

Welcome to the first lesson of the "Introduction to Data Analysis using R" course on Big Data University.

Each lesson of this course includes a video and a set of student exercises.

At the end of the course there is a final evaluation available to validate your learning.

Agenda

- What is R?
- Why use R?
- Installing R



2

© 2013 BigDataUniversity.com

In this lesson we will initially learn about the features and uses of R.

We will then learn how to install R on various operating platforms.

We will then get started using R within the Console using R in an interactive and batch mode.

Finally, in this lesson, we will discuss how the R can be extended using packages.

What is R?



is a free software environment for data analysis and graphics.

- Programming language
- Data visualization tool
- And more...

R is a software environment that is excellent for data analysis and graphics.

It was initially created in 1993 by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand. They created R as a language to help teach introductory statistics to their students. They based R on the S language that was developed earlier at Bell Labs in the 1970s.

After some time they made R available as an open source GNU project. A very active R community now exists around the world.

R is considered a Domain Specific Language as it was designed primarily for data analysis.

R programs are typically created using functions and the programs are executed by an R interpreter.

R is not just a programming language as it has native support for creating high quality data visualizations.

As we progress through the course, we will learn how to explore datasets to extract insight.

When to Use R?



Need to analyze **structured** or **unstructured** datasets to **gain insight**.

1. Data Exploration and Visualization (Descriptive)
2. Predictive Analytics

R is used across many industries such as healthcare, retail, and financial services.

R can be used to analyze both structured and unstructured datasets.

In this course we will focus on using R to analyze data from files and not databases.

R can help you explore a new dataset and perform descriptive analysis.

R is also excellent at building predictive models.

Why Learn R?

Data Analyst / Data Scientist

Perform comprehensive data analytics than can not be achieved using spreadsheet-based tools alone.

Software Developer

Fulfill the user demands for improved data analysis features in new or existing applications

There are many reasons why learning R is beneficial.

As a Data Analyst or Data Scientist – R can be used to dig deeper into your data than is possible using spreadsheet-based tools alone

As a software developer – R can enable data analytics computations and graphics into new or existing applications with minimal effort.

With the explosion of Big Data, there are many new scenarios where using R is an excellent choice to help meet user demands.

R – as an interactive "data analysis tool"

- classical statistical tests
- linear and nonlinear modeling
- time-series analysis
- classification
- clustering
- more..

As a data analyst, R can be used to perform classical statistical tests and predictive models.

R also has native support for handling time-series datasets.

Classification and clustering models can be used to better detect patterns.

Throughout this course you will learn how to use some of the most commonly used R functions, but there are many capabilities that we will not have the time to cover in an introductory course.

R – as a "programming language"

- Expressions
- Built-in Functions
- Data structures
 - Vectors, Matrices, Lists, Data Frames
- External libraries – packages
- User-defined Functions
- User-defined Classes

As a developer R is a powerful functional programming language.

In this course you will learn how to create and test simple R programs. Note that R programs are usually referred to as scripts.

Since R scripts are interpreted it encourages an interactive approach to development.

R scripts are typically written using expressions and built-in functions.

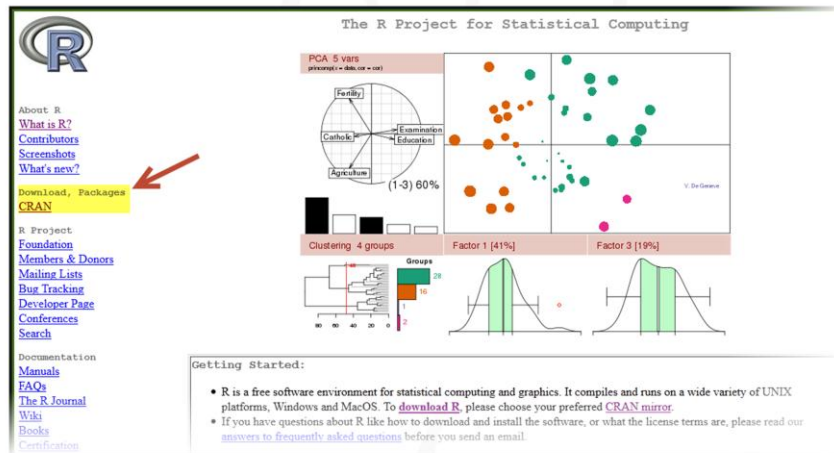
R provides native support for many useful types of data structures. Many of these data structures will be explored in other lessons.

External libraries can be used to extend the capabilities of R.

As your R skills improve you will likely start to define your own functions and possibly new Classes to meet the demands of your users.

Getting Started - Installing R

Installation (<http://www.r-project.org/>)



Installing R is quite simple.

Simply navigate to the R Project page and click on the Comprehensive R Archive Network or CRAN link.

CRAN is a set of servers around the world that store identical, up-to-date, versions of code and documentation for R.

There are binary installers available for Windows, Linux, and Mac OS platforms. It is possible to build R from source, but it is best to avoid this step if possible so you can get started using R more quickly.

For the purposes of this course you can select any of these platforms to learn R.

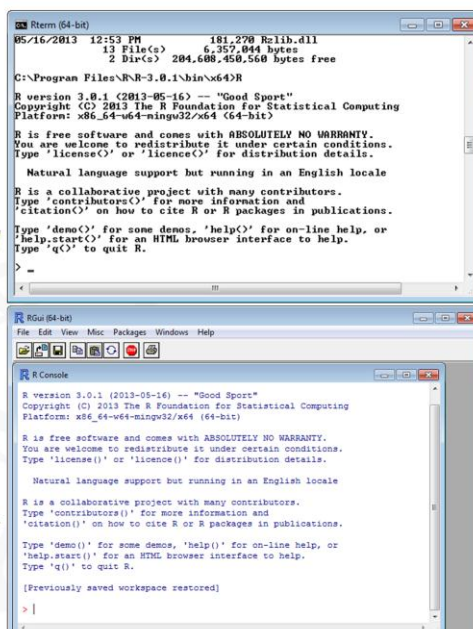
Installing R on Windows

Windows

- CRAN mirror site
- Download MSI (32 / 64 bit)
- Install

Using R

- Command Line:
C:\Program Files\R\R-3.0.1\bin
- GUI – Start Menu



Installing R on Windows involves downloading the MSI file and executing it.

There are 32-bit and 64-bit installation options available. We will use the 64-bit version for our coursework as it has higher limits on the amount of memory that can be used.

Once the Windows installation has finished you can get started with R by launching the **R command line environment** or the **RGui** tool.

RGui provides some useful productivity features beyond the R command line environment for R users.

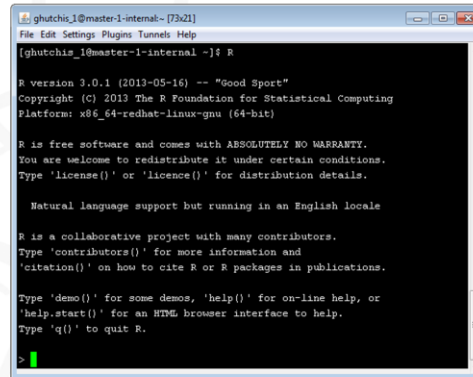
Installing R on Linux

Linux

- CRAN mirror site
- Download RPM
`rpm -ivh <.rpm>`
- OR
`yum install R`

■ Using R

- Command Line:
`/usr/bin/R`
- GUI
`/usr/bin/R -g Tk &`
- RCmdr, RStudio (optional – next slide)



```
ghutchis_1@master-1-internal: [7321]
[ghutchis_1@master-1-internal ~]$ R

R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
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.

>
```

Installing R on Linux involves either: downloading the appropriate RPM file from the CRAN website or use of a Linux package manager such as YUM as shown.

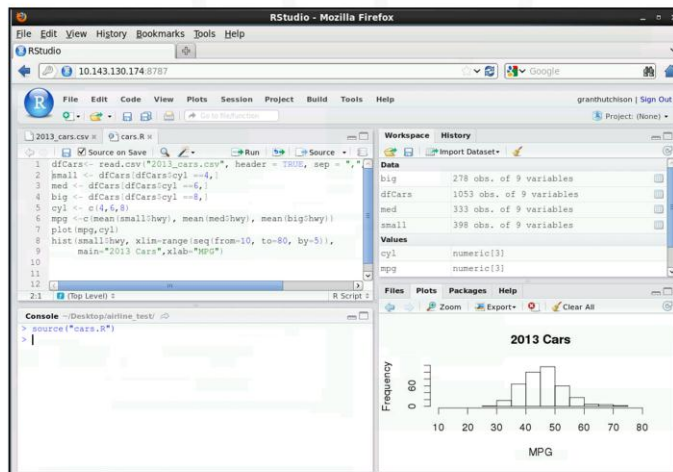
Note that you must be logged in as a root user or have sudo privileges on your Linux system to complete the installation.

Once installed on the system any user can use R.

By default, there is an R command line and GUI provided, but many R users prefer to use a more comprehensive Integrated Development Environment (IDE) such as RCmdr or Rstudio.

GUI Alternatives - RStudio

Rstudio is a free IDE that makes it easier to create and test R scripts. (rstudio.org)



11

© 2013 BigDataUniversity.com

RStudio is an excellent alternative to the RGui tool provided with R. RStudio is available on Linux, Mac OS X, and Windows.

In this configuration we are using RStudio on a Linux server from within a browser.

This environment is ideal for occasional R users as they would not need to install R on their own computer to use it.

Let's examine some of the tiled windows shown here:

- In the top left corner we are able to view the **2013_cars.csv** data file and an R source file called **cars.R**.

- In the bottom left corner we have the R Console.

- In the top right corner we have access to the objects in the current R workspace and a history of recently used R commands.

- In the bottom right corner we have a histogram plot of data along with access to the R help utility.

It is worth the time and effort to install an IDE such as RStudio as you learn R.

Extending R - Packages

Task Views

– categories



CRAN Task Views	
Bayesian	Bayesian Inference
ChemPhys	Chemometrics and Computational Physics
ClinicalTrials	Clinical Trial Design, Monitoring, and Analysis
Cluster	Cluster Analysis & Finite Mixture Models
DifferentialEquations	Differential Equations
Distributions	Probability Distributions
Econometrics	Computational Econometrics
Environmetrics	Analysis of Ecological and Environmental Data
ExperimentalDesign	Design of Experiments (DoE) & Analysis of Experimental Data
Finance	Empirical Finance
Genetics	Statistical Genetics
Graphics	Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization
HighPerformanceComputing	High-Performance and Parallel Computing with R
MachineLearning	Machine Learning & Statistical Learning
MedicalImaging	Medical Image Analysis
MetaAnalysis	Meta-Analysis
Multivariate	Multivariate Statistics
NaturalLanguageProcessing	Natural Language Processing
OfficialStatistics	Official Statistics & Survey Methodology
Optimization	Optimization and Mathematical Programming
Pharmacokinetics	Analysis of Pharmacokinetic Data
Phylogenetics	Phylogenetics, Especially Comparative Methods
Psychometrics	Psychometric Models and Methods
ReproducibleResearch	Reproducible Research
Robust	Robust Statistical Methods
SocialSciences	Statistics for the Social Sciences
Spatial	Analysis of Spatial Data
SpatioTemporal	Handling and Analyzing Spatio-Temporal Data
Survival	Survival Analysis

Previously we stated that R can be extended using packages.

There are over 4000 different packages available in CRAN and more being added frequently.

The packages published in CRAN are categorized based on their functionality into Task Views.

During this course we will primarily use the built-in or standard set of packages, but you may wish to explore some of the additional packages along the way.

R - Packages

Working with addition R packages

- Installing new packages from CRAN

```
>install.packages("RJDBC")
```
- Installing new packages from a downloaded file

```
>install.packages("file.tar.gz")
```
- List the currently loaded packages

```
>library()
```
- Loading a package into a session

```
>library("stringr")  
>require("stringr")
```

The base R environment provides a significant set of functions for data analysis, but there are many excellent packages available from the R Community.

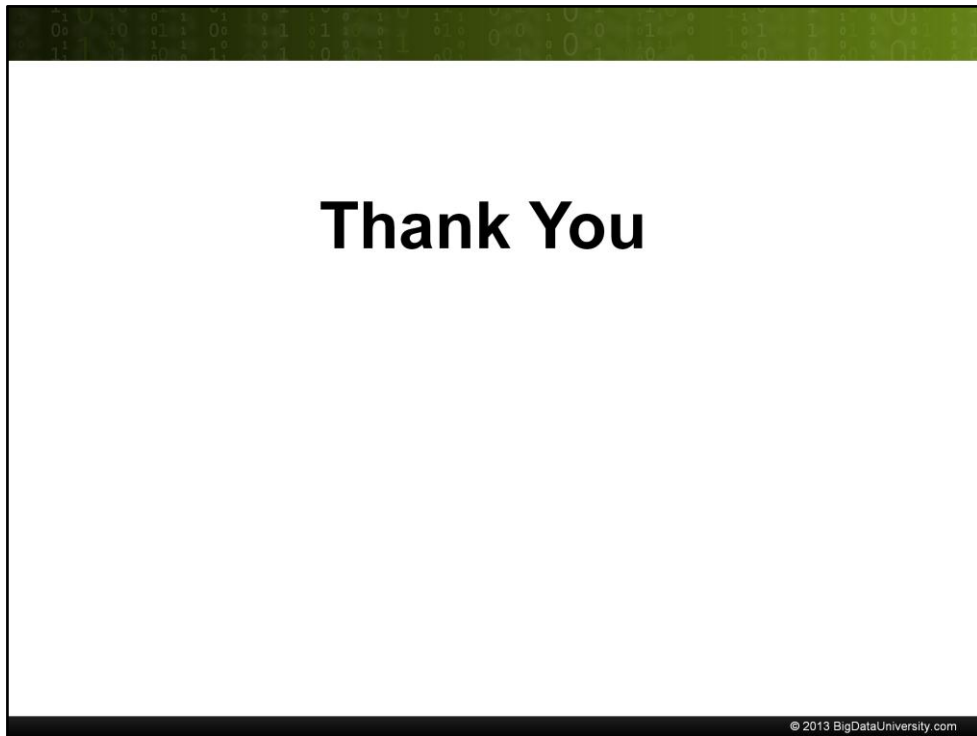
The new packages can be installed add using the `install.packages()` function.

CRAN will be searched for the package or you may have given a new package that is not available in CRAN.

Simply use the same function and direct it to the compressed archive file for the new package.

Here we see that the RJDBC package is being installed to enable connectivity to database servers such as Informix or DB2 through a JDBC driver.

If your develop an R script that uses functions that are not part of the R base your script should contain the `library()` or `require()` functions within the first few lines in the script so the package is loaded into memory during runtime.



Thank you for completing this lesson. Spend some time to get R installed on your computer and then continue with the course.