R Programming

Presented By

Shobha Rani
Associate Professor

Master of Computer Applications
Dr. Ambedkar Institute of Technology, Bengaluru-56

What is R

- R is a language used for statistical computations, data analysis and graphical representation of data.
- **R** was created by **R**oss Ihaka and **R**obert Gentleman at the University of Auckland, New Zealand in 1990.
- **R** is **named** partly after the first names of the first two **R** authors partly as a play on the name of S.
- R was designed as a statistical platform for data cleaning, analysis, and representation.
- **R** allows you to integrate with other languages (C, C++).

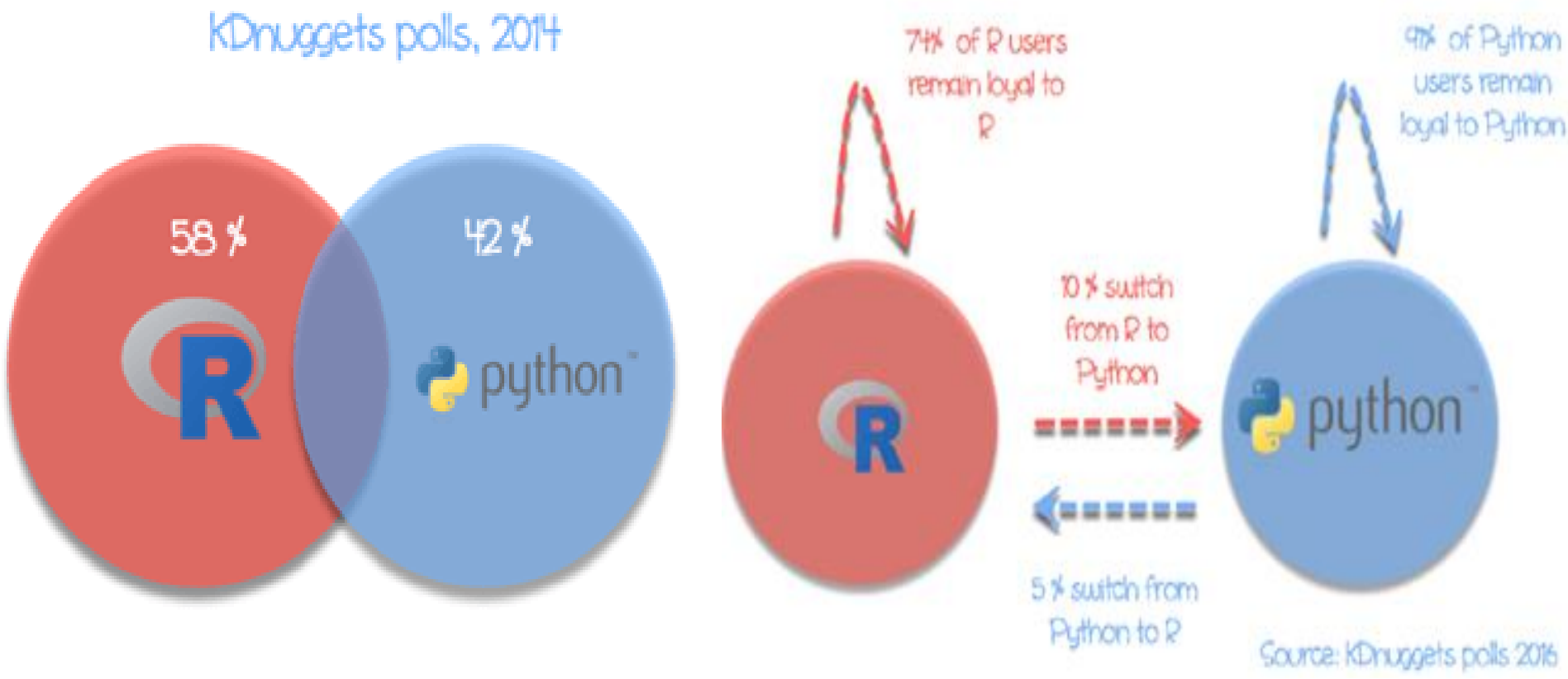
Why R

- It has been in use even before the word "Data Science" was coined.
- Statisticians and Data Scientists are most familiar with **R** than any other programming languages.

- Out of all surveyed data scientist, **40% prefer R, 34% prefer SAS and 26% Python.**

- **R** was built as a statistical language, it suits much **better** to do statistical learning.
- **Python is a better** choice for machine learning with its flexibility for production use, especially when the data analysis tasks need to be integrated with web applications

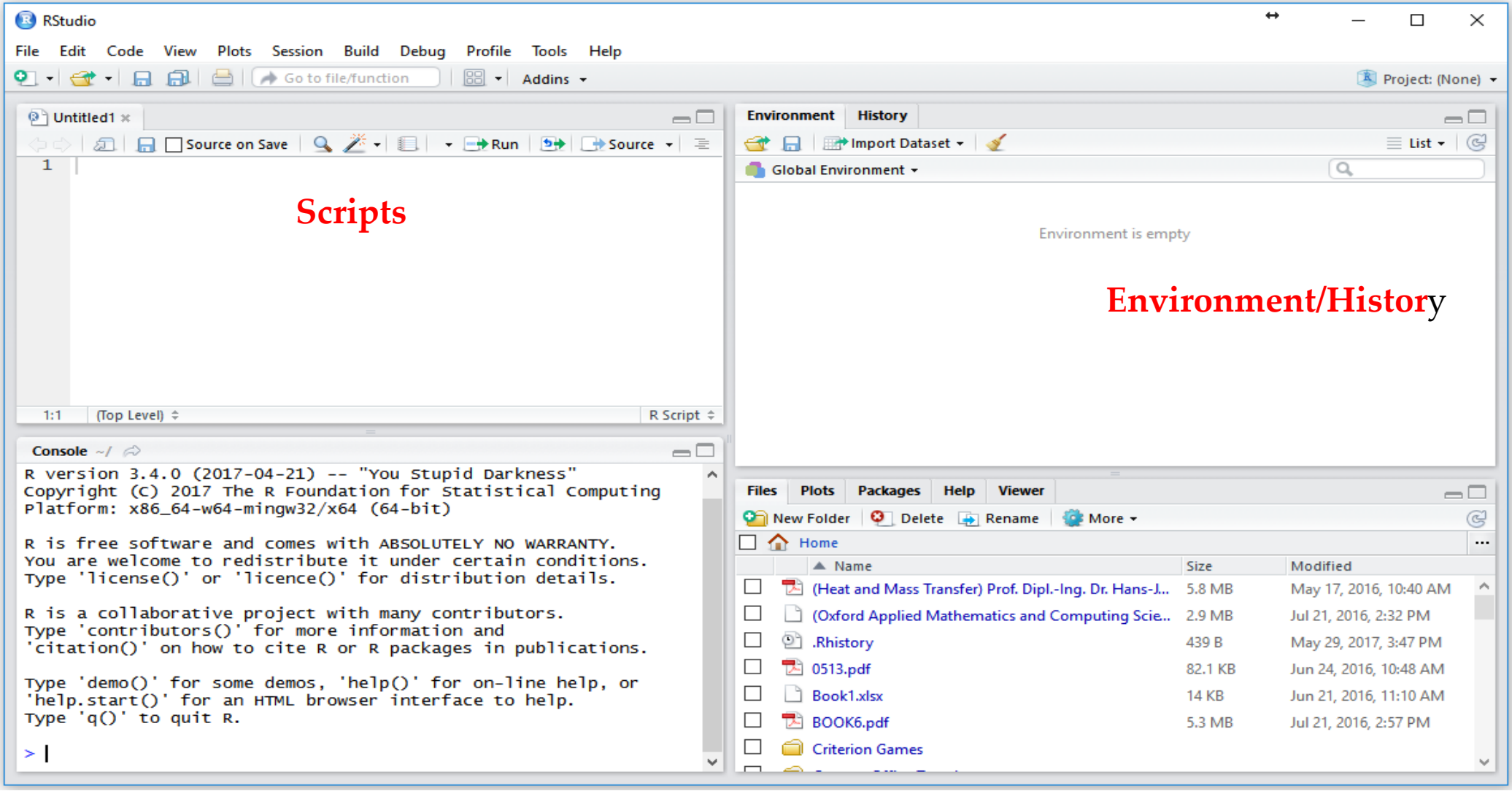
R Vs Python


KDnuggets polls, 2014
58 %
42 %
R
python
74% of R users remain loyal to R
97% of Python users remain loyal to Python
10 % switch from R to Python
5 % switch from Python to R
Source: KDnuggets polls 2016

Why R

- It offers an interface for many database like SQL and even spread sheets.
- R interface with NoSQL databases and analyze unstructured data.
- Developers can easily write their own software and distribute it in the form of add-on packages.
- It includes machine learning algorithms, linear regression, time series, statistical inference to name a few.
- Industries like Google, LinkedIn and Facebook, rely on R for many of their operations

First Look of R Studio



Set the working directory

- **setwd("directory path")**

  or

- ***Choose a suitable location by clicking on the indicated icon from*** *Files/plots/packages Window*

- ***Once directory is chosen, select the more icon and choose*** *"Set as Working Directory"*

Creation of R script File

- From **FILE** Menu

  Or

- From **NEW Icon** of Toolbar

Writing Script File

- Write R Script on to R file or can Run the Commands directly from the Console.

- Save the file to the location set as working directory

- Or

- Can use **Save Icon** from the Toolbar

Run the Script file

- Use **RUN** icon from the Toolbar
- Or

  Press Ctrl + Enter
- **Run can be used to execute selected lines**
- Source/ Source with echo is for a whole file
- Advantages – using Run :
  - troubleshooting/debugging
- Disadvantages – using Run :
  - For large section, console will be over populated and messy

Comments in R

- Add comments –single line
- For single line comment, insert '#' at the start of the line
- Add comments –Multi line

1) Select multiple lines using cursor, then press "**Ctrl + Shift + C**"

(OR)

2) Select multiple lines using cursor, click on "Code" in menu and select "**Comment/Uncomment lines**"

Clear the Console

- To Clear the console

  Use "**control +L**"

- Clear the environment –rm()

  -Single variable: Enter in console/R script : **rm(variable)**

  -All variables: Enter in console/R script : **rm(list=ls())**

Assignment Operations

- > is the prompt sign in R.
- The assignment operators are the left arrow with dash **<- and equal sign =.**

> x <- 20 assigns the value 20 to x.

> x = 20 assigns the value 20 to x.

- Initially only <- was available in R.

> x = 20 assigns the value 20 to x.

> y = x * 2 assigns the value 2*x to y.

> z = x + y assigns the value x + y to z.

Case Sensitive

- Capital and small letters are different.
- > X <- 20 and
- > x <- 20 are different

Variables & Constants

- **Rules**

➢ Allowed characters are Alphanumeric, '_' and '.'
➢ Always start with alphabets
➢ No special characters like !,@,#,$,....

- **<u>Predefined constants</u>**

	Constant	Symbol in R
1.	***Pi***	pi
2.	***letters***	a,b,c,.....x,y,z
3.	***LETTERS***	A,B,,.....,X,Y,Z
4.	***Months in a year***	month.name, month.abb

Arithmetic Operations

> 2+3 # Command

o/p: [1] 5 # Output

> 2*3 # Command

o/p: [1] 6 # Output

>2-3 # Command

o/p: [1] -1 # Output

> 3/2 # Command

o/p: [1] 1.5 # Output

> 2*3-4+5/6 # Command

o/p: [1] 2.8333 # Output

Arithmetic Operators

> 2^3 # Command

o/p: [1] 8 # Output

> 23 # Command**

o/p: [1] 8 # Output

> 2^0.5 # Command

o/p: [1] 1.414214 # Output

> 20.5 # Command**

o/p: [1] 1.414214 # Output $2^{1/2}$

> 2^-0.5 # Command

o/p: [1] 0.7071068 # Output

Basic data types

Basic data types	Values
1. **Logical**	TRUE and FALSE
2. **Integer**	Set of all integers, Z
3. **Numeric**	Set of all real numbers
4. **Complex**	Set of complex numbers
5. **Character**	“a”,”b”,”c”,….,”x”,”y”,”z”,”@”,”#”,”$”, “”,”*”, “1”,”2”,… etc..

Basic objects

Object	Values
1. ***Vector***	Ordered collection of same data types
2. ***List***	Ordered collection of objects
3. ***Data frame***	Generic tabular object
4. Matrices	
5. Arrays	
6. Data Frames	

Vectors

- Vector : an ordered collection of basic data types of given length
- All the elements of a vector must be of same data type

Example:

```
X = c(2.3,4.5,6.7,8.9)
print(X)
```

Built-in Functions

- **min()** **Minimum value**
- **max()** **Maximum value**
- **abs()** Absolute value
- **sqrt()** Square root
- **round(), floor(), ceiling()** Rounding, up and down
- **sum(), prod()** Sum and product
- **log(), log10(), log2()** Logarithms
- **exp()** Exponential function
- **sin(), cos(), tan()** Trigonometric functions

Math Functions

- abs(c(-3,-6,-1,9)
- max(c(4.5,6.9,23.4,12.7)
- round(2.8)

Types of Operators

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Miscellaneous Operators

Conditional statements

- **if (condition) {executed commands if condition is TRUE}**

- **if (condition) {executed commands if condition is TRUE}**
  else { executed commands if condition is FALSE }

- **ifelse(test, yes, no)**

Example

```
> x <- 1:10
>x
[1] 1 2 3 4 5 6 7 8 9 10
> ifelse( x<6, x^2, x+1 )
[1] 1 4 9 16 25 7 8 9 10 11
```

Loops

1. **for loop**
2. while loop
3. repeat loop

Syntax

for (name in vector) {commands to be executed}

Example

- **> for (i in 1:5) { print(i^2) }**

o/p: 1 4 9 16 25

- **> for (i in c(2,4,6,7)) { print(i^2) }**
- 4 16 36 49

While Loop

1.while(condition){ commands to be executed as long as condition is TRUE }

Example

```
> i <- 1
> while (i<5) {
print(i^2)
 i <- i+2 }
```

2. repeat{ commands to be executed }

```
 i <- 1
 > repeat{
 print( i^2 )
 i <- i+2
 if ( i > 10 ) break
 }
```

sequence

- A sequence is a set of related numbers, events, movements, or items that follow each other in a particular order.

Syntax

>**seq()**

- **seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)), length.out = NULL, along.with = NULL, ...)**

Example:

✓ **seq(from=2, to=4)**

✓ **seq(from=-4, to=4)**

Sequence with constant increment

- Generate a sequence from 10 to 20 with an increment of 2 units

- > seq(from=10, to=20, by=2)
- [1] 10 12 14 16 18 20

- Generate a sequence from 3 to -2 with a decrement of 0.5 units
- > seq(from=3, to=-2, by=-0.5)

- Sequences with a predefined length with default increment +1
- > **seq(to=10, length=10)**
- [1] 1 2 3 4 5 6 7 8 9 10

- Sequences with a predefined length with constant fractional increment
- > **seq(from=10, length=10, by=0.1)**

Lists

- List : a generic object consisting of an ordered collection of objects
- A list could consist of a numeric vector, a logical value, a matrix, a complex vector, a character array, a function, and so on

Example:

```
ID = c(1,2,3,4)
std.name =c("Rachana","Hasini","Shaila","Danush")
num.std = 4
std.list = list(ID, std.name, num.std)
print(std.list)
```

Accessing components (indices)

- To access top level components, use double slicing operator " [[ ]]" and for lower/inner level components use "[ ]" along with "[[ ]]"

Example:

```
print(std.list[[1]])
print(std.list[[2]])
print(std.list[[1]][1])
print(std.list[[2]][1])
```

Thanks for your attention!

Any questions?