1. Have you installed R on your computer? Do you have access to the command line via a shell? This will likely be Terminal on a Mac (which is built-in) or git bash on Windows.

R on the File System

Assume you start a new R session with your working directory at `~`. Which R functions would you call to do each of the following?

```
~
├── Project-1
│   ├── raw-data-1.txt
│   ├── processed-data-1.csv
│   └── data-processing-1.r
├── Project-2
│   ├── raw-data-2.txt
│   ├── processed-data-2.csv
│   └── data-processing-2.r
```

2. Change your working directory to be `Project-1`?

3. Read the data in `raw-data-1.txt` and save it to the R object named `raw_data_1`.

4. Without changing your working directory, read the data in `raw-data-2.txt` and save it to the R object named `raw_data_2`.

5. You may have noticed that when creating R objects, I avoid certain types of names. What characters are permissable when naming new objects?[1]

[1]You can research the answer to this question online.

Atomic Vectors

If you recorded data on peoples' answers to the following questions and read each one into R as a vector, what data type could be used to store them?

6. How much do you spend on average per week, in dollars and cents, on caffeinated beverages?

7. Did you vote in the last US Election?

8. How many US states have you visited?

9. Which US states have you visited?

While each of the following vectors *could* be created by simply using `c()`, instead write a command that can generate each one more efficiently using the functions from class. Read more about them by pulling up their help files with `?`.

10. `_________________________`

```
 [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48
```

11. `_________________________`

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

12. `_________________________`

```
[1] "A" "A" "B" "B" "B" "C" "C" "C" "C"
```

13. `_________________________`

```
 [1]  0  0  0  5  5  5 10 10 10 15 15 15
```

14. What single command can be run on each of these vectors

```r
x <- c("A", "B", "C")
x <- seq(1, 5, 1)
x <- c(TRUE, FALSE)
```

and produce this output?

```
[1] "A" "B" "B" "C" "C" "C"
```

```
 [1] 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5
```

```
[1]  TRUE FALSE FALSE
```

Working with Vectors

15. Predict the output of the following code.

```
1:3 + 1:2 + 1:1
```

16. Predict the output of the following code.

```
1:1 + 1:2 + 1:3
```

17. Which vector results from running the following code? Does it throw a message, warning, or error and why?

```
c(5, 5, 5) * c(2, 1, NULL)
```

18. Which vector results from running the following code? Does it throw a message, warning, or error and why?

```
c(5, 5, 5) * c(2, 1, NA)
```

19. Predict the output and type of the following three vectors.

```
c(1, FALSE)
c("a", 1)
c(TRUE, 1L)
```

20. Use square bracket subsetting to create the following vector from `z <- c(apple = 2, banana = 4, cherry = 8)`.

```
cherry banana  apple
     8      4      2
```

21. Write down *three* subsetting methods (index, name, and logical) to generate the following vector from `z`. If you're not able to do it in one subsetting operation, say why not.

```
cherry cherry
     8      8
```

22. What happens when you subset a vector with a negative number?

23. What happens when you subset a vector with square brackets with nothing in them?

Extending Vectors

24. Provide the code that will create a matrix object of type double with 3 rows and 4 columns. The entries should be the integers between 1 and 12 arrayed *rowwise*. Name the object `mat_twelve`.

25. Subset the matrix to isolate the element with the value `5`.

26. Provide the code that will break the `mat_twelve` into two separate matrices: `mat_left` with the 6 leftmost entries, and `mat_right` with the remaining six.

27. What does `mat_left + mat_right` evaluate to?

You can load the "like" matrix into R using the following code.

```r
install.packages("lda") # install lda package
library(lda)            # load lda package
data(sampson)           # load data
samplik <- sampson[["SAMPLK1"]] # extract the samplik matrix
```

28. Provide the code to extract the vector of the ranks that Basil gave other monks from 0 (unranked) to 3 (most liked).

29. Provide the code to extract the vector of the ranks that other monks gave Basil from 0 (unranked) to 3 (most liked).

30. Matrices make it easy and fast to apply the same function across all rows or columns. Use `rowSums()` or `colSum()` and the sort function to return the three most like monks and the three least liked / ranked monks.