DECUS
PROGRAM LIBRARY

| | |
|---|---|
| DECUS NO. | FOCAL8-170 |
| TITLE | SAINT PETER'S COLLEGE STATISTICAL PACKAGE |
| AUTHOR | Professor Robert W. Carter |
| COMPANY | Saint Peter's College<br>Jersey City, New Jersey |
| DATE | April 12, 1971 |
| SOURCE LANGUAGE | FOCAL-69 |

## ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

# FLGPLT

DECUS Program Library Write-up DECUS NO. FOCAL8-170.1

The present routine is an extension and combination of two earlier routines, FLHSTO and FLPCTL. FLGPLT, in contrast to FLHSTO, requires that data be input in frequency table format. In comparison with FLPCTL, FLGPLT operates on the input data to produce a grouped frequency distribution, which is subsequently plotted.

Perhaps the major difference between FLGPLT and FLHSTO is that the output from FLGPLT will be "neat" and in the traditional format. Cells are plotted even if their frequencies are zero, and data values are treated as if from an interval scale of measurement. In contrast, FLHSTO treats data as coming from a nominal scale, and FLHSTO does not group the frequencies.

FLGPLT begins by requesting input of the total frequency sum, which will generally be supplied from a previous pass with FLHSTO. The input total is later used in a checksum which compares the stated and actual input frequencies. A second request is then made for input of the cell width and maximum midpoint value for grouping of the data cells. See FLPCTL for a description of the grouping rules. In essence, however, the cell width should divide the score range into from 10 to 20 units, and the maximum midpoint can be a score at or slightly above the largest actual score.

With these input requests satisfied, the routine proceeds to request data input with the format: X ( the score ), F ( the score frequency ). Data are collected with a "tight" loop for flexibility and for independent computation of the checksum. After all data have been entered, the tight loop is halted with a control/c combination and control transfered to statement 3.1 for the plot.

FLGPLT does not use the extended functions in FOCAL, and these functions should be deleted to maximize data space.

The Binomial distribution is concerned with the results of collections of "Bernoulli" trials, where such trials have the following properties:

1. The number of trials, "N," must be fixed in advance of execution.

2. Each trial must have only 2 possible outcomes. Examples are coin tossing (Head,Tail) and tests (Pass,Fail).

3. The probability, or relative frequency, (referred to as "p") of the first kind of trial outcome must be constant or stationary over trials.

4. Trials must be mutually independent.

In describing the results of a "Binomial" experiment having N Bernoulli trials, it is convenient to define a "random" variable, "X," where X is the count of the number of outcomes of the first kind in N total trials. A Binomial experiment, in general, always has a total of $2^N$ possible outcomes or sample points, but X always has only N+1 possible values. As a result of X's smaller number of values, X tends to reduce the amount of information needed to describe a given experiment.

The description usually needed for a Binomial experiment is actually a function in which the "domain" is composed of the values which X can take on and which the "range" is composed of the probabilities of those values of X. These probabilities can be computed with the rule:

$$P(X=k) = \frac{N!}{(k!)\cdot(N-k)!}\, p^k q^{N-k} \text{ , where:}$$

p=probability of 1st kind of trial outcome
q=1-p
N=Number of trials
N!=(N)(N-1)....(1)
k=some value of X or Ø,1,....N
(Note that Ø!=1)

The present routine computes Binomial probability functions or distributions with the rule above. To eliminate the complications of computing factorial products, a strategy was chosen in which the natural logs (FLOG) of the probabilities are computed and followed by the computation of antilogs to get actual answers. This strategy replaces products with sums and generally reduces computing time. The accuracy of this strategy within FOCAL was found to be quite adequeate for most problems.

FLBIND begins by requesting input of "N," the number of trials. After input of N, a request is made for input of "p," the probability of the first kind of outcome. With these critical parameters in, a third request is made for input of whether or not a full distribution is needed. If the answer to this query is "1" for yes, then the entire probability function is immediately computed and displayed. If the answer to this query is "2" for no, then a fourth request is made for input of "k," a single value of X, for which only 1 probability will be computed.

After all necessary probabilities have been computed and displayed, FLBIND reinitializes and requests another value of N.

# FLPCTL

## A FOCAL ROUTINE TO COMPUTE PERCENTILE SCORES

A "percentile" score is defined as that score which cumulatively exceeds a certain percentage of a distribution of all scores. Percentile scores are very versatile measures and can be used to answer at least three basic statistical questions:

i) What score is typical of the distribution?

ii) What noise is present?

iii) How does one score relate to all the others?

In addition, percentile scores are easily understood by people who have not had statistical training.

The present routine, FLPCTL, accepts input data from the frequency table output of FLHSTO, a general-purpose FOCAL histogram routine. Useage of FLPCTL requires a judgement, which is based on FLHSTO, on how "raw" scores should be grouped in order to compute cumulative frequencies. Specifically, the user of FLPCTL must supply the cell ("class" or "bin") width, the top or largest cell midpoint value and the sum of all the frequencies to be accepted for analysis. Supplying these specifications to FLPCTL is simplified by looking at the frequency table supplied by FLHSTO.

A reasonable rule of thumb for the judgements of cell width and maximum cell midpoint is that there should be about 10 to 20 cells with an odd cell width (3,5,7,9 etc.) Suppose that the frequency table from FLHSTO reveals a score range of about 0 to 100. A reasonable cell width would then be about (100- 0)/20 = 5.0 If the cell width is chosen as an odd number, then cell midpoints will be round numbers spaced a cell width apart. If the largest score within the range above was actually 95, then the choice for the maximum midpoint could be 93 or larger for a cell width of 5.0. In general, the top cell midpoint must allow coverage of the largest scores. FLPCTL will, however, diagnose an improper choice for the maximum midpoint which will generally call for an increase in the value supplied.

After receiving input values for the cell width, the maximum midpoint, and the frequency sum, FLPCTL enters a "tight" loop in which cell frequencies are tallied by means of a pushdown list. Speed is optimized if the user enters data values from the largest midpoint downward, as supplied by FLHSTO.

After all data values have been entered in pairs of (score, frequency), an exit from the tight loop is made with a control/c input combination. A display of the results can then be made by a direct transfer of control to statement 3.1 (i.e. G 3.1), which causes cell statistics to be computed and displayed. The display concludes with results for the 3 quartile scores, i.e. the 25,50 and 75 percentiles. Storage is optimized in the display phase by overlaying frequencies with cumulative frequencies when computational requirements allow.

Data from non-numerical measurement can be numerically coded (e.g. for "grade-point" data) for analysis with FLPCTL, but the quartiles are computed by linear interpolation which assumes actual numerical data are from an interval scale or better. No extended functions are used in this routine. When FLPCTL completes the display of results, it loops back to the initialization instructions which begin the routine and waits for new input.

# FLSDEV

## A FOCAL ROUTINE TO COMPUTE MEANS & RELATED MEASURES

The present routine computes a basic set of descriptive statistics based upon_the arithmetic mean or "average." The arithmetic mean (X) is simply the sum of a set of scores divided by the count of the set. The scores, of course, must be from at least an interval scale of measurement. In essence, X gives the balance point for a distribution, somewhat in analogy with a center of mass or "gravity," and provides a measure of the typical score.

Although $\bar{X}$ is a very useful measure, it has the major drawback of sizeable sensitivity to extreme scores, thus giving distorted results in distributions that are skewed, or asymmetrical. In an attempt at overcoming the distortion possible with X, the present routine also computes two other useful measures of the typical score, the geometric mean (G.M.) and the harmonic mean (H.M.).

The G.M. provides a measure of the typical score after the data have been subjected to a log transformation. In essence, the G.M. is the antilog of the arithmetic mean of the logs of scores. Since logs are computed for each original datum, data must be only nonzero and positive values.

The H.M. provides a measure of the typical score after the data have been subjected to a reciprocal (exponent of -1) transformation. The H.M. is the reciprocal of the arithmetic mean of the reciprocals of scores.

After computing the means above, the present routine concludes by computing the variance($S_x^2$) and standard deviation ($S_x$) for the data set involved. The variance provides a measure of noise ( or variability or spread) by computing the arithmetic mean of the squared distance from each score to the arithmetic mean of the scores. As noise in a set of scores increases, the variance also increases. A large family of techniques, generally known as the Analysis of Variance (ANOVA), is based upon the variance measure and has the objective of partitioning the effects of experiments and related processes.

The $S_x$ is simply the square root of the $S_x^2$. The notation "$S_x$" is used to speak of the standard deviation of the variable "X" in a context having multiple possible variables. $S_x$ is a very useful measure and often used as the unit for noise measurement since it retains the original unit of measurement while the $S_x^2$ retains the unit squared. As a rule of "thumb," all of the scores in a distribution generally fall between $\overline{X} \pm 3S_x$, while about 50% of a distribution generally falls between $\overline{X} \pm S_x$ .

The present routine starts by requesting input of the count (N) of the set of scores. The routine assumes prior useage of FLHSTO, a routine which supplies N, among other things. After N has been supplied, data are then entered in the format: X,f, where X is a score or datum and f is its frequency or count. The X,f values are also supplied as output by FLHSTO.

Data values are entered with a "tight" loop within FLSDEV, in order to compute and independent frequency checksum. The tight loop is halted after all data have been entered by typing a control/c combination. Cpntrol should then be transferred to statement 3.1 (e.g. with a G 3.1) for the checksum comparison and the display of results.

# FLHMES

## A FOCAL ROUTINE TO COMPUTE "H," THE INFORMATION MEASURE OF NOISE

Data values resulting from measurement with nominal scales simply reflect accurately named objects or events. Since all scales must be at least "nominal" in construction, techniques developed for the nominal scale are universally applicable, although not necessarily universally appropriate.

The standard technique for measuring the uncertainty or "noise" in a set of data values obtained with a nominal scale from one variable is to compute "H," the average number of binary questions needed to locate any given score within an overall distribution of scores.

The present routine is designed to operate on the frequency table output from FLHSTO, a general-purpose FOCAL histogram routine. With the frequency table available, execution of FLHMES begins with a request for user input of the total sum of all frequencies as supplied by FLHSTO. Next, the frequency of each score is gathered from the user with a "tight" loop. After all frequencies have been entered, the tight loop is halted with a control/c input combination. Control should be then transferred (eg. G 3.1) directly to statement 3.1.

Before the measure "H" is displayed, a checksum is computed between the total frequency sum first given as input and the frequency sum computed by the tight gathering loop. If these sums agree, then the computed value of "H" is displayed and control transferred to restart the routine for another data set.

FLHMES usesthe FOCAL log function.

# FLTMES

## A FOCAL ROUTINE TO COMPUTE "T", THE INFORMATION MEASURE OF RELATIONSHIP

The present routine computes the measure "T" which describes the extent to which two nominally scaled variables are related. Computation of "T" actually calls for repeated computation of "H," the measure of noise. As a result, the present routine uses a related routine, FLHMES, as a subroutine. In brief, "T" is the difference between the noise in a 2 variable data set before and after useage of the relationship, if any, between the variables.

FLTMES starts by requesting user input of the number of score possibilities (or columns) for the independent variable (X), followed by the total sum of frequencies for the dependent variable (Y). The "original" uncertainty in the dependent variable is next computed after user input for row (Y) frequency sums. After all row sums have been entered, the routine is halted with a control/c input combination and control then transferred directly to statement 1.5 (e.g. G 1.5).

The "residual" uncertainty beyond a relationship is next computed with user input of the score frequencies in each column of the independent variable. After the frequencies have been entered for a column, processing must be stopped with a control/c combination and control transferred directly to statement 1.7 (e.g. with a G 1.7). When each and every column has been processed in this way, control should then be transferred directly to statement 1.9 (e.g. G 1.9). A display then results for the original and residual uncertainty measures and "T". Checksums are computed for the frequencies entered, and faulty data will be diagnosed without the faulty display of "T".

This routine uses the FOCAL log function.

# FLPEAR

## A FOCAL ROUTINE TO COMPUTE A PEARSON LINEAR CORRELATION & REGRESSION ANALYSIS

FLPEAR USES THE FOCAL SQUARE ROOT FUNCTION.

When data values come from interval or ratio scales of measurement, one is frequently in need of measures of linear correlation and regression which state the extent, direction and form of relationship. Although approximate methods, such as the method of averages or fitting by "eye," are sometimes useful, the technique due to Pearson using a least squares solution is most appropriate when computing machinery is available.

The present routine stands alone and computes the count, arithmetic means, covariance, Pearson coefficient, regression slopes, and best-fitting straight lines for a data set of paired values.

After starting execution, the routine simply asks for user input of the data pairs in the form: (independent value (X), dependent value (Y)). Input proceeds within a "tight" loop which allows processing of arbitrary counts and which is halted with a control/c input combination after all data have been supplied.

After the control/c combination, control should be transferred to statement 3.1 to produce a display of:

1. Count of points (N).
2. Arithmetic mean of x.
3. Arithmetic mean of y.
4. Covariance.
5. Pearson R.
6. Slope of line to predict y from x.
7. Slope of line to predict x from y.
8. Line to predict y from x.
9. Line to predict x from y.

# FLSPER

## A FOCAL ROUTINE TO COMPUTE SPEARMAN'S RANK-ORDER CORRELATION COEFFICIENT

If two ordinally-scaled variables are constructed from pairs of ranks, then one can measure the extent and direction of relationship between their ranks with a special case of the Pearson coefficient which is due to Spearman. Spearman's coefficient falls between the limits of +1 and -1. A zero value for the coefficient implys that no evidence of a linear relationship between the ranks has been detected. Values of 1 imply, of course, a perfect linear relationship between the ranks.

The present routine does not use other routines and begins by requesting user input of the number of rank pairs. Next, a "tight" loop is entered which requests user input of each pair of ranks in turn. One variable is assumed to be always on the left of a rank pair with the other variable always on the right.

After all pairs of ranks have been supplied, the tight loop is halted with a control/c combination, and control should then be transferred (e.g. G 3.1) directly to statement 3.1.

A checksum is made between the original pair count and the computed pair count within the gathering loop. If the sums match, then the coefficient is displayed and control returned to initialize the routine for a new data set. No extended functions are used within the routine.