Stats 102A - Homework 1 Instructions

Homework 1 Requirements

You will submit three files.

The files you submit will be:

1. `102a_hw_01_script_first_last.R` Your R script file containing the functions you write for the homework assignment. Write comments as necessary.
2. `102a_hw_01_output_first_last.Rmd` Take the provided R Markdown file and make the necessary edits so that it generates the requested output. The first line of your .Rmd file should be to source the R script file you wrote. You will also need to go through the rest of the file and insert your answers to any questions.
3. `102a_hw_01_output_first_last.pdf` Your output PDF file. This is the primary file that will be graded. Make sure all requested output is visible in the output file.

Failure to submit all 3 files will result in an automatic 40 point penalty.

Academic Integrity

At the top of your R markdown file, be sure to include the following statement after modifying it with your name.

"By including this statement, I, Joe Bruin, declare that all of the work in this assignment is my own original work. At no time did I look at the code of other students nor did I search for code solutions online. I understand that plagiarism on any single part of this assignment will result in a 0 for the entire assignment and that I will be referred to the dean of students."

If you collaborated verbally with other students, please also include the following line to credit them.

"I did discuss ideas related to the homework with Josephine Bruin for parts 2 and 3, with John Wooden for part 2, and with Gene Block for part 5. At no point did I show another student my code, nor did I look at another student's code."

Part 1

Write a function called `by_type()` that takes an atomic vector as input and an optional argument 'sort' with default value FALSE.

The function looks at each element in the atomic vector and sees if it can coerce them into integer or floating-point values. The function outputs a list that has separated the values into a list with integers, doubles, and character values. If the optional argument `sort` is `TRUE`, then it will sort the results of each vector.

For example, if x is the following vector,

```
x <- c("house", "6", "2.2", "a", "3.4", "1")
```

Then the output of `by_type(x)` will be a list:

```
$integers
[1] 6 1
```

```
$doubles
[1] 2.2 3.4

$character
[1] "house"     "a"
```

If `sort = TRUE` then the output will sort each section and return:

```
by_type(x, sort = TRUE)

$integers
[1] 1 6

$doubles
[1] 2.2 3.4

$character
[1] "a"     "house"
```

If the vector contains logical TRUE/FALSE values, those should be put into the character section.

Part 2

Write a function named `prime_factor(x)` that will find the prime factorization of an integer `x`. The function will output a numeric vector. All of the values in the output vector will be prime. The product of the numeric vector will be the original value x. There should be a check to make sure that the input value is a number greater than 2 with no decimal values with appropriate error messages.

If you're stuck on getting started with this, I suggest doing some prime factorizations yourself. For example, try to find the prime factors of 171 or 364. Note the steps you go through to find these prime factors and see if you can create code to do it.

(The point of the exercise is to give you practice writing code. Yes, I am aware that solutions to this problem already exist on the Internet. Don't search for them. While the problem is simple, it is complex enough that it is incredibly unlikely for students to create identical code solutions. You are free to *talk* about your approach and I fully expect many students to take identical approaches, but students writing identical code is an entirely different matter.)

Part 3

Write a function named `month_convert(x, from_lang, to_lang)` that will convert month names from one language to another. The function will accept three arguments: `x`, which is a factor with month information; the language from which we are tranlating `from_lang`; and the language to which we are translating `to_lang`.

For example:

```
x <- factor(c("March","March","February","June"))
month_convert(x, "English", "Spanish")
```

will output:

```
[1] marzo   marzo   febrero junio
Levels: febrero junio marzo
```

If the input contains a value that is not a real month, then it will be replaced with NA.

```
x <- factor(c("March","March","February","June","Jaly"))
month_convert(x, "English", "Spanish")
[1] marzo   marzo   febrero junio   <NA>
Levels: febrero junio marzo
```

I have uploaded a file called "month_names.txt" on CCLE which has the month names in several European languages. Use `month_names <- read.delim("month_names.txt", encoding="UTF-8", row.names=1)` to import the file.

Part 4

There are several questions in the Rmd file that you will need to answer with words.