Introduction into R A Short Overview

Thomas Girke

December 8, 2012

#### Introduction

Look and Feel of the R Environment R Library Depositories Installation Getting Around Basic Syntax Data Types and Subsetting Basic Calculations Reading and Writing External Data Some Great R Functions Graphics Utilities

**Online Tutorial** 

### Outline

#### Introduction

Look and Feel of the R Environment R Library Depositories Installation Getting Around Basic Syntax Data Types and Subsetting Basic Calculations Reading and Writing External Data Some Great R Functions Graphics Utilities

#### **Online Tutorial**

# What You'll Get?



#### Command-line R: Linux/OS X

#### R Gui: Windows

Introduction into R

## RStudio: Alternative Working Environment for R

New integrated development environment (IDE) for R that works well for beginners and developers.

Rhule		
ie Edit View Workspace Plots Help		
C damandhiona I' x damandi x	Workspace History	-
E Therese an Save Q. Z	🞯 Load+ 🕞 Save+ 🖙 Import Dataset+ 🕑 Clear All	
(ibrary(poplet2)	Data	
(iew(diamonds))	diamonds 53940 obs. of 10 variables	
ummary(diamonds)	Values	
ummary(diamonds Sprice)	avestage 0.7979 clarity character[6]	
(larity = levels(diamondsSclarity)		
plot(price, carat, data = diamonds)		
<pre>glot(price, carat, data = diamonds, color=clarity, sisb = "Price", ylab = "Carat", main = "Diamond Pricing") = : opts(glot.title = theme_text(size = 22))</pre>		
	Files Plots Packages Help	-
	😔 😳 🎢 Zoom 🗯 Export 🔛 Print 🖉 Clear All	
	Diamond Pricing	
Lift On: -1.72         Lift On: -1.72         Lift On: -1.70         Lift On: -1.70         Lift On: -1.70           Lift On: -1.72         Lift On: -1.70         Lift On: -1.70         Lift On: -1.70         Lift On: -1.70           Def On: -1.81         Def On: -1.81         Def On: -1.80         Pero -1.80<	and the second s	elanty • 11 • 52 • 91 • 123 • 123
$\label{eq:constraints} \begin{split} & (1+1)^{1/2} \left( 1+1)^{1/2} \left( 1+1)^{1/2} \left( 1+1)^{1/2} \left( 1+1)^{1/2} \left( 1+1)^{1/2} \left( 1+1\right)^{1/2} $	The second secon	- <b>n</b> - 2000

# Why Using R

- Complete statistical package and programming language
- Efficient functions and data structures for data analysis
- Powerful graphics
- Access to fast growing number of analysis packages
- Most widely used language in bioinformatics
- Is standard for data mining and biostatistical analysis
- Technical advantages: free, open-source, available for all OSs

#### Books & Documentation

- simpleR Using R for Introductory Statistics (Gentleman et al., 2005)
- Bioinformatics and Computational Biology Solutions Using R and Bioconductor (John Verzani, 2004)
- UCR Manual (Thomas Girke)

### Package Depositories

- CRAN (>3000 packages) general data analysis
- BioConductor (>500 packages) bioscience data analysis
- Omegahat (>30 packages) programming interfaces

#### Installation

Install R binary for your operating system from: http://cran.at.r-project.org

Installation of CRAN Packages

- > install.packages(c("pkg1", "pkg2"))
- > install.packages("pkg.zip", repos=NULL)

Installation of BioConductor Packages

- > source("http://www.bioconductor.org/biocLite.R")
- > biocLite()
- > biocLite(c("pkg1", "pkg2"))

# Startup/Closing Behavior

#### Starting R

The R GUI versions under Windows and Mac OS X can be opened by double-clicking their icons. Alternatively, one can start it by typing 'R' in a terminal (default under Linux).

#### Startup/Closing Behavior

The R environment is controlled by hidden files in the startup directory: .RData, .Rhistory and .Rprofile (optional).

```
## Closing R
> q()
Save workspace image? [y/n/c]:
```

#### Note

When responding with 'y', then the entire R workspace will be written to the .RData file which can become very large. Often it is sufficient to just save an analysis protocol in an R source file. This way one can quickly regenerate all data sets and objects.

### Getting Around

```
## Create an object with the assignment operator '<-' (or '=')
> object <- ...
## List objects in current R session
> ls()
## Return content of current working directory
> dir()
## Return path of current working directory
> getwd()
## Change current working directory
> setwd("/home/user")
```

# Basic R Syntax

```
## General R command syntax
> object <- function(arguments)</pre>
> object <- object[arguments]</pre>
## Execute an R script
> source("my_script.R")
## Execute an R script from command-line
$ R CMD BATCH my_script.R
$ R --slave < my_script.R</pre>
## Finding help
> ?function
## Load a library
> library("my_library")
## Summary of all functions within a library
> library(help="my_library")
## Load library manual (PDF file)
> vignette()
```

## Data Types

```
## Numeric data: 1, 2, 3
> x <- c(1, 2, 3); x; is.numeric(x); as.character(x)</pre>
## Character data: "a", "b", "c"
> x <- c("1", "2", "3"); x; is.character(x); as.numeric(x)</pre>
## Complex data: 1, b, 3
> c(1, "b", 3)
## Logical data: TRUE, FALSE, TRUE
> x < -1:10 < 5; x
> !x
## Return indices for the 'TRUEs' in logical vector
> which(x)
```

### Data Objects

```
## Vectors (1D)
> myVec <- 1:10; names(myVec) <- letters[1:10]</pre>
> myVec[1:5]; myVec[c(2,4,6,8)]; myVec[c("b", "d", "f")]
## Factors (1D): vectors with grouping information
> factor(c("dog", "cat", "mouse", "dog", "dog", "cat"))
## Matrices (2D), Data Frames (2D) and Arrays (>2D)
> myMA <- matrix(1:30, 3, 10, byrow = T)
> myDF <- data.frame(Col1=1:10, Col2=10:1)</pre>
> myDF[1:4, ]; myDF[ ,c("Col2", "Col1", "Col1")]
## Lists: containers for any object type
> myL <- list(name="Fred", wife="Mary", no.children=3,</pre>
child.ages=c(4,7,9))
> myL[[4]][1:2]
## Functions: piece of code
> myfct <- function(arg1, arg2, ...) { function_body }</pre>
```

### General Subsetting Rules

```
## Subsetting by indices
> myVec <- 1:26; names(myVec) <- LETTERS
> myVec[1:4]
## Subsetting by same length logical vectors
> myLog <- myVec > 10
> myVec[myLog]
## Subsetting by field names
> myVec[c("B", "K", "M")]
## Special case
> iris$Species
```

# Basic Operators and Calculations

- Comparison operators: ==, ! =, <, >, <=, >= ## Example:
  - > 1==1
- Logical operators: AND: &, OR: |, NOT: !
   ## Example:
  - > x <- 1:10; y <- 10:1
  - > x > y & x > 5
- Calculations: ## Example: > x + y; sum(x); mean(x), sd(x); sqrt(x) > apply(iris[,1:3], 1, mean)

#### Reading and Writing External Data

```
## Import Data into R
> read.delim("myData.xls", sep="\t")
```

## Export Data from R to File
> write.table(myframe, file="myfile.xls", sep="\t", quote=F)

### Some Great R Functions

## The unique() function to make vector entries unique
> unique(iris\$Sepal.Length); length(unique(iris\$Sepal.Length))

## The table() function counts the occurrences of entries
> table(iris\$Species)

## The aggregate() function computes statistics of data
aggregates

> aggregate(iris[,1:4], by=list(iris\$Species), FUN=mean, na.rm=T)

## The %in% function returns the intersect between two vectors
> month.name[month.name %in% c("May", "July")]

## The merge() function joins data frames based on a common key
column

> merge(frame1, frame2, by.x=1, by.y=1, all = TRUE)

# **Graphics Utilities**





Unique objects: All = 25; S1 = 18; S2 = 16; S3 = 20; S4 = 22; S5 = 18

# Some Graphics Commands

```
## Dot plots
> plot(1:10)
> plot(iris[,1:4])
## Barplot
> barplot(1:10)
## Help with plots
> ?plot; ?par
```

#### **Demo Graphics Utilities**

# Outline

#### Introduction

Look and Feel of the R Environment R Library Depositories Installation Getting Around Basic Syntax Data Types and Subsetting Basic Calculations Reading and Writing External Data Some Great R Functions Graphics Utilities

#### **Online Tutorial**

### More Details and Exercises

#### • Continue in R Manual