Quantitative Methods I: Data & measurement

Johan A. Elkink

University College Dublin

10 September 2018

1 Introduction to course

2 Introduction to R

3 Additional exercise

Outline

Objectives

- Understand data conceps and basic descriptive quantitative analysis tools.
- Work with real datasets to perform basic quantitative analyses.
- Graph data effectively for presentation and analysis.
- Recognize and understand the basics of the linear regression model.
- Use the R statistical software package for analyzing and graphing data.
- Understand sufficient theoretical and practical material to build on in a second, more advanced quantitative methods course.

(Benoit, 2009)

Frequentist vs Bayesian

Frequentist statistics interprets **probability** as the frequency of occurrance in (hypothetically) many repetitions. E.g. if we throw this dice infinitely many times, what proportion of times would it be heads? We can here also talk of **conditional probabilities**: what would this frequency be if ... and some condition follows.

Bayesian statistics interprets probability as a **belief**: if I throw this dice, what do you think is the chance of getting heads? We can now talk of conditional probabilities in a different way: how would your belief change given that ... and some condition follows.

This course is a course in the basics of frequentist statistics.

Homeworks

- 50 % Five homeworks
- 50 % Regression paper
- No exam

Homeworks

- 50 % Five homeworks
- 50 % Regression paper
- No exam

Working with others is a good idea

Grade conversions

Homeworks	UCD	TCD	Homeworks	UCD	TCD
97-100%	A+	A+	54-64%	E+	D
94-96%	A	A	44-53%	E	D
91-93%	A-	A	33-43%	E-	D
88-90%	B+	B+	0-32%	F	F
85-87%	B	B			
83-84%	B-	B			
80-82%	C+	C+			
77-79%	C	C			
74-76%	C	C			
71-73%	D+	C			
68-70%	D	C			
65-67%	D-	C			

Syllabus and website

- Website: http://www.joselkink.net/teaching
- Syllabus downloadable there
- Slides and notes on website
- Data for exercises on website

Outline

R software

`http://www.r-project.org` and `http://www.rstudio.com`

Multi-platform; powerful; lots of free documentation.

There are a lot of built-in functions. Functions have required arguments, optional arguments, and can handle named arguments. R can also be programmed to add new functions.

(Benoit, 2009)

R objects

Everything in R is an **objects**.

Every objects has a **mode** (e.g. logical, numeric, character) and a **class** (e.g. numeric, logical, list, matrix, factor).

To see all objects currently in the work space: `ls()` — to empty the work space: `rm(list = ls())`.

(Benoit, 2009)

R environment and RStudio

RStudio work environment:

- Main interpreter
- List of objects in workspace
- Help area
- Plots
- `View()` area
- R file editor

Note: *always* use the R editor rather than the interpreter (except to test bits of code).

Code good practice

Replicability, but also your own sanity, requires that you write clear R files that are easy to understand. For this reason:

- Always give human-readable names to objects.
- Always add enough whitespace to your code.
- Always include *all* steps of your analysis in your code, and *only* those steps. (You can always keep a folder with old code, or have code "commented out" with explanation.)
- Always add comments (using ##) to your R file.
- Be consistent in your use of capitalization and separating words in a variable name.

R data types

vector – one-dimensional set of items with all the same mode (e.g. a numeric vector).

matrix – two-dimensional (or more) set of items with all the same mode.

list – multi-dimensional set of items that can be of different types, including lists.

data frame – two-dimensional set, where each column is a vector and each row an observation.

function – object that can be executed, taking a set of parameters and producing a return object.

(Benoit, 2009)

Vectors in R

create	`a <- c(1, 4, 2, 3, 3, 2)`
get length	`length(a)`
get subset	`a[3:5]`
compute mean	`mean(a)`
subtract 1	`a - 1`
define sequence	`1:20` *or* `seq(1, 20, 1)`
test condition	`a <3`
select on condition	`a[a <3]`

Matrices in R

create identity matrix	`i <- diag(5)`
get size	`dim(i)`
take row sums	`rowSums(i)`
take diagonal	`diag(i)`
create by binding rows	`rbind(a, a, a <3)`
create by binding columns	`cbind(a, 1:length(a))`

Data frames in R

open example data frame	`data(cars)`
get names	`names(cars)`
get size	`dim(cars)`
get variable summaries	`summary(cars)`
get structure information	`str(cars)`
get first few rows	`head(cars)`
get one variable	`cars$speed`
get subset of rows	`cars[cars$speed >15, ]`

Lists in R

An example of a list is the output of the `lm()` command. E.g.

```
model <- lm(dist ~ speed, data = cars)
```

get names	`names(model)`
get length	`length(model)`
get element	`model[[1]]` *or* `model$coefficients`
get structure information	`str(model)`
add element	`model$myelement <- c(1, 3, 4)`

Saving and loading in R

Saving or loading any object in R can be done with `load()` and `save()`, which is also the most efficient way to save data.

```
save(cars, i, x, file = "test.Rdata")
```

saves three objects in a file, which can simply be opened with

```
load("test.Rdata")
```

Note that there is no assignment to any object – simply all saved objects are added to the work space environment.

Importing and exporting in R

Data frames can also be loaded or saved in other, exchangeable formats, using the `rio` library.

```
data <- import("myData.xlsx")

export(data, file = "myData.dta")
```

Note that loading always requires assignment to an R object.

Basic plots in R

scatterplot

```
plot(dist ~ speed, data = cars)
plot(dist ~ speed, data = cars, pch = 19,
   bty = "n", col = "blue", cex = 1.5,
   main = "Distance by speed",
   xlab = "Speed", ylab = "Distance")
```

saving as PDF

```
dev.copy2pdf("speed_by_distance.pdf")
```

or alternatively

```
pdf("speed_by_distance.pdf")
plot(...)
dev.off()
```

getting help

```
help(plot)
```

Although I would now generally revert to the `ggplot2` library (Wickham, 2010), which we will discuss later.

Finding help in R

getting help	`help(plot)` or `?plot`
search for help	`??plot`
help on package	`help(package = "lattice")`

`http://www.rseek.org/`

Outline

Exercise: Data in R

In addition to the lab sheet:

1. Download the British Election Survey face-to-face 2005 data from `http://www.essex.ac.uk/bes/`
2. Open the data in R
3. Inspect the data
4. Extract a variable as a single vector
5. Select elements of one variable on the basis of scores on another variable
6. Add a new variable based on a computation on (an)other variable(s)
7. Generate a plot and save as PDF
8. Try all commands in the previous slides using this data (or subsets thereof)

Benoit, Kenneth. 2009. "PO 7001: Quantitative Methods I." Lecture slides, Trinity College Dublin.

Wickham, Hadley. 2010. "A Layered Grammar of Graphics." *Journal of Computational and Graphical Statistics* 19(1):3–28.
URL: *http://vita.had.co.nz/papers/layered-grammar.pdf*