# 12
# Compare It

Mr Smith grows two varieties of tomato in his greenhouse. He has the same number of plants of each variety and looks after them in the same way. Each season he keeps a record ofhow many kilograms of tomatoes are produced by all of the plants of each variety. After six seasons he looks at his figures:

| *Variety* | *Seasonal production (kg)* | | | | | |
|---|---|---|---|---|---|---|
| Catsup | 140 | 120 | 150 | 125 | 160 | 130 |
| Ketchup | 140 | 110 | 180 | 150 | 165 | 125 |

The total production of Catsup is 785kg, while that of Ketchup is 870kg. There is difference - but this is to be expected. After all, it would be very unusual if both varieties gave exactly the same total. The figures show that some years give better yields for both varieties, some give worse for both. Although Ketchup has the higher total, there are two years in which it does worse than Catsup. The situation is far from clear-cut. Mr Smith wants to know if Ketchup is a higher yielding variety than Catsup in the long run.

If only figures would give a straightforward answer to our queries ! More often than not, there are uncertainties, as in the story above. The difference between the yields of the two varieties, totalled over the six years, might be due to Ketchup being the better-yielding variety overall. But it is also possible that the difference is merely the result of chance variations, of the kind that are obviously occuring with both varieties from year to year. What this program does is to help you compare two sets of data to see if the difference between them is more likely to be real, or more likely to be due to chance. The program tells you how significant the difference is.

COMPARE IT is not limited to finding out about tomatoes. In almost any instance in which there are two sets of data to be compared, this program can be brought into use. It is interesting, and possibly profitable, to use it to compare the weights ofsweets, potato crisps, etc., found in packets sold by two different manufacturers. Some firms put more into their packets, on average, than do other firms, even though the quantity marked on the packet may be exactly the same for both firms. Weighing the contents of several packets from each of two firms can show you which is the better buy. Use this test to find out if the difference between the quantities packeted by each firm really is significant. The program is able to accept several different kinds of data. It accepts integer values, such as those Mr Smith obtained by weighing his tomatoes. It can also accept numbers with decimal places, and negative numbers. You might measure the minimum temperature in two different parts of your garden during the winter. This would give you two sets of temperatures, which would almost certainly contain some negative (below freezing-point) values. Comparing the two sets would tell you whether one part of the garden was warmer than the other, on average. If the difference turned out to be significant, the warmer spot would be the place to site a greenhouse, a rabbit hutch, or a bee-hive, or perhaps to plant a tender shrub.

The usefulness of the program is widened by the fact that it works just as well with numbers which represent ranks, grades, or ratings as it does with ordinary numbers. If you are trying to find out which of two makes ofjam have the best taste, it is not possible to rate the taste in grams, centimetres or any other recognised unit. Instead, you would ask, say, twenty different people to taste samples ofthe jam. You would ask them to grade the twojams on a scale running from 1 (poor), through 5 (average) to 10 (excellent). The sets of grades for each kind ofjam are compared, using the program.

Grading is used in many kinds of application where quantitative measurements are difficult or impossible:

Tasting of foods, teas, wine.

Assessing visual features, such as the colours of flowers.

Assessing other features such as the texture of fabrics.

Rating human performance, such as musical and sporting skills.

Rating human reactions, such as satisfaction with a certain product or service, happiness in a givenjob, friendliness with others.

There seems to be no end to the applications of COMPARE IT!

If you bear a few special points in mind when using the program, you will be more likely to get a clear answer from the test. There must be at least five items of data in each of the two sets to be compared. There need not be the same number of items in each set: for example, ifyou wanted to conduct a popularity survey oftwo TNJ programmes, you could ask fifteen passers-by to give their rating of a particular TV programme, yet interview twenty other passers-by about the other programme. But the more nearly equal the sizes of the two groups, the more reliable will be the result of the test.

## Using the program

All you have to do is to key in the data and then wait a few tens of seconds for the Electron to work out the results.

First of all you are asked to give a name to each of the two sets of data. For example, Mr Smith would have typed.in 'CATSUP' and 'KETCHUP'. Then you are asked to key in the items of one set of data. Key them in one after another, pressing RETURN after you have entered each item. The program allows you to key in as many as 50 items in each set. When you have keyed in all the items ofone set, key in '9999' as the final item. This figure will not be included among the data. It is just a convenient way of letting the Electron know when the last item has been typed in.

Then you are asked to type in the data for the second set, finishing with '9999' in the same way.

After you have finished entering each set, you are asked 'GK? Y/N'. If you have many items of data, they are displayed a screenful atatime. Press SHIFT toview the next screenful. Pressing'N'atthe end gives you a chance to start again ifyou have made a mistake in the keying. When you key 'Y' after the second set of data, the screen clears and a message is displiiyed reminding you that some time may be needed for the calculations to be completed. If you have only, say, ten or twenty items in each set, only a few tens ofseconds will be required, but it could take a minute or two with larger amounts of data.

The final display gives you the computer's assessment of how much meaning you may safely attach to your data. Mr Smith would have been given the following assessment:

'THERE IS A CHANCE BETWEEN 65% AND 70% THAT THE CATSUP DATA DIFFERS FROM THE KETCHUP DATA'

This statement is explained as follows. The differences between the total six-year yields ofthe two varieties must be due to one oftwo things:

(1) One variety is a truly better-yielding variety than the other,
or
(2) There is no real difference between varieties: differences that exist are solely due to chance variations from piant to plant and from season to season.

From what Mr Smith saw on his computer screen he concluded that there is a 65% to 70% chance of the first explanation being the true one. Conversely, there is a 30% to 35% chance that the difference is purely random. Although there is a difference between the crops over the years, this difference is not significant when compared with the comparatively large variations in yield between different plants and from year to year. If the chance of there being a real difference is only 65% to 70%, it is not sensible to base any drastic decision on the yield figures. There is no real reason for giving up growing Catsup tomatoes, for example. If Mr Smith still wants to know if one variety is better than the other, he must carry on for a few more years and gather more data.

To sum up, the program tells us the likelihood of the difference between two sets of figures being significant. Most people would take a 90% chance (or more) to indicate a true difference, and take a 65% chance (or less) to mean that no significant diffbrence exists.

Users of this program, normally want to show that there is a difference between the *average* of values in one set of data and the *average* of values in the other. This is what Mr Smith wanted to do. It is also possible to have two sets of data in which the averages do not differ markedly, but in which the values of one set are spread over a much wider *range* than those of another. For example, two different brands of Cheddar cheese might, on average, be equally palatable. But one brand might be consistently average in palatability, while the other brand varies wildly. The worst cheeses of the second brand might be almost impossible to eat, while their best cheese might be superb. Given the results of tasting tests, this program would detect such a difference and assess the likelihood of its being significant.

The program detects difference of average values and difference of spread. It does not tell you which kind ofdifference (ifany) occurs in

your figures. Having obtained a result which declares that two sets of data are significantly different, inspect the original figures to see what kind of difference they show.

## Keying in

Note the decimal point before the ' 5ln line 400. The crucial lines in the calculation are lines 500 to 520. Check these very carefully - especially the brackets. The upward pointing arrow in line 510 is the exponentiation sign. Obtain this by pressing SHIFT and the ' ^ ~ left-arrow ' key.

When typing the DATA statements of lines 770 to 790, note that it contains both full stops (i.e. decimal points) and commas. These must all be typed exactly as printed in the book.

## Program design

20-30 Initialising.
40-160 Collecting two sets of data.
170-180 Transferring data to arrays, ready for processing.
190 Warning of delay.
200-210 Sorting data into numerical order.
220-430 Resolving tied values.
440-490 Counting the number of runs.
500-520 Calculating the statistics.
530-720 Assessing probabilities and displaying the results.
730-760 Inviting another analysis.
770-790 Statistical look-up table.
800-890 PROCquicksort to sort data into numerical order.

## Points of interest

The program uses PROCquicksort, a fast sorting routine. It could be used in any other program in which numbers are to be sorted into ascending numerical order. The data is to be in an array S(). The procedure sorts all items in the array from S(P%) to S(Q%), inclusive. It is interesting that this procedure is a recursive one, calling itself at lines 880 and 890.

Note for the more statistically-minded reader: this program uses the

Wald-Wolfowitz runs test, adapted for analysing two-sample data. Ties are resolved alternately one way or the other, any odd tie being resolved at random. The DATA statements contain the values of z associated with selected probability (one-tailed) levels.

## The program

```
   10 REM ** COMPARE IT **
   20 MODE 4:VDU 14
   30 DIM D(2,50),S(101),R(101),N$(2),N(
2):K=0:FT=0
   40 CLS:INPUT''"NAME FOR FIRST SET OF
DATA "N$(1)
   50 INPUT'"NAME FOR SECOND SET OF DATA
 "N$(2)
   60 FOR J=1 TO 2
   70 N(J)=1
   80 D(J,K)=0:CLS:PRINT''"KEY IN DATA F
OR "N$(J)
   90 PRINT';N(J);:INPUT"  "D(J,N(J))
  100 IF N(J)<50 AND D(J,N(J))<>9999 THE
N N(J)=N(J)+1:GOTO 90
  110 CLS:PRINT''"DATA FOR " N$(J)" IS:"
  120 N(J)=N(J)-1:FOR K=1 TO N(J):PRINT'
D(J,K):NEXT
  130 PRINT'" ":PRINT'"ALL OK? (Y/N)"
  140 REPEAT:KEY$=GET$:UNTIL KEY$="Y" OR
 KEY$="N"
  150 IF KEY$="N" THEN 70
  160 NEXT
  170 FOR J=1 TO N(1):S(J)=D(1,J):R(J)=0
:NEXT
  180 FOR J=1 TO N(2):S(J+N(1))=D(2,J):R
(J+N(1))=1:NEXT
  190 CLS:PRINT'''TAB(7)"COMPUTING MAY T
AKE A WHILE"
  200 NT=N(1)+N(2)
  210 PROCquicksort(1,NT)
  220 TS=1:TT=0:RS=0:RF=0
  230 IF TS>NT THEN 390
  240 IF S(TS)<>S(TS+1) THEN TS=TS+1:GOT
O 230
  250 TF=TS+1
```

```
  260 IF TF>NT THEN 390
  270 IF S(TF)=S(TF+1) THEN TF=TF+1:GOTO
 260
  280 NZ=0
  290 FOR J=TS TO TF
  300 IF R(J)=0 THEN NZ=NZ+1
  310 NEXT
  320 IF NZ=0 OR NZ=TF-TS+1 THEN TS=TF+1
:GOTO 230
  330 RS=TS:RF=TF
  340 FOR J=TS TO TS+NZ-1:R(J)=FT:NEXT
  350 IF FT=0 THEN FT=1:GOTO 370
  360 FT=0
  370 FOR J=TS+NZ TO TF:R(J)=FT:NEXT
  380 TS=TF+1:GOTO 230
  390 IF FT=0 OR RS=0 THEN 460
  400 IF RND(1)<.5 THEN 460
  410 NZ=0:FOR J=RS TO RF
  420 IF R(J)=0 THEN NZ=NZ+1
  430 NEXT
  440 FOR J=RS TO RF-NZ:R(J)=1:NEXT
  450 FOR J=RF-NZ+1 TO RF:R(J)=0:NEXT
  460 R(NT+1)=R(NT)+1
  470 U=0:FOR J=1 TO NT
  480 IF R(J)<>R(J+1) THEN U=U+1
  490 NEXT
  500 UU=2*N(1)*N(2)/(N(1)+N(2))+1
  510 S=SQR(2*N(1)*N(2)*(2*N(1)*N(2)-N(1
)-N(2))/(N(1)+N(2))^2/(N(1)+N(2)+1))
  520 Z=(ABS(UU-U)-.5)/S
  530 IF Z<0 THEN Z=.001
  540 IF N(1)*N(2)<18 OR N(1)<5 OR N(2)<
5 THEN PRINT'"NOT ENOUGH DATA":GOTO 740
  550 R=0
  560 R=R+1:READ ZL
  570 IF ZL=Z THEN 590
  580 IF ZL>Z THEN 560
  590 IF R<10 THEN P=R/10
  600 IF R>9 AND R<19 THEN P=R-9
  610 IF R>18 THEN P=10+5*(R-19)
  620 IF ZL<Z THEN 670
  630 CLS:PRINT'"THERE IS A ";100-P;"%
CHANCE THAT THE"
  640 PRINT'N$(1);" DATA DIFFERS FROM TH
```

```
E "
  650 PRINT'N$(2);" DATA"
  660 GOTO 730
  670 IF R<11 THEN PL=(R-1)/10
  680 IF R>10 AND R<20 THEN PL=R-10
  690 IF R>19 THEN PL=10+5*(R-20)
  700 CLS:PRINT''"THERE IS A CHANCE BETW
EEN ";100-P;"% AND "
  710 PRINT';100-PL;"% THAT THE "+N$(1);
" DATA";
  720 PRINT''"DIFFERS FROM THE "N$(2)" D
ATA"
  730 PRINT''''''TAB(3)"< SPACE BAR FOR A
NOTHER ANALYSIS >"
  740 REPEAT:KEY$=GET$:UNTIL KEY$=" "
  750 RESTORE:GOTO 40
  760 DATA 3.085,2.88,2.75,2.65,2.575,2.
51,2.455,2.41,2.365
  770 DATA 2.327,2.052,1.881,1.751,1.645
,1.555,1.476,1.405
  780 DATA 1.341,1.282,1.037,.842,.675,.
5224,.385,.253,.126,0
  790 DEF PROCquicksort(P%,R%)
  800 LOCAL I%,J%,V,W,X
  810 I%=P%:J%=R%:X=S((P%+R%)DIV2)
  820 REPEAT
  830 IF S(I%)<X I%=I%+1:GOTO 830
  840 IF X<S(J%) J%=J%-1:GOTO 840
  850 IF I%<=J% W=S(I%):S(I%)=S(J%):S(J%
)=W:V=R(I%):R(I%)=R(J%):R(J%)=V:I%=I%+1:
J%=J%-1
  860 UNTIL I%>J%
  870 IF P%<J% PROCquicksort(P%,J%)
  880 IF I%<R% PROCquicksort(I%,R%)
  890 ENDPROC
```