

R Tutorial

This tutorial will give you a brief introduction to the R-environment. You will be provided with precise R-commands to enter and the corresponding output it should yield (if any).

The command prompt for the R-environment is a “greater than” symbol (>). Following a command, comments are often included to indicate the function of the code. Comments are set off by a pound symbol (#). In the exercises that follow, R-code will be written in boldface type with the prompt included as follows:

```
> This is an example of R-code typeset      #and this is the corresponding comment
```

When appropriate, the resulting output that should appear in R will also be included in the exercise with the following typeset:

```
[1] This is what the output will look like.
```

1. To assign a name to an object, you can use an equal sign (=), a left-arrow (<-), or a right-arrow (->).

```
(a) > a = 15      #This assigns to the object 'a' the value 15  
    > a          #To look at the object, enter the name  
    [1] 15
```

```
(b) > b <- 3+7     #Here we include simple arithmetic in the assignment  
    > b  
    [1] 10
```

```
(c) > “Colorado” -> c    #Of course, not all data is numerical  
    > c  
    [1] ‘Colorado’
```

2. To create a list (or array) of data values, you can use the c command. Basic functions can then be applied to the array.

```
(a) > x = c(1, 3, 3, 7)    #The object x is a list of 4 data values  
    > x  
    [1] 1 3 3 7
```

```
(b) > x = c(x, 2, 5)     #And two more data values have been added to x  
    > x  
    [1] 1 3 3 7 2 5
```

```
(c) > length(x)         #This command computes the number of elements in x.  
    [1] 6
```

```
(d) > x[5]              #We can look at just the 5th element in the list.  
    [1] 2
```

```
(e) > mean(x)           #Some statistics can be computed.  
    [1] 3.5
```

```
(f) > xave = mean(x)     #The value of the mean of the list x is assigned the name 'xave'  
    > xave               #Now we must call the object name to see the output value.  
    [1] 3.5
```

3. To save our progress (to return later or to submit for grading), all the objects in the workspace need to be saved as is. It is strongly recommended that you attempt to save this tutorial workspace now, quit R, reopen R, and reload the saved workspace.

- R: In the R-environment
 - Mac OS X: There is a “Workspace” menu which includes a “Save Workspace File...” option. Use this to save your workspace now. Once saved, quit R and restart the application. Now load that saved workspace by using the “Load Workspace File...” option under the “Workspace” menu.
 - Windows: There is a “Save Workspace As...” option under the “File” menu. Use this to save your workspace now. Once saved, quit R and restart the application. Now load that saved workspace by using the “Load Workspace” option under the “File” menu.

Once saved, quit R and restart the application. Now load that saved workspace by using the “Load Workspace File...” option under the “Workspace” menu.

- R-Studio: In R-Studio, there is a “Session” menu which includes a “Save Workspace As...” option. Use this to save your workspace now. Once saved, quit R-Studio and restart the application. Now load that saved workspace by using the “Load Workspace...” option under the “Session” menu.

Check that you have reloaded the saved data correctly.

```
> length(x)
[1] 6
> b
[1] 10
```

4. When in the R-environment, the up- and down-arrow keys can be used to scroll through previously entered lines of code. This can be especially useful if you want to re-execute a command or realize that there is a typo and can then just fix the typo rather than retype the entire line of code.

- (a) The following code searches a list Y for the first occurrence of the data value 7 and returns the location of that first 7.

```
> Y = c(5, 2, 4, 4, 5, 8, 2, 4, 6, 7, 2, 1, 7, 2, 9, 4)      #We can see the first 7 is the
10th element.
> n = 1           #We will start the search with element number 1 in the list
> while(Y[n] != 7){n=n+1}      #We check each element of Y until the value equals 7.
> n               #We call for the value of n.
[1] 10
```

- (b) At this point having just run the code to find the first occurrence of the value 7, you realize that you really want the first occurrence of the value 8. Rather than retype all 5 lines, use the arrow keys to recall (and edit, as necessary).

```
Arrow up 3 times to > n=1      #Restart the search from the start of the list.
Arrow up 3 times to > while(Y[n] != 7){n=n+1}
#But use the back-arrow to get to the 7, backspace and type 8, then enter.
Arrow up 3 times to > n
[1] 6
```