UNIVERSITY OF COLOMBO SCHOOL OF COMPUTING

SCS2211 - Laboratory II

Practical 1

Instructions

- Try out the commands discussed in the practical and take screenshots of the output.
- Create a report.
- Report must be in **PDF** format.
- Report name should be <Index Number>.pdf
    - **eg – 18000000.pdf**
- Any form of plagiarism or collusion is not allowed.

1. Start R and run the following commands.
    - > 2+2
    - > exp(-2) #call the exponential function and pass the value -2
    - > log(100, base=10) #call the log function. Notice that this function takes 2 arguments, the value and the base. (log10 100 = 2)
    - > runif(10) #generate 10 random numbers in the range [0,1]. Run this command twice and compare the results
2. Variables
    - Try the following;
    - > x = 2
    - > x + x #the answer is 4
    - > y = x + 3
    - > y #the answer is 5
    - > s="this is a char str"
    - > s
3. Create a vector

- Create a vector using the c() construct

```
> weight = c(60,70,86,97,45,67)
> weight
[1] 60 70 86 97 45 67
```

- Plot the values of weight vector,
- > plot(weight)
- Create a vector of regularly spaced numbers,
- > seq(0,1, length = 11)

```
> seq(0,1, length = 11)
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
```

- Also try
- > seq (4, 10, 0.5)
- > seq (length=10)
- You would have noticed that the same function (e.g. seq) can be used in many ways. And the output depends on how we call the function.
- > help(seq) #This will open the help file for the function seq(). It will explain typical usages and the arguments.
- The c() function can be used to combine vectors as well as scalars

```
> x=seq(10)
> c(x,1:10,100)
 [1]   1   2   3   4   5   6   7   8   9  10   1   2   3   4   5   6   7   8   9  10 100
```

- Common arithmetic operations (including +, -, *, /, ^) and mathematical functions (e.g. sin(),cos(), log()) work element-wise on vectors, producing another vector

```
> a=c(1,2,3,4,5,6,7)
> a^2
[1]  1  4  9 16 25 36 49
```

4. Summaries

- Many functions summarize a data vector by producing a scalar from a vector.

```
> sum(a)
[1] 28
> length(a)
[1] 7
```

- Simple summary statistics (mean, median, s.d., variance) can be computed from numeric vectors using appropriately named functions

```
> x=rnorm(100)
> x
  [1]  0.563435076 -2.082429002  1.209769033 -0.187011523 -1.465571655  0.5
 [21]  1.710293526 -1.204407353 -0.185805150  1.170403935 -0.496001207  2.0
 [41] -1.256443743 -0.016674572  0.664074923  1.401153620  1.202129365  0.8
 [61]  0.709347947 -0.811596385  0.507797175  0.374891053  0.390989165  0.6
 [81] -1.289735598 -1.112787165 -0.139899752  0.584979667  2.083280116  1.4
> mean(x)
[1] 0.1119943
> sd(x)
[1] 1.034006
> var(x)
[1] 1.069167
> median(x)
[1] -0.0003030917
```

- Quantiles can be computed using the quantile() function.
- IQR() computes the Inter-quartile range (midspread or middle fifty).

```
> quantile(x)
           0%           25%           50%           75%          100%
-2.6507749756 -0.5060044006 -0.0003030917  0.7336578229  2.3648533090
> IQR(x)
[1] 1.239662
```

- The five-number summary (minimum, maximum, and quartiles) is given by fivenum(). A slightly extended summary is given by summary().

```
> fivenum(x)
[1] -2.6507749756 -0.5109344443 -0.0003030917  0.7361951337  2.3648533090
> summary(x)
      Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
-2.6507750 -0.5060044 -0.0003031  0.1119943  0.7336578  2.3648533
```

5. Simple File reading

- First check your working directory
- > getwd() #gives the current working directory
- > setwd() #sets the working directory or do this using the menu option File -> Change dir

```
> getwd()
[1] "/Users/kavinda"
> setwd("/Users/kavinda/Documents/UCSC/2021/2nd Sem/SCS2211 Lab 2/Practicals/Practical 1")
> getwd()
[1] "/Users/kavinda/Documents/UCSC/2021/2nd Sem/SCS2211 Lab 2/Practicals/Practical 1"
```

- > list.files() #lists the files in the current working directory
- Download the file "d1.txt" from the LMS and save it in your current working directory. This is a file with two columns of data (V1 and V2) and 500 rows.
- > d1 = read.table("d1.txt") #reads the data.
- Check the summary statistics
- > summary(d1) #you will get summaries for both columns separately
- Draw a simple scatter plot
- > plot(d1) #this will plot one column against the other
- If you want to get the values of the first column (V1)
- > col1 = d1[1] #col1 will be in list format, not a vector
- Convert this to a vector
- > v1 = as.numeric(unlist(col1))
- Create a histogram
- > hist(v1)
- > hist(v1, 5) #notice the difference
- > hist(v1, 100)

6. Scripts
   - A script is just a plain text file with R commands in it.
   - Type the following in a file and save it as "myscript.txt"

myscript.txt

```
# A comment: this is a sample script.
y=c(12,15,28,17,18)
x=c(22,39,50,25,18)
mean(y)
mean(x)
plot(x,y)
```

- To run the script;
- > source("myscript.txt")

7. R Data Types

There are many data types in R. The following note is a simple explanation of data types available in R.

7.1 Numeric

these are generally positive and negative numbers with the decimal point

- 10
- -2
- 0.02
- 1.5e2 = 1.5 x 102 = 150
- 1e-7 =1 x 10-7 = 0.0000001

can be also hexadecimal (starting with ‘0x’ or ‘0X’ followed by zero or more digits, and ‘a-f’ or ‘A-F’)

- 0XF = 15
- 0XFA = 15*16^1 + 10*16^0 = 250
- Hexadecimal floating point constants are supported using C99 syntax, e.g.
- 0x1.1p1

7.2 Integer

- created by using the qualifier L (e.g. : 123L)
- can be used with (non-complex) numbers given by hexadecimal or scientific notation
  - Valid integer constants: try 1L, 0x10L, 1000000L, 1e6L
  - However, if the value is not a valid integer, a warning is emitted and the numeric value created. (try 1.1L, 1e-3L, 0x1.1p-2)
- Try the following;
  - > typeof(2) #this is a “double” value
  - > typeof(2L) #this is an “integer” value

7.3 Logical

- either TRUE or FALSE

7.4 Complex

- A numeric constant immediately followed by i is regarded as an imaginary complex number. (e.g. 2i, 2+4.1i, 1e-2i)

7.5 String

- Delimited by a pair of single ('') or double ('"') quotes and can contain all other printable characters. Quotes and other special characters within strings are specified using escape sequences (e.g.: \n):
- Try the following;
  - > name = "Anne"
  - > name
  - > name = "Anne Mary"
  - > name

7.6 Special Types

In addition, there are four special constants,

- NULL : used to indicate the empty object
- NA : for absent ("Not Available") data values
- Inf : denotes infinity
- NaN: is not-a-number
- E.g..: Try the following; 1/0, 0/1, 0/0, -2/0

8. Logical Comparisons

Less than	<	Less than or equal to	<=
Greater than	>	Greater than or equal to	>=

Equals	==	Not equal	!=

AND	&	NOT	!
OR	|		

- All the usual logical comparisons are possible:
- Element-wise boolean operations are also possible.
- Try the following:
  - 2 > 1
  - 4 <= 3
  - “Mary” == “Mary”
  - “Monday” == “monday”
  - 2 == 2L
  - identical(2, 2L)
  - !(TRUE)
  - !(2>1)
  - (2>1) & (5<10)
  - (“A”==“A”) | (“B”==”b”)

9. Saving objects
   - R has the ability to save objects, to be loaded again later.
   - Whenever exiting, R asks to save all the variables created by the user, and restores
   - them when starting up the next time (in the same directory).
   - This is actually a special case of a very powerful feature of R called serialization.
   - All R objects, however complex, can be saved as a file on disk, and re-read in a later session.
   - See ?save and ?load for details.
10. How to write a function
    - Defined using the construct expression function(arglist)

- The args() function, gives the arguments that a function accepts (along with their default values)

```
> sumproduct = function(a=1,b=2,c)
+ + return(list(sum=a+b+c,product=a*b*c))
> sumproduct(1,2,3)
$sum
[1] 6

$product
[1] 6
```

- Try > sumproduct gives the full definition of the function

```
> sumproduct
function(a=1,b=2,c)
+ return(list(sum=a+b+c,product=a*b*c))
> args(sumproduct)
function (a = 1, b = 2, c)
NULL
```

11. Packages
    - Each package is a collection of functions (and data) with a common theme;
    - The core of R itself is a package called base.
    - A collection of packages is called a library.
    - Some packages are already attached when R starts up.
    - Other packages need to be attached using the library() function.
    - More from the Comprehensive R Archive Network (CRAN) at http://cran.fhcrc.org/web/packages/
    - To see the pre-installed packages

```
> installed.packages()
```

```
> installed.packages()
           Package      LibPath                                                    Version   Priority      Depends
KernSmooth "KernSmooth" "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "2.23-17" "recommended" "R (>= 2.5.0), stats"
MASS       "MASS"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "7.3-53"  "recommended" "R (>= 3.1.0), grDevices, graphics, stats, u
Matrix     "Matrix"     "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "1.2-18"  "recommended" "R (>= 3.2.0)"
base       "base"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
boot       "boot"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "1.3-25"  "recommended" "R (>= 3.0.0), graphics, stats"
class      "class"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "7.3-17"  "recommended" "R (>= 3.0.0), stats, utils"
cluster    "cluster"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "2.1.0"   "recommended" "R (>= 3.3.0)"
codetools  "codetools"  "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "0.2-16"  "recommended" "R (>= 2.1)"
compiler   "compiler"   "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
datasets   "datasets"   "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
foreign    "foreign"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "0.8-80"  "recommended" "R (>= 4.0.0)"
grDevices  "grDevices"  "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
graphics   "graphics"   "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
grid       "grid"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
lattice    "lattice"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "0.20-41" "recommended" "R (>= 3.0.0)"
methods    "methods"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
mgcv       "mgcv"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "1.8-33"  "recommended" "R (>= 2.14.0), nlme (>= 3.1-64)"
nlme       "nlme"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "3.1-149" "recommended" "R (>= 3.4.0)"
nnet       "nnet"       "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "7.3-14"  "recommended" "R (>= 3.0.0), stats, utils"
parallel   "parallel"   "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
rpart      "rpart"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.1-15"  "recommended" "R (>= 2.15.0), graphics, stats, grDevices"
spatial    "spatial"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "7.3-12"  "recommended" "R (>= 3.0.0), graphics, stats, utils"
splines    "splines"    "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
stats      "stats"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
stats4     "stats4"     "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
survival   "survival"   "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "3.2-7"   "recommended" "R (>= 3.4.0)"
tcltk      "tcltk"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
tools      "tools"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
utils      "utils"      "/Library/Frameworks/R.framework/Versions/4.0/Resources/library" "4.0.3"   "base"        NA
           Suggests                                  Enhances                                License                     License_is_FOSS License_restricts
KernSmooth "MASS"                                    NA                                      "Unlimited"                 NA              NA
MASS       "lattice, nlme, nnet, survival"           NA                                      "GPL-2 | GPL-3"             NA              NA
Matrix     "expm, MASS"                              "MatrixModels, graph, SparseM, sfsmisc" "GPL (>= 2) | file LICENCE" NA              NA
base       "methods"                                 NA                                      "Part of R 4.0.3"           NA              NA
boot       "MASS, survival"                          NA                                      "Unlimited"                 NA              NA
class      NA                                        NA                                      "GPL-2 | GPL-3"             NA              NA
cluster    "MASS, Matrix"                            NA                                      "GPL (>= 2)"                NA              NA
codetools  NA                                        NA                                      "GPL"                       NA              NA
compiler   NA                                        NA                                      "Part of R 4.0.3"           NA              NA
datasets   NA                                        NA                                      "Part of R 4.0.3"           NA              NA
foreign    NA                                        NA                                      "GPL (>= 2)"                NA              NA
```

- Some packages are already attached (you have to attach a package before using it)
- when R starts up. Use search() function to see the already attached packages
- To attach, use library() function. (e.g.: > library(class))
- To download and install, use install.packages()

12. Getting Help
    - help.start() starts a browser window with an HTML help interface.
    - help(topic) or ?topic displays the help page for a particular topic. Every R
    - function has a help page.
    - help.search("search string") or ??”search string” performs a
    - subject/keyword search.
    - To directly run the examples given in help pages, use the example() function

- The apropos() function, lists all functions (or other variables) whose name matches a specified character string.