

UCLA Department of Statistics
Statistical Consulting Center

Basic R

Ryan Rosario
ryan@stat.ucla.edu

April 19, 2009

Outline

- I. Preliminaries
- II. Variable Assignment
- III. Working with Vectors
- IV. Working with Matrices
- V. From Vectors to Matrices
- VI. More on Handling Missing Data
- VII. The Help System
- VIII. Datasets in R
- IX. Overview of Plots in R
- X. R Environment
- XI. Common Bugs and Fixes
- XII. Online Resources for R
- XIII. Exercises
- XIV. Upcoming Mini-Courses

Part I

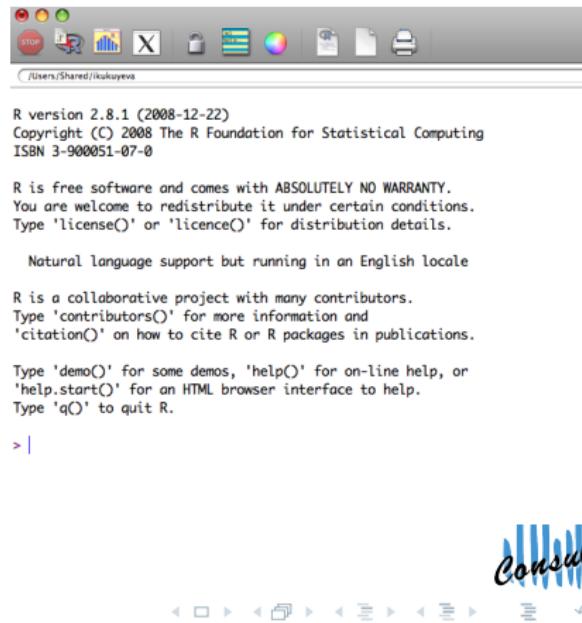
Preliminaries

Installing R on Mac

- ① Go to

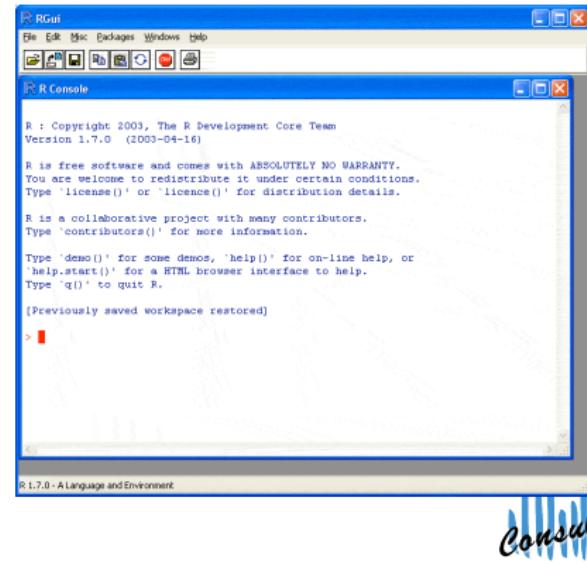
<http://cran.r-project.org>
and select MacOS X.

- ② Select to download the latest version: 2.8.1
- ③ Install and Open. The R window should look like this:



Installing R on Windows

- ➊ Go to
<http://cran.r-project.org>
and select Windows.
- ➋ Select base to install the
R system.
- ➌ Click on the large
download link. There is
other information
available on the page.
- ➍ Install and Open. The R
window should look like
this:



Consulting
ucla

Part II

Variable Assignment

Creating Variables I

- To use R as a calculator, type `2 + 5` and hit ENTER.
(Note how R prints the result.) Your output should look like this:

```
[1] 7
```

- To create variables in R, use either `<-` or `=`:

```
1 # Approach 1  
2 a=5  
3 a  
4 # Approach 2  
5 b<-5  
6 b
```

Creating Variables II

Caution!

Be careful when using `<-` to compare a variable with a negative number!

```
1 #Assign a value to a
2 a<-2
3 #Is a less than -5?
4 a<-5
5 a
6 [1] 5 #Expected FALSE
```



Creating Variables III

Use spaces so that R will not be confused. It is better to use parentheses instead.

```
1 a <- 5  
2 a < -2  
3 [1] FALSE
```

Creating Variables IV

Caution!

It is important not to name your variables after existing variables or functions. For example, a bad habit is to name your data frames data. `data` is a function used to load some datasets.

If you give a variable the same name as an existing constant, that constant is overwritten with the value of the variable. So, it is possible to define a new value for π .

Creating Variables V

Caution!

On the other hand, if you give a variable the same name as an existing function, R will treat the identifier as a variable if used as a variable, and will treat it as a function when it is used as a function:

```
c <- 2 #typing c yields "2"  
c(c,c) #yields a vector containing two 2s.
```

Creating Variables VI

Caution!

As we have seen, you can get away with using the same name for a variable as with an existing function, but you will be in trouble if you give a name to a function and a function with that name already exists.

Part III

Working with Vectors

Creating Vectors I

- Scalars are the most basic vectors.
- To create vectors of length greater than one, use the concatenation function `c()`:

```
1 d=c(3,4,7); d
```

```
[1] 3 4 7
```

The More You Know...

The semicolon ; is used to combine multiple statements on one line.



Creating Vectors II

- To create a null vector:

```
1 x=c() ; x
```

NULL

Creating Vectors III

- Creating a vector with equal spacing, use the sequence function `seq()`:

```
1 e = seq(from=1, to=3, by=0.5); e
```

```
[1] 1.0 1.5 2.0 2.5 3.0
```

- Creating a vector of a given length, use the repeat function `rep()`:

```
1 f = rep(NA, 6); f
```

```
[1] NA NA NA NA NA NA
```

Some Useful Vector Functions I

- To find the length of the vector, use `length()`:

```
1 length(d)
```

```
[1] 3
```

- To find the maximum value of the vector, use the maximum function `max()`:

```
1 max(d)
```

```
[1] 7
```

Some Useful Vector Functions II

- To find the minimum value of the vector, use the minimum function `min()`:

```
1 min(d)
```

```
[1] 3
```

- To find the mean of the vector, use `mean()`:

```
1 mean(d)
```

```
[1] 4.666667
```



Some Useful Vector Functions III

- To sort the vector, use `sort()`:

```
1 g<-c(2,6,7,4,5,2,9,3,6,4,3)
2 sort(g, decreasing=TRUE)
```

```
[1] 9 7 6 6 5 4 4 3 3 2 2
```

Caution!

Although T and F work in place of TRUE and FALSE, it is not recommended.

Some Useful Vector Functions IV

- To find the unique elements of the vector, use `unique()`:

```
1 unique(g)
```

```
[1] 2 6 7 4 5 9 3
```

Some Useful Vector Functions V

- Alternatively, to find the elements of the vector that repeat, use `duplicated()`:

1 `duplicated(g)`

```
[1] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE  
[10] TRUE TRUE
```

Some Useful Vector Functions VI

To determine if a value is missing (NA), use `is.na`. This is useful for finding missing values and removing them, or doing something else with them.

```
1 a <- c(1, 2, 3, NA, 6)
2 is.na(a)
```

```
[1] FALSE FALSE FALSE TRUE FALSE
```

But some functions do not tolerate missing values.

Some Useful Vector Functions VII

Caution!

```
mean(a)
```

```
[1] NA
```

```
mean(a, na.rm=TRUE)
```

```
[1] 1.5
```

Some Useful Vector Functions VIII

To get the number of missing values in a vector,

```
1 sum(is.na(a))
```

```
[1] 1
```

There are other ways to handle missing values. See
?na.action.

Some Useful Vector Functions IX

One final common function you can use on vectors (and other objects) is `summary`.

```
1 summary(a)
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	1.00	1.75	2.50	3.00	3.75	6.00
NA's						
	1.00					

Some Useful Vector Functions X

There are many, many other functions you can use on vectors.

See your reference card for more information!



Comparisons in R

&: in logic, is used to mean AND

Comparisons in R

&: in logic, is used to mean AND
|: in logic, is used to mean OR

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

`>`: in logic, is used to mean GREATER THAN

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

`>`: in logic, is used to mean GREATER THAN

`<`: in logic, is used to mean LESS THAN

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

`>`: in logic, is used to mean GREATER THAN

`<`: in logic, is used to mean LESS THAN

`<-` OR `=`: for variable assignment

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

`>`: in logic, is used to mean GREATER THAN

`<`: in logic, is used to mean LESS THAN

`<-` OR `=`: for variable assignment

`?`: for use with a function in R, to determine what arguments to use, examples and background information

Comparisons in R

`&`: in logic, is used to mean AND

`|`: in logic, is used to mean OR

`==`: in logic, is used for comparison (double equal sign)

`>`: in logic, is used to mean GREATER THAN

`<`: in logic, is used to mean LESS THAN

`<-` OR `=`: for variable assignment

`?`: for use with a function in R, to determine what arguments to use, examples and background information

- Example: `?mean`

Subsetting with Vectors I

- To find out what is stored in a given element of the vector, use [] :

```
1 d[2]
```

```
[1] 4
```

- To see if the elements of a vector equal a certain number, use ==:

```
1 d==3
```

```
[1] TRUE FALSE FALSE
```

Subsetting with Vectors II

- To see if any of the elements of a vector do not equal a certain number, use !=:

```
1 d != 3
```

```
[1] FALSE TRUE TRUE
```

- To obtain the element number of the vector when a condition is satisfied, use which():

```
1 which(d == 4)
```

```
[1] 2
```

To store the result, type: a=which(d==4); a.

Subsetting with Vectors III

- We can also tell R what we *do not* want when subsetting by using the minus – sign. To obtain everything but the 2nd element,

```
1 d <- seq(1,10,2)  
2 d[-2]
```

```
[1] 1 5 7 9
```

Subsetting with Vectors IV

There is no function to remove an element from a vector.
Instead, we use this subsetting by exclusion.

```
1 d <- seq(1,10,2)  
2 d <- d[-2]
```

Subsetting with Vectors V

We can use subsetting to explicitly tell R what observations we want to use. To get all elements of d greater than or equal to 2,

```
1 d[d >= 2]
```

```
[1] 3 5 7 9
```

- R will return values of d where the expression within brackets is TRUE. Think of these statements as: "give me all d such that $d \geq 2$."

We will see more sophisticated (realistic) examples in the exercises...

Part IV

Working with Matrices

Creating Matrices I

To create a matrix, use the `matrix()` function:

```
1 mat<-matrix(10:15, nrow=3, ncol=2); mat
```

```
[,1] [,2]  
[1,] 10 13  
[2,] 11 14  
[3,] 12 15
```



Some Useful Matrix Functions I

To add two matrices, use +

`mat+mat`

```
[,1] [,2]  
[1,] 20 26  
[2,] 22 28  
[3,] 24 30
```

Some Useful Matrix Functions II

To find the transpose of a matrix, use `t()`:

```
t(mat)
```

```
[,1] [,2] [,3]  
[1,] 10 11 12  
[2,] 13 14 15
```

To find the dimensions of a matrix, use `dim()`:

1 dim(mat)

```
[1] 3 2
```

Some Useful Matrix Functions III

To multiply two matrices, use `%*%`.

Note: If you use `*` instead, you will be performing matrix multiplication element-wise.

1 `mat %*% t(mat)`

```
[,1] [,2] [,3]
[1,] 269 292 315
[2,] 292 317 342
[3,] 315 342 369
```

Alternatively, we can find the rows and columns of the matrix, by `nrow()` and `ncol()`:



Some Useful Matrix Functions IV

```
1 nrow(mat) ; ncol(mat)
```

Subsetting with Matrices I

- To see what is stored in the first element of the matrix, use
[] :

```
1 mat[1,1]
```

```
[1] 10
```

- To see what is stored in the first row of the matrix:

```
1 mat[1,]
```

```
[1] 10 13
```

Subsetting with Matrices II

- To see what is stored in the second column of the matrix:

```
1 mat[, 2]
```

```
[1] 13 14 15
```

- To extract elements 1 and 3 from the second column, use `c()` and `[]`:

```
1 mat[c(1,3), 2]
```

```
[1] 13 15
```

Part V

From Vectors to Matrices

Creating Matrices from Vectors I

- To stack two vectors, one below the other, use `rbind()`:

```
1 mat1 <- rbind(d,d); mat1
```

```
[,1] [,2] [,3]  
d     3     4     7  
d     3     4     7
```

- To stack two vectors, one next to the other, use `cbind()`:

```
1 mat2 <- cbind(d,d); mat2
```



Creating Matrices from Vectors II

```
d d  
[1,] 3 3  
[2,] 4 4  
[3,] 7 7
```

Part VI

More on Handling Missing Data

Missing Data in Matrices I

- Start by creating a matrix with missing data:

```
1 h=matrix(c(NA,3,1,7,-8,NA), nrow=3,  
          ncol=2, byrow=TRUE); h
```

```
[,1] [,2]  
[1,] NA    3  
[2,] 1     7  
[3,] -8   NA
```

Missing Data in Matrices II

- To see if any of the elements of a vector are missing use `is.na()`:

```
1 is.na(h)
```

```
[,1] [,2]
```

```
[1,] TRUE FALSE
```

```
[2,] FALSE FALSE
```

```
[3,] FALSE TRUE
```

Missing Data in Matrices III

- To see how many missing values there are, use `sum()` and `is.na()` (TRUE=1, FALSE=0):

```
1 sum(is.na(h))
```

```
[1] 2
```

- To obtain the element number of the matrix of the missing value(s), use `which()` and `is.na()`:

```
1 which(is.na(h))
```

```
[1] 1 6
```



Missing Data in Matrices IV

- To keep only the rows without missing value(s), use `na.omit()`

```
1 na.omit(h)
```

```
[,1] [,2]  
[1,]    1    7  
attr(,"na.action")  
[1] 1 3  
attr(,"class")  
[1] "omit"
```

Part VII

The Help System

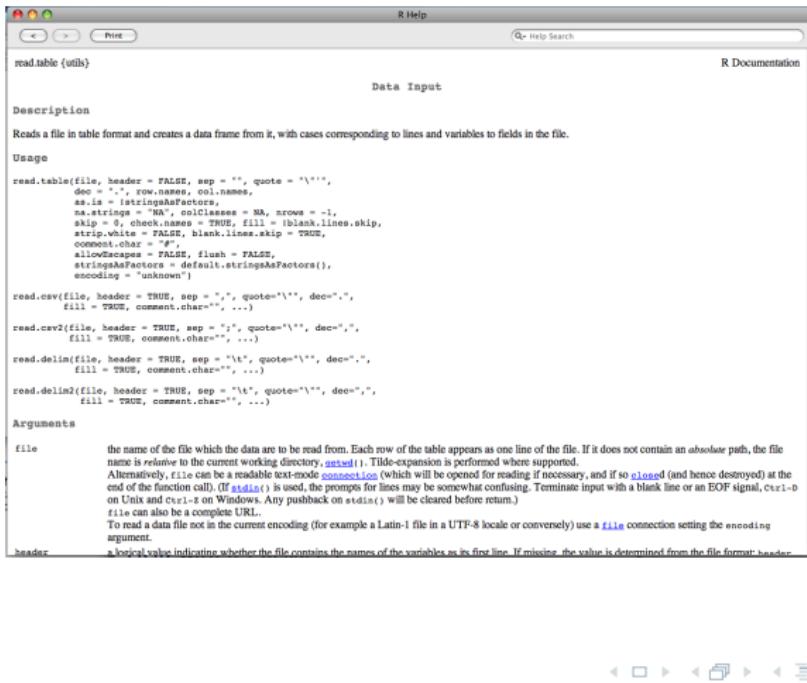
Help with a Function I

- To get help with a function in R, use ? followed by the name of the function.

```
?read.table
```

- By the way, `help(function_name)` also works.

Help with a Function II



Help with a Package I

To get help with a package, use `help(package="name")`.

```
help(package="MASS")
```

Help with a Package

Help with a Package II

Information on package 'MASS'

Description:

Bundle: VR
Priority: recommended
Contains: MASS class nnet spatial
Version: 7.2-44
Date: 2008-08-13
Depends: R (>= 2.4.0), grDevices, graphics, stats, utils
Suggests: lattice, nlme, survival
Author: S original by Venables & Ripley. R port by Brian Ripley <ripley@stats.ox.ac.uk>, following earlier work by Kurt Hornik and Albrecht Gebhardt.
Maintainer: Brian Ripley <ripley@stats.ox.ac.uk>
BundleDescription: Functions and datasets to support Venables and Ripley, 'Modern Applied Statistics with S' (4th edition).
License: GPL-2 | GPL-3
URL: http://www.stats.ox.ac.uk/pub/MASS4/
Packaged: Wed Aug 13 14:07:54 2008; ripley
Package: MASS
Description: The main library and the datasets
Title: Main Package of Venables and Ripley's MASS
LazyLoad: yes
LazyData: yes
Built: R 2.8.0; universal-apple-darwin8.11.1; 2008-11-25 11:22:59; unix
Index:
Functions:
=====

Null Null Spaces of Matrices
addterm Try All One-Term Additions to a Model
anova.negbin Likelihood Ratio Tests for Negative Binomial GLMs
area Adaptive Numerical Integration
bandwidth.nrd Bandwidth for density() via Normal Reference



Searching for Help I

To get help with a package, use `help.search()`.

`help.search("regression")`

Searching for Help II

Help topics matching 'regression'		
Topic	Package	Description
lm.ridge	MASS	Ridge Regression
lqs	MASS	Resistant Regression
polr	MASS	Ordered Logistic or Probit Regression
rms.curv	MASS	Relative Curvature Measures for Non-Lin
alrweb	alr3	Access to the Applied Linear Regression v
boot.case	alr3	Case bootstrap for regression models
conf.intervals	alr3	Compute marginal confidence intervals fo
delta.method	alr3	Standard error of a nonlinear function of
mantel	alr3	Mantel's artificial data for stepwise regres
sigma.hat	alr3	Return the scale estimate for a regressor
mat.regress	psych	Multiple Regression from matrix input
addplot	sm	Internal sm functions

Part VIII

Datasets in R

Data from the Internet I

When downloading data from the internet, use `read.table()`. In the arguments of the function:

- `header`: if TRUE, tells R to include variables names when importing
- `sep`: tells R how the entries in the data set are separated
 - `sep=","`: when entries are separated by COMMAS
 - `sep="\t"`: when entries are separated by TAB
 - `sep=" "`: when entries are separated by SPACE

Data from the Internet II

```
1 df<-read.table("http://www.stat.ucla.edu  
2 /~rosario/scc/textbooks.csv",  
3 header=TRUE, sep=",")
```

Importing Data from Your Computer I

- ① Check what folder R is working with now:

```
1 getwd()
```

- ② Tell R in what folder the data set is stored (if different from (1)). Suppose your data set is on your desktop:

```
1 setwd("~/Desktop")
```

- ③ Now use the `read.table()` command to read in the data, substituting the name of the file for the website.

Using Data Available in R I

To use a data set available in one of the R packages, install that package (if needed).

Load the package into R, using the `library()` function.

```
1 library(alr3)
```

Extract the data set you want from that package, using the `data()` function. In our case, the data set is called UN2.

```
1 data(UN2)
```

Working with Datasets in R I

To use the variable names when working with data, use
`attach()`:

```
1 data(UN2)  
2 attach(UN2)
```

After the variable names have been "attached", to see the variable names, use `names()`:

```
1 names(UN2)
```

To see the descriptions of the variables, use `?:`

```
1 ?UN2
```

Working with Datasets in R II

After modifying variables, use `detach()` and `attach()` to save the results:

```
# Make a copy of the data set  
UN2.copy<-UN2  
detach(UN2)  
attach(UN2.copy)  
# Change the 10th observation for logFertility  
UN2.copy[10, 2]<-999
```

Working with Datasets in R III

To get an overview of the data sets and its variables, use the `summary()` function:

```
1 # Check that the change has been made  
2 summary(UN2)  
3 summary(UN2.copy)
```

Working with Datasets in R IV

Caution!

Avoid using `attach()` if possible. Many strange things can occur if you accidentally attach the same data frame multiple times, or forget to `detach`. Instead, you can refer to a variable using `$`. To access the `Locality` variable in data frame `UN2`, use `UN2$Locality`.

“`attach` at your own risk!”

Working with Datasets in R V

To get the mean of all the variables in the data set, use
`mean()`:

```
mean(df,na.rm=TRUE)
```

Working with Datasets in R VI

```
> mean(df,na.rm=TRUE)
```

Author	Title	Department
NA	NA	NA
Amazon.List.Price	Amazon.Sale.Price	Barnes.Noble.Price
46.83325	42.48925	47.78401
Textbooks.com.Price	Borders.Price	
88.71250	52.38154	

Warning messages:

```
1: In mean.default(X[[1L]], ...) :  
  argument is not numeric or logical: returning NA
```

....

Working with Datasets in R VII

To get the correlation matrix of all the (numerical) variables in the data set, use `cor()`:

```
1 cor(df[c(4,5), c(4,5)])
```

	Amazon.List.Price	Amazon.Sale.Price
Amazon.List.Price	1	-1
Amazon.Sale.Price	-1	1

Can obtain the variance-covariance matrix using `var`.

Part IX

Overview of Plots in R



Creating Plots in R I

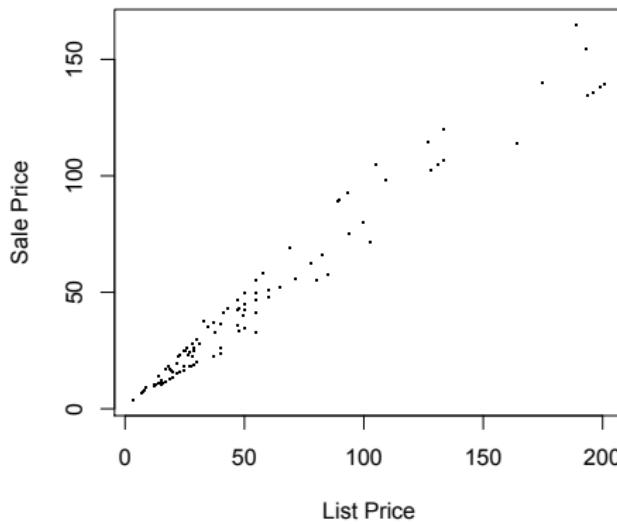
To make a plot in R, you can use `plot()`:

```
1 plot(Amazon.Sale.Price ~ Amazon.List.Price,  
2       main="Comparison of Amazon List and  
            Sale Price",  
3       xlab="List Price",  
4       ylab="Sale Price", pch='.')
```

Creating Plots in R I

Creating Plots in R II

Comparison of Amazon List and Sale Price



Creating Plots in R I

To make a histogram in R, you can use `hist()`:

```
1 hist(Amazon.Sale.  
      Price, main = "  
      Distribution of  
      Amazon Sale Prices  
      ", xlab = "Sale Price  
      ", br = 20)
```

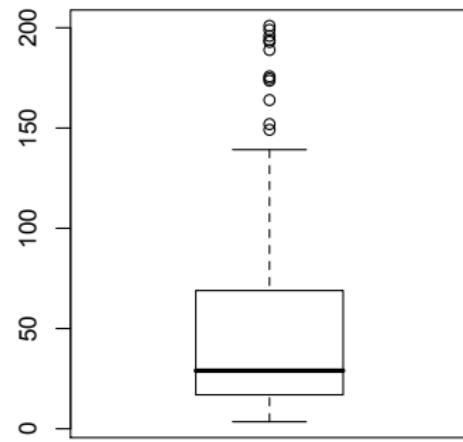


Creating Plots in R II

To make a boxplot in R, you can use `boxplot()`:

```
1 boxplot(Borders.  
          Price, main="  
          Distribution of  
          Borders Prices")
```

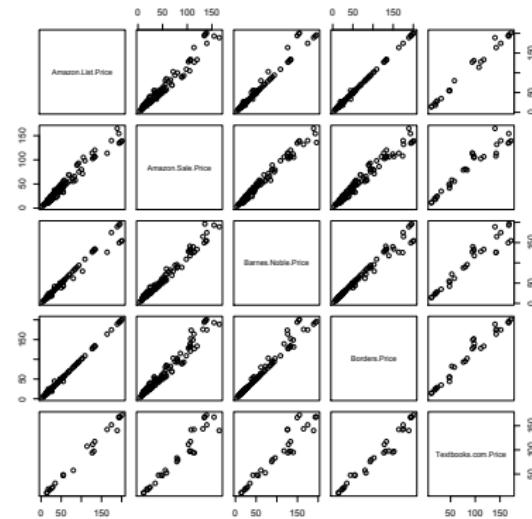
Distribution of Borders Prices



Creating Plots in R III

To make a scatterplot for all the (numerical) variables, you can use `pairs()`:

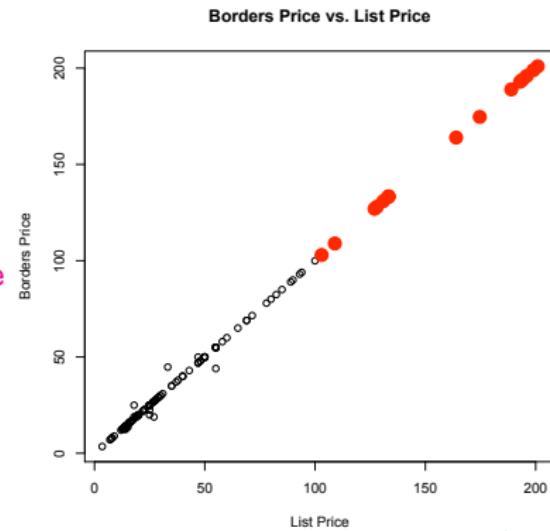
```
1 pairs(df[, c(4:8)])
```



Creating Plots in R IV

To add more points to an existing plot, use `points()`. Here, we add a red dot for books with list cost > \$100.

```
1 plot(Borders.Price~Amazon.List.  
      Price, xlab="List Price",  
      ylab="Borders Price", main="  
      Borders Price vs. List Price  
      ")  
2 points(Borders.Price[Borders.  
      Price > 100], Amazon.List.  
      Price[Borders.Price>100],  
      col="red", pch=19, cex=2)
```



Creating Plots in R V

Caution!

Once a plot is constructed using `plot`, it cannot be modified.

To overlay things on a rendered plot, use one of the following

- ① `abline` - add a line with slope b , intercept a or horizontal/vertical.
- ② `points` - add points.
- ③ `lines` - add lines.



Saving Plots as a PDF I

Note: The files will be saved in the folder specified with `setwd()`.

To save a plot in R as a PDF, you can use `pdf()`:

```
1 pdf("myniftyplot.pdf", width=6, height=6,  
     onefile=FALSE)  
2 boxplot(Borders.Price, main="Distribution  
          of Borders Prices")  
3 dev.off()
```

Part X

R Environment

Exploring R Objects I

- To see the names of the objects available to be saved (in your current workspace), use `ls()`.

1 ls()

```
[1] "UN2" "a" "b" "d" "data" "e" "f" "h" "mat1" "mat2"
```

- To remove objects from your workspace, use `rm()`.

1 rm(d)

2 ls()

```
[1] "UN2" "a" "b" "data" "e" "f" "h" "mat1" "mat2"
```



Exploring R Objects II

- To remove *all* the objects from your workspace, type:

```
1 rm(list=ls())
2 ls()
```

character(0)

Saving and Loading R Objects I

- To save (to the current directory) all the objects in the workspace, use `save.image()`.

```
1 save.image("basicR.RData")
```

- To load (from the current directory), use `load()`.

```
1 load("basicR.RData")
```

Saving and Loading R Objects I

- To save (to the current directory) a **single** object in the workspace, use `save()`.

```
1 save(df, file = "textbooks.RData")
```

- To load (from the current directory), use `load()`.

```
1 load("textbooks.RData")
```

Exporting R Objects to Other Formats I

- To save (to the current directory) certain objects in the workspace to be used in Excel, use `write.csv()`.

```
1 write.csv(df, file="textbooks.csv")
```

Saving R Commands

Saving R Commands I

- To see all of the commands you typed in an R session, click on the Yellow and Green Tablet



```
ion 2.7.2 (2008-08-25)
ght (C) 2008 The R Foundation for Statistical Computing
-900051-07-0
```

```
ree software and comes with ABSOLUTELY NO WARRANTY.
e welcome to redistribute it under certain conditions.
license()' or 'licence()' for distribution details.
```

```
ral language support but running in an English locale
```

```
collaborative project with many contributors.
contributors()' for more information and

```

Saving R Commands II

- To save all of the commands you typed in an R session, use:

```
1 savehistory(file="history.log")
```

Saving R Commands III

- Alternatively, use a `.r` file to store your commands.
 - ① Go to: File -> New Document
 - ② Type your commands
 - ③ Save the file as "code.r"
 - ④ Go back to the R Console
 - ⑤ To run all the commands, use:

```
1 source("code.r")
```

The More You Know...

Use the # sign to write comments in your code. Use them!



Part XI

Common Bugs and Fixes

Error: syntax error

Possible causes:

- Incorrect spelling (of the function, variable, etc.)
- Including a "+" when copying code from the Console
- Having an extra parenthesis at the end of a function
- Having an extra bracket when subsetting

Trailing +

Possible causes:

- Not closing a function call with a parenthesis
- Not closing brackets when subsetting
- Not closing a function you wrote with a squiggly brace

You can escape this sticky situation by hitting the ESCAPE key to exit your command.

Error in . . . : requires numeric matrix/vector arguments

Possible causes:

- ① Objects are data frames, not matrices
- ② Elements of the vectors are characters

Possible solutions:

- ① Coerce (a copy of) the data set to be a matrix, with the `as.matrix()` command
- ② Coerce (a copy of) the vector to have numeric entries, with the `as.numeric()` command

Part XII

Online Resources for R

CRAN

<http://cran.stat.ucla.edu/>



The Comprehensive R Archive Network

Frequently used pages

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:

- [Linux](#)
- [MacOS X](#)
- [Windows](#)

Source Code for all Platforms

Windows and Mac users most likely want 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 (2008-12-22): [R-2.8.1.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).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

Questions About R

- 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.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the [R project homepage](#) for

R-Seek Search Engine



[Volunteer to add R sites](#) - [Add to Google Toolbar](#) - [Add to Firefox/IE](#) - [Task Views](#) - [Ref Card](#) - [Google Code Search](#) - [Email suggestions to Sasha Goodman](#)



UCLA Statistics Bootcamp Resources

R Bootcamp is a day-long version of this mini-course.
Handouts and datasets from Bootcamp 2007 and 2008 can be
found on my website:

<http://www.stat.ucla.edu/~rosario/boot08/>

<http://www.stat.ucla.edu/~rosario/boot/>

UCLA Statistics Information Portal

<http://info.stat.ucla.edu/grad/>

UCLA Department of Statistics
Department Information Portal

SPRING 2008 - WEEK 2

Help

- Software
 - » Home
 - » R Help
 - » GRASS Help
 - » LaTeX Help

 **R**
 An open source solution to statistical computing

Editor notes appear in *italics*. Sites marked with  are hosted on UCLA campus.

Primary References

R Project Main Site
 The main web site for the developers of R. A great resource worth browsing.
 Download R for your System:
 Mac Intel  | Windows 

UCLA Statistics CRAN Repository 
 Optimal source for downloading R and R contributed packages. This repo is fastest due to its network proximity.

RSeek Search Engine
 Search tons of R resources. Can't (easily) find what you're looking for? This site at Stanford should be the next place you look!
 The most useful resource when you are looking for something specific.

R Wiki
 provides user-editable help pages for many R-related topics and problems.

R Graphics Gallery
 Provides several different graphics fully created with R. You can browse exemplar graphs by topic, package, or user ranking.

R Reference Card 
 Lists frequently used functions and commands in R!

Links

- Department**
 - » Info
 - » Web Mail
 - » Department Directory
 - » Support
 - » Student Services
 - » Centers
 - » Graduate Courses
 - » Undergraduate Courses
 - » Main Home Page
- University**
 - » myUCLA
 - » URSA
 - » BruinWalk
 - » Graduate Division
 - » Schedule of Classes
 - » Academic Calendar
 - » Campus Map
 - » UCLA Directory

UCLA Statistical Consulting Center E-consulting and Walk-in Consulting

<http://scc.stat.ucla.edu>

Statistical Consulting Center

Forum for the R statistical computing and graphical language and environment.

R : Statistical Consulting Center Forums				
Subject	Views	Written By	Posted	Pages: 1
> weights case	161	joscani	03/09/2009 09:01AM	
> Re: weights case	75	David Diez	03/10/2009 08:35AM	
> Re: weights case	96	joscani	03/10/2009 11:41AM	
> Re: weights case	66	Jan de Leeuw	03/11/2009 06:07PM	
> Adding matrices in an array	190	Mine Cetinkaya	03/02/2009 03:06PM	
> Re: Adding matrices in an array	85	David Diez	03/02/2009 06:33PM	
> Re: Adding matrices in an array	54	Mine Cetinkaya	03/03/2009 07:49AM	
> Re: Adding matrices in an array	53	David Diez	03/03/2009 07:59AM	
> Re: Adding matrices in an array	55	Mine Cetinkaya	03/03/2009 09:11AM	



Part XIII

Exercises

Exercises in R

- ① Load in the UCLA textbook price comparison data from <http://www.stat.ucla.edu/~rosario/scc/textbooks.csv>. It is a CSV file with a header. Find the average list price of textbooks for courses in Management. (Hint: `summary(data$Department)`)
- ② Plot the Amazon list price vs. the Amazon sales price and Amazon list price vs. Barnes & Noble price on the same plot. Use different plotting symbols for Amazon and Barnes and Noble. Add a dashed line representing the location on the plot where the sales price of a book is equal to the list price. Highlight in red those Amazon books that differ from the list price by more than 25%. Label the plot and the axes.

Solution to the Exercises

Exercise 1 Solution

```
1  mean(Amazon.List.Price[Department == "XMANG"],  
  na.rm=TRUE)
```

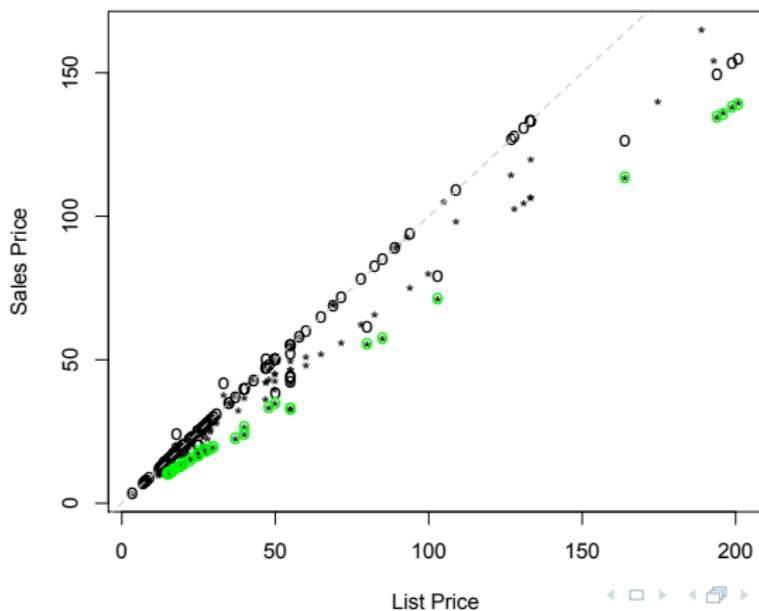
Solution Exercise 2

```
1 plot(Amazon.Sale.Price ~ Amazon.List.Price, pch='*',
      main="Comparison of UCLA Textbook
      Prices: Amazon vs. BN", xlab="List Price",
      ylab="Sales Price")
2 points(Barnes.Noble.Price ~ Amazon.List.Price,
      pch='o')
3 abline(a=0, b=1, lty=2, col="grey")
4 good.deals <- which(abs((Amazon.List.Price -
      Amazon.Sale.Price) / Amazon.List.Price) >=
      0.25)
5 points(Amazon.Sale.Price[good.deals] ~ Amazon.
      List.Price[good.deals], pch='o', col="green")
```

Solution to Exercise 2

Solution Exercise 2

Comparison of UCLA Textbook Prices: Amazon vs. BN



Part XIV

Upcoming Mini-Course

Upcoming Mini-Courses

- Thursday (April 16) and Tuesday (April 21):
 - Migrating to R for SAS/SPSS/Stata Users
- For a schedule of all mini-courses offered please visit <http://scc.stat.ucla.edu/mini-courses>.

We Want to Hear From You!

Please complete our online survey. Your feedback will help us improve our mini-course series. We greatly appreciate your time.

<http://scc.stat.ucla.edu/survey>