A COMPUTER METHOD FOR CALCULATING KENDALL'S TAU WITH UNGROUPED DATA

WILLIAM R. KNIGHT
Fisheries Research Board of Canada, Biological Station, St. Andrews, N.B.
and
The University of New Brunswick[1]

Adapting the usual manual methods of computing Kendall's *tau* to automatic computation result in a running time of order N^2. A method is described with running time of order $N \log N$.

1. INTRODUCTION

OF THE usual methods for calculation of the rank correlation coefficient, Kendall's *tau*, none is efficient for automatic computation with large samples. In every case known to the author each of the N pairs, (X_i, Y_i), is compared with every other pair, (X_j, Y_j), resulting in a running time of order N^2. The process of calculating *tau* is closely related to that of ordering a list of numbers in internal storage, i.e., performing an internal sort, but internal sorting algorithms with running time of order $N \log N$ are known. The reader is referred to Flores [1], Friend [2], and Gotlieb and Hume (section 10.4) [4] for descriptions of particular sorting methods and general discussion of the field.

Considering this relation between calculation of *tau* and an internal sort, we propose a method with running time of order $N \log N$. Its running time is illustrated here and corrections for ties are discussed. The method is not recommended for manual computation.

The problem of calculating Kendall's *tau* arose while attempting to evaluate species associations in catches by the Canadian east coast offshore fishery. Sample sizes ranging up to 400 were common, making manual calculations out of the question; indeed, an initial program using an asymptotically inefficient method proved expensively slow.

2. INTERNAL SORTING AND COMPUTATION OF TAU

Suppose a list of numbers is ordered as follows. The list is scanned, every time two adjacent numbers not in order are encountered, the pair is exchanged. Scanning is repeated until the list is in order. If the list is first ordered with respect to the first variable, X, then with respect to the second variable, Y, and a count, s, of the number of exchanges made in the second ordering is taken, then

$$\tau = 1 - [4s/N(N-1)]. \tag{1}$$

This ordering method is called *sorting by exchanging*. The reader may recognize it as one of the methods given by Kendall [5] section 1.13 for calculating *tau*.

Any method of sorting which permits the exchange count, s, to be carried

[1] Present address: Department of Mathematics, University of New Brunswick, Fredericton, N. B.

can be used to calculate *tau*. Linear, quadratic, and pth power selection, sorting by merging, counting, and insertion can be used and several have been, e.g., Kendall (section 1.9) [5] (counting) and Lieberson [7] (inserting?). In the other hand, digital sorting and address calculation will not yield an exchange count, although these can be used for the initial X sort. The author is grateful to one of the referees for pointing this out.

Of those sorting methods known to the author which are adaptable to computing *tau*, the merge sort is asymptotically the fastest. One version proceeds by ordering in successive stages sublists of length 2, 4, 8, $\cdots$, 2^k, $\cdots$. To avoid trivial complications, N will be assumed a power of 2. On the kth stage the list is divided into sublists of length 2^k, and each of these sublists ordered within itself. For example, the list:

$$2, 7, 5, 3, 4, 8, 6, 1$$

would become after the first stage

$$2, 7; 3, 5; 4, 8; 1, 6$$

after the second stage

$$2, 3, 5, 7; 1, 4, 6, 8$$

and be completely ordered after the third stage.

Every time an item is moved forward (not backward) in the list the exchange count is augmented by the number of places moved. In the above example the exchange count is 2 after the first stage, 7 after the second stage, and 14 after the third stage.

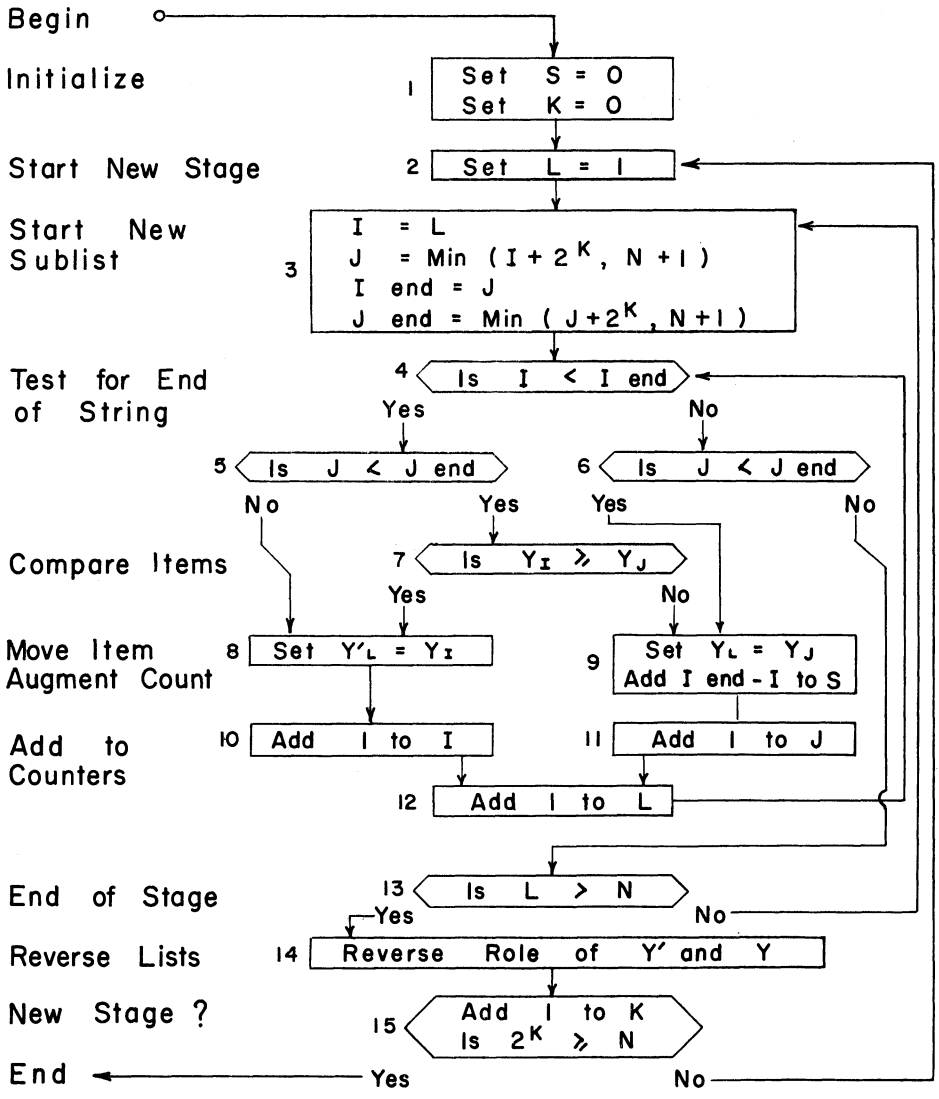
A flow chart of this process is given as Figure 1, to which the reader wishing more detail is referred.

3. RUNNING TIME

Running time for various sorting methods is discussed by Flores [1] and Gotlieb and Hume [4]. Most methods allowing an exchange count have running time of order N^2, exceptions being pth power selection with time of order $N^{1+1/p}$, and merging with running time or order $N \log N$. The latter is derived for the version described here by noting that at the kth stage the sublists being sorted consist of two strings of length 2^{k-1} already in order from previous stages; thus, finding the largest item to be moved to the head of the sublist requires only one comparison, that of the two items at the head of each string. Generally, finding the rth largest item to be moved to rth place requires only the one comparison of items at the head of the unused portions of each string. The running time for one stage is thus of order N and there are $\log_2 N$ stages.

Comparison of the proposed with one of the faster and more simply programmed of the order N^2 methods, that based on sorting by counting, is made in Tables I and II. Table I shows the number of operations in the inner loop, in both cases. Table II compares machine runs for both methods made on the IBM 1620 at the University of New Brunswick.

FLOW CHART FOR SORTING AND COUNTING BY MERGING



4. TIES

If ties are present they can be broken in numerous ways yielding different values of the exchange count ranging from, say, s^- to s^+. Usually s is taken as $(s^-+s^+)/2$.

To obtain s^-, sort as follows: First, in the preliminary ordering upon X, break all ties according to Y. (If Y is also tied the choice is immaterial.) Second, if during the second sort the two currently compared Y values are equal, the one higher on the list is chosen to move to the head of the sublist. The reverse of this procedure yields s^+, which is equal to $s^-+T+U-V$, where T, U, and V are respectively the number of tied pairs in X, in Y, and the number of jointly tied pairs. Formulae are given for T and U by Kendall (section

TABLE I. OPERATIONS IN INNER LOOP

Operation	Sorting by	
	Counting	Merging
	Multiply by	
	$N(N-1)/2$	$N \log_2 N$
Comparisons	2	3
Additions	1	2.5
Subscript calculations	2	3
Data transfers	0	1

TABLE II. EMPIRICAL RUNNING TIMES (ON AN IBM 1620)

Sample size N	Approximate running time in seconds using sorting by	
	Counting	Merging
20	0.8	1.1
30	1.2	1.2
40	2.0	1.6
50	2.6	1.8
100	9.4	3.9
200	36.	8.0
500	217.	22.

3.4) [5], and that for V is analogous. Given s^-, T, U, and V, Kendall's τ_b and indices like the Goodman-Kruskal (Kruskal [6], Goodman and Kruskal [3]) *gamma* can be calculated.

Since straightforward methods of calculating T, U, and V lead to running times of order N, no details are given, save the obvious remark that T should be calculated while the list is in X order, U while in Y order, and V while in either order.

REFERENCES

[1] Flores, Ivan, "Analysis of internal computer sorting," *Journal of the Association for Computing Machinery*, 8, 1961, 41–80.

[2] Friend, Edward Harry, "Sorting on electronic computer systems," *Journal of the Association for Computing Machinery*, 3, 1956, 134–69.

[3] Goodman, Leo A. and Kruskal, William H., "Measures of association for cross classifications. III: Approximate sampling theory," *Journal of the American Statistical Association*, 58, 1963, 310–64.

[4] Gotlieb, C. C. and Hume, J. N. P., *High Speed Data Processing*. Toronto: McGraw-Hill Book Co., 1958.

[5] Kendall, Maurice, *Rank Correlation Methods*. London: Charles Griffin and Co., 1948.

[6] Kruskal, William H., "Ordinal measures of association," *Journal of the American Statistical Association*, 53, 1958, 814–61.

[7] Lieberson, Stanley, "Non-graphic computation of Kendall's tau," *The American Statistician*, 14, 1961, 20–1.