An overview of R

MMU
MULTIMEDIA UNIVERSITY

Background

- First developed by Ross Ihaka and Robert Gentleman in 1990s at the Department of Statistics, University of Auckland in New Zealand.
- Based on S that developed at Bell Labs in the 1970s.
- Development shifted to a larger core group in 1997.
- Distributed under the GNU General Public License.

R Working Directory

- R working directory is where R loads, stores data and executes program.
- To view the current working directory: getwd()
- To set the working directory: setwd("path")

Some useful tidbits

- The console windows issue a ">" for user to input command.
- If the input command is not complete at the end of the line, R will issue a "+".
- The entered commands in the console window can be recalled by pressing the up and down arrow keys.
- Command are separated with semicolon (";") or by a newline (the same in R editor).
- The hash tag is used for comments (the same in R editor).

```
# This is an R comment
```

- The question mark will display help text.

```
? sin
```

Basic Data Types

- logical - TRUE / FALSE
- numeric
- characters (string)

Creating Variables

- The left arrow operator assigns values to variables.

```
a <- 30
b <- “This is a R string”
c <- 1<2
```

- The equal sign can also be used for assignment.

```
a = 30
```

- After doing this in R, a has the value of 30.

```
a
[1] 30
a == 30
[1] TRUE
```

Names

- Object names cannot contain symbols like !, +, -, #
- A dot (.) and an underscore (_) are allowed, a name can start with a dot.
- Object names can contain a number but cannot start with a number.
- R is case sensitive, X and x are two different objects, as well as temp and temP.

Operators

arithmetic	+ – * / %% ^
relational	> >= < <= == !=
logical	! & |

Basic Data Structures

- Vectors
  - A one-dimensional array that only hold one type of data.
  - Can be created in different ways
    :, c(), seq(), rep()
  - Take note on recycling for operations on vector.
- Matrices
  - A two-dimensional array for the same data type
  - Functions that are commonly used to create matrices.
    rbind(), cbind(), matrix()
- Use [ ] to refer elements of a vector/matrix.

Data Frames

- R refers to datasets as data frames.
- A data frame is a matrix-like structure, where the columns can be of different data types.
- Data frames can be created using data.frame().
- R has many built-in dataset, to load built-in data use data()

  data("mtcars")
- Dataset stored in Excel or text file can be imported into R with

  read.csv(), read.xlsx(), read.table()

R Packages

- R packages refers to a collection of previously programmed functions, often including functions for specific tasks.
- R currently has thousands of packages.
- To install R package: `install.packages("package name")`
- The above command saves the files to your machine. To use the package in an R session: `library(package name)`

If-Then-Else statements

- If-then-else statements make it possible to choose between two expressions depending on the value of a (logical) condition.

  if (*condition*) *expr1* else *expr2*

  ```
  if (x > 0) y = sqrt(x) else y = -sqrt(-x)
  y = if (x > 0) sqrt(x) else -sqrt(-x)
  y = ifelse(x > 0, sqrt(x), -sqrt(-x))
  ```

If-Then-Else statements

- Syntax of if-then-else statements for compound expressions:

```
if (condition) {
    expr1
    expr2
    expr3
} else {
    expr4
    expr5
}
```

For ... loop

- As part of a computing task we often want to repeatedly carry out some computation for each element of a vector.

- In R, one way is using a for loop.

for(*variable* in vector) *expr*

```
x = 1:15
s = 0
for(i in 1:length(x)) s = s + x[i]
s
```

For ... loop

- Sometimes, when given conditions are met, it is useful to be able to skip to the end of a loop, without carrying out the intervening statements. This can be done by executing a next statement when the conditions are met.

```
for(variable in vector) {
        expr1
        expr2
        if (condition)
                next
        expr3
        expr4
}
```

While ... loop

- For-loop evaluates an expression based on fixed number of times. Sometimes it is useful to repeat a calculation until a particular condition is false. A while-loop provides this form of control flow.

  while (*condition*) *expression*

```
threshold = 100
n = 0
s = 0
while (s <= threshold) {
    n = n + 1
    s = s + n
}
```

Function

- Many things in R are done using function calls
  getwd(), log(), plot()

- Additional functionality is added to R by defining new functions.

  function name=function (*arglist*) *expr1*
  function name=function(*arglist*) {
  expr1
  expr2
  }

arglist is a comma separated list of variable names known as the formal arguments of the function.

Function

- Functions are usually, but not always, assigned a name so that they can be used in later expressions.

```
factorial=function(n) {
    f=1
    for(i in 1:n) f=f*i
    f
}
```

Function

- A defined function can be used in other function definitions.

```
combination=function(n,k){
    factorial(n)/(factorial(k)*factorial(n - k))
}
```

Function

- Once we have finished writing a function, we can save the code to a file and then use the source() function to read the contents of the file when the function is needed.
- Using source() is similar to loading a package with library().

Importing Data from Files

- To import text file, read.table()

  read.table(file, header = FALSE, sep = "", skip, as.is, …)

  file: the name of the text file to import

  header: logical, does the first row contain column labels

  sep: field separator character

  sep=" " space (default)

  sep="\t" tab-delimited

  sep="," comma-separated

  skip Number of lines to skip before reading data

- To import .csv file, read.csv()

Importing Excel Files

- To read an .xlsx file, we need to install and load the xlsx package.
- Must also have a Java JRE installed

```
install.packages("xlsx")
install.packages("rJava")
library(xlsx)
```

```
read.xlsx(file, sheetIndex, header=TRUE, ...)
```

file: the name of the text file to import

header: logical, does the first row contain column labels

sheetIndex: a number representing the sheet to import

Data Frames

- After all files loaded into R, it is treated as a data frame.
- We can then use all R functions that are applicable to data frames.

  [ ], $, attach(), detach(), subset(), transform(), within(),.....

Exporting Data Frames

To export data frames to text file or csv file:

```
write.table(x, file = "filename.txt", ...)

write.csv(x, file = "filename.csv", ...)
```

x: The object to be written.

Numerical Variables

- Functions for calculating summary statistics of numerical variables.

mean(x), median(x)	Mean, median of x
var(x), sd(x)	Variance, standard deviation of x
min(x), max(x), range(x)	Minimum, maximum of x
quantile(x)	Quantiles of x for the given probabilities
summary()	Summary statistics of each column in a data frame
cov(x,y), cor(x,y)	Covariance, correlation of x and y

The Base Plotting System

- Loaded automatically in a standard installation of R.

- Some common functions for producing plots

R function	Description
pie()	Pie chart
hist()	Histogram
stem()	Stem-and-leaf plot
boxplot()	Boxplot
barplot()	Barplot
plot()	Scatterplot
pairs()	Scatterplot matrix