INTRODUCTION TO PYTHON & R

DAY 1 (29 Sept.): Introduction to R

paul j. van staden (PhD)
Department of Statistics
University of Pretoria

Basics of R

- R software is a free open-source programming language for statistical computing, data analysis & visualisation.
- Although one can directly use R software by typing code in the R console, it is preferable to use an IDE (Integrated Development Environment) for R such as RStudio.

Basic mathematical operators & functions

Operator or function		R code
Addition	$7 + 13$	`7 + 13`
Subtraction	$7 - 13$	`7 - 13`
Multiplication	7×13	`7 * 13`
Division	$7 \div 13$	`7 / 13`
Exponentiation	7^{13}	`7 ^ 13`
Square root function	$\sqrt{7}$	`sqrt(7)`
Logarithmic function	$\ln(13)$	`log(13)`

Variables

- A value in R can be stored in a variable to be used again in further calculations & analyses.

- The variable's name can be a single character, say `x` or `y`, or can be more descriptive, for instance `age` & `gender`.

- Although you may technically use built-in R constants or functions as variable names, it is definitely not recommended.

- Also, avoid variable names that are too long.

- R is case sensitive: `y` & `Y` will be two different variables.

EXERCISE: Circumference & area of a circle

- **The formulae for the circumference & area of a circle with radius r are:**

$$c = 2\pi r \quad \& \quad a = \pi r^2$$

- **Calculate the circumference & area for a circle with $r = 5$.**
  - **Assign the value of the radius to a variable named `radius`.**
  - **Assign the calculated values of the circumference & area to variables named `circ` & `area`.**

Data types

- The 5 most commonly used data types in R are:

Numeric:	Values are numbers or contain decimals.
Integer:	Numeric data without decimals.
Character:	Text strings.
Factor:	Categorical data with limited levels.
Logical:	Boolean values, `TRUE` or `FALSE`.

Comparison & logical operators

	R code
x is less than y	`x < y`
x is greater than y	`x > y`
x is less than or equal to y	`x <= y`
x is greater than or equal to y	`x >= y`
x is equal to y	`x == y`
x is not equal to y	`x != y`
x AND y	`x & y`
x OR y	`x | y`

Functions

- **There are numerous built-in functions in R, for example:**

Mathematics:	`sqrt()`	`log()`	`abs()`
Statistics:	`mean()`	`sd()`	`median()`
Graphics:	`plot()`	`hist()`	`barplot()`
Creation & manipulation:	`c()`	`seq()`	`subset()`
Exploration:	`View()`	`str()`	`class()`

- **Users can also create their own functions in R.**

EXAMPLE: Function to calculate the area of a circle

- The R code below creates a function called `area()` to calculate the area of a circle using the argument `radius`:

```
area <- function(radius){
f <- pi * (radius ^ 2)
return(f)
}
```

- The area is then calculated by specifying a value for the argument `radius` into the function `area()`:

```
area(5)
```

Packages

- R packages are bundled collections of resources, including functions, sample datasets and compiled code, which are stored in libraries.

- System Libraries in R contain the packages that are installed by default together with R, for instance the `base`, `datasets`, `graphics`, `stats` & `utils` packages.

- Users may install additional libraries from repositories such as CRAN, which will appear under the User Library in R.

Sequences & concatenation

- The colon operator, `:`, creates simple integer sequences with an increment of one.
- The `seq()` function gives more flexibility by letting you specify the start value, the end value, and the step size.
- The `rep()` function is useful for creating sequences with repeating patterns.
- The `c()` function is used to combine elements into data structures such as vectors.

Conditional statements

- The `if` statement is used for execution of code only when the specified condition is `TRUE`.

- An `else` statement can be used in conjunction with the `if` statement to provide alternative code to execute when the condition is `FALSE`.

EXAMPLE: Determine whether a person is a teenager

- The following R code verifies whether a person is a teenager based on the value assigned to the variable `age`:

```
age <- 13
if(age > 12 & age < 20){
  print("Person is a teenager")
}
```

- The R code below assigns `TRUE` or `FALSE` to the variable `teen` based on the value of `age`:

```
age <- 7
if(age > 12 & age < 20){
  teen <- TRUE
} else{
  teen <- FALSE
}
```

Loops

- Loops are used in R to repeatedly execute a block of code.
- The `for` loop is used to repeat a block of code for each element in a sequence
- With the `while` loop, a block of code is repeatedly executed as long as a specified condition remains `TRUE`.
- The `repeat` loop executes a block of code until a `break` statement is encountered within the loop.

EXAMPLE: Gauss summation

- The `for` loop can be used to calculate

$$\sum_{j=1}^{100} j$$

```
n <- 100
sum <- 0
for(j in 1:n){
  sum <- sum + j
}
```