UNIVERSITY OF QUEENSLAND
COMPUTER CENTRE

COMPUTER

CENTRE

BULLETIN

NEWSLETTERS AND BULLETINS

The Weekly Newsletters were first produced because of the need to provide
accurate up-to-date information as soon as possible. The monthly Bulletin;
because of the time involved in editing and collating of information,
preparation of masters and actual printing; could not possibly produce
information that was completely up-to-date. Although the content of the
Newsletters is consolidated and included in each Bulletin this time lag is
still apparent. Users may receive a copy of a Bulletin detailing an error
that has been reported as fixed in a Newsletter of a later date. For this
reason, in this and future Bulletins, any item extracted from a Newsletter
will be identified by number of that Newsletter. In this way users may check
on the date of the item in the Bulletin and any subsequent Newsletters that
have appeared, in order to be aware of the true state of events.

CHRISTMAS SHUT DOWN

The Computer Centre will shut down on Christmas Eve and remain closed during
the intervening days between Christmas and New Year. No further work will be
processed after 11.30 p.m. on Thursday 23 December. However, the Centre will
remain open until 1 p.m. on Friday 24 December to allow staff to finalize
end-of-year accounting and users to collect any work or card punching. The
Centre will re-open at 9 a.m. on Tuesday 4 January 1972.

EXTENSION OF CONSULTING HOURS

[WN-59,60]

The hours available for programmer consultations have been extended. The
new hours are:

Monday	2.00 p.m. - 4.30 p.m.
Tuesday	2.00 p.m. - 4.30 p.m.
Wednesday	10.30 a.m. - 1.00 p.m.
Thursday	2.00 p.m. - 4.30 p.m.
Friday	10.30 a.m. - 1.00 p.m.

While prior appointments for a consultation are not necessary during these
hours, preference is given to clients who make appointments in advance.
Note that, as previously, requests for consultation must be made through the
Administrative Officer, Mr John Jauncey. Similarly, phone enquiries on
programming problems must be directed to Mr Jauncey, extension 8471.

The member of staff available for consultation has other responsibilities for the smooth running of the system which will take priority over his consulting functions. These arrangements have been made in part to reduce the number of interruptions to staff members who have a very heavy commitment to the developmental work of the Centre. Consequently, consultation will not be given outside the above hours. Any direct approaches to programmers or requests for consultation outside the advertised times will be refused.

COMPUTER CENTRE HARDWARE DEVELOPMENT

The Computer Centre is currently engaged in a number of modifications to existing equipment, and is planning development of new hardware.

Nearing completion are modifications to the line printer to provide automatic page counting, and a facility for bursting forms online. The burst facility will be under program control, and will alleviate the onerous task of separating output, as approximately 300 printer files are produced each day.

A hardware character decoder has been designed for the card reader, which will substantially reduce the amount of processor time previously required for software decoding. A common facility for determining the status of the card reader and printer has also been installed to simplify symbiont operation.

Design is proceeding on a general purpose communications interface that will allow devices of various characteristics to be connected to the system. Investigations into methods of improving the performance of the fixed head disks used for swapping are currently being carried out.

A very serious problem of transistor lead failure due to corrosion has been investigated, resulting in the design of adequate filters throughout the system.

NEW SYSTEMS PROGRAMS

(a) New Login, Logout and Decode

[WN-55]

In conjunction with the release of the new version of Batch on 9 September, new versions of Login, Logout, Loader and Decode were implemented. As well as incorporating changes made necessary by the introduction of the new Batch, these programs were changed in order to speed up their operation, and to provide improved error messages.

Two of the changes to Decode need to be mentioned specifically:

(i)   RUN (with no arguments) will not attempt to execute, if there were errors in compilation.

(ii)   LOG PJ =1 (for example) gave an 'ILLEGAL ASSIGNMENT' because of the space before the equals. This has been fixed.

A further change will be made to Logout as soon as possible. If the user's logged out limit is exceeded while logging out in Batch, the user's file directory will be listed once only before any files are deleted. At present, the directory is listed three times.

(b) <u>New Monitor</u>

[WN-59]

A new monitor became operational on Thursday 23 September. The changes incorporated do not affect the user directly, but have improved the handling of error situations.

COPY THROUGH BATCH

There appears to be some confusion about the use of the COPY command to produce disk files from cards.

Batch automatically deletes trailing blanks to make records shorter and processing faster, so that the effect of a COPY command through Batch would be to produce a file with no trailing blanks.

However COPY with the COMPRESS option is slightly different, and is intended for copying FORTRAN and MACRO program files. The COMPRESS option changes the contents of columns 73-80 (sequence numbers) to spaces and then deletes trailing blanks.

Users loading data cards containing useful information in columns 73-80 should not use the COMPRESS option.

COMMAND LANGUAGE SPECIFICATIONS

When running a program from an area other than your own, for example in Batch, a card of the form

        .256.filename

should not contain any spaces. At present, the following problem will occur if spaces are included within the verb. The Command Decoder, when scanning and interpreting a card such as

        .256Δ.Δfilename

will assume that the verb is 256.filename but it sets its pointer at the first non-blank character it finds, in this case

        .256Δ.Δfilename
           ↑

The result is that the correct file is loaded for execution but the rest of the command is incorrectly taken to be the first input data record.

OVERLAY COMMAND ERROR

The command

        OVERLAY(∅) filename

gives the message 'OVERLAY SUBROUTINE NOT FOUND' at loading. To overcome this, users should compile the main program first, and thus have it on the loadlist when the first overlay command is given. This error will be fixed shortly.

USE OF REWIND, ENDFILE AND CALL RELEASE STATEMENTS

With the increased use of disk files, many users have reported getting the monitor message,

        ERROR IN JOB n
        ADDRESS CHECK FOR DEVICE DSK;   UUO AT USER location

This has in some cases been traced to excessive use of CALL RELEAS (n) statements which release the device but do not reclaim the memory locations used for buffers.

ENDFILE n statements should be used if the assignment of a filename to the logical device is not to be broken.

If another file is to be accessed on the logical unit, the programmer should use REWIND n.

For further details on the REWIND n and ENDFILE n statements, see the FORTRAN manual, MNT-5, and the Computer Centre Bulletin Vol.4, p.122.

PDP-10 FORTRAN

(a) When attempting to skip records in a non-formatted disk file, it is necessary to include a dummy variable in the write statement, thus

```
READ(1Ø) X is satisfactory but
READ(1Ø) is not.
```

(b) The following FORTRAN statements do not compile correctly under version 23 of the FORTRAN compiler.

```
13ØØ U(I,J)=U(I,J)+L(K,I,1)*L(K,J,2)+L(K,I,2)*L(K,J,1)
14ØØ B=B+2.Ø*(LM(I,J,1)*H(J,I,1)+LM(I,J,2)*H(J,I,2))+U(I,J)*
    1(H(J,I,1)+H(J,I,2))
```

While the FORTRAN manual does not explicitly state any limit to the length or complexity of a statement, users should attempt to avoid unnecessarily complicated statements. Failure to compile an expression usually occurs when it is very long or there are a number of subexpressions or subscript calculations to be performed.

NEW FORTRAN OPERATING SYSTEM

Version 30.2 of the FORTRAN operating system (LIB4Ø) was released on Monday 27 September. The changes made in this version were primarily to correct errors reported following the initial release of the previous version, but a number of other improvements have been made, principally in the area of input/output control.

1.   IMPROVEMENT AND RATIONALIZATION OF INPUT/OUTPUT ERROR REPORTING

This affects the reporting of most input/output errors. For reference, the previous message as given in Vol.4 Nos 8 and 9 p.117 of the Bulletin, is quoted in parentheses.

If formatted input/output was in progress, each of these messages will be
followed by the format statement (or the first 72 characters in the case of a
large format statement). If it was found that the memory locations specified
as the format statement did not contain a legitimate character string, then the
message

FORMAT AT adr IS NOT A VALID CHARACTER STRING

will be output. Following this message, a line of the form

LAST FORTRAN I/O AT USER LOC adr

will be output.

DEVICE dev NOT AVAILABLE - LOGICAL UNIT n

(DEVICE dev: NOT AVAILABLE)

> The operating system tried to initialize a device that either does not
> exist or has been assigned to another job.

END OF FILE, FILE name/ext DEVICE dev LOGICAL UNIT n

(END OF FILE ON dev)

> A premature end-of-file has occurred on an input device.

Note that in the above and subsequent messages, a filename will be quoted only
if it is relevant to the particular device. Further, it is not possible to
give the extension part of the filename in all circumstances and in such cases
the /ext will be omitted.

CANNOT FIND FILE name/ext DEVICE dev LOGICAL UNIT n

(FILE NAME filename NOT ON DEVICE dev)

> The file cannot be found in the directory of the device that was
> accessed as logical unit n.

INPUT DEVICE ERROR, FILE name/ext DEVICE dev LOGICAL UNIT n

(INPUT DEVICE ERROR ON dev)

> A data transmission error has been detected during input from the
> device accessed as logical unit n.

NO ROOM FOR FILE name/ext DEVICE dev LOGICAL UNIT n

(NO ROOM FOR FILE filename ON DEVICE dev)

> There is no room for the file in the directory of the device accessed
> as logical unit n.

OUTPUT DEVICE ERROR, FILE name/ext DEVICE dev LOGICAL UNIT n

(OUTPUT DEVICE ERROR ON dev)

>   A data transmission error has been detected during output to the device
>   accessed as logical unit n.

PARITY ERROR, FILE name/ext DEVICE dev LOGICAL UNIT n

(PARITY ERROR ON dev)

>   A parity error has been detected.

WRITE FAILURE FOR FILE name/ext DEVICE dev LOGICAL UNIT n

(dev: WRITE PROTECTED)

>   The device is WRITE locked. This error can occur if a binary write
>   operation is given after a read on the same logical unit.

LOGICAL UNIT n IS ILLEGAL

(DEVICE NUMBER n IS ILLEGAL)

>   A non-existent unit number was selected.

In addition, the following messages that were not listed previously could
occur under exceptional circumstances and are reported to complete the list
of possible errors from the FORTRAN operating system.

RECORD TOO SHORT ON FILE name/ext DEVICE dev LOGICAL UNIT n

(TAPE RECORD TOO SHORT ON UNIT dev)

>   The data list of a binary read operation exceeds the size of the
>   logical record to be read.

BAD PHYSICAL RECORD COUNT, FILE name/ext DEVICE dev LOGICAL UNIT n

(ILLEGAL PHYSICAL RECORD COUNT, TAPE dev)

>   There is an inconsistency in the format of the physical records of
>   a binary logical record comprising more than 1 physical record.

END OF MEDIUM ON FILE name/ext DEVICE dev LOGICAL UNIT n

(END OF TAPE ON dev)

>   This message would normally only occur when writing magnetic tapes
>   and in this situation indicates end-of-tape.

ILLEGAL OPERATION, FILE name/ext DEVICE dev LOGICAL UNIT n

(ILLEGAL MAGNETIC TAPE OPERATION, TAPE dev)

> A non-permitted sequence of operations was attempted on the device.

?CANNOT FIND FOR302/SHR

> The sharable FORTRAN operating system segment is not capable of being accessed.

?FOR302/SHR DOES NOT CORRESPOND TO IMP40

> There is a problem in the organization of the System directory.

2.   PRINTER CONTROL CHARACTERS

Additional characters can now be used for control of printer positioning on output. Previously only the characters +, blank, Ø, and 1 were correctly acted upon. The following characters now have the action indicated:

+	do not advance
1	preslew to VFU channel 1 - Top-of-form
2	preslew to VFU channel 2 - Bottom-of-form
3	preslew to VFU channel 3 - Half of form
4	preslew to VFU channel 4 - Third of form
5	preslew to VFU channel 5 - Sixth of form
6	preslew to VFU channel 6 - Single spacing over the whole page
7	preslew to VFU channel 7 - Top-of-page
-	preslew 3 lines*
Ø	preslew 2 lines*
blank	preslew 1 line*
(or any other character)	

* These three operations are obtained by repeated slewing to VFU Channel 8, continuing onto the next form if necessary.

The VFU (Vertical Format Unit) is controlled by a special tape, which has positions corresponding to each line in a page. As normally setup, a *page* is assumed to have 66 lines. Within a page, a *form* of 60 lines is centrally located. Within the page there will thus be a margin of 3 lines, a form of 60 lines, and a further margin of 3 lines. The VFU loop or tape has the following punchings:

```
Channel 1    -    line 4
Channel 2    -    line 61
Channel 3    -    lines 4 and 34
Channel 4    -    lines 4, 24 and 44
Channel 5    -    lines 4, 14, 24, 34, 44 and 54
Channel 6    -    all lines throughout the page
Channel 7    -    line 1
Channel 8    -    all lines between 4 and 63
                  (this channel is used for the -, ∅, and blank printer
                   control characters).
```

Users should be aware that the only printer control characters defined for
standard FORTRAN by the American Standards Institute are +, blank, ∅ and 1.
The other control characters defined above are additional features for this
installation and apply only to the line printer.

3.   PLOTTING ROUTINES

Several changes have been made to the plotting routines, mainly to improve
the handling of error conditions. No changes to functional programs should
be necessary.

(a)   An attempt to run a program that does not contain a call to PLOTI will
      result in any other plotting routines used being undefined at loading
      and execution will be deleted.

(b)   If a pen movement call is given after a plot file is closed, the message
      'NO PLOT FILE OPEN' is given. However, to prevent a large number of
      meaningless messages of this type, pen movement calls are merely ignored
      if they occur after the user program has regained control on an error.

(c)   After a 'COORDINATE OUT OF BOUNDS' error, the user program may now
      determine the coordinates of the offending point by calling WHERE,
      provided, of course, that the call to PLOTI included an error return
      statement number.

(d)   A call to SCALE with the number of points less than one will result in
      the message 'TOO FEW POINTS TO SCALE' instead of the less explicit
      'FLOATING DIVIDE CHECK' and other messages that currently result from
      this condition. The results AMIN, and DA are set to zero and
      processing continues.

4.   ERRORS CORRECTED

(a)   DEFINE FILE and CALL OFILE statements often produced files with
      extensions containing random characters. Both have been fixed. (See
      Weekly Newsletter No 53).

(b)   Loss of a character when inputting an ASCII string between single quotes
      no longer occurs. (See Weekly Newsletter No 53).

(c) The 'Illegal character X, in format' now gives the offending character correctly.

(d) Another attempt has been made to cure the problems caused by Batch removing the trailing blanks from data cards.

(e) If a program running under Batch uses PAUSE, or has a source level error, the message 'TYPE G TO CONTINUE, X TO EXIT' is not given. Instead the program will exit immediately.

(f) In Batch, a card with data up to column 78, with columns 79 and 80 blank caused the next input to have an extra leading blank. This problem has been fixed. (See Weekly Newsletter No 53).

(g) All control characters, except carriage return, line feed, control-Z and tab are now ignored. The tab character is treated as a space, control-Z is the end of file character for terminals, and the carriage return and line feed are record separating characters. The file separator is now detected in all cases as the end of file character for Batch. (See Weekly Newsletter No 53).

(h) As reported in Weekly Newsletter No 52 3(f) a repeated part of a format specification that did not use an item from the output list caused one to be ignored. This has been fixed so that

$$(2(/), 5F4.\emptyset)$$

will now work correctly.

LIBRARY ACCESSIONS

GT BRIT. Ministry of Public Building and Works. Directorate of Research and Information

Computer-aided architectural design 1969 (Qto GRE Arch.)

OETTINGER, Anthony G.

Run, computer, run; the mythology of educational innovations 1969 (LB1027.O32 Main)

ACM SYMPOSIUM ON THE THEORY OF COMPUTING, 2nd Northampton 1970

Conference record 1970 (Qto 510.7834 ACM Engin.)

COMPUTER USAGE COMPANY

Computer usage c1970 (QA76.C5738 Main)

GRUENBERGER, F.J. ed.

Fourth generation computers 1970 (QA76.5.G77 Main)

SYMPOSIUM ON THE APPLICATION OF COMPUTERS TO THE PROBLEMS OF URBAN SOCIETY
4th New York 1969

Papers   1969   (Qto 301.36 SYM Engin.)

BRINK, V.Z.
Computers and management   1971
(HF5548.2.B759 Main)

GALLER, B.A.
A view of programming languages   1963
(001.6424 GAL Engin.)

JOHNSON, L.R.
*System structure in data, programs, and
computers*   1970   (651.8 JOH Engin.)

KATZAN, Harry
APL programming and computer techniques
1970   (651.8 KAT Engin.)

McCRACKEN, D.D.
A guide to COBOL programming   1970
(Qto 651.26 MACC Engin.)

MALMSTADT, H.V.
Computer logic   1970   (Qto QA76.5.M196 Phys.)

VARIAN DATA MACHINES
Varian data 620/i computer manual   1968
(651.82 VAR Engin.)

WATTERS, J.
Cobol programming   1970   (QA76.5.W37 Main)

OCKENDEN, J.M.
Focus on medical computer development
1970   (610.28 OCK Vet.)

WISP ON THE GE-225

[WN-60]

A new version of WISP has been implemented on the GE-225, which corrects
some errors in the garbage collection scheme of compiler and execution
package. Operational procedures are unaltered.

This new version has an extended set of standard forms and also uses a
special version of MACRO GAP to simplify the definition of standard forms.

MULTIPLE LINEAR REGRESSION

[WN-61]

MLRP-SLIP on the GE-225 has been causing problems to a number of clients.
It appears that BMD$\emptyset$2R is a more satisfactory program for multiple linear
regressions. BMD$\emptyset$2R runs on the PDP-10 and can analyze data files created
at a remote terminal or input through Batch.

Several BMD programs have been released for general use.

1.   INTRODUCTION

Many problems require extensive analysis of large amounts of recorded data.
As far as possible, the data handling processes should be made automatic and
rapid and the Biomedical Computer Program (BMD) Series has been written for
that purpose. These programs are provided in a 'package' form that enables
users to effect their desired computations with simple coded instructions.
The data is usually presented in a standard form, so that a wide variety of
analyses may be carried out by specifying different parameters to various
programs.

A number of these statistical programs, which were developed at the Health
Sciences Computing Facility, UCLA, are now available on the PDP-10. The
programs are batch-oriented but can be used from a terminal if great care is
taken with data formats.

2.   WRITE-UPS

The original writeups published in the BMD Manual, UCLA Press, 1968 have had
to be modified slightly to conform with certain PDP-10 restrictions and to
allow input from files as well as from card decks. These writeups each rely
in some measure on the introductory descriptive material in the early chapters
of the BMD manual. No attempt has been made here to provide a complete
writeup for each separate program. Instead the user has been referred to the
appropriate section in the BMD Manual and given the necessary changes required
for the program to run on the PDP-10.

3.   BMD MANUAL

The BMD manual 'Biomedical Computer Programs' is edited by W.J. Dixon. It is
the second in a series of University of California Publications in Automatic
Computation and is published by the University of California Press. There is
a copy of the manual in the Engineering Library, and copies also exist at the
Medical School in the following libraries; Social and Preventive Medicine,
Microbiology, and Clinical Sciences.

4.   RELEASE AND RUNNING OF PROGRAMS

BMD programs will be progressively converted and released, according to demand
from users. Initially the following programs are available:

BMD∅1D	Simple Date Description
BMD∅2D	Correlation with Transgeneration
BMD∅8D	Cross Tabulation with Variable Stacking
BMD∅1M	Principal Component Analysis
BMD∅4M	Discriminant Analysis for Two Groups
BMD∅7M	Stepwise Discriminant Analysis
BMD∅2R	Stepwise Regression
BMD∅1V	One Way Analysis of Variance
BMD∅4V	Analysis of Covariance with Multiple Covariates

The programs are invoked by a single command following the .JOB card. The programs are in the $BMD library and each has a filename derived from the last three identifying characters. For example, BMD∅1D has the filename D∅1, BMD∅4M is M∅4, BMD∅2R is R∅2, etc.

The command required to initiate BMD∅2R would therefore be

    $BMD.R∅2

An up-to-date list of BMD programs available will be published from time to time in the Bulletin.

5.   PROGRAM CLASSIFICATION

The BMD programs are classified as Type 4 programs, which means that they are unsupported by the Computer Centre and are made available on a 'user beware' basis. (See Computer Centre Bulletin Vol.4 No 4, p.42). Although the Computer Centre is making these writeups available, no knowledge of the workings of each program is claimed by the Computer Centre personnel.

The programs are not guaranteed to work and, in general, no personal assistance will be given if a user has difficulty with a program.

However, the test data listed in the BMD Manual has been run with each program that has been released, and the results agree to within rounding error.

<div align="center">

BMD∅1D

SIMPLE DATA DESCRIPTION
</div>

This program computes simple averages and measures of dispersion of variables, omitting those values which the user specifies for exclusion from the computations.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 42-48, with the following exceptions:

(a) Blanks

    Blanks are interpreted as zeros so methods 1 and 2 for specifying exclusion of blanks are not available.

<div align="center">159</div>

(b) The Problem Card - PROBLM

   (i)   Cols 69-70   Not used.

   (ii)   Cols 73-74   Output device number, 1∅ for output to a disk file,
                        FOR1∅; otherwise, leave blank for output to the job
                        output device.

   (iii) Cols 75-79   Input filename if data is on disk; otherwise, leave
                        blank for input from the job input device.

BMD∅2D

CORRELATION WITH TRANSGENERATION
(Boolean Selection of Cases)

This program computes simple correlation coefficients, averages and measures
of dispersion on entering variables and/or transgenerated variables from
selected cases whose values for specified variables have a precise logical
relationship in agreement with a specified Boolean expression.

For a writeup of this program, consult 'Biomedical Computer Programs' edited
by W.J. Dixon, pages 49-59, with the following exceptions:

(a) Limitations per problem

   (i) p, the number of original variables $(2 \leqslant p \leqslant 100)$

   (ii) g, the number of variables added to the original set after
          transgeneration $(-98 \leqslant q \leqslant 98)$

   (iii) k, the number of Variable Format Cards $(1 \leqslant k \leqslant 8)$

(b) The Problem Card - PROBLM

   (i)   Cols 13-15   Number of original variables $(2 \leqslant p \leqslant 100)$

   (ii)   Cols 23-26   +q q variables added to the original set after
                        transgeneration $(2 \leqslant p + q \leqslant 100)$

   (iii) Cols 33-34   Not used

   (iv)   Cols 69-70   Not used

   (v)   Cols 71-72   Number of Variable Format Cards $(1 \leqslant k \leqslant 8)$

   (vi)   Cols 73-74   Output device number, 1∅ for output to a disk file,
                        FOR1∅; otherwise, leave blank for output to the job
                        output device.

   (vii) Cols 75-79   Input filename if data is on disk; otherwise, leave
                        blank for input from the job input device.

(c) Plot Selection Card - PLOTSL

   (i)   Cols 10-11   Number of variables to be cross-plotted with this
                        base variable $(\leqslant 10)$.

160

BMD⌀8D

CROSS-TABULATION WITH VARIABLE STACKING

This program computes two-way frequency tables of data input. Frequency
tables are computed from specified ranges of the original variables,
variables after transgeneration, stacked variables, or combinations of
these. Data input may be positive or negative integers only. The program
will not accept data input fields that have a punched decimal point.

For a writeup of this program, consult 'Biomedical Computer Programs'
edited by W.J. Dixon, pages 109-119, with the following exceptions.

(a) Limitations per problem

    (i)   The limitation on the maximum value of each data item is
          $\pm 1 \times 10^{38}$ approximately.

(b) The Problem Card - PROBLM

    (i)   Cols 23-25   Note that $N(p+q) \leqslant 8{,}000$, $p+q \leqslant 100$

    (ii)   Cols 73-74   Output device number, 1⌀ for output to a disk file,
                      FOR1⌀; otherwise, leave blank for output to the
                      job output device.

    (iii) Cols 75-79   Input filename if data is on disk; otherwise,
                      leave blank for input from the job input device.

BMD⌀1M

PRINCIPAL COMPONENT ANALYSIS

This program computes the principal components of standardized data and rank
orders each standardized case by the size of each principal component
separately.

For a writeup of this program, consult 'Biomedical Computer Programs'
edited by W.J. Dixon, pages 150-158, with the following exceptions:

(a) Limitations per problem

    (i)   n, number of cases $(3 \leqslant n \leqslant 220)$

(b) The Problem Card - PROBLM

    (i)   Cols 15-17   Number of cases $(3 \leqslant n \leqslant 220)$

    (ii)   Cols 71-72   Number of Variable Format Card(s) $(1 \leqslant k \leqslant 8)$

    (iii) Cols 73-74   Output device number, 1⌀ for output to a disk file,
                      FOR1⌀; otherwise, leave blank for output to the
                      job output device.

    (iv)   Cols 75-79   Input filename if data is on disk; otherwise,
                      leave blank for input from the job input device.

BMD04M

DISCRIMINANT ANALYSIS FOR TWO GROUPS

This program computes a linear function of p variables measured on each individual of two groups. This function can serve as an index for discrimination between the groups. It is determined from the criterion of 'best' in that the difference between the mean indices for the two groups divided by a pooled standard deviation of the indices is maximized.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 185-195, with the following exceptions:

(a) The Problem Card - PROBLM

    (i)   Cols 21-22   Not used.

    (ii)   Cols 73-74   Output device number, 10 for output to a disk file, FOR10; otherwise, leave blank for output to the job output device.

    (iii) Cols 75-79   Input filename if data is on disk; otherwise, leave blank for input from the job input device.

BMD07M

STEPWISE DISCRIMINANT ANALYSIS

This program performs a multiple discriminant analysis in a stepwise manner. At each step one variable is entered into the set of discriminating variables. The variable entered is selected by the first of the following equivalent criteria:

(1) The variable with the largest F value (see computational procedure).

(2) The variable that when partialed on the previously entered variables has the highest multiple correlation with the groups.

(3) The variable that gives the greatest decrease in the ratio of within to total generalized variances.

A variable is deleted if its F value becomes too low. The program also computes canonical correlations and coefficients for canonical variables. It plots the first two canonical variables to give an optimal two-dimensional picture of the dispersion.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 214a-214t, with the following exceptions:

(a) The Problem Card - PROBLM

    (i)   Cols 41-42   Not used

    (ii)   Col 43   Not used

    (iii) Cols 73-74   Output device number, 1∅ for output to a disk file, FOR1∅; otherwise, leave blank for output to the job output device.

    (iv)   Cols 75-79   Input filename if data is on disk; otherwise, leave blank for input from the job input device.

BMD∅2R

STEPWISE REGRESSION

This program computes a sequence of multiple linear regression equations in a stepwise manner. At each step one variable is added to the regression equation. The variable added is the one that makes the greatest reduction in the error sum of squares. Equivalently it is the variable that has highest partial correlation with the dependent variable partialed on the variables that have already been added; and equivalently it is the variable which, if it were added, would have the highest F value. In addition, variables can be forced into the regression equation. Non-forced variables are automatically removed when their F values become too low. Regression equations with or without the regression intercept may be selected.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 233-257d, with the following exceptions:

(a) The Problem Card - PROBLM

    (i)   Cols 73-74   Output device number, 1∅ for output to a disk file, FOR1∅; otherwise, leave blank for output to the job output device.

    (ii)   Cols 75-79   Input filename if data is on disk; otherwise, leave blank for input from the job input device.

BMD∅1V

ANALYSIS OF VARIANCE FOR ONE-WAY DESIGN

This program computes an analysis-of-variance table for one variable of classification, with unequal group sample sizes. Optionally data may be read in from a file.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 488-494, with the following exceptions:

163

(a) Limitations of the problem size

 (i) n_i, the number of observations or cases (sample size) for the ith treatment group or category $(1 \leqslant n_i \leqslant 16{,}000)$

 (ii) N, the total number of observations in all groups or categories combined,
$$N = \Sigma n_i, \quad (N \leqslant 80{,}000{,}000)$$

(b) The problem card - PROBLM

 (i) Cols 64-70 Ignored, all input is in ASCII

 (ii) Cols 73-74 Output device number, $1\emptyset$ for output to a disk file, FOR1$\emptyset$; otherwise, leave blank for output to the job output device.

 (iii) Cols 75-79 Input filename if data is on disk; otherwise, leave blank for input from the job input device.

BMD$\emptyset$4V

ANALYSIS OF COVARIANCE WITH MULTIPLE COVARIATES

This program is designed to compute analysis-of-covariance information for one analysis-of-variance variable with multiple covariates and unequal treatment group sizes. Cases may be specified by the user as being in certain treatment groups, or cases may be placed in treatment groups by the program in accordance with a specified Boolean expression.

For a writeup of this program, consult 'Biomedical Computer Programs' edited by W.J. Dixon, pages 525-542, with the following exceptions:

(a) The Problem Card - PROBLM

 (i) Cols 69-70 Not used

 (ii) Cols 73-74 Output device number, $1\emptyset$ for output to a disk file, FOR1$\emptyset$; otherwise, leave blank for output to the job output device.

 (iii) Cols 75-79 Input filename if data is on disk; otherwise leave blank for input from the job input device.

A.N.U. SEMINAR ON OPTIMIZATION

Each year the Computer Centre of the Australian National University organizes a one day seminar on a topic of general interest. This year's seminar, to be held on Wednesday 8 December, will be on the topic *Optimization*.

Speakers both from within the A.N.U. and from other parts of Australia will join with two prominent speakers from overseas to present papers on a wide selection of topics from the field of optimization. These contributions will range from discussions of theoretical and mathematical aspects of optimization to practical considerations of optimization both in problem formulation and problem solving.

This seminar should be of considerable interest to workers in the fields of numerical analysis, statistics, operations research, engineering, physics, chemistry, economics and computer science. A copy of the day's program is given below. Anybody interested in attending the seminar should complete the attached form and return it to the Australian National University.

PROGRAM

Papers dealing with topics from a number of fields of optimization theory will be presented at the seminar. The program reflects the broad classification of these topics under the headings:

1. Unconstrained Optimization.
   Includes contributions on function minimization, global optimization and statistical optimization.

2. Constrained Optimization.
   Includes contributions on projection methods, methods for nonlinear programming, integer programming, dynamic programming and optimal control theory.

Short discussions of applications will also be included to illustrate both the problem source and the solution methods.

0850   START

0900   Optimization in the Real World
                Prof. R.B. Potts, Applied Mathematics, University of
                Adelaide.
0930   A New Algorithm for Minimizing a Function of Several Variables
            Without Calculating Derivatives
                Dr R.P. Brent, IBM, New York.

1005   MORNING TEA

165

1030	Global Optimization
	Dr R.S. Anderssen, Computer Centre, A.N.U.
1100	Curve Fitting by Splines as an Application of Unconstrained Optimization
	Mr D.L. Jupp, Mathematics Department, Flinders University.
1115	Bayesian Inference
	Mr J.H.T. Morgan, Statistics, S.G.S., A.N.U.
1135	The Estimation of a Nonlinear System
	Prof. E.J. Hannan and Dr R.C. Boston, Statistics. R.S.S.S., A.N.U.
1150	Projection Methods for Linearly Constrained Problems
	Mr D.M. Ryan, Computer Centre, A.N.U.

1220   LUNCH

1320	Penalty Function Methods
	Dr M.R. Osborne, Computer Centre, A.N.U.
1350	An Eclectic Approach to Nonlinear Programming
	Dr Y. Bard, IBM, New York.
1420	Experience with a Method for Handling Large Structural Linear Programming Problems with Integer Restrictions
	Prof. J.M. Bennett and Miss J.J. Edwards, Basser Computing Department, University of Sydney.
1440	An Improperly Posed Quadratic Programming Problem
	Mr L.S. Jennings, Computer Centre, A.N.U.

1455   DISCUSSION

1500   AFTERNOON TEA

1530	Fractional Programming
	Prof. B. Mond, Mathematics Department, La Trobe University.
1600	The Optimised Placing of Files on Disk Using Dynamic Programming
	Mr A.J. Bayes, IBM, Canberra.
1615	Problems of Optimal Control Theory – An Introduction
	Dr B.D.O. Anderson, Electrical Engineering Department, University of Newcastle.
1645	An Application of Optimal Control Theory
	Dr B.P. Molinari and Mr K. Arnold, School of Electrical Engineering, University of New South Wales.

1700   DISCUSSION

Please Mail to:

One-day seminar committee, Computer Centre,

Australian National University,

P.O. Box 4, Canberra, A.C.T. 2600

No registration fee is required.

I/We shall attend the one-day seminar on Optimization ☐

I/We shall require further information about the proceedings ☐

(please tick where appropriate)

Organization, Institution, University: ....

Postal Address: ....

....

Name: ....

Name: ....

Name: ....