

# Introduction to R

Venkat Reddy

# Contents

- What is R
- R basics
- Working with R
- R packages
- R Functions
- Importing & Exporting data
- Creating calculated fields
- R Help & Tutorials

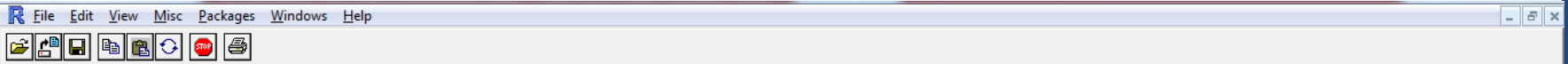
# R

- Programming “environment”
- Runs on a variety of platforms including Windows, Unix and MacOS.
- Provides an unparalleled platform for programming new statistical methods in an easy and straightforward manner.
- Object-oriented
- Open source
- Excellent graphics capabilities
- Supported by a large user network

# Downloading

- Google it using R or CRAN (Comprehensive R Archive Network)
- <http://www.r-project.org>

# R



```
R version 2.15.1 (2012-06-22) -- "Roasted Marshmallows"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-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
```

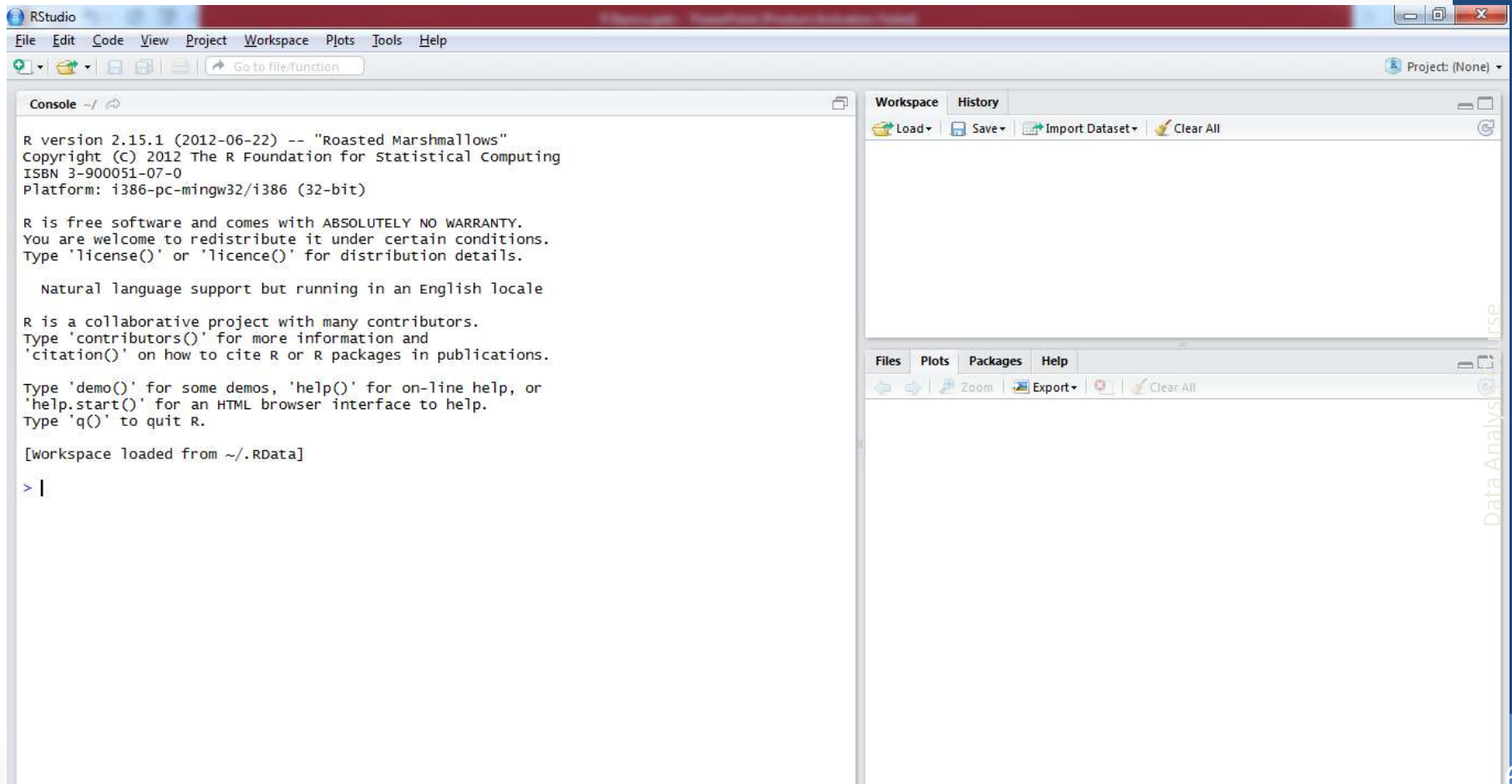
```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
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.
```

```
[Previously saved workspace restored]
```

```
> |
```

# R Studio



# R-Demo

- `2+2`
- `log(10)`
- `help(log)`
- `summary(airquality)`
- `demo(graphics)` # pretty pictures...

# R-Basics :Naming convention

- must start with a letter (A-Z or a-z)
- can contain letters, digits (0-9), and/or periods “.”
- **R is a case sensitive language.**
  - `mydata` different from `MyData`



# R-Basics :Assignment

- “<-” used to indicate assignment
  - `x<-c(1,2,3,4,5,6,7)`
  - `x<-c(1:7)`
  - `x<-1:4`
- Assignment to an object is denoted by “<-” or “->” or “=”.
- If you see a notation “= =”, you’ll looking at a comparison operator.
  - Many other notations can be found from the documentation for the Base package or R.

# Workspace

- during an R session, all objects are stored in a temporary, working memory
- Commands are entered interactively at the **R** user prompt. **Up** and **down arrow keys** scroll through your command history.
- list objects `ls ( )`
- remove objects `rm ( )`
- `data ( )`

# Demo: Working with R

- `x <- round(rnorm(10,mean=20,sd=5)) # simulate data`
- `x`
- `mean(x)`
- `m <- mean(x)`
- `m`
- `x - m # notice recycling`
- `(x - m)^2`
- `sum((x - m)^2)`
- `data()`
- `UKgas`

# R Packages

- R consists of a core and packages. Packages contain functions that are not available in the core.
- Collections of R functions, data, and compiled code
- Well-defined format that ensures easy installation, a basic standard of documentation, and enhances portability and reliability
- When you download R, already a number (around 30) of packages are downloaded as well.
- You can use the function `search()` to see a list of packages that are currently attached to the system, this list is also called the search path.
- `search()`

# R packages

- Select the 'Packages' menu and select 'Install package...', a list of available packages on your system will be displayed.
- Select one and click 'OK', the package is now attached to your current R session. Via the library function
- The library can also be used to list all the available libraries on your system with a short description. Run the function without any arguments

# Demo: R packages

- Install cluster package
- Install plyr package (for string operations)

# Importing Data

- Reading CSV file
  - `X <- read.csv("file.csv")`
- `read.table()`
  - reads in data from an external file
  - `d <- read.table("myfile", header=TRUE)`
- R has ODBC for connecting to other programs
- **R** gets confused if you use a path in your code like *c:\mydocuments\myfile.txt*
- This is because R sees "\" as an escape character. Instead, use *c:\\my documents\\myfile.txt*  
or  
*c:/mydocuments/myfile.txt*

# Demo: Importing data

- Reading CSV file
  - `petrol<-read.csv("C:\\Users\\VENKAT\\Google Drive\\Training\\R\\Data\\Petrol_Consumption.csv")`
  - `sales_data<-read.table("C:\\Users\\VENKAT\\Google Drive\\Training\\R\\Data\\sales.txt")`
  - `sales_data<-read.table("C:\\Users\\VENKAT\\Google Drive\\Training\\R\\Data\\sales.txt",header=TRUE)`



# Exporting data

- To A Tab Delimited Text File
  - `write.table(mydata, "c:/mydata.txt", sep="\t")`
- To an Excel Spreadsheet
  - `library(xlsReadWrite)`
  - `write.xls(mydata, "c:/mydata.xls")`
- To SAS
  - `library(foreign)`
  - `write.foreign(mydata, "c:/mydata.txt", "c:/mydata.sas", package="SAS")`

# Demo: Exporting data

- `write.table(sales_data, "C:\\Users\\VENKAT\\Google Drive\\Training\\R\\Data\\sales_export.txt", sep="\t")`
- `write.table(sales_data, "C:\\Users\\VENKAT\\Google Drive\\Training\\R\\Data\\sales_export.csv", sep=",")`

# R- Functions

## Numeric Functions

Function	Description
<b>abs</b> ( $x$ )	absolute value
<b>sqrt</b> ( $x$ )	square root
<b>ceiling</b> ( $x$ )	ceiling(3.475) is 4
<b>floor</b> ( $x$ )	floor(3.475) is 3
<b>trunc</b> ( $x$ )	trunc(5.99) is 5
<b>round</b> ( $x$ , digits= $n$ )	round(3.475, digits=2) is 3.48
<b>signif</b> ( $x$ , digits= $n$ )	signif(3.475, digits=2) is 3.5
<b>cos</b> ( $x$ ), <b>sin</b> ( $x$ ), <b>tan</b> ( $x$ )	also <b>acos</b> ( $x$ ), <b>cosh</b> ( $x$ ), <b>acosh</b> ( $x$ ), etc.
<b>log</b> ( $x$ )	natural logarithm
<b>log10</b> ( $x$ )	common logarithm
<b>exp</b> ( $x$ )	$e^x$

# Demo: Numeric Functions

- `y<-abs(-20)`
- `x<-Sum(y+5)`
- `Z<-Log(x)`
- `round(x,1)`

# Character Functions

## Function

## Description

**substr**(*x*, **start**=*n1*, **stop**=*n2*)

Extract or replace substrings in a character vector.

`x <- "abcdef"`

`substr(x, 2, 4)` is "bcd"

`substr(x, 2, 4) <- "22222"` is "a222ef"

**grep**(*pattern*, *x*,  
**ignore.case**=FALSE,  
**fixed**=FALSE)

Search for *pattern* in *x*. If **fixed**=FALSE then *pattern* is a [regular expression](#). If **fixed**=TRUE then *pattern* is a text string. Returns matching indices.

`grep("A", c("b", "A", "c"), fixed=TRUE)` returns 2

**sub**(*pattern*, *replacement*, *x*,  
**ignore.case**=FALSE,  
**fixed**=FALSE)

Find *pattern* in *x* and replace with *replacement* text. If **fixed**=FALSE then *pattern* is a regular expression.

If **fixed** = T then *pattern* is a text string.

`sub("\\s", ".", "Hello There")` returns "Hello.There"

**strsplit**(*x*, *split*)

Split the elements of character vector *x* at *split*.

`strsplit("abc", "")` returns 3 element vector "a", "b", "c"

**paste**(..., **sep**="")

Concatenate strings after using *sep* string to separate them.

`paste("x", 1:3, sep="")` returns c("x1", "x2", "x3")

`paste("x", 1:3, sep="M")` returns c("xM1", "xM2", "xM3")

`paste("Today is", date())`

**toupper**(*x*)

Uppercase

# Demo :Character Functions

- `cust_id<-"Cust1233416"`
- `id<-substr(cust_id, 5,10)`
- `Up=toupper(cust_id)`

# Calculated Fields in R

- Use the assignment operator `<-` to create new variables. A wide array of operators and functions are available here.
  - `mydata$sum <- mydata$x1 + mydata$x2`
  - `mydata$mean <- (mydata$x1 + mydata$x2)/2`
  - `attach(mydata)`
  - `mydata$sum <- x1 + x2`
  - `mydata$mean <- (x1 + x2)/2`
  - `detach(mydata)`

# Demo Calculated Fields in R

- `sales_data$reduce<-(sales_data$Sales)*0.2`
- `View(sales_data)`
- `sales_data$new_sales<-sales_data$Sales- sales_data$reduce`
- `attach(petrol)`
- `ratio=Income/consum_mill_gallons`
- `View(petrol)`
- `petrol$ratio=Income/Consum_mill_gallons`
- `View(petrol)`



# R-Help

- If you encounter a new command during the exercises, and you'd like to know what it does, please consult the documentation. All R commands are listed nowhere, and the only way to get to know new commands is to read the documentation files, so we'd like you to practise this yourself.

- **Tutorials**

Each of the following tutorials are in PDF format.

- P. Kuhnert & B. Venables, [An Introduction to R: Software for Statistical Modeling & Computing](#)
- J.H. Maindonald, [Using R for Data Analysis and Graphics](#)
- B. Muenchen, [R for SAS and SPSS Users](#)
- W.J. Owen, [The R Guide](#)
- D. Rossiter, [Introduction to the R Project for Statistical Computing for Use at the ITC](#)
- W.N. Venables & D. M. Smith, [An Introduction to R](#)

# R-Tutorials

- Paul Geissler's [excellent R tutorial](#)
- [Dave Robert's Excellent Labs](#) on Ecological Analysis
- [Excellent Tutorials by David Rossitier](#)
- [Excellent tutorial on nearly every aspect of R](#) (c/o Rob Kabacoff) **MOST of these notes follow this web page format**
- [Introduction to R by Vincent Zoonekynd](#)
- [R Cookbook](#)
- [Data Manipulation Reference](#)

# Thank you