



# Data Science Using R

**LIVE PROJECTS, REAL DATASETS, CASE  
STUDIES AND PRACTICAL EXAMPLES**

**- A PUBLICATION OF LISTENDATA.COM -**



## CHAPTER ONE

---

# Introduction to R

[LISTENDATA.COM](http://LISTENDATA.COM)

# Introduction to R

---



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



R is a language and environment for statistical computing and graphics



Has its root in S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies)



You can perform a variety of tasks using R language. Some are as follows –

- ☐ Exploring and Manipulating Data
- ☐ Building and validating predictive models
- ☐ Applying machine learning and text mining algorithms
- ☐ Creating visual appealing graphs
- ☐ Connecting with Databases
- ☐ Building online dynamic reports or dashboards
- ☐ Send emails or push notification via R

# How to get R on your machine ?



To download R, visit official site of R - <http://www.r-project.org/>.

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



[Home]

Download

CRAN

## The R Project for Statistical Computing

### Getting Started

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. To [download R](#), please choose your preferred CRAN mirror.



Choose your country and click the link under it

Secure | <https://cran.r-project.org/mirrors.html>

<https://cran.biotools.fr/>

<http://cran.biotools.fr/>

<https://ftp.igh.cnrs.fr/pub/CRAN/>

<http://ftp.igh.cnrs.fr/pub/CRAN/>

<http://cran.irsn.fr/>

<https://cran.univ-paris1.fr/>

<http://cran.univ-paris1.fr/>

#### Germany

<https://ftp.gwdg.de/pub/misc/cran/>

<http://ftp.gwdg.de/pub/misc/cran/>

<https://cran.uni-muenster.de/>

<http://cran.uni-muenster.de/>

#### Greece

<https://ftp.cc.uoc.gr/mirrors/CRAN/>

<http://ftp.cc.uoc.gr/mirrors/CRAN/>

#### Hungary

<http://cran.rapporter.net/>

#### Iceland

<https://cran.hafro.is/>

<http://cran.hafro.is/>

#### India

<https://ftp.iitm.ac.in/cran/>

<http://ftp.iitm.ac.in/cran/>

IBDM, Marseille

IBDM, Marseille

Institut de Genetique Humaine, Montpellier

Institut de Genetique Humaine, Montpellier

French Nuclear Safety Institute, Paris

SAMM, Université Paris 1 Panthéon-Sorbonne

SAMM, Université Paris 1 Panthéon-Sorbonne

GWDG Göttingen

GWDG Göttingen

University of Münster, Germany

University of Münster, Germany

University of Crete

University of Crete

Rapporter.net, Budapest

Marine Research Institute

Marine Research Institute

Indian Institute of Technology Madras

Indian Institute of Technology Madras

# How to get R on your machine ?



## Click Download R for Windows (For Windows)



R logo

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](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)



## Click on base

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  $\geq$  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.

[old contrib](#)

Binaries of contributed CRAN packages for outdated versions of R (for R  $<$  2.13.x; managed by Uwe Ligges).

[Rtools](#)

Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.



## Click on “Download R.3.4.2 for Windows”



R logo

R-3.4.2 for Windows (32/64 bit)

[Download R 3.4.2 for Windows](#) (75 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

CRAN

# How to get R on your machine ?

---



## Definition of Sub directories

<a href="#"><u>base</u></a>	Binaries for base distribution. This is what you want to <a href="#"><u>install R for the first time</u></a> .
<a href="#"><u>contrib</u></a>	Binaries of contributed CRAN packages (for R $\geq$ 2.13.x; managed by Uwe Ligges). There is also information on <a href="#"><u>third party software</u></a> available for CRAN Windows services and corresponding environment and make variables.
<a href="#"><u>old contrib</u></a>	Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.13.x; managed by Uwe Ligges).
<a href="#"><u>Rtools</u></a>	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

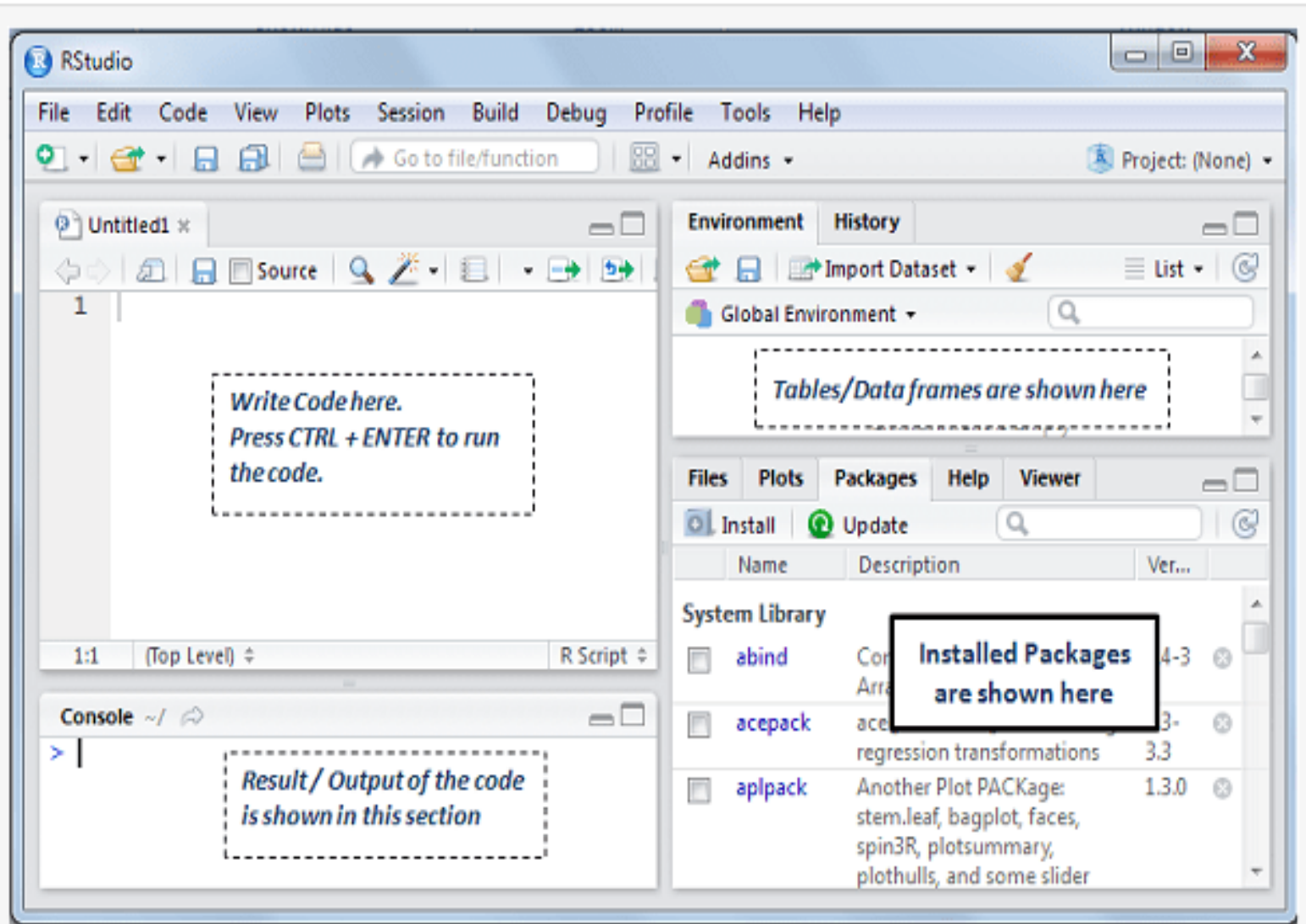
# Another Way to Use R



Time to introduce a jazzy IDE to work on R :-  
**R Studio**

***Download RStudio by clicking on***

***<https://www.rstudio.com/products/rstudio/download/>***



# Rstudio vs. Standard R

---

## What is RStudio and its benefits over standard R?

RStudio was built to make your life easy as a R programmer. It is available in open source for **FREE**. Unlike standard R, it supports various premium features such as intelligent code completion, syntax highlighting, structured R documentation, interactive debugging tool etc.

*In layman's term, RStudio is just an interface enhancement to standard R. Programming is similar in both the platforms.*

# Basics of R Programming

---



## Some Useful RStudio Shortcuts

1. Press CTRL + Enter to submit code
2. Press CTRL + SHIFT + C to comment/ uncomment code
3. Press CTRL + SHIFT + N to create a new R script

## Some Useful Standard R



1. Press F5 to submit the code
2. Press CTRL+N for new script
3. Press CTRL +O to open existing script
4. Press Ctrl +S to save a script
5. Press q() to quit R session

**R is a case sensitive language.  
FOO, Foo, and foo are three different  
objects**

# Basics of R Programming

---



## Write your first equation in R

Enter  $5*3$  in the RStudio code editor window and hit **CTRL + ENTER** in RStudio (or press **F5** in Standard R)

```
> 5 * 3
[1] 15
```



The **#** character at the beginning of a line signifies a **comment**.

```
> # Simple Calculator
> 3 * 3
[1] 9
```



The operator "**<=**" (without quotes) is equivalent to "**=**" sign . You can use either of the operators

```
> x <- 23
> y <- 22
> x + y
[1] 45
```

```
> x = 23
> y = 22
> x + y
[1] 45
```

# Basics of R Programming

---



R uses **forward slashes** instead of **backward slashes** in filenames (as shown in the image above).



The **setwd()** function tells R where you would like your files to save (changes the working directory).

```
> setwd("E:/R")  
> |
```



The **getwd()** function shows the working directory

```
> getwd()  
[1] "E:/R"
```



The **c function** is widely used to **combine values** to form a vector.

```
> x = c(1, 2, 3)  
> mean(x)  
[1] 2
```



R uses **NA** to represent Not Available, or missing values.

```
> x = c(1, 2, 3, NA, 5)  
> sum(x)  
[1] NA
```

# Basics of R Programming

---



To calculate sum excluding NA, use **na.rm= TRUE** (By default, it is FALSE).

```
> x = c(1,2,3,NA,5)
> sum(x)
[1] NA
> sum(x, na.rm = TRUE)
[1] 11
```



The form **1:10** generates the integers from 1 to 10.

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



Use **Seq()** command to generate sequence ,both continuous or .discrete

**Seq( from , to , by )**

```
> seq(12,56,by =1)
[1] 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
[26] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
> seq(100,0, by= -10)
[1] 100 90 80 70 60 50 40 30 20 10 0
> |
```

# Basics of R Programming

---



To get help for a certain function such as `sum`, use the form: **`help (sum)`**



Object names in R can be any length consisting of letters, numbers, underscores “`_`” or the period “`.`”  
Object names in R should begin with a letter.



## **Use `fix()` function**

You can use `fix()` function and give the name of an existing function, R shows you the code for that function in a Notepad window and you can type whatever you like.

***You can see the source code of any R function by using the following code.***

```
fix(colSums)
```



Retrieve your previous command

In R Console, you can retrieve it with the **UP arrow key** and edit it to run again.



Install and Load Packages

You can **install packages** by submitting the following line of code  
**`install.packages("sas7bdat")`**

# Basics of R Programming

---



To use the installed package, add the following line of code  
**library(forecast)**



To tell R which data set to use  
**attach(employee\_details)**

If you finish with that dataset and wish to use another, you can detach it with: `detach( employee_details)`



To clear console, press **Ctrl +L**

To list the objects that you have in your current R session use the function “ls”  
**ls()**

```
> ls()
[1] "a"      "age"    "b"      "birthw" "blood"  "calorie"
[7] "city"   "counts" "ctl"    "dead"   "dose"   "f"
[13] "g"      "glm.p84" "group"  "l0"     "l1"     "li"
[19] "lmcsl"  "lmwg"   "m"      "n"      "op"     "outcome"
[25] "p"      "sex"    "site"   "treatment" "trt"    "trtA"
[31] "trtB"   "type"   "weight" "x"      "y"      "z"
[37] "z.o4"   "z0"     "z1"     "z2"     "zi"     "zz"
_
```

# Basics of R Programming

---



To search for a specific pattern while listing the objects that you have in your current R session use  
`ls (pattern="z")`

```
> ls(pattern="z")
```

```
[1] "z"    "z.04" "z0"   "z1"   "z2"   "zi"   "zz"
```



To delete an object from your current session use **rm()**  
.This command will delete all objects from your memory

For selective deletion, use command **rm(zx,cv)**



List the Files in a Directory/Folder, use **list.files()**



## CHAPTER TWO

---

# Import Data in R

[LISTENDATA.COM](http://LISTENDATA.COM)

# Data Import in R

---

## Reading a comma-delimited text file (CSV)

✓ If you don't have the names of the variables in the first row  
**`employee_details <- read.csv("c:/employee_details.csv",  
header=FALSE)`**

✓ If you have the header row in the first row  
**`employee_details <- read.csv("c:/employee_details.csv",  
header=TRUE)`**

✓ If you want to set any value to a missing value  
**`employee_details <- read.csv("c:/employee_details.csv",  
header=TRUE, na.strings=".")`**

*In this case "." will be treated as missing*

**Note : R uses forward slash instead of backward slash in filename**

# Data Import in R

---

## Reading a tab-delimited text file

✓ If you don't have the names (headers) in the first row  
**`employee_details <- read.table(" mydata.txt ")`**

✓ If you have the names (headers) in the first row  
**`employee_details <-  
read.table("mydata.txt",sep="|",header=TRUE)`**

✓ If you want to set any value to a missing value  
**`employee_details <-  
read.table("mydata.txt",sep="|",header=TRUE,na.strings="-")`**

In this case “.” will be treated as missing

**Note : R uses forward slash instead of backward slash in filename**

# Data Import in R

---

## Reading Excel File

The best way to read an Excel file is to save it to a CSV format and import it using the CSV method

```
employee_details <- read.csv("c:/employee_details.csv",  
header=TRUE)
```

Use readxl package if you want to read in excel format only

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

*Specify sheet with a number or name*

```
read_excel("DataSet1.xlsx", sheet = "Employee-Attrition")  
read_excel("DataSet1.xlsx", sheet = 1)
```

If NAs are represented by something other than blank cells, set the na argument

```
read_excel(" DataSet1.xlsx ", na = "NA")
```

# Data Import in R

---



## For Importing Big files

*Use fread function of data.table package*

```
install.packages("data.table")
```

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

```
employee_details = fread("c:/employee_details.csv")
```



## Read SAS datafile in R

- Use package haven for reading sas datafile  
install.packages("haven")  
library(haven)

```
mydata = read_sas("callcenter.sas7bdat")
```