If an error occurs, the easiest thing to do is to immediately call the traceback() function. This function returns the function call stack just before the error occurred so that you can see what level of function calls the error occurred. If you have many functions calling each other in succeeding, the traceback()output can be useful for identifying where to go digging first.

For example, the following code gives an error.

```
check_n_value <- function(n) {
        if(n > 0) {
                stop("n should be <= 0")
        }
}
error_if_n_is_greater_than_zero <- function(n){
        check_n_value(n)
        n
}
error_if_n_is_greater_than_zero(5)
Error in check_n_value(n): n should be <= 0
```

Running the traceback() function immediately after getting this error would give us

```
traceback()
3: stop("n should be <= 0") at #2
2: check_n_value(n) at #2
1: error_if_n_is_greater_than_zero(5)
```

From the traceback, we can see that the error occurred in the check_n_value() function. Put another way, the stop() function was called from within the check_n_value() function.

Mark as completed