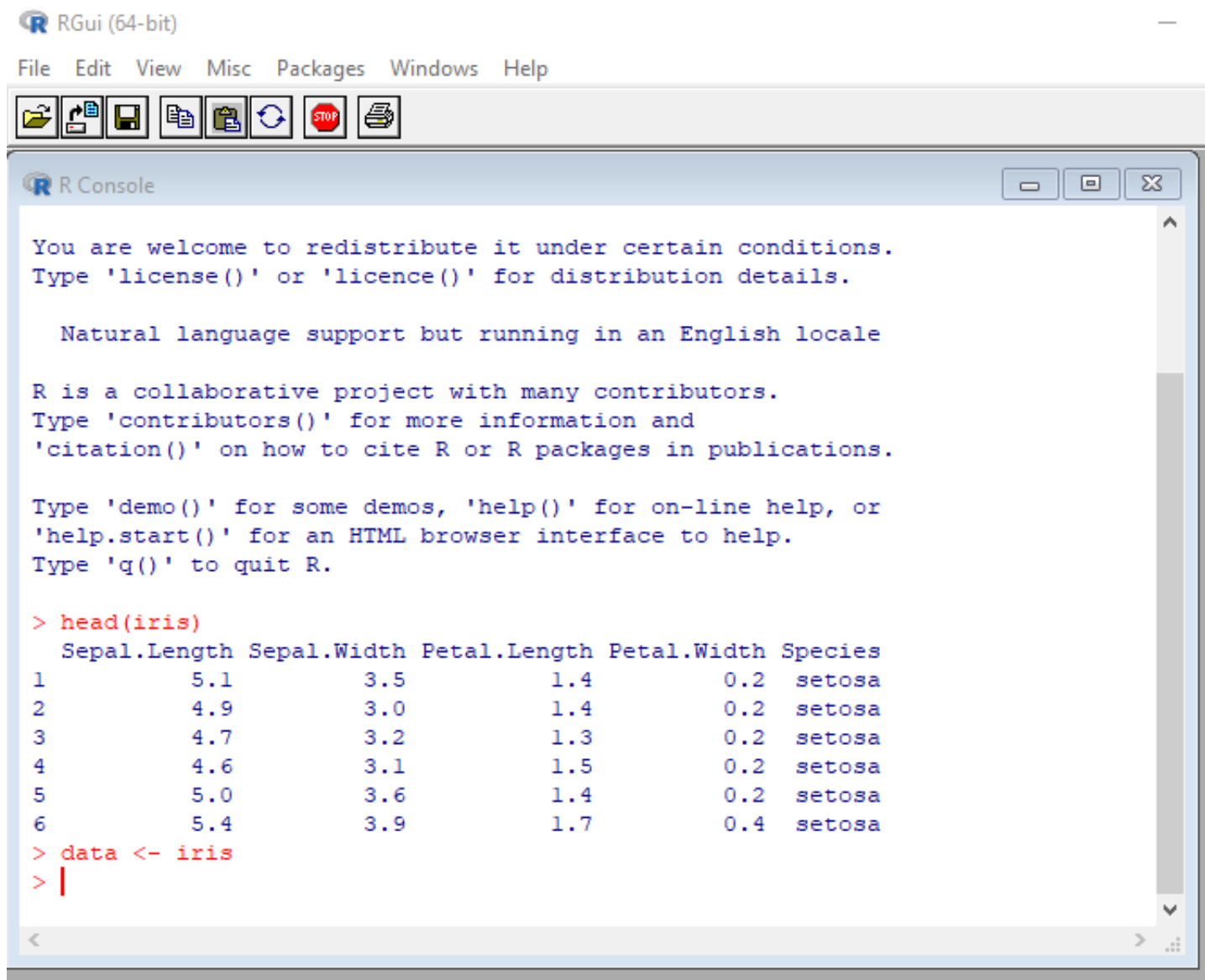


**Statistical  
Analysis**



**Visualization**

# R interface – R GUIs (Graphical User Interfaces)



RGui (64-bit)

File Edit View Misc Packages Windows Help

R Console

```
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit 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  
> data <- iris  
> |
```

# R interface – RStudio

The screenshot shows the RStudio R interface with several components highlighted by red boxes and labels:

- R Script:** The main editor window on the left, containing the R code:

```
1 head(iris)
2 data <- iris
3
```
- Environment, History:** The top-right pane, showing the **Environment** tab with the **Global Environment** selected. It displays a variable **data** with 150 observations of 5 variables.
- Console:** The bottom-left pane, showing the output of the R code:

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1          3.5          1.4          0.2
2          4.9          3.0          1.4          0.2
3          4.7          3.2          1.3          0.2
4          4.6          3.1          1.5          0.2
5          5.0          3.6          1.4          0.2
6          5.4          3.9          1.7          0.4
Species
```
- Files, Plots, Packages, Help:** The bottom-right pane, showing the **Files** tab with a list of files in the current directory, including **R workshop.Rproj** and **write-test-10e1bfe6-42af-4f17-b71...**.

r software

## 1. Search R



Images

Download

Tutorial

Course

Videos

Books

Online

Engineering

Suite

About 12,320,000,000 results (0.56 seconds)

### Sponsored



Alteryx

<https://www.alteryx.com>

### R Software - Get Your 30-Day Free Trial

Ready For Insights Faster Than You Ever Thought Possible? Start Your Free Trial Today. Make The Most Of Your Organization's Data To Extract The Best Value To Your Business. Unified Platform. Data Validation. Analytics Visualizations. 7,000+ Global Customers.



The R Project for Statistical Computing

<https://www.r-project.org>

### The R Project for Statistical Computing

R is a free **software** environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

Results from r-project.org



### An Introduction to R

This manual provides information on data types, programming ...

### R-4.3.1 for Windows

This build requires UCRT, which is part of Windows since Windows

### About R

R is a language and environment fo

### R for macOS

Tools - Index of /bin/macosx/base - FAQs - ...

## 2. Click Windows or macOS depending on your OS

# Installation

## R-4.3.1 for Windows

### 2. Click download

[Download R-4.3.1 for Windows](#) (79 megabytes, 64 bit)

[README on the Windows binary distribution](#)

[New features in this version](#)



r studio download



For windows

For Android

For Mac

Full Crack

For Windows 10

CRAN

Images

About 2,540,000,000 results (0.46 seconds)



Posit

<https://posit.co> › download › rstudio-desktop

### RStudio Desktop

Used by millions of people weekly, the **RStudio** integrated development environment (IDE) is a set of tools built to help you be more productive with **R** and ...

[R Packages](#) · [Conf\(2023\)](#) · [Resources](#) · [Videos](#)

<https://posit.co> › downloads

[Download RStudio](#)

### 3. Click download RStudio

The most popular coding environment for **R**, built with love by Posit. Used by millions of people weekly, the **RStudio** integrated development environment (IDE) is ...

# Installation

## 4. Click download RStudio

DOWNLOAD

## RStudio IDE

The most popular coding environment for R, built with love by Posit.

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution. It also features tools for plotting, viewing history, debugging and managing your workspace.

If you're a professional data scientist and want guidance on adopting open-source tools at your organization, don't hesitate to [book a call with us](#).

DOWNLOAD RSTUDIO

DOWNLOAD RSTUDIO SERVER

## RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to [book a call with us](#).

## 1: Install R

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

DOWNLOAD AND INSTALL R

## 2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

Size: 212.77 MB | [SHA-256: A8325AD5](#) | Version: 2023.06.1+524 |  
Released: 2023-07-07

## 5. Click download RStudio



## R works via USER made packages

- R works via USER make packages
  - There are 20066 packages
  - [https://cran.r-project.org/web/packages/available\\_packages\\_by\\_name.html](https://cran.r-project.org/web/packages/available_packages_by_name.html)
  - Submission and general verification is done by volunteers with CRAN
  - Fundamentally different than Stata and SAS
- Two general frameworks for working with data
  - Base R
  - Tidyverse (Developed by RStudio)
  - If you google around know which one you are looking for

## Good coding practices

- Good idea to follow
  - <https://google.github.io/styleguide/Rguide.html>
  - <https://www.r-bloggers.com/2018/09/r-code-best-practices/>
    - There are 5 naming conventions to choose from:
      - *alllowercase*: e.g. adjustcolor
      - *period.separated*: e.g. plot.new
      - *underscore\_separated*: e.g. numeric\_version
      - *lowerCamelCase*: e.g. addTaskCallback
      - *UpperCamelCase*: e.g. SignatureMethod
    - Pick one naming convention and stick to it. My suggestion:
      - **Files**: underscore\_separated, all lower case: e.g. numeric\_version
      - **Functions**: *UpperCamelCase*, all lower case: e.g. MyFunction
      - **Variables**: underscore\_separated, all lower case: e.g. numeric\_version

## Entering data (vector)

# if we put # sign, R will not recognize it

# entering numeric data/value **1** into environment (named as **a**)

*a <- 1*

# to see data

*a*

# entering **hello world** into environment (named as **b**)

# hello is character data, use single or double quote

*b <- "hello world"*

*b*

# Given names such a, b are not case sensitive

## Entering data (vector)

# entering multiple **numeric** data into environment (named as **a\_1**)

```
a_1 <- c(22, 55, 26, 30, 42, 24)
```

```
a_1
```

# capital C will not work

# no space or unequal space between values will work

# entering multiple **character** data into environment (named as **b\_1**)

```
b_1 <- c("m", "f", "m", "m", "f", "m")
```

```
b_1
```

# Entering data, Creating dataset

```
c.1 <- c(1, 2, 3, 4, 5, 6)
```

```
c.1
```

```
# creating data set named 'practice' using variable a_1, b_1, c_1
```

```
practice <- data.frame(age=a_1, sex=b_1, id=c_1)
```

```
# length of a_1, b_1, c_1 should be same
```

```
# giving name as 'age' using a_1
```

```
# giving name as 'sex' using b_1
```

```
# giving name as 'id' using c_1
```

```
# data.frame is a function and should be typed as it is
```

# Exploring dataset

# to know about the data

*str(practice)*

# to get summary statistics

*summary(practice)*

# to get summary statistics only for age

*summary(practice\$age)*

# Exploring dataset

# to see frequency distribution of variables  
# we need to install a package called 'dplyr'

```
install.packages('dplyr')  
library(dplyr)
```

# to see frequency distribution of age  
*dplyr::count(practice, age)*

# to see frequency distribution of sex  
*dplyr::count(practice, sex)*

# Exploring dataset

#to see summary data

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

```
st(practice)
```

#individual function can be used to get certain descriptives

```
sd(practice$age)
```

#sd is a function



# Saving data set

# Export/save data 'practice' in csv format

# Where is the data going to go?

*getwd()*

*setwd()*

*write.csv(practice, file = "location/practice.csv")*

## Importing data named 'test'

```
test <- read.csv("location of the data/test.csv", header = TRUE)
```

```
head(test,10)
```

```
tail(test,10)
```

# To know about data

```
str(test)
```

```
summary(test)
```

## Creating sub data set

```
# keep selected variables age, gender, id in a new data set named 'newdata'  
newdata <- dplyr::select(test, age, gender, id)  
str(newdata)
```

```
# creating sub data set named 'newdata1' with female only
```

```
newdata_female <- dplyr::filter(test, gender == "f")  
str(newdata_female)
```

# Data visualization

# to see histogram

```
hist_age <- hist(test$age)
```

```
plot(hist_age)
```

#visualize your data using scatter plots

```
install.packages("ggplot2")
```

```
library(ggplot2)
```

## histogram for age

```
hist_age_ggplot <- ggplot(test, aes(age)) +
```

```
  geom_histogram()
```

```
plot(hist_age_ggplot)
```

# Data visualization

```
## scatter plot ldl1 and ldl2  
# ldl1 and ldl2 are continuous variables
```

```
scatter_ldl <- ggplot(test, aes(x = ldl1, y = ldl2)) +  
  geom_point()  
plot(scatter_ldl)
```

```
## scatter plot ldl1 and ldl2 with a regression line
```

```
scatter_ldl_lm <- ggplot(test, aes(x = ldl1, y = ldl2)) +  
  geom_point() +  
  geom_smooth(method = "lm")  
plot(scatter_ldl_lm )
```

# Data visualization

## save your image

```
ggsave(scatter_lm.pdf, plot = scatter_ldl_lm, dpi = 300)
```

Acknowledgment  
Jiwon Yoon  
Prosanta Mondal