Exercises and solutions for chapter 'R Language Fundamentals'

August 11, 2008

Exercise 1
What does the `typeof is.na` *mean? Why is it different from that of* `mean`*?*

Solutions: The function `is.na` is a `builtin` function, that means that it is a special type of function, that has been written to be very efficient.

Exercise 2
Find help for the colon operator; what does it do? What is the type of its return value? Use the predicate testing functions to determine the storage mode of the expressions `1:3` *and* `1.3:4.2`*.*

Solutions: The colon operator returns a sequence of values from the starting value to the end, in steps of size 1. If the starting value is an integer then the return value will be of type integer. If the starting value is double then the return is also of type double.

Exercise 3
Create vectors of each of the different primitive types. Create matrices and arrays by attaching `dim` *attributes to those vectors. Look up the help for* `dimnames` *and attach dimnames to a matrix with two rows and five columns.*

Exercise 4
What is the largest integer that can be represented on your computer? What happens if you add one to this number? What is the smallest negative integer that can be represented?

Solutions: You must be careful in answering the second question to add an integer 1, and then you get `NA` and a warning. If you add a real 1 then the maximum integer is coerced and 1 is added to the result, but the result is not an integer. The negative of the maximum integer seems to be the smallest negative integer, at least subtracting `1L` from it yields a warning.

```
> .Machine$integer.max
[1] 2147483647
> .Machine$integer.max + 1L
[1] NA
> -.Machine$integer.max
[1] -2147483647
> -.Machine$integer.max - 1L
[1] NA
```

Exercise 5
Create a list of length 4 and then add a `dim` *attribute to it. What happens?*

Solutions: We take the `dim` attribute off at the end, so that `y` behaves normally.

```
> tl = list(1:3, "a", 3:9, mean)
> dim(tl) = c(2, 2)
> tl
     [,1]      [,2]
[1,] Integer,3 Integer,7
[2,] "a"       ?
```

Exercise 6
Look up the help page for `data.frame` *and use the example code to create a small data frame.*

Exercise 7
What does `sessionInfo` *report? How do you interpret it?*

Solutions: One of the differences is that the order of the packages in the search path is not given while package versions are (and this is important if you ever need help). `sessionInfo` also reports functions that are loaded via a name space (this is described in more detail in Chapter **??**).

Exercise 8
Use the `seq` *function to generate a subscript vector that selects those elements of a vector that have even-numbered subscripts.*

Exercise 9
Use the function `seq` *to generate a sequence of indices so that those elements of a vector with odd-numbered indices can be excluded. Verify this on the built-in* `letters` *data. Verify the statement about zero length subscript vectors.*

Exercise 10
Verify that vectors can have duplicated names and that if a subscript matches a duplicated name, only the first value is returned. What happens with `x[NA]`*, and why does that not contradict the claims made here about* `NA` *subscripts? Hint: it might help to look back at Section* **??**.

Exercise 11
Use logical subscripts to extract the even-numbered elements of the `letters` *vector.*

Exercise 12
Let `x` *be a vector of length 10 and has a dimension attribute so that it is a matrix with 2 columns and 5 rows. What is the matrix location of the 7th element of* `x`*? That is, which row and column is it in? Alternatively, which element of* `x` *is in the second row, first column?*

Solutions:

```
> x = 11:20
> dim(x) = c(5, 2)
> x[7]
[1] 17
```

Which is in `x[2,2]`. The second element of `x` is in the second row, first column.

Exercise 13
Verify the claims made for character subsetting of matrices and data.frames.

Exercise 14
Verify the claims made about negative subscript assignments. Create a named vector, `x`*, and set one of the names to* `NA`*. What happens if you execute* `x[NA]=20` *and why does that not contradict the statements made above? What happens if you use* `x[as.character(NA)]=20`*?*

Solutions:

```
> x = 1:5
> names(x) = letters[1:5]
> names(x)[1] = NA
> x[NA] = 20
> x[as.character(NA)] = 20
```

The first subset assignment has no effect, because the subscript is a logical value, and logical subsetting applies. The second one, is more peculiar, as one might expect that the missing values match, but they do not, and yet another value is added to the end of `x`.

Exercise 15
Write a replacement function, called `rowrep`*, that replaces the indicated row of a matrix,* `x` *say, with the value on the right-hand side of the assignment. That is, we want a syntax like* `rowrep(x, 4) = c(11, 12)` *to replace the fourth row of* `x` *with the values 11 and 12.*

Solutions:

```
> "rowrep<-" = function(x, i, value) {
+     x[i, ] = value
+     x
+ }
```

Exercise 16
Write a function that takes a string as input and returns that string with a caret prepended. If you name your function `ppc` *then we want* `ppc("xx")` *to return* `"^xx"`*. You will probably find the function* `paste` *helpful.*

Solutions:

```
> ppc = function(x) paste("^", x, sep = "")
```

Exercise 17
Write an R function that takes a matrix as input and standardizes each row by subtracting the median and dividing by the MAD. You might want to look at the functions `mad` *and* `median`*. Next, generalize your function so that these are the default values, but allow the user to select other measures of the center and spread, should they want to use them.*

Exercise 18
Use `repeat`, `next` *and* `break` *to print the odd integers between 1 and 10. Repeat this exercise using* `while`, *instead of* `repeat`, *and print the even integers.*

Exercise 19
Explain the output in the preceding code chunk.

Exercise 20
Many implementations of `switch` *include a capability to return a default value if there is no match. Show that this can be done in R using* `switch` *and named arguments, but that for numbered arguments this is not possible.*

Solutions:
An unnamed value is used, for

```
> sw1 = function(x) {
+     switch(x, "10" = 11, "20" = 21, NA)
+ }
> sw1("10")
[1] 11
> sw1("x")
[1] NA
> sw1(10)
NULL
```

But with numbers, you cannot supply a default, you will get `NULL` if you address outside the length of the alternatives. Notice that character inputs are treated a bit peculiarly.

```
> sw2 = function(x) {
+     switch(x, 11, 21, NA)
+ }
> sw2(1)
[1] 11
> sw2(4)
NULL
> sw2("a")
[1] 11
```

Exercise 21
- *Produce a bitmap image of a plot. Which parameters must you set? Which parameters are optional?*

- *Use* `layout` *to create a scatterplot with histograms on the sides. Hint: see the manual page.*
- *Use* `dev.copy` *to copy this to a PDF device and then open the resulting PDF document using your favorite viewer.*
- *What does the graphics parameter* `cex` *do?*
- *Can you find the size of the figure? What units (e.g., pixels, inches, etc.) can this be obtained in?*