Advanced R exercises

A 1) *Let's sort*

Difficulty: Easy

Sorting is easy in R. You can just use `sort(x)` . Let's try to build your own sort function. One straightforward algorithm is called bubble-sort, check the 'Bubblesort' entry on Wikipedia for more information (https://en.wikipedia.org/wiki/Bubble_sort).

- ☐ Create a function that can sort a vector using the bubble-sort approach.

A 2) *T-test*

Difficulty: Easy

R had a built-in function `t.test()` that performs a t-test. Read the Wikipedia entry for the T-test (https://en.wikipedia.org/wiki/Student%27s_t-test) to find out exactly what is going on in this function.

- ☐ Program a function that can be used to perform a t-test. Think about what results the user would like to get and how the user should use the function.

A 3) *Mann-Whitney U test*

Difficulty: Easy

Read the Wikipedia entry for the Mann-Whitney U Test for some first information (https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test). The Mann-Whitney U test is the non-parametric equivalent of the two sample t-test and is used when the assumptions of the parametric t-test are violated.

- ☐ Program your own Mann-Whitney U function and compare it to the built-in Mann-Whitney U test in R to see whether you get the same results.

A 4) *Caesar cipher*

Difficulty: Medium

For some initial information, read the Wikipedia entry on the Caesar cipher encryption/decryption method (https://en.wikipedia.org/wiki/Caesar_cipher).

- ☐ Program a function that can be used to encrypt a character string using the Caesar cipher and program another function that can be used to decrypt a character string using the Caesar cipher.

- ☐ Use the decrypt function to decrypt the following string that is encrypted using a Caesar cipher and a key of 13:

```
Nznmvat jbex! Lbh qrpelcgrq guvf zrffntr pbeerpgyl.
```

A 5) *Infinite monkeys, infinite typewriters*

Difficulty: Medium

The infinite monkey theorem says that a monkey hitting keys at random on keyboard for an infinite amount of time will almost surely type any given text, such as the entire contents of this workbook. In fact, the monkey would almost surely type every possible finite text an infinite number of times. For further information, read the Wikipedia entry for the Infinite Monkey Theorem (https://en.wikipedia.org/wiki/Infinite_monkey_theorem).

- ☐ Simulate one monkey, typing random letters on a typewriter in sequences of 5 letters (so, typing 5 letters, then a space, and so on). Turn this into a function that returns the number of letters typed before making a coherent 5-letter word. Use a file that contains all English 5 letter words to check whether a word is valid (e.g., http://www-cs-faculty.stanford.edu/~knuth/sgb-words.txt).

- ☐ Run the function 500 times and make a nice plot with the results (WARNING: a 500 times will take approximately 30 minutes, so start with a much smaller number for testing your code).

A 6) *Hangman*

Difficulty: Hard

Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word, phrase or sentence and the other(s) tries to guess it by suggesting letters within a certain number of guesses. Read the Wikipedia entry on "Hangman" for some information about the rules of the game (https://en.wikipedia.org/wiki/Hangman_%28game%29).

- ☐ Play Hangman against the computer by programming the game in R. For an additional challenge, try to include graphics for wrong answers.

A 7) *Blackjack*

Difficulty: Hard

Blackjack is a casino card game between one or more players and a dealer, where each player in turn competes against the dealer. Read the Wikipedia entry for the "Blackjack" casino game for some more information about the rules of the game (https://en.wikipedia.org/wiki/Blackjack).

- ☐ Program the Blackjack game. For an additional challenge, try to include a betting system.