COMPUTING FOR BUSINESS AND HOME

INTERFACE AGE™

6/9

SEPTEMBER 1981 $2.50/CANADA/MEXICO $3.00

COMPUTER OPERATIONS IN MEDICINE

Evaluating Computer Heart Attack Risk

Micro Power in Your Pocket

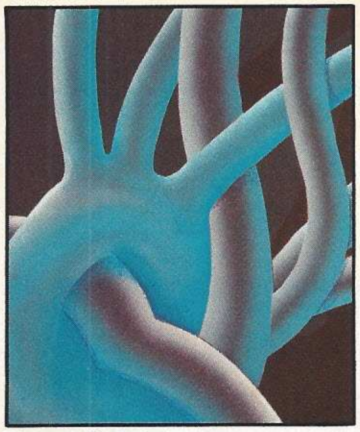
0 71486 02651 09
CC 02651

VOL. 6, ISSUE 9
SINCE DECEMBER 1975

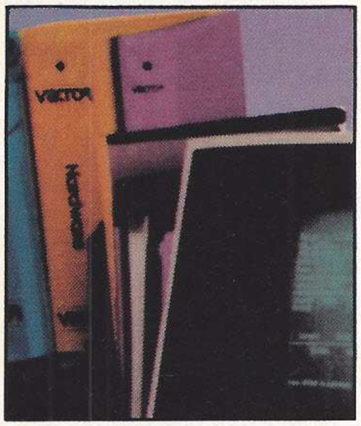
SEPTEMBER 1981

INTERFACE AGE™

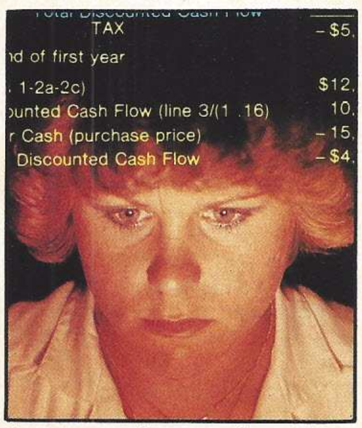
COMPUTING FOR BUSINESS AND HOME APPLICATIONS



Heart II: Risk Evaluation Revisited . . . 66



Choosing a Medical Office Computer 70



Programming Equipment Purchase Options 74

Cover: Operating room courtesy of La Palma Community Hospital, La Palma, CA.

Photograph by Don May

Models: Kay Lee, Tom Fox, John Smith

BPA SELECTED MARKET AUDIT DIVISION

FEATURES

COLUMNS

DEPARTMENTS

Contact authors of monthly columns by writing to them at INTERFACE AGE, P.O. Box 1234, Cerritos, CA 90701 in care of their respective columns.

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $18.00, Canada $24.00, all other countries $35.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher. Circulation Department, (213) 926-9540.

INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. USPS No. 580-310. ISSN Publication No. 0147-2992.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Second class postage paid at Lincoln, Nebraska and Artesia, California.

THE MICRO-MATHEMATICIAN

by Dr. John C. Nash

Fundamental Statistical Calculations

This month we consider some elementary statistics. The mean and variance are the most widely used measures of location and dispersion of data.

Suppose x_i is the i'th observation of some variate X. We shall use the sum of the observations

$$T(1,m) = \sum_{i=1}^{m} x_i \qquad (1)$$

to define the arithmetic mean as the average

$$A = T(1,m) / m. \qquad (2)$$

The variance is concerned with deviations from the mean. We shall define the sum of squares

$$S(1,m) = \sum_{i=1}^{m} (x_i - A)^2, \qquad (3)$$

so that the variance is given as

$$V = S(1,m) / (m - 1). \qquad (4)$$

(Some authors divide by m instead of (m − 1) in equation (4). I can offer no certain opinion as to which form is best, as arguments can be made in favor of both, but the definition (4) seems to be most common in practice.)

Calculation of the mean and variance via the sums (1) and (3) is straightforward. However, we must pass through the data twice—once to compute the mean and once to compute the variance. This may be costly when large amounts of data are involved, so that we are forced to read the data from tape or disk. Therefore, we seek to compute both the mean and the variance in a single pass through the data.

Expansion of the sum (3) gives

$$\begin{aligned} S(1,m) &= \left(\sum_{i=1}^{m} x_i^2\right) - 2A \sum_{i=1}^{m} x_i + A^2 \sum_{i=1}^{m} (1) \\ &= \sum_{i=1}^{m} x_i^2 - 2m A^2 + m A^2 = \sum_{i=1}^{m} x_i^2 - m A^2. \end{aligned} \qquad (5)$$

This "textbook" formula for the variance (sum of squares) may also be written

$$S(1,m) = \sum_{i=1}^{m} x_i^2 - (T(1,m))^2 / m. \qquad (6)$$

Unfortunately it is a very poor formula by which to compute the variance, even though we form the sums of x_i and x_i^2 simultaneously.

Consider, for instance, the following data.

x	x^2
1000	1000000
1001	1002001
1002	1004004
1003	1006009
4006	4012014

Thus, equation (6) in exact arithmetic gives

$$\begin{aligned} S(1,4) &= 4012014 - (16048036) / 4 \\ &= 4012014 - 4012009 = 5. \end{aligned}$$

Clearly, this formula gives rise to a subtraction involving nearly equal large numbers and would fail to compute the correct variance on a machine working in less than 8 digit (decimal) precision. Yet the magnitudes of the four data elements are hardly abnormal—think only of the Dow Jones index over a week of trading.

In order to avoid the problem of subtractions involving nearly equal large numbers, we could *shift* the data by some number z, and find the mean and variance of a new variate

$$Y = X - z. \qquad (7)$$

The mean of X is then found by adding z to the mean of Y, while their variances are theoretically identical, as can be verified by substitutions in equations (1) through (4). Some practitioners take the first data value for z; others enter a "guess" of the mean of the variate X. The first strategy is poor when x_1 is not representative of the rest of the data. The second strategy cannot be automated. Nevertheless, in the example above, choosing z = 1000 gives

$$\begin{aligned} S(1,4) &= (0 + 1 + 4 + 9) - (0 + 1 + 2 + 3)^2 / 4 \\ &= 14 - (36/4) = 5, \end{aligned}$$

which can easily be carried out correctly even on a 2 digit computer.

A number of papers have been published in the last few years on computing the mean and variance in a stable way, while making only a single pass through the data. Such methods are commonly referred to as *updating* methods because they alter the values of the sums T and S by using new data values x_i.

The general updating formula is derived as follows. Suppose we have two sets of observations on the variate X, one with m data elements, the other with n. For these two sets of data, we already have the sums

$$T(1,m),\ T(m+1,\ m+n),\ S(1,m), \text{ and } S(m+1,\ m+n).$$

The notation used allows us to number the data elements consecutively. In each sum, the index of summation is taken to vary from the first value inside the brackets to the second value.

We wish to find the sums T(1,m+n) and S(1,m+n). From equation (1) it is obvious that

$$T(1,m+n) = T(1,m) + T(m+1,m+n). \qquad (8)$$

The sum of the squares is more tricky, but is tedious rather than difficult to derive. The formula is

$$\begin{aligned} S(1,m+n) = {} & S(1,m) + S(m+1,m+n) + (m/(n^*(m+n))) \\ & [(n/m)\, T(1,m) - T(m+1,m+n)]^2 \end{aligned} \qquad (9)$$

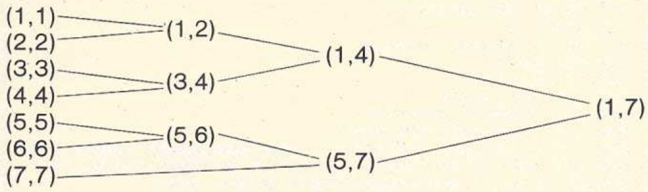
It is easily verified by expanding all terms as in equation (6) and noting equation (8).

We could use these two formulae (8) and (9) to compute the mean and variance by updating the sums S and T one data element at a time. That is, n always has the value 1 in the expressions. We start with both sums zero.

Data value	m	T	S
1000	0	1000	0
1001	1	2001	$0 + (1/2)[1000 - 1001]^2 = 1/2$
1002	2	3003	$1/2 + (2/3)[2001/2 - 1002]^2 = 1/2 + 3/2 = 2$
1003	3	4006	$2 + (3/4)[3003/3 - 1003]^2 = 2 + 3 = 5$

Clearly, we have found the correct sums and not read the data but once, though there is considerably more effort at each step of the update. This is probably worthwhile in cases involving many thousands of data elements. Of course, when there are many observations, equation (8) should not be used one element at a time, since this will mean adding a small number to a potentially large one. For instance, if we have a million observations of a variate in the range 1 to 10, those added into the sum T late in the calculation are likely to have only their most significant digits counted in the sum. This is the old "small number added to big number" problem. However, there is a delightfully elegant method of finding the mean and variance in a stable way which avoids this difficulty by using equations (8) and (9) on balanced sub-sets of the data. This is the Pairwise Algorithm due to T.F. Chan, G.H. Golub and R.J. LeVeque which is described, along with many other interesting ideas on this subject, in a Stanford University Computer Science report (STAN-CS-79-773).

The Pairwise Algorithm is true to its name. We take all pairs of data elements and form T and S appropriately. Then we combine pairs into fours, fours into eights, and so on. Odd observations do not give rise to difficulties in the general updating formulae (8) and (9). Illustrated below is the structure of the calculation for 7 observations. We use the notation (i,j) to denote the sums S and T for observations i through j.



In preparing a program to implement these ideas, it is important to observe that we do not have to have all the sums of two observations before proceeding to compute those for four observations. In fact, for the calculation involving 7 observations, we can form the sums in the following order

(1,2), (3,4), (1,4), (5,6), (5,7), (1,7).

The calculations can therefore be arranged to use a stack structure which stores S, T and the number of observations, P, for each node in the illustration. Observations are pushed onto the T stack, with the number 1 (or a weighting) onto the P stack and zero onto the sum of squares S. That is, the sum T of a single observation is just the number itself, there is only one such number, and it has a zero sum of squares. Thereafter, the observation can be treated like any other node in the figure.

Manipulating the stacks

If the two top items on the P stack are equal (that is, an equal number of observations), we use the updating formulae (8) and (9) to pop the stacks by combining the sums. Remember that the number of observations on the P stack must also be added together to form the new top element on this stack. This stack pop is repeated if the top two items on the P stack are again equal. Otherwise, a new observation is pushed onto the stack. When the data is exhausted, the updating formulae are used to collapse the stacks to a single level which gives the sums S, T and the number of observations. Note that this means that the amount of data does not have to be known when we begin the calculation—in fact, the data could be coming in over a communications line from a remote instrument.

Listing 1 shows the stack structure at each step in computing mean and variance for the seven data elements 1000, 1001, 1002, ..., 1006.

Listing 2 gives a Basic program implementing the Pairwise Algorithm. This has a different organization from the Fortran program given by Chan, Golub and LeVeque, which is hopefully better suited to Basic programming and debugging. The program requires a subroutine at line 0660 to return the I'th observation on the variate in variable X. This subroutine may be called with I = 0 to provide initialization of the problem. The subroutine should change I to −1 when the data has been exhausted. An example subroutine is included in the listing which generates observations as scaled integers from Q to R. It can be fairly easily shown that the sum of squares is

$$S(Q,R) = F^2 (R - Q) (R - Q + 1) (R - Q + 2) / 12,$$

where F is a user-supplied scaling factor. Moreover, the product $(R - Q)(R - Q + 1)(R - Q + 2)$ is always exactly divisible by 6. This gives a test for variance programs, which is easily generated yet avoids the need for extended precision arithmetic to calculate the "exact" results to which answers are to be compared. While there are statistical and possibly numerical objections to tests based on the integers, this test should still be of value in detecting programming errors. □

Listing 1

```
PVAR - PAIRWISE ALGORITHM
COMPUTES MEAN AND VARIANCE OF DATA
IN A SINGLE PASS BY UPDATING FORMULAE
USE INTEGERS FROM ?1000
   TO  ?1006
SCALING FACTOR (0 USES 1.0/MEAN) ?1

OBSERVATION # 1 =  1000
STACKS  T             S              P
      1000            0              1

OBSERVATION # 2 =  1001
STACKS  T             S              P
      1001            0              1
      1000            0              1

PAIRED SUMS - POP STACK
STACKS  T             S              P
      2001           .5              2

OBSERVATION # 3 =  1002
STACKS  T             S              P
      1002            0              1
      2001           .5              2

OBSERVATION # 4 =  1003
STACKS  T             S              P
      1003            0              1
      1002            0              1
      2001           .5              2

PAIRED SUMS - POP STACK
STACKS  T             S              P
      2005           .5              2
      2001           .5              2

PAIRED SUMS - POP STACK
STACKS  T             S              P
      4006            5              4

OBSERVATION # 5 =  1004
STACKS  T             S              P
      1004            0              1
      4006            5              4
```

```
OBSERVATION # 6 =   1005
STACKS  T            S           P
      1005           0           1
      1004           0           1
      4006           5           4

PAIRED SUMS - POP STACK
STACKS  T            S           P
      2009           .5          2
      4006           5           4

OBSERVATION # 7 =   1006
STACKS  T            S           P
      1006           0           1
      2009           .5          2
      4006           5           4

END OF DATA - COLLAPSE STACK
STACKS  T            S           P
      3015           2           3
      4006           5           4

STACKS  T            S           P
      7021           28          7

RESULTS
 7 POINTS  SUM OF DATA= 7021
MEAN= 1003  SUMSQUARES= 28
VARIANCE= 4.6666667
STOP IN LINE 470
```

Listing 2

```
10 PRINT 'PVAR - PAIRWISE ALGORITHM '
20 PRINT 'COMPUTES MEAN AND VARIANCE OF DATA'
30 PRINT 'IN A SINGLE PASS BY UPDATING FORMULAE'
40 REM AUTHOR J C NASH 810501
50 DIM T(30),S(30),P(30)
60 REM DESIGNED TO SHOW THE ACCUMULATION
70 REM S STORES SUMSQUARES
80 REM P STORES THE NUMBER OF POINTS
90 REM T STORES THE SUM (MEAN  *  # POINTS)
100 REM INITIALIZE DATA
110 LET I=0
120 GOSUB 660 \ REM CALL SUBROUTINE WITH I=0 FOR INIT'L
130 REM STACK POINTER IS K (TOP)
140 LET K=0 \ REM INITIALLY ZERO
150 REM BEGIN READING OBSERVATIONS AND PUSHING ON STACK
160 LET I=I+1 \ REM NEXT DATA ELEMENT
170 GOSUB 660 \ REM GET IT
180 IF I<0 THEN 350 \ REM END DATA?
190 LET K=K+1 \ REM INCREMENT TOP OF STACK
200 IF K<=30 THEN 230
210 PRINT 'STACK OVERFLOW'
220 STOP
230 LET P(K)=1 \ REM 1 OBSERVATION AT A TIME
240 LET S(K)=0 \ REM IT HAS ZERO VARIANCE
250 LET T(K)=X \ REM AND IS ITS OWN MEAN
260 PRINT 'OBSERVATION #',I,' = ',X
270 GOSUB 590
280 IF K=1 THEN 160 \ REM STACK ONLY 1 DEEP
290 REM COLLAPSE STACK?
300 IF P(K)<>P(K-1) THEN 160 \ REM ARE SAMPLES SAME SIZE
310 GOSUB 490 \ REM COMBINES K,K-1
320 PRINT 'PAIRED SUMS - POP STACK'
330 GOSUB 590
340 GOTO 280
350 REM END OF DATA -- COMBINE REGARDLESS
360 PRINT 'END OF DATA - COLLAPSE STACK'
370 IF K<1 THEN 470
380 IF K=1 THEN 420
390 GOSUB 490
400 GOSUB 590
410 IF K>1 THEN 390
420 PRINT 'RESULTS'
430 PRINT P(1),' POINTS  SUM OF DATA=',T(1)
440 PRINT 'MEAN=',T(1)/P(1),'  SUMSQUARES=',S(1)
450 IF P(1)>1 THEN PRINT 'VARIANCE=',S(1)/(P(1)-1)
460 STOP
470 PRINT 'NO DATA ???'
480 STOP
490 REM COMBINING FORMULA
500 LET N=P(K)
510 LET M=P(K-1)
520 LET P(K-1)=M+N
530 LET T1=N*T(K-1)/M-T(K)
540 LET T2=M*T1*T1/(N*(M+N))
550 LET S(K-1)=S(K-1)+S(K)+T2 \ REM SUMSQUARES UPDATE (9)
560 LET T(K-1)=T(K-1)+T(K) \ REM SUM OF OBSERVATIONS (8)
570 LET K=K-1 \ REM REDUCE STACK HEIGHT (POP)
580 RETURN
590 PRINT 'STACKS  T              S              P'
600 FOR J=K TO 1 STEP -1
610 PRINT TAB(5),T(J),TAB(20),S(J),TAB(33),P(J)
620 NEXT J
630 PRINT
640 REM INPUT Z$ \ REM HIT (CR) TO CONTINUE
650 RETURN
660 REM NASH TEST USING INTEGERS Q TO R
670 IF I>0 THEN 780
680 PRINT 'USE INTEGERS FROM ',
690 INPUT Q
700 PRINT '   TO  ',
710 INPUT R
720 IF R<Q THEN 680 \ REM ERROR CHECK
730 PRINT 'SCALING FACTOR (0 USES 1.0/MEAN) ',
740 INPUT F
750 PRINT
760 IF F=0 THEN LET F=2/(Q+R)
770 RETURN \ REM END INITIALIZATION
780 REM CHECK FOR END OF DATA
790 IF I<=R-Q+1 THEN 820
800 LET I=-1 \ REM END OF DATA
810 RETURN
820 LET X=I+Q-1 \ REM SIMPLE SHIFT TO GET OBSERVATION
830 LET X=X*F \ REM THEN SCALE
840 RETURN
850 END
```