SMM069: Modelling and Data Analysis

Group coursework No. 2: Applications in **VBA** and **R**

Your submission should be:

- A **single** `PDF` file that contains your full report (i.e. brief theory background and implementation in VBA and **R**);
- a separate **single** `EXCEL` file containing your **VBA** solution code, and
- a separate **single R script** file containing your **R** solution code for the task described below.

Marks will be awarded on the basis of the content of all three files.

The `EXCEL` spreadsheets and VBA file must be self-contained with suitable comments, and not linked to any external files and contain sufficient VBA comments that it is readily comprehensible. I should be able to run your VBA code through a button on the worksheet without opening up the VBA editor. In the case that you use the `Macro Recorder` to write some of your VBA code, then you must tidy up the macro code to make it more legible and efficient – failure to do this will lead to loss of marks.

Similarly, the **R** script file should be readily executable in **R** and contain sufficient comments that the program flow is easily tractable and unambiguous. Also, you should provide some application examples with some relevant data samples of your own choice (e.g. either simulated or from standard **R** data sets).

Background

The Wilcoxon signed-rank test is a non-parametric test that can be used to determine whether two dependent samples were selected from populations having the same distribution. It is an alternative to the classical t-test for paired samples when the Normality assumption of this test is not satisfied. The test is formally described as follows.

Suppose we have two dependent samples consisting of n pairs, so in total $2n$ data points. For pairs $i = 1, \ldots, n$, let x_i and y_i denote the paired measurements. We formulate the testing problem:

$$\begin{aligned} H_0 &: \text{differences between measurements are null} \\ H_1 &: \text{differences between measurements are not null} \end{aligned}$$

The Wilcoxon signed-rank test for testing H_0 follows from the steps:

1. Calculate the pairwise differences, $d_i = x_i - y_i$.
2. Remove the zero differences and adjust the sample size as the number of non-zero differences. Denote this number by n_r .
3. Without regard to sign, order the differences from least to greatest, and allocate ranks to the absolute differences. In the event of ties, allocate the average rank to the tied absolute differences.
4. Now reintroduce the signs of the differences. Denote by $SR_{(+)}$, the sum of positive ranks, and by $SR_{(-)}$, the absolute value of sum of negative ranks.
5. Calculate the Wilcoxon test statistic as $T = \min(SR_{(+)}, SR_{(-)})$.

Assuming that the distribution of T is known the test can be concluded deriving the rejection region for a given significance level $0 < \alpha < 1$, or calculating the p-value. Deriving the exact distribution of T is not trivial and instead a Normal approximation is suggested.
As n_r increases the statistic T has a Normal distribution with mean and variance given by:

$$E[T] = \frac{n_r(n_r+1)}{4} \quad \text{and} \quad Var(T) = \frac{n_r(n_r+1)(2n_r+1)}{24}$$

Hence, for n_r large, the test can be concluded continuing the above algorithm as follows.

6. Compute the z-score as $z = \dfrac{T - E[T]}{\sqrt{Var(T)}}$.
7. Reject the null hypothesis of no differences if z is in the either of the $\alpha/2$ extreme tails of $N(0,1)$, for a test with significance level α.
8. Calculate the p-value as $2(1 - \Phi(|z|))$, where Φ is the *c.d.f.* of the $N(0,1)$.

In practice, it is recommended a sample size n_r of at least 10 for valid test, as the Normal approximation breaks down with fewer observations.

Here is a worked example with tied ranks:

A bank employs two firms, A and B, to appraise the value of the real estate properties on which they make loans. To review the consistency of the two firms the bank selected a sample of 10 residential properties and scheduled both firms for an appraisal. The results, reported in \$000, are:

Home	1	2	3	4	5	6	7	8	9	10
A	135	110	131	142	105	130	131	110	125	149
B	128	105	119	140	98	123	127	115	125	145

Calculations for the Wilcoxon signed-rank test are as follows.

Home (i)	x_i	y_i	d_i	d_i (non-zero)	Abs. d_i	ranks	signed-ranks
1	135	128	7	7	7	7	7
2	110	105	5	5	5	4.5	4.5
3	131	119	12	12	12	9	9
4	142	140	2	2	2	1	1
5	105	98	7	7	7	7	7
6	130	123	7	7	7	7	7
7	131	127	4	4	4	2.5	2.5
8	110	115	-5	-5	5	4.5	-4.5
9	125	125	0	exclude	exclude	exclude	exclude
10	149	145	4	4	4	2.5	2.5

From the values in the table we find:

$$\begin{aligned} n &= 10 \text{ and } n_r = 9 \\ SR_{(+)} &= 40.5 \text{ and } SR_{(-)} = 4.5 \quad \Rightarrow T = 4.5 \\ E[T] &= 9 \times 10/4 = 22.5 \\ Var(T) &= 9 \times 10 \times 19/24 = 71.25 \\ z &= \frac{4.5 - 22.5}{\sqrt{71.25}} = -2.13245 \\ p - value &= 0.03297 \end{aligned}$$

At level $\alpha = 0.05$, compare $|z|$ to the percentage point $z_{0.025} = \Phi^{-1}(0.975)$. Since $|z| = 2.13245 > z_{0.025} = 1.959964$ then reject H_0. Alternatively, we can reach the same conclusion by observing that $\alpha > p - value$, that is: $0.05 > 0.03297$.

NOTE: There is a lack of consensus on the definition of Wilcoxon signed-rank test in the literature, so be careful if you look at different sources. The version of the test you should consider for this coursework has been fully described above.

Task 1: VBA implementation of the Wilcoxon signed-rank test

[30 marks]

Unfortunately, the Wilcoxon signed-rank test does not appear to be implemented in `EXCEL` in the `Data Analysis Toolpak`. So, your task is to create a `Dialog` or `User Form (UF)` that appears when one presses a button on the worksheet, with which they can carry out the Wilcoxon signed-rank test for data on the same worksheet. The user should be able to input the cell ranges into the dialog by mouse clicking and dragging, and have an option to include the labels in the data selected. Further, your VBA program (i.e. `UF`) should allow the user to input the significance level of the test, which might be set to a default value of 5%.

They also need to be able to specify the cell relative to which the outputs from your analysis appears at. There should be two type of outputs. The first type, should show in a tabular format all the intermediate calculations of the test (e.g. similar as in the illustration above). This should be provided ***only*** at the request of the user (e.g. when a checkbox is ticked in the UF). The second type, should provide the overall results of the Wilcoxon test. Namely, it should show the rank sums of each positive and negative ranks, the T test statistic, the mean and variance of T, the z-score and the associated p-value.

In addition, the program should be robust and cater for sufficient error traps and provide interactive warnings for possible errors resulting from unexpected (incorrect) user entries. For example, insufficient data (i.e. $n < 10$), incorrect range references, non-numeric significance level, or you might think of the case where the data ranges provided by the user during the execution of the program are not pairwise data but two samples of different length, etc.. Further, the program should check whether or not there is any existent data on the worksheet that might be overwritten by any of the outputs and, if necessary, give adequate warnings that allow the user to make corrections or ignore.

HINT: I suggest you study some of the `Data Analysis Toolpak` applications for inspirations regarding the usual features of their dialogs and outputs.

Your written report should give descriptions of the algorithms you have implemented, with copies of the VBA macro code, and a screenshot of the user-input dialog. Your VBA code should also be commented to that I can understand your algorithms and the chosen approach.

Task 2: R implementation of the Wilcoxon signed-rank test

[20 marks]

Your final task is to implement the Wilcoxon signed-rank test but now using **R**.

You have to create a function in **R** which calculates the test from data in the form of two vectors or a single `data.frame` with two columns `x` and `y`. Thus, correspondingly, the function will require two arguments for the data input: `x` and `y` (i.e. sample data), whereas the second will be redundant in case the first one is in the form of a `data.frame` with the right columns. In addition, the function should have two optional arguments: one for the significance level (that is set by default to 5%) and one for the type of output (as in `Task 1`, that is set by default to basic only – i.e. without the full table/`data.frame` output).

Similarly, to the VBA version, the function should fully validate the data input provided by the user and give necessary warning messages in case of errors and/or incorrect input (i.e. as in `Task 1`). Alternatively, the function could carry out automatic adjustments to make the input suitable (e.g. remove NAs, transform data to numeric, etc.), while providing relevant warnings. Finally, you should aim to write an efficient program flow that is as short as possible to perform the task.

WARNING: You might note that there is a function `wilcox.test()` in the package `stats` which performs a more general version of the test. You shouldn't use this function in your solution! The idea is that you write original code for the test, using only R commands and basic R functions. One of the functions which you might use is `rank()` to easily derive the ranks for the test. This function is in the package `base`.

Your solution to this task should be fully described in your report. You should include your R code, with suitable comments so I can follow it, and the output provided when the code is executed with some data.