



Changes in the activity of the active and passive market in various countries. Established positive trends in various market segments.



Distribution of market share among the major industry players.



Distribution of market share among the major industry players. Player A has 55% share, Player B has 35% share, and Player C has 10% share. A business change in the market will be observed, resulting in the market will be dominated by one or two dominant types of players than major players.

Introduction to R

DS Part IV – Python Hands on

Sasken Training, Adyar , 9840014739 Chennai – 600 020



Agenda

-  Introduction to R
-  R and Rstudio set up
-  Basic data types
-  Data Structures in R
-  Variables and Control structures
-  Functions
-  Modules and Packages

What is R ?

- ▶ R is a programming language for statistical Computing and graphics
- ▶ R is free and open source , allowing anyone to use it and, importantly, to modify it.
- ▶ It was originated from S language which was developed by John Chambers in the Bell laboratories in the 80's
- ▶ R is cross platform, it can run on many operating systems and different hardwares
- ▶ R is extensible and offers rich functionality for developers to build their own tools and methods for analyzing data

Advantages :

- ▶ It provides vast number of packages
- ▶ R has over 4800 packages available from multiple repositories specializing in topics like econometrics, data mining, spatial analysis, and bio-informatics.
- ▶ R has active user groups where questions can be asked and are often quickly responded to

Disadvantages :

- Its main disadvantage is it's not scalable

Rstudio

- ▶ RStudio is an open-source Integrated Development Environment (code editor) for R
- ▶ Rstudio makes R easier to use, it includes code editor, debugging & visualization tools
- ▶ It has easy access to R help
- ▶ Provides completion facility
- ▶ Rstudio offers many handy keyboard features

<https://support.rstudio.com/hc/en-us/articles/200711853-Keyboard-Shortcuts>



Sasken Training, Adyar , 9840014739, Chennai – 600 020

R & Rstudio Installation

- ▶ Download & Install R from this below URL
<https://www.r-project.org/>

- ▶ Download & Install Rstudio from this below URL
<https://www.rstudio.com/products/rstudio/download/>

R software download site

The screenshot shows a web browser window displaying the official R Project website at <https://www.r-project.org>. The page title is "The R Project for Statistical Computing". On the left, there is a sidebar with links for "Home", "Download", "CRAN", "R Project" (with sub-links for About R, Logo, Contributors, What's New?, Reporting Bugs, Development Site, Conferences, Search), "R Foundation" (with sub-links for Foundation, Board, Members, Donors, Donate), "Help With R" (with sub-links for Getting Help), and "Documentation" (with sub-links for Manuals and FAQs). The main content area starts with a "Getting Started" section, which includes a paragraph about R being a free software environment for statistical computing and graphics, followed by a link to "download R". This link is circled in red. Below this, there is a section titled "News" containing a bulleted list of recent events and releases.

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

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 3.4.1 (Single Candle) prerelease versions** will appear starting Tuesday 2017-06-20. Final release is scheduled for Friday 2017-06-30.
- **R version 3.4.0 (You Stupid Darkness)** has been released on Friday 2017-04-21.
- **R version 3.3.3 (Another Canoe)** has been released on Monday 2017-03-06.
- **useR! 2017** (July 4 - 7 in Brussels) has opened registration and more at <http://user2017.brussels/>
- Tomas Kalibera has joined the R core team.
- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Dianne Cook, Julie Josse, Tomas Kalibera, and Balasubramanian Narasimhan.
- **The R Journal Volume 8/1** is available.
- The **useR! 2017** conference will take place in Brussels, July 4 - 7, 2017.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.

Sasken Training, Adyar , 9840014739, Chennai – 600 020

R Software CRAN mirror

The screenshot shows a web browser window titled "CRAN - Mirrors". The URL in the address bar is <https://cran.r-project.org/mirrors.html>. The page displays a list of CRAN mirrors categorized by country. The countries listed include Greece, Hungary, Iceland, India, Indonesia, Iran, Ireland, Italy, Japan, and Korea. The India section is highlighted with an orange background. The Indian mirrors listed are:

Country	Mirror Details
India	Indian Institute of Technology Madras Indian Institute of Technology Madras

Sasken Training, Adyar , 9840014739, Chennai – 600 020

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](#)

R is part of many Linux distributions. You should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (Friday 2017-04-21, You Stupid Darkness) [R-3.4.0.tar.gz](#).
read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).

The image shows two side-by-side web browser windows. Both windows have a blue header bar with a back button, forward button, refresh button, and a URL bar showing 'Secure | https://cran.cnr.berkeley.edu'. The left window title is 'R for Windows' and the right window title is 'R-3.4.0 for Windows (32/64 bit)'. Both windows feature a large blue 'R' logo on the left.

Left Window (R for Windows):

- Subdirectories:**
 - [base](#): Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).
 - [contrib](#): Binaries of contributed CRAN packages (for R >= 2.11.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 < 2.11.x; managed by Uwe Ligges).
 - [Rtools](#): Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.
- Text:**

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

Right Window (R-3.4.0 for Windows):

- Section:** Download R 3.4.0 for Windows (76 megabytes, 32/64 bit)
- Links:**
 - [Installation and other instructions](#)
 - [New features in this version](#)
- Text:**

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.
- Section:** Frequently asked questions
- List:**
 - [Does R run under my version of Windows?](#)
 - [How do I update packages in my previous version of R?](#)
 - [Should I run 32-bit or 64-bit R?](#)
- Text:**

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.
- Section:** Other builds
- List:**
 - Patches to this release are incorporated in the [r-patched snapshot build](#).
 - A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).

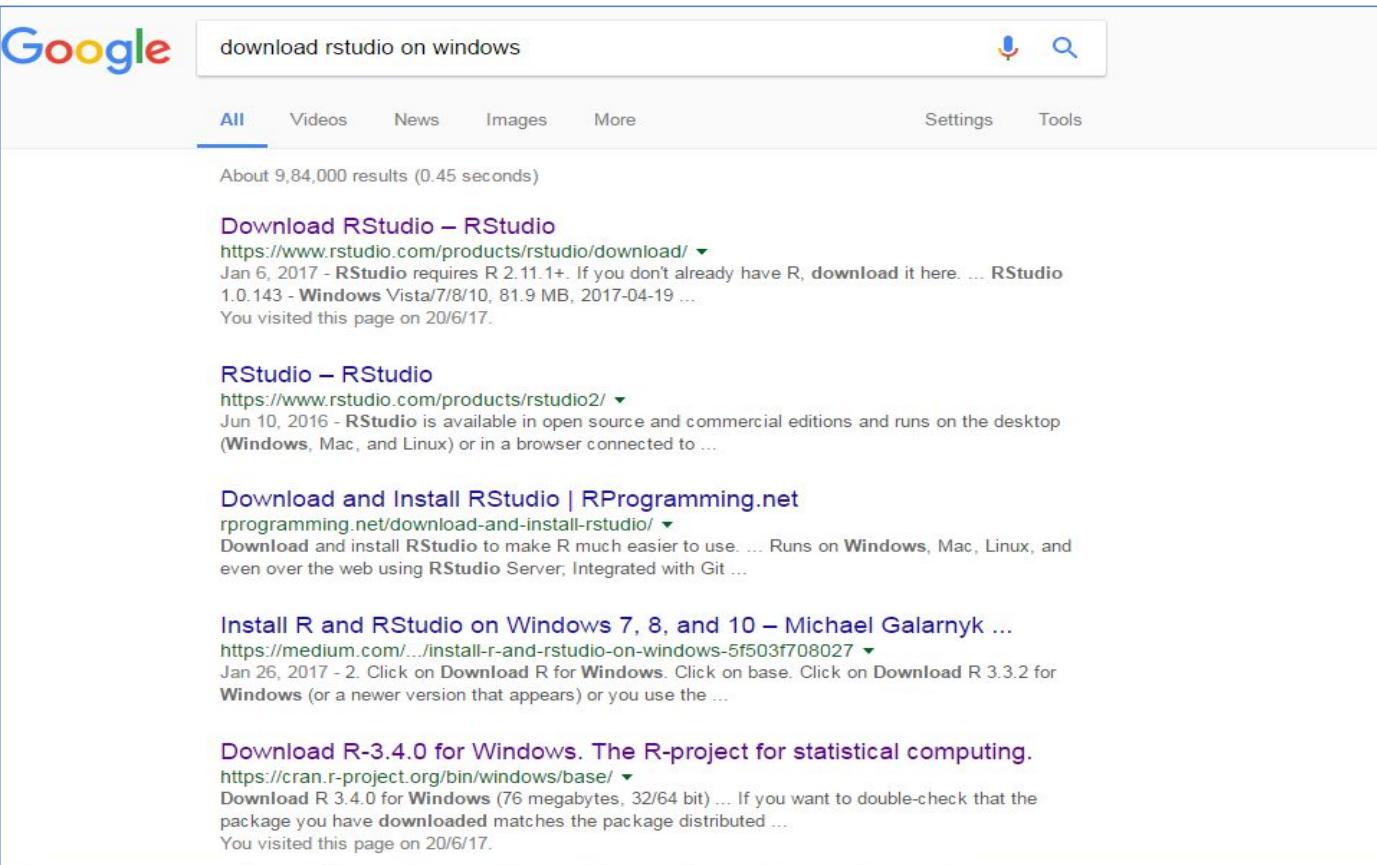
R Console

The screenshot shows the R GUI interface with several panes:

- R Console:** Displays the R startup message and a command prompt (> |) followed by the code entered in the screenshot.
- History:** Shows the history of commands run in the session.
- Environment:** Shows that the environment is empty.
- Packages:** Shows package management options.
- Viewer:** Shows a file browser with the following contents:

Name	Size
..	13 B
.gitignore	3.4 KB
.httr-oauth	8.1 MB
.RData	14.8 KB
.Rhistory	110.2 KB
_final_data.R	

Rstudio Download



Google search results for "download rstudio on windows". The results page shows five links related to RStudio download:

- Download RStudio – RStudio**
<https://www.rstudio.com/products/rstudio/download/> ▾
Jan 6, 2017 - RStudio requires R 2.11.1+. If you don't already have R, download it here. ... RStudio 1.0.143 - Windows Vista/7/8/10, 81.9 MB, 2017-04-19 ...
You visited this page on 20/6/17.
- RStudio – RStudio**
<https://www.rstudio.com/products/rstudio2/> ▾
Jun 10, 2016 - RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to ...
- Download and Install RStudio | RProgramming.net**
rprogramming.net/download-and-install-rstudio/ ▾
Download and install RStudio to make R much easier to use. ... Runs on Windows, Mac, Linux, and even over the web using RStudio Server; Integrated with Git ...
- Install R and RStudio on Windows 7, 8, and 10 – Michael Galarnyk ...**
<https://medium.com/.../install-r-and-rstudio-on-windows-5f503f708027> ▾
Jan 26, 2017 - 2. Click on Download R for Windows. Click on base. Click on Download R 3.3.2 for Windows (or a newer version that appears) or you use the ...
- Download R-3.4.0 for Windows. The R-project for statistical computing.**
<https://cran.r-project.org/bin/windows/base/> ▾
Download R 3.4.0 for Windows (76 megabytes, 32/64 bit) ... If you want to double-check that the package you have downloaded matches the package distributed ...
You visited this page on 20/6/17.

Rstudio Download page

The screenshot shows a web browser displaying the RStudio download page at <https://www.rstudio.com/products/rstudio/>. The page has a blue header with the text "RStudio Download page". Below the header, there is a video player showing a video titled "R Studio™" with a duration of 01:31. A "CLICK HERE TO SEE ADDITIONAL FEATURES" button is located below the video player. On the left side of the page, there are two main options: "Desktop" and "Server". The "Desktop" option is circled in black, indicating it is the selected or recommended edition. The "Server" option is also available for those who want to centralize access and computation. The rest of the page contains descriptive text about RStudio's features and availability.

Sasken Training, Adyar , 9840014739, Chennai – 600 020

The screenshot shows a web browser window displaying the RStudio product page. The URL in the address bar is <https://www.rstudio.com/products/rstudio/#Desktop>. The page compares the **Open Source Edition** and **Commercial License**.

Open Source Edition

- Access RStudio locally
- Syntax highlighting, code completion, and smart indentation
- Execute R code directly from the source editor
- Quickly jump to function definitions
- Easily manage multiple working directories using projects
- Integrated R help and documentation
- Interactive debugger to diagnose and fix errors quickly
- Extensive package development tools

Commercial License

All of the features of open source; plus:

- A commercial license for organizations not able to use AGPL software
- Access to priority support

Overview

Support Community forums only

License AGPL v3

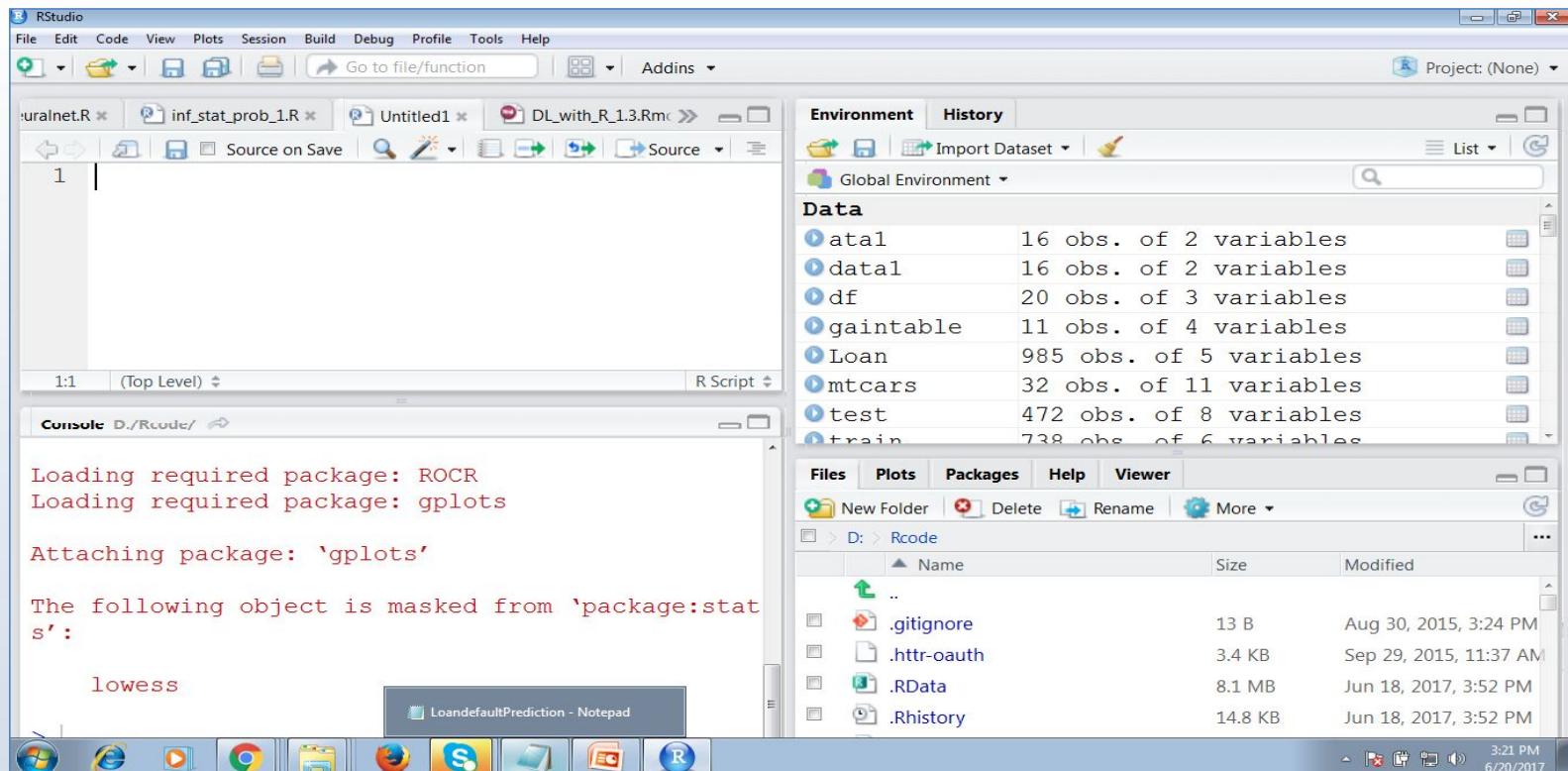
Pricing Free

DOWNLOAD RSTUDIO DESKTOP (button circled in red)

BUY NOW

RStudio License Agreement

RStudio - Window pane Layout



Installing packages in R

► What is a package ?

- Packages are collections of R functions, data , and complied code in a well-defined format.
- The directory where packages are stored is called the library
- R comes with a standard set of packages , others are available for download and installation.

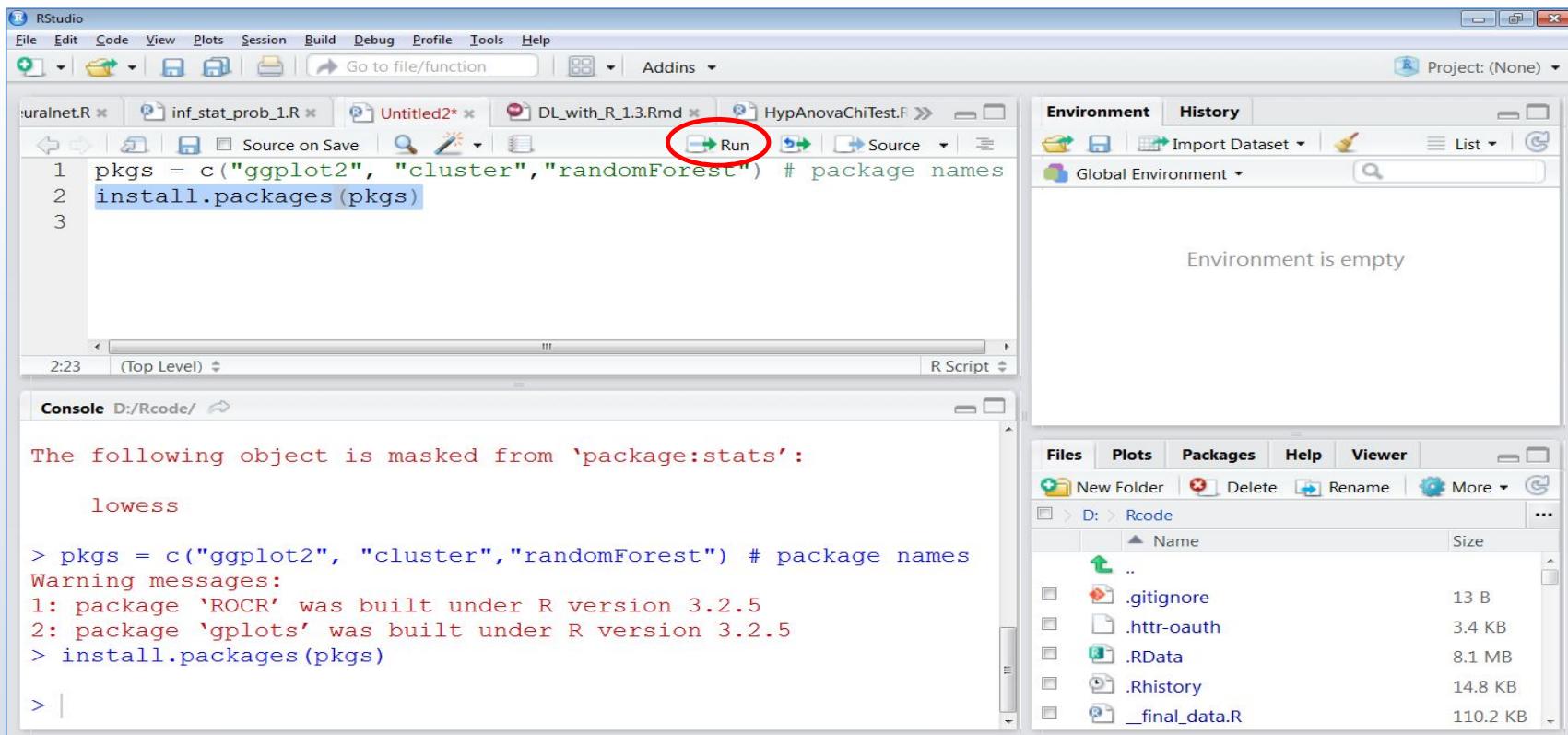
► How to install packages in R?

```
pkgs = c("ggplot2", "cluster", "randomForest") # package names  
install.packages(pkgs)
```

► How to update R packages ?

- update.packages()

Installing packages in RStudio





Changes in the activity of the active and passive market in uncertain, established positive trends in various market segments.



Distribution of market share among the major industry players.



Distribution of market share among the major industry players. The 16% share of MNCs and SMEs is 7 over 100 and 200 parts respectively. Note: A business change in the market environment will be observed by year-end due to the entry of major new players.

Projected sales of main products in 2013

Getting started with R

DS Part IV – Python Hands on

Sasken Training, Adyar , 9840014739 Chennai – 600 020



Getting Started with R

Working Directory

- ▶ Is the directory where all input to and output from R are stored
Function `getwd()` is used to get the current working directory.
 - Usage: Type `getwd()` at the command line
- ▶ Function `setwd()` is used to assign a desired path as working directory
 - Usage: Type `setwd({Path})` at the command line.
 - Example (Mac): `setwd("D:/workingSet")`
 - Example (Windows): `setwd("C:\Program Files\R")`
- ▶ In R Studio, access Tools > Set Working Directory from the menu –
Three options available.
Select “Choose Directory” to assign a desired path

What are packages ?

- ▶ Collections of R functions, data , and compiled code in a well-defined format
- ▶ The directory where packages are stored on your computer is called the library
- ▶ To view the installed packages ?
`library()`
- ▶ How to install packages or install a package with its dependencies ?
`install.packages("MASS", dependencies = TRUE)`
- ▶ How to load the package ?
▶ `library(MASS)`
- ▶ How to update the package ?
▶ `update.packages()`

Getting help in R

► Just type ? Prior to the command

? as.integer()

Assignment

Assignment

- ➔ To assign a value or a formula to a variable, the assignment operator [<- or =] is used
 - Usage: a {Value} or a {Formula}. Example: a <- 3
- ✓ Continuation prompt
 - ➔ In some cases, an assignment needs to be made over a couple of lines.
For example, when using two input vectors in the same formula
 - ➔ In these cases, the Enter key may be used to add a line to the command.
- ✓ Case Sensitivity
 - ➔ All function names, variable names, keywords etc in R are case-sensitive
 - ➔ Example: a 3 and A 4 will retain their case-specific values
- ✓ Assignment rules
 - ➔ Blank spaces are ignored
 - ➔ No standard names or key words should be used as object names or variables



Comments

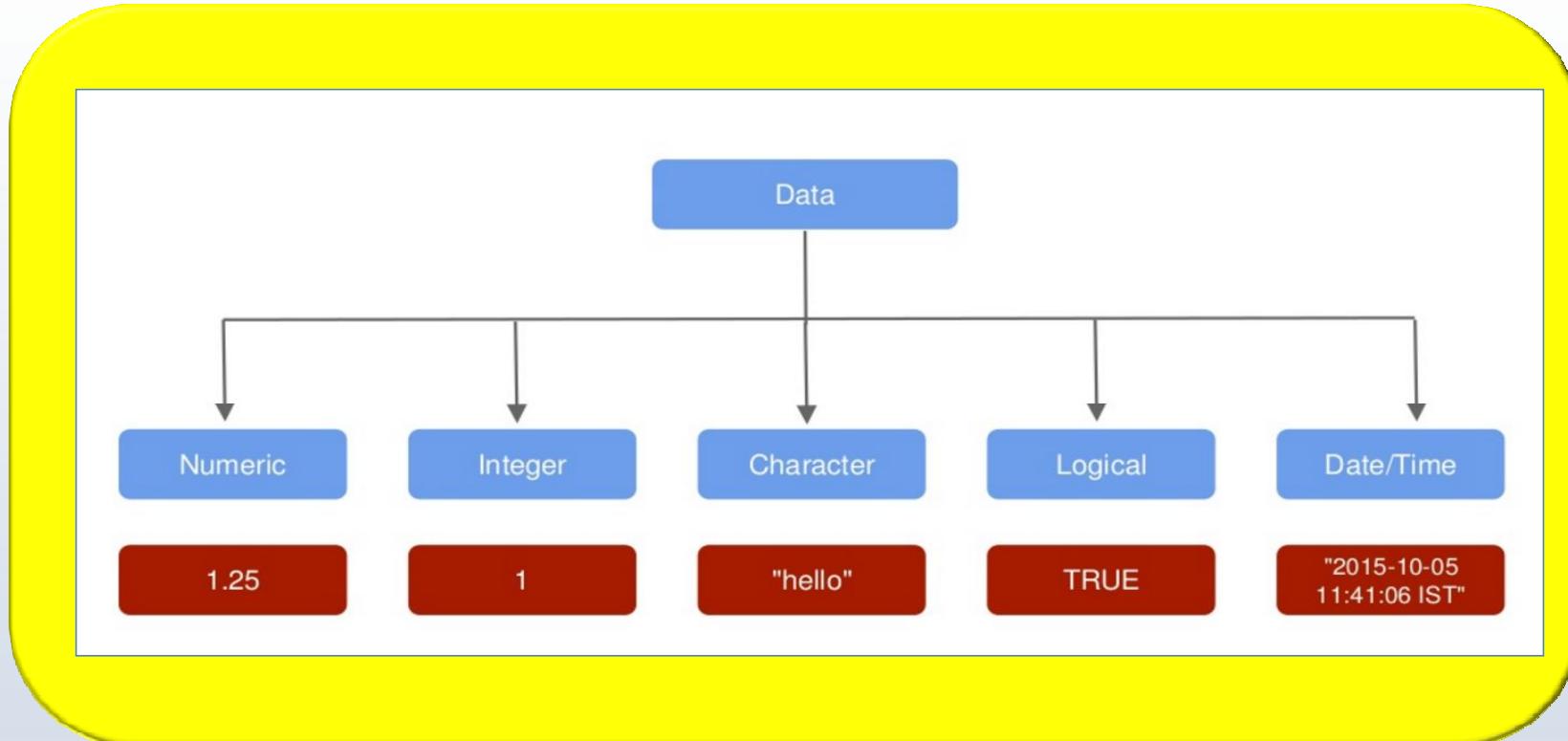
Comments may be added to a function or assignment using the # key

- Anything from the # to the end of the line is ignored
- May be used in conjunction with a Continuation prompt
- Example: a <- 3 #assign the value of 3 to a

Last expression

- The last value or expression returned by R is stored in .Last.value
- This may be used as a variable in a function or in an assignment
- NOTE: if the last expression was a function called, .Last.value will behave like a function , Example: a .Last.value

Data Types in R



What is scalar ?

- ▶ A scalar is the one element vector, for instance
 - ▶ $x <- 10$



Vector

- ▶ A vector is the most common and basic data structure in R
 - ▶ It's the simple and fundamental data structure in R, these are one dimensional array
- ```
a <- c(1,2,3,4)
```
- ▶ members in a vector is called components, it is represented as c() ?
    - ▶ To hold the numeric values c(7,8,11)
    - ▶ To hold the character values c('Ram','Ashok','Robert')
    - ▶ To hold the logical values c(TRUE,TRUE,FALSE)
  - ▶ using c() ,you can combine the vectors
  - ▶ b <- (5,4,a)

## Vector Arithmetic

```
a = c(1, 3, 5, 7)
```

```
b = c(1, 2, 4, 8)
```

- ▶ multiply it by 5, we would get a vector with each of its members multiplied by 5.s

```
> 5 * a
```

```
[1] 5 15 25 35
```

- ▶ Add a and b together, the sum would be a vector whose members are the sum of the corresponding members from a and b.

```
> a + b
```

```
[1] 2 5 9 15
```



## Vector Arithmetic

```
> x <- c(1,2,3,4,5,13,1,1)
> length(x)
[1] 8
> prod(x)
[1] 1560
> sum(x)
[1] 30
> min(x)
[1] 1
> max(x)
[1] 13
> median(x)
[1] 2.5
> sd(x)
[1] 4.026697
> cumsum(x)
[1] 1 3 6 10 15 28 29 30
`-
```



## How to check the data type of a vector

`class()` - what kind of object is it (high-level)?

`typeof()` - what is the object's data type (low-level)?

`length()` - how long is it? What about two dimensional objects?

`attributes()` - does it have any metadata?

`str()`



## How to name a vector

```
x <- c(10,13,14.5,12)
```

```
names(x) <- c('Quiz1', 'Quiz3', 'Quiz2', 'Quiz4')
```



## Integer

- ▶ Integers are treated as type numeric.

`a <- c(1,2,3,4)`

Here, a is a Numeric vector and are treated as double precision real numbers. It can also be converted as an integer using `as.integer(a)`

- ▶ To explicitly create integers , add a L at the end

`x1 <- c(1L, 2L, 3L)`

- ▶ To verify if this x1 vector is an integer type

`is.integer(x1)`

## Character

- ▶ Words, group of words are represented by type character
- ▶ All data of type character must be enclosed in single or double quotation marks
- ▶ we can use `is.character()` to check if a variable is of type character

## Logical

- ▶ Logical takes only two values. Either TRUE or FALSE
- ▶ we coerce other data types to type logical using as.logical()

## Date and Time

- ▶ Date and time are represented by the data type date.
- ▶ Use as.date() to coerce data type to type date.

## Coercion

- ▶ `as.character(5)`
- ▶ `as.integer(100.82)`
- ▶ `as.numeric("20.8")`
- ▶ `as.numeric("Hello World")`

## Summary

- ▶ Numeric, Integer, Character, Logical and Date are basic data types in R
- ▶ class/typeof functions returns the data type of an object
- ▶ is.data\_type tests whether an object is of the specified data type
  - ▶ is.numeric()
  - ▶ is.integer()
  - ▶ is.character() &
  - ▶ Is.logical()
- ▶ as.data\_type is used to coerce objects to the specified data type
  - ▶ as.numeric()
  - ▶ as.integer()
  - ▶ as.character()
  - ▶ as.logical()



Changes in the activity of the active and passive market in various unstructured positive trends in various market segments.



Distribution of market share among the major industry players.



Distribution of market share among the major industry players. Player A has 55%, Player B has 34%, and Player C has 11%.

Projected sales of main products in 2013

# Data Structures in R

## DS Part IV – Python Hands on

Sasken Training, Adyar , 9840014739 Chennai – 600 020



## Data Structures in R

- ▶ Scalar
- ▶ Vector
  - ▶ One dimensional Array
- ▶ Matrix
  - ▶ Two Dimensional array
- ▶ Array
  - ▶ Multidimensional Array
- ▶ List
  - ▶ Collection of objects
- ▶ Data Frame
  - ▶ Identical to spreadsheet or table data

**A vector is the most common and basic data structure in R**

**How to create a vector ?**

```
x <- c(1,2,3,4) # numeric vector
```

```
y <- c(TRUE, TRUE, FALSE, FALSE) # Logical vector
```

```
z <- c("Alec", "David", "Ravi", "Ram") # Character Vector
```

## Lists

- ▶ A list is a collection of elements , we also call this as recursive vectors, because a list can contain other lists or data frames.
- ▶ A list can have different types of data

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc", "dd", "ee")
b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
x = list(n, s, b, 3) # x contains copies of n, s, b
print(x)
```

### List Slicing

> x[2]

To access a specific member from a list

x[[2]][1]

To modify a content in a List

x[[2]][1]='AA1'

## Lists

```
> g <- list(Firstname=c("Ashok","Malik","Joe"), Age=c(26,32,38))
> g
$Firstname
[1] "Ashok" "Malik" "Joe"

$Age
[1] 26 32 38

> g$Firstname
[1] "Ashok" "Malik" "Joe"
> g$Age
[1] 26 32 38
> g$Firstname[3]
[1] "Joe"
>
```

```
> g[[2]][3]
[1] 38
> g
$Firstname
[1] "Ashok" "Malik" "Joe"

$Age
[1] 26 32 38
```

## Lists

```
> g <- append(g,"shiva",after=1)
> g
$Firstname
[1] "Ashok" "Malik" "Joe"

[[2]]
[1] "shiva"

$Age
[1] 26 32 38

> names(g) <- c("Firstname","LastName","Age")
> g
$Firstname
[1] "Ashok" "Malik" "Joe"

$LastName
[1] "shiva"

$Age
[1] 26 32 38
```

## Arrays

- ▶ Arrays is a structure that contains data of the same type, whether that's strings or characters , or integers.
- ▶ Arrays can be multi-dimensional as well, so that the data can be contained in multiple rows and columns

```
> test_data <- c("Akhil","Bhaskar",
+ "Chandra","Akshaya","Mihir","Dakshit","Harsha","Gajodhaar")
>
> my_array <- array(test_data,dim=c(3,4))
> my_array
 [,1] [,2] [,3] [,4]
[1,] "Akhil" "Akshaya" "Harsha" "Bhaskar"
[2,] "Bhaskar" "Mihir" "Gajodhaar" "Chandra"
[3,] "Chandra" "Dakshit" "Akhil" "Akshaya"
>
```

rows, cols

## Arrays

```
> c <- array(c(1:6),dim = c(2,3))
> c
 [,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
> |
```

```
arr1 <- c(1:12)
dim(arr1) <- c(2,3,2)
```

```
> d <- array(c(7:12),dim=c(2,3))
> d
 [,1] [,2] [,3]
[1,] 7 9 11
[2,] 8 10 12
> |
```

## Array Slicing

```
> a1[1:5]
[1] 1 2 3 4 5
> a1[,1]
[1] 1 2
> a1 <- array(c(1:6), dim=c(2,3))
> a1
 [,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
> a1[,1]
[1] 1 2
> a1[1]
[1] 1
> a1[1]=10
> a1
 [,1] [,2] [,3]
[1,] 10 3 5
[2,] 2 4 6
```

## Matrices

- ▶ It's a special kind of vector.
- ▶ A matrix is a vector with two additional attributes : The number of rows and the number of columns.

```
x <- matrix(c(1,2,3,4), nrow=2, ncol=2)
```

```
x
```

## Accessing members of Matrices

```
> m <- matrix(c('Ashok',8,10.2,4),nrow=2,ncol=2)
> print(m)
 [,1] [,2]
[1,] "Ashok" "10.2"
[2,] "8" "4"
> class(m)
[1] "matrix"
> m[1]
[1] "Ashok"
> m[1][1]
[1] "Ashok"
> m[2][1]
[1] "8"
```

Can I add an array and Matrix with different dimensions ?

```
> a <- matrix(c(3,4,5,8), nrow=2,ncol=2)
> a1 <- array(c(1:6), dim=c(2,3))
> a1
 [,1] [,2] [,3]
[1,] 1 3 5
[2,] 2 4 6
> a
 [,1] [,2]
[1,] 3 5
[2,] 4 8
> a1+a
Error in a1 + a : non-conformable arrays
```

## Arithmetic operations

Can I perform addition with the combination of array and Matrix ?

```
> a1 <- array(c(1:6), dim=c(2,2))
> a <- matrix(c(3,4,5,8), nrow=2, ncol=2)
> a1
 [,1] [,2]
[1,] 1 3
[2,] 2 4
> a
 [,1] [,2]
[1,] 3 5
[2,] 4 8
> a1+a
 [,1] [,2]
[1,] 4 8
[2,] 6 12
>
```

## Accessing members of Matrices

```
> m <- matrix(c('Ashok',8,10.2,4),nrow=2,ncol=2)
```

```
> print(m)
 [,1] [,2]
[1,] "Ashok" "10.2"
[2,] "8" "4"
```

```
> m[3][1]
[1] "10.2"
```

```
> is.numeric(m[3][1])
[1] FALSE
> is.character(m[3][1])
[1] TRUE
>
```

## Matrix Arithmetic

```
> a <- matrix(c(3,4,5,8), nrow=2,ncol=2)
> b <- matrix(c(2,1,6,3), nrow=2,ncol=2)
> a+b
 [,1] [,2]
[1,] 5 11
[2,] 5 11
> a
 [,1] [,2]
[1,] 3 5
[2,] 4 8
> b
 [,1] [,2]
[1,] 2 6
[2,] 1 3
> a+b
 [,1] [,2]
[1,] 5 11
[2,] 5 11
```

```
> a <- matrix(c(3,4,5,8), nrow=2,ncol=2)
> a
 [,1] [,2]
[1,] 3 5
[2,] 4 8
> b <- matrix(c(2.54,1,6,3), nrow=2,ncol=2)
> b
 [,1] [,2]
[1,] 2.54 6
[2,] 1.00 3
> a+b
 [,1] [,2]
[1,] 5.54 11
[2,] 5.00 11
```

## Array, Matrix dim & attributes()

```
> b <- matrix(c(2.54,1,6,3), nrow=2,ncol=2)
> b
 [,1] [,2]
[1,] 2.54 6
[2,] 1.00 3
> attributes(b)
$dim
[1] 2 2

> dim(b)
[1] 2 2
> dim(b) <- NULL
> b
[1] 2.54 1.00 6.00 3.00
> dim(b)
NULL
>
```

```
> a1 <- array(c(1:6), dim=c(2,2))
> a1
 [,1] [,2]
[1,] 1 3
[2,] 2 4
> attributes(a1)
$dim
[1] 2 2

> dim(a1)
[1] 2 2
> dim(a1) <- NULL
> a1
[1] 1 2 3 4
```

## Data Frames

- ▶ A data frame is collections of data arranged in rows and columns.
- ▶ Data frames are used to store spreadsheet or table-like data

## Data Frames

To delete a row use -ve values inside data frame and assign NULL to a column

```
> df <- df[-3]
> df
 name length
1 Mike 21
2 Lucy 32
3 John 35
4 <NA> 34
> df["length"] <- NULL
Error: object 'NULL' not found
> df["length"] <- NULL
> df
 name
1 Mike
2 Lucy
3 John
4 <NA>
> |
```

# Data Frames

## Data frame creation

```
> name <- c("Mike", "Lucy", "John")
> name
[1] "Mike" "Lucy" "John"
> df <- data.frame(name)
> df
 name
1 Mike
2 Lucy
3 John
```

## Adding a column to an existing data frame

```
> df['length'] <- c(21,32,35)
> df
 name length
1 Mike 21
2 Lucy 32
3 John 35
> df['age'] <- c(21,32,35)
> df
 name length age
1 Mike 21 21
2 Lucy 32 32
3 John 35 35
`
```

## Data Frames

To delete a row use -ve values inside data frame and assign NULL to a column

```
> df <- df[-3]
> df
 name length
1 Mike 21
2 Lucy 32
3 John 35
4 <NA> 34
> df["length"] <- null
Error: object 'null' not found
> df["length"] <- NULL
> df
 name
1 Mike
2 Lucy
3 John
4 <NA>
> |
```

# Data Frames

## Creating a data frame

```
> employee <- c('John Doe', 'Peter Gynn', 'Jolie Hope')
> salary <- c(21000, 23400, 26800)
> startdate <- as.Date(c('2010-11-1', '2008-3-25', '2007-3-14'))
>
> df1 <- data.frame(employee, salary, startdate)
> df1
 employee salary startdate
1 John Doe 21000 2010-11-01
2 Peter Gynn 23400 2008-03-25
3 Jolie Hope 26800 2007-03-14
> is.data.frame(df1)
[1] TRUE
>
```

```
> name <- c("Mike", "Lucy", "John")
> age <- c(20, 25, 30)
> student <- c(TRUE, FALSE, TRUE)
> df = data.frame(name, age, student)
>
> df
 name age student
1 Mike 20 TRUE
2 Lucy 25 FALSE
3 John 30 TRUE
>
```

## Data Frames

- ▶ Extracting the dimensions of a table
  - ▶ `dim(df)`
- ▶ To find out the number of rows and columns
  - ▶ `nrow(df)`
  - ▶ `ncol(df)`
- ▶ To access the variable/column names
  - ▶ `names(df)`
- ▶ The unique name of all 1000 persons are stored in character vector
  - ▶ `length(df[, "col3"])`
- ▶ The unique name of all 1000 persons are stored in character vector
  - ▶ `class(df[, "col3"])`