Statistical Modelling

Contents

Preliminary remarks 5

1 Introduction to statistical inference 7

2 Linear regression 9

3 Generalized linear models 11

4 Correlated and longitudinal data 13

5 Linear mixed models 15

6 Survival analysis 17

7 Likelihood-based inference 19

R 21

.1 Basics of R 21

Preliminary remarks

While we show how to implement statistical tests and models in SAS in class, these note will illustrate the concepts using R: visit the R-project website to download the program. The most popular graphical cross-platform front-end is RStudio Desktop.

Chapter 1

Introduction to statistical inference

Chapter 2

Linear regression

Chapter 3

Generalized linear models

Chapter 4

Correlated and longitudinal data

Chapter 5

Linear mixed models

Chapter 6

Survival analysis

Chapter 7

Likelihood-based inference

R

R is an object-oriented interpreted language. It differs from usual programming languages in that it is designed for interactive analyses.

You can find several introductions to R online. Have a look at the R manuals or better at contributed manuals. A nice official reference is An introduction to R. You may wish to look up the following chapters of the R language definition (Evaluation of expressions and part of the Objects chapter).

If you favor online courses, Data Camp offers a free introduction to R.

.1 Basics of R

.1.1 Help

Help can be accessed via `help` or simply `?`, e.g., `help("Normal")`. See R page about help files.

.1.2 Basic commands

Basic R commands are fairly intuitive, especially if you want to use R as a calculator. Elementary functions such as `sum`, `min`, `max`, `sqrt`, `log`, `exp`, etc., are self-explanatory.

Some (unconventional) features of the language:

- R is case sensitive.
- Use `<-` for assignments to a variable, and `=` for matching arguments inside functions
- Indexing in R starts at 1, not 0.
- Most functions in R are vectorized and loops are typically inefficient.
- Integers are obtained by appending `L` to the number, so `2L` is an integer and `2` a double (`numerical`).

Besides integers and doubles, the common types are

- logical (`TRUE` and `FALSE`);
- null pointers (`NULL`), which can be assigned to arguments;
- missing values, namely `NA` or `NaN`. These can also be obtained a result of invalid mathematical operations such as `log(-2)`.

Beware! In R, invalid calls will often returns something rather than an error. It is therefore good practice to check that the output is sensical.

.1.3 Linear algebra in R

R is an object oriented language, and the basic elements in R are (column) vector. Below is a glossary with some useful commands for performing basic manipulation of vectors and matrix operations:

- `c` concatenates elements to form a vector
- `cbind` (`rbind`) binds column (row) vectors
- `matrix` and `vector` are constructors
- `diag` creates a diagonal matrix (by default with ones)
- `t` is the function for transpose
- `rep` creates a vector of duplicates, `seq` a sequence. For integers i, j with $i < j$, `i:j` generates the sequence $i, i+1, \ldots, j-1, j$.

Subsetting is fairly intuitive and general; you can use vectors, logical statements. For example, if `x` is a vector, then

- `x[2]` returns the second element
- `x[-2]` returns all but the second element
- `x[1:5]` returns the first five elements
- `x[(length(x) - 5):length(x)]` returns the last five elements
- `x[c(1, 2, 4)]` returns the first, second and fourth element
- `x[x > 3]` return any element greater than 3. Possibly an empty vector of length zero!
- `x[ x < -2 | x > 2]` multiple logical conditions.
- `which(x == max(x))` index of elements satisfying a logical condition.

For a matrix `x`, subsetting now involves dimensions: `[1,2]` returns the element in the first row, second column. `x[,2]` will return all of the rows, but only the second column. For lists, you can use `[[` for subsetting by index or the `$` sign by names.

.1.4 Packages

The great strength of R comes from its contributed libraries (called packages), which contain functions and datasets provided by third parties. Some of these (`base`, `stats`, `graphics`, etc.) are loaded by default whenever you open a session.

To install a package from CRAN, use `install.packages("package")`, replacing `package` by the package name. Once installed, packages can be loaded using `library(package)`; all the functions in `package` will be available in the environment.

– There are drawbacks to loading packages: if an object with the same name from another package is already present in your environment, it will be hidden. Use the double-colon operator `::` to access a single object from an installed package (`package::object`).

.1.5 Datasets

- datasets are typically stored inside a `data.frame`, a matrix-like object whose columns contain the variables and the rows the observation vectors.

- The columns can be of different types (`integer`, `double`, `logical`, `character`), but all the column vectors must be of the same length.
- Variable names can be displayed by using `names(faithful)`.
- Individual columns can be accessed using the column name using the `$` operator. For example, `faithful$eruptions` will return the first column of the `faithful` dataset.
- To load a dataset from an (installed) R package, use the command `data` with the name of the `package` as an argument (must be a string). The package `datasets` is loaded by default whenever you open R, so these are always in the search path.

The following functions can be useful to get a quick glimpse of the data:

- `summary` provides descriptive statistics for the variable.
- `str` provides the first few elements with each variable, along with the dimension
- `head` (`tail`) prints the first (last) n lines of the object to the console (default is $n = 6$).

We start by loading a dataset of the Old Faithful Geyser of Yellowstone National park and looking at its entries.

```r
# Load Old faithful dataset
data(faithful, package = "datasets")
# Query the database for documentation
?faithful
# look at first entries
head(faithful)
```

```
##   eruptions waiting
## 1       3.6      79
## 2       1.8      54
## 3       3.3      74
## 4       2.3      62
## 5       4.5      85
## 6       2.9      55
```

```r
str(faithful)
```

```
## 'data.frame':    272 obs. of  2 variables:
##  $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
##  $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

```r
# What kind of object is faithful?
class(faithful)
```

```
## [1] "data.frame"
```

Other common classes of objects:

- `matrix`: an object with attributes `dim`, `ncol` and `nrow` in addition to `length`, which gives the total number of elements.

- `array`: a higher dimensional extension of `matrix` with arguments `dim` and `dimnames`.
- `list`: an unstructured class whose elements are accessed using double indexing `[[ ]]` and elements are typically accessed using `$` symbol with names. To delete an element from a list, assign `NULL` to it.
- `data.frame` is a special type of list where all the elements are vectors of potentially different type, but of the same length.

.1.6 Graphics

The `faithful` dataset consists of two variables: the regressand `waiting` and the regressor `eruptions`. One could postulate that the waiting time between eruptions will be smaller if the eruption time is small, since pressure needs to build up for the eruption to happen. We can look at the data to see if there is a linear relationship between the variables.

An image is worth a thousand words and in statistics, visualization is crucial. Scatterplots are produced using the function `plot`. You can control the graphic console options using `par` — see `?plot` and `?par` for a description of the basic and advanced options available.

Once `plot` has been called, you can add additional observations as points (lines) to the graph using `point` (`lines`) in place of `plot`. If you want to add a line (horizontal, vertical, or with known intercept and slope), use the function `abline`.

Other functions worth mentioning for simple graphics:

- `boxplot` creates a box-and-whiskers plot
- `hist` creates an histogram, either on frequency or probability scale (option `freq = FALSE`). `breaks` control the number of bins. `rug` adds lines below the graph indicating the value of the observations.
- `pairs` creates a matrix of scatterplots, akin to `plot` for data frame objects.

– There are two options for basic graphics: the base graphics package and the package `ggplot2`. The latter is a more recent proposal that builds on a modular approach and is more easily customizable — I suggest you stick to either and `ggplot2` is a good option if you don't know R already, as the learning curve will be about the same. Even if the display from `ggplot2` is nicer, this is no excuse for not making proper graphics. Always label the axis and include measurement units!