

## Problem set 3

**Problem 1.** Re-create exactly the following output:

```
## "The quick brown fox
##   jumped over
##     the lazy dogs"
```

**Problem 2.** Suppose you've stored the values `num1 <- 4` and `num2 <- 0.75`. Write a line of R code that returns the following string:

```
## [1] "The result of multiplying 4 by 0.75 is 3."
```

**Problem 3.** Store the string “Two 6-packs for \$12.99”. Then do the following:

- Use a check for equality to confirm that the substring beginning with character 5 and ending with character 10 is “6-pack”.
- Make it a better deal by changing the price to \$10.99.

**Problem 4.** Recreate exactly the following output using paste:

```
## [1] "Group 1" "Group 2" "Group 3" "Group 4" "Group 5" "Group 6"
## [7] "Group 7" "Group 8" "Group 9" "Group 10"
```

**Problem 5.** Create a list that contains, in this order, a sequence of 20 evenly spaced numbers between -4 and 4; a  $3 \times 3$  matrix of the logical vector `c(F,T,T,T,F,T,T,F,F)` filled column-wise; a character vector with the two strings “kella” and “fateh”; and a factor vector containing the observations `c("LOW", "MED", "LOW", "MED", "MED", "HIGH")`. Then, do the following:

- Give appropriate name to each element of the list.
- Extract row elements 2 and 1 of columns 2 and 3, in that order, of the logical matrix.
- Use `sub` to overwrite “fateh” with “Fateh” and “kella” with “Kella” inside the list. Then, using the newly overwritten list member, concatenate to the console screen the following statement exactly:

```
## "Found the killing weapon!"
```

```
## -\Kella Fateh/-
```

- Using `which`, determine which indexes in the factor vector are assigned the “MED” level.

**Problem 6.** Create and store this data frame as `dframe` in your R workspace:

person	sex	funny
Stan	M	High
Francine	F	Med
Steve	M	Low
Roger	M	High
Hayley	F	Med
Klaus	M	Med

Here the attribute `person` should be a character vector, `sex` should be a factor with levels F and M, and `funny` should be a factor with levels Low, Med, and High.

- Stan and Francine are 41 years old, Steve is 15, Hayley is 21, and Klaus is 60. Roger is extremely old - 1,600 years. Append these data as a new numeric column variable in `dframe` called `age`.

- b. Use your knowledge of reordering the column variables based on column index positions to overwrite dframe, bringing it in line with mydata. That is, the first column should be person, the second column age, the third column sex, and the fourth column funny.
- c. Now store the following data to a data frame named mydata2.

person	age	sex	funny
Peter	42	M	High
Lois	40	F	High
Meg	17	F	Low
Chris	14	M	Med
Stewie	1	M	High
Brian	7	M	Med

- d. Now, combine mydata2 with dframe, naming the resulting object mydataframe.
- e. Write a single line of code that will extract from mydataframe just the names and ages of any records where the individual is female and has a level of funniness equal to Med OR High.
- f. Use your knowledge of handling character strings in R to extract all records from mydataframe that correspond to people whose names start with S. (Hint: substr can be used. Use ?substr to see the help)

**Problem 7.** Store the following vector:

```
foo <- c(13563,-14156,-14319,16981,12921,11979,9568,8833,-12968, 8133)
```

Then do the following:

- a. Output all elements of foo that, when raised to a power of 75, are NOT infinite.
- b. Return the elements of foo, excluding those that result in negative infinity when raised to a power of 75.