

## R Script Introduction

# R SCRIPT INTRODUCTION



# R Script Introduction

## In this session

- What is R?
- R current popularity rank
- Why use R language in Machine Learning?
- R Script interpreter installation
- R Studio installation
- Hello world
- Basic calculation
- Variable assignment
- Basic Operator
- Data Structure (Array, Matrix, List, Data Frame)
- If Statement
- For Loop
- Basic plotting

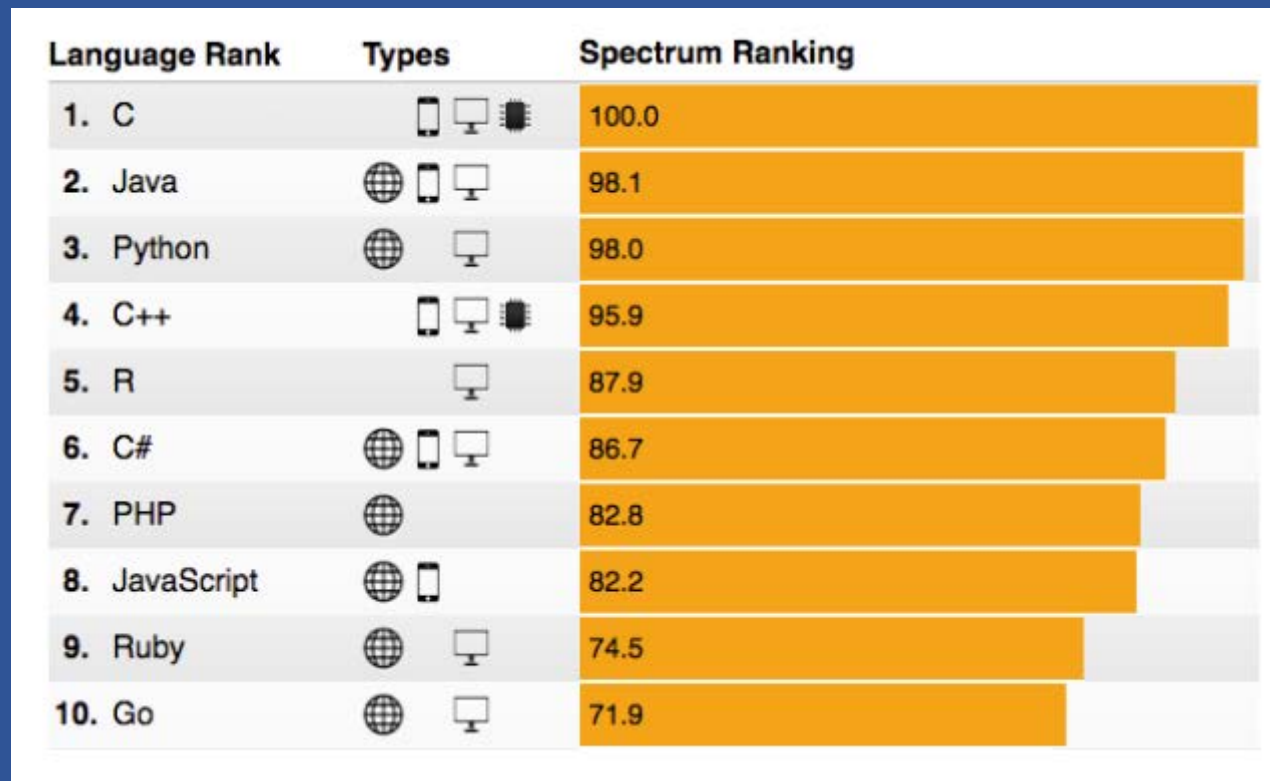
# R Script Introduction

## What is R?

- Computer language
- Interpreter
- Multi-paradigm: (OOP, imperative, functional, procedural)
- Typing: dynamic
- Good for: Statistical and graphics
- Origin: New Zealand
- Age: 23 (C# 17)
- Free Software (GNU project)
- Linux, Windows and MacOS
- One of the most powerful ML language
- Tool for ML exploration
- NOT for building a production model
- Supported in Azure ML Studio

# R Script Introduction

R current popularity rank



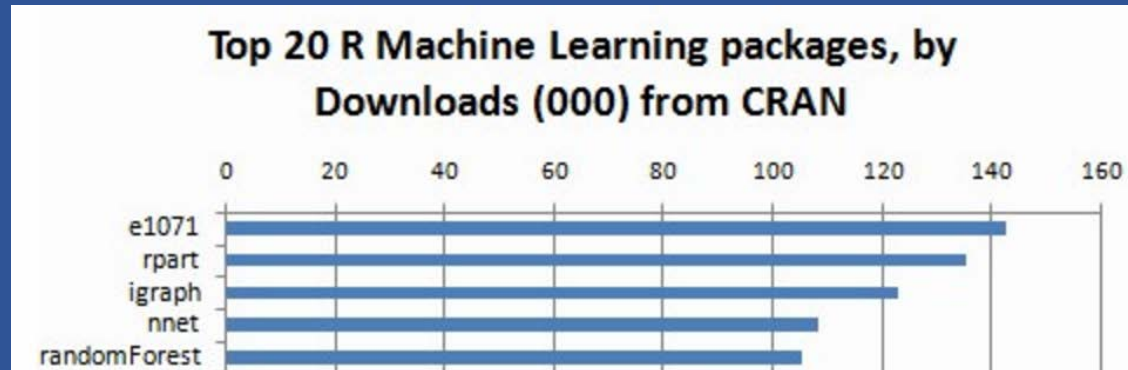
Source: The 2016 Top Programming Languages

<http://spectrum.ieee.org/static/interactive-the-top-programming-languages-2016>



# R Script Introduction

## Why use R language in Machine Learning?

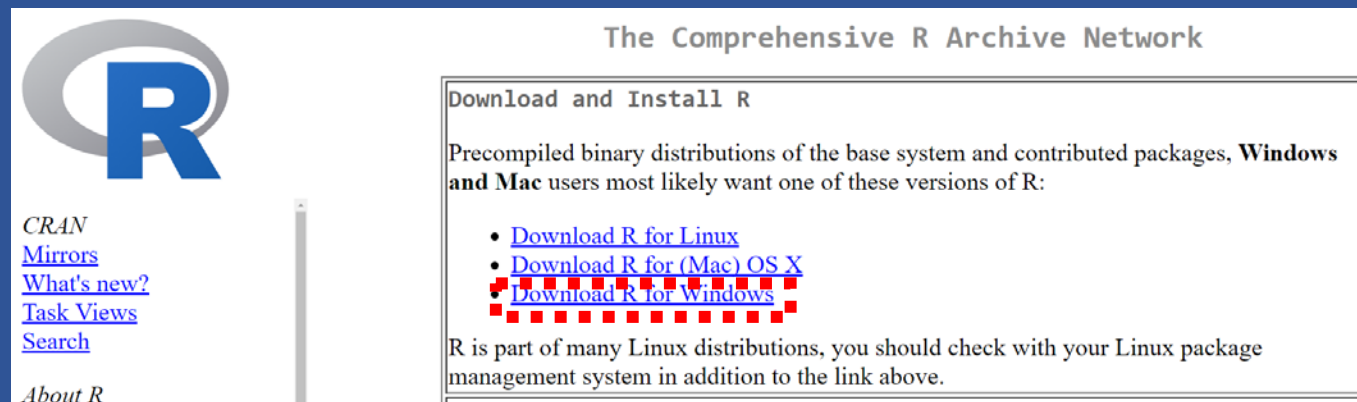


- Microsoft Azure ML support
- Free and open source
- Data Scientist's tools of trade
- Simple syntax
- Large community
- Over 7,800 package listed on CRAN
- Good ML packages (e1071, caret, etc.)
- Visualizations
- Full-set tools

# R Script Introduction

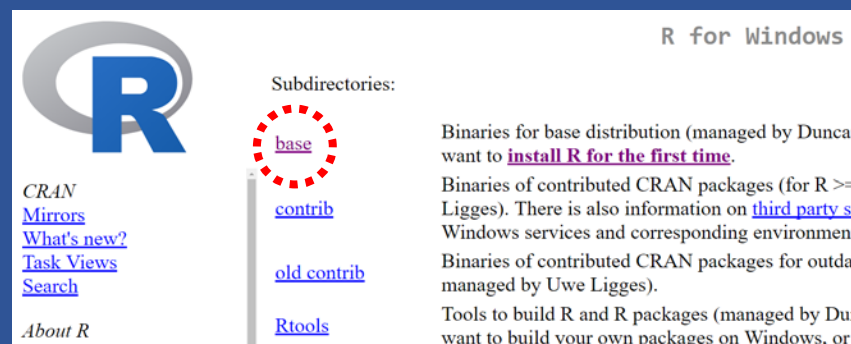
## R Script interpreter installation

1. Go to CRAN website <https://cran.rstudio.com>



2. Click [download R for Windows](#)

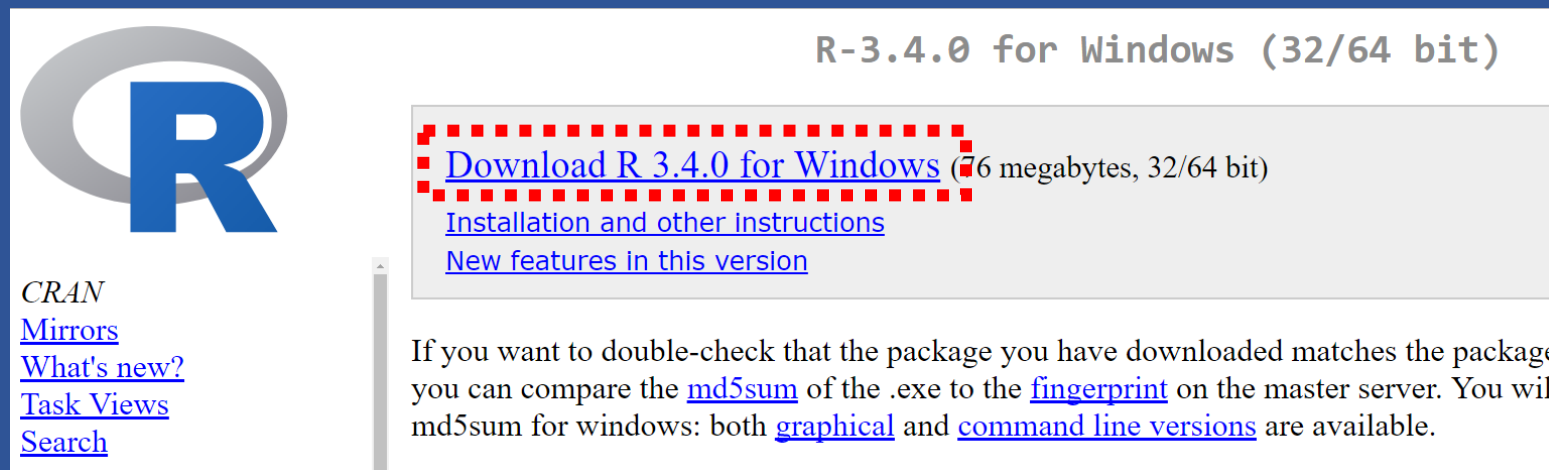
3. Click [base](#)



# R Script Introduction

## R Script interpreter installation

4. Click **Download R 3.4.0 for Windows (76 megabytes, 32/64 bit)**



When downloading done, open downloaded file to run setup



# R Script Introduction

## R Studio installation

Go to R Studio download page

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



Click RStudio Desktop FREE download





# R Script Introduction

## R Studio installation

1. Click RStudio 1.0.143 - Windows Vista/7/8/10

### Installers for Supported Platforms

#### Installers

RStudio 1.0.143 - Windows Vista/7/8/10

RStudio 1.0.143 - Mac OS X 10.6+ (64-bit)

RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (32-bit)

RStudio 1.0.143 - Ubuntu 12.04+/Debian 8+ (64-bit)

RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit)

RStudio 1.0.143 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit)

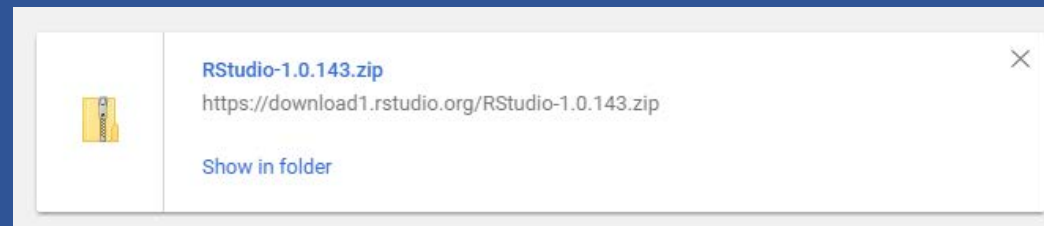
2. Create a folder **R Studio** on the desktop



# R Script Introduction

## R Studio installation

### 3. Open Zip file



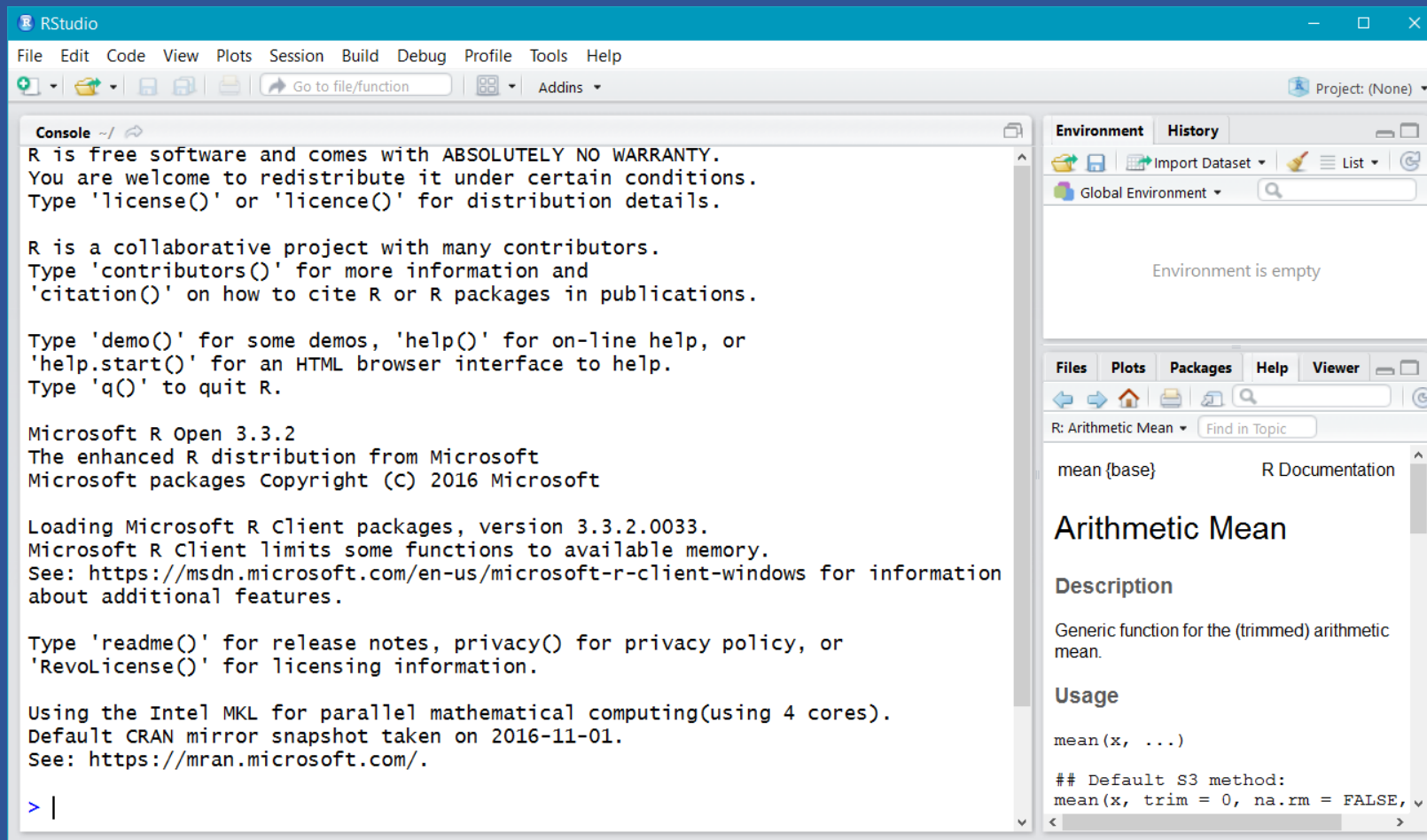
### 4. Drag & drop items from zip file to folder R Studio

Name	Type	Compressed size	Size
bin	File folder		
R	File folder		
resources	File folder		
www	File folder		
www-symbolmaps	File folder		
COPYING	File	12 KB	35 KB
INSTALL	File	3 KB	6 KB
NOTICE	File	47 KB	167 KB
README.md	MD File	1 KB	2 KB
SOURCE	File	1 KB	1 KB
VERSION	File	1 KB	1 KB

# R Script Introduction

## R Studio installation

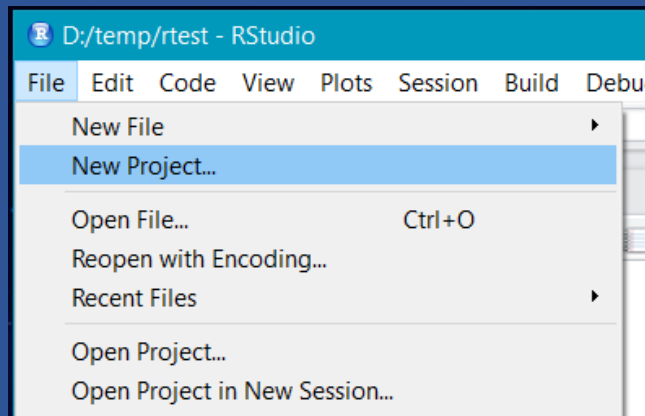
5. Click icon `C:\Desktop\R Studio\bin\rstudio.exe`



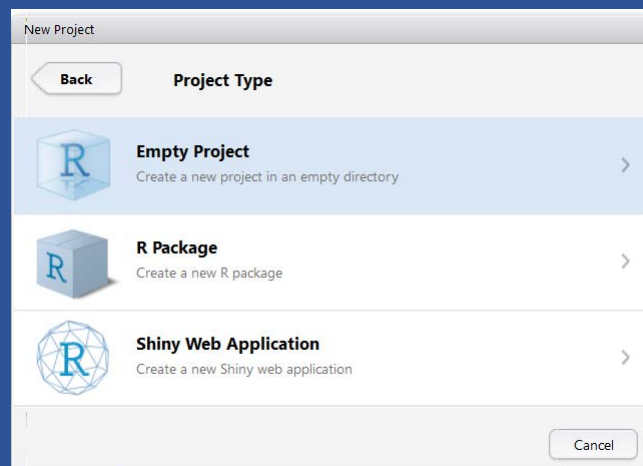
# R Script Introduction

Hello world

Create project



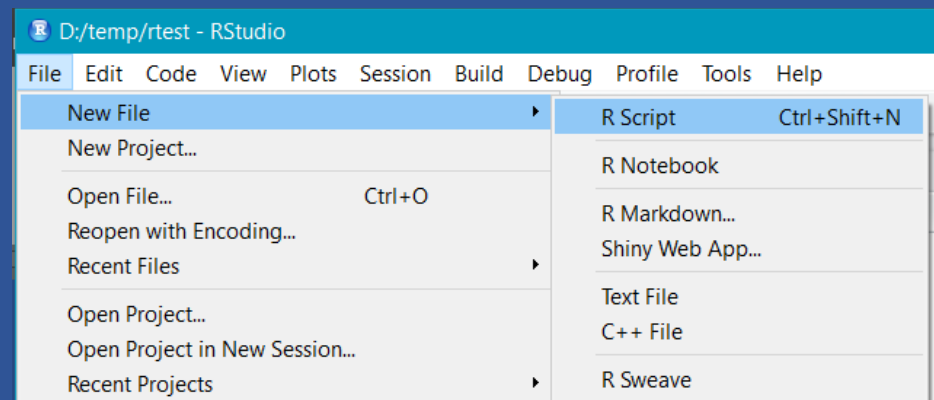
Click empty Project



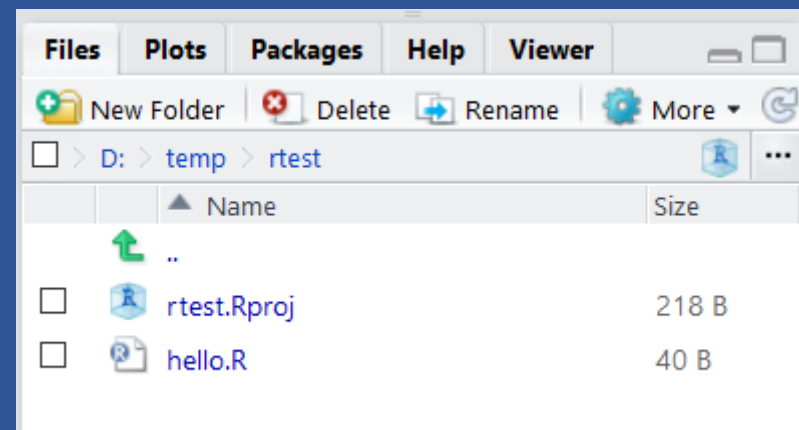
# R Script Introduction

## Hello world

Add R Script file to project



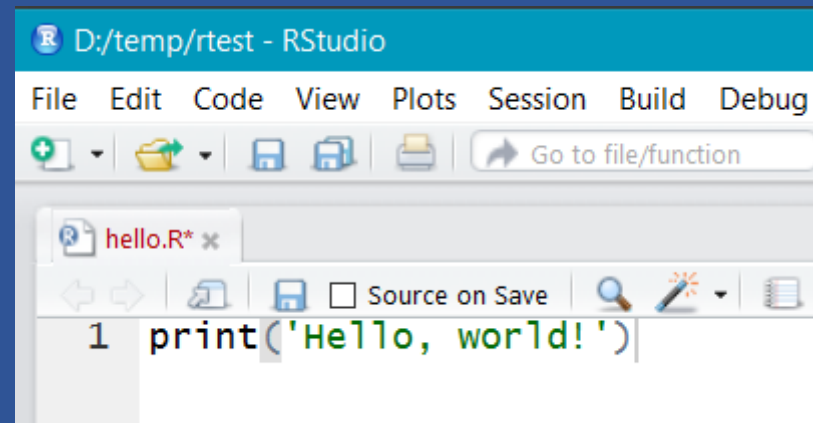
Save as hello.R



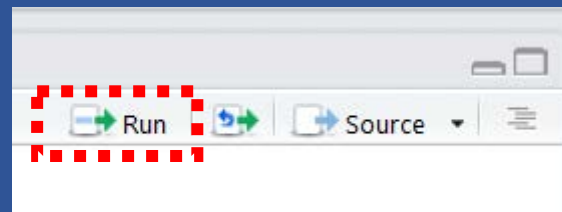
# R Script Introduction

Hello world

Enter code

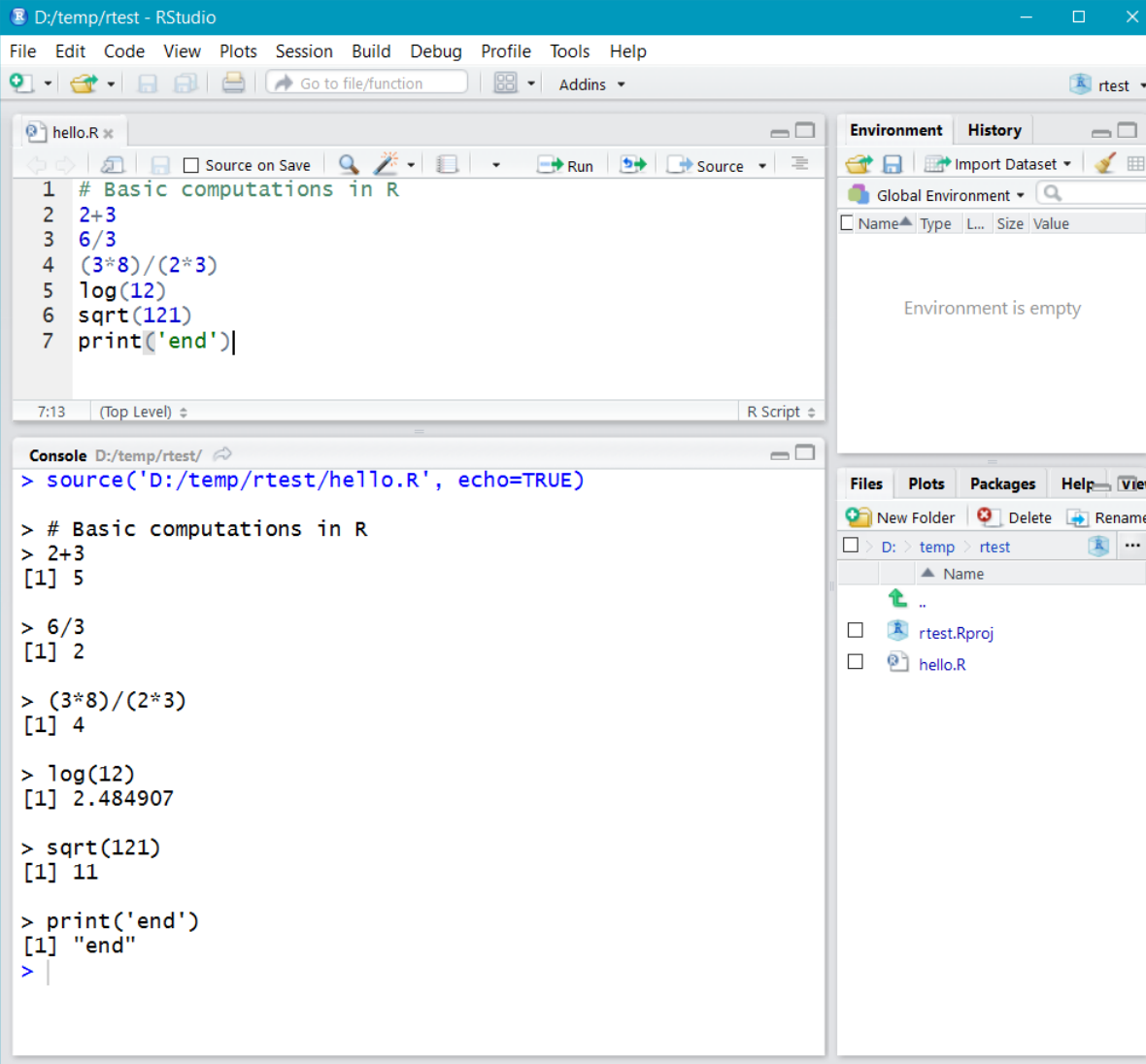


Click Run button to run Script



# R Script Introduction

## Basic calculation



The screenshot shows the RStudio interface with a script file named 'hello.R' and its execution results in the console. The script performs basic arithmetic and logical operations. The console output shows the results of these operations, including the execution of the script as a whole.

```
1 # Basic computations in R
2 2+3
3 6/3
4 (3*8)/(2*3)
5 log(12)
6 sqrt(121)
7 print('end')|
```

```
> source('D:/temp/rtest/hello.R', echo=TRUE)

> # Basic computations in R
> 2+3
[1] 5

> 6/3
[1] 2

> (3*8)/(2*3)
[1] 4

> log(12)
[1] 2.484907

> sqrt(121)
[1] 11

> print('end')
[1] "end"
> |
```

# R Script Introduction

## Variable assignment

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains a script named `hello.R` with the following code:

```
1 # Variable assignment
2 a <- 10
3 b <- 20
4 c <- a + b
5 c -> y
6 (y) + (a * b) -> m
```
- Console:** Shows the execution of the script using `source()` with `echo=TRUE`. The output is:

```
> source('D:/temp/rtest/hello.R', echo=TRUE)
> # Variable assignment
> a <- 10
> b <- 20
> c <- a + b
> c -> y
> (y) + (a * b) -> m
>
```
- Environment:** Displays the Global Environment with the following variables:

Name	Type	Length	Size	Value
a	numeric	1	48 B	10
b	numeric	1	48 B	20
c	numeric	1	48 B	30
m	numeric	1	48 B	230
y	numeric	1	48 B	30
- Files:** Shows the file explorer for the `D:/temp/rtest` directory, listing:
  - `..` (parent directory)
  - `rtest.Rproj` (218 B, May 31, 2017, 8:37 AM)
  - `hello.R` (78 B, May 31, 2017, 10:36 AM)



# R Script Introduction

## Basic data type

```
hello.R*
1 # Basic data type
2 rm(list = ls()) # clear objects
3 name <- 'laploy' # character
4 who <- paste(name, 'v.') # string concat
5 price <- 1500 # numeric
6 x <- 1; y <- 2; z <- 3
7 # c() = combine function
8 v <- c(x,y,z) # double vector
9 bar <- 6:11 # integer vector
10 e <- exp(1) # double form function
11
12 |
```

Environment		History				
Global Environment						
<input type="checkbox"/>	Name	Type	Length	Size	Value	
<input type="checkbox"/>	bar	integer	6	72 B	int [1:6] 6 7 8 9 10 ...	
<input type="checkbox"/>	e	numeric	1	48 B	2.71828182845905	
<input type="checkbox"/>	name	character	1	96 B	"laploy"	
<input type="checkbox"/>	price	numeric	1	48 B	1500	
<input type="checkbox"/>	v	numeric	3	72 B	num [1:3] 1 2 3	
<input type="checkbox"/>	who	character	1	104 B	"laploy v."	
<input type="checkbox"/>	x	numeric	1	48 B	1	
<input type="checkbox"/>	y	numeric	1	48 B	2	
<input type="checkbox"/>	z	numeric	1	48 B	3	

# R Script Introduction

## Basic Operator

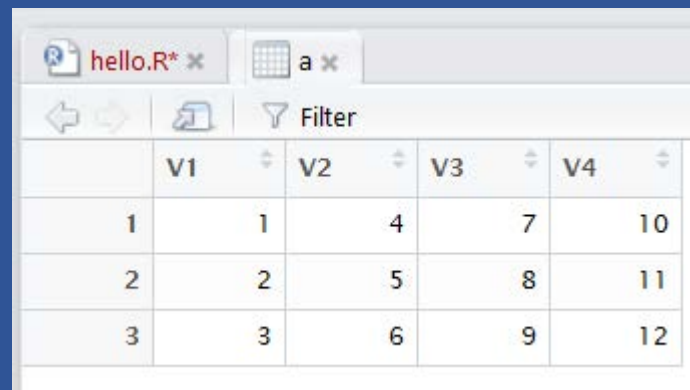
```
1 rm(list = ls()) # clear objects
2 # Arithmetic Operators
3 v <- c(1,2,3)
4 t <- c(2,2,1)
5 a <- v + t      # Add two vectors
6 s <- v - t      # subtracts
7 m <- v * t      # multiply
8 d <- v / t      # divide
9 r <- v %% t      # remainder
10 e <- v ^ t      # exponent
11 # Relation operators
12 g <- v > t      # is greater?
13 b <- v < t      # is less?
14 b <- v == t     # is equal?
15 f <- v != t     # is NOT equal?
```

# R Script Introduction

## Data Structure

### Array

```
hello.R* x a x
Source on Save Run
1 rm(list = ls()) # clear objects
2 # 2 dimensions Array
3 a <- array(c(1,2,3,4,5,6,7,8,9,10,11,12),dim=c(3,4))
4 b <- a[1,3]      # get row 1 column 3 value to b
5 a[1,3] <- 123    # write to array element
```

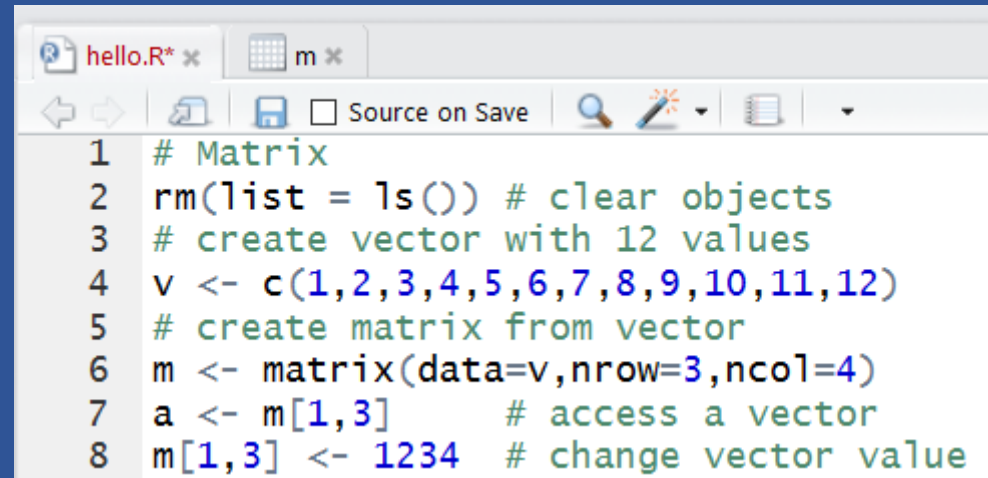


	V1	V2	V3	V4
1	1	4	7	10
2	2	5	8	11
3	3	6	9	12

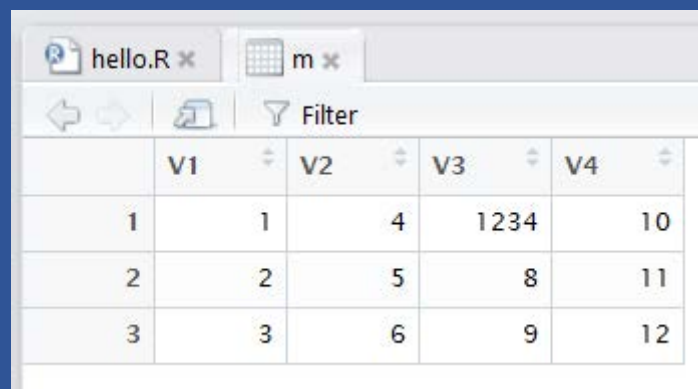
# R Script Introduction

## Data Structure

### Matrix



```
1 # Matrix
2 rm(list = ls()) # clear objects
3 # create vector with 12 values
4 v <- c(1,2,3,4,5,6,7,8,9,10,11,12)
5 # create matrix from vector
6 m <- matrix(data=v,nrow=3,ncol=4)
7 a <- m[1,3]      # access a vector
8 m[1,3] <- 1234   # change vector value
```

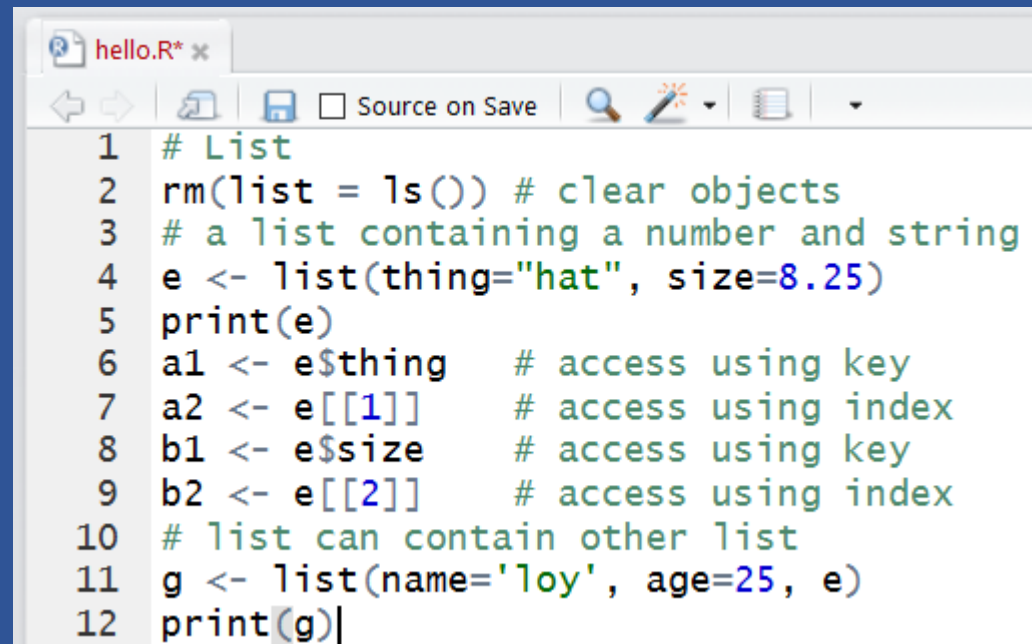


	V1	V2	V3	V4
1	1	4	1234	10
2	2	5	8	11
3	3	6	9	12

# R Script Introduction

## Data Structure

### List



```
1 # List
2 rm(list = ls()) # clear objects
3 # a list containing a number and string
4 e <- list(thing="hat", size=8.25)
5 print(e)
6 a1 <- e$thing    # access using key
7 a2 <- e[[1]]     # access using index
8 b1 <- e$size     # access using key
9 b2 <- e[[2]]     # access using index
10 # list can contain other list
11 g <- list(name='loy', age=25, e)
12 print(g)|
```

# R Script Introduction

## Data Structure

### Data Frame

	name	age	gender
1	Loy	19	M
2	Jim	17	M
3	Bo	22	F
4	Alice	12	F
5	Tan	24	M

```
1 # Data frame
2 rm(list = ls()) # clear objects
3 # create name vector variable
4 name <- c('Loy', 'Jim', 'Bo', 'Alice', 'Tan')
5 # create age vector variable
6 age <- c(19, 17, 22, 12, 24)
7 # create gender vector variable
8 gender <- c('M', 'M', 'F', 'F', 'M')
9 # create data frame from vector
10 student <- data.frame(name,age,gender)
11 student$gender == 'F' # look for female student
12 student$age > 20      # look for student older than 20
```

# R Script Introduction

## If Statement

```
1 # If Statement
2 rm(list = ls()) # clear objects
3 x <- 1
4 if (x == 1){
5   print('same')
6 } else if (x > 1){
7   print('bigger')
8 } else {
9   print('smaller')
10 }
11 # ifelse function
12 a = c(5,7,2,9)
13 ifelse(a %% 2 == 0,"even","odd")
```

```
Console D:/temp/rtest/ ↵
> source('D:/temp/rtest/hello.R')
[1] "same"
>
>
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd" "odd" "even" "odd"
> |
```

# R Script Introduction

## For Loop

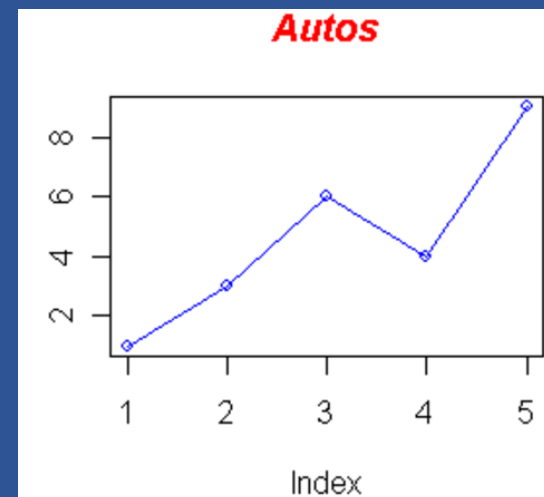
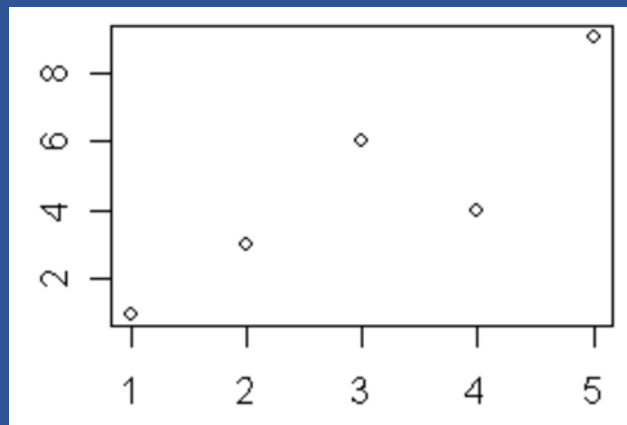
```
1 # For loop
2 x <- c(2,5,3,9,8,11,6)
3 # iterate through elements
4 for (v in x) {
5   print(v)
6 }
7 # count even element
8 count <- 0
9 for (val in x) {
10   if(val %% 2 == 0) count = count+1
11 }
12 print(count)
```



# R Script Introduction

## Basic plotting

```
1 # Line Charts
2 # Define the cars vector with 5 values
3 cars <- c(1, 3, 6, 4, 9)
4 # Graph the cars vector with all defaults
5 plot(cars)
6 #-----
7 # Graph cars using blue points overlayed by a line
8 plot(cars, type="o", col="blue")
9 # Create a title with a red, bold/italic font
10 title(main="Autos", col.main="red", font.main=4)|
```



# R Script Introduction

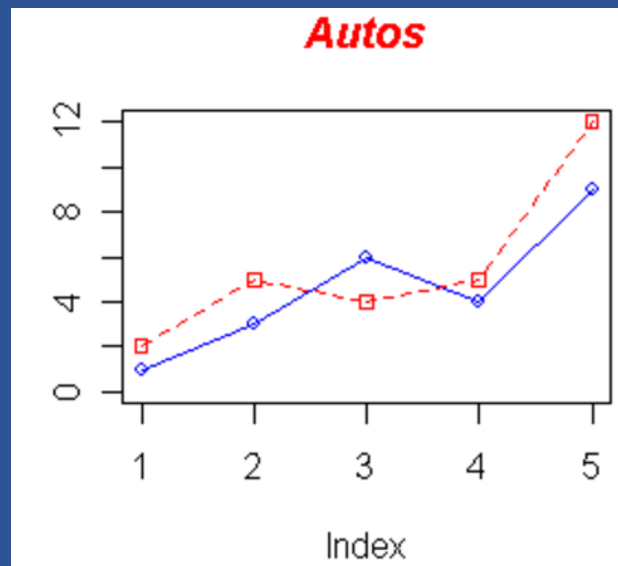
## Basic plotting

```
# Define 2 vectors
cars <- c(1, 3, 6, 4, 9)
trucks <- c(2, 5, 4, 5, 12)

# Graph cars using a y axis that ranges from 0 to 12
plot(cars, type="o", col="blue", ylim=c(0,12))

# Graph trucks with red dashed line and square points
lines(trucks, type="o", pch=22, lty=2, col="red")

# Create a title with a red, bold/italic font
title(main="Autos", col.main="red", font.main=4)
```



# R Script Introduction

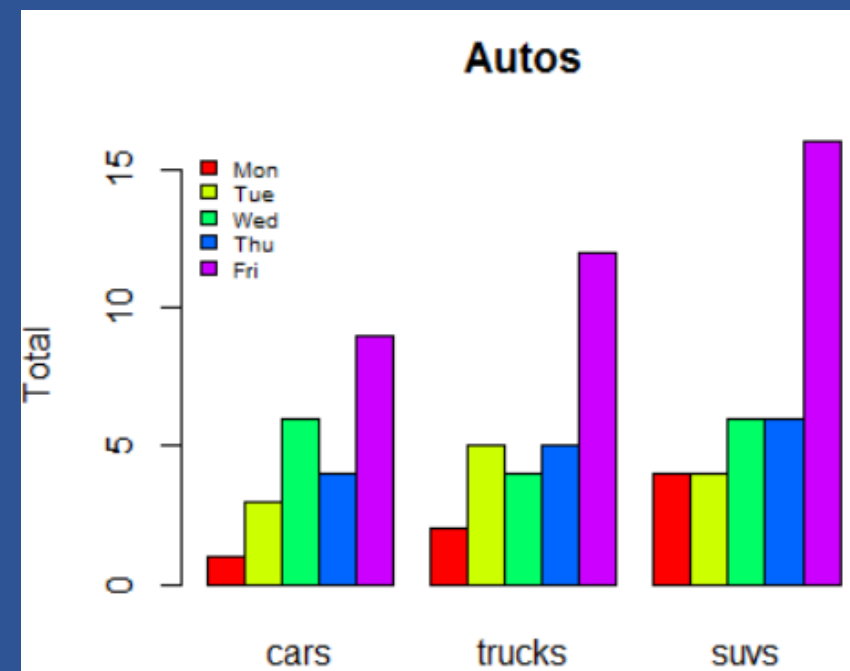
## Basic plotting

```

1 # Read values from tab-delimited autos.dat
2 autos_data <- read.table("d:/temp/autos.dat", header=T, sep="\t")
3 # Graph autos with adjacent bars using rainbow colors
4 barplot(as.matrix(autos_data), main="Autos", ylab= "Total", beside=TRUE, col=rainbow(5))
5 # Place the legend at the top-left corner with no frame
6 # using rainbow colors
7 legend("topleft", c("Mon","Tue","Wed","Thu","Fri"), cex=0.6, bty="n", fill=rainbow(5))

```

	cars	trucks	susvs
1	1	2	4
2	3	5	4
3	6	4	6
4	4	5	6
5	9	12	16



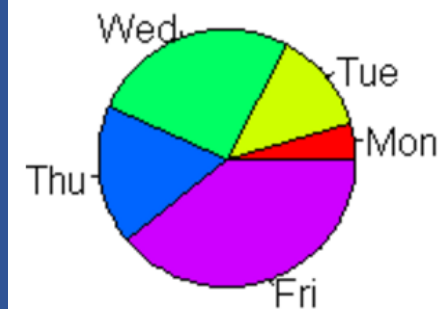
# R Script Introduction

## Basic plotting

```
1 # Define cars vector with 5 values
2 cars <- c(1, 3, 6, 4, 9)
3 # Create a pie chart for cars
4 pie(cars)
5 # -----
6 pie(cars, main="Cars", col=rainbow(length(cars)),
7     labels=c("Mon", "Tue", "Wed", "Thu", "Fri"))
```



**Cars**



**Cars**

# R Script Introduction

More information

## More information on R Script Basic

A Complete Tutorial to learn Data Science in R from Scratch

<https://www.analyticsvidhya.com/blog/2016/02/complete-tutorial-learn-data-science-scratch/>