CS636 Lab 1

1. Install UsingR and ISwR packages to your own directory

2. Using R for Introductory Statistics

Page 18: **1.1-1.12** (see below)

1.1 Use R as you would a calculator to find numeric answers to the following:

1. $1+2(3+4)$
2. 4^3+3^{2+1}
3. $\sqrt{(4+3)(2+1)}$
4. $\left(\frac{1+2}{3+4}\right)^2$

1.2 Rewrite these R expressions as math expressions, using parentheses to show the order in which R performs the computations:

1. `2 + 3 - 4`
2. `2 + 3 * 4`
3. `2/3/4`
4. `2^3^4`

1.3 Use R to compute the following

$$\frac{1+2\cdot 3^4}{5/6-7}.$$

Copyrighted Material

1.4 Use R to compute the following

$$\frac{0.25-0.2}{\sqrt{0.2\cdot(1-0.2)/100}}.$$

1.5 Assign the numbers 2 through 5 to different variables, then use the variables to multiply all the values.

1.6 The `rivers` data set is loaded when R is. View the data by typing its name and then the return key. What is the last value listed?

1.7 The `exec.pay` (`UsingR`) data set is available from the command line after loading the package `UsingR`. Load the package, and inspect the data set. Scan the values to find the largest one.

1.8 For the `exec.pay` (`UsingR`) data set, apply the functions `mean`, `min`, and `max`. What are the values found?

1.9 The basic `mean` function has an additional argument `trim`. When given, the specified proportion of the data is trimmed from the sorted data before the mean is taken. Compare the difference between `mean(exec.pay)` and `mean(exec.pay, trim=0.10)`.

1.10 The `Orange` data set is stored as a data frame with three variables. What are the three variables?

1.11 Compute the average age of the trees in the `Orange` data set using `mean`.

1.12 Compute the largest circumference of the trees in the `Orange` data set.

1.18 Define x and y with

```
> x = c(1, 3, 5, 7, 9)
> y = c(2, 3, 5, 7, 11, 13)
```

Try to guess the results of these R commands:

1. x+1
2. y*2
3. length (x) and length (y)
4. x+y (recycling)
5. sum(x>5) and sum(x[x>5])
6. sum(x>5|x<3)
7. y[3]
8. y[−3]
9. y[x] (What is NA?)
10. y[y>=7]

Remember that you access entries in a vector with [].

1.19 Consider the following "inequalities." Can you determine how the comparisons are being done?

```
> "ABCDE" == "ABCDE"
[1] TRUE
> "ABCDE" < "ABCDEF"
[1] TRUE
> ”ABCDE" < "abcde"
[1] TRUE
> ”ZZZZZ" < "aaaaa"
[1] TRUE
> "11" < "8"
[1] TRUE
```