

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified

1. ORIGINATING ACTIVITY (Corporate author) Princeton University Computer Center Princeton, New Jersey	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP -

3. REPORT TITLE  
P-STAT - A USER-ORIENTED LANGUAGE, IMPLEMENTED ON THE IBM 360, FOR STATISTICAL ANALYSIS AND FILE MANIPULATION OF SOCIAL SCIENCE DATA.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)  
Final technical report. June 1, 1965 - Dec. 31, 1967

5. AUTHOR(S) (First name, middle initial, last name)  
Roald Buhler

6. REPORT DATE January 7, 1971	7a. TOTAL NO. OF PAGES 129	7b. NO. OF REFS -
-----------------------------------	-------------------------------	----------------------

8. CONTRACT OR GRANT NO. a. NONR-1858 (53) b. PROJECT NO. c. NR 142-216 d.	9a. ORIGINATOR'S REPORT NUMBER(S) -
	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) -

10. DISTRIBUTION STATEMENT  
This document has been approved for public release and sale. Its distribution is unlimited. Reproduction in whole or in part is permitted for any purpose by the United States Government.

11. SUPPLEMENTARY NOTES Design and implementation of a statistical system.	12. SPONSORING MILITARY ACTIVITY Office of Naval Research Arlington, Va. 22217
---	--

13. ABSTRACT  
P-STAT is a system of 285 Fortran IV programs that does statistical operations on files of data. It runs in a region size of 250K bytes on an IBM 360. Files are created, manipulated, saved between runs, retrieved, printed, etc., by P-STAT commands. Many files can be in use during one P-STAT run. A file is referred to by its name throughout a run. It can have thousands of rows and up to 450 columns. Missing data is permitted. Each row and column has an 8 character label. A free format selection and data modification language permits parts of files to be accessed and new variables to be created. Macros of P-STAT operation can be defined and then used repeatedly. The system performs many statistical operations, including factor analysis, regression, correlation, discriminant analysis, cross-tabulation, frequency distribution and analysis of variance.

(U)

D D C  
RECEIVED  
JAN 5 1972  
ALLEN  
D

AD 734697

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
STATISTICAL PROGRAMS FILE MANIPULATION CORRELATION REGRESSION FACTOR ANALYSIS CROSS-TABULATION FORTRAN						

PRINCETON UNIVERSITY

COMPUTER CENTER

JANUARY 7, 1971

FINAL REPORT  
CONTRACT NONR-1858(53)

NP 142-216

WITH THE OFFICE OF NAVAL RESEARCH

\*\*\*\*\*  
\*  
\* P-STAT - A USER-ORIENTED LANGUAGE, IMPLEMENTED ON \*  
\* THE IBM 360, FOR STATISTICAL ANALYSIS AND FILE \*  
\* MANAGEMENT OF SOCIAL SCIENCE DATA. \*  
\*  
\* ROALD BUHLER \*  
\*  
\*\*\*\*\*

THIS DESCRIBES P-STAT..... VERSION 52.5

THIS WORK MADE USE OF COMPUTER FACILITIES SUPPORTED IN PART BY NATIONAL SCIENCE FOUNDATION GRANTS NSF-GP-579, NSF-GJ-34, AND NSF-GU-3157.

COPYRIGHT (C) 1971, ROALD BUHLER

THIS DOCUMENT HAS BEEN APPROVED FOR PUBLIC RELEASE AND SALE. ITS DISTRIBUTION IS UNLIMITED. REPRODUCTION IN WHOLE OR IN PART IS PERMITTED FOR ANY PURPOSE BY THE UNITED STATES GOVERNMENT.

Reproduced by  
NATIONAL TECHNICAL  
INFORMATION SERVICE  
Springfield, Va. 22151

DDC  
REFERENCE  
JAN 6 1972  
REGISTRY  
D.

134

PRINCETON UNIVERSITY

COMPUTER CENTER

JANUARY 7, 1971

FINAL REPORT  
CONTRACT NONR-1858(53)  
NR 142-216  
WITH THE OFFICE OF NAVAL RESEARCH

```
*****  
*  
* P-STAT - A USER-ORIENTED LANGUAGE, IMPLEMENTED ON *  
* THE IBM 360, FOR STATISTICAL ANALYSIS AND FILE *  
* MANAGEMENT OF SOCIAL SCIENCE DATA. *  
* ROALD BUHLER *  
*  
*****
```

THIS DESCRIBES P-STAT..... VERSION 52.5

THIS WORK MADE USE OF COMPUTER FACILITIES SUPPORTED IN PART BY NATIONAL SCIENCE FOUNDATION GRANTS NSF-GP-579, NSF-GJ-34, AND NSF-GU-3157.

COPYRIGHT (C) 1971, ROALD BUHLER

THIS DOCUMENT HAS BEEN APPROVED FOR PUBLIC RELEASE AND SALE. ITS DISTRIBUTION IS UNLIMITED. REPRODUCTION IN WHOLE OR IN PART IS PERMITTED FOR ANY PURPOSE BY THE UNITED STATES GOVERNMENT.

LIST OF MAJOR SECTIONS - A COMPLETE TABLE OF CONTENTS WITH  
PAGE NUMBERS IS AT THE END OF THIS MANUAL.

1. INTRODUCTION
2. CHANGES WITHIN RELEASES OF VERSION 52
3. THE P-STAT CONTROL LANGUAGE
4. SELECTING SPECIFIC ROWS AND/OR COLUMNS FROM A FILE
5. SYSTEM-INVOLVED OPERATIONS
6. INPUT-OUTPUT OPERATIONS
7. SAVING AND USING P-STAT FILES ON TAPES
8. FILE MANIPULATION OPERATIONS
9. MATRIX OPERATIONS
10. CORRELATION OPERATIONS
11. FACTOR ANALYSIS
12. CROSS-TABULATION ( CROSSTAB )
13. DISCRIMINANT ANALYSIS ( DISCRIM )
14. FREQUENCY DISTRIBUTIONS ( FREQ )
15. REGRESSION ( MULTR )
16. TWO WAY ANALYSIS OF VARIANCE ( PRYQ )
17. OTHER PROGRAMS
18. CREATING A P-STAT FILE FROM CARDS ( DATA )
19. DATA MODIFICATION AND GENERATION USING FOR/IF/SET/SETX
20. MACROS OF P-STAT OPERATIONS
21. DO LOOPS OF P-STAT OPERATIONS
22. JCL FOR REFERENCING TAPES ON THE 360
23. ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM
24. ORGANIZATION OF P-STAT DISTRIBUTION TAPES
25. A LARGER 360 VERSION
26. SYSTEM USE OF STORAGE
27. ACKNOWLEDGMENTS
28. A LIST OF THE LEGAL IDENTIFIERS FOR EACH OPERATION

## 1. INTRODUCTION

P-STAT IS A SYSTEM OF PROGRAMS THAT DOES STATISTICAL OPERATIONS ON FILES OF DATA. AMONG ITS FEATURES ARE.....

A SYNTAX ORIENTED CONTROL LANGUAGE, WITH THE ABILITY TO  
SPELCT PARTS OF FILES WHILE THEY ARE IN USE. FOR EXAMPLE...

IN = X( R 21-40, 77-100 )

SAYS THAT THE INPUT TO SOME P-STAT OPERATION IS ROWS 21  
THROUGH 40 AND 77 THROUGH 100 OF A FILE NAMED 'X'.

IN = Y( SETX TOTAL TO SUM( 1-15 ) )

SAYS...SET AN EXTRA VARIABLE, NAMED 'TOTAL', TO THE SUM OF THE  
FIRST 15 VARIABLES. THIS WOULD BE DONE FOR EACH ROW OF FILE Y.

A FILE HANDLING CAPABILITY WITHIN THE SYSTEM - ANY FILE GENERATED  
BY A P-STAT OPERATION CAN BE USED ( WHENEVER PRACTICAL ) BY  
ANY SUBSEQUENT P-STAT OPERATION, DURING THE SAME RUN OR  
DURING A LATER RUN. THE USER REFERS TO FILES BY THEIR NAMES,  
AND USUALLY HAS NO CONCERN FOR THEIR LOCATION OR FORM.

MACROS - THE ABILITY TO DEFINE, AS AN OPERATION, AN OFTEN USED  
SEQUENCE OF P-STAT CONTROL STATEMENTS. LATER USE OF THE  
NAME OF THE MACRO, WITH APPROPRIATE FILE NAMES AS ARGUMENTS,  
INVOKES THE ENTIRE SEQUENCE.

MACHINE INDEPENDENCE - THIS SYSTEM, WITH A FEW EXCEPTIONS, IS  
WRITTEN IN VERY SIMPLE FORTRAN IV. THE SECTION 'ORGANIZATION  
OF P-STAT DISTRIBUTION TAPES' DISCUSSES THE EXCEPTIONS.

ERROR RECOVERY - WHEN THE USER DOES SOMETHING WRONG, THE P-STAT  
SYSTEM ( ASSUMING THE ERROR IS SOMETHING IT UNCOVERS ) WILL  
REPORT THE PROBLEM, CLEAN UP ITS BUFFERS AND SO FORTH, AND  
ATTEMPT TO CONTINUE WITH THE NEXT P-STAT OPERATION UNTIL TOO  
MANY CONSECUTIVE ERRORS HAVE OCCURRED.

A GENERAL DATA INPUT PROGRAM - THIS OPERATION ( CALLED 'DATA' )  
READS INPUT CARDS AND CREATES A P-STAT FILE. IT CHECKS THAT  
THE CARDS WITHIN A CASE ARE IN ORDER, AND THAT NUMERIC FIELDS,  
TO BE READ AS NUMBERS, ARE REALLY NUMBERS. IT KEEPS CONTROL  
WHEN SUCH ERRORS OCCUR. A SUMMARY OF THE INPUT CARDS IS  
PRINTED.

SELF-DOCUMENTATION - THIS MANUAL IS PUNCHED ON CARDS. A P-STAT  
OPERATION RETRIEVES THE INFORMATION AND PRINTS IT, NUMBERING  
THE PAGES AND CREATING A TABLE OF CONTENTS. A SHORT PRIMER  
IS ALSO AVAILABLE. MANY USERS WILL FIND IT HELPFUL TO READ  
THE PRIMER BEFORE TRYING TO READ THIS ENTIRE MANUAL.

OPEN-ENDEDNESS - A USER OF A STATISTICAL SYSTEM FREQUENTLY NEEDS TO  
ADD A NEW PROGRAM TO THE SYSTEM, PERHAPS TO BE USED ONLY ONCE  
OR TWICE. A CATALOGED PROCEDURE, 'ESTAT52U', PERMITS ( ON  
PRINCETON'S 360/91 ) A USER TO COMPIL A PROGRAM, CONSTRUCT  
HIS OWN P-STAT OVERLAY AND EXECUTE IT ( I.E., DO A P-STAT RUN  
WHICH INCLUDES HIS PROGRAM ). ANY SUCH OPERATION WRITTEN AT

ONE COMPUTING INSTALLATION SHOULD BE EASY TO ADD TO A P-STAT SYSTEM AT ANOTHER INSTALLATION. SEE THE SECTION 'ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM'.

A STEP TOWARDS INTER-SYSTEM COMPATIBILITY - A P-STAT OPERATION, 'OSIPIS.P', ALLOWS ( ON A 360 ) A FILE GENERATED BY THE UNIVERSITY OF MICHIGAN OSIRIS SYSTEM TO BE CONVERTED TO A P-STAT FILE WITHIN A P-STAT RUN.

```

*****
*
*   P-STAT DISTRIBUTION POLICY   *
*
*****

```

INFORMATION ABOUT THE CURRENT P-STAT DISTRIBUTION POLICY CAN BE REQUESTED FROM.....

ROALD BUHLER  
 PRINCETON UNIVERSITY COMPUTER CENTER  
 87 PROSPECT STREET  
 PRINCETON, NEW JERSEY      08540

TELEPHONE NUMBER IS... 609-452-6042      OR    609-452-6000

\*\*\*\*\* DISCLAIMER \*\*\*\*\*

ALTHOUGH THESE PROGRAMS HAVE BEEN TESTED, NO WARRANTY OF ANY KIND IS MADE BY THE AUTHOR, BY ANY SPONSORING AGENCIES, OR BY PRINCETON UNIVERSITY, AS TO THE ACCURACY AND FUNCTIONING OF THESE PROGRAMS AND THE RELATED DOCUMENTS.

\*\*\*\*\*

2. CHANGES WITHIN RELEASES OF VERSION 52.

VERSION 52.5 IS AN OVERLAY REQUIRING A REGION SIZE OF 250K BYTES ON A 360. A LARGF VERSION OF PSTAT52 FOR THE 360 ( USING 600K BYTES ) IS DESCRIBED IN THE SECTION 'A LARGER 360 VERSION'.

VERSION 52.5 HAS ABOUT 285 DECKS ( I.E., SUBROUTINES ) TOTALLING ABOUT 23,000 FORTRAN SOURCE CARDS. ABOUT 7,400 OF THESE ARE COMMENTS. THERE ARE ABOUT 570 CALLS TO ERROR ROUTINES. EACH OF THESE PRINTS A DIAGNOSTIC MESSAGE.

THE LINKS IN P-STAT ARE ALL WRITTEN IN SINGLE PRECISION. THERE IS, OF COURSE, A TRADE-OFF OF ACCURACY VRSUS SPEED, SIZE, AND BETWEEN-COMPUTER COMPATIBILITY. SO FAR, THE ACCURACY LIMITATIONS OF SINGLE PRECISION HAVE NOT BEEN A SERIOUS PROBLEM. PLACING SOME OF THE PROGRAMS IN DOUBLE PRECISION, OR HAVING ALTERNATE VFRSIONS IN DOUBLE PRECISION, IS BEING CONSIDFRED. SEVFRAL PROGRAMS, INCLUDING 'INTCDS' AND 'DISCRIM', SCALE THE DATA INTERPNALLY TO LESSEN THE POSSIBILITY OF ACCURACY LOSS DUE TO VARIABLES WITH LARGE MEANS AND SMALL VARIANCES.

THE FOLLOWING IS A BRIEF SUMMARY OF THE CHANGES INCORPORATID INTO THE MOST RECENT RELEASES.

## VERSION 52.3 (APRIL, 70) HAS THESE CHANGES...

- CONCATENATION OF FILES. 'IN=A+B' WILL BE TREATED AS ONE LINKED FILE.
- MUDMERGE, MULTIPLE UP-DOWN MERGE. THIS IS SOMEWHAT OBSOLETE ALREADY BECAUSE THE ABOVE CONCATENATION, WITH NO.OP OR SCAN, DOES VIRTUALLY THE SAME THING.
- FIVE ADDITIONAL FUNCTIONS IN 'SET' OR 'SETX' STATEMENTS.....  
LOG10, EXP, SIN, COS, ATAN.
- INTCDS, INTMDS AND INTMDA NOW ATTEMPT TO MODIFY THE DATA INTERNALLY SO THAT CROSSPRODUCT CALCULATIONS ARE PERFORMED ON SCORES WHOSE ( INTERNAL ) MEANS ARE CLOSER TO ZERO.
- A CONTROL, 'EQUALCAT', HAS BEEN ADDED TO 'FREQ' AND 'FREQ.C' ALLOWING MORE CONTROL OVER CATEGORY SIZES.
- A BUG IN 'MULTR' CAUSING OCCASIONAL (AND, FORTUNATELY, IRRELEVANT) O.S. ERROR MESSAGES HAS BEEN FIXED.
- THE 'DATA' PROGRAM WILL DO THE BEST IT CAN WITH ROW LABELS FOUND ON INPUT CARDS. AN INVALID LABEL USED TO BE DISASTEROUS. ALSO, THE 'DATA' PROGRAM CAN NOW RECODE X(-) OR Y(+) PUNCHES INTO, FOR EXAMPLE, 11 AND 12.
- INFORMATION REGARDING USE OF 800 SPT TAPES WHEN THE NORM IS 1600 IS AT THE END OF THE JCL SECTION.
- A PORTRAN H VERSION IS NOW STANDARD.
- THE OVERALL SIZE OF THE OVERLAY IS SMALLER, DUE TO SOME INTERNAL CHANGES AND ALSO TO A MORE INVOLVED OVERLAY STRUCTURE.
- THE ABILITY, IN A 'FOR, SETX' PHASE TO GENERATE NEW LABELS BASED IN PART ON THE OLD LABELS. ALSO, THE USE OF 'SETX .X+10.' OR 'SETX .C7.' TO DEFINE THE ACTUAL POSITION OF A NEW VARIABLE HAS BEEN REPLACED BY AN EASIER CONVENTION.

## VERSION 52.4 ( NOVEMBER, 70 ) HAS THESE CHANGES...

- THE OVERLAY SIZE IS DOWN TO 225K, THEREFORE A REGION OF 250K SHOULD BE LARGE ENOUGH TO RUN MOST P-STAT JOBS.
- CROSSTAB HAS BEEN REVISED.
- 'IN = W ( B 1-10 )' MEANS SELECT BOTH ROWS 1-10 AND COLUMNS 1-10 OF FILE 'W'.
- SEVERAL BUGS HAS BEEN FIXED. LARGE 'DO' LOOPS, MULTIPLE PASSES OF CONCATENATED FILES ( A+B+C, FOR EXAMPLE ), AND (\*) USAGE INVOLVING 'R' SELECTION ALL WORK BETTER.
- A CARD IMAGE FILE, USED AS INPUT BY 'DATA' OR 'SDATA', NO LONGER MUST HAVE A '\*END' CARD TO DEFINE ITS END. P-STAT NOW PROVIDES A '\*END' CARD WHEN THE END OF FILE IS HIT.
- A NEW OPERATION, 'XHEAD' ( FOR EXTRA HEADING ), ALLOWS ONE TO MODIFY PART OF THE PREVIOUS HEADING. THIS FEATURE CAN BE USEFUL IN MACROS.
- 'MANUAL' NOW SUPPORTS THE OPTIONAL FLAGGING, WHEN THE P-STAT MANUAL IS PRINTED, OF RECENTLY INSERTED FEATURES.
- 'EDIT' WILL NOW ( OPTIONALLY ) PRINT THE VARIABLE ( COLUMN ) LABELS OF EACH FILE WHILE IT EDITS A P-STAT PERMANENT DATA TAPE.
- 'EDIT', 'SAVE', AND 'FIND' RUN CONSIDERABLY FASTER.
- A NEW OPERATION, 'COPY.PDT', COPIES A P-STAT PERMANENT DATA TAPE. THIS IS MUCH FASTER THAN USING... 'FIND,TAPE=61' AND 'SAVE,TAPE=62'.
- 'MERGE' NOW HAS A 'MIDDLE' OPTION SO THAT A LEFT/MIDDLE/RIGHT MERGE CAN BE DONE, ACCOMPLISHING THE SIDWAYS MERGING OF 3 FILES IN ONE STEP.
- IN A MACRO DEFINITION, IF A FILE NAME OR A LABEL IS ENCLOSED BY (( )), THE PARENTS ARE DELETED WHEN THE MACRO IS EXPANDED, LEAVING THE NAME OR LABEL INTACT. THIS PERMITS THE FILE NAME OR LABEL TO BE, IN EFFECT, A CONSTANT IN THE MACRO.
- THE CONVENTIONS FOR ADDING TEMPORARY LINKS TO THE SYSTEM HAVE BEEN CHANGED. A TYPICAL LINK SHOULD BE CALLED 'USER1' RATHER THAN 'LNK101'.
- 'CLEANCOR' IS A NEW OPERATION WHICH CLEANS A CORRELATION MATRIX. VARIABLES ARE DELETED WHOSE LARGEST ABSOLUTE CORRELATION WITH ANY OTHER VARIABLE IS LESS THAN OR EQUAL TO A USER SUPPLIED VALUE.
- 'P.COFF' HAS SEVERAL OPTIONS TO USE WHEN THE STANDARDIZED INPUT FILE HAS MISSING DATA.
- 'SCAN', WHEN THE INPUT FILE HAS MISSING DATA, CAN SET THE OUTPUT VALUE IN A 'SDATA' FILE TO EITHER 0. OR MISSING.

VERSION 52.5 ( JAN., 71 ) HAS THESE CHANGES...

- A 'SET' PHRASE CAN NOW SET A VARIABLE TO THE STANDARD DEVIATION OF A LIST. FOR EXAMPLE.....  
SET VAR10 TO SDEV ( VAR1, VAR3 - VAR8 )
  
- 'DISCRIM', A MULTIPLE GROUP DISCRIMINANT ANALYSIS PROGRAM, HAS BEEN ADDED.

PLANS FOR VERSION 53.

VERSION 53 IS CURRENTLY UNDER DEVELOPMENT. IT WILL HAVE SOME ( PERHAPS MOST ) OF THE FOLLOWING FEATURES....

- 16 CHARACTER ROW AND COLUMN LABELS ( P-STAT NOW ALLOWS 8 ).
- AN OPTIONAL SECONDARY ROW LABEL, OF VARIABLE SIZE, UP TO 100 CHARACTERS.
- THREE SIZES OF OVERLAY, 190K ALLOWING 150 VARIABLES, 250K ALLOWING THE CURRENT 450 VARIABLES, AND 700K ALLOWING 1500 VARIABLES.
- PROVISION FOR SCORE LABELS, WHICH WOULD ALLOW BETTER CROSSTAB PROGRAMS.
- INCLUSION OF A NUMBER OF ADDITIONAL STATISTICAL OPERATIONS, POSSIBLY A MANOVA PROGRAM, CANONICAL CORRELATION, STEPWISE REGRESSION, PATH ANALYSIS, SOME ADDITIONAL ROTATION PROGRAMS FOR FACTOR ANALYSIS, ETC.
- MORE INTERFACING WITH OTHER SOCIAL SCIENCE STATISTICAL SYSTEMS.
- THE USE OF LOWER CASE ( AS WELL AS UPPER ) WHEN PRINTING THIS MANUAL.

### 3. THE P-STAT CONTROL LANGUAGE

---

CONSIDER THIS TYPICAL P-STAT STATEMENT...

INTCDS, IN = X, COR = XCOR \$

IT HAS AN OPERATION NAME (INTCDS) AND TWO PHRASES. THE OPERATION NAME IMPLIES... INTERCORRELATE, COMPLETE DATA, SYMMETRIC. THE PHRASES SUPPLY INFORMATION TO THE INTCDS OPERATION. THE FIRST ( IN = X ) SAYS THAT THE INPUT IS A FILE NAMED X, THE SECOND (COR=XCOR) SAYS THAT A FILE OF CORRELATIONS SHOULD BE COMPUTED AND NAMED XCOR.

NAMES AND FILES ARE DISCUSSED BELOW.

---

#### NAMES AND LABELS

A NAME MAY HAVE NO MORE THAN 8 CHARACTERS. THE FIRST CHARACTER MUST BE A LETTER. THE REST CAN BE LETTERS, NUMBERS, AND DECIMAL POINTS. BLANKS WITHIN A NAME ARE NOT ALLOWED.

THESE ARE EXAMPLES OF LEGAL NAMES...

A  
ZZ10  
X...3  
ARIZONA  
J.1

THESE ARE SOME ILLEGAL NAMES...

NEWJERSEY  
2A  
A/3  
NEW YORK

THESE RULES HOLD FOR ALL NAMES OR LABELS IN THE P-STAT SYSTEM. THIS INCLUDES FILE NAMES, LABELS FOR THE ROWS AND COLUMNS OF A FILE, NAMES OF P-STAT OPERATIONS, AND NAMES OF IDENTIFIERS FOR OPTIONS WITHIN OPERATIONS (SEE BELOW).

WHEN A P-STAT FILE IS GENERATED FROM OTHER P-STAT FILES BY SOME P-STAT OPERATION, IT IS GIVEN ROW AND COLUMN LABELS WHICH ARE USUALLY BASED ON THE LABELS FOUND IN THE INPUT FILES USED IN THE OPERATION.

---

## FILES

A FILE IS A RECTANGULAR COLLECTION OF NUMERIC DATA, CARRIED IN SINGLE PRECISION.

IT HAS ROWS AND COLUMNS.

IN SOCIAL SCIENCE APPLICATIONS, THE ROWS ARE FREQUENTLY PEOPLE, AND THE COLUMNS ARE USUALLY VARIABLES.

EACH ROW HAS A LABEL, AND EACH COLUMN HAS A LABEL ( FOR EXAMPLE, 'AGE' ).

MISSING DATA IS PERMITTED IN A FILE. MANY OF THE P-STAT OPERATIONS WERE DESIGNED TO HANDLE FILES WITH SOME MISSING DATA.

IN GENERAL, FILES ARE REFERRED TO BY NAME. USERS DO NOT NEED TO KNOW WHERE A FILE IS.

THE MAXIMUM NUMBER OF COLUMNS IN A FILE IN VERSION 52 IS 450.

THE LIMITING FACTOR IN THE MAXIMUM SIZE OF A FILE IS THE CAPACITY OF A SINGLE 2400 FOOT TAPE ( APPROXIMATELY 13,000 ROWS BY 300 COLUMNS, 40,000 ROWS BY 100 COLUMNS, 80,000 ROWS BY 50 COLUMNS, ETC. ). THIS ASSUMES 800 BPI AND 2000 BYTE RECORDS ON THE TAPE.

P-STAT DOES NOT DO ANYTHING PARTICULARLY TRICKY WITH TAPE USAGE. IF AN OPERATING SYSTEM PERMITS MULTI-REEL FILES, A P-STAT FILE COULD BE LARGER THAN THE NUMBERS CITED ABOVE. HOWEVER, THERE HAS BEEN LITTLE IF ANY MULTI-REEL P-STAT EXPERIENCE AT PRINCETON.

IN THIS MANUAL, THE WORD 'MATRIX' IS OCCASIONALLY USED INSTEAD OF THE WORD 'FILE'. 'FILE' IS GENERAL, 'MATRIX' TENDS TO BE A SQUARISH FILE.

-----  
THE CARDS NECESSARY TO RUN A P-STAT JOB CONSIST OF...

1. JOB DEFINITION CARDS TO THE SUPERVISORY OPERATING SYSTEM.
  2. P-STAT CONTROL STATEMENTS. EACH DOES A P-STAT OPERATION. SOME OF THESE MUST BE FOLLOWED BY...
  3. DATA CARDS FOR THE P-STAT OPERATION, WHEN NEEDED. THIS CAN BE NUMERIC DATA, OR CAN CONTAIN FURTHER INFORMATION NEEDED BY THE SPECIFIC P-STAT OPERATION.
-

THE FOLLOWING IS A SIMPLE PRINCETON 360 P-STAT RUN. IT READS IN SOME DATA, PRINTS IT, COMPUTES AN INTERCORRELATION MATRIX, AND PRINTS IT TWICE (TO 2 DECIMAL PLACES AND THEN TO 4 DECIMAL PLACES).

```
// JOB PTC.
// EXEC PSTAT52
//PSTAT.SYSIN DD *
HEAD = (SAMPLE PSTAT RUN) $
SDATA = A, NV = 5, FORMAT = ( 5F1.0 ) $
      (HERE ARE DATA CARDS)
*END
PRINT = A $
INTCDS, IN = A, COR = AINTER $
PRINT = 2 / AINTER / 4 / AINTER $
END $
/*
```

CONSIDER THE STATEMENT WHICH STARTS WITH INTCDS...

'INTCDS' IS THE OPERATION NAME.  
 'IN' IS AN IDENTIFIER LINKING THE PROGRAM TO THE NAME OF THE INPUT FILE.  
 'A' IS THE NAME OF THE FILE TO BE USED AS INPUT.  
 'COR' IS AN IDENTIFIER WHOSE PRESENCE ON THE CARD INDICATES, TO THIS OPERATION, THAT CORRELATIONS ARE WANTED.  
 'AINTER' WILL BE THE NAME GIVEN TO THE FILE OF INTERCORRELATIONS.  
 '\$' ENDS THE P-STAT CONTROL STATEMENT.

NOW, CONSIDER THE ENTIRE RUN...

THE FIRST THREE CARDS, // JOB AND THE TWO FOLLOWING, ARE 360 JOB CONTROL CARDS FOR THE PRINCETON 360 OPERATING SYSTEM. P-STAT CONTROL STATEMENTS AND DATA CARDS OCCUR IMMEDIATELY AFTER THE //PSTAT.SYSIN DD \* CARD.

'HEAD' IS A P-STAT OPERATION WHICH PROVIDES A HEADING WHICH WILL APPEAR ON SUBSEQUENT PRINTING.

'SDATA' IS AN OPERATION WHICH READS IN CARDS, CREATING A P-STAT FILE NAMED A ( IN THIS EXAMPLE ), WITH 5 VARIABLES ( COLUMNS ) IN IT. IT IS FOLLOWED BY SOME NUMBER OF DATA CARDS, READ USING FORMAT 5F1.0. READING OF CARDS CONTINUES UNTIL A CARD WITH '\*END' IN COLUMNS 1-4 IS FOUND.

AT THIS POINT, FILE 'A' EXISTS SOMEWHERE IN THE SYSTEM. WE COULD USE IT IMMEDIATELY AS INPUT TO SOME OTHER OPERATION, OR WE COULD DO SOME ACTIVITY NOT INVOLVING FILE A, FOR EXAMPLE, USING SDATA AGAIN TO READ IN A FILE TO BE NAMED B. IN EITHER EVENT, FILE A IS AVAILABLE AT ANY TIME DURING THE REST OF THIS RUN.

'PRINT = A' PRINTS THAT FILE.



AN IDENTIFIER CAN HAVE MULTIPLE ARGUMENTS...

PRINT = 2 / AINTER / 4 / AINTER .

ONLY CERTAIN OPERATIONS ALLOW FOR MULTIPLE ARGUMENTS. THIS USAGE WOULD BE ORGANIZED INTERNALLY ( BY THE SYSTEM ) AS FOUR SEPARATE PHRASES THAT HAPPEN TO HAVE THE SAME IDENTIFIER.

AN IDENTIFIER CAN REFER TO CONCATENATED FILES...

IN = A + B + C

THIS WOULD CAUSE THE COLUMN LABELS OF 'A', THEN THE ROWS OF 'A', THE ROWS OF 'B' AND THE ROWS OF 'C' TO BE TREATED AS ONE FILE BY THE OPERATION. THEY MUST HAVE THE SAME NUMBER OF COLUMNS. THE COLUMN LABELS OF 'B' AND 'C' ARE SKIPPED OVER WITHOUT COMPARING AGAINST THE LABELS OF 'A'. A FILE CAN BE REPEATED...

IN = A ( C 1-20 ) + A ( C 21-40 )

FURTHER COMMENTS ABOUT THE IDENTIFIER COP IN THE INTCDS STEP...

COP IS AN IDENTIFIER SAYING, IN EFFECT, THAT CORRELATIONS ARE WANTED. THE OPERATION INTCDS CAN PROVIDE MORE THAN ONE OUTPUT. IN FACT, IT COULD PROVIDE ANY OR ALL OF FIVE DIFFERENT OUTPUT MATRICES ( CROSS-PRODUCTS, COVARIANCES, CORRELATIONS, SLOPES, AND INTERCEPTS ). THE USER TELLS THE OPERATION WHICH OPTION(S) HE WANTS BY USING THE APPROPRIATE IDENTIFIER(S).

IN GENERAL, AN IDENTIFIER SPECIFIES AN OPTION. IN SOME OPERATIONS, AN IDENTIFIER IS USED WITHOUT AN ARGUMENT SIMPLY TO INDICATE THAT THE PROGRAM SHOULD DO SOME SPECIFIC THING. USUALLY, THE IDENTIFIER LINKS THE OPTION TO SOME INFORMATION THAT THE OPTION WILL NEED ( A NAME, A NUMBER, OR A LIST ).

SOME IDENTIFIERS MUST BE PROVIDED. FOR EXAMPLE, INTCDS MUST HAVE AN INPUT FILE ( IN = A ) AND THE STEP IS POINTLESS WITHOUT USING AT LEAST ONE OF THE OUTPUT OPTIONS.

PHRASES WITHIN A CONTROL STATEMENT CAN OCCUR IN ANY ORDER.

A P-STAT CONTROL STATEMENT CAN BE PUNCHED ON SEVERAL CARDS.

SEVERAL P-STAT CONTROL STATEMENTS CAN BE PUNCHED ON 1 CARD.

CONTROL STATEMENT LIMITS, IN VERSION 52, ARE.....

1. 25 PHRASES WITHIN A STATEMENT.

2. 15 FULL CARDS FOR A SINGLE STATEMENT. THIS IS ACTUALLY 1200 CHARACTERS (15X80). IF A GIVEN CARD IN A P-STAT STATEMENT HAS, FOR EXAMPLE, PUNCHING THROUGH COLUMN 57, THAT AND THE FIRST BLANK AFTER ARE RETAINED. THUS, 58 RATHER THAN 80 CHARACTERS COUNT TOWARDS THE 1200. IN OTHER WORDS, WHEN 'N' BLANKS END A CARD, 'N-1' OF THEM ARE IGNORED. THIS MAY BE USEFUL WHEN DECODES ON MANY VARIABLES OF A FILE ARE BEING DONE. FOR EXAMPLE, 30 CARDS, EACH PUNCHED ONLY IN THE FIRST 35 COLUMNS, WOULD BE ACCEPTABLE.

---

CATALOGED PROCEDURES ON PRINCETON'S 360

CURRENTLY, TWO CATALOGED PROCEDURES ALLOW USE OF THE VERSTON 52 SYSTEM AT PRINCETON. THEY ARE...

PSTAT52 THIS ACCESSES THE SYSTEM DESCRIBED IN THIS MANUAL.

A LARGE VERSION, DESCRIBED IN THE SECTION 'A LARGER 360 VERSION', CAN ALSO BE REFERENCED BY THIS PROCEDURE BY THE USE OF AN ADDITIONAL PARAMETER.

PSTAT52U THIS COMPILES A USER LINK, LINKEDITS AN OVERLAY INCLUDING IT ( IF IT COMPILES ), AND EXECUTES THE NEW OVERLAY. IT IS DISCUSSED IN THE SECTION 'ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM'.

A DECK FOR THE SIMPLEST RUN WOULD CONSIST OF...

```
// JOB ETC.  
// EXEC PSTAT52  
//PSTAT.SYSIN DD *  
  ( P-STAT CARDS )  
/*
```

A SEPARATE SECTION, 'JCL FOR REFERENCING TAPES ON THE 360' COVERS USE OF THESE PROCEDURES WHEN P-STAT DATA TAPES ARE USED. THESE TAPES ARE GENERATED BY OPERATIONS 'SAVE' AND 'ASSIGN'.

TWO OTHER CATALOGED PROCEDURES ARE USED AT PRINCETON....

PSTAT52M PRINTS A COPY OF THE MANUAL.

PSTAT52P PRINTS A COPY OF THE PRIMER.

4. SELECTING SPECIFIC ROWS AND/OR COLUMNS FROM A FILE

IN GENERAL, ANY REFERENCE TO AN EXISTING FILE NAME CAN BE QUALIFIED BY PARENTHESES AFTER THE FILE NAME. THE CONTENTS OF THE PARENTHESES CONTAIN TEXT THAT SELECTS OR MODIFIES THE DATA IN A FILE AS IT IS FED INTO AN OPERATION.

THERE ARE NOW FIVE TYPES OF QUALIFICATION...

1. ROW SELECTION.
2. COLUMN SELECTION AND RE-ORDERING.
3. BOTH ( THE SELECTION APPLIES BOTH TO ROWS AND TO COLUMNS ).
4. STAR ( \* SAYS... REAPPLY THE MOST RECENT QUALIFICATION USAGE TO THE CURRENT FILE ).
5. FOR, IF, SET, SETX. THESE MODIFY DATA AND ALSO GENERATE NEW VARIABLES. THEY ARE DESCRIBED IN A LATER SECTION, 'DATA MODIFICATION AND GENERATION USING FOR/IF/SET/SETX'.

A. ROW SELECTION

THIS SELECTS CERTAIN ROWS OF A FILE. THE FIRST NON-BLANK CHARACTER WITHIN THE PARENTHESES MUST BE R. FOR EXAMPLE, IN = A(R 41-100). THE EFFECTIVE INPUT TO WHATEVER OPERATION IS INVOLVED IS ROWS 41 THROUGH 100 OF FILE A.

CONSIDER....

IN = A ( R JONES - 200 )

ROWS FROM THE ONE LABELLED JONES THROUGH THE ONE IN POSITION 200 SURVIVE THE ROW STATEMENT. IN GENERAL, THE OUTPUT OF THE FIRST EXECUTED PARENTHESIS IS THE INPUT TO THE NEXT PARENTHESES, IF ANY. NOTE THAT POSITIONS AND LABELS CAN BE INTERMIXED. IF FILE B HAS MORE THAN 400 ROWS, AND YOU WISH ROWS 401 ON, USE

IN = B ( R 401 + )

TO GET EVERY FIFTH ROW IN B, USE IN=B(R 1+(10000)). THE (10000) IS A BINARY MASK, DIRECTING THE SYSTEM TO USE THE FIRST, SKIP THE NEXT 4, ETC. TO GET EVERY OTHER ROW, USE... IN=B(R 1+(10) ) . TO GET THE EVEN NUMBERED ROWS, USE... IN=B(R 1+(01) ) .

IN = B(R 51-60, 75, 81-90(10), 201-215(100)) WILL SELECT ROWS 51-60, 75, 81, 83, 85, 87, 89, 201, 204, 207, 210, AND 213.

THE GENERAL FORM IS...

1. (R
2. ONE OR MORE EXPRESSIONS ( WITH COMMAS BETWEEN THE EXPRESSIONS IF MORE THAN ONE )
3. )

AN EXPRESSION CAN BE ONE OF THE FOLLOWING...

- A ROW NUMBER
- A ROW LABEL
- A RANGE, SUCH AS 7-30, OR JONES-SMITH, OR 7+, OR JONES+
- A MASKED RANGE, SUCH AS JONES-200(10)

FOUR POINTS TO NOTE.....

1. ROW REFERENCE (LABELS OR NUMBERS) MUST OCCUR IN THE FILE IN ASCENDING ORDER. IN = A(P 21-40, 1-5) IS NOT ACCEPTED BY THE SYSTEM. THIS IS DUE TO THE SERIAL ACCESS RESTRICTIONS INVOLVED IN READING ROWS OF DATA FROM A FILE, SUCH AS A TAPE.
2. A ROW CANNOT BE REPEATED. 'IN = A( R 10, 10 )' WILL CAUSE AN ERROR MESSAGE.
3. WHEN THERE ARE SEVERAL QUALIFICATIONS AFTER A FILE NAME ( INCLUDING A ROW QUALIFICATION ), THE ROW QUALIFICATION IS ALWAYS EXECUTED FIRST, EVEN THOUGH IT MAY NOT BE THE LEFTMOST ONE. IT IS WISE, BUT NOT OBLIGATORY, FOR THE USER TO PUT IT FIRST.
4. THERE CAN BE ONLY 1 ROW QUALIFICATION PHRASE FOR ANY GIVEN FILE REFERENCE. IN=A(R 1-3 ) ( R 4-6 ) WILL CAUSE AN ERROR MESSAGE TO OCCUR.

B. COLUMN SELECTION

THE WORD 'COLUMN' REFERS TO A COLUMN IN A P-STAT FILE, AND NOT TO A COLUMN OF A PUNCHED CARD.

THE FIRST (NON-BLANK) CHARACTER IN COLUMN SELECTION PARENTHESES MUST BE C. THE REST IS THE SAME AS ROW SELECTIONS, BUT WITHOUT THE FOUR RESTRICTIONS. REFERENCES TO COLUMNS CAN BE OUT OF ORDER, A COLUMN REFERENCE CAN BE REPEATED, A COLUMN PHRASE CAN BE EXECUTED WHEN IT IS FOUND, AND SEVERAL DIFFERENT (C) REFERENCES CAN OCCUR.

IN = A (C 20, 11-20(10), 20) WILL PROVIDE 7 RESULTING COLUMNS. THEY WILL BE THE SAME AS ORIGINAL COLUMNS 20, 11, 13, 15, 17, 19, AND 20, IN THAT ORDER.

'PRINT = R(C 21-AGE) \$' WILL PRINT, FROM FILE B, THOSE COLUMNS FROM THE COLUMN IN POSITION 21 THROUGH THE COLUMN LABELLED AGE. IF, FOR EXAMPLE, AGE WERE THE 29TH COLUMN, THE PRINTOUT WOULD HAVE HAD JUST 9 COLUMNS.

'INTCDS, IN = A(C 21+), COR = AA \$' WILL PROVIDE INTERCORRELATIONS OF ALL VARIABLES EXCEPT THE FIRST 20.

C. BOTH ROW AND COLUMN SELECTION.

IN = CMAT ( B 1-12, 15, 18-21 )  
IS THE SAME AS...  
IN = CMAT ( B 1-12, 15, 18-21 ) ( C 1-12, 15, 18-21 ) .  
RESTRICTIONS OF 'R' TYPE USAGE APPLY.

D. STAR

IN = R (\*). THE MOST RECENTLY ENCOUNTERED QUALIFICATION TEXT USED ON ANY FILE, IN THIS OR A PREVIOUS STEP, IS APPLIED TO FILE B.

ERRORS CAN CAUSE A PROBLEM WITH STAR USAGE. IF SOME ROW SELECTION IS USED IN STEP 1, SOME COLUMN SELECTION IS USED IN STEP 2, AND A STAR IS USED IN STEP 3, ALL IS FINE IF NO ERRORS HAVE OCCURRED. IF THE OPERATION NAME IN STEP 2 WERE MISPELLED, HOWEVER, THE STAR IN STEP 3 WOULD BE AMBIGUOUS. STEP 2 SORT OF DISAPPEARS, AND THE STAR IN STEP 3 IS LEFT HANGING. CLEARLY IT IS WRONG TO GO BACK TO USE THE QUALIFICATION IN STEP 1. THEREFORE, STAR USAGE WILL USE THE MOST RECENT QUALIFICATION, UNLESS AN ERROR HAS OCCURRED SINCE. IF SO, THE STAR USAGE WILL BE REPORTED AS AN ERROR.

LEFT = X(C 1-40), RIGHT = R (\*). COLUMNS ONE TO FORTY OF B WILL BE USED.

CONSIDER...

LEFT = X (C1-AGE), RIGHT = R (\*). IF AGE IS VARIABLE 40 IN FILE X, IT IS NOT NECESSARILY TRUE THAT COLUMNS 1-40 OF FILE B WILL BE USED BECAUSE OF THE (\*). IF AGE IN B IS COLUMN 20, COLUMNS 1-20 WILL BE USED.

IN = X ( B 21 - 50 ) ( C \* ) DOES NOT WORK. (\*) MUST BE THE ONLY THING BETWEEN A FILE NAME AND A COMMA OR DOLLAR SIGN.

5. SYSTEM-INVOLVED OPERATIONS

C  
END  
HEAD  
MANUAL  
MAXERROR  
PPSOURCE  
PRIMER  
PURGE  
RETAIN  
SOURCE  
XHEAD

\*\*\*\*\*

C= (A COMMENT CAN GO HERE.) \$

AS IN FORTRAN, C STANDS FOR COMMENT. IT SHOULD BE...

C = ( LIST )\$

THERE CAN BE 160 CHARACTERS INSIDE THE PARENTHESES ( BUT NO DOLLAR SIGNS, OR EXTRA RIGHT PARENTHESES ).

\*\*\*\*\*

END \$

THIS ENDS A P-STAT RUN. USING 'END \$ END \$' IS POSSIBLY CLEVER. IT CANNOT HURT, AND OCCASIONALLY IS USEFUL WHEN THE FIRST 'END \$' WAS SWALLOWED UP AS PART OF AN IMMEDIATELY PREVIOUS ERROR.

THIS ALSO PRINTS THE TOTAL NUMBER OF ERROR MESSAGES THAT OCCURRED DURING THE RUN.

\*\*\*\*\*

HEAD= (THIS HEADING CAN BE 80 CHARACTERS LONG.) \$

IT IS USEFUL TO HAVE A HEADING DESCRIBING THE CURRENT RUN, OR A SPECIFIC PART OF A RUN, SO THAT BOTH PRINTED RESULTS AND SAVED FILES HAVE INFORMATION DESCRIBING WHAT THE RUN WAS ALL ABOUT.

A SINGLE RUN CAN CONTAIN ANY NUMBER OF HEAD CARDS.

EACH NEW 'HEAD' CARD REPLACES THE HEADING PROVIDED BY THE PREVIOUS 'HEAD' CARD. SEE 'XHEAD' FOR ADDITIONAL HEADING CAPABILITY.

\*\*\*\*\*

MANUAL \$ OF MANUAL = NUMBER \$

THIS CAUSES THE MANUAL TO BE PRINTED. IT TAKES ABOUT 26 SECONDS OF COMPUTING TIME ( ON A 360/91 ) AND PRODUCES ABOUT 90 FULL PAGES.

MANUAL = 52.3 \$ IF THIS IS DONE, THE SYSTEM ASSUMES YOU WISH EVERY NEW PART OF THE MANUAL SINCE VERSION 52.3 TO BE FLAGGED, SO THAT CHANGES CAN BE EASILY NOTICED.

AT PRINCETON, THE FOLLOWING JOB IS THE LEAST EXPENSIVE WAY TO PRINT A CURRENT MANUAL( WITHOUT ANY FLAGGING ) .....

```
// JOB.....          REGION=40, P=100
// EXEC PSTAT52M
/*
```

\*\*\*\*\*

MAXERROR = 1 \$

A RUN WILL NORMALLY TERMINATE IF FIVE CONSECUTIVE ERRORS OCCUR. THIS EXAMPLE CHANGES THE INITIAL SYSTEM SETTING OF FIVE TO ONE, SO THAT THE RUN WILL END WITH THE FIRST ERROR. MAXERROR CAN BE SET TO ANY INTEGER, AND CAN BE MODIFIED SEVERAL TIMES DURING A RUN.

\*\*\*\*\*

```
PFSOURCE $
PR      PSINIT
PU      PSINOP
PPPU    EXEC
*END
```

PFSOURCE PERMITS THE PRINTING AND/OR PUNCHING OF SPECIFIED PARTS OF THE SOURCE CARD IMAGE FILE. THE ABOVE EXAMPLE CALLS FOR SUBROUTINE 'PSINIT' TO BE PRINTED, 'PSINOP' TO BE PUNCHED, AND 'EXEC' TO BE BOTH PRINTED AND PUNCHED.

PP MEANS PRINT.
PU MEANS PUNCH.

THE FIRST 'PR' OR 'PU' ON A CARD MUST APPEAR IN COLUMNS 1 AND 2. THE SECOND ( IF BOTH ARE USED ) MUST APPEAR IN COLUMNS 3 AND 4. IN EITHER CASE, AT LEAST 1 BLANK MUST PRECEDE THE NAME OF THE DECK TO BE PROCESSED. THIS NAME CANNOT GO BEYOND COLUMN 72.

A NAME CANNOT BE ON TWO CARDS.

```
PR GLOP
PU GLOP
```

THE ABOVE WOULD NOT WORK. ACTUALLY, THE FIRST REQUEST WOULD BE DONE, BUT NOT THE SECOND.

A '\*\*END' CARD MUST FOLLOW THE LAST REQUEST CARD.

SEE 'SOURCE' FOR THE RCO/JCL INFORMATION NEEDED TO ACCESS THE SOURCE FILE AT PRINCETON.

\*\*\*\*\*

PRIMER \$

THIS CAUSES A SHORT ( LESS THAN 20 PAGE ) PRIMER TO BE PRINTED. IT IS SIMPLER THAN THE MANUAL, AND SHOULD BE READ BY NEW P-STAT USERS. IT HAS A NUMBER OF SHORT EXAMPLES. USE OF ...

PRIMER, PUNCH \$

WILL CAUSE THE EXAMPLES TO BE PUNCHED. A USER CAN THEN RUN THE EXAMPLES HIMSELF.

AT PRINCETON, THE FOLLOWING JOB IS THE LEAST EXPENSIVE WAY TO PRINT A CURRENT PRIMER.....

```
// JOB.....          REGION=40, P=20
// EXEC PSTAT52P
/*
```

\*\*\*\*\*

PURGE = A / B / C \$

FILES A, B, AND C ARE TO BE PURGED. A FILE GENERALLY IS USING SPACE IN A P-STAT SCRATCH FILE ON DISK, AND SOME TABLE SPACE IN CORE SO THAT THE P-STAT I/O ROUTINES KNOW WHERE IT IS, ETC. PURGING A FILE FREES UP THE DISK SPACE AND DELETES THE INFORMATION ABOUT IT IN P-STAT'S TABLES. IN GENERAL, P-STAT RUNS FASTER WHEN IT HAS FEWER FILES TO WORRY ABOUT.

ASSIGNED OR ATTACHED FILES (SEE BELOW) ARE NOT THEMSELVES HURT BY A PURGE ( I.E., THEY ARE STILL ON THEIR TAPPS ) BUT P-STAT NO LONGER KNOWS ABOUT THEM.

PURGE \$ MEANS PURGE EVERYTHING.

\*\*\*\*\*

RETAIN = X \$

A USER WOULD DO THIS IF X WERE THE ONLY FILE HE WISHED TO KEEP ON USING. IT IS LIKE SAYING PURGE EVERYTHING BUT X. HAVING DONE THIS NEW FILES CAN, OF COURSE, BE CREATED AND USED.

BOTH PURGE AND RETAIN CAN HAVE ONE OR MORE ARGUMENTS. USERS SHOULD USE THESE OPERATIONS, SINCE IT PERMITS THE SYSTEM TO USE STORAGE MORE EFFECTIVELY. QUALIFICATION OF THE FILE NAMES IS IGNORED, AS IT DOES NOT MAKE ANY SENSE.

NOTE - A FILE CAN BE SAVED ( SEE SAVE AND FIND, BELOW ) AND THEN PURGED. SAVING MEANS THAT IT IS COPIED ONTO THE USER'S PERMANENT DATA TAPE. THE FIND OPERATION, IN THIS OR ANOTHER RUN, CAN FIND IT AND MAKE IT ACCESSABLE FOR P-STAT OPERATIONS. PURGE MEANS THAT

THE SYSTEM FORGETS ABOUT THE FILE FOR THE REST OF THE RUN, THUS FREEING UP THE TAPE OR DISK SPACE USED BY THE FILE.

\*\*\*\*\*

SOURCE \$

THIS PRODUCES A LISTING OF THE P-STAT SOURCE STATEMENTS. THERE IS SOME EDITING ... EACH PROGRAM STARTS ON A NEW PAGE AND THERE IS A TABLE OF CONTENTS WITH PAGE NUMBERS AT THE END. ABOUT 500 PAGES ARE PRODUCED.

AT PRINCETON, THE SOURCE FILE IS NOT KEPT ON A RESIDENT DISK PACK. THEREFORE, THE FOLLOWING CARD MUST BE INCLUDED BETWEEN THE '// JOB' CARD AND THE '// EXEC PSTAT52' CARD ...

/\*SETUP UNIT=2314, ID=88R201

THE '// EXEC PSTAT52' CARD SHOULD INCLUDE ...

// EXEC PSTAT52, SNAME='U. PUHLER.SOURCE52'

\*\*\*\*\*

XHEAD = ( SECONDARY HEADING ) \$

CONSIDER A HEADING WHICH HAS A SLASH (/) SOMEWHERE IN IT. IT IS SOMETIMES USEFUL TO BE ABLE TO CHANGE THE PART OF THE CURRENT HEADING THAT IS TO THE RIGHT OF THE SLASH WITHOUT AFFECTING THE LEFT PART. FOR EXAMPLE.....

HEAD = ( OCT 70, WPRS DATA / CARD 1 ANALYSIS ) \$

XHEAD = ( CARD 2 ANALYSIS ) \$

AFTER THESE TWO CARDS, THE RESULTING HEADING WOULD BE.....

' OCT 70, WPRS DATA / CARD 2 ANALYSIS '

THE FOLLOWING WOULD CAUSE THE NEW TEXT TO REPLACE ALL OF THE OLD HEADING STARTING TO THE RIGHT OF THE SECOND SLASH IN FROM THE RIGHT.

XHEAD = ( SOMETHING ), SLASH = 2 \$

THE NEW HEADING CAN ITSELF CONTAIN SLASHES, ALLOWING IT TO BE PARTIALLY MODIFIED LATER. XHEAD IS FREQUENTLY HELPFUL WHEN NESTED MACROS ARE BEING USED.

6. INPUT-OUTPUT OPERATIONS

BPRINT  
OSIRIS.P  
PLOT  
PRINT  
PUNCH  
SDATA

( NOTE- THE OPERATION 'DATA' IS IN A SFPARATE SECTION )

\*\*\*\*\*

BPRINT = A / .2 / A / .4 / A \$

BPRINT STANDS FOR BLANK PRINT. THE ARGUMENTS ARE THRESHOLD VALUES (GENERALLY FROM 0 TO .4 OF SC) AND FILE NAMPS.

THIS PROGRAM PRINTS A FILE 18 COLUMNS, 50 ROWS TO A PAGE. IT IS INTENDED FOR CORRELATION OR FACTOR MATRICES. EACH SCORE, BEFORE PRINTING, IS MULTIPLIED BY 100 AND PRINTED WITHOUT A DECIMAL POINT. IF, HOWEVER, THE ABSOLUTE VALUE OF THE ORIGINAL SCORE IS LESS THAN THE CURRENT THRESHOLD, THE LOCATION ON THE OUTPUT PAGE WHERE IT WOULD HAVE PRINTED IS SIMPLY LEFT BLANK.

IN THE EXAMPLE, A IS PRINTED 3 TIMES USING DIFFERENT THRESHOLDS (0, .2, AND .4).

VERSION 52 - THE FILE CAN BE UP TO 150 BY 150.

MISSING DATA ( WHICH IS SELDOM FOUND IN THIS TYPE OF FILE ) IS PRINTED AS AN 'X'.

\*\*\*\*\*

OSIRIS.P, OUT = X \$

THIS PROGRAM TAKES AN OSIRIS DATA FILE AND DICTIONARY AND CREATES A P-STAT FILE. THE OSIRIS SYSTEM WAS WRITTEN AT THE INSTITUTE FOR SOCIAL RESEARCH AT THE UNIVERSITY OF MICHIGAN. IT RUNS ONLY ON 360 COMPUTERS. SINCE THIS P-STAT LINK USES SOME OF THEIR SUBROUTINES, THIS LINK WILL ONLY WORK ON A 360.

THE P-STAT STATEMENT MUST BE FOLLOWED BY CERTAIN OSIRIS CARDS....

- 1 - A FILTER CARD ( OPTIONAL )
- 2 - A LABEL CARD
- 3 - VARIABLE LIST CARD(S)

THESE CARDS ARE DESCRIBED IN THE OSIRIS MANUAL.

THE COLUMN LABELS IN THE P-STAT FILE WILL BE OSIR.1, OSIR.2, OSIR.3, ETC.

THE RUN MUST ACCESS A DATA FILE AND A DICTIONARY. THESE ARE FREQUENTLY TWO FILES ON A SINGLE TAPE. TWO 'DD' CARDS ARE NEEDED.

//PSTAT.DICTIN DD ETC., FOR THE DICTIONARY FILE.

//PSTAT.DATAIN DD ETC., FOR THE DATA.

THESE CARDS FOLLOW THE '// EXFC' CARD AND PRECFDE THE '//PSTAT.SYSIN DD \*' CARD.

PLOT = A, ONE \$  
XY Y XX  
\*END

THIS PLOTS SPECIFIED COLUMNS OF FILE A AGAINST EACH OTHER. EACH PLOT USES A PAGE OF OUTPUT. MISSING SCORES ARE SKIPPED.

ONE, IF THERE, PRINTS BOTH AXES FROM +1 TO -1.

SYM, IF THERE, PRINTS BOTH AXES THE SAME, SYMMETRICALLY, IN STEPS OF 0.2 .

IF THE LARGEST ABSOLUTE VALUE IN THE TWO COLUMNS FOR A GIVEN PLOT IS, FOR EXAMPLE, 0.73, BOTH DIMENSIONS OF THE PLOT ARE FROM -0.80 TO +0.80.

IF NEITHER IS THERE, SCALE AS IS.

THERE ARE TWO TYPES OF PLOT CONTROL CARD, A C CARD AND AN XY CARD.

A C CARD HAS C OR BLANK IN EACH COLUMN. IF C IN COLUMN 2, 7, 10, AND 16, PLOTS WILL BE OF VARIABLES 2-7, 2-10, 2-16, 7-10, 7-16, AND 10-16. EACH C IS PLOTTED ONCE AGAINST EVERY OTHER C.

AN XY CARD HAS X, Y, OR BLANK IN EACH COLUMN. IF X IN COLUMN 1, 6, 7, AND Y IN COLUMN 2 AND 4, PLOTS WILL BE OF EACH X VARIABLE AGAINST EACH OF THE Y VARIABLES, 1-2, 1-4, 6-2, 6-4, 7-2, AND 7-4.

THERE CAN BE ANY NUMBER OF THESE CARDS. IT MUST END WITH A \*END CARD.

VERSION 52- FILE A CANNOT HAVE OVER 80 COLUMNS, AND THE PRODUCT ROWS \* (COLUMNS + 2) CANNOT EXCEED 15,000.

PRINT = A / 3 / A / B / 2 / C \$

IDENTIFIERS ARE  
PRINT, AND  
PRINTLOW (SEE BELOW).

THE ARGUMENT LIST CAN CONTAIN SINGLE DIGIT INTEGERS (0 THRU 9) AND NAMES OF FILES, WHICH, AS USUAL, CAN BE QUALIFIED.

ALL FILES OCCURRING BEFORE A DIGIT (IF ANY) IN THE LIST

OF ARGUMENTS ARE PRINTED TO ZERO PLACES. WHENEVER AN INTEGER OCCURS, FILES AFTER IT IN THE LIST ARE PRINTED TO THAT NUMBER OF PLACES (UNTIL ANOTHER INTEGER OCCURS).

THIS EXAMPLE PRINTS A TO NO PLACES, THEN A AND P TO 3 PLACES, THEN C TO 2 PLACES. PRINTING IS 8 VARIABLES AND 50 ROWS TO A PAGE. MISSING DATA IS PRINTED AS AN X (LITERALLY). IF, IN EACH HUNDRED ROWS, ANY ABSOLUTE SCORE ON A VARIABLE IS SO LARGE THAT ITS MOST SIGNIFICANT DIGIT OR ITS SIGN WILL NOT BE PRINTED, THE FORMAT FOR THAT VARIABLE IS CHANGED TO AN F TYPE FORMAT.

THE USUAL FORMAT IS 1X,F13.N, WHERE N IS EITHER ZERO OR THE MOST RECENTLY FOUND DIGIT (0 TO 9) IN THE ARGUMENT LIST. IF A SCORE IS -2344.1 AND N=8, THE 8 PLACES, DECIMAL POINT, 4 HIGH ORDER DIGITS, AND MINUS SIGN MAKE 14 CHARACTERS, AND THE MINUS SIGN COULD, IN SOME COMPUTERS, BE LOST. INSTEAD THE PROGRAM CONVERTS THE FORMAT FOR THAT VARIABLE TO 1X, E13.6, SO THAT -0.234410E+03 (OR -2.344100E+02) WOULD PRINT. SUCH A FORMAT EVALUATION IS MADE EACH HUNDRED ROWS. THEREFORE, THE FIRST 100 ROWS FOR VARIABLES 1-8 MIGHT HAVE VARIABLE 4 IN F FORMAT, AND ON ROWS 101-200 IT MIGHT BE ABLE TO BE PRINTED IN F FORMAT.

A RELATED PROBLEM IS THE VALUE 0.004 WHEN 1X,F13.2 FORMAT IS IN USE, THAT IS, SCORES TOO SMALL TO PRINT IN THE DESIGNATED NUMBER OF PLACES. IF IT IS A CORRELATION MATRIX, THE USER PROBABLY IS SATISFIED WITH 0.00 PRINTING FOR THAT SCORE (I.E., WITH F FORMAT BEING KEPT). HOWEVER, THE USER OF THE IDENTIFIER..... PRINTLOW.... WILL CAUSE F FORMAT TO BE USED WHENEVER THE SMALLEST ABSOLUTE VALUE ON A VARIABLE IS TOO SMALL TO HAVE ITS LEAD DIGIT APPEAR.

PRINT = A / 2 / B , PRINTLOW \$ WILL PRINT A AND B IN THAT MANNER.

ANY SIZE FILE CAN BE PRINTED.

VERSION 52 - UP TO 200 COLUMNS ARE PRINTED IN A PASS THROUGH THE FILE. IN OTHER WORDS, IF A FILE HAS 420 COLUMNS, THE PROGRAM WILL AUTOMATICALLY MAKE THREE PASSES THROUGH THE FILE.

\*\*\*\*\*

PUNCH=A, CARDS=1 / 10 / (10F7.2) / 11 / 30 / (20F3.0) \$

THIS PUNCHES A FILE ONTO CARDS, IN A FORMAT WHICH CAN BE USED AS INPUT TO THE DATA PROGRAM. THE FIRST CARDS THAT ARE PUNCHED ARE CONTROL CARDS FOR THE DATA PROGRAM ( DATA= \*LAB \*FMT \*MISS AND \*CARDS ) AND THEN THE FILE IS PUNCHED, BY ROWS, WITH TWO LIMITATIONS ...

1- ONLY COLUMNS 11-80 ARE AVAILABLE FOR DATA. COLUMNS 1-10 ARE USED FOR THE ROW LABEL AND FOR SEQUENCING.

2- ONLY 8 CARDS PER ROW CAN BE USED.

A '\*END\*' CARD IS PUNCHED AFTER THE LAST ROW IS FINISHED.



IN THE EXAMPLE, FILE A IS PUNCHED. 'CARDS=' DESCRIBES THE OUTPUT CARDS, USING 3 ARGUMENTS PER OUTPUT CARD.

- 1- THE FIRST VARIABLE TO BE PUNCHED ON THAT CARD.
- 2- THE LAST VARIABLE TO BE PUNCHED ON THAT CARD.
- 3- THE FORMAT FOR THOSE VARIABLES, USING F, X, OR E FORMAT. THIS DESCRIBES COLUMNS 11-80 ON THE CARD TO BE PUNCHED.

IN THE EXAMPLE, VARIABLES 1-10 WILL BE PUNCHED IN THE FIRST CARD IN 2A4,I2,10F7.2 FORMAT. THE 2A4,I2, ARE GENERATED INTERNALLY. THEN VARIABLES 11-30 ARE PUNCHED ON THE SECOND CARD, USING 2A4,I2,20F3.0 FORMAT. TWO CARDS ARE PUNCHED FOR EACH ROW OF FILE A.

\*\*\*\*\*

```
SDATA = B, NV = 10, FORMAT = ( 10F2.0 ), PRINT $
(DATA CARDS)
*END
```

THIS IS A SIMPLE DATA PROGRAM. IT LACKS THE FEATURES OF THE DATA PROGRAM, BUT IT IS FAST.

SDATA= THE NAME OF THE OUTPUT FILE.

NV = THE NUMBER OF VARIABLES IN THE OUTPUT FILE. CANNOT EXCEED 80.

FORMAT = (FORMAT). THE FORMAT ITSELF CAN BE UP TO 160 CHARACTERS. IT CAN ONLY USE X, F, OR E TYPE SPECIFICATION IN THE FORMAT.

PRINT, THE INPUT CARDS ARE LISTED. IF THIS OPTION IS USED.

TAPE = A FILE NUMBER WHERE THE CARDS ARE, IF NOT WITHIN THE P-STAT DECK. IT IS NOT NECESSARY, IF THE CARDS ARE ON A TAPE, FOR A '\*END' CARD TO BE THE FINAL CARD THERE.

THERE CAN ONLY BE 1 DATA CARD PER CASE. THERE IS NO PROVISION FOR LABELS (THEY GET GENERATED), NOR FOR MISSING OR INVALID DATA. IF THESE ARE A PROBLEM, USE THE DATA PROGRAM.

DO NOT FORGET THE \*END CARD AFTER THE LAST DATA CARD, IF THE CARDS ARE WITHIN THE P-STAT INPUT DECK.

7. SAVING AND USING P-STAT FILES ON TAPES

```

NEWPDT
SAVE
FIND
COPY.PDT
EDIT
TRIM
-
ASSIGN
ATTACH
-
REWIND

```

NOTE - OF THESE, SAVE IS THE ONLY OPERATION THAT PERMITS QUALIFICATION OF FILE NAMES.

\*\*\*\*\*

NEWPDT, TAPE = 62 \$

CREATE A NEW PERMANENT DATA TAPE (I. E., PUT IT INTO P-STAT DATA FORMAT).

TAPE (OR FILE) 62 IS REWOUND AND A DUMMY RECORD IS WRITTEN, DEFINING, IN EFFECT, THE END OF THE P-STAT DATA ON THE TAPE. 'NEWPDT' ONLY MAKES THE TAPE READY FOR FILES. THE 'SAVE' OPERATION CAN THEN BE USED TO STORE FILES ON THAT TAPE FOR USE DURING THIS OR SUBSEQUENT RUNS. THE USER SHOULD CLEARLY NOTE THAT, WHEN A NEWPDT IS DONE, ANY INFORMATION PREVIOUSLY ON THE TAPE IS LOST.

SEVERAL PERMANENT DATA TAPES CAN BE USED AT THE SAME TIME. 61 THROUGH 69 ARE RESERVED FOR PERMANENT DATA TAPES (WHICH CAN BE EITHER TAPES OR DISK FILES ON THE 360).

IF NO 'TAPE=N' IS FOUND, 61 IS ASSUMED. THIS MULTIPLE PDT FEATURE PERMITS COPYING OF ALL OR SELECTED PARTS (USING 'FIND') OF SEVERAL PDT'S ONTO ANOTHER. ANOTHER USE IS TO CHECKPOINT FILES DURING A RUN ONTO ONE PDT, USING ANOTHER PDT FOR SERIOUS SAVING. ( NOTE - THE OPERATION 'COPY.PDT' IS THE BEST WAY TO COPY A PDT. )

THE NUMBERS 61-63 ARE THOSE PDT NUMBERS EASIEST TO USE WITH PRINCETON'S 360 CATALOGED PROCEDURES FOR P-STAT USE.

'NEWPDT' IS NECESSARY FOR NEW 'SAVE' TAPES, BUT SHOULD NOT BE USED FOR AN 'ASSIGN' TAPE.

\*\*\*\*\*

SAVE = A / B (B 1-30) \$

THIS COPIES FILES ONTO A USER'S PERMANENT DATA TAPE.

FIRST A, AND THEN ROWS 1-30 OF B WILL BE WRITTEN (AS SEPARATE P-STAT FILES) ONTO A PDT (PERMANENT DATA TAPE), BEYOND ANY OTHER FILES ALREADY THERE.

AS MANY FILES AS CAN FIT CAN GO ONTO A PDT.

A 2400 FOOT TAPE CAN HOLD ABOUT 5000 BLOCKS (OF 465 WORDS EACH).  
THUS, A FILE OF 4000 ROWS BY 120 COLUMNS WOULD USE ABOUT 450 FEET OF  
THE TAPE.

FILE NAMES BEING SAVED CAN BE QUALIFIED IN ANY WAY.

SEE 'NEWPDT' REGARDING THE 'TAPF=' OPTION.  
SAVE \$ COPIES ALL CURRENT FILES ONTO THE DESIGNATED PDT.

NEWPDT, TAPE = 62 \$ SAVE, TAPF = 62 \$  
WOULD COPY ALL CURRENTLY ACTIVE FILES ONTO A NEW TAPE 62.

\*\*\*\*\*

FIND = A / B / C \$

THIS EXAMPLE ASSUMES PDT 61 IS PRESENT. FILES A, B, AND C WILL BE  
FOUND ON THE PDT AND COPIED AS IS INTO THE SYSTEM. QUALIFICATION OF  
THE FILES AT THE TIME THEY ARE FOUND AND BROUGHT INTO THE SYSTEM IS NOT  
POSSIBLE.

A FIND CAN BE DONE AT ANY POINT DURING A RUN.

IN THE ABOVE EXAMPLE, FILES A, B, AND C ARE STILL ON THE PERMANENT  
DATA TAPE AFTER THE FIND IS DONE. HOWEVER, BECAUSE OF THE FIND  
OPERATION, THEY ARE ALSO ON DISK ( OR A SCRATCH TAPE ) IN A FORM  
DIRECTLY ACCESSABLE TO THE OTHER P-STAT OPERATIONS.

3 FORMS OF FIND ARE POSSIBLE. ..

- 1 - FIND \$ THIS FINDS EVERY FILE ON THE PDT AND MAKES THOSE FILES  
USABLE ON THIS RUN. (THEY ARE COPIED INTO DISK, WHICH, IF  
MANY FILES ARE THERE, CAN TAKE SOME TIME.) TWO FILES WITH  
THE SAME NAME ARE NOT ALLOWED IF FIND \$ IS USED.
- 2 - FIND = A/R \$ THESE ARE FOUND AND COPIED. AGAIN, THERE  
MUST BE ONE AND ONLY ONE 'A', AND ONE AND ONLY ONE 'R', IF THIS  
FORM IS USED.
- 3 - FIND = 3/X/R/12/Y \$ THIS FORM PERMITS A FILE ON THE PDT  
TO BE RENAMED AS IT IS FOUND. IN OTHER WORDS, THE NEW NAME  
IS GIVEN TO IT AS IT IS COPIED (THE PDT ITSELF IS UNCHANGED).  
IF TWO OR MORE FILES ON THE PDT HAVE THE SAME NAME, THE USE  
OF 'EDIT \$' REVEALS THEIR POSITIONS ON THE PDT. THE ABOVE  
EXAMPLE FINDS FILE 3 AND RENAMES IT X, FINDS FILE B NORMALLY  
WHEREVER IT IS, AND FINDS FILE 12, RENAMING IT Y.

SEE 'NEWPDT' REGARDING THE 'TAPF=' OPTION.

\*\*\*\*\*

COPY.PDT, NEWTAPF = 62 \$

THIS MAKES A COPY OF PERMANENT DATA TAPE 61 ONTO TAPF 62. THIS IS  
MUCH FASTER THAN USING...

FIND \$ NEWPDT, TAPP = 62 \$ SAVE, TAPE = 62 \$

IF THE INPUT TAPE IS NOT 61, 'TAPE=' MUST BE USED, FOR EXAMPLE...

COPY.PDT, TAPE = 62, NEWTAPE = 63 \$

SEE 'NEWPDT' REGARDING THE 'TAPP' OPTION.

\*\*\*\*\*

EDIT \$

THE PDT IS READ THROUGH AND A REPORT IS PRINTED, LISTING THE NAMES AND SIZES OF ALL FILES ON THE TAPE.

IF, FOR EXAMPLE, 'EDIT, LABELS \$' WERE USED, THE COLUMN LABELS FOR EACH FILE WOULD ALSO BE PRINTED.

SEE 'NEWPDT' REGARDING THE 'TAPE=' OPTION.

\*\*\*\*\*

TRIM = 3 \$ OR TRIM = B \$

THE PDT HAS SOME NUMBER OF FILES ON IT, SOME OF WHICH ARE USEFUL AND SOME ARE NOT. TRIM PERMITS ALL FILES BEYOND A CERTAIN ONE TO BE SCRATCHED.

TRIM = 3 \$ TRIMS THE TAPE SO THAT THE THIRD FILE IS THE LAST ON THE TAPE.

TRIM = B \$ CAUSES FILE B TO BE THE LAST FILE ON THE TAPE. THE LAST BLOCK OF THE LAST FILE TO BE KEPT IS FOUND, AND A DUMMY BLOCK THAT SIGNIFIES THE END OF P-STAT DATA IS WRITTEN NEXT. AGAIN, QUALIFICATION OF THE FILE NAME IS NOT PERMITTED.

THESE SIX OPERATIONS, 'NEWPDT', 'SAVE', 'COPY.PDT', 'FIND', 'EDIT', AND 'TRIM' ARE THE ONLY OPERATIONS THAT USE A PDT.

SEE 'NEWPDT' REGARDING THE 'TAPE=' OPTION.

\*\*\*\*\*

ASSIGN = XX, TAPE = 51 \$

WHEN A USER HAS A LARGE BASIC FILE THAT IS TO BE USED FREQUENTLY, ON DIFFERENT RUNS, IT IS DESIRABLE TO HAVE THE FILE ON TAPE INSTEAD OF ON DISK, AND ALSO TO HAVE THE FILE (ALREADY IN P-STAT FORM) ON A SEPARATE TAPE AND THEREBY SKIP THE 'FIND' COPYING TIME. THE FILE WILL, AT SOME POINT, BE CREATED BY SOME P-STAT OPERATION.

THE 'ASSIGN' OPERATION INFORMS THE P-STAT SYSTEM THAT SUCH A TAPE IS TO BE CREATED. THE 'ATTACH' OPERATION, DESCRIBED NEXT, ALLOWS THE TAPE TO BE USED ON A SUBSEQUENT RUN.

THE ASSIGN STATEMENT MUST OCCUR BEFORE THE FILE IS CREATED.

TAPF NUMBERS 51-59 SHOULD BE USED FOR ASSIGN OR ATTACH. NUMBERS 51-55 ARE THOSE ASSIGN/ATTACH NUMBERS EASIEST TO USE WITH PRINCETON'S CATALOGED PROCEDURES FOR P-STAT USE.

THE ABOVE ASSIGN OPERATION TELLS THE SYSTEM THAT, WHEN FILE XX IS CREATED, IT SHOULD GO ONTO TAPF 51 AS THE ONLY FILE ON THAT TAPE. TAPF 51 IS NOT USED BY P-STAT UNTIL THEN.

THE ASSIGNED FILE, ONCE CREATED, CAN BE USED IN A NORMAL WAY DURING THE SAME RUN.

\*\*\*\*\*

ATTACH = XX. TAPE = 55 \$

ON A SUBSEQUENT RUN, THE TAPE WITH XX ON IT (DUE TO AN ASSIGN ON A PREVIOUS RUN) MIGHT BE DEFINED TO THE OPERATING SYSTEM AS FILE 55. THIS OPERATION TELLS THE SYSTEM THAT FILE XX EXISTS AND IT IS ON TAPE 55. WHEREVER XX IS USED AFTER THIS, DURING THIS RUN, THE SYSTEM WILL FIND IT THERE.

TAPF NUMBERS 51-59 SHOULD BE USED FOR ASSIGN OR ATTACH. NUMBERS 51-55 ARE THOSE ASSIGN/ATTACH NUMBERS EASIEST TO USE WITH PRINCETON'S CATALOGED PROCEDURES FOR P-STAT USE.

A GIVEN TAPE COULD BE ATTACHED AS 51 ON ONE RUN AND, FOR EXAMPLE, 52 ON ANOTHER RUN. THE NUMBER DOES NOT HAVE TO BE THE SAME NUMBER THAT WAS USED WHEN IT WAS ASSIGNED. THE NAME, HOWEVER, MUST MATCH THE NAME OF THE FILE ON THE TAPF.

THE ASSIGN-ATTACH FEATURE CAN GREATLY IMPROVE P-STAT PERFORMANCE WHEN SEVERAL LARGE FILES ARE BEING USED ON REPEATED RUNS.

IT MIGHT BE NOTED THAT AN ATTACH TAPE LOCKS EXACTLY LIKE A PDT WITH ONLY ONE FILE ON IT. IF ONE WISHED, THEY COULD BE INTERCHANGED. HOWEVER, THE 'FIND' OPERATION COPIES THE FILE ONTO A DISK RATHER THAN ACCESSING IT DIRECTLY. THEREFORE USING 'FIND' FOR AN ATTACH TAPE WILL TAKE FAR MORE TIME (PERHAPS MINUTES COMPARED TO LESS THAN A SECOND) AND MAY USE MORE SPACE ON THE DISK THAN IS SUPPLIED FOR IT IN THE CATALOGED PROCEDURES.

\*\*\*\*\*

REWIND = 91 \$

TAPF 91 IS REWOUND. REWINDING A TAPE IS SOMETIMES DONE BEFORE USING THE 'TAPE =' OPTION IN 'DATA'.

8. FILE MANIPULATION OPERATIONS

COLLATE  
 CORESORT  
 DIF  
 LAG  
 IRFILL  
 MATCH  
 MERGE  
 MUDMERGE  
 NO.OP  
 NORM.COL  
 NORM.ROW  
 SCAN  
 SMERGE

\*\*\*\*\*

COLLATE, LEFT=A, RIGHT=B, OUT=C\$

THIS PROGRAM IS VERY MUCH LIKE A LEFT-RIGHT MERGE, EXCEPT IT PERMITS SOME OF THE ROWS OF THE RIGHT HAND FILE TO BE MISSING.

- LEFT = NAME OF LEFT HAND DATA FILE. THIS CAN HAVE ANY NUMBERS OF ROWS.
- RIGHT = NAME OF RIGHT HAND FILE (WITH FEWER ROWS THAN THE LEFT HAND FILE). THE COMBINED COLUMNS OF LEFT AND RIGHT CANNOT EXCEED THE SYSTEM MAXIMUM.
- OUT = NAME FOR THE OUTPUT FILE.

ALL LABELS OF ROWS OF THE RIGHT HAND FILE ALSO OCCUR IN ROWS OF THE LEFT HAND FILE, AND THEY MUST OCCUR IN THE SAME ORDER. ALL ROWS OF THE LEFT HAND FILE THAT DO NOT HAVE A ROW OF THE RIGHT HAND FILE WITH THE SAME LABEL WILL HAVE THE MISSING DATA CONSTANT GENERATED FOR VARIABLES OF THE RIGHT HAND FILE.

THIS OPERATION IS USEFUL FOR THE FOLLOWING SITUATION. ASSUME THAT ONE HAS ONE THOUSAND CASES AND FORTY VARIABLES AS ONE DATA FILE. ONE ALSO HAS DATA ON TWENTY ADDITIONAL VARIABLES FOR FIVE HUNDRED OF THE PEOPLE. WHAT IS DESIRED IS A FILE OF ONE THOUSAND PEOPLE BY SIXTY VARIABLES, WITH THE FIVE HUNDRED ROWS OF THE NEW TWENTY VARIABLES CONTAINING DATA AND THE OTHER FIVE HUNDRED ROWS CONTAINING THE MISSING DATA CONSTANT.

\*\*\*\*\*

CORESORT, IN=A, OUT=B \$

IN-CORE SORT. THIS PROGRAM ORDERS A FILE ACCORDING TO ONE OF ITS COLUMNS, OR ACCORDING TO ITS ROW LABELS.

IN = NAME OF INPUT FILE. IT MUST FIT IN CORE.  
 OUT = NAME FOR OUTPUT FILE.  
 LABEL = COLUMN TO SORT ON. IF NOT USED, THE PROGRAM WILL SORT ON  
 THE ROW LABELS.  
 IF, FOR EXAMPLE, LABEL=AGE, SORT ON VARIABLE AGE.

VERSION 52 - THE PRODUCT NR \* (NC+2) OF 'IN' CAN BE UP TO 23,000.

FOR EXAMPLE, IF A IS 100 BY 40, IT USES 4200  
 CORE LOCATIONS. THE EXTRA 2 IS THE 2 WORD  
 LABEL FOR EACH ROW.

\*\*\*\*\*

DIF, IN=A, OUT=B, V1=7, V2=9, STEP=1 \$

GET THE DIFFERENCE BETWEEN A VARIABLE ON ONE ROW AND  
 THE SAME VARIABLE ON SOME LATER ROW. IF A IS 100 BY 10, AND  
 WE WISH DIFFERENCES ON VARIABLES 7, 8, AND 9 AT A STEP OF ONE  
 ROW, WE GET A 99 BY 13 FILE OUT. THE FIRST ROW HAS SCORES  
 (1,1) TO (1,10) AS THEY WERE READ IN, BUT (1,11) IS (1,7) -  
 (2,7). (1,12) IS (1,8) - (2,8), AND SO ON. IF ANYTHING  
 INVOLVED IN A NEW SCORE IS MISSING, THE NEW SCORE IS MISSING.

IN = INPUT FILE, ANY SIZE.  
 OUT = OUTPUT FILE.  
 V1 = FIRST VARIABLE TO BE DIFFED.  
 V2 = LAST VARIABLE TO BE DIFFED.  
 STEP = STEP SIZE.

VERSION 52 - STEP SIZE CANNOT EXCEED 10.

\*\*\*\*\*

LAG, IN=A, OUT=B, V1=7, V2=9, STEP=2 \$

APPEND DATA OF A LATER ROW TO AN EARLIER ROW. IT  
 WORKS JUST LIKE DIF EXCEPT THERE IS NO SUBTRACTION.

VERSION 52 - STEP SIZE CANNOT EXCEED 10.

\*\*\*\*\*

LRMILL, LEFT=A, RIGHT=B, OUT=C \$

THIS DOES A LEFT-RIGHT MERGE OF 2 INPUT FILES.  
 HOWEVER, IF ONE HAS MORE ROWS THAN THE OTHER, THE SHORTER HAS  
 ROWS OF MISSING DATA FILLED IN. LABELS DO NOT HAVE TO LINE UP,  
 AND, IN FACT, ARE NOT CHECKED AT ALL.

THE RESULTING NUMBER OF COLUMNS CANNOT  
 EXCEED THE SYSTEM MAXIMUM.

\*\*\*\*\*

MATCH, IN=B, CONTROL=A, OUT=C \$

THIS OPERATION IS IN SOME RESPECTS SIMILAR TO COLLATE. IN THIS OPERATION THERE IS ONE FILE, A, WHICH IS TREATED AS A CONTROL FILE. THE GOAL OF THE OPERATION IS TO RETAIN THOSE ROWS OF B THAT HAVE A LABEL THAT MATCHES A LABEL OF A ROW OF A, AND, AT THE SAME TIME, ORDERING THOSE ROWS OF B THAT ARE RETAINED IN THE SAME ORDER AS THEIR MATCHING LABELS OCCURRED IN A.

B (THE INPUT FILE) IS READ AND HELD IN CORE. THEN A PASS IS MADE THROUGH THE CONTROL FILE. A ROW OF IT IS READ AND ITS LABEL IS COMPARED WITH EVERY ROW LABEL OF B UNTIL A MATCH IS FOUND. THAT ROW OF B IS WRITTEN TO C. A NEW ROW OF A IS READ, AND SO ON. A ROW OF B IS ONLY USED ONCE (IF AT ALL).

CONTROL = NAME OF INPUT CONTROL FILE (A).  
IN = NAME OF INPUT DATA FILE (B).  
OUT = NAME OF OUTPUT DATA FILE (C). THIS WILL CONSIST OF THE ROWS OF B THAT HAVE A LABEL THAT MATCHES A LABEL OF A, ORDERED BY THE ORDER OF THE LABELS OF A.

VERSION 52 - THE PRODUCT NR \* (NC+2) OF B CAN BE UP TO 23,000. THE 2 IN NC+2 ALLOWS ROOM FOR THE LABEL OF EACH ROW.

FOR AN EXAMPLE OF HOW TO USE THIS OPERATION, CONSIDER TWO DATA FILES...

X HAS 300 PEOPLE BY 15 VARIABLES, Y HAS 400 PEOPLE BY 10 VARIABLES, THE VARIABLES ARE DIFFERENT, AND 200 OF THE PEOPLE EXIST IN BOTH FILES. WHAT IS DESIRED IS A 200 BY 25 FILE.

THE FIRST OPERATION WOULD BE TO MATCH X, THE 300 BY 15 FILE, AS THE CONTROL FILE AGAINST Y, THE 400 BY 10 FILE. SINCE 200 OF THE PEOPLE MATCH, THE OUTPUT FILE (C) WOULD BE OF ORDER 200 BY 10. THESE WOULD BE THE 200 PEOPLE OF Y WHO WERE FOUND AMONG THE 300 PEOPLE IN X. THEY ARE NOW IN THE SAME ORDER AS THEY OCCUR IN X. NOW, AS A NEW STEP, IF THIS OUTPUT FILE IS USED AS THE CONTROL FILE, WE MATCH A 200 BY 10 FILE AS THE CONTROL FILE AGAINST THE 300 BY 15 FILE, YIELDING A 200 BY 15 FILE. THESE TWO FILES ARE THEN MERGED TOGETHER, SINCE THE SAME PEOPLE ARE IN THE SAME ORDER, AND A 200 BY 25 FILE IS PRODUCED. NOTE, IT TOOK TWO MATCHES AND A MERGE TO DO THIS.

\*\*\*\*\*

MERGE, UP = A, DOWN = B, OUT = C \$

FILE C WILL BE THE ROWS OF A, FOLLOWED BY THE ROWS OF B. A AND B MUST HAVE THE SAME NUMBER OF COLUMNS. THEIR COLUMN LABELS

MUST MATCH (UNLESS THE IDENTIFIER NOMATCH WAS ALSO USED).

MERGE, LEFT = A, RIGHT = B, OUT = C \$

FILES A AND B MUST HAVE THE SAME NUMBER OF ROWS, AND THEIR COMBINED COLUMNS MUST NOT EXCEED THE MAXIMUM NUMBER OF COLUMNS THAT THE SYSTEM PERMITS. THEIR ROW LABELS MUST MATCH UNLESS.... MERGE, LEFT=A, RIGHT=B, OUT=C, NOMATCH \$

A 3-WAY LEFT-RIGHT MERGE IS POSSIBLE...

MERGE, LEFT = A, MIDDLE = B, RIGHT = C, OUT = D \$

\*\*\*\*\*

MUDMERGE, IN = A / B ( P 1-50 ) / C / D, OUT = E \$

MULTIPLE UP-DOWN MERGE. A NUMBER OF FILES ( WITH QUALIFICATION PERMITTED ) ARE UP-DOWN MERGED TOGETHER. THE COLUMN LABELS MUST MATCH UNLESS THE IDENTIFIER 'NOMATCH' IS USED. THE INPUT FILES MUST EACH HAVE THE SAME NUMBER OF COLUMNS.

\*\*\*\*\*

NO.OP, IN = A(C1-40), OUT = P \$

THIS IS A NULL OPERATION, IN A SENSE. IT READS AN INPUT FILE AND PRODUCES AN OUTPUT FILE. THE INPUT FILE CAN ( AND FREQUENTLY WILL ) HAVE QUALIFICATION. IF ONE HAS A FILE AND WISHES TO RE-ORGANIZE IT ( USING ROW, COLUMN AND IF EXPRESSIONS, FOR EXAMPLE ), ONE MIGHT NO.OP THE FILE AND GET THE QUALIFIED FILE AS OUTPUT. THIS COULD BE PRINTED OR SCANNED, AND SAVED TO BE USED IF THE PRINTOUT LOOKS REASONABLE.

\*\*\*\*\*

NORM.COL, IN = A, OUT = B \$

THIS SCALES EACH COLUMN OF THE INPUT FILE SO THAT THE SQUARES OF THE ELEMENTS IN EACH COLUMN OF THE OUTPUT FILE WILL SUM TO 1.

ANY SIZE FILE IS ALLOWED.  
MISSING DATA IS NOT ALLOWED.

\*\*\*\*\*

NORM.ROW, IN = A, OUT = B \$

THIS SCALES EACH ROW OF THE INPUT FILE SO THAT THE SQUARES OF THE ELEMENTS IN EACH ROW OF THE OUTPUT FILE WILL SUM TO 1.

ANY SIZE FILE IS ALLOWED.  
MISSING DATA IS NOT ALLOWED.

\*\*\*\*\*

SCAN, IN = A, DES = D \$

THIS GETS A DESCRIPTION FILE OF THE INPUT FILE.

IN = AN INPUT DATA FILE.

OUT = A COPY OF THE ACTUAL INPUT FILE. THE USE OF IN, DES, AND OUT WILL BE HELPFUL WHEN THE INPUT FILE IS SUBSTANTIALLY MODIFIED BY ROW/COLUMN/IF PHASES AND THE RESULT, WITH ITS DESCRIPTION, NEEDS TO BE REUSED.

DES = AN OUTPUT DESCRIPTION FILE OF THE INPUT FILE. THIS WILL HAVE 6 COLUMNS (MEAN, VARIANCE, STANDARD DEVIATION, LOW, HIGH, AND N). IT WILL HAVE 1 ROW FOR EACH VARIABLE (COLUMN) OF THE INPUT FILE.

SCAN CAN BE USED, WITH OTHER IDENTIFIERS, TO PLUG IN MEANS FOR MISSING DATA, AND/OR TO STANDARDIZE DATA, THAT IS, GIVING EACH COLUMN A MEAN OF ZERO AND VARIANCE OF ONE. THIS CAN BE DONE EITHER USING THE MEANS, ETC. OF THE DATA ITSELF, OR ANOTHER MATRIX OF MEANS AND STANDARD DEVIATIONS CAN BE USED. THUS, IF AN ORIGINAL DATA FILE HAS 1,000 CASES AND MEANS AND STANDARD DEVIATIONS ARE COMPUTED ON IT, AND ANOTHER 20 CASES ARE BROUGHT INTO THE SYSTEM AT A LATER DATE, THOSE 20 CASES CAN BE STANDARDIZED, USING THE ORIGINAL MEANS AND STANDARD DEVIATIONS.

INDES = NAME OF MATRIX OF MEANS, STANDARD DEVIATIONS, TO BE USED IF THIS MATRIX ALREADY EXISTS IN THIS SYSTEM. IF NOT USED, THE PROGRAM WILL COMPUTE THE MEANS, STANDARD DEVIATIONS OF THE DATA MATRIX ITSELF AND USE THOSE TO STANDARDIZE IT OR TO PLUG IN MEANS.

MDATA = NAME FOR OUTPUT FILE OF DATA WITH MEANS PLUGGED IN FOR MISSING SCORES.

SDATA = NAME FOR STANDARDIZED DATA FILE. MISSING SCORES IN THE IN FILE BECOME ZEROS IN THE SDATA FILE ( IN THE SPIRIT OF PLUGGING MEANS FOR MISSING SCORES), UNLESS.....,

STAYMISS, IF USED, MISSING SCORES IN THE 'IN' FILE STAY MISSING IN THE 'SDATA' FILE.

\*\*\*\*\*

SMERGE, A11=A, A12=B, A22=C, OUT=ABC \$

THIS OPERATION, SYMMETRIC MERGE, FEEDS IN A SYMMETRIC UPPER LEFT MATRIX, AN ASYMMETRIC UPPER RIGHT MATRIX AND A SYMMETRIC LOWER RIGHT MATRIX. THE LOWER LEFT IS FILLED IN WITH THE TRANSPOSE OF THE UPPER RIGHT, SO THAT THE OUTPUT IS

SYMMETRIC.

VERSION 52 - OUT CANNOT EXCEED 150 BY 150.

A11 = NAME OF UPPER LEFT SYMMETRIC INPUT MATRIX .  
A12 = NAME OF UPPER RIGHT ASYMMETRIC MATRIX, WHOSE ROW  
LABELS MATCH THOSE OF A11.  
A22 = NAME OF LOWER RIGHT SYMMETRIC INPUT MATRIX, WHOSE  
COLUMN LABELS MATCH THOSE OF A12.  
OUT = NAME FOR THE OUTPUT MATRIX.

\*\*\*\*\*

MULTIPLY, PRE=A, POST=B, OUT=C \$

MATRIX MULTIPLY.

C = A TIMES B.

VERSION 52 - A CAN BE ANY NUMBER OF ROWS BY UP TO 150 COLUMNS.  
B CAN BE UP TO 150 BY 150.

\*\*\*\*\*

TRANSPOS, IN = A, OUT = B \$

FILE A IS TRANSPOSED. THE RESULT IS NAMED B.

SINCE THE ROWS OF IN BECOME THE COLUMNS OF OUT, THE NUMBER  
OF ROWS OF IN CANNOT EXCEED THE MAXIMUM NUMBER OF COLUMNS  
PERMITTED BY THE SYSTEM.

VERSION 52 - 450 COLUMNS MAXIMUM.

10. COPRELATION OPERATIONS

BISERIAL  
INTCDS  
INTMDA  
INTMDS  
TFT

\*\*\*\*\*

BISERIAL, IN=A, OUT=B, NCV=12 \$

BISERIAL CORRELATION. A BISERIAL CORRELATION IS USED WHEN ONE VARIABLE IS DISTRIBUTED CONTINUOUSLY AND THE OTHER VARIABLE HAS BEEN DICHOTOMIZED (THAT IS, SCORES ARE EITHER ZERO OR ONE, OR SOME SUCH ARRANGEMENT).

IN = NAME OF FILE OF INPUT DATA. IT SHOULD HAVE CONTINUOUS VARIABLES PRECEDING DICOTOMOUS VARIABLES. A (C) PHRASE CAN ACCOMPLISH THIS, IF NECESSARY. MISSING DATA IS ALLOWED.

NCV = NUMBER OF CONTINUOUS VARIABLES.

OUT = NAME FOR RESULT. A BISERIAL CORRELATION MATRIX.

ZERO = SUBSTITUTE VALUE FOR ZERO. NOT NECESSARY IF THE DATA IS INDEED IN 0,1 FORM. HOWEVER, IF THE DATA IS FOR EXAMPLE IN 1,3 FORM THEN THE VALUE WOULD BE EITHER 1. OR 3., DEPENDING UPON WHICH SCORE IS INTENDED TO BE REPRESENTING THE ABSENCE OF THE QUALITY THAT THE DICOTOMOUS SCORE USUALLY IMPLIES. ONE CAN REVERSE THE SIGN OF BISERIAL INTERCORRELATIONS INVOLVING 0,1 DATA BY SAYING ZERO=1.

IF DATA IS 0,1 AND ZERO IS NOT MODIFIED, A PLUS BISERIAL SHOWS COPRELATION OF HIGH CONTINUOUS SCORES WITH OCCURANCES OF 1'S.

THE OUTPUT MATRIX IS NECESSARILY ASYMMETRIC. THE CONTINUOUS VARIABLES MUST BE IN THE FIRST COLUMNS OF 'IN' AND WILL BE LABELED AS ROWS OF THE OUTPUT. THE DICOTOMOUS VARIABLES WILL BE LABELED AS THE COLUMNS OF THE OUTPUT.

VERSION 52 - THE PRODUCT OF ROWS AND COLUMNS CANNOT EXCEED 4500.

HOWEVER, THE SHAPE CAN BE, FOR EXAMPLE, 50 BY 30, OR 30 BY 50, OR 150 BY 10. MISSING DATA IS ALLOWABLE IN ANY VARIABLE, AND EACH INDIVIDUAL BISERIAL CORRELATION IS COMPUTED ON THE DATA AVAILABLE TO IT.

NMAT - OUTPUT FILE SHOWING THE N FOR EACH CORRELATION.

CROSS - OUTPUT CROSSPRODUCTS ( NOTE, THESE ARE NOT CORRECTED FOR THE VARYING N IN EACH CELL ).

COV - OUTPUT COVARIANCES.

COR - OUTPUT CORRELATIONS.

ROWS - INPUT VALUE WHICH MUST BE SUPPLIED. IF 'IN' HAS 60 COLUMNS AND 'ROWS = 20' IS SUPPLIED, THE FIRST 20 VARIABLES OF 'IN' WILL BE CORRELATED WITH THE REMAINING 40, CREATING NMAT, COV, AND COP OF ORDER 20 BY 40. THE COLUMN SELECTION FEATURE IS USEFUL HERE.

NOROW1, SEE INTDCS

VERSION 52 - THE OUTPUT IS NR BY NC, WHERE NR \* NC CANNOT BE MORE THAN 3600. THUS, 30 BY 40, OR 120 BY 30, BUT NOT 70 BY 70.

SOME EXAMPLES...

INTMDA, IN= A (C 21-30, 51-70), COR=B, ROWS=10 \$  
COR WILL BE 10 BY 20, CONTAINING THE CORRELATIONS OF VARIABLES 21-30 WITH 51-70.

INTMDA, IN=A (C 41-60, 41-60), COR=B, ROWS=20 \$  
THIS WILL WORK. B WILL BE 20 BY 20 AND WILL IN FACT BE SYMMETRIC.

INTMDA, IN=A (C 21-100, AGE), COR=B, ROWS= 80 \$  
B WILL BE A VECTOR, 80 BY 1, OF VARIABLES 21-100 CORRELATED WITH VARIABLE AGE.

\*\*\*\*\*

INTMDS, IN = A, COR = AIN \$

INTERCORRELATE, MISSING DATA, SYMMETRIC.

VERSION 52 - THE INPUT FILE CAN HAVE UP TO 75 COLUMNS.

PEARSON PRODUCT MOMENT INTERCORRELATION, ALLOWING MISSING DATA, PRODUCING SYMMETRIC OUTPUT. EACH CORRELATION COEFFICIENT IS BASED UPON THE NUMBER OF CASES WHICH HAD NON-MISSING DATA ON THE TWO VARIABLES REPRESENTED IN THAT INTERCORRELATION COEFFICIENT. IN OTHER WORDS, MISSING DATA IS NOT USED IN ANY WAY, NOR IS IT ESTIMATED. A CORRELATION MATRIX MAY POSSIBLY BE BASED UPON A DIFFERENT N FOR EACH ELEMENT.

IN = NAME OF INPUT DATA FILE.

\*\*\*\*\*

INTCDS, IN = A, COR = R \$

INTERCORRELATE, COMPLETE DATA, SYMMETRIC.

INTERCORRELATIONS (PEARSON PRODUCT MOMENT), EACH INPUT VARIABLE IS CORRELATED WITH EACH OTHER, MISSING DATA CAUSES AN ERROR.

- IN = INPUT DATA FILE.
- CROSS = NAME FOR CROSSPRODUCTS.
- COV = NAME FOR COVARIANCES.
- COR = NAME FOR CORRELATIONS.
- SLOPE = NAME FOR SIMPLE BETA WEIGHT MATRIX. THIS IS IN EFFECT A LEAST SQUARE SLOPE THROUGH THE SCATTER PLOT OF THE TWO VARIABLES INVOLVED, PREDICTING THE COLUMN VARIABLE FROM THE ROW VARIABLE.
- INCEPT = NAME FOR INTERCEPT MATRIX. THIS IS THE POINT AT WHICH THE SLOPE WOULD CROSS THE Y AXIS. SLOPE AND INCEPT ARE USEFUL IN PRE-POST STUDIES FOR GETTING A DEVIATION POST SCOPE (IN OTHER WORDS, OBSERVED SCORE MINUS EXPECTED SCORE BASED UPON SIMPLE REGRESSION).
- NOROW1 , IF USED, THIS DIRECTS THE PROGRAM NOT TO USE THE FOLLOWING MEAN-GUESSING STRATEGY.

THE CALCULATION OF CROSSPRODUCTS IN SINGLE PRECISION CAN POSE PROBLEMS IF ONE OR MORE VARIABLES HAS A LARGE MEAN AND A SMALL VARIANCE. THE STRATEGY USED HERE IS TO TAKE THE FIRST NON-MISSING OBSERVED SCORE ON EACH VARIABLE ( FOR INTCDS, THE FIRST ROW OF DATA ), AND USE IT AS A GUESS OF THE MEAN OF THAT VARIABLE. ALL COMPUTATION THEN IS DONE USING DEVIATIONS FROM THE GUESSED MEAN. VALUES AFFECTED INTERNALLY ( MEANS, CROSSPRODUCTS, INTERCEPTS ) ARE RESTORED TO THEIR ORIGINAL FORM WHEN PRINTED OR WRITTEN AS AN OUTPUT FILE.

VERSION 52 - THE INPUT FILE CAN HAVE UP TO 150 COLUMNS.

\*\*\*\*\*

INTMDA, IN = A, COR = ACOR, ROWS = 20 \$

INTERCORRELATION, MISSING DATA ALLOWED, ASYMMETRIC.

IN - INPUT DATA FILE.

- NMAT = NAME FOR THE N MATRIX. THIS IS A MATRIX THAT SHOWS HOW MANY CASES ARE REPRESENTED IN EACH CORRELATION CO-EFFICIENT. SOME OF THE INFORMATION IN THIS MATRIX IS PRINTED OUT AUTOMATICALLY WHENEVER THIS ROUTINE IS USED. THE FOLLOWING ARE PRINTED - MEAN, VARIANCE, STANDARD DEVIATION, NUMBER OF CASES WITH GOOD DATA ON THAT VARIABLE (MAXIMUM N), AND WORST CELL IN THAT ROW OF THE N MATRIX (MINIMUM N).
- CROSS = NAME FOR CROSSPRODUCT MATRIX ( NOTE, THESE ARE NOT CORRECTED FOR THE VARYING N IN EACH CELL ).
- COV = NAME FOR CO-VARIANCE MATRIX.
- COR = NAME FOR INTERCORRELATION MATRIX.
- SLOPE = NAME FOR SIMPLE BETA MATRIX TO PREDICT COLUMN VARIABLE FROM ROW VARIABLE. THIS IS, IN EFFECT, THE SLOPE OF A LEAST SQUARES LINE DRAWN THROUGH THE SCATTER PLOT OF THE TWO VARIABLES.
- INCEPT = NAME FOR INTERCEPTS ON THE Y AXIS OF THE SLOPES IN THE PREVIOUS MATRIX. THESE LAST TWO ARE USED IN PREPOST TYPE STUDIES FOR GETTING A DEVIATION POST SCORE (THE OBSERVED POST SCORE MINUS THE EXPECTED POST SCORE BASED UPON SIMPLE REGRESSION).
- NOROW1, SEE INTCDS

\*\*\*\*\*

TET, IN = A, CTPT = B \$

VERSION 52 - THE INPUT FILE CAN HAVE UP TO 145 COLUMNS.

THIS DOES A TETRACHORIC CORRELATION. THE INPUT FILE SHOULD HAVE DICHOTOMOUS SCORES. USUALLY THE SCORES ARE ZERO VERSUS NON-ZERO. THEREFORE, THEY DO NOT HAVE TO BE LITERALLY ZERO OR ONE. IN FACT, ZERO CAN BE REDEFINED (USING THE IDENTIFIER ZERO, BELOW) SO THE DICOTOMIZATION CAN BE, FOR ALL THE VARIABLES, THAT SCORE VERSUS NOT THAT SCORE. MISSING DATA IS NOT ALLOWED.

- IN - INPUT DATA FILE, NO MISSING DATA, IN ZERO VERSUS NON-ZERO FORM (USUALLY).
- CROSS - CROSS COUNT MATRIX. THE NUMBER OF ROWS OF 'A' THAT HAD NON-ZERO SCORES ON BOTH THE VARIABLES OF A GIVEN CORRELATION.
- TET - TETRACHORIC CORRELATIONS OF ALL THE INPUT VARIABLES.
- SPLIT - SPLIT TEST VALUE. IF FILE 'A' HAS 100 ROWS AND 97 ARE ZERO ON A CERTAIN VARIABLE, THAT VARIABLE HAS A SPLIT OF .97, AND CORRELATIONS INVOLVING THAT VARIABLE ARE NOT TO BE POOR. IF NO SPLIT TEST VALUE IS GIVEN, IT IS SET TO .95. IF A FILE IS

VERY LARGE ( PERHAPS 20,000 ROWS ), A SPLIT OF .98 WOULD BE ALL RIGHT, SINCE 400 CASES ARE INVOLVED.

ZERO - SUBSTITUTE VALUE FOR ZERO. IF THE DATA WERE ALL 3, 4, AND 5, AND ONE WISHED TO CORRELATE 4 VERSUS 3 AND 5, ZERO = 4 WOULD ACCOMPLISH THIS.

CTET - CLEANED UP TETRACHORIC CORRELATION. VARIABLES WITH BAD SPLITS OR A CORRELATION OF +1. OR OF -1. WITH OTHER VARIABLES ARE DISCARDED.

CTET WILL BE THE SAME AS TET WITH POSSIBLY SOME VARIABLES DROPPED. FIRST ALL VARIABLES THAT SPLIT WORSE THAN THE SPLIT VALUE ARE DROPPED. THEN THE VARIABLE WITH THE MOST OFF-DIAGONAL CORRELATIONS OF +1.0 OR -1.0 IS DROPPED, UNTIL THERE ARE NO OFF-DIAGONAL CORRELATIONS OF +1.0 OR -1.0. THIS CLEANING USUALLY PRODUCES A MATRIX THAT WILL SUPPORT A FACTOR ANALYSIS BETTER THAN THE UNCLEANNED VERSION.

11. FACTOR ANALYSIS

CLEANCOR  
F.COFF  
GROUPCOR  
ITFACT  
NEWFAC  
PROMAX  
ROTATE

\*\*\*\*\*

CLEANCOR, IN = A, OUT = B, DELETE = 0.15 \$

THIS 'CLEANS' A CORRELATION MATRIX, WHERE CLEANING IS THE DISCARDING OF ALL VARIABLES WHOSE LARGEST ABSOLUTE CORRELATION ( WITH ANY OTHER VARIABLE ) IS NOT ABOVE THE 'DELETE=' VALUE. IF A DELETE VALUE IS NOT SUPPLIED, THE ONLY VARIABLES DELETED WOULD BE THOSE WITH ALL ZERO CORRELATIONS WITH THE OTHER VARIABLES. THIS CAN OCCUR IN A MISSING DATA CORRELATION SITUATION WHEN THE DATA ON A VARIABLE IS TOTALLY MISSING.

VERSION 52 - THE INPUT MATRIX CANNOT EXCEED 150 BY 150.

\*\*\*\*\*

F.COFF, SDATA = AS, FACTOR = VF, FC = XS

THIS COMPUTES FACTOR COEFFICIENTS.

SDATA IS A STANDARDIZED SCORE MATRIX. FACTOR IS A FACTOR MATRIX. THIS PROGRAM COMPUTES (SDATA TIMES (FACTOR TIMES (FACTOR TRANSPOSED TIMES FACTOR) INVERSE)).

VERSION 52 - THE FACTOR MATRIX CAN BE UP TO 150 BY 50.

SDATA = NAME OF STANDARDIZED SCORE FILE (CAN BE PRODUCED BY SCAN). THIS CAN BE ANY SIZE. IT CAN HAVE MISSING DATA, SEE THE IDENTIFIER 'BADINPUT' BELOW.

FACTOR = NAME OF FACTOR MATRIX. EACH VARIABLE IN THE FACTOR MATRIX MUST BE PRESENT IN SDATA. HOWEVER, SDATA CAN HAVE EXTRA COLUMNS. IF SO, THEY ARE NOT USED.

FC = NAME FOR FACTOR COEFFICIENT - OUTPUT FILE. THIS WILL HAVE AS MANY ROWS AS SDATA, AND AS MANY COLUMNS AS FACTOR. EACH COEFFICIENT WILL HAVE A MEAN OF 50. (SEE BELOW) AND A VARIANCE OF ABOUT 1.

THE GENERIC TERM FOR THIS SORT OF COMPUTATION SEEMS TO BE FACTOR SCORES. THAT COMPUTATION IS CLASSICALLY SDATA TIMES THE INVERSE OF THE FULL

CORRELATION MATRIX TIMES THE FACTORS. THIS INVERSE IS SOMETIMES AWKWARD TO GET, AND THE PSEUDO-INVERSE PROCEDURE IS FREQUENTLY USED, AS IN THIS ROUTINE. IT SHOULD BE NOTED THAT FACTOR COEFFICIENTS COMPUTED WITH A PSEUDO INVERSE ARE ALGEBRAICALLY IDENTICAL TO FACTOR SCORES COMPUTED USING THE INVERSE OF THE CORRELATION MATRIX WITH THE LAST USED COMMUNALITIES IN THE DIAGONAL.

PMULT = NAME FOR THE POST MULTIPLIER MATRIX THAT SDATA WAS FINALLY MULTIPLIED THROUGH. IF ADDITIONAL DATA IS OBTAINED AND STANDARDIZED ACCORDING TO THE ORIGINAL MEANS AND VARIANCES, THEN THAT MATRIX CAN SIMPLY BE MULTIPLIED BY THIS MULTIPLIER MATRIX AND FACTOR SCORES GOTTEN ON THOSE LATER CASES MUCH AS IF THEY HAD BEEN IN THE ORIGINAL COMPUTATION.

MEAN = A CONSTANT WHICH IS ADDED TO EACH FACTOR SCORE WHEN IT IS COMPUTED. NORMALLY THIS IS NOT USED AND 50 IS USED BY THE PROGRAM SO THAT IT IS LIKELY THAT ALL THE FACTOR COEFFICIENTS WILL BE POSITIVE. HOWEVER, ONE MAY WISH TO USE SOME VALUE OTHER THAN 50 ( FOR EXAMPLE, ZERO).

BADINPUT = 'MEAN', 'MISSING' OR 'USEGOOD'. IF MISSING DATA IS FOUND IN SDATA AND THIS IDENTIFIER WAS NOT USED, THE OPERATION ABORTS WITH AN ERROR MESSAGE. THESE THREE OPTIONS ALLOW ALTERNATE ACTIONS.

BADINPUT = MEAN. AN INPUT ROW WITH MISSING DATA WILL HAVE ALL OF ITS OUTPUT SCORES SET TO 50 ( SEE 'MEAN=' ).

BADINPUT = MISSING. ALL SCORES IN THE OUTPUT ROW ARE SET TO BE MISSING.

BADINPUT = USEGOOD. MISSING INPUT SCORE IS SKIPPED. OUTPUT SCORES ARE COMPUTED USING ALL AVAILABLE NON-MISSING INPUT SCORES.

\*\*\*\*\*

GROUPCOR, INCOR=A, INFAC=AF, OUTCOR=GL, OUTFAC=GF &

THIS REORGANIZES A CORRELATION MATRIX AND A FACTOR MATRIX OR IT (OR PART OF IT) ACCORDING TO THE PATTERN OF FACTOR LOADINGS. THE INPUT IS A CORRELATION MATRIX AND A FACTOR MATRIX. THE FACTOR MATRIX CAN HAVE FEWER ROWS (VARIABLES) THAN THE CORRELATION MATRIX, BUT ALL VARIABLES OF THE FACTOR MATRIX MUST HAVE A MATCHING ROW LABEL IN THE CORRELATION MATRIX.

A FREQUENTLY USED SEQUENCE IS...

ITRACE, ROTATE, GROUPCOR, AND PRINT.

VERSION 52 - INCOR CAN BE 120 BY 120, INFAC CAN BE 120 BY 20.

INCOR = INPUT CORRELATION MATRIX NAME.  
 INFAC = INPUT FACTOR MATRIX NAME.  
 OUTCOR= OUTPUT CORRELATION MATRIX NAME. THE FIRST GROUP OF  
 VARIABLES HAD ABSOLUTE FACTOR LOADINGS HIGHER ON  
 FACTOR ONE THAN ANY OTHER, AND SO FORTH.  
 OUTFAC= OUTPUT FACTOR MATRIX NAME, REORDERED.

\*\*\*\*\*

ITFACT, IN = A, SP = B \$

VERSION 52 - IN CAN BE 120 BY 120.

THIS DOES AN ITERATIVE FACTOR ANALYSIS ON INPUT FILE  
 'A' ( PROBABLY A CORRELATION MATRIX ). THE EXAMPLE SHOWS A  
 SIMPLE RUN, JUST GETTING A SIGNIFICANT FACTOR MATRIX OUT.

IT REFACTORS A NUMBER OF TIMES, FIRST UNTIL THE NUMBER OF  
 SIGNIFICANT ROOTS HAS BEEN DECIDED, AND THEN UNTIL THE DIAGONAL  
 ESTIMATE STABILIZES. THIS OCCURS WHEN THE SUM OF SQUARES OF THE  
 FACTOR LOADINGS FOR A VARIABLE IS WITHIN, FOR EXAMPLE, 0.02 OF THE  
 DIAGONAL VALUE IN THE FACTORED CORRELATION MATRIX. (THAT DIAGONAL  
 VALUE IS THE SUM OF SQUARES OF THE PREVIOUS FACTOR LOADINGS).

THE OPERATION 'CLEANCOR' IS SOMETIMES USED TO PROVIDE THE  
 INPUT TO ITFACT.

THIS IS THE COMPLETE SET OF OPTIONS. IN, ROOTS, SP, AND NFAC  
 ARE THE MOST FREQUENTLY USED.

IN = TYPE OF INPUT, SQUARE, SYMMETRIC.

ROOTS = OUTPUT FILE, NV BY 3 (WHERE NV IS THE SIZE OF IN),  
 THE COLUMNS ARE- ROOTS, STARTING DIAGONAL, AND  
 FINAL DIAGONAL.

VEC = OUTPUT VECTORS, NV BY NV. THE FIRST COLUMN IS THE VECTOR  
 ASSOCIATED WITH THE LARGEST FOOT, ETC.

FAC = OUTPUT FACTORS, NV BY NUMBER OF POSITIVE ROOTS.

SP = OUTPUT SIGNIFICANT FACTORS. THIS IS THE OUTPUT  
 THAT IS USUALLY ROTATED. IT IS NV BY NUMBER OF  
 SIGNIFICANT ROOTS. THE CONTENTS OF ITS COLUMNS ARE THE  
 SAME AS THOSE OF FAC, BUT USUALLY IT HAS FEWER COLUMNS.

DROOT = OUTPUT FILE, NV BY NV, A DIAGONAL MATRIX WITH THE  
 ROOTS IN THE DIAGONAL, IN DECREASING ALGEBRAIC ORDER.  
 ALL OFF-DIAGONAL VALUES ARE ZEROS.

TEST = INPUT VALUE, DETERMINING HOW STABLE THE DIAGONAL  
 NEEDS TO BE BEFORE THE PROGRAM FINISHES. IF NOT SET

ON THE CARD (AND IT USUALLY IS NOT) THE PROGRAM USES 0.02 .  
 THE RUN KEEPS REFACTORIZING UNTIL NO DIAGONAL  
 ELEMENT CHANGES BY MORE THAN THIS VALUE.  
 IF THERE ARE MORE THAN 15 ITERATIONS OR IF  
 A DIAGONAL ELEMENT CLIMBS OVER 1.00, THE ITERATION  
 FINISHES. THIS MAY HAPPEN WHEN THERE IS LINEAR DEPENDENCE  
 AMONG THE VARIABLES. IF THE USER INSERTS A VERY LARGE  
 VALUE FOR TEST, THE PROGRAM WILL NOT ITERATE.

NFAC = INPUT VALUE. THIS TELLS THE PROGRAM HOW MANY  
 COLUMNS SF SHOULD HAVE (I.E., HOW MANY ROOTS  
 ARE SIGNIFICANT). IF NOT SUPPLIED, THE PROGRAM  
 TRIES TO DECIDE.

THE USE OF ' NFAC=0 ' IS THE SAME AS NOT USING NFAC AT  
 ALL. THIS MAY OCCUR IN A MACRO EXPANSTION. SOMETIMES A  
 SPECIFIC NUMBER OF ROOTS IS WANTED AND A NUMBER LIKE 7 IS  
 PASSED, SOMETIMES THE USER WANTS THE PROGRAM TO DECIDE,  
 SO HE PASSES A ZERO.

A VERY SIMPLE DECISION RULE IS BEING USED. AFTER EACH OF  
 THE FIRST FIVE ITERATIONS ( IF IT GOES THAT FAR ) THE  
 LARGEST GAP BETWEEN THE ROOTS IN THE 1.1 TO 0.7 RANGE  
 DEFINES THE CUTOFF BETWEEN SIGNIFICANT AND NON-SIGNIFICANT  
 ROOTS. THE FIRST GAP TESTED IS BETWEEN THE SMALLEST ROOT  
 OVER 1.1 AND THE LARGEST ROOT LESS THAN 1.1 . THE LAST  
 GAP IS BETWEEN THE 2 SMALLEST ROOTS LARGER THAN 0.7 .  
 THEN THE FACTORS OF THOSE SIGNIFICANT ROOTS ARE USED TO  
 RE-ESTIMATE THE DIAGONAL, AND SO ON FOR 5 ITERATIONS.

ONE OF THE FOLLOWING THREE IDENTIFIERS MAY BE USED.  
 IF NONE IS USED, THE INPUT DIAGONAL IS USED AS IS FOR THE  
 FIRST FACTORING.

- ONE, THIS SETS THE DIAGONAL OF THE INPUT TO 1.
- ZERO, THIS SETS THE DIAGONAL OF THE INPUT TO 0.
- LARGE, EACH INPUT DIAGONAL VALUE IS SET TO THE LARGEST  
 ABSOLUTE SCORE IN ITS ROW AND COLUMN.

THE PROGRAM QUIETS... 1- IF THE LARGEST DIAGONAL  
 CHANGE IS LESS THAN 0.02 (OR THE 'TEST=' VALUE,  
 IF SUPPLIED), 2- IF 15 ITERATIONS HAVE BEEN  
 DONE, OR 3- IF ANY DIAGONAL HAS CLIMBED OVER 1.0  
 WHEN IT IS DONE, THE PROGRAM USES THE VECTORS AND  
 ROOTS TO RECREATE THE INPUT MATRIX, AND THE LARGEST  
 ERROR IS REPORTED. A TRI-DIAGONALIZATION ROUTINE  
 IS USED HERE. IT IS A SLIGHTLY MODIFIED VERSION  
 OF SHAPE 3202-01 SUBMITTED BY P. F. FUNDERLIC AND  
 M. ELSON OF UNION CARBIDE AT OAK RIDGE, TENN.

\*\*\*\*\*

NEWFAC, INCOR=A, INFAC=B, OUTFAC= C \$

CORRELATION MATRIX A HAS NRI VARIABLES, SOME OF WHICH WERE FACTORED. FACTOR MATRIX B HAS THESE FACTOR LOADINGS. THIS PROGRAM WILL TAKE THE EXTRA VARIABLES IN A (BUT NOT IN B) AND SOLVE A LEAST SQUARES ESTIMATE FOR FACTOR LOADINGS ON THESE ADDITIONAL VARIABLES. THESE NEW FACTOR LOADINGS ARE MERGED ONTO THE BOTTOM OF B TO CREATE THE OUTPUT FACTOR MATRIX. THE VARIABLES IN B CAN BE ANYWHERE IN A, IN ANY ORDER.

VERSION 52 - INCOR CAN BE 120 BY 120, INFAC CAN BE 120 BY 20.

INCOR = NAME OF INPUT CORRELATION MATRIX  
INFAC = NAME OF INPUT FACTOR MATRIX  
OUTFAC= NAME FOR OUTPUT EXTENDED FACTOR MATRIX.

\*\*\*\*\*

PROMAX, INFAC=A, OUTFAC=B, COP=C \$

THIS DOES AN OBLIQUE FACTOR ROTATION. GIVEN INPUT FACTOR MATRIX A, IT CREATES Y BY RAISING EACH ELEMENT OF A TO A POWER AND THEN RESTORING THE ORIGINAL SIGN. THEN AX = Y IS PERFORMED, SOLVING FOR X, A LEAST SQUARES SOLUTION.  $X = (A^T * A)^{-1} * A^T * Y$ . COLUMNS OF X ARE THEN NORMALIZED, AND THE PROMAX FACTORS (B) ARE A \* NORMALIZED X.

VERSION 52 - THE INPUT FACTORS CAN BE UP TO 150 BY 30.

A SEPARATE POWER IS COMPUTED FOR EACH COLUMN. THIS IS NUMBER OF POWS \* (SUM OF 4TH POWERS OF THE MEAN DEVIATION OF THE ELEMENTS OF THE COLUMN / SUM OF SQUARES, SQUARED).

INFAC = NAME OF INPUT FACTORS.  
OUTFAC= NAME FOR OUTPUT PROMAX FACTORS.  
COR = NAME FOR NC BY NC MATRIX OF CORRELATIONS OF FACTORS, WHERE NC IS THE NUMBER OF COLUMNS OF INFAC.  
TRANS = NAME FOR TRANSFORMATION MATRIX.  
COSINP= NAME FOR COSINES.

\*\*\*\*\*

ROTATE, IN = SF, VF = VFAC \$

THIS PROGRAM DOES ORTHOGONAL ROTATIONS ON AN INPUT MATRIX. IT WILL DO A QUARTIMAX, VERIMAX, OR EQUAMAX ROTATION. THE ROTATION SUBROUTINE IS A DESCENDANT OF THE RIND VERIMAX PROGRAM.

THE POWS OF THE MATRIX ARE FIRST NORMALIZED SO THAT THE SUM

OF SQUARES OF EACH ROW EQUALS ONE. THE VARIANCE OF THE SQUARED ELEMENTS IN EACH COLUMN IS COMPUTED AND THE SUM OF THOSE VARIANCES, ONE FOR EACH COLUMN, IS SAVED. ONE ROTATION CYCLE IS THEN PERFORMED. A NEW SUM OF VARIANCES IS COMPUTED AND IS COMPARED AGAINST THE PREVIOUS SUM OF VARIANCES. THE SUBROUTINE RETURNS WHEN THE DIFFERENCE OF THOSE SUMS OF VARIANCES IS LESS THAN 0.0001 ON THREE DIFFERENT (NOT NECESSARILY SUCCESSIVE) ITERATIONS. THE SUBROUTINE WILL ALSO RETURN IF 15 ITERATIONS HAVE BEEN PERFORMED, NO MATTER WHAT THE CONVERGENCE SITUATION IS. A RECORD OF THE VARIANCES DURING THE ROTATION IS PRINTED UPON RETURN TO THE MAIN PROGRAM. WHEN THE ROTATION IS FINISHED, THE COLUMNS ARE ORDERED ACCORDING TO DECREASING SUM OF SQUARES OF ELEMENTS IN THEM.

QF, VF, EF, ARE IDENTIFIERS FOR QUARTTMAX, VERIMAX, AND EQUIMAX FACTOR OUTPUT.

QTR, VTR, AND ETR PROVIDE THE TRANSFORMATION MATRICES.

VERSION 52 - THE INPUT FILE CAN BE UP TO 200 BY 50.

## 12. CROSS-TABULATION ( CROSSTAB )

\*\*\*\*\*

CROSSTAB, IN = A \$  
 X YCC C  
 \*END

CROSSTAB, IN = A, VAR = AGE/EDUC \$

IN ITS SIMPLEST FORM, THIS PROGRAM PRODUCES TABLES WITH N'S (NUMBER OF CASES) IN EACH CELL. THE SUMMARY STATISTICS ARE TOTAL N, MEANS, STANDARD DEVIATIONS AND A CHI SQUARE. THREE VARIETIES OF PERCENT TABLES CAN ALSO BE OBTAINED, THE PERCENT THAT EACH CELL IS IN RELATION TO 1) ITS COLUMN, 2) ITS ROW, 3) THE TOTAL N. TABLES OF MEAN SCORES ON UP TO TEN OTHER VARIABLES CAN BE OBTAINED. ANY OR ALL OF THESE TYPES OF TABLES CAN BE PRODUCED IN A SINGLE RUN.

VARIABLES TO BE CROSSTABBED SHOULD BE CODED SO THAT THE ROW VARIABLE RANGE IS BETWEEN 0 AND 19, AND THE COLUMN VARIABLE RANGE IS BETWEEN 0 AND 9. SCORES OUTSIDE THESE RANGES WILL BE IGNORED. NON-INTEGERS ARE TRUNCATED, FOR EXAMPLE, 7.9 BECOMES 7. THE RCODE AND NCOT FEATURES IN THE FOR/IF/SET STATEMENTS ARE LIKELY TO BE USEFUL HERE.

ANY NUMBER OF TABLES CAN BE ASKED FOR.

VERSION 52 - ONE PASS THROUGH THE INPUT FILE WILL BE DONE FOR EACH 80 TABLES. IF MEANS ON THIRD VARIABLES ARE REQUESTED, ONE PASS WILL BE MADE FOR EVERY 40 'N' TABLES. THE PROGRAM WILL MAKE AS MANY PASSES THROUGH THE INPUT FILE AS ARE NECESSARY TO PRODUCE ALL THE REQUESTED TABLES.

KEYWORDS TELL THE PROGRAM ABOUT THE INPUT FILE AND WHAT TYPE OF CROSSTAB IS DESIRED.

IN = NAME OF INPUT FILE, UP TO 450 COLUMNS.

MASK CARDS (I.F., X YCC C)

MASK CARDS ARE READ IN 50A1 FORMAT. THERE MUST BE ENOUGH MASK CARDS TO TAKE CARE OF ALL THE CURRENT VARIABLES IN THE INPUT FILE. A FILE WITH 70 VARIABLES WILL NEED 2 CARDS - THE FIRST TO MASK VARIABLES 1-50, THE SECOND TO MASK VARIABLES 51-70. LEGAL CHARACTERS ARE BLANK, X, Y, OR C. A MASK CARD WITH X IN COLUMN 2, Y IN COLUMNS 4 AND 5, AND C IN COLUMNS 6, 7, 9 WOULD INSTRUCT THE PROGRAM TO DO CROSSTABS OF THE 2ND VARIABLE (X) IN THE FILE, BY THE 4TH AND 5TH VARIABLES (THE 2 Y'S), AND ALSO ALL POSSIBLE COMBINATIONS OF VARIABLES 6, 7, AND 9 (THE C'S). NUMERIC ROW AND COLUMN LABELS ARE GENERATED. IN OTHER WORDS, EACH X VARIABLE IS CROSSTABBED AGAINST EACH Y VARIABLE. EACH C VARIABLE IS CROSSTABBED AGAINST EACH OTHER C VARIABLE (ONLY ONCE FOR EACH PAIR). \*END MUST FOLLOW THE LAST SUCH CARD.

VAR = IF THIS OPTION IS USED THERE MUST BE AT LEAST TWO ARGUMENTS, I.E., VAR = AGE/EDUC. THE FIRST ARGUMENT (AGE) WILL BE THE ROW VARIABLE. THE SECOND ARGUMENT (EDUC) WILL BE THE COLUMN VARIABLE.

'VAR =' MAY HAVE MORE THAN TWO ARGUMENTS. I.E., VAR=AGE/EDUC/REGION/3/6, WILL PRODUCE A CROSSTAB OF AGE BY EDUC FOR VALUES ON THE VARIABLE REGION OF 3, THEN 4, 5, AND 6. VAR = AGE/SEX/INCOME/70/75/EDUC/0/2, WILL PRODUCE A CROSSTAB OF AGE BY SEX FOR EACH COMBINATION OF INCOME (70,71,72,73,74,75) BY EDUC (0,1,2). THIS CAN RAPIDLY PRODUCE MANY TABLES. THIS LAST EXAMPLE WOULD PRODUCE 19 TABLES ( 6 \* 3 PLUS A TABLE FOR AGE BY EDUCATION.

CONTROLS, IF THIS IS USED THE 'VAR=' OPTION MUST ALSO BE USED. IT TELLS THE PROGRAM TO READ CARDS FURTHER DEFINING THE TABLES. THERE ARE TWO TYPES OF CONTROLS. ONE TYPE OF CONTROLS DEFINES A SUBSET FOR EACH TABLE WITH THE ROW AND COLUMN VARIABLES DEFINED BY THE 'VAR=' OPTION. THE SECOND TYPE IS THE MASK CARDS EXPLAINED ABOVE. IF MASK CARDS ARE USED BY THEMSELVES 'CONTROLS' IS NOT NECESSARY. IF MASK CARDS ARE USED IN ADDITION TO THE 'VAR=' OPTION, CONTROLS MUST BE USED OR THE MASK CARDS WILL NOT BE READ BY THE PROGRAM.

#### SUBSET DEFINITION CARDS

ROW AND COLUMN VARIABLES ARE DESIGNATED BY USING THE 'VAR =' OPTION. THE ROWS TO BE SELECTED FOR EACH TABLE ARE DEFINED BY THEIR SCORES ON UP TO 6 DIFFERENT VARIABLES. THE FIRST CARD FOR EACH TABLE CONTAINS IN COLUMNS 1-5 THE WORD 'CARDS' AND IN COLUMN 10 THE NUMBER OF CARDS THE PROGRAM MUST READ TO DEFINE THE TABLE ( 1 - 6 ). COLUMNS 11-50 MAY BE USED FOR A HEADING FOR THE TABLE. THIS CARD IS FOLLOWED BY FROM 1 TO 6 CARDS CONTAINING THE VARIABLE NAME IN COLUMNS 1-8, LOW SCORE IN COLUMNS 11-20 AND HIGH SCORE IN COLUMNS 21-30. THESE SCORES ARE READ IN F10.0 FORMAT. A ROW IS USED IN THE TABLE IF ITS SCORES ON ALL THE CONDITIONAL VARIABLES ARE NOT MISSING AND NOT BEYOND THE LOW AND HIGH RANGE.

\*END MUST FOLLOW THE LAST CONTROL CARD (IF ANY ARE USED AT ALL). MASK CARDS AND SUBSET TYPE CARDS MAY BE USED IN THE SAME RUN IN ANY ORDER.

#### OPTIONAL KEYWORDS

MEANS = LIST, THIS PROVIDES THE NAMES OF VARIABLES FOR WHICH MEANS ARE TO BE PRODUCED. I.E., MEANS = INCOME, THE MEAN SCORE ON THE VARIABLE LABELLED INCOME, WILL BE GIVEN FOR EVERY CELL IN EVERY CROSSTAB. MEANS = INCOME/AGE\*ARRY, WILL PRODUCE MEAN SCORES FOR THE VARIABLES LABELLED INCOME AND AGE\*ARRY.

SUMS = LIST, LIKE 'MEANS' EXCEPT THE SUM OF THE SCORES IS CALCULATED.

- COLPCT,        PRODUCES THE PERCENT THAT EACH CELL IS OF ITS COLUMN.
- CUMCOL,        THE COLUMN PERCENTS WILL BE CUMULATIVE.
- ROWPCT,        PRODUCES THE PERCENT THAT EACH CELL IS OF ITS ROW.
- CUMROW,        THE ROW PERCENTS WILL BE CUMULATIVE.
- TOTPCT,        PRODUCES THE PERCENT THAT EACH CELL IS OF THE TOTAL N.

LABELS = N/N/N/N, IF THIS OPTION IS USED THE PROGRAM WILL READ ONE SET OF ROW AND COLUMN LABELS. THE ROW LABELS COME FIRST IN CARD COLUMNS 1-8, 11-18, 21-28,....., 71-78. IF THE FIRST ROW VARIABLE IS CODED 1-14, 2 CARDS WILL BE READ FOR LABELS. COLUMN LABELS FOLLOW IN THE SAME FORMAT AND START WITH A NEW CARD. THESE CARDS MUST FOLLOW THE CROSSTAB CARD AND PRECEDE ANY CONTROL CARDS. THIS OPTION WILL PROBABLY BE USED ONLY WHEN THE 'VAR =' OPTION IS USED. LABELS = 1/7/2/5, WILL INSTRUCT THE PROGRAM TO READ LABELS FOR SCORES ON THE ROW VARIABLE BETWEEN 1 AND 7 AND FOR SCORES ON THE COLUMN VARIABLE BETWEEN 2 AND 5. NOTE... THESE LABELS DO NOT NEED TO CONFORM TO THE P-STAT RULES FOR NAMES AND LABELS.

COMBOS,        USEFUL ONLY WHEN 3 OR 4 VARIABLES ARE ARGUMENTS IN THE 'VAR=' OPTION. TABLES FOR THE RANGE OF THE THIRD VARIABLE FOR EACH LEVEL OF THE FOURTH VARIABLE AND FOR THE RANGE OF THE FOURTH VARIABLE FOR EACH LEVEL OF THE THIRD VARIABLE WILL BE GENERATED.

WEIGHT = V,    THE TABLES WILL BE WEIGHTED USING THE VALUE OF VARIABLE V.

PZ,            USUALLY A CELL IN THE PRINTOUT WILL BE BLANK IF THE N IS ZERO. IF PZ IS USED, THE ZEROS WILL BE PRINTED OUT.

EXAMPLES.....

-----

CROSSTAB, IN=RD,VAR= AGE/FDUC \$

THIS WILL PRODUCE 1 CROSSTAB OF THE VARIABLE LABELLED AGE BY THE VARIABLE Labeled FDUC.

-----

CROSSTAB, IN = A \$  
 XXXYY CCCXYCCXCY  
 \*END

THIS WILL PRODUCE 35 TABLES. 1 FOR EACH X AGAINST EACH Y AND 1 FOR ALL THE COMBINATIONS OF C'S.

```

-----
CROSSTAB, IN = A, VAR = AGE/FDUC, CONTROLS $
CARDS 2 CATHOLICS FARNING UNDER 5,000
RELIGW 1. 2.
INCOME 1. 5.
*END

```

THIS WILL PRODUCE 2 TABLES, THE ROW VARIABLE 'AGE' BY THE COLUMN VARIABLE 'FDUC' FOR THE TOTAL SAMPLE AND THEN FOR THOSE ROWS WITH SCORES BETWEEN 1 AND 2 ON THE VARIABLE LABELLED 'RELIGW' AND BETWEEN 1 AND 5 ON THE VARIABLE LABELLED 'INCOME'.

```

-----
CROSSTAB, IN=X, VAR=AGE/FDUC/REGION/1/4, LABELS = 1/5/1/4 $
UNDEF 18 18-25 26-35 36-45 46-55
COLLEGE H.S. 4 H.S. 1-3 ELEMSCH

```

5 TABLES OF NS WILL BE PRODUCED. 1 FOR THOSE WITH GOOD SCORES ON AGE AND FDUC AND 1 TABLE FOR EACH LEVEL ( 1,2,3,4 ) OF REGION. 5 LABELS WILL BE READ FOR THE ROW VARIABLE ( AGE ) AND 4 LABELS FOR THE COLUMN VARIABLE ( EDUC ).

```

-----
CROSSTAB, IN = PAWDATA, VAR = AGE/SFX, CONTROLS, MEANS = INCOME$
CARDS 1 COLLEGE EDUCATED WOMEN
EDUC 0. 1.
CARDS 3 SOUTHERN FARMERS UNDER 45
REGION 3. 3.
OCCUPTON 5. 5.
AGE 1. 4.
CCCC
*END

```

THIS WILL PRODUCE 9 N TABLES, 1) THE RAW VARIABLE 'AGE' BY THE COLUMN VARIABLE 'SFX', 2) 'AGE' BY 'SFX' FOR COLLEGE EDUCATED WOMEN, 3) 'AGE' BY 'SFX' FOR SOUTHERN FARMERS UNDER 45, AND 4), 6 N TABLES FOR THE COMBINATIONS OF VARIABLES 1-4. THE MEAN SCORES FOR THE VARIABLE INCOME WILL ALSO BE INCLUDED IN EACH OF THESE N TABLES.

```

-----
CROSSTAB, IN = YZ, VAR = FDUC/RELIGW/AGE/0/4/OCCUP/1/4, COMBOS,
MEANS = INCOME/NCHILD, ROWPCT, TOTPCT $

```

THIS WILL PRODUCE MANY PAGES OF PAPER. FDUC AND RELIGW WILL BE THE ROW AND COLUMN VARIABLES. SINCE LABELS WERE NOT REQUESTED, NUMERIC LABELS WILL BE GENERATED. AN N TABLE WILL BE PRODUCED FOR THE OVERALL SAMPLE, AND FOR EACH COMBINATION OF THE SCORES ON AGE BETWEEN 0 AND 4 AND OCCUP BETWEEN 1 AND 4 (FOR ROWS WITH NON-MISSING SCORES ON THE VARIABLE INCOME). IN ADDITION THERE WILL BE 4 TABLES FOR AGE BETWEEN 0 AND 4 FOR EACH LEVEL OF THE VARIABLE OCCUP AND 5 TABLES FOR OCCUP BETWEEN

1 AND 4 FOR EACH LEVEL OF THE VARIABLE AGE. TABLES OF MEAN VALUES ON THE VARIABLE INCOME WILL BE PRODUCED. THE SAME PROCEDURE WILL BE FOLLOWED FOR THOSE ROWS WITH NON-MISSING SCORES ON THE VARIABLE NCHILD AND APPROPRIATE N AND MEANS TABLES PRODUCED. THE VALUES FOR ROW AND TOTAL PERCENT WILL BE INCLUDED IN THE N TABLES.



13. DISCRIMINANT ANALYSIS ( DISCRIM )

\*\*\*\*\*

DISCRIM, IN = A, NG = 4, NVA = 15, OUT = B \$

THIS DOES A MULTIPLE GROUP DISCRIMINANT ANALYSIS.

EACH OF THE ROWS OF A DATA FILE CAN BE THOUGHT OF AS BELONGING TO ONE OF SEVERAL POSSIBLE GROUPS. IF, FOR EXAMPLE, A VARIABLE NAMED 'GROUP' WERE PART OF THE FILE, AND EACH ROW HAD A SCORE OF 1 OR 2 OR 3 ON IT, THE FILE COULD BE SEPARATED INTO THREE GROUPS. THIS PROGRAM TAKES SUCH A FILE, USES SCORES ON THE NON-ANALYSIS VARIABLES TO SEE WHICH GROUP ( IF ANY ) THAT A ROW BELONGS TO, AND USES THE ANALYSIS VARIABLES TO CONSTRUCT SEVERAL SETS OF WEIGHTS, ONE SET FOR EACH GROUP. THEN, MAKING A SECOND PASS THROUGH THE FILE, THE DATA IN EACH ROW IS USED, WITH THE WEIGHTS, TO PRODUCE A SCORE FOR THAT ROW FOR EACH GROUP. IN OTHER WORDS, IF THERE ARE 7 GROUPS, 7 SCORES ARE COMPUTED. THE ROW IS THEN RE-ASSIGNED TO THE GROUP WHOSE WEIGHTS YIELDED THE HIGHEST SCORE.

THE STATISTICAL PROCEDURE IS SIMILAR TO THAT USED BY THE IBM SCIENTIFIC SUBROUTINE PACKAGE AND BY THE BMD DISCRIMINANT PROGRAMS. THE SUBROUTINE WHICH ESTIMATES THE PROBABILITY FOR AN 'F' VALUE COMES FROM D. VELDMAN'S 'FORTRAN PROGRAMMING FOR THE BEHAVIORAL SCIENCES'.

VERSION 52 PERMITS AS MANY AS 20 GROUPS AND 90 ANALYSIS VARIABLES.

\*\*\*\*\*  
\* OUTPUT PRINTED BY THE PROGRAM \*  
\*  
\*\*\*\*\*

1. THE NUMBER OF CASES AND THE DEFINITION RULES FOR EACH GROUP.
2. AN ERROR CHECK INVOLVING THE INVERSE COMPUTATION.
3. MEANS ON EACH VARIABLE FOR ALL GROUPS TOGETHER AND FOR EACH GROUP SEPARATELY.
4. UNIVARIATE 'F' VALUES FOR EACH VARIABLE. THESE ARE NOT AFFECTED BY THE INCLUSION OR DELETION OF OTHER ANALYSIS VARIABLES. THEY ARE IDENTICAL TO THE 'F' VALUES PRODUCED BY THE 'REQ' PROGRAM, ASSUMING THE SAME GROUPS WERE COMPARED.
5. MULTIVARIATE 'F' VALUES FOR EACH VARIABLE. THESE ARE CALLED 'F TO REMOVE' IN THE STEPWISE DISCRIMINANT PROGRAM IN THE BMD SERIES. QUOTING THE BMD MANUAL, '...THESE ARE THE LIKELIHOOD RATIO TESTS OF THE EQUALITY, OVER ALL G GROUPS, OF THE CONDITIONAL DISTRIBUTION OF VARIABLE J, GIVEN THE OTHER VARIABLES'. IF SUCH AN 'F' IS VERY LOW, FOR EXAMPLE 0.1, THAT VARIABLE COULD BE DELETED WITH VERY LITTLE LOSS IN DISCRIMINATING POWER.
6. MANALANOBIS D SQUARE, WHICH TESTS THE HYPOTHESIS THAT THE GROUP

MEANS ON THE VARIABLES DO NOT DIFFER.

- 7. A CLASSIFICATION TABLE, DISPLAYING THE DEGREE OF SUCCESS OF THE RE-CLASSIFYING PROCEDURE. IF A CASE FROM ORIGINAL GROUP 2 IS RE-CLASSIFIED INTO GROUP 4 ( ON THE BASIS OF HIS DISCRIMINANT SCORES ), HE IS COUNTED IN THE 2ND ROW AND 4TH COLUMN OF THE TABLE. IF ALL CASES ARE CORRECTLY RE-CLASSIFIED, THE TABLE WILL HAVE NON-ZERO COUNTS ONLY IN THE DIAGONAL.

```

*****
*                                     *
*               IDENTIFIERS          *
*                                     *
*****

```

'NG' IS THE NUMBER OF GROUPS. NG CARDS MUST FOLLOW, DESCRIBED BELOW, EACH DEFINING A GROUP.

'NVA' IS THE NUMBER OF VARIABLES TO BE ANALYSED. THESE VARIABLES MUST BE THE LEFTMOST VARIABLES IN THE INPUT FILE ( I.E., 1 THROUGH NVA ). THE ( C ) TYPE QUALIFICATION PHRASE CAN BE USED TO REARRANGE THE VARIABLES INTO THE PROPER ORDER.

'IN' IS THE INPUT FILE OF DATA. ITS SIZE IS NR ROWS BY NVA+K COLUMNS. NR IS THE NUMBER OF ROWS, AND CAN BE ANY SIZE. K IS THE NUMBER OF ADDITIONAL VARIABLES NEEDED TO DEFINE THE GROUPS. NR SHOULD BE SUBSTANTIALLY MORE THAN NVA + NG. SCORES ON THE NVA ANALYSIS VARIABLES SHOULD BE COMPLETE. ROWS WITH MISSING DATA IN THE NVA VARIABLES ARE SKIPPED. NR SHOULD BE THOUGHT OF AS THE NUMBER OF ROWS THAT ARE INCLUDED IN THE ANALYSIS, IN OTHER WORDS, THE NUMBER OF ROWS THAT CAN BE INITIALLY ASSIGNED TO SOME GROUP.

'FUN' IS AN OUTPUT FILE, NVA + 1 BY NG. THE FIRST COLUMN CONTAINS THE FIRST DISCRIMINANT FUNCTION, WITH THE CONSTANT IN THE LAST ROW. THIS IS USED TO PRODUCE SCORES FOR MEMBERSHIP IN THE FIRST GROUP. THE SECOND COLUMN IS INVOLVED WITH THE SECOND GROUP, AND SO FORTH. IF NVA WERE 15 AND 'FUN=FMAT' WERE USED, THE FOLLOWING P-STAT STATEMENT WOULD PRODUCE A NR BY NG FILE OF DISCRIMINANT SCORES.....

```

MULTIPLY,
PRE = A ( C 1-15 ) ( SETX UNIT TO 1 )
      ( IF ANY( 1-15 ) MISSING, DELETE ),
POST = FMAT, OUT = SCORE $

```

'OUT' IS AN OUTPUT FILE OF PROBABILITIES AND DISCRIMINANT SCORES. ITS SIZE IS NR BY NG + 4. THE COLUMNS ARE.....

- 1- THE GROUP THAT THE ROW WAS ORIGINALLY IN, AS DEFINED BY THE GROUP CARDS. THUS, A VALUE OF 1 THROUGH NG.
- 2- THE GROUP THAT THE ROW MOST PROBABLY BELONGS TO, BASED ON ITS COMPUTED DISCRIMINANT SCORES ON EACH GROUP'S FUNCTION. ALSO 1 THROUGH NG.
- 3- THE PROBABILITY OF BELONGING TO ITS ORIGINAL GROUP, 0 THROUGH 1.

- 4- THE PROBABILITY OF BELONGING TO ITS HIGHEST SCORE GROUP. IF A ROW IS ASSIGNED BY THE PROGRAM TO ITS OWN GROUP, ITS SCORES ON THE FIRST TWO VARIABLES WILL BE IDENTICAL, AND ITS SCORES ( PROBABILITIES ) ON VARIABLES 3 AND 4 WILL BE IDENTICAL.
- 5- DISCRIMINANT SCORE FOR MEMBERSHIP IN THE FIRST GROUP.
- 6- DISCRIMINANT SCORE FOR MEMBERSHIP IN THE SECOND GROUP, ETC.

```

*****
*                                     *
*          GROUP SELECTION CARDS     *
*                                     *
*****

```

A SELECTION CARD MUST BE SUPPLIED FOR EACH GROUP.

- COL 1-8 J LABEL FOR THE GROUP. THIS MUST BE A VALID P-STAT LABEL, FIRST CHARACTER A LETTER, ETC.
- COL 13-20 THE NAME OF A SELECTION VARIABLE IN THE INPUT FILE. SELECTION VARIABLES CANNOT BE ANY OF THE FIRST NVA VARIABLES.
- COL 21-30 LOW SCORE.
- COL 31-40 HIGH SCORE.
- COL 43-50 ANOTHER SELECTION VARIABLE ( OPTIONAL ).
- COL 51-60 ITS LOW SCORE.
- COL 51-70 ITS HIGH SCORE.

A ROW IS ORIGINALLY ASSIGNED INTO GROUP J IF...

IT HAS NO MISSING DATA IN THE ANALYSIS VARIABLES, AND ITS SCORE ON THE FIRST SELECTION VARIABLE ( GROUP CARD COLUMN 13-20 ) FOR GROUP J IS AT OP WITHIN THE RANGE DEFINED BY THE LOW AND HIGH SCORES, AND IT ALSO PASSES THE SECOND SELECTION VARIABLE TEST, IF ONE IS THERE.

A ROW IS ASSIGNED TO THE FIRST GROUP FOR WHICH IT QUALIFIES. IF IT FAILS THE TESTS FOR ALL THE GROUPS, IT IS SKIPPED.

14. FREQUENCY DISTRIBUTIONS ( FREQ )

FREQ  
FREQ.C

\*\*\*\*\*

FREQ, IN = A, DES = D \$

FREQUENCY DISTRIBUTIONS, ONE PAGE (AT LEAST) PER INPUT VARIABLE, WILL BE PRODUCED.

THIS PROGRAM TAKES A DATA FILE AND COMPUTES FREQUENCY DISTRIBUTIONS ON ANY NUMBER OF VARIABLES. THE MEAN AND RANGE OF EACH VARIABLE IS NEEDED. THAT RANGE IS SLICED INTO A NUMBER OF CATEGORIES OF EQUAL WIDTH, FROM THE LOWEST SCORE UP TO THE MEAN, AND ANOTHER SET OF CATEGORIES FROM THE MEAN TO THE HIGHEST SCORE. THIS HELPS ON SKEWED DISTRIBUTIONS.

A NUMBER OF GROUPS MAY BE DEFINED IN TERMS OF THEIR SCORES ON CERTAIN VARIABLES IN THE DATA FILE. A SIMPLE ANALYSIS OF VARIANCE IS PERFORMED ON THE SUB-GROUP DATA IF THERE IS MORE THAN ONE GROUP. THE FREQUENCY DISTRIBUTION ALSO INCLUDES THE DISTRIBUTION OF ALL MEMBERS OF THE DATA FILE, WHETHER THEY WERE DEFINED IN A GROUP OR NOT. THE OUTPUT ALSO HAS PERCENTS AND CUMULATIVE PERCENTS.

VERSION 52 - THERE CAN BE UP TO 10 GROUPS AND 30 CATEGORIES. THE PROGRAM WILL PROCESS ALL AVAILABLE VARIABLES, MAKING ONE PASS THROUGH THE FILE FOR EACH 47 VARIABLES.

NG = NUMBER OF GROUPS, IF ANY. A DATA CARD DEFINING EACH GROUP FOLLOWS ( IF THERE ARE GROUPS ).

IN = NAME OF INPUT DATA FILE.

DES = NAME OF MATRIX OF MEANS, VARIANCES, ETC. THIS MATRIX CONTAINS AS ITS FOURTH AND FIFTH COLUMNS THE LOWEST AND HIGHEST SCORES ON THAT VARIABLE. THIS MATRIX IS OF ORDER NV BY 6, AND IS USUALLY THE OUTPUT OF SCAN OR DATA.

ALLCAT, IF THIS IDENTIFIER USED, THE PROGRAM WILL PRINT ALL CATEGORIES, EVEN IF A CATEGORY DID NOT HAVE ANY SCORES IN IT. IF ALLCAT IS NOT USED, THE PROGRAM PRINTS ONLY THOSE CATEGORIES THAT HAD SCORES. THE REASON FOR THIS OPTION IS TO MAKE IT EASIER FOR THE USER TO VISUALIZE DIFFERENCES IN DISTRIBUTIONS BETWEEN THE VARIOUS GROUPS. IF HE IS TRYING TO DO THIS BY LOOKING AT HIS PRINTED OUTPUT, THE EFFECT IS DISTORTED IF CATEGORIES WITHOUT SCORES ARE NOT PRINTED AND THE OUTPUT AS A RESULT IS COMPRESSED TOGETHER, REPORTING ONLY THOSE CATEGORIES THAT DID HAVE SCORES.

- ALLGRP, IF USED, A CASE WILL BE INCLUDED IN ALL THE GROUPS THAT IT QUALIFIES. USUALLY A CASE FALLS IN ONLY THE FIRST GROUP FOR WHICH IT QUALIFIES. ( USING THIS OPTION INVALIDATES THE ANALYSIS OF VARIANCE. )
- NCAT = THIS (RARELY USED) FEATURE ALLOWS THE USER TO SPECIFY THE NUMBER OF CATEGORIES TO BE USED. IT MUST BE AN EVEN NUMBER, AND CANNOT EXCEED THE MAXIMUM NUMBER OF CATEGORIES. ONE MIGHT WISH, FOR EXAMPLE, TO SAY ' NCAT = 20 ' IN ORDER TO FORMAT THE PAGES DIFFERENTLY.
- EQUALCAT, IF USED, THE CATEGORIES WILL BE DIVIDED EQUALLY OVER THE RANGE OF EACH VARIABLE. IF ' EQUALCAT ' IS NOT USED, HALF THE CATEGORIES WILL BE BELOW THE MEAN AND HALF WILL BE ABOVE THE MEAN.

GROUPS - A GROUP IS DEFINED AS THOSE CASES WHO, ON VARIABLE I, HAVE A NON-MISSING SCORE AT OR BETWEEN A AND B, AND ALSO, ON VARIABLE J, HAVE A NON-MISSING SCORE AT OR BETWEEN C AND D. THESE VARIABLES I AND J CAN BE ANY VARIABLES IN THE DATA FILE. IF ONLY VARIABLE I IS NEEDED, INFORMATION ON J NEED NOT BE PUNCHED.

#### GROUP CARD FORMAT

- COL 1- 8 NAME TO BE USED FOR THAT GROUP.
- COL 13-20 LABEL OF GROUP-DEFINING VARIABLE I. CAN BE PUNCHED ANYWHERE IN THOSE COLUMNS.
- COL 21-30 LOW BOUNDARY SCORE ON THAT VARIABLE.
- COL 31-40 HIGH BOUNDARY SCORE ON THAT VARIABLE.
- COL 43-50 VARIABLE LABEL FOR VARIABLE J, IF USED.
- COL 51-60 LOW BOUNDARY SCORE ON J VARIABLE.
- COL 61-70 HIGH BOUNDARY SCORE ON J VARIABLE.

THESE SCORES SHOULD BE PUNCHED WITH A DECIMAL PLACE. IF SO, THE PROGRAM WILL FIND THE DECIMAL PLACE WITHIN THE TEN COLUMNS INVOLVED AND BELIEVE IT, WHEREVER IT IS. IF NO POINT IS PUNCHED, THE INTERNAL FORMAT USED IS F10.4, SO THAT THE LAST FOUR COLUMNS IN EACH OF THOSE TEN COLUMN FIELDS WOULD BE INTERPRETED AS DECIMAL PLACES. IN GENERAL, IT IS FAR BETTER TO PUNCH THE POINT.

A SIMPLE ANALYSIS OF VARIANCE IS PERFORMED AFTER THE FREQUENCY DISTRIBUTION IS PRINTED OUT. THIS IS DONE IF THERE ARE TWO OR MORE GROUPS THAT HAD DATA. MISSING DATA IS NOT INCLUDED IN THESE COMPUTATIONS AND IS REPORTED SEPARATELY IN THE FREQUENCY DISTRIBUTIONS.

CONSIDER...

```

FREQ, IN = A (C 11-40), DES = D, NG = 3$
NO.SMOKE      SMOKE      0.          0.
MOD.SMOK      SMOKE      1.          19.
PACKADAY      SMOKE      20.         9999.

```

NOTE.... IF VARIABLE SMOKE IS NOT ONE OF VARIABLES 11-40, IF MUST BE INCLUDED, AS..IN=A(C SMOKE, 11-40).

THIS WILL PRODUCE 30 PAGES OF OUTPUT, SINCE 30 VARIABLES IN A ARE MADE AVAILABLE. D IS PROBABLY A DESCRIPTION FILE OF ALL OF A, BUT, USING LABEL MATCHING, THE APPROPRIATE 30 PAGES AND MEANS WILL BE FOUND. THREE GROUPS ARE INVOLVED, SINCE THE USER WISHES TO COMPARE NO SMOKING, MODERATE SMOKING, AND HEAVY SMOKING. CASES WITH 0. ON VARIABLE SMOKE (WHICH MUST BE ONE OF THE 11-40 IN THIS EXAMPLE) ARE INCLUDED IN GROUP 'NO.SMOKE', AND SO FORTH.

IF THIS WERE DONE FOR MALES ONLY, (CODED 1.), THE GROUP CARDS WOULD BE...

```

NO.SMOKE      SMOKE      0.          0. SEX 1.    1.
MOD.SMOK      SMOKE      1.          19. SEX 1.    1.
PACKADAY      SMOKE      20.         9999. SEX 1.    1.

```

\*\*\*\*\*

FREQ.C, IN=A, DES=B, NG=4 \$

THIS IS LIKE 'FREQ' WITH 3 EXCEPTIONS...

- 1 - MORE CATEGORIES ( HENCE THE C ) ARE ALLOWED.  
VERSION 52 - 5 GROUPS, 100 CATEGORIES, 23 VARIABLES PER PASS.
- 2 - THE IDENTIFIER 'PAIR' IS AVAILABLE. IF USED, IT ASSUME THE ROWS IN THE INPUT FILE COME IN PAIRS. NG WOULD BE 2, SO THAT THE FIRST GROUP HAS ALL FIRST ROWS AND THE SECOND GROUP HAS ALL THE SECOND ROWS. A CORRELATED T RATHER THAN A SIMPLE T OR F IS PRODUCED.
- 3 - THE IDENTIFIER 'ALLGRP' SHOULD NOT BE USED WHEN 'PAIR' IS USED.

15. REGRESSION ( MULTR )

MULTR  
RESIDU.1

\*\*\*\*\*

MULTR, COR=A, DES=B\$  
I I IDI I  
DIIII  
\*END

MULTIPLE REGRESSION. THIS PROGRAM HAS THE ADVANTAGE OF DOING A NUMBER OF MULTIPLE REGRESSIONS QUITE RAPIDLY WITH VERY SIMPLE CONTROLS DETERMINING WHICH VARIABLES ARE TO BE INCLUDED IN EACH GIVEN REGRESSION. ITS DISADVANTAGE IS THAT IT MAKES NO USE OF RAW DATA. AS A RESULT, RESIDUALS ARE NOT COMPUTED IN THIS PROGRAM.

COR= NAME OF THE INPUT CORRELATION MATRIX.

DES= NAME OF NV BY 6 MATRIX OF MEANS, VARIANCES, ETC. THIS MATRIX IS USED FOR - (1) REPORTING THE MEANS AND STANDARD DEVIATIONS OF EACH VARIABLE, (2) COMPUTING THE RATIO OF THE STANDARD DEVIATIONS OF EACH INDEPENDENT VARIABLE WITH THE S.D. OF THE DEPENDENT VARIABLE, SO THAT A RAW SCORE CO-EFFICIENT CAN BE COMPUTED FROM THE STANDARDIZED SCORE CO-EFFICIENT, AND (3) FINDING THE NUMBER OF CASES INVOLVED FOR DETERMINING THE DEGREES OF FREEDOM IN THE F RATIOS.

MISSING DATA CORRELATIONS...

SINCE THE INPUT IS A CORRELATION MATRIX ( AND IS NOT RAW DATA ), THE PROGRAM HAS NO WAY OF KNOWING WHETHER THE DATA INVOLVED IN THE CORRELATION MATRIX WAS COMPLETE OR PERHAPS PARTIALLY MISSING. THE PROGRAM SCANS THE MATRIX OF MEANS, STANDARD DEVIATIONS, RANGES, AND N'S, AND TAKES THE SMALLEST N OF ANY VARIABLE USED IN A PARTICULAR REGRESSION. BECAUSE OF THE UNCERTAINTY ABOUT DEGREES OF FREEDOM, THE USE OF MISSING DATA CORRELATION MATRICES IN THIS PROGRAM IS DISCOURAGED. PROBABLY IF THE N WAS VERY LARGE AND ONLY A FEW CASES WERE MISSING, THE RESULTING REGRESSIONS WILL BE REASONABLY ACCURATE. HOWEVER, IF SOME OF THE VARIABLES INVOLVED HAD VERY SMALL N'S, THEN IT IS VERY LIKELY THAT THE INVERSE WILL NOT BE COMPUTABLE, AND EVEN IF IT IS, NOT TOO MUCH CONFIDENCE SHOULD BE PLACED IN THE RESULTING REGRESSION WEIGHTS.

MAXIMUM SIZES...

VERSION 52 -

THE INPUT CORRELATIONS CAN BE UP TO 50 BY 50.  
UP TO 24 INDEPENDENT VARIABLES CAN BE USED AT ONE TIME.

THE MATRIX OF MEANS AND STANDARD DEVIATIONS CAN BE ANY SIZE, AND THE PROGRAM WILL MATCH THE LABELS OF THE CORRELATION MATRIX WITH THE LABELS OF THE MEANS AND STANDARD DEVIATION MATRIX, FINDING THE APPROPRIATE MEANS, ETC. ALL VARIABLES IN THE CORRELATION MATRIX MUST ALSO BE IN THE 'DES' MATRIX.

#### REGRESSION CONTROL CARDS...

READING THE INITIAL CONTROL CARD ONLY CAUSES THE MATRICES NAMED ON IT TO BE FOUND AND BROUGHT INTO CORE. NOW, A CARD MUST BE READ, WHICH TELLS THE PROGRAM WHICH VARIABLES ARE THE INDEPENDENT VARIABLES AND WHICH IS THE DEPENDENT VARIABLE FOR THIS PARTICULAR REGRESSION. A CARD IS READ UNDER 50A1 FORMAT.

CONSIDER A MATRIX WITH FORTY VARIABLES. IF ONE WISHED TO USE VARIABLES 1, 3, 5, 7, AND 9 AS INDEPENDENT VARIABLES AND 6 AS THE DEPENDENT VARIABLE, THE CARD WOULD BE PUNCHED AS FOLLOWS - IBIPIDIBI (WHERE B MEANS BLANK) AND THE REST OF THE CARD IS BLANK. THE CODING IS - ALL VARIABLES ASSOCIATED WITH I PUNCHED IN THE CARD ARE CONSIDERED TO BE INDEPENDENT VARIABLES, ALL VARIABLES WITH BLANK ARE OMITTED IN THIS PARTICULAR REGRESSION, AND THE VARIABLE WITH D IS THE DEPENDENT VARIABLE.

IF THE SAME INDEPENDENT VARIABLES ARE TO BE RUN AGAINST A NUMBER OF DEPENDENT VARIABLES, ONE CARD WITH ALL THE D'S ON IT IN ADDITION TO THE I'S FOR THE INDEPENDENT VARIABLES WILL CAUSE AS MANY REGRESSION RUNS AS THERE ARE D'S, DOING ALL THE INDEPENDENT AGAINST THE LEFTMOST D THE FIRST TIME, ETC. THE PROGRAM WILL CONTINUE READING CARDS UNTIL A CARD WITH \*END IN COLUMNS 1-4 IS READ.

#### OUTPUT...

THIS PROGRAM WAS WRITTEN AT PRINCETON, BUT WAS BASED UPON USE OF BOTH THE BIND6 PROGRAM AND THE REGRESSION PROGRAM IN THE COOLEY-LOHNE'S BOOK. THE OUTPUT IS VERY SIMILAR TO THE BIND6 OUTPUT. IT INCLUDES THE VARIABLE NUMBER, LABEL, MEAN, STANDARD DEVIATION, INTERCORRELATION WITH THE DEPENDENT VARIABLE, A RAW REGRESSION WEIGHT, A WEIGHT FOR STANDARDIZED DATA, THE STANDARD ERROR OF THE REGRESSION WEIGHT, THE T OF THE REGRESSION WEIGHT, THE PARTIAL CORRELATION (AS WELL AS THE MULTIPLE CORRELATION), AND THE F TEST OF SIGNIFICANCE OF THE REGRESSION.

\*\*\*\*\*

RESIDU.1, IN = A, SLOPE = E, INCEPT = C, OUT = D \$

IN IS A DATA FILE WHICH HAS AS VARIABLES N PPE SCORES FOLLOWED BY N POST SCORES. IT HAS, THEREFORE, N+N COLUMNS. IT CAN HAVE ANY NUMBER OF ROWS, AND DATA CAN BE MISSING. IF THIS WERE RUN THROUGH INTCD5 OR INTMD5, A SLOPE MATRIX AND AN INTERCEPT MATRIX COULD HAVE BEEN PRODUCED. BOTH THESE WOULD BE OF SIZE N+N BY N+N.

THIS PROGRAM PRODUCES AN OUTPUT FILE OF RESIDUALS, THAT PART OF THE POST SCORE NOT PREDICTABLE BY THE PRE SCORE. IF, FOR ANY CASE, EITHER THE PRE OR THE POST SCORE IS MISSING FOR A VARIABLE, THE ASSOCIATED RESIDUAL SCORE IS SET TO BE MISSING.

THE OUTPUT FILE WILL BE NR BY N, WHERE NR IS THE NUMBER OF ROWS IN THE 'IN' FILE.

VERSION 52 - 'IN' CAN HAVE UP TO 150 COLUMNS.

## 16. TWO WAY ANALYSIS OF VARIANCE ( PBYQ )

\*\*\*\*\*

PBYQ, IN=A, ROW=C.AGE/3, COL=EDUCAT/3, FIRST=40 \$

THIS PROGRAM DOES AN ANALYSIS OF VARIANCE OR MEANS IN A TWO-WAY TABLE. THE MEANS CAN BE BASED ON UNEQUAL FREQUENCIES IN THE CELLS. THE PROCEDURE USED IS FROM WINER, PAGE 241. IN GENERAL, 3 VARIABLES ARE INVOLVED. ONE DETERMINES THE ROW LEVEL OF THE P BY Q TABLE, THE SECOND THE COLUMN LEVEL, AND THE THIRD VARIABLE IS THE ONE ACTUALLY ANALYZED.

IN = INPUT DATA FILE. CONTAINS THE ROW VARIABLE, THE COLUMN VARIABLE ( OR VARIABLES - SEE BELOW ), AND THE VARIABLES TO BE ANALYZED. MISSING DATA IS ALLOWED ANYWHERE.

ROW = LABEL OF THE ROW VARIABLE / HIGHEST ROW VALUE

COL = LABEL OF THE COL VARIABLE / HIGHEST COLUMN VALUE

FIRST = POSITION ( I.E., NOT A LABEL ) OF THE FIRST VARIABLE TO BE ANALYZED. IF OMITTED, ALL VARIABLES ARE ANALYZED, INCLUDING THE ROW AND CURRENT COLUMN VARIABLES.

CONSIDER THE FOLLOWING FILE OF DATA ABOUT EMPLOYEES IN A MANAGEMENT TRAINING PROGRAM.

VARIABLE 1	AGE(WHEN HIRED)	21 TO 30
VARIABLE 2	C.AGE ( CODED AGE )	1 = 21 TO 23 2 = 24 TO 26 3 = 27 TO 30
VARIABLE 3	EDUCAT ( EDUCATION )	0=LESS THAN HIGH SCHOOL 1=HIGH SCHOOL 2=COLLEGE DEGREE 3=GRADUATE WORK
VARIABLES 4-20	THESE ARE SCORES ON A TEST ( CODED 1 TO 6 ) GIVEN AT THE TIME THE PERSON WAS EMPLOYED. THEREFORE, THEY MIGHT BE PREDICTORS OF LATER SUCCESS.	
VARIABLES 21-40	THESE ARE EVALUATIONS OF THE EMPLOYEE AFTER A PERIOD OF TIME AND COULD BE USED AS CRITERIA OF THE PERSONNEL SELECTION PROCESS.	

CONSIDER A 2 WAY TABLE OF CODED AGE AND EDUCATION,  
A 3 BY 4 TABLE. IT MIGHT LOOK LIKE THIS...

	0	1	2	3
1	0	5	7	5
2	1	6	9	11
3	1	3	3	4

THERE ARE 7 PEOPLE BETWEEN 21 AND 23 WITH A COLLEGE DEGREE,  
AND SO FORTH. SO FAR, 2 VARIABLES ARE INVOLVED, A ROW VARIABLE  
( C.AGE ), AND A COLUMN VARIABLE ( EDUCAT ). THE HIGHEST LEVEL  
IN C.AGE IS 3, SINCE IT IS CODED 1, 2, AND 3.

IF VARIABLE 40 IS AN EVALUATION AFTER 2 YEARS, WHERE 1 IS  
BAD AND 9 IS EXCELLENT, WE COULD GATHER THE SCORES ON VARIABLE  
40 FOR THE 7 PEOPLE AND COMPUTE A MEAN FOR THAT CELL. SIMILARLY,  
MEANS FOR THE OTHER CELLS COULD BE FOUND. AND A TABLE OF MEANS  
ON VARIABLE 40 COULD BE PRINTED. THIS PROGRAM PRODUCES SUCH  
TABLES AND DOES A FACTORIAL ANALYSIS OF VARIANCE ON THEM,  
YIELDING P VALUES FOR THE ROW, COLUMN, AND INTERACTION EFFECTS.

THE ORIGINAL EXAMPLE WOULD DO SUCH AN ANALYSIS.

-----TABLE SIZE-----

VERSION 52 - THE TABLE CAN BE 5 BY 10. IT ASSUMES, HOWEVER, THAT  
0 IS THE LOWEST ( FIRST ) LEVEL. THEREFORE, 4 IS THE HIGHEST ROW  
VALUE, AND 9 IS THE HIGHEST COLUMN VALUE IN THE TABLE.  
50 VARIABLES ARE ANALYZED IN EACH PASS THROUGH THE FILE ( FOR A  
GIVEN ROW AND COLUMN PAIR ). IF MORE THAN 50 VARIABLES ARE TO BE  
ANALYZED, THE PROGRAM AUTOMATICALLY MAKES SEVERAL PASSES.

-----RF-CODING WITHIN PRVQ-----

THE ROW VARIABLE AND/OR THE COLUMN VARIABLE CAN BE RE-CODED.  
THIS IS SIGNALLED BY A NEGATIVE VALUE ON THE P-STAT CARD FOR THE  
HIGHEST VALUE.

ROW = AGE / -4                      SAYS...

VARIABLE AGE IS THE ROW VARIABLE AND IS TO BE RE-CODED TO HAVE  
FIVE LEVELS ( 0, 1, 2, 3, AND 4 ). A DATA CARD MUST FOLLOW  
THE PRVQ CARD WITH THE ACTUAL RANGES ASSOCIATED WITH EACH LEVEL.  
FOR EXAMPLE.....

20. 22. 23. 23. 24. 25. 26. 27. 28. 29.

CAUSES PEOPLE WITH AGES OF 20 TO 22 TO FALL INTO ROW 0,  
THOSE OF AGE 23 WILL BECOME A 1, ETC.

IF BOTH ROWS AND COLUMNS ARE RF-CODED, THE ROW CARD(S) COMES  
FIRST. THE COLUMN RANGES BEGIN ON A NEW CARD. THE FORMAT OF



7 ACTUAL COLUMNS, 0-1-2-3-4-5-6 ). TWENTY TABLES ARE RUN  
( VARIABLES 21-40 ) BECAUSE FIRST=21 WAS CITED, AND 40 IS THE  
LAST VARIABLE IN THIS FILE. AFTER THIS, COLUMN 4 ON THE LOOP  
CARD IS BLANKED ( SINCE IT HAS NOW BEEN DONE ) AND ANOTHER  
COLUMN IS FOUND, AND SO FORTH.

17. OTHER PROGRAMS

DURWAT  
GENVAR  
TCOR  
TTEST  
VBAR

\*\*\*\*\*

DURWAT, IN=A, OUT=B \$

THIS TAKES A COMPLETE DATA ( NO MISSING DATA )  
INPUT FILE AND, FOR EACH COLUMN, COMPUTES A DURBIN-WATSON  
VALUE. THESE, GROUPED AS A COLUMN VECTOR, MAKE UP THE OUTPUT  
MATRIX. IF, FOR EXAMPLE, THE INPUT IS 600 BY 45, THE RESULT  
WILL BE 45 BY 1.

\*\*\*\*\*

GENVAR, IN=A, OUT=B\$  
LABEL CARDS (IF NEEDED)  
TRANSFORMATION CARDS  
\*END

GIVEN A DATA FILE, AND PERHAPS A SUMMARY FILE  
OF IT (MEANS, VARIANCES, ETC.), PRODUCE AN OUTPUT FILE OF  
DATA. THIS CAN BE THE ORIGINAL VARIABLES AND/OR TRANSFORMA-  
TIONS OF THEM, OR OF SOME OF THEM.

IN = NAME OF INPUT FILE  
OUT = NAME FOR OUTPUT FILE  
DES = NAME OF MEANS, VARIANCES, RANGES, ETC., OF IN. IF NO  
OPERATIONS REQUIRING THESE ARE USED, DES CAN BE OMITTED.  
NC = NUMBER OF OUTPUT COLUMNS, IF MORE OR LESS THAN THE  
NUMBER OF INPUT COLUMNS.  
THIS IS NOT NEEDED IF THE OUTPUT FILE IS TO HAVE THE SAME  
NUMBER OF COLUMNS AS THE INPUT FILE.

IF THE OUTPUT HAS MORE COLUMNS THAN THE INPUT, LABELS  
FOR THE NEW VARIABLES MUST BE SUPPLIED. EACH NEW  
LABEL IS ON A CARD, IN COLUMN 1-8. THESE CARDS, IF NEEDED,  
FOLLOW THE GENVAR STATEMENT AND ARE FOLLOWED BY THE  
TRCARD CARDS (TRANSFORMATION CARDS).

THERE CAN BE A NUMBER (UP TO 200) OF TRANSFORMATION CARDS,  
EACH SPECIFYING AN OPERATION ON A VARIABLE (OR A  
RANGE OF VARIABLES). A CARD WITH \*END IN COLUMNS 1-4 MUST FOLLOW  
THE LAST TRCARD.

FORMAT OF TRANSFORMATION CARDS...

COL 1-6 TRCARD  
COL 9-10 OPERATION CODE.  
COL 11-15 FIRST I VARIABLE, REFERRED TO NOW AS JA. IF BLANK

OR ZERO, IT IS SET TO 1 AND IB (SEE BELOW) IS SET TO THE LAST COLUMN NUMBER. IN OTHER WORDS, ALL VARIABLES WOULD BE INCLUDED IN THE RANGE.

COL 16-20 LAST I VARIABLE (CALLED IB). IF BLANK OR ZERO, IT GETS SET TO IA (UNLESS IT WAS ALREADY SET TO THE LAST COLUMN NUMBER).

COL 21-25 FIRST J VARIABLE. IF BLANK OR ZERO IT IS SET TO IA.

COL 26-30 FIRST K VARIABLE. IF BLANK OR ZERO IT IS SET TO IA.

COL 31-35, 36-40, ...55-60. SIX CONSTANTS IF NEEDED, READ IN 6F5.0 FORMAT.

IN GENERAL, IA OF J GOES INTO K. A J VARIABLE IS NEEDED IF COLUMNS 9-10 HAVE THE CODE OF A BINARY OPERATOR. IF IT IS AN ADD, AND IA=4, IB=8, J=20, K=30, THEN VARIABLE 4 + VARIABLE 20 ARE PLACED INTO LOCATION 30. SINCE 4 IS NOT 8, 1 GETS ADDED TO IA, J, AND K, AND V5 + V21 GOES INTO 31, ETC., UNTIL V8 + V24 GOES INTO V34. IF, IN THE LAST STEP, EITHER V8 OR V24 WERE MISSING, V34 WOULD BE SET TO MISSING, FOR EXAMPLE.

THE DATA FILE IS READ IN ONE ROW AT A TIME. ALL TRCARD ACTIVITIES ARE PERFORMED ON THAT ROW IN THE ORDER THAT THEY WERE THEMSELVES READ. IN OTHER WORDS, THE SECOND TRCARD OPERATES ON THE RESULTS OF THE FIRST. THE AREA WHERE THE ROW IS READ IS 1000 WORDS LONG, SO A TEMPORARY VARIABLE CAN BE GENERATED IN ONE TRCARD AND USED BY THE NEXT TRCARD.

THE OPERATIONS, WHERE X IS THE IA VARIABLE, Y IS THE J VARIABLE, AND Z IS THE RESULT (THE K VARIABLE). THE FIRST 24 OF THESE ARE BASED ON THE BINM TRANSGENERATIONS. A STARRED OPERATION NEEDS THE SECOND INPUT MATRIX.

1 SQRT(X)  
 2 SQRT(X) + SORT(Y+1.)  
 3 LOG10(X)  
 4 EXP(X)  
 5 ASN(SQRT(X)) ... ARCSINE OF SQUARE ROOT OF X  
 6\* ASN(SQRT(Y/(N+1))) + ASN(SQRT((X+1.)/(N+1)))  
 7 1/X  
 8 X+C (NOTE... C IS THE FIRST CONSTANT)  
 9 X\*C  
 10 X\*\*C  
 11 X+Y  
 12 X-Y  
 13 X\*Y  
 14 X/Y  
 15 IF X .GE. C, SET Z TO 2. ELSE SET Z TO 1. (NOTE -  
 .GE. MEANS GREAT ER THAN OR EQUAL TO)  
 16 IF X .GE. Y, SET Z TO 2. ELSE SET Z TO 1.  
 17 LOG(Y)  
 18\* X-MEAN THE RESULT IS THE ORIGINAL SCORE WITH THE MEAN  
 SUBTRACTED FROM IT. THE MEAN IS FOUND IN THE DES FILE.  
 19\* X/(SD OF X). DIVIDE BY THE STANDARD DEVIATION.

20 SIN (X)  
 21 COS (X)  
 22 ATAN (X)  
 23 X\*\*Y  
 24 C\*\*X  
 25 ABSOLUTE VALUE OF X  
 26 IF X = CONSTANT 2,3,4,5, OR 6, SET Z TO C (I.E.,  
 CONSTANT 1). IF NOT, SET Z TO X. IF THERE ARE NOT  
 CONSTANTS 2-6, REPEAT THE LAST ONE THROUGH THE SIXTH  
 POSITION.  
 27\* IF X .GE. XMEAN, THEN Z=2. ELSE Z=1.  
 28\* (X-XMEAN)/XSD. CONVERT TO A STANDARDIZED FORM.  
 29 IF X .GE. C, Z=1. IF X .LE. -C, Z=3. ELSE Z=2.  
 HERE, IF THE X'S ARE NORMAL WITH MEAN 0, VARIANCE  
 1, THE USE OF 0.43 AS THE CONSTANT WILL SEPARATE  
 THE SCORES INTO 3 ROUGHLY EQUAL GROUPS.  
 30 IF X .LE. C (1), Z=1. THEN, IF X .LE. C (2), Z=2.,  
 AND SO FORTH THROUGH 6. AS IN OPERATION 26,  
 REPEAT THE LAST CONSTANT, IF NECESSARY, UNTIL THE SIXTH  
 CONSTANT IS FILLED. IF GREATER THAN ALL  
 OF THEM, SET Z TO MISSING. THIS IS USEFUL IN SET-  
 TING UP GROUPS FOR THE PBYQ PROGRAM.  
 31 IF X IS MISSING, Z=C. ELSE Z=X.  
 32 IF C (1) .LE. X .AND. X .IF. C (2), SET Z TO Y. ELSE  
 SET Z MISSING. THIS IS USED TO CHECK RANGES ON  
 DATA.  
 33 IF (AS IN 32) X IS BETWEEN C (1) AND C (2), SET Z TO  
 C (3). IF NOT, IF X IS BETWEEN C (4) AND C (5), SET Z  
 TO C (6). IF STILL NOT, SET Z TO MISSING.  
 34 Z=X

SOME EXAMPLES.....

TRCARD 01 1 5 1

THE ABOVE CARD SAYS... COMPUTE THE SQUARE ROOT OF  
VARIABLES 1-5 AND PLACE THE RESULTS IN LOCATIONS 1-5  
(I.E., REPLACING THE ORIGINAL VALUES).

TRCARD 11 13 15 21 31

VARIABLE 31 WILL BE VARIABLE 13 + VARIABLE 21,  
VARIABLE 32 WILL BE VARIABLE 14 + VARIABLE 22,  
VARIABLE 33 WILL BE VARIABLE 15 + VARIABLE 23.

\*\*\*\*\*

TCOR, IN=A, OUT=B \$

T VALUES OF PAIRED OR CORRELATED DATA. THIS TAKES A  
DATA FILE AND PRODUCES A SYMMETRIC OUTPUT MATRIX OF T VALUES  
OF DIFFERENCES BETWEEN SCORES IN ALL PAIRS OF COLUMNS OF  
THE DATA MATRIX TAKEN IN PAIRS. MISSING DATA IS ALLOWED.  
WHEN A SCORE IS FOUND TO BE MISSING, IT IS NOT USED, AND ANY T

SCORES INVOLVING THAT VARIABLE DO NOT USE THAT ROW OF DATA AT ALL.

VERSION 52 - THE INPUT FILE CAN HAVE UP TO 80 COLUMNS.

IN = INPUT DATA FILE NAME.  
OUT = NAME OF SYMMETRIC MATRIX OF T SCORES.

\*\*\*\*\*

TTEST, IN=A, OUT=B \$

T VALUES OF UNCORRELATED DATA. THIS TAKES A DATA FILE AND PRODUCES A SYMMETRIC OUTPUT MATRIX OF T VALUES. THE 1, 2 POSITION OF THE OUTPUT MATRIX REPRESENTS A TEST OF THE SIGNIFICANCE OF THE DIFFERENCE BETWEEN THE MEAN OF ALL THE NON-MISSING SCORES IN COLUMN 1 OF THE INPUT AND COLUMN 2'S MEAN. THE POSITIONS OF THE SCORES IN COLUMN 2 OF THE INPUT FILE COULD BE REARRANGED IN SOME RANDOM WAY, KEEPING COLUMN 1 UNCHANGED (EXAMPLE, EXCHANGE 4,2 AND 7,2) AND THE T OF 1,2 WOULD NOT CHANGE. (IN SUCH A SITUATION, THE OUTPUT OF TCOR WOULD PROBABLY BE AFFECTED.)

VERSION 52 - THE INPUT FILE CAN HAVE UP TO 150 COLUMNS.

IN = INPUT DATA FILE.  
OUT = NAME OF OUTPUT T MATRIX. THE DIAGONAL CONTAINS STANDARD ERRORS.

\*\*\*\*\*

VBAR = (I) \$

THE 360 P-STAT SYSTEM USES A VERTICAL BAR CHARACTER ( A 12/7/8 PUNCH ) IN CROSSTAB TO FORMAT TABLES. THIS OPERATION REPLACES THAT CHARACTER WITH THE FIRST CHARACTER INSIDE THE PRENS.

## 18. CREATING A P-STAT FILE FROM CARDS ( DATA )

\*\*\*\*\*

DATA = A, NV = 20, DES = D \$

THE PURPOSE OF THIS PROGRAM IS TO READ DATA PUNCHED ON CARDS AND PRODUCE A P-STAT DATA FILE ON TAPE OR ON DISK. IF THIS IS DONE CORRECTLY, IT SHOULD NOT BE NECESSARY TO TOUCH THESE CARDS AGAIN AS LONG AS THE OPERATIONS THAT ARE TO BE PERFORMED ON THE DATA ARE WITHIN THE RANGE OF P-STAT'S OPERATIONS. A CONSIDERABLE AMOUNT OF CHECKING AND VALIDATING IS DONE AS THE CARDS ARE READ.

IT IS USUALLY COSTLY TO READ THE SAME RAW DATA CARDS THROUGH THE DATA PROGRAM ON REPEATED RUNS. IF THERE ARE MORE THAN A HUNDRED OR SO INPUT CARDS, AND IF THE FILE IS GOING TO BE USED ON A NUMBER OF SEPARATE P-STAT RUNS, SAVING THE FILE ( USING THE SAVE OPERATION ) ON A USER'S PERMANENT DATA TAPE IS ADVISABLE. IF THE FILE IS VERY LARGE ( PERHAPS 2000 BY 150 ), SO THAT THE TIME SPENT USING THE FIND OPERATION IS SIGNIFICANT, USING AN ASSIGN / ATTACH TAPE AS WELL IS A PRUDENT THING TO DO. A TYPICAL RUN MIGHT DO SOME FREQUENCIES, WITH THE INPUT FILE ON AN ATTACH TAPE AND THE DESCRIPTION FILE OF IT ON A PERMANENT DATA TAPE ( OR ON CARDS ).

IDENTIFIERS...

DATA=A, NV=20\$

THE RESULTING FILE WILL BE NAMED A, AND IT WILL HAVE 20 COLUMNS (VARIABLES). THESE TWO IDENTIFIERS ARE ALWAYS NEEDED.

VERSION 52 - NV CANNOT EXCEED 450. FROM 1 TO 50 DATA CARDS PER FILE ROW ARE ALLOWED.

DES=D,

THIS IS FREQUENTLY USED. D WILL BE A DESCRIPTION FILE, NV ROWS BY 6 COLUMNS. EACH OF THE ROWS HAS THE MEAN, VARIANCE, STANDARD DEVIATION, LOWEST SCORE, HIGHEST SCORE, AND N (THE NUMBER OF NON-MISSING SCORES) OF THE CORRESPONDING COLUMN OF THE DATA FILE.

PRINT,

USE OF THIS IDENTIFIER CAUSES EACH DATA CARD TO BE PRINTED AS IT IS READ IN.

PLUS = 12,

CONSIDER A VARIABLE CODED ZERO TO NINE, AND ALSO MINUS AND PLUS. IF 'PLUS=12' IS INDICATED, ANY VARIABLE PUNCHED AS A PLUS SIGN ( OR, IF MORE THAN ONE COLUMN, PLUS SIGNS ), WILL BE CONVERTED BY P-STAT INTO A TWELVE. IF 'PLUS = 33' WERE CITED, THE PLUSES ( SOMETIMES CALLED '12' OR 'Y' PUNCHES ) WOULD BECOME 33, ETC.

NOTE - 'PLUS' WILL CAUSE SEARCHING FOR BOTH A 12 PUNCH ( A + ON AN 026 KEYPUNCH ) AND A 12-6-8 PUNCH ( A + ON AN 029 KEYPUNCH ).

MINUS = 11,

THIS HANDLES MINUS SIGNS ( SOMETIMES CALLED '11' PUNCHES OR 'X' PUNCHES ). SEE ABOVE DESCRIPTION OF 'PLUS'. WHEN THESE ARE USED, IT IS OCCASIONALLY NECESSARY TO RECODE ZEROS INTO TENS. THIS CAN BE DONE IN A SUBSEQUENT STEP USING, FOR EXAMPLE, ( FOR ( 1-20), IF .X. EQ 0 SFT .X. TO 10 )

TAPE = 91,

THIS INDICATES THAT THE CARDS ARE NOT INCLUDED IN THE USUAL INPUT CARD DECK, BUT ARE INSTEAD ON TAPE 91, AS 80 CHARACTER RECORDS. NUMBERS 91-95 ARE NOT USED BY PSTAT. A USER, ON PRINCETON'S 360, WOULD NEED TO SUPPLY THE 'DD' CARD DESCRIBING TAPE 91 ( IF THIS OPTION WERE USED ). SEE THE SECTION 'JCL FOR REFERENCING TAPES ON THE 360'.

EPRINT= MAXIMUM NUMBER OF ERRORS TO BE PRINTED.

IF THIS IS OMITTED, THE FIRST 100 BAD SCORES WILL BE PRINTED, BUT NO MORE. THIS IS SO A REPEATED ERROR, ONE THAT IS WRONG ON EVERY CASE, DOES NOT PRINT INDEFINITELY. AN ERROR IS A SCORE THAT WAS NOT BLANK OR MISSING BUT WAS STILL SET TO BE MISSING BY THE HIGH, LOW, OR VALID PUNCHING TESTS (DESCRIBED BELOW).

EKILL= MAXIMUM NUMBER OF ERRORS PERMITTED AT ALL.

IF OMITTED, THIS IS SET TO NV\*10. WHEN EXCEEDED, THE DATA PROGRAM PRINTS OUT THE SUMMARY OF THE CARDS READ SO FAR, BUT THEN KILLS THE RUN.

CARDS= UPPER LIMIT ON THE NUMBER OF DATA CARDS TO BE READ. IF THIS IS OMITTED, THE DATA PROGRAM GENERATES A DEFAULT VALUE OF 1,000,000. IF, FOR EXAMPLE, 'CARDS = 100' WERE PUNCHED, THE PROGRAM WOULD READ ONLY THE FIRST 100 CARDS FROM THE FILE WHERE THE CARDS ARE. THE PROGRAM WOULD THEN PRINT ITS REPORT AND GO ON TO THE NEXT STEP.

ONE SHOULD ALWAYS CHECK OUT ONE'S LABELS AND FORMATS WITH A SMALL NUMBER OF INPUT CARDS BEFORE READING A LARGE AMOUNT OF INPUT CARDS. NOTE...THE DEFAULT SETTING OF 1,000,000 IMPLIES THAT THE END OF THE INPUT CARD FILE IS INDICATED BY.....

1. A '\*END' CARD IF THE CARDS FOLLOW THE '\*CARDS' CARD.

OR...

2. A '\*END' CARD ( OR AN ENDFILE ) IF 'TAPE=' IS BEING USED.

'CARDS=' SHOULD NOT BE USED UNLESS 'TAPE=' IS ALSO USED.

CONSIDER...

DATA=AX, NV=21, DES=D, EPRINT=30, EKILL=500, TAPE=91, CARDS=350,  
MINUS=11 \$

THE FILE WILL BE NAMED 'AX'.

IT WILL HAVE 21 COLUMNS.

A DESCRIPTION FILE OF AX WILL BE NAMED D.

THE FIRST 30 ERRORS (BAD PUNCHES, HIGH OR LOW SCORES) WILL BE PRINTED.

IF 500 ERRORS OCCUR, THE RUN WILL BE KILLED.

THE DATA CARDS ARE ON TAPE 91.  
350 CARDS WILL BE READ FROM THAT TAPE.  
A SCORE PUNCHED AS JUST A MINUS SIGN WILL BECOME AN 11. .

-----  
THERE ARE SEVERAL TYPES OF CARDS OTHER THAN DATA CARDS  
READ BY THE PROGRAM. THEIR USE IS SHOWN BY EXAMPLES BELOW.  
THEY ARE...

- 1 \*FMT THIS HAS THE FORMAT OF THE DATA CARDS.
- 2 \*LAB IF USED, PROVIDES LABELS FOR THE VARIABLES.
- 3 \*HIGH IF USED, A CASE FOLLOWS IT IN WHICH EACH PUNCHED  
SCORE IS THE HIGHEST PERMITTED FOR THAT VARIABLE.  
ANY HIGHER ARE BAD.
- 4 \*LOW IF USED, LIKE \*HIGH BUT A LOW RANGE TEST.
- 5 \*MISS IF USED, A DUMMY CASE FOLLOWS IT. LATER, ANY  
SCORES ON REAL DATA THAT MATCH ITS PUNCHING  
ARE SET TO MISSING. THE PUNCHING CAN BE  
ALPHA-NUMERIC.
- 6 \*CARDS THIS SAYS.. THATS ALL THE CONTROL INFORMATION.  
THIS CARD MUST BE PRESENT.  
DATA CARDS FOLLOW THIS (UNLESS TAPE=N WAS USED).
- 7 \*END THIS FOLLOWS THE LAST ACTUAL DATA CARD TO THE 'DATA'  
PROGRAM AND INDICATES THAT ALL THE DATA CARDS HAVE  
BEEN READ. IF CARDS=N WAS USED, \*END SHOULD NOT BE USED.  
IF TAPE=N WAS USED, THE END OF FILE SERVES AS A \*END.

-----  
AN EXAMPLE.....

DATA=A, NV=10, PRINT \$  
\*FMT ( 10I1 )  
\*CARDS  
HERE ARE THE DATA CARDS, PERHAPS 30 OF THEM.  
\*END

THIS WILL PRODUCE A FILE WITH 10 COLUMNS, NAMED A. THE  
DATA CARDS ARE PUNCHED AS DESCRIBED IN THE \*FMT CARD (USING  
COLUMNS 1-10, PUNCHED AS INTEGERS, ONE COLUMN PER VARIABLE).  
THE \*CARDS CARD SAYS,... HERE COME THE USERS ACTUAL DATA  
CARDS. THERE CAN BE ANY NUMBER OF THEM. THE \*END CARD SAYS  
THAT IS THE END OF THE DATA.

THE RESULTING FILE A HAS THE ROWS LABELLED L1 THRU L30  
(IF 30 ROWS) AND THE COLUMNS ARE LABELLED L1 THRU L10. IF A  
VARIABLE WAS ENTIRELY BLANK (OR WAS MISPUNCHED, LIKE A LETTER  
INSTEAD OF A NUMBER), THAT VARIABLE FOR THAT CASE IS CONSIDERED  
BY P-STAT TO BE MISSING.

THE INPUT CARDS WILL BE PRINTED BECAUSE 'PRINT' WAS SPECIFIED.  
-----

AN EXAMPLE.....

```

DATA=B, NV=3$
*LAB INPORT, EXPORT, GROSS
*FMT ( 10X F10.2, F10.2, F10.2 )
*CARDS
  HERE ARE THE DATA CARDS, FOR EXAMPLE, 50 OF THEM.
*END

```

HERE, B WILL BE 50 BY 3. THE COLUMNS ARE LABELLED INPORT, EXPORT, AND GROSS. THE ROWS ARE LABELLED L1 THRU L50. ON THE DATA CARDS, INPORT IS FOUND IN COLUMNS 11-20 (10X MEANS SKIP 10 COLUMNS), EXPORT IN COLUMNS 21-30, AND GROSS IN COLUMNS 31-40. IF IN COL. 11-20 THERE IS NO DECIMAL POINT PUNCHED, IT WILL BE ASSUMED TO BE BETWEEN COL. 18 AND 19. F10.2 MEANS A DECIMAL POINT TYPE NUMBER (A FLOATING POINT NUMBER) USING 10 COLUMNS ON THE CARD, OF WHICH THE RIGHTMOST 2 ARE TENTHS AND HUNDRETHS PLACES ( IF NO POINT WAS EXPLICITLY PUNCHED ).

THIS \*FMT CARD COULD HAVE BEEN

```

*FMT ( 10X 3F10.2 )

```

AGAIN, IF A VARIABLE ON A CASE IS ENTIRELY BLANK OR IS MISPUNCHED, THAT SCORE IS SET TO THE MISSING SCORE.

-----

NOW, SOME \*LAB EXAMPLES, AND THE RULES FOR LABELS...

```

DATA=A, NV=4$
*LAB AGE, HEIGHT, WEIGHT, SEX

```

..... OR .....

```

DATA=A, NV=4$
*LAB AGE
*LABHEIGHT
*LAB WEIGHT,      SEX

```

- 1.- THERE CAN BE AS MANY \*LAB CARDS AS ONE NEEDS. EACH MUST START WITH \*LAB IN COLUMNS 1 TO 4.
- 2.- LABEL CAN BE 1-8 CHARACTERS. THE FIRST MUST BE A LETTER. THE REST (IF ANY) CAN BE LETTERS, NUMBERS, OR A DECIMAL POINT. A BLANK CANNOT BE PART OF A LABEL.
- 3.- A COMMA ENDS A LABEL (BUT IS NOT PART OF THE LABEL).
- 4.- YOU MAY LEAVE OFF A COMMA IF THE REST OF THE CARD IS BLANK.
- 5.- IF YOU DO NOT SUPPLY LABELS, THE PROGRAM GENERATES THEM (L1 FOR VARIABLE 1, L2 FOR VARIABLE 2, ETC.).
- 6.- THE \*LAB (IN COL 1-4) CARD OR CARDS CAN BE ANYWHERE BEFORE THE \*CARDS CARD. IF SEVERAL, THEY MUST BE TOGETHER.
- 7.- IF NV=20 AND IF YOU HAVE A \*LAB CARD(S), THERE MUST THEN BE

EXACTLY 20 LABELS.

CONSIDER.....

DATA=A, NV=6\$

\*LAB AGE, ITEM21, ITEM22, ITEM23, ITEM24, ITEM25

THIS CAN BE ABBREVIATED BY.....

\*LAB AGE, ITEM\*(21,25)

- 1.- THE STAR FORM OF LABEL CAN GENERATE A NUMBER OF LABELS.
- 2.- ITS FORM IS ITEM\*(NA,NB).
- 3.- IT WILL GENERATE NB-NA+1 LABELS.
- 4.- IT, OF COURSE, PLUGS NA IN FOR THE STAR, THEN NA+1, UNTIL NB HAS BEEN USED.
- 5.- AFTER THE PLUG-IN, IT STILL MUST BE 1-8 CHARACTERS.
- 6.- \*(5,8) IS FINE. ( HOWEVER, YOU WILL GET L5,L6,L7,AND L8. )
- 7.- ITEM\*(1,9) IS FINE.
- 8.- VLABEL\*(101,110) IS WRONG BECAUSE IT WILL BE OVER 8 CHARACTERS.
- 9.- \*ITEM(1,30) IS WRONG. THE STAR MUST BE ON THE RIGHT.
- 10- \*(61,80) IS FINE.
- 11- \*LAB AGE, ITEM\*(11,20), HEIGHT, SCORE\*(31,40) IS FINE.

-----

NOW, FORMAT ASPECTS. HOW TO READ ROW LABELS, AND HOW TO BE SURE ALL THE CARDS OF A ROW ARE TOGETHER. A-TYPE FORMAT IS USED FOR THESE PURPOSES.

\*FMT ( A6, 10F1.0 ) SAYS... USE COL. 1-6 AS A ROW LABEL, FIND 10 VARIABLES IN COLUMNS 7-16.

\*FMT ( A5, 75F1.0 / 30X A5, 20F1.0 ) SAYS... THERE ARE 2 DATA CARDS FOR EACH CASE. USE COL. 1-5 OF CARD 1 AS THE ROW LABEL, FIND 75 VARIABLES IN COL. 6-80 OF CARD 1. THEN, ON CARD 2 FOR THIS CASE, COL. 31-35 MUST MATCH EXACTLY ALL EARLIER A FIELDS FOR THE CASE (NAMPLY, COL. 1-5 OF CARD 1).

- 1.- A-FIELDS, IF MORE THAN 1, MUST BE THE SAME WIDTH AS EACH OTHER.
- 2.- IF OVER 8 CHARACTERS, ALL RIGHT, BUT ONLY THE FIRST 8 WILL BE USED AS THE ROW LABEL.
- 3.- THERE CAN BE ONLY 1 A FORMAT FIELD PER INPUT CARD. IT CAN BE ANYWHERE ON THE CARD, AND CAN CONSIST OF ANYTHING AT ALL (THAT CAN BE READ BY THE CARD READER). WHEN, FOR EXAMPLE, THE 63RD CASE IS READ, THE PROGRAM TRIES TO MAKE A ROW LABEL OUT OF THE 'A' FIELD. IT SUBSTITUTES '.' FOR ANY CHARACTER NOT A LETTER, PERIOD, OR NUMBER AND ALSO SQUEEZES OUT BLANKS. IF ALL BLANK OR ALL PERIODS, IT SUPPLIES 'L63'. IF THE LABEL DOES NOT START WITH A LETTER, IT GENERATES AN INITIAL 'L'. THUS, '3A \*R' BECOMES 'L3A.R'. ALSO '12345678' BECOMES 'L2345678'.
- 4.- IF CARDS ARE OUT OF ORDER (IF, IN THE ABOVE, CARD 1 CCL.

1-5 IS NOT THE SAME AS CARD 2 COL. 31-35) THE PROGRAM GRUMBLES, THROWS OUT ALL BUT THE LAST READ CARD OF THAT ROW AND TRIES AGAIN WITH THAT (LAST) CARD AS THE FIRST OF A NEW ROW.

5.- IF IT FAILS 5 TIMES WITHOUT ANY SUCCESS, IT GIVES UP.

6.- \*PMT ( A6, F10.2 /

\*PMT 10X A6, F3.1 ) IS FINE.

COLUMNS 5-80 CAN BE USED FOR THE FORMAT TEXT.

IF THE FORMAT TEXT DOES NOT FIT IN COLUMN 5-80 OF THE FIRST

\*PMT CARD, ANOTHER \*PMT CARD FOLLOWS, AND SO ON.

-----  
HOW TO BE SURE THE CARDS OF A ROW ARE REALLY IN THE RIGHT ORDER...

\*PMT ( A5, 1X 10F1.0 / A5, 1X 3F1.0 )

HERE, THE A-CHECK CAN BE GOOD EVEN THOUGH THE 2 CARDS FOR THIS ROW WERE SWAPPED. SUPPOSE COL. 6 ON CARD 1 CONTAINS THE CHARACTER 1, COL. 6 ON CARD 2 CONTAINS THE CHARACTER 2. THEN...

\*PMT ( A5, \*1\*, 10F1.0 / A5, \*2\*, 3F1.0 )

WHEREVER 2 STARS APPEAR WITH BLANK OR NCN BLANK CHARACTERS BETWEEN THEM, IT MEANS THE CONTENTS OF THE INDICATED COLUMNS OF THE DATA CARDS SHOULD EXACTLY DUPLICATE WHAT IS BETWEEN THE STARS.

\*PMT ( \*IR\*, A10, F2.0, \*RELHUB\* / \*2W\*, A10 )

HERE, COL 1-2 OF CARD 1 ( OF A CASE ) MUST BE IR, COL 15-20 MUST BE RELHUB, AND COL. 1-2 OF CARD 2 MUST BE 2W.

\*PMT ( \* \*, 9X 3\*KZ\* / A5, 20F1.0, 55\* \* )

HERE, COL. 1 OF CARD 1 MUST BE BLANK, COL. 11-16 MUST BE KZKZKZ (NOTE, 3 USES OF KZ), AND COL. 26-80 OF CARD 2 MUST BE BLANK.

1.- THE ONLY KEYPUNCHABLE CHARACTER THAT CANNOT BE BETWEEN 2 STARS IS A STAR.

2.- STAR FIELDS CAN BE USED WITHOUT RESTRICTION.

3.- IF A CHECK FAILS, THE PROGRAM GRUMBLES, THROWS OUT ALL BUT THE LAST READ CARD OF THAT ROW, AND TRIES AGAIN WITH THAT (LAST) CARD AS THE FIRST OF A NEW ROW. AGAIN, 5 QUICK ERRORS LOSES THE GAME.

-----

OTHER FEATURES..... -X AND T.

\*PMT ( A6, -X F1.0 )

COL. 1-6 ARE A LABEL, -X SAYS MOVE BACK 1 COLUMN, SO THE F1.0 CAN READ COL. 6 ALSO.

1.- THE FOLLOWING ARE ALL LEGAL. -X, -7X, -60X, AS LONG AS YOU DO NOT MOVE OFF THE CARD. ( A10, -20X... IS A LOSER. SO IS ( A10, 80X...

\*PMT ( A6, T80, F1.0, T6, F1.0, T80, \*7\* )

TN MEANS... SET THE POINTER SO THAT THE NEXT COLUMN USED IS COLUMN N. HERE THE LABEL IS IN COL. 1-6, THE FIRST VARIABLE IS READ FROM COL. 80, THE SECOND FROM COL. 6 (AGAIN). THEN COL. 80 IS CHECKED OUT AS A 7. THEREFORE, ALL ROWS OF THIS FILE WILL HAVE A SCORE OF 7 ON VARIABLE 1.

-----  
A LARGE EXAMPLE

```
DATA=A, DES=D, NV=20$
*LAB AGE, *(2,20)
*PMT ( A6, 4X I5, T31, *12332*, 19F1.0, T71, E10.5 )
*HIGH
    A CARD      ( ONE, BECAUSE THERE IS ONLY 1 CARD PER CASE)
*LOW
    A CARD
*MISS
    A CARD
*CARDS
    A NUMBER OF DATA CARDS
*END
```

THERE ARE 3 NEW THINGS HERE.

\*HIGH- AFTER THIS CARD SHOULD COME A SET OF DATA (A ROW, A CASE) CORRECTLY SEQUENCED ( A FIELDS AND STAR FIELDS, IF USED). HOWEVER, THE NON BLANK SCORES ON THIS CASE DEFINE THE HIGHEST ALLOWABLE SCORE ON EACH VARIABLE. WHEN ACTUAL DATA IS READ, ANY SCORES EXCEEDING IT ARE SET TO MISSING. IF A SCORE IS BLANK ON THE \*HIGH CASE, THE PROGRAM IGNORES THE HIGH TEST ON THAT VARIABLE.

\*LOW- SAME AS \*HIGH BUT THE LOW END.

\*MISS- AGAIN, MUCH THE SAME, BUT THIS IS READ IN 8CA1 FORMAT. THIS PERMITS MISSING DATA TO BE OTHER PUNCHINGS BESIDES BLANK.

1.- EVEN IF \*MISS IS USED, BLANKS STILL DEFINE MISSING SCORES.

THIS GIVES 2 SHOTS AT IT.

- 2.- \*MISS CAN COME ANYWHERE BEFORE THE \*CARDS CARD.
- 3.- \*HIGH AND/OR \*LOW MUST BE AFTER THE \*FMT CARDS, BECAUSE THE FORMAT IS USED TO READ THEM.

-----

### VALIDITY CHECKING

CONSIDER THE CHARACTERS 12.73 READ IN 5A1 FORMAT FROM A CARD. WE WISH TO BE SURE THEY ARE A VALID NUMBER. ASSUME THEY ARE IN AREA C, DIMENSIONED 5, AND WE ALSO HAVE THE FOLLOWING TABLE LTAB, DIMENSIONED 7 BY 9.

+ OP -	200 007 000
BLANK	125 056 700
DIGIT	443 408 883
DECIMAL POINT	990 300 000
E	006 660 000
OTHER	000 000 000
RESULT	124 342 252

WE USE THE FOLLOWING FORTRANISH CODE...

```

      LCOL = 1
      DO 50 I = 1, 5
      CALL PSNVAL ( C(I), LROW )
C      PSNVAL SETS LROW TO 1 IF C(I) IS A + OR A -, TO 2 IF ITS
C      A BLANK, 3 IF ITS A DIGIT, 4 IF A DECIMAL POINT,
C      5 IF AN E, AND 6 IF OTHER.
      LCOL = LTAB ( LROW, LCOL )
      IF ( LCOL ) ERROR, ERROR, 50
50 CONTINUE
C      WE ARE DONE. NOW ALL DEPENDS ON WHAT COLUMN WE
C      FINISHED WITH.
      LTYPE = LTAB ( 7, LCOL )
      GO TO ( BLANKS, ERROR, INTEGER, F TYPE, E TYPE ), LTYPE

```

THIS PROCEDURE CHECKS VALID CHARACTERS IN 80A1 FORM BEFORE THEY ARE CONVERTED. OF COURSE, IT TAKES TIME TO DO THIS. STRAIGHT-FORWARD READING IN ONE TEST TOOK 32 SECCNDS. THIS PROGRAM WITH CHECKING TOOK 56 SECCNDS.

IT SHOULD BE NOTED THAT PSNVAL ( ON THE 7094 ) WAS A 5 INSTRUCTION FAP PROGRAM ( ZAC, LDO\*, CAQ, SLW\*, TRA). WHEN THIS ACTIVITY WAS IN CRUDE FORTRAN, THE TEST RUN TOOK 90 SECONDS.

NOTE THAT ALL THE FOLLOWING FAIL...

```

12..7
12.K2
++723
127.2
1 .73

```

19. DATA MODIFICATION AND GENERATION USING FOR/IF/SET/SETX

FOR, IF, SET, SETX STATEMENTS.

THESE STATEMENTS PERMIT ONE TO...

- GENERATE NEW VARIABLES (I.E., COLUMNS) IN THE FILE.
- MODIFY EXISTING VARIABLES.
- GENERATE AND/OR MODIFY VARIABLES, CONTINGENT UPON THE RELATIONSHIPS OF VALUES IN THE ROW.
- DELETE ROWS, QUIT THE OPERATION, OR QUIT THE ENTIRE RUN, ALSO CONTINGENT UPON THE DATA IN A GIVEN ROW.
- WEIGHT ROWS (USE CERTAIN ROWS MORE THAN OTHERS).

ALL THIS TAKES ADVANTAGE OF THE FACT THAT EACH ROW IS PROCESSED SEPARATELY.

AN IF STATEMENT MUST BEGIN WITH FOR, IF, SET, OR SETX. THERE CAN BE UP TO THREE MAIN SECTIONS.

1. 'FOR' DEFINES A SERIES OF VARIABLES, EACH OF WHICH IS USED, IN TURN, BY THE REST OF THE STATEMENT. IT IS SOMEWHAT LIKE A DO LOOP. IF A 'FOR' LIST IS USED, IT MUST BE THE FIRST THING IN THE STATEMENT.
2. A LOGICAL TEST (OR A SERIES OF TESTS) YIELDS A RESULT FOR EACH ROW WHICH IS EITHER TRUE, FALSE, OR MISSING ( IF SOME OF THE TESTED DATA IS MISSING ). THIS PART BEGINS WITH 'IF'. THE 'IF' SECTION CAN FOLLOW A 'FOR' SECTION, IT CAN BE THE FIRST SECTION ( WHEN 'FOR' IS NOT USED ), OR IT CAN BE OMITTED.
3. A CONSEQUENCE SECTION IS INDICATED BY 'SET', 'SETX', 'DELETE', 'RETAIN', 'QUITFILE', 'QUITOP', AND 'QUITRUN'. 'FOR' AND/OR 'IF' ARE OPTIONAL, BUT THERE MUST ALWAYS BE A CONSEQUENT.

THUS, THE FOLLOWING FOUR SEQUENCES ARE POSSIBLE.....

1. FOR IF CONSEQUENCE(S)
2. FOR CONSEQUENCE(S)
3. IF CONSEQUENCE(S)
4. CONSEQUENCE(S)

CONSIDER A FILE NAMED A, 8 ROWS BY 5 COLUMNS, FOR REFERENCE IN THE EXAMPLES BELOW. (MISSING DATA IS PRINTED AS X )

POSITION		1	2	3	4	5
	LABEL	AGE	SEX	TESTAA	TESTBB	TESTCC
1	L1	14.	1.	82.	87.	X
2	L2	16.	2.	93.	100.	88.
3	L3	17.	X	77.	73.	68.
4	L4	16.	1.	X	88.	77.
5	L5	15.	1.	90.	93.	91.
6	L6	14.	2.	82.	100.	90.
7	L7	X	2.	83.	92.	80.
8	L8	16.	2.	X	X	X

\*\*\*\*\*  
\* FOR \*  
\*\*\*\*\*

'FOR', IF IT IS USED, MUST BEGIN THE PHRASE. 'FOR' MUST BE FOLLOWED, IN PARENTHESES, BY THE VARIABLES TO BE USED WHEREVER .X. APPEARS. THE LIST IS JUST LIKE A COLUMN SELECTION LIST, IN OTHER WORDS, (AGE-5) MEANS ALL VARIABLES FROM THE ONE LABELLED AGE THROUGH THE FIFTH VARIABLE.

THE FOLLOWING STATEMENT TRANSFORMS THE NON-MISSING SCORES IN VARIABLES TESTAA, TESTBB, AND TESTCC INTO THEIR SQUARE ROOTS.

```
IN = A ( SET TESTAA TO SQRT ( TESTAA), SET TESTBB TO
        SQRT ( TESTBB ), SET TESTCC TO SQRT ( TESTCC ) )
```

THE FOLLOWING DOES THE SAME THING USING A 'FOR' LOOP.

```
IN = A ( FOR ( TESTAA - TESTCC ), SET .X. TO SQRT ( .X. ) )
```

NOTE...A COMPLETE P-STAT STATEMENT MIGHT BE...

```
INTMDS, IN=A (FOR (TESTAA-TESTCC), SET .X. TO SQRT(.X.) ), COR = AC $
```

\*\*\*\*\*  
\* IF \*  
\*\*\*\*\*

'IF' IS FOLLOWED BY A LOGICAL RELATIONSHIP.  
A LOGICAL RELATIONSHIP BEGINS WITH ONE OF THE FOLLOWING SIX THINGS....

1. A SUBSTITUTION, REFERENCING THE CURRENT VARIABLE IN THE 'FOR' LIST. THIS HAS THE FORM .X. OR .XJ., WHERE J IS AN INTEGER, WITH OR WITHOUT A SIGN. FOR EXAMPLE,  
.X. .X+3. .X-5. .X5.
2. A VARIABLE, SUCH AS AGE.
3. A CONSTANT, SUCH AS 3.1416, OR 7 .
4. A POSITION, SUCH AS .C7. .

LATER IN THIS DESCRIPTION, THE TERM SVCP IS USED TO INDICATE THAT A SUBSTITUTION (S), A VARIABLE LABEL (V), A CONSTANT (C), OR A POSITION (P) CAN BE USED. SVP INDICATES, FOR EXAMPLE, THAT A CONSTANT ( NOTE SVP LACKS THE C ) IS NOT ALLOWED THERE. THESE ABBREVIATIONS OCCUR IN SOME OF THE P-STAT DIAGNOSTIC MESSAGES.

5. ANY (LIST)
6. ALL (LIST)

'ANY' OR 'ALL' ARE GENERALLY NOT USED WHEN 'FOR' IS USED.

THE LIST, IN 'ANY' AND 'ALL', IS JUST LIKE THE 'FOR' LIST. ROW AND COLUMN SELECTION LISTS START WITH AN 'R' OR A 'C', THESE DO NOT.

A LOGICAL OPERATOR MUST COME NEXT. THERE ARE 10 OF THESE...

OPERATOR	MEANING	FOLLOWED BY
1. GT	GREATER THAN	SVCP
2. GE	GREATER THAN OR EQUAL TO	SVCP
3. EQ	EQUAL TO	SVCP
4. NE	NOT EQUAL TO	SVCP
5. LE	LESS THAN OR EQUAL TO	SVCP
6. LT	LESS THAN	SVCP
7. MISSING		NOTHING
8. GOOD	IN OTHER WORDS, NOT MISSING	NOTHING
9. OUTRANGE		( SVCP, SVCP )
10. INRANGE		( SVCP, SVCP )

THE FIRST 6 MUST BE FOLLOWED BY A SVCP (SUBSTITUTION, VARIABLE, CONSTANT, OR POSITION). MISSING AND GOOD DO NOT NEED ANYTHING ELSE. OUTRANGE AND INRANGE ARE FOLLOWED BY (SVCP,SVCP), AS...

( IF AGE INRANGE ( 11, 20 ), RETAIN )

\*\*\*\*\*  
\* OP, AND \*  
\*\*\*\*\*

'OR' AND 'AND' CAN BE USED TO LINK SEVERAL LOGICAL RELATIONSHIPS.

(IF TESTAA GE 70 OR TESTBB GE 75 OR TESTCC GE 80 SET...)

(IF AGE GE 14 AND SEX EQ 1 SFT...)

(IF AGE EQ 14 AND SEX EQ 1 OR TESTAA EQ 77 OR TESTBB EQ 92, DELETE)

THIS LAST EXAMPLE WILL BE TRUE FOR ROW 1 ( AGE EQ 14 AND SEX EQ 1 ),  
ROW 3 ( TESTAA EQ 77 ) AND ROW 7 ( TESTBB EQ 92 ).

NOTE... 'OR' AND 'AND' CAN BE USED IN THE SAME STATEMENT. IF BOTH ARE  
USED, THE 'ANDS' ARE DONE AS IF PARENTHESES WERE AS FOLLOWS...

A OR ( B AND C ) OR D

HOWEVER, USE OF PARENTHESES TO GROUP 'OR' AND 'AND' SECTIONS IS  
NOT ALLOWED.

WHEN OR/AND/ANY/ALL ARE BEING USED, THE ONLY SENSIBLE  
CONSEQUENCE TESTING SHOULD BE TRUE VERSUS MISSING/FALSE.

CONSIDER.....

( IF AGE EQ 14 AND SEX EQ 1, T.SET TESTAA TO 1, M.SET TESTAA TO 2,  
F.SET TESTAA TO 3 )

ROW 3 WILL BE FALSE FOR THE ABOVE STATEMENT BECAUSE ITS SCORE ON  
AGE IS 17 AND NOT 14. HOWEVER...

( IF SEX EQ 1 AND AGE EQ 14, ETC. )  
WOULD GIVE MISSING FOR ROW 3, BECAUSE ITS SCORE ON SEX IS MISSING.

THE PROGRAM IS WRITTEN SO THAT THE BALANCE OF A SERIES OF 'ANDS'  
IS SKIPPED AS SOON AS ANY PART IS NOT TRUE. THUS, IF THE FIRST PART  
IS FALSE AND THE SECOND PART IS MISSING, SAYING THAT THE 'AND'  
SERIES IS NOT TRUE IS VALID, SAYING THAT IT IS FALSE RATHER  
THAN MISSING IS SOMEWHAT ARBITRARY.

\*\*\*\*\*  
\* CONSEQUENCES \*  
\*\*\*\*\*

AT LEAST ONE OF THE FOLLOWING MUST APPEAR.....

- SET
- SETX
- QUITFILE
- QUITOP
- QUITRUN
- DELETF
- RETAIN

THESE CAN BE IMMEDIATELY PRECEDED BY SOME PART OF FMT. FOR EXAMPLE..

- SFT
- T.SET ( SET AND T.SET ARE EQUIVALENT )
- F.SET
- M.SET
- FM.SET
- MF.SET
- FMT.SET, ETC.

NOTE - FMT.SET IS A POSSIBILITY. CONSIDER...

IN = W ( SETX GOOD.N TO 0, SETX TOTAL.N TO 0 )  
 ( FOR (AA-BB ) IF .X. GOOD, SET GOOD.N TO GOOD.N + 1,  
 FMT.SET TOTAL.N TO TOTAL.N + 1 )  
 ( SETX PCT.GOOD TO GOOD.N / TOTAL.N )  
 ( SET PCT.GOOD TO PCT.GOOD \* 100 )

\*\*\*\*\*  
\* QUITFILE, DELETF, ETC. \*  
\*\*\*\*\*

- QUITFILE WHEN EXECUTED, THE PROGRAM IS TOLD THAT THE FILE IS DONE.
- QUITOP WHEN EXECUTED, THE CURRENT P-STAT STEP ENDS AND THE RUN CONTINUES WITH THE NEXT P-STAT STEP.
- QUITRUN WHEN EXECUTED, THAT FINISHES THE RUN.
- DELETF WHEN EXECUTED, THE ROW BEING WORKED ON IS DELETED, AND ANOTHER ROW IS READ AND THE CHECKING AND SO FORTH STARTS ALL OVER.
- RETAIN IF 'RETAIN' IS THERE AND IS NOT EXECUTED, THE ROW IS DELETED. 'RETAIN' IS THE SAME AS 'FM.DELETF'.

\*\*\*\*\*  
\* SETX \*  
\*\*\*\*\*

SETX IS USED (INSTEAD OF SPT) TO INDICATE THAT A NEW VARIABLE IS BEING GENERATED. IT SHOULD APPEAR ONLY ONCE FOR EACH NEW VARIABLE.

THREE DIFFERENT WAYS OF DEFINING THE LABEL ( OR LABELS ) OF THE NEW VARIABLE(S) CAN FOLLOW THE 'SETX'. THESE ARE...

1. A DUMMY REFERENCE, '\*', WHICH TELLS P-STAT TO GENERATE A LABEL FOR THE VARIABLE. IF THE NEW VARIABLE WILL GO IN AS THE TWELFTH COLUMN, THE GENERATED LABEL WILL BE 'L12'. IN EFFECT, THE \* SAYS... MAKE A LABEL AND PUT THIS NEW VARIABLE IN A POSITION ONE TO THE RIGHT OF THE CURRENTLY RIGHTMOST COLUMN.

( FOR ( 3-5 ), SETX \* TO LOG10 ( .X. ) )

2. THE ACTUAL LABEL FOR THE NEW VARIABLE.

( SETX TOTSCORE TO SUM ( TESTAA, TESTBB, TESTCC ) )

3. A PAIR OF PARENTHESES WHICH ENCLOSE A PREFIX OR SUFFIX FOR LABEL GENERATION. THIS CAN ONLY BE USED WHEN 'FOR' IS IN USE. THE NEW LABEL IS CREATED USING THE LABEL OF THE CURRENT 'FOR' VARIABLE, WITH THE PREFIX OR SUFFIX ADDED TO IT.

( ABC\* ) WILL BE A PREFIX.  
( \*ABC ) WILL BE A SUFFIX.

HERE, THE \* SHOWS WHERE THE OLD LABEL GOES RELATIVE TO THE PREFIX OR SUFFIX.

( FOR ( 3-5 ), SETX (\*.L ) TO LOG10 ( .X. ) )

SINCE VARIABLE 3 IS TESTAA, THE NEW LABEL IN POSITION SIX WILL BE 'TESTAA.L'. THE SEVENTH VARIABLE WILL BE 'TESTBB.L', AND SO ON.

NOTE - THE NEWLY CREATED LABEL MUST START WITH A LETTER, ETC., AND SHOULD NOT DUPLICATE ANY COLUMN LABEL CURRENTLY IN THE FILE.

\*\*\*\*\*  
\* SET \*  
\*\*\*\*\*

FOUR TYPES OF ARGUMENTS CAN COME AFTER 'SET' .....

- 1 SUBSTITUTION .X.
- 2 VARIABLE AGE
- 3 POSITION .C10.
- 4 .W. THIS INDICATES WEIGHTING

NOTE THAT A CONSTANT IS NOT ALLOWED. YOU CANNOT SET 7 TO ANYTHING.

THE .W. PERMITS WEIGHTING OF SOME OF THE ROWS IN A FILE.

IN = A ( IF AGE GE 17 SET .W. TO 2 )

THIS WOULD CAUSE ROW 3 IN THE EXAMPLE FILE A TO BE SENT TWICE TO THE CURRENT P-STAT OPERATION, PROVIDING, OF COURSE, THE ROW GETS SENT AT ALL. ONE MIGHT WEIGHT IT FOR ONE REASON AND THEN DELETE IT FOR ANOTHER REASON. ONLY ONE ACTIVE FILE AT A TIME CAN BE USING SET STATEMENTS WITH .W. IN THEM. THERE MAY BE .W. USAGES IN EACH OF SEVERAL 'IF' PHRASES MODIFYING A FILE, AND PERHAPS SEVERAL .W. USAGES IN A SINGLE 'IF' PHRASE. IN THESE EVENTS, THE LARGEST SETTING OF .W. IS USED AS THE WEIGHTING FACTOR FOR THE ROW BEING WORKED ON.

THE WORD 'TO' MUST FOLLOW THE ARGUMENT ( WHICH FOLLOWED 'SET' OR 'SETX' ) . THE FOLLOWING CAN COME AFTER THE 'TO' .....

\*\*\*\*\*  
\* 7 SIMPLE SETTINGS \*  
\*\*\*\*\*

- 1. .X.,
- 2. AGE,
- 3. 7,
- 4. .C10.,
- 5. .IN., THE POSITION OF THIS ROW IN THE INPUT FILE BEFORE ANY DELETING OR WEIGHTING.
- 6. .USED., THE NUMBER OF ROWS WHICH HAVE ALREADY SURVIVED ALL THIS, PLUS 1. SINCE WEIGHTING REPEATS A ROW EXACTLY, REPEATING A ROW DOES NOT INCREASE THE .USED. COUNTER. IF WEIGHTING IS NOT INVOLVED, .USED. WILL PROVIDE SEQUENCE NUMBERING OF NONDELETED ROWS. THEREFORE, REPEATS IN THE SERIALIZATION INDICATE WEIGHTING.

7. .M., THIS GENERATES A MISSING SCORE. .M. IS USED BECAUSE M, BY ITSELF, MIGHT SOMEDAY BE A LABEL IN SOMEONE'S FILE.

```

*****
*
*           ABS, LOG, SQRT, ETC.
*
*****

```

- 1. ABS (SVCP) ABSOLUTE VALUE
- 2. LOG (SVCP) NATURAL LOG (BASE E)
- 3. SQRT (SVCP) SQUARE ROOT
- 4. LOG10 (SVCP) COMMON LOG (BASE 10)
- 5. EXP (SVCP) E \*\* SVCP
- 6. SIN (SVCP) SINE
- 7. COS (SVCP) COSTINE
- 8. ATAN (SVCP) ARCTANGENT

```

*****
*
*           NCOT AND RECODE
*
*****

```

1. NCOT( SVP, C,...,C). NCOT IS A GENERAL FORM OF DICOTIMIZATION. FOR EXAMPLE, NCOT(AGE, 15). THIS WOULD GIVE 1 IF A SCORE ON AGE WERE 15 OR IESS, AND 2 IF MORE THAN 15. NCOT(AGE,14, 16). IF 14 OR LESS, SET TO 1. IF MORE THAN 14 AND NOT MORE THAN 16, SET TO 2. IF OVER 16, SET TO 3. ANY NUMBER OF CONSTANTS IS ALLOWED, BUT THEY MUST ASCEND. THE NCOT RESULTS USUALLY START AT 1 AND INCREASE BY 1. IF SOME OTHER STARTING VALUE IS PREFERRED, IT CAN BE INSERTED AS THE FIRST THING IN THE PARENTHESES...

NCOT(0,AGE,14) GENERATES 0 AND 1, INSTEAD OF 1 AND 2 (AND OF COURSE MISSING IF AGE IS MISSING).

A TYPICAL USE MIGHT BE...

IN = A( SETX CODE.AGE TO NCOT(AGE, 14) )

NOTE ... NCOT ( AGE, 15 ) COULD BE THOUGHT OF AS ...  
 NCOT ( AGE, UP THROUGH 15, OVER 15 ) .

2. RECODE ( SVP, 11/13 = 1, 14 = 2, X = 3, M = 4 )

THE SVP IS A SUBSTITUTION OR VARIABLE OR POSITION.  
THREE THINGS CAN FOLLOW...

1 1/5=1 SCORES OF 1 THROUGH 5 ARE RECODED TO 1.  
6=2 SIXES BECOME TWOS  
7=1 SEVENS BECOME ONES  
8/10, 12, 14 = 3 EIGHTS THROUGH TENS, ALSO TWELVES  
AND FOURTEENS BECOME THREES.

2 M = 3 THIS CONVERTS A MISSING SCORE INTO A THREE.  
M = M THIS WORKS, BUT IS UNNECESSARY. IF NO M= IS  
FOUND, A MISSING SCORE SIMPLY REMAINS MISSING.  
ONLY ONE M= SHOULD BE USED ( IF ANY ).

3 X = 4 ALL NON-MISSING SCORES NOT OTHERWISE  
DEFINFD ARE CONSIDERED TO BE EXTRA.  
X=4 SAYS CHANGE THEM ALL INTO 4S.  
ONLY ONE X= CAN BE USED IN A RECODE.  
  
IF X= IS NOT USED, NON-REFERENCED SCORES STAY  
AS IS, UNLESS THE SCORE IS MISSING AND  
THEREFORE UNDER THE CONTROL OF M= .

X = M THIS WILL PROBABLY BE USED FREQUENTLY. IT  
RECODES AN EXTRA SCORE TO MISSING.

THE FOLLOWING TWO STATEMENTS DO THE SAME THING ( ASSUMING  
POSITIVE SCORES ON AGE ) ...

NCOT ( AGE, 14 )

RECODE ( AGE, 0 / 14 = 1, X = 2 )

\*\*\*\*\*  
\* \* \* \* \*  
\* 10 OPERATIONS ON LISTS OF VARIABLES \*  
\* \* \* \* \*  
\*\*\*\*\*

- 1. SUM (LIST) THIS ADDS UP THE VALUES, IN THIS ROW, ON THE  
VARIABLES IN THE LIST.  
SUM(1-100) ADDS THE FIRST 100 VARIABLES.  
IF ANY ARE MISSING, THE SUM IS MISSING.  
THESE LISTS HAVE THE SAME RULES AS 'POP' LISTS.
- 2. SUM.A (LIST) THIS ADDS THE NON-MISSING SCORES.  
THE LETTER A IMPLIES USAGE OF AVAILABLE DATA.
- 3. MEAN (LIST) THE MEAN SCORE

- 4. MEAN.A (LIST)
- 5. MAX (LIST) THE LARGEST VALUE
- 6. MAX.A (LIST)
- 7. MIN (LIST) THE SMALLEST VALUE
- 8. MIN.A (LIST)
- 9. SDEV (LIST) THE STANDARD DEVIATION
- 10. SDEV.A (LIST)

```
*****
*                               *
*   SIMPLE ARITHMETIC IS POSSIBLE   *
*                               *
*****
```

- 1. SVCP + SVCP
- 2. SVCP - SVCP
- 3. SVCP \* SVCP
- 4. SVCP / SVCP
- 5. SVCP \*\*SVCP

```
.....
.   EXAMPLES   .
.....
```

CONSIDER.....

IN = A (IF AGE GE 15, DELETE ),

A P-STAT SYSTEM ROUTINE, AS EACH ROW IS READ, LOOKS FIRST AT THE SCORE IN THAT ROW ON THE VARIABLE LABELED AGE. IF THAT SCORE IS GREATER THAN OR EQUAL TO (GE) 15, THE RELATIONSHIP IS TRUE, AND IS TRUE FOR ROWS 2, 3, 4, 5, AND 8 OF THE ABOVE EXAMPLE FILE. THEREFORE, THOSE ROWS WOULD BE DELETED. THE COMPLETE OPERATION MIGHT BE ...

```
SCAN, IN = A ( IF AGE GE 15, DELETE), DES = A15.D,
OUT = A15 $
```

ASSUME THE SCORES ON VARIABLE TESTCC ARE ALL TEN 000 HIGH. THEN

IN = A (SET TESTCC TO TESTCC - 10 )

WOULD SUBTRACT 10 FROM EACH NON-MISSING SCORE IN VARIABLE 'TESTCC'.

IN = A (SETX MAXTEST TO MAX ( TESTAA - TESTCC ))

SETX (SET AN EXTRA VARIABLE) GENERATES A NEW VARIABLE, MAXTEST, WHICH IS, IN THIS EXAMPLE, THE MAXIMUM OF THE SCORES ON TESTAA THROUGH TESTCC. EACH ROW OF THE FILE, AS IT IS READ FROM WHEREVER IT IS (DISK, TAPE, ETC), WOULD HAVE THE ABOVE EVALUATION DONE UPON ITS SCORES ON TESTAA, TESTBB, AND TESTCC.

NOTE... THE SYMBOL '-' INDICATES MINUS IN ARITHMETIC SITUATIONS, AND INDICATES A RANGE IN LISTS.

PUTTING T. BEFORE A CONSEQUENCE MEANS...EXECUTE THE CONSEQUENCE ONLY IF THE RESULT OF THE 'IF' SECTION IS TRUE. F. STANDS FOR FALSE, M. FOR MISSING, TM. FOR EITHER TRUE OR MISSING, ETC. AS SHOWN ABOVE, A CONSEQUENCE WITH NO SUCH LETTERS PRECEDING IT IS EXECUTED IF TRUE. IN OTHER WORDS, 'T.DELETE' AND 'DELETE' ARE EQUIVALENT.

IN = A ( IF AGE GE 15, T.SETX NEWAGE TO 1, M.SET NEWAGE TO 2, P.SET NEWAGE TO 3 )

HERE, A NEW VARIABLE NAMED NEWAGE IS BEING GENERATED. ROWS 2, 3, 4, 5, AND 8 WILL HAVE A SCORE OF 1 ON IT BECAUSE, FOR THOSE ROWS, THE (IF AGE GE 15) RELATIONSHIP IS TRUE. ROW 7 WILL BE 2, AND ROWS 1 AND 6 WILL BE 3 ON THE NEW VARIABLE.

NOTE- WHEN A NEW VARIABLE IS BEING DEFINED, 'SETX' MUST BE USED. HOWEVER, ONE SUCH 'SETX' DEFINES THE VARIABLE. SUBSEQUENT REFERENCES SHOULD USE 'SET' .

IF A STATEMENT HAS 'SETX' IN IT, THAT VARIABLE IS SET TO MISSING BEFORE THE STATEMENT BEGINS EXECUTION. THUS, IF A LOGICAL RELATIONSHIP CAUSES THE EXECUTION OF THE 'SETX' PART TO BE BYPASSED, THE NEW VARIABLE IS THERE FOR THAT ROW, AND ITS VALUE IS MISSING.

IN = A ( SETX \* TO .C4. + .C5. )

THIS GENERATES A NEW VARIABLE, WHICH WILL BE THE SIXTH COLUMN OF THE FILE BECAUSE FILE 'A', IN THIS EXAMPLE, ALREADY HAS FIVE COLUMNS. THE NEW VARIABLE, LABELLED 'L6', WILL BE THE SUM OF COLUMNS 4 AND 5. HOWEVER, IF THE VALUE IN COLUMN 4 OR COLUMN 5 IS MISSING, THE NEW VARIABLE, IN COLUMN 6, WILL ALSO BE SET TO MISSING.

IN = A ( FOR (3-5) SET .X. TO SQRT(.X.) )

HERE, FOR (3-5) DEFINES A LOOP, INVOLVING THE SCORES OF VARIABLES (I.E., COLUMNS) 3, 4, AND 5. THE REST OF THE STATEMENT IS EXECUTED 3 TIMES, ONCE WITH THE SCORE ON VARIABLE 3 INSERTED WHEREVER .X. IS FOUND, THEN WITH THE SCORE ON VARIABLE 4, AND FINALLY WITH THE SCORE ON VARIABLE 5 USED. IN THIS EXAMPLE THE RESULTING VARIABLES 3-5 WOULD BE THE SQUARE ROOTS OF THE ORIGINAL SCORES ON 3-5.

IN = A ( FOR ( 3-5 ), SETX \* TO SORT ( .X. ) )

HERE, NEW VARIABLES 6-8 ARE CREATED. VARIABLE 6 IS THE SQUARE ROOT OF VARIABLE 3, ETC. FILE 'A' HAS 5 COLUMNS, THEREFORE THE 'SETX', WHEN THE 'FOR' INVOLVES THREE VARIABLES, WILL GENERATE VARIABLES 6, 7 AND 8. THEY WILL HAVE COLUMN LABELS 'L6', 'L7', AND 'L8'.

IN = A ( FOR(3-5), SETX ( \*.R ) TO SQRT( .X. ) )

THE ABOVE SUPPLIES BETTER LABELS FOR THE 3 NEW VARIABLES. THE '(\*.P)' INDICATES THAT '.R' SHOULD BE ADDED TO THE ASSOCIATED .X. VARIABLE'S LABEL AS A SUFFIX. THUS TESTAA.R, TESTBB.R, AND TESTCC.R WILL BE PRODUCED. SOME EXAMPLES WILL DESCRIBE THIS FEATURE. ASSUME THAT 'AGE' IS THE LABEL OF THE .X. VARIABLE.

(*SORT)	GENERATES	AGE.SQRT
(SORT.*)	GENERATES	SQRT.AGE
(*3)	GENERATES	AGE3
(*...X)	GENERATES	AGE...X
(3*)	FAILS	3AGE IS NOT A VALID LABEL
(ABCDEF*)	GENERATES	ABCDEFGE (NOTE THE LOSS OF 'A' IN 'AGE')

NOTE- THE WORD COLUMN REFERS TO A POSITION IN A FILE. IT HAS NOTHING TO DO WITH THE IDEA OF A COLUMN IN A PUNCHED CARD.

IN = A ( FOR( 3-4), SETX NEWVAR TO SQRT ( .X. ) )

THE ABOVE IS WRONG. ONE CANNOT CREATE TWO NEW VARIABLES, EACH LABELLED 'NEWVAR'. FOR THAT MATTER, A NEWLY DEFINED LABEL MUST BE DIFFERENT FROM ANY LABEL, ORIGINAL OR ALSO NEW, THAT IS CURRENTLY IN THE FILE.

IN = A ( FOR (4-5) SET .X-1. TO .X. - 1 )

FOR ROW 2, FOR EXAMPLE, TESTAA BECOMES 99 AND TESTBB BECOMES 87. .X-1. REFERS TO THE SCORE ON THE VARIABLE BEFORE THE CURRENT 'FOR' VARIABLE. .X. - 1 REFERS TO THE SCORE MINUS 1 ON THE CURRENT 'FOR' VARIABLE.

IN = A ( FOR (1-5) SETX \* TO LOG ( .X. ) )

THIS GENERATES 5 NEW VARIABLES IN POSITIONS 6 TO 10 ( LABELED L6, L7, ETC.). VARIABLE 6 WILL BE THE LOG OF VARIABLE 1, 7 OF 2, ETC.

IN = A ( FOR(4-5) IF .X. LT .X-1., PM.DELETE )

THIS RETAINS ONLY THOSE ROWS FOR WHICH THE SCORES ON VARIABLES 3, 4, AND 5 ARE DECREASING ( I.E., 4 LT 3 AND 5 LT 4 ). ROW 3 WOULD BE THE ONLY ROW RETAINED. NOTE - USE OF 'RETAIN' INSTEAD OF 'PM.DELETE' WOULD GIVE THE SAME RESULT.

IN = A ( FOR ( 1+ ) SET .X. TO SQRT ( .X. ) )

THIS SETS ALL THE VALUES IN FILE A TO THEIR SQUARE ROOTS.

IN = A (IF ANY (3-5) MISSING, DELETED)

HERE, IF A ROWS SCORE ON ANY OF VARIABLES 3, 4, OR 5 IS MISSING, THE ROW IS DELETED. ANOTHER WAY OF DOING THIS IS...

IN = A ( POP (3-5) IF .X. MISSING, DELETE)

ROWS 2, 3, 5-7 WOULD SURVIVE. THE REST HAVE AT LEAST ONE MISSING SCORE ON VARIABLE 3, 4, OR 5.

NOTE- FILE 'A' ITSELF IS NOT AFFECTED BY DELETING ROWS. THE CURRENT OPERATION SIMPLY GETS A READING OF 'A' WITHOUT CERTAIN OF ITS ROWS.

IN = A (SETX COUNT TO C.) ( POP (TESTAA - TESTCC) IF .X. GOOD SET COUNT TO COUNT + 1)

THIS GENERATES A NEW VARIABLE, COUNT, WHICH WILL HAVE THE NUMBER OF GOOD (I. E., NON-MISSING) SCORES ON VARIABLES 3-5. THE VALUES FOR COUNT COULD THEREFORE RANGE FROM ZERO TO THREE.

IN = A ( SETX UNIT TO 1 )

THIS GENERATES A VECTOR OF ONES AS THE SIXTH COLUMN OF THE FILE.

INCIDENTALLY, WORDS AND NUMBERS ( OR WORDS AND WORDS ) SHOULD NOT BE RUN TOGETHER. IN PARTICULAR, COLUMN 80 ON ONE CARD AND COLUMN ONE ON THE NEXT CARD ARE TREATED AS IF THEY RAN TOGETHER.

IN = A ( SETXUNIT TO 1 )

THIS CAUSES ALL KINDS OF PROBLEMS. ALL OF THE CHARACTERS IN 'SETXUNIT' ARE LEGAL IN A NAME. THEREFORE, ALL EIGHT CHARACTERS WILL BE TREATED AS A SINGLE NAME. THE USER SHOULD USE BLANKS FREELY, EXCEPT WITHIN A NAME OR A NUMBER. 'SE TX' IS A LOSER. CONSTANTS CAN BE, FOR EXAMPLE, 7 OR 7. OR 7.0, BUT '1.357' SHOULD NOT BE '1.3 57'.

COMMAS BETWEEN SECTIONS, THAT IS, AFTER THE 'POP' PART, OR AFTER THE 'IF' PART, OR AFTER EACH CONSEQUENCE, ARE OPTIONAL.

```

.....
. COLUMN / IF INTERACTION .
.....

```

THE USER MUST BE CAREFUL HERE.

AS NOTED ABOVE, THE OUTPUT OF ONE PHRASE IS THE INPUT TO THE NEXT.

IN = A(C 1-3) (IF TESTCC LE 79, TM.DELETE) ... WOULD BE AN ERROR, BECAUSE 'TESTCC' IS VARIABLE 5 AND THE (C 1-3) DISCARDED ALL COLUMNS EXCEPT COLUMNS 1-3. THEREFORE, WHEN THE 'IF' STATEMENT WAS EXECUTED, THERE WAS NO LONGER ANY VARIABLE WITH THE LABEL 'TESTCC'.

IN = A( IF TESTCC LE 79, TM.DELETE ) (C 1-3) ... WORKS, BECAUSE THE TRUNCATION TO COLUMNS 1-3 OCCURRED AFTER THE SCORE ON TESTCC WAS EXAMINED.

```

.....
. ROW / IF INTERACTION .
.....

```

THE P-STAT SYSTEM HANDLES THIS PROBLEM INTERNALLY.

IN = A ( IF SEX EQ 1 RETAIN ) ( R L2+ )

IF THE 'IF' PHRASE WERE EXECUTED FIRST, THE FIRST ROW WOULD SURVIVE THE 'IF' AND BE REJECTED BY THE 'R' PHRASE. THEN ROW 2, LABELLED 'L2', WOULD BE REJECTED BY THE 'IF' PHRASE AND NEVER GET TO THE 'R' PHRASE. THUS THE 'R' PHRASE WOULD KEEP REJECTING ROWS 4 AND 5 BECAUSE IT HAS NOT YET SEEN ROW 'L2'.

THAT IS WHY THE SYSTEM AUTOMATICALLY DOES THE ROW PHRASE FIRST.

## 20. MACROS OF P-STAT OPERATIONS

A MACRO IS A SERIES OF P-STAT STATEMENTS THAT, TAKEN TOGETHER, MAKE UP A SUBROUTINE OF P-STAT STATEMENTS. IT IS GIVEN A NAME AND, LATER IN THE RUN, WHEN THE NAME IS USED AS A P-STAT OPERATION, THE ENTIRE SERIES IS EXECUTED.

FOR EXAMPLE, FILE A HAS SOME NUMBER OF ROWS AND 31 COLUMNS. THE FIRST 30 COLUMNS ARE SCORES ON A 30 ITEM TPST, THE LAST VARIABLE IS LABELLED CLASS. FRESHMEN HAVE SCORES OF 1 ON CLASS, SOPHOMORES ARE 2, JUNIORS ARE 3, AND SENIORS ARE 4. ONE MIGHT WISH TO DO A CORRELATION, FACTOR ANALYSIS, ROTATION, PRINT SEQUENCE ON THE FIRST 30 VARIABLES A NUMBER OF TIMES, ONCE FOR ALL ROWS OF A, AND ONCE FOR EACH CLASS. THIS CAN BE DONE BY DEFINING THE SEQUENCE ( THE MACRO ITSELF ) AND THEN CALLING IT FIVE TIMES.

(IT MAY BE HELPFUL TO READ THE DESCRIPTIONS OF THE OPERATIONS USED IN THESE EXAMPLE MACROS. THE TABLE OF CONTENTS AT THE END OF THIS MANUAL HAS THEIR PAGE NUMBERS.)

```
*****
*
*          SOME EXAMPLES
*
*****
```

THESE SIX LINES DEFINE THE MACRO...

```
MACRO, FAC = DATA/VPAC $
INTCDS, IN=DATA, COR = X $
ITFACT, IN = X, SF = C $
ROTATE, IN = C, VF = VPAC $
PRINT = 2 / X / VPAC $
ENDMACRO $
```

THESE ARE FIVE USES OF IT ...

```
FAC = A(C 1-30) / VPALL $
FAC = A( IF CLASS EQ 1, RETAIN ) ( C 1-30 ) / VF1 $
FAC = A( IF CLASS EQ 2, RETAIN ) ( C 1-30 ) / VF2 $
FAC = A( IF CLASS EQ 3, RETAIN ) ( C 1-30 ) / VF3 $
FAC = A( IF CLASS EQ 4, RETAIN ) ( C 1-30 ) / VF4 $
```

IN THIS MACRO DEFINITION, THERE WERE TWO DUMMY ARGUMENTS, DATA AND VPAC.

EACH CALL TO THE MACRO ALSO HAD TWO ARGUMENTS. WHEN A MACRO IS CALLED (THAT IS, USED), THE ORIGINAL DEFINITION IS RETRIEVED. THEN THE CURRENT ARGUMENTS - THOSE USED IN THE CALL - ARE SUBSTITUTED FOR THE DUMMY ARGUMENTS THAT WERE USED IN THE DEFINITION. IN THE FIRST USAGE ABOVE, 'A(C 1-30)' IS PLUGGED WHEREVER THE NAME 'DATA' WAS FOUND IN THE ORIGINAL MACRO ( I.E., IN THE 'INTCDS' STATEMENT ). 'VPALL' IS PLUGGED WHEREVER THE NAME 'VPAC' WAS FOUND ... IN THE ROTATE AND PRINT STATEMENTS.

ANOTHER EXAMPLE-- IF ONE WISHED TO MERGE 3 FILES SIDEWAYS (IGNORING THE FACT THAT 'MERGE', USING 'MIDDLE=', WOULD DO IT QUICKER) --

```

MACRO, MERGE3 = A/B/C/D$
MERGE, LEFT = A, RIGHT = B, OUT = X$
MERGE, LEFT = X, RIGHT = C, OUT = D$
ENDMACRO $

```

THE NAMES USED IN THIS MACRO DEFINITION ARE A, B, C, D, AND X. ALL EXCEPT X ARE DUMMY ARGUMENTS. X IS USED TO IDENTIFY A RESULT OF A STEP WHICH IS NEEDED AS INPUT FOR A LATER STEP, ALL WITHIN THE MACRO. IT IS A LOCAL NAME. WHEN THE MACRO IS USED, ACTUAL FILE NAMES WILL REPLACE A, B, C, AND D. THE LOCAL NAMES ARE CHANGED ALSO, TO SOMETHING LIKE QQQQ12. EACH TIME A MACRO IS USED, A NEW, HOPEFULLY UNIQUE NAME IS INVENTED FOR EACH DIFFERENT LOCAL NAME IN THE MACRO, STARTING WITH QQQQ1. THE FIRST EXAMPLE MACRO, PAC, HAS TWO LOCAL NAMES, X AND C. IN THE FIFTH USAGE, X WOULD HAVE APPEARED AS QQQQ9. THIS PERMITS A USER, WHEN DEFINING A MACRO, TO USE ANY NAMES (NOT STARTING WITH QQQQ) THAT HE WISHES, WITHOUT CONFLICTING WITH NAMES ACTUALLY IN ORDINARY USE.

```

*****
*
*   GENERAL RULES AND RESTRICTIONS
*
*
*****

```

SOME FACTS ABOUT MACROS .... VERSION 52

ARGUMENTS, WHEN CALLING A MACRO, CAN HAVE QUALIFICATION.

DATA CARDS CAN BE DEFINED WITHIN A MACRO AND USED EACH TIME THE MACRO IS USED.

20 DIFFERENT MACROS CAN BE DEFINED IN A RUN.

THE MACRO DEFINITIONS CAN TOTAL UP TO 2400 CHARACTERS (30 SOLIDLY PUNCHED CARDS). OBVIOUSLY, THIS IS ALSO THE LIMIT FOR ANY SINGLE MACRO.

IF A MACRO NAME IS THE SAME AS THE NAME OF AN EXISTING P-STAT OPERATION, SUBSEQUENT USE OF THE NAME WILL REFER TO THE MACRO. HOWEVER, A FEW P-STAT NAMES CANNOT BE USED FOR MACRO NAMES (PURGE, MAC.THRU, DO, ENDDO, MACRO, ENDMACRO, FND, MACDATA, MAC..D AND DUMP).

A MACRO CAN CALL ANOTHER MACRO ( UP TO 10 DEEP - ( UNLIKELY ) ).

MACROS CAN BE INSIDE A DO LOOP (SEE SECTION ON DO LOOPS).

A DO LOOP CAN BE INSIDE A MACRO.

A MACRO CAN HAVE NO ARGUMENTS, OR FROM 1 TO 20.

UP TO 30 DIFFERENT LOCAL NAMES CAN BE USED IN ANY MACRO.

WHEN EXPANDED, A MACRO CANNOT EXCEED 3000 CHARACTERS.

THINGS TO BE CHANGED-USUALLY JUST THE NUMBER OF TIMES ANY FILE NAME IS MENTIONED IN ALL DEFINED MACROS- CANNOT EXCEED 200.

ANY LOCAL NAMES GENERATED IN A MACRO USAGE ARE AUTOMATICALLY PURGED WHEN THAT MACRO IS FINISHED. ( THESE ARE THE QQQQ NAMES. )

```

*****
*
*          TYPES OF ARGUMENTS          *
*
*****

```

ANY NAME (IDENTIFIER, ARGUMENT, OR EVEN OPERATION NAME) IN THE MACRO DEFINITION CAN BE A DUMMY NAME. WHEN USED, IT CAN BE REPLACED BY A NAME, A NUMBER, OR A LIST. FOR EXAMPLE,

```

MACRO, CORFACT = A / B / C $
A, IN = B, COR = X$
ITFACT, IN = X, SF = C$
ENDMACRO $

```

USAGE MIGHT BE,

```

CORFACT = INTCDS / YA / XAA $
CORFACT = INTMDS / XB / XBB $

```

THUS THE TYPE OF CORRELATION, COMPLETE DATA OR MISSING DATA, IS AN OPTION WHENEVER THAT MACRO IS USED.

A LIST CAN BE USED, OR A NUMBER--

```

MACRO, CORFAC = A/B/C/D/E $
HEAD = D$
A, IN = B, COR = X $
ITFACT, IN = X, SF = C, NFAC = P $
ENDMACRO $

```

USAGE MIGHT BE

```

CORFAC = INTCDS / XA / XC / (FPB 68) / 5 $

```

IF A LIST, HOWEVER, HAS THE FORM (( LIST )), IT IS PLUGGED WITHOUT THE (( )). CONSIDER.....

```

MACRO, GLOP = A $
DO = A$
(OTHER STATEMENTS)
ENDDO $
ENDMACRO $

```

IN USE

```

GLOP = (( 3/7 )) $

```

```

*****
*
*   QUALIFICATION OF FILE NAMES, BOTH
*   WHEN DEFINED AND WHEN CALLED.
*
*****

```

A DUMMY NAME CAN HAVE QUALIFICATION INSIDE THE DEFINITION. THE USAGE ARGUMENT CAN ALSO BE QUALIFIED. THE USAGE ARGUMENT WITH ITS QUALIFICATION IS PLUGGED WHERE THE DUMMY NAME WAS, SO THAT THE USAGE QUALIFICATION IS EXECUTED BEFORE THE DEFINITION QUALIFICATION.

```

MACRO      MACRO, GLOP = A / B $
DEFINITION SCAN, IN = A ( C 3-20 ), DES = B $
           PRINT = 4 / B $
           ENDMACRO $

```

```

CALL      GLOP = X ( R 1-40 ) / XDES $

```

```

RESULTING  SCAN, IN = X ( R 1-40 ) ( C 3-20 ),
OPERATIONS   DES = XDES $
           PRINT = 4 / XDES $

```

```

*****
*
*   DATA CARDS WITHIN A MACRO
*
*****

```

MANY OPERATIONS REQUIRE DATA CARDS. FOR EXAMPLE,

```

MACRO, FACPLOT = A/D $
INTCDS, IN = A, COR = B $
ITFACT, IN = B, SP = C, NFAC=5 $
ROTATE, IN = C, VF = D $
PLOT = D $
ENDMACRO $

```

USAGE MIGHT BE...

```

FACPLOT = X ( C 21-40 ) / VFX $
CCCCC
*END

```

THE CCCCC CARD AND \*END CARD ARE DATA CARDS USED BY THE PLOT OPERATION. IT SHOULD BE NOTED THAT THEY OCCUR AFTER THE CALL OF THE MACRO.

IF, HOWEVER, THESE DATA CARDS WERE TO BE USED WITHOUT CHANGE EACH TIME THE MACRO WAS CALLED, THEY COULD BE INCLUDED IN THE MACRO DEFINITION.

```

MACRO, FACPLOT = A / D $
INTCDS, IN = A, COR = B $
ITFACT, IN = B, SF = C $
ROTATE, IN = C, VF = D $
PLOT = D $
MACDATA = 2 $
CCCCC
*END
ENDMACRO $

```

THE 'MACDATA = N \$' SAYS THAT N DATA CARDS FOLLOW. SEVERAL SETS OF DATA CAN BE INSERTED AT APPROPRIATE PLACES IN A MACRO.

```

*****
*
*           ((GLOBAL))  USAGE           *
*           IN THE MACRO DEFINITION.    *
*
*****

```

IT IS SOMETIMES NECESSARY TO REFERENCE A LABEL IN A MACRO DEFINITION, FOR EXAMPLE, 'PBYQ, ROW=AGE,...'. HOWEVER, THE MACRO WILL THINK THAT 'AGE' IS A LOCAL, TEMPORARY FILE NAME AND WILL GENERATE A 'QQQ7' TYPE NAME FOR IT. THIS IS AVOIDED BY ENCLOSING THE LABEL WITH (( )), AS, 'PBYQ, ROW=((AGE)),...'.

A FILE NAME, TO BE USED EACH TIME THE MACRO EXPANDS, SHOULD BE HANDLED THE SAME WAY.

```

*****
*
*           AN EXAMPLE USING           *
*           NESTED MACROS.             *
*
*****

```

ASSUME THAT DATA FILE 'X' HAS ABOUT 2,000 ROWS ( PATIENTS IN A DRUG TREATMENT EVALUATION ANALYSIS ) AND THE FOLLOWING COLUMNS.....

- 1            'MEDICATN', THE MEDICATION INVOLVED, 1=DRUG A, 2=DRUG B, 3=DRUG C, 4=DRUG D.
- 2            'AGE', FROM 21 TO 70.
- 3-42        40 PSYCHOLOGICAL VARIABLES, MEASURED JUST BEFORE TREATMENT BEGAN, IN OTHER WORDS, WEEK ZERO.
- 43-82       THE SAME 40 VARIABLES RE-MEASURED AFTER 2 WEEKS.
- 83-122      AGAIN, AFTER 4 WEEKS OF TREATMENT.
- 123-162     AFTER 8 WEEKS.
- 163-202     A FINAL EVALUATION, AFTER 12 WEEKS.

WE WISH TO SEE INTERCORRELATIONS OF EACH SET OF THE VARIABLES ( I.E., FIVE 40 BY 40 MATRICES ) FOR EACH OF THE FOUR TREATMENT DRUGS SEPARATELY, AND FOR AGES 21-30, 31-50, AND 50-70 SEPARATELY. THIS TOTALS 5 X 4 X 3

OR 60 PRINTOUTS, EACH A 40 BY 40 MATRIX, AND IT IS DESIRABLE THAT EACH HEADING INDICATE WHICH MATRIX IS WHICH.

THE FOLLOWING P-STAT STATEMENTS WOULD BE ONE WAY OF DOING ALL THIS. NOTE THE USE OF XHEAD AND SLASHES. THE PRINTOUT ORDER WILL BE... DRUG 1, AGE 21-30, EACH WEEK IN TURN, THEN DRUG 1 STILL, AGE 31-50, EACH WEEK, ETC.

HEAD = ( NOV 20, 1970. FILE X DATA. /// ) \$  
FIND = X \$

MACRO, LEVEL1 = H / P \$  
XHEAD = H, SLASH = 3 \$  
LEVEL2 = ( AGE 21-30 / ) / P( IF AGE INRANGE( 21,30 ), RETAIN ) \$  
LEVEL2 = ( AGE 31-50 / ) / P( IF AGE INRANGE( 31,50 ), RETAIN ) \$  
LEVEL2 = ( AGE 51-70 / ) / P( IF AGE INRANGE( 51,70 ), RETAIN ) \$  
ENDMACRO \$

MACRO, LEVEL2 = H / P \$  
XHEAD = H, SLASH = 2 \$  
LEVEL3 = ( WEEK 0 ) / P( C 1, 2, 3- 42 ) \$  
LEVEL3 = ( WEEK 2 ) / P( C 1, 2, 43- 82 ) \$  
LEVEL3 = ( WEEK 4 ) / P( C 1, 2, 83-122 ) \$  
LEVEL3 = ( WEEK 8 ) / P( C 1, 2, 123-162 ) \$  
LEVEL3 = ( WEEK 12 ) / P( C 1, 2, 163-202 ) \$  
ENDMACRO \$

MACRO, LEVEL3 = H / Y \$  
XHEAD = H \$  
INTMDS, IN = Y ( C 3+ ), COR = R \$  
BPRINT = R \$  
ENDMACRO \$

LEVEL1 = ( DRUG 1, A // ) / X( IF MEDICATN EQ 1, RETAIN ) \$  
LEVEL1 = ( DRUG 2, B // ) / X( IF MEDICATN EQ 2, RETAIN ) \$  
LEVEL1 = ( DRUG 3, C // ) / X( IF MEDICATN EQ 3, RETAIN ) \$  
LEVEL1 = ( DRUG 4, D // ) / X( IF MEDICATN EQ 4, RETAIN ) \$  
END \$

IT MAY BE HELPFUL TO NOTE THAT THE HEADING OF THE FIRST PRINTED CORRELATION MATRIX WOULD BE.....

NOV 20, 1970. FILE X DATA. / DRUG 1, A / AGE 21-30 / WEEK 0

## 21. DO LOOPS OF P-STAT OPERATIONS

A DO LOOP PERMITS A SEQUENCE OF P-STAT STATEMENTS TO BE EXECUTED A NUMBER OF TIMES. THE NUMBER OF TIMES DEPENDS ON THE RANGE OF AN INDEX, WHICH IS DEFINED IN THE DO STATEMENT ITSELF. EACH TIME THROUGH, CERTAIN FILE NAMES ARE CHANGED, USING THE CURRENT VALUE OF THE INDEX AS PART OF THE FILE NAME.

A DO LOOP HAS THREE PARTS...

----- 1 -----

A DO CARD. THIS HAS DO AS THE OPERATION NAME. FOR EXAMPLE...  
DO = 1/5 \$

THIS CARD, STARTING THE LOOP, DEFINES THE INDEX RANGE. IN THE ABOVE EXAMPLE, A LOOP WOULD BE EXECUTED 5 TIMES, WITH THE INDEX BEING 1, THEN 2, THEN 3 AND 4 AND 5. AS IN FORTRAN DO STATEMENTS, A THIRD INTEGER ( DO = 11 / 30 / 5 \$ ) SAYS... USE 11,16,21, AND 26 AS INDICES IN THE 4 PASSES THROUGH THE LOOP. THUS, THE DO CARD DEFINES AN INITIAL INDEX, A FINAL INDEX, AND PERMITS AN OPTIONAL STEP SIZE, SET TO 1 IF THERE WAS NO THIRD INTEGER IN THE DO STATEMENT.

----- 2 -----

VARIOUS P-STAT STATEMENTS. THESE ARE THE STATEMENTS THAT ARE LOOPED THROUGH. THERE WILL BE REFERENCES TO FILES IN SOME OF THESE STATEMENTS. EVERY FILE NAME ENDING IN .0 WILL BE CHANGED ( THE NAME, NOT THE FILE ITSELF ) EACH TIME THROUGH THE LOOP. FILE NAMES NOT ENDING IN .0 ARE NOT MODIFIED. THE NAME X.0 WOULD BE CHANGED TO X.1 THE FIRST TIME THROUGH ( ASSUMING THE INDEX STARTED WITH 1 ).

----- 3 -----

AN ENDDO\$ CARD, DEFINING THE END OF THE LOOP.

WHEN A DO CARD IS FOUND, ALL CARDS BETWEEN THE DO AND THE ENDDO\$ ARE READ AND STORED IN AN AREA THAT CAN HOLD UP TO 1600 CHARACTERS (I.E., THAT IS THE MAXIMUM SIZE OF A DO, 20 FULL CARDS). THEN THE LOOP IS EXECUTED WITH THE INDEX VALUE PLUGGED WHEREVER .0 WAS FOUND AT THE END OF FILE NAME. THE STEP SIZE IS THEN ADDED TO THE INDEX. IF IT IS NOT OVER THE FINAL INDEX VALUE, THE LOOP IS DONE AGAIN WITH THIS NEW INDEX PLUGGED WHERE .0 WAS IN THE BEGINNING, AND SO ON.

A DO LOOP CANNOT HAVE ANOTHER DO LOOP INSIDE IT.

DATA CARDS NECESSARY FOR OPERATIONS WITHIN THE DO MUST BE PLACED AFTER THE ENDDO, ONE SET FOR EACH PASS THROUGH THE DO, OR ELSE A CALL TO A MACRO CAN BE MADE (MACROS CAN HAVE INTERNAL DATA) IF THE DATA CARDS ARE THE SAME EACH TIME THROUGH THE DO LOOP.

CONSIDER THIS SIMPLE DO LOOP.

```
DO = 1/30 $
INTCDS, IN = A.0, COR = AIN.0 $
PRINT = 3/AIN.0$
ENDDO $
```

THIS MIGHT BE DONE IF A USER HAD DATA FROM 30 STATES, EACH A SEPARATE FILE ( NAMED A.1, A.2, THROUGH A.30 ), AND HE NEEDED CORRELATIONS OF THE VARIABLES IN EACH FILE. EXECUTION OF THIS LOOP WILI CAUSE 30 CORRELATION MATRICES TO PRINT. ( IF THIS CAUSES TOO MANY ACTIVE FILE NAMES FOR THE INTERNAL P-STAT SYSTEM TO MAINTAIN...

PURGE = AIN.0 \$  
COULD BE PUT AFTER THE PRINT STATEMENT. )

DO LOOPS AND MACROS (SEE SECTION ON MACROS) CAN INTERACT.

FOR EXAMPLE...

```
MACRO, DOMAC = A/G/B/C/D$
DO = A$
MERGE, LEFT = G(C 1-4), RIGHT = B, OUT = C$
D$
ENDDO$
ENDMACRO$
```

THE ABOVE DEFINES THE MACRO. THIS CARD USES THE MACRO...

```
DOMAC = ((2/6)) / X.0 / Y.0 / Z.0 / ((PRINT = Z.0)) $
```

IF A DO LOOP NEEDED A FEW DATA CARDS FOR EACH PASS THROUGH THE LOOP, AND THE DATA CARDS WERE THE SAME EACH TIME, A MACRO COULD BE DEFINED WHICH EXECUTES THE STEP USING THE DATA, AND INCLUDES THE DATA IN THE DEFINITION. THE 'DO' LOOP COULD THEN CALL THE MACRO EACH TIME.

## 22. JCL FOR REFERENCING TAPES ON THE 360

A RUN USING AN ALREADY CREATED PERMANENT DATA TAPE WOULD BE...

```
// JOB PTC.
/*SETUP UNIT=TAPE9, ID=(1357, READ, KEEP, SL)
// EXEC PSTAT52, P61=1357, NAME61=FILENAME
//PSTAT.SYSIN DD *
   ( P-STAT CARDS )
/*
```

CONSIDER THE '/\*SETUP' CARD....

'TAPE9' REQUESTS MOUNTING ON A NINE-TRACK TAPE DRIVE.

'1357' IS AN EXAMPLE OF A TAPE NUMBER AT PRINCETON. THE OPERATOR USES THIS INFORMATION TO LOCATE THE PROPER TAPE ON THE TAPE RACK.

'READ' MEANS THE TAPE WILL BE MOUNTED WITHOUT A RING ( I.E. IT CANNOT BE WRITTEN ON ). SAYING 'WRITE' WOULD OF COURSE PERMIT WRITING ON THE TAPE DURING THIS RUN.

'KEEP' IS MILDLY SUPERFLUOUS BUT SHOULD STILL BE THERE.

'SL' INDICATES THAT THE TAPE HAS AN CS/360 STANDARD TAPE LABEL. IT IS FOLLY FOR A 360 TAPE NOT TO BE STANDARD LABELLED.

IN THE '// EXEC' CARD....

'PSTAT52' IS THE NAME OF A CATALOGED PROCEDURE THAT DEFINES THE P-STAT SCRATCH FILES, ETC., AND BRINGS IN THE P-STAT OVERLAY.

'P61' MEANS THAT THIS IS A PERMANENT DATA TAPE USING LOGICAL TAPE NUMBER 61.

'1357' IS A PRINCETON TAPE NUMBER. IT IS CHECKED AGAINST THE VOLUME NUMBER IN THE STANDARD LABEL TO ENSURE THAT THE CORRECT TAPE HAS BEEN MOUNTED.

'NAME61' SAYS THAT A DATA-SET NAME FOLLOWS, AND IT IS THE NAME OF THE FIRST DATA-SET ON TAPE 61.

'FILENAME' IS THE NAME GIVEN TO THE DATA-SET WHEN THE DATA-SET WAS WRITTEN FOR THE FIRST TIME. THIS OCCURS WHEN 'DISP61=NEW', FOR EXAMPLE, WAS PRESENT. SEE BELOW.

AT PRINCETON, A VOLUME LABEL, INCLUDING THE PRINCETON TAPE NUMBER, IS WRITTEN WHEN THE TAPE IS STANDARD LABELLED. THIS OCCURS WHEN THE TAPE IS ISSUED TO THE USER. WHEN THE TAPE IS USED IN A P-STAT RUN FOR THE FIRST TIME, A DATA-SET IS WRITTEN. THAT DATA-SET HAS A

LABEL, OR NAME, A DSNAME. WHEN THE TAPE IS USED AGAIN, CHECKING THAT THE DSNAME IS CORRECT IS A GOOD SOURCE OF PROTECTION.

A DATA-SET NAME CAN BEGIN WITH UP TO 8 LETTERS AND/OR NUMBERS. IT MUST START WITH A LETTER. IT CAN BE A SINGLE NAME, OR SEVERAL NAMES, SEPARATED BY PERIODS.

IF THE NAME HAS PERIODS IN IT, THE ENTIRE NAME MUST BE IN QUOTES....

```
NAME61='NAME.OF.FILE'
```

IF A SINGLE NAME IS USED, QUOTES ARE OPTIONAL....

```
NAME61='NEWDATA'      OR
NAME61=NEWDATA
```

A RUN USING A PERMANENT DATA TAPE AND AN ASSIGN/ATTACH TAPE (ASSUME THE PERMANENT DATA TAPE IS NUMBER 1234 AND IS TO BE WRITTEN ON, AND THE ATTACH TAPE IS NUMBER 5555 AND IS ONLY TO BE READ, AND BOTH ARE 9 TRACK) WOULD BE...

```
// JOB ETC.
/*SETUP UNIT=TAPE9, ID=(1234, WRITE, KEEP, SL)
/*SETUP UNIT=TAPE9, ID=(5555, READ, KEEP, SL)
// EXEC PSTAT52, P61=1234, NAME61='PDT.DSNAME',
// A51=5555, NAME51='ATTACH.DSNAME'
//PSTAT.SYSIN DD *
( P-STAT CARDS )
/*
```

THE CATALOGED PROCEDURES ALLOW UP TO 3 PERMANENT DATA TAPES AND UP TO 5 ASSIGN/ATTACH TAPES. THE PERMANENT TAPES IN THE PROCEDURE INVOLVE NUMBERS 61-63 ( SEE OPERATION 'NEWPDT' ). IF NUMBERS 64-69 NEED TO BE USED, SEPARATE 'DD' CARDS MUST BE USED. THE ASSIGN/ATTACH TAPES INVOLVE NUMBERS 51-55. AGAIN, MORE CAN BE USED WITH 'DD' CARDS.

THE CATALOGED PROCEDURE MAKES USE OF THE 'NULLFILE' FACILITY IN OS/360 JCL. AS MENTIONED ABOVE, PERMANENT TAPES 61-63 AND ASSIGN/ATTACH TAPES 51-55 ARE ALLOWED, IF YOU REFERENCE THEM ON THE 'EXEC' CARD. 'P61' INVOKES USAGE OF PERMANENT TAPE 61, 'A52' INVOKES ASSIGN/ATTACH TAPE 52. THUS, P61, P62, P63, A51, A52, A53, A54, AND A55 ARE KEYWORDS ON THE PROCEDURE WITH WHICH A TAPE NUMBER CAN BE ASSOCIATED.

NOTE - NUMBERS LIKE 51 AND 61 ARE P-STAT NUMBERS. A PRINCETON TAPE NUMBER LIKE 1713 IS USED TO FIND THE TAPE IN THE COMPUTER ROOM. THE 360 SYSTEM READS THE STANDARD LABEL TO SEE THAT IT IS REALLY THE CORRECT TAPE ( THE TAPE NUMBER IS PART OF THE STANDARD LABEL ). 'P61 = 1713' ESTABLISHES A CORRESPONDENCE. WHEN P-STAT REFERS TO 61, IN THIS EXAMPLE, IT ACTUALLY GETS TAPE 1713.

WHEN, FOR EXAMPLE, 'P63=0723' IS USED, THE NAME FOR THE DATA-SET ON TAPE 0723 MUST ALSO BE SUPPLIED. THIS NAME WAS SUPPLIED WHEN THE DATA SET WAS FIRST BEGUN. THIS IS A USEFUL BIT OF REDUNDANCY. IF 'SEPT69.NJDATA' IS THE DSNAME ON TAPE 0723 AND YOU CITE TAPE 0732 ( ALSO, PERHAPS, ONE OF YOURS, BUT NOT WHAT YOU WANT TO BE USING HERE ), THE DSNAMP CHECK KILLS THE JOB. KEYWORDS NAME51, NAME52, NAME53, NAME54, NAME55, NAME61, NAME62, NAME63 ARE ALLOWED. IF P63 IS USED, NAME63 MUST BE USED WITH IT.

SO FAR, TWO KEYWORDS MUST BE SUPPLIED FOR EACH P-STAT TAPE BEING USED. ONE ADDITIONAL KEYWORD IS POSSIBLE. 'DISP61=NEW' SHOULD BE USED, FOR EXAMPLE, WHEN A NEW P61 IS BEING INITIALIZED ( BY 'NEWPDS' ). IF THIS IS NOT PUNCHED, THE PROCEDURE WILL DEFAULT TO 'OLD' AND THEREFORE CHECKS THAT THE HEADER LABEL OF THAT DATA-SET ON THAT TAPE IS ALREADY THERE AND THAT THE DATA-SET NAME IS AS ADVERTISED. AGAIN, DISP51, DISP52, ETC., CAN BE USED, BUT FOR THIS KEYWORD, 'NEW' IS THE ONLY SENSIBLE SETTING, IF IT IS USED AT ALL.

JUST FOR THE RECORD, DISP STANDS FOR DISPOSITION.

ASSUME TAPE 0123 IS A PERMANENT DATA TAPE WITH FILE 'A' SOMEWHERE ON IT. A NEW TAPE, 4567, IS AVAILABLE AND WE WISH TO PLACE FILE A ON IT SO THAT IT CAN LATER BE 'ATTACHED'.

```
// JOB ETC.
/*SETUP UNIT=TAPE9, ID=(0123, READ, KEEP, SL)
/*SETUP UNIT=TAPE9, ID=(4567, WRITE, KEEP, SL)
// EXEC PSTAT52,
// P61=0123, NAME61='PDT.ABL',
// A51=4567, NAME51='SENIOR.RAW.DATA', DISP51=NEW
//PSTAT.SYSIN DD *
  ASSIGN = A, TAPE = 51 $
  FIND = A $
  END $
/*
```

USE OF NON-LABELLED OR MULTI-FILED( O.S. FILES, NOT P-STAT FILES ) TAPES NEEDS SEPARATE JCL TO OVERRIDE THE PROCEDURE.

NOTE - IN THE '// EXEC PSTAT52' CARD, THE TAPE INFORMATION, IF IN USE, MUST BE PUNCHED WITH NO INTERNAL BLANKS.

```
// EXEC PSTAT52, P61=0123, NAME61 = PDT1
```

WOULD ELICIT A JCL ERROR MESSAGE.

NOTE - AT PRINCETON, THE IBM MODEL 2401-6 TAPE DRIVES PERMIT EITHER 800 OR 1600 DENSITY. IN JCL STATEMENTS, 'DEN=2' MEANS 800 AND 'DEN=3' MEANS 1600. THE P-STAT PROCEDURES ASSUME 1600 DENSITY ( AS OF LATE JANUARY, 1970 ) FOR FILES 51-55 AND 61-63. THE FOLLOWING CARDS, INSERTED BEFORE THE '//PSTAT.SYSIN DD \*'

CARD, WOULD ALLOW THE USE OF 800 DENSITY TAPES IN THE PREVIOUS EXAMPLE.

```
//PSTAT.FT51F001 DD DCB=DEN=2
//PSTAT.FT61F001 DD DCB=DEN=2
```

NOTE - THEY ARE ORDERED BY NUMBER, 51 BEFORE 61. THE ORDERING IS NECESSARY IN 'DD' CARDS SUCH AS THESE, BUT IS NOT NECESSARY WITHIN THE '// EXEC PSTAT52,ETC. ' STATEMENT.

```
*****
* CARD IMAGE TAPES FOR 'DATA'      *
*****
```

THIS EXAMPLE SHOWS A 'DATA' OPERATION WHOSE INPUT IS ON TAPE 92, NUMBERED 3456, WHOSE DATA SET NAME IS 'CARDS'.

```
// JOB
/*SETUP UNIT=TAPE9, ID= (3456, READ, KEEP, SL)
// EXEC PSTAT52
//PSTAT.FT92F001 DD DSNAME=CARDS, UNIT=TAPE9, LABEL=(1, SL),
VOL=SEP=3456, DISP= (OLD, KEEP)
//PSTAT.SYSIN DD *
DATA = A, TAPE = 92, ETC.
```

23. ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM

NEW LINKS ( OPERATIONS, PROGRAMS, ETC. ) CAN BE ADDED TO THE P-STAT SYSTEM BY USERS. THESE CAN BE EITHER TEMPORARY ( CALLABLE BY THE OPERATION NAMES USER1, USER2, USER3, USER4, OR USER5, OR THEY CAN BE PERMANENT LINKS ( WITH THE OPERATION NAMES ENTERED INTO THE SYSTEM DIRECTORY ).

THERE ARE 2 STEPS IN DOING THIS,

A - WRITING THE LINK - WHAT CONVENTIONS ARE PROVIDED TO USE INFORMATION ON CONTROL CARDS, TO ACCESS A FILE, ETC.

B - ATTACHING THE LINK TO THE SYSTEM AND USING IT - HERE, TECHNIQUES FOR VARIOUS COMPUTERS DIFFER GREATLY. THIS DESCRIPTION HOLDS FOR THE 360 OVERLAY VERSION.

=====  
\* A - WRITING THE LINK \*  
=====

FIRST, HOW DOES THE SYSTEM KNOW ABOUT THIS LINK.  
WHEN A CONTROL STATEMENT IS READ, FOR EXAMPLE.

MERGE, LEFT = A, RIGHT = D, OUT = C\$

THE OPERATION NAME MERGE (FOUND ON THE CARD) IS COMPARED WITH ALL KNOWN NAMES. WHEN A MATCH IS FOUND WITH A KNOWN NAME, A TRANSFER IS TAKEN, OUT OF THE EXECUTIVE ROUTINE, TO THE APPROPRIATE PROGRAM. THE OVERLAY IS ORIGINALLY GENERATED WITH NULL OPERATIONS IN THE USER1 THROUGH USER5 POSITIONS.

A USER-PROVIDED LINK CAN BE A TEMPORARY LINK OR A PERMANENT ONE. A TEMPORARY LINK USES THE OPERATION NAMES USER1 THROUGH USER5. THESE NAMES ARE ALREADY IN THE SYSTEM. THE SYSTEM, WHENEVER A CONTROL CARD SAYS USER1, FINDS THE MATCH AND TRANSFERS BLINDLY, ASSUMING THE USER'S PROGRAM IS INDEED THERE.

MAKING A LINK PERMANENT REQUIRES ADDING THE NEW NAME TO SYSTEM SUBROUTINE PSINOP (WHERE THE KNOWN NAMES ARE GENERATED), MODIFYING EXEC AND PSMAIN TO MAKE THE APPROPRIATE TRANSFER, AND, OF COURSE, MAKING THE NEW LINK A PART OF THE OVERLAY STRUCTURE. ( I.P., MODIFYING THE OVERLAY DEFINITION DECK ).

IN ANY CASE, A NEW OVERLAY MUST BE GENERATED. ON A 91 WITH MVT, A REGION OF 150K, CPU TIME OF 10 SECONDS, AND 9K I/O REQUESTS ARE INVOLVED. ABOUT 4 ACTUAL MINUTES ARE USED, AND THE FULL PRINTOUT IS ABOUT 60 PAGES.

CONSIDER A LINK THAT, FOR A FILE UP TO 60 BY 60, CREATES A NEW FILE BY ADDING 3 TO EACH VALUE IN THE OLD FILE. THIS PROGRAM COULD BE PUNCHED AS IS AND RUN ON THE 360 P-STAT SYSTEM.

## SUBROUTINE USER1

```

C
C     TEMPORARY LINK TO ADD 3. TO EACH SCORE
C     FIRST, P-STAT COMMON USABLE BY A LINK.
C
COMMON/PSLCLL/ QQRL(2,450), QQCL(2,450), QQL(2), QQLL(2),
1 QQLL(2), QQHEAD(20), QQGL, KKIN, KKOUT, KK5, KK6, QQBLAN,
2 QOMISS, QQVBAR, KCOL, KKUF1, KKUF2, KKUF3
C     NOW, STORAGE FOR THIS PROGRAM
C
COMMON X(60,60)
C     STORAGE FOR IDENTIFIERS
C
DIMENSION IN(2), OUT(2)
C     SET UP COMMUNICATION WITH THE CONTROL CARD
C     THE CARD COULD SAY,
C     USER1, IN = A, CUT = A3 $
C
DATA IN / 4H      , 4H IN /
DATA OUT / 4H     , 4H OUT /
C     GET THE FILE WHOSE NAME WAS ASSOCIATED WITH 'IN'
C     ON THE CONTROL CARD, ' IN = A '
C     SUBROUTINE PSPIN WILL PUT THE DATA IN X, THE ROW
C     LABELS IN QQRL, THE COLUMN LABELS IN QQCL, AND WILL
C     SET NR BY NC TO ITS ACTUAL SIZE
C     IT CANNOT BE OVER 60 BY 60, THE DIMENSIONS OF X
C
CALL PSPIN ( IN, X, QQRL, QQCL, NR, NC, 60, 60)
DO 50 I = 1, NR
DO 40 J = 1, NC
IF ( X(I,J) -QOMISS) 40, 30, 40
30 CALL PSBAD (23H *IN HAS MISSING DATA..)
40 X(I,J) = X(I,J) +3.
50 CONTINUE
C     NOW, CAUSE THE SYSTEM TO CREATE A NEW FILE
C
CALL PSFOUT (OUT, X, QQRL, QQCL, NR, NC, 60, 60,
1 26H *IN WITH 3. ADDED TO IT..)
CALL PSNFX
STOP
END

```

THERE ARE A NUMBER OF SPECIFIC TOPICS TO EXPLAIN,

1. P-STAT SYSTEM DIMENSIONING.
2. THE DIMENSIONING WITHIN A LINK.
3. THE DIMENSIONING AND INITIALIZATION OF THE IDENTIFIERS.
4. COORDINATION BETWEEN THE SUBROUTINE NAME AND THE P-STAT OPERATION NAME.
5. ACCESSING THE CONTROL CARD.
6. WHAT IS A FILE.
7. HOW TO GET A COPY OF IT IN CORE.
8. HOW TO CREATE A NEW FILE.
9. ERROR MESSAGES.
10. WHEN DONE, RETURNING TO THE EXECUTIVE ROUTINE.
11. RE-FORMATTING INFORMATION INTERNALLY.

\*\*\*\*\*

1. P-STAT SYSTEM DIMENSIONING\*

\*\*\*\*\*

A \* IS NEXT TO A NAME IF A TYPICAL LINK IS LIKELY TO USE IT.

\*QOQL (2,450), QOCL (2,450). RI = ROW LABELS, CL = COLUMN LABELS. THE SYSTEM PERMITS A FILE TO HAVE UP TO 450 COLUMNS AND ANY NUMBER OF ROWS (NO PROMISES ABOVE 100,000 - FOR THAT MATTER, NO PROMISES BELOW 100,000 ROWS, BUT ABSOLUTELY NO PROMISES ABOVE 100,000). EACH ROW AND EACH COLUMN HAS A LABEL, UP TO 8 CHARACTERS, STORED 4 CHARACTERS PER WORD, HENCE 2 BY 450. THE NEED FOR 450 COLUMN LABELS IS CLEAR. PERMANENT SPACE FOR 450 ROW LABELS IS LESS MANDATORY, BUT CONVENIENT ENOUGH SO IT IS THERE. A USER MAY USE THESE ANY WAY HE WISHES.

\* QOL (2), QOLL (2), QOLLL (2). THESE ARE 3 SEPARATE LOCATIONS FOR LABEL STORAGE. FOR EXAMPLE, WHEN A ROW OF A FILE IS READ, THE LABEL (A TWO WORD LABEL) NEEDS TO GO SOMEWHERE. QOL, QOLL, AND QOLLL ARE PROVIDED FOR TEMPORARY LABEL STORAGE.

\*QOHEAD(20). THIS CONTAINS THE CURRENT HEADING, IN 20A4 FORMAT. IF A USER LINK PRINTS SOME RESULTS, IT MAY WISH TO HEAD THE RESULTS.

QOGL - USE OF QOGL FOR A LABEL ( OR LABELS ) TELLS THE SYSTEM TO GENERATE THEM ( L1, L2, ETC. ).

KKIN, KKOUT - NAMES OF PSEUDO-BUFFERS ( OR INTERNAL FILES ) FOR WRITING AND READING, I.E., RE-FORMATTING.

KK5 - THE INPUT FILE

\*KK6 - THE OUTPUT FILE. IF A LINK PRINTS RESULTS, IT WRITES TO FILE KK6.

QOBLAN - CONTAINS BLANKS READ IN A4 FORMAT.

\*QOMISS - CONTAINS THE MISSING DATA VALUE.

QOVBAR - SPE THE OPERATION 'VBAR'.

KKCOL - CONTAINS THE MAXIMUM NUMBER OF COLUMNS PERMITTED - CURRENTLY 450.

KKUP1, KKUP2, KKUP3. THESE ARE SCRATCH FILES USABLE BY A LINK FOR UNFORMATTED WRITES AND READS.

\*\*\*\*\*

2. A LINK'S OWN DIMENSIONING\*

\*\*\*\*\*

WHEN CHOOSING NAMES FOR VARIABLES, NOTE THAT ALL P-STAT SYSTEM NAMES START WITH KK OR QQ. NAMES OF P-STAT SYSTEM SUBROUTINES START WITH PS OR PX. A PROGRAMMER ABOUT TO WRITE A NEW LINK OUGHT TO CHECK HIS PROSPECTIVE SUBROUTINE NAMES WITH THE ALPHABETIC ( I.E., SORTED ) LIST AT THE END OF THE 'SOURCE' OUTPUT LISTING.

A USER SHOULD, OBVIOUSLY, AVOID CONFLICTING NAMES.

VERSION 52 PERMITS ABOUT 23,000 WORDS OF DIMENSIONING IN BLANK (UNLABELLED) COMMON FOR A USER LINK. A LINK CAN, OF COURSE, USE MORE DIMENSIONING. IF SIGNIFICANTLY MORE IS USED, THE REGION SIZE FOR THE OVERLAY WILL NEED TO BE THAT MUCH LARGER.

\*\*\*\*\*

3. IDENTIFIERS\*

\*\*\*\*\*

A P-STAT LINK, WHEN IT CALLS A SYSTEM SUBROUTINE, GENERALLY TELLS THE SUBROUTINE WHICH IDENTIFIER IT IS INTERESTED IN. IF A CONTROL CARDS SAYS... IN = A, THE LINK NEVER KNOWS THE NAME OF THE FILE. IT WILL INSTEAD READ WHATEVER FILE ( HERE, FILE 'A' ) IS ASSOCIATED WITH IDENTIFIER 'IN'.

CALL PSFIN (IN, X, ETC...)

THE USE OF 'IN' IN THE CALL IS NOT ADEQUATE, BECAUSE 'IN' IS THE NAME OF A VARIABLE WHEN IT WAS COMPILED, AND IS NOT NECESSARILY ITS CONTENTS DURING EXECUTION.

CALL PS\*IN(2HIN, X, ETC...)

COULD WORK, BUT 2HIN IS ILLEGAL IN FUNCTION CALLS. THEREFORE, EACH IDENTIFIER IS DIMENSIONED(2), AND A DATA STATEMENT PLACES THE LITERAL CHARACTERS INTO THE DIMENSIONED AREA.

FOR EXAMPLE..

```

DIMENSION LEFT(2), RIGHT(2), OUT(2)
DATA LEFT / 4H      , 4HLEFT /
DATA RIGHT / 4H    R, 4HRIGHT /
DATA OUT  / 4H      , 4H OUT /

```

THE CHARACTERS MUST BE RIGHT JUSTIFIED.

\*\*\*\*\*

4. COORDINATION\*

\*\*\*\*\*

WHEN THE EXECUTIVE LINK IN P-STAT FINDS USE OF THE OPERATION NAME 'USER1' IT TRANSFERS CONTROL TO A SUBROUTINE NAMED 'USER1'. USE OF 'USER2' GETS 'USER2', ETC.

\*\*\*\*\*

5. ACCESSING THE INFORMATION ON A CONTROL CARD\*

\*\*\*\*\*

ALL ACCESS CALLS USE IDENTIFIERS. IN ADDITION, THERE ARE OTHER P-STAT SUBROUTINES WHICH MAY BE USED IN LINKS. FOR EXAMPLE - THE CONTROL CARD TEXT IS IN THE FIRST PARENTHESES -

```

(NFAC = 7)      CALL PSI(NFAC, N)
                N WILL BE SET TO A FIXED POINT 7.

```

```

(TOL = 0.2)    CALL PSP(TOL, X)
                X GETS SET TO 0.2 .

```

```

(LEFT = A)     IF (PSUSED (LEFT)) 40, 60, 40
                IF LEFT WAS USED AS AN IDENTIFIER ON A CONTROL CARD, THE
                ZERO BRANCH (60) IS TAKEN.

```

```

(PRINT = A/B)  IF (PSMARG ( PRINT ))140, 60, 140
HERE, PRINT WAS USED AS AN IDENTIFIER TWICE. THE PROGRAM
MAY HAVE USED THE FIRST FILE, AND NOW NEEDS TO KNOW WHETHER TO EXIT,
OR TO LOOP BACK AND DO WHATEVER IT DOES TO THE NEXT FILE, AND SO ON.
MARG STANDS FOR MULTIPLE ARGUMENT.

```

AN INTERNAL TABLE EXISTS, SOMEWHAT LIKE,

IDENTIFIER	FILE NAME
PRINT	A
PRINT	B

PSMARG SEARCHES THE LEFT COLUMN, LOOKING FOR AN IDENTIFIER WHICH MATCHES ITS OWN ARGUMENT (IN THIS CASE PRINT). ONE HIT MUST BE FOUND - THE REFERENCE TO THE FILE ALREADY DONE. THAT USE OF PRINT IS CHANGED TO // // // // // . THEN ANOTHER USE OF PRINT IS SOUGHT.

IF NONE, TAKE THE + BRANCH. IF USED AGAIN, TAKE THE ZERO BRANCH.

\*\*\*\*\*

6. WHAT IS A FILE\*

\*\*\*\*\*

INPUT OF FILES (OR, FOR THAT MATTER, OUTPUT OF FILES) CAN BE DONE TWO WAYS, BY THE ENTIRE FILE, OR ONE ROW AT A TIME. THIS SUGGESTS FILES ARE STORED SOMEWHERE BY ROWS, WHICH IS TRUE. A NR BY NC FILE HAS, A 20 WORD HEADING (WHATEVER QOHEAD WAS SET TO WHEN IT WAS CREATED), A 15 WORD HISTORY ( IF FILE 'C' WAS CAUSED BY A LEFT-RIGHT MERGE OF A AND B, ITS HISTORY READS ' LEFT-RIGHT MERGE OF A AND B ', AN 8 CHARACTER NAME, NC 8 CHARACTER COLUMN LABELS, AND THEN NR ROWS. EACH ROW HAS AN 8 CHARACTER ROW LABEL AND THEN NC SCORES. EACH LABEL IS STORED IN TWO WORDS.

\*\*\*\*\*

7. HOW TO GET A COPY OF A FILE INTO CORE\*

\*\*\*\*\*

A LINK MAY BRING AN ENTIRE FILE INTO CORE (IF IT CAN FIT IN CORE). CALL PSPIN ( IN, X, QOQL, QOCL, NR, NC, 80, 40 ) SAYS..... FIND THE FILE ASSOCIATED WITH 'IN=' ON THE CARD, READ IT INTO X, DIMENSIONED 80 BY 40, PUT COLUMN LABELS INTO QOCL, ROW LABELS INTO QOQL, AND SET NR AND NC TO ITS SIZE.

READING A FILE INTO CORE DOES NOT CAUSE IT TO VANISH FROM WHEREVER P-STAT HAS BEEN STORING IT. ALSO, NR AND NC ARE THE SIZE OF WHAT GETS TO THE LINK. FILE A COULD BE 200 BY 60, BUT IF THE CARD SAID IN = A(R31-100) (C20, 41-55), NR WILL BE 70 AND NC WILL BE 16.

IF ONE ROW AT A TIME IS DESIRED, TWO CALLS ARE INVOLVED.

CALL PSRLAB ( IN, QOCL, NC )

THIS SUBROUTINE GETS THE COLUMN LABELS OF THE FILE ASSOCIATED WITH 'IN', PUTS THEM INTO QOCL ( OR WHEREVER ONE SAYS, AS LONG AS IT IS DIMENSIONED AT LEAST 2 BY NC), AND TELLS US THERE ARE NC COLUMNS - NOT NECESSARILY IN THE FILE, BUT COMING INTO THIS LINK. I.E., COLUMN SELECTION MAY BE OCCURRING.

THEN, TO GET A ROW, ONE SAYS

IF ( PSRROW ( IN, QOQL, T ) ) 140, 140, 500

IF THERE WAS INDEED A ROW LEFT IN THE FILE, THAT NEXT ROW OF THE FILE IS COPIED INTO T (WHICH BETTER BE DIMENSIONED AT LEAST NC). THE LABEL IS PUT INTO QOQL, AND WE GO TO 140. IF NO MORE ROWS OF THE FILE ARE AVAILABLE, QOQL AND T ARE UNTOUCHED AND WE GO TO 500.

\*\*\*\*\*  
8. HOW TO CREATE A NEW FILE\*  
\*\*\*\*\*

AS IN ACCESSING FILES, A NEW FILE CAN BE CREATED ALL AT ONCE, OR ONE ROW AT A TIME.

CALL PSFOUT(OUT,X,QQRL, QQCL, NR, NC, 80, 40, 26H \*IN CHANGED IN SOME WAY..)

AS WITH PSFIN, OUT IS AN IDENTIFIER, X IS DIMENSIONED 80 BY 40 BUT IS ACTUALLY NR BY NC, NR ROW LABELS ARE IN QQPL, AND NC COLUMN LABELS ARE IN QQCL. THE LAST ARGUMENT, 26H ETC, IS A HOLLERITH FIELD PERMITTING A SHORT HISTORY TO BE ASSOCIATED WITH THIS FILE. IT IS NICE, IN THIS FIELD, TO INCLUDE THE NAME OF THE INPUT FILE (OR FILES) THAT WERE INVOLVED IN CREATING THE NEW FILE, BUT THE NAME IS NOT KNOWN TO THE LINK. HOWEVER, THE IDENTIFIER OF THE NAME IS KNOWN, SO THE SUBROUTINE SCANS THE HOLLERITH FIELD LOOKING FOR A \* FOLLOWED BY AN IDENTIFIER AND SUBSTITUTES THE CURRENT FILE NAME ASSOCIATED WITH THE IDENTIFIER. THE HOLLERITH ARGUMENT IS OF ANY LENGTH (UP TO 50 CHARACTERS) AND ITS END MUST BE INDICATED BY A DOUBLE PERIOD (INCLUDED IN THE 60 CHARACTERS).

THE ROW-AT-A-TIME SEQUENCE HAS 3 CALLS, WRITE LABELS (CALLED ONCE), WRITE A ROW (CALLED ONCE FOR EACH ROW), AND CLOSE THE FILE (CALLED ONCE).

CALL PSWLAB ( OUT, QQCL, NC, 26H \*IN CHANGED IN SOME WAY.. )  
CALL PSWROW ( OUT, QQL, )  
CALL PSCLOS ( OUT )

\*\*\*\*\*  
9. ERROR MESSAGES\*  
\*\*\*\*\*

ANY LINK CAN CALL PSBAD WITH A SINGLE HOLLERITH ARGUMENT, UP TO 120 CHARACTERS, WITH THE SAME CONVENTIONS AS IN PSWLAB AND PSFOUT ( \*IDENTIFIER, AND ..).

SUCH A CALL ADOPTS THE LINK, PRINTS THE MODIFIED MESSAGE, AND GOES BACK TO THE EXECUTIVE ROUTINE.

FOUR OTHER ERROR ROUTINES ARE AVAILABLE, PSBADA, PSBDAA, PSBADI, AND PSBADR. THEY PERMIT, RESPECTIVELY, THE TRANSMISSION OF AN ALPHABETIC LABEL, TWO SUCH LABELS, AN INTEGER, OR A REAL NUMBER. FOR EXAMPLE, IF QQCL(1,J) HAS AN UNMATCHING LABEL,

CALL PSBADA ( 23H LABEL DOES NOT MATCH.., QQCL(1,J) )

\*\*\*\*\*  
 10. WHEN DONE, GETTING BACK TO THE EXECUTIVE ROUTINE\*  
 \*\*\*\*\*

THE FOLLOWING CALL RETURNS CONTROL TO THE P-STAT EXECUTIVE ROUTINE, WHICH PROCESSES THE NEXT CONTROL CARD.

CALL PSNEXT

\*\*\*\*\*  
 11. REFORMATTING INFORMATION INTERNALLY\*  
 \*\*\*\*\*

THREE SPECIAL SUBROUTINES EXIST..

CALL PSWB                    PRECEDES A WRITE TO THE INTERNAL BUFFER. THE FILE NUMBER USED IN WRITE STATEMENTS IS KKOUT.

CALL PSRB                    PRECEDES A READ FROM THE INTERNAL BUFFER. THE INTERNAL WRITE MUST HAVE PRECEDED THIS.

CALL PSRRB                    MEANS RE-READ BUFFER, USED IF THE INTERNAL FILE IS READ MORE THAN ONCE. THIS IS CALLED BEFORE EACH SUBSEQUENT RE-READ.

THIS EXAMPLE CONVERTS J, AN INTEGER VARIABLE, INTO ITS ALPHABETIC FORM, FOR USE AS A NAME.

```

CALL PSWB
WRITE (KKOUT,20) J
20  FORMAT (1X,I4)
CALL PSRB
READ (KKIN, 40) AJ
40  FORMAT (1X,A4)

```

(THE INITIAL 1X IN THESE FORMATS IS TO AVOID PROBLEMS IF THE PROGRAM IS RUN ON A GE-635.)

```
=====
* B - MAKING AND USING A VERSION OF P-STAT WITH THE NEW LINK *
=====
```

A CATALOGED PROCEDURE, 'PSTAT52U', EXISTS AT PRINCETON WHICH

1. COMPILES A NEW USER-SUBROUTINE WHICH SHOULD BE NAMED 'USER1' ( OR USER2, ETC ). OTHER NEW SUBROUTINES WHICH USER1 CALLS CAN ALSO BE COMPILED DIRECTLY FOLLOWING USER1.
2. CONSTRUCTS AN OVERLAY INCLUDING THE NEW USER1.
3. EXECUTES THE NEW ( TEMPORARY ) OVERLAY.

FOR EXAMPLE.....

```
// JOB   ETC.
/*SETUP UNIT=2304, ID=BBB201
// EXEC PSTAT52U, TAPE-INFORMATION-AS-BEFORE
//PORT.SYSIN DD *
          ( SUBROUTINE(S) TO BE COMPILED, INCLUDING USER1 )
/*
//PSTAT.SYSIN DD *
          ( THE P-STAT CARDS, INCLUDING A USER1 COMMAND )
/*
```

THE MAKING OF THE OVERLAY REQUIRES ACCESS TO SEVERAL LIBRARIES ON DISK BBB201. IF NO TAPE INFORMATION IS NEEDED, THE '// EXEC' CARD BECOMES SIMPLY ...

```
// EXEC PSTAT52U
```

## A USER LINK EXAMPLE.....

TAPE 0333 ( WHICH HAS A DSNAME OF 'BUHLER.TESTDATA' FOR ITS FIRST DATASET ) HAS FILE 'A' ON IT, 3000 ROWS BY 60 COLUMNS. WE WISH TO CREATE FILE 'B', WHICH WILL HAVE ROWS 1, 4, 9, 16, 25, 36, 49, 64, ETC., OF FILE 'A'. FILE 'B' WILL THEN BE PRINTED AND SAVED.

```
// JOB.....
/*SETUP UNIT=2314, ID=EBB201
/*SETUP UNIT=TAPE9, ID=(0333, WRITE, KEEP, SL)
// EXEC PSTAT520, P61=0333, NAME61='BUHLER.TESTDATA'
//PORT.SYSIN DD *
  SUBROUTINE USER1
    DIMENSION IN(2), OUT(2)
    COMMON X(60), CL(2,60), RL(2)
    DATA IN / 4H      , 4H IN /, OUT / 4H      , 4H OUT /
    CALL PSPLAB ( IN, CL, NC )
    CALL PSWLAB ( OUT, CL, NC, 28H ROWS 1,4,9,16,ETC. OF *IN.. )
    L = 1
    NR = 0
  20 IF ( PSROW ( IN, RL, X ) ) 40, 40, 100
  40 NR = NR + 1
    IF ( NR - L * L ) 20, 60, 50
  50 CALL PSBAD1 ( 17H ERROR IN USER1.., L )
  60 CALL PSWROW ( OUT, RL, X )
    L = L + 1
    GO TO 20
 100 CALL PSCLOS ( OUT )
    CALL PSNFXT
    STOP
  END

/*
//PSTAT.SYSIN DD *
HEAD = ( TAKE A, MAKE B USING TAPE 0333 ) $
FIND = A $
USER1, IN = A, OUT = B $
PRINT = 2 / B $      SAVE = B $      END $
/*
```

24. ORGANIZATION OF P-STAT DISTRIBUTION TAPES

SEE SECTION 1, 'INTRODUCTION', REGARDING P-STAT DISTRIBUTION POLICY.

THE P-STAT SYSTEM IS DISTRIBUTED ON TAPE IN TWO DIFFERENT FORMS - A 360 VERSION AND A VERSION FOR NON-360 COMPUTERS.

.....  
.....THE NON-360 TAPE.....  
.....

THE NON-360 TAPE IS AT 800 BPI. IT HAS ABOUT 23,000 SOURCE CARDS, AN ENDFILE, ABOUT 4600 MANUAL CARDS, AN ENDFILE, ABOUT 900 PRIMER CARDS, AN ENDFILE, ABOUT 400 CARDS DESCRIBING THE OVERLAY STRUCTURE, AND AN ENDFILE. THESE ARE UNBLOCKED 80 CHARACTER RECORDS. THE TAPE IS 2400 FEET, 7 TRACK, AND IS WRITTEN IN EVEN PARITY.

A NON-360 INSTALLATION WOULD NEED TO .....

- 1 - REWRITE 'PSALCO' AND 'PSNVAL'. NOTE THE COMMENTS IN PSNVAL. PSALCO IS INVOLVED WITH COLIATING SEQUENCES, PSNVAL DOES A TABLE LOOKUP. BOTH ARE IN FORTRAN, BUT TOTALLY 360 HARDWARE DEPENDENT FORTRAN.
- 2 - CHANGE 'PSINIT' SO THAT KKDISC = 2, TURNING OFF USE OF LINKLIST STORAGE. ALSO CHANGE THE 'VBAR' DATA STATEMENT.
- 3 - SUBSTITUTE DUMMY ROUTINES FOR 'PSRDDX' AND 'PSWRDX'. LINKLIST MAY BE IMPLEMENTABLE IN SOME WAY ON OTHER COMPUTERS. SEE THE SECTION ' SYSTEM USE OF STORAGE'. IF NOT, THE ABOVE POINTS 2 AND 3 SHOULD BE DONE.
- 4 - PROVIDE AN INTERNAL FORMATTED WRITE-THEN-READ FACILITY. SEE POINT 11, 'REFORMATTING INFORMATION INTERNALLY' IN THE SECTION NAMED 'ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM'. ON THE 360, THE INCLUSION OF SUBROUTINE 'TAPE99' ACCOMPLISHES INTERNAL FORMATTING.
- 5 - CHECK THE OVERLAY STRUCTURE. THE SUPPLIED DECK RUNS ON A 360. HOWEVER, EACH OPEATING SYSTEM SEEMS TO HAVE DIFFERENT RULES FOR OVERLAYS, AND THIS DECK SHOULD BE EXAMINED WITH CARE. SOME CODE ( IN PSMAIN, FOR EXAMPLE ) MAY NEED TO BE CHANGED IF THE WAY OF CALLING A LINK IS NOT HOW THE 360 DOES IT ( IMPLICITLY, I.E., AS IF CALLING A SUBROUTINE ).
- 6 - SUBSTITUTE A DUMMY ROUTINE FOR 'OSIRIS'. THIS PROGRAM, USED BY THE P-STAT OPERATION 'OSIRIS.P', INTERFACES TO THE OSIRIS SYSTEM WHICH ONLY RUNS ON 360'S. IT MUST BE DUMMIED BECAUSE IT CALLS YET OTHER ROUTINES IN THE PSTAT52.ISRLTB LIBRARY.
- 7 - NOTE THE USE, IN PSRDC, OF 'END=' IN A PEAD STATEMENT. THIS MAY NOT BE ALLOWED IN SOME FORTRANS.

.....  
.....THE 360 TAPE.....  
.....

THE 360 P-STAT DISTRIBUTION TAPE IS A 9 TRACK, 2400 FOOT, 800 DENSITY TAPE WRITTEN WITH STANDARD LABELS ( I.E., USING SL ). THE VOLUME ID IS PS5205 ( FOR P-STAT, VERSION 52.5 ). THESE FILES WERE PRODUCED USING FORTRAN H (OPT2) AND O.S. VERSION 18. THE OVERLAYS INCLUDE THE EXTENDED ERROR MESSAGE FEATURE.

GENERALLY, UPON RECEIVING A TAPE, AN INSTALLATION SHOULD BEGIN BY PRINTING AND PUNCHING FILE 5, SEE BELOW.

THERE ARE 15 FILES, AS FOLLOWS.....

( NOTE - THE DSNAM OF THE FILE IS IN PRENS )

- 1 - (PSTAT52.MANUAL) THE PSTAT MANUAL, ABOUT 4600 CARDS. THIS IS THE INPUT TO THE MANUAL PROGRAM.
- 2 - (PSTAT52.IEMANUAL) THE P-STAT OPERATION.....  
MANUAL, TAPE = 91 \$  
PUTS PRINT LINE IMAGES INTO DATA SET 91 IN ADDITION TO JUST PRINTING IT. THIS IS THAT DATA SET, ABOUT 5000 132-CHARACTER RECORDS. IEGENER CAN THEN BE USED TO PRINT COPIES CHEAPLY.
- 3 - (PSTAT52.PRIMER) THE PRIMER, ABOUT 900 CARDS. THIS IS THE INPUT TO THE PRIMER PROGRAM.
- 4 - (PSTAT52.IEPRIMER) THIS FILE, CREATED BY PRIMPR,TAPE=91\$ CONTAINS ABOUT 600 132-CHARACTER RECORDS. IT CAN BE PRINTED USING IEGENER.
- 5 - (PSTAT52.DECKS) VARIOUS USFFUL DECKS, SEE BELOW. ABOUT 1000 CARDS.
- 6 - (PSTAT52.ASSEM) TWO ASSEMBLY LANGUAGE DECKS, TAPF99 AND AN ASSEMBLY LANGUAGE VERSION ( VERY FAST ) OF PSNVAL. TAPF99 IS A ROUTINE THAT CAUSES C.S. TO TREAT FILE 99 AS AN INTERNAL BUFFER, PERMITTING FAST REFORMATTING OF DATA WITHIN FORTRAN. ABOUT 100 CARDS.
- 7 - (PSTAT52.CONTROLS) A CARD IMAGE FILE OF THE OVERLAY CONTROLS. THIS DEFINES THE OVERLAY STRUCTURE. IT IS USED BY THE USER LINK CATALOGED PROCEDURE PSTAT52U. IT IS NEEDED TO MAKE ANY P-STAT OVERLAY. ABOUT 400 CARDS.
- 8 - (PSTAT52.OVERLAY) THE VERSION 52.5 OVERLAY. THIS CAN BE MOVED FROM THE TAPE TO A DISK AND EXECUTED.

A SAMPLE DECK WHICH USES IEHMOVE TO DO THIS IS IN FILE 5 ( PSTAT52.DECKS ). THE MAXIMUM SIZE OF THE OVERLAY BY ITSELF IS CURRENTLY 226K BYTES. AN EXECUTION REGION OF 250K (SPACE FOR BUFFERS, ETC.) SHOULD BE AMPLE.

- 9 - (PSTAT52.OVLARGE) THE VERSION 52.5 LARGE ( 600K ) OVERLAY.
- 10 - (PSTAT52.LIB) THE SOURCE, COMPILED AND LINK-EDITED INTO A PARTITIONED DATA SET, I.E., A LIBRARY.
- 11 - (PSTAT52.ASMLIB) THE COMPILED AND LINK-EDITED FORM OF FILE 6.
- 12 - (PSTAT52.ISRLIB) THOSE PARTS OF THE MICHIGAN OSIRIS LIBRARY NEEDED TO CONSTRUCT AN OVERLAY WITH OSIRIS ( OSIRIS.P ) IN IT.
- 13 - (PSTAT52.LARGELIB) A LIBRARY OF THE VERSIONS OF THOSE PROGRAMS WHICH ARE DIFFERENT IN THE LARGE OVERLAY.
- 14 - (PSTAT52.SOURCE) THE SOURCE CARDS, ABOUT 23,000.
- 15 - (PSTAT52.LSOURCE) THE SOURCE CARDS OF THE PROGRAMS USED IN THE LARGE OVERLAY. ABOUT 2600 CARDS. THESE ARE DUPLICATES OF 26 OF THE DECKS IN FILE 14, EXCEPT FOR LARGER DIMENSIONING.

THE FOLLOWING DCB'S DESCRIBE THE TAPE ORGANIZATION. THE DISK ORGANIZATION IS IN THE SAMPLE DECKS IN FILE 5.

DCB=(RECFM=FB,LRECL=80,BLKSIZE=800) IS USED FOR THE LOAD MODULE LIBRARIES AND THE OVERLAYS ( FILES 8-13 ).

DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200) IS USED IN THE CARD FILES ( FILES 1, 3, 5-7, 14-15 ).

DCB=(RECFM=FB,LRECL=132,BLKSIZE=1320) IS USED FOR THE PRINT IMAGE FILES ( 2 AND 4 ).

\*\*\*\*\*  
\* DESCRIPTION OF THE DECKS IN FILE 5 \*  
\*\*\*\*\*

- 1 - MOVE A PARTITIONED DATA SET ( OVERLAY OR LIBRARY ) FROM TAPE TO DISK.
- 2 - MOVE A CARD IMAGE FILE ( A SEQUENTIAL DATA SET ) FROM TAPE TO DISK.
- 3 - LIKE 2, BUT USABLE FOR THE PRINT LINE IMAGE FILES ( FILES 2 AND 4 ).
- 4 - A TEST DECK, WITH JOBLIB, TO SEE THAT THE OVERLAY WORKS.
- 5 - 4 CATALOGED PROCEDURES... PSTAT52 - USE THE 250K OVERLAY ( OR THE 600K OVERLAY ).

PSTAT52U - COMPILE A USER LINK, MAKE  
AN OVERLAY, EXECUTE IT.  
PSTAT52P - PRINT THE PRIMER FROM ITS  
PRINT IMAGE FILE.  
PSTAT52M - PRINT THE MANUAL FROM ITS  
PRINT IMAGE FILE

- 6 - COMPILE A SOURCE FILE AND MAKE A LIBRARY.
- 7 - COMPILE A SINGLE SOURCE DECK AND UPDATE THE LIBRARY.
- 8 - MAKE THE 250K OVERLAY.
- 9 - MAKE THE 600K OVERLAY.

## 25. A LARGER 360 VERSION

PRINCETON HAS A 360/91 WITH A 2 MILLION BYTE MEMORY, AND IT IS POSSIBLE TO RUN JOBS WITH A LARGE REGION SIZE. A SEPARATE OVFPLAY EXISTS WHICH PERMITS LARGER FACTOR ANALYSES, ETC. ( SEE BELOW ). IT NEEDS A REGION SIZE OF 600K BYTES.

THE FOLLOWING OPERATIONS MAKE USE OF THE INCREASED CORE ...

BISERIAL	15,000 ELEMENTS
BPRINT	300 BY 300
CLEANCOR	300 BY 300
F.COEF	300 VARIABLES, 100 FACTORS
GROUPC	250 VARIABLES, 75 FACTORS
INTCDS	300 VARIABLES
INTMDA	15,000 ELEMENTS
INTMDS	150 VARIABLES
INVERT	200 BY 200
ITFACT	250 VARIABLES
MATCH	100,000 ELEMENTS
MULTIPLY	POST CAN BE 300 BY 300
NEWFAC	250 VARIABLES, 75 FACTORS
PROMAX	250 VARIABLES, 75 FACTORS
ROTATE	300 VARIABLES, 100 FACTORS
SMERGE	300 BY 300
SSORT	100,000 ELEMENTS
TCOR	150 VARIABLES
TET	300 VARIABLES
TTEST	300 VARIABLES

AT PRINCETON, A DISK MUST BE MOUNTED AND THE REGULAR PROCEDURE IS USED, ADDING ONE FIELD..

```
/*SETUP UNIT=2314, ID=BBB201
// EXEC PSTAT52, SIZE=LARGE
```

THIS MODIFICATION INVOKES THE LARGE (600K) VERSION INSTEAD OF THE REGULAR ONE. ANY TAPE REFERENCES WOULD FOLLOW (OR PRECEDE), AS...

```
// EXEC PSTAT52, SIZE=LARGE, P61=1277, NAME61='TEST.DATA'
```

IT IS NOT POSSIBLE, USING JUST THE CURRENT CATALOGED PROCEDURES, TO ADD A USER LINK TO A LARGE TEMPORARY OVERLAY.

## 26. SYSTEM USE OF STORAGE

THE P-STAT SYSTEM WORKS WITH P-STAT FILES, AND THESE FILES MUST BE MAINTAINED (I.E., ORGANIZED, LOCATED) IN SOME MANNER.

THERE ARE, IN FACT, FOUR LEVELS.

- 1 - A LINK-LIST STORAGE AREA IN COPE, OR A DEFINE FILE (360 FORTRAN DIRECT ACCESS I/O) STRUCTURE ON A DISK, OR SOME OF BOTH.
- 2 - SCRATCH 'TAPES', EACH HOLDING A SINGLE P-STAT FILE. ON A 360, THESE ARE USUALLY DEFINED BY JCL TO BE ON DISK. 'ASSIGN/ATTACH' TAPES ARE A SPECIAL CASE OF THIS LEVEL.
- 3 - A BULK TAPE, SOMETIMES CALLED A TEMPORARY DATA TAPE, OR 'TDT'. THIS HAS MANY P-STAT FILES ON IT, ONE AFTER ANOTHER. THIS IS ALSO USUALLY ON DISK.
- 4 - A USER'S 'SAVE' TAPE, OR PERMANENT DATA TAPE.

FILES IN THE FOURTH LEVEL, A USER'S SAVE TAPE, ARE NOT ACCESSABLE TO THE P-STAT SYSTEM WITHOUT THE USE OF THE 'FIND' OPERATION. THIS PLACES COPIES OF THE DESIGNATED FILES IN ONE OF THE FIRST THREE LEVELS. THUS, P-STAT INTERNALLY HAS THREE LEVELS WITH WHICH TO WORK.

P-STAT WILL ALWAYS TRY TO PLACE A FILE IN THE FIRST LEVEL (LINK-LIST). THIS IS AN OPTIONAL LEVEL. THE SYSTEM IS READY TO USE IT. A SINGLE FLAG (KKDISC = 1, WHEN COMPILING SUBROUTINE PSINIT) INDICATES THE AVAILABILITY OF THIS TYPE OF STORAGE.

IF NOT THERE, OR FULL, A NEW FILE WILL BE PLACED ON A SCRATCH TAPE. IN VERSION 52 THERE ARE 9 OF THESE. THERE MAY OF COURSE ALREADY BE A FILE ON EACH OF THESE, AND SOME MAY BE ACTIVE (IN USE BY THE CURRENT P-STAT OPERATION) AT THIS MOMENT. SOME MUST BE DORMANT, HOWEVER, BECAUSE ONLY FOUR FILES CAN BE SIMULTANEOUSLY IN USE. THE SMALLEST DORMANT FILE IS COPIED ON THE END OF THE BULK TAPE (LEVEL THREE) AND THE NEW FILE GOES ON THE NOW AVAILABLE SCRATCH TAPE.

USE OF THE BULK TAPE (LEVEL 3) IS HORRIBLY INEFFICIENT. IF LEVEL 1 IS UNAVAILABLE, 9 FILES FILL LEVEL 2 AND LEVEL 3 STARTS BEING USED WITH 10 OR MORE FILES.

WHEN A FILE IN LEVEL 2 IS NEEDED, SOME PROGRAMS ARE ABLE TO TELL P-STAT INTERNALLY THAT THE FILE CAN SAFELY BE READ DIRECTLY FROM THE TDT. THIS IS NOT TOO BAD EXCEPT FOR THE TIME SPENT GOING THROUGH THE TAPE TO LOCATE THE FILE. THIS COULD BE EITHER READING OR BACKSPACING, WHICHEVER WILL BE QUICKER. IN OTHER CASES, A P-STAT LINK REQUIRES THE FILE TO BE IN LEVEL 2. ONE CANNOT READ ALTERNATE ROWS FROM TWO FILES, BOTH ON THE SAME TAPE. THERE MUST THEN BE COPYING FROM LEVEL 3 TO LEVEL 2. IF LEVEL 2 IS FULL, A COPYING OF A DORMANT FILE FROM LEVEL 2 TO LEVEL 3 PRECEDES THE LEVEL 3 TO LEVEL 2 COPY. ALL THIS TAKES TIME.

'PURGE' AND 'RETAIN' ARE AVAILABLE TO CLEAR OUT OBSOLETE FILES FROM THE SYSTEM. IF A PURGED FILE IS ALREADY ON THE BULK TAPE, THERE IS NO GAIN IN STORAGE USE, BUT IF IT WERE IN LEVEL 1 OR 2, THERE COULD BE CONSIDERABLE GAIN. VERSION 52 HAS TABLE SPACE FOR INFORMATION CONCERNING 70 FILES. PURGING ALSO HELPS TO STAY BELOW THIS TOTAL.

'ASSIGN/ATTACH' FILES ARE EFFICIENT FOR SEVERAL REASONS ...

- 1 - THEY ARE IMMEDIATELY USABLE.  
IF A FILE IS 'FOUND' ON A SAVE TAPE, IT IS COPIED INTO LEVEL 1 OR 2 AS PART OF THE 'FIND' STEP. FOR LARGE FILES THIS CAN TAKE TIME.  
ALSO, THE SEARCH OF THE 'SAVE/FIND' TAPE TAKES TIME. A FILE ON AN ATTACH TAPE IS IMMEDIATELY USABLE FROM THAT TAPE.
- 2 - THE ATTACH TAPE IS TREATED WITHIN P-STAT AS A SPECIAL TYPE OF LEVEL TWO TAPE, BUT DOES NOT TIE UP THE ORIGINAL 9 SCRATCH TAPES. IN OTHER WORDS, ITS USE DOES NOT MAKE OTHER ASPECTS OF THE SYSTEM LESS EFFICIENT.

AN ATTACH TAPE IS ORGANIZED EXACTLY LIKE A ONE FILE SAVE TAPE (I.E., A PDT WITH ONLY ONE FILE SAVED ON IT). A TAPE GENERATED BY ASSIGN COULD BE USED IN A LATER RUN AS A SAVE TAPE. ALSO, IF A SAVE TAPE HAS MANY FILES ON IT AND ONLY THE FIRST IS TO BE USED, THAT TAPE COULD BE USED IN A GIVEN RUN AS AN ATTACHED TAPE. OF COURSE, THESE 'TAPES' COULD ACTUALLY BE ON A DISK IN A USER'S PERMANENT STORAGE AREA, BUT THEY WILL USUALLY BE TAPES.

ON A 360, THERE ARE PERHAPS 3 WAYS OF USING LINKLIST STORAGE. THIS STORAGE USES BLOCKS OF STORAGE, EACH 465 WORDS LONG. THESE BLOCKS ARE NUMBERED 1 THROUGH N, WHERE N IS THE NUMBER OF SUCH BLOCKS THAT WAS SPECIFIED IN SUBROUTINE PSINIT. P-STAT KNOWS THE LOCATION OF THE FIRST SUCH BLOCK OF A FILE, AND THAT BLOCK HAS WITHIN IT THE ADDRESS OF THE NEXT BLOCK. ASSUME FILE 'A' USES BLOCK 1, 'B' USES BLOCKS 2 AND 3, THEN 'A' IS PURGED, AND FILE 'C' IS CREATED, NEEDING 3 BLOCKS. IT WILL GET 1, 4, AND 5. WHEN A FILE, BEING PLACED IN THESE BLOCKS, NEEDS BLOCK N+1, THE SYSTEM FINDS THE LARGEST FILE IN THE LINKLIST AREA, INCLUDING THAT PART OF THE CURRENT FILE ALREADY INTO BLOCKS, AND MOVES IT TO LEVEL 2.

THE BEST ALLOCATION ( WITHOUT CONSIDERING FREQUENCY OF USE ) IS SHORT FILES IN LEVEL 1, LONG FILES IN LEVEL 2, AND SHORT FILES IN LEVEL 3. IF, WHEN TRYING TO GO FROM LEVEL 1 TO LEVEL 2, LEVEL 2 IS FULL, THE SHORTEST FILE NOT IN USE DURING THE CURRENT STEP IS MOVED TO LEVEL 3, PERMITTING A 1 TO 2 MOVE, FINALLY PERMITTING THE OUTPUT OF THE CURRENT FILE TO CONTINUE.

#### THREE WAYS TO ORGANIZE LINKLIST STORAGE

ONE WAY IS LABELLED COMMON AREAS, DIMENSIONED 465 BY N. THIS, OF COURSE, IS VERY EFFICIENT, BUT USES A LOT OF CORE. ANOTHER WAY INVOLVES 360 DIRECT ACCESS I/O. SUBROUTINE 'PSWRDX' WOULD HAVE A 'DEFINE FILE' STATEMENT. THIS STATEMENT CAUSES OS/360 TO INITIALIZE THE BLOCKS ON A SCRATCH DISK. IF 1000 BLOCKS ( ABOUT 15 CYLINDERS )

ARE ASKED FOR, IT TAKES SEVERAL MINUTES. THE 'DD' CARD IN THE CATALOGED PROCEDURE WOULD ASK FOR NEW SCRATCH SPACE TO BE ALLOCATED EACH RUN.

A THIRD WAY WOULD INVOLVE WRITING SOME 360 ASSEMBLY LANGUAGE ROUTINES TO GAIN THE EFFECT OF 'DEFINE FILE' TYPE USAGE WITHOUT THE INITIALIZATION TIME. THIS IS BEING CONSIDERED.

VERSION 52 EQUIVOCATES - IT USES 100 SCRATCH TRACKS, AND INITIALIZES THEM AT THE START OF EACH P-STAT RUN. THE SPECIFICATION OF 100 IS FOUND IN THE CATALOGED PROCEDURE ( THE DD CARD FOR FILE 46 ) AND IN SUBROUTINE PSWRDX. NOTE - USE OF THE STATEMENT 'USEDISK = 0 \$' AT THE START OF A P-STAT RUN TURNS OFF LINKLIST USE OF THE DISK ENTIRELY.

## 27. ACKNOWLEDGMENTS

THE AUTHOR'S INVOLVEMENT WITH STATISTICS AND COMPUTING BEGAN IN 1959 AT RUTGERS WITH SOME IBM 650 RUNS FOR PROFESSOR J.R. WITTENBORN, WHOSE USE AND ENCOURAGEMENT OF P-STAT AS IT DEVELOPED HAS CONTINUED DURING THESE 10 YEARS. PROFESSORS O.K. BUROS AND F. FENDER WERE ALSO MOST HELPFUL IN THOSE FIRST VITAL YEARS.

THE NEXT STEP WAS COMPLETING A MATRIX PACKAGE ON THE RCA 501, WORKING WITH F. LORD AND H. GULLIKSEN, AT EDUCATIONAL TESTING SERVICE. P-STAT AS SUCH BEGAN IN 1963 ON THE 7090 AT PRINCETON UNIVERSITY, AND MANY PRINCETON COMPUTER USERS HAVE INFLUENCED ITS PROGRESS, INCLUDING C. BINGHAM, C. HELM, G. SANDE, W. STUDDIFORD, H. TROTTER, AND J. TUKEY.

THE USE OF A 1301 DISK ON THE 7094 FOR LIBRARY UPDATING, P-STAT RESIDENCE ( RATHER THAN A CHAIN TAPE), AND LINK-LIST DATA STORAGE GREATLY HELPED THE PROGRESS OF THE P-STAT SYSTEM. A. JONES AND L. VARIAN HELPED DEVELOP THESE FACILITIES.

SOME OF THE PROGRAMS DID NOT ORIGINATE AT PRINCETON, INCLUDING PARTS OF PLOT, MULTR, AND POTATF (BIMED), INVERT (NYU), TET (ETS), AND AN EIGENROOTS PROGRAM (UNION CARBIDE). THE CROSSTAB PROGRAM AND THE PRIMER WERE WRITTEN AT PRINCETON BY SHIRRELL BUHLER.

THE CATALOGED PROCEDURES FOR VERSION 52 WERE DEVELOPED BY SHIRRELL BUHLER.

APPRECIATION FOR HELP WITH THE P-STAT PROJECT IS DUE TO PETER SCHAY, DOUG AND GREG TOMLINSON, BOB MILLS, BILL MCCUSKER, NANCY SCOTT, AND PARTICULARLY TO PAM TOMLINSON AND ELLIOTT FRANK.

R. NUTTAL OF BOSTON COLLEGE HELPED IN DEVELOPING THE OSIRIS LINK. J. RATTENBURY AND J. SONQUIST OF THE OSIRIS PROJECT HAVE BEEN MOST COOPERATIVE IN THIS ATTEMPT TO LINK SYSTEMS TOGETHER BY PERMITTING USE OF PART OF THEIR SUBROUTINE LIBRARY.

F. KEPLINGER AND V. FREE OF NYU SUPPORTED THE CONVERSION OF AN EARLY P-STAT VERSION ( 42 OR SO ) FROM THE 7094 TO THE CDC 6600. C. COPE HELPED ON THE NYU PROJECT.

J. BLYNN OF BELL LABS WORKED WITH A PROTOTYPE OF VERSION 51 ON A G.E. 635. MUCH WAS LEARNED ABOUT 'CLEAN FORTRAN' ON THE 6600 AND 635.

W. GOPHAM AND H. OZER OF THE CIVIL SERVICE COMMISSION HAVE MADE A NUMBER OF USEFUL SUGGESTIONS DURING THEIR YEARS OF P-STAT USE ON 7094'S AND 360'S.

THE COMPUTING ENVIRONMENT AT PRINCETON SINCE 1962 HAS PROVIDED A MOST FRUITFUL OPPORTUNITY TO DEVELOP THIS SYSTEM.

THE OFFICE OF NAVAL RESEARCH SUPPORTED P-STAT DURING 1965-1967. WITHOUT THE TIME ( TWO YEARS, HALF TIME ) PROVIDED BY THIS SUPPORT, P-STAT WOULD PROBABLY NOT HAVE PROGRESSED BEYOND ITS FIRST FORM.

## 28. A LIST OF THE LEGAL IDENTIFIERS FOR EACH OPERATION

THIS DOES NOT INCLUDE OPERATIONS WHERE THE ONLY IDENTIFIER IS THE OPERATION NAME ITSELF, SUCH AS HEAD OR BPRINT.

NOTE - A \* BEFORE THE OPERATION NAME IS A REMINDER THAT A \*END CARD MUST FOLLOW THE LAST DATA CARD FOR THAT OPERATION.

OPNAME	IDEN	IDEN	IDEN	IDEN	IDEN
ASSIGN	TAPE				
ATTACH	TAPE				
BISERIAL	IN	OUT	NCV	ZERO	
CLEANCOP	IN	OUT	DELETE		
COLLATE	LEFT	RIGHT	OUT		
COPY.PDT	TAPE	NEWTAPE			
CORESORT	IN	OUT	LABEL		
*CROSSTAB	IN TOTPCT PZ	VAR CONTROLS CUMPOW	MEANS LABELS CUMCOL	COLPCT COMBOS SUMS	ROWPCT WEIGHT
*DATA	DES CARDS	NV PRINT	TAPE MINUS	EPRINT PLUS	*KILL
DIP	IN	OUT	V1	V2	STEP
DISCRIM	IN	OUT	NG	NVA	FUN
DUPWAT	IN	OUT			
E.ADD	FILE1	FILE2	OUT		
E.SUB	FILE1	FILE2	OUT		
E.MULT	FILE1	FILE2	OUT		
E.DIV	FILE1	FILE2	OUT		
EDIT	TAPE	LABELS			
E.COFF	SDATA RADINPUT	FACTOR	FC	PMULT	MEAN
FIND	TAPE				

FREQ	IN NCAT	NG EQUALCAT	DES	ALLCAT	ALLGPP
FREQ.C	IN NCAT	NG EQUALCAT	DFS	ALLCAT	PAIR
*GFNVAR	IN	OUT	DES	NC	
GROUPCOR	INCOR	INFAC	OUTCOR	OUTFAC	
INTCDS	IN INCEPT	CROSS NOROW1	COV	COR	SLOPE
INTMDA	IN ROWS	NMAT NOROW1	CROSS	COV	COR
INTMDS	IN SLOPE	NMAT INCEPT	CROSS NOROW1	COV	COR
INVERT	IN	OUT			
ITFACT	IN DROOT LARGE	ROOTS TEST	VEC NFAC	FAC ONE	SF ZERO
IAG	IN	OUT	V1	V2	STEP
LPFILL	LEFT	RIGHT	OUT		
MATCH	CONTROL	IN	OUT		
MERGE	UP NOMATCH	DOWN OUT	LEFT	MIDDLE	RIGHT
MUDMERGE	IN	OUT	NOMATCH		
MULTIPLY	PRE	POST	OUT		
*MULTR	COR	DES			
NEWFAC	INCOR	INFAC	OUTFAC		
NEWPDT	TAPP				
NO.OP	IN	OUT			
NORM.COL	IN	OUT			
NORM.ROW	IN	OUT			
OSIRIS.P	OUT				
PBYQ	IN	ROW	COL	FIRST	

*PLOT	ONE	SYM			
*PPSOURCE					
PRIMER	PUNCH				
PRINT	PRINTLOW				
PROMAX	INFAC	JUTFAC	COR	TRANS	COSINE
PUNCH	CARDS				
RESIDU.1	IN	SLOPE	INCEPT	OUT	
ROTATE	IN VTR	QP ETR	VF	EF	QTR
SAVE	TAPE				
SCAN	IN OUT	DES STAYMISS	INDES	M DATA	S DATA
*S DATA	NV	FORMAT	PRINT	TAPE	
SMERGE	A11	A12	A22	OUT	
TCOR	IN	OUT			
TET	IN CTFT	CROSS	TET	SPLIT	ZERO
TRANSPOS	IN	OUT			
TRIM	TAPE				
TTEST	IN	OUT			

\*\*\*\*\*

TABLE OF CONTENTS.

\*\*\*\*\*

1.	INTRODUCTION	PAGE	1
2.	CHANGES WITHIN RELEASES OF VERSION 52.	PAGE	3
	VERSION 52.3 (APRIL, 70) HAS THESE CHANGES...	PAGE	4
	VERSION 52.4 ( NOVEMBER, 70 ) HAS THESE CHANGES..	PAGE	5
	VERSION 52.5 ( JAN., 71 ) HAS THESE CHANGES...	PAGE	6
	PLANS FOR VERSION 53.	PAGE	7
3.	THE P-STAT CONTROL LANGUAGE	PAGE	8
4.	SELECTING SPECIFIC ROWS AND/OR COLUMNS FROM A FILE	PAGE	14
5.	SYSTEM-INVOLVED OPERATIONS	PAGE	17

C	PAGE	17
FND	PAGE	17
HEAD	PAGE	17
MANUAL	PAGE	17
MAXERROR	PAGE	18
PPSOURCE	PAGE	18
PRIMER	PAGE	19
PURGE	PAGE	19
RETAIN	PAGE	19
SOURCE	PAGE	20
XHEAD	PAGE	20

6. INPUT-OUTPUT OPERATIONS

PAGE 21

BPRINT	PAGE	21
OSIRIS.P	PAGE	21
PLOT	PAGE	22
PRNT	PAGE	22
PUNCH	PAGE	23
SDATA	PAGE	24

7. SAVING AND USING P-STAT FILES ON TAPES

PAGE 25

NEWPDT	PAGE	25
SAVE	PAGE	25
FIND	PAGE	26
COPY.PDT	PAGE	26
EDIT	PAGE	27
TRIM	PAGE	27
ASSIGN	PAGE	27
ATTACH	PAGE	28
REWIND	PAGE	28

8. FILE MANIPULATION OPERATIONS

PAGE 29

COLLATE	PAGE	29
CORESORT	PAGE	29
DIF	PAGE	30
LAG	PAGE	30
LRFILL	PAGE	30
MATCH	PAGE	31
MERGE	PAGE	31
MUDMERGE	PAGE	32
NO.OP	PAGE	32
NORM.COL	PAGE	32
NORM.POW	PAGE	32
SCAN	PAGE	33
SMERGE	PAGE	33

9. MATRIX OPERATIONS

PAGE 35

F.ADD	PAGE	35
F.SUB	PAGE	35
F.MULT	PAGE	35
F.DIV	PAGE	35
INVERT	PAGE	35
MULTIPLY	PAGE	36
TRANSPOS	PAGE	36

10. CORRELATION OPERATIONS

PAGE 37

BISERIAL PAGE 37  
INTCDS PAGE 38  
INTNDA PAGE 38  
INTMDS PAGE 39  
TPT PAGE 40

11. FACTOR ANALYSIS

PAGE 42

CLEANCOR PAGE 42  
P.CORP PAGE 42  
GROUPCOR PAGE 43  
ITPACT PAGE 44  
NEWFAC PAGE 46  
PROMAX PAGE 46  
ROTATE PAGE 46

12. CROSS-TABULATION ( CPCSSTAB )

PAGE 48

CROSSTAB PAGE 48

13. DISCRIMINANT ANALYSIS ( DISCRIM )

PAGE 53

DISCRIM PAGE 53

14. FREQUENCY DISTRIBUTIONS ( FREQ )

PAGE 56

FREQ PAGE 56  
FREQ.C PAGE 59

15. REGRESSION ( MULTF )

PAGE 59

MULTR PAGE 59  
RESIDU.1 PAGE 60

16. TWO WAY ANALYSIS OF VARIANCE ( PBYQ )

PAGE 62

PBYQ PAGE 62

17. OTHER PROGRAMS

PAGE 66

DURWAT PAGE 66  
GENVAR PAGE 66  
TCOR PAGE 68

TTEST PAGE 69  
VBAR PAGE 69

18. CREATING A P-STAT FILE FROM CARDS ( DATA )	PAGE 70
DATA PAGE 70	
19. DATA MODIFICATION AND GENERATION USING FOR/IF/SET/SETX	PAGE 78
20. MACPOS OF P-STAT OPERATIONS	PAGE 92
21. DO LOOPS OF P-STAT OPERATIONS	PAGE 98
22. JCL FOR REFERENCING TAPES ON THE 360	PAGE 100
23. ADDING USER-WRITTEN TEMPORARY LINKS TO THE SYSTEM	PAGE 104
24. ORGANIZATION OF P-STAT DISTRIBUTION TAPES	PAGE 114
25. A LARGER 360 VERSION	PAGE 118
26. SYSTEM USE OF STORAGE	PAGE 119
27. ACKNOWLEDGMENTS	PAGE 122
28. A LIST OF THE LEGAL IDENTIFIERS FOR EACH OPERATION	PAGE 125