1. Download and install RStudio. Open it up. Play around. Try some syntax you know from Python. Does it work? If not, see if you can figure out how to do it in R. In particular:
2. Figure out how to get the length of a vector or list
3. Figure out the equivalent of range(a,b) from Python
4. Figure out how to initialize a vector of all 0s
5. Get used to working with vectorized objects
6. Make two vectors, v1 = c(3,6,7,3,1) and v2 = c(6,3,0,6,1)
7. What do you expect the outcome of v1 + v2, v1 \* v2, v1 – v2, and v1/v2 will be? Does something go wrong with one of them?
8. Try some other mathematical operations, like exponentiation, square root, log, etc. Do you get the kind of result you expect?
9. R is **very** good at doing matrix operations, like matrix multiplication. The operator for matrix multiplication is %\*% (yes, including the percent signs).
10. Reach way back, and try to remember matrix multiplication. Maybe look it up on Wikipedia. Predict what you should get if you do v1 %\*% v2. Now try it. Do you get it?
11. Figure out how to make a matrix in R. Create a matrix m with 2 rows and 2 columns, first row is 3, 6; second row is 7, 1. Also create a vector v = c(3,1).
12. What should you get for m%\*%v? How about v%\*%m? Do it and seeif you’re right!
13. Let’s learn how to data.table.
14. Go through the data.table vignette, <https://rawgit.com/wiki/Rdatatable/data.table/vignettes/datatable-intro.html>. Actually do the things. That will include actually downloading the flights14.csv that it links to.
15. As part of the last problem set, you computed FPKM for two different yeast experiments. Revisit that data, and (using Python) create a 4 column file (with column names): column 1 = gene\_name, column 2 = gene\_length, column 3 = BY\_expression (in COUNTS, not FPKM), column 4 = RM\_expression (in COUNTS, not FPKM)
16. Read the resulting table in R using fread.
17. Using data.table syntax, compute the log fold change of every gene in ONE LINE. NB: log fold-change means log2(BY\_expression/RM\_expression) i.e. how many factors of 2 you need to get from the RM expression level to the BY expression level.
18. In part d, you probably got a ton of NA values. Why? Using data.table syntax, filter out the “bad” rows, and redo part d. This should take ONE LINE
19. Another option to solve the NAs in part d is to have pseudo-counts. A pseudo-count is simply a phony observation of exactly one extra read per gene. Using data.table syntax, add pseudo-counts to every gene. This should take ONE line
20. Re-compute log-fold change using the pseudo-counted data. ONE line.
21. So far, we’ve been working with raw counts, not FPKM. Using ONE line, compute the FPKM values for every gene.
22. Finally, compute the log fold change of the FPKM values. How does it compare to the FPKM of the counts?