CSC 120 (R Section, L0201), Spring 2015 — Assignment #3

Worth 10% of the course grade. Due by 5pm on April 2, by email (see end of this handout). This assignment may be handed in late, with a 20% penalty, by 5pm on April 6. Assignments will not usually be accepted after that. Contact the instructor as soon as possible if you have a legitimate excuse (eg, documented illness) for handing in the assignment late.

This assignment is to be done by each student individually. You may discuss it in general terms with other students, but the work you hand in should be your own. In particular, you shouldn't leave a discussion with someone else with any written notes (either paper or electronic).

In this assignment, you will analyse data from an imaginary randomized clinical trial of a drug to treat osteoporosis (low bone density), using random permutations to assess whether the differences seen are statistically significant. I generated data for this assignment artificially with a simulation program, but you should pretend that the data is real.

The data is from a clinical trial of a new drug to treat osteoporosis, which is a condition in which bone mineral density (BMD) is lower than normal, which can lead to increased risk of bone fracture. Men tend to have higher BMD than women, and younger people tend to have higher BMD than older people. BMD is measured in mg/cm^2. There is of course some error in the measurements.

For the clinical trial, men and women were recruited whose age was between 30 and 79. Their BMD was measured, and only those whose measured BMD at the start of the study was less than 800 mg/cm^2 were retained as subjects in the study. Recruiting continued until 500 suitable subjects were found. These subjects were randomly divided into a group of 250 who received the drug being tested (a pill, taken weekly with the evening meal), and a group of 250 who received a placebo, a pill which looked just like the drug, but contained only inactive ingredients. Both groups were given standard medical advice on how to improve BMD using better diet and exercise.

After ten years, the BMD of all subjects was measured again. During the study, records were also kept of how often subjects reported having a stomach ache the evening after taking the pill, and how often they reported having a head ache the next morning.

The data is in the file at

`http://www.cs.utoronto.ca/~radford/csc120/a3data`

It should be read with `read.table` with the `header=TRUE` option. The resulting data frame will have 500 rows (one for each subject) and the following columns:

`treatment`	Either "drug" or "placebo"
`sex`	Either "M" or "F"
`age`	Age in years
`BMD1`	Measured BMD at the start of the trial
`BMD2`	Measured BMD at the end of the trial
`headache`	Fraction of times taking the pill was followed by headache
`stomachache`	Fraction of times taking the pill was followed by stomach ache

None of this data is missing, and the subjects always took the pills they were supposed to take. (This is rather unrealistic — it would be very unusual in a real study!)

We are primarily interested in inferring from this data whether taking the drug increases BMD over a ten year period (the duration of the study), compared to not taking the drug. Note that it is possible that BMD might decrease in both the group taking the drug and the group taking the placebo, because all subjects were ten years older at the end of the study. It's also possible that BMD might increase in both groups, because all subjects received advice on improving their diet and exercise. So we are interested in the *difference* between the group taking the drug and the group taking the placebo, since that difference is what might be due to the drug.

We are also interested in whether the drug has undesirable side-effects, in particular, whether it causes an increase in headaches or stomach aches. Again, we must focus on the differences between the two groups, in order to see the effects of the drug.

To estimate the effects of the drug, you should write an R function called `treatment_effects` that takes two arguments — a data frame containing the data, and a vector of names for variables for which the effect of taking the drug should be estimated. This function can assume that the data frame has a column named `treatment` whose values are "drug" or "placebo", but it should not assume that the other columns in the data frame are exactly those listed above, since we might want to use this function to analyse other data sets too. The value returned by this function should be a numeric vector the same length as the second argument, that has names for its elements the same as the names of the variables given by the second argument. The element in this returned value with name `n` should be the difference in the average value of variable `n` in the rows of the data frame for which `treatment` is "drug" and the corresponding average in the rows of the data frame for which `treatment` is "placebo".

To estimate the effect of the drug on BMD, we could look at the difference in the average of the `BMD2` variable (the measurement of BMD at the end of the study) between the drug and placebo groups. But we can also look at the change in BMD from the start to the end of the study, which is `BMD2` minus `BMD1`. You should create a new column in the data frame called `BMD_change` that is equal to this difference. You can then look at the difference in the average value of `BMD_change` between the drug group and the placebo group. You might expect this to the be same as the difference in `BMD2` between these groups, and it would be if the average value of `BMD1` was the same in the two groups. But just due to random variation, the average value of `BMD1` won't be exactly the same in the two groups, so looking at the change may give a more precise estimate.

So the full set of variables for which you should estimate the difference between the drug and placebo groups is `BMD2`, `BMD_change`, `headache`, and `stomachache`.

The differences you estimate almost certainly won't be zero, even if the drug in fact has no effect. To see whether we should be confident that any effect of the drug we see is real, you should find a "p-value" for each variable's effect. The p-value is the probability that a difference between the drug and placebo groups that is as large or larger than what you got could come about just by chance, when there is no real effect of the drug. For this assignment, we'll assume that if the drug has any effect on BMD, headaches, or stomach ache, it will be to increase the average values of these variables, so we'll compute a "one-sided" p-value that looks only at the probability of getting a difference equal to or greater than the one you see, and in the same direction.

To find this p-value, we can consider randomly permuting the values of the treatment variable, so that whether a subject is said to be in the drug group or the placebo group is no longer related to whether they actually took the drug or the placebo. In a data set with the treatment variable permuted, any differences in BMD or other variables between the drug and placebo groups must be due just to chance. So we can get a p-value for our estimate of treatment effect on some variable

by seeing what fraction of random permutations of the treatment variable give a difference in the average of that variable between drug and placebo groups that is at least as large as what we got for the real data set.

You should write a function called `permutation_pvalues` that takes as arguments a data frame, a vector of names for variables that we want to find p-values for, and the number of random permutations to use in finding the p-values (which should have a default of 1000). It should return a numeric vector with one named element for each variable, which is the p-value for that variable, found as described in the previous paragraph. Your `permutation_pvalues` function should call your `treatment_effects` function to find estimated effects for both the actual data and the data with the treatment permuted (in many ways), and use these estimates to compute the p-value.

Finally, you should write a function called `print_effects_with_pvalues`, which also takes as arguments a data frame, a vector of names for variables that we want to find effect estimates and p-values for, and the number of random permutations to use in finding the p-values (which should have a default of 1000). This function should print a matrix with two rows, having row names of `effect` and `pvalue`, and with as many columns as there are variables for which effects are estimated (with the names of these variables as column names). The first row should be just the result of calling `treatment_effects`, and the second row the result of calling `permutation_pvalues`. After printing the matrix, it should also print the number of rows in the data frame it was given.

To test your functions, you should create a small data frame for which you can work out the correct answers manually, and use to test your functions. This data frame should have four rows, and three columns named `treatment`, `A`, and `B`. You should set the `treatment` column to have two "drug" values and two "placebo" values. You should set the other two columns to values that seem like they will make a good test (for example, the two variables should produce different p-values). You should work out manually what the treatment effects should be for this data frame, and what the p-values should be. Working out the treatment effects should be easy — you just need to do a few additions and subtractions. To work out the p-values, you need to consider all 24 possible permutations of the treatment values. However, these 24 possible permutations come in 6 groups of 4, with the 4 permutations in each group doing the same thing, since they give the same "drug" or "placebo" value to each subject. It therefore shouldn't be too hard to figure out exactly what fraction of permutations result in an estimated effect that is at least as large as the one for the actual data in the data frame you created.

You should compare these manually computed treatment effects and p-values with what your `print_effects_with_pvalues` function produces, when the number of permutations used is large. The results should be close, but not necessarily exactly the same. You should put this test in an R script, that creates the data frame you use for testing, and then runs your `print_effects_with_pvalues` function on it.

Once you have tested your functions, and think they are working properly, you should create an R script that runs your `print_effects_with_pvalues` function on the entire data set I supplied (for the four variables mentioned above), and on four subsets of the data — just the men, just the women, just the women younger than 50, and just the women 50 or more years old. You should use 1000 permutations to compute the p-values. You should set the random seed to some fixed number at the start of your script, so that the results can be reproduced.

To submit your assignment, send an email to `radford@cdf.utoronto.ca`, with subject line

"A3 your-family-name, your-given-name". The body of the email can be blank (but you can include a note if you like). You should attach three files. The first file should be the .R file containing **only** the definitions of functions `treatment_effects`, `permutation_pvalues`, and `print_effects_with_pvalues`, with suitable documentation on what the functions do. These functions could be used to analyse other data sets as well as the one for this assignment, so this file **must not** contain any references to any specific data set. The second file should be the .html file created by `knitr::spin` when you ran your test script, which as described above creates a data set for testing, and runs `print_effects_with_pvalues` on it. This file should also contain a comment giving the exact values of the correct effects and pvalues that you worked out manually. The third file should be the .html file created by `knitr::spin` when you ran your script that reads the data for the imaginary clinical trial that I supplied, and runs `print_effects_with_pvalues` on this data, and on subsets of it, as specified above. Your two script files should contain brief comments where appropriate that describe what they do.

A final note: To do this assignment, you have to first understand this assignment handout. The March 18 lab exercise was designed to help with this, by showing another example of using permutations to check for statistical significance. (It also gives practice in selecting subsets of a data frame.) If you didn't come to that lab, you should try the lab exercise on your own. You should also read this handout carefully, probably several times. If you don't understand it, you should ask questions. The most useful questions are ones where you indicate as well as you can what part of the handout you didn't understand.