Preparing the workshop

Before we begin, please download the "hmnrghts.txt" and "Simulation Script.txt" files from the GradQuant website, under the R series.

gradquant.ucr.edu/gradquant-resources/workshop-resources/

Introduction to R

Jianan Hui

GradQuant Sponsored

University of California Riverside

April 27, 2015

Overview

Introduction

What is R?

- R is a platform for the object-oriented statistical programming language S.
- R is a shareware version of S-plus, and they are quite similar.
- Essentially R can be used as either a matrix-based programming language or as a standard statistical package that operates much like STATA or SPSS.

Where can I get R?

- The beauty of R is that it's shareware, so it's free to anyone.
- To obtain R for Windows (or Mac) go to The Comprehensive R Archive Network (CRAN) at `http://www.r-project.org`. Just download the executable file, and it will install itself.

Advantages and Disadvantages

Advantages:

- Open Source (Free and lots of flexibility)
- Cross platform
- Graphical Capabilities

Disadvantages:

- Primarily syntax based
- Not enough documentation sometimes

Getting Started

```
> print("Welcome to R")
```

Preliminaries on entering command

- R is case sensitive.
- Commands are separated either by a semi-colon (';'), or by a new line.
- Comments can be put in text, starting with a hashmark #.
- An object name must start with an alphabetical character, but may contain numeric characters thereafter. A period may also form part of the name of an object. For example, x.1 is a valid name.
- If a command is not complete at the end of a line, R will give a different prompt, by default (+).

Need help?

- Use help(functionname) to get help.
  ?functionname would also work.
- Use args(functionname) for a quick reminder of the function arguments
- Use example (functionname) to see examples.
- Use ??keyword for help searching.
  help.search(keyword)

Working directory

A working directory is the default location where R looks for files.

- To see the current working directory:
  > getwd()
  "C:/Documents and Settings/user/My Documents"
- To change working directory (examples)
  > setwd("C:\\Documents\\ $\cdots$) — For Windows
  > setwd("~/Documents/ $\cdots$ ") — For Linux/Mac
- Note the direction of the slashes for different operating systems.

Inputting Data as objects

- Scalar:
  $> a = 2$
- Vector:
  $> x = c(2,4,5)$
- Matrix:
  $> M = matrix(c(1,2,3,4,5,6),3,2)$

Displaying Objects

To display your data, type the object name:

- x:
  [1] 2 4 5
- M:

```
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

Displaying and Removing Objects

- Often we would like to know which objects are stored in memory. We can call a list of objects by using the command:
  > ls()
  > "a" "x" "M"
- Lets remove our scalar object by:
  > rm(a)
- More examples!

Dataset Preparation

A working directory is the default location where R looks for files.

- We will use the hmnrghts datafile.
  To download this file, go to:
  `http://www.gradquant.ucr.edu/workshop-resources/`
- Please move this file into your working directory.

Reading in Data

- Once you have the working directory correctly set, reading in data requires just a single command.
- Let's start with reading in a delimited text file. Reading in a delimited text file (.txt) requires the read.table command.
- > hmnrghts = read.table("hmnrghts.txt", header=TRUE)

External R packages

- The library function is used to enable installed external R packages.
- Lets load the package foreign for future use:
  >library(foreign)
- Note: assume it loaded if no error messages.

More options of reading in data

- When we ran library(foreign), we have opened up a whole list of commands for importing files from other statistical software.
- You can import the Stata data using this command:
  > read.dta(*.dta)
- For missing data, you can use the na.omit() command. Note that changes we make to hmnrghts will not affect the file "hmnrghts.txt".

Data Frame

- hmnrghts is a data frame. Let's see what it looks like:
  >hmnrghts
- Each column of this data frame is an object. To display a specific object:
  >hmnrghts$country
- To bypass having to specify the "dataframe$object", we can attach the dataframe:
  >attach(hmnrghts)
- Now you can display country by:
  >country
- Try displaying the object 'lpop' that is in the hmnrghts dataframe.

Modifying Data

Modifying Data

Often we need to recode variables, and there are a variety of ways to do this in R.

- Suppose we want the actual population of each country instead of its logarithm:
  > pop = exp(lpop)
- Dummy variable:
  - Want to compare the highest category of sdnew (scale of Political Terror) to the rest.
    > sdnew.dummy = as.numeric(sdnew>3)
  - Want a 3 category ordinal level variable for the population.
    > pop.3 = lpop
    > pop.3[lpop <15.18] = 1
    > pop.3[lpop >= 15.18 & lpop < 17.11] = 2
    > pop.3[lpop >= 17.11] = 3

Logical Statements

Logical statements in R are evaluated as to whether they are TRUE or FALSE.

Table 1: Logical Operators in R

Operator	Means
<	Less Than
<=	Less Than or Equal To
>	Greater Than
>=	Greater Than or Equal To
==	Equal To
!=	Not Equal To
&	And
|	Or

For example, suppose we wanted to know which countries have had a civil war and have above average scale of political terror:

```
> wealthWar = hmnrghts$civ_war ==1 & hmnrghts$sdnew>2.5
```

Modifying Data

- There is one quirk in R when it comes to recoding variables or creating new ones. When you recode or create new variables, those changes are made to the global version of the data frame and not to the specific data frame that you think you are changing.
- To make changes to a specific data set detach it before making the changes and then re-attach it.
- A better way to proceed is referring directly to the data frame:
  $>$ hmnrghts\$pop $=$ exp(hmnrghts\$lpop)

Basic Models

Ordinary Least Squares

Estimating an OLS model in R is done with the lm command.

- Let's predict human rights violations as function of the variables democ and gnpcats:
  > hmnrghts.model = lm(sdnew~democ+gnpcats, data=hmnrghts)
- After typing the above command, R will not automatically print any output. To see the output you have to type the following command:
  > summary(hmnrghts.model)

Ordinary Least Squares

- It is now worth noting that R is an object-oriented environment.
- Whenever we run a model or create a data frame, we create an object, which consists of several components.
- Hence, we can call on these components of the hmnrghts.model object to obtain several other regression outputs including the coefficients, residuals, cov.unscaled (the variance-covariance matrix) and fitted.values.
  > hmnrghts.model$coefficients

Generalized Linear Models

- The glm command allows you to run a variety of models.
- To use the glm command you must specify your formula, the family, and the link function.
- For example, let's say we wanted to predict whether a country had a military regime as a function of GNP and population using a logit model:

```
> mil.model = glm(military~lpop + gnpcats,
family=binomial(link=logit),data=hmnrghts)
> summary(mil.model)
```

Generating Data from Random Variables

Generating Data from Random Variables

- Lets generate 1000 random standard normal variables (mean = 0, standard deviation = 1)
  > z = rnorm(1000,mean = 0, sd = 1)
- Is the mean and standard deviation approximately 0 and 1 respectively?
  > mean(z)
  > sd(z)

One Population T Test

- Lets see if our sample mean is significantly away from 0:
  $>$ t.test(z,mu=0)
- Does anyone have p-value < 0.05?

Exercise

- Try generating 50 random normal variables with mean 0 and standard deviation as 20 and store the results in a variable named z2.
- Then try using a t test to see if the mean is significantly away from 0.

Answer

- Generating 50 normal random variables with mean 0 and standard deviation 20.
  > z2 = rnorm(50,0,20)
- Conducting a t test:
  > t.test(z2,mu=0)

Loops

Loops

- Loops are easy to write in R and can be used to repeat calculations. The basic structure for a loop is:
  > for (i in 1:itr) {COMMANDS}
  > for (i in seq(start,finish,by)) {COMMANDS}
- Example::
  > for(i in seq(1,5)){print(i)}

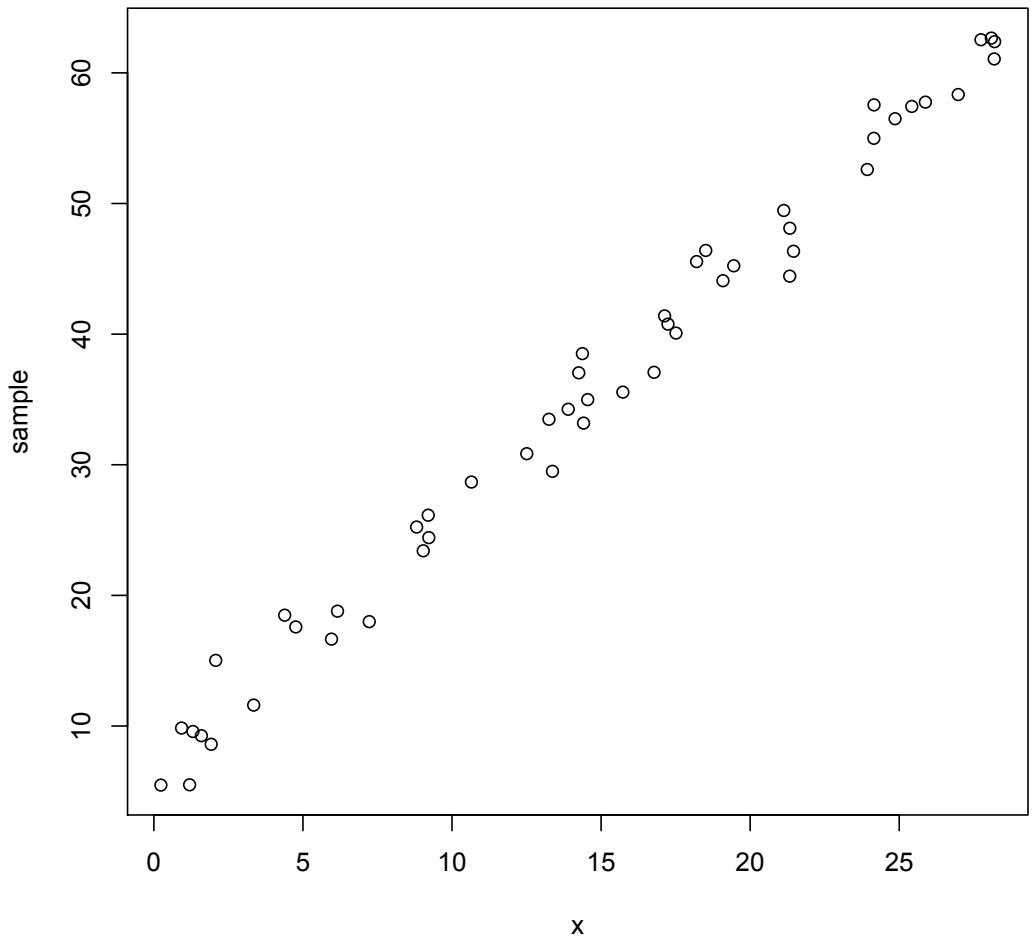
Simulation

Simulation code:

```
intercept = 6
slope = 2
rep = 50
x = double(rep)
sample = double(length(x))
j = 1
for(i in seq(1,rep)){
x[i] = runif(1,0,30)
}
x = sort(x)
for(i in x){
sample[j] = rnorm(1,intercept + slope*i,2)
j = j+1
}
plot(x,sample)
```

Simulation

Simulated Points:



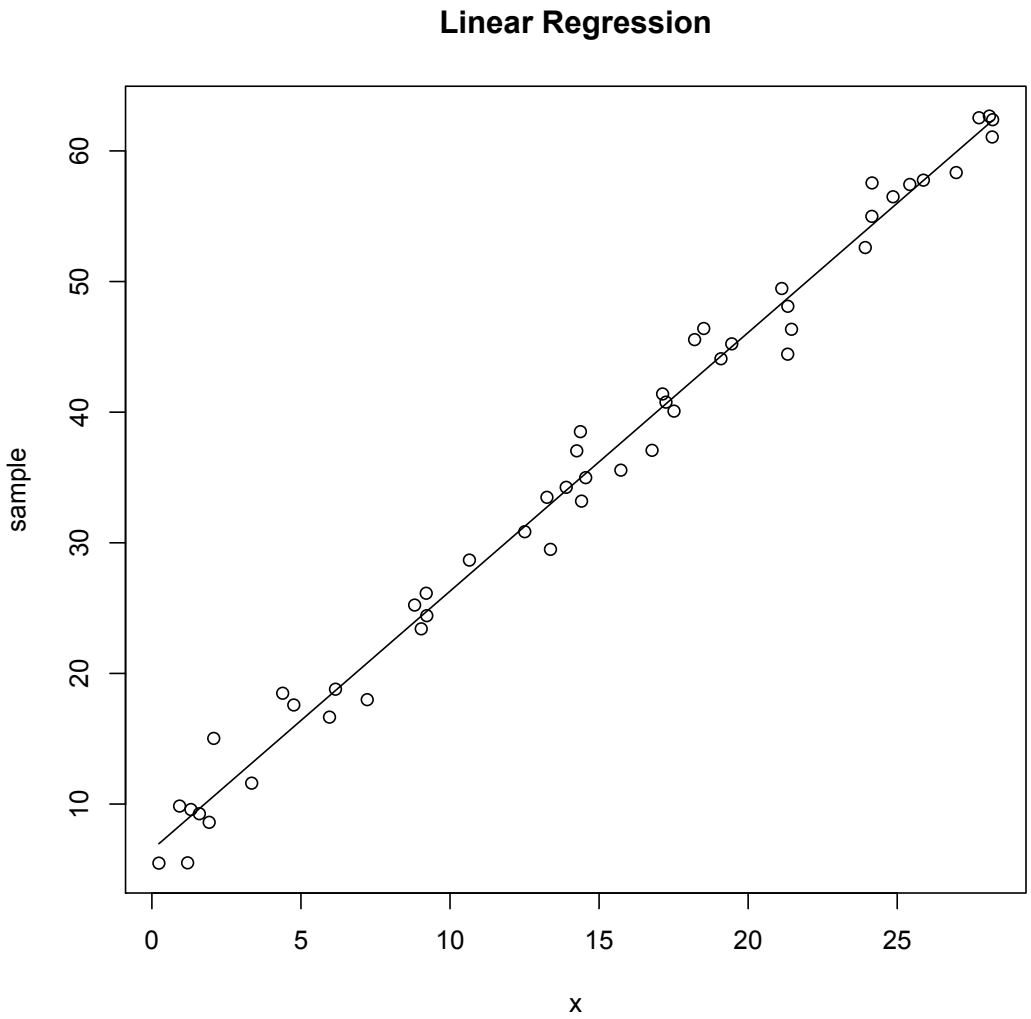
Exercise

- Can you fit a linear model to this data? Save the results as model3. (The y values are sample and the x values are x).
- Is the intercept close to 6 and slope (coefficient of x) close to 2? (Use summary())
- Try using the plot and lines to display your model with the data.
  Hint: The fitted command generates the predicted outcomes. The lines command interpolates the predicted values and adds them to our scatterplot.

Answer

- Fitting the model:
  > model3 = lm(sample ~ x)
- To view summary:
  > summary(model3)
- Plotting the model:
  > plot(x,sample,main=Linear Regression)
  > lines(x=x, y = fitted(model3))

Simulation



Want to learn more?

`http://tryr.codeschool.com/`

Thank you!