

Introduction to R Programming

Description

R is an open-source free programming language for statistical computing, data analysis, and graphics. R is used by a growing number of managers and data analysts inside corporations and academia. R has also found followers among statisticians, engineers and scientists without computer programming skills who find it easy to use. Its popularity is due to the increasing use of data mining for various goals such as set ad prices, find new drugs more quickly or fine-tune financial models. R has a wide variety of packages for data mining. It's a language that many non-programmers can easily work with, naturally extending a skill set that is common to high-end Excel users. It's the perfect tool for when the analyst has a statistical, numerical, or probabilities-based problem based on real data and they've pushed Excel past its limits.

Delegates will learn

- R Programming JumpStart is a hands-on course covers the manipulation of objects in R including reading data, accessing R packages, writing R functions, and making informative graphs. It includes analyzing data using common statistical models. The course teaches how to use the R software (<http://www.r-project.org>) both on a command line and in a graphical user interface (GUI).

Audience

This is an introductory level programming course.

Attendees for this course should have prior practical hands-on experience with another programming language. Prior exposure to working with statistics and probability, as

well as hands-on working knowledge of Excel would also be helpful but is not required. We will collaborate with you to design the best solution to ensure your needs are met, whether we customize the material, or devise a different educational path to help your team best prepare for this training.

Outline

R and statistics

- Using R interactively
- An introductory session
- Getting help with functions and features
- R commands, case sensitivity, etc.
- Recall and correction of previous commands
- Executing commands from or diverting output to a file
- Data permanency and removing objects

Simple manipulations; numbers and vectors

- Vectors and assignment
- Vector arithmetic
- Generating regular sequences
- Logical vectors
- Missing values
- Character vectors
- Index vectors; selecting and modifying subsets of a data set
- Other types of objects

Objects, their modes and attributes

- Intrinsic attributes: mode and length
- Changing the length of an object
- Getting and setting attributes

- The class of an object

Ordered and unordered factors

- A specific example
- The function `tapply()` and ragged arrays
- Ordered factors

Arrays and matrices

- Arrays
- Array indexing. Subsections of an array
- Index matrices
- The `array()` function
- Mixed vector and array arithmetic. The recycling rule
- The outer product of two arrays
- Generalized transpose of an array
- Matrix facilities
- Matrix multiplication
- Linear equations and inversion
- Eigenvalues and eigenvectors
- Singular value decomposition and determinants
- Least squares fitting and the QR decomposition
- Forming partitioned matrices, `cbind()` and `rbind()`
- The concatenation function, `()`, with arrays
- Frequency tables from factors

Lists and data frames

- Lists
- Constructing and modifying lists
- Concatenating lists
- Data frames
- Making data frames
- `attach()` and `detach()`
- Working with data frames
- Attaching arbitrary lists
- Managing the search path

Reading data from files

- The `read.table()` function
- The `scan()` function
- Accessing builtin datasets
- Loading data from other R packages
- Editing data

Probability distributions

- R as a set of statistical tables
- Examining the distribution of a set of data
- One- and two-sample tests

Grouping, loops and conditional execution

- Grouped expressions
- Control statements
- Conditional execution: if statements
- Repetitive execution: for loops, repeat and while
- Session: Writing your own functions
- Simple examples
- Defining new binary operators
- Named arguments and defaults
- The `'...'` argument
- Assignments within functions
- More advanced examples
- Efficiency factors in block designs
- Dropping all names in a printed array
- Recursive numerical integration
- Scope
- Customizing the environment
- Classes, generic functions and object orientation

Statistical models in R

- Defining statistical models; formulae
- Contrasts
- Linear models
- Generic functions for extracting model information
- Analysis of variance and model comparison
- ANOVA tables

- Updating fitted models
- Generalized linear models
- Families
- The `glm()` function
- Nonlinear least squares and maximum likelihood models
- Least squares
- Maximum likelihood
- Some non-standard models

Graphical procedures

- High-level plotting commands
- The `plot()` function
- Displaying multivariate data
- Display graphics
- Arguments to high-level plotting functions
- Low-level plotting commands
- Mathematical annotation
- Hershey vector fonts
- Interacting with graphics
- Using graphics parameters
- Permanent changes: The `par()` function
- Temporary changes: Arguments to graphics functions
- Graphics parameters list
- Graphical elements
- Axes and tick marks
- Figure margins
- Multiple figure environment
- Device drivers
- PostScript diagrams for typeset documents
- Multiple graphics devices
- Dynamic graphics

Packages

- Standard packages
- Contributed packages and CRAN
- Namespaces